



SIEMENS

Curriculum SCE

Siemens Automation Cooperates with Education | 09/2017

Module 052-201 TIA Portal
Programmation en langage évolué
avec SCL et SIMATIC S7-1500

Cooperates
with Education

Automation

SIEMENS

Packages SCE pour formateurs adaptés à ces supports d'apprentissage/de formation

- **SIMATIC S7 CPU 1516F-3 PN/DP**
N° d'article : 6ES7516-3FN00-4AB2
- **SIMATIC STEP 7 Professional V14 SP1 - Licence monoposte**
N° d'article : 6ES7822-1AA04-4YA5
- **SIMATIC STEP 7 Professional V14 SP1 - Licence 6 postes pour salle de classe**
N° d'article : 6ES7822-1BA04-4YA5
- **SIMATIC STEP 7 Professional V14 SP1 - Licence de mise à niveau 6 postes**
N° d'article : 6ES7822-1AA04-4YE5
- **SIMATIC STEP 7 Professional V14 SP1 - Licence 20 postes pour étudiants**
N° d'article : 6ES7822-1AC04-4YA5

Veuillez noter que les packages pour formateurs ont parfois été remplacés par de nouveaux packages.

Vous pouvez consulter les packages SCE actuellement disponibles sous : [siemens.com/sce/tp](https://www.siemens.com/sce/tp)

Formations

Pour les formations Siemens SCE régionales, contactez votre interlocuteur SCE régional : [siemens.com/sce/contact](https://www.siemens.com/sce/contact)

Plus d'informations sur le programme SCE

[siemens.com/sce](https://www.siemens.com/sce)

Indications d'utilisation

Le curriculum SCE concernant la solution intégrée d'automatisation Totally Integrated Automation (TIA) a été spécialement élaboré pour le programme éducatif "Siemens Automation Cooperates with Education (SCE)" destiné aux établissements d'enseignement et aux organismes de R&D. Siemens AG n'assume aucune responsabilité quant au contenu de la formation.

Le présent dossier de formation ne peut être utilisé que dans le cadre d'une formation initiale sur les produits/systèmes Siemens. Cela signifie qu'il peut être copié, en partie ou en intégralité et fourni aux étudiants dans le cadre de leur formation. La diffusion et la reproduction de la présente documentation, ainsi que la communication de son contenu, sont autorisées au sein des établissements d'enseignement et de formation à des fins éducatives. Toute exception à ces modalités nécessite l'autorisation écrite de Siemens AG. Interlocuteur : Monsieur Roland Scheuerer roland.scheuerer@siemens.com.

Toute infraction entraînera un dédommagement. Tous les droits sont réservés, y compris ceux liés à la traduction, en particulier pour la délivrance d'un brevet ou l'enregistrement d'un modèle communautaire.

L'utilisation dans le cadre d'une formation de la clientèle industrielle est expressément interdite. L'usage commercial du présent curriculum n'est pas autorisé.

Nous souhaitons remercier l'Université Technique de Dresde, la société Michael Dziallas Engineering, ainsi que toutes les personnes ayant contribué à l'élaboration de ce curriculum SCE.

Sommaire

1.	Objectif	4
2.	Conditions.....	4
3.	Matériel et logiciel requis	5
4.	Théorie	6
4.1	A propos du langage de programmation S7-SCL.....	6
4.2	A propos de l'environnement de développement S7-SCL	6
5.	Énoncé	9
5.1	Exemple d'application : contenu d'une citerne.....	9
5.2	Extension de l'exemple d'application	9
6.	Réalisation.....	9
6.1	Bloc de données global "Données_citerne"	9
6.2	Fonction "calculer_volume_citerne" ("calculate_volume_tank").....	10
6.3	Extension de la fonction "Calculer_volume_citerne".....	10
7.	Instructions structurées étape par étape	11
7.1	Désarchivage d'un projet existant.....	11
7.2	Enregistrement du projet sous un autre nom	12
7.3	Création du bloc de données "Données_citerne"	12
7.4	Création de la fonction "Calculer_volume" ("Calculate_volume").....	14
7.5	Définition de la fonction "Calculer_volume" ("Calculate_volume")	15
7.6	Programmation de la fonction "Calculer_volume" ("Calculate_volume")	16
7.7	Programmation du bloc d'organisation "Main [OB1]"	17
7.8	Compiler et charger le programme	19
7.9	Visualiser et tester le bloc d'organisation	20
7.10	Extension de la fonction "Calculer_Volume" ("Calculate_Volume").....	22
7.11	Modifier le bloc d'organisation	27
7.12	Compiler et charger le programme	28
7.13	Visualiser et tester le bloc d'organisation	29
7.14	Visualiser et tester la fonction "Calculer_Volume" ("Calculate_Volume").....	31
7.15	Archivage du projet	34
8.	Liste de contrôle.....	34
9.	Exercice.....	35
9.1	Énoncé – Exercice	35
9.2	Réalisation.....	35
9.3	Liste de contrôle – exercice.....	36
10.	Informations complémentaires.....	37

Programmation en langage évolué avec S7-SCL

1. Objectif

Dans ce chapitre, vous vous familiariserez avec les fonctions de base du langage évolué S7-SCL. Les fonctions test permettant de résoudre les erreurs logiques de programmation seront également abordées.

Les automates SIMATIC S7 énumérés sous le chapitre 3 peuvent être utilisés.

2. Conditions

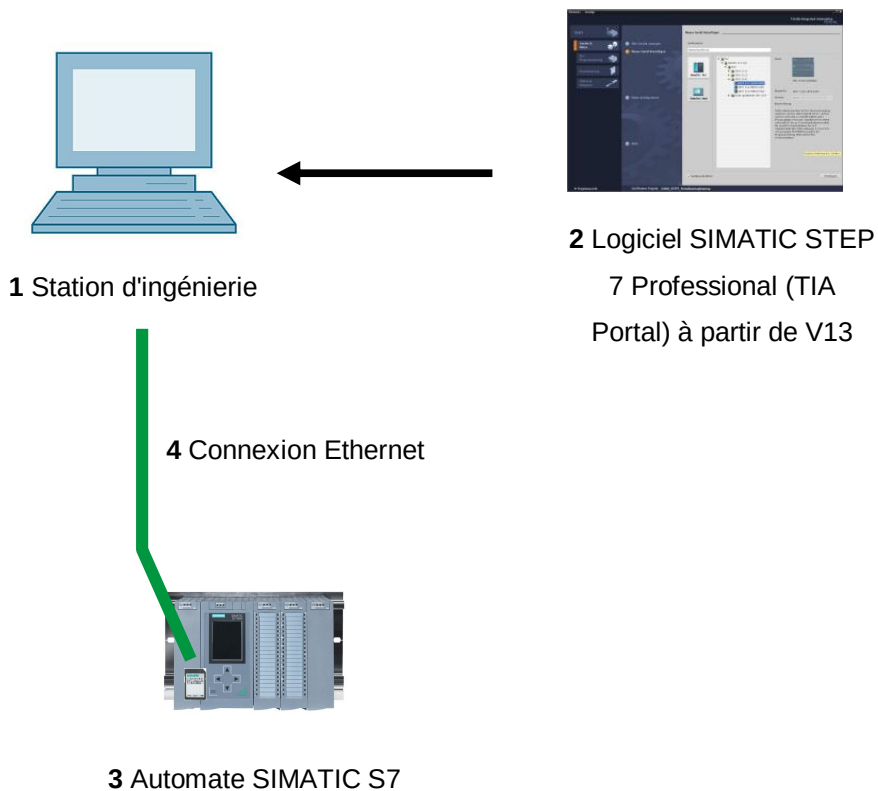
Ce chapitre est basé sur la configuration matérielle d'un SIMATIC S7. Il peut être réalisé avec toute configuration matérielle possédant des cartes d'entrées/sorties TOR. Pour la mise en œuvre de ce chapitre, vous pouvez par exemple utiliser le projet suivant :

"SCE_FR_012_101_configuration_matérielle_CPU1516F.....zap13"

Par ailleurs, des connaissances de base sur la programmation en langage évolué, comme le langage Pascal, sont requises.

3. Matériel et logiciel requis

- 1 Station d'ingénierie : Les conditions concernent le matériel et le système d'exploitation (pour plus d'informations, voir le fichier Lisezmoi sur les DVD d'installation de TIA Portal)
- 2 Logiciel SIMATIC STEP 7 Professional dans TIA Portal – à partir de V13
- 3 Automate SIMATIC S7-1500/S7-1200/S7-300, par exemple CPU 1516F-3 PN/DP – à partir de la version de firmware V1.6 avec carte mémoire et 16DI/16DO ainsi que 2AI/1AO
- 4 Connexion Ethernet entre la station d'ingénierie et l'automate



4. Théorie

4.1 A propos du langage de programmation S7-SCL

S7-SCL (Structured Control Language) est un langage de programmation évolué apparenté au langage PASCAL qui permet une programmation structurée. Le langage est conforme aux normes EN-61131-3 (CEI 61131-3) définissant le diagramme fonctionnel en séquence SFC "Sequential Function Chart". En plus des éléments de langage évolué, S7-SCL contient également des éléments typiques de l'API, des éléments de langage tels que les entrées, sorties, temporisations, mémentos, appels de blocs, etc. Il prend en charge le concept de blocs de STEP 7 et permet ainsi une programmation des blocs conforme aux normes en plus de la liste d'instructions (LIST), du schéma à contacts (CONT) et du logigramme (LOG). C.à.d. que S7-SCL complète et étend le logiciel de programmation STEP 7 avec ses langues de programmation CONT, LOG et LIST.

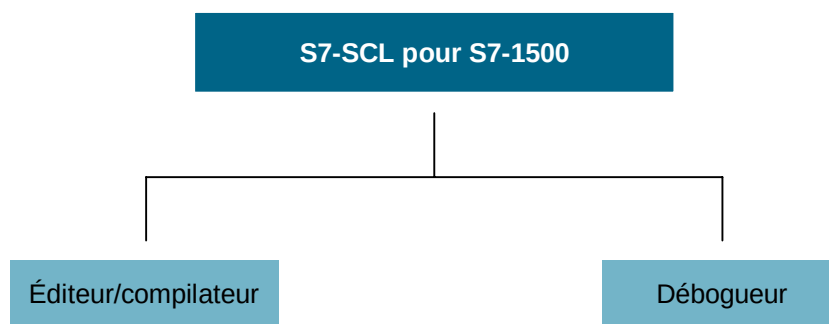
Il n'est pas nécessaire de créer chaque fonction, vous pouvez utiliser des blocs préprogrammés, comme les fonctions systèmes et les blocs fonctionnels système qui se trouvent dans le système d'exploitation de l'unité centrale.

Vous pouvez mélanger des blocs programmés avec S7-SCL avec des blocs LIST, CONT et LOG. Cela signifie qu'un bloc programmé avec S7-SCL peut appeler un autre bloc programmé dans LIST, CONT ou LOG. De la même façon, les blocs S7-SCL peuvent également être appelés dans des programmes LIST, CONT ou LOG.

Les fonctions de test de S7-SCL permettent de rechercher les erreurs logiques de programmation dans une compilation correcte.

4.2 A propos de l'environnement de développement S7-SCL

Pour utiliser et mettre en œuvre au mieux S7-SCL, vous disposez d'un environnement de développement conçu pour répondre à la fois aux particularités de S7-SCL et à celles de STEP 7. Cet environnement est composé d'un éditeur/compilateur et d'un débogueur.



Éditeur/compilateur

L'éditeur S7-SCL est un éditeur de texte permettant d'éditer tous types de textes. Il vous permet principalement de créer et d'éditer des blocs pour des programmes STEP 7. Durant la saisie, un contrôle de syntaxe approfondi permet d'assurer avec facilité une programmation sans erreur. Les erreurs de syntaxe sont affichées en différentes couleurs.

L'éditeur offre les possibilités suivantes :

- Programmation d'un bloc S7 dans le langage S7-SCL.
- Ajout d'éléments de langage et d'appels de bloc par simple glisser-déposer.
- Contrôle de syntaxe directement pendant la programmation.
- Configuration de l'éditeur en fonction des besoins spécifiques, p. ex. colorisation des différents éléments de langage en fonction de leur syntaxe.
- Vérification du bloc terminé par compilation.
- Affichage des erreurs et avertissements qui surviennent pendant la compilation.
- Localisation de l'erreur dans le bloc, éventuellement avec description de l'erreur et proposition de correction.

Débogueur

Le débogueur S7-SCL permet de contrôler l'exécution d'un programme dans l'AS et de détecter ainsi les erreurs logiques éventuelles.

S7-SCL propose à cet effet deux types de test :

- Visualisation en continu
- Visualisation par étape

La "visualisation en continu" permet de tester un groupe d'instructions dans un bloc. Pendant le test, les valeurs des variables et des paramètres s'affichent chronologiquement et sont actualisées, autant que possible, de manière cyclique.

En mode "visualisation par étape", le programme est suivi dans son déroulement logique. Vous pouvez exécuter l'algorithme instruction par instruction et observer la modification des variables dans une fenêtre de résultats.

La "visualisation par étape" est possible ou non en fonction de la CPU utilisée. Celle-ci doit prendre en charge l'utilisation de points d'arrêt. La CPU utilisée dans ce document ne prend pas en charge les points d'arrêt.

5. Énoncé

5.1 Exemple d'application : contenu d'une citerne

La première partie consiste à programmer le calcul du contenu d'une citerne.

5.2 Extension de l'exemple d'application

Dans la deuxième partie, la tâche doit être complétée par la programmation d'un traitement d'erreurs.

6. Réalisation

La citerne a la forme d'un cylindre vertical. Le niveau du contenu est mesuré avec un capteur analogique. Pour la première partie, la valeur du niveau doit être affichée de manière normée, l'unité étant le mètre.

Les paramètres globaux, comme le diamètre et la hauteur de la citerne, doivent être enregistrés de manière structurée dans un bloc de données global "Données_citerne".

Le programme pour le calcul du contenu doit être écrit dans une fonction appelée "calculer_volume_citerne" ("calculate_volume_tank") et les paramètres doivent utiliser l'unité mètre ou litre.

6.1 Bloc de données global "Données_citerne"

Les paramètres globaux sont stockés dans un bloc de données global dans plusieurs structures.

Nom	Type de données	Valeur initiale	Commentaire
Dimensions	STRUCT		
Hauteur	REAL	12,0	en mètres
Diamètre	REAL	3,5	en mètres
Valeurs de mesure	STRUCT		
Niveau_pér	INT	0	Valeur entre 0...27648
Cadrage_niveau	REAL	0,0	Valeur entre 0...12,0
Contenu	REAL	0,0	Contenu de la citerne en litres
Indicateurs d'erreur	STRUCT		
calculer_volume (calculate_volume)	BOOL		En cas d'erreur = TRUE

Tableau 1 : Paramètres dans le bloc de données "Données_citerne"

6.2 Fonction "calculer_volume_citerne" ("calculate_volume_tank")

Ce bloc calcule le contenu de la citerne en litres.

Dans la première étape, il n'est pas nécessaire de vérifier la pertinence des paramètres transférés.

Les paramètres suivants sont requis pour cette étape :

Entrée	Type de données	Commentaire
Diamètre	REAL	Diamètre en mètres de la citerne cylindrique
Niveau	REAL	Niveau en mètres de la citerne cylindrique
Sortie		
Contenu	REAL	Contenu de la citerne cylindrique en litres

Tableau 2 : Paramètres pour le FC "calculer_volume_citerne" dans la première étape

La formule de calcul du volume d'un cylindre vertical est utilisée pour résoudre la tâche. Le facteur de conversion 1000 est utilisé pour calculer le résultat en litres.

$$V = \frac{d^2}{4} \cdot p \cdot h \quad \Rightarrow \quad \#Contenu = \frac{\#Diamètre^2}{4} \cdot 3.14159 \cdot \#Niveau \cdot 1000$$

6.3 Extension de la fonction "Calculer_volume_citerne"

La deuxième étape vérifie si le diamètre est supérieur à zéro. Elle doit également vérifier si le niveau est supérieur ou égal à zéro et inférieur ou égal à la hauteur de la citerne. En cas d'erreur, le nouveau paramètre "er" est défini sur TRUE et le paramètre "Contenu" obtient la valeur -1.

Pour ce faire, ajoutez les paramètres "er" et "Hauteur" à l'interface.

Entrée	Type de données	Commentaire
Hauteur	REAL	Hauteur en mètres de la citerne cylindrique
Diamètre	REAL	Diamètre en mètres de la citerne cylindrique
Niveau	REAL	Niveau en mètres de la citerne cylindrique
Sortie		
er	BOOL	Indicateur d'erreur ; si erreur = TRUE
Contenu	REAL	Contenu de la citerne cylindrique en litres

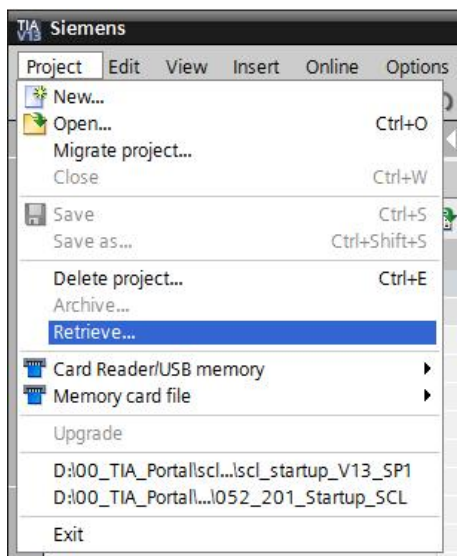
Tableau 3 : Paramètres pour le FC "Calculer_volume_citerne" dans la deuxième étape

7. Instructions structurées étape par étape

Vous trouverez ci-après des instructions vous indiquant comment réaliser la démarche pratique. Si vous possédez déjà une bonne compréhension générale, il vous suffit de vous concentrer sur les étapes numérotées. Sinon, suivez simplement les étapes détaillées suivantes des instructions.

7.1 Désarchivage d'un projet existant

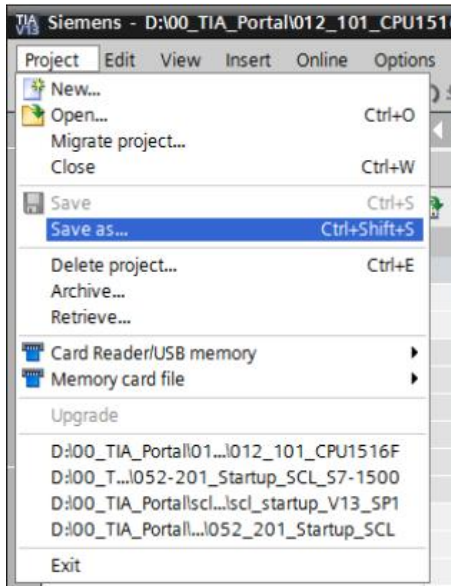
- ® Avant de pouvoir commencer la programmation, nous avons besoin d'un projet avec une configuration matérielle (par ex. SCE_FR_012-101_configuration_matérielle_S7-1516F_...zap). Pour désarchiver un projet existant, vous devez rechercher l'archive correspondante depuis la vue du projet sous ® Projet (Project) ® Désarchiver (Retrieve). Confirmez ensuite votre sélection avec Ouvrir (Open). (® Projet (Project) ® Désarchiver (Retrieve) ® sélection d'une archive .zap ® ouvrir)



- ® Le répertoire cible dans lequel le projet désarchivé doit être enregistré peut ensuite être sélectionné. Confirmez votre sélection par "OK". (® Projet (Project) ® Enregistrer sous (Save as) ® OK)

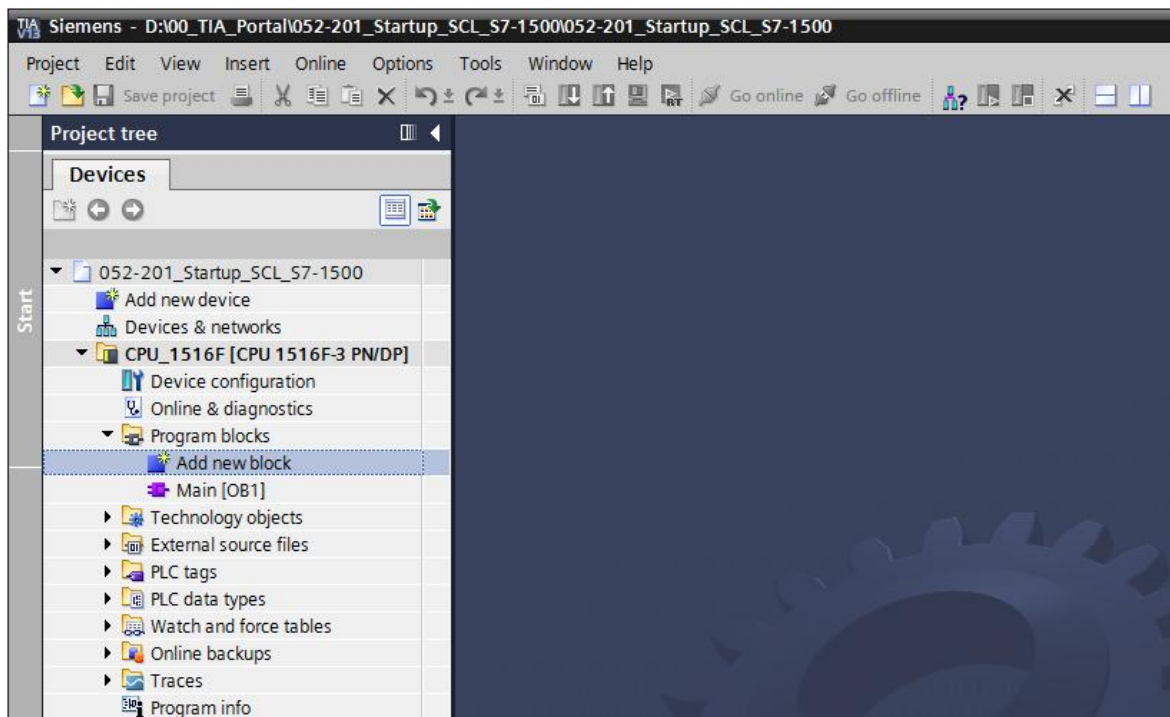
7.2 Enregistrement du projet sous un autre nom

- ® Vous enregistrez le projet ouvert sous le nom 052-201_Startup_SCL.
 (® Projet (Project) ® Enregistrer sous ... (Save as ...) ® 052-201_Startup_SCL ® Enregistrer (Save))



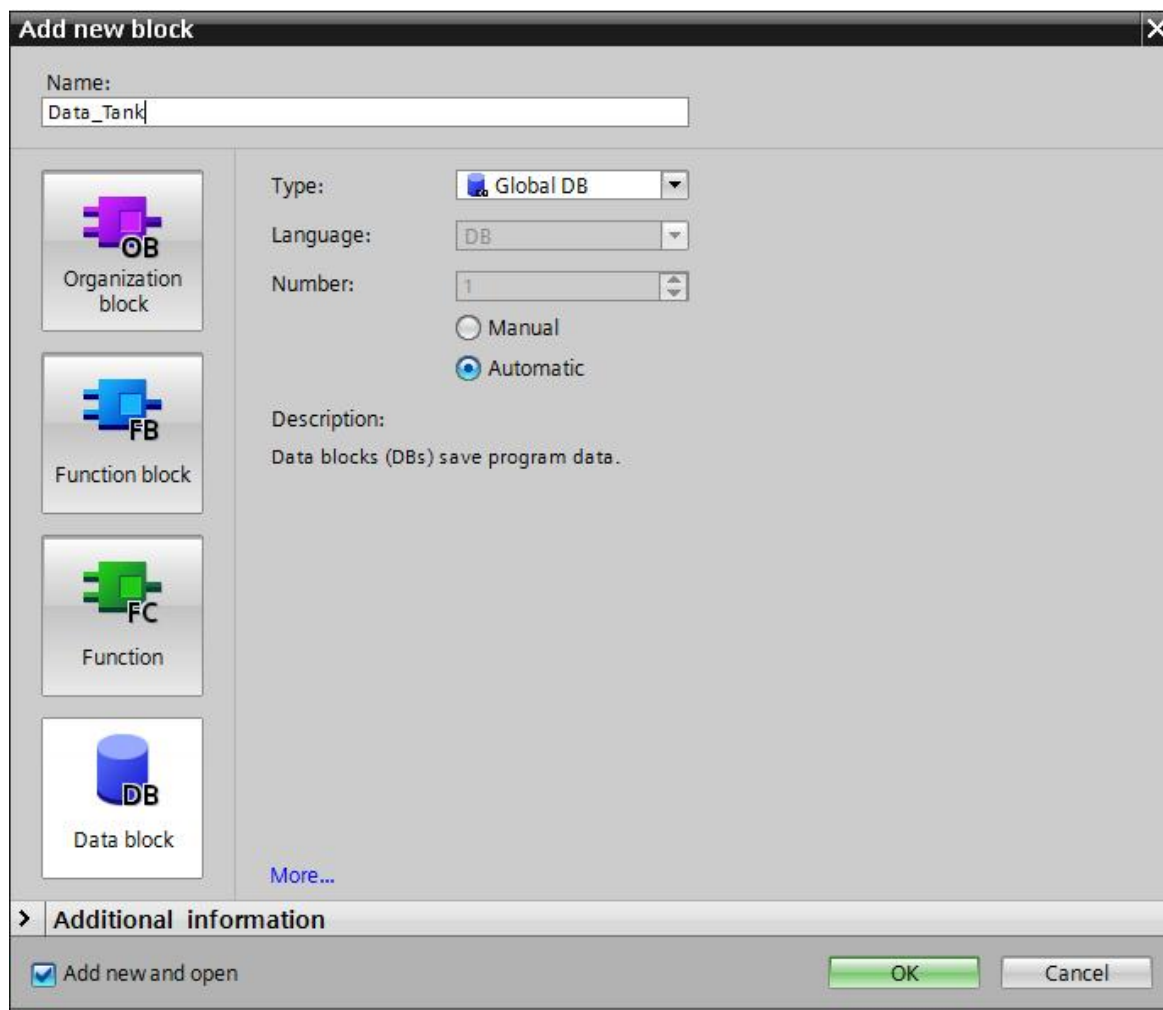
7.3 Création du bloc de données "Données_citerne"

- ® Dans la vue du projet, allez jusqu'à ® Blocs de programme (Program blocks) et créez un nouveau bloc en double-cliquant sur ® Ajouter nouveau bloc (Add new block).



® Sélectionnez maintenant un bloc de données et attribuez lui un nom.

(®  ® "Données_citerne" ("Data_Tank") ® OK)



- ® Entrez maintenant les noms de variables indiqués ci-dessous avec le type de données, la valeur initiale et le commentaire.

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

	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 Création de la fonction "Calculer_volume" ("Calculate_volume")

- ® Ajoutez maintenant une fonction, indiquez un nom et sélectionnez la langue.

(® Add new block (Ajouter nouveau bloc) ®  Function ® "Calculer_volume" ("Calculate_volume")
 ® SCL ® OK)

Add new block

Name:

Language:

Number:

☐ Manual
☒ Automatic

Description:
 Functions are code blocks or subroutines without dedicated memory.

Organization block (OB)
 Function block (FB)
Function (FC)
 Data block (DB)

More...

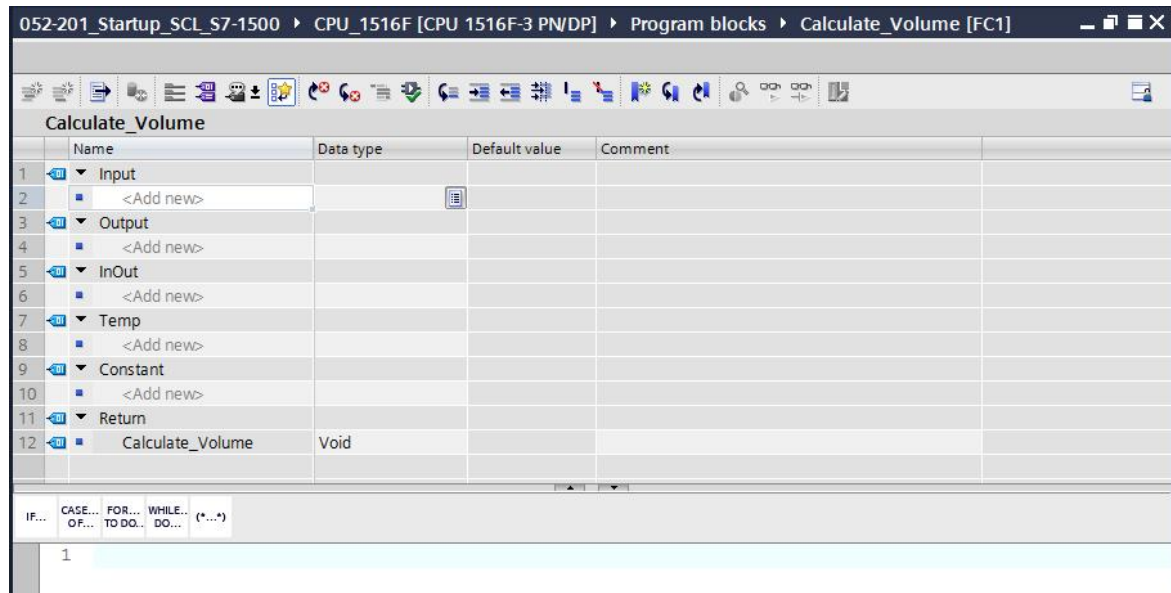
> Additional information

☒ Add new and open

OK Cancel

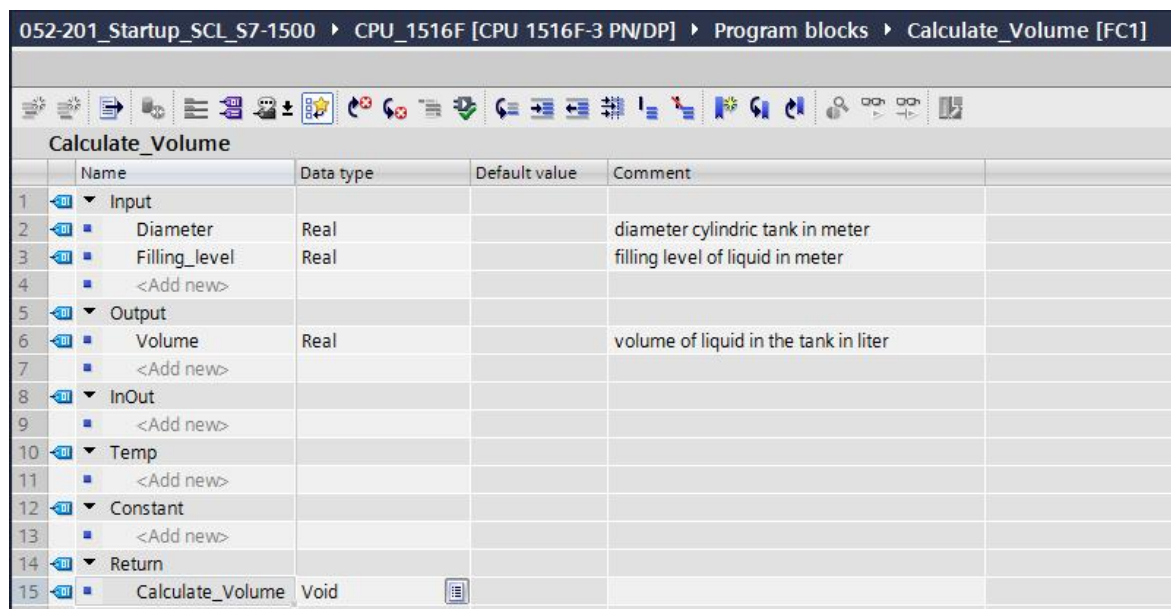
7.5 Définition de la fonction "Calculer_volume" ("Calculate_volume")

- ® Dans la partie supérieure de votre vue de programmation se trouve la description de l'interface de votre fonction.



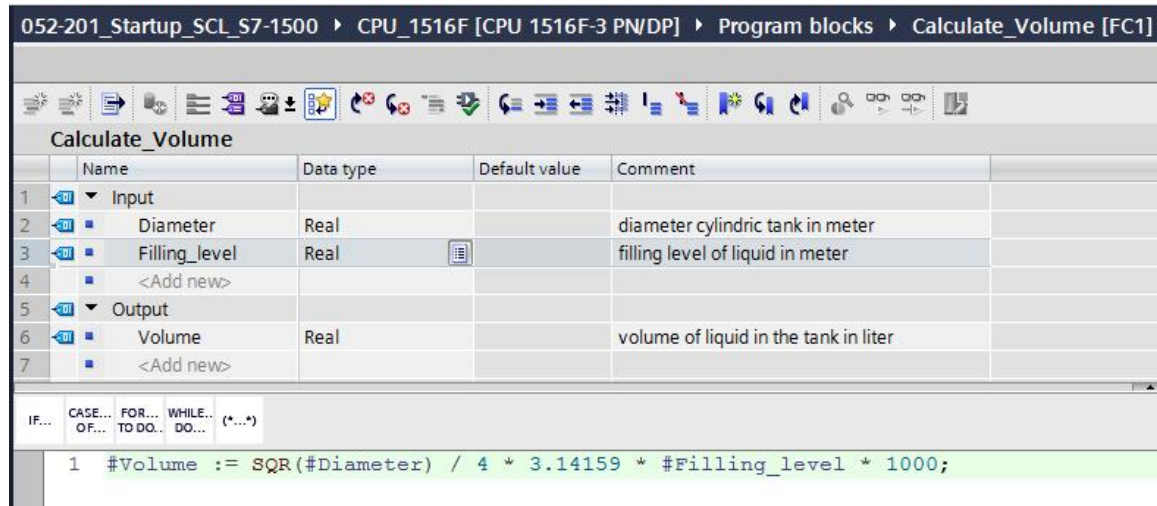
- ® Créez les paramètres d'entrée et de sortie suivants.

(® Nom (Name) ® Type de données (Data type) ® Commentaire (Comment))



7.6 Programmation de la fonction "Calculer_volume" ("Calculate_volume")

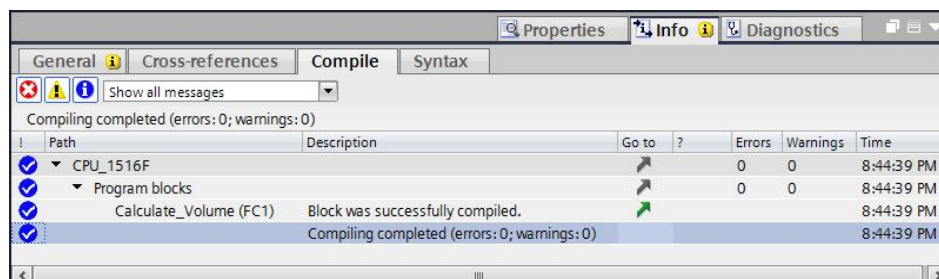
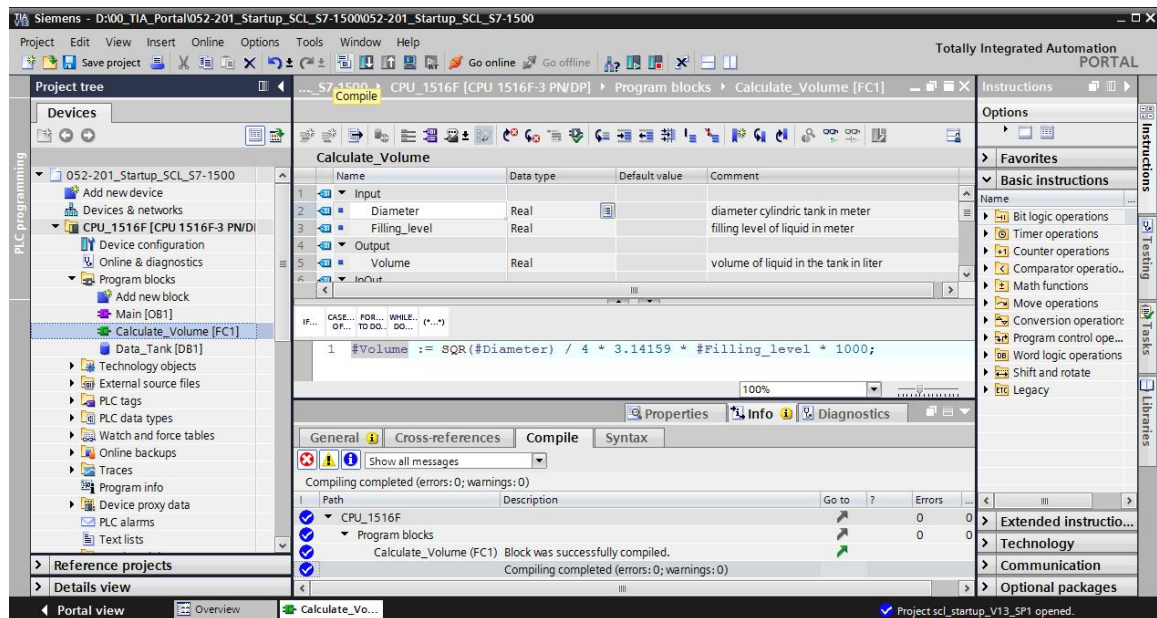
® Entrez le programme ci-dessous. (® Entrer le programme)



® Compilez maintenant votre programme et vérifiez si des erreurs de syntaxe sont présentes. Celles-ci sont affichées dans la fenêtre d'inspection sous la programmation. Corrigez les erreurs éventuelles et relancez ensuite la compilation. Puis, enregistrez votre programme. (®

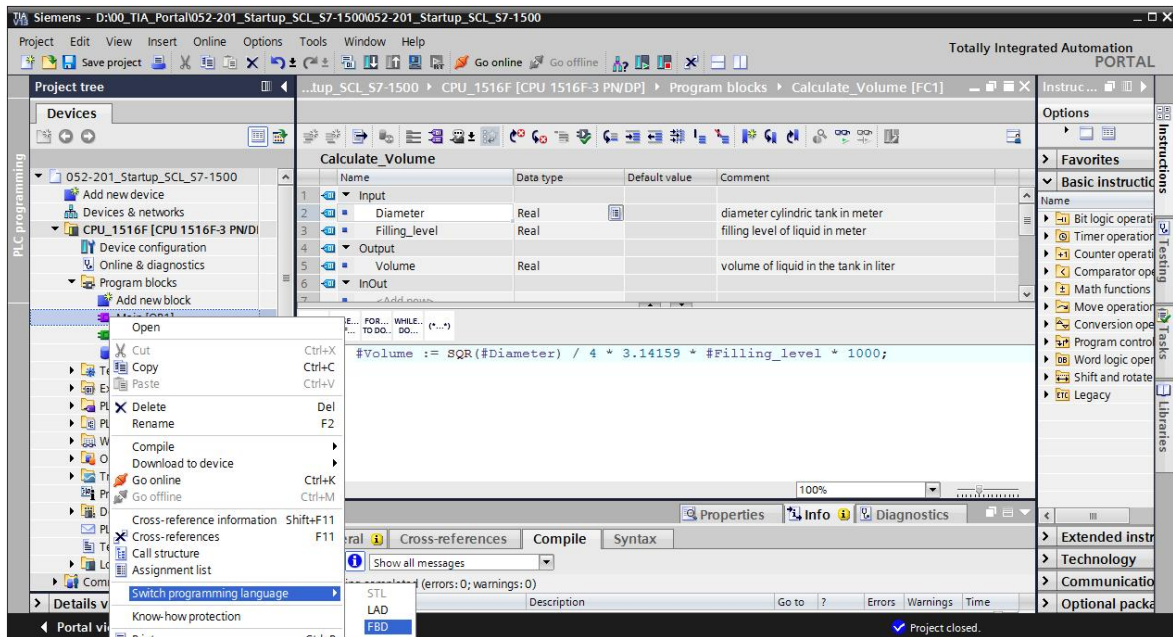


® Corriger les erreurs® Save project)

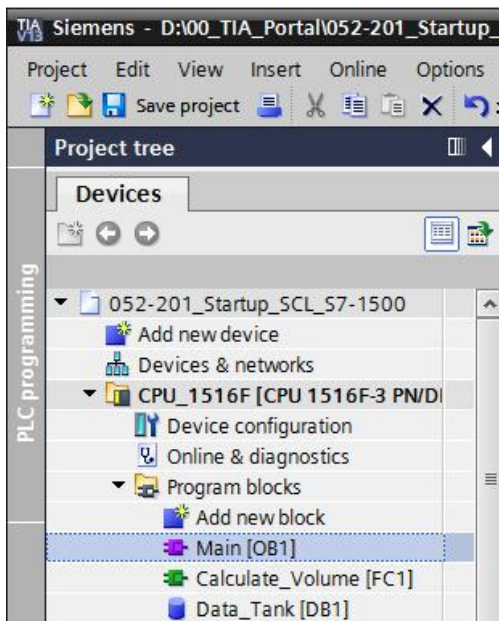


7.7 Programmation du bloc d'organisation "Main [OB1]"

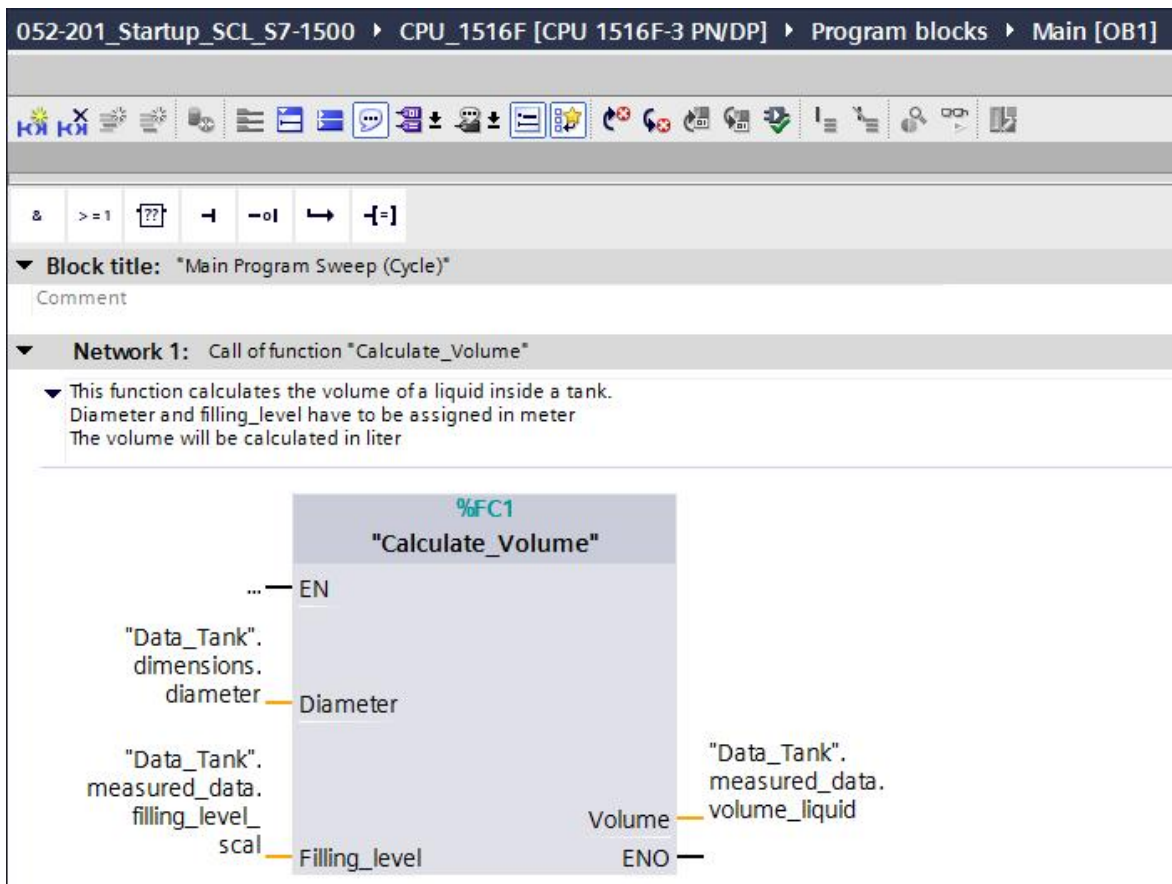
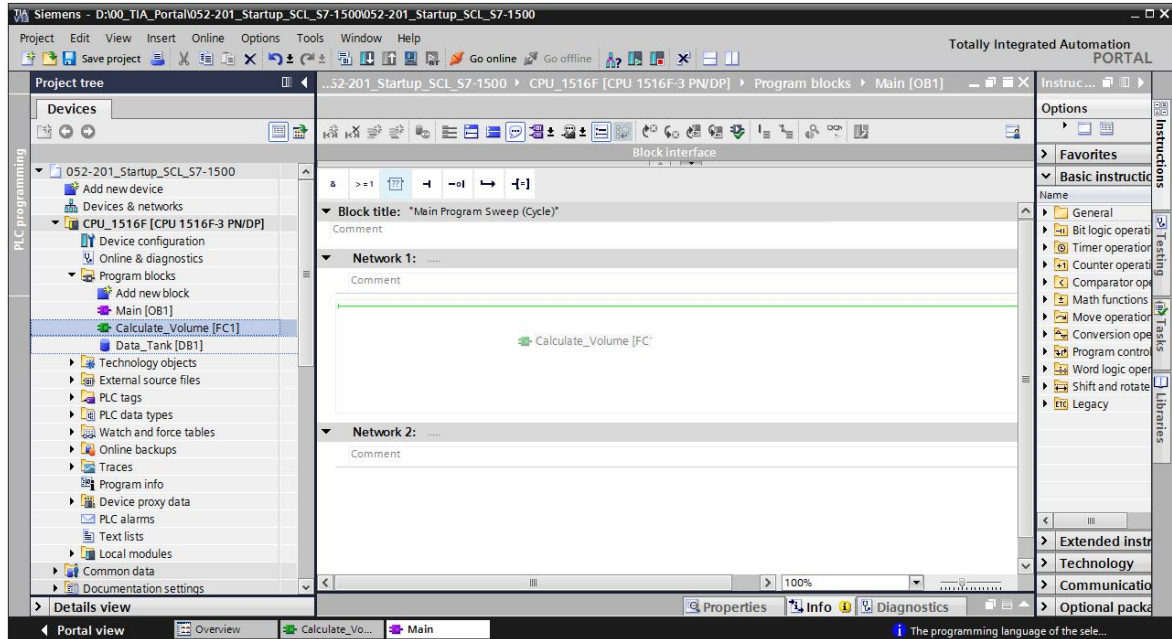
- ® Avant la programmation du bloc d'organisation ""Main [OB1]"" , nous basculons en langage de programmation LOG. Cliquez pour cela au préalable avec le bouton gauche de la souris sur ""Main [OB1]"" dans le dossier "Blocs de programme" ("Program Blocks").
- (® CPU_1516F[CPU 1516F-3 PN/DP] ® Blocs de programme (Program blocks) ® Main [OB1] ® Changer le langage de programmation (Switch programming language) ® LOG (FBD))



- ® Ouvrez maintenant le bloc d'organisation "Main [OB1]" à l'aide d'un double-clic.

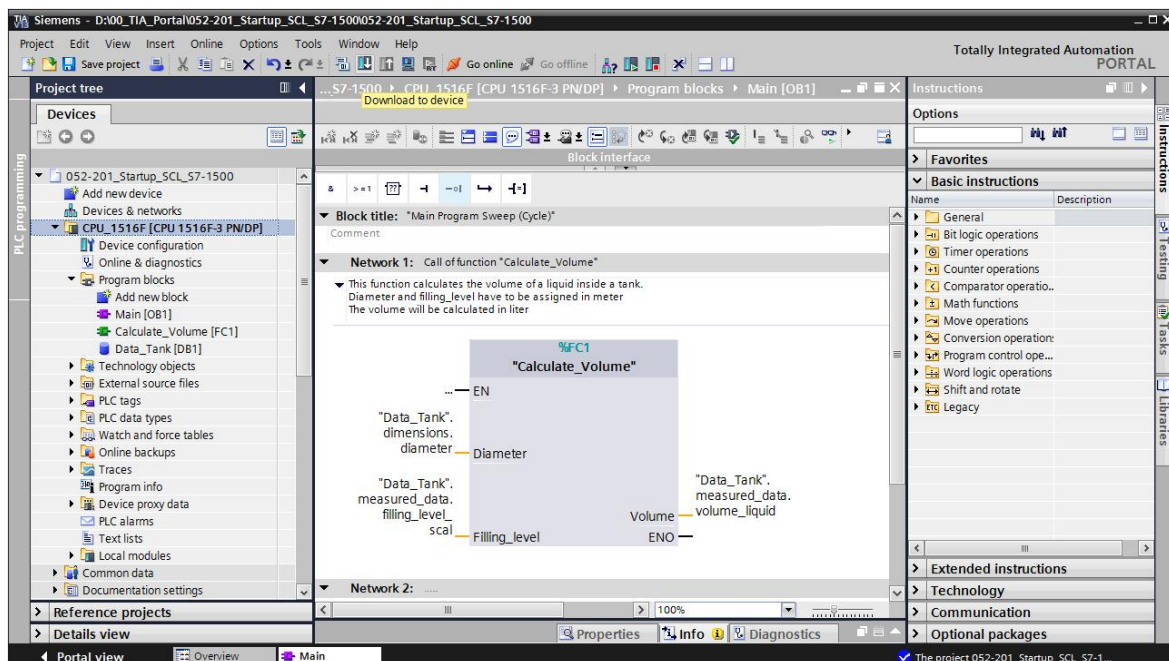


- ® Appelez la fonction "Calculer_Volume" ("Calculate_Volume") dans le premier réseau.
 Attribuez un titre de réseau, un commentaire et connectez les paramètres. Enregistrez ensuite votre projet. (® Appel "Calculer_Volume" ("Calculate_Volume") ® Attribuer titre de réseau ® Écrire un commentaire de réseau ® Connecter les paramètres ®  Save project)

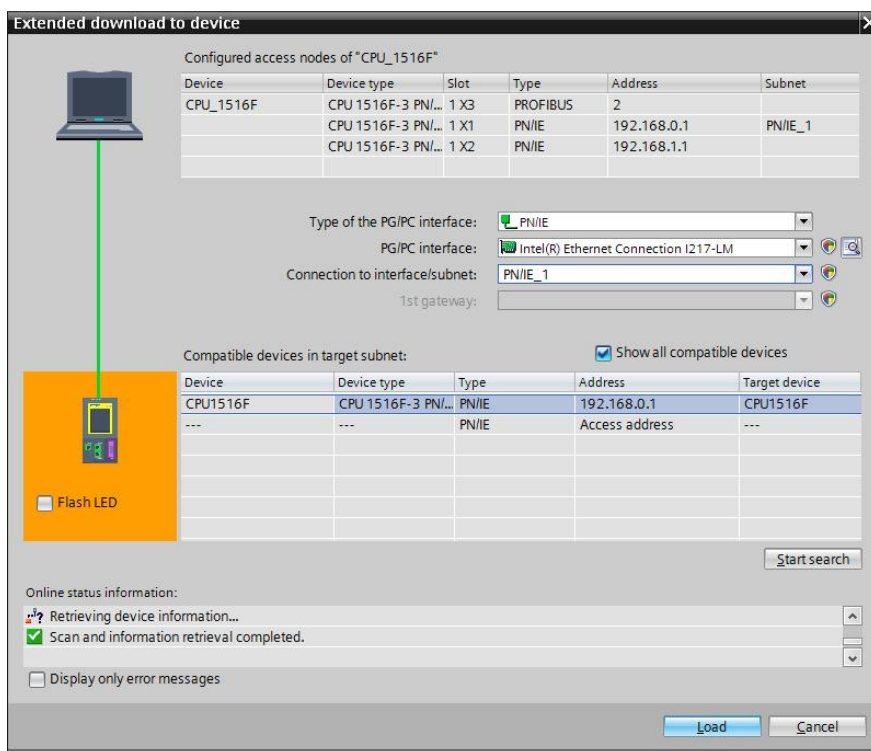


7.8 Compiler et charger le programme

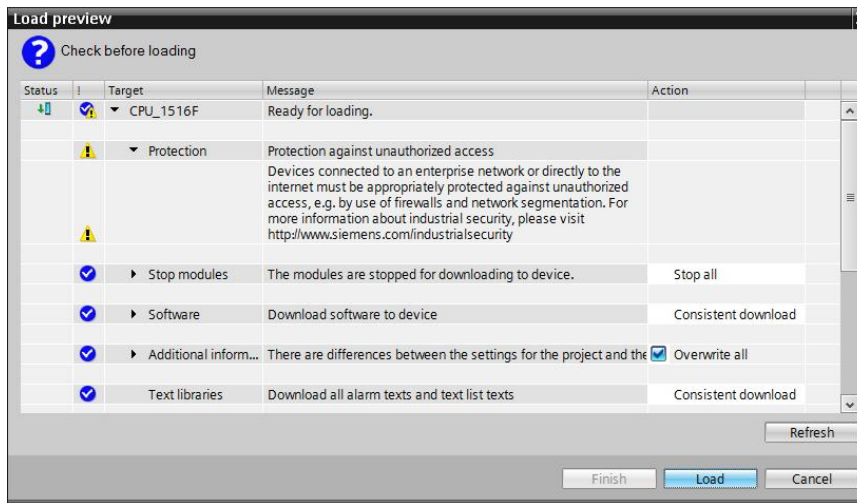
- ® Cliquez sur le dossier "Blocs de programme" ("Program blocks") et compilez le programme entier. Une fois la compilation terminée, chargez le projet dans l'automate. (®  ® )



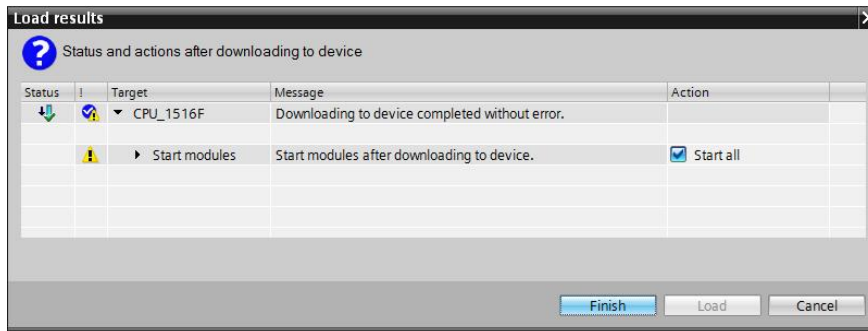
- ® Sélectionner interface PG/PC ® Sélectionner le sous-réseau ® Démarrer la recherche (Start search) ® Charger (Load)



® Effectuer éventuellement une sélection ® Charger (Load)

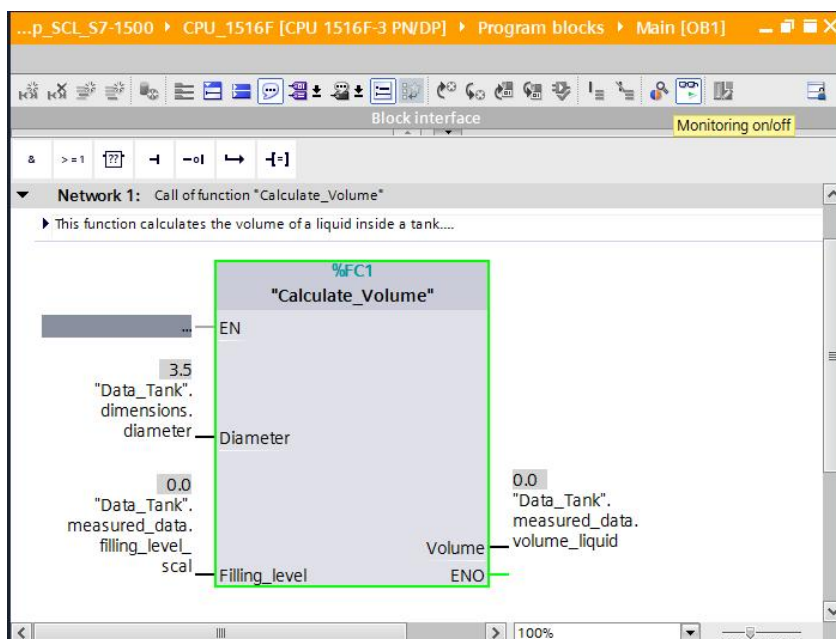


® Terminer (Finish)



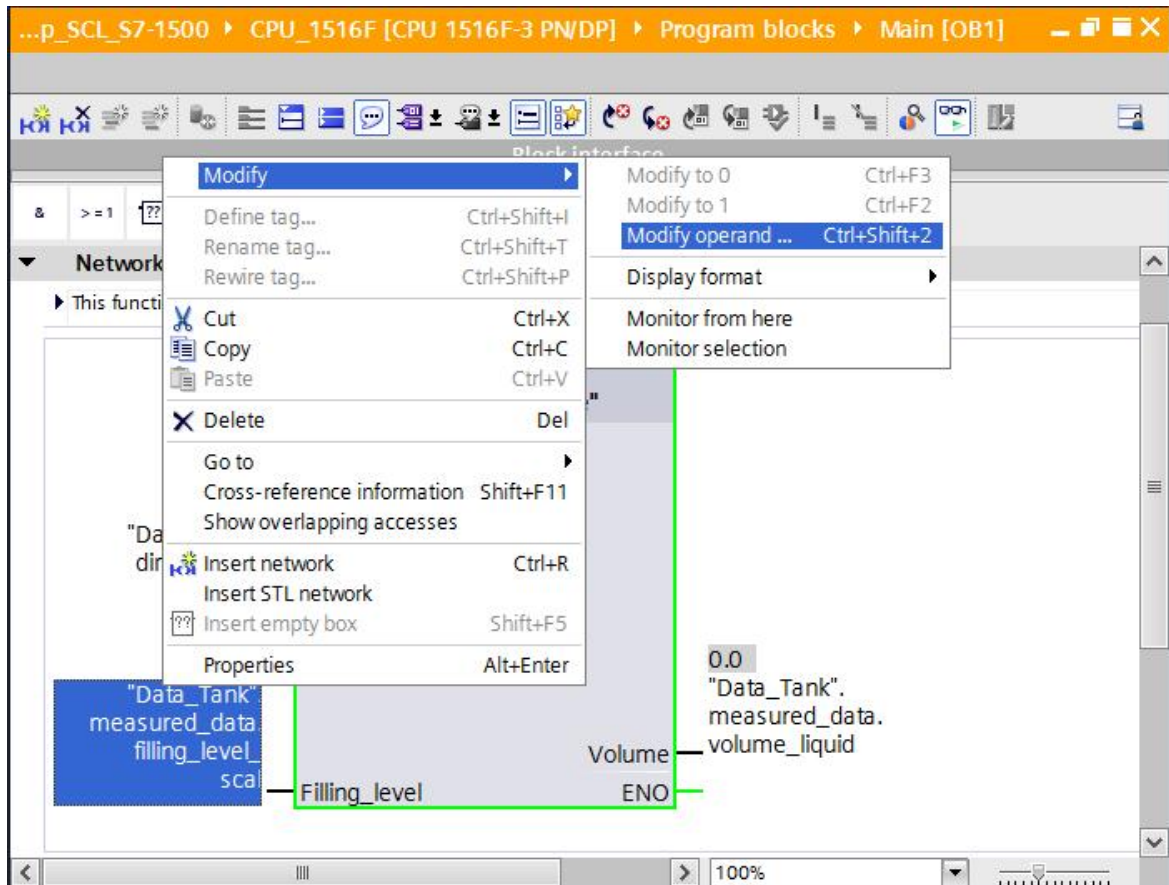
7.9 Visualiser et tester le bloc d'organisation

® Cliquez dans l'OB1 ouvert sur l'icône  pour visualiser le bloc.

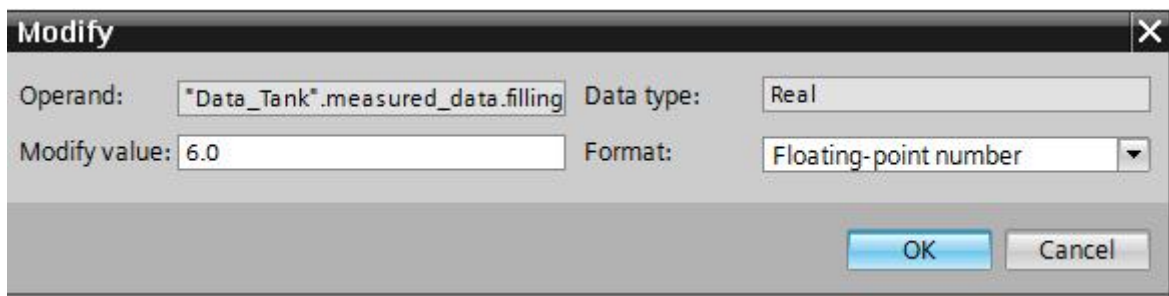


Testez votre programme en entrant une valeur dans la variable "Cadrage_niveau" dans le bloc de données.

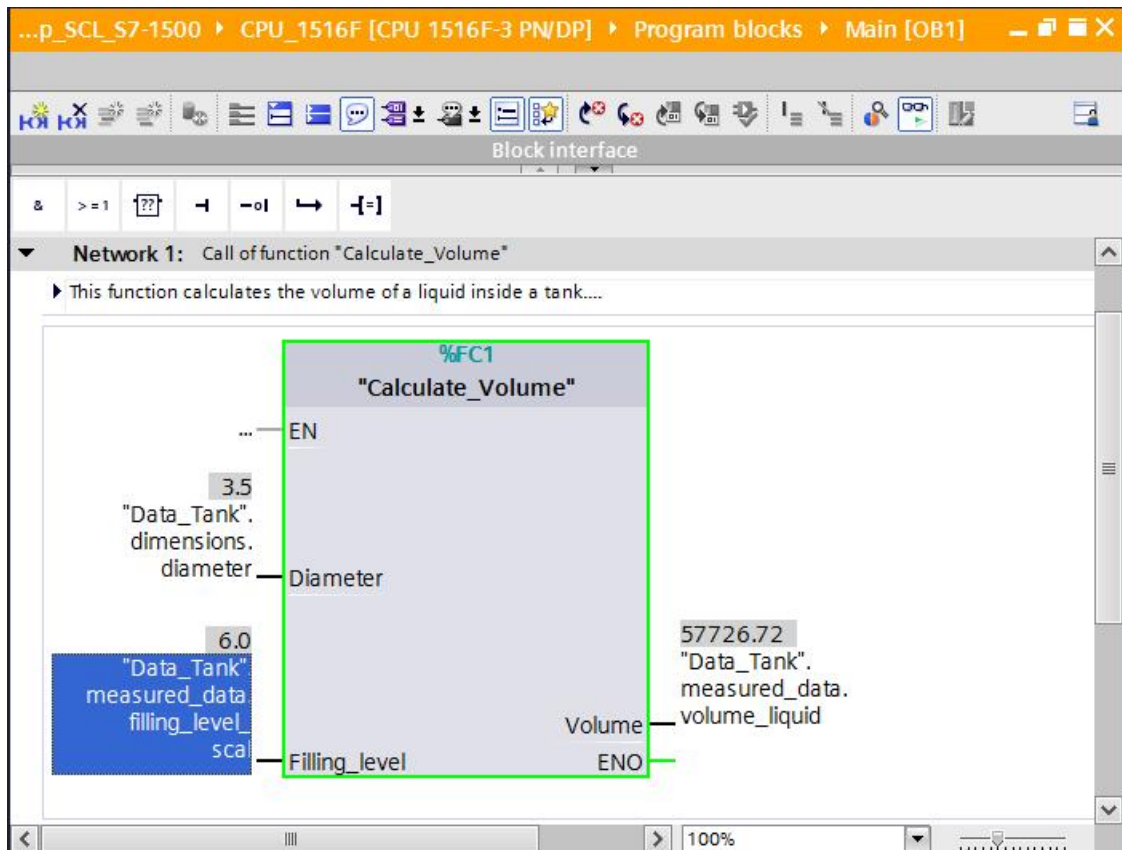
(® Clic droit sur "Cadrage_niveau" ("filling_level_scal") ® menu "Forcer" ("Modify") ® Forcer opérande (Modify operand ...)



(® Entrer la valeur 6,0 ® OK



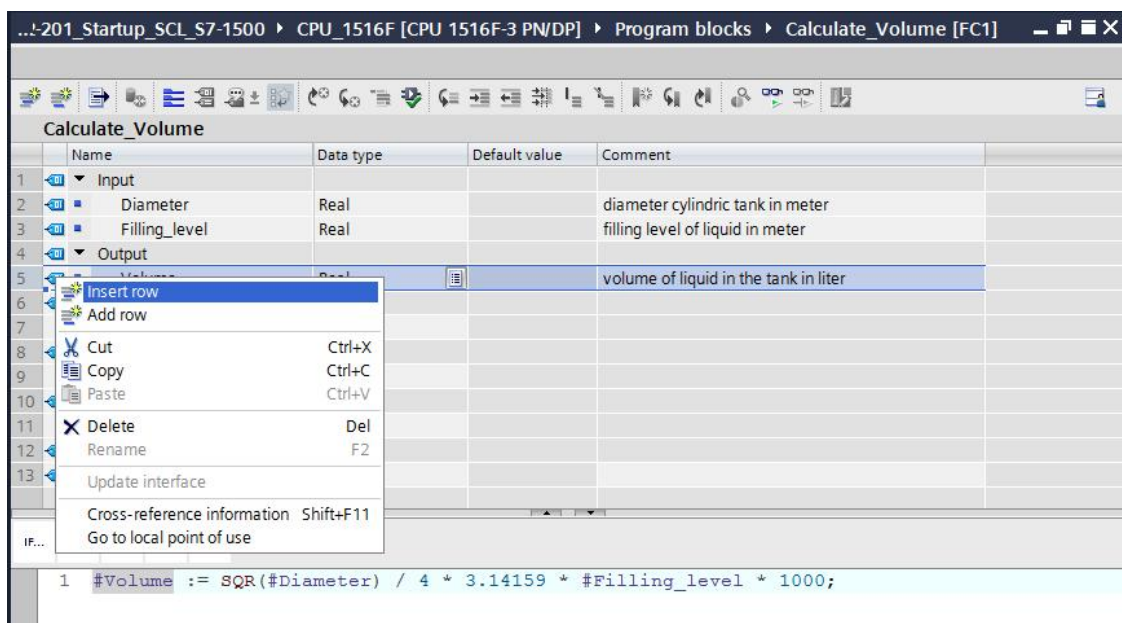
® Vérifiez si le résultat est correct.



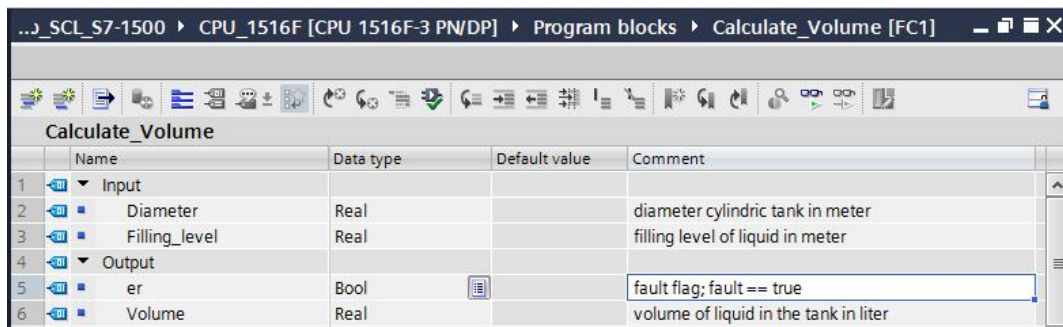
7.10 Extension de la fonction "Calculer_Volume" ("Calculate_Volume")

Ouvrez la fonction "Calculer_Volume" ("Calculate_Volume") et ajoutez une ligne pour les paramètres de sortie en cliquant avec le bouton droit de la souris sur la ligne dans l'interface.

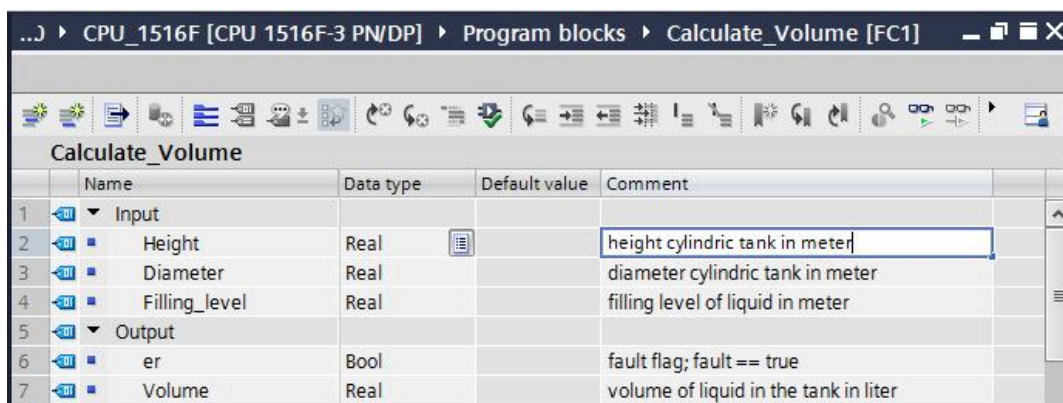
(® Ouvrir "Calculer_Volume" ("Calculate_Volume") ® clic droit sur la ligne 5 ® Insérer une ligne (Insert row))



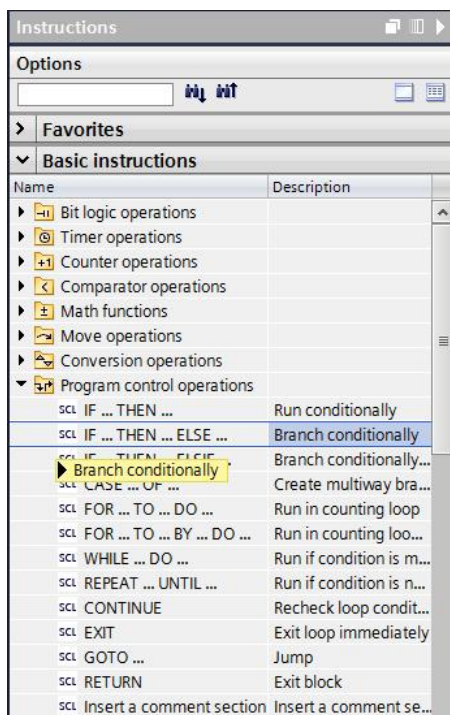
- ® Entrez le paramètre "er" avec le type de données BOOL et un commentaire.



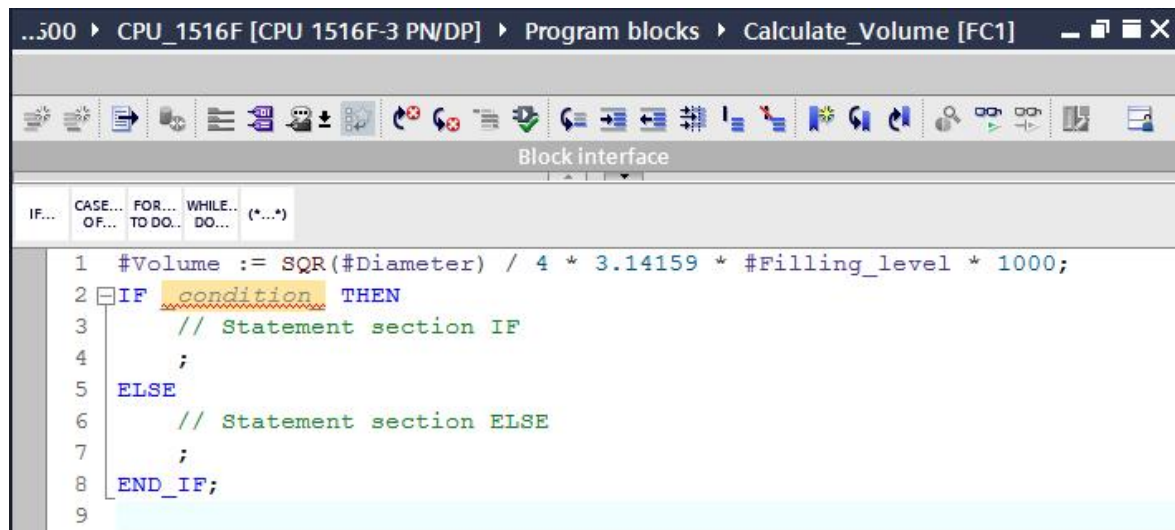
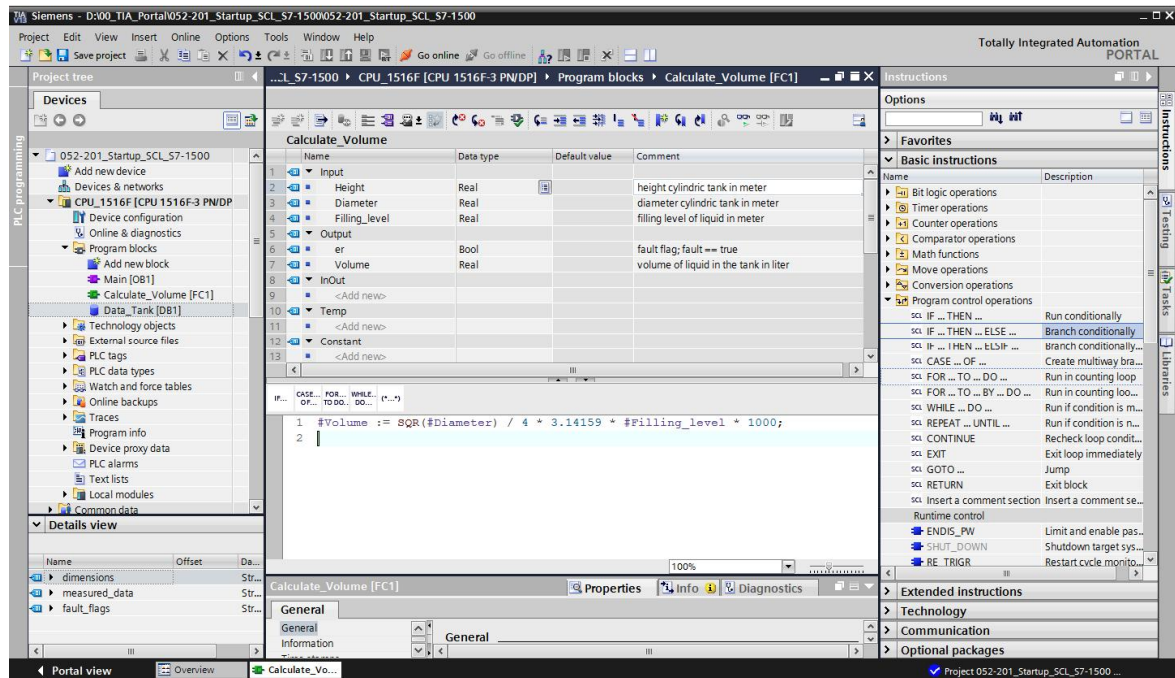
- ® De la même manière, ajoutez ensuite la variable "Hauteur" ("Height") avec le type de données Real et un commentaire.



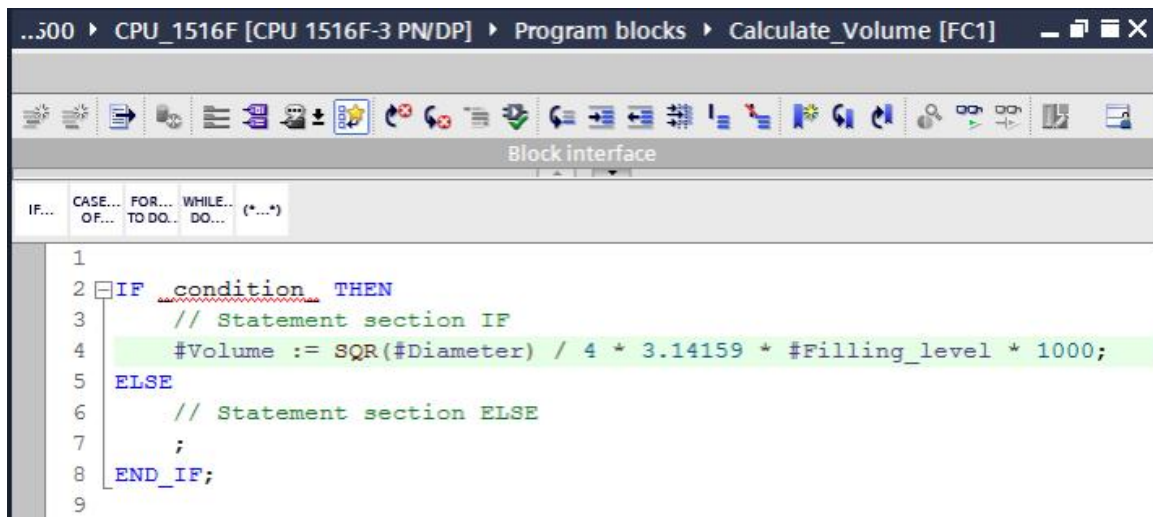
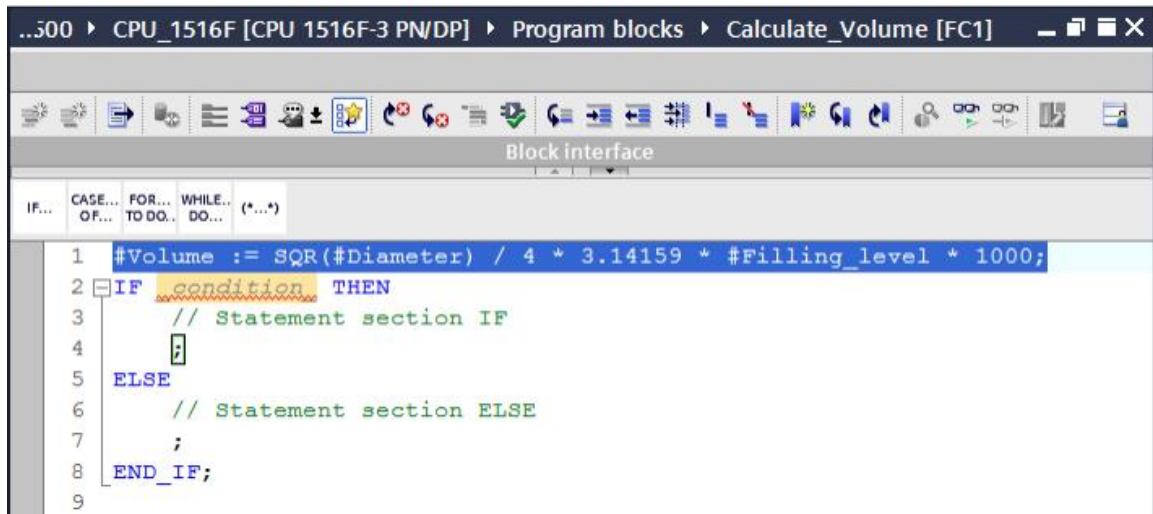
- ® Allez ensuite dans la structure de contrôle "IF...THEN...ELSE" du dossier "Gestion du programme" ("Program control operations") des instructions simples (Basic instructions).
 (® Instructions (Instructions) ® Instructions simples (Basic instructions) ® Gestion du programme (Program control operations) ® "IF...THEN...ELSE")



- ® Amenez ensuite la structure de contrôle "IF...THEN...ELSE" dans la deuxième ligne du programme par glisser-déposer. (® "IF...THEN...ELSE" ® Glisser-déposer)

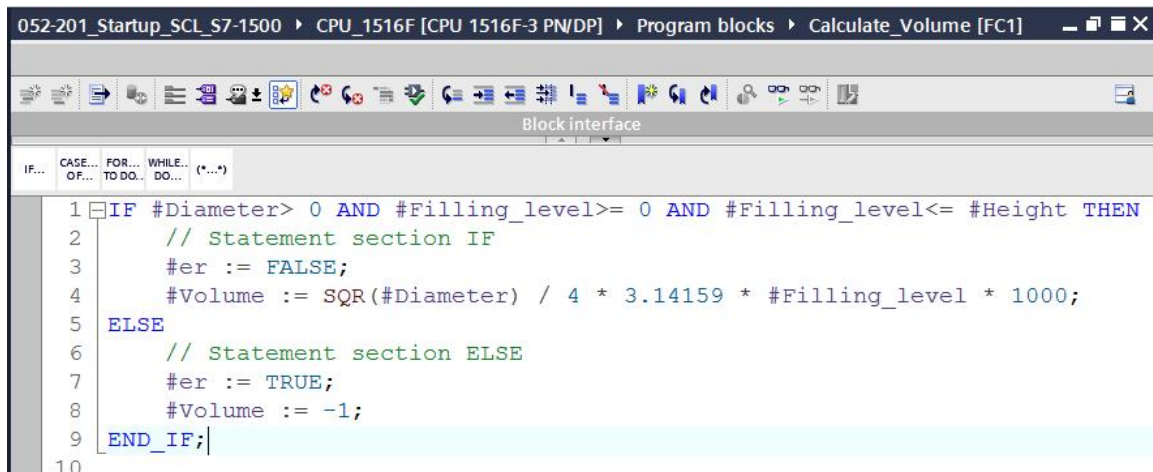


- ® Sélectionnez la formule mathématique et faites-la glisser sur le point-virgule devant ELSE.
(® sélectionner ® Glisser-déposer)



- ® Complétez la fonction et vérifiez votre programme par une compilation.

(® Compléter programme ® )



- ® Les commentaires peuvent être insérés avec (**) comme commentaire de bloc et avec "/" comme commentaire de ligne. Vous pouvez maintenant compléter votre programme par des commentaires.

(® Insérer un commentaire de bloc à partir de la ligne 1 ® Insérer un commentaire de ligne dans les lignes 12 et 16)

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

	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			

```

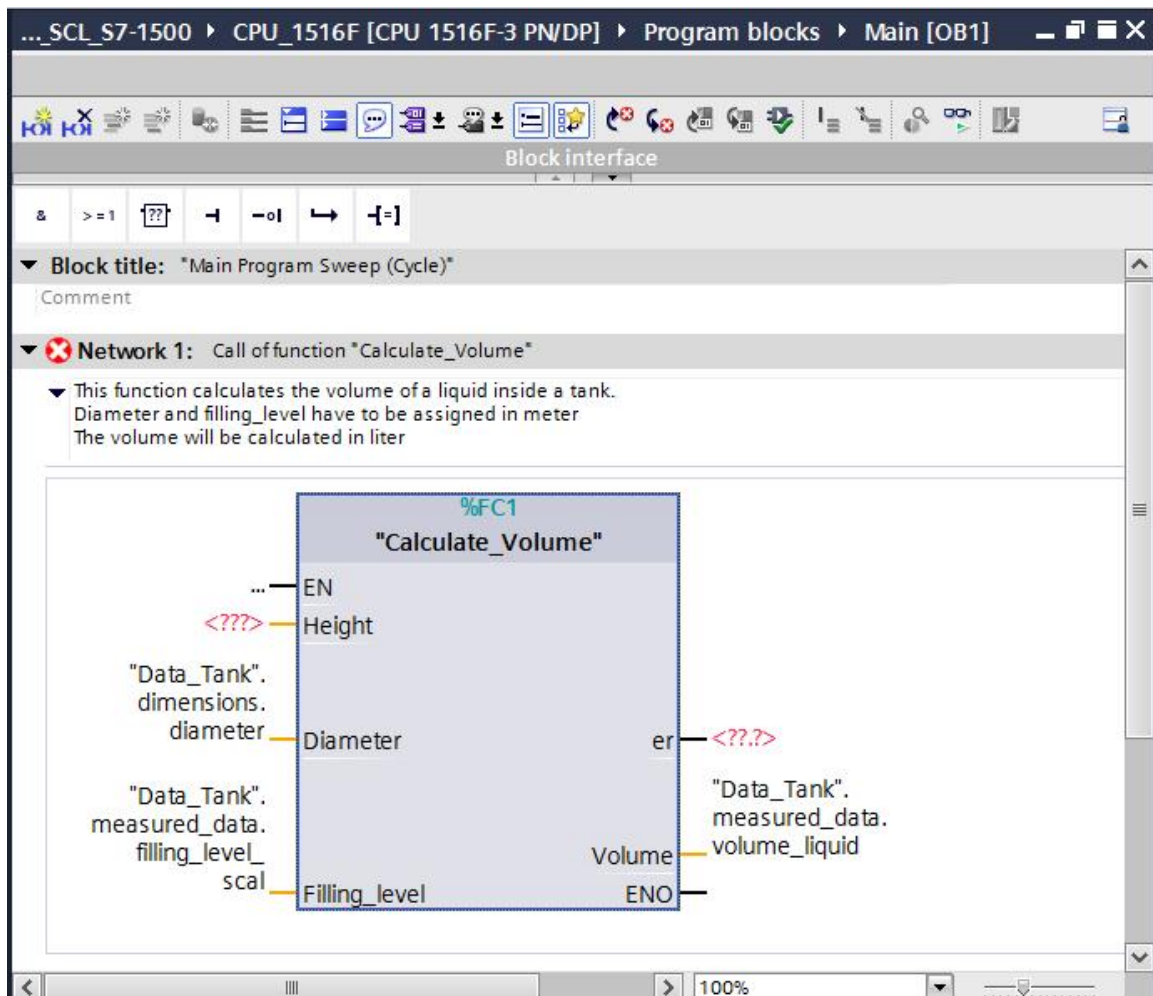
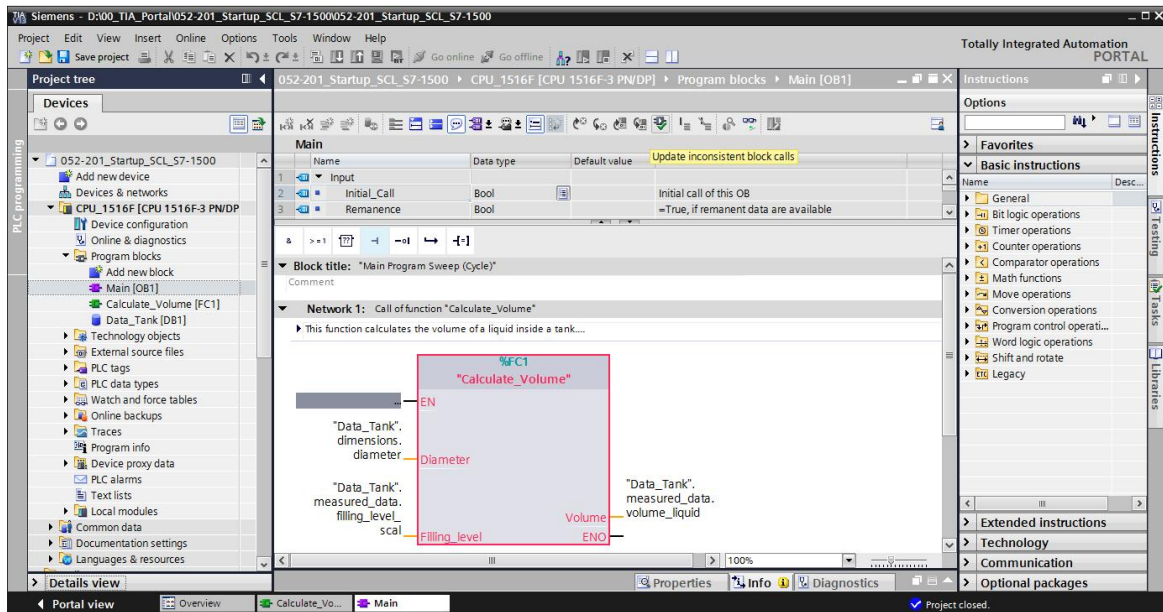
1  (*
2  This function calculates the volume of a liquid inside a tank.
3  Input-parameters #Height, #Filling_level and #Diameter have to be assigned in meter.
4  Output-parameter #Volume will be calculated in liter.
5  In case of an error the fault flag output-parameter #er will be set TRUE
6  and the 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 := SQRT(#Diameter) / 4 * 3.14159 * #Filling_level * 1000;
15 ELSE
16     // fault
17     #er := TRUE;
18     #Volume := -1;
19 END_IF;
20
  
```

100%

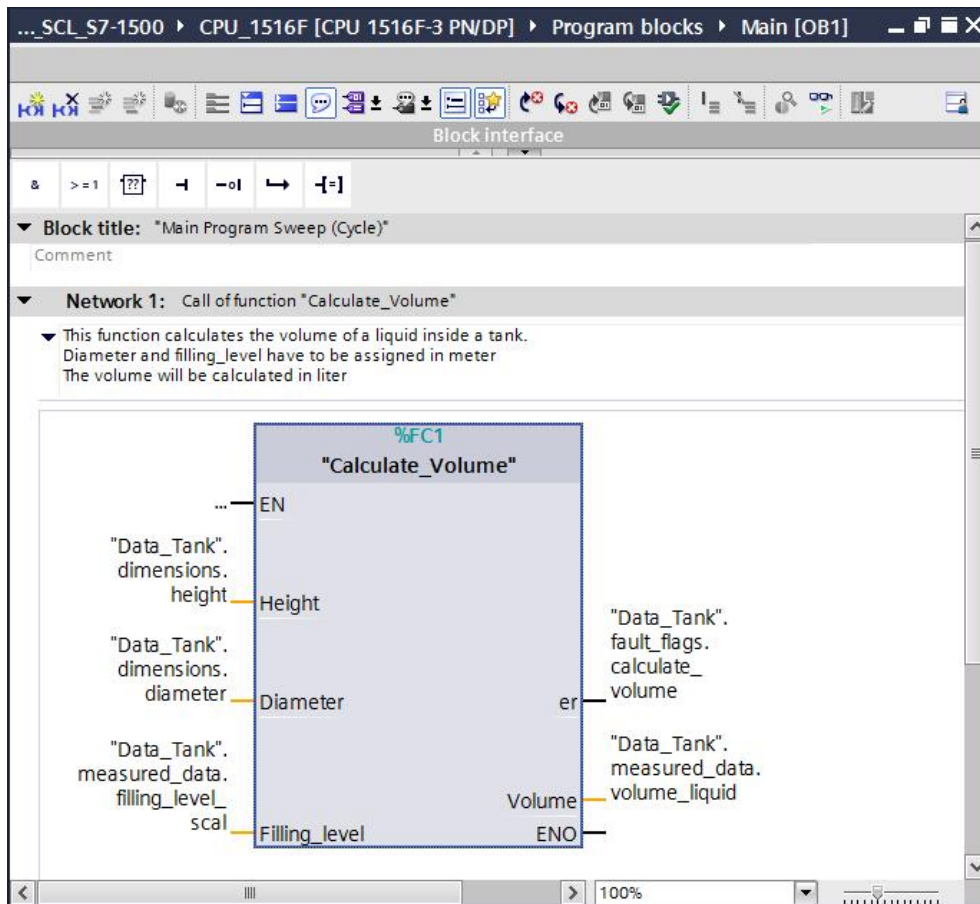
7.11 Modifier le bloc d'organisation

® Ouvrez l'OB1 et actualisez les appels de bloc incohérents en cliquant sur .

(® Ouvrir l'OB1 ® )

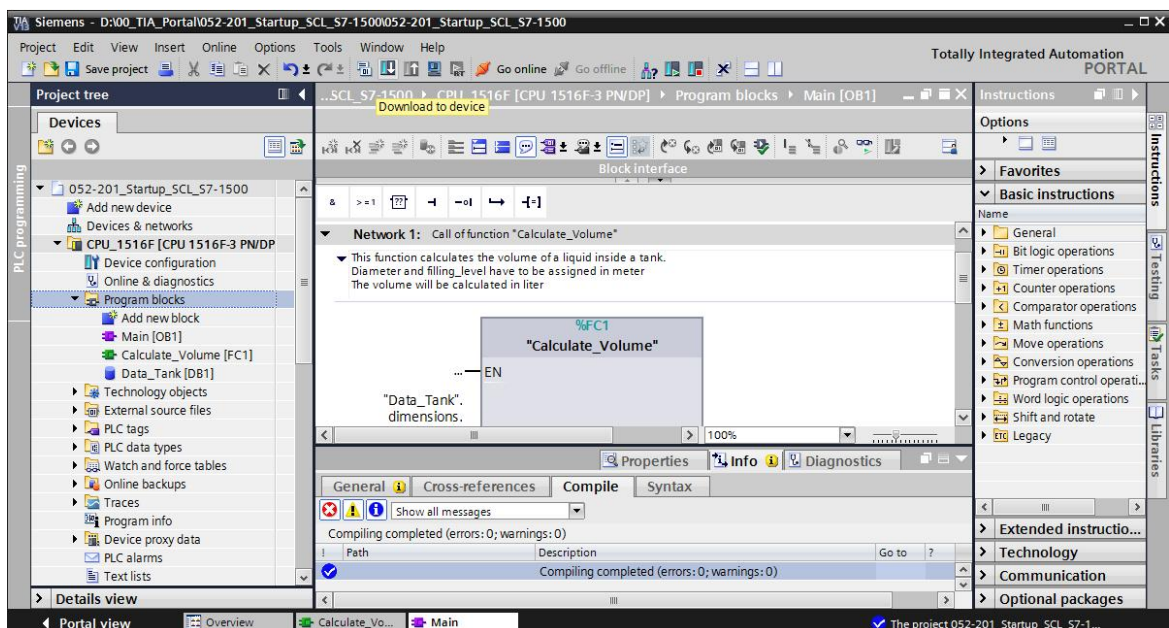


- ® Complétez les paramètres "er" et "Hauteur" ("Height").



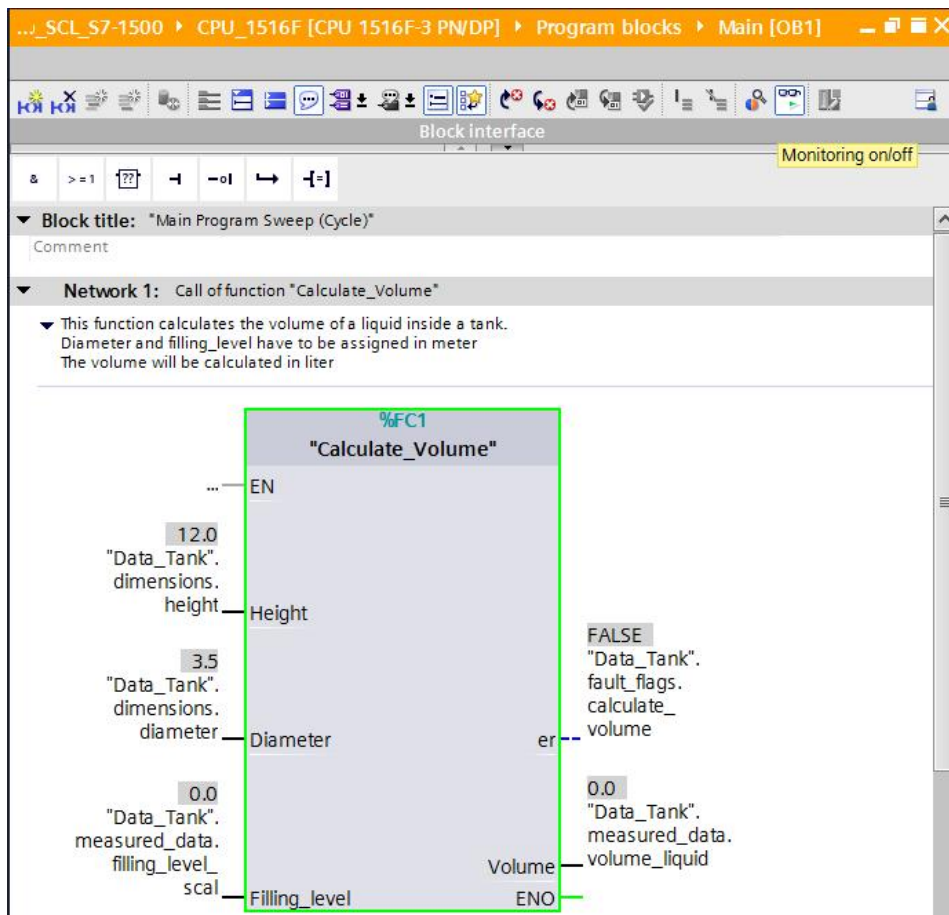
7.12 Compiler et charger le programme

- ® Cliquez sur le dossier "Blocs de programme" ("Program blocks") et compilez le programme entier. Une fois la compilation terminée, chargez le projet dans l'automate. Enregistrez ensuite votre projet. (® Blocs de programme ® ® Save project)

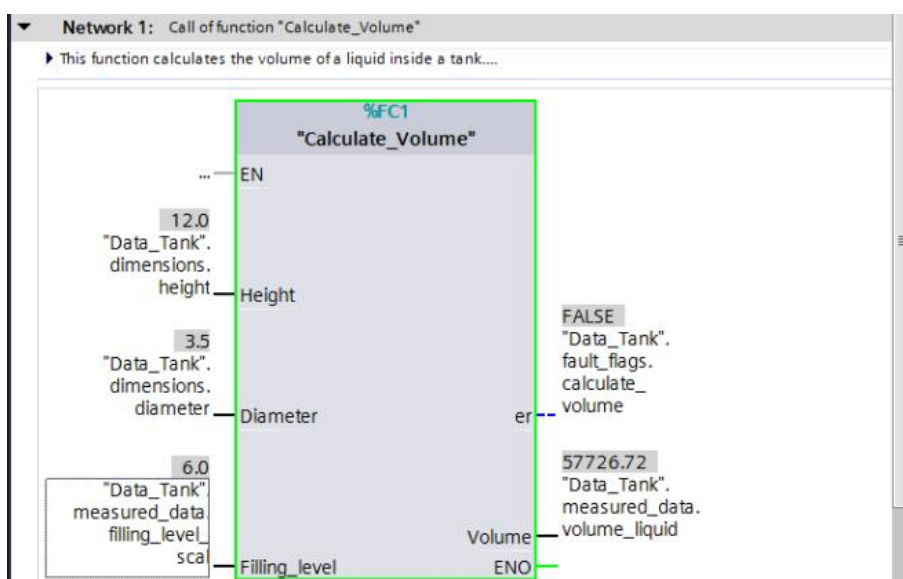


7.13 Visualiser et tester le bloc d'organisation

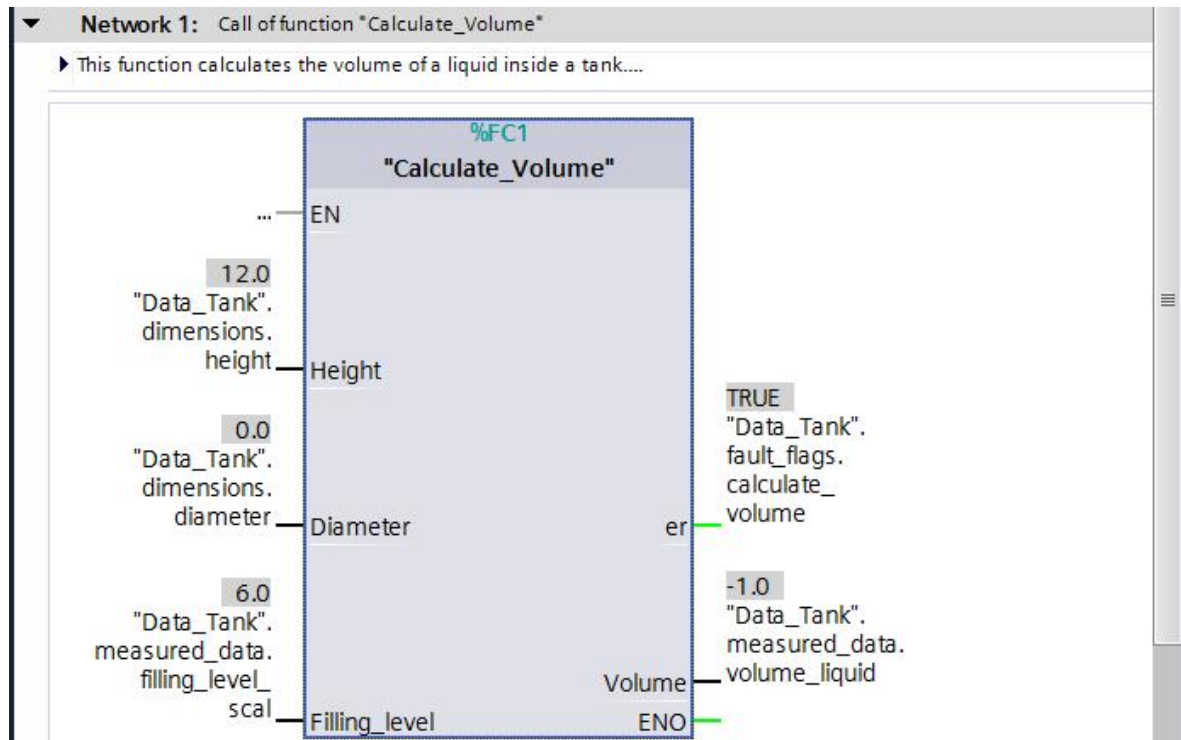
- ® Cliquez dans l'OB1 ouvert sur l'icône  pour visualiser le bloc.



- ® Testez votre programme en entrant une valeur dans la variable "Cadrage_niveau" dans le bloc de données. (® Clic droit sur "Cadrage_niveau" ("filling_level_scal") ® menu "Forcer" ("Modify") ® Forcer opérande (Modify operand ...) ® entrer la valeur 6,0 ® OK ® vérifier)



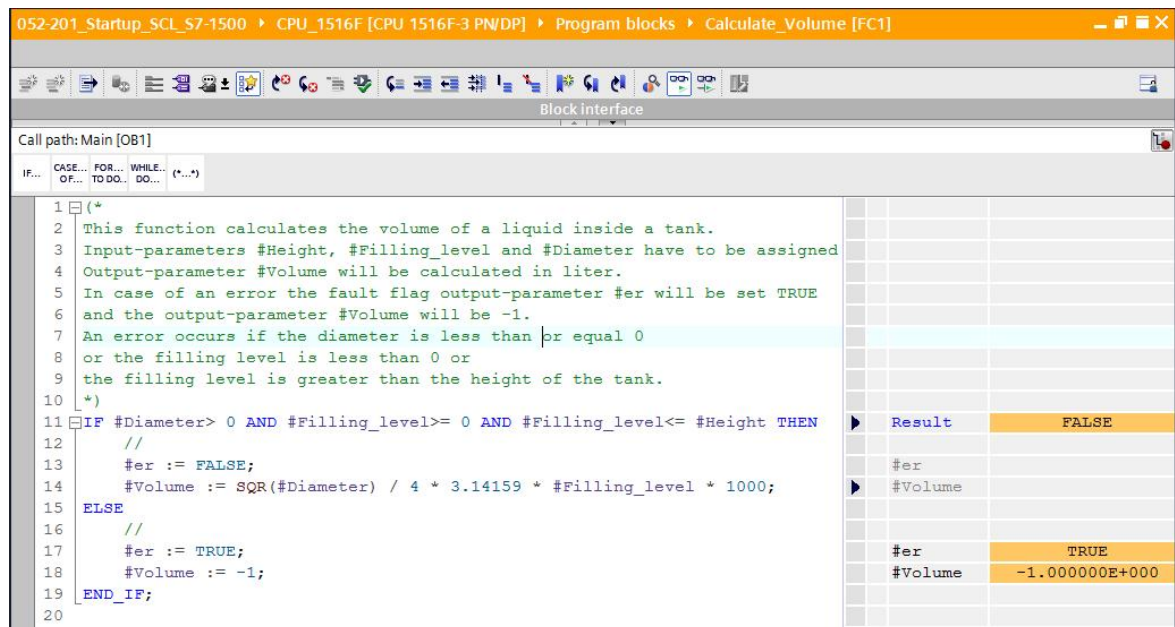
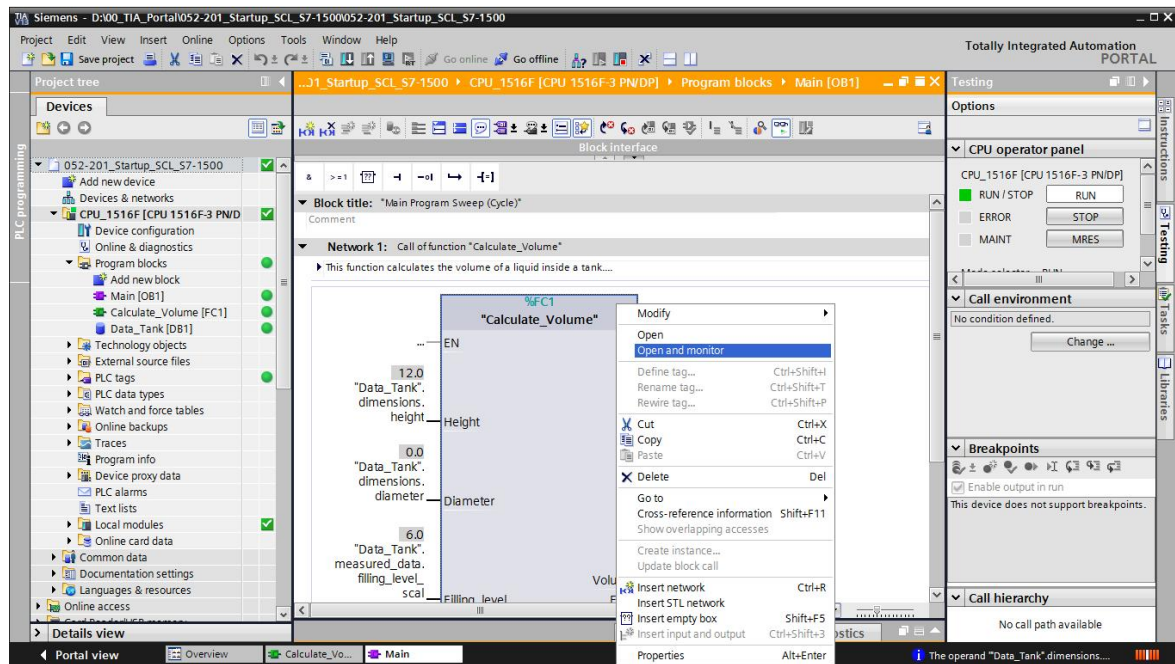
- ® Testez ensuite si une erreur s'affiche lorsque vous réglez le diamètre sur zéro.
 (® Clic droit sur "Diamètre" ("Diameter") ® menu "Forcer" ("Modify") ® Forcer opérande
 (Modify operand ...) ® entrer la valeur 0,0 ® OK ® vérifier)



7.14 Visualiser et tester la fonction "Calculer_Volume" ("Calculate_Volume")

- ⑧ Enfin, ouvrez et visualisez la fonction "Calculer_Volume" ("Calculate_Volume") en cliquant avec le bouton droit de la souris sur la fonction et en sélectionnant la commande "Ouvrir et visualiser" ("Open and monitor").

(⑧ Clic droit sur la fonction ⑧ "Ouvrir et visualiser" ("Open and monitor"))



- ® Vous pouvez afficher les valeurs des différentes variables de la requête IF en cliquant sur la flèche noire ▼. (® ▼)

▼	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

The screenshot shows the Siemens TIA Portal interface for the 'Calculate_Volume [FC1]' function block. The main editor displays the following LAD code:

```

10 (*)
11 IF #Diameter > 0 AND #Filling_level >= 0 AND #Filling_level <= #Height THEN
12
13     //
14     #er := FALSE;
15     #Volume := SQR(#Diameter) / 4 * 3.14159 * #Filling_level * 1000;
16
17 ELSE
18     //
19     #er := TRUE;
20     #Volume := -1;
21 END_IF;

```

The right-hand pane shows the 'Block interface' with the following table:

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

The bottom status bar indicates: "The operand 'Data_Tank' dimensions..."

- ® Le format d'affichage peut être modifié en cliquant avec le bouton droit de la souris sur la variable. (® Clic droit sur la variable ® Format d'affichage (Display format) ® Virgule flottante (Floating-point))

Variable	Value
Result	FALSE
#Diameter	0.000000E+000
#Filling_level	
#Filling_level	
#Height	
#er	
#Volume	
#er	TRUE
#Volume	-1.000000E+000

Context menu options:

- Display format ▸
 - Automatic
 - Decimal
 - Hexadecimal
 - Floating-point
- Cut Ctrl+X
- Copy Ctrl+C
- Paste Ctrl+V
- Delete Del

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

- ® Testez maintenant l'autre branche de la branche IF en redéfinissant le diamètre dans l'OB1 sur 3,5 mètres. (® Ouvrir l'OB1 ® Modifier le diamètre sur 3,5 ® Ouvrir et visualiser la fonction)

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... FOR... WHILE... OF... TO DO... DO... (*...*)

```

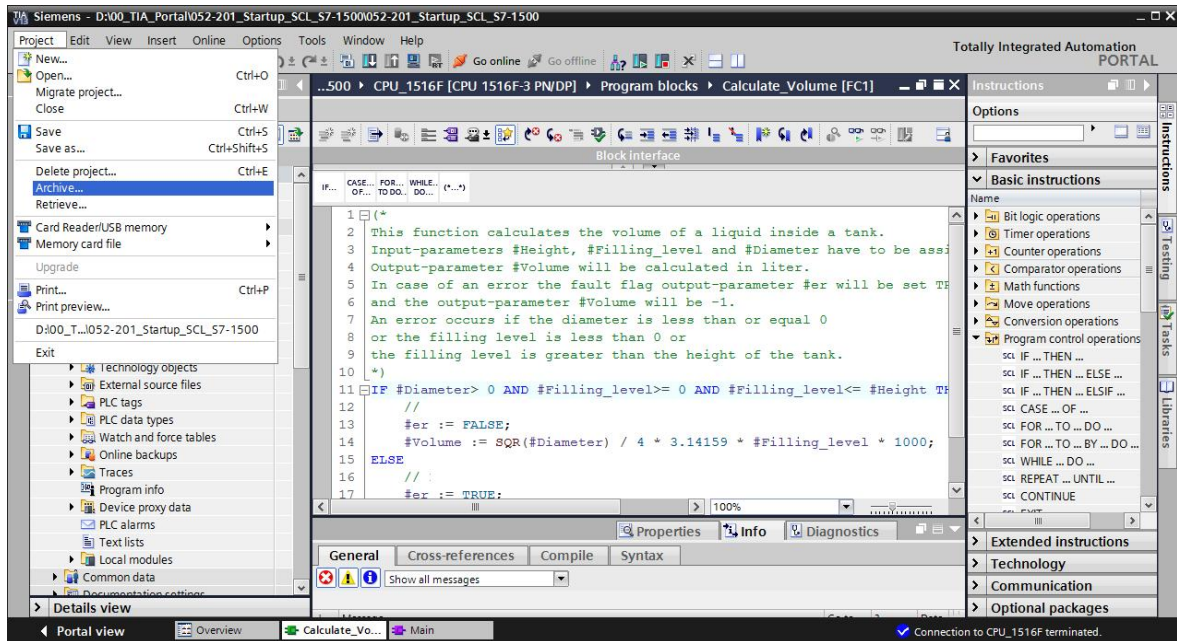
10 [*]
11 IF #Diameter > 0 AND #Filling_level >= 0 AND #Filling_level <= #Height THEN
12
13     //
14     #er := FALSE;
15     #Volume := SQR(#Diameter) / 4 * 3.14159 * #Filling_level * 1000;
16
17 ELSE
18     //
19     #er := TRUE;
20     #Volume := -1;
21 END_IF;

```

Variable	Value
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 Archivage du projet

- ® Pour finir, il vous faut encore archiver le projet complet. Veuillez sélectionner ® 'Projet' ('Project') ® 'Archiver...' ('Archive...'). Ouvrez le dossier dans lequel vous souhaitez archiver votre projet et enregistrez-le en choisissant le type de fichier 'Archives de projet TIA Portal' ('TIA Portal Project archives'). (® Projet (Project) ® Archiver (Archive) ® Archives de projet TIA Portal (TIA Portal Project archives) ® SCE_FR_052-201 Startup SCL_S7-1500... ® Enregistrer (Save))



8. Liste de contrôle

N°	Description	Contrôlé
1	Compilation réussie et sans message d'erreur	
2	Chargement réussi et sans message d'erreur	
3	Forçage de l'opérande (diamètre = 0,0) Résultat de la variable Volume = -1 Résultat de la variable "er" = TRUE	
4	Forçage de l'opérande (diamètre = 3,5 et cadrage_niveau = 0) Résultat de la variable Volume = 0 Résultat de la variable "er" = FALSE	
5	Forçage de l'opérande (cadrage_niveau = 6,0) Résultat de la variable Volume = 57726,72 Résultat de la variable "er" = FALSE	
6	Forçage de l'opérande (cadrage_niveau = 12,0) Résultat de la variable Volume = 115453,4 Résultat de la variable "er" = FALSE	
7	Forçage de l'opérande (cadrage_niveau = 14,0) Résultat de la variable Volume = -1 Résultat de la variable "er" = TRUE	
8	Projet archivé avec succès	

9. Exercice

9.1 Énoncé – Exercice

Une fonction "Cadrage" est programmée dans cet exercice. Le programme doit être applicable de manière générale pour toute valeur analogique positive. Dans notre exemple de tâche "Citerne", le niveau est lu par un capteur analogique et stocké sous forme de valeur cadrée dans le bloc de données à l'aide de cette fonction.

En cas d'erreur, le bloc doit placer l'indicateur d'erreur "er" sur TRUE et définir en conséquence le paramètre "Cadrage_val_analog" sur zéro. Une erreur apparaît si le paramètre "mx" est inférieur ou égal à "mn".

La fonction doit contenir les paramètres suivants.

Entrée	Type de données	Commentaire
Val_analog_pér	INT	Valeur analogique de la périphérie entre 0..27648
mx	REAL	Maximum de la nouvelle échelle
mn	REAL	Minimum de la nouvelle échelle
Sortie		
er	BOOL	Indicateur d'erreur, pas d'erreur = 0, erreur = 1
Cadrage_val_analog	REAL	Valeur analogique cadrée entre mn..mx En cas d'erreur = 0

La formule suivante est utilisée pour accomplir la tâche :

$$\# \text{Cadrage_val_analog} = \frac{\# \text{Val_analog_pér}}{27648} \cdot (\# \text{mx} - \# \text{mn}) + \# \text{mn}$$

Un signal analogique est nécessaire pour cette tâche. L'opérande utilisé à cet effet doit être entré dans la table des variables API.

Nom	Type de données	Adresse	Commentaire
B1	INT	%EW64	Niveau entre 0..27648

9.2 Réalisation

Vous pouvez à présent réaliser vous-même la démarche pratique !

9.3 Liste de contrôle – exercice

N°	Description	Contrôlé
1	Opérande inséré dans la table des variables API	
2	Fonction FC : "Cadrage" créé	
3	Interface définie	
4	Fonction programmée	
5	Fonction "Cadrage" ajoutée dans le réseau 1 de l'OB1	
6	Variables d'entrée connectées	
7	Variables de sortie connectées	
8	Compilation réussie et sans message d'erreur	
9	Chargement réussi et sans message d'erreur	
10	Valeur analogique pour le niveau mise à zéro Résultat cadrage_niveau = 0 Résultat "er" = FALSE	
11	Valeur analogique pour le niveau définie sur 27648 Résultat cadrage_niveau = 12,0 Résultat "er" = FALSE	
12	Valeur analogique pour le niveau définie sur 13824 Résultat cadrage_niveau = 6,0 Résultat "er" = FALSE	
13	Forçage de l'opérande (mx = 0,0) Résultat cadrage_niveau = 0 Résultat de la variable "er" = TRUE	
14	Projet archivé avec succès	

10. Informations complémentaires

Si vous voulez vous mettre à niveau ou approfondir le sujet, des informations complémentaires sont disponibles sous forme de guide, comme par exemple : Getting Started, vidéos, tutoriels, applis, manuels, guides de programmation et logiciels/firmware démo (Trial Software/Firmware), sous le lien suivant :

www.siemens.com/sce/s7-1500