

SCE Training Curriculum pro Integrované Automatizační Řešení Totally Integrated Automation (TIA)

Simatic do škol

TIA Portal Module 010-090

Startup – Programování ve vyšších jazycích S7-SCL a
SIMATIC S7-1200

Cooperates
with Education

Automation

SIEMENS

SCE tréninkové balíčky pro tyto manuály

- **SIMATIC S7-1200 AC/DC/RELAY 6er "TIA Portal"**
Order No.: 6ES7214-1BE30-4AB3
- **SIMATIC S7-1200 DC/DC/DC 6er "TIA Portal"**
Order No.: 6ES7214-1AE30-4AB3
- **SIMATIC S7-SW for Training STEP 7 BASIC V11 Upgrade (for S7-1200) 6er "TIA Portal"**
Order No.: 6ES7822-0AA01-4YE0

Přehled nabízených balíčků najdete na adrese: siemens.com/sce/tp

Další možnosti

Pro regionální Siemens SCE trénink kontaktujte osobu z vašeho regionu:
siemens.com/sce/contact

Další informace o SCE

siemens.com/sce

Informace o používání

Tento SCE training curriculum pro integrované automatizační řešení Totally Integrated Automation (TIA) byl připraven pro program "Siemens Automation Cooperates with Education (SCE)" speciálně pro výcvikové účely pro veřejnost a R&D. Siemens AG negarantuje obsah.

Tento dokument je pro úvodní trénink pro Siemens produkty/systémy; i.e., může být kopírován celý nebo po částech a předán těm kteří se zrovna zacvičují. Předávání a kopírování dokumentu stejně tak jako sdílení jeho obsahu je povoleno pro výcvikové účely.

Výjimky je třeba si vyžádat písemně od kontaktní osoby ze Siemens AG: Roland Scheuerer roland.scheuerer@siemens.com.

Porušení pravidel bude hnáno k zodpovědnosti. Všechna práva včetně překladu jsou rezervována, především při udílení patentů nebo registraci designu.

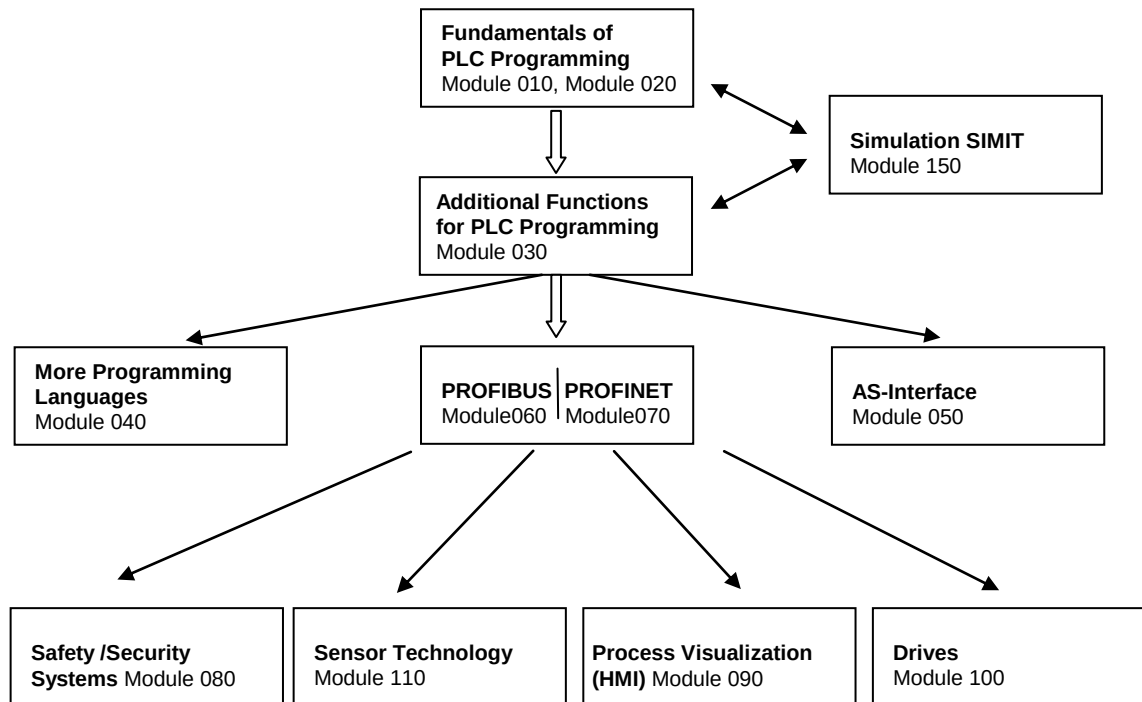
Použití pro kurzy průmyslových zákazníků je přímo zakázáno. Nechceme, aby byl dokument využit komerčně.

Děkujeme mnohokrát Michael Dziallas Engineering Corporation a všem ostatním zainteresovaným osobám za přípravu tohoto dokumentu.

	PAGE
1. Předmluva	4
2. Představení jazyku S7 SCL	6
3. S7-SCL vývojové prostředí	7
4. Vzorová úloha objem v nádrži	8
4.1 Popis úlohy	8
4.2 List úloh/ tabulka tagů	8
4.3 Struktura programu	9
4.4 Rozhraní bloku calculate_volume_tank [FC140]	10
4.5 Poznámka k řešení	11
5. Programování výpočtu objemu nádrže pro SIMATIC S7-1200 v S7-SCL	12
5.1 Nastavení projektu a konfigurace hardware	12
5.2 Nastavení programu	15
5.3 Testování programu	23
5.4 Rozšíření programu	26

1. Předmluva

Modul 010-090 je součástí 'Basics of PLC Programming' a poskytuje **rychlý vstupní bod** pro programování SIMATIC S7-1200 s využitím jazyka **S7 SCL** a TIA portal.



Cíle tréninku

Tento modul 010-090, představuje čtenáři základní vlastnosti a funkce S7 SCL development. Navíc jsou ukázány testovací procedury pro odstranění logických chyb při testování.

Prerekvizity

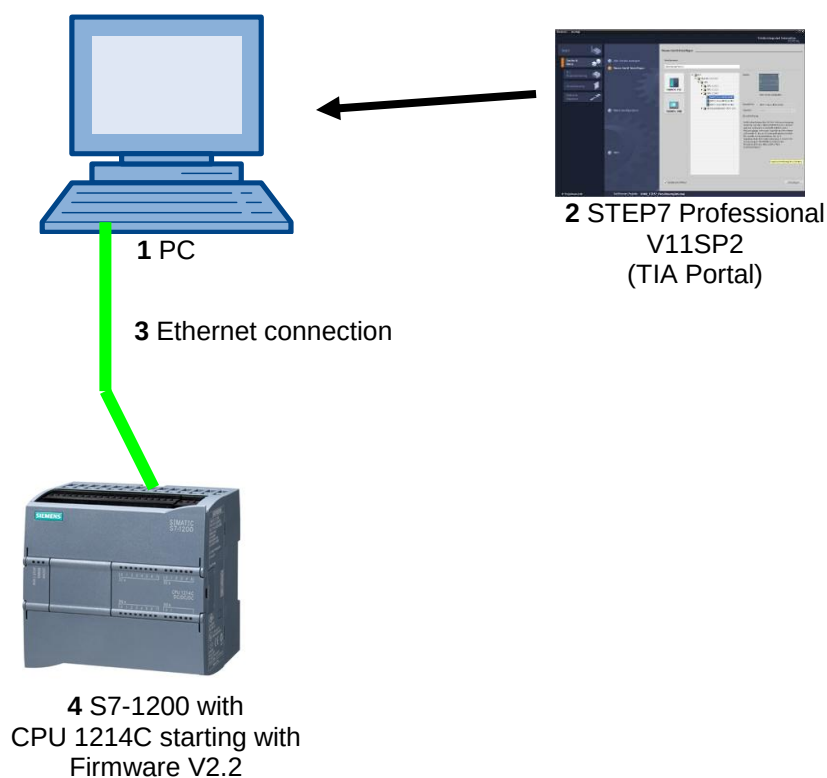
Pro úspěšnou práci s modulem 010-090, jsou potřeba následující znalosti:

- Práce s Windows
- Základy programování PLC se STEP7 Professional V11 (například, Modules 010-010 to 010-040)
- Základní znalost vyšších programovacích jazyků např. Pascal

Vyžadovaný hardware a software

- 1 PC Pentium 4, 1.7 GHz 1 (XP) – 2 (Vista) GB RAM, free disk storage approx. 2 GB
Operating system 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 Ethernet connection between PC and CPU 1214C
- 4 PLC SIMATIC S7-1200 starting with Firmware V2.2; for example, CPU 1214C with 14DI/10DO
signal module.

Výstupy musí být vyvedené na panel.



2. Představení jazyku S7 SCL

S7 SCL (Structured Control Language) je vysokoúrovňový programovací jazyk založený na pascalu a umožňuje strukturované programování. Jazyk odpovídá sekvenčnímu jazyku SFC "Sequential Function Chart" specifikovaném standardem DIN EN-61131-3 (IEC 61131-3). Navíc k vysokoúrovňovým vlastnostem přidává, S7-SCL typické PLC prvky jako vstupy, výstupy, časovače, flagy apod. jako své součásti. Podporuje koncept STEP 7 jako bloků a navíc k STL, LAD a FBD- standardům vyhovující programovací bloky. To znamená, S7-SCL doplňuje a rozšiřuje programovací software STEP 7 se svými jazyky LAD, FBD a STL.

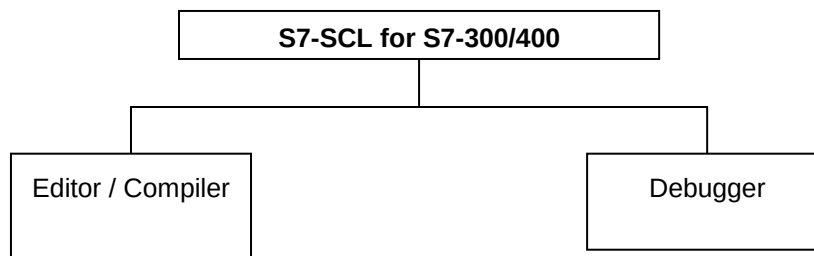
Nemusíte každou funkci vytvářet sami; jsou dostupné před sestavené bloky jako systémové funkce nebo systémové funkční bloky v operačním systému CPU.

Bloky programované v S7-SCL můžeme míchat s bloky v STL, LAD a FBD. To znamená, že blok naprogramovaný v S7-SCL může volat jiný blok programovaný v STL, LAD nebo FBD. Stejně tak S7-SCL blok může být volán z programů v STL, LAD a FBD.

S7-SCL test funkce umožňují hledat logické chyby v programu při chybné kompilaci.

3. S7-SCL vývojové prostředí

Pro použití S7-SCL, Vývojové prostředí je upravené pro jistá specifika S7-SCL spolu se STEP 7. Toto vývojové prostředí se skládá z Editor/Compiler a Debugger.



Editor/Compiler

S7-SCL editor je testový editor který umožňuje upravit jakýkoliv text. Základní úloha jakou budete provádět je vytváření a editace bloků pro programy ve STEP 7. Během psaní je prováděna základní kontrola sytů což zjednodušuje bezchybné programování. Chyby v syntaxi se zobrazují jinou barvou.

Editor nabízí následující:

- Programování bloků S7 v jazyce S7-SCL
- Pohodlné vkládání prvků jazyka a volání bloků pomocí drag&drop
- Přímá kontrola syntaxi během programování.
- Plné nastavení dle vašich zvyklostí a přání.
- Kontrola hotových bloků úplnou kompilací.
- Zobrazení všech chyb a varování, které se vyskytnou při kompilaci.
- Identifikace chyb v bloku, popis chyby a informace týkající se jejího odstranění.

Debugger

S7-SCL debugger dovoluje testování programu během běhu v AS a zjišťování možných logických chyb.

Do této chvíle nabízí S7-SCL dva různé testovací přístupy:

- Step by step monitoring
- Průběžný monitoring

V případě “step by step monitoring”, sleduje se logická sekvence programu. Můžete projít algoritmus programu instrukci po instrukci, a sledovat v okně tagů jak jsou jejich hodnoty průběžně měněny.

S “průběžným monitoringem” testujeme skupinu instrukcí v bloku. Během testovacího běhu jsou hodnoty tagů zobrazeny v chronologickém seznamu a aktualizovány periodicky.

4. Vzorová úloha objem v nádrži

4.1 Popis úlohy

Jako naši první úlohu, programujeme výpočet objemu v nádrži.

Nádrž je tvaru válce. Hladina obsahu je měřena analogovým senzorem. V úloze je hodnota hladiny už zjištěná a dodávána ve standardní jednotce 'Meter'.

Program bude naprogramován ve funkci FC140 'calculate_volume_tank'. Přenášené parametry jsou průměr nádrže a měřená hladina v metrech. Výsledkem je objem nádrže v litrech.

4.2 List úloh/ tabulka tagů

Nejdříve musíme deklarovat **globální PLC tagy**.

Tyto globální PLC tagy jsou popisnými jmény s komentáři pro vstupy a výstupy použité v našem programu. Později během programování k nim můžeme přistupovat jen pomocí jejich jmen. Globální tagy můžeme využít v celém programu ve všech blocích.

Základní tabulka tagů

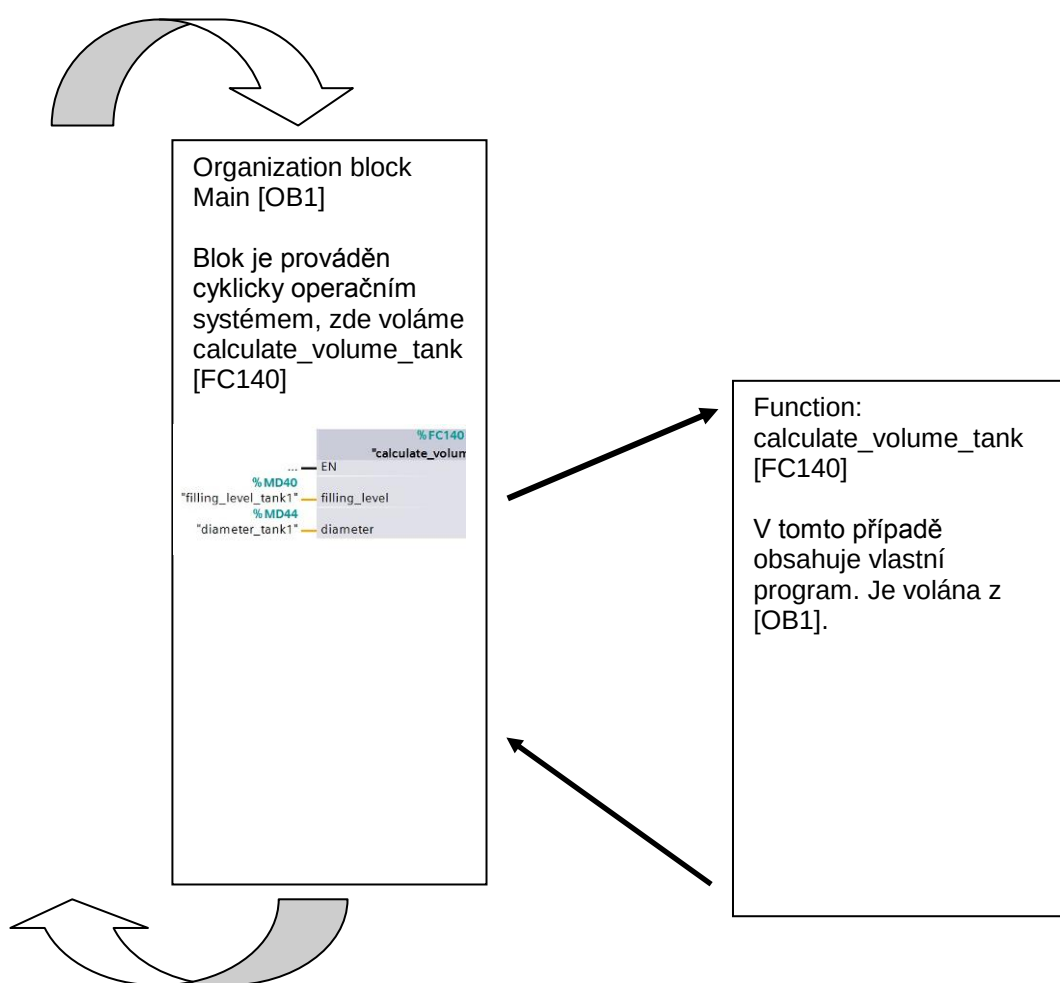
Jméno	Datový tip	Adresa	Komentář
filling_level_tank1	REAL	%MD40	naplnění tank1 (meter)
diameter_tank1	REAL	%MD44	průměr tank1 (meter)
volume_tank1	REAL	%MD48	objem tank1 (liter)

4.3 Struktura programu

Struktura programu je v blocích. Jako standard organizační blok Main [OB1] je vytvořen automaticky. Je to rozhraní mezi operačním systémem CPU a programem, blok je volán automaticky a prováděn cyklicky. Z tohoto bloku voláme další bloky programu, jako například funkci calculate_volume_tank [FC140].

Účelem je rozdělit program na více menších úloh, které můžeme vyřešit a testovat jednodušeji než velkou a komplexní úlohu.

Struktura vzorové úlohy výpočet objemu nádrže



4.4 Rozhraní bloku calculate_volume_tank [FC140]

Předtím než napíšeme samotný program musíme vytvořit blok rozhraní. Když je deklarované rozhraní, jsou v bloku známé pouze lokální tagy.

Tagy nebo parametry rozhraní jsou ze dvou skupin:

- Blokové parametry generující rozhraní bloku pro volání programu.

Typ	Jméno	Funkce	Dostupné v
Input parameter	Input	Parametr jehož hodnoty blok čte.	Functions, function blocks některé typy organization blocks
Output parameter	Output / Return	Parametr jehož hodnoty blok zapisuje.	Functions a function blocks
In/out parameter	InOut	Parametr jehož hodnoty blok při volání čte a po proběhnutí na něj zapisuje.	Functions a function blocks

- Lokální data využité k ukládání mezivýsledků.

Typ	Jméno	Funkce	Dostupné v
Temporary local data	Temp	Tag pro uložení dočasných dat, jsou uchovány jen jeden cyklus.	Functions, function blocks a organization blocks
Static local data	Static	Tag pro uložení statických dat, data jsou udržována tak dlouho dokud je nepřepíšeme znova.	Function blocks

Parametry rozhraní použité v našem programu 'calculate_volume_tank [FC140] jsou:

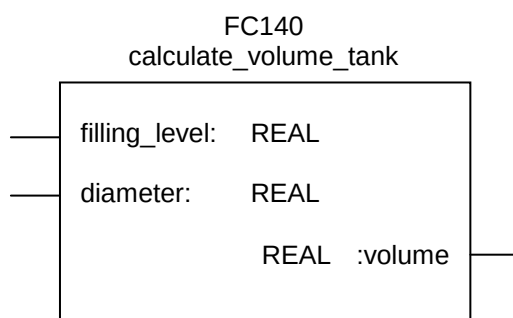
Rozhraní funkce FC140:

calculate_volume_tank

Typ	Jméno	Data type	Komentář
IN	filling_level	REAL	Hladina naplnění tank (meter)
IN	diameter	REAL	Průměr nádrže tank (meter)
OUT	volume	REAL	Objem tekutiny v nádrži (liter)/ -1 chyba

Volání funkce v FBD vypadá následně:

Volání funkce: Reprezentace v FBD



4.5 Poznámka k řešení

Pro řešení úlohy je využit vzorec pro objem válce. Pro výsledky v litrech musíme vynásobit 1000.

$$V = \frac{d^2}{4} \cdot \pi \cdot h \quad \Rightarrow \quad \text{objem} = \frac{\text{průměr}^2}{4} \cdot 3.14159 \cdot \text{filling_level} \cdot 1000$$

5. Programování výpočtu objemu nádrže pro SIMATIC S7-1200 v S7-SCL

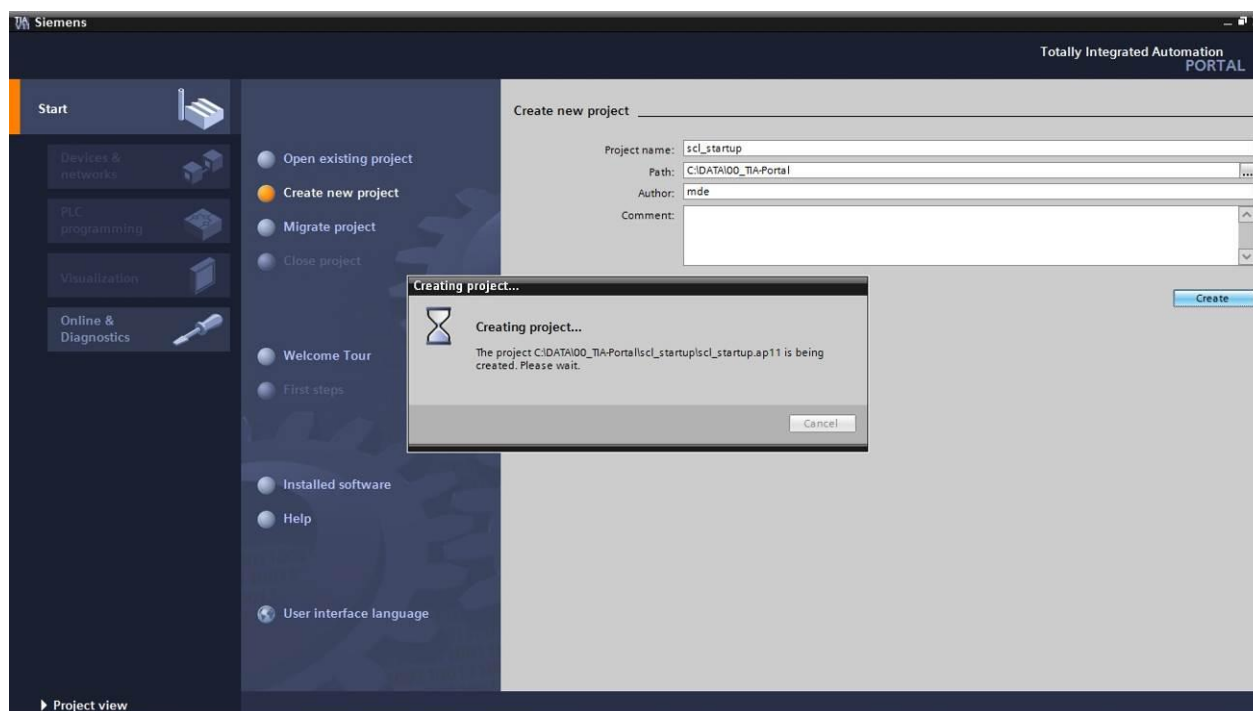
V krocích níže, nastavíme S7-1200 a naprogramujeme naši úlohu.

5.1 Nastavení projektu a konfigurace hardware

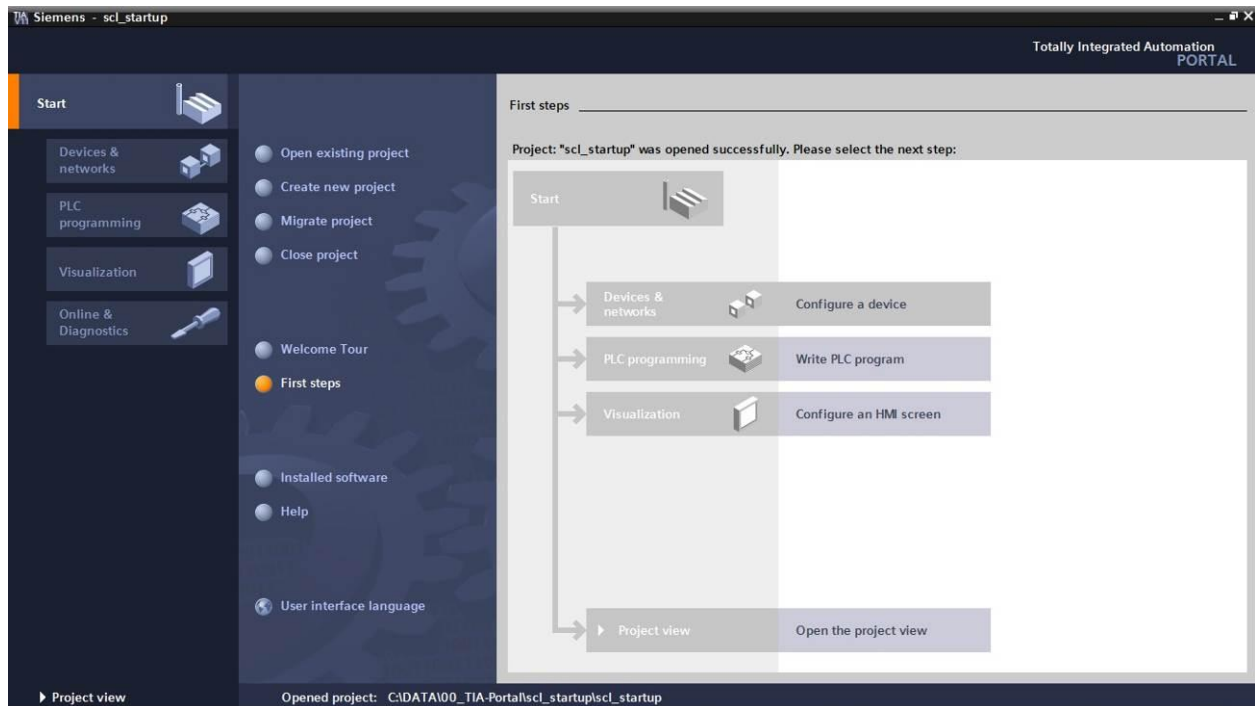
1. Centrálním nástrojem je '**Totally Integrated Automation Portal**'. Otevřeme dvojklikem. (→ TIA Portal V11)



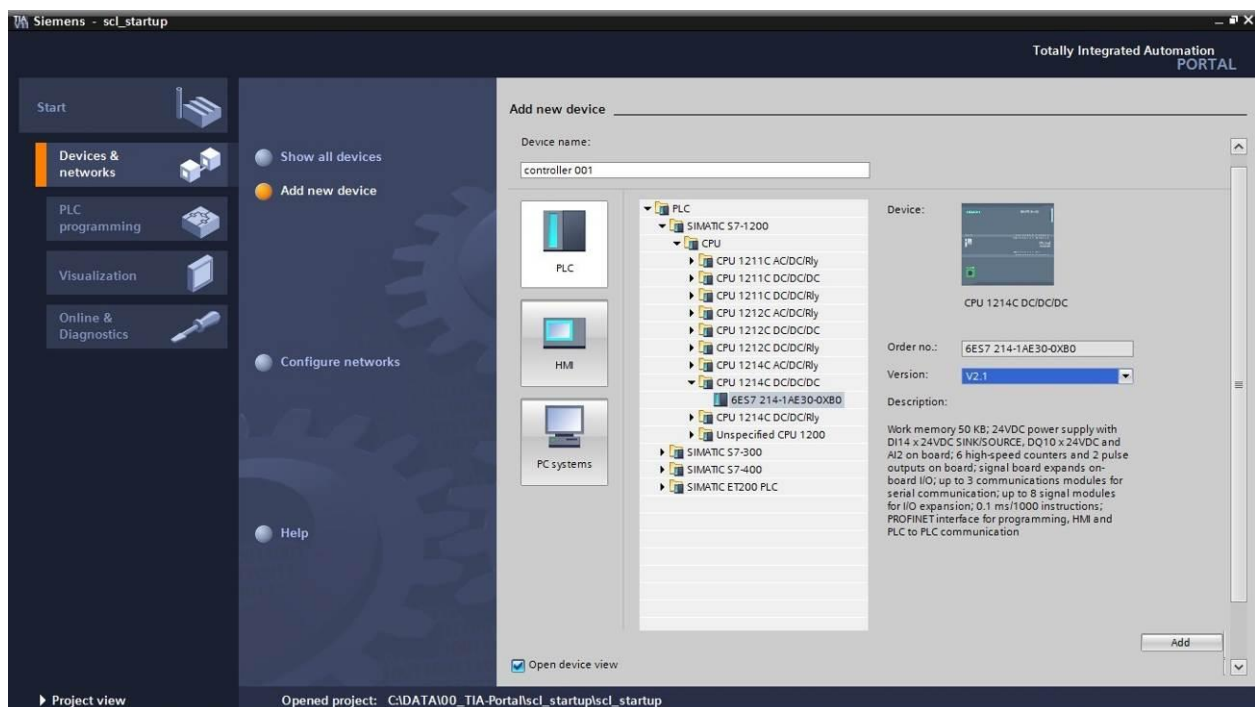
2. Programy pro SIMATIC S7-1200 jsou spravovány v projektech. Nyní nastavíme takový projekt v portal view. (→ Create new project → scl_startup → Create)



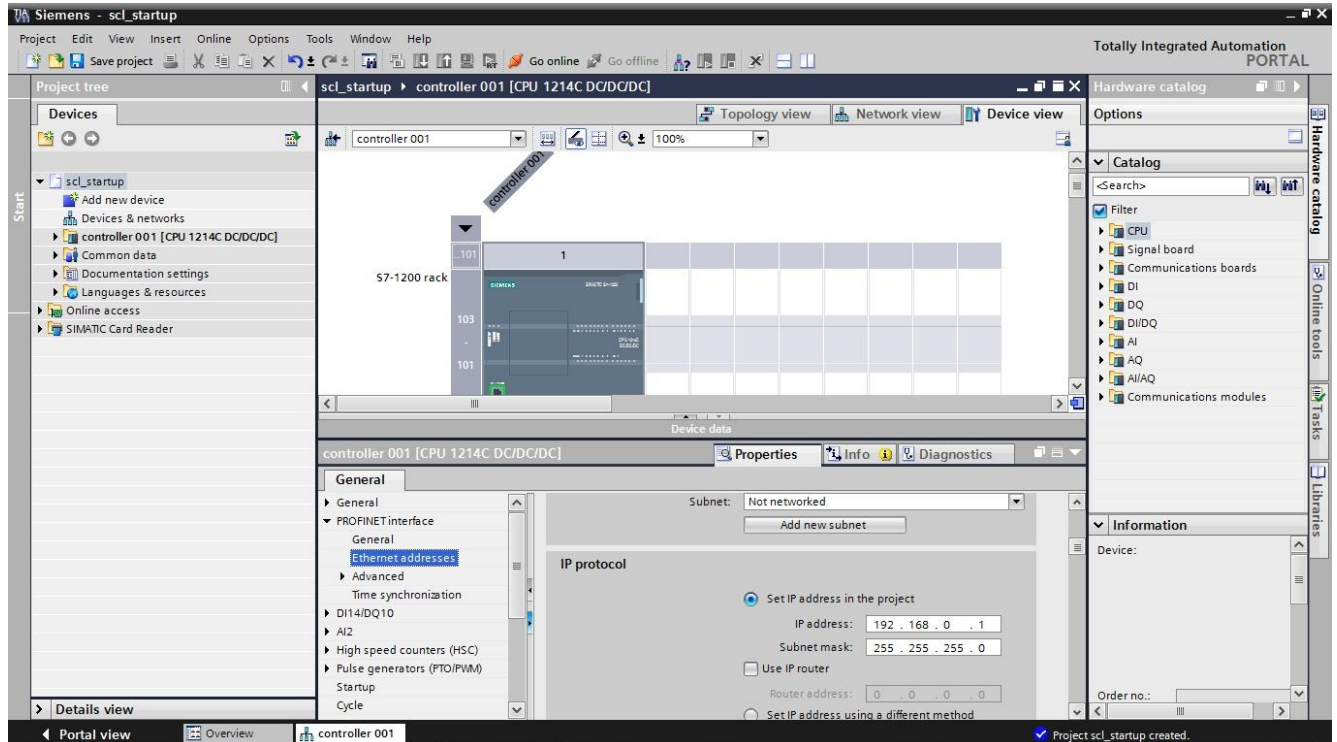
3. Nyní, 'First steps' jsou nabídnuty kroky pro konfiguraci. Nejprve chceme 'Configure a device'. (→ First steps → Configure a device)



4. Poté 'Add new device' s 'Device name controller 001'. Vybereme z katalogu 'CPU1214C AC/DC/Rly' se shodným objednáčím číslem. (→ Add new device → controller 001 → PLC → SIMATIC S7-1200 → CPU → CPU1214 AC/DC/Rly → 6ES7 214-1BE30-0XB0 → V2.2 → Add)

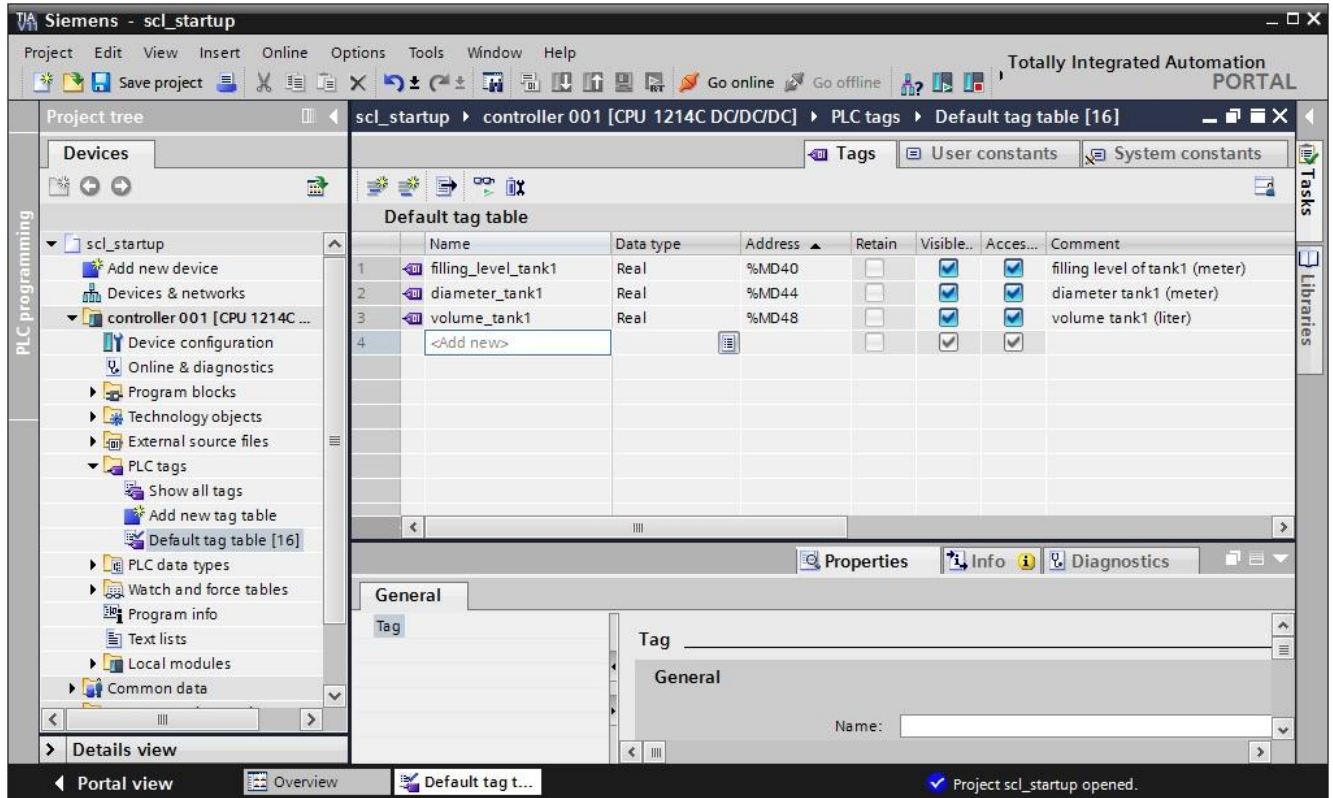


- Software se sám přepne do project view a otevře nám konfiguraci zařízení. Pro pozdější přístup k CPU musíme nejdříve nastavit **'ETHERNET address'**.
(→ Properties → General → ETHERNET address → IP address: 192.168.0.1 → Subnet mask: 255.255.255.0)

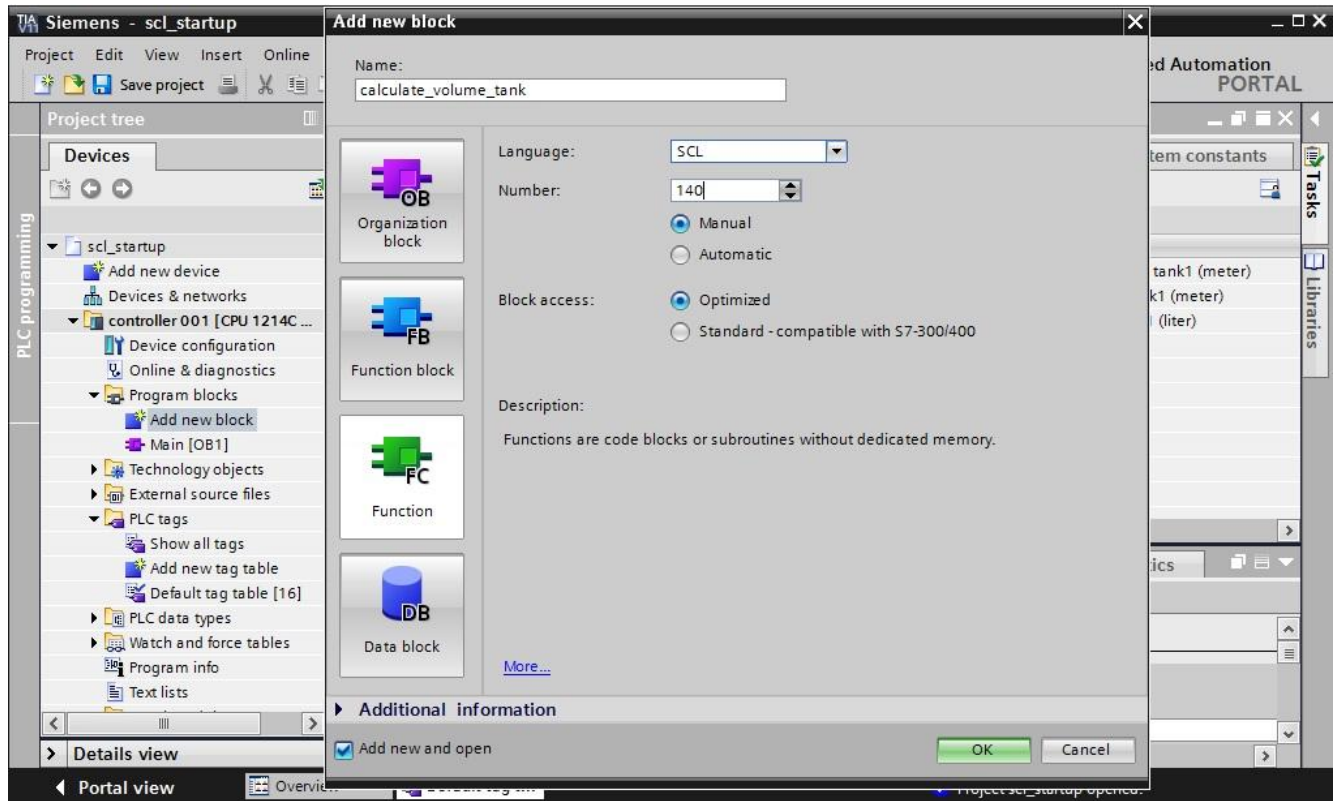


5.2 Nastavení programu

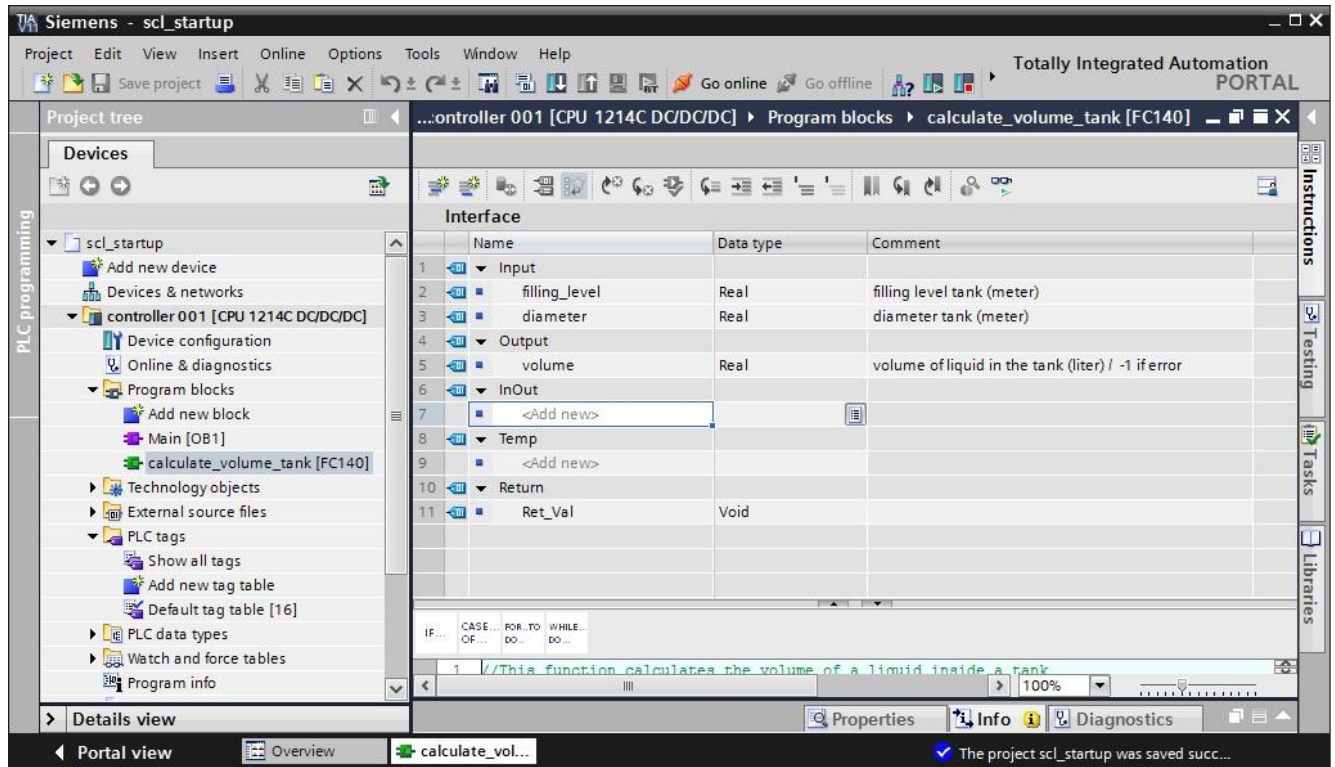
1. Otevřeme '**Default tag table**' a zadáme '**Operands**' s jejich jmény a datovými typy.
(→ controller 001[CPU1214 AC/DC/Rly] → PLC tags → Default tag table → Add new)



- Pro vytvoření funkce calculate_volume_tank [FC140], v projektové navigaci vybereme 'controller 001[CPU1214 AC/DC/Rly]' a poté 'Program blocks'. Dvojklik na 'Add new block'. Vybereme 'Function (FC)' a přiřadíme jméno 'calculate_volume_tank'. Změníme programovací jazyk na 'SCL' Číslování můžeme mít automatické i manuální. Zadáme číslo 140. A potvrdíme 'OK'.
(→ controller 001[CPU1214 AC/DC/Rly] → Program blocks → Add new block → Function (FC) → calculate_volume_tank → SCL → manual → 140 → OK)



3. Blok 'calculate_volume-tank [FC140]' je otevřen automaticky. Nyní zadáme '**Input and Output parameter**'. Všechny lokální tagy patřičně okomentujeme '**sufficient comment**' pro lepší porozumění
(→ Enlarge interface of block FC140 → enter interface parameters)

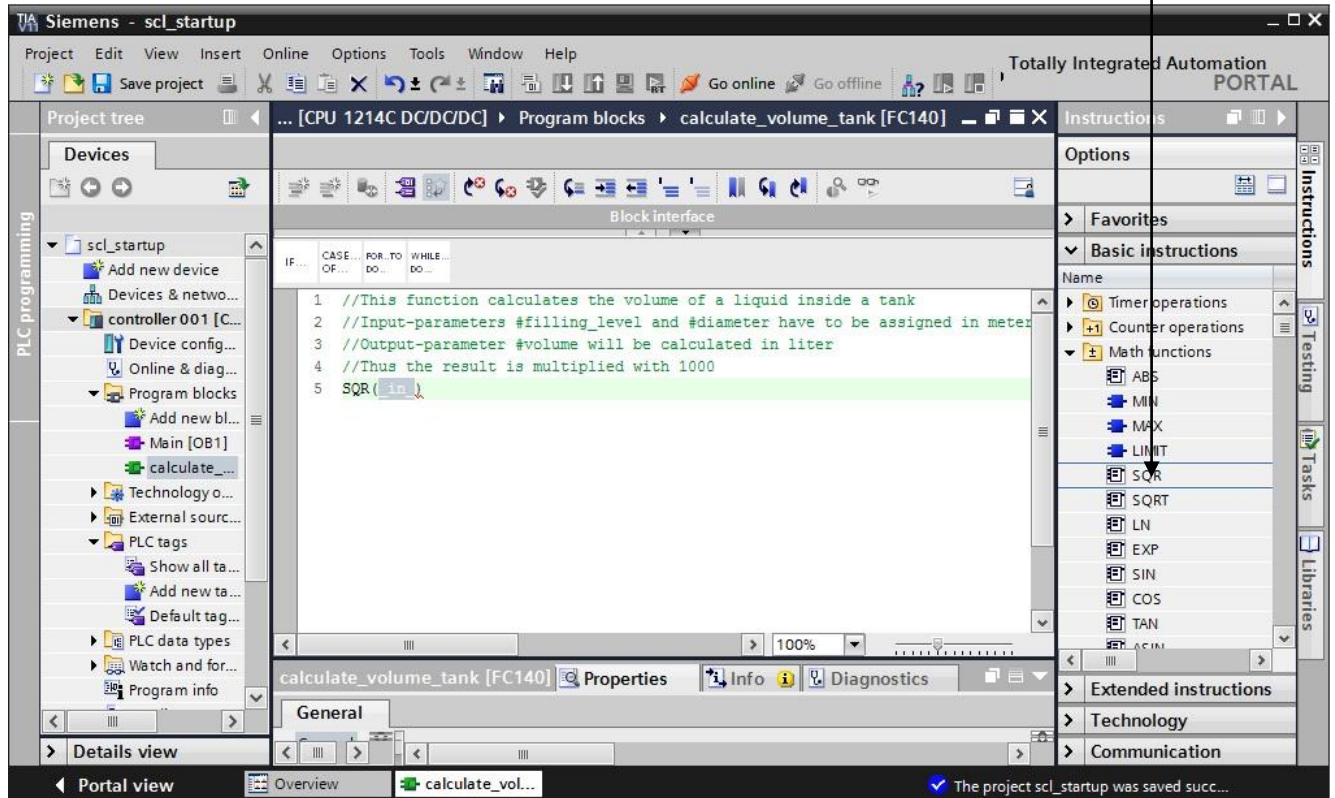


Poznámka:

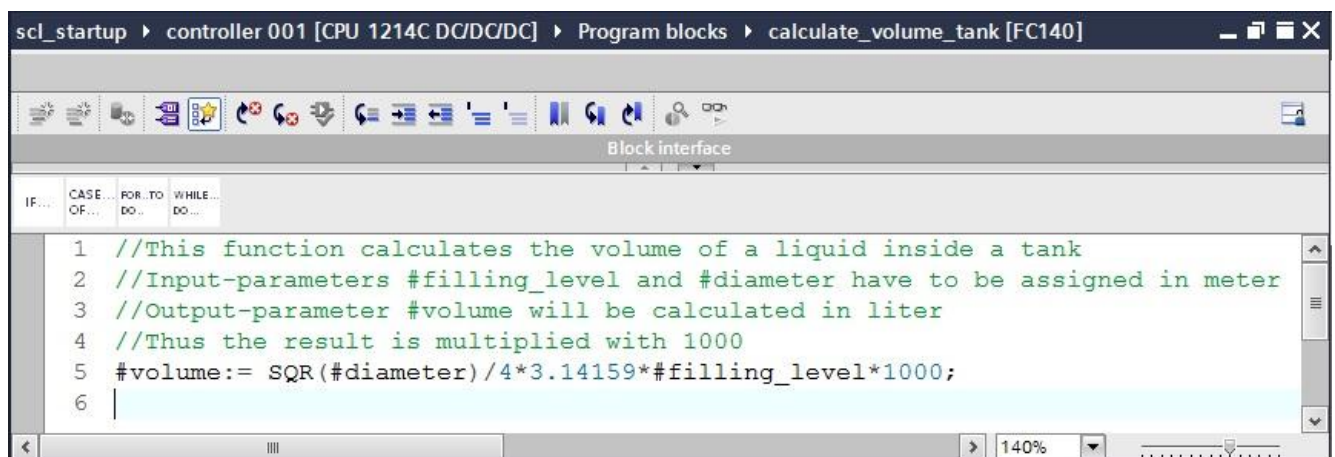
Pro vyhnutí se zmatení s PLC tagy je lepší psát lokální tagy malými písmeny.

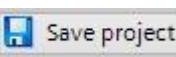
- Po zadání lokálních tagů můžeme začít vytvářet program. Funkce na druhou 'SQRT' je integrovaná v SCL. Číslo které chceme mocnit je v závorce. Pro vložení funkce ji přetáhneme na místo kde ji chceme použít.

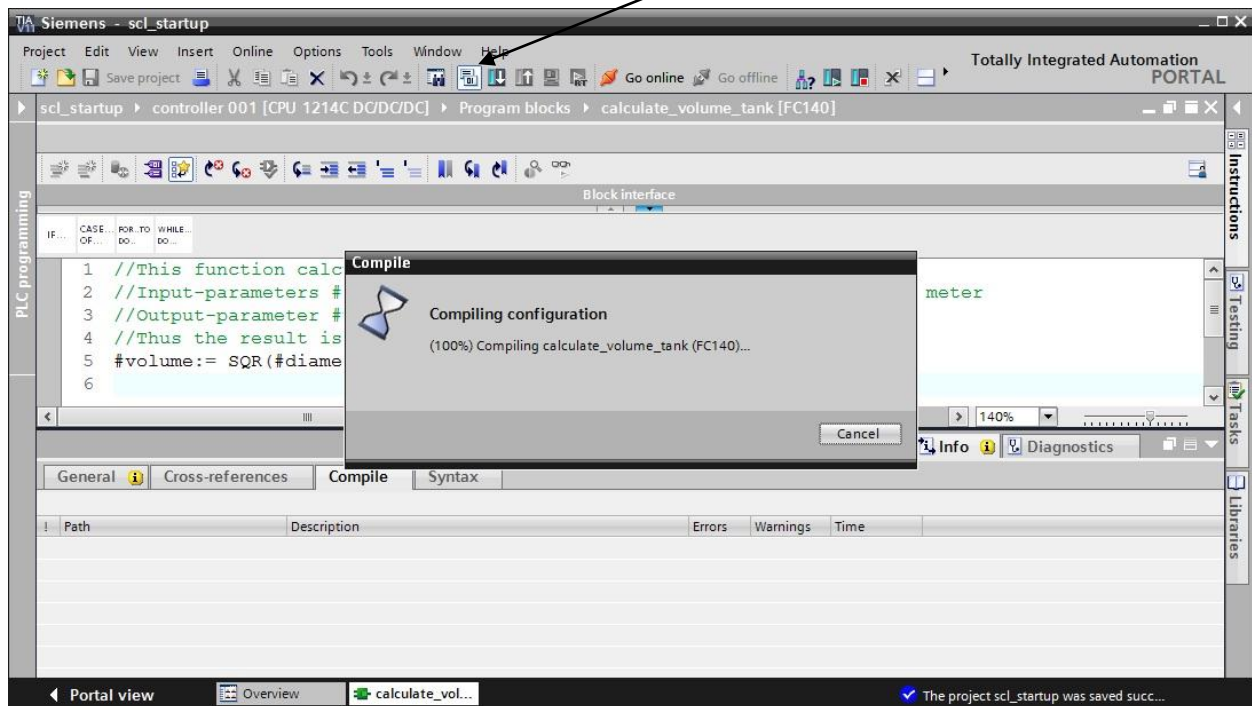
(→ Instructions (menu to the right!) → Simple instructions → Math functions → SQR)



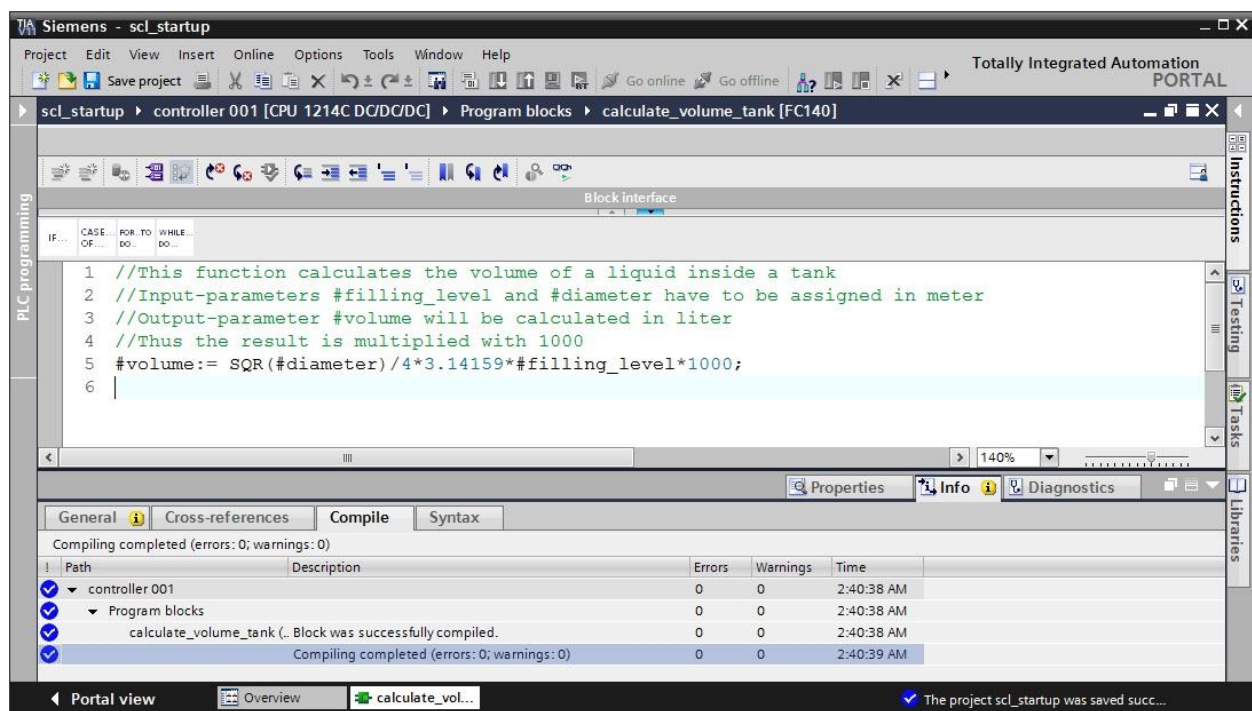
- Program vypadá jako níže.



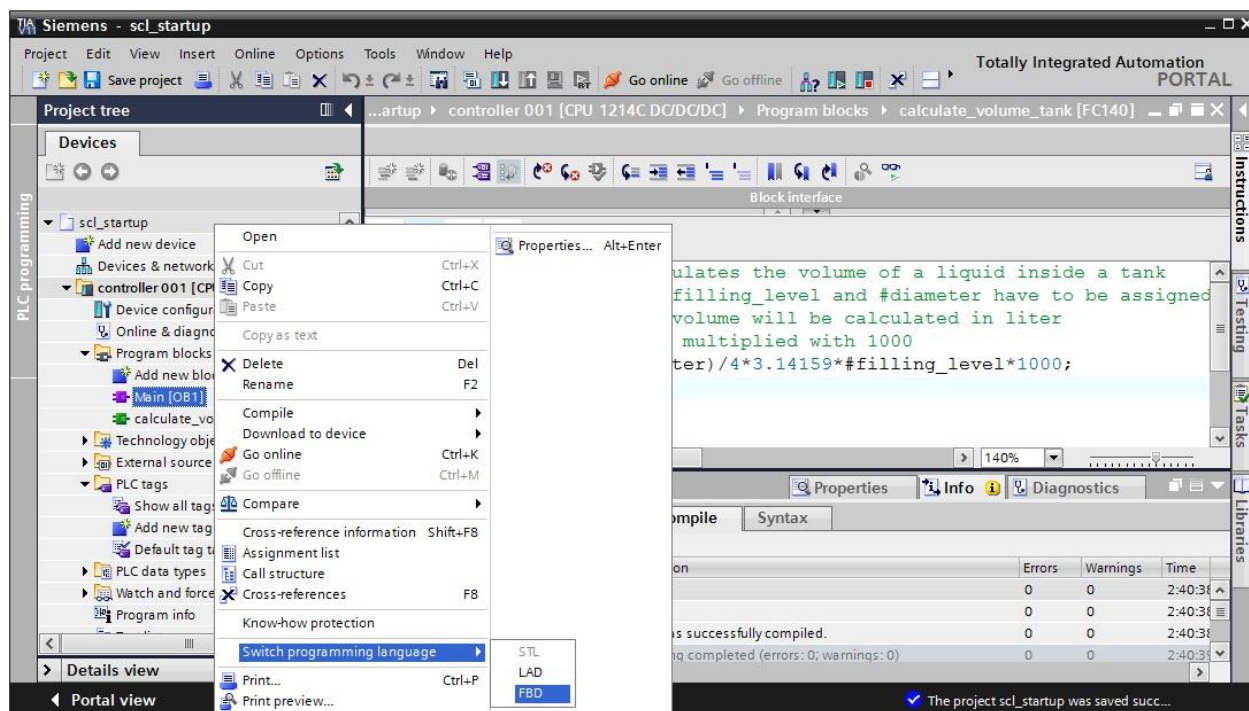
6. Nyní program uložíme  a zkompilujeme .
- (→ Save project  → Compile )



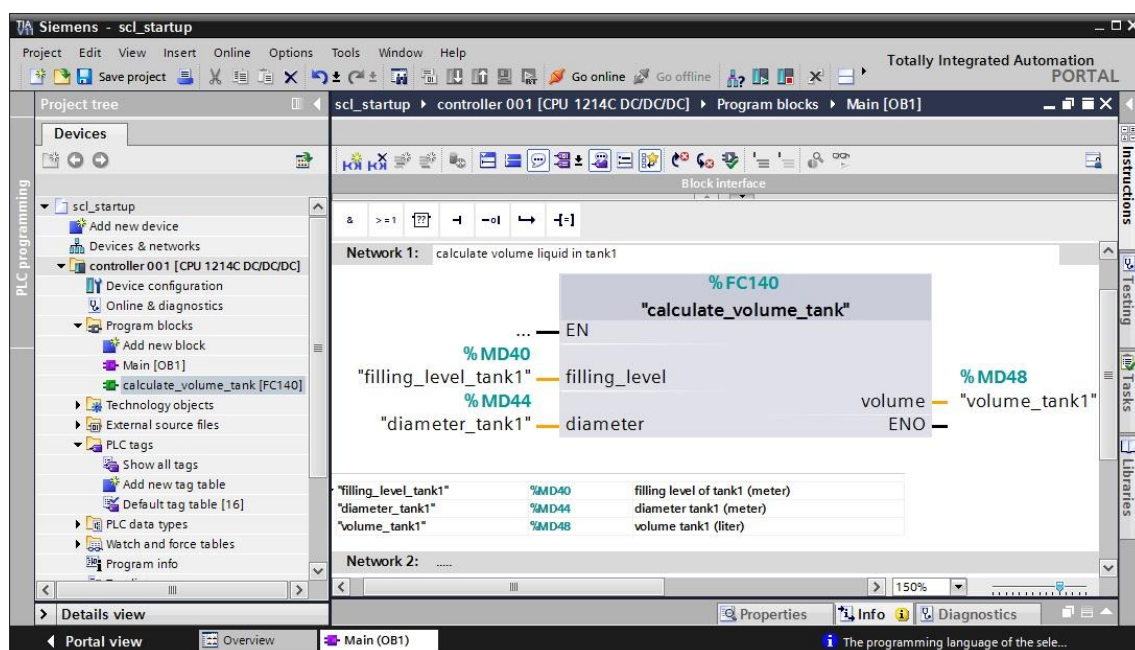
7. Chyby v syntaxi zjištěné během kompilace se zobrazí v 'Menu info/Compile'. (→ Info → Compile)

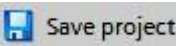
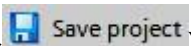


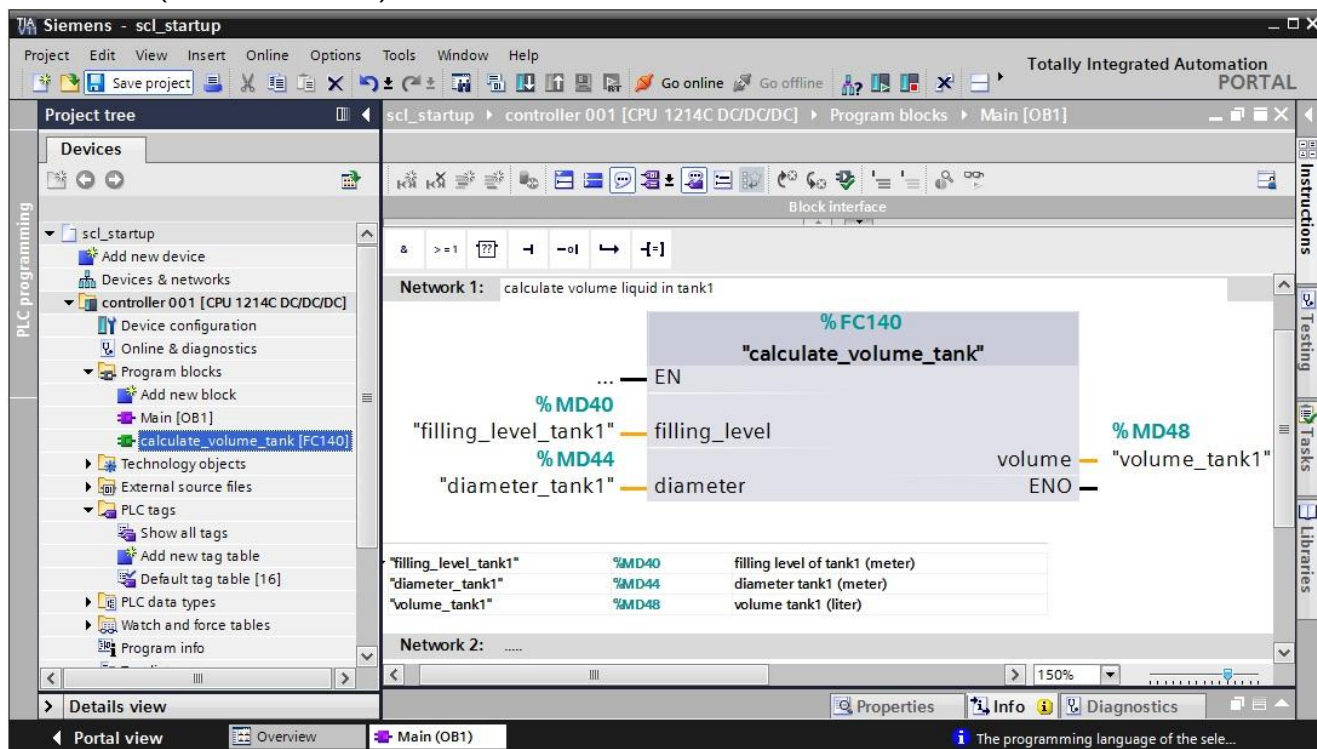
8. Nyní můžeme volat funkci v 'Main[OB1]'. Předtím než otevřeme blok 'Main[OB1]' dvojklikem změníme jeho programovací jazyk na 'FBD'.
(→ Main[OB1] → Switch programming language → FBD)



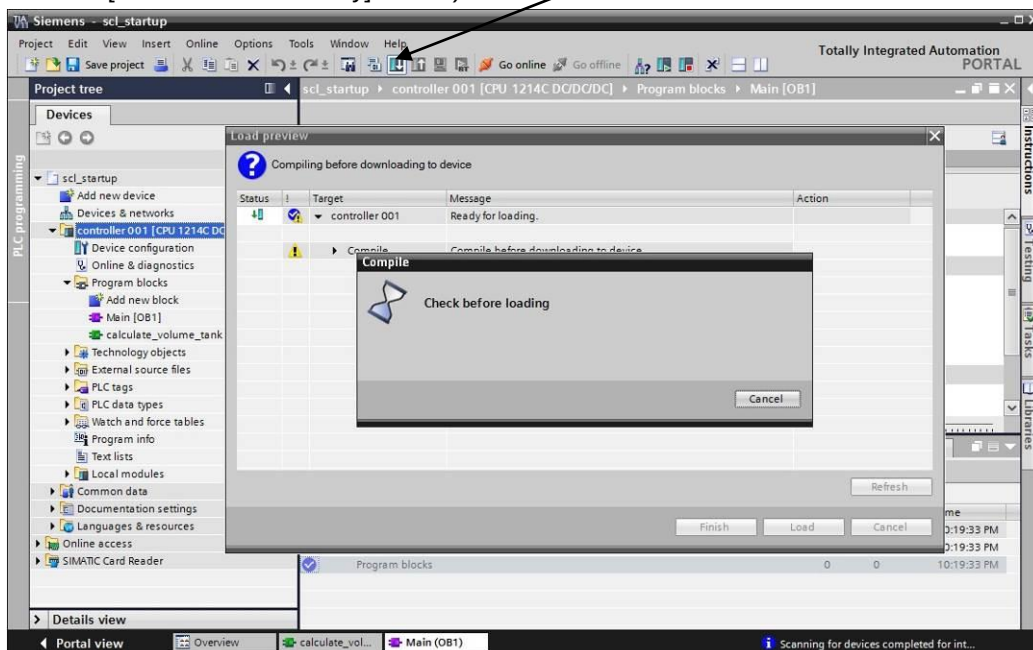
9. Nyní jednoduše přetáhneme funkci 'calculate_volume_tank[FC140]' do Network 1 v Main[OB1]. Parametry rozhraní funkce 'calculate_volume_tank[FC140]' musíme připojit ke globálním PLC tagům, tak jak je ukázáno. Nezapomeneme dokumentovat síť v Main[OB1]. (→ Main[OB1] → Program blocks → calculate_volume_tank[FC140])



10. Uložíme opět projekt  Save project
(→  Save project)



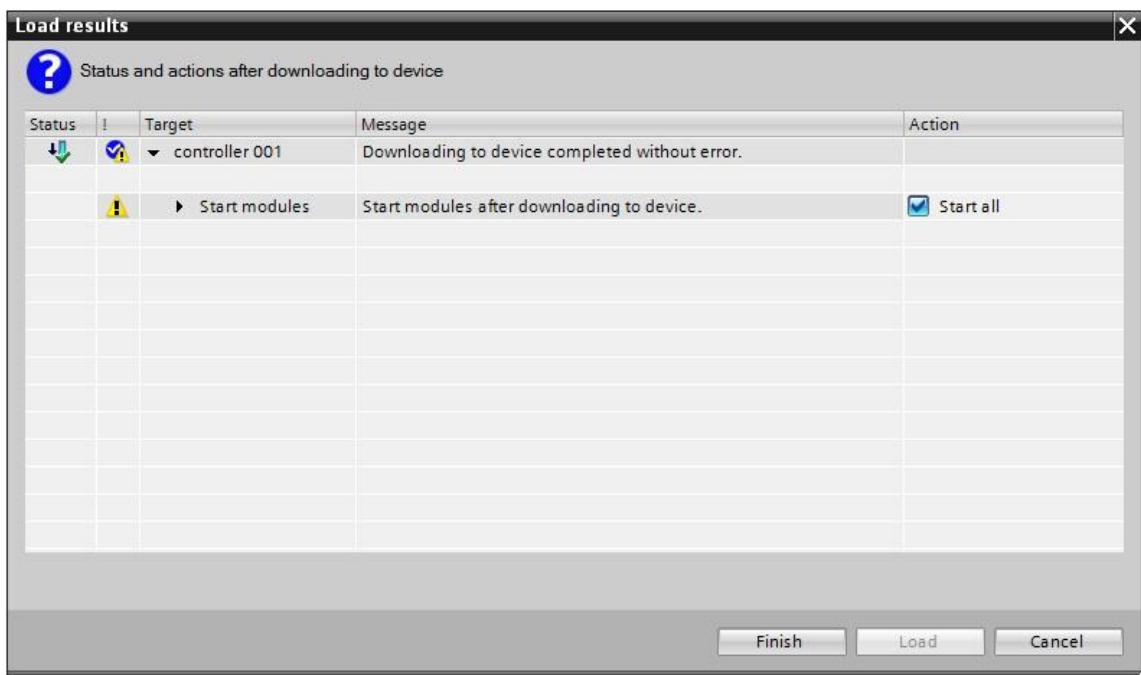
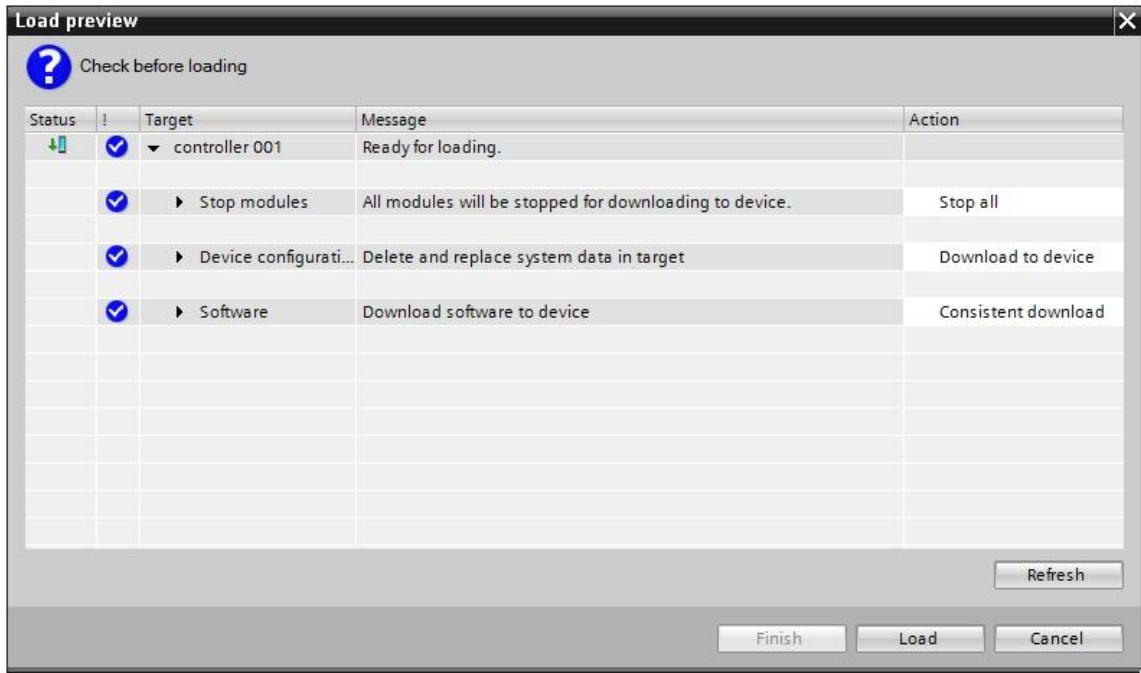
11. Nyní pro nahrání programu a konfigurace do CPU, nejdříve vybereme složku '**controller 001[CPU1214 AC/DC/Rly]**' a pak klik na symbol  nahrajeme do zařízení. (→ controller 001[CPU1214 AC/DC/Rly] → )



Poznámka:

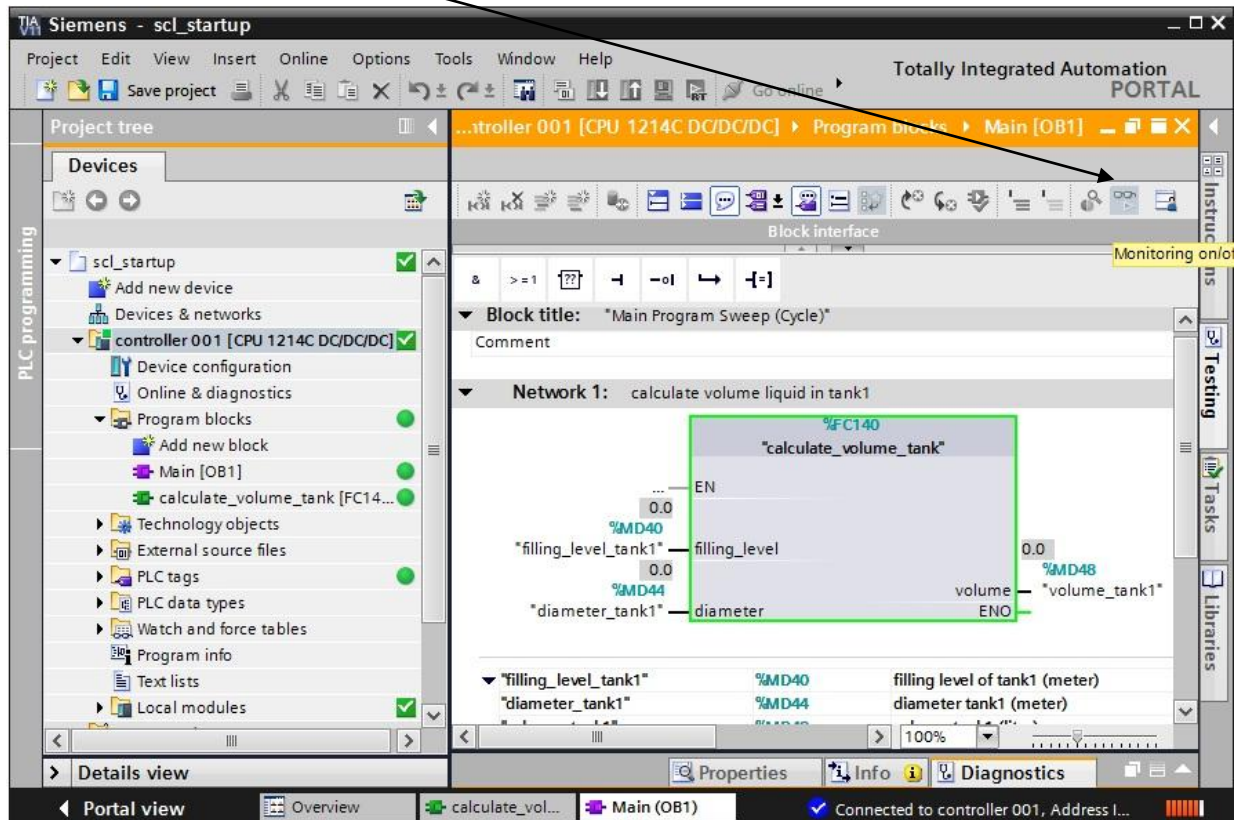
Během nahrání je projekt ještě jednou kompilován a hledají se syntaktické chyby.

12. Před nahráním se zobrazí seznam úkonů, které proběhnou před samotným nahráním. Spustíme seznam kliknutím na 'Load', a spustíme CPU 'Start all' 'Finish'. (→ Load → Start all → Finish)



5.3 Testování programu

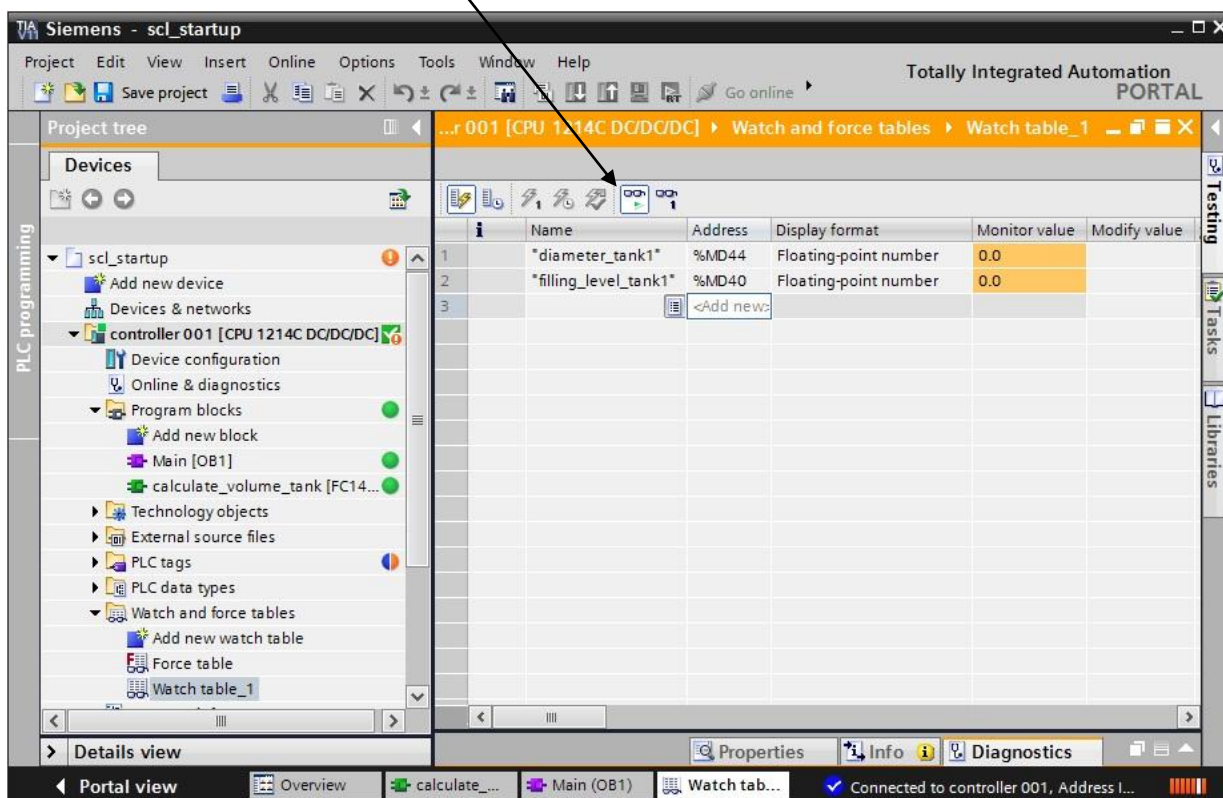
1. Kliknutím na symbol  Monitoring on/off, můžete sledovat status vstupů a výstupů bloku 'calculate_volume_tank' během testování programu.



2. Protože nemáme analogový senzor musíme si hodnoty 'diameter_tank1' a 'filling_level_tank1' vytvořit sami v monitorovací tabulce. Nastavíme 'new monitoring table' a zadáme dvě hodnoty. Spustíme 'Monitoring mode' pro sledování aktuálních hodnot.

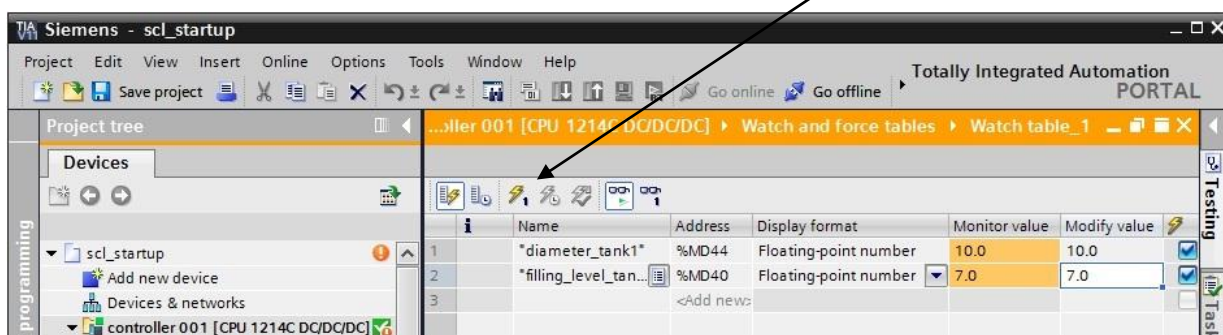
(→ controller 001 → Monitoring and force table → New monitoring table → diameter_tank1,

filling_level_tank1 → )

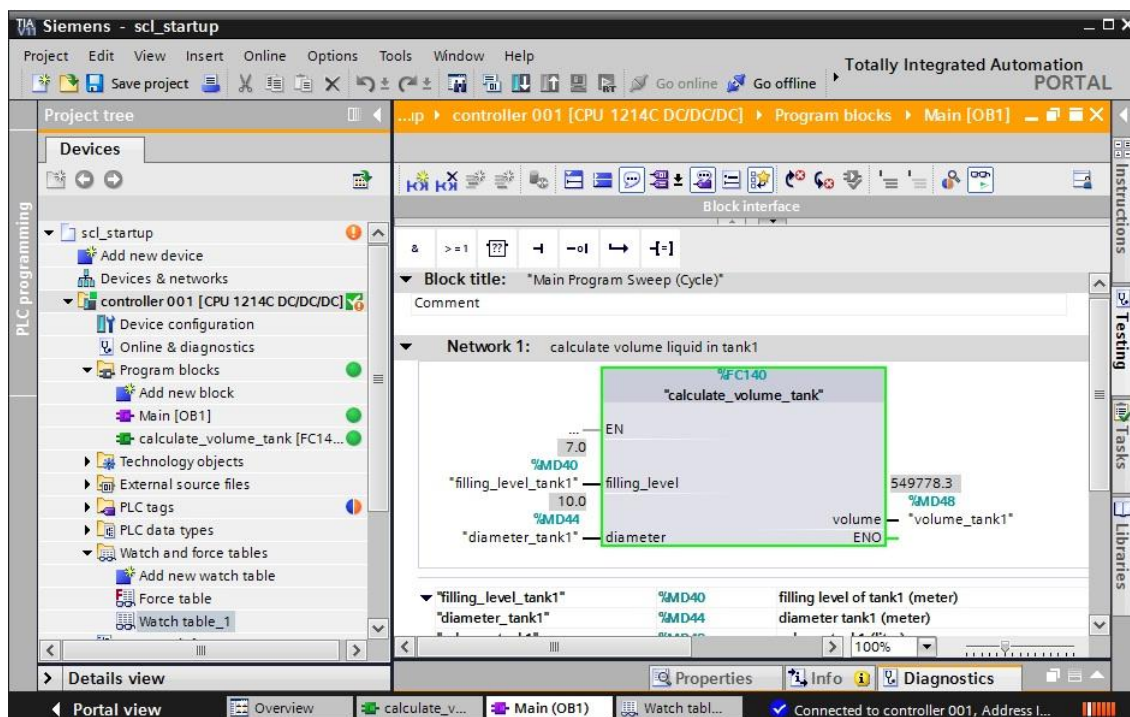


3. Pro specifikaci hodnot, přidáme 'Control value'. Tlačítkem 'Write control values once and immediately' , jsou hodnoty přeneseny do CPU.

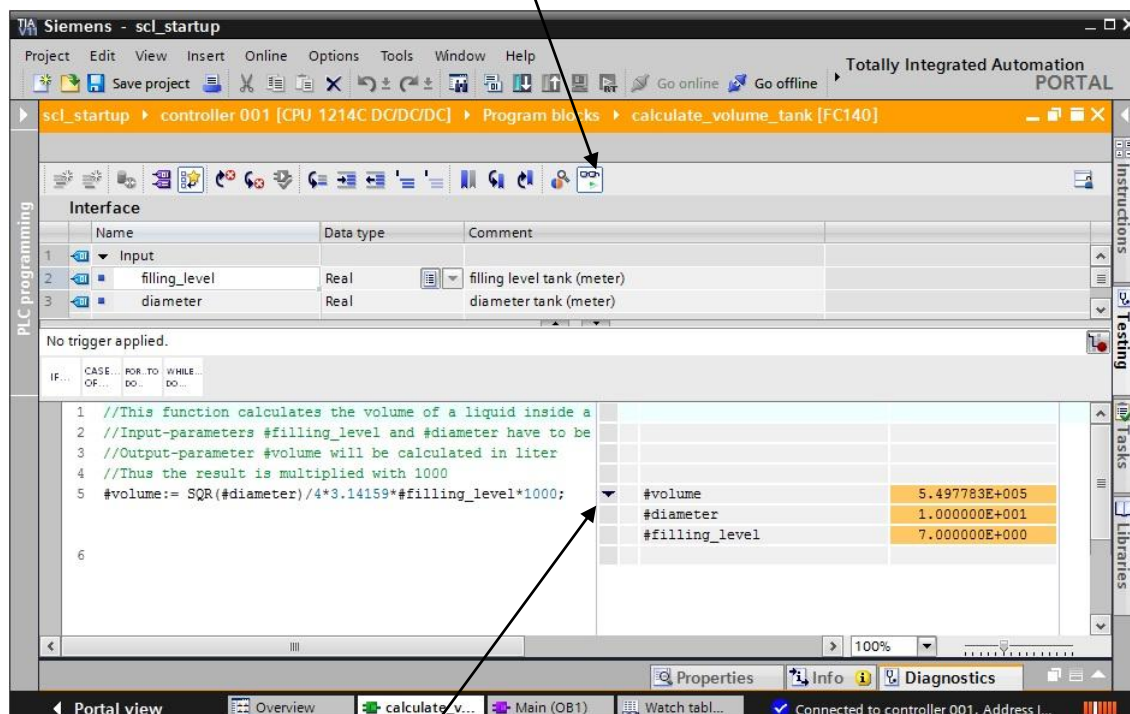
(→ diameter_tank1 = 10.0 → filling_level_tank1 = 7.0 → )



4. Nyní otestujeme program v OB1. (→ )



5. Můžeme sledovat hodnoty jednotlivých tagů v SCL editoru. Pro teď spustíme monitorovací mod 'Monitoring mode on'. (→ )



Poznámka:

Kliknutím na tlačítko , jsou zobrazeny aktuální hodnoty tagů z daného řádku.

5.4 Rozšíření programu

Nyní blok **'calculate_volume_tank'** kontroluje jestli na vstup nepřichází nesmyslná data. Navíc je hodnota **'height_max'** maximální hladiny přidána do bloku. Specifikuje výšku nádrže.

Blok vyhodnocuje, jestli je hladina méně než 0 anebo více než je max. výška nádrže. Navíc také kontroluje správnost zadání průměru nádrže. Pokud je někde chyba je boolean parametr **'er'** TRUE, a hodnota parametru **'volume'** -1.

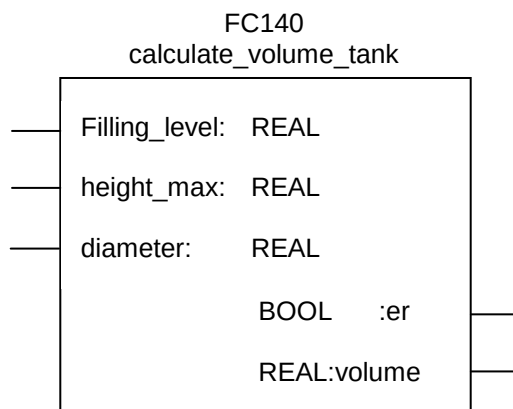
Rozšíření listu úlohy/ tabulky tagů:

adresa	symbol	Data Type	komentář
%Q1.7	error_bit	BOOL	chyba, nelze vypočítat

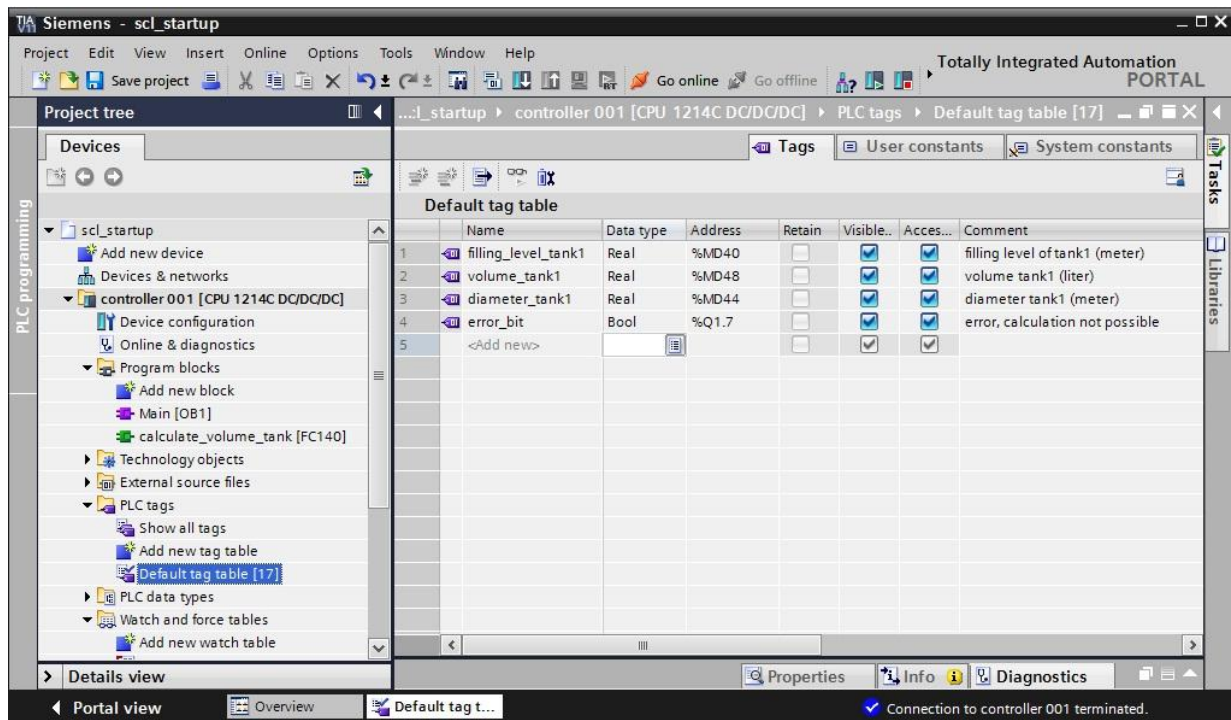
Rozšíření rozhraní funkce FC140: calculate_volume_tank

Typ	Symbol	Data Type	Komentář
IN	height_max	REAL	Maximální výška (meter)
OUT	er	BOOL	er = true -> error

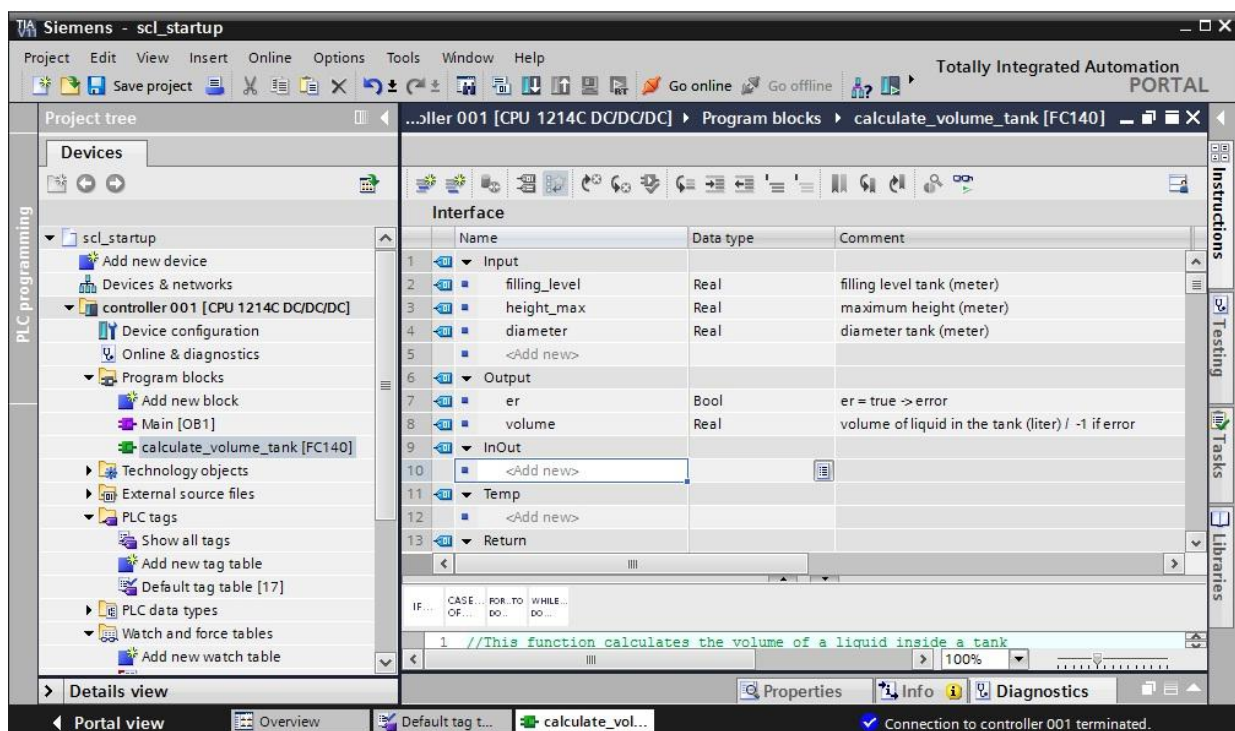
Volání funkce: reprezentace v FBD



1. Rozšíříme blokovou 'tag table' jak je specifikováno výše.
(→ controller 001[CPU1214 AC/DC/Rly] → PLC tags → default tag table → enter operands)

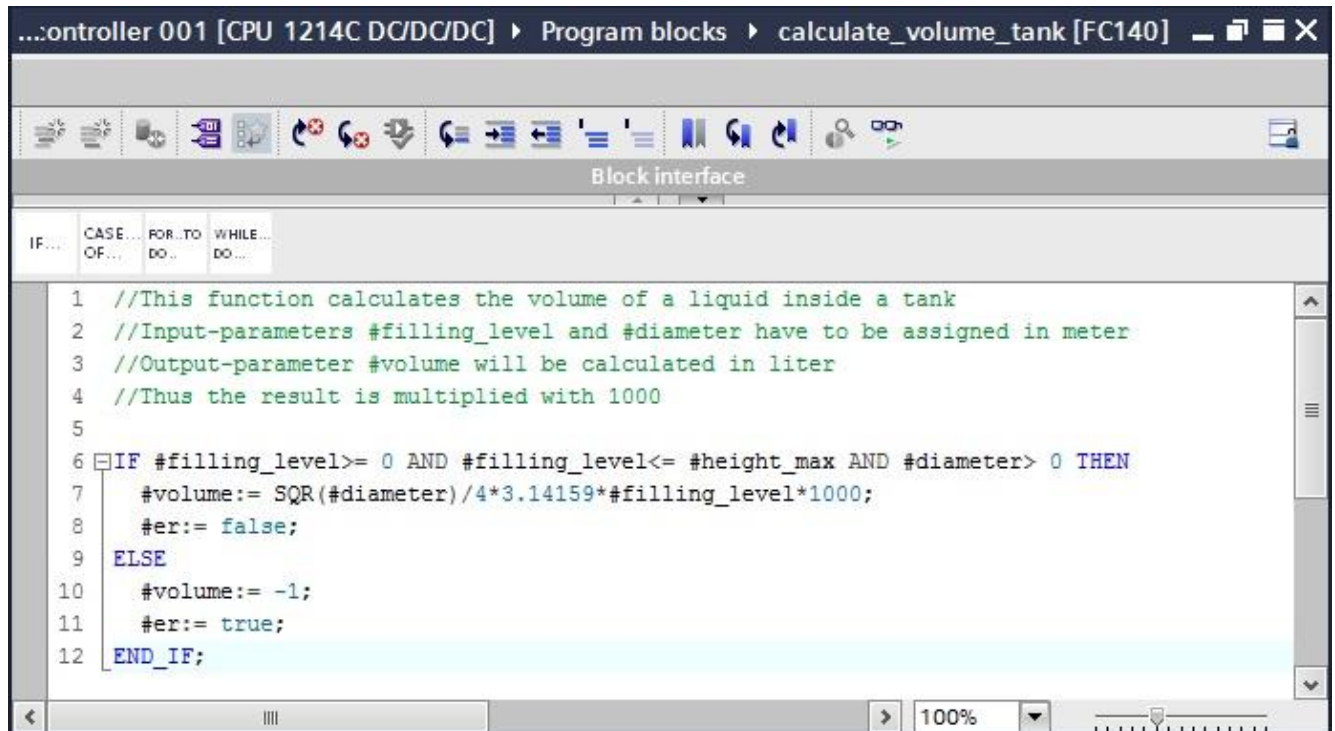


2. Rozšíříme blokové rozhraní 'Interface parameters' jak specifikováno výše.
(→ controller 001[CPU1214 AC/DC/Rly] → Program blocks → calculate_volume_tank) → enter parameters)

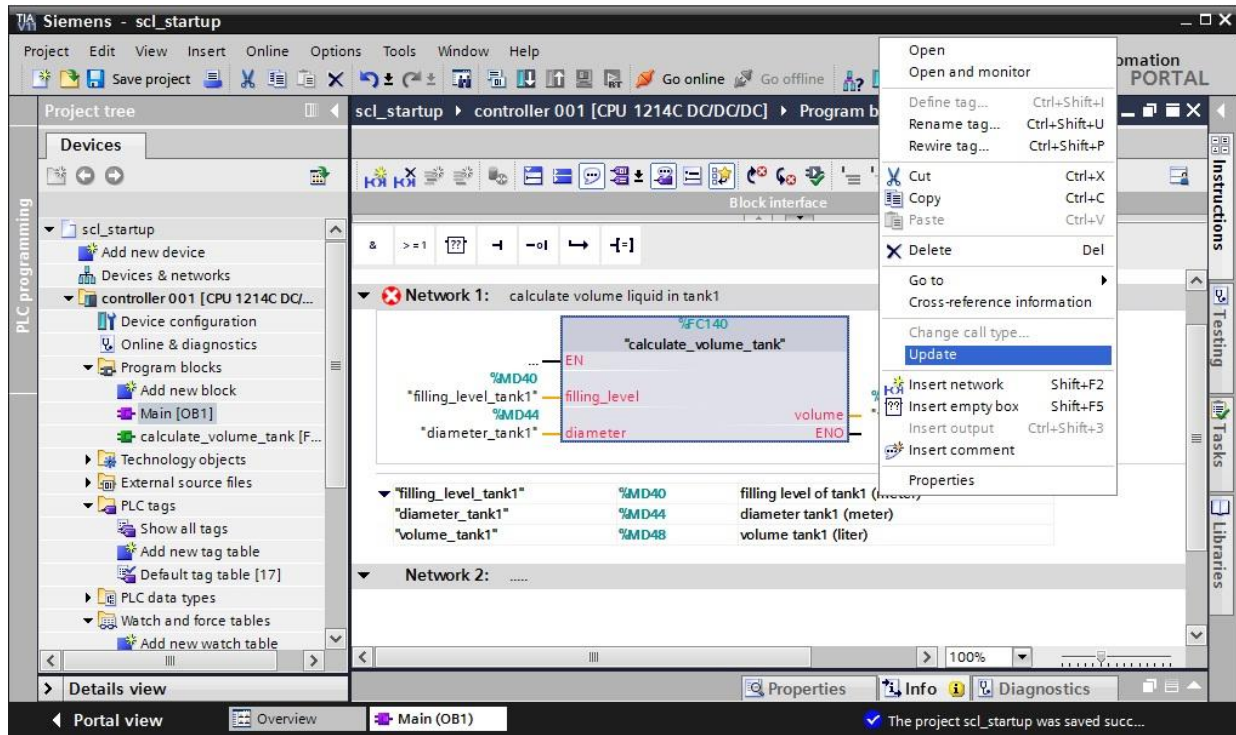


3. Nyní rozšíříme program jak je ukázáno níže, zkompilujeme pro nalezení syntaktických chyb. Uložíme a nahrajeme do CPU.

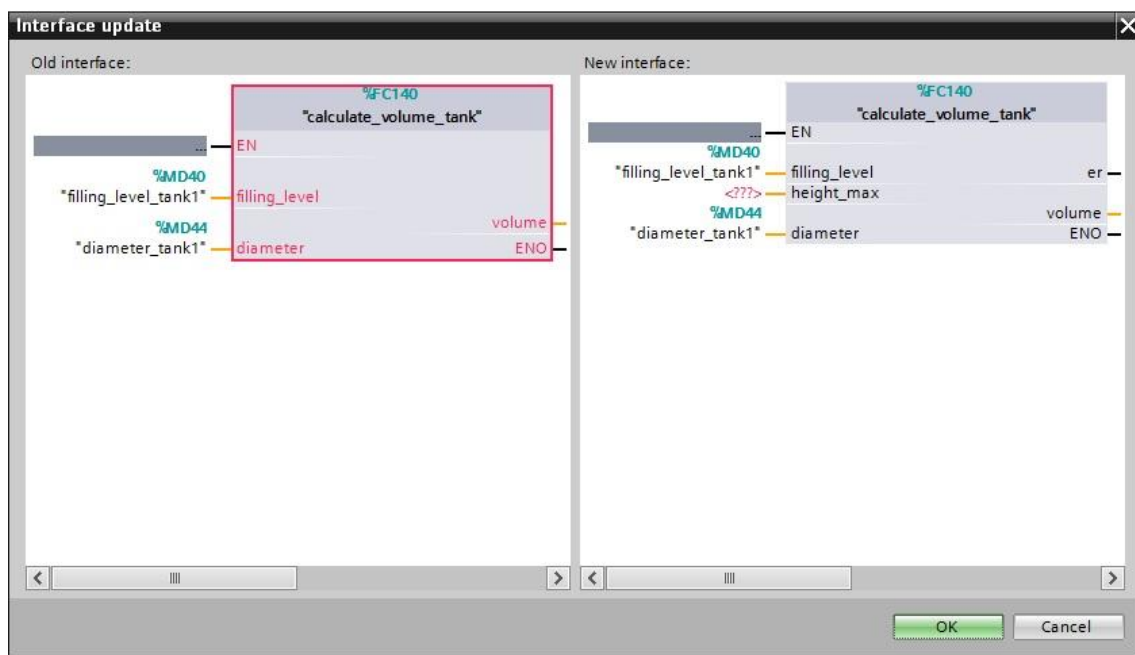
(Write program →  →  Save project → )



- Protože blokové parametry jsou změněny musíme upravit i volání z OB1. Otevřeme OB1 a najdeme místo kde je blokové volání. Pravou myší otevřeme kontext menu a klikneme na 'Update'.
(→ controller 001[CPU1214 AC/DC/Rly] → Program blocks → Main [OB1]
→ right mouse key → Update)

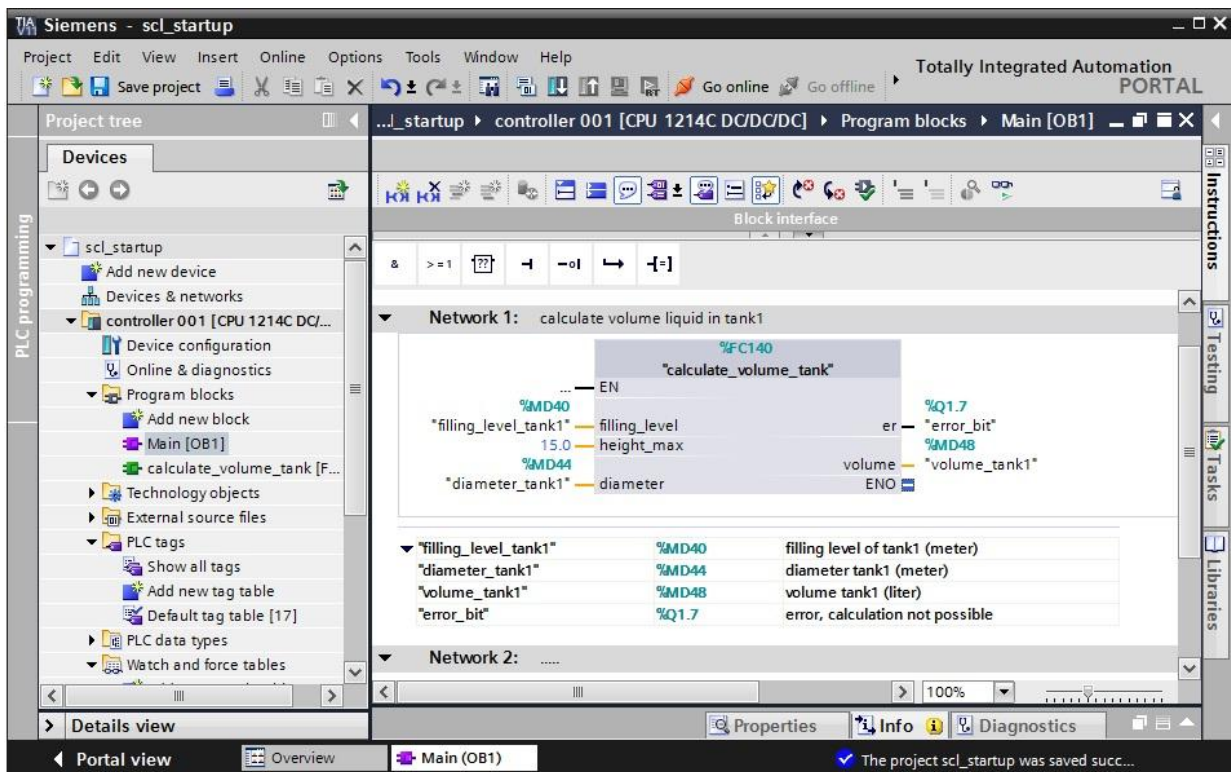


- Nyní se zobrazí nové a staré rozhraní, potvrdíme 'OK'.
(→ OK)



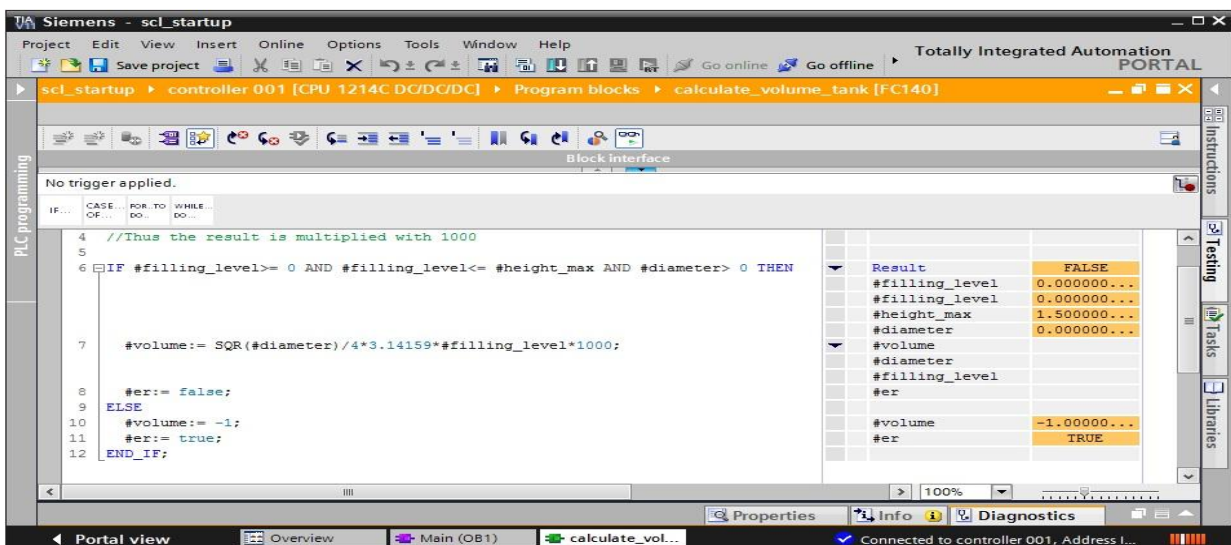
6. Doplníme hodnotu do parametru '**height_max**' a na výstupní parametr '**er**' jak je ukázáno níže. Zkompilujeme, uložíme a nahrajeme do CPU.

(Supplement parameters →  →  Save project →  → )



7. Zkontrolujeme změny v 'Monitoring mode' bloku 'calculate_volume_tank'.

(→ controller 001[CPU1214 AC/DC/Rly] → Program block → calculate_volume_tank → )



poznámka:

Formát zobrazených hodnot lze změnit, pravým kliknutím a výběrem v kontextovém menu.