



## Documentazione didattica SCE per una soluzione di automazione omogenea Totally Integrated Automation (TIA)

### **Modulo TIA Portal 010-090**

Avvio alla programmazione in linguaggio evoluto  
con S7-SCL e SIMATIC S7-1200

## Trainer Package SCE adatti a questa documentazione

- **SIMATIC S7-1200 AC/DC/RELAIS pacchetti da 6 postazioni PLC "TIA Portal"**  
Nr. di ordinazione: 6ES7214-1BE30-4AB3
- **SIMATIC S7-1200 DC/DC/DC pacchetti da 6 postazioni PLC "TIA Portal"**  
Nr. di ordinazione: 6ES7214-1AE30-4AB3
- **SIMATIC S7-SW for Training STEP 7 BASIC V11 Upgrade per 6 pacchetti STEP 7 Basic (per S7-1200) "TIA Portal"**  
Nr. di ordinazione: 6ES7822-0AA01-4YE0

Tenere presente che questi Trainer Package potrebbero essere sostituiti da successivi pacchetti.

Potete consultare i pacchetti SCE attualmente disponibili su: [siemens.com/sce/tp](https://www.siemens.com/sce/tp)

## Corsi di formazione

Per corsi di formazione regionali di Siemens SCE contattare il partner di contatto SCE regionale

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

## Ulteriori informazioni su SCE

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

## Avvertenza importante sulla traduzione

La presente documentazione didattica è stata tradotta sulla base dei documenti redatti in tedesco. Gli screenshot sono stati riprodotti dalla lingua inglese. Per agevolare la comprensione, anche all'interno del testo sono stati adottati i comandi di programma in inglese con traduzione nella lingua straniera tra parentesi.

## Avvertenze per l'impiego

La documentazione di formazione per una soluzione di automazione omogenea Totally Integrated Automation (T I A) è stata creata per il programma "Siemens Automation Cooperates with Education (SCE)" specialmente per scopi di formazione per enti di formazione, di ricerca e di sviluppo pubblici. La Siemens AG declina qualunque responsabilità riguardo ai contenuti di questa documentazione.

Questa documentazione può essere utilizzata solo per la formazione base di prodotti e sistemi Siemens. Ciò significa che può essere copiata in parte, o completamente, e distribuita agli studenti nell'ambito della loro formazione professionale. La riproduzione, distribuzione e divulgazione di questa documentazione è consentita solo all'interno di istituzioni di formazione pubbliche e a scopo di formazione professionale.

Qualsiasi eccezione richiede un'autorizzazione scritta dal partner di riferimento di Siemens AG: Sig. Roland Scheuerer [roland.scheuerer@siemens.com](mailto:roland.scheuerer@siemens.com).

Le trasgressioni obbligano al risarcimento dei danni. Tutti i diritti sono riservati, incluso anche quelli relativi alla distribuzione e in particolare quelli relativi ai brevetti e ai marchi GM.

L'utilizzo per corsi rivolti a clienti del settore industria è esplicitamente proibito e non è inoltre permesso l'utilizzo commerciale della documentazione.

Ringraziamo Michael Dziallas Engineering e tutte le persone coinvolte nella creazione di questa documentazione.

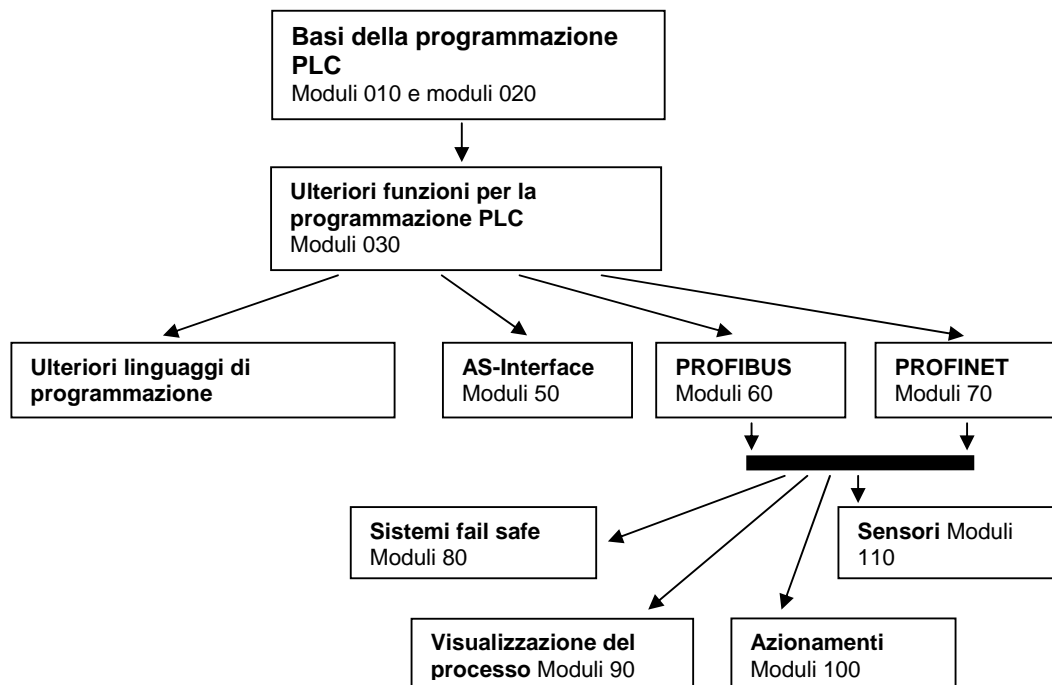


**Pagina:**

1.	Prefazione .....	5
2.	Avvertenza sul linguaggio di programmazione S7-SCL.....	7
3.	L'ambiente di sviluppo S7-SCL .....	8
4.	Esempio di programmazione: contenuto di un serbatoio.....	9
4.1	Descrizione del compito .....	9
4.2	Lista di attribuzione/tabella delle variabili.....	9
4.3	Struttura del programma .....	10
4.4	Interfaccia del blocco calculate_volume_tank [FC140].....	11
4.5	Avvertenza per la soluzione .....	12
5.	Programmazione del calcolo del contenuto di un serbatoio per SIMATIC S7-1200 in S7-SCL .....	13
5.1	Creazione di un progetto e configurazione hardware .....	13
5.2	Creazione del programma.....	16
5.3	Test del programma .....	25
5.4	Ampliamento del programma .....	28

## 1. Prefazione

Il modulo SCE\_DE\_010-090 appartiene per contenuti all'unità didattica '**Basi della programmazione PLC**' e rappresenta un **rapido approccio** alla programmazione di SIMATIC S7 1200 con il linguaggio di programmazione **S7-SCL** in TIA Portal.



### Obiettivo didattico:

L'obiettivo di questo modulo è far conoscere al lettore le funzioni di base dell'ambiente di sviluppo S7-SCL. Inoltre verranno presentate funzioni di test per eliminare gli errori logici dalla programmazione.

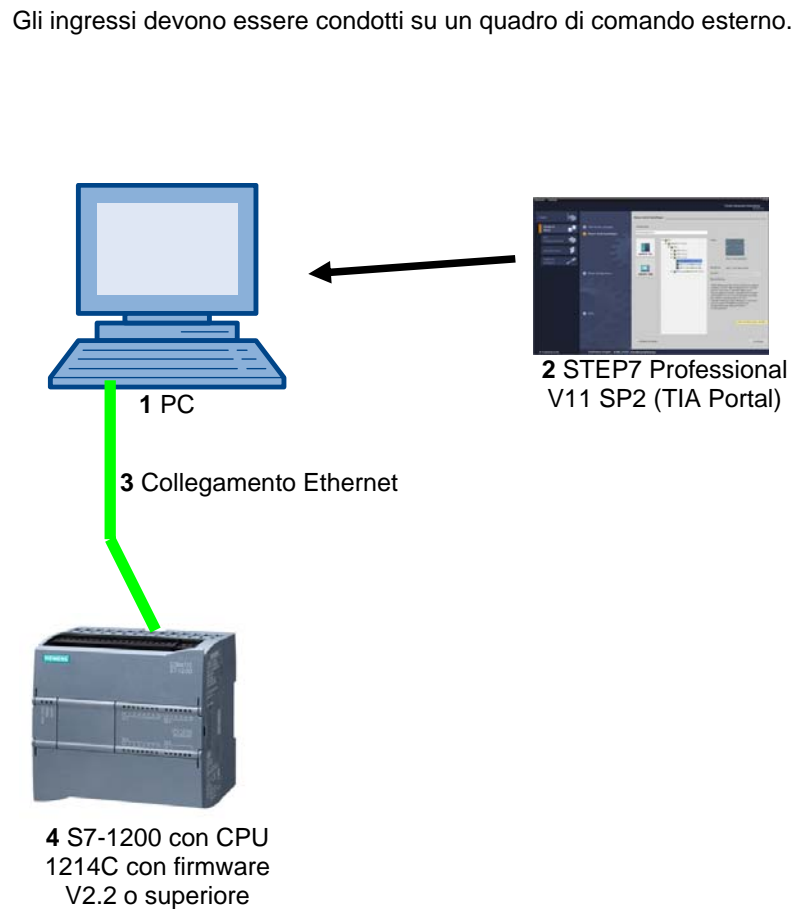
### Presupposti:

Per una corretta elaborazione di questo modulo si presuppongono le conoscenze seguenti:

- Esperienza nell'uso di Windows
- Basi di programmazione dei PLC con STEP 7 Professional V11 (ad es. moduli da 010-010 a 010-040 )
- Basi di programmazione con linguaggi evoluti come ad es. Pascal.

## Requisiti hardware e software

- 1 PC Pentium 4, 1.7 GHz 1 (XP) – 2 (Vista) GB RAM, memoria su disco rigido ca. 2 GB, sistema operativo Windows XP Professional SP3 / Windows 7 Professional / Windows 7 Enterprise / Windows 7 Ultimate / Windows 2003 Server R2 / Windows Server 2008 Premium SP1, Business SP1, Ultimate SP1
- 2 Software STEP7 Professional V11 SP2 (Totally Integrated Automation (TIA) Portal V11)
- 3 Collegamento Ethernet tra PC e CPU 1214C
- 4 PLC SIMATIC S7-1200 con firmware V2.2 o superiore, ad es. CPU 1214C con unità di ingresso/uscita 14DI/10DO.



## 2. Avvertenza sul linguaggio di programmazione S7-SCL

S7-SCL (Structured Control Language) è un linguaggio di programmazione evoluto che si orienta al PASCAL e consente una programmazione strutturata. Il linguaggio è conforme al linguaggio sequenziale SFC "Sequential Function Chart" definito nella norma DIN EN-61131-3 (IEC 61131-3). Oltre agli elementi di un linguaggio evoluto, S7-SCL comprende anche elementi tipici dei PLC come ingressi, uscite, temporizzatori, merker, richiami di blocchi ecc. come elementi di linguaggio. Supporta il concetto dei blocchi di STEP 7 e pertanto, con AWL, KOP e FUP, consente la programmazione di blocchi a norma. S7-SCL integra e amplia il software di programmazione STEP 7 con i suoi linguaggi di programmazione KOP, FUP e AWL.

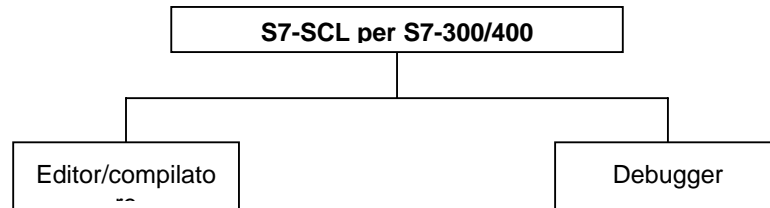
Non è necessario creare individualmente ogni funzione ma è possibile ricorrere a blocchi preconfigurati come funzioni di sistema o blocchi funzionali di sistema disponibili nel sistema operativo dell'unità centrale.

I blocchi programmati con S7-SCL si possono combinare con blocchi di AWL, KOP e FUP. Ciò significa che un blocco programmato con S7-SCL può richiamare un altro blocco programmato in AWL, KOP o FUP. Allo stesso modo è possibile richiamare blocchi S7-SCL anche nei programmi AWL, KOP e FUP

Le funzioni di test di S7-SCL consentono la ricerca di errori logici di programmazione in una compilazione senza errori.

### 3. L'ambiente di sviluppo S7-SCL

Per l'applicazione di S7-SCL è disponibile un ambiente di sviluppo adeguato sia a proprietà specifiche di S7-SCL sia a STEP 7. Questo ambiente di sviluppo è costituito da un editor/compilatore e un debugger.



#### Editor/compilatore

L'editor S7-SCL è un editor di testo che consente di elaborare qualsiasi testo. Il suo compito principale è creare e modificare i blocchi per i programmi STEP 7. Durante l'immissione del testo viene eseguita un'accurata verifica della sintassi, che semplifica una programmazione senza errori. Gli errori di sintassi vengono rappresentati con diversi colori.

L'editor offre le seguenti possibilità:

- Programmazione di un blocco S7 nel linguaggio S7-SCL.
- Comodo inserimento di elementi del linguaggio e di richiami dei blocchi con drag&drop.
- Verifica diretta della sintassi durante la programmazione.
- Impostazione dell'editor in base alle proprie esigenze, ad es. con i diversi elementi del linguaggio evidenziati con colori in base alla sintassi.
- Verifica del blocco finito con l'aiuto della compilazione.
- Visualizzazione di tutti gli errori e gli avvisi che si verificano durante la compilazione.
- Localizzazione del punto del blocco con l'errore, con descrizione dell'errore e indicazioni sul rimedio in via opzionale.

#### Debugger

Il debugger S7-SCL offre la possibilità di controllare il ciclo di un programma nell'AS e di individuare così eventuali errori logici.

A questo scopo S7-SCL offre due diversi modi di test:

- Controllo passo passo
- Controllo continuo

Il "controllo passo passo" segue la sequenza logica del programma. È possibile eseguire l'algoritmo del programma istruzione per istruzione e controllare in una finestra dei risultati come cambiano i contenuti delle variabili elaborate

Il "controllo continuo" consente di testare un gruppo di istruzioni all'interno di un blocco. Durante il test i valori delle variabili e i parametri vengono visualizzati in ordine cronologico e - se possibile - aggiornati ciclicamente.



## 4. Esempio di programmazione: contenuto di un serbatoio

### 4.1 Descrizione del compito

Il nostro primo programma consiste nella programmazione del calcolo del contenuto di un serbatoio.

Il serbatoio ha la forma di un cilindro verticale. Il livello del contenuto viene misurato con un sensore analogico. Nel nostro compito il valore del livello deve essere già normalizzato e disponibile nell'unità 'metri'.

Il programma deve essere programmato in una funzione FC140 'calculate\_volume\_tank'. I parametri di trasferimento sono il diametro e il livello di riempimento in metri. Il risultato è il contenuto del serbatoio in litri.

### 4.2 Lista di attribuzione/tabella delle variabili

Poiché nella moderna programmazione non si utilizzano gli indirizzi assoluti ma delle variabili, qui è necessario innanzitutto definire le **variabili PLC globali**.

Queste variabili PLC globali sono nomi descrittivi corredati di commento per gli ingressi e le uscite che vengono utilizzati nel programma. In un secondo momento, durante la programmazione, da questi nomi si potrà accedere alle variabili PLC globali.

Queste variabili globali si possono utilizzare in tutto il programma in tutti i blocchi.

#### Tabella delle variabili standard

Nome	Tipo di dati	Indirizzo	Commento
filling_level_tank 1	REAL	%MD40	filling level of tank1 (meter)
diameter_tank1	REAL	%MD44	diameter tank1 (meter)
volume_tank1	REAL	%MD48	volume tank1 (liter)

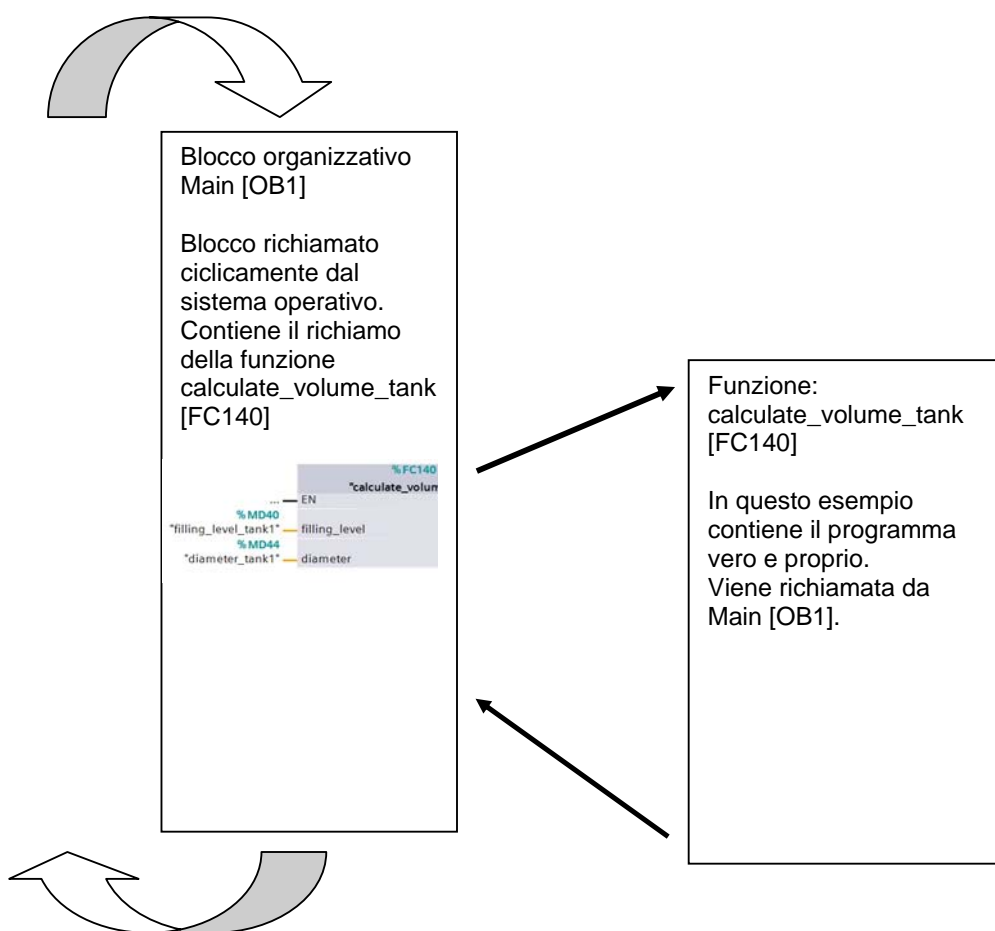
### 4.3 Struttura del programma

La sequenza del programma viene scritta in cosiddetti blocchi. Per default è già disponibile il blocco organizzativo Main [OB1]. Questo OB rappresenta l'interfaccia con il sistema operativo della CPU, dal quale viene richiamato automaticamente ed elaborato ciclicamente.

D'altra parte da questo blocco organizzativo è possibile richiamare ulteriori blocchi, come ad es. la funzione calculate\_volume\_tank [FC140], per strutturare la programmazione.

Lo scopo è quello di suddividere un compito complessivo in compiti più piccoli che sono più facili da risolvere e da testare nella loro funzionalità.

#### Struttura del programma di esempio "contenuto del serbatoio"



#### 4.4 Interfaccia del blocco calculate\_volume\_tank [FC140]

Prima di poter scrivere il programma è necessario dichiarare l'interfaccia del blocco. Con la dichiarazione dell'interfaccia si definiscono le variabili locali, conosciute solo in questo blocco.

Le variabili o parametri dell'interfaccia si suddividono in due gruppi:

- Parametri del blocco che costituiscono l'interfaccia del blocco per il richiamo nel programma.

Tipo	Denominazione	Funzione	Disponibile in
Parametri di ingresso	Input	Parametri i cui valori vengono letti dal blocco.	Funzioni, blocchi funzionali e alcuni tipi di blocchi organizzativi
Parametri di uscita	Output / Return	Parametri i cui valori vengono scritti dal blocco.	Funzioni e blocchi funzionali
Parametri di transito	InOut	Parametri il cui valore viene letto dal blocco al momento del richiamo e riscritto nello stesso parametro dopo l'elaborazione.	Funzioni e blocchi funzionali

- Dati locali che permettono il salvataggio di risultati intermedi.

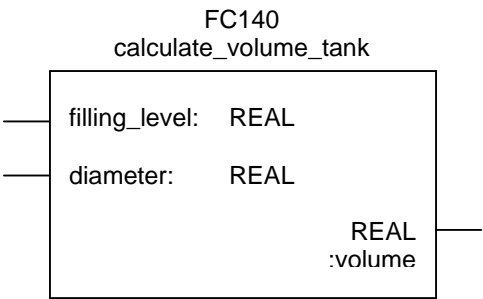
Tipo	Denominazione	Funzione	Disponibile in
Dati locali temporanei	Temp	Variabili che permettono il salvataggio di risultati intermedi temporanei. I dati temporanei vengono mantenuti solo per un ciclo.	Funzioni, blocchi funzionali e blocchi organizzativi
Dati locali statici	Static	Variabili che permettono il salvataggio di risultati intermedi statici nel blocco dati di istanza. I dati statici vengono mantenuti anche per più cicli finché non vengono riscritti.	Blocchi funzionali

I parametri di interfaccia utilizzati nel nostro programma di esempio per il blocco 'calculate\_volume\_tank [FC140]' sono i seguenti.

Interfaccia della funzione FC140:		calculate_volume_tank	
Tipo	Nome	Tipo di dati	Commento
IN	filling_level	REAL	filling level tank (meter)
IN	diameter	REAL	diameter tank (meter)
OUT	volume	REAL	volume of liquid in the tank (liter) / -1 if error

Il richiamo della funzione in un blocco programmato con FUP sarà il seguente:

Richiamo della funzione:      **rappresentazione in FUP**



4.5      **Avvertenza per la soluzione**

Per la soluzione del compito si applica la formula di calcolo del volume di un cilindro verticale. Per calcolare il risultato in litri viene utilizzato il fattore di conversione 1000.

$$V = \frac{d^2}{4} \cdot \pi \cdot h$$

=>

$$\text{volume} = \frac{\text{diameter}^2}{4} \cdot 3.14159 \cdot \text{filling\_level} \cdot 1000$$

## 5. Programmazione del calcolo del contenuto di un serbatoio per SIMATIC S7-1200 in S7-SCL

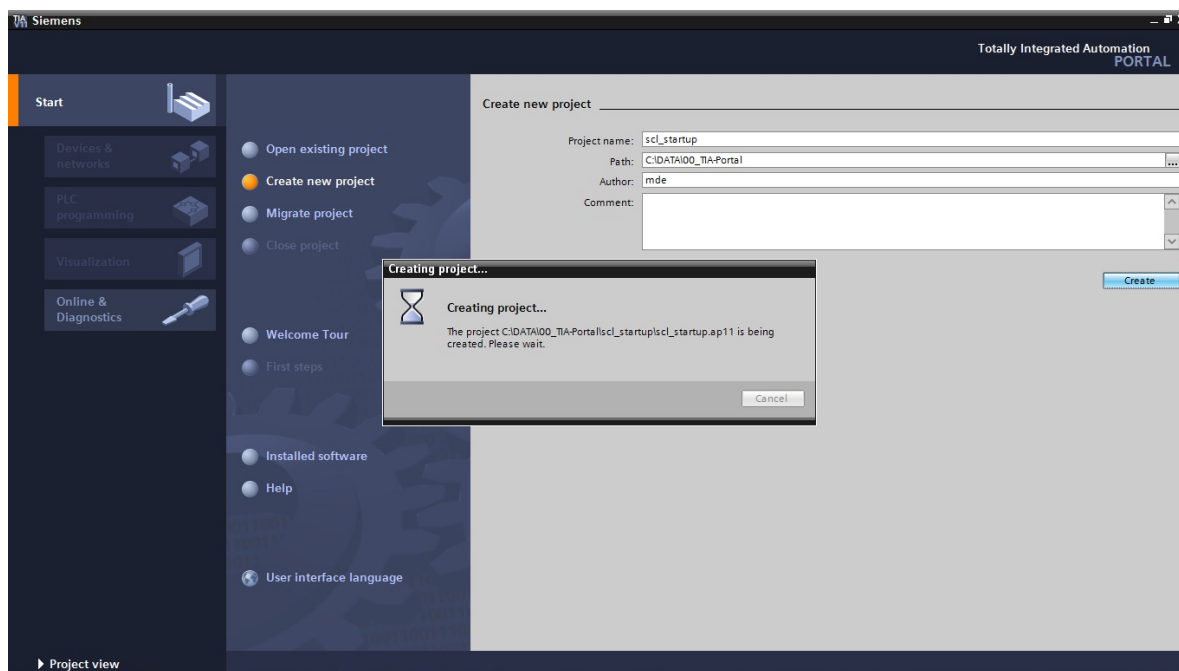
Con i passi seguenti è possibile creare un progetto per il SIMATIC S7-1200 e programmare la soluzione del compito che ci siamo proposti:

### 5.1 Creazione di un progetto e configurazione hardware

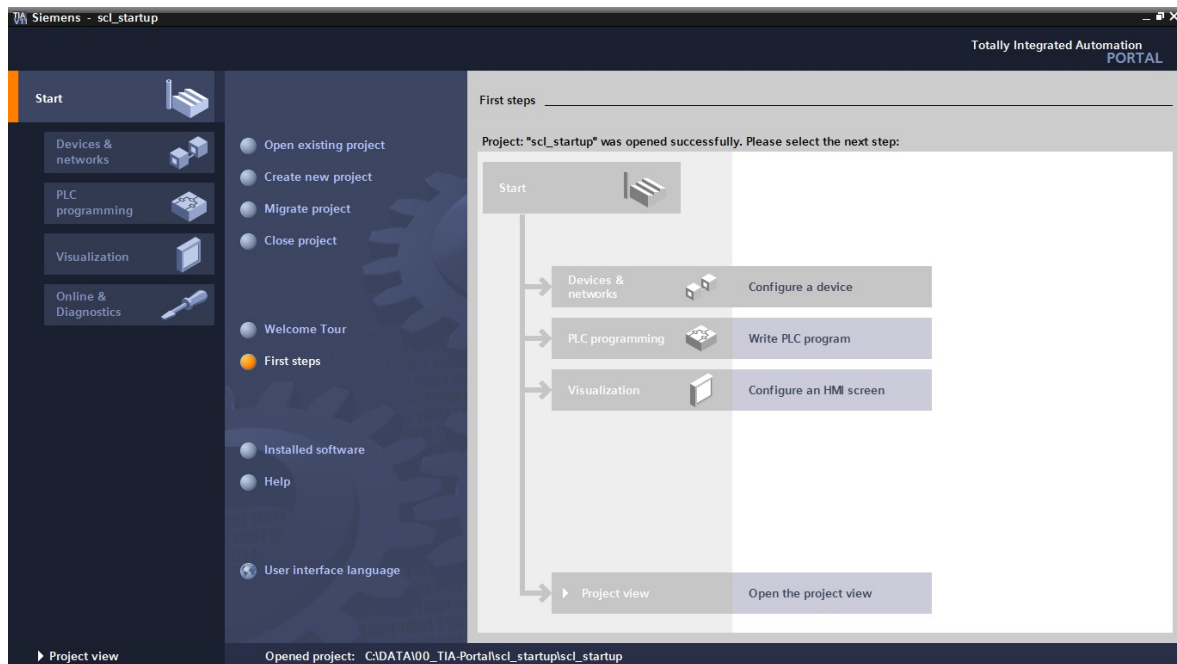
1. Il tool principale è '**Totally Integrated Automation Portal**', che si richiama qui con un doppio clic. ( → TIA Portal V11)



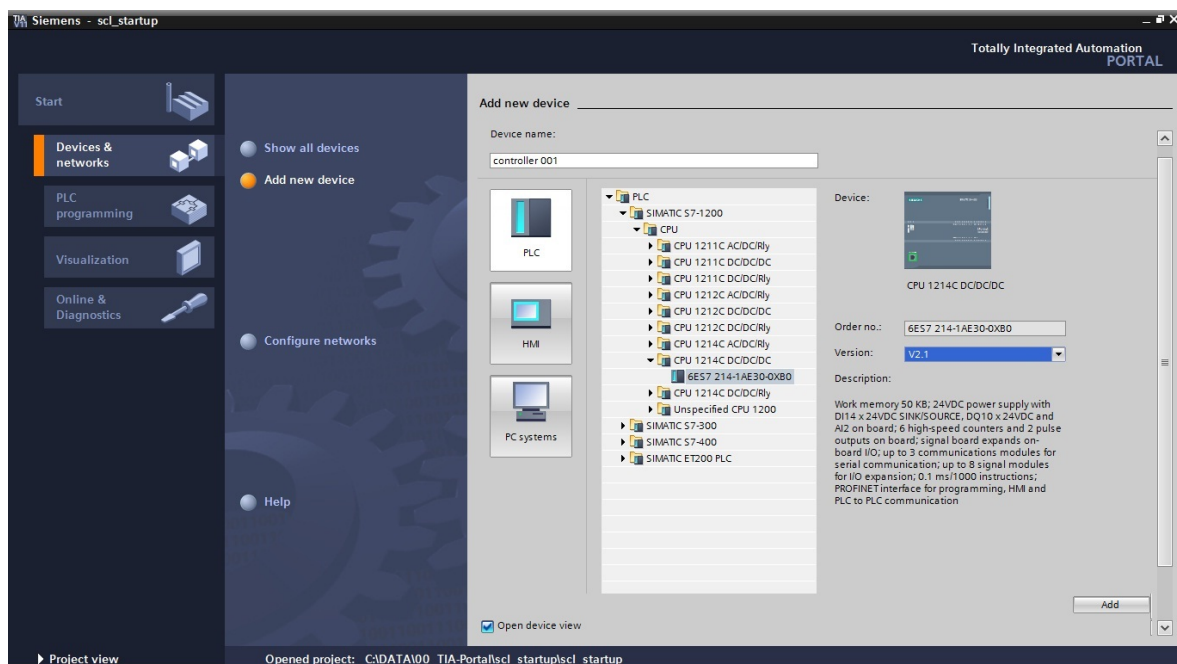
2. I programmi per il SIMATIC S7-1200 vengono gestiti in progetti. Uno di questi progetti verrà ora creato nella vista portale ( → Create new project (Crea nuovo progetto) → scl\_startup → Create (Crea))



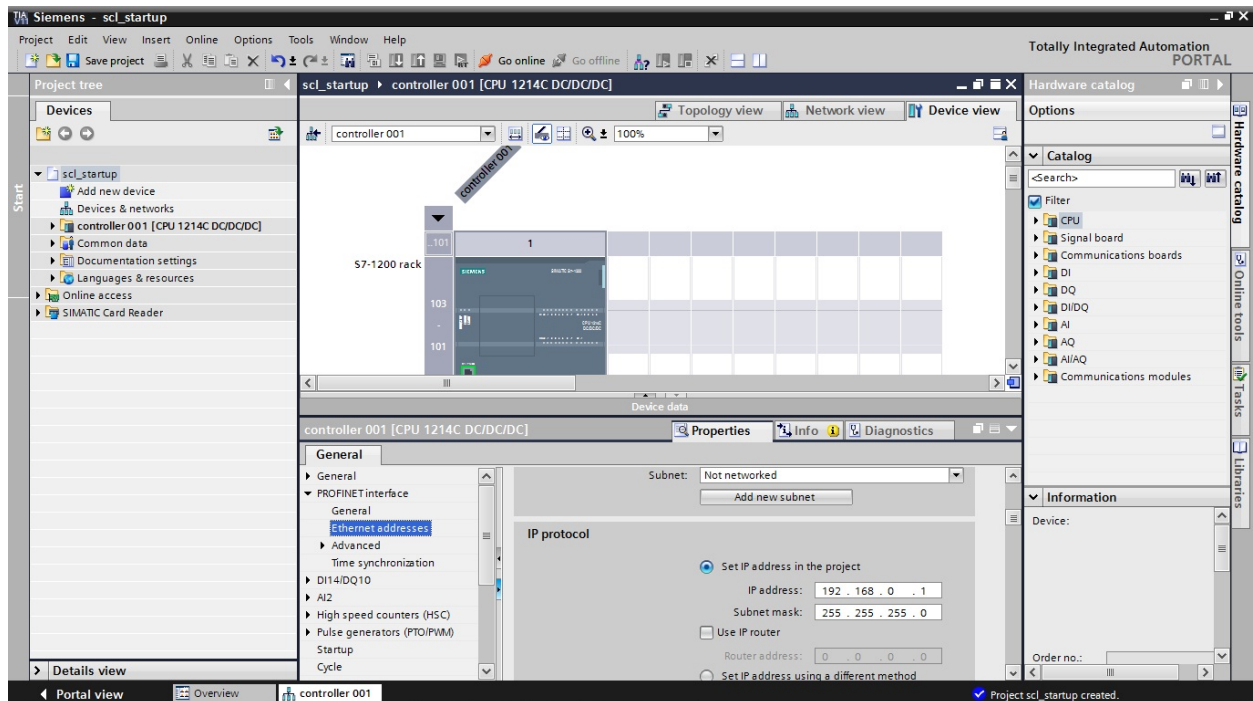
- Ora alla voce **'First steps'** vengono proposti i primi passi per la progettazione. Innanzitutto vogliamo creare un dispositivo con **'Configure a device'**. ( → Primi passi → Configura un dispositivo)



- Successivamente inseriremo con **'Add new device'** un nuovo dispositivo denominato **'controller 001'**. Selezionare dal catalogo la **'CPU1214C AC/DC/Rly'** con il numero di ordinazione adeguato. ( → Aggiungi nuovo dispositivo → Controllore 001 → PLC → SIMATIC S7-1200 → CPU → CPU1214 AC/DC/Rly → 6ES7 214-1BE30-0XB0 → V2.2 → Aggiungi)

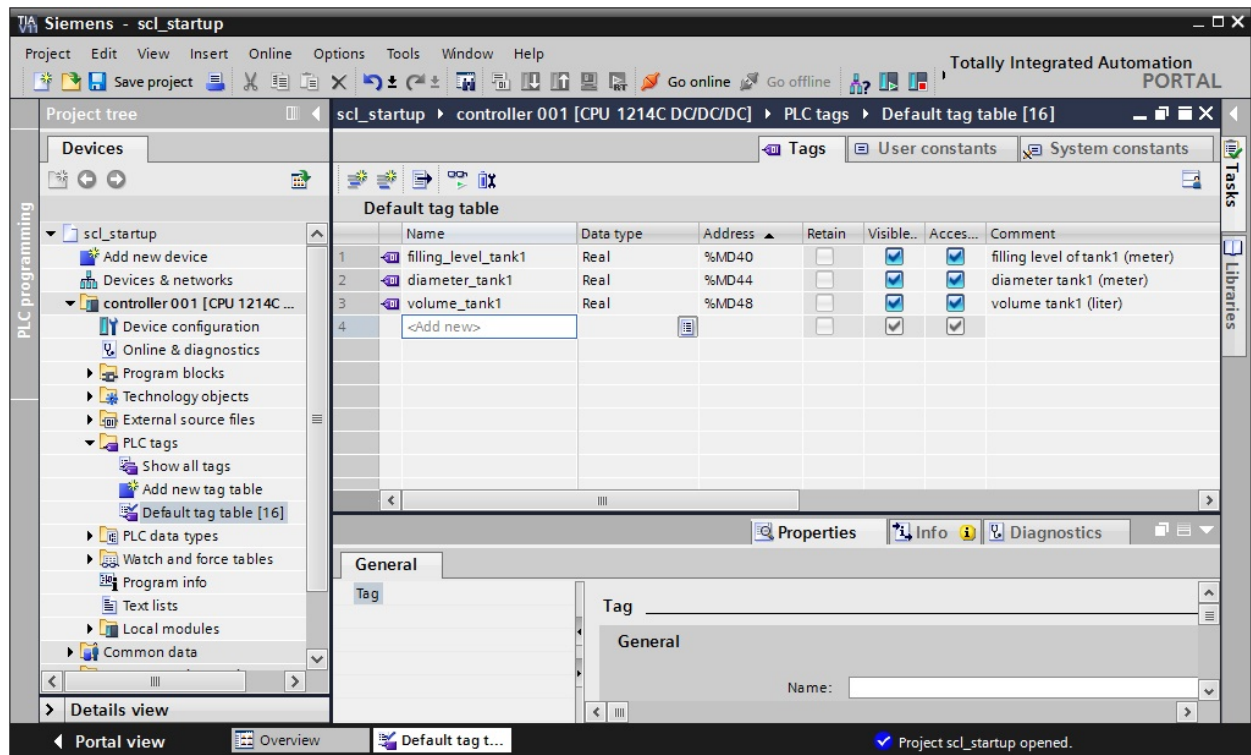


5. Ora il software passa automaticamente alla vista progetto con la configurazione hardware aperta nella vista dispositivi. Per far sì che in un secondo tempo il software acceda alla CPU corretta è necessario impostare l'indirizzo ETHERNET in '**ETHERNET addresses**' .  
 (→ Properties (Proprietà) → General (Generale) → ETHERNET addresses (Indirizzo ETHERNET)  
 → IP address (Indirizzo IP): 192.168.0.1 → Subnet mask (Maschera di sottorete): 255.255.255.0)



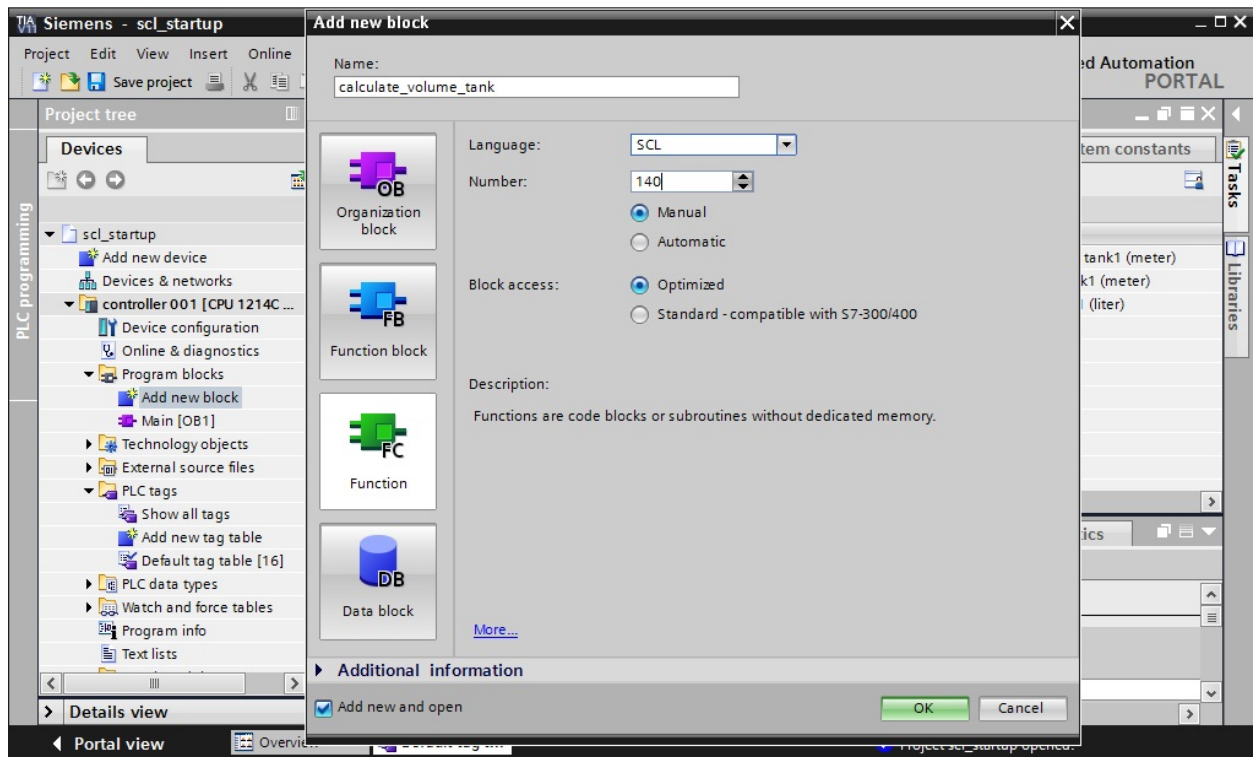
## 5.2 Creazione del programma

1. Aprire **'Default tag table'** (Tabella delle variabili standard) e inserire gli **operandi** con nome e tipo di dati.  
( → Controllore 001[CPU1214 AC/DC/Rly] → Variabili PLC → Tabella delle variabili standard → inserire gli operandi)

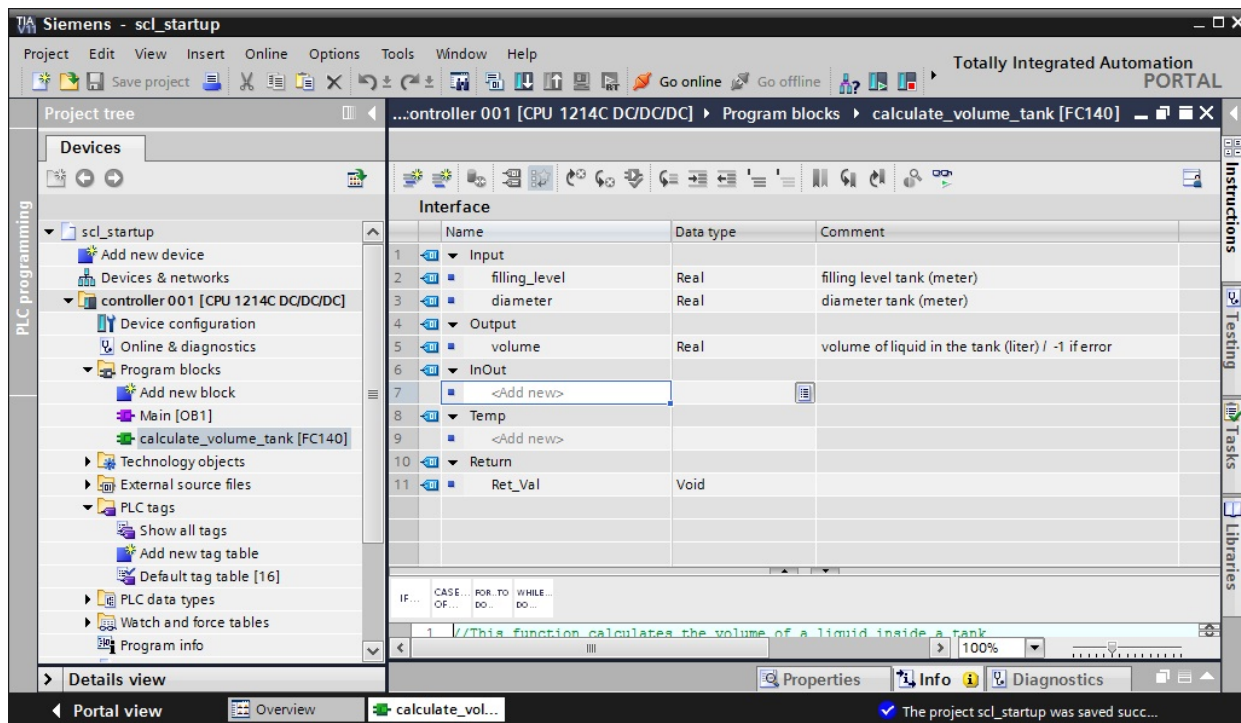




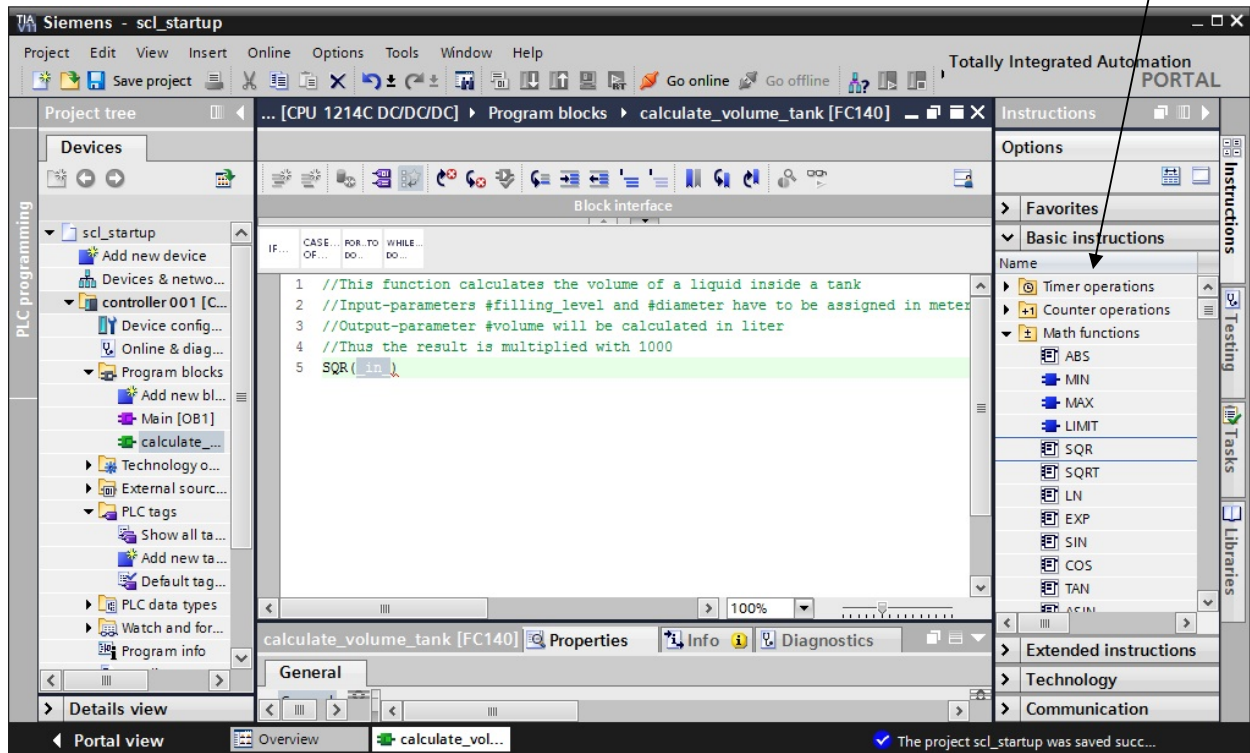
2. Per creare la funzione `calculate_volume_tank` selezionare nella navigazione del progetto '**controller 001[CPU1214 AC/DC/Rly]**' e quindi '**Program blocks**'. Fare doppio clic su '**Add new block**'. Selezionare '**Function (FC)**' e assegnare il nome '**calculate\_volume\_tank**'. Modificare il linguaggio di programmazione in '**SCL**'. La numerazione si può modificare passando dal modo automatico al manuale. Immettere il numero 140. Applicare i dati inseriti con '**OK**'.  
( → Controllore 001[CPU1214 AC/DC/Rly] → Blocchi di programma → Inserisci nuovo blocco → Funzione (FC) → `calculate_volume_tank` → SCL → Manuale → 140 → OK)



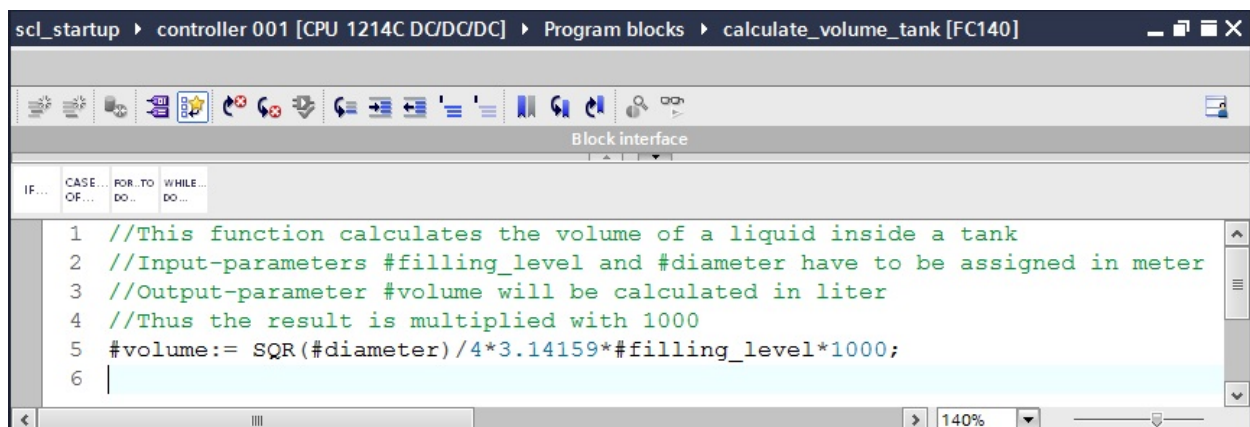
3. Il blocco 'calculate\_volume\_tank [FC140]' si aprirà automaticamente. Immettere i **parametri di ingresso e di uscita** del blocco come indicato in figura. Per una migliore comprensione, tutte le variabili locali dovrebbero essere corredate di un commento **sufficientemente esplicativo**.  
( → Ingrandire l'interfaccia del blocco FC140 → inserire i parametri di interfaccia )

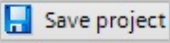

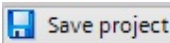



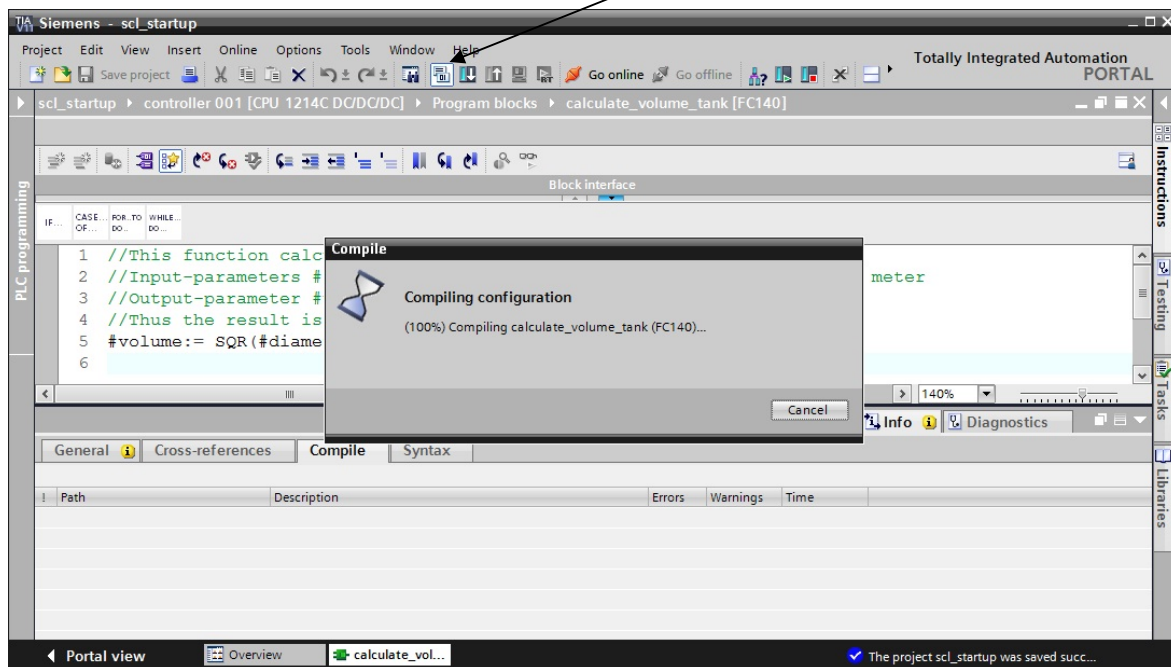
4. Dopo aver dichiarato le variabili locali è possibile iniziare la creazione del programma qui riportato. Viene utilizzata la funzione quadratica '**SQR**' integrata in SCL per generare il quadrato di un numero. Il numero da elevare al quadrato è riportato tra parentesi. Per inserire la funzione trascinarla con il mouse sul punto di applicazione nel programma.  
(→ Instructions (Istruzioni, menu a destra!) → Basic instructions (Istruzioni di base) → Math functions (Funzioni matematiche) → SQR)



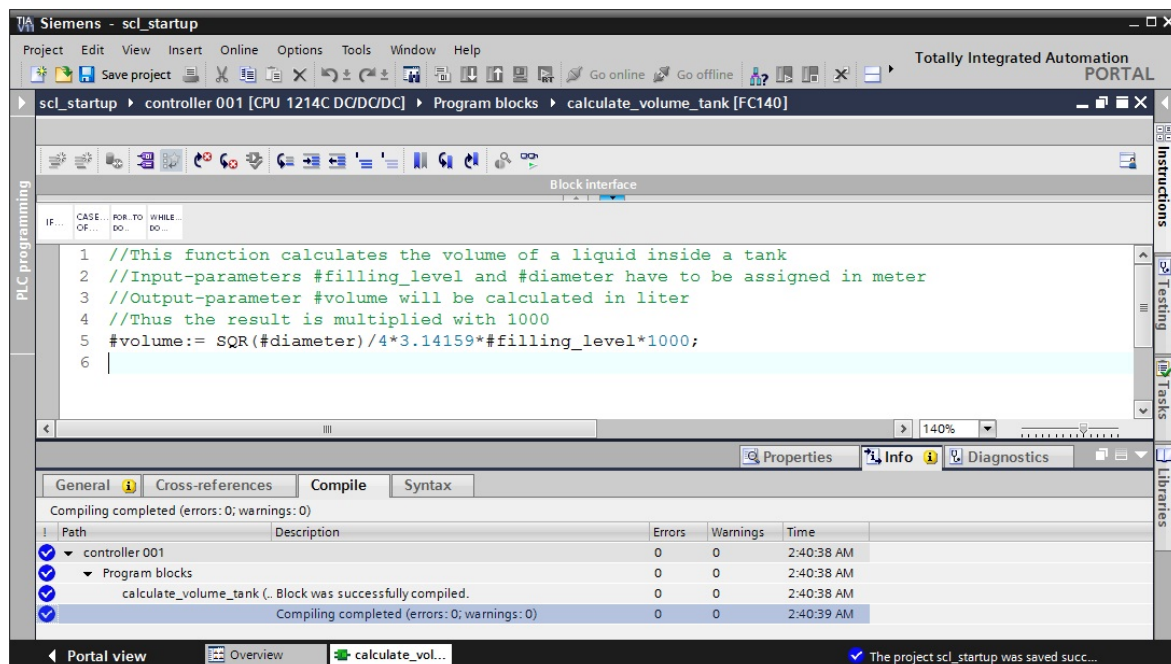
5. Completare il programma come indicato qui di seguito.



6. Il programma può essere salvato  e compilato .
- (→ Salva  → Compila )

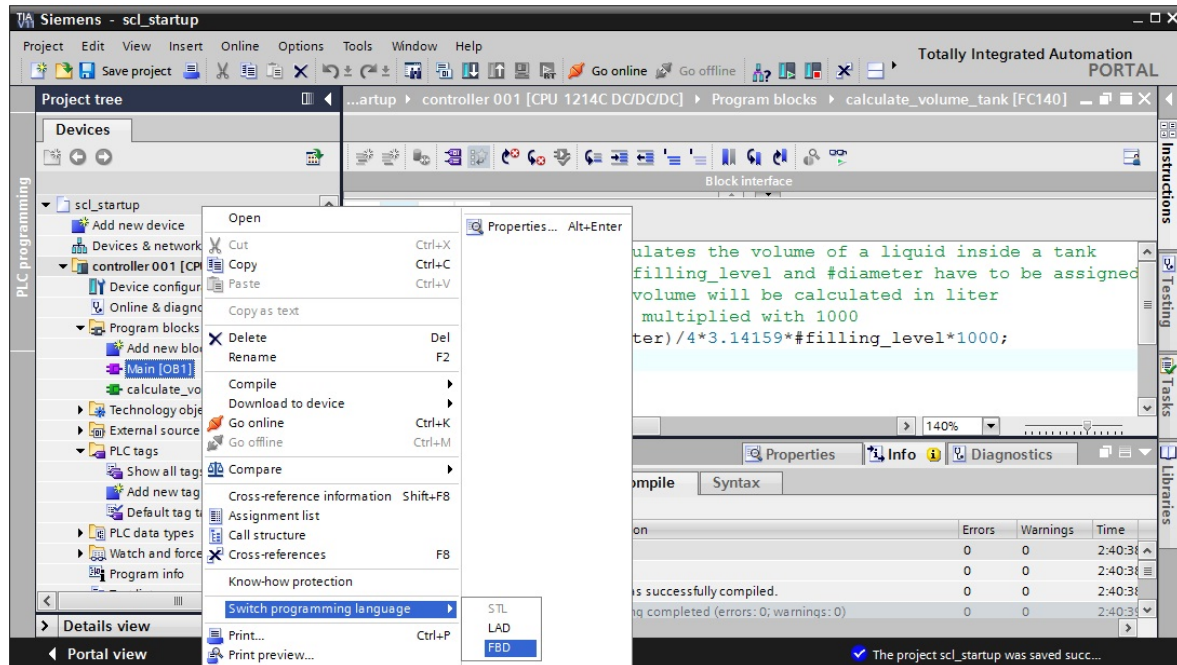


7. La compilazione riconosce gli errori di sintassi, che vengono visualizzati nel menu 'Info/Compile'. (→ Informazioni → Compila)

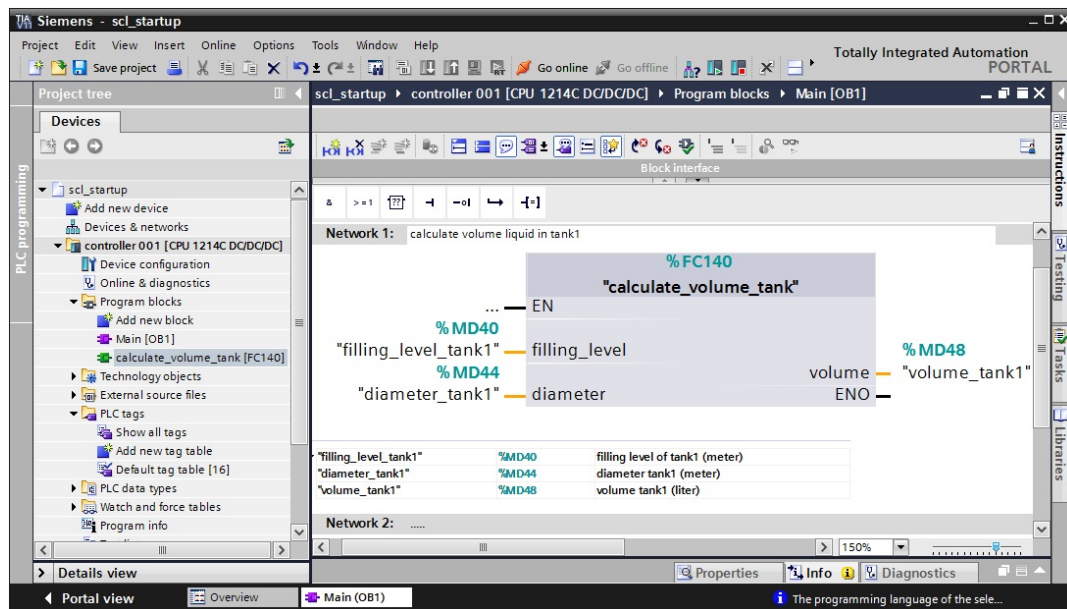


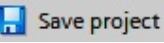


8. Ora è possibile richiamare la funzione nel **'Main[OB1]'**. Prima di richiamare il blocco **'Main[OB1]'** con un doppio clic reimpostare il linguaggio di programmazione a **'FBD'**.  
(→ Main[OB1] → Switch programming language (Commuta linguaggio di programmazione) → FBD (FUP))

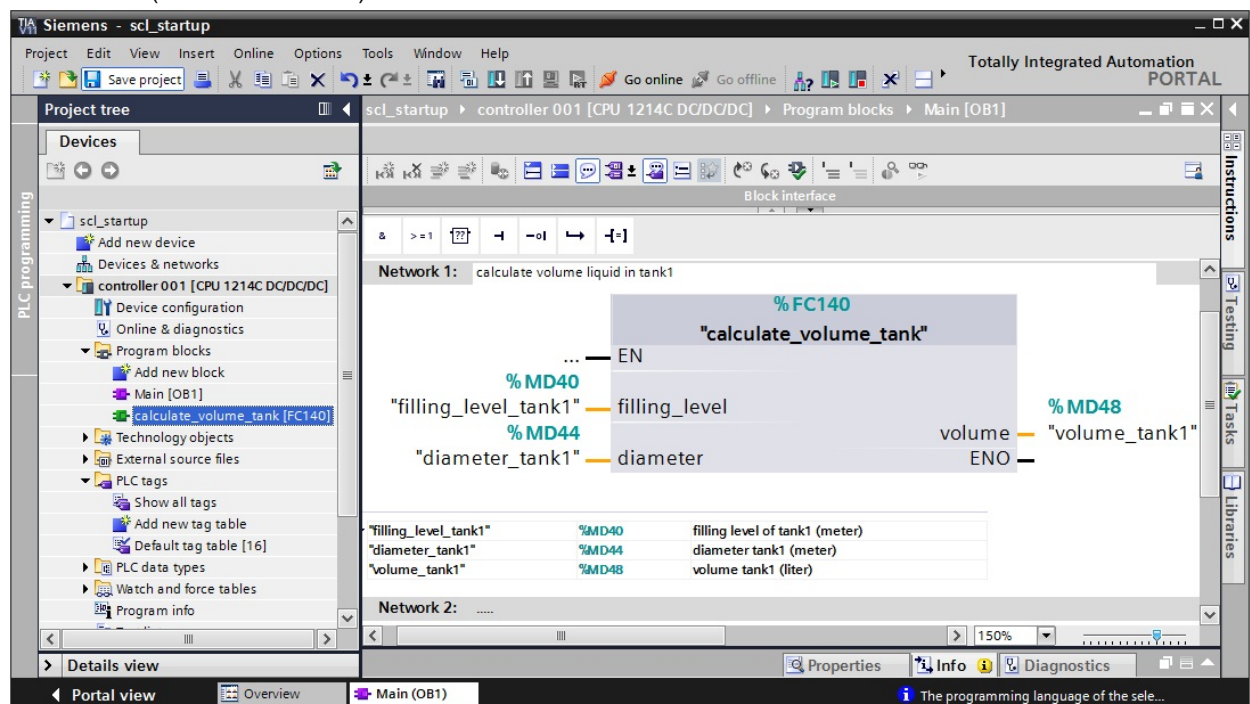




9. La funzione **'calculate\_volume\_tank[FC140]'** può essere semplicemente trascinata con il mouse nel segmento 1 del blocco Main[OB1]. I parametri dell'interfaccia della funzione **'calculate\_volume\_tank [FC140]'** devono essere collegati con le variabili PLC globali - come mostrato qui. Non dimenticare di documentare i segmenti anche nel blocco Main[OB1]. (→ Main[OB1] → Program blocks (Blocchi di programma) → calculate\_volume\_tank[FC140])

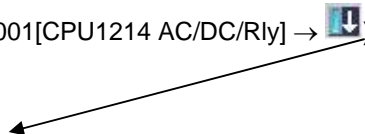


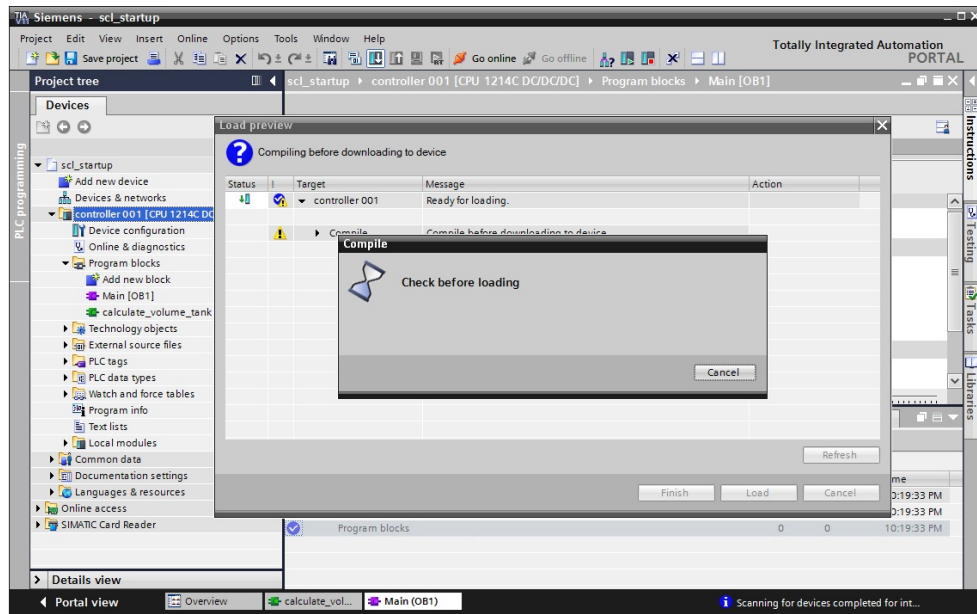
10. Salvare ancora un volta il progetto con il pulsante  (Salva progetto).

(→  Save project)



11. Per caricare i blocchi di programma e la configurazione del dispositivo nella CPU selezionare prima la cartella '**controller 001[CPU1214 AC/DC/Rly]**' quindi fare clic sul simbolo  (Carica nel dispositivo). (→ Controllore 001[CPU1214 AC/DC/Rly] → )

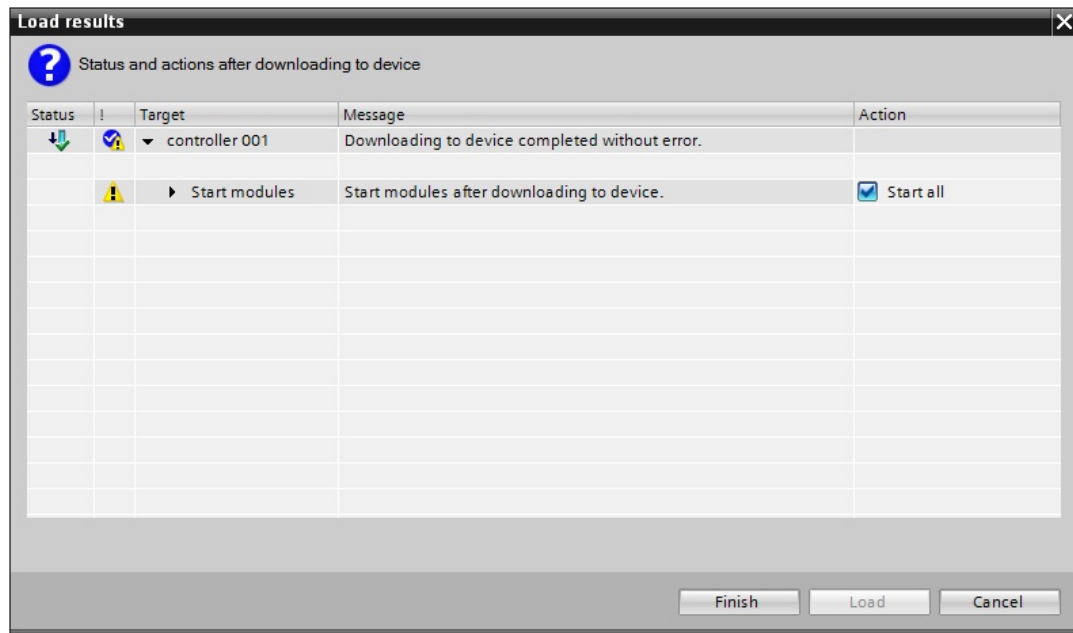
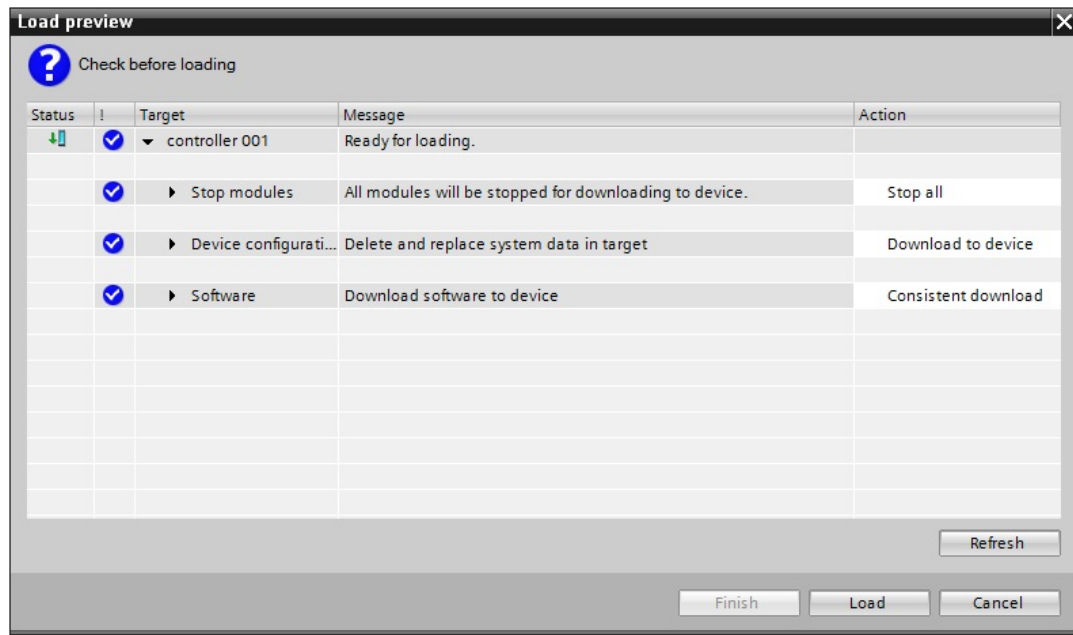




**Avvertenza:**



Con il caricamento il progetto viene automaticamente ricompilato per individuare eventuali errori.

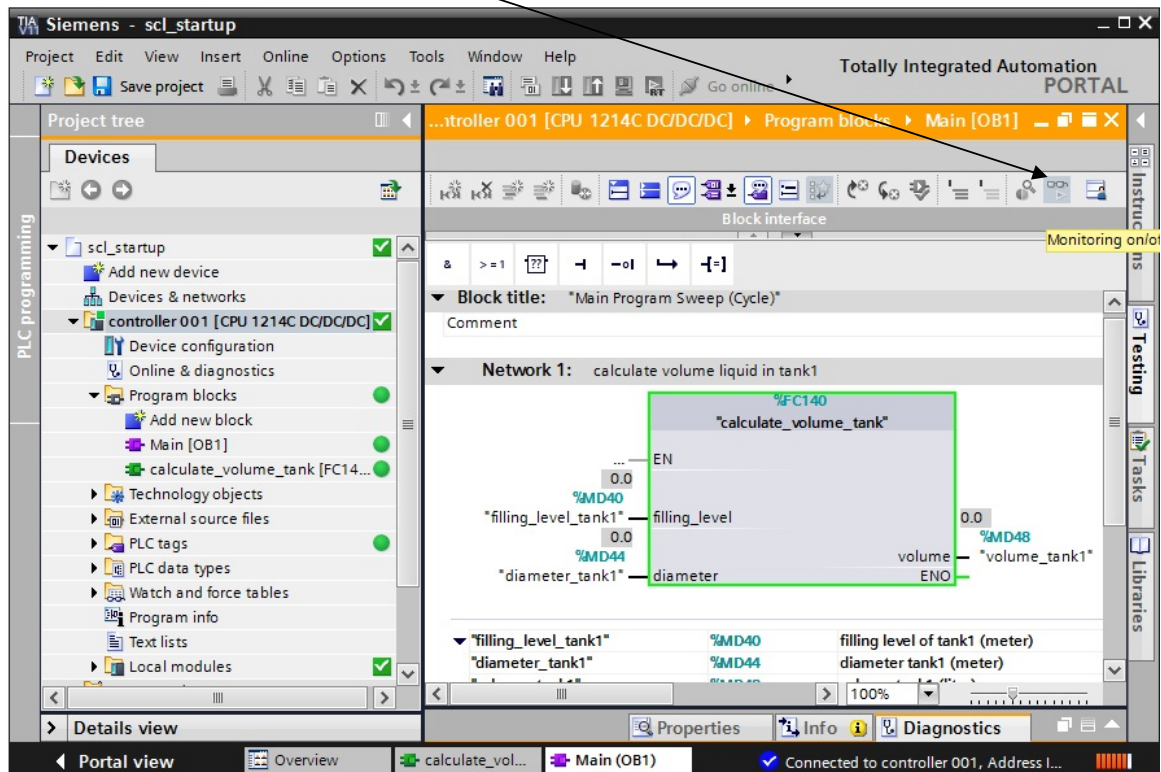
12. Prima del caricamento viene visualizzata ancora una volta una panoramica per controllare tutti i passi da eseguire. Selezionare **'Load'** e avviare la CPU con **'Start all'**, **'Finish'**. ( → Carica → Avvia tutto → Fine)






### 5.3 Test del programma

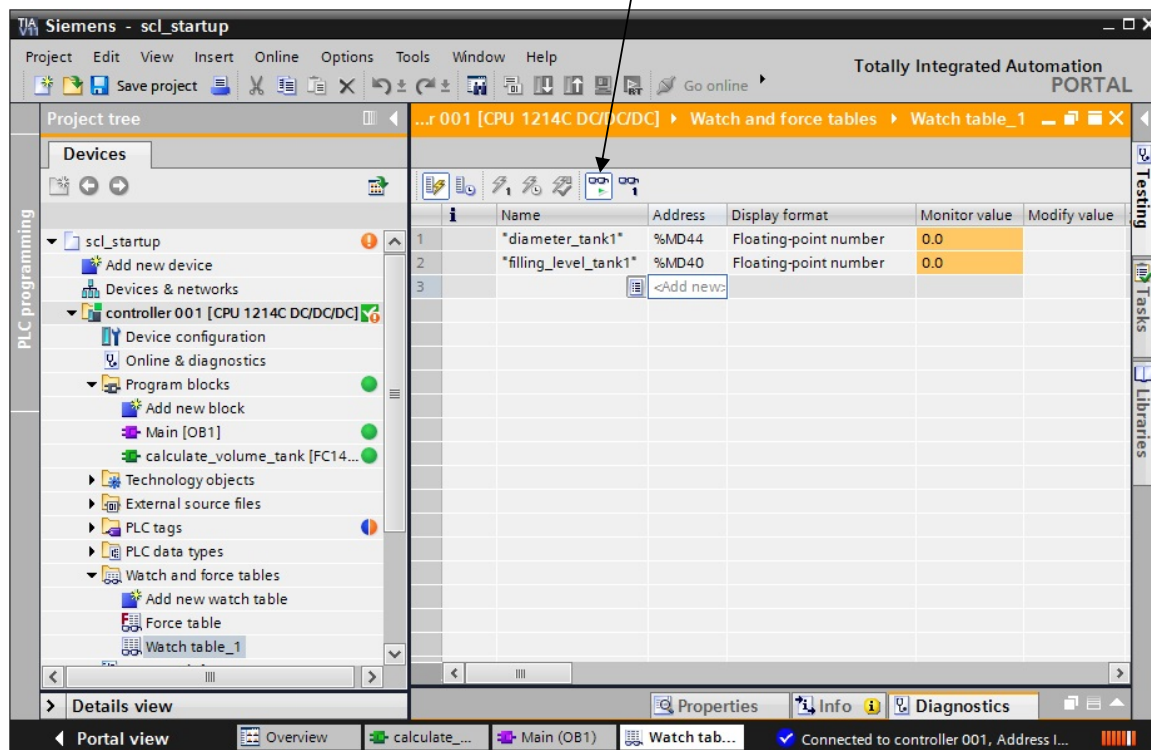
1. Con un clic del mouse sul simbolo  Monitoring on/off (Controllo on/off) è possibile controllare lo stato delle variabili di ingresso e di uscita nel blocco 'calculate\_volume\_tank' durante il test del programma. ( →  )





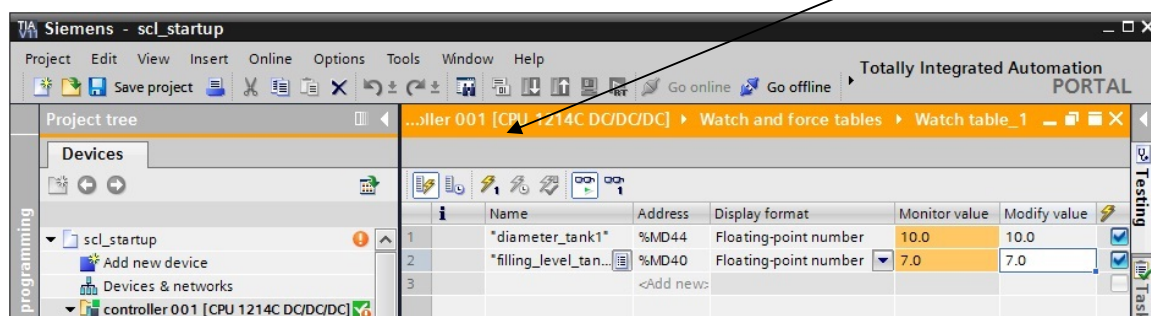
2. Poiché non disponiamo di un sensore analogico e quindi di un corrispondente valore di processo dobbiamo predefinire i valori 'diameter\_tank1' e 'filling\_level\_tank1' con l'aiuto di una tabella di controllo. Creare una nuova tabella di controllo con 'Add new watch table' e immettere entrambi i valori. Attivare la modalità 'Monitoring on' per vedere i valori attuali.


(→ Controller 001 → Tabella di controllo e di forzamento → Aggiungi nuova tabella di controllo →

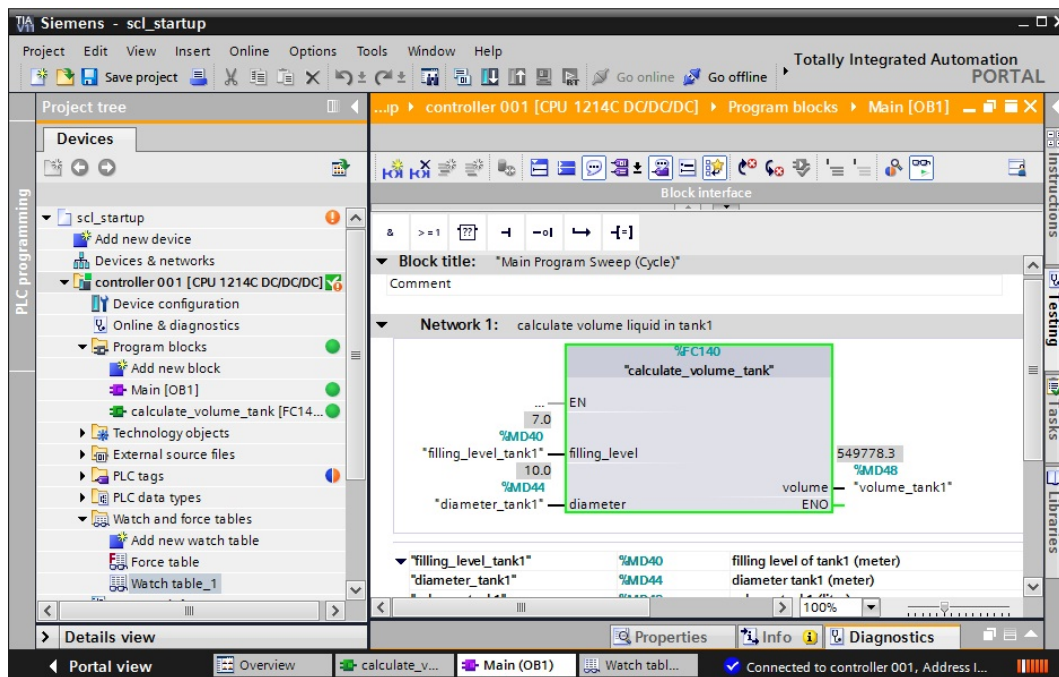
diameter\_tank1, filling\_level\_tank1 → )




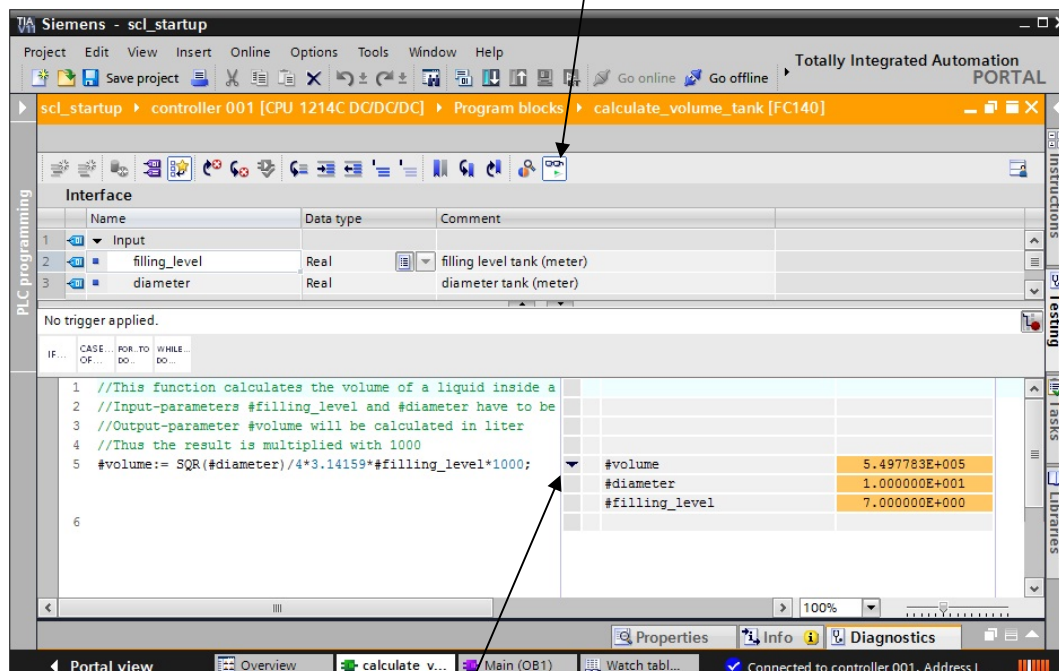
3. Per predefinire i valori è necessario immettere un valore di comando nella colonna 'Modify value' (Valore di comando). Con il pulsante 'Modify all selected values once and now'  (Comanda subito una sola volta tutti i valori attivati) i valori vengono applicati nella CPU. (→ diameter\_tank1 = 10.0 → filling\_level\_tank1 = 7.0 → )




4. Ora è possibile verificare il programma nell'OB1. (→ )



5. Nell'editor SCL è possibile controllare i valori delle singole variabili. Attivare la modalità 'Monitoring on' (Controllo on). (→ )



**Avvertenza:**

Facendo clic sul pulsante  si visualizzano i valori attuali della variabile programmata in questa riga.

5.4    **Ampliamento del programma**

Il blocco **'calculate\_volume\_tank'** deve verificare che i parametri di ingresso non contengano indicazioni errate. Inoltre, al blocco viene assegnato un ulteriore valore **'height\_max'** che indica l'altezza del serbatoio.  
Il blocco deve valutare se il livello di riempimento è minore di zero o maggiore dell'altezza specificata. Inoltre si deve verificare se il diametro specificato è minore di zero.  
In presenza di un errore un parametro di uscita booleano **'er'** deve restituire TRUE e il valore del parametro **'volume'** deve essere -1.

**Ampliamento della lista di attribuzione/tabella delle variabili:**

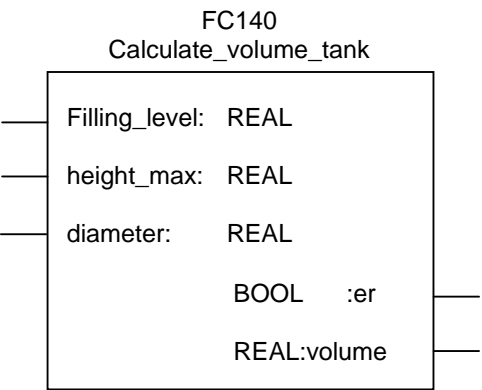
Indirizzo	Simbolo	Tipo di dati	Commento
%Q1.7	error_bit	BOOL	error, calculation not possible

**Ampliamento dell'interfaccia della funzione FC140:**

**calculate\_volume\_tank**

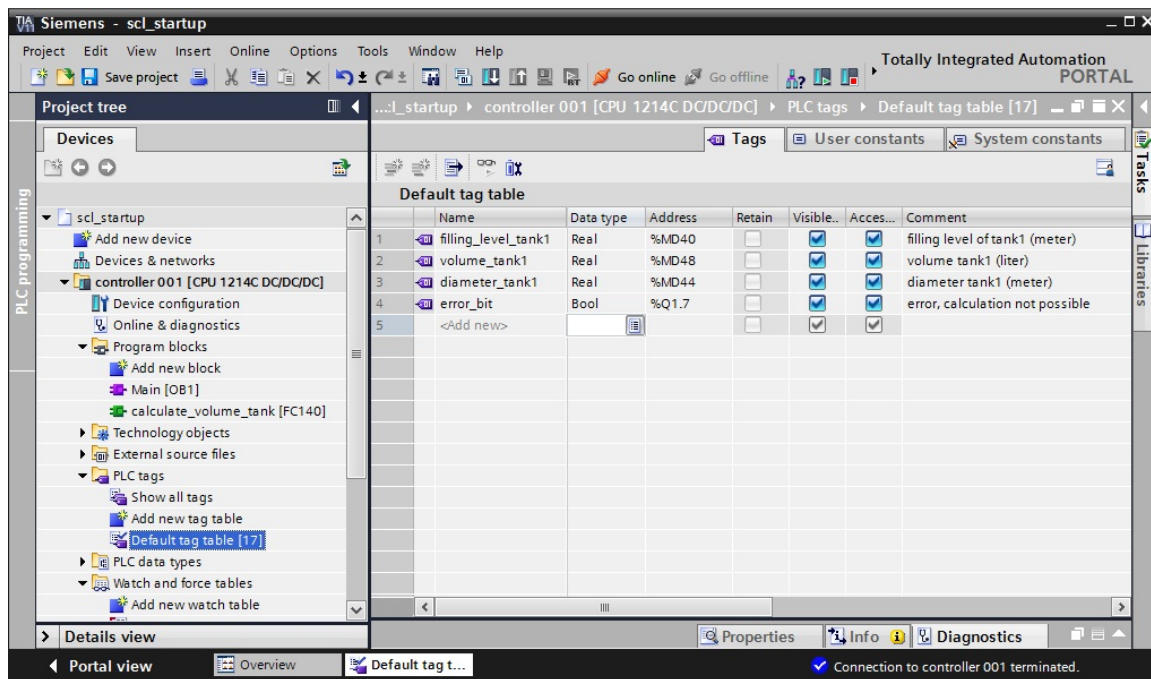
Tipo	Simbolo	Tipo di dati	Commento
IN	height_max	REAL	maximum height (meter)
OUT	er	BOOL	er = true -> error

**Richiamo della funzione:      rappresentazione in FUP**

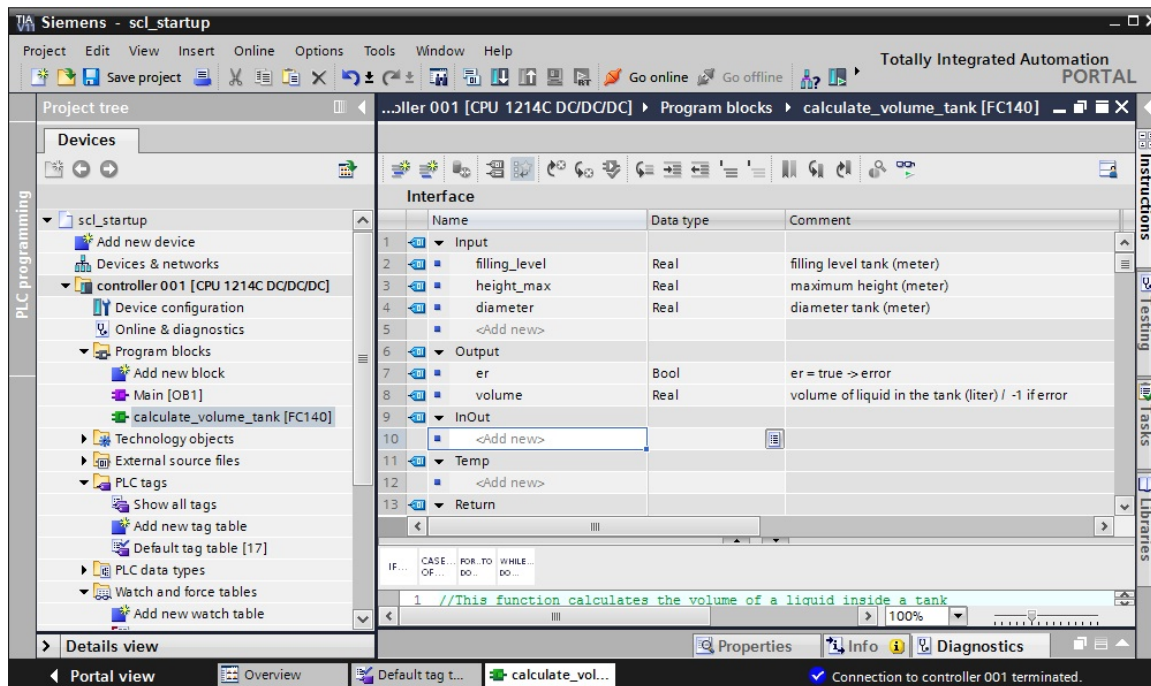







1. Ampliare la **tabella delle variabili** del blocco come sopra indicato.  
( → Controllore 001[CPU1214 AC/DC/Rly] → Variabili PLC → Tabella delle variabili standard → inserire gli operandi)

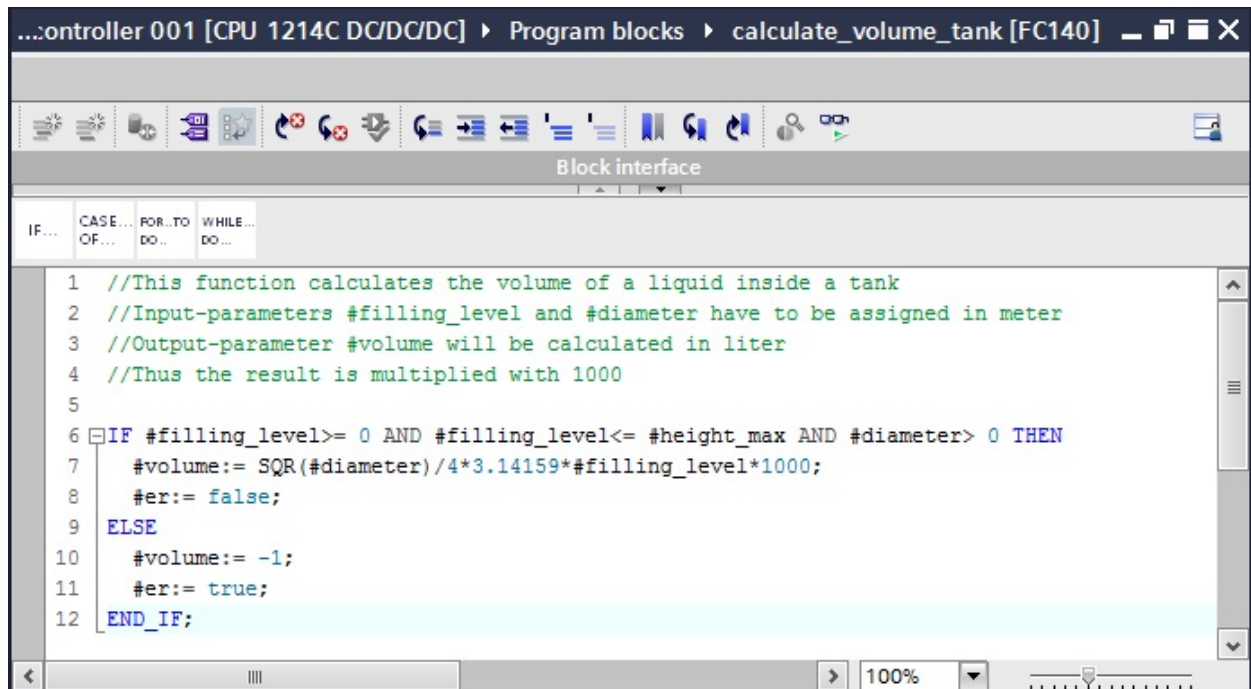


2. Ampliare i **parametri dell'interfaccia** del blocco come sopra indicato.  
( → Controllore 001[CPU1214 AC/DC/Rly] → Blocchi di programma → calculate\_volume\_tank) → inserire i parametri)

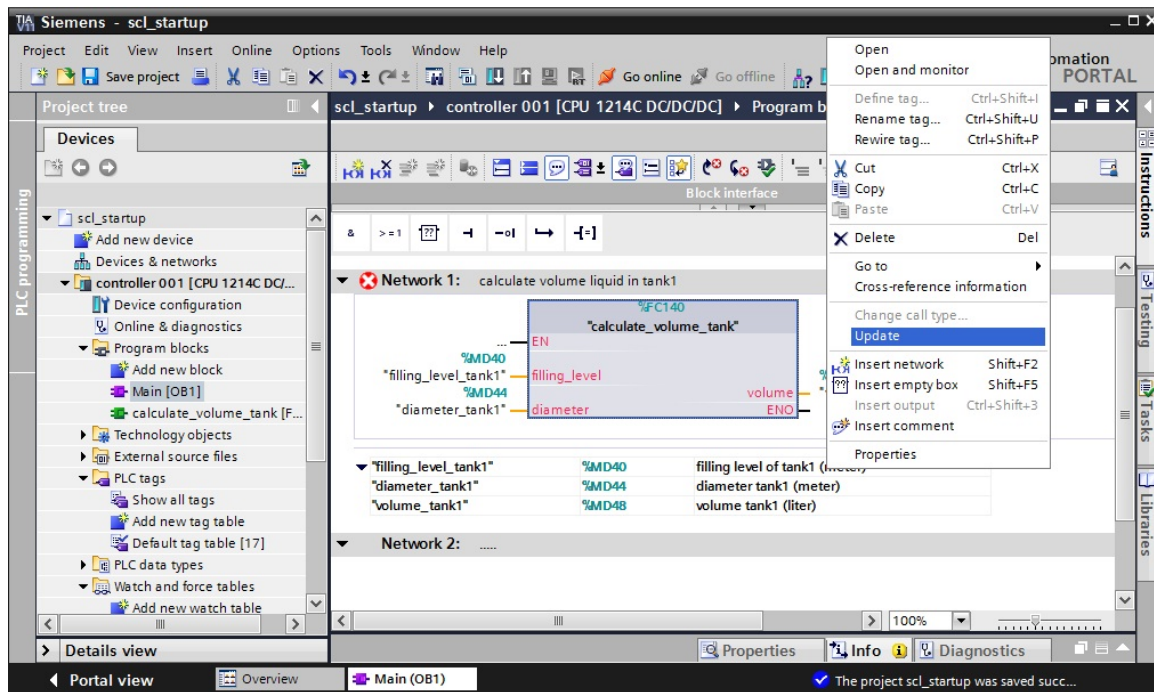


3. Completare il programma come indicato qui di seguito e compilarlo per cercare gli errori di sintassi. Salvare il programma e caricarlo nel controllore.

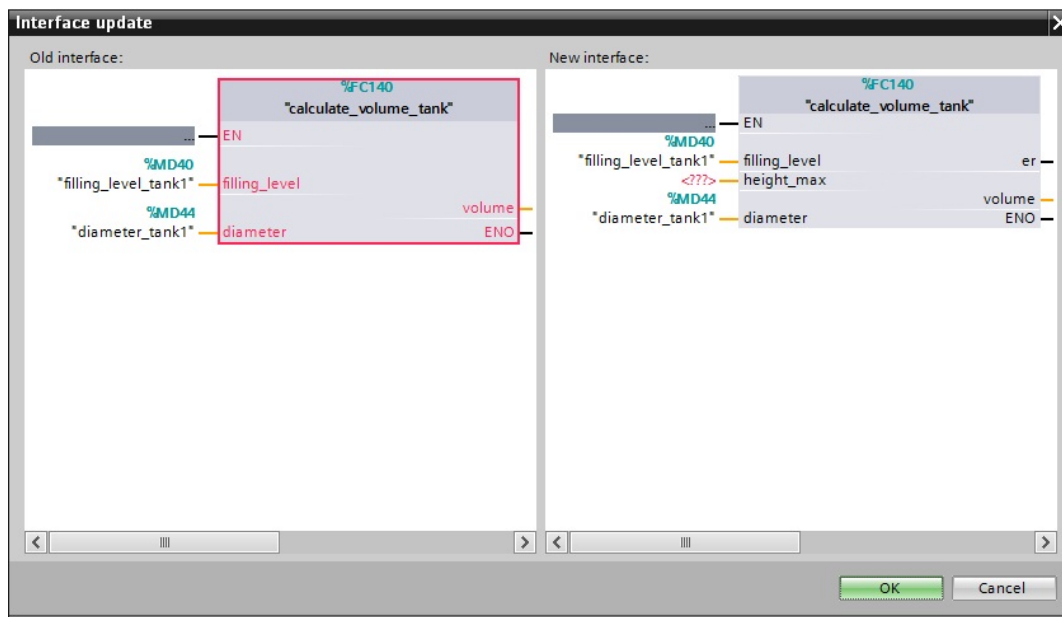
( Scrivi programma →  →  Save project →  )







4. Poiché i dati del blocco sono stati modificati è necessario aggiornare il richiamo nell'OB1. Aprire l'OB1 e scorrere fino al punto del richiamo del blocco. Aprire con il tasto destro del mouse il menu di scelta rapida e selezionare **'Update'**.  
( → Controllore 001[CPU1214 AC/DC/Rly] → Blocchi di programma → Main [OB1] → tasto destro del mouse → Aggiorna)

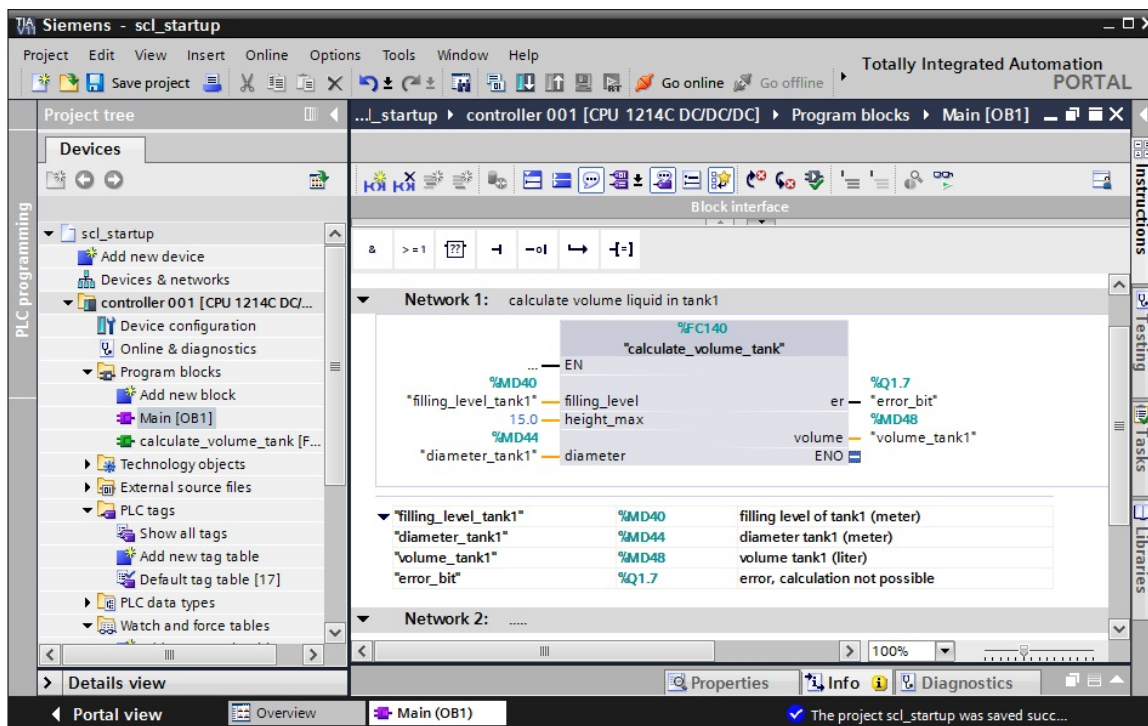


5. Vengono visualizzate sia l'interfaccia precedente che la nuova. Confermare con **'OK'**.  
( → OK)




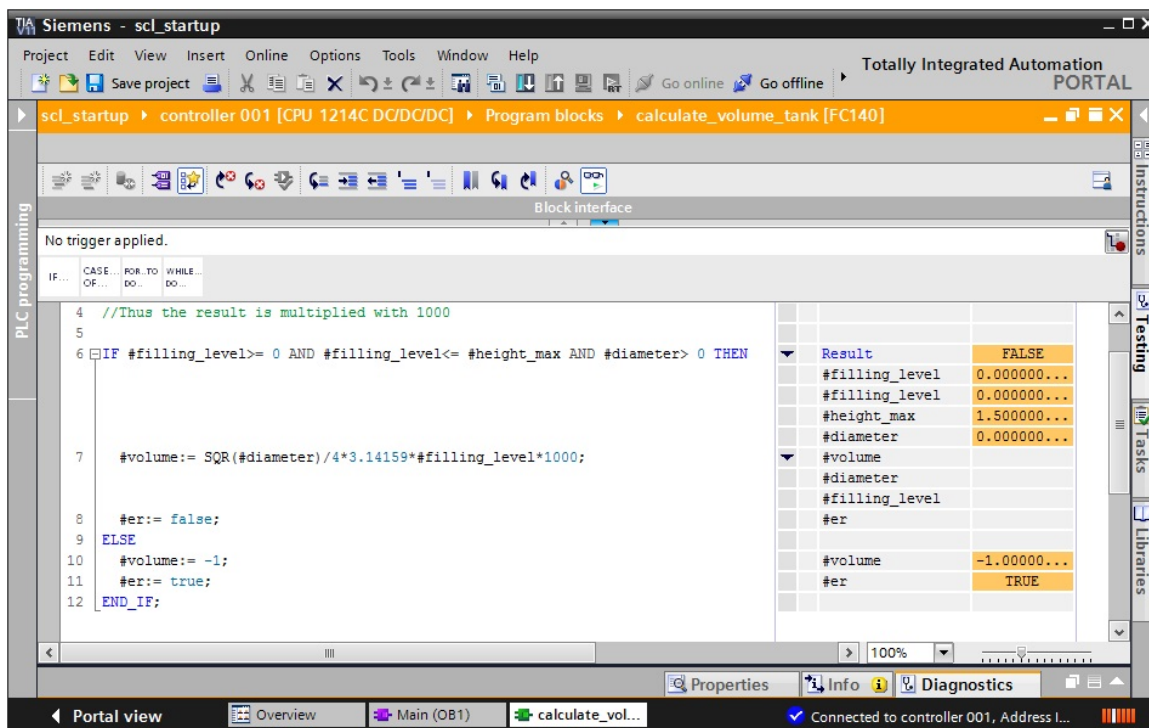
6. Completare i valori nel parametro di ingresso **'height\_max'** e nel parametro di uscita **'er'** come indicato qui di seguito. Compilare e salvare il programma e caricarlo nel controllore.

( Completare i parametri →  →  Save project →  →  )





7. Verificare le modifiche del blocco 'calculate\_volume\_tank' in modalità di controllo.  
 ( → Controllore 001[CPU1214 AC/DC/Rly] → Blocchi di programma → calculate\_volume\_tank →  )



**Avvertenza:** Il formato di visualizzazione dei valori attuali si può modificare dal menu di scelta rapida che si apre facendo clic sul valore con il tasto destro del mouse.