

57-1500

MA-BHA

Siemens Automation Cooperates with Education | 05/2017

**SIEMENS** 

## Módulo TIA Portal 032-200 Principios básicos de la programación FB con SIMATIC S7-1500



Libre utilización para centros de formación e I+D. © Siemens AG 2017. Todos los derechos reservados.

#### 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: <u>www.siemens.com/sce/tp</u> y <u>www.siemens.es/sce</u>

#### **Cursos avanzados**

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

#### 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 <u>roland.scheuerer@siemens.com</u>.

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		Obje	etivos	. 5
2		Req	uisitos	. 5
3		Hard	dware y software necesarios	. 6
4		Тео	ría	. 7
	4.1	1	Sistema operativo y programa de usuario	. 7
	4.2	2	Bloques de organización	. 8
	4.3	3	Imagen de proceso y ejecución cíclica del programa	. 9
	4.4	4	Funciones	11
	4.5	5	Bloques de función y bloques de datos de instancia	12
	4.6	6	Bloques de datos globales	13
	4.7	7	Bloques lógicos aptos para librería	14
	4.8	3	Lenguajes de programación	15
5		Tare	ea planteada	16
6		Plar	ificación	16
	6.′	1	PARADA DE EMERGENCIA	16
	6.2	2	Modo automático: Motor de cinta	16
7		Insti	rucciones paso a paso estructuradas	17
	7.′	1	Desarchivación de un proyecto existente	17
	7.2	2	Creación de una nueva tabla de variables	18
	7.3	3	Creación de nuevas variables dentro de una tabla de variables	20
	7.4	4	Importación de la "Tag table_sorting station (Tabla_variables_planta_clasificación)"	21
	7.5	5	Creación del bloque de función FB1 "MOTOR_AUTO" para el motor de cinta en modo automático	25
	7.6	6	Definición de la interfaz del FB1 "MOTOR_AUTO"	27
	7.7	7	Programación del FB1: MOTOR_AUTO	30
	7.8	3	Programación del bloque de organización OB1: control de la marcha de la cinta hacia delante en modo automático	38
	7.9	9	En el lenguaje de programación KOP (Esquema de contactos), el resultado es como sigue	43
	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	49
8		Lista	a de comprobación	50
9		Ejer	cicio	51
	9.1	1	Tarea planteada: ejercicio	51
	9.2	2	Planificación	51
	9.3	3	Lista de comprobación: ejercicio	52
1(	)	Info	rmación adicional	53

## PRINCIPIOS BÁSICOS DE LA PROGRAMACIÓN 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 lo utilizará para programar 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 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

- 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
- 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:

 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: Bloque de función apto 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.

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 a 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 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_automático_activo	BOOL	Modo de operación Modo automático activado
Comando_arranque	BOOL	Comando de arranque para el modo automático
Comando_parada	BOOL	Comando de parada para el modo automático
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_modo_automático	BOOL	Control del motor de cinta en modo automático
Static		
Memoria_modo_automático_arranque/parada	BOOL	Memoria para la función de parada y arranque en modo automático

Tabla 2: Parámetros de FB "MOTOR\_AUTO"

El parámetro Memoria\_modo\_automático\_arranque/parada se conecta con memoria con el comando Arranque, pero solo si no se cumplen las condiciones de reseteo.

El parámetro Memoria\_modo\_automático\_arranque/parada se desactiva si está presente el comando Parada, la desconexión de seguridad está activa o el modo automático no está activado (operación manual).

La salida Motor\_cinta\_modo\_automático se controla cuando está activado el parámetro Memoria\_modo\_automático\_arranque/parada y se cumplen las condiciones de habilitación.

### 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, limítese 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\_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))

Project Edit View	Insert	Online	Options						
<ul> <li>New</li> <li>Open</li> <li>Migrate project</li> </ul>			Ctrl+0						
Close			Ctrl+W						
Save			Ctrl+S						
Save as		Ctrl	+Shift+S						
Delete project Archive			Ctrl+E						
Retrieve									
Card Reader/USB me Memory card file	emory		;						
Upgrade									
D:lAutomation\013_ D:lAutomation\012_ D:lVorlagenprojekt_ D:l\032-200_FB-Pro D:lAutomatisi\012	10\013 10\012 Webserv ogrammic -100_CPU	_101_CPU _101_CPU \Tank_V1 erung_S7- 11500_V13	314C 1516F 3_SP1 314 3_SP1						
Exit									

→ 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)



 $\rightarrow$  A continuación, ábrala haciendo doble clic. ( $\rightarrow$  Tag table\_sorting station

(Tabla\_variables\_planta\_clasificación))

M Siemens - D:\Automation\012_101_CPU151	6F\012_101_CP	U1516F	_					_	_		Deperties				
Save project = X = - X	10015 WINDOV	UNG BUG	🖁 🍯 Go online	Co online & Co office A Contract Automation SF > CPU_1516F (CPU 1516F-3 PN/DP) > PLC tags > Tag table_sorting station (0) Tags © User constants x mg station W Constants Tag Constants W Constants Comment Comm											
Project tree	□ ◀ 0	12_101_CP	U1516F • CF	PU_1516F [	CPU 1516F-	3 PN/DP] → PL	Ctags ▶ Ta	ag table	_sorting	station [0]				_ # =>	< <
Devices											4	Tags	🔳 Use	r constants	1
1900		0 0 🕞	92 AX												-
5		Tag table	sorting static	n											KS
▼ 1 012 101 CPU1516F		Nam			Data type	Address	Retain	Visibl	Acces	Comment					
Add new device	1		add news		boto type		The contra			connent					- Ц
Bevices & petworks	-					130									5
CPU 1516E [CPU 1516E-3 PN/DP]															Idt
P Device configuration															sat
Q Online & diagnostics															
Program blocks	=														
Technology objects															11
External source files															11
PLC tags								Totally Integrated Automation PORTAL Tag table_sorting station (0) Tags © User constants in Visibl_ Acces Comment Properties 1/1 Info Diagnostics ed Properties							
Show all tags											Totally Integrated Automation PORTA Tags User constants	1			
Add new tag table															11
Standard-Variablentabelle [54]															1
Tag table sorting station [0]															11
PLC data types															
Watch and force tables									0	Properties	*i Info	N. Dia	anostics		7
Online backups		e 1	12									1200	- <u>g</u>	and the second second	-
🕨 📴 Traces		General													
Program info		Tag		Tag										1	^
Device proxy data				rug											5
PLC alarms				Ge	eneral										
E) Text lists															
Local modules				1		Name:									
Common data						Data type:								(3)	
Documentation rettinor	~			Ĥ		Address	[								
✓ Details view						nuoress.	Patainad								
							Netomed								
Data type Comment Name				_		Comment:	L								-
A Bostal store	Tag table so										2 Province also				

#### 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)

							🕣 Tags	User constants
Tags User constants      Tag table_sorting station      Name     Data type     Address     Retain     Visibl     Acces     Comment      Q1     Sol     Sol								
Tan	table_sorting st	ation						
·ug								
·ug	Name	Data type	Address	Retain	Visibl	Acces	Comment	
.ug	Name Q1	Data type Bool	Address %I0.0	Retain	Visibl	Acces	Comment	

→ 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 → 🜌)

ro	gram	ming ► CPU_1	1516F [C	PU	1516F-3	PN/	DP] 🕨	PLC tag	s ► Tag	g table_so	rting st	ation [1]	_ <b>=</b> = ×
										-	Tags	🗉 Use	r constants
		🕈 😤 🖬 🗲											
1	Tag ta	ble_sorting st	ation										
	1	lame	Data t	ype	Address		Retain	Visibl	Acces	Comment			
1	-	Q1	Bool		%10.0	-							
2		<add new=""></add>				c	Operand Oper Bi	identifier: rand type: Address: t number:	I Q M				
											<b>V</b>	×	

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

rc	ogram	nming + CPU_1	516F [CPU 1	1516F-3 PN	DP] 🕨	PLC tag	s ▶ Tag	g table_sorting st	ation [1]	_ <b>= =</b> ×
								🕣 Tags	🔳 User	constants
-	2 <sup>3</sup>	🖻 😤 🕅								
	Tag ta	able_sorting sta	ation							
		Name	Data type	Address	Retain	Visibl	Acces	Comment		
1	-	Q1	Bool 🔳	%Q0.0 💌				conveyor motor -M1	ation [1] _ I I I I I I I I I I I I I I I I I I	
2		<add new=""></add>		- 62 - 63 - 64		<ul> <li>Image: A start of the start of</li></ul>				
rograf Tag 1 2						-	0			

→ 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)".

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

								Tags	User constants
20	*	🖻 😤 🕅 🗶							2
1	lag t	able_sorting st	ation						
		Name	Data type	Address	Retain	Visibl	Acces	Comment	
	-	Q1	Bool	%Q0.0				conveyor motor -M1	forwards fixed speed
	-	Q2	Bool I	%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)". Seleccione en el menú contextual "Import file (Importar archivo)".

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

WA	Siemens - D:\Automation\032_100_FC-Progra	mming\0	32_10	0_FC-F	rogrammir	ıg					
Pr	oject Edit View Insert Online Options P 🎦 🔒 Save project 블 💥 🗐 🗎 🗙 🌖	Taols W	indow	Help	2 5 ø	Go or	nline 🚀 Go d	offline		× =	
	Project tree			ogram	ming 🕨 CF	U_1	516F [CPU	1516F-3 F	N/DP] 🕨	PLC tag	s ▶ T
	Devices										
	10 0 0	(m) -3		-45 1		1					
	000		9		<b>⊒7</b>   76 UX						
ing.		-		Tag ta	ble_sortin	g sta	ation				
	032_100_FC-Programming	^		1	Vame		Data type	Address	Retain	Visibl	Acces
	Add new device		1	-00	Q1		Bool	%Q0.0	1000		
Ē	Devices & networks		2	•	Q2		Bool	%Q0.1			
5	CPU_1516F [CPU 1516F-3 PN/DP]		3		<add new<="" td=""><td>&gt;</td><td></td><td></td><td></td><td>~</td><td>4</td></add>	>				~	4
Ч	Device configuration	Ξ				351	insert row				
	😟 Online & diagnostics					₹° /	Add row				
	Program blocks					X	Cut		Ctr	1+X	
	Technology objects					11日(	Сору		Ctr	14C	
	External source files					面	Paste		Ctr	l+V	
	<ul> <li>PLC tags</li> </ul>					×	Delete			Del	
	a Show all tags					1	Rename			F2	
	🚔 Add new tag table							71 B. PR. 1993	et to a		
	💥 Standard-Variablentabelle [54]					3	cross-reterenc	e informati	on Suut+r	23	
	🔚 Tag table_sorting station [2]	1				00.1	Monitor all				
	PLC data types						mpart file				
	Watch and force tables					E	Export file			_	
	Online backups					Fell a	Properties				
	🕨 📴 Traces					3	ropentes				
	📴 Program info										

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

 $(\rightarrow$  SCE\_ES\_020-100 Tag table\_sorting station...  $\rightarrow$  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).

Import co	mpleted with warnings (0032:000031) X
	Import completed with warnings.
	Detailed information is shown in the import log file.
	Click here to view the log file.
	ОК

- → 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.

 $(\rightarrow$  Clic con el botón derecho en las variables seleccionadas  $\rightarrow$  Delete (Eliminar))

							•	Tags	User constants	ę.
di.	🗎 😤 🖬								-	
Tag	table_sorting	station								
	Name	Data tj	pe Address	Retain	Visibl	Acces	Comment			
-	Q1	Bool	≣ %Q0.0	-			conveyor motor -M1 forwards fixed speed			-
-6	a 02	Bool	\$00.1	-			conveyor motor -M1 backwards fixed speed			
The second	insert row						return signal emergency stop ok (nc)			
m.	Add row						main switch _ON" (no)			
X	Cut		Ctrl+X				mode selector manual(0) / automatic(1)			
睻	Сору		Ctrl+C				pushbutton automatic start (no)			
	Paste		Ctrl+V				pushbutton automatic stop (nc)			-
×	Delete		Del				sensor cylinder -M4 retracted (no)			
-	Rename		F2				sensor cylinder -M4 extended (nc)			
	Crock reference	information	chi4.511				sensor motor -M1 actice (pulse signal for positioning) (no	)		
	crossreierence	mormation	Shint+Pit I				sensor part at slide (no)			
-	Monitor all			0			sensor metal part (no)			L
	Import file						sensor part in front of cylinder -M4 (no)			
	Export file						sensor part at end of conveyor (no)			
d	Properties						pushbutton manual mode conveyor -M1 forwards (no)			
2	a -24	0001					pushbutton manual mode conveyor -M1 backwards (no)			
4	-55	Bool	%11.6	10			pushbutton manual mode cylinder -M4 retract (no)			
-	-56	Bool	%11.7				pushbutton manual mode cylinder -M4 extend (no)			
-	-Q1	Bool	%Q0.0				conveyor motor -M1 forwards fixed speed			
-	-Q2	Bool	%Q0.1				conveyor motor -M1 backwards fixed speed			
-	-Q3	Bool	%Q0.2	0			conveyor motor -M1 variable speed			
	-M2	Bool	%Q0.3				cylinder -M4 retract			
-	-M3	Bool	%Q0.4				cylinder -M4 extend			
-	-P1	Bool	%Q0.5				display "main switch on"			
-	-P2	Bool	%Q0.6				display_manual mode"			
-6		Bool	\$600.7				display automatic mode"			1

→ 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.

 $(\rightarrow$  Project (Proyecto)  $\rightarrow$  Save as... (Guardar como...)  $\rightarrow$  032-200 FB Programming  $\rightarrow$  Save (Guardar))

* New	) ±	C#±		16 8	🗔 🍠 o	io online 🖉	Go offline 🔒	2 III II	×			Totally Int	egrated Automation PORTA
Open Ctr	1+0	Π 4	017	101	CPU1516	F CPU 1	516F [CPU 1	516E-3	PN/DP1	▶ PICt	ans  Tag table sorting station [28]		- 2 - 2
Migrate project	LOW .			_			Stor [di o i	5101 0	THE T		age ing ablo_sering station (co)		
		-										a Tags	User constants
Save Ctr	1+S		-	1	🕈 👶 🕅								
Save as Ctri+Shi	1+5			lag ta	ble_sortin	g station							
Delete project Ctr	rl+E	^		N	ame	Data typ	e Address	Retain	Visibl	Acces	Comment		
Archive			1	-	-A1	Bool	10.0				return signal emergency stop ok (nc)		
Retrieve			2	-0	-КО	Bool	%10.1				main switch "ON" (no)		1
Card Reader/USB memory	•		3	-	-50	Bool	%10.2				mode selector manual(0) / automatic(1)		
Memory card file	- F.		4	-	-51	Bool	%10.3				pushbutton automatic start (no)		
Upprade			5	-	-52	Bool	%10.4				pushbutton automatic stop (nc)		
	1.4	=	6	-	-81	Bool	%10.5				sensor cylinder -M4 retracted (no)		
Print Cti	rl+P		7	-	-B2	Bool	%10.6				sensor cylinder -M4 extended (nc)		
Print preview			8	-01	-83	Bool	%10.7				sensor motor -M1 actice (pulse signal for		
D:lAutomation\012_10\012_101_CPU1516	F		9	-	-84	Bool	%11.0				sensor part at slide (no)		
C:lUsers\spe\Do\032_100_FC-Programmin	9		10	-03	-85	Bool	%11.1				sensor metal part (no)		
D:lAutomation\013_10\013_101_CPU3140			11	-00	-86	Bool	%11.2				sensor part in front of cylinder -M4 (no)		
D:\Vorlagenprojekt_Webserv\Tank_V13_SP	Ч		12	-03	-87	Bool	%11.3				sensor part at end of conveyor (no)		
D:11032-200_FB-Programmierung_S7-314	-		13	-01	-53	Bool	%11.4				pushbutton manual mode conveyor -M1		
D:Automatisi1012-100_CPU1500_V13_SP1			14	-01	-54	Bool	%11.5				pushbutton manual mode conveyor -M1		
Exit			15	-00	-55	Bool	%11.6				pushbutton manual mode cylinder -M4 re		
Online backups			16	-00	-56	Bool	%11.7				pushbutton manual mode cylinder -M4 ex		
Traces			17	-01	-Q1	Bool	%Q0.0				conveyor motor -M1 forwards fixed speed		
Program info			18	-0	-Q2	Bool	%Q0.1				conveyor motor -M1 backwards fixed speed		
Device proxy data			19	-00	-Q3	Bool	%Q0.2				conveyor motor -M1 variable speed		
PLC alarms			20	-	-M2	Bool	%Q0.3				cylinder -M4 retract		
Text lists			21	-	-M3	Bool	%Q0.4				cylinder -M4 extend		
Local modules			22	-00	-P1	Bool	%Q0.5				display_main switch on"		
🕨 🙀 Common data			23	-03	-P2	Bool	%Q0.6				display "manual mode"		
Documentation settings		~	24	-01	-P3	Bool	%Q0.7				display_automatic mode"		
✓ Details view			25	-00	-P4	Bool	%Q1.0				display "emergency stop activated"		
			26	-67	-85	Bool	%011				display, automatic mode started"		
Data type Comment Name											Properties	Info 1 8 Dia	agnostics

# 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.

 $(\rightarrow$  PLC programming (Programación de PLC)  $\rightarrow$  Add new block (Agregar nuevo



emens - chosersispewocument	avatomatialerungio32_200_rb-rrogramm	ingosz_zoo_ro-rogramming	
			Totally Integrated Automation PORTAL
tart 🦓	Device: CPU1516F	Add new block	
Devices &	Show all objects	Name: Block_1	E
PLC programming	🥚 Add new block	Language: FBD V DB Number: 1 C	
Motion & 🔅		Organization Manual	
Drive parameterization	Show cross-references	Description: Functions are code blocks or subroutines without dev	dicated memory.
Visualization	Show program structure		
Online & Standard Contract Online & Standard Con	1 have	Function	
	Melp		
		Data block More	
		> Additional information	
		Add new and open	Add

→ Asigne al nuevo bloque el nombre: "MOTOR\_AUTO", seleccione el lenguaje FDB (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)".

 $(\rightarrow \text{Name (Nombre): MOTOR_AUTO} \rightarrow \text{Language (Lenguaje): FDB (FUP)} \rightarrow \text{Number}$ (Número): Automatic (Automático)  $\rightarrow \blacksquare$  Add new and open (Agregar y abrir)  $\rightarrow$  Add (Agregar))

Add new block				
Name:				
MOTOR_AUTO				
	Language:	FBD		
OB	Number:	1		
Organization		🔘 Manual		
DIOCK		<ul> <li>Automatic</li> </ul>		
	Description:			
гв	Function blocks are	code blocks that store their val	ues permanently in instanc	e data blocks,
Function block	so that they remain	available after the block has be	en executed.	
FC				
Function				
БВ				
Data block				
	More			
> Additional inform	nation			
Add new and open				Add

#### 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.

РО	MC	TOR AUTO					
		Name	Data type	Default value	Retain	Accessible f	Visible in
1	-0	▼ Input					
2		Add new>			*		
3	-	▼ Output			-		
4		Add new>				a	
5	-	▼ InOut					
6		Add new>					
7	-0	▼ Static					
8		Add new>					
9	-0	▼ Temp					
10		Add new>					
11	-0	▼ Constant					
12		Add new>					
	<		111	-			3
•	Blo	ck title:		- Incom			1
C	om	ment					-
•	1	Network 1:					
	C	Comment					

→ Para controlar el motor de cinta se requiere una señal de salida binaria. Por ello debe crearse previamente la variable de salida local #Conveyor\_motor\_automatic\_mode (Motor\_cinta\_modo\_automático), del tipo "Bool". Agregue al parámetro el comentario "Control of the conveyor motor 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 motor in automatic mode (Control del motor de cinta en modo automático))

03	2-2	00	_FB-Programming   CPU_1516F	[CPU 15	16F-3 P	N/DP] →	Progra	am bloc	ks ▶	MOTOR_AUTO [FB1]	_ # = ×
ιð	ыð MC	i I	🖗 👻 🍓 🖿 🚍 💬 🕾 ± 🎕 DR_AUTO	2 ± 🖂 1	\$ <b>6</b>	¢₀ ¢≣	98 <b>1</b> 2	I≣ <i>1</i> ≣	¢, °	° 12	<b>4</b>
		Na	me	Data t	Defaul	Retain	Acce	Visibl	Setp	Comment	
1	-	•	Input								
2		=	<add new=""></add>								
3	-	-	Output								
4	-		Conveyor_motor_automatic_mode	Bool 🔳	false	N 💌				Control of the conveyor motor in	automatic mode
5			<add new=""></add>								
6	-	•	InOut								
7			<add new=""></add>								
8	-	•	Static								
9			<add new=""></add>								
10	-	•	Temp								
11			<add new=""></add>								
12	-	•	Constant								
13			<add new=""></add>								

→ En primer lugar, agregue el parámetro #Automatic\_mode\_active

(Modo\_automático\_activo) como interfaz de entrada en Input (Entrada) 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)".

 $(\rightarrow$  Automatic\_mode\_active (Modo\_automático\_activo)  $\rightarrow$  Bool  $\rightarrow$  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 de seguridad activa), y compruebe los tipos de datos de dichos parámetros. Añada comentarios descriptivos.

03	2-2	00	_FB-Programming    CPU_1516F	[CPU 15	16F-3 P	N/DP] ▶	Progra	am bloo	:ks ▶	MOTOR_AUTO [FB1] 📃 🖬 🖬	i X
1Ô	ы	٤ :	/ / <b>.</b> E E E <b>.</b> 93±4	2 ± 😑 🗄	\$ CO	<b>€</b> ₀ (⊞	<b>€ ₽</b>	I <sub>≣</sub> % <sub>≡</sub>	e .	o 🔢	4
	MC	T	DR_AUTO								
		Na	me	Data t	Defaul	Retain	Acce	Visibl	Setp	Comment	
1	-	•	Input								^
2	-		Automatic_mode_active	Bool	false	Non-r				Automatic mode activated	
3	-		Start	Bool	false	Non-r				Pushbutton automatic start	
4	-		Stop	Bool	false	Non-r				Pushbutton automatic stop	
5	-		Enable_OK	Bool	false	Non-r				All enable conditions OK	
6	-		Safety_shutoff_active	Bool 🔳	false	N				Safety shutoff active e.g. emergency stop operated	
7			<add new=""></add>								
8	-	-	Output								2
9	-		Conveyor_motor_automatic_mode	Bool	false	Non-r				Control of the conveyor motor in automatic mode	
10			<add new=""></add>								
11	-	-	InOut								
12			<add new=""></add>								
13	-	•	Static								
14			<add new=""></add>								
15	-0	•	Temp								
16	<		<add new=""></add>								>

Libre utilización para centros de formación e I+D. © Siemens AG 2017. Todos los derechos reservados. SCE\_ES\_032-200 FB-Programming\_S7-1500\_R1703.docx 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 #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)". (→Memory automatic start stop

(Memoria\_modo\_automático\_arranque/parada)  $\rightarrow$  Bool  $\rightarrow$  Memory used for start/stop automatic mode (Memoria para la función de arranque y parada en modo automático)) 032-200\_FB-Programming  $\rightarrow$  CPU\_1516F [CPU\_1516F-3 PN/DP]  $\rightarrow$  Program blocks  $\rightarrow$  MOTOR\_AUTO [FB1]

1ÔI	19	1 =	≠ ≠ • E E E © 2 ± \$	it 🖂	\$ <b>6</b> 0	60 ell	(iii 4)	1= 1=	6.0	6 II	4
	MC	TC	R_AUTO								
-		Na	me	Data t	Defaul	Retain	Acce	Visibl	Setp	Comment	
1	-	•	Input								^
2	-00		Automatic_mode_active	Bool	false	Non-r				Automatic mode activated	
3	-		Start	Bool	false	Non-r				Pushbutton automatic start	
4	-		Stop	Bool	false	Non-r				Pushbutton automatic stop	-
5	-0		Enable_OK	Bool	false	Non-r				All enable conditions OK	
6	-		Safety_shutoff_active	Bool	false	Non-r				Safety shutoff active e.g. emergency stop operated	
7		=	<add new=""></add>								
8	-	•	Output								
9	-		Conveyor_motor_automatic_mode	Bool	false	Non-r				Control of the conveyor motor in automatic mode	
10			<add new=""></add>								
11	-	•	InOut								
12			<add new=""></add>								
13	-	•	Static								
14	-		Memory_automatic_start_stop	Bool 🔳	false	N 💌				Memory used for start/ stop automatic mode	

→ 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 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))

	MO	TC	DR_AUTO									
		Na	me	Data t	Defaul	Retain	Acce	Visibl	Setp	Comment		
1		٠	Input									^
2	-		Automatic_mode_active	Bool	false	Non-r		and the second		Automatic mode activated		
3	-		Start	Bool	false	Non-r				Pushbutton automatic start		
4	-		Stop	Bool	false	Non-r				Pushbutton automatic stop		-
5	-		Enable_OK	Bool	false	Non-r		and the second		All enable conditions OK		
6	-		Safety_shutoff_active	Bool	false	Non-r				Safety shutoff active e.g. emer	genc.	
7	-	•	Output									
8	-		Conveyor_motor_automatic_mode	Bool	false	Non-r				Control of the conveyor motor	in a	
9		•	InOut									
10			<add new=""></add>									
11	-	•	Static									
12	-		Memory_automatic_start_stop	Bool	false	Non-r				Memory used for start/ stop au	tom	~
	<					111					>	
-	RIO	ck	title: Motor control in automatic mode		11.3							~
▼ C T a T a I I F T	onv he b re n he b ctive Men fem or re here	eyo ot f it l ater ory easi	rr motor in automatic mode: Wemory_automatic_start_stop is set with fufilled. Wemory_automatic_start_stop is reset with d or if the automatic mode is not activant ry_automatic_start_stop is set, the enable _conveyor_start_stop is set the output C ons of energy efficiency the conveyor mode e Memory_conveyor_start_stop is set if the endemory_conveyor_start_stop is set if the endemory_conveyor_start_stop is set if the endemory_conveyor_start_stop is set if the endemory_conveyor_start_stop is set if the endemory_conveyor_start_stop is set if the endemory_conveyor_start_stop is set if the endemory_conveyor_start_stop is set if the endemory_conveyor_start_stop is set if the	h the inpu vith the in ted (man ole conditi conveyor_ otor shoul chere is a	ut Start, bi put Stop ual mode ions are g motor_au ld only rur part dete	ut only if t or if the sa ). ranted ar tomatic_t n if a part cted in fro	he reset afetyshu nd mode is is preser ent of Ser	conditio toff is activate nt. nsor_slid	ns d.			
•	nd r rift	he l	et with a negative edge at Sensor_end_ automatic mode is not activated (manu work 1: Memoryautomatic_start_sto	ot_convey ual mode) p and con	or or if the ). htrol of the	e satety sl e conveyo	r motor i	activate in autom	a atic mo 100%	de		~

#### 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.



 $\rightarrow$  En el lado derecho de la ventana de programación hay una lista de instrucciones que pueden utilizarse en el programa. Busque en  $\rightarrow$  Basic instructions (Instrucciones

básicas) → Bit logic operations (Operaciones lógicas con bits) la función E --[=] (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 +).

 $(\rightarrow$  Instructions (Instrucciones)  $\rightarrow$  Basic instructions (Instrucciones básicas)  $\rightarrow$  Bit logic operations (Operaciones lógicas con bits)  $\rightarrow$  E - [=])

	_20(	0_FB-Programming  ▶ CPU1516F [(	CPU 1516	F-3 PN/DP]	Program	blocks	• MOTO	DR_AUTC	) [FB1] 🛛 🗖 🗖	x	Instr	uction	S		∎ □	Þ
											Opti	ons				
LŐ.		:	2 ± 🖃 📾	eo (~ #	Q = 12 I.	1_ 0	90 HJ			4	Ċ.			iti j	•	
10.0	MC				•== •		-			•	V F	avorite	26			-
		Name	Data type	Default value	Retain	Accessi	Visible	Setpoint	Comment			avona				-
1	-	▼ Input								^	å	> = 1	??	-	-01	
2		<ul> <li>Automatic_mode_active</li> </ul>	Bool	false	Non-retain				Automatic mode a							
3	-	<ul> <li>Start</li> </ul>	Bool	false	Non-retain				Pushbutton autom	≣	4	-1=1	SR	RS		
4	-00	<ul> <li>Stop</li> </ul>	Bool	false	Non-retain		<b></b>		Pushbutton autom							
5	-	Enable_OK	Bool	false	Non-retain				All enable conditio							
6	-00	<ul> <li>Safety_shutoff_active</li> </ul>	Bool	false	Non-retain				Safety shutoff activ.	.	✓ B	asic in	struct	ions		
7		<ul> <li>Output</li> </ul>									٠ 🗖	Genera	al			~
8		<ul> <li>Conveyor_motor_automatic_mode</li> </ul>	Bool	false	Non-retain	<b></b>			Control of the conv.		-	Bit logi	c opera			
9		<ul> <li>InOut</li> </ul>										🗉 & 🗉		AN	D logic	
10	<	· Add areas							>			E >='	1	OR	logic o	≡
	_				-							🗉 🗴		EXC	LUSIV	
a	3	>=1 [??] -I -oI -→ -[=] SR	RS									🗉[=	]	Ass	ignme	
-		Johnnyk 1. Nomen automatic start stor	and contro	l of the conve	or motor in a	utomotic	modo					🗉 [/:	=]	Neg	gate as	
•		Network 1. Memory automatic_start_stop	o anu contre	of the conver	for motor in a	utomatici	noue					🗉 [R	]	Res	et outp	
	C	omment										🗉 [S	1	Set	output	
	-											SET	_BF	Set	bit field	
										≡		C RES	ET_BF	Res	et bit fi	
												🗉 SR		Set	/reset fl	
												🗉 RS		Res	et/set fl.	
												🗉 IP		Sca	n oper	

→ Ahora, desplace el parámetro de salida #Conveyor\_motor\_automatic\_mode (Motor\_cinta\_modo\_automático), 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

(→ <sup>4</sup> 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 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 <sup>a</sup> de la barra de herramientas lógicas para insertar una combinación mediante operador Y antes 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 <??.?>. (→ 
 Enable\_OK (Habilitación\_OK))

g	ran	nm	ing	CF	PU_	1516	5F [(	CPU	1516F-3	PN/DI	P]	Progr	am bloc	ks ▶ I	NOTOR	_AUTC	[FB1]	_ @ =	×
юÎ	10	1	i i	*	8				🗩 🕄 🗄	2 ± 🗄		₽ <b>¢</b> ⊖	¢⊕ ¢≣	<b>GE 4</b>	I <sup>≡</sup> <i>x</i> <sup>±</sup>	e 6 9	×* 🔢		-
	MC		JR_A		2							D. (. )	le este		hear				
	-	Na	ime							Data	t	Defaul	Retain	Acce	VISIDI	Setp	Comment		-
1			inpu				d		-	Real		false	Nenz		8		Automatic	made	^
4 0		5	-	tart	isauc		ue_a	cuv	e	Bool		falco	Non-r			H	Ruchhutto	noue	. =
3	-	-	-	ton						Bool		falce	Non-r				Pushbutto	n auto	
5	-		F	nabl	e O	ĸ				Bool	1	false	N				All enable	conditi	
6			5	afet	v shi	utoff	activ	e		Bool		false	Non-r.				Safety shu	toff acti.	-
7	-	•	Out	out										Ä	E E	ň			
8	-		C	onve	evor	mote	or au	iton	natic mode	Bool		false	Non-r				Control of	the con.	
9	-	-	InOu	ıt		1000000	-								- O	n			~
	<											IIII							>
• (	onv	>= eye	or mot	f tor in	auto	-ol omat	ic mo	ode	- <b>1</b> =] I										^
-		Net	twork nment	(1:	Mer	nory	auto	mat	ic_start_sto	p and	con	trol of the	e conveyo	or motor	in autom	natic mo	de		ſ
				<11 <11 <11	2.3>-	- **	8	•		#Conve moto autom moo	eyor or_ atic de	-							
			200	1+1															~
														100%			·		

→ 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) ESR a la primera entrada de la combinación Y de mediante

(Setear/resetear biestable) 🔤 🏧 a la primera entrada de la combinación Y 🖜 mediante "arrastrar y soltar".

(→ Instructions (Instrucciones) → Basic instructions (Instrucciones básicas) → Bit logic operations (Operaciones lógicas con bits) →  $\boxed{ESR} \rightarrow \boxed{-}$ )

		Tiogi			NO TON				Intions	
		l bro		6- JR	1 M	0.5			puons	
iκă ≇ ≝ 💁 📑 🗐 署 ¥ 🖉		+ C~	60 CE	9 V	i≡ ,≡	6			auf a	
MOTOR_AUTO								>	Favorites	
Name	Data t	Defaul	Retain	Acce	Visibl	Setp	Comment	~	Basic instructions	
Conveyor_motor_automatic_mode	Bool	false	Non-r	. 🗹			Control of the con	A Na	ame	Descrip
InOut									General	
Add new>								-	Bit logic operations	
Static									E &	AND Io
Memory_automatic_start_stop	Bool 🔳	false	N	•			Memory used for s	~	E >=1	OR logi
<		1011	_				>		E ×	EXCLUS
	. P		4						E -[=]	Assign
· >=1 [??] → -0 → -[=]									E -[/=]	Negate
Block title: Motor control in automatic mode									€ -[R]	Reset o
Conveyor motor in automatic mode:									🖭 -[S]	Set out
									E SET_BF	Set bit f
Network 1: Memory automatic_start_stop	and con	trol of the	e convey	or motor	in autom	atic mo	de		E RESET_BF	Reset bi
Comment									E SR	Set/rese
SATTITATIN								-	🖭 RS	Reset/se
									E - P -	Scan o
=	Conveyor	-							E - N -	Scan o
	motor_								🖭 -[P]-	Set ope
8	mode	-							E −[N]−	Set ope
<0.22 - R	-	-							P_TRIG	Scan RL
	-								TET N TRIC	Scan DI
#Enable OK-CAT		-								SCOTT NL

→ El biestable SR necesita una variable de memoria. Desplace el parámetro Static
 #Memory\_automatic\_start\_stop (Memoria\_modo\_automático\_arranque/parada) mediante
 "arrastrar y soltar" a <??.?> por encima del biestable SR. (→

Memory\_automatic\_start\_stop (Memoria\_modo\_automático\_arranque/parada))

		Nan	ne					Data t	Defaul	Retain	Acce	Visibl	Setp	Comment
3	-		Cor	iveyor	_moto	r_auto	matic_mode	Bool	false	Non-r				Control of the con.
9.	-	•	InOut											
0			<ad< td=""><td>ld nev</td><td>V&gt;</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></ad<>	ld nev	V>									
11	-	-	Static											
12	1		Mei	mory_	autom	atic_st	tart_stop	Bool	false	N 💌				Memory used for s
	۲								IIII					
0	onv	Netv	work 1	in au : Me	tomati emory a	t mod	e: atic_start_sto	p and co	ntrol of the	e conveyo	r motor i	n autom	atic mo	de

Libre utilización para centros de formación e I+D. © Siemens AG 2017. Todos los derechos reservados. SCE\_ES\_032-200 FB-Programming\_S7-1500\_R1703.docx

→ 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 la palabra "Start" para ver una lista de las variables disponibles que empiezan por esa palabra. Haga clic en la variable #Start (Arranque) y pulse  $\rightarrow$  Intro para aceptar.

 $(\rightarrow \text{SR-Flipflop (Biestable SR}) \rightarrow \langle ?? ? \rangle \rightarrow \text{Start (Arranque)} \rightarrow \# \text{Start (Arranque)} \rightarrow \text{Intro)}$ 

.gra	amn	ning	> CPU	1516	5F [CP	U 1516F-3	PN/DP]	▶ Progr ফু ৫০	am bloci	ks ► N 1881 😵	NOTOR <u></u> I <sub>≣</sub> <sup>1</sup> ≣		(FB1)		
M	NOT	OR_A	UTO												
1	N	ame					Data t	Defaul	Retain	Acce	Visibl	Setp	Commen	t	
*	•	(	Convey	or_moto	or_auto	omatic_mode	Bool	false	Non-r				Control o	f the con	. ^
4	•	InO	ut												
0			Add ne	-W>											=
1 -	<b>0</b> •	Stat	ic												
2 -	• 10	1	vlemory	_autom	natic_s	tart_stop	Bool 📃	] false	N 💌				Memory	used for s.	
	<							IHI						3	
C	Ne	tworl	c1: N	lemory	autom	atic_start_sto	p and cor	ntrol of the	e conveyo	r motor	in autom	natic mo	de		
	Con	nmen	t												
				# a s	Memo utoma tart_st	tic_ top			#Con						
5	tart			5	SR		&		automat	tic_					
	#3	Start				Bool			Pus	hbutton	aut	^			

**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 <sup>>=1</sup> de la barra de herramientas lógicas para insertar una combinación mediante operador O.



→ Inicialmente, el bloque O tiene únicamente 2 entradas. Para poder combinar una variable de entrada adicional, haga clic en el asterisco amarillo <sup>32</sup> del elemento O.

	>=1	
?.?		
?.?		-

→ Inserte en las tres entradas del elemento O las variables de entrada #Stop (Parada),
 #Safety\_shutoff\_active (Desconexión de 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 en Save project. A continuación se muestra el bloque de función ya creado "MOTOR\_AUTO" [FB1] de FUP.

ŝ	ю		- <u>B</u> a	•	EE	] 🚍	97	8± 4	) ± 🖃	¢°	<b>60 (</b> 11)	18 V	I <sup>≞</sup> <i>x</i> <sup>≞</sup>	e •	¢ 🔢
M	NO	тоі	R_AUT	0											
	1	Nam	ne						Data t	Defaul	Retain	Acce	Visibl	Setp	Comment
-	•	• 1	Input												
-			Auto	matic	_mod	le_acti	ve		Bool	false	Non-r				Automatic mode activated
-			Star	t					Bool	false	Non-r				Pushbutton automatic star
-			Stop	N.					Bool	false	Non-r				Pushbutton automatic stop
-			Enal	ble_OI	к				Bool	false	Non-r				All enable conditions OK
+			Safe	ty_shu	utoff_a	active			Bool	false	Non-r				Safety shutoff active e.g. e.
4		•	Output												
ì	1	-	-						~ '				-		e . 1 / d
	N	etv	work 1:	Mer	n ory a	utoma	atic_st	art_stop	o and con	trol of the	e conveyo	r motor i	in autom	atic moo	de
	#	Safi #	# ety_shu automi node_au	Stop - toff_ ctive - atic_ ctive	-	>=1			#	Start —	#Memo automat start_st SR S R1	y_ ic_ op		&	#Conveyor_ motor_ automatic_ mode

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

MOTOR_AUTO [	F81]	<b>Properties</b>	🗓 Info 😟 🛯 Diagnostics	■■▼
General				
General	Carrowski			
Information	General			
Time stamps				
Compilation	Name:	MOTOR_AUTO		
Protection	Type:	FB		
Attributes	ijpe.			
Download	Language:	LAD		-
	Number:	EBD		
		() manual		_
		<ul> <li>automatic</li> </ul>		

→ 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 (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]".

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



 $\rightarrow$  Abra el bloque de organización "Main [OB1]" haciendo doble clic.



- → Asigne al Segmento 1 el nombre "Control de la marcha de la cinta hacia delante en modo automático".
  - $(\rightarrow$  Segmento 1:...  $\rightarrow$  Control de la marcha de la cinta hacia delante en modo automático)

03	32-2	00	_FB	-Pro	grar	nmin	g ▶ (	CPU_	1516F [CPU	1516F	-3 PN/DP] 🕨 I	Program blocks 🕨 Main [OB1]	_ # = ×
н	Śн	ĸ	ý.	÷	20	E		1	2±2±	:= 😥	🥐 📞 🖑 🤅		
	Ma	in											
		Na	me						Data type		Default value	Comment	
1		•	Inp	ut									^
2				Initia	al_Ca				Bool			Initial call of this OB	
3	-			Rem	aner	nce			Bool			=True, if remanent data are available	
4	-	•	Ten	np									
5				<ad< td=""><td>d nev</td><td>V&gt;</td><td></td><td></td><td></td><td></td><td></td><td></td><td>~</td></ad<>	d nev	V>							~
				_				1					
2		>=1	1	2	-	-01	4	-[=]					
-	Blo	ck	title		Main	Progra	m Sw	eep (C	Cycle)*				
- 0	Com	me	nt			-							
-	1	Net	two	k 1:	: Co	ntrol c	onvey	or mo	tor forwards in	n automa	stic mode		
	0	lom	mer	it									

→ Ahora desplace el bloque de función "MOTOR\_AUTO [FB1]" hasta la línea verde del Segmento 1, mediante "arrastrar y soltar".

Ma Siemens - C:\Users\spe\Documents\A	utom	atisierung\032-200_FB-Programmi	ng\032-200_FB-Progr	ramming						-	a x
Project Edit View Insert Online O	ptions	Tools Window Help St (#± 🗄 🗓 🗓 🖳 🙀 🎽	Go online 🖉 Go offi	ine 🏠 🖪 📭	* =		Totall	y Integ	grated Auton	nation PORTA	L
Project tree	•	032-200_FB-Programming + 0	CPU_1516F [CPU 15	16F-3 PN/DP] 🔸	Program blocks    Main [OB1]	_ # = ×	< Ins				
Devices							Opt	ions			- 18
000		N N I I I I I I I I I I I I I I I I I I	93:3:E	10 to 60 to	98 9 1 1 8 0° 10	<b>3</b>			,		Ins
2		Main					~	Favori	tes		Tu
O32-200_FB-Programming     Add new device     Action Devices & networks	^	Name 1 ≪3 ▼ Input 2 ≪3 = Initial_Call	Data type Bool	Default value	Comment Initial call of this OB	4	۵ ا	>=1	127 -	-01	tions
CPU_1516F [CPU 1516F-3 P     Device configuration     Online & diagnostics		3	Bool		=True, if remanent data are availe	ible	~				Testin
Add new block  Main [081]  Main [081]  MotoR_AUTO [F81]  Main [081]  Main [08		S >=1 1 1 01 →     Block title: "Main Program Swe Comment     Notwork 1: Control conveye	<b>H=]</b> eep (Cycle)*	tomatic mode							(III) Tasks
PLC tags     PLC tags     PLC tags     PLC tags     Watch and force tables     PLC Online backups		Comment	OTOR_AUTO [F81]			_					Libraries
Program info     Program info     Program unter u	>		یک +				Nam	Basic i e ] Gene	nstructions ral		
Name Address							< >	Extend	gic operations operations iii led instructio	) ) ) ) ) )	
					1.000		>	Techno	ology		
4					0 Properties	Diagnostics		Comm	unication		-
		1		_	Shoperues Sinto 3	o pragnosaus		opuor	ar packages		

→ 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))

Call options		×
Single instance	Data block Name Number Number Autoo DB Automatic Automatic The called function block saves its data in its own instance data block. More	
	OK Cancel	

→ Se insertará en el Network 1 (Segmento 1) un bloque con la interfaz que se ha definido, el bloque de datos de instancia 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 <sup>a</sup> de la barra de herramientas lógicas. (→ <sup>a</sup>)

	> = 1	??	-	-01	↦	-[=]	
Þ F	ND I	ogic op	erati	on (Shi	ft+F2]		
						9/DB	
						"MOTOR_/	AUTO_
						DB"	and the second sec
						%F8	1
						"MOTOR_	AUTO"
			***	- EN			
				Au	tomati	c_	
			false	- ma	de_ad	tive	
			false	- Sta	irt		
			false	- Sto	р		Conveyor_
			false	En En	able_C	ж	automatic_
				Sa	fety		mode
			false	- sh	utoff a	ctive	ENO

- → 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\_modo\_automático)", la variable de salida "-Q1" (%Q0.0).



 $\rightarrow$  Niegue las consultas de las variables de entrada "-S2" y "-A1" seleccionándolas y



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

Aain		C~ 40 (m 42	A	
Name	Data type	Default value	Comment	
🔟 🔻 Input				
Initial_Call	Bool	■)	Initial call of this OB	
H/H → → → →	notor forwards in autom	natic mode		
	*MO	%DB1 TOR_AUTO_ DB* %FB1		
	1MO EN	TOR_AUTO"	o4	
	%40.2 Automatic_ *-S0* — mode_active %40.3 *-S1* — Start	automatic motor automatic mod	e → *-Q1*	
*40.4 *-52*	Stan			
	stop			
%40.1 %40.5 *-K0* *-B1*				
— I	Enable_OK			

#### 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) → Compilación.



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

	Q Properties	🗓 Info 🤢 🗓 Diagnosti	cs 🛛 🗆 🗸
General (1) Cross-references	Compile Syntax		
🕄 🚹 🕤 Show all messages			
Compiling completed (errors: 0; warning	gs: 0)		
! Path	Description	Go to	? Errors
✓ ▼ CPU_1516F		~	0
Program blocks		7	0
MOTOR_AUTO (FB1)	Block was successfully compiled.	7	
MOTOR_AUTO_DB (DB1)	Block was successfully compiled.	~	
Main (OB1)	Block was successfully compiled.	7	
0	Compiling completed (errors: 0; warnin	ngs: 0)	
111111			

#### 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. (→ III)



#### 7.12 Visualización de los bloques de programa

→ Para observar el programa cargado, debe estar abierto el bloque deseado. Haciendo clic

en el icono  $\square$  se activa/desactiva la visualización. ( $\rightarrow$  Main [OB1]  $\rightarrow$   $\square$ ) ..0\_FB-Programming > CPU\_1516F [CPU 1516F-3 PN/DP] > Program blocks > Main [OB1] TX 🔥 🖓 👻 👘 🖿 🚍 📼 웹 ± 월 ± 월 🕼 🧐 🦛 🖓 📲 🦆 🖉 🔢 E Monitoring on/off >=1 ??? -[=] 8 - - - 01 -~ Network 1: Control conveyor motor forwards in automatic mode Comment %DB1 "MOTOR\_AUTO\_ DB" FB "MOTOR\_AUTO" EN %10.2 Automatic "-SO" mode\_active 8 %0 3 "-S1" Start %10.1 "-K0" %10.4 Conveyor "-S2" -%10.5 Stop motor\_ "-B1" -Enable\_OK %Q0.0 automatic mode . 01 40 0 Safety "-A1 shutoff\_active ENC 👸 🕉 👻 🐁 📰 🚍 💬 웹 ± 월 ± 달 😥 🧐 😘 웬 행 🌞 🖕 🍾 🔗 🕎 🔢 >=1 [??] - -ol - -[=] ۵ Network 1: Control conveyor motor forwards in automatic mode Comment %D81 "MOTOR\_AUTO\_ DB" FB "MOTOR\_AUTO" EN TRUE 40.2 Automatic "-SO" mode\_active FALSE TRUE "-S1" Start -ко TRUE %10.4 TRUE "-S2" Stop Conveyor "-B1 Enable\_OK TRUE TRUE automatic Q0.0 mode -01 %10.0 -A1 Safety\_ shutoff\_active ENO

**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 el bloque de función "Main [OB1]" llamado en el bloque de organización "MOTOR\_AUTO" [FB1], selecciónela directamente tras hacer clic en ella con el botón derecho del ratón. (→ "MOTOR\_AUTO" [FB1] → 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 visualizar una ubicación concreta de un bloque de función "MOTOR\_AUTO"
 [FB1] que se llama varias veces, puede usarse el icono <sup>™</sup>. Existen dos posibilidades: usar el entorno de llamada o definir el entorno de llamada mediante el bloque de datos de instancia. (→ <sup>™</sup>. 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))

MOT	OR_AUTO_DB [DB1]			
Call er	nvironment			
	Dependency structure	1	Address	Details
1	Hain ("MOTOR_AUTO_DB")		OB1	Main NW1 (Co.
4				
		Trans	fer to "adjust	ed manually*
Manua	ally adjusted call environment			

) Insta	nce data block TOR_AUTO_DB [DB1]			
Call e	environment			
	Dependency structure	1	Address	Details
1	- Main ("MOTOR_AUTO_DB")		OB1	Main NW1 (Co
-				
		Trans	sfer to "adjust	ed manually"
Man	ually adjusted call environment			

#### 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) → 032-200 FB Programming.... → Save (Guardar))

Main Siemens - D:\00_DATA\SIEMENS\Unterlagen\0	08_Ausbildungsu	unterlage_TIA-Portal	_R1503_e\SCE_EN	_032-100 FC-Programming\032_200_FB-Programming\032_200_FB-Programming _ D
Project Edit View Insert Online Options	Tools Window	Help		Totally Integrated Automation
I New ) ±	(~ ± 🖥 🛄	🚹 🖳 📮 💋 Go or	nline 🖉 Go offline	h? 🖪 🖪 🗶 📃 PORTAL
Migrate project	032_200_FB	-Programming 🕨	CPU1516F [CPU	1516F-3 PN/DP] → Program blocks → Main [OB1] _ II = X
Close Ctrl+W				
🔚 Save Ctrl+S 📑	• <mark>% _X</mark> ⇒ :	0 🐁 🖻 🚍 🚍	💬 🖀 ± 🖀 ± 🖡	
Save as Ctrl+Shift+S	TOT TOT -			Block interface
Delete project Ctrl+E				
Archive	& >=1 [?	?' • - →	-[=] SR RS	
Retrieve				
Card Reader/USB memory     Memory card file	<ul> <li>Network</li> </ul>	k 1: Control conveyor	r motor forwards in	automatic mode
Linera de	Commen	t		
				ů.
A Print preview				%DB1
D-100 DATAL 1032 200 EB-Programming				MOTOR_AUTO_DBT
D:100_TIA1032-100_FC-Programmierung				
D:\00_TIA\032-200_FB-Programmierung				
D:100_TIA_Po1032_100_FC-Programming				
D:tAbschlusspruerung_reiri_wechatr				%IO.2 Automatic_mode_
Exit				-su — acuve
Online backups			&	%IO.3
Traces		%0.1		-s) — Start
Program info		-KU —		%i0.4
Device proxy data		%10.5		
PLC alarms		-BI — *		Enable_OK
Local modules				%IO.0 Safety_shutoff_
Common data				-A1 -•• active
Documentation settings				%i1.0
Languages & resources				-b4 — Sensor_slide Conveyor_motor_ %Q0.0
Gard Peader/LISP memory				%11.3 Sensor end of automatic_mode — -Q1"
Details adam	×.			
Details view				🙁 Properties 🛛 🛄 Into 🛄 🖄 Diagnostics 👘 👘 — 🦳
Portal view	r Main			Project 032 200 FB-Programming ope

## 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 AUTOMÁTICO (-S0 = 1) Pulsador de parada automática no accionado (-S2 = 1) Accionar brevemente el pulsador de arranque automático (-S1 = 1), luego se conecta el motor cinta hacia delante, velocidad fija (-Q1 = 1) y permanece encendido.	
4	Accionar brevemente el pulsador Parada automática $(-S2 = 0) \rightarrow -Q1 = 0$	
5	Activar PARADA DE EMERGENCIA (-A1 = 0) $\rightarrow$ -Q1 = 0	
6	Modo de operación manual (-S0 = 0) $\rightarrow$ -Q1 = 0	
7	Desconectar la instalación (-K0 = 0) $\rightarrow$ -Q1 = 0	
8	Cilindro no introducido (-B1 = 0) $\rightarrow$ -Q1 = 0	
9	Proyecto archivado correctamente	

## 9 Ejercicio

#### 9.1 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).

#### 9.2 Planificación

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

**Nota:** Encontrará información acerca del uso del flanco negativo en SIMATIC S7-1500 en la Ayuda en pantalla.

### 9.3 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), luego se conecta el motor cinta hacia delante, velocidad fija (-Q1 = 1) y permanece conectado.	
4	Sensor fin de cinta, activado (-B7 = 1) $\rightarrow$ -Q1 = 0	
5	Accionar brevemente el pulsador Parada automática (-S2 = 0) $\rightarrow$ -Q1 = 0	
6	Activar PARADA DE EMERGENCIA (-A1 = 0) $\rightarrow$ -Q1 = 0	
7	Modo de operación manual (-S0 = 0) $\rightarrow$ -Q1 = 0	
8	Desconectar la instalación (-K0 = 0) $\rightarrow$ -Q1 = 0	
9	Cilindro no introducido (-B1 = 0) $\rightarrow$ -Q1 = 0	
10	Proyecto archivado correctamente	

## 10Información 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