SIEMENS

SCE Training Curriculum

Siemens Automation Cooperates with Education | 09/2017

TIA Portal Module 052-201

High-Level Language Programming
with SCL and SIMATIC S7-1500

Cooperates with Education
Automation
SIEMENS

Matching SCE Trainer Packages for these Learn-/Training Document

- **SIMATIC S7 CPU 1516F-3 PN/DP**
  Order no.: 6ES7516-3FN00-4AB2
- **SIMATIC STEP 7 Professional V14 SP1 - Single license**
  Order no.: 6ES7822-1AA04-4YA5
- **SIMATIC STEP 7 Professional V14 SP1 - Classroom license (set of 6)**
  Order no.: 6ES7822-1BA04-4YA5
- **SIMATIC STEP 7 Professional V14 SP1 - Upgrade license (set of 6)**
  Order no.: 6ES7822-1AA04-4YE5
- **SIMATIC STEP 7 Professional V14 SP1 - Student license (set of 20)**
  Order no.: 6ES7822-1AC04-4YA5

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, contact your regional SCE representative: siemens.com/sce/contact

Additional information regarding SCE

siemens.com/sce

Notes on use

This SCE training curriculum for the integrated automation solution Totally Integrated Automation (TIA) was prepared for the program "Siemens Automation Cooperates with Education (SCE)" specifically for training purposes for public educational and R&D institutions. Siemens AG assumes no responsibility for the content.

This curriculum may be used only for initial education with respect to Siemens products/systems. That is, it may be copied in part or in whole and handed out to trainees for use within the framework of their education. Transmission and reproduction of this curriculum as well as communication of its content is permitted within public educational institutions for educational purposes. Any exceptions require written consent from Siemens AG. Contact: Roland Scheuerer roland.scheuerer@siemens.com.

Parties breaching this provision shall be liable for damages. All rights reserved, including those relating to translation and in particular those rights created as a result of a patent being granted or utility model being registered.

Use for industry customers is expressly prohibited. Commercial use of the curriculum is not permitted.

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

Table of contents

HIGH-LEVEL LANGUAGE PROGRAMMING WITH S7-SCL

1. Objective

In this section, you will learn more about the basic functions of the S7-SCL high-level language. Test functions for eliminating logical programming errors will also be presented.

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

2. Requirement

This section is based on the hardware configuration of SIMATIC S7 CPU1516F-3 PN/DP but can also be implemented with other hardware configurations that have digital and analog input and output cards. To implement this section, you can use the following project, for example:

SCE_EN_012_101_Hardwarekonfiguration_CPU1516F.zap13

You should also have basic knowledge of high-language programming, for example, using Pascal.

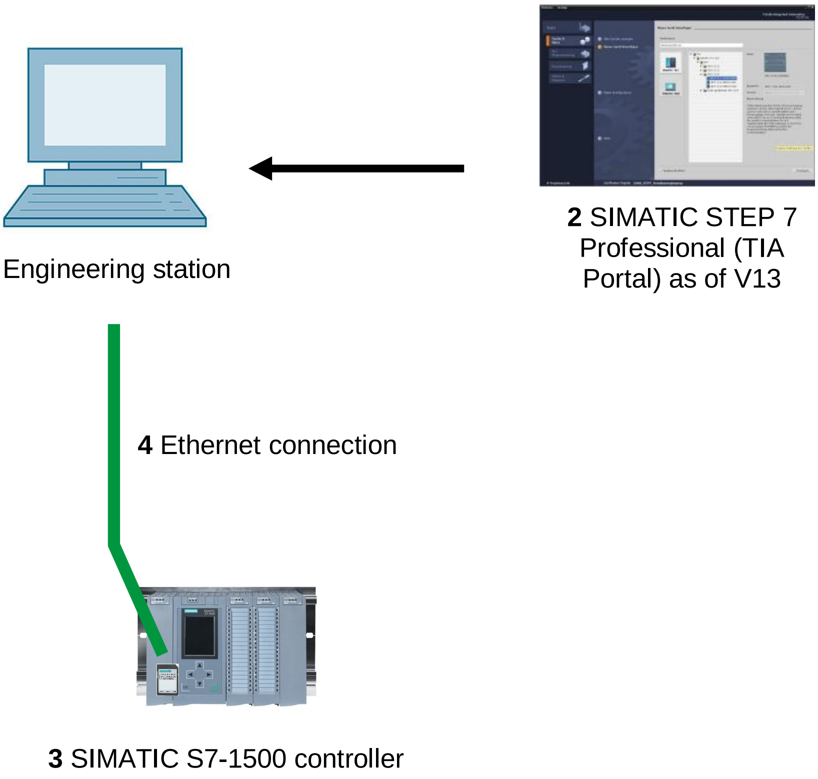
3. Required hardware and software

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

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

3 SIMATIC S7-1500/S7-1200/S7-300 controller, e.g. CPU 1516F-3 PN/DP – Firmware as of V1.6 with memory card and 16DI/16DO and 2AI/1AO

4 Ethernet connection between engineering station and controller


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

4. Theory

4.1 The S7-SCL programming language

S7-SCL (Structured Control Language) is a high-level, Pascal-based programming language that allows for structured programming. The language corresponds to the "Sequential Function Chart" (SFC) language specified in the standard DIN EN-61131-3 (IEC 61131-3). In addition to high-level language elements, S7-SCL also includes typical PLC elements as language elements, such as inputs, outputs, timers, bit memories, block calls, etc. S7-SCL particularly supports the STEP 7 block concept and, in addition to the statement list (STL), ladder logic (LAD) and function block diagram (FBD), S7-SCL makes it possible to program blocks that conform to the standards. This means S7-SCL supplements and expands the STEP 7 programming software with its programming languages LAD, FBD and STL.

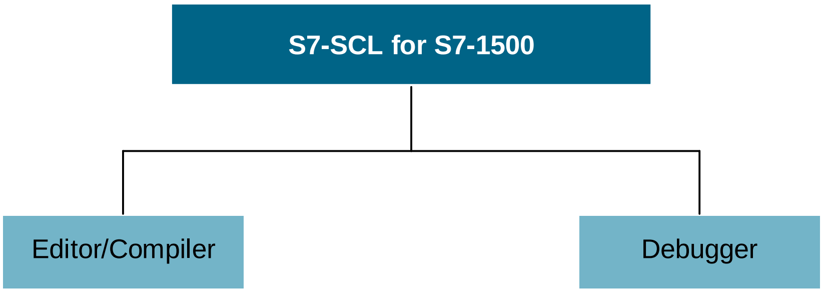
You do not have to create every function yourself but can use precompiled blocks, such as system functions and system function blocks that are present in the CPU's operating system.

Blocks that are programmed with S7-SCL can be mixed with STL, LAD and FBD blocks. This means that a block programmed with S7-SCL can call another block that is programmed in STL, LAD or FBD. Accordingly, S7-SCL blocks can also be called in STL, LAD and FBD programs.

The S7-SCL test functions make it possible to search for logical programming errors in an error-free compilation.

4.2 The S7-SCL development environment

With regard to the use of S7-SCL, there is a development environment that is tailored to the specific properties of both S7-SCL and STEP 7. This development environment consists of an editor/compiler and a debugger.



Editor/Compiler

The S7-SCL editor is a text editor that can be used to edit any kind of text. The central task of the S7-SCL editor is creating and editing 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 language S7-SCL.
- Convenient adding of language elements and block calls with drag & drop.
- Direct syntax check during programming.
- Customization of the editor to meet your needs, e.g. colors 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 S7-SCL debugger enables you to check a program while it is running in the automation system (AS) and thus find any potential logical errors.

S7-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 execute 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 programming an error evaluation.

6. Planning

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

Global parameters, such as the diameter and the height of the tank, shall be stored in a "Data_Tank" data block.

The program for calculation of the content shall be written in a "Calculate_Volume" function and the parameters shall 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	range 0...27648
filling_level_scal	REAL	0.0	range 0...12.0
volume_liquid	REAL	0.0	in liter
fault_flags	STRUCT		
calculate_volume	BOOL		fault == true

Table 1: Parameters in the "Data_Tank" data block

6.2 "Calculate_Volume" function

This block calculates the content of the tank in liters.

In the first step, the transferred parameters are not to be checked for reasonableness.

The following parameters are required for this step:

Input	**Data type**	**Comment**
Diameter	REAL	diameter 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. You also want to check whether the filling level is greater than or equal to zero or less than or greater than the height of the tank. In case of an error, the new parameter "er" is set to TRUE, and the parameter "Volume" is set to the value -1.

To do so, expand the interface by the parameters "er" and "Height".

Input	**Data type**	**Comment**
Height	REAL	height cylindric tank in meter
Diameter	REAL	diameter 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

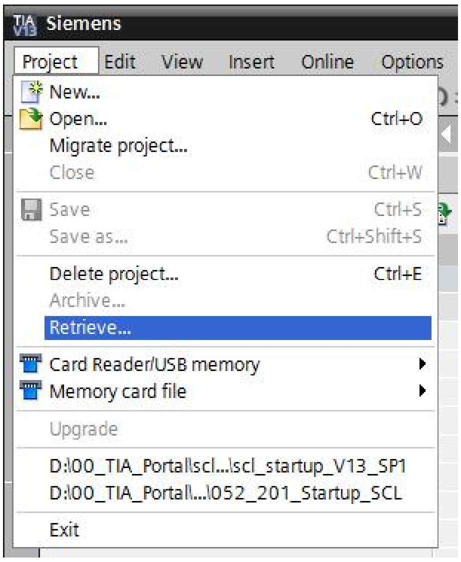
Table 3: Parameters for "Calculate_Volume" function in the second step

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_012-101_Hardware_configuration_S7-1516F_....zap).
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)

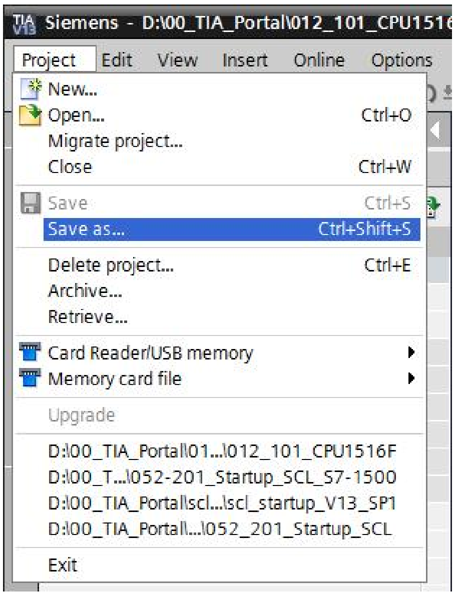


® Next you can select the target directory to which you want to save the retrieved project. Confirm your selection with "OK".
(® Project ® Save as... ® OK)

7.2 Saving the project under a new name

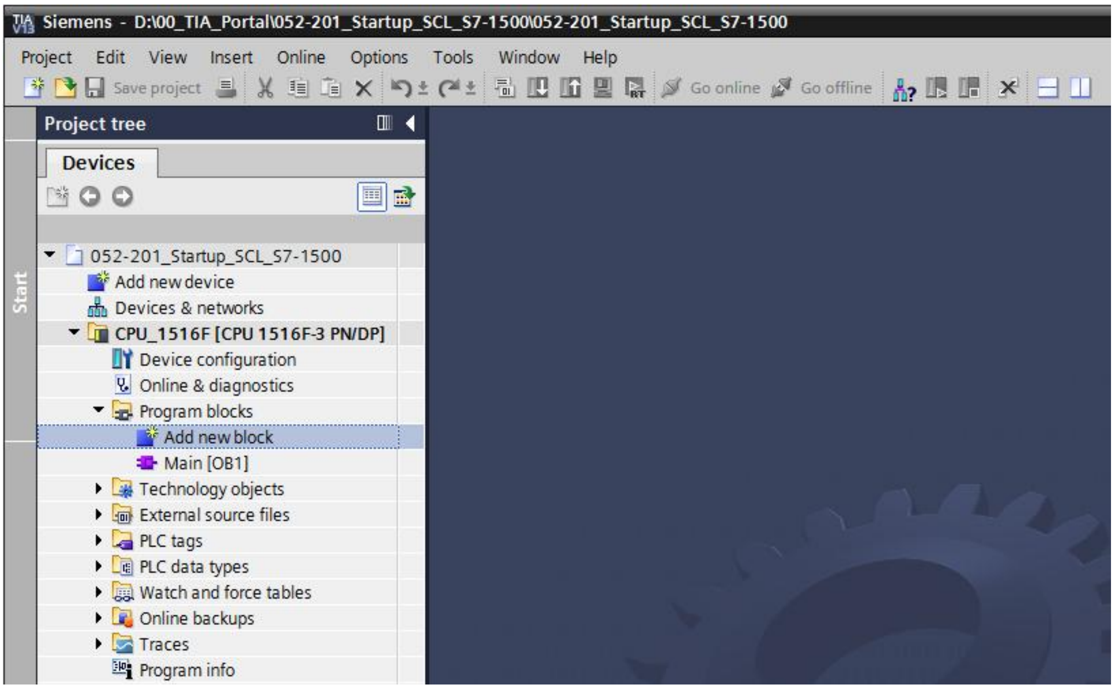
® You save the opened project under the name 052-201_Startup_SCL.

(® Project ® Save as … ® 052-201_Startup_SCL ® Save)



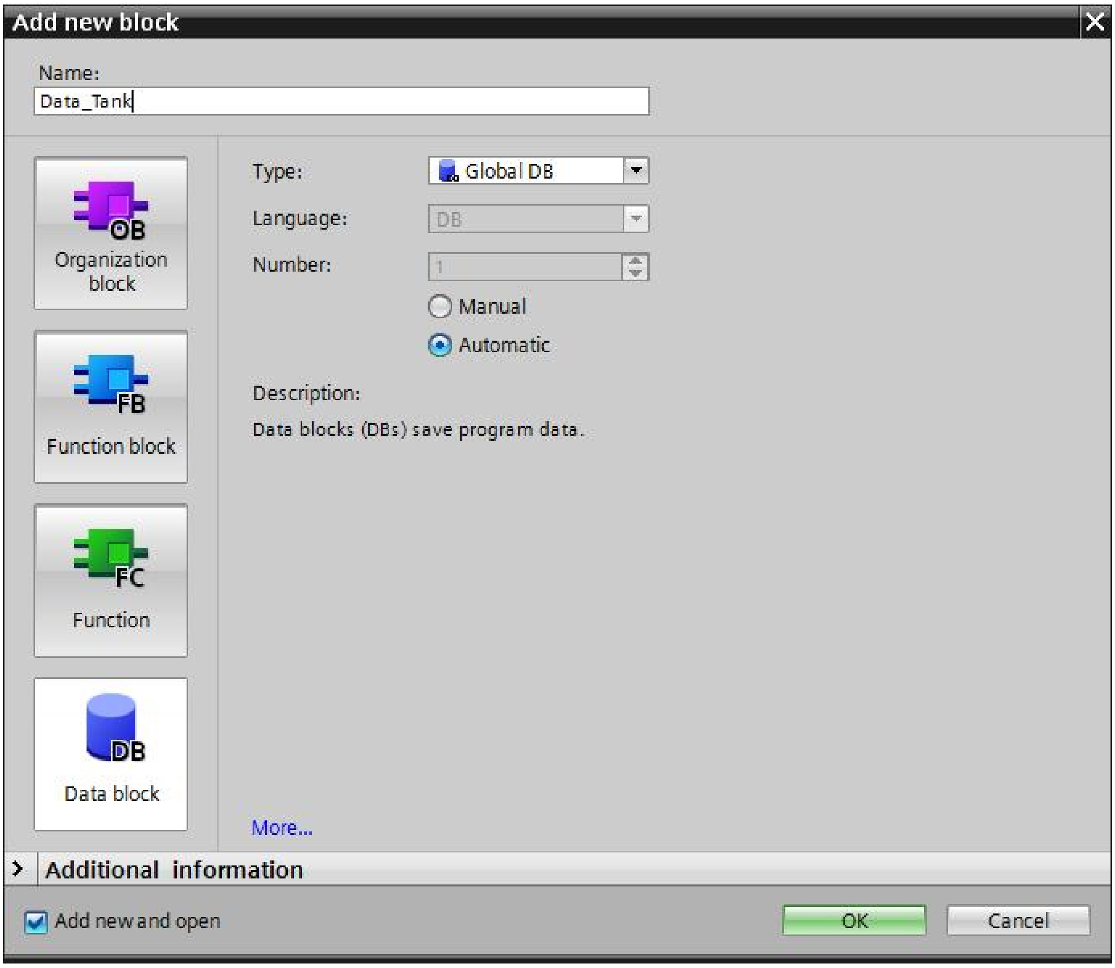
7.3 Creating the "Data_Tank" data block

® In the Project view, go to ® Program blocks and create a new bock with a double-click on ® Add new block.



® Now select a data block and enter the name.

(® DB ® "Data_Tank" ® OK)



Add new block

Name: Data_Tank

Type: Global DB

Language: DB

Number: 1

○ Manual

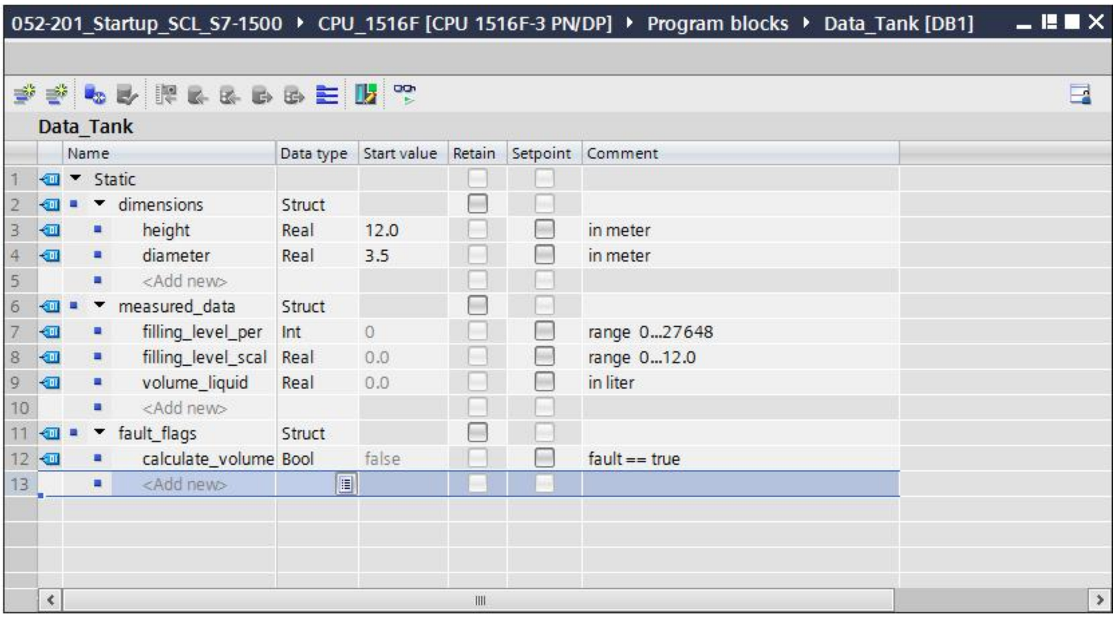
● Automatic

Description:

Data blocks (DBs) save program data.

☑ Add new and open

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



052-201_Startup_SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Data_Tank [DB1]

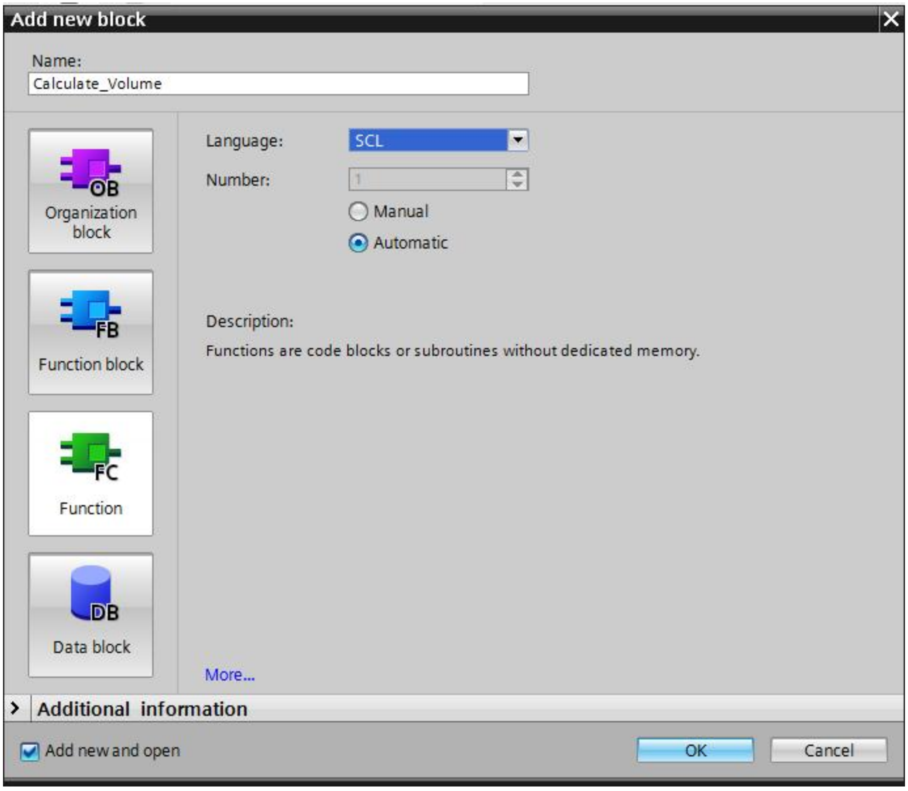
Data_Tank

	Name	Data type	Start value	Retain	Setpoint	Comment
1	▾ Static			□	□	
2	▾ dimensions	Struct		□	□	
3	height	Real	12.0	□	□	in meter
4	diameter	Real	3.5	□	□	in meter
5	<Add new>			□	□	
6	▾ measured_data	Struct		□	□	
7	filling_level_per	Int	0	□	□	range 0...27648
8	filling_level_scal	Real	0.0	□	□	range 0...12.0
9	volume_liquid	Real	0.0	□	□	in liter
10	<Add new>			□	□	
11	▾ fault_flags	Struct		□	□	
12	calculate_volume	Bool	false	□	□	fault == true
13	<Add new>			□	□	

7.4 Creating the "Calculate_Volume" function

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

(® Add new block ® FC Funktion ® "Calculate_Volume" ® SCL ® OK)



Add new block

Name: Calculate_Volume

Language: SCL

Number: 1

○ Manual

● Automatic

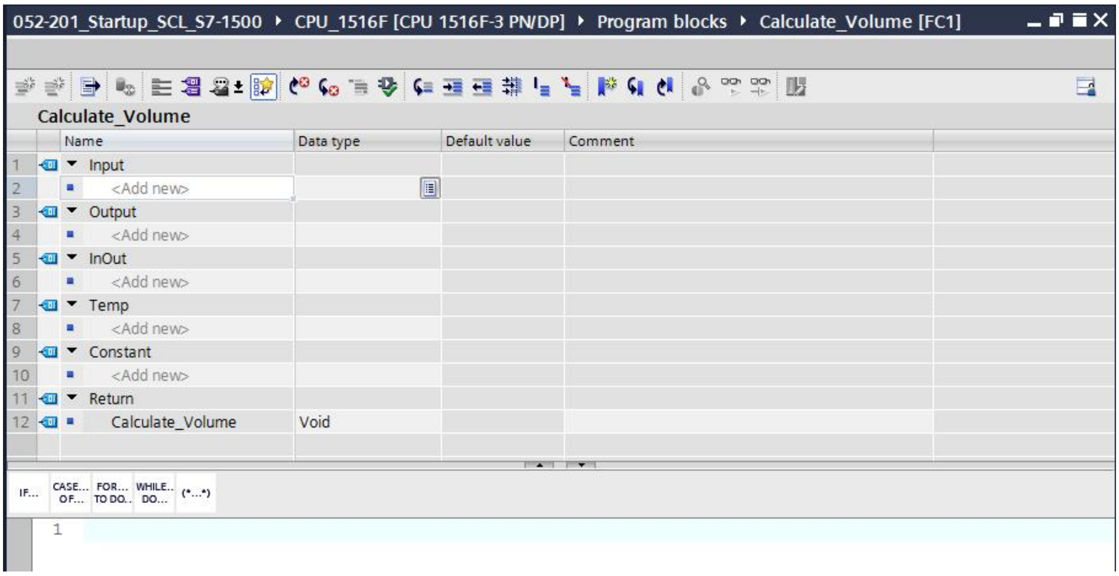
Description:

Functions are code blocks or subroutines without dedicated memory.

☑ Add new and open

7.5 Specifying the interface of the "Calculate_Volume" function

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



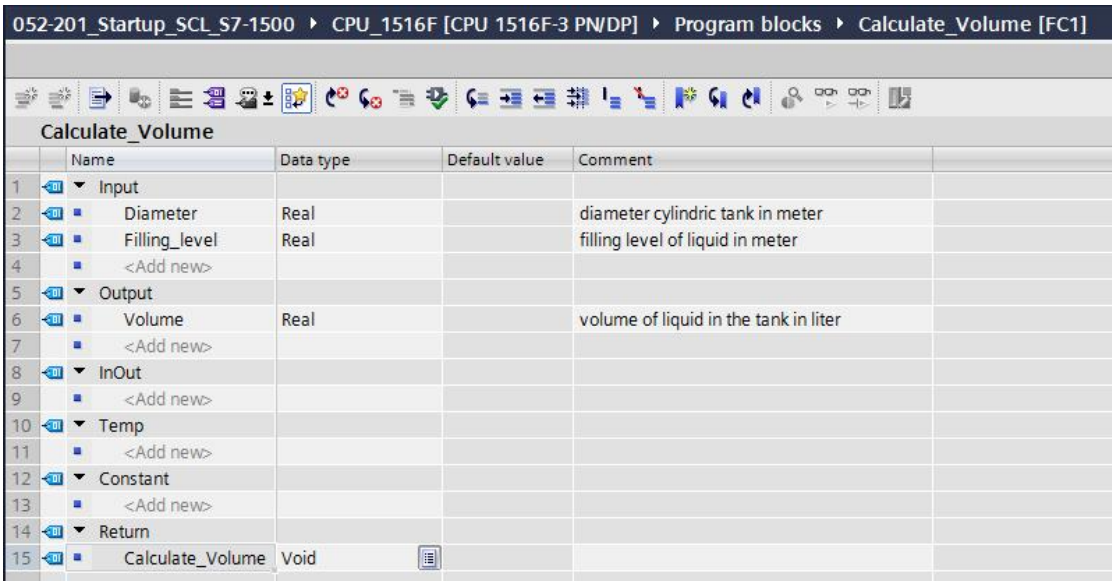
052-201_Startup_SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Calculate_Volume [FC1]

Calculate_Volume

	Name	Data type	Default value	Comment
1	▼ Input			
2	<Add new>			
3	▼ Output			
4	<Add new>			
5	▼ InOut			
6	<Add new>			
7	▼ Temp			
8	<Add new>			
9	▼ Constant			
10	<Add new>			
11	▼ Return			
12	Calculate_Volume	Void		

® Create the following input and output parameters.

(® Name ® Data type ® Comment)



052-201_Startup_SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Calculate_Volume [FC1]

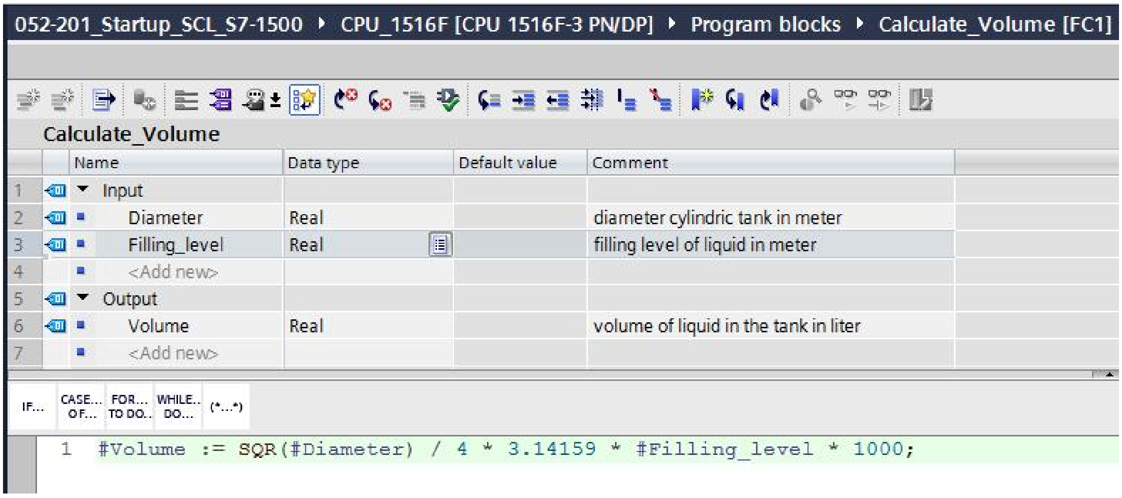
Calculate_Volume

	Name	Data type	Default value	Comment
1	▼ Input			
2	Diameter	Real		diameter cylindric tank in meter
3	Filling_level	Real		filling level of liquid in meter
4	<Add new>			
5	▼ Output			
6	Volume	Real		volume of liquid in the tank in liter
7	<Add new>			
8	▼ InOut			
9	<Add new>			
10	▼ Temp			
11	<Add new>			
12	▼ Constant			
13	<Add new>			
14	▼ Return			
15	Calculate_Volume	Void		

7.6 Programming the "Calculate_Volume" function

® Enter the program shown below.

(® Enter program)



052-201_Startup_SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Calculate_Volume [FC1]

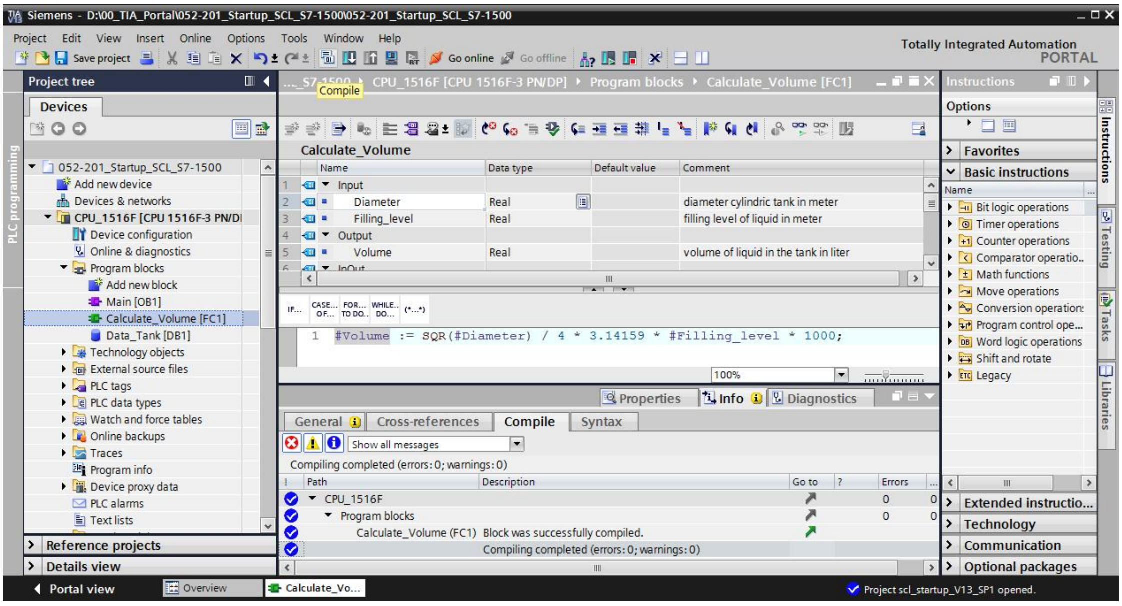
Calculate_Volume

	Name	Data type	Default value	Comment
1	▼ Input			
2	Diameter	Real		diameter cylindric tank in meter
3	Filling_level	Real		filling level of liquid in meter
4	<Add new>			
5	▼ Output			
6	Volume	Real		volume of liquid in the tank in liter
7	<Add new>			

```
#Volume := SQR(#Diameter) / 4 * 3.14159 * #Filling_level * 1000;
```

® 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.

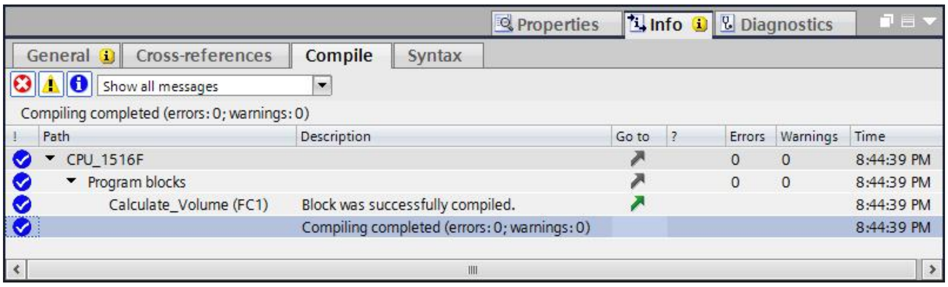
(® ® Eliminate errors® Save project)



Siemens - D:\00_TIA_Portal\052-201_Startup_SCL_S7-1500\052-201_Startup_SCL_S7-1500

```
#Volume := SQR(#Diameter) / 4 * 3.14159 * #Filling_level * 1000;
```

Compiling completed (errors: 0; warnings: 0)

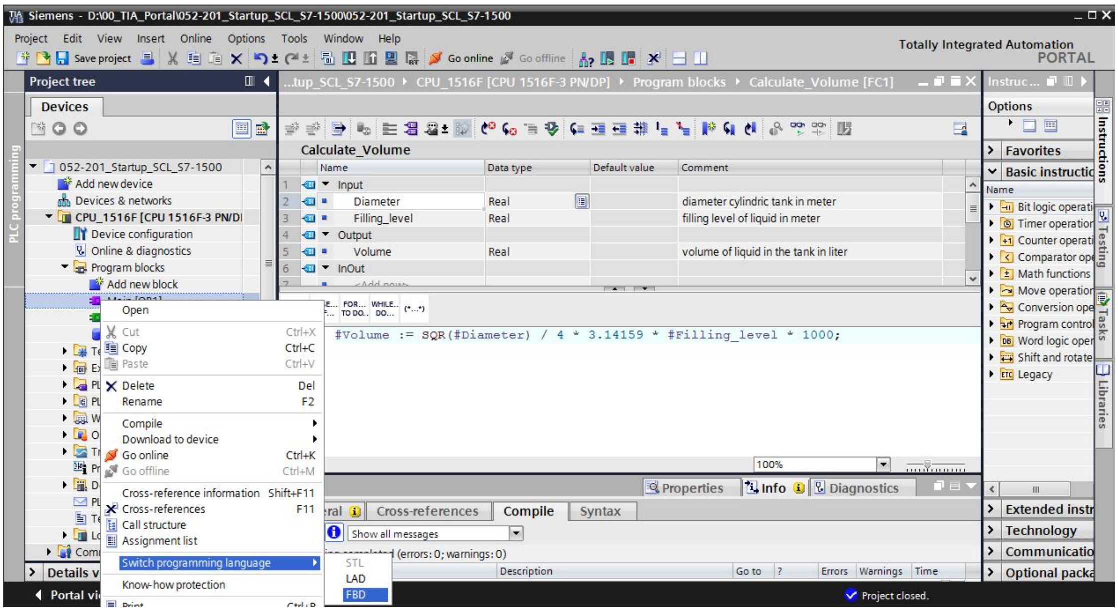


Compiling completed (errors: 0; warnings: 0)

!	Path	Description	Go to	?	Errors	Warnings	Time
✓	▼ CPU_1516F		↗		0	0	8:44:39 PM
✓	▼ Program blocks		↗		0	0	8:44:39 PM
✓	Calculate_Volume (FC1)	Block was successfully compiled.	↗				8:44:39 PM
✓		Compiling completed (errors: 0; warnings: 0)					8:44:39 PM

7.7 Programming the "Main [OB1]" organization block

® Before programming the "Main [OB1]" organization block, switch the programming language to FBD. To do so, left-click "Main [OB1]" in the "Program blocks" folder. (® CPU_1516F[CPU 1516F-3 PN/DP] ® Program blocks ® Main [OB1] ® Switch programming language ® FBD)



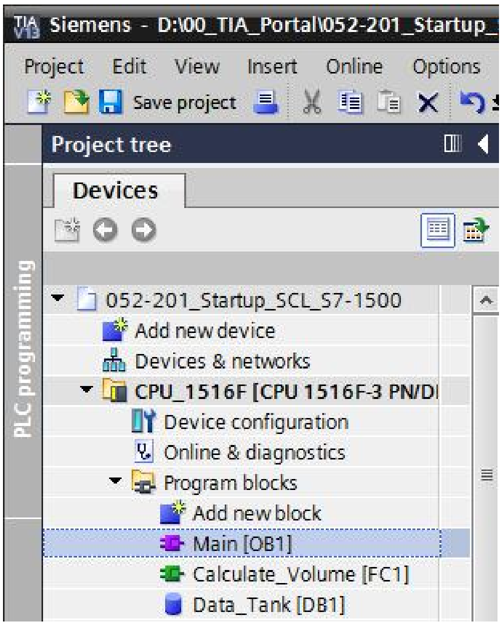
Calculate_Volume

	Name	Data type	Default value	Comment
1	Input			
2	Diameter	Real		diameter cylindric tank in meter
3	Filling_level	Real		filling level of liquid in meter
4	Output			
5	Volume	Real		volume of liquid in the tank in liter
6	InOut			
	<Add new>			

```
#Volume := SQR(#Diameter) / 4 * 3.14159 * #Filling_level * 1000;
```

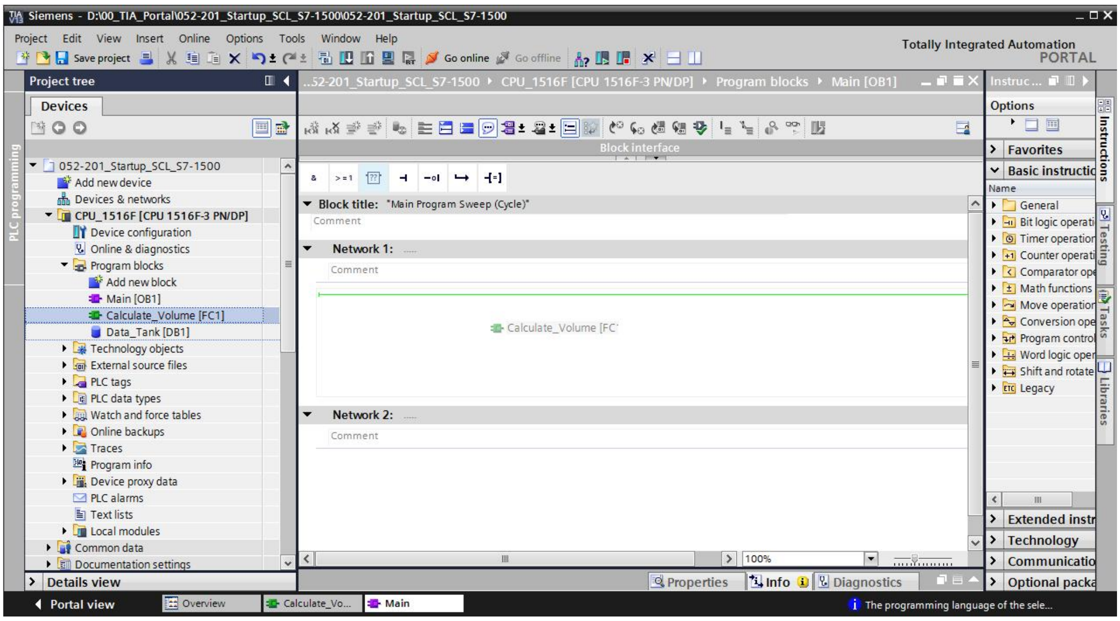
(errors: 0; warnings: 0)

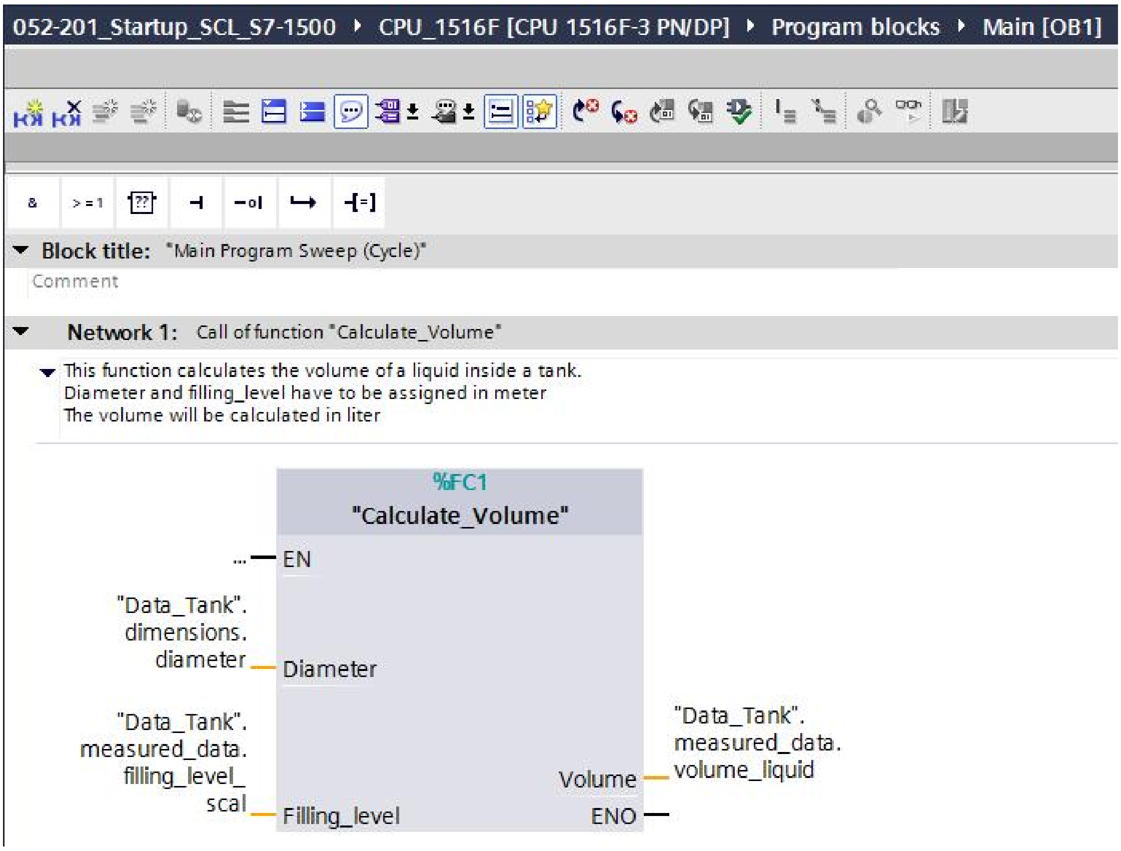
® Now double-click the "Main [OB1]" organization block to open it.



® Call the "Calculate_Volume" function in the first network. Assign network title, comment and connect the parameters. Then save your project.

(® Call "Calculate_Volume" ® Assign network title ® Write network comment ® Connect parameters ® Save project)




052-201_Startup_SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Main [OB1]
& >=1 ??
Comment
%FC1
EN
ENO

Block title: "Main Program Sweep (Cycle)"

Network 1: Call of function "Calculate_Volume"

This function calculates the volume of a liquid inside a tank.
Diameter and filling_level have to be assigned in meter
The volume will be calculated in liter

"Calculate_Volume"

"Data_Tank".dimensions.diameter — Diameter

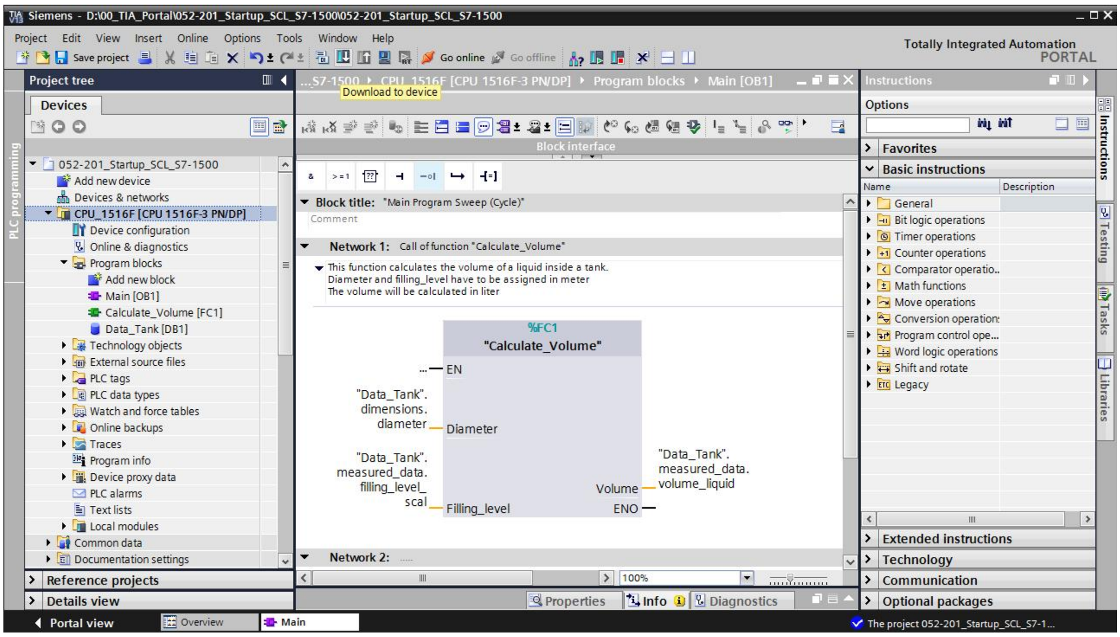
"Data_Tank".measured_data.filling_level_scal — Filling_level

Volume — "Data_Tank".measured_data.volume_liquid

7.8 Compiling and downloading the program

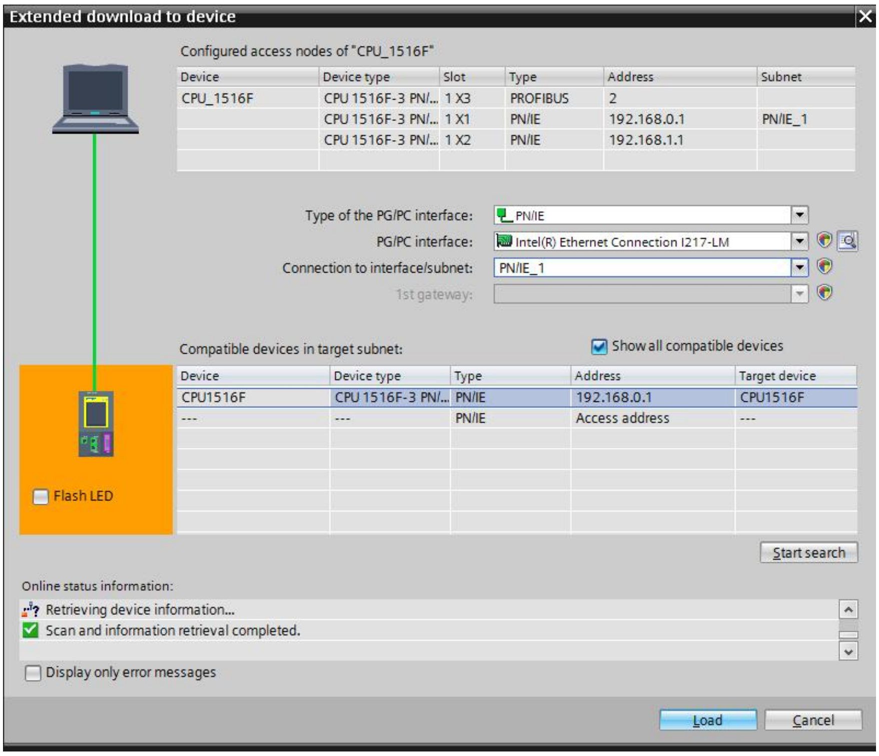
® Click the "Program blocks" folder and compile the entire program. After successful compilation, download the project to the PLC.

(® ®)



This function calculates the volume of a liquid inside a tank.
Diameter and filling_level have to be assigned in meter
The volume will be calculated in liter

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



Extended download to device

Configured access nodes of "CPU_1516F"

Device	Device type	Slot	Type	Address	Subnet
CPU_1516F	CPU 1516F-3 PN/...	1 X3	PROFIBUS	2	
	CPU 1516F-3 PN/...	1 X1	PN/IE	192.168.0.1	PN/IE_1
	CPU 1516F-3 PN/...	1 X2	PN/IE	192.168.1.1	

Type of the PG/PC interface: PN/IE

PG/PC interface: Intel(R) Ethernet Connection I217-LM

Connection to interface/subnet: PN/IE_1

1st gateway:

Compatible devices in target subnet: ☑ Show all compatible devices

Device	Device type	Type	Address	Target device
CPU1516F	CPU 1516F-3 PN/...	PN/IE	192.168.0.1	CPU1516F
---	---	PN/IE	Access address	---

□ Flash LED

Start search

Online status information:

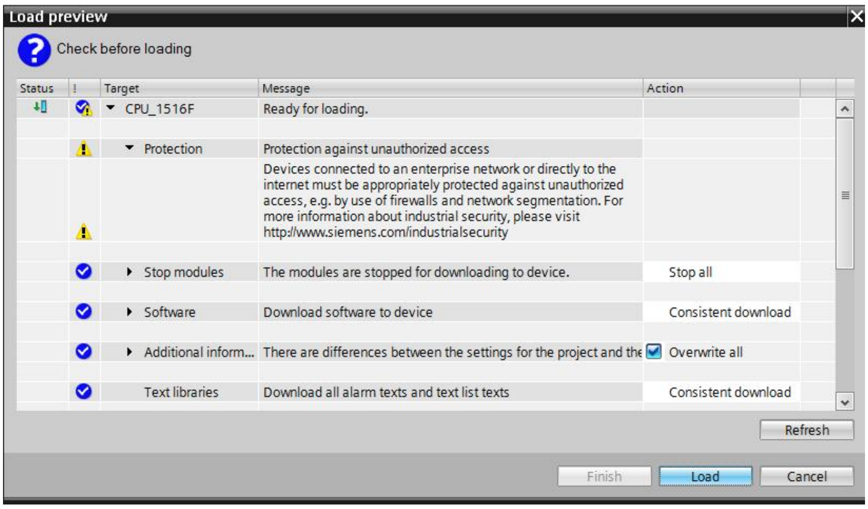
Retrieving device information...

✅ Scan and information retrieval completed.

□ Display only error messages

Load Cancel

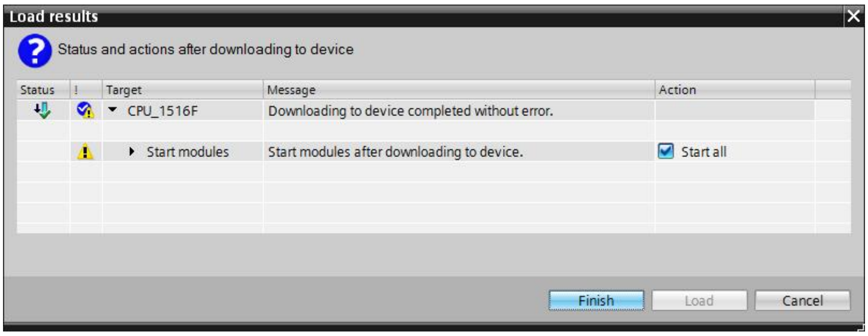
® Make selection, if necessary ® Load



Check before loading

Status	!	Target	Message	Action
		CPU_1516F	Ready for loading.	
		Protection	Protection against unauthorized access	
			Devices connected to an enterprise network or directly to the internet must be appropriately protected against unauthorized access, e.g. by use of firewalls and network segmentation. For more information about industrial security, please visit http://www.siemens.com/industrialsecurity	
		Stop modules	The modules are stopped for downloading to device.	Stop all
		Software	Download software to device	Consistent download
		Additional inform...	There are differences between the settings for the project and the	☑ Overwrite all
		Text libraries	Download all alarm texts and text list texts	Consistent download

® Finish

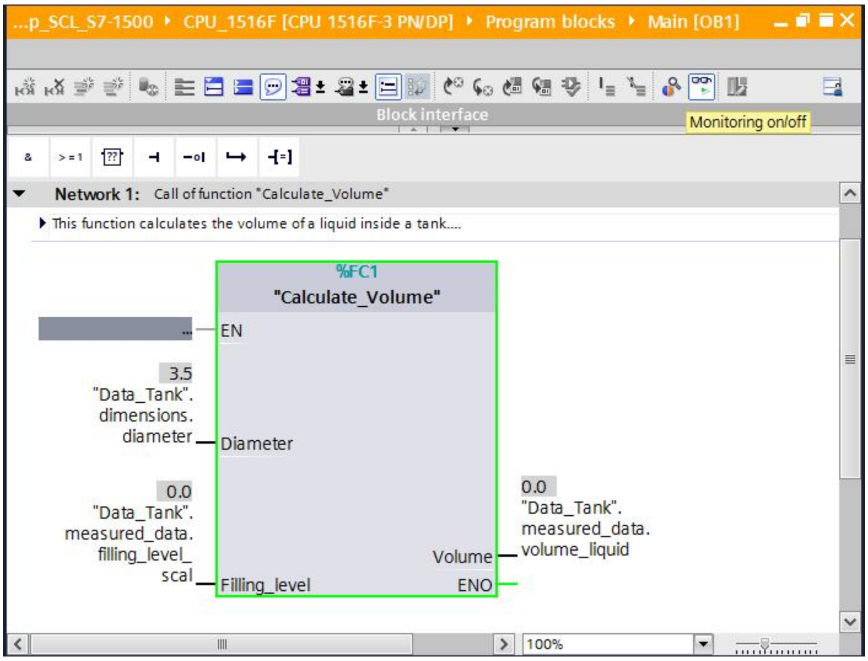


Status and actions after downloading to device

Status	!	Target	Message	Action
		CPU_1516F	Downloading to device completed without error.	
		Start modules	Start modules after downloading to device.	☑ Start all

7.9 Monitoring and testing the organization block

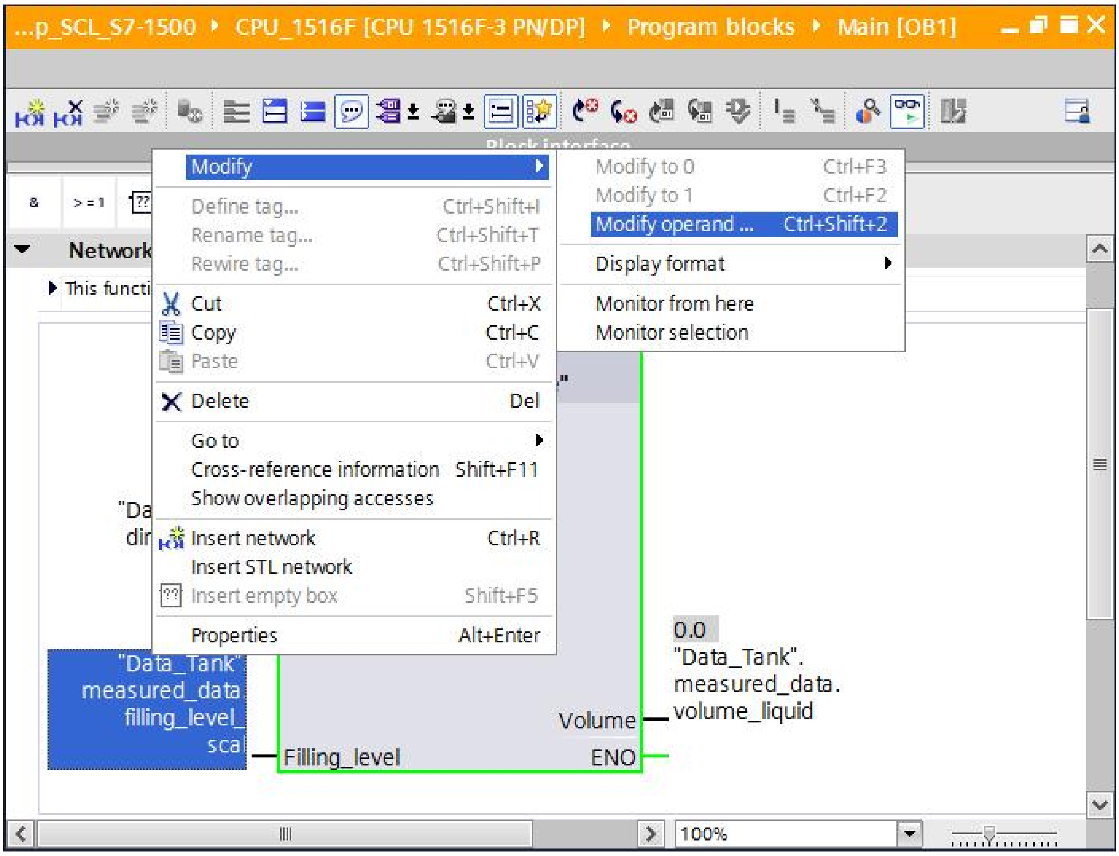
® In the open OB1 click the icon to monitor the block.


...p_SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Main [OB1]
Block interface
Monitoring on/off
& >=1 ??
%FC1
"Calculate_Volume"
EN
3.5
"Data_Tank". dimensions. diameter
Diameter
0.0
"Data_Tank". measured_data. filling_level_scal
Filling_level
Volume
ENO
0.0
"Data_Tank". measured_data. volume_liquid
100%

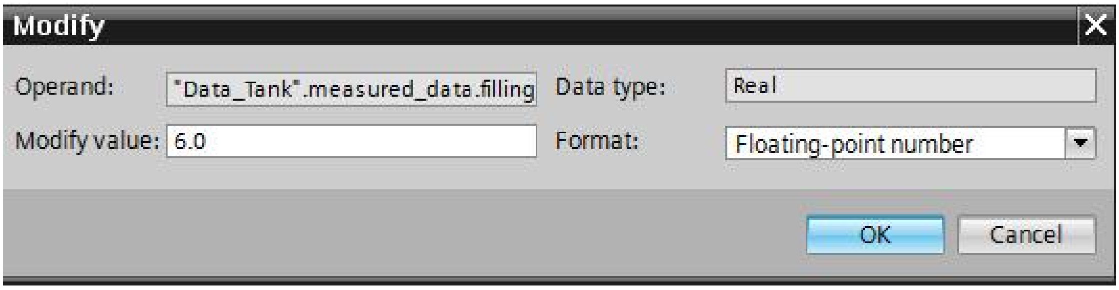
Network 1: Call of function "Calculate_Volume"

This function calculates the volume of a liquid inside a tank....

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)


...p_SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Main [OB1]
Network
This functi
Modify
Define tag... Ctrl+Shift+I
Rename tag... Ctrl+Shift+T
Rewire tag... Ctrl+Shift+P
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Go to
Cross-reference information Shift+F11
Show overlapping accesses
Insert network Ctrl+R
Insert STL network
Insert empty box Shift+F5
Properties Alt+Enter
Modify to 0 Ctrl+F3
Modify to 1 Ctrl+F2
Modify operand ... Ctrl+Shift+2
Display format
Monitor from here
Monitor selection
"Data_Tank".measured_data.filling_level_scal
Filling_level
Volume
ENO
0.0
"Data_Tank".measured_data.volume_liquid
100%

® Enter value 6.0 ® OK


Modify

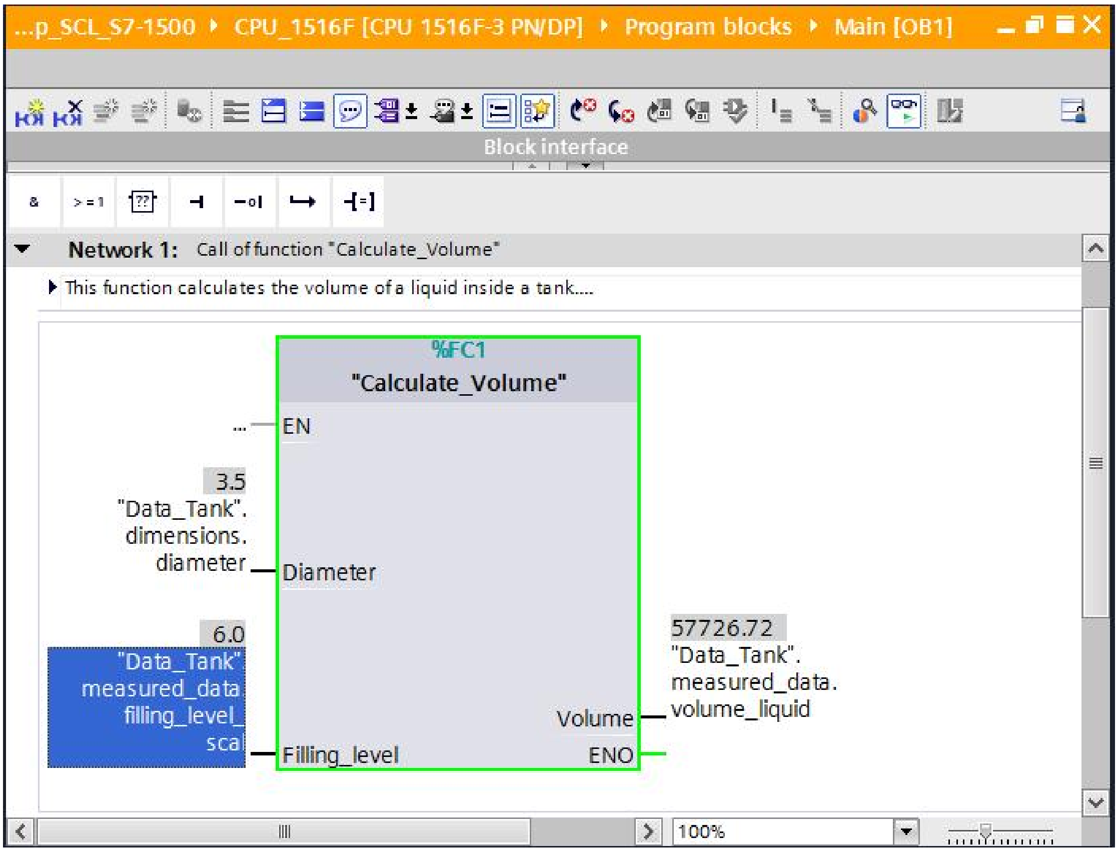
Operand: "Data_Tank".measured_data.filling

Data type: Real

Modify value: 6.0

Format: Floating-point number

® Check the result for correctness.

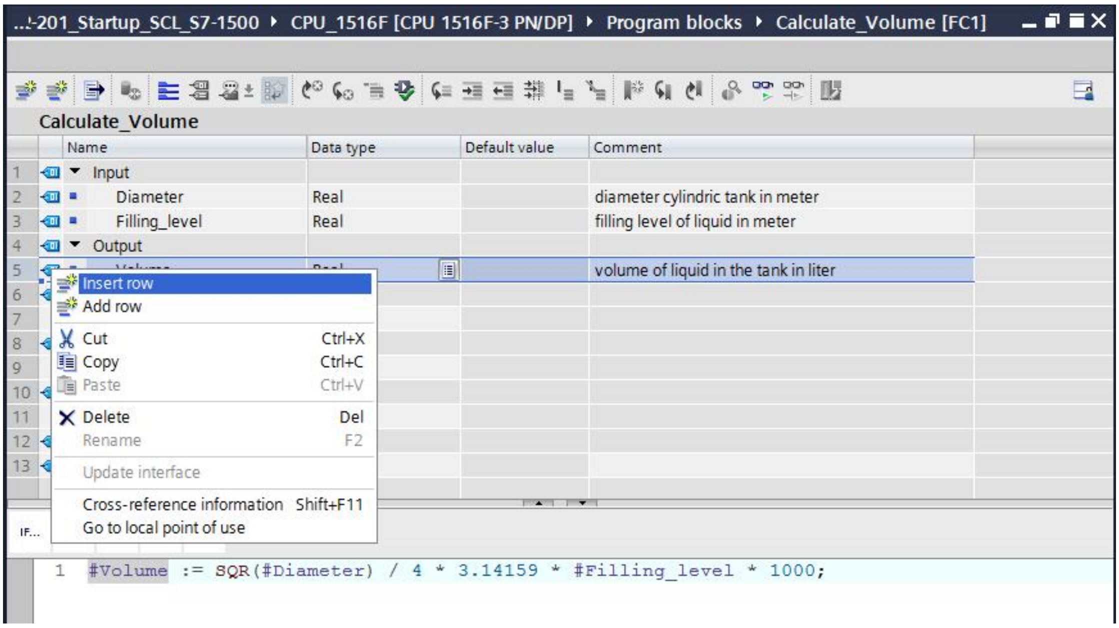


...p_SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Main [OB1]
Block interface
&
>=1
Network 1: Call of function "Calculate_Volume"
This function calculates the volume of a liquid inside a tank....
%FC1
"Calculate_Volume"
EN
3.5
"Data_Tank".dimensions.diameter
Diameter
6.0
"Data_Tank".measured_data.filling_level_scal
Filling_level
57726.72
"Data_Tank".measured_data.volume_liquid
Volume
ENO
100%

7.10 Expansion of the "Calculate_Volume" function

® Open the "Calculate_Volume" function, and insert a row in the output parameters with a right-click on the row in the interface.

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

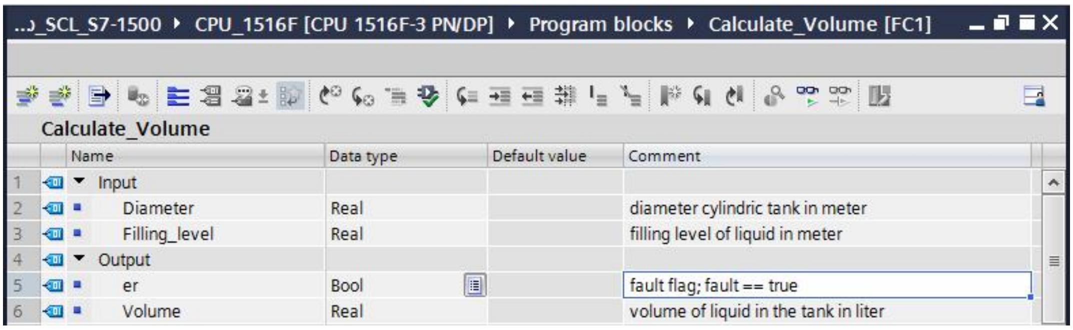


...-201_Startup_SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Calculate_Volume [FC1]
Calculate_Volume
Insert row
Add row
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Delete Del
Rename F2
Update interface
Cross-reference information Shift+F11
Go to local point of use
IF...

	Name	Data type	Default value	Comment
1	Input			
2	Diameter	Real		diameter cylindric tank in meter
3	Filling_level	Real		filling level of liquid in meter
4	Output			
5	Volume	Real		volume of liquid in the tank in liter

```
#Volume := SQR(#Diameter) / 4 * 3.14159 * #Filling_level * 1000;
```

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

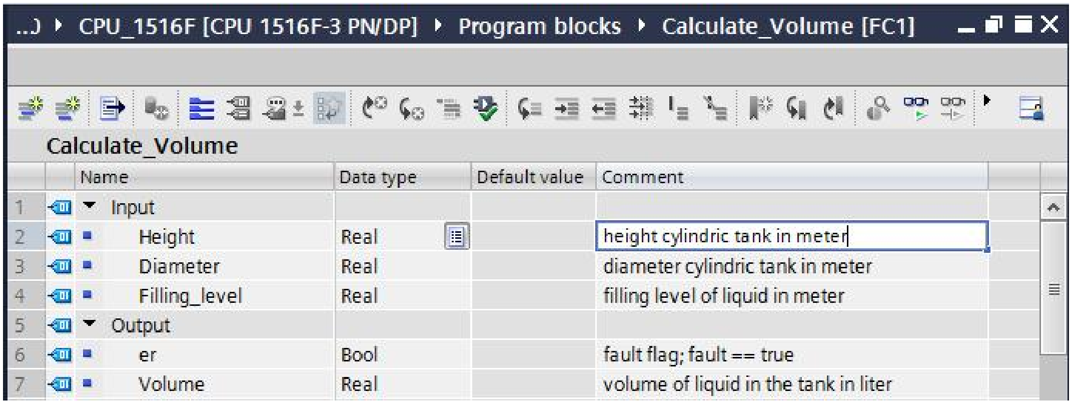


...J_SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Calculate_Volume [FC1]

Calculate_Volume

	Name	Data type	Default value	Comment
1	▼ Input			
2	Diameter	Real		diameter cylindric tank in meter
3	Filling_level	Real		filling level of liquid in meter
4	▼ Output			
5	er	Bool		fault flag; fault == true
6	Volume	Real		volume of liquid in the tank in liter

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



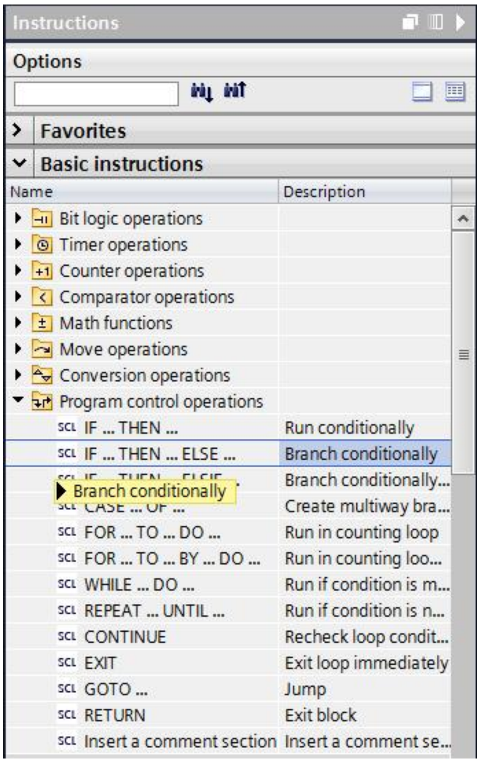
...J ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Calculate_Volume [FC1]

Calculate_Volume

	Name	Data type	Default value	Comment
1	▼ Input			
2	Height	Real		height cylindric tank in meter
3	Diameter	Real		diameter cylindric tank in meter
4	Filling_level	Real		filling level of liquid in meter
5	▼ Output			
6	er	Bool		fault flag; fault == true
7	Volume	Real		volume of liquid in the tank in liter

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

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



Instructions

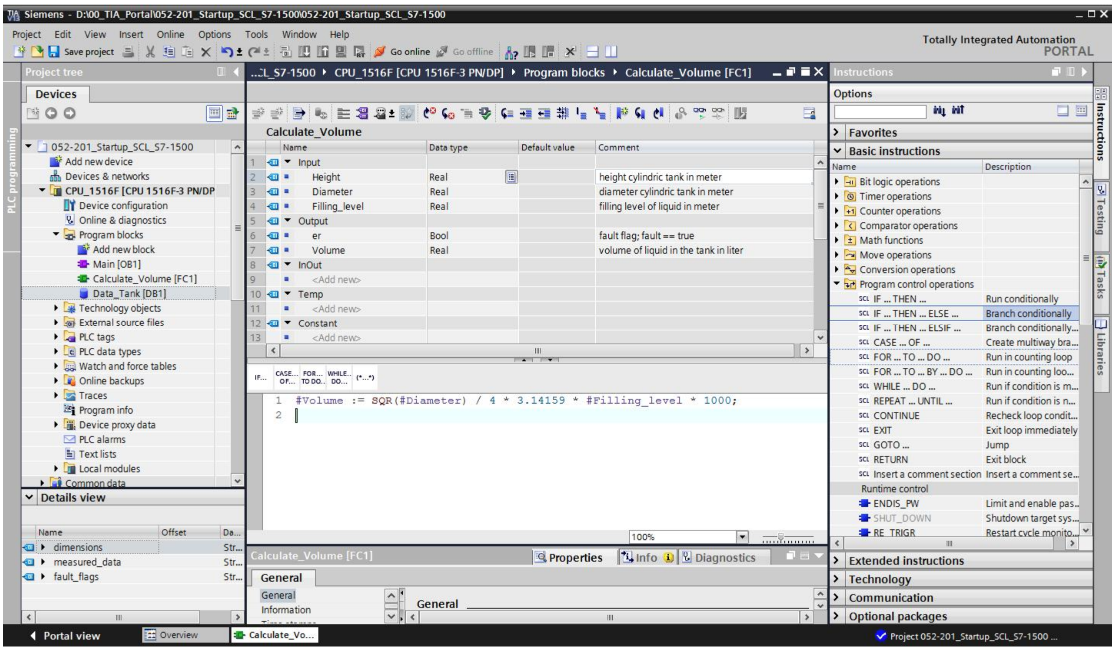
Options

> Favorites

∨ Basic instructions

Name	Description
▸ Bit logic operations	
▸ Timer operations	
▸ Counter operations	
▸ Comparator operations	
▸ Math functions	
▸ Move operations	
▸ Conversion operations	
▼ Program control operations	
SCL IF ... THEN ...	Run conditionally
SCL IF ... THEN ... ELSE ...	Branch conditionally
SCL IF ... THEN ... ELSIF ...	Branch conditionally...
SCL CASE ... OF ...	Create multiway bra...
SCL FOR ... TO ... DO ...	Run in counting loop
SCL FOR ... TO ... BY ... DO ...	Run in counting loo...
SCL WHILE ... DO ...	Run if condition is m...
SCL REPEAT ... UNTIL ...	Run if condition is n...
SCL CONTINUE	Recheck loop condit...
SCL EXIT	Exit loop immediately
SCL GOTO ...	Jump
SCL RETURN	Exit block
SCL Insert a comment section	Insert a comment se...

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


Siemens - D:\00_TIA_Portal\052-201_Startup_SCL_S7-1500\052-201_Startup_SCL_S7-1500
Project Edit View Insert Online Options Tools Window Help
Save project
Go online
Go offline
Totally Integrated Automation PORTAL
Project tree
Devices
052-201_Startup_SCL_S7-1500
Add new device
Devices & networks
CPU_1516F [CPU 1516F-3 PN/DP]
Device configuration
Online & diagnostics
Program blocks
Add new block
Main [OB1]
Calculate_Volume [FC1]
Data_Tank [DB1]
Technology objects
External source files
PLC tags
PLC data types
Watch and force tables
Online backups
Traces
Program info
Device proxy data
PLC alarms
Text lists
Local modules
Common data
Details view
dimensions
measured_data
fault_flags
...L_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Calculate_Volume [FC1]
IF... CASE... OF... FOR... TO DO... WHILE.. DO... (*...*)
Properties Info Diagnostics
General Information
Portal view Overview Calculate_Vo...
Project 052-201_Startup_SCL_S7-1500 ...

Calculate_Volume

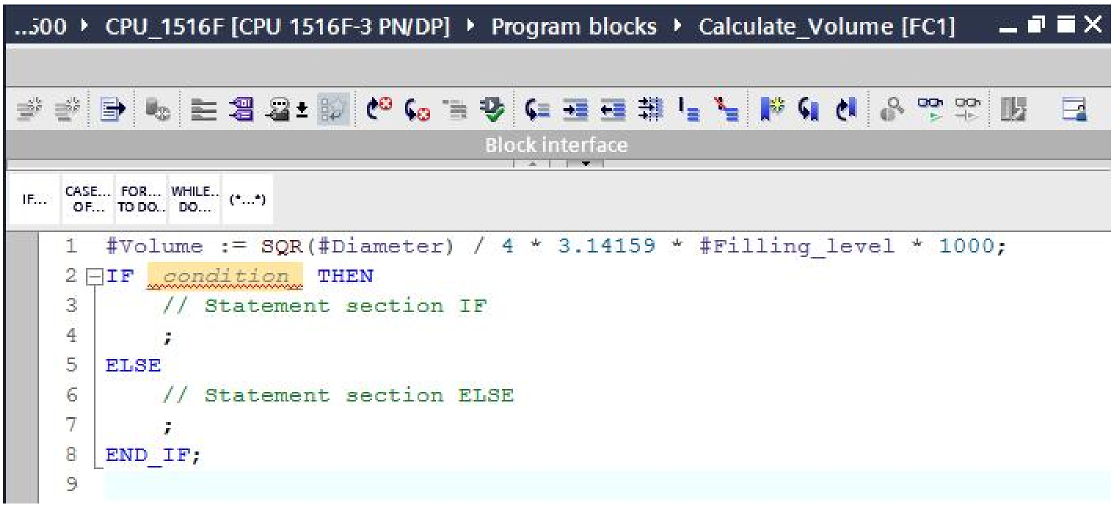
	Name	Data type	Default value	Comment
1	Input			
2	Height	Real		height cylindric tank in meter
3	Diameter	Real		diameter cylindric tank in meter
4	Filling_level	Real		filling level of liquid in meter
5	Output			
6	er	Bool		fault flag; fault == true
7	Volume	Real		volume of liquid in the tank in liter
8	InOut			
9	<Add new>			
10	Temp			
11	<Add new>			
12	Constant			
13	<Add new>			

```
#Volume := SQR(#Diameter) / 4 * 3.14159 * #Filling_level * 1000;

```

Instructions – Basic instructions

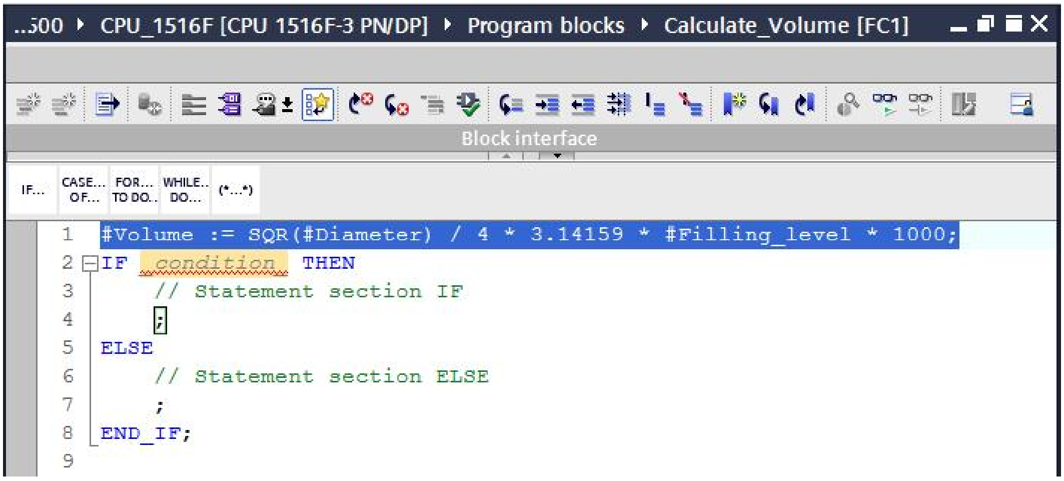
Name	Description
Bit logic operations	
Timer operations	
Counter operations	
Comparator operations	
Math functions	
Move operations	
Conversion operations	
Program control operations	
IF ... THEN ...	Run conditionally
IF ... THEN ... ELSE ...	Branch conditionally
IF ... THEN ... ELSIF ...	Branch conditionally...
CASE ... OF ...	Create multiway bra...
FOR ... TO ... DO ...	Run in counting loop
FOR ... TO ... BY ... DO ...	Run in counting loo...
WHILE ... DO ...	Run if condition is m...
REPEAT ... UNTIL ...	Run if condition is n...
CONTINUE	Recheck loop condit...
EXIT	Exit loop immediately
GOTO ...	Jump
RETURN	Exit block
Insert a comment section	Insert a comment se...
Runtime control	
ENDIS_PW	Limit and enable pas...
SHUT_DOWN	Shutdown target sys...
RE_TRIGR	Restart cycle monito...

Extended instructions
Technology
Communication
Optional packages

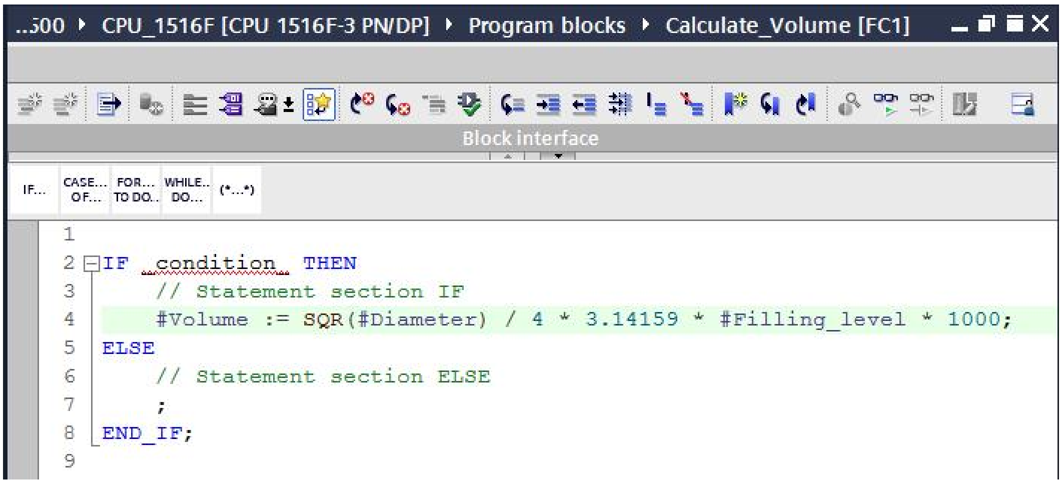

...500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Calculate_Volume [FC1]
Block interface
IF... CASE... OF... FOR... TO DO... WHILE.. DO... (*...*)

```
#Volume := SQR(#Diameter) / 4 * 3.14159 * #Filling_level * 1000;
IF  condition  THEN
    // Statement section IF
    ;
ELSE
    // Statement section ELSE
    ;
END_IF;

```

® Highlight the mathematical formula and drag it to the semicolon in front of the ELSE.
(® highlight ® drag & drop)


..500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Calculate_Volume [FC1]
Block interface
IF...
CASE... OF...
FOR... TO DO...
WHILE... DO...
(*...*)

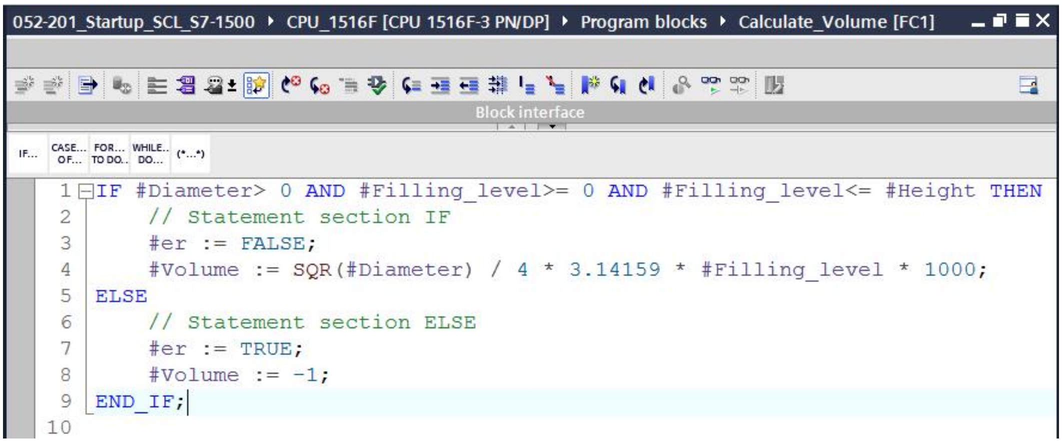
```
#Volume := SQR(#Diameter) / 4 * 3.14159 * #Filling_level * 1000;
IF condition THEN
    // Statement section IF
    ;
ELSE
    // Statement section ELSE
    ;
END_IF;
```


..500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Calculate_Volume [FC1]
Block interface
IF...
CASE... OF...
FOR... TO DO...
WHILE... DO...
(*...*)

```

IF condition THEN
    // Statement section IF
    #Volume := SQR(#Diameter) / 4 * 3.14159 * #Filling_level * 1000;
ELSE
    // Statement section ELSE
    ;
END_IF;
```

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

(® Complete program ®)

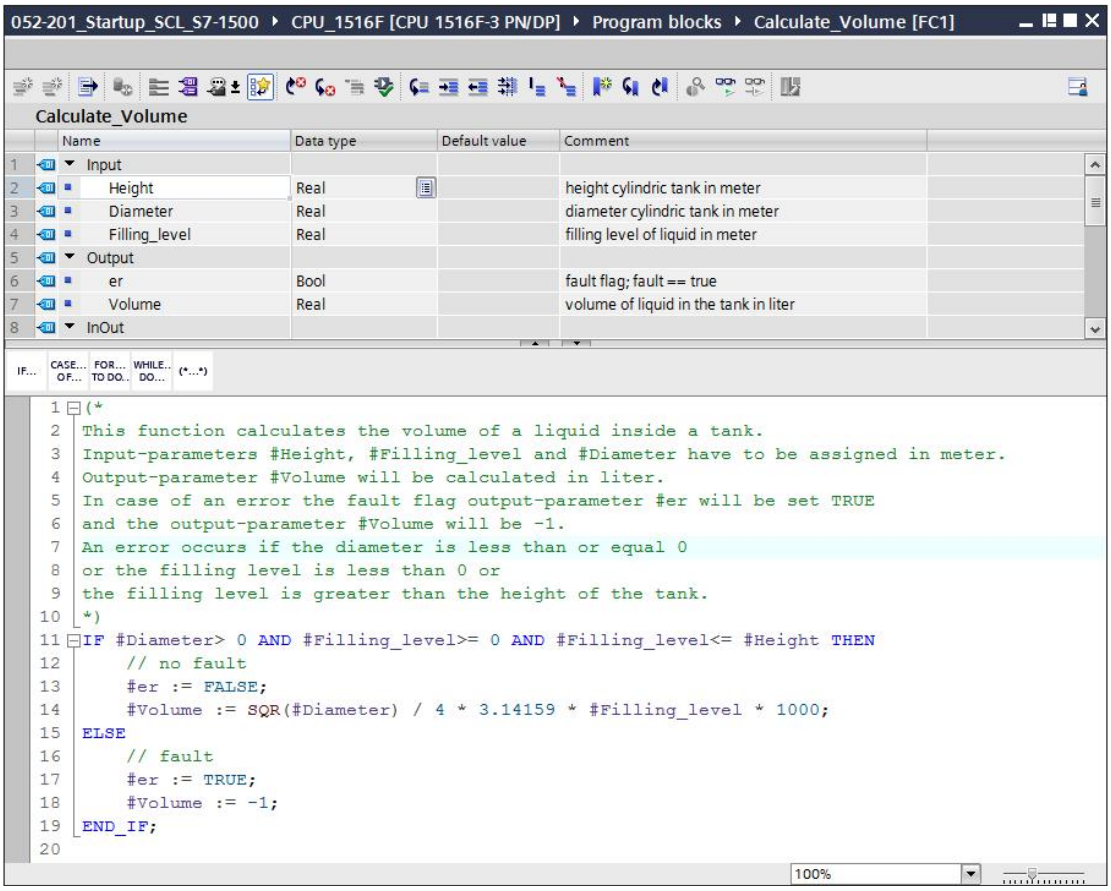

052-201_Startup_SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Calculate_Volume [FC1]
Block interface
IF...
CASE... OF...
FOR... TO DO...
WHILE... DO...
(*...*)

```
IF #Diameter> 0 AND #Filling_level>= 0 AND #Filling_level<= #Height THEN
    // Statement section IF
    #er := FALSE;
    #Volume := SQR(#Diameter) / 4 * 3.14159 * #Filling_level * 1000;
ELSE
    // Statement section ELSE
    #er := TRUE;
    #Volume := -1;
END_IF;
```

® Comments can be added with "(**)" as block comment and with "//" as row comment.

You can now complete your program with comments.

(® Add block comment starting with row 1 ® Add row comments in rows 12 and 16)


052-201_Startup_SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Calculate_Volume [FC1]
Calculate_Volume
IF...
CASE... OF...
FOR... TO DO..
WHILE.. DO...
(*...*)
100%

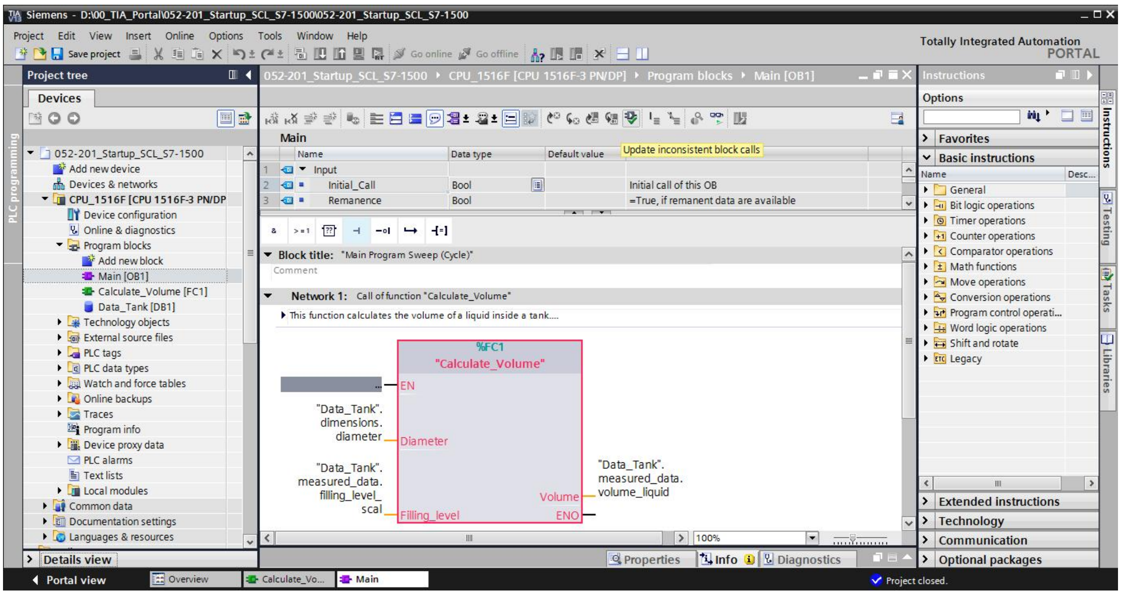
	Name	Data type	Default value	Comment
1	▼ Input			
2	Height	Real		height cylindric tank in meter
3	Diameter	Real		diameter cylindric tank in meter
4	Filling_level	Real		filling level of liquid in meter
5	▼ Output			
6	er	Bool		fault flag; fault == true
7	Volume	Real		volume of liquid in the tank in liter
8	▼ InOut			

```
 1 (*
 2 This function calculates the volume of a liquid inside a tank.
 3 Input-parameters #Height, #Filling_level and #Diameter have to be assigned in meter.
 4 Output-parameter #Volume will be calculated in liter.
 5 In case of an error the fault flag output-parameter #er will be set TRUE
 6 and the output-parameter #Volume will be -1.
 7 An error occurs if the diameter is less than or equal 0
 8 or the filling level is less than 0 or
 9 the filling level is greater than the height of the tank.
10 *)
11 IF #Diameter> 0 AND #Filling_level>= 0 AND #Filling_level<= #Height THEN
12     // no fault
13     #er := FALSE;
14     #Volume := SQR(#Diameter) / 4 * 3.14159 * #Filling_level * 1000;
15 ELSE
16     // fault
17     #er := TRUE;
18     #Volume := -1;
19 END_IF;
20
```

7.11 Customizing the organization block

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

(® Open OB1 ®)



Siemens - D:\00_TIA_Portal\052-201_Startup_SCL_S7-1500\052-201_Startup_SCL_S7-1500

052-201_Startup_SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Main [OB1]

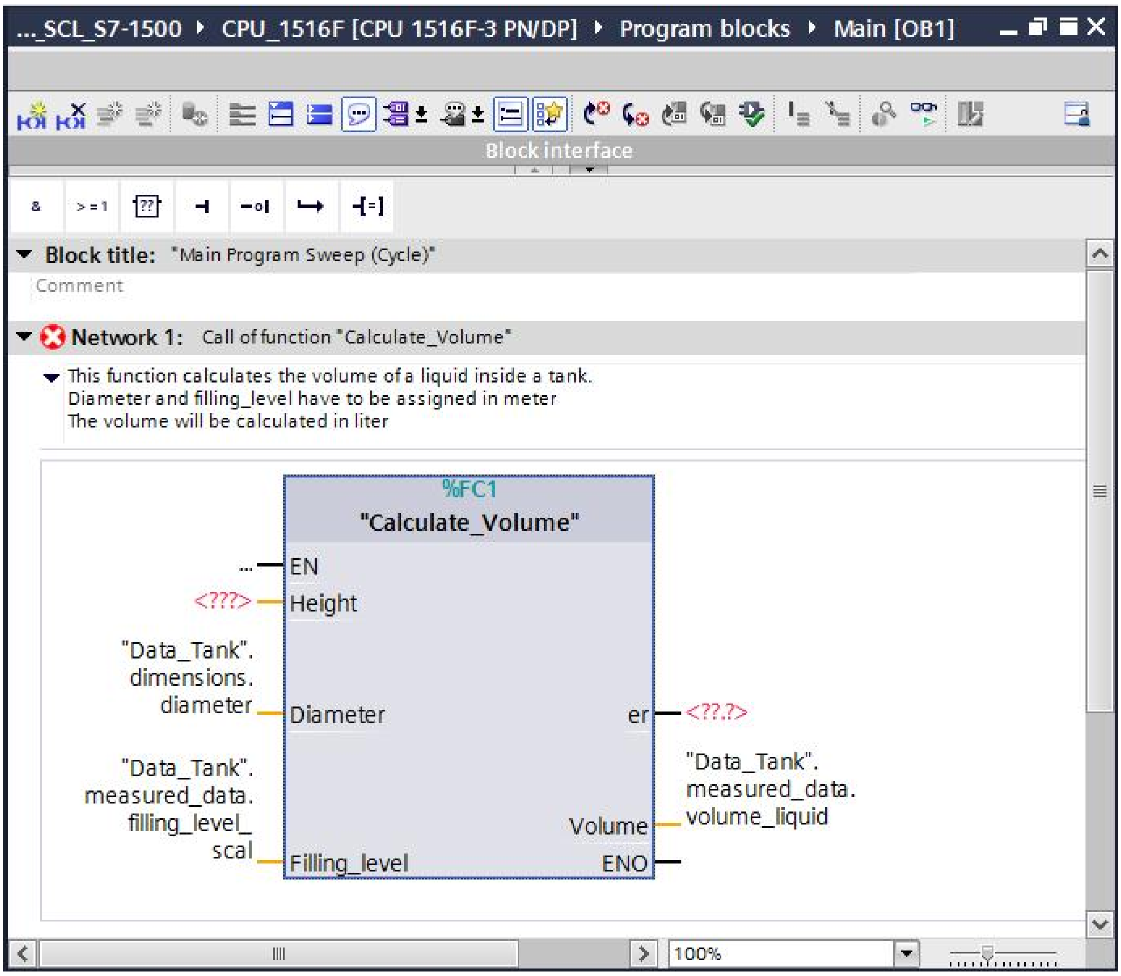
Update inconsistent block calls

	Name	Data type	Default value	Comment
1	▾ Input			
2	Initial_Call	Bool		Initial call of this OB
3	Remanence	Bool		=True, if remanent data are available

Block title: "Main Program Sweep (Cycle)"

Network 1: Call of function "Calculate_Volume"

This function calculates the volume of a liquid inside a tank....



..._SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Main [OB1]

Block title: "Main Program Sweep (Cycle)"

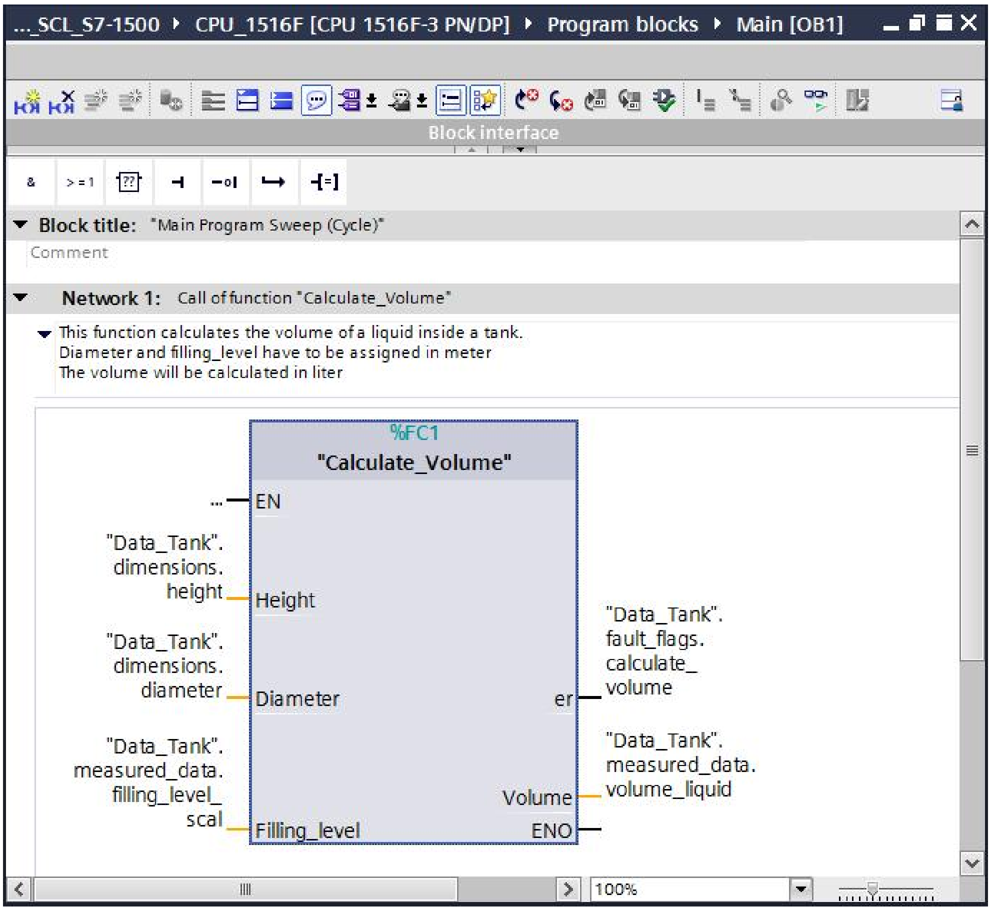
Comment

Network 1: Call of function "Calculate_Volume"

This function calculates the volume of a liquid inside a tank.
Diameter and filling_level have to be assigned in meter
The volume will be calculated in liter

Block interface
%FC1
"Calculate_Volume"
...
EN
<???>
Height
"Data_Tank".dimensions.diameter
Diameter
er
<??.?>
"Data_Tank".measured_data.filling_level_scal
Filling_level
Volume
"Data_Tank".measured_data.volume_liquid
ENO
100%

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



..._SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Main [OB1]
Block interface
& >=1 ??
%FC1
"Calculate_Volume"
EN
Height
Diameter
Filling_level
er
Volume
ENO
...
"Data_Tank". dimensions. height
"Data_Tank". dimensions. diameter
"Data_Tank". measured_data. filling_level_ scal
"Data_Tank". fault_flags. calculate_ volume
"Data_Tank". measured_data. volume_liquid
100%

Block title: "Main Program Sweep (Cycle)"

Comment

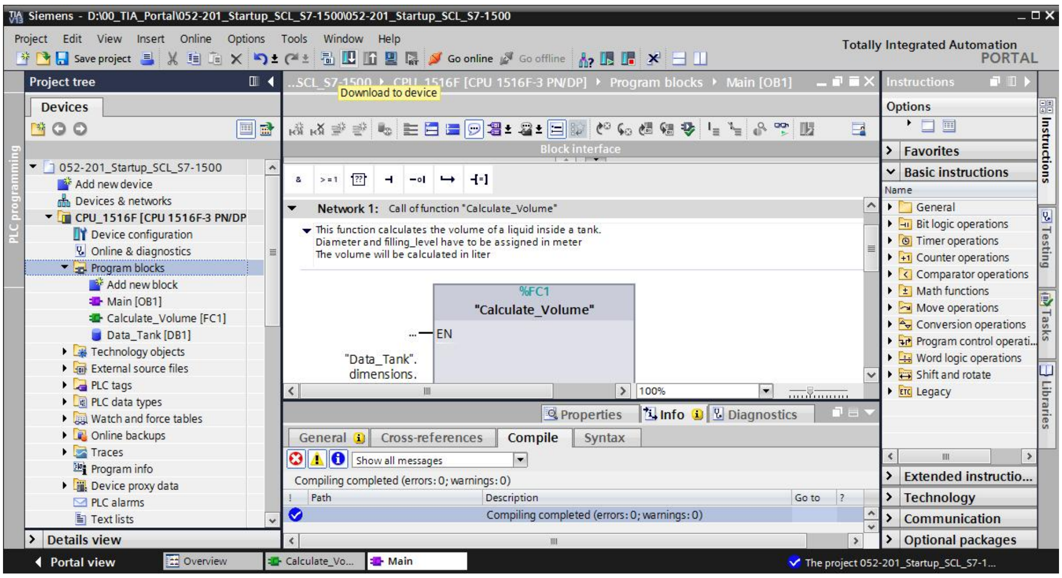
Network 1: Call of function "Calculate_Volume"

This function calculates the volume of a liquid inside a tank.
Diameter and filling_level have to be assigned in meter
The volume will be calculated in liter

7.12 Compiling and downloading the program

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

(® Program blocks ® ® ® Save project)



Network 1: Call of function "Calculate_Volume"

This function calculates the volume of a liquid inside a tank.
Diameter and filling_level have to be assigned in meter
The volume will be calculated in liter

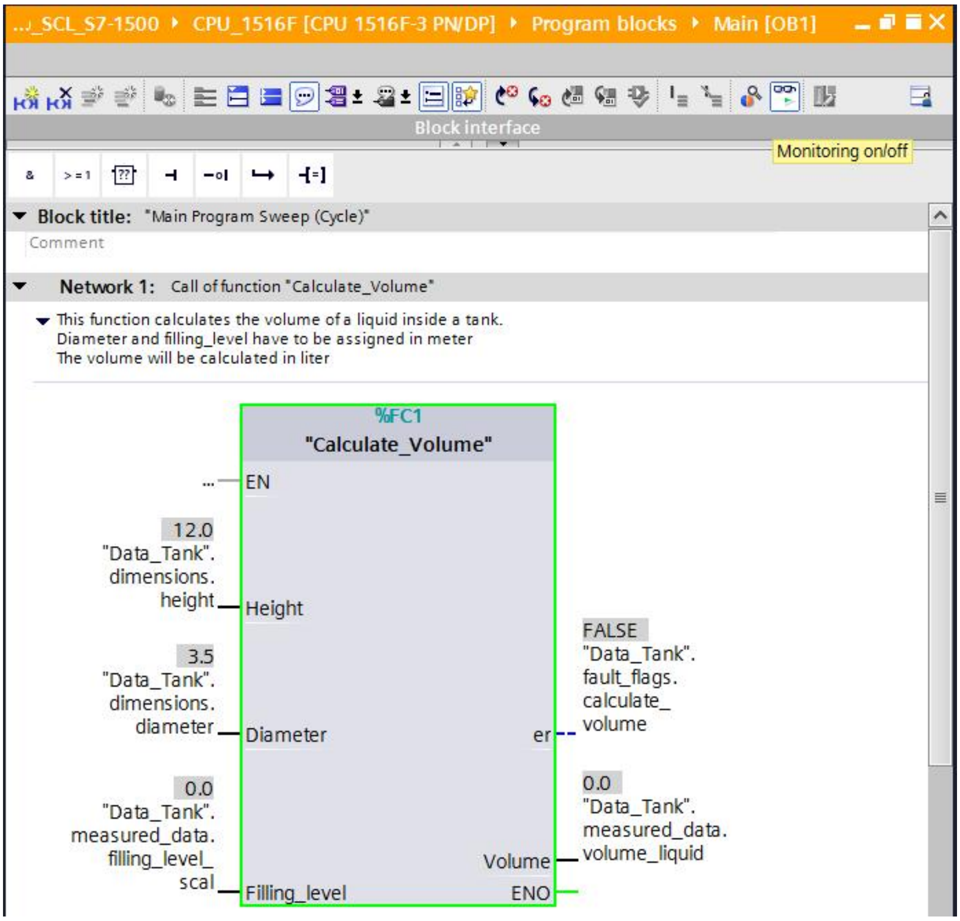
Compiling completed (errors: 0; warnings: 0)

Compiling completed (errors: 0; warnings: 0)

The project 052-201_Startup_SCL_S7-1...

7.13 Monitoring and testing the organization block

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



..._SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Main [OB1]

Block title: "Main Program Sweep (Cycle)"

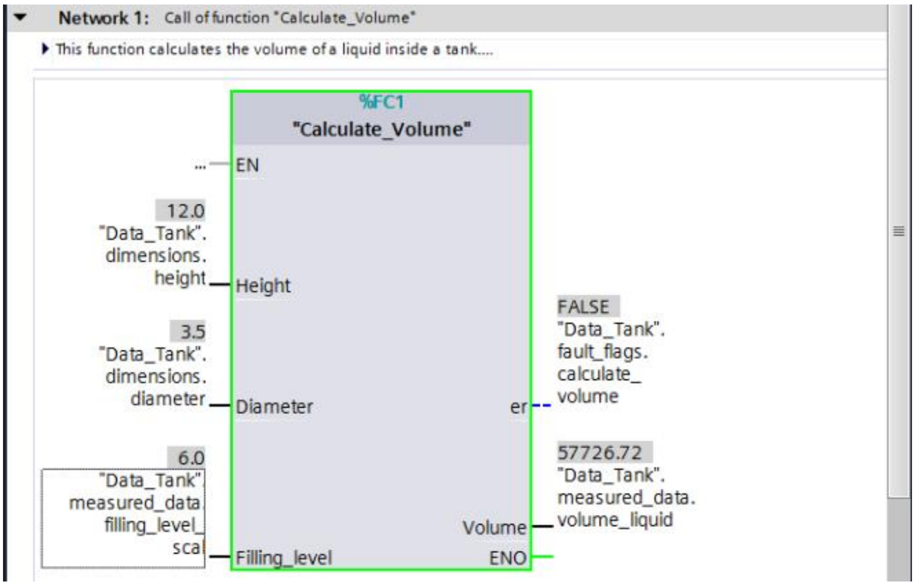
Comment

Network 1: Call of function "Calculate_Volume"

This function calculates the volume of a liquid inside a tank.
Diameter and filling_level have to be assigned in meter
The volume will be calculated in liter

Block interface
Monitoring on/off
%FC1
"Calculate_Volume"
EN
12.0
"Data_Tank".dimensions.height
Height
3.5
"Data_Tank".dimensions.diameter
Diameter
0.0
"Data_Tank".measured_data.filling_level_scal
Filling_level
FALSE
"Data_Tank".fault_flags.calculate_volume
er
0.0
"Data_Tank".measured_data.volume_liquid
Volume
ENO

® 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)

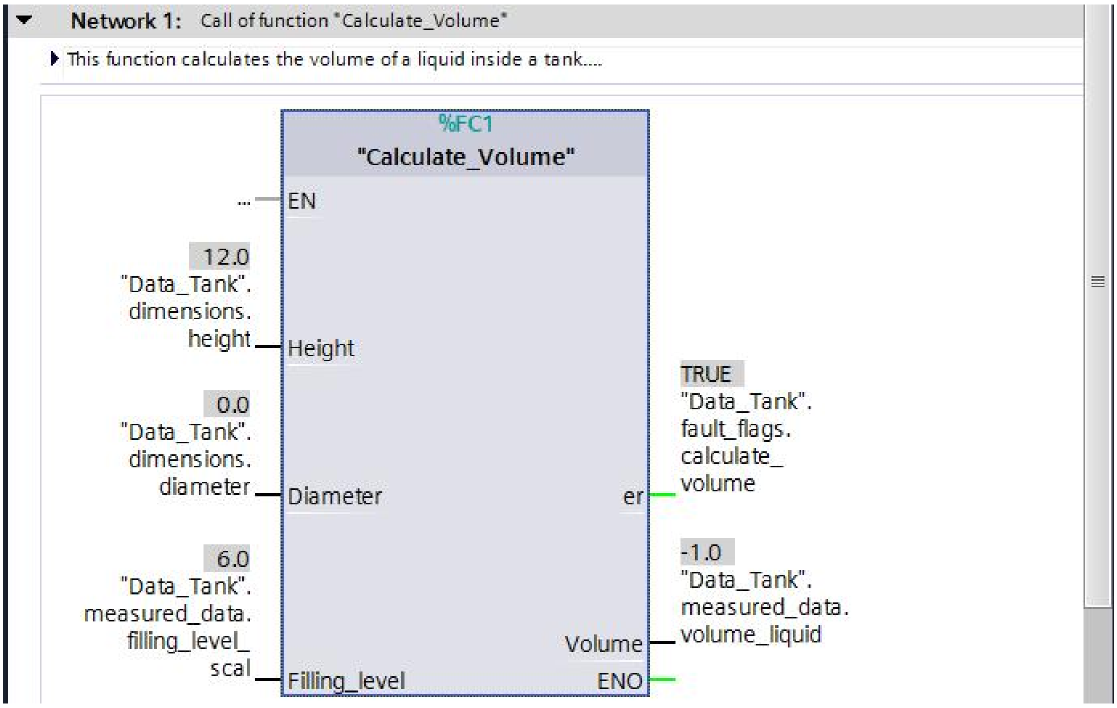


Network 1: Call of function "Calculate_Volume"

This function calculates the volume of a liquid inside a tank....

%FC1
"Calculate_Volume"
EN
12.0
"Data_Tank".dimensions.height
Height
3.5
"Data_Tank".dimensions.diameter
Diameter
6.0
"Data_Tank".measured_data.filling_level_scal
Filling_level
FALSE
"Data_Tank".fault_flags.calculate_volume
er
57726.72
"Data_Tank".measured_data.volume_liquid
Volume
ENO

® 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)



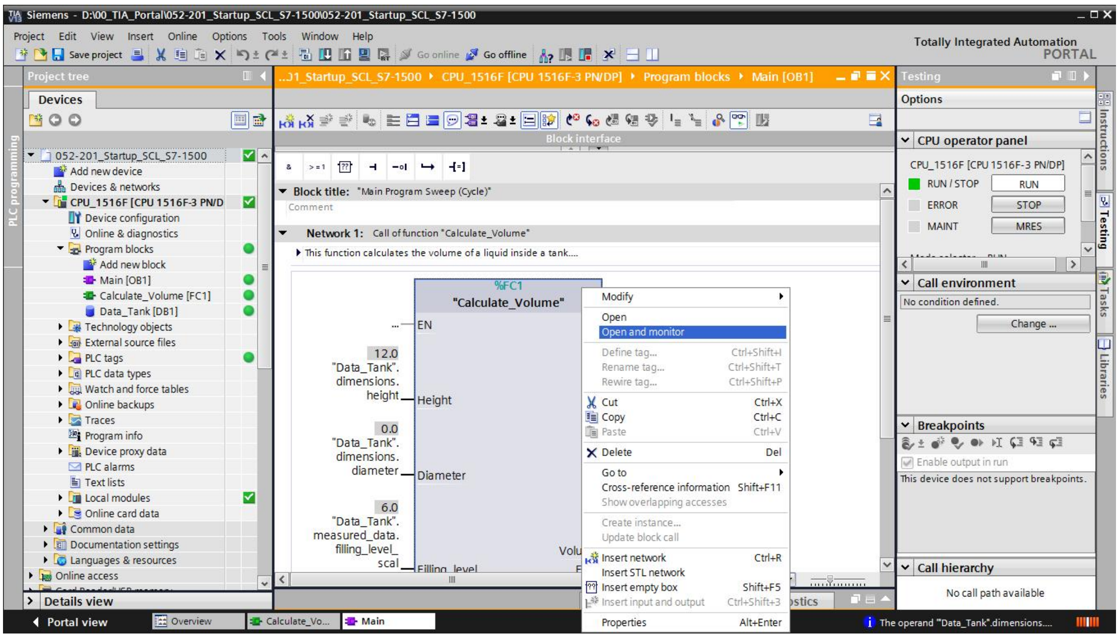
Network 1: Call of function "Calculate_Volume"

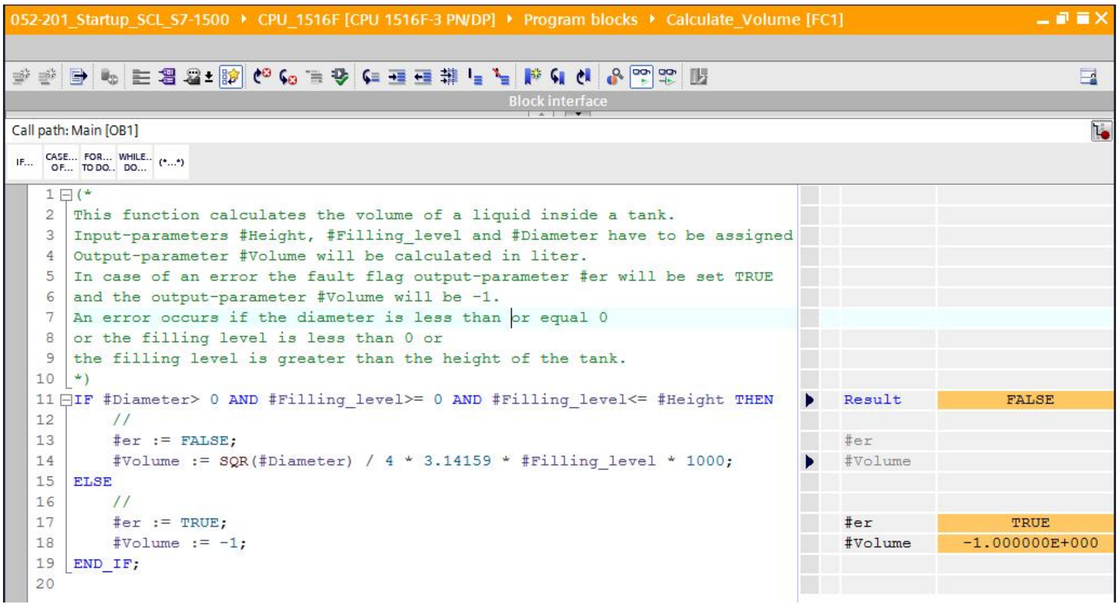
This function calculates the volume of a liquid inside a tank....

7.14 Monitoring and testing the "Calculate_Volume" function

® Finally, open and monitor the "Calculate_Volume" function with a right-click on the function and selection of the "Open and monitor" menu command.

(® Right-click on function ® Open und monitor)

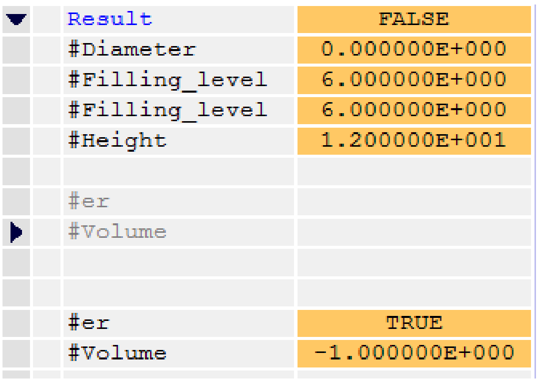



052-201_Startup_SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Calculate_Volume [FC1]
Block interface
Call path: Main [OB1]
IF... CASE... OF... FOR... TO DO... WHILE... DO... (*...*)

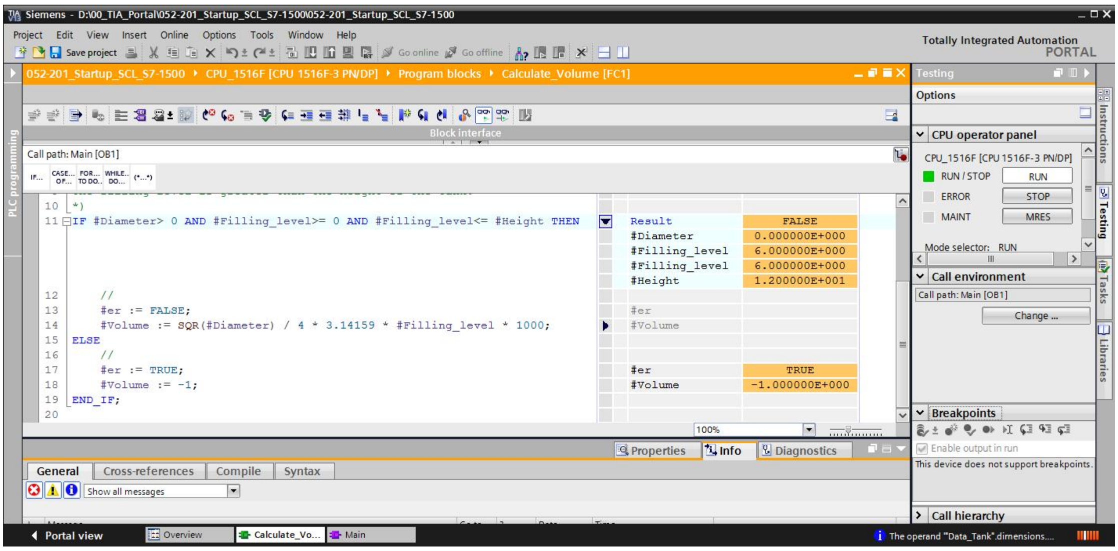
```
(*
This function calculates the volume of a liquid inside a tank.
Input-parameters #Height, #Filling_level and #Diameter have to be assigned
Output-parameter #Volume will be calculated in liter.
In case of an error the fault flag output-parameter #er will be set TRUE
and the output-parameter #Volume will be -1.
An error occurs if the diameter is less than or equal 0
or the filling level is less than 0 or
the filling level is greater than the height of the tank.
*)
IF #Diameter> 0 AND #Filling_level>= 0 AND #Filling_level<= #Height THEN
    //
    #er := FALSE;
    #Volume := SQR(#Diameter) / 4 * 3.14159 * #Filling_level * 1000;
ELSE
    //
    #er := TRUE;
    #Volume := -1;
END_IF;
```

Result	FALSE
#er	
#Volume	
#er	TRUE
#Volume	-1.000000E+000

® You can show the values of the individual tags of the IF query with a click on the black arrow ▼.
(® ▼)



▼	Result	FALSE
	#Diameter	0.000000E+000
	#Filling_level	6.000000E+000
	#Filling_level	6.000000E+000
	#Height	1.200000E+001
	#er	
▶	#Volume	
	#er	TRUE
	#Volume	-1.000000E+000



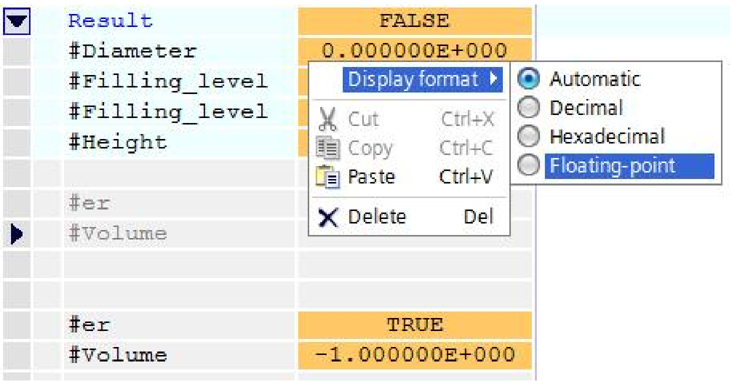
Siemens - D:\00_TIA_Portal\052-201_Startup_SCL_S7-1500\052-201_Startup_SCL_S7-1500
Project Edit View Insert Online Options Tools Window Help
Save project
Go online
Go offline
Totally Integrated Automation PORTAL
052-201_Startup_SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Calculate_Volume [FC1]
PLC programming
Block interface
Call path: Main [OB1]
IF... CASE... OF... FOR... TO DO... WHILE... DO... (*...*)
100%
Properties
Info
Diagnostics
General Cross-references Compile Syntax
Show all messages
Portal view
Overview
Calculate_Vo...
Main
The operand "Data_Tank".dimensions....
Testing
Options
CPU operator panel
CPU_1516F [CPU 1516F-3 PN/DP]
RUN / STOP RUN
ERROR STOP
MAINT MRES
Mode selector: RUN
Call environment
Call path: Main [OB1]
Change ...
Breakpoints
Enable output in run
This device does not support breakpoints.
Call hierarchy
Instructions
Testing
Tasks
Libraries

```
*)
IF #Diameter> 0 AND #Filling_level>= 0 AND #Filling_level<= #Height THEN



    //
    #er := FALSE;
    #Volume := SQR(#Diameter) / 4 * 3.14159 * #Filling_level * 1000;
ELSE
    //
    #er := TRUE;
    #Volume := -1;
END_IF;
```

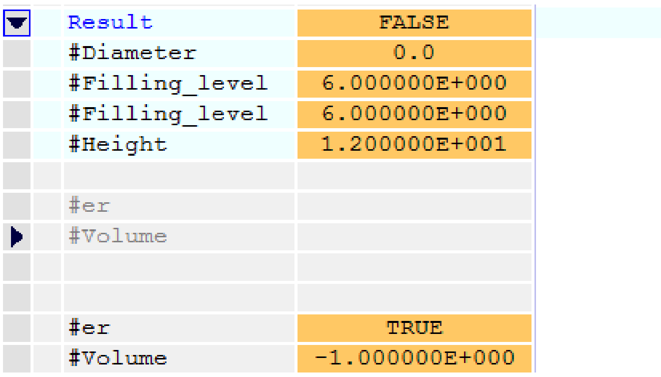
▼	Result	FALSE
	#Diameter	0.000000E+000
	#Filling_level	6.000000E+000
	#Filling_level	6.000000E+000
	#Height	1.200000E+001
	#er	
▶	#Volume	
	#er	TRUE
	#Volume	-1.000000E+000

® Right-click the tag to adjust the display format.

(® Right-click tag ® Display format ® Floating point)



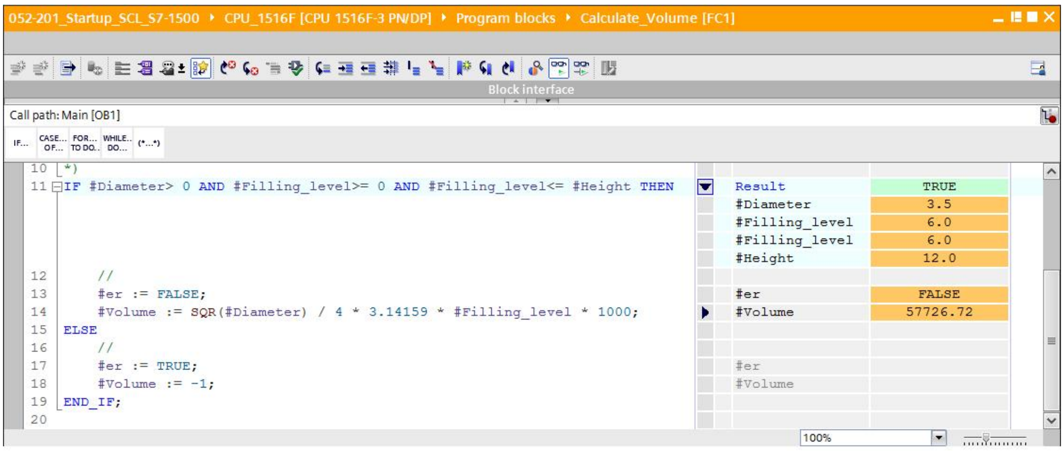
```
Result             FALSE
#Diameter          0.000000E+000
#Filling_level
#Filling_level
#Height

#er
#Volume


#er                TRUE
#Volume            -1.000000E+000
```



```
Result             FALSE
#Diameter          0.0
#Filling_level     6.000000E+000
#Filling_level     6.000000E+000
#Height            1.200000E+001

#er
#Volume


#er                TRUE
#Volume            -1.000000E+000
```

® 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)



052-201_Startup_SCL_S7-1500 ▸ CPU_1516F [CPU 1516F-3 PN/DP] ▸ Program blocks ▸ Calculate_Volume [FC1]

Call path: Main [OB1]

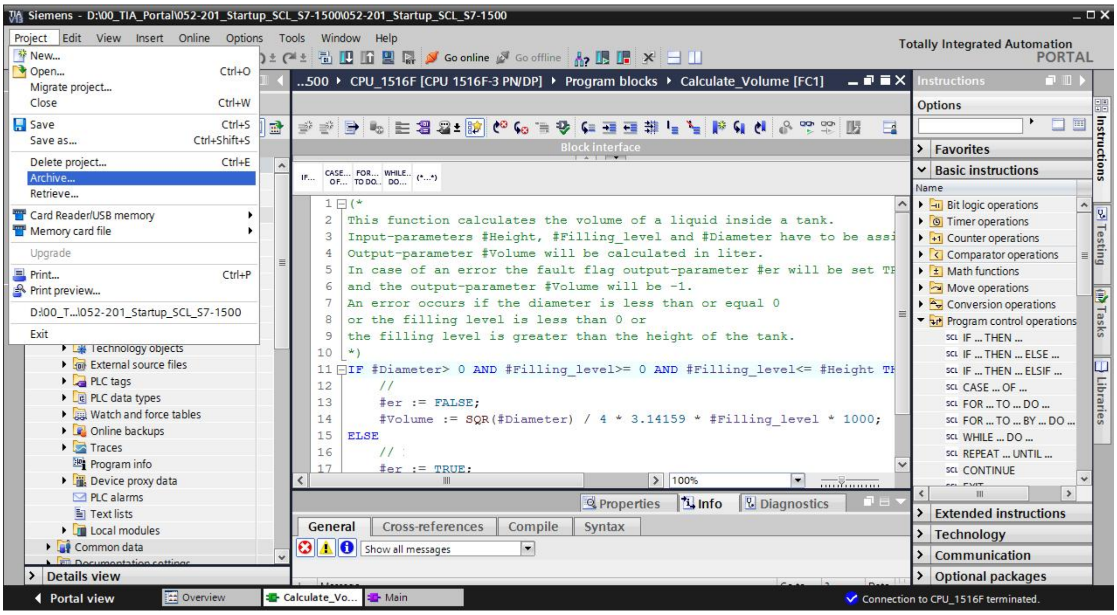
```
*)
IF #Diameter> 0 AND #Filling_level>= 0 AND #Filling_level<= #Height THEN
    //
    #er := FALSE;
    #Volume := SQR(#Diameter) / 4 * 3.14159 * #Filling_level * 1000;
ELSE
    //
    #er := TRUE;
    #Volume := -1;
END_IF;
```

```
Result             TRUE
#Diameter          3.5
#Filling_level     6.0
#Filling_level     6.0
#Height            12.0

#er                FALSE
#Volume            57726.72


#er
#Volume
```

7.15 Archiving the project

® Finally, you want to archive the complete project. 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 archives'.

(® Project ® Archive ® TIA Portal Project archives ® SCE_EN_052-201 Startup SCL_S7-1500… ® Save)



```
(*
This function calculates the volume of a liquid inside a tank.
Input-parameters #Height, #Filling_level and #Diameter have to be assi
Output-parameter #Volume will be calculated in liter.
In case of an error the fault flag output-parameter #er will be set TR
and the output-parameter #Volume will be -1.
An error occurs if the diameter is less than or equal 0
or the filling level is less than 0 or
the filling level is greater than the height of the tank.
*)
IF #Diameter> 0 AND #Filling_level>= 0 AND #Filling_level<= #Height TH
    //
    #er := FALSE;
    #Volume := SQR(#Diameter) / 4 * 3.14159 * #Filling_level * 1000;
ELSE
    //
    #er := TRUE;
```

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 Filling_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 shall be generally applicable to any positive analog value. In our example task "Tank", the filling level is read by an analog sensor and stored as scaled value in the data block with this function.

In case of an error, the block shall 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:

$$\#\mathrm{Analog\_scal} = \frac{\#\mathrm{Analog\_per}}{27648} \cdot (\#\mathrm{mx} - \#\mathrm{mn}) + \#\mathrm{mn}$$

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

Additional information for more details is available as orientation help, such as Getting Started, Videos, Tutorials, Apps, Manuals, Programming Guidelines and Trial Software/Firmware, at the following link:

www.siemens.com/sce/s7-1500