SIEMENS

Learn-/Training Document

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

TIA Portal Module 031-600
Global Data Blocks for the 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

Please 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, please contact 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

1 Goal 4
2 Prerequisite 4
3 Required hardware and software 5
4 Theory 6
4.1 Data blocks 6
4.2 Data types of the SIMATIC S7-1200 7
4.3 Optimized blocks 8
4.4 Downloading without reinitialization 8
5 Task 9
6 Planning 9
6.1 Global data block for speed control and speed monitoring of the motor 9
6.2 Technology diagram 10
6.3 Reference list 11
7 Structured step-by-step instructions 12
7.1 Retrieve an existing project 12
7.2 Create the global data block "SPEED_MOTOR" 14
7.3 Access to data of the data block in the organization block 19
7.4 Save and compile the program 23
7.5 Download the program 24
7.6 Monitor/modify values in data blocks 25
7.7 Initialize setpoints / reset start values 26
7.8 Snapshots in data blocks 28
7.9 Expand data block and download it without reinitialization 31
7.10 Archive the project 35
8 Checklist 36
9 Exercise 37
9.1 Task – Exercise 37
9.2 Technology diagram 37
9.3 Reference list 38
9.4 Planning 38
9.5 Checklist – Exercise 39
10 Additional information 40

Global Data Blocks for the SIMATIC S7-1200

1 Goal

In this chapter, you will become acquainted with the use of global data blocks for the SIMATIC S7-1200 with the TIA Portal programming tool.

The module explains the structure and creation of and access to global data blocks for the SIMATIC S7-1200. It also shows the steps for creating a global data block in the TIA Portal and for accessing this data in the program with read and write access.

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

2 Prerequisite

This chapter builds on the chapter Analog Values with the SIMATIC S7 CPU1214C DC/DC/DC. You can use the following project for this chapter, for example: "SCE_EN_031-500_Analog_Values_S7-1200.zap14“.

3 Required hardware and software

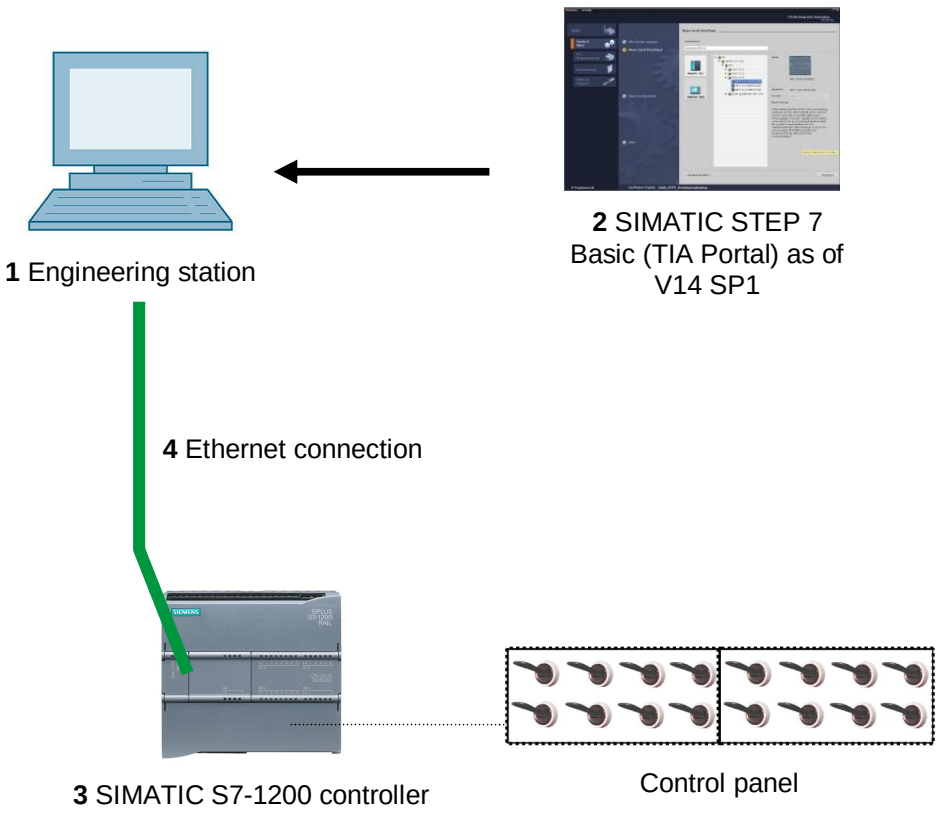
1 Engineering station: requirements include hardware and operating system (for additional information, see Readme on the TIA Portal Installation DVDs)

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 and analog inputs and outputs should be fed out to a control panel.

4 Ethernet connection between engineering station and controller



4 Theory

4.1 Data blocks

In contrast to logic blocks, data blocks contain no instructions. Rather, they serve as memory for user data.

Data blocks thus contain variable data that is used by the user program. You can define the structure of global data blocks as required.

Global data blocks store data that can be used ***by all other blocks*** (see Figure 1). Only the associated function block should access instance data blocks. The maximum size of data blocks varies depending on the utilized CPU.

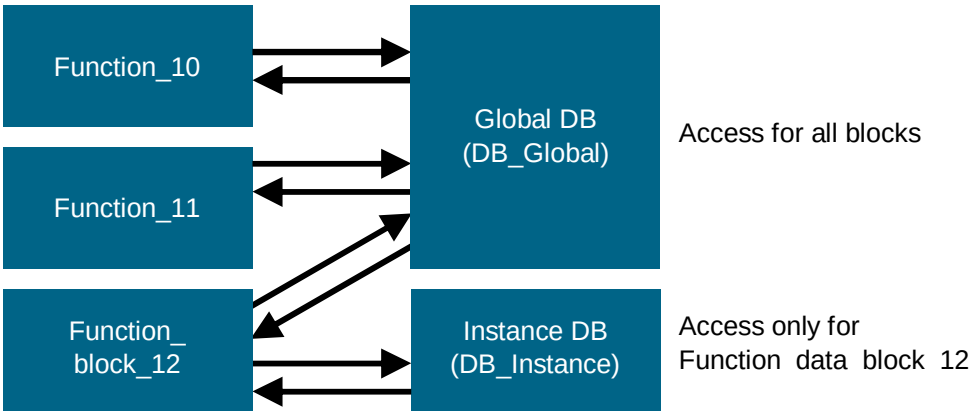


Figure 1: Difference between global DB and instance DB.

Application examples for **global data blocks** are:

– Saving of information about a storage system. "Which product is located where?"

– Saving of recipes for particular products.

The data in data blocks is stored retentively in most cases. This data is then retained in the event of a power failure or after a STOP/START of the CPU.

4.2 Data types of the SIMATIC S7-1200

The SIMATIC S7-1200 has many different data types for representing different numerical formats. A list of some of the elementary data types is given below.

Data type	Size (bits)	Range	Example of constant entry
Bool	1	0 to 1	TRUE, FALSE, O, 1
Byte	8	16#00 to 16#FF	16#12, 16#AB
Word	16	16#0000 to 16#FFFF	16#ABCD, 16#0001
DWord	32	16#00000000 to 16#FFFFFFFF	16#02468ACE
Char	8	16#00 to 16#FF	'A', ‘r’, ‘@’
Sint	8	-128 to 127	123,-123
Int	16	-32,768 to 32,767	123, -123
Dint	32	-2,147,483,648 to 2,147,483,647	123, -123
USInt	8	0 to 255	123
Ulnt	16	0 to 65,535	123
UDInt	32	0 to 4,294,967,295	123
Real	32	$+/-1.18 \times 10^{-38}$ to $+/-3.40 \times 10^{38}$	123.456, -3.4, 1.2E+12 3.4E-3
LReal	64	$+/-2.23 \times 10^{-308}$ to $+/-1.79 \times 10^{308}$	12345.123456789 -1.2E+40
Time	32	T#-24d_20h_31 m_23s_648ms to T#24d_20h_31 m_23s_647ms Saved as: -2,147,483,648 ms to +2,147,483,647 ms	T#5m_30s 5#-2d T#1d_2h_15m_30x_45ms
String	Variable	0 to 254 characters in byte size	'ABC'
Array		With arrays, data of a uniform data type is arranged one after the other and addressed consecutively in the address area. The properties of each array element are identical and are configured in the array tag.	
Struct		The STRUCT data type represents a data structure that consists of a fixed number of components of different data types. Components of STRUCT or ARRAY data type can also be nested in a structure.	
…		For other data types, refer to the online help.	

4.3 Optimized blocks

S7-1200 controllers have optimized data storage. In optimized blocks all tags are automatically sorted based on their data type. The sorting ensures that data gaps between the tags are minimized and the tags are stored in a manner that optimizes their access by the controller.

- The tags are always accessed as fast as possible because the file storage by the system is optimized and is independent of the declaration.
- There is no danger of inconsistencies due to incorrect, absolute accesses because symbolic access is generally used.
- Declaration changes do not result in access errors because accesses by process visualization systems, for example, occur symbolically.
- Individual tags can be selectively defined as retentive.
- No settings are needed or possible in the instance data block. Everything will be set in the assigned FB (e.g., retentivity).
- Memory reserves in the data block enable changes to be made without loss of actual values (download without reinitialization).

4.4 Downloading without reinitialization

To enable the subsequent editing of user programs that are already running in a CPU, the S7-1200 controllers support the option of expanding the interfaces of optimized function or data blocks during operation. You can download the modified blocks without switching the controller to STOP mode and without affecting the actual values of previously downloaded tags.

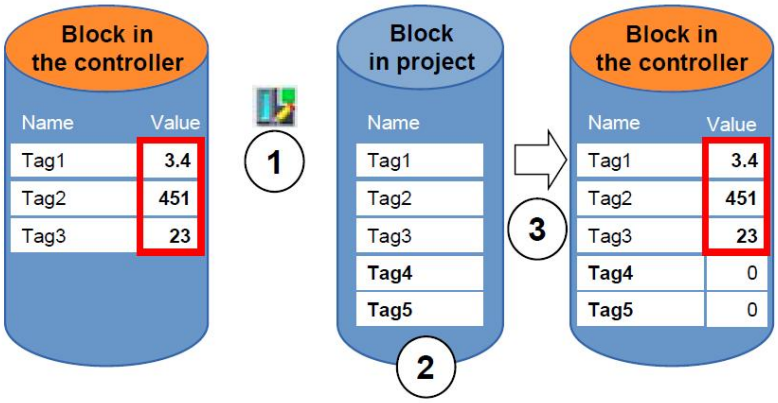


Figure 2: Download without reinitialization

The following steps can be performed while the controller is in RUN mode:

1. Activate "Download without reinitialization"
2. Insert newly defined tags in an existing block
3. Download expanded block to the controller

The newly defined tags are initialized. The existing tags retain their current value.

Prerequisite: a memory reserve must have been defined for the block beforehand and the block with this memory reserve must have downloaded to the CPU.

5 Task

In this chapter, the program from chapter "SCE_EN_031-500 Analog Values_S7-1200“ will be expanded to include a data block that centrally provides the parameters for the two functions "MOTOR_SPEEDCONTROL“ [FC10] and "MOTOR_ SPEEDMONITORING“ [FC11].

6 Planning

The data management and setpoint setting for the "MOTOR_SPEEDCONTROL" [FC10] and "MOTOR_SPEEDMONITORING" [FC11] functions will be carried out using the global data block "SPEED_MOTOR" [DB2].

This will be added to the "031-500_Analog_Values_S7-1200" project. This project must be retrieved from the archive beforehand.

In the "Main" [OB1] organization block, the two functions "MOTOR_SPEEDCONTROL" [FC10] and "MOTOR_SPEEDMONITORING" [FC11] must then be connected with the tags from global data block "SPEED_MOTOR" [DB2].

6.1 Global data block for speed control and speed monitoring of the motor

Speed setpoint and actual speed value will be created in Real data format (32-bit floating-point number) as the first tags in the "SPEED_MOTOR" [DB2] data block. The speed setpoint is thereby given the start value + 10 rpm.

A structure (Struct) 'Positive_Speed' will then be created for monitoring the positive speed limits.

This structure contains the 2 tags 'Threshold_Error' (start value + 15 rpm) and 'Threshold_ Warning' (start value + 10 rpm) in Real data format (32-bit floating-point number) and the 2 tags 'Error' and 'Warning' in Bool data format (binary number).

The structure (Struct) 'Positive_Speed' will then be inserted again as a copy and renamed to 'Negative_Speed' for monitoring the negative speed limits.

The 'Threshold_Error' tag is given the start value - 16 rpm and the 'Threshold_Warning' tag the start value - 14 rpm.

6.2 Technology diagram

Here you see the technology diagram for the task.

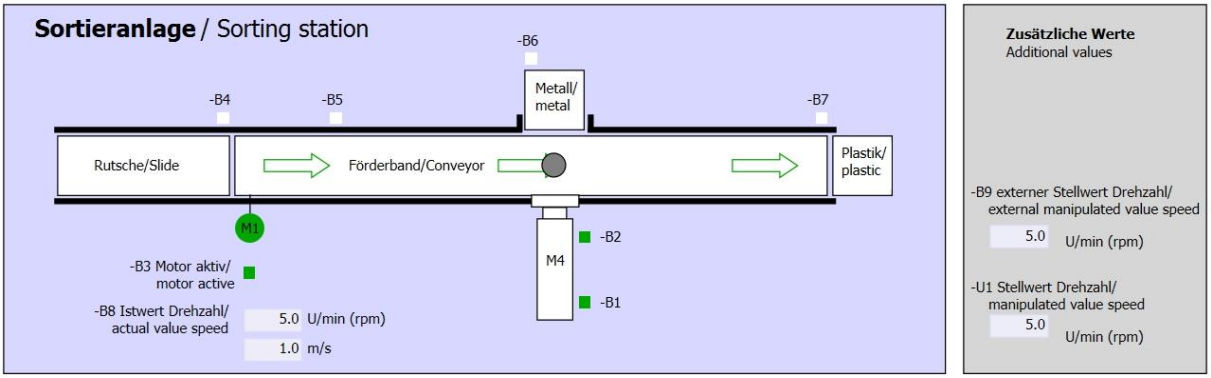


Figure 3: Technology diagram

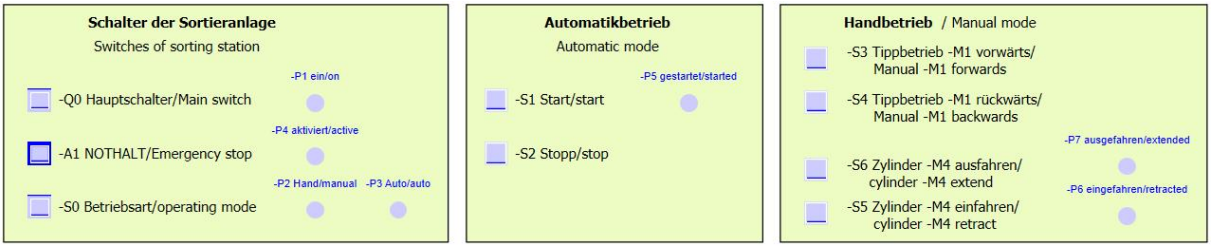


Figure 4: Control panel

6.3 Reference list

The following signals are required 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 part at slide	NO
I 1.3	BOOL	-B7	Sensor part at end of conveyor	NO
IW64	BOOL	-B8	Sensor actual value speed of the motor +/-10V corresponds to +/- 50 rpm	

DO	Type	Identifier	Function	
Q 0.2	BOOL	-Q3	Conveyor motor -M1 variable speed	
QW 64	BOOL	-U1	Manipulated value speed of the motor in 2 directions +/- 10V corresponds to +/- 50 rpm	

Legend for reference list

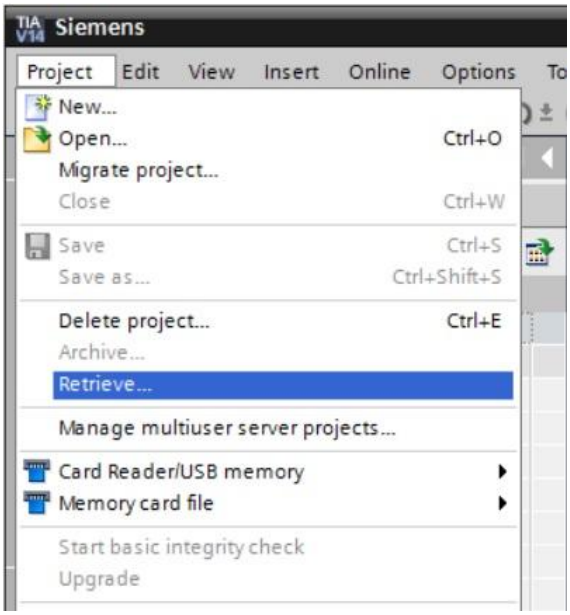
DI	Digital Input	DO	Digital Output
AI	Analog Input	AO	Analog Output
I	Input	Q	Output
NC	Normally Closed		
NO	Normally Open		

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

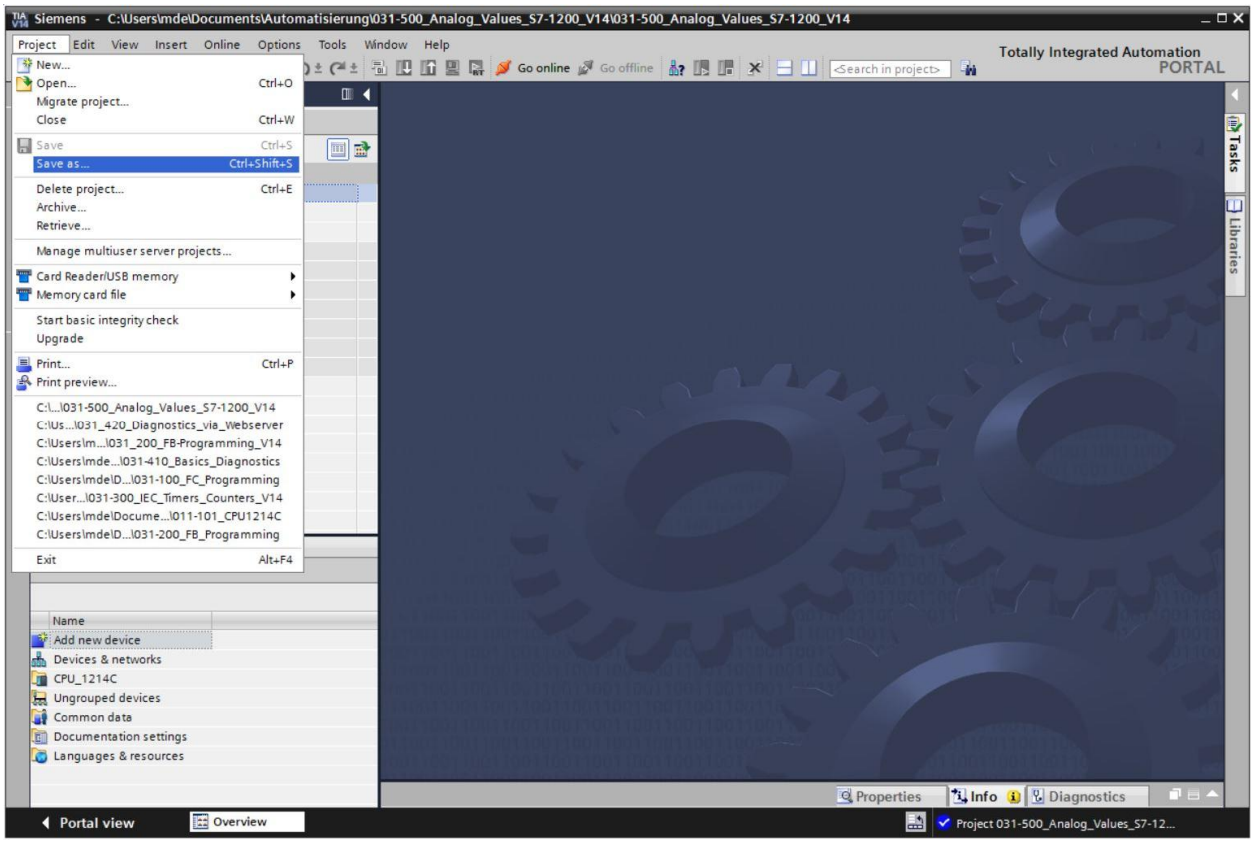
® Before we can expand the "SCE_EN_031-500_Analog_Values_S7-1200.zap14" project from chapter "SCE_EN_031-500_Analog_Values_S7-1200", we must retrieve this project from the archive. To retrieve an existing project that has been archived, you must select the relevant archive with ® Project ® Retrieve in 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)

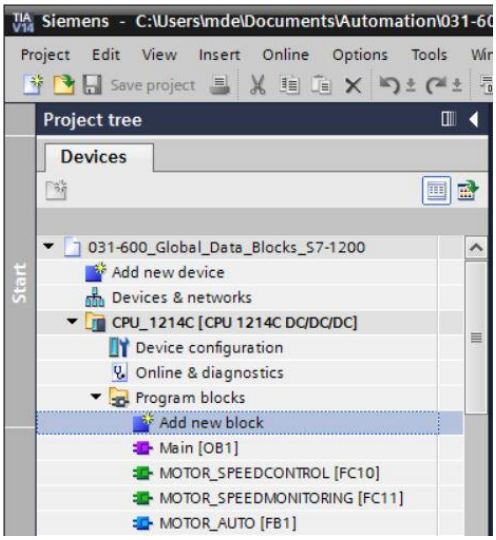
® Save the opened project under the name 031-600_Global_Data_Blocks_S7-1200. (® Project ® Save as … ® 031-600_Global_Data_Blocks_S7-1200 ® Save)



7.2 Create the global data block "SPEED_MOTOR"

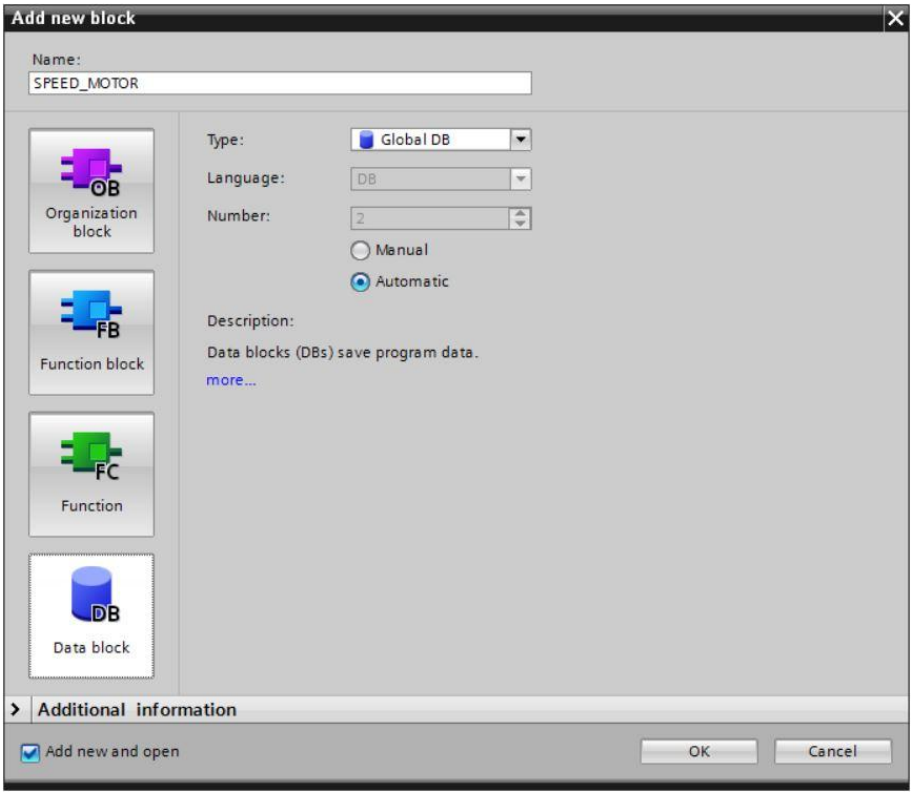
® Select the 'Program blocks' folder of your CPU 1214C DC/DC/DC and then click "Add new block" to create a new global data block there.

(® CPU_1214C [CPU 1214C DC/DC/DC] ® Add new block)



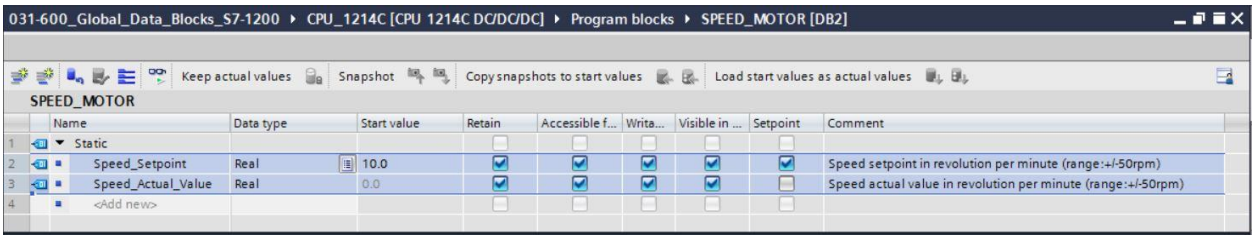
® Select Data block in the next dialog and rename your new block to: "SPEED_MOTOR". Select 'Global DB' as the type. The number '2' will be automatically assigned. Select the "Add new and open" check box. Click "OK".

(® Data block ® Name: SPEED_MOTOR ® Type: Global DB ® ☑ Add new and open ® OK)



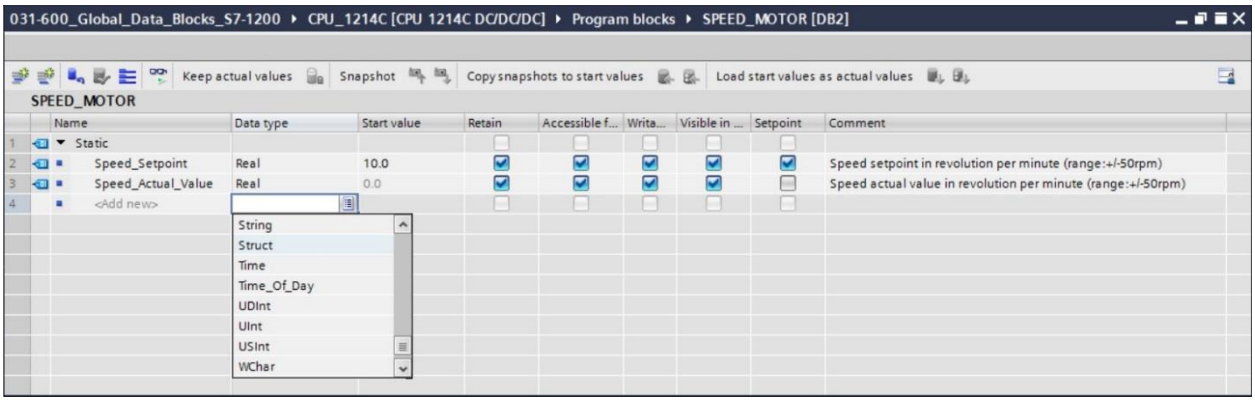
® The "SPEED_MOTOR" data block is automatically displayed. Start by creating the 'Speed_Setpoint' and 'Speed_Actual_Value' tags shown here with their associated comments. Select 'Real' as the data type. Also set a start value of 10.0 rpm for the 'Speed_Setpoint'.

(® Speed_Setpoint ® Real ® 10.0 ® Speed_Actual_Value ® Real)



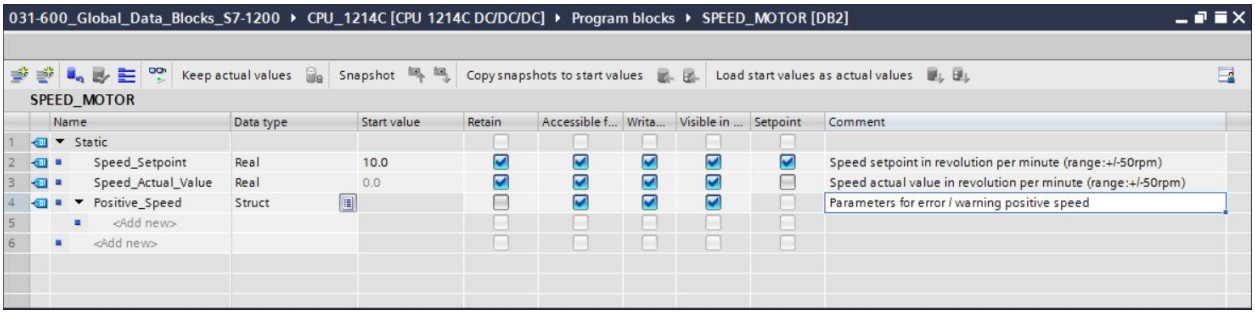
Note: *Be sure to use the correct data types.*

® Next we create a tag structure 'Struct' so it can be duplicated later. (® Struct)

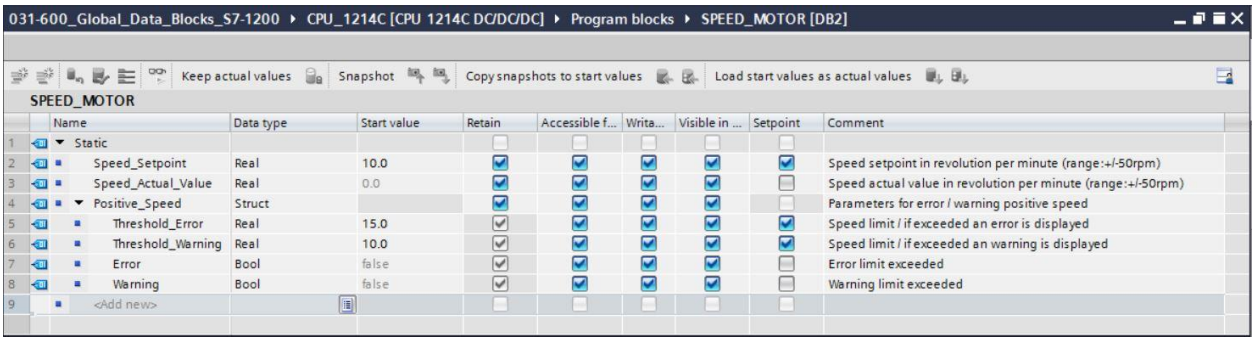


® Name the structure 'Positive_Speed' and enter a comment.

(® Positive_Speed)



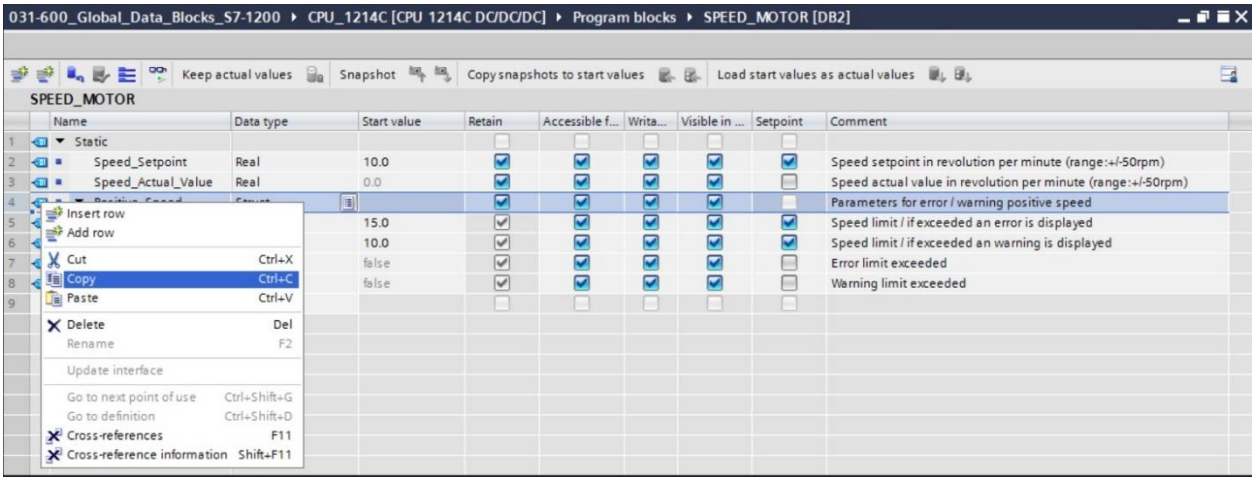
® Create the tags for the speed monitoring with the corresponding start values below the structure as shown here.



Note: *Be sure to use the correct data types.*

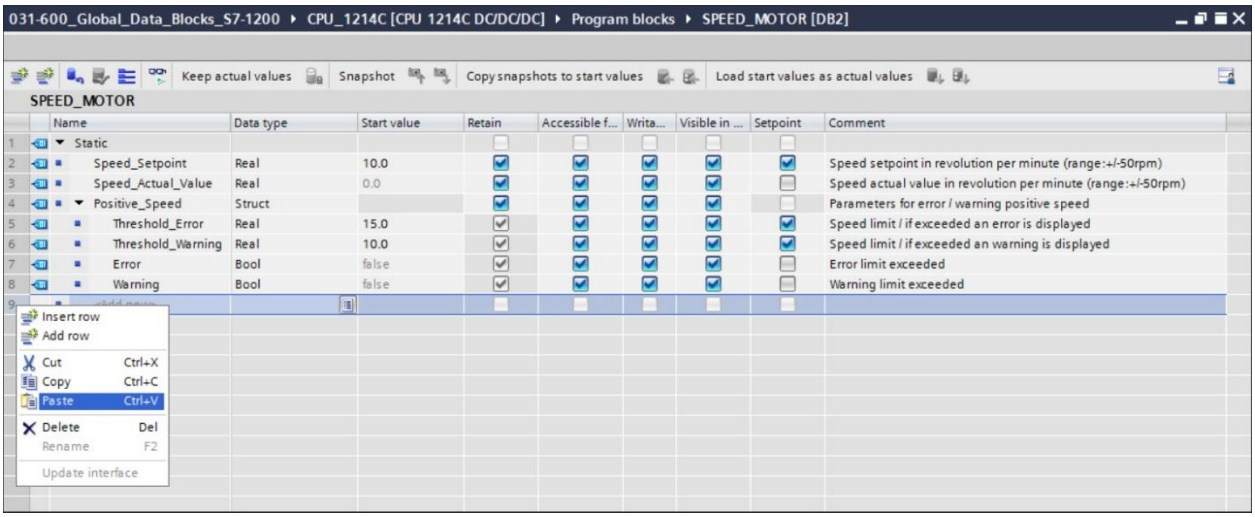
® Then select the structure and copy it.

(® Copy)



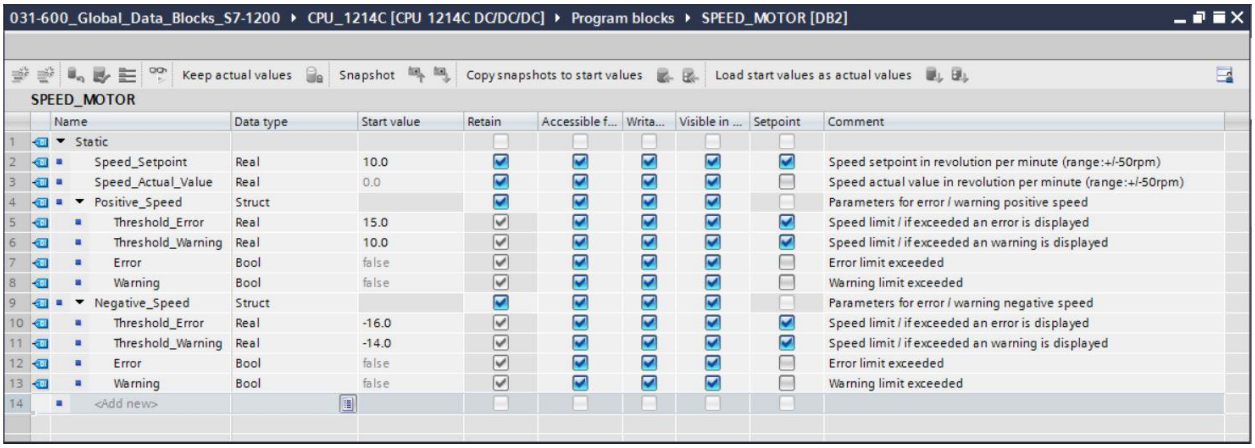
® Paste the copied structure below the 'Positive_Speed' structure again.

(® Paste)



® Rename the new structure to 'Negative_Speed' and enter a comment.

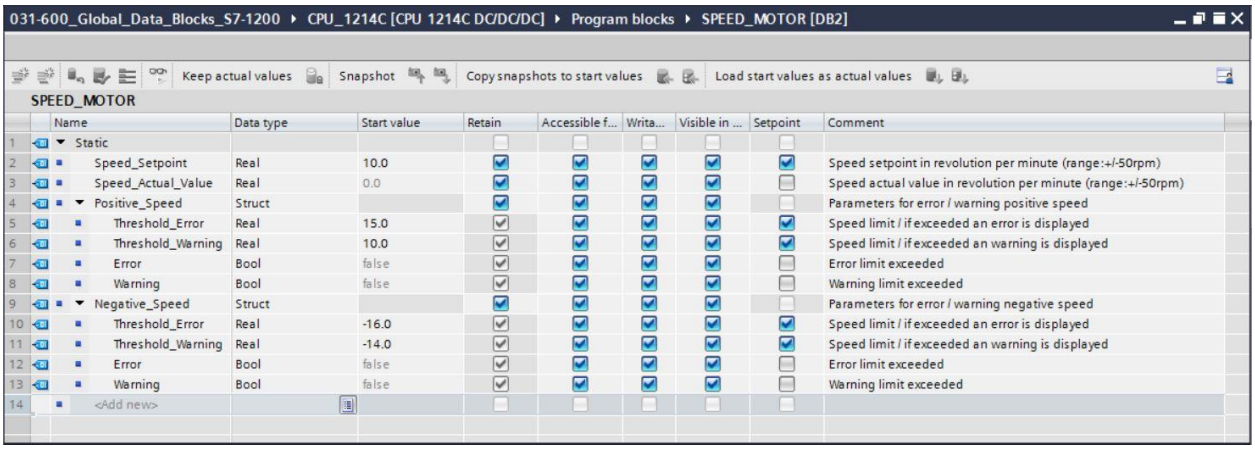
(® Negative_Speed)



	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Static								
2	Speed_Setpoint	Real	10.0	☑	☑	☑	☑	☑	Speed setpoint in revolution per minute (range:+/-50rpm)
3	Speed_Actual_Value	Real	0.0	☑	☑	☑	☑	☐	Speed actual value in revolution per minute (range:+/-50rpm)
4	Positive_Speed	Struct		☑	☑	☑	☑	☐	Parameters for error / warning positive speed
5	Threshold_Error	Real	15.0	☑	☑	☑	☑	☑	Speed limit / if exceeded an error is displayed
6	Threshold_Warning	Real	10.0	☑	☑	☑	☑	☑	Speed limit / if exceeded an warning is displayed
7	Error	Bool	false	☑	☑	☑	☑	☐	Error limit exceeded
8	Warning	Bool	false	☑	☑	☑	☑	☐	Warning limit exceeded
9	Negative_Speed	Struct		☑	☑	☑	☑	☐	Parameters for error / warning negative speed
10	Threshold_Error	Real	-16.0	☑	☑	☑	☑	☑	Speed limit / if exceeded an error is displayed
11	Threshold_Warning	Real	-14.0	☑	☑	☑	☑	☑	Speed limit / if exceeded an warning is displayed
12	Error	Bool	false	☑	☑	☑	☑	☐	Error limit exceeded
13	Warning	Bool	false	☑	☑	☑	☑	☐	Warning limit exceeded
14	<Add new>			☐	☐	☐	☐	☐	

® Do not forget to click Save project. The finished global data block "SPEED_MOTOR" [DB2] is shown below. Check to verify that Retain is selected ☑ and the corresponding start value is entered for all tags. The data will thus be retained in the data block even after a power failure or a STOP/START of the CPU. The check boxes for 'Accessible from HMI' ☑ and 'Visible in HMI' ☑ should also all have a check mark so that all tags in future expansions of this project will be accessible by the visualization systems (HMI). We will select the 'Setpoint' check box ☑ only for the default values in our data block.

(® ☑☑☑☑)

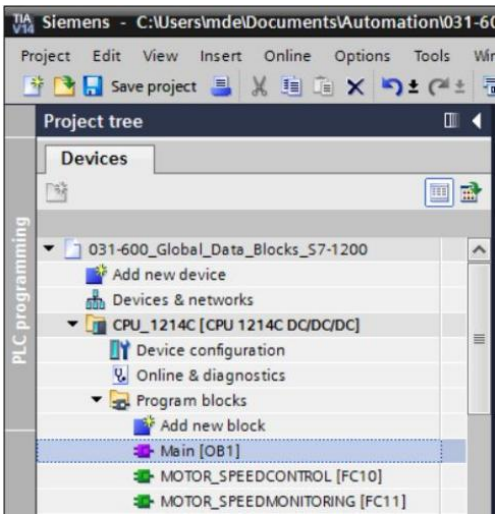


	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Comment
1	Static								
2	Speed_Setpoint	Real	10.0	☑	☑	☑	☑	☑	Speed setpoint in revolution per minute (range:+/-50rpm)
3	Speed_Actual_Value	Real	0.0	☑	☑	☑	☑	☐	Speed actual value in revolution per minute (range:+/-50rpm)
4	Positive_Speed	Struct		☑	☑	☑	☑	☐	Parameters for error / warning positive speed
5	Threshold_Error	Real	15.0	☑	☑	☑	☑	☑	Speed limit / if exceeded an error is displayed
6	Threshold_Warning	Real	10.0	☑	☑	☑	☑	☑	Speed limit / if exceeded an warning is displayed
7	Error	Bool	false	☑	☑	☑	☑	☐	Error limit exceeded
8	Warning	Bool	false	☑	☑	☑	☑	☐	Warning limit exceeded
9	Negative_Speed	Struct		☑	☑	☑	☑	☐	Parameters for error / warning negative speed
10	Threshold_Error	Real	-16.0	☑	☑	☑	☑	☑	Speed limit / if exceeded an error is displayed
11	Threshold_Warning	Real	-14.0	☑	☑	☑	☑	☑	Speed limit / if exceeded an warning is displayed
12	Error	Bool	false	☑	☑	☑	☑	☐	Error limit exceeded
13	Warning	Bool	false	☑	☑	☑	☑	☐	Warning limit exceeded
14	<Add new>			☐	☐	☐	☐	☐	

Note: *The use of setpoints is described further below in the step-by-step instructions.*

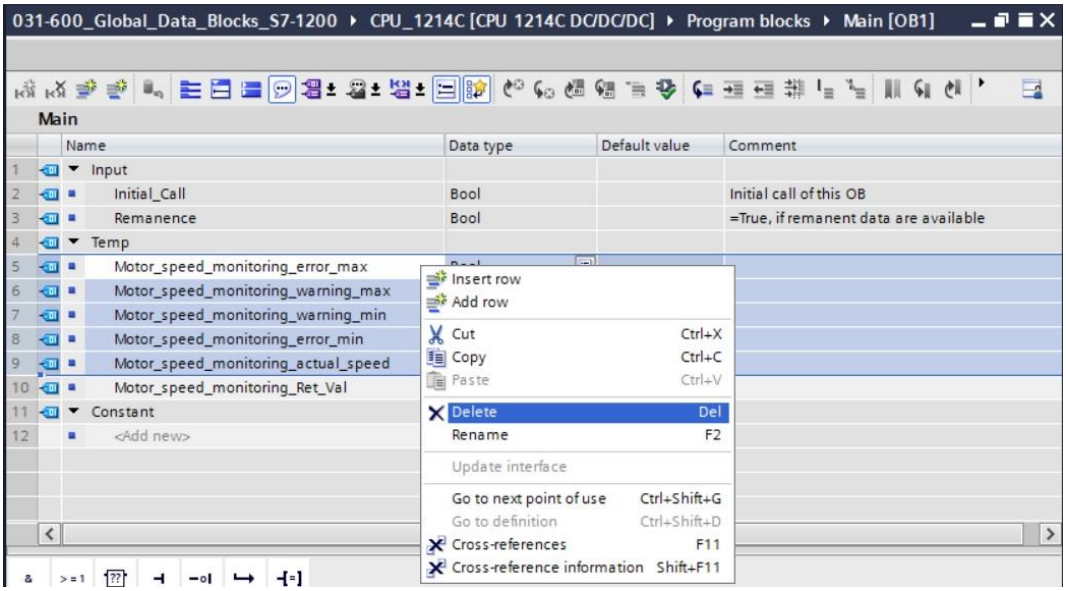
7.3 Access to data of the data block in the organization block

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

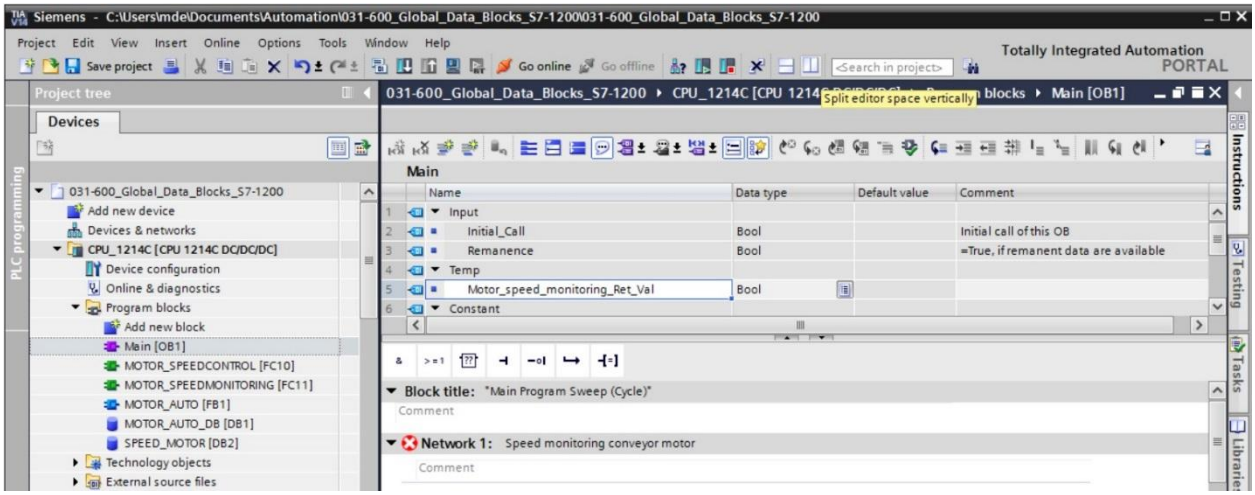


® Delete the temporary tags in “Main” [OB1] that are no longer needed. Only the Boolean tag 'Motor_Speed_Control_Ret_Val' is still needed.

(® Delete)

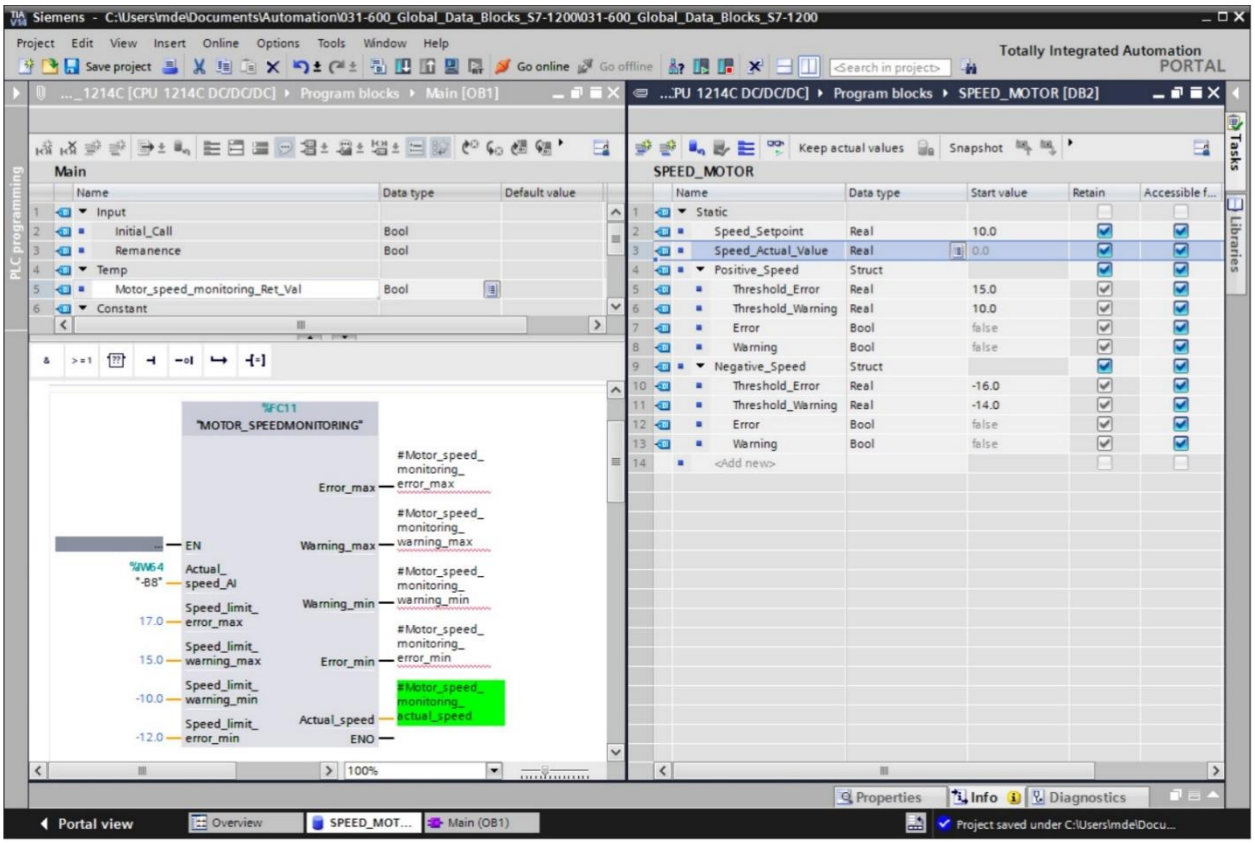


® Have the "SPEED_MOTOR" [DB2] data block and the "Main" [OB1] organization block displayed side by side by clicking the '' icon to vertically split the editor area. (®)

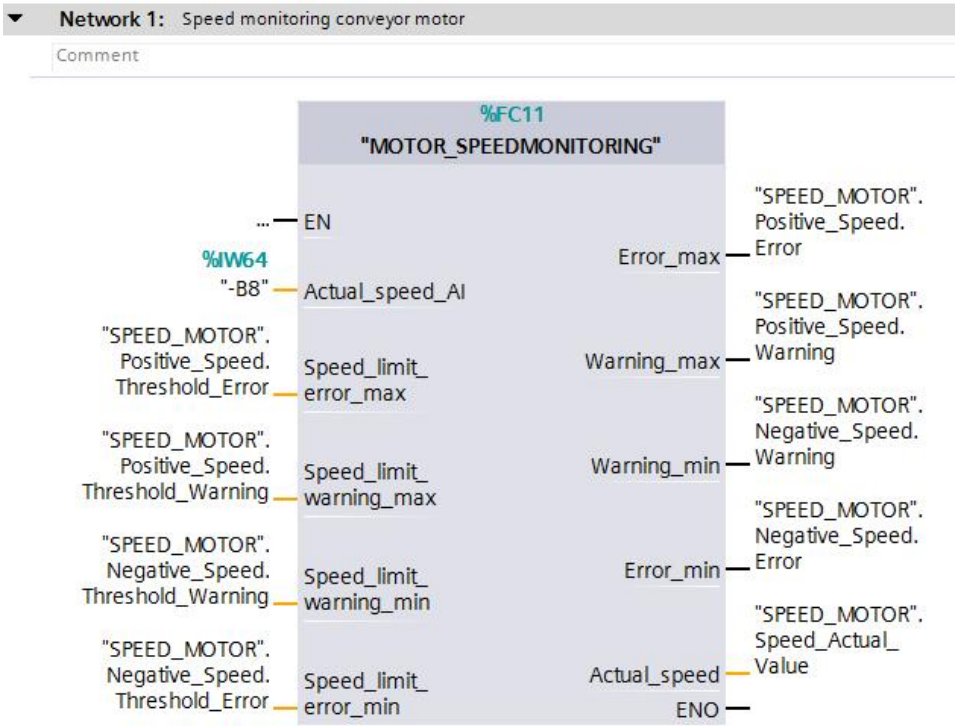


® Use drag & drop to move the tags needed for the interconnection from the "SPEED_MOTOR" [DB2] data block onto the connections of the called functions and function blocks in the "Main" [OB1] organization block. First we move the 'Speed_Actual_Value' tag onto the 'Actual_speed' output of the "MOTOR_SPEEDMONITORING" [FC11] block.

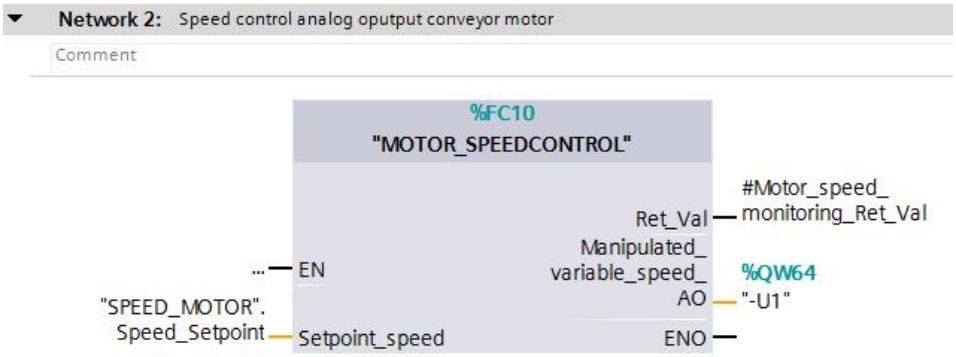
(® Speed_Actual_Value)



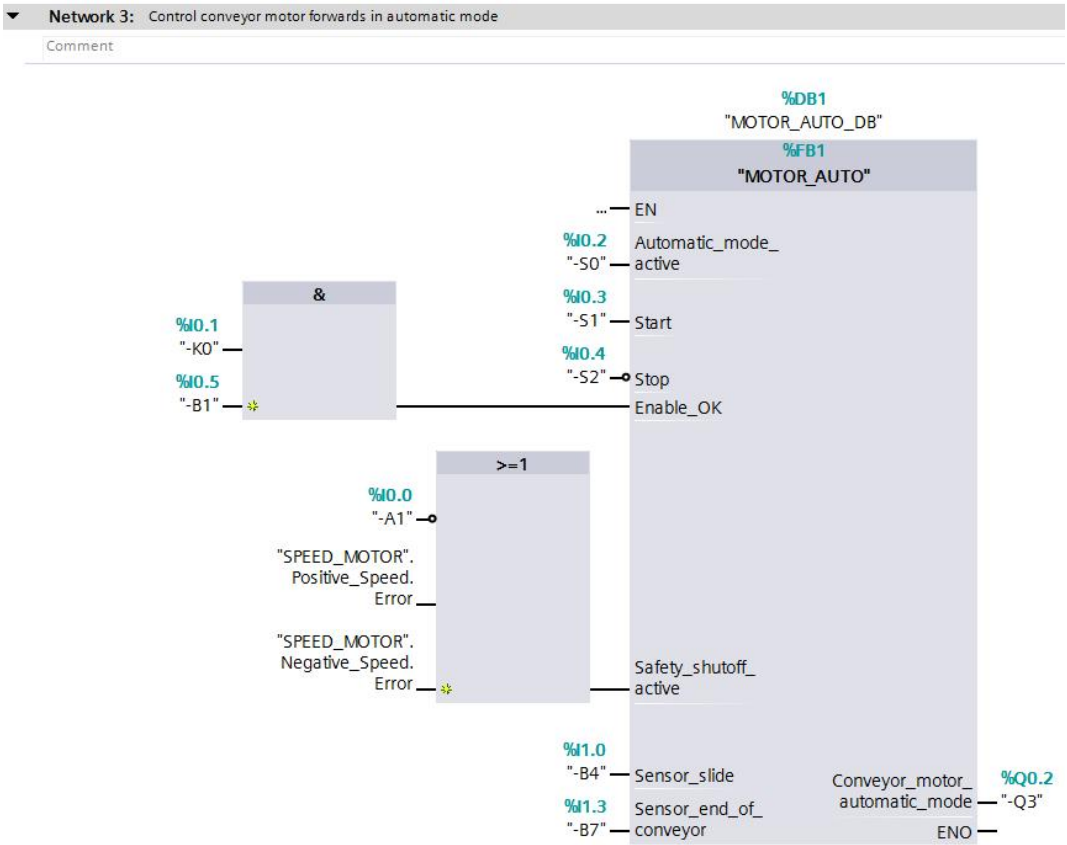
® Also connect the other contacts in Network 1 with tags from the "SPEED_MOTOR" [DB2] data block as shown here.



® Connect the contacts in Network 2 with tags from the "SPEED_MOTOR" [DB2] data block as shown here.

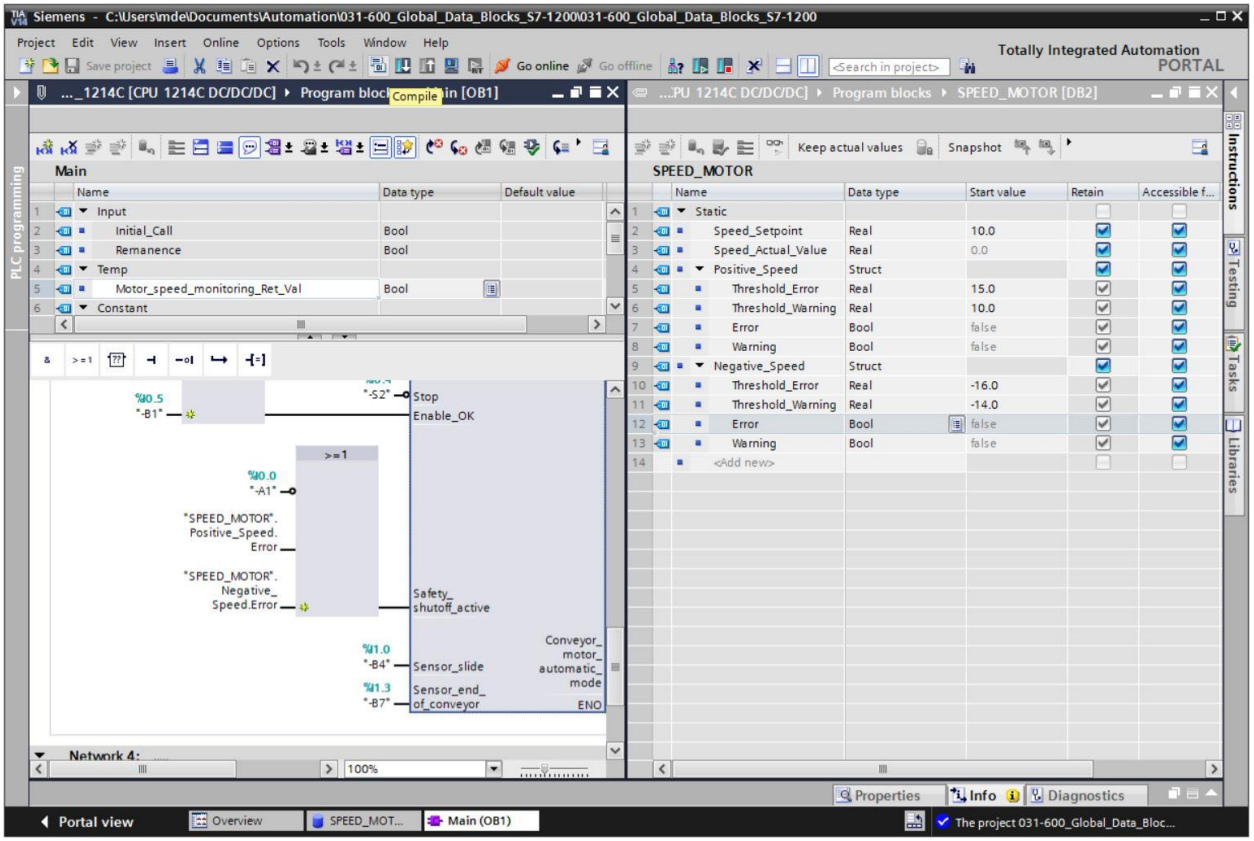


® Connect the contacts in Network 3 with tags from the "SPEED_MOTOR" [DB2] data block as shown here.

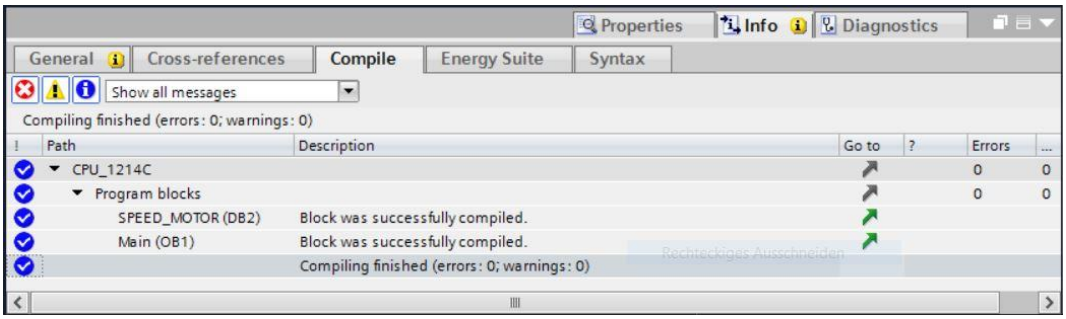


7.4 Save and compile the program

® To save your project, click 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 ®)

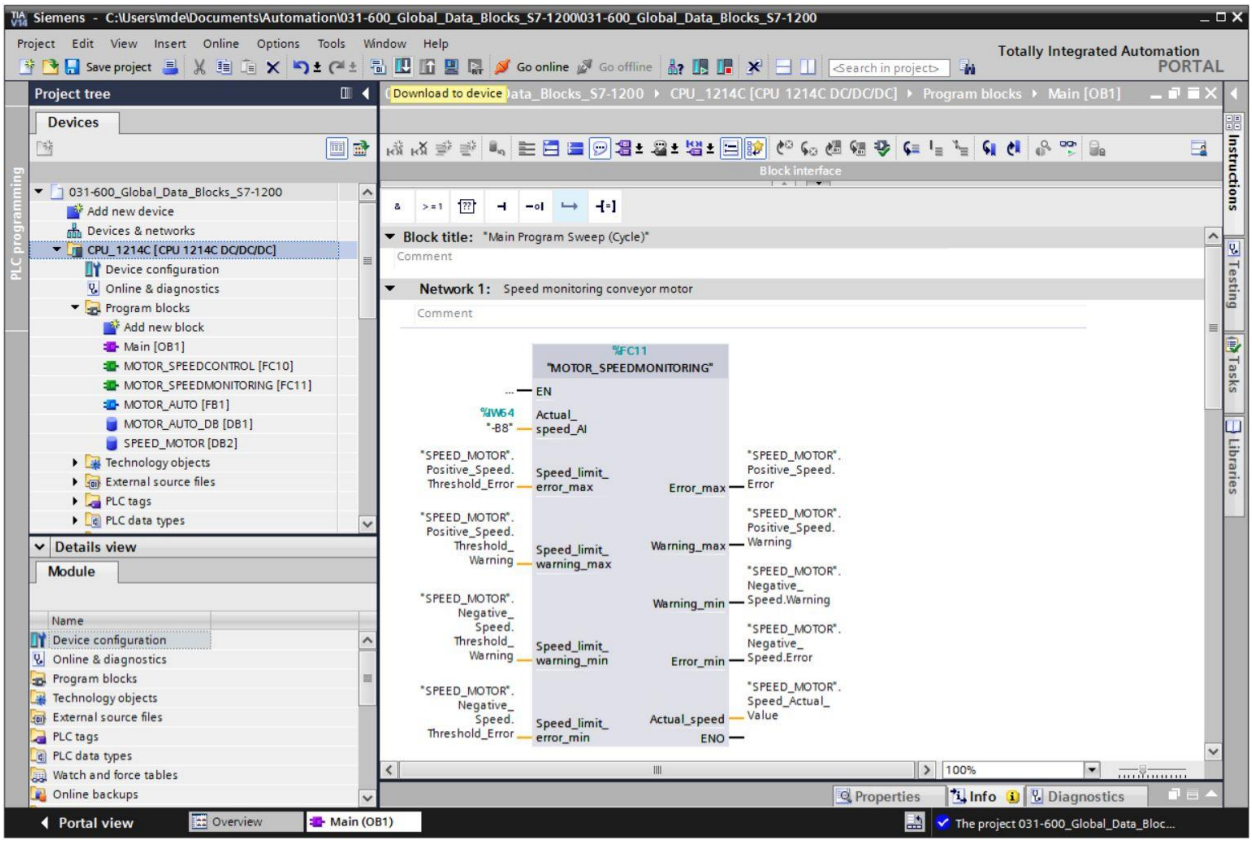


® The "Info", "Compile" area shows which blocks were successfully compiled.



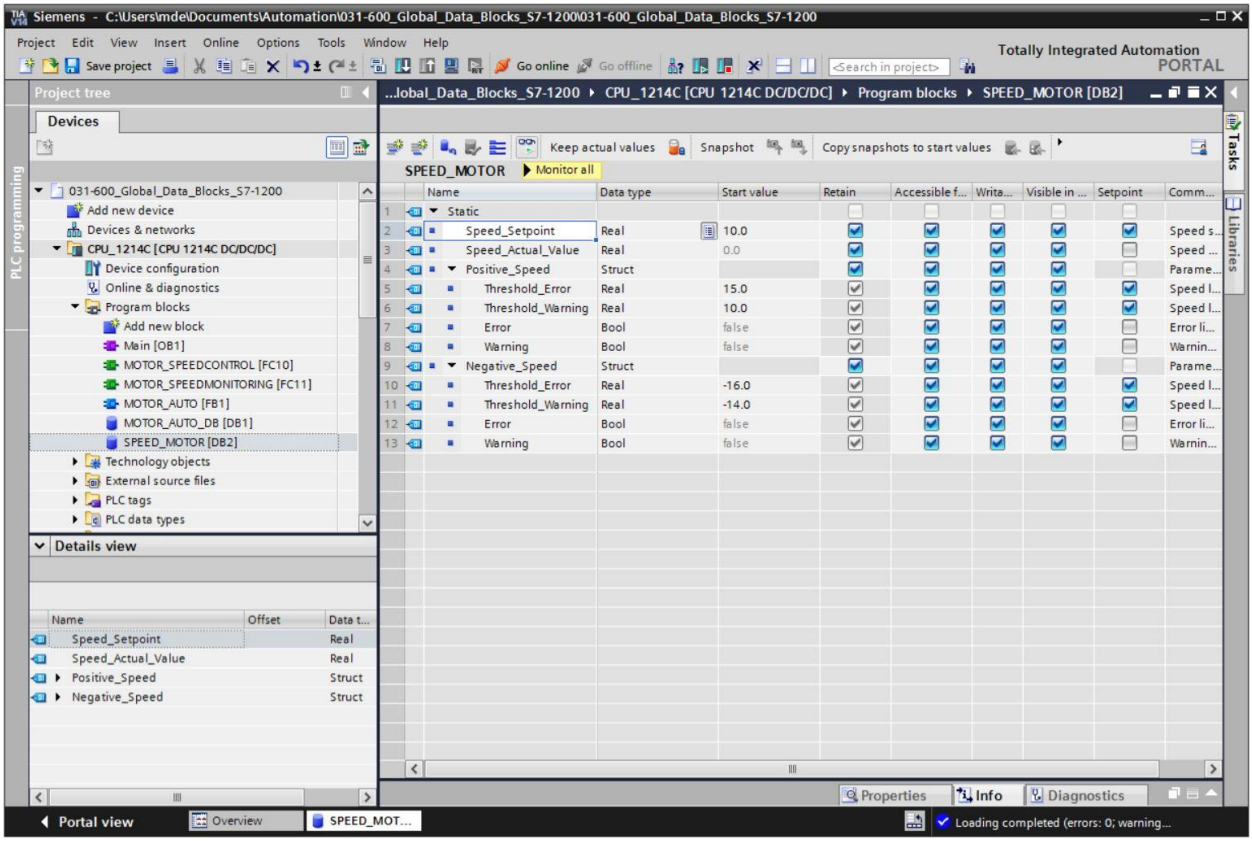
7.5 Download the program

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

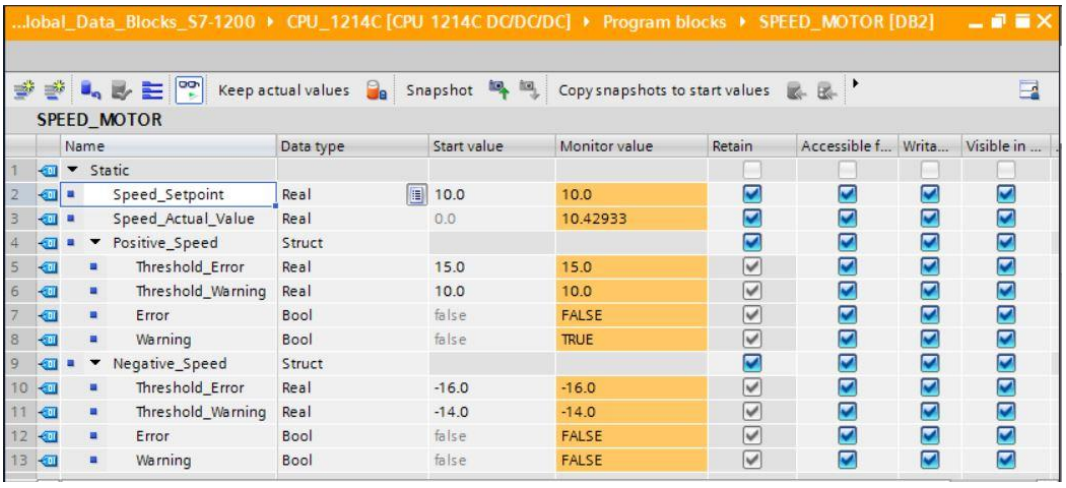


7.6 Monitor/modify values in data blocks

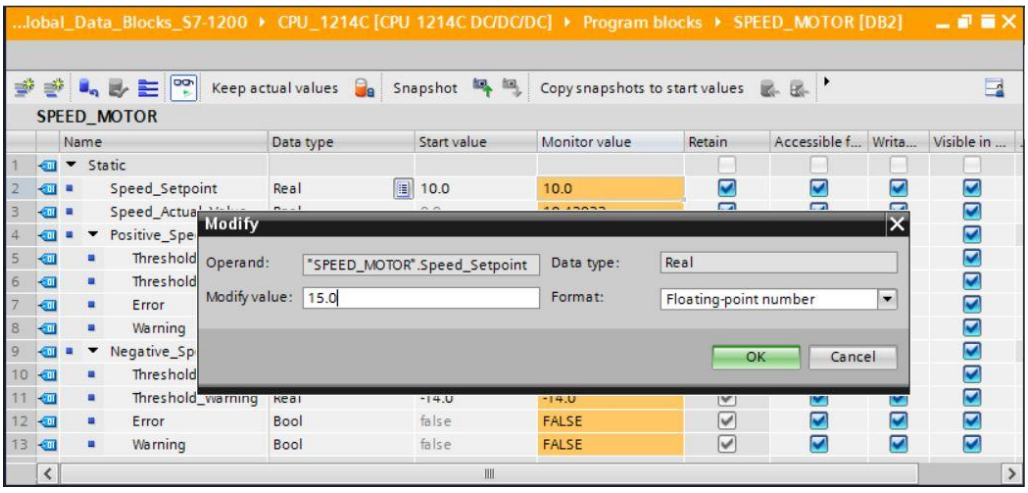
® The desired block must be open for monitoring the tags of a downloaded data block. The monitoring can then be activated/deactivated by clicking the  icon. (® SPEED_MOTOR [DB2] ® )



® In the 'Monitor value' column, the values currently available in the CPU can be monitored.

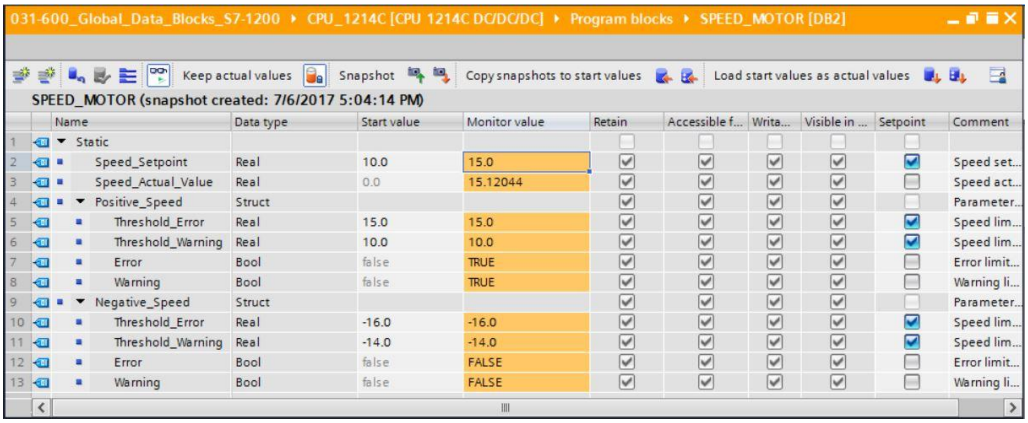


® If you right-click on one of the values, the 'Modify' dialog for modifying this value opens (® Modify ® Modify value: 15.0 ® OK)



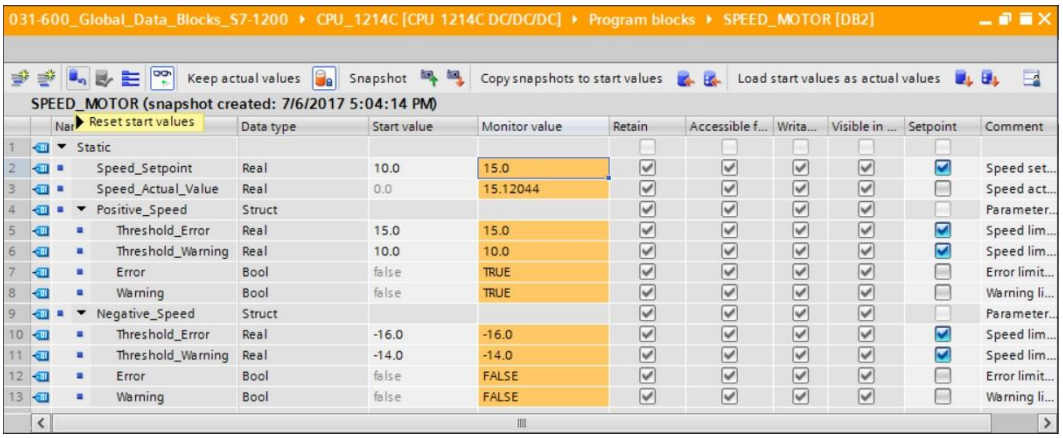
7.7 Initialize setpoints / reset start values

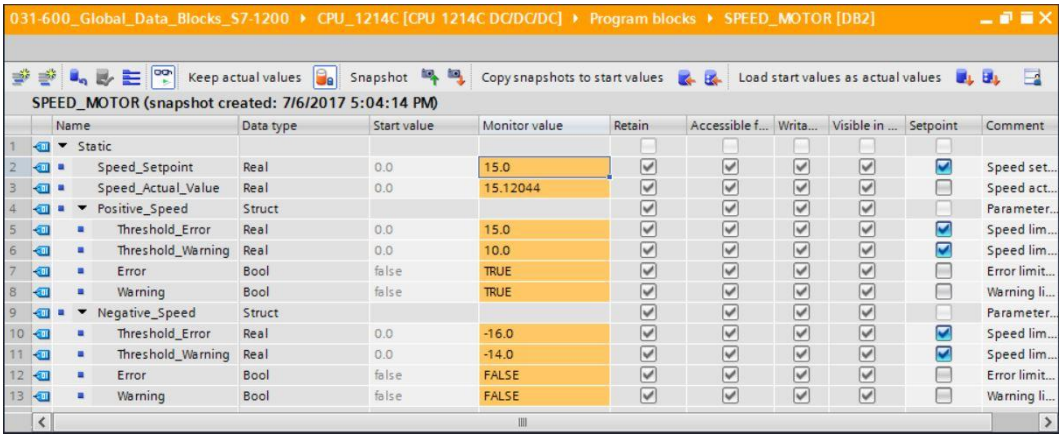
® The setpoints can be initialized by clicking the '  ' icon. For the tags whose 'Setpoint' check box is selected  , the start value will then be applied as the current value. (® )



® All start values can be reset by clicking the icon.

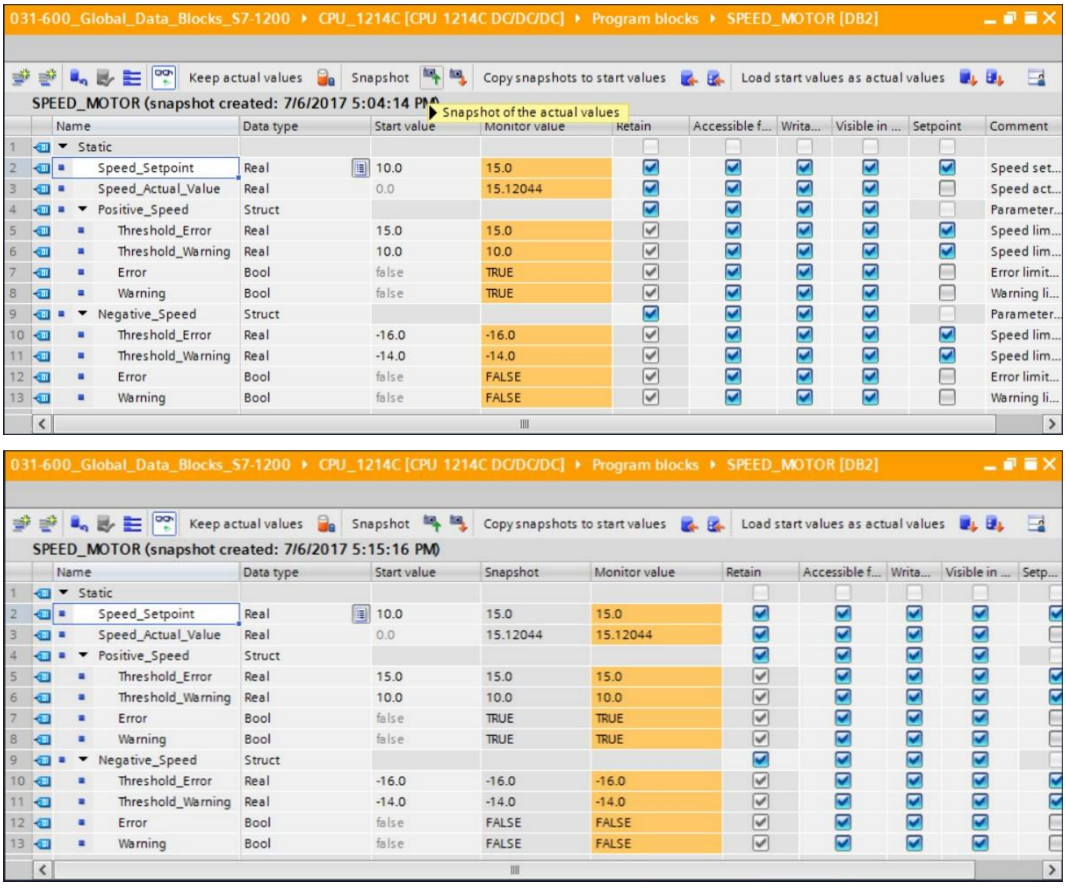
(®)





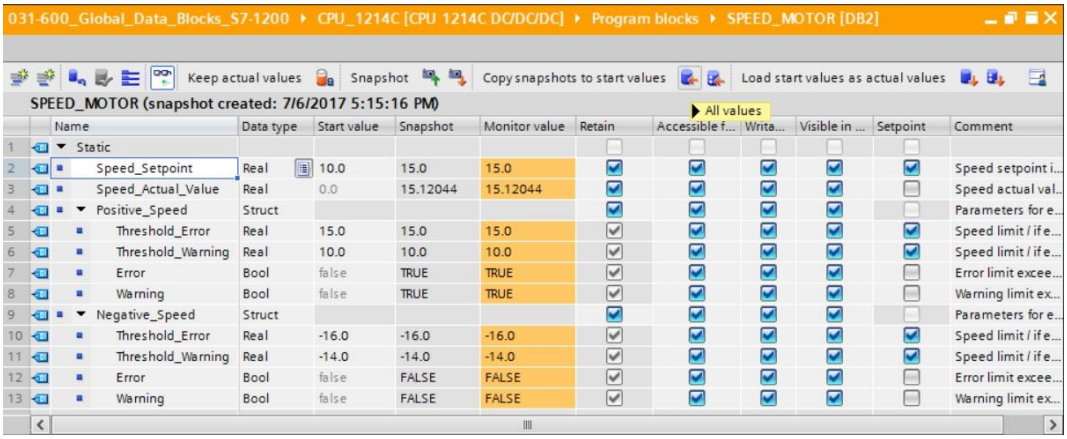
7.8 Snapshots in data blocks

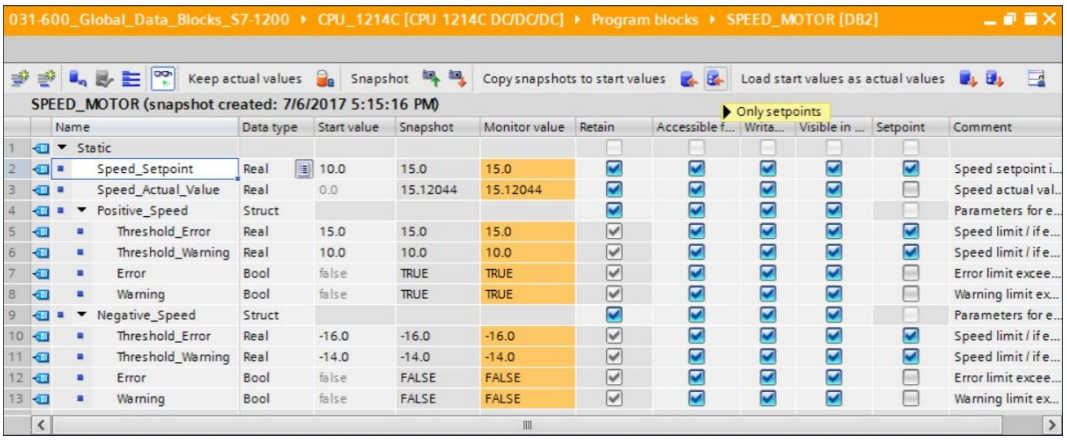
® If you click the ' ' icon, a snapshot of the actual values can be taken in order to apply these values as start values or to transfer them back to the CPU later by clicking the icon ' ' (® ®).



® Alternatively, values from the snapshot can be copied to the start values by clicking the ' icon for all values or by clicking the ' ' icon for the setpoints only. Only the setpoints are needed here in most cases.

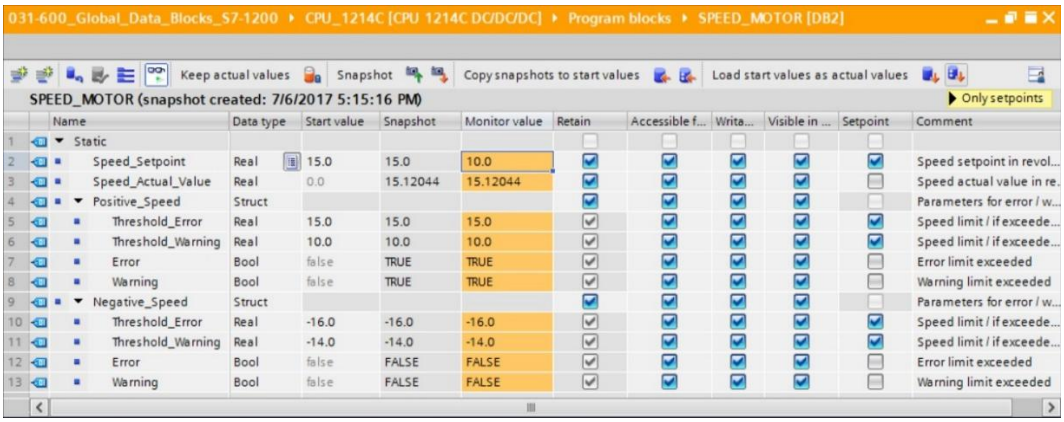
(®)

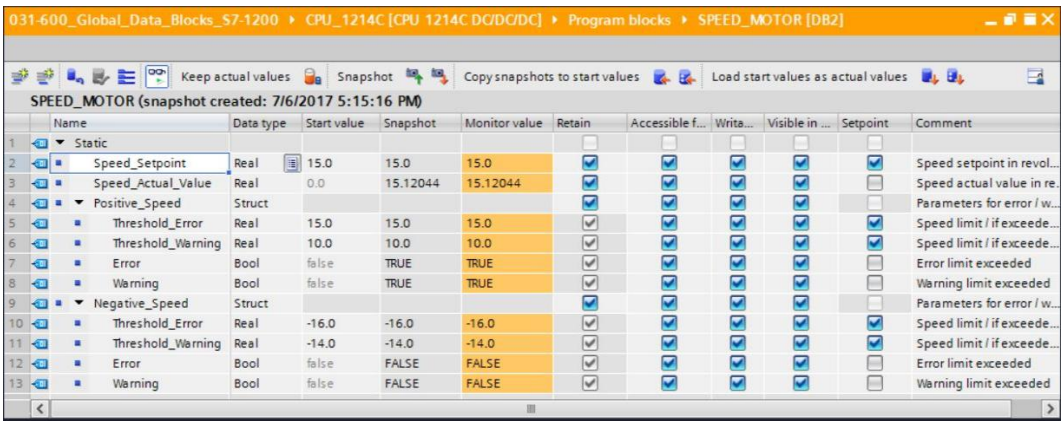




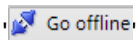
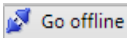
® If you want to load the start values back into the actual values there are two possibilities. Alternatively all start values can be copied to the actual values by clicking the '  ' icon or only the setpoints by clicking the '  ' icon.

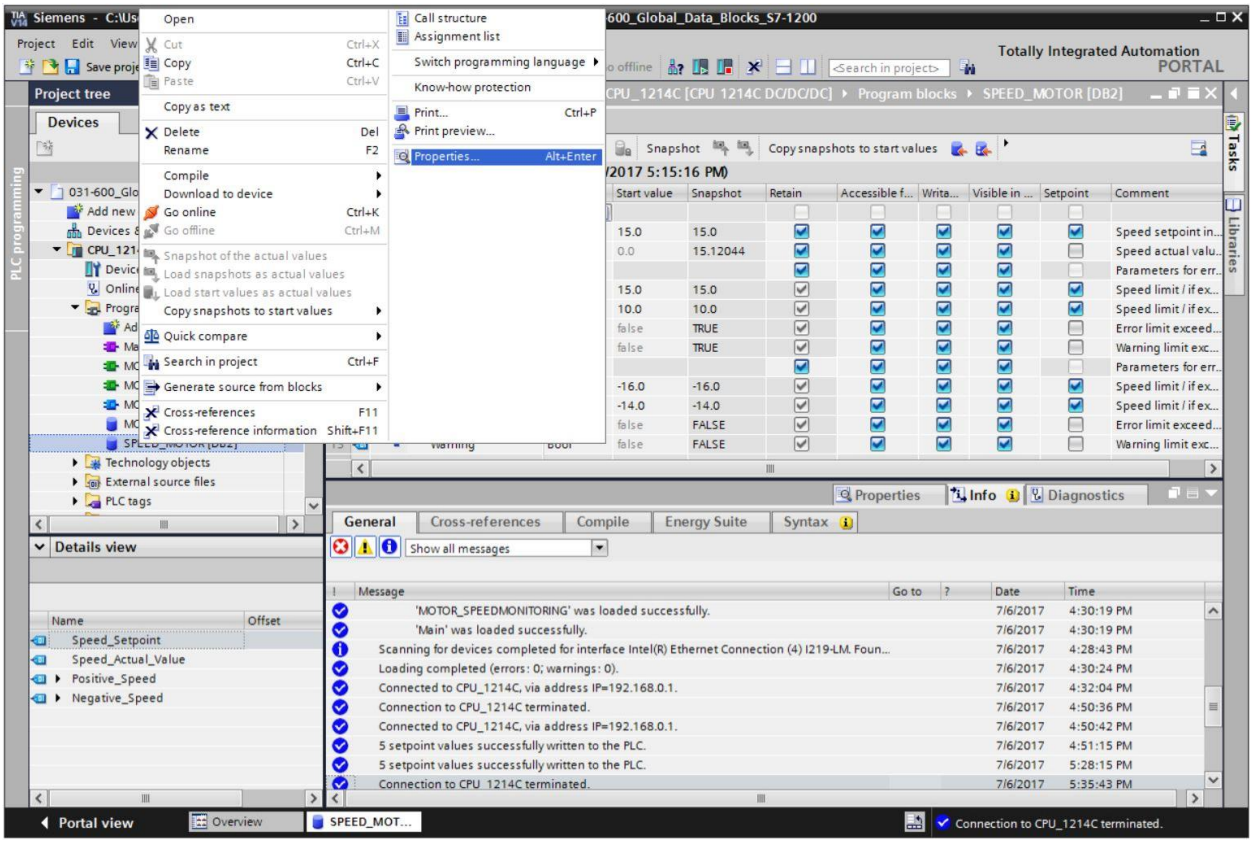
(® )





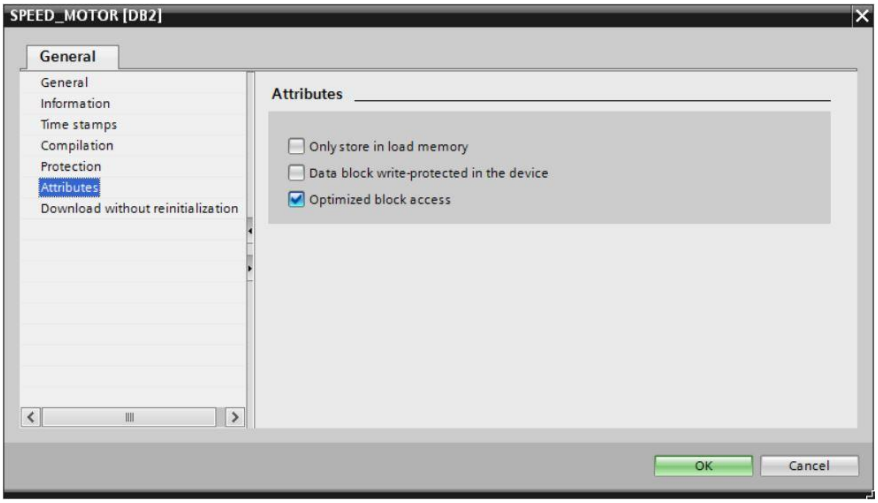
7.9 Expand data block and download it without reinitialization

® To enable 'Download without reinitialization' for the "SPEED_MOTOR" [DB2] data block, you must go offline  and then open the properties of the data block. (®  ® SPEED_MOTOR [DB2] ® Properties)

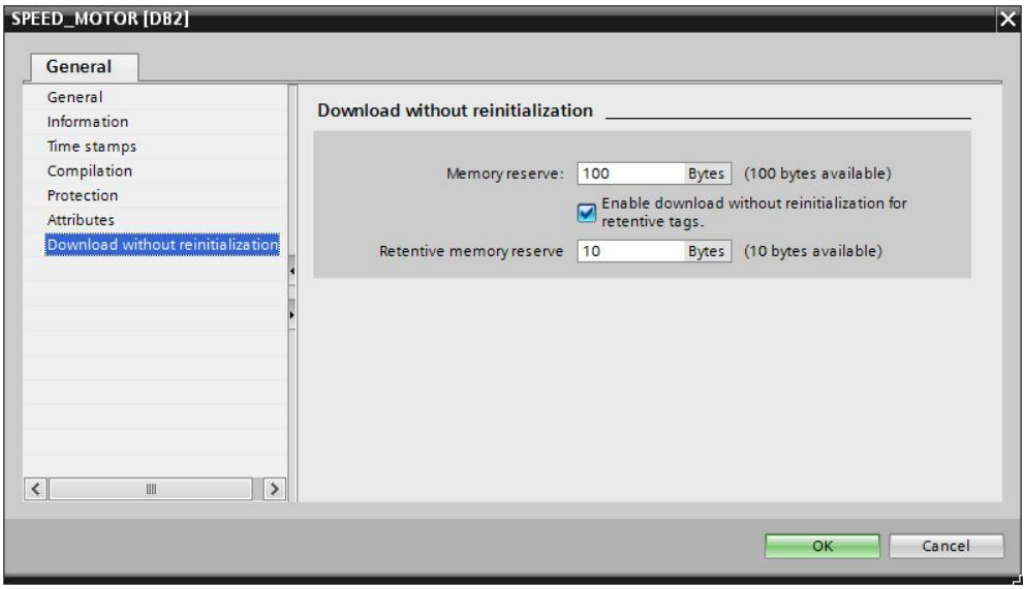


® Select the 'Optimized block access' check box ☑ in the properties under 'General', 'Attributes'.

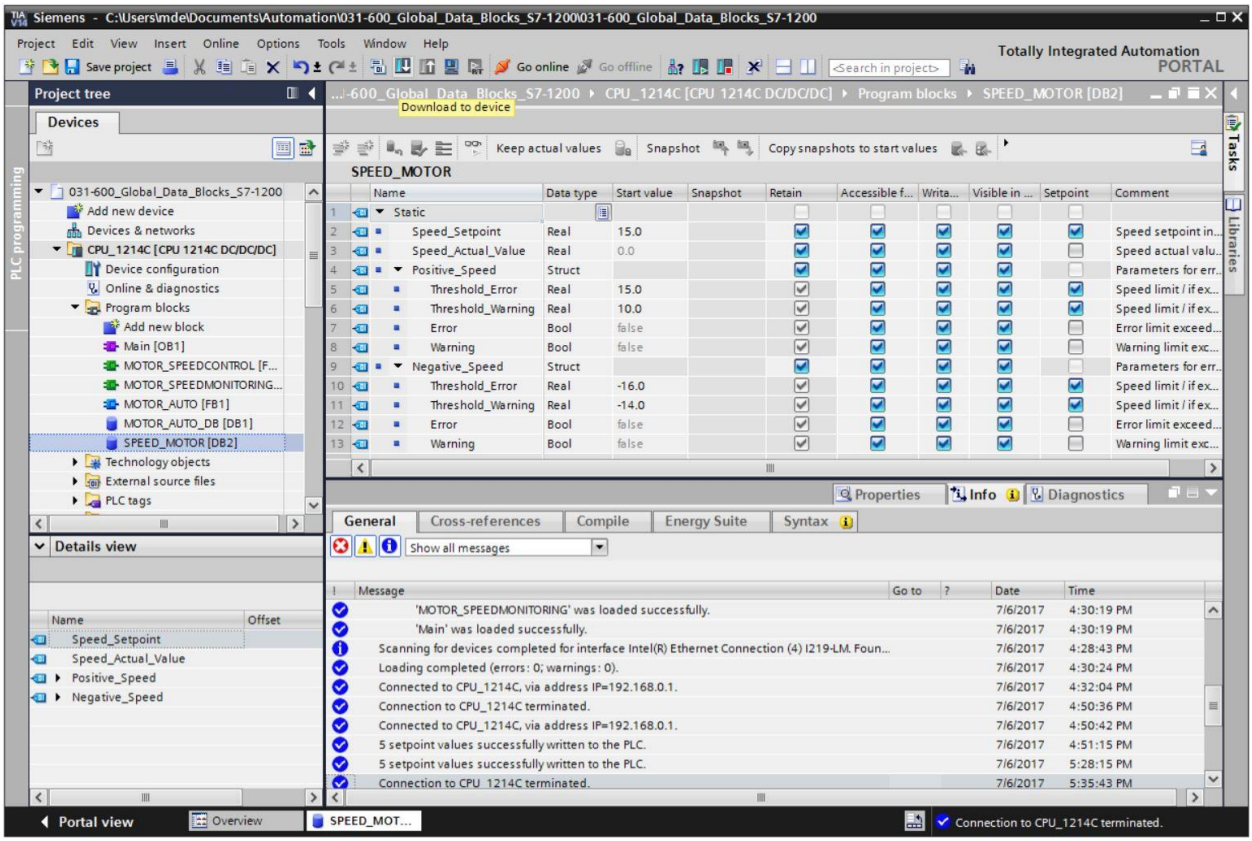
(® General ® Attributes ® ☑ Optimized block access)



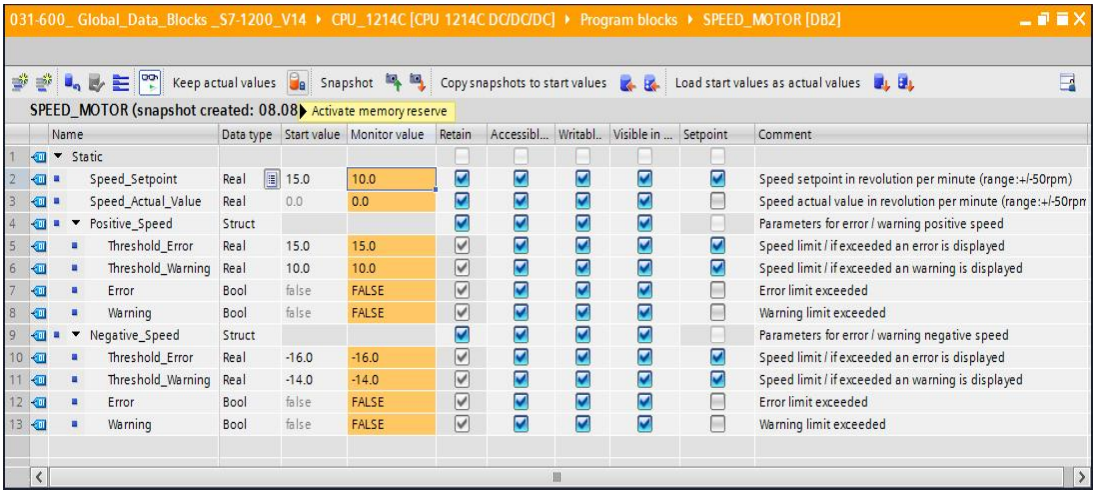
® Assign a 'Retentive memory reserve' to the data block for 'Download without reinitialization'.
(® Download without reinitialization ® Retentive memory reserve ® 10 bytes ® OK)

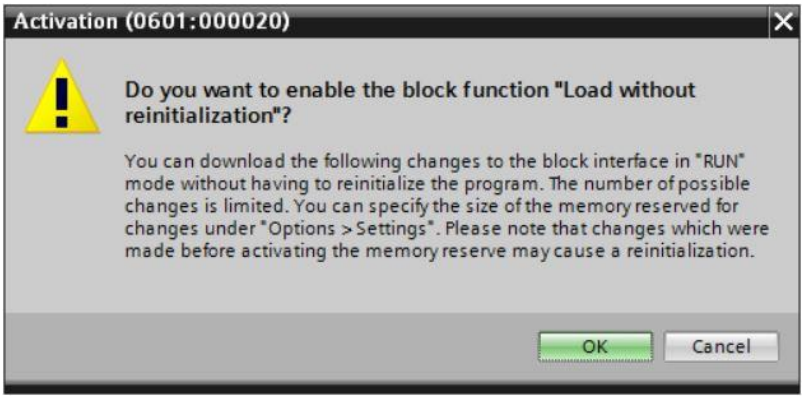


® Download your "SPEED_MOTOR" [DB] data block to the controller again and select 'Go online'.
(® SPEED_MOTOR [DB] ® ® Go online)



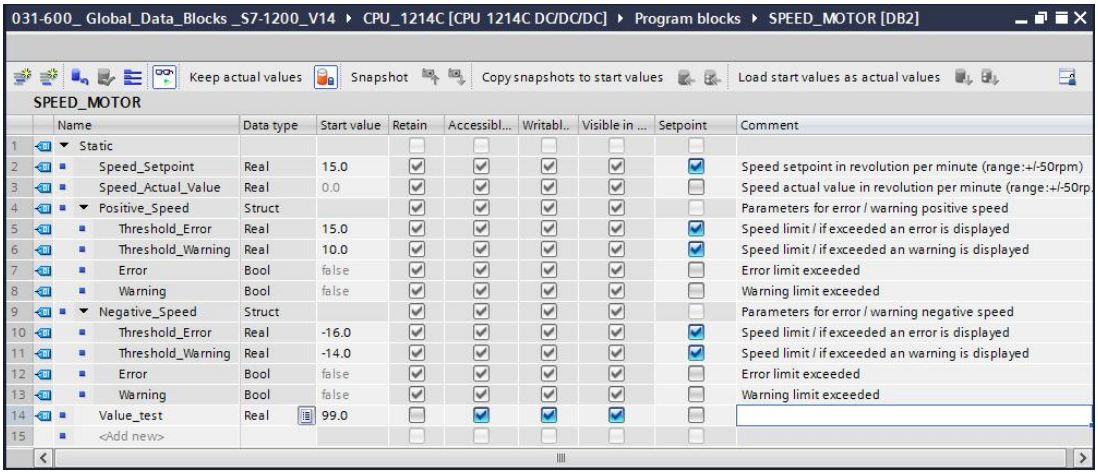
® Then click the '  ' icon to activate memory reserve and thus activate downloading without reinitialization for keeping actual values. Confirm the safety prompt with 'OK'. (®  ® OK)





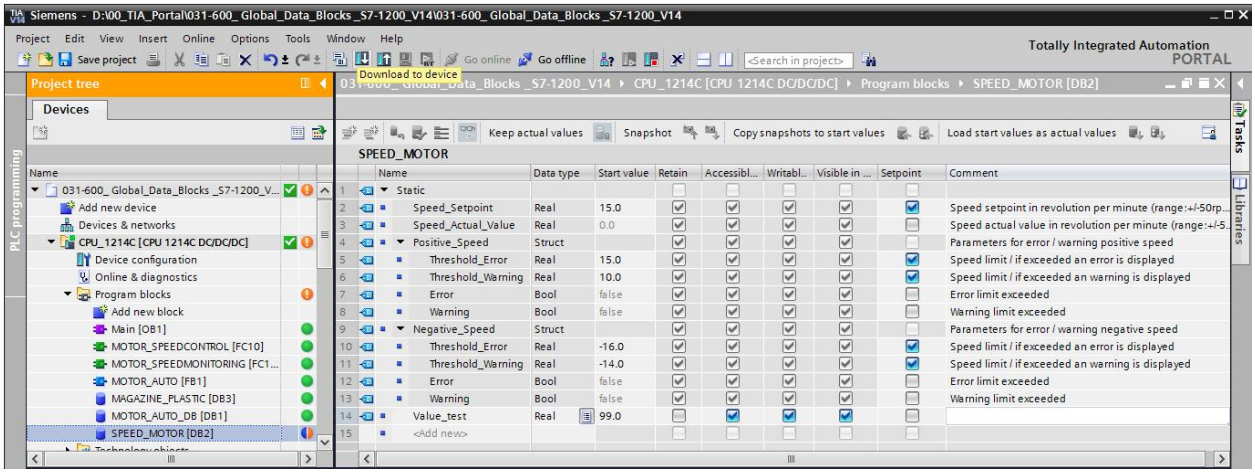
® Next add any tag in your data block 99.0

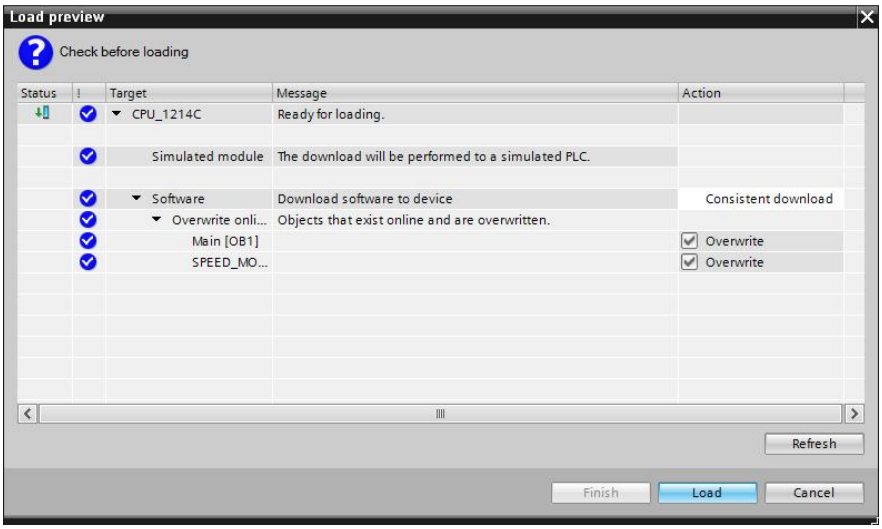
(® Name: Value_test ® Data type: Real ® Start value: 99.0)



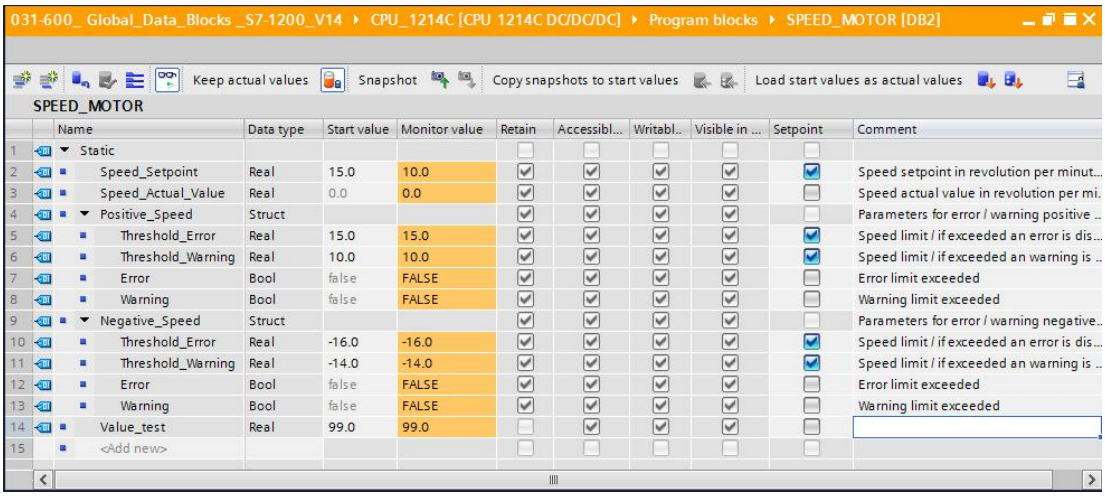
® Download your "SPEED_MOTOR" [DB] data block to the controller again.

® SPEED_MOTOR [DB] ®  ® Download)





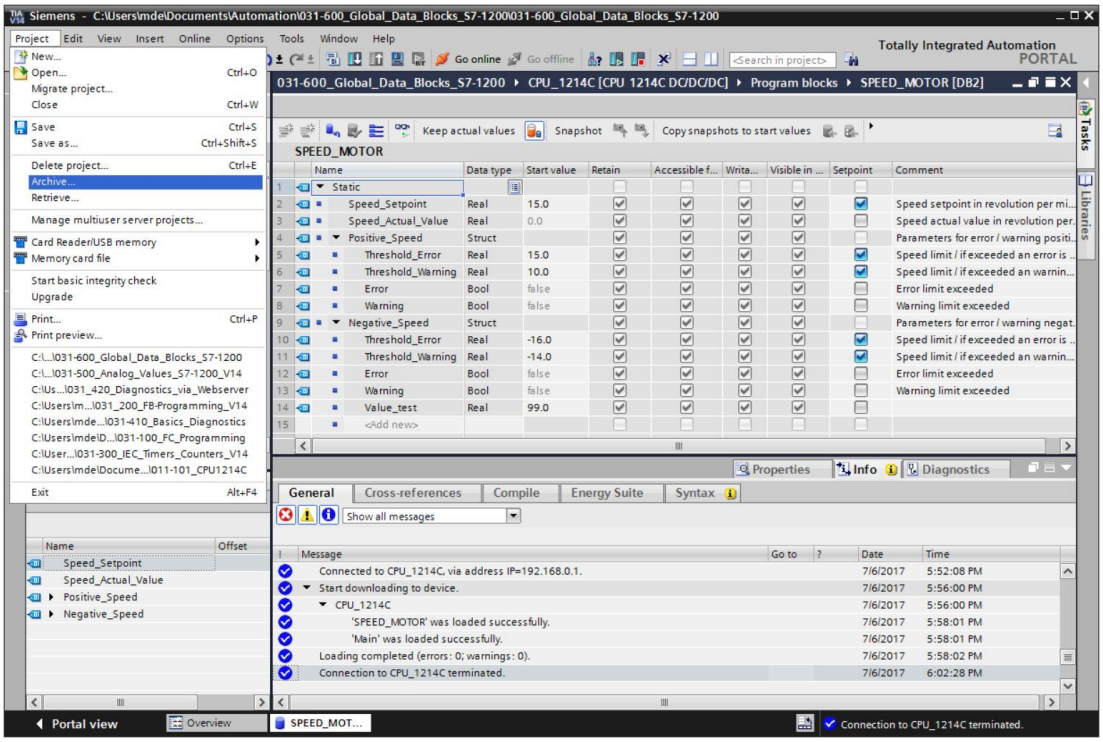
® If you click '  ' to monitor the block again, you will see that the monitored values for the previously existing tags have not been overwritten with the start values. (® )



7.10 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 ® 031-600_Global_Data_Blocks_S7-1200…. ® Save)



8 Checklist

No.	Description	Completed
1	Data block SPEED_MOTOR [DB2] successfully created.	
2	Program changes made in Main [OB1].	
3	Compiling successful and without error message	
4	Download successful and without error message	
5	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 part at slide activated (-B4 = 1) then Conveyor motor M1 variable speed (-Q3 = 1) switches on and stays on. The speed corresponds to the speed setpoint in the range +/- 50 rpm	
6	Sensor part at end of conveyor activated (-B7 = 1) ® -Q3 = 0 (after 2 seconds)	
7	Briefly press the automatic stop pushbutton (-S2 = 0) ® -Q3 = 0	
8	Activate EMERGENCY OFF (-A1 = 0) ® -Q3 = 0	
9	Manual mode (-S0 = 0) ® -Q3 = 0	
10	Switch off station (-K0 = 0) ® -Q3 = 0	
11	Cylinder not retracted (-B1 = 0) ® -Q3 = 0	
12	Speed > Motor_speed_monitoring_error_max ® -Q3 = 0	
13	Speed < Motor_speed_monitoring_error_min ® -Q3 = 0	
14	Project successfully archived	

9 Exercise

9.1 Task – Exercise

In this exercise a global data block "MAGAZINE_PLASTIC" [DB3] will be created additionally.

The setpoint and actual value of the counter for the plastic parts will be specified and displayed in this data block.

A connectable input for the setpoint setting and an output for displaying the actual value will also be added to the "MOTOR_AUTO" [FB1] function block.

9.2 Technology diagram

Here you see the technology diagram for the task.

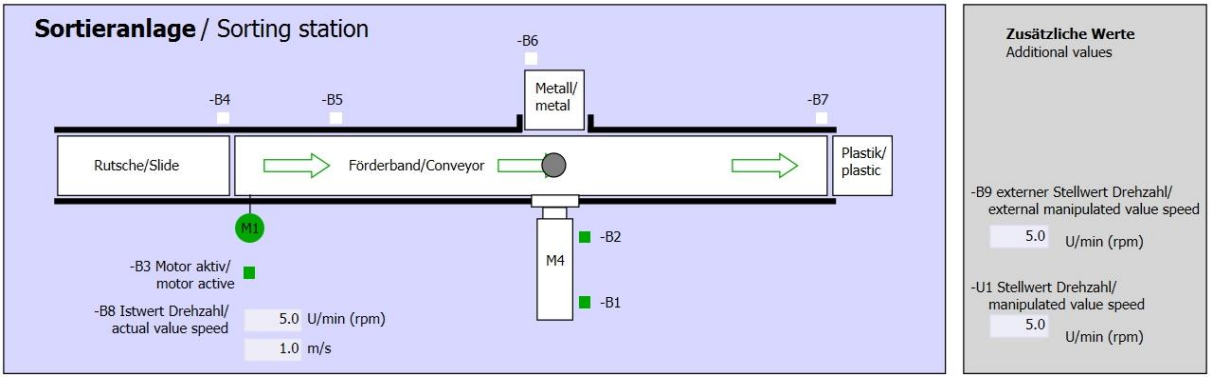


Figure 5: Technology diagram

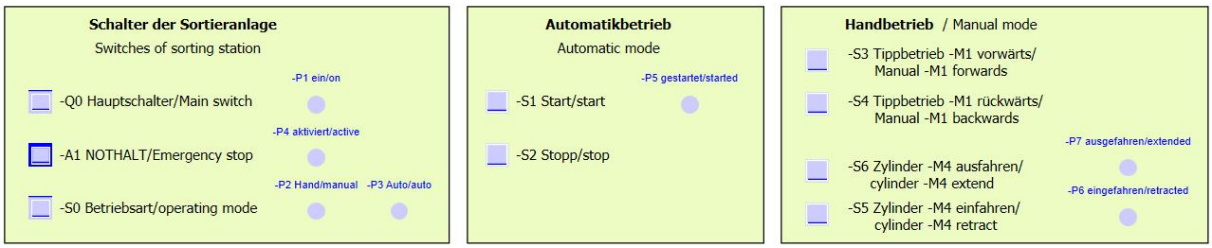


Figure 6: Control panel

9.3 Reference list

The following signals are required 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 part at slide	NO
I 1.3	BOOL	-B7	Sensor part at end of conveyor	NO
IW64	BOOL	-B8	Sensor actual value speed of the motor +/-10V corresponds to +/- 50 rpm	

DO	Type	Identifier	Function	
Q 0.2	BOOL	-Q3	Conveyor motor -M1 variable speed	
QW 64	BOOL	-U1	Manipulated value speed of the motor in 2 directions +/- 10V corresponds to +/- 50 rpm	

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		

9.4 Planning

Plan the implementation of the task on your own.

9.5 Checklist – Exercise

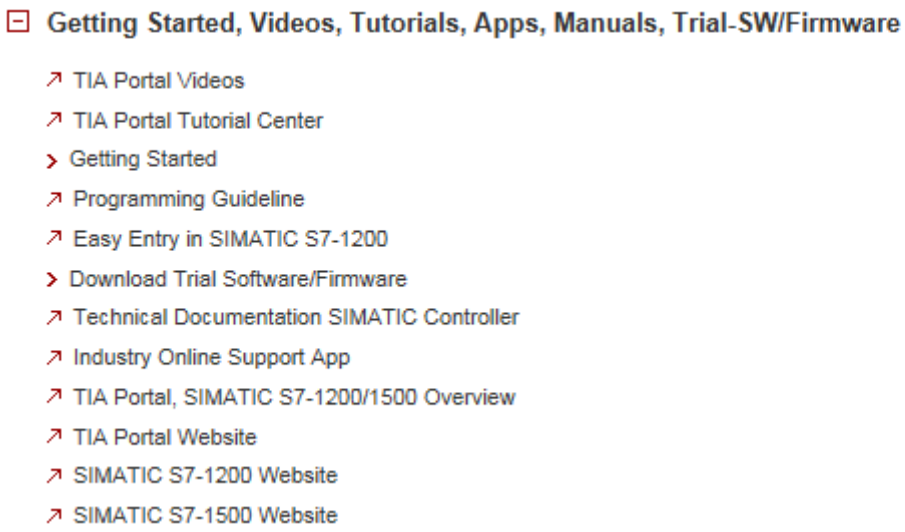
No.	Description	Completed
1	Data block MAGAZINE_PLASTIC [DB3] successfully created.	
2	Program changes made in MOTOR_AUTO [FB1].	
3	Program changes made in Main [OB1].	
4	Compiling successful and without error message	
5	Download successful and without error message	
6	Switch on station (-K0 = 1) Cylinder 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 part at slide activated (-B4 = 1) then Conveyor motor M1 variable speed (-Q3 = 1) switches on and stays on. The speed corresponds to the speed setpoint in the range +/- 50 rpm	
7	Sensor part at end of conveyor activated (-B7 = 1) ® -Q3 = 0 (after 2 seconds)	
8	Briefly press the automatic stop pushbutton (-S2 = 0) ® -Q3 = 0	
9	Activate EMERGENCY OFF (-A1 = 0) ® -Q3 = 0	
10	Manual mode (-S0 = 0) ® -Q3 = 0	
11	Switch off station (-K0 = 0) ® -Q3 = 0	
12	Cylinder not retracted (-B1 = 0) ® -Q3 = 0	
13	Speed > Motor_speed_monitoring_error_max ® -Q3 = 0	
14	Speed < Motor_speed_monitoring_error_min ® -Q3 = 0	
15	Project successfully archived	

10 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