SIEMENS

Learn-/Training Document

Siemens Automation Cooperates with Education (SCE) | From Version V14 SP1

TIA Portal Module 031-300
IEC Timers and IEC Counters
Multi-instances for SIMATIC S7-1200

siemens.com/sce

SIEMENS
Global Industry Partner of WorldSkills International

worldskills

Matching SCE Trainer Packages for these Learn-/Training Document

- **SIMATIC S7-1200 AC/DC/RELAY (set of 6) "TIA Portal"**
  Order no.: 6ES7214-1BE30-4AB3
- **SIMATIC S7-1200 DC/DC/DC (set of 6) "TIA Portal"**
  Order no.: 6ES7214-1AE30-4AB3
- **Upgrade SIMATIC STEP 7 BASIC V14 SP1 (for S7-1200) (set of 6) "TIA Portal"**
  Order no.: 6ES7822-0AA04-4YE5

Note that these trainer packages are replaced with successor packages when necessary.
An overview of the currently available SCE packages is available 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 Learn-/Training Document 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 facilities 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, which 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 Learn-/Training Document 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 person: 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 explicitly not permitted. We do not consent to commercial use of the Learn-/Training Document.

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

Table of contents

IEC Timers and IEC Counters Multi-instances for SIMATIC S7-1200

1 Goal

In this chapter, you will become acquainted with the use of single instances and multi-instances for programming of SIMATIC S7-1200 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 for SIMATIC S7 CPU1214C. For this chapter, you can use the following project, for example:

031-200_FB-Programming_S7-1200….zap14

3 Required hardware and software

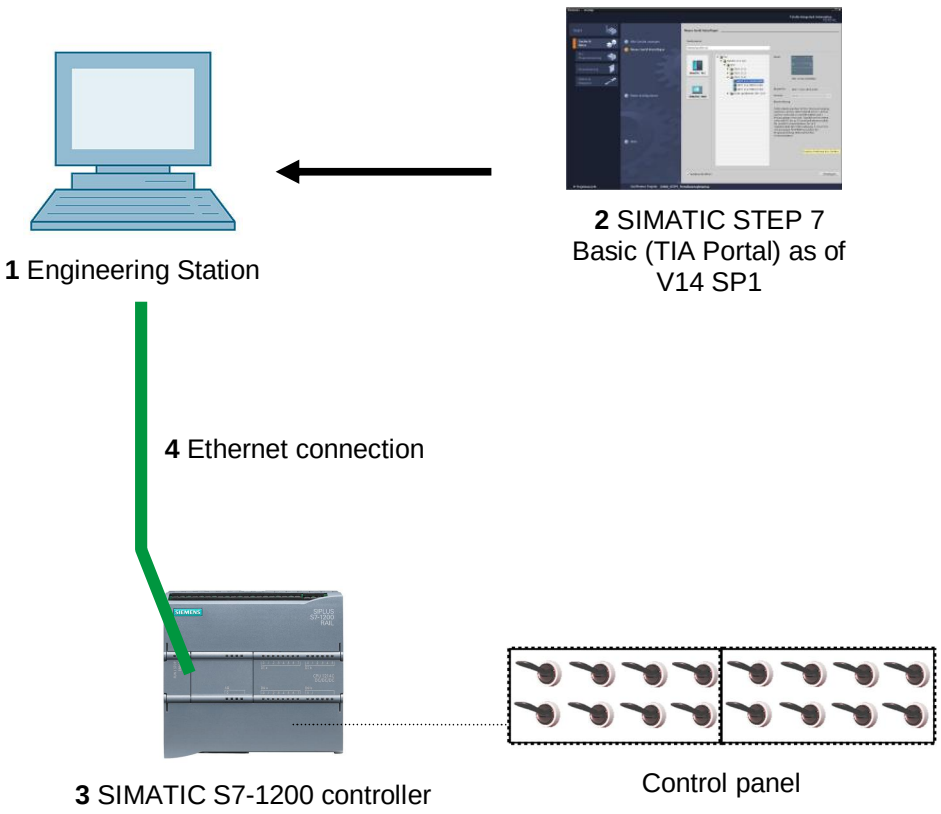
1 Engineering station: requirements include hardware and operating system (for additional information, see Readme on the TIA Portal Installation DVDs)

2 SIMATIC STEP 7 Basic software in TIA Portal – as of V14 SP1

3 SIMATIC S7-1200 controller, e.g. CPU 1214C DC/DC/DC with ANALOG OUTPUT SB1232 signal board, 1 AO – Firmware as of V4.2.1

Note: The digital inputs should be fed out to a control panel.

4 Ethernet connection between engineering station and controller



4 Theory

4.1 Instances and multi-instances in SIMATIC S7-1200

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**.

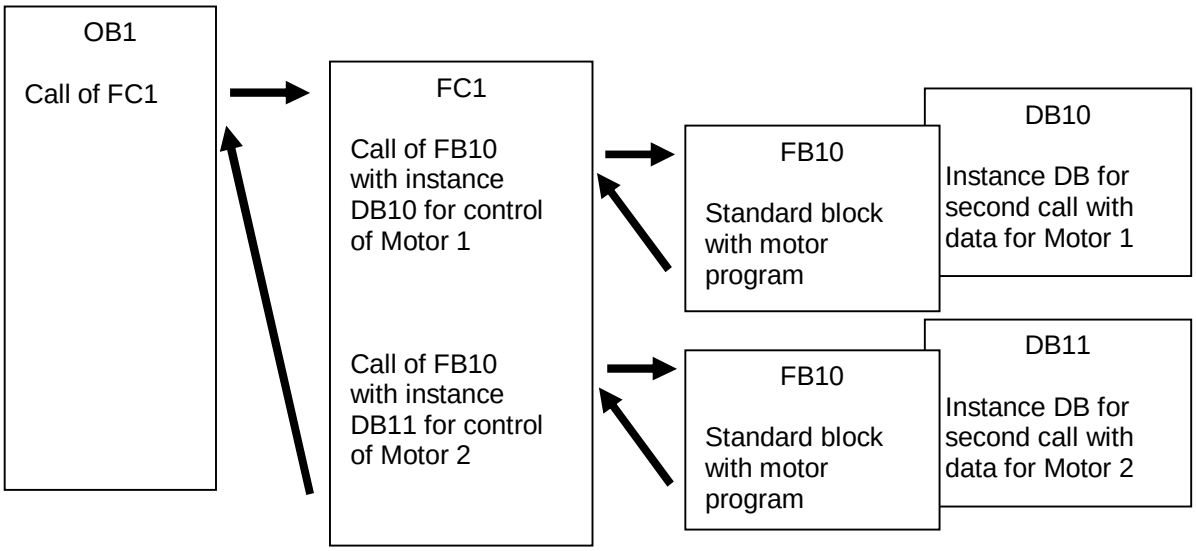
If the function block was created according to the rules for library-compatible standard blocks, it can also be called multiple times.

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.



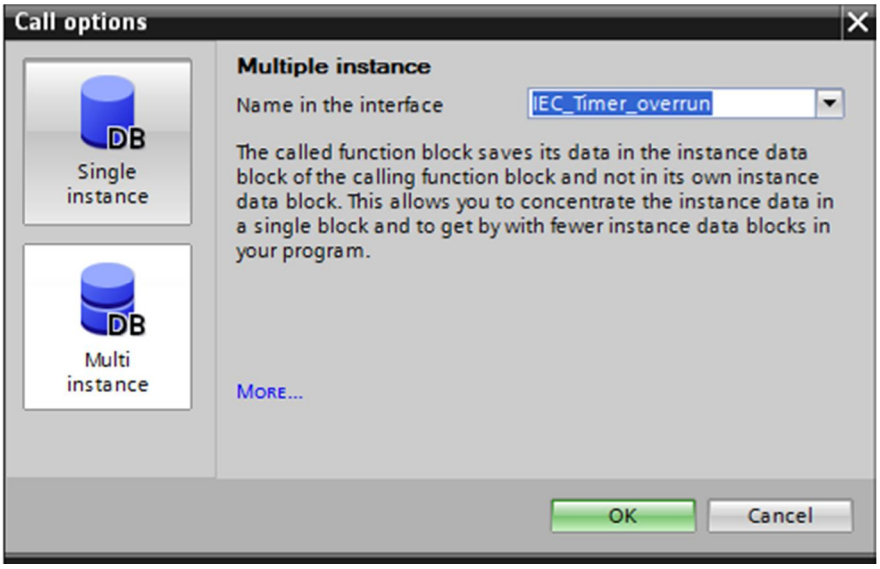
Note: *Some commands, such as timers and counters, behave like function blocks. When these are called, they also require an assigned memory area, e.g., in the form of an instance data block.*

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 are to 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:



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.

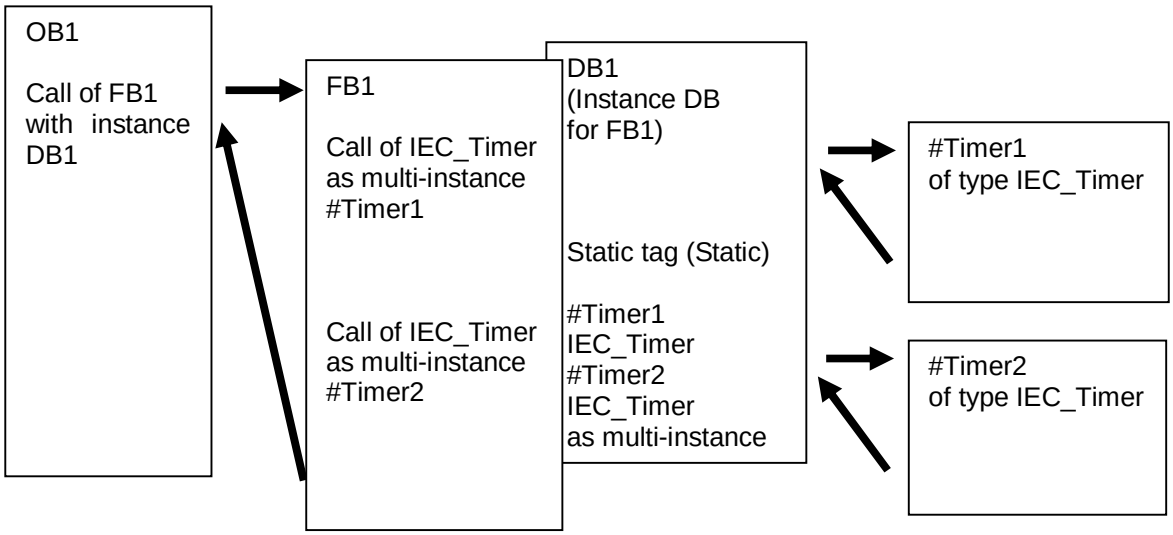
This allows you to concentrate the instance data in one instance data block and thus make better use of the number of DBs available.

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.



5 Task

In this chapter, an IEC timer will be added to the function block from chapter "SCE_EN_031-200 FB Programming S7-1200".

6 Planning

The IEC timer is programmed as an addition to the MOTOR_AUTO [FB1] function block from the "031-200_FB-Programming_S7-1200.zap13" project. This project must be retrieved in order to now 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.

6.2 Technology diagram

Here, you see the technology diagram for the task.

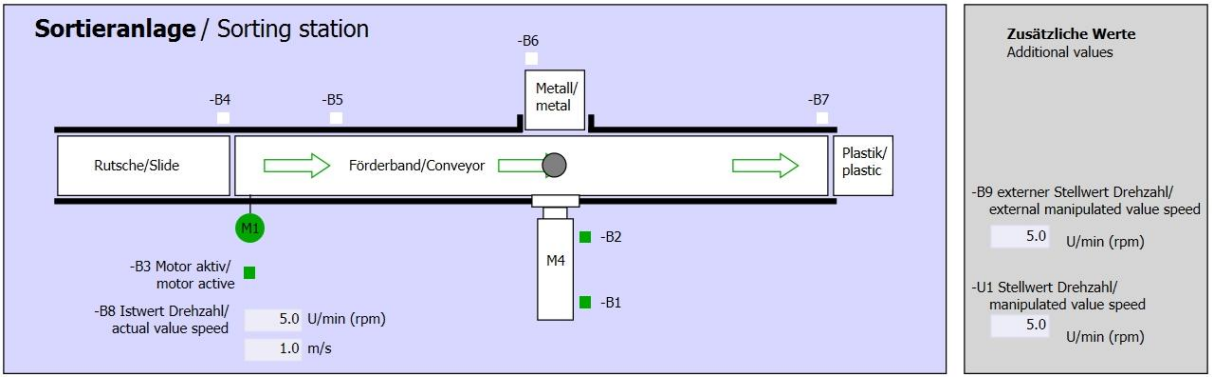


Figure 1: Technology diagram

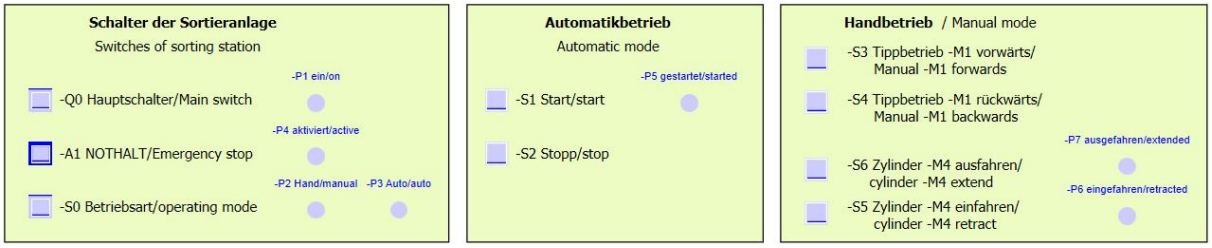


Figure 2: Control panel

6.3 Reference list

The following signals are needed as global operands for this task.

DI	Type	Identifier	Function	NC/NO
I 0.0	BOOL	-A1	Return signal emergency stop ok	NC
I 0.1	BOOL	-K0	Main switch "ON"	NO
I 0.2	BOOL	-S0	Mode selector manual (0)/ automatic (1)	Manual = 0 Auto = 1
I 0.3	BOOL	-S1	Pushbutton automatic start	NO
I 0.4	BOOL	-S2	Pushbutton automatic stop	NC
I 0.5	BOOL	-B1	Sensor cylinder M4 retracted	NO
I 1.0	BOOL	-B4	Sensor at chute occupied	NO
I 1.3	BOOL	-B7	Sensor part at end of conveyor	NO

DO	Type	Identifier	Function	
Q 0.0	BOOL	-Q1	Conveyor motor M1 forwards fixed speed	

Legend for reference list

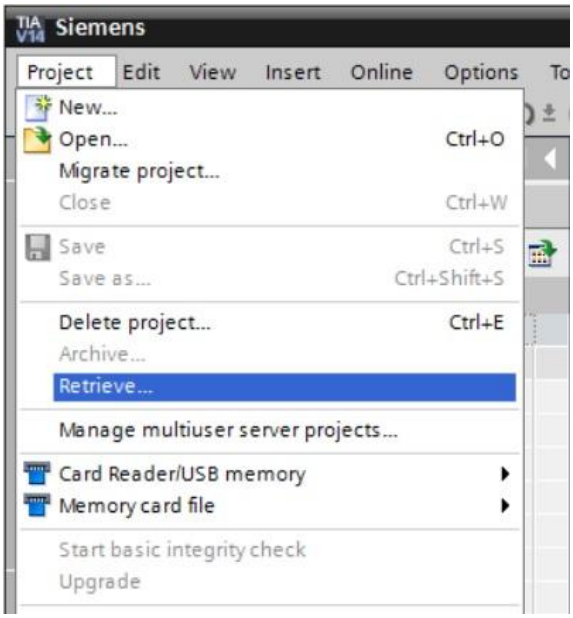
DI	Digital Input	DO	Digital Output
AI	Analog Input	AO	Analog Output
I	Input	Q	Output
NC	Normally Closed		
NO	Normally Open		

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.

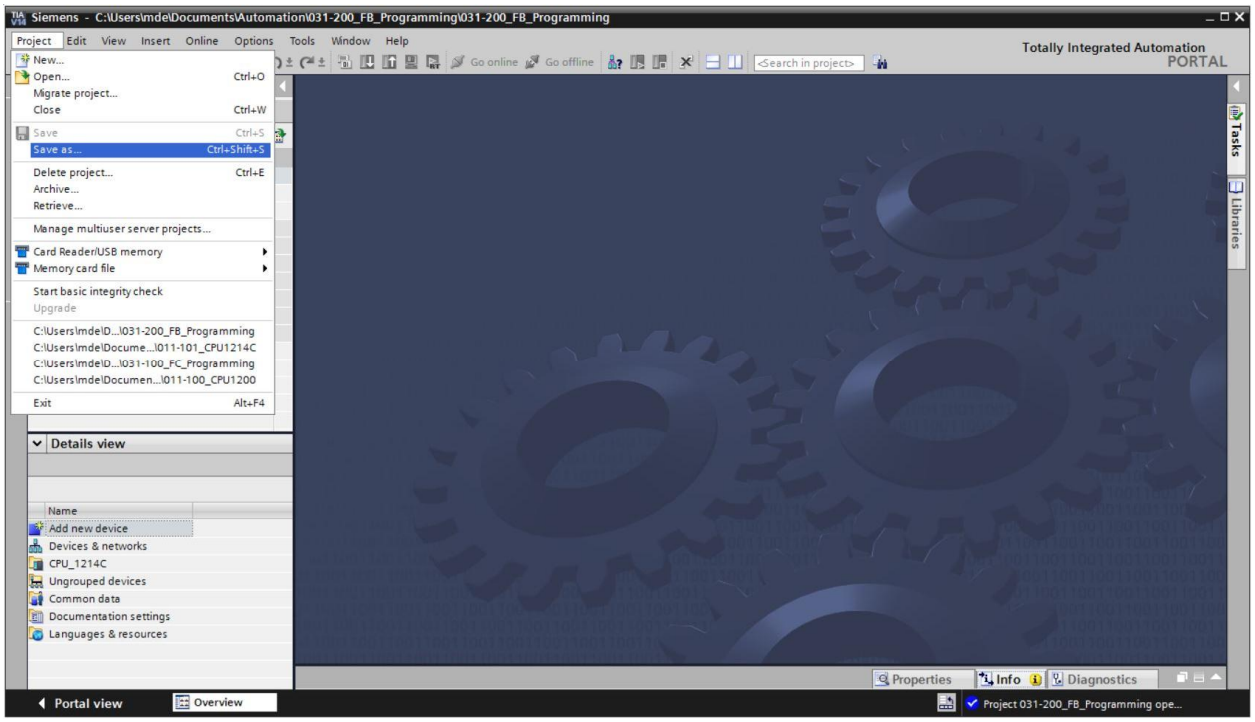
7.1 Retrieve an existing project

® Before we can expand the "MOTOR_AUTO [FB1]" function block, we must retrieve the "031-200_FB-Programming_S7-1200.zap14" project from chapter "SCE_EN_031-200 FB Programming S7-1200". 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 ® 031-200_FB-Programming_S7-1200.zap14 ® Open).



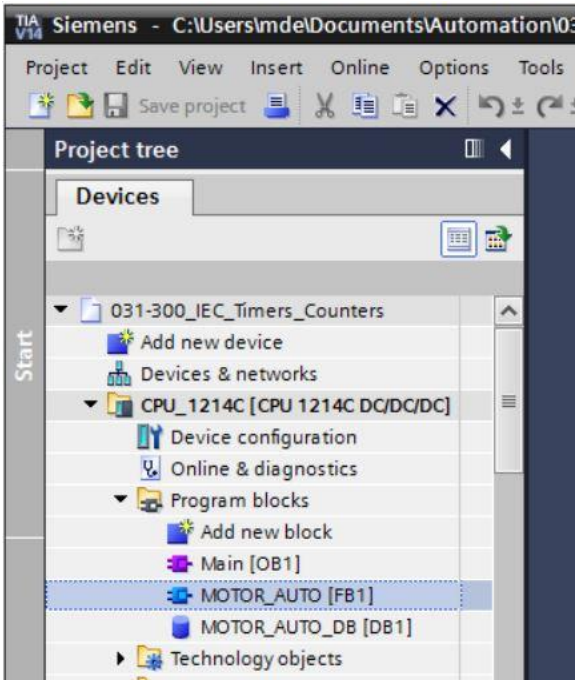
® The next step is to select the target directory where the retrieved project will be stored. Confirm your selection with "OK".

® Save the opened project under the name 031-300_IEC_Timers_Counters
(® Project ® Save as ... ® 031-300_IEC_Timers_Counters ® Save)

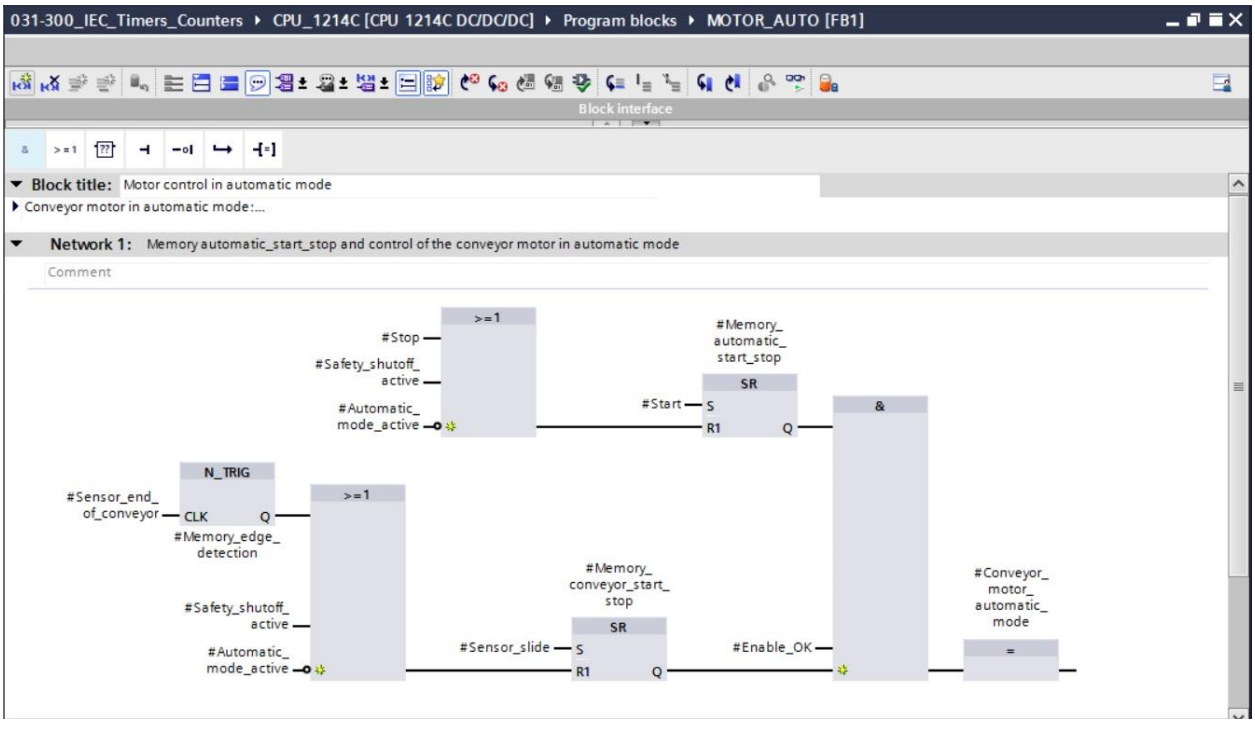


7.2 Addition of an IEC timer TP to function block FB1 "MOTOR_AUTO"

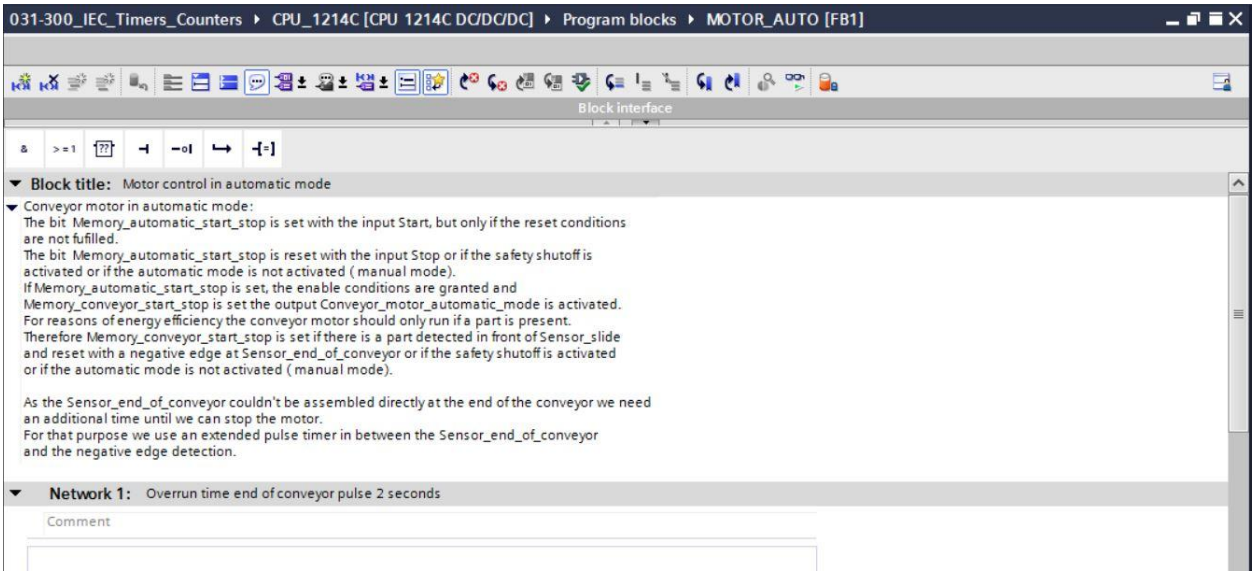
® First, open the "MOTOR_AUTO [FB1]" function block with a double-click.



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

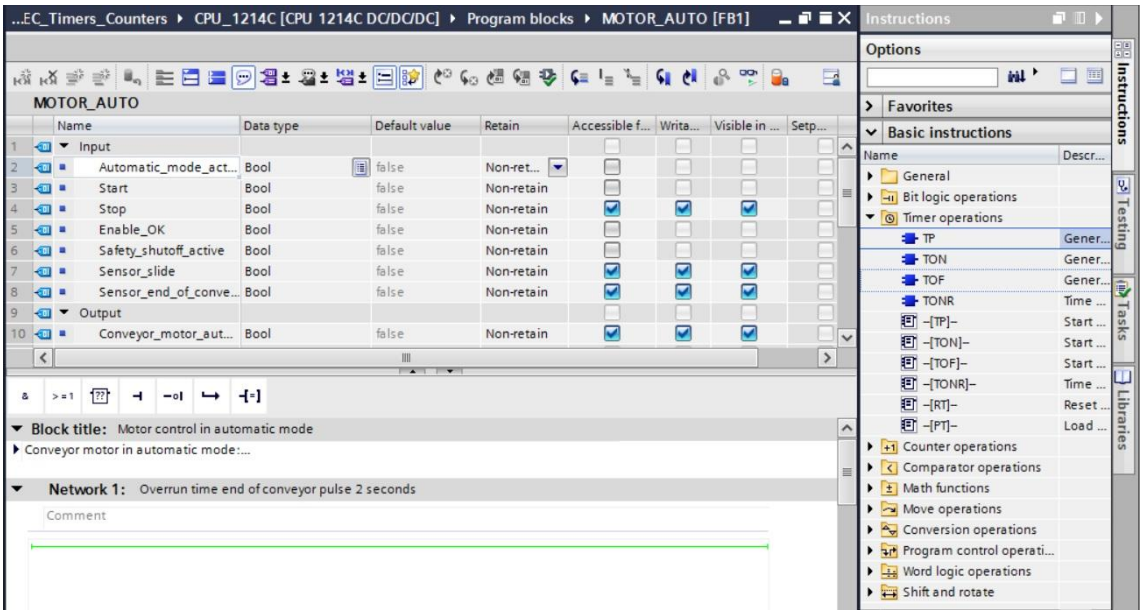


® Add helpful information to the block comment and the network title of "Network 1:".

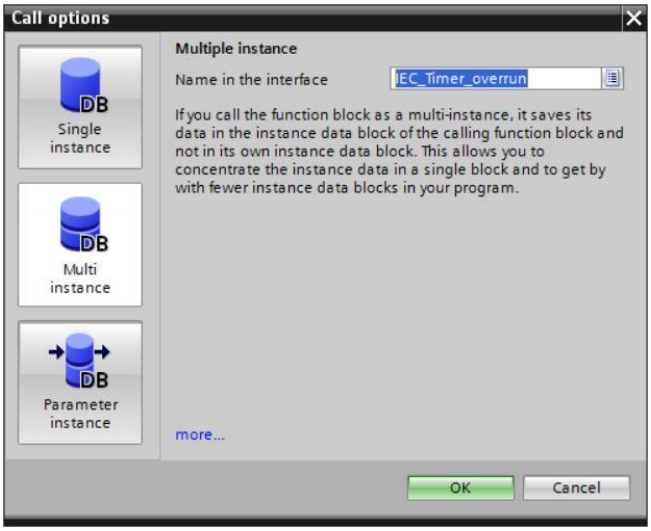


® On the right side of your programming window, you will see the timer functions in the list of instructions. Under ® Basic instructions ® Timer operations, find function TP (Generate pulse) and use a drag & drop operation to move it to Network 1 (green line appears, mouse pointer with + symbol).

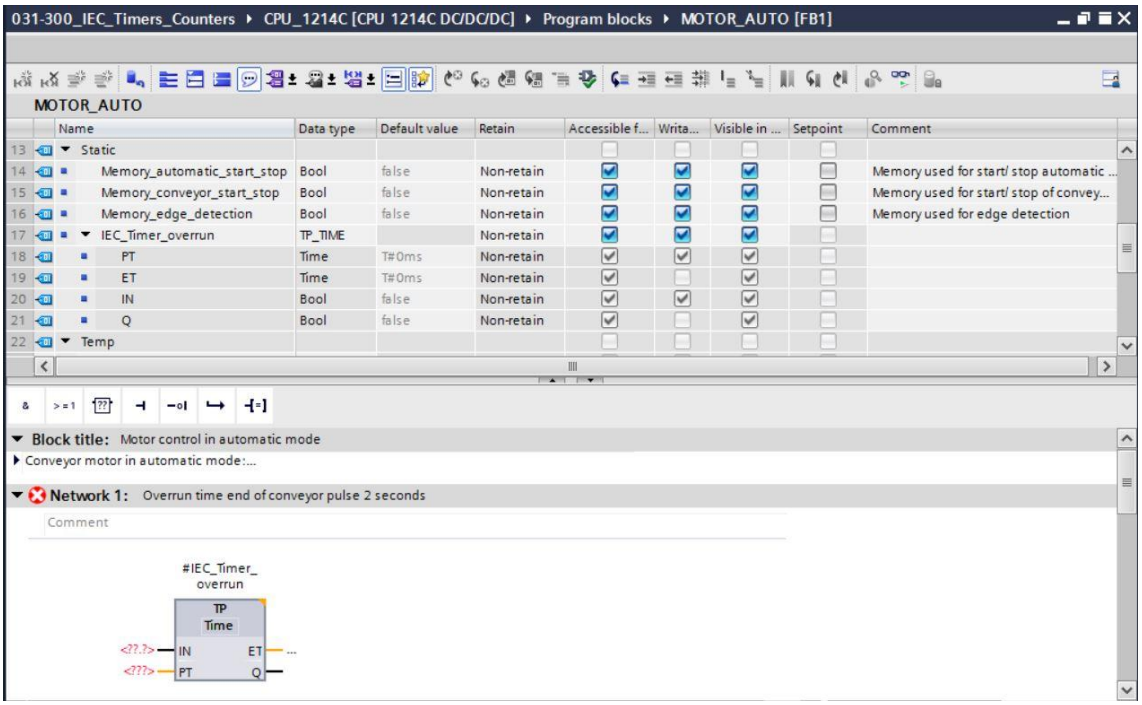
(® Instructions ® Basic instructions ® Timer operations ® TP)



® The timer function requires a memory. This memory is provided in this case within the instance data block of the function block without creating 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)

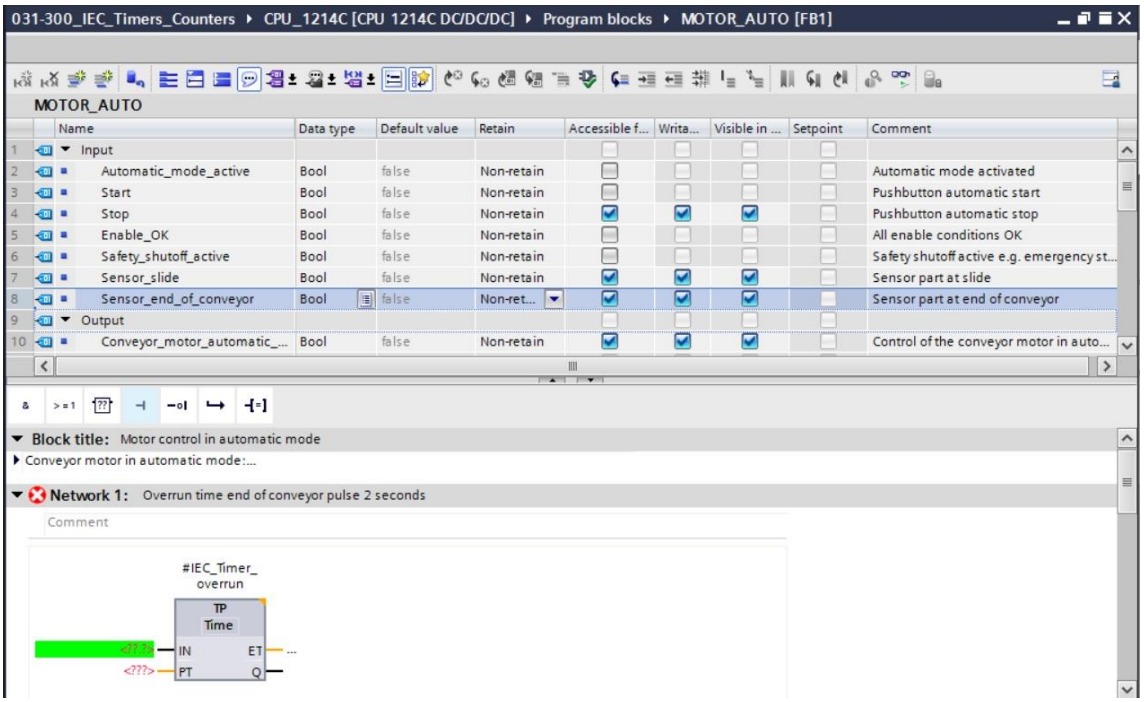


® As a result, a tag structure of "Static" type suitable for TP Timer will be created in the interface description.

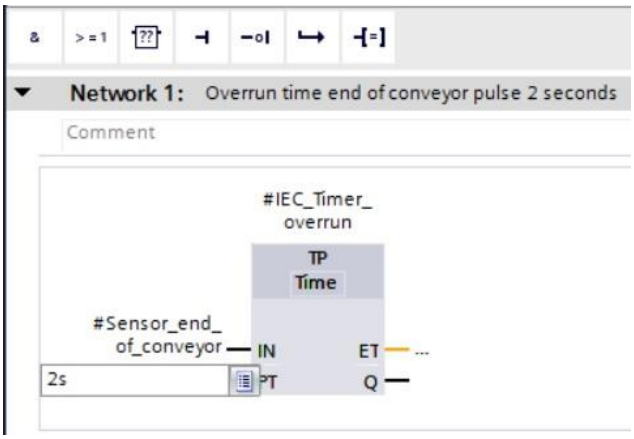


Note: *A multi-instance can only be used for programming within a function block because static tags are only available there.*

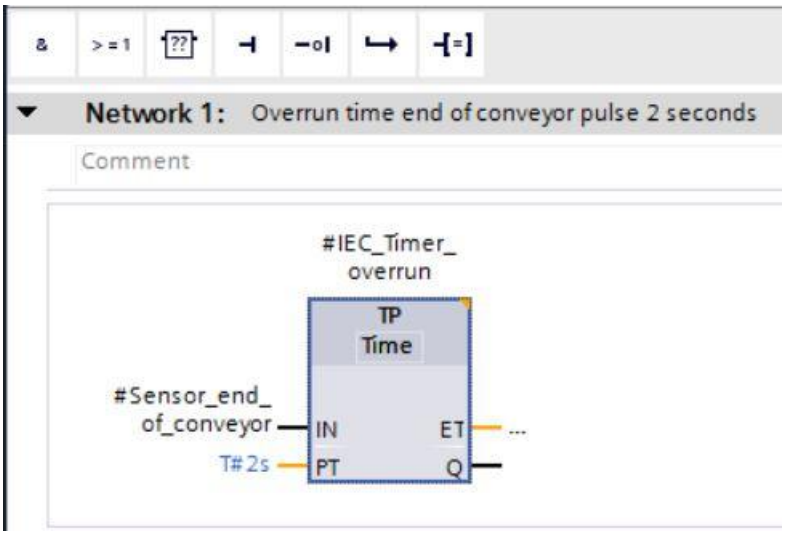
® Use drag & drop to move input parameter #Sensor_end_of_conveyor to <??.?> in front of parameter "IN" of TP Timer so that this will be started at a positive edge at input #Sensor_end_of_conveyor. The best way to select a parameter in the interface description is by "grabbing" it at the blue symbol (®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



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

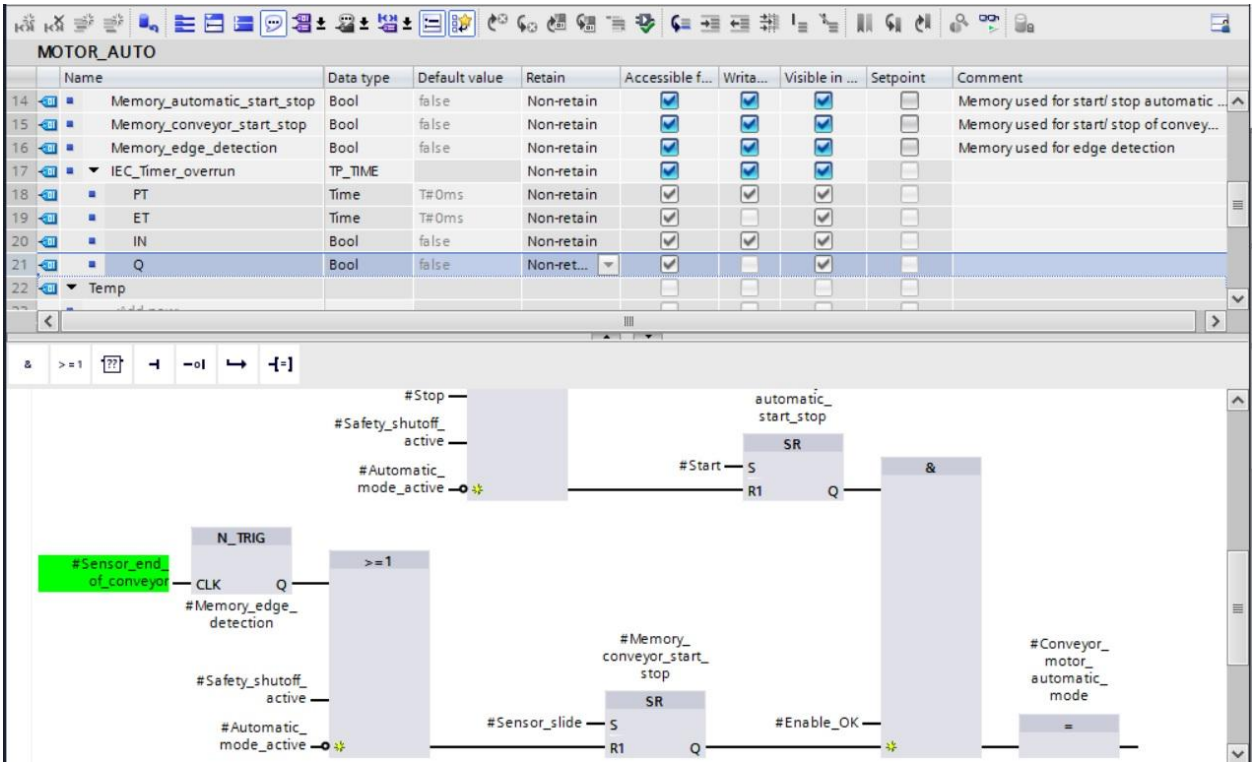


® The entry of 2s is converted automatically to the IEC-Time format suitable for the IEC timer and is shown as constant "T#2s".

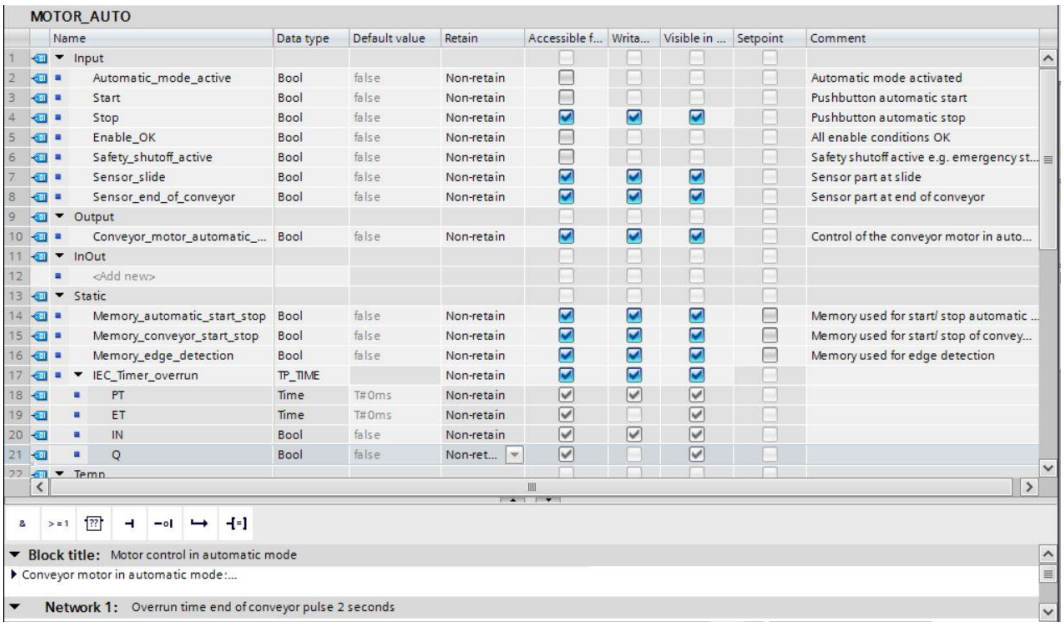


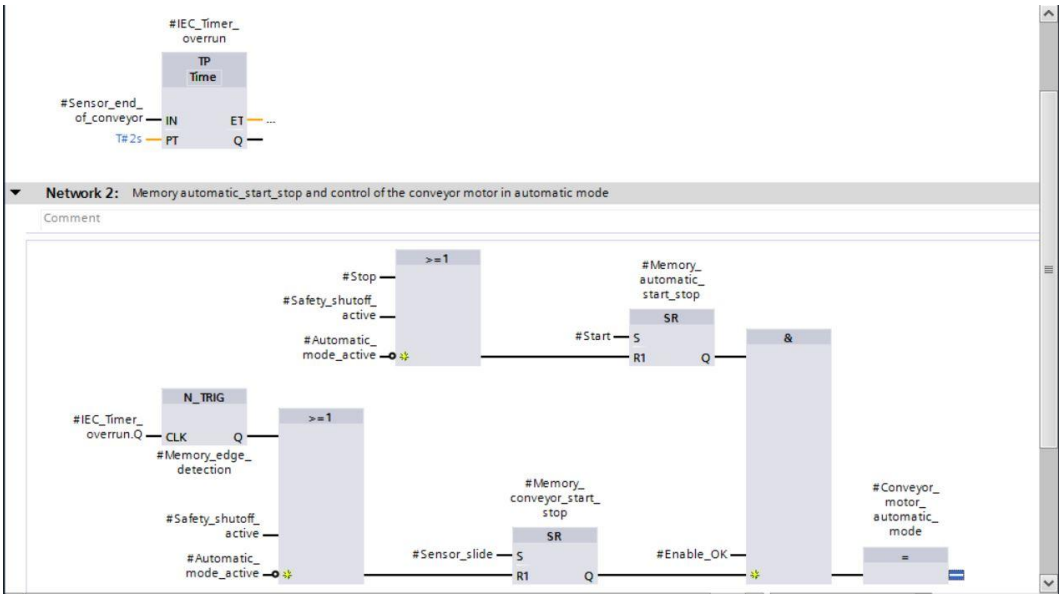
® Now move output "Q" from tag structure "IEC_Timer_overrun" onto input "CLK" of negative edge "N_TRIG" in Network 2. This will replace the #Sensor_end_of_conveyor input tag previously entered there and the conveyor will be stopped by a negative edge of the IEC_Timer_overrun pulse.

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

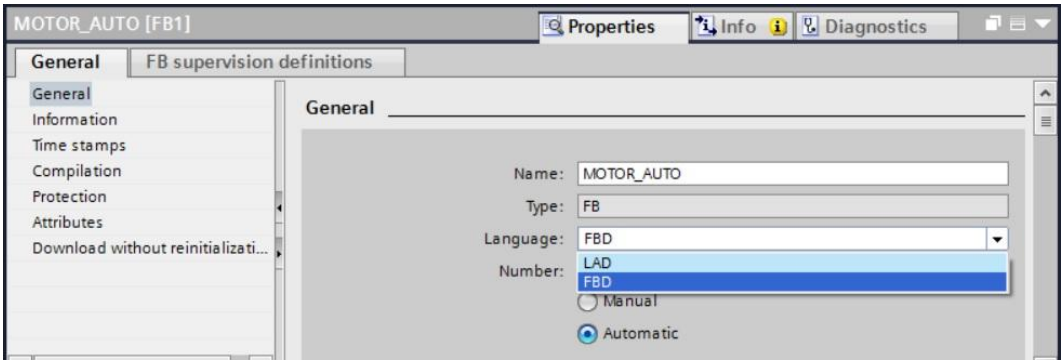


® Do not forget to click Save project regularly. The finished function block "MOTOR_AUTO" [FB1] with the timer is shown in FBD below.

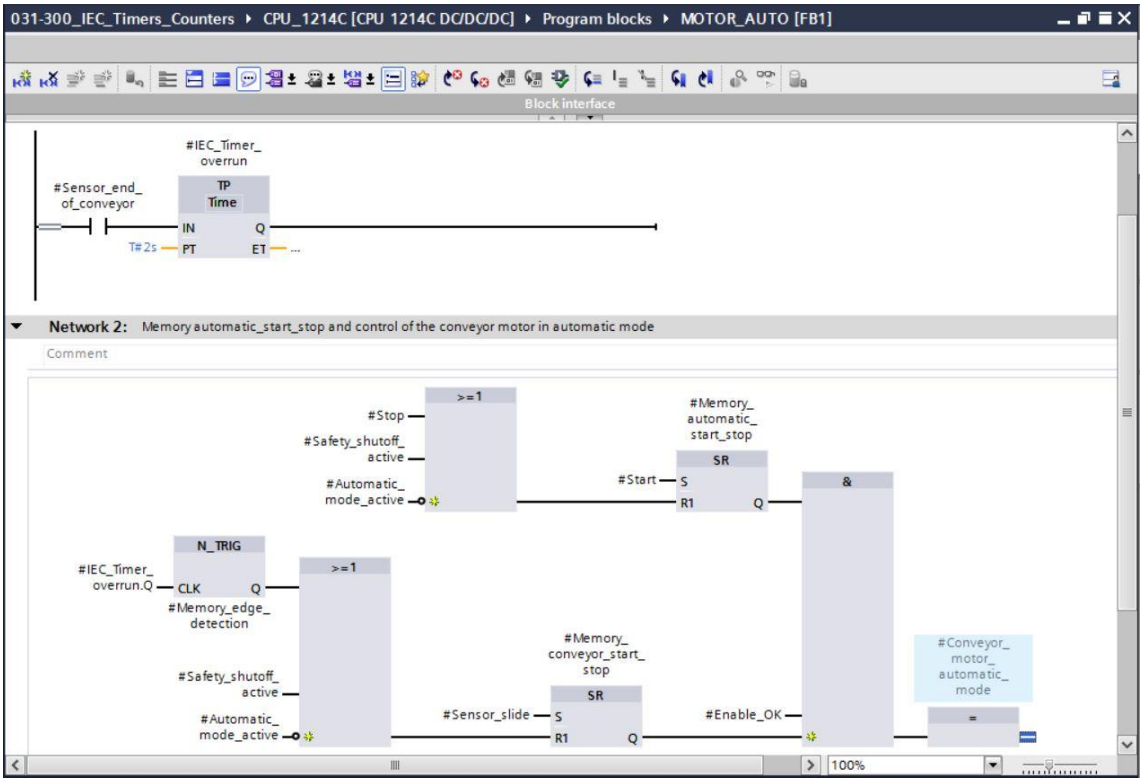




® Under "General" in the properties of the block, you can change the "Language" to LAD (Ladder Logic) (® Properties ® General ® Language: LAD)

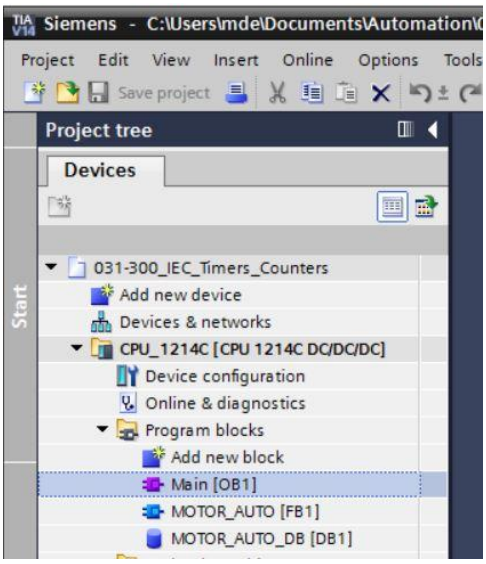


® This is what networks 1 and 2 look like in LAD.

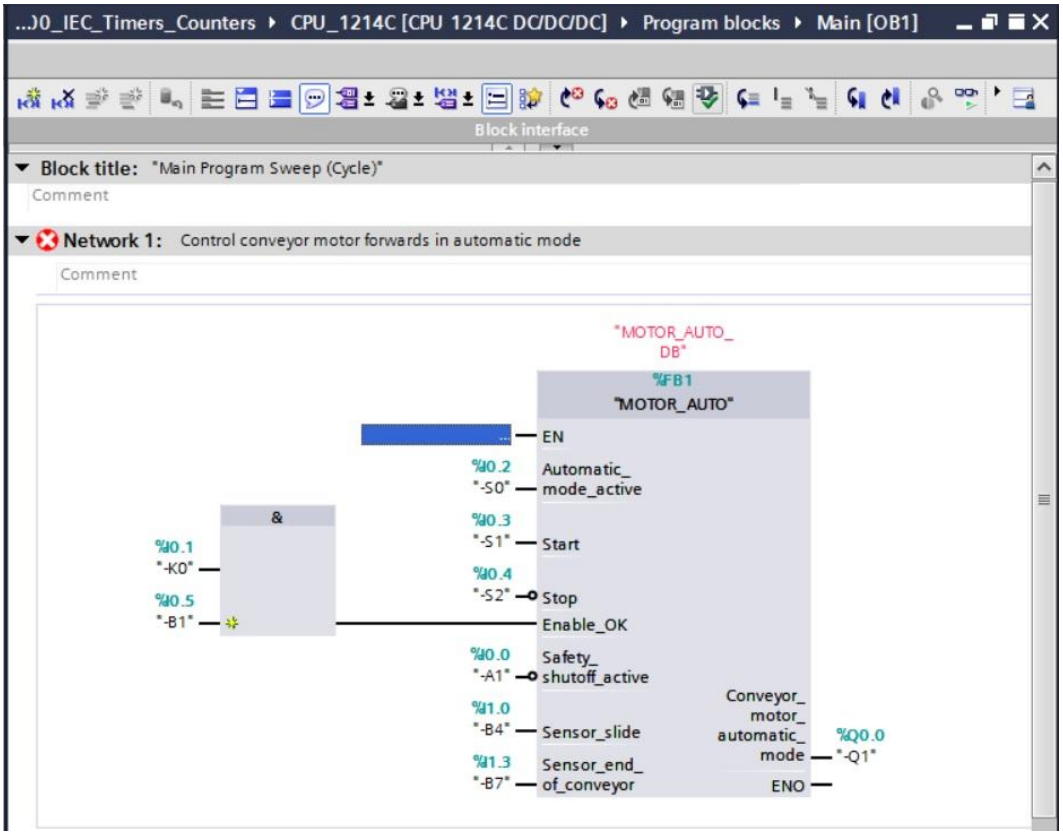


7.3 Update the block call in the organization block

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

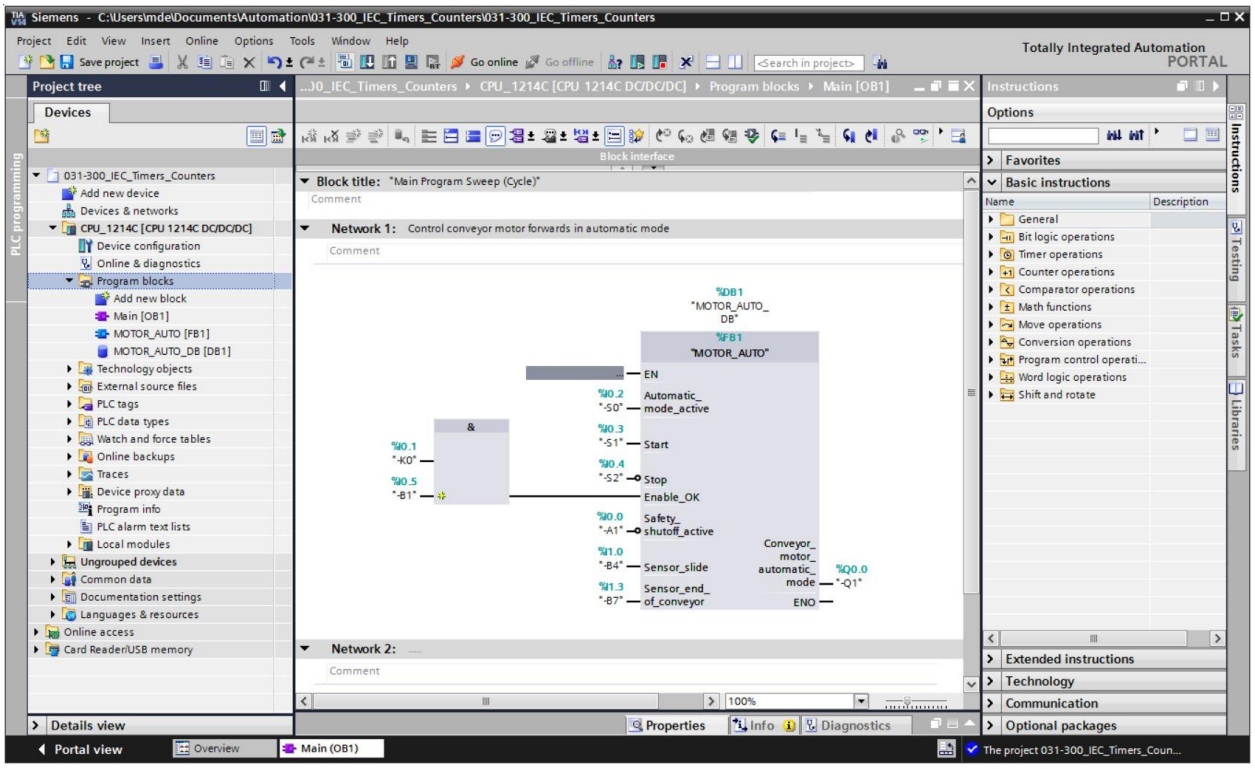


® In Network 1 of the "Main [OB1" organization block, instance data block "MOTOR_AUTO_DB1" for the "MOTOR_AUTO [FB1]" function block appears incorrect, because the additional memory for the TP Timer has not yet been added there. Click the ® "  " icon for "Update inconsistent block calls". This will add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

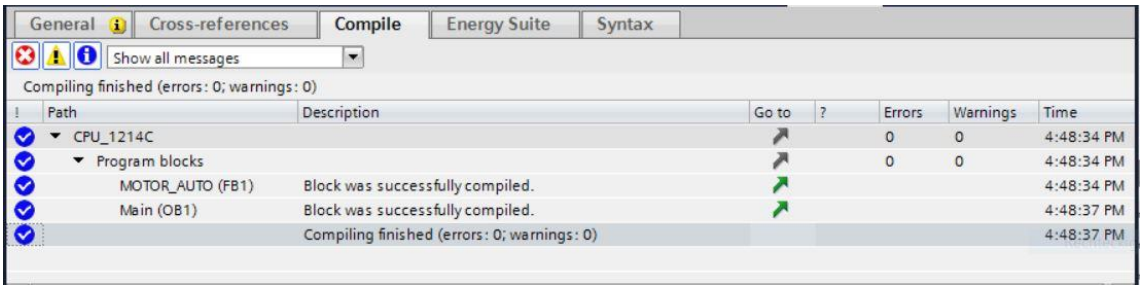


7.4 Save and compile the program

® 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 ®).

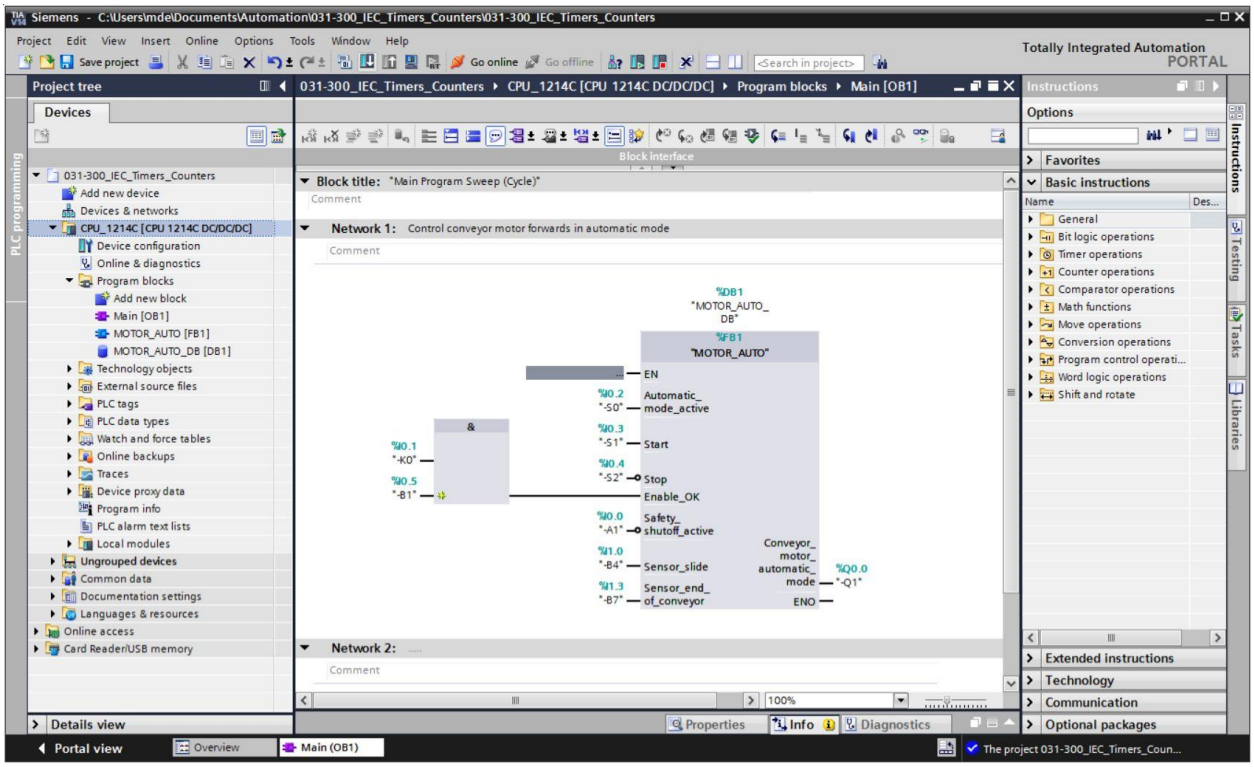


® The "Info", "Compile" area shows which blocks were successfully compiled.



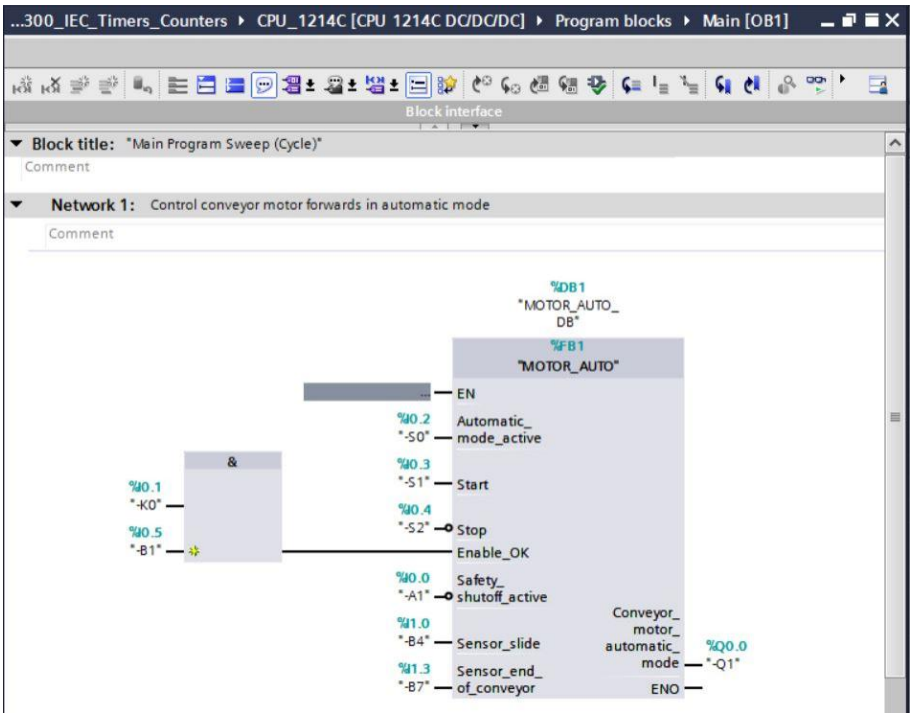
7.5 Download the program

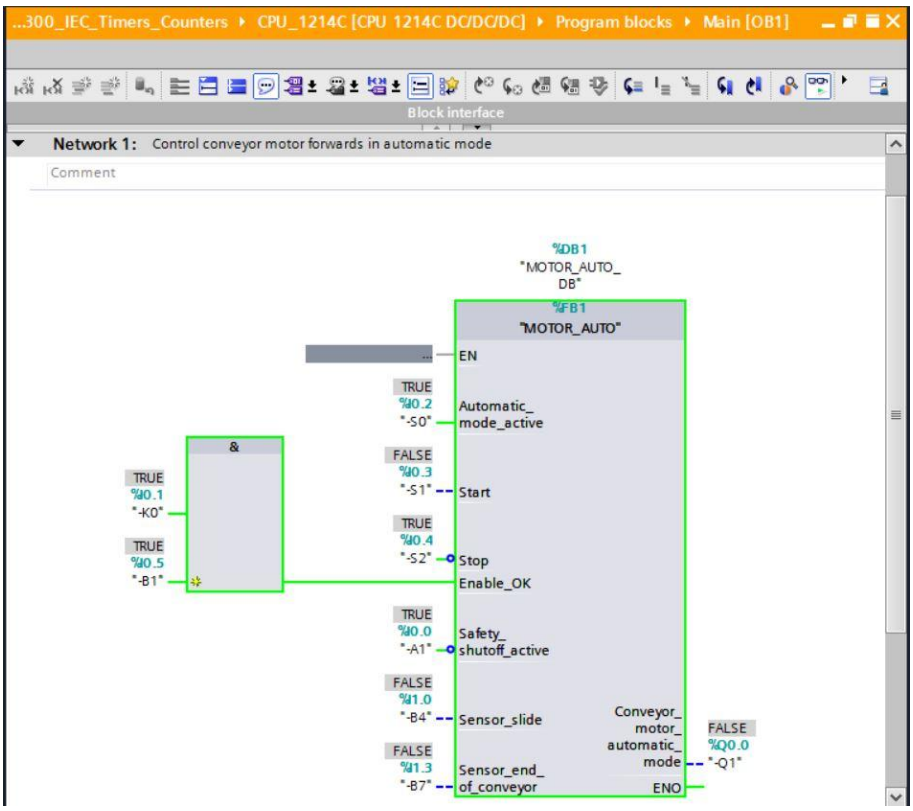
® After successful compilation, the complete controller with the created program including the hardware configuration, as previously described in the modules, can be downloaded (® ).



7.6 Monitor program blocks

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

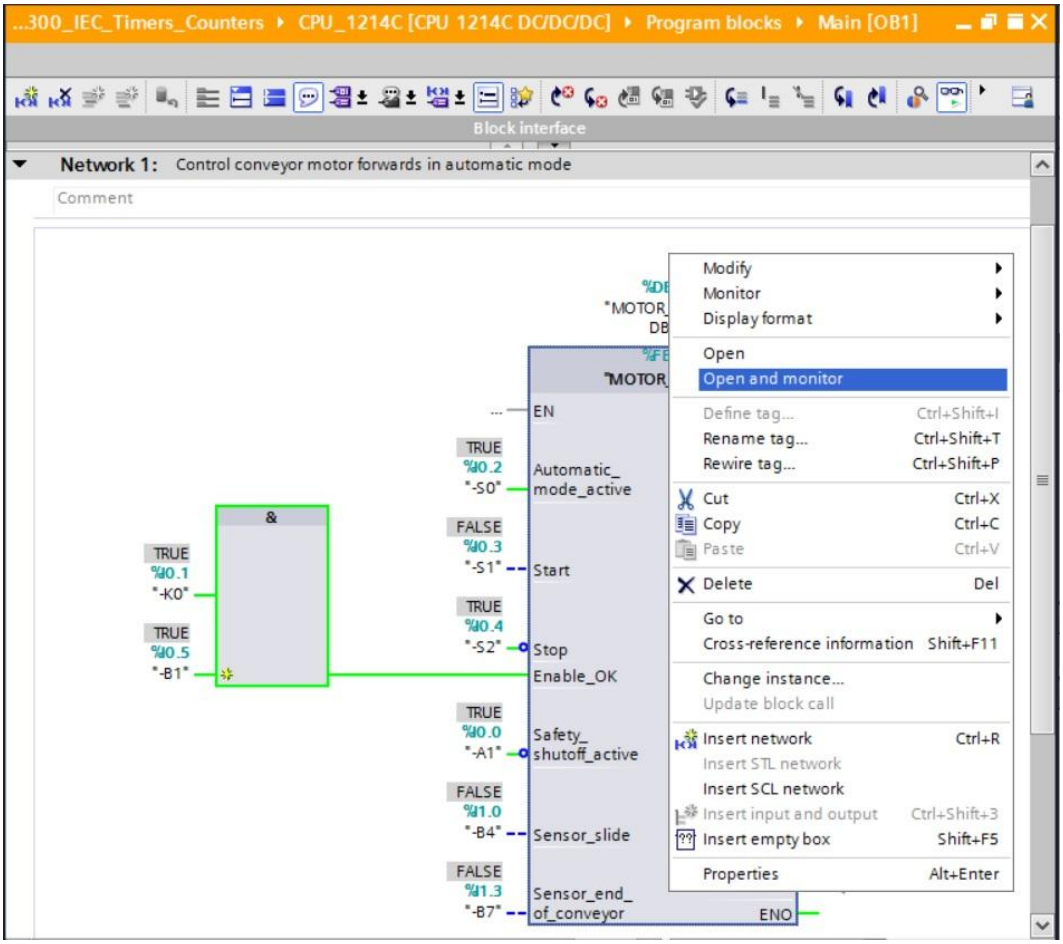


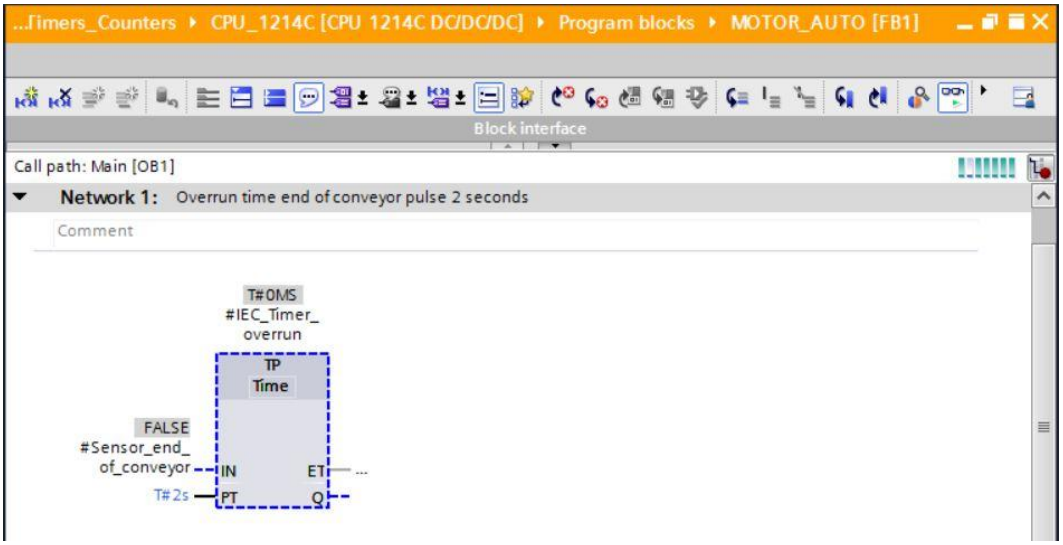


Note: *The monitoring here is signal-related and controller-dependent. The signal states at the terminals are indicated with TRUE or FALSE.*

® 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).

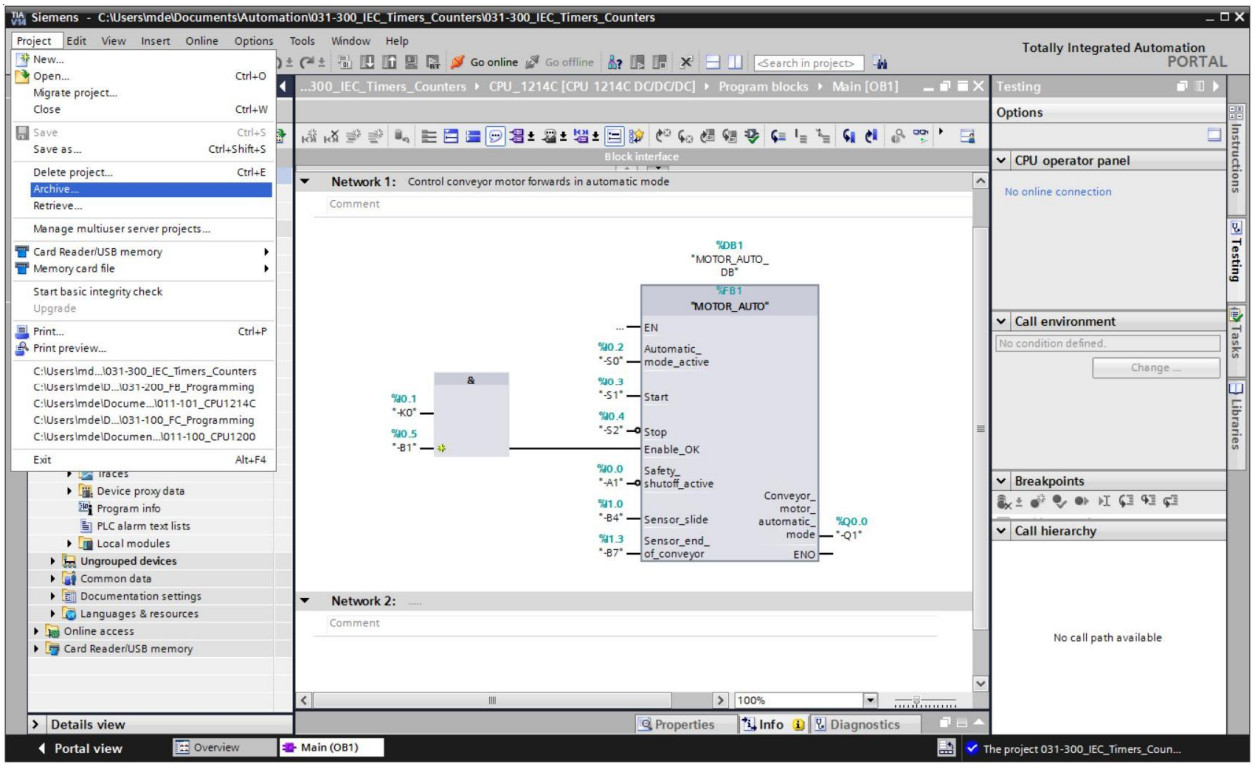




Note: *The monitoring here is function-related and controller-independent. The actuation of sensors and the station status are shown here with TRUE or FALSE.*

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 ® SCE_EN_031-300_IEC_Timers_Counters_S7-1200…. ® Save)



7.8 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) Conveyor motor forwards fixed speed then switches on (-Q1 = 1) 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	

8 Exercise

8.1 Task – Exercise

In this exercise, an IEC counter is 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. The parts are 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.

8.2 Technology diagram

Here, you see the technology diagram for the task.

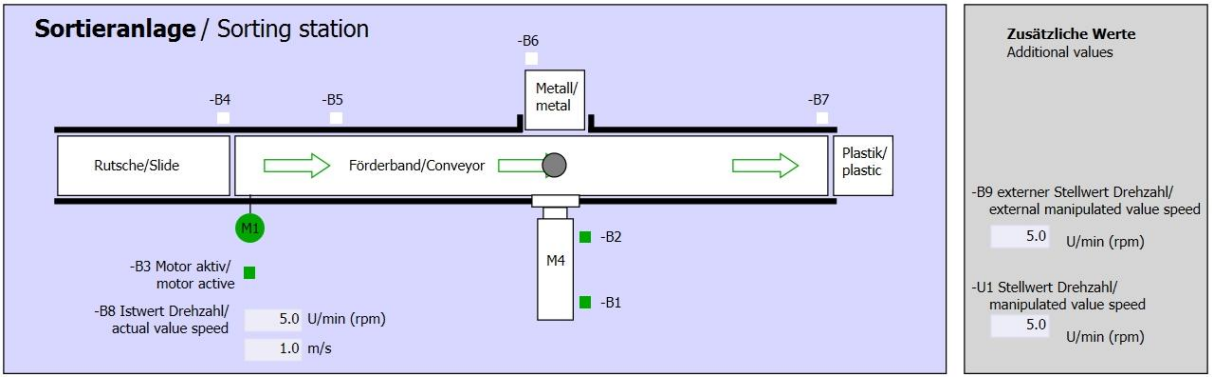


Figure 3: Technology diagram

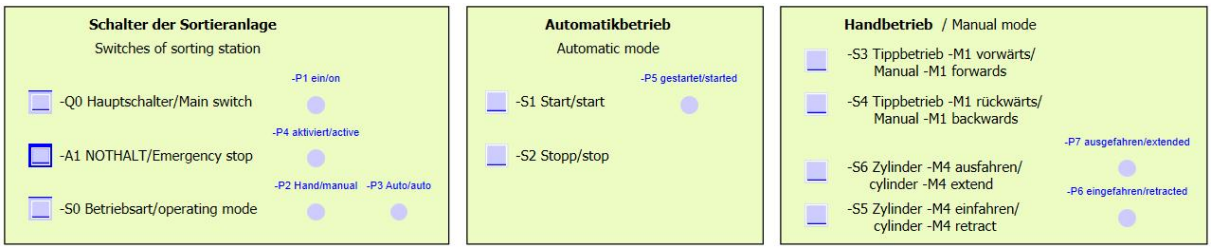


Figure 4: Control panel

8.3 Reference list

The following signals are needed as global operands for this task.

DI	Type	Identifier	Function	NC/NO
I 0.0	BOOL	-A1	Return signal emergency stop ok	NC
I 0.1	BOOL	-K0	Main switch "ON"	NO
I 0.2	BOOL	-S0	Mode selector manual (0)/ automatic (1)	Manual = 0 Auto = 1
I 0.3	BOOL	-S1	Pushbutton automatic start	NO
I 0.4	BOOL	-S2	Pushbutton automatic stop	NC
I 0.5	BOOL	-B1	Sensor cylinder M4 retracted	NO
I 1.0	BOOL	-B4	Sensor at chute occupied	NO
I 1.3	BOOL	-B7	Sensor part at end of conveyor	NO

DO	Type	Identifier	Function	
Q 0.0	BOOL	-Q1	Conveyor motor M1 forwards fixed speed	

Legend for reference list

DI	Digital Input	DO	Digital Output
AI	Analog Input	AO	Analog Output
I	Input	Q	Output
NC	Normally Closed		
NO	Normally Open		

8.4 Planning

Plan the implementation of the task on your own.

Note: *Learn about the use of IEC counters in SIMATIC S7-1200 in the online help.*

8.5 Checklist – Exercise

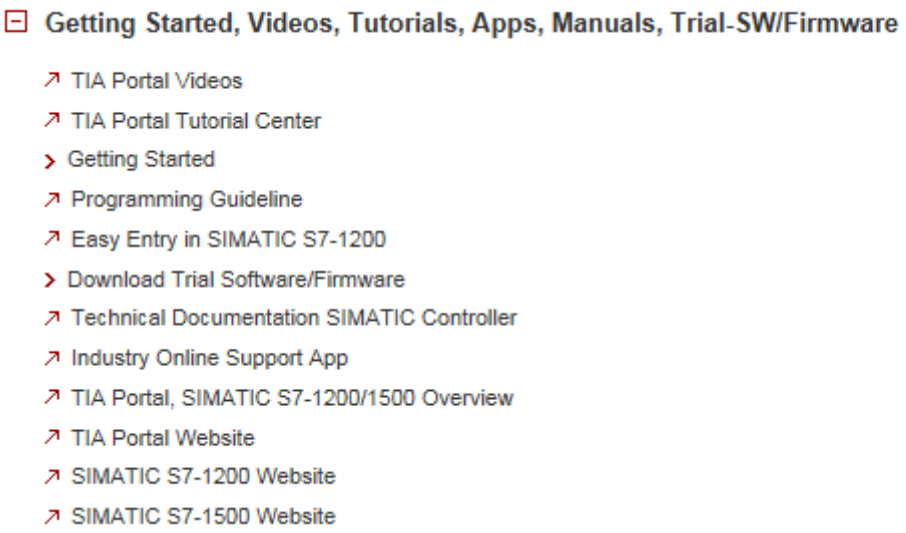
No.	Description	Completed
1	Compiling successful and without error message	
2	Download successful and without error message	
3	Switch on station (-K0 = 1) Cylinder retracted / Feedback activated (-B1 = 1) EMERGENCY OFF (-A1 = 1) not activated AUTOMATIC mode (-S0 = 1) Pushbutton automatic stop not actuated (-S2 = 1) Briefly press the automatic start pushbutton (-S1 = 1) Sensor at chute activated (-B4 = 1) Conveyor motor forwards fixed speed then switches on (-Q1 = 1) 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	

9 Additional information

More information for further practice and consolidation is available as orientation, for example: Getting Started, videos, tutorials, apps, manuals, programming guidelines and trial software / firmware, under the following link:

www.siemens.com/sce/s7-1200

Preview „Additional information“



Further Information

Siemens Automation Cooperates with Education
siemens.com/sce

SCE Learn-/Training Documents
siemens.com/sce/documents

SCE Trainer Packages
siemens.com/sce/tp

SCE Contact Partners
siemens.com/sce/contact

Digital Enterprise
siemens.com/digital-enterprise

Industrie 4.0
siemens.com/future-of-manufacturing

Totally Integrated Automation (TIA)
siemens.com/tia

TIA Portal
siemens.com/tia-portal

SIMATIC Controller
siemens.com/controller

SIMATIC Technical Documentation
siemens.com/simatic-docu

Industry Online Support
support.industry.siemens.com

Product catalogue and online ordering system Industry Mall
mall.industry.siemens.com

Siemens AG
Digital Factory
P.O. Box 4848
90026 Nuremberg
Germany