SIEMENS

SCE Training Curriculum

Siemens Automation Cooperates with Education | 05/2017

TIA Portal Module 032-300

IEC Timers and IEC Counters
Multi-instances for SIMATIC S7-1500

Cooperates with Education
Automation
SIEMENS

Matching SCE trainer packages for these training curriculums

SIMATIC Controllers

- **SIMATIC ET 200SP Open Controller CPU 1515SP PC F and HMI RT SW**
  Order no.: 6ES7677-2FA41-4AB1
- **SIMATIC ET 200SP Distributed Controller CPU 1512SP F-1 PN Safety**
  Order no.: 6ES7512-1SK00-4AB2
- **SIMATIC CPU 1516F PN/DP Safety**
  Order no.: 6ES7516-3FN00-4AB2
- **SIMATIC S7 CPU 1516-3 PN/DP**
  Order no.: 6ES7516-3AN00-4AB3
- **SIMATIC CPU 1512C PN with Software and PM 1507**
  Order no.: 6ES7512-1CK00-4AB1
- **SIMATIC CPU 1512C PN with Software, PM 1507 and CP 1542-5 (PROFIBUS)**
  Order no.: 6ES7512-1CK00-4AB2
- **SIMATIC CPU 1512C PN with Software**
  Order no.: 6ES7512-1CK00-4AB6
- **SIMATIC CPU 1512C PN with Software and CP 1542-5 (PROFIBUS)**
  Order no.: 6ES7512-1CK00-4AB7

SIMATIC STEP 7 Software for Training

- **SIMATIC STEP 7 Professional V14 SP1 - Single license**
  Order no.: 6ES7822-1AA04-4YA5
- **SIMATIC STEP 7 Professional V14 SP1- Classroom license (up to 6 users)**
  Order no.: 6ES7822-1BA04-4YA5
- **SIMATIC STEP 7 Professional V14 SP1 - Upgrade license (up to 6 users)**
  Order no.: 6ES7822-1AA04-4YE5
- **SIMATIC STEP 7 Professional V14 SP1 - Student license (up to 20 users)**
  Order no.: 6ES7822-1AC04-4YA5

Note that these trainer packages are replaced with successor packages when necessary.
An overview of the currently available SCE packages is provided at: siemens.com/sce/tp

Continued training

For regional Siemens SCE continued training, get in touch with your regional SCE contact
siemens.com/sce/contact

Additional information regarding SCE

siemens.com/sce

Information regarding use

The SCE training curriculum for the integrated automation solution Totally Integrated Automation (TIA) was prepared for the program "Siemens Automation Cooperates with Education (SCE)" specifically for training purposes for public educational and R&D institutions. Siemens AG does not guarantee the contents.

This document is to be used only for initial training on Siemens products/systems. This means it can be copied in whole or part and given to those being trained for use within the scope of their training. Circulation or copying this training curriculum and sharing its content is permitted within public training and advanced training facilities for training purposes.

Exceptions require written consent from the Siemens AG contact: Roland Scheuerer roland.scheuerer@siemens.com.

Offenders will be held liable. All rights including translation are reserved, particularly if a patent is granted or a utility model or design is registered.

Use for industrial customer courses is expressly prohibited. We do not consent to commercial use of the training curriculums.

We wish to thank the TU Dresden, particularly Prof. Dr.-Ing. Leon Urbas, Michael Dziallas Engineering Corporation and all other involved persons for their support during the preparation of this training curriculum.

Table of contents

IEC TIMERS AND IEC COUNTERS MULTI-INSTANCES FOR SIMATIC S7-1500

1 Goal

In this chapter, you will become acquainted with the use of single instances and multi-instances for programming of the SIMATIC S7-1500 with the TIA Portal programming tool.

The module explains the various types of instance data blocks and shows step-by-step how to add IEC timers and IEC counters to a program block.

The SIMATIC S7 controllers listed in Chapter 3 can be used.

2 Prerequisite

This chapter builds on the FB programming with the SIMATIC S7 CPU1516F-3 PN/DP. You can use the following project for this chapter, for example:
032-200_FBProgramming_R1503.zap13

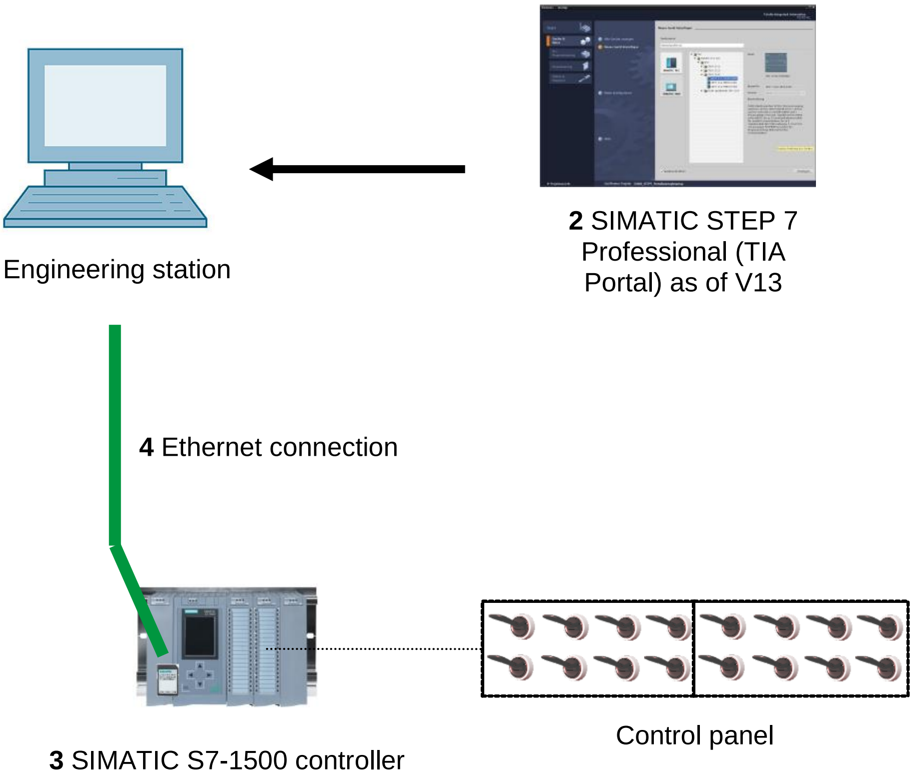
3 Required hardware and software

1 Engineering station: requirements include hardware and operating system (for additional information, see Readme on the TIA Portal Installation DVDs)

2 SIMATIC STEP 7 Professional software in TIA Portal – as of V13

3 SIMATIC S7-1500/S7-1200/S7-300 controller, e.g. CPU 1516F-3 PN/DP – Firmware as of V1.6 with memory card and 16DI/16DO and 2AI/1AO
Note: The digital inputs should be fed out to a control panel.

4 Ethernet connection between engineering station and controller


1 Engineering station
2 SIMATIC STEP 7 Professional (TIA Portal) as of V13
4 Ethernet connection
3 SIMATIC S7-1500 controller
Control panel

4 Theory

4.1 Instances and multi-instances in SIMATIC S7-1500

The call of a function block is referred to as an **instance**. An **instance** is assigned to every call of a function block and serves as a data memory. It stores the actual parameters and the static data of the function block.

The tags declared in the function block determine the structure of the instance data block.

Use of single instances and multi-instances

You can assign instances as follows:

Call as a **single instance**:

- A separate instance data block for each instance of a function block

Call as a **multi-instance**:

- One instance data block for several instances of one or more function blocks

4.1.1 *Instance data blocks/single instances*

The call of a function block that is assigned its own instance data block is called a **single instance**.

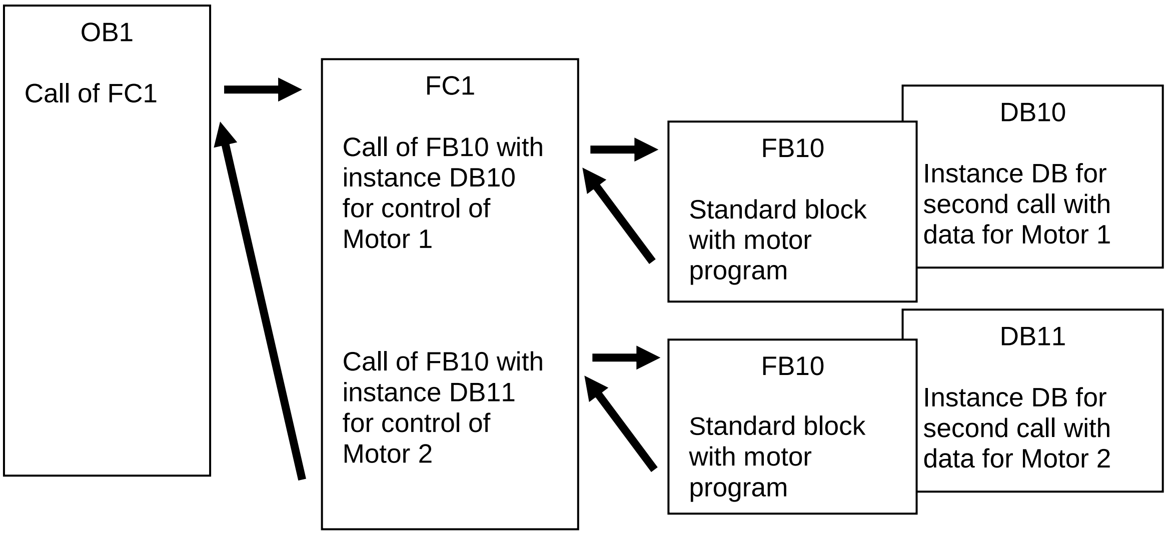
If the function block was created according to the rules for library-compatible standard blocks, it can also be called multiple times.

However, you must assign another instance data block for each call as a single instance.

Example of single instances:

The following figure shows the control of two motors using one function block FB10 and two different data blocks:

The different data for the individual motors, such as speed, acceleration time and total operating time, are saved in the instance data blocks DB10 and DB11.


OB1
Call of FC1
FC1
Call of FB10 with instance DB10 for control of Motor 1
Call of FB10 with instance DB11 for control of Motor 2
FB10
Standard block with motor program
FB10
Standard block with motor program
DB10
Instance DB for second call with data for Motor 1
DB11
Instance DB for second call with data for Motor 2

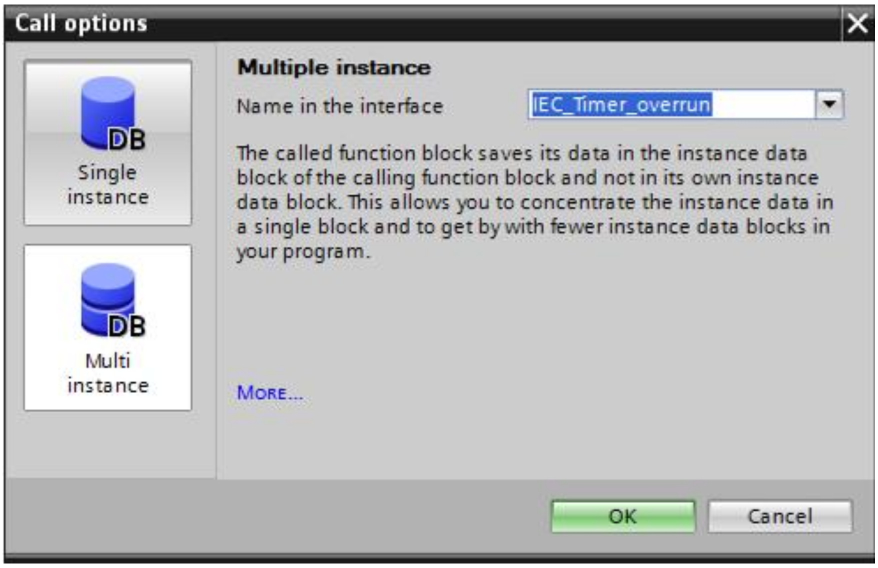
***Note:** Some commands, such as timers and counters, react like function blocks. When these are called, they also require an assigned memory area, e.g. in the form of an instance data block.*

4.1.2 Multi-instances

You may want to limit the number of data blocks used for instances or this may be necessary due to lack of memory in the utilized CPU.

If other function blocks, timers, counters, etc. that already exist will be called in a function block in your user program, you can call these other function blocks without separate (i.e., additional) instance DBs.

Simply select ‘**Multi-instance**’ for the call options:



Call options

Multiple instance

Name in the interface: IEC_Timer_overrun

The called function block saves its data in the instance data block of the calling function block and not in its own instance data block. This allows you to concentrate the instance data in a single block and to get by with fewer instance data blocks in your program.

***Notes:** Multi-instances enable a called function block to store its data in the instance data block of the calling function block.*

In this case, the calling block must always be a function block.

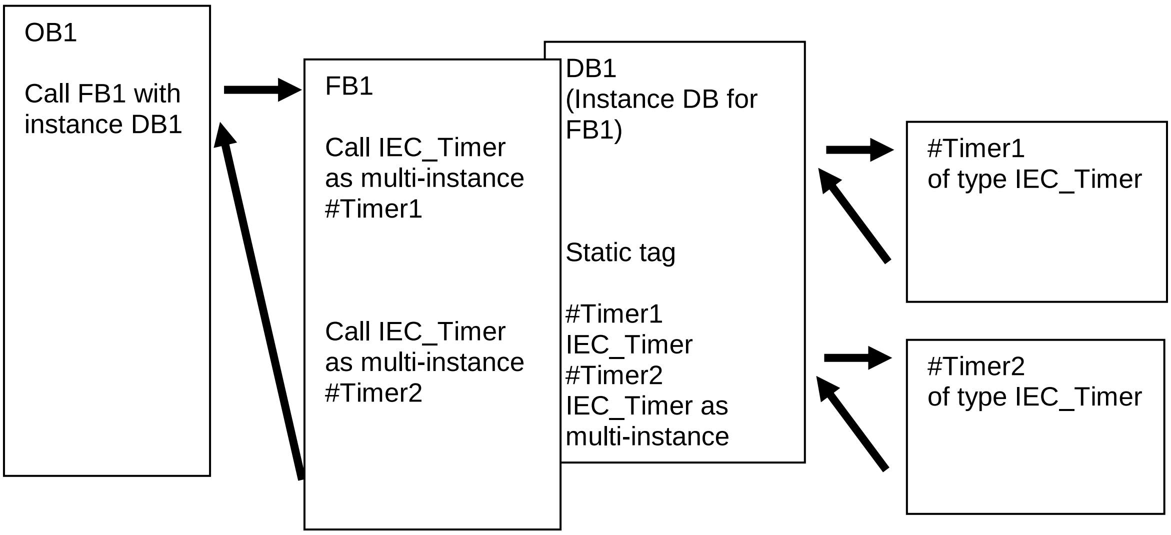
This allows you to concentrate the instance data in one instance data block and thus make better use of the number of DBs available.

Incidentally, this is always required when the calling block is to remain available for reuse as a standard block.

Example of multi-instances:

The following figure shows two calls of an IEC_Timer of type TP (pulse) within a function block.

The different data for the two counters is stored as different **multi-instances** in the instance data block DB1 of the calling function block FB1.


OB1
Call FB1 with instance DB1
FB1
Call IEC_Timer as multi-instance #Timer1
Call IEC_Timer as multi-instance #Timer2
DB1 (Instance DB for FB1)
Static tag
#Timer1 IEC_Timer
#Timer2 IEC_Timer as multi-instance
#Timer1 of type IEC_Timer
#Timer2 of type IEC_Timer

5 Task

In this chapter, an IEC timer will be added to the function block from chapter "SCE_EN_032-200 FB Programming".

6 Planning

The IEC timer is programmed as an addition to the MOTOR_AUTO [FB1] function block from the "032-200_FBProgramming.zap13" project. This project must be retrieved in order to add the IEC timer TP (latching pulse). A multi-instance will be created as a memory for the timer.

6.1 Automatic mode - Conveyor motor with time function

The Memory_automatic_start_stop is latched with Start but only if the reset conditions are not present.

The Memory_automatic_start_stop is reset if Stop is present or safety shutoff is active or automatic mode is not activated (manual mode).

The Conveyor_motor_automatic_mode output is activated when Memory_automatic_start_stop is set, the enable conditions are met and Memory_conveyor_start_stop is set.

To save energy, the conveyor should only run when a part is present.

For this reason, the Memory_conveyor_start_stop is set when Sensor_chute_occupied signals a part and reset when Sensor_end_of_conveyor produces a negative edge or safety shutoff is active or automatic mode is not activated (manual mode).

Addition of time function:

Because the Sensor_end_of_conveyor is not able to be mounted directly at the end of the conveyor, the Sensor_end_of_conveyor signal must be stretched.

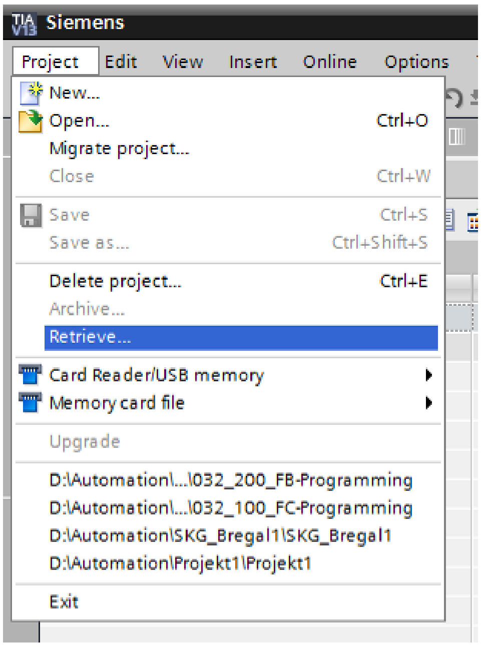
To achieve this, a latching pulse will be inserted between Sensor_end_of_conveyor and the negative edge detection.

7 Structured step-by-step instructions

You can find instructions on how to carry out planning below. If you already have a good understanding of everything, it will be sufficient to focus on the numbered steps. Otherwise, simply follow the detailed steps in the instructions.

7.1 Retrieve an existing project

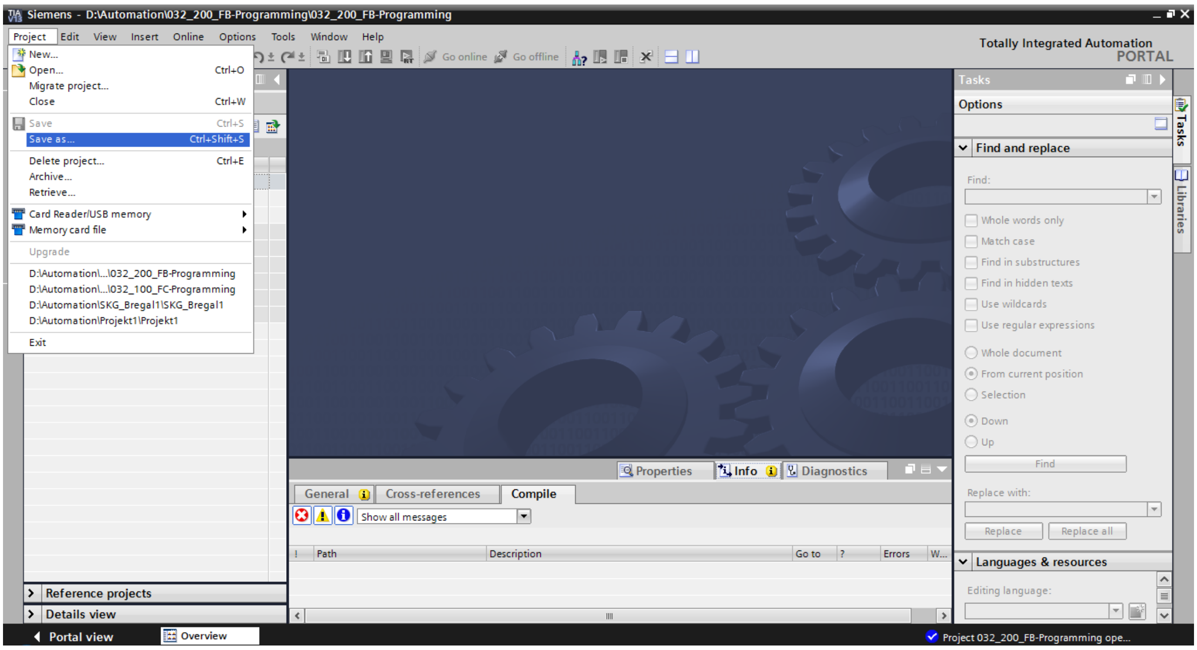
→ Before we can expand the "MOTOR_AUTO [FB1]" function block, we must retrieve the "032-200_FBProgramming.zap13" project from chapter "SCE_EN_032-200 FBProgramming". To retrieve an existing project that has been archived, you must select the relevant archive with → Project → Retrieve in the project view. Confirm your selection with Open. (→ Project → Retrieve → Select a .zap archive → Open)



→ The next step is to select the target directory where the retrieved project will be stored. Confirm your selection with "OK". (→ Target directory → OK)

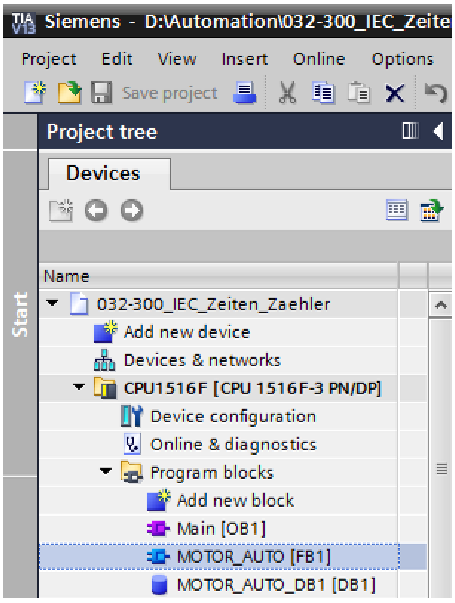
Save the opened project under the name 032-300_IEC_Timers_Counters.

→ Project → Save as ... → 032-300-IEC_Timers_Counters → Save)

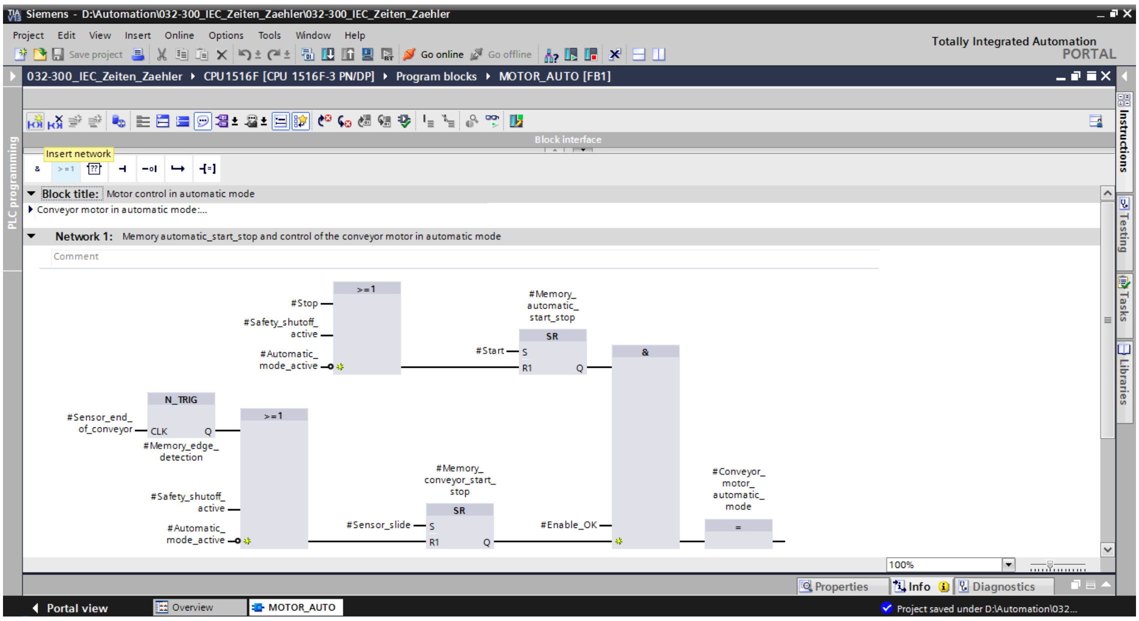


7.2 Addition of an IEC timer TP to function block FB1 "MOTOR_AUTO"

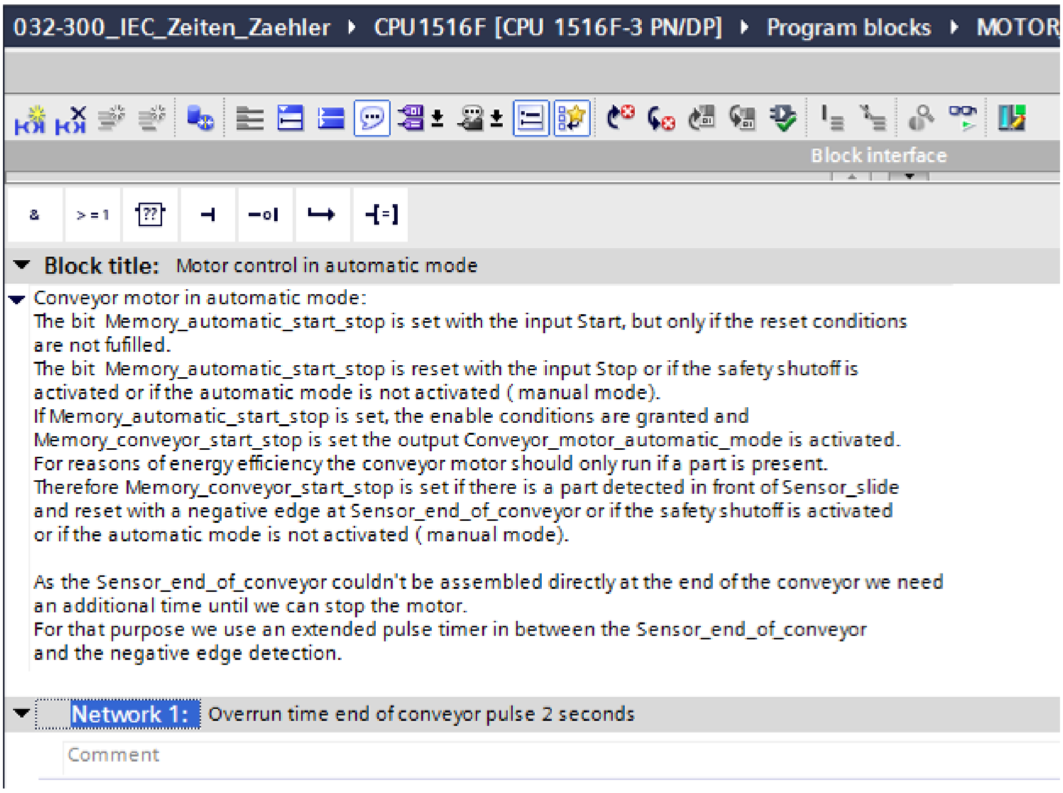
→ First, open the "MOTOR_AUTO [FB1]" function block with a double-click.



→ Insert another network at the beginning of the "MOTOR_AUTO [FB1]" function block by selecting the → "block title" and then clicking the → icon for "Insert network".


Siemens - D:\Automation\032-300_IEC_Zeiten_Zaehler\032-300_IEC_Zeiten_Zaehler
Project Edit View Insert Online Options Tools Window Help
Save project
Go online
Go offline
Totally Integrated Automation PORTAL
032-300_IEC_Zeiten_Zaehler ▸ CPU1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ MOTOR_AUTO [FB1]
PLC programming
Block interface
Insert network
Block title: Motor control in automatic mode
Conveyor motor in automatic mode:...
Network 1: Memory automatic_start_stop and control of the conveyor motor in automatic mode
Comment
>=1
#Stop
#Safety_shutoff_ active
#Automatic_ mode_active
#Memory_ automatic_ start_stop
SR
#Start
S
R1
Q
&
N_TRIG
#Sensor_end_ of_conveyor
CLK
Q
#Memory_edge_ detection
>=1
#Safety_shutoff_ active
#Automatic_ mode_active
#Memory_ conveyor_start_ stop
SR
#Sensor_slide
S
R1
Q
#Enable_OK
#Conveyor_ motor_ automatic_ mode
=
100%
Instructions
Testing
Tasks
Libraries
Properties
Info
Diagnostics
Portal view
Overview
MOTOR_AUTO
Project saved under D:\Automation\032...

→ Add helpful information to the block comment and the network title of "Network 1:".


032-300_IEC_Zeiten_Zaehler ▸ CPU1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ MOTOR
Block interface
& >=1 [??] ⊣ –o| ↦ -[=]

Block title: Motor control in automatic mode

Conveyor motor in automatic mode:
The bit Memory_automatic_start_stop is set with the input Start, but only if the reset conditions are not fufilled.
The bit Memory_automatic_start_stop is reset with the input Stop or if the safety shutoff is activated or if the automatic mode is not activated (manual mode).
If Memory_automatic_start_stop is set, the enable conditions are granted and Memory_conveyor_start_stop is set the output Conveyor_motor_automatic_mode is activated.
For reasons of energy efficiency the conveyor motor should only run if a part is present.
Therefore Memory_conveyor_start_stop is set if there is a part detected in front of Sensor_slide and reset with a negative edge at Sensor_end_of_conveyor or if the safety shutoff is activated or if the automatic mode is not activated (manual mode).

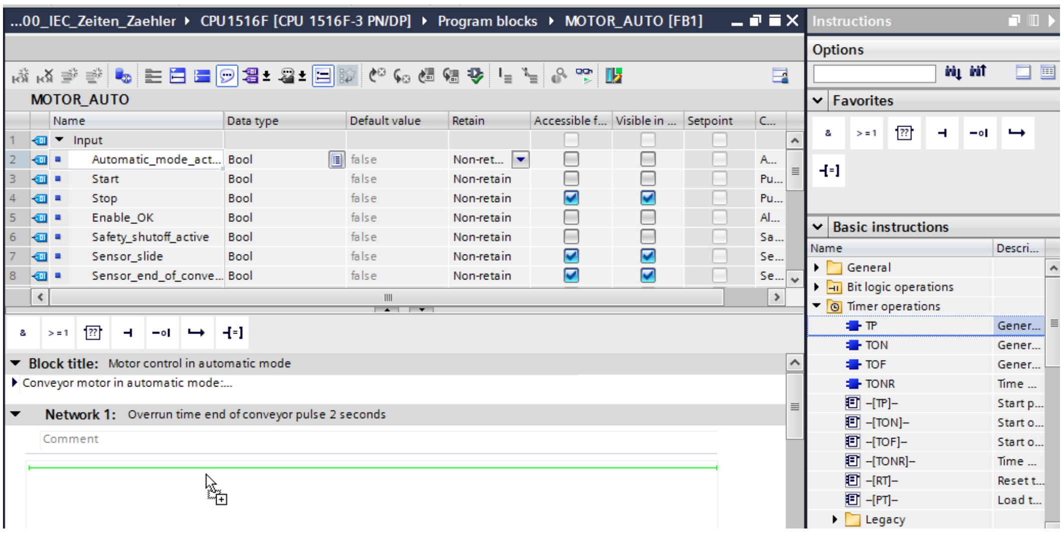
As the Sensor_end_of_conveyor couldn't be assembled directly at the end of the conveyor we need an additional time until we can stop the motor.
For that purpose we use an extended pulse timer in between the Sensor_end_of_conveyor and the negative edge detection.

Network 1: Overrun time end of conveyor pulse 2 seconds

Comment

→ On the right side of your programming window, you will see the timer functions in the list of instructions. Under → Basic instructions → Timer operations, find function TP (Generate pulse) and use a drag-and-drop operation to move it to Network 1 (green line appears, mouse pointer with + symbol).

(→ Instructions → Basic instructions → Timer operations → TP)


...00_IEC_Zeiten_Zaehler ▸ CPU1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ MOTOR_AUTO [FB1]
Instructions
Options
Favorites
& >=1 [??] ⊣ –o| ↦ -[=]
Basic instructions

MOTOR_AUTO

	Name	Data type	Default value	Retain	Accessible f...	Visible in ...	Setpoint	C...
1	Input				□	□	□	
2	Automatic_mode_act...	Bool	false	Non-ret...	□	□	□	A...
3	Start	Bool	false	Non-retain	□	□	□	Pu...
4	Stop	Bool	false	Non-retain	☑	☑	□	Pu...
5	Enable_OK	Bool	false	Non-retain	□	□	□	Al...
6	Safety_shutoff_active	Bool	false	Non-retain	□	□	□	Sa...
7	Sensor_slide	Bool	false	Non-retain	☑	☑	□	Se...
8	Sensor_end_of_conve...	Bool	false	Non-retain	☑	☑	□	Se...

Block title: Motor control in automatic mode

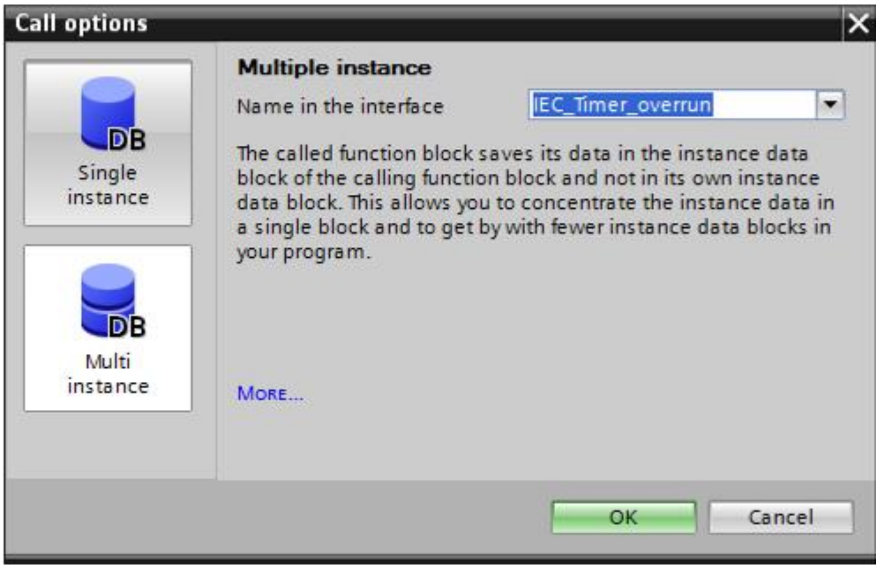
Conveyor motor in automatic mode:...

Network 1: Overrun time end of conveyor pulse 2 seconds

Comment

Name	Descri...
General	
Bit logic operations	
Timer operations	
TP	Gener...
TON	Gener...
TOF	Gener...
TONR	Time ...
--[TP]--	Start p...
--[TON]--	Start o...
--[TOF]--	Start o...
--[TONR]--	Time ...
--[RT]--	Reset t...
--[PT]--	Load t...
Legacy	

→ The timer function requires a memory. Here, this memory is made available within the instance data block by the function block without the creation of a new instance data block. Select the →"Multi-instance" option for this. Enter a name for the multi-instance and confirm with → "OK". (→ Multi-instance → IEC_Timer_overrun → OK)



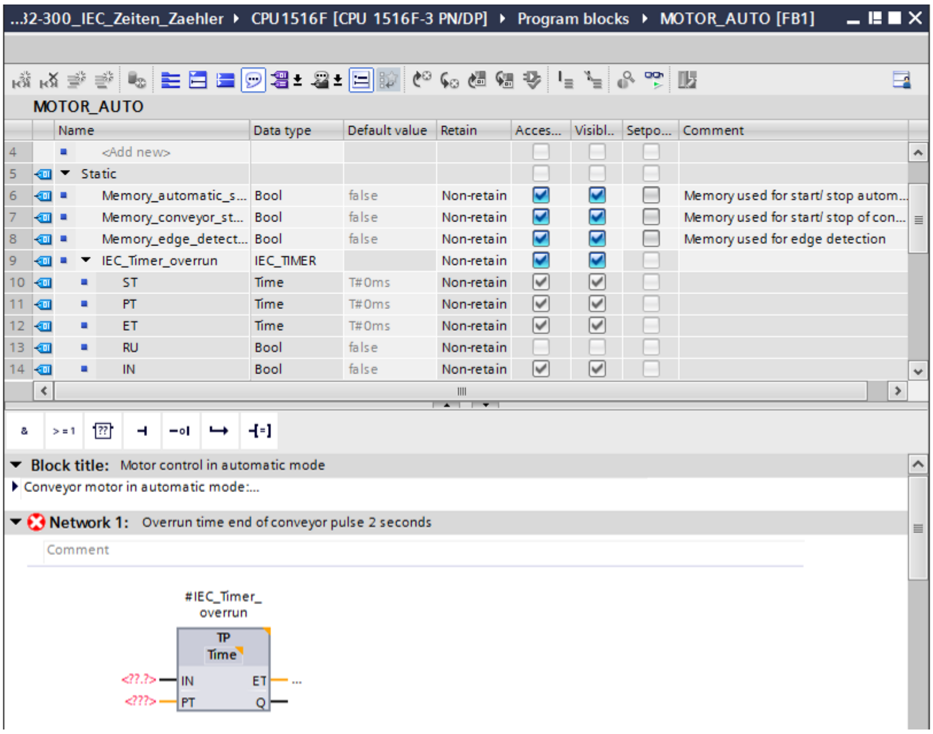
Call options

Multiple instance

Name in the interface: IEC_Timer_overrun

The called function block saves its data in the instance data block of the calling function block and not in its own instance data block. This allows you to concentrate the instance data in a single block and to get by with fewer instance data blocks in your program.

→ As a result, a tag structure of "Static" type suitable for TP Timer will be created in the interface description.



...32-300_IEC_Zeiten_Zaehler ▸ CPU1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ MOTOR_AUTO [FB1]

MOTOR_AUTO

	Name	Data type	Default value	Retain	Acces...	Visibl..	Setpo...	Comment
4	<Add new>				□	□	□	
5	▼ Static				□	□	□	
6	Memory_automatic_s...	Bool	false	Non-retain	☑	☑	□	Memory used for start/ stop autom...
7	Memory_conveyor_st...	Bool	false	Non-retain	☑	☑	□	Memory used for start/ stop of con...
8	Memory_edge_detect...	Bool	false	Non-retain	☑	☑	□	Memory used for edge detection
9	▼ IEC_Timer_overrun	IEC_TIMER		Non-retain	☑	☑	□	
10	ST	Time	T#0ms	Non-retain	☑	☑	□	
11	PT	Time	T#0ms	Non-retain	☑	☑	□	
12	ET	Time	T#0ms	Non-retain	☑	☑	□	
13	RU	Bool	false	Non-retain	□	□	□	
14	IN	Bool	false	Non-retain	☑	☑	□	

Block title: Motor control in automatic mode

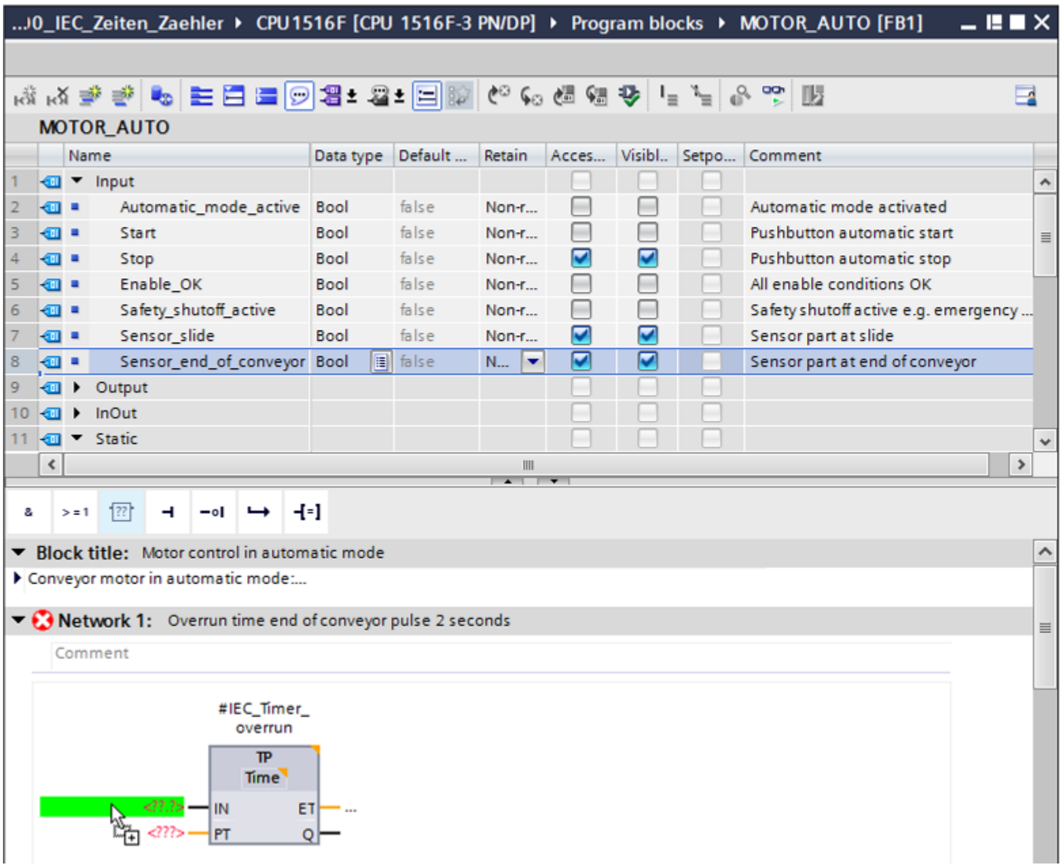
Conveyor motor in automatic mode:...

Network 1: Overrun time end of conveyor pulse 2 seconds

Comment

***Note:** A multi-instance can only be used for programming within a function block because static tags are only available there.*

→ Use drag-and-drop to move input parameter #Sensor_end_of_conveyor to <??.?> in front of parameter "IN" of TP Timer so that this will be started at a positive edge at input #Sensor_end_of_conveyor. The best way to select a parameter in the interface description is by "grabbing" it at the blue symbol. (→ Sensor_end_of_conveyor)



..J0_IEC_Zeiten_Zaehler ▸ CPU1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ MOTOR_AUTO [FB1]

MOTOR_AUTO

	Name	Data type	Default ...	Retain	Acces...	Visibl..	Setpo...	Comment
1	▼ Input				□	□	□	
2	Automatic_mode_active	Bool	false	Non-r...	□	□	□	Automatic mode activated
3	Start	Bool	false	Non-r...	□	□	□	Pushbutton automatic start
4	Stop	Bool	false	Non-r...	☑	☑	□	Pushbutton automatic stop
5	Enable_OK	Bool	false	Non-r...	□	□	□	All enable conditions OK
6	Safety_shutoff_active	Bool	false	Non-r...	□	□	□	Safety shutoff active e.g. emergency ...
7	Sensor_slide	Bool	false	Non-r...	☑	☑	□	Sensor part at slide
8	Sensor_end_of_conveyor	Bool	false	N...	☑	☑	□	Sensor part at end of conveyor
9	▸ Output				□	□	□	
10	▸ InOut				□	□	□	
11	▼ Static				□	□	□	

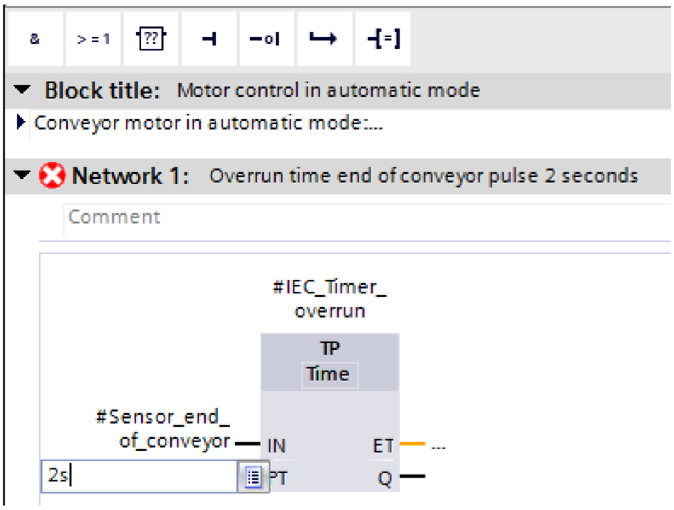
▼ Block title: Motor control in automatic mode

▸ Conveyor motor in automatic mode:...

▼ Network 1: Overrun time end of conveyor pulse 2 seconds

Comment

→ Enter the required pulse duration of 2 seconds in front of parameter "PT" (→ 2s)



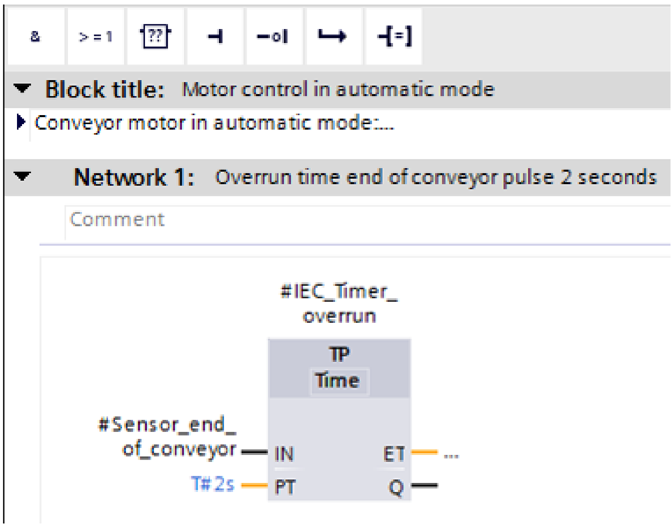
▼ Block title: Motor control in automatic mode

▸ Conveyor motor in automatic mode:...

▼ Network 1: Overrun time end of conveyor pulse 2 seconds

Comment

→ The entry of 2s is converted automatically to the IEC-Time format suitable for the IEC timer and is shown as constant "T#2s".



Block title: Motor control in automatic mode

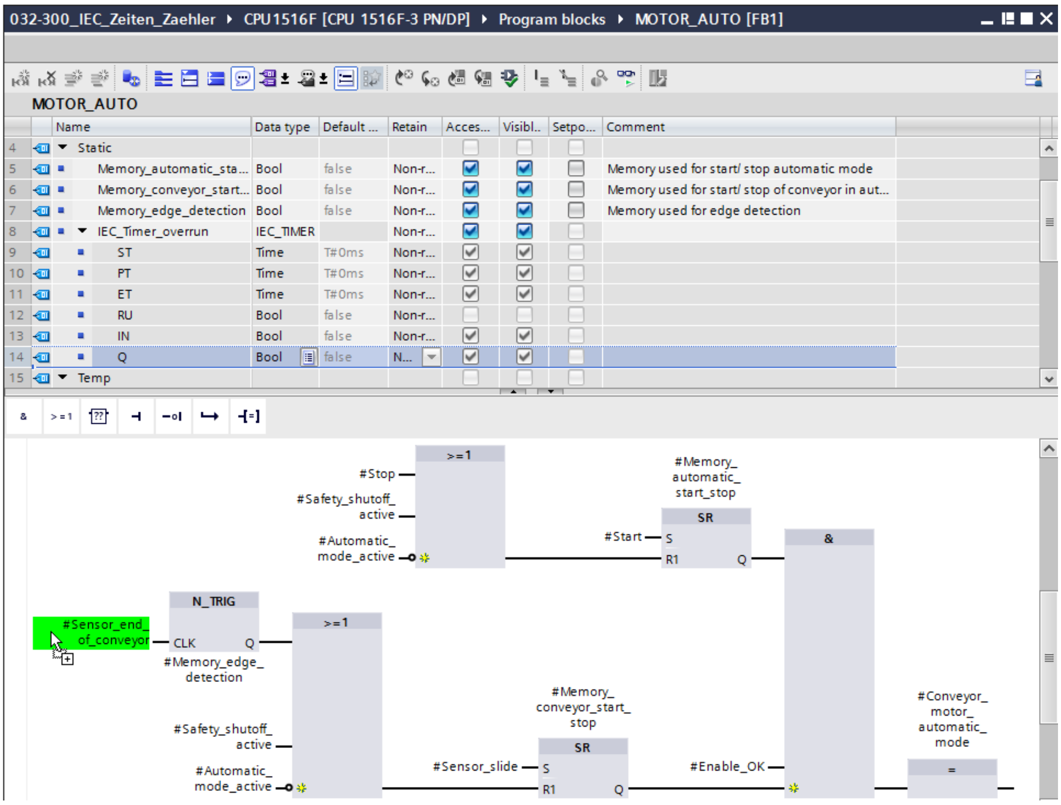
Conveyor motor in automatic mode:...

Network 1: Overrun time end of conveyor pulse 2 seconds

Comment

→ Now move output "Q" from tag structure "IEC_Timer_overrun" onto input "CLK" of negative edge "N_TRIG" in Network 2. This will replace the #Sensor_end_of_conveyor input tag previously entered there and the conveyor will be stopped by a negative edge of the IEC_Timer_overrun pulse.

(→ Network 2 → IEC_Timer_overrun→ Q → #Sensor_end_of_conveyor)

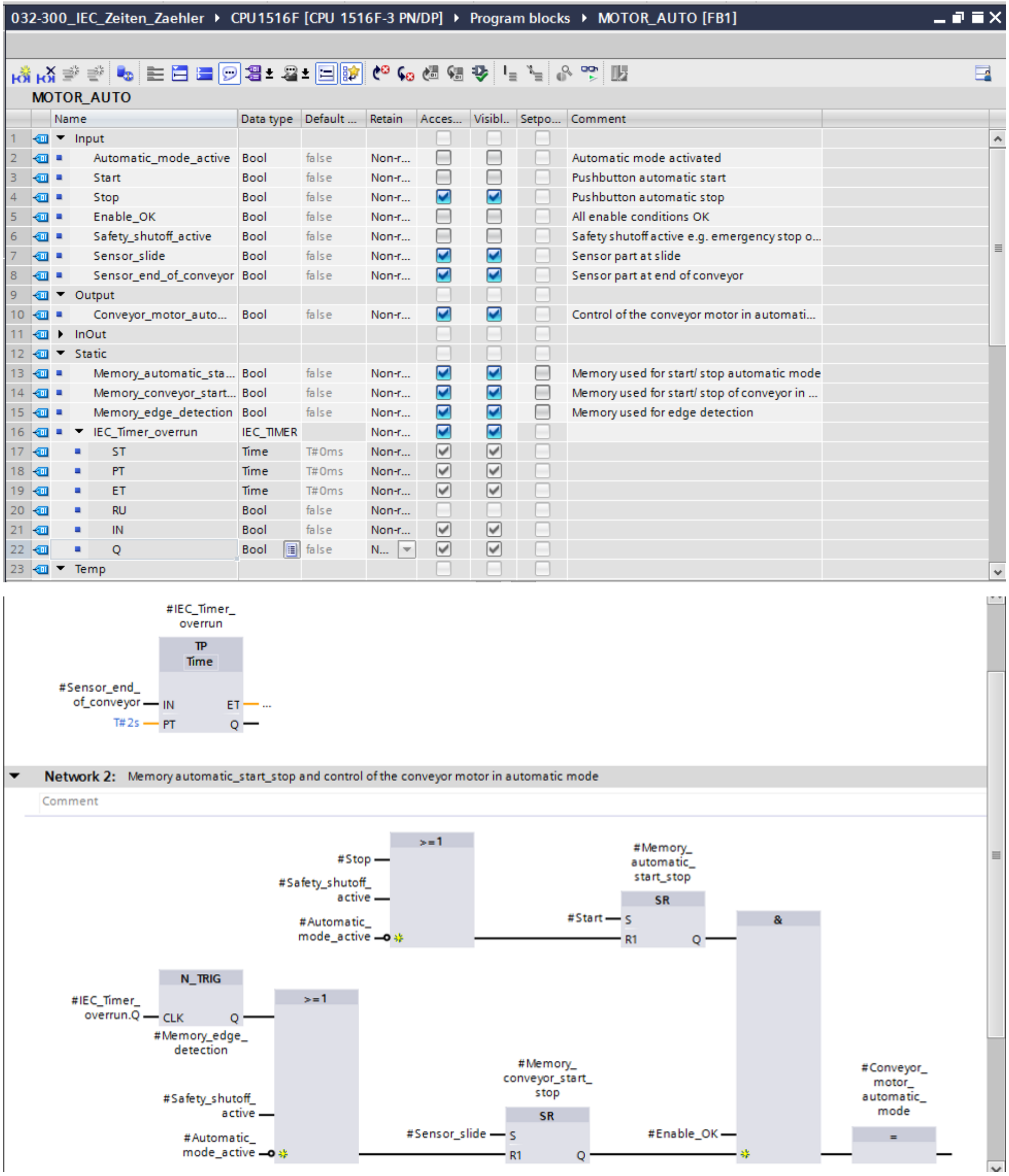


032-300_IEC_Zeiten_Zaehler ▸ CPU1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ MOTOR_AUTO [FB1]

MOTOR_AUTO

	Name	Data type	Default ...	Retain	Acces...	Visibl...	Setpo...	Comment
4	Static				□	□	□	
5	Memory_automatic_sta...	Bool	false	Non-r...	☑	☑	□	Memory used for start/ stop automatic mode
6	Memory_conveyor_start...	Bool	false	Non-r...	☑	☑	□	Memory used for start/ stop of conveyor in aut...
7	Memory_edge_detection	Bool	false	Non-r...	☑	☑	□	Memory used for edge detection
8	IEC_Timer_overrun	IEC_TIMER		Non-r...	☑	☑	□	
9	ST	Time	T#0ms	Non-r...	☑	☑	□	
10	PT	Time	T#0ms	Non-r...	☑	☑	□	
11	ET	Time	T#0ms	Non-r...	☑	☑	□	
12	RU	Bool	false	Non-r...	□	□	□	
13	IN	Bool	false	Non-r...	☑	☑	□	
14	Q	Bool	false	N...	☑	☑	□	
15	Temp				□	□	□	

→ Do not forget to click Save project. The finished function block "MOTOR_AUTO" [FB1] with the timer is shown in FBD below.



032-300_IEC_Zeiten_Zaehler ▸ CPU1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ MOTOR_AUTO [FB1]

MOTOR_AUTO

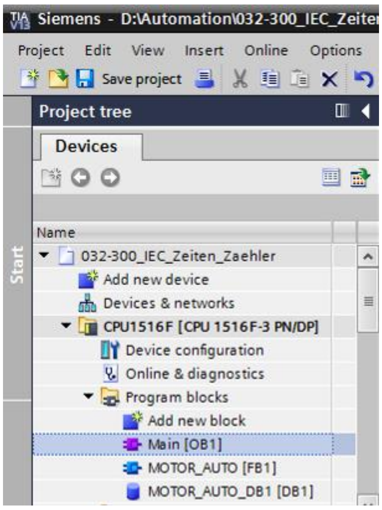
	Name	Data type	Default ...	Retain	Acces...	Visibl..	Setpo...	Comment
1	▼ Input				□	□	□	
2	Automatic_mode_active	Bool	false	Non-r...	□	□	□	Automatic mode activated
3	Start	Bool	false	Non-r...	□	□	□	Pushbutton automatic start
4	Stop	Bool	false	Non-r...	☑	☑	□	Pushbutton automatic stop
5	Enable_OK	Bool	false	Non-r...	□	□	□	All enable conditions OK
6	Safety_shutoff_active	Bool	false	Non-r...	□	□	□	Safety shutoff active e.g. emergency stop o...
7	Sensor_slide	Bool	false	Non-r...	☑	☑	□	Sensor part at slide
8	Sensor_end_of_conveyor	Bool	false	Non-r...	☑	☑	□	Sensor part at end of conveyor
9	▼ Output				□	□	□	
10	Conveyor_motor_auto...	Bool	false	Non-r...	☑	☑	□	Control of the conveyor motor in automati...
11	▸ InOut				□	□	□	
12	▼ Static				□	□	□	
13	Memory_automatic_sta...	Bool	false	Non-r...	☑	☑	□	Memory used for start/ stop automatic mode
14	Memory_conveyor_start...	Bool	false	Non-r...	☑	☑	□	Memory used for start/ stop of conveyor in ...
15	Memory_edge_detection	Bool	false	Non-r...	☑	☑	□	Memory used for edge detection
16	▼ IEC_Timer_overrun	IEC_TIMER		Non-r...	☑	☑	□	
17	ST	Time	T#0ms	Non-r...	☑	☑	□	
18	PT	Time	T#0ms	Non-r...	☑	☑	□	
19	ET	Time	T#0ms	Non-r...	☑	☑	□	
20	RU	Bool	false	Non-r...	□	□	□	
21	IN	Bool	false	Non-r...	☑	☑	□	
22	Q	Bool	false	N...	☑	☑	□	
23	▼ Temp				□	□	□	

Network 2: Memory automatic_start_stop and control of the conveyor motor in automatic mode

Comment

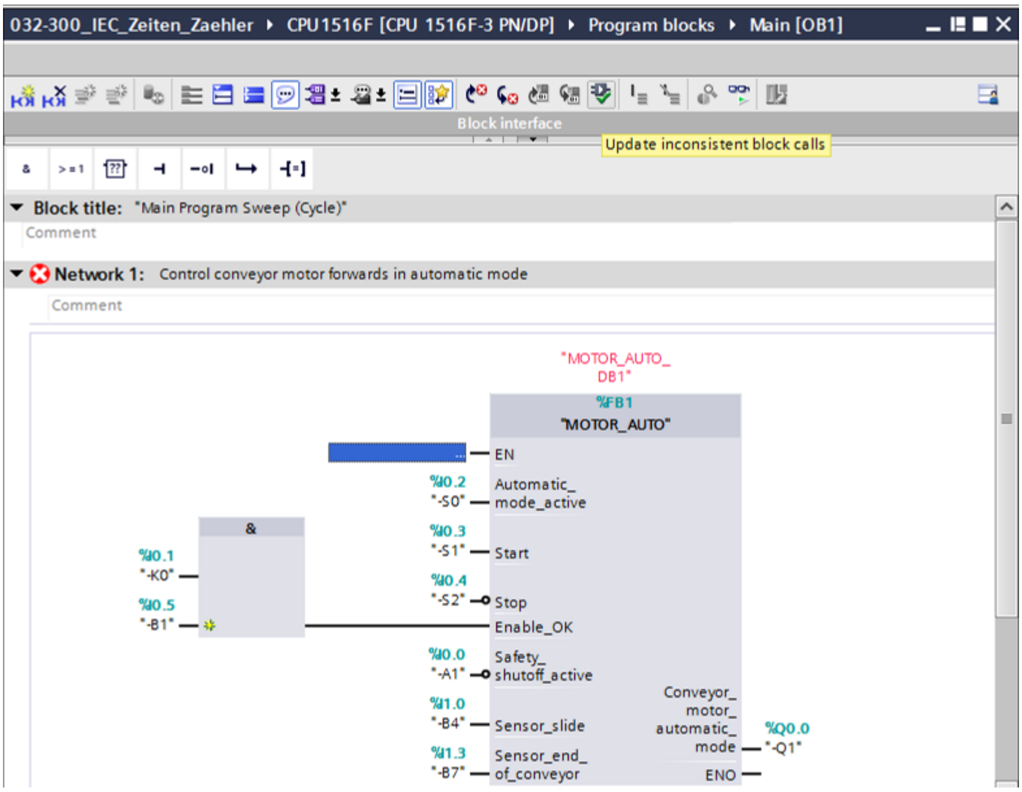
7.3 Update the block call in the organization block

→ Open the "Main [OB1]" organization block with a double-click.


Siemens - D:\Automation\032-300_IEC_Zeite
Project Edit View Insert Online Options
Save project
Project tree
Devices
Start
Name

- 032-300_IEC_Zeiten_Zaehler
  - Add new device
  - Devices & networks
  - CPU1516F [CPU 1516F-3 PN/DP]
    - Device configuration
    - Online & diagnostics
    - Program blocks
      - Add new block
      - Main [OB1]
      - MOTOR_AUTO [FB1]
      - MOTOR_AUTO_DB1 [DB1]

→ In Network 1 of the "Main [OB1]" organization block, instance data block "MOTOR_AUTO_DB1" for the "MOTOR_AUTO [FB1]" function block appears incorrect, because the additional memory for the TP Timer has not yet been added there. Click the → "" icon for "Update inconsistent block calls". This will add the "MOTOR_AUTO_DB1" instance data block correctly again. (→)


032-300_IEC_Zeiten_Zaehler ▸ CPU1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Main [OB1]

Update inconsistent block calls

Block title: "Main Program Sweep (Cycle)"

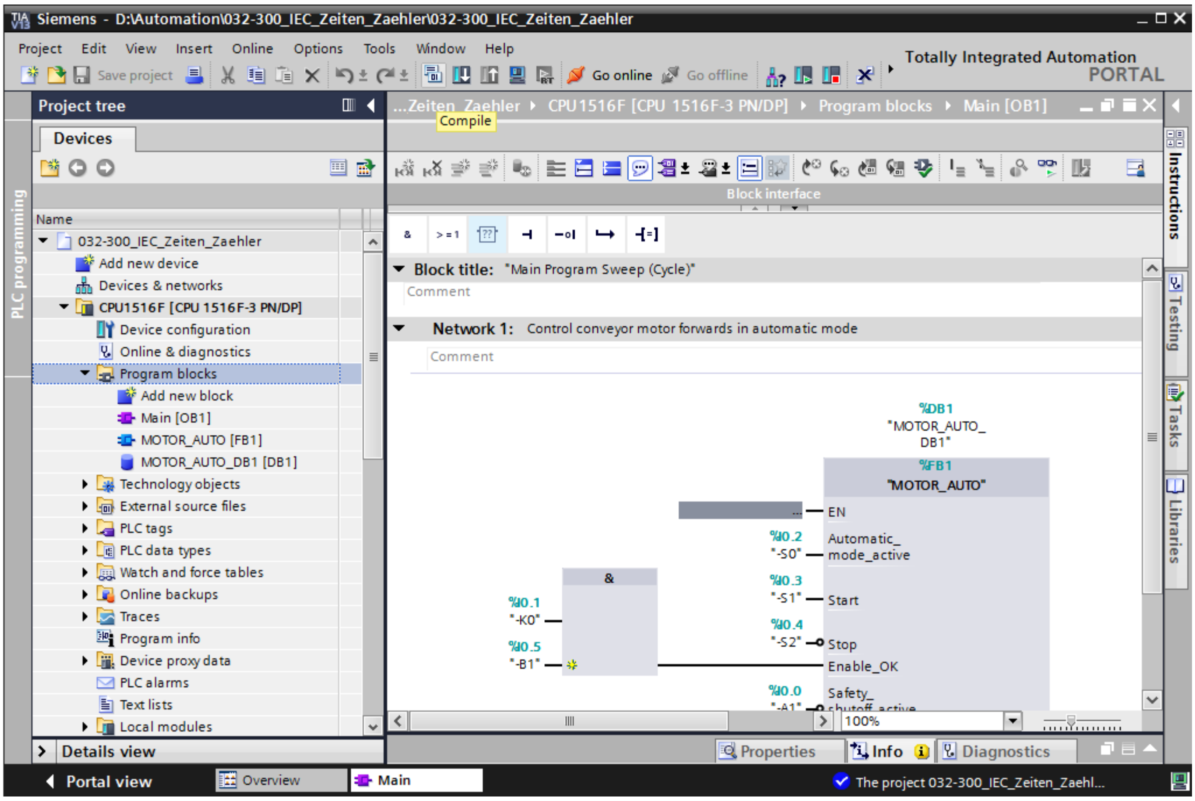
Comment

Network 1: Control conveyor motor forwards in automatic mode

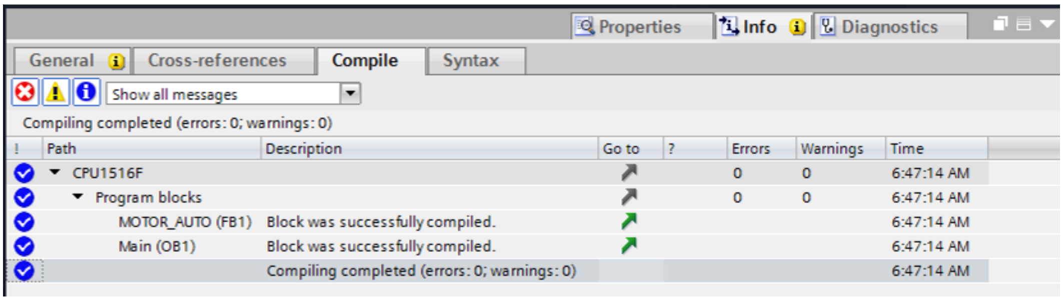
Comment

7.4 Save and compile the program

→ To save your project, select the Save project button in the menu. To compile all blocks, click the "Program blocks" folder and select the icon for compiling in the menu (→ Save project → Program blocks →).



→ The "Info", "Compile" area shows which blocks were successfully compiled.

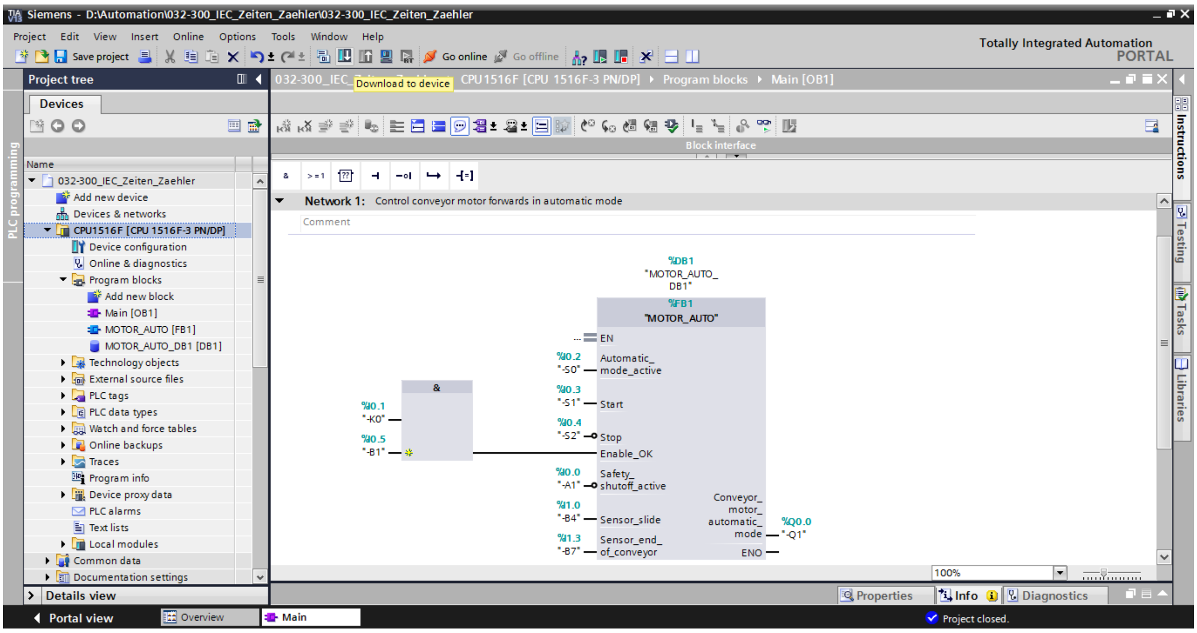


Compiling completed (errors: 0; warnings: 0)

!	Path	Description	Go to	?	Errors	Warnings	Time
✓	CPU1516F		↗		0	0	6:47:14 AM
✓	Program blocks		↗		0	0	6:47:14 AM
✓	MOTOR_AUTO (FB1)	Block was successfully compiled.	↗				6:47:14 AM
✓	Main (OB1)	Block was successfully compiled.	↗				6:47:14 AM
✓		Compiling completed (errors: 0; warnings: 0)					6:47:14 AM

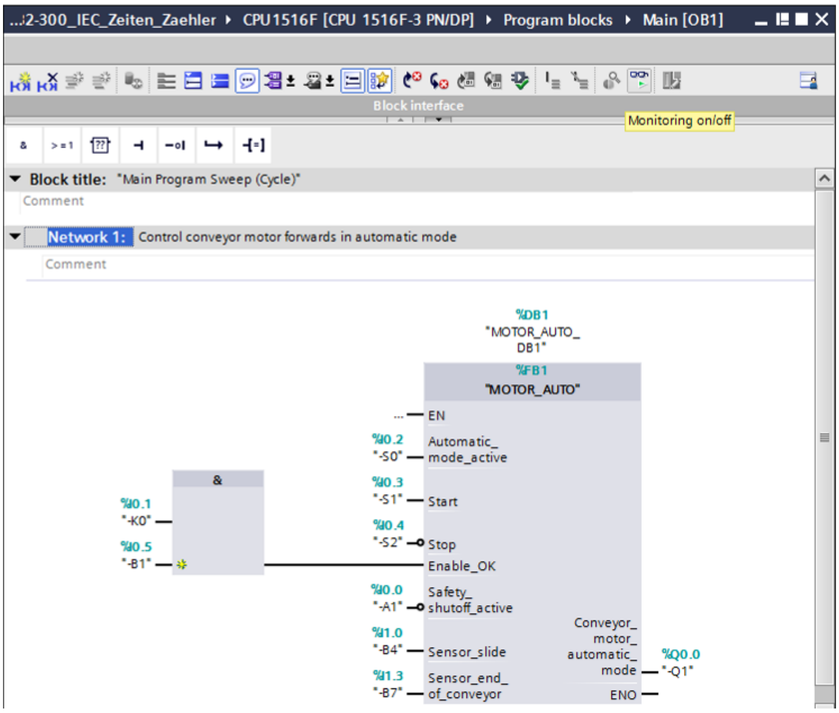
7.5 Download the program

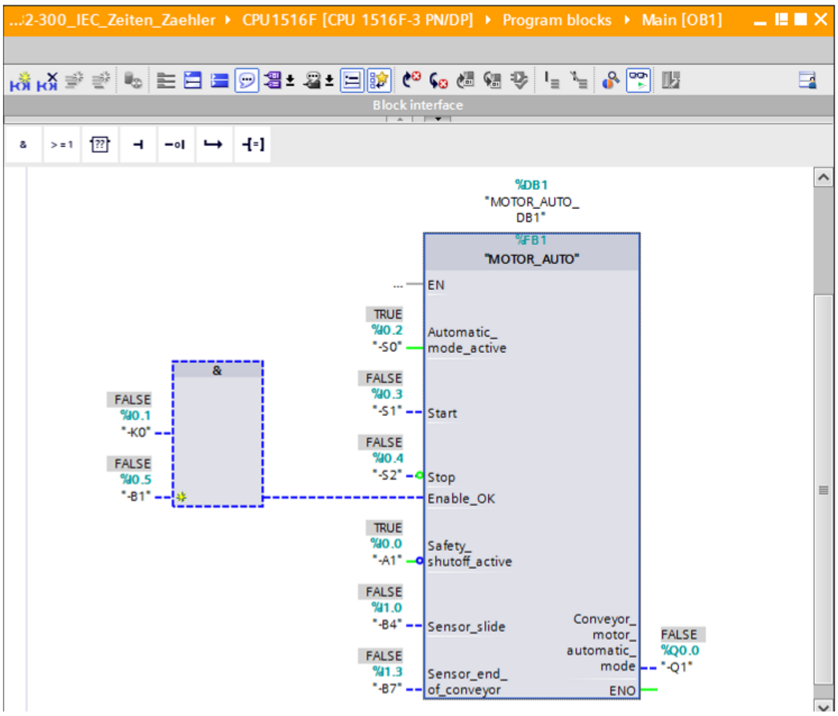
→ After successful compilation, the complete controller with the created program including the hardware configuration, as previously described in the modules, can be downloaded. (→)



7.6 Monitor program blocks

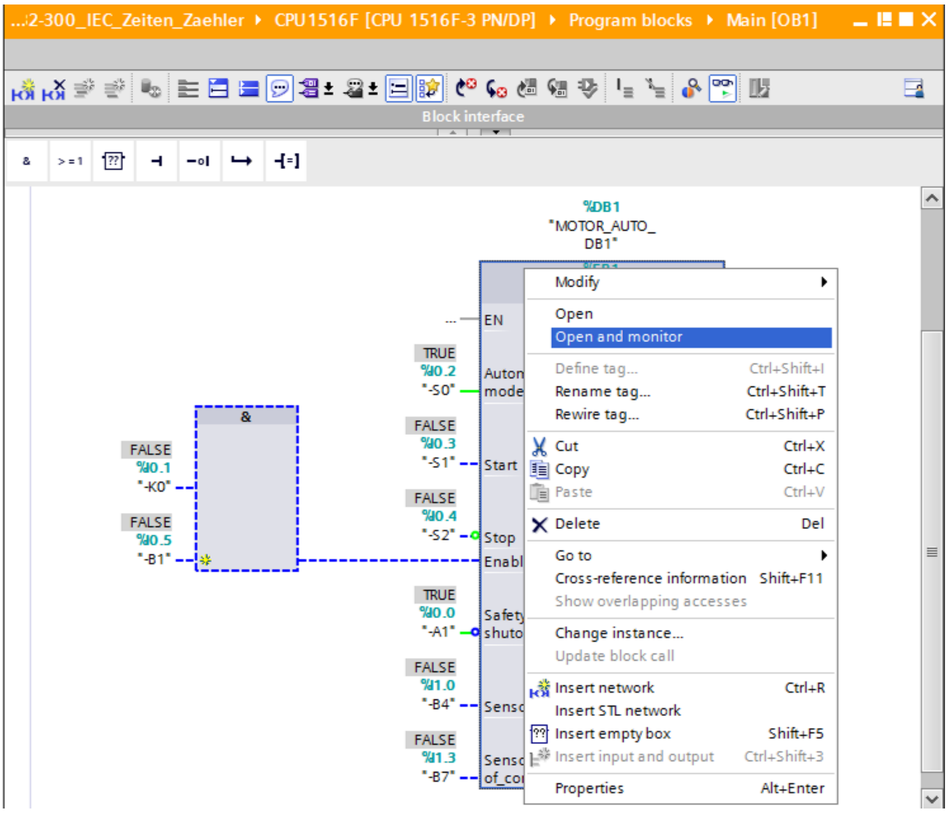
→ The desired block must be open for monitoring the downloaded program. The monitoring can now be activated/deactivated by clicking the icon. (→ Main [OB1] →)

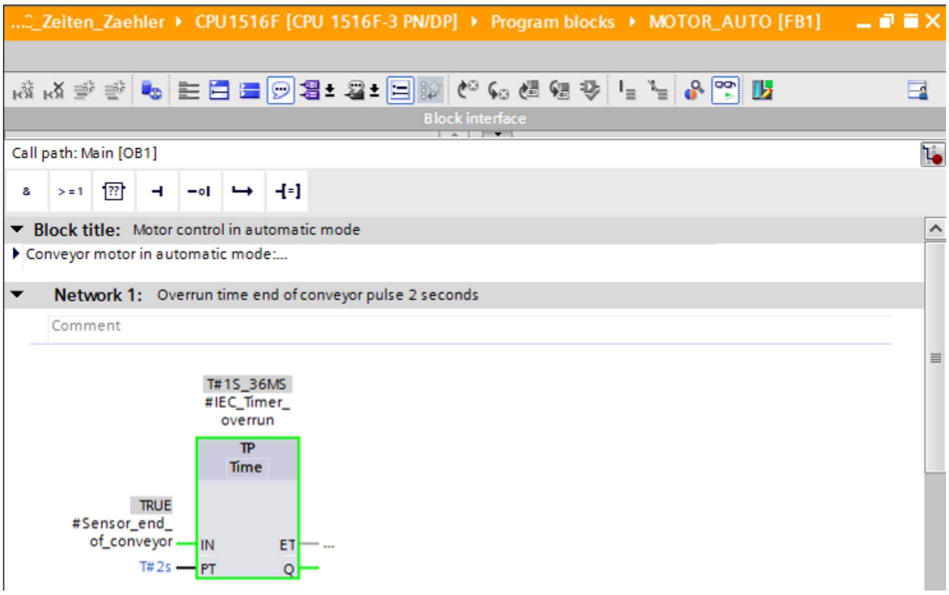

...2-300_IEC_Zeiten_Zaehler ▸ CPU1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Main [OB1]
Block interface
Monitoring on/off
Block title: "Main Program Sweep (Cycle)"
Comment
Network 1: Control conveyor motor forwards in automatic mode
Comment
%DB1
"MOTOR_AUTO_DB1"
%FB1
"MOTOR_AUTO"
EN
%I0.2 "-S0" Automatic_mode_active
%I0.3 "-S1" Start
%I0.4 "-S2" Stop
Enable_OK
%I0.0 "-A1" Safety_shutoff_active
%I1.0 "-B4" Sensor_slide
%I1.3 "-B7" Sensor_end_of_conveyor
Conveyor_motor_automatic_mode
%Q0.0 "-Q1"
ENO
&
%I0.1 "-K0"
%I0.5 "-B1"


...2-300_IEC_Zeiten_Zaehler ▸ CPU1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Main [OB1]
Block interface
%DB1
"MOTOR_AUTO_DB1"
%FB1
"MOTOR_AUTO"
EN
TRUE %I0.2 "-S0" Automatic_mode_active
FALSE %I0.3 "-S1" Start
FALSE %I0.4 "-S2" Stop
Enable_OK
TRUE %I0.0 "-A1" Safety_shutoff_active
FALSE %I1.0 "-B4" Sensor_slide
FALSE %I1.3 "-B7" Sensor_end_of_conveyor
Conveyor_motor_automatic_mode
FALSE %Q0.0 "-Q1"
ENO
&
FALSE %I0.1 "-K0"
FALSE %I0.5 "-B1"

***Note:** The monitoring here is signal-related and controller-dependent. The signal states at the terminals are indicated with TRUE or FALSE.*

→ The "MOTOR_AUTO" [FB1] function block called in the "Main [OB1]" organization block can be selected directly for "Open and monitor" after right-clicking, thereby allowing the program code in the function block with the TP Timer to be monitored.
(→ "MOTOR_AUTO" [FB1] → Open and monitor)

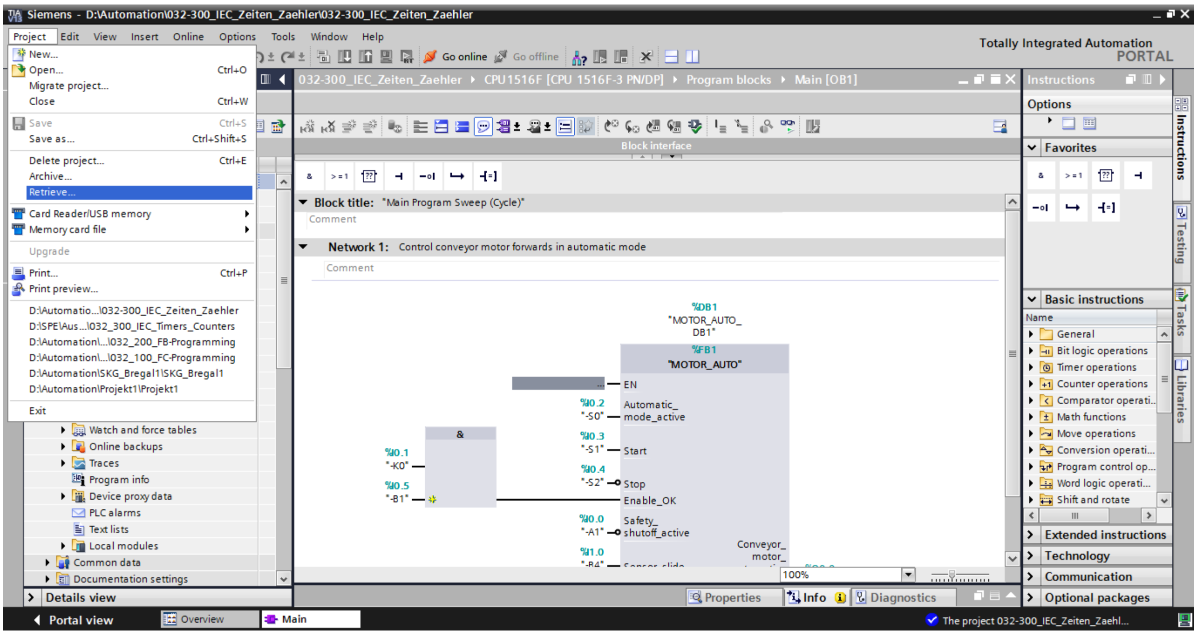

...:2-300_IEC_Zeiten_Zaehler ▸ CPU1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Main [OB1]
Block interface
%DB1
"MOTOR_AUTO_DB1"
EN
TRUE %I0.2 "-S0" Autom mode
FALSE %I0.3 "-S1" Start
FALSE %I0.4 "-S2" Stop
Enabl
TRUE %I0.0 "-A1" Safety shuto
FALSE %I1.0 "-B4" Senso
FALSE %I1.3 "-B7" Senso of_co
FALSE %I0.1 "-K0"
FALSE %I0.5 "-B1"
&
Modify
Open
Open and monitor
Define tag... Ctrl+Shift+I
Rename tag... Ctrl+Shift+T
Rewire tag... Ctrl+Shift+P
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Go to
Cross-reference information Shift+F11
Show overlapping accesses
Change instance...
Update block call
Insert network Ctrl+R
Insert STL network
Insert empty box Shift+F5
Insert input and output Ctrl+Shift+3
Properties Alt+Enter


...:_Zeiten_Zaehler ▸ CPU1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ MOTOR_AUTO [FB1]
Block interface
Call path: Main [OB1]
Block title: Motor control in automatic mode
Conveyor motor in automatic mode:...
Network 1: Overrun time end of conveyor pulse 2 seconds
Comment
T#1S_36MS
#IEC_Timer_overrun
TP
Time
TRUE
#Sensor_end_of_conveyor
IN
ET
T#2s
PT
Q

Note: *The monitoring here is function-related and controller-independent. The actuation of sensors and the station status are shown here with TRUE or FALSE.*

7.7 Archive the project

→ As the final step, we want to archive the complete project. Select the → "Archive ..." command in the → "Project" menu. Select a folder where you want to archive your project and save it with the file type "TIA Portal project archive". (→ Project → Archive → TIA Portal project archive → 032-300_IEC_Timers_Counters…. → Save)



8 Checklist

No.	Description	Completed
1	Compiling successful and without error message	
2	Download successful and without error message	
3	Switch on station (-K0 = 1) Cylinder retracted / Feedback activated (-B1 = 1) EMERGENCY OFF (-A1 = 1) not activated AUTOMATIC mode (-S0 = 1) Pushbutton automatic stop not actuated (-S2 = 1) Briefly press the automatic start pushbutton (-S1 = 1) Sensor at chute activated (-B4 = 1) then conveyor motor forwards fixed speed (-Q1 = 1) switches on and stays on.	
4	Sensor at end of conveyor activated (-B7 = 1) → -Q1 = 0 (after 2 seconds)	
5	Briefly press the automatic stop pushbutton (-S2 = 0) → -Q1 = 0	
6	Activate EMERGENCY OFF (-A1 = 0) → -Q1 = 0	
7	Manual mode (-S0 = 0) → -Q1 = 0	
8	Switch off station (-K0 = 0) → -Q1 = 0	
9	Cylinder not retracted (-B1 = 0) → -Q1 = 0	
10	Project successfully archived	

9 Exercise

9.1 Task – Exercise

In this exercise, an IEC counter is also to be added to the MOTOR_AUTO [FB1] function block. The expanded function block will be planned, programmed and tested:

The magazine for plastic holds only 5 parts and the parts will therefore be counted at the end of the conveyor.

When 5 parts are stored in the magazine, automatic mode is to be stopped.

Once the magazine has been emptied, automatic mode will be restarted with

Start_command is started again and the counter is reset.

9.2 Planning

Plan the implementation of the task on your own.

Note: *Learn about the use of IEC counters in SIMATIC S7-1500 in the online help.*

9.3 Checklist – Exercise

No.	Description	Completed
1	Compiling successful and without error message	
2	Download successful and without error message	
3	Switch on station (-K0 = 1) Cylinder retracted / Feedback activated (-B1 = 1) EMERGENCY OFF (-A1 = 1) not activated AUTOMATIC mode (-S0 = 1) Pushbutton automatic stop not actuated (-S2 = 1) Briefly press the automatic start pushbutton (S1 = 1) Sensor at chute activated (-B4 = 1) then conveyor motor forwards fixed speed (-Q1 = 1) switches on and stays on.	
4	Sensor at end of conveyor activated (-B7 = 1) → -Q1 = 0 (after 2 seconds)	
5	Briefly press the automatic stop pushbutton (-S2 = 0) → -Q1 = 0	
6	Activate EMERGENCY OFF (-A1 = 0) → -Q1 = 0	
7	Manual mode (-S0 = 0) → -Q1 = 0	
8	Switch off station (-K0 = 0) → -Q1 = 0	
9	Cylinder not retracted (-B1 = 0) → -Q1 = 0	
10	5th part in magazine → -Q1 = 0	
11	Project successfully archived	

10 Additional information

You can find additional information as an orientation aid for initial and advanced training, for example: Getting Started, videos, tutorials, apps, manuals, programming guidelines and trial software/firmware, at the following link:

www.siemens.com/sce/s7-1500