

CONTROLADORES LÓGICOS PROGRAMABLES

**siemens.com/sce**

Módulo TIA Portal 031-100

Principios básicos de la programación

de FC con SIMATIC S7-1200

Índice de contenido

[1 Objetivos 5](#_Toc56777400)

[2 Requisitos 5](#_Toc56777401)

[3 Hardware y software necesarios 6](#_Toc56777402)

[4 Teoría 7](#_Toc56777403)

[4.1 Sistema operativo y programa de usuario 7](#_Toc56777404)

[4.2 Bloques de organización 8](#_Toc56777405)

[4.3 Imagen de proceso y ejecución cíclica del programa 9](#_Toc56777406)

[4.4 Funciones 11](#_Toc56777407)

[4.5 Bloques de función y bloques de datos de instancia 12](#_Toc56777408)

[4.6 Bloques de datos globales 13](#_Toc56777409)

[4.7 Bloques lógicos aptos para librería 14](#_Toc56777410)

[4.8 Lenguajes de programación 15](#_Toc56777411)

[5 Tarea planteada 16](#_Toc56777412)

[6 Planificación 16](#_Toc56777413)

[6.1 PARADA DE EMERGENCIA 16](#_Toc56777414)

[6.2 Operación manual: motor de la cinta en modo Jog 16](#_Toc56777415)

[6.3 Esquema tecnológico 17](#_Toc56777416)

[6.4 Tabla de asignación 18](#_Toc56777417)

[7 Instrucciones paso a paso estructuradas 19](#_Toc56777418)

[7.1 Desarchivación de un proyecto existente 19](#_Toc56777419)

[7.2 Creación de una nueva tabla de variables 20](#_Toc56777420)

[7.3 Creación de nuevas variables dentro de una tabla de variables 22](#_Toc56777421)

[7.4 Importación de la "Tag table\_sorting station (Tabla\_variables\_planta\_clasificación)" 23](#_Toc56777422)

[7.5 Creación de la función FC1 "MOTOR\_MANUAL" para el motor de cinta en modo Jog 27](#_Toc56777423)

[7.6 Definición de la interfaz de la función FC1 "MOTOR\_MANUAL": 29](#_Toc56777424)

[7.7 Programación del FC1: MOTOR\_MANUAL 32](#_Toc56777425)

[7.8 Programación del bloque de organización OB1: control de la marcha de la cinta hacia delante en modo manual 39](#_Toc56777426)

[7.9 Guardado y compilación del programa 44](#_Toc56777427)

[7.10 Carga del programa 45](#_Toc56777428)

[7.11 Visualización de los bloques de programa 46](#_Toc56777429)

[7.12 Archivar proyecto 48](#_Toc56777430)

[7.13 Lista de comprobación 49](#_Toc56777431)

[8 Ejercicio 50](#_Toc56777432)

[8.1 Tarea planteada: ejercicio 50](#_Toc56777433)

[8.2 Esquema tecnológico 50](#_Toc56777434)

[8.3 Tabla de asignación 51](#_Toc56777435)

[8.4 Planificación 51](#_Toc56777436)

[8.5 Lista de comprobación: ejercicio 52](#_Toc56777437)

Principios básicos de la programación de FC

# 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 una función FC1 y un bloque de organización OB1.

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

# 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

# Hardware y software necesarios

**1** Estación de ingeniería: Se requieren el hardware y el sistema operativo

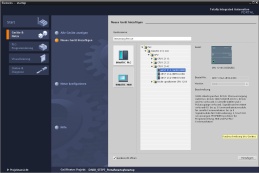
(Para más información, ver Readme/Léame en los DVD de instalación del TIA portal)

**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



**2** SIMATIC STEP 7 Basic (TIA Portal) V14 SP1 o superior



**1** Estación de ingeniería

**4** Conexión Ethernet



**3** Controlador SIMATIC S7-1200



Cuadro

# Teoría

* 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 rearranque (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.

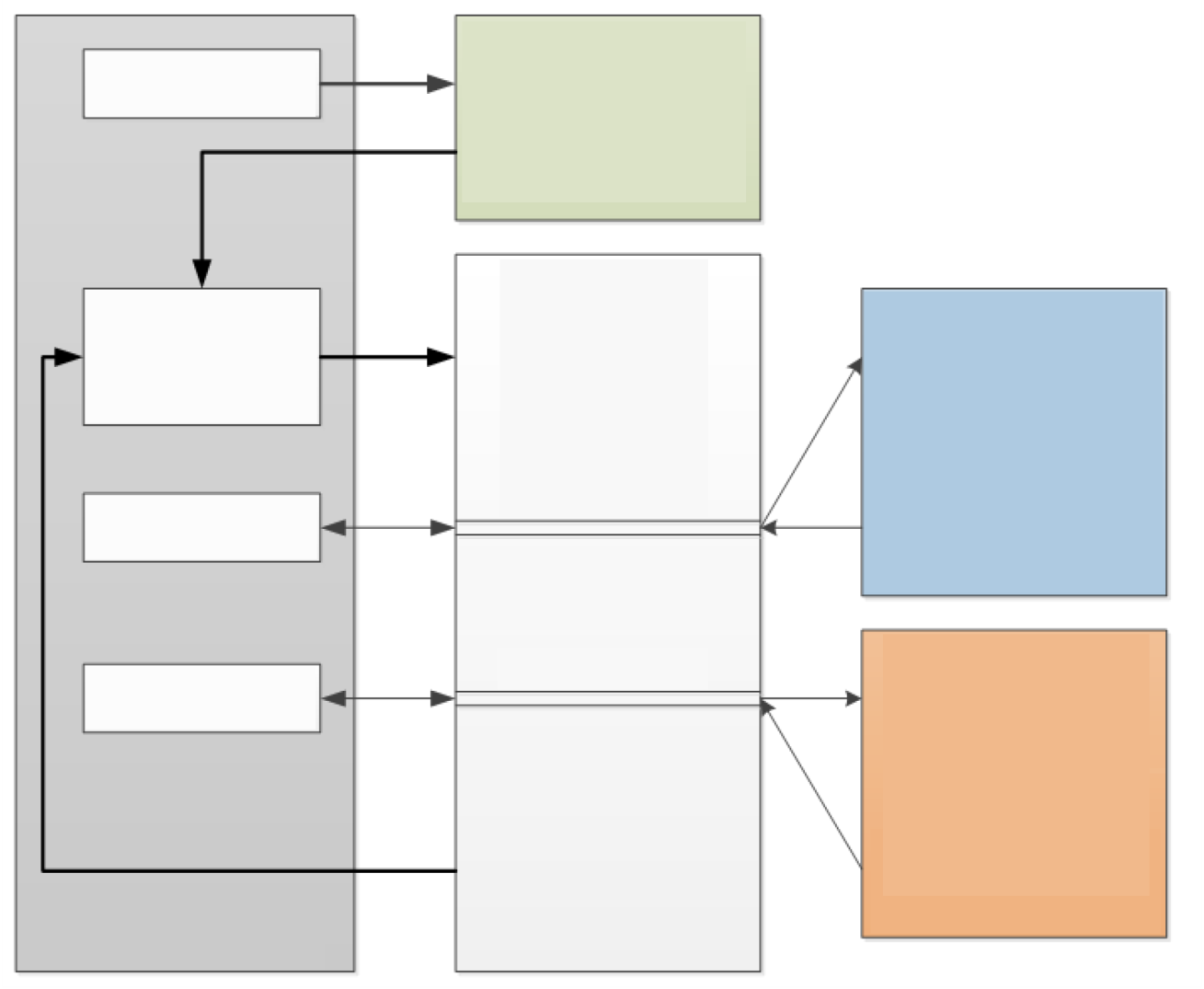
El programa de usuario contiene todas las funciones necesarias para ejecutar la tarea de automatización específica. Algunas de las tareas del programa de usuario son:

* Comprobación de los requisitos previos para un rearranque 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
  1. 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.



*Interrupción*

*Interrupción*

**Programa de arranque**

OB 100 Arranque en caliente  
…

**Ejecución**   
**cíclica del programa**

OB 1

***Ejecución del programa controlada por alarmas***

OB 40…

***Tratamiento de errores***

OB 80

OB 82

…

*Error*

*Alarma*

**Ciclo**

**ON (Run)**

**Sistema operativo**

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, ≥ 123 | Ignorar |
| Programa cíclico | 1, ≥ 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

* 1. 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.

**IPE**

Programa del PLC en la memoria de programa

1.ª instrucción

2.ª instrucción

3.ª instrucción

4.ª instrucción

...

Última instrucción

**2.** Procesamiento del programa instrucción por instrucción con acceso a IPE e IPS

**Datos locales**

**Marcas**

**Bloques de datos**

**IPS**

**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.

* 1. 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.

Bloque de organización

Main [OB1]

Llamada de una función MOTOR\_MANUAL [FC1]

Función MOTOR\_MANUAL [FC1]

Contiene, por ejemplo, un programa para el control de una cinta en modo manual.

La función carece de memoria.

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

* 1. 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.

Bloque de datos de instancia MOTOR\_AUTO\_DB1 [DB1] como memoria

para la llamada

del bloque de

función

MOTOR\_AUTO [FB1]

Bloque de organización

Main [OB1]

Llamada de un bloque de función MOTOR\_AUTO [FB1] junto con su bloque de datos de instancia MOTOR\_AUTO\_DB1 [DB1]

Bloque de función MOTOR\_AUTO [FB1]

Contiene, por ejemplo, un programa para el control de una cinta en modo automático.

En esta llamada, el bloque de función utiliza como memoria el bloque de datos de instancia MOTOR\_AUTO\_DB1 [DB1].

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

* 1. 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.



Acceso solo para el bloque de datos de función\_12

Acceso para todos los bloques

DB de instancia  
(DB\_instancia)

DB global  
(DB\_global)

Bloque de función\_12

Función\_11

Función\_10

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

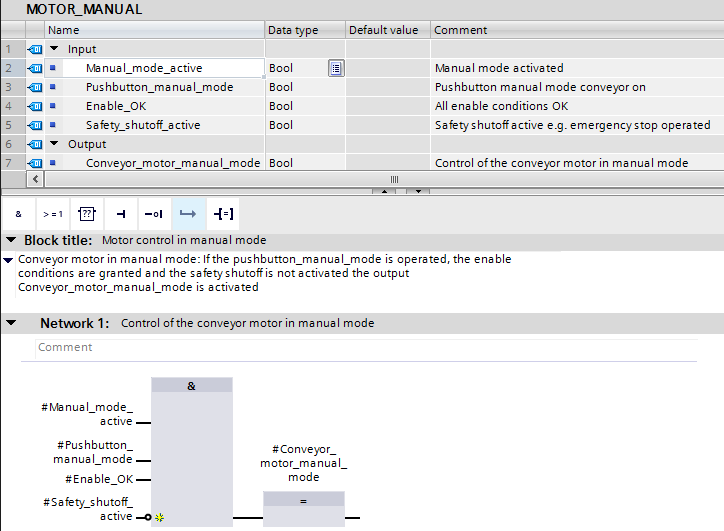
Ejemplos de uso para **bloques de datos globales**:

* Guardar la información en un sistema de almacén: "¿Qué producto está en cada lugar?"
* Guardar recetas de determinados productos.
  1. Bloques lógicos aptos para librería

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

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

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



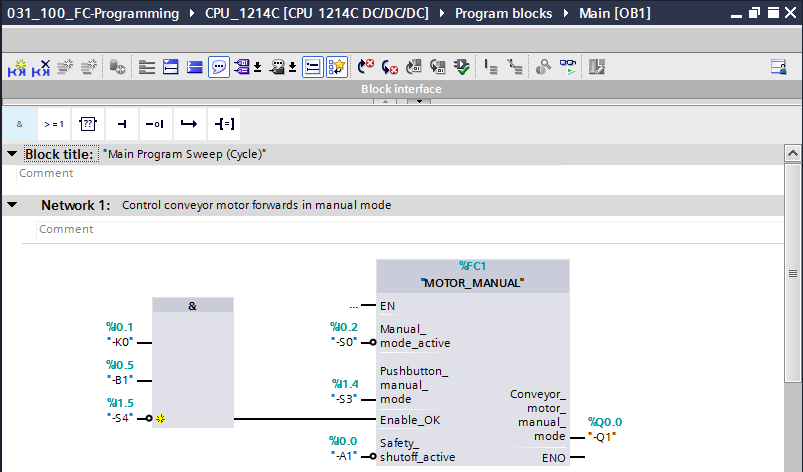


Figura 6: Función apta para librería con llamada en el OB1

* 1. Lenguajes de programación

Para programar funciones y bloques de función para 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).

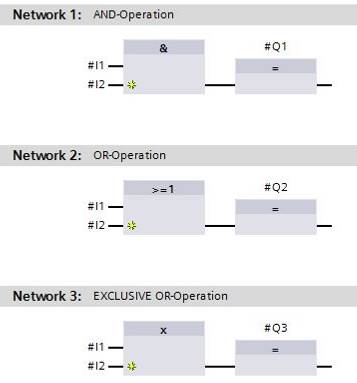
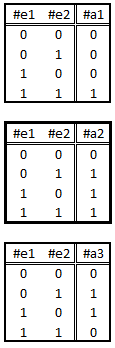
 

Figura 7: Funciones binarias en FUP y tabla lógica correspondiente

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.

# Tarea planteada

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

* Operación manual: control de la marcha de la cinta hacia delante en modo manual o Jog.

# Planificación

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

* 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.

* 1. Operación manual: motor de la cinta en modo Jog

El modo Jog del motor de la cinta se encapsulará en una función (FC) "MOTOR\_MANUAL". Con ello, por un lado, se 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.

|  |  |  |
| --- | --- | --- |
| **Input (Entrada)** | Tipo de datos | Comentario |
| Manual\_mode\_active (Modo\_manual\_activo) | BOOL | Modo de operación manual activado |
| Pushbutton\_manual\_mode (Pulsador\_modo\_Jog) | BOOL | Pulsador para conectar el motor de la cinta en modo Jog |
| 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\_manual\_mode (Motor\_cinta\_modo\_Jog) | BOOL | Control del motor de la cinta en modo Jog |

Tabla 2: Parámetros para la FC "MOTOR\_MANUAL"

La salida Conveyor\_motor\_manual\_mode (Motor\_cinta\_modo\_Jog) estará en estado ON mientras esté presionado el Pushbutton\_manual\_mode (Pulsador\_modo\_Jog), esté activado el modo manual, esté otorgada la habilitación y no esté activa la desconexión de seguridad.

* 1. Esquema tecnológico

Este es el esquema tecnológico para la tarea asignada.

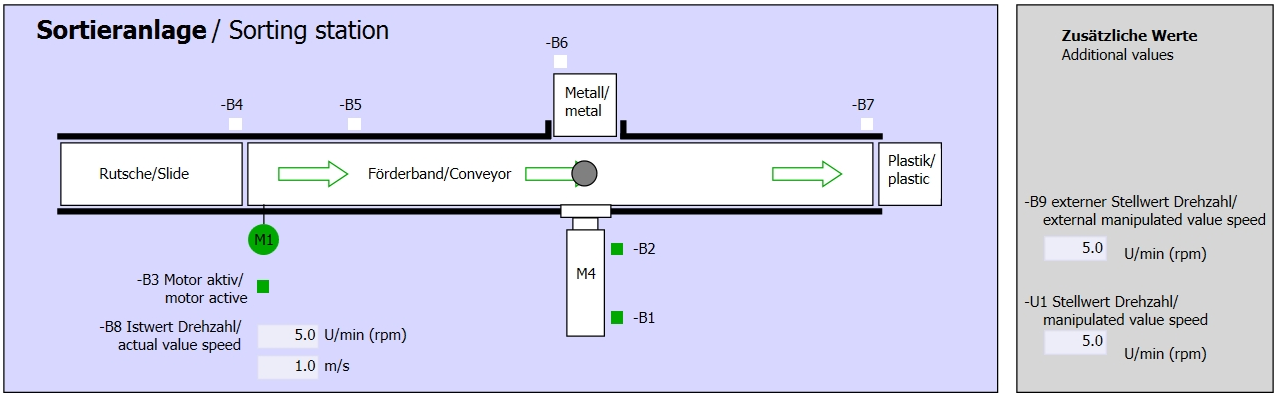


Figura 8: Esquema tecnológico

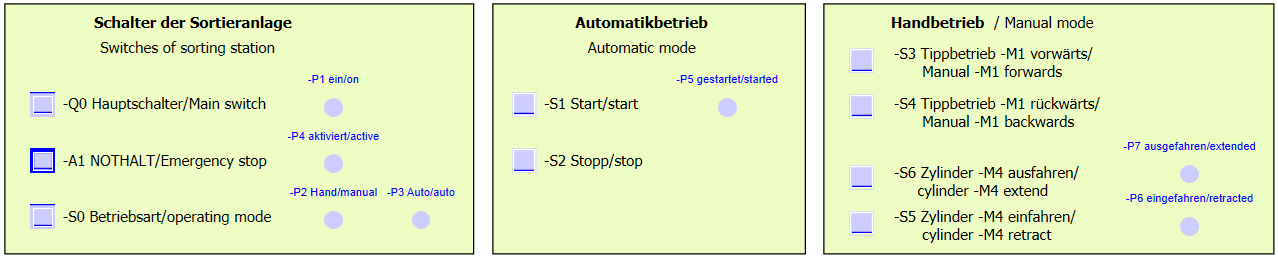


Figura 9: Pupitre de mando

* 1. 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 0.5 | BOOL | -B1 | Sensor cilindro -M4 introducido | NA |
| I 1.4 | BOOL | -S3 | Pulsador modo Jog cinta -M1 hacia delante | NA |
| I 1.5 | BOOL | -S4 | Pulsador modo Jog cinta -M1 hacia atrás | NA |

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

***Leyenda de la lista de asignación***

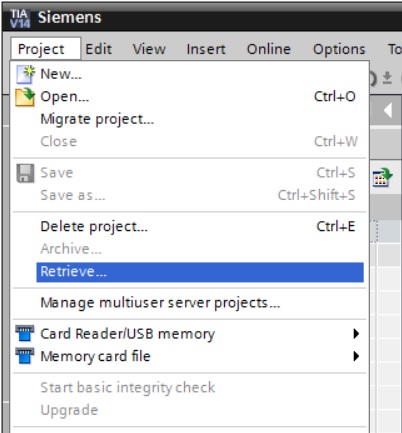
|  |  |
| --- | --- |
| DQ | Salida digital |
| AQ | Salida analógica |
| S | Salida |

|  |  |
| --- | --- |
| DI | Entrada digital |
| AI | Entrada analógica |
| E | Entrada |
| NC | Normalmente cerrado |
| NA | Normalmente abierto |

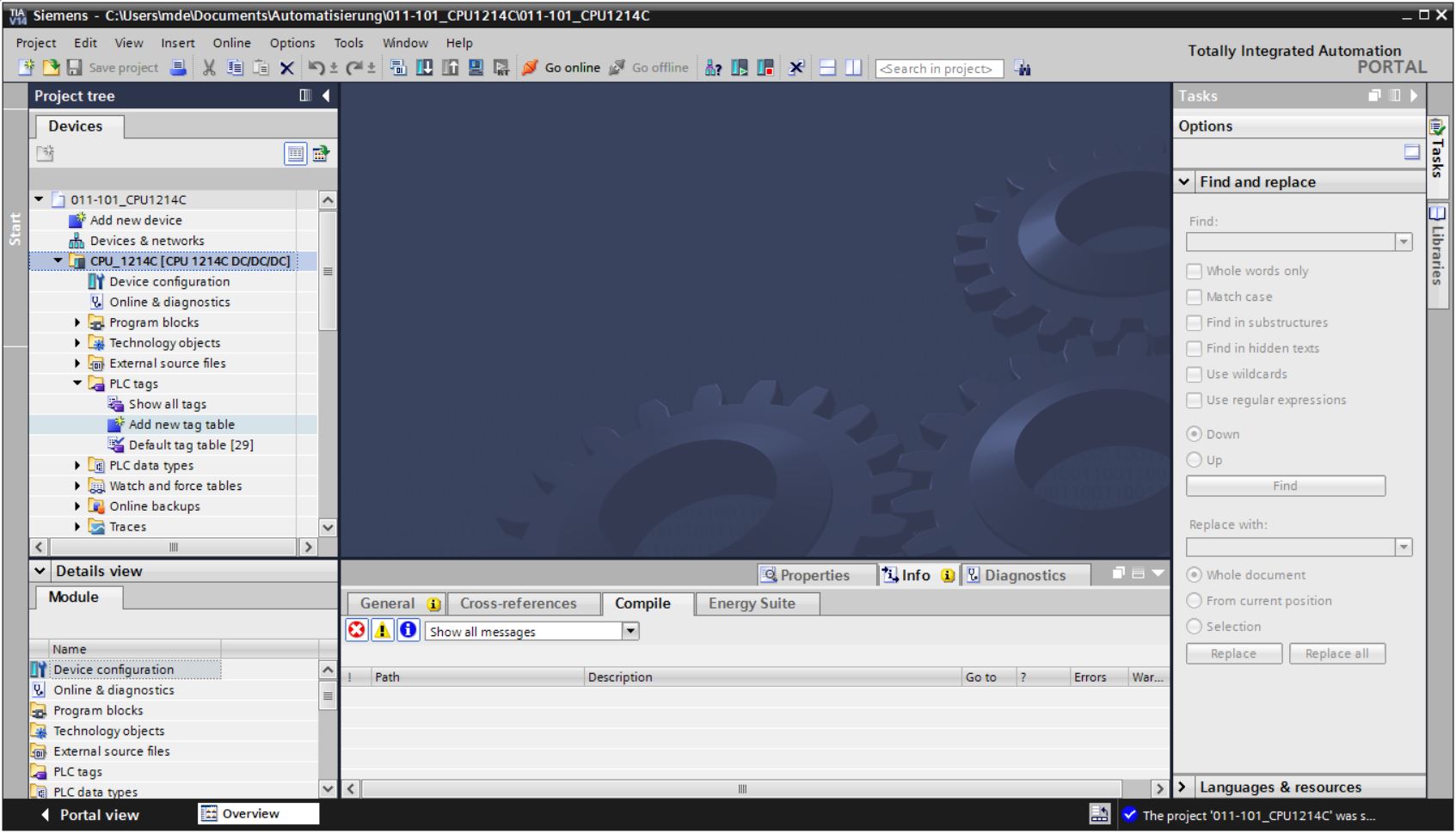
# Instrucciones paso a paso estructuradas

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.

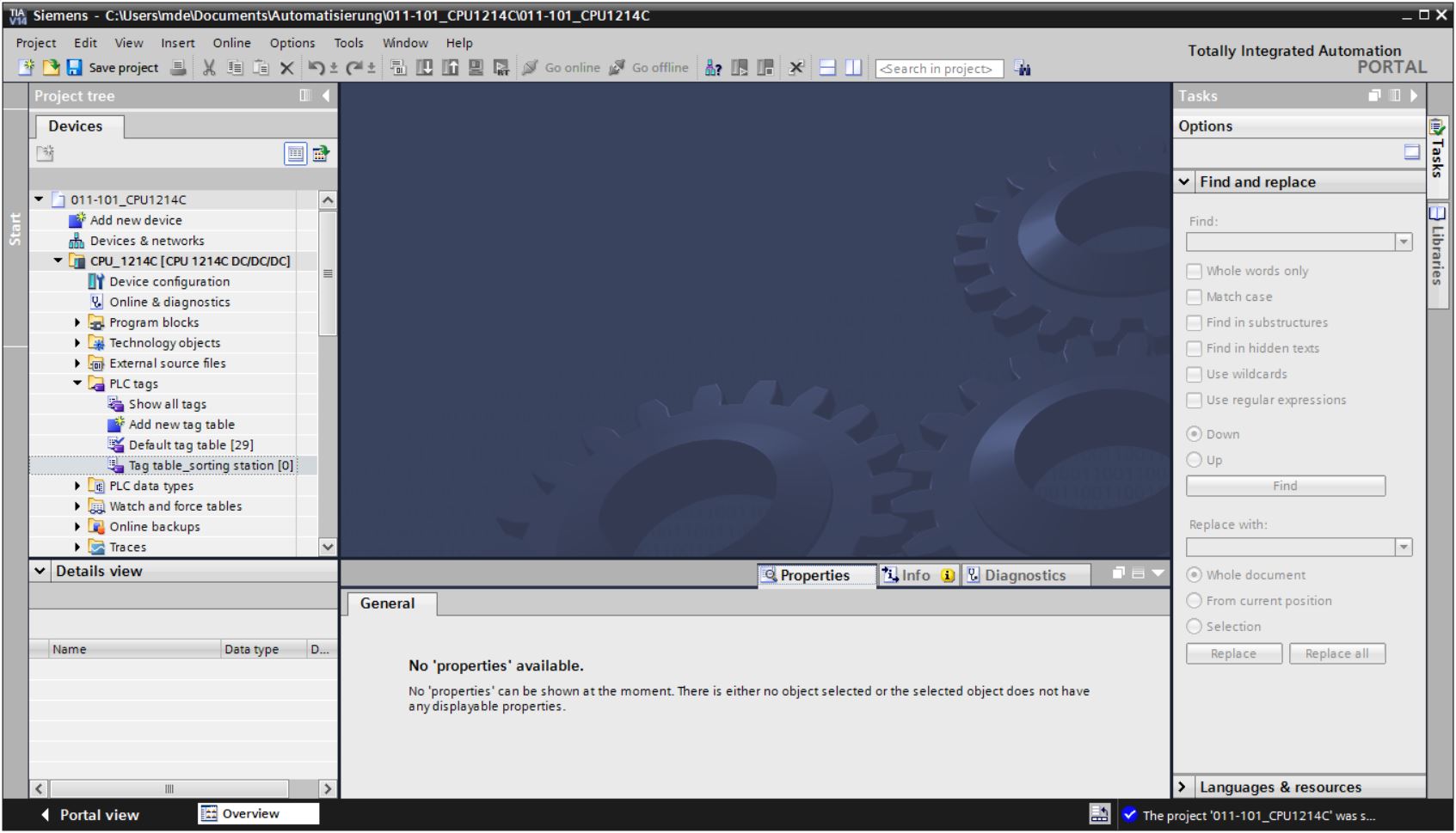
* 1. Desarchivación de un proyecto existente
* Antes de empezar a programar la función (FC) "MOTOR\_MANUAL", se necesita un proyecto con una configuración hardware (p. ej., SCE\_ES\_011\_101\_Hardware Configuration\_ CPU1214C.zap14). Para desarchivar un proyecto existente desde la vista del proyecto, 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)).
  1. 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).

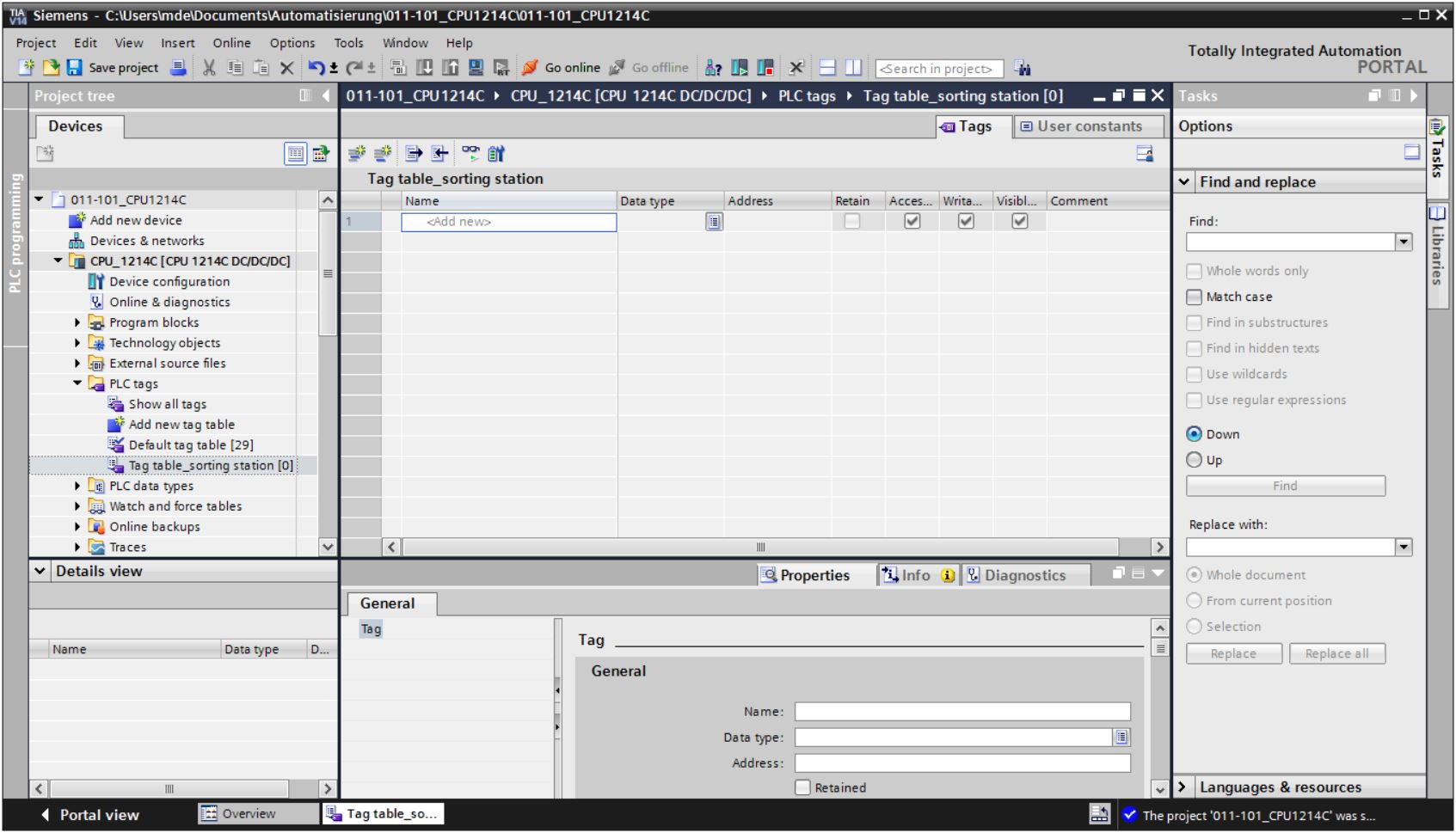


* Asigne el nombre "Tag table\_sorting station (Tabla\_variables\_planta\_clasificación)" a la tabla de variables que acaba de crear (→ clic con el botón derecho en "Tag table\_1 (Tabla\_variables\_1)" → "Rename (Cambiar nombre)" → Tag table\_sorting station (Tabla\_variables\_planta\_clasificación)).

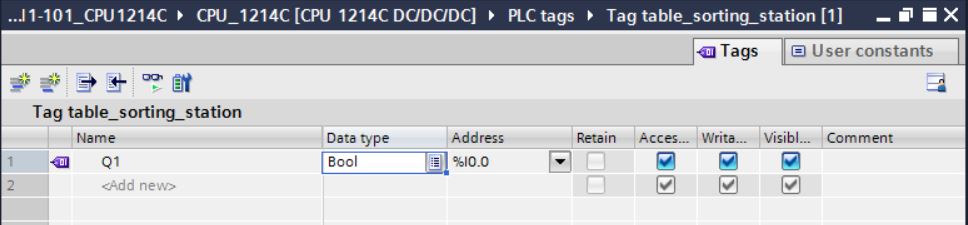


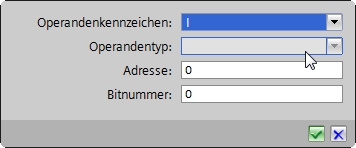
* A continuación ábrala haciendo doble clic (→ Tag table\_sorting station

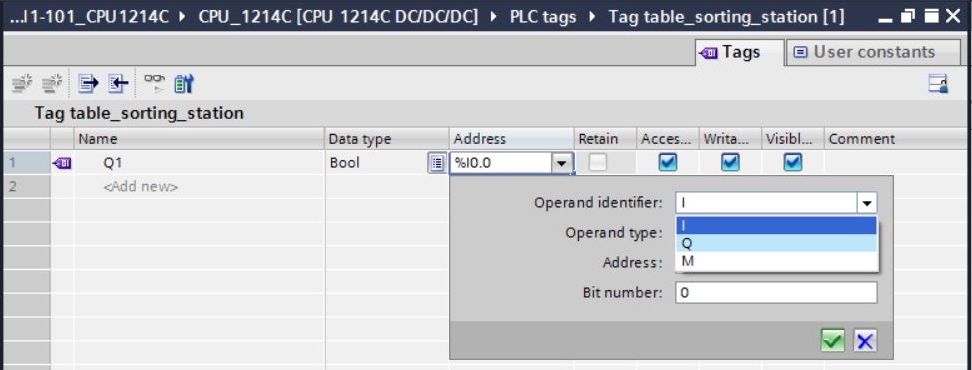
(Tabla\_variables\_planta\_clasificación)).



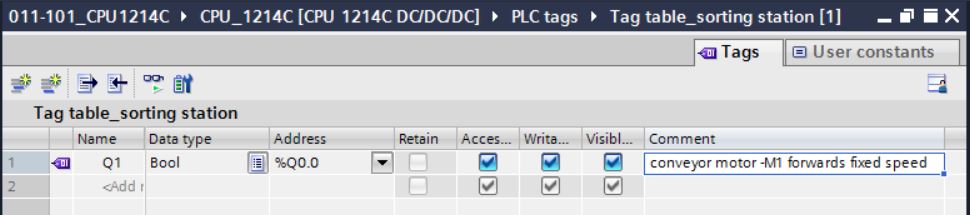
* 1. Creación de nuevas variables dentro de una tabla de variables
* Agregue el nombre Q1 y confirme con la tecla Intro. Si todavía no ha creado ninguna otra variable, el TIA Portal habrá asignado automáticamente el tipo de datos "Bool" y la dirección %I0.0 (→<Add new> (Agregar) → Q1 → Intro).



* 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 → ).

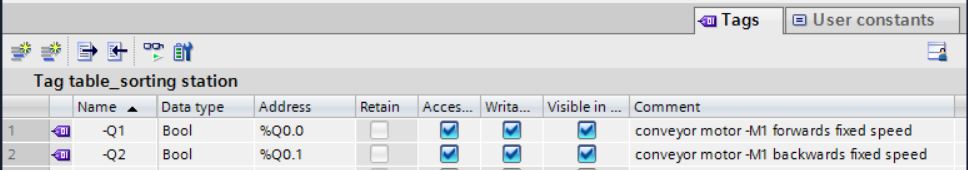


* Asigne a la variable el comentario "conveyor motor -M1 forwards fixed speed (Motor de cinta -M1 hacia delante, velocidad fija)".



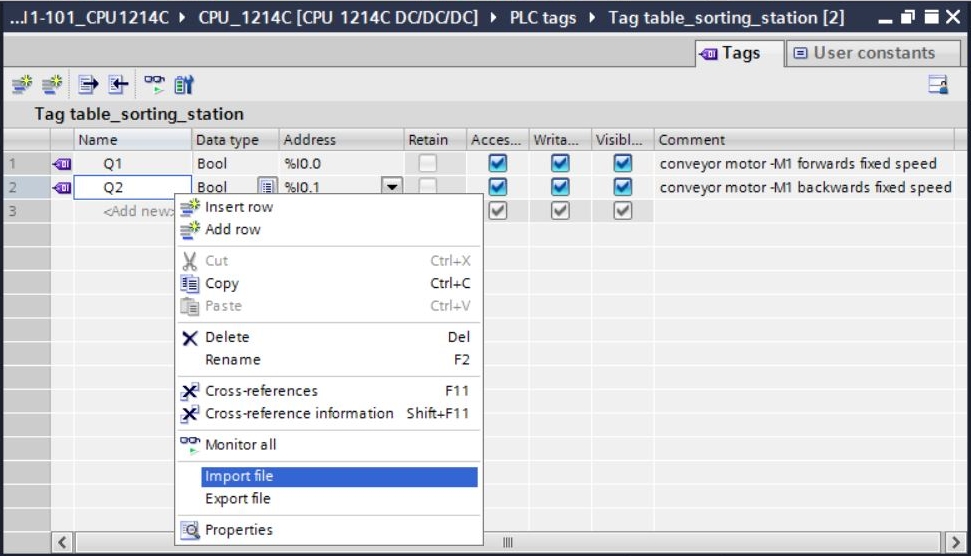
* Agregue en la línea 2 una nueva variable Q2. El TIA Portal ha asignado automáticamente el mismo tipo de datos que en la línea 1 y ha incrementado la dirección en 1 hasta %Q0.1. Introduzca el comentario "conveyor motor -M1 backwards fixed speed (Motor de cinta M1 hacia atrás, velocidad fija)".

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



* 1. 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)".

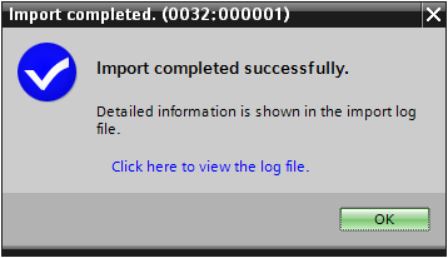
(→ Clic con el botón derecho en un campo vacío de la tabla de variables → Import file (Importar archivo))



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

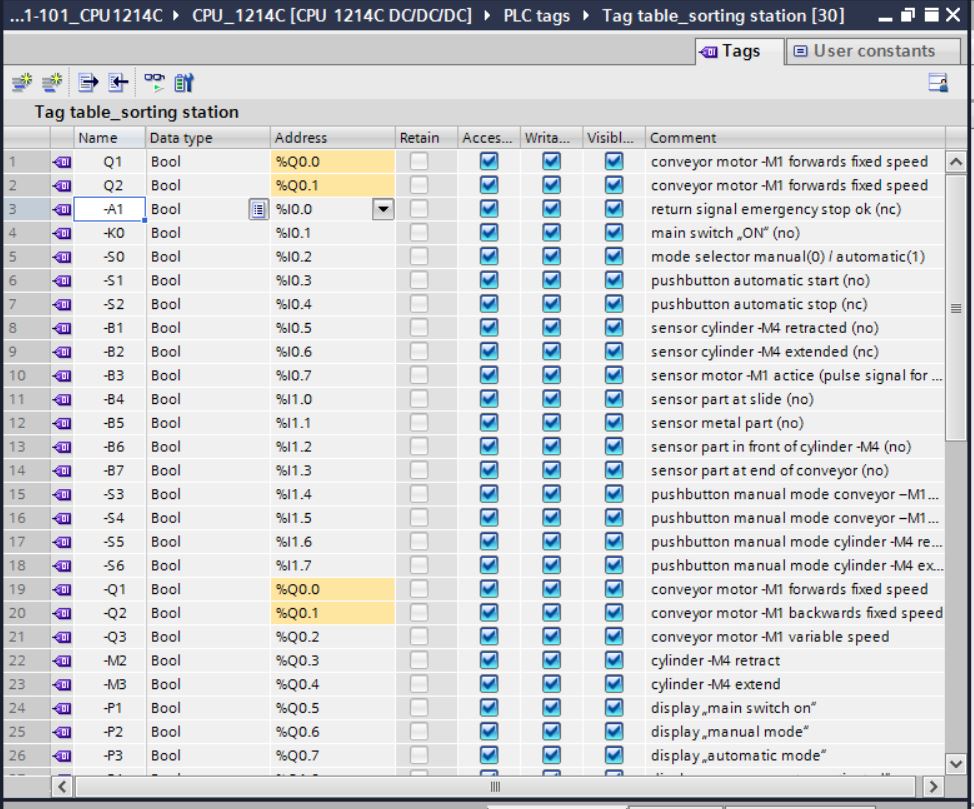
(→ SCE\_ES\_020-100\_Process Description Sorting Station → Open (Abrir))

* 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).



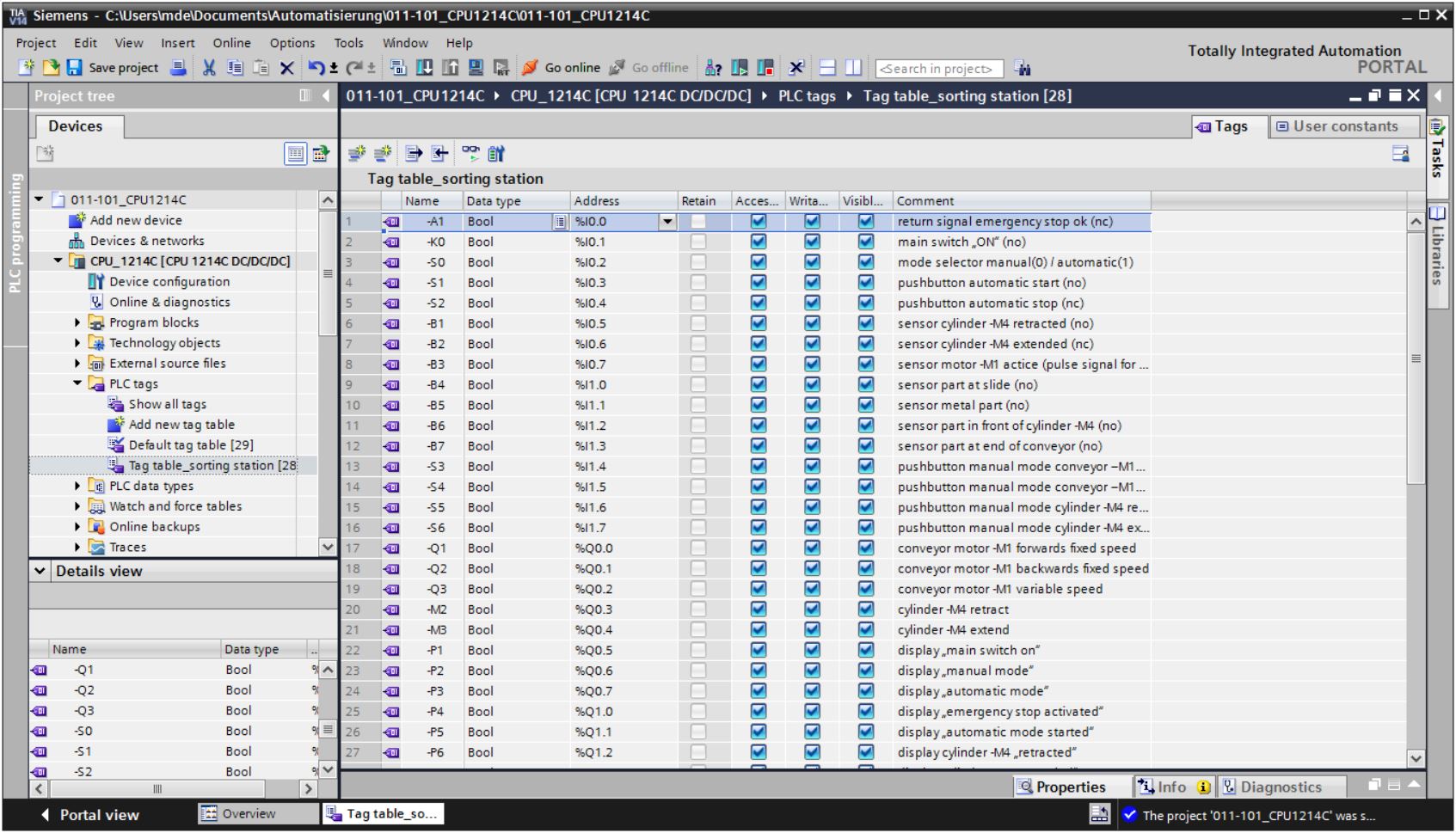
* Como verá, algunas direcciones aparecen resaltadas en naranja. Se trata de las que están duplicadas; se han numerado automáticamente las variables correspondientes a fin de evitar ambigüedades.
* Borre las variables duplicadas seleccionando las líneas correspondientes mediante la tecla Supr de su teclado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

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



* Ahora tendrá en su pantalla una tabla de símbolos completa de las entradas y salidas digitales. Guarde el proyecto con el nombre 031-100\_Programación\_FC.

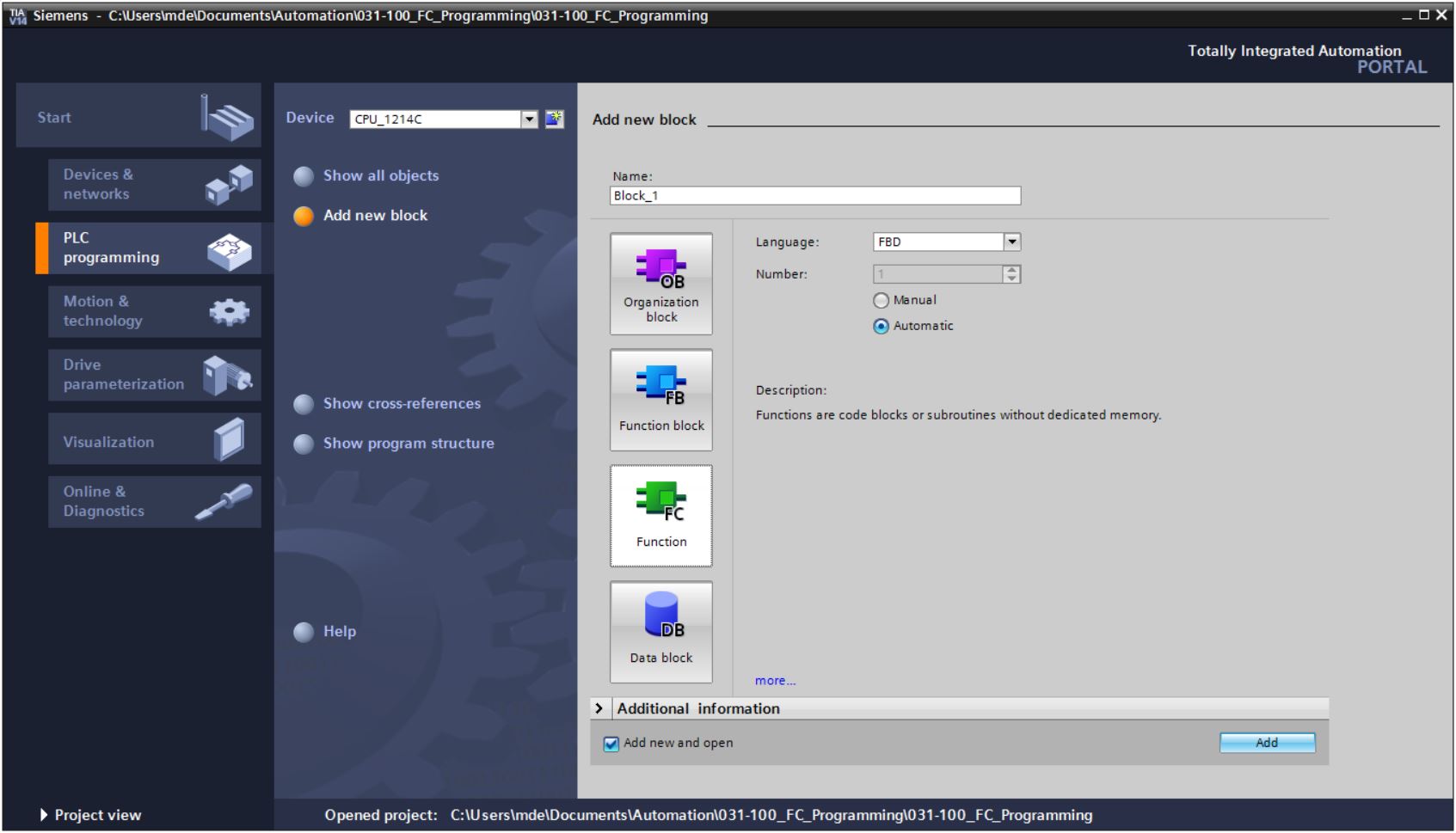
(→ Project (Proyecto) → Save as … (Guardar como …) → 031-100\_Programación\_FC → Save (Guardar))



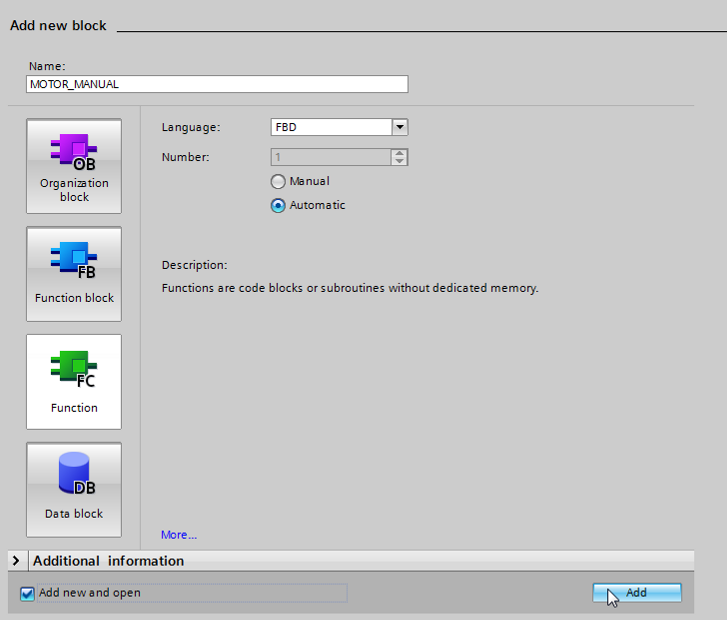
* 1. Creación de la función FC1 "MOTOR\_MANUAL" para el motor de cinta en modo Jog
* En la vista del portal, haga clic en el apartado "PLC programming (Programación de PLC)", opción "Add new block (Agregar nuevo bloque)" para crear una nueva función.

(→ PLC programming (Programación de PLC) → Add new block

(Agregar nuevo bloque)→ )



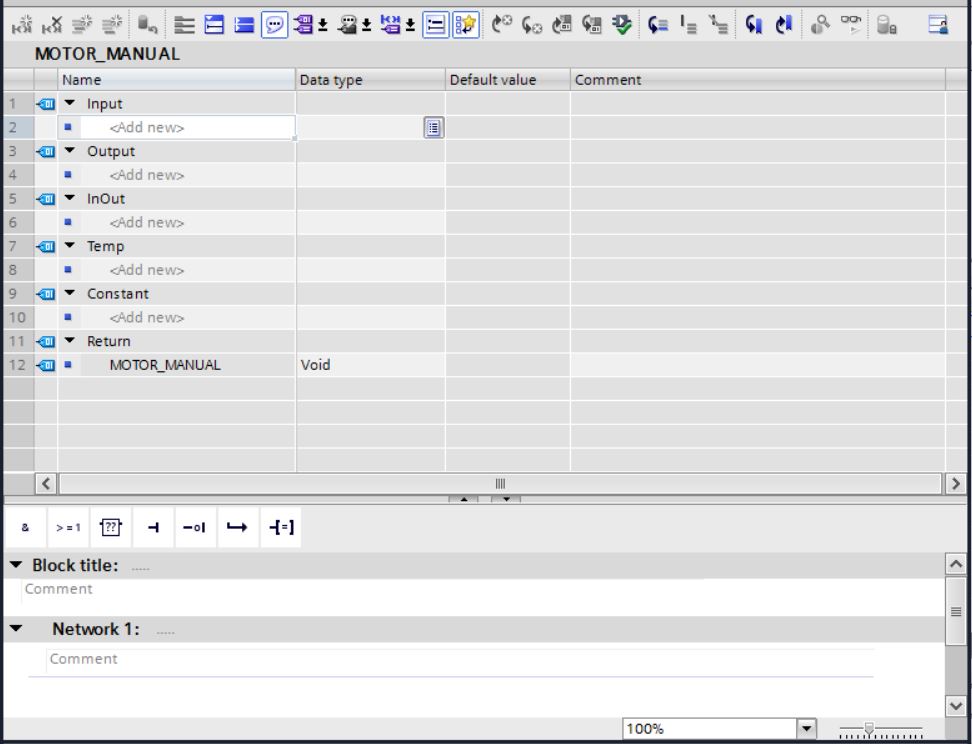
* Asigne al nuevo bloque el nombre: "MOTOR\_MANUAL", seleccione el lenguaje FDB (FUP) y deje que el programa asigne el número automáticamente. Si activa la marca de verificación "Add new and open (Agregar y abrir)", pasará automáticamente en la vista del proyecto al bloque de función que acaba de crear. Haga clic en "Add (Agregar)". (→ Name (Nombre): MOTOR\_MANUAL→ Language (Lenguaje): FDB (FUP) → Number (Número): Automatic (Automático) →  Add new and open (Agregar y abrir) → Add (Agregar))



* 1. Definición de la interfaz de la función FC1 "MOTOR\_MANUAL":

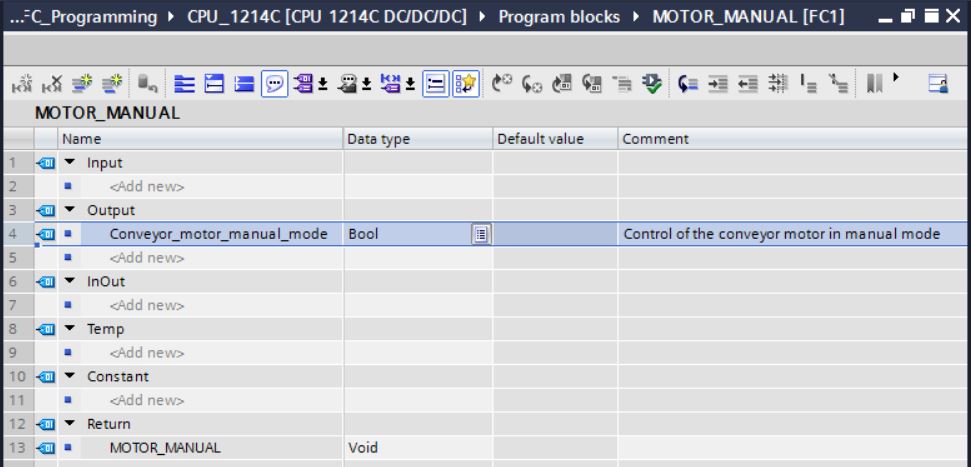
Al hacer clic en "Add new and open (Agregar y abrir)", se abre la vista del proyecto con una ventana para editar el bloque que se acaba de crear.

* En la parte superior de la vista de programación encontrará la descripción de interfaces de la función.



* Para controlar el motor de cinta se requiere una señal de salida binaria. Por ello debe crearse previamente la variable Output local #Conveyor\_motor\_manual\_mode (Motor\_cinta\_modo\_Jog), del tipo "Bool". Se asignará al parámetro el comentario "Control of the conveyor motor in manual mode (Control del motor de la cinta en modo Jog)".

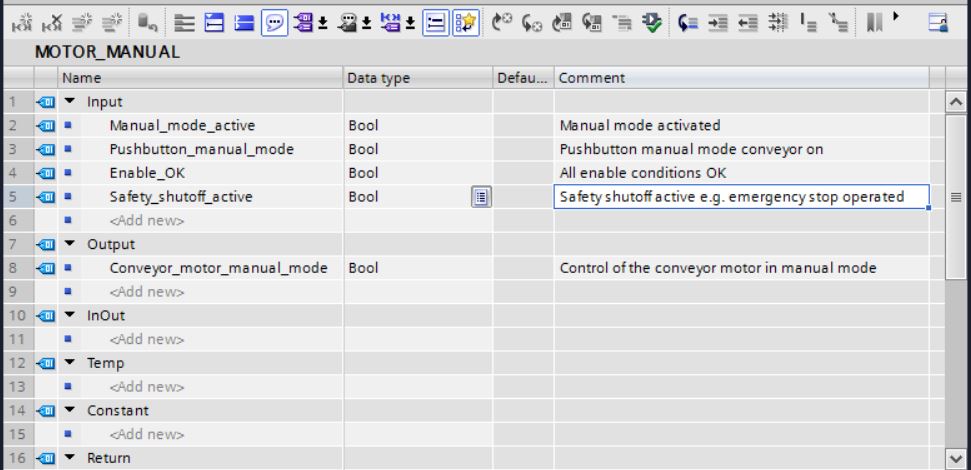
(→ Output (Salida): Conveyor\_motor\_manual\_mode (Motor\_cinta\_modo\_Jog) → Bool → Control of the conveyor motor in manual mode (Control del motor de la cinta en modo Jog))



* En primer lugar, agregue como interfaz de entrada en Input (Entrada) el parámetro #Manual\_mode\_active (Modo\_manual\_activo) y confirme con la tecla Intro o saliendo del campo de entrada. Se asignará automáticamente el tipo de datos "Bool". El tipo de datos se mantendrá. A continuación introduzca el comentario "Manual mode activated (Modo de operación manual activado)".

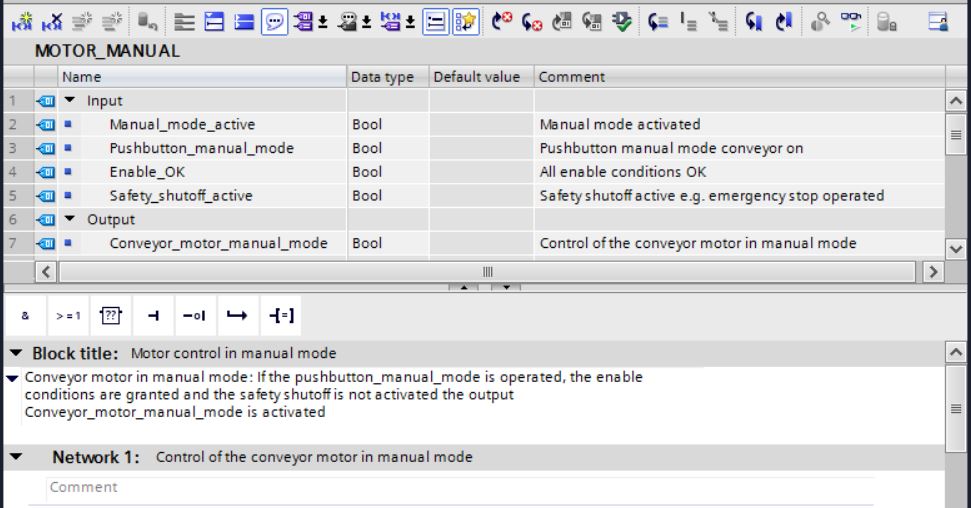
(→ Manual\_mode\_active (Modo\_manual\_activo) → Intro → Bool → Manual mode activated (Modo de operación manual activado))

* A continuación introduzca en Input (Entrada), como parámetros de entrada binarios, #Pushbutton\_manual\_mode (Pulsador\_modo\_Jog), #Enable\_OK (Habilitación\_OK) y #Safety\_shutoff\_active (Desconexión\_seguridad\_activa) y compruebe los tipos de datos de dichos parámetros. Añada comentarios descriptivos.



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

(→ Block title (Título del bloque): Motor control in manual mode (Control del motor en modo manual) → Network 1 (Segmento 1): Control of the conveyor motor in manual mode (Control del motor de la cinta en modo Jog))

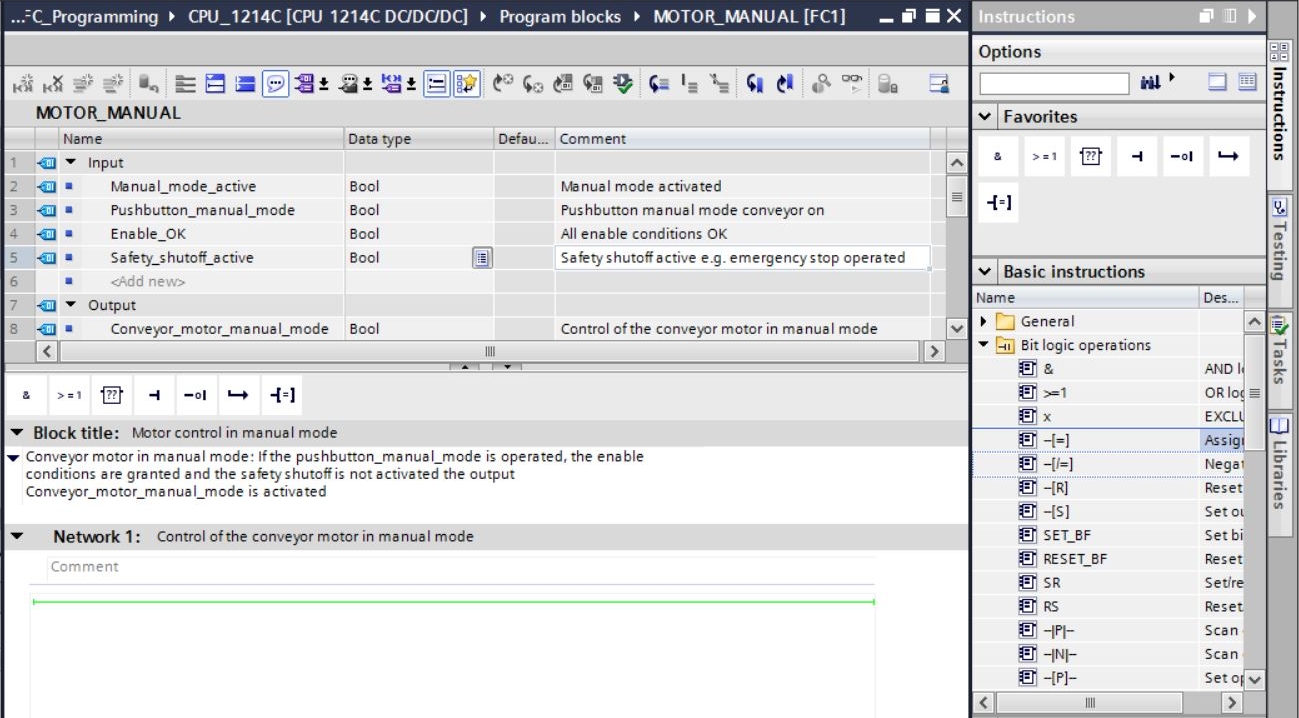


* 1. Programación del FC1: MOTOR\_MANUAL
* En la ventana de programación, debajo de la descripción de interfaces, encontrará una barra de herramientas con varias funciones lógicas y, debajo de ella, un área con segmentos. Hemos definido ya en ella el título del bloque y el título del primer segmento. Dentro de los segmentos, la programación se realiza utilizando distintos bloques lógicos. Para mayor claridad, se recomienda crear varios segmentos. A continuación le presentamos las distintas posibilidades para insertar bloques lógicos.



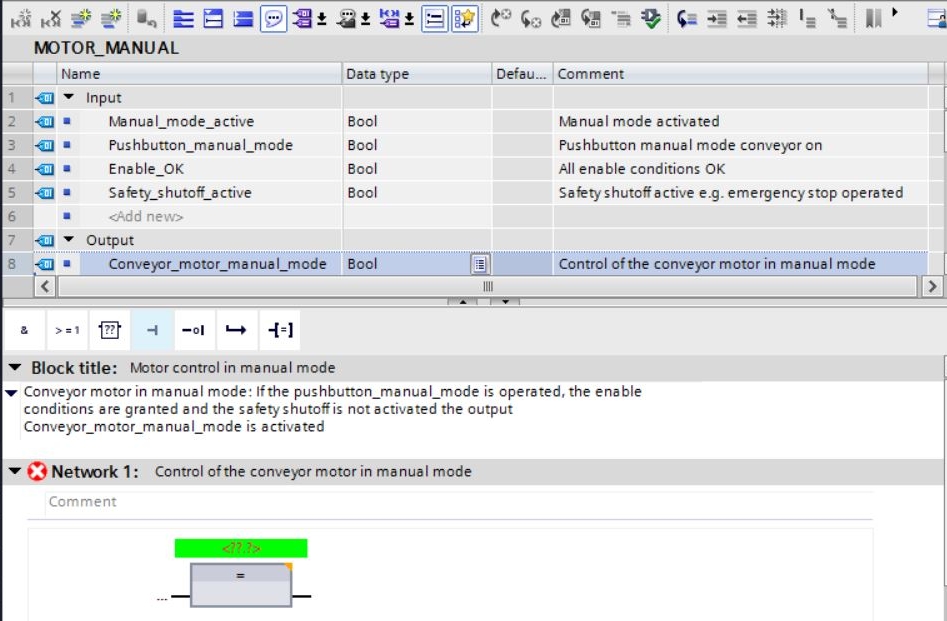
* En el lado derecho de la ventana de programación hay una lista de instrucciones que pueden utilizarse en el programa. Busque en → Basic instructions (Instrucciones básicas) → Bit logic operations (Operaciones lógicas con bits) la función –[=] (asignación) y cópiela mediante "arrastrar y soltar" a su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

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



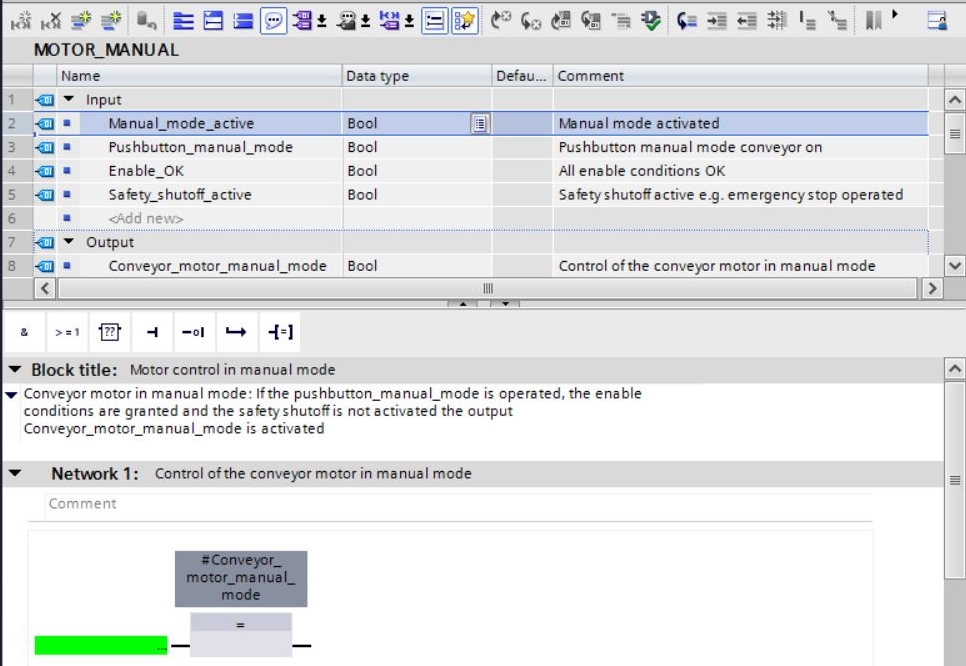
* Ahora, desplace el parámetro de salida #Conveyor\_motor\_manual\_mode (Motor\_cinta\_ modo\_Jog), mediante "arrastrar y soltar", a <??.?>, por encima del bloque que acaba de insertar. La mejor manera de seleccionar un parámetro en la descripción de interfaces es agarrarlo por el icono azul .

(→  Conveyor\_motor\_manual\_mode (Motor\_cinta\_modo\_Jog))

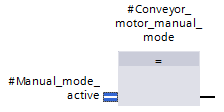


* Con esto se determina que el parámetro #Conveyor\_motor\_manual\_mode (Motor\_cinta\_ modo\_Jog) se escriba mediante este bloque. Para que esto sea efectivamente así, faltan todavía las condiciones de entrada. Desplace el parámetro de entrada #Manual\_mode\_active (Modo\_manual\_activo), mediante "arrastrar y soltar", a "…" en el lado izquierdo del bloque de asignación.

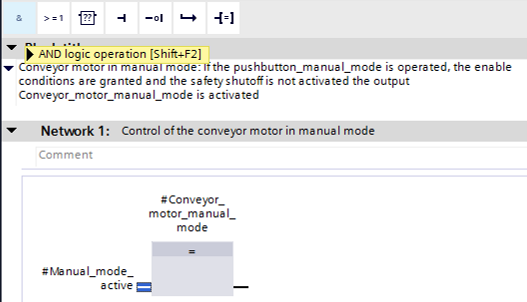
(→  Manual\_mode\_active (Modo\_manual\_activo))



* Además, se combinará la entrada del bloque de asignación con otros parámetros mediante el operador lógico Y. Para ello, haga clic en primer lugar en la entrada del bloque en el que está interconectado #Manual\_mode\_active (Modo\_manual\_activo). La raya de entrada aparecerá sobre fondo azul.

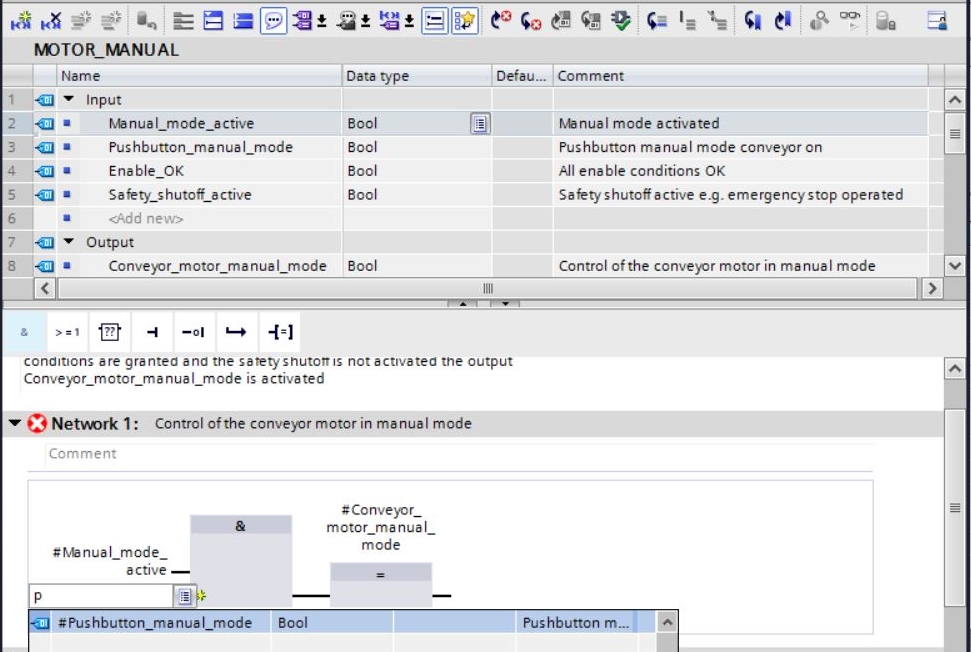


* Haga clic en el icono  de la barra de herramientas lógicas para insertar una combinación mediante operador Y entre la variable #Manual\_mode\_active (Modo\_manual\_activo) y el bloque de asignación.



* Haga doble clic en la segunda entrada de la combinación lógica Y <??.?> y, en el campo que se abrirá, introduzca la letra "P" para ver una lista de las variables disponibles que empiezan por esa letra.Haga clic en la variable #Pushbutton\_manual\_mode (Pulsador\_modo\_Jog) y pulse → Intro para aceptar.

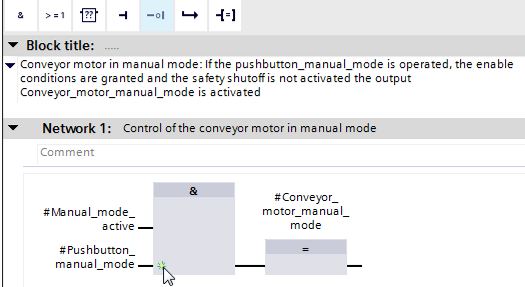
(→ Bloque & → <??.?> → P → #Pushbutton\_manual\_mode (Pulsador\_modo\_Jog) → Intro)



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

* Para que solo sea posible controlar la salida cuando se haya otorgado la habilitación y no esté activa la desconexión de seguridad, deben combinarse mediante el operador Y las variables de entrada #Enable\_OK (Habilitación\_OK) y #Safety\_shutoff\_active

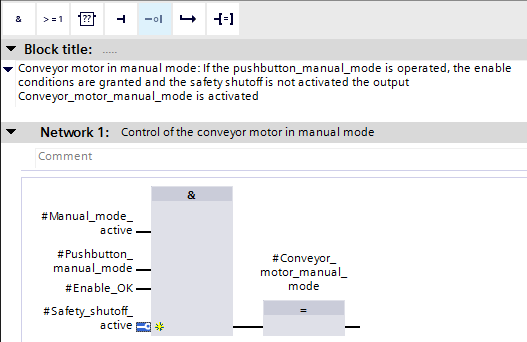
(Desconexión\_seguridad\_activa). Para ello, haga doble clic en el asterisco amarillo  del elemento Y a fin de agregar dos entradas más.



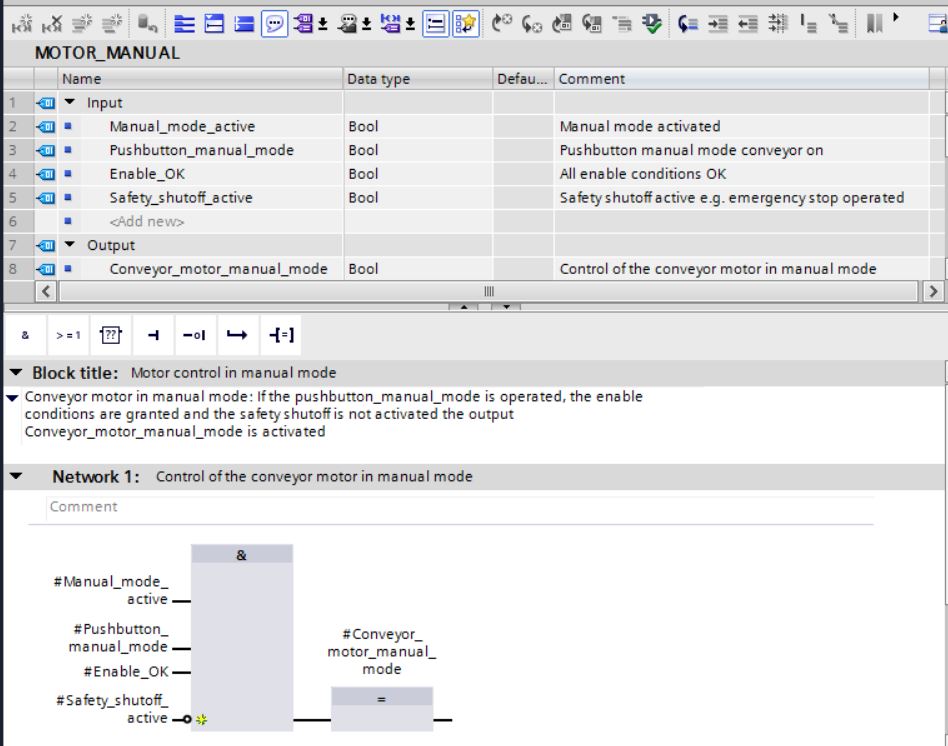
* Inserte las variables de entrada #Enable\_OK (Habilitación\_OK) y #Safety\_shutoff\_active (Desconexión\_seguridad\_activa) en las nuevas entradas del elemento Y.



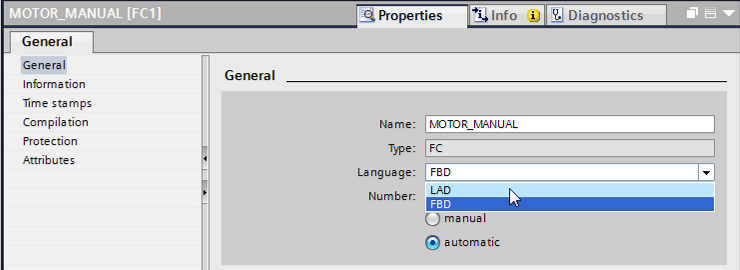
* Niegue la entrada conectada con el parámetro #Safety\_shutoff\_active (Desconexión\_ seguridad\_activa); para ello, selecciónelo y haga clic en .



* No olvide hacer clic regularmente en 27_save. A continuación se muestra la función ya creada "MOTOR\_MANUAL [FC1]" 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))

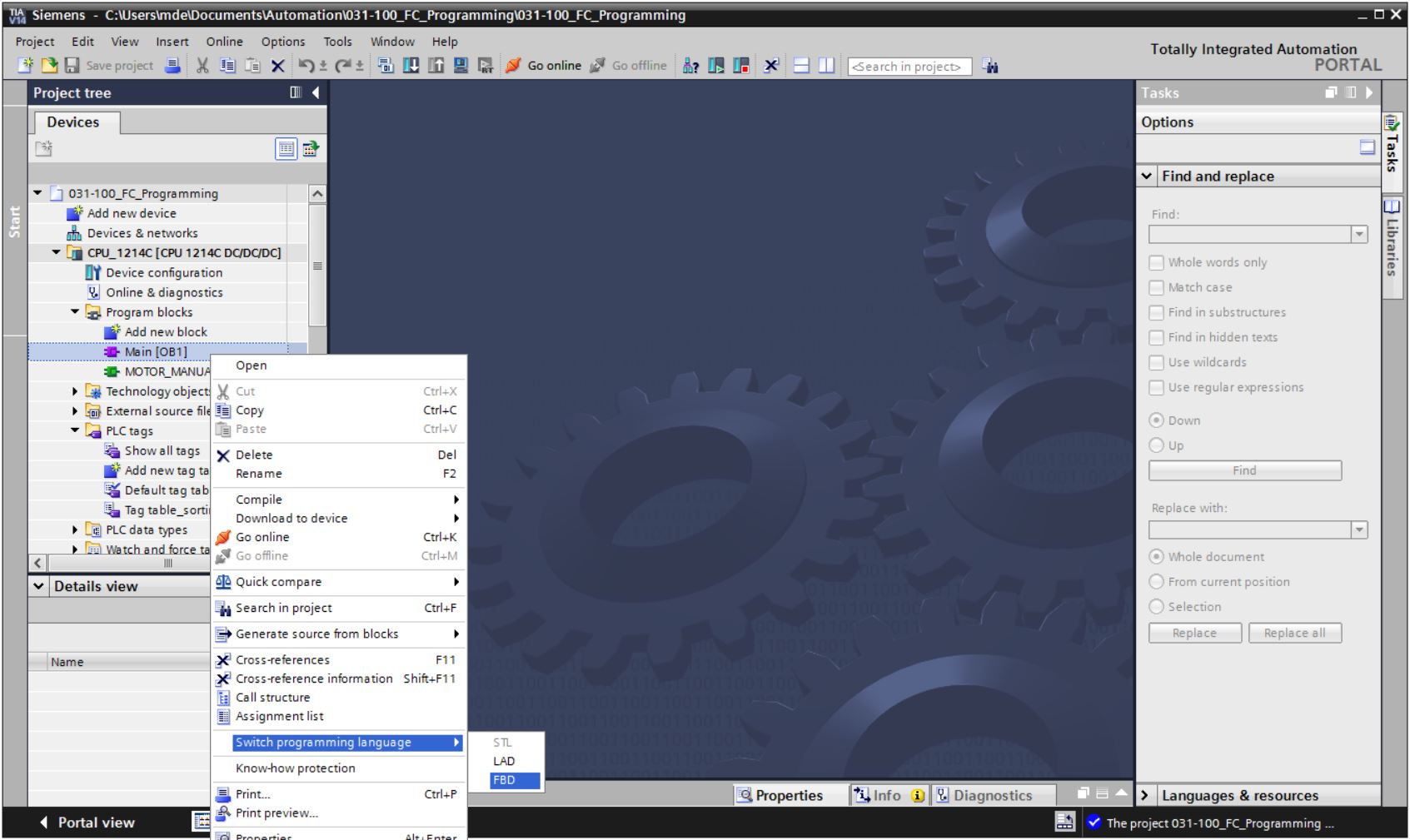


* En KOP, el programa es como se indica a continuación.

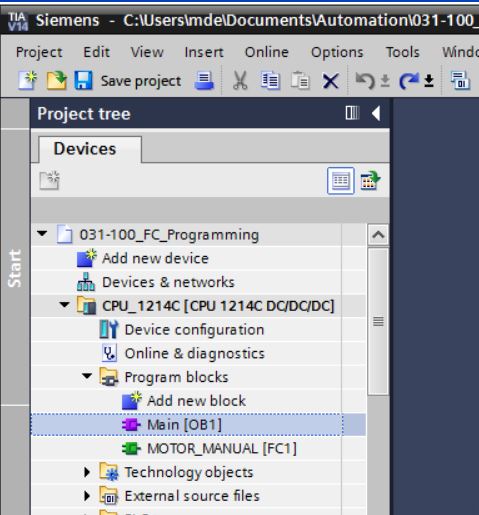


* 1. Programación del bloque de organización OB1: control de la marcha de la cinta hacia delante en modo manual
* Antes de programar el bloque de organización "Main[OB1]", cambiaremos el lenguaje de programación a FUP (Esquema de funciones). Para ello, haga clic en primer lugar con el botón izquierdo 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.

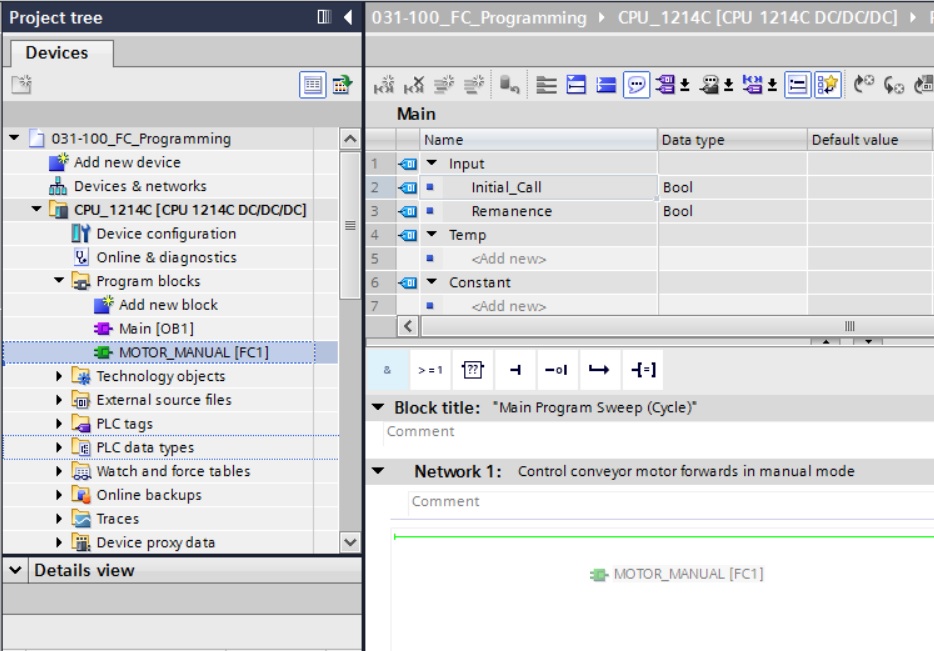


* Asigne al Network 1 (Segmento 1) el nombre "Control conveyor motor forwards in manual mode (Control de la marcha de la cinta hacia delante en modo manual o Jog)".

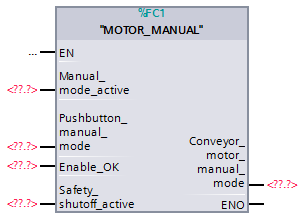
(→ Network 1 (Segmento 1):… → Control conveyor motor forwards in manual mode   
(Control de la marcha de la cinta hacia delante en modo manual o Jog))



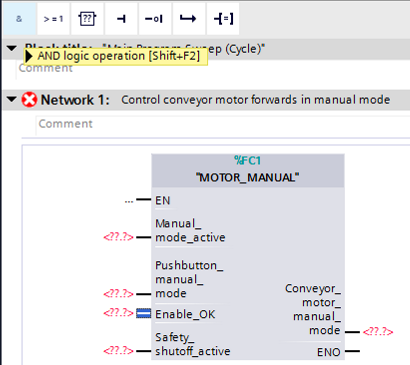
* Ahora desplace la función "MOTOR\_MANUAL [FC1]" hasta la línea verde del Network 1 (Segmento 1) mediante "arrastrar y soltar".



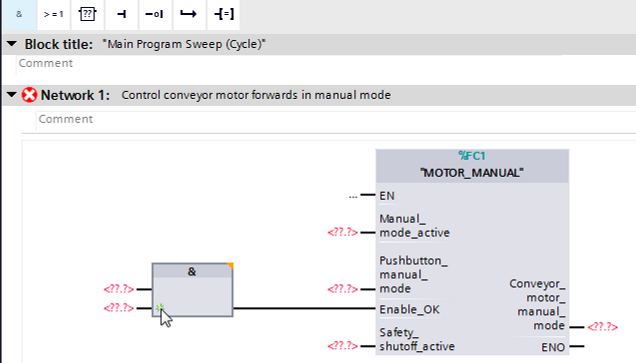
* Se insertará en el Network 1 (Segmento 1) un bloque con la interfaz que se ha definido y las conexiones EN y ENO.



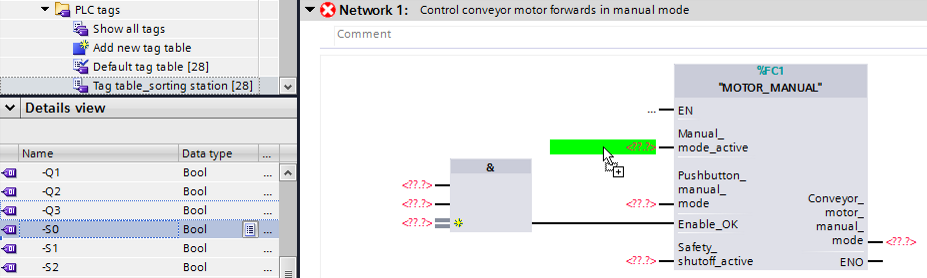
* Para insertar un Y antes del parámetro de entrada "Enable\_OK (Habilitación\_OK)", seleccione dicha entrada e inserte el Y haciendo clic en el icono  de la barra de herramientas lógicas. (→)



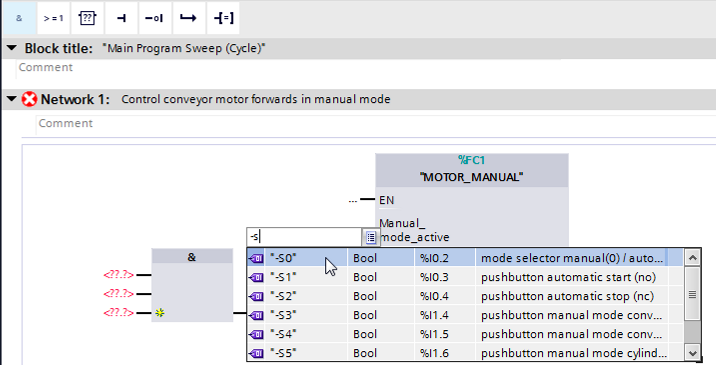
* Haga clic en el asterisco amarillo  del elemento Y para agregar otra entrada. (→)



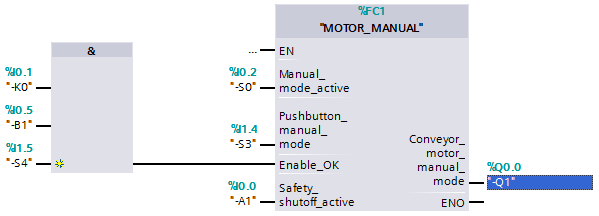
* Para interconectar el bloque con las variables globales de la "Tag\_table\_sorting\_station (Tabla\_variables\_planta\_clasificación)", disponemos de dos posibilidades:
* La primera consiste en seleccionar la "Tag\_table\_sorting\_station (Tabla\_variables\_ planta\_clasificación)" en el árbol del proyecto y copiar la variable global deseada desde la vista detallada a la interfaz de FC1 mediante "arrastrar y soltar" (→ Tag\_table\_sorting\_station (Tabla\_variables\_planta\_clasificación) → Details view (Vista detallada) → -S0 → Manual\_mode\_active (Modo\_manual\_activo)).



* La segunda, en introducir en <??.?> las letras iniciales de la variable global deseada (p. ej., "-S") y, en la lista que aparecerá, seleccionar la variable global de entrada "-S0" (%I0.2) (→ Manual\_mode\_active (Modo\_manual\_activo) → -S → -S0).



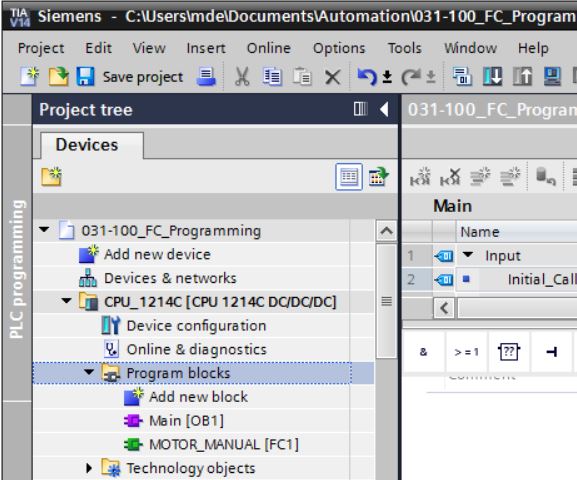
* Inserte las restantes variables de entrada "-S3", "-K0", "-B1", "-S4" y "-A1" y, a continuación, en la salida "Conveyor\_motor\_manual\_mode (Motor\_cinta\_modo\_Jog)", la variable de salida "-Q1" (%Q0.0).



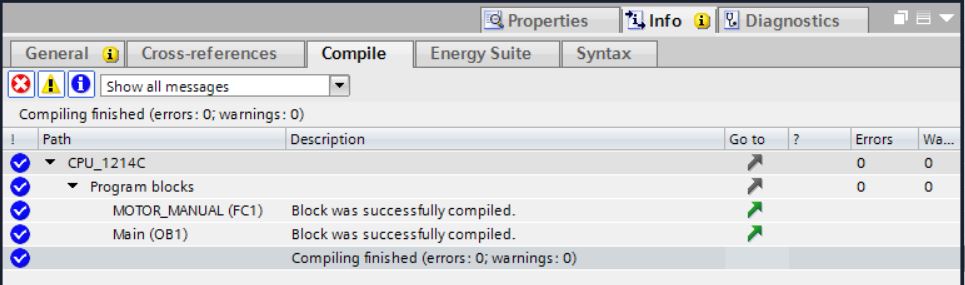
* Niegue las consultas de las variables de entrada "-S0", "-S4" y "-A1" seleccionándolas y haciendo clic a continuación en  (→ -S0 →  → -S4 →  → -A1 → ).



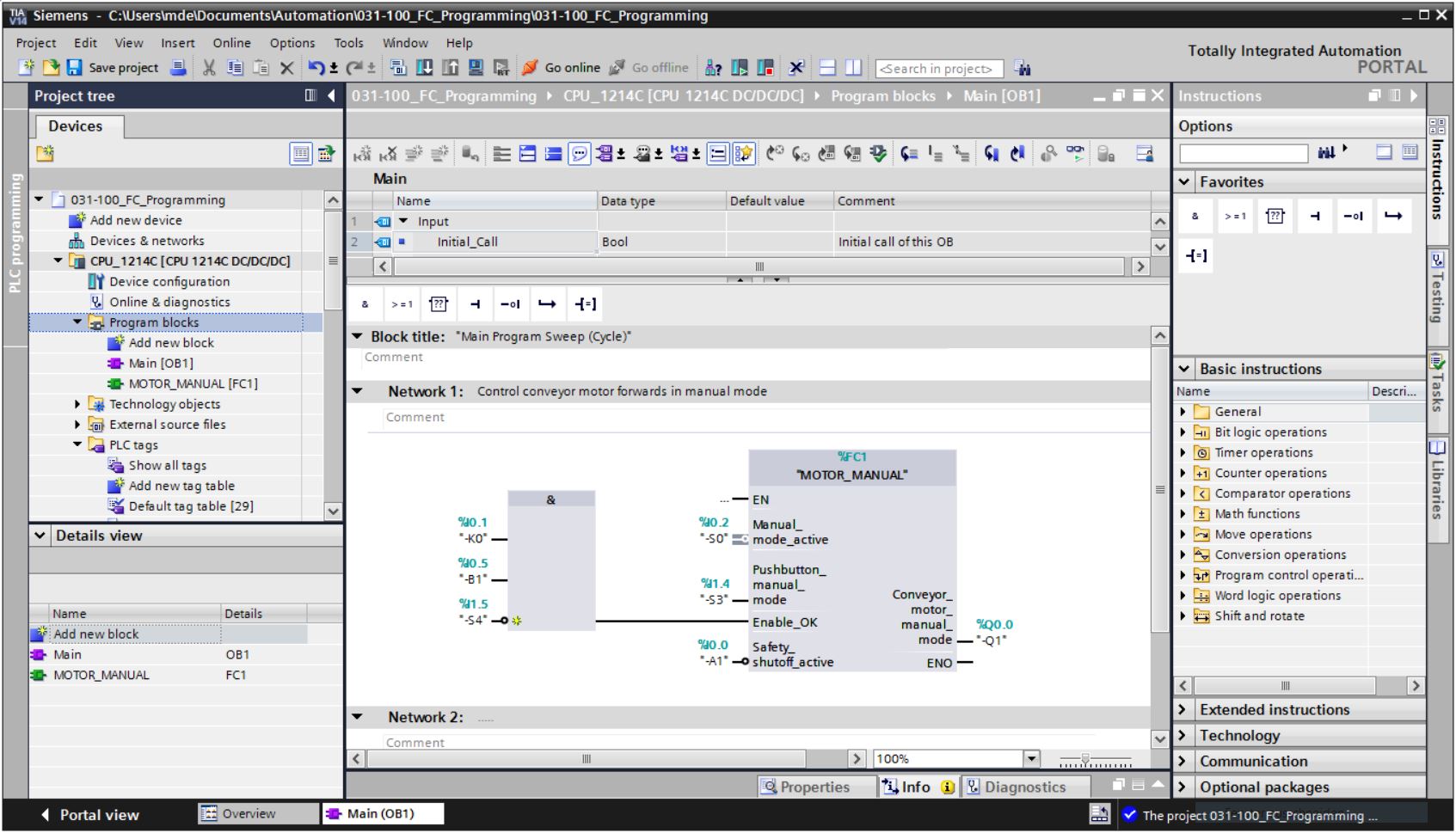
* 1. Guardado y compilación del programa
* Para guardar el proyecto, seleccione en el menú el botón 27_save. Para compilar todos los bloques, haga clic en la carpeta "Program blocks (Bloques de programa)" y seleccione en el menú el icono de compilación D:\00_DATA\SIEMENS\Unterlagen\08_Ausbildungsunterlage_TIA-Portal_R1502_dt\032-100 FC-Programmierung\pics\052.jpg (→ 27_save → Program blocks (Bloques de programa) → D:\00_DATA\SIEMENS\Unterlagen\08_Ausbildungsunterlage_TIA-Portal_R1502_dt\032-100 FC-Programmierung\pics\052.jpg).



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

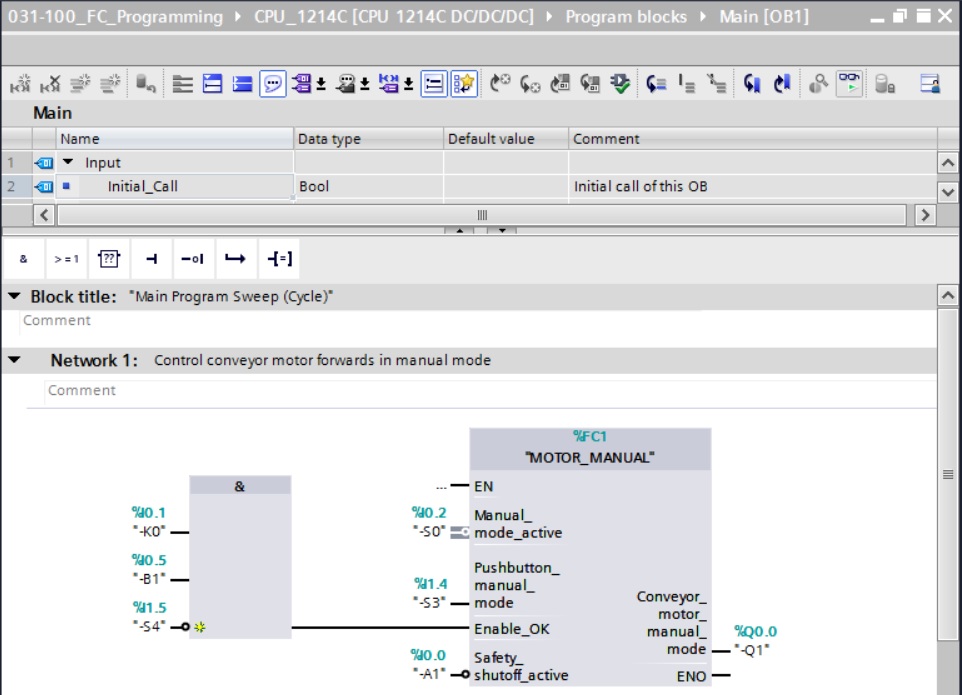


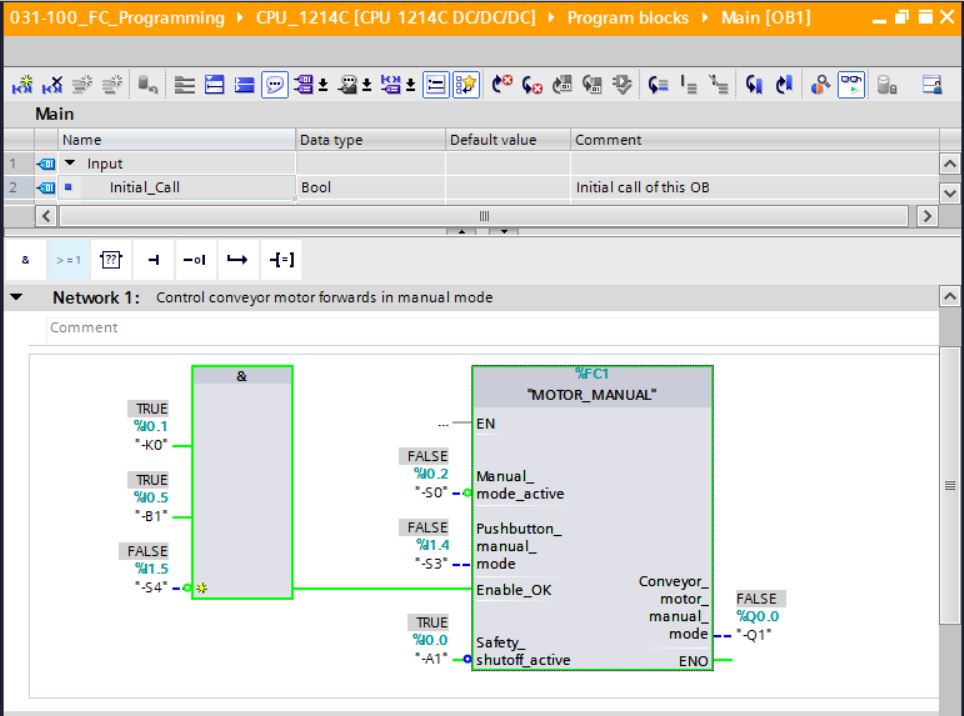
* 1. Carga del programa
* Una vez realizada la compilación correctamente, puede cargar el controlador completo, con el programa que ha creado, del modo descrito en los módulos dedicados a la configuración hardware (→ ).



* 1. 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 D:\00_DATA\SIEMENS\Unterlagen\08_Ausbildungsunterlage_TIA-Portal_R1502_dt\032-100 FC-Programmierung\pics\055b.jpg

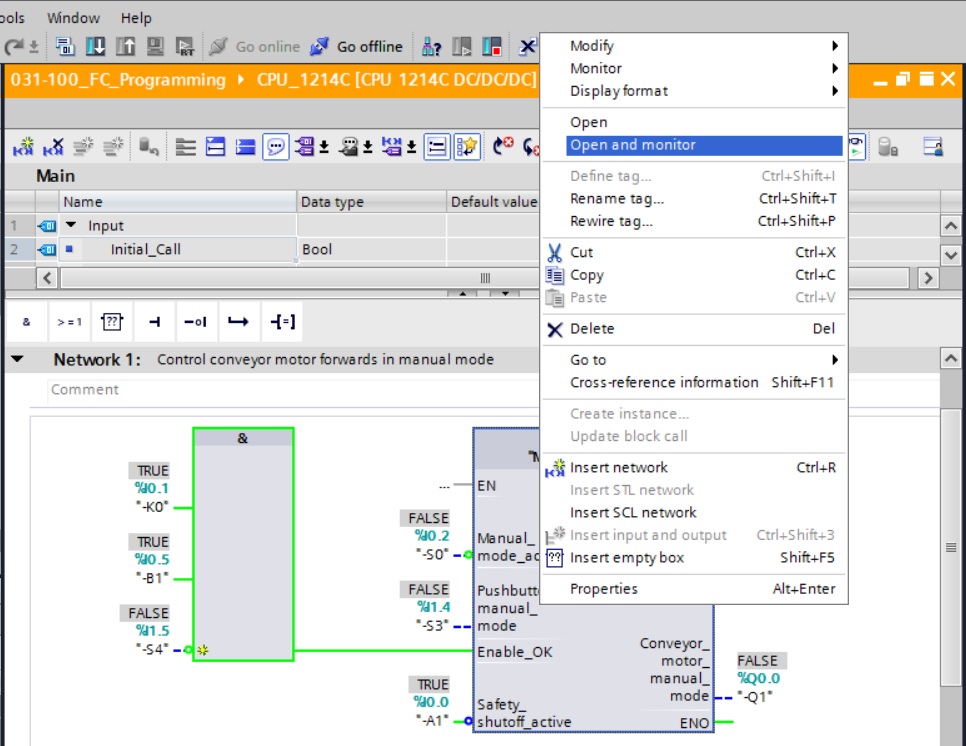
(→ Main [OB1] → D:\00_DATA\SIEMENS\Unterlagen\08_Ausbildungsunterlage_TIA-Portal_R1502_dt\032-100 FC-Programmierung\pics\055b.jpg).

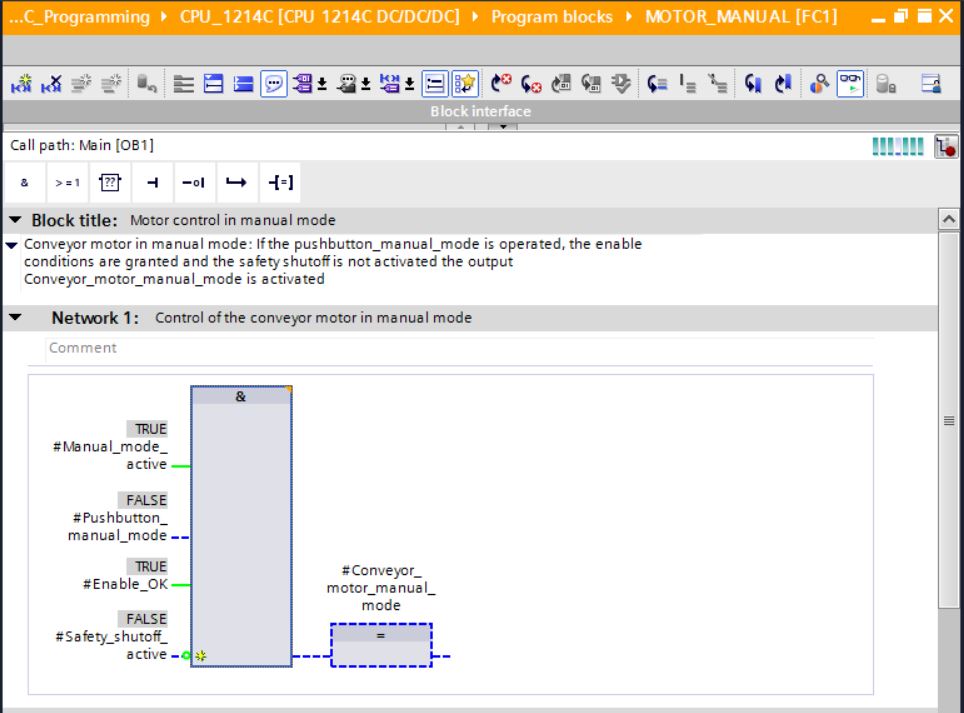




***Nota:*** *la visualización se realiza con referencia a la señal y dependiendo del controlador. Los estados lógicos de los bornes se indican mediante TRUE y FALSE.*

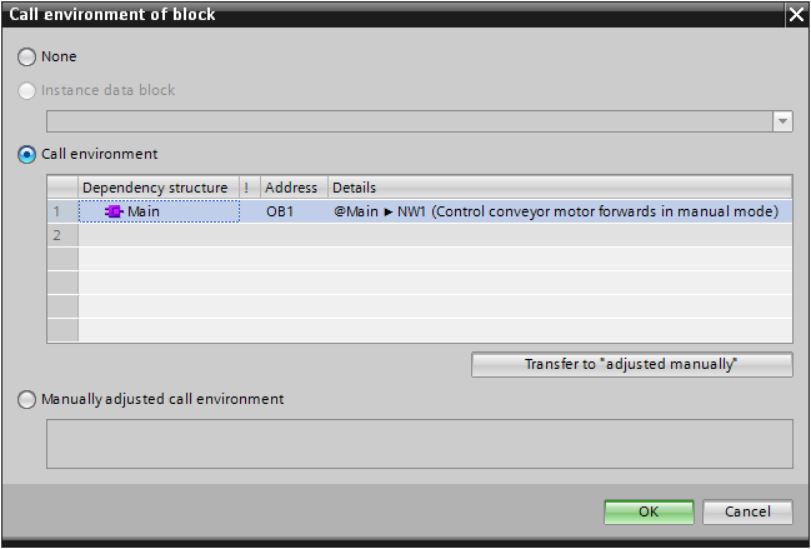
* Para abrir y visualizar la función "MOTOR\_MANUAL" [FC1] llamada en el bloque de organización "Main [OB1]", selecciónela directamente tras hacer clic en ella con el botón derecho del ratón (→ "MOTOR\_MANUAL" [FC1] → Open and monitor (Abrir y visualizar)).



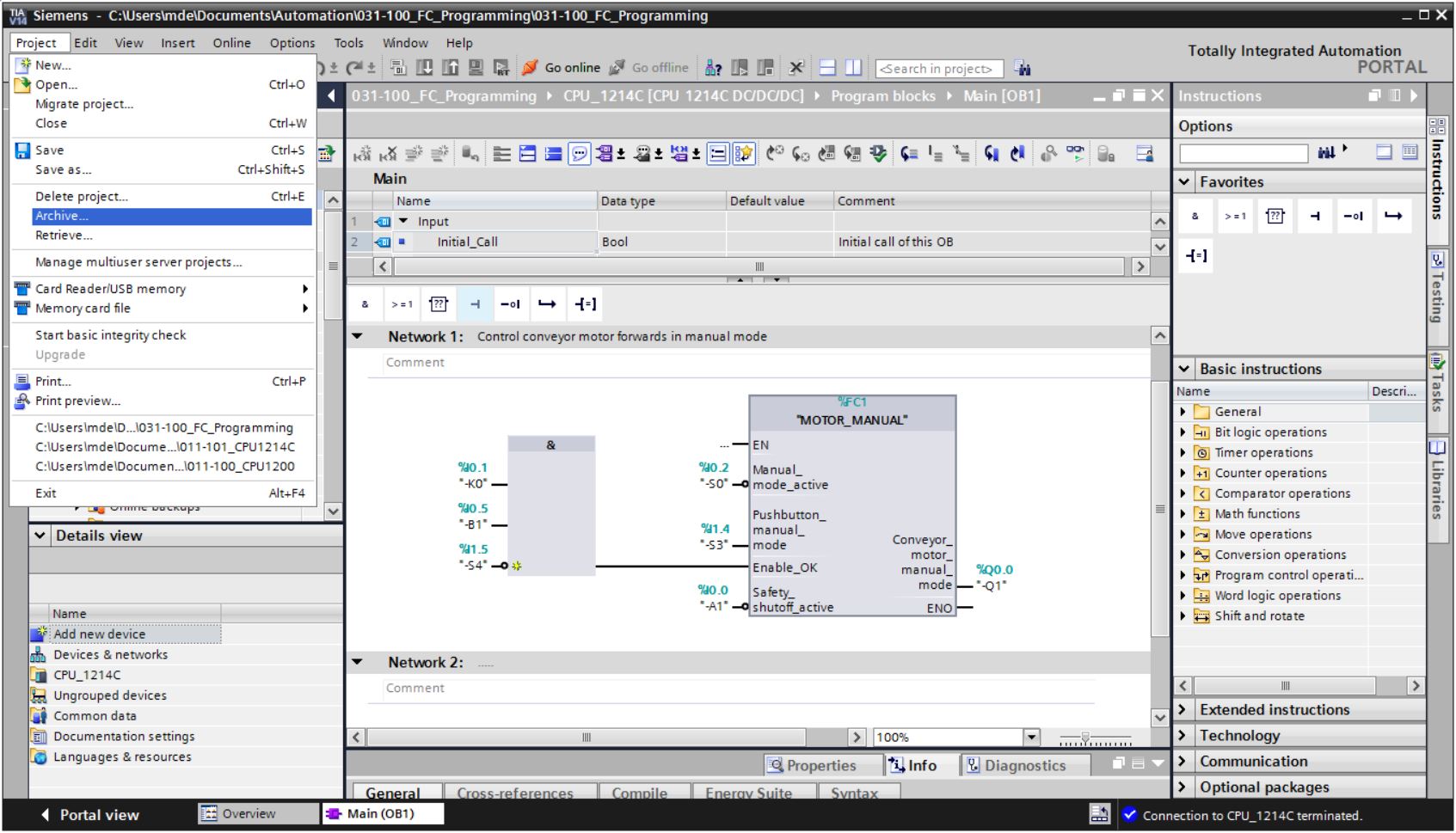


***Nota:*** *la visualización se realiza con referencia a la función y dependiendo del controlador. La activación de los sensores y el estado de la instalación se indican mediante TRUE y FALSE.*

* Si se desea visualizar una ubicación concreta de la función "MOTOR\_MANUAL" [FC1], debe seleccionarse el entorno de llamada mediante el icono D:\00_DATA\SIEMENS\Unterlagen\08_Ausbildungsunterlage_TIA-Portal_R1502_dt\032-100 FC-Programmierung\pics\058b.jpg (→ D:\00_DATA\SIEMENS\Unterlagen\08_Ausbildungsunterlage_TIA-Portal_R1502_dt\032-100 FC-Programmierung\pics\058b.jpg → Call environment (Entorno de llamada) → OK (Aceptar)).



* 1. 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) → 031-100\_Programación\_FC…. → Save (Guardar)).



* 1. Lista de comprobación

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

# Ejercicio

* 1. Tarea planteada: ejercicio

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

* Operación manual: control de la marcha de la cinta hacia atrás en modo manual o Jog.
  1. Esquema tecnológico

Este es el esquema tecnológico para la tarea asignada.

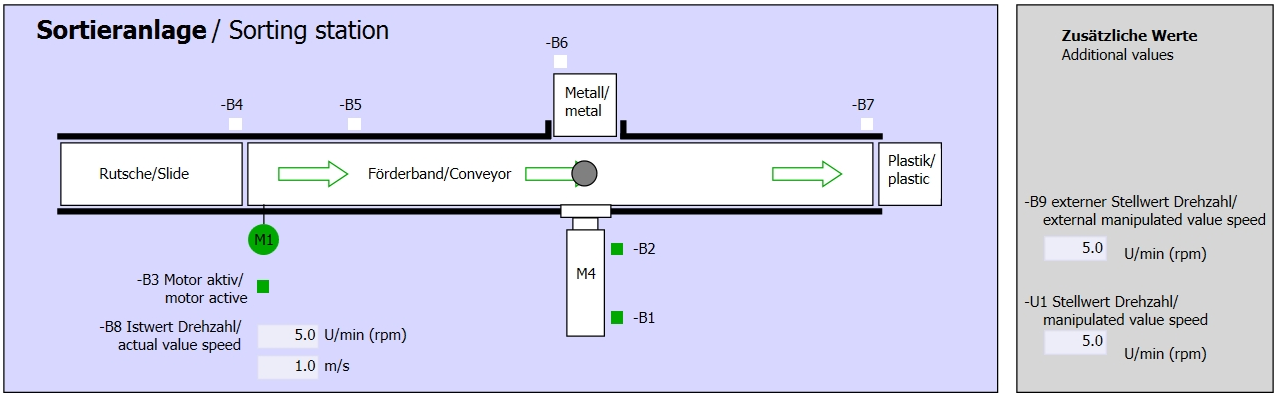


Figura 10: Esquema tecnológico

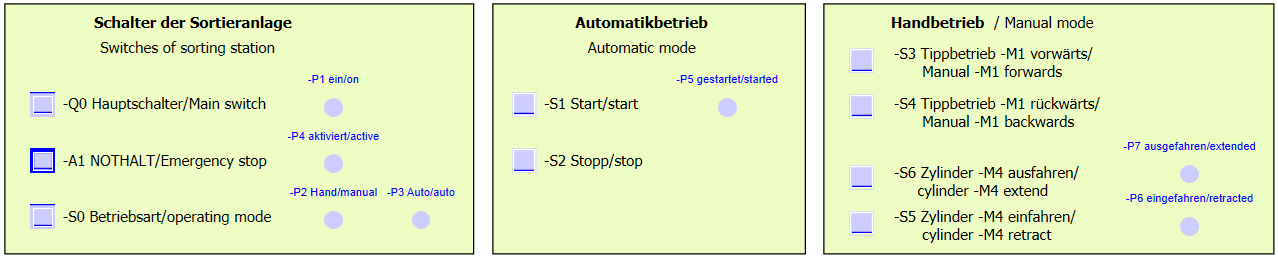


Figura 11: Pupitre de mando

* 1. 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 0.5 | BOOL | -B1 | Sensor cilindro -M4 introducido | NA |
| I 1.4 | BOOL | -S3 | Pulsador modo Jog cinta -M1 hacia delante | NA |
| I 1.5 | BOOL | -S4 | Pulsador modo Jog cinta -M1 hacia atrás | NA |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DO** | **Tipo** | **Identificador** | **Función** |  |
| Q 0.1 | BOOL | -Q2 | Motor de cinta -M1 hacia atrás, velocidad fija |  |

***Leyenda de la lista de asignación***

|  |  |
| --- | --- |
| DQ | Salida digital |
| AQ | Salida analógica |
| S | Salida |

|  |  |
| --- | --- |
| DI | Entrada digital |
| AI | Entrada analógica |
| E | Entrada |
| NC | Normalmente cerrado |
| NA | Normalmente abierto |

* 1. Planificación

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

* 1. 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 MANUAL (-S0 = 0)  Activar el modo Jog cinta hacia atrás (-S4 = 1)  Motor de cinta hacia atrás, velocidad fija (-Q2 = 1) |  |
| 4 | Como 8, pero activando PARADA DE EMERGENCIA (-A1 = 0) →  -Q2 = 0 |  |
| 5 | Como 8, pero en modo de operación AUTO (-S0 = 1) → -Q2 = 0 |  |
| 6 | Como 8, pero desconectando la instalación (-K0 = 0) → -Q2 = 0 |  |
| 7 | Como 8, pero con el cilindro no introducido (-B1 = 0) → -Q2 = 0 |  |
| 8 | Como 8, pero activando también Modo Jog cinta hacia delante  (-S3 = 1) → -Q1 = 0 y también -Q2 = 0 |  |
| 9 | Proyecto archivado correctamente |  |