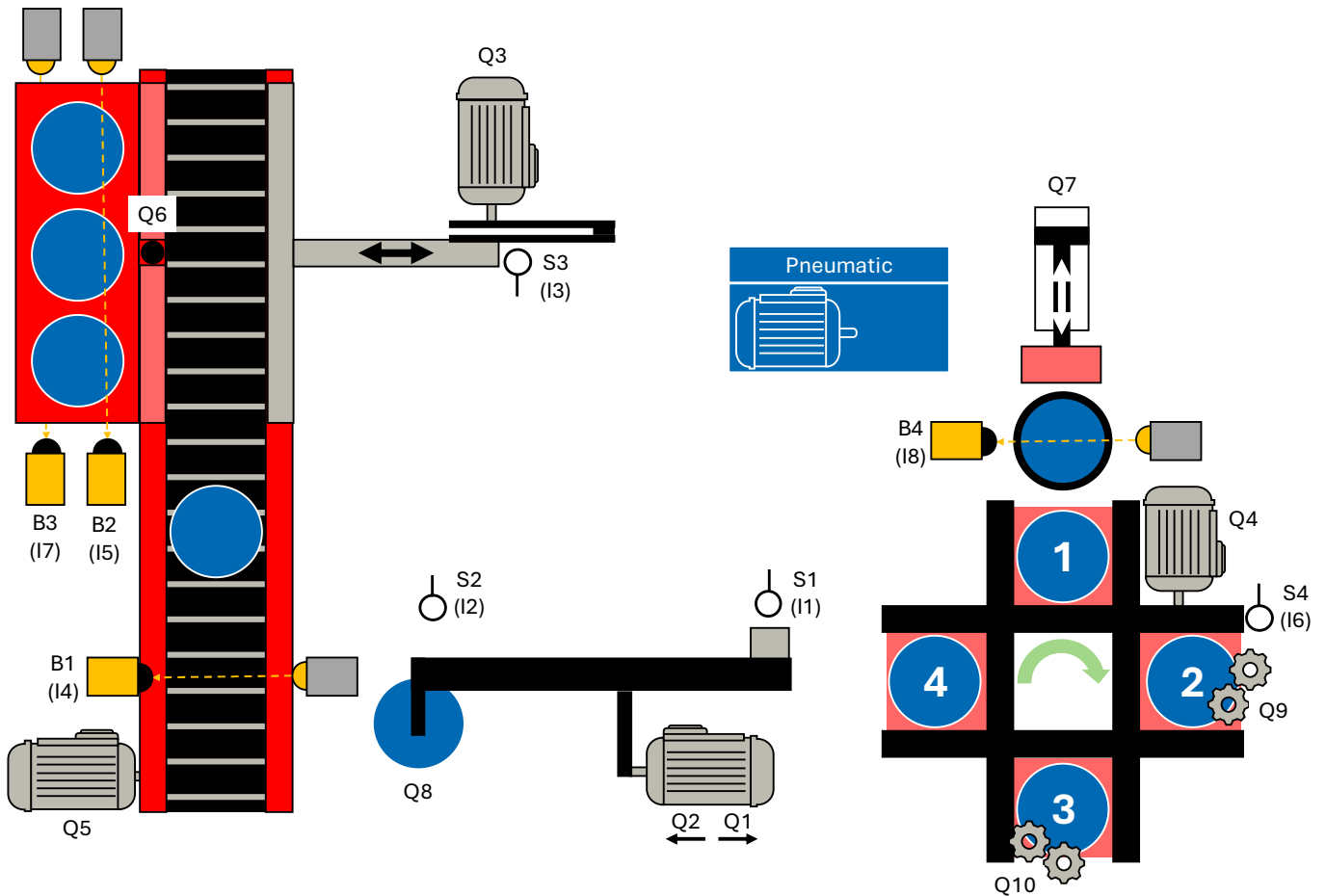


# 24V production line

Planning and implementing automatic processes



## Table of contents

10	Plan and implement automatic sequence.....	1
10.1	Exercise: Planning the process chain with GRAFCET - transport route.....	1
10.2	Exercise: Planning the sequence chain with GRAFCET - converter.....	5
10.3	Exercise: Planning the sequence chain with GRAFCET - Magazine.....	9
10.4	Exercise: Planning the sequence chain with GRAFCET - Processing station 12	
10.5	Exercise: Convert GRAFCET sequence chain into program code [FBD] - Transport route .....	15
10.6	Exercise: Converting the GRAFCET sequence chain into program code [ST / SCL] - Transport route.....	23
10.7	Exercise: Convert GRAFCET sequence chain into program code [FBD] - Converter.....	31
10.8	Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Converter.....	39
10.9	Exercise: Convert GRAFCET sequence chain into program code [FUP] - Magazine.....	46
10.10	Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Magazine.....	53
10.11	Exercise: Convert GRAFCET sequence chain into program code [FUP] - Processing station.....	60
10.12	Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Processing station.....	67
10.13	Exercise: Create function block for rotary table [FBD].....	74
10.14	Exercise: Create function block for rotary table [ST / SCL].....	81

## 10 Planning and implementing automatic processes



### 10.1 Exercise: Planning the process chain with GRAFCET - transport route

#### Target:

I can use the functional description to create a GRAFCET sequence chain for the transport section of the production line.

#### Task:

Create a sequence chain in GRAFCET for the transport route of the production line so that the function is implemented based on the system description.

#### Function:

##### 1. Initial step

No actions are carried out in the initial step.

The step chain remains in this step until an empty pallet is inserted. The pallet interrupts the light barrier B3. B2 must not be interrupted.

##### 2. Move slider to home position

After an empty pallet has been inserted, the slider (Q3) can be moved to the home position. The slider may only be activated if it is not in the home position and a pallet is in front of it.

If the pusher is activated, the counter for the workpieces on the belt must also be reset.

If the slider is in the home position (S3 actuated), you can switch to the next step.

##### 3. Close divider

Now that the slider is in the home position, the separator can be closed. The valve (Q6) must be permanently activated to keep the separator closed.

500 milliseconds after the valve has been activated, it can be assumed that the separator is closed. This means you can switch to the next step.

##### 4. Waiting for workpiece

In this step, the transfer unit remains in place until a new workpiece has been placed on the belt; no actions are performed. A new workpiece is detected by interrupting light barrier B1. Before switching to the next step, the transfer unit must have left the belt again (S1 not actuated).

##### 5. Removing the workpiece

If a workpiece is on the belt and the transfer unit has left the station, it can be transported away. The belt (Q5) may only be switched on if the slider is in the home position (S3 actuated), the separator is closed (Q6 actuated) and the transfer unit is not in the area of the belt (S1).

The conveyor belt must run for at least 3 seconds before changing to the next step.

##### 6. Increase counter

After the workpiece has reached the end of the belt, the corresponding counter must be increased by one.

If there are already 3 workpieces on the belt, ejection can begin.

If the number has not yet been reached, you must return to step 2.

**7. Open divider**

Once there are 3 workpieces on the conveyor belt, they can be pushed onto a pallet. To do this, first open the divider (Q6).

500 milliseconds after the valve is no longer actuated, it can be assumed that the separator is fully open. To switch to the next step, an empty pallet must also be available.

**8. Push workpiece onto pallet**

If a pallet is ready (B3 interrupted) and the divider is open, the workpieces can be pushed onto it.

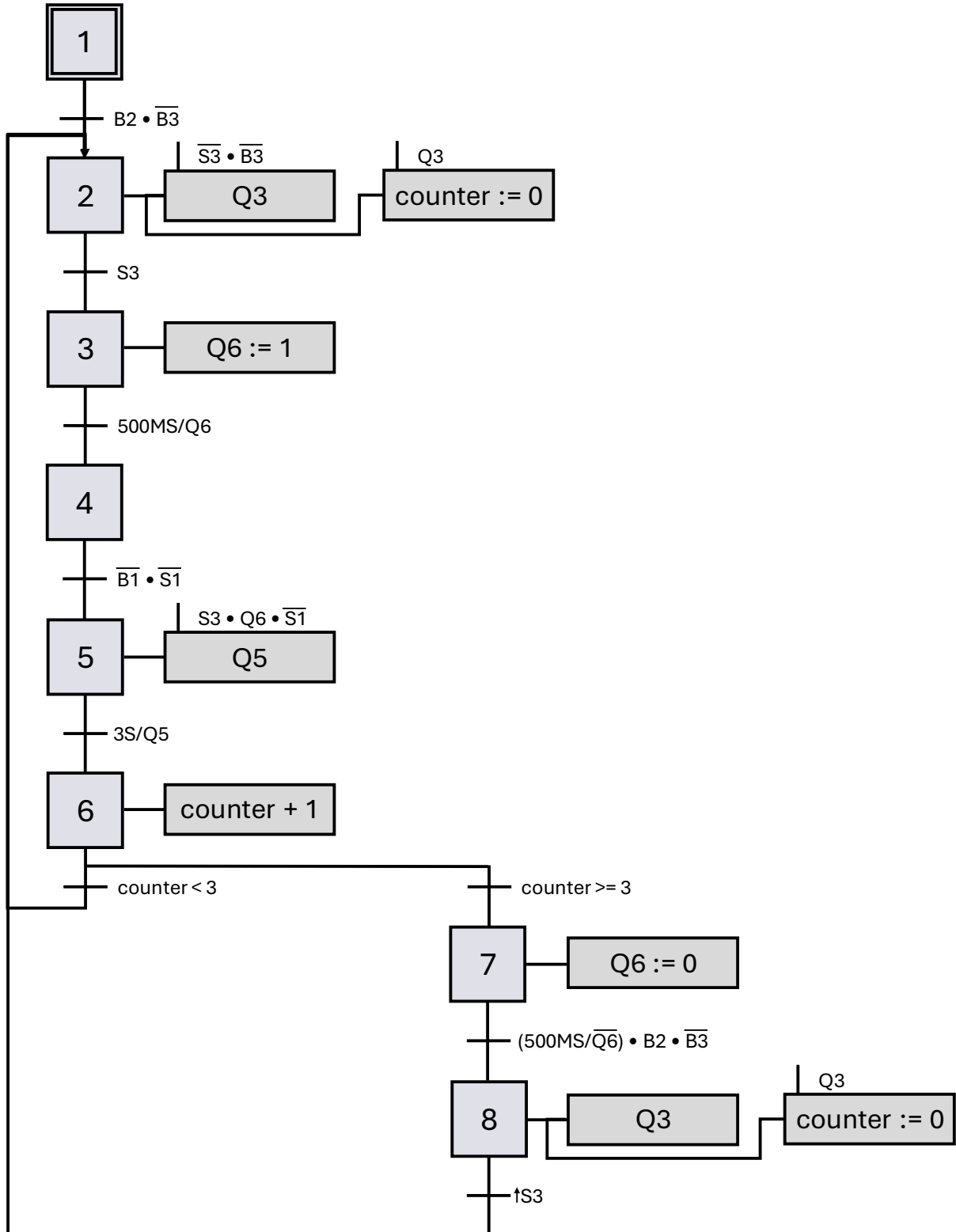
If the pusher is activated, the counter for the workpieces on the belt must also be reset.

If the slider reaches its home position again (positive edge S3), you can jump back to step 2.



Solution

Solution:





## 10.2 Exercise: Planning the sequence chain with GRAFCET - converter

### Target:

I can use the functional description to create a GRAFCET sequence chain for the production line converter.

### Task:

Create a sequence chain in GRAFCET for the converter of the production line so that the function is given based on the system description.

### Function:

#### 1. Initial step

In the initial step, the converter is moved to the home position. The home position is defined in such a way that neither of the two end positions is reached.

If the transfer unit is in end position S1, it must be moved in the direction of the turntable (Q1) until it has left this position.

If the transfer unit is in the end position S2, it must be moved in the direction of the conveyor belt (Q2) until it has left this position.

Once the converter has reached the home position, you can switch to the next step.

#### 2. Waiting for workpiece

No actions are performed in this step. The system waits until the turntable provides a finished workpiece that can be picked up.

If the turntable is in position (S4 actuated) and a workpiece is ready to be picked up (wstReady), the system switches to the next step.

#### 3. Move the transfer unit towards the turntable

The converter is moved in the direction of the turntable by activating Q1. The action is only executed as long as the end position (S2) has not been reached and the turntable is in position (S4).

If the transfer unit is on the side of the turntable, you can switch to the next step.

#### 4. Switch on vacuum

The vacuum is switched on by activating Q8. If the vacuum is switched on, the information that a workpiece is ready on the rotary table must be reset (wstReady := 0).

500 milliseconds after switching on, it can be assumed that this has built up and can therefore be switched to the next step.

#### 5. Move the converter towards the belt

The transfer unit is moved in the direction of the turntable by activating Q2. The action is only executed as long as the end position (S1) has not been reached and the deposit position on the conveyor belt is free (B1) and the conveyor belt is not running (Q5).

Once the converter has reached the end position, you can switch to the next step.

**6. Switch off the vacuum**

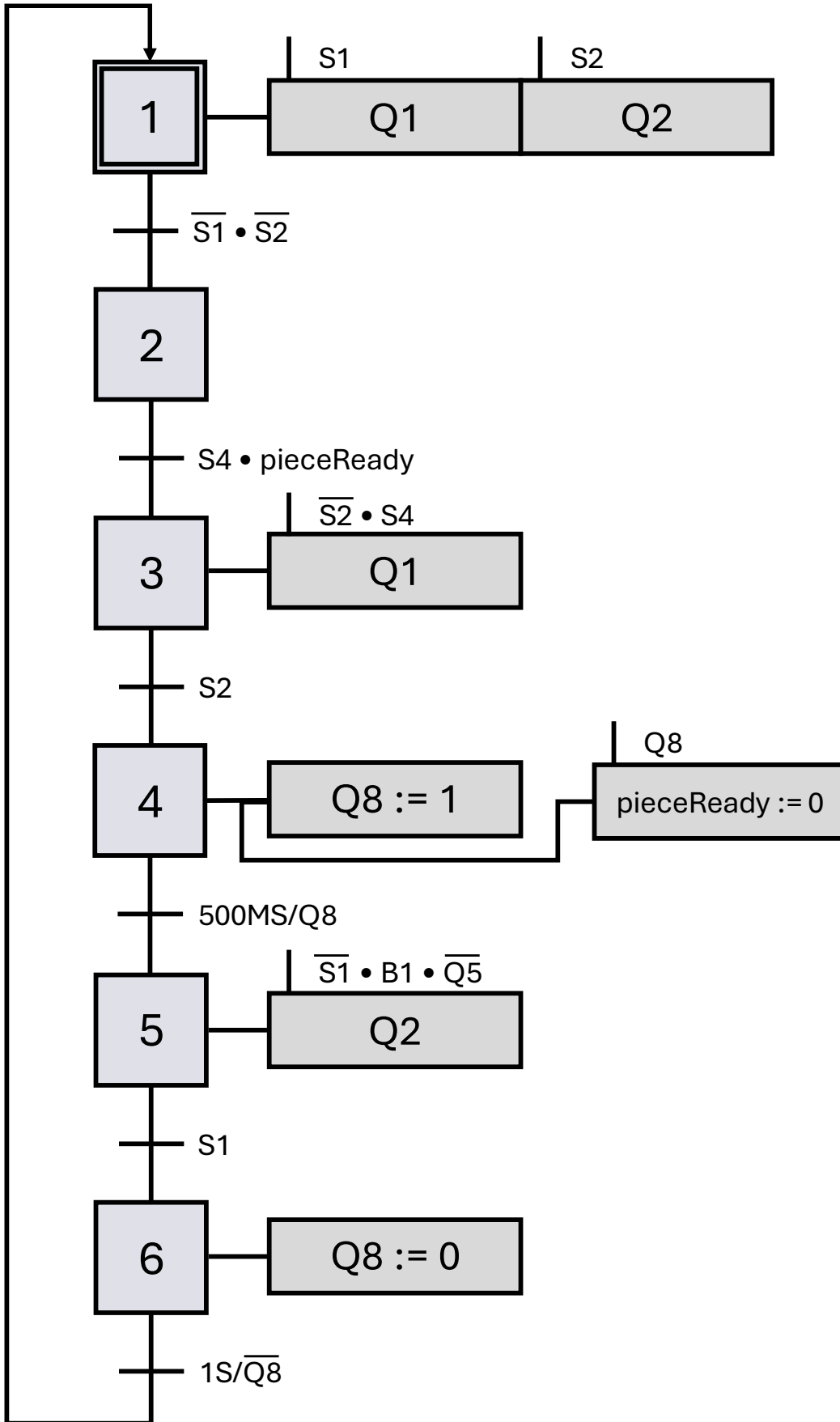
The vacuum is switched off by resetting Q8. 1 second after switching off, it can be assumed that the vacuum has also dissipated and you can therefore jump back to the initial step.





Solution

Solution:





## 10.3 Exercise: Planning the process chain with GRAFCET - Magazine

### Target:

I can create a GRAFCET sequence chain for the magazine on the rotary table of the production line using the functional description.

### Task:

Create a sequence chain in GRAFCET for the magazine of the production line so that the function is given based on the system description.

### Function:

#### 1. Initial step

No actions are carried out in the initial step.

If the turntable is in position (S4 activated), workpieces are in the magazine (B4 interrupted) and the space (nest) on the turntable is empty ("nestBelegt" = "FALSE"), you can jump to the next step.



It is advisable to delay the signal from light barrier B4 by approx. 1 second, as otherwise the pusher could extend as soon as the magazine is filled, which could lead to the workpieces becoming wedged.

#### 2. Extend slider

By activating Q7, the slider is extended and a workpiece is pushed out of the magazine onto the turntable. The slider may only be extended when the turntable is in position (S6).

1 second after the valve is activated, it can be assumed that the slider is fully extended. You can jump to the next step.

#### 3. Set nest occupied

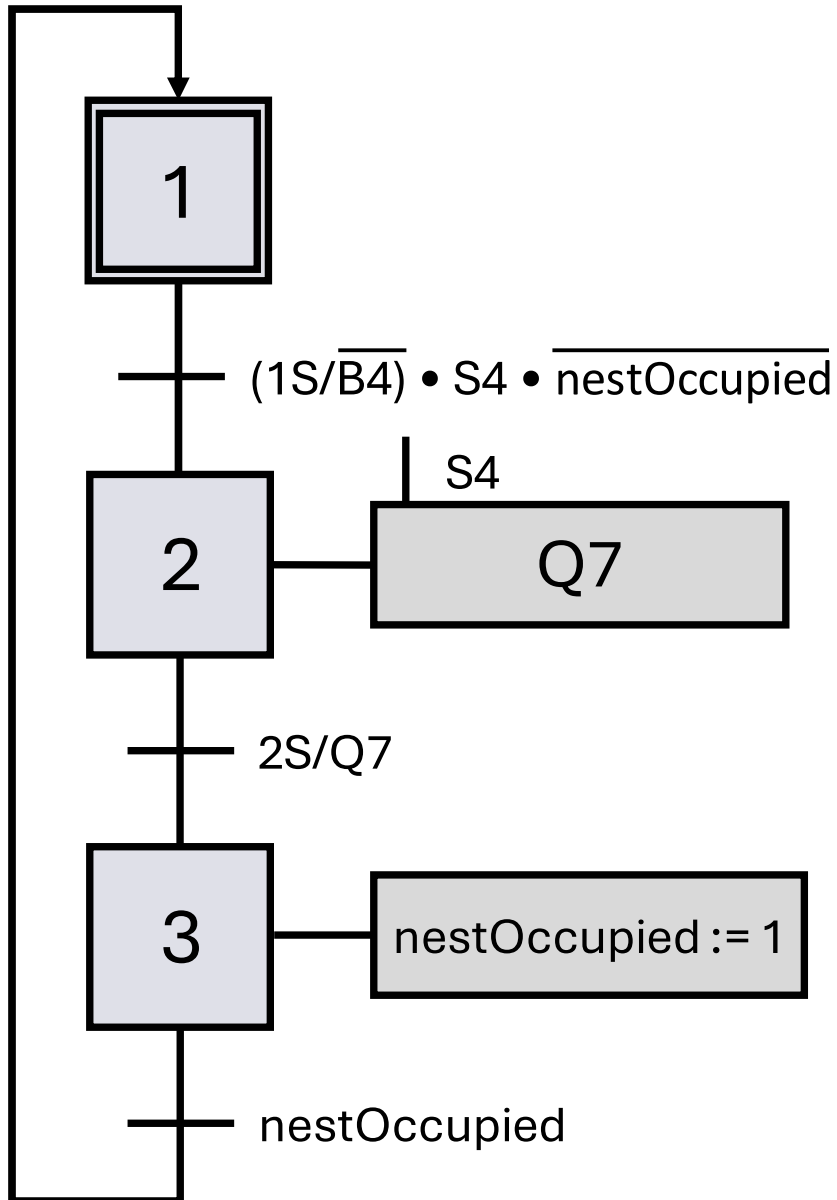
As there is now a workpiece on the turntable, the nest must be written as occupied. To do this, the "nest occupied" bit must be set to "TRUE".

If the "nestOccupied" bit has the value "TRUE", you can jump to the initial step.



Solution

Solution:





## 10.4 Exercise: Planning the process chain with GRAFCET - Processing station

### Target:

I can use the functional description to create a GRAFCET sequence chain for a processing station on the rotary table of the production line.

### Task:

Create a sequence chain in GRAFCET for a processing station of the production line so that the function is given based on the system description.

Both the "Drilling" station and the "Welding" station have the same sequence. It is therefore only necessary to create a general sequence chain that is valid for both stations.

### Function:



The process is started when a raw part is in the station and runs for a defined period of time. Instead of directly triggering the output for drilling (Q9) or welding (Q10), the term "machining" is generally used. No defined time is assumed as the transition, but "time" is generally assumed. After the work step, the "raw part" variable must be reset and the "finished part" variable set.

#### 1. Initial step

No actions are carried out in the initial step.

If the turntable is in position (S4 activated) and a blank is in the station, you can jump to the next step.

#### 2. Editing

The workpiece is processed in this step. The "process" output must be activated.

If the output is active for the defined "time", you can jump to the next step.

#### 3. Set precast element

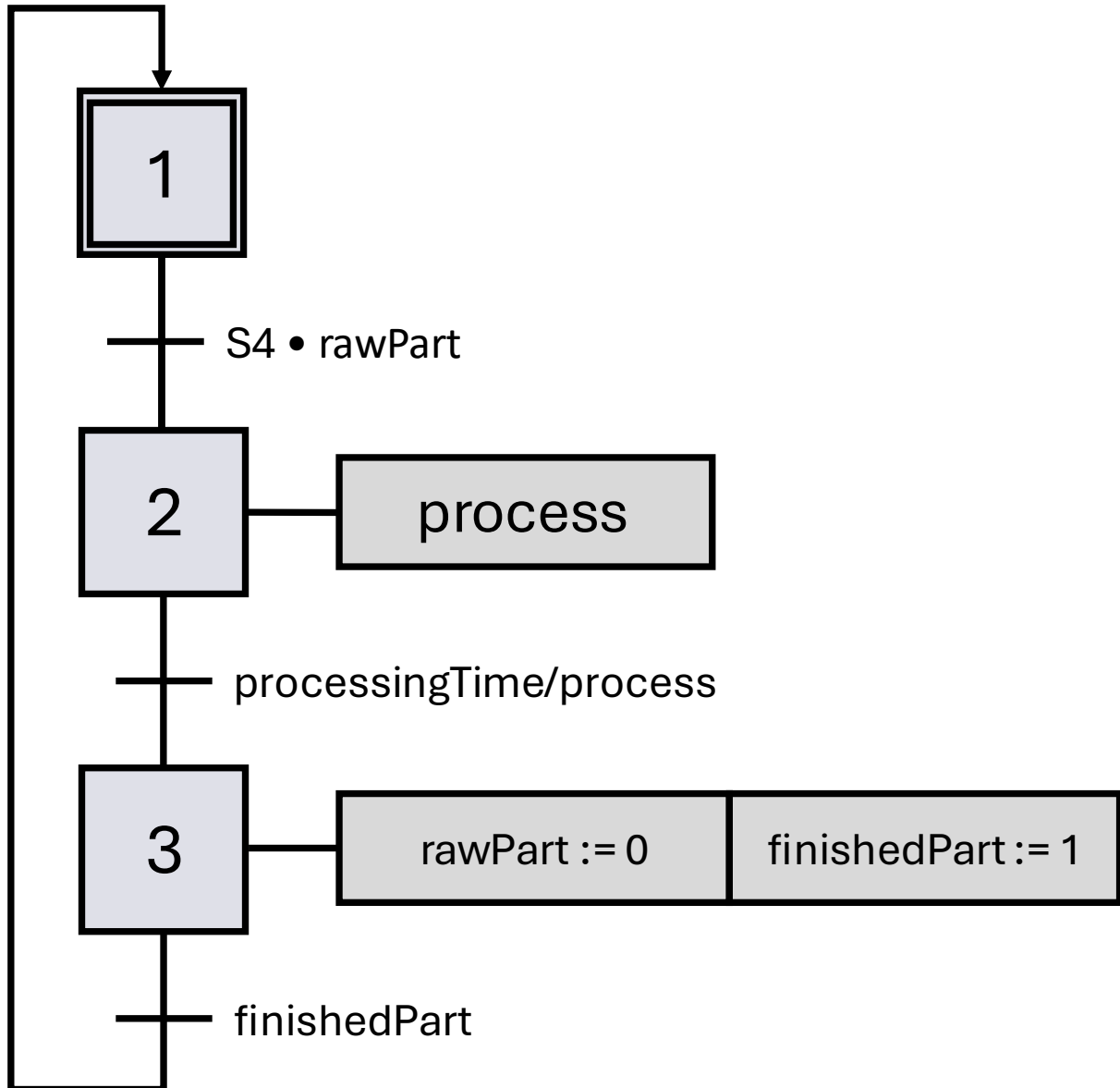
The "Unmachined part" variable must be reset. The "finished part" variable must be set.

If "Finished part" is set, you can jump back to the initial step.



Solution

Solution:







## 10.5 Exercise: Convert GRAFCET sequence chain into program code [FBD] - Transport route

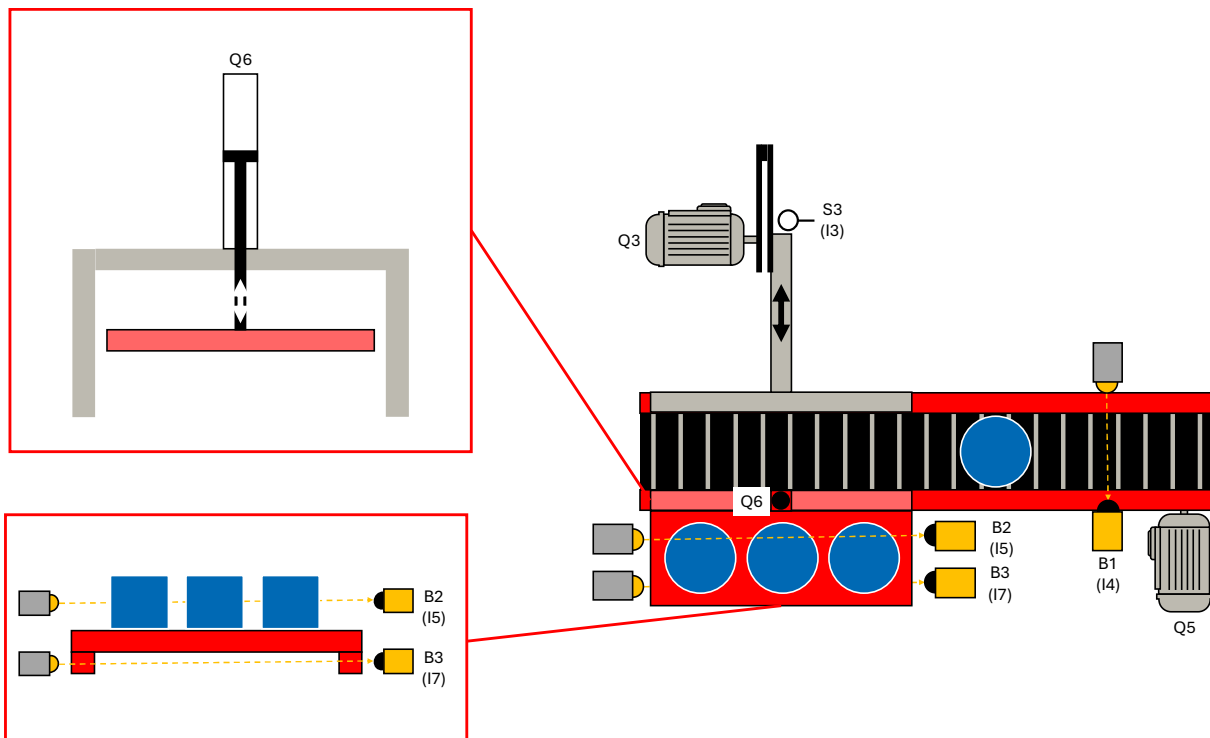
Target:

I can create the automatic program for the transport section of the production line based on the functional description and the sequence chain created in GRAFCET.

Task:

Create the PLC program for the automatic sequence so that the function is implemented based on the system description.

The chain is initialized when the control unit is switched on.

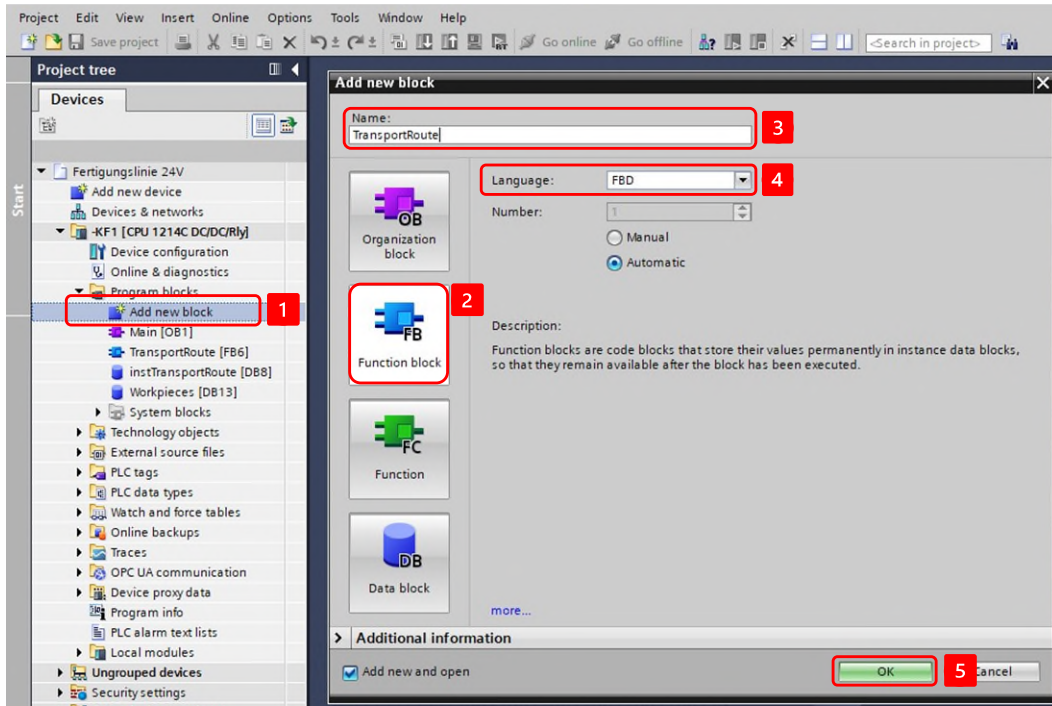


Picture 1 System diagram - transport route

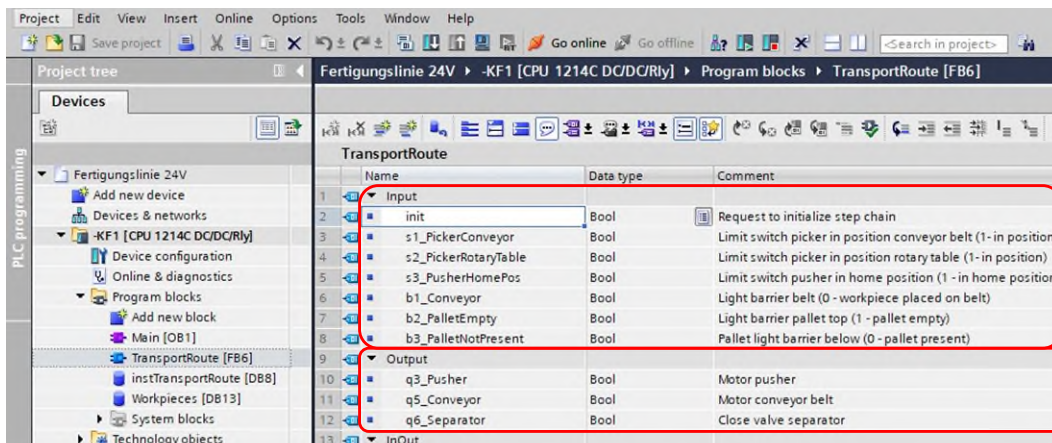
## Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FBD] - Transport route

### Procedure:

1. Create a new function block, select the desired programming language and assign a meaningful name:

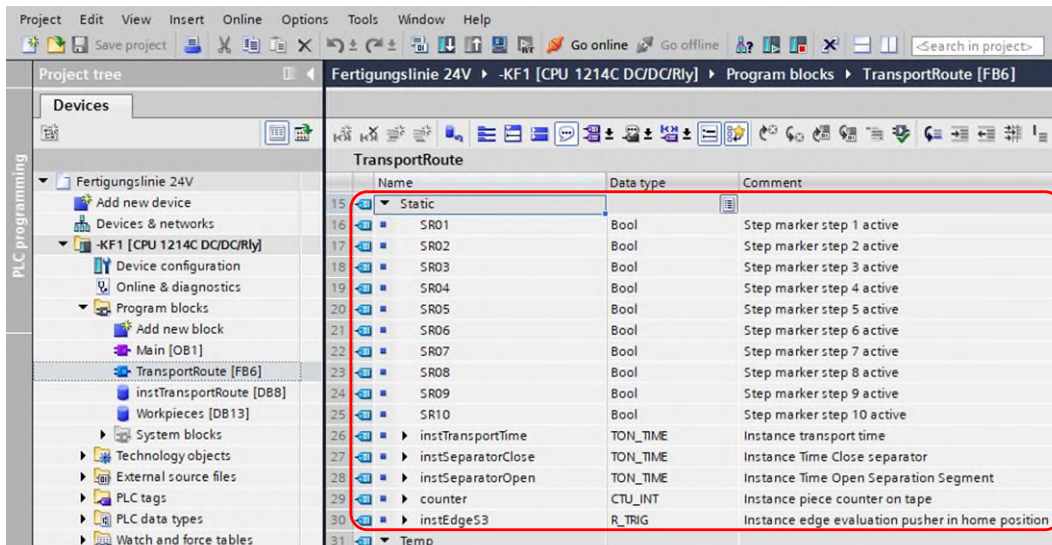


2. Declare variables for the sensors and actuators, as well as a variable for initializing the step chain in the function block interface:

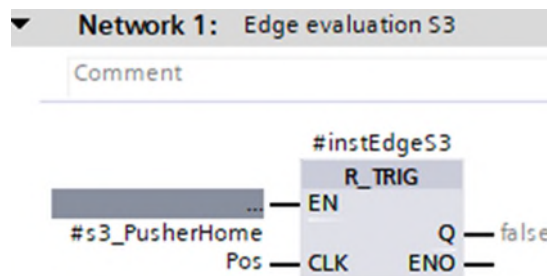


Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FBD] - Transport route

3. Declare the step flags in the static area of the function block interface, an instance for the edge evaluation of the limit position switch S3, instances for the delay times and an instance for the workpiece counter:



4. Program the edge evaluation for the limit position switch S3 in the first network:



5. In the following networks, implement the individual steps using flip-flops based on the GRAFCET. A new network must be used for each step:



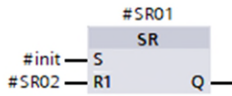
Step 1 is the initial step. In the automatic sequence, the system jumps back from step 6 to step 2 if the count has not yet been reached. If workpieces have to be ejected, the system jumps back to step 2 after step 8.

Step 6 is reset by the following step, or by step 2 in the case of a return.

Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FBD] - Transport route

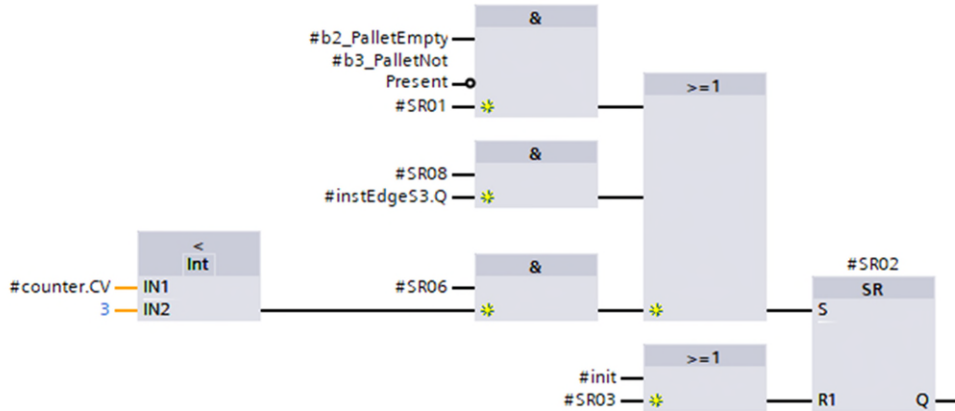
**Network 2: Step 1 - Initial step**

Comment



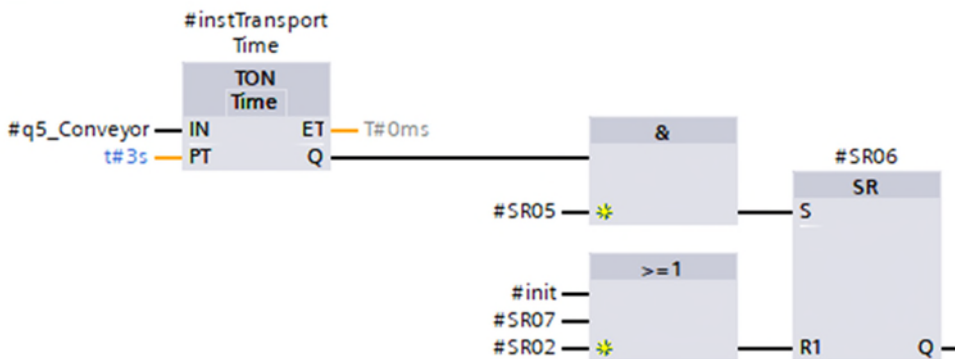
**Network 3: Step 2 - Move pusher to home position**

Comment



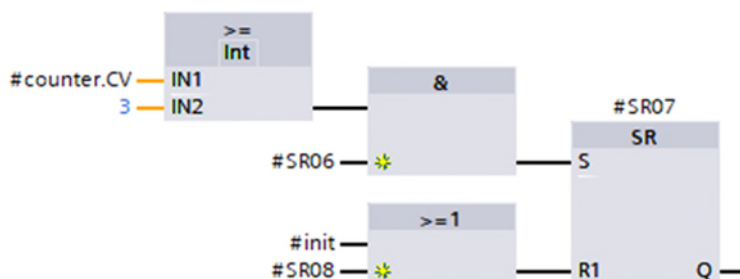
**Network 7: Step 6 - Increase counter**

Comment



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

Comment

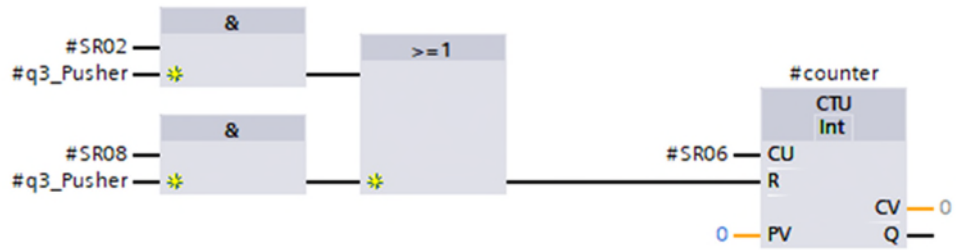


Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FBD] - Transport route

6. Assign the actions below the step chain in the next networks:

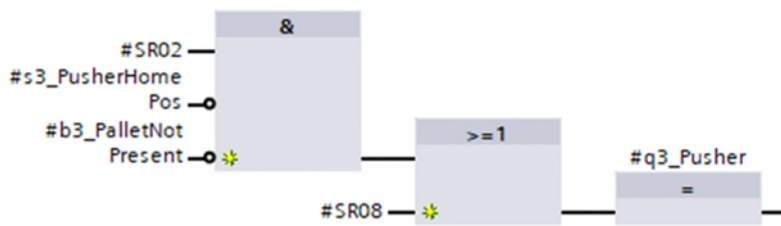
**Network 10:** counter

Comment



**Network 11:** Motor pusher

Comment



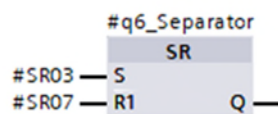
**Network 12:** Motor conveyor belt

Comment



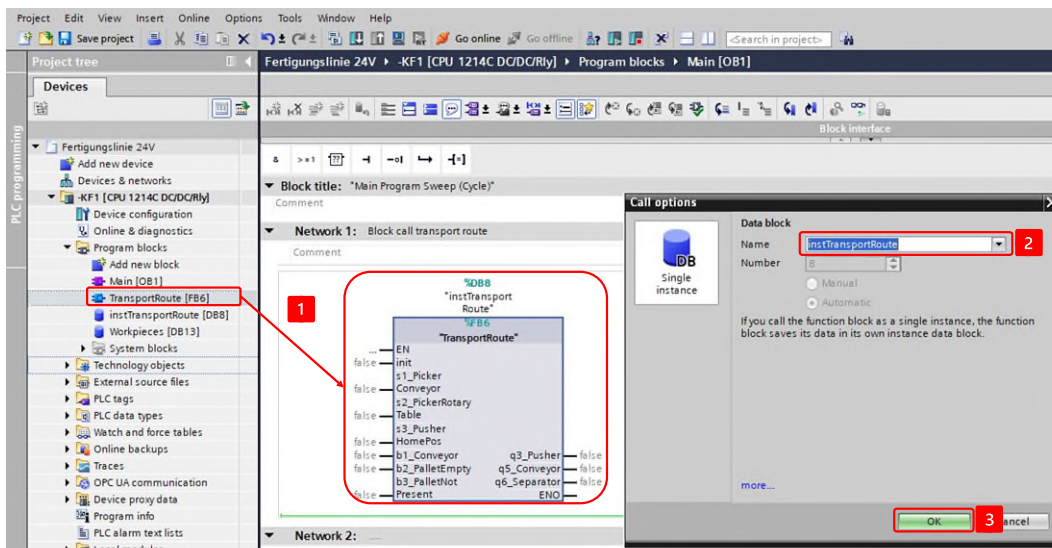
**Network 13:** Close valve separator

Comment

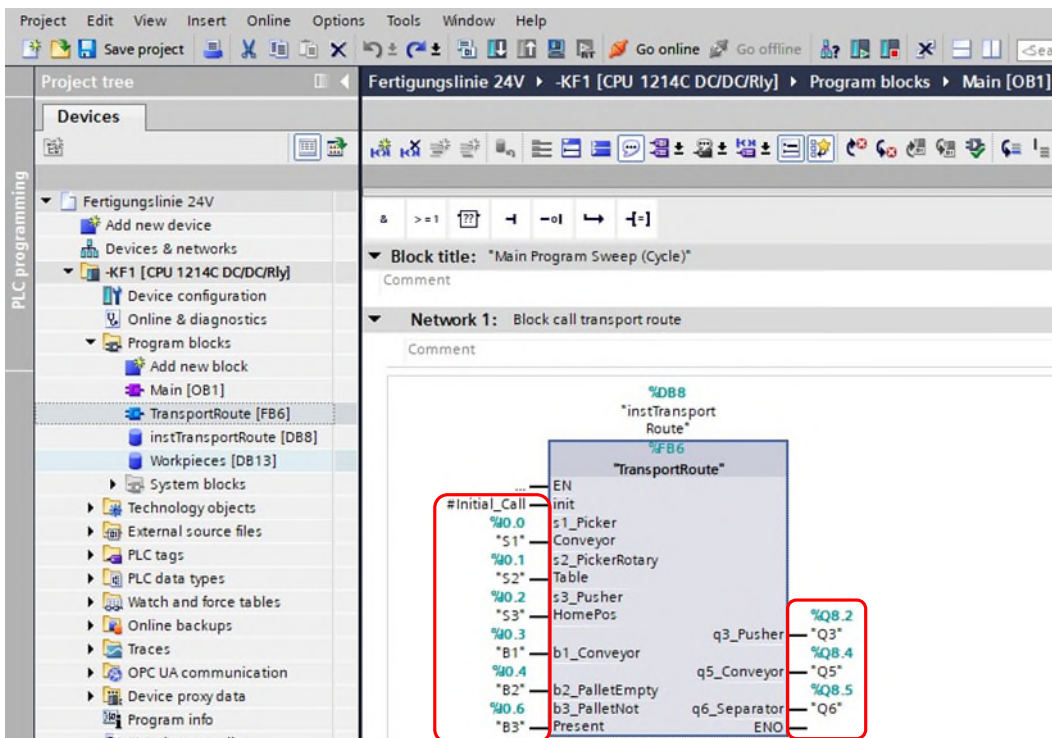


Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FBD] - Transport route

7. Call up the function module in "MAIN" and create an instance:



8. Connect the function block interface with the input and output variables from your variable table:



The "Initial\_Call" system bit provided by Siemens is used as the initialization request. This is "TRUE" when the MAIN is run through for the first time.

9. Commission the system in a structured manner with the aid of a commissioning protocol.



Solution

Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FBD] - Transport route

**Solution:**

The solution can be found in the TIA Portal project "Fertigungslinie\_02\_Transportstrecke\_FUP.zap17".





## 10.6 Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Transport route

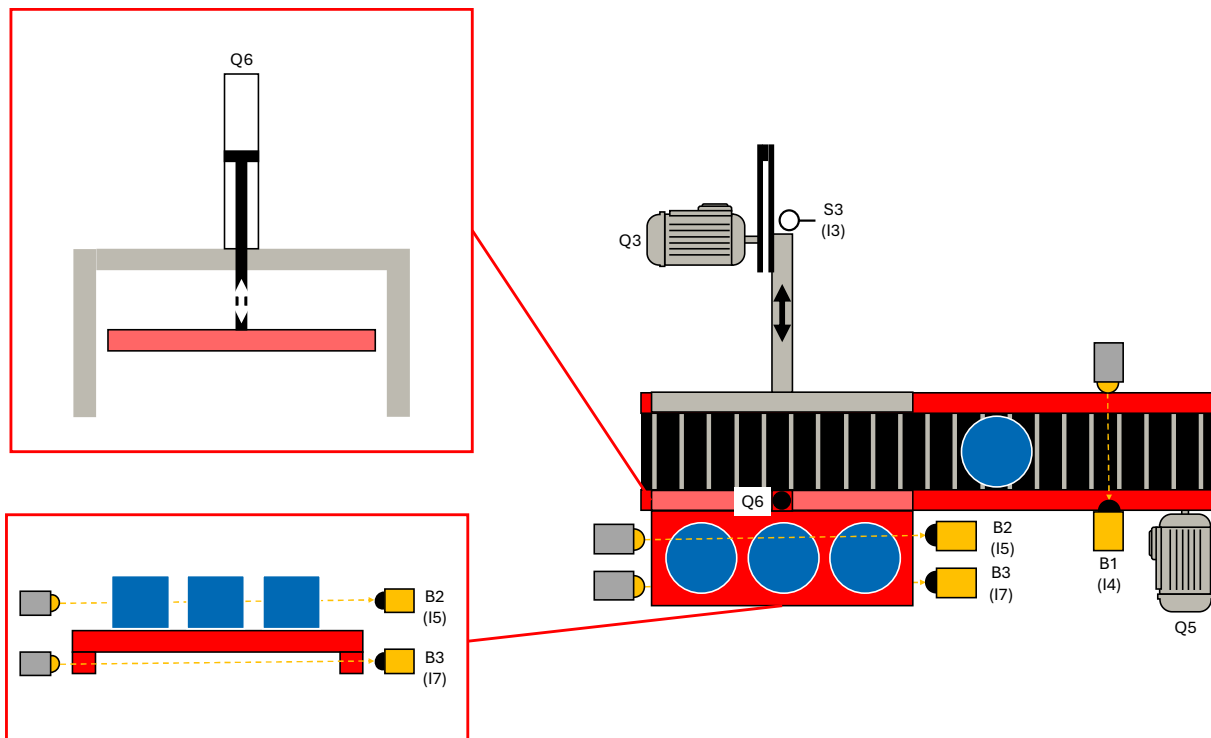
Target:

I can create the automatic program for the transport section of the production line using the functional description and the sequence chain created in GRAFCET.

Task:

Create the PLC program for the automatic sequence so that the function is implemented based on the system description.

The chain is initialized when the control unit is switched on.

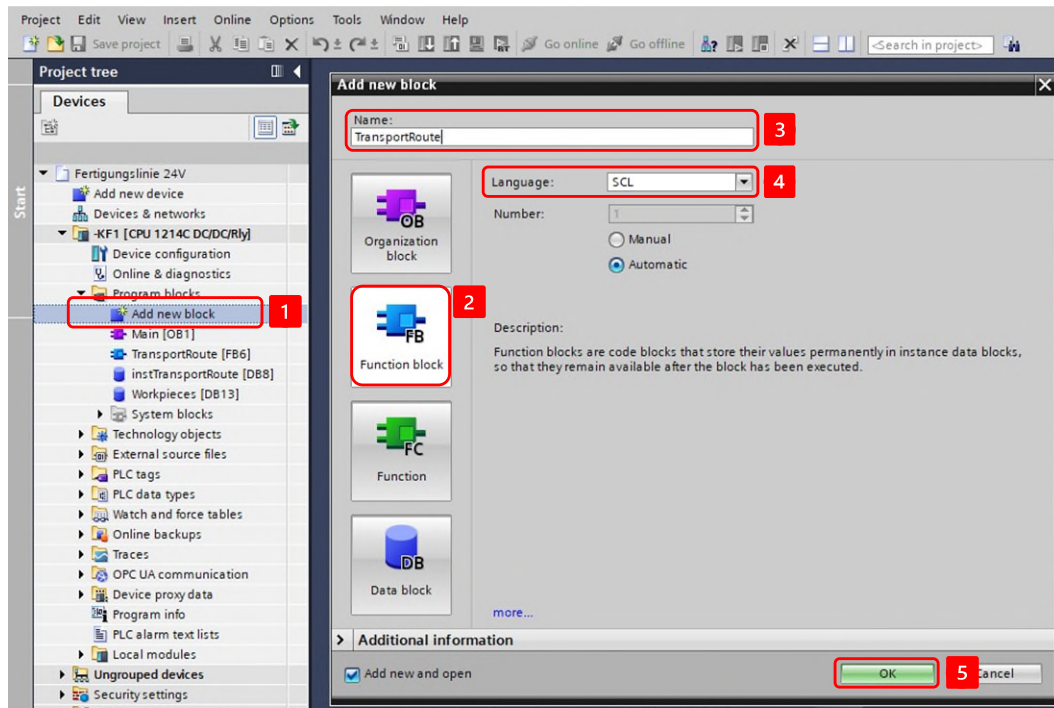


Picture 2 System diagram - transport route

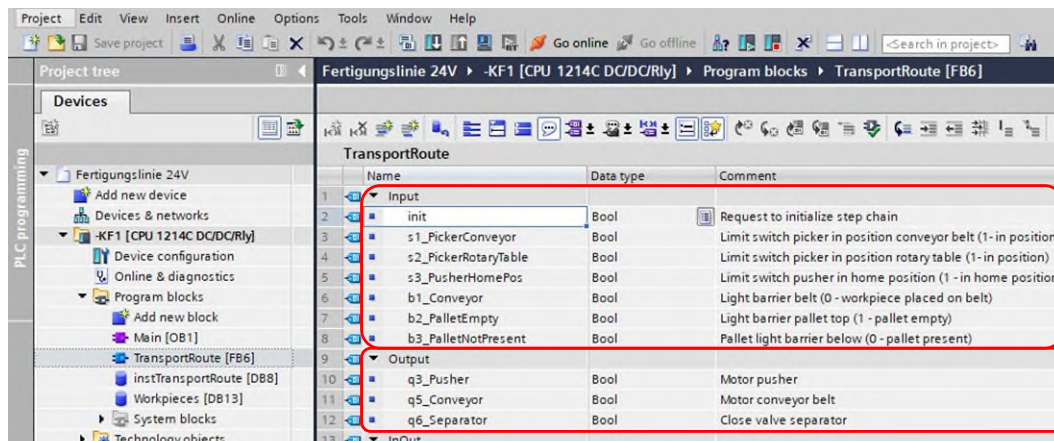
## Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Transport route

### Procedure:

1. Create a new function block, select the desired programming language and assign a meaningful name:

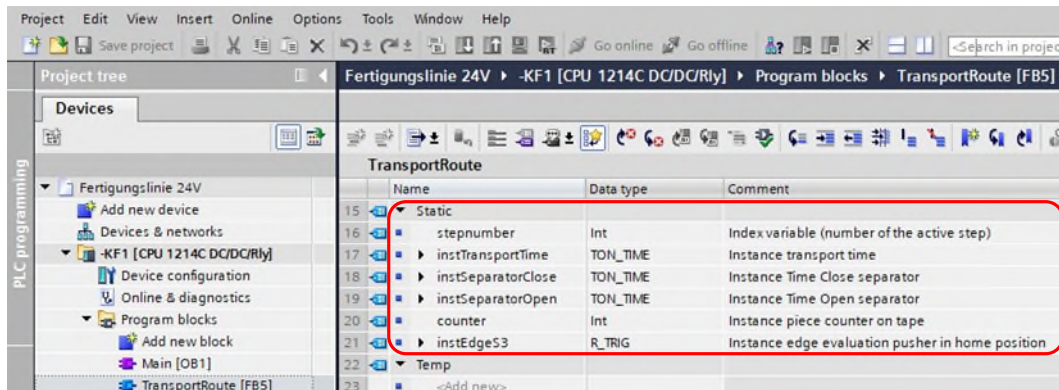


2. Declare variables for the sensors and actuators, as well as a variable for initializing the step chain in the function block interface:



Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Transport route

3. Declare the index variable in the static area of the function block interface, an instance for the edge evaluation of the limit position switch S3, instances for the delay times and a variable for the counter of the workpieces:



4. Program the initialization of the chain:

```

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

```

5. Program the edge evaluation for the limit position switch S3:

```

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

```

6. Implement the individual steps from the GRAFCET in the following CASE structure. A new CASE must be created in the structure for each step, which represents the step number:



Step 1 is the initial step. In the automatic sequence, the system jumps back from step 6 to step 2 if the count has not yet been reached. If workpieces have to be ejected, the system jumps back to step 2 after step 8.



All actions are reset in the initial step. This ensures that no actions remain set if an active step chain is aborted by an initialization request.

Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Transport route

```

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;

```

Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Transport route

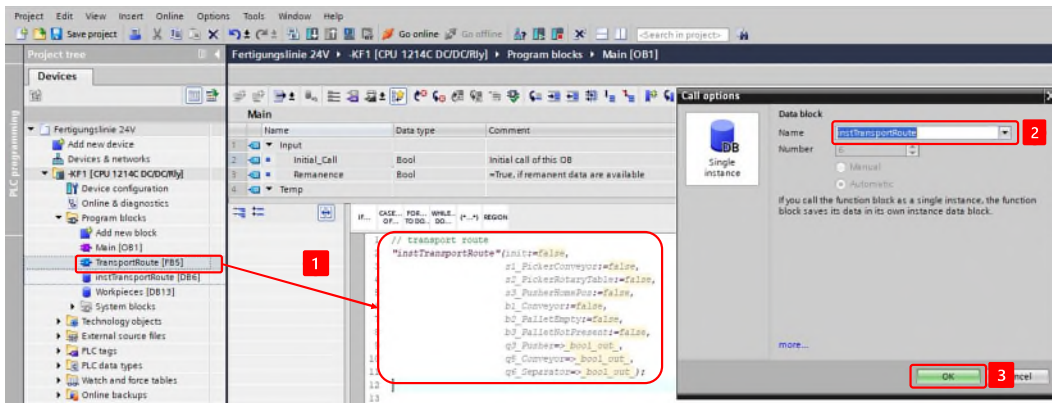
7. Program the time functions according to the CASE structure:

```

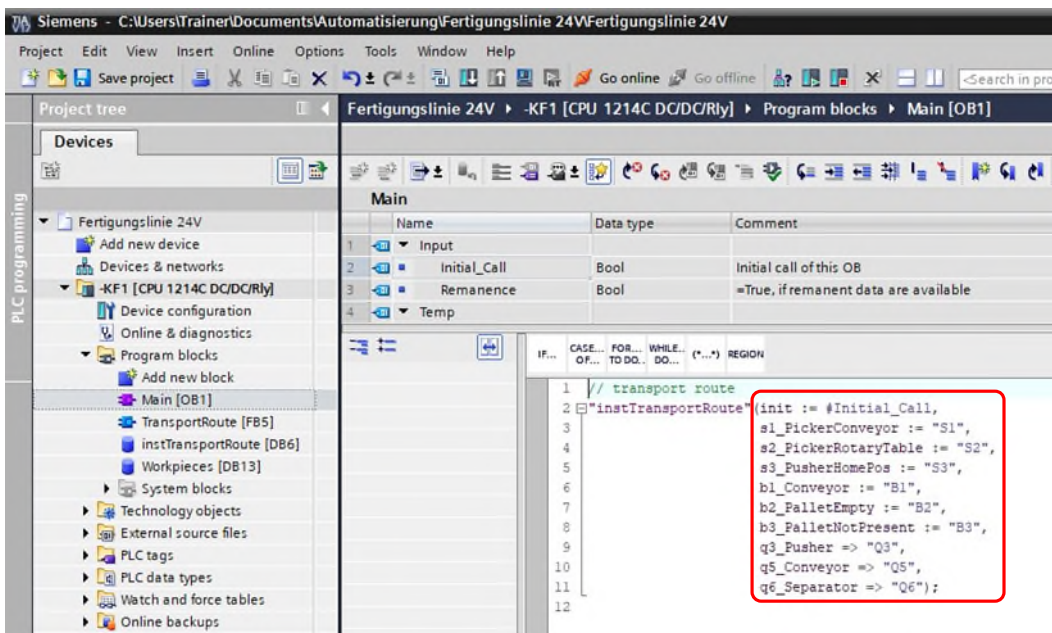
106 // timer
107 // transport time
108 #instTransportTime(IN := #q5_Conveyor,
109 | PT := t#3s);
110
111 // Time Close separator
112 #instSeparatorClose(IN := #q6_Separator,
113 | PT := t#500ms);
114 // Time Open separator
115 #instSeparatorOpen(IN := NOT #q6_Separator,
116 | PT := t#500ms);

```

8. Call up the function module in "MAIN" and create an instance:



9. Connect the function block interface with the input and output variables from your variable table:



Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Transport route



The "Initial\_Call" system bit provided by Siemens is used as the initialization request. This is "TRUE" when the MAIN is run through for the first time.

10. Commission the system in a structured manner with the aid of a commissioning protocol.



Solution

Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Transport route

**Solution:**

The solution can be found in the TIA Portal project "Fertigungslinie\_02\_Transportstrecke\_SCL.zap17".





## 10.7 Exercise: Convert GRAFCET sequence chain into program code [FBD] - Converter

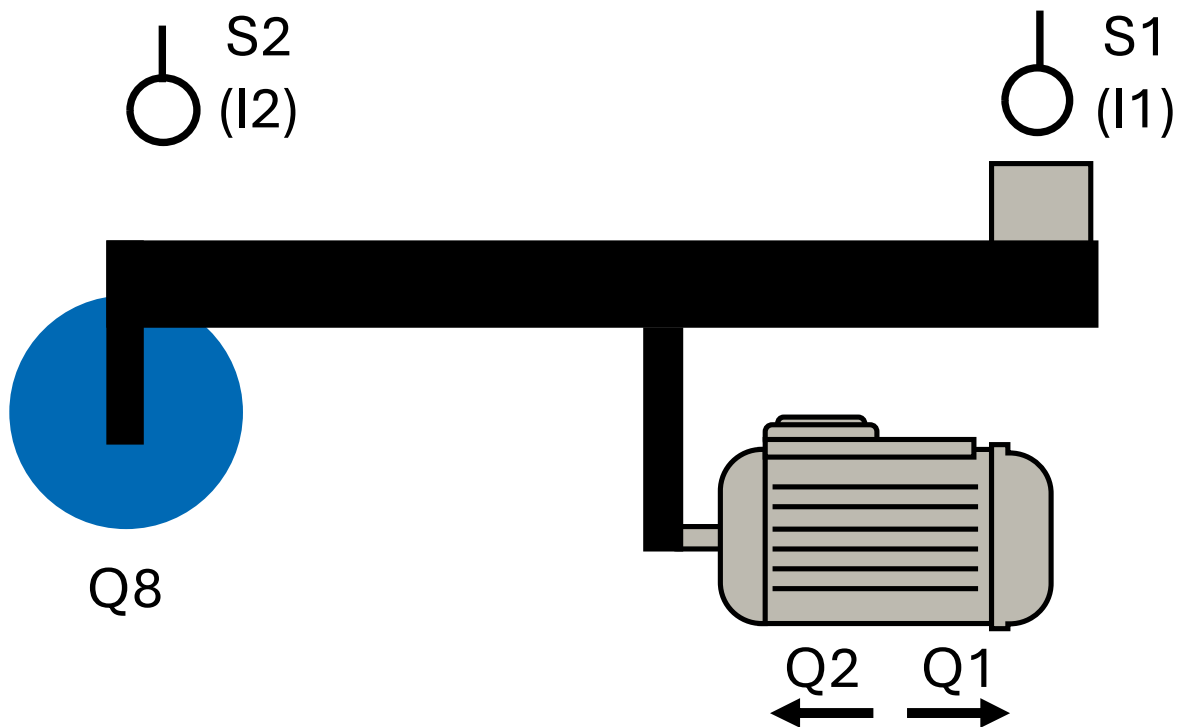
Target:

I can create the automatic program for the converter of the production line using the function description and the sequence chain created in GRAFCET.

Task:

Create the PLC program for the automatic sequence so that the function is implemented based on the system description.

The chain is initialized when the control unit is switched on.

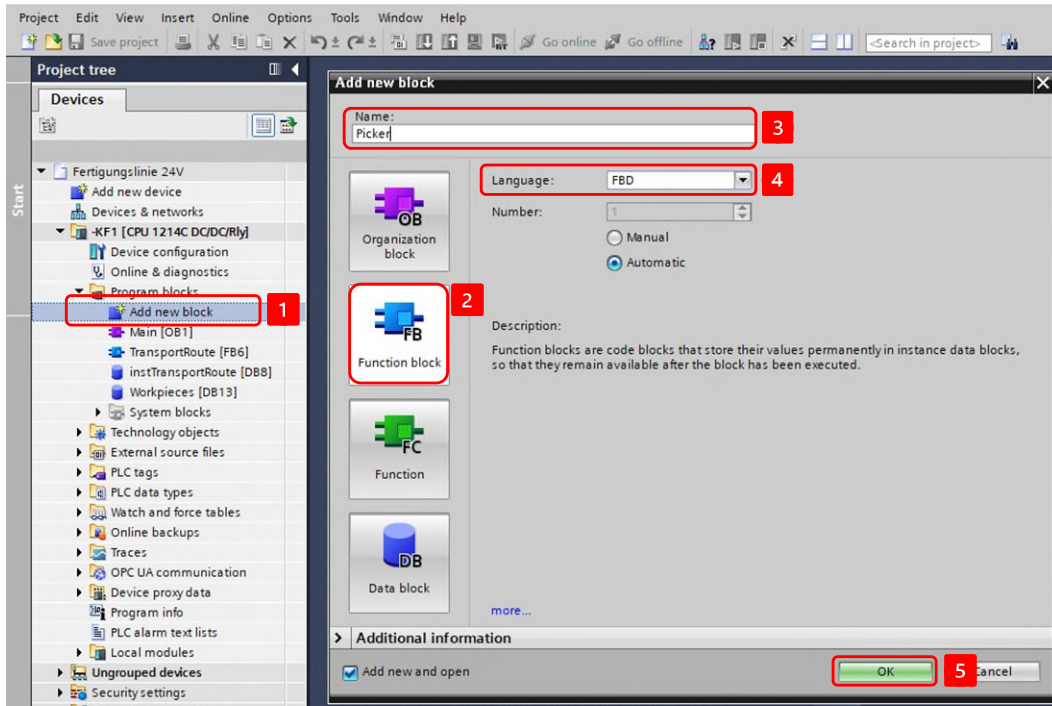


Picture 3 System diagram - converter

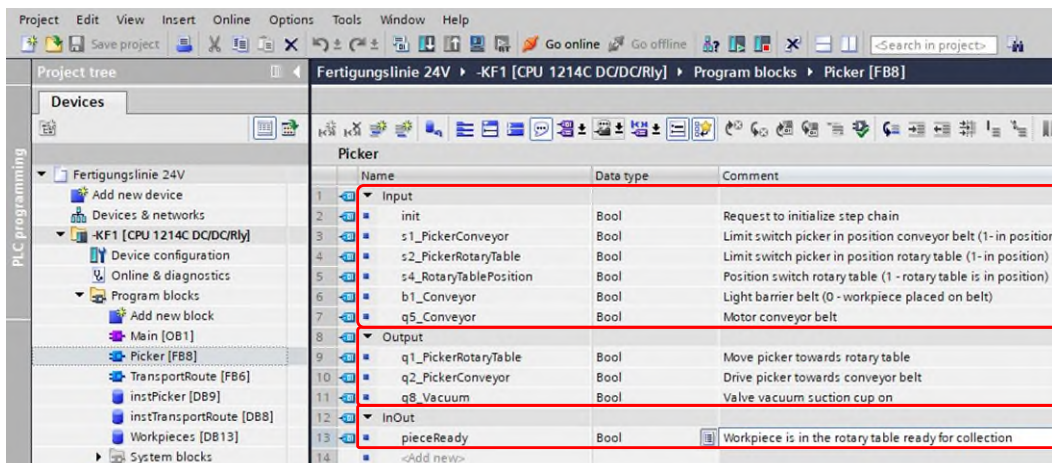
## Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FBD] - Converter

### Procedure:

1. Create a new function block, select the desired programming language and assign a meaningful name:

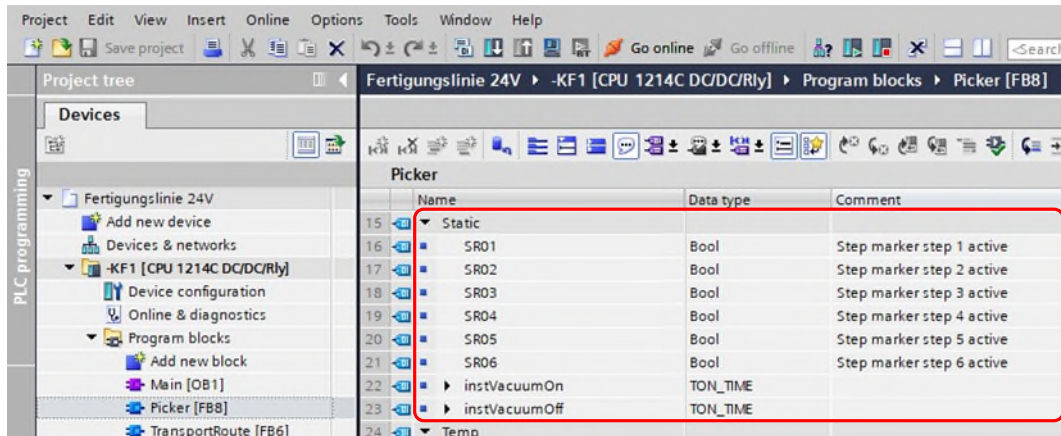


2. Declare variables for the sensors and actuators, a variable for initializing the step chain and a variable for transferring the workpiece information in the function block interface:

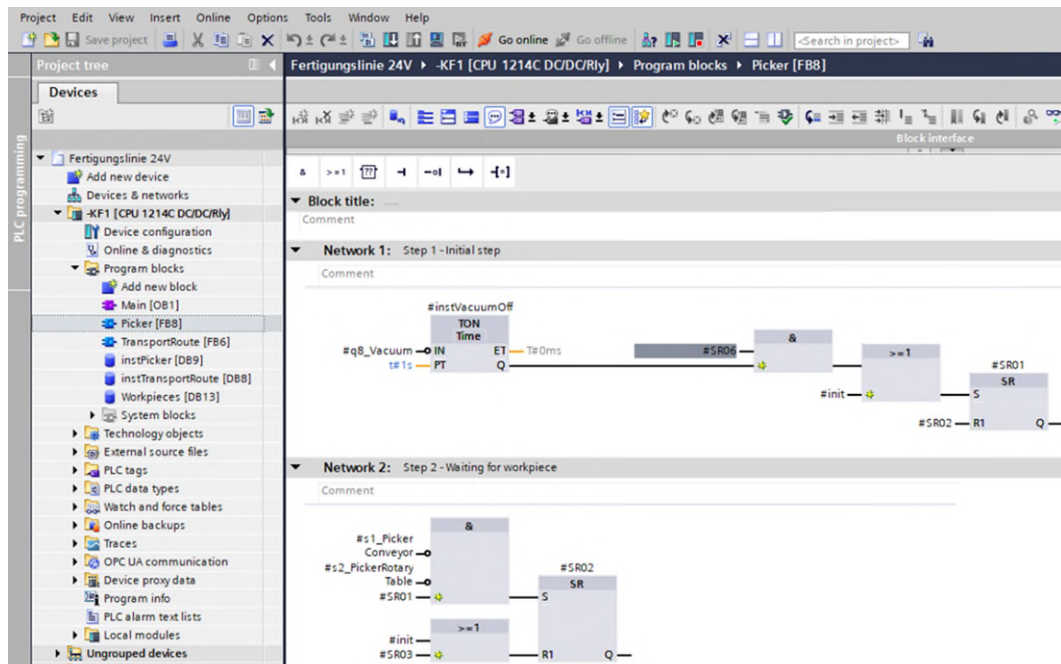


Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FBD] - Converter

3. Declare the step flags in the static area of the function block interface, as well as two instances for the delay of the vacuum signal:



4. Implement the individual steps using flip-flops based on the GRAFCET. A new network must be used for each step:



Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FBD] - Converter

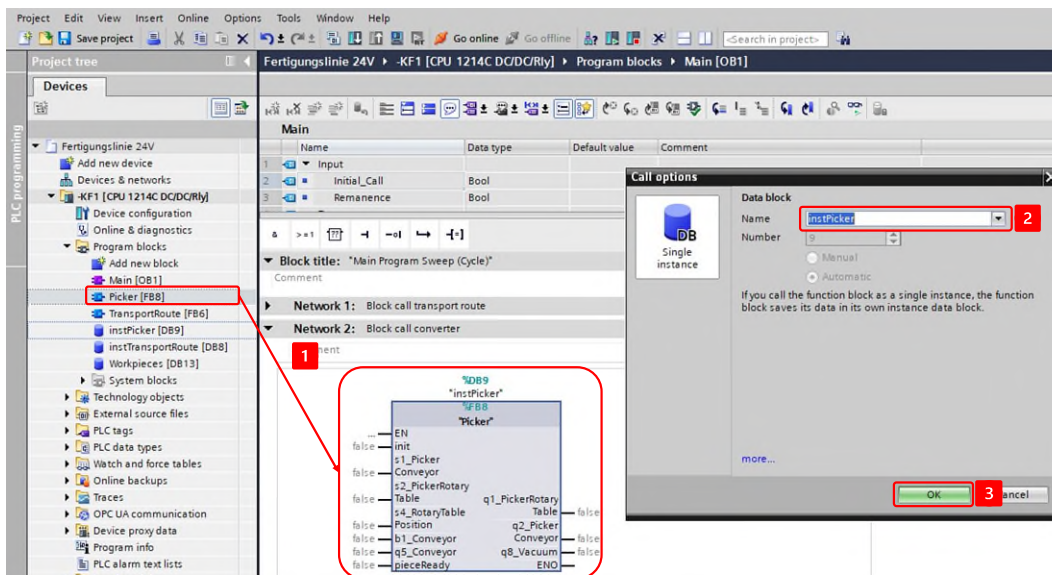
5. Assign the actions below the step chain in the next networks.

The screenshot shows the Siemens SIMATIC Manager interface for a PLC program. The project tree on the left indicates the current project is 'Fertigungsline 24V' with a sub-project '-KF1 [CPU 1214C DC/DC/Rly]'. The main workspace displays three ladder logic networks:

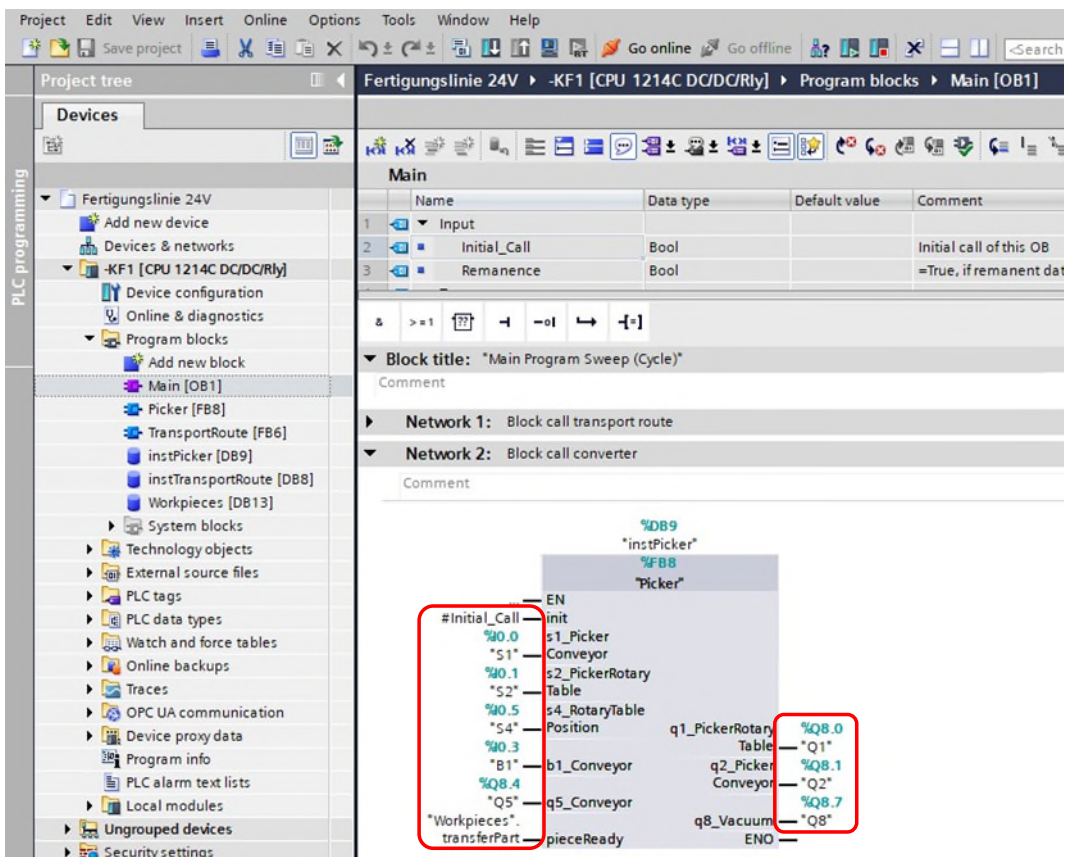
- Network 7:** Commented 'Workpiece is in the rotary table ready for collection'. It features a reset coil (R) for the variable #pieceReady, which is reset when both #q8\_Vacuum and #SR04 are active.
- Network 8:** Commented 'Move picker towards rotary table'. It features a set coil (S) for #q1\_PickerRotary Table, which is set when #s1\_Picker Conveyor is active and #s4\_RotaryTable Position is active, with a step timer >=1.
- Network 9:** Commented 'Drive picker towards conveyor belt'. It features a set coil (S) for #q2\_Picker Conveyor, which is set when #s2\_PickerRotary Table is active and #s5\_Conveyor is active, with a step timer >=1.

Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FBD] - Converter

6. Call up the function module in "MAIN" and create an instance:



7. Connect the function block interface with the input and output variables from your variable table:



The "Initial\_Call" system bit provided by Siemens is used as the initialization request. This is "TRUE" when the MAIN is run through for the first time.

Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FBD] - Converter

- Commission the system in a structured manner with the aid of a commissioning protocol.



In order for the transfer unit to pick up a workpiece from the turntable, it can be manually set to the corresponding status in the workpiece management.

	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



Solution

Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FBD] - Converter

**Solution:**

The solution can be found in the TIA Portal project "Fertigungslinie\_03\_Umsetzer\_FUP.zap17".





## 10.8 Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Converter

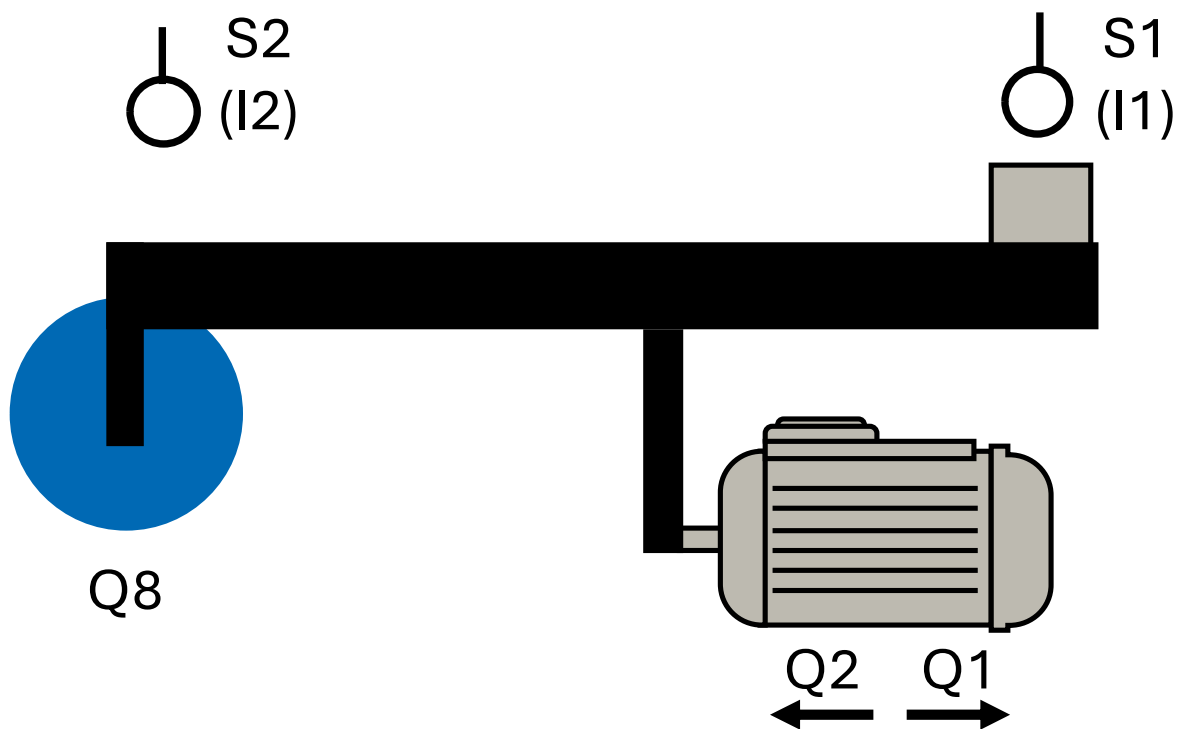
Target:

I can create the automatic program for the converter of the production line based on the function description and the sequence chain created in GRAFCET.

Task:

Create the PLC program for the automatic sequence so that the function is implemented based on the system description.

The chain is initialized when the control unit is switched on.

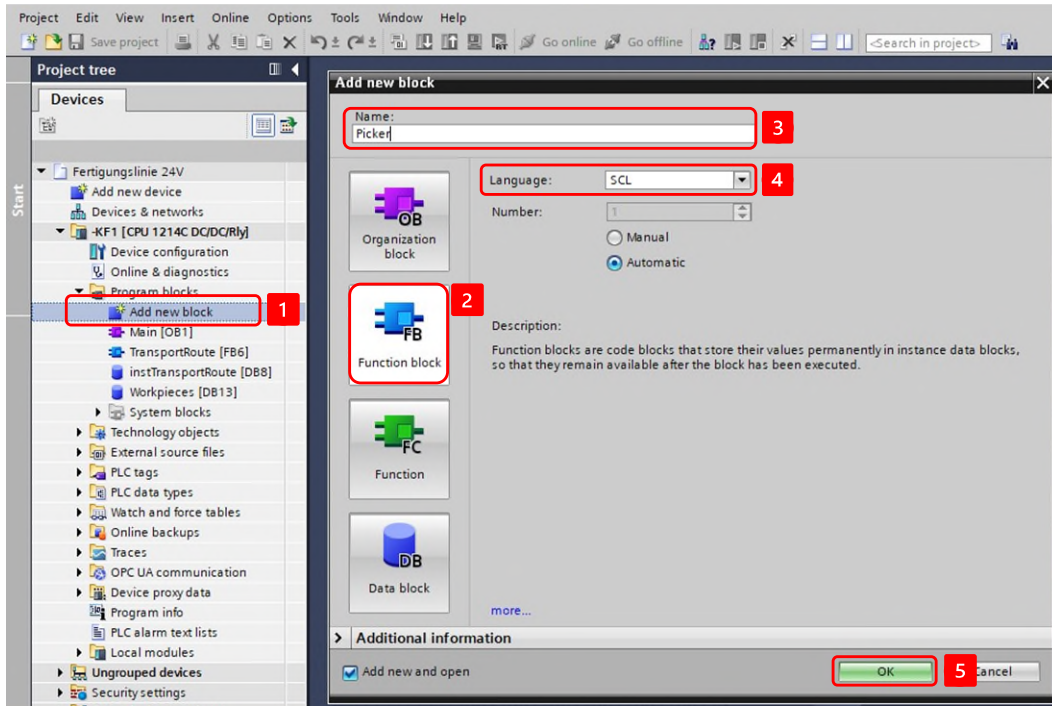


Picture 4 System diagram - converter

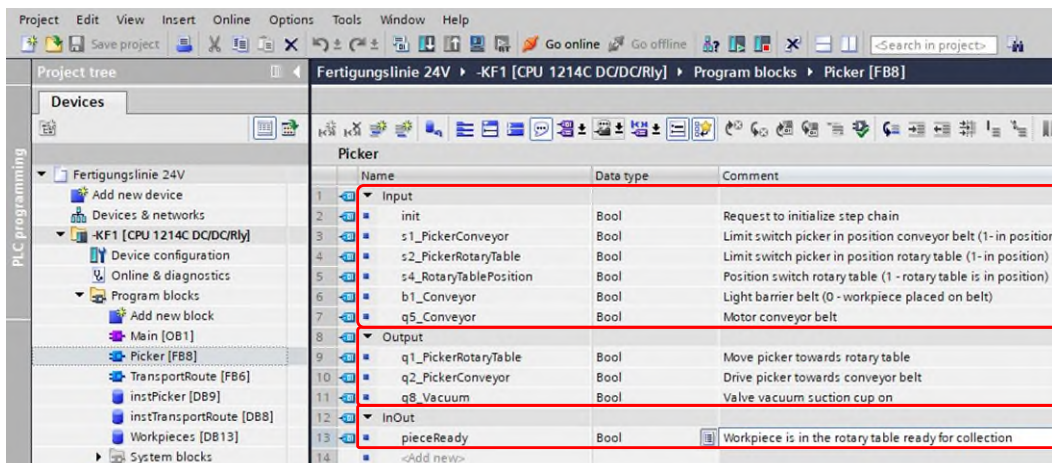
## Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Converter

### Procedure:

1. Create a new function block, select the desired programming language and assign a meaningful name:

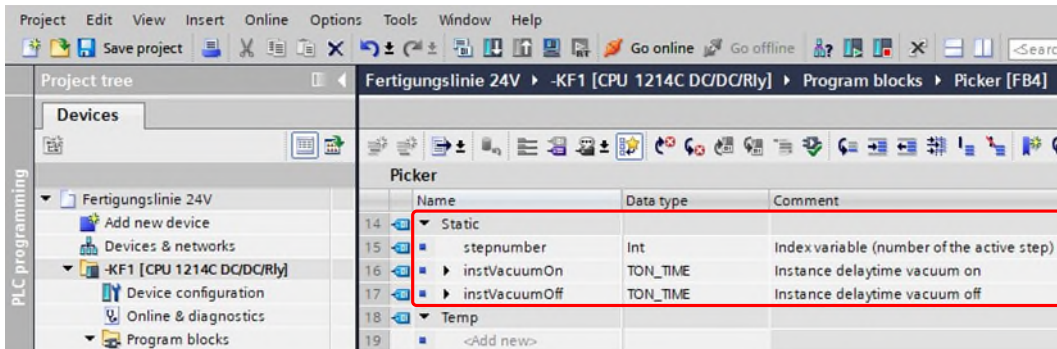


2. Declare variables for the sensors and actuators, a variable for initializing the step chain and a variable for transferring the workpiece information in the function block interface:



Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Converter

3. Declare the index variable in the static area of the function block interface, as well as two instances for the delay of the vacuum signal:



4. Program the initialization of the chain:

```

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

```

5. Implement the individual steps from the GRAFCET in the following CASE structure. A new CASE must be created in the structure for each step, which represents the step number:



All actions are reset in the initial step. This ensures that no actions remain set if an active step chain is aborted by an initialization request.

```

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;

```

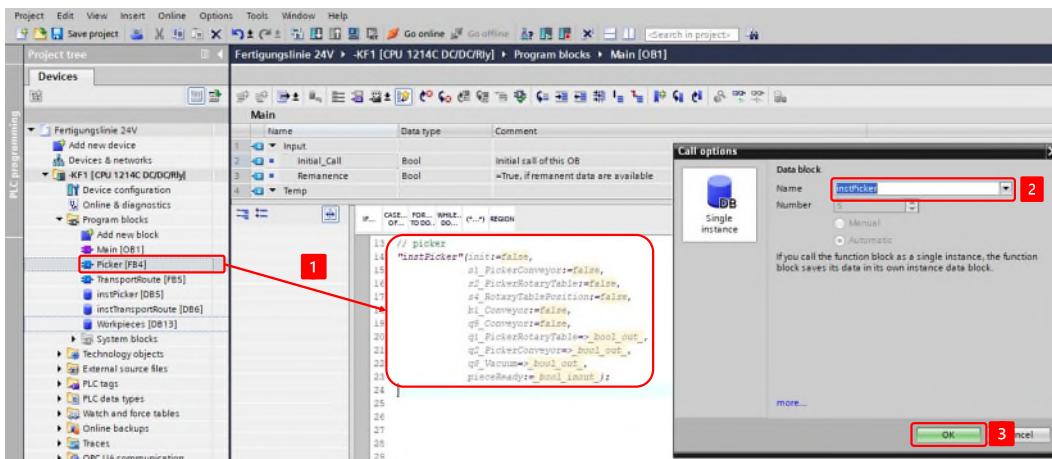
Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST/SCL] - Converter

6. Program the time functions according to the CASE structure:

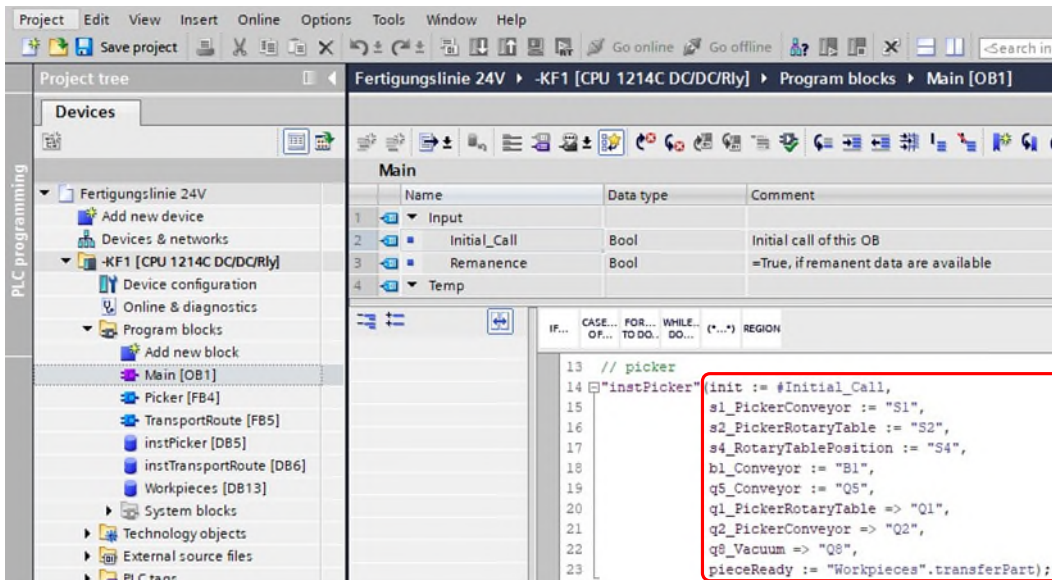
```

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. Call up the function module in "MAIN" and create an instance:



8. Connect the function block interface with the input and output variables from your variable table:



The "Initial\_Call" system bit provided by Siemens is used as the initialization request. This is "TRUE" when the MAIN is run through for the first time.

Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Converter

9. Commission the system in a structured manner with the aid of a commissioning protocol.



In order for the transfer unit to pick up a workpiece from the turntable, it can be manually set to the corresponding status in the workpiece management.

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	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



Solution

Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Converter

**Solution:**

The solution can be found in the TIA Portal project "Fertigungslinie\_03\_Umsetzer\_SCL.zap17".



## 10.9 Exercise: Convert GRAFCET sequence chain into program code [FUP] - Magazine

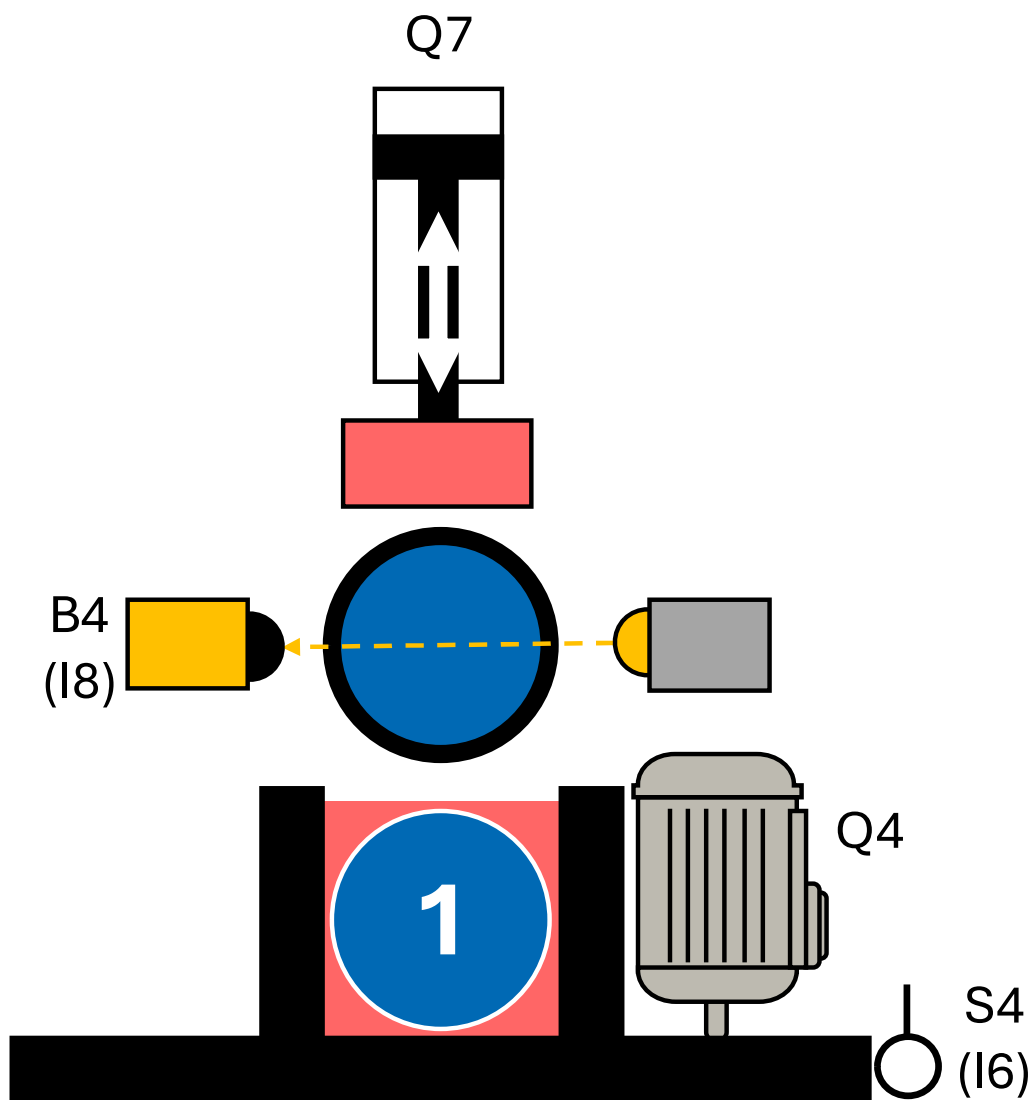
Target:

I can create the automatic program for the magazine of the production line based on the functional description and the sequence chain created in GRAFCET.

Task:

Create the PLC program for the automatic sequence so that the function is implemented based on the system description.

The chain is initialized when the control unit is switched on.



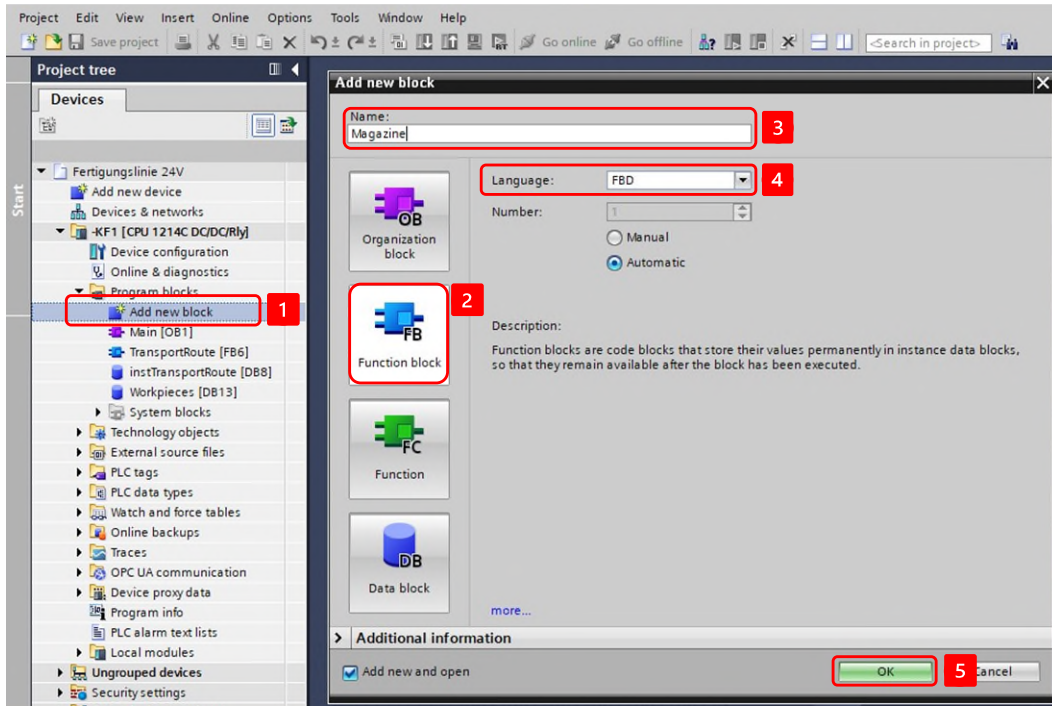
Picture 5 System diagram - Magazine



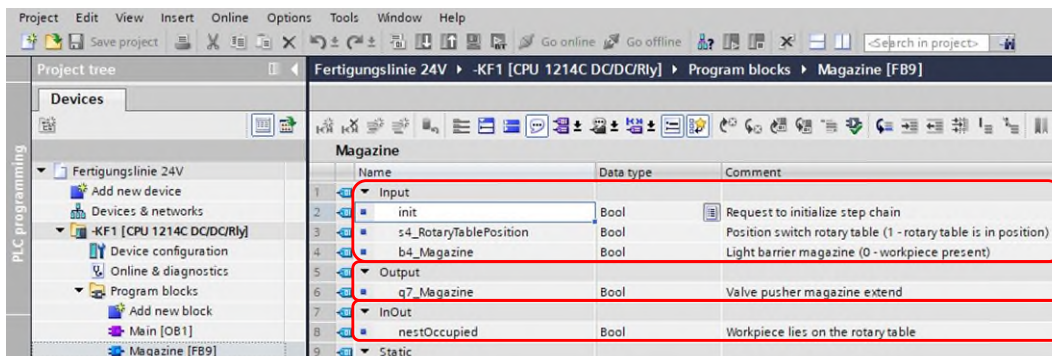
# Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FUP] - Magazine

## Procedure:

1. Create a new function block, select the desired programming language and assign a meaningful name:

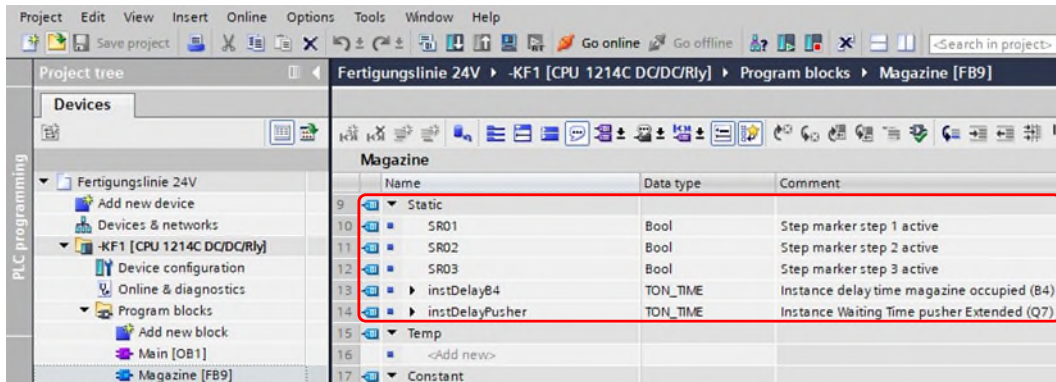


2. Declare variables for the sensors and actuators, a variable for initializing the step chain and a variable for transferring the workpiece information in the function block interface:

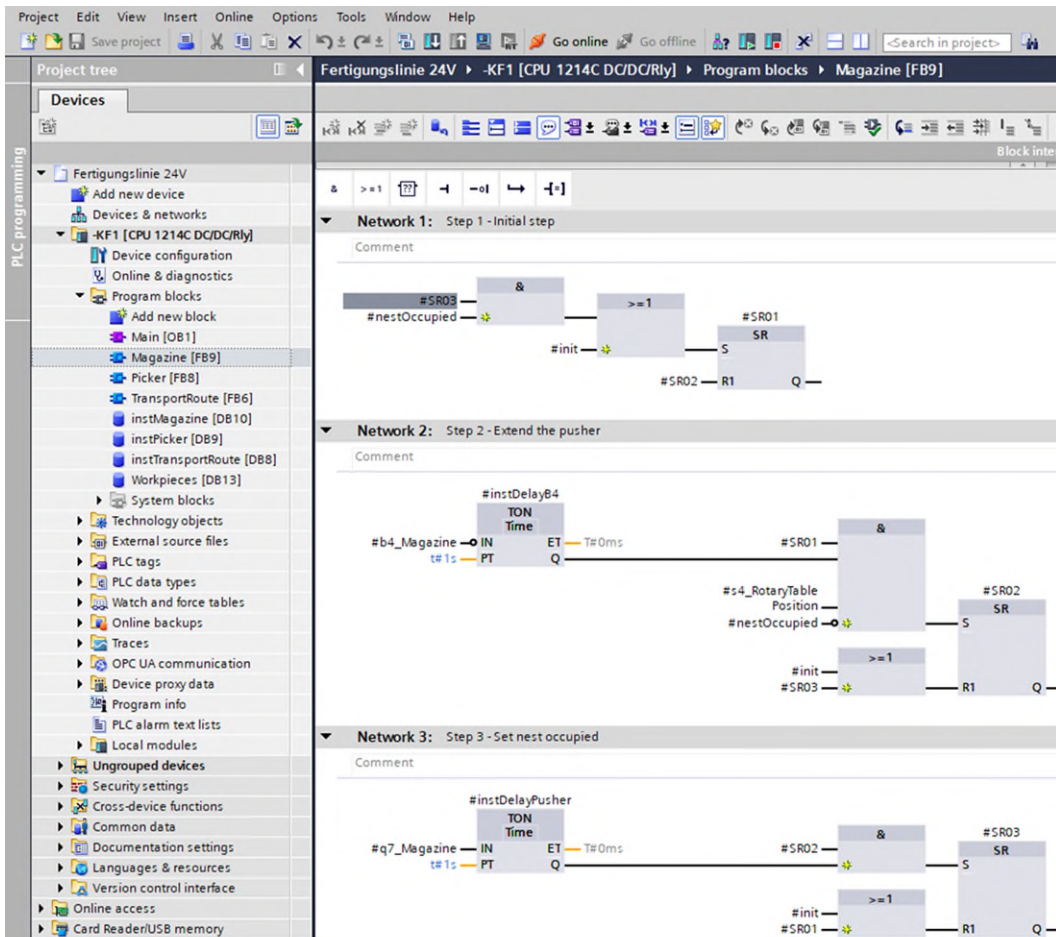


Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FUP] - Magazine

3. Declare the step flags in the static area of the function block interface, as well as two instances for the wait times:

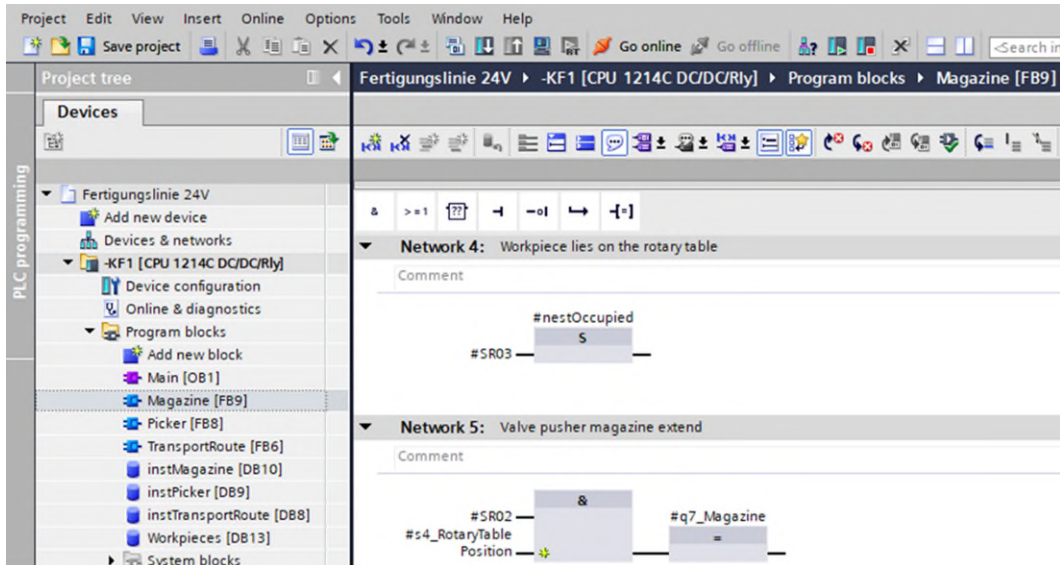


4. Implement the individual steps using flip-flops based on the GRAFCET. A new network must be used for each step:

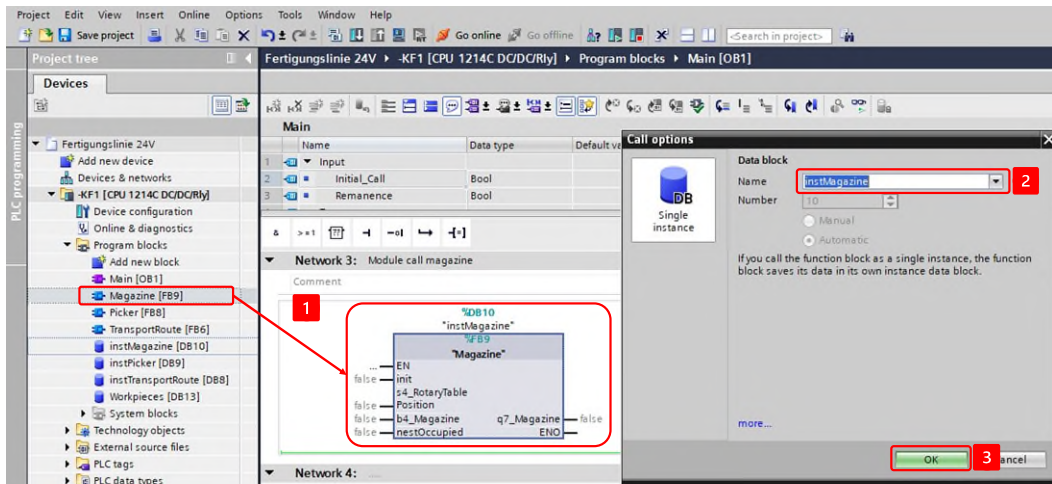


Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FUP] - Magazine

5. Assign the actions below the step chain in the next networks:

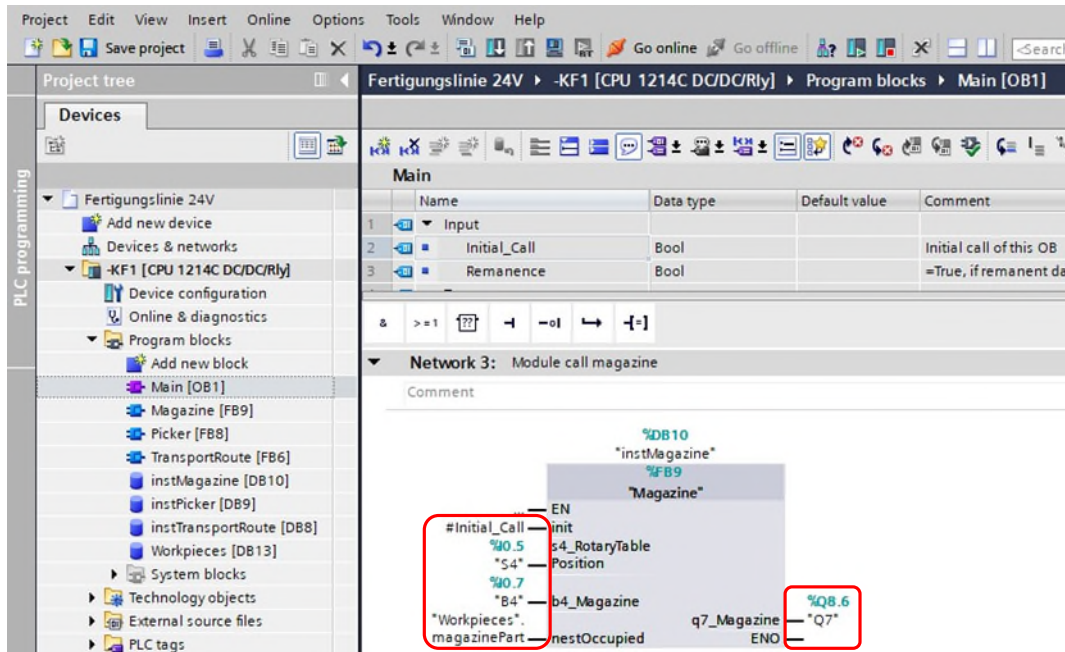


6. Call up the function module in "MAIN" and create an instance:



Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FUP] - Magazine

- Connect the function block interface with the input and output variables from your variable table:



**i** The "Initial\_Call" system bit provided by Siemens is used as the initialization request. This is "TRUE" when the MAIN is run through for the first time.

- Commission the system in a structured manner with the aid of a commissioning protocol.

**i** To place another workpiece on the turntable, it can be manually set to the corresponding FB8 status in the workpiece management.

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

	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



Solution

Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FUP] - Magazine

**Solution:**

The solution can be found in the TIA Portal project "Fertigungslinie\_04\_Magazin\_FUP.zap17".



## 10.10 Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Magazine

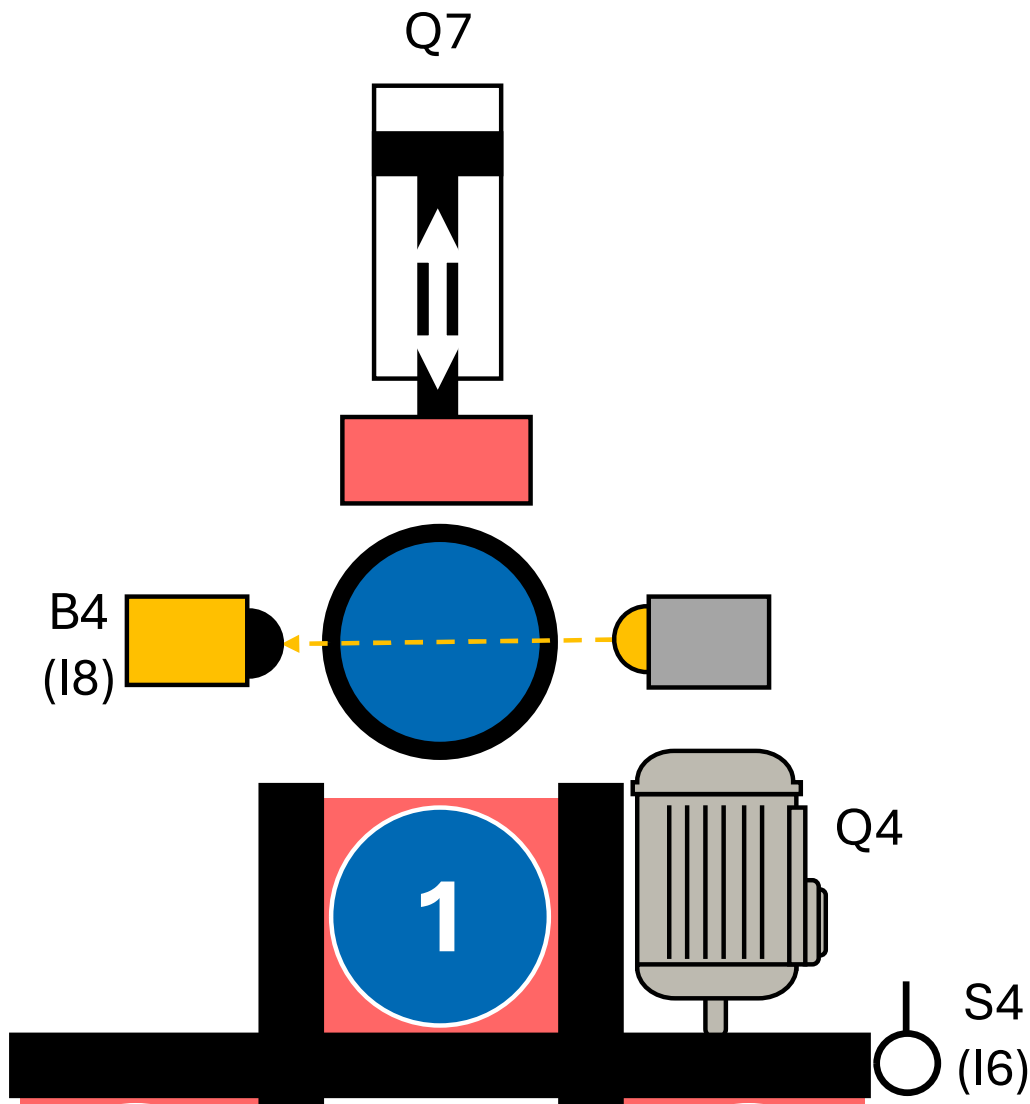
Target:

I can create the automatic program for the magazine of the production line based on the functional description and the sequence chain created in GRAFCET.

Task:

Create the PLC program for the automatic sequence so that the function is implemented based on the system description.

The chain is initialized when the control unit is switched on.

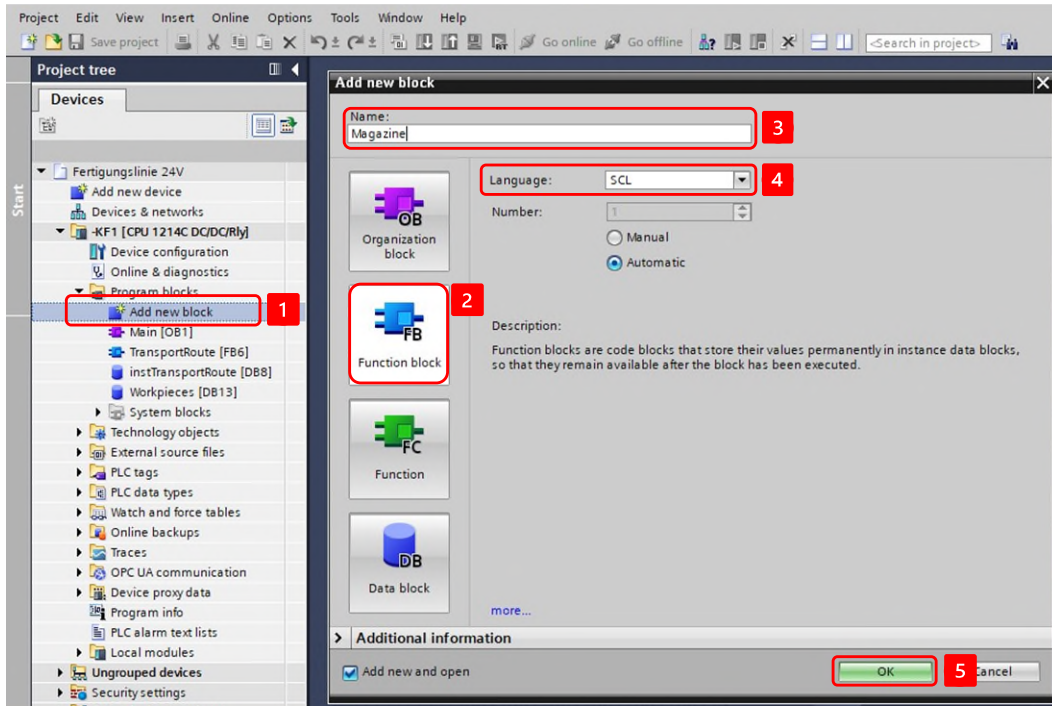


Picture 6 System diagram - Magazine

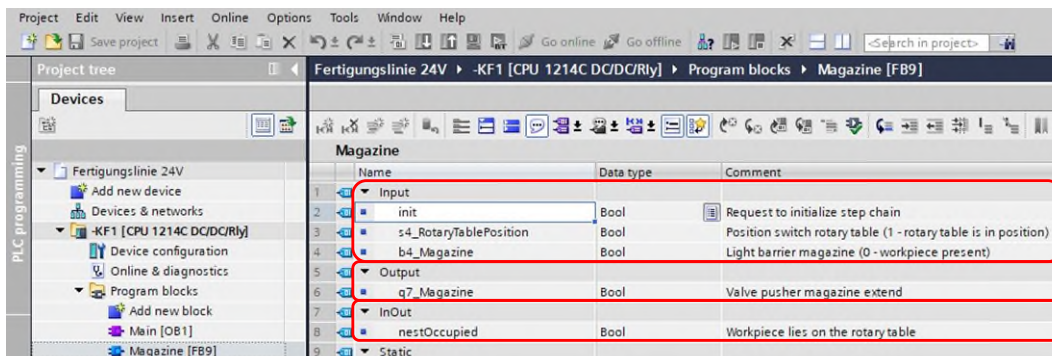
## Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Magazine

### Procedure:

1. Create a new function block, select the desired programming language and assign a meaningful name:



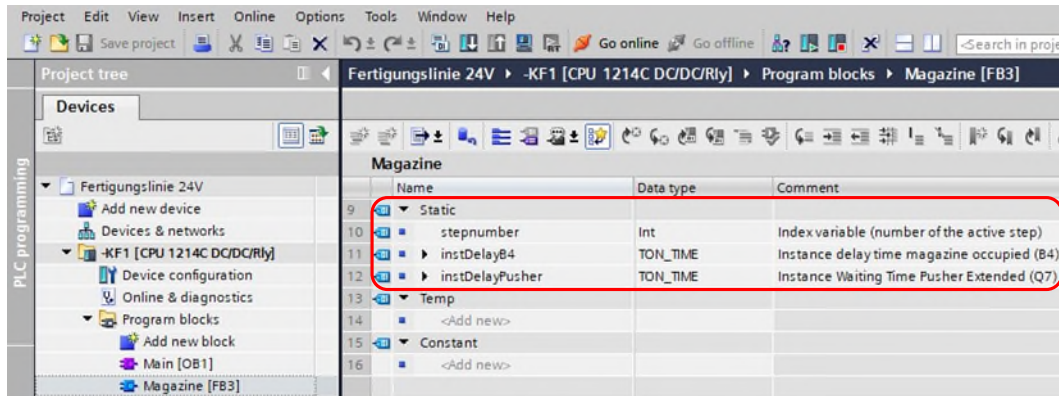
2. Declare variables for the sensors and actuators, a variable for initializing the step chain and a variable for transferring the workpiece information in the function block interface:





Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Magazine

3. Declare the index variable in the static area of the function block interface, as well as two instances for the waiting times:



4. Program the initialization of the chain:

```

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

```

5. Implement the individual steps from the GRAFCET in the following CASE structure. A new CASE must be created in the structure for each step, which represents the step number:



All actions are reset in the initial step. This ensures that no actions remain set if an active step chain is aborted by an initialization request.

```

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;

```

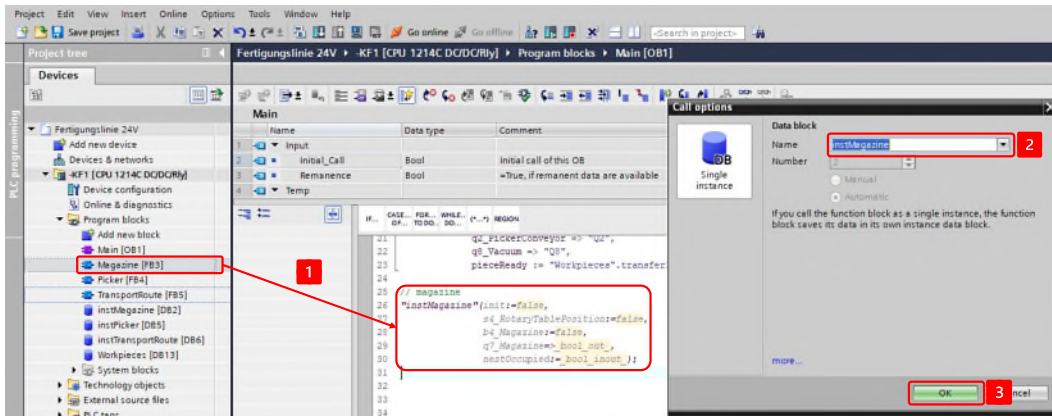
6. Program the time functions according to the CASE structure:

Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST/SCL] - Magazine

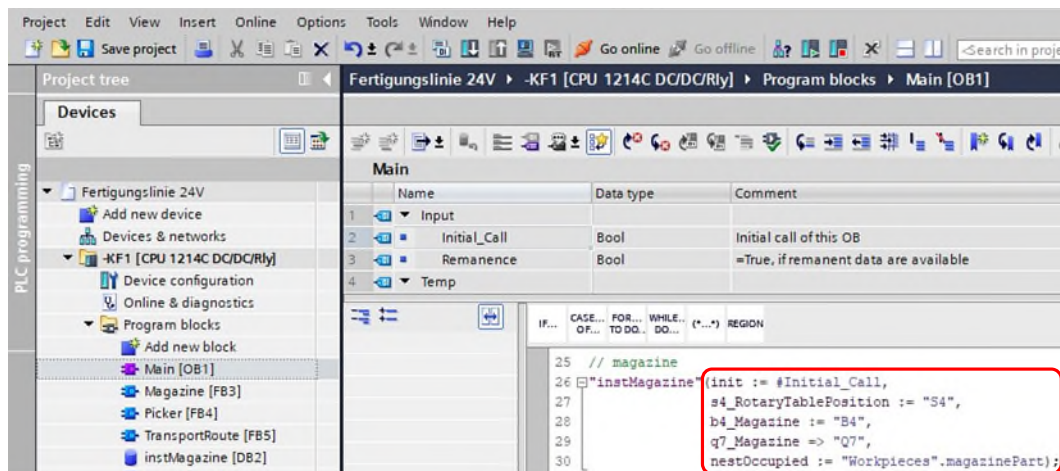
```

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. Call up the function module in "MAIN" and create an instance:



8. Connect the function block interface with the input and output variables from your variable table:



The "Initial\_Call" system bit provided by Siemens is used as the initialization request. This is "TRUE" when the MAIN is run through for the first time.

Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Magazine

9. Commission the system in a structured manner with the aid of a commissioning protocol.



To place another workpiece on the turntable, it can be manually set to the corresponding status in the workpiece management.

	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



Solution

Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Magazine

**Solution:**

The solution can be found in the TIA Portal project "Fertigungslinie\_04\_Magazin\_SCL.zap17".



## 10.11 Exercise: Convert GRAFCET sequence chain into program code [FBD] - Processing station

Target:

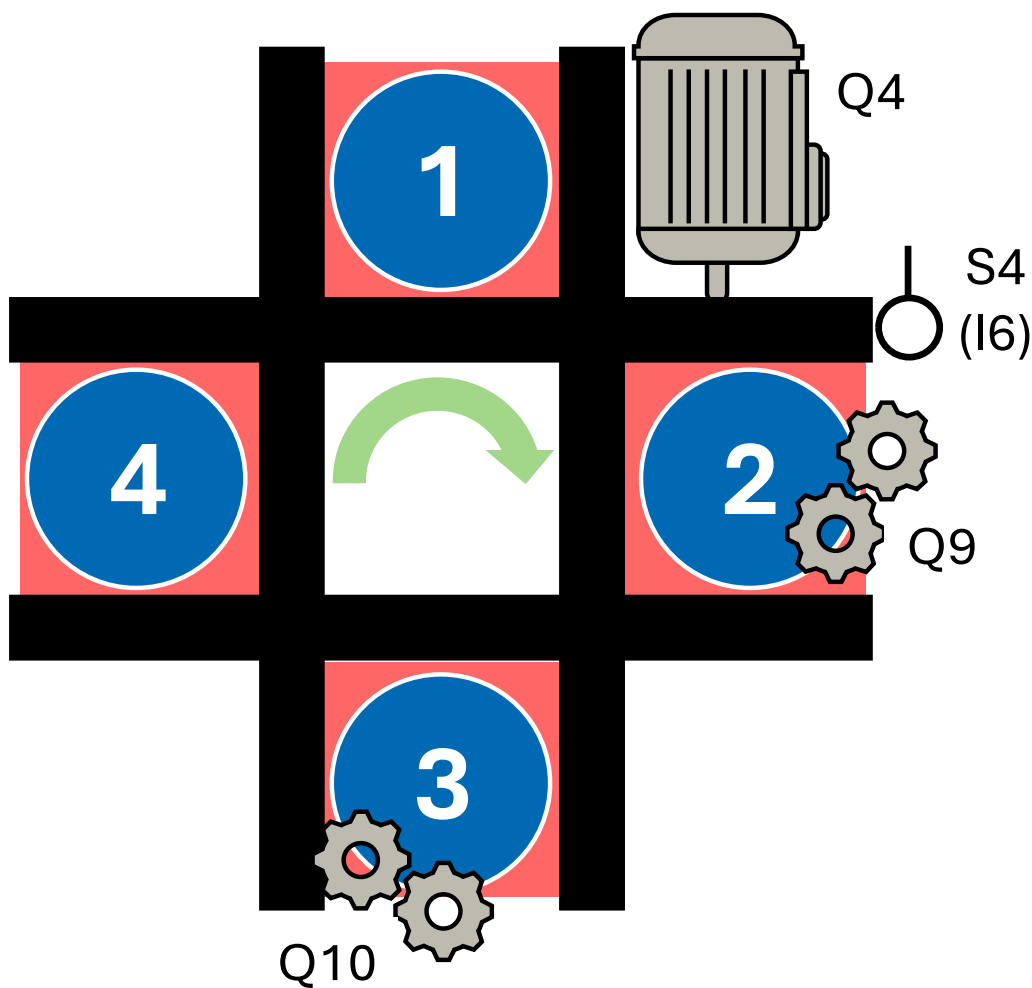
I can create the automatic program for a processing station on the production line using the functional description and the sequence chain created in GRAFCET.

Task:

Create the PLC program for the automatic sequence so that the function is implemented based on the system description.

The chain is initialized when the control unit is switched on.

The function block must be called twice, one instance for the drilling station (Q9) and another instance for welding (Q10).

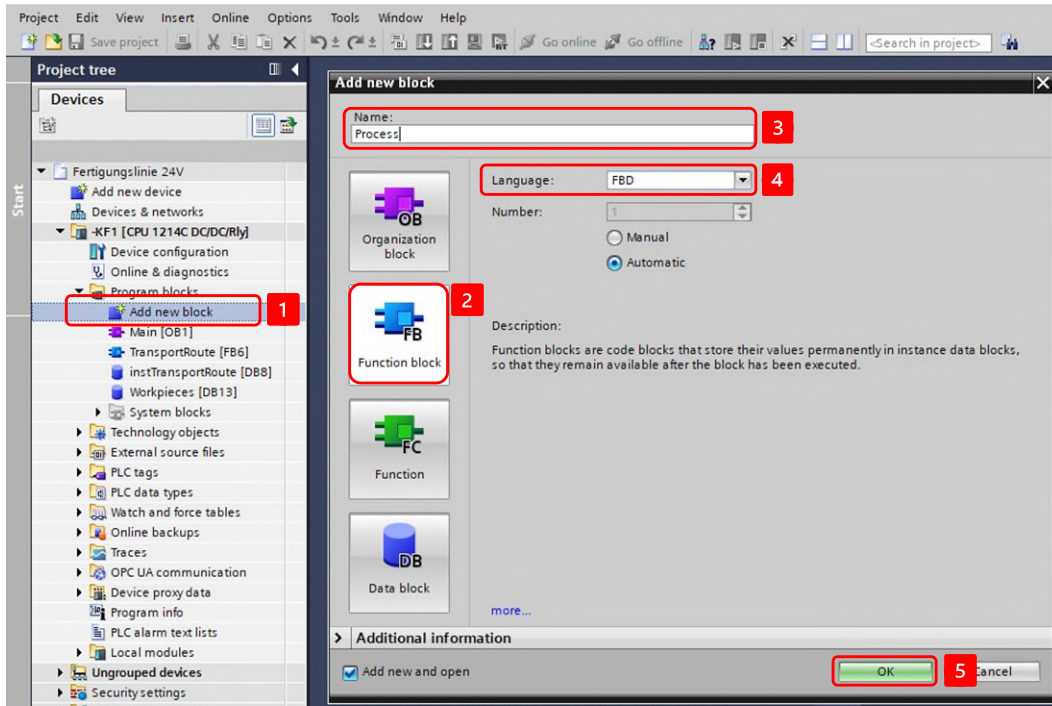


Picture 7 System diagram - processing station

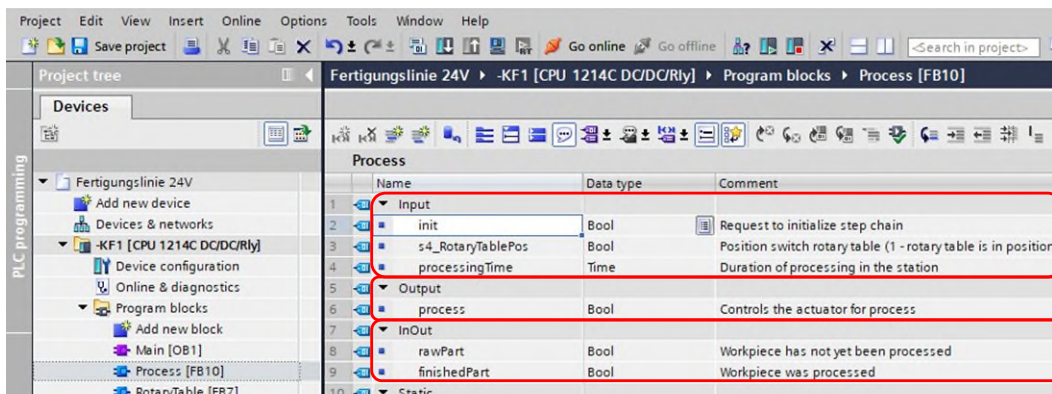
## Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FBD] - Processing station

### Procedure:

1. Create a new function block, select the desired programming language and assign a meaningful name:

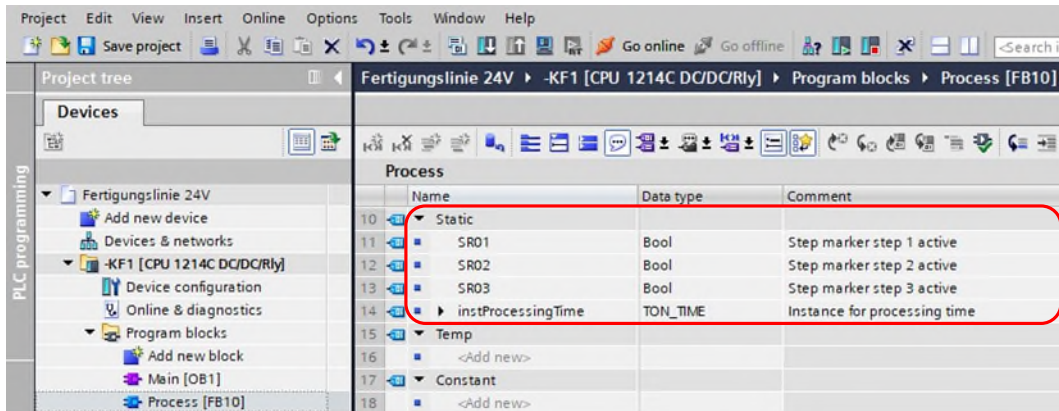


2. Declare variables for the sensors and actuators, a variable for initializing the step chain, for the processing time and two variables for transferring the workpiece information in the function block interface:

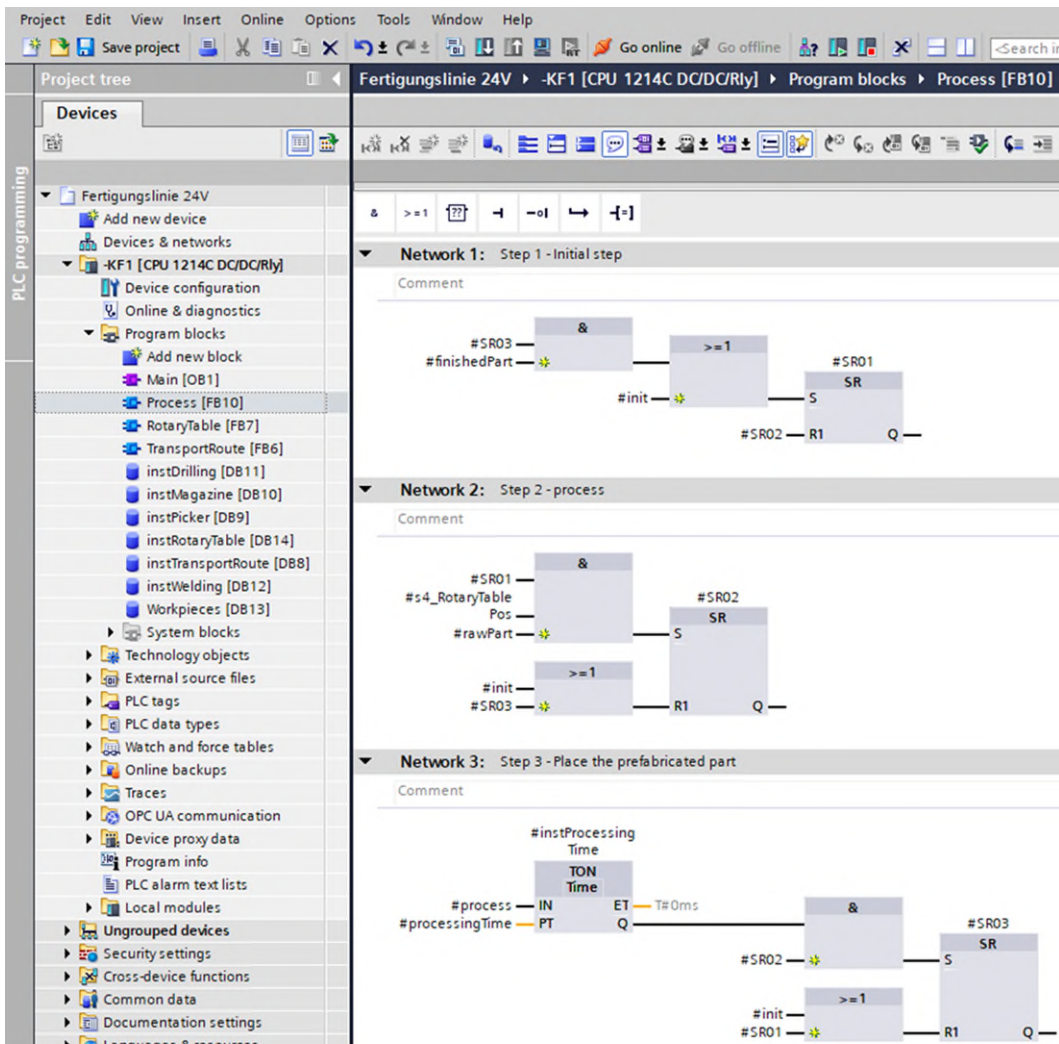


Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FBD] - Processing station

3. Declare the step flags in the static area of the block interface, as well as an instance for the processing time:



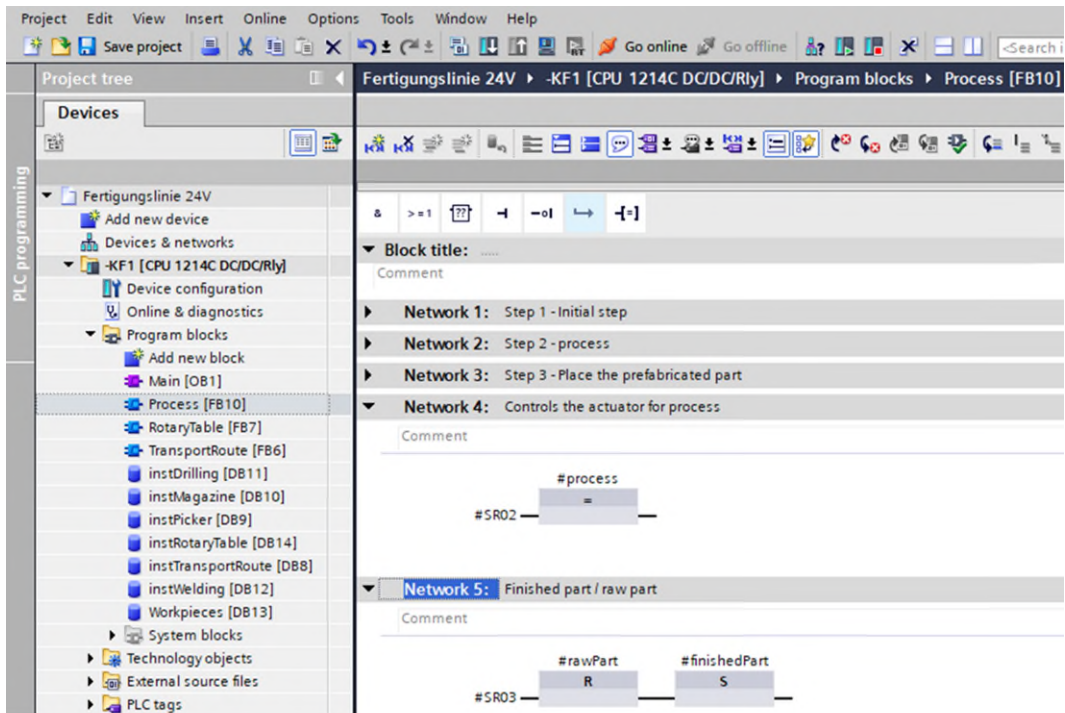
4. Implement the individual steps using flip-flops based on the GRAFCET. A new network must be used for each step:



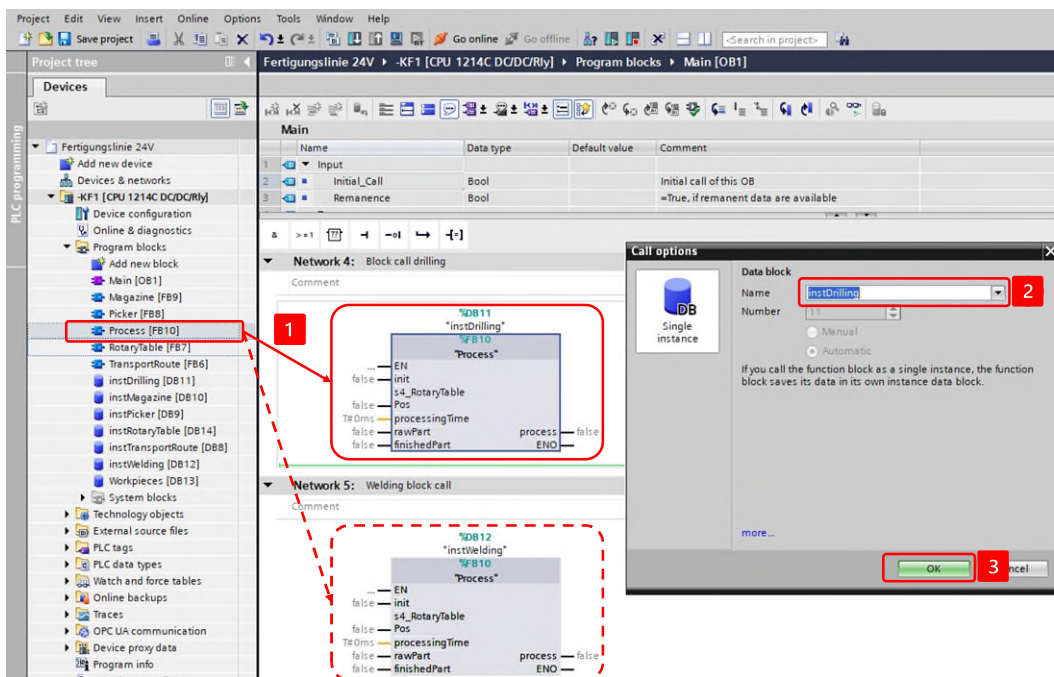


Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FBD] - Processing station

5. Assign the actions below the step chain in the next networks:



6. Call the function module twice in "MAIN" and create a separate instance for each time:



Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FBD] - Processing station

- Connect the function block interface with the input and output variables from your variable table, as well as the processing time:

The screenshot shows the SIMATIC Manager interface. The left pane displays the project tree for 'Fertigungsline 24V'. The main window shows the 'Main [OB1]' program block configuration. Below the configuration, two networks are visible:

- Network 4: Block call drilling** - Shows a function block call for 'instDrilling' (FB10) with inputs: #Initial\_Call, %Q0.5, s4\_RotaryTable, Pos, t#3s, processingTime, and outputs: drillingRawPart, drillingFinished, Part, finishedPart. The output %Q9.0 is also shown.
- Network 5: Welding block call** - Shows a function block call for 'instWelding' (FB10) with inputs: #Initial\_Call, %Q0.5, s4\_RotaryTable, Pos, t#5s, processingTime, and outputs: weldingRawPart, weldingFinished, Part, finishedPart. The output %Q9.1 is also shown.

**i** The "Initial\_Call" system bit provided by Siemens is used as the initialization request. This is "TRUE" when the MAIN is run through for the first time.

- Commission the system in a structured manner with the aid of a commissioning protocol.

**i** In order for another workpiece to be processed in the station, it can be manually set to the corresponding status in the workpiece management.

The screenshot shows the 'Workpieces [DB13]' table in SIMATIC Manager. The table has the following columns: Name, Data type, Monitor value, and Comment.

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



Solution

Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [FBD] - Processing station

**Solution:**

The solution can be found in the TIA Portal project "Fertigungslinie\_05\_Bearbeiten\_FUP.zap17".



## 10.12 Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Processing station

Target:

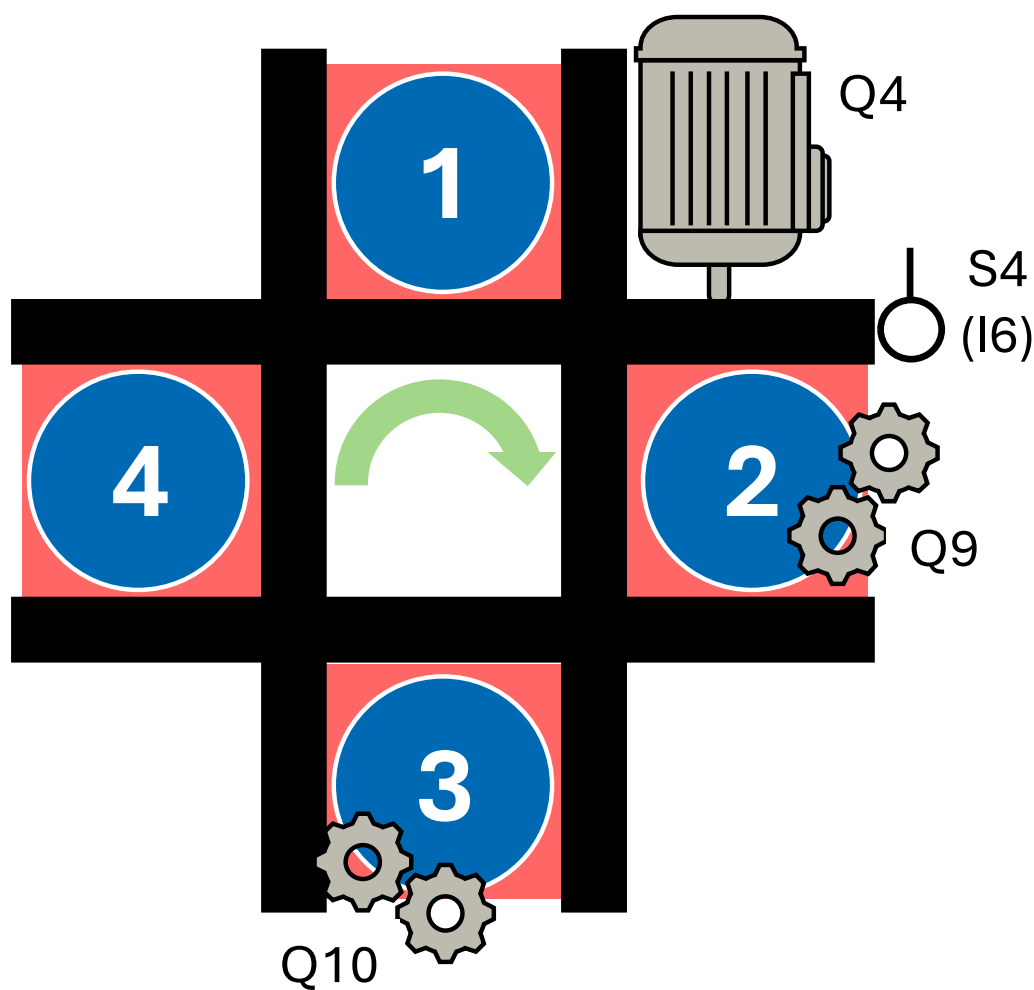
I can use the functional description and the sequence chain created in GRAFCET to create the automatic program for a processing station on the production line.

Task:

Create the PLC program for the automatic sequence so that the function is implemented based on the system description.

The chain is initialized when the control unit is switched on.

The function block must be called twice, one instance for the drilling station (Q9) and another instance for welding (Q10).

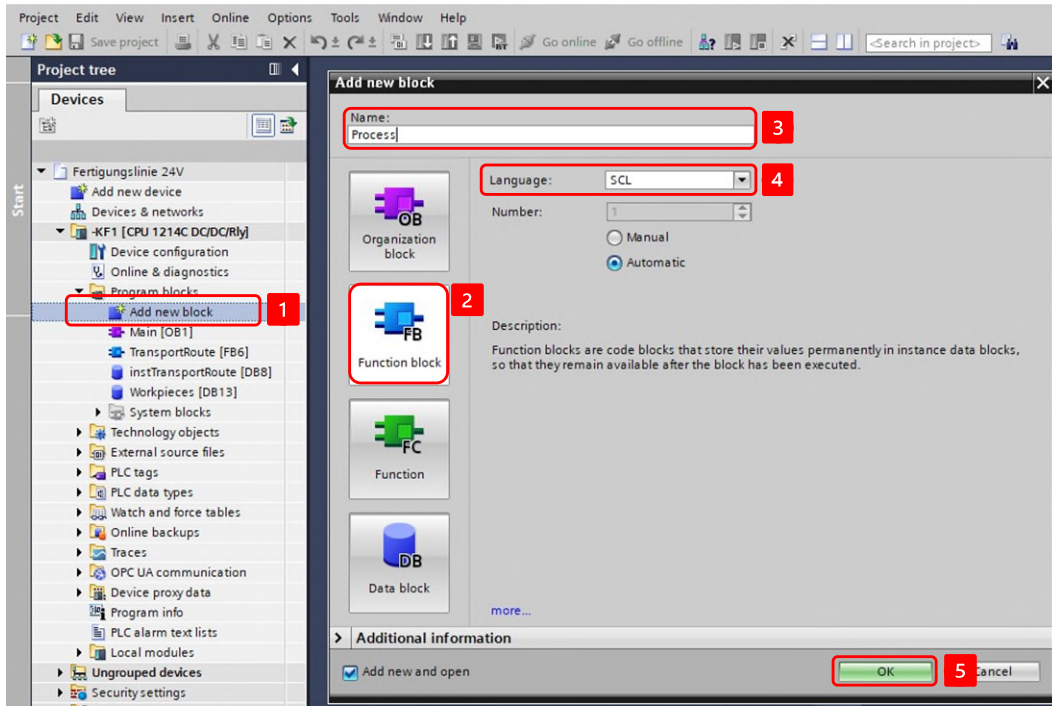


Picture 8 System diagram - processing station

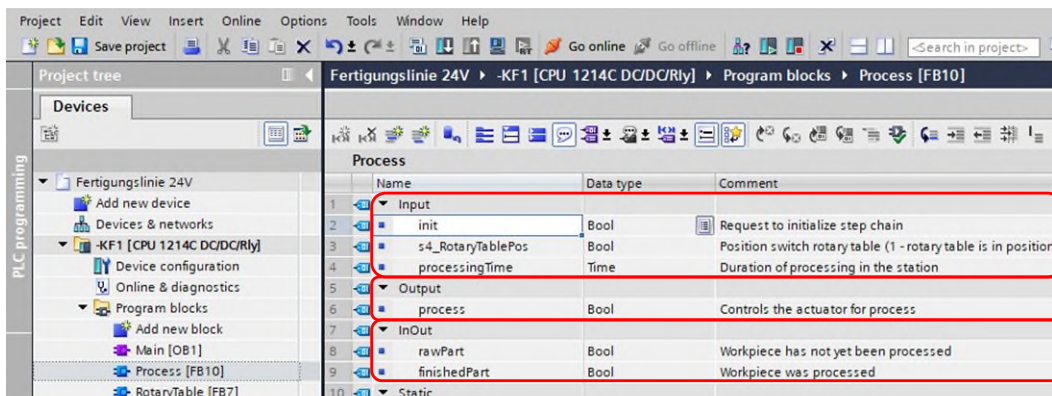
Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Processing station

Procedure:

1. Create a new function block, select the desired programming language and assign a meaningful name:

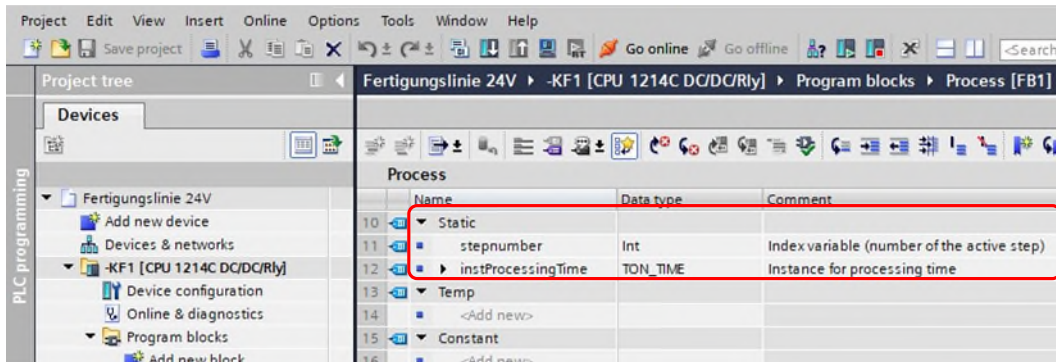


2. Declare variables for the sensors and actuators, a variable for initializing the step chain, for the processing time and two variables for transferring the workpiece information in the function block interface:



Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Processing station

3. Declare the index variable in the static area of the block interface, as well as an instance for the processing time:



4. Program the initialization of the chain:

```

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

```

5. Implement the individual steps from the GRAFCET in the following CASE structure. A new CASE must be created in the structure for each step, which represents the step number:



All actions are reset in the initial step. This ensures that no actions remain set if an active step chain is aborted by an initialization request.

```

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;

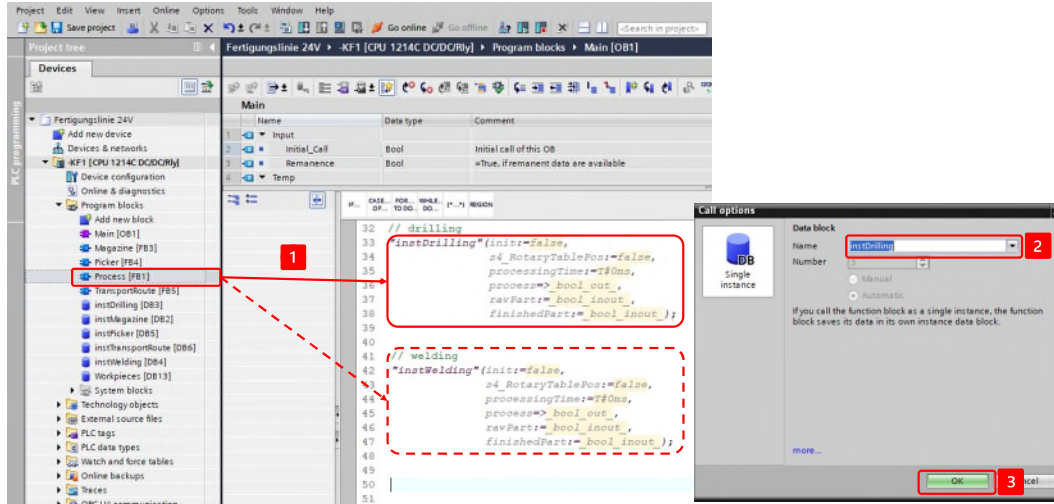
```

Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Processing station

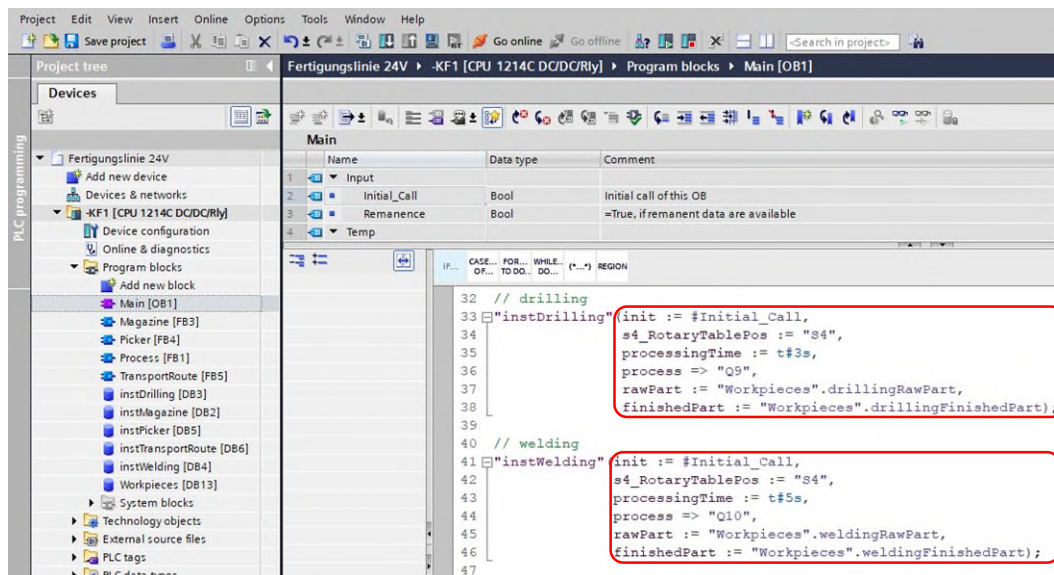
6. Program the time functions according to the CASE structure:

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

7. Call the function module twice in "MAIN" and create a separate instance for each time:



8. Connect the function block interface with the input and output variables from your variable table, as well as the processing time:




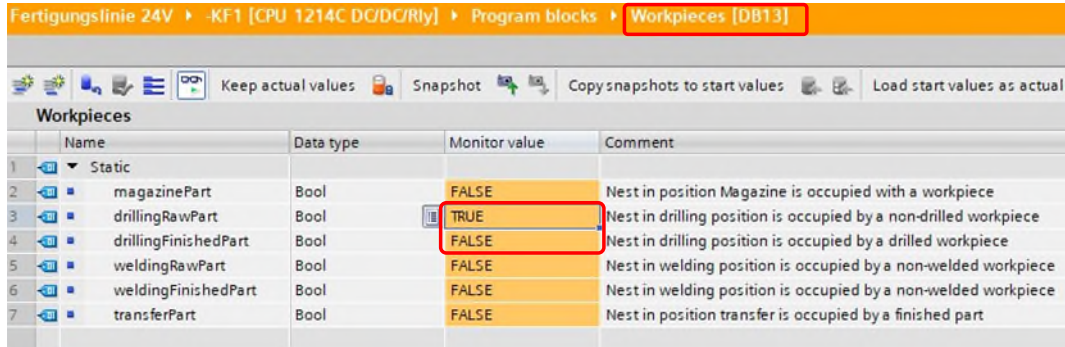
The "Initial\_Call" system bit provided by Siemens is used as the initialization request. This is "TRUE" when the MAIN is run through for the first time.



## Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Processing station

Commission the system in a structured manner with the aid of a commissioning protocol.

-  In order for another workpiece to be processed in the station, it can be manually set to the corresponding status in the workpiece management.



The screenshot shows the 'Workpieces' table in the SIMATIC Manager. The table has four columns: Name, Data type, Monitor value, and Comment. The 'drillingRawPart' row is highlighted with a red box, indicating its 'Monitor value' is 'TRUE'. Other rows show 'FALSE' values for their respective 'Monitor value' columns.

	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



Solution

Planning and implementing automatic processes - Exercise: Convert GRAFCET sequence chain into program code [ST / SCL] - Processing station

**Solution:**

The solution can be found in the TIA Portal project "Fertigungslinie\_05\_Bearbeiten\_SCL.zap17".



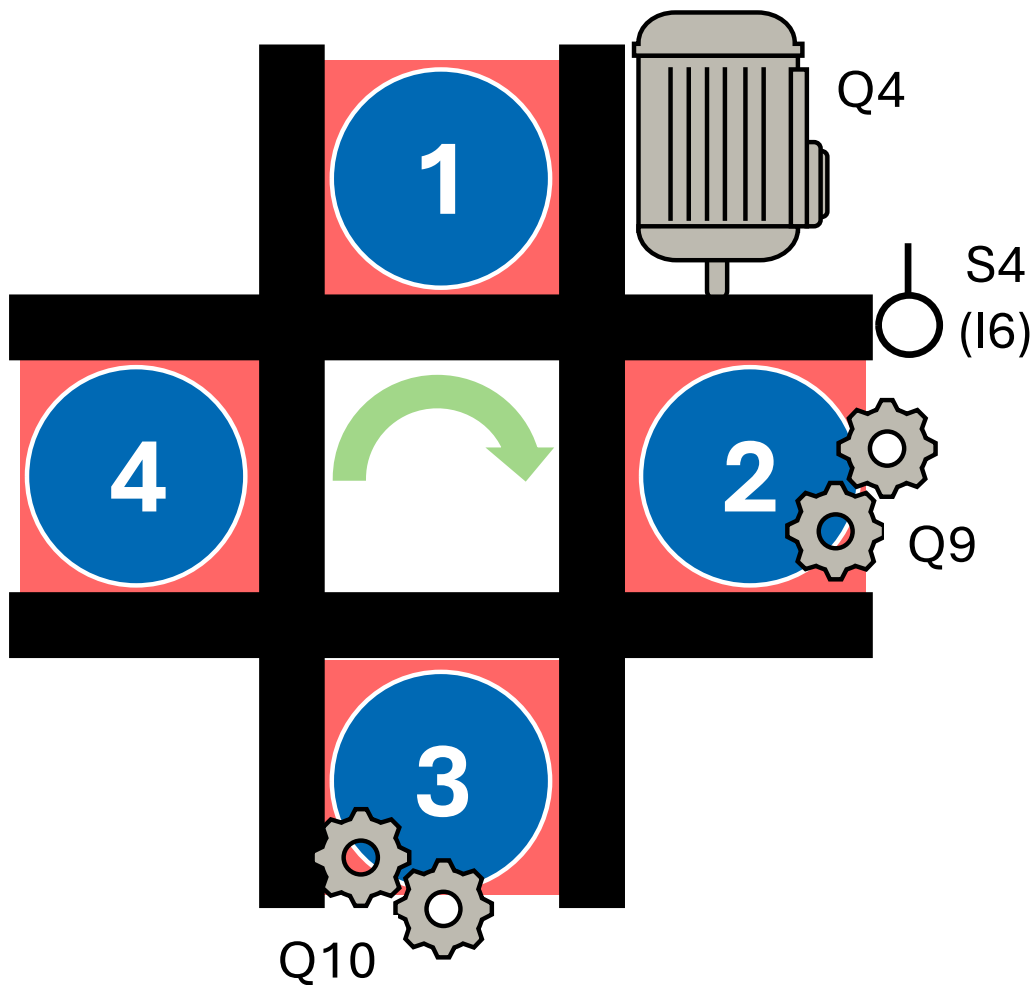
## 10.13 Exercise: Create function block for rotary table [FBD]

Target:

I can create the automatic program for the rotary table of the production line using the function description.

Task:

Create the PLC program for the automatic sequence so that the function is implemented based on the system description.



Picture 9 System diagram - turntable

Function:

To control the turntable, the following enable conditions must be met at all times:

- Slider Q7 not activated
- Converter (S2) not in the end position on the turntable

The table must be moved until it is back in position (S4).

The table can be moved if,

- a workpiece is in the nest of the magazine (I).
- a finished part is located in one of the two processing stations (2/3).

The table may not be moved as long as

- a workpiece is in the nest of the transfer station (4).
- a raw part is located in one of the two processing stations (2/3).

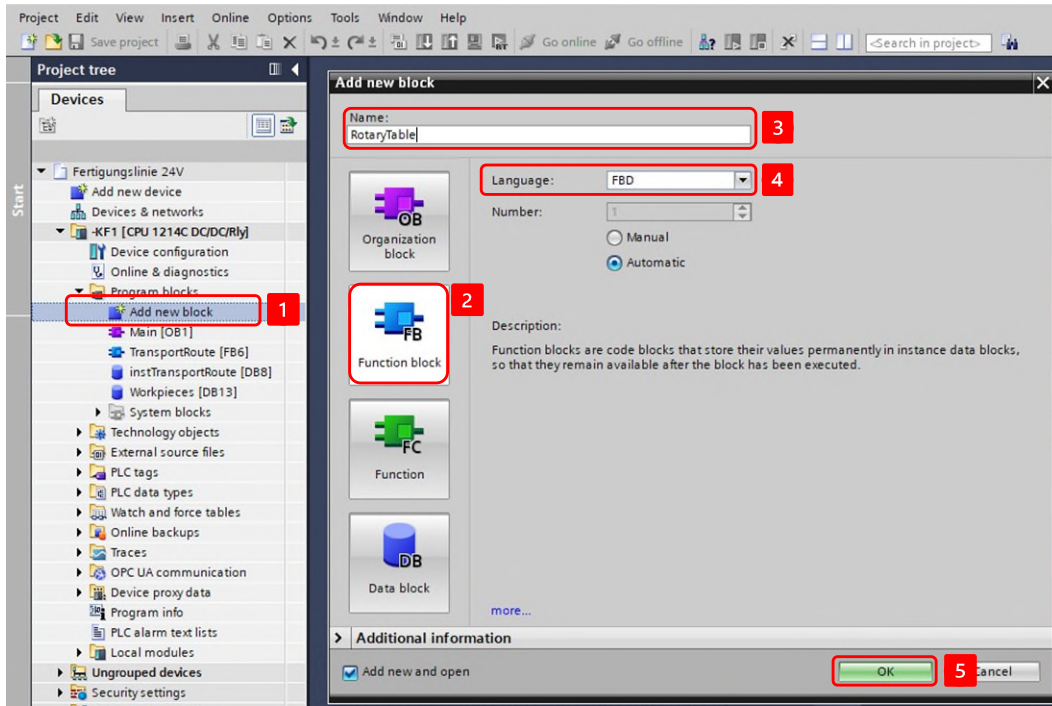
If the table has been turned to the next position (positive edge S4), the workpiece data must also be rotated by one station:

- If a finished part is in the "Welding" station, it must be reset and set as a workpiece in the transfer station.
- If a finished part is in the "Drilling" station, this must be reset and set as a raw part in the "Welding" station.
- If a workpiece is in the "Magazine" station, it must be reset and set as a raw part in the "Drilling" station.

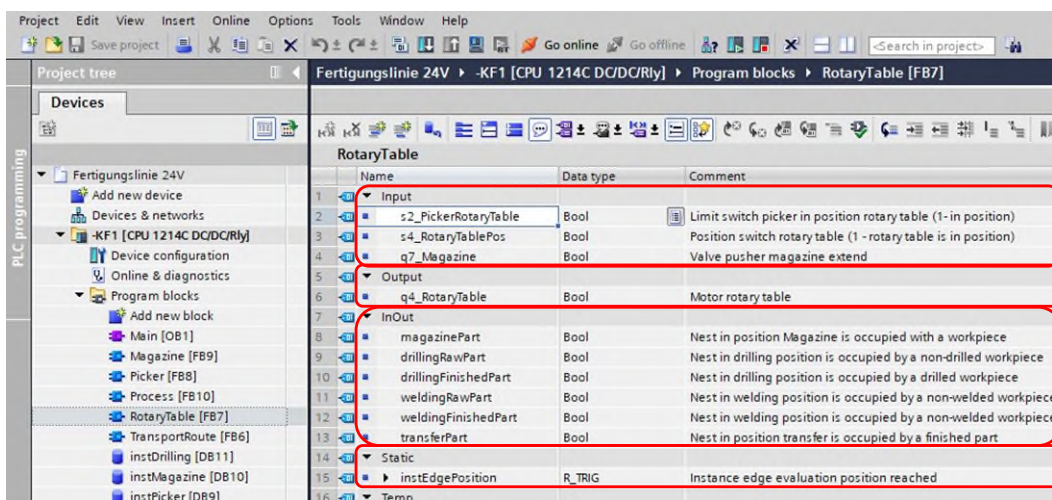
## Planning and implementing automatic processes - Exercise: Create function block for rotary table [FBD]

### Procedure:

1. Create a new function block, select the desired programming language and assign a meaningful name:

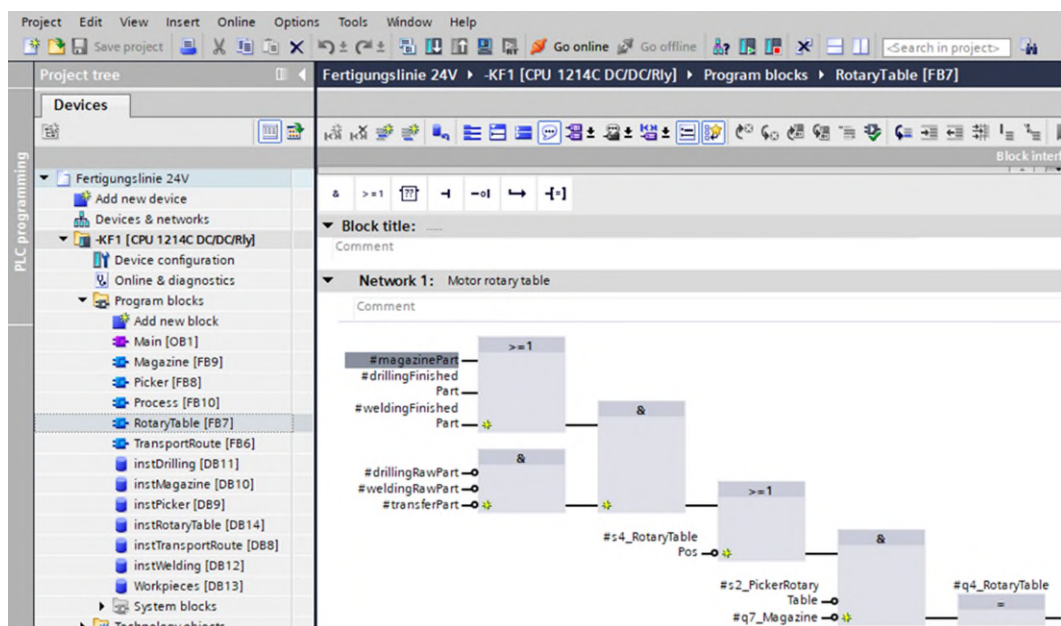


2. Declare variables for the sensors and actuators, the variables for transferring the workpiece information and the instance for edge evaluation in the function block interface:

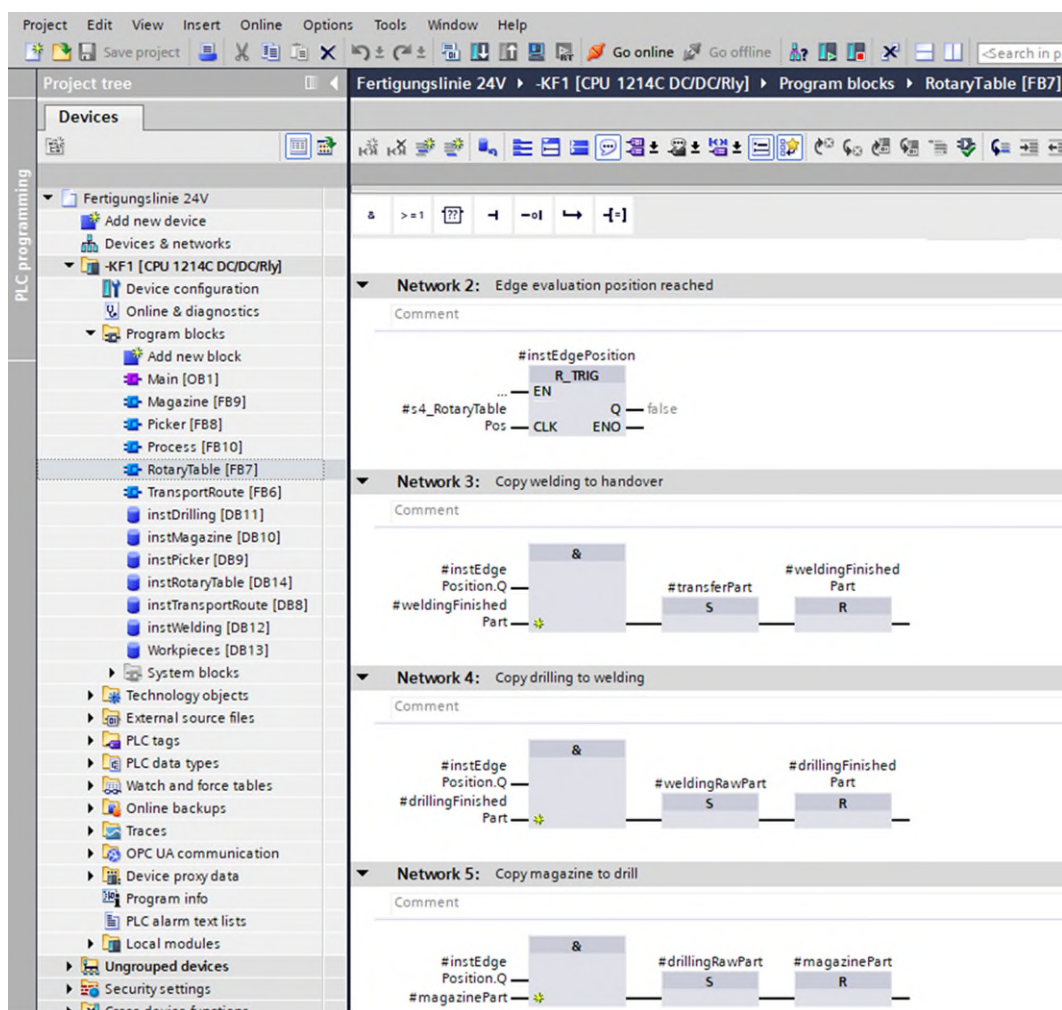


Planning and implementing automatic processes - Exercise: Create function block for rotary table [FBD]

3. Program the control of the turntable (Q4) in the first network:

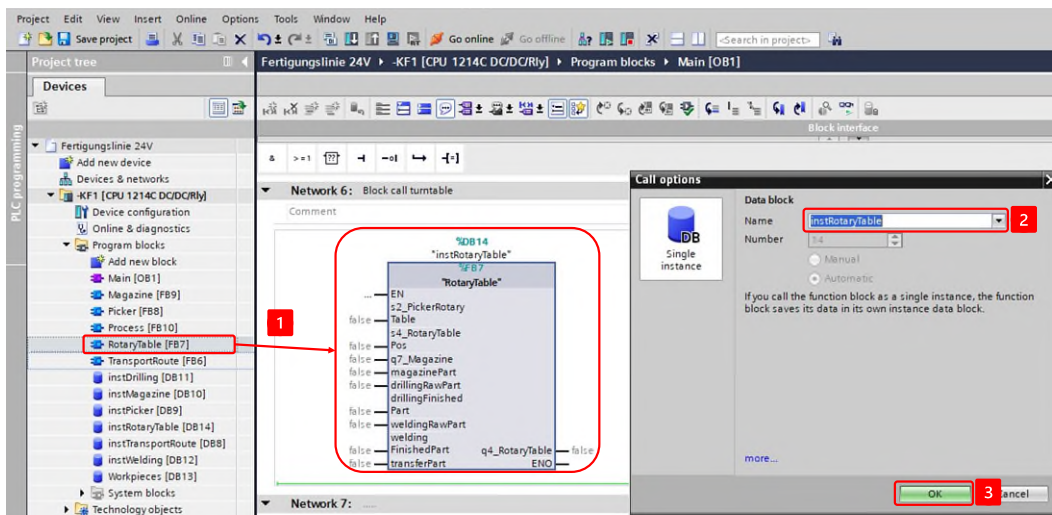


4. Program the edge evaluation (S4) and the rotation of the workpiece data in the following networks:

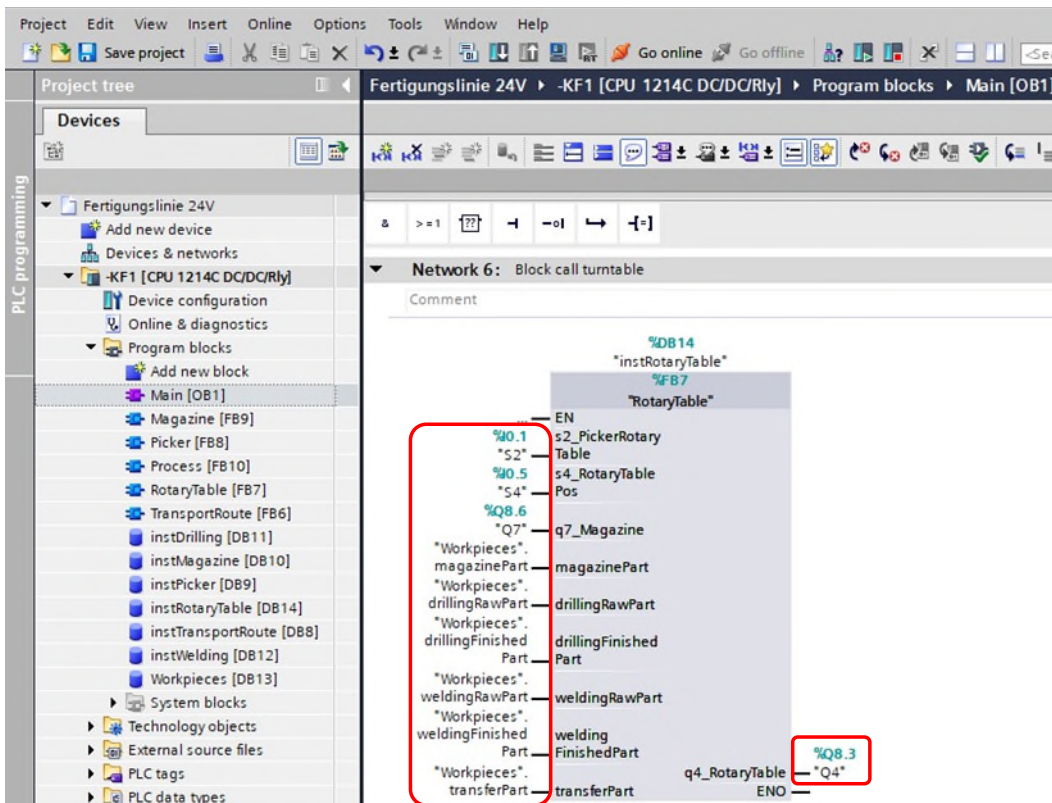


Planning and implementing automatic processes - Exercise: Create function block for rotary table [FBD]

5. Call up the function module in "MAIN" and create an instance:



6. Connect the module interface:



7. Commission the system in a structured manner with the aid of a commissioning protocol.





Solution

Planning and implementing automatic processes - Exercise: Create function block for rotary table [FBD]

**Solution:**

The solution can be found in the TIA Portal project "Fertigungslinie\_06\_Drehtisch\_FUP.zap17".



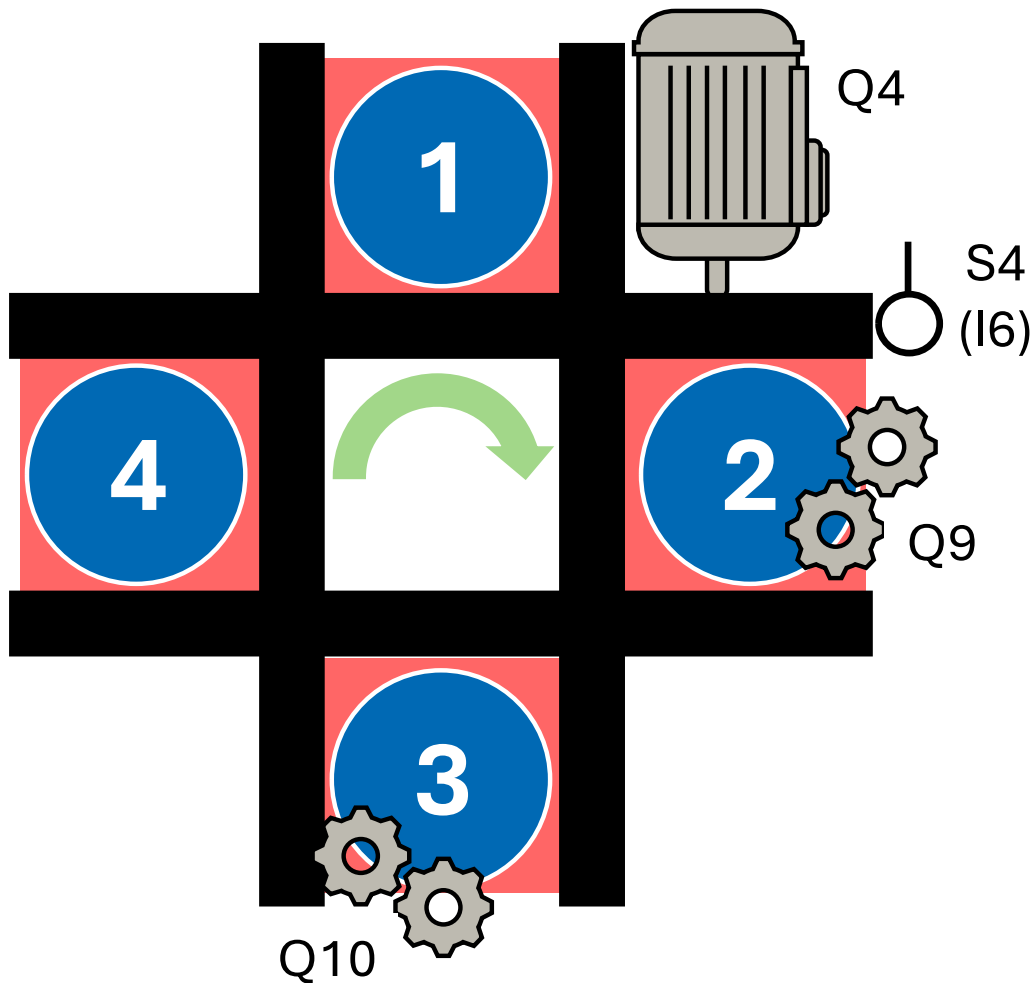
## 10.14 Exercise: Create function block for rotary table [ST / SCL]

Target:

I can create the automatic program for the rotary table of the production line using the function description.

Task:

Create the PLC program for the automatic sequence so that the function is implemented based on the system description.



Picture 10 System diagram - turntable

Function:

To control the turntable, the following enable conditions must be met at all times:

- Slider Q7 not activated
- Converter (S2) not in the end position on the turntable

The table must be moved until it is back in position (S4).

The table can be moved if,

- a workpiece is in the nest of the magazine (1).
- a finished part is located in one of the two processing stations (2/3).

The table may not be moved as long as

- a workpiece is in the nest of the transfer station (4).
- a raw part is located in one of the two processing stations (2/3).

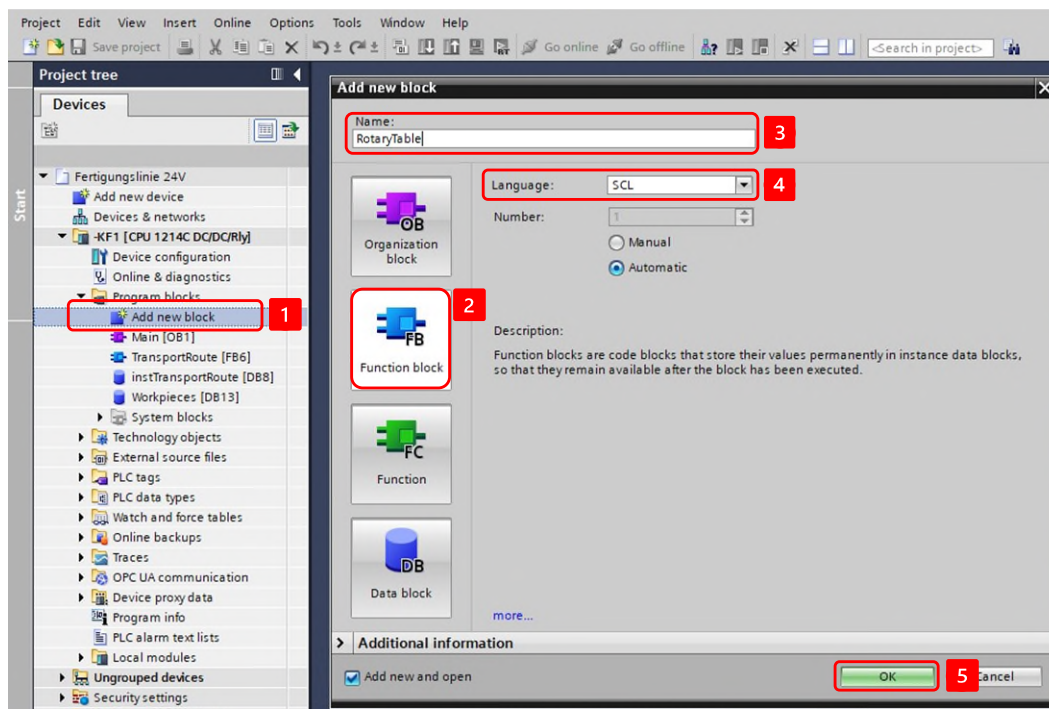
If the table has been turned to the next position (positive edge S4), the workpiece data must also be rotated by one station:

- If a finished part is in the "Welding" station, it must be reset and set as a workpiece in the transfer station.
- If a finished part is in the "Drilling" station, it must be reset and set as a raw part in the "Welding" station.
- If a workpiece is in the "Magazine" station, it must be reset and set as a raw part in the "Drilling" station.

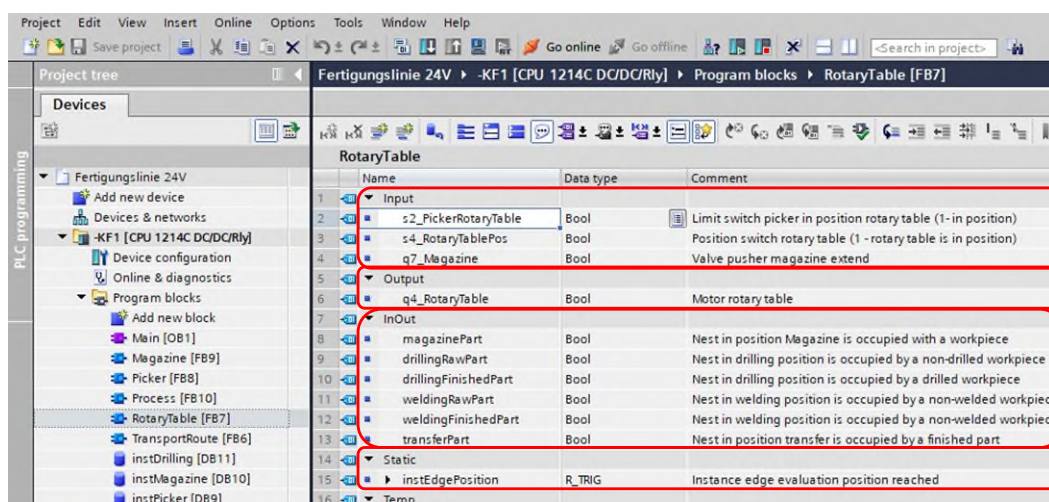
## Planning and implementing automatic processes - Exercise: Create function block for rotary table [ST / SCL]

### Procedure:

1. Create a new function block, select the desired programming language and assign a meaningful name:



2. Declare variables for the sensors and actuators, the variables for transferring the workpiece information and the instance for edge evaluation in the function block interface:



Planning and implementing automatic processes - Exercise: Create function block for rotary table [ST / SCL]

3. Program the control of the turntable (Q4):

```

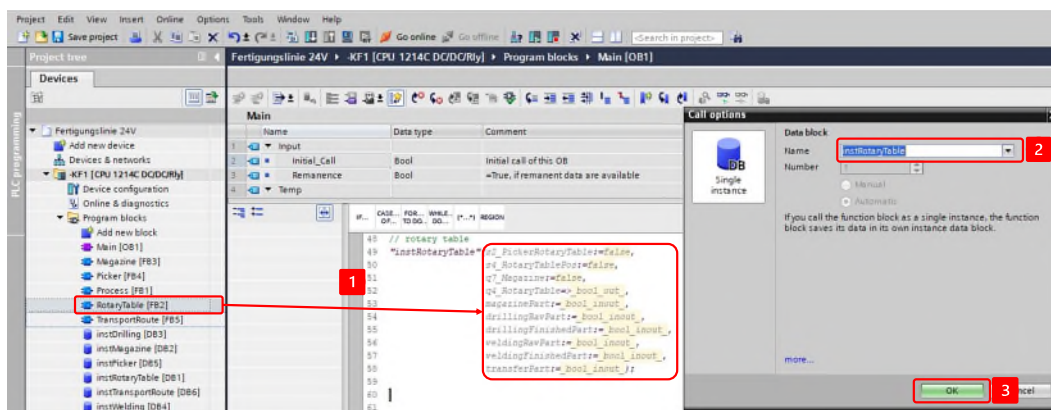
1 // control the rotary table motor
2 #q4_RotaryTable :=
3 NOT #q7_Magazine AND NOT #s2_PickerRotaryTable // permanent interlocks
4 AND (NOT #s4_RotaryTablePos // rotate until back into position
5 OR ((NOT #drillingRawPart AND NOT #weldingRawPart AND NOT #transferPart) // workpiece status locks rotate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

4. Program the edge evaluation (S4) and the rotation of the workpiece data:

```

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. Call up the function module in "MAIN" and create an instance:



Planning and implementing automatic processes - Exercise: Create function block for rotary table [ST / SCL]

6. Connect the module interface:

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 pane shows the 'Main' program block with a table of variables and a code editor. The code editor contains the following SCL code for the 'rotary table' function block:

```
47
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. Commission the system in a structured manner with the aid of a commissioning protocol.



Solution



Planning and implementing automatic processes - Exercise: Create function block for rotary table [ST / SCL]

**Solution:**

The solution can be found in the TIA Portal project "Fertigungslinie\_06\_Drehtisch\_SCL.zap17".