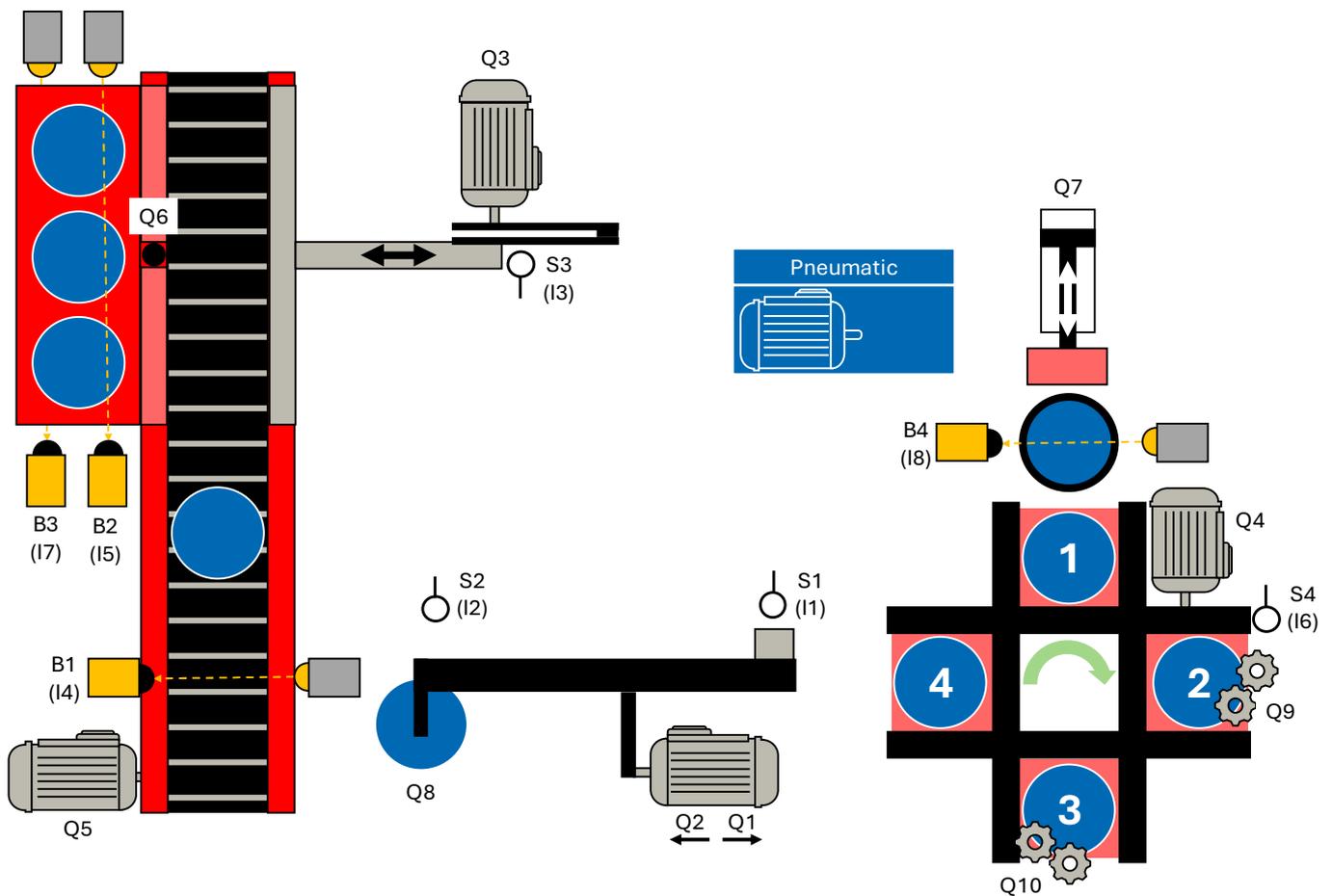


# Línea de producción de 24 V

Planificar y aplicar la secuencia automática



## Índice

10	Planificar y aplicar la secuencia automática .....	1
10.1	Ejercicio: Planificación de la cadena de procesos con GRAFCET - ruta de transporte.....	1
10.2	Ejercicio: Planificación de la cadena de secuencias con GRAFCET - convertidor.....	5
10.3	Ejercicio: Planificación de la cadena de procesos con GRAFCET - Revista	10
10.4	Ejercicio: Planificación de la cadena de secuencias con GRAFCET - Estación de procesamiento.....	13
10.5	Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [FBD] - Ruta de transporte.....	16
10.6	Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Ruta de transporte.....	24
10.7	Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FBD] - Convertidor .....	32
10.8	Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Conversor.....	40
10.9	Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FUP] - Revista .....	47
10.10	Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Revista .....	54
10.11	Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FUP] - Estación de tratamiento.....	61
10.12	Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Estación de tratamiento .....	68
10.13	Ejercicio: Crear bloque de función para mesa giratoria [FBD].....	75
10.14	Ejercicio: Crear bloque de funciones para mesa giratoria [ST / SCL].....	82

## 10 Planificar y aplicar procesos automáticos



### 10.1 Ejercicio: Planificación de la cadena de procesos con GRAFCET - ruta de transporte

#### Objetivo:

Puedo utilizar la descripción funcional para crear una cadena de secuencia GRAFCET para la sección de transporte de la línea de producción.

#### Tarea:

Cree una cadena secuencial en GRAFCET para la ruta de transporte de la línea de producción de forma que la función se implemente basándose en la descripción del sistema.

#### Función:

##### 1. Paso inicial

En el paso inicial no se realiza ninguna acción.

La cadena de peldaños permanece en este paso hasta que se introduce una paleta vacía. La paleta interrumpe la barrera de luz B3. B2 no debe interrumpirse.

##### 2. Mover el control deslizante a la posición inicial

Una vez insertada una paleta vacía, la corredera (Q3) puede desplazarse a la posición inicial. La corredera sólo puede activarse si no está en la posición inicial y hay una paleta delante de ella.

Si se activa el empujador, también debe ponerse a cero el contador de piezas en la cinta.

Si la corredera está en la posición inicial (S3 accionado), puede pasar al siguiente paso.

##### 3. Cerrar separador

Ahora que la corredera está en la posición inicial, se puede cerrar el separador. La válvula (Q6) debe estar permanentemente activada para mantener cerrado el separador.

500 milisegundos después de la activación de la válvula, se puede suponer que el separador está cerrado. Esto significa que puede pasar al siguiente paso.

##### 4. Esperando pieza

En este paso, el sistema se detiene hasta que la unidad de transferencia ha colocado una nueva pieza en la cinta; no se realiza ninguna acción. La nueva pieza se reconoce mediante la interrupción de la barrera luminosa B1. Antes de pasar al siguiente paso, la unidad de transferencia debe haber abandonado de nuevo la cinta (S1 no accionado).

##### 5. Retirar la pieza

Si una pieza de trabajo se encuentra sobre la cinta y la unidad de transferencia ha abandonado la estación, puede transportarse fuera de ella. La cinta (Q5) sólo puede conectarse si la corredera se encuentra en la posición inicial (S3 accionado), el separador está cerrado (Q6 accionado) y la unidad de transferencia no se encuentra en la zona de la cinta (S1).

La cinta transportadora debe funcionar durante al menos 3 segundos antes de cambiar al siguiente paso.

**6. Aumentar contador**

Una vez que la pieza ha llegado al final de la cinta, el contador correspondiente debe incrementarse en uno.

Si ya hay 3 piezas en la cinta, puede comenzar el proceso de expulsión.

Si aún no se ha alcanzado el número, debe saltar de nuevo al paso 2.

**7. Divisor abierto**

Una vez que hay 3 piezas en la cinta transportadora, pueden empujarse a un palet. Para ello, abra primero el separador (Q6).

500 milisegundos después de que deje de accionarse la válvula, puede suponerse que el separador está completamente abierto. Para pasar al siguiente paso, también debe haber una paleta vacía.

**8. Empujar la pieza sobre el palet**

Si un palet está listo (B3 interrumpido) y el divisor está abierto, las piezas pueden empujarse sobre él.

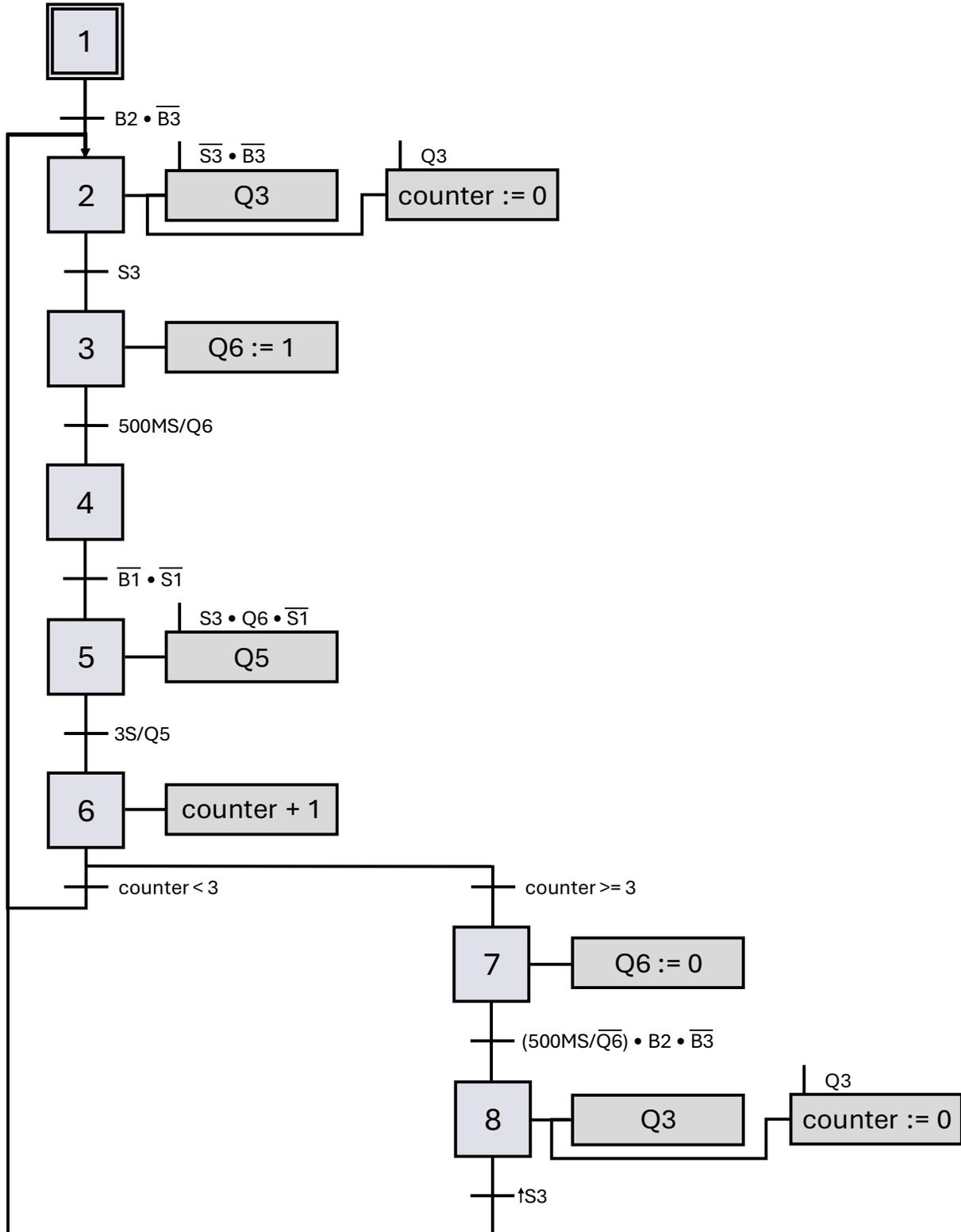
Si se activa el empujador, también debe ponerse a cero el contador de piezas en la cinta.

Si el deslizador alcanza de nuevo su posición inicial (borde positivo S3), puede volver al paso 2.



Solución

Solución:





## 10.2 Ejercicio: Planificación de la cadena de secuencias con GRAFCET - convertidor

### Objetivo:

Puedo utilizar la descripción funcional para crear una cadena de secuencia GRAFCET para el convertidor de la línea de producción.

### Tarea:

Crear una cadena secuencial en GRAFCET para el convertidor de la línea de producción de forma que la función se dé en base a la descripción del sistema.

### Función:

#### 1. Paso inicial

En el paso inicial, el convertidor se desplaza a la posición inicial. La posición inicial se define de forma que no se alcance ninguna de las dos posiciones finales.

Si la unidad de transferencia se encuentra en la posición final S1, debe desplazarse en el sentido de la placa giratoria (Q1) hasta que abandone esta posición.

Si la unidad de transferencia se encuentra en la posición final S2, debe desplazarse en la dirección de la cinta transportadora (Q2) hasta que haya abandonado esta posición.

Una vez que el convertidor ha alcanzado la posición inicial, puede pasar al siguiente paso.

#### 2. Esperando pieza

En este paso no se realiza ninguna acción. El sistema espera hasta que la mesa giratoria proporciona una pieza acabada que se puede recoger.

Si la plataforma giratoria está en posición (S4 accionado) y una pieza está lista para ser recogida (wstReady), el sistema pasa al siguiente paso.

#### 3. Mueva la unidad de transferencia hacia la plataforma giratoria

La unidad de transferencia se desplaza en la dirección de la plataforma giratoria activando Q1. La acción sólo se ejecuta mientras no se alcance la posición final (S2) y la unidad de giro se encuentre en posición (S4).

Si la unidad de transferencia se encuentra en el lateral del plato giratorio, puede pasar al siguiente paso.

#### 4. Encender la aspiradora

El vacío se conecta activando Q8. Si se conecta el vacío, debe restablecerse la información de que hay una pieza lista en la mesa giratoria (wstReady := 0).

500 milisegundos después del encendido, se puede suponer que se ha acumulado y, por tanto, se puede pasar al siguiente paso.

#### 5. Mueva el convertidor hacia la cinta

La unidad de transferencia se desplaza en la dirección de la plataforma giratoria activando Q2. La acción sólo se ejecuta mientras no se haya alcanzado la posición final (S1) y la posición de depósito en la cinta transportadora esté libre (B1) y la cinta transportadora no esté en marcha (Q5).

Una vez que el convertidor ha alcanzado la posición final, puede pasar a la etapa siguiente.

Planificar y aplicar procesos automáticos - Ejercicio: Planificación de la cadena de secuencias con GRAFCET - convertidor

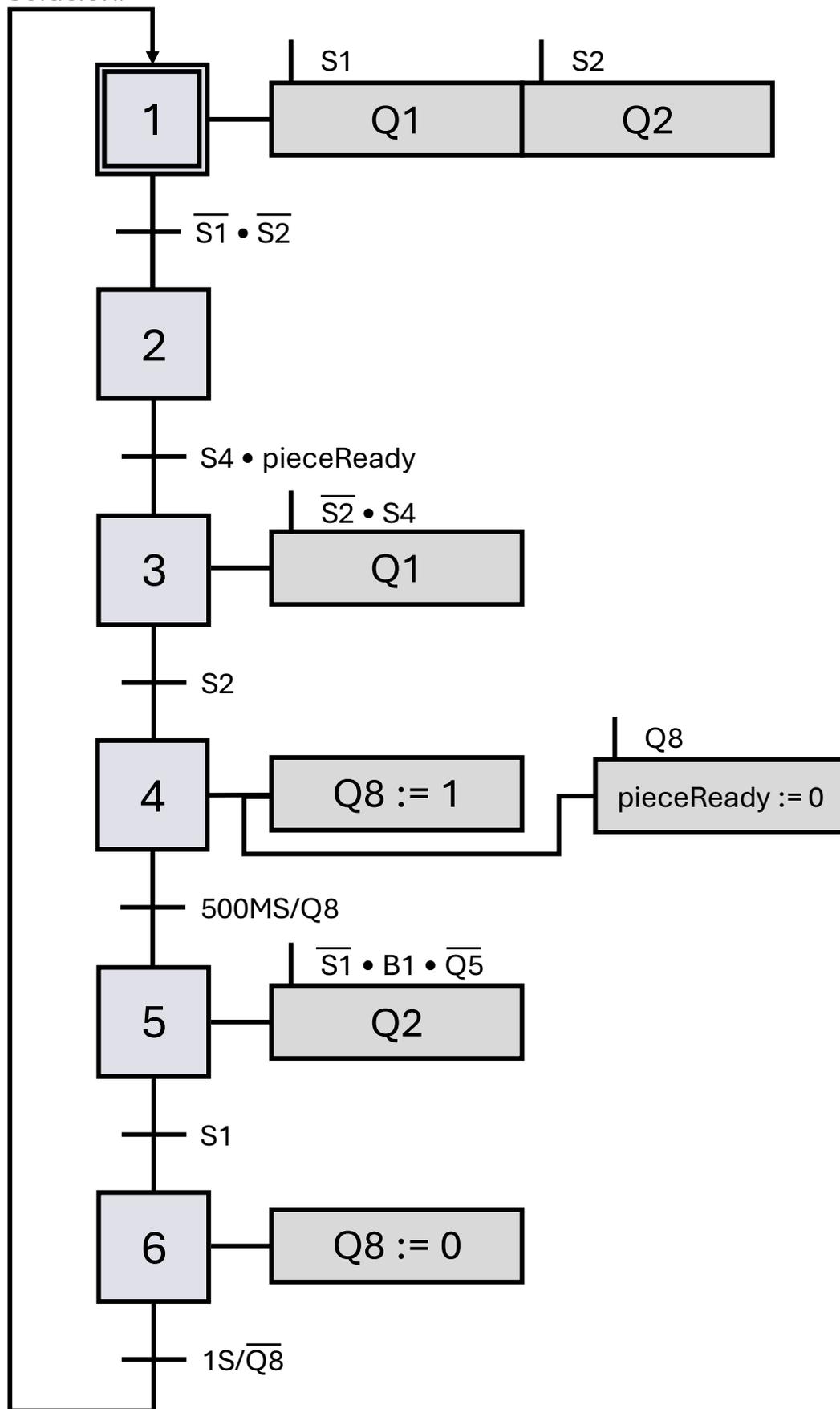
**6. Desconectar el vacío**

El vacío se desconecta restableciendo Q8. 1 segundo después de la desconexión, se puede suponer que el vacío también se ha disipado y, por lo tanto, se puede volver al paso inicial.



Solución

Solución:





## 10.3 Ejercicio: Planificación de la cadena de procesos con GRAFCET - Revista

### Objetivo:

Puedo crear una cadena de secuencias GRAFCET para el almacén de la mesa giratoria de la línea de producción utilizando la descripción funcional.

### Tarea:

Crear una cadena secuencial en GRAFCET para el almacén de la línea de producción de forma que la función se dé en base a la descripción del sistema.

### Función:

#### 1. Paso inicial

En el paso inicial no se realiza ninguna acción.

Si la plataforma giratoria está en posición (S4 activado), las piezas de trabajo están en el almacén (B4 interrumpido) y el espacio (nest) de la plataforma giratoria está vacío ("nestBelegt" = "FALSE"), puede pasar al siguiente paso.



Es aconsejable retrasar la señal de la barrera de luz B4 aprox. 1 segundo, ya que de lo contrario el empujador podría extenderse durante el llenado del almacén, lo que podría provocar el enclavamiento de las piezas.

#### 2. Extender la corredera

Al accionar Q7, se extiende la corredera y se empuja una pieza del almacén a la mesa giratoria. La corredera sólo puede extenderse cuando la plataforma giratoria está en posición (S6).

1 segundo después de activar la válvula, se puede suponer que la corredera está totalmente extendida. Puede saltar al siguiente paso.

#### 3. Establecer nido ocupado

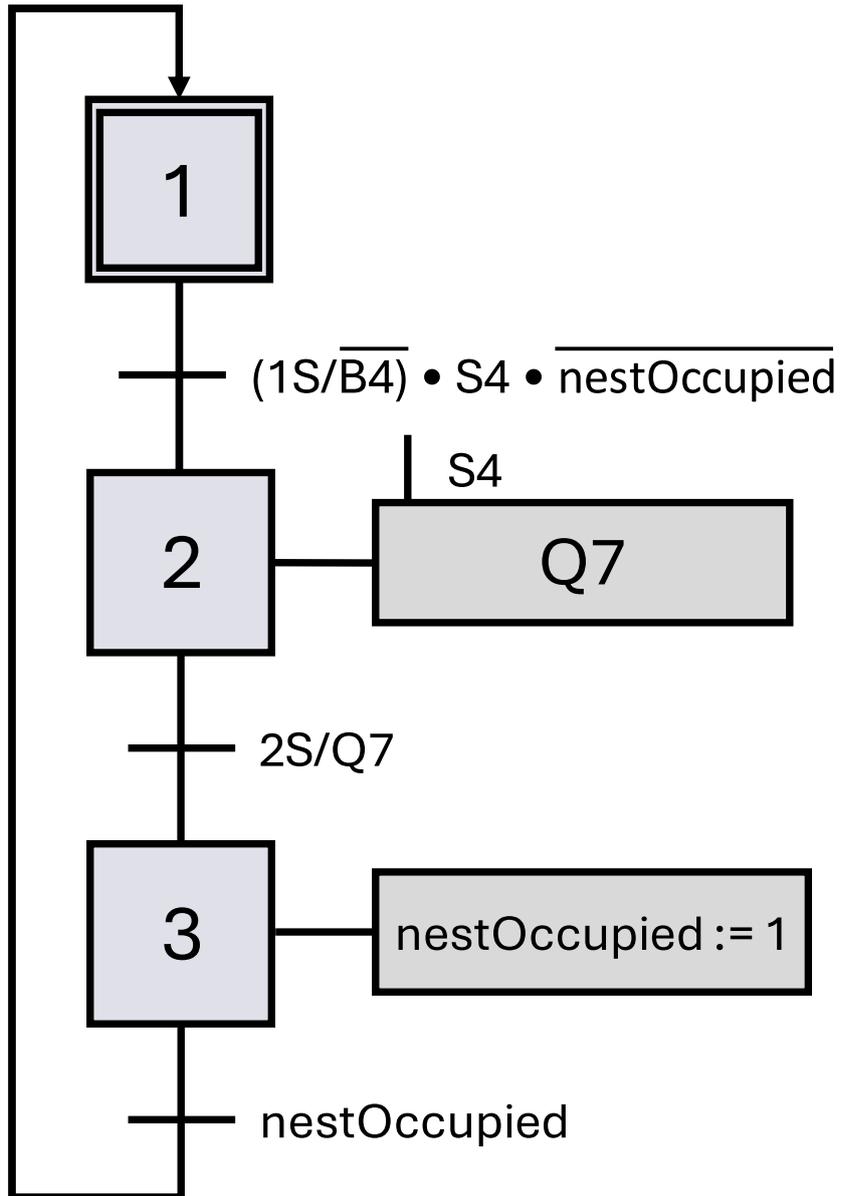
Como ahora hay una pieza en la mesa giratoria, el nido debe escribirse como ocupado. Para ello, el bit "nido ocupado" debe ponerse a "TRUE".

Si el bit "nestOccupied" tiene el valor "TRUE", puede saltar al paso inicial.



Solución

Solución:





## 10.4 Ejercicio: Planificación de la cadena de procesos con GRAFCET - Estación de procesamiento

### Objetivo:

Puedo crear una cadena de secuencias GRAFCET para una estación de mecanizado en la mesa giratoria de la línea de producción utilizando la descripción funcional.

### Tarea:

Cree una cadena de secuencia en GRAFCET para una estación de procesamiento de la línea de producción de forma que la función se dé en base a la descripción del sistema.

Tanto la estación "Taladrado" como la estación "Soldadura" tienen la misma secuencia. Por lo tanto, sólo es necesario crear una cadena de secuencias general que sea válida para ambas estaciones.

### Función:



El proceso se inicia cuando hay una pieza en bruto en la estación y se ejecuta durante un periodo de tiempo definido. En lugar de activar directamente la salida para taladrar (Q9) o soldar (Q10), se suele utilizar el término "mecanizado". No se asume un tiempo definido como transición, pero sí se asume generalmente "tiempo". Tras el paso de trabajo, se debe restablecer la variable "pieza bruta" y fijar la variable "pieza acabada".

#### 1. Paso inicial

En el paso inicial no se realiza ninguna acción.

Si la unidad de giro está en posición (S4 activado) y hay un espacio en blanco en la emisora, puede saltar al siguiente paso.

#### 2. Tratamiento

En este paso se procesa la pieza. La salida "proceso" debe estar activada.

Si la salida está activa durante el "tiempo" definido, puede saltar al siguiente paso.

#### 3. Fijar elemento prefabricado

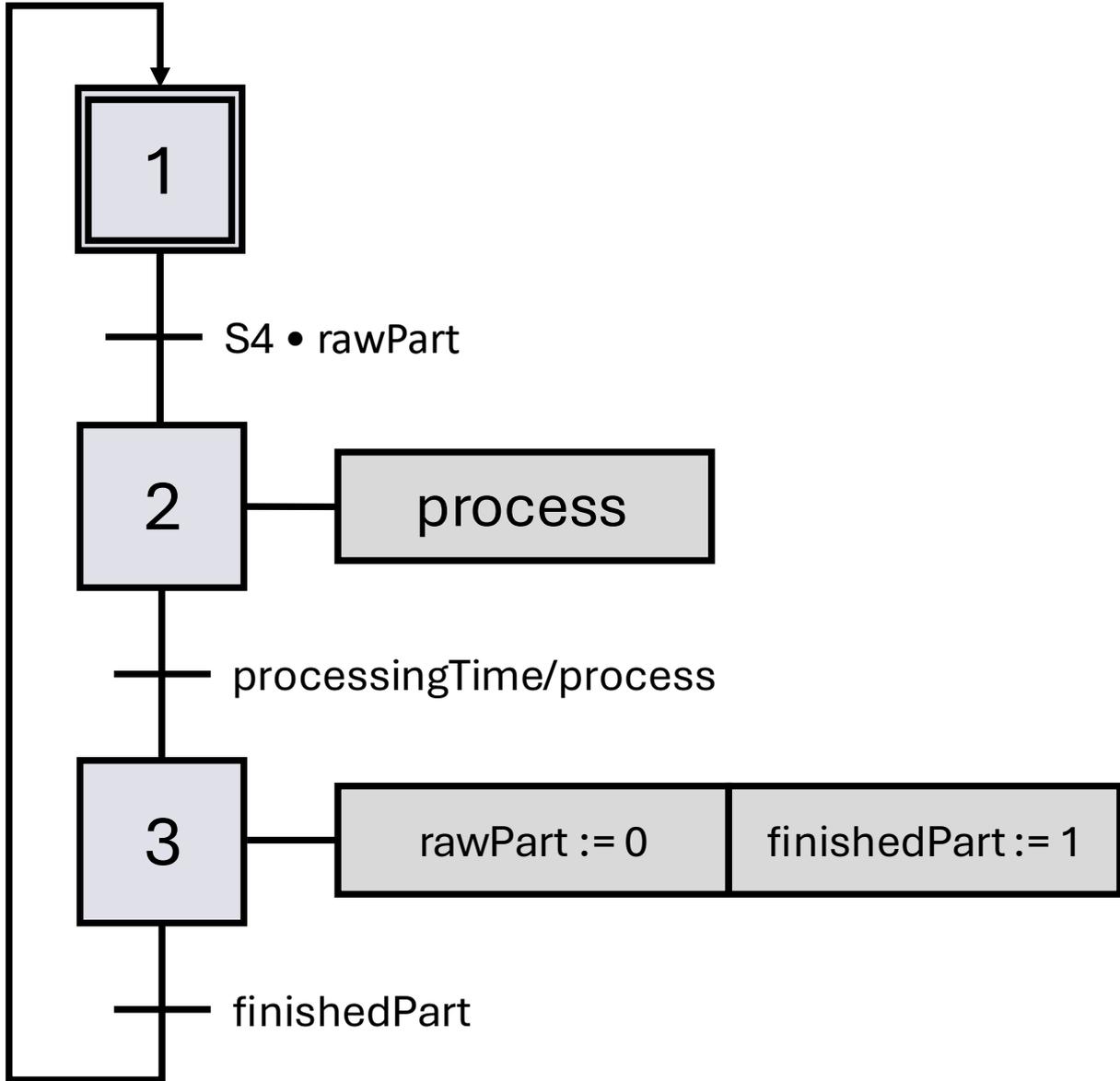
Hay que poner a cero la variable "Pieza sin mecanizar". La variable "Pieza acabada" debe estar ajustada.

Si "Parte terminada" está activada, puede volver al paso inicial.



Solución

Solución:





## 10.5 Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FBD] - Ruta de transporte

Objetivo:

Puedo crear el programa automático para la sección de transporte de la línea de producción basándome en la descripción funcional y en la cadena de secuencias creada en GRAFCET.

Tarea:

Cree el programa PLC para la secuencia automática de forma que la función se implemente basándose en la descripción del sistema.

La cadena se inicializa al encender la unidad de control.

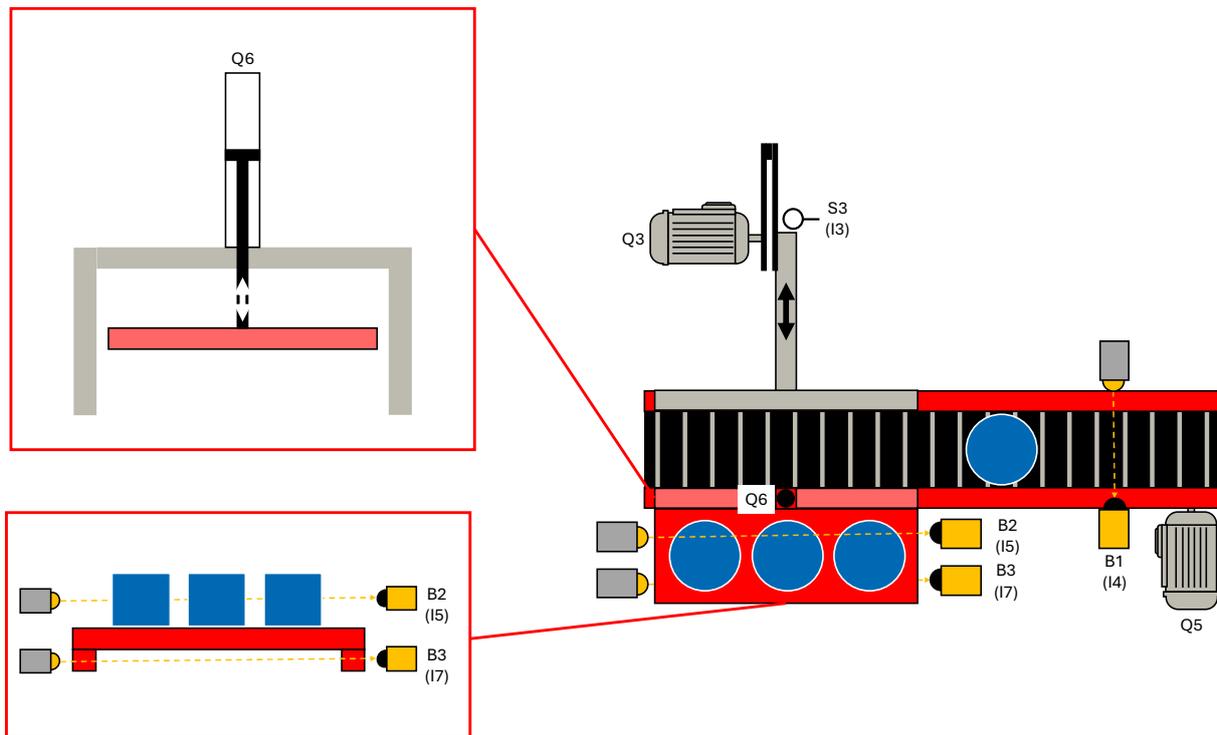
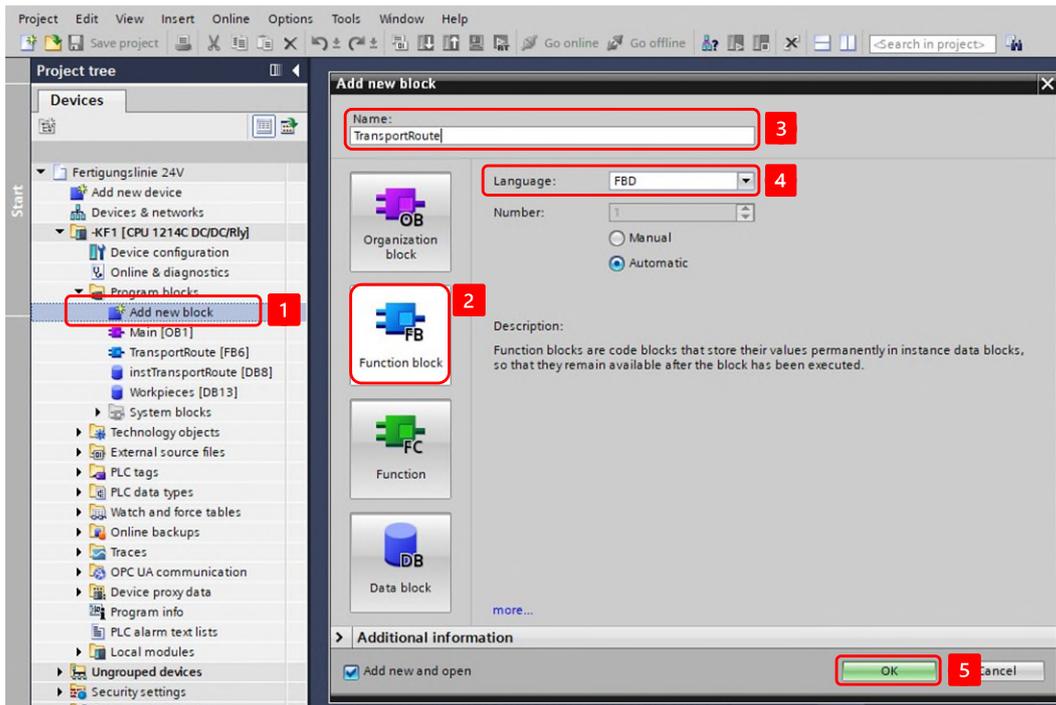


Imagen 1 Esquema del sistema - ruta de transporte

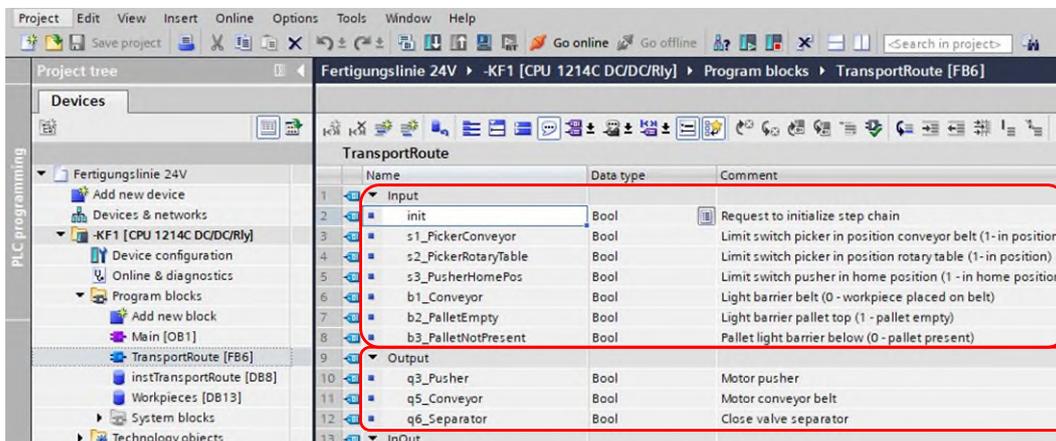
## Planificar y aplicar procesos automáticos - Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FBD] - Ruta de transporte

### Procedimiento:

1. Cree un nuevo bloque de funciones, seleccione el lenguaje de programación deseado y asígnele un nombre significativo:

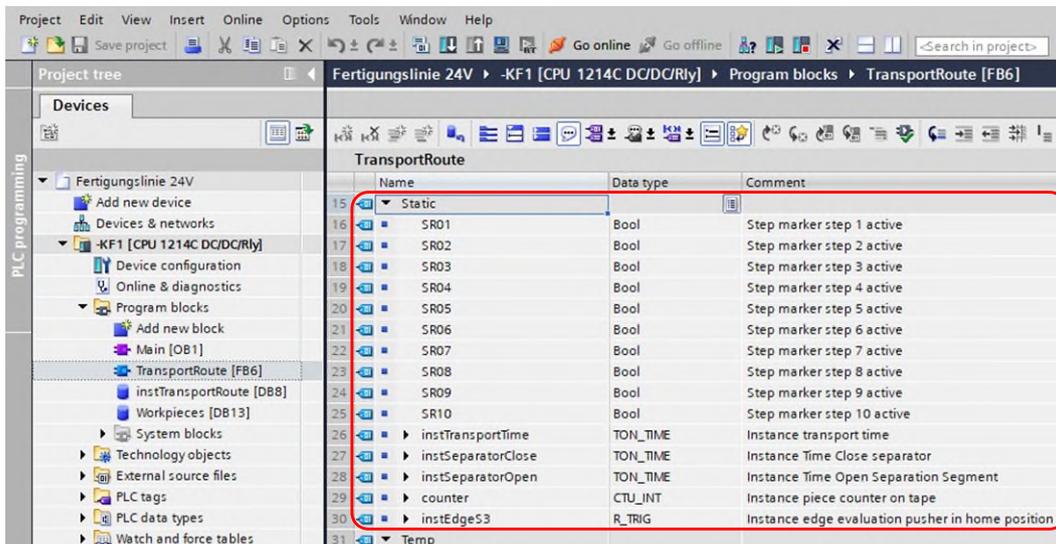


2. Declare variables para los sensores y actuadores, así como una variable para inicializar la cadena de pasos en la interfaz del bloque de funciones:

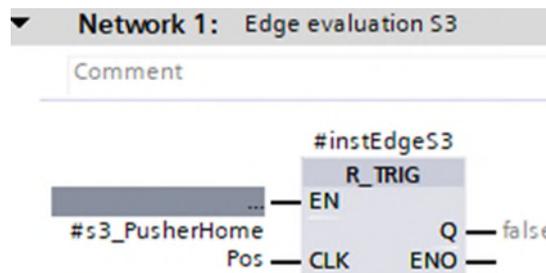


Planificar y aplicar procesos automáticos - Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FBD] - Ruta de transporte

3. Declare los indicadores de paso en el área estática de la interfaz del bloque de funciones, una instancia para la evaluación de aristas del interruptor de posición límite S3, instancias para los tiempos de retardo y una instancia para el contador de piezas:



4. Programe la evaluación de bordes para el interruptor de posición límite S3 en la primera red:



5. Implementar los pasos individuales en las siguientes redes utilizando flip-flops basados en el GRAFCET. Debe utilizarse una red nueva para cada paso:



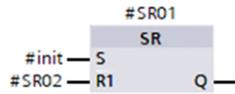
El paso 1 es el paso inicial. En la secuencia automática, el sistema salta del paso 6 al paso 2 si aún no se ha alcanzado el recuento. Si hay que expulsar piezas, el sistema salta de nuevo al paso 2 después del paso 8.

El paso 6 se reinicia con el paso siguiente, o con el paso 2 en caso de devolución.

Planificar y aplicar procesos automáticos - Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FBD] - Ruta de transporte

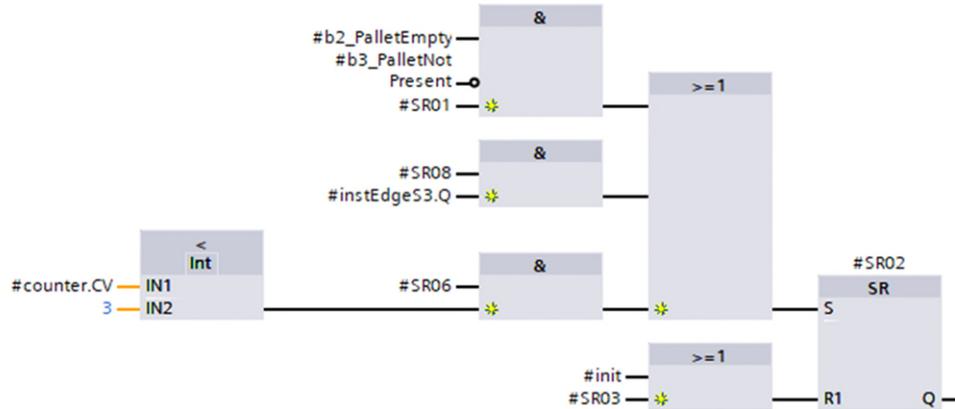
**Network 2: Step 1 - Initial step**

Comment



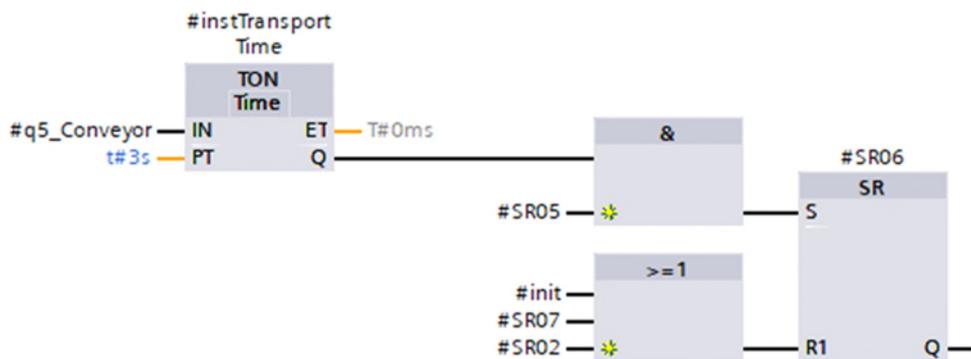
**Network 3: Step 2 - Move pusher to home position**

Comment



**Network 7: Step 6 - Increase counter**

Comment



**Network 8: Step 7 - Open the separator**

Comment

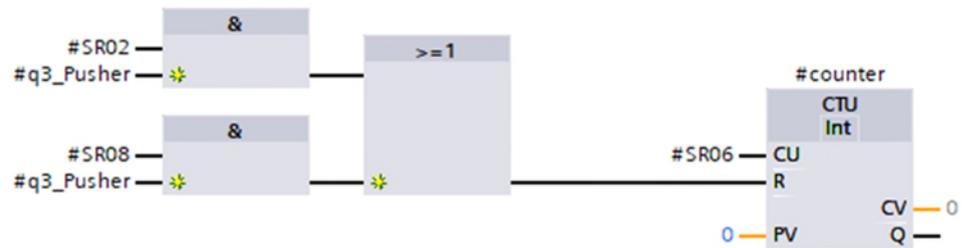


Planificar y aplicar procesos automáticos - Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FBD] - Ruta de transporte

6. Asigne las acciones debajo de la cadena de pasos en las redes siguientes:

**Network 10: counter**

Comment



**Network 11: Motor pusher**

Comment



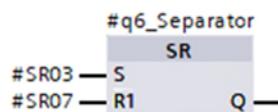
**Network 12: Motor conveyor belt**

Comment



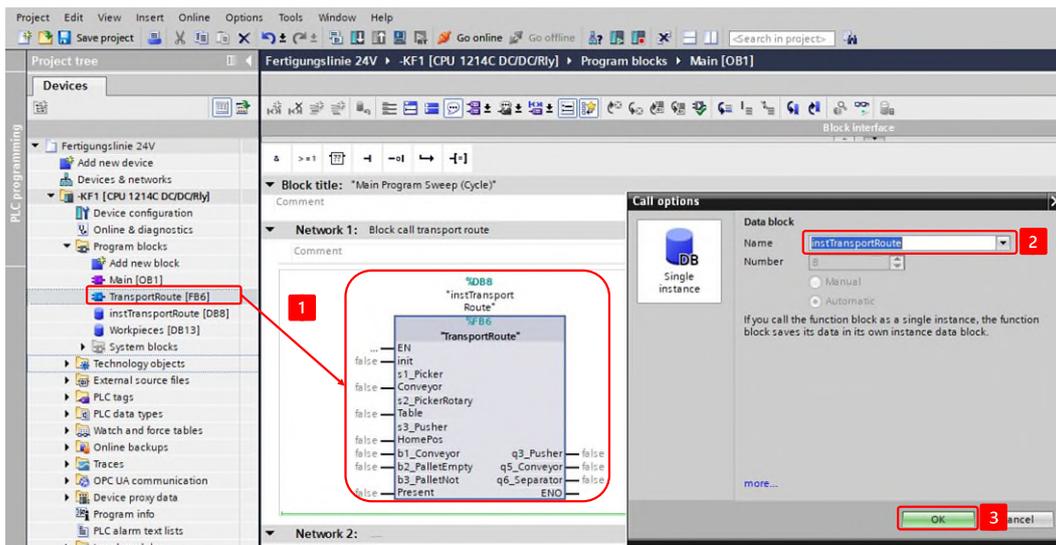
**Network 13: Close valve separator**

Comment

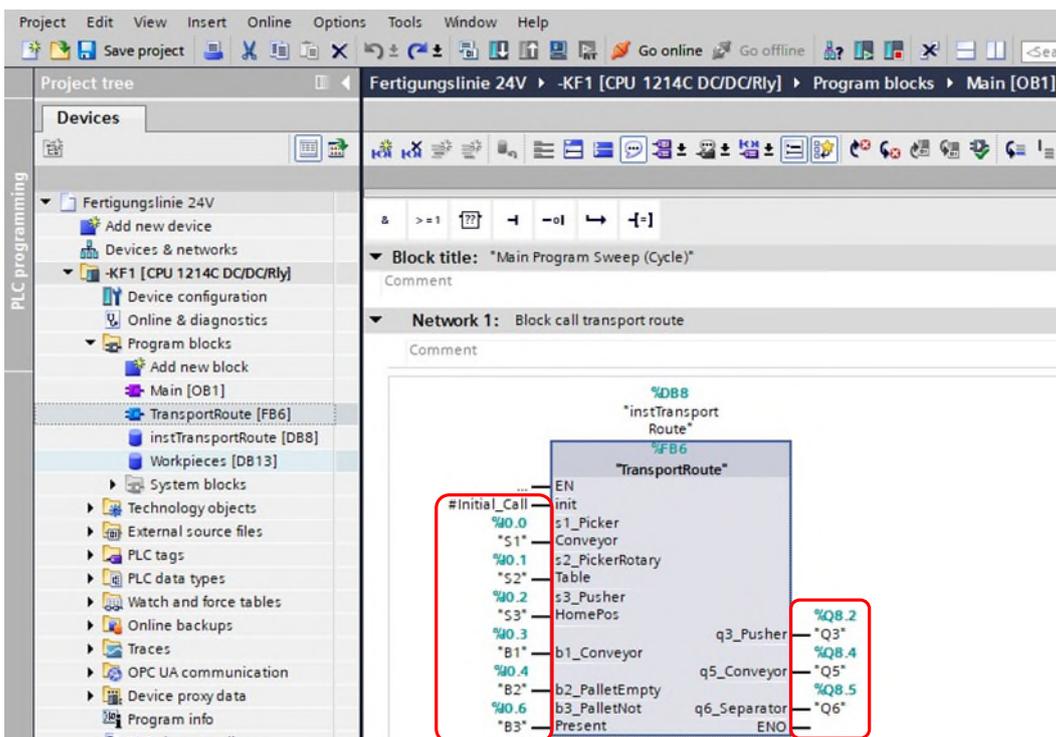


Planificar y aplicar procesos automáticos - Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FBD] - Ruta de transporte

7. Llama al módulo de funciones en "MAIN" y crea una instancia:



8. Conecte la interfaz del bloque de funciones con las variables de entrada y salida de su tabla de variables:



El bit de sistema "Initial\_Call" proporcionado por Siemens se utiliza como petición de inicialización. Es "TRUE" cuando la MAIN se ejecuta por primera vez.

9. Poner en servicio el sistema de forma estructurada con ayuda de un protocolo de puesta en servicio.

Planificar y aplicar procesos automáticos - Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FBD] - Ruta de transporte



Solución

Planificar y aplicar procesos automáticos - Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FBD] - Ruta de transporte

**Solución:**

La solución se encuentra en el proyecto del Portal TIA "Fertigungslinie\_02\_Transportstrecke\_FUP.zap17".



## 10.6 Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Ruta de transporte

Objetivo:

Puedo crear el programa automático para la sección de transporte de la línea de producción basándome en la descripción funcional y en la cadena de secuencias creada en GRAFCET.

Tarea:

Cree el programa PLC para la secuencia automática de forma que la función se implemente basándose en la descripción del sistema.

La cadena se inicializa al conectar la unidad de control.

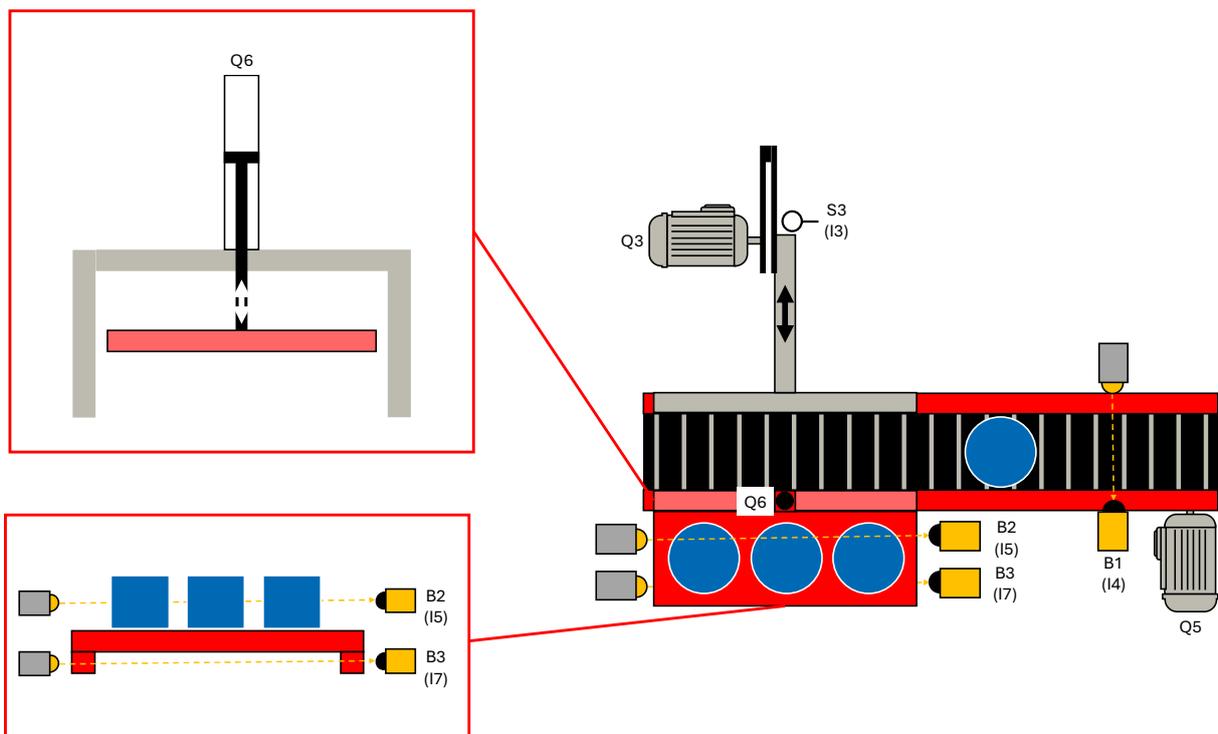
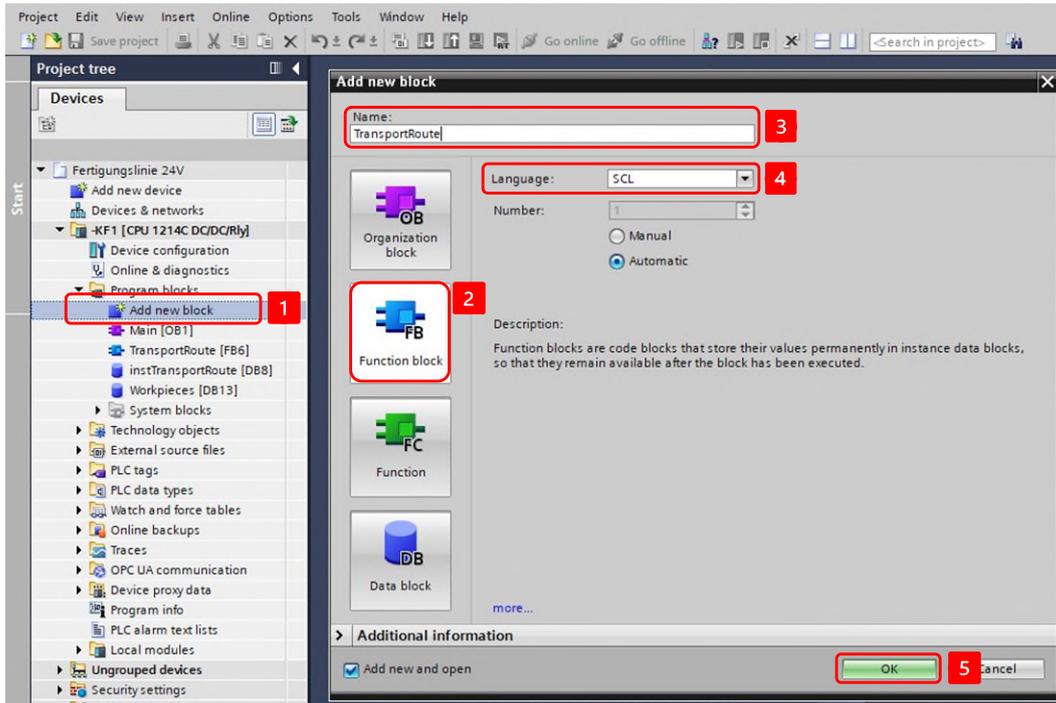


Imagen 2 Diagrama del sistema - ruta de transporte

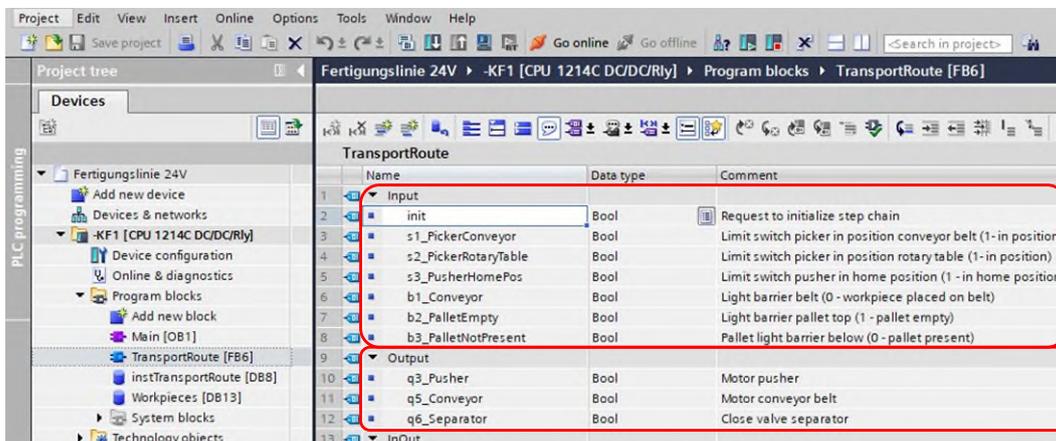
## Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Ruta de transporte

### Procedimiento:

1. Cree un nuevo bloque de funciones, seleccione el lenguaje de programación deseado y asígnele un nombre significativo:

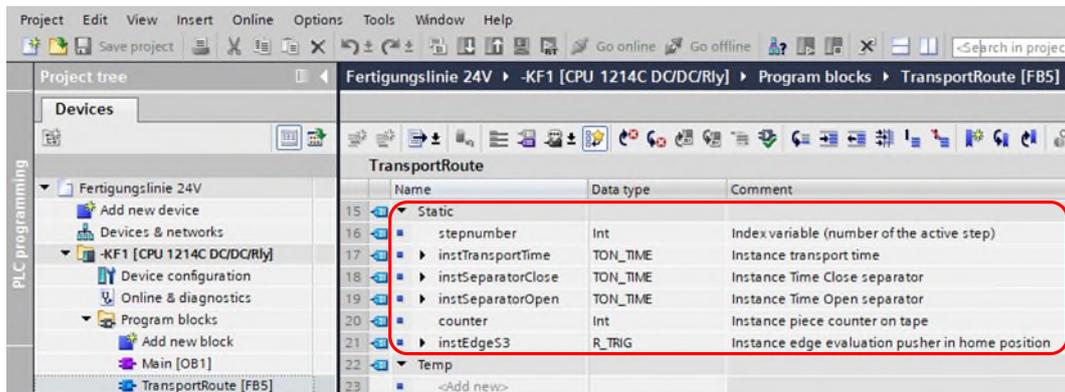


2. Declare variables for sensors and actuators, as well as a variable to initialize the step chain in the function block interface:



Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Ruta de transporte

3. Declare la variable de índice en el área estática de la interfaz del bloque de funciones, una instancia para la evaluación del borde del interruptor de posición límite S3, instancias para los tiempos de retardo y una variable para el contador de las piezas:



4. Programe la inicialización de la cadena:

```

1 // initialize step chain
2 IF #init THEN
3     #stepnumber := 1; // initialize step chain
4 END_IF;

```

5. Programe la evaluación de bordes para el interruptor de posición final S3:

```

6 // edge evaluation pusher in home position
7 #instEdgeS3(CLK := #s3_PusherHomePos);

```

6. Implemente los pasos individuales del GRAFCET en la siguiente estructura CASE. Debe crearse un nuevo CASO en la estructura para cada paso, que representa el número de paso:

**i** El paso 1 es el paso inicial. En la secuencia automática, el sistema salta del paso 6 al paso 2 si aún no se ha alcanzado el recuento. Si hay que expulsar piezas, el sistema salta de nuevo al paso 2 después del paso 8.

**i** Todas las acciones se restablecen en el paso inicial. De este modo se garantiza que no quede ninguna acción activada si se cancela una cadena de pasos activa mediante una solicitud de inicialización.

Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Ruta de transporte

```

9 // step chain
10 CASE #stepnumber OF
11     1: // step 1 - Initstep
12         // Reset all actions
13         #q3_Pusher := FALSE;
14         #q5_Conveyor := FALSE;
15         #q6_Separator := FALSE;
16
17         // transitions
18     IF NOT #b3_PalletNotPresent           // pallet inserted
19         AND #b2_PalletEmpty               // pallet empty
20     THEN
21         #stepnumber := 2;                 // next step
22     END_IF;
23
24     2: // step2 - move Pusher to home position
25         // actions
26         #q3_Pusher := NOT #s3_PusherHomePos // move pusher to home position
27         AND NOT #b3_PalletNotPresent;      // pallet inserted
28
29     IF #q3_Pusher THEN                     // if pusher is activated
30         #counter := 0;                     // reset counter
31     END_IF;
32
33         // transitions
34     IF #s3_PusherHomePos THEN              // pusher is in home position
35         #stepnumber := 3;                 // next step
36     END_IF;
37
38     6: // step 6 - count up
39         // actions
40         #counter := #counter + 1;          // count up workpiece
41
42     IF #counter < 3 THEN                    // if counter is smaller than 3
43         #stepnumber := 2;                 // jump to 2
44     ELSE
45         #stepnumber := 7;                 // Start discharge
46     END_IF;
47
48     7: // step 7 - open separator
49         // actions
50         #q6_Separator := FALSE;           // open separator
51
52         // transitions
53     IF #instSeparatorOpen.Q                // separator opened
54         AND #b2_PalletEmpty                // pallet is empty
55         AND NOT #b3_PalletNotPresent        // palette is present
56     THEN
57         #stepnumber := 8;                 // next step
58     END_IF;
59
60     8: // step 8 - drive pusher
61         // actions
62         #q3_Pusher := TRUE;                // drive pusher
63
64     IF #q3_Pusher THEN                      // if pusher driven
65         #counter := 0;                     // reset counter
66     END_IF;
67
68         // transitions
69     IF #instEdgeS3.Q THEN                  // pusher back in home position arrived
70         #stepnumber := 2;                 // jump to 2
71     END_IF;
72 END_CASE;

```

Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Ruta de transporte

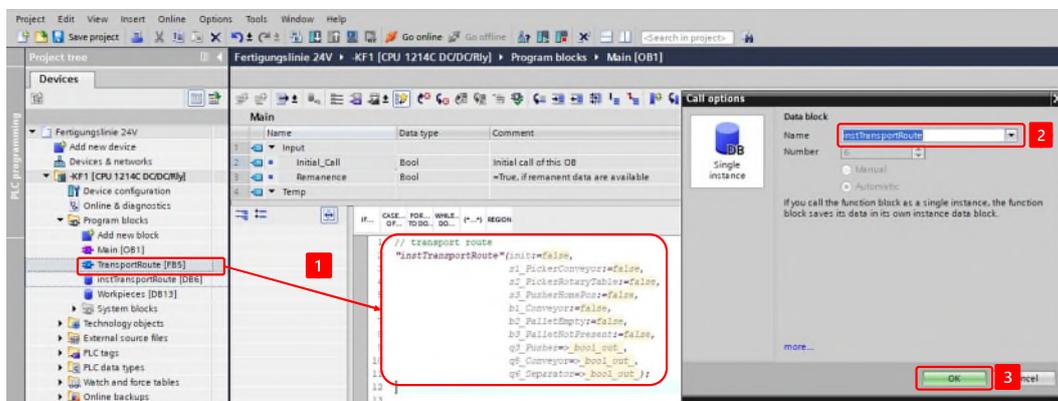
7. Programe las funciones horarias según la estructura CASE:

```

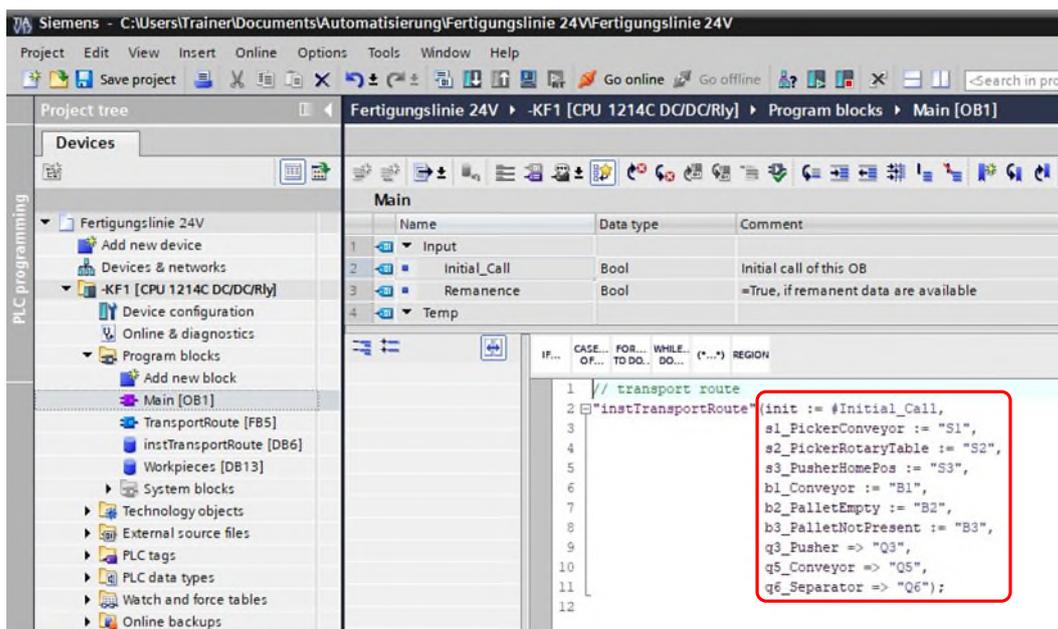
106 // timer
107 // transport time
108 #instTransportTime(IN := #q5_Conveyor,
109 | PT := t#3s);
110
111 // Time Close separator
112 #instSeparatorClose(IN := #q6_Separator,
113 | PT := t#500ms);
114 // Time Open separator
115 #instSeparatorOpen(IN := NOT #q6_Separator,
116 | PT := t#500ms);

```

8. Llama al módulo de funciones en "MAIN" y crea una instancia:



9. Conecte la interfaz del bloque de funciones con las variables de entrada y salida de su tabla de variables:



Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Ruta de transporte



El bit de sistema "Initial\_Call" proporcionado por Siemens se utiliza como petición de inicialización. Es "TRUE" cuando la MAIN se ejecuta por primera vez.

10. Poner en servicio el sistema de forma estructurada con ayuda de un protocolo de puesta en servicio.



Solución

Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Ruta de transporte

**Solución:**

La solución se encuentra en el proyecto del Portal TIA  
"Fertigungslinie\_02\_Transportstrecke\_SCL.zap17".



## 10.7 Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FBD] - Convertidor

Objetivo:

Puedo crear el programa automático para el convertidor de la línea de producción basándome en la descripción de la función y en la cadena de secuencias creada en GRAFCET.

Tarea:

Cree el programa PLC para la secuencia automática de modo que la función se realice basándose en la descripción del sistema.

La cadena se inicializa al conectar la unidad de control.

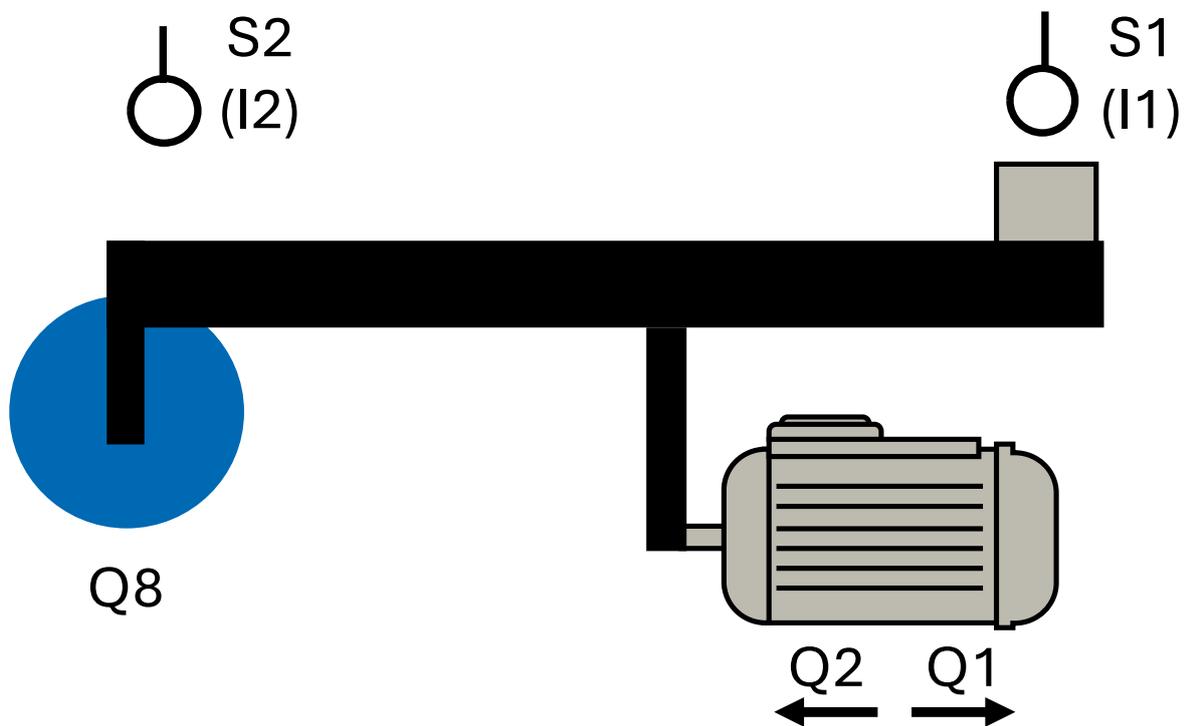
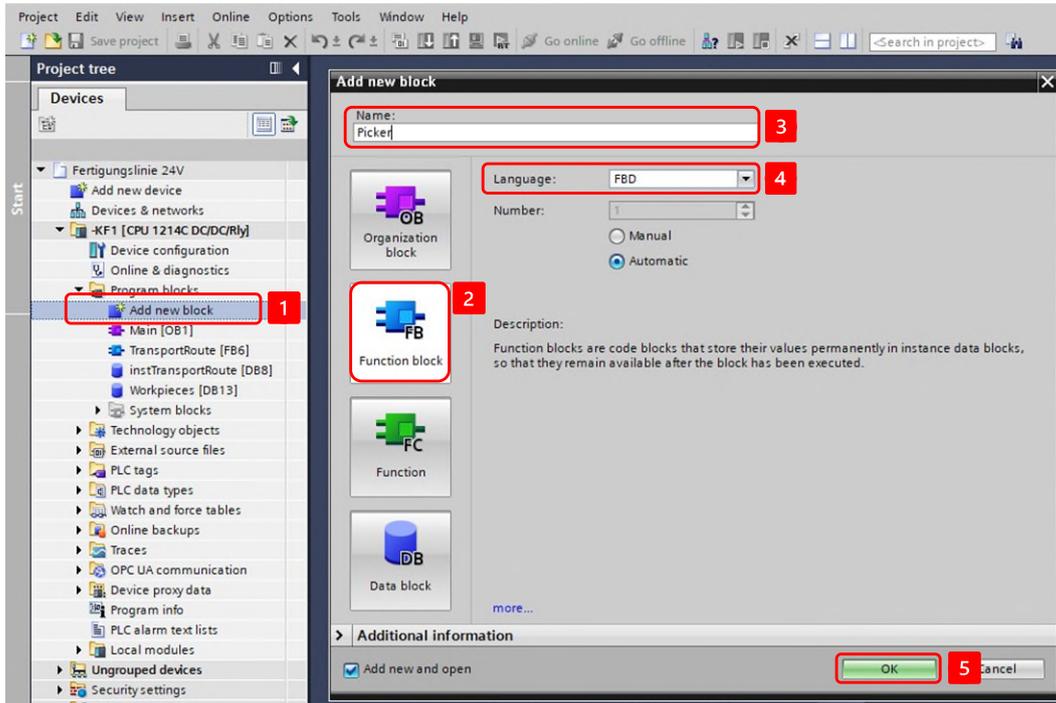


Imagen 3 Diagrama del sistema - convertidor

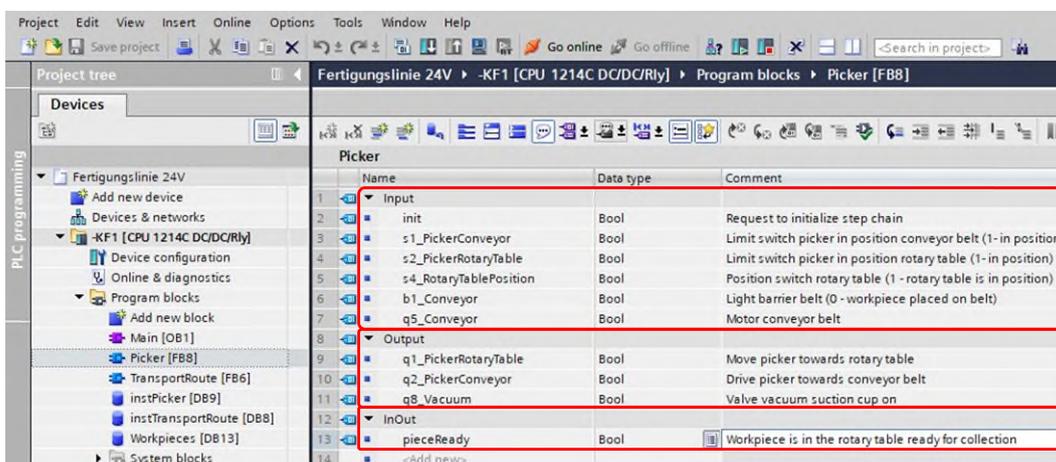
Planificar y aplicar procesos automáticos - Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FBD] - Convertidor

Procedimiento:

1. Cree un nuevo bloque de funciones, seleccione el lenguaje de programación deseado y asígnele un nombre significativo:

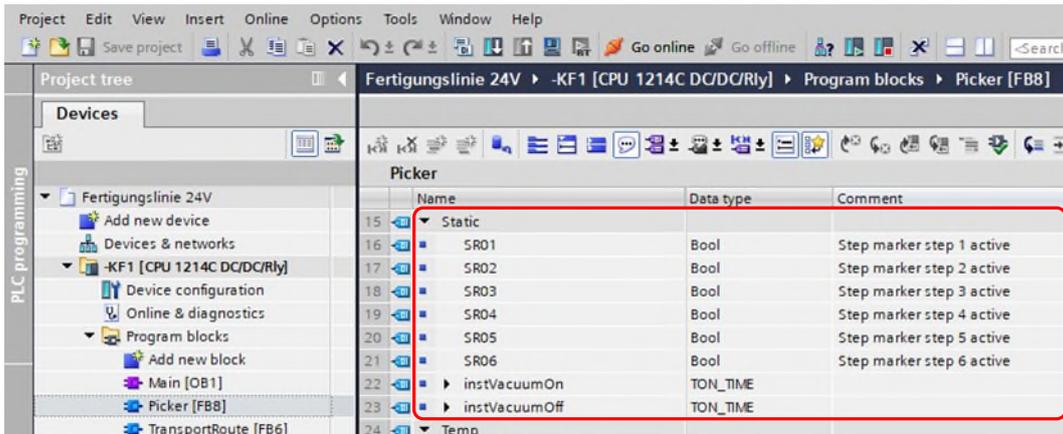


2. Declare variables para los sensores y actuadores, una variable para inicializar la cadena de pasos y una variable para transferir la información de la pieza de trabajo en la interfaz del bloque de funciones:

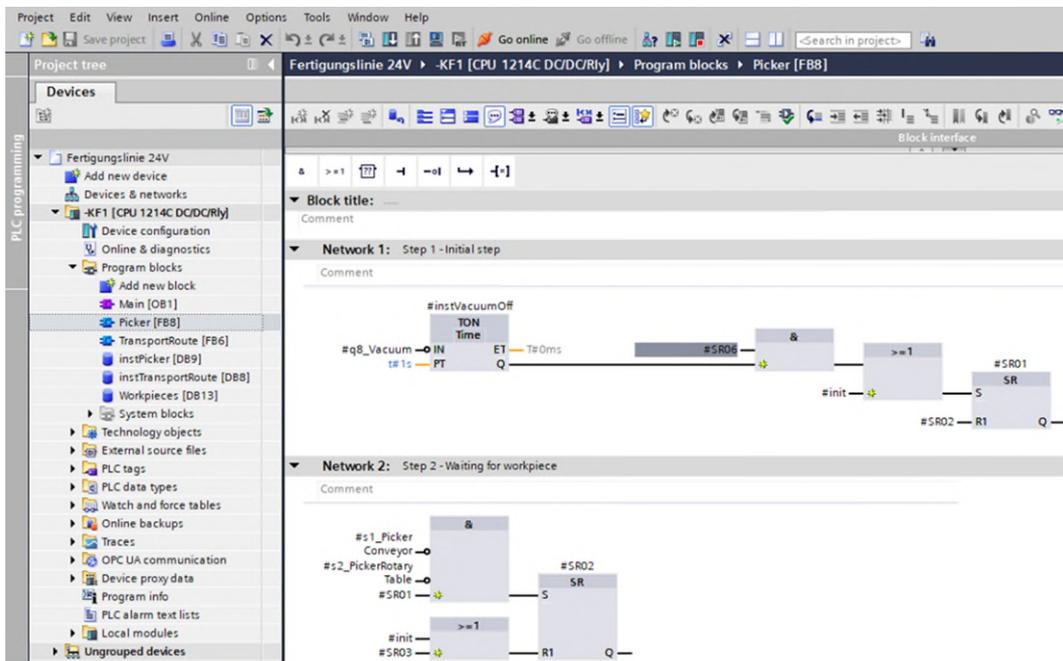


Planificar y aplicar procesos automáticos - Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FBD] - Convertidor

3. Declare los indicadores de paso en el área estática de la interfaz del bloque de funciones, así como dos instancias para el retardo de la señal de vacío:



4. Implementar los pasos individuales utilizando flip-flops basados en el GRAFCET. Debe utilizarse una red nueva para cada paso:



Planificar y aplicar procesos automáticos - Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FBD] - Convertidor

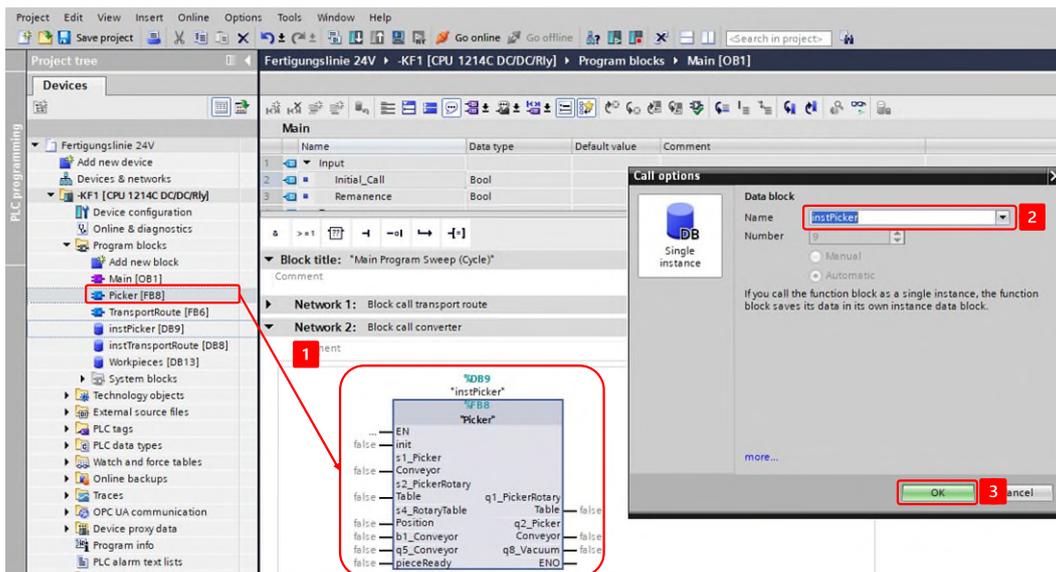
5. Asigne las acciones debajo de la cadena de pasos en las redes siguientes.

The screenshot displays the SIMATIC Manager interface for a PLC program. The left pane shows the project tree for 'Fertigungslinie 24V' with the 'Picker [FB8]' block selected. The main workspace shows three ladder logic networks:

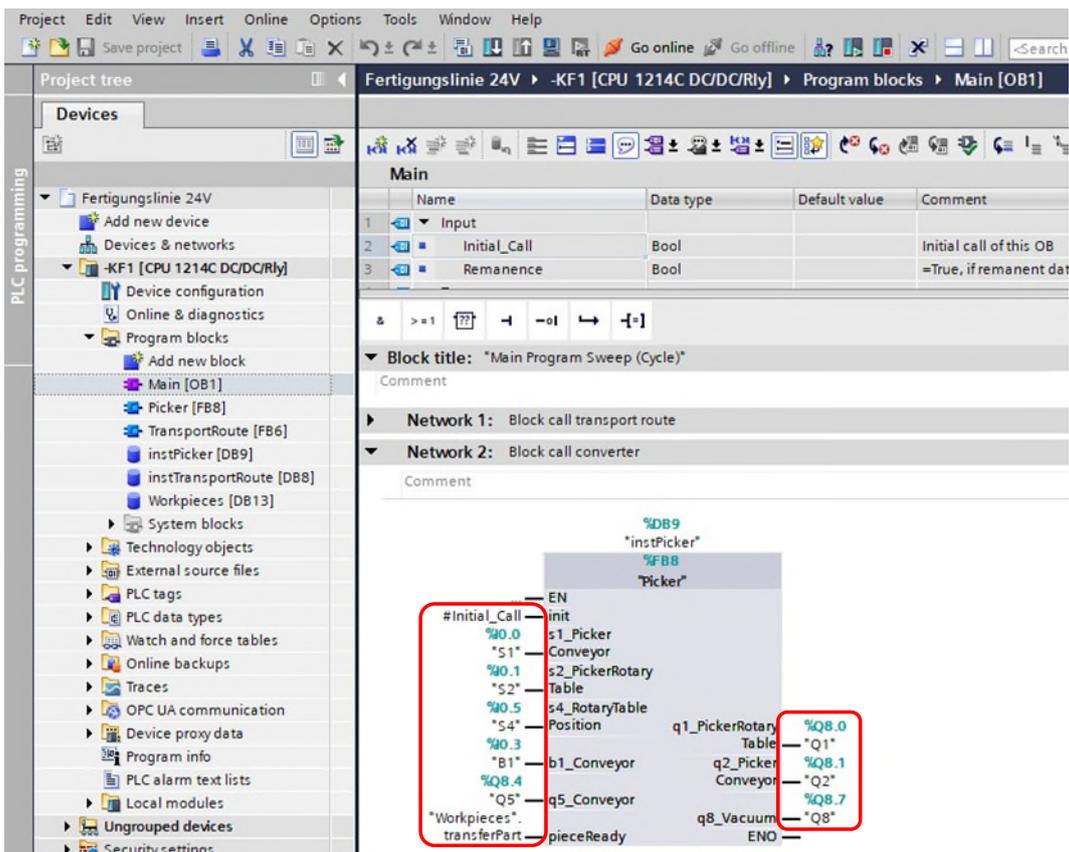
- Network 7: Workpiece is in the rotary table ready for collection**  
Ladder logic:  $\#q8\_Vacuum \ \& \ \#SR04 \rightarrow \#pieceReady \ R$
- Network 8: Move picker towards rotary table**  
Ladder logic:  $\#s1\_Picker \ Conveyor \ \& \ \#s2\_PickerRotary \ Table \ \& \ \#s4\_RotaryTable \ Position \rightarrow \geq 1 \rightarrow \#q1\_PickerRotary \ Table \ =$
- Network 9: Drive picker towards conveyor belt**  
Ladder logic:  $\#s2\_PickerRotary \ Table \ \& \ \#q5\_Conveyor \ \rightarrow \geq 1 \rightarrow \#s1\_Picker \ Conveyor \ \& \ \#b1\_Conveyor \ \& \ \#q5\_Conveyor \rightarrow \#q2\_Picker \ Conveyor \ =$

Planificar y aplicar procesos automáticos - Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FBD] - Convertidor

6. Llama al módulo de funciones en "MAIN" y crea una instancia:



7. Conecte la interfaz del bloque de funciones con las variables de entrada y salida de su tabla de variables:



El bit de sistema "Initial\_Call", que Siemens proporciona, se utiliza como petición de inicialización. Es "TRUE" cuando la MAIN se ejecuta por primera vez.

Planificar y aplicar procesos automáticos - Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FBD] - Convertidor

8. Poner en servicio el sistema de forma estructurada con ayuda de un protocolo de puesta en servicio.



Para que la unidad de transferencia recoja una pieza de la mesa giratoria, puede ajustarse manualmente el estado correspondiente en la gestión de piezas.

Fertigungslinie 24V ▶ -KF1 [CPU 1214C DC/DC/Rly] ▶ Program blocks ▶ Workpieces [DB13]

Keep actual values Snapshot Copy snapshots to start values Load start values as actual

	Name	Data type	Monitor value	Comment
1	Static			
2	magazinePart	Bool	FALSE	Nest in position Magazine is occupied with a workpiece
3	drillingRawPart	Bool	FALSE	Nest in drilling position is occupied by a non-drilled workpiece
4	drillingFinishedPart	Bool	FALSE	Nest in drilling position is occupied by a drilled workpiece
5	weldingRawPart	Bool	FALSE	Nest in welding position is occupied by a non-welded workpiece
6	weldingFinishedPart	Bool	FALSE	Nest in welding position is occupied by a non-welded workpiece
7	transferPart	Bool	TRUE	Nest in position transfer is occupied by a finished part



Solución

Planificar y aplicar procesos automáticos - Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FBD] - Convertidor

### Solución:

La solución se encuentra en el proyecto del Portal TIA "Fertigungslinie\_03\_Umsetzer\_FUP.zap17".



## 10.8 Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Conversor

Objetivo:

Puedo crear el programa automático para el convertidor de la línea de producción basándome en la descripción de la función y en la cadena de secuencias creada en GRAFCET.

Tarea:

Cree el programa PLC para la secuencia automática de modo que la función se realice basándose en la descripción del sistema.

La cadena se inicializa al conectar la unidad de control.

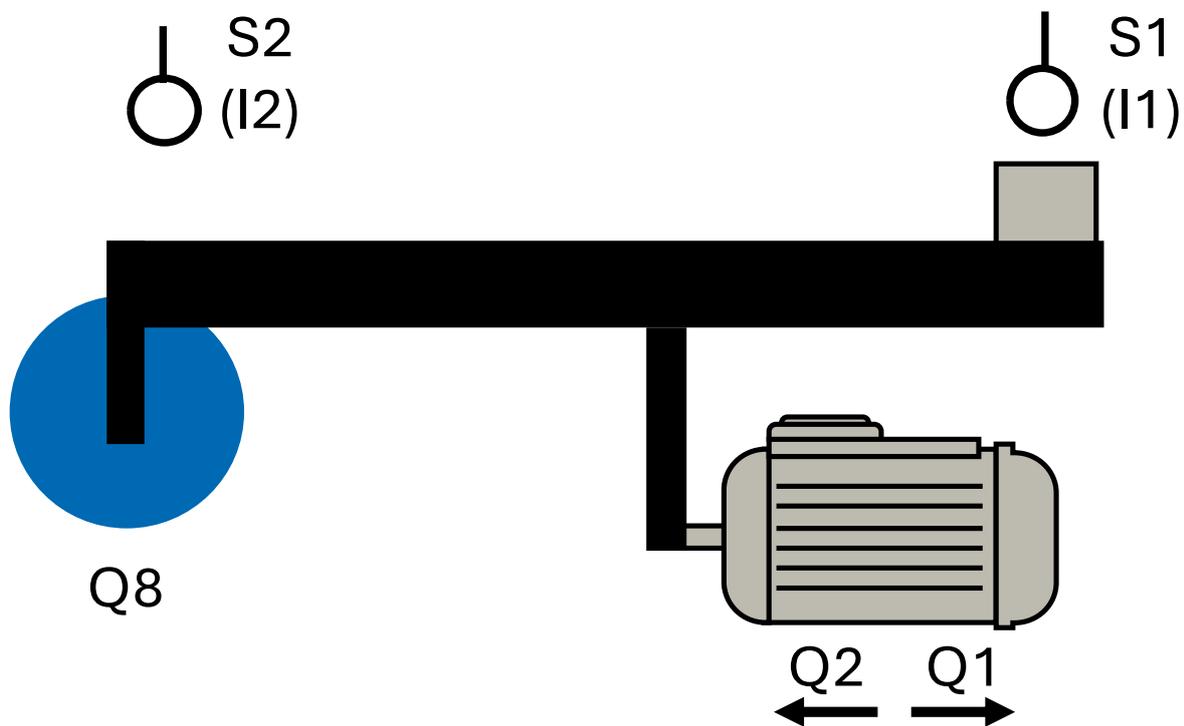
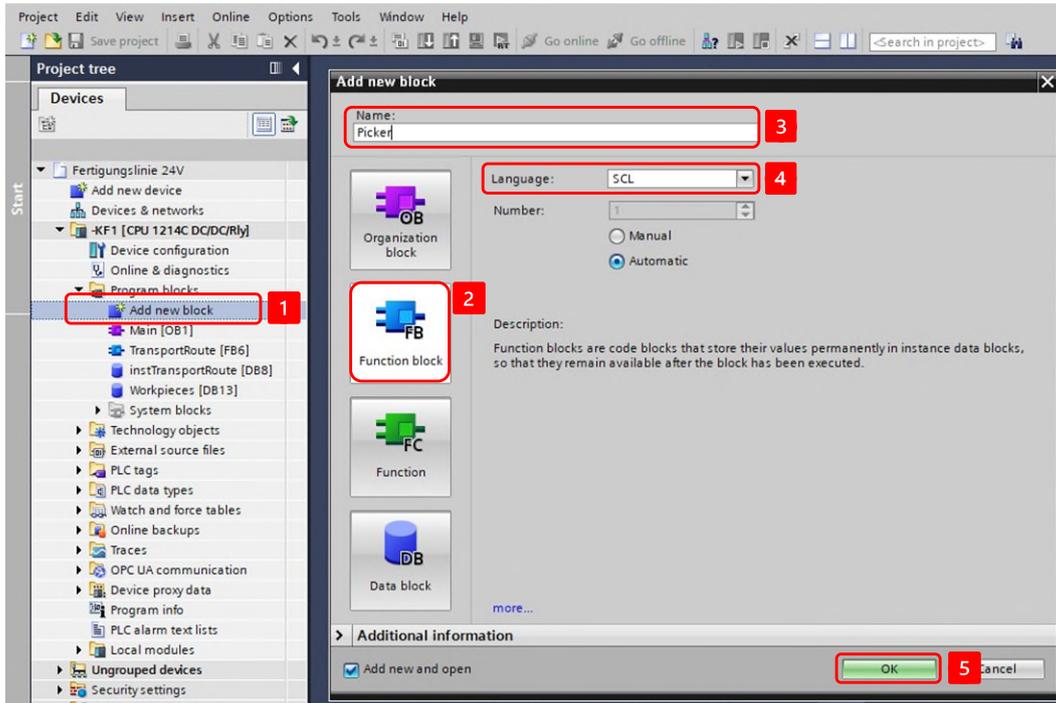


Imagen 4 Esquema del sistema - convertidor

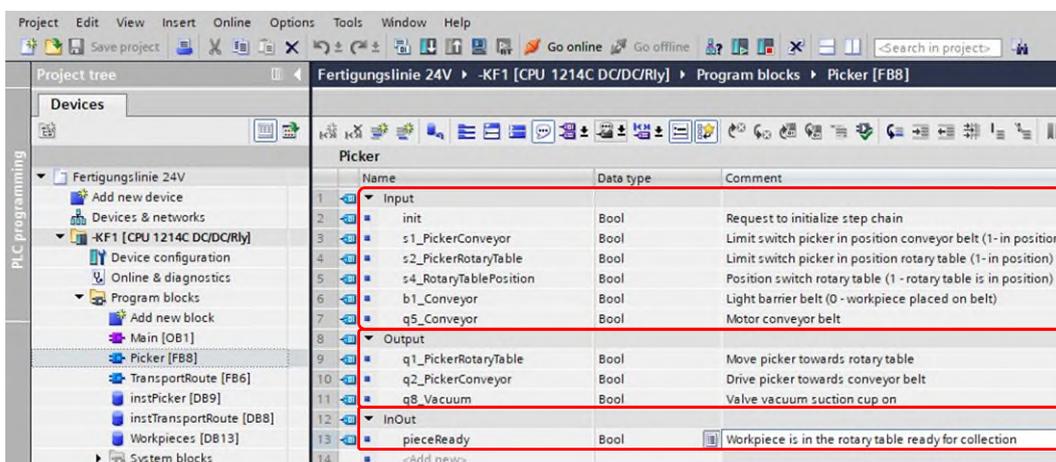
Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Conversor

Procedimiento:

1. Cree un nuevo bloque de funciones, seleccione el lenguaje de programación deseado y asígnele un nombre significativo:

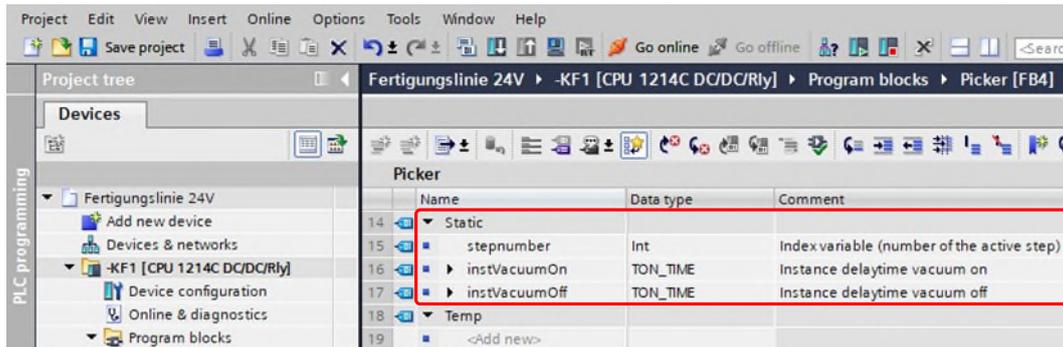


2. Declarar variables para los sensores y actuadores, una variable para inicializar la cadena de pasos y una variable para transferir la información de la pieza de trabajo en la interfaz del bloque de funciones:



Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Conversor

3. Declare la variable de índice en el área estática de la interfaz del bloque de funciones, así como dos instancias para el retardo de la señal de vacío:



4. Programe la inicialización de la cadena:

```

1 // initialize step chain
2 IF #init THEN
3     #stepnumber := 1; // initialize step chain
4 END_IF;

```

5. Implemente los pasos individuales del GRAFCET en la siguiente estructura CASE. Debe crearse un nuevo CASO en la estructura para cada paso, que representa el número de paso:



Todas las acciones se restablecen en el paso inicial. De este modo se garantiza que no quede ninguna acción activada si se cancela una cadena de pasos activa mediante una solicitud de inicialización.

```

7 // step chain
8 CASE #stepnumber OF
9     1: // step 1 - init step
10         // actions
11         #q1_PickerRotaryTable := #s1_PickerConveyor;
12         #q2_PickerConveyor := #s2_PickerRotaryTable;
13
14         #q8_Vacuum := FALSE; // reset action
15
16         // transitions
17 IF NOT #s1_PickerConveyor
18     AND NOT #s2_PickerRotaryTable
19 THEN
20     #stepnumber := 2;
21 END_IF;
22
23     2: // step 2 - waiting for workpiece
24         // actions
25
26         // transitions
27 IF #s4_RotaryTablePosition
28     AND #pieceReady
29 THEN
30     #stepnumber := 3;
31 END_IF;

```

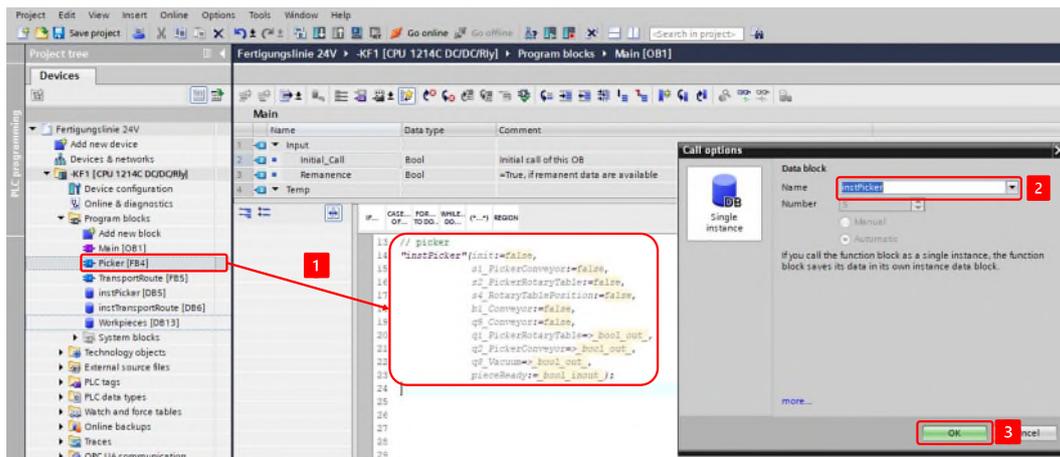
Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Conversor

6. Programe las funciones horarias según la estructura CASE:

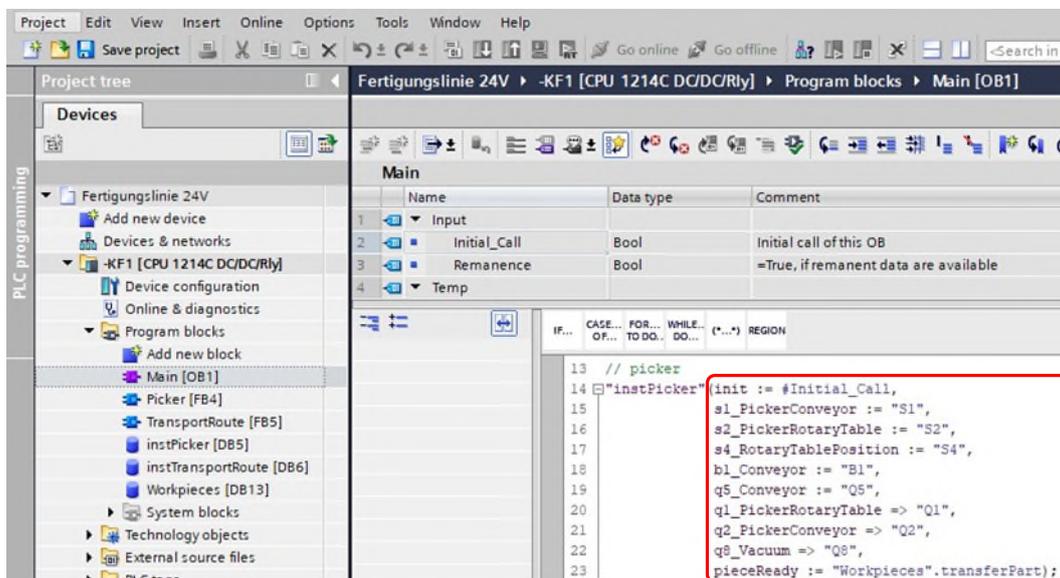
```

78 // delaytimes
79 // delaytime vacuum on
80 #instVacuumOn(IN := #q8_Vacuum,           // vacuum on
81             PT := t#500ms);
82
83 // delaytime vacuum off
84 #instVacuumOff(IN := NOT #q8_Vacuum,     // vacuum off
85              PT := t#1s);
    
```

7. Llama al módulo de funciones en "MAIN" y crea una instancia:



8. Conecte la interfaz del bloque de funciones con las variables de entrada y salida de su tabla de variables:



El bit de sistema "Initial\_Call", que Siemens proporciona, se utiliza como petición de inicialización. Es "TRUE" cuando la MAIN se ejecuta por primera vez.

Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Conversor

9. Poner en servicio el sistema de forma estructurada con ayuda de un protocolo de puesta en servicio.



Para que la unidad de transferencia recoja una pieza de la mesa giratoria, puede ajustarse manualmente el estado correspondiente en la gestión de piezas.

Fertigungslinie 24V ▶ -KF1 [CPU 1214C DC/DC/Rly] ▶ Program blocks ▶ Workpieces [DB13]

Keep actual values Snapshot Copy snapshots to start values Load start values as actual

	Name	Data type	Monitor value	Comment
1	Static			
2	magazinePart	Bool	FALSE	Nest in position Magazine is occupied with a workpiece
3	drillingRawPart	Bool	FALSE	Nest in drilling position is occupied by a non-drilled workpiece
4	drillingFinishedPart	Bool	FALSE	Nest in drilling position is occupied by a drilled workpiece
5	weldingRawPart	Bool	FALSE	Nest in welding position is occupied by a non-welded workpiece
6	weldingFinishedPart	Bool	FALSE	Nest in welding position is occupied by a non-welded workpiece
7	transferPart	Bool	TRUE	Nest in position transfer is occupied by a finished part



Solución

Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Conversor

### Solución:

La solución se encuentra en el proyecto del Portal TIA "Fertigungslinie\_03\_Umsetzer\_SCL.zip17".



## 10.9 Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FUP] - Revista

Objetivo:

Puedo crear el programa automático para el almacén de la línea de producción utilizando la descripción funcional y la cadena de secuencias creada en GRAFCET.

Tarea:

Cree el programa PLC para la secuencia automática de modo que la función se realice basándose en la descripción del sistema.

La cadena se inicializa al encender la unidad de control.

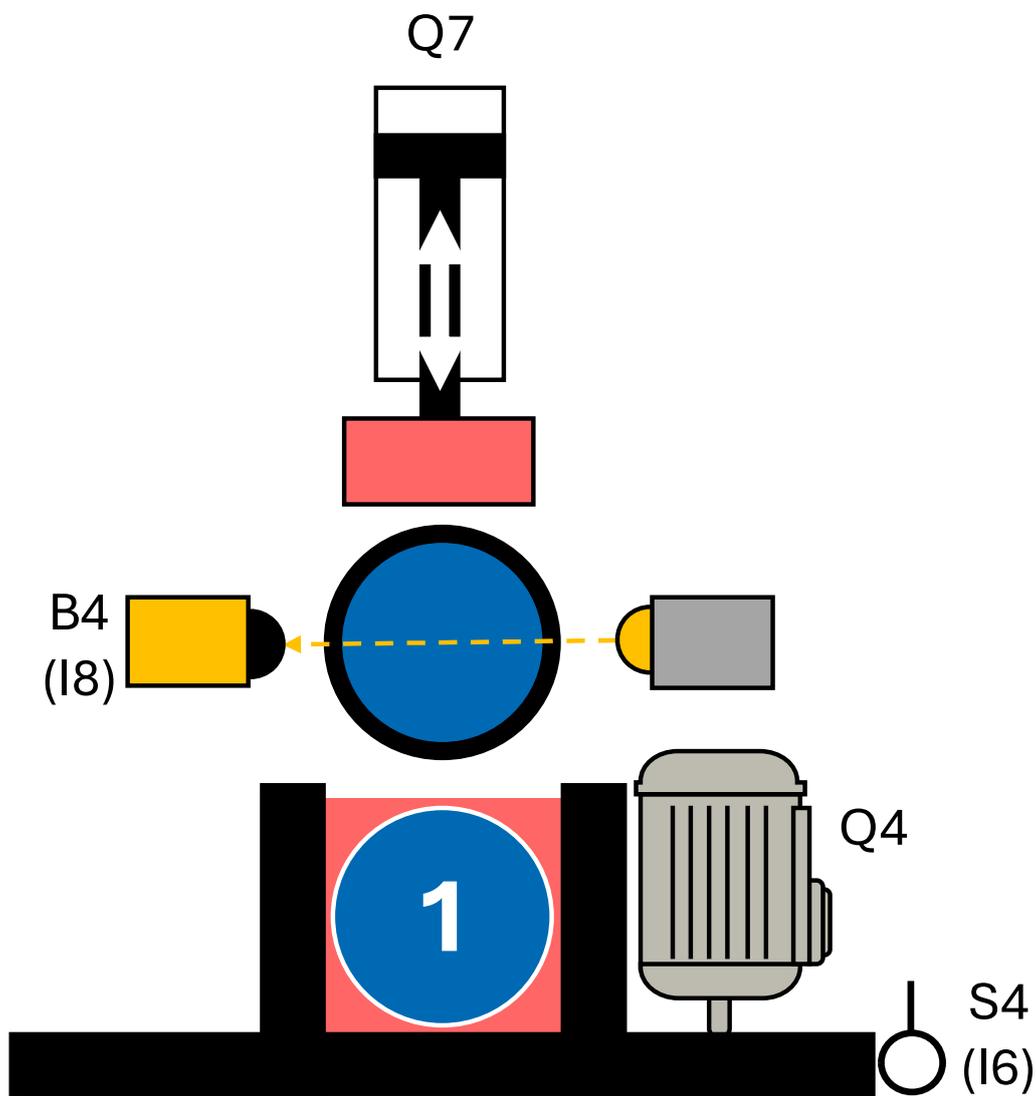
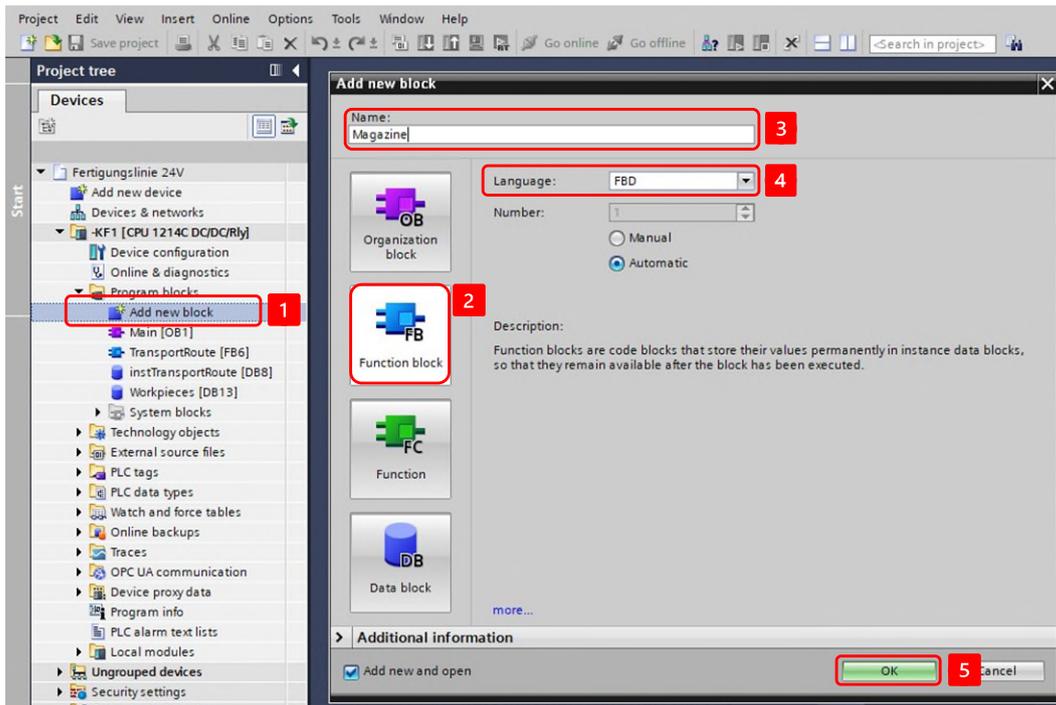


Imagen 5 Esquema del sistema - Revista

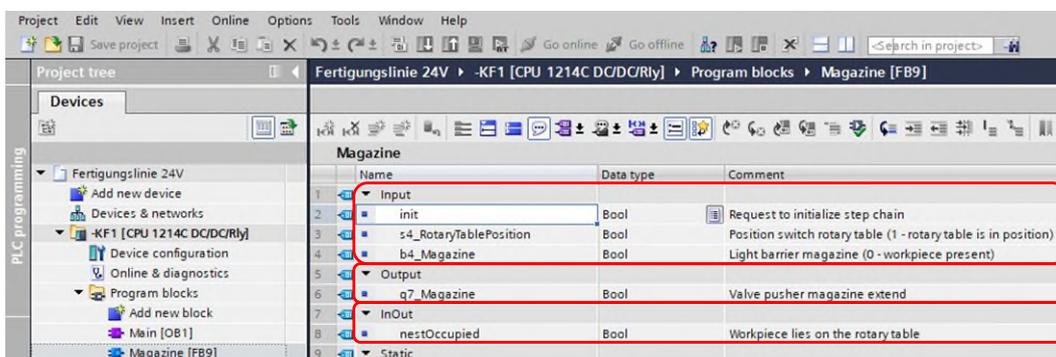
## Planificar y aplicar procesos automáticos - Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FUP] - Revista

### Procedimiento:

1. Cree un nuevo bloque de funciones, seleccione el lenguaje de programación deseado y asígnele un nombre significativo:

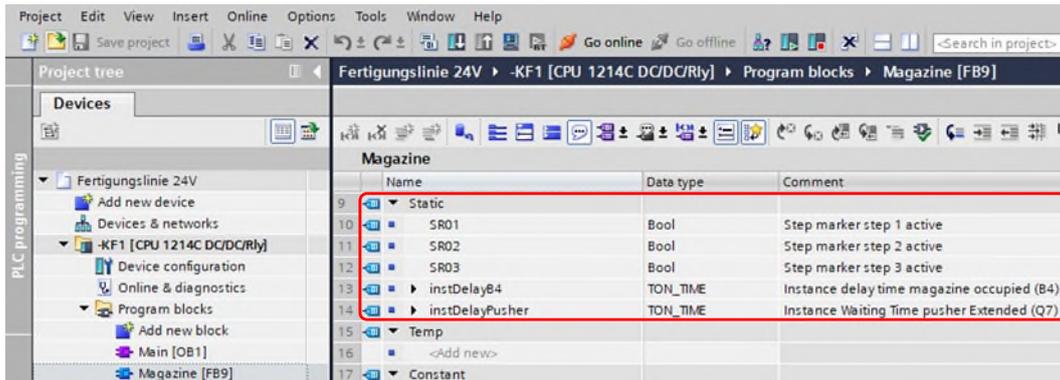


2. Declarar variables para los sensores y actuadores, una variable para inicializar la cadena de pasos y una variable para transferir la información de la pieza de trabajo en la interfaz del bloque de funciones:

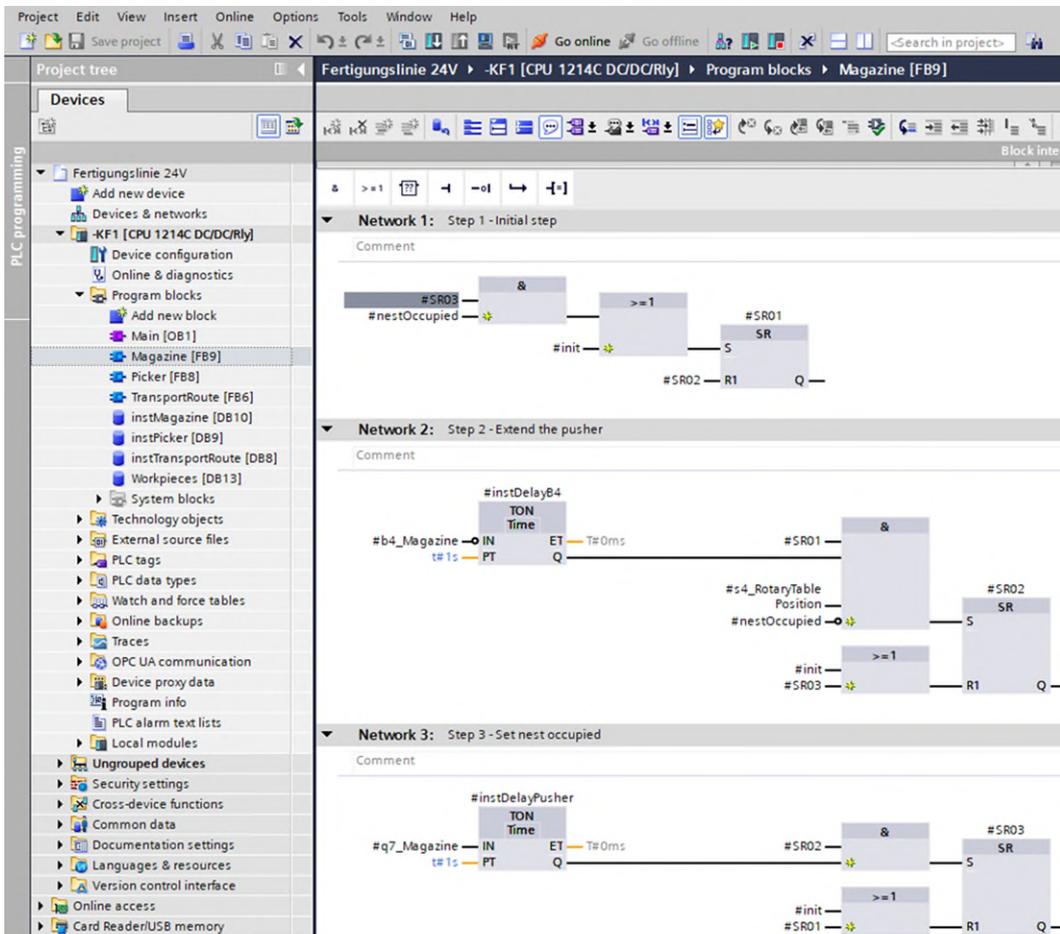


Planificar y aplicar procesos automáticos - Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FUP] - Revista

3. Declare los indicadores de paso en el área estática de la interfaz del bloque de funciones, así como dos instancias para los tiempos de espera:

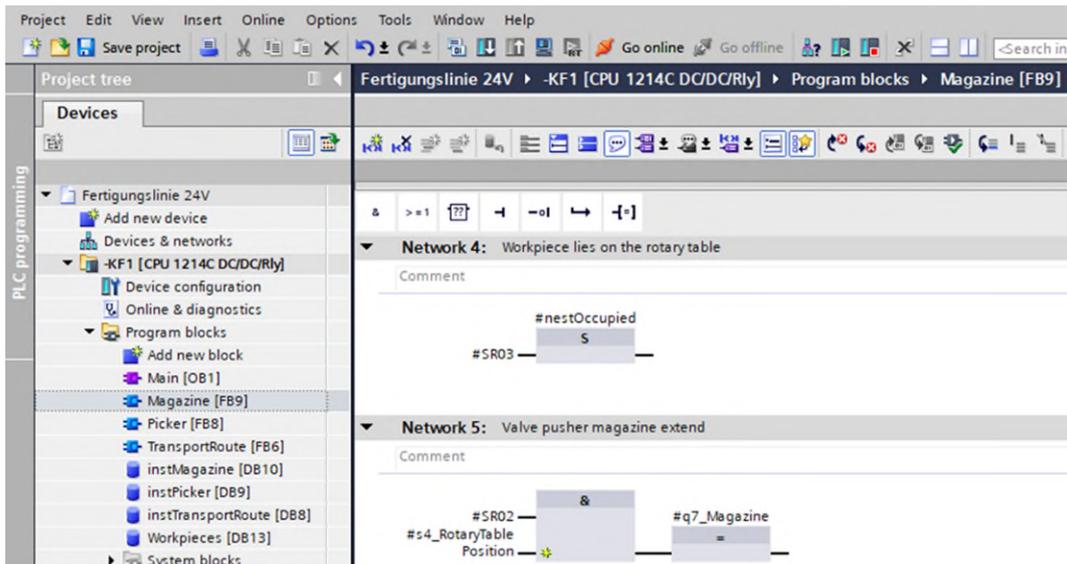


4. Implementar los pasos individuales utilizando flip-flops basados en el GRAFCET. Debe utilizarse una red nueva para cada paso:

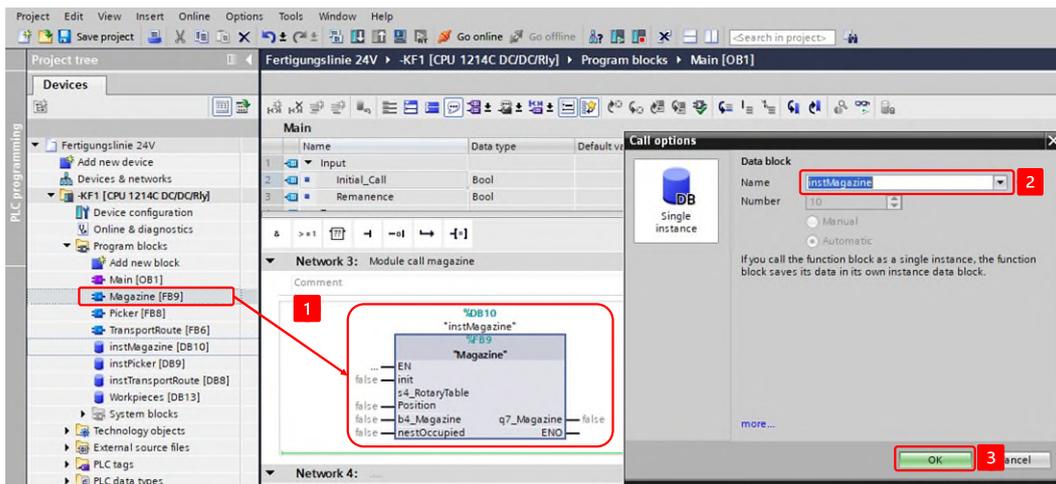


Planificar y aplicar procesos automáticos - Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FUP] - Revista

5. Asigne las acciones debajo de la cadena de pasos en las redes siguientes:

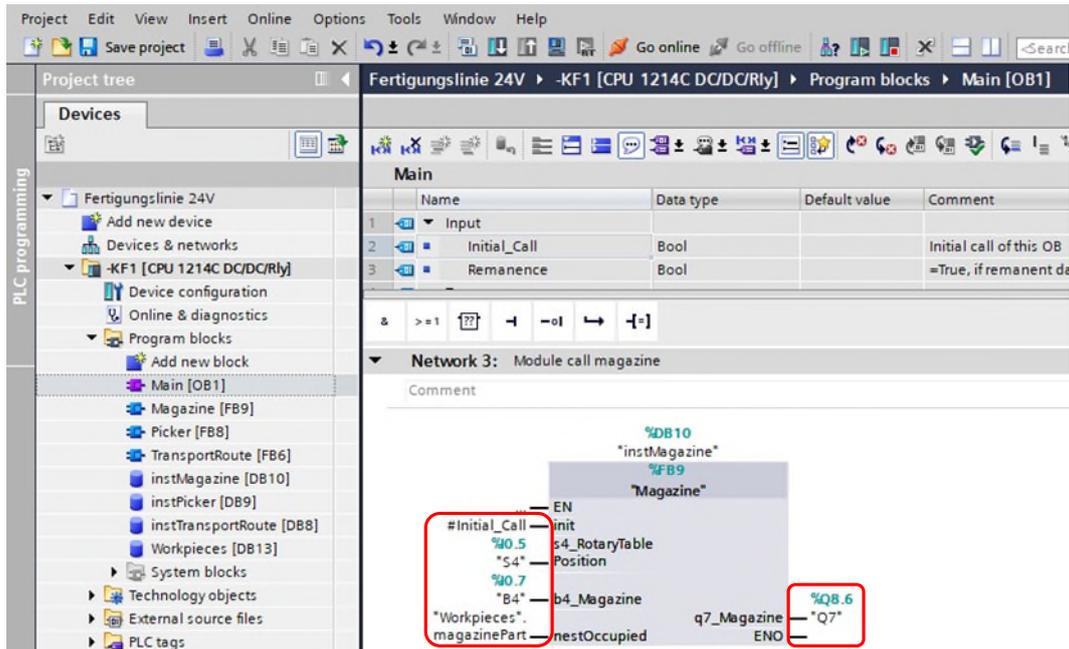


6. Llama al módulo de funciones en "MAIN" y crea una instancia:



Planificar y aplicar procesos automáticos - Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FUP] - Revista

7. Conecte la interfaz del bloque de funciones con las variables de entrada y salida de su tabla de variables:



**i** El bit de sistema "Initial\_Call" proporcionado por Siemens se utiliza como petición de inicialización. Es "TRUE" cuando la MAIN se ejecuta por primera vez.

8. Poner en servicio el sistema de forma estructurada con ayuda de un protocolo de puesta en servicio.

**i** Para colocar otra pieza en la mesa giratoria, puede ajustarse manualmente el estado correspondiente en la gestión de piezas.

Fertigungslinie 24V > -KF1 [CPU 1214C DC/DC/Rly] > Program blocks > Workpieces [DB13]

Keep actual values Snapshot Copy snapshots to start values Load start values as actual

	Name	Data type	Monitor value	Comment
1	Static			
2	magazinePart	Bool	FALSE	Nest in position Magazine is occupied with a workpiece
3	drillingRawPart	Bool	FALSE	Nest in drilling position is occupied by a non-drilled workpiece
4	drillingFinishedPart	Bool	FALSE	Nest in drilling position is occupied by a drilled workpiece
5	weldingRawPart	Bool	FALSE	Nest in welding position is occupied by a non-welded workpiece
6	weldingFinishedPart	Bool	FALSE	Nest in welding position is occupied by a non-welded workpiece
7	transferPart	Bool	TRUE	Nest in position transfer is occupied by a finished part



Solución

Planificar y aplicar procesos automáticos - Ejercicio: Convertir la cadena de secuencias GRAFCET en código de programa [FUP] - Revista

### Solución:

La solución se encuentra en el proyecto del Portal TIA "Fertigungslinie\_04\_Magazin\_FUP.zap17".



## 10.10 Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Revista

Objetivo:

Puedo crear el programa automático para el almacén de la línea de producción utilizando la descripción funcional y la cadena de secuencias creada en GRAFCET.

Tarea:

Cree el programa PLC para la secuencia automática de modo que la función se realice basándose en la descripción del sistema.

La cadena se inicializa al conectar la unidad de control.

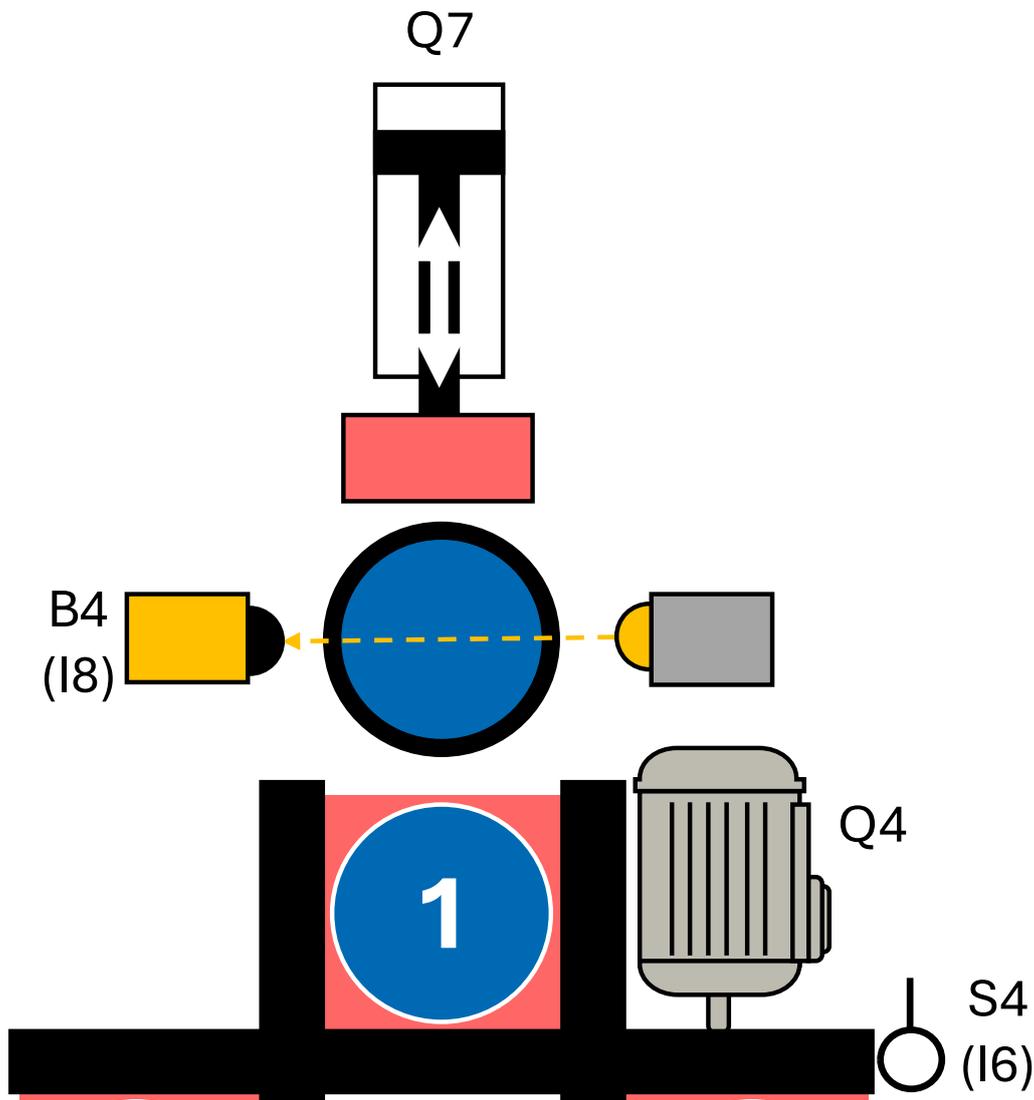
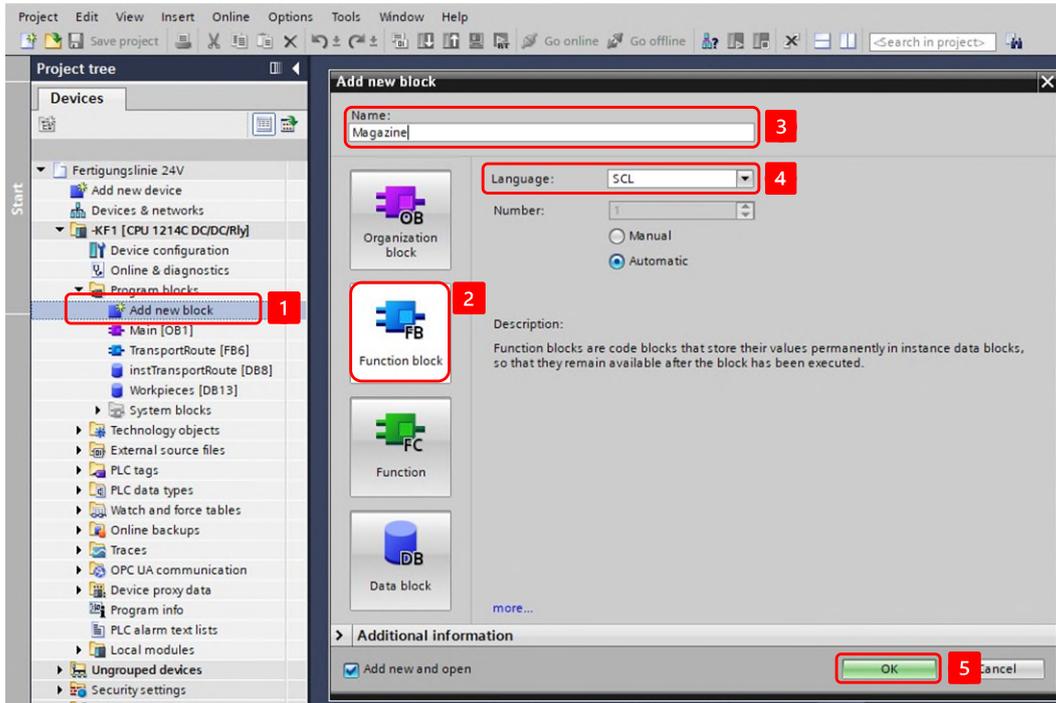


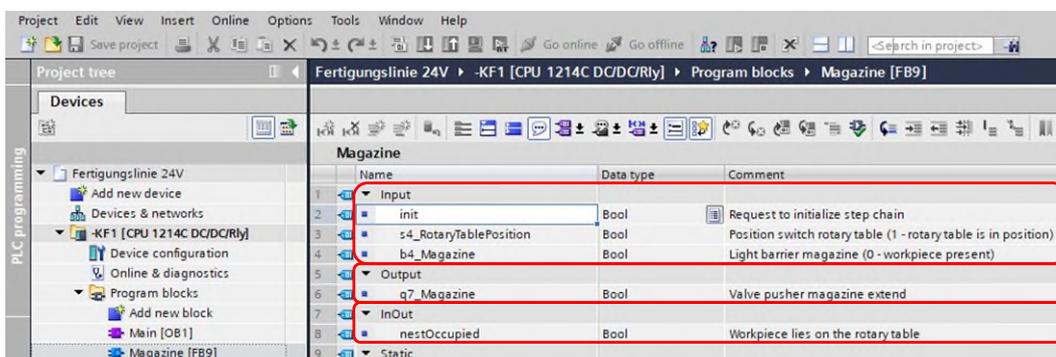
Imagen 6 Esquema del sistema - cargador

### Procedimiento:

1. Cree un nuevo bloque de funciones, seleccione el lenguaje de programación deseado y asígnele un nombre significativo:

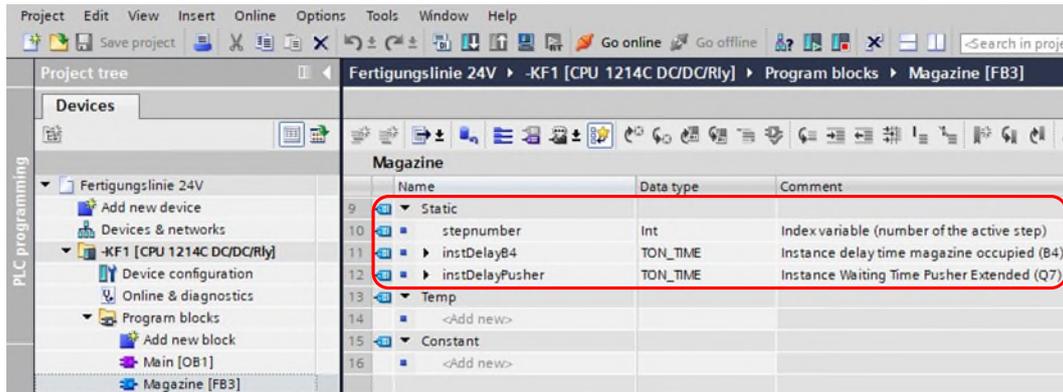


2. Declarar variables para los sensores y actuadores, una variable para inicializar la cadena de pasos y una variable para transferir la información de la pieza de trabajo en la interfaz del bloque de funciones:



Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Revista

3. Declare la variable de índice en el área estática de la interfaz del bloque de funciones, así como dos instancias para los tiempos de espera:



4. Programe la inicialización de la cadena:

```

1 // initialize step chain
2 IF #init THEN
3     #stepnumber := 1; // initialize step chain
4 END_IF;

```

5. Implemente los pasos individuales del GRAFCET en la siguiente estructura CASE. Debe crearse un nuevo CASO en la estructura para cada paso, que representa el número de paso:



Todas las acciones se restablecen en el paso inicial. De este modo se garantiza que no quede ninguna acción activada si se cancela una cadena de pasos activa mediante una solicitud de inicialización.

```

7 // step chain
8 CASE #stepnumber OF
9     1: // step 1 - initstep
10        // reset all actions
11        #q7_Magazine := FALSE;
12
13        // transition
14        IF #instDelayB4.Q // magazine not empty
15            AND #s4_RotaryTablePosition // rotary table is in position
16            AND NOT #nestOccupied // rotary table nest is empty
17        THEN
18            #stepnumber := 2; // next step
19        END_IF;
20
21     2: // step 2 - extend pusher
22        // actions
23        #q7_Magazine := #s4_RotaryTablePosition; // extend pusher aufahren, when table in position
24
25        // transitions
26        IF #instDelayPusher.Q THEN // pusher extended long time enough
27            #q7_Magazine := FALSE; // reset action
28            #stepnumber := 3; // next step
29        END_IF;
30
31     3: // step 3 - Nest belegt setzen
32        // actions
33        #nestOccupied := TRUE; // set nest is occupied
34
35        // transitions
36        IF #nestOccupied THEN // rotary table nest is occupied
37            #stepnumber := 1; // jump to init step
38        END_IF;
39 END_CASE;

```

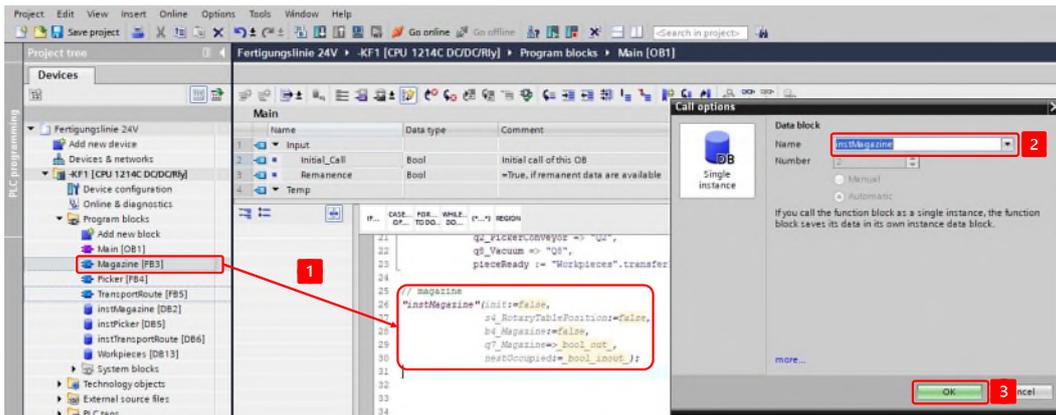
Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Revista

6. Programe las funciones horarias según la estructura CASE:

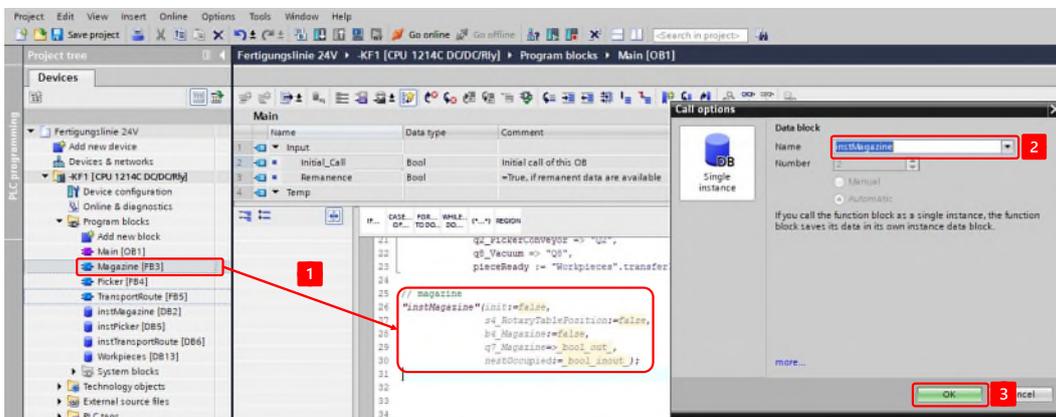
```

42 // delaytimes
43 // Light barrier interrupted B4
44 #instDelayB4(IN := NOT #b4_Magazine, // Light barrier interrupted
45 | PT := t#1s);
46
47 // pusher activated Q7
48 #instDelayPusher(IN := #q7_Magazine, // pusher activated
49 | PT := t#2s);
    
```

7. Llama al módulo de funciones en "MAIN" y crea una instancia:



8. Conecte la interfaz del bloque de funciones con las variables de entrada y salida de su tabla de variables:



El bit de sistema "Initial\_Call", que Siemens proporciona, se utiliza como petición de inicialización. Es "TRUE" cuando la MAIN se ejecuta por primera vez.

9. Poner en servicio el sistema de forma estructurada con ayuda de un protocolo de puesta en servicio.



Para colocar otra pieza en la mesa giratoria, puede ajustarse manualmente el estado correspondiente en la gestión de piezas.

Fertigungsline 24V ▶ -KF1 [CPU 1214C DC/DC/Rly] ▶ Program blocks ▶ Workpieces [DB13]

Keep actual values Snapshot Copy snapshots to start values Load start values as actual

Workpieces				
	Name	Data type	Monitor value	Comment
1	Static			
2	magazinePart	Bool	FALSE	Nest in position Magazine is occupied with a workpiece
3	drillingRawPart	Bool	FALSE	Nest in drilling position is occupied by a non-drilled workpiece
4	drillingFinishedPart	Bool	FALSE	Nest in drilling position is occupied by a drilled workpiece
5	weldingRawPart	Bool	FALSE	Nest in welding position is occupied by a non-welded workpiece
6	weldingFinishedPart	Bool	FALSE	Nest in welding position is occupied by a non-welded workpiece
7	transferPart	Bool	TRUE	Nest in position transfer is occupied by a finished part



Solución

Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Revista

### Solución:

La solución se encuentra en el proyecto del Portal TIA "Fertigungslinie\_04\_Magazin\_SCL.zap17".



## 10.11 Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [FBD] - Estación de tratamiento

Objetivo:

Puedo crear el programa automático para una estación de procesamiento en la línea de producción utilizando la descripción de la función y la cadena de secuencia creada en GRAFCET.

Tarea:

Cree el programa PLC para la secuencia automática de modo que la función se realice basándose en la descripción del sistema.

La cadena se inicializa al conectar la unidad de control.

El bloque de funciones debe llamarse dos veces, una instancia para la estación de taladrado (Q9) y otra instancia para la soldadura (Q10).

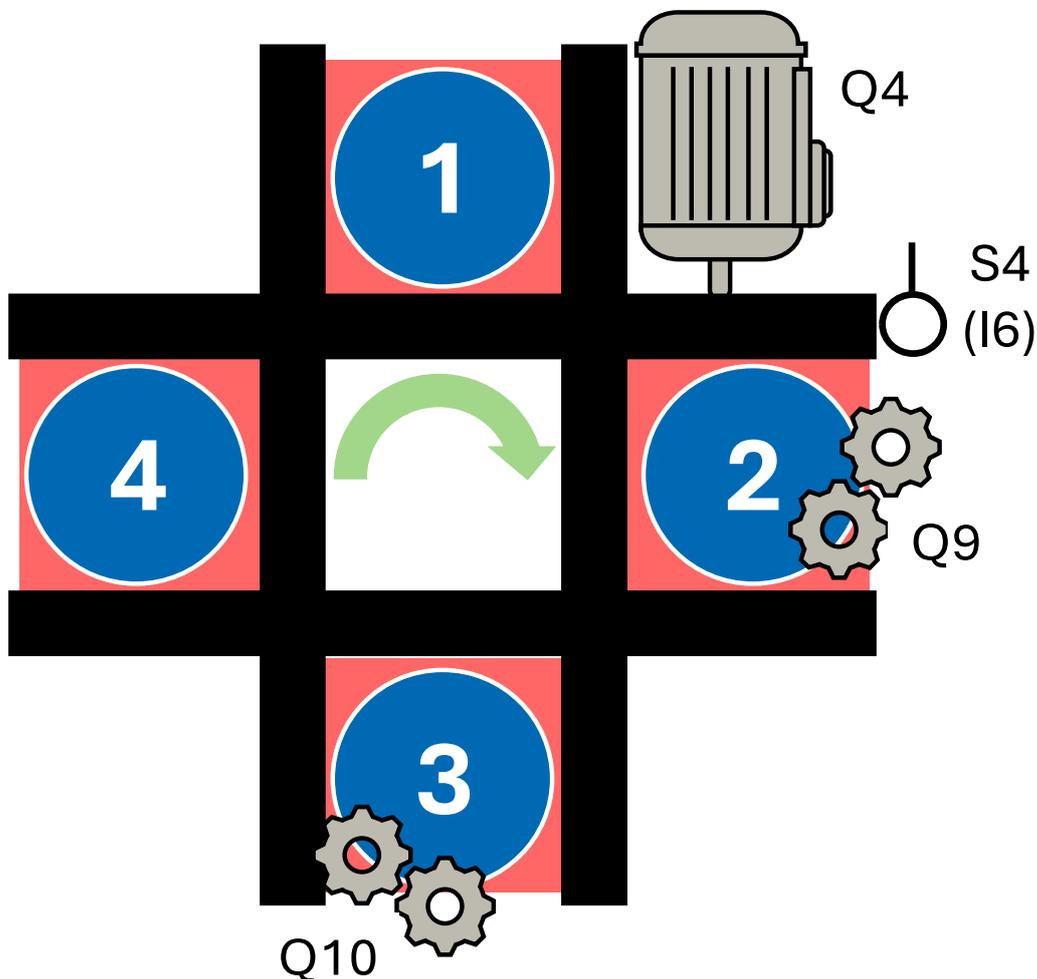
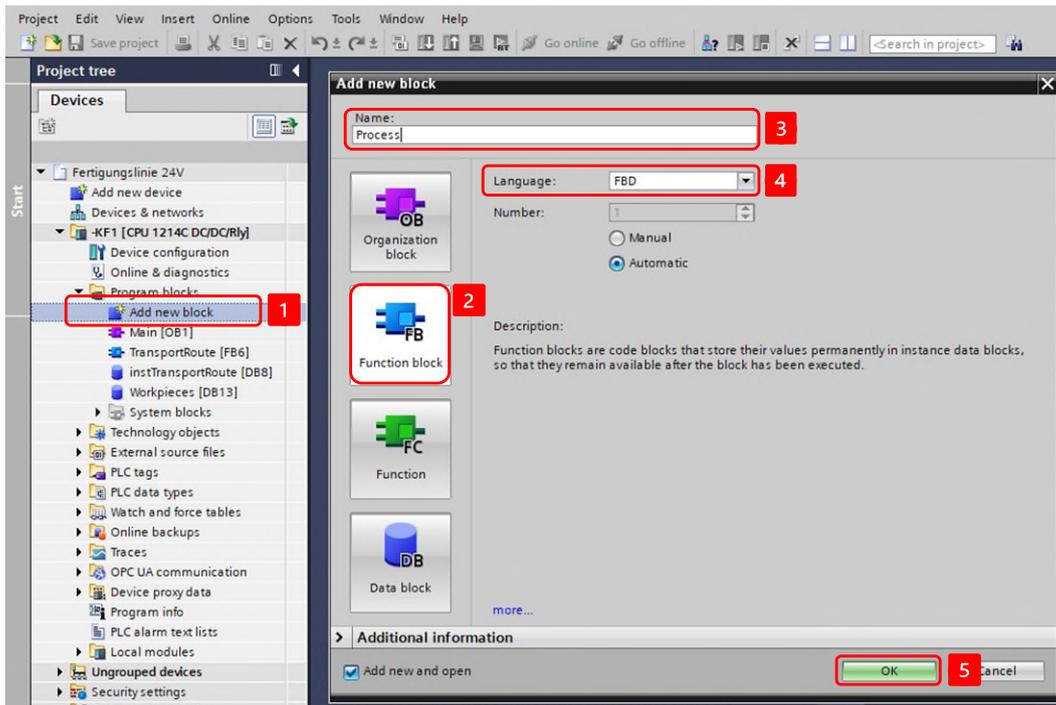


Imagen 7 Esquema del sistema - estación de tratamiento

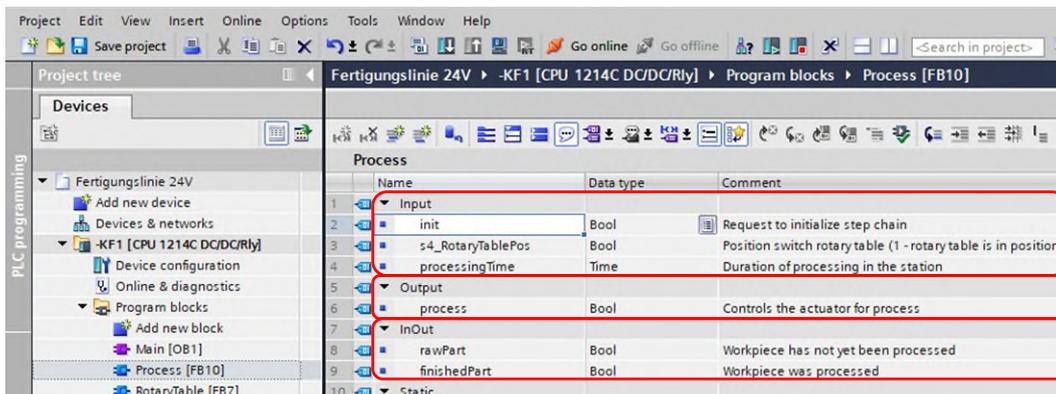
## Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [FBD] - Estación de tratamiento

### Procedimiento:

1. Cree un nuevo bloque de funciones, seleccione el lenguaje de programación deseado y asígnele un nombre significativo:

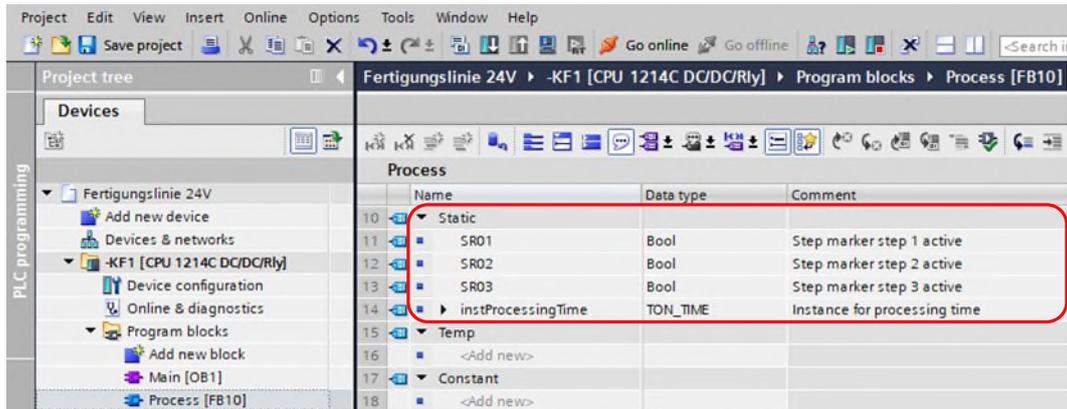


2. Declare variables para los sensores y actuadores, una variable para inicializar la cadena de pasos, para el tiempo de procesamiento y dos variables para transferir la información de la pieza de trabajo en la interfaz del bloque de funciones:

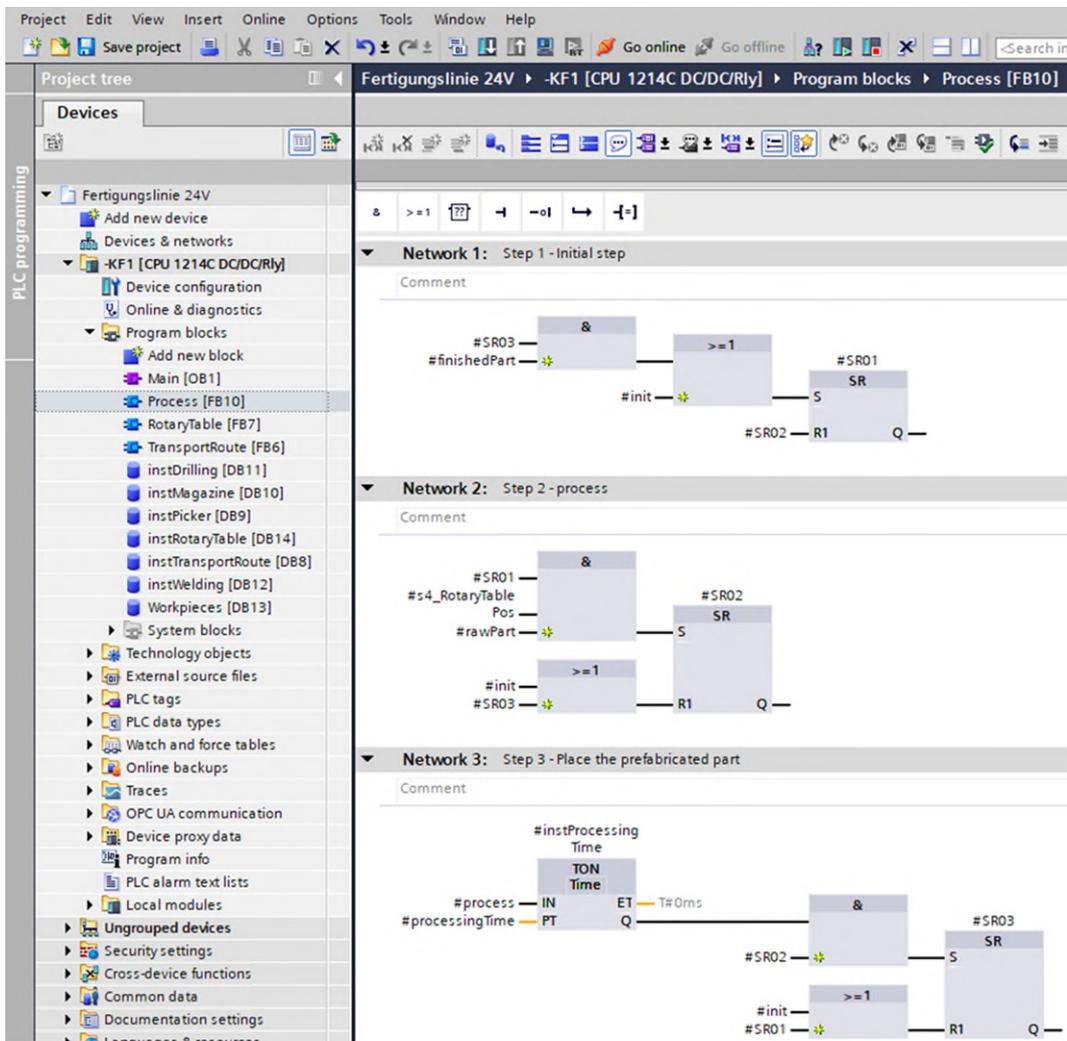


Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [FBD] - Estación de tratamiento

3. Declare los indicadores de paso en el área estática de la interfaz del bloque de funciones, así como una instancia para el tiempo de procesamiento:

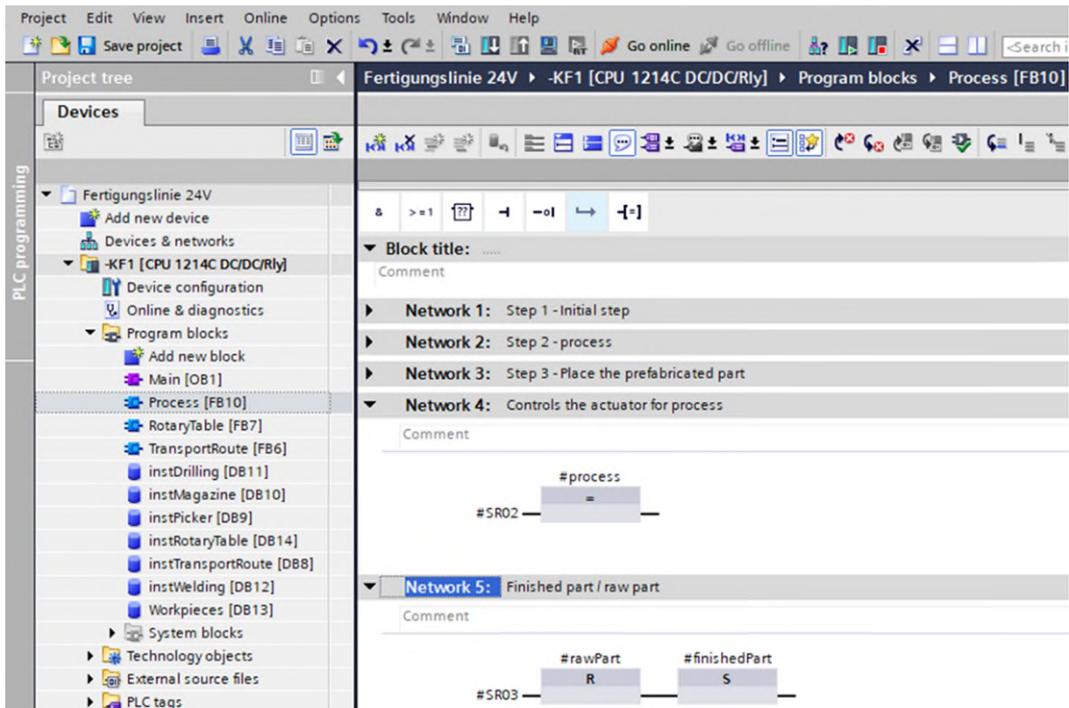


4. Implementar los pasos individuales utilizando flip-flops basados en el GRAFCET. Debe utilizarse una red nueva para cada paso:

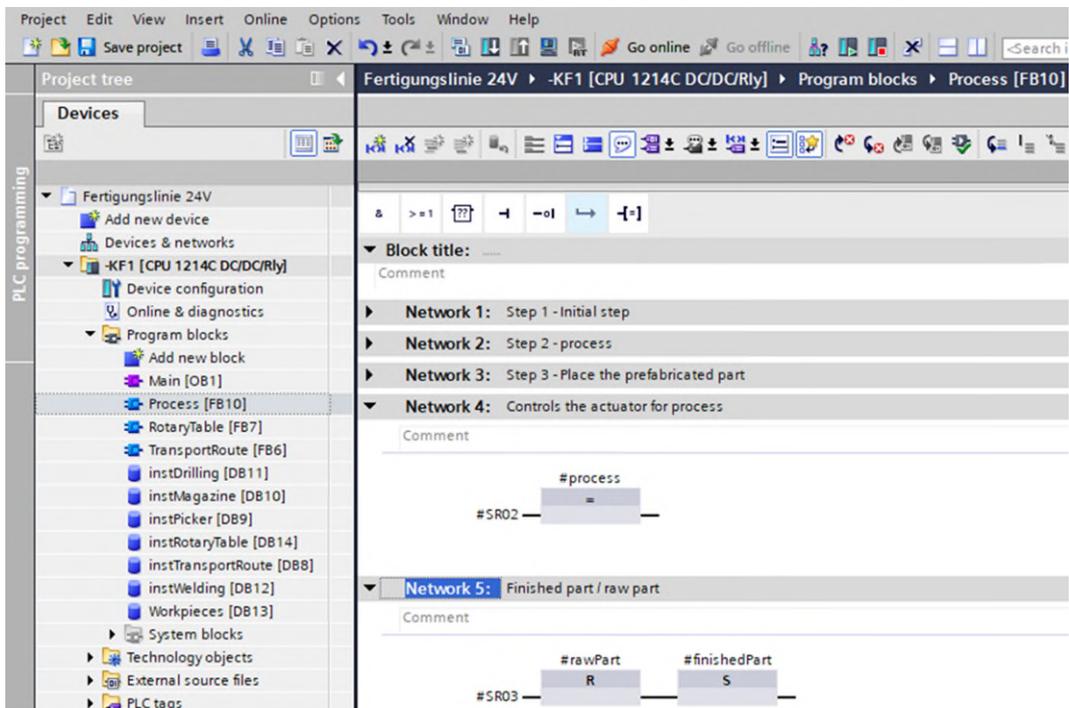


Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [FBD] - Estación de tratamiento

5. Asigne las acciones debajo de la cadena de pasos en las redes siguientes:



6. Llame al módulo de funciones dos veces en "MAIN" y cree una instancia distinta cada vez:



Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [FBD] - Estación de tratamiento

7. Conecte la interfaz del bloque de funciones con las variables de entrada y salida de su tabla de variables, así como el tiempo de procesamiento:

**i** El bit de sistema "Initial\_Call" proporcionado por Siemens se utiliza como petición de inicialización. Es "TRUE" cuando la MAIN se ejecuta por primera vez.

8. Poner en servicio el sistema de forma estructurada con ayuda de un protocolo de puesta en servicio.

**i** Para que otra pieza pueda procesarse en la estación, puede ajustarse manualmente al estado correspondiente en la gestión de piezas.

Fertigungslinie 24V - KF1 [CPU 1214C DC/DC/Rly] - Program blocks - Workpieces [DB13]				
Keep actual values   Snapshot   Copy snapshots to start values   Load start values as actual				
Workpieces				
	Name	Data type	Monitor value	Comment
1	Static			
2	magazinePart	Bool	FALSE	Nest in position Magazine is occupied with a workpiece
3	drillingRawPart	Bool	TRUE	Nest in drilling position is occupied by a non-drilled workpiece
4	drillingFinishedPart	Bool	FALSE	Nest in drilling position is occupied by a drilled workpiece
5	weldingRawPart	Bool	FALSE	Nest in welding position is occupied by a non-welded workpiece
6	weldingFinishedPart	Bool	FALSE	Nest in welding position is occupied by a non-welded workpiece
7	transferPart	Bool	FALSE	Nest in position transfer is occupied by a finished part



Solución

Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [FBD] - Estación de tratamiento

### Solución:

La solución se encuentra en el proyecto del Portal TIA "Fertigungslinie\_05\_Bearbeiten\_FUP.zap17".



## 10.12 Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Estación de tratamiento

Objetivo:

Puedo crear el programa automático para una estación de procesamiento en la línea de producción utilizando la descripción funcional y la cadena de secuencia creada en GRAFCET.

Tarea:

Cree el programa PLC para la secuencia automática de modo que la función se realice basándose en la descripción del sistema.

La cadena se inicializa al encender la unidad de control.

El bloque de funciones debe llamarse dos veces, una instancia para la estación de taladrado (Q9) y otra instancia para la soldadura (Q10).

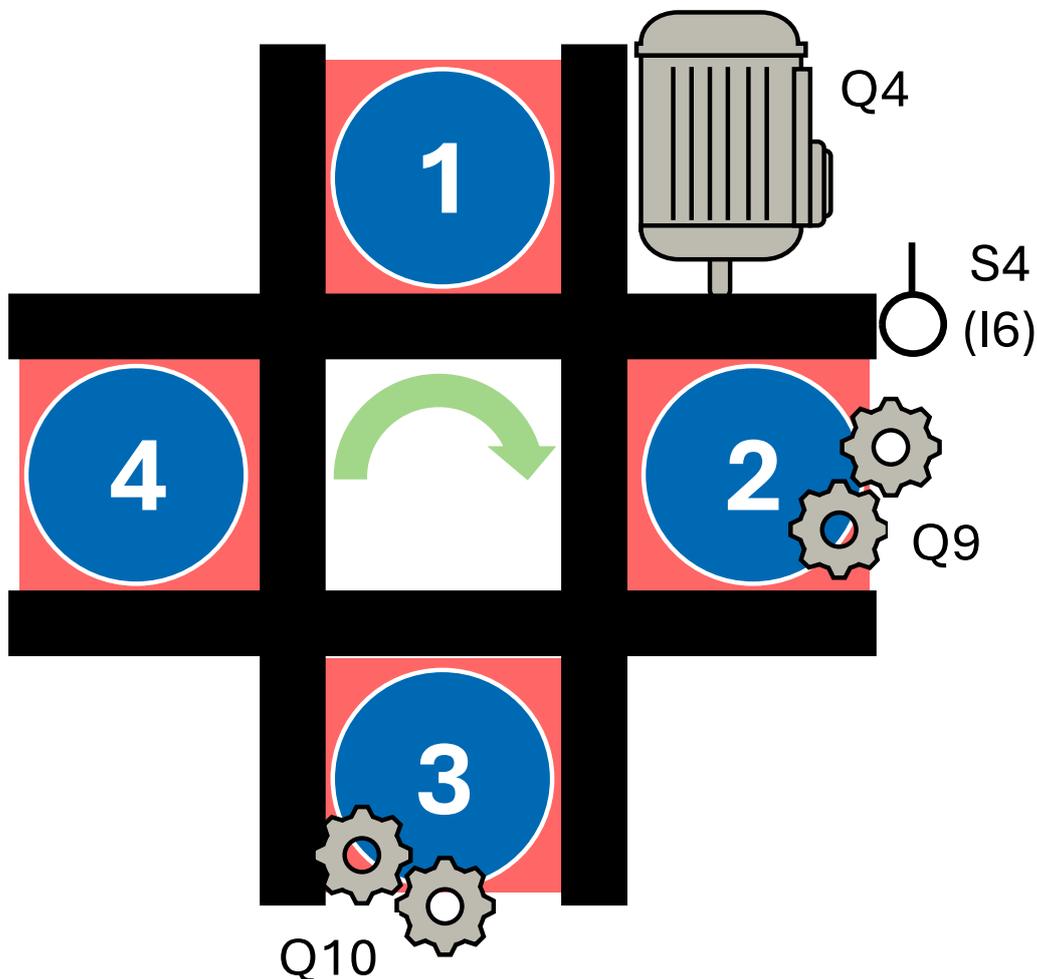
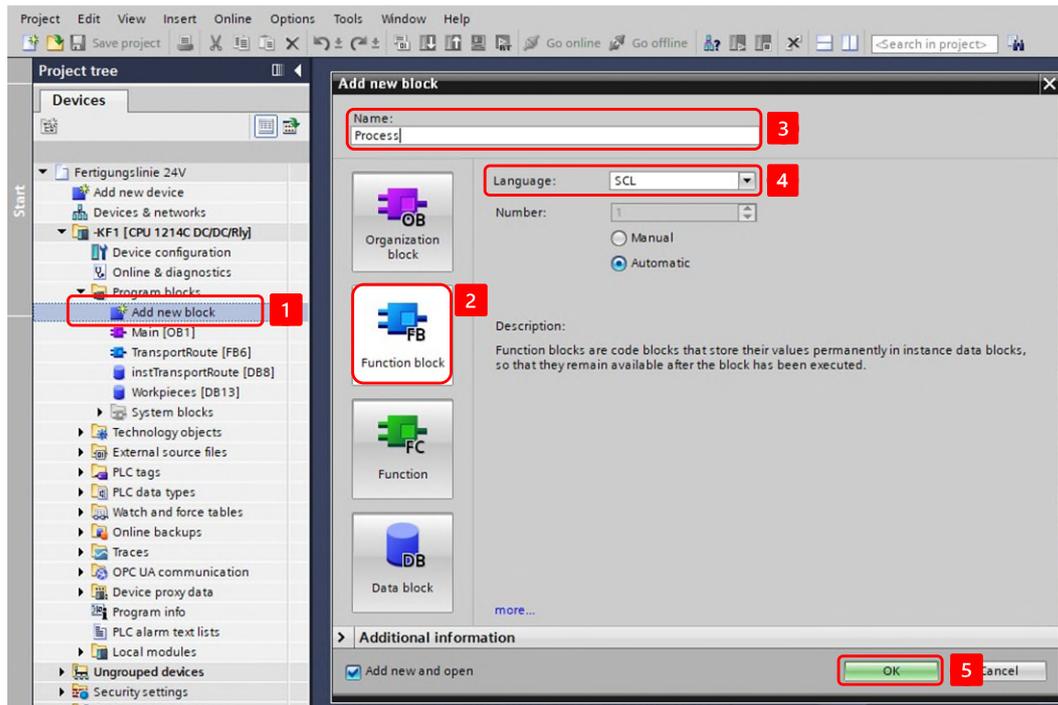


Imagen 8 Esquema del sistema - estación de tratamiento

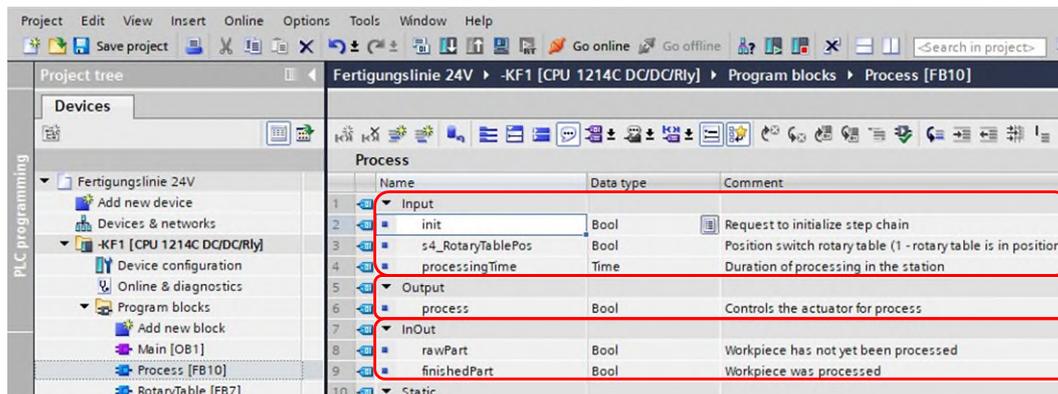
## Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Estación de tratamiento

### Procedimiento:

1. Cree un nuevo bloque de funciones, seleccione el lenguaje de programación deseado y asígnele un nombre significativo:

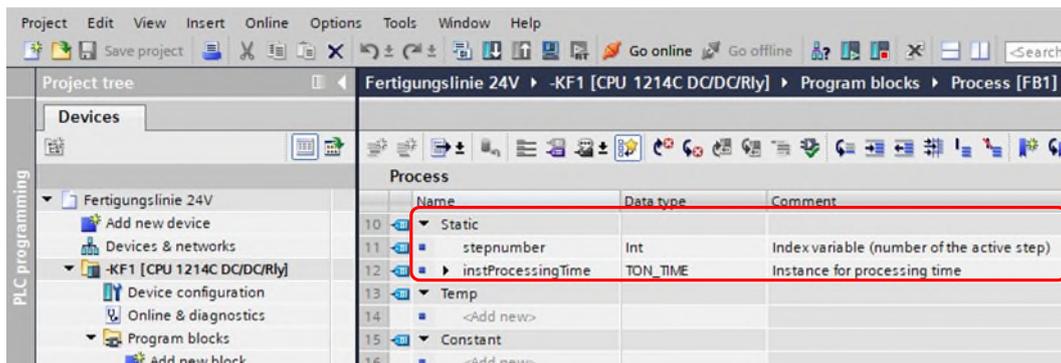


2. Declare variables para los sensores y actuadores, una variable para inicializar la cadena de pasos, para el tiempo de procesamiento y dos variables para transferir la información de la pieza de trabajo en la interfaz del bloque de funciones:



Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Estación de tratamiento

3. Declare la variable de índice en el área estática de la interfaz del bloque de funciones, así como una instancia para el tiempo de procesamiento:



4. Programe la inicialización de la cadena:

```

1 // initialize step chain
2 IF #init THEN
3     #stepnumber := 1; // initialize step chain
4 END_IF;

```

5. Implemente los pasos individuales del GRAFCET en la siguiente estructura CASE. Debe crearse un nuevo CASO en la estructura para cada paso, que representa el número de paso:



Todas las acciones se restablecen en el paso inicial. De este modo se garantiza que no quede ninguna acción activada si se cancela una cadena de pasos activa mediante una solicitud de inicialización.

```

7 // step chain
8 CASE #stepnumber OF
9     1: // step 1 - initstep
10        // reset all actions
11        #process := FALSE;
12
13        // transitions
14        IF #s4_RotaryTablePos // rotary table is in position
15            AND #rawPart // raw part in station
16        THEN
17            #stepnumber := 2; // next step
18        END_IF;
19     2: // step 2 - process
20        // actions
21        #process := TRUE; // set process
22
23        // transitions
24        IF #instProcessingTime.Q THEN // Processing time expired
25            #process := FALSE; // reset action
26            #stepnumber := 3; // next step
27        END_IF;
28
29     3: // step 3 - set finished part
30        // actions
31        #finishedPart := TRUE; // set finished part in station
32        #rawPart := FALSE; // reset raw part in station
33
34        // transitions
35        #stepnumber := 1; // jump to init step
36 END_CASE;

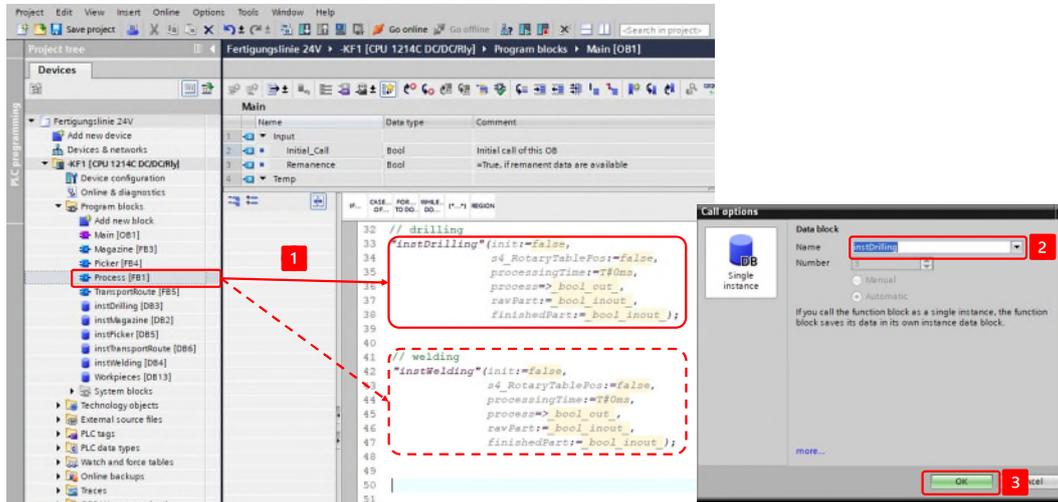
```

Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Estación de tratamiento

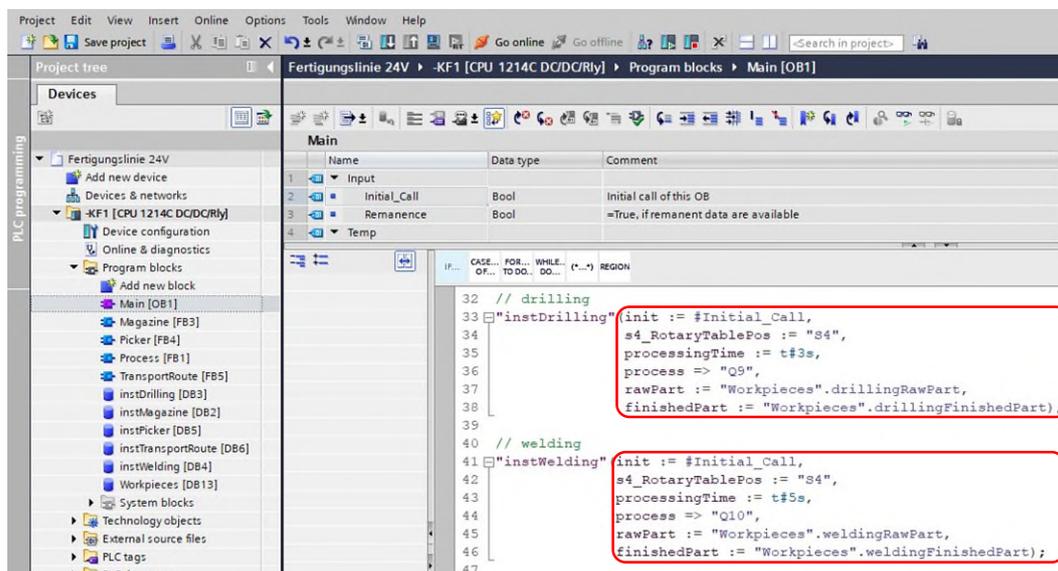
6. Programe las funciones horarias según la estructura CASE:

```
40 // delay Processing time
41 #instProcessingTime(IN := #process,           // start timer
42                    PT := #processingTime); // duration
```

7. Llame al módulo de funciones dos veces en "MAIN" y cree una instancia distinta cada vez:



8. Conecte la interfaz del bloque de funciones con las variables de entrada y salida de su tabla de variables, así como el tiempo de procesamiento:



El bit de sistema "Initial\_Call" proporcionado por Siemens se utiliza como petición de inicialización. Es "TRUE" cuando la MAIN se ejecuta por primera vez.

Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Estación de tratamiento

9. Poner en servicio el sistema de forma estructurada con ayuda de un protocolo de puesta en servicio.



Para que otra pieza pueda procesarse en la estación, puede ajustarse manualmente al estado correspondiente en la gestión de piezas.

Fertigungslinie 24V ▶ -KF1 [CPU 1214C DC/DC/Rly] ▶ Program blocks ▶ Workpieces [DB13]

Keep actual values Snapshot Copy snapshots to start values Load start values as actual

Workpieces				
	Name	Data type	Monitor value	Comment
1	Static			
2	magazinePart	Bool	FALSE	Nest in position Magazine is occupied with a workpiece
3	drillingRawPart	Bool	TRUE	Nest in drilling position is occupied by a non-drilled workpiece
4	drillingFinishedPart	Bool	FALSE	Nest in drilling position is occupied by a drilled workpiece
5	weldingRawPart	Bool	FALSE	Nest in welding position is occupied by a non-welded workpiece
6	weldingFinishedPart	Bool	FALSE	Nest in welding position is occupied by a non-welded workpiece
7	transferPart	Bool	FALSE	Nest in position transfer is occupied by a finished part



Solución

Planificar y aplicar procesos automáticos - Ejercicio: Conversión de la cadena de secuencias GRAFCET en código de programa [ST / SCL] - Estación de tratamiento

### Solución:

La solución se encuentra en el proyecto del Portal TIA "Fertigungslinie\_05\_Bearbeiten\_SCL.zap17".

## 10.13 Ejercicio: Crear bloque de función para mesa giratoria [FBD]

Objetivo:

Puedo crear el programa automático para la mesa giratoria de la línea de producción utilizando la descripción de la función.

Tarea:

Cree el programa PLC para la secuencia automática de forma que la función se implemente basándose en la descripción del sistema.

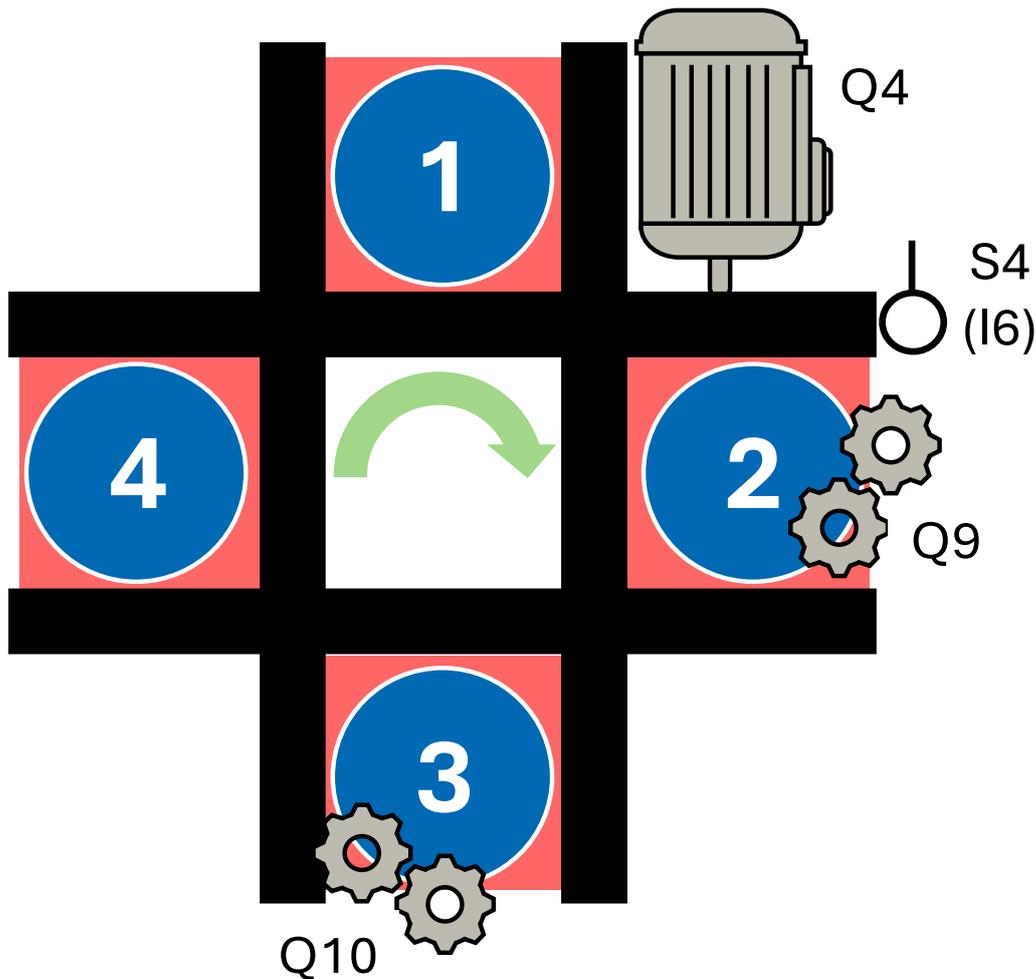


Imagen 9 Esquema del sistema - plataforma giratoria

Función:

Para controlar la unidad de giro, deben cumplirse constantemente las siguientes condiciones de desbloqueo:

- Deslizador Q7 no activado
- El convertidor (S2) no está en la posición final en la plataforma giratoria.

La mesa debe desplazarse hasta que vuelva a su posición (S4).

La mesa se puede mover si,

- una pieza de trabajo se encuentra en el nido del cargador (I).

Planificar y aplicar procesos automáticos - Ejercicio: Crear bloque de función para mesa giratoria [FBD]

- una pieza acabada se encuentra en uno de los dos puestos de tratamiento (2/3).

La mesa no podrá moverse mientras

- una pieza de trabajo se encuentra en el nido de la estación de transferencia (4).
- una pieza bruta se encuentra en una de las dos estaciones de tratamiento (2/3).

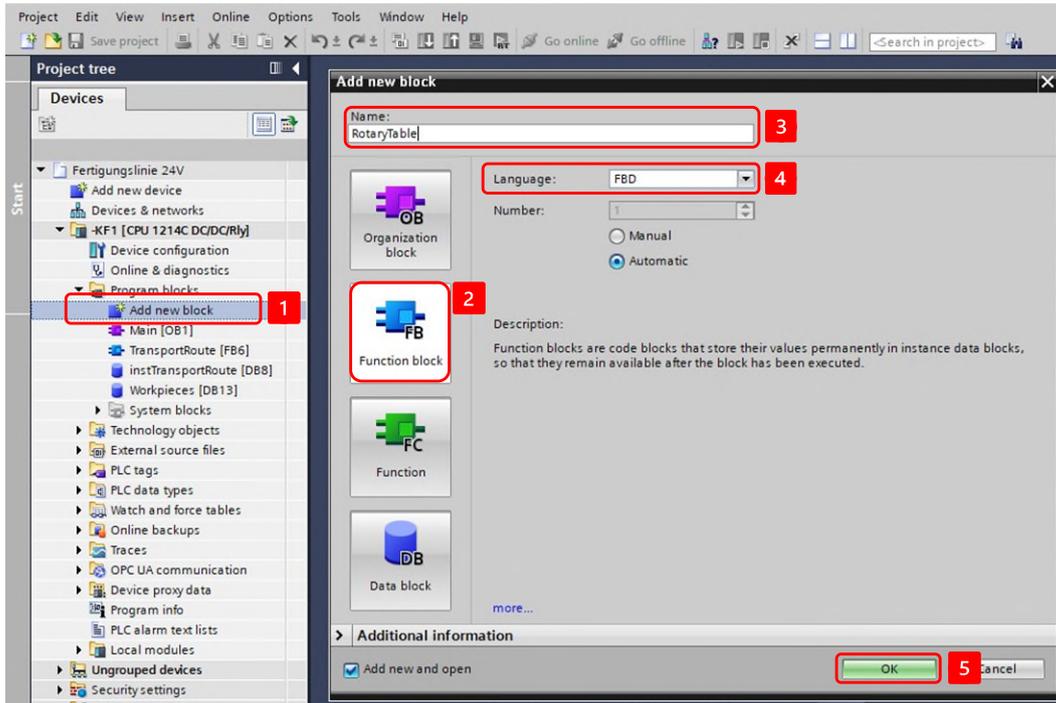
Si la mesa se ha girado a la siguiente posición (arista positiva S4), los datos de la pieza también deben girarse una estación:

- Si una pieza acabada se encuentra en la estación de "Soldadura", debe reiniciarse y fijarse como pieza de trabajo en la estación de transferencia.
- Si una pieza acabada se encuentra en la estación "Taladrado", debe restablecerse y configurarse como pieza bruta en la estación "Soldadura".
- Si una pieza se encuentra en la estación "Almacén", debe reiniciarse y configurarse como pieza bruta en la estación "Taladrado".

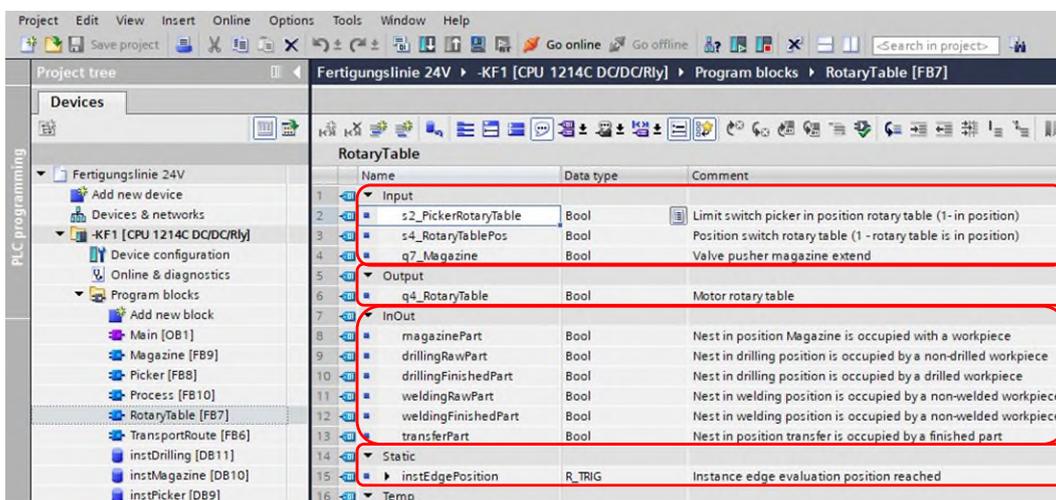
Planificar y aplicar procesos automáticos - Ejercicio: Crear bloque de función para mesa giratoria [FBD]

Procedimiento:

1. Cree un nuevo bloque de funciones, seleccione el lenguaje de programación deseado y asígnele un nombre significativo:

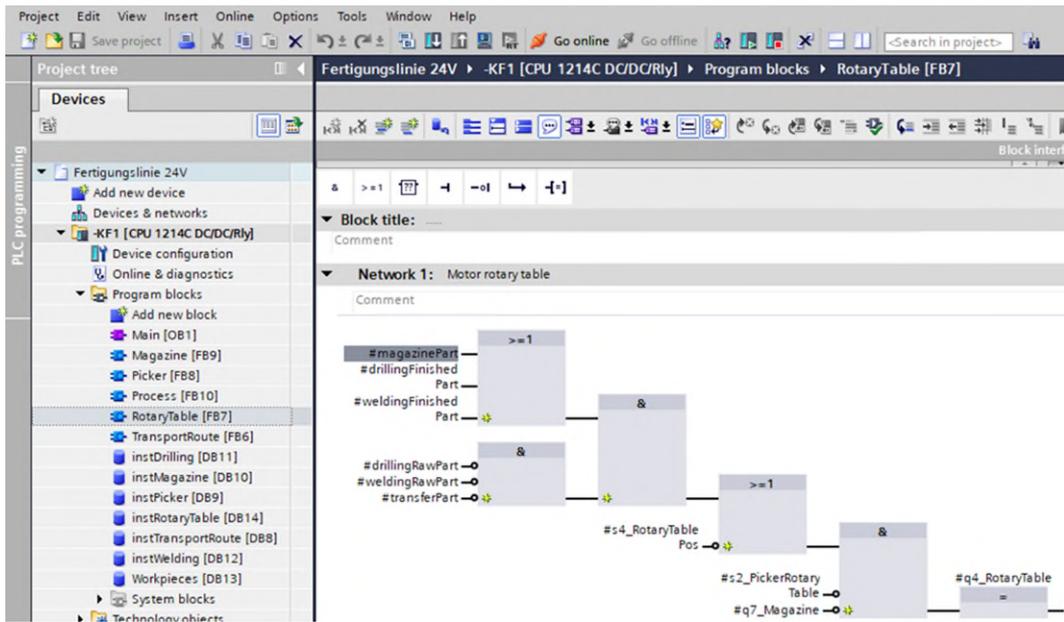


2. Declare variables para los sensores y actuadores, las variables para transferir la información de la pieza y la instancia para la evaluación de aristas en la interfaz del bloque de funciones:

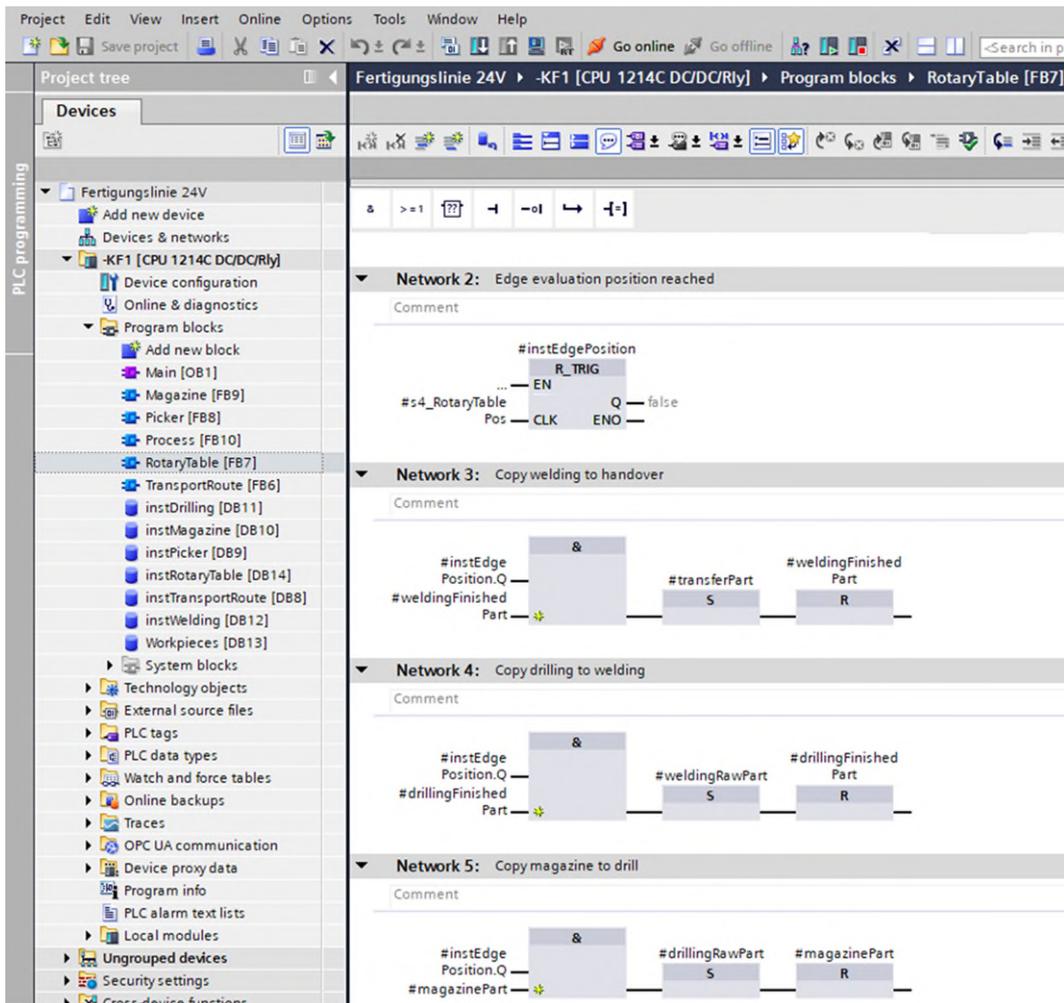


Planificar y aplicar procesos automáticos - Ejercicio: Crear bloque de función para mesa giratoria [FBD]

3. Programar el control de la plataforma giratoria (Q4) en la primera red:

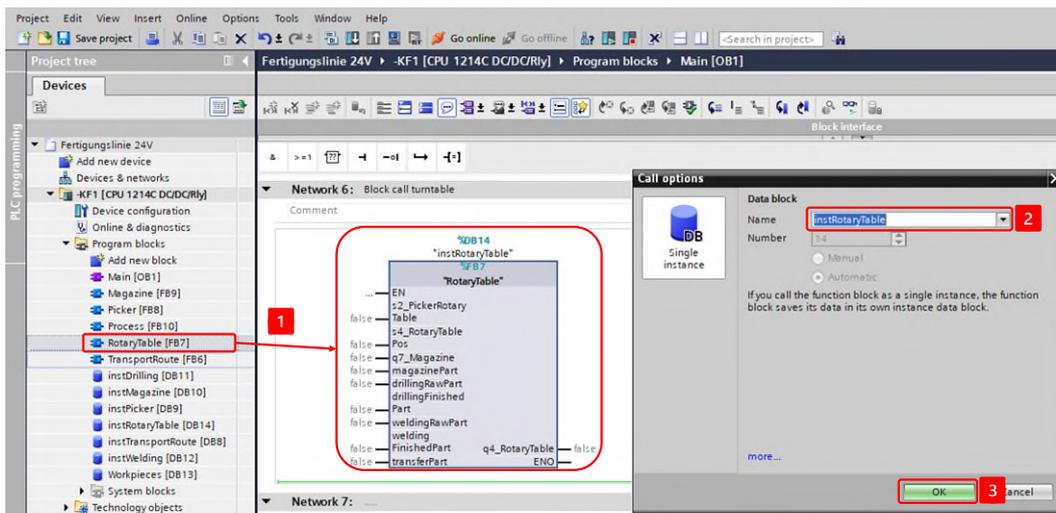


4. Programe la evaluación de bordes (S4) y la rotación de los datos de la pieza en las siguientes redes:

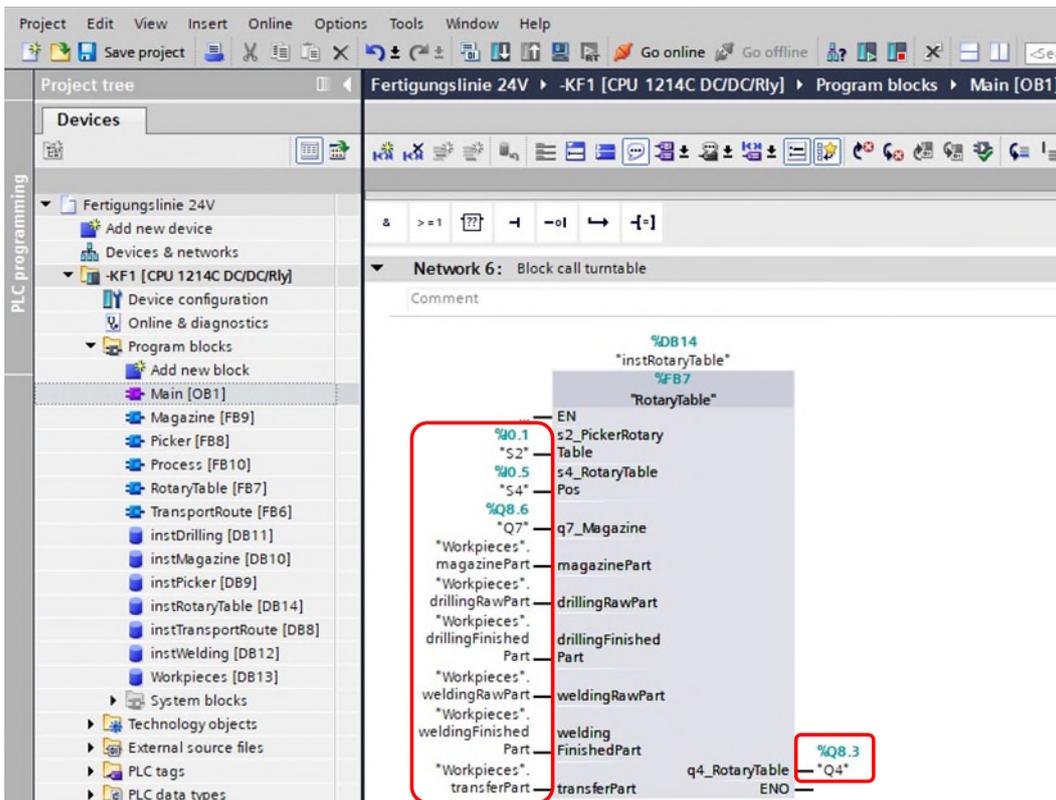


Planificar y aplicar procesos automáticos - Ejercicio: Crear bloque de función para mesa giratoria [FBD]

5. Llama al módulo de funciones en "MAIN" y crea una instancia:



6. Conecte la interfaz del módulo:



7. Poner en servicio el sistema de forma estructurada con ayuda de un protocolo de puesta en servicio.



Solución

Planificar y aplicar procesos automáticos - Ejercicio: Crear bloque de función para mesa giratoria [FBD]

### Solución:

La solución se encuentra en el proyecto del Portal TIA "Fertigungslinie\_06\_Drehtisch\_FUP.zap17".



## 10.14 Ejercicio: Crear bloque de función para mesa giratoria [ST / SCL].

Objetivo:

Puedo crear el programa automático para la mesa giratoria de la línea de producción utilizando la descripción de la función.

Tarea:

Cree el programa PLC para la secuencia automática de forma que la función se implemente basándose en la descripción del sistema.

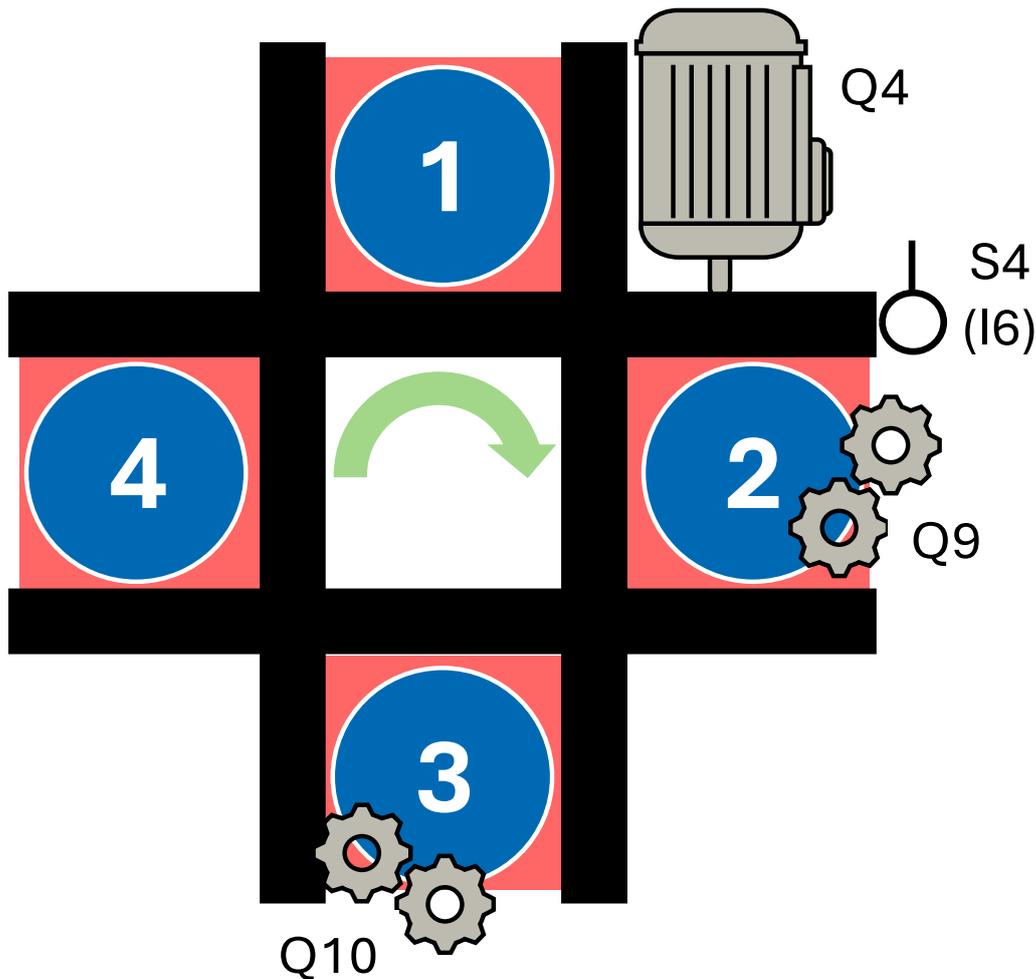


Imagen 10 Esquema del sistema - plataforma giratoria

Función:

Para controlar la unidad de giro, deben cumplirse constantemente las siguientes condiciones de desbloqueo:

- Deslizador Q7 no activado
- El convertidor (S2) no está en la posición final en la plataforma giratoria.

La mesa debe desplazarse hasta que vuelva a su posición (S4).

La mesa se puede mover si,

- una pieza de trabajo se encuentra en el nido del cargador (I).

Planificar y aplicar procesos automáticos - Ejercicio: Crear bloque de función para mesa giratoria [ST / SCL].

- una pieza acabada se encuentra en uno de los dos puestos de tratamiento (2/3).

La mesa no podrá moverse mientras

- una pieza de trabajo se encuentra en el nido de la estación de transferencia (4).
- una pieza bruta se encuentra en una de las dos estaciones de tratamiento (2/3).

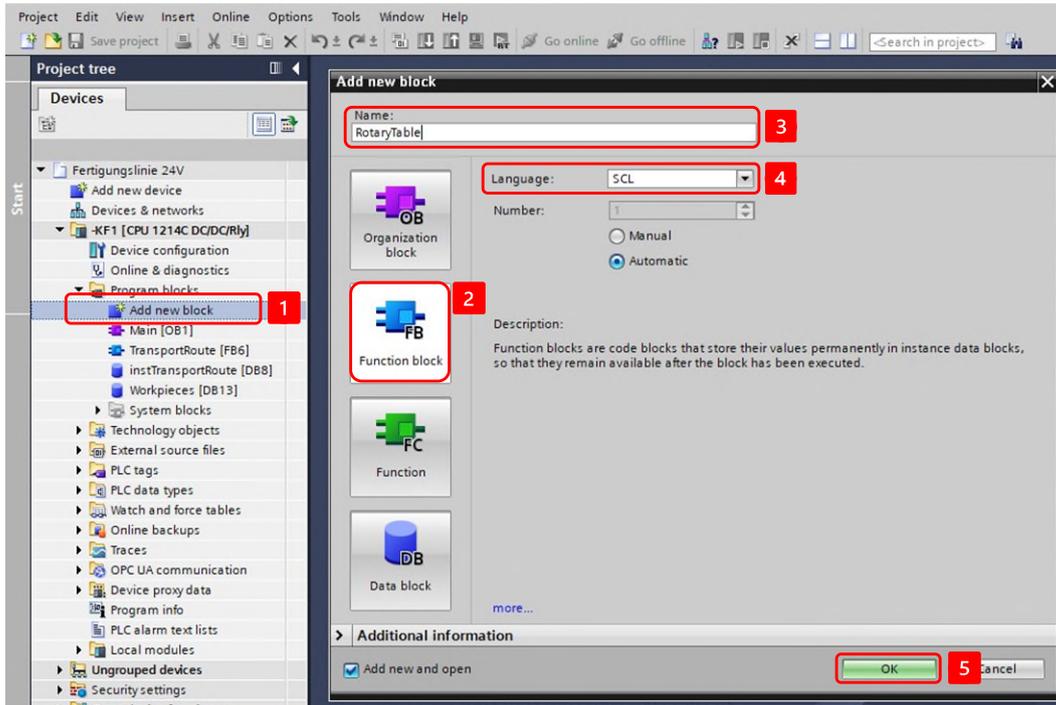
Si la mesa se ha girado a la siguiente posición (arista positiva S4), los datos de la pieza también deben girarse una estación:

- Si una pieza acabada se encuentra en la estación de "Soldadura", debe reiniciarse y fijarse como pieza de trabajo en la estación de transferencia.
- Si una pieza acabada se encuentra en la estación "Taladrado", debe restablecerse y configurarse como pieza bruta en la estación "Soldadura".
- Si una pieza se encuentra en la estación "Almacén", debe reiniciarse y configurarse como pieza bruta en la estación "Taladrado".

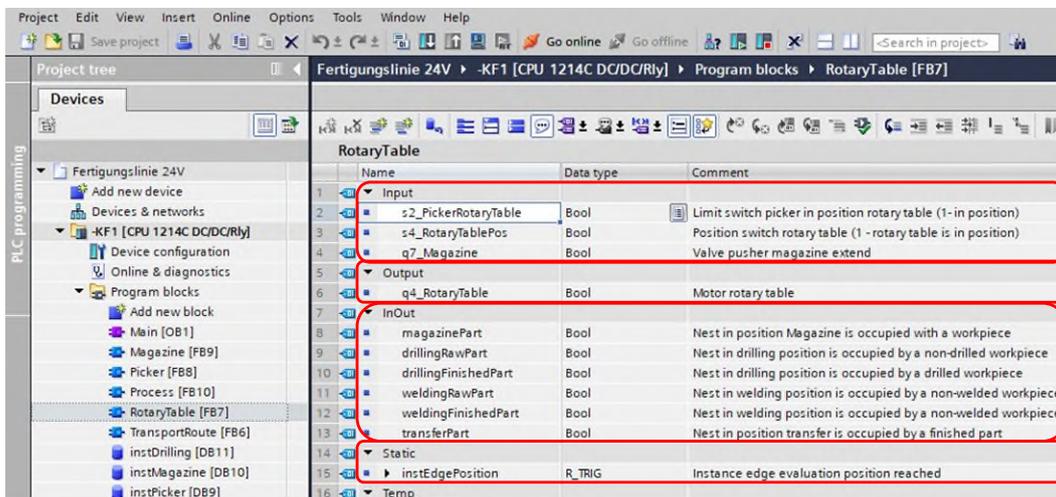
Planificar y aplicar procesos automáticos - Ejercicio: Crear bloque de función para mesa giratoria [ST/SCL].

**Procedimiento:**

1. Cree un nuevo bloque de funciones, seleccione el lenguaje de programación deseado y asígnele un nombre significativo:



2. Declare variables para los sensores y actuadores, las variables para transferir la información de la pieza y la instancia para la evaluación de aristas en la interfaz del bloque de funciones:



Planificar y aplicar procesos automáticos - Ejercicio: Crear bloque de función para mesa giratoria [ST/SCL].

3. Programe el control de la plataforma giratoria (Q4):

```

1 // control the rotary table motor
2 #q4_RotaryTable :=
3 NOT #q7_Magazine AND NOT #s2_PickerRotaryTable // permanent interlocks
4 AND (NOT #s4_RotaryTablePos // rotate until back into position
5 OR ((NOT #drillingRawPart AND NOT #weldingRawPart AND NOT #transferPart) // workpiece status locks rotate
6 AND (#magazinePart OR #drillingFinishedPart OR #weldingFinishedPart)); // workpiece status requires turning

```

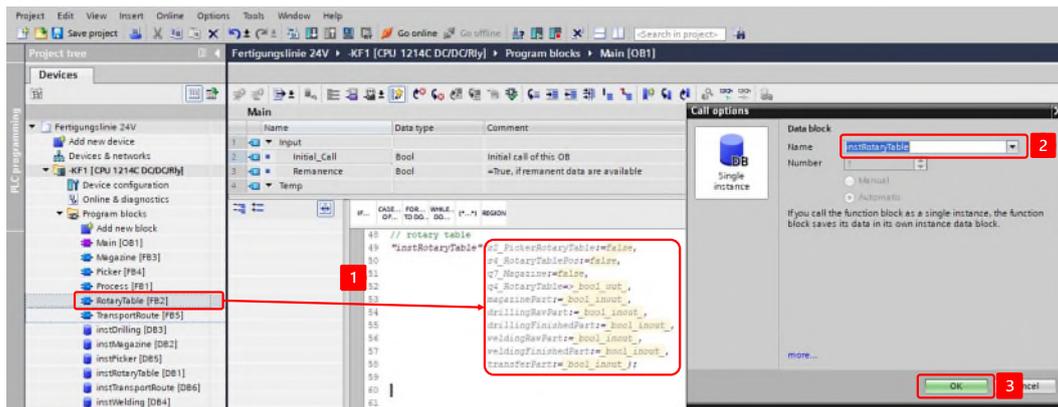
4. Programe la evaluación de bordes (S4) y la rotación de los datos de la pieza:

```

9 // edge evaluation position reached
10 #instEdgePosition(CLK := #s4_RotaryTablePos);
11
12
13 // rotate workpiece data
14 IF #instEdgePosition.Q THEN
15     #drillingRawPart := #magazinePart;
16     #weldingRawPart := #drillingFinishedPart;
17     #transferPart := #weldingFinishedPart;
18
19     #magazinePart := #drillingFinishedPart := #weldingFinishedPart := FALSE;
20 END_IF;

```

5. Llama al módulo de funciones en "MAIN" y crea una instancia:



Planificar y aplicar procesos automáticos - Ejercicio: Crear bloque de función para mesa giratoria [ST/SCL].

6. Conecte la interfaz del módulo:

The screenshot shows the SIMATIC Manager interface for a PLC project. The left pane displays the project tree for 'Fertigungsline 24V' and 'KF1 [CPU 1214C DC/DC/Rly]'. The main workspace shows the 'Main' program block with a table of variables and a code editor. The code editor contains the following SCL code for the 'instRotaryTable' function block:

```
48 // rotary table
49 instRotaryTable s2_PickerRotaryTable := "S2",
50               s4_RotaryTablePos := "S4",
51               q7_Magazine := "Q7",
52               q4_RotaryTable => "Q4",
53               magazinePart := "Workpieces".magazinePart,
54               drillingRawPart := "Workpieces".drillingRawPart,
55               drillingFinishedPart := "Workpieces".drillingFinishedPart,
56               weldingRawPart := "Workpieces".weldingRawPart,
57               weldingFinishedPart := "Workpieces".weldingFinishedPart,
58               transferPart := "Workpieces".transferPart;
```

7. Poner en servicio el sistema de forma estructurada con ayuda de un protocolo de puesta en servicio.

Planificar y aplicar procesos automáticos - Ejercicio: Crear bloque de función para mesa giratoria [ST / SCL].



Solución

Planificar y aplicar procesos automáticos - Ejercicio: Crear bloque de función para mesa giratoria [ST / SCL].

### Solución:

La solución se encuentra en el proyecto del Portal TIA "Fertigungslinie\_06\_Drehtisch\_SCL.zap17".