

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

Simatic do škol

TIA Portal Module 010-050

Zpracování analogových hodnot s SIMATIC S7-1200

Cooperates
with Education

Automation

SIEMENS

SCE tréninkové balíčky pro tyto manuály

- **SIMATIC S7-1200 AC/DC/RELAY 6er "TIA Portal"**
Oper 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](https://www.siemens.com/sce/tp)

Další možnosti

Pro regionální Siemens SCE trénink kontaktujte osobu z vašeho regionu:
[siemens.com/sce/contact](https://www.siemens.com/sce/contact)

Další informace o SCE

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

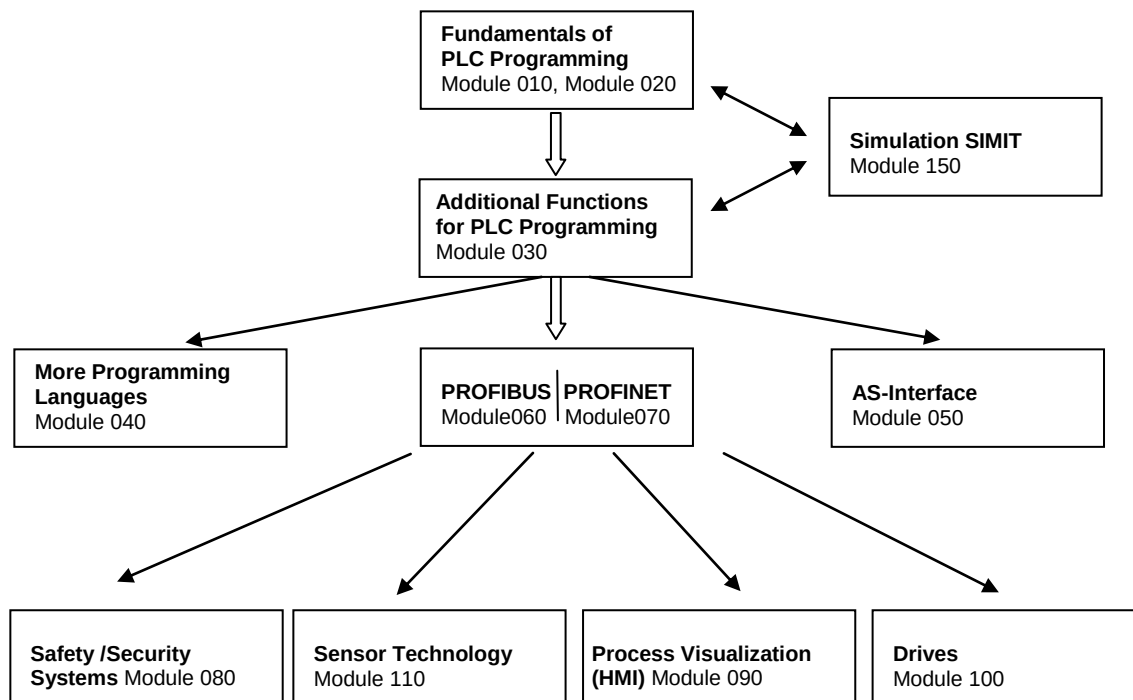
Děkujeme mnohokrát Michael Dziallas Engineering Corporation a všem ostáním zainteresovaným osobám za přípravu tohoto dokumentu.

Obsah

1. Úvod	4
2. Poznámky k programování SIMATIC S7-1200	6
2.1 Automatizační systém SIMATIC S7-1200	6
2.2 Programovací software STEP 7 professional 11 (TIA portal V11)	6
3. Analogové signály	7
4. Datové typy v SIMATIC S7-1200.....	8
5. Čtení a zápis analogových hodnot	9
5.1 Normalizace analogových hodnot.....	10
6. Vzorová úloha – sledování hladiny nádrže.....	10
6.1. Programování sledování hladiny pomocí SIMATIC S7-1200	11

1. Úvod

Vzhledem ke svému obsahu je modul SCE_CZ_010-020 součástí jednotky **‘Základy PLC programování’** a je rychlým vstupním bodem pro programování SIMATIC S7-1200 pomocí TIA portálu.



Cíle tréninku:

V modulu SCE_CZ_010-020, se čtenář seznámí s různými bloky používanými pro programování SIMATIC S7-1200 v nástroji TIA portal. Modul vysvětluje různé blokové typy, a ukazuje v krocích níže uvedených jak generovat program pomocí funkčních bloků.

- Generování funkčního bloku
- Definování interních proměnných
- Programování pomocí interních proměnných v bloku
- Volání a parametrizace funkčního bloku v OB1

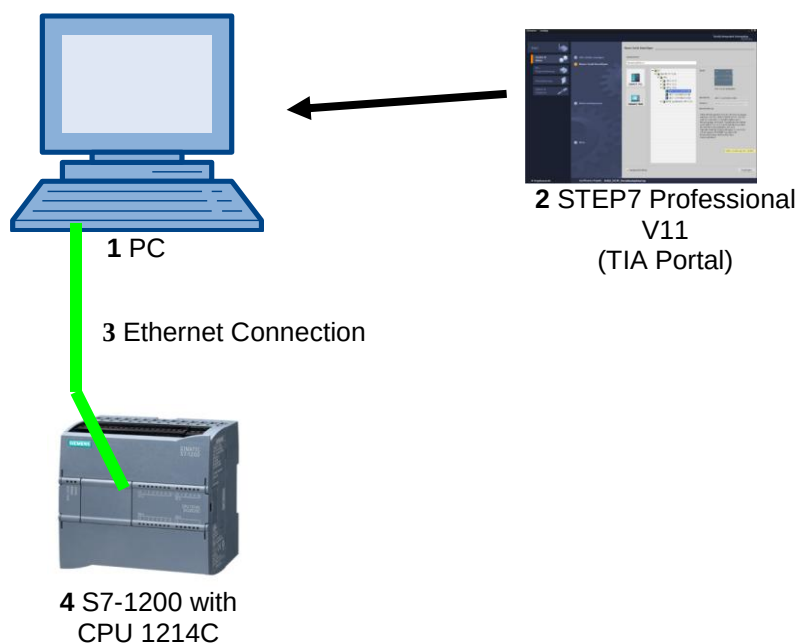
Prerekvizity:

Pro úspěšné absolvování této lekce je třeba znát:

- Práci s Windows operačním systémem
- Základy programování PLC s TIA portal

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 SP
- 2 Software STEP7 Professional V11 SP1 (Totally Integrated Automation (TIA) Portal V11)
- 3 Ethernet connection between PC and CPU 315F-2 PN/DP
- 4 PLC SIMATIC S7-1200; for example, CPU 1214C.
The inputs have to be brought out to a panel.



2. Poznámky k programování SIMATIC S7-1200

2.1 Automatizační systém SIMATIC S7-1200

Automatizační systém SIMATIC S7-1200 je modulární mikrokontrolér pro menší a střední automatizační úlohy.

Rozsáhlé spektrum modulů je k dostání pro optimální adaptaci na danou automatizační úlohu. Kontrolér S7 se stává ze zdrojového modulu, samotné řídicí jednotky CPU a vstupně/výstupních modulů pro digitální a analogové signály.

Pokud je třeba, komunikační procesor a funkční moduly je možné upravit pro speciální úlohy jako třeba řízení krokového motoru.

S S7 programem, programovatelný logický automat (PLC) sleduje a řídí zařízení nebo proces, kde IO moduly jsou zapsány v S7 programu jako %I pro vstupy a %Q pro výstupy.

Celý systém je programován softwarem STEP 7.

2.2 Programovací software STEP 7 Professional 11 (TIA portal V11)

Software STEP 7 Professional V11 (TIA Portal V11) je programovací nástroj pro následující automatizační systémy.

- SIMATIC S7-1200
- SIMATIC S7-300
- SIMATIC S7-400
- SIMATIC WinAC

Se STEP 7 Professional V11, mohou být navrženy následující funkce pro automatizaci provozu:

- Konfigurace a parametrizace hardwaru
- Definování komunikace
- Programování
- Testování, prověření a servis s provozními/diagnostickými funkcemi
- Dokumentace
- Vytváření displejů pro SIMATIC basic panely s integrovaným WinCC Basic
- S doplňkovými WinCC balíčky, můžete připravit zobrazovací řešení pro PCs a ostatní panely

Všechny funkce jsou připraveny spolu s detailní online nápovědou.

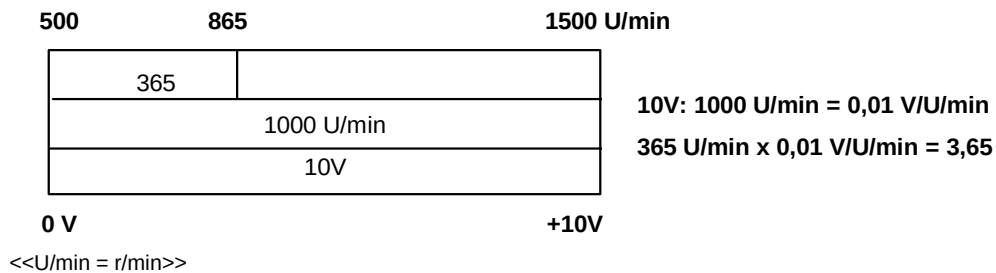
3. Analogové signály

Oproti binárním signálům, které můžeme považovat jen za dvoustavové "Napětí přítomno +24V" a "Napětí nepřítomno 0V", analogové signály v určitém rozsahu můžeme považovat za jakoukoliv číselnou hodnotu. Typickým příkladem analogového signálu je potenciometr. Záleží na pozici otáčejícího se tlačítka, a jím nastavené hodnotě jakéhokoliv odporu až do maxima.

Níže jsou příklady analogových hodnot vyskytujících se v řídicí technice:

- teplota -50... +150°C
- tok 0... 200l/min
- rychlost 500... 1500 r/min
- apod.

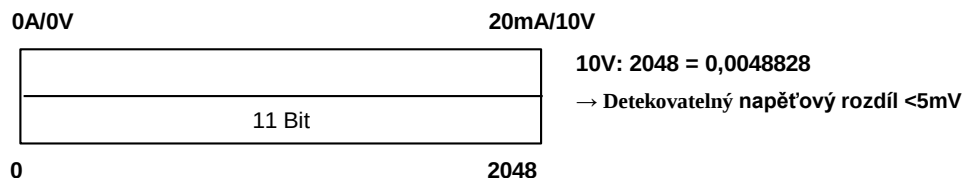
S využitím převodníku, můžeme převést tyto proměnné na napětí, proud nebo odpor. Pokud například chceme zaznamenávat rychlost, můžeme rozsah od 500 do 1500 r/min převést pomocí napěťového převodníku na napětí o hodnotě od 0 do +10V. Pokud je rychlost 865 rpm převodník by měl dávat napětí + 3.65 V.



Tyto elektrické veličiny jsou připojeny na analogový modul, který digitalizuje signál.

Když je analogová hodnota zpracována s PLC, hodnoty napětí, proudu a odporu které jdou do PLC musí být převedeny na hodnoty digitální. Tento převod se nazývá Analog/Digital převod (A/D převodník). To znamená, například, že napětí 3,65V je uloženo jako v sérii binárních čísel jako informace. Čím více číslic využijeme pro digitální reprezentaci, tím lepší je rozlišení. Pokud, by kupříkladu, byl využit jen jeden bit pro napětí od 0 do +10V, zjistili bychom jen, jestli je napětí v hodnotě mezi 0 a +5V nebo mezi +5V a +10V. Se dvěma bity můžeme rozsah rozdělit na 4 individuální rozsahy; to je 0... 2.5/2.5... 5/5... 7.5/7.5... 10V. A/D převodníky využívají standardně v řídicí technice 8 nebo 11 bitů.

S 8 bity máme k dispozici až 256 individuálních rozsahů a s 11bity až 2048 rozsahů.



4. Datové typy v SIMATIC S7-1200

SIMATIC S7-1200 poskytuje široké spektrum datových typů, které můžeme využít pro reprezentaci různých číselných formátů. Seznam níže ukazuje základní datové typy.

Datový typ	Velikost (Bit)	Rozsah	Příklad vstupní konstanty
Bool	1	0 to 1	TRUE, FALSE, 0,1
Byte	8	16#00 to 16#FF	16#12, 16#AB
Word	16	16#0000 to 16#FFFF	16#ABCD, 16#0001
DWord	32	16#00000000 to 16#FFFFFFFF	16#02468ACE
Char	8	16#00 to 16#FF	'A', 't', '@'
Sint	8	-128 to 127	123, -123
Int	16	-32768 to 32767	123, -123
Dint	32	-2.147.483.648 to 2.147.483.647	123. -123
USInt	8	0 to 255	123
UInt	16	0 to 65.535	123
UDInt	32	0 to 4.294.967.295	123
Real	32	+/-1.18 x 10 ⁻³⁸ to +/-3.40 x 10 ³⁸	123,456, -3.4, -1.2E+12, 3.4E3
LReal	64	+/-2.23 x 10 ⁻³⁰⁸ to +/-1.79 x 10 ³⁰⁸	12345.123456789 -1.2E+40
Time	32	T#-24d_20h_31m_23s_648ms_ to T#-24d_20h_31m_23s_647ms Stored as -2.147.483.648 ms to +2.147.483.647 ms	T#5m_30s 5#-2d T#1d_2h_15m_30x_45ms
String	Variable	0 to 254 characters in byte size	'ABC'

Poznámka:

Pro zpracování analogových hodnot hrají důležitou roli datové typy **'INT'** a **'REAL'** protože vložená analogová hodnota je reprezentována jako **'INT'** a pro pozdější přesnější zpracování, máme proměnnou s pohyblivou řádovou čárkou **'REAL'** kvůli zaokrouhlovací chybě v případě **'INT'**.

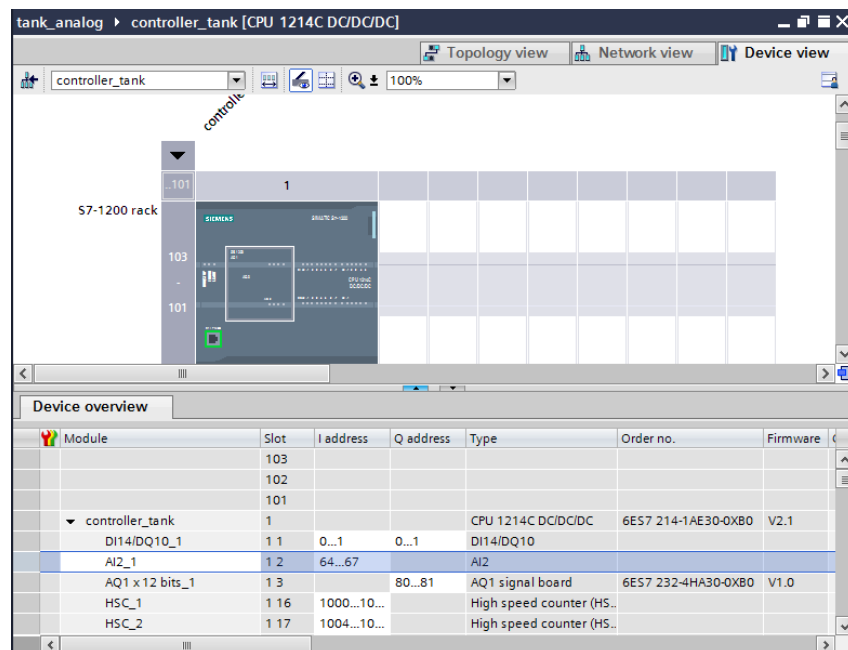
5. Čtení a zápis analogových hodnot

Analogové hodnoty jsou čteny/zapisovány v PLC jako informace typu Word. Tyto Word můžeme získat pomocí operandů

%IW 64 Analogový vstup Word 64
 %QW 80 Analogový výstup Word 80
 Pro příklad.

Každé analogové hodnotě „kanálu“ je přiřazeno jedno vstupní a jedno výstupní slovo. Formát je **Int**.

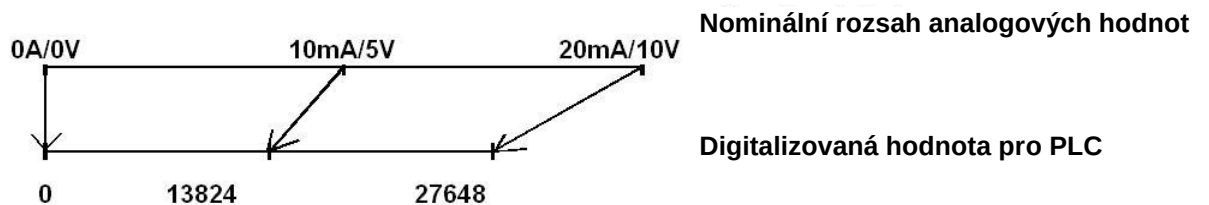
Adresování vstupů/výstupů záleží na adresování v přehledu zařízení. Kupříkladu:



Module	Slot	I address	Q address	Type	Order no.	Firmware
controller_tank	1			CPU 1214C DC/DC/DC	6ES7 214-1AE30-0XB0	V2.1
DI14/DQ10_1	2	0...1	0...1	DI14/DQ10		
AI2_1	3	64...67	80...81	AI2		
AQ1 x 12 bits_1	4			AQ1 signal board	6ES7 232-4HA30-0XB0	V1.0
HSC_1	5	1000...10...		High speed counter (HS..)		
HSC_2	6	1004...10...		High speed counter (HS..)		

Adresa prvního analogového vstupu je %IW 64, adresa druhého analogového vstupu je %IW 66, a adresa analogového výstupu je %QW 80.

Transformace analogových hodnot v PLC je pro vstupy a výstupy je stejná. Digitalizace hodnoty vypadá následně:



Často, je třeba tyto digitalizované hodnoty normalizovat stejně pro další zpracování v PLC.

5.1 Normalizace analogových hodnot

Pokud je analogová hodnota prezentována jako digitální, musí být běžně normalizována, aby číselná hodnota odpovídala fyzikální hodnotě v procesu.

Stejně tak analogový výstup do IO výstupního slova běžně přichází na řadu až po normalizaci výstupní hodnoty.

V programech ve STEP 7 jsou pro normalizaci využity výpočetní operace. Pro to aby bylo možné dosáhnout co největší přesnosti, musí být hodnota převedena na typ REAL pro snížení zaokrouhlovací chyby na minimum.

V kapitole níže, je ukázán příklad s využitím sledování hladiny nádrže.

6. Vzorová úloha – sledování hladiny nádrže

Naprogramujeme sledování hladiny nádrže.

Senzor převádí výšku hladiny v nádrži na napětí o rozsahu od 0 do 10V.
0V odpovídá hladině o 100 litrech a 10V hladině o 1000 litrech.

Senzor je připojený na analogový vstup SIMATIC S7-1200.
Nyní, je signál přiveden do funkce FC1 a normalizován.

Dále, naprogramujeme následující: monitorování a zobrazování maximální přípustné hladiny 990 litrů a monitorování minimální přípustné hladiny 110 litrů.

Seznam úloh:

Adresa	Symbol	Datový typ	Komentář
%IW 64	AI_level_tank1	Int	Analogový vstup hladina Tank1
%Q 0.0	Tank1_max	Bool	Zobrazení hladina > 990 liters
%Q 0.1	Tank1_min	Bool	Zobrazení hladina < 110 liters

6.1. Programování sledování hladiny pomocí SIMATIC S7-1200

Projekt je spravován pomocí software '**Totally Integrated Automation Portal**'.

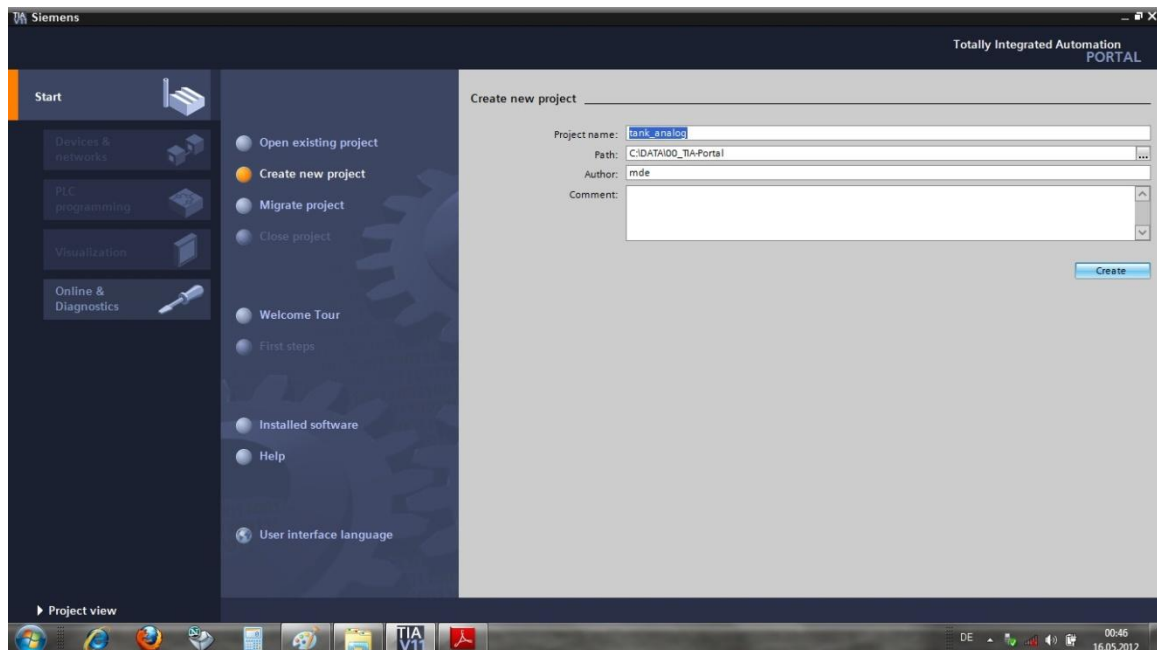
Zde pod jednotným rozhraním, jsou komponenty jako řídicí systém, vizualizace a sítě nastaveny, parametrizovány a programovány. Pro diagnostiku chyb jsou k dispozici online nástroje.

V krocích níže, si připravíme projekt pro SIMATIC S7-1200 a naprogramujeme v něm naše řešení.

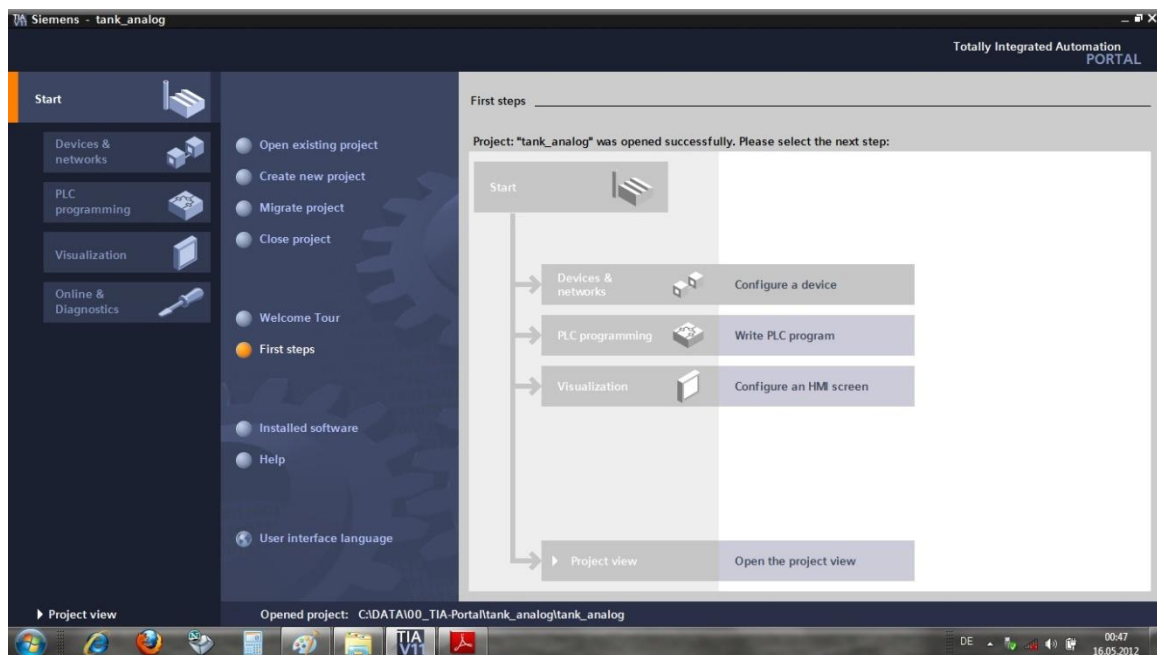
1. Centrálním nástrojem je '**Totally Integrated Automation Portal**'. Spustíme ho dvojklikem. (→ Totally Integrated Automation Portal V11)



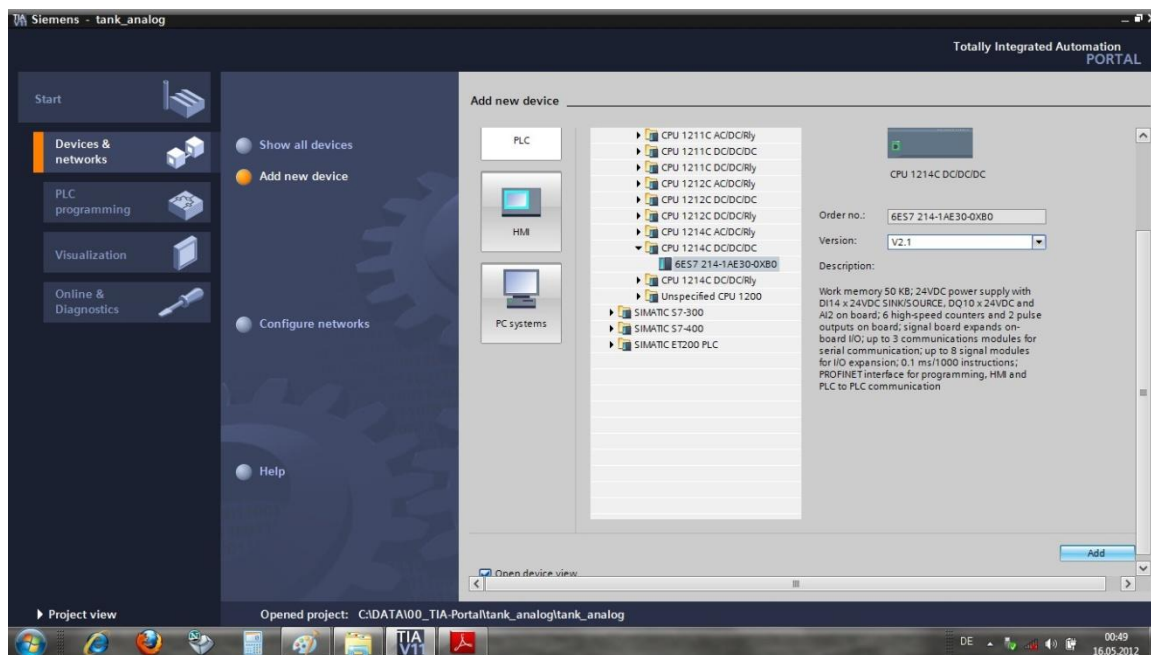
2. Programy pro SIMATIC S7-1200 jsou spravovány v projektech, takový projekt je nastaven v portal view. (→ Create new project → Tank_Analog → Create)



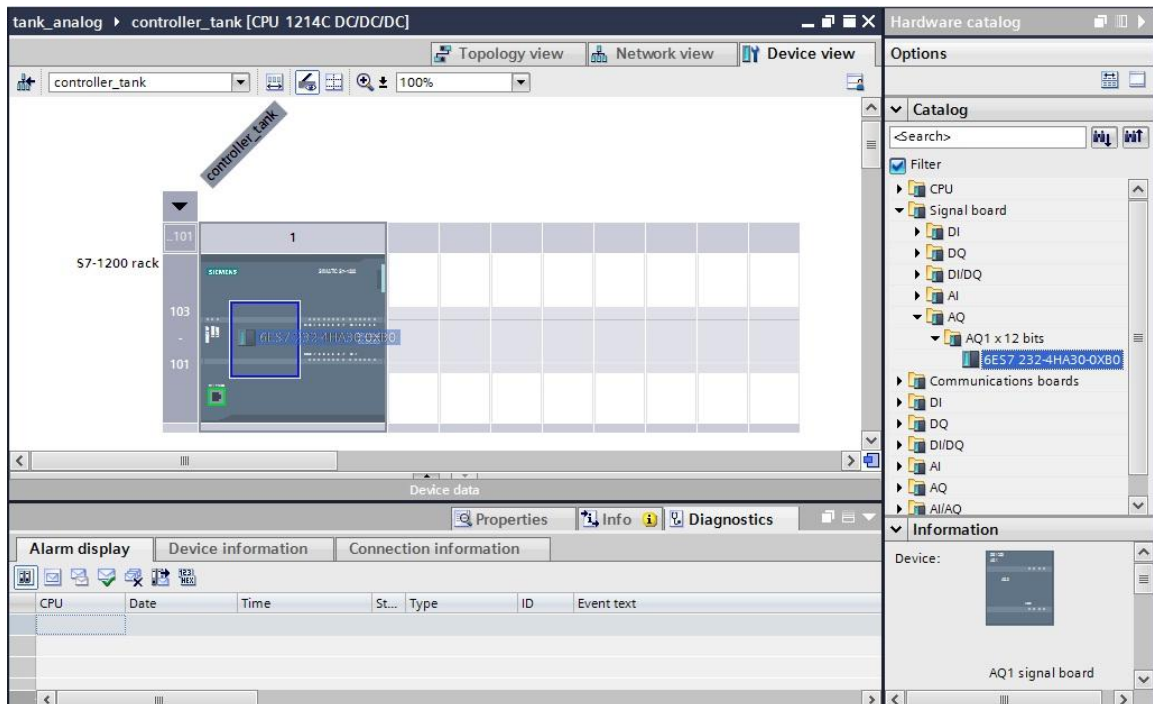
3. Nyní, '**First Steps**' první kroky pro konfiguraci. Nejdříve chceme '**Configure a device**'. (→ First steps → Configure a device)



4. Poté přidáme zařízení 'Add new device' se jménem "controller_tank". Z katalogu vybereme 'CPU1214C' se shodným objednáčím číslem. (→ Add new device → controller_tank → CPU1214C → 6ES7 → Add)



5. Software se nyní sám přepne do Project view a otevře hardware konfiguraci. Teď můžeme přidat z hardware katalogu další moduly a karty. (→ Catalog → Signal board → AO1 x 12Bit → 6ES7 232-...))



6. V '**Device view**', můžeme nastavit adresy vstupů a výstupů nebo je resetovat. Integrované analogové výstupy CPU, mají adresy od %IW64 do %IW66 a integrované digitální výstupy adresy od %Q0.0 do %Q1.1

Analogový výstup signálové karty má adresu %QW80

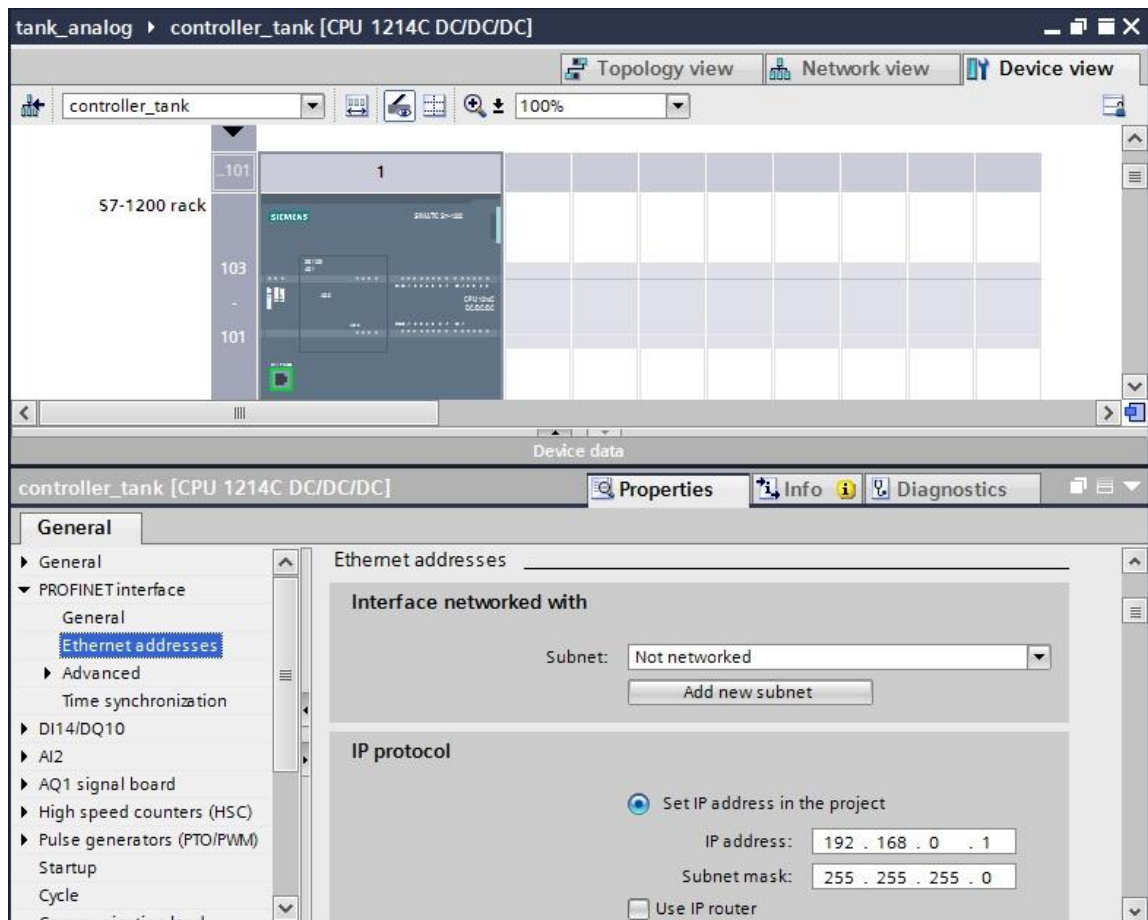
The screenshot displays the 'Device view' of a SIMATIC S7-1200 rack. The rack is labeled 'S7-1200 rack' and contains the following modules:

- Slot 1: CPU 1214C DC/DC/DC
- Slot 2: DI14/DQ10_1
- Slot 3: AI2_1
- Slot 16: HSC_1
- Slot 17: HSC_2

The 'Device overview' table provides detailed information for each module:

Module	Slot	I address	Q address	Type	Order no.	Firmware
controller_tank	1			CPU 1214C DC/DC/DC	6ES7 214-1AE30-0XB0	V2.1
DI14/DQ10_1	1 1	0...1	0...1	DI14/DQ10		
AI2_1	1 2	64...67		AI2		
AQ1 x 12 bits_1	1 3		80...81	AQ1 signal board	6ES7 232-4HA30-0XB0	V1.0
HSC_1	1 16	1000...10...		High speed counter (HS..		
HSC_2	1 17	1004...10...		High speed counter (HS..		

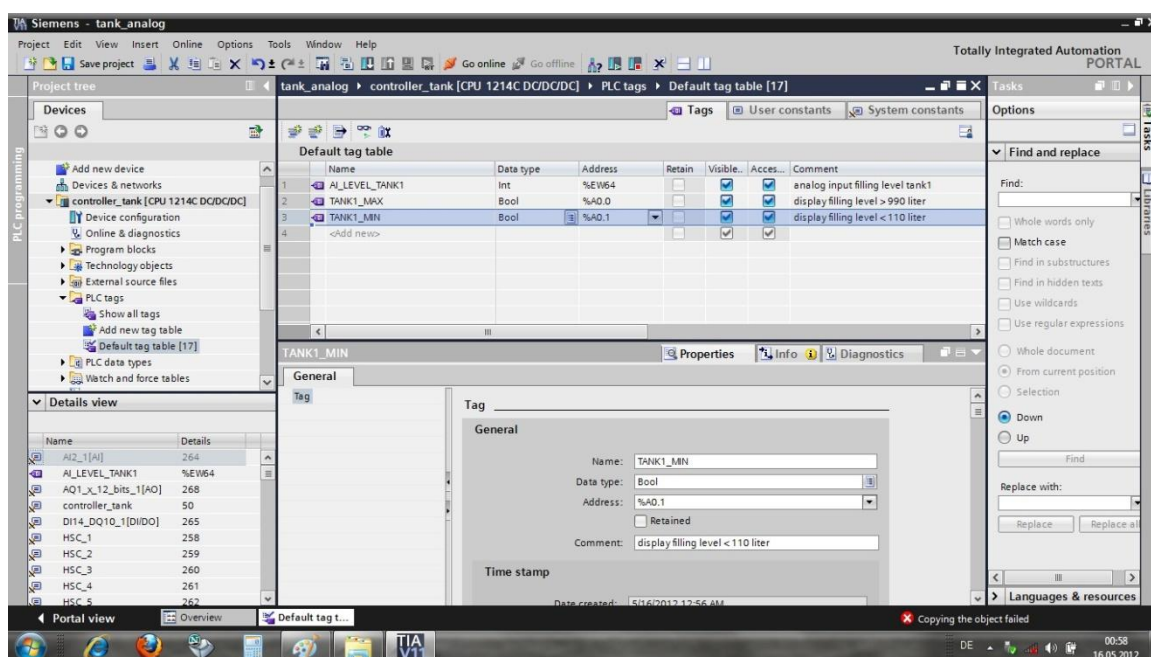
7. Aby později software přistoupil na správné CPU, musíme nastavit správnou IP adresu a masku podsítě. (→ Properties → General → PROFINET interface → Ethernet addresses → IP address: 192.168.0.1 → subnet mask: 255.255.255.0)



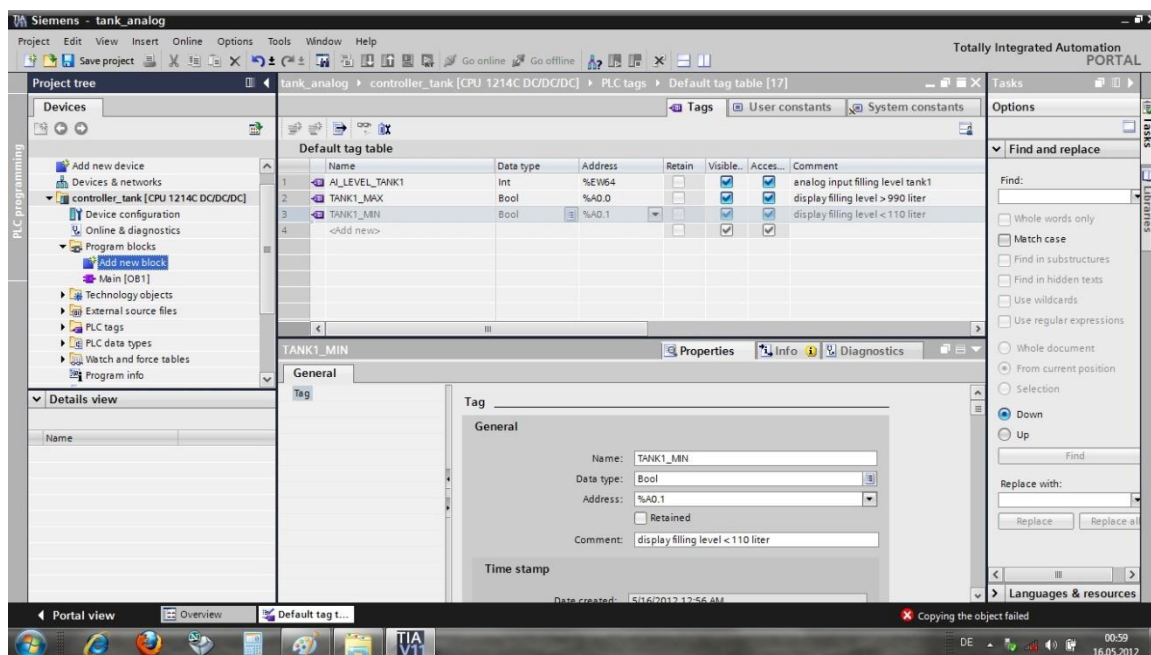
8. Protože moderní programování se nezabývá absolutním adresováním ale tagy, musíme specifikovat **globální PLC tagy**.

Tyto globální tagy mají popisná jména s komentáři pro vstupy a výstupy, které využíváme v programu. Později v programu na tyto tagy můžeme přistoupit pouze s využitím jejich jmen. Globální tagy můžeme využít skrz celý program ve všech blocích.

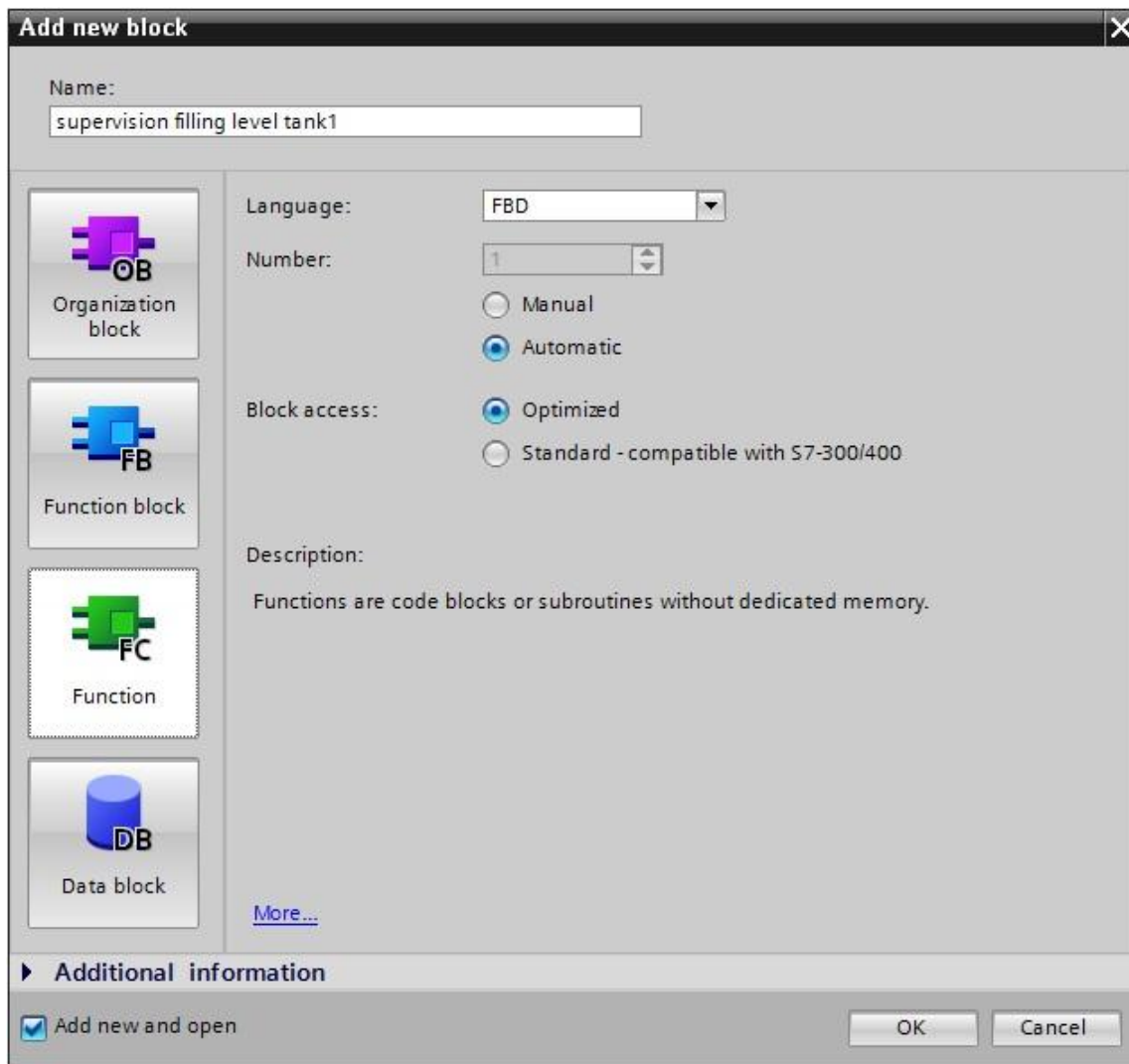
Vybereme v projektové navigaci '**controller_tank [CPU1214C DC/DC/DC]**' a poté '**PLC tags**'. Otevřeme tabulku '**PLC tags**' a zadáme jména vstupů a výstupů jak je ukázáno níže. (→ controller_tank[CPU1214C DC/DC/DC] → PLC tags→ Default tag table)



9. Pro vytvoření funkčního bloku FC1, vybereme v projektové navigaci **controller_tank [CPU1214C DC/DC/DC]** a pak **'Program blocks'**. Dvojklik na **'Add new block'**. (→ controller_tank[CPU1214C DC/DC/DC] → Program block → Add new block)



10. Vybereme '**Function(FC)**' a přiřadíme mu jméno '**supervision filling level tank1**'. Jako programovací jazyk vybereme, 'FBD' (function block diagram). Číslování je automatické. Protože tento FC1, voláme později symbolickým jménem, není pro nás číslo nadále relevantní. Potvrdíme vstupy pomocí '**OK**'. (→ Function (FC1) → supervision filling level tank1 → FBD → OK)



11. Blok '**supervision filling level tank1**' [FC1] se otevře automaticky. Před napsáním programu, musíme nejdříve deklarovat rozhraní bloku.

Když je deklarované rozhraní, specifikujeme lokální proměnné v bloku.

Proměnné jsou ze dvou skupin:

- Blokové parametry, které jsou rozhraním bloku pro volání z programu.

Typ	Jméno	Funkce	Dostupné v
Input parameter	Input	Parametr, jehož hodnoty blok čte.	Functions, function blocks and some types of organization blocks
Output parameter	Output	Parametr, jehož hodnoty blok zapisuje	Functions and function blocks
InOut parameter	InOut	Parametr, jehož hodnoty blok čte když je volán a po zpracování zapisuje zpět do stejného parametru.	Functions and function blocks

- Lokální data používané pro ukládání mezivýsledků.

Typ	Jméno	Funkce	Dostupné v
Temporary local data	Temp	Proměnná pro ukládání mezivýsledků, ty jsou uchovány jen po dobu jednoho cyklu.	Functions, function blocks and organization blocks
Static local data	Static	Proměnná použitá pro ukládání mezivýsledků, data jsou uchována dokud nejsou znovu přepsána.	Function blocks

12. Když deklarujeme lokální proměnné, potřebujeme následující proměnné pro náš příklad.

Vstup:

tank_level_AI Zde senzor zapíše analogovou hodnotu.

Výstup:

tank_max Zde je status maximálního zobrazení zapsán na výstup

tank_min Zde je status minimálního zobrazení zapsán na výstup

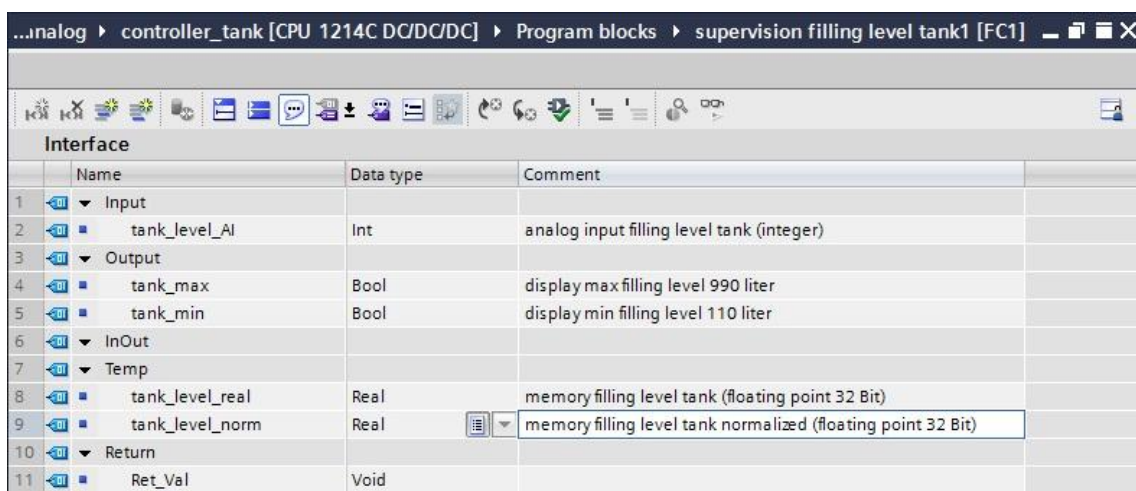
Dočasné:

tank_level_real Tato proměnná je potřeba pro uložení mezivýsledků

tank_level_norm Zde, je hodnota hladiny, normalizovaná ve formátu pohyblivé desetinné čárky, v rozsahu od 100 do 1000 litrů.

V tomto případě je obzvlášť důležité použít správné datové typy, jinak by nemuseli být kompatibilní s následujícím programem využívajícím konverzní funkce.

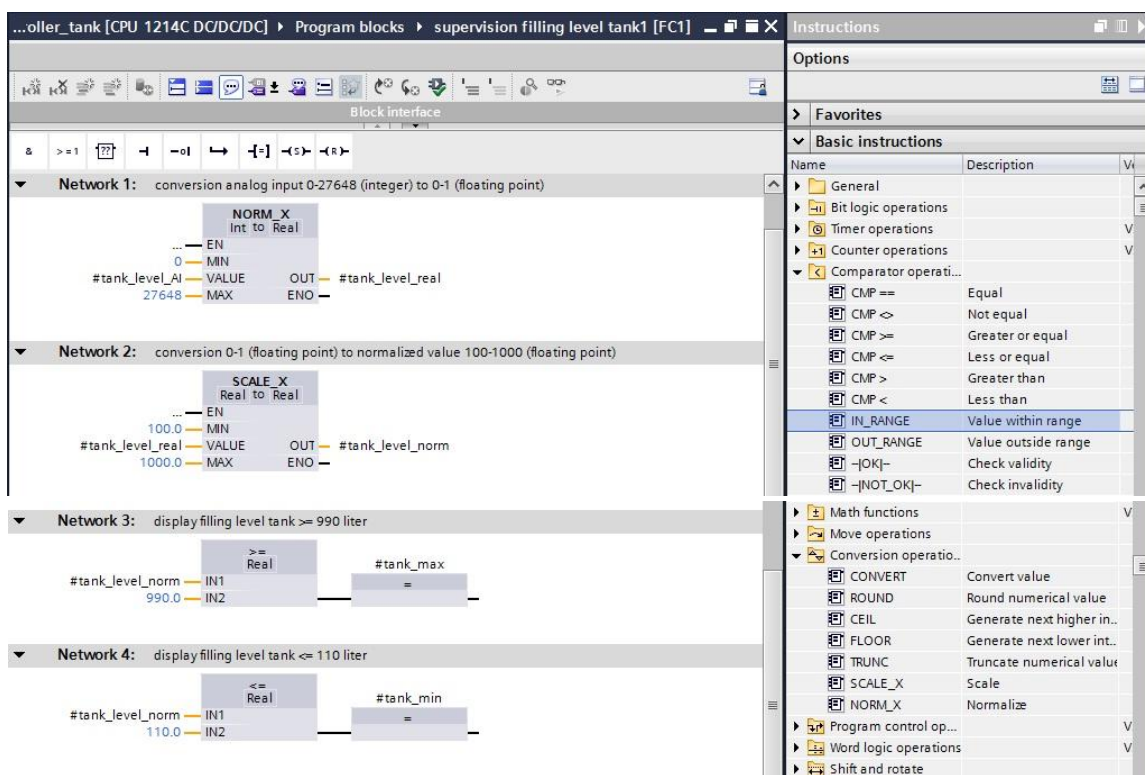
Pro přehlednost, by měli být všechny proměnné patřičně okomentovány.



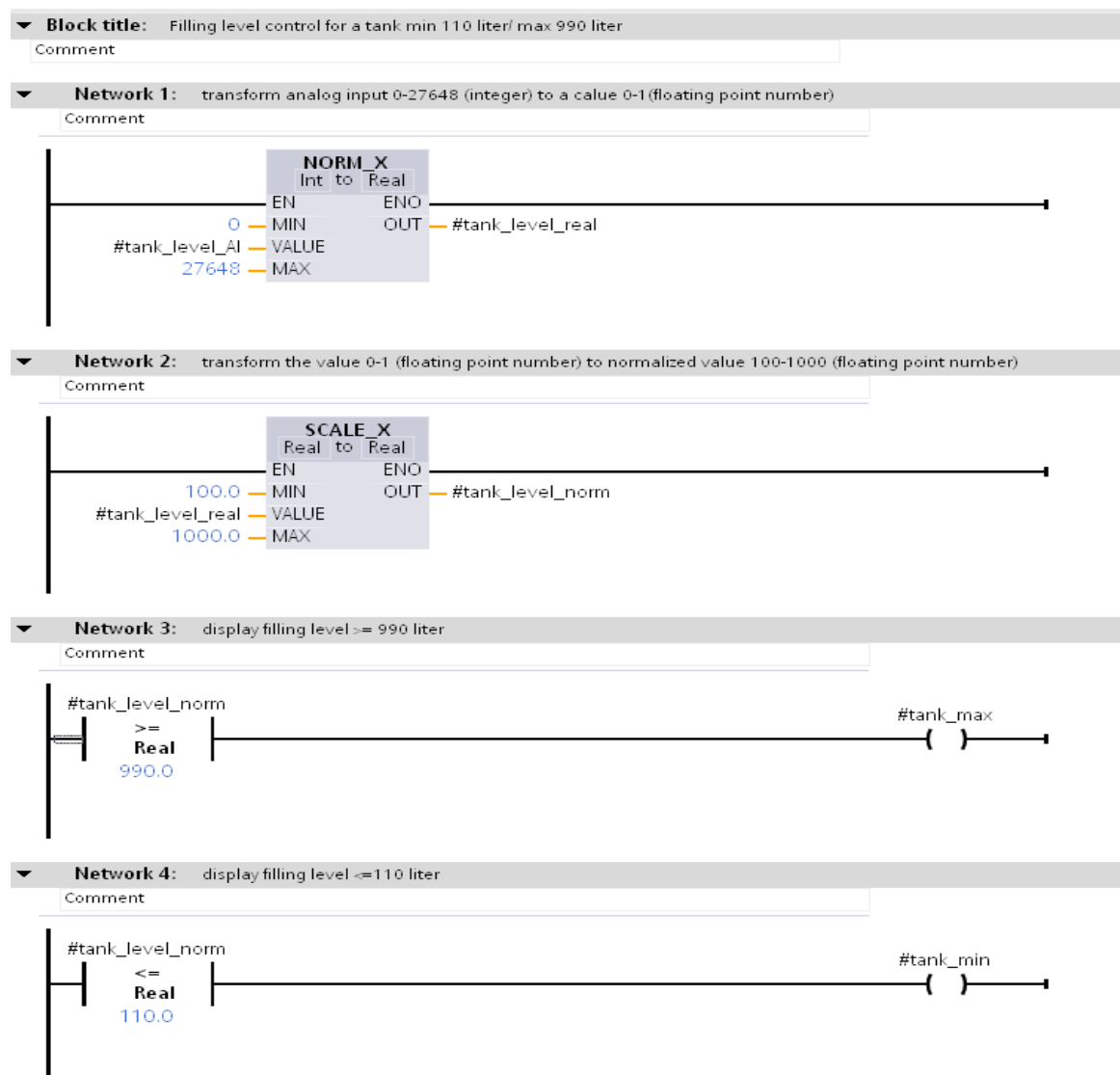
	Name	Data type	Comment
1	Input		
2	tank_level_AI	Int	analog input filling level tank (integer)
3	Output		
4	tank_max	Bool	display max filling level 990 liter
5	tank_min	Bool	display min filling level 110 liter
6	InOut		
7	Temp		
8	tank_level_real	Real	memory filling level tank (floating point 32 Bit)
9	tank_level_norm	Real	memory filling level tank normalized (floating point 32 Bit)
10	Return		
11	Ret_Val	Void	

13. Poté co deklarujeme lokální proměnné, program můžeme vytvořit jen pomocí jmen proměnných (identifikovaných symbolem '#'). Příklad v **FBD**, vypadá takhle:

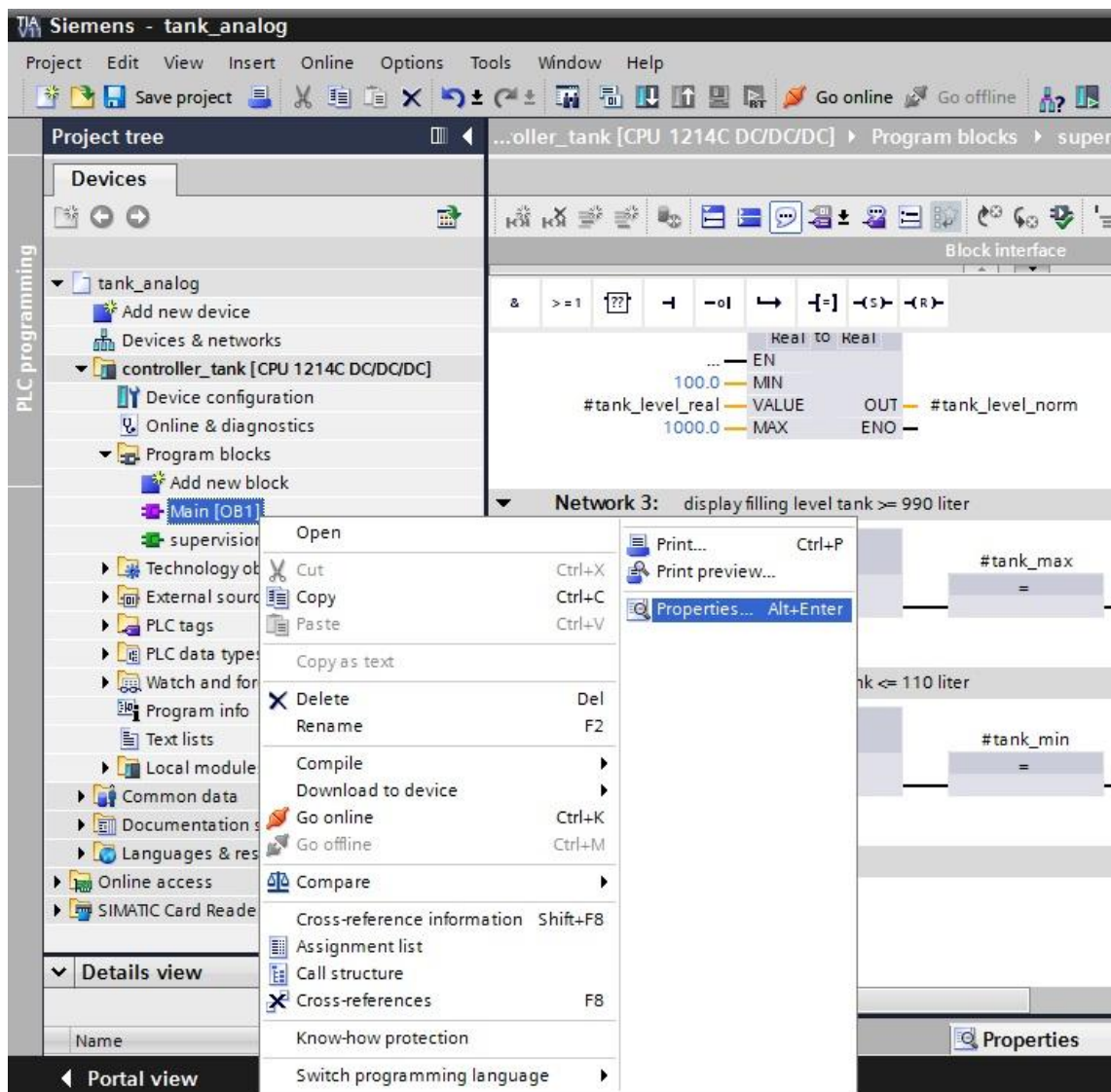
Program ve funkčním blokovém diagramu (FBD):



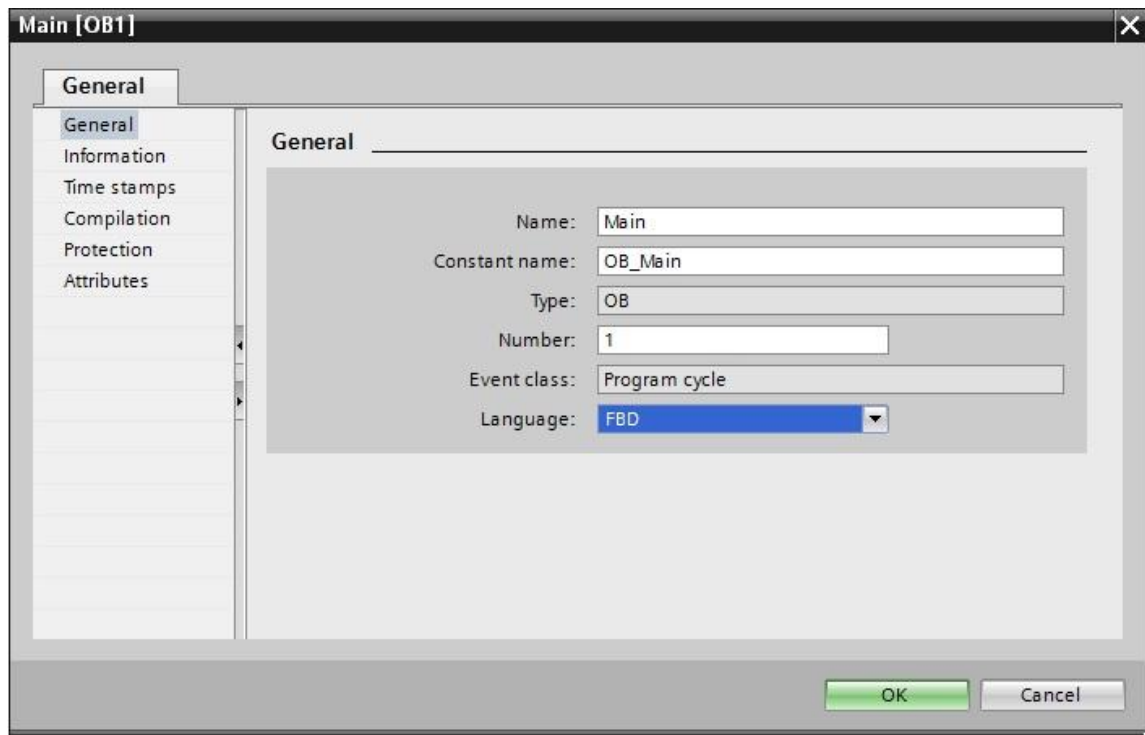
Program v žebříkovém diagramu (LAD):



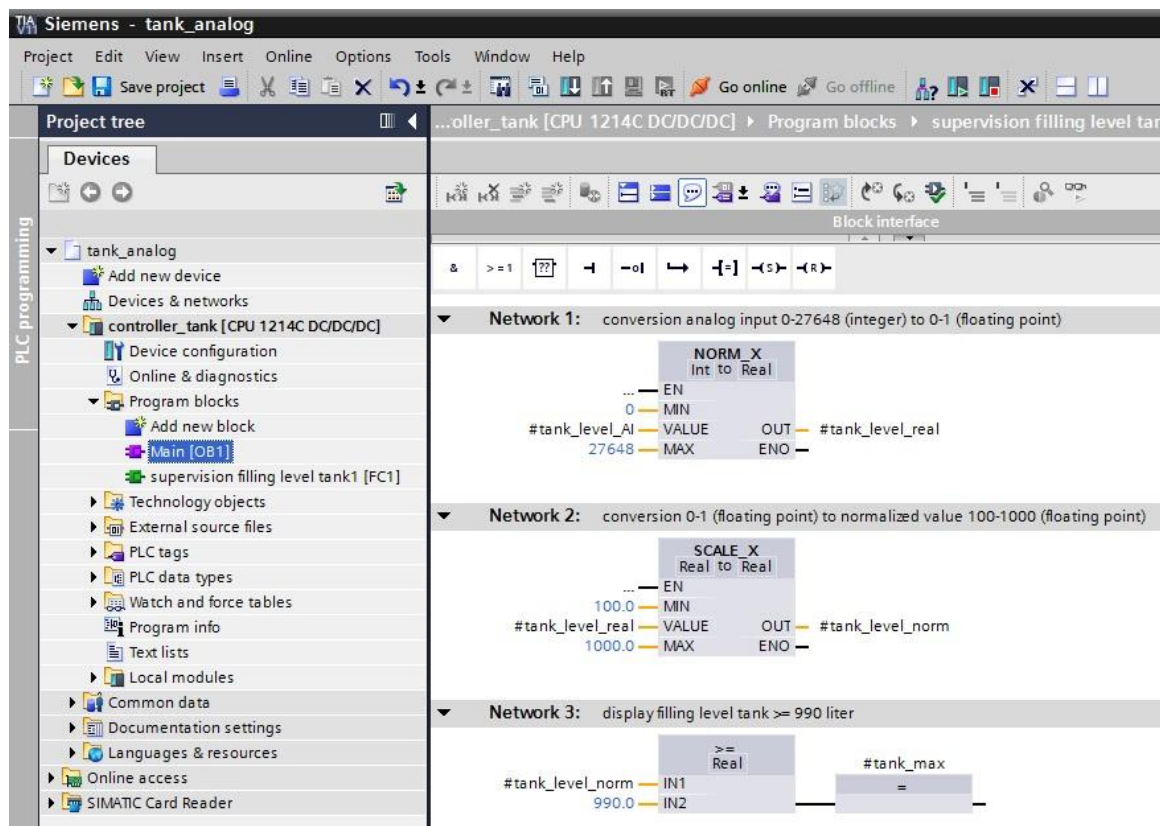
14. Poté otevřeme 'Properties' bloku 'Main[OB1]' který je prováděn v cyklech. Můžeme upravit vlastnosti bloku. (→ Properties → Main[OB1])



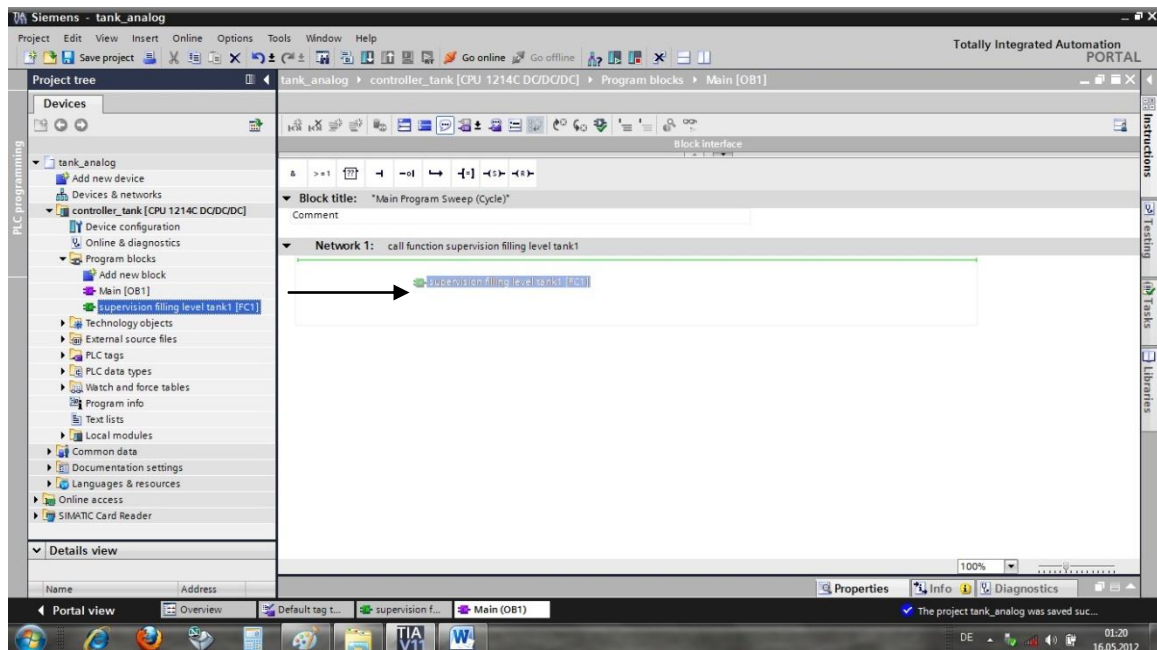
15. V nastavení vybereme programovací jazyk '**Language**' a zadáme funkční blokový diagram '**FBD**'.
(→ FBD → OK)



16. Nyní blok "supervision filling level tank1 [FC1]" musíme volat z bloku Main [OB1]. Jinak by blok nebyl proveden. Dvojklikem '**Main [OB1]**' otevřeme tento blok. (→ Main [OB1])

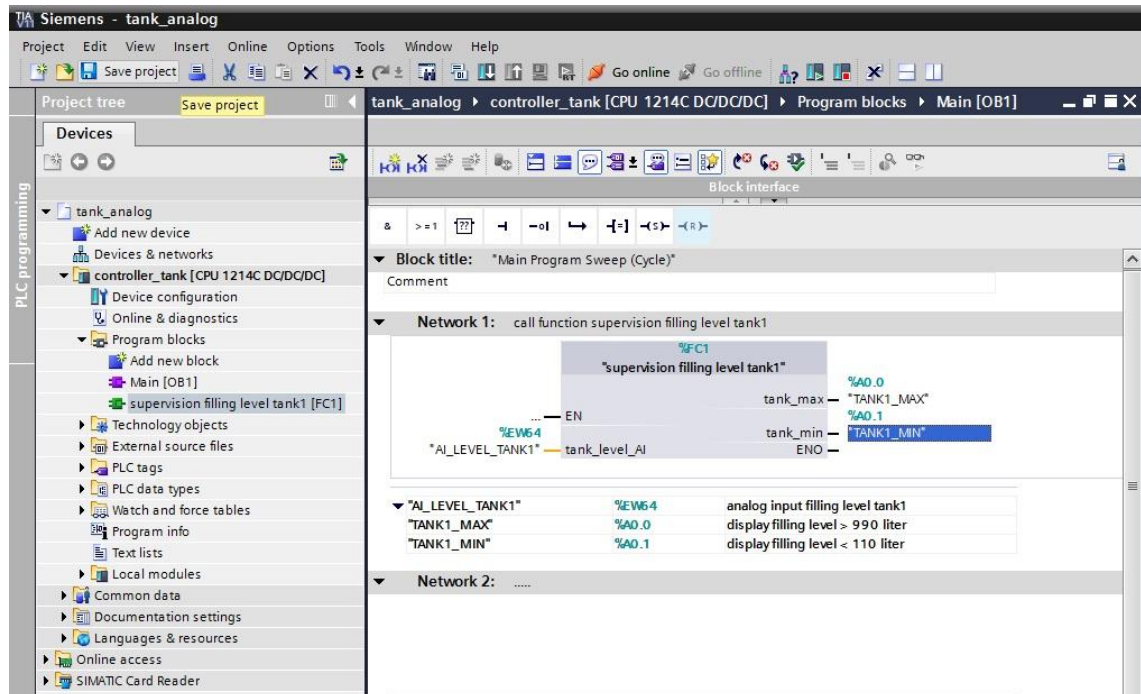


17. Pomocí drag&drop, blok **"supervision filling level tank1 [FC1]"** přetáhneme do Network 1 bloku Main [OB1]. (→ supervision filling level_tank1 [FC1])

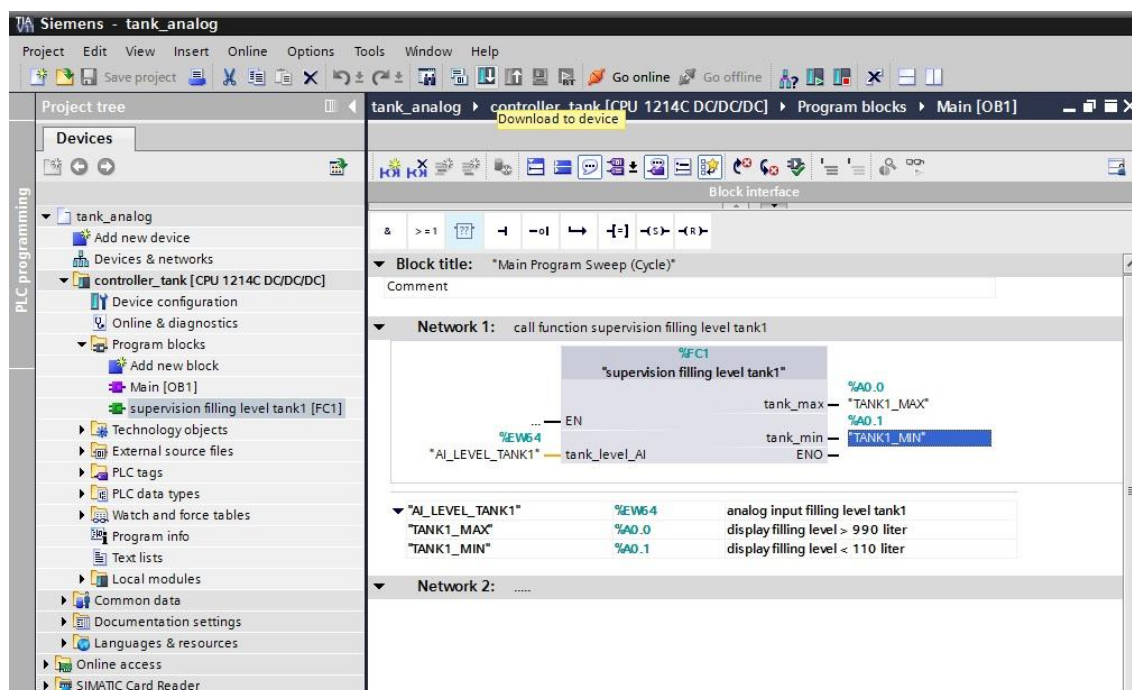


18. Poté zkontrolujeme, jestli jsou připojeny vstupy a výstupy v OB1 pomocí PLC tagů jak je ukázáno.

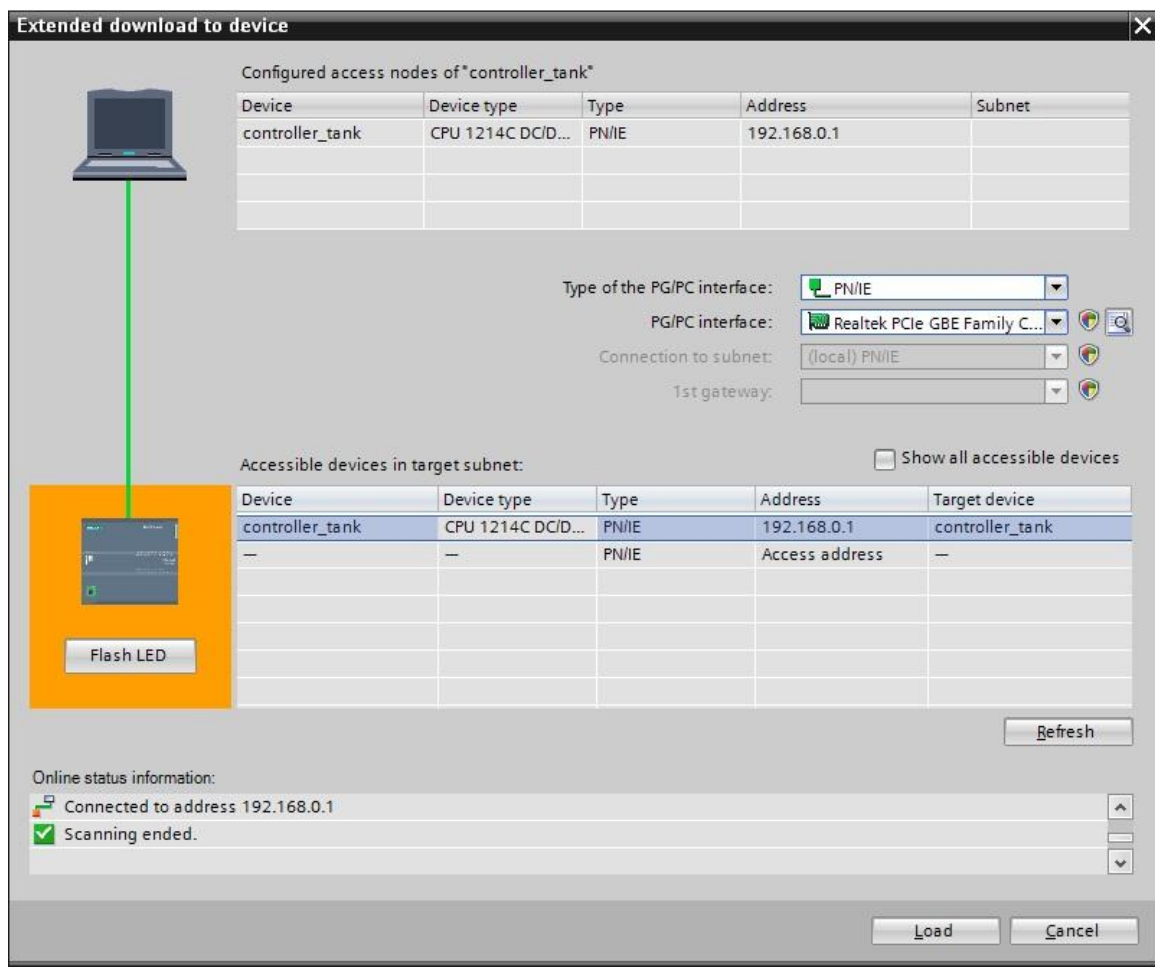
Kliknutím na  Save project uložíme projekt. (→ „AI_LEVEL_TANK1“ → "TANK1_MAX" → "TANK1_MIN" → )



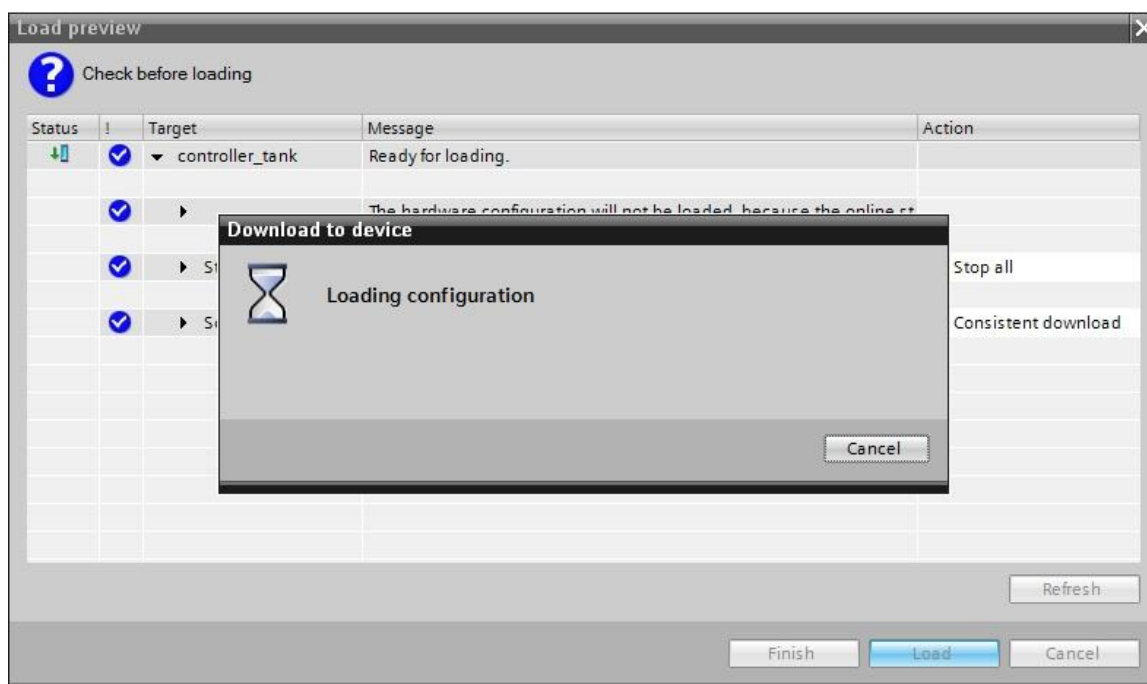
19. Pro nahrání celého programu do CPU, nejdříve zvýrazníme složku '**controller_tank**' pak klik na symbol  nahrání do zařízení. (→ controller_tank → )



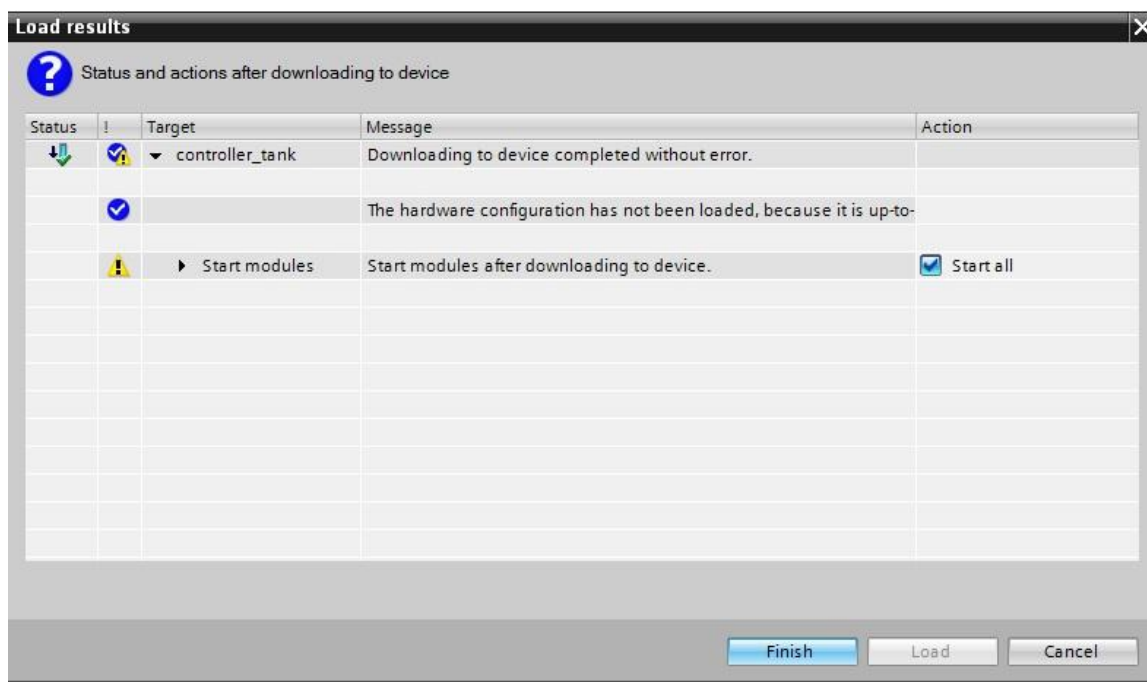
20. Pokud není rozhraní PG/PC specifikováno, zobrazí se okno ve kterém je možné tak udělat. (→ PG/PC interface for loading → load)



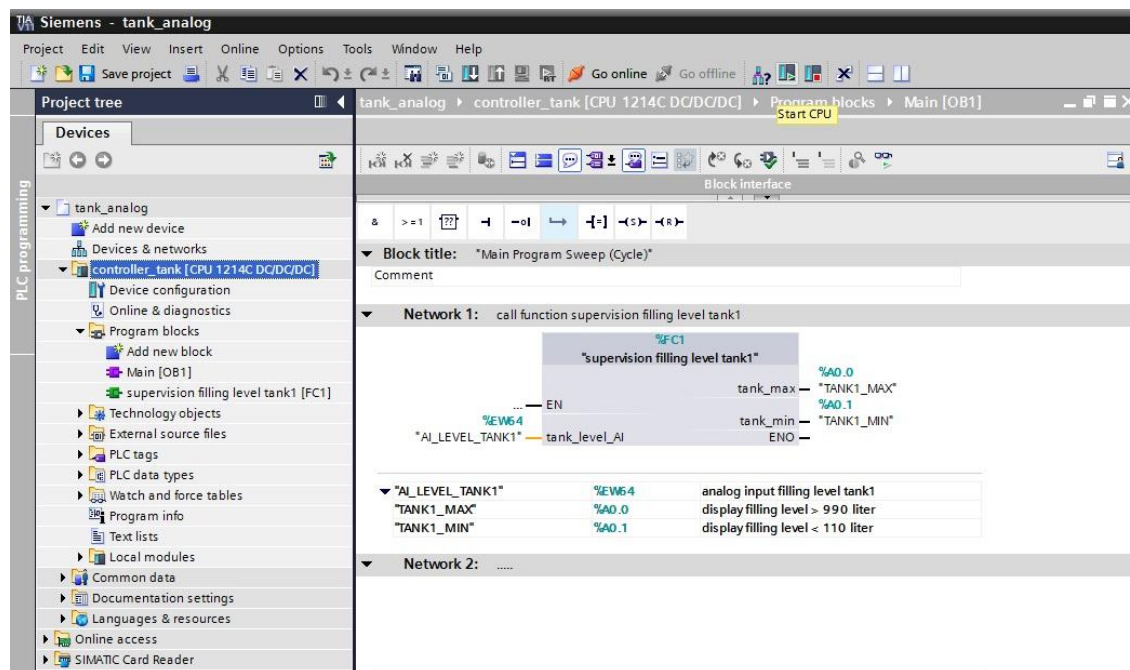
21. Klikneme na 'Load' ještě jednou. Během nahrávání je zobrazen status v okně. (→ Load)



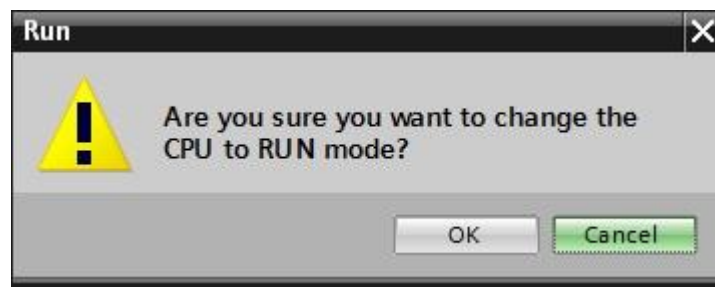
22. Úspěšné nahrání je zobrazeno v okně, nyní už jen klik na 'Finish'. (→ Finish)



23. Poté spustíme, CPU kliknutím na . (→ )



24. Potvrdíme, že chceme opravdu spustiť CPU kliknutím na **'OK'**. (→ OK)



25. Kliknutím na symbol  Monitoring on/off, můžeme sledovat status proměnných během testování programu. (→ supervision filling level tank1[FC1] → )

