SIEMENS

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

Simatic do škol

TIA Portal Module 010-030

IEC časovače a IEC čítače pro 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)“ specielně 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 využít komerčně.

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

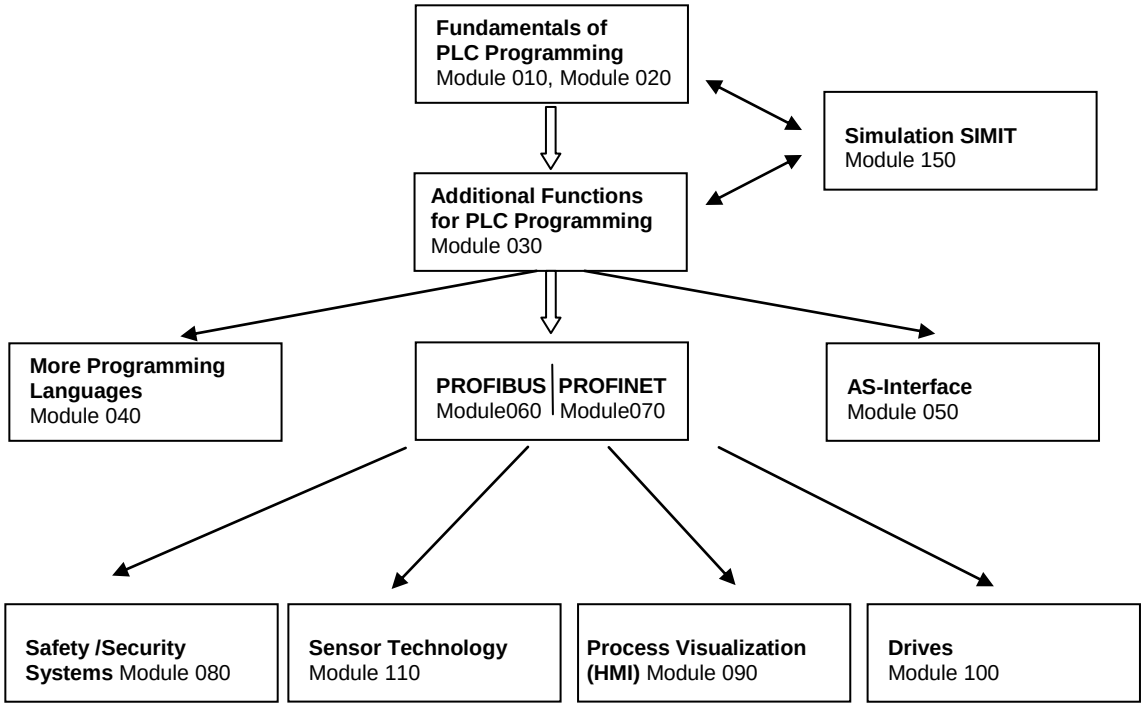
PAGE

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. Instance a multi-instance při programování SIMATIC S7-1200 7
   3.1 Instanční data bloky / jedno-instanční 7
   3.2 Multi-instance 9
4. Vzorová úloha: Kontrola lisu s časovačem a Instančním DB 11
5. Programování lisu s časovým zpožděním pro SIMATIC S7-1200 11
6. Vzorová úloha pro řízení pásového dopravníku s čítačem a Multi-instancí 29
7. Programování pásového dopravníku se SIMATIC S7-1200 30

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

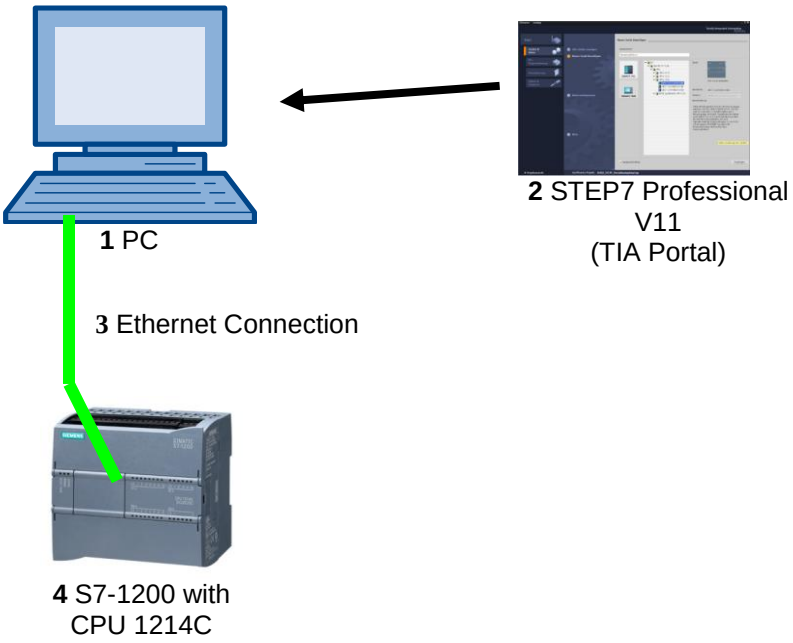
Prerekvizity:

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 system 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 řídící 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 system 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. Instance a multi-instance při programování SIMATIC S7-1200

Volání funkčního bloku nazýváme **instancí.** Při každém volání funkčního bloku je vytvořen **instanční data blok** používaný k ukládání dat. Zde jsou uloženy vlastní parametry a statická data.

Proměnné deklarovaná ve funkčním bloku předurčují strukturu instančního data bloku.

Použití jednoduché a multi-instancí:

Instanční data blok můžeme využít následně:

- Volání jako **jedno-instanční**:

  - Separátní instanční data blok pro každou instanci funkčního bloku.

- Volání jako **multi-instanci**:

  - Jeden instanční data blok pro několik instancí jednoho nebo více funkčních bloků.

3.1 Instanční data bloky / jedno-instanční

Volání funkčních bloků, které mají každý svůj instanční data blok se nazývá **jedno-instanční**.

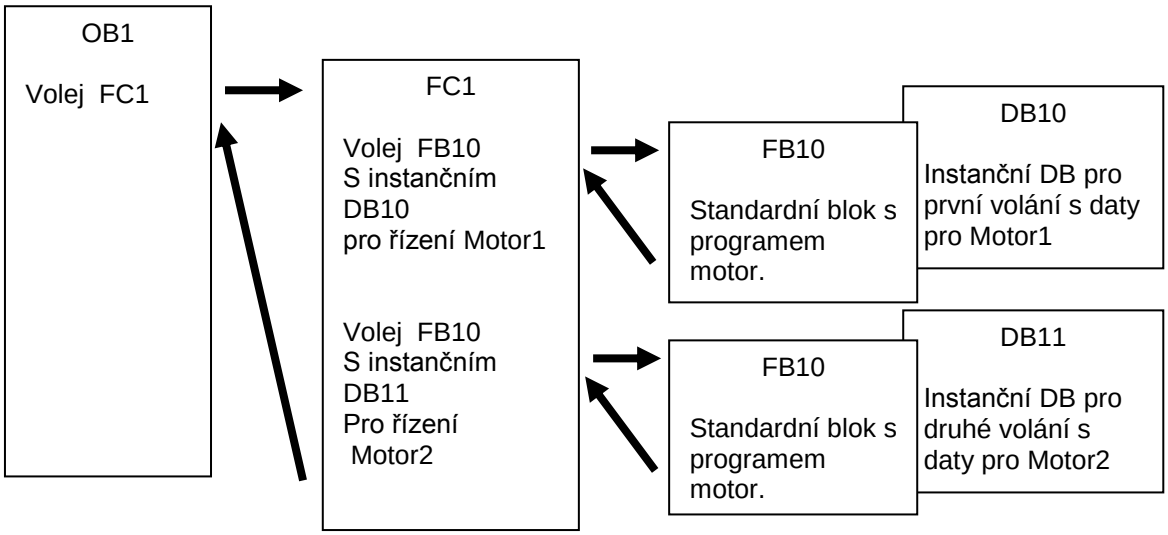
Pokud je funkční blok generován podle pravidel pro standardní blok (Modul 010-020), může být volán vícekrát.

Nicméně, pro každé jedno-instanční volání, musíte přiřadit jiný instanční data blok.

Příklad jedno-instancí:

Na náčrtu níže, jsou dva motory kontrolovány pomocí funkčního bloku FB10 a dvěma různými data bloky:

Různá data pro každý motor – například, rychlost, startovní čas, celkový čas běhu jsou uložena v různých instančních data blocích DB10 a DB11.

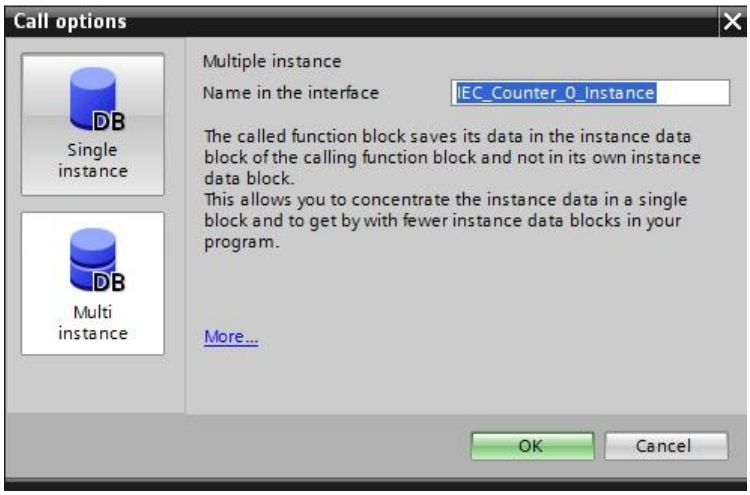


Poznámka:
Některé instrukce jako čítače a časovače se chovají jako funkční bloky. Pokud je voláme, také reprezentují instance a je třeba jim přidělit oblast paměti, například ve formě instančního data bloku.

3.2 Multi-instance

Kvůli kapacitě paměti, kterou může, CPU poskytnout, je možné, že budete chtít, nebo muset alokovat pouze omezený počet data bloků pro instanční data.

Pokud ve vašem uživatelském programu navíc už existují funkční bloky, časovače, čítače apod. volané ve funkčním bloku, je možné tyto bloky volat bez jejich vlastního instančního DB. Jednoduše zvolíme možnost **'Multi-Instance'**:



Poznámka:

Pro funkční bloky, které byly volané, multi-instance umožňuje uložit data do instančního data bloku pro blok, který teprve volán bude.

V tomto případě musí být volaný blok vždy funkční blok.

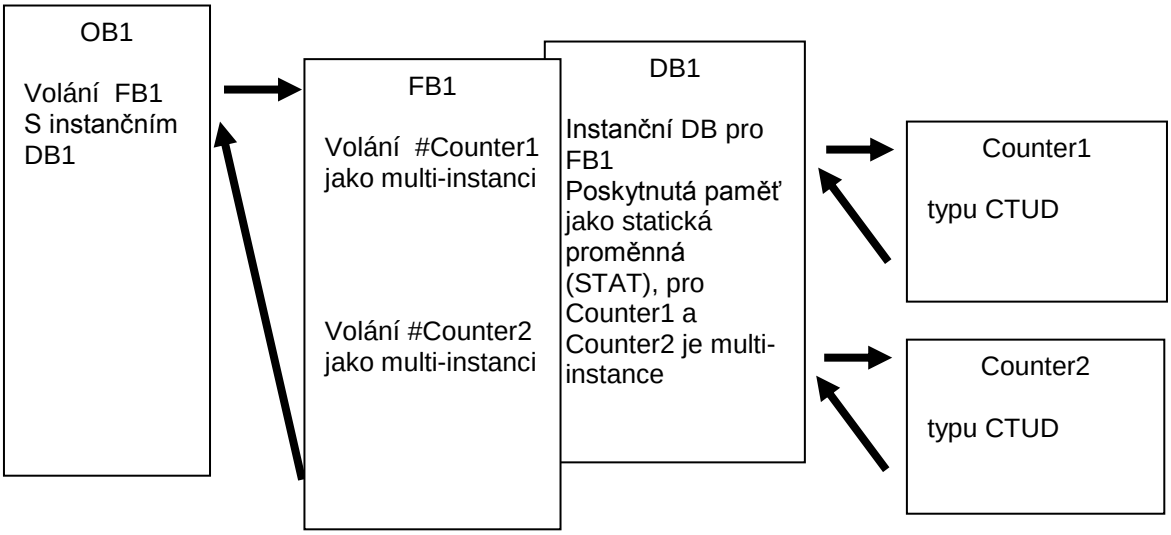
Touto cestou, můžete koncentrovat instanční data v jednom instančním data bloku, můžete tedy využít dostupný počet DB efektivněji.

Tuto cestu musíme využít vždy, když chceme, aby blok, který voláme, byl využit jako standardní blok.

Příklady pro Multi-instance:

Náčrtek níže ukazuje jak je čítač typu CTUP (čítej nahoru a dolů) volán dvakrát.

Různá data pro dva čítače jsou uložena v různých multi-instancí v instančním data bloku DB1 při volání funkčního bloku FB1.

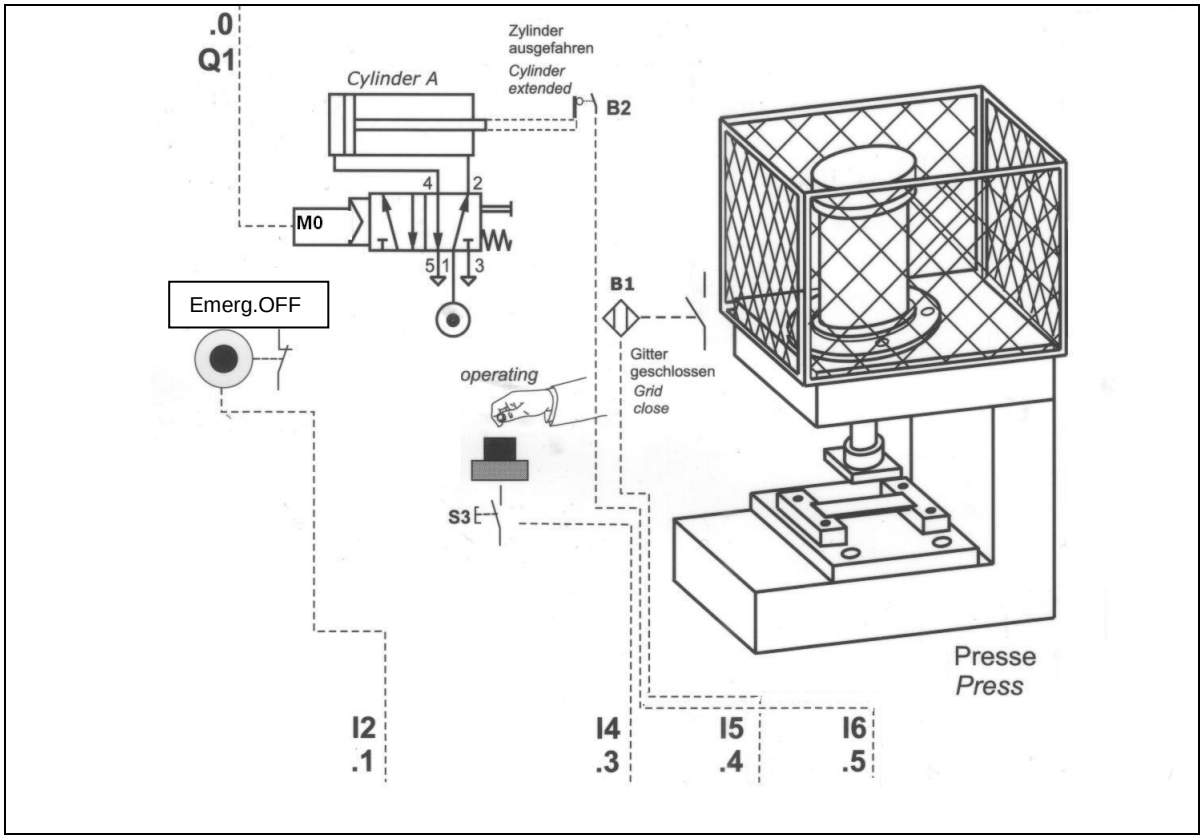


4. Vzorová úloha: Kontrola lisu s časovačem a Instančním DB

Úloha bude prováděná následně:
Lis s bezpečnostním plotem se spustí stiskem tlačítka S3 START pouze pokud je bezpečnostní plot uzavřený. Tento stav je monitorován senzorem Safety fence closed B1.
Pokud tomu tak je, ventil M0 otevře cestu 5/2 a hlavice lisu je aktivována, plastový výlisek je proveden.
Lis se stáhne v okamžiku kdy je stisknuté bezpečnostní tlačítko EMERGENCY OFF (NC), nebo se rozepne senzor Safety fence (B1).
Když se sepne senzor vytažení pístu Cylinder extended B2 lis se zatáhne po 5 vteřinách zpět do výchozí pozice.
Instanční DB je použita jako paměť pro časovač.

Seznam úloh:

Adresa	Symbol	Komentář
%I 0.1	EMERGENCY OFF	EMERGENCY OFF tlačítko NC
%I 0.3	S3	Start tlačítko S3 NO
%I 0.4	B1	senzor Safety fence closed NO
%I 0.5	B2	senzor Cylinder extended NO
%Q 0.0	M0	Vysunout válec A



5. Programování lisu s časovým zpožděním pro SIMATIC S7-1200

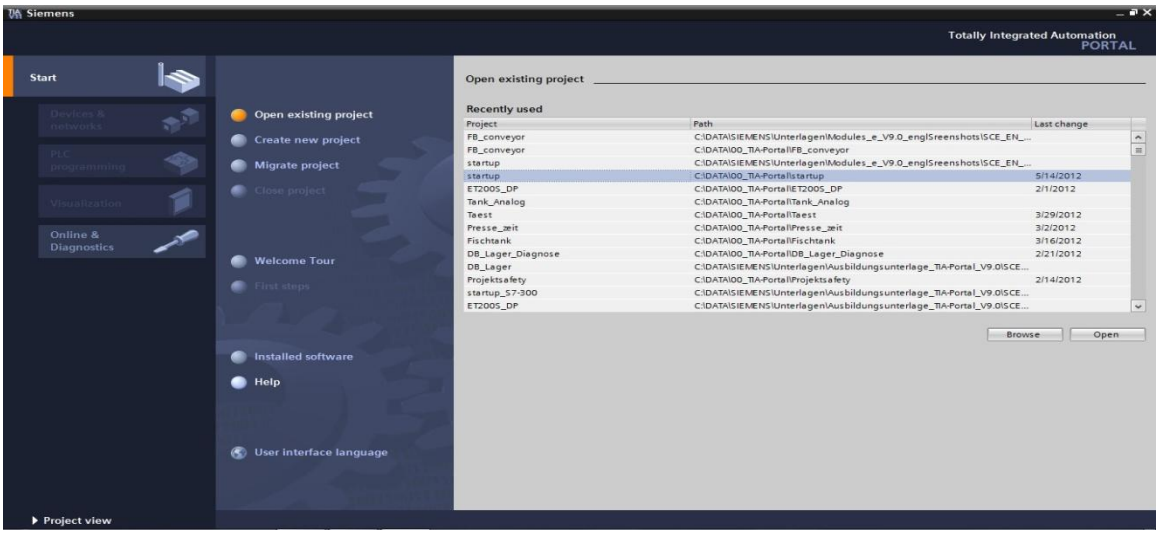
Projekt je vytvořen a programován v software ‘Totally Integrated Automation Portal’.

Zde pod společným rozhraním, jsou komponenty jako kontrolér, vizualizační panely a síťování automatizačních řešení nastaveny a programovány. Můžeme využít také online modu pro diagnostiku.

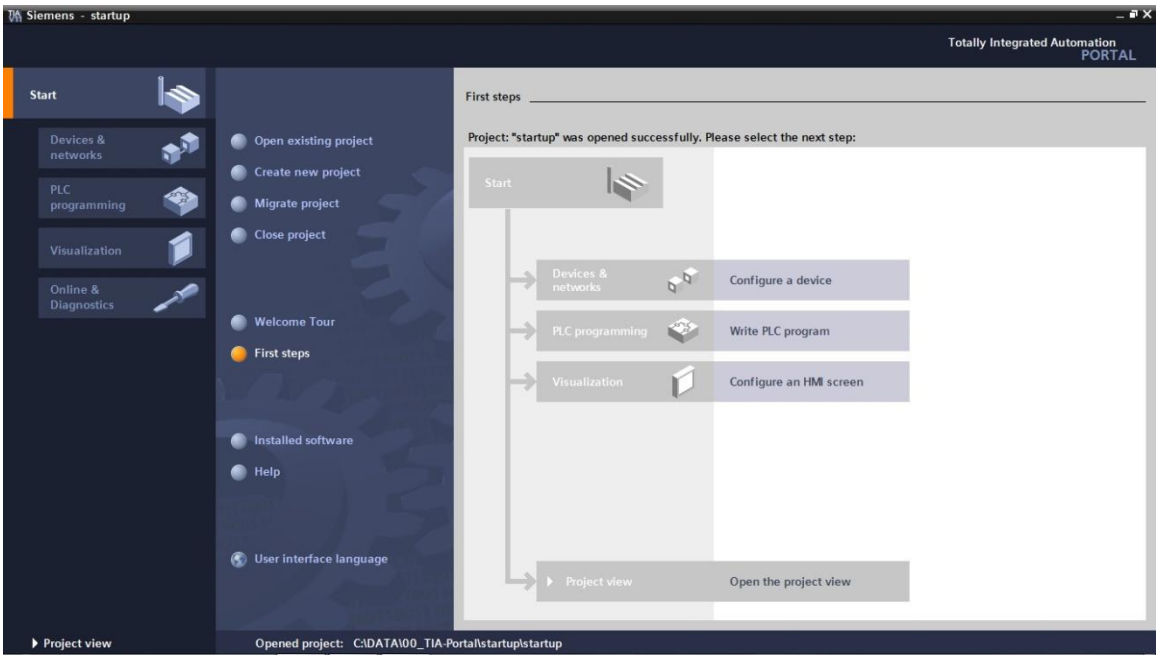
V krocích níže nastavíme projekt pro SIMATIC S7-1200 a naprogramujeme řešení úlohy:

1. Centrální nástroj je ’Totally Integrated Automation Portal’, který spustíme dvojklikem (→ Totally Integrated Automation Portal V11)

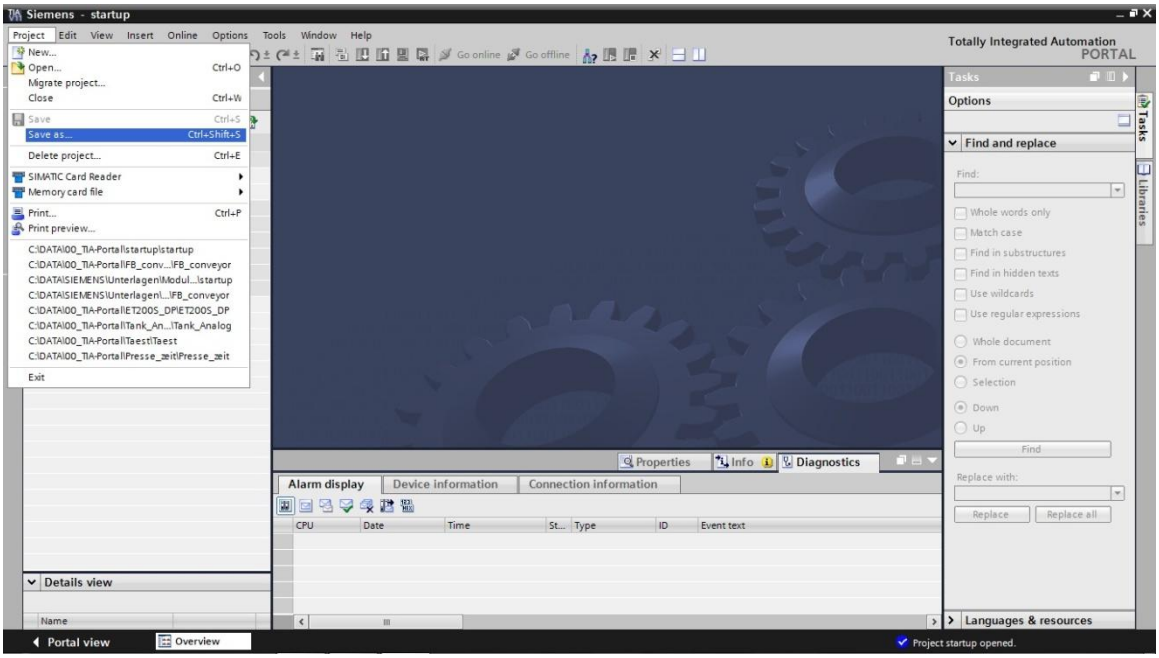
2. Projekt **"startup“** z modulu 010-010 je nyní otevřen v portal view jako základ pro tento program. (→ Open existing project → startup → open)



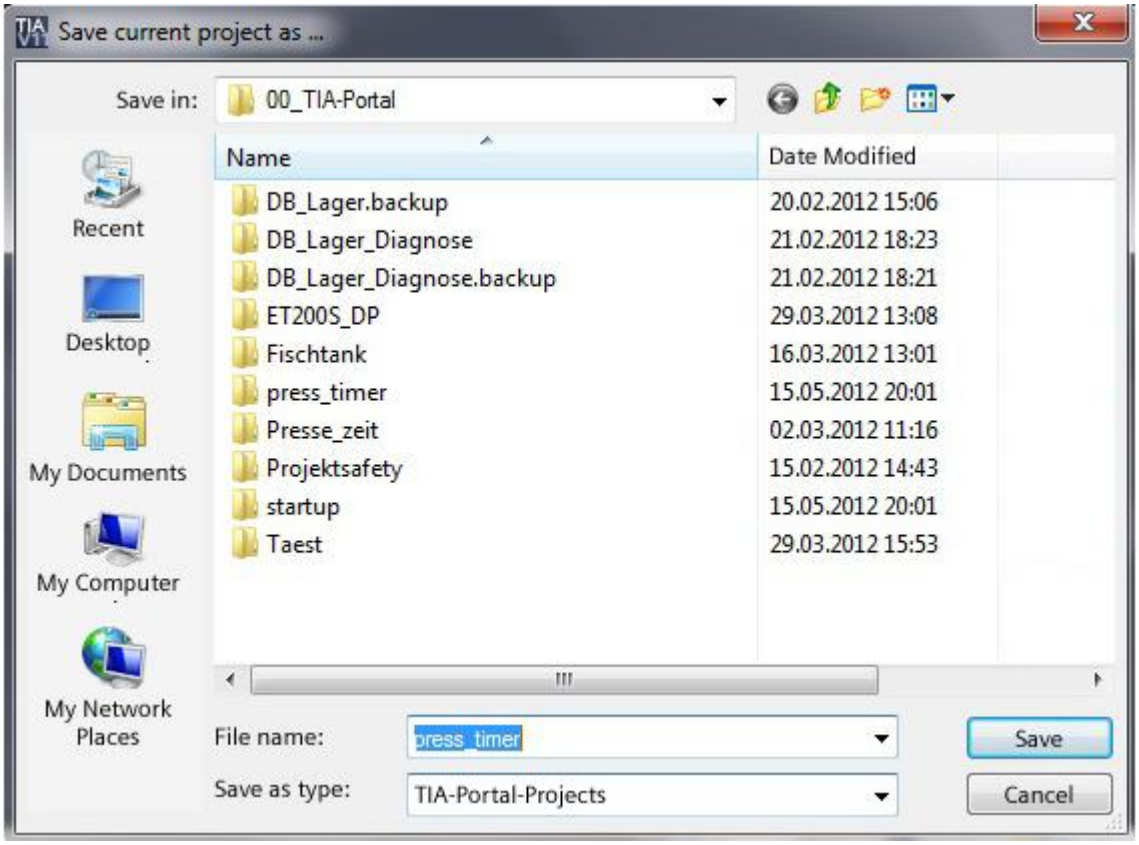
3. Dále, **'First Steps'** pro konfiguraci. My ale klikneme na **'Open project view'.** (→ Open project view)



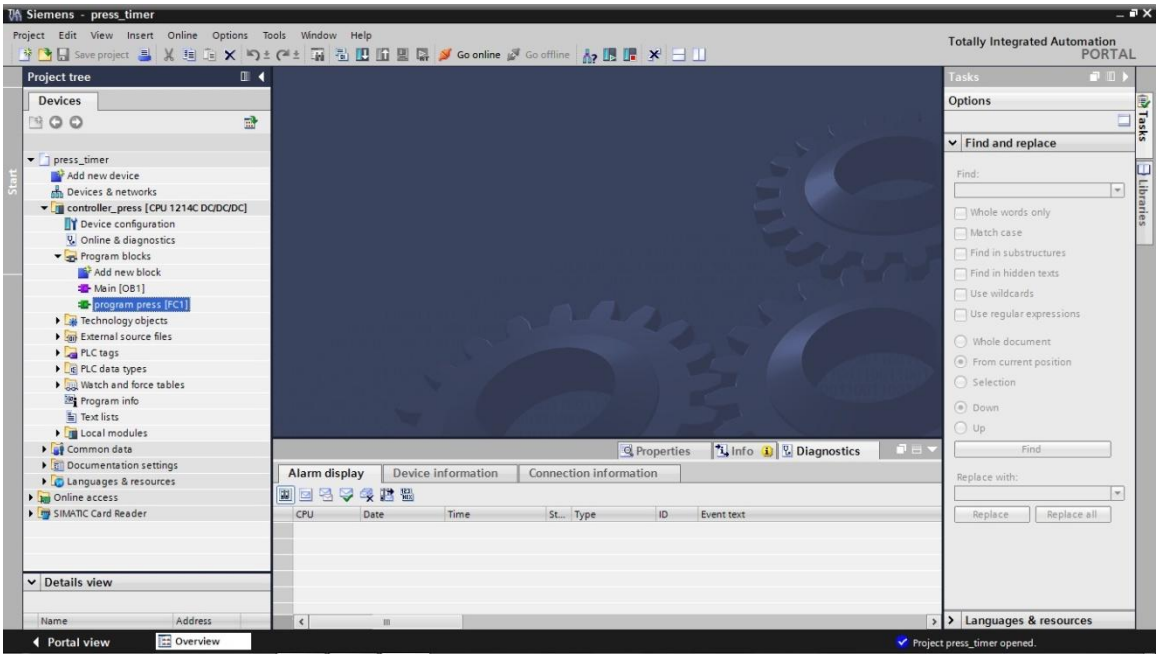
4. Nejdříve uložíme projekt pod jiným jménem. (→ Project → Save as)



5. Teď, **'Save'** uloží projekt pod novým jménem **'press_timer'.** (→ press_timer → Save)

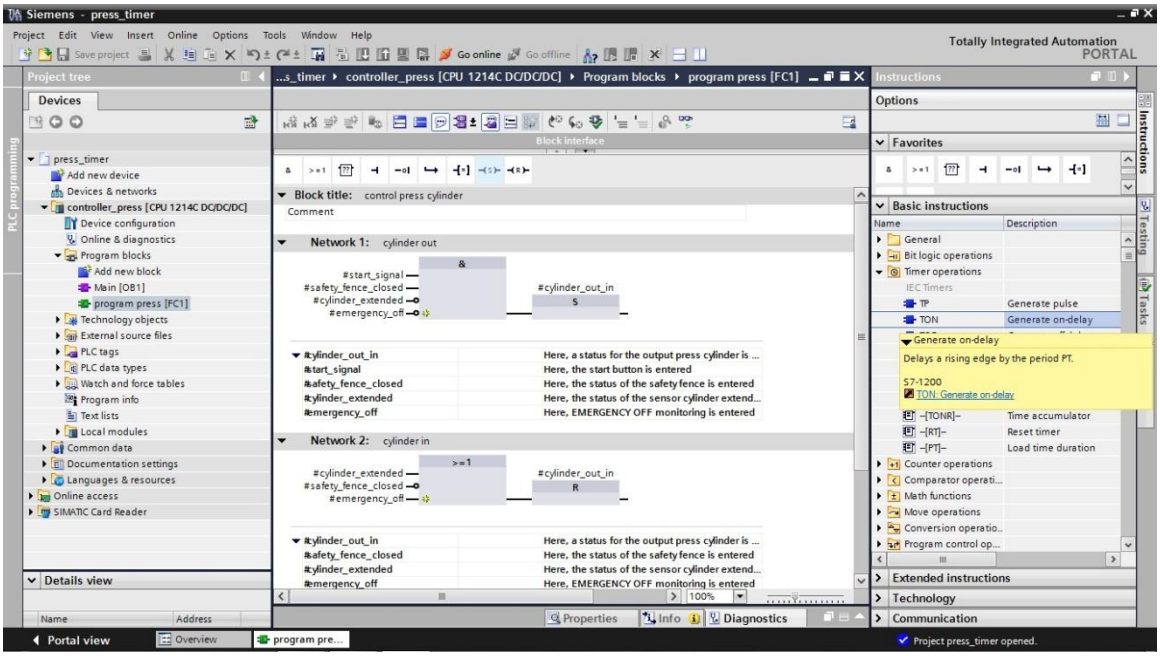


6. Pro potřebné úpravy otevřeme blok **'program press[FC1]'** dvojklikem. (→ program press[FC1])

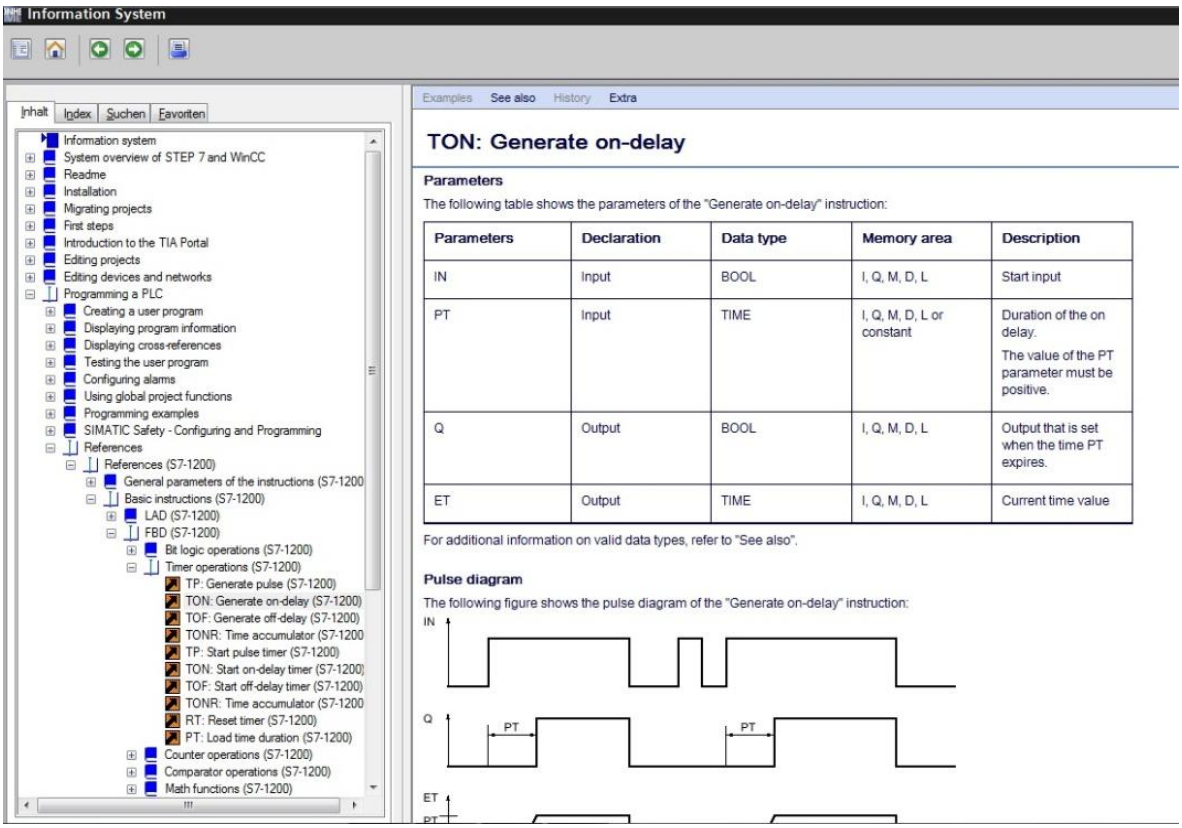


7. Nyní můžeme začít měnit program.

Při vytváření našeho řešení se zpožděním budeme potřebovat tzv. ON zpoždění **'TON'**. Najdeme ho v **'Instructions'** ve složce **'Timer operations'**. Když najedete myší na položku jako třeba TON, zobrazí se vám detaily o objektu. (→ Instructions → Timer operations → TON)



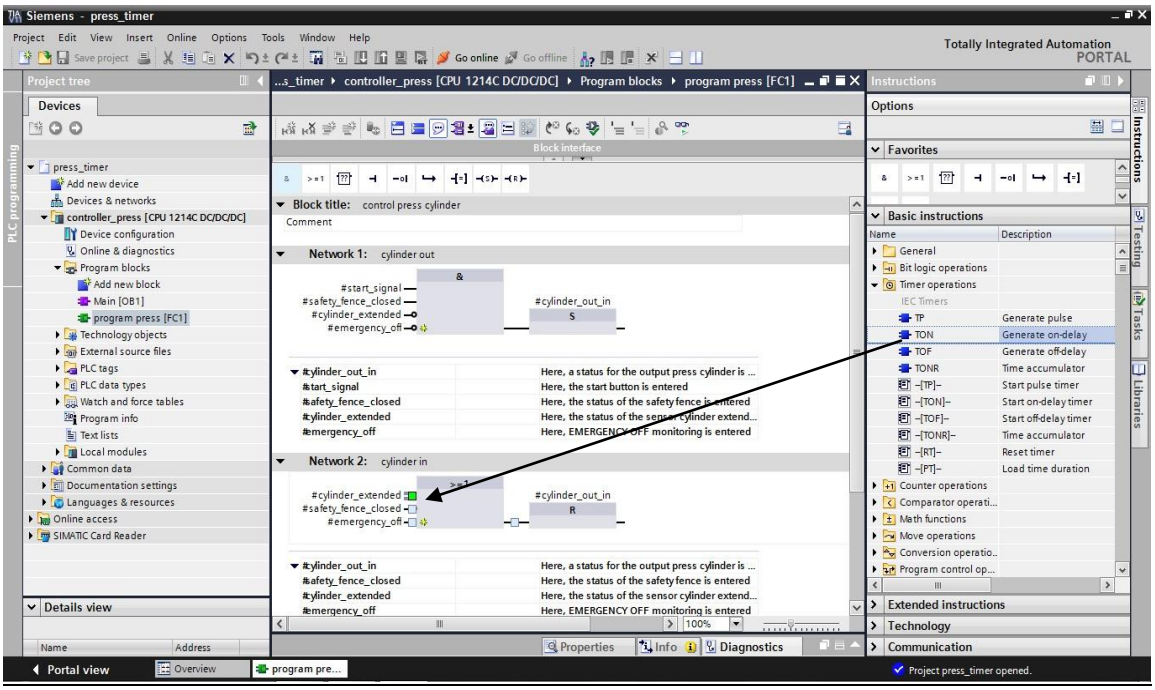
8. Když zvýrazníte objekt a stisknete F1 na vašem PC, zobrazí se vám online nápověda k objektu.. (→ F1)



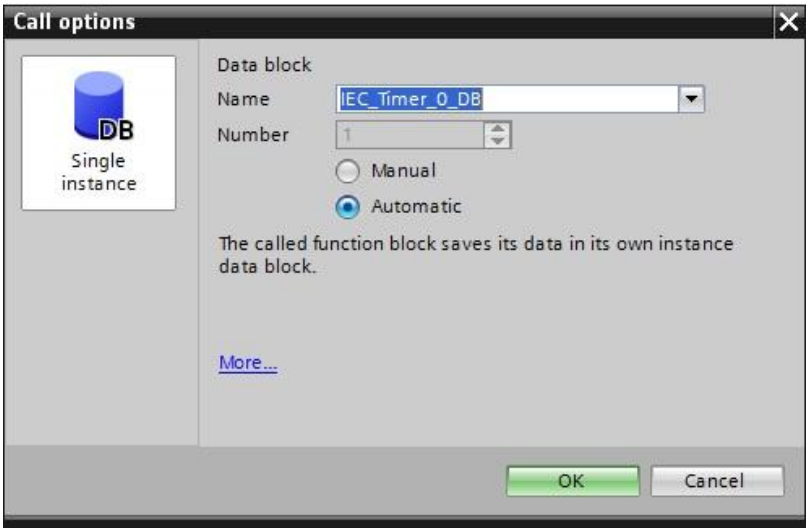
Poznámka

Zobrazená online nápověda pro všechny časovače.

9. Dále, přetáhneme časovač **'TON'** myší na třetí kontakt funkce OR za proměnnou **'#cylinder_extended'** (→ TON → #cylinder_extended)

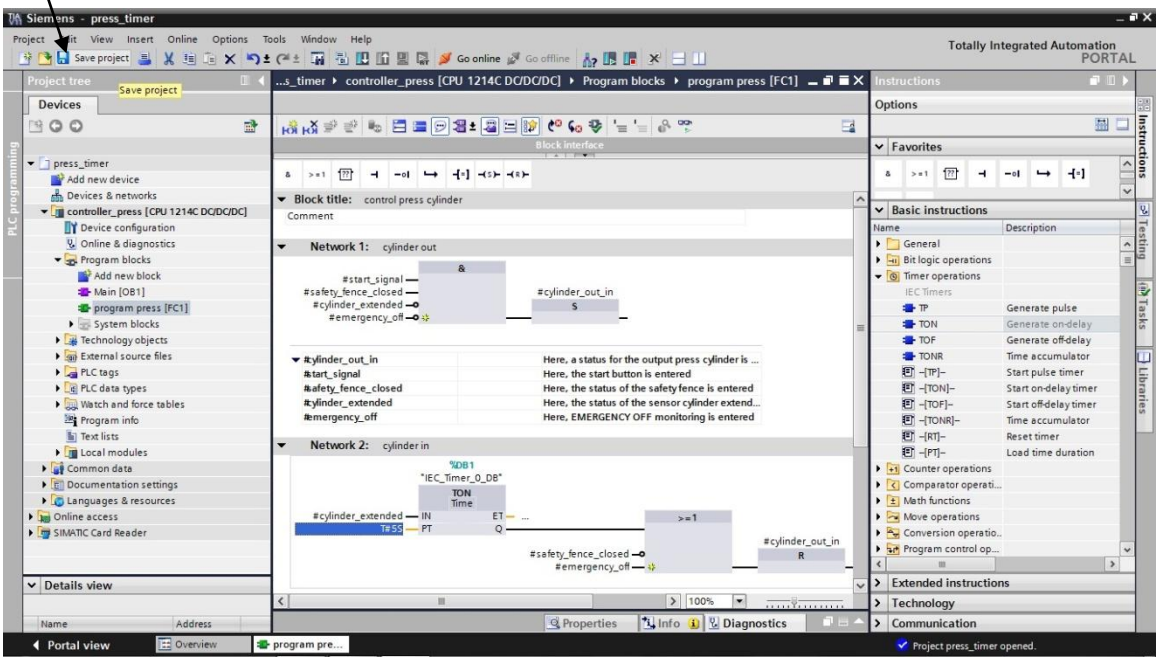


10. Pro časové funkce potřebujeme vyhradit paměť. Zde to můžeme udělat jen pomocí instančního data bloku jako **'Single instance'.** (→ OK)

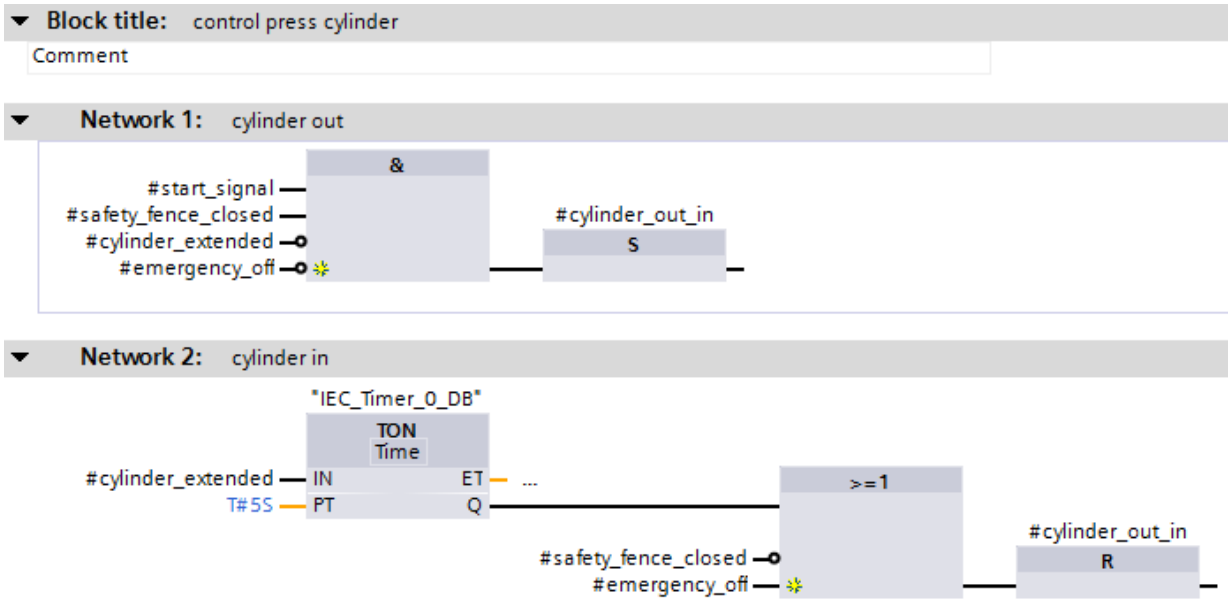


Poznámka
Multi-instanční může být použit pouze při programování ve funkčním bloku. To bude ukázáno níže v příkladu s IEC čítačem.

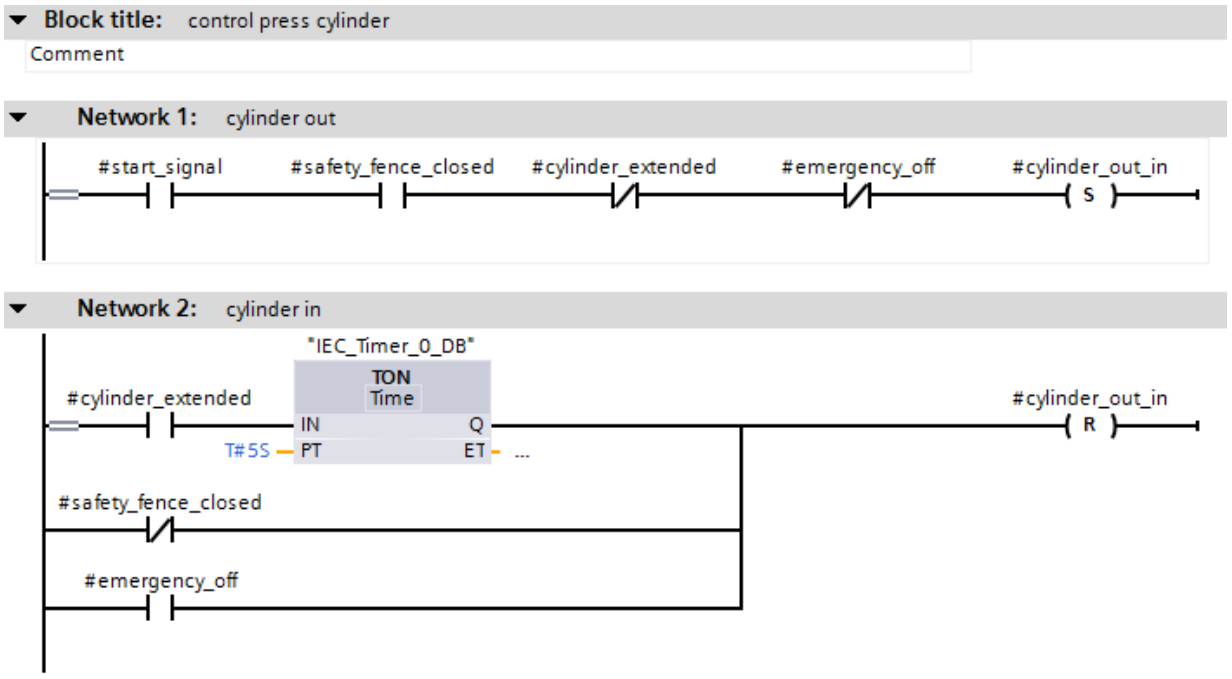
11. Nyní připojíme časové zpoždění **'TON'** se základnou **'t#5s'** 5 sekund. Kliknutím na Save project, je projekt uložen. (→ t#5s → Save project)



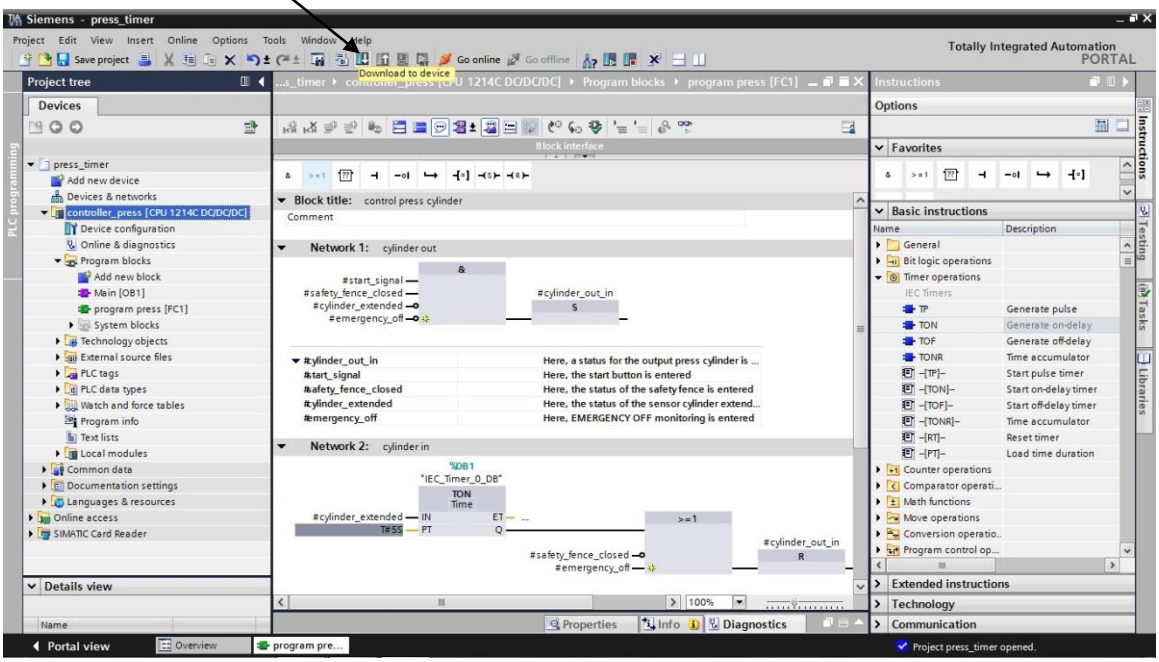
Program ve funkčním blokovém diagramu (FBD)



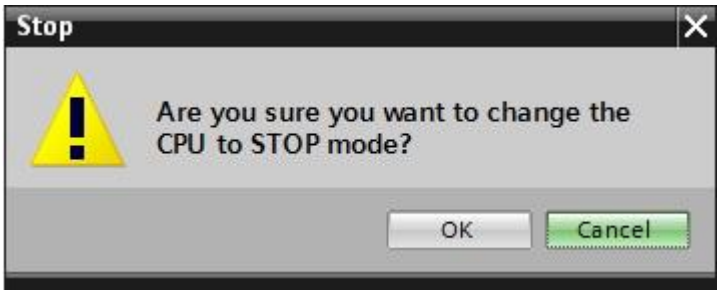
Program v žebříkovém diagramu (LAD)



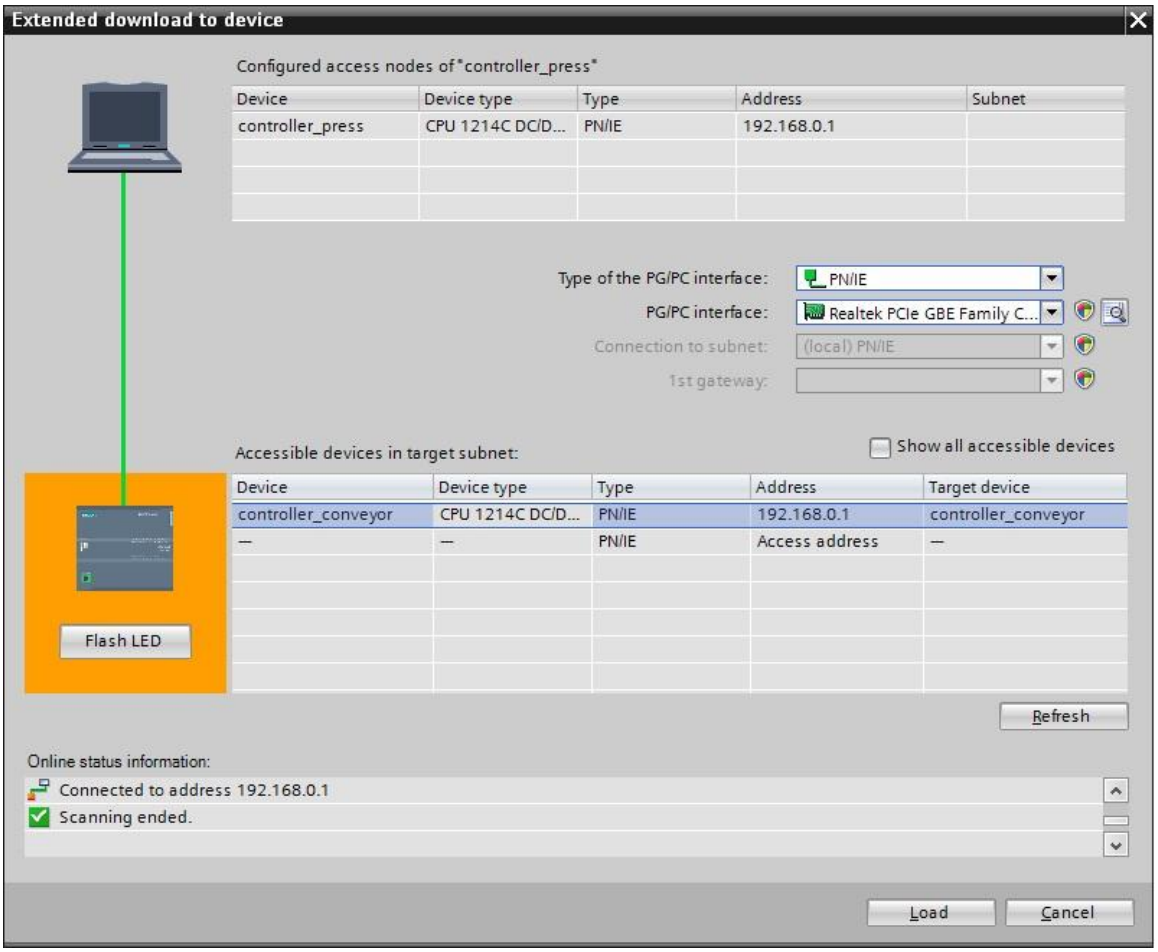
12. Pro nahrání celého programu do CPU, zvýrazníme složku **'controller_press'** klikneme na symbol Nahrání do zařízení. (→ controller_press →)



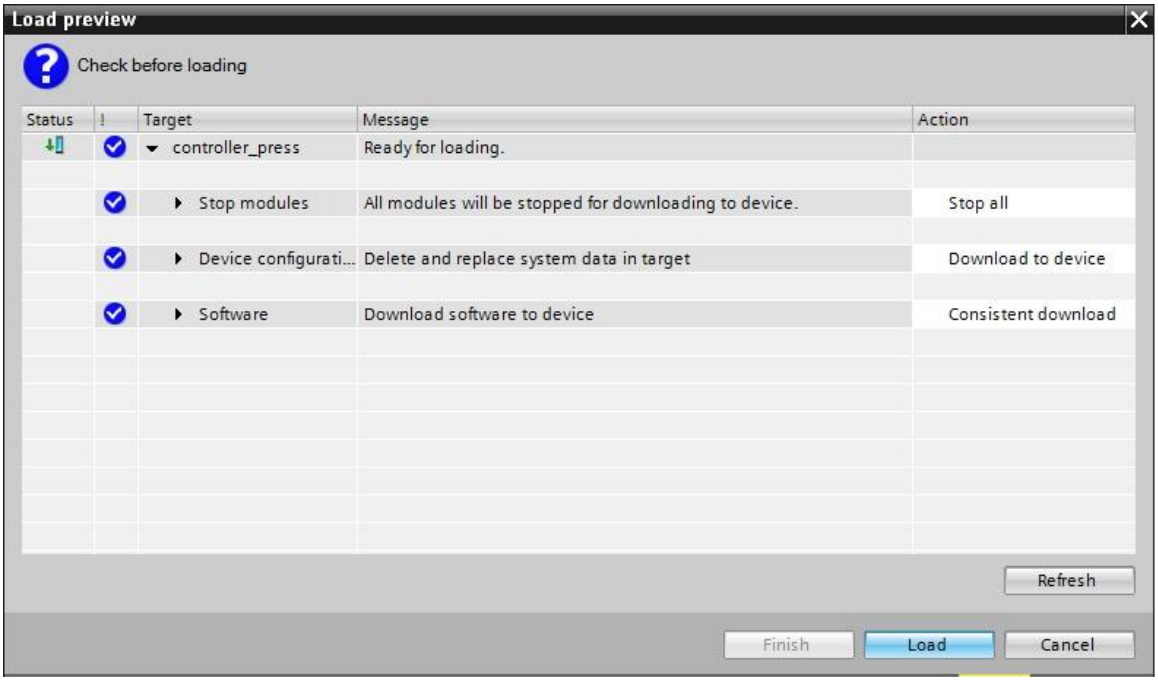
13. Pokud je CPU v režimu **'RUN',** budete dotázáni, jestli ho opravdu chcete převést do režimu **'STOP'**. Potvrdíme **'OK'.** (→ OK)



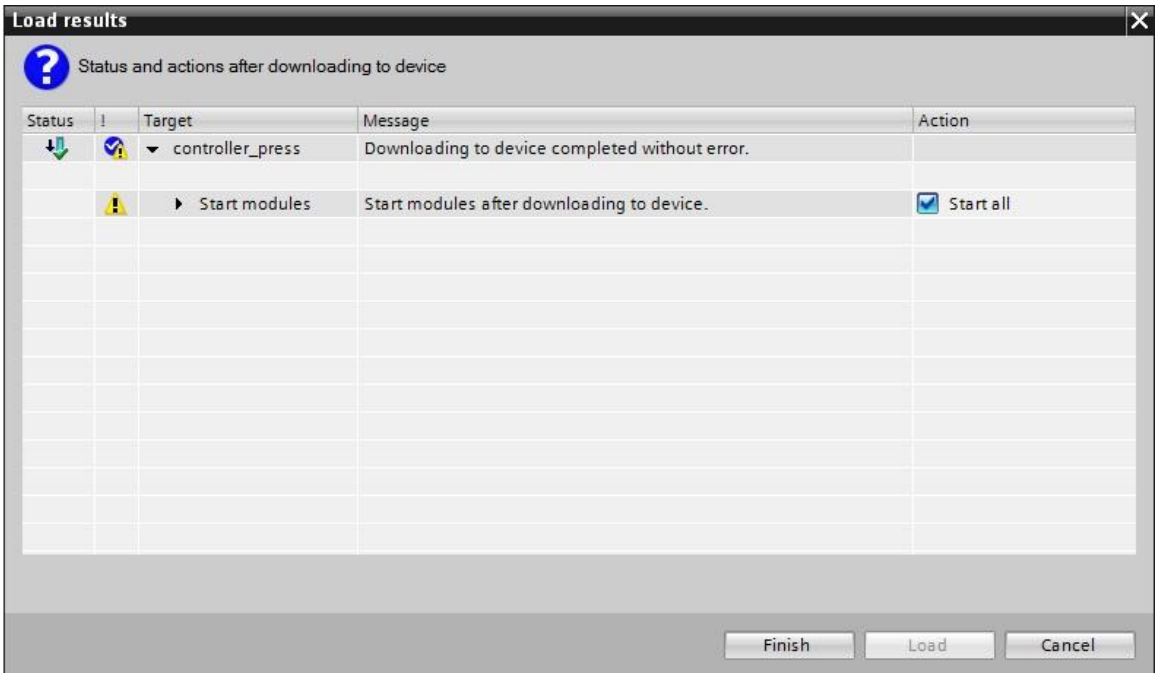
14. Pokud jste opomněli specifikovat rozhraní PG/PC předtím, v zobrazeném okně to můžete udělat teď. (→ PG/PC interface for loading → Load)



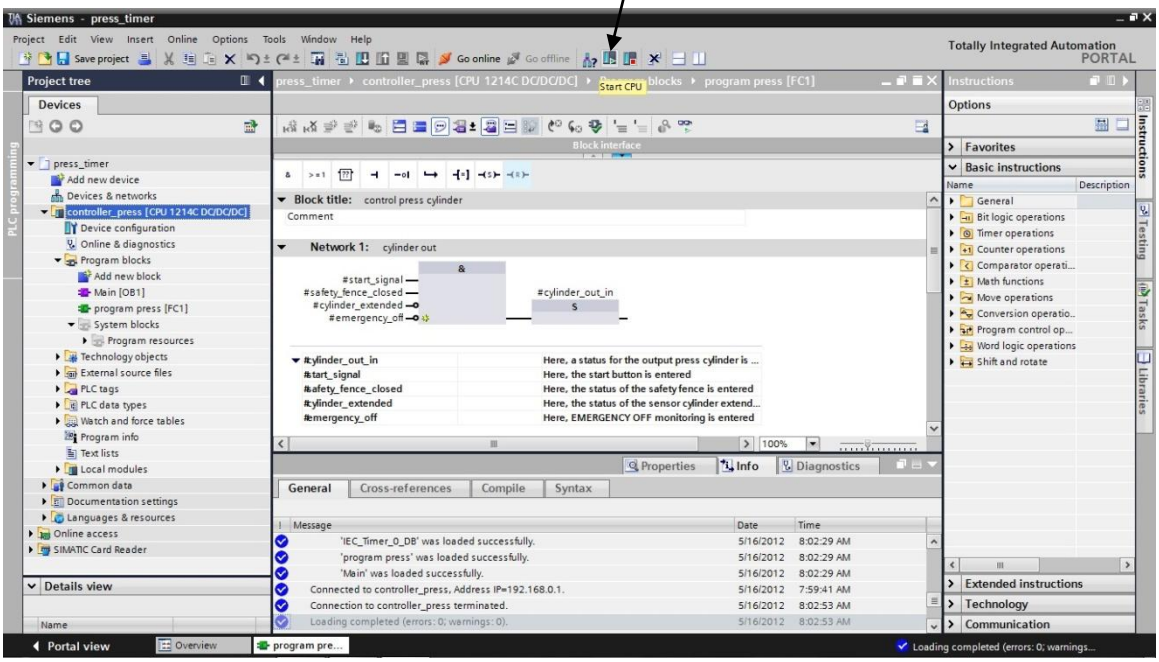
15. Potvrdíme **'Load'** ještě jednou. Během nahrávání vidíme status v okně. (→ Load)



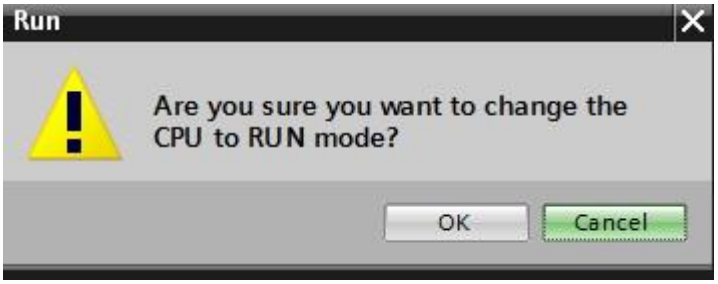
16. Pokud bylo nahrávání úspěšné, zobrazí se nám zpráva v okně. Klikneme na **'Finish'.** (→ Finish)



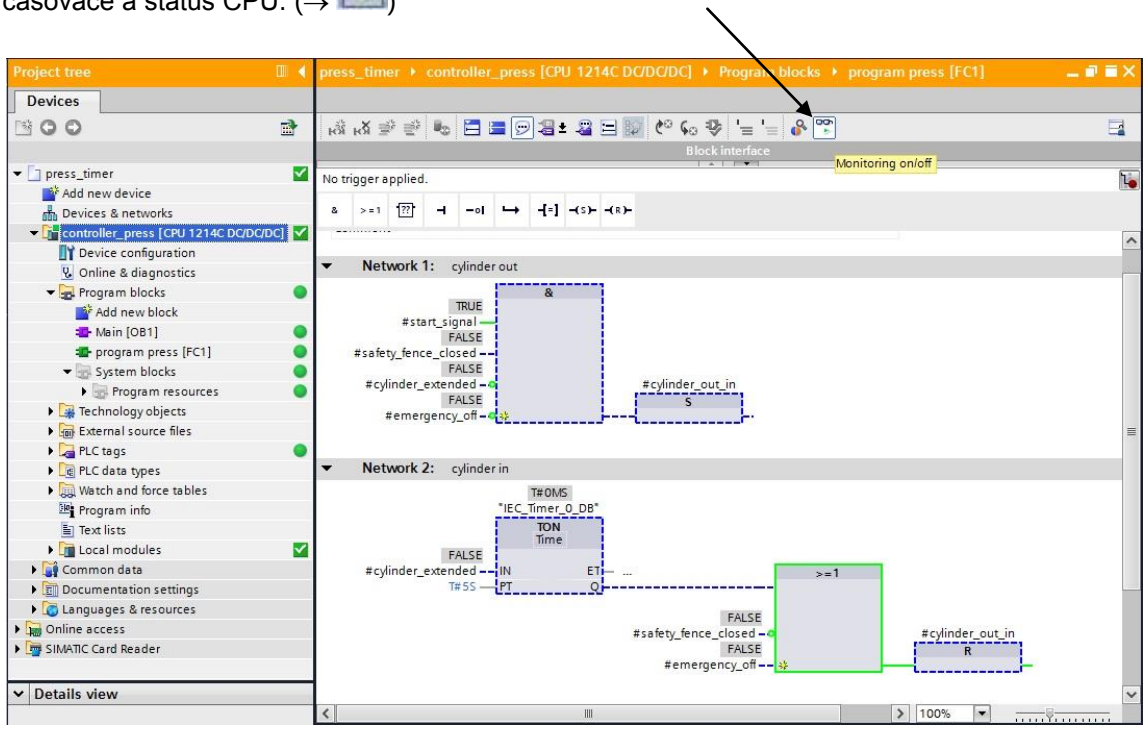
17. Dále spustíme, CPU kliknutím na  . (→ )



18. Potvrdíme, že chceme opravdu spustit CPU kliknutím na **'OK'.** (→ OK)



19. Kliknutím na symbol  monitoring on/off můžeme sledovat během testování programu stav časovače a status CPU. (→ )



6. Vzorová úloha pro řízení pásového dopravníku s čítačem a Multi-instancí

Chceme-li vytvořit blok, tvářící se jako ‚černá skříňka' v jakémkoliv programu, musíme pro něj využít proměnné. V tomto případě využijeme následující pravidla: že v těchto blocích, nebude absolutní adresování vstupů a výstupů, flagy apod. v bloku použijeme jen konstanty a proměnné.

Pokud voláme z bloku sekundární funkční blok nebo časovač/čítač vícekrát za sebou. Nesmíme jim přidělit jejich vlastní data blok.
Požadovanou paměť poskytneme ve formě **multi-instance** pomocí instančního DB, který je přiřazen k bloku, vyvolávajícímu tyto funkce.

V příkladě níže, přidáme počítadlo lahví do funkčního bloku, který už obsahuje řízení dopravníkového pásu závislé na operačním módu.

Pomocí dopravníku, chceme transportovat 20 lahví do krabice. Když je krabice plná, dopravník se zastaví a krabici musíme vyzvednout.

Tlačítkem 'S1', chceme zvolit operační mód 'Manual', a tlačítkem 'S2' operační mód 'Automatic'.
V operačním módu 'Manual', je motor zapnut, dokud je stisknuté tlačítko 'S3'; tlačítko 'S4' nesmí být stisknuto.
V operačním módu 'Automatic', se zapne motor dopravníku tlačítkem 'S3' a vypne tlačítkem 'S4' (rozpojení kontaktu).
Navíc je zde senzor 'B0', který počítá lahve jdoucí do přepravky. Po napočítání 20 lahví se pás zastaví.
Když dáme na místo novou přepravku musí to být potvrzeno tlačítkem 'S5'.

Seznam úloh:

Adresa	Symbol	Komentář
%I 0.0	S1	Tlačítko pro mód Manual S1 NO
%I 0.1	S2	Tlačítko pro mód Automatic S2 NO
%I 0.2	S3	Tlačítko zapnutí S3 NO
%I 0.3	S4	Tlačítko vypnutí S4 NC
%I 0.6	S5	Tlačítko S5 NO Reset čítače/nová přepravka
%I 0.7	B0	Senzor B0 NO čítač lahví
%Q 0.2	M1	Motor dopravníku M1

7. Programování pásového dopravníku se SIMATIC S7-1200

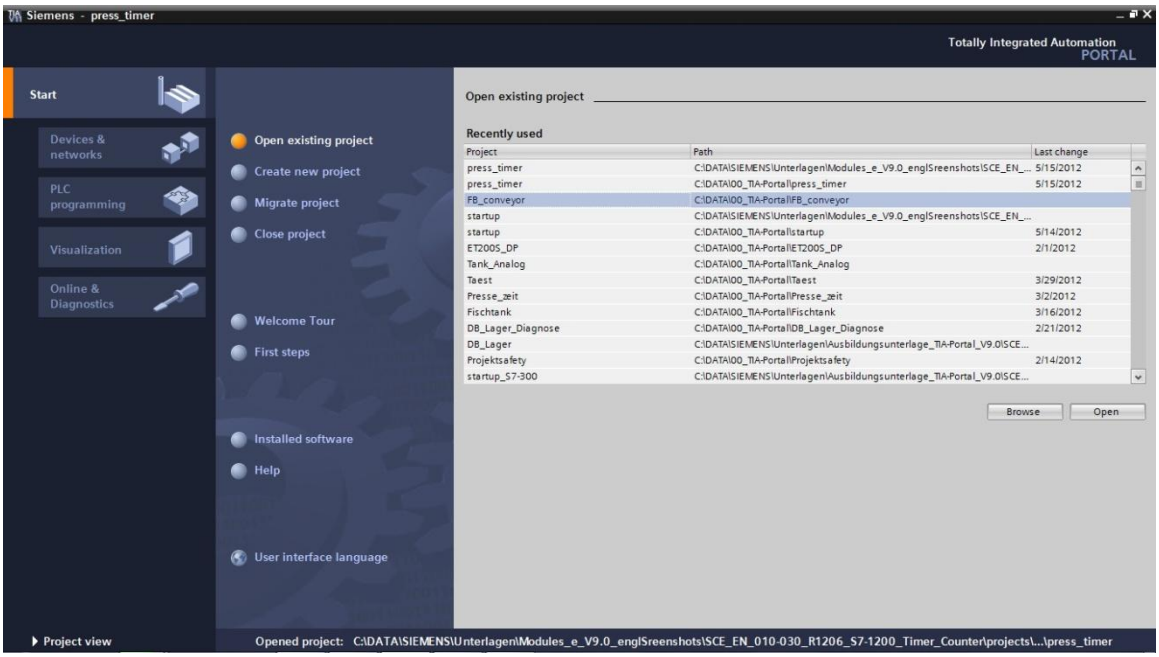
Projekt je vytvořen a programován v software ‘Totally Integrated Automation Portal’.

Zde pod společným rozhraním, jsou komponenty jako kontrolér, vizualizační panely a síťování automatizačních řešení nastaveny a programovány. Můžeme využít také online modu pro diagnostiku.

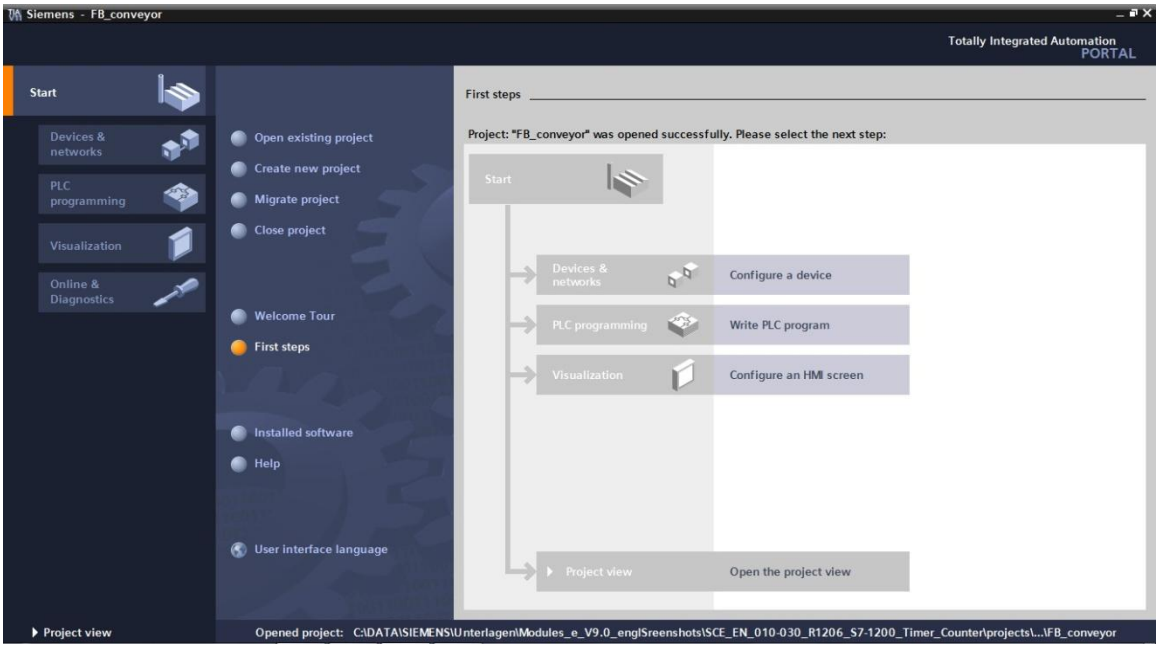
V krocích níže nastavíme projekt pro SIMATIC S7-1200 a naprogramujeme řešení úlohy:

1. Centrální nástroj je ’Totally Integrated Automation Portal’, který spustíme dvojklikem (→ Totally Integrated Automation Portal V11)

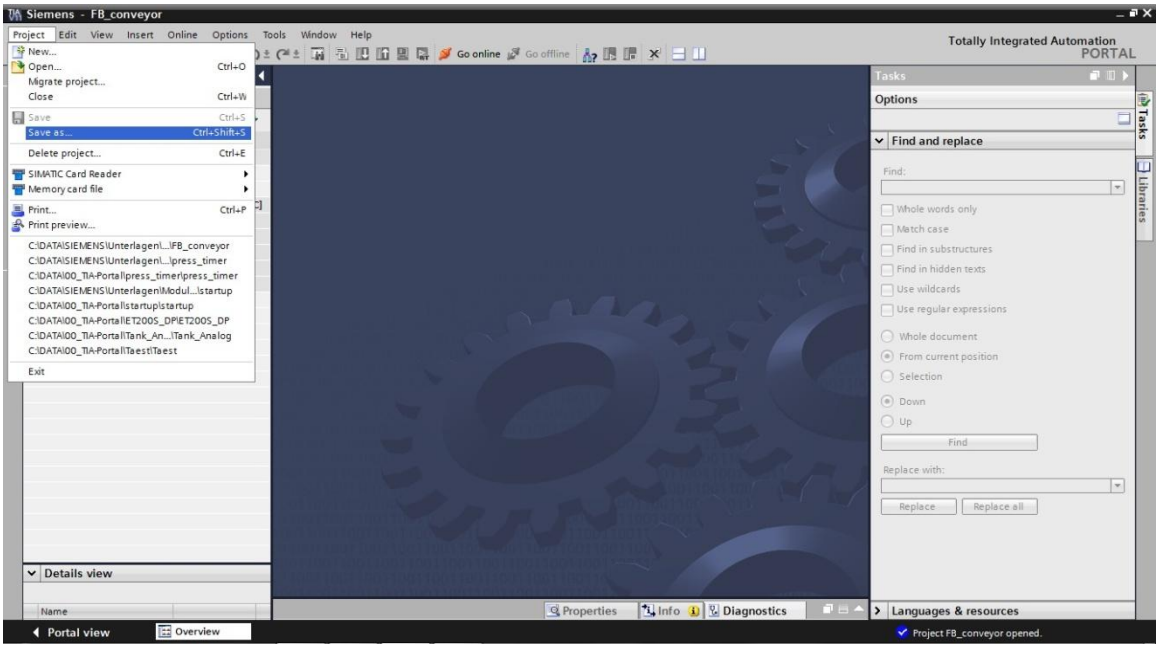
2. Otevřeme projekt **"FB_conveyor“** z modulu 010-020 v portal view jako základ pro náš program. (→ Open existing project → FB_conveyor → Open)



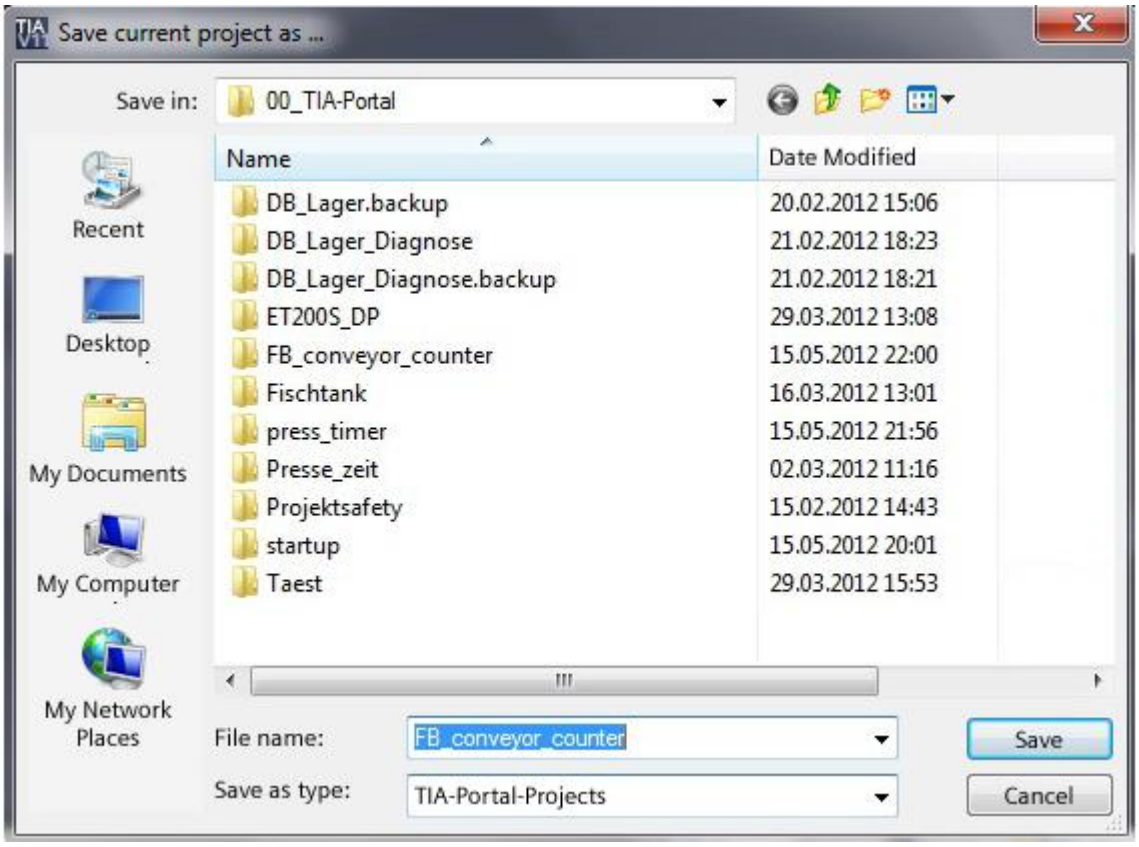
3. Dále, **'First Steps'** jsou navrženy pro konfiguraci a nastavení. My chceme **'Open project view'.** (→ Open project view)



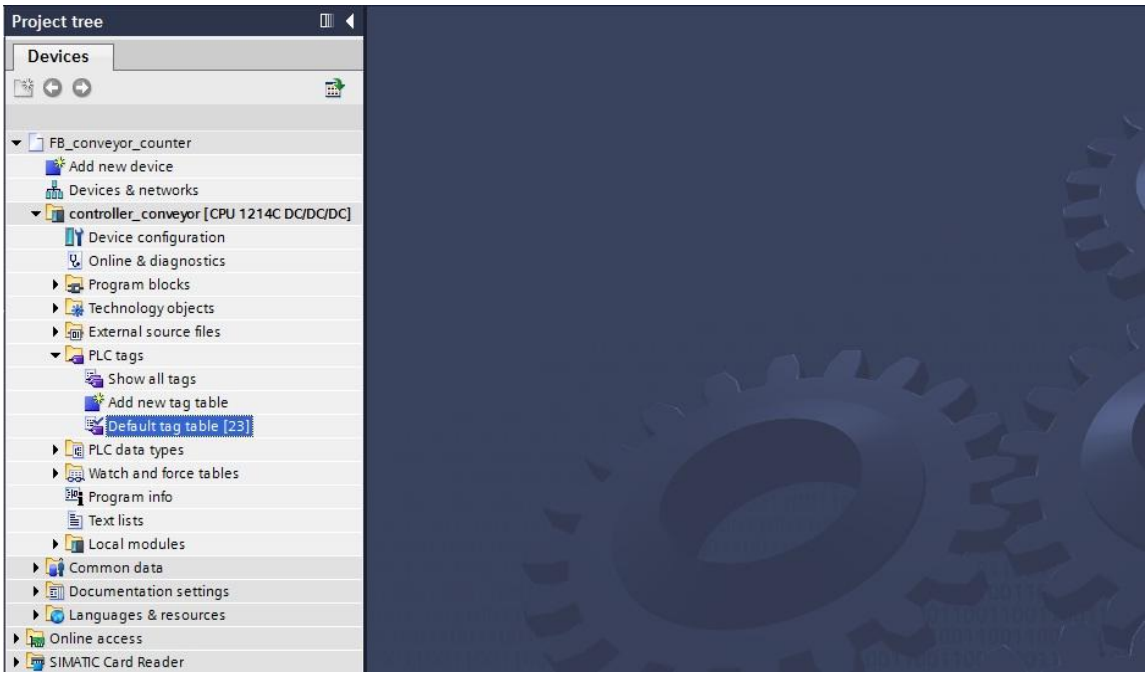
4. Nyní, nejdříve uložíme projekt pod jiným jménem. (→ Project → Save as)



5. Dále, **'Save'** projekt uloží pod jiným jménem **'FB_conveyor_counter'.** (→ FB_conveyor_counter → Save)



6. Pro nastavení nových globálních proměnných, otevřeme tabulku PLC tagů **'PLC tags'** v **'controller_conveyor'** 'Default **tag table.** (→ controller_conveyor → PLC tags → Default tag table)

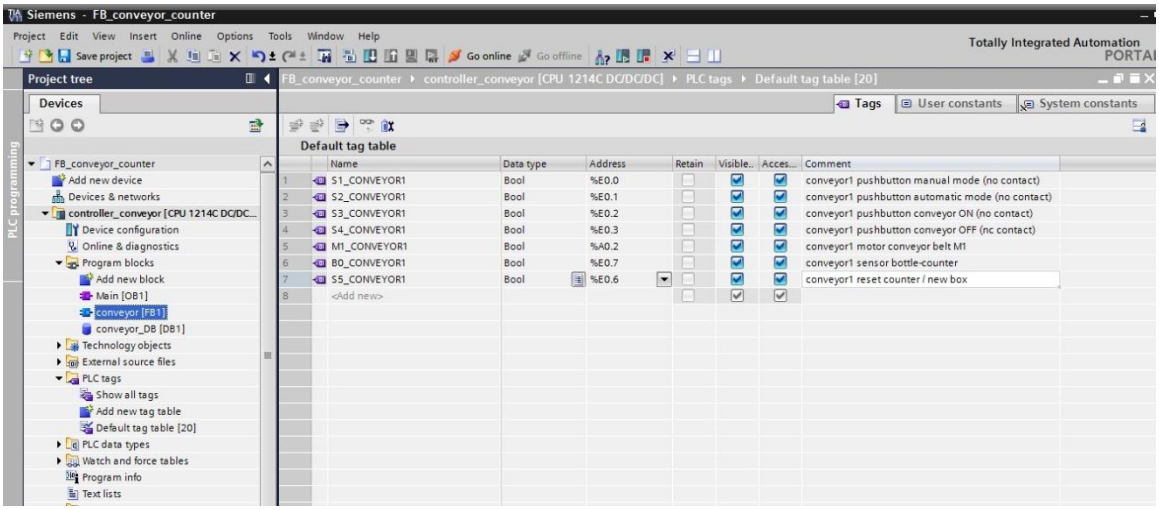


7. Změníme tabulku podle základního nastavení.

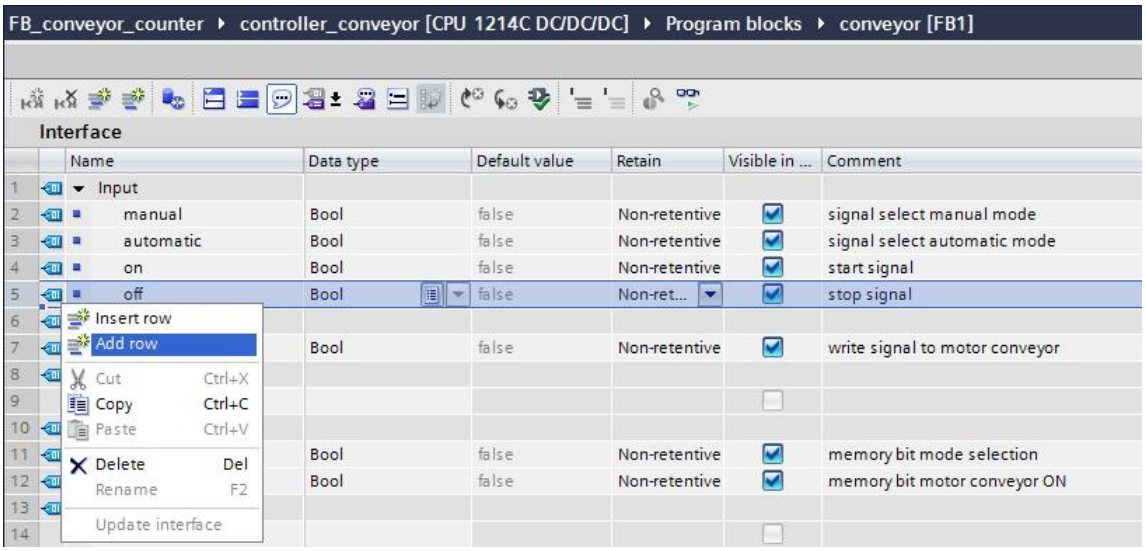
Dále nastavíme dvě globální proměnné **'B0'** a **'S5'.** (→ B0/Bool/%I0.7/sensor bottle counter → S5/Bool/%I0.6/reset counter/new box)

	Name	Data type	Address	Retain	Visible..	Acces...	Comment
1	S1_CONVEYOR1	Bool	%E0.0	☐	☑	☑	conveyor1 pushbutton manual mode (no contact)
2	S2_CONVEYOR1	Bool	%E0.1	☐	☑	☑	conveyor1 pushbutton automatic mode (no contact)
3	S3_CONVEYOR1	Bool	%E0.2	☐	☑	☑	conveyor1 pushbutton conveyor ON (no contact)
4	S4_CONVEYOR1	Bool	%E0.3	☐	☑	☑	conveyor1 pushbutton conveyor OFF (nc contact)
5	M1_CONVEYOR1	Bool	%A0.2	☐	☑	☑	conveyor1 motor conveyor belt M1
6	B0_CONVEYOR1	Bool	%E0.7	☐	☑	☑	conveyor1 sensor bottle-counter
7	S5_CONVEYOR1	Bool	%E0.6	☐	☑	☑	conveyor1 reset counter / new box

8. Pro změnu programu otevřeme blok **'conveyor[FB1]'** dvojklikem.
(→ conveyor[FB1])



9. Nejdříve, přidáme dva řádky pod interface pro vstupní proměnné. (→ Interface → Input → Add row)



10. Při deklarování lokálních proměnných, přidáme následující proměnné.

Vstup:

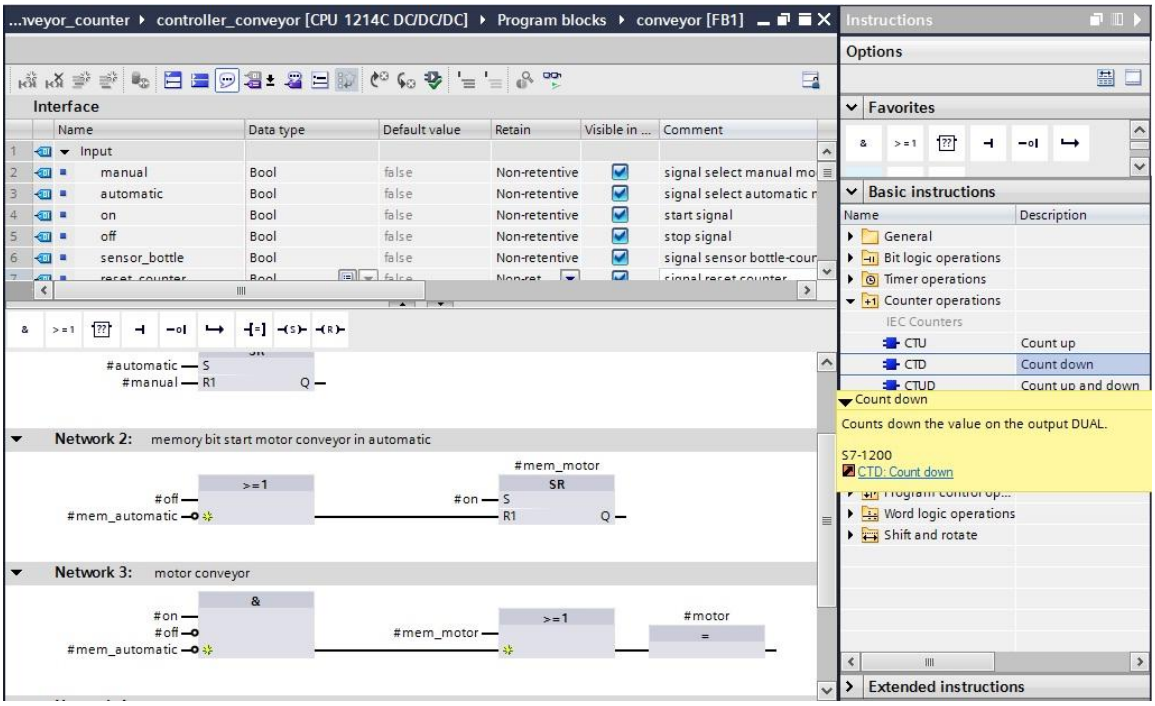
sensor_bottle	Senzor lahví je dotázán na stav.
reset_counter	Signál pro resetování čítače.

Interface

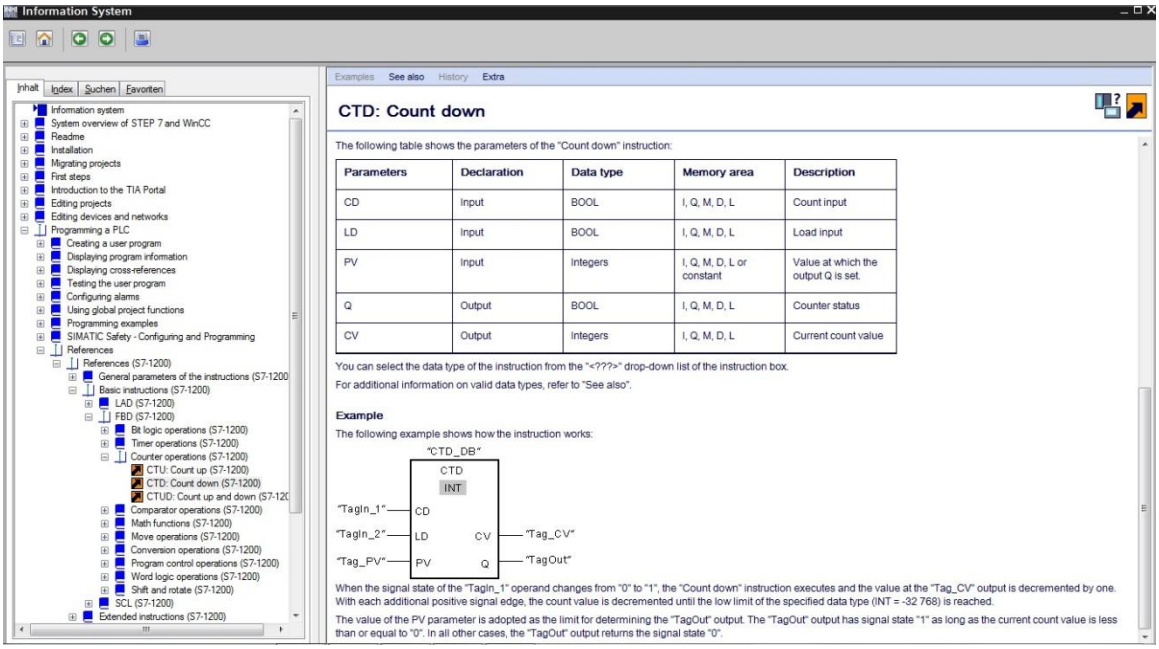
		Name	Data type	Default value	Retain	Visible in ...	Comment
1		Input					
2		manual	Bool	false	Non-retentive	☑	signal select manual mode
3		automatic	Bool	false	Non-retentive	☑	signal select automatic mode
4		on	Bool	false	Non-retentive	☑	start signal
5		off	Bool	false	Non-retentive	☑	stop signal
6		sensor_bottle	Bool	false	Non-retentive	☑	signal sensor bottle-counter
7		reset_counter	Bool	false	Non-ret...	☑	signal reset counter
8		<Add new>				☐	
9		Output					
10		motor	Bool	false	Non-retentive	☑	write signal to motor conveyor
11		InOut					
12		<Add new>				☐	
13		Static					
14		mem_automatic	Bool	false	Non-retentive	☑	memory bit mode selection
15		mem_motor	Bool	false	Non-retentive	☑	memory bit motor conveyor ON
16		Temp					
17		<Add new>				☐	

11. Nyní můžeme začít měnit program.

Pro naše řešení, potřebujeme čítač, který počítá dolů **'CTD'**. Najdeme ho pod **'Instructions'** ve složce **'Counters'**. Pokud najedeme myší na položku jako třeba CTD, zobrazí se nám informace o položce. (→ Instructions → Counters → CTD)



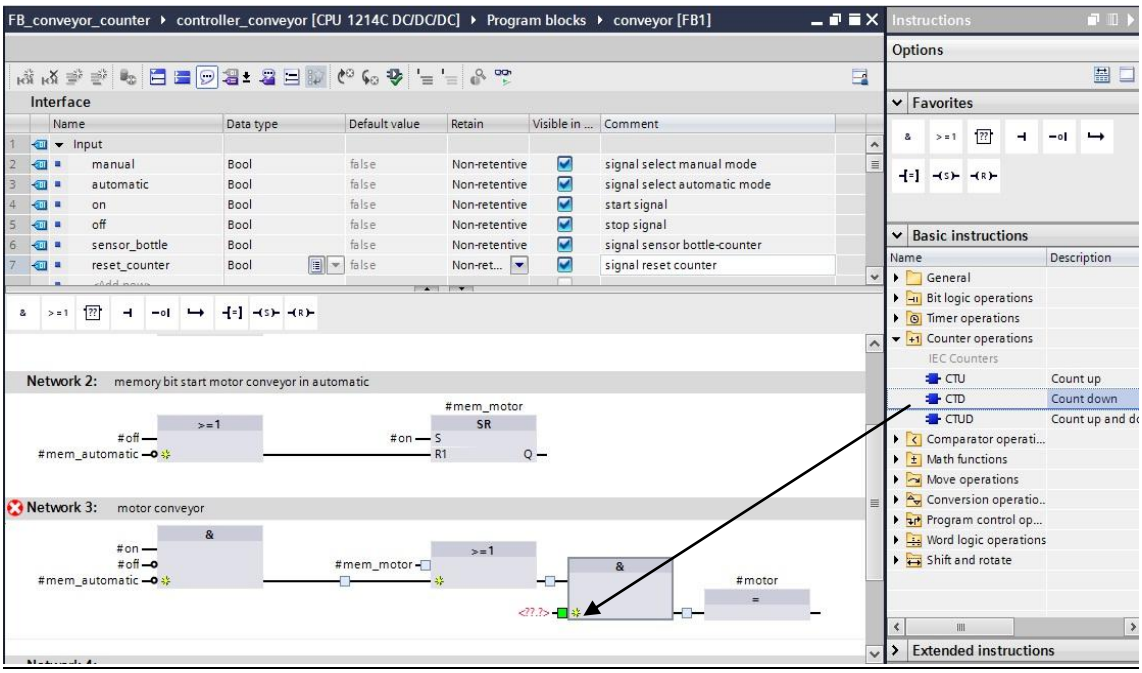
12. Když objekt zvýrazníme a zmáčkneme **'F1'** klávesu na PC, zobrazí se nám k objektu online nápověda. (→ F1)



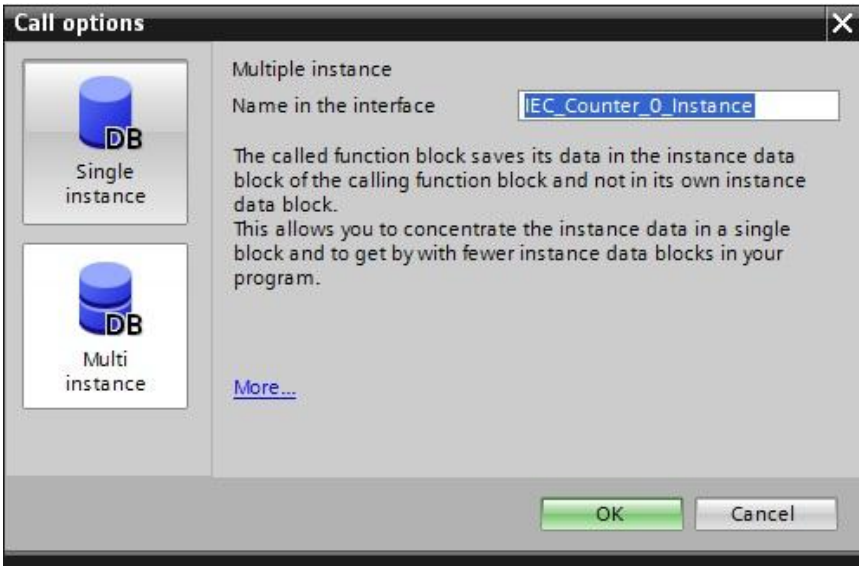
Poznámka

Zde se nám zobrazuje online nápověda ke všem čítačům.

13. Nyní nejdříve přidáme AND mezi OR a úlohu, pak přetáhneme čítač **'CTD'** na druhý kontakt funkce AND. (→ & → CTD)



14. Pro čítač potřebujeme paměť. Tady jí dostaneme jako instanční data blok v podobě **'Multi-Instance'**, bez nutnosti vytvářet nový funkční blok. (→ Multi-Instance → OK)

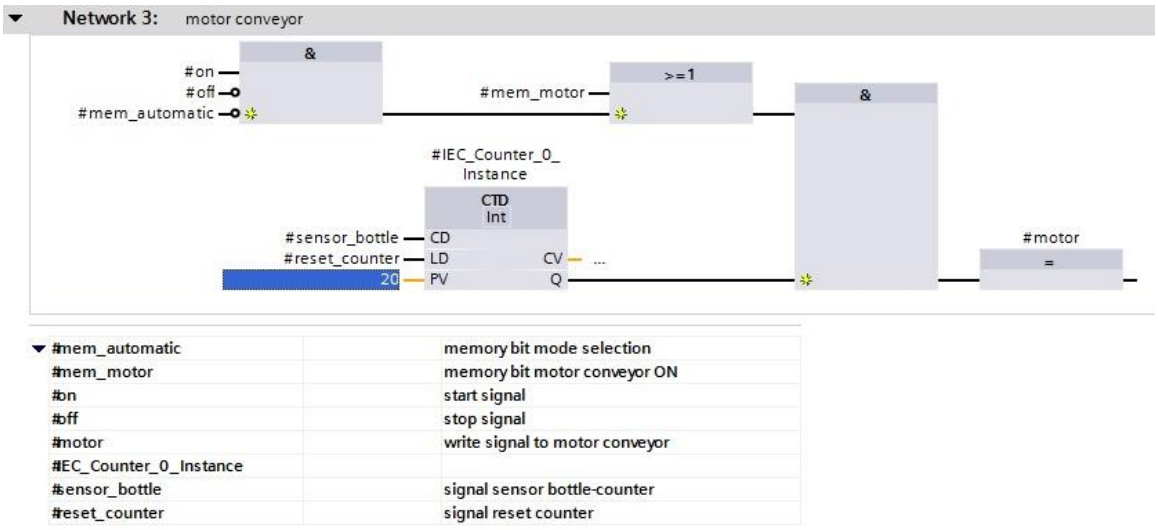


Poznámka

Multi-instance se dá využít pouze když programujeme ve funkčním bloku.

15. Nyní spojíme čítač dolů **'CTD'** se specifickou hodnotou **'PV'** pro 20 lahví a připojíme vstup **'CD'** na **'#sensor_bottle'**, a vstup **'LD'** na **'#reset_counter'**. Nakonec znegujeme druhý kontakt AND funkce.

Kliknutím na Save project se projekt uloží. (→ 20 → #sensor_bottle → #reset_counter → −o| → Save project)

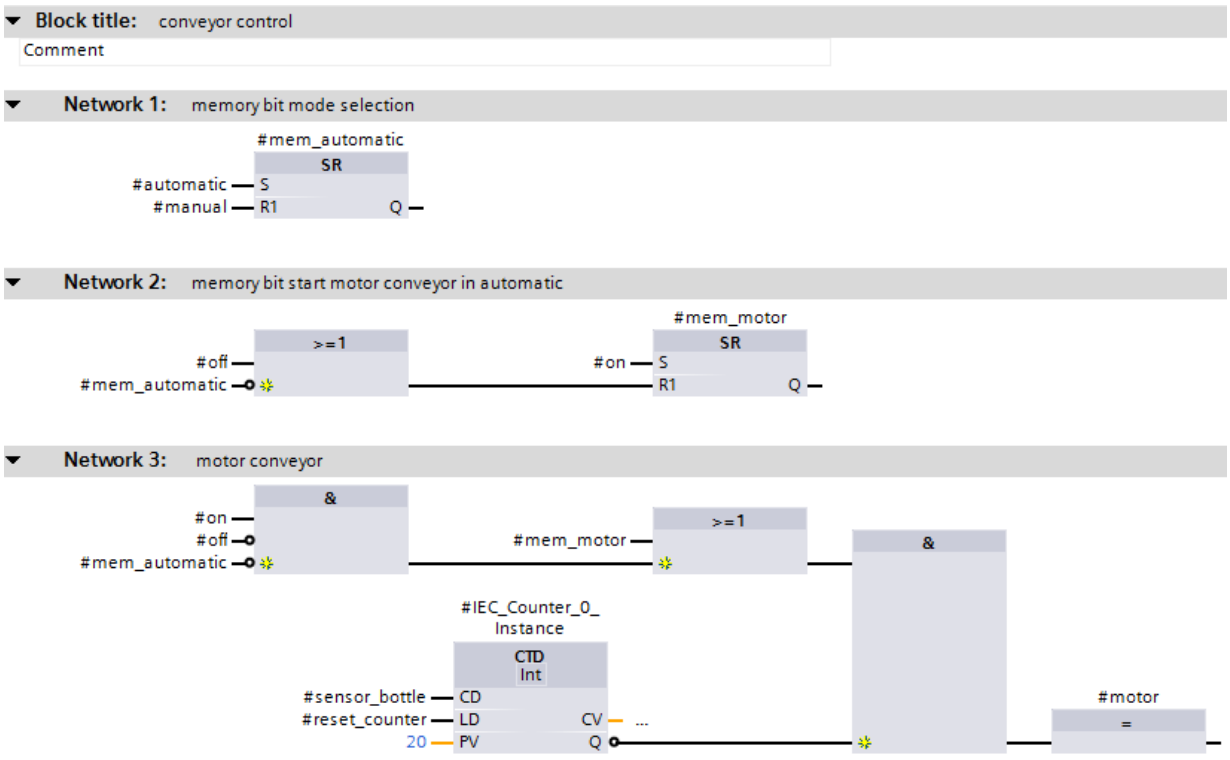


#mem_automatic		memory bit mode selection
#mem_motor		memory bit motor conveyor ON
#on		start signal
#off		stop signal
#motor		write signal to motor conveyor
#IEC_Counter_0_Instance		
#sensor_bottle		signal sensor bottle-counter
#reset_counter		signal reset counter

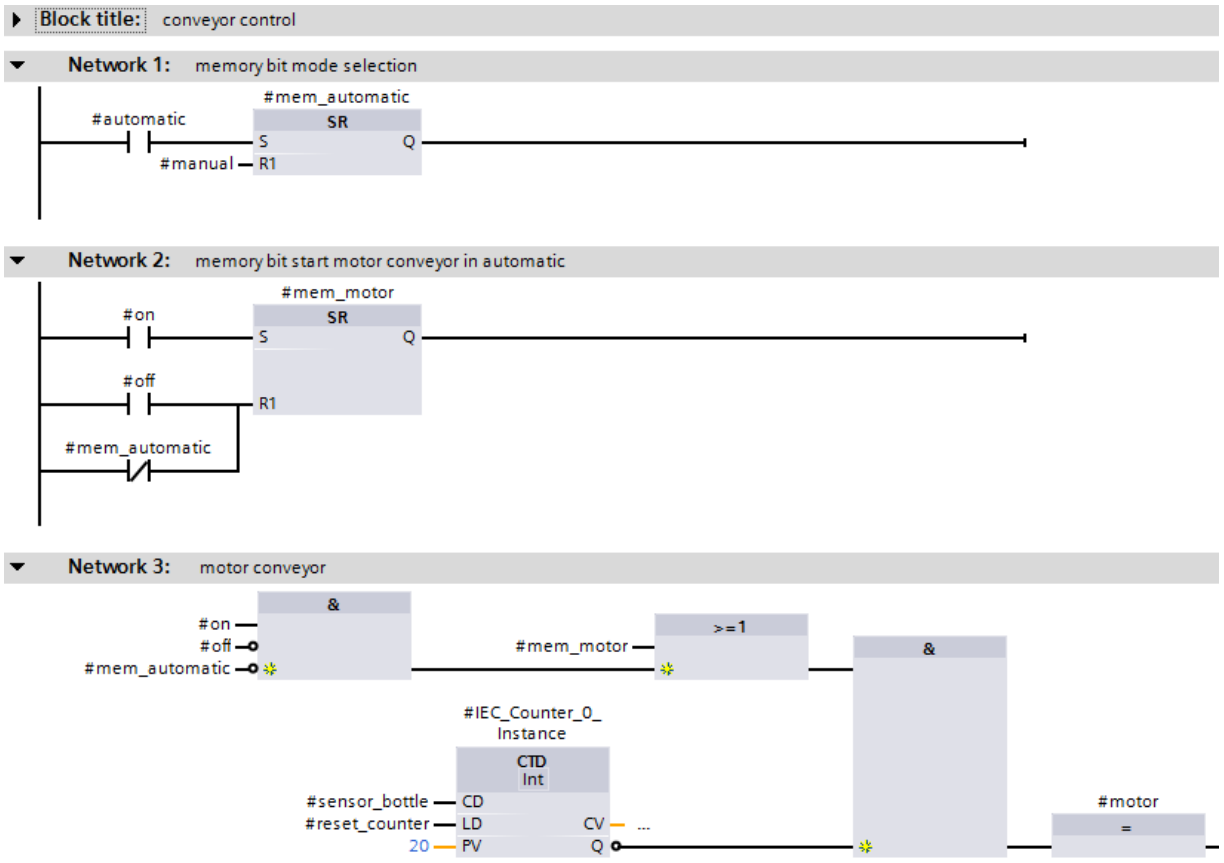
Poznámka

Čítač dolů se nejlépe hodí na počítání určitého množství, protože můžeme jednoduše využít binární výstup **'Q'**. Jinak bychom museli naprogramovat komparátor.

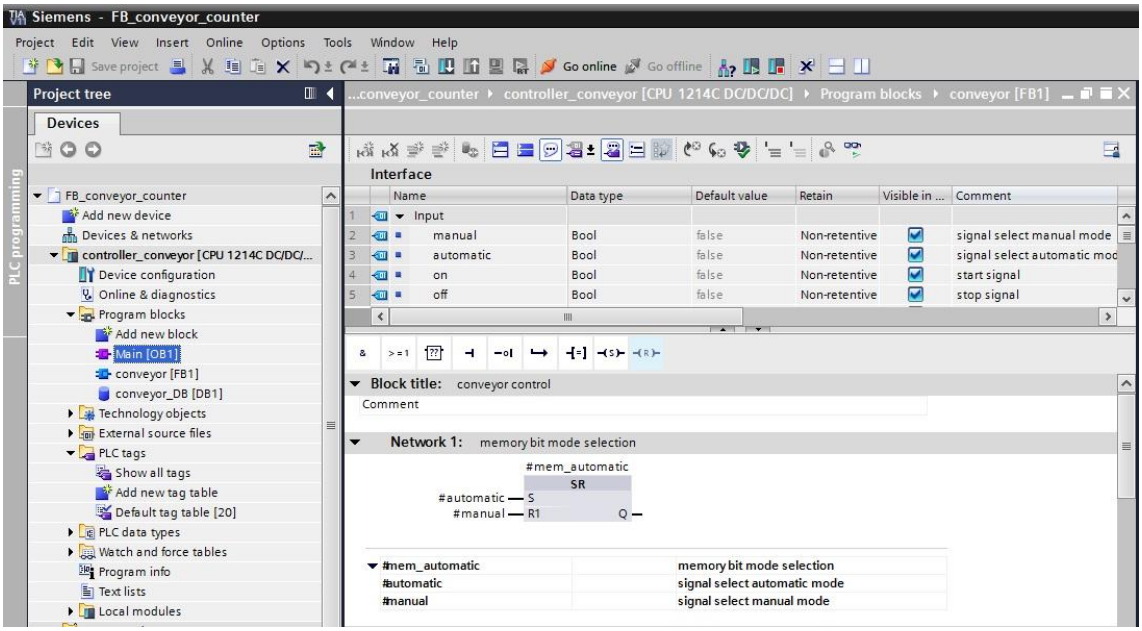
Program ve funkčním blokovém diagramu (FBD)



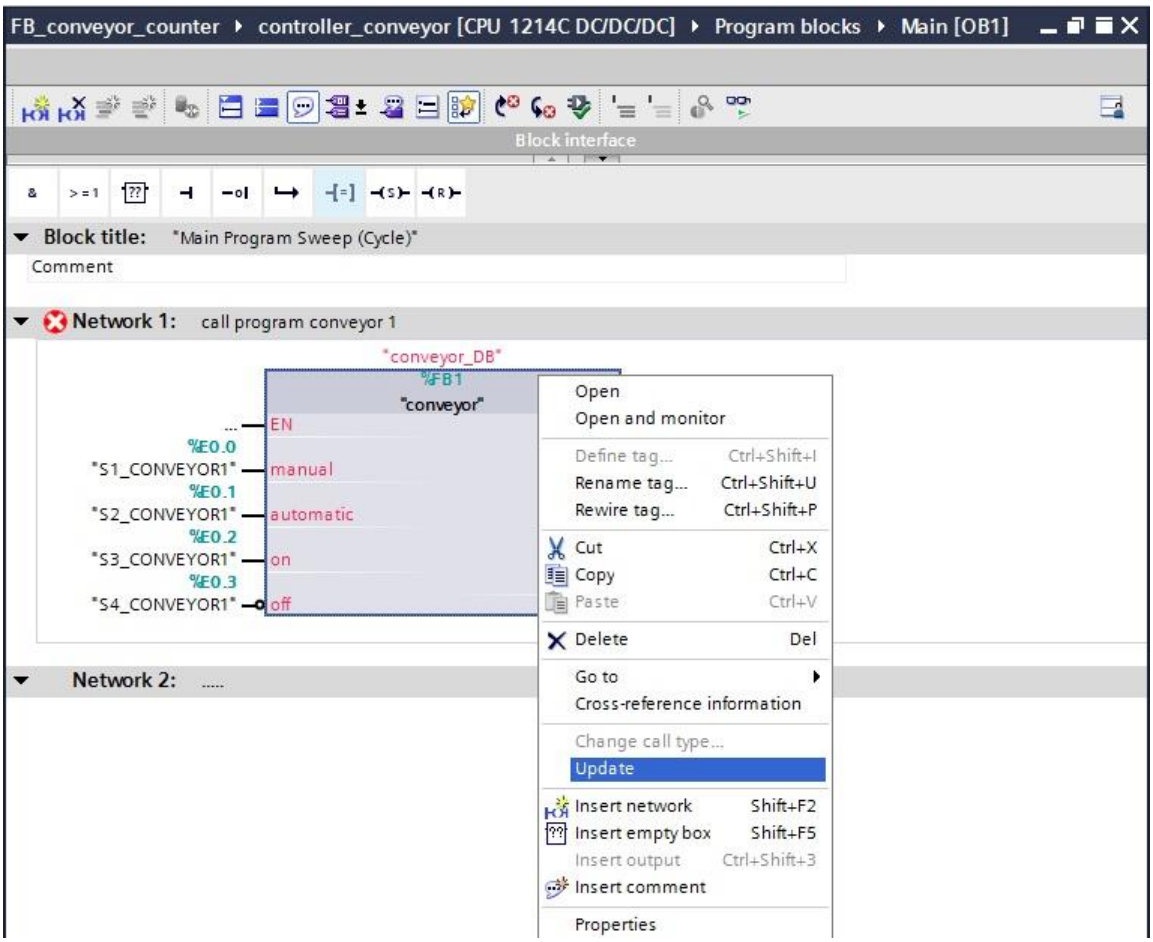
Program v žebříkovém diagramu (LAD)



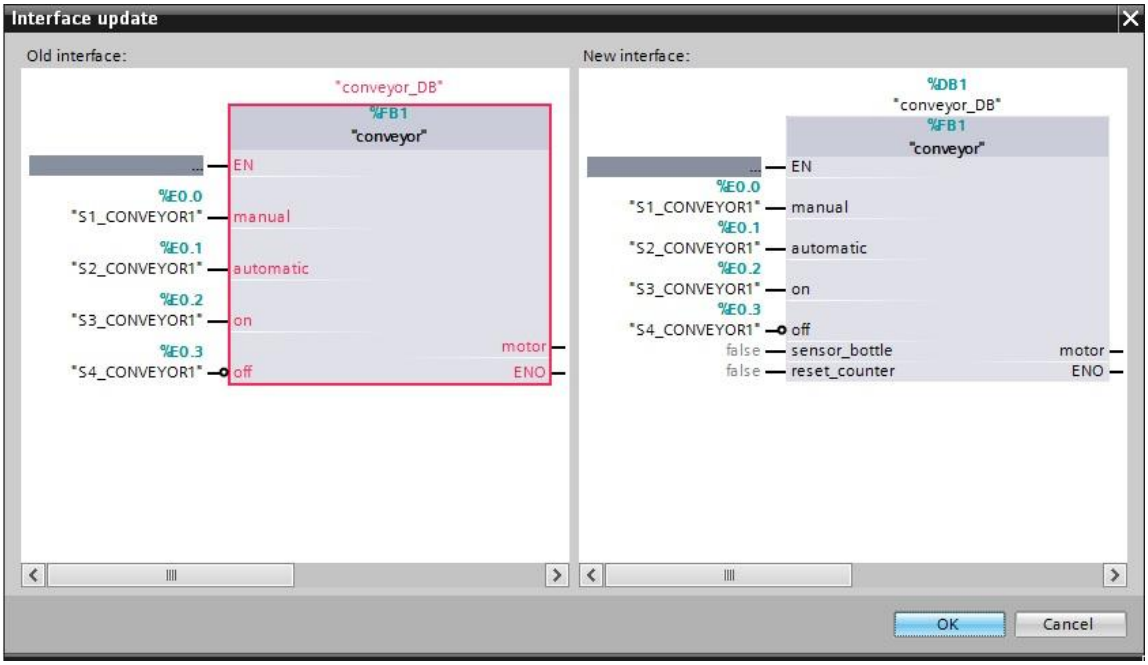
16. Nyní otevřeme blok **'Main[OB1]'** pro aktualizaci volání bloku **'conveyor[FB1]'** (→ Main[OB1])



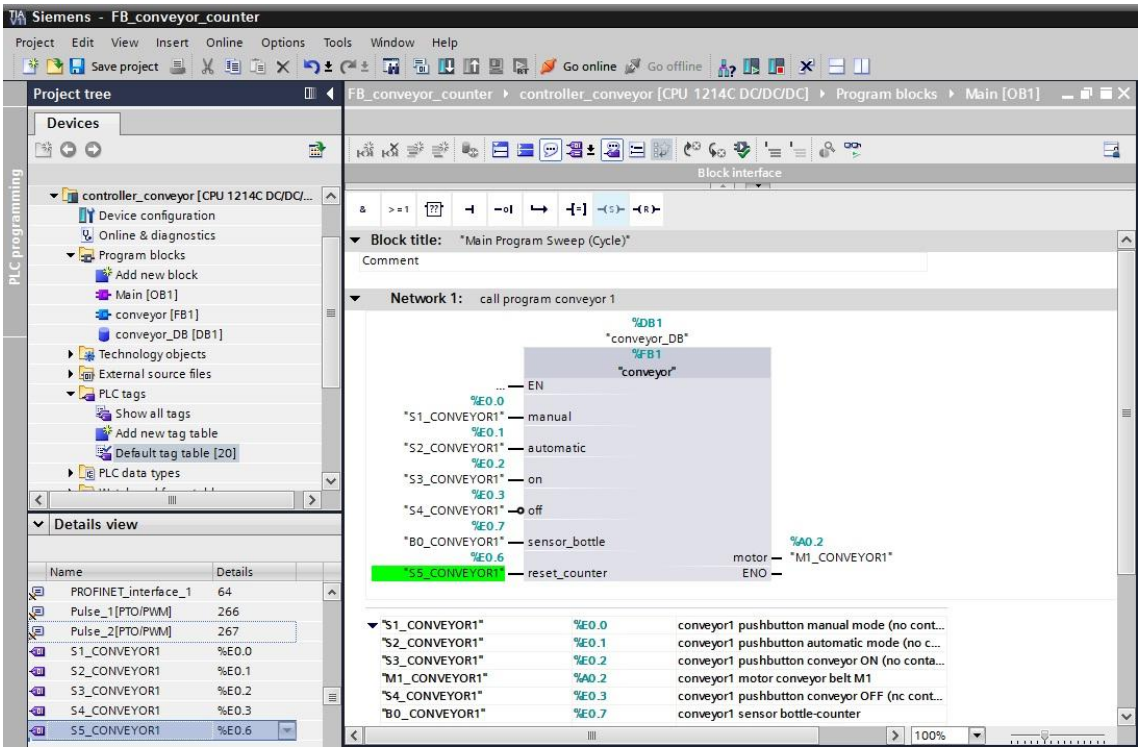
17. V bloku **'Main[OB1]'**, klikneme pravým tlačítkem myši na **"conveyor"** a pak na **'Update'.** (→ Main[OB1] → Update)



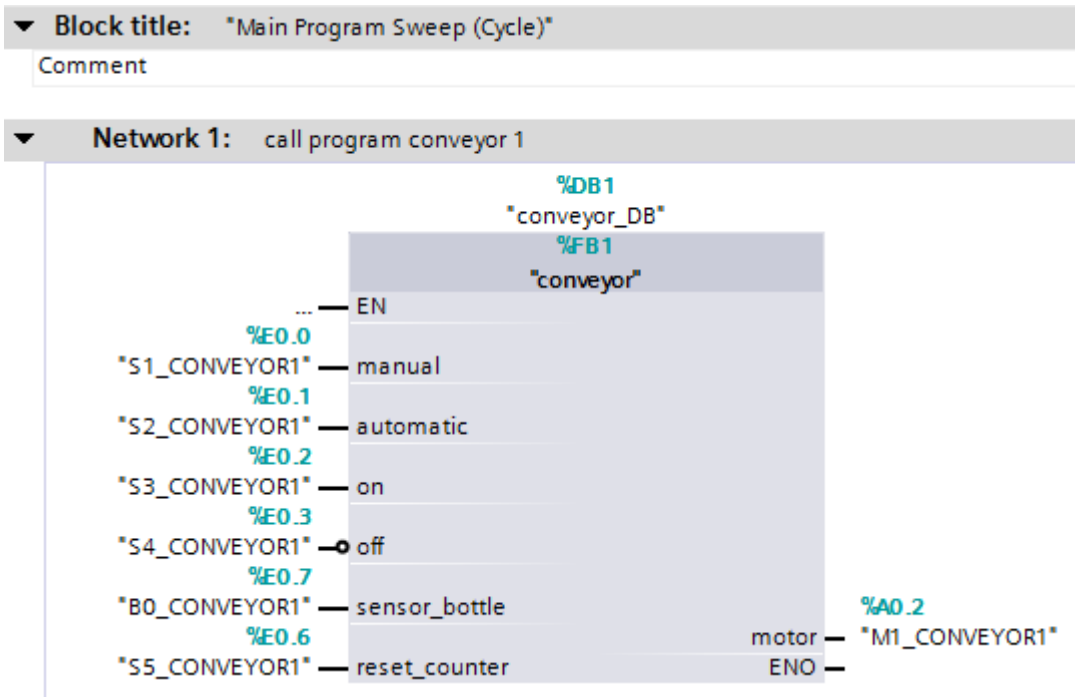
18. Dále vybereme **'New Interface'** a potvrdíme s **'OK'.** (→ New interface → OK)



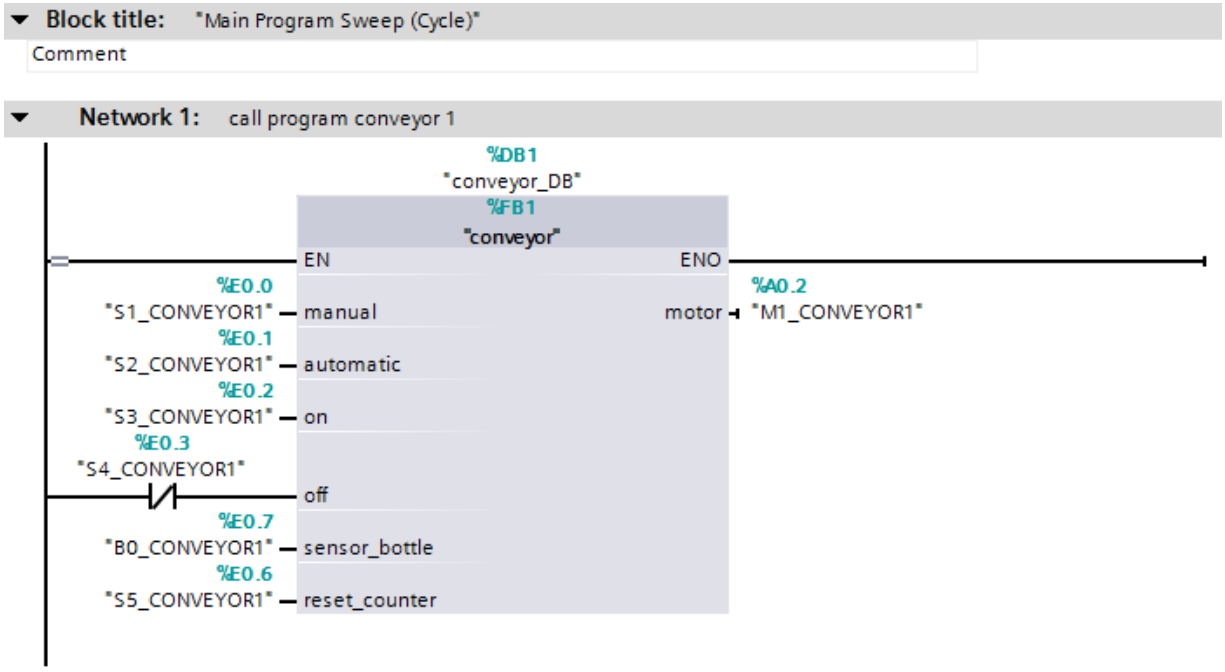
19. Nyní připojíme dvě nové proměnné do PLC tagů “B0” a “S5” jsou zobrazeny. Kliknutím na Save project, uložíme projekt. (→ Save project)



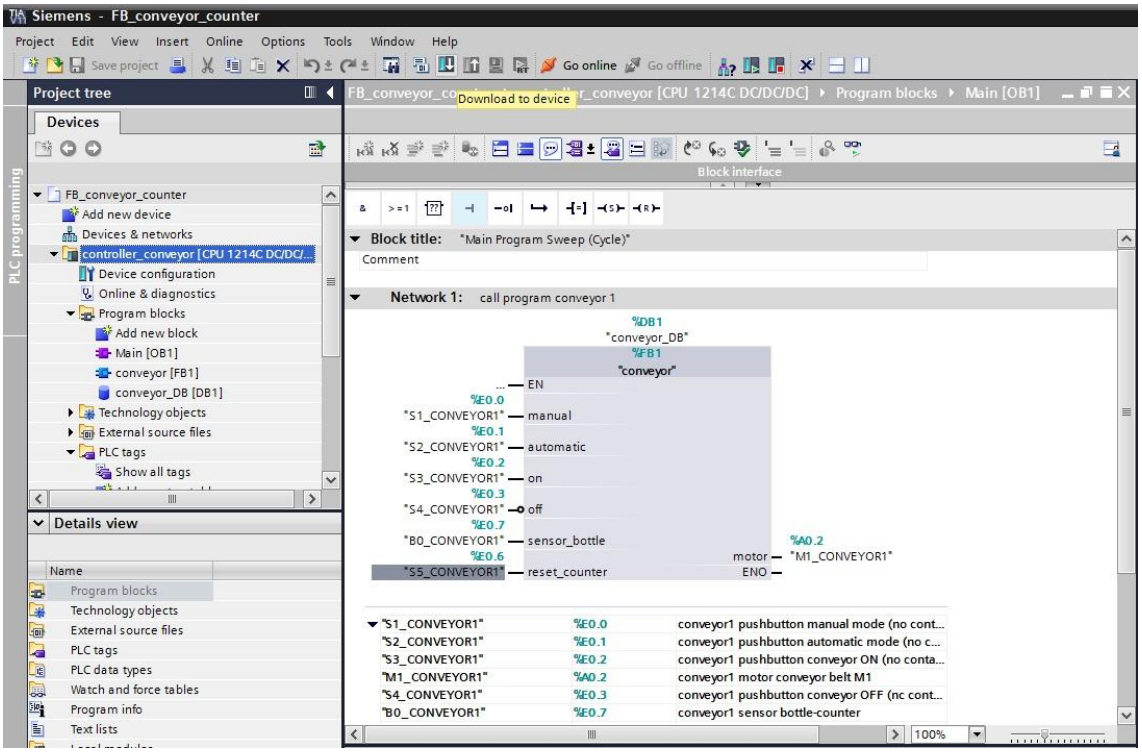
Program ve funkčním blokovém diagramu (FBD)



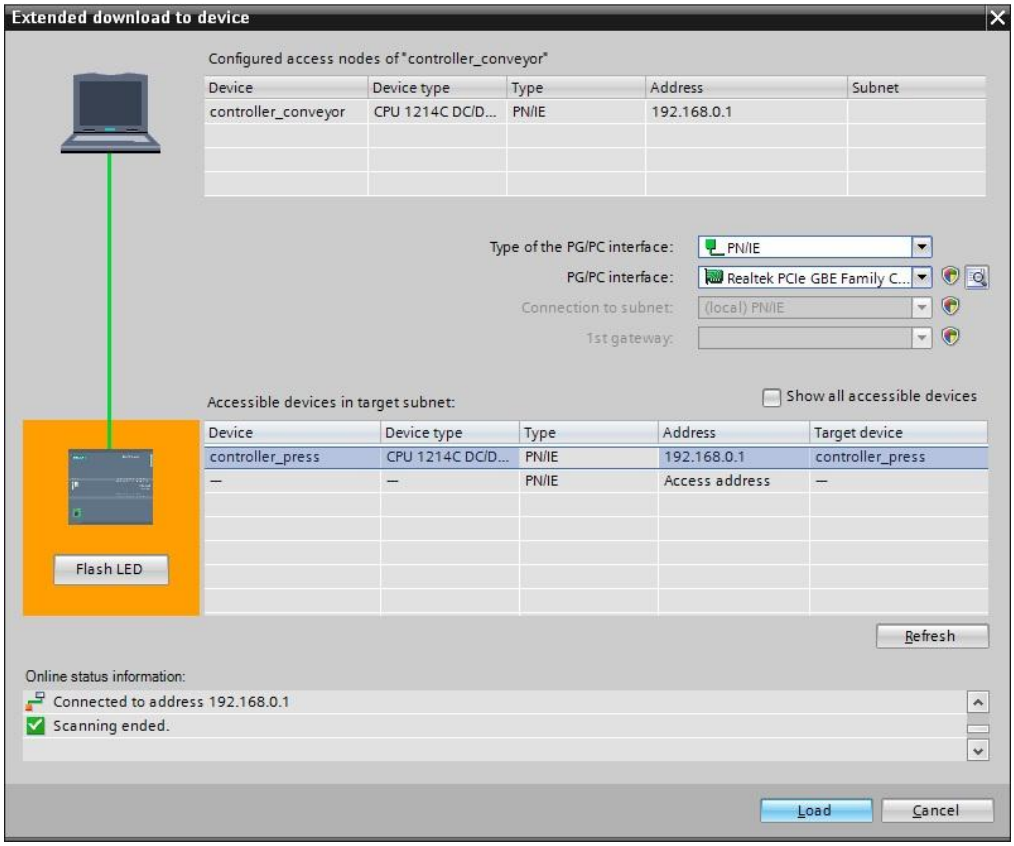
Program v žebříkovém diagramu (LAD)



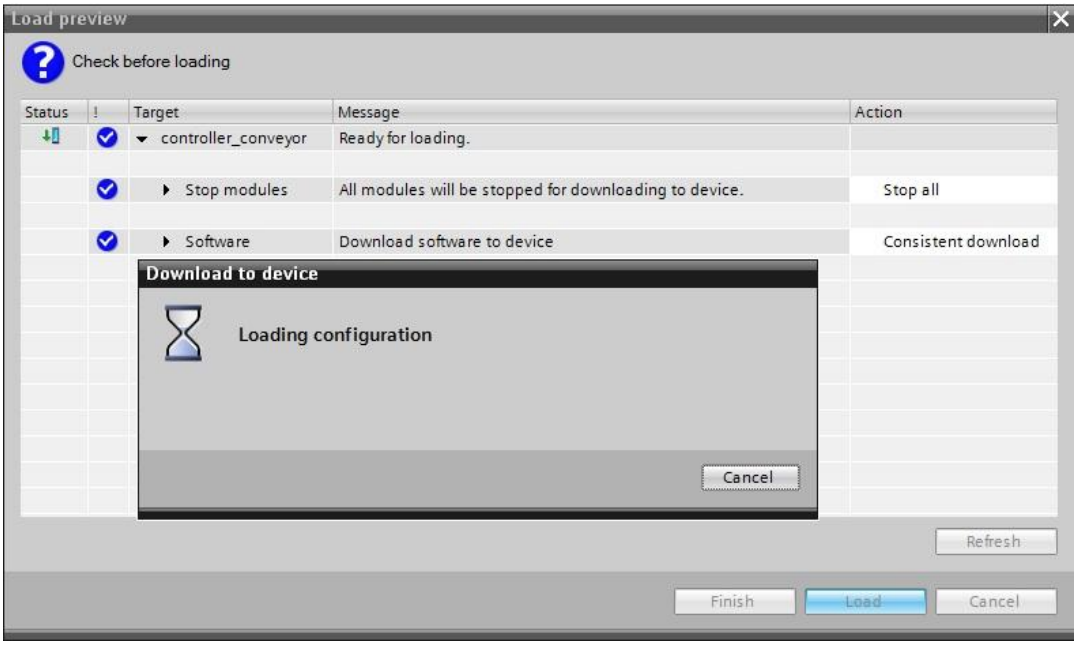
20. Pro nahrání celého programu do CPU, zvýrazníme složku **'controller_conceyor'**, a poté kliknutím na  Nahrajeme do zařízení. (→ controller_conveyor → )



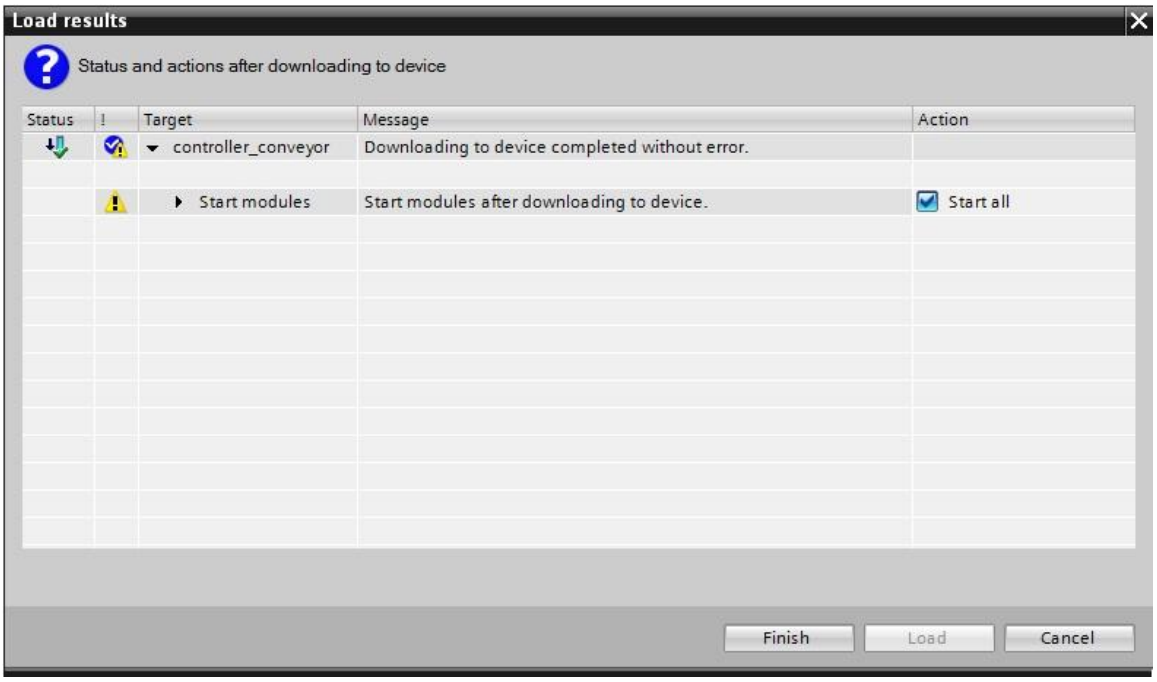
21. Nastavíme rozhraní.



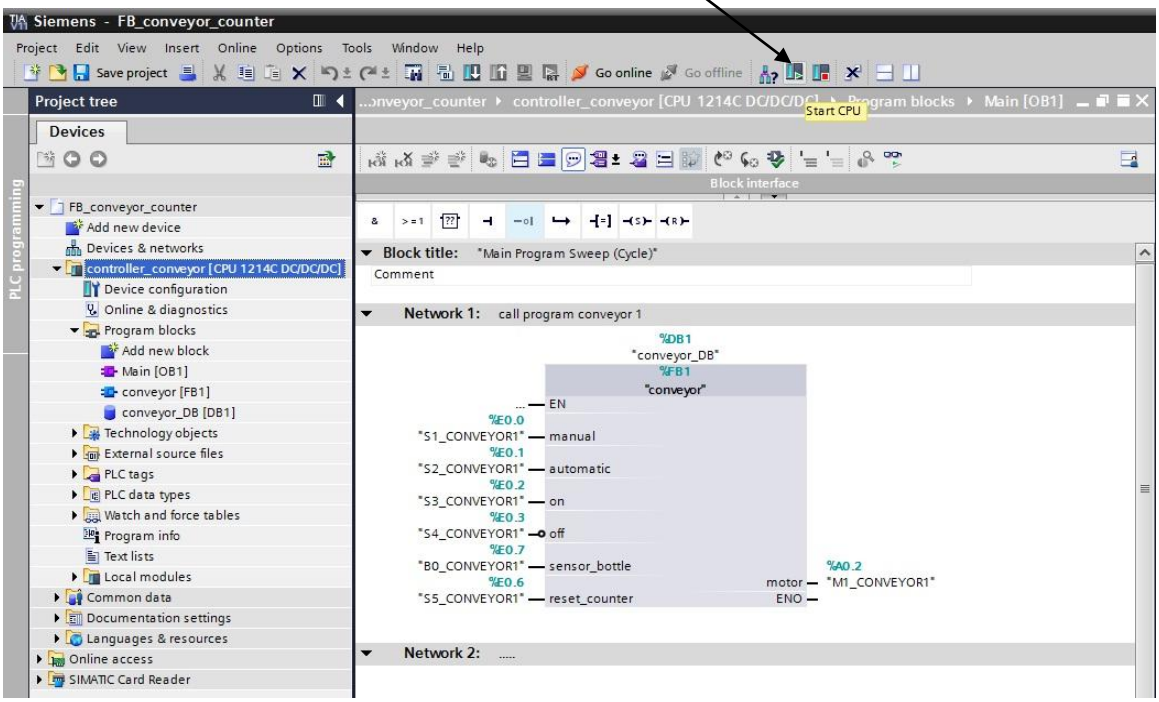
22. Potvrdíme **'Load'** ještě jednou. Během nahrávání se nám v okně zobrazuje status. (→ Load)



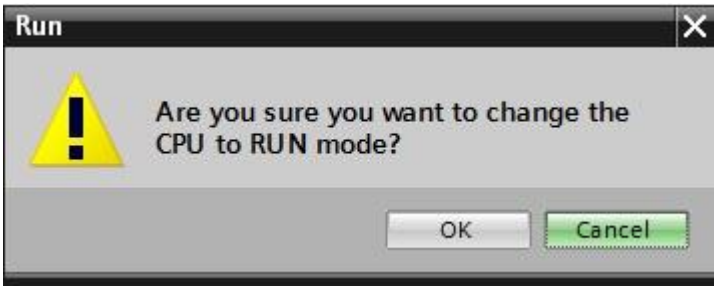
23. Pokud bylo nahrání úspěšné, zobrazí se nám to v okně. Klikneme na **'Finish'.** (→ Finish)



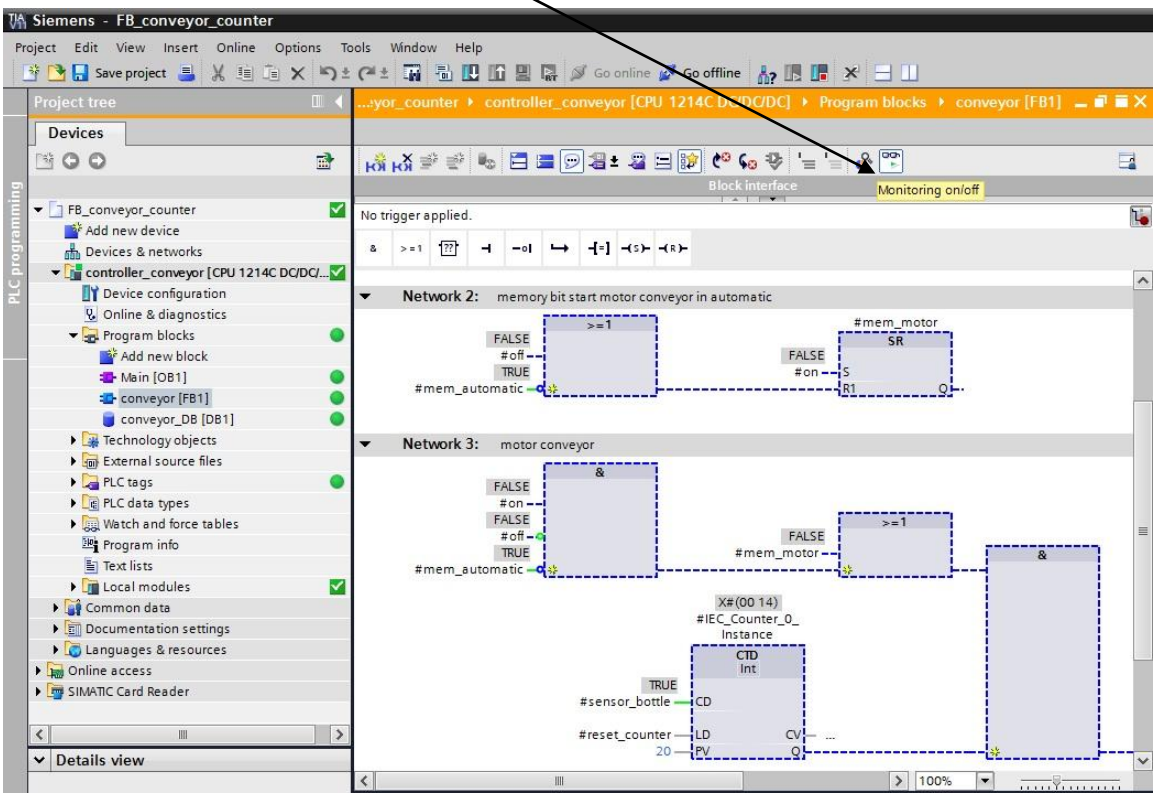
24. Dále spustíme, CPU kliknutím na  .



25. Potvrdíme, že opravdu chceme spustit CPU kliknutím na **'OK'.** (→ OK)



26. Kliknutím na symbol  Monitoring On/Off, můžeme během testování programu sledovat status čítače.



27. Kliknutím na symbol  Monitoring On/Off můžeme sledovat hodnoty otevřeného data bloku během testování.