

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

AUTOMATIZACE POMOCÍ B&R LOGICKÝCH AUTOMATŮ
Petr Čepelák

Bakalářská práce
2018

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2017/2018

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Petr Čepelák**
Osobní číslo: **I15257**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Řízení procesů**
Název tématu: **Automatizace pomocí B&R logických automatů**
Zadávající katedra: **Katedra řízení procesů**

Z á s a d y p r o v y p r a c o v á n í :

Cílem bakalářské práce je vytvoření sady demonstračních úloh pro automatizaci pomocí programovatelných logických automatů B&R Automation. Součástí práce je i tvorba aspoň jednoho laboratorního modelu a vypracování srozumitelného návodu pro práci s PLC. Výsledkem bude výukový materiál s popisem metod programování PLC, představení navržených úloh a podrobný popis a komplexní testování vzorových řešení.

Teoretická část: popis základní problematiky programování PLC, úvod k technologiím B&R Automation, popis návrhu laboratorního modelu pro realizaci úlohy.

Praktická část: konstrukce laboratorního modelu, návrh výukových úloh s přehledným popisem a analýzou vzorového řešení, tvorba vizualizace, vytvoření podrobné dokumentace k jednotlivým úlohám.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

KWASNIEWSKI, J. Programmable Logic Controllers. Cracow: ROMA-POL, 2002. ISBN 83-86320-45-1.

BALÁTĚ, J. Automatické řízení. Praha: BEN, 2003. 654 s. ISBN 80-7300-020-2.

Vedoucí bakalářské práce:

doc. Ing. Petr Doležel, Ph.D.

Katedra řízení procesů

Datum zadání bakalářské práce:

6. prosince 2017

Termín odevzdání bakalářské práce:

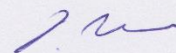
11. května 2018



Ing. Zdeněk Němec, Ph.D.
děkan



L.S.



Ing. Daniel Honc, Ph.D.
vedoucí katedry

V Pardubicích dne 12. prosince 2017

Prohlášení

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 12. 5. 2018

Petr Čepelák

Poděkování

Tímto bych chtěl poděkovat doc. Ing. Petru Doleželovi, Ph.D., vedoucímu bakalářské práce, za vstřícný přístup a cenné rady při zpracování. Dále firmě Andorset s.r.o. za odbornou konzultaci a poskytnutí hardwarových komponent. Poděkovat bych chtěl také své rodině za neustálou podporu a motivaci.

V Pardubicích dne 12. 5. 2018

Petr Čepelák

ANOTACE

Bakalářská práce se zabývá programovatelnými logickými automaty. Popisuje základní problematiku rozdělení a programování průmyslových logických automatů. Seznamuje s produkty společnosti B&R a vývojovým prostředím Automation Studio. Představuje zkonstruovaný laboratorní model a sestavené laboratorní úlohy.

KLÍČOVÁ SLOVA

PLC, B&R, Automation Studio, laboratorní model, testovací úlohy.

TITLE

AUTOMATION BY B&R LOGIC CONTROLLERS

ANNOTATION

The bachelor thesis is about programmable logic controllers. The thesis describes the basic issue of distribution and programming of programmable logic controllers. It familiarizes the reader with B&R Automation products and development environment, Automation Studio. The work also introduces a constructed laboratory model and proposed laboratory tasks.

KEYWORDS

PLC, B&R, Automation Studio, laboratory model, testing tasks.

OBSAH

Seznam zkratk a značek	10
Seznam symbolů proměnných veličin a funkcí	11
Seznam ilustrací	12
Seznam tabulek	15
ÚVOD	16
1 OBECNÉ SEZNÁMENÍ S PLC	17
1.1 Výhody řízení pomocí PLC	17
1.2 Typy PLC	18
1.2.1 Mikro.....	18
1.2.2 Kompaktní.....	18
1.2.3 Modulární.....	19
1.3 Vykonávání programu PLC	19
1.4 Programovací jazyky.....	20
1.4.1 Strukturovaný text.....	21
1.4.2 Seznam instrukcí	21
1.4.3 Příčkový diagram	22
1.4.4 Schéma funkčních bloků	23
1.4.5 Sekvenční funkční diagram.....	23
2 AUTOMATIZACE POMOCÍ B&R AUTOMATION.....	25
2.1 Představení společnosti	25
2.2 Programovatelné automaty (systémy X20 a X67)	26
2.3 Komunikace	27
2.3.1 „Powerlink“	27
2.3.2 „X2X Link“	27
2.3.3 „CANopen“	27
2.4 „Opensafety“ a redundance.....	28
2.5 „Mapp“ technologie	28
2.6 „Reaction“	29
3 AUTOMATION STUDIO	30
3.1 „Automation runtime“	30
3.2 Úvodní strana	30
3.3 Pracovní prostředí	31
3.4 Program.....	32

3.5	Simulace.....	33
3.6	Diagnostické nástroje.....	33
3.6.1	„Watch“.....	33
3.6.2	„Trace“.....	33
3.6.3	Správce diagnostických nástrojů.....	34
3.7	Vizualizace a vizualizační prostředí.....	35
3.8	Nápověda.....	36
3.9	Konfigurace reálného zařízení.....	37
4	LABORATORNÍ MODEL.....	38
4.1	Zkonstruovaný laboratorní model.....	39
4.2	Použitá zařízení.....	40
4.2.1	PLC X20CP0482.....	40
4.2.2	Napájecí modul X20PS9600.....	41
4.2.3	Vstupní digitální modul X20DI9371.....	42
4.2.4	Výstupní digitální modul X20DO6322.....	43
4.2.5	Vstupní analogový modul X20AT2222.....	44
4.3	Vstupy a výstupy.....	45
5	ZADÁNÍ LABORATORNÍCH ÚLOH.....	47
5.1	Úloha č. 1.....	47
5.2	Úloha č. 2.....	47
5.3	Úloha č. 3.....	48
5.4	Úloha č. 4.....	48
6	NÁVOD AUTOMATION STUDIO.....	50
6.1	Získání softwaru.....	50
6.2	Instalace balíčků komponent.....	50
6.3	Založení projektu.....	51
6.4	Přidání hardwarového modulu.....	54
6.5	Konfigurace analogového modulu.....	55
6.6	Přidání knihovny.....	56
6.7	Vytvoření programu.....	57
6.8	Tvorba programu.....	57
6.9	Mapování proměnných.....	58
6.10	Simulátor.....	59
6.11	Diagnostický nástroj „Watch“.....	59
6.12	Vytvoření vizualizačního prostředí.....	60
6.13	Připojení k PLC pomocí sítě Ethernet.....	62

7	VZOROVÁ ŘEŠENÍ LABORATORNÍCH ÚLOH.....	66
7.1	Úloha č. 1	66
7.1.1	Psaní programu	67
7.1.2	Mapování proměnných	69
7.2	Úloha č. 2	70
7.3	Úloha č. 3	71
7.4	Úloha č. 4	73
7.4.1	Tvorba vizualizace	75
8	ZÁVĚR	78
	POUŽITÁ LITERATURA	79
	PŘÍLOHY	81

SEZNAM ZKRATEK A ZNAČEK

A/D	analogově/digitální
B&R	Bernecker & Rainer
CAN	řídící síť
CAT 4	kategorie 4
CNC	číslicové řízení pomocí počítače
DI	digitální vstup
DO	digitální výstup
DIV	dělit
EPL	Ethernet Powerlink
ETH	Ethernet
GE	větší nebo zároveň
GND	uzemnění
HW	hardware
I/O	vstup/výstup
LE	menší nebo zároveň
LED	světelná dioda
MUL	násobit
PLC	programovatelný logický automat
PLe	úroveň typu „e“
REAL	datový typ REAL
RS232	komunikační sériová linka
SIL	úroveň integrity bezpečnosti
UINT	neznaménkové celé číslo
USB	univerzální sériová sběrnice
VC	vizualizační prostředí
X2X	komunikační sběrnice

SEZNAM SYMBOLŮ PROMĚNNÝCH VELIČIN A FUNKCÍ

U elektrické napětí, V

SEZNAM ILUSTRACÍ

Obr. 1.1 – Ukázka kompaktního PLC.....	18
Obr. 1.2 – Ukázka modulárního PLC	19
Obr. 1.3 – Cyklus programu PLC	20
Obr. 1.4 – Strukturovaný text	21
Obr. 1.5 – Seznam instrukcí.....	22
Obr. 1.6 – Příčkový diagram.....	22
Obr. 1.7 – Schéma funkčních bloků	23
Obr. 1.8 – Sekvenční funkční diagram	24
Obr. 2.1 - B&R pod křídly ABB.....	25
Obr. 2.2 – Hlavní sídlo společnosti B&R	25
Obr. 2.3 – Prvky systému řady X20.....	26
Obr. 2.4 – I/O systém řady X67	26
Obr. 2.5 – Bezpečnostní modul	28
Obr. 3.1 – Úvodní strana softwaru Automation Studio	30
Obr. 3.2 – Pracovní prostředí softwaru Automation Studio	31
Obr. 3.3 – Cyklické třídy	32
Obr. 3.4 – Diagnostický nástroj Watch.....	33
Obr. 3.5 – Diagnostický nástroj Trace.....	33
Obr. 3.6 – Správce diagnostických nástrojů	34
Obr. 3.7 – Vizualizační prostředí.....	35
Obr. 3.8 – Pracovní prostředí nápovědy	36
Obr. 3.9 – Configuration View	37
Obr. 3.10 – Physical View	37
Obr. 4.1 – Model křižovatky od společnosti Edumat	38
Obr. 4.2 – Model dopravníkového pásu od společnosti Stroza s.r.o.	38
Obr. 4.3 – Laboratorní model	39
Obr. 4.4 – Modul X20CP0482 s LED indikátory	41
Obr. 4.5 – Modul X20PS9600 s LED indikátory	42
Obr. 4.6 – Modul X20DI9371 s LED indikátory.....	43
Obr. 4.7 – Modul X20DO6322 s LED indikátory	44
Obr. 4.8 – Modul X20AT2222 s LED indikátory.....	45
Obr. 6.1 – Online instalace balíčků.....	50

Obr. 6.2 – Ruční instalace balíčků	51
Obr. 6.3 – Volba možnosti New Project	51
Obr. 6.4 – Položka An empty project	51
Obr. 6.5 – Základní nastavení projektu.....	52
Obr. 6.6 – Založení nové hardwarové konfigurace	52
Obr. 6.7 – Přidání hlavního modulu do HW konfigurace.....	53
Obr. 6.8 – Přidání Bus Base modulu do HW konfigurace.....	53
Obr. 6.9 – Přidání HW modulu.....	54
Obr. 6.10 – Spuštění konfigurace modulu X20AT2222	55
Obr. 6.11 – Konfigurace modulu X20AT2222	55
Obr. 6.12 – Vyhledání položky B&R Libraries	56
Obr. 6.13 – Implementace knihovny standard	56
Obr. 6.14 – Vytvoření programu	57
Obr. 6.15 – Záložka Logical View	58
Obr. 6.16 – Mapování proměnných	58
Obr. 6.17 – Aktivace simulátoru.....	59
Obr. 6.18 – Signalizace spuštěného simulátoru.....	59
Obr. 6.19 – Spuštění diagnostické nástroje Watch	59
Obr. 6.20 – Diagnostický nástroj Watch.....	60
Obr. 6.21 – Implementace vizualizačního prostředí	60
Obr. 6.22 – Konfigurace vizualizačního prostředí.....	61
Obr. 6.23 – Výběr šablony.....	61
Obr. 6.24 –Přiřazení vizualizačního prostředí aktivní HW konfiguraci	62
Obr. 6.25 – Spuštění konfigurace připojení	62
Obr. 6.26 – Vyhledání připojených PLC	63
Obr. 6.27 – Nastavení IP adresy PLC	63
Obr. 6.28 – Konfigurace Ethernetu na cílovém zařízení	64
Obr. 6.29 – Zadání IP adresy do položky Mode	64
Obr. 6.30 – Připojení PLC	65
Obr. 6.31 – Tlačítko Transfer	65
Obr. 7.1 – Číslování sítě	66
Obr. 7.2 – Vzorové řešení úlohy č. 1	67
Obr. 7.3 – Volba cyklické části Cyclic.ld	68
Obr. 7.4 – Ikonka Add Variable a seznam proměnných.....	68

Obr. 7.5 – Položka Variables.var	68
Obr. 7.6 – Položka I/O Mapping.....	69
Obr. 7.7 – Mapování proměnných pro úlohu č. 1	69
Obr. 7.8 – Vzorové řešení úlohy č. 2	70
Obr. 7.9 – Vzorové řešení úlohy č. 3	72
Obr. 7.10 – Vzorové řešení úlohy č. 4	74
Obr. 7.11 – Spuštění vizualizačního prostředí	75
Obr. 7.12 – Výběr položky Init_Page	75
Obr. 7.13 – Pracovní plocha s vizualizačními prvky	76
Obr. 7.14 – Konfigurační okno	76
Obr. 7.15 – Vzorové řešení vizualizace pro úlohu č. 4.....	77

SEZNAM TABULEK

Tab. 4.1 – Popis LED indikace modulu X20CP0482	41
Tab. 4.2 – Popis LED indikace modulu X20PS9600	42
Tab. 4.3 – Popis LED indikace modulu X20DI9371	43
Tab. 4.4 – Popis LED indikace modulu X20DO6322	44
Tab. 4.5 – Popis LED indikace modulu X20AT2222	45
Tab. 4.6 – Použité vstupy modulu X20DI9371	46
Tab. 4.7 – Použité výstupy modulu X20DO6322	46
Tab. 4.8 – Použitý vstup modulu X20AT2222	46
Tab. 5.1 – Pravdivostní tabulka k úloze č. 2	47
Tab. 7.1 – Mapování modulu X20DI9371 pro úlohu č. 1	67
Tab. 7.2 – Mapování modulu X20DO6322 pro úlohu č. 1	67
Tab. 7.3 – Mapování modulu X20DI9371 pro úlohu č. 2	71
Tab. 7.4 – Mapování modulu X20DO6322 pro úlohu č. 2	71
Tab. 7.5 – Mapování modulu X20DI9371 pro úlohu č. 3	72
Tab. 7.6 – Mapování modulu X20DO6322 pro úlohu č. 3	73
Tab. 7.7 – Mapování modulu X20DO6322 pro úlohu č. 4	74
Tab. 7.8 – Mapování modulu X20AT2222 pro úlohu č. 4	74

ÚVOD

Cílem bakalářské práce je sestavení sady laboratorních úloh, které demonstrují práci s logickými automaty společnosti B&R Automation. Dalším důležitým bodem je zkonstruování laboratorního modelu, pomocí kterého si lze otestovat funkčnost vytvořených aplikací.

V teoretické části je čtenář seznámen s logickými programovatelnými automaty a jejich metodami programování. Následuje popis produktů společnosti B&R včetně podrobného rozboru vývojového prostředí Automation Studio. V poslední kapitole teoretické části je popsán zhotovený laboratorní model.

Praktická část věnuje pozornost navržené sadě laboratorních úloh. Součástí je přehledný návod popisující úkony potřebné pro realizaci aplikací ve vývojovém prostředí Automation Studio. Praktickou část uzavírají vzorová řešení sestavených zadání.

V závěru práce je shrnut a diskutován praktický přínos bakalářské práce a jednotlivých laboratorních úloh.

1 OBECNÉ SEZNÁMENÍ S PLC

PLC („Programmable Logic Controller“) neboli programovatelný logický automat či pouze programovatelný automat je uživatelsky programovatelný řídicí systém přizpůsobený pro řízení technologických a průmyslových procesů.

V dnešní době programovatelné logické automaty zasahují téměř do všech oblastí průmyslu. Jsou základním kamenem pro řízení automatizovaných linek, robotů, dopravy a mnoha dalších oblastí. Značné využití mají i mimo průmysl. Lze je využít pro realizaci inteligentního domu, klimatizačního zařízení, dále pro ovládání garážových vrat, mycích linek, balicích strojů, prodejních automatů apod. (Šmejkal, 1999).

1.1 VÝHODY ŘÍZENÍ POMOCÍ PLC

- Rychlá realizace – možnost rychlé realizace systému je jedna z hlavních předností programovatelných automatů. Stačí navrhnout a včas zakoupit vhodnou sestavu modulů PLC pro danou aplikaci, zhotovit projekt, napsat a odladit uživatelský program. Následně vše realizovat a uvést do chodu.
- Spolehlivost, odolnost a diagnostika – po technické stránce jsou programovatelné logické automaty navrženy tak, aby byly extrémně spolehlivé i v drsných průmyslových podmínkách. Musí být schopny odolávat působení vnějších vlivů, rušení i poruchám. Velká většina programovatelných logických automatů bývá vybavena diagnostickými funkcemi, které slouží pro kontrolu činnosti systému a případnou závadu snadno lokalizují.
- Nekončící změny v zadání – požadavky na výsledné řešení se průběžně vyvíjejí a rozšiřují. Mállokdy se podaří, že první varianta zůstane tou poslední. Průběžné opravy, rozšíření nebo změny uživatelského programu jsou při použití programovatelného logického automatu snadno realizovatelné. Také může dojít k situaci, kdy nové požadavky vyžadují vyšší počet vstupů a výstupů. V takovém případě stačí doplnit potřebné rozšiřující moduly (Šmejkal, 1999).

1.2 TYPY PLC

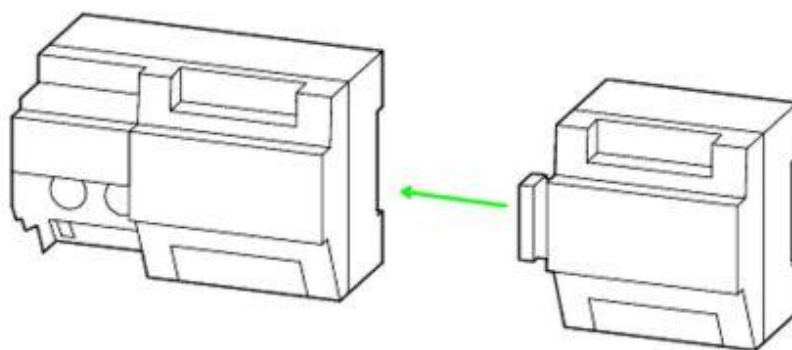
Skutečnou hardwarovou konfiguraci volí uživatel na základě požadavků řešené úlohy. Každý projekt průmyslové automatizace vyžaduje odlišné zacházení se vstupními/výstupními veličinami, jiný přístup k řízení a jiné vlastnosti PLC. Z důvodu rozmanitosti lze programovatelné logické automaty rozdělit do těchto skupin: mikro, kompaktní a modulární (Šmejkal, 1999).

1.2.1 Mikro

Jde o nejmenší a nejlevnější typ programovatelného logického automatu, který je vhodný pro realizaci nejméně technologicky náročných strojů a mechanismů. Nabízená soustava je pevná, bez možnosti případného rozšíření o další moduly. Obvykle tato varianta řešení poskytuje určitý počet vstupů a výstupů, např. 6 binárních vstupů / 6 binárních výstupů (Šmejkal, 1999).

1.2.2 Kompaktní

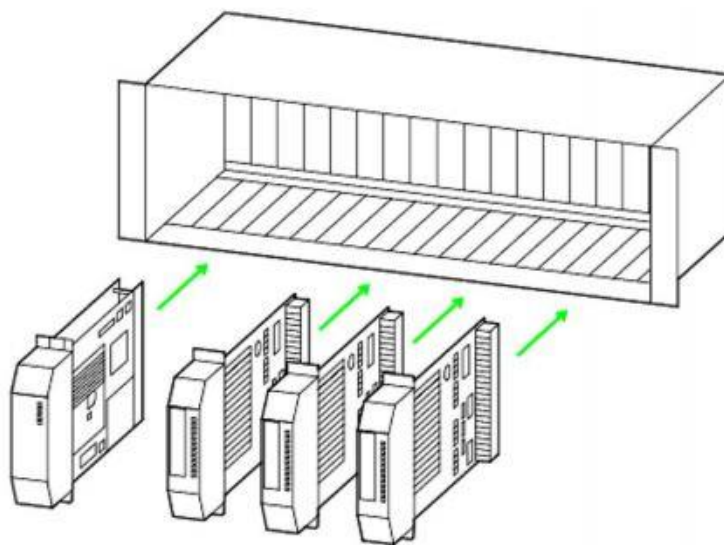
PLC v kompaktním provedení, znázorněném na obr. 1.1, již nabízí určitý stupeň volnosti. Skládají se ze základního modulu, který lze rozšířit o další přídatné moduly. Počet a výběr rozšiřujících modulů je omezený, jako příklad lze uvést modul s binárními či analogovými vstupy a výstupy, modul rychlých čítačů, modul regulátorů apod. (Šmejkal, 1999).



Obr. 1.1 – Ukázka kompaktního PLC (Zemanová, 2010, s. 19)

1.2.3 Modulární

Modulární programovatelné logické automaty, uvedené na obr. 1.2, nabízejí nesrovnatelně větší volnost ve volbě konfigurace a jsou používány pro nejnáročnější průmyslové aplikace. Celý systém je tvořen několika moduly sestavenými do jednoho celku, který lze jednoduše rozšiřovat o další potřebné periferie, a to v nepoměrném rozsahu oproti kompaktním PLC (Šmejkal, 1999).



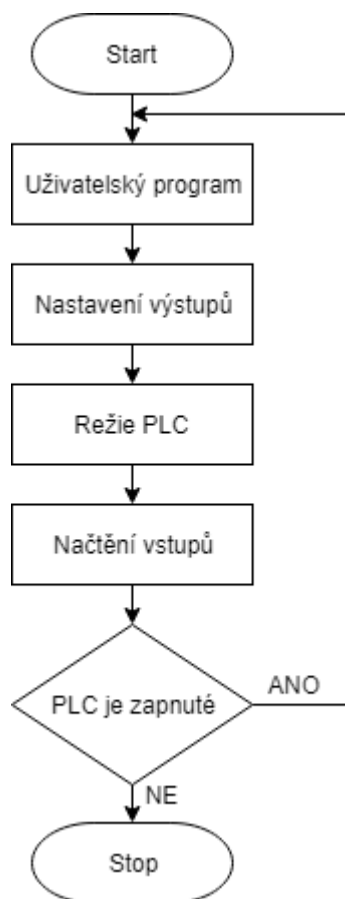
Obr. 1.2 – Ukázka modulárního PLC (Zemanová, 2010, s. 22)

1.3 VYKONÁVÁNÍ PROGRAMU PLC

Program PLC je tvořen posloupností instrukcí a příkazů jazyka. Základním znakem je jeho cyklické vykonávání v programové smyčce, a proto se programátor nemusí starat o to, aby po konci programu vrátil jeho vykonávání opět na začátek. Naopak při dlouhodobém setrvání programu v programové smyčce může dojít k překročení doby cyklu a následnému selhání.

Způsob vykonávání programového cyklu je znázorněn na obr. 1.3. Po každém vykonání poslední instrukce uživatelského programu je předáno řízení systémovému programu, který zajistí tzv. otočku cyklu. V ní jsou nejprve na výstupní periferie vyslány aktuálně vyčíslené hodnoty obrazů výstupů, dále jsou provedeny režijní operace systému, které zahrnují aktualizaci časových údajů pro časovače a systémové registry, ošetření komunikace, diagnostiku a řadu dalších režijních úkonů. Na závěr jsou načteny aktuální hodnoty fyzických vstupů, které jsou po dobu následujícího cyklu uloženy do paměťových obrazů. Poté je opět předáno řízení první instrukci uživatelského programu.

Pro program PLC je typické, že nepracuje s aktuálními hodnotami vstupů a výstupů, ale s jejich obrazy uloženými v registrech. Aktualizace jejich hodnot je prováděna pouze v otočce cyklu, díky tomu je zajištěna synchronizace vstupních a výstupních dat s během programu (Šmejkal, 1999).



Obr. 1.3 – Cyklus programu PLC

1.4 PROGRAMOVACÍ JAZYKY

Původně každý z výrobců PLC nabízel své systémy s vlastními variantami programovacích jazyků, a to postupem času přineslo značné komplikace. Tuto nepřehlednost následně vyřešila mezinárodní norma IEC 61131-3, která sjednocuje programovací jazyky pro PLC. Základní programovací jazyky lze rozdělit na textové a grafické:

- textové: strukturovaný text (Structured Text), seznam instrukcí (Instruction List),
- grafické: příčkový diagram (Ladder Diagram), schéma funkčních bloků (Function Block Diagram) a jazyk sekvenčních funkčních diagramů (Sequential Function Chart) (Kochaniček, 2010).

Pro demonstraci jednotlivých programovacích jazyků je u každého z nich uvedena ukázka programu vytvořeného v daném jazyce.

1.4.1 Strukturovaný text

Strukturovaný text, znázorněný na obr. 1.4, je obdobou vyšších programovacích jazyků (např. Pascal nebo C). Díky své úspornosti a názornému zápisu algoritmů je vhodný pro programování složitějších logických úloh a definici komplexních funkčních bloků, které mohou být dále použity v jiných programovacích jazycích. Jednotlivé příkazy jsou odděleny středníkem (Šmejkal, 1999; Kochaníček, 2010).

```
PROGRAM _CYCLIC
  IF Tlacitko THEN
    Svetlo := TRUE;
  END_IF

  IF Svetlo THEN
    Casovac_Start := 1;
    Casovac_Konec := Casovac.Q;

    IF Casovac_Konec THEN
      Svetlo := 0;
      Casovac_Start := 0;
      Casovac_Konec := 0;
    END_IF
  END_IF

  Casovac(IN := Casovac_Start , PT := T#10s );

END_PROGRAM
```

Obr. 1.4 – Strukturovaný text

1.4.2 Seznam instrukcí

Seznam instrukcí, uvedený na obr. 1.5, patří do skupiny textových jazyků a je obdobou strojově orientovaného assembleru. Samotný program se skládá z posloupnosti instrukcí (např. ADD pro součet dvou hodnot). Každá instrukce začíná na samotném řádku a odpovídá ji stejně pojmenovaný příkaz jazyka. Tento programovací jazyk umožňuje efektivní využití instrukční sady PLC a nejlepší přizpůsobení úlohy vzhledem ke konfiguraci hardwaru (Šmejkal, 1999).

```

PROGRAM _CYCLIC
  LD ib1_Tlacitko
  S obl_Svetlo
  LDN obl_Svetlo
  JMPC Navesti
  LD 1
  ST Casovac_Start
  LD T#10s
  ST Casovac.PT
  LD Casovac.Q
  ST Casovac_Konec
  R obl_Svetlo
  R Casovac_Konec
  R Casovac_Start
  Navesti:
  CAL Casovac(IN := Casovac_Start , PT := T#10s )

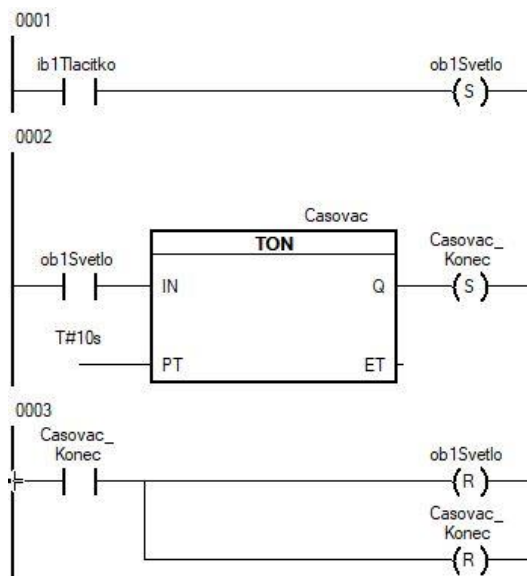
END_PROGRAM

```

Obr. 1.5 – Seznam instrukcí

1.4.3 Příčkový diagram

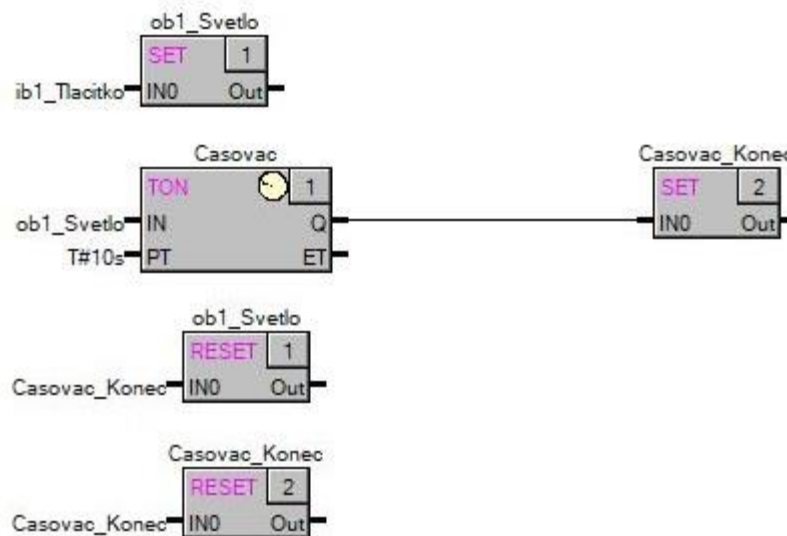
Příčkový diagram, s ukázkou na obr. 1.6, je založen na grafické interpretaci reléové logiky a vizuálně je velice podobný liniovým schémátům v elektrotechnice. Síť kontaktů, cívek, funkcí a funkčních bloků je zleva i zprava ohraničena svislými čarami, které reprezentují levou a pravou napájecí sběrnici. Jazyk je vhodný pro programování jednodušších logických úloh a zvládne jej obsluhovat i personál, který nezná složitější programovací jazyky. Při programování rozsáhlejších logických úloh se stává nepřehledným (Šmejkal, 1999; Kochaníček, 2010; Vojáček, 2011).



Obr. 1.6 – Příčkový diagram

1.4.4 Schéma funkčních bloků

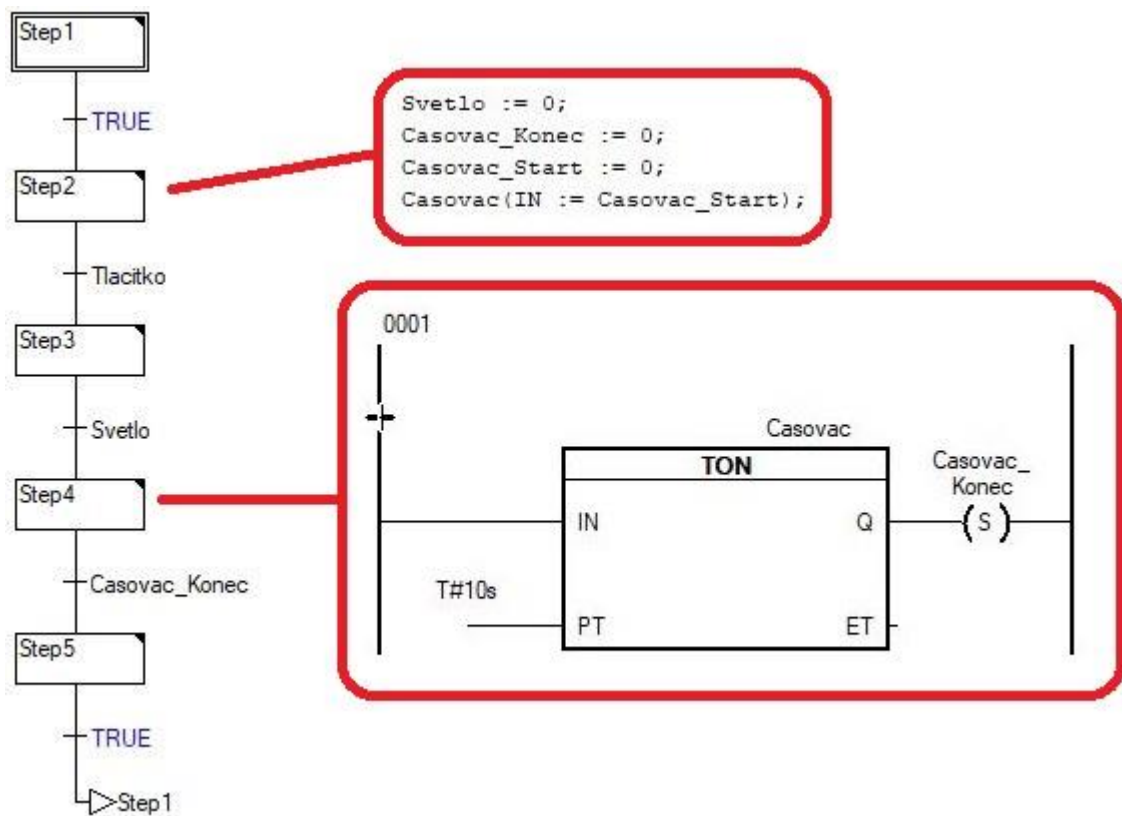
Schéma funkčních bloků, znázorněné na obr. 1.7, je dalším ze zástupců grafických programovacích jazyků. Program je sestaven z obdélníkových značek, které popisují základní logické operace, čítače, časovače apod. Podobně jako u příčkového diagramu jsou i zde jednotlivé funkční bloky liniově propojeny. Jazyk je ideální pro práci s logickými signály, nicméně při programování složitých algoritmů ztrácí na přehlednosti a pozdější korekce programu jsou složitější (Šmejkal, 1999; Kochaníček, 2010; Vojáček, 2011).



Obr. 1.7 – Schéma funkčních bloků

1.4.5 Sekvenční funkční diagram

Grafický jazyk sekvenčních funkčních diagramů, uvedený na obr. 1.8, funguje na principu vývojového diagramu. Skládá se z posloupnosti kroků a přechodů. Každému kroku je přiřazen blok akcí, který může být naprogramován v libovolném programovacím jazyce definovaném normou (programování bloků různými programovacími jazyky je také znázorněno na obr. 1.8). Jednotlivé kroky jsou spojeny podmínkami, které musí být splněny pro aktivaci následujícího bloku. Jazyk nabízí velmi přehledný zápis chování programu, nicméně je nevhodný pro přímou realizaci zpracování analogových signálů (Kochaníček, 2010; Vojáček, 2011).



Obr. 1.8 – Sekvenční funkční diagram

2 AUTOMATIZACE POMOCÍ B&R AUTOMATION

Ve světě existuje nepřeberné množství výrobců programovatelných logických automatů. Mezi ty nejvýznamnější a nám nejbližší patří česká společnost Teco, německý Siemens, japonské Mitsubishi či americký Rockwell Automation. Významnou pozici na trhu zastává i společnost B&R Automation (dále také B&R). Právě této společnosti, jejím produktům a technologiím se bude bakalářská práce věnovat.

2.1 PŘEDSTAVENÍ SPOLEČNOSTI

Rakouská společnost B&R Automation byla založena v roce 1979 s hlavním sídlem v rakouském městě Eggelsberg (obr. 2.2). Na českém trhu působí od roku 1997. Ve vlastnictví zakladatelů Erwina Berneckera a Josefa Rainera se společnost stala největším nezávislým dodavatelem automatizační techniky.

K zásadní události došlo v dubnu 2017, kdy se společnost stala součástí korporace ABB (obr. 2.1). B&R vyplnila historicky vzniklou mezeru v nabídce automatizace ABB a společně tvoří jediného dodavatele v oboru průmyslové automatizace, který nabízí úplné spektrum technologií a softwarového řešení (Akvizice společnosti B&R společností ABB, 2017)

Společnost B&R zajišťuje vše potřebné pro automatizaci jednoduchých strojů i složitých systémů. Zakládá si na hladké spolupráci mezi jednotlivými produkty. Pro programování a konfiguraci všech produktů B&R slouží softwarové rozhraní Automation Studio (Produkty, 2018).



Obr. 2.1 - B&R pod křídly ABB (Akvizice společnosti B&R společností ABB, 2017)



Obr. 2.2 – Hlavní sídlo společnosti B&R (Chalupa, 2016)

2.2 PROGRAMOVATELNÉ AUTOMATY (SYSTÉMY X20 A X67)

Společnost B&R nabízí programovatelné automaty ve dvou provedeních, jde o řady typu X20 a X67. Názvy odpovídají stupni krytí, který daný systém nabízí – IP20 a IP67 (I/O Systems, 2018).

PLC typu X20 umožňuje kombinovat potřebné komponenty v závislosti na požadavcích uživatele. Zakládá si na bezproblémové integraci, vysokém výkonu a flexibilitě. Z konstrukčního hlediska lze systém X20 rozložit na tři prvky, které jsou uvedeny na obr. 2.3: svorkovnice, elektronický modul a modul se sběrníci. Mechanické rozdělení s sebou přináší mnoho výhod. Jednou z nich je efektivnější využití prostoru pro dosažení vyšší „hustoty“ komponent, dále také umožňuje vyměňování elektroniky za chodu systému a do svorek lze zapojit až tři vodiče, tudíž nejsou potřeba další propojovací svorky (X20 Systems, 2018).

Systémy X67, zobrazené na obr. 2.4, jsou konstruovány pro nasazení do náročných podmínek. Nabízí kompatibilitu s prvky X20 a lze je podle potřeby kombinovat (X67 Systems, 2018).



Obr. 2.3 – Prvky systému řady X20
(X20 Systems, 2018)



Obr. 2.4 – I/O systém řady X67
(X67 Systems, 2018)

2.3 KOMUNIKACE

Obecně jsou průmyslové komunikační sítě používány k přenosu dat mezi systémy automatického řízení. Oproti počítačovým sítím musí být schopny přenášet data formou krátkých zpráv s velkou četností a komunikovat v reálném čase. Důležitá je i odolnost a spolehlivost síťových prvků, které bývají často nasazovány do nepříznivých průmyslových podmínek.

2.3.1 „Powerlink“

Průmyslový komunikační protokol Ethernet Powerlink (dále také EPL) byl vydán v roce 2001 společností B&R. EPL nabízí vysoký výkon a funkce reálného času na základě standardu Ethernet, proto je plně kompatibilní se všemi systémy určené pro Ethernet a nevyžaduje žádný dodatečný hardware. Pro zajištění přenosu dat v reálném čase se v EPL síti používají rozbočovače („hub“), u kterých na rozdíl od přepínačů („switch“) nedochází k nežádoucímu zpoždění. Dalším mechanismem pro zvýšení výkonu a propustnosti sítě je využití komunikačního modelu producent/konzument (Zezulka, 2008).

Díky přenosové rychlosti až 100 Mb/s a přesnosti synchronizace +/- 100 μ s umožňuje zkombinovat do jediné sítě i ty nejnáročnější úkoly řízení, robotiky a CNC (Powerlink, 2018).

2.3.2 „X2X Link“

Komunikaci mezi jednotlivými moduly zajišťuje sběrnice X2X Link. Spojené moduly od sebe mohou být navzájem vzdáleny až do vzdálenosti 100 m. Lze kombinovat prvky různých řad (X20 a X67). Každý z modulů obsahuje X2X přijímač a vysílač. Veškerá data, která dorazí na přijímač, jsou okamžitě odeslána na vysílač (BÍLEK, 2005).

2.3.3 „CANopen“

Komunikační protokol CANopen je vyvinutý pro stroje s víceosým pohybem a řízení vzdálených I/O systémů. „CAN“ (neboli „Controller Area Network“) sběrnice zajišťuje vysokou odolnost proti rušení, vysokorychlostní přenos dat a snadnou implementaci do systému (CANopen, 2018).

2.4 „OPENSAFETY“ A REDUNDANCE

Pro zajištění bezpečnosti obsluhy a systému lze využít bezpečnostní modul (obr. 2.5) komunikující na bezpečnostním protokolu OpenSAFETY. Protokol je plně nezávislý na typu sběrnice, tudíž ho lze použít se všemi systémy průmyslových sběrnic (OpenSAFETY, 2018).

Dalšího stupně zabezpečení systému lze dosáhnout při využití tzv. redundance. Jde o velmi snadnou implementaci sekundárního PLC, které v případě poruchy primárního PLC převezme jeho funkci. Redundanci lze realizovat i pro komunikační kabeláž. Při mechanickém poškození kabeláže systém automaticky přepne na funkční (sekundární) spoj, který vede jinou trasou (Redundance, 2018).



Obr. 2.5 – Bezpečnostní modul (Taking sorting to a whole new level, 2018)

2.5 „MAPP“ TECHNOLOGIE

Mapp technologie nabízí předpřipravené modulární bloky, které výrazně usnadňují a urychlují vývoj softwaru pro nové systémy. Programátor již nemusí ztrácet čas vývojem základních funkcí. Stačí do projektu implementovat předpřipravené softwarové bloky, které tyto funkce obsahují a určit jejich parametry. Jednotlivé Mapp komponenty jsou součástí vývojového prostředí Automation Studio (B&R zkracuje dobu vývoje o 67 % prostřednictvím Mapp Technology, 2015).

Mapp komponenty lze rozdělit do těchto kategorií:

- „Mapp Control“ – obsahuje bloky usnadňující práci s regulátory, simulací soustav, manipulaci se zavěšeným břemenem, řízení hydrauliky apod.
- „Mapp View“ – vizualizace na bázi webových technologií, které lze zobrazit na jakémkoli zařízení a přizpůsobit konkrétním požadavkům konkrétních uživatelů.

Veškerá grafická rozhraní jsou implementována pomocí předpřipravených bloků, které i poté lze editovat.

- „Mapp Motion“ – slouží k jednodušší konfiguraci pohybu CNC strojů a robotů.
- „Mapp Safety“ – certifikovaný systém zajišťující funkční bezpečnost úrovně „SIL 3/PL e/CAT 4“.
- „Mapp Services“ – nabízí balíčky zjednodušující práci s alarmovým, recepturním a souborovým systémem. Výhodné funkce přináší také pro správu událostí a uživatelů (Mapp Technology, 2018).

2.6 „REACTION“

Pomocí technologie Reaction je možné snížit reakční dobu v aplikacích průmyslové automatizace až na 1 μ s, a to za použití standardního hardwaru. Časově náročné procesy jsou spouštěny přímo ve vstupně/výstupních modulech, což vede k jejich rychlému vyřízení. Výsledkem je vysoký nárůst výkonu bez zvýšení nákladů. Využití této technologie je možné při použití speciálních funkčních bloků ve vývojovém prostředí Automation Studio (ReACTION Technology, 2018).

3 AUTOMATION STUDIO

Automation Studio je integrované vývojové prostředí vyvinuté pro práci s produkty společnosti B&R. Jde o vývojové prostředí, které obsahuje nástroje pro konfiguraci všech fází projektu – od konfigurace řídicí jednotky, pohonu, komunikace až po vizualizaci (Automation Studio, 2018).

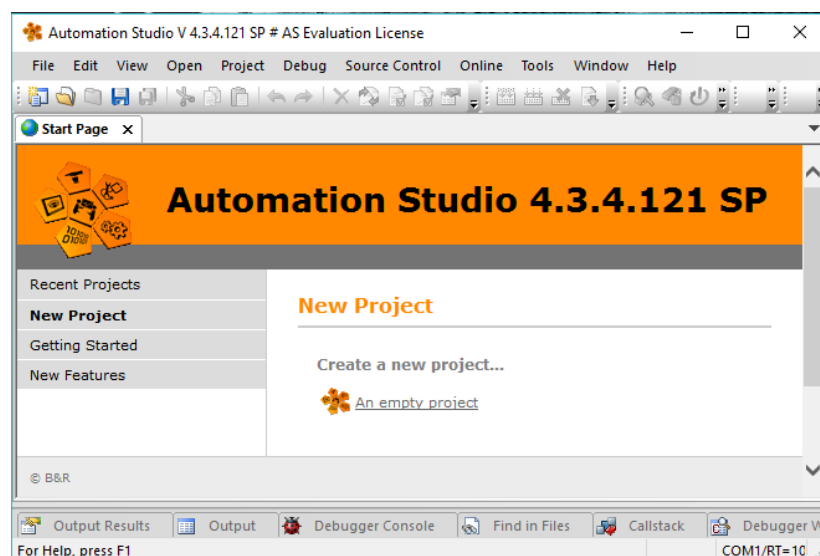
3.1 „AUTOMATION RUNTIME“

Nedílnou součástí prostředí Automation Studio je deterministický operační systém reálného času Automation Runtime, který tvoří softwarovou platformu všem produktům společnosti B&R. Operační systém zajišťuje chod programů a poskytuje možnost konfigurace cílových systémů. Další důležitou funkcí je možnost využití osmi různých cyklických tříd a spouštění aplikací nezávisle na hardwaru (Automation Runtime, 2018).

3.2 ÚVODNÍ STRANA

Po instalaci a následném spuštění softwaru Automation Studio se zobrazí úvodní strana, která je znázorněna na obr. 3.1. Z úvodní strany lze vytvořit nový projekt, nebo spustit a dále editovat již existující. Postup pro vytvoření projektu je popsán v 7. kapitole.

Vybráním položky „Getting Started“ se zobrazí nabídka před vypracovaných praktických úloh, které lze použít jako demonstrativní a výukové programy. Úlohy byly naprogramovány programátory společnosti B&R, a některé jejich části lze implementovat do budoucích projektů.

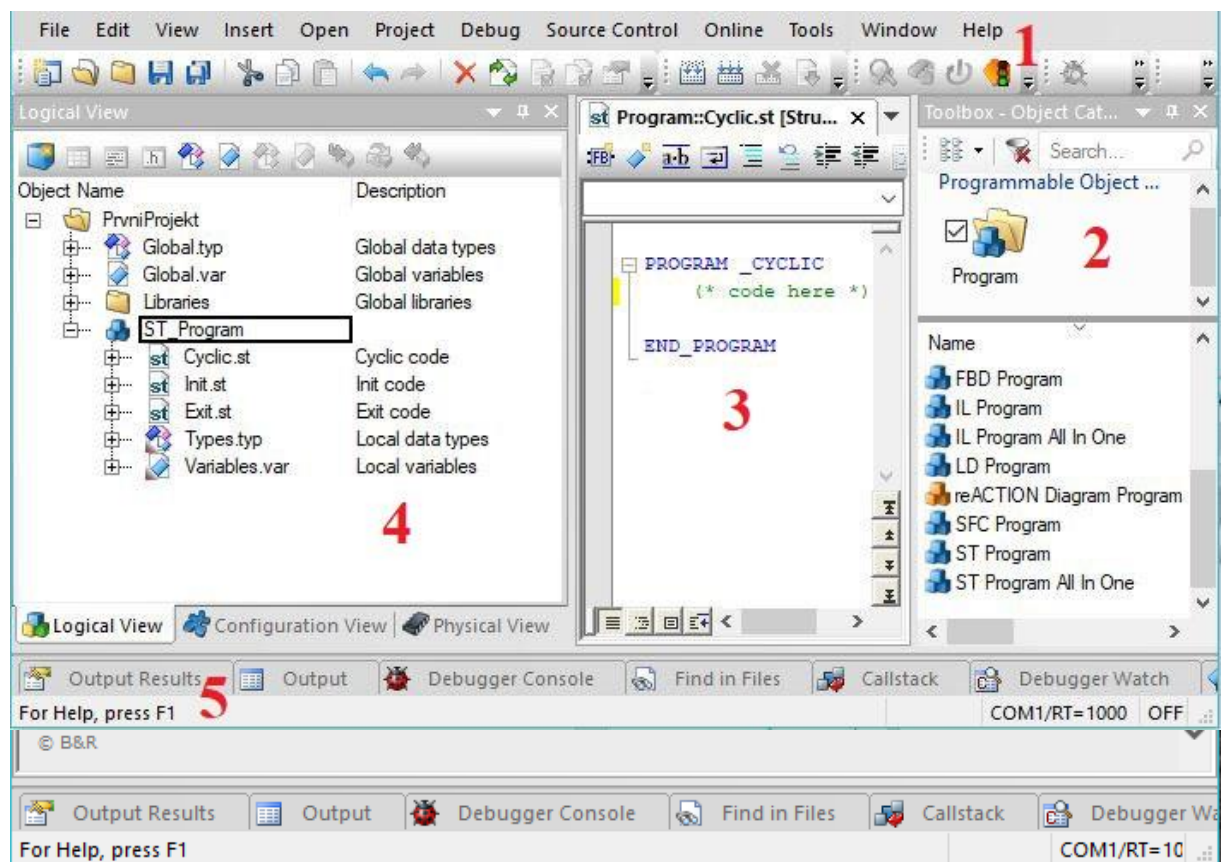


Obr. 3.1 – Úvodní strana softwaru Automation Studio

3.3 PRACOVNÍ PROSTŘEDÍ

Pracovní prostředí softwaru Automation Studio, znázorněné na obr. 3.2, je rozděleno do několika oblastí. Každá z těchto oblastí plní specifické funkce:

1. V horní části jsou umístěny záložky, které společně s pracovní lištou umožňují rychlý přístup k základním ale i rozšiřujícím funkcím programu.
2. Objektový katalog (Toolbox – Object Catalog) umožňuje snadnou implementaci nástrojů, funkcí, objektů a hardwarových modulů. Zobrazovaná nabídka vždy odpovídá objektu, ve kterém se aktuálně pracuje.
3. Uprostřed se nachází hlavní okno, které slouží ke konfiguraci dat projektu a psaní programu.
4. V levé části se nachází organizátor projektu, který je rozdělen do tří specifických záložek – Logical View, Configuration View a Physical View. Pomocí těchto záložek lze spravovat a konfigurovat důležité části projektu.
5. V dolní části pracovního prostředí je umístěno výstupní okno (Output Results), ve kterém je zobrazován průběh kompilace projektu.



Obr. 3.2 – Pracovní prostředí softwaru Automation Studio

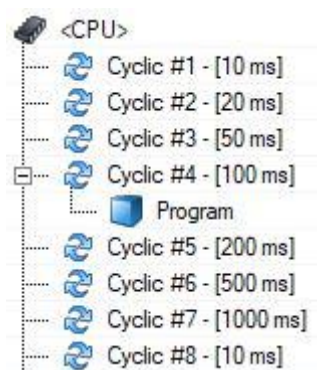
3.4 PROGRAM

Záložka Logical View, jejíž umístění je znázorněno na obr. 3.2, slouží ke správě programové části projektu a je nezávislá na hardwaru. Každý program je rozdělen podle způsobu vykonávání:

- „Cyclic.st“ – kód se provede v každém cyklu.
- „Init.st“ – kód se vykoná jen jednou, a to po spuštění PLC (inicializační podprogram).
- „Exit.st“ – kód se provede pouze na konci (výstupní podprogram).

V každém projektu existuje možnost práce s globálními proměnnými (Global.var), které je možné použít pro celý projekt a všechny jeho programy. Jednotlivé programy poté obsahují lokální proměnné (Variables.var), které jsou deklarovány a používány jen konkrétním programem. Podobně tomu je i při práci s datovými typy, které lze také rozdělit na globální (Global.typ) a lokální (Types.typ). Dále se v projektu nachází knihovny (Libraries), které jsou sbírkou opakovaně použitelných funkcí. Umožňují oddělit zdrojový kód specifický pro určitou funkci od skutečného programového kódu.

Záložka „Configuration View“ umožňuje softwarovou správu použitého hardwaru. V této záložce lze vytvořeným programům přiřadit jednu z osmi cyklických tříd, které se liší rychlostí vykonání cyklu. Zároveň je nutné si uvědomit, že rychlejší cyklické třídy jsou náročnější na procesor. Cyklické třídy včetně přiřazeného programu jsou znázorněny na obr. 3.3.



Obr. 3.3 – Cyklické třídy

3.5 SIMULACE

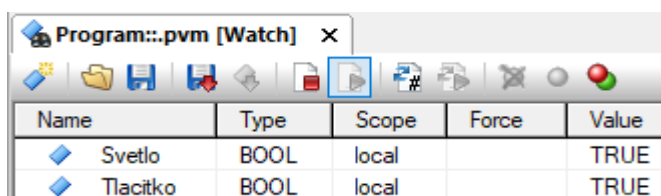
Software hraje v celkovém vývoji stroje nebo systému zásadní roli. Proto je důležité ověřovat funkčnost kódu ještě před uvedením do provozu. Simulátor slouží ke snadnému testování konfigurace projektu, a to bez nutnosti připojení k reálnému PLC procesoru. Praktické použití simulátoru je demonstrováno v kapitole 7.

3.6 DIAGNOSTICKÉ NÁSTROJE

Simulaci lze monitorovat pomocí některého z diagnostických nástrojů. Diagnostické nástroje slouží k vyladění nedostatků programu a celkové optimalizaci projektu. Mezi základní diagnostické nástroje patří Watch, Trace a Správce diagnostických nástrojů.

3.6.1 „Watch“

Umožňuje cyklické sledování a konfiguraci hodnot proměnných. Díky tomu je možné otestovat reakci programu na změnu konkrétní hodnoty. Programátor si sám zvolí, jaké hodnoty chce sledovat. Mimo samotné hodnoty je zobrazován i typ a rozsah dané proměnné. Diagnostický nástroj Watch je znázorněn na obr. 3.4. a jeho použití popsáno v kapitole 7.

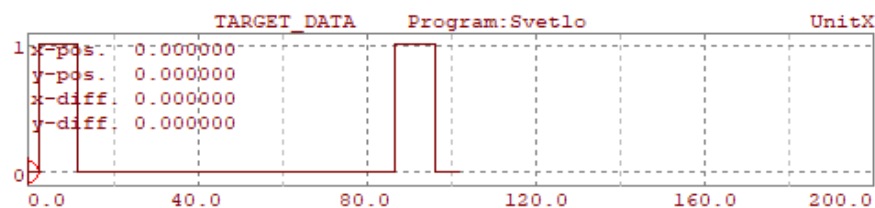


Name	Type	Scope	Force	Value
♦ Svetlo	BOOL	local		TRUE
♦ Tlacitko	BOOL	local		TRUE

Obr. 3.4 – Diagnostický nástroj Watch

3.6.2 „Trace“

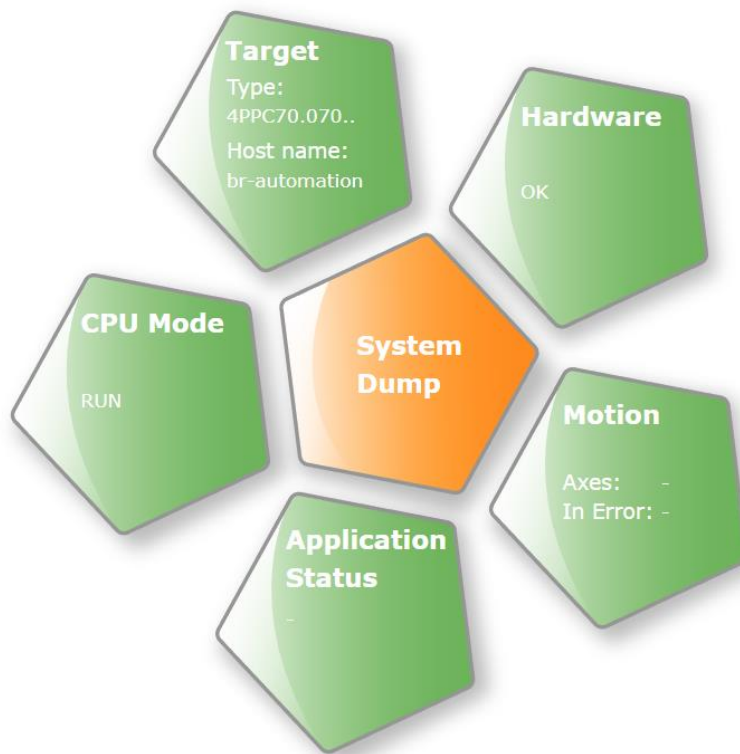
Diagnostický nástroj Trace, uvedený na obr. 3.5, zaznamenává hodnoty sledovaných proměnných v reálném čase po určitou dobu. Sledovaná data jsou graficky interpretována ve formě křivky.



Obr. 3.5 – Diagnostický nástroj Trace

3.6.3 Správce diagnostických nástrojů

Správce diagnostických nástrojů zajišťuje přístup informacím o systémovém hardwaru a softwaru odkudkoliv na světě. Jediným požadavkem je běžný webový prohlížeč, pomocí kterého lze snadno a rychle zobrazit aktuální stav systému. Potřebné diagnostické funkce jsou již implementovány. Vzhled systému ve webovém prostředí může být individuálně přizpůsoben požadavkům uživatele. Podoba rozhraní ve výchozím nastavení je znázorněna na obr. 3.6 (Advanced system diagnostics via the web, 2011).

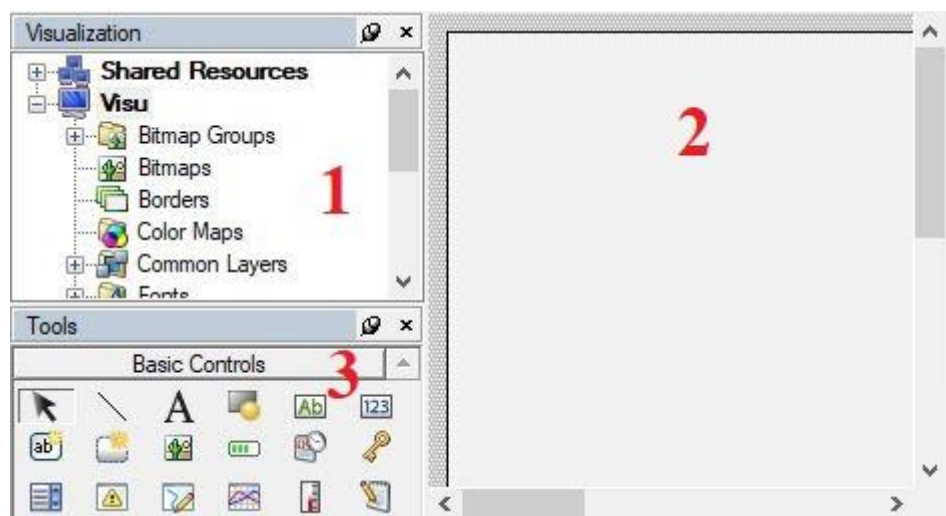


Obr. 3.6 – Správce diagnostických nástrojů

3.7 VIZUALIZACE A VIZUALIZAČNÍ PROSTŘEDÍ

Vizualizace slouží k přehlednému zobrazení mnohdy složitých dějů technické automatizace. Jde o grafické zobrazení procesu, které se snaží centralizovat ovládání na vizualizační obrazovku. Společnost B&R nabízí několik druhů zobrazovacích panelů, které mohou být ovládány pomocí dotykového displeje, funkčních kláves nebo kombinací obou možností (HMI, 2018).

V softwaru Automation Studio je integrované vizualizační prostředí Visual Components (VC). To umožňuje tvorbu a spouštění vizualizace společně s hlavní částí projektu. Pracovní prostředí pro tvorbu vizualizace je popsáno pomocí obr. 3.7.



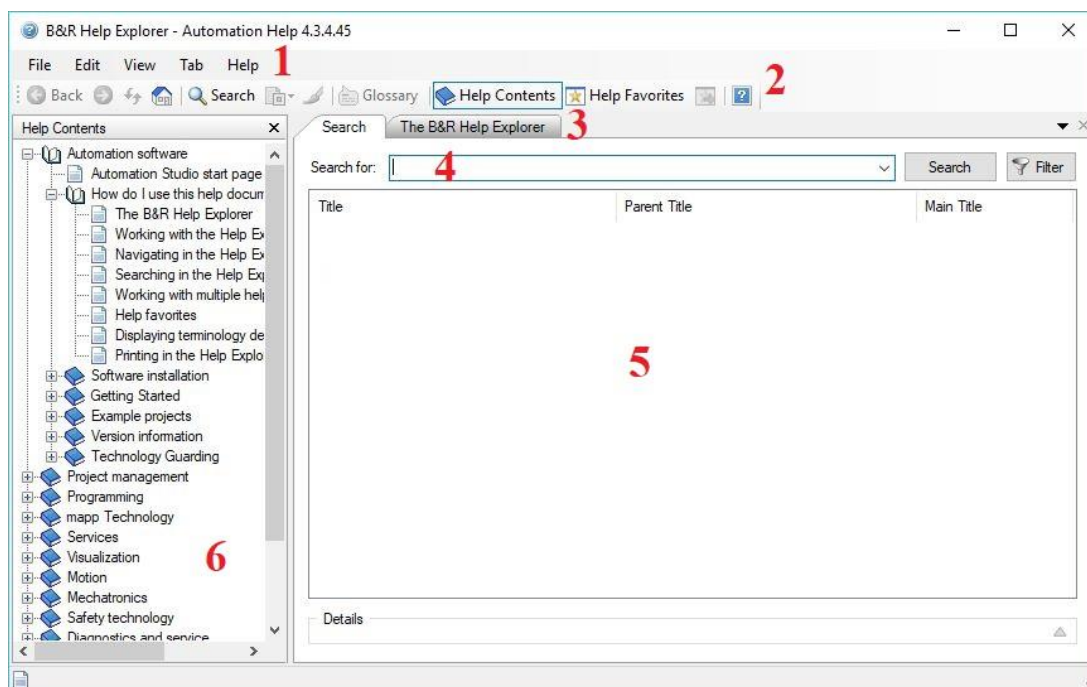
Obr. 3.7 – Vizualizační prostředí

1. Ve stromovém adresáři jsou umístěny veškeré komponenty spojené s vizualizací. Adresář je rozdělen do dvou oblastí. V první z nich se nachází globální komponenty (Shared Resources), které jsou společné pro všechny spravované vizualizace. Další oblast tvoří konkrétní tvořené vizualizace (Visu), které obsahují své lokální komponenty.
2. Pracovní prostředí slouží ke správě a editaci jednotlivých komponent. Po dvojitém kliknutí na komponentu se zde spustí příslušný editor.
3. Soubor nástrojů obsahující všechny potřebné grafické komponenty, které lze přesunutím implementovat do tvořené vizualizace. Obsahuje nástroje typu tlačítko, tvar, text, graf apod.

3.8 NÁPOVĚDA

Nápoověda (tzv. „help“) je neocenitelným zdrojem informací v průběhu vývoje, konfigurace a uvádění projektu do provozu. Slouží jako příručka pro použití vývojového prostředí Automation Studio. Poskytuje všechny informace potřebné pro tvorbu programů a vizualizací. Dále nabízí přístup k veškeré dokumentaci produktů společnosti B&R. Při instalaci nových hardwarových a softwarových aktualizací je i systém nápovědy rozšířen tak, aby zahrnoval potřebnou dokumentaci. Rozhraní nápovědy lze spustit stisknutím klávesy F1. Objeví se okno, jehož pracovní prostředí je popsáno prostřednictvím obrázku obr. 3.8.

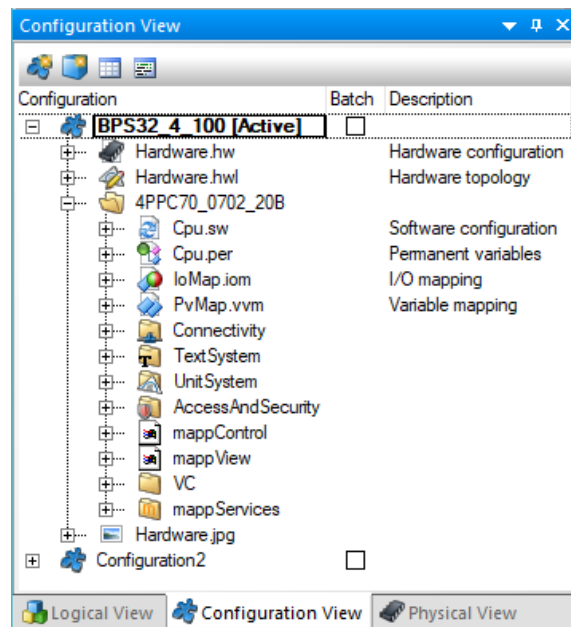
1. Horní lišta poskytuje přístup ke všem dostupným funkcím.
2. Panel nástrojů obsahuje ikony, které poskytují snadné použití nejčastěji užívaných funkcí.
3. Systém záložek umožňuje přehlednější práci s vyhledanými dokumenty.
4. Vyhledávací okno slouží k vyhledání konkrétního termínu, který musí obsahovat více než dva znaky.
5. Výsledky hledání jsou zobrazeny v pracovním prostoru.
6. Navigační strom je průběžně aktualizován tak, aby jeho obsah odpovídal právě zobrazené stránce.



Obr. 3.8 – Pracovní prostředí nápovědy

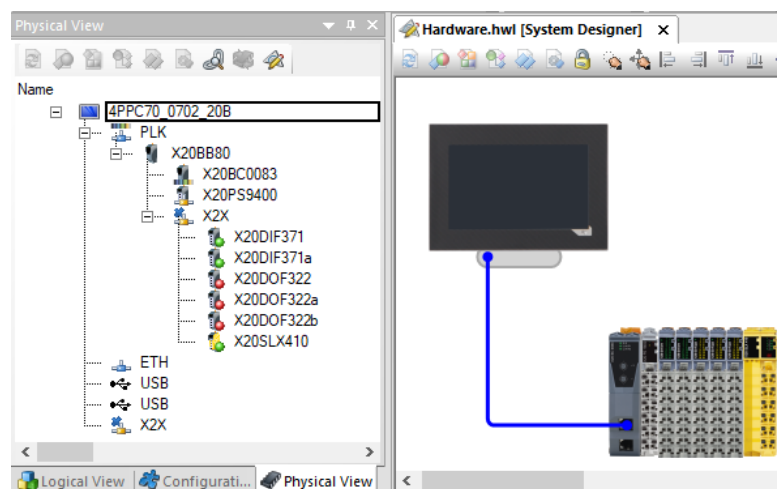
3.9 KONFIGURACE REÁLNÉHO ZAŘÍZENÍ

Záložka Configuration View, zobrazená na obr. 3.9, slouží ke správě hardwarové a softwarové konfigurace projektu v závislosti na použitém hardwaru. Každý projekt může obsahovat více takových konfigurací, nicméně pouze jedna může být aktivní (Active).



Obr. 3.9 – Configuration View

V záložce Physical View se nachází hardware právě aktivní konfigurace. Pomocí této záložky lze přidávat a konfigurovat nové hardwarové moduly, mapovat vstupy a výstupy a konfigurovat rozhraní sítě. Použitá zařízení jsou zobrazena pomocí stromového adresáře. Další možností zobrazení je vizuální interpretace, kterou lze aktivovat vybráním tlačítka System Designer. Obě možná zobrazení jsou znázorněna na obr. 3.10.



Obr. 3.10 – Physical View

4 LABORATORNÍ MODEL

Již existuje široká řada zhotovených laboratorních modelů, které demonstrují práci s programovatelnými logickými automaty. Příkladem mohou být produkty od společnosti Edumat, které slouží k výuce logických systémů realizovaných pomocí PLC. Mezi nabízený sortiment patří výukový model křižovatky, pračky, nápojového automatu a další. Jako ukázka poslouží model křižovatky, který je zobrazen na obr. 4.1 (Kohout, 2008).



Obr. 4.1 – Model křižovatky od společnosti Edumat (Kohout, 2008)

Další laboratorní modely pro automatizaci nabízí společnost Stroza s.r.o. Jako příklad lze uvést stavebnicový model dopravníkového systému, který je uveden na obrázku 4.2. Soustava slouží ke třídění válečků dle výšky (Výukový model mechatroniky, 2000).



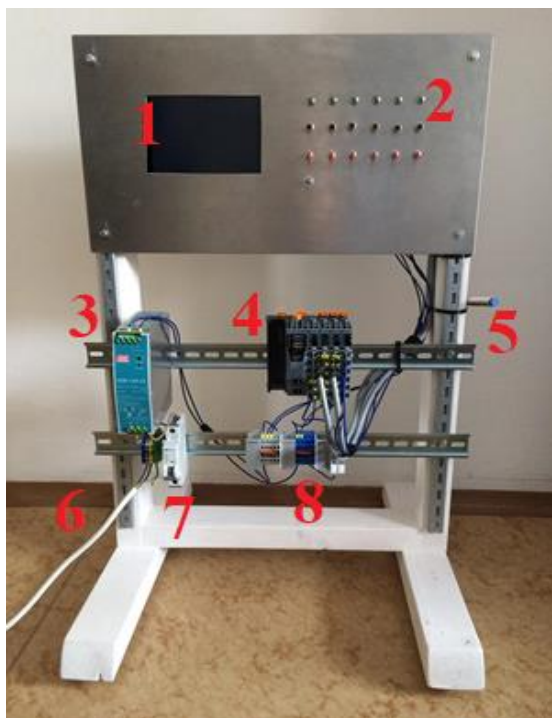
Obr. 4.2 – Model dopravníkového pásu od společnosti Stroza s.r.o. (Výukový model mechatroniky, 2000)

Ohledně výroby podobných laboratorních modelů bylo také zpracováno nepřehledné množství bakalářských prací, které mohou být inspirací pro vlastní tvorbu. I v rámci této práce byl sestaven laboratorní model, jehož konstrukcí a principem se bude tato kapitola zabývat.

4.1 ZKONSTRUOVANÝ LABORATORNÍ MODEL

Zkonstruovaná soustava, znázorněná na obr 4.3, slouží jako učební pomůcka k praktickému testování navržených aplikací včetně vizualizace. Konstrukci tvoří dřevěný rám, na který jsou osazeny nosné lišty s použitým zařízením. Řízené a řídicí prvky modelu jsou nainstalovány do plechovém panelu.

V celkovém rozsahu je laboratorní model vybaven 6 digitálními výstupy, jejichž stav je signalizován pomocí kontrolky. Dále 12 digitálními vstupy, které jsou ovládány tlačítky, spínači a indukčním čidlem. Součástí soustavy je také 1 analogový vstup s potenciometrem. Příklady laboratorních úloh, které na zkonstruované soustavě lze realizovat, jsou uvedeny v kapitole 5.



Obr. 4.3 – Laboratorní model

1. Zobrazovací panel s dotykovým displejem,
2. kontrolky, tlačítka, spínače a potenciometr,
3. zdroj stejnosměrného napětí $U = 24 \text{ V}$,
4. programovatelný logický automat,
5. indukční čidlo s detekční vzdáleností 0–4 mm,
6. napájecí síťový kabel,
7. jistič,
8. svorkovnice a pojistky.

4.2 POUŽITÁ ZAŘÍZENÍ

Pro realizaci laboratorního modelu byla použita zařízení od společnosti B&R. Základním kamenem celé soustavy je kompaktní PLC typu X20CP0482 a k němu připojené rozšiřující moduly: napájecí modul X20PS9600, vstupní digitální modul X20DI9371, výstupní digitální modul X20DO6322 a vstupní analogový modul X20AT2222. Vizualizace laboratorních úloh jsou zobrazovány na panelu 6PPT30.043F-20W. Napájení komponent zajišťuje zdroj stejnosměrného napětí 24 V od společnosti Mean Well.

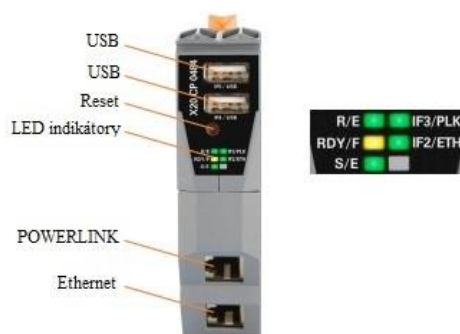
Součástí soustavy jsou i prvky zajišťující bezpečnost napájení použitých zařízení. Mezi tyto prvky patří jistič, který slouží jako ochrana zdroje stejnosměrného napětí 24 V před zkratem. Pro ochranu PLC a zobrazovacího panelu jsou použity pojistky o velikosti 0,5 A.

V pododdílech 4.2.1 – 4.2.5 jsou stručně představeny jednotlivé komponenty PLC. Součástí každého modulu jsou LED indikátory, pomocí kterých lze snadno identifikovat typ stavového hlášení. Z důvodu snadnější práce s modelem a pohotovějšího jednání v případě chybového hlášení jsou v popisu jednotlivých komponent uvedeny tabulky popisující stavy LED signalizací.

Podrobné informace o produktech lze dohledat v příslušných dokumentacích, které jsou volně dostupné na stránkách společnosti B&R Automation (<https://www.br-automation.com/cs/perfection-in-automation/>) nebo v nápovědě vývojového prostředí. Pro vyhledání potřebné dokumentace stačí do vyhledávače zadat název odpovídajícího modulu.

4.2.1 PLC X20CP0482

Hlavní modul, znázorněný na obr. 4.4, je v základním provedení osazen do tzv. Bus Base modulu X20BB67. Součástí tohoto celku jsou komunikační porty typu Powerlink, Ethernet a USB. Na modulu se dále nachází tlačítko Reset, jehož stisknutím je spuštěn hardwarový restart zařízení. V takovém případě jsou všechny aplikační programy zastaveny a veškeré výstupy nastaveny na nulu. Stavy LED signalizace jsou uvedeny v tab. 4.1 (X20CP041x and X20CP048x, 2018).



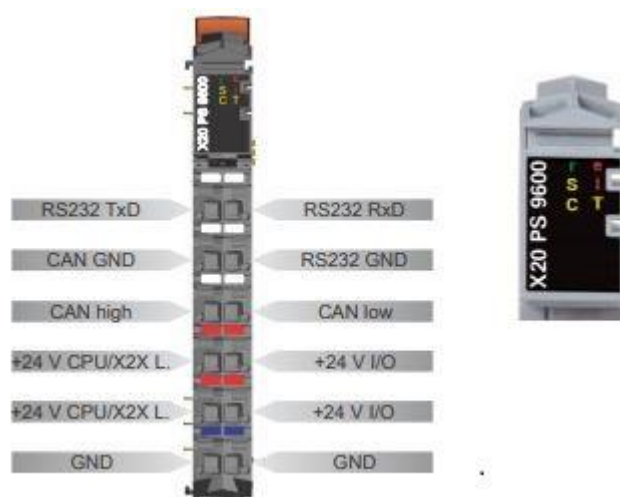
Obr. 4.4 – Modul X20CP0482 s LED indikátory (X20CP041x and X20CP048x, 2018)

Tab. 4.1 – Popis LED indikace modulu X20CP0482 (X20CP041x and X20CP048x, 2018)

LED	Barva	Stav	Popis
R/E	Zelená	Svíí	Aplikace je spuštěna a běží.
		Bliká	Probíhá inicializace aplikace.
	Červená	Svíí	PLC se nachází v módu SERVICE.
		Bliká	Došlo k porušení licence.
		Dvojité blikání	PLC se nachází v módu BOOT.
RDY/F	Žlutá	Svíí	PLC se nachází v módu SERVICE/BOOT.
		Bliká	Došlo k porušení licence.
S/E	Zelená/červená		Signalizace v závislosti na pracovním režimu.
IF3/PLK	Zelená	Svíí	POWERLINK je připojen.
		Bliká	Probíhá činnost na Powerlink sběrnici.
IF2/ETH	Zelená	Svíí	Ethernet je připojen.
		Bliká	Probíhá činnost na Ethernet sběrnici.

4.2.2 Napájecí modul X20PS9600

Nedílnou součástí základní konfigurace je napájecí modul X20PS9600. Modul zajišťuje napájení procesoru, komunikačního portu RS232, X2X linky, CAN sběrnice a interních vstupů a výstupů. Rozmístění prvků na svorkovnici modulu je zobrazeno na obr. 4.5. Stav signalizace LED indikátorů, detailně zobrazené na obr. 4.5, jsou popsány v tab. 4.2 (X20PS9600, 2018).



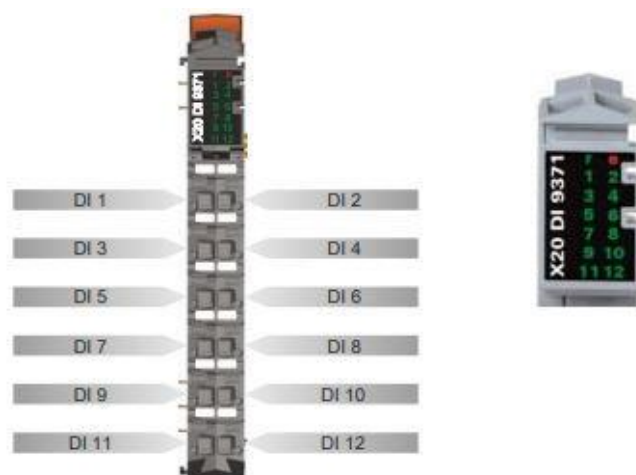
Obr. 4.5 – Modul X20PS9600 s LED indikátory (X20PS9600, 2018)

Tab. 4.2 – Popis LED indikace modulu X20PS9600 (X20PS9600, 2018)

LED	Barva	Stav	Popis
r	Zelená	Zhasnutá	Není připojeno napájení modulu.
		Jednotné bliknutí	PLC se nachází v módu RESET.
		Bliká	Probíhá inicializace.
		Svíí	Modul je aktivní.
e	Červená	Zhasnutá	Modul není připojen / Vše je v pořádku.
		Dvojité blikání	Problém s napájením modulu.
e + r	Červená a blikající zelená		Firmware je neplatný.
I	Červená	Zhasnutá	Napájení procesoru a X2X linky je v pořádku.
		Svíí	Napájení procesoru a X2X linky není v pořádku.
S	Žlutá	Zhasnutá	Rozhraní RS232 není aktivní.
		Svíí	Rozhraní RS232 je aktivní.
C	Žlutá	Zhasnutá	Sběrnice CAN není aktivní.
		Svíí	Sběrnice CAN je aktivní.
T	Žlutá	Zhasnutá	Omezovací rezistor sběrnice CAN není aktivní.
		Svíí	Omezovací rezistor sběrnice CAN je aktivní.

4.2.3 Vstupní digitální modul X20DI9371

Vstupní modul je dalším důležitým prvkem hardwarové konfigurace. Modul poskytuje 12 digitálních vstupů, ke kterým jsou v rámci sestaveného modelu připojeny tlačítka (DI 1 – DI 6), spínače (DI 7 – DI 11) a indukční čidlo (DI 12). Na obr. 4.6 je znázorněno rozmístění digitálních vstupů. Součástí obrázku je i detailní záběr LED indikátorů, jejichž popis je uveden v tab. 4.3 (X20cDI9371, 2018).



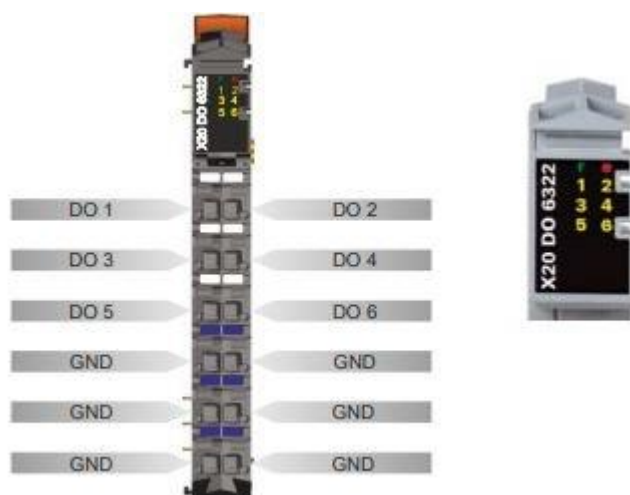
Obr. 4.6 – Modul X20DI9371 s LED indikátory
(X20cDI9371, 2018)

Tab. 4.3 – Popis LED indikace modulu X20DI9371 (X20cDI9371, 2018)

LED	Barva	Stav	Popis
r	Zelená	Zhasnutá	Není připojeno napájení modulu.
		Jednotné bliknutí	PLC se nachází v módu RESET.
		Bliká	Probíhá inicializace.
		Svíí	Modul je aktivní.
e	Červená	Zhasnutá	Modul není připojen. / Vše je v pořádku.
e + r	Červená a blikající zelená		Firmware je neplatný.
1–12	Zelená	Svíí	Odpovídající vstup je aktivní.

4.2.4 Výstupní digitální modul X20DO6322

Výstupní modul je vybaven 6 digitálními výstupy. Stav jednotlivých výstupů je signalizován pomocí připojených kontrolky (DO 1 – DO 6). Při realizaci byl zvolen způsob 1vodičového zapojení, tudíž je využito pouze prvních šesti svorek modulu. Pro implementaci 2vodičového zapojení by mohla být použita celá 12kolíková svorkovnice s využitím svorek GND (uzemnění). Rozvržení digitálních výstupů na svorkovnici včetně detailního záběru LED indikátorů popisuje obr. 4.7. Stavů LED indikací jsou uvedeny v tab. 4.4 (X20cDO6322, 2018).



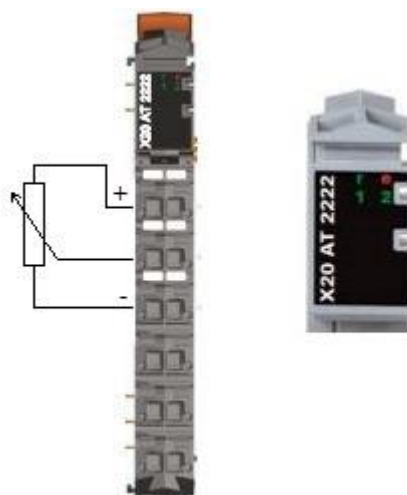
Obr. 4.7 – Modul X20DO6322 s LED indikátory (X20cDO6322, 2018)

Tab. 4.4 – Popis LED indikace modulu X20DO6322 (X20cDO6322, 2018)

LED	Barva	Stav	Popis
r	Zelená	Zhasnutá	Není připojeno napájení modulu.
		Jednotné bliknutí	PLC se nachází v módu RESET.
		Bliká	Probíhá inicializace.
		Svíí	Modul je aktivní.
e	Červená	Zhasnutá	Modul není připojen / Vše je v pořádku.
		Jednotné bliknutí	Chyba na I/O kanálu.
e + r	Červená a blikající zelená		Firmware je neplatný.
1–6	Oranžová		Odpovídající výstup je aktivní.

4.2.5 Vstupní analogový modul X20AT2222

Vstupní analogový modul poskytuje možnost práce s analogovou veličinou na základě přímého měření odporu. V rámci laboratorního modelu je na tento analogový vstup připojen potenciometr (5000 Ω), jehož zapojení je znázorněno na obr. 4.8. LED indikátory stavů analogového modulu jsou také znázorněny na obr. 4.8 a jejich popis je uveden v tab. 4.5 (X20AT2222, 2018).



Obr. 4.8 – Modul X20AT2222 s LED indikátory
(X20AT2222, 2018)

Tab. 4.5 – Popis LED indikace modulu X20AT2222 (X20AT2222, 2018)

LED	Barva	Stav	Popis
r	Zelená	Zhasnutá	Není připojeno napájení modulu.
		Jednotné bliknutí	PLC se nachází v módu RESET.
		Bliká	Probíhá inicializace.
		Svítlí	Modul je aktivní.
e	Červená	Zhasnutá	Modul není připojen / Vše je v pořádku.
		Svítlí	Chyba nebo stav RESET
		Jednotné bliknutí	Analogová hodnota je mimo rozsah.
e + r	Červená a blikající zelená	Firmware je neplatný.	
1–2	Zelená	Zhasnutá	Vstup je vypnutý.
		Bliká	Analogová hodnota vstupu je mimo rozsah.
		Svítlí	A/D převodník běží, hodnota je v pořádku.

4.3 VSTUPY A VÝSTUPY

Z důvodu přehledného mapování vstupů a výstupů byly zhotoveny tabulky, ve kterých je uvedeno připojení komponent na reálné vstupy a výstupy jednotlivých modulů programovatelného logického automatu. Popis v kolonce „Název“ odpovídá pojmenování daného vstupu či výstupu ve vývojovém prostředí Automation Studio. Pomocí tab. 4.6 jsou popsány digitální vstupy modulu X20DI9371. Tab. 4.7 charakterizuje digitální výstupy modulu X20DO6322 a popis v tab. 4.8 odpovídá zapojení vstupu na analogovém modulu X20AT2222.

Tab. 4.6 – Použité vstupy modulu X20DI9371

X20DI9371	
Název vstupu	Prvek
DigitalInput01	Tlačítko 1
DigitalInput02	Tlačítko 2
DigitalInput03	Tlačítko 3
DigitalInput04	Tlačítko 4
DigitalInput05	Tlačítko 5
DigitalInput06	Tlačítko 6
DigitalInput07	Spínač 1
DigitalInput08	Spínač 2
DigitalInput09	Spínač 3
DigitalInput10	Spínač 4
DigitalInput11	Spínač 5
DigitalInput12	Indukční čidlo

Tab. 4.7 – Použité výstupy modulu X20DO6322

X20DO6322	
Název	Prvek
DigitalOutput1	Kontrolka 1
DigitalOutput2	Kontrolka 2
DigitalOutput3	Kontrolka 3
DigitalOutput4	Kontrolka 4
DigitalOutput5	Kontrolka 5
DigitalOutput6	Kontrolka 6

Tab. 4.8 – Použitý vstup modulu X20AT2222

X20AT2222	
Název	Prvek
Resistor01	Potenciometr

5 ZADÁNÍ LABORATORNÍCH ÚLOH

V rámci bakalářské práce byla sestavena sada demonstračních úloh, která svým rozsahem odpovídá zhotovenému laboratornímu modelu. Model tedy poslouží k reálnému ověření funkčnosti vytvořených aplikací. V celkovém rozsahu byly vytvořeny a popsány čtyři laboratorní úlohy, pomocí kterých se uživatel seznámí s vývojovým prostředím Automation Studio. Vypracováním laboratorních úloh si uživatel vyzkouší použití základních programovacích funkcí, jako jsou časovače, čítače a komparátory. Součástí jedné z úloh je i tvorba vizualizace. Veškeré potřebné úkony pro vytvoření funkční aplikace jsou popsány v návodu, který se nachází v kapitole 6. Vzorová řešení jsou uvedena v kapitole 7.

5.1 ÚLOHA Č. 1

Pomocí první laboratorní úlohy je čtenář seznámen s vývojovým prostředím Automation Studio. Na úloze jsou demonstrovány důležité kroky konfigurace, které jsou potřebné pro realizaci jakéhokoliv zadání. Jednotlivé úkony včetně příkladu řešení této úlohy jsou uvedeny v kapitole 7. Realizaci úlohy si uživatel vyzkouší rozsvícení a zhasnutí kontrolky pomocí jednoho tlačítka.

K realizaci úlohy bude potřeba jedno tlačítko a šest kontrolky. Ve výchozím stavu budou kontrolky zhasnuty. Po stisknutí tlačítka (vstup) dojde k rozsvícení všech šesti kontrolky (výstup). Po opakovaném stisknutí tlačítka budou kontrolky zhasnuty.

5.2 ÚLOHA Č. 2

Cílem druhé laboratorní úlohy je rozblikání kontrolky s určitou blikající frekvencí. Vypracováním úlohy si uživatel ujasní nedostatky spojené s použitím časovačů a implementací logické funkce AND pro spuštění blikání pomocí spínačů. Zadání je upřesněno pomocí tab. 5.1.

Tab. 5.1 – Pravdivostní tabulka k úloze č. 2

1. spínač	2. spínač	Blikání kontrolky
0	0	0
0	1	0
1	0	0
1	1	1

K realizaci úlohy budou potřeba dva spínače a šest kontrolky. Ve výchozím stavu budou všechny kontrolky zhasnuty. Pokud dojde k sepnutí dvou spínačů zároveň, nastane rozblikání kontrolky s blikající frekvencí 1 vteřina. Bude-li některý ze spínačů přepnut zpátky do výchozí pozice, tak kontrolky blikat nebudou.

5.3 ÚLOHA Č. 3

Smyslem úlohy je počítání počtu zakrytí čidla, které bude signalizováno postupným rozsvěcováním kontrolky. Úloha demonstruje použití binárního čidla, kde uživatel simuluje jeho zakrytí. Vypracováním úlohy je objasněno použití čítačů a komparátorů.

K realizaci úlohy bude potřeba indukční čidlo, jedno tlačítko a šest kontrolky. Ve výchozím stavu nesvítí žádná kontrolka. Uživatel simuluje zakrytí čidla přiložením kovového předmětu (detekční vzdálenost čidla je 0–4 mm). Páté zakrytí bude signalizováno rozsvícením kontrolky 1, 2 a 3. Při desátém zakrytí se rozsvítí kontrolky 4, 5 a 6. Tudíž rozsvícením všech šesti kontrolky je signalizován počet deseti zakrytí.

Součástí úlohy je i naprogramování resetovacího tlačítka, které může být stisknuto kdykoliv v průběhu programu. Po jeho stisknutí dojde k vynulování čítače (rozsvícené kontrolky zhasnou) a zakrývání čidla se bude počítat znovu od nuly.

5.4 ÚLOHA Č. 4

Úloha 4 je úvodem do programování analogových veličin a tvorby vizualizace. Cílem je vytvoření programu, pomocí kterého bude řízena signalizace překročení limitů hladiny ve vodní nádrži. Součástí zadání je i tvorba vizualizace. Reálný model vodní nádrže není součástí sestaveného laboratorního modelu, nicméně na tomto příkladu lze zřetelně demonstrovat potřebné vizualizační konfigurace (model vodní nádrže je pouze vizualizován). Vyhotovením úlohy je uživatel seznámen se zpracováním analogových veličin, převodem datových typů, použitím matematických operací a tvorbou vizualizace.

K realizaci úlohy bude potřeba potenciometr a šest kontrolky. Pro ujasnění zadání si lze představit vodní nádrž, ve kterém je hlídán dolní a horní limit hladiny. Aktuální výška hladiny bude simulována pomocí potenciometru a uváděna v procentech (0 % až 100 %). Důležitým bodem je nastavení limitů hladiny (dolní a horní), jejichž překročení bude signalizováno na laboratorním modelu. Dolní limit bude nastaven na 10 % z maximální výšky hladiny a při jeho dosažení dojde k rozsvícení kontrolky 1, 2 a 3. Naopak při překročení horního limitu, který bude nastaven na 90 % z maximální výšky hladiny, se rozsvítí kontrolky 4, 5 a 6.

Vizualizace bude obsahovat model nádrže, ve kterém bude graficky zobrazována aktuální výška hladiny. V modelu budou graficky vyznačeny hlídané limity. Vedle modelu bude umístěna číselná hodnota hladiny, která bude prezentovat stav nádrže (z kolika procent je nádrž naplněna). Součástí vizualizace budou i chybová hlášení. Na překročení dolního a horního limitu bude obsluha upozorněna pomocí libovolné textové zprávy zobrazené v blízkosti modelu.

6 NÁVOD AUTOMATION STUDIO

V rámci bakalářské práce byl sestaven návod pro práci ve vývojovém prostředí Automation Studio 4.3, který popisuje vše potřebné pro zhotovení funkčního projektu.

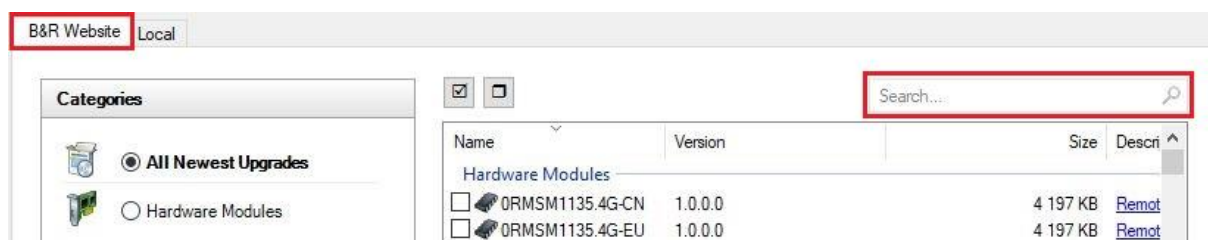
6.1 ZÍSKANÍ SOFTWARE

Z webových stránek společnosti B&R Automation lze po zadání emailu zdarma stáhnout 90denní zkušební verzi, kterou je možné po vypršení lhůty opakovaně obnovovat. I tato základní verze nabízí vše potřebné pro realizaci projektů. Další možností je požádání o poskytnutí studentské verze, která je oproti té základní zpřístupněna na dobu 400 dní. V rámci žádosti jde pouze o vyplnění základních informací, jako je jméno studenta a název školy. Existuje i možnost zakoupení plné verze softwaru, jehož pořizovací cena se pohybuje v řádech několika tisíců.

6.2 INSTALACE BALÍČKŮ KOMPONENT

Součástí laboratorního modelu jsou reálné hardwarové komponenty, které je zapotřebí definovat v softwaru Automation Studio. Může nastat situace, kdy tyto komponenty nebudou součástí katalogu. V takovém případě bude potřeba stáhnout a nainstalovat odpovídající balíčky. Pro stahování a instalaci balíčků slouží nástroj **Upgrades**, který lze spustit ze záložky **Tools**. Nástroj umožňuje online stahování přímo z webových stránek společnosti B&R, nebo ruční instalaci již stažených balíčků. Pro úspěšné vypracování laboratorních úloh je potřeba mít nainstalované tyto balíčky (název instalačního balíčku odpovídá názvu komponenty): X20CP0482, X20BB67, X20PS9600, X20DI9371, X20DO6322, X20AT2222 a v případě tvorby vizualizace balíček zobrazovacího panelu 6PPT30.043F-20W.

Při online instalaci (záložka **B&R Website**) je možné daný balíček vyhledat a následně nainstalovat pomocí vyhledávacího okna (obr. 6.1).



Obr. 6.1 – Online instalace balíčků

Pro ruční instalaci (záložka **Local**) je nutné si potřebné balíčky stáhnout pomocí webového prohlížeče ze stránek společnosti B&R. Následně pomocí okna **Local Directory** definovat cestu ke staženým balíčkům a spustit jejich instalaci (obr. 6.2).



Obr. 6.2 – Ruční instalace balíčků

6.3 ZALOŽENÍ PROJEKTU

Po spuštění vývojového prostředí Automation Studio se zobrazí úvodní stránka, ze které lze zahájit tvorbu nového projektu zvolením možnosti **New Project** (obr. 6.3) a vybráním položky **An empty project** (obr. 6.4).



Obr. 6.3 – Volba možnosti New Project



Obr. 6.4 – Položka An empty project

Zobrazí se okno pro základní nastavení projektu (obr. 6.5), ve kterém je nutné zadat název projektu (**Name of the project**). Dále je možné editovat cestu uložení, zkopírovat Automation Runtime a přidat popis projektu.

Name of the project:

Path of the project:
E:\BrAutomation\Projects\

Note: A subfolder with the same name as the project will be created automatically.

Copy Automation Runtime files into project

Description of the project:

Obr. 6.5 – Základní nastavení projektu

Dalším bodem je založení nové hardwarové konfigurace (obr. 6.6). V tomto okně je nutné vyplnit název konfigurace (**Name of the configuration**) a zaškrtnout možnost ruční konfigurace (**Define a new hardware configuration manually**). Dále existuje volba mezi online konfigurací nebo použitím již existující konfigurace. Ve spodní části lze přidat popis tvořené konfigurace.

Name of the configuration:
Config1

Hardware Configuration

Define a new hardware configuration manually

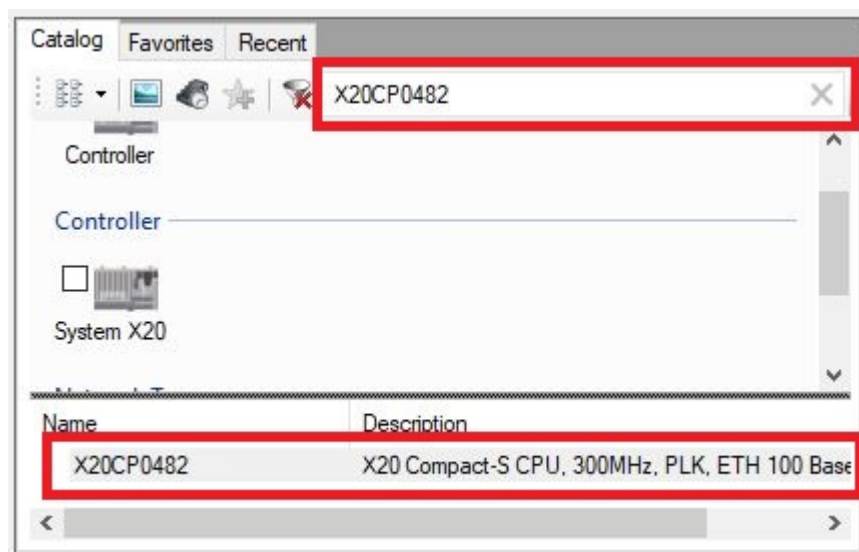
Identify hardware configuration online

Reference an existing hardware configuration (*.hw).

Description of the configuration:

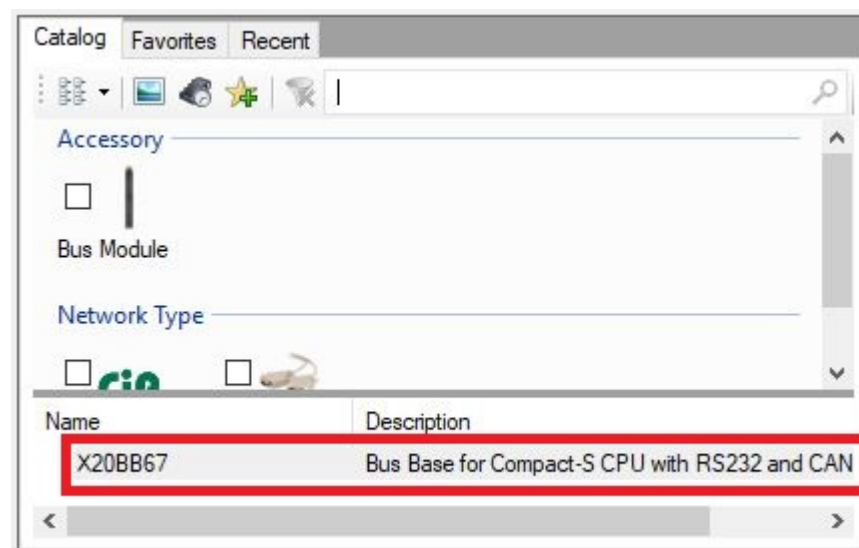
Obr. 6.6 – Založení nové hardwarové konfigurace

Následuje přidání hlavního modulu programovatelného logického automatu z katalogu produktů. Daný model je možné vyhledat pomocí vyhledávacího okna. Pro realizaci úloh na zhotoveném laboratorním modelu je nutné přidat modul typu **X20CP0482** (obr. 6.7).



Obr. 6.7 – Přidání hlavního modulu do HW konfigurace

Dále je potřeba přiřadit tzv. Bus Base typu **X20BB67** (obr. 6.8), který je nedílnou součástí používaného PLC. Pro dokončení tvorby nového projektu stisknout tlačítko **Finish**.

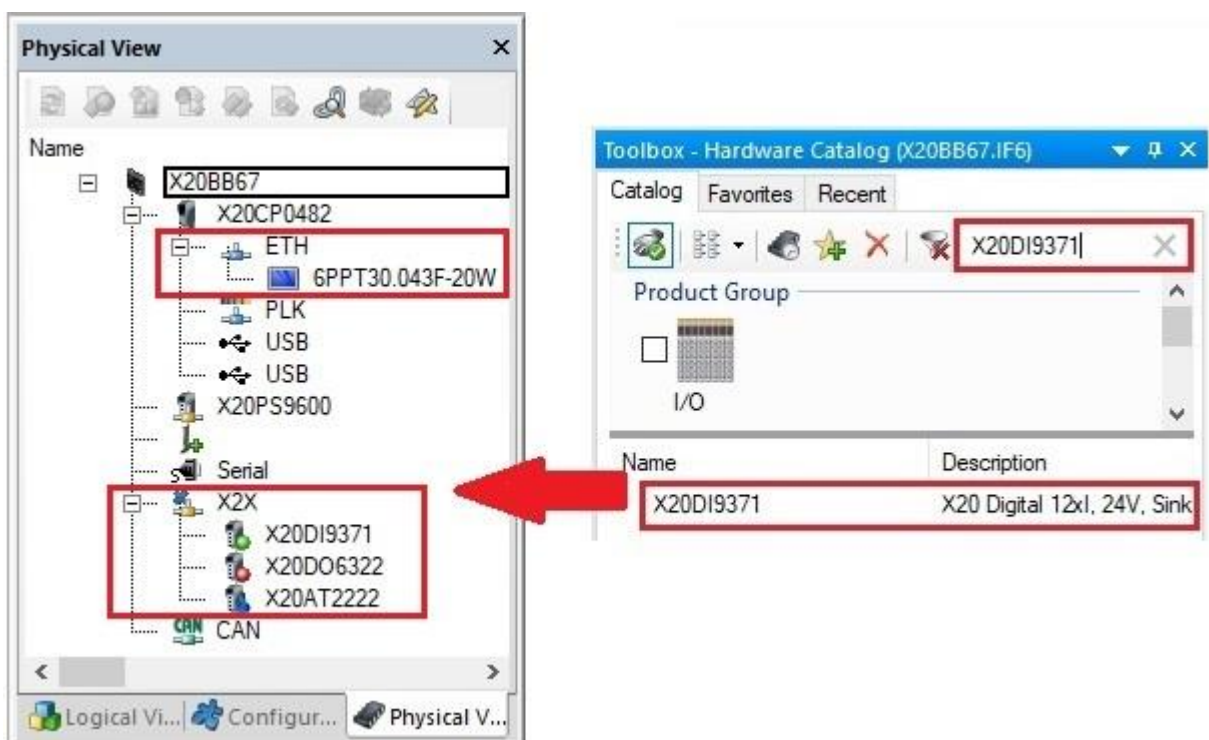


Obr. 6.8 – Přidání Bus Base modulu do HW konfigurace

6.4 PŘIDÁNÍ HARDWAROVÉHO MODULU

Sestavený laboratorní model obsahuje hlavní modul s procesorem typu X20CP0482 a tzv. Bus Base X20BB67. Tyto základní moduly byly přidány již při zakládání projektu. Nyní je potřeba hardwarovou konfiguraci rozšířit o další použité komponenty: vstupní digitální modul **X20DI9371**, výstupní digitální modul **X20DO6322**, vstupní analogový modul **X20AT2222** a zobrazovací panel **6PPT30.043F-20W**. Napájecí modul X20PS9600 je k HW konfiguraci přiřazen automaticky.

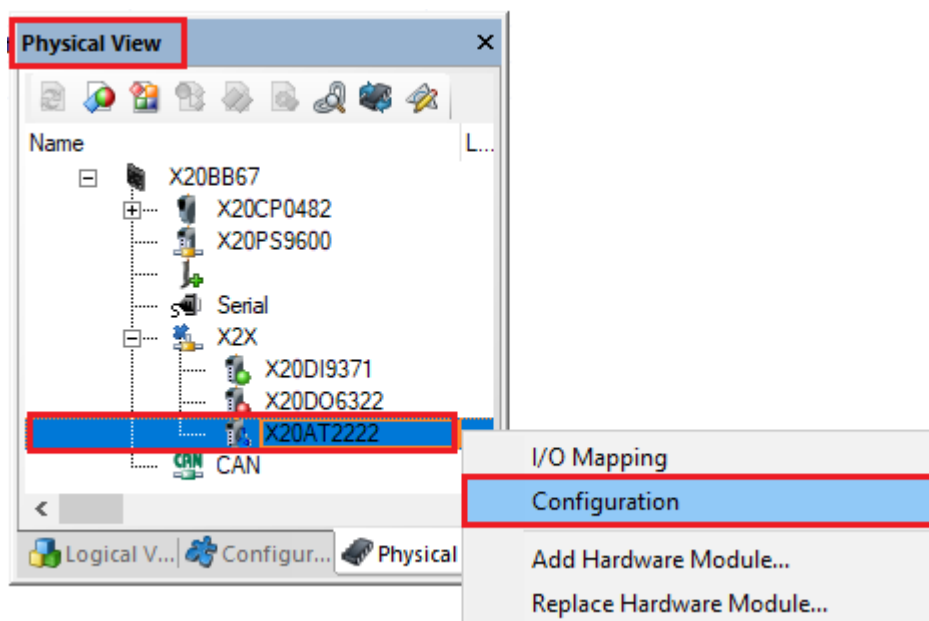
Pro přidávání a odebírání hardwarových prvků slouží záložka **Physical View**. Označením položky **X2X** se v **Toolbox – Hardware Catalog** (ve vývojovém prostředí napravo) zobrazí výběr všech dostupných komponent. Potřebné komponenty lze ručně dohledat podle nabízených možností. Dalším způsobem je vyhledávání pomocí vyhledávacího okna. Nalezenou komponentu je zapotřebí dvojklikem nebo přetažením implementovat pod položku **X2X** (ve vývojovém prostředí nalevo). Výjimkou je zobrazovací panel (6PPT30.043F-20W), který musí být přiřazen pod položku **ETH**. Na obr. 6.9 je znázorněn postup pro přidání potřebného modulu.



Obr. 6.9 – Přidání HW modulu

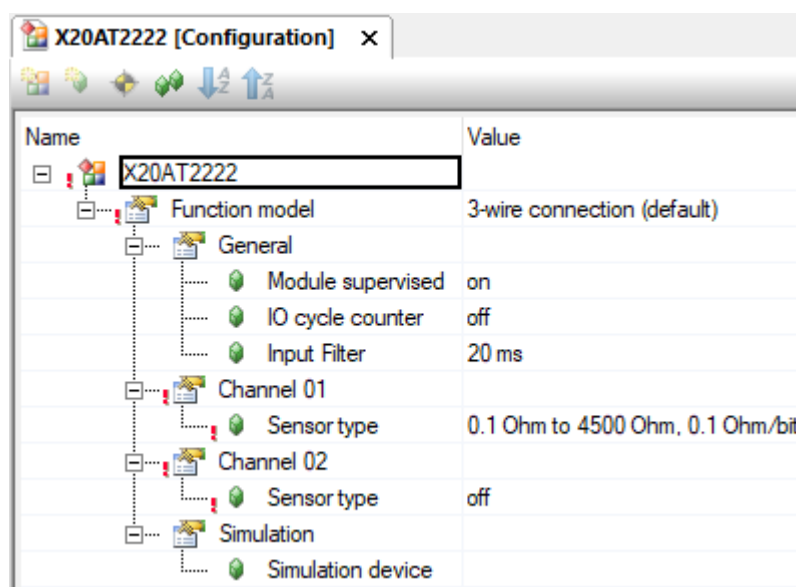
6.5 KONFIGURACE ANALOGOVÉHO MODULU

Vstupní analogový modul X20AT2222 jako jediný z modulů vyžaduje zvláštní konfiguraci (v rámci laboratorního modelu). Spuštění konfigurace ze záložky **Physical View** je znázorněno na obr. 6.10.



Obr. 6.10 – Spuštění konfigurace modulu X20AT2222

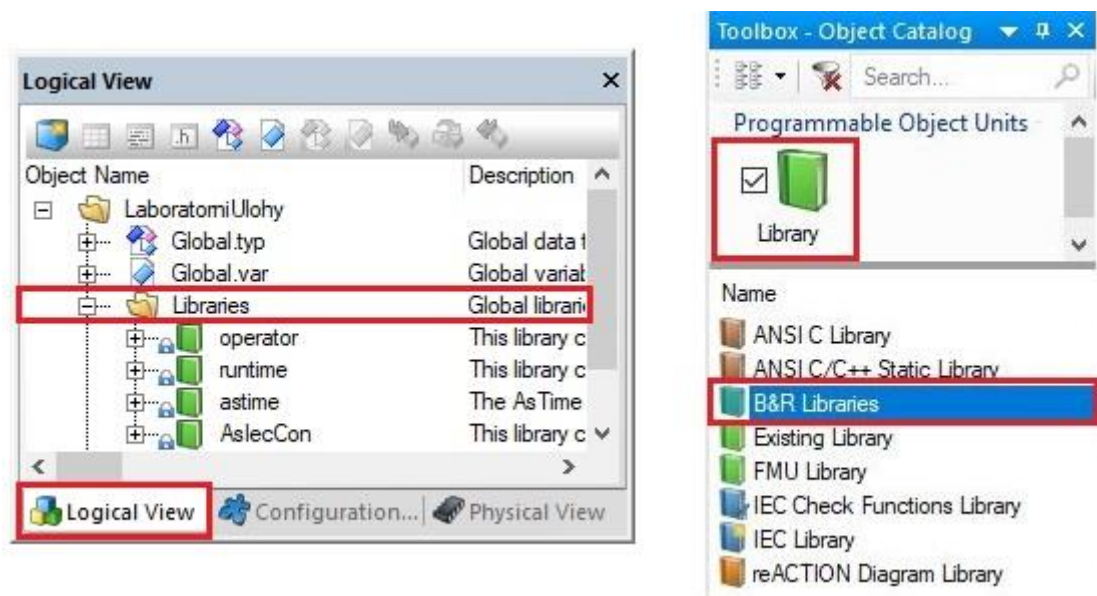
Objeví se konfigurační okno, jehož parametry je nutné nastavit podle obr. 6.11.



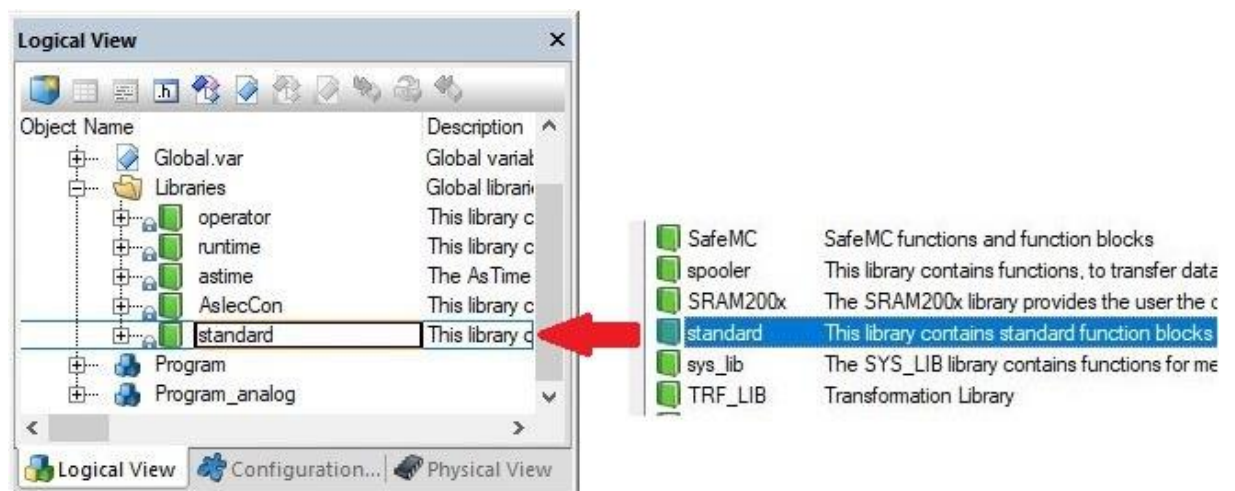
Obr. 6.11 – Konfigurace modulu X20AT2222

6.6 PŘIDÁNÍ KNIHOVNY

Ještě před začátkem tvorby samotného programu je vhodné do projektu přidat knihovnu **standard**, která obsahuje základní funkční bloky. Přidání knihovny lze realizovat ze záložky **Logical View** (ve vývojovém prostředí nalevo). Označením položky **Libraries** se v **Toolbox – Object Catalog** (ve vývojovém prostředí napravo) zobrazí možnosti pro práci s knihovnami – výběr lze zjednodušit zaškrtnutím složky **Libraries**. Ze zobrazeného výběru je potřeba otevřít položku **B&R Libraries** a v ní vyhledat knihovnu **standard**. Dvojitým kliknutím se vybraná knihovna implementuje do projektu. Postup pro přidání knihovny je znázorněn na obr. 6.12 a obr. 6.13.



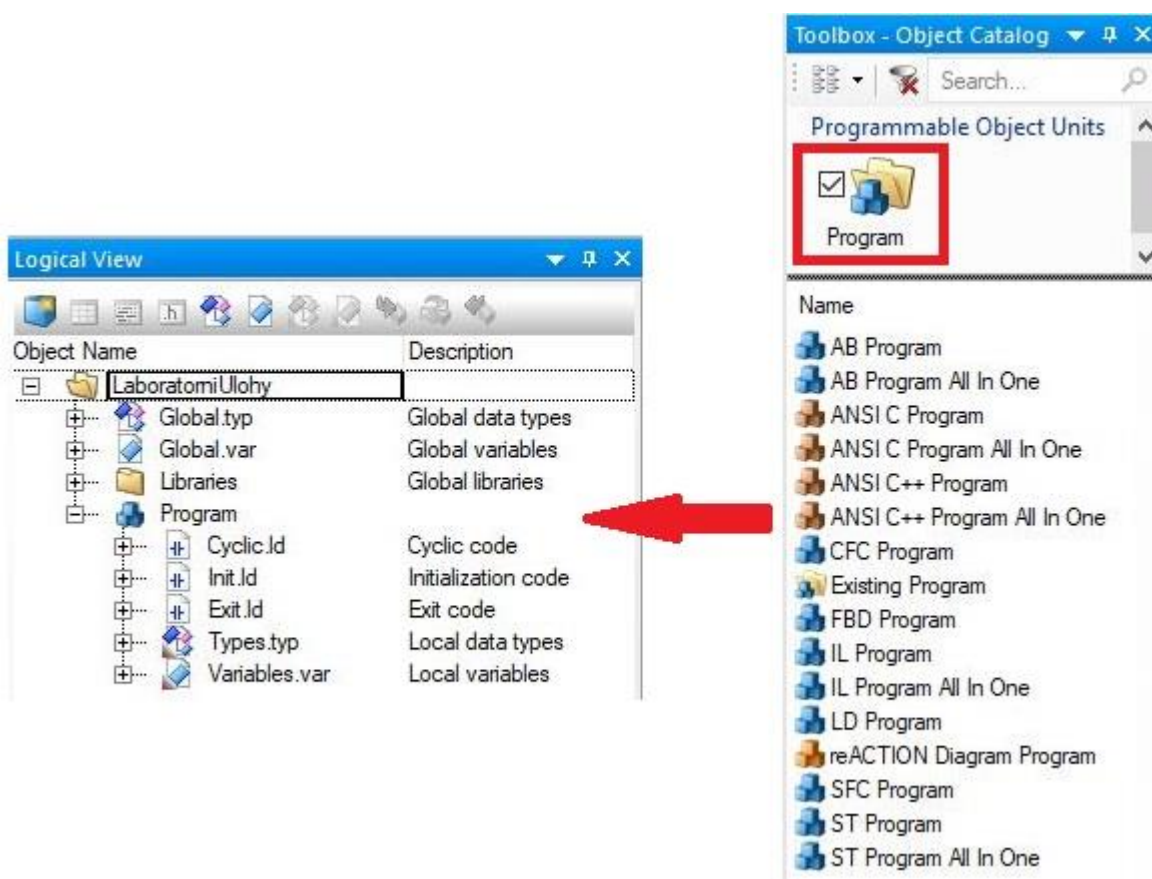
Obr. 6.12 – Vyhledání položky B&R Libraries



Obr. 6.13 – Implementace knihovny standard

6.7 VYTVOŘENÍ PROGRAMU

Vytvoření programu ve vybraném programovacím jazyce lze uskutečnit pomocí **Toolbox – Object Catalog**. Pro přehlednější výběr je možné zaškrtnout položku **Program**. Přesunutím nebo dvojklikem se zvolený program implementuje do záložky **Logical View** a z této záložky již s daným programem lze pracovat. Postup je znázorněn na obr. 6.14.

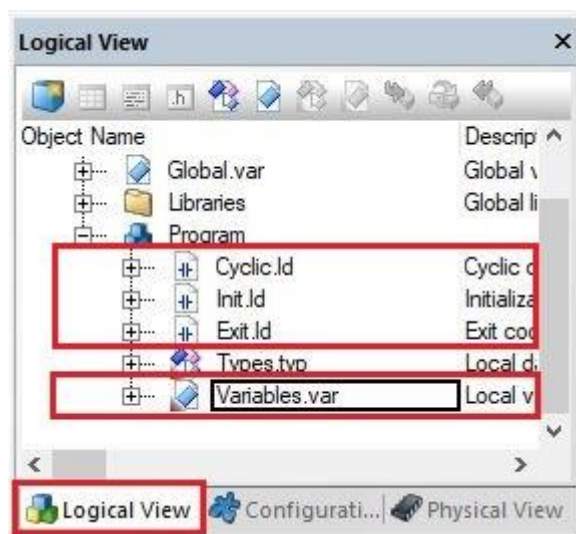


Obr. 6.14 – Vytvoření programu

6.8 TVORBA PROGRAMU

Pro zahájení tvorby je zapotřebí otevřít jednu z částí programu: **Cyclic.Id**, **Init.Id**, **Exit.Id**, které se nachází v záložce **Logical View** (obr. 6.15). Otevře se pracovní prostor, ve kterém je možné daný program sestavit. Rozdělení programu podle způsobu vykonávání je vysvětleno v kapitole 3. Součástí psaní programu je i definice proměnných. Proměnné lze definovat pomocí položky **Variables.var**, která se taktéž nachází v záložce **Logical View** a je také znázorněna na obr.

Konkrétní příklad psaní programu a definice proměnných je součástí vzorového řešení úlohy č.1, které je uvedeno v kapitole 7.

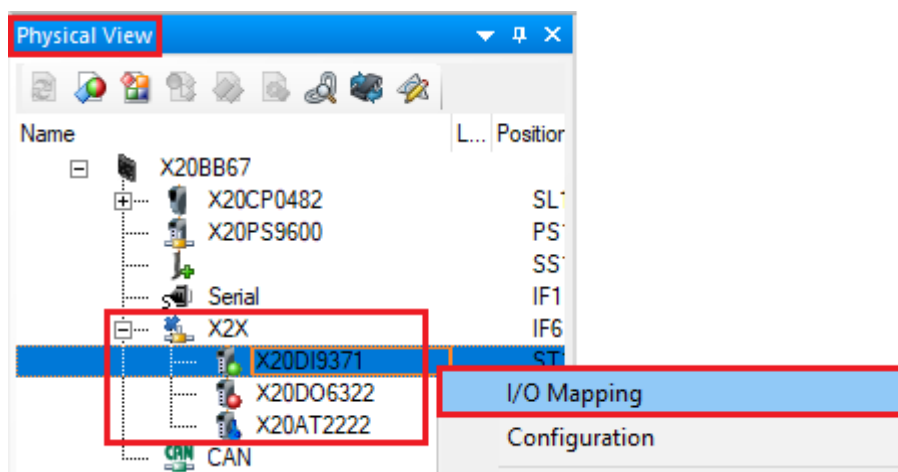


Obr. 6.15 – Záložka Logical View

6.9 MAPOVÁNÍ PROMĚNNÝCH

Mapování proměnných lze realizovat ze záložky **Physical View**, kde se pod položkou **X2X** nachází nakonfigurované vstupní a výstupní moduly. Výběrem možnosti **I/O Mapping** u příslušného modulu se otevře konfigurační okno, ve kterém je možné jednotlivé proměnné přiřadit odpovídajícím vstupům a výstupům. Postup otevření konfiguračního okna je znázorněn na obr. 6.16.

Praktický příklad mapování proměnných je součástí vzorového řešení úlohy č. 1, které se nachází v kapitole 7.



Obr. 6.16 – Mapování proměnných

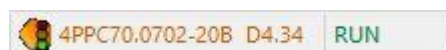
6.10 SIMULÁTOR

Aktivaci simulátoru lze provést pomocí položky **Activate Simulation**, která se nachází v záložce **Online**. Postup aktivace je znázorněn na obr. 6.17.



Obr. 6.17 – Aktivace simulátoru

Úspěšné spuštění simulátoru (RUN) je signalizováno na spodní liště vývojového prostředí (obr. 6.18). Pro nahrání programu do simulátoru slouží tlačítko **Transfer** (obr. 6.31), které je umístěno na horní pracovní liště.

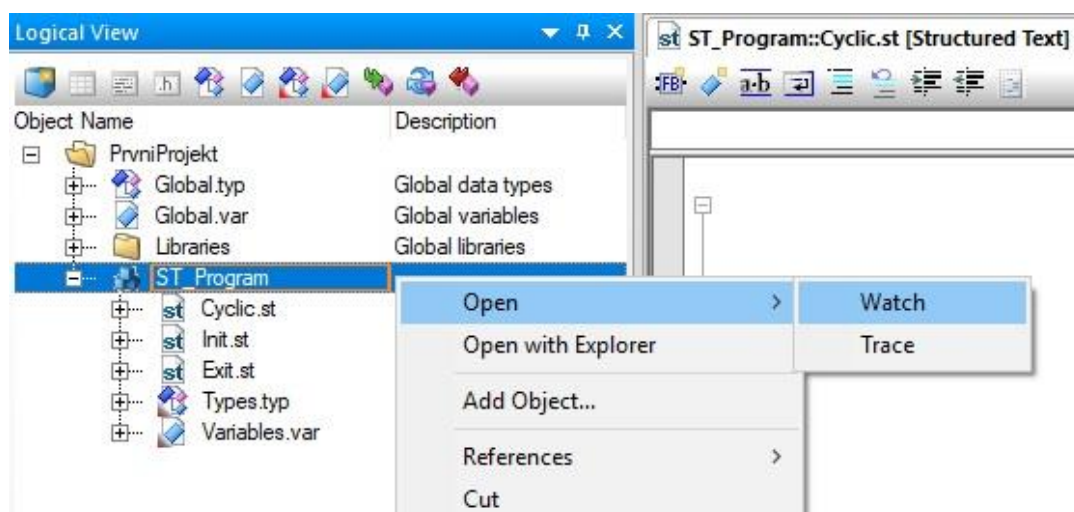


Obr. 6.18 – Signalizace spuštěného simulátoru

Pro následnou práci v simulátoru lze použít diagnostický nástroj Watch, jehož konfigurace je popsána v oddílu 6.11.

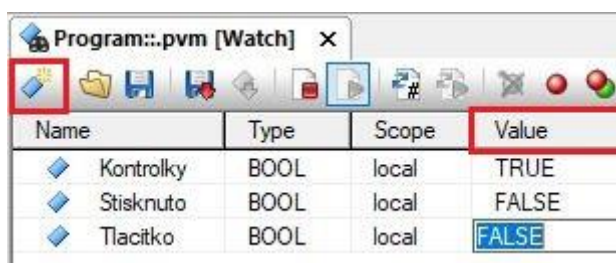
6.11 DIAGNOSTICKÝ NÁSTROJ „WATCH“

Po úspěšném nahrání programu do simulátoru lze ze záložky **Logical View** spustit diagnostický nástroj **Watch**. Spuštění nástroje je zřetelně znázorněno na obr. 6.19.



Obr. 6.19 – Spuštění diagnostické nástroje Watch

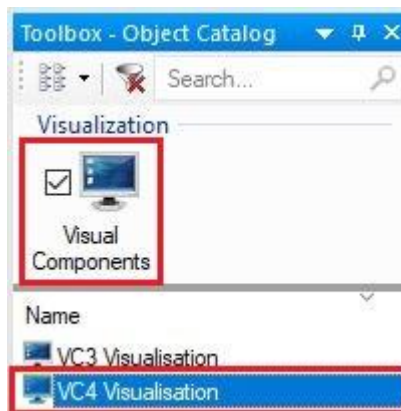
Otevře se pracovní oblast diagnostického nástroje (obr. 6.20). Do této pracovní oblasti lze přidat proměnné, které mají být sledovány. Proměnné je možné implementovat pomocí tlačítka **Insert Variable**, které je umístěno na levé straně pracovní lišty nástroje. Hodnoty sledovaných proměnných lze editovat v kolonce **Value**.



Obr. 6.20 – Diagnostický nástroj Watch

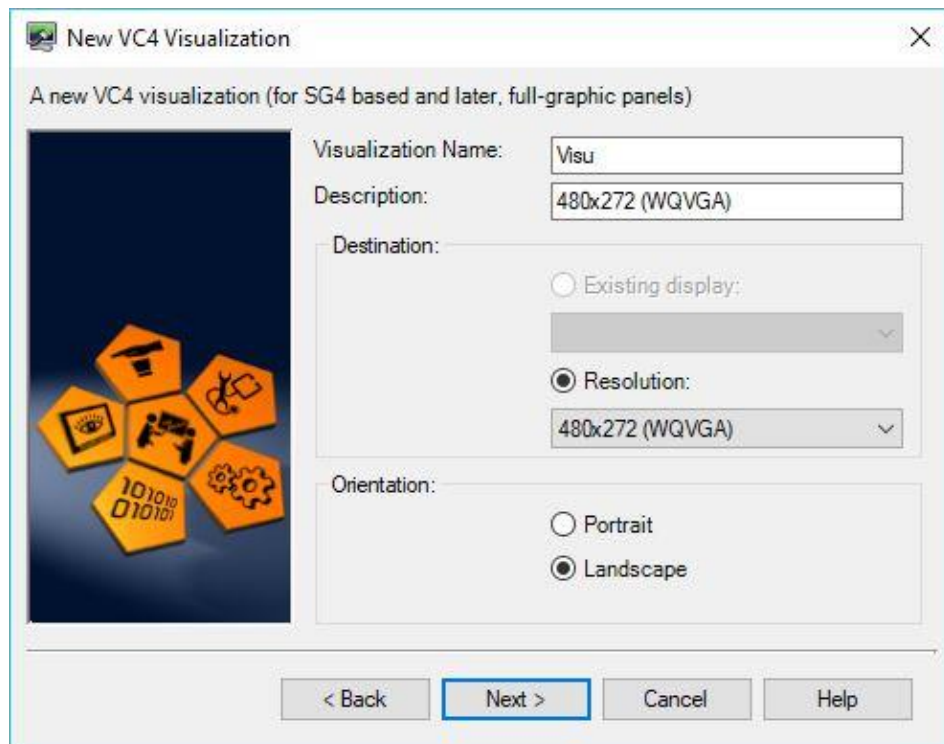
6.12 VYTVOŘENÍ VIZUALIZAČNÍHO PROSTŘEDÍ

Vizualizační prostředí (Visual Components) lze do projektu implementovat z **Toolbox – Object Catalog**. Zaškrtnutím položky **Visual Components** se zúží nabízený výběr možností. Dvojitým kliknutím na **VC4 Visualisation** se spustí tvorba vizualizačního prostředí (obr. 6.21).



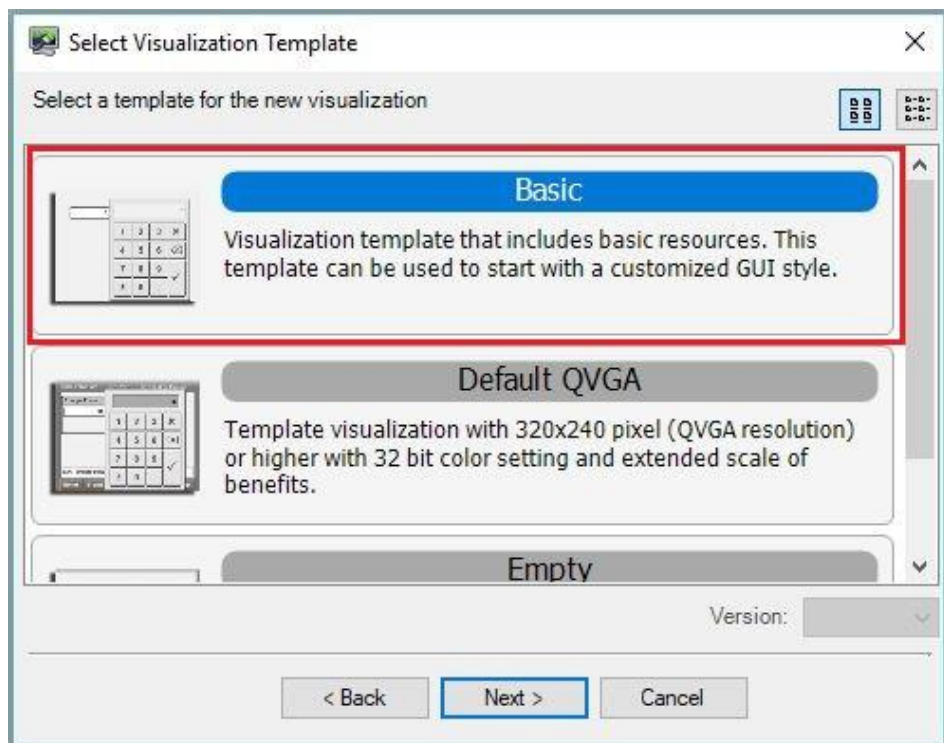
Obr. 6.21 – Implementace vizualizačního prostředí

Zobrazí se konfigurační okno, ve kterém lze editovat název vizualizace, popis, rozlišení a orientaci zobrazení. Na obr. 6.22 je znázorněna konfigurace odpovídající použitému zobrazovacímu panelu.



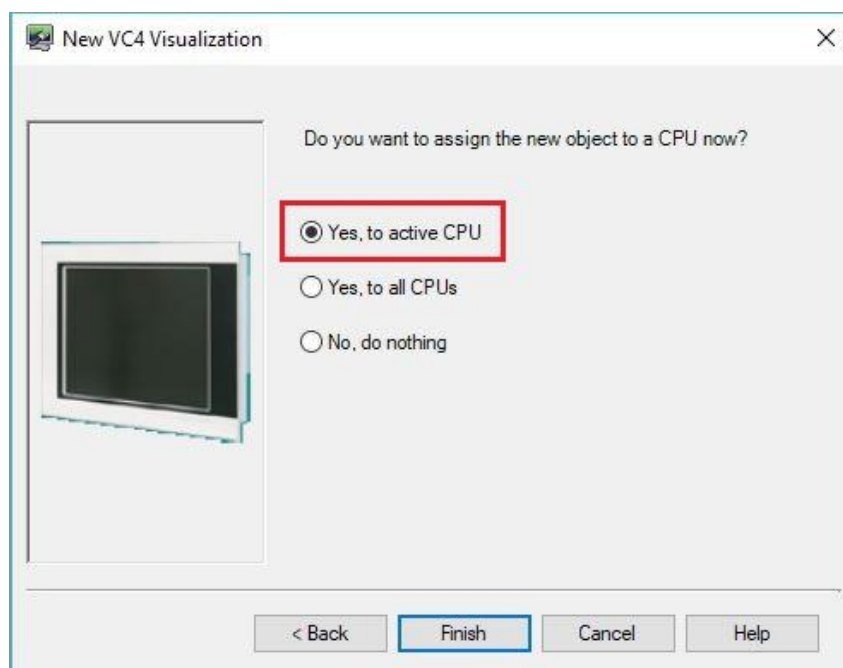
Obr. 6.22 – Konfigurace vizualizačního prostředí

V další části je potřeba vybrat šablonu **Basic** (obr. 6.23).



Obr. 6.23 – Výběr šablony

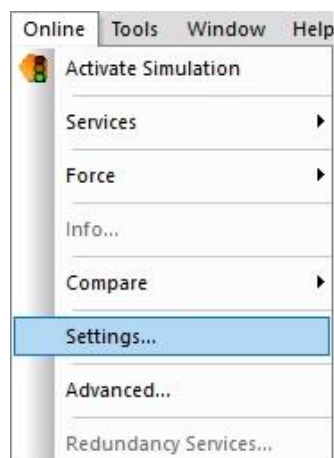
Posledním bodem je přiřazení vizualizačního prostředí aktivní hardwarové konfiguraci, které je znázorněno na obr. 6.24. Nutno zvolit položku **Yes, to active CPU** a stisknout tlačítko **Finish**. Práce ve vytvořeném vizualizačním prostředí je demonstrována na úloze č. 4. Příslušný postup tvorby vizualizace je popsán v kapitole 7.



Obr. 6.24 –Přiřazení vizualizačního prostředí aktivní HW konfiguraci

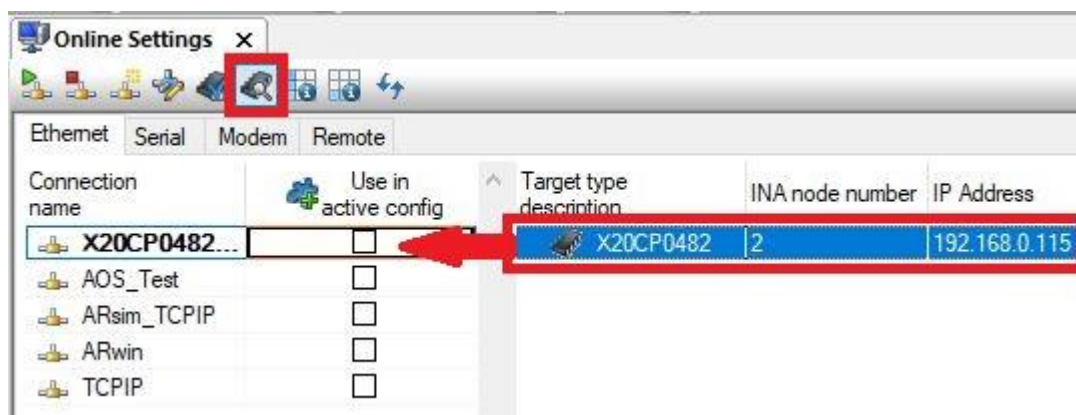
6.13 PŘÍPOJENÍ K PLC POMOCÍ SÍTĚ ETHERNET

Konfiguraci online připojení lze otevřít z hlavní lišty pomocí záložky **Online** a položky **Settings**. Spuštění konfigurace je znázorněno na obr. 6.25.



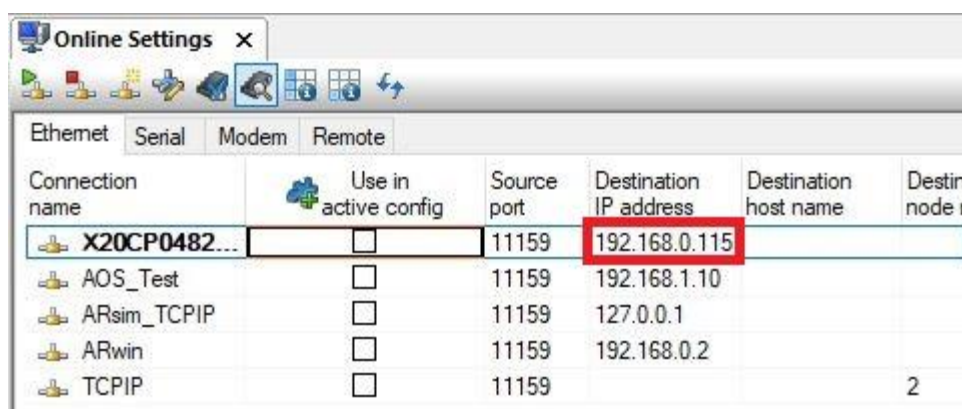
Obr. 6.25 – Spuštění konfigurace připojení

Otevře se okno **Online Settings**. Stisknutím tlačítka **Browse** se spustí vyhledávání připojených PLC. Nalezená zařízení se objeví v pravé části. Z tohoto seznamu je nutné vybrat používané PLC a myši jej přetáhnout do levé části. Postup je znázorněn na obr. 6.26.



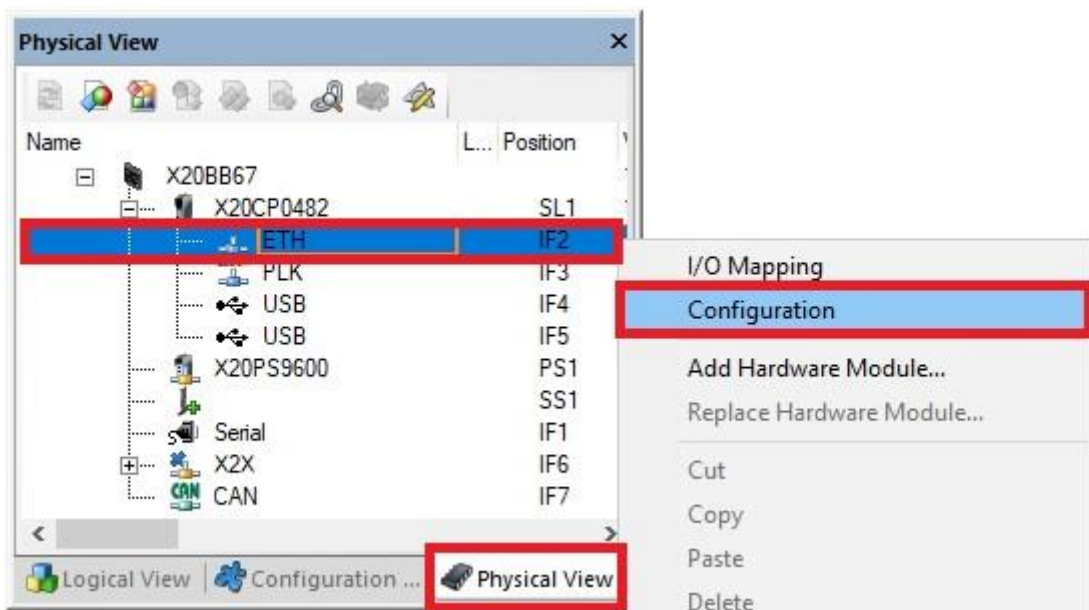
Obr. 6.26 – Vyhledání připojených PLC

Dalším bodem je nastavení IP adresy PLC, které se provádí taktéž v okně **Online Settings**. Zadaná IP adresa se musí nacházet ve stejné síti jako počítač, ke kterému je PLC připojeno. Kolonka pro nastavení IP adresy PLC je znázorněna na obr. 6.27.



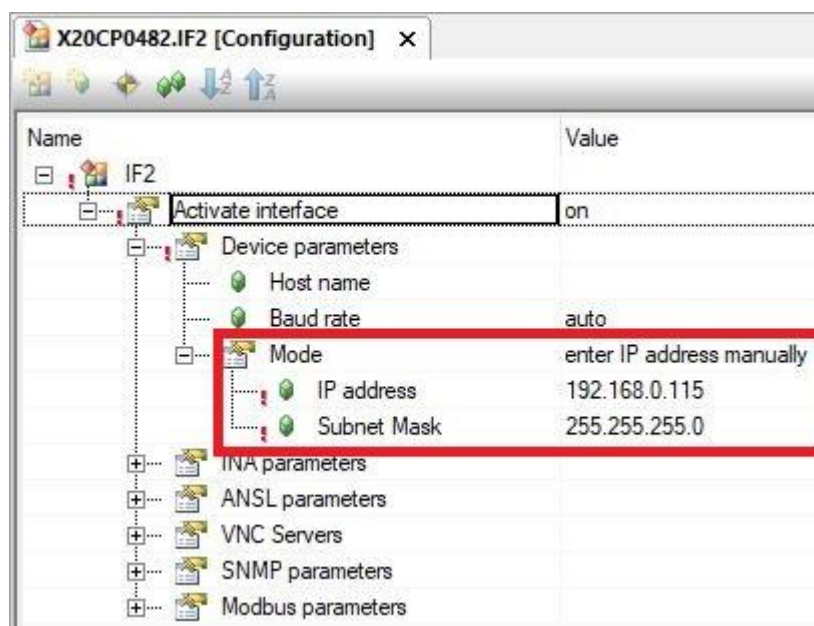
Obr. 6.27 – Nastavení IP adresy PLC

Následuje konfigurace Ethernetu na cílovém zařízení. Konfiguraci lze otevřít ze záložky **Physical View**, kliknutím pravým tlačítkem na položku **ETH** a vybráním možnosti **Configuration**. Postup je znázorněn na obr. 6.28.



Obr. 6.28 – Konfigurace Ethernetu na cílovém zařízení

V konfiguračním okně, zobrazeném na obr. 6.29, je nutné rozkliknout položku **Mode** a zvolit možnost **enter IP address manually**. Do kolonky **IP address** zapsat zvolenou IP adresu používaného PLC a v případě potřeby upravit masku sítě (**Subnet Mask**).



Obr. 6.29 – Zadání IP adresy do položky Mode

Pro připojení PLC je nutné znovu otevřít okno **Online Settings**, jehož otevření je popsáno pomocí obr. 6.25. V seznamu zařízení označit používané PLC a pomocí tlačítka **Connect** jej připojit. Postup je znázorněn na obr. 6.30.



Obr. 6.30 – Připojení PLC

Po úspěšném připojení stisknout tlačítko **Transfer**, které spustí nahrávání vytvořeného programu a konfigurace do PLC. Tlačítko **Transfer** se nachází na horní pracovní liště a jeho ikonka je znázorněna na obr. 6.31.



Obr. 6.31 – Tlačítko Transfer

7 VZOROVÁ ŘEŠENÍ LABORATORNÍCH ÚLOH

V této kapitole jsou uvedena vzorová řešení laboratorních úloh zadaných v kapitole 5. Součástí řešení je i popis důležitých částí programu. Jednotlivé úlohy jsou vyřešeny pomocí programovacího jazyka Ladder. Program napsaný v jazyce Ladder je rozdělen do tzv. sítí, které jsou automaticky číslovány. V samotném popisu vzorového řešení je odkazováno právě na čísla sítí, pomocí kterých je vysvětlena odpovídající část programu. Umístění číslování v programu je znázorněno na obr. 7.1, ve kterém je vyznačeno očíslování sítě 1.



Obr. 7.1 – Číslování sítě

Součástí jednotlivých řešení jsou i tabulky popisující mapování použitých proměnných k reálným vstupům a výstupům PLC. Podrobné informace o použitých programovacích prvcích, jako jsou kontakty, časovače, čítače, komparátory apod., lze dohledat v nápovědě. Nápovědu lze spustit označením konkrétní části programu a stisknutím tlačítka F1.

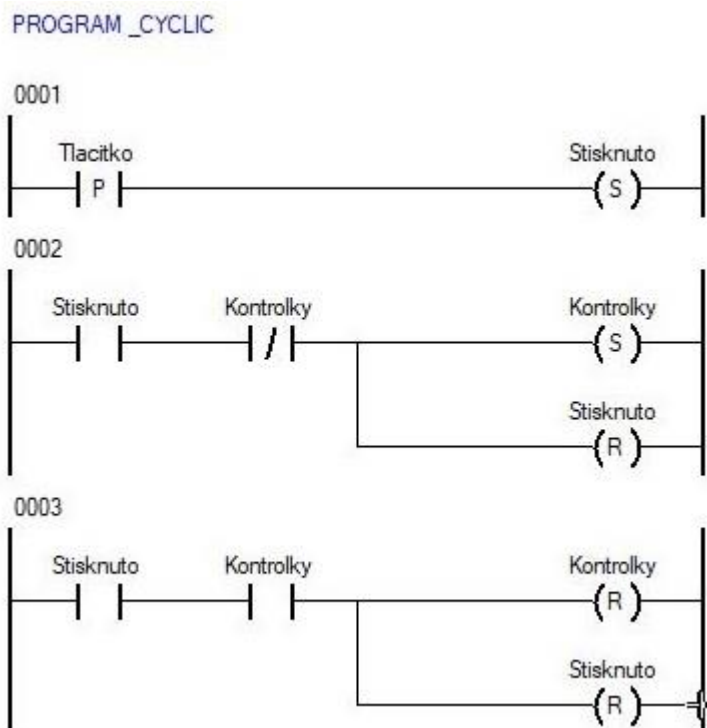
7.1 ÚLOHA Č. 1

Potřebné úkony pro sestavení funkčního programu včetně mapování proměnných jsou demonstrovány na úloze č.1. Jednotlivé kroky konfigurace jsou uvedeny v pododdílech 7.1.1 a 7.1.2. V tab. 7.1 a tab. 7.2 je popsáno mapování příslušných proměnných. Vzorové řešení úlohy je znázorněno na obr. 7.2, popis programu je následující:

Síť 1: Pokud dojde ke stisknutí tlačítka (spínací kontakt „P“ reaguje na náběžnou hranu signálu), bude proměnná **Stisknuto** v logické 1 (výstup S = SET).

Síť 2: Jeli sepnutý kontakt **Stisknuto** (tlačítko bylo stisknuto) a není rozepnutý kontakt **Kontrolky** (kontrolky nesvítí), dojde ke změně stavu proměnné **Kontrolky** do logické 1 (kontrolky se rozsvítí) a proměnné **Stisknuto** do logické 0 (výstup R = RESET).

Síť 3: Je-li sepnutý kontakt **Stisknuto** (tlačítko bylo stisknuto) a kontakt **Kontrolky** (kontrolky svítí), dojde ke změně stavu proměnné **Kontrolky** do logické 0 (kontrolky zhasnou) a proměnné **Stisknuto** do logické 0.



Obr. 7.2 – Vzorové řešení úlohy č. 1

Tab. 7.1 – Mapování modulu X20DI9371 pro úlohu č. 1

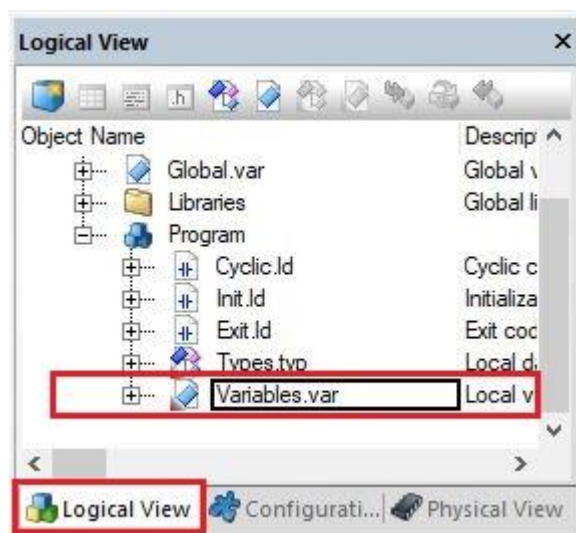
X20DI9371		
Název vstupu	Prvek	Proměnná
DigitalInput01	1. tlačítko	Tlacitko

Tab. 7.2 – Mapování modulu X20DO6322 pro úlohu č. 1

X20DO6322		
Název výstupu	Prvek	Proměnná
DigitalOutput1	1. kontrolka	Kontrolky
DigitalOutput2	2. kontrolka	Kontrolky
DigitalOutput3	3. kontrolka	Kontrolky
DigitalOutput4	4. kontrolka	Kontrolky
DigitalOutput5	5. kontrolka	Kontrolky
DigitalOutput6	6. kontrolka	Kontrolky

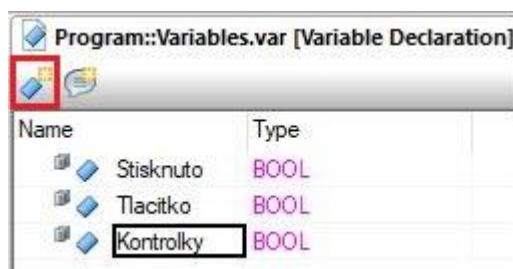
7.1.1 Psaní programu

Na začátku psaní programu je vhodné si definovat veškeré proměnné, které budou v aplikaci potřeba. Proměnné lze definovat pomocí položky **Variables.var**, která se nachází v záložce **Logical View** (obr. 7.3).



Obr. 7.5 – Položka Variables.var

Ikonka **Add variable** pro přidání nových proměnných a seznam použitých proměnných jsou znázorněny na obr. 7.4. V softwaru Automation Studio je možné jednu proměnou přiřadit více výstupům, a proto je pro všech šest kontrolků (výstupů) vytvořena pouze jedna proměnná **Kontrolky**.



Obr. 7.4 – Ikonka Add Variable a seznam proměnných

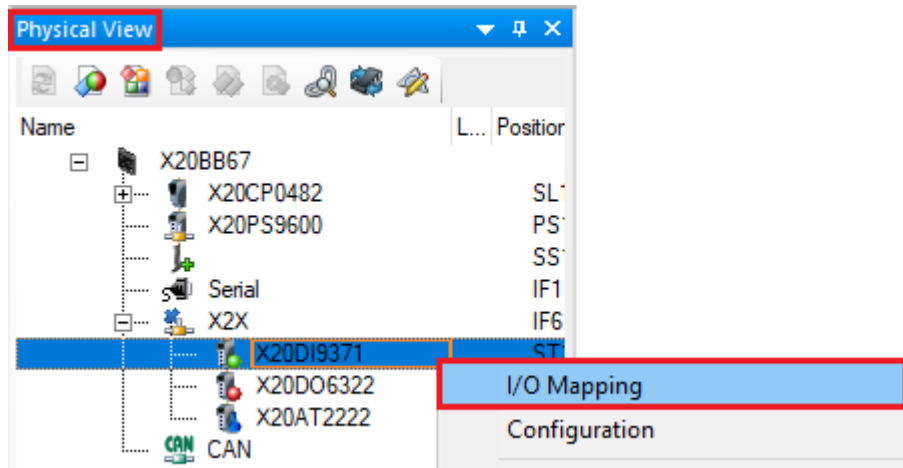
Pro psaní programu je zapotřebí otevřít cyklickou část **Cyclic.Id** (obr. 7.5). Řešení zadané aplikace v programovacím jazyce příčkového diagramu je znázorněno na obr. 7.2.



Obr. 7.3 – Volba cyklické části Cyclic.Id

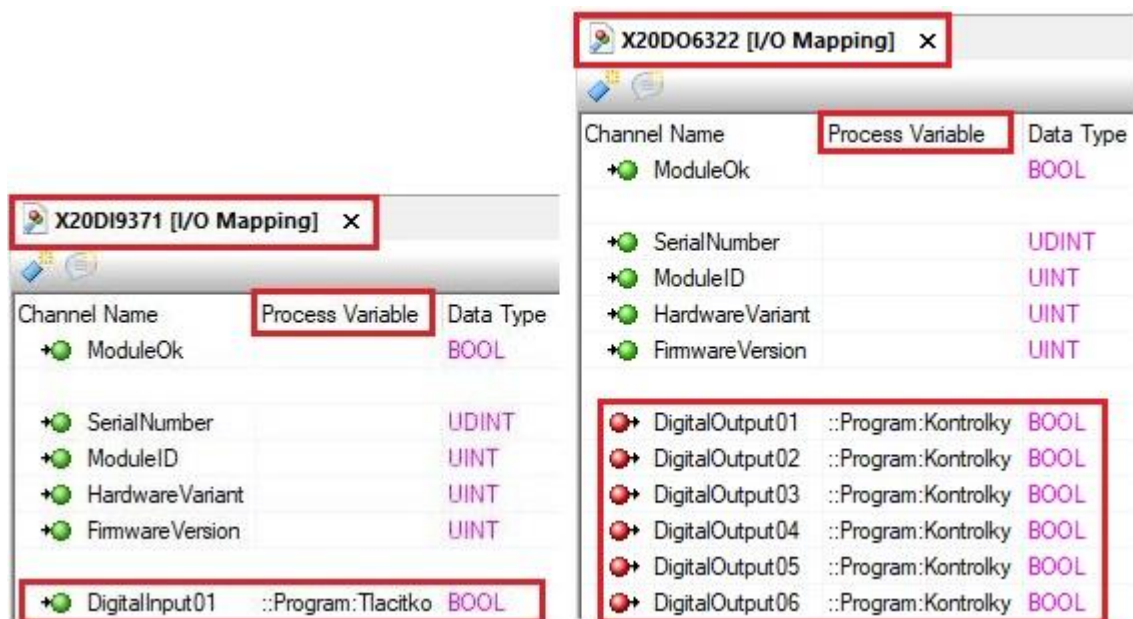
7.1.2 Mapování proměnných

Nakonfigurované vstupní a výstupní moduly se nachází v záložce **Physical View** pod položkou **X2X**. Pro mapování je potřeba pravým tlačítkem rozkliknout možnosti příslušného modulu a vybrat položku **I/O Mapping**. Postup ilustruje obr. 7.6.



Obr. 7.6 – Položka I/O Mapping

V kapitole 4 je popsáno připojení použitých prvků na reálné vstupy a výstupy PLC. Na obr. 7.7 je znovu znázorněno mapování proměnných úlohy č. 1 pro vstupní modul X2DI9371 a výstupní modul X2DO6322. Použité proměnné se k jednotlivým vstupům či výstupům přiřazují pomocí kolonky **Process Variable**.



Obr. 7.7 – Mapování proměnných pro úlohu č. 1

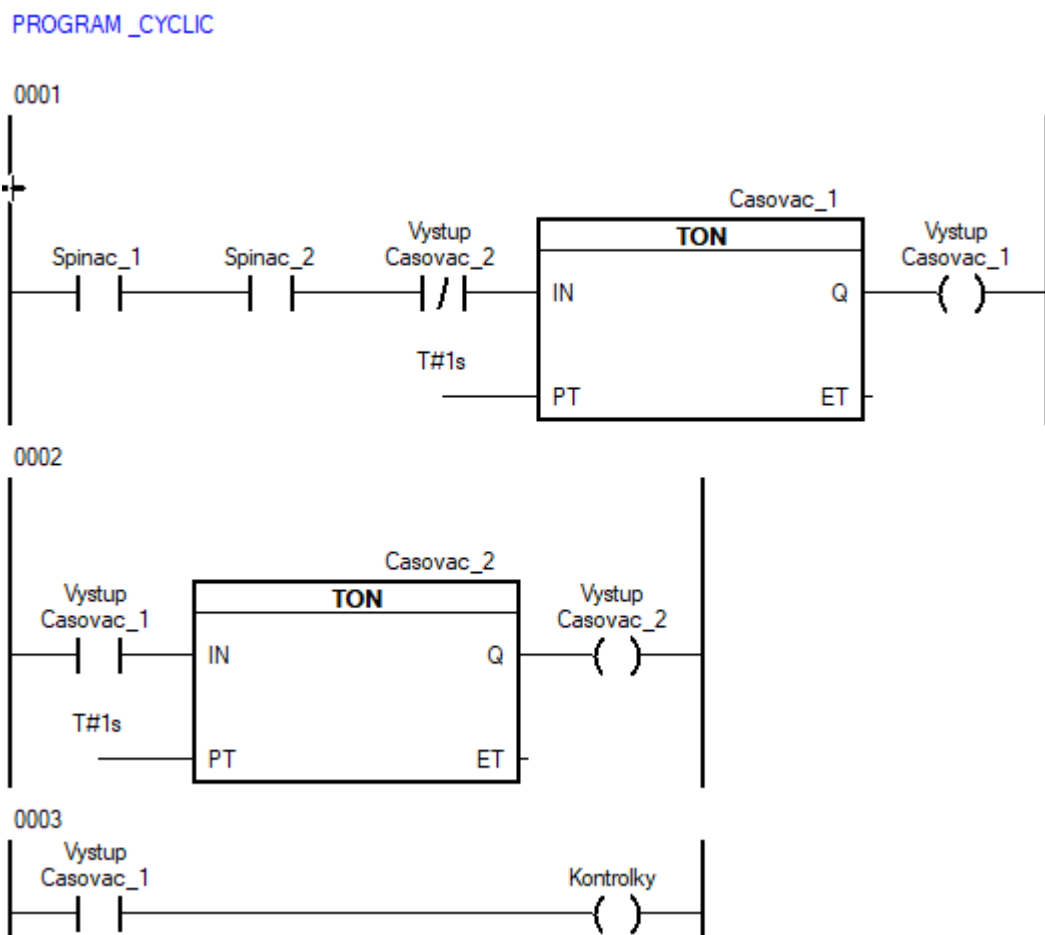
7.2 ÚLOHA Č. 2

Mapování proměnných pro tuto úlohu je uvedeno v tab. 7.3 a tab. 7.4. Vzorové řešení programu je znázorněno na obr. 7.8, popis je následující:

Síť 1: Dojde-li k sepnutí obou spínačů (**Spinac_1** a **Spinac_2**) a není rozepnutý kontakt **VystupCasovac_2**, bude aktivován časovač TON (**Casovac_1**), jehož čas počítání je nastaven na 1 vteřinu ($PT = T\#1\text{ s}$). Po uplynutí této doby je aktivován výstup časovače a proměnná **VystupCasovac_1** je v logické 1.

Síť 2: Po sepnutí kontaktu **VystupCasovac_1** dojde k aktivaci druhého časovače TON (**Casovac_2**), který je také nastaven na dobu počítání 1 vteřina. Po uplynutí doby je aktivován výstup časovače a proměnná **VystupCasovac_2** je v logické 1.

Síť 3: Je-li sepnutý kontakt **VystupCasovac_1**, je výstup **Kontrolky** v logické 1 (kontrolky svítí).



Obr. 7.8 – Vzorové řešení úlohy č. 2

Tab. 7.3 – Mapování modulu X20DI9371 pro úlohu č. 2

X20DI9371		
Název vstupu	Prvek	Proměnná
DigitalInput07	1. spínač	Spinac_1
DigitalInput08	2. spínač	Spinac_2

Tab. 7.4 – Mapování modulu X20DO6322 pro úlohu č. 2

X20DO6322		
Název výstupu	Prvek	Proměnná
DigitalOutput1	1. kontrolka	Kontrolky
DigitalOutput2	2. kontrolka	Kontrolky
DigitalOutput3	3. kontrolka	Kontrolky
DigitalOutput4	4. kontrolka	Kontrolky
DigitalOutput5	5. kontrolka	Kontrolky
DigitalOutput6	6. kontrolka	Kontrolky

7.3 ÚLOHA Č. 3

Tab. 7.5 a tab. 7.6 definují mapování příslušných proměnných. Vzorové řešení úlohy je znázorněno na obr. 7.9, popis programu je následující:

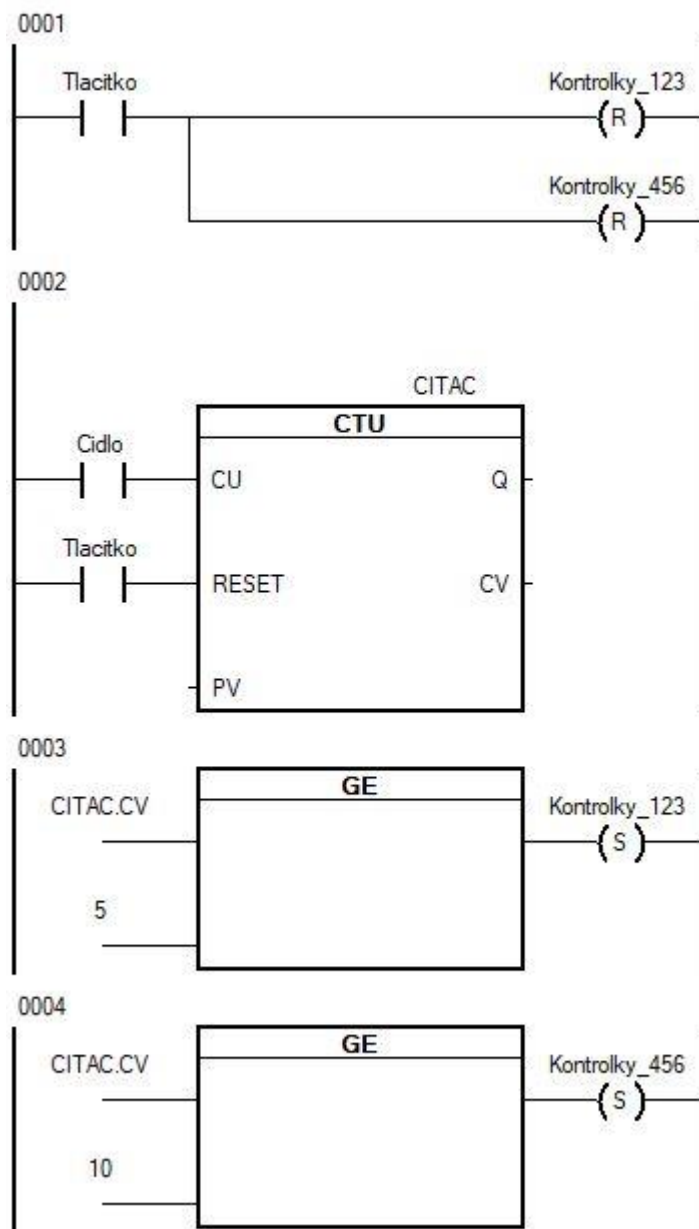
Síť 1: Na začátek cyklu je umístěn spínací kontakt tlačítka, kdy při jeho stisknutí dojde k resetování výstupů (**Kontrolky_123** a **Kontrolky_456**). To znamená, že pokud svítí kontrolky a je stisknuto tlačítko, dojde k jejich zhasnutí.

Síť 2: Zakrytí čidla sepne kontakt **Cidlo**, který aktivuje čítač **CITAC** a inkrementuje jeho hodnotu o 1. Zároveň je na RESET čítače připojen spínací kontakt tlačítka (při stisknutí tlačítka je čítač resetován a počítá se znovu od nuly).

Síť 3: Komparátor porovnává inkrementovanou hodnotu čítače (**CITAC.CV**) s nastaveným parametrem (**5**). Pokud je hodnota proměnné stejná nebo větší než nastavený parametr, dojde k aktivaci výstupu **Kontrolky_123** (rozsvítí kontrolky 1, 2 a 3).

Síť 4: Funguje na stejném principu jako část programu v síti 3. Jediným rozdílem je, že inkrementovaná hodnota čítače je porovnávána s hodnotou **10**. V tomto případě tedy dojde k aktivaci výstupu **Kontrolky_456** až při desátém zakrytí čidla (rozsvítí se kontrolky 4, 5 a 6).

PROGRAM_CYCLIC



Obr. 7.9 – Vzorové řešení úlohy č. 3

Tab. 7.5 – Mapování modulu X20DI9371 pro úlohu č. 3

X20DI9371		
Název vstupu	Prvek	Proměnná
DigitalInput01	1. tlačítko	Tlacitko
DigitalInput12	Čidlo	Cidlo

Tab. 7.6 – Mapování modulu X20DO6322 pro úlohu č. 3

X20DO6322		
Název výstupu	Prvek	Proměnná
DigitalOutput1	1. kontrolka	Kontrolky_123
DigitalOutput2	2. kontrolka	Kontrolky_123
DigitalOutput3	3. kontrolka	Kontrolky_123
DigitalOutput4	4. kontrolka	Kontrolky_456
DigitalOutput5	5. kontrolka	Kontrolky_456
DigitalOutput6	6. kontrolka	Kontrolky_456

7.4 ÚLOHA Č. 4

Tab. 7.7 a tab. 7.6 charakterizují mapování použitých proměnných. Vzorové řešení programové části je znázorněno na obr. 7.10 a tvorba příslušné vizualizace je vysvětlena v pododdíle 7.4.1. Popis sestaveného programu je následující:

Síť 1: V základu je analogová hodnota datového typu UINT (neznaménkové celé číslo). Z důvodu počítání hodnot s desetinnou čárkou je pomocí funkčního bloku převedeno na datový typ REAL.

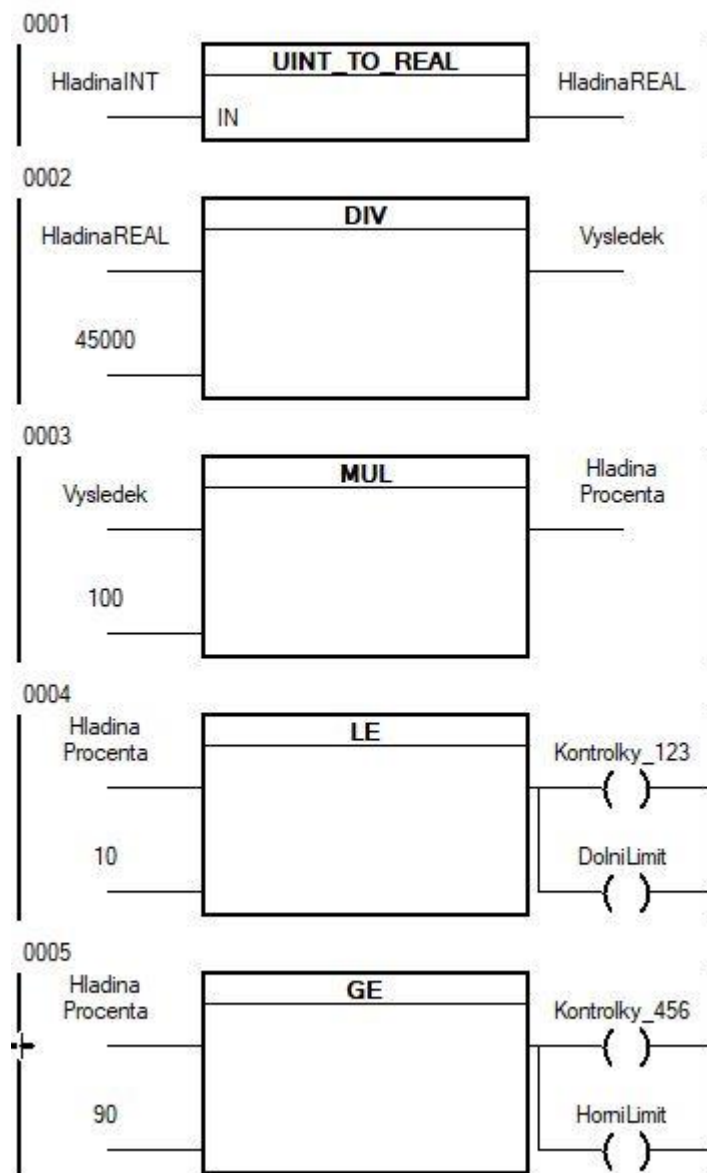
Síť 2: V této síti je zahájeno přepočítání číselné hodnoty hladiny z datového typu REAL na procenta, což je realizovatelné pomocí trojčlenky. Funkční blok DIV slouží k vydělení hodnot na vstupu. Kde **HladinaREAL** je aktuální hodnota analogového vstupu a **45000** je maximální dosažitelná hodnota na tomto vstupu (tato hodnota se odvíjí od typu použitého potenciometru a rozlišení A/D převodníku v modulu). Výsledek dělení je uložen do proměnné **Vysledek**.

Síť 3: Funkční blok MUL vynásobí hodnoty na vstupu. Kde proměnná **Vysledek** je výsledkem předchozího výpočtu a vynásobením hodnotou **100** je hodnota hladiny vyčíslena v procentech. Výsledek je uložen do proměnné **HladinaProcenta**.

Síť 4: Pomocí funkčního bloku LE je porovnávána aktuální hodnota hladiny v procentech (**HladinaProcenta**) s nastavenou hodnotou dolního limitu **10** (10 %). Pokud je hodnota hladiny menší než 10 %, dojde k aktivaci výstupů **Kontrolky_123** (rozsvítí se kontrolky 1, 2 a 3) a **DolniLimit** (aktivuje se chybové hlášení ve vizualizaci).

Síť 5: Pomocí funkčního bloku GE je porovnávána aktuální hodnota hladiny v procentech (**HladinaProcenta**) s nastavenou hodnotou dolního limitu **90** (90 %). Pokud je hodnota hladiny větší než 90 %, dojde k aktivaci výstupů **Kontrolky_123** (rozsvítí se kontrolky 4, 5 a 6) a **HorniLimit** (aktivuje se chybové hlášení ve vizualizaci).

PROGRAM_CYCLIC



Obr. 7.10 – Vzorové řešení úlohy č. 4

Tab. 7.7 – Mapování modulu X20DO6322 pro úlohu č. 4

X20DO6322		
Název výstupu	Prvek	Proměnná
DigitalOutput1	1. kontrolka	Kontrolky_123
DigitalOutput2	2. kontrolka	Kontrolky_123
DigitalOutput3	3. kontrolka	Kontrolky_123
DigitalOutput4	4. kontrolka	Kontrolky_456
DigitalOutput5	5. kontrolka	Kontrolky_456
DigitalOutput6	6. kontrolka	Kontrolky_456

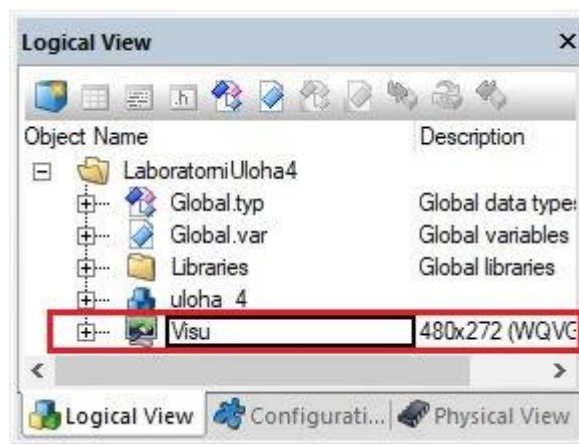
Tab. 7.8 – Mapování modulu X20AT2222 pro úlohu č. 4

X20AT2222		
Název vstupu	Prvek	Proměnná
Resistor01	Potenciometr	HladinaINT

7.4.1 Tvorba vizualizace

Vytvoření vizualizačního prostředí je vysvětleno v kapitole 6 a jeho popis uveden v kapitole 3. V tomto pododdíle je demonstrována tvorba vizualizace pro úlohu č. 4, jejíž proces hlídání výšky hladiny je vizualizován.

Implementované vizualizační prostředí (**vizu**) lze otevřít ze záložky **Logical View** (obr. 7.11).



Obr. 7.11 – Spuštění vizualizačního prostředí

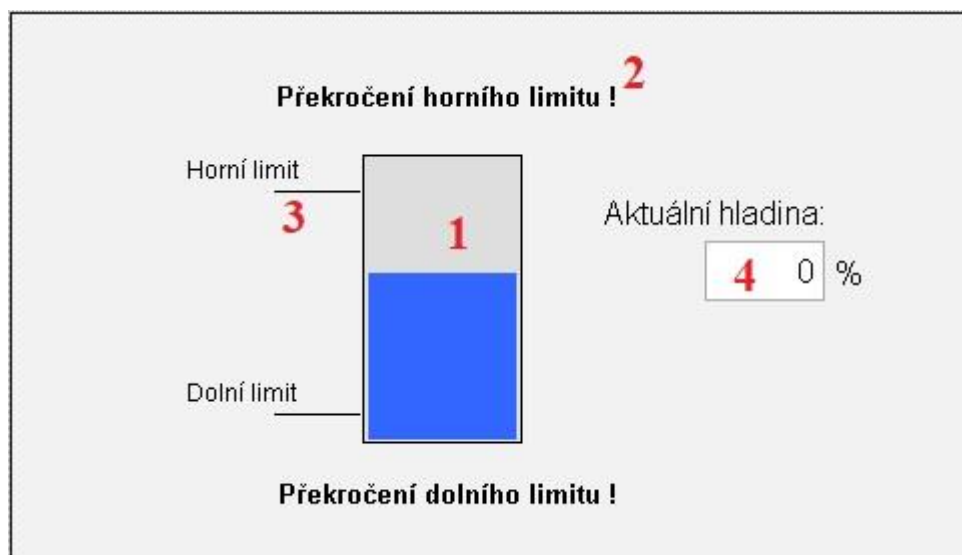
Pracovní plochu je možné otevřít ze stromového adresáře, který je umístěn na levé straně vývojového prostředí, a to vybráním položky **Init_Page** (obr. 7.12).



Obr. 7.12 – Výběr položky Init_Page

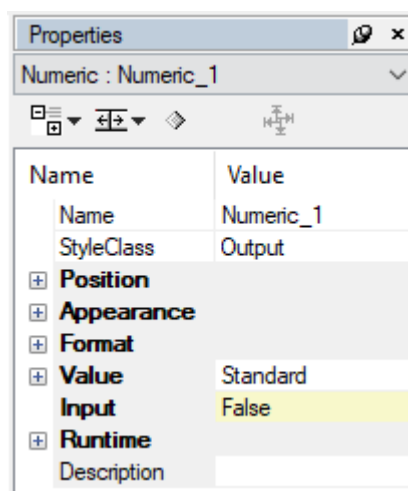
Na obr. 7.13 je znázorněna pracovní plocha s již umístěnými vizualizačními prvky, které jsou potřebné pro splnění zadání úlohy. Jednotlivé prvky lze implementovat ze souboru nástrojů. Seznam použitých prvků je následující:

1. „BarGraph“ – graficky znázorňuje výšku hladiny v nádrži.
2. „Text“ – použito pro veškerý text ve vizualizaci.
3. „Line“ – zobrazuje umístění horního a dolního limitu.
4. „Numeric“ – podává číselnou informaci o zaplnění nádrže.

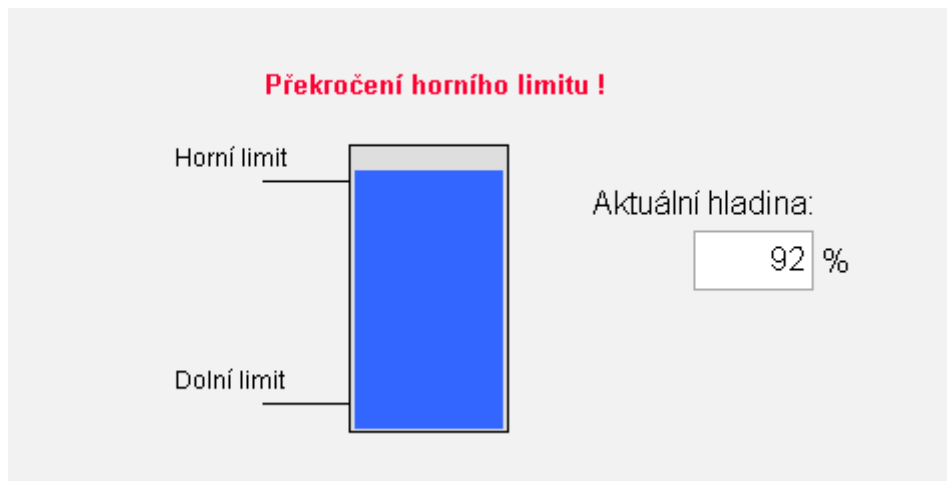


Obr. 7.13 – Pracovní plocha s vizualizačními prvky

Označením vloženého prvku se v pravé části vývojového prostředí objeví odpovídající konfigurační okno (obr. 7.14). V konfiguračním okně lze editovat vzhled, způsob zobrazení, formát, mapování proměnných atd. Příklad výsledného řešení vizualizace je znázorněn na obr. 7.15.



Obr. 7.14 – Konfigurační okno



Obr. 7.15 – Vzorové řešení vizualizace pro úlohu č. 4

8 ZÁVĚR

V rámci bakalářské práce byla sestavena sada čtyř laboratorních úloh, která svým rozsahem odpovídá zkonstruovanému laboratornímu modelu. Laboratorní model slouží k reálnému testování vytvořených aplikací a demonstruje automatizaci pomocí programovatelných logických automatů od společnosti B&R Automation. Součástí práce je podrobný návod popisující práci ve vývojovém prostředí Automation Studio.

Vypracováním laboratorní úlohy č. 1 je uživatel seznámen s pracovním prostředím softwaru Automation Studio. Za pomoci zpracovaného návodu je obeznámen s úkony potřebnými pro realizaci úloh. Vypracováním programové části si vyzkouší ovládání výstupu jedním tlačítkem.

Realizací laboratorní úlohy č. 2 je objasněno použití časovačů a logické funkce AND pro ovládání výstupu dvěma spínači. Sestavením programové části si uživatel vyzkouší rozblikání kontrolky s využitím dvou časovačů. Podobný postup je v praxi často používán.

Úloha č. 3 demonstruje práci s binárním čidlem, čítači a komparátory. Uživatel simuluje zakrytí čidla přiložením kovového předmětu. Programovou částí je zajištěno zpracování signálů od čidla a následné ovládání výstupů.

Úloha č. 4 se zaměřuje na práci s analogovou veličinou. Uživatel je seznámen s převodem datových typů, použitím matematických funkcí a vizualizačním prostředím. Programovou částí je obstarána kontrola limitů výšky hladiny v nádrži. Aktuální výška hladiny je simulována potenciometrem. Součástí úlohy je i tvorba vizualizace daného procesu.

Funkčnost všech laboratorních úloh byla otestována na sestaveném laboratorním modelu. Model lze dále využít pro testování nově navržených aplikací. Nabízí se také možnost rozšíření modelu o další ovládací prvky, či zkonstruování doplňujících soustav, které mohou být tímto modelem řízeny.

POUŽITÁ LITERATURA

- Advanced system diagnostics via the web. *B&R Perfection in Automation* [online]. 16. 2. 2011. [cit. 2018-04-01]. Dostupné z: <https://www.br-automation.com/en/about-us/press-room/advanced-system-diagnostics-via-the-web/>.
- Akvizice společnosti B&R společností ABB. *ABB* [online]. 4. 4. 2017 [cit. 2018-04-01]. Dostupné z: <http://www.abb.cz/cawp/seitp202/cfa3145a17f78ba4c12580f900367e6d.aspx>.
- Automation Runtime. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: <https://www.br-automation.com/en-gb/products/software/automation-runtime/>.
- Automation Studio. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: <https://www.br-automation.com/cs/produkty/software/automation-studio/>.
- B&R zkracuje dobu vývoje o 67 % prostřednictvím mapp Technology. *Automatizace v potravinářství* [online]. 29. 01. 2015. [cit. 2018-04-01]. Dostupné z: <http://www.prumyslovaautomatizace.com/b-r-zkracuje-dobu-vyvoje-o-67-prostrednictvim-mapp-technology>.
- BÍLEK, K. Produkty B&R – automatizace která má vtip. *Automa* [online]. 2005 [cit. 2018-04-02]. Dostupné z: http://automa.cz/cz/casopis-clanky/produkty-b-r-automatizace-ktera-ma-vtip-2005_03_30375_2006/.
- CANopen. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: <https://www.br-automation.com/en/products/networks-and-fieldbus-modules/canopen/>.
- HMI. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: <https://www.br-automation.com/en/products/hmi/>.
- CHALUPA, K. Čtvrtá průmyslová revoluce posílí konkurenceschopnost. *Moravské hospodářství* [online]. 9. 10. 2016 [cit. 2018-04-01]. Dostupné z: <http://moravskehospodarstvi.cz/article/ekonomika/ctvrta-prumyslova-revoluce-posili-konkurenceschopnost/>.
- I/O Systems. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: <https://www.br-automation.com/cs/produkty/io-systems/>.
- KOHOUT.L. Produkty. *Kohout* [online]. ©2008. [cit. 2018-04-08]. Dostupné z: <http://www.edumat.cz/produkty.php?produkt=krizovatka>.
- KOCHANÍČEK, L. Programovací jazyky pro PLC. *Elektrotechnika* [online]. 8. 4. 2010 [cit. 2018-04-01]. Dostupné z: <https://coptkm.cz/portal/?doc=3905>.
- Mapp Technology. *B&R Help Explorer*. ©2018. [cit. 2018-04-01].
- OpenSAFETY. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: <https://www.br-automation.com/en/technologies/opensafety/>.
- Powerlink. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: <https://www.br-automation.com/cs/technologie/powerlink/>.
- Produkty. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: <https://www.br-automation.com/cs/produkty/>.
- ReACTION Technology. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: <https://www.br-automation.com/en/technologies/reaction-technology/>.

- Redundance. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: <https://www.br-automation.com/cs/technologie/redundance/>.
- ŠMEJKAL, L.; MARTINÁSKOVÁ M. 1999. *PLC a automatizace*. 1. vyd. Praha: BEN – technická literatura. ISBN 80-860-5658-9.
- Taking sorting to a whole new level. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: <https://www.br-automation.com/en/about-us/customer-magazine/taking-sorting-to-a-whole-new-level/>.
- VOJÁČEK, A. Programovací režimy pro PLC dle IEC 61131-3 (CoDeSys). *Automatizace* [online]. 3. 3. 2011 [cit. 2018-04-01]. Dostupné z: <https://automatizace.hw.cz//programovaci-rezimy-pro-plc-dle-iec-611313-codesys>.
- Výukový model mechatroniky. *Stroza* [online]. [2000?]. [cit. 2018-04-08]. Dostupné z: <http://www.stroza.cz/25150-c4-vyukovy-model-mechatroniky.php>.
- X20AT2222. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: http://www.br-automation.com/downloads_br_productcatalogue/BRP4440000000000000535940/X20AT2222-ENG_V3.16.pdf.
- X20CP041x and X20CP048x. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: http://www.br-automation.com/downloads_br_productcatalogue/BRP4440000000000000512505/X20CP04xx-ENG_V1.02.pdf.
- X20DI9371. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: http://www.br-automation.com/downloads_br_productcatalogue/BRP4440000000000000506749/X20DI9371-ENG_V3.15.pdf.
- X20DO6322. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: http://www.br-automation.com/downloads_br_productcatalogue/BRP4440000000000000504945/X20DO6322-ENG_V3.23.pdf.
- X20PS9600. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: http://www.br-automation.com/downloads_br_productcatalogue/BRP4440000000000000514126/X20PS9600-ENG_V1.02.pdf.
- X20 Systems. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: <https://www.br-automation.com/cs/produkty/io-systems/x20-system/>.
- X67 Systems. *B&R Perfection in Automation* [online]. ©2018 [cit. 2018-04-01]. Dostupné z: <https://www.br-automation.com/cs/produkty/io-systems/x67-system/>.
- ZEMANOVÁ, B. 2010. *PLC – programovatelné automaty: Automatizační technika*. [online]. [cit. 2018-04-01]. Dostupné z: <http://slideplayer.cz/slide/2535859/#>.
- ZEZULKA, F.; HYNČICA, O. Průmyslový Ethernet VIII: Ethernet Powerlink, Profinet. *Automa* [online]. 2008 [cit. 2018-04-02]. Dostupné z: http://automa.cz/cz/casopis-clanky/prumyslovy-ethernet-viii-ethernet-powerlink-profinet-2008_05_37288_6341/.

PŘÍLOHY

A - CD

Příloha k bakalářské práci
AUTOMATIZACE POMOCÍ B&R LOGICKÝCH AUTOMATŮ
Petr Čepelák

CD

Obsah

- 1 Text bakalářské práce ve formátu PDF
- 2 Úplné zdrojové kódy sestavených aplikací
- 3 Doplnující informace