



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

NÁVRH A REALIZACE VZOROVÝCH ÚLOH PRO PLC B&R AUTOMATION

DESIGNE AND IMPLEMENTATION OF TUTORIALS FOR PLC B&R AUTOMATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

DANIEL SÝKORA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. et Ing. STANISLAV LANG

BRNO 2014

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky
Akademický rok: 2013/2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Daniel Sýkora

který/která studuje v bakalářském studijním programu

obor: Strojní inženýrství (2301R016)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Návrh a realizace vzorových úloh pro PLC B&R Automation

v anglickém jazyce:

Design and implementation of tutorials for PLC B&R Automation

Stručná charakteristika problematiky úkolu:

Cílem bakalářské práce je vytvoření jednoduchých návodů pro práci na programovatelných automatech firmy B&R Automation.

Návody pro cvičení doplněné o ilustrace budou tvořit významnou část práce a budou připojeny jako příloha. Samotná práce se seznámí čtenáře s HW a SW produkty firmy B&R Automation, představí navržené úlohy a uvede vzorová řešení v různých programovacích jazycích podporovaných užitým PLC.

Cíle bakalářské práce:

- 1) Úvod k technologiím B&R Automation.
- 2) Přehledný popis úloh včetně jejich řešení.
- 3) Vypracování laboratorních návodů demonstrujících práci se zvoleným PLC B&R Automation.

Seznam odborné literatury:

- [1] ŠVARC, I., ŠEDA, M., VÍTEČKOVÁ, M. Automatické řízení. 1. vyd. Brno, CERM., 2007. 324 s. ISBN 978-80-214-3491-2.
- [2] NĚMEC, Z. Prostředky automatického řízení: elektrické. 2. vyd. VUT Brno, 2008.
- [3] ZEZULKA, F. Prostředky průmyslové automatizace. VUT Brno, 2004. 161 s.
- [4] B&R Automation. [on-line] <http://www.br-automation.com/>

Vedoucí bakalářské práce: Ing. et Ing. Stanislav Lang

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2013/2014.

V Brně, dne 16.12.2013

L.S.

Ing. Jan Roupec, Ph.D.
Ředitel ústavu

prof. RNDr. Miroslav Doupovec, CSc., dr. h. c.
Děkan fakulty

ABSTRAKT

Tato bakalářská práce se zabývá návrhem a realizací úloh pro PLC B&R a podrobným návodem práce v Automation Studio. Obsahuje také informace o firmě B&R Automation Ges. m. b. H. a jejích produktech. Návrh úloh probíhal ve vývojovém prostředí B&R Automation Studio. Praktická úloha na regulaci teploty byla řešena a odzkoušena na B&R PLC X20.

ABSTRACT

This bachelor's thesis deals with the design and the implementation tutorials for PLC B&R as well as the detailed instructions of work in the Automation Studio. It contains information about the company B&R Ges. m. b. H. and its products. The design tutorials have been made in the Automation Studio. The practical task has been solved and tested in B&R PLC X20.

KLÍČOVÁ SLOVA

PLC, B&R, Programovatelný automat, B&R X20, Automation Studio

KEYWORDS

PLC, B&R, Programmable Logic Controller, B&R X20, Automation Studio

PROHLÁŠENÍ O ORIGINALITĚ

Tímto prohlašuji, že jsem bakalářskou práci vypracoval samostatně s použitím literatury a zdrojů uvedených v bibliografické citaci.

BIBLIOGRAFICKÁ CITACE

SÝKORA, D. *Návrh a realizace vzorových úloh pro PLC B&R Automation*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014. 41 s. Vedoucí bakalářské práce Ing. et Ing. Stanislav Lang.

PODĚKOVÁNÍ

Děkuji vedoucímu práce Ing. et Ing. Stanislavu Langovi za vstřícný přístup, cenné rady a připomínky při zpracování bakalářské práce. Dále firmě B&R Automation Ges. m. b za přínosné školení. Poděkovat bych chtěl také své rodině a přítelkyni za podporu.

OBSAH

| | |
|--|-----------|
| ZADÁNÍ ZÁVĚREČNÉ PRÁCE | 3 |
| ABSTRAKT | 5 |
| PROHLÁŠENÍ O ORIGINALITĚ | 7 |
| PODĚKOVÁNÍ | 9 |
| 1 ÚVOD..... | 13 |
| 2 B&R AUTOMATION | 15 |
| 2.1 Společnost B&R Automation Ges. m. b. H. | 15 |
| 2.2 Produkty B&R | 16 |
| 2.2.1 Průmyslová PC a vizualizace | 16 |
| 2.2.2 Bezpečnostní technika a řízení pohybu | 16 |
| 2.2.3 Programovatelné automaty (Systém X20) | 17 |
| 2.3 Komunikační standarty | 18 |
| 2.3.1 Powerlink | 18 |
| 2.3.2 X2X Link..... | 18 |
| 2.3.3 Profibus, DeviceNet, Can..... | 18 |
| 3 AUTOMATION STUDIO..... | 19 |
| 3.1 Úvodní strana a nový projekt..... | 19 |
| 3.2 Vytvoření programu a pracovní prostředí | 20 |
| 3.3 Práce se simulátorem | 21 |
| 3.4 Popis konfigurace reálného PLC, konfigurace vstupů a výstupů | 21 |
| 3.5 Diagnostické nástroje | 23 |
| 3.5.1 Monitorování a analýza hodnot..... | 23 |
| 3.5.2 Sběr systémových informací | 24 |
| 3.5.3 Softwarová analýza při programování | 24 |
| 4 PROGRAMOVACÍ JAZYKY..... | 25 |
| 4.1 Grafické jazyky..... | 25 |
| 4.2 Textové jazyky | 26 |
| 5 PŘÍKLADY FYZIKÁLNÍCH A LABORATORNÍCH ÚLOH..... | 29 |
| 5.1 Regulace teploty na laboratorním přípravku | 29 |
| 5.2 Praktické úlohy | 32 |
| 5.2.1 Osvětlení a vlakový přejezd | 32 |
| 5.2.2 Výrobní linka na vstřikování plastů | 33 |
| 6 ZÁVĚR..... | 37 |

| | |
|--|-----------|
| SEZNAM POUŽITÉ LITERATURY | 39 |
| SEZNAM PŘÍLOH | 41 |

1 ÚVOD

V dnešní době složitých výrobních procesů je již nezbytností použití výpočetní techniky jako řídicího prvku. Právě tuto úlohu v automatizaci zastupují programovatelné automaty (PLC), které se oproti klasickým počítačům vyznačují velkou odolností a přizpůsobivostí v průmyslových podmínkách.

Tato bakalářská práce se zabývá programovatelnými automaty firmy B&R Automation Ges. m. b. H., které jsou na dnešním trhu špičkou v oboru. V této práci budou demonstrovány přednosti produktů firmy, jako je například programové prostředí Automation Studio. V bakalářské práci byly zpracovány dvě přílohy, které budou sloužit studentům Vysokého učení technického v Brně jako podpůrné materiály ve cvičeních.

2 B&R AUTOMATION

2.1 Společnost B&R Automation Ges. m. b. H.

B&R Automation Ges. m. b. H. (dále jen „B&R“) je společnost, která byla založena roku 1979 v Rakousku a na českém trhu působí od roku 1997. Své hlavní sídlo má v Eggelsbergu, kde v dnešní době provozuje 3 závody. V moderních halách se nachází vše od plně automatických výrobních linek, až po školící střediska Automation Academy [1].

Firma B&R se během více než 30 let stala jedním z hlavních výrobců v oblasti automatizace. V roce 1980 přišla na trh s čítačem Unicount a také programovatelným PLC Unicontrol, který byl ve své době moderním řešením automatizace. Dalším důležitým datem v historii firmy je rok 1993, kdy firma představila jeden ze svých prvních průmyslových počítačů (IPC2000). V letech 1999 a 2000 se firma podílela na vývoji komunikačního standardu Ethernet Powerlink, který od té doby využívá ve svých produktech. V tomto období také vyvinula a začala nabízet software Automation Studio, který v dnešní době již dosáhl 4. generace [1] [20].

Dnes má společnost přibližně 1700 zaměstnanců a z toho 30 v České republice. B&R patří mezi přední dodavatele automatizace v České republice. Činnost společnosti vychází z centra v Brně a má podpůrná lokální pracoviště v Praze, Plzni a Zábřehu [1].

Společnost B&R má 168 poboček po celém světě a dceřiné společnosti ve světových zemích, mezi které patří např.: Belgie, Brazílie, Kanada, Čína, Česká republika, Dánsko, Švédsko, Švýcarsko, Ukrajina, USA a další. Automatizační řešení společnosti B&R využívá mnoho firem zabývajících se obalovou technikou, dopravou, výrobou papíru, textilu, automobilů, potravin, nápojů atd [1].



Obr. 1 Hlavní sídlo v Eggelsbergu [1]

2.2 Produkty B&R

B&R zajišťuje kompletní řešení pro průmyslovou automatizaci, nabízí vše potřebné pro automatizaci strojů, od jednoduchých až po velmi rozsáhlé systémy. Produkty B&R se zakládají na hladké spolupráci mezi sebou. Firma nabízí k programování, konfiguraci a vizualizaci všech svých produktů software Automation Studio.

2.2.1 Průmyslová PC a vizualizace

B&R nabízí celou řadu průmyslových PC, u kterých je kladen velký nárok na odolnost a dlouhodobou spolehlivost, stejně tak musí být zachována výkonnost PC. Průmyslové PC lze ovládat funkčními klávesy, kapacitním displejem, nebo kombinací obou. Průmyslové PC od B&R tvoří výkonný výpočetní systém, který lze použít i v omezených prostorových podmínkách [1].

Nabídka firmy je široká, od přenosných panelů až po multitouch panely s vysokým rozlišením. Displeje poskytují přehled o strojích a systémových stavech daného systému [1].



Obr. 2 Ovládání kapacitním displejem [1]



Obr. 3 Bezpečnostní technika [1]

2.2.2 Bezpečnostní technika a řízení pohybu

Při použití bezpečnostní techniky se požaduje chytrá a bezpečná reakce na neočekávané situace. Cílem je zajištění bezpečnosti, aniž by se musel zastavit výrobní proces. Firma B&R nabízí celou řadu bezpečnostní techniky, která pracuje na bezpečnostním komunikačním protokolu OpenSafety. Bezpečnostní protokol OpenSafety je první zcela otevřený protokol pro přenos dat s funkční bezpečností, jehož doba komunikačních cyklů se pohybuje řádově kolem mikrosekund a je nezávislá na typu sběrnice [1] [19].

K pohonům B&R patří i víceosé systémy, nebo například hnací součásti s krytím IP65 pro obzvláště drsné podmínky. Firma poskytuje širokou nabídku motorů (kompaktních, asynchronních atd.). V nabídce lze najít převodovky, frekvenční měniče, servopohony a další [1] [19].

2.2.3 Programovatelné automaty (Systém X20)

B&R PLC pokrývá průmyslové požadavky s jediným systémem. V podstatě se jedná o průmyslové zařízení určené k montáži na lištu. Obsahuje standardní počítačový procesor, matematický koprocesor, paměť RAM, SRAM a další komponenty běžných PC. Tento systém může být naprogramován pomocí Automation Studio.



Obr. 4 Systém X20 [1]

Systém X20 lze z konstrukčního provedení rozdělit do tří základních rozebíratelných částí:

- a) Svorka
- b) Modul s elektronikou
- c) Sběrníkový modul

Z důvodu lepšího zapojování byly použity pružinové svorky, které lze zapojovat bez použití nástrojů. Díky tvaru svorek lze zapojit až dva vodiče. Mechanické rozdělení má mnoho výhod, jednou z nich je například možnost vyměňování elektroniky za provozu, nebo zapojování bez elektroniky, kdy elektronika může být zakoupena a přidělena, až po sestavení celého stroje. Pomocí aretační páčky lze vyměňovat I/O moduly [1] [16] [17] [18].

B&R nabízí několik modelů procesorových jednotek ve více výkonnostních třídách. K ukládání dat využívá systém X20 flash paměť s kapacitou až 4 GB. V zařízeních B&R PLC jsou rozhraní RS232, Ethernet a USB. Každá procesorová jednotka má konektor Powerlink a CAN pro komunikaci v reálném čase. Některé řídicí jednotky jsou zcela bezúdržbové, protože neobsahují žádné ventilátory ani baterie. Nejdelší spojení přes X2X Link mezi dvěma sousedními I/O moduly je 100 m [1] [16] [17] [18].

2.3 Komunikační standarty

2.3.1 Powerlink

B&R využívá komunikační standart Powerlink. Powerlink je systém vycházející ze standartu Ethernet a využívá komunikační model producent/konzument, což vede k větší propustnosti sítě a lepšímu výkonu. Tím že Powerlink vychází ze standartu Ethernet, tak nepotřebuje žádný dodatečný hardware a lze pro něj využít všechny čipy, systémy vytvořené pro Ethernet. Powerlink je ideálním protokolem pro pohony, vstupy a výstupy, vizualizaci a výměnu dat mezi systémy automatů [1] [11] [12] [13] [14].

Protokol Powerlink rozšiřuje Ethernet podle normy IEEE 802.3 o smíšený mechanismus pro dotazování a rozdělování času. Protokol Ethernet Powerlink je cyklický a lze zde nastavit dobu cyklu. Stanice na síti mohou vysílat dlouhé rámce dat a v každém cyklu sběrnice je místo i pro asynchronní datový rámec B&R Automation Net (a tedy i TCP/IP). Všichni účastníci, kteří jsou připojeni k této síti, mohou být synchronizováni. Velkou výhodou protokolu Ethernet Powerlink je možnost nastavovat parametry komunikace tak, že si je uživatel může přizpůsobit svým požadavkům. Organizace IAONA, zastřešující v Evropě aplikace Ethernetu v průmyslové praxi, jej jako jediný protokol zařadila do nejvyšší čtvrté třídy sítí. O efektivnosti a výjimečnosti Ethernetu Powerlink vypovídá například využití, kde v cyklu 2,4 ms navzájem komunikuje padesát v/v stanic a přes padesát servozesilovačů, to vše s rozptylem menším než 1 μ s. Dalším příkladem je synchronizace devatenácti os v cyklu 800 μ s [1] [11] [12] [13] [14].

2.3.2 X2X Link

Všechny moduly X20 jsou mezi sebou spojeny sběrnici X2X Link. Standart X2X Link spojuje moduly vedle sebe, nebo umožňuje spojení dvou modulů vzdálených až 100 m od sebe. Každý modul má přijímač X2X a vysílač X2X. Všechna data, která dorazí na přijímač, jsou okamžitě předána na vysílač [1].

X2X protokol definuje pevnou délku cyklu, ve kterém se opětovně odesílají čtyři základní části (SyncOut, SyncIn, AsynOut, AsynIn). Nejrychlejší možný časový cyklus je 100 μ s. Časové cykly se konstantně opakují [1].

2.3.3 Profibus, DeviceNet, Can

Profibus je jednoduchý vysokorychlostní protokol pro klasické automatizační úlohy v oblasti výrobního inženýrství. Hlavním rysem protokolu Profibus je poskytování extrémně bezpečných procesů [1] [14].

DeviceNet je otevřený standart průmyslové sběrnice. Specifikace protokolu a údržba standartu DeviceNet jsou pod dohledem nezávislé a otevřené uživatelské organizace ODVA (Open DeviceNet Vendor Association) [1] [14].

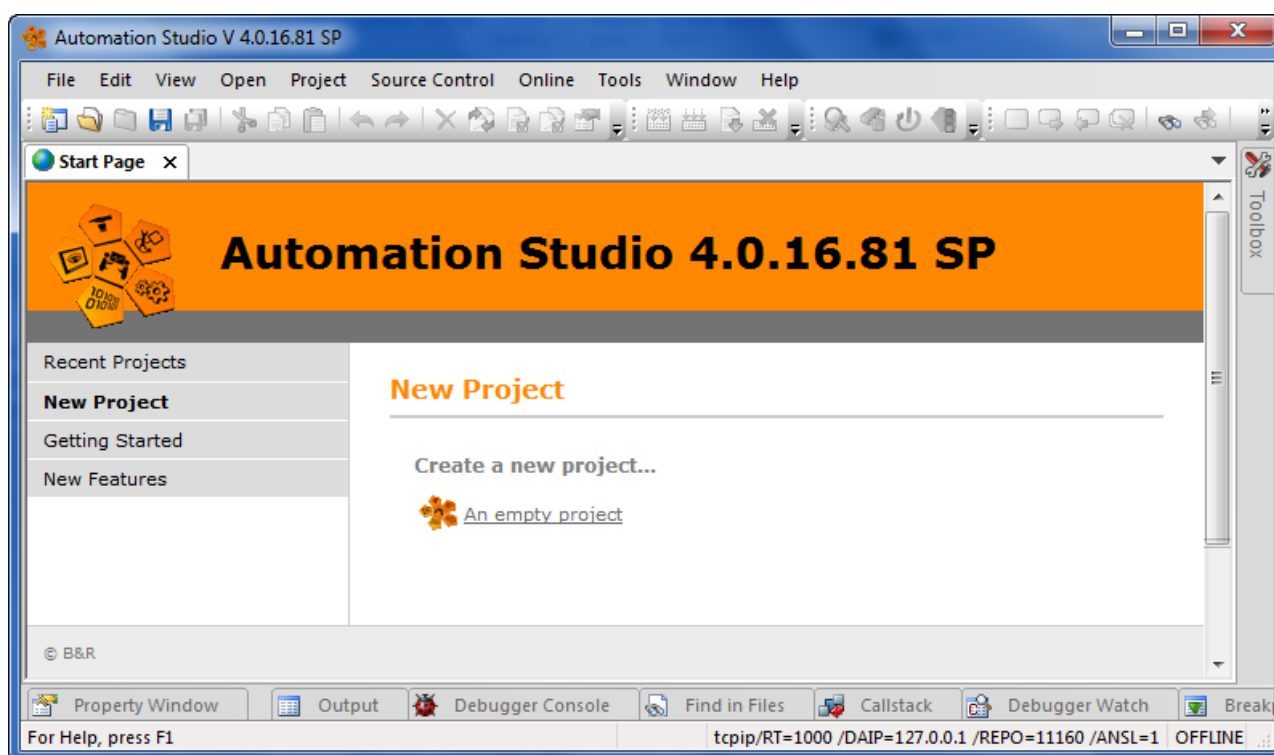
Can (Controller area network) sběrnice je hojně používaná pro vybrané B&R I/O systémy, které fungují přes speciální protokol na CAN sběrnici [1].

3 AUTOMATION STUDIO

Automation Studio je programové prostředí vyvinuté pro B&R produkty, jako jsou například regulátory, řídicí jednotky nebo vizualizační jednotky. Uživatel si může vybrat z mnoha programovacích jazyků, diagnostických nástrojů a editorů k vytvoření svých projektů.

3.1 Úvodní strana a nový projekt

Po nainstalování a zapnutí programu Automation Studio se otevře úvodní strana. Z této strany může být vytvořen nový projekt, nebo otevřen a editován již existující projekt.



Obr. 5 Úvodní strana

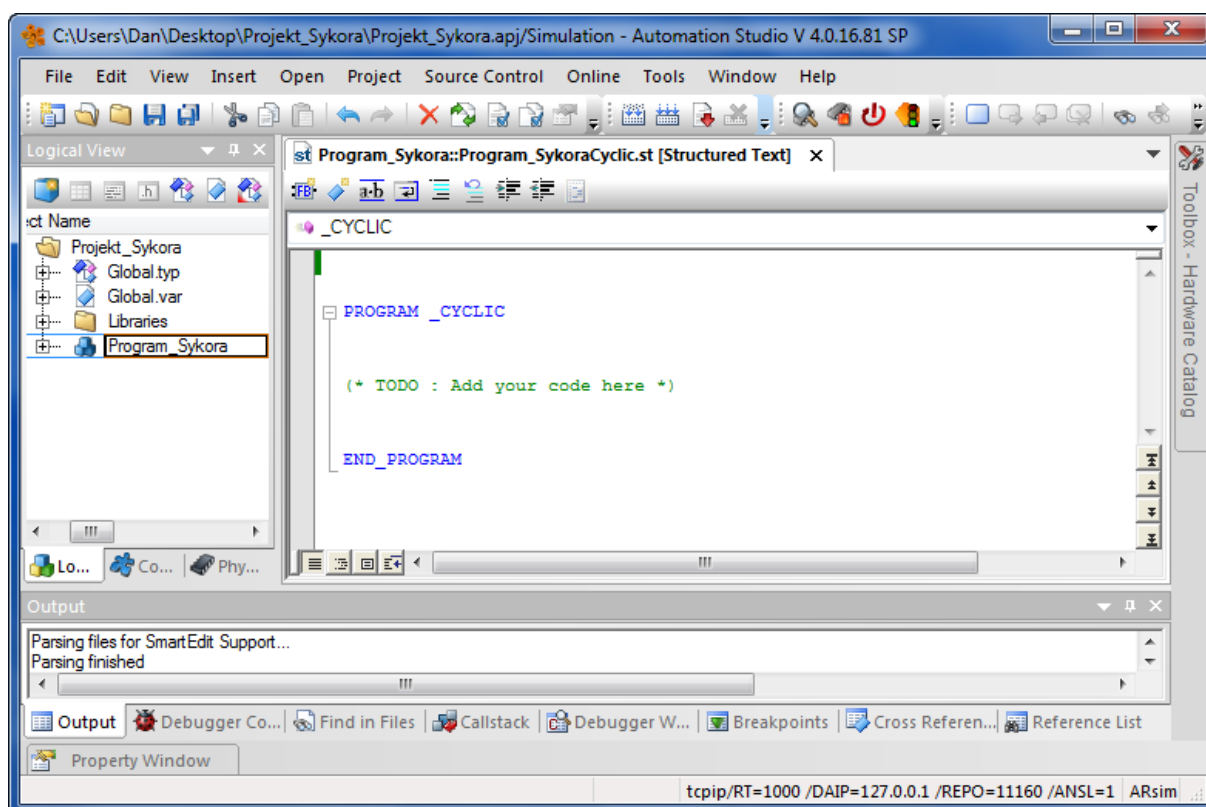
Nové projekty se vytvoří pomocí záložky New projekt nebo ikonou na horní liště. V nastavení projektu lze vyplnit název projektu, popis projektu a také lze vybrat kopírování již vytvořeného Automation Runtime.

Součástí programu Automation Studio je operační systém reálného času Automation Runtime. Tento operační systém v reálném čase tvoří softwarový základ, který umožňuje spustit projekty na cílovém systému. Automation Runtime je součástí všech B&R produktů, které mohou mezi sebou přenášet jednotlivé konfigurace. Další výhodou Automation Runtime je deterministické chování a možnost využití osmi různých cyklických tříd. Podporuje řadu programovacích jazyků.

3.2 Vytvoření programu a pracovní prostředí

Automation Studio je rozděleno do několika částí, každá z nich má své specifické funkce a úlohy.

- Menu a pracovní lišta poskytují přístup k rozsáhlým funkcím
- V levé části pracovního prostředí se nachází organizátor projektu, který obsahuje tři základní záložky, kterými se vytváří nebo editují komponenty projektu (*Logical View*, *Configuration View* a *Physical View*)
- V pravé části pracovního prostředí se nachází hlavní okno projektu, kde lze programovat a konfigurovat data projektu
- Dolní část pracovního prostředí obsahuje výstupní okno, které zobrazuje informace při kompilaci projektu



Obr. 6 Pracovní prostředí

Po vytvoření programu v levé části pracovního prostředí uživatel volí v posledním okně možnost vykonání Init programu (provede se jen jednou na začátku), nebo Exit programu (provede se vždy jen jednou na konci). V této části si také uživatel volí programovací jazyk. Každý vytvořený projekt obsahuje globální proměnné, které jsou využívány pro celý projekt. Dále jednotlivé programy obsahují lokální proměnné, které jsou deklarovány a využívány jen daným programem. Lze také vytvořit vlastní datové typy (např. struktury), které se dají dále vnořovat. Stejně jako proměnné, tak i struktury můžeme rozdělit na globální (*Global.typ*) a lokální (*Program.type*).

Nedílnou součástí každého projektu jsou také knihovny. Lze vytvářet i vlastní knihovny a poté je využívat ve svých projektech anebo si stáhnout již vytvořené knihovny.

3.3 Práce se simulátorem

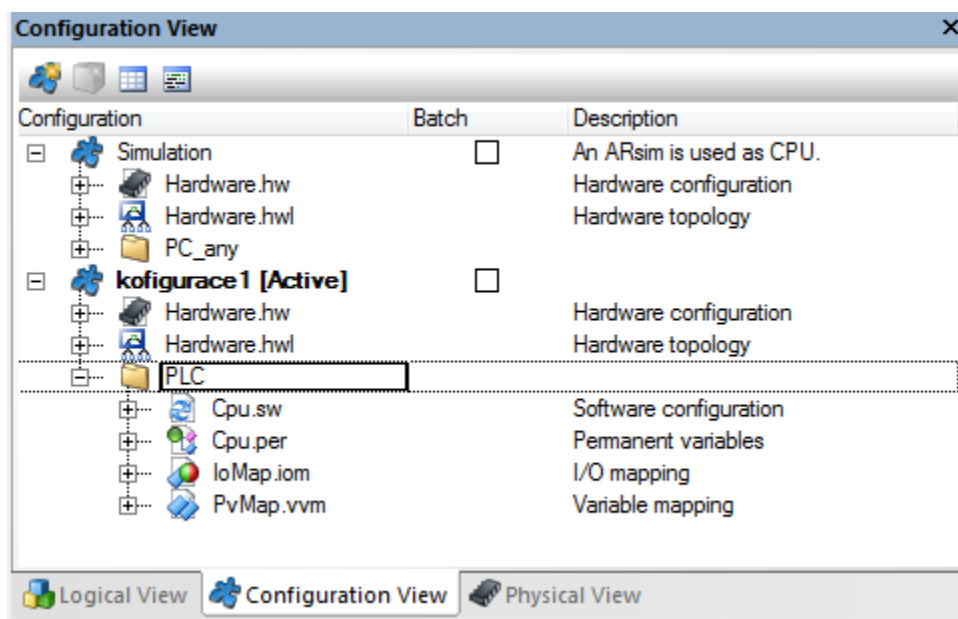
Práce se simulátorem je nedílnou součástí každého projektu. Simulace umožňují simulovat konfigurace, jako kdyby byly na skutečném PLC procesoru. Simulace jsou převážně využívány v rané části projektu, kdy eliminují složité hardwarové zapojení a zároveň umožňují rychlé testování projektu.

V záložce *Logical View* se vytváří programy. V záložce *Physical View*, která je hardwarovou částí organizátoru, přiřadíme do taskových tříd vytvořené programy. Automation Studio nabízí 8 základních taskových tříd, které se liší svou rychlostí cyklu (rychlejší taskové třídy jsou náročnější na procesor). Pro přesun projektu se používá tlačítko *Transfer*, které se nachází v horní liště. Vytvořený program se nahraje do simulátoru. Při práci se simulátorem se simulace sleduje pomocí diagnostického nástroje *Watch*.

3.4 Popis konfigurace reálného PLC, konfigurace vstupů a výstupů

V záložce *Configuration View* se řídí hlavní konfigurace projektu. Konfigurace rozhoduje o tom, jaký hardware a software je právě aktivní. Hardwarová konfigurace se vytvoří ikonou *Add Configuration*, poté se otevře okno, kde lze automaticky vyhledat softwarovou strukturu, nebo ručně vybrat. Poté se do záložky *Configuration View* přidá nově vytvořená konfigurace. Nachází se zde i dříve vytvořená konfigurace simulátoru.

Jen jedna konfigurace může být aktivní [**Active**]. Jinou konfiguraci lze aktivovat dvojklikem a tato aktivovaná konfigurace se zobrazí v záložce *Physical View*. Projekty většinou obsahují více programů a konfigurací. Programátor nakonfiguruje v rané části projektu více možných konfigurací, které se poté otestují ve výrobě, kde se vybere ideální konfigurace pro dané provozní podmínky.



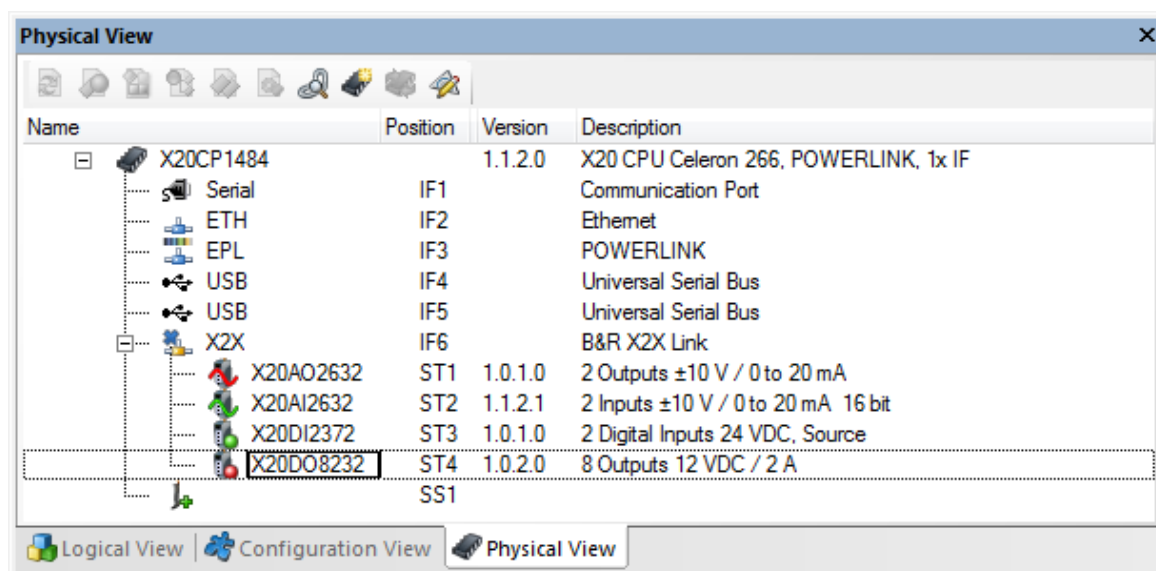
Obr. 7 Konfigurace projektu

Hardwarové části systému se konfiguruji a ovládají v záložce *Physical View*.

Následující úkony jsou prováděny v záložce *Physical View*:

- Přidání a konfigurace I/O modulů
- Přiřazení proměnných do I/O datových bodů
- Konfigurace rozhraní sítě
- Přidání a konfigurace hardwarových modulů

Do *Physical View* se jednotlivé moduly přiřazují liniově, tak jak jsou zapojeny hardwarově vedle sebe komunikací X2X. Jednotlivé proměnné se přiřazují na moduly pomocí „I/O mapping“ v záložce *Physical View*.



Obr. 8 Hardwarová část projektu

Při prvním přesunu programu z Automation Studio do reálného PLC se využívá generování Compact flash. Automation Studio rozdělí Compact flash na více diskových oddílů, jejichž velikost lze nastavit. Po vygenerování stačí vložit Compact flash do vypnutého PLC. Jedním z hlavních důvodů vytváření Compact flash je přesun operačního systému Automation Runtime. Další upravené konfigurace lze nahrávat do PLC pomocí komunikačních standartů.

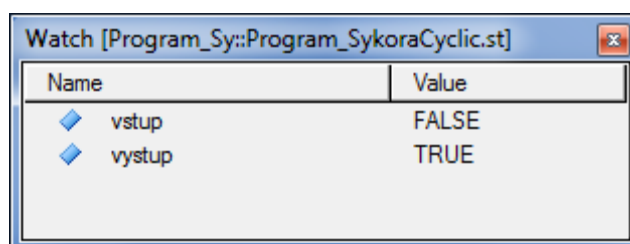
3.5 Diagnostické nástroje

Automation Studio poskytuje širokou škálu diagnostických nástrojů k optimalizaci projektu a hledání chyb. Od prostého sledování I/O stavů, sledování proměnných stavů v průběhu času až po ladící nástroje knihoven.

Výběr správného diagnostického nástroje je rozhodující pro rychlé řešení problému.

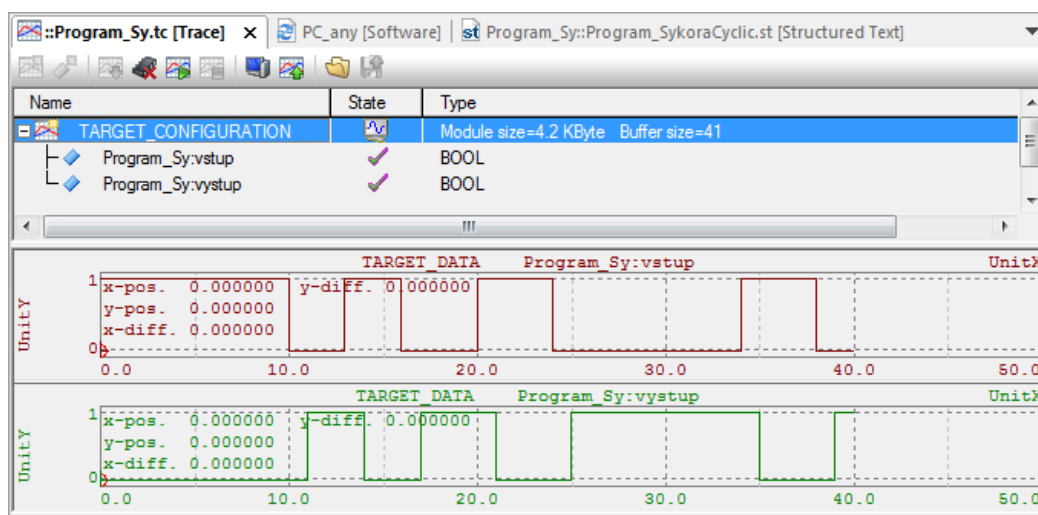
3.5.1 Monitorování a analýza hodnot

- 1) Watch (Variable watch) - umožňuje zobrazovat a upravovat hodnoty proměnných. Kromě hodnot také zobrazuje důležité informace (typ dané proměnné, rozsah).



Obr. 9 Watch

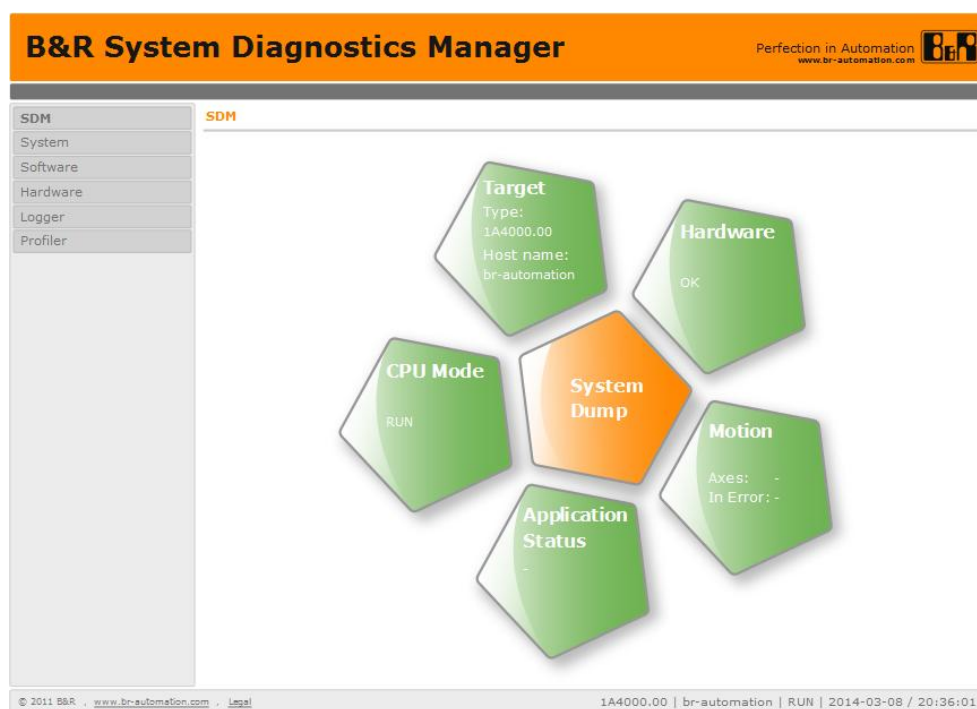
- 2) Trace (Variable oscilloscope) - zaznamenává data několika proměnných v reálném čase po stanovenou dobu. Tyto data lze využít k zobrazení ve formě křivky, či jiné grafické interpretace.



Obr. 10 Trace

3.5.2 Sběr systémových informací

- 1) Logger - zobrazuje události, které se zobrazují v rámci projektu. Mezi tyto události patří například Fatal errors (např. časová havárie), warnings, systém messages aj.
- 2) Status bar - zobrazuje informace o připojení.
- 3) Systém Diagnostics Manager - lze použít k diagnostice pomocí standardního internetového prohlížeče z jakéhokoliv místa. Jedinými požadavky je přítomnost prohlížeče a připojení k internetu.



Obr. 11 B&R Systém Diagnostics Manager

3.5.3 Softwarová analýza při programování

- 1) Debugger - usnadňuje hledání chyby ve zdrojovém kódu.
- 2) Profiler - může být použit k měření a zobrazení důležitých systémových dat. Profiler může také zaznamenávat a analyzovat události určené uživatelem (uživatelské akce). Výsledky měření jsou uvedeny v tabulkové formě jako minimální, maximální a průměrné hodnoty, nebo v grafu.

4 PROGRAMOVACÍ JAZYKY

Norma IEC 1131-3 sjednocuje programovací jazyky pro PLC. Automation Studio nabízí mnoho programovacích jazyků: B&R Automation Basic, Instrukční list, Ladder diagram, Schéma funkčních bloků, Strukturovaný text, SFC, CFC, ANSI C, ANSCI C++.

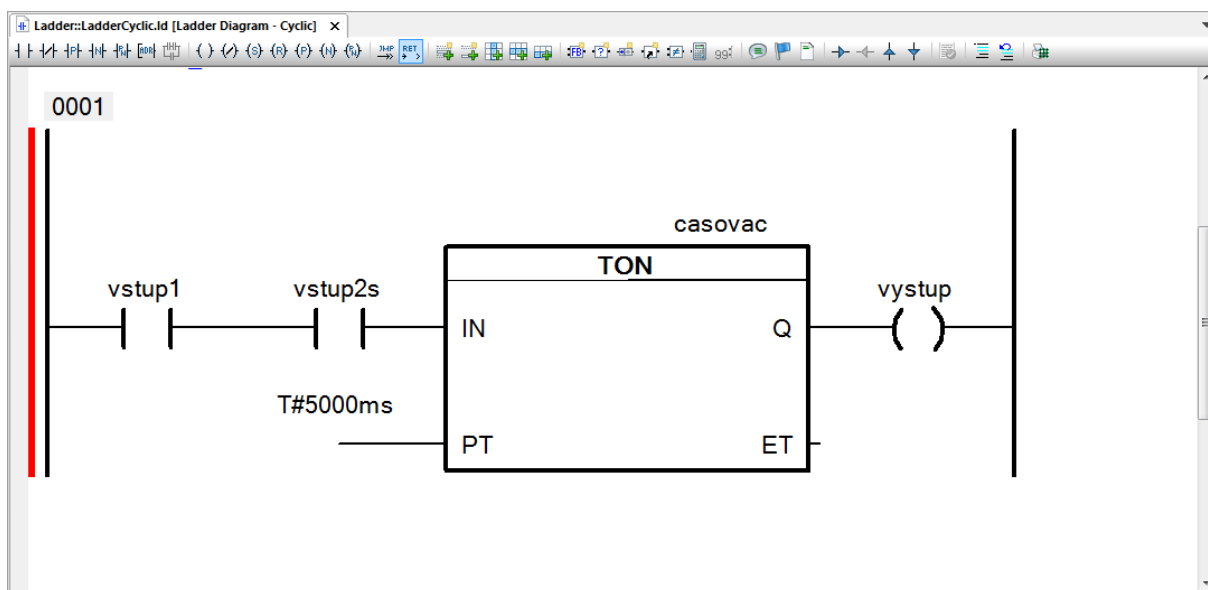
Pro demonstraci programovacích jazyků byl vytvořen program. Popis programu je následující: po zapnutí vypínačů *vstup1* a *vstup2* se po 5 sekundách nastaví světlo *vystup* do logické 1, při vypnutí jednoho z vypínačů se světlo nastaví do logické 0.

4.1 Grafické jazyky

a) Ladder diagram

Ladder diagram (LD) je jedním ze základních grafických jazyků využívaných v programu Automation Studio. LD je založen na grafické interpretaci reléové logiky a je podobný liniovým schémátům v elektrotechnice. Mezi dvěma svislými čarami (které se nazývají levá a pravá napájecí sběrnice) se nachází síť kontaktů, cívek, funkcí a funkčních bloků. Tato sérioparalelní síť je zapojena zleva doprava.

LD se využívá při programování jednodušších logických úloh, může jej obsluhovat i personál, který nezná pokročilejší programovací jazyky. Při složitějších a delších instrukcích se stává LD již nepřehledný a je lepší zvolit jiný programovací jazyk [2] [3] [4] [5].



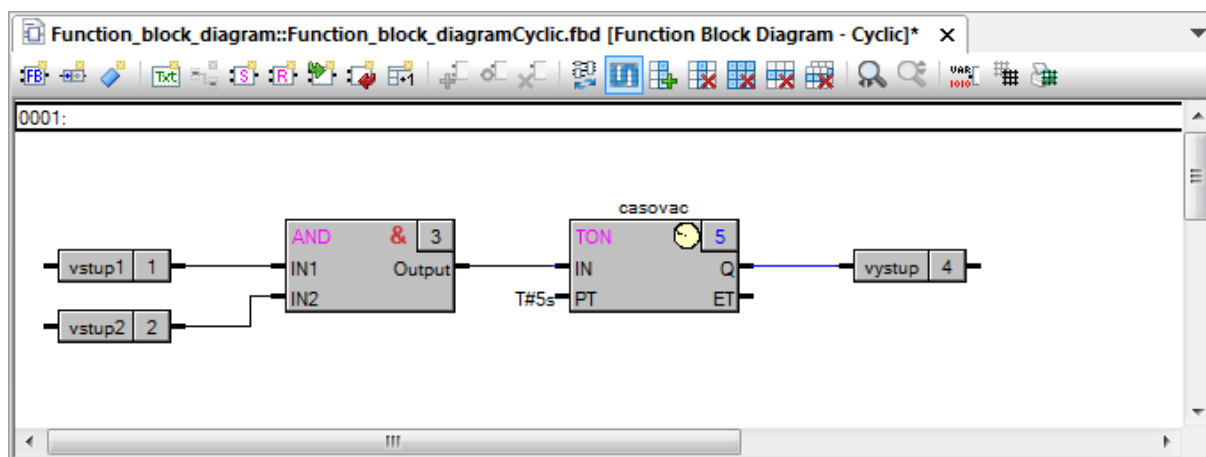
Obr. 12 Ladder diagram

b) Schéma funkčních bloků

Schéma funkčních bloků (FBD) je dalším grafickým jazykem využívaným v programu Automation Studio. FBD je podobný LD, ale na rozdíl od něj využívá na místo reléové logiky vzájemné propojení obdélníkových značek výrokové logiky, čítačů, časovačů atd.

Jednou z největších výhod FBD je online sledování stavů. FBD je ideální pro programy tvořené digitálními vstupy (senzory). Není vhodný pro realizaci složitějších funkcí

a delších programů, protože jako většina grafických programovacích jazyků ztrácí na přehlednosti a pozdější korekce programu jsou s rostoucí délkou složitější [2] [3] [4] [5].



Obr. 13 Schéma funkčních bloků

4.2 Textové jazyky

a) Strukturovaný text

Strukturovaný text (ST) je jeden z nejpoužívanějších vyšších textových programovacích jazyků v programu Automation Studio. Svou strukturou a příkazy je velice podobný Pascalu a ANSI C. ST obsahuje klasické prvky vyšších programovacích jazyků, jako jsou např. příkazy větvení (IF, CASE OF), nebo smyčky (FOR, WHILE). Všechny příkazy jsou odděleny středníkem.

ST se využívá v programech, kde se složitější operace (např. práce s databázemi) naprogramují mnohonásobně rychleji, než u grafických jazyků. ST je vhodný pro práci s daty a řetězci. Velký nárok je kladen na znalost všech příkazů, syntaxí a na schopnosti programátora. ST není vhodný při zpracování velkého počtu logických vstupů a výstupů [2] [3] [4] [5].

```

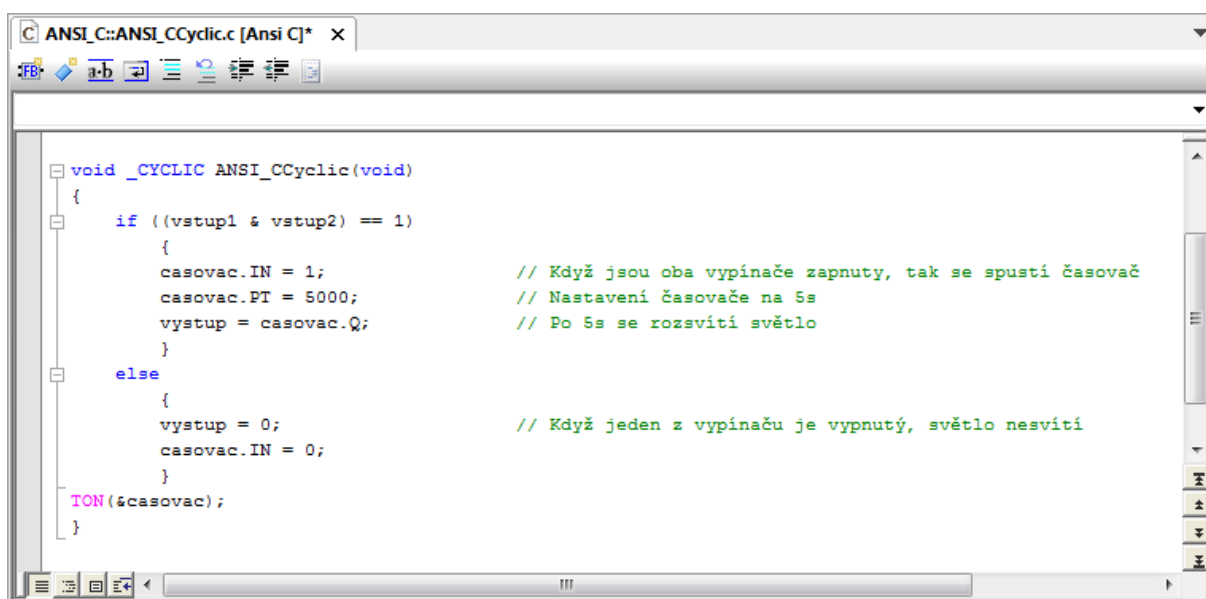
PROGRAM _CYCLIC
  IF vstup1 AND vstup2 = 1 THEN
    start:=1; // Když jsou oba vypínače zapnuty, tak se spustí časovač
    IF casovac.Q=1 THEN
      vystup:=1; // Po 5s se rozsvítí světlo
    END_IF
  ELSE vystup:=0; // Když jeden z vypínačů je vypnutý, světlo nesvítí
    start:=0;
  END_IF
  casovac(IN :=start, PT:=T#5s ); // Nastavení časovače na 5s
END_PROGRAM

```

Obr. 14 Strukturovaný text

b) ANSIC

V programu Automation Studio lze také programovat v ANSIC a ANSIC++. ANSIC je jeden z nejrozšířenějších programovacích textových jazyků pro psaní systémového softwaru ale i pro psaní aplikací. Tento programovací jazyk začíná využívat čím dál víc programátorů pro programování PLC. Programování v ANSIC se doporučuje spíše pokročilým programátorům, jelikož jeho syntaxe je složitější [9].



```
ANSI_C::ANSI_CCyclic.c [Ansi C]* x
void _CYCLIC ANSI_CCyclic(void)
{
    if ((vstup1 & vstup2) == 1)
    {
        casovac.IN = 1;           // Když jsou oba vypínače zapnuty, tak se spustí časovač
        casovac.PT = 5000;       // Nastavení časovače na 5s
        vystup = casovac.Q;     // Po 5s se rozsvítí světlo
    }
    else
    {
        vystup = 0;             // Když jeden z vypínaču je vypnutý, světlo nesvítí
        casovac.IN = 0;
    }
    TON(&casovac);
}
```

Obr. 15 ANSIC

5 PŘÍKLADY FYZIKÁLNÍCH A LABORATORNÍCH ÚLOH

5.1 Regulace teploty na laboratorním přípravku

Přípravek na regulaci teploty je dostupný ve dvou provedeních: laboratorní přípravek (viz Obr. 16), nebo obdobný model přímo v simulačním panelu připojený k PLC (panel 4SIM.00-01).



Obr. 16 Laboratorní přípravek na regulaci teploty

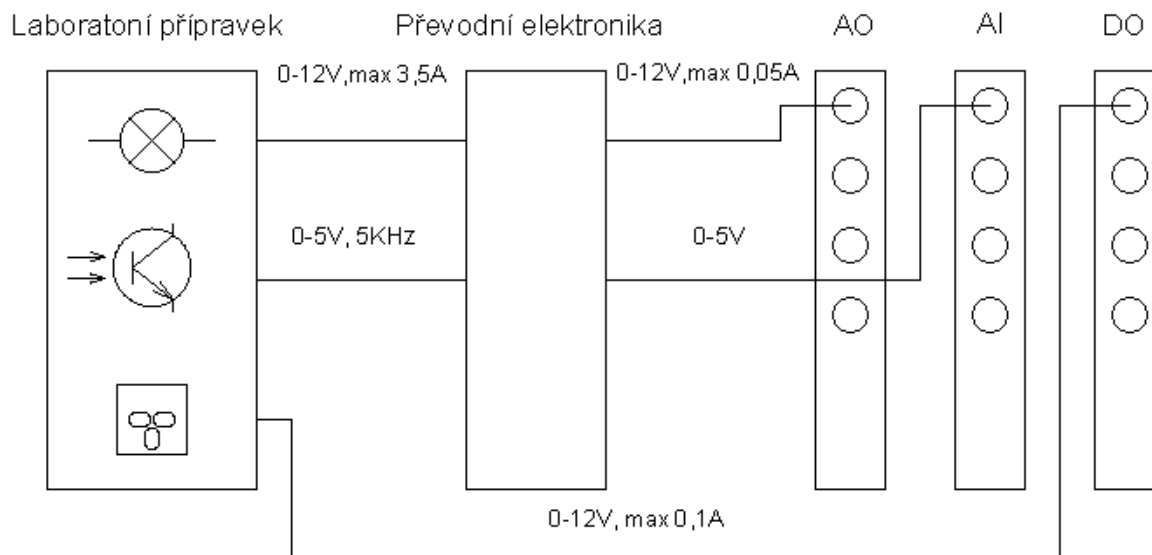
Laboratorní přípravek se skládá ze tří základních částí:

1. Zdroj tepla (Žárovka – 12 V, 35 W) - žárovku je potřeba připojit k analogovému výstupu (AO) přes převodní elektroniku, protože proudová zatížitelnost karty je pouze 0,1 A a žárovka má odběr až 3 A.
2. Snímač (Termistor) - v laboratorním přípravku funkci snímače zastává digitální teploměr s PWM modulovaným výstupem 0-5 V. Poté převodní elektronika převede signál na analogový 0-5 V.
3. Chladič (Ventilátor) – ventilátor je připojen přímo k digitálnímu výstupu automatu. Napěťový rozsah do 12 V a proud maximálně do 0,1 A (karta zvládá až 0,5 A). Ventilátor lze také připojit na speciální karty s PVM výstupy (nezávislost na cyklické

třídě). PVM princip lze také uskutečnit programově pomocí digitálního výstupu. Dalším způsobem by bylo převedení signálu na analogový a připojení přes převodní elektroniku na AO kartu.

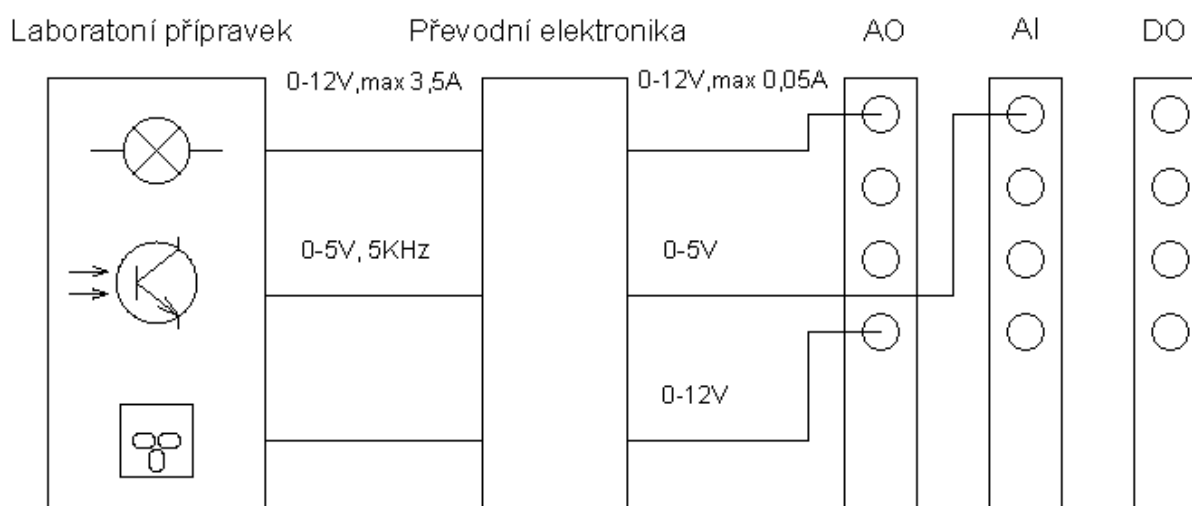
Zapojení laboratorního přípravku:

a) Zapojení ventilátoru na digitální výstup



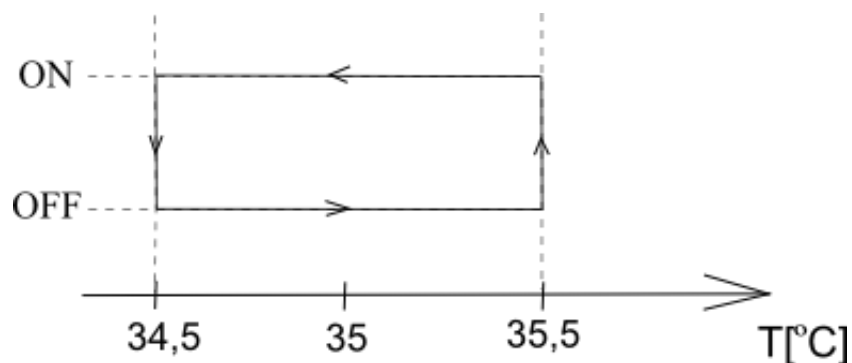
Obr. 17 Schematický obrázek zapojení ventilátoru na digitální výstup

b) Zapojení ventilátoru na analogový výstup



Obr. 18 Schematický obrázek zapojení ventilátoru na analogový výstup

V Automation Studio lze řízení teploty programově realizovat opět více způsoby. Nejjednodušším způsobem je regulace typu ON/OFF. Aby nedocházelo k příliš častému zapínání a vypínání, doplníme o hysterezi $0,5^{\circ}\text{C}$. Úlohu lze realizovat ve dvou variantách. A to sepínám a vypínáním žárovky, při konstantní procentuální hodnotě ventilátoru, nebo naopak sepínám a vypínáním ventilátoru, při konstantní procentuální hodnotě výkonu žárovky. ON/OFF regulace je značně jednoduchá a pro pomalejší děje, jako je třeba regulace teploty naprosto dostačující.



Obr. 19 Teplotní hystereze

Ukázka cyklické části programu:

PROGRAM_CYCLIC

```
t_zadana:=REAL_TO_INT(kolecko1*(100.0/32767)*0.5;
zarovka:=REAL_TO_INT(kolecko2*(100.0/32767));
t_real:=teplota_tranzistor*20;
IF t_real < (t_zadana - 0.5) THEN
    ventilator := 0;
END_IF
IF t_real > (t_zadana + 0.5) THEN
    ventilator := 1;
END_IF
```

END_PROGRAM

Popis:

Na panelu (kolecko1, kolecko2) se nastavuje žádaná hodnota teploty a procentuální výkon žárovky. Žádaná hodnota (t_zadana) byla převedena pomocí přepočtu z datového typu INT na REAL a lze ji nastavovat v rozsahu 0°C - 50°C . Výkon žárovky je možné nastavovat v rozsahu 0-100%. Reálná teplota je měřena pomocí digitálního teploměru v rozsahu 0°C - 100°C . Je-li reálná hodnota (t_real) menší než žádaná hodnota o $0,5^{\circ}\text{C}$ ventilátor se vypne, je-li větší o $0,5^{\circ}\text{C}$ ventilátor sepne.

5.2 Praktické úlohy

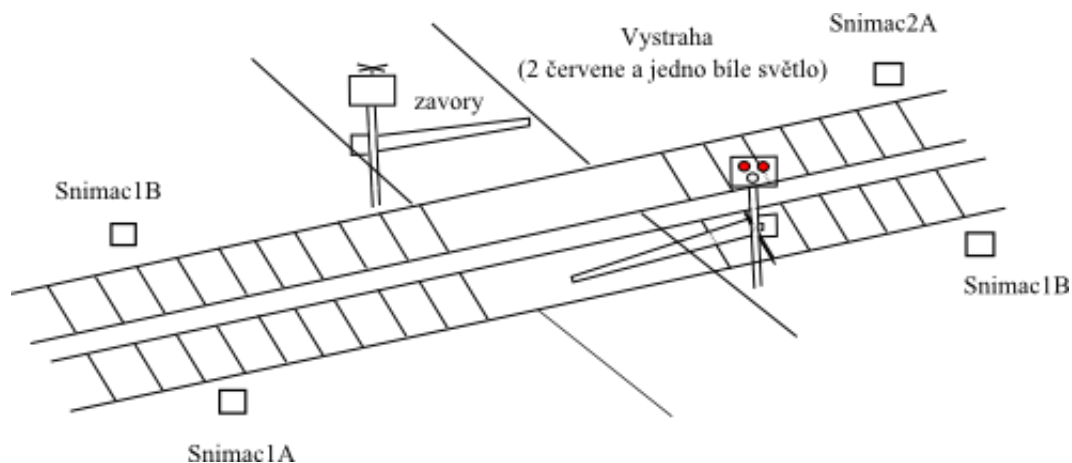
Automation Studio nabízí řadu předprogramovaných projektů, které slouží jako demonstrativní a výukové programy (např. projekt Coffemachine). Tyto projekty byly naprogramovány zkušenými programátory B&R a jsou tedy připraveny k použití v praxi. Použít lze i jen některé části (Init programy, Vizualizace aj.).

V příloze bakalářské práce byly vypracovány 3 praktické úlohy:

5.2.1 Osvětlení a vlakový přejezd

V první úloze se jedná o jednoduché osvětlení, tento program je použit jako demonstrativní příklad v části o programovacích jazycích, kde byl naprogramován ve 4 základních programovacích jazycích.

Druhá úloha řeší automatizaci vlakového přejezdu. Na přejezdu se uvažují dvě koleje se 4 snímači. Na každé koleji jezdí vlak v jednom směru. Snímače (Snimac1A a Snimac2A) detekují příjezd vlaku na přejezd a snímače (Snimac1B a Snimac2B) detekují odjezd vlaku. Při detekci vlaku se spustí výstraha, poté za 5 s se spustí i závory. Po odjetí všech vagonů trvá 5 s, než se zvednou závory a vypne se výstraha. Při zapnuté výstraze střídavě blikají červená světla s periodou 1 s a bílé světlo je vypnuté. Při vypnuté výstraze červená světla nesvítí a bílé světlo bliká s dvojnásobnou periodou, než blikají červená (tj. 2 s). Program viz příloha.



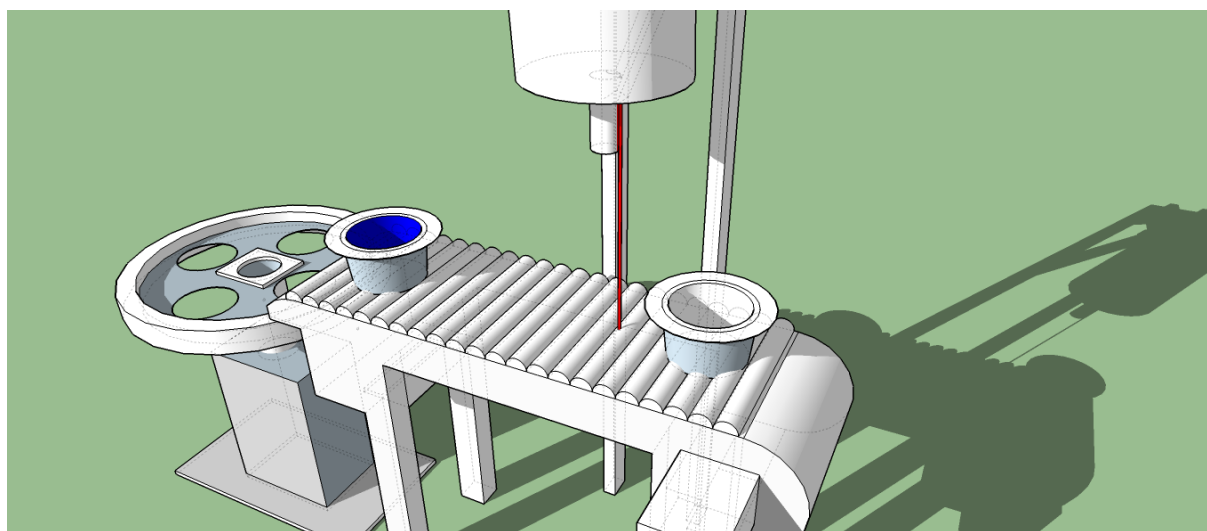
Obr. 20 Schematický obrázek vlakového přejezdu

5.2.2 Výrobní linka na vstřikování plastů

Poslední praktická úloha se zabývala zautomatizovanou linkou na vstřikování plastů do jednoduchých forem. Projekt se skládá ze dvou základních částí (programy řešící dopravní pás a úložnou paletu). Dopravní pás je poháněn asynchronním motorem, takže se formy po něm pohybují konstantní rychlostí. Bylo navrženo, že z předchozí linky přijíždějí prázdné formy každých 5 s. Tekutý plast je v zásobníku s ventilem.

Paleta je rotačního charakteru. Krokový motor sepne, když je plný slot a paleta pootočí tak, aby byl další prázdný slot připraven na příjezd další plné formy. Pro demonstrativní účely se paleta skládá ze 4 prázdných slotů. Jakmile se naplní všechny sloty, na ovládacím panelu se rozsvítí signalizace pro obsluhu, aby vyměnila paletu (Paleta.Vymenit). Při manipulaci s paletou je motor vypnut. Paleta dále pokračuje k dalšímu zpracování.

Dopravní pás obsahuje kontrolní elektroniku, která hlídá správný chod pásu. Při jakémkoliv poruše, nebo při zmáčknutí vypínacího tlačítka (Ovladani.VYP) se okamžitě vypne celá linka. Porucha je signalizována na ovládacím panelu červeným světlem. Při bezproblémovém chodu se na ovládacím panelu vypíše: „Pas v pořadku“. Při poruše se vypíše: „POZOR porucha“.



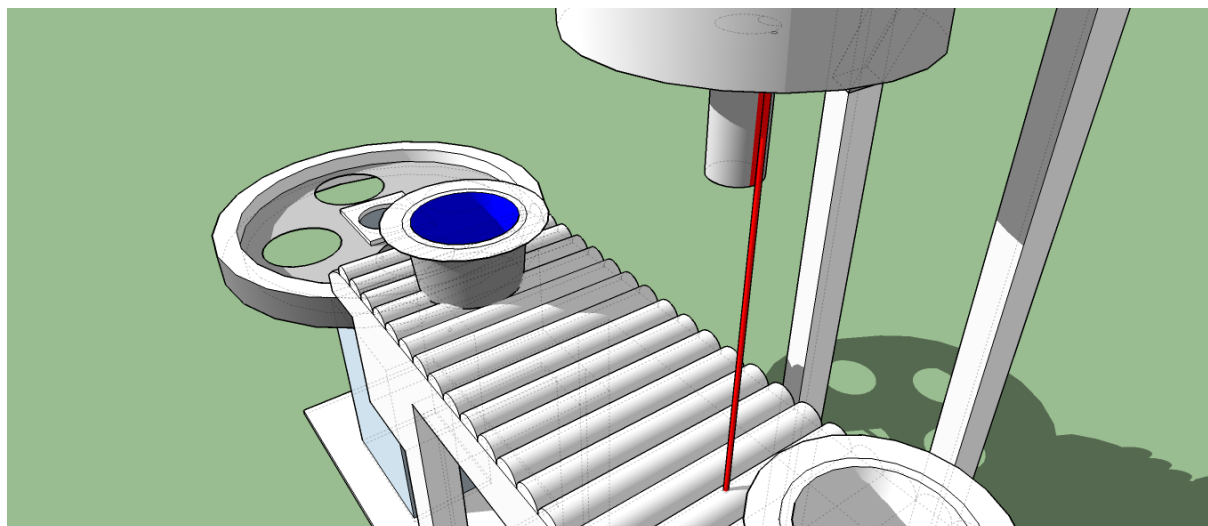
Obr. 21 Model výrobní linky

Pod zásobníkem se nachází snímač (Pas.Forma), který dává logickou 1, když je prázdná forma pod zásobníkem. V ten okamžik se otevře ventil a plast se vstříkne do prázdné formy. Po naplnění formy jede plná forma dále po pásu až do palety. Pod kruhovou paletou se rovněž nachází optický snímač, který dává logickou 1, právě tehdy, když přijede forma do slotu. Program viz příloha.

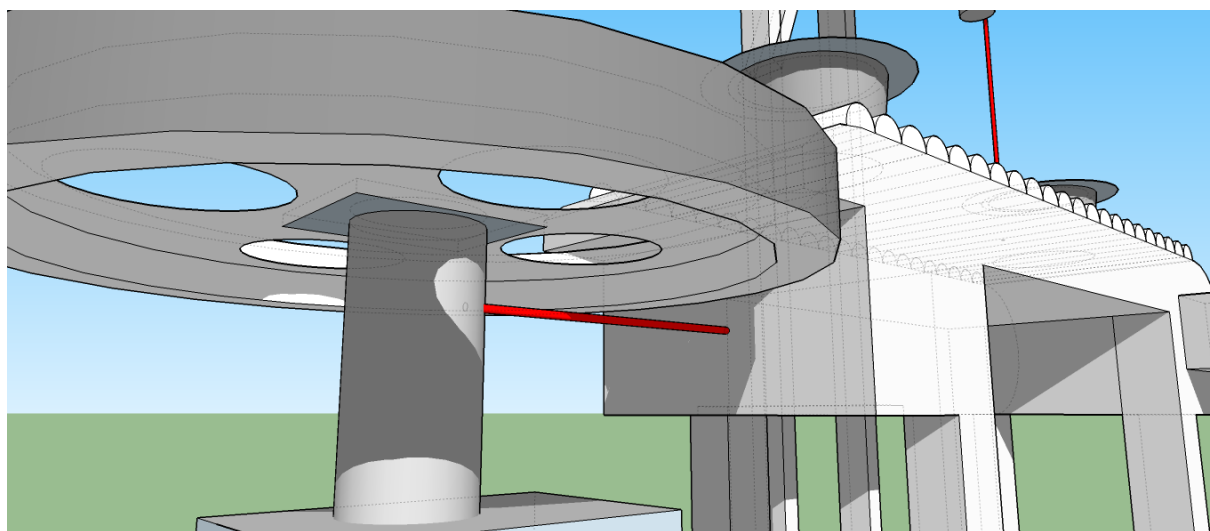
Všechny proměnné byly řešeny pomocí struktur, jelikož to zvyšuje přehlednost projektu.

| Name | Type | & Re... | Value | Description [1] |
|--------------------------|------------|--------------------------|-------|----------------------------|
| [-] tOvladani | | | | Struktura ovládání |
| [-] ZAP | BOOL | <input type="checkbox"/> | FALSE | Zapínací tlačítko |
| [-] VYP | BOOL | <input type="checkbox"/> | TRUE | Vypínací tlačítko |
| [-] tPas | | | | Struktura pásu |
| [-] Motor | BOOL | <input type="checkbox"/> | FALSE | Motor pásu |
| [-] Ventil | BOOL | <input type="checkbox"/> | | Ventil pásu |
| [-] Kontrola | BOOL | <input type="checkbox"/> | FALSE | Kontrolní elektronika pásu |
| [-] Forma | BOOL | <input type="checkbox"/> | | Snímač přítomnosti formy |
| [-] Vystraha | tPOZOR | <input type="checkbox"/> | | Struktura výstrahy |
| [-] tPaleta | | | | Struktura palety |
| [-] Vymenit | BOOL | <input type="checkbox"/> | | Plná paleta |
| [-] Plny_slot | BOOL | <input type="checkbox"/> | | Plný slot |
| [-] Pocet | INT | <input type="checkbox"/> | | Počet prázdných slotů |
| [-] K_Motor | BOOL | <input type="checkbox"/> | | Krokový motor palety |
| [-] tPOZOR | | | | Struktura upozornění |
| [-] Svetelna_signalizace | BOOL | <input type="checkbox"/> | | Červené světlo |
| [-] Upozomeni | STRING[80] | <input type="checkbox"/> | | Nápis na ovládacím panelu |

Obr. 22 Globální proměnné řešené pomocí struktur



Obr. 23 Optický snímač přítomnosti formy pod ventilem



Obr. 24 Optický snímač přítomnosti formy v paletě

6 ZÁVĚR

Cílem této bakalářské práce bylo představení firmy B&R a jejích produktů, vypracování návodu pro práci v programu Automation Studio, dále navrhnutí a zpracování vzorových a praktických úloh v programu Automation Studio.

Úvodní kapitola se zabývá představením firmy B&R, jejími hlavními produkty a využitím komunikačních standartů.

Další kapitola pojednává o programovém prostředí Automation Studio. Popisuje základní části programu Automation Studio a použití simulátoru v projektech. Dále tato kapitola demonstuje základy konfigurace reálného PLC.

Ve čtvrté kapitole jsou ukázány 4 nejčastěji používané programovací jazyky a jejich aplikace na jednoduchý příklad osvětlení.

Poslední kapitola se zabývá praktickou úlohou na laboratorním přípravku, vypracováním 3 vzorových úloh v programu Automation Studio a jejich podrobným popisem. Model výrobní linky byl vytvořen v programu SketchUp.

Bakalářská práce obsahuje 2 přílohy. První z nich je podrobný návod práce v Automation Studiu. Ve druhé příloze jsou zpracovány 3 vzorové úlohy v programu Automation Studio a jejich řešení.

Byl vytvořen kvalitní návod pro práci v programu Automation Studio, který lze použít jako výukový materiál. Dle mého názoru jsou produkty firmy B&R na velmi dobré úrovni. Největší výhodou je programové prostředí Automation Studio, které je velice přehledné a intuitivní.

SEZNAM POUŽITÉ LITERATURY

- [1] B&R AUTOMATION. [online]. 2012 [cit. 2014-05-23]. Dostupné z: <http://www.br-automation.com/cs/perfection-in-automation/>
- [2] URBAN, Luboš. Programování PLC podle normy IEC EN 61131-3 – víc než jednotné jazyky. *Automa: časopis pro automatizační techniku* [online]. roč. 2005, č. 02 [cit. 2014-05-23]. Dostupné z: http://www.odbornecasopisy.cz/index.php?id_document=30310
- [3] ŠMEJKAL, Ladislav. Esperanto programátorů PLC: programování podle normy IEC/EN 61131-3 (část 4). *Automa: časopis pro automatizační techniku* [online]. roč. 2011, č. 12 [cit. 2014-05-23]. Dostupné z: http://www.odbornecasopisy.cz/index.php?id_document=45413
- [4] VOJÁČEK, Antonín. Programovací režimy pro PLC dle IEC 61131-3 (CoDeSys). *Automatizace.hw.cz* [online]. 2011 [cit. 2014-05-23]. Dostupné z: <http://automatizace.hw.cz/programovaci-rezimy-pro-plc-dle-iec-61131-3-codesys>
- [5] Programovací jazyky PLC. *Elektroprůmysl.cz: informace ze světa průmyslu a elektrotechniky* [online]. 2013 [cit. 2014-05-23]. Dostupné z: <http://www.elektroprumysl.cz/software/programovaci-jazyky-plc>
- [6] THAYER, Ted. Babylon – vyznejte se v programovacích jazycích pro IEC 61131-3. *Control engineering* [online]. 2009 [cit. 2014-05-23]. Dostupné z: <http://www.controlengcesko.com/hlavni-menu/artikuly/artikul/article/babylon-vyznejte-se-v-programovacich-jazycich-pro-iec-61131-3/>
- [7] ŠMEJKAL, Ladislav. Esperanto programátorů PLC: programování podle normy IEC/EN 61131-3 (část 4). *Automa: časopis pro automatizační techniku* [online]. roč. 2011, č. 12 [cit. 2014-05-23]. Dostupné z: http://www.odbornecasopisy.cz/index.php?id_document=45413
- [8] PLC - programovací jazyky. *Zandruv svět* [online]. 2005 [cit. 2014-05-23]. Dostupné z: <http://www.zandruvsvet.opatnet.cz/view.php?cisloclanku=2005051001>
- [9] NĚMEC, Jan. C/C++ (1) - Úvod. *Linuxsoft.cz* [online]. 2004 [cit. 2014-05-23]. Dostupné z: http://www.linuxsoft.cz/article.php?id_article=370
- [10] CHARVÁT, Robert. Svět průmyslových počítačů B&R. *Automa: časopis pro automatizační techniku* [online]. roč. 2002, č. 12 [cit. 2014-05-23]. Dostupné z: http://www.odbornecasopisy.cz/index.php?id_document=28639

- [11] BÍLEK, Karel. Ethernet, internetové technologie a systémy B&R. *Automa: časopis pro automatizační techniku* [online]. roč. 2002, č. 02 [cit. 2014-05-23]. Dostupné z: <http://www.odbornecasopisy.cz/ethernet-internetove-technologie-a-systemy-b-r-28429.html>
- [12] Ethernet Powerlink – komunikace v reálném čase. *Automa: časopis pro automatizační techniku* [online]. roč. 2001, č. 03 [cit. 2014-05-23]. Dostupné z: <http://www.odbornecasopisy.cz/ethernet-internetove-technologie-a-systemy-b-r-28429.html>
- [13] Průmyslový Ethernet. *Elektroprůmysl.cz* [online]. 2014 [cit. 2014-05-23]. Dostupné z: <http://www.elektroprumysl.cz/automatizace-rizeni-a-regulace/prumyslovy-ethernet>
- [14] Komunikace není jen otázkou spojení. *Automa: časopis pro automatizační techniku* [online]. roč. 2001, č. 11 [cit. 2014-05-23]. Dostupné z: [http://www.odbornecasopisy.cz/komunikace-neni-jen-otazkou-spojeni-33719.html](http://www.odbornecasopisy.cz/komunikace-neni-jen-otazkou-spojzeni-33719.html)
- [15] BÍLEK, Karel. Produkty B&R – automatizace, která má vtip. *Automa: časopis pro automatizační techniku* [online]. roč. 2005, č. 03 [cit. 2014-05-23]. Dostupné z: http://www.odbornecasopisy.cz/index.php?id_document=30375
- [16] BÍLEK, Karel. Systém X20 – v inovacích je konjunktura stále. *Automa: časopis pro automatizační techniku* [online]. roč. 2005, č. 02 [cit. 2014-05-23]. Dostupné z: http://www.odbornecasopisy.cz/index.php?id_document=30320
- [17] Kompaktní programovatelné automaty X20 s integrovanými I/O. *Automatizace v potravinářství* [online]. 2013 [cit. 2014-05-23]. Dostupné z: <http://www.prumyslovaautomatizace.com/kompaktni-programovatelne-automaty-x20-s-integrovanymi-io>
- [18] BARTOŠÍK, Petr. X20: hybnou silou inovací jsou potřeby zákazníků. *Automa: časopis pro automatizační techniku* [online]. roč. 2006, č. 02 [cit. 2014-05-23]. Dostupné z: <http://www.odbornecasopisy.cz/x20:-hybnou-silou-inovaci-jsou-potreby-zakazniku-30938.html>
- [19] BILÍK, Karel. OpenSafety – stejné bezpečí pro všechny. *Automa: časopis pro automatizační techniku* [online]. roč. 2011, č. 02 [cit. 2014-05-23]. Dostupné z: <http://www.odbornecasopisy.cz/opensafety-%E2%80%93-stejne-bezpeci-pro-vsechny-42959.html>
- [20] Společnost B&R bilancuje. *Automa* [online]. roč. 2009, č. 12 [cit. 2014-05-23]. Dostupné z: http://www.odbornecasopisy.cz/index.php?id_document=40133
- [21] B&R APROL pro řízení procesů s povinnou validací. *Automa: časopis pro automatizační techniku* [online]. roč. 2005, č. 06 [cit. 2014-05-23]. Dostupné z: http://www.odbornecasopisy.cz/index.php?id_document=30526

SEZNAM PŘÍLOH

Návod práce Automation Studio

Sbírka úloh Automation Studio

CD-R obsahující: Tento dokument ve formátu PDF

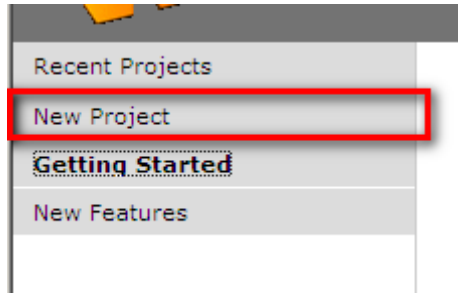
Vypracované programy v Automation Studio

Návod
AUTOMATION STUDIO

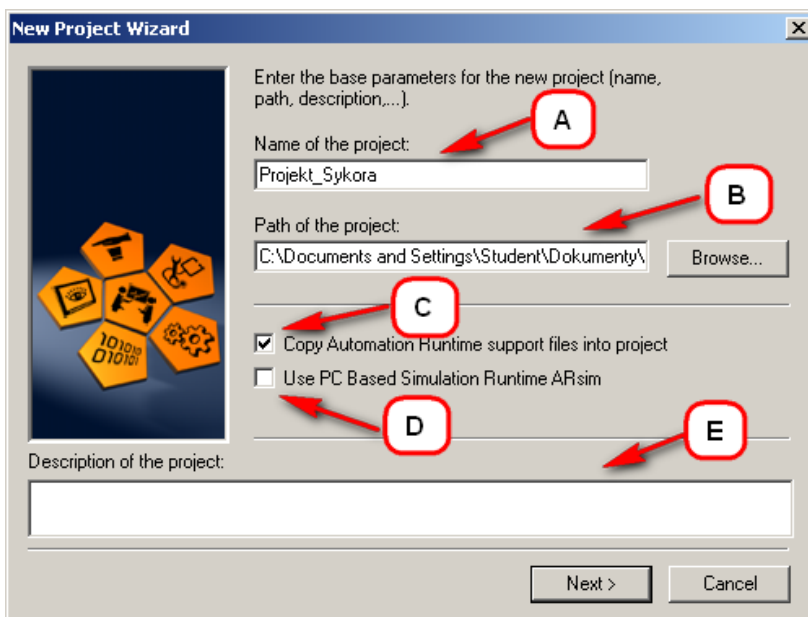
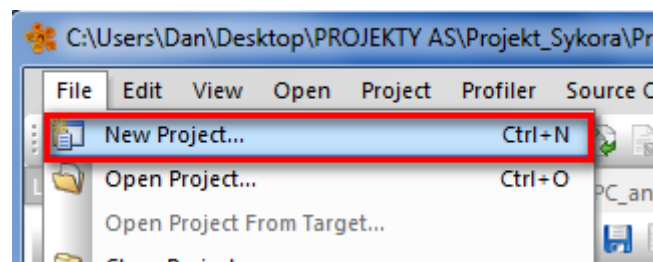
1 Založení projektu

1.1 Založení projektu v Automation Studio 3

Pro spuštění klikněte na < **Start** > / < **Automation Studio** >, nebo ikonu na ploše. Při prvním spuštění Automation Studio se zobrazí Start Page. Zde klikněte v levé části na **New Project**.



Nový projekt lze také vytvořit přes horní lištu < **File** > / < **New project** >

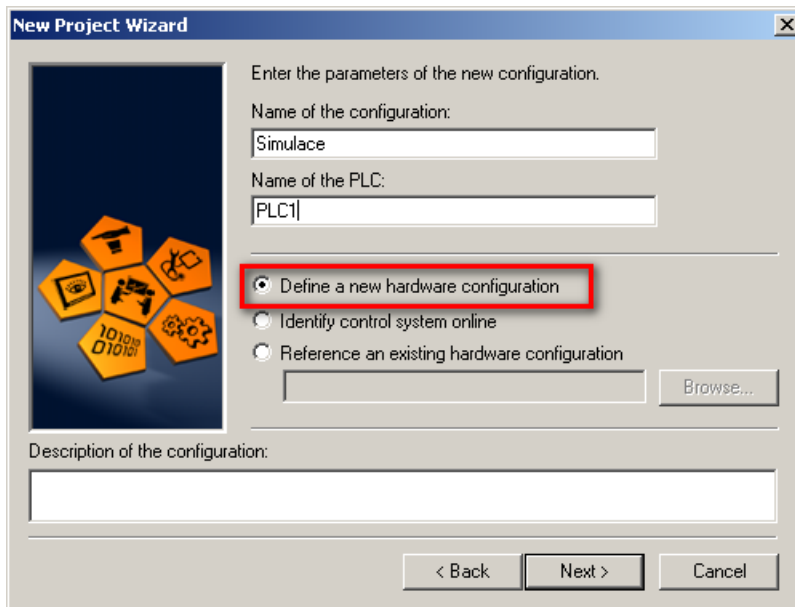
A screenshot of the 'New Project Wizard' dialog box. The dialog has a title bar and a close button. On the left, there is a graphic with several icons. The main area contains the following fields and options:

- 'Name of the project:' with a text box containing 'Projekt_Sykora'. A red arrow labeled 'A' points to this text box.
- 'Path of the project:' with a text box containing 'C:\Documents and Settings\Student\Dokumenty\' and a 'Browse...' button. A red arrow labeled 'B' points to this text box.
- Two checkboxes: 'Copy Automation Runtime support files into project' (checked) and 'Use PC Based Simulation Runtime ARsim' (unchecked). A red arrow labeled 'C' points to the first checkbox.
- 'Description of the project:' with a large empty text box. A red arrow labeled 'D' points to the top of this text box, and another red arrow labeled 'E' points to the bottom of this text box.

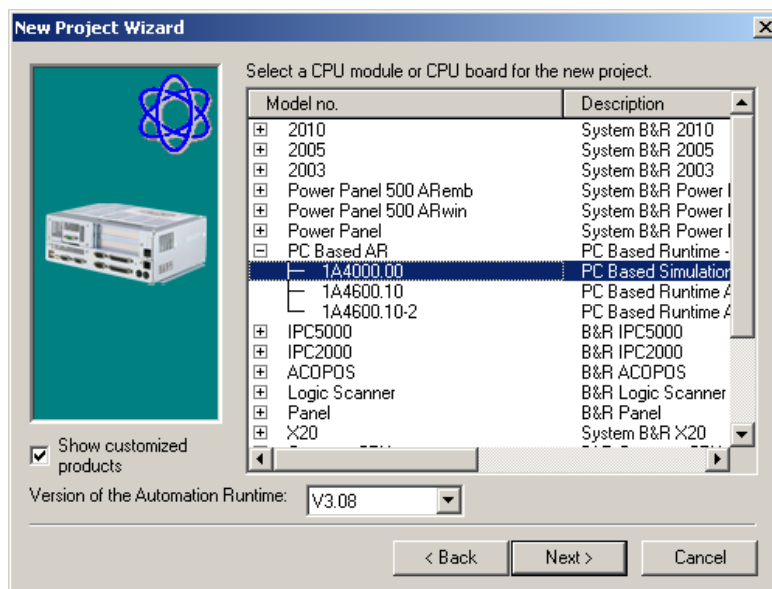
At the bottom, there are 'Next >' and 'Cancel' buttons.

- A - Název projektu
- B - Uložení projektu
- C – Kopírování Automation Runtime – Zde zatrhněte, že chcete kopírovat Automation Runtime
- D – Použití simulátoru v projektu – Zde nezatrhávejte, simulátor se nastaví až v dalším kroku
- E – Popis projektu

Jestliže máte vše vyplněné pro pokračování zmačkněte **NEXT** >



Pro definování simulátoru zatrhněte **Define a new hardware configuration** a **NEXT >**

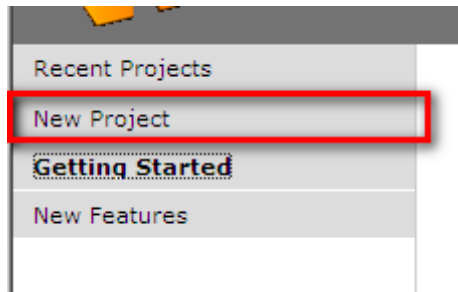


Z nabídky vyberte model 1A4000.00 a potvrďte tlačítkem **NEXT >**. V dalším okně zkontrolujte všechny nastavené parametry a pro dokončení simulátoru zmáčkněte **Finish**

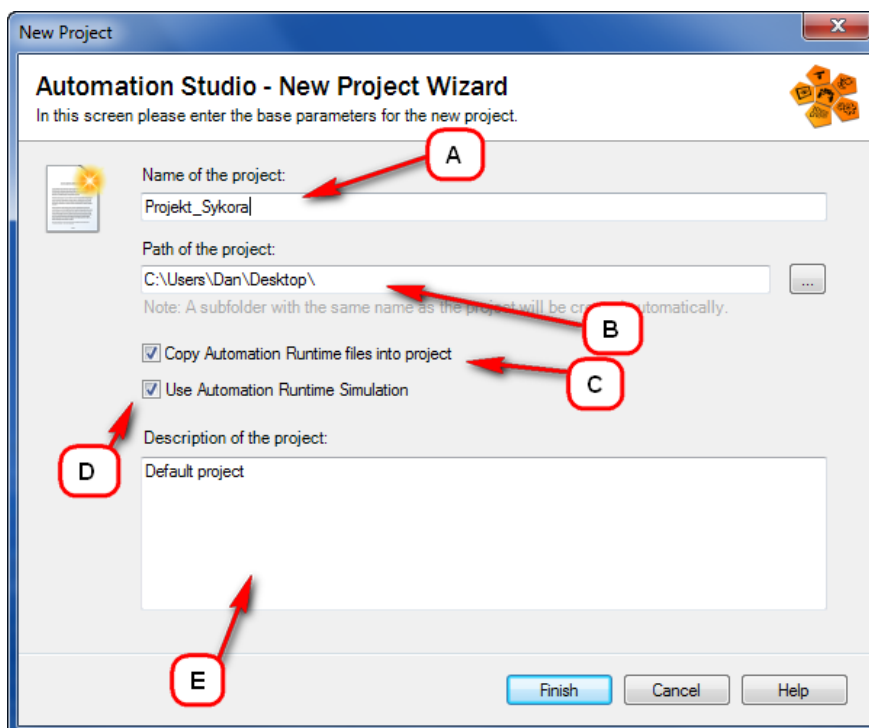
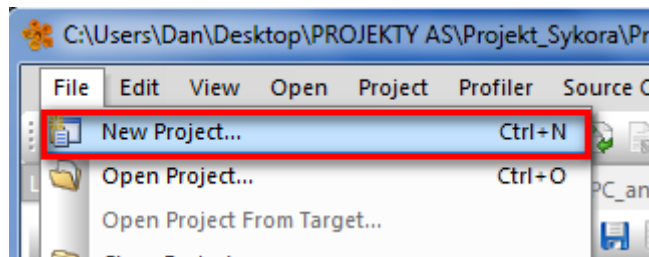
..

1.2 Založení projektu v Automation Studio 4

Pro spuštění klikněte na < **Start** > / < **Automation Studio** >, nebo ikonu na ploše. Při prvním spuštění Automation Studio se zobrazí Start Page. Zde klikněte v levé části na **New Project**.



Nový projekt lze také vytvořit přes horní lištu < **File** > / < **New Project** >



Automation Studio - New Project Wizard
In this screen please enter the base parameters for the new project.

Name of the project: (A)

Path of the project: (B)

Copy Automation Runtime files into project (C)

Use Automation Runtime Simulation (D)

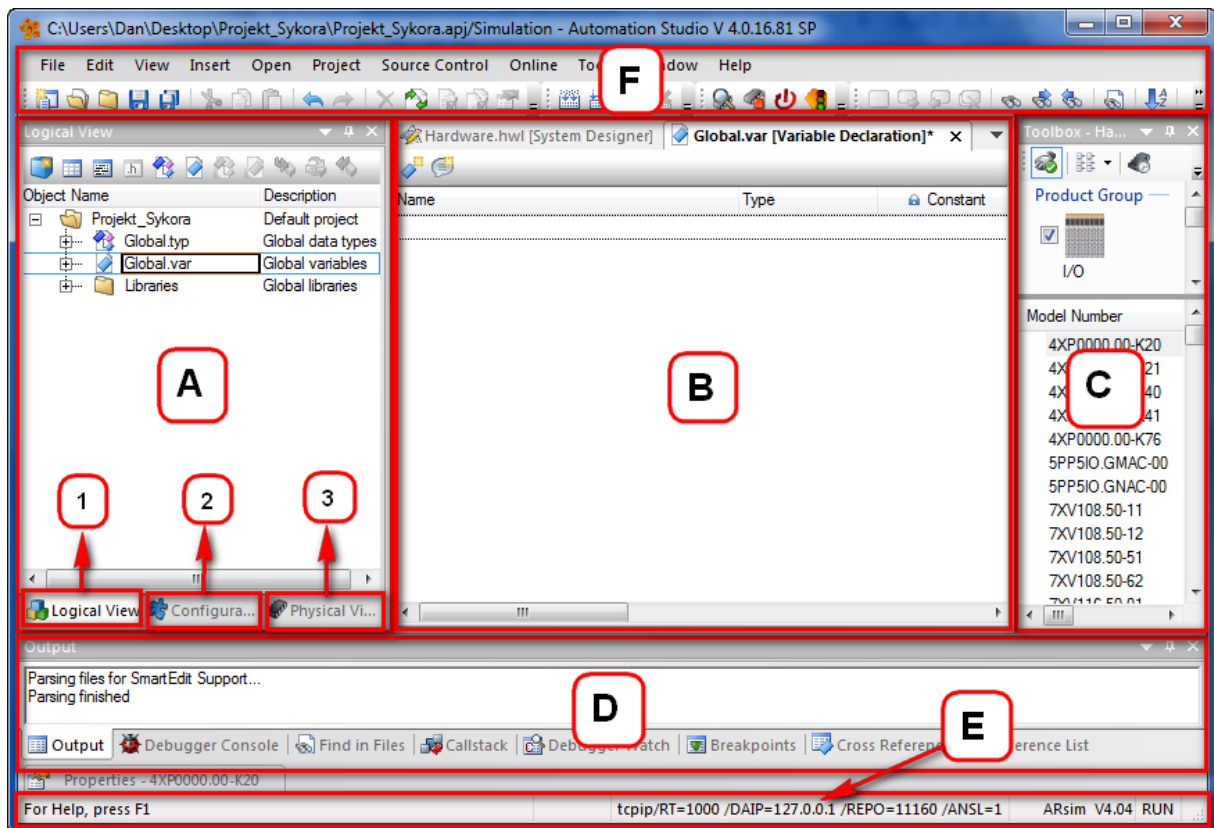
Description of the project:

(E)

- A - Název projektu
- B - Uložení projektu
- C – Kopírování Automation Runtime v projektu
- D – Použití simulátoru v projektu
- E – Popis projektu

Pro dokončení zmáčkněte **Finish**

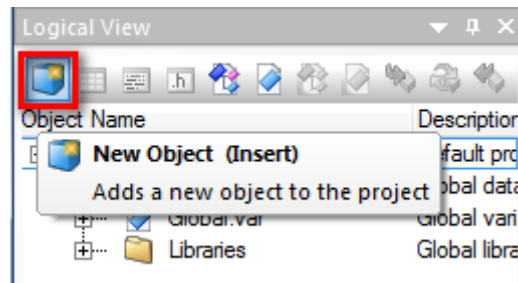
2 Popis pracovního prostředí



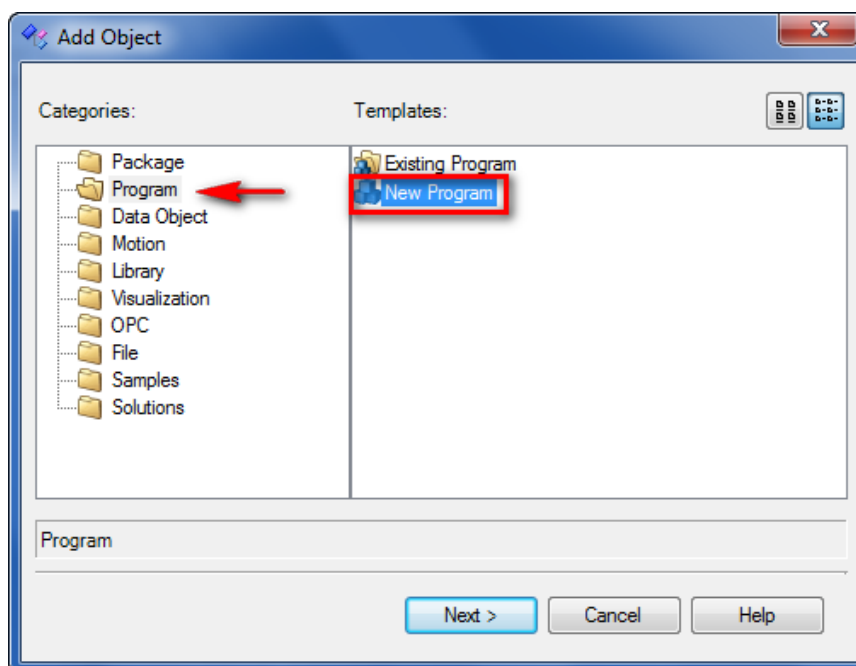
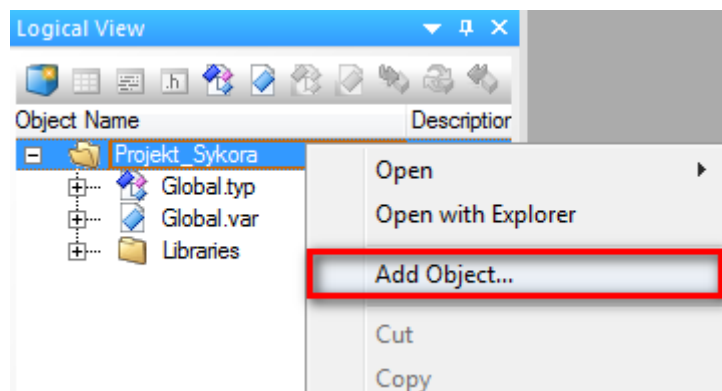
- A. V levé části pracovního prostředí se nachází organizátor projektu, který obsahuje tři základní záložky, kterými se vytváří nebo editují komponenty projektu. (*Logical View*, *Configuration View* a *Physical View*)
 - B. V prostřední části se nachází hlavní část, kde lze programovat a konfigurovat data projektu
 - C. V pravé části se nachází Toolbox k přidávání hardwarových modulů atd.
 - D. Dolní část pracovního prostředí obsahuje výstupní okno, které zobrazuje informace při kompilaci projektu
 - E. Lišta, která dává stav projektu, data o spojení atd.
 - F. Menu a pracovní lišta poskytují přístup k rozsáhlým funkcím programu Automation Studia
-
- 1) *Logical View* - je záložka organizátoru projektu, ve kterém se nachází všechny programy
 - 2) *Configuration View* - je záložka organizátoru projektu, ve které se provádí veškeré konfigurace projektu
 - 3) *Physical View* - je záložka organizátoru projektu, ve které se přidávají, upravují a nastavují všechny hardwarové části projektu

3 Vytvoření programu

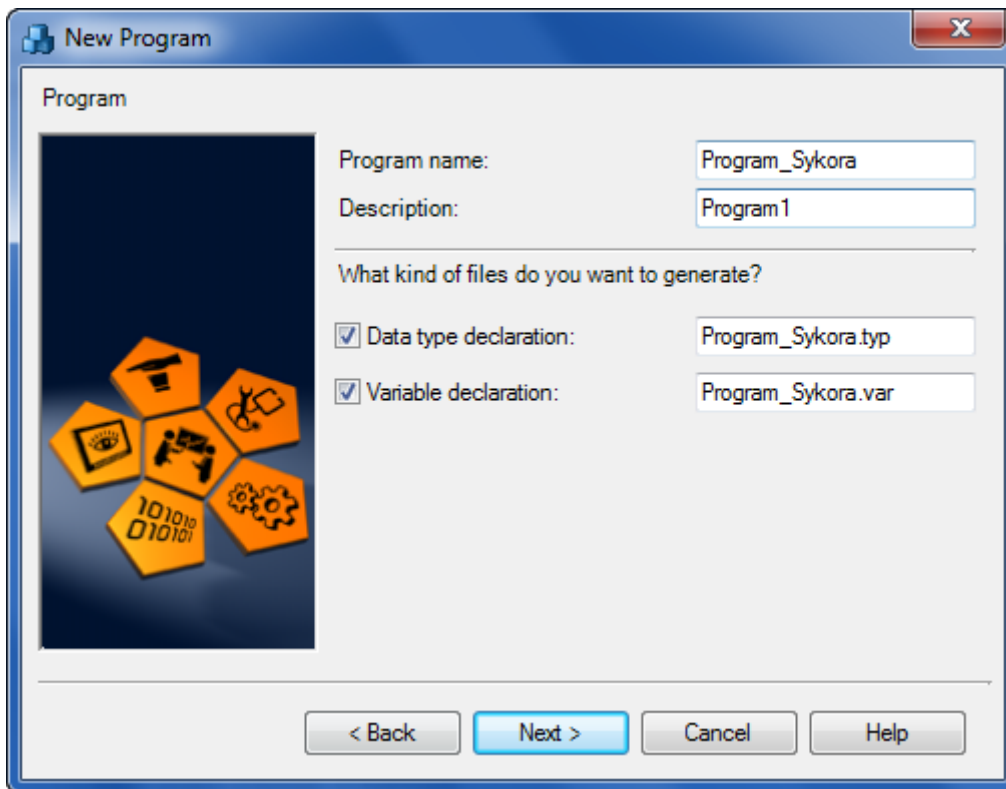
V záložce *Logical View*, která se nachází na levé straně pracovního prostředí, zvolíte pro přidání nového programu ikonu **New Object**



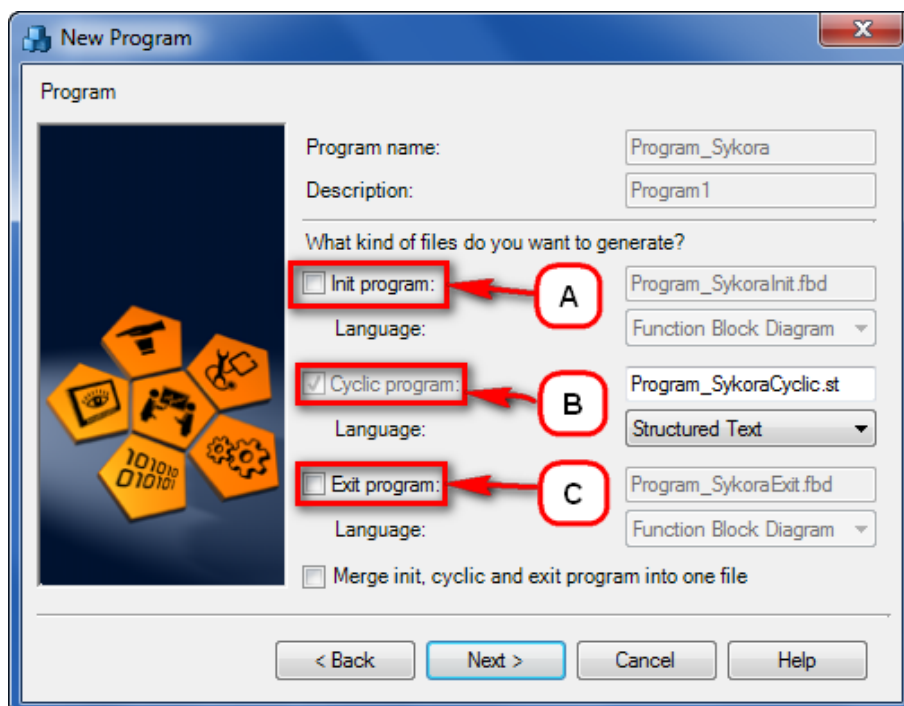
Nový objekt lze také vytvořit kliknutím pravého tlačítka myši na název projektu a zvolením **Add object**



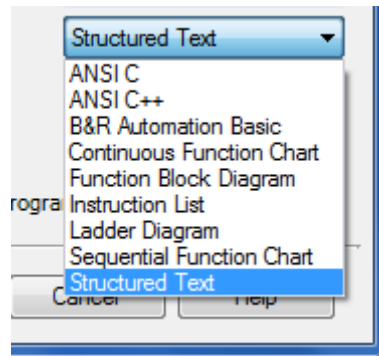
V levé části vyberete složku **Program** a v pravé části **New Program** a potvrdíte **NEXT >**



Zde vyplníte název programu a popis programu a vše necháte označené, pro pokračování zmáčkněte **NEXT >**



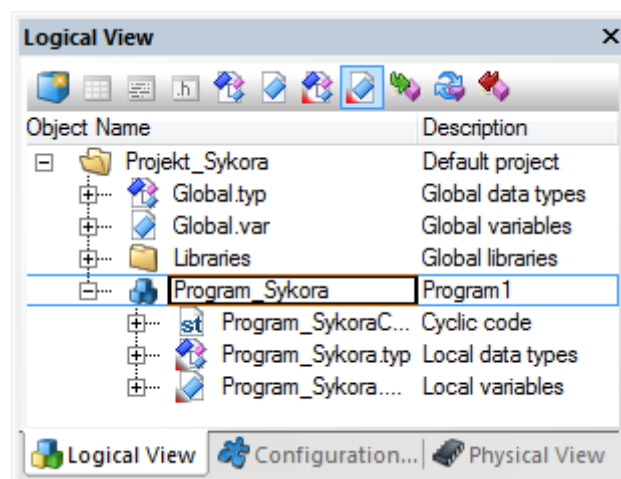
- A. **Init program** – Při označení se vytvoří nový Init program, který se vykoná vždy jen jednou na začátku programu
- B. **Cyclic program** – Se vytvoří cyklická část programu
- C. **Exit program** – Při označení se vytvoří nový Exit program, který se vykoná vždy jen jednou na konci programu



Zde si zvolíte programovací jazyk, ve kterém budete psát svůj program. Poté dáte **NEXT>**.

V dalším okně zkontrolujete všechny údaje a program vytvoříte tlačítkem **Finish**

Vytvořený program se Vám zobrazí v záložce *Logical View*:



4 Práce se simulátorem

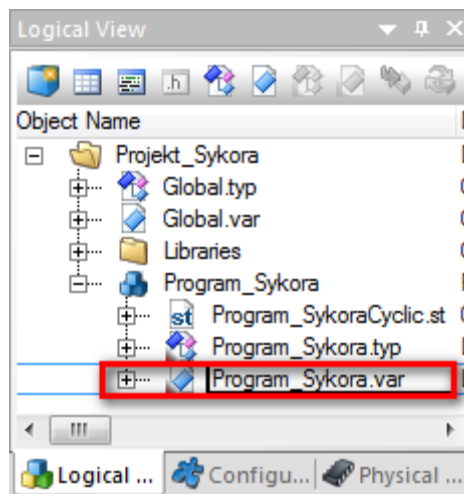
Pro demonstraci práce se simulátorem vytvořte program. Popis programu je následující:

- Po zapnutí vypínačů *vstup1* a *vstup2* se po 5 sekundách rozsvítí světlo *vystup* do logické 1, při vypnutí jednoho z vypínačů se světlo nastaví do logické 0.

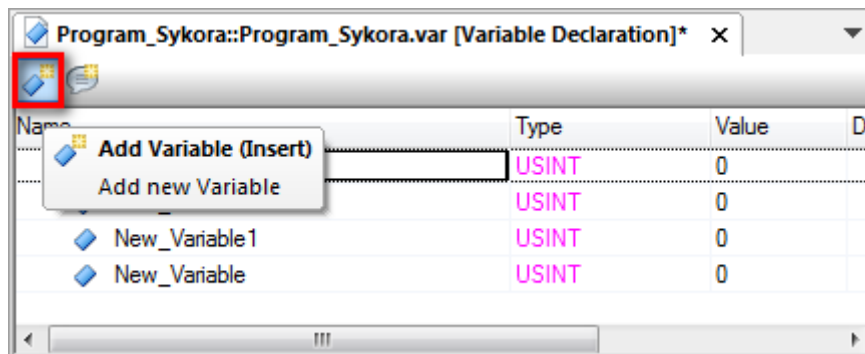
4.1 Deklarace a psaní programu

V záložce *Logical View* si nadeklarujte proměnné:

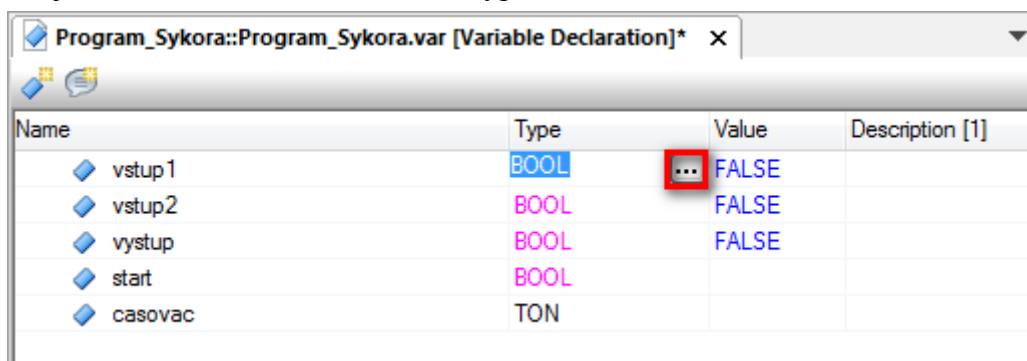
1. Zvolte si deklarační okno.var



2. Vložení proměnných



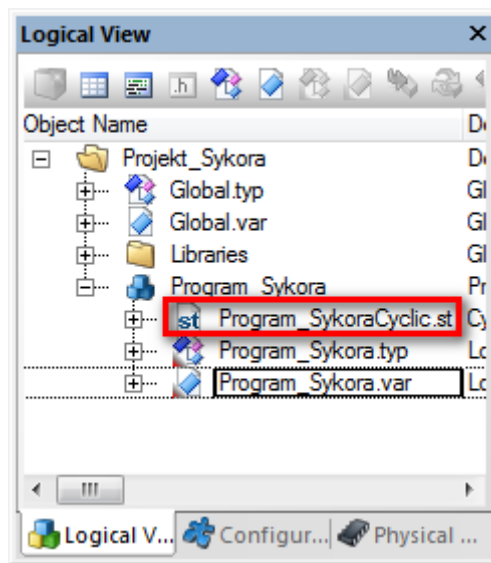
3. Přejmenování a nastavení datového typu



Název proměnné přejmenujte dvojklikem, datový typ změňte kliknutím na označenou ikonu

V záložce *Logical View* si napište cyklickou část programu:

1. Zvolte cyklickou část

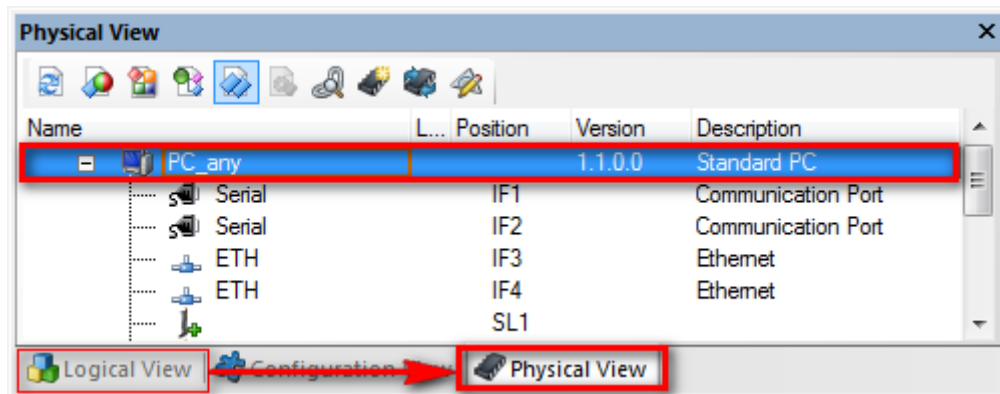


2. Napište samotný program

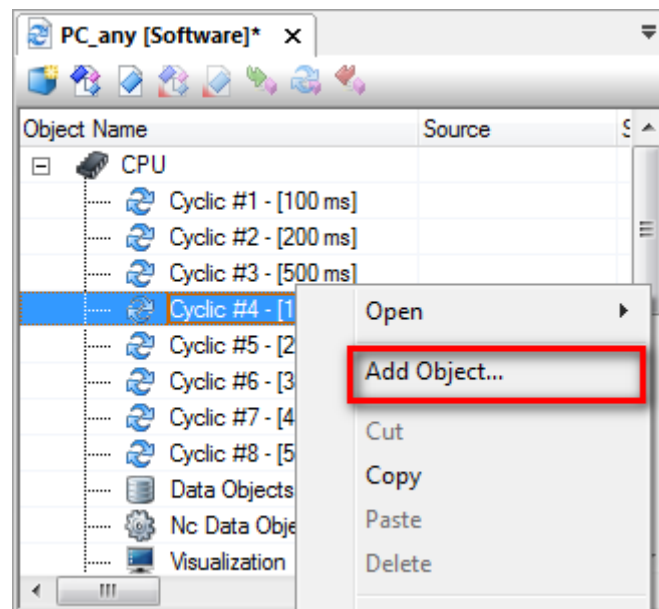
```
st Program_Sykora:Program_SykoraCyclic.st [Structured Text]* x Program_Sykora:Program_Sykora.var [Variable Declaration]*
//
*_CYCLIC
PROGRAM _CYCLIC
    IF vstup1 AND vstup2 = 1 THEN
        start:=1; // Když jsou oba vypínače zapnuty, tak se spustí časovač
        IF casovac.Q=1 THEN
            vystup:=1; // Po 5s se rozsvítí světlo
        END_IF
    ELSE vystup:=0; // Když jeden z vypínačů je vypnutý, světlo nesvítí
        start:=0;
    END_IF
    casovac(IN :=start, PT:=T#5s ); // Nastavení časovače na 5s
END_PROGRAM
```

4.2 Přiřazení programu do cyklické třídy

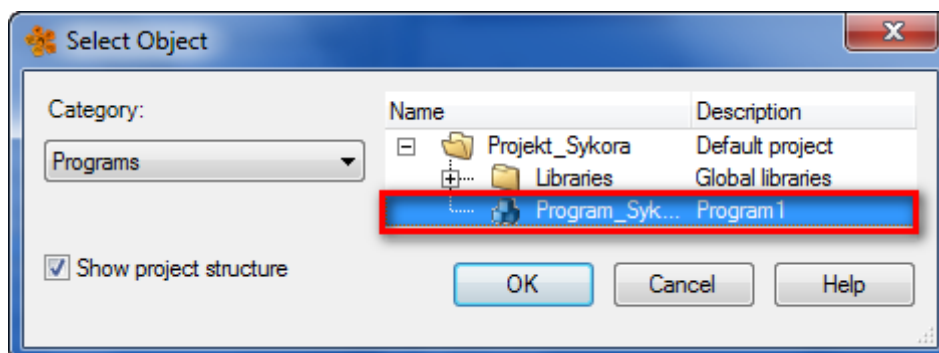
V levé části prostředí změňte *Logical View* na *Physical View* a dvojklikem označte **PC_any**



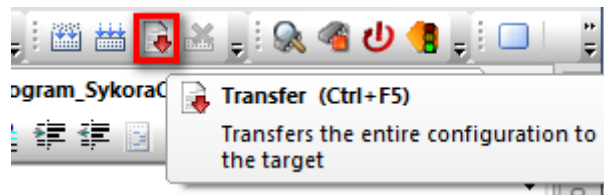
Otevře se Vám softwarová část, kde se přiřazují programy do cyklických tříd



Vyberte program, který chcete přiřadit do dané cyklické třídy a potvrďte tlačítkem **OK**. Lze také provést přetáhnutím myši z *Logical View* do dané cyklické třídy



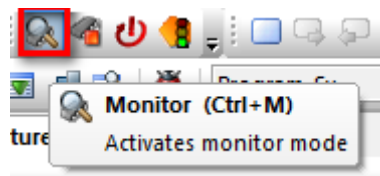
Pro nahrání programu do simulátoru zvolte tlačítko **Transfer** v hlavní liště




4.3 Ověření funkčnosti programu

4.3.1 Watch

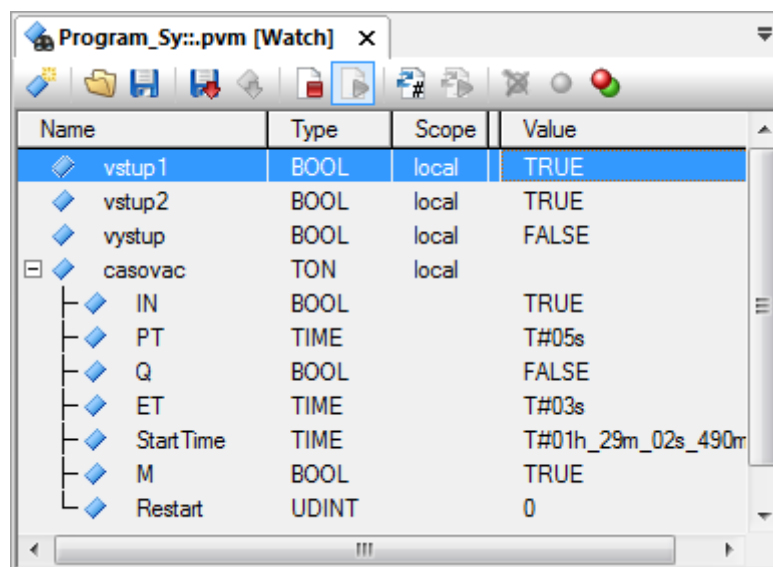
Pro sledování změn proměnných zapněte v hlavní liště online mód tlačítkem **Monitor**, nebo kombinací tlačítek (**Ctrl+M**)



Poté se automaticky otevře Watch, lze jej otevřít i pomocí hlavní nabídky **< Open > / < Watch >** nebo kombinací kláves Ctrl+W.

Pro přidání proměnných klikněte pravým tlačítkem do prostoru Watch a zvolte **Insert Variable**, nebo ikonou  Zvolte proměnné, které chcete sledovat

Hodnoty měníte přepsáním ve sloupci *Value*

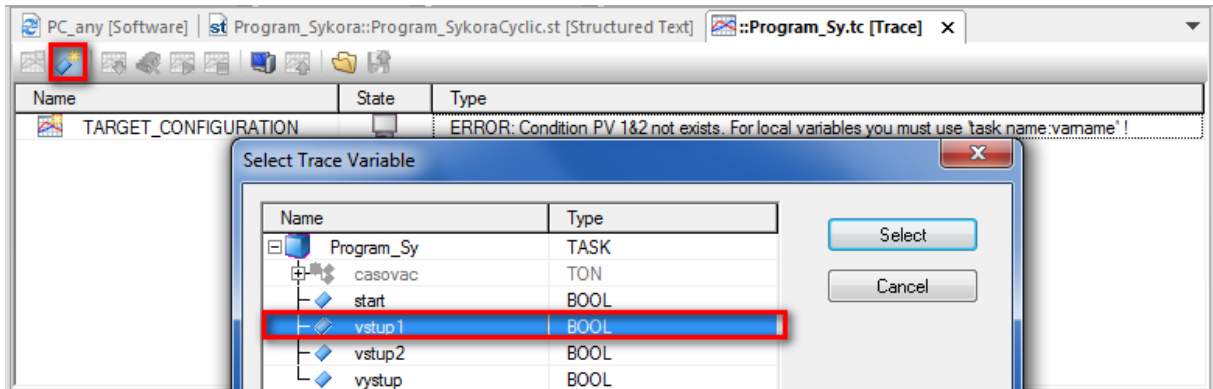


| Name | Type | Scope | Value |
|-----------|-------|-------|--------------------|
| vstup1 | BOOL | local | TRUE |
| vstup2 | BOOL | local | TRUE |
| vystup | BOOL | local | FALSE |
| casovac | TON | local | |
| IN | BOOL | | TRUE |
| PT | TIME | | T#05s |
| Q | BOOL | | FALSE |
| ET | TIME | | T#03s |
| StartTime | TIME | | T#01h_29m_02s_490m |
| M | BOOL | | TRUE |
| Restart | UDINT | | 0 |

4.3.2 Trace

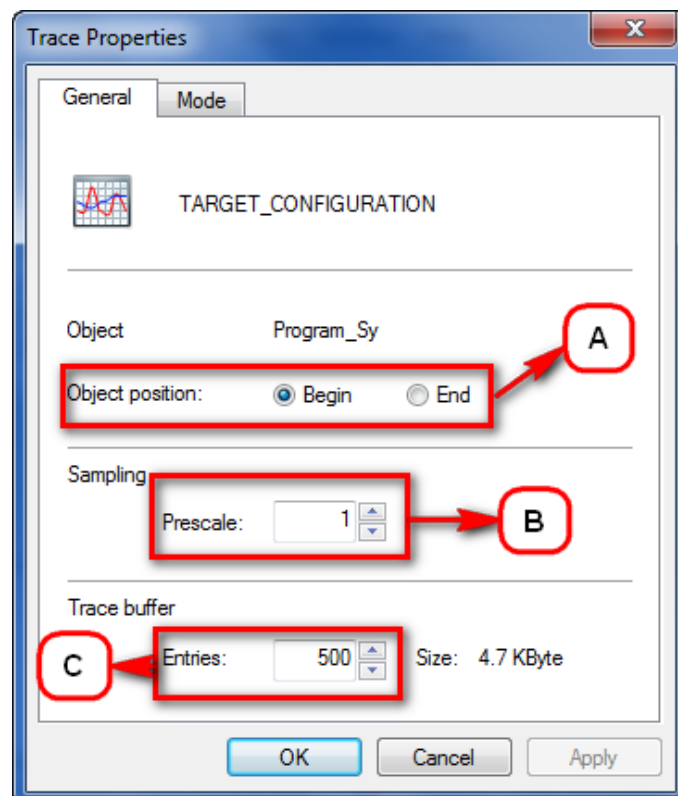
Označte program (< **Physical View** > / < **PC_any** >), který chcete vyšetřovat a poté otevřete Trace v hlavní liště <**Open**> / < **Trace** >.

Po spuštění Trace přiřadíte proměnné ikonou **Insert a New Variable**, vyberete proměnnou a zvolte **Select**



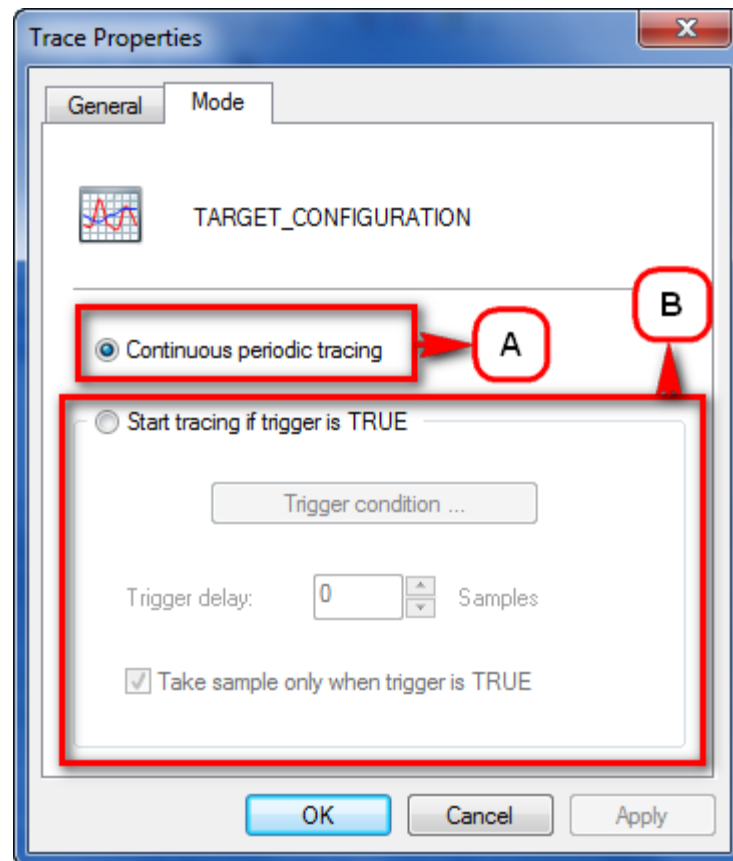
Po přiřazení všech sledovaných proměnných klikněte pravým tlačítkem na **TARGET_CONFIGURATION** a zvolte **Properties**:

1. Propertie/General



- A. Volíte, kde chcete skenovat (na začátku, nebo na konci)
- B. Počet naměřených hodnot, které se hromadí v datové vyrovnávací paměti před začátkem nahrávání stopy
- C. Počet položek, které se musí vejít do datové vyrovnávací paměti

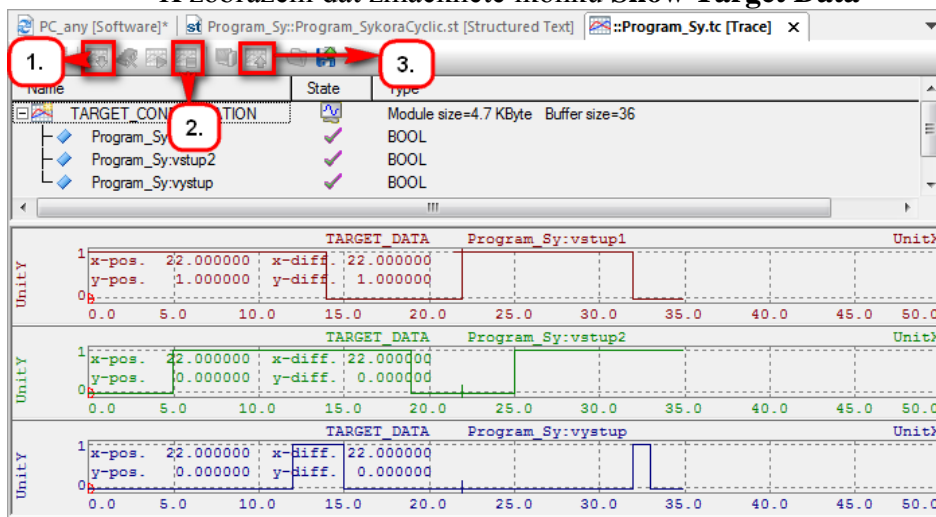
2. Properties/Mode



- A. Trace proces začíná ihned po instalaci (viz dále). Pokud je Trace proces zastaven a restartován, stopa začíná tam, kde byla přerušena.
- B. Lze nastavit Trigger podmínku, která když platí, tak je Trace proces aktivní. Pomocí Trigger delay nastavíte zpoždění procesu.

Poté zmáčkněte ikonu **Install**, ve Watch změňte hodnoty proměnných a až budete chtít ukončit Trace proces zmáčkněte **Stop**.

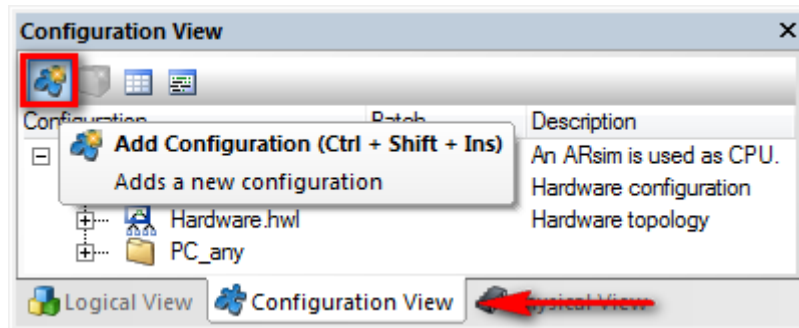
K zobrazení dat zmáčkněte ikonku **Show Target Data**



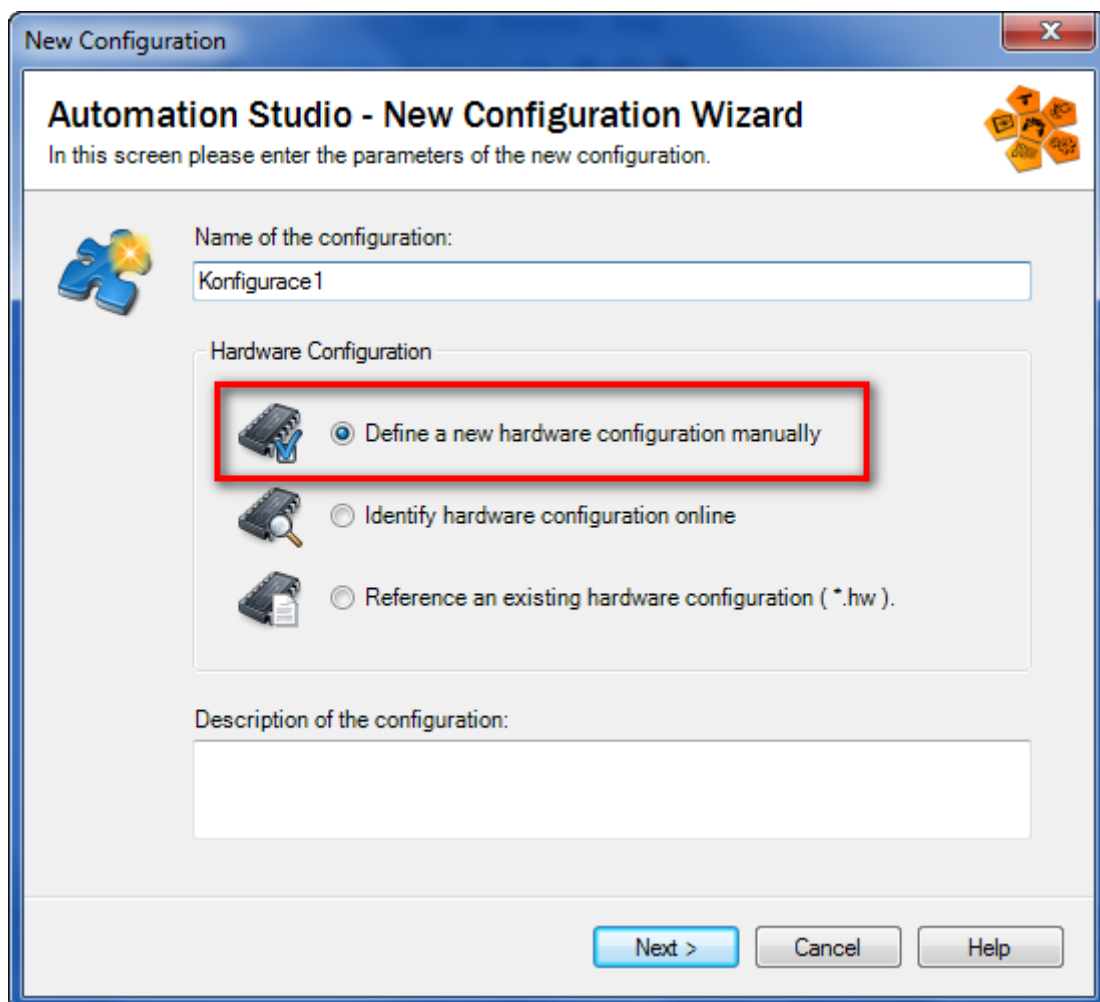
5 Konfigurace reálného PLC

5.1 Nová konfigurace

Nové konfigurace se vytvářejí v záložce *Configuration View* v levé části.



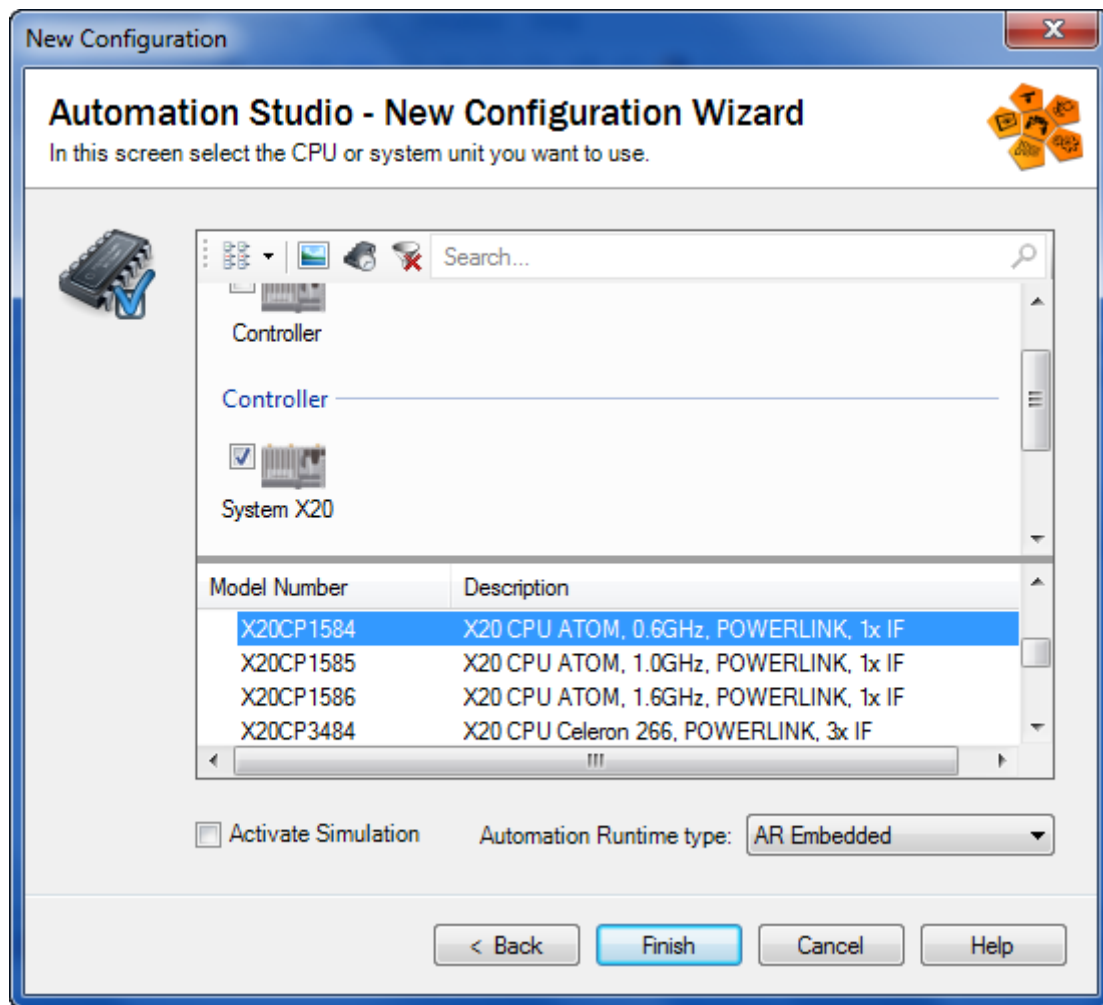
Pro přidání nové konfigurace klikněte na ikonu **Add Configuration**



Vyplňte jméno konfigurace. Definovat hardware lze třemi způsoby:

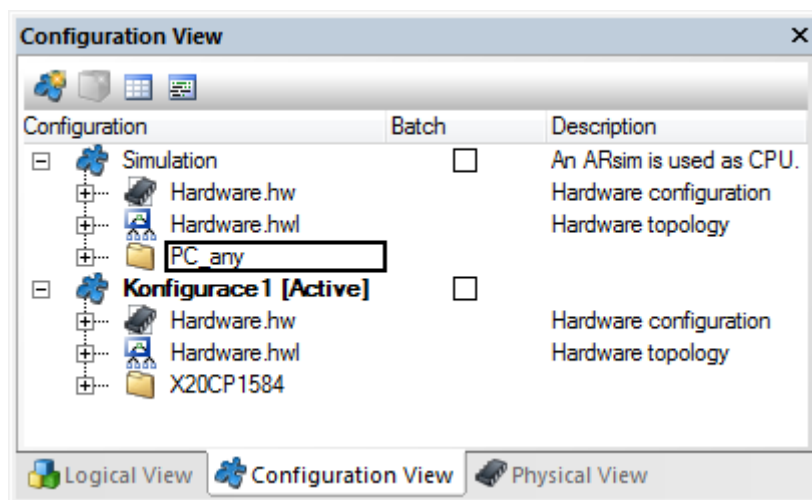
- Definovat ručně
- Definovat online
- Použít již vytvořenou konfiguraci

Pro přechod do dalšího okna zmáčkněte **Next>**. V dalším okně vyberte vaše PLC



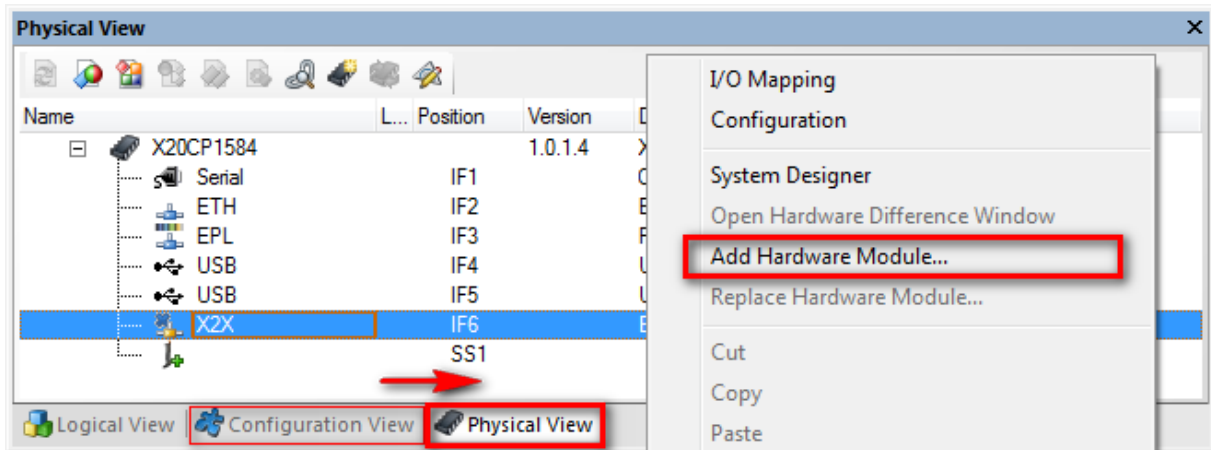
Ikonou **Finish** potvrdíte výběr PLC a konec vytváření nové konfigurace

Nová konfigurace se zobrazí v záložce *Configuration View* a stane se aktivní [**Active**], aktivní konfigurace lze libovolně mezi sebou přepínat dvojklikem na název konfigurace

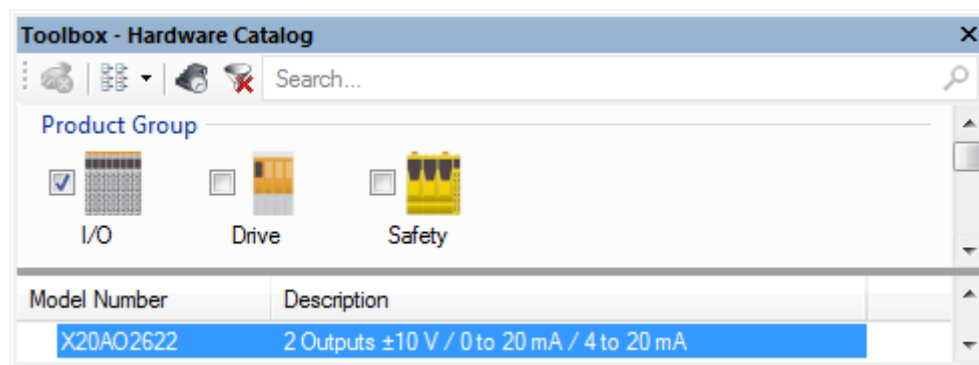


5.2 Přidávání modulů

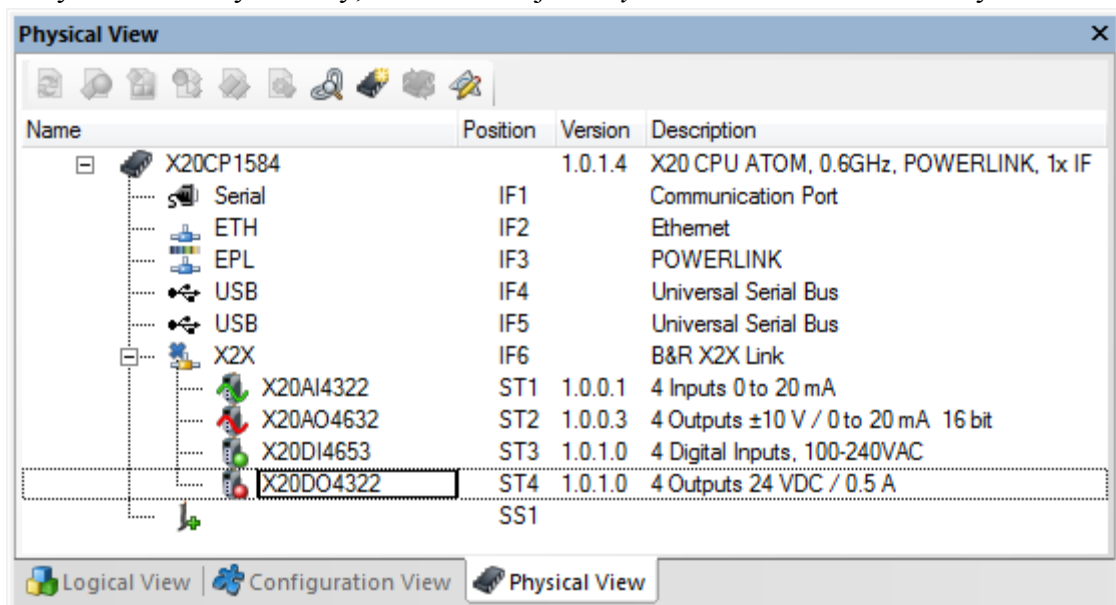
Přidavné moduly (moduly vstupů, výstupů atd.) se přidávají v záložce *Physical View*, kde pravým tlačítkem kliknete na připojovací značku X2X a vyberte **Add Hardware Module**



Poté z tabulky postupně vyberte moduly, tak jak jsou liniově zapojeny:

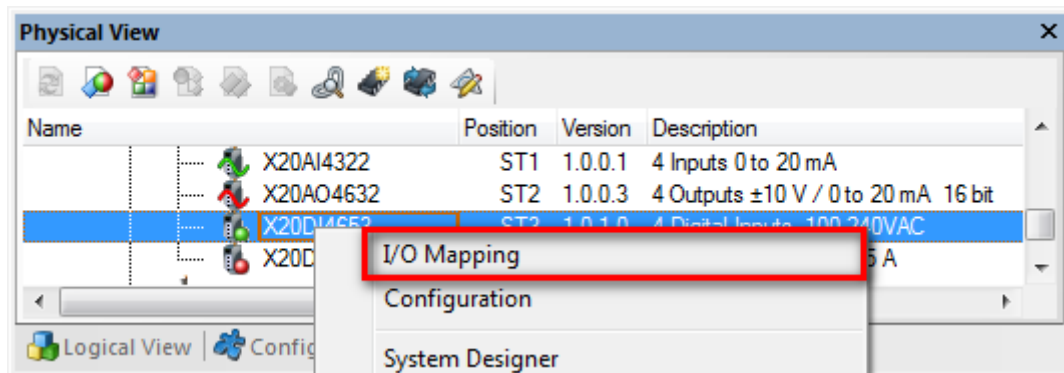


Takto vyberte všechny moduly, které nám dají v *Physical View* tzn. Hardwarový strom

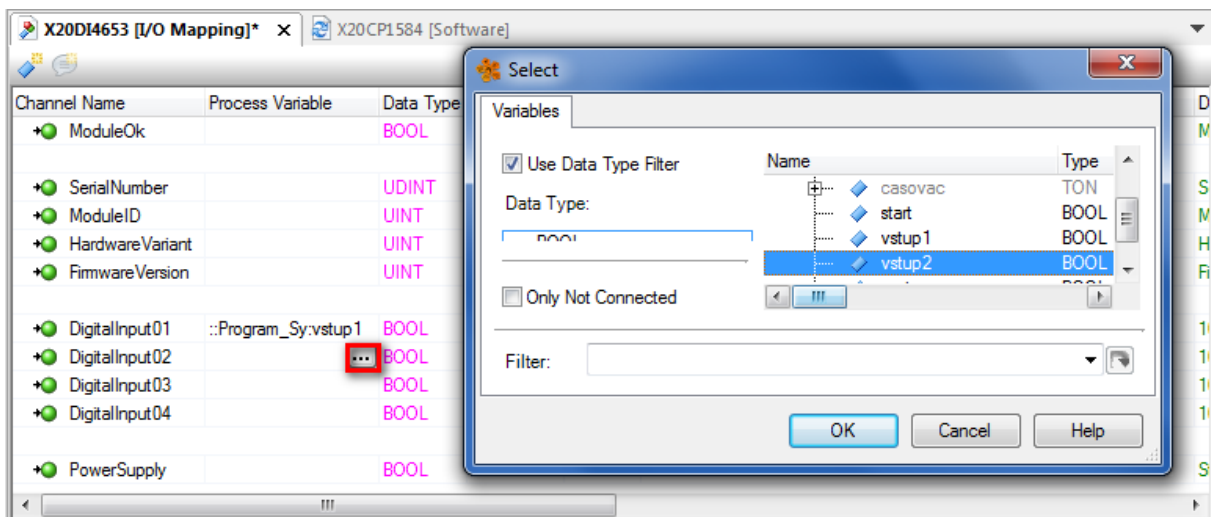


5.3 Mapování proměnných na vstupy a výstupy

V záložce *Physical View* klikněte pravým tlačítkem na modul a zvolte **I/O Mapping**



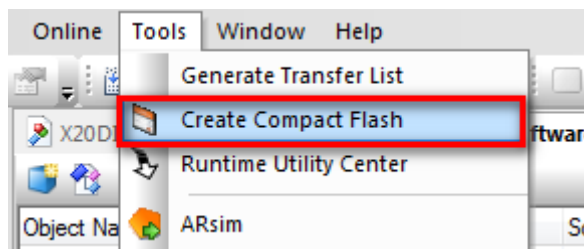
Poté se otevře I/O Mapping, kde v odstavci *Process Variable* přiřadíte proměnnou kliknutím na ikonu s třemi tečkami, vyberete proměnnou a potvrdíte **OK**.



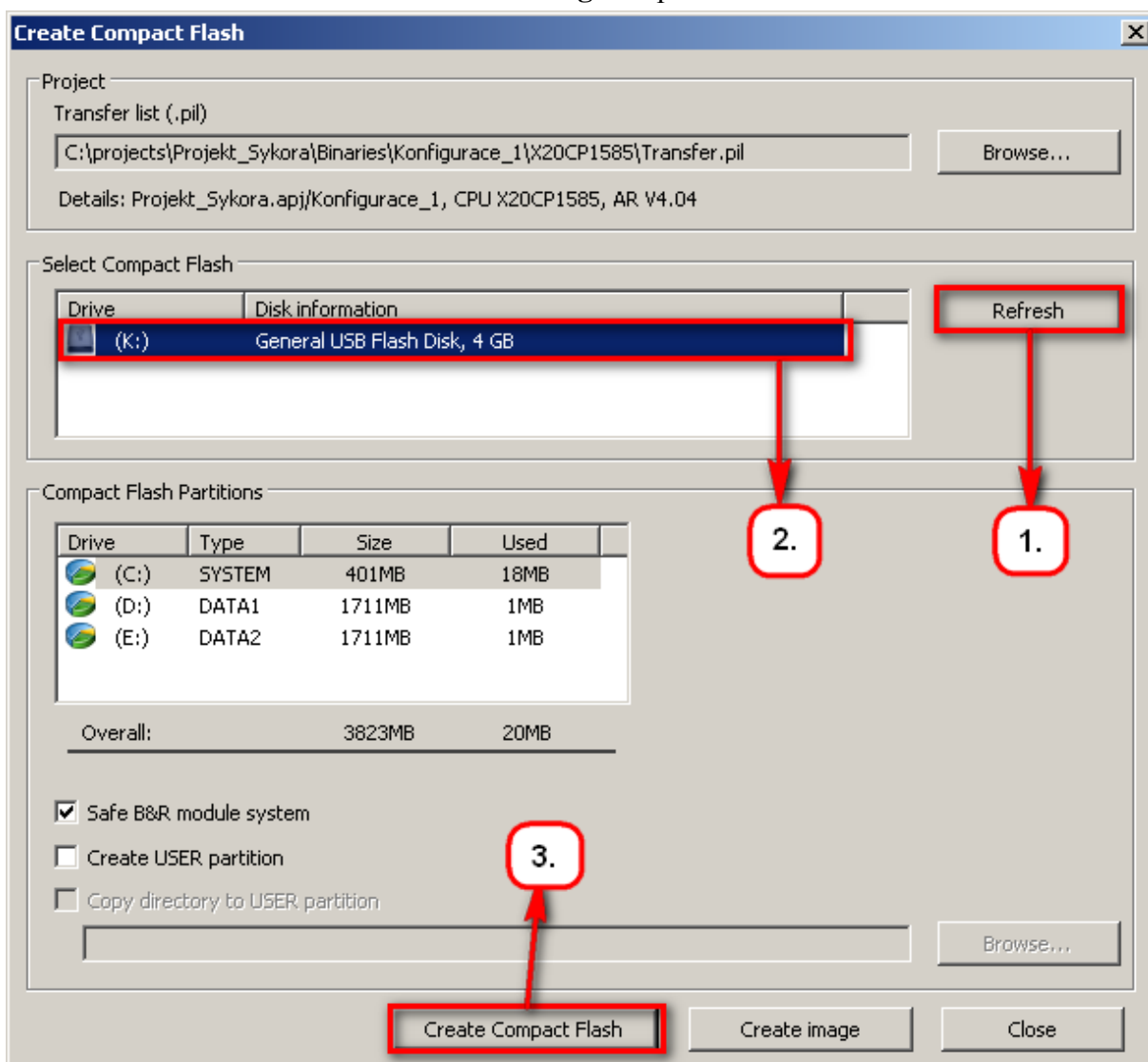
Takto přiřadíte proměnné do všech modulů

5.4 Generování Compact Flash

Při prvním nahrávání použijte k přenosu programu do PLC Compact Flash. Připojte Compact flash ke čtečce a připojte přes USB port k Vašemu počítači. Generaci Compact Flash najdete na hlavní liště na < **Tools** > / < **Create Compact Flash** >

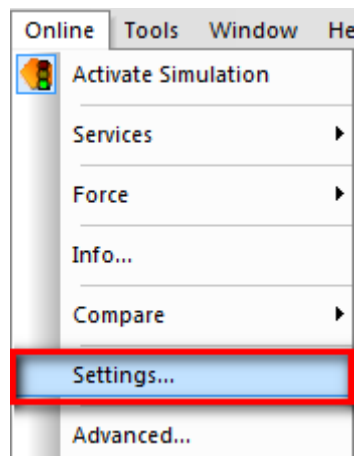


Poté se otevře okno, kde dáte **Refresh** a vyberte Vaši Compact Flash a potvrďte tlačítkem **Create Compact Flash**. Poté jen vložte vygenerovanou Compact Flash do Vašeho PLC. Tlačítkem **Create image** lze provést zálohu

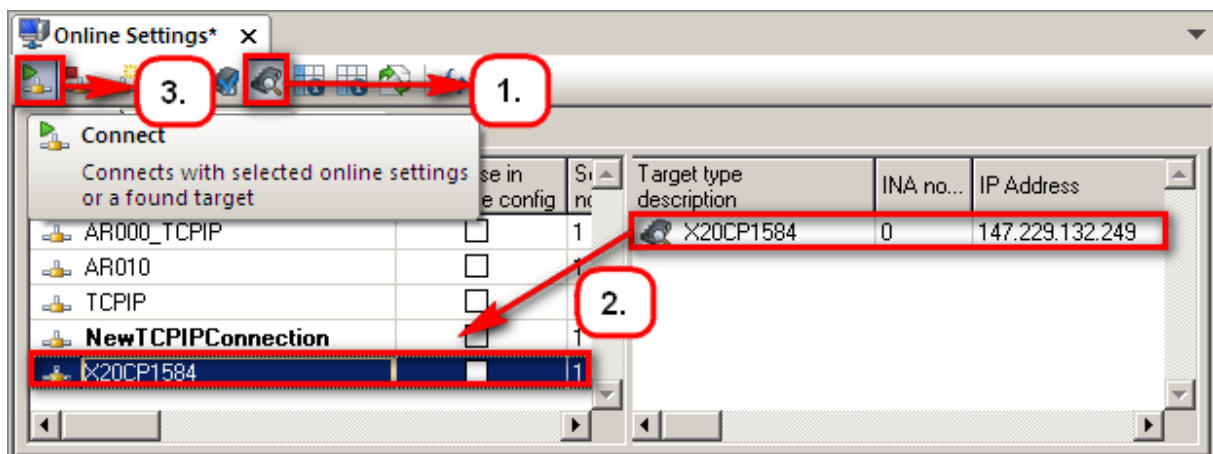


5.5 Spojení s PLC

Online spojení naleznete v hlavní liště < **Online** > / < **Settings** >



Zmáčkněte tlačítko **Browse**, které Vám vyhledá všechny dostupná PLC v pravé části, vyberte Vaše PLC a přetáhněte jej myší do levé části, označte jej v levé části a zmáčkněte tlačítko **Connect**



Po navázání spojení změny programu do PLC nahrajete přes ikonu **Transfer**, již nemusíte generovat Compact Flash.

Vypracoval: Daniel Sýkora
Výukový text vznikl jako příloha bakalářské práce

Sbírka úloh
AUTOMATION STUDIO

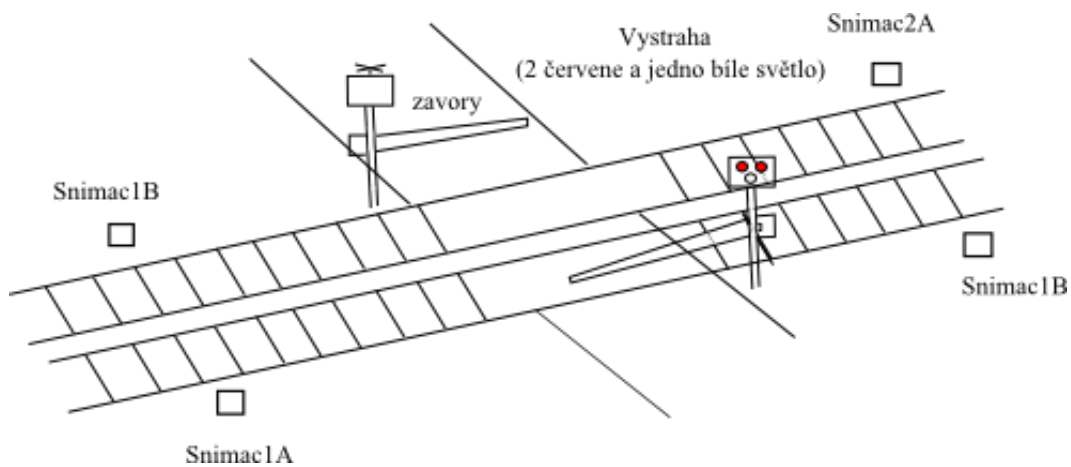
Zadání

Příklad 1:

Navrhněte jednoduché osvětlení domu. Po zapnutí vypínačů *vstup1* a *vstup2* se po 5 sekundách nastaví světlo *vystup* do logické 1, při vypnutí jednoho z vypínačů se světlo nastaví do logické 0.

Příklad 2:

Vytvořte program pro vlakový přejezd. Na přejezdu se uvažují dvě koleje se 4 snímači. Na každé koleji jezdí vlak v jednom směru. Snímače (Snimac1A a Snimac2A) detekují příjezd vlaku na přejezd a snímače (Snimac1B a Snimac2B) detekují odjezd vlaku. Při detekci vlaku se spustí výstraha, poté za 5 s se spustí i závory. Po odjetí všech vagonů trvá 5 s, než se zvednou závory a vypne se výstraha. Při zapnutí výstraže střídavě blikají červená světla s periodou 1 s a bílé světlo je vypnuté. Při vypnutí výstraže červená světla nesvítí a bílé světlo bliká s dvojnásobnou periodou, než blikají červená (tj. 2 s).

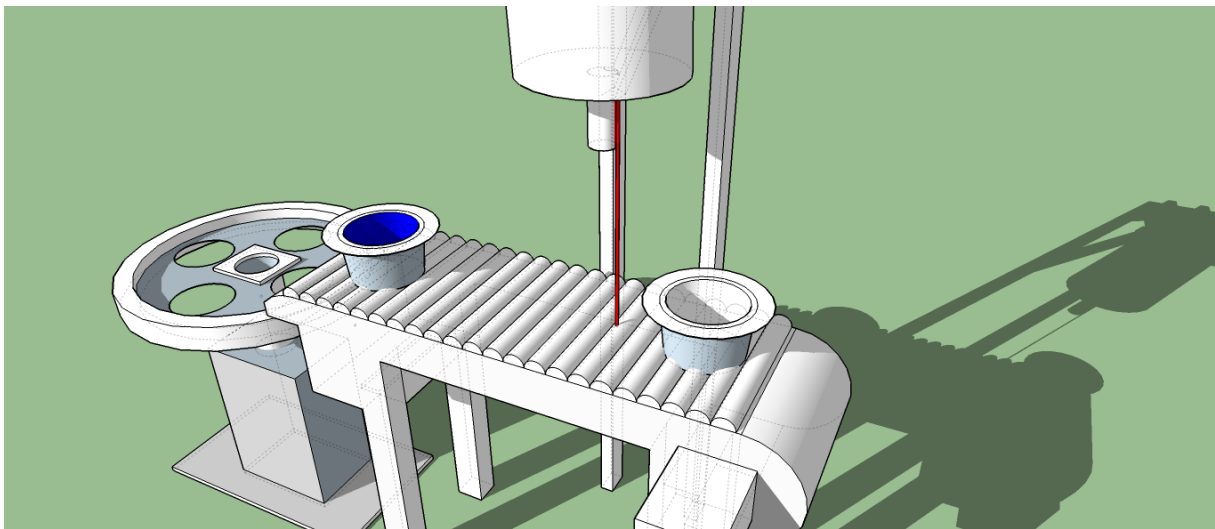


Příklad 3:

Zautomatizujte linku na vstřikování plastů do forem (viz obr). Vytvořte dva samostatné programy. Jeden na ovládání pásu, druhý na ovládání palety. Dopravní pás je poháněn asynchronním motorem, takže se formy po něm pohybují konstantní rychlostí. Bylo navrženo, že z předchozí linky přijíždějí prázdné formy každých 5 s. Tekutý plast je v zásobníku s ventilem.

Paleta je rotačního charakteru. Krokový motor sepne, když je plný slot a paleta pootočí tak, aby byl další prázdný slot připraven na příjezd další plné formy. Pro demonstrativní účely se paleta skládá ze 4 prázdných slotů. Jakmile se naplní všechny sloty, na ovládacím panelu se rozsvítí signalizace pro obsluhu, aby vyměnila paletu (Paleta.Vymenit). Při manipulaci s paletou je motor vypnut. Paleta dále pokračuje k dalšímu zpracování.

Dopravní pás obsahuje kontrolní elektroniku, která hlídá správný chod pásu. Při jakékoliv poruše, nebo při zmáčknutí vypínacího tlačítka (Ovladani.VYP) se okamžitě vypne celá linka. Porucha je signalizována na ovládacím panelu červeným světlem. Při bezproblémovém chodu se na ovládacím panelu vypíše: „Pas v poradku“. Při poruše se vypíše: „POZOR porucha“.



Vzorové řešení

Příklad 1:

```
PROGRAM _CYCLIC

    IF vstup1 AND vstup2 = 1 THEN
        start:=1;
        IF casovac.Q=1 THEN
            vystup:=1;
        END_IF
    ELSE vystup:=0;
        start:=0;
    END_IF

casovac(IN :=start, PT:=T#5s );

END_PROGRAM
```

Příklad 2:

a) Program Snímače

```
PROGRAM _CYCLIC

    IF pocet_naprav < 0 THEN
        pocet_naprav:=0;
    END_IF

R_TRIG_01(CLK:=Snimac1A);
R_TRIG_02(CLK:=Snimac2A);
F_TRIG_01(CLK:=Snimac1B);
F_TRIG_02(CLK:=Snimac2B);

    IF R_TRIG_01.Q THEN
        pocet_naprav:=pocet_naprav+1;
    END_IF

    IF R_TRIG_02.Q THEN
        pocet_naprav:=pocet_naprav+1;
    END_IF

    IF F_TRIG_01.Q THEN
        pocet_naprav:=pocet_naprav-1;
    END_IF

    IF F_TRIG_02.Q THEN
        pocet_naprav:=pocet_naprav-1;
    END_IF

END_PROGRAM
```

b) Program Vlakovy_prejezd

```
PROGRAM _CYCLIC

IF vystraha THEN
    cs1:= casovac2.Q ;
    cs2:= NOT cs1;
    bs:=0;

ELSE
    bsstart:=1;
    bs:=0;
    IF casovac3.Q THEN
        bs:=1 ;
        bsstart:=0;
    END_IF
    cs1:=0;
    cs2:=0;
END_IF

casovac2(IN :=cs2, PT:=T#1s );
casovac3(IN :=bsstart, PT:=T#2s );

CASE stav OF

    0:
        vystraha:=0;
        IF pocet_naprav > 0 THEN
            stav:=1;
        END_IF

    1:
        vystraha:=1;
        start:=1;
        IF casovac.Q THEN
            stav:=2;
            start:=0;
        END_IF

    2:
        zavory:=1;

        IF pocet_naprav = 0 THEN
            stav:=3;
        END_IF

    3:
        start:=1;
        IF casovac.Q THEN
            zavory:=0;
            vystraha:=0;
            stav:=0;
            start:=0;
        END_IF

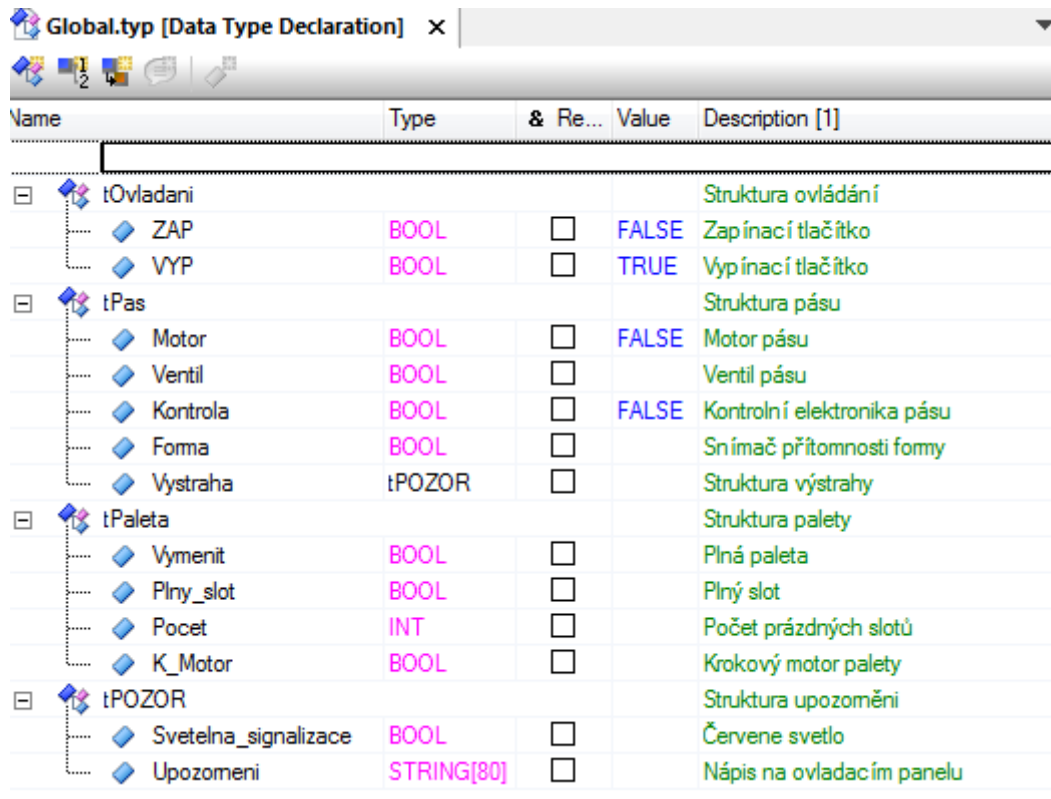
END_CASE

casovac(IN :=start, PT:=T#10s );

END_PROGRAM
```

Příklad 3:

Proměnné řešené pomocí struktur ke zvýšení přehlednosti :



| Name | Type | & Re... | Value | Description [1] |
|----------------------|------------|--------------------------|-------|----------------------------|
| tOvladani | | | | Struktura ovládání |
| ZAP | BOOL | <input type="checkbox"/> | FALSE | Zapínací tlačítko |
| VYP | BOOL | <input type="checkbox"/> | TRUE | Vypínací tlačítko |
| tPas | | | | Struktura pásu |
| Motor | BOOL | <input type="checkbox"/> | FALSE | Motor pásu |
| Ventil | BOOL | <input type="checkbox"/> | | Ventil pásu |
| Kontrola | BOOL | <input type="checkbox"/> | FALSE | Kontrolní elektronika pásu |
| Forma | BOOL | <input type="checkbox"/> | | Snímač přítomnosti formy |
| Vystraha | tPOZOR | <input type="checkbox"/> | | Struktura výstrahy |
| tPaleta | | | | Struktura palety |
| Vymenit | BOOL | <input type="checkbox"/> | | Plná paleta |
| Plny_slot | BOOL | <input type="checkbox"/> | | Plný slot |
| Pocet | INT | <input type="checkbox"/> | | Počet prázdných slotů |
| K_Motor | BOOL | <input type="checkbox"/> | | Krokový motor palety |
| tPOZOR | | | | Struktura upozornění |
| Svetelna_signalizace | BOOL | <input type="checkbox"/> | | Červené světlo |
| Upozomeni | STRING[80] | <input type="checkbox"/> | | Nápis na ovladacím panelu |

a) Program Podavac

```
PROGRAM _CYCLIC
IF Ovladani.VYP OR Pas.Kontrola THEN
  step:=1;
END_IF

CASE step OF
  1:
    Pas.Motor:=0;
    Pas.Ventil:=0;
    IF (NOT Pas.Kontrola AND Ovladani.ZAP) THEN
      step:=2;
    END_IF
  2:
    Pas.Motor:=1;
    start:=1;
    IF casovac.Q THEN
      step:=3;
      start:=0;
    END_IF
  3:
    IF Pas.Forma THEN
      Pas.Ventil:=1;
      step:=4;
    END_IF
```

```

4:      IF Paleta.Pocet = pocet THEN
          step:=5;
          Paleta.Vymenit:=1;
      END_IF
5:      Pas.Ventil:=0;
          start:=1;
          IF casovac.Q THEN
              step:=6;
              start:=0;
          END_IF
6:      IF Paleta.Vymenit = 0 THEN
          Paleta.Pocet:=0;
          step:=2;
      END_IF

END_CASE

R_TRIG_01(CLK := Pas.Forma);
casovac(IN :=start, PT:=T#5s );

IF R_TRIG_01.Q THEN
    Paleta.Pocet:=Paleta.Pocet+1;
END_IF

IF Pas.Forma = 0 THEN
    Pas.Ventil:=0;
END_IF

IF Paleta.Vymenit THEN
    Pas.Motor:=0;
END_IF

IF Pas.Kontrola THEN
    Pas.Vystraha.Upozorneni:='POZOR porucha';
    Pas.Vystraha.Svetelna_signalizace:=1;
ELSE
    Pas.Vystraha.Upozorneni:='Pas v poradku';
    Pas.Vystraha.Svetelna_signalizace:=0;
END_IF

Ovladani.VYP:=NOT Ovladani.ZAP;

END_PROGRAM

```

b) Program Paleta

```

PROGRAM _CYCLIC

    IF Paleta.Plny_slot AND NOT Paleta.Vymenit THEN
        start2:=1;
        Paleta.K_Motor:=1;
        IF casovac2.Q THEN
            Paleta.Plny_slot:=0;
            start2:=0;
            Paleta.K_Motor:=0;
        END_IF
    END_IF

casovac2(IN :=start2, PT:=T#1s );

END_PROGRAM

```