

24V production line

Planning and implementing automatic processes



Table of contents

10 Pla	an 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 stat 12	tion
10.5 Trans	Exercise: Convert GRAFCET sequence chain into program code [FBD] sport route	- 15
10.6 / SCL	Exercise: Converting the GRAFCET sequence chain into program code - Transport route	e [ST 23
10.7 Conv	Exercise: Convert GRAFCET sequence chain into program code [FBD] /erter	- 31
10.8 Conv	Exercise: Convert GRAFCET sequence chain into program code [ST / Soverter	CL] - 39
10.9 Maga	Exercise: Convert GRAFCET sequence chain into program code [FUP] azine	- 46
10.10 Maga	Exercise: Convert GRAFCET sequence chain into program code [ST / So azine	CL] - 53
10.11 Proce	- Exercise: Convert GRAFCET sequence chain into program code [FUP] essing station	60
10.12 Proce	Exercise: Convert GRAFCET sequence chain into program code [ST / SC essing station	CL] - 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.



Planning and implementing automatic processes - Exercise: Planning the process chain with GRAFCET - transport route

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



Planning and implementing automatic processes - Exercise: Planning the process chain with GRAFCET - transport route



GROLLMUS

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.





Planning and implementing automatic processes - Exercise: Planning the sequence chain with GRAFCET - converter

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.

fischertechnik 📼

Planning and implementing automatic processes - Exercise: Planning the sequence chain with GRAFCET - converter



Solution



Planning and implementing automatic processes - Exercise: Planning the sequence chain with GRAFCET - converter

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. I 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.

Planning and implementing automatic processes - Exercise: Planning the process chain with GRAFCET - Magazine



Solution



Planning and implementing automatic processes - Exercise: Planning the process chain with GRAFCET - Magazine





GROLLMUS

fischertechnik 📼

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



Planning and implementing automatic processes - Exercise: Planning the process chain with GRAFCET - Processing station





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



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:

Project tree 0	🛛 🖣 Fertigungslinie 24V 🕨 -KF1 [0	Fertigungslinie 24V → -KF1 [CPU 1214C DC/DC/Rly] → Program blocks → TransportRoute [FB6]							
Devices									
	🖻 📑 🖉 👻 👘 🗮 🗖 🚍	1	1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2						
2	TransportRoute								
▼ 🔄 Fertigungslinie 24V	Name	Data type	Comment						
Add new device	1 📲 🕶 Input								
Devices & networks	2 📲 init	Bool	Request to initialize step chain						
- KF1 [CPU 1214C DC/DC/Rly]	3 📲 🔹 s1_PickerConveyor	Bool	Limit switch picker in position conveyor belt (1- in position)						
Device configuration	4 📲 s2_PickerRotaryTable	Bool	Limit switch picker in position rotary table (1- in position)						
🖞 Online & diagnostics	5 🕣 = s3_PusherHomePos	Bool	Limit switch pusher in home position (1 - in home position)						
🖛 🛃 Program blocks	6 - b1_Conveyor	Bool	Light barrier belt (0 - workpiece placed on belt)						
Add new block	7 📲 b2_PalletEmpty	Bool	Light barrier pallet top (1 - pallet empty)						
🖀 Main [OB1]	8 - b3_PalletNotPresent	Bool	Pallet light barrier below (0 - pallet present)						
TransportRoute [FB6]	9 📲 🔻 Output								
instTransportRoute [DB8]	10 - q3_Pusher	Bool	Motor pusher						
Workpieces [DB13]	11 - q5_Conveyor	Bool	Motor conveyor belt						
System blocks	12 - q6_Separator	Bool	Close valve separator						
Technology objects	13 - InOut								



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:

Project tree	Fertigungslinie 24V + -KF1 [CPU 1214C DC/DC/Rly] + Program blocks + TransportRoute [FB6]							
Devices								
	🚍 🗖 🚄 💄 🌾 🏷 🖏		日 😥 🍋 📞 補 🤬 🏣 🥵 🕻 🖬 🏭 🗒					
	TransportRoute	<u></u>						
💌 🧾 Fertigungslinie 24V	Name	Data type	Comment					
Add new device	15 🕣 💌 Static							
Devices & networks	16 - SR01	Bool	Step marker step 1 active					
-KF1 [CPU 1214C DC/DC/Rly]	17 - SR02	Bool	Step marker step 2 active					
Device configuration	18 📲 🔹 SR03	Bool	Step marker step 3 active					
🖞 Online & diagnostics	19 📲 🔹 SR04	Bool	Step marker step 4 active					
 Program blocks 	20 📲 🛎 SR05	Bool	Step marker step 5 active					
Add new block	21 - SR06	Bool	Step marker step 6 active					
Hain [OB1]	22 - SR07	Bool	Step marker step 7 active					
TransportRoute [FB6]	23 - SR08	Bool	Step marker step 8 active					
instTransportRoute [DB8]	24 - SR09	Bool	Step marker step 9 active					
Workpieces [DB13]	25 📲 🔹 SR10	Bool	Step marker step 10 active					
System blocks	26 🕣 = 🕨 instTransportTime	TON_TIME	Instance transport time					
Technology objects	27 📲 🔹 🕨 instSeparatorClose	TON_TIME	Instance Time Close separator					
External source files	28 📲 🔹 🕨 instSeparatorOpen	TON_TIME	Instance Time Open Separation Segment					
PLC tags	29 📲 🔹 🕨 counter	CTU_INT	Instance piece counter on tape					
PLC data types	30 🔄 = 🕨 instEdgeS3	R_TRIG	Instance edge evaluation pusher in home positio					
Watch and force tables	31 🕣 🔻 Temp							

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.





GROLLMUS

fischertechnik 📼







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:



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.

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





Solution



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



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:

Project tree 🛛 🛛	Fertigungslinie 24V → -KF1 [CPU 1214C DC/DC/Rly] → Program blocks → TransportRoute [FB6]									
Devices										
1 III III III III III III III III III I	🖻 🗟 🖓 🤨 🔮 🐛 🗮 🚍 🚍	成 & 솔 칼 🐂 臣 🖻 🗩 웹 ± 웹 ± 템 발 🕑 % 66 셴 앤 뉴 🍄 🖨 팬 뷰 님 님 📗								
	TransportRoute									
 Fertigungslinie 24V 	Name	Data type	Comment							
Add new device	1 📲 🔻 Input									
Devices & networks	2 📲 init	Bool	I Request to initialize step chain							
 -KF1 [CPU 1214C DC/DC/Rly] 	3 📲 s1_PickerConveyor	Bool	Limit switch picker in position conveyor belt (1- in position)							
Device configuration	4 📲 s2_PickerRotaryTable	Bool	Limit switch picker in position rotary table (1-in position)							
🖞 Online & diagnostics	5 📶 = s3_PusherHomePos	Bool	Limit switch pusher in home position (1 - in home position)							
🖛 🙀 Program blocks	6 - b1_Conveyor	Bool	Light barrier belt (0 - workpiece placed on belt)							
Add new block	7 - b2_PalletEmpty	Bool	Light barrier pallet top (1 - pallet empty)							
Main [OB1]	8 a b3_PalletNotPresent	Bool	Pallet light barrier below (0 - pallet present)							
TransportRoute [FB6]	9 - Output									
instTransportRoute [DB8]	10 - q3_Pusher	Bool	Motor pusher							
Workpieces [DB13]	11 - q5_Conveyor	Bool	Motor conveyor belt							
System blocks	12 - q6_Separator	Bool	Close valve separator							
Technology objects	13 - InOut									



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:

Project Edit View Insert Online Option	ns Tools Window Help 이 ± (여 호 급 만 대 말 다)	🗊 Go online 🖉 Go o	ffline 🎄 🖪 🐨 🗶 🖃 🛄 <eþrech in="" project<="" th=""></eþrech>			
Project tree 🔲 🕯	Fertigungslinie 24V → -KF1 [C	PU 1214C DC/DC/RI	/] > Program blocks > TransportRoute [FB5]			
Devices						
B	# # } ± % ⊨ # # #	😥 🥙 💊 🖑 🖗	· · · ◆ ← = = = = = = + • • • • • • •			
2	TransportRoute					
Fertigungslinie 24V	Name	Data type	Comment			
Add new device	15 📲 🔻 Static					
Devices & networks	16 🕣 🔹 stepnumber	Int	Index variable (number of the active step)			
E + KF1 [CPU 1214C DC/DC/Rly]	17 - instTransportTime	instTransportTime TON_TIME Instance transport time				
Device configuration	18 🕣 🔹 🕨 instSeparatorClose	TON_TIME	Instance Time Close separator			
🖞 Online & diagnostics	19 📹 🔹 🕨 instSeparatorOpen	TON_TIME	Instance Time Open separator			
 Program blocks 	20 - counter	Int	Instance piece counter on tape			
Add new block	21 🚭 🔹 🕨 instEdgeS3	R_TRIG	Instance edge evaluation pusher in home position			
Hain [OB1]	22 🕣 🔻 Temp					
TransportRoute [FB5]	23 Add new>					

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:
- i

i

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.



```
9 // step chain
10 CASE #stepnumber OF
      1: // step 1 - Initstep
11
12
           // Reset all actions
13
           #q3 Pusher := FALSE;
           #q5_Conveyor := FALSE;
14
15
           #q6_Separator := FALSE;
16
          // transitions
17
                                            // pallet inserted
          IF NOT #b3_PalletNotPresent
18
19
               AND #b2 PalletEmpty
                                                  // pallet empty
          THEN
20
21
            #stepnumber := 2;
                                             // next step
22
          END IF;
23
24
      2: // step2 - move Pusher to home position
25
           // actions
           #q3_Pusher := NOT #s3_PusherHomePos // move pusher to home position
26
27
           AND NOT #b3 PalletNotPresent;
                                                  // pallet inserted
28
29 白
                                                  // if pusher is activated
          IF #q3 Pusher THEN
30
             #counter := 0;
                                                  // reset counter
31
          END IF;
32
          // transitions
33
34 白
          IF #s3 PusherHomePos THEN
                                                 // pusher is in home position
35
               #stepnumber := 3;
                                                  // next step
36
          END IF;
69 6: // step 6 - count up
70
         // actions
          #counter := #counter + 1;
71
                                            // count up workpiece
72
         IF #counter < 3 THEN
73 🗄
                                              // if counter is smaller than 3
 74
              #stepnumber := 2;
                                             // jump to 2
75
           ELSE
76
           #stepnumber := 7;
                                             // Start discharge
77
         END_IF;
78
     7: // step 7 - open separator
79
          // actions
80
81
         #q6_Separator := FALSE;
                                            // open separator
82
83
          // transitions
84
           IF #instSeparatorOpen.Q
                                             // separator opened
85
              AND #b2 PalletEmpty
                                             // pallet is empty
86
              AND NOT #b3 PalletNotPresent
                                             // palette is present
87
           THEN
88
             #stepnumber := 8;
                                             // next step
89
          END IF;
90
91
     8: // step 8 - drive pusher
92
         // actions
93
          #q3_Pusher := TRUE;
                                             // drive pusher
94
95 🗄
                                             // if pusher driven
           IF #q3 Pusher THEN
96
            #counter := 0;
                                              // reset counter
97
          END IF;
98
99
          // transitions
                                             // pusher back in home position arrived
100 白
         IF #instEdgeS3.Q THEN
101
             #stepnumber := 2;
                                              // jump to 2
102
          END IF;
103 END_CASE;
```

GROLLMUS

fischertechnik 📼

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:







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



Solution:

The solution can be found in the TIA Portal project "Fertigungslinie_02_Transportstrecke_SCL.zap17".



R

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



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:

Project tree	🔨 Fe	Fertigungslinie 24V + -KF1 [CPU 1214C DC/DC/Rly] + Program blocks + Picker [FB8]							
Devices									
1 III III III III III III III III III I	e 16	ă _I că	# # L E E = 🗩	웰± 월± 월± 드 😥	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				
		Pick	ter						
 Fertigungslinie 24V 		1	Name	Data type	Comment				
Add new device	1	-	 Input 						
Devices & networks	2	-	 init 	Bool	Request to initialize step chain				
-KF1 [CPU 1214C DC/DC/Rly] Device configuration		-	 s1_PickerConveyor 	Bool	Limit switch picker in position conveyor belt (1- in position)				
		-	s2_PickerRotaryTable	Bool	Limit switch picker in position rotary table (1- in position)				
🖞 Online & diagnostics	5	-01	 s4_RotaryTablePosition 	Bool	Position switch rotary table (1 - rotary table is in position)				
🔻 🕁 Program blocks	6	-	 b1_Conveyor 	Bool	Light barrier belt (0 - workpiece placed on belt)				
Add new block	7	-	 q5_Conveyor 	Bool	Motor conveyor belt				
Hain [OB1]	8		 Output 						
 Picker (FB8) TransportRoute (FB6) instPicker (D89) 		-	 q1_PickerRotaryTable 	Bool	Move picker towards rotary table				
		-	 q2_PickerConveyor 	Bool	Drive picker towards conveyor belt				
		-00	q8_Vacuum	Bool	Valve vacuum suction cup on				
instTransportRoute [DB8]	12	-00	 InOut 						
Workpieces [DB13]	13	-0	 pieceReady 	Bool	Workpiece is in the rotary table ready for collection				
System blocks	14		Add new>						



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:

Project tree		Fe	rtig	ingsl	inie 24V → -KF1 [CPU 1	1214C DC/DC/Rly] >	Program blocks > Picker [FB8
Devices							
留	E =	iði	i joji	÷	🛃 🐛 🖿 🚍 💬	월: 월: 월: 달	😥 🥙 👡 🕮 🖼 🖷 🤤 🕻
			Picl	er			
- Fertigungslinie 24	V	-		Name		Data type	Comment
Add new devic		15	-	• St	atic		
Devices & netv	orks	16	-		SR01	Bool	Step marker step 1 active
-KF1 [CPU 121-	C DC/DC/Rly]	17	-0		SR02	Bool	Step marker step 2 active
Device cont	iguration	18	-0		SR03	Bool	Step marker step 3 active
😵 Online & dia	ignostics	19	-		SR04	Bool	Step marker step 4 active
🔻 🙀 Program blo	cks	20	-		SR05	Bool	Step marker step 5 active
Add new	block	21	-		SR06	Bool	Step marker step 6 active
🏰 Main [Oi	11]	22	-0		instVacuumOn	TON_TIME	
🟩 Picker (F	B8]	23	-0		instVacuumOff	TON_TIME	
🛃 Transpo	tRoute [FB6]	24	-00	 Te 	mp		

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





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

Pr	roject Edit View Insert Online Optio	ns Tools Window Help	
E	🛉 🎦 🖬 Save project 🛛 昌 🐰 💷 🛍 🗙	"이 ± (여 ± 🗟 🗓 🕼 🖳 🐺 💋 Go online 🖉 Go offline 🎎 📑 🗶 🚍	Cepro
	Project tree 🛛 🖉 🕯	Fertigungslinie 24V → -KF1 [CPU 1214C DC/DC/Rly] → Program blocks → Pi	cker [FB8]
	Devices		
	1 I I I I I I I I I I I I I I I I I I I		≿ <u>(= </u> _)
5			
-E	 Fertigungslinie 24V 		
me	Add new device	s >=1 127 → -01 → -[=]	
ogr	Devices & networks	 Maturada 7: Wardeniace is in the saturatable ready for collection 	
pr	 -KF1 [CPU 1214C DC/DC/Rly] 	Network 7: Workpiece is in the rotary table ready for collection	
FC	Device configuration	Comment	
	😵 Online & diagnostics	n #pieceDeady	
	🔻 🚘 Program blocks	#q8_Vacuum — R	
	Add new block	#SR04	
	Main [OB1]		
	Picker [FB8]		
	TransportRoute [FB6]	 Network 8: Move picker towards rotary table 	
	instricker [DB9]	Comment	
	Workpiecer [DB13]		
	System blocks	*5 001	
	Technology objects	#st_Picker >=1	
	External source files	Conveyor — 🔅	
	PLC tags		
	PLC data types	#SR03	
	 Watch and force tables 	#s2_PickerRotary #q1_PickerRotary	
	Online backups	able	
	Traces	Position + +	
	OPC UA communication		
	 Device proxy data 	Network 9: Drive picker towards conveyor belt	
	Program info		
	E PLC alarm text lists	Comment	
	Local modules	8	
	Security settings	#SR01	
	Cross-device functions	#s2_PickerRotary	
	Common data	#q5_Conveyor -0 *	
	Documentation settings		
	Languages & resources	#SR05	
	Version control interface	#s1_Picker #q2_Picker	
	Online access	Conveyor - Conveyor	
	Card Reader/USB memory	#of_conveyor	

GROLLMUS
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:



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.



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

6

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.

Fe	rtig		slinie 24V → -KF1 [CPI	J 1214C DC/DC/Rly]	Program bloc	ks Workpieces [DB13]		
10	1	1	🖌 🏷 🔚 😚 Keep ac	tual values 🔒 Sna	pshot 🐂 🖳 Co	ppy snapshots to start values 🔹 🖳 Load start values as actual		
	Wo	orkp	ieces					
		Na	me	Data type	Monitor value	Comment		
1	-	•	Static					
2	-01		magazinePart	Bool	FALSE	Nest in position Magazine is occupied with a workpiece		
3	-0		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	-0		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	-01		transferPart	Bool	TRUE	Nest in position transfer is occupied by a finished part		





Solution



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



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:

Project tree	Fe	ertig	ungslinie 24V 🔸 -KF1 [CPI	U 1214C DC/DC/Rly] + Pro	ogram blocks 🕨 Picker [FB8]
Devices					
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	ьđ	ă _I dă	🥩 🐮 🐛 🖿 🚍 🖉	🗩 🗶 ± 🚟 ± 🖼 😫	(°℃偏偏正会 (*田田田市下下)
		Picl	ker		
 Fertigungslinie 24V 			Name	Data type	Comment
Add new device	1	-00	 Input 		
Devices & networks	2	-	 init 	Bool	Request to initialize step chain
KF1 [CPU 1214C DC/DC/Rly]	3	-	s1_PickerConveyor	Bool	Limit switch picker in position conveyor belt (1-in position
Device configuration	4	-	s2_PickerRotaryTable	Bool	Limit switch picker in position rotary table (1- in position)
😵 Online & diagnostics	5	-00	s4_RotaryTablePosition	Bool	Position switch rotary table (1 - rotary table is in position)
🔻 🙀 Program blocks	6	-	b1_Conveyor	Bool	Light barrier belt (0 - workpiece placed on belt)
Add new block	7	-00	q5_Conveyor	Bool	Motor conveyor belt
🖀 Main [OB1]	8	-0	 Output 		
Picker [FB8]	9	-	q1_PickerRotaryTable	Bool	Move picker towards rotary table
TransportRoute [FB6]	10	-0	q2_PickerConveyor	Bool	Drive picker towards conveyor belt
📒 instPicker [DB9]	11	-00	q8_Vacuum	Bool	Valve vacuum suction cup on
instTransportRoute [DB8]	12	-00	 InOut 		
Workpieces [DB13]	13	-	pieceReady	Bool	Workpiece is in the rotary table ready for collection
System blocks	14		Add news		



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:

Project Edit View Insert Online Option	s Tools Window Help St (** 🖥 🗓 🕼 🚇 📪 🌶	🖉 Goonline 🖉 Goo	ffline 🔐 🖪 🖪 🗶 🤜 📖 (Search
Project tree 🔲 🕻	Fertigungslinie 24V → -KF1 [Cf	U 1214C DC/DC/RI	/] • Program blocks • Picker [FB4]
Devices			
	[# 관 달 보 5 , 1 월 월 보	😥 🎨 💊 🖑 🖗	· ● ◆ ← 田 田 田 田 指 ┣ ¢
	Picker		
Fertigungslinie 24V	Name	Data type	Comment
Add new device	14 🕣 🕶 Static		
Devices & networks	15 🕣 = stepnumber	Int	Index variable (number of the active step)
	16 🕣 = 🕨 instVacuumOn	TON_TIME	Instance delaytime vacuum on
Device configuration	17 🕣 🔹 🕨 instVacuumOff	TON_TIME	Instance delaytime vacuum off
& Online & diagnostics	18 🕣 🔻 Temp		
 Program blocks 	19 Add new>		

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
           // actions
10
           #q1 PickerRotaryTable := #s1 PickerConveyor;
11
           #q2_PickerConveyor := #s2_PickerRotaryTable;
12
13
14
           #q8 Vacuum := FALSE; // reset action
15
16
           // transitions
           IF NOT #s1 PickerConveyor
17 白
               AND NOT #s2 PickerRotaryTable
18
19
           THEN
20
               #stepnumber := 2;
21
           END IF;
22
       2: // step 2 - waiting for workpiece
23
24
           // actions
25
26
           // transitions
27 白
           IF #s4 RotaryTablePosition
28
               AND #pieceReady
29
           THEN
30
               #stepnumber := 3;
31
           END_IF;
```



i

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



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:

Project tree	■ Fertigungslinie 24V	[CPU 1214C DC/D	C/Rly] → Program blocks → Main [OB1]			
Devices						
11 II I	* # # } ± % E # #	i ± 😥 🥙 💊 🦪	(1) 11 11 11 11 11 11 11 11 11 11 11 11 1			
	Main					
 Fertigungslinie 24V 	Name	Data type	Comment			
Add new device	1 📶 🔻 Input					
Devices & networks	2 💶 = Initial_Call	Bool	Initial call of this OB			
-KF1 [CPU 1214C DC/DC/Rly]	3 😋 🖷 Remanence	Bool	=True, if remanent data are available			
Device configuration	4 🚤 🕶 Temp					
Solution Contine & diagnostics						
🔻 🕁 Program blocks		CASE FOR WHILE	(**) REGION			
Add new block						
📲 Main [OB1]		3 // picker				
🔹 Picker [FB4]		s - instricker (al DickerConveyor - "Sl"			
TransportRoute [FB5]		6	s2 PickerRotaryTable := "52".			
instPicker [DB5]		7	s4 RotaryTablePosition := "S4".			
instTransportRoute [DB6]	1	.8	bl Conveyor := "Bl",			
Workpieces [DB13]	1	.9	q5_Conveyor := "Q5",			
System blocks	2	0	<pre>ql_PickerRotaryTable => "Q1",</pre>			
Technology objects	2	1	q2_PickerConveyor => "Q2",			
External source files	2	2	q8_Vacuum => "Q8",			
PIC taos	2	3	<pre>pleceReady := "Workpieces".transferPart);</pre>			



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.

6

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.

Fe			islinie 24V → -KF1 [CPL	J 1214C DC/DC/RIy]	Program block	s Vorkpieces [DB13]			
1	1	1	🖌 🛃 🔚 😤 Keep act	tual values 🧧 Sna	pshot 🙀 🖏 Coj	py snapshots to start values 🛛 😹 Load start values as actual			
	Wo	orkp	ieces						
		Na	me	Data type	Monitor value	Comment			
1	-	•	Static						
2	-01		magazinePart	Bool	FALSE	Nest in position Magazine is occupied with a workpiece			
3	-0		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	-0		weldingRawPart	Bool	FALSE	Nest in welding position is occupied by a non-welded workpiece			
6	-	🗉 = weldingFinishedPart Bool		Bool	FALSE	Nest in welding position is occupied by a non-welded workpiece			
7	-01		transferPart	Bool	TRUE	Nest in position transfer is occupied by a finished part			





Solution



Solution:

The solution can be found in the TIA Portal project "Fertigungslinie_03_Umsetzer_SCL.zap17".

GROLLMUS

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



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:

Project tree	< Fe	Fertigungslinie 24V → -KF1 [CPU 1214C DC/DC/Rly] → Program blocks → Magazine [FB9]						
Devices								
8	i i							
		Mag	azine					
 Fertigungslinie 24V 			Name	Data type	Comment			
Add new device	1		 Input 					
Devices & networks	2	-	init init	Bool	Request to initialize step chain			
 KF1 [CPU 1214C DC/DC/Rly] 	3	-	s4_RotaryTablePosition	Bool	Position switch rotary table (1 - rotary table is in position			
Device configuration	4		b4_Magazine	Bool	Light barrier magazine (0 - workpiece present)			
😨 Online & diagnostics	5	-	 Output 					
🔻 🛃 Program blocks	6	-	q7_Magazine	Bool	Valve pusher magazine extend			
Add new block	7	-01	 InOut 					
Main [OB1]	8	-	nestOccupied	Bool	Workpiece lies on the rotary table			
- Magazine (FB9)	9	-01	 Static 		AND STREET STREETS AND ADDRESS STREETS AND ADDRESS STREETS			



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

Project tree	Fertigungslinie 24V → -KF1 [CPU	1214C DC/DC/Riy] +	Program blocks 🔸 Magazine [FB9]
Devices			
8	🗐 📰 🚍 🚚 🐑 🐑 Ku Ku	8: 8: 8: E	· · · · · · · · · · · · · · · · · · ·
	Magazine		
▼ 🔄 Fertigungslinie 24V	Name	Data type	Comment
Add new device	9 📶 🔻 Static		
📩 Devices & networks	10 - SR01	Bool	Step marker step 1 active
 -KF1 [CPU 1214C DC/DC/Rly] 	11 - SR02	Bool	Step marker step 2 active
Device configuration	12 - SR03	Bool	Step marker step 3 active
🛂 Online & diagnostics	13 - instDelay84	TON_TIME	Instance delay time magazine occupied (84
 Program blocks 	14 📹 = 🕨 instDelayPusher	TON_TIME	Instance Waiting Time pusher Extended (Q7)
Add new block	15 🕣 🔻 Temp		
Hain [OB1]	16 Add new>		
Magazine [FB9]	17 JI T Constant		

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



GROLLMUS

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



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:



i

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.

8. 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.

Fe	rtig		gslinie 24V 🔸 -KF1 [CPU	1214C DC/DC/Rly]	Program blocks	s Vorkpieces [DB13]
_					1. 10 10 c	
3	1	1	🛰 🕼 🖿 💽 Keep act	ual values 📕 Snap	osnot 🥆 🥆 Cop	sysnapshots to start values 📓 📓 Load start values as actual
	We	ork	pieces			
		Na	ame	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

GROLLMUS



Solution



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



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:

	Project tree 🛛 🕄 📢			Fertigungslinie 24V → -KF1 [CPU 1214C DC/DC/Rly] → Program blocks → Magazine [FB9]						
	Devices									
	1 I I I I I I I I I I I I I I I I I I I	ið	Ř H	X I	? ? L E E = 91	1 2 2 1 1 1 1 1	1	◎ 6 6 6 6 1 9 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1		
			Magazine							
	 Fertigungslinie 24V 			Na	me	Data type		Comment		
	Add new device	1	-	6	Input					
š	Devices & networks	2	-0		init	Bool		Request to initialize step chain		
	 -KF1 [CPU 1214C DC/DC/Rly] 	3	-0		s4_RotaryTablePosition	Bool	1	Position switch rotary table (1 - rotary table is in position)		
¥.	Device configuration	4	-	l.	b4_Magazine	Bool	1	Light barrier magazine (0 - workpiece present)		
	😨 Online & diagnostics	5	-	F	Output					
	Program blocks	6	-		q7_Magazine	Bool	,	Valve pusher magazine extend		
	Add new block	7	-0	•	InOut					
	🖀 Main [OB1]	8	-		nestOccupied	Bool	1	Workpiece lies on the rotary table		
	- Magazine [FB9]	9	-60	•	Static			CALLS NOT DE LES SUCCESSION DE LES SUCCESSIONS		



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

Project Edit View Insert Online Option	ns Tools Window Help 비) ± (제 ± 🐻 🛄 🌆 🚆 🐺 💋 Go	online 🖉 Go offline	å? 🖪 🖪 🗶 🖃 💷 <earch in="" proje<="" th=""></earch>
Project tree 🛛 🔳 🖣	Fertigungslinie 24V 🕨 -KF1 [CPU 12	14C DC/DC/Rly] >	Program blocks Magazine [FB3]
Devices			
	** * * • * * * # # # # #	eo 6a 68 68 Ta	응 다 ㅋ ㅋ 원 나 나 나 아 이 이
2	Magazine		
🗧 💌 📑 Fertigungslinie 24V	Name	Data type	Comment
Add new device	9 💶 🔻 Static		
Devices & networks	10 💷 = stepnumber	Int	Index variable (number of the active step)
	11 💷 = 🕨 instDelay84	TON_TIME	Instance delay time magazine occupied (B4)
Device configuration	12 📶 = 🕨 instDelayPusher	TON_TIME	Instance Waiting Time Pusher Extended (Q7)
Se Online & diagnostics	13 📶 👻 Temp		
 Program blocks 	14 Add new>		
Add new block	15 🕣 🐨 Constant		
Main [OB1]	16 Add new>		
🚁 Magazine [FB3]			

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

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



i



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.

6

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

Fe			linie 24V → -KF1 [CPL	J 1214C DC/DC/RIy	/] 🕨 Program b	locks + Workpieces [DB13]
		× .	Keep act	tual values 🔒 Sr	aoshot 🛤 🛤	Copy spanshots to start values R. R. Load start values as actual
-	Wo	orkpie	eces			
		Nam	e	Data type	Monitor value	Comment
1	-	* 5	tatic			
2	-00		magazinePart	Bool	FALSE	Nest in position Magazine is occupied with a workpiece
3	-00		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	-00		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



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



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:

Project tree	🔹 Fe	rtig	ungslinie 24V 🔸 -KF1 [CPI	J 1214C DC/DC/Rly] >	Program blocks Process [FB10]
Devices					
	101	10	(🔊 🥐 🔍 🖿 🚍 🛯	🔊 📲 ± 📲 ± 🔚 🖿	12 で 6 徳 徳 19 李 6 三 三 井 4 1
		Pro	cess		
 Fertigungslinie 24V 			Name	Data type	Comment
📫 Add new device	1	-01	 Input 		
Devices & networks	2	-01	 init 	Bool	Request to initialize step chain
▼ 🕞 -KF1 [CPU 1214C DC/DC/Rly]	з	-00	s4_RotaryTablePos	Bool	Position switch rotary table (1 - rotary table is in position)
Device configuration	4	-00	processingTime	Time	Duration of processing in the station
🗓 Online & diagnostics	5	-01	 Output 		
Program blocks	6	-01	process	Bool	Controls the actuator for process
Add new block	7	-01	▼ InOut		
- Main [OB1]	8	-01	rawPart	Bool	Workpiece has not yet been processed
Process [FB10]	9	-0	finishedPart	Bool	Workpiece was processed
RotaryTable [FB7]	10	-00	▼ Static		



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

Project tree 🛛 🕮	Ferti	gungslinie 24V → -KF1 [CPU	1214C DC/DC/R	iy] → Program blocks → Process [FB
Devices				
11 II I	1 iối 1	X 🖈 👻 🐛 🖿 🚍 💬	2 ± 2 ± 2	± 🖃 🌮 📞 🧶 🐨 🕸 🗲
	P	ocess		
 Fertigungslinie 24V 		Name	Data type	Comment
Add new device	10 -	Static		
Devices & networks	11 -	SR01	Bool	Step marker step 1 active
 KF1 [CPU 1214C DC/DC/Rly] 	12 -	SR02	Bool	Step marker step 2 active
T Device configuration	13 -	sro3	Bool	Step marker step 3 active
😨 Online & diagnostics	14 -	instProcessingTime	TON_TIME	Instance for processing time
 Program blocks 	15 -	Temp		
Add new block	16	Add new>		
🖀 Main [OB1]	17 -	Constant		
Process [FB10]	18	Add new>		

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





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:





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



Ĵ

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.

8. 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.

1	Wo	🖡 🛃 🧮 🛅 Keep a orkpieces	ctual values 🧧 Sn	apshot 🐴 🖳 C	copy snapshots to start values 🛛 🕵 🖗 🛛 Load start values as actua
		Name	Data type	Monitor value	Comment
1	-0	▼ Static			
2	-	 magazinePart 	Bool	FALSE	Nest in position Magazine is occupied with a workpiece
3	-0	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	-01	 weldingRawPart 	Bool	FALSE	Nest in welding position is occupied by a non-welded workpiece
5	-01	weldingFinishedPart	Bool	FALSE	Nest in welding position is occupied by a non-welded workpiece
7	-01	transferPart	Bool	FALSE	Nest in position transfer is occupied by a finished part





Solution



Solution:

The solution can be found in the TIA Portal project "Fertigungslinie_05_Bearbeiten_FUP.zap17".

GROLLMUS

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



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:

📑 🔄 Save project 📑 🔏 💷 💷 🤉	(<mark>-)</mark> ± (*	🐂 🛨 (📲 🔟 🔛 🔛 🔛 🖉 Go online 🖉 Go orfline 👔 🖪 🕼 🗶 🖃 🛄 <earch in="" project<="" th=""></earch>								
Project tree	I									
Devices										
1 II I I I I I I I I I I I I I I I I I	Ka Ka	(高)(조) 왕 왕 4, 臣曰曰曰:3:3:4 월 1 월 2 월 2 월 2 월 2 월 2 월 2 월 2 월 2 월 2								
2	Proc									
📕 🖛 📋 Fertigungslinie 24V	N	lame	Data type	Comment						
Add new device	1 -	 Input 								
Devices & networks	2 📶 🖷	init	Bool	Request to initialize step chain						
KF1 [CPU 1214C DC/DC/Rly]	3 🕣 🗉	s4_RotaryTablePos	Bool	Position switch rotary table (1 - rotary table is in position)						
Device configuration	4 🕣 🖷	processingTime	Time	Duration of processing in the station						
V. Online & diagnostics	5 📶 🗖	Output								
🔻 🛃 Program blocks	6 🕣 🗉	process	Bool	Controls the actuator for process						
Add new block	7 📶 🗖	 InOut 								
- Main [OB1]	8 🕣 🖷	rawPart	Bool	Workpiece has not yet been processed						
Process [FB10]	9 📶 🖷	finishedPart	Bool	Workpiece was processed						
RotaryTable [FB7]	10 🕣	Static								



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

P	roject Edit View Insert Online Option	5	Tool	s pila	Window Help	of Go online of Go o				
-	Project tree	Fe	rtig	un	gslinie 24V → -KF1 [(CPU 1214C DC/DC/RI	y] > Program blocks > Process [FB1]			
	Devices									
	1 I I I I I I I I I I I I I I I I I I I	10	1	7	∋t 🖏 🗄 🗃 🖓	t 😥 🥙 💊 🖑 🖗	◎● ● 画 画 部 旨 】 ┣ 9			
Ð				Process						
Ē	 Fertigungslinie 24V 	-		Na	me	Data type	Comment			
am	Add new device	10	-	-	Static					
b	Devices & networks	11	-		stepnumber	Int	Index variable (number of the active step)			
a.	-KF1 [CPU 1214C DC/DC/Rly]	12	-		instProcessingTime	TON_TIME	Instance for processing time			
ž	Device configuration	13	-	*	Temp					
	😧 Online & diagnostics	14			<add new=""></add>					
	 Program blocks 	15	-	-	Constant					
	Add new block	16			<add new=""></add>					

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



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:



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.


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.

0

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

			gslinie 24V ♦ -KF1 [CP	U 1214C DODORIy	/] • Program b	locks • Workpieces [DB13]				
\$	1		🔩 🛃 🗮 📴 Keep ad	tual values 🔒 Sn	apshot 🙀 🖳	Copy snapshots to start values 🔹 🧟 Load start values as actual				
	Wo	ork	pieces	1						
		Na	ime	Data type	Monitor value	Comment				
1	-00	-	Static							
2	-0		magazinePart	Bool	FALSE	Nest in position Magazine is occupied with a workpiece				
3	-0		drillingRawPart	Bool	TRUE	Nest in drilling position is occupied by a non-drilled workpiece				
4	-0	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	-0		transferPart	Bool	FALSE	Nest in position transfer is occupied by a finished part				



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



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.





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, 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.



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:

	Project tree 🛛 🛛 🖣	Fe	ertigu	ngslinie 24V → -KF1 [CPU	1214C DC/DC/Rly]	Program blocks 🕨 RotaryTable [FB7]				
	Devices									
	1 I I I I I I I I I I I I I I I I I I I	ю́	кя й	2 2 4, 2 2 2 5	3 ± 2 ± 13 ± E					
PLC programming		RotaryTable								
	 Fertigungslinie 24V 			Name	Data type	Comment				
	Add new device	7	-00	 Input 						
	Devices & networks	2	-00	s2_PickerRotaryTable	Bool	Limit switch picker in position rotary table (1- in position)				
	 -KF1 [CPU 1214C DC/DC/Rly] 	3	-00	s4_RotaryTablePos	Bool	Position switch rotary table (1 - rotary table is in position)				
	Device configuration Online & diagnostics General Program blocks		-	q7_Magazine	Bool	Valve pusher magazine extend				
			-00	 Output 						
			-00	q4_RotaryTable	Bool	Motor rotary table				
	Add new block	7	-01	InOut						
	Main [OB1]	8	-	 magazinePart 	Bool	Nest in position Magazine is occupied with a workpiece				
	Magazine [FB9]	9	-00	 drillingRawPart 	Bool	Nest in drilling position is occupied by a non-drilled workpiece				
	Picker [FB8]	10	-00	 drillingFinishedPart 	Bool	Nest in drilling position is occupied by a drilled workpiece				
	Process [FB10]	11	-00	 weldingRawPart 	Bool	Nest in welding position is occupied by a non-welded workpiece				
	😅 RotaryTable [FB7]	12	-00	 weldingFinishedPart 	Bool	Nest in welding position is occupied by a non-welded workpiece				
	TransportRoute [FB6]	13	-00	transferPart	Bool	Nest in position transfer is occupied by a finished part				
	instDrilling [DB11]		-00	 Static 						
	🧧 instMagazine [DB10]	15	-	instEdgePosition	R_TRIG	Instance edge evaluation position reached				
	instPicker [DB9]	16	-570	• Temp						



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:





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.

78







Solution



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.



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:

	Project tree 🛛 🛛 🖣	Fe	ertig	Jun	gslinie 24V → -KF1 [CPU	1214C DC/DC/R	lly] ▶	Program blocks RotaryTable [FB7]	
	Devices								
	1 I I I I I I I I I I I I I I I I I I I	ьő	à d	K 3	0 2 4 1 1 1 1 1 5	3 = 2 + 13	± :=	でんほほうや ↓ = = = = = = = = = = = = = = = = = =	
PLC programming		RotaryTable							
	 Fertigungslinie 24V 			Na	me	Data type		Comment	
	Add new device	7	-00	(~	Input				
	Devices & networks	2	-0		s2_PickerRotaryTable	Bool		Limit switch picker in position rotary table (1- in position)	
	 -KF1 [CPU 1214C DC/DC/Rly] 	3	-0		s4_RotaryTablePos	Bool		Position switch rotary table (1 - rotary table is in position)	
	Device configuration	4	-00		q7_Magazine	Bool		Valve pusher magazine extend	
	S Online & diagnostics	5	-00	-	Output				
	🔻 🙀 Program blocks		-00		q4_RotaryTable	Bool		Motor rotary table	
	Add new block	7	-00	7	InOut				
	🖀 Main [OB1]	8	-00		magazinePart	Bool		Nest in position Magazine is occupied with a workpiece	
	🚭 Magazine [FB9]		-0		drillingRawPart	Bool		Nest in drilling position is occupied by a non-drilled workpiece	
	Picker [FB8]	10	-00		drillingFinishedPart	Bool		Nest in drilling position is occupied by a drilled workpiece	
	Process [FB10]	11	-0		weldingRawPart	Bool		Nest in welding position is occupied by a non-welded workpiece	
	🚭 RotaryTable [FB7]	12	-01		weldingFinishedPart	Bool		Nest in welding position is occupied by a non-welded workpiece	
	TransportRoute [FB6]	13	-00	-	transferPart	Bool		Nest in position transfer is occupied by a finished part	
	instDrilling [DB11]		-00	-	Static				
	🥫 instMagazine [DB10]	15	-		instEdgePosition	R_TRIG		Instance edge evaluation position reached	
	instPicker [DB9]	16	-670	*	Temp				



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 DIF #instEdgePosition.Q THEN
15
       #drillingRawPart := #magazinePart;
       #weldingRawPart := #drillingFinishedPart;
16
17
       #transferPart := #weldingFinishedPart;
18
19
       #magazinePart := #drillingFinishedPart := #weldingFinishedPart := FALSE;
20 END_IF;
```

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

Project Edit View Insert Online Options	Taals Window Help	💋 Goonline 🔊 Goo	ffline 🔐 🖪 🖪 💥 🗄 📖 Search in p	iroject»							
Project tree II 4	Fertigungslinie 24V + -KF1	[CPU 1214C DC/DC/RI	y] > Program blocks > Main [OB1]								
Devices	양 만 말 바 씨 臣 김 정 1 만 안 66 년 연 개 왕 64 권 편 위 님 일 만 64 년 과 약 약 36. Main										
Fertigungslinie 24V Add new device Device so networks Device configuration	Name 1	Bool Bool	Comment Initial call of this OB =True, if remanent data are available	Data bio DB Number Single instance	ck statementele 2						
Gonine & diagnostics Generation (Compare blocks Add new block Mad anew block Magazine (P83) Process (P84) Process (P84) BransportBioute (P85) instructing (D82) instructing (D84)	23 22 (A)	CALL, FOR, WELL, P7 DA, TOBO, SOL, P7 A, TOBO, SOL, P7 A, TOBO, SOL, P7 MinstRotaryTable 11 12 13 14 15 15 16 16 16 16 16 16 16 16 16 16	NEON () Distantionation faile, () Distantionation faile, () Distantionations, () Distantions, () Distantions, () Distantions, () Distantionation, () Distantionation, () Distantionation, () Distantionation, () Distantionation, () Distantionation, () Distantion, () Distanti	Moreer	OK 3 cci						



6. Connect the module interface:



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



Solution



Solution:

The solution can be found in the TIA Portal project "Fertigungslinie_06_Drehtisch_SCL.zap17".

GROLLMUS

87