

Knihovna funkčních bloků umožňujících simulovat reálné modely z laboratoře PLC

Patrik Ptáček

Bakalářská práce
2022



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky

Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Patrik Ptáček**
Osobní číslo: **A18616**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Informační a řídicí technologie**
Forma studia: **Kombinovaná**
Téma práce: **Knihovna funkčních bloků umožňujících simulovat reálné modely z laboratoře PLC**
Téma práce anglicky: **A Function Block Library Enabling Simulation of Real Models from the PLC Laboratory**

Zásady pro vypracování

1. Popište PLC Tecomat Foxtrot z hlediska programování i hardwarových možností.
2. Popište vedoucím vybrané moduly –jejich funkce a způsob připojení k řídicímu PLC.
3. Vytvořte knihovnu modulů pro simulaci výše popsaných modelů, která bude umožňovat ladit program pro řízení reálných modelů bez jejich fyzického připojení.
4. Vypracujte vzorové úlohy pro řízení všech zadaných modelů.
5. Tvorbu programu s využitím vytvořených knihoven popište pro potřebu následného využití ve výuce.

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

Seznam doporučené literatury:

1. ŠMEJKAL, L.: Esperanto programátorů PLC: programování podle normy IEC/EN 61131-3, (19 částí), AUTOMA, 08/2011 –03/2014, on-line: <http://tecoacademy.cz/wp-content/uploads/2017/04/Esperanto-final.pdf>
2. TECOMAT: Programování PLC podle normy IEC61131-3 v prostředí Mosaic, 2007, on-line: https://www.tecomat.cz/modules/DownloadManager/download.php?alias=txv00321_01_mosaic_progiec_cz
3. ŠMEJKAL, L.: PLC a automatizace 1. a 2. díl, BEN – technická literatura, Praha, 2006.
4. PETRUZELLA, F. D. : Programmable logic controllers. 4th ed. ISBN 978-0-07-351088-0. McGraw-Hill.,New York, 2010.
5. BOLTON, W. Programmable Logic Controllers – fourth edition, Elsevier India Private Limited, Oxford, UK, 2006, ISBN 13: 978-0-7506-8112-4, ISBN 10: 0-7506-8112-8.
6. REHG, J. A., SARTORI, G. Programmable Logic Controllers, Prentice Hall, 2006, ISBN-13: 9780134328812, ISBN: 0134328817.

Vedoucí bakalářské práce: **Ing. Tomáš Sysala, Ph.D.**
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: **15. ledna 2022**
Termín odevzdání bakalářské práce: **20. května 2022**

doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan



prof. Ing. Vladimír Vašek, CSc. v.r.
ředitel ústavu

Ve Zlíně dne 15. ledna 2022

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

Patrik Ptáček v.r.
podpis studenta

ABSTRAKT

Tato bakalářská práce má praktický charakter. Hlavním záměrem práce je tvorba simulačních knihoven pro výuku studentů v oblasti programování PLC. Obsahem teoretické části práce je seznámení s vývojovým prostředím použitým při programování a představení hardwaru využívaného v laboratoři. V praktické části práce jsou popsány reálné modely využívané v laboratoři. Následně jsou navrženy simulační knihovny spolu s podrobným popisem jejich výukového využití.

Klíčová slova: PLC, automatizace, Mosaic, Tecomat, EDU-mod.

ABSTRACT

This bachelor thesis is practical in nature. The main purpose of the thesis is to create simulation libraries for teaching students PLC programming. The content of the theoretical part of the thesis is an introduction to the development environment used in programming and an introduction to the hardware used in the laboratory. The practical part of the thesis describes the real models used in the laboratory. Subsequently, simulation libraries are proposed along with a detailed description of their educational use.

Keywords: PLC, automatization, Mosaic, Tecomat, EDU-mod.

Rád bych poděkoval svému vedoucímu Ing. Tomáši Sysalovi, Ph.D. za odborné vedení, konzultace a rady v průběhu tvorby mé bakalářské práce.

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	11
1 SOFTWAREVÉ PROSTŘEDKY	12
1.1 NORMA IEC EN-61131-3.....	12
1.2 ZÁKLADNÍ INFORMACE O PROSTŘEDÍ MOSAIC	15
1.3 NÁSTROJ WEBMAKER.....	20
2 HARDWAROVÉ PROSTŘEDKY	26
2.1 PLC TECOMAT CP-1005.....	27
2.2 PLC TECOMAT CP-2000.....	28
3 POPIS VYBRANÝCH MODELŮ	30
3.1 HYDRAULICKÁ POSUVNÁ JEDNOTKA	30
3.2 KŘÍŽOVATKA	31
3.3 MÍSICÍ JEDNOTKA	32
3.4 PRAČKA.....	33
II PRAKTICKÁ ČÁST.....	35
4 NÁVRH SIMULAČNÍCH MODULŮ	36
4.1 HYDRAULICKÁ POSUVNÁ JEDNOTKA	36
4.2 KŘÍŽOVATKA	38
4.3 MÍSICÍ JEDNOTKA	38
4.4 PRAČKA.....	40
5 VZOROVÉ ÚLOHY PRO ŘÍZENÍ ZADANÝCH MODULŮ	42
5.1 HYDRAULICKÁ POSUVNÁ JEDNOTKA	42
5.2 KŘÍŽOVATKA	42
5.3 MÍSICÍ JEDNOTKA	43
5.4 PRAČKA.....	44
5.5 VYTVOŘENÍ KNIHOVEN.....	45
6 VYUŽITÍ KNIHOVEN VE VÝUCE.....	49
6.1 PROSTÉ POUŽITÍ KNIHOVNY	49
6.2 OTEVŘENÍ ARCHIVOVANÉHO PROJEKTU.....	55
ZÁVĚR	57
SEZNAM POUŽITÉ LITERATURY	58
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	61
SEZNAM OBRÁZKŮ.....	63

SEZNAM TABULEK	65
SEZNAM PŘÍLOH	66

ÚVOD

Průmyslová automatizace je čím dál zajímavějším oborem a její využití se stále rozšiřuje. Je nesmírně populární jak ve specializovaných montovnách a pásových výrobnách, tak i při ovládání různých objektů, jako jsou třeba čerpací stanice vody, čističky vod a jiné.

Důkazem nám jsou předchozí průmyslové revoluce. Ty zásadním způsobem změnilы pohled na způsob výroby pomocí nových technologií, které postupně přicházely. Přes industrializaci, elektrifikaci, digitalizaci a zlepšení komunikačních prostředků jsme se dostali do dnešní doby, kdy pozorujeme zřetelný trend na vzestupu, a tím je automatizace. Z historie jasně vidíme, že zmíněné inovace dokázaly nesmírně zvýšit životní úroveň každé země, která je začala využívat. Postupně opadávala fyzická náročnost mnohých povolání, bylo možno vyprodukovat více výrobků a plodin. To mělo přímý vliv na snížení stresu obyvatel, prodloužila se délka jejich života a výrazně se zvýšila populace dané země. Od té doby je také možné sledovat znatelné zvýšení ekonomického růstu v dosud nebývalé míře.

Od prvních průmyslových revolucí však technologie hodně pokročily a velkou spoustu opakujících se dějů nemusí provádět člověk, ale stroje ovládané programovatelným logickým automatem neboli PLC. Ten se využívá pro automatizaci procesů v reálném čase. Je pro něj charakteristické, že se program neustále opakuje a vykonává v tzv. cyklech. Na rozdíl od stolních počítačů jsou periferie PLC uzpůsobeny pro připojení na technologické procesy. Majoritu portů PLC tvoří digitální vstupy – DI, digitální výstupy – DO, které pracují s dvoustavovými proměnnými. Analogové vstupy – AI a výstupy – AO jsou zase schopny pracovat se spojitým signálem. K PLC je možno připnout i další periferní jednotky, jež doplňují potřebné funkcionality, které základní PLC nemá. Jsou to na příklad funkční moduly pro sběr a přenos dat, připojení k LTE a další speciální moduly, které poskytuje výrobce daného systému a hardwaru. [6]

Přitom jedny z prvních PLC sloužily jako náhrada reléových automatů. S postupem času však začaly být PLC více dostupnými. Navíc se zvyšující se náročností ovládaných procesů začaly dávat i ekonomický smysl. Dokázaly totiž postupně pracovat s čísly s pevnou desetinnou řádkou a následně i plovoucí, zpracovávat analogové hodnoty, komunikovat s jinými systémy atd. Jsou to navíc velmi odolná zařízení – jejich životnost se dnes pohybuje kolem 20 let – a jsou uzpůsobeny pro chod v horších podmínkách.

Nyní jsou programovatelné automaty nedílnou součástí mnohých výrobních a řídicích systémů. S vývojem technologií jsou tyto procesy rovněž čím dál složitější. Tím pádem rostou i nároky na programátory a studenty programování. I díky vysokým pořizovacím nákladům jsou však možnosti studentů pracovat s reálným PLC značně omezené: ve školních laboratořích jich mnoho není a pořízení vlastního programovatelného automatu i s několika moduly je pro studenta nemyslitelné. Proto je skvělé, že mnoho dodavatelů programovatelných automatů nabízí ve svých softwarech také možnost využít simulovaného PLC. Studenti se tak mohou naučit programovat PLC bez nutnosti prezence reálného automatu.

Samotná možnost simulovaného PLC je však jen částečným ulehčením výuky. Od automatu totiž nedostaneme zpětnou vazbu, a nedozvíme se tedy okamžitě, zdali testovaný kód funguje správně. Rozhodl jsem se proto vytvořit simulační moduly v podobě knihoven, které budou studentovi poskytovat potřebnou zpětnou vazbu. Tyto moduly mu umožní pracovat tak, jako by přehrával svůj program na reálném PLC, jež je připojeno ke všem potřebným čidlům, které vysílají onu zpětnou vazbu. Student by se tak mohl soustředit pouze na svůj program ovládající daný přístroj, a nemusel se tudíž zatěžovat tvorbou testovacího programu.

V teoretické části práce jsou popisovány převážně způsoby a technologie využití při tvorbě praktické části. Zároveň se však také snažím rozšířit studentům obzory ukázkami dalších způsobů využití prostředí Mosaic. Následuje představení hardwaru, se kterým se setkávají v laboratoři a jemuž se věnuje praktická část. K němu příkládám i popis novějšího modelu, jenž je určen pro prezentaci nových technologií použitých na hardwaru od firmy Teco a. s. Praktická část následně popisuje funkce vytvořených knihoven a příklady programů, které využívají tyto knihovny. V poslední části je popsán podrobný návod, jak je možné tyto knihovny využít ve výuce a připravit si prostředí Mosaic pro jejich správné fungování.

I. TEORETICKÁ ČÁST

1 SOFTWAREVÉ PROSTŘEDKY

Program **Mosaic** je vývojové prostředí umožňující ladění a tvorbu programů určených pro programovatelné logické automaty (Programmable Logic Controller, PLC) TECOMAT® a TECOREG® od firmy Teco a. s. Kolín. Program **Mosaic** je používán již od roku 2000. Vývoj prostředí probíhá ve shodě s mezinárodní normou IEC EN-61131-3, definující strukturu programů a programovací jazyky pro PLC. [3]

1.1 Norma IEC EN-61131-3

Norma IEC EN-61131-3 neslouží jako soubor pravidel, které by mělo každé vývojové prostředí v rámci programovacího jazyka dodržovat. Jde spíše o návod či doporučení, jak by programovací jazyky ve vývojovém prostředí měly vypadat. Proto také norma obsahuje tabulku o 62 rysech či vlastnostech, do nichž jednotlivá vývojová prostředí zapisují, které body normy splňují a které ne. Proto nemusí být kód přenositelný z jednoho vývojového prostředí do druhého, neboť každé může splňovat jiné body normy, a tudíž bude potřeba kód upravit pro kompatibilitu s jiným prostředím. I přesto je norma uznávána většinou výrobců PLC po celém světě, a stala se tak v posledních letech jedinou celosvětově uznávanou normou. Tato norma je tedy zásadní pro zjednodušení programování PLC, navíc ušetří mnoho peněz, neboť programátor nemusí znát různé programovací jazyky, ale stačí jeden pro všechna PLC. [2]

Hlavní dvě skupiny programovacích jazyků pro PLC jsou **textové** a **grafické**. Ty se dají vzájemně kombinovat pro tvorbu přehlednějších programů. [5]

Textové programovací jazyky:

IL – Instruction List – jazyk seznamu funkcí

- Jde o evropský protějšek Ladder diagramu. Tento textový jazyk se nejvíce podobá assembleru. [5]
- Instrukce se zde vykonávají postupně shora dolů. [5]

- Výsledek instrukcí je ukládán na vrchol zásobníku, kde s ním můžeme provádět další operace. [5]

Table 1
List of basic IL commands

Instruction	Modifier	Operand	Description
LD	N	variable, constant	loads operand
ST	N	variable, constant	stores operand
S		variable	sets operand to <i>true</i>
R		variable	sets operand to <i>false</i>
NOT			Boolean negation
AND	N	variable, constant	Boolean AND
OR	N	variable, constant	Boolean OR
XOR	N	variable, constant	Boolean XOR
ADD		variable, constant	addition
SUB		variable, constant	subtraction
MUL		variable, constant	multiplication
DIV		variable, constant	integer division
GT		variable, constant	comparison greater than
GE		variable, constant	comparison greater equal
LT		variable, constant	comparison less than
LE		variable, constant	comparison less equal
EQ		variable, constant	comparison equal
NE		variable, constant	comparison unequal
JMP	N, C	label	jump to label
RET			return from function (block)

Obrázek 1: Příkazy používané v programovacím jazyce IL [15]

ST – Structured Text – jazyk strukturovaného jazyka

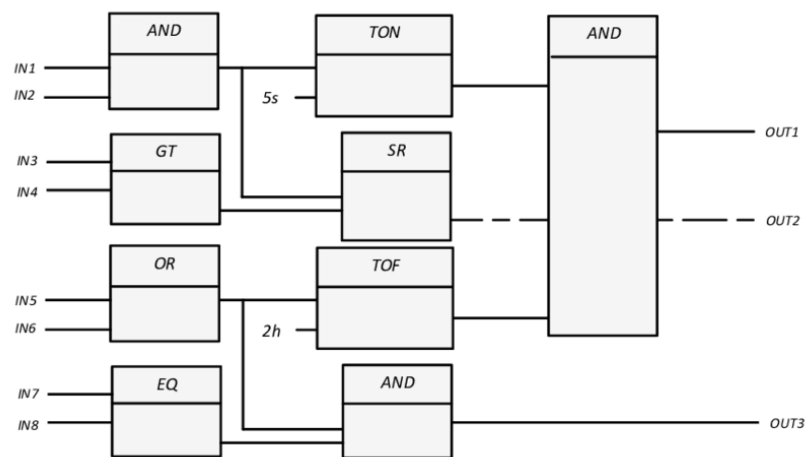
- Velmi výkonný a často používaný vyšší programovací jazyk; jeho základy pocházejí ze známých programovacích jazyků, jako jsou Ada, C a Pascal. [5]
- Obsahuje mnohé z podstatných prvků moderních vysokoúrovňových programovacích jazyků, jako jsou možnosti větvení programu pomocí příkazů (**IF – THEN – ELSE a CASE OF**) či iterační smyčky (**FOR, REPEAT a WHILE**). Zmíněné prvky mohou být i vnořovány. [5]
- Jazyk ST je výborným nástrojem pro definování funkčních bloků o větší komplexitě. Je přehledný a jednoduchý na pochopení jak během tvorby programu, tak i během analýzy funkcí v programu, který napsal někdo jiný. [5]

- Nevýhodami této vysoké abstrakce mohou být pomalejší výpočetní rychlost a délka programu. Operátory mají také jinou výši priority a programátor si musí pohlídat, jaký výsledek dostane. [1]

Grafické programovací jazyky:

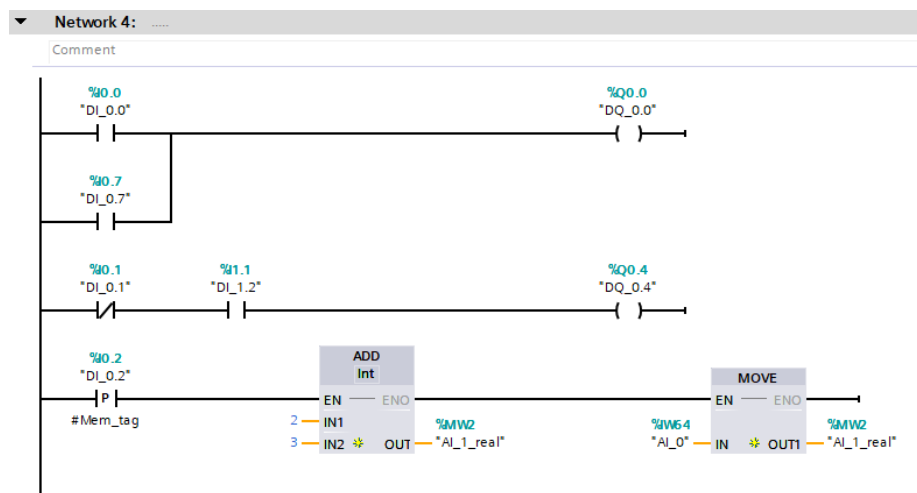
FBD – Function Block Diagram – jazyk funkčního blokového schématu

- Je velmi blízký procesnímu průmyslu. Popisuje chování programů, funkčních bloků a funkcí jako soubor vzájemně propojených grafických bloků. [5]
- Zobrazení se podobá diagramům elektronických obvodů. [5]



Obrázek 2: Ukázka jazyku funkčního blokového schématu [16]

LD – Ladder diagram – jazyk příčkového diagramu. Jeho původ sahá do USA. Představuje grafické zobrazení reléové logiky. [5]



Obrázek 3: Ukázka programu v jazyku příčkového diagramu [17]

Dále je možné v programu Mosaic používat editor **CFC** (Continuous Function Chart), který je určen pro grafické kreslení plovoucích schémat. [3]

Výše zmiňované programovací jazyky se dají vzájemně libovolně kombinovat. Výběr programovacího jazyka je čistě na programátorovi, který by jej měl volit podle obtížnosti a typu úlohy. Velice populární je psaní složitějších funkcí pomocí jazyka ST. Funkci napsanou v ST je přehledné si zavolat přes FBD a následně jednotlivé funkční bloky spojovat pomocí LD. Součástí vývojového prostředí MOSAIC je také řada nástrojů, které usnadňují ladění a vývoj aplikací. Jedním z nich je nástroj WebMaker, který se používá při tvorbě vizualizace pro námi napsaný program.

Programy tvořené ve vývojovém prostředí MOSAIC se skládají z elementů zvaných **programové organizační jednotky (Program Organisation Unit, POU)**. Jsou to tyto jednotky: **funkce, funkční blok a program**. [1]

Funkce (FUN) je nejjednodušší z POU. Je volána se vstupními parametry žádaného datového typu. Neobsahuje však vlastní statické proměnné, a nemá tedy vlastní vnitřní paměť. Vstupů do funkce může být více, výsledek však vrací pouze jeden. Standardní funkce jsou třeba SQRT, SIN apod. Programátor si zároveň dokáže tvořit vlastní funkce, které nejsou předem definované ve vývojovém prostředí. [1]

Funkční blok (FB) má na rozdíl od funkcí vlastní paměť a má rovněž vlastní statické proměnné, jejichž hodnotu si pamatuje. Obsahuje data i algoritmy. Oproti funkci dokáže vracet více výsledků, tedy výstupních hodnot. Funkční bloky mohou představovat integrované obvody hardwarového řešení specializované řídicí funkce. [1]

Program (PROG) v hierarchii POU představuje „main program“, a tedy vrcholnou programovou jednotku. Více programům přiřazuje prioritu **centrální jednotka**, ve které jsou dané programy obsaženy. Je to v podstatě směsice funkčních bloků a funkcí. [1]

1.2 Základní informace o prostředí Mosaic

Program Mosaic při instalaci obsahuje všechny nástroje, které jsou v danou chvíli k dispozici. Po nainstalování programu Mosaic je možné zadat HW klíč, kterým lze odemknout plnou verzi tohoto programu. Bez zadání HW klíče funguje Mosaic ve verzi **Lite**, jež se liší od plné verze pouze tím, že nemůžeme definovat větší počet I/O modulů. Verze Lite je tedy naprosto dostačující výukový program, ve kterém můžeme programovat nejmenší PLC z řady TECOMAT. Několikrát ročně jsou vydávány nové verze Mosaicu, jež

převážně opravují předchozí nedostatky a rozšiřují funkční možnosti. Mosaic je možné nainstalovat na libovolné množství zařízení. Teco a. s. sleduje zejména možnou zpětnou kompatibilitu, aby dané PLC byly použitelné i desítky let. Mosaic lze stáhnout i v češtině a jakýkoli upgrade dostupné verze je poskytován zdarma. Zároveň je možné Mosaic nainstalovat na každé zařízení, které pracuje v systému Windows 2000, XP, 7, 8, 10. [3]

V prostředí Mosaic je nutno při vytváření **nového projektu** sestavit i novou **skupinu projektů** nebo jej přidat do již existující skupiny, pokud je takováto skupina již vytvořena. Projekty ve stejné skupině mezi sebou mohou komunikovat a jeví se tak jako jeden celek. Každý jeden projekt však má svoji složku, která obsahuje všechny zdrojové a pracovní soubory. [3]

Samotné programování v prostředí Mosaic je velmi **intuitivní** a je podobné jiným vývojovým prostředím. Pokud tedy uživatel původně programoval v Tia Portal pro programování PLC od společnosti Siemens, nebude mít nejspíš větší problémy s orientací i zde. Podobně na tom také budou ti, kteří přecházejí od vyšších programovacích jazyků, jako jsou například C nebo C++. Velké množství kompilátorů je opravdu velmi podobné prostředí v Mosaicu.

V horní části najdeme hlavní nabídku, tedy hlavní možnosti nastavení prostředí Mosaic, jako jsou volba uložení projektů, spouštění simulace a další. Dominantní část okna tvoří hlavní panel pro zápis kódu a tvorbu výkonné části projektu. Zde si lze otevírat okna editorů, jejichž záložky jsou hned nad těmito editory. V levé části se nacházejí hlavně okna POU, volba skupiny projektů, soubory projektu, seznam otevřených souborů a **IEC manažér**. V dolní části najdeme výsledky ladění projektu. Během připojení k PLC nebo při zapnutí simulaci sledujeme na pravé straně stavy jednotlivých proměnných. [3]

Všechny zdrojové kódy lze převést do textové podoby. Tím je myšleno, že projekt vytvořený pomocí grafických programovacích jazyků je možné převést do takové textové podoby, v níž lze pokračovat ve tvorbě pomocí jazyka ST. [3]

Nástroj **IEC manažér** nám však dovoluje používat v jednom projektu jakoukoli kombinaci programovacích jazyků. A to jak grafických, tak i textových. Dále pomocí něj definujeme proměnné, datové typy, funkce, programové jednotky a funkční bloky. Dokonce umožňuje deklaraci vlastních knihoven. [3]

PIDMaker je zase vizuální nadstavba nad PIDMA a PID a slouží ke snadnému ladění a implementaci regulačních algoritmů. Při otevření generuje část kódu s PID regulátory,

zároveň usnadňuje nastavení parametrů regulátorů. Rovněž je jednoduché sledovat průběh regulace během chodu programu. [3] [8]

PanelMaker generuje část kódu pro obsluhu textových panelů HMI. Je hlavně určen pro definování obsahu obrazovek těchto textových operátorských panelů. Avšak funguje pouze tehdy, pokud je v manažeru projektu textový projekt propojen s daným komunikačním kanálem ve správném režimu. V nastavení panelu musí být rovněž povoleno použití tohoto nástroje. [3]

Grafický PanelMaker slouží k definování obsahu obrazovek pro grafické operátorské panely a automaticky generuje soubory s popisy obrazovek pro správu a obsluhu grafických panelů HMI. Za účelem použití je také potřeba v projektu manageru vybrat grafický panel. [3]

POU Inspektor slouží jako pomocník při ladění programů v režimu RUN. Kompilátor s jeho pomocí dokáže zobrazovat real-time hodnoty a stavy proměnných. Dále umožňuje používat ladící body. [3]

Pomocí nástroje **WebMaker** dokážeme tvořit XML stránky pro webový prohlížeč. Je hojně využíván zejména z důvodu snadné vizualizace námi napsaného projektu. [3]

PanelSim je simulátor operátorských panelů, který dovoluje testovat vytvořené dialogy PanelMakerem. Funguje jak se simulovaným, tak i reálným PLC, a to bez nutnosti připojení skutečného panelu. [3]

GraphMaker dokáže vykreslit průběh až šestnácti proměnných ve formě spojitého grafu, a to jak binární, tak i spojitě proměnné. Má dva režimy – jedním je zobrazování právě probíhajících událostí, druhý umožňuje offline zobrazení událostí, které se staly v minulosti. Oba režimy mají minimální vzorkovací periodu nastavenou na délku jednoho cyklu PLC. Online režim však zobrazuje maximálně 3600 sec, zatímco offline režim je schopen vykreslit stavy proměnných až 65 535 sec zpětně. [3]

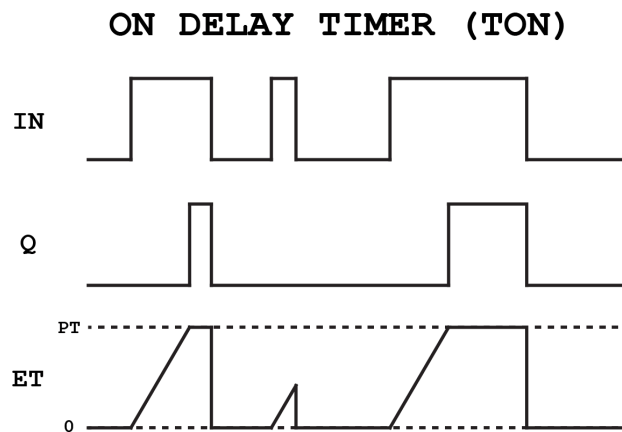
Mosaic umožňuje výběr typu PLC, včetně přídatných modulů a nastavení jejich parametrů. Danou konfiguraci je také možné nahrát z připojeného PLC. [3]

Sít' PLC – pomocí ní lze graficky vytvořit vazby mezi PLC, popsat síť PLC a další objekty. Jedná se zejména o nadřazené PC, huby, zařízení na CanOpen, zobrazovací panely, switche, Profibus apod. [3]

Pro komunikaci s měřicími přístroji, měniči apod. je velmi využíván protokol MODBUS. Je to komunikační protokol a pracuje na aplikační vrstvě OSI modelu nad vrstvou TCP/IP. Poskytuje komunikaci „client/server“ po síti mezi jednotlivými zařízeními v rámci různých typů sběrnic a sítí. Funguje na principu dotazu od zařízení „master“ a odpovědi od zařízení „server“. Možné je komunikovat i v režimu „master/master“. [9]

Project Manager – umožňuje komfortní a přehlednou správu projektu, ukládání, zálohování apod. Přehledně zobrazuje strom projektu a všechny jeho komponenty. [3]

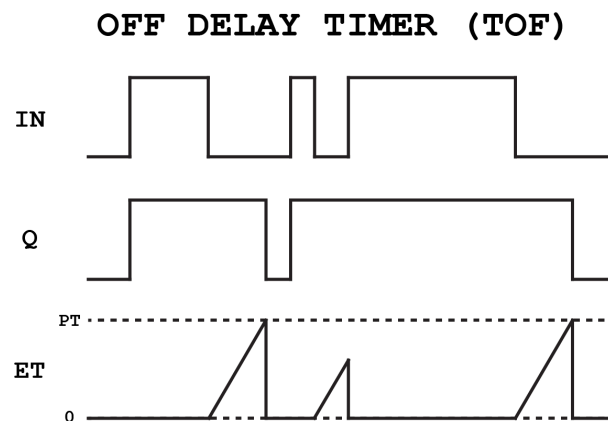
asovače – je jich více druhů a každý se chová trochu jinak. První z nich je časovač **TON (on delay timer)**. Jeho chod je popsán na následujícím obrázku.



Obrázek 4: Schéma průběhu časovače TON
[14]

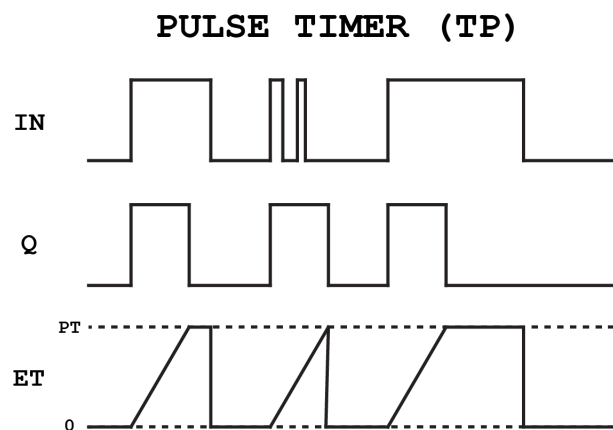
Spustí se po kladné náběžné hraně, kdy je tedy splněna podmínka IN. Výstup (Q) je však kladný až po uběhnutí zpoždění (ET). Poté je výstup kladný po celou dobu, kdy je kladný i vstup. Pokud je vstup kladný kratší dobu než odchylka, výstup kladný nebude. [7] [14]

Další je časovač **TOF (off delay timer)**. Jeho chod je popsán na následujícím obrázku.



Obrázek 6 Schéma průběhu časovače TOF
[14]

Výstup (Q) je kladný vždy po kladné náběžné hraně a kladný je i po dobu zpoždění (ET) po nesplnění podmínek vstupu (IN). Poslední je časovač **TP (pulse timer)**. Jeho chod je popsán na následujícím obrázku. Výstup (Q) je kladný hned po kladné náběžné hraně a je kladný po celou dobu zpoždění (ET). Což však znamená, že impuls nelze přerušit shozením vstupní podmínky (IN). [7] [14]



Obrázek 5 Schéma průběhu časovače TP [14]

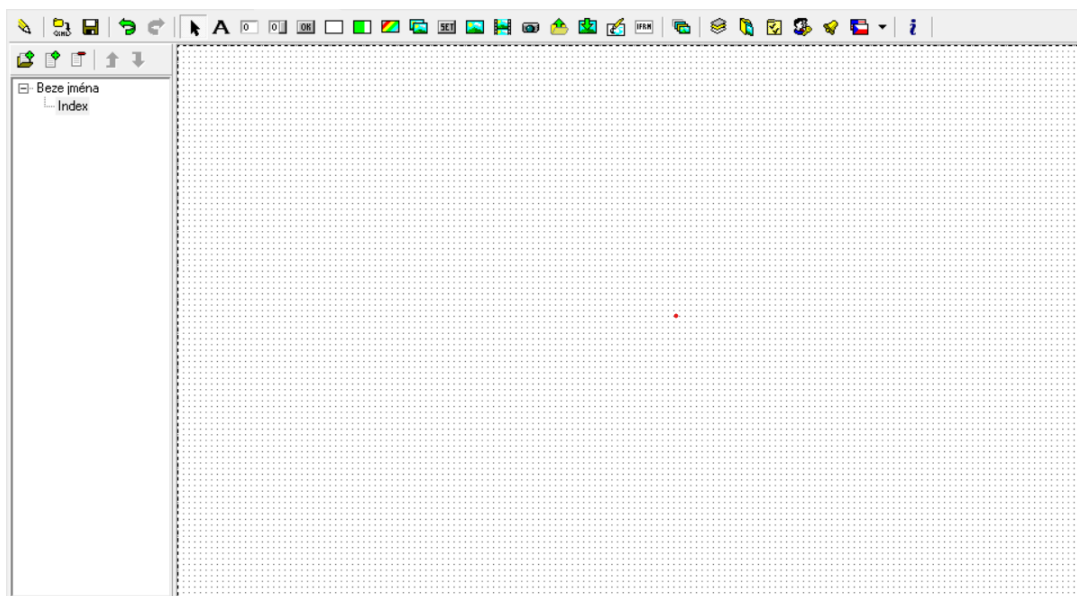
Mosaic umožňuje u některých modelů online úpravu programu PLC i úpravu či výměněnu I/O modulů za chodu programu bez potřeby stopu. Zároveň je přepnutí mezi starým a novým programem otázka desetininy doby potřebné pro zpracování programu. Komunikace mezi PC a PLC se dá zprostředkovat přes následující druhy spojení: [3]

- Ethernet (lokální síť, internet aj.) používaný hlavně pro propojení PLC s řídicím počítačem nebo dalšími PLC; [3]
- USB kabel (pouze lokální napojení); [3]
- COM sériový kanál (RS-232, RS485, RS422, modem apod.) – velké využití prostřednictvím protokolu MODBUS, přes který je jednoduché komunikovat se zařízeními, jako jsou čidla, měřicí přístroje, měniče atd. [9]
- síť CAN (sériová provozní datová sběrnice); [3]
- GSM modem; [3]
- Wifi. [3]

1.3 Nástroj WebMaker

Součástí Mosaicu je i nástroj WebMaker. Ten se využívá zejména při vytváření webových stránek s webovým serverem systémů Tecomat. Zároveň se dá použít jako nástroj pro vizualizaci projektů. Bez webového serveru je rovněž možno tento nástroj využít, pokud byla při programování dodržována norma IEC 61113. [4]

WebMaker je možno spustit v Mosaicu buď z hlavního menu Nástroje > WebMaker, nebo použitím ikony z nástrojové lišty. Ihned po jeho spuštění se otevře okno WebMakeru s editační plochou. [4]

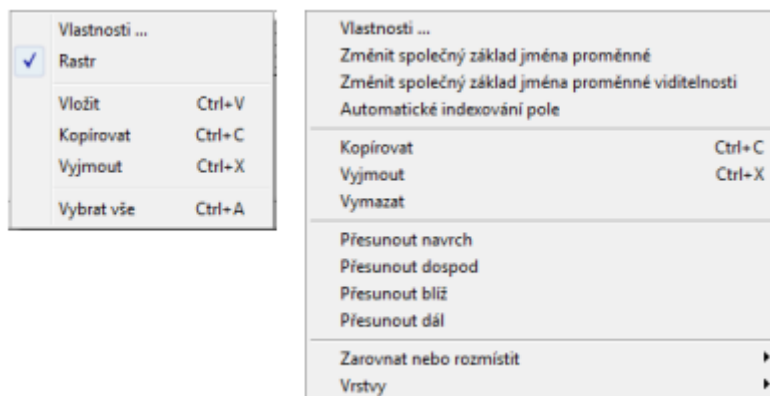


Obrázek 7: Pracovní plocha WebMakeru [18]]

- **Tlačítko OK** – slouží jako tlačítko pro pole, která nemají vlastní tlačítko. Pokud je tlačítko na webové stránce, dokáže odeslat hodnoty proměnných ze zadávacího pole do PLC. [4]
- **Obdélník** – vloží jednobarevný objekt obdélníkového tvaru, který slouží pro rozčlenění plochy. S nastavením podmíněné viditelnosti je možno objekt využít i pro signalizaci stavu proměnných. [4]
- **Částečně vyplněný obdélník** – jako obdélník, jen na grafických panelech 18/28 lze zobrazit sloupec, jež je stínován gradientem. [4]
- **Obdélník s proměnnou barvou** – obdélník s možností výběru barvy definovanou proměnnou se složkami RGB. [4]
- **Dvojice obrázků** – slouží k zobrazování hodnot typu BOOL pomocí dvoustavového obrázku. Jiný obrázek je znázorněn při hodnotě 1, následně odlišný při hodnotě 0. [4]
- **Tlačítko SET** – jednorázově stanoví přednastavenou hodnotu proměnné. Má přiřazeny dva obrázky. Jeden se zobrazuje při nastavení hodnoty tak, aby poskytoval zpětnou vazbu, druhý ve stavu klidovém. [4]
- **Filmový pás** – dokáže znázornit více obrázků. Ty zobrazuje podle toho, jaké nabývá hodnoty. Začíná hodnotou 0 a končí u hodnoty o jedna menší, než je počet obrázků. [4]
- **Obrázek** – vloží statický obrázek naší volby. [4]
- **Fotoaparát** – produkuje obraz pomocí IP kamery. Ten dokáže načítat periodicky. Na každé stránce však může být pouze jeden a není aktivní v simulaci prostředí Mosaic. [4]
- **Složka se šipkou** – slouží k odesílání souborů z webové stránky do PLC. [4]
- **Obrázek se šipkou** – dokáže otevřít obrázek, který je zadán proměnnou typu STRING. Takto umí znázornit jak obrázek uložený lokálně na paměťové kartě, tak i obrázky z jiných serverů na webu. [4]
- **Skupina objektů** – otevírá seznam objektů na stránce, kde je možno objekty vybírat a modifikovat. [4]

- **Objekty nad sebou** – Vrstvy – zobrazuje seznam všech vrstev. Každý objekt může být přidán do jedné z šestnácti vrstev a každá z vrstev může být zmrazena proti úpravám. Každá z vrstev také může být skryta. [4]
- **Složka s obrázky** – Správce obrázků – přes ni přidáváme do projektu fotografie, které můžeme následně použít na stránkách. [4]
- **Složka se zatržítky** – Společná nastavení – otevírá globální nastavení pro celý projekt. [4]
- **Zvonek** – Nastavení alarmů – nastaví stránky, které se mají zobrazit, pokud je řídicí proměnná nulová. [4]
- **Hlava s klíčem** – Nastavení přístupu – slouží k zadání MAC adres a hesel pro přístup bez přihlašování. [4]
- **Vlajky** – Nastavení jazyků – umožňuje import a export textů, dále definici jazykových mutací. [4]
- **Písmenko i** – poskytuje informace o verzi. [4]

V editační ploše, která je ze začátku prázdná, si zobrazíme námi zvolenou stránku a nastavíme její rozlišení. Pravé tlačítko myši nám kdekoli na editační ploše stránky zobrazí lokální menu. Jiná situace ovšem nastane, pokud pravé tlačítko myši zmáčkneme na nějakém objektu. V tu chvíli se opět objeví lokální menu daného objektu, a ne celé stránky. [4]



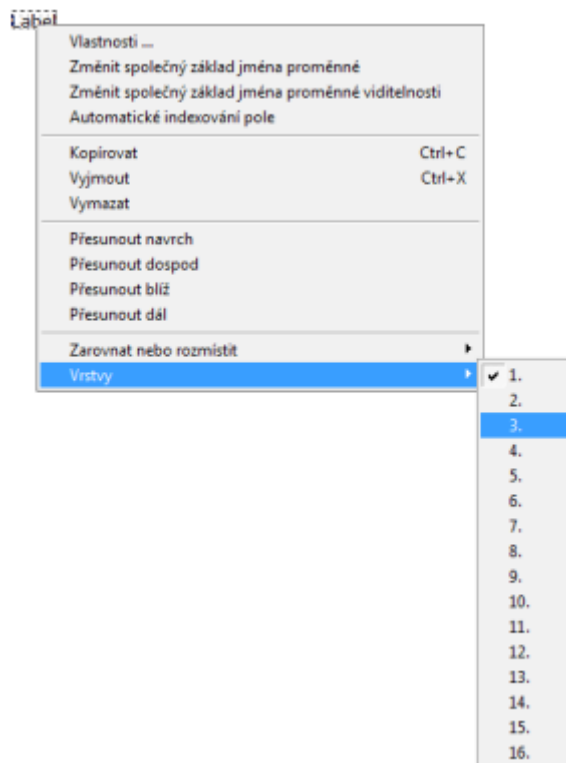
Obrázek 9: Lokální menu WebMakeru [18]

Součástí lokálního menu jsou následující položky.

- **Vlastnosti** – zobrazí vlastnosti stránky nebo objektu/objektů, pokud jsou nějaké objekty vybrány. Při výběru více objektů jsou změny provedeny na všech objektech stejného typu, jako u objektu, na kterém byly vlastnosti změněny. [4]

- **Rastr** – zapíná a vypíná rastr (rozlišení editační plochy). Rozměr rastru je možné nastavit ve Společných nastaveních. [4]
- **Změnit společný základ jména proměnné** – ve zvolených objektech vyvolá dialog ke změně společného jména proměnných. V dialogu je společný základ jména předvyplněn a nahrazen vložením nového textu. [4]
- **Změnit společný základ jména proměnné viditelnosti** – ve zvolených objektech vyvolá dialog ke změně společného jména proměnných řídicí viditelnost. Ten je předvyplněn v dialogu a poté nahrazen textem novým. [4]
- **Automatické indexování polí** – umožňuje automaticky číslovat objekty datového typu **pole**. [4]
- **Přesunout navrch** – přesune zvolený objekt při vykreslování před všechny ostatní. [4]
- **Přesunout dospod** – přesune zvolený objekt při vykreslování za všechny ostatní. [4]
- **Přesunout blíž** – přesune zvolený objekt při vykreslování před objekt, který měl být původně vykreslen před tímto objektem. [4]
- **Přesunout dál** – přesune zvolený objekt při vykreslování za objekt, který měl být původně vykreslen až po vykreslení tohoto objektu. [4]

- **Vrstvy** – označuje seznam 16 vrstev, které jsou označeny buď čísly od 1 do 16, nebo uživatelským popisem, pokud je nějaký k dispozici. Vrstvy mají u sebe podtržítka, pokud do dané vrstvy označené objekty patří. Všechny takto označené objekty jsou zařazeny do dané vrstvy při výběru vrstvy. V dialogu Vrstvy lze nastavit vlastnosti vrstev. Ten nalezneme v nástrojové liště. [4]



Obrázek 10: Vrstvy objektů ve WebMakeru [18]

2 HARDWAROVÉ PROSTŘEDKY

Předností programovatelných automatů TECOMAT FOXTROT je především jejich nesporná variabilita. Jsou to totiž malé kompaktní automaty s možností modulárního rozšíření, které jsou převážně určeny k řízení technologií v různých oblastech průmyslu a dalších odvětvích. Moduly jsou chráněny plastovým pouzdem, které je možné namontovat na U lištu normy ČSN EN 50022. Proto lze s moduly a CPU manipulovat bez nebezpečí poškození CMOS součástek citlivých na fyzické poškození. [19]

PLC může komunikovat s jiným PLC, několika jinými PLC, nadřazeným PC nebo ostatními zařízeními obvykle pomocí sériových přenosů. Podpora těchto přenosů pomocí sítě Ethernet nebo EPSNET je samozřejmostí a každá centrální jednotka obsahuje rozhraní Ethernet 10/100Mb, čímž umožňuje provoz více logických spojení najednou. Model CP-2000 dokáže komunikovat po GSM síti pomocí rozhraní PLC. [19]

Jeden sériový asynchronní kanál je na pevně obsazen rozhraním RS-232. Druhý je možno obsadit podle volby zákazníka různými fyzickými rozhraními (RS-232, RS-485, RS-422). Pomocí rozhraní RS-485 je možné na jedné úrovni sítě EPSNET připojit až 32 účastníků a délka této sériové linky činí až 1200 m. Je možné využít i jiné podporované protokoly a sběrnice, jako třeba CAN, PROFIBUS DP, MODBUS apod. Také je možné využít asynchronních komunikací přes univerzální přenosové kanály, které jsou ovládané přímo z PC. Je možné rozšířit PLC až o dva sériové kanály. To závisí na typu PLC. [19]

Pomocí sériové sběrnice je možné rozšiřovat periferní moduly decentralizovaně, a to hned u ovládaných technologií, díky čemuž je možné šetřit silovou kabeláž. [19]

Počítače standardu PC hrají ve spojení s PLC klíčovou roli. Programy psané pro PLC jsou totiž programované na PC. PC následně slouží k monitorování procesů, které PLC řídí. PLC umožňuje rovněž komunikaci s dalšími komponenty, které splňují požadavky sítě EPSNET. Jsou to hlavně jiné PLC nebo operátorské panely, které jsou taktéž velmi početně využívané k vizualizaci probíhajících procesů. [19]

Tyto PLC poskytují dva pohledy na řešení automatizace průmyslových procesů, a těmi jsou distribuované řešení a hierarchické řešení. Výhodou distribuovaného řešení je fakt, že se dá provádět „odspodu“, kdy se původně autonomní systémy postupně propojují a doplňují o úroveň nad nimi, která slouží jako řídicí prvek. Díky decentralizaci řízení je tato síť schopna autonomního řízení i při výpadku centra, samostatné části se dají postupně uvádět do provozu bez nutnosti zastavení celého systému. Stejně tak se celý systém nemusí

zastavovat při úpravách a laděních jednotlivých částí a zároveň dochází k úsporám při montáži na rozvaděčích a kabeláži. [19]

Všechny moduly jsou široké celým násobkem čísla M, které představuje hodnotu 17,5 mm. Neměl by tedy nastat problém s instalací modulů na U lištu v rozvaděčích či jističích a dalších prvků elektroinstalace. Základní modul CP-1005 je široký 6M, rozměry periférií pak dosahují hodnot 4M, 3M nebo 1M. Šířka základního modulu CP-200 odpovídá hodnotě 9M, rozměr periférií zůstává stejný jako u CP-1005. [19][21]

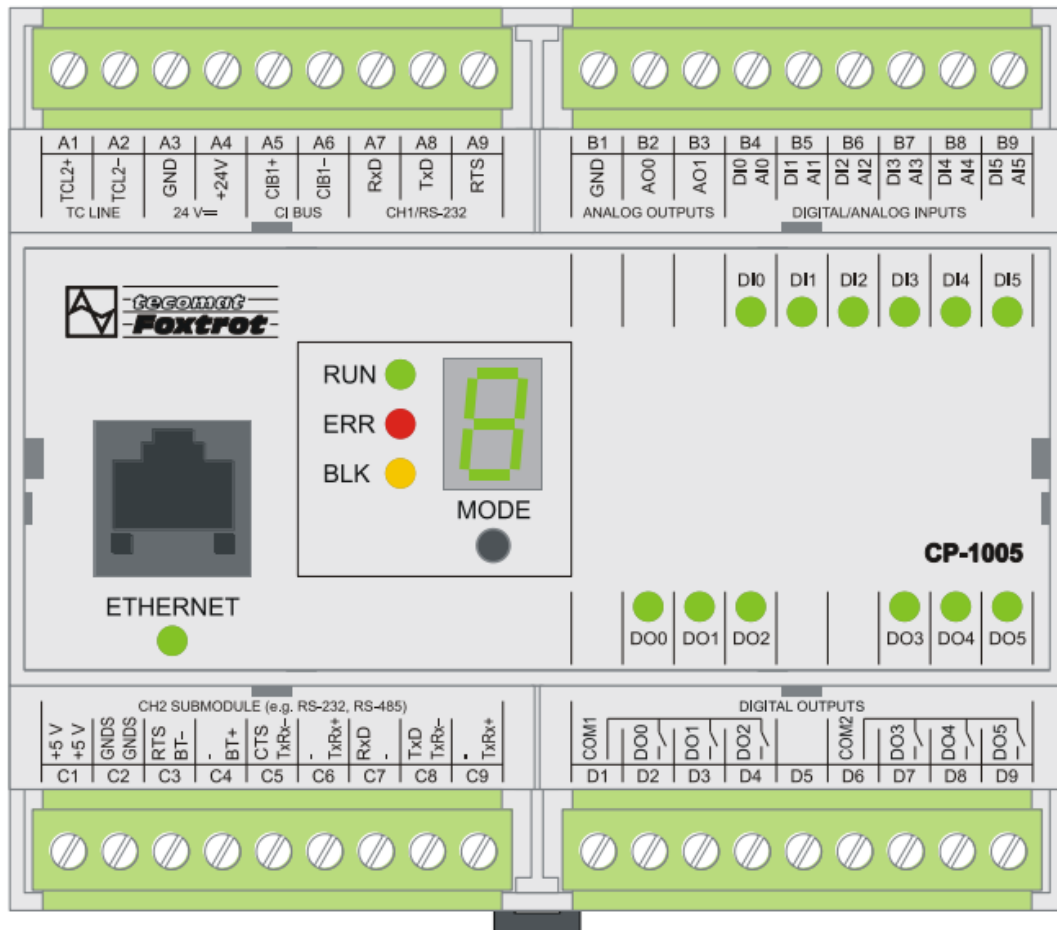
2.1 PLC TECOMAT CP-1005

Základní modely jsou složeny z více částí. Tou první je centrální jednotka s hlavním procesorem, rozhraním Ethernet, dvěma sériovými kanály a systémovou sběrnici TCL2 pro komunikaci s perifériemi. Druhá část je tvořena procesorem zařizujícím komunikaci na sběrnici CIB s moduly skupiny CFox. Třetí část je periferní a je tvořena deskou IR-1064. Ta pak komunikuje přes systémovou sběrnici ohledně jejich vstupů a výstupů.

Základní technické parametry PLC řady CP-1005:

- centrální jednotka řady K, 32bit RISC procesor
- 6x 250 V reléové výstupy
- 6x volitelný vstup – binární 24 V/analogový (měření napětí, proudu nebo odporu, rozlišení 16 bitů)
- 4x sériový kanál (CH1 – RS-232, zbylé jsou volitelné)
- 2x analogový výstup 0–10 V (10 bitů)
- 1x linka sběrnice TCL2 pro připojení periférií (10 I/O modulů, 4 operační panely)
- 1x linka sběrnice CIB (19,2 kbit/s)
- 1x rozhraní Ethernet 10/100 Mb přes konektor RJ-45
- 1x slot paměťové karty MMC/SD, SCDHC, vestavěný souborový systém FAT32
- možnost osazení modulů s binárními výstupy a vstupy
- možnost osazení modulů s dalšími dvěma sériovými kanály
- 192 + 64 KB paměti pro uživatelský program a tabulky
- 2 MB paměti pro zálohu celého projektu

- zálohování RAM a RTC (bez baterie 500 h, s baterií 20 000 h)
- 4096 časovačů a 8192 čítačů [19][20]



Obrázek 11: PLC Tecomat CP-1005 [18]

2.2 PLC TECOMAT CP-2000

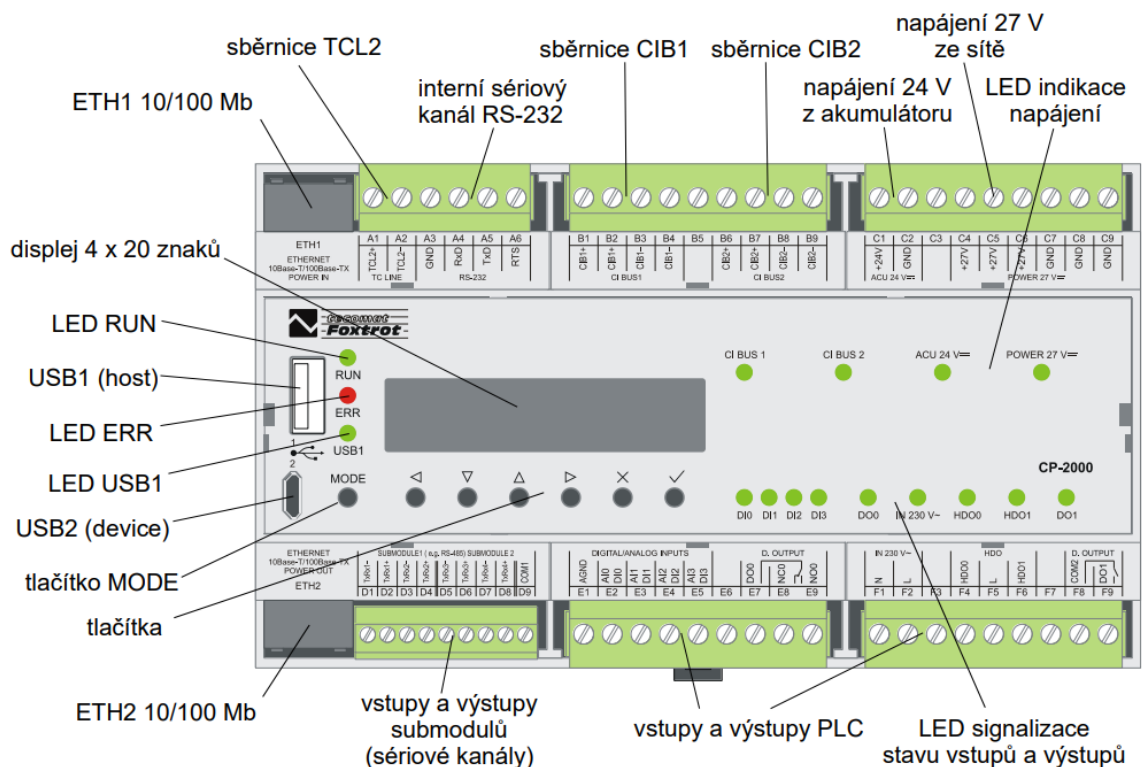
Centrální jednotka základního modulu obsahuje dvě nezávislá rozhraní Ethernet, jedno rozhraní USB device pro možnost připojení nadřízeného systému a jedno rozhraní USB host umožňující připojení externí paměti. K dispozici je také sériové rozhraní RS-232. [21]

Základní modul CP-200 je dodáván ve více variantách, které se od sebe liší kombinací znaků za tečkou v objednávce. Jde převážně o rozlišení podle velikosti paměti pro DataBox, dále o volitelně osazená rozšíření WLAN1 a LTE1 a integrovaný display o různé velikosti. [21]

Základní technické parametry PLC řady CP-2000:

- centrální jednotka řady I

- 4x volitelný vstup – binární bezpotenciálové/analogové (pasivní odporové snímače s rozlišením 12 bitů)
- 3x binární výstup 230 V AC
- 2x reléový výstup 250 V/3 A (z toho jeden přepínací)
- napájení ze zdroje 27 V DC a záložní akumulátor 24 V s diagnostikou
- OLED display 4 x 20 znaků, 6 uživatelských tlačítek
- 2x rozhraní Ethernet 10/100 Mb
- 1x rozhraní USB device a 1x rozhraní USB host
- 1x sériový kanál rozhraní RS-232
- 1x linka sběrnice TCL2
- 2x linka sběrnice CIB2x pozice pro osazení submodulů s dalšími sériovými kanály (až 4) nebo jinými zařízeními. [21]



Obrázek 12. PLC Tecomat CP-2000 [18]

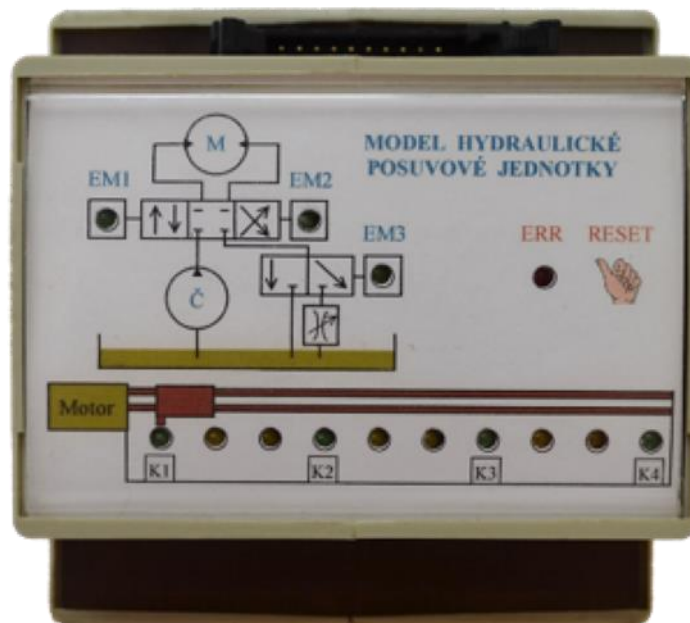
Dále lze PLC FOXTROT 2000 rozšířit o DataBox. Na výběr máme dvě možnosti, a to zvolit velikost paměti 128 KB nebo 256 KB. Další rozšíření může být realizováno prostřednictvím adaptéru sítě WLAN1 nebo LTE1. [21]

3 POPIS VYBRANÝCH MODELŮ

Při výuce problematiky automatizace na programovatelných automatech je pro studenta největší překážkou dostupnost těchto zařízení. Zároveň je také vhodné, aby existoval nějaký vyhodnocovací prvek, který dokáže studentovi ihned ukázat, zda je jeho kód správný. Student pak dokáže během chvilky odhalit chybu a následně ji odstranit, aby program prováděl to, k čemu je určen. Přesně tyto zpětnovazební systémy obsahuje učební pomůcka EDU-mod určená pro systémy a zařízení Tecomat od české firmy Teco a. s. Kolín. [22]

V současné době jsou v laboratoři na Fakultě aplikované informatiky UTB využívány reálné modely EDU-mod připojené k PLC Tecomat TC-1005. Tyto modely mají vlastní vnitřní inteligenci řízenou mikroprocesorem, odpovídají na podněty řídicího programu a vytvářejí vlastní indikační signály. Jsou to modely simulující hydraulickou posuvnou jednotku či křížovátku, dále pak mísicí jednotka a pračka.

3.1 Hydraulická posuvná jednotka



Obrázek 13: Model hydraulické posuvné jednotky [11]

Funkce modelu:

- Čtyři z deseti LED diod slouží jako snímače polohy K1–K4 a společně se zbylými diodami simulují pohyb suportu.

- Tři signálové bity řídí model. Bit EM1 řídí pohyb vpřed, EM2 řídí pohyb vzad a bit EM3 dvoupolohově ovládá rychlost (EM3 = 0 => rychloposuv, EM3 = 1 => pracovní posuv).

Iniciační stav:

- Model se po zpuštění nebo restartu automaticky nastaví do iniciačního stavu – pozice na snímači K1.

Chybová hlášení – model dokáže vyhodnotit dva druhy funkčních chyb:

- přejezd krajních snímačů (K1, K4) → rozsvítí se zelené LED snímačů K1 až K4 a červené LED ERR; po stisku tlačítka RESET se systém vrátí do výchozího stavu;
- současné sepnutí EM1 a EM2 → signalizační dioda ERR bliká, po odstranění chybového stavu systém bez restartu pokračuje v činnosti. [11]

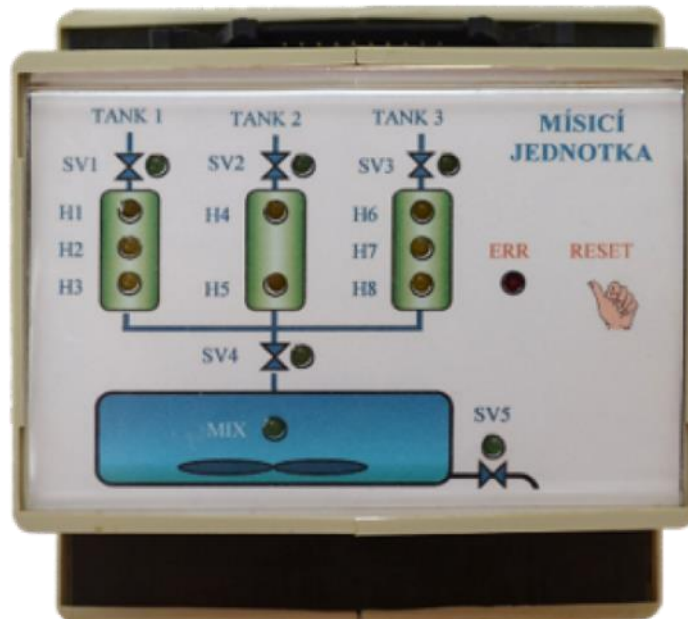
3.2 Křižovatka



Obrázek 14: Model křižovatky [13]

Model křižovatky neobsahuje procesorovou jednotku, a je tedy modulem pasivním. V základu dokáže pouze rozsvěcovat a zhasínat LED diody na základě výstupních signálů řídicího algoritmu. Model je možné rozšířit o další sadu vstupních modulů. Jako třeba systém EDUtech s modulem tlačítek a spínačů s možností simulace tlačítka „chodci“ a možností pozměnění chodu programu. [13]

3.3 Mísicí jednotka



Obrázek 15: Model mísicí jednotky [12]

Funkce modelu:

- Mísicí jednotka je model skládající se ze tří plnicích tanků a mísicí nádoby.
- Modul má vlastní inteligenci.
- Je řízen šesti výstupy – pět představují solenoidové ventily, jeden je mixér.
- Generování chybových hlášení a řízení LED diod simulujících výšky hladin ovládá vnitřní procesorová jednotka.
- Jednotlivé tanky s objemem 84 litrů se začnou plnit rychlostí 6 l/s po sepnutí ventilů SV1 až SV3.
- Minimální hladinu (cca 10 l) signalizují snímače H3, H5 a H8, polovinu nádrže snímače H2 a H7. Plnou nádrž pak signalizují snímače H1, H4 a H6.
- Průtok do a z mísicí nádoby o objemu 253 l je prováděn přes napouštěcí a vypouštěcí potrubí rychlostí průtoku 18 l/s.

Iniciační stav:

- Po restartu tlačítkem RESET nebo pouhém zapnutí napájení se model sám uvede do iniciačního stavu (všechny nádrže jsou prázdné).

- Až do začátku ovládání kteréhokoli solenoidu bude blikat červená LED dioda ERR.

Chybová hlášení:

- Přeteče-li kterákoli nádoba (i mísicí), je vyhodnocena chyba. Rozsvítí se červená LED dioda ERR a až tlačítkem RESET je systém navrácen do výchozího stavu.

[12]

3.4 Pračka



Obrázek 16: Model pračky [10]

Funkce modelu:

- Model pračky je řízen šesti výstupy na úrovni 24V logiky. Stav výstupů jsou signalizovány LED diodami.
- Pohyb bubnu pračky je ovládán dvěma výstupy (BUBEN -> a BUBEN <-), které řídí směr otáčení.
- Rychlost otáčení bubnu je ovládán výstupem OTÁČKY, který při hodnotě 1 ždímá a hodnotě 0 obyčejně pere.
- Napouštění vody do pračky je simulováno výstupem VODA a čerpání vody z pračky výstupem ČERPADOLO.

- Ohřev vody je řízen výstupem TOPENÍ. Při vypnutí topení pračka automaticky chladne.
- LED diody signalizují hladinu v pračce na úrovni 50 a 100 %.
- Ohřátí vody na 90 stupňů Celsia při plné vaně trvá 60 s a teplota je snímána na úrovních 30, 40, 60 a 90 stupňů Celsia.

Chybová hlášení:

- Opravitelná chyba: Začne blikat červená LED dioda ERR, pokud je současně vyslán povel BUBEN -> a BUBEN <-. Buben se přestane otáčet a po odstranění této kolize se začne znovu otáčet ve směru, který je zvolen.
- Neopravitelná chyba č. 1: Přeteče prací vana pračky.
- Neopravitelná chyba č. 2: Voda přesáhne teplotu 90 stupňů Celsia.
- Nastane-li neopravitelná chyba, je rozsvícena LED dioda ERR. Řešením tohoto stavu je pouze stisk tlačítka RESET.

Iniciační stav:

- Po stisknutí tlačítka RESET nebo zapnutí napájení je model připraven k chodu v iniciačním stavu. Teplota je nižší, než je úroveň prvního snímače, a prací nádoba je prázdná. [10]

II. PRAKTICKÁ ČÁST

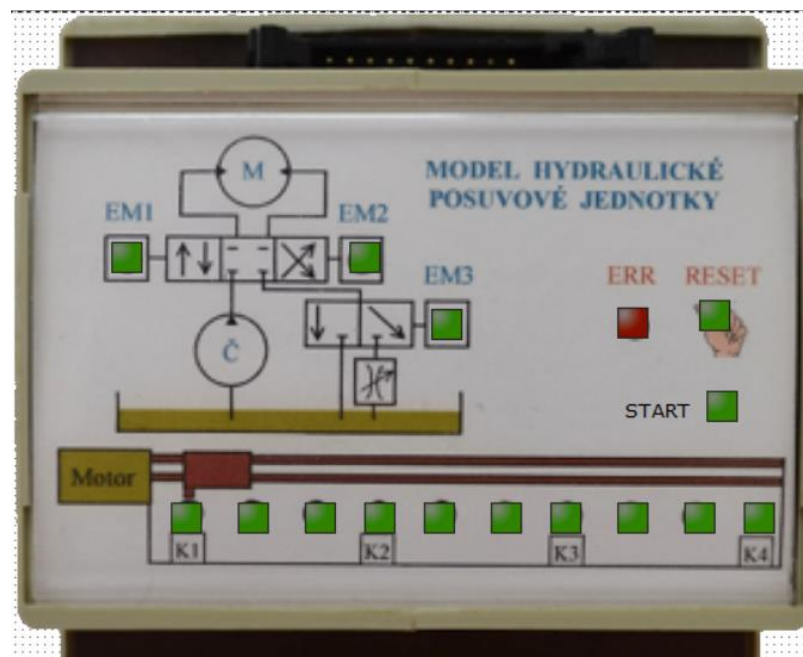
4 NÁVRH SIMULAČNÍCH MODULŮ

V minulé kapitole byly jednotlivé modely popsány z pohledu jejich funkce. V tomto oddíle je blíže popsán způsob jejich tvorby a programování.

Každý z vypracovaných modelů má stejné vlastnosti a funkce, které byly popsány v teoretické části, konkrétně v podkapitole popisující modely procesů EDU-mod. Ke každému z nich – kromě modelu „křižovatka“ – bylo přidáno tlačítko „START“, které je propojeno s globální proměnnou „START“. Tlačítko ovšem není v modelu nijak využíváno. Podle rozhodnutí studenta je možné jej využít pro odstartování chodu programu až ve chvíli, kdy se dívá na vizualizaci, a tu tedy může sledovat od samého začátku.

Podoba modelů je ve vizualizaci stejná jako u jejich reálných verzí. Jako LED diody byly použity dvoustavové obrázky různých barev. Ve všech modelech je využívána knihovna SysLib 4.3, hlavně pro její periodické funkce. Ty jsou použity jako čítače nebo při algoritmu blikání diod.

4.1 Hydraulická posuvná jednotka



Obrázek 17: Vizuál hydraulické posuvné jednotky [18]

Při prvním spuštění nebo po restartu se vynulují všechny hodnoty proměnných modelu. Posuv se přesune na začátek, tedy do bodu K1. Aktivací spínače EM1 se začne posuvník pohybovat směrem doprava. Naopak po aktivaci spínače EM2 se posuvník pohybuje doleva.

Aktivací spínače EM3 se spustí pracovní posuv, který může být kombinován jak se spínačem EM1, tak EM2. Záleží na tom, na kterou stranu chceme pracovní posuv spustit. [11]

Na následujícím obrázku vidíme část kódu, který řeší vizualizaci posuvu posuvníku při pracovním posuvu. Poloha je řešena pomocí proměnné „poloha“ typu integer, která nabývá hodnot od 1 do 10, aby obsáhla všech 10 diod signalizujících posuv. Rychlost „pracovního posuvu“ je nastavena na 1 krok za 1 sec. Každou jednu vteřinu se tedy zvětší proměnná „poloha“ o hodnotu jedna, a posuvník se posune o jeden krok doprava. [11]

```

if EM1_loc then
  cas_dioda( IN := EM1_loc, PT :=T#1s, Q => dioda_step );
  if (dioda_step) then
    dioda := dioda * 2;
    cas_dioda( IN := false);
    poloha := poloha + 1;
  end_if;
end_if;

```

Obrázek 19: Kód ovládající LED diody při vizualizaci pohybu směrem doprava [18]

```

if EM2_loc then
  cas_dioda( IN := EM2_loc, PT :=T#1s, Q => dioda_step );
  if (dioda_step) then
    dioda := dioda / 2;
    cas_dioda( IN := false);
    poloha := poloha - 1;
  end_if;
end_if;

```

Obrázek 18: Kód ovládající LED diody při vizualizaci pohybu směrem doleva [18]

Rychloposuv obsahuje velmi podobný kód, perioda jednoho kroku je však 0.5 sec. Rychloposuv tedy zdvojnásobí rychlost posuvu. [11]

Model obsahuje deset LED diod signalizujících polohu, ale jen čtyři z nich fungují také jako snímače polohy, které se dají využít k řízení posuvníku. Jedná se o LED diody s popisem K1–K4. [11]

U tohoto modelu existují dva možné případy chyby. Pokud dojde k současnému sepnutí spínačů EM1 a EM2, rozblíká se LED s popisem ERR a uživatel musí vypnout jeden z těchto spínačů, neboť posuvník se nemůže pohybovat zároveň doprava i doleva. Pokud však posuvník přejede přes jeden z krajních snímačů polohy, tedy K1 nebo K4, trvale se rozsvítí LED ERR a z tohoto stavu se lze dostat pouze stiskem spínače „RESET“. [11]

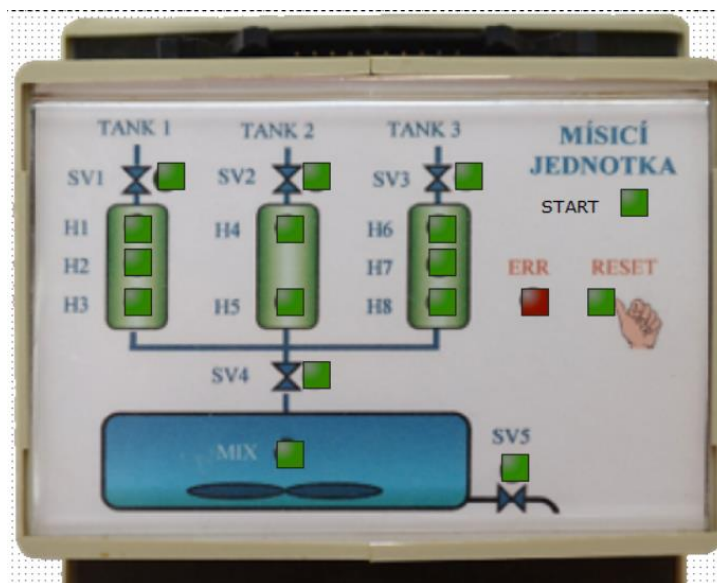
4.2 Křižovatka



Obrázek 20: Vizuál křižovatky [18]

Jak již bylo řečeno, model „křižovatka“ je pasivní modul a pouze zobrazuje pomocí LED diod stavy výstupních signálů řídicího automatu. Tento model tudíž nemá žádnou simulaci. Vytvořil jsem však ve WebMakeru model křižovatky, který je totožný s modelem z originálních textů, a obohatil ji o LED diody, které může student ovládat přes prgMain. [13]

4.3 Mísicí jednotka



Obrázek 21: Vizuál mísicí jednotky [18]

V případě spuštění modelu nebo při jeho restartu se vynulují všechny hodnoty proměnných modelu. Při spuštění některého ze spínačů SV1–SV3 se začnou napouštět nádrže 1–3. Míru jejich naplněnosti zaznamenávají snímače H1–H8. Napouštění nádrží řeším pomocí funkce z knihovny SysLib v4.3, kdy každou vteřinu přičtu 6 l k hladině. Snímače se rozsvěčují podle míry hladiny (při hodnotách 10 l, 42 l a 84 l, tedy v případě plné nádrže). Následující kód ukazuje tuto funkčnost na nádrži číslo 1. [12]

```
if SV1_loc then
  if System_S.R_EDGE_1SEC then
    hladina_1 := hladina_1 + 6;
  end_if;

  if hladina_1 >= 10 then
    H3_loc := true;
  end_if;

  if hladina_1 >= 42 then
    H2_loc := true;
  end_if;

  if hladina_1 >= 84 then
    H1_loc := true;
  end_if;
end_if;
```

Obrázek 22: Algoritmus simulující stoupající hladinu nádrže [18]

Při spuštění spínače SV4 se začnou tekutiny z nádrží 1–3 přelévat do mísicí nádoby. Z každé nádrže se vylévají tekutiny rychlostí 6 l/sec. Kapaliny tedy proudí z nádrží konstantní rychlostí, a to samozřejmě pouze tehdy, pokud v nich nějaké tekutiny jsou. Pokud některá z nádrží tekutiny neobsahuje, napouští se mísicí nádrž pouze z těch nádrží, které jsou nějakou tekutinou naplněny. Spínač SV5 pak spouští vypouštění mísicí nádrže. To se děje rychlostí 18 l/sec. [12]

Zde může rovněž dojít ke dvěma případným poruchám. Jedna nastává při spuštění modelu nebo po jeho restartu, kdy není žádný ze spínačů SV1–SV5 spuštěn. V takovém případě se rozblíká LED ERR a tento stav lze jednoduše opravit spuštěním jakéhokoli spínače SV1–SV5. Pokud však dojde k přetečení jakékoli z nádrží 1–3 nebo mísicí nádrže, dojde k neopravitelné chybě a tu lze odstranit pouze stiskem tlačítka RESET. [12]

4.4 Pračka



Obrázek 23: Vizuál pračky [18]

Při spuštění modelu nebo při jeho restartu se vynulují všechny jeho hodnoty proměnných. Po zapnutí otáčení bubnem vlevo nebo vpravo započne vizualizace otáčení na osmi LED rozmístěných okolo LED signalizujících hladinu vody. Rychlost otáčení bubnu můžeme následně zdvojnásobit pomocí spínače „Otáčky“. Na obrázku vidíme část kódu, jenž ovládá otáčení bubnu doleva a také pohyb směrem doleva při zvýšených otáčkách. [10]

```

if (buben_vlevo_loc) then
  // otáčení se zvýšenými otáčkama
  if (otacky_loc) then
    cas_dioda( IN := buben_vlevo_loc, PT :=T#0.25s, Q => dioda_step );
    if (dioda_step) then
      dioda := dioda * 2;
      cas_dioda( IN := false);
      if dioda > 8 then dioda := 1; end_if;
    end_if;
  else
    // otáčení běžnou rychlostí
    cas_dioda( IN := buben_vlevo_loc, PT :=T#0.5s, Q => dioda_step );
    if (dioda_step) then
      dioda := dioda * 2;
      cas_dioda( IN := false);
      if dioda > 8 then dioda := 1; end_if;
    end_if;
  end_if;
end_if;

```

Obrázek 24: Kód ovládající simulaci otáčení bubnem pračky [18]

Napouštění vody do bubnu se spouští spínačem „Voda“. Hladina vody je hlídána dvěma snímači. Jeden rozsvítí LED, když je hladina v polovině. Druhý rozsvítí LED, když se pračka naplní. Spínačem „Čerpadlo“ zase pračku vypouštíme. [10]

Teplotu pračky ovládáme spínačem „Topení“ a je snímána čtyřmi čidly s ukazateli různých teplot (30 °C, 40 °C, 60 °C a 90 °C). Pokud topení vypneme, samovolně začne probíhat proces chladnutí, který snižuje teplotu vody v pračce. Kód znázorňuje následující obrázek. [10]

```
// chládnutí pračky
if chladnutí THEN
  if not topení_loc and teplota > 0 AND System_S.R_EDGE_1SEC THEN
    teplota := teplota - 3;
  END_IF;
  if teplota < 30 THEN teplota_30_loc := false;
  END_IF;
  if teplota < 40 THEN teplota_40_loc := false;
  END_IF;
  if teplota < 60 THEN teplota_60_loc := false;
  END_IF;
  if teplota < 90 THEN teplota_90_loc := false;
  END_IF;
END IF;
```

Obrázek 25: Kód ovládající simulaci chladnutí pračky [18]

Pokud je zároveň spuštěno otáčení bubnu doprava i doleva, nastane opravitelná chyba, kterou signalizuje LED ERR. Tento stav jednoduše odstraníme tím, že vypneme druhý směr otáčení a necháme běžet pouze jeden. Buben se pochopitelně nemůže točit na obě strany zároveň. [10]

Neopravitelná chyba nastává v případě, kdy přeteče buben pračky nebo je překročena teplota 90 °C. V tu chvíli začne trvale svítit LED ERR a tato chyba může být opravena pouze stiskem spínače RESET. [10]

5 VZOROVÉ ÚLOHY PRO ŘÍZENÍ ZADANÝCH MODULŮ

Předem popsané knihovny jako takové jsou samy o sobě pouze program, který dokáže poskytovat zpětnou vazbu řídicímu programu. Proto byly vytvořeny i vzorové úlohy, jež názorně předvedou funkcionalitu vytvořených modelů. Následující vzorové úlohy jsou zjednodušené a zrychlené verze různých zařízení z reálného světa a měly by pouze prezentovat funkčnost vytvořených knihoven.

5.1 Hydraulická posuvná jednotka

Hned po startu se posuvník stisknutím tlačítka „EM1“ začne pohybovat dopředu. Dojede až na konec dráhy. Tam se na snímači „K4“ otočí a jede zpět prostřednictvím tlačítka „EM2“. Zastaví se na snímači „K3“, kde pomocí časovače TON počká 5 vteřin, a následně po stisknutí tlačítka „EM3“ pokračuje v pracovním posuvu až na začátek dráhy ke snímači „K1“. Zde končí první cyklus a vyresetují se všechny proměnné značící fázi průběhu programu. Zároveň začíná cyklus další. Program tedy vykonává tuto činnost do té doby, dokud je připojen k PLC nebo simulovanému PLC.

Vstupy modelu:			
Proměnná:	Adresa bitu z popisu:	Adresa bitu v laboratoři UTB:	Význam:
EM1	Y0.0	Y0.0	Posun doprava
EM2	Y0.1	Y0.1	Posun doleva
EM3	Y0.2	Y0.2	Rychloposuv
RESET	Bez adresy - vnitřní proměnná		Reset modulu
Výstupy modelu:			
Proměnná:	Adresa bitu z popisu:	Adresa bitu v laboratoři UTB:	Význam:
K1	X0.0	X10.2	První pozice
K2	X0.1	X10.3	Druhá pozice
K3	X0.2	X10.4	Třetí pozice
K4	X0.3	X10.5	Čtvrtá pozice
ERR	Bez adresy - vnitřní proměnná		Indikace chyby

Tabulka 1: Vstupy a výstupy modulu posuvník

5.2 Křižovatka

Tato úloha spočívá se využití jednoduchého časovače a cyklu procesů. Na začátku je spuštěn časovač, který po určitých časových intervalech postupně rozsvěcuje LED diody demonstrující světla na semaforu. Nejprve je ovládán spodní semafor, kde dochází ke střídání signálních fází od zelené, přes oranžovou až k červené barvě. Poté jsou spuštěny cykly na zbylých dvou semaforech, z nichž jeden je určen pro chodce. U těch rovněž

proběhne jedno kolo změny barev. Nakonec cyklus skončí zpět na červené. V tuto chvíli se resetuje časovač a celý proces se může opakovat.

Proměnná:	Adresa bitu z popisu:	Vstupy modelu:		Význam:
		Adresa bitu v laboratoři	UTB:	
cerpv_spodek	Y0.0		Y2.0	Červená na spodním semaforu Oranžová na spodním semaforu
oran_spodek	Y0.1		Y2.1	Zelená na spodním semaforu
zel_spodek	Y0.2		Y2.2	Červená na horním semaforu
cerv_vrch	Y0.3		Y2.3	Oranžová na horním semaforu
oran_vrch	Y0.4		Y24.0	Zelená na horním semaforu
zel_vrch	Y0.5		Y24.1	Zelená pro chodce
cerv_chodci	Y0.6		Y24.2	Červená pro chodce
zel_chodci	Y0.7		Y24.3	

Tabulka 2 Vstupy a výstupy modulu křižovatka

5.3 Mísicí jednotka

Na začátku program počká 5 vteřin prostřednictvím pokynů časovače TON. Během té doby se u prvního startu přerušovaně rozsvítí červená LED ERR. Kontrolka tedy bliká do prvního vybuzení některého ze solenoidů. Poté se otevrou ventily „SV1“, „SV2“ a „SV3“. To zapříčiní napouštění nádrží 1–3. Ve chvíli, kdy snímače „H1“, „H4“ a „H6“ zahlásí, že jsou nádrže plné, vypnou se ventily „SV1–3“ a otevře se ventil „SV4“, který umožní vypouštění těchto tří nádrží. V okamžiku, kdy kontrolky „H3“, „H5“ a „H8“ zahlásí minimální hladinu v nádržích, pokračuje jejich vyprázdňování ještě 2 vteřiny. Následně je obsah mísicí nádoby míchán po dobu 5 vteřin, což znázorňuje blikající LED dioda „MIX“. Poté je veškerý obsah mísicí nádoby vyprázdňěn pomocí ventilu „SV5“. Zde celý proces končí a vyresetují se všechny proměnné značící fázi průběhu programu. Následuje další cyklus programu, který opět začíná pětivteřinovou přestávkou, při které již LED dioda „ERR“ neblinká. Program se tak stále cyklicky opakuje.

Vstupy modelu:			
Proměnná	Adresa bitu z	Adresa bitu v laboratoři	Význam:
:	popisu:	UTB:	
SV1	Y0.0	Y2.0	Ventil pro napouštění TANK1
SV2	Y0.1	Y2.1	Ventil pro napouštění TANK2
SV3	Y0.2	Y2.2	Ventil pro napouštění TANK3
SV4	Y0.5	Y24.0	Ventil pro vypouštění všech TANKŮ
SV5	Y0.4	Y24.1	Ventil pro vypouštění mísicí nádoby
MIX	Y0.6	Y24.2	Zapnutí mixování
RESET	Bez adresy - vnitřní proměnná		Reset modulu
Výstupy modelu:			
Proměnná	Adresa bitu z	Adresa bitu v laboratoři	Význam:
:	popisu:	UTB:	
H1	X0.3	X10.5	Hladinoměř H1
H2	X0.2	X10.4	Hladinoměř H2
H3	X0.1	X10.3	Hladinoměř H3
H4	X0.0	X10.2	Hladinoměř H4
H5	X0.7	X84.3	Hladinoměř H5
H6	X0.4	X84.0	Hladinoměř H6
H7	X0.5	X84.1	Hladinoměř H7
H8	X0.6	X84.2	Hladinoměř H8
ERR	Bez adresy - vnitřní proměnná		Indikace chyby

Tabulka 3: Vstupy a výstupy modulu mísicí jednotka

5.4 Pračka

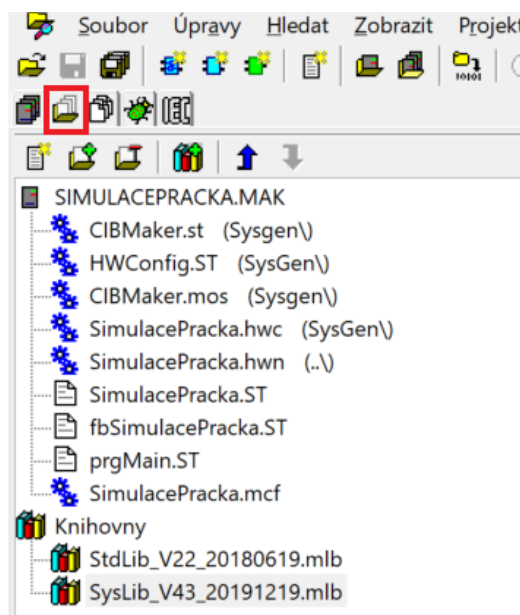
Po spuštění dochází v pračce k postupnému ohřevu a zároveň napouštění vody. V této fázi se přístroj snaží udržet teplotu na úrovni 30 °C, což hlásí kontrolky teploty nacházející se na modelu vpravo dole. Ve chvíli, kdy je vana pračky naplněna na 100 %, což značí kontrolka hladiny, zastaví se napouštění vody a provede se pětivteřinová předpírka. Následně se celý buben pračky vypustí. Program pokračuje fází praní na 60 °C, kdy se pračka zahřeje na požadovanou teplotu, buben se opět naplní a probíhá pětivteřinové praní. Následně je vypuštěna veškerá voda a ve chvíli, kdy je buben pračky prázdný, vypne se topení, a zařízení tak začne automaticky chladnout. Zároveň se začne napouštět buben do maximální kapacity a ve chvíli, kdy je plný, se začne opět vypouštět. Jakmile je buben prázdný, pokračuje cyklus pětivteřinovým ždímáním, kdy dochází k otáčení bubnu. V této fázi lze zapnout i zvýšené otáčky. Tímto je první cyklus pračky dokončen a všechny proměnné značící fázi praní jsou vyresetovány. Program tedy započne další cyklus a bude se stále opakovat.

Vstupy modelu:			
Proměnná:	Adresa bitu z popisu:	Adresa bitu v laboratoři UTB:	Význam:
buben_vpravo	Y0.0	Y2.0	Otáčení bubnu doprava
buben_vlevo	Y0.1	Y2.1	Otáčení bubnu doleva
otacky	Y0.2	Y2.2	Ždímání
topeni	Y0.4	Y24.0	Ohřev vody
napousteni	Y0.5	Y24.1	Napouštění vody do bubnu
cerpadlo	Y0.6	Y24.2	Vypouštění vody z bubnu
RESET	Bez adresy - vnitřní proměnná		Reset modulu
Výstupy modelu:			
Proměnná:	Adresa bitu z popisu:	Adresa bitu v laboratoři UTB:	Význam:
teplota_90	X0.0	X10.2	Signalizace 90°C
teplota_60	X0.1	X10.3	Signalizace 60°C
teplota_40	X0.2	X10.4	Signalizace 40°C
teplota_30	X0.3	X10.5	Signalizace 30°C
voda_50	X0.6	X84.2	Hladinoměr 50%
voda_100	X0.7	X83.3	Hladinoměr 100%
ERR	Bez adresy - vnitřní proměnná		Indikace chyby

Tabulka 4: Vstupy a výstupy modulu pračka

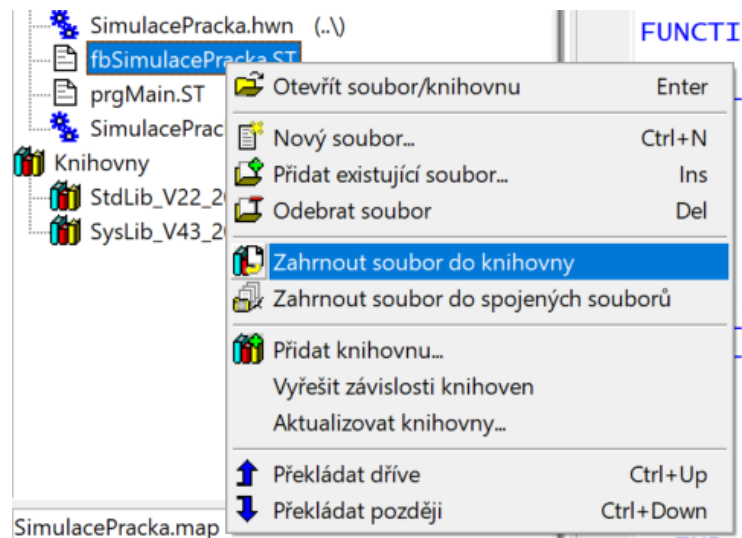
5.5 Vytvoření knihoven

Z vytvořených simulačních funkčních bloků je ještě třeba vytvořit knihovny, aby byly co nejjednodušeji použitelné pro studenty. Tento proces předvedu pouze na jednom z programů, konkrétně u modelu pračky, neboť postup je u každého programu stejný. Nejprve přejdeme do sekce „Soubory projektu“ označené červeným čtvercem na následujícím obrázku.



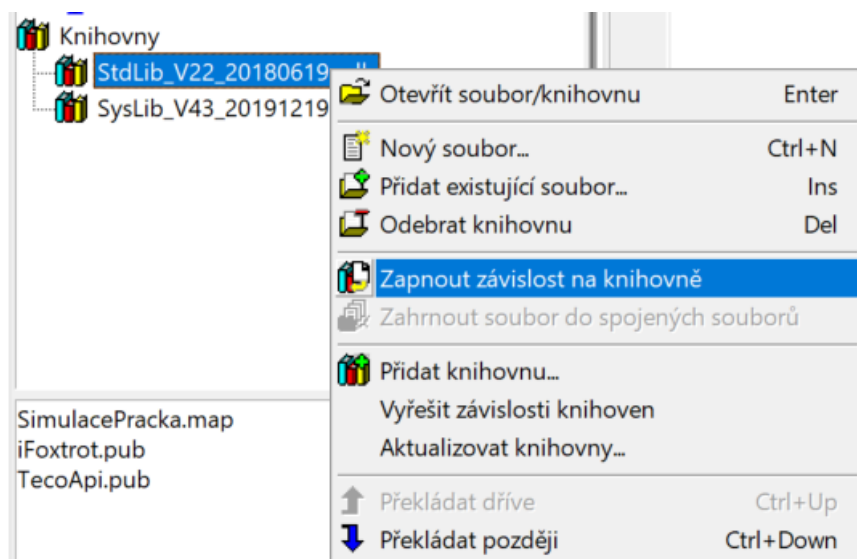
Obrázek 26: Návod tvorby knihoven – krok 1 [18]

Následně si označíme funkční blok, který má tvořit knihovnu.



Obrázek 27: Návod tvorby knihoven – krok 2 [18]

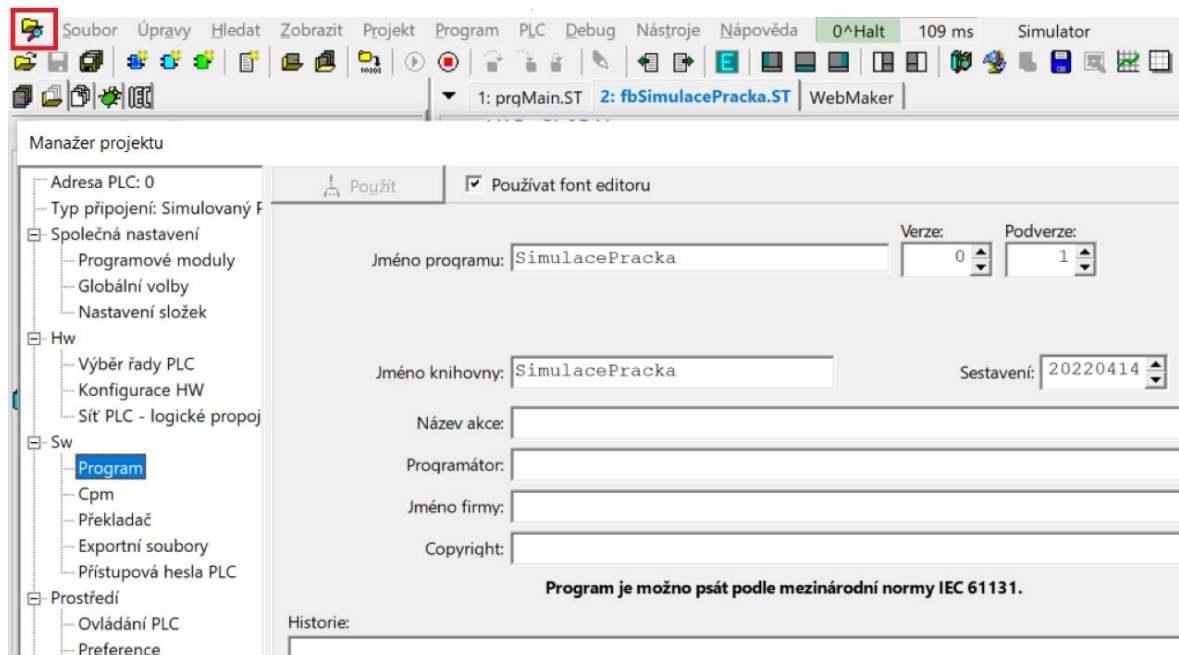
V tomto případě je třeba s námi tvořenou knihovnou propojit i jiné knihovny, které byly použity při její tvorbě. Jedná se o knihovny StdLib_V22 a Std_LibV43. Proto u každé z těchto knihoven nastavíme „Zapnout závislost na knihovně“.



Obrázek 28: Návod tvorby knihoven – krok 3 [18]

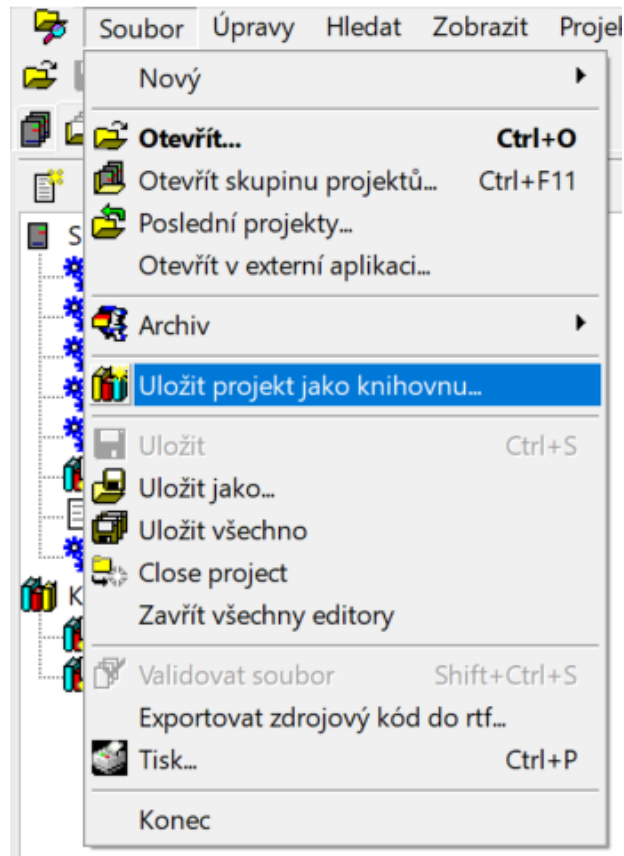
Dále se v levém horním rohu dostaneme přes „Manažer projektů“ do sekce „SW“ a následně vybereme „Program“, kde si pojmenujeme naši budoucí knihovnu.

Jeden z posledních kroků spočívá již pouze v přeložení celého projektu.



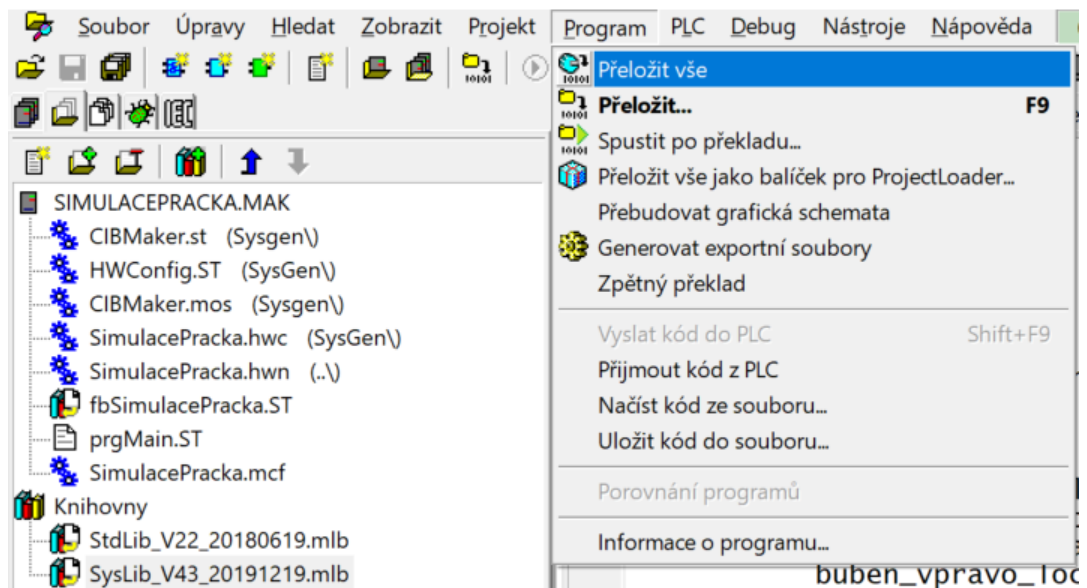
Obrázek 29: Návod tvorby knihoven – krok 4 [18]

Nakonec tedy v kolonce „Soubor“ vybereme „Uložit projekt jako knihovnu“.

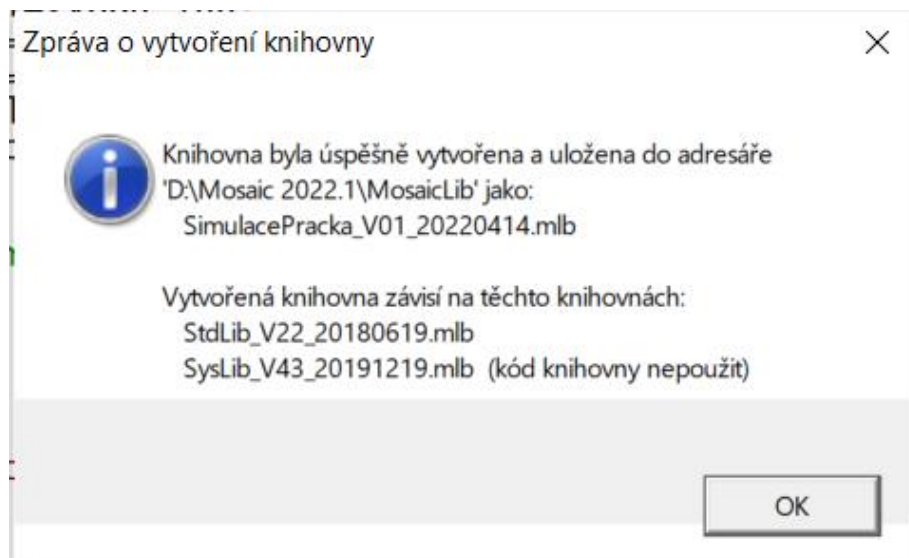


Obrázek 30: Návod tvorby knihoven – krok 5 [18]

V této fázi se program naposledy „přeloží“ a objeví se hlášení „Zpráva o vytvoření knihovny“.



Obrázek 32: Návod tvorby knihoven – krok 6 [18]



Obrázek 31: Návod tvorby knihoven – krok 7 [18]

6 VYUŽITÍ KNIHOVEN VE VÝUCE

Na konci minulé kapitoly jsme si ukázali, jak lze vytvořit knihovnu v systému Mosaic. V tomto oddíle bude popsáno využití vytvořené simulační knihovny přímo ve výuce. Student tak může učinit dvěma způsoby, jejichž charakteristika následuje.

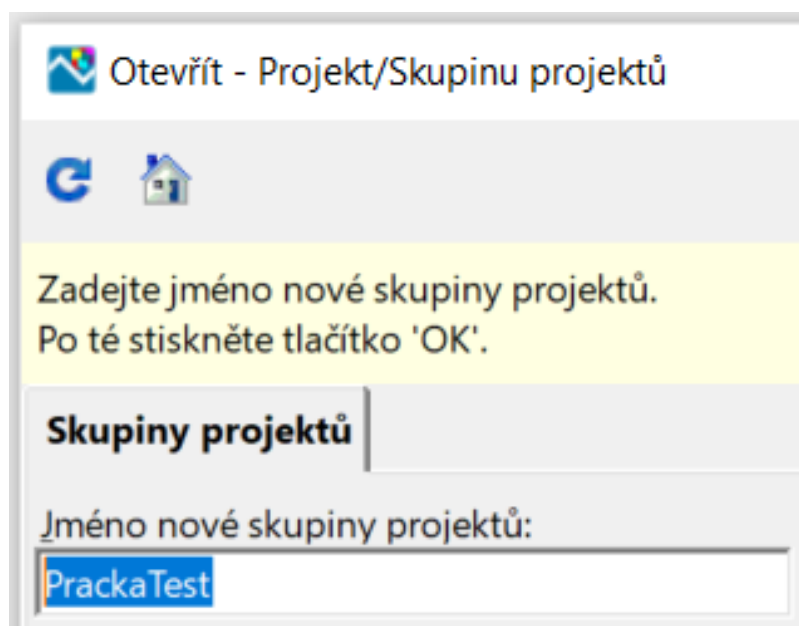
6.1 Prosté použití knihovny

Nejprve si zapneme vývojové prostředí Mosaic, následně vybereme kolonku vlevo nahoře „Nová skupina projektů“.



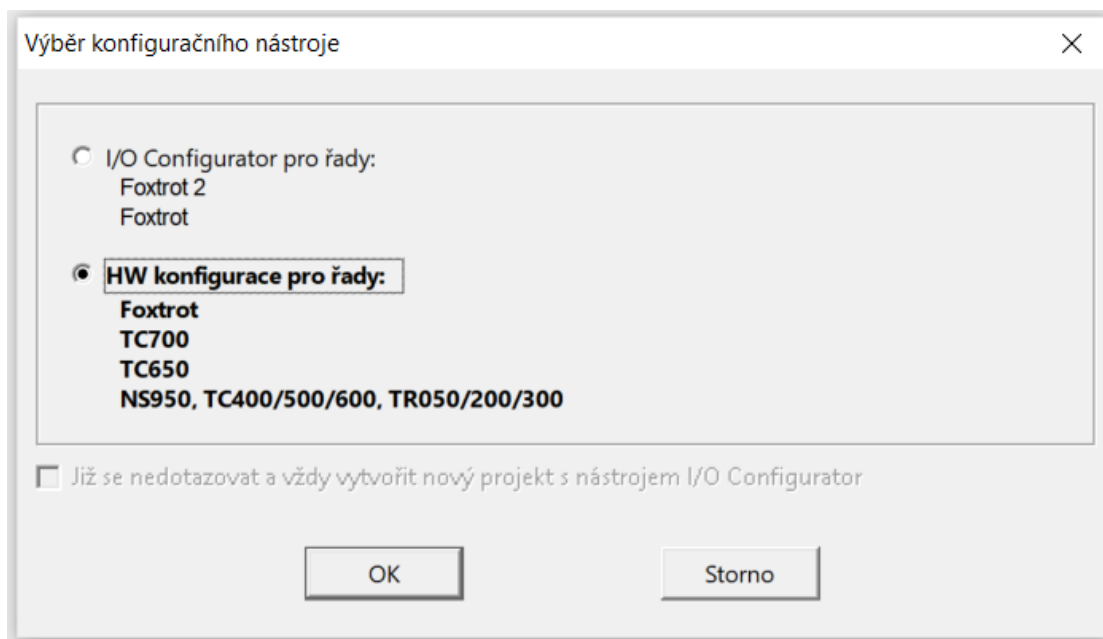
Obrázek 33: Vložení knihoven do projektu – krok 1 [18]

Hned v dalším kroku zvolíme „Jméno nové skupiny projektů“.



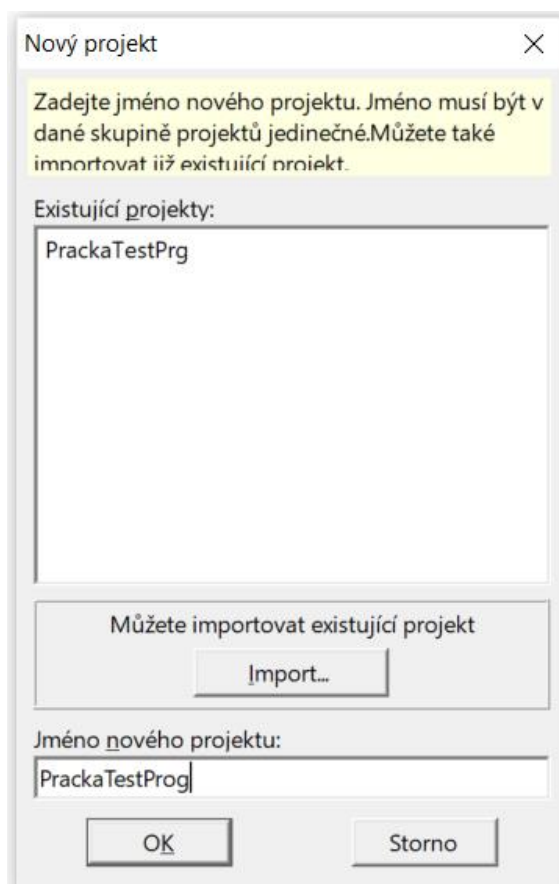
Obrázek 34: Vložení knihoven do projektu – krok 2 [18]

Označíme příslušnou konfiguraci.



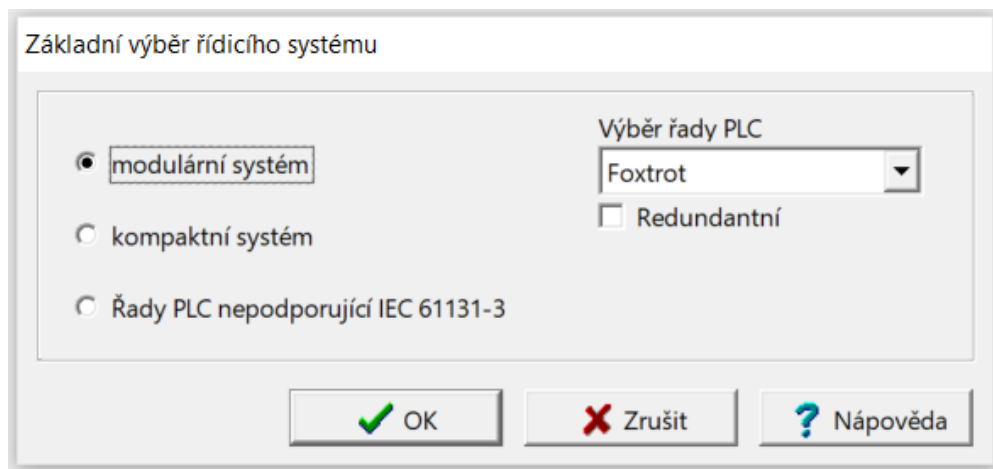
Obrázek 35: Vložení knihoven do projektu – krok 3 [18]

V dalším kroku vybereme „Jméno nového projektu“



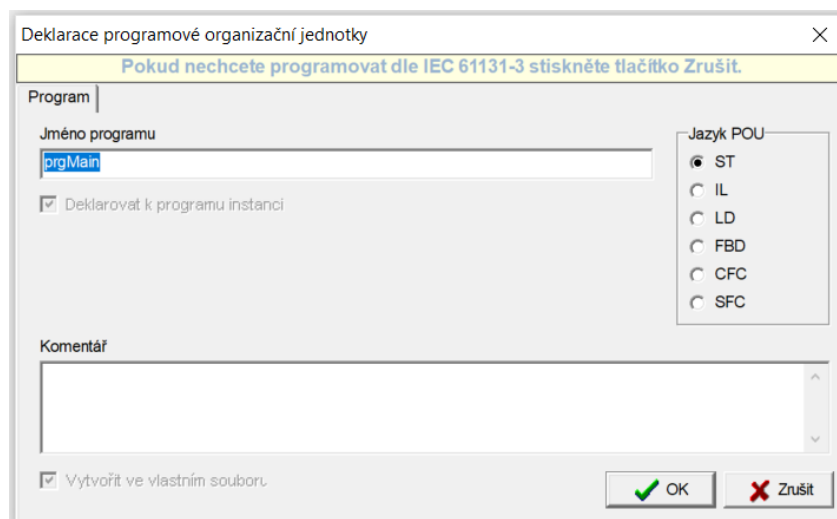
Obrázek 36: Vložení knihoven do projektu – krok 4 [18]

a zvolíme „Modulární systém“.



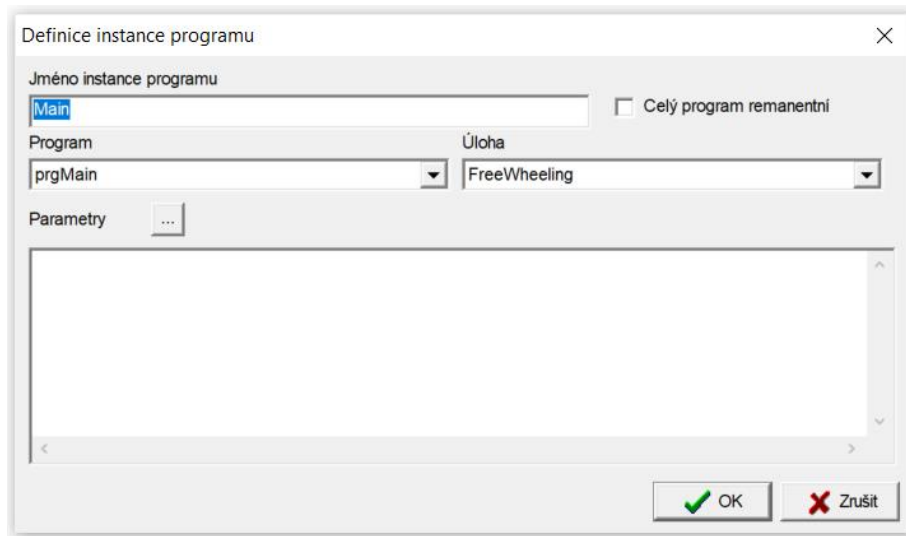
Obrázek 37: Vložení knihoven do projektu – krok 5 [18]

V dalším kroku potvrdíme název hlavní programové jednotky „prgMain“.



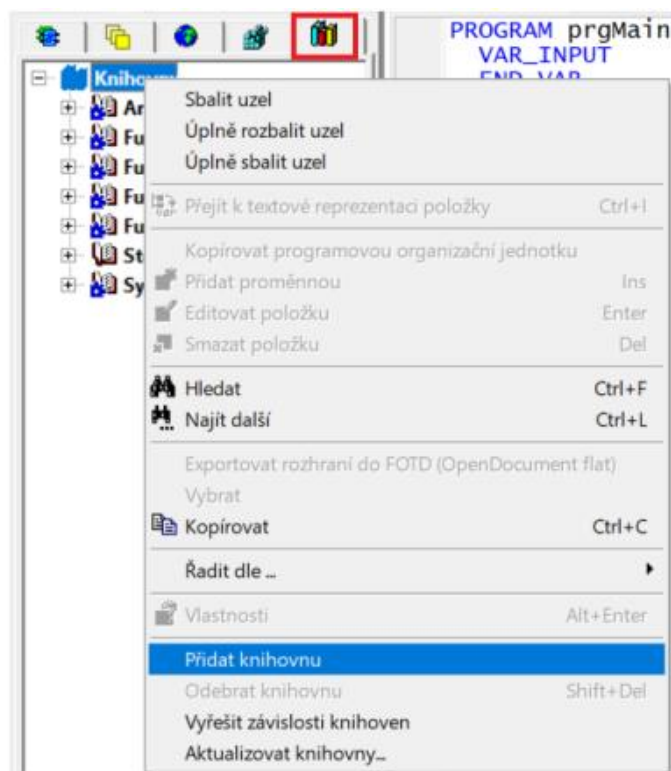
Obrázek 38: Vložení knihoven do projektu – krok 6 [18]

Instanci s názvem „Main“ rovněž pouze potvrdíme.



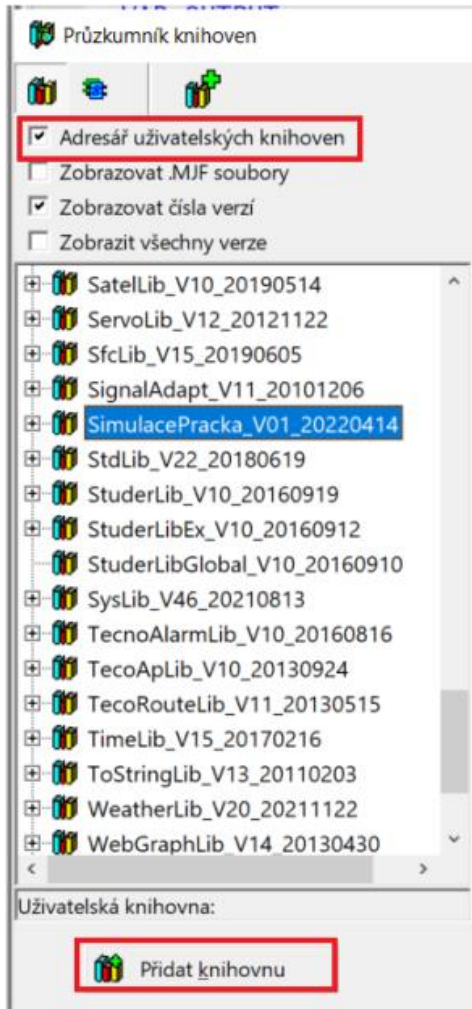
Obrázek 39: Vložení knihoven do projektu – krok 7 [18]

V tuto chvíli máme založen nový projekt. Nyní je třeba přidat simulační knihovnu do projektu, vytvořit její instanci a tzv. ji zavolat. Nejprve proto v sekci „Knihovny“ zadáme „Přidat knihovnu“.

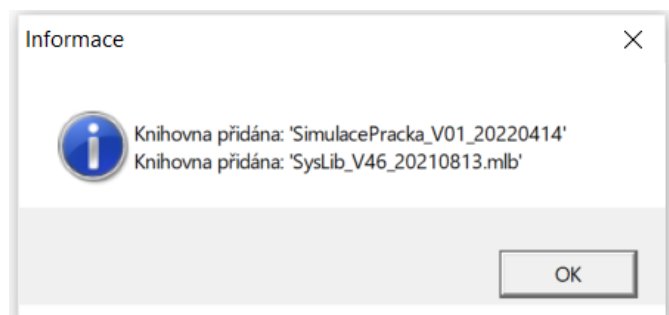


Obrázek 40: Vložení knihoven do projektu – krok 8 [18]

Následně vlevo nahoře zaškrtneme „Adresář uživatelských knihoven“, v seznamu na levé straně označíme námi požadovanou knihovnu, kterou bychom chtěli využít, a potvrdíme „Přidat knihovnu“. Načež dostaneme informaci o přidání knihovny.



Obrázek 41: Vložení knihoven do projektu – krok 9 [18]



Obrázek 42: Vložení knihoven do projektu – krok 10 [18]

V tuto chvíli máme knihovnu přidanou v projektu a můžeme ji použít.

Vytvoříme si tedy instanci simulační knihovny a tuto instanci zavoláme s následujícími parametry. Proto ještě nastavíme verzi HW na CP-1005.

```

1* prgMain.ST | 2: PrackaTestProg.mcf
PROGRAM prgMain
VAR_INPUT
END_VAR
VAR_OUTPUT
END_VAR
VAR
END_VAR

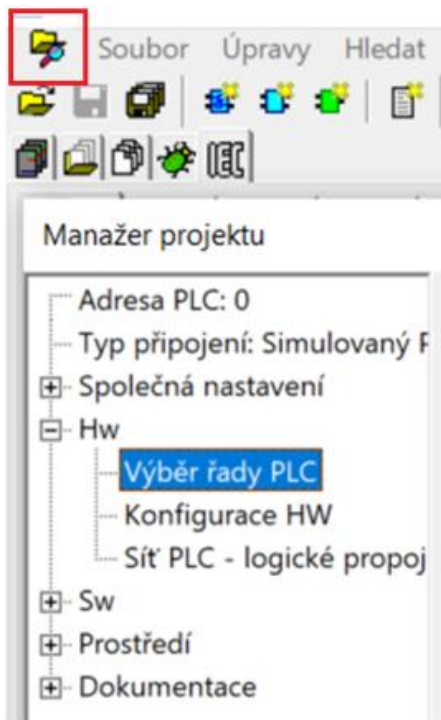
//SIMULACE !!!!!
Simulace : fbSimulacePracka;

END_VAR
VAR_TEMP
END_VAR

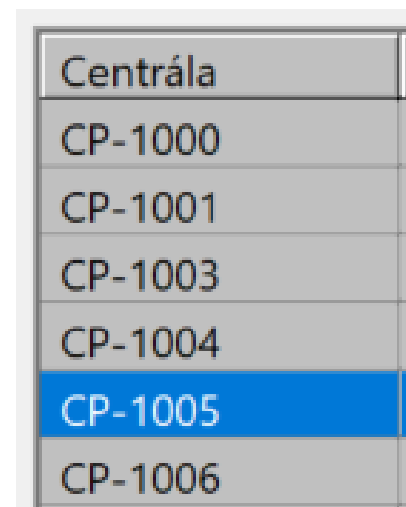
// Simulace
// Pokud nemáte přístup k PLC v laboratoři, odkomentujte následující řádek,
// který spouští bunkční blok, jež simuluje chování PLC v laboratoři.
Simulace(   topeni_local:=topeni,
            cernadlo_local:=cernadlo,
            napousteni_local:=napousteni,
            teplota_30_local=>teplota_30,
            teplota_40_local=>teplota_40,
            teplota_60_local=>teplota_60,
            teplota_90_local=>teplota_90,
            voda_100_local=>voda_100,
            voda_50_local=>voda_50,
            otacky_local:= otacky,
            buben_vpravo_local:= buben_vpravo,
            buben_vlevo_local:= buben_vlevo
);

```

Obrázek 45: Vložení knihoven do projektu – krok 11 [18]



Obrázek 44: Vložení knihoven do projektu – krok 12 [18]

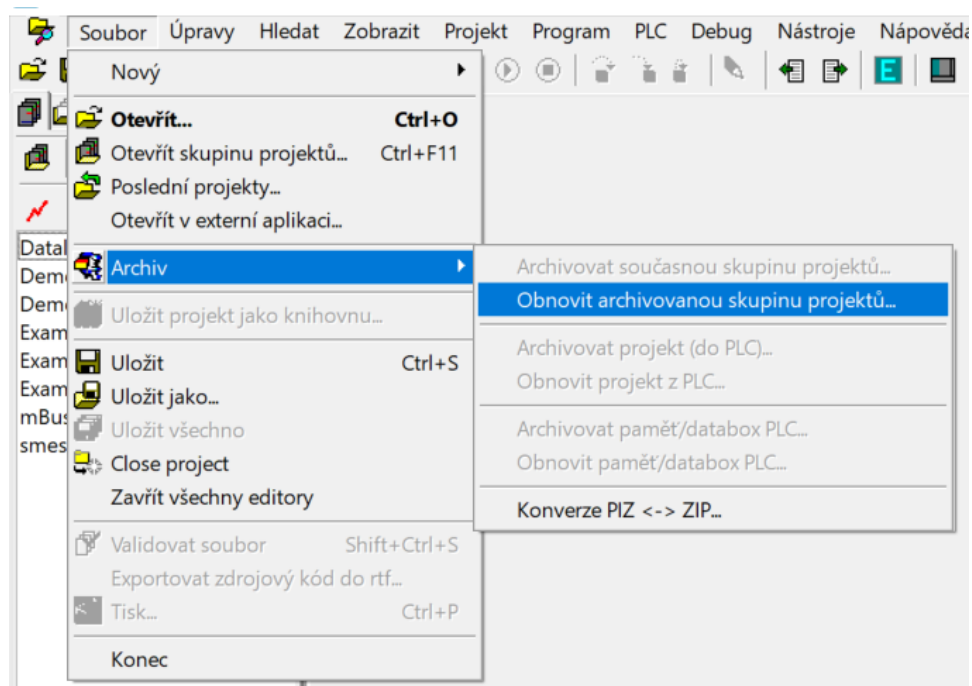


Obrázek 43: Vložení knihoven do projektu – krok 13 [18]

Nakonec ještě přiřadíme fyzické vstupy a výstupy na správné svorky tak, jak bylo v případě jednotlivých knihoven popsáno v kapitole 5.

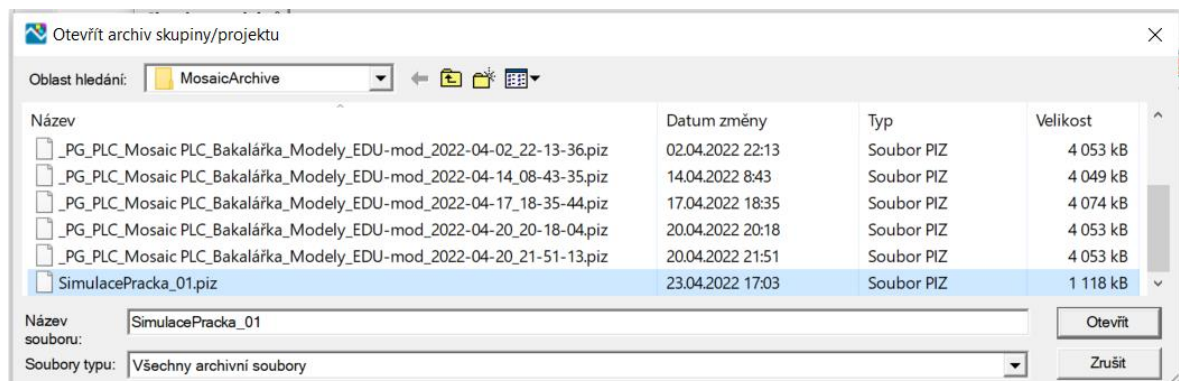
6.2 Otevření archivovaného projektu

Nevýhodou prostého využití simulačních knihoven, které byly vytvořeny v praktické části této práce, je fakt, že student si stále musí vytvářet vizualizaci, aby pro něj bylo pohodlné ladit napsaný program. Proto také v příloze uvádím celé projekty, které jsem během tvorby knihoven vytvořil. Dáme „Obnovit archivovanou skupinu projektů“



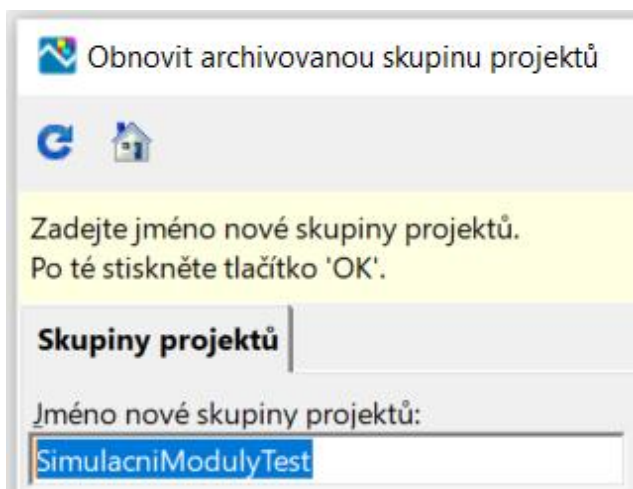
Obrázek 46: Obnovení archivovaného projektu krok – 1 [18]

a vyhledáme soubor v počítači.

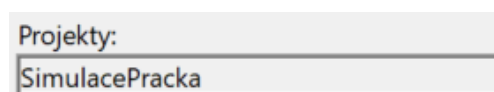


Obrázek 47: Obnovení archivovaného projektu – krok 2 [18]

Vybereme název nové skupiny projektů a v dalším kroku zvolíme na pravé straně projekt „SimulacePracka“.



Obrázek 48: Obnovení archivovaného projektu – krok 3 [18]



Obrázek 49: Obnovení archivovaného projektu – krok 4 [18]

Po potvrzení tohoto výběru se projekt následně načte, a student jej tedy může začít využívat.

ZÁVĚR

Záměrem mé bakalářské práce byl návrh a vytvoření simulačních knihoven reálných modelů, jako i jejich popis s cílem usnadnit studentům výuku. Celou práci jsem také pojal jako určitý manuál a teoretickou přípravu za účelem pomoci studujícím s programováním PLC v prostředí Mosaic.

V teoretické části jsou popsány způsoby a technologie, které jsem využil při tvorbě praktické části. Zároveň se také snažím studentům rozšířit obzory ukázkami dalších způsobů využití prostředí Mosaic. Následně představuji i hardware, se kterým se setkávají v laboratoři a na který byla praktická práce tvořena. K němu příkládám i popis novějšího modelu se záměrem prezentovat nové technologie použité v případě hardwaru od firmy Teco a. s.

Výstupem praktické části mé práce jsou čtyři knihovny, které simulují reálné modely EDU-mod. Knihovny jsou vytvořeny tak, aby co nejvěrněji simulovaly reálné modely z laboratoře. Ty ovšem vykazují drobný nedostatek z hlediska podávání zpětné vazby. Neposílají informaci o chybě modulu z důvodu překročení určitých mezí. Do modulu rovněž nelze vyslat požadavek na reset. Pro studenty je to pak složitější na uchopení, neboť bez této zpětné vazby není možné program zastavit, a ten tedy pokračuje dál. Pokud si však student pohlídá napsaný program tak, aby se modul nedostal do chybového stavu, jsou tyto moduly skutečně pro výuku dostačující. I tak mě mrzí, že prozatím nelze využít plný potenciál progresivního přístupu a nejsou napsány kvalitnější knihovny.

Práce obsahuje popis, jak fungují modelové příklady ukazující funkcionalitu knihoven. Rovněž je podrobně popsáno jejich využití. V příloze jsou uvedeny všechny soubory obsahující i kompletní výčet vytvořených projektů, které lze jednoduše extrahovat a psát přímo do nich. To také považuji za vhodnější způsob, jak knihovny použít, neboť v projektu je vytvořena i vizualizace ve WebMakeru a projekt obsahuje i zdrojový kód vzorového příkladu. Lze se zde seznámit i se zdrojovým kódem vytvořených knihoven.

Věřím, že tato práce do jisté míry usnadní studentům výuku, umožní jim praktický náhled na teoretickou problematiku PLC programování a přispěje k pochopení jeho funkčních principů.

SEZNAM POUŽITÉ LITERATURY

[1] SINGH, Jitender a Monika DESWAL. *PLC and SCADA*. NEW DELHI: LAXMI PUBLICATIONS, 2015. ISBN 978-93-5138-242-3.

[2] HEINZ, Karl a Michael TIEGELKAMP. IEC 61131-3: Programming Industrial Automation Systems: Concepts and Programming Languages, Requirements for Programming Systems, Decision-Making Aids. 2. vydání. Berlin, Heidelberg: Springer, 2010. ISBN 978-3-642-12014-5.

[3] *ZAČÍNÁME V PROSTŘEDÍ MOSAIC*. 8. vydání. Kolín: Teco, 2010. Dostupné také z: https://www.tecomat.cz/modules/DownloadManager/download.php?alias=txv00320_01_mosaic_progstart_cz

[4] *Nástroj WebMaker*. Desáté vydání. Kolín: Teco, 2013. Dostupné také z: https://www.tecomat.cz/modules/DownloadManager/download.php?alias=txv00328_01_mosaic_webmaker_cz

[5] Programování PLC podle normy IEC 61 131-3 v prostředí Mosaic. Desáté vydání. Kolín: Teco, 2007. Dostupné také z: https://www.tecomat.cz/modules/DownloadManager/download.php?alias=txv00321_01_mosaic_progiec_cz

[6] PŘÍRUČKA PROGRAMÁTORA PLC TECOMAT. 13. vydání. Kolín: Teco, 2007. Dostupné také z: [https://web.rcmt.cvut.cz/users/cerny/PLC_sup/TXV00109_\(v13\)_Prirucka_programatora_PLC_TECOMAT.pdf](https://web.rcmt.cvut.cz/users/cerny/PLC_sup/TXV00109_(v13)_Prirucka_programatora_PLC_TECOMAT.pdf)

[7] Knihovny pro programování PLC Tecomat podle IEC 61 131-3. Osmé vydání. Kolín: Teco, 2006. Dostupné také z: <https://adoc.pub/knihovny-pro-programovani-plc-tecomat-podle-iec.html>

[8] REGULAČNÍ KNIHOVNY PRO MOSAIC. 4. vydání. Kolín: Teco, 2008. Dostupné také z: https://www.tecomat.cz/modules/DownloadManager/download.php?alias=txv00323_01_mosaic_regolib_cz

[9] MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b. 2006. Dostupné také z: https://modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf

- [10] KOHOUT, Luděk. Modely EDU-mod: Pračka [online]. 2008 [cit. 2022-04-26]. Dostupné z: <http://www.edumat.cz/produkty.php?produkt=pracka>
- [11] KOHOUT, Luděk. Modely EDU-mod: Suport [online]. 2008 [cit. 2022-04-26]. Dostupné z: <http://www.edumat.cz/produkty.php?produkt=suport>
- [12] KOHOUT, Luděk. Modely EDU-mod: Mísicí jednotka [online]. 2008 [cit. 2022-04-26]. Dostupné z: <http://www.edumat.cz/produkty.php?produkt=mixer>
- [13] KOHOUT, Luděk. Modely EDU-mod: Křížovátka [online]. 2008 [cit. 2022-04-26]. Dostupné z: <http://www.edumat.cz/produkty.php?produkt=krizovatka>
- [14] PETER. Timers in PLC Programming. PLC Academy [online]. 20.12.2018 [cit. 2022-05-07]. Dostupné z: <https://www.plcademy.com/plc-timers/#on-delay-timer>
- [15] HUUCK, Ralf. Semantics and Analysis of Instruction List Programs: Figures, Tables, and Topics from this paper. SEMANTIC SCHOLAR [online]. 18.1.2005 [cit. 2022-05-07]. Dostupné z: <https://www.semanticscholar.org/paper/Semantics-and-Analysis-of-Instruction-List-Programs-Huuck/f9a4169ea302ee002eabd4eb8820bf2085906758/figure/0>
- [16] AHMED, Bestoun. An example of a PLC control program written using the FBD programming language.: Figure 5. ResearchGate [online]. Zář 2020 [cit. 2022-05-07]. Dostupné z: https://www.researchgate.net/figure/An-example-of-a-PLC-control-program-written-using-the-FBD-programming-language_fig3_339137394
- [17] B, Eugeniu. PLC Programming Languages – Short Overview: Ladder Logic. Isd [online]. 9.7.2019 [cit. 2022-05-07]. Dostupné z: <https://isd-soft.com/wp-content/uploads/2019/07/PLC-programming-languages.-EBa-1.png>
- [18] PTÁČEK, Patrik. Vlastní tvorba pomocí programu Mosaic firmy Teco a.s.. [cit. 2022-05-08].
- [19] PROGRAMOVATELNÉ AUTOMATY TECOMAT FOXTROT CP-1005, CP-1015. 9. vydání. Kolín: Teco, 2017. Dostupné také z: https://www.tecomat.cz/modules/DownloadManager/download.php?alias=txv00435_01_foxtrot_cp_1005
- [20] PLC Tecomat Foxtrot – základní moduly. Kolín: Teco. Dostupné také z: <https://www.tecomat.cz/modules/DownloadManager/download.php?alias=foxtrot-cz-cp-1005>

[21] REGULAČNÍ KNIHOVNY PRO MOSAIC. 4. vydání. Kolín: Teco, 2008. Dostupné také z:

https://www.tecomat.cz/modules/DownloadManager/download.php?alias=txv00455_01_foxtrot2_cp-2000_u02_c

[22] PRODUKTY. Luděk Kohout [online]. 2008 [cit. 2022-05-08]. Dostupné z:

<http://www.edumat.cz/produkty.php>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

a.s.	Akciová společnost
AC	Střídavý proud
AI	Analog input = analogový vstup
AO	Analog output = analogový výstup
APOD	A podobně
ATD	A tak dále
CCA	Circa = přibližně
CFC	Continuous Function Chart
CMOS	Complementary Metal–Oxide–Semiconductor
CPU	Central Procesor Unic = procesor
DC	Stejnoseměrný proud
DI	Digital input = digitální vstup
DO	Digital output = digitální výstup
ERR	Error = chyba
FB	Function Block = funkční blok
FBD	Function Block Diagram = diagram funkčních bloků
FUN	Function = funkce
HMI	Human Machine Interface = uživatelské rozhraní
HW	Hardware
I/O	Input/output = Vstup/výstup
IEC	International Electrotechnical Commission = Mezinárodní elektrotechnická komise
IL	Instruction list = jazyk seznamu funkcí
LAN	Local Area Network = místní síť
LD	Ladder diagram = jazyk příčkového diagramu
LED	Light-Emitting Diode = světelná dioda

- LTE Long Term Evolution = dlouhodobý vývoj
- MAC Media Access Control = MAC adresa
- Mb Mega bite
- PID Proportional–integral–derivative controller =
- PLC Programmable logic controller = programovatelný automat
- POU Program Organisation Unit = programové organizační jednotky
- PROG Program Organisation Unit = programové organizační jednotky
- RAM Random acces memory = paměť s náhodným přístupem
- RGB Red-green-blue = systém označení barev
- RTC Real-time clock = hodiny reálného času
- ST Structured text = jazyk strukturovaného textu
- SW Software
- TOF Off delay timer = časovač bez zpoždění
- TON On delay timer = časovač se zpožděním
- TP Pulse timer = pulzní časovač
- XML Extensible Markup Language = rozšiřitelný značkovací jazyk

SEZNAM OBRÁZKŮ

Obrázek 1: Příkazy používané v programovacím jazyce IL [15]	13
Obrázek 2: Ukázka jazyku funkčního blokového schématu [16]	14
Obrázek 3: Ukázka programu v jazyku příčkového diagramu [17]	14
Obrázek 4: Schéma průběhu časovače TON [14]	18
Obrázek 6 Schéma průběhu časovače TP [14]	19
Obrázek 5 Schéma průběhu časovače TOF [14]	19
Obrázek 7: Pracovní plocha WebMakeru [18]]	20
Obrázek 8: Lišta funkcí WebMakeru [18]	21
Obrázek 9: Lokální menu WebMakeru [18]	23
Obrázek 10: Vrstvy objektů ve WebMakeru [18]	25
Obrázek 11: PLC Tecomat CP-1005 [18]	28
Obrázek 12. PLC Tecomat CP-2000 [18]	29
Obrázek 13: Model hydraulické posuvné jednotky [11]	30
Obrázek 14: Model křížovanky [13]	31
Obrázek 15: Model mísicí jednotky [12]	32
Obrázek 16: Model pračky [10]	33
Obrázek 17: Vizuál hydraulické posuvné jednotky [18]	36
Obrázek 19: Kód ovládající LED diody při vizualizaci pohybu směrem doleva [18]	37
Obrázek 18: Kód ovládající LED diody při vizualizaci pohybu směrem doprava [18]	37
Obrázek 20: Vizuál křížovanky [18]	38
Obrázek 21: Vizuál mísicí jednotky [18]	38
Obrázek 22: Algoritmus simulující stoupající hladinu nádrže [18]	39
Obrázek 23: Vizuál pračky [18]	40
Obrázek 24: Kód ovládající simulaci otáčení bubnem pračky [18]	40
Obrázek 25: Kód ovládající simulaci chladnutí pračky [18]	41
Obrázek 26: Návod tvorby knihoven – krok 1 [18]	45
Obrázek 27: Návod tvorby knihoven – krok 2 [18]	46
Obrázek 28: Návod tvorby knihoven – krok 3 [18]	46
Obrázek 29: Návod tvorby knihoven – krok 4 [18]	47
Obrázek 30: Návod tvorby knihoven – krok 5 [18]	47
Obrázek 32: Návod tvorby knihoven – krok 7 [18]	48
Obrázek 31: Návod tvorby knihoven – krok 6 [18]	48
Obrázek 33: Vložení knihoven do projektu – krok 1 [18]	49
Obrázek 34: Vložení knihoven do projektu – krok 2 [18]	49

Obrázek 35: Vložení knihoven do projektu – krok 3 [18].....	50
Obrázek 36: Vložení knihoven do projektu – krok 4 [18].....	50
Obrázek 37: Vložení knihoven do projektu – krok 5 [18].....	51
Obrázek 38: Vložení knihoven do projektu – krok 6 [18].....	51
Obrázek 39: Vložení knihoven do projektu – krok 7 [18].....	52
Obrázek 40: Vložení knihoven do projektu – krok 8 [18].....	52
Obrázek 41: Vložení knihoven do projektu – krok 9 [18].....	53
Obrázek 42: Vložení knihoven do projektu – krok 10 [18].....	53
Obrázek 45: Vložení knihoven do projektu – krok 13 [18].....	54
Obrázek 44: Vložení knihoven do projektu – krok 12 [18].....	54
Obrázek 43: Vložení knihoven do projektu – krok 11 [18].....	54
Obrázek 46: Obnovení archivovaného projektu krok – 1 [18].....	55
Obrázek 47: Obnovení archivovaného projektu – krok 2 [18].....	55
Obrázek 48: Obnovení archivovaného projektu – krok 3 [18].....	56
Obrázek 49: Obnovení archivovaného projektu – krok 4 [18].....	56

SEZNAM TABULEK

Tabulka 1: Vstupy a výstupy modulu posuvník	42
Tabulka 2 Vstupy a výstupy modulu křížovatka	43
Tabulka 3: Vstupy a výstupy modulu mísicí jednotka	44
Tabulka 4: Vstupy a výstupy modulu pračka	45

SEZNAM PŘÍLOH

Příloha P I: CD obsahující vytvořené projekty a knihovny

