

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

Siemens Automation Cooperates with Education

TIA Portal Module 010-060 Návrh řízení se 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](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 začínají. 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.

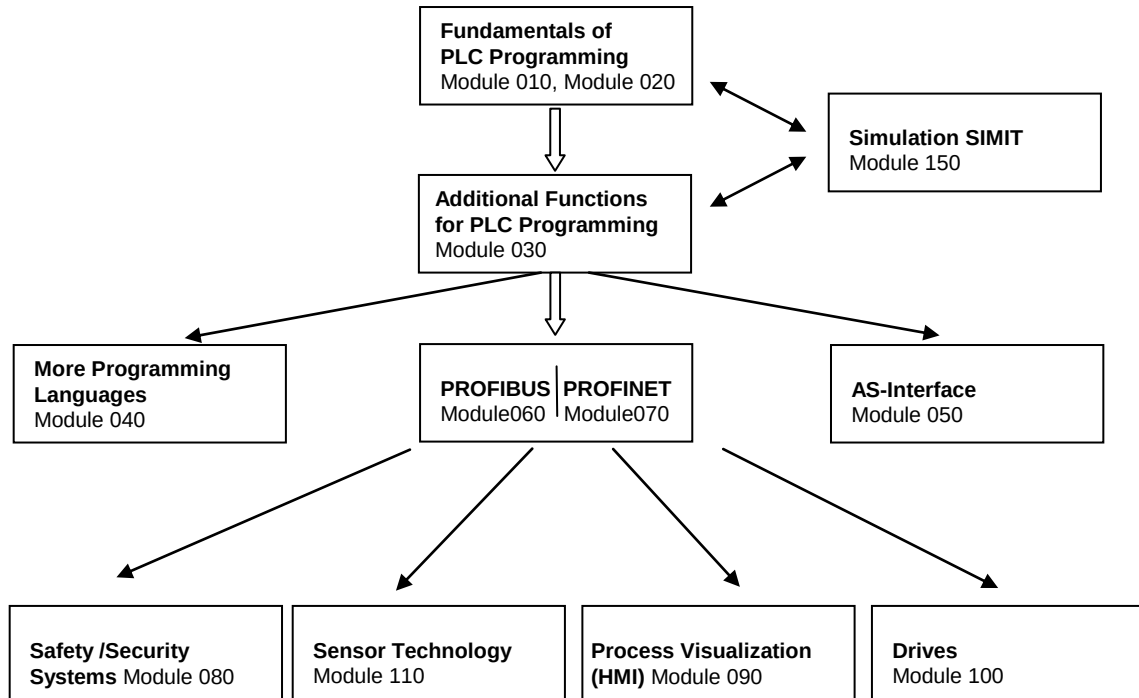
Použití pro kurzy průmyslových zákazníků je přímo zakázáno. Nechceme, aby byl dokument využit komerčně.

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

	PAGE
1. 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. Základy automatického řízení.....	7
3.1 Úlohy automatického řízení.....	7
3.2 Komponenty uzavřené smyčky.....	8
3.3 Jednotkový skok pro vyšetřování řídicích systémů.....	11
3.4 Řízené systémy s kompenzací.....	12
3.4.1 Proporcionální řízení bez zpoždění.....	12
3.4.2 Proporcionální řízení s jedním časovým zpožděním.....	13
3.4.3 Proporcionální řízení se dvěma časovými zpožděními.....	14
3.4.4 Proporcionální řízení s n časovými zpožděními.....	15
3.5 Řídicí systémy bez kompenzace.....	16
3.6 Základní typy spojitých regulátorů.....	17
3.6.1 Proporcionální regulátor (P–Regulátor).....	18
3.6.2 Integrální regulátor (I-regulátor).....	20
3.6.3 PI regulátor.....	21
3.6.4 Diferenciální regulátor (D-regulátor).....	22
3.6.5 PID regulátor.....	22
3.7 Postup nastavení regulátoru.....	23
3.8 Nastavení řídicího systému.....	25
3.8.1 Nastavení PI-regulátoru podle Zieglera-Nicholse.....	26
3.8.2 Nastavení PI regulátoru podle Chien, Hrones a Reswick.....	26
3.9 Digitální řízení.....	28
4. Vzorová úloha – řízení hladiny v nádrži.....	30
5. Programování řízení hladiny pro SIMATIC S7-1200.....	31

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

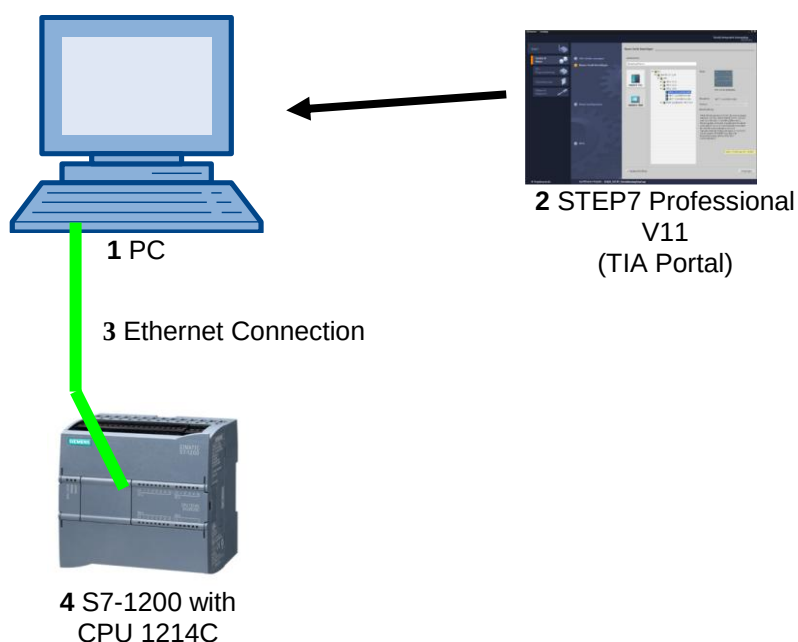
Prerekvizity:

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 V11 (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.

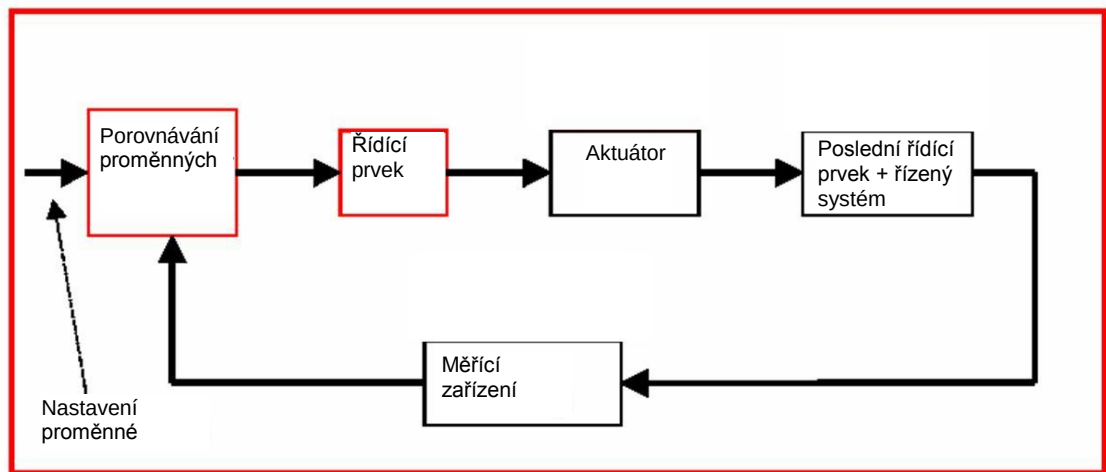
3. Základy automatického řízení

3.1 Úlohy automatického řízení

„Řízení je proces, kdy je hodnota proměnné nastavena a udržována pomocí zásahů závislých na měření této proměnné. Řídicí sekvence probíhá v tzv. uzavřené smyčce, protože proces závisí na měření proměnné a ovlivňuje tak sám sebe.“

Řízená proměnná je měřena spojitě a porovnávána s jinou specifikovanou proměnnou stejného druhu. V závislosti na výsledku tohoto srovnání, řídicí proces upraví hodnotu řízené proměnné na hodnotu specifikované proměnné.

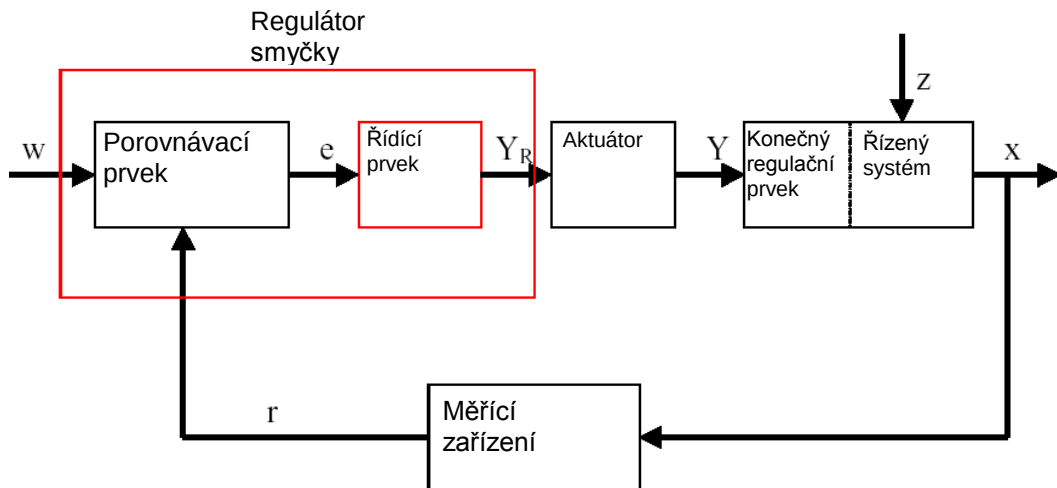
Diagram ukazující řízení v uzavřené smyčce



3.2 Komponenty uzavřené smyčky

Níže si vysvětlíme blíže základní termíny automatického řízení.

Diagram ukazuje přehled:



1. Řízená proměnná x

Ovlivnění této řízené proměnné tak aby byla konstantní je účelem celého systému. V našem případě to bude teplota v místnosti. Současná hodnota řízené proměnné existuje v určitém časovém okamžiku a říkáme jí „skutečná hodnota“

2. Zpětnovazební proměnná r

V uzavřené smyčce je řízená proměnná neustále kontrolována a upravována v případě nečekaných změn. Měřená hodnota poměrná k řízené proměnné je nazývána zpětnovazební proměnnou.

V příkladu „ohříváč“, může korespondovat s měřeným napětím vnitřního teploměru.

3. Porucha z

Porucha je proměnná, která, bez našeho záměru ovlivňuje řízenou veličinu a vzdaluje jí od naší referenční hodnoty. V našem příkladu řízení teploty to může být například venkovní teplota, která ovlivňuje teplotu v místnosti.

4. Referenční hodnota (setpoint) w

Referenční hodnota je ideální hodnota řízené proměnné. Je třeba poznamenat, že při určitých typech řízení, se reference průběžně mění na základě různých podmínek. V našem příkladu je referencí požadovaná teplota v místnosti.

5. Porovnávací prvek

V tomto místě se porovnávají hodnoty reference a změřené hodnoty ze zpětné vazby. Ve většině případů, jsou dvě proměnné měřená napětí. Rozdíl mezi těmito proměnnými se nazývá „**regulační odchylka**“ ta je předána řídicímu prvku a v něm vyhodnocena.

6. Řídící/regulační prvek (regulátor)

Řídící prvek je jádrem celého systému řízení. Vyhodnocuje regulační odchylku, informaci jestli a jak je řízená proměnná odchýlená od reference a na výstup posílá „akční **zásah regulátoru**“ Y_R který ovlivňuje řízenou proměnnou. V příkladu s vytápěním, je akčním zásahem napětí na mixážním motoru. Způsob jakým regulační prvek určuje akční zásah z regulační odchylky je základním kritériem systému. Ve druhé části je toto téma rozebráno dopodrobna.

7. Aktuátor (akční prvek)

Aktuátor je vykonavatelem práce v celém řídicím systému. Z pohledu řídicího systému, poskytuje regulátor informaci, jak má být řízená proměnná upravena. Podle této informace aktuátor zasahuje do hodnoty řídicí proměnné tak aby byla co nejbližší referenci. V našem případě je aktuátorem mixážní motor. V závislosti na napětí poskytnutém regulátorem upravuje polohu směšovacího zařízení.

8. Konečný regulační prvek

Tento prvek uzavřené smyčky více méně přímo ovlivňuje hodnotu řízené proměnné v závislosti na **řídicí proměnné Y**. V našem příkladu, je to kombinace skládající se ze směšovacího zařízení, tepelných kabelů a topení. Mixážní motor nastaví směšovací zařízení, které ovlivní teplotu pokoje pomoví teploty vody.

9. Řízený systém

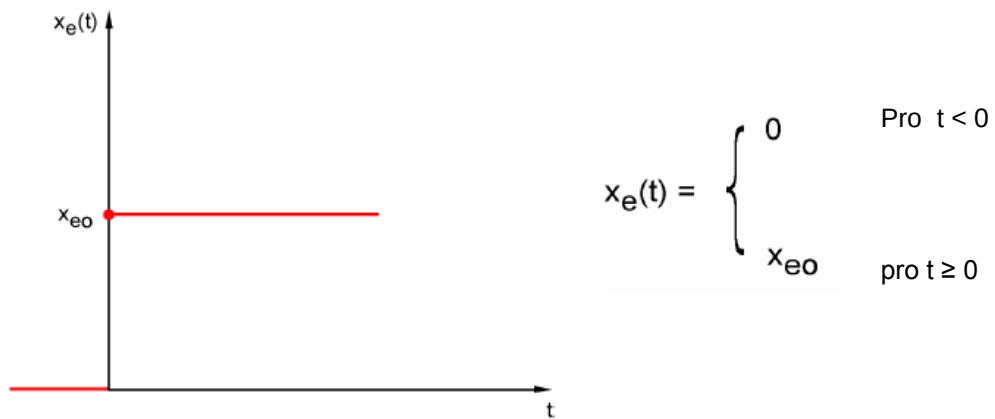
Řízený systém je systém kde se nachází naše řízená proměnná. V našem příkladu s vytápěním je to vytápěný pokoj.

10. Dopravní zpoždění

Dopravním zpožděním se rozumí čas, za který se projeví změna provedená regulátorem na regulované veličině. V našem případě je to čas mezi změnou napětí na mixážním motoru a měřitelnou změnou teploty místnosti.

3.3 Jednotkový skok pro vyšetřování řídicích systémů

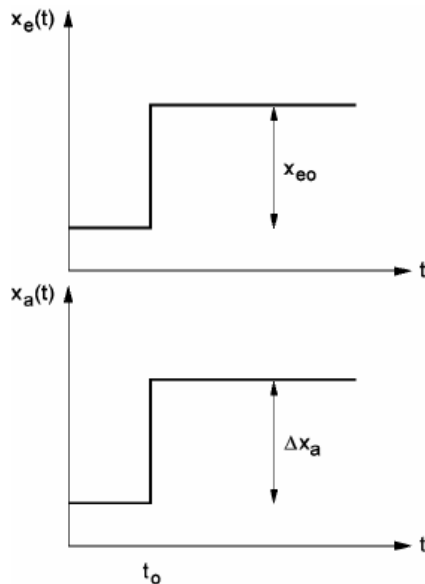
Pro vyšetření chování řízeného systému, regulátoru a řídicí smyčky, je používána jako vstupní signál unifikovaná funkce; jednotkový skok.



3.4 Řízené systémy s kompenzací

3.4.1 Proporcionální řízení bez zpoždění

Zkráceně P-regulace.



Náhlá změna vstupní proměnné v čase t_0

Řízená proměnná / řídící proměnná:

$$x = K_{ss} \cdot y$$

K_{ss} : Zesílení, nebo také proporcionální konstanta P-regulátoru

$$K_{ss} = \frac{\Delta x}{\Delta y} = \tan \alpha$$

Řízená proměnná / poruchou:

$$x = K_{sz} \cdot z$$

K_{sz} : Hodnota zesílení pro změnu vlivu poruchy.

Rozsah:

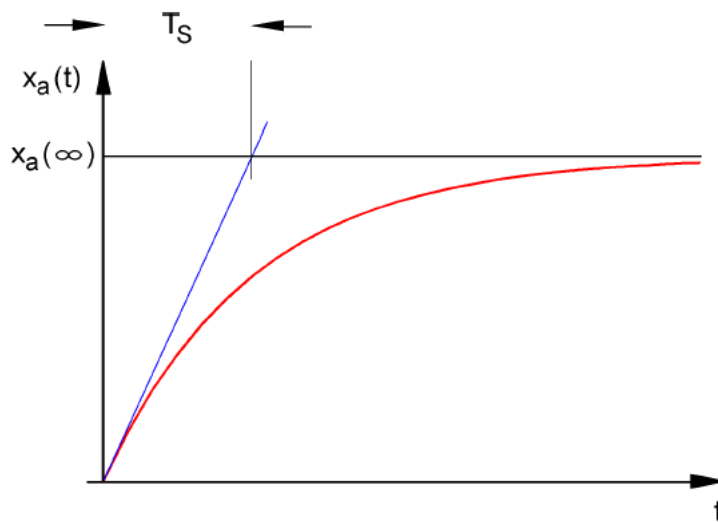
$$y_h = y_{\max} - y_{\min}$$

Řízený rozsah:

$$x_h = x_{\max} - x_{\min}$$

3.4.2 Proporcionální řízení s jedním časovým zpožděním

Systém nazýváme P-T1.



Graf: skoková odezva pro P-T1

Diferenciální rovnice pro obecný výstupní signál

$x_e(t)$:

$$T_S \cdot \dot{x}_a(t) + x_a(t) = K_{PS} \cdot x_e(t)$$

Řešení diferenciální rovnice pro jednotkový skok na vstupu (skoková odezva)

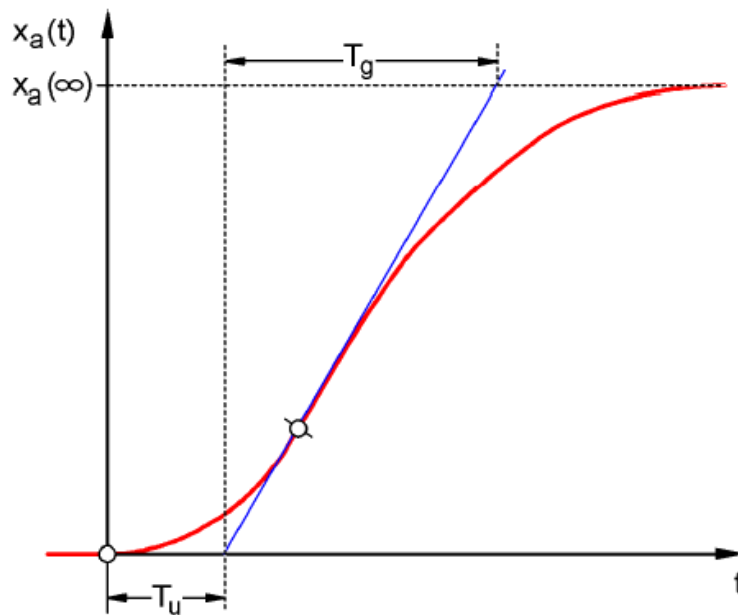
$$x_a(t) = K_{PS} (1 - e^{-t/T_S}) \cdot x_{e0}$$

$$x_a(t = \infty) = K_{PS} \cdot x_{e0}$$

T_S : časová konstanta

3.4.3 Proporcionální řízení se dvěma časovými zpožděními

Systém označujeme jako P-T2.



Graf: skoková odezva pro P-T2

T_u: Dopravní zpoždění T_g: Compensation time

Systém je vytvořen sériovým spojením dvou P-T1 systémů a má časové konstanty TS1 a TS2

Řiditelnost P-Tn systému:

$$\frac{T_u}{T_g} < \frac{1}{10} \rightarrow \boxed{\text{Jednoduše řiditelný}} \quad \frac{T_u}{T_g} \approx \frac{1}{6} \rightarrow \boxed{\text{Stále řiditelný}} \quad \frac{T_u}{T_g} > \frac{1}{3} \rightarrow \boxed{\text{Obtížně řiditelný}}$$

S rostoucím poměrem T_u/T_g , se systém stává méně a méně řiditelným.

3.4.4 Proporcionální řízení s n časovými zpožděními

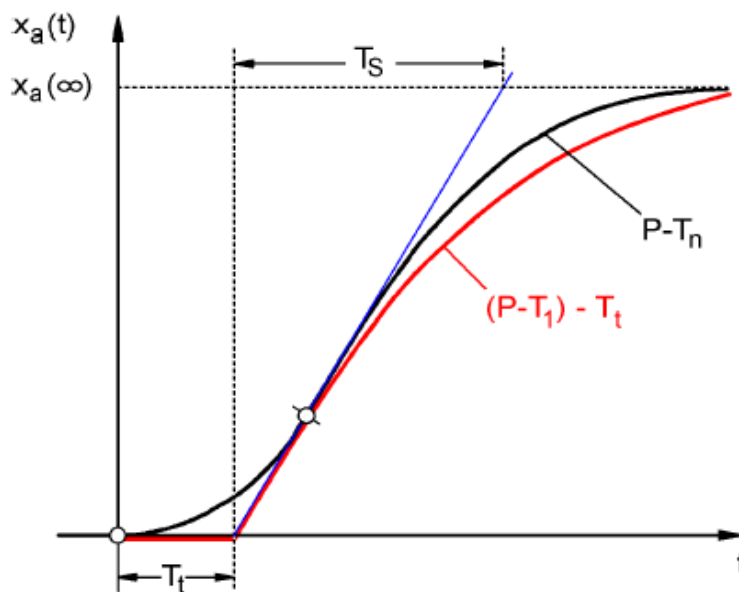
Systém nazýváme P-Tn.

Časová odezva je popsána diferenciální rovnicí n-tého stupně.

Charakteristika skokové odezvy je podobná P-T2 systému. Časová odezva je popsána pomocí T_u a T_g .

Substituce: Systém s mnoha zpožděními můžeme aproximovat pomocí sériového spojení P-T1 systémů.

Platí-li následující: $T_t \gg T_u$ a $T_S \gg T_g$.

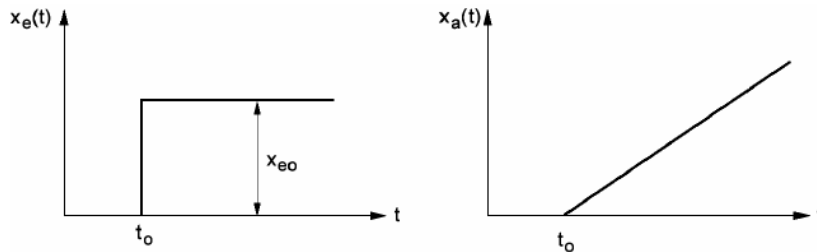


Substituovaná skoková odezva pro P-Tn systém.

3.5 Řídicí systémy bez kompenzace

Po poruše roste řízená proměnná dál bez tlumení a bez dosažení nějaké konečné hodnoty.

.



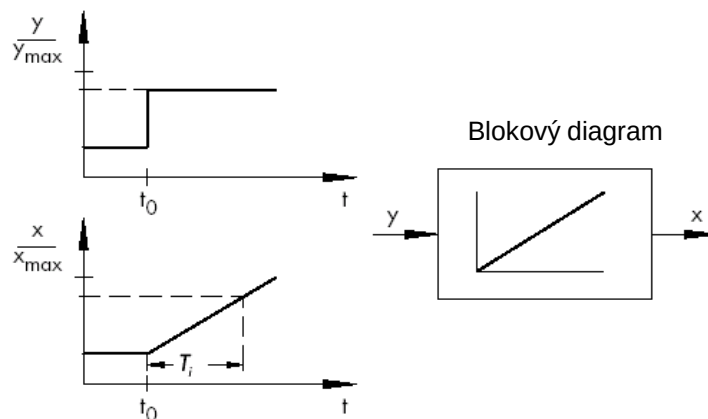
Příklad: Řízení výšky hladiny

Nádrž s výpustí, která má stejný přítok i odtok dosáhne konstantní hladiny. Pokud manipulujeme s velikostí přítoku nebo odtoku, hladině poroste respektive klesne. Hladina se mění rychleji čím větší je rozdíl mezi přítokem a odtokem.

Příklad to názorně ukazuje, integrální vstup má obvykle limit. Řízená proměnná roste a klesá, dokud nedosáhne limitu v systému: nádrž přeteče nebo je prázdná, tlak dosáhne v soustavě minima nebo maxima apod.

Obrázky ukazují časové chování I-systému, jehož vstup se náhle mění, diagram je z toho odvozen.

.



Pokud skoková funkce přejde na vsup funkce $x_e(t)$, pak

$$x_a(t) = K_{IS} \int x_e(t) dt \Rightarrow \text{Integrační řídicí systém}$$

K_{IS} : Integrační koeficient řídicího systému

* Obrázek převzatý z SAMSON Technical Information - L102 – Controllers and Controlled Systems, Edition August 2000

(http://www.samson.de/pdf_en/l102en.pdf)

3.6 Základní typy spojitých regulátorů

Regulátory zmíněné výše mají jak již bylo řečeno výhodu v jednoduchosti. Regulátor, aktuátor a konečný regulační prvek jsou jednoduché a tedy levnější než spojitý regulátor. Nicméně, diskrétní regulátory mají své nevýhody, například pokud spouštíme velké zátěže jako elektromotory nebo chladicí jednotky, špičková zátěž může poškodit zdroje.

Z toho důvodu nepřepínáme mezi „On“ a „Off“, ale mezi plnou zátěží a znatelně nižším výkonem aktuátoru/konečného regulačního prvku (základním výkonem). Ale i s tímto vylepšením, diskrétní řízení není vhodné pro všechny aplikace. Představte si motor automobilu, jehož rychlost je řízena přímo. Nebylo by nic jiného než plný plyn nebo neutrál.

Navíc je nemožné přenášet plný výkon motoru okamžitě pomocí pneumatik na vozovku, takové auto by bylo velice nevhodné do reálného provozu. Z toho důvodu používáme spojitě regulátory.

V tomto případě, matematický vztah, který vytváří regulátor mezi odchylkou systému a svým výstupem má teoreticky neomezený rozsah. Prakticky nicméně rozlišujeme mezi třemi základními typy, které si představíme níže.

3.6.1 Proporcionální regulátor (P-Regulátor)

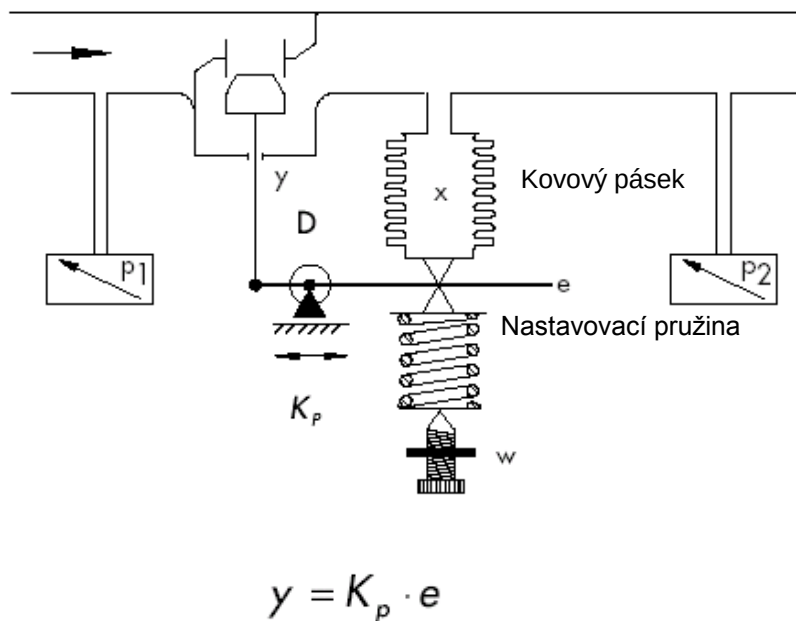
V případě P-regulátoru, řídicí proměnná y je vždy proporční k zaznamenané odchylce ($y \sim e$).

Výsledkem je, že P-regulátor reaguje na poruchu bez zpoždění a jen tehdy když je v systému porucha e .

Proporcionální regulátor tlaku na obrázku níže, srovnává sílu FS nastavovací pružiny se silou FB která vytváří tlak p_2 v elastickém kovovém pásku pod sebou když nejsou síly v rovnováze, vahadlo se otáčí kolem bodu D. Mění se tím poloha ventilu, současně řídíme velikost tlaku p_2 tak aby se opět vytvořila rovnováha sil.

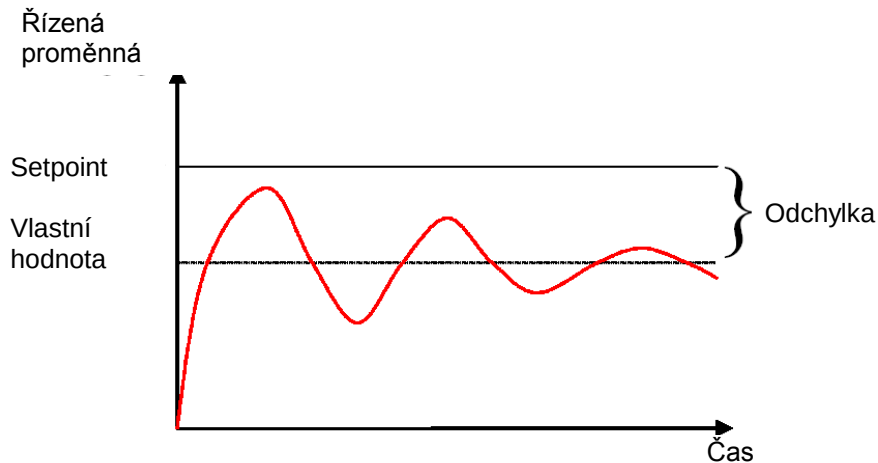
Obrázek ukazuje chování P-regulátoru, když se objeví náhlá porucha. Amplituda řídicí proměnné y záleží na hodnotě řídicí odchylka (poruchy) e a velikosti proporčního koeficientu K_p :

Pro udržení nízké odchylky, proporční faktor musí být co největší. Zvětšení faktoru způsobí zrychlení akčního zásahu regulátoru. Nicméně, moc vysoká hodnota způsobí překmit a kmitání části regulátoru.



* Obrázek a text z SAMSON Technical Information - L102 – Controllers and Controlled Systems, Edition: August 2000 (http://www.samson.de/pdf_en/l102en.pdf)

Diagram ukazuje chování P-regulátoru



Ve srovnání s jinými typy regulátorů, výhodou tohoto typu je jednoduchost (v nejjednodušším případě, elektronické provedení obsahuje jen rezistor) a okamžitá odezva.

Hlavní nevýhodou P-regulátoru je neustálá regulační odchylka, i během delšího časového úseku, nikdy nedosáhne přesné hodnoty reference. Tato nevýhoda a také rychlost odezvy, která ještě není ideální a může být minimalizována pouze nedostatečně velkým P-faktorem, čím způsobíme překmit regulátoru.

V tom nejhorším případě dosáhneme toho, že regulátor bude neustále oscilovat, to vzdaluje řízenou proměnnou od reference, ne ovlivněním hodnoty ale regulátorem samotným.

Problém neustálé regulační odchylky je řešen integrálním regulátorem.

3.6.2 Integrální regulátor (I-regulátor)

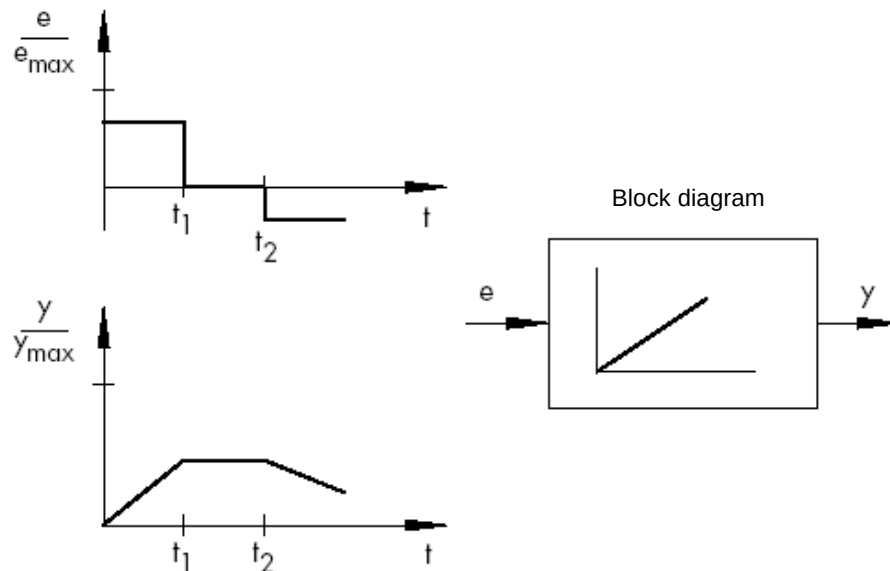
Integrální regulátor se využívá pro úplné odstranění regulační odchylky v každém pracovním bodě. Tak dlouho dokud není odchylka nulová se mění hodnota řídicí proměnné. Pouze pokud jsou reference a řízená proměnná shodné nebo když dosáhne řídicí proměnná systémových limitů, je regulátor v rovnovážném stavu.

Matematická formulace je následující:

Řídicí proměnná je proporcionální k integrálu za čas regulační odchylky e :

$$y = K_i \int e \, dt \quad \text{s:} \quad K_i = \frac{1}{T_n}$$

Jak rychle řídicí proměnná reaguje, záleží na regulační odchylce a iteračním čase:

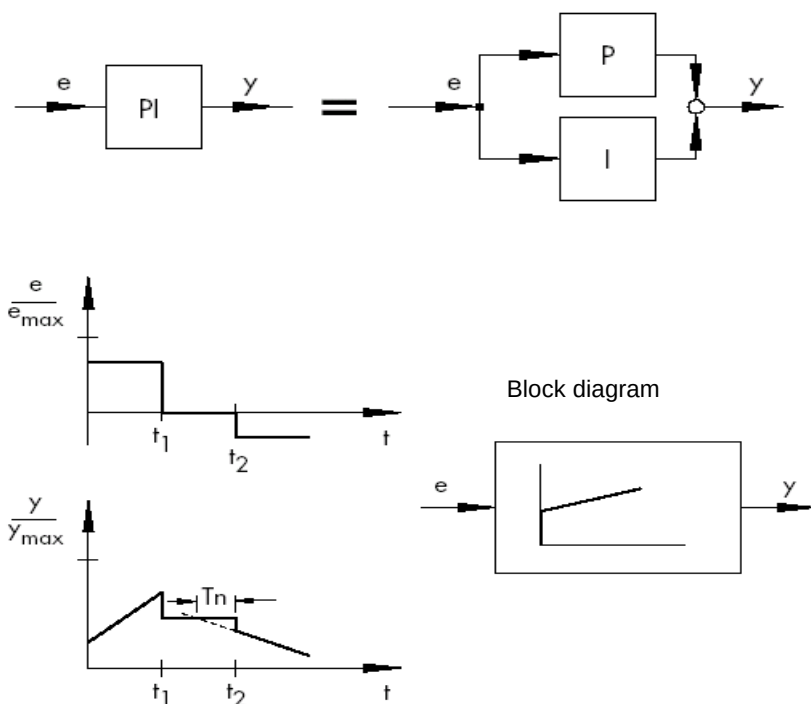


* Figure and text from SAMSON Technical Information - L102 – Controllers and Control Systems, Edition: August 2000 (http://www.samson.de/pdf_en/l102en.pdf)

3.6.3 PI regulátor

PI regulátor je typ nejčastěji využívaný v praxi. Je výsledkem paralelního spojení P a I regulátorů.

Pokud je nalaďen správně bere si výhody obou typů (stabilní, rychlý a bez odchylky v ustáleném stavu) takže jejich nevýhody jsou vzájemně kompenzovány.



Chování s ohledem na čas je upraveno pomocí koeficientu K_p a integračního času T_n . Protože proporcionální část hýbe s řídicí proměnnou okamžitě, když dojde k změně odchylky e , zatímco integrální část s ní manipuluje v průběhu času. T_n představuje čas jaký uběhne do doby než I komponent dosáhne stejné amplitudy zásahu, která je vytvořena okamžitě P-komponentem (K_p). Tak jako pro I-regulátor, integrační čas T_n se musí zkrátit pokud chceme aby vzrostl integrační komponent.

Návrh regulátoru:

V závislosti na K_p a T_n nastavení, překmit řízené proměnné může být zmenšen za cenu snížení dynamiky systému.

Oblast využití PI regulátorů je rychlé zpětnovazební řízení, které nedovoluje odchylku v ustáleném stavu. Například tlak, teplota nebo řízení poměru.

* Figure and text from SAMSON Technical Information - L102 – Controller and Control Systems, Edition: August 2000 (http://www.samson.de/pdf_en/l102en.pdf)

3.6.4 Diferenciální regulátor (D-regulátor)

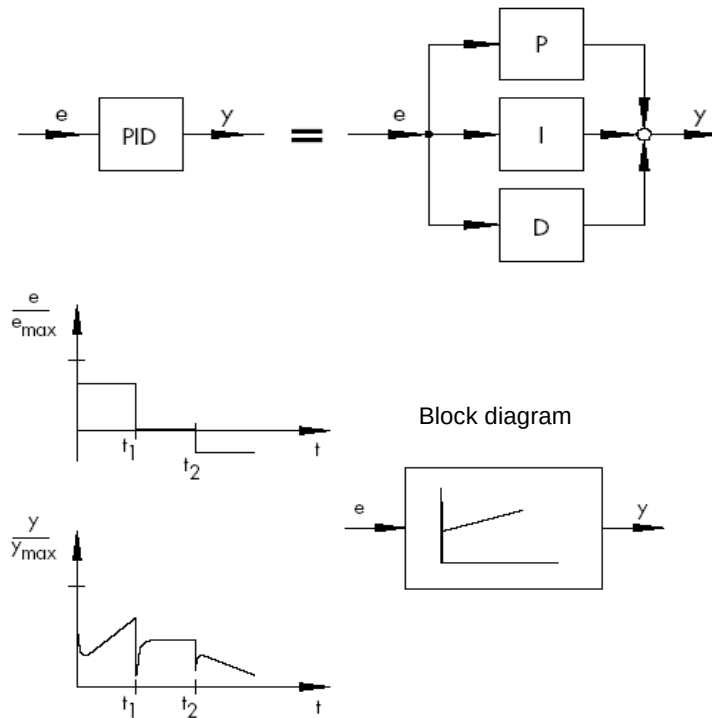
D-regulátor vytváří řídicí proměnnou z časové derivace kontrolní odchylky a ne jako P-regulátor z její amplitudy. Z toho důvodu reaguje znatelně rychleji než P-regulátor. Dokonce, i když je odchylka malá, vytváří (předpověď na základě toho jak se systém doteď choval) velké amplitudy zásahu ihned když se projeví změna amplitudy.

Nicméně D-regulátor nerozpozná odchylku v ustáleném stavu, protože bez ohledu na to jak velká je její časová derivace je nulová. Z tohoto důvodu D-regulátor sám je v praxi skoro nevyužíván. Spíše je využit ve spojení s ostatními typy většinou ve spojení s proporcionálním prvkem.

3.6.5 PID regulátor

Pokud rozšíříme PI-regulátor o D komponentu, získáme univerzální PID regulátor.

Tak jako v případě PD regulátoru, přidání D komponenty nám způsobí, že pokud je D komponenta správně naladěná dosáhneme setpoint mnohem rychleji a ustálený stav bude navozen dříve.



$$y = K_p \cdot e + K_i \int e dt + K_D \frac{de}{dt} \quad \text{with} \quad K_i = \frac{K_p}{T_n}; \quad K_D = K_p \cdot T_v$$

* Figure and text from SAMSON Technical Information - L102 – Controllers and Control Systems, Edition: August 2000 (http://www.samson.de/pdf_en/l102en.pdf)

3.7 Postup nastavení regulátoru

Pro uspokojivé výsledky řízení musíme správně vybrat typ regulátoru. Ještě mnohem důležitější ale je správné nastavení parametrů K_p , T_n a T_v které musí být vyladěny na chování řízeného systému. Většinou musíme balancovat mezi stabilním, ale pomalým řízením a dynamickým ale nestabilním a možná oscilujícím řízením.

V případě nelineárních systémů, které musí například vždy pracovat ve stejném pracovním bodě, musíme nastavit parametry na chování systému v tomto konkrétním bodě. Pokud jako v případě kaskádního řízení není žádný pevný pracovní bod, nastavení regulátoru musí poskytovat dostatečnou stabilitu a rychlost na celé pracovní oblasti.

V praxi, regulátory normálně nastavujeme na základě empirických hodnot. Pokud nejsou k dostání, musíme analyzovat chování řízeného systému, abychom pomocí teoretických a praktických procedur specifikovali hodnoty parametrů.

Jedním z možných způsobů je na oscilaci založená metoda podle Zieglera a Nicholse. Poskytuje jednoduchý a v mnoha způsobech dostatečný postup pro ladění regulátorů. Nicméně, tato metoda lze použít jen pro systémy, které dovolí řízené proměnné autonomně oscilovat. Metoda se používá následně:

- Nastavíme K_p a T_v regulátoru na nejnižší možnou hodnotu a T_n na nejvyšší (nejnižší efekt jaký můžou v systému mít).
- Dostaneme systém manuálně do požadovaného pracovního bodu.
- Nastavíme řídicí proměnnou ručně na specifikovanou hodnotu a zapneme automatický mód.
- Zvyšováním K_p (Snižováním X_p) dosáhneme harmonických oscilací systému. Pokud je to možné, vyvoláme oscilace v kontrolní smyčce pomocí malých náhlých změn setpointu.
- Uložíme si hodnotu K_p jako kritický koeficient $K_{p,krit}$
- Odhadneme trvání celé oscilace jako T_{krit} , pokud je třeba tak jako aritmetický průměr několika oscilací.
- Vynásobíme $K_{p,krit}$ a T_{krit} násobky podle tabulky níže a nastavíme hodnoty zjištěné pro K_p , T_n a T_v do regulátoru.

* Text from SAMSON Technical Information - L102 – Controllers and Controlled Systems, Edition: August 2000 (http://www.samson.de/pdf_en/l102en.pdf)

	K_p	T_n	T_v
P	$0,50 \cdot K_{p, knř.}$	-	-
PI	$0,45 \cdot K_{p, knř.}$	$0,85 \cdot T_{knř.}$	-
PID	$0,59 \cdot K_{p, knř.}$	$0,50 \cdot T_{knř.}$	$0,12 \cdot T_{knř.}$

* Figure and text from SAMSON Technical Information - L102 – Controllers and Control Systems, Edition: August 2000 (http://www.samson.de/pdf_en/l102en.pdf)

3.8 Nastavení řídicího systému

Pro nastavení řídicího systému, například PT2 použijeme.

T_u - T_g Aproximaci

Základem pro metody podle Zieglera- Nicholse a Chien, Hrones-Reswicka je T_u - T_g Aproximace, kde ze skokové odezvy systému, získáme pomocí přenesení parametru hodnoty K_S , T_u a T_g .

P - T_N systémy můžeme popsat dostatečně přesně pomocí T_u - T_g aproximace; tj. aproximací na P - T_1 - T_L systém.

Začátkem je skoková odezva systému na vstupní skok o výšce K . Požadovaný přenosový koeficient systému K_S , časové zpoždění T_u a čas zotavení T_g jsou zjištěny, jak je ukázáno níže.

Změřením přechodové funkce až do ustálené hodnoty ($K \cdot K_S$) je nezbytné, abychom mohli určit hodnotu přenosového koeficientu systému K_S .

Podstatnou výhodou této metody je, že můžeme určit vlastnosti systému, i když systém nejde popsat analyticky.

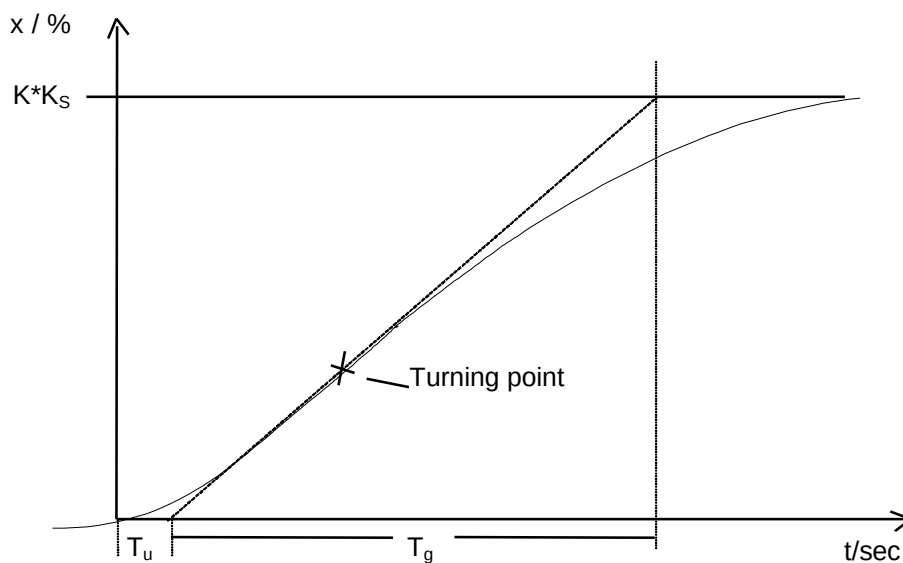


Figure: T_u - T_g -Approximation

3.8.1 Nastavení PI-regulátoru podle Zieglera-Nicholse

Vyšetřením P-T₁-T_L systému, Ziegler a Nichols zjistíme následující optimální nastavení regulátoru pro pevný pracovní bod.

$$K_{PR} = 0,9 \frac{T_g}{K_S T_u}$$

$$T_N = 3,33 T_u$$

Tyto nastavení obecně poskytují vcelku dobrou odezvu na poruchy [7]

3.8.2 Nastavení PI regulátoru podle Chien, Hrones a Reswick

Pro tuto metodu, je vyšetřena odezva na poruchu i na změnu vstupu, pro získání nejlepších hodnot ladění. Výsledkem jsou různé hodnoty pro oba případy. Navíc získáme dvě různá nastavení pro různé požadavky zohledňující kvalitu řízení.

Následující nastavení:

- Chování při poruše:

Aperiodická přechodová reakce s kratším trváním.

20% překmit, minimální čas oscilace

$$K_{PR} = 0.6 \frac{T_g}{K_S T_u}$$

$$K_{PR} = 0.7 \frac{T_g}{K_S T_u}$$

$$T_N = 4 T_u$$

$$T_N = 2,3 T_u$$

- Změna reference:

Aperiodická přechodová
reakce s kratším trváním.

$$K_{PR} = 0.35 \frac{T_g}{K_S T_u}$$

$$T_N = 1,2 T_g$$

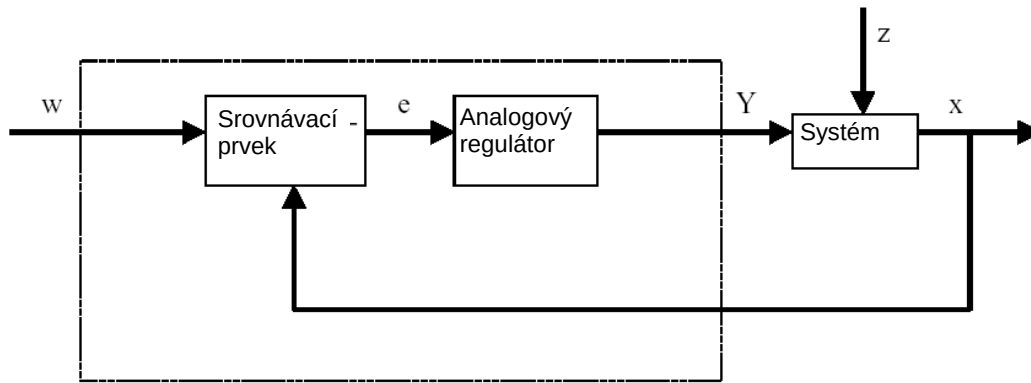
20% překmit, minimální čas
oscilace

$$K_{PR} = 0.6 \frac{T_g}{K_S T_u}$$

$$T_N = T_g$$

3.9 Digitální řízení

Zatím jsme mluvili jen o analogových regulátorech takové regulátory, které z odchylky systému zjištěné jako analogová hodnota získají řídicí proměnnou také jako analogovou hodnotu. Schéma takového cyklu vypadá následně:

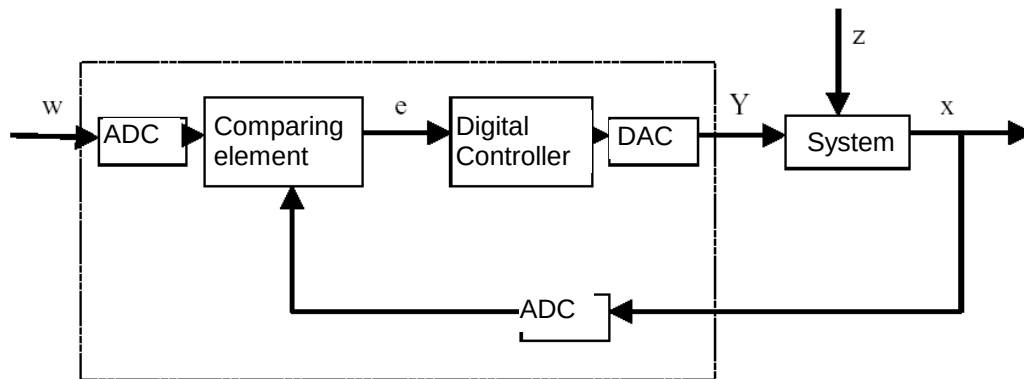


Nicméně, často je výhodné vyhodnocovat odchylku systému digitálně. Zaprvé, vztah mezi odchylkou a výstupem regulátoru může být specifikován snadněji, když je definovaný algoritmem nebo vzorcem implementovatelným do programu spíše než když je potřeba na něj vytvořit analogový obvod. Zadruhé, vyšší integrální obvody jsou snadněji stvořitelné v digitální technologii, takže můžeme umístit několik regulátorů do malého prostoru. A konečně, je možné distribuovat výpočetní čas, takže můžeme využít jeden regulátor pro víc řízených okruhů.

Aby bylo digitální zpracování možné, nejdříve musí být referenční proměnná a zpětnovazební proměnná převedeny pomocí AC převodníku na digitální signál.

Digitální srovnávací prvek je vzájemně srovná a rozdíl předá do digitálního regulátoru. Jeho výstupní proměnná je poté převedena na analogovou hodnotu pomocí DA převodníku. Celá tato jednotka skládající se z převodníků, srovnávacího elementu a řídicího elementu se na venek tváří jako analogový konvertor.

Diagram ukazuje strukturu digitálního regulátoru:



Ke všem výhodám jaké digitální regulátor má jsou tu ovšem nevýhody. Z tohoto důvodu, musíme vybírat některé proměnné dostatečně velké, aby kvalita a přesnost řízení neutrpěla kvůli digitalizaci signálu. Kritéria kvality jsou:

- Kvantifikační rozlišení DA převodníků

Indikuje jak kvalitně je spojitá funkce převedena na digitální rastr. Rozlišení musí být tak velké, aby se neztratili žádné vlastnosti podstatné pro řízení.

- Vzorkovací frekvence AD převodníku

Je frekvencí, s jakou je hodnota analogové funkce měřena na převodníku a digitalizována. Musí být tak velká aby mohl regulátor reagovat i na rychlé změny v řízené proměnné.

- Čas cyklu

Na rozdíl od analogových počítačů, každý digitální proces sebere trochu času procesoru. Rychlost počítače musí být taková, aby se během jednoho cyklu výrazně nezměnila řízená proměnná. Kvalita digitálního regulátoru musí být taková, aby z vnějšího pohledu reagoval rychle a přesně jako analogový regulátor.

4. Vzorová úloha – řízení hladiny v nádrži

Naprogramujeme řízení hladiny pro náš program.

Senzor měří hladinu v nádrži a převádí hodnotu na napětí v rozmezí 0 až 10V: 0V odpovídá 0 litrům
10V odpovídá 1000 litrů.

Senzor je připojen na první analogový vstup SIMATIC S7-1200.

Nyní, hodnota hladiny bude 0 litrů (S1 == 0) nebo 700 litrů (S1 == 1).

Nyní **"PID_Compact"** regulátor integrovaný v STEP 7 Basic V10.5 je využit. Tento PID regulátor řídí pumpu jako řídící proměnnou pro hodnoty mezi 0 a 10V

Seznam úloh:

Adresa	Symbol	Data Type	Komentář
%IW 64	X_Level_Tank1	Int	analogový vstup vlastní hladina Tank1
%QW 80	Y_Level_Tank1	Int	analogový výstup řídící proměnná pump1
%I 0.0	S1	Bool	náhlé nastavení plnění na 0 (0) nebo 700 liters (1)

5. Programování řízení hladiny pro SIMATIC S7-1200

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

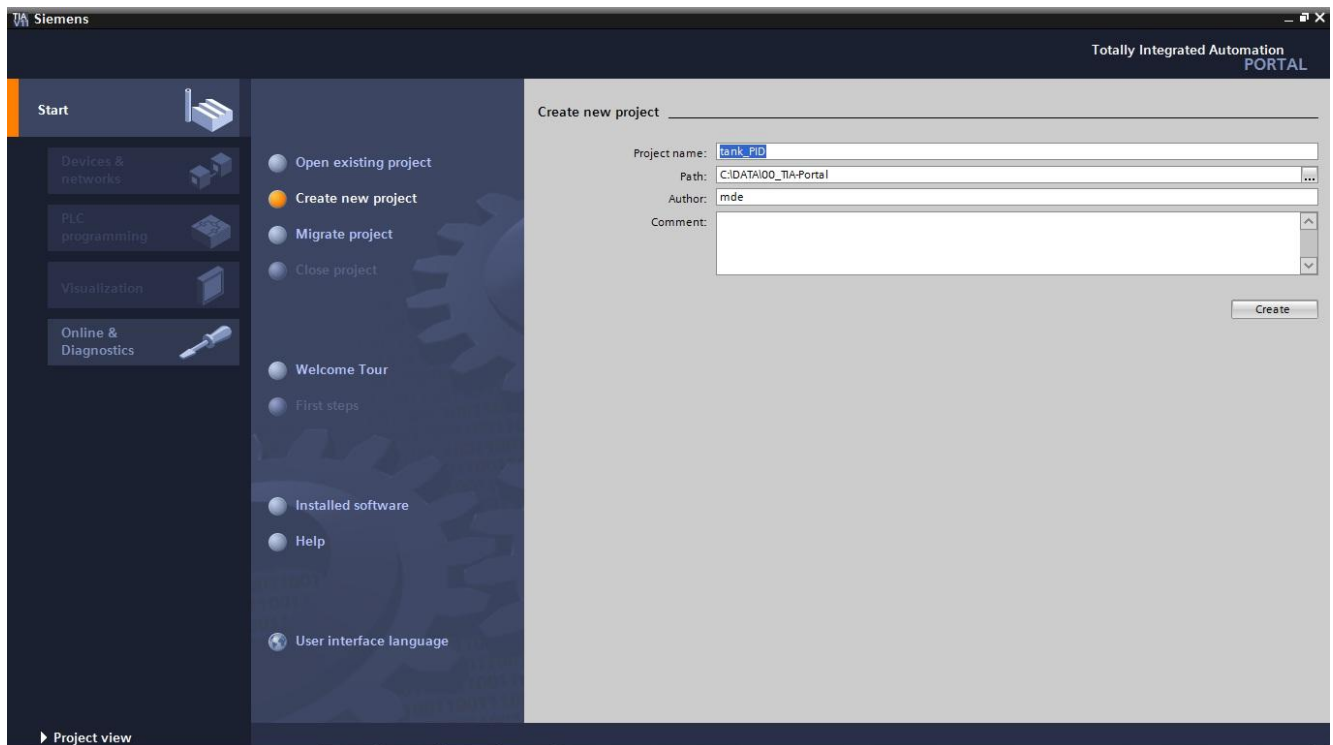
Zde pod jednotným rozhraním, jsou komponenty jako řídicí systém, vizualizace a síť nastaveny, parametrizovány a programovány. Pro diagnostiku chyb jsou k dispozici online nástroje.

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

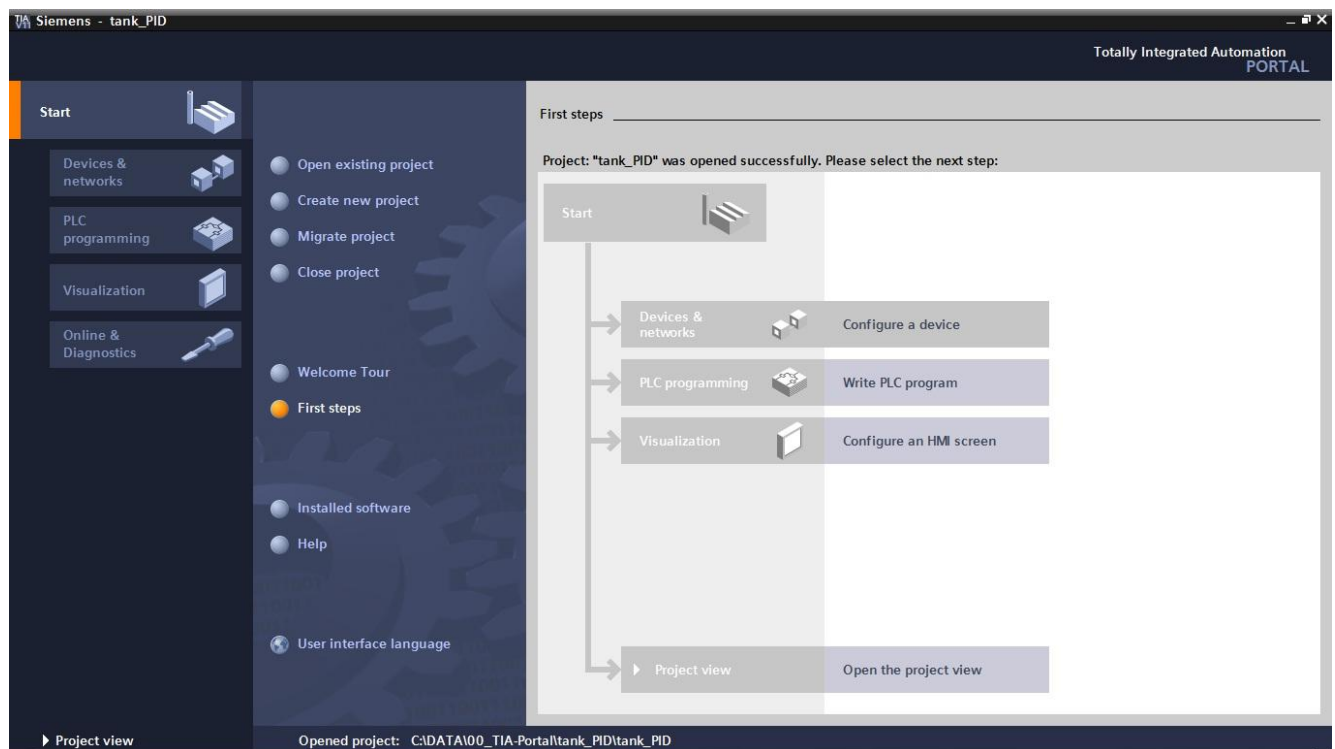
1. Centrálním nástrojem je '**Totally Integrated Automation Portal**'. Spustíme ho dvojklikem. (→ Totally Integrated Automation Portal V11)



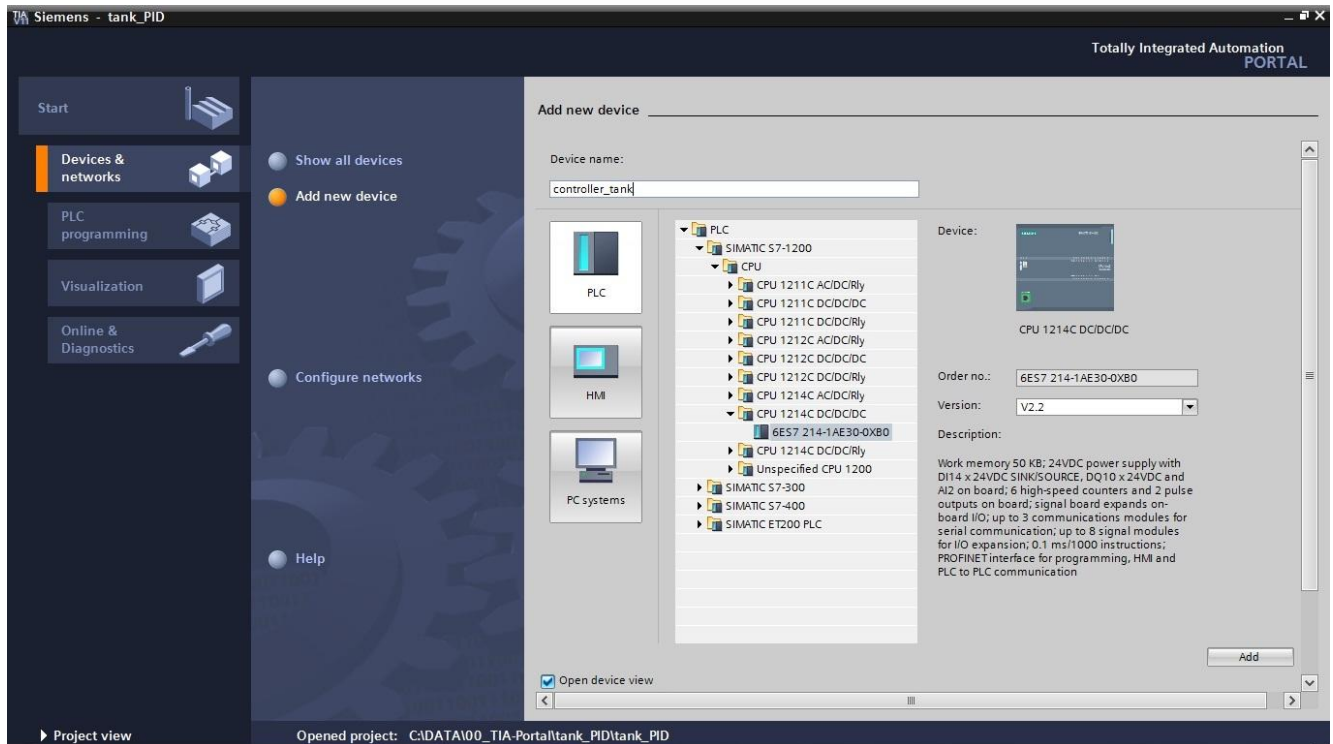
2. Programy jsou spravovány v projektech. Nyní vytvoříme takový projekt. (→ Create new project → Tank_PID → Create)



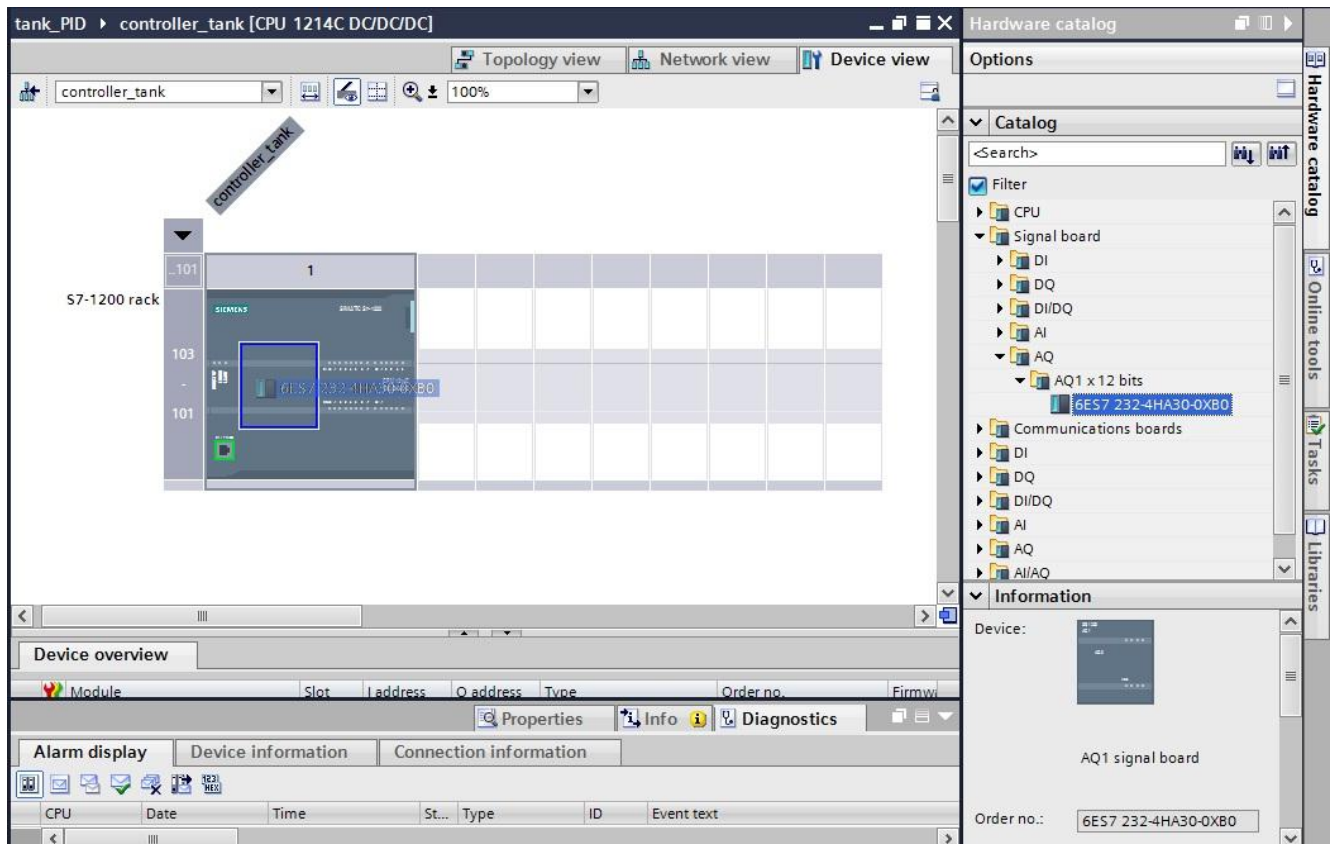
3. Nyní, **'First Steps'** první kroky pro konfiguraci. Nejdříve chceme **'Configure a device'**. (→ First steps → Configure a device)



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



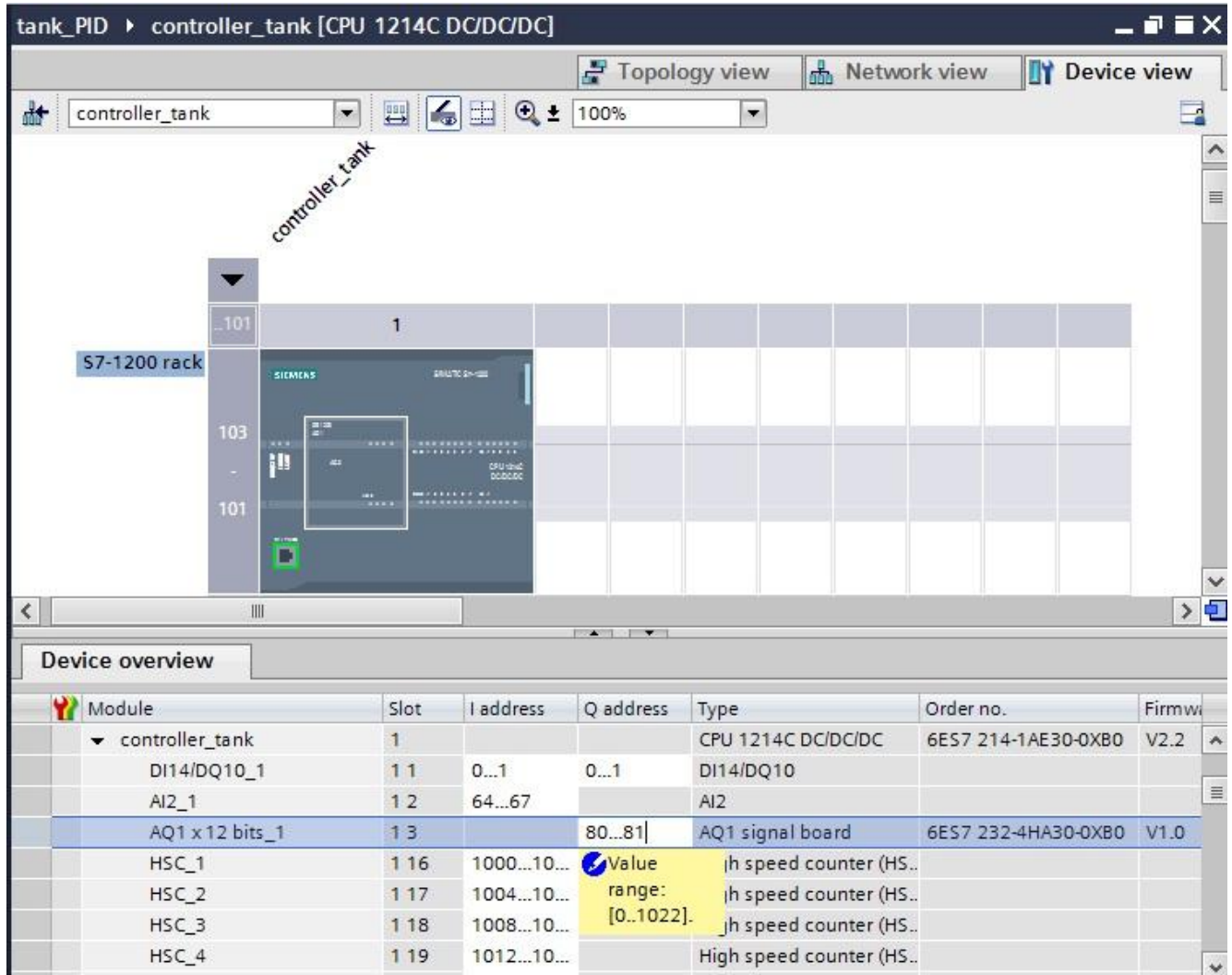
5. Software se nyní sám přepne do Project view a otevře hardware konfiguraci. Teď můžeme přidat z hardware katalogu další moduly a karty. (→ Catalog → Signal board → AO1 x 12Bit → 6ES7 232-...)



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

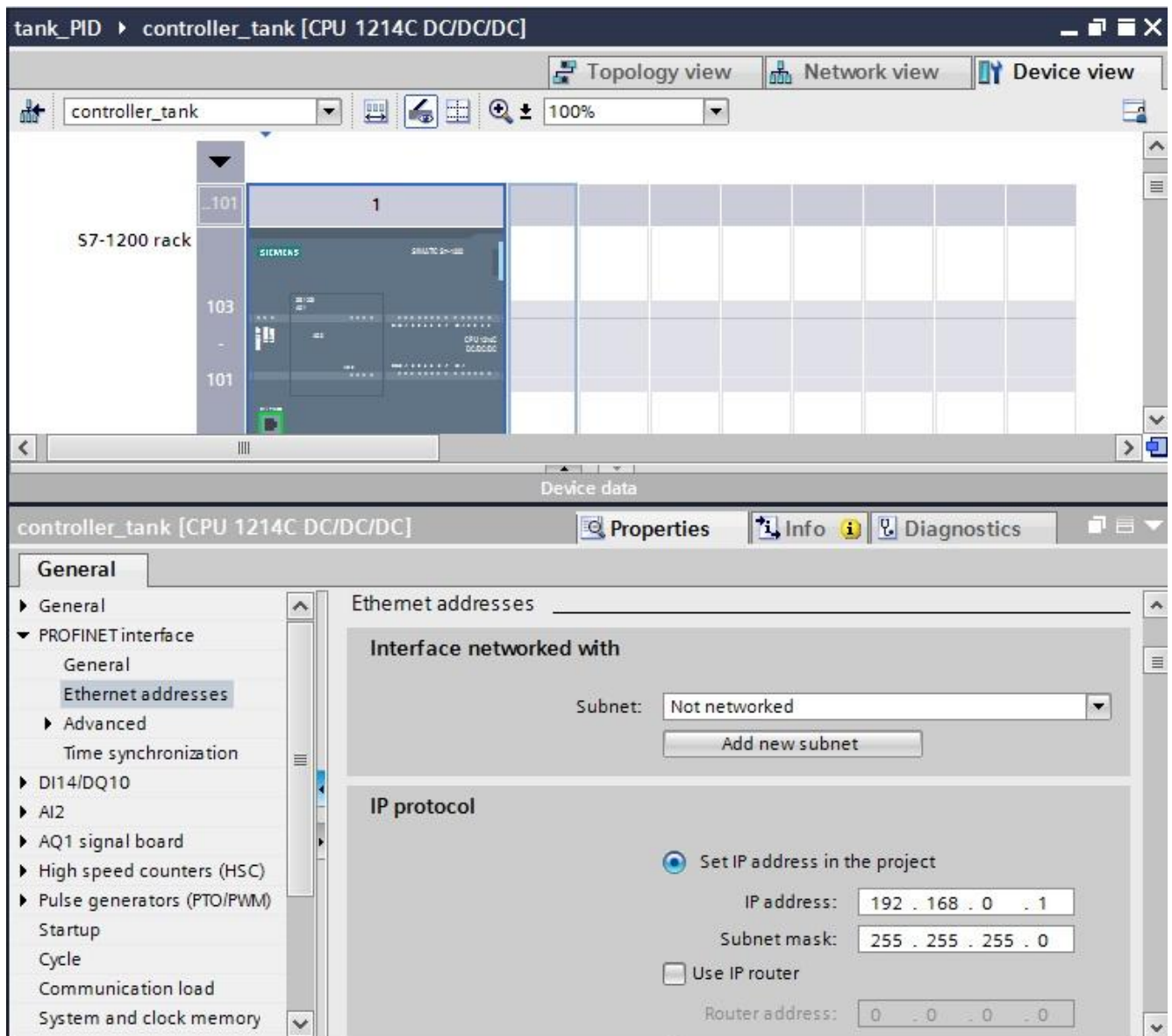
(→ Device overview → AQ1 x 12Bit → 80...81)



Device overview

Module	Slot	I address	Q address	Type	Order no.	Firmware
controller_tank	1			CPU 1214C DC/DC/DC	6ES7 214-1AE30-0XB0	V2.2
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..		
HSC_3	1 18	1008...10...		High speed counter (HS..		
HSC_4	1 19	1012...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)

The screenshot displays the Siemens TIA Portal interface for a project named 'tank_PID'. The 'Project tree' on the left shows the hierarchy: 'tank_PID' > 'controller_tank [CPU 1214C DC/DC/DC]' > 'PLC tags' > 'Default tag table [17]'. The 'Default tag table' is shown as a table with the following data:

	Name	Data type	Address	Retain	Visible..	Acces...	Comment
1	X_Level_Tank1	Int	%IW64	☐	☒	☒	analog input actual level filling level tank1
2	Y_Level_Tank1	Int	%QW80	☐	☒	☒	analog output manipulated value pump1
3	S1	Bool	%I0.0	☐	☒	☒	sudden setpoint change filling level 0 (0) or 700 liter (1)
4	<Add new>			☐	☒	☒	

The 'Properties' window for tag 'S1' is open, showing the following details:

- General**
 - Name: S1
 - Data type: Bool
 - Address: %I0.0
 - ☐ Retained
 - Comment: sudden setpoint change filling level 0 (0) or 700 liter
- Time stamp**
 - Date created: 7/6/2012 6:18 AM
 - Last modified: 7/6/2012 6:20 AM
- Usage**

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)

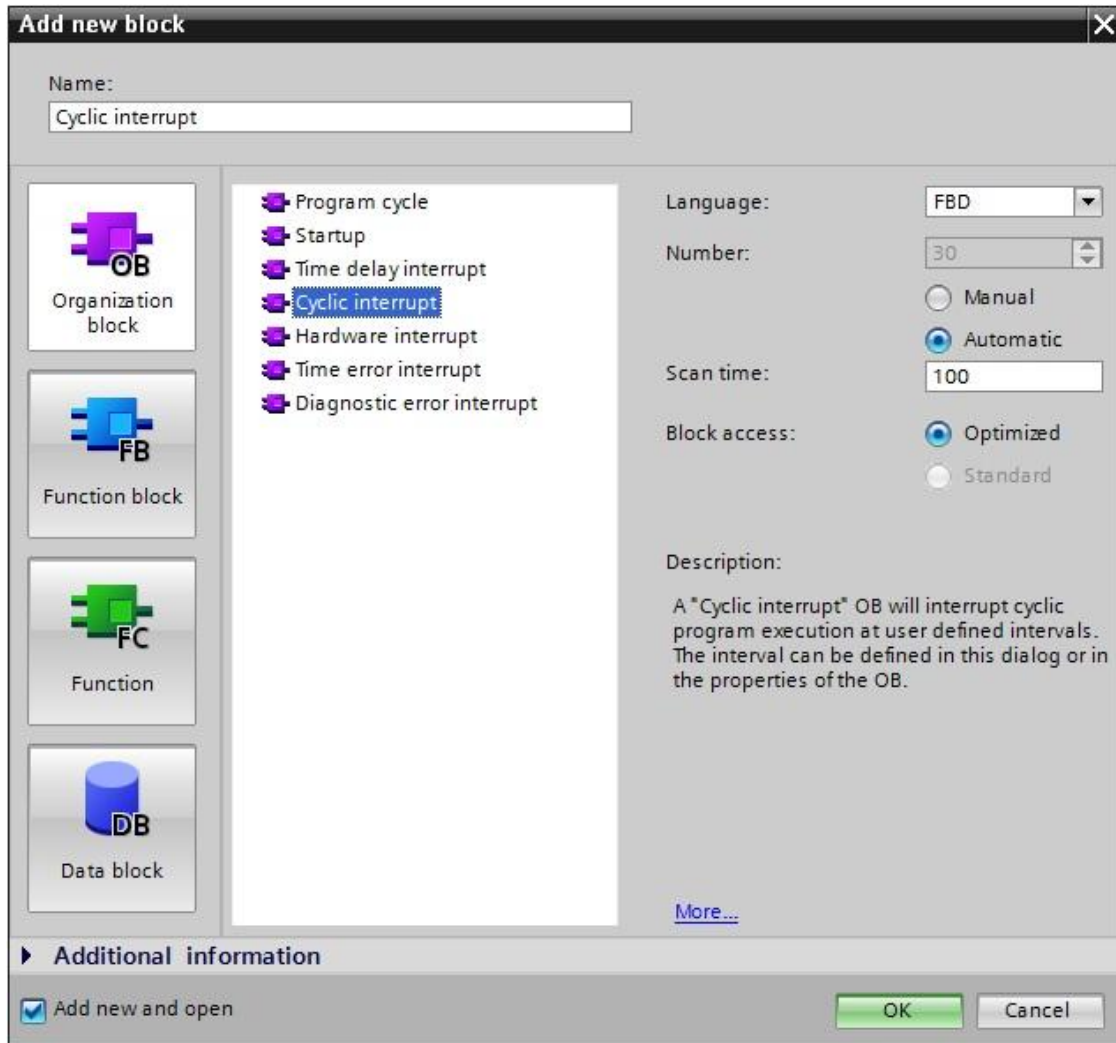
The screenshot shows the Siemens TIA Portal interface. On the left, the 'Project tree' displays the hierarchy: tank_PID > controller_tank [CPU 1214C DC/DC/DC] > PLC tags > Default tag table [17]. The 'Default tag table' is shown as a table with the following data:

	Name	Data type	Address	Retain	Visible...	Acces...	Comment
1	X_Level_Tank1	Int	%IW64	☐	☒	☒	analog input actual level filling level tank1
2	Y_Level_Tank1	Int	%QW80	☐	☒	☒	analog output manipulated value pump1
3	S1	Bool	%I0.0	☐	☒	☒	sudden setpoint change filling level 0 (0) or 700 liter (1)
4	<Add new>			☐	☒	☒	

The 'Properties' window for tag S1 is open, showing the 'General' tab. The fields are as follows:

- Name: S1
- Data type: Bool
- Address: %I0.0
- ☐ Retained
- Comment: sudden setpoint change filling level 0 (0) or 700 liter
- Time stamp:
 - Date created: 7/6/2012 6:18 AM
 - Last modified: 7/6/2012 6:20 AM

10. Vybereme '**Organization block(OB)**', a jako typ '**Cyclic interrupt**'. Jako programovací jazyk '**FBD**'. Číslo je přiřazeno automaticky (OB200). Necháme skenovací čas na hodnotě 100ms. Potvrdíme '**OK**'. (→ Organization block (OB) → Cyclic interrupt → FBD → Scan time 100 → OK)

**Poznámka:**

PID regulátor musí být vyvoláván s pevným časem (here 100ms) vzhledem k tomu, že jeho zpracování je kritické. Regulátor nemůže být optimalizován, pokud není patřičně volán.

11. Organizační blok '**Cyclic interrupt**'[OB200] se otevře sám, než napíšeme program musíme specifikovat tagy.

V tomto bloku je využit jen jeden typ tagu:

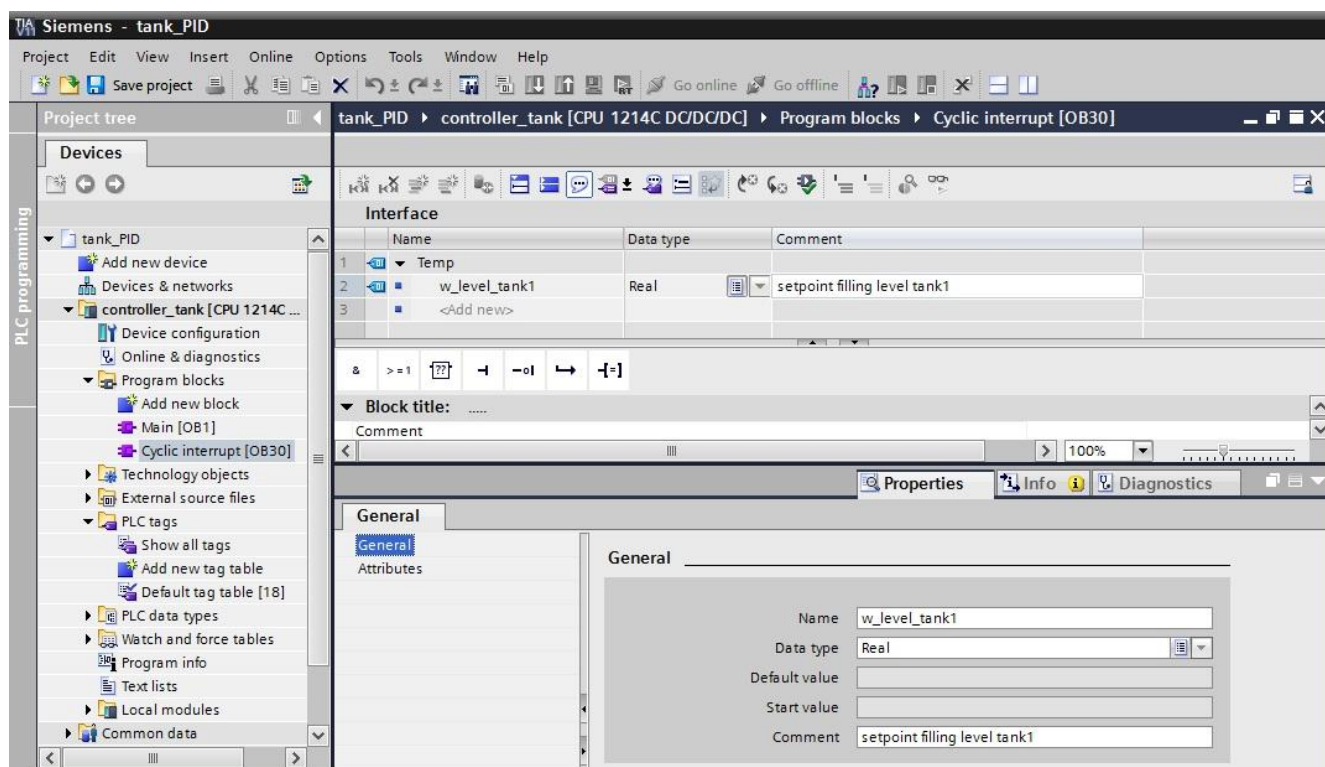
Typ	Jméno	Funkce	Dostupné v
Temporary Local data	Temp	Tag využitý k dočasnému uložení výsledků. Data jsou držena jen jeden cyklus.	Functions, function blocks and organization blocks

12. V našem případě potřebujeme následující lokální tagy.

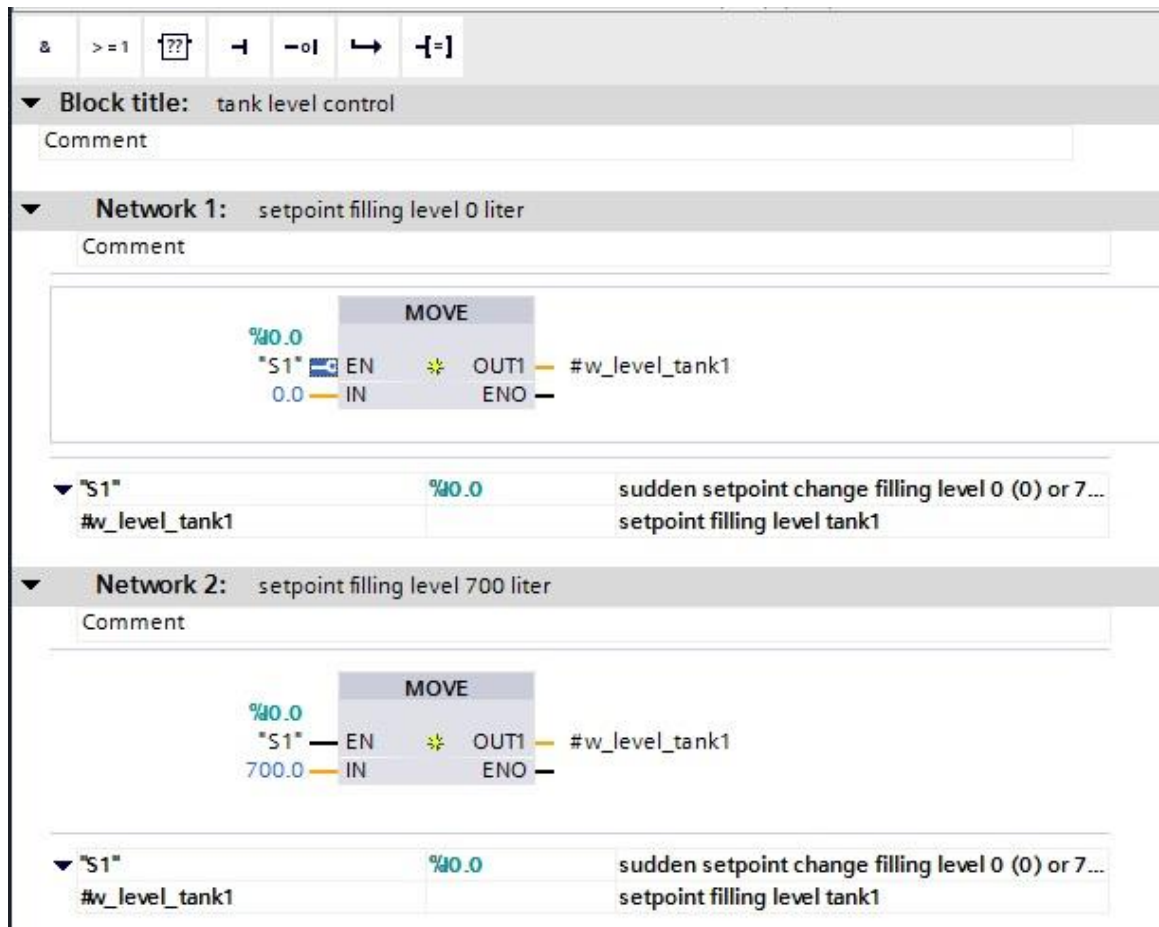
Temp:

w_level_tank1 Real Tag ukládá setpoint pro Tank1 jako dočasnou hodnotu.

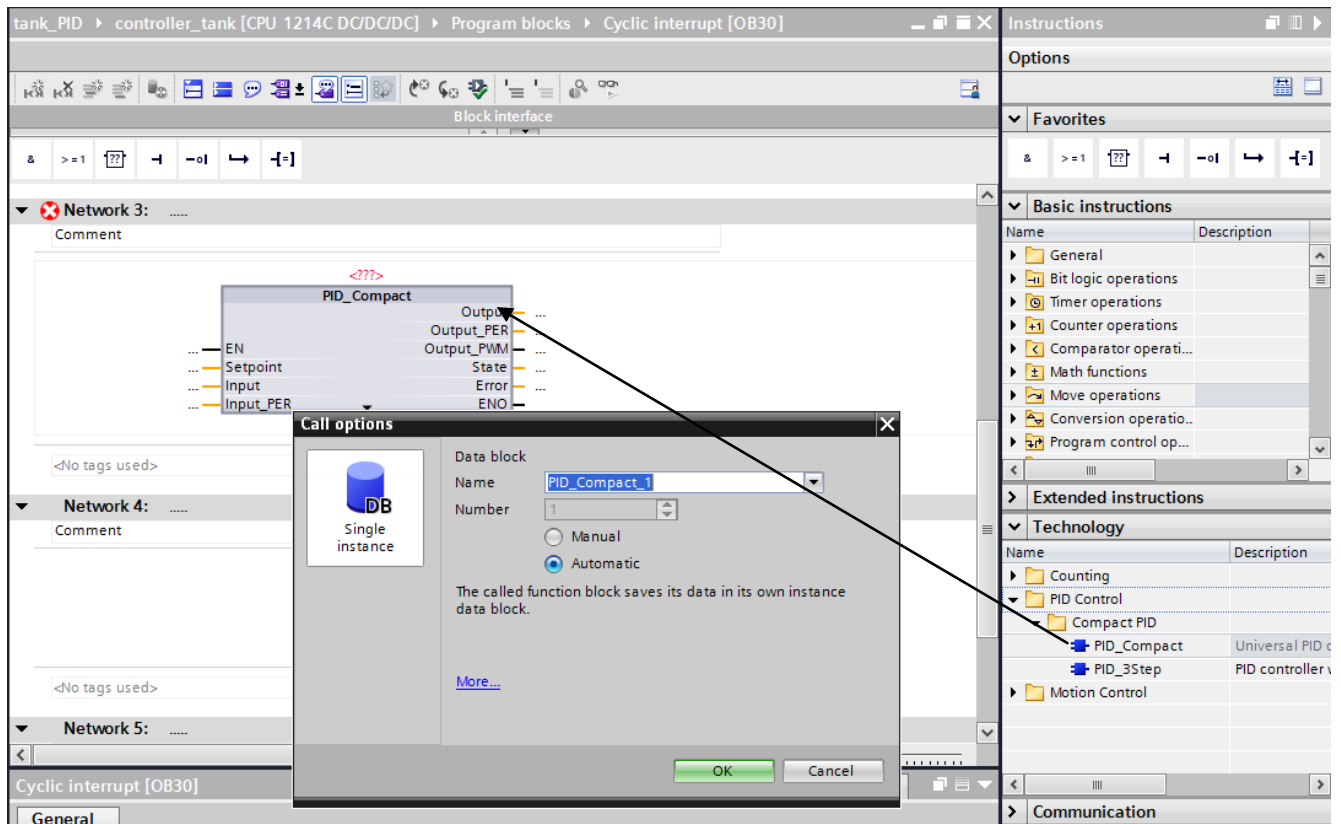
- Opět v tomto případě důležité, využití správného datového typu Real; jinak není kompatibilní s následným programem PID regulátoru.
Pro přehlednost je důležité komentovat všechny použité tagy.



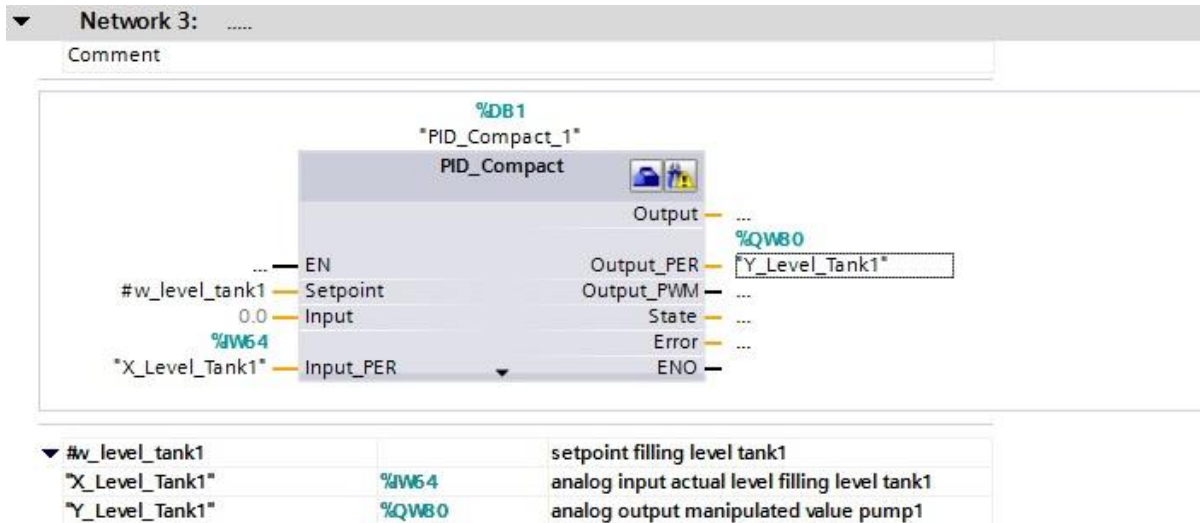
13. Po deklaraci tagů, můžeme tagy využívat pod jejich symbolickými jmény. (Tagy jsou značeny symbolem '#'.) Zde v prvních dvou sítích - s 'MOVE' instrukcí- je číslo s plovoucí desetinou čárkou 0.0 (S1 == 0) nebo 700.0 (S1 == 1) zkopírováno do lokálního tagu #w_fuell_tank1. (→ Instructions → Move → MOVE)



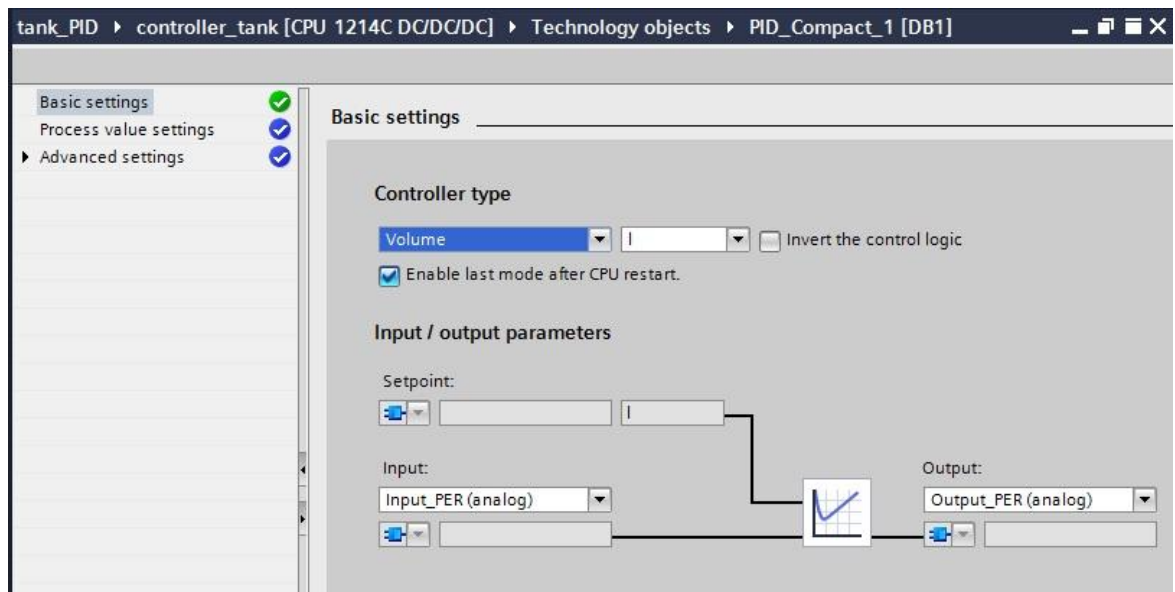
14. řídící blok '**PID_Compact**' je přesunut do třetí sítě. Protože tento blok nepodporuje více instancí, musíme mu přiřadit single-istanční data blok. Ve STEP7 je vytvořen automaticky.
(→ Advanced instructions → PID → PID_Compact → OK)



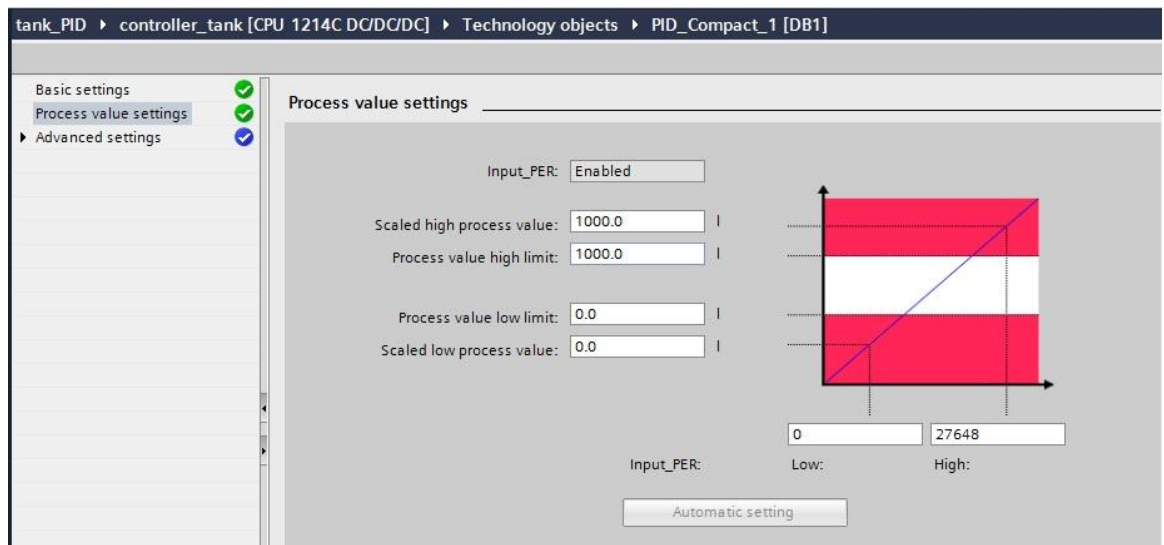
15. Připojíme blok, jak je ukázáno (local tag #w_fuell_tank1), vlastní hodnota (global tag "X_Level_Tank1") a řídicí proměnná (global tag "Y_Level_Tank1"). Pak můžeme otevřít konfigurační okno kontrolního bloku . (→ #w_level_tank1 → "X_Level_Tank1" → "Y_Level_Tank1" → )



16. Tady musíme udělat '**Basic settings**' jako typ regulátoru a jeho vnitřní strukturu. (→ Basic settings → Controller type Volume → I → actual value: Input_PER(analog) → manipulated value: Output_PER)

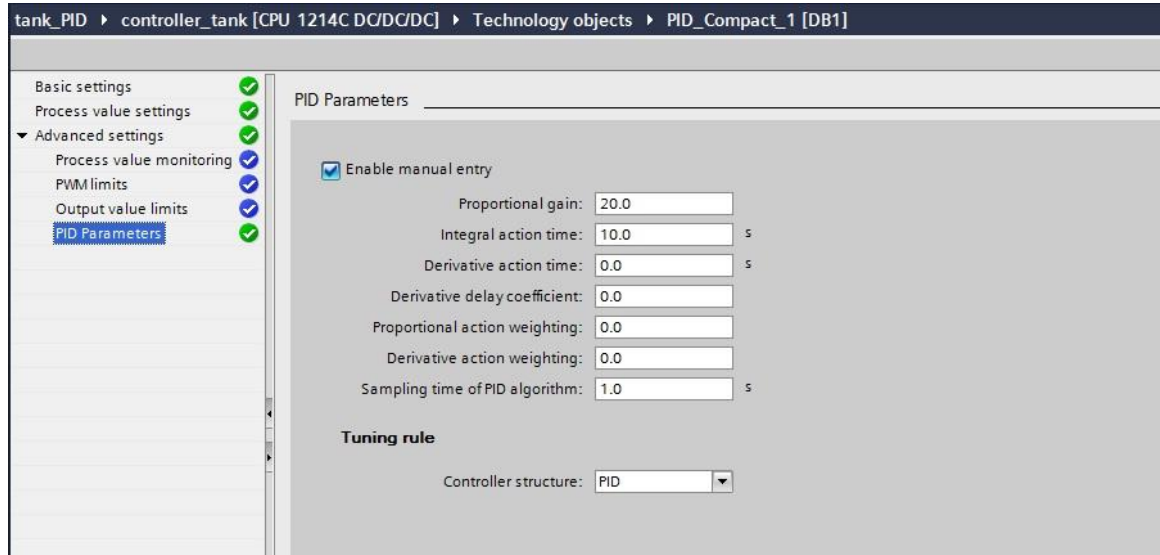


17. V 'Process value settings' nastavíme měřený rozsah od 0 litrů do 1000 litrů. Také nastavíme limity. (→ Process value settings → Scaled high 1000.0 l → high limit 1000.0 l → Low limit 0.0 l → scale low 0.0 l)

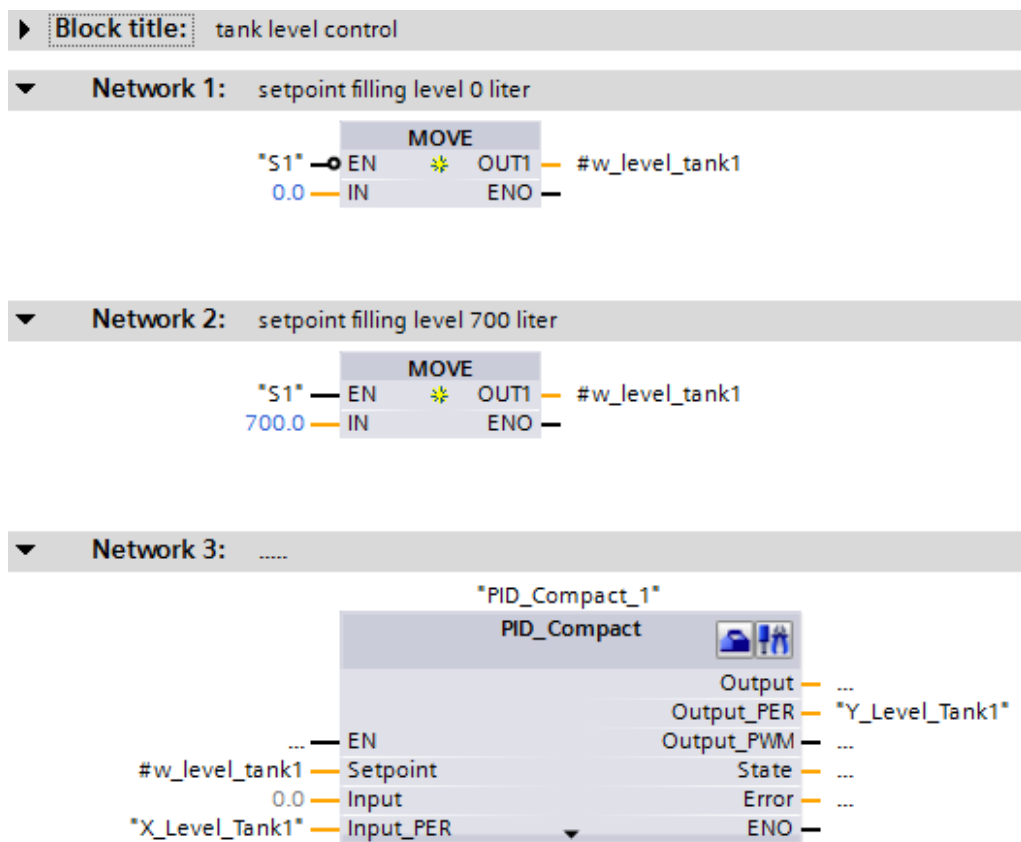


18. V 'Advanced settings', je také 'Process value monitoring' a manuální nastavení 'PID

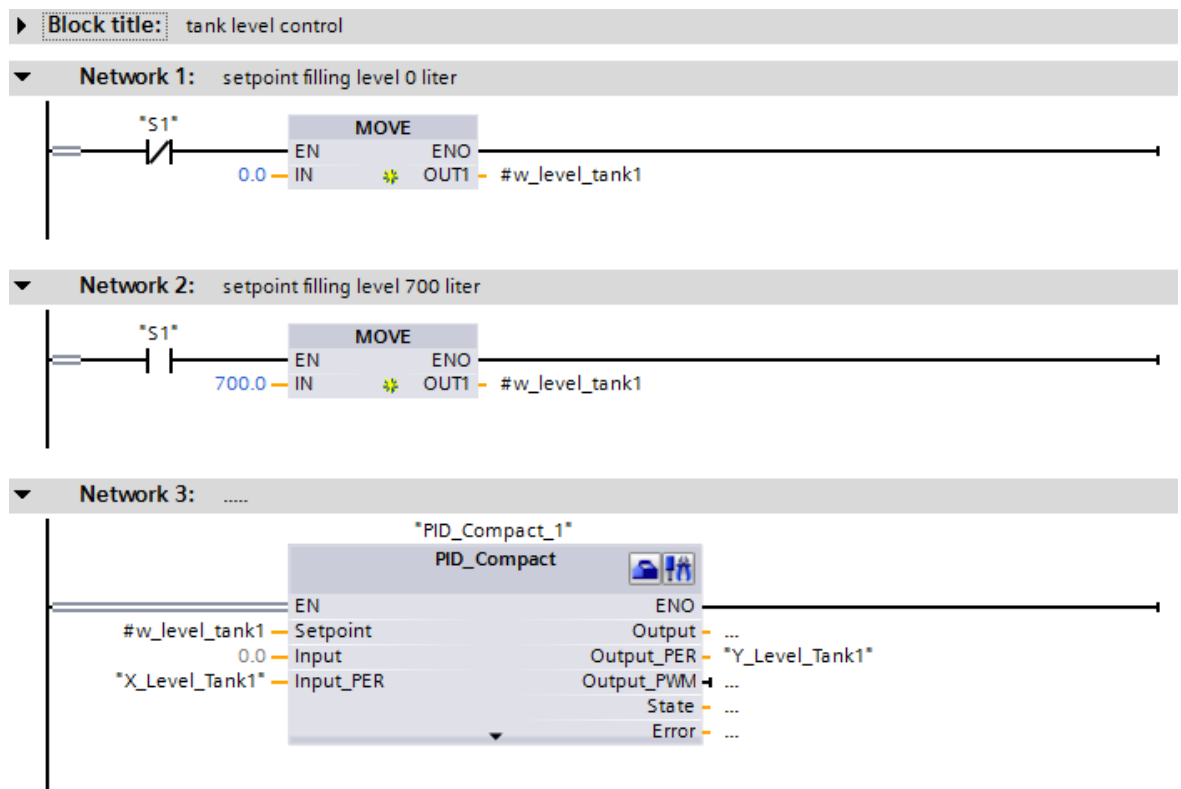
Parameters'. Kliknutím na  Save project projekt uložíme. (→ Advanced settings → Process value monitoring → PID Parameters →  Save project)



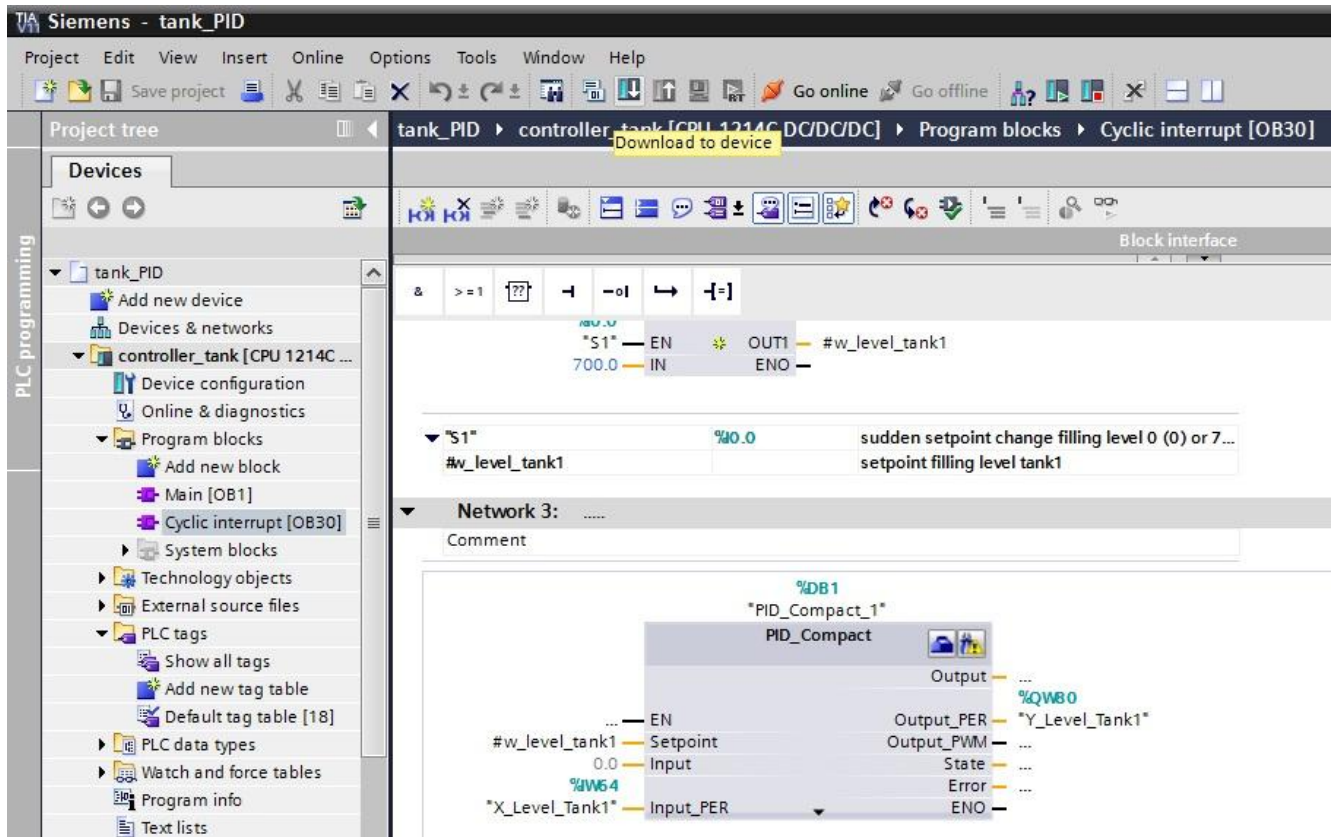
Program in function block diagram (FBD):



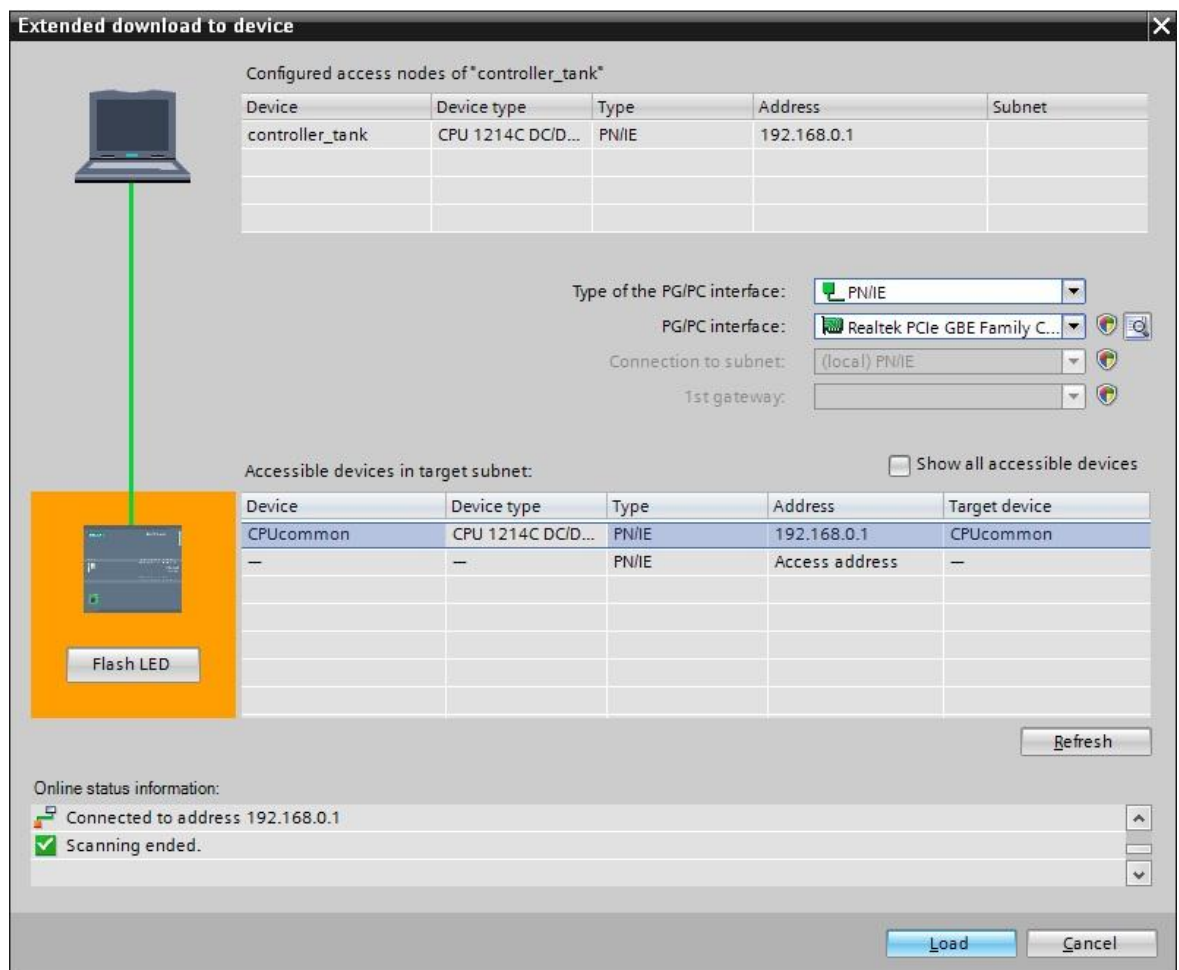
Program v žebříkovém diagramu (LAD):



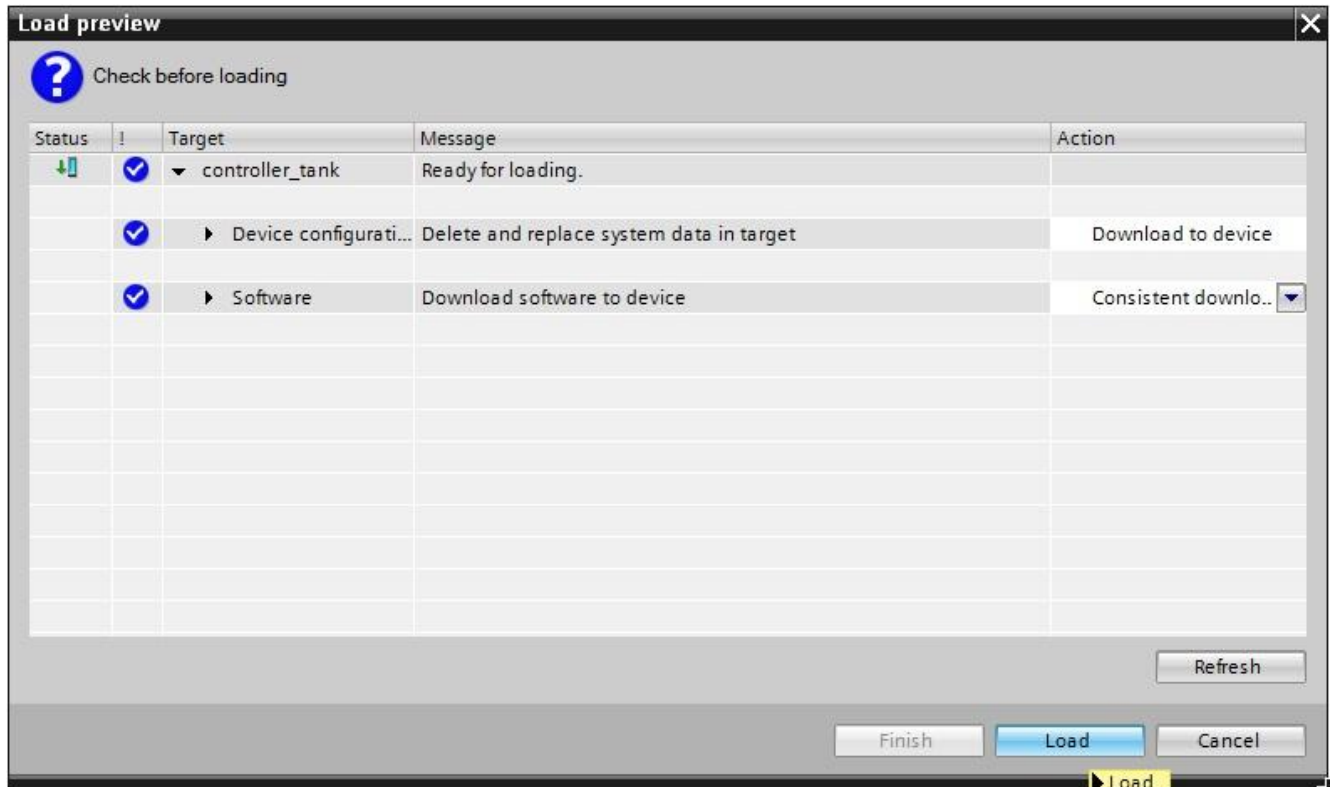
19. Pro nahrání celého projektu do CPU, vybereme složku '**controller_tank**' a pak kliknutím na symbol  nahrajeme do zařízení.
(→controller_tank → )



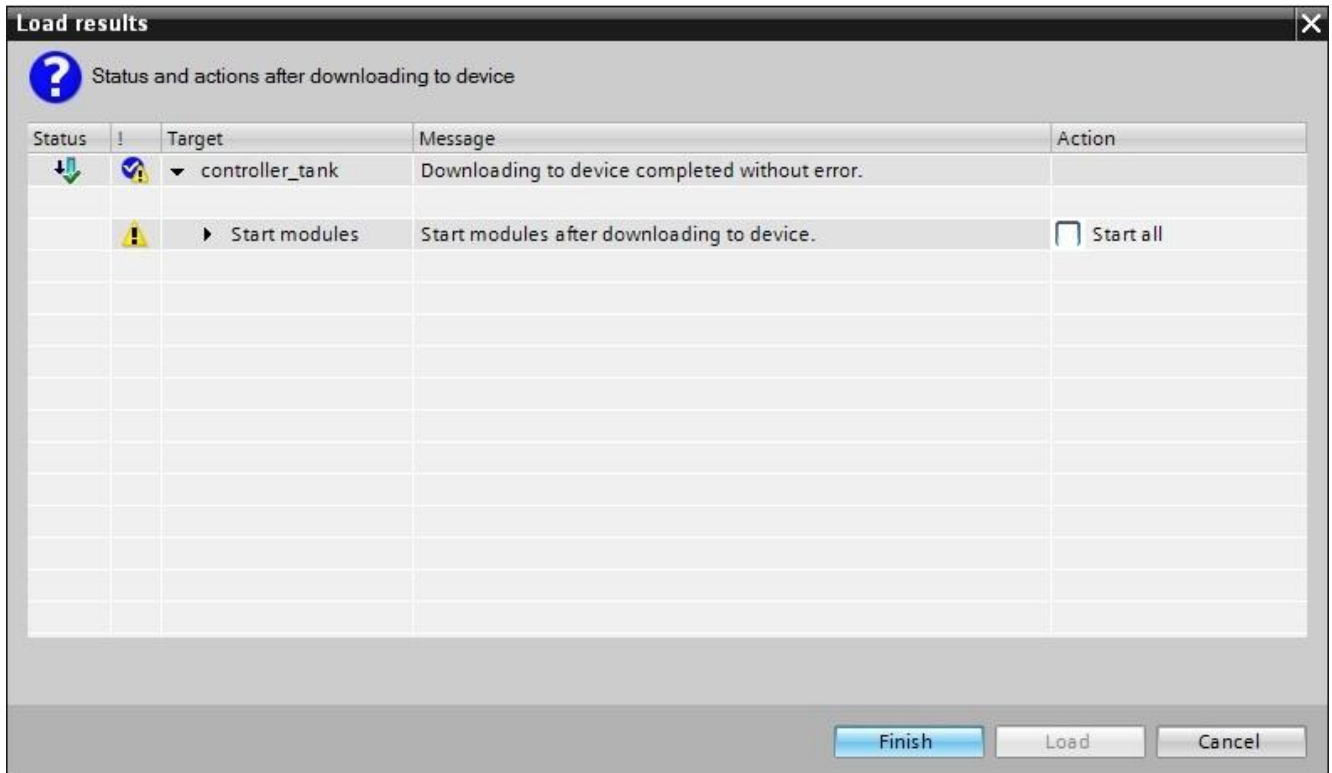
20. Pokud není rozhraní PG/PC specifikováno, zobrazí se okno ve kterém je možné tak udělat. (→ PG/PC interface for loading → load)



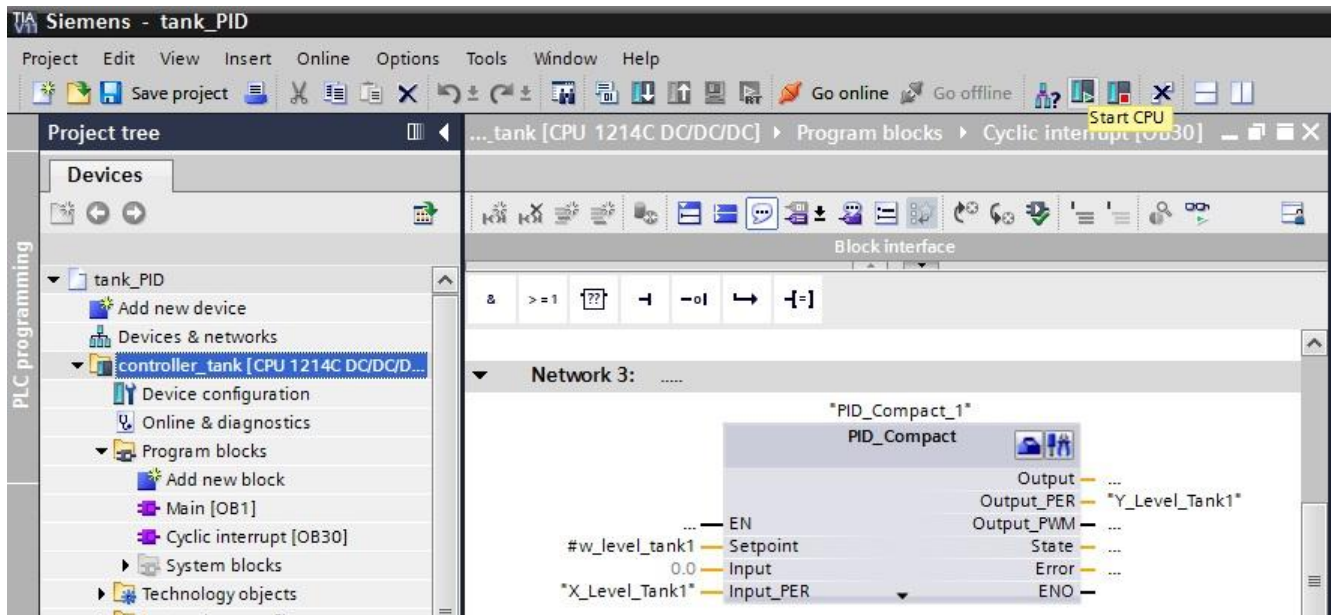
21. Klikneme na 'Load' ještě jednou. Během nahrávání je zobrazen status v okně. (→ Load)



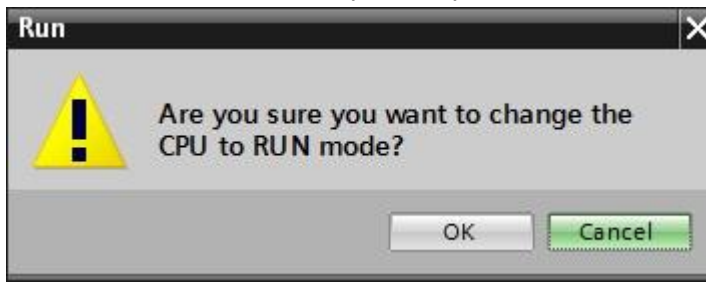
22. Úspěšné nahrání je zobrazeno v okně, nyní už jen klik na 'Finish'. (→ Finish)



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

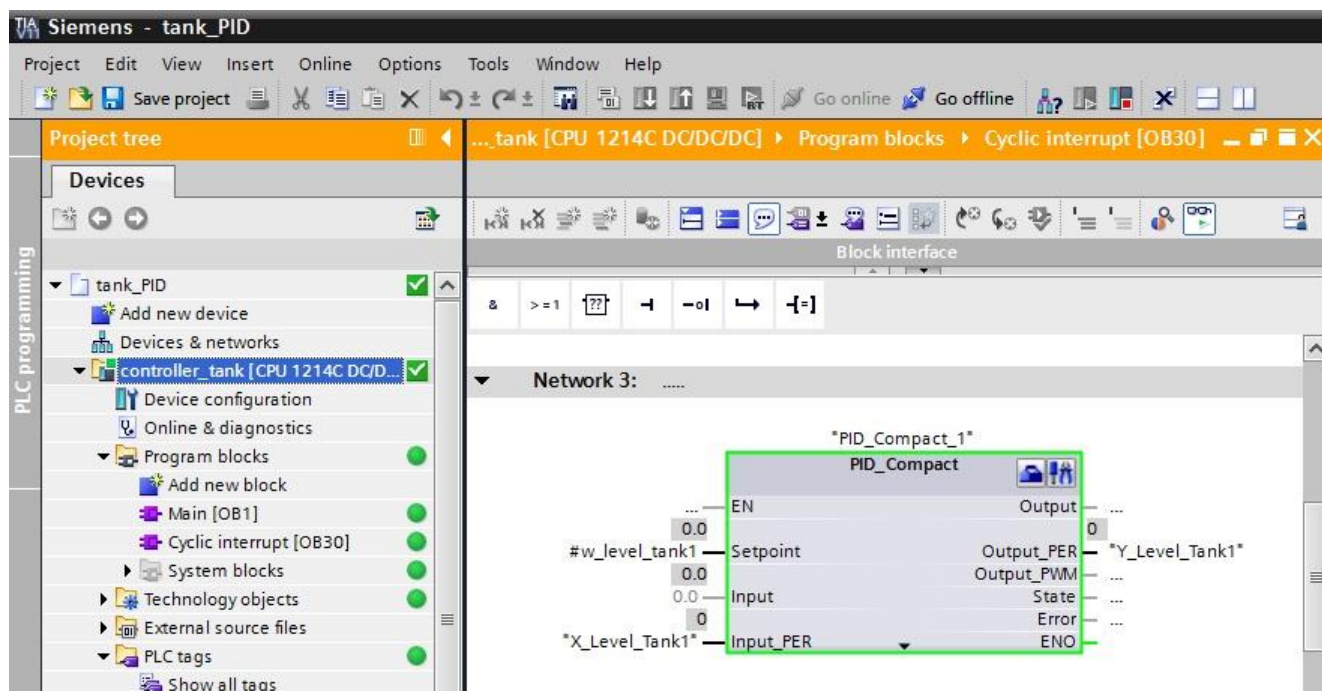


24. Potvrdíme, že chceme opravdu spustit CPU kliknutím na 'OK'. (→ OK)

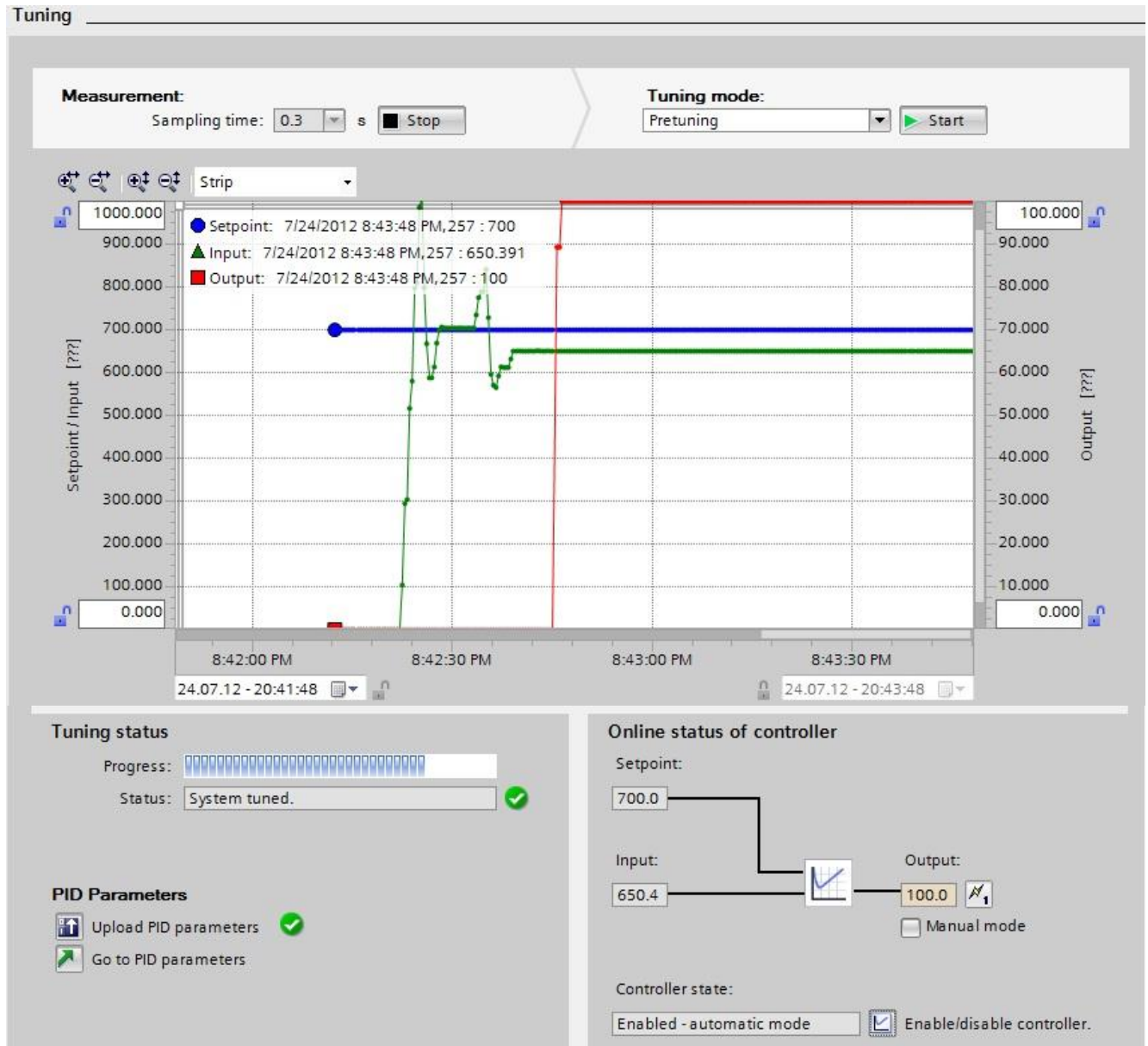


25. Kliknutím na symbol  Monitoring on/off, můžeme sledovat status proměnných během testování programu.

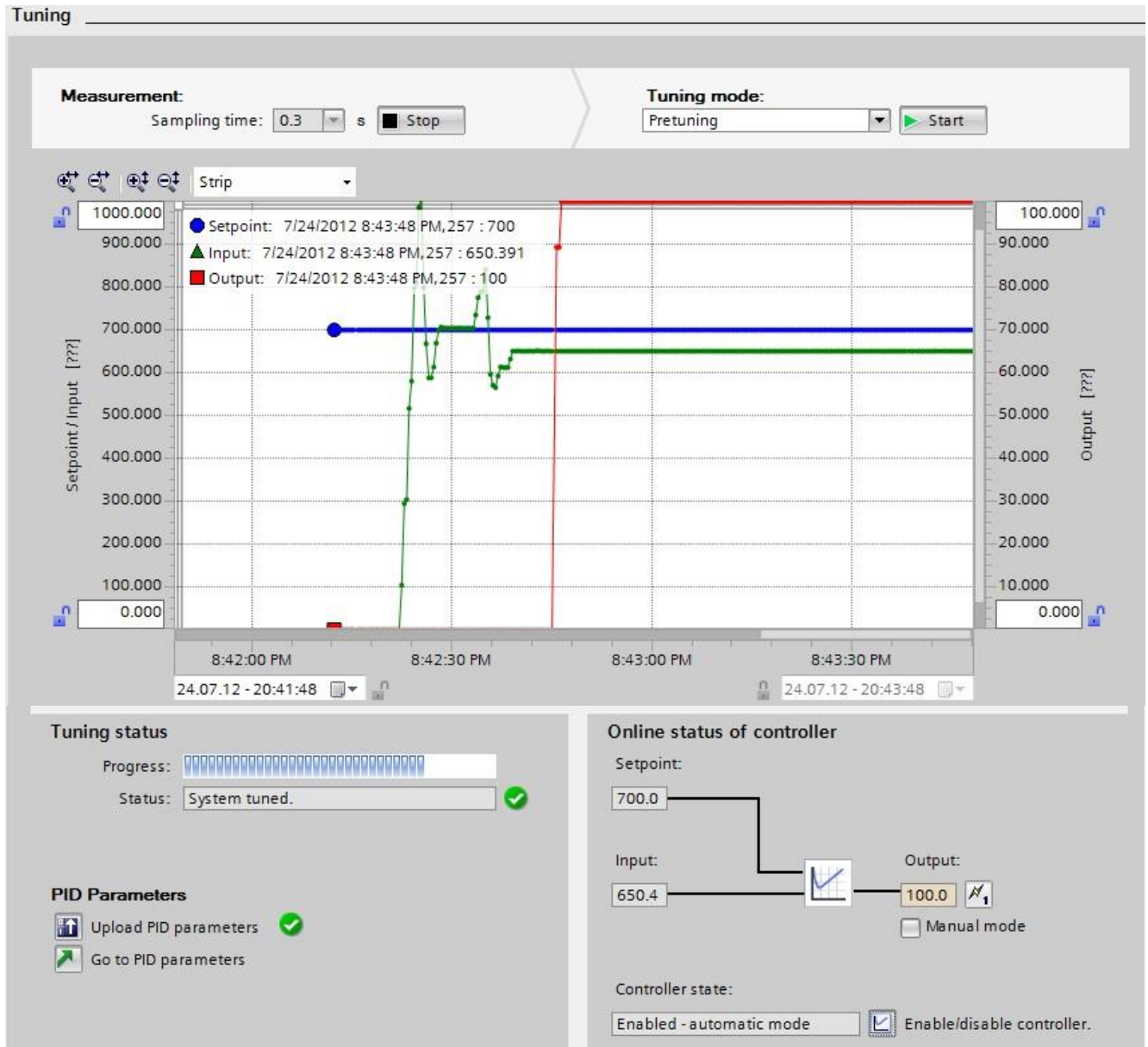
Nicméně když spustíme, CPU regulátor '**PID_Compact**' není aktivní. Nyní spustíme správu kliknutím na . (→ Cyclic interrupt[OB200] →  → PID_Compact →  Commissioning)



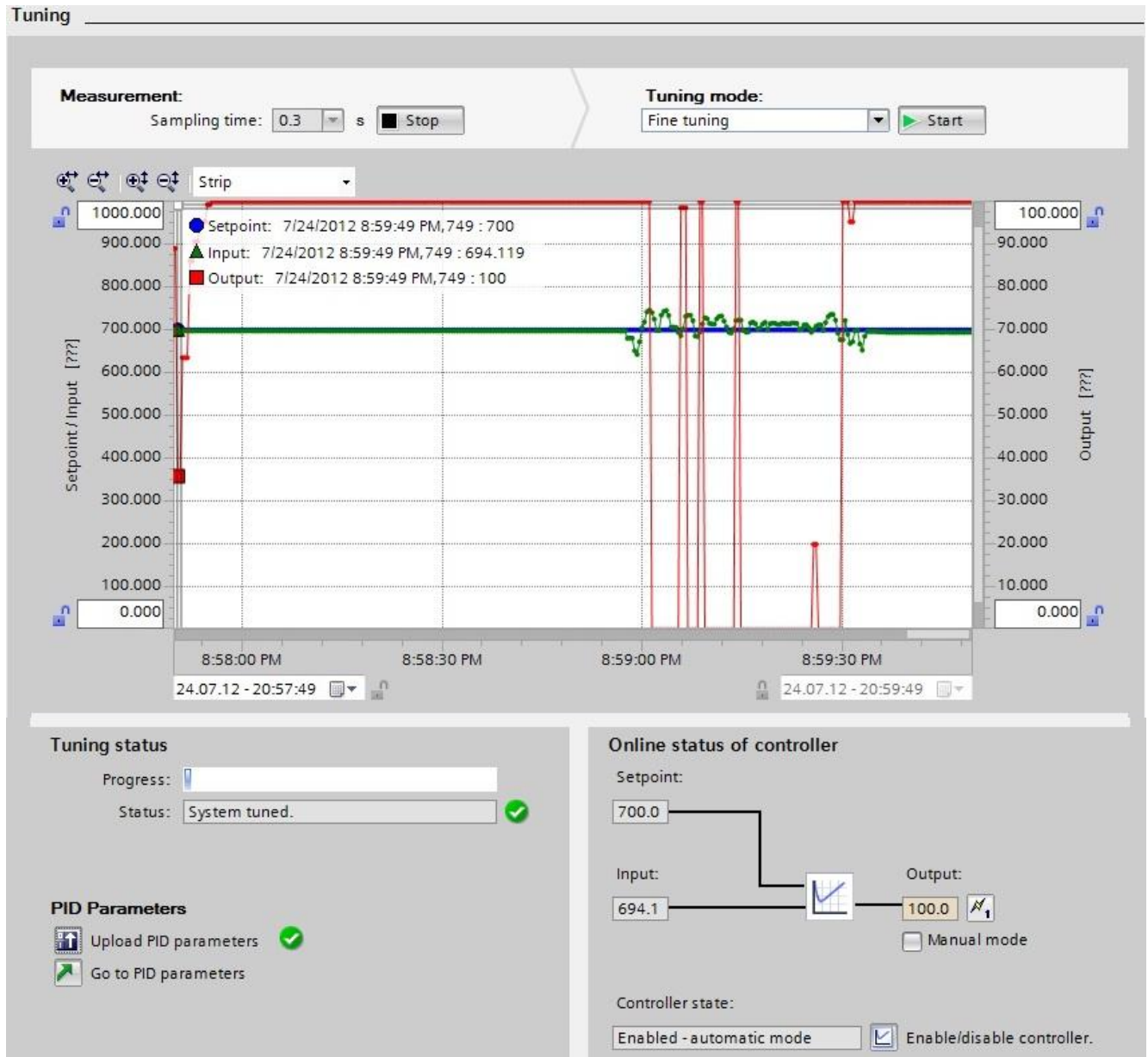
26. Na pracovní obrazovce můžeme zobrazit následující pomocí **'Measurement Start/Stop'** v diagramu: setpoint, vlastní hodnotu a řídicí proměnnou.
Po nahrání do CPU je regulátor stále neaktivní. To znamená, že řídicí proměnná zůstává na 0%.
Nyní spustíme **'Tuning mode' 'Pretuning'** a pak **'Pretuning' 'Start'**. (→ Measurement → Start → Tuning mode → Pretuning → Start)



27. Nyní je spuštěné automatické ladění. V poli '**Tuning status**' jsou zobrazeny současné kroky a chyby. Progress bar nám zobrazuje úroveň, do jaké jsme se už dostali.



28. Pokud autoladění proběhlo bez chyb, parametry PID byli optimalizováni. PID regulátor přepne do automatického módu a použije optimalizované parametry. Optimalizované parametry jsou drženy během ON modu a i po restartu CPU. Tlačítkem  nahrajeme PID parametry do projektu. (→



Poznámka:

Pro rychlejší zpracování, jako pro řízení rychlosti, **fine tuning** využijeme pro optimalizaci. Cyklus proběhne po několik minut tady, kde jsou všechny PID parametry nastaveny.

Hodnoty parametrů můžeme sledovat po nahrání projektu v data bloku.