SIEMENS

Learn-/Training Document

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

TIA Portal Module 051-201
High-Level Language Programming
with SCL and 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

This SCE Learning/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 Learning/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

High-Level Language Programming with SCL and S7-1200

1 Objective

In this section, you will become familiar with the basic functions of the SCL high-level language. Test functions for eliminating logical programming errors will also be presented.

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

2 Requirements

This section builds on the hardware configuration of a SIMATIC S7-1200. It can be implemented with any hardware configurations that have digital input and output cards. To implement this section, you can use the following project, for example:

"SCE_EN_011_101_Hardware_Configuration_CPU1214C…..zap14"

You should also be familiar with high-level language programming, such as Pascal.

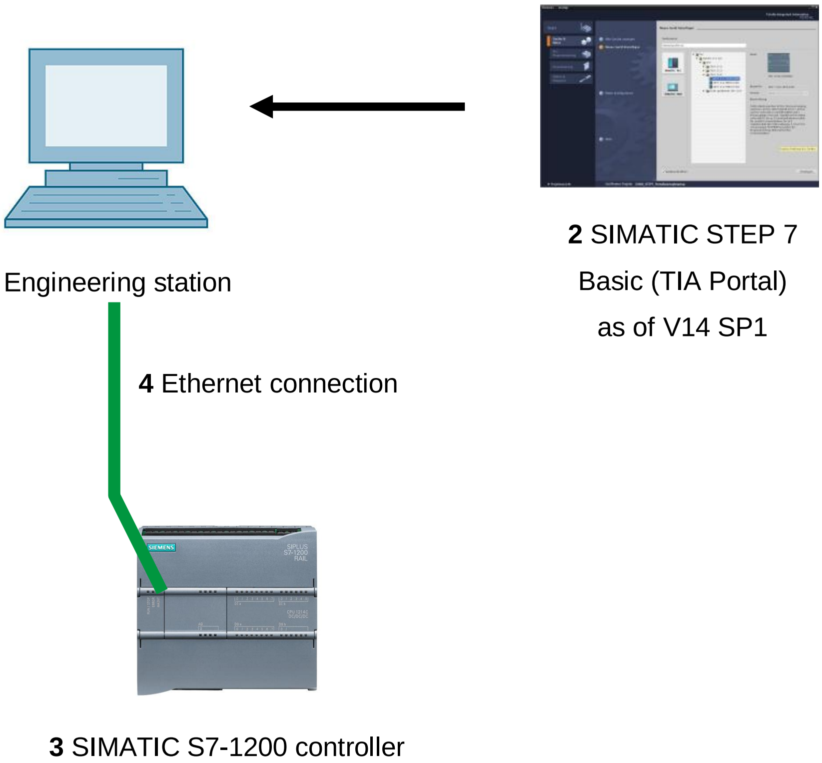
3 Hardware and software required

1 Engineering Station: The requirements are hardware and operating system (for additional information, see Readme on the TIA Portal Installation DVD)

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

3 SIMATIC S7-1200 controller, e.g. CPU 1214C DC/DC/DC – Firmware V4.2.1 or higher

4 Ethernet connection between the engineering station and controller



4 Theory

4.1 SCL programming language

SCL (Structured Control Language) is a high-level, Pascal-based programming language that enables structured programming. The language corresponds to the "Structured Text" (ST) programming language specified in DIN EN-61131-3 (IEC 61131-3). In addition to high-level language elements, SCL contains typical elements of the PLC as language elements such as inputs, outputs, timers, block calls, etc. It supports the STEP 7 block concept and enables block programming in compliance with standards in addition to programming with Ladder Logic (LAD) and Function Block Diagram (FBD). This means SCL supplements and expands the STEP 7 programming software with its LAD and FBD programming languages.

You do not have to create every function yourself but can use pre-compiled blocks, such as system functions and system function blocks that are present in the CPU's operating system.

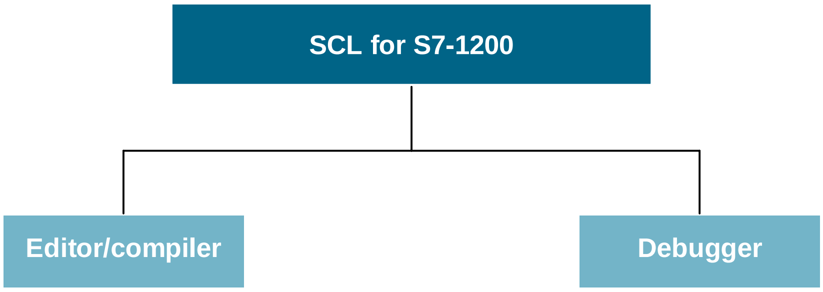
Blocks that are programmed with SCL can be mixed with LAD and FBD blocks. This means that a block programmed with SCL can call another block that is programmed in LAD or FBD. Accordingly, SCL blocks can also be called in LAD and FBD programs.

SCL networks can also be inserted in LAD and FBD blocks.

The SCL test functions can be used to find logical programming errors in an error-free compilation.

4.2 SCL development environment

There is a development environment that is tailored to the specific properties of both SCL and STEP 7 for use and application of SCL. This development environment consists of an editor/compiler and a debugger.



Editor/compiler

The SCL editor is a text editor that can be used to edit any text. The main task of the SCL editor is the creation and editing of blocks for STEP 7 programs. A basic syntax check is performed during the input which makes it easier to avoid errors during programming. Syntax errors are displayed in different colors.

The editor offers the following options:

– Programming of an S7 block in the SCL language

– Convenient insertion of language elements and block calls using drag & drop

– Direct syntax check during programming

– Customization of the editor to meet your needs, e.g. color-coding for the different language elements according to syntax

– Checking of the finished block through compiling

– Display of all errors and warnings that occur during compiling

– Localization of error locations in the block, optionally with error description and information on troubleshooting

Debugger

The SCL debugger enables you to check a program while it is running in the automation system (AS) and thus find potential logical errors.

SCL provides two different test modes:

– Continuous monitoring

– Step-by-step monitoring

With "Continuous monitoring" you can test a group of instructions within a block. During the test, the values of the tags and parameters are displayed in chronological order and – if possible – updated cyclically.

With "Step-by-step monitoring" the logical program sequence is followed. You can run the program algorithm instruction-by-instruction and observe how the contents of the processed tags change in a result window.

The type of CPU you are using determines whether or not you can use "Step-by-step monitoring". The CPU must support the use of breakpoints. The CPU used in this document does not support breakpoints.

5 Task

5.1 Example task – Tank volume

In the first part, you are to program the calculation of the tank volume.

5.2 Expansion of the sample task

In the second part, the task is expanded and you are to program an error evaluation.

6 Planning

The tank is in the shape of a vertical cylinder. The filling level is measured with an analog sensor. For the first test, the filling level value should be available as a scaled value (in meters).

Global parameters, such as the diameter and height of the tank, are to be stored in a structured manner in a global data block "Data_Tank".

The program for calculation of the volume should be written in a "Calculate_Volume" function and the parameters are to use the unit 'meter' or 'liter'.

6.1 Global data block "Data_Tank"

The global parameters are stored in multiple structures in a global data block.

Name	Data type	Start value	Comment
Dimensions	STRUCT		
Height	REAL	12.0	in meter
Diameter	REAL	3.5	in meter
measured_data	STRUCT		
filling_level_per	INT	0	value between 0...27648
filling_level_scal	REAL	0.0	range 0…12.0.
Volume	REAL	0.0	Volume of tank in liter
fault_flags	STRUCT		
calculate_volume	BOOL		fault == true
Scaling	BOOL		fault == true

Table 1: Parameters in the "Data_Tank" data block

6.2 "Calculate_Volume" function

This block calculates the volume of the tank in liters.

In the first step, there is to be no check of the transferred parameters for reasonableness.

The following parameters are required for this step:

Input	**Data type**	**Comment**
Diameter	REAL	Diameter of cylindric tank in meter
Filling_level	REAL	Filling level of liquid in meter
Output		
Volume	REAL	Volume of liquid in the tank in liter

Table 2: Parameters for "Calculate_Volume" function in the first step

The formula for calculating the volume of a vertical cylinder is used to solve the task. The conversion factor 1000 is used to calculate the result in liters.

$$V = \frac{d^2}{4} \cdot p \cdot h \quad => \quad \#\mathrm{Volume} = \frac{\#\mathrm{Diameter}^{2}}{4} \cdot 3.14159 \cdot \#\mathrm{Filling\_level} \cdot 1000$$

6.3 Expansion of the "Calculate_Volume" function

The second step checks whether the diameter is greater than zero. In addition, a test is to be performed to determine whether the filling level is greater than or equal to zero and less than or equal to the height of the tank.

In case of an error, the new parameter "er" is set to TRUE, and the "Volume" parameter is set to the value -1.

For this purpose, add the "er" and "Height" parameters to the interface.

Input	**Data type**	**Comment**
Height	REAL	Height of cylindric tank in meter
Diameter	REAL	Diameter of cylindric tank in meter
Filling_level	REAL	Filling level of liquid in meter
Output		
er	BOOL	fault flag; fault == true
Volume	REAL	Volume of liquid in the tank in liter

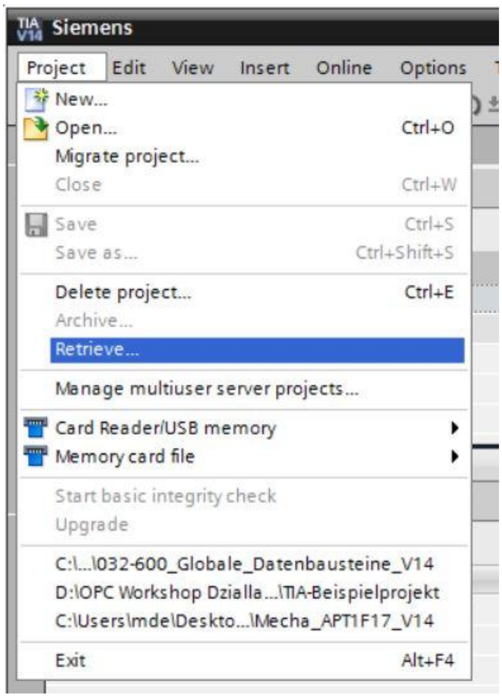
Table 3: Parameters for "Calculate_Volume" function in the second step

7 Structured step-by-step instructions

You can find instructions on how to implement the planning below. If you already have a good understanding of everything, it is sufficient to focus on the numbered steps. Otherwise, simply follow the steps of the instructions explained below.

7.1 Retrieving an existing project

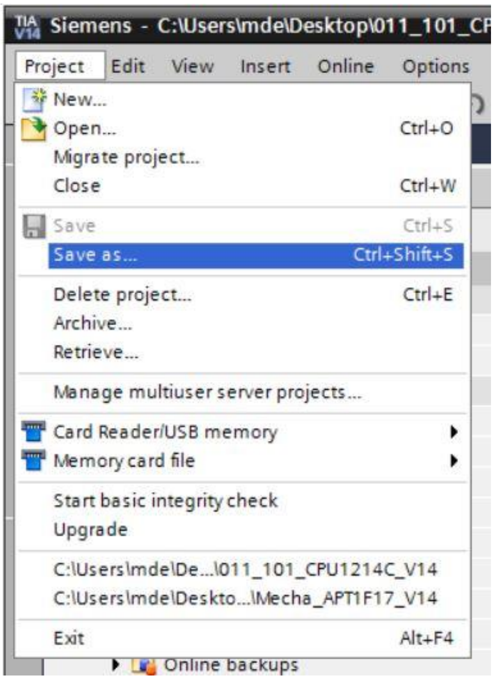
® Before you can start programming, you need a project with a hardware configuration. (e.g. SCE_EN_011-101_Hardware_Configuration_CPU1214C_....zap14).
To retrieve an existing project, you must select the respective archive from the Project view under ® Project ® Retrieve. Confirm your selection with "Open".
(® Project ® Retrieve ® Selection of a .zap archive ® Open)



® Next you can select the target directory to which you want to save the retrieved project. Confirm your selection with "OK". (® Project ® Save as... ® OK)

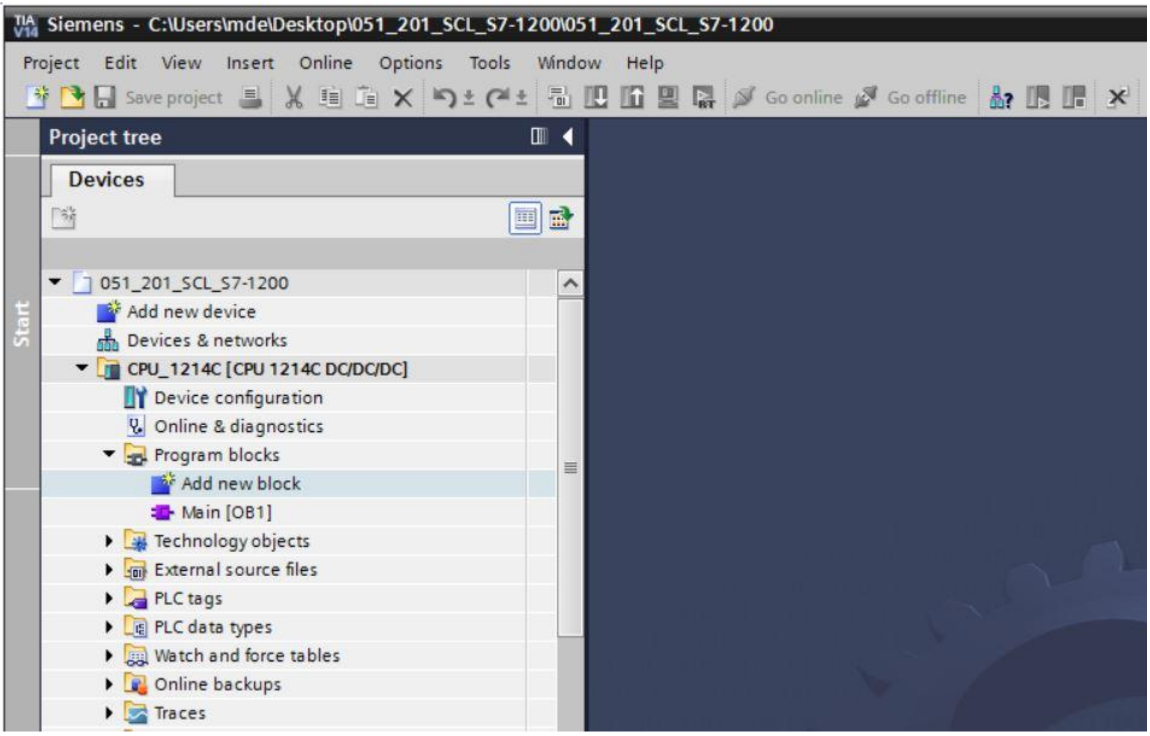
7.2 Saving the project under a new name

® You save the opened project under the name 051-201_SCL_S7-1200.
(® Project ® Save as … ® 051-201_SCL_S7-1200 ® Save)



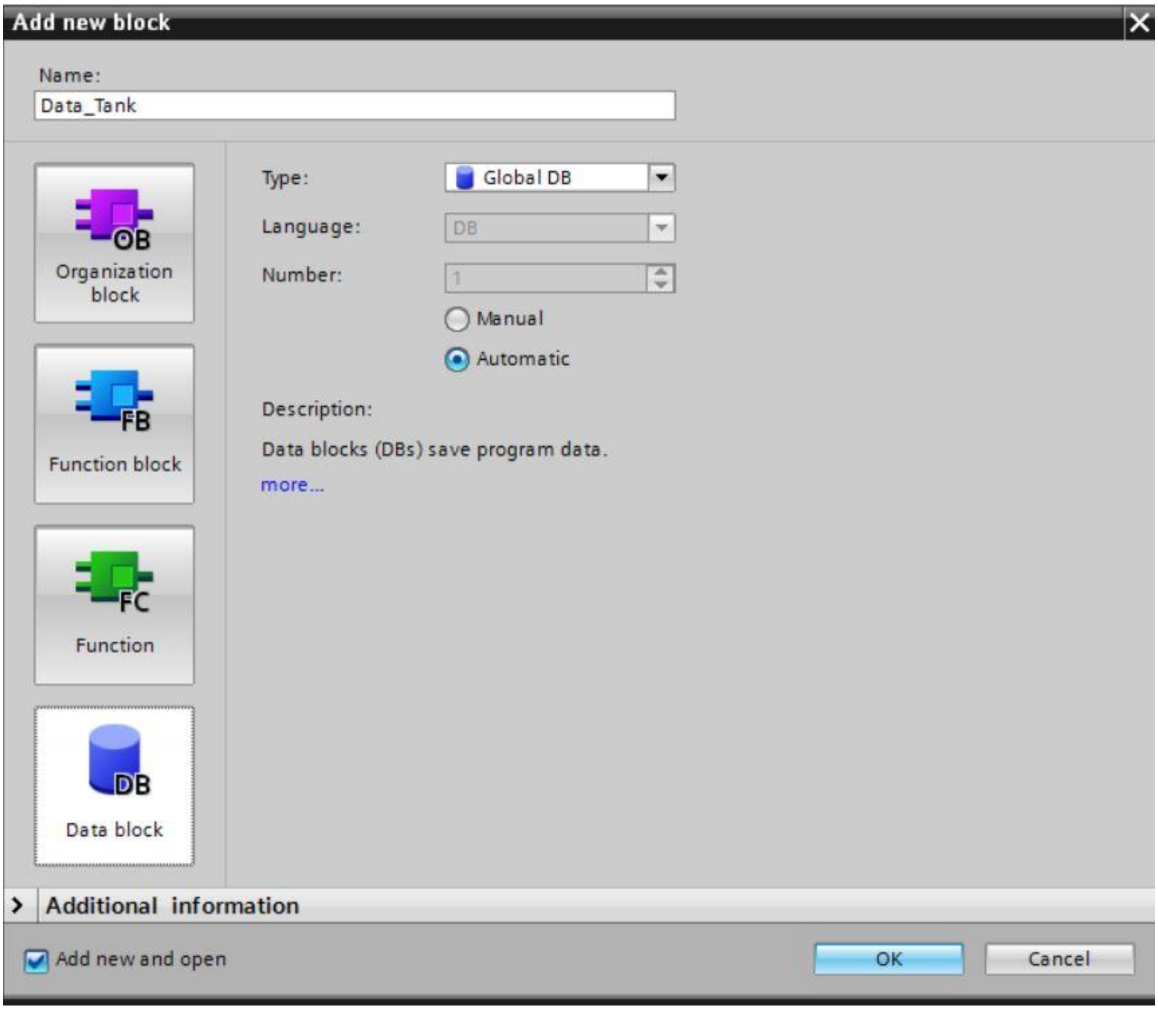
7.3 Creating the "Data_Tank" data block

® In the Project view, navigate to ® the program blocks and create a new block by double-clicking ® Add new block.

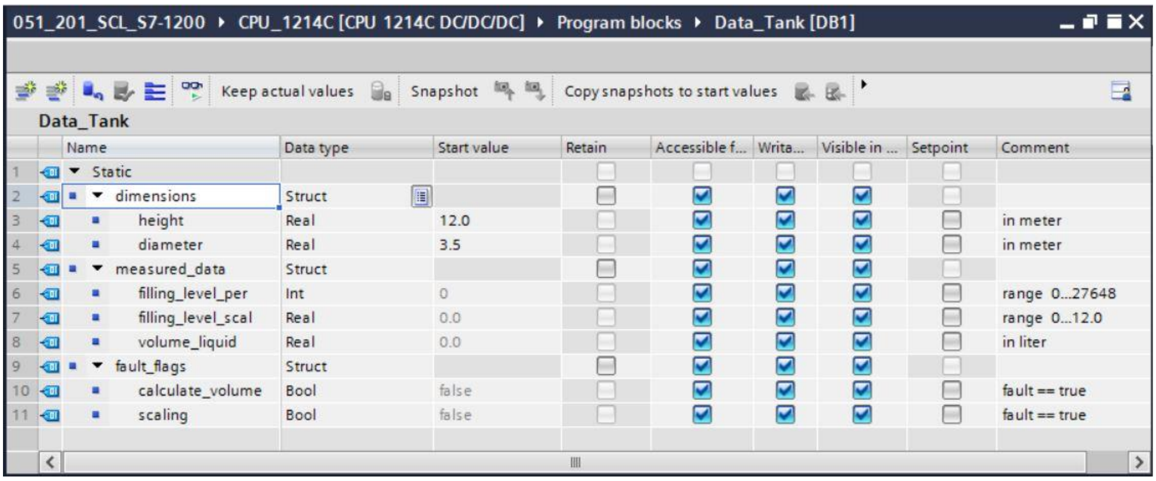


® Now select a data block and enter the name.

(®  ® "Data_Tank" ® OK)



® Next, enter the names of the tags listed below with data type, start value and comment.

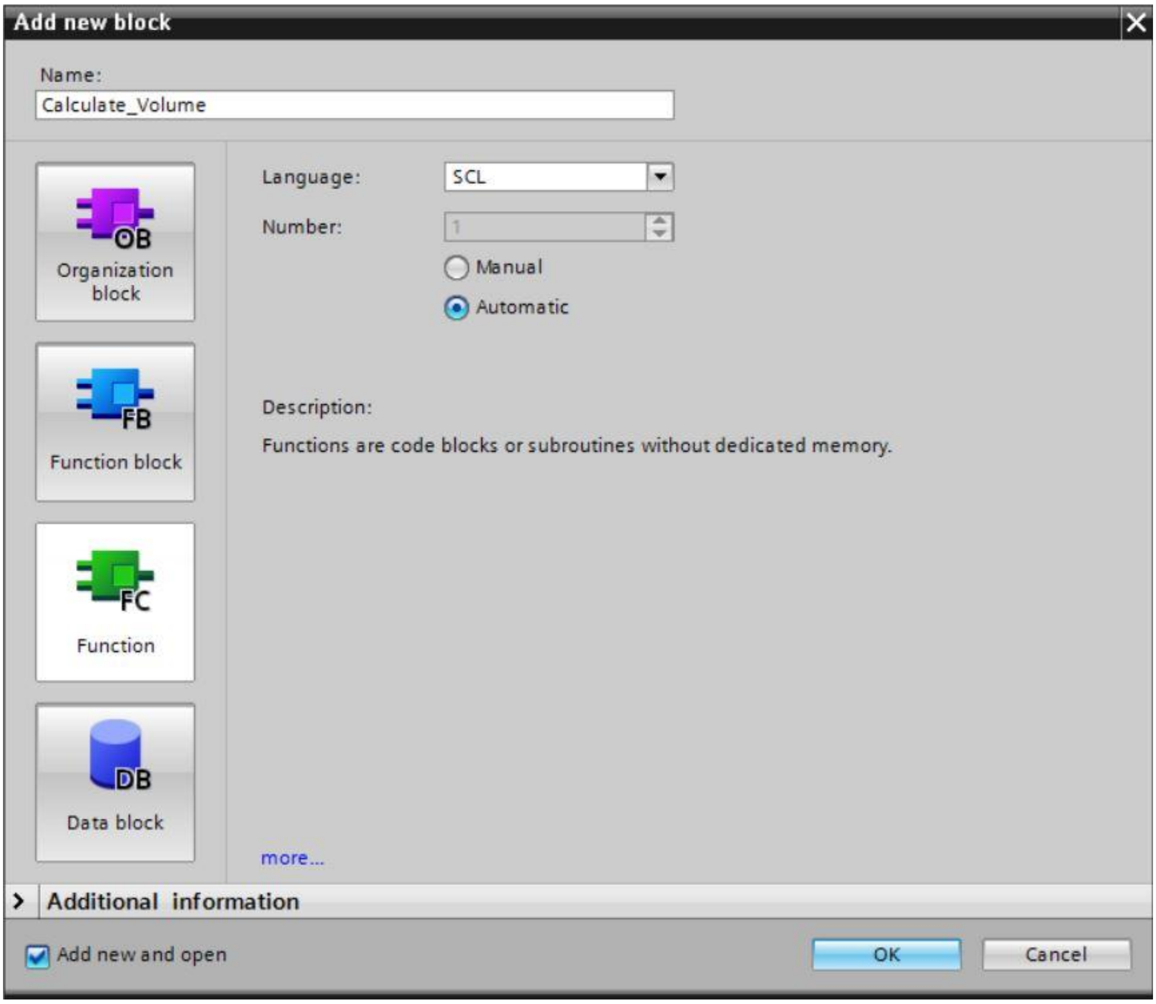


7.4 Creating the "Calculate_Volume" function

® Next, add a function, enter the name and select the language.

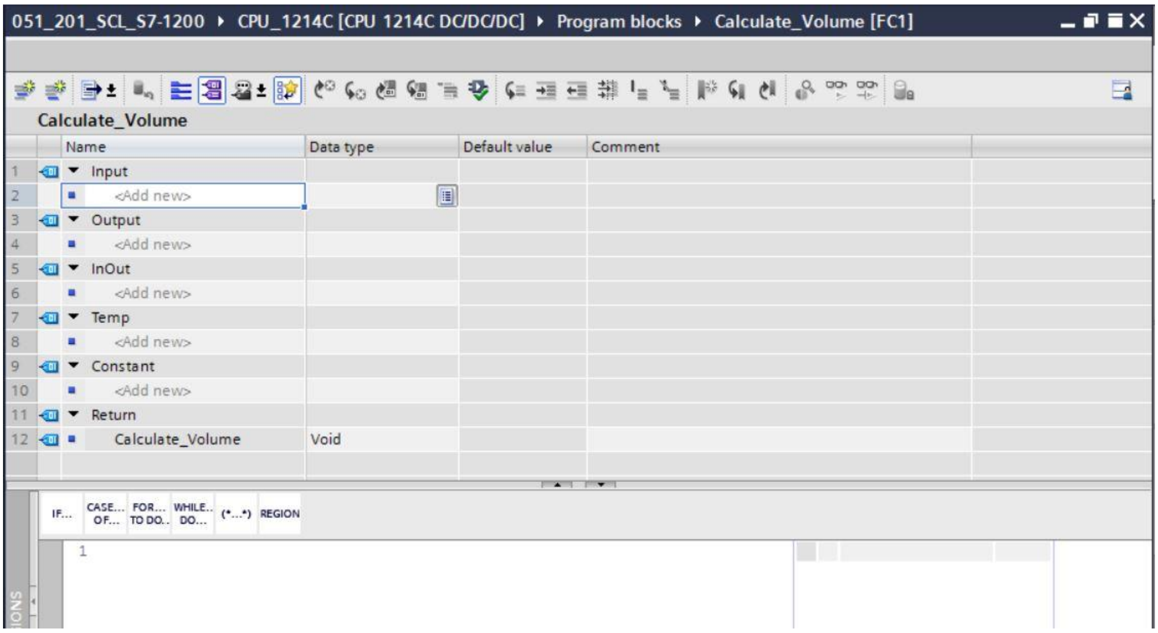
(® Add new block ® 

® "Calculate_Volume" ® SCL ® OK)



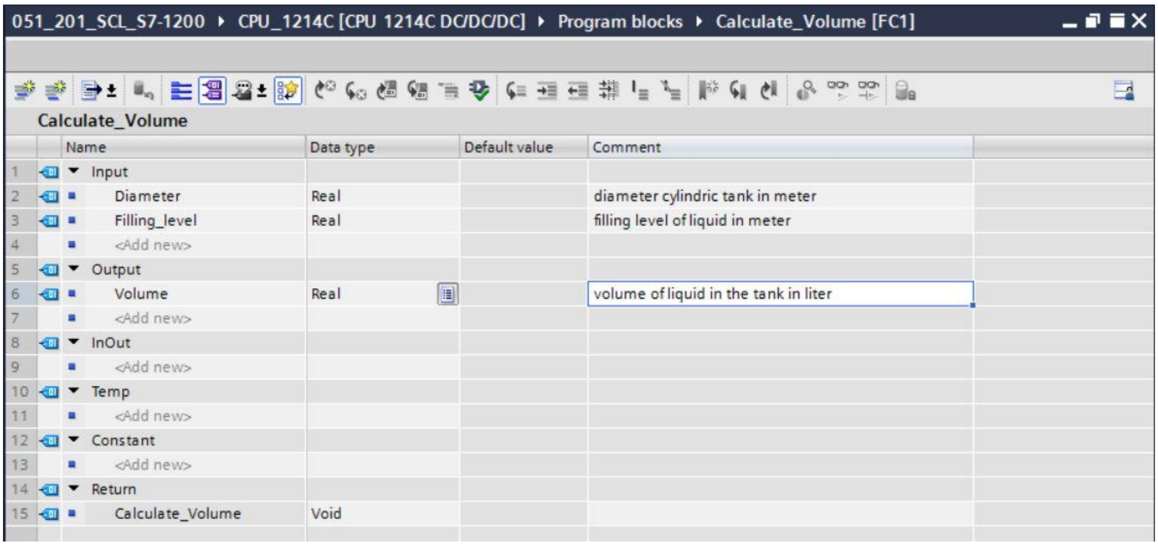
7.5 Specifying the interface of the "Calculate_Volume" function

® The top section of your programming view shows the interface description of your function.



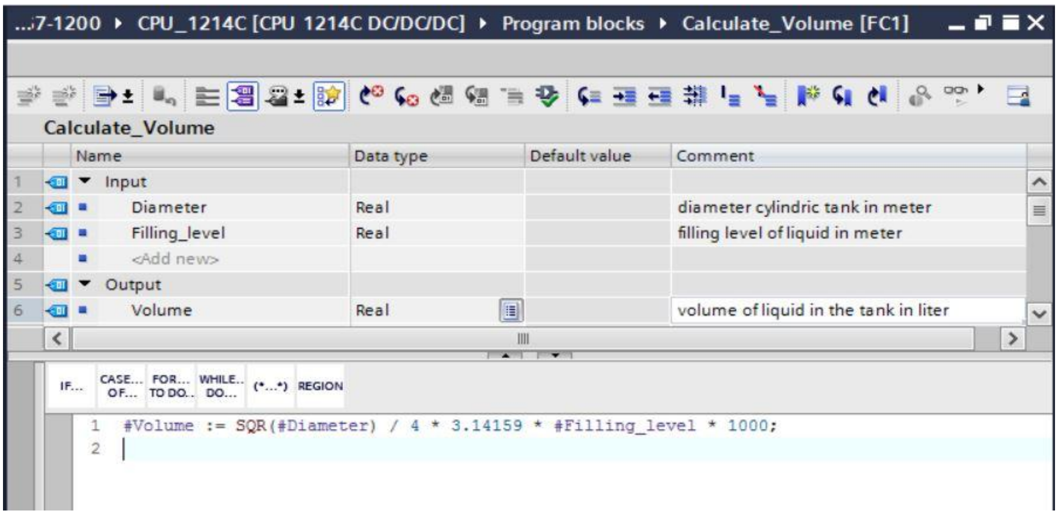
® Create the following input and output parameters.

(® Name ® Data type ® Comment)



7.6 Programming the "Calculate_Volume" function

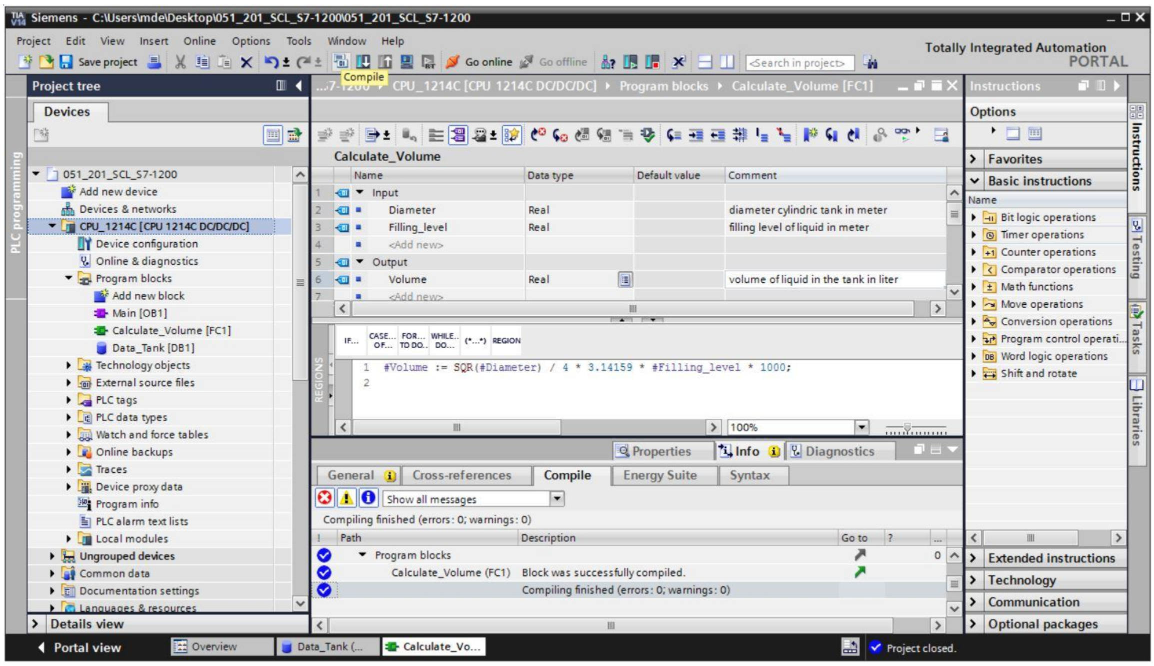
® Enter the program shown below. (® Enter program)

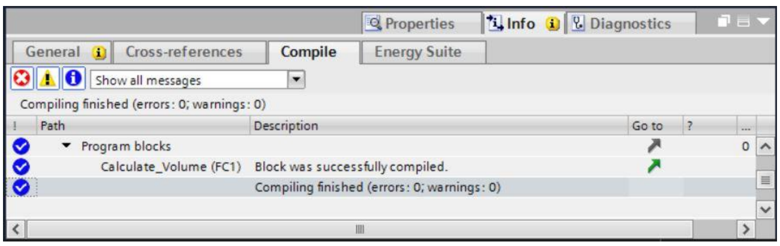


® Now compile your program and check it for syntax errors. These are displayed in the Inspector window below the programming. Correct any errors and compile the program again.

Then save your program. (®  ® Eliminate errors ® 

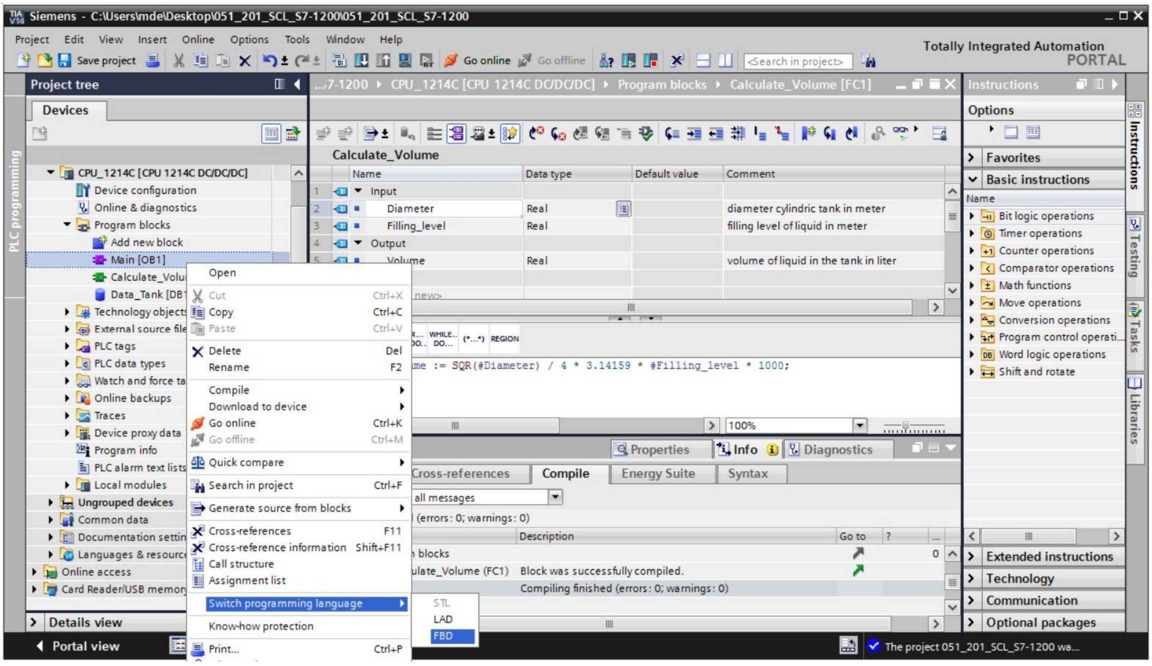
)



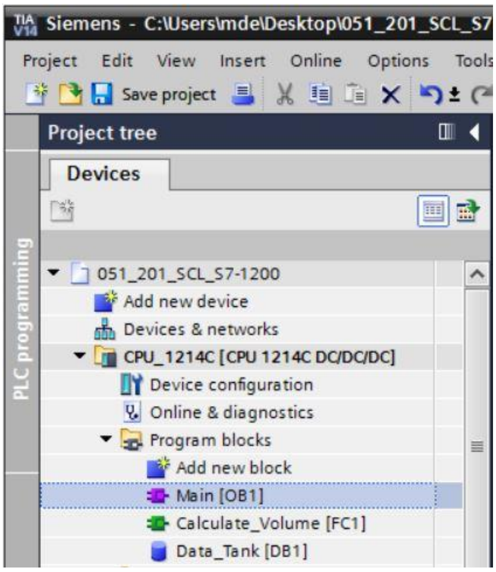


7.7 Programming the "Main [OB1]" organization block

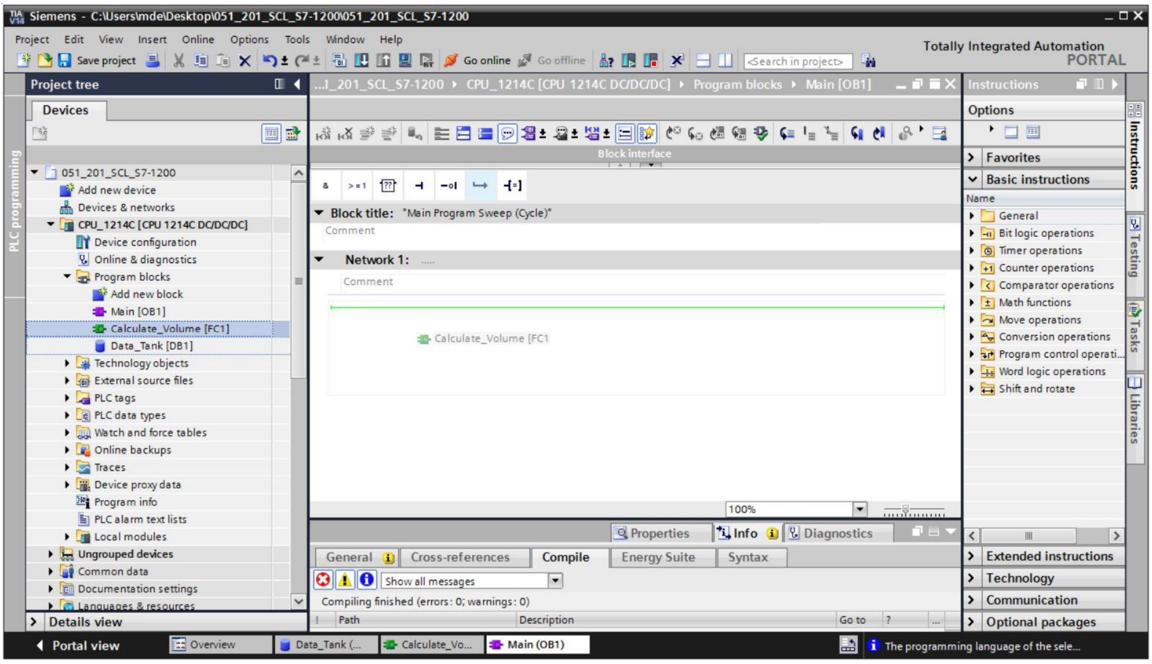
® Before programming the "Main [OB1]" organization block, switch the programming language to FBD. To do this, click on "Main [OB1]" in the "Program blocks" folder. (® CPU_1214C[CPU 1214C DC/DC/DC] ® Program blocks ® Main [OB1] ® Switch programming language ® FBD)

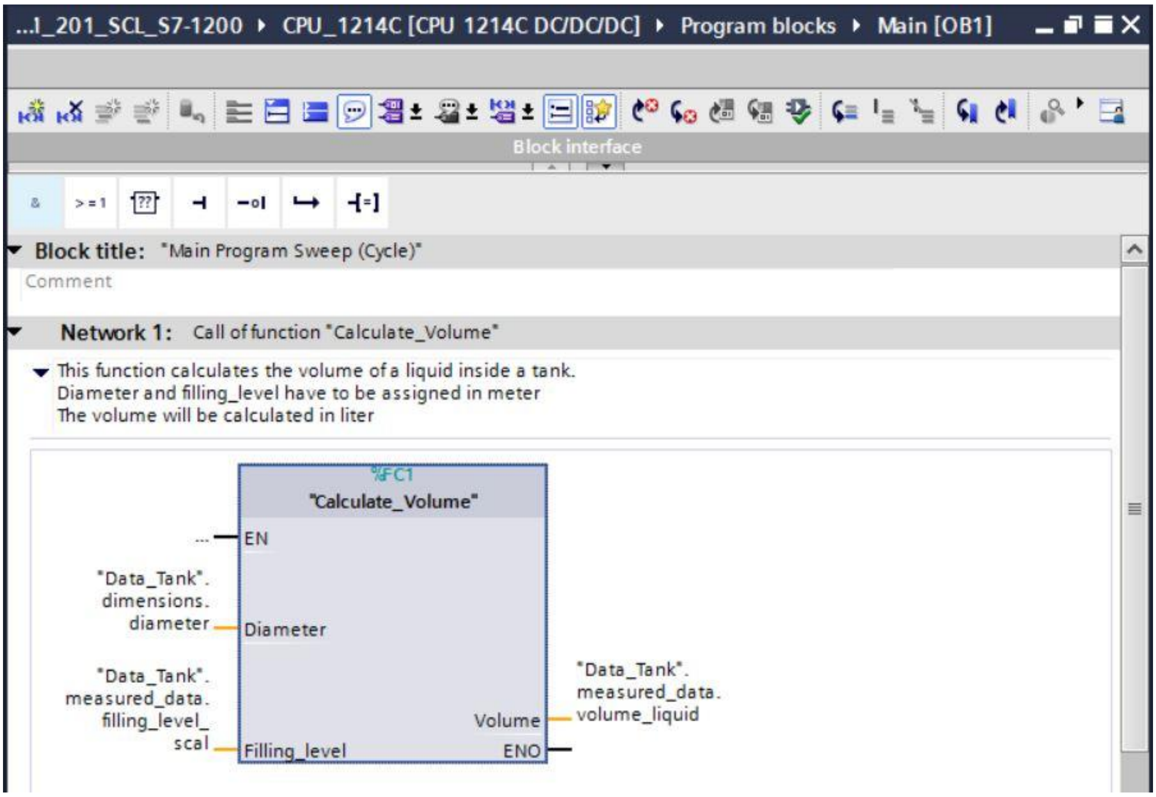


® Now double-click the "Main [OB1]" organization block to open it.



® Call the "Calculate_Volume" function in the first network. Assign network title, comment and connect the parameters. (® Call "Calculate_Volume" ® Assign network title ® Write network comment ® Connect parameters)

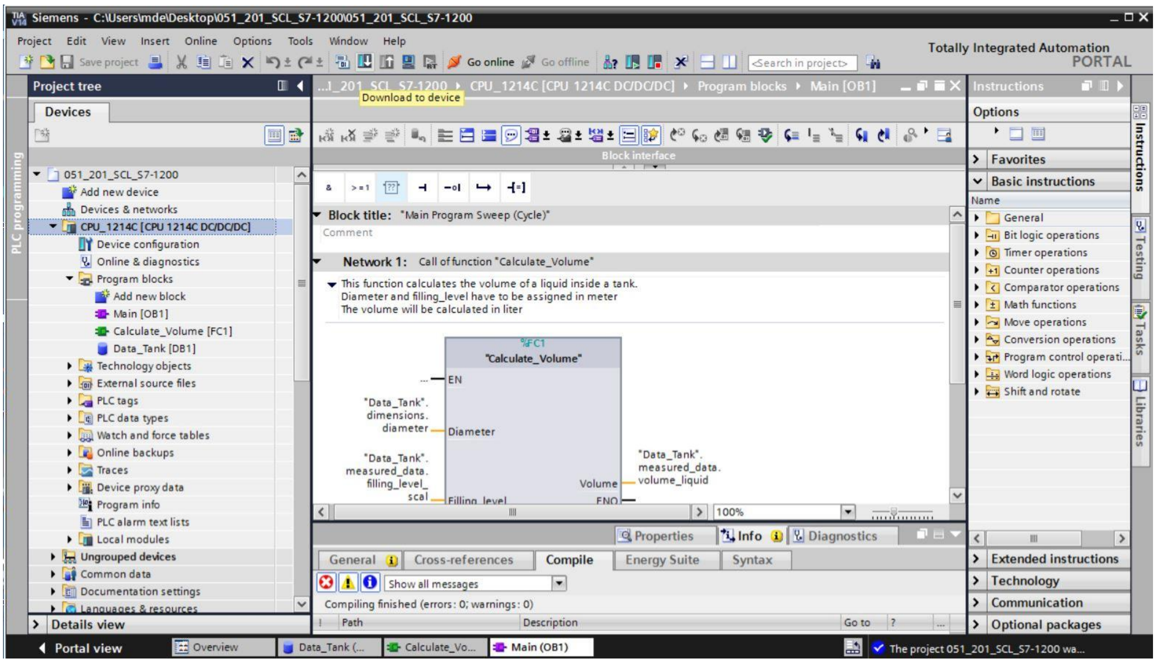




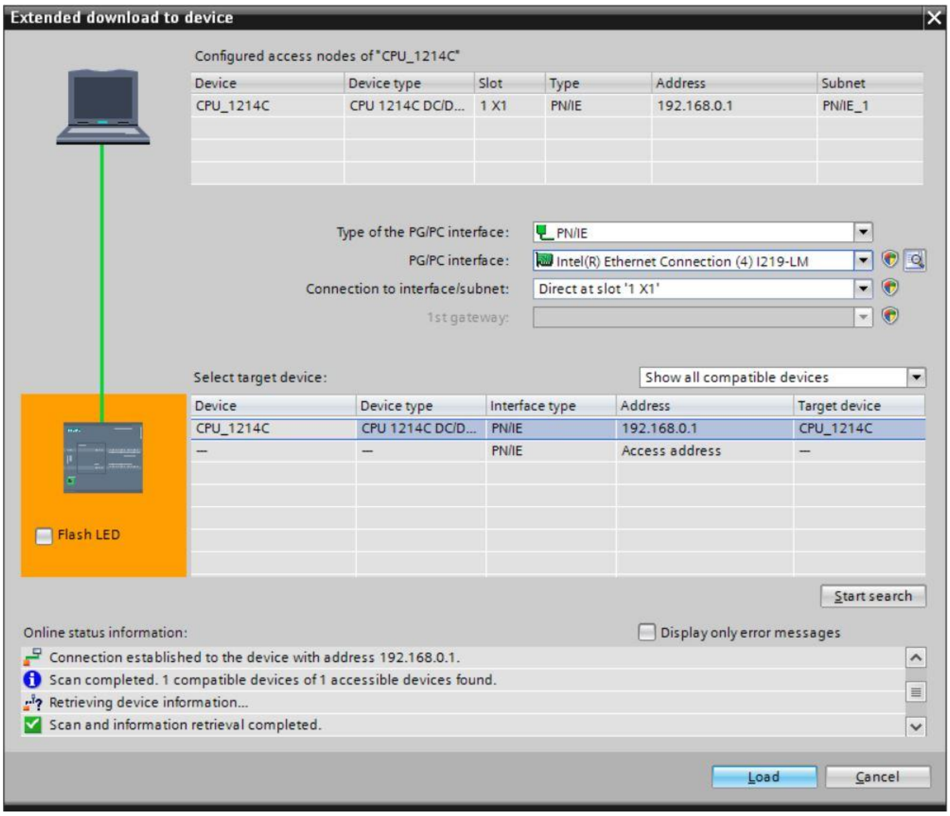
7.8 Compiling and downloading the program

® Click the "Program blocks" folder and compile the entire program. After successful compilation, save your project and download it to the controller.

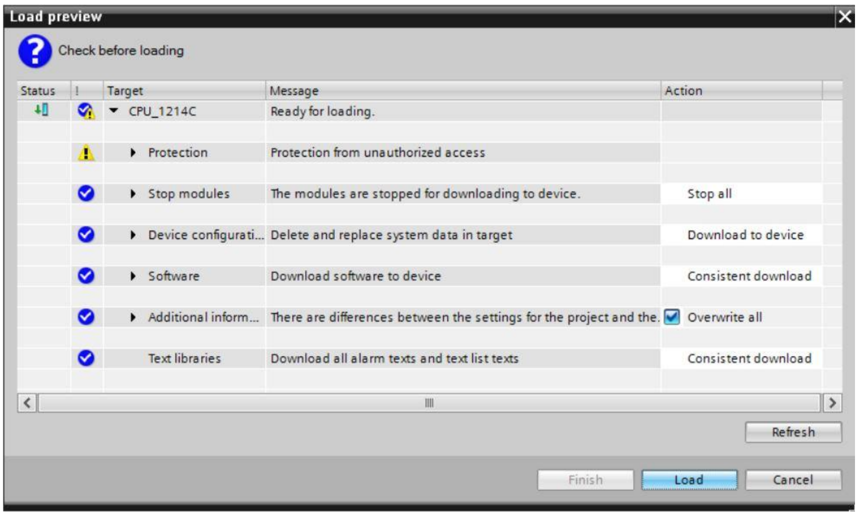
(® ® Save project ®)



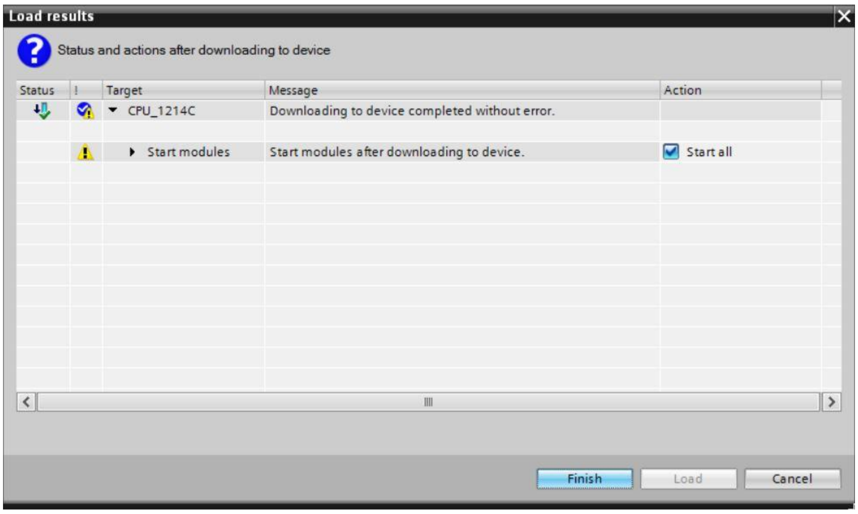
® Select PG/PC interface ® Select subnet ® Start search ® Load



® Make a selection, if necessary ® Load

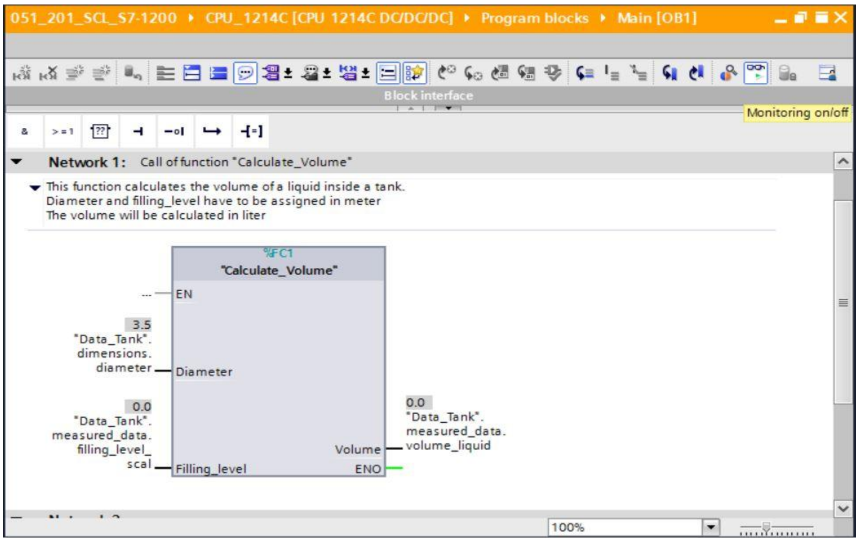


® Finish

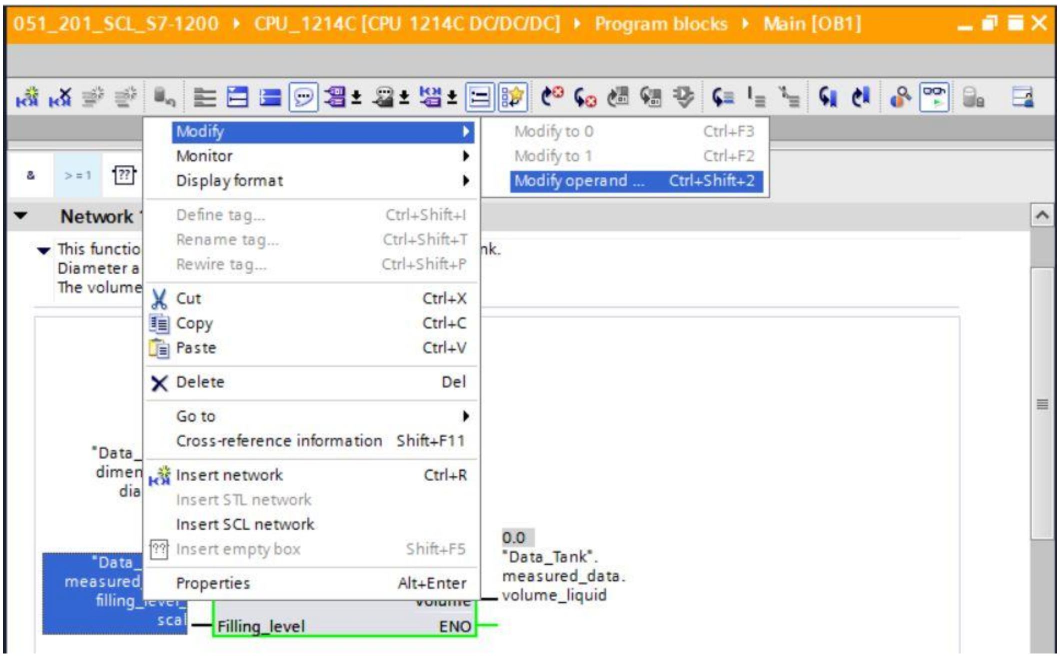


7.9 Monitoring and testing the organization block

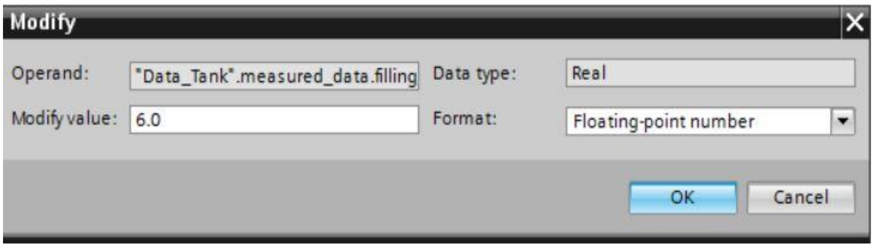
® In the open OB1, click the icon to monitor the block.



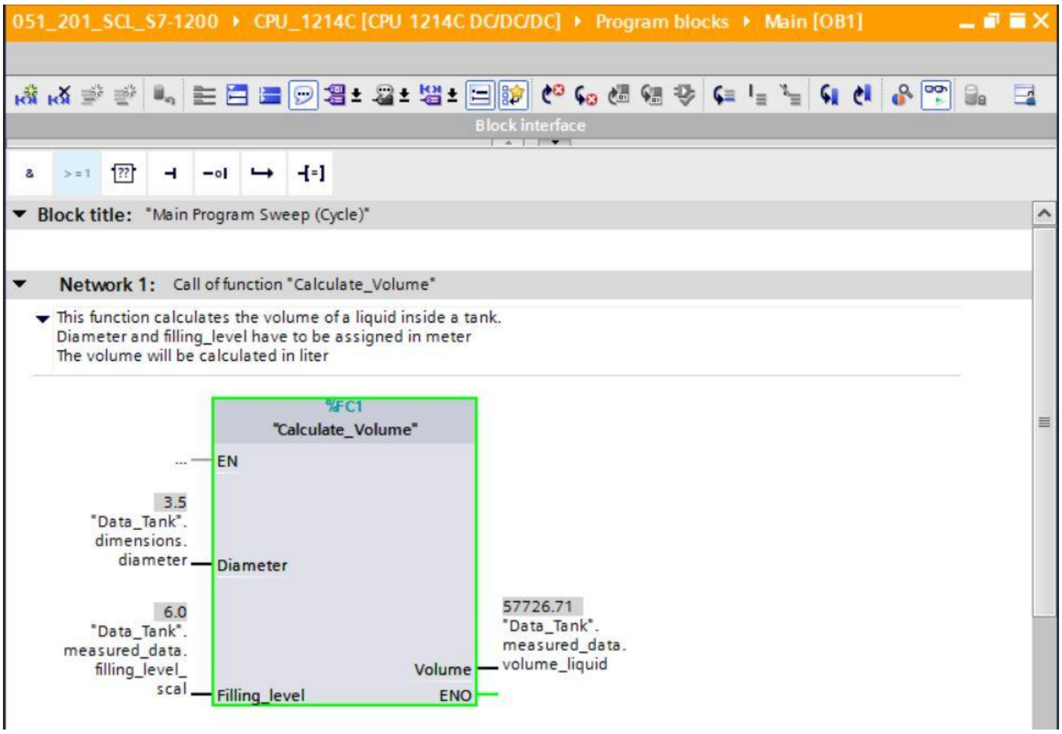
® Test your program by writing a value to the "Filling_level_scal" tag in the data block. (® Right-click on "Filling_level_scal" ® "Modify" menu ® Modify operand)



® Enter value 6.0 ® OK



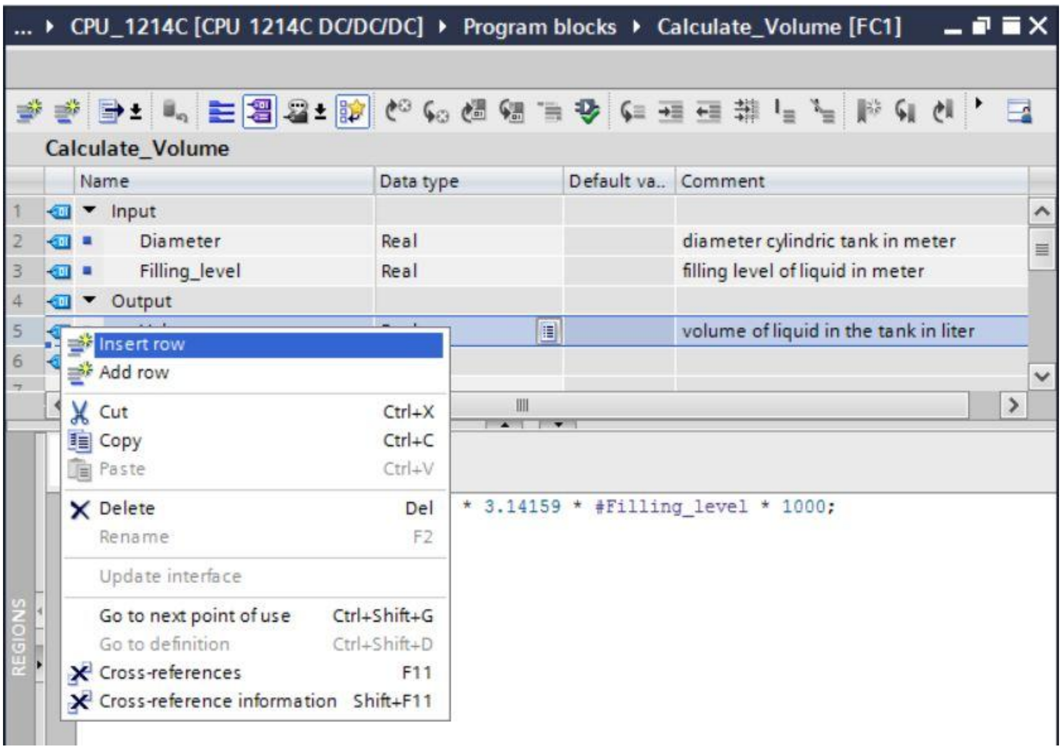
® Check the result for correctness.



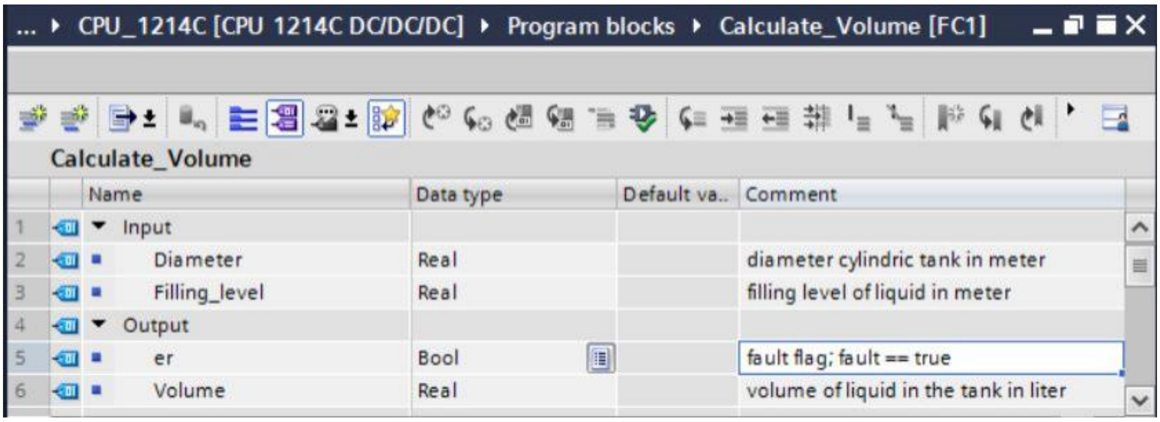
7.10 Expansion of the "Calculate_Volume" function

® Open the "Calculate_Volume" function, and insert a row in the output parameters by right-clicking the row in the interface.

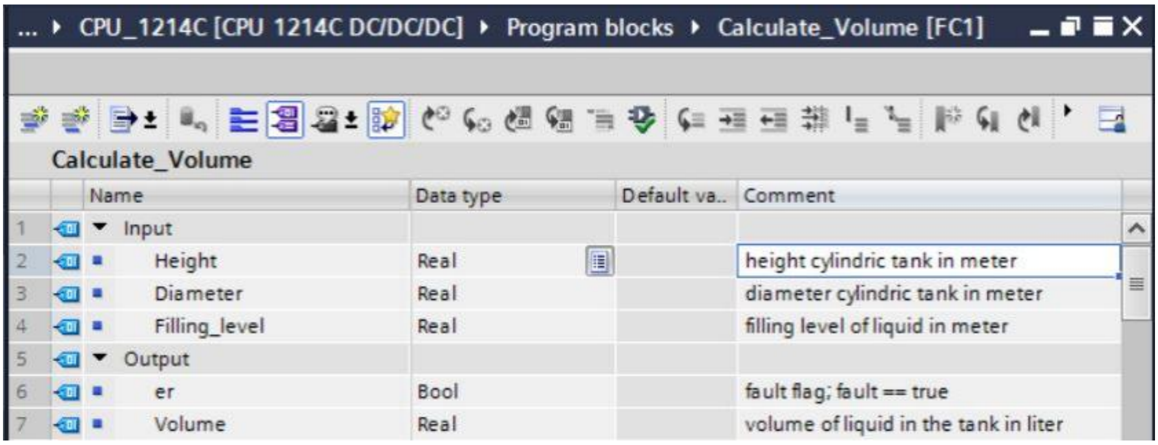
(® Open "Calculate_Volume" ® Right-click on row 5 ® Insert row)



® Enter the parameter "er" with data type BOOL and comment.

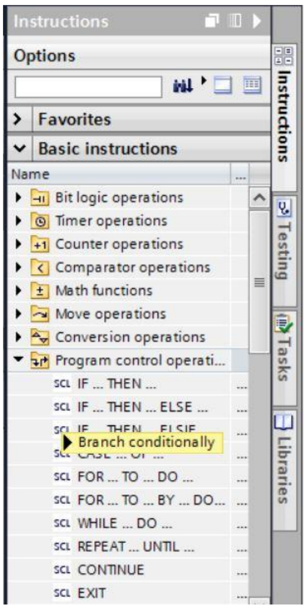


® Follow the same steps to add the "Height" tag with data type Real and comment.

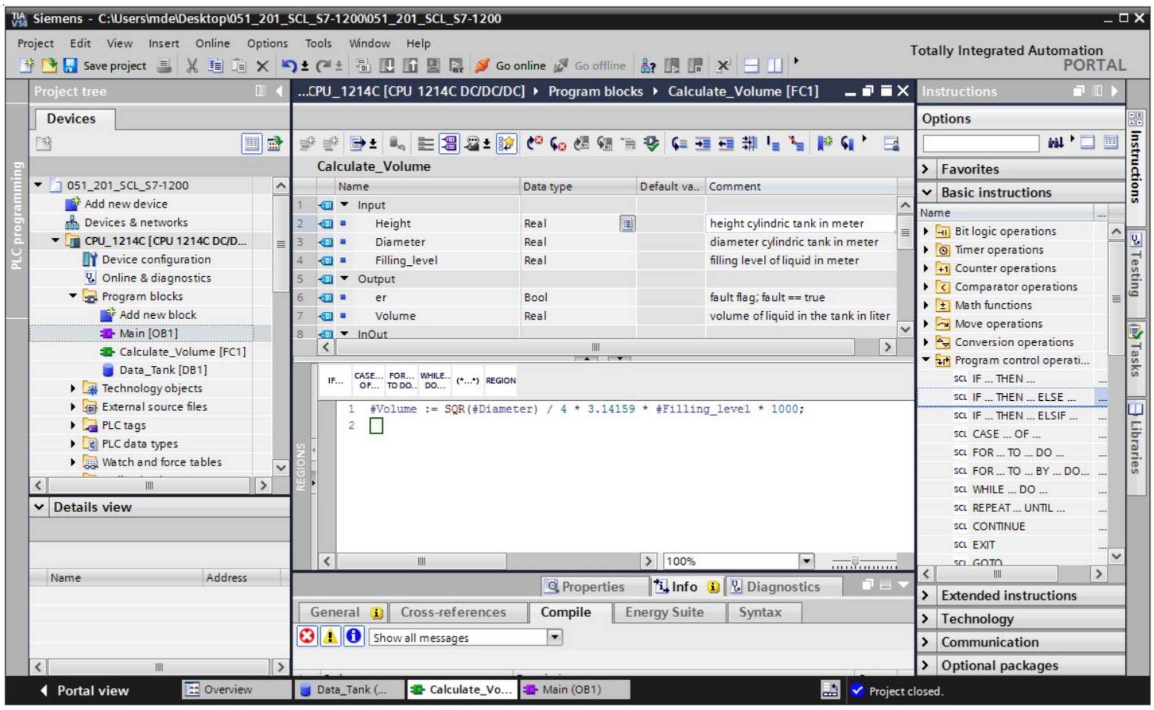


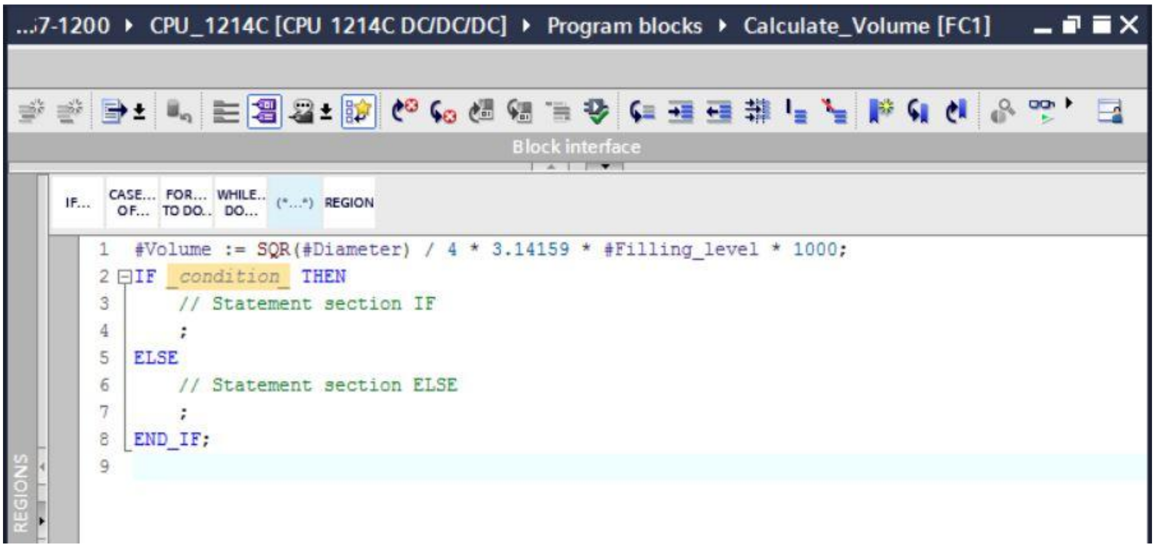
® Then go to the "IF…THEN…ELSE" control statement from the "Program control operations" folder of Basic instructions.

(® Instructions ® Basic instructions ® Program control operations ® "IF...THEN…ELSE")



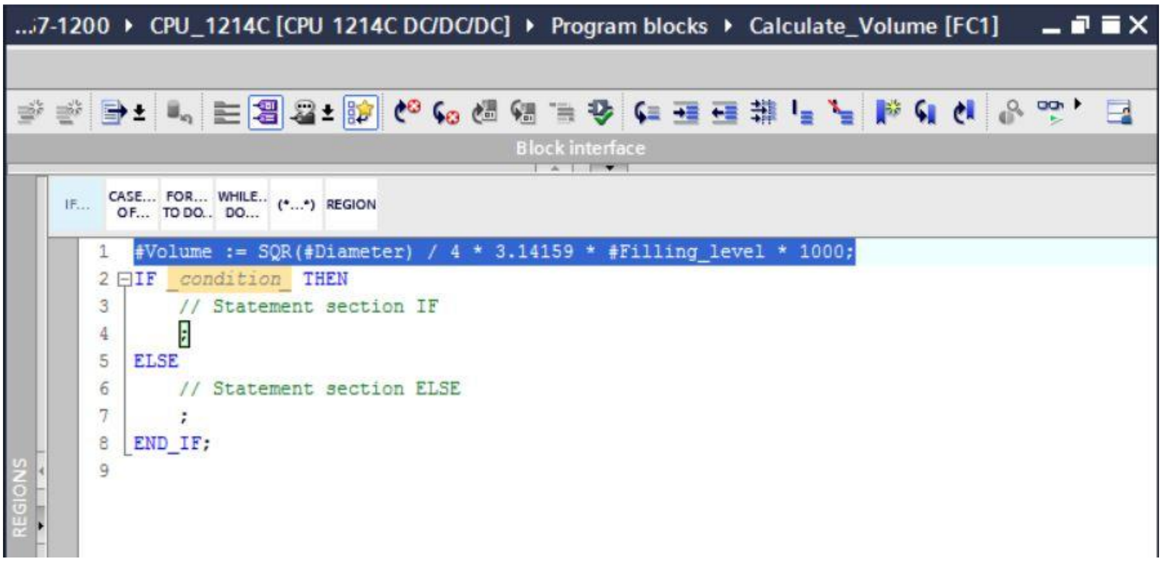
® Then drag the "IF...THEN...ELSE" control statement to the second row of the program. (® "IF…THEN…ELSE" ® drag & drop)

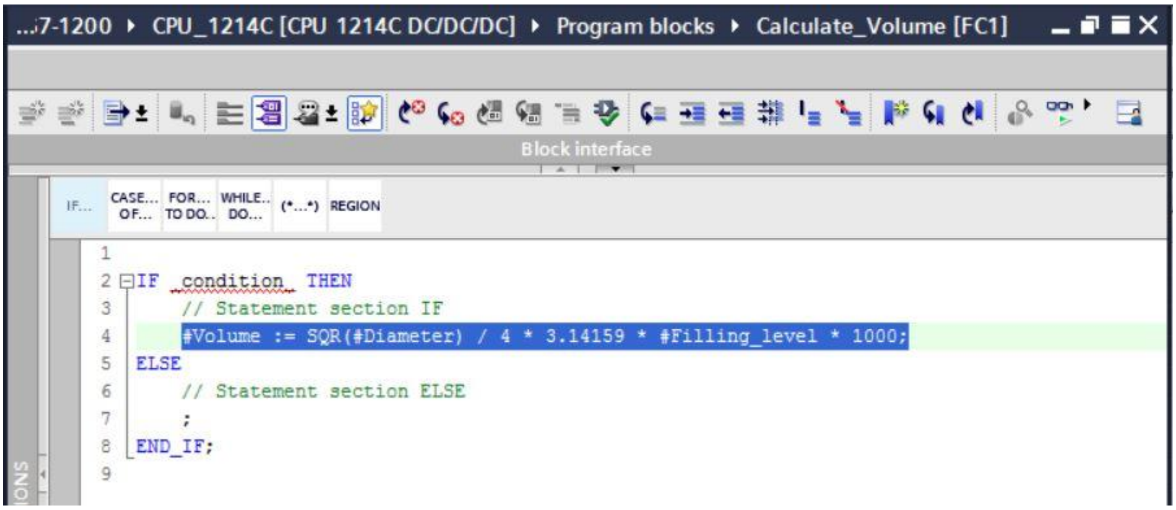




® Highlight the mathematical formula and move it onto the semicolon in front of the ELSE using drag & drop.

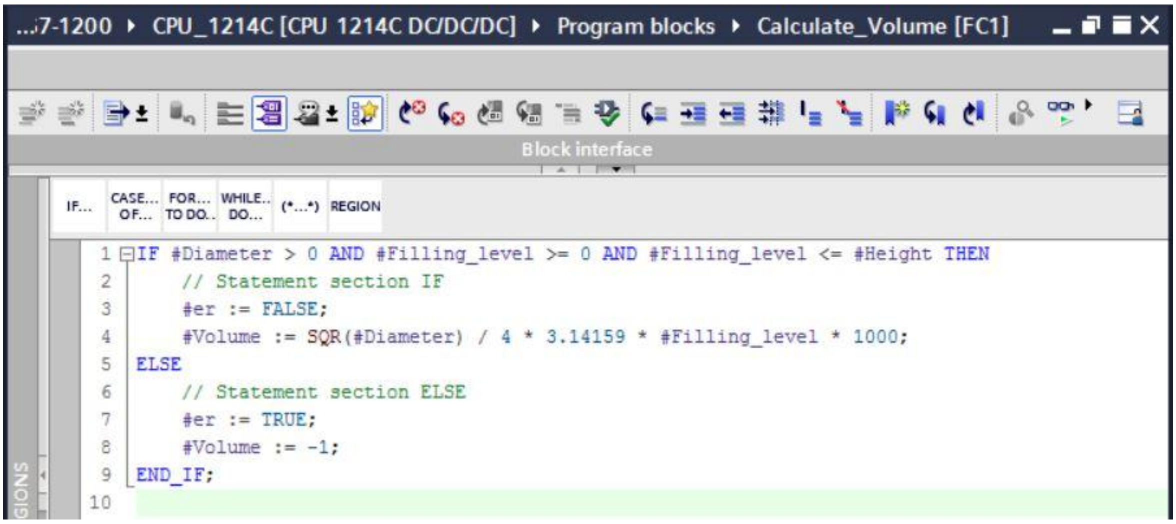
(® Select ® drag & drop)





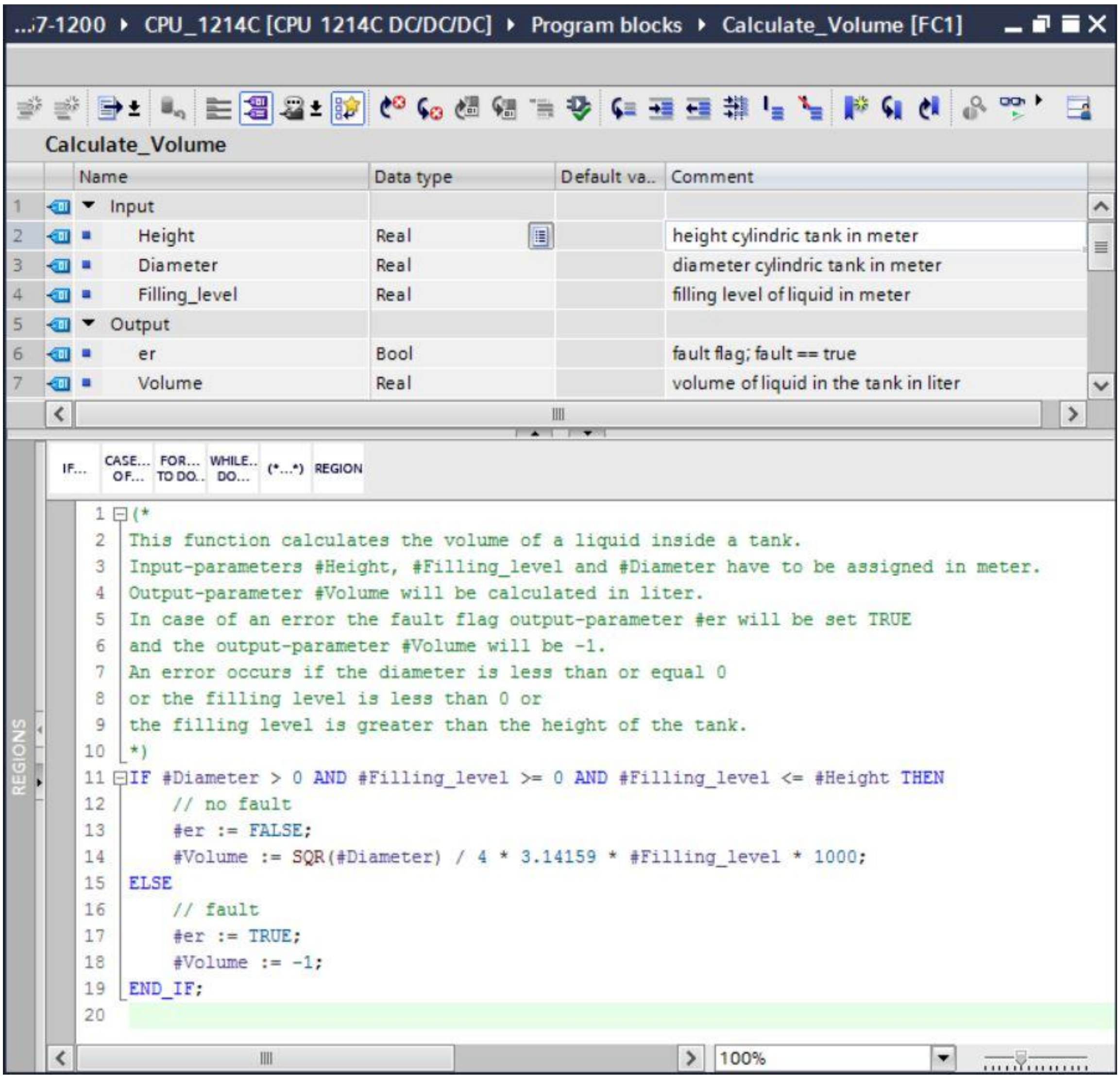
® Complete the function and check your program by compiling it.

(® Complete program ® )



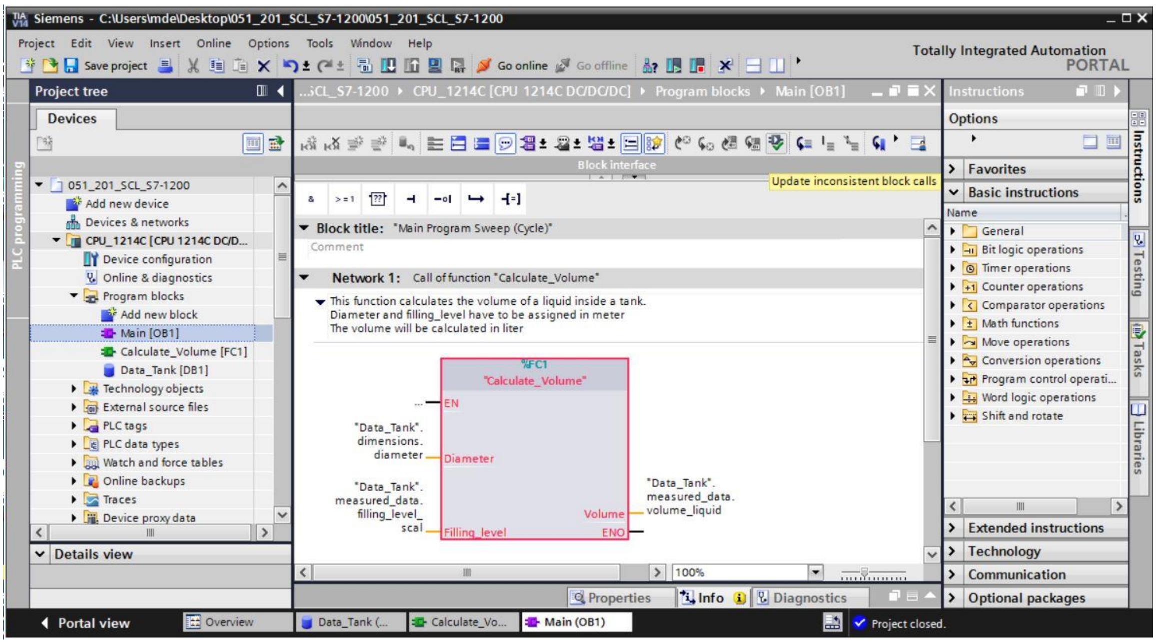
® Comments can be added with "(**)" as block comment and with "//" as row comment. You can now add comments to your program.

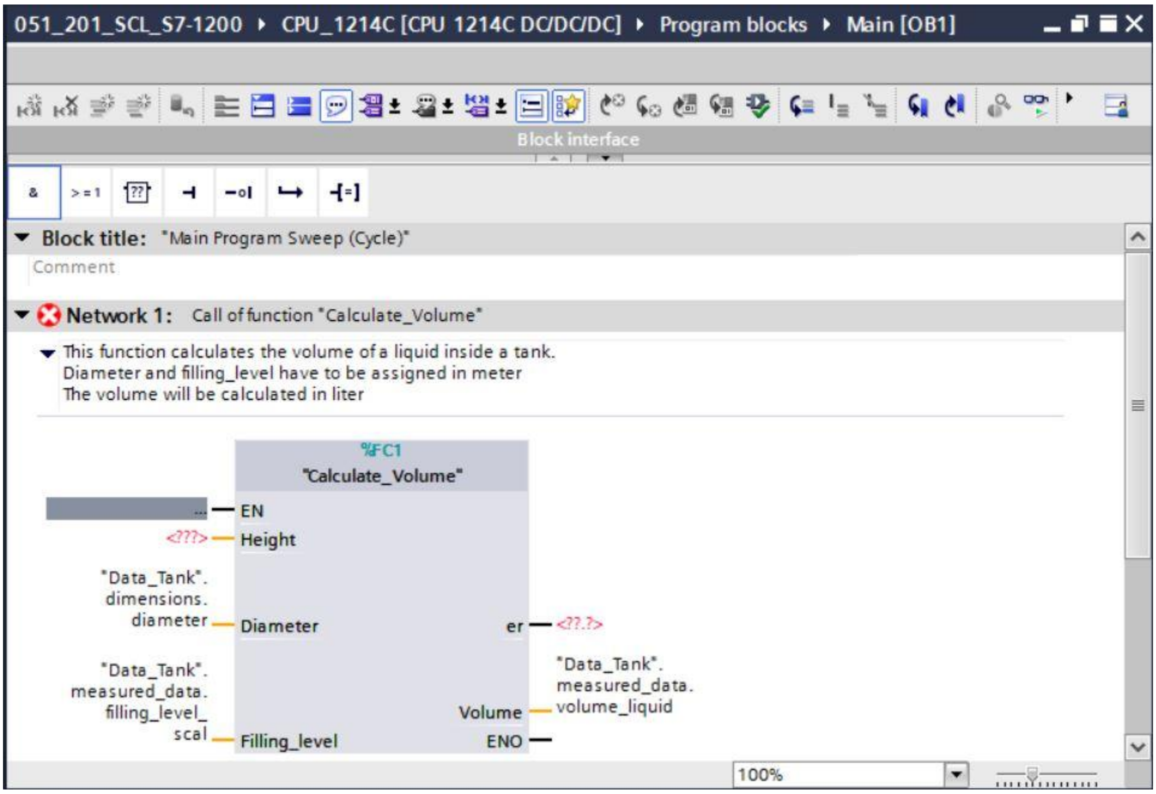
(® Insert block comment starting with row 1 ® Insert row comments in rows 12/16)



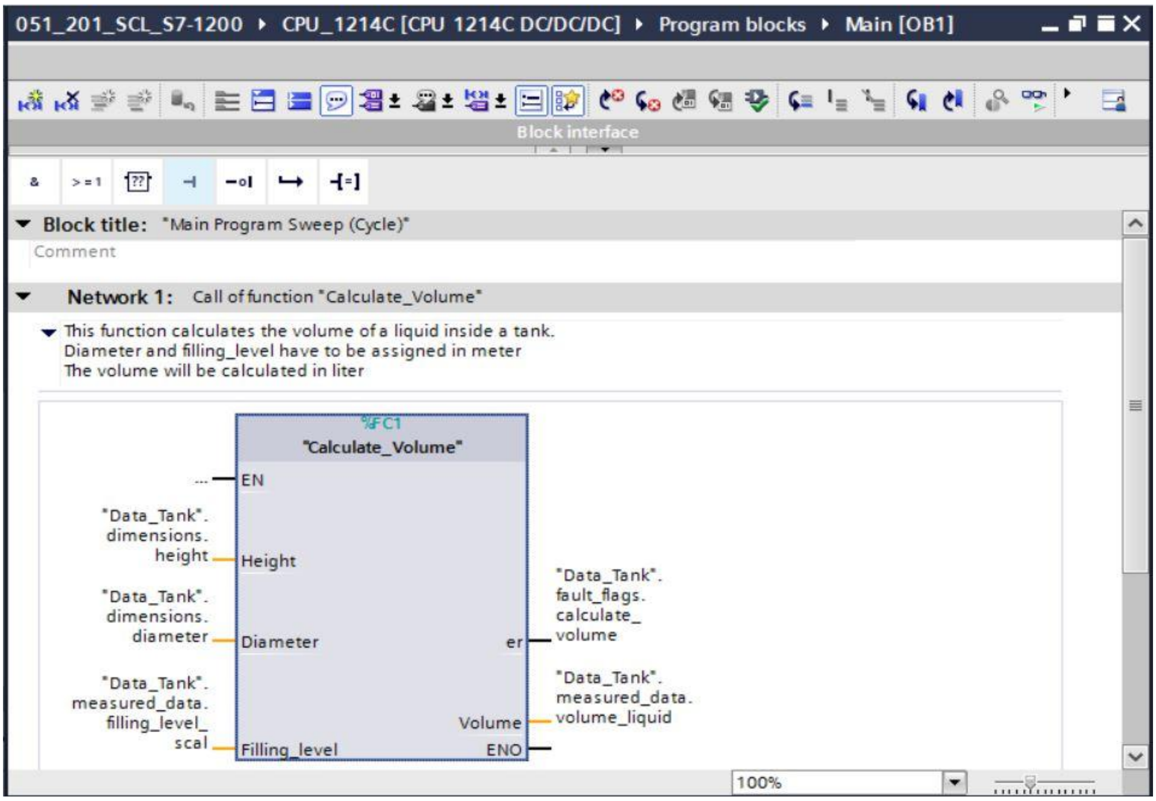
7.11 Customizing the organization block

® Open OB1 and update the inconsistent block calls by clicking . (® Open OB1 ® )



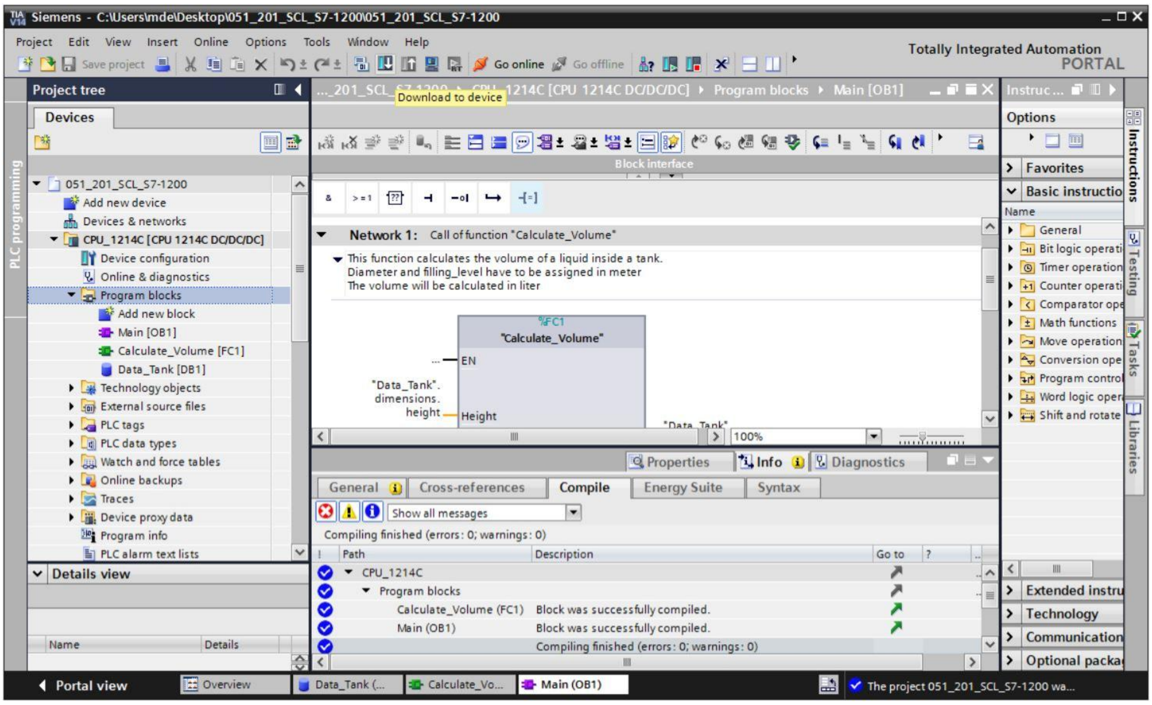


® To do this, add the parameters "er" and "Height".



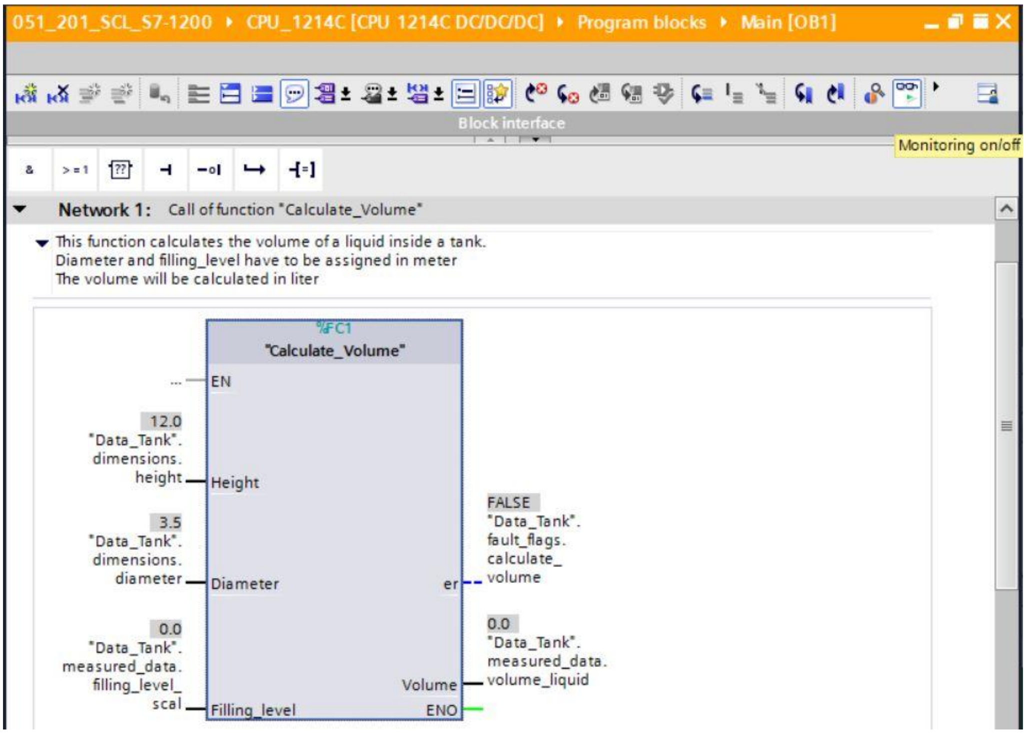
7.12 Compiling, saving and downloading the program

® Click the "Program blocks" folder, compile the entire program and then save it. After successful compilation and saving, download the project to the controller. (® Program blocks ® ® Save project ®)

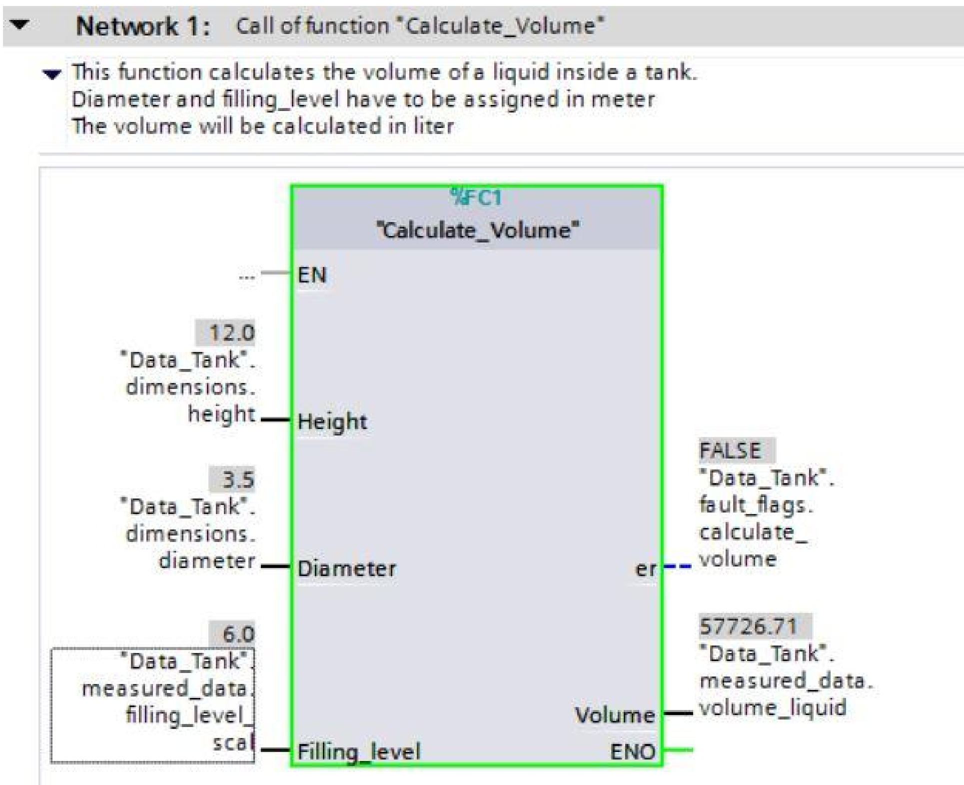


7.13 Monitoring and testing the organization block

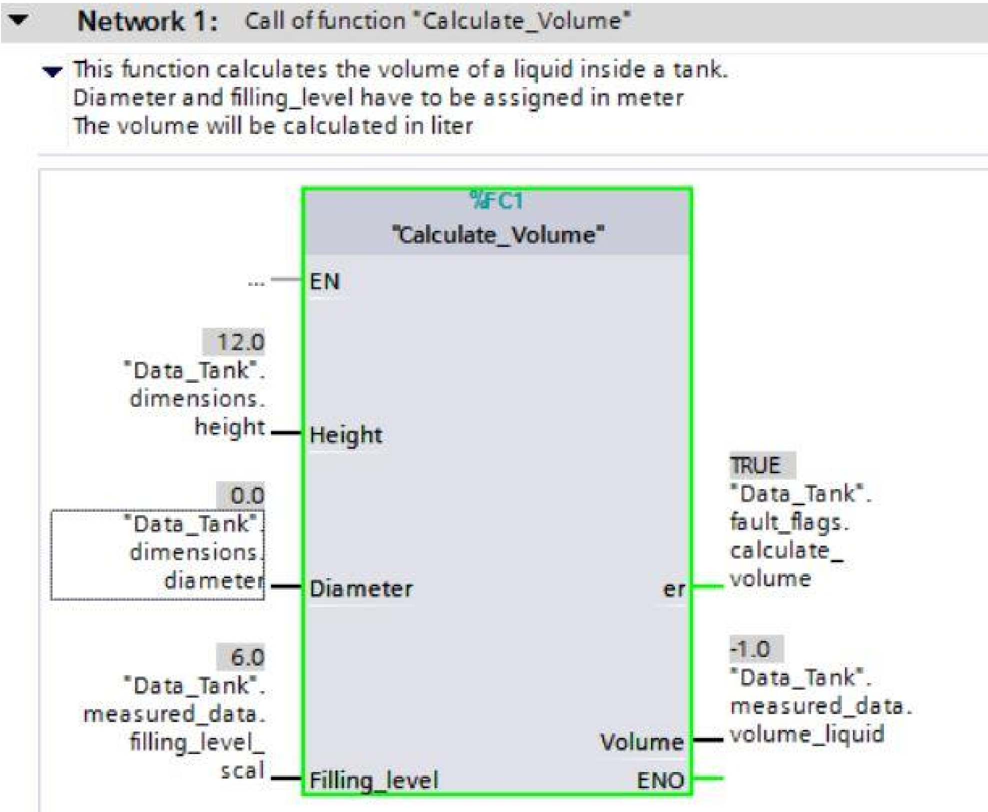
® In the open OB1, click the icon to monitor the block.



® Test your program by writing a value to the "Filling_level_scal" tag in the data block. (® Right-click on "Filling_level_scal" ® "Modify" menu ® Modify operand ® Enter value 6.0 ® OK ® Check)

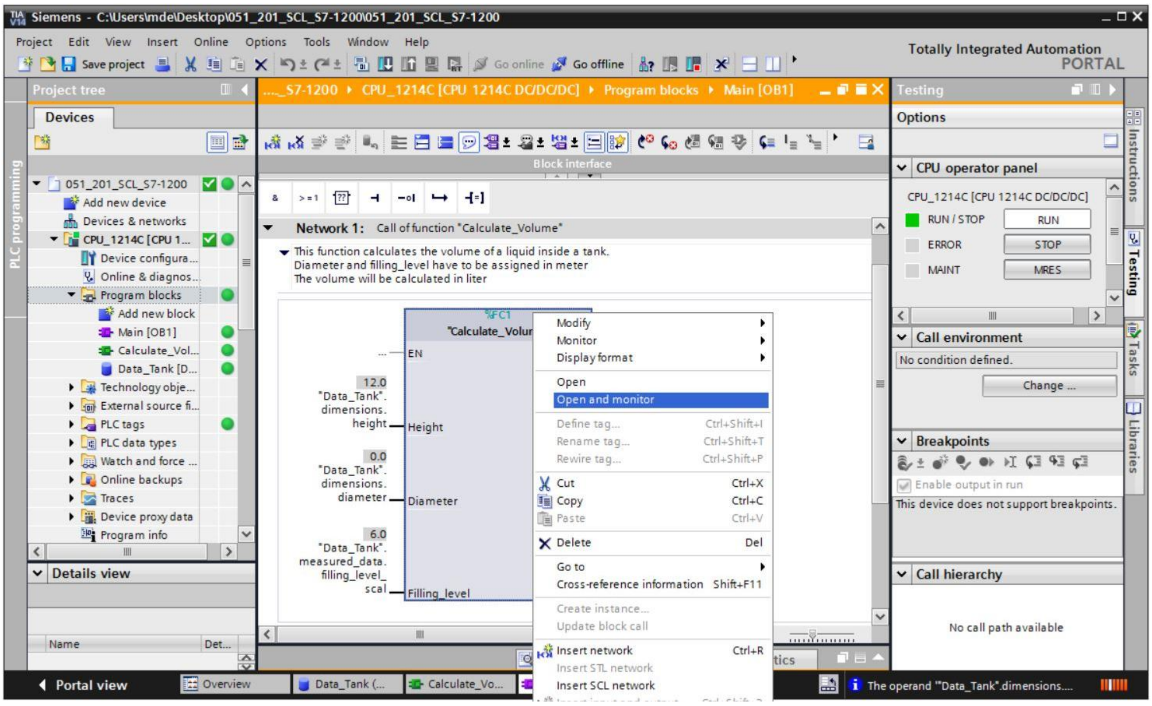


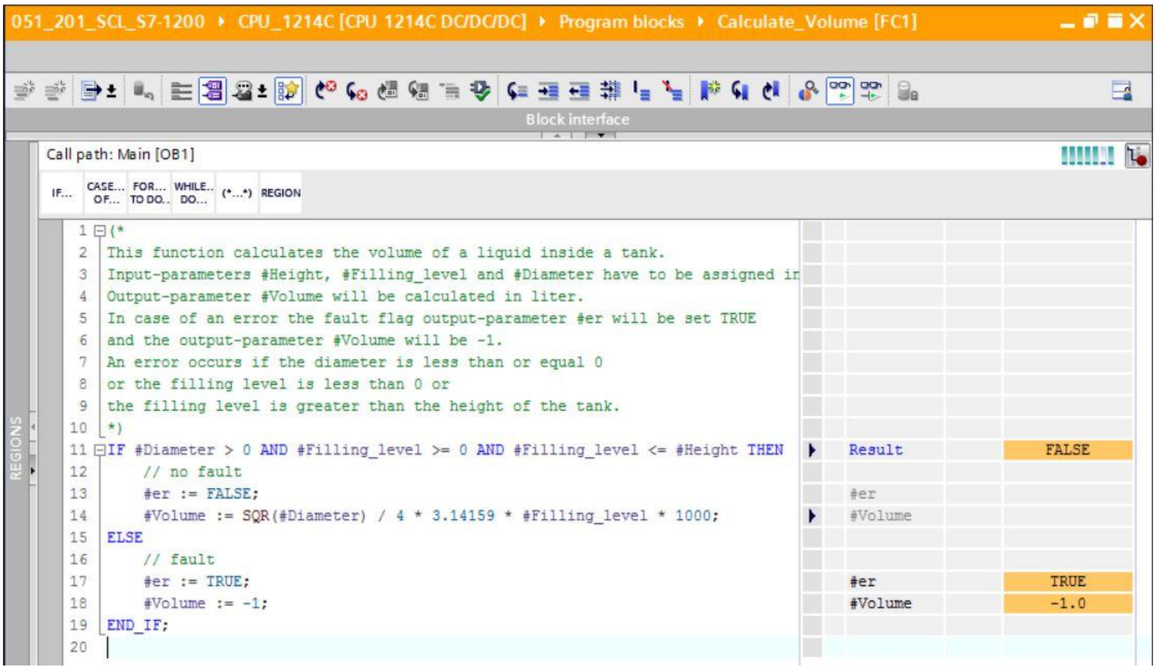
® Now test if an error is output by setting the diameter to zero.
(® Right-click on "Diameter" ® "Modify" menu ® Modify operand ® Enter value 0.0 ® OK ® Check)



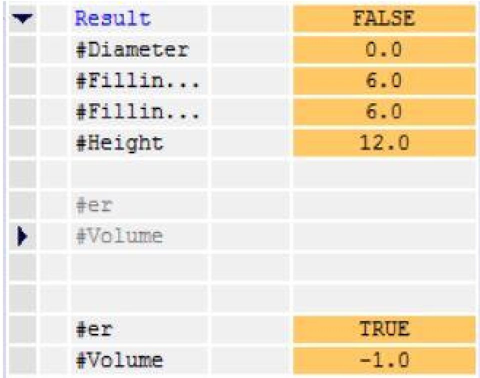
7.14 Monitoring and testing the "Calculate_Volume" function

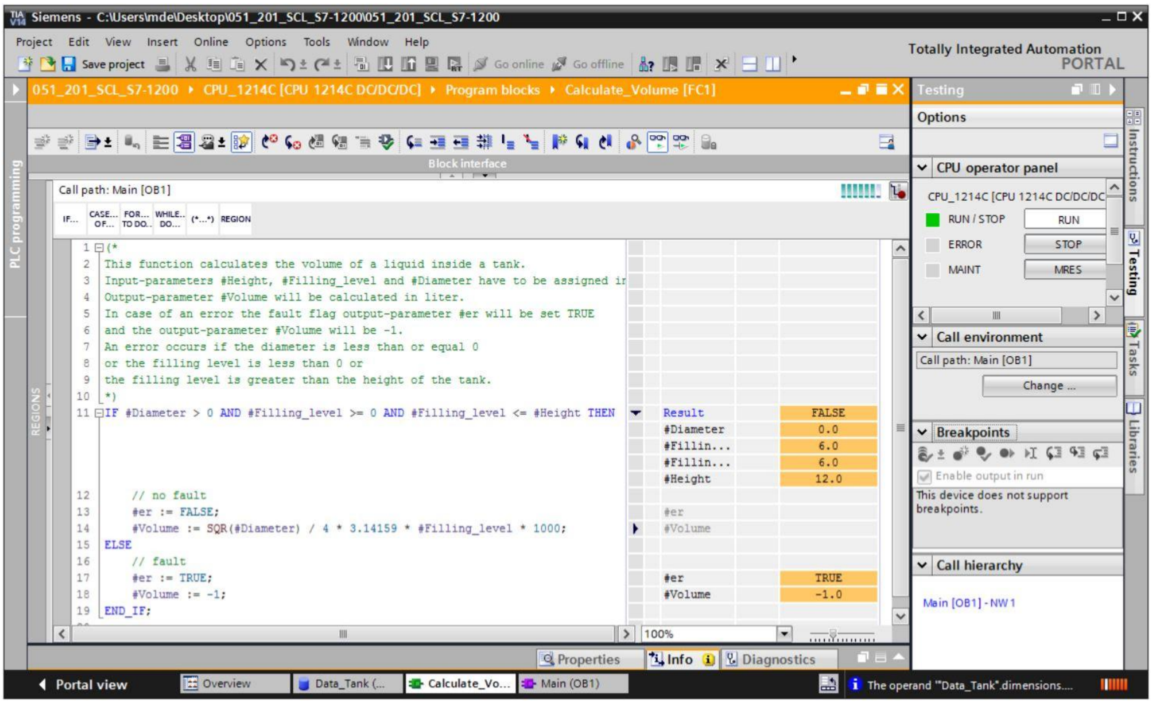
® Finally, open and monitor the "Calculate_Volume" function by right-clicking the function and selecting the "Open and monitor" menu command. (® Right-click on function ® Open and monitor)



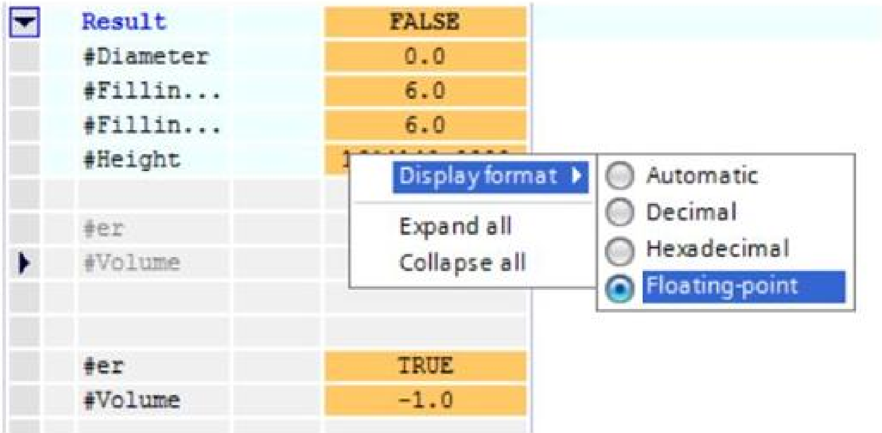


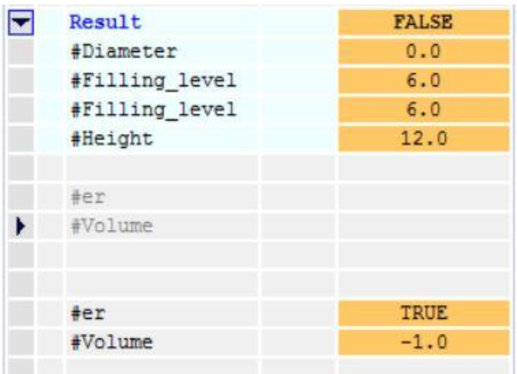
® You can show the values of the individual tags of the IF query by clicking the black arrow . (®)



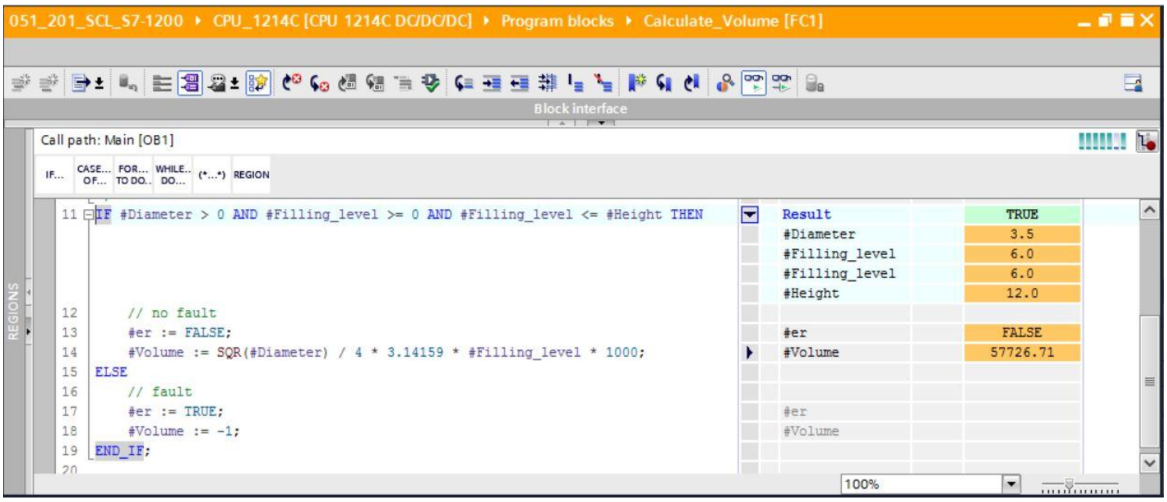


® Right-click the tag to adjust the display format. (® Right-click tag ® Display format ® Floating point)



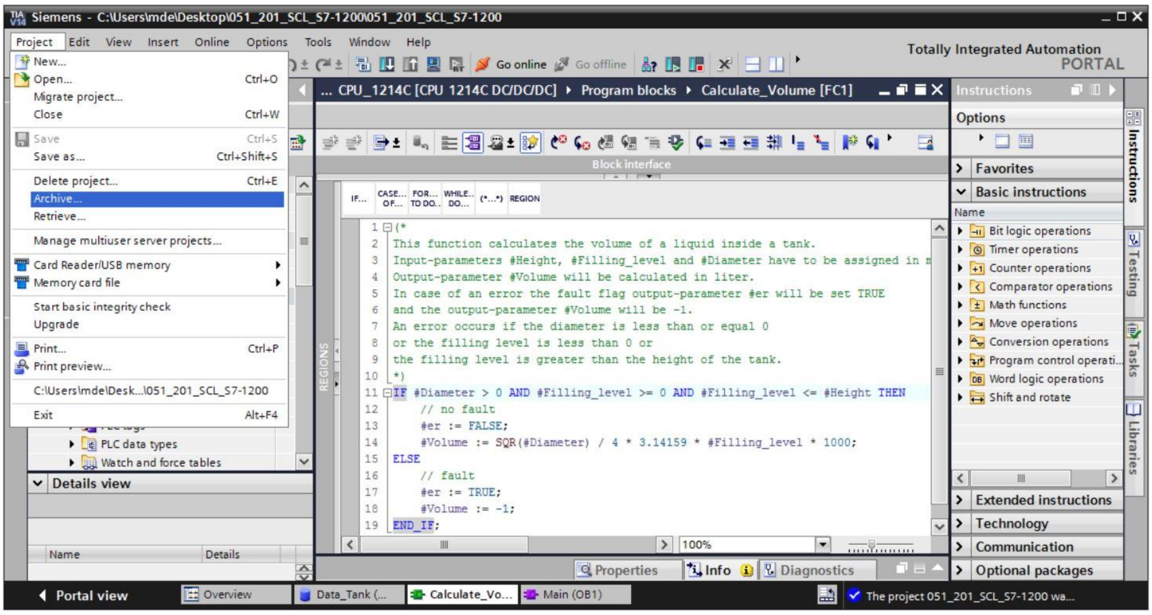


® Now test the other branch of the IF branch by modifying the diameter in OB1 back to 3.5 meters. (® Open OB1 ® Modify diameter to 3.5 ® Open and monitor function)



7.15 Archiving the project

® Finally, the complete project is to be archived. Select ® "Project" ® "Archive …" in the menu. Open the folder in which you want to archive your project and save it as file type "TIA Portal Project archive". (® Project ® Archive ® TIA Portal Project archive ® File name: SCE_EN_051-201 SCL_S7-1200… ® Archive)



8 Checklist

No.	Description	Checked
1	Successful compilation without error message	
2	Successful download without error message	
3	Modify operand (Diameter = 0.0) Result tag Volume= -1 Result tag "er" = TRUE	
4	Modify operand (Diameter = 3.5 and Level_scal = 0) Result Volume = 0 Result tag "er" = FALSE	
5	Modify operand (Filling_level_scal= 6.0) Result Volume = 57726.72 Result tag "er" = FALSE	
6	Modify operand (Filling_level_scal= 12.0) Result Volume = 115453.4 Result tag "er" = FALSE	
7	Modify operand (Filling_level_scal= 14.0) Result Volume = -1 Result tag "er" = TRUE	
8	Project successfully archived	

9 Exercise

9.1 Task description – Exercise

In this exercise you are going to program a "Scaling" function. The program is to be generally applicable to any positive analog values. In our example task "Tank", the filling level is read by an analog sensor and stored as a scaled value in the data block using this function.

In case of an error, the block is to set the error flag "er" to TRUE and set the parameter "Analog_scal" to zero as a result. An error exists when the "mx" parameter is less than or equal to "mn".

The function must contain the following parameters.

Input	**Data type**	**Comment**
Analog_per	INT	Analog value of the IO between 0..27648
mx	REAL	Maximum of the new scale
mn	REAL	Minimum of the new scale
Output		
er	BOOL	Error flag, no error = 0, error = 1
Analog_scal	REAL	Analog value scaled between mn..mx In case of an error = 0

The following formula is used to solve the task:

$$\#\text{Analog\_sca l} = \frac{\#\text{Analog\_per}}{27648} \cdot (\#\text{mx} - \#\text{mn}) + \#\text{mn}$$

An analog signal is required for this task. The operand used for this task must be entered in the PLC tag table.

Name	**Data type**	**Address**	**Comment**
B1	INT	%IW64	Filling level between 0..27648

9.2 Planning

Now solve this task on your own.

9.3 Checklist – Exercise

No.	Description	Checked
1	Operand added to PLC tag table	
2	Function FC: "Scaling" created	
3	Interface defined	
4	Function programmed	
5	"Scaling" function added to network 1 of OB1	
6	Input tags connected	
7	Output tags connected	
8	Successful compilation without error message	
9	Successful download without error message	
10	Analog value for filling level set to zero Result Filling_level_scal = 0 Result er = FALSE	
11	Analog value for filling level set to 27648 Result Filling_level_scal = 12.0 Result er = FALSE	
12	Analog value for filling level set to 13824 Result Filling_level_scal = 6.0 Result er = FALSE	
13	Modify operand (mx = 0.0) Result Filling_level_scal = 0 Result tag er = TRUE	
14	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:

siemens.com/sce/s7-1200

Preview "Additional information"

Getting Started, Videos, Tutorials, Apps, Manuals, Trial-SW/Firmware

- TIA Portal Videos
- TIA Portal Tutorial Center
- Getting Started
- Programming Guideline
- Easy Entry in SIMATIC S7-1200
- Download Trial Software/Firmware
- Technical Documentation SIMATIC Controller
- Industry Online Support App
- TIA Portal, SIMATIC S7-1200/1500 Overview
- TIA Portal Website
- SIMATIC S7-1200 Website
- SIMATIC S7-1500 Website

Further information

Siemens Automation Cooperates with Education
siemens.com/sce

SCE Training Curriculums
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