

**Vyšší odborná škola, Střední škola, Centrum odborné přípravy,
Sezimovo Ústí**



ABSOLVENTSKÁ PRÁCE

Učební pomůcka k PLC Tecomat Foxtrot

VYŠŠÍ ODBORNÁ ŠKOLA, STŘEDNÍ ŠKOLA, CENTRUM ODBORNÉ PŘÍPRAVY
SEZIMOVO ÚSTÍ, BUDĚJOVICKÁ 421



ZADÁNÍ ABSOLVENTSKÉ PRÁCE

Student: **Lukáš Plášil**
Obor studia: **26-41-N/01 Elektrotechnika – mechatronické systémy**
Název práce: **Učební pomůcky k PLC Tecomat Foxtrot**

Zásady pro vypracování:

1. Navrhnete učební pomůcku k výuce předmětu PGS na bázi PLC Tecomat Foxtrot
2. Porovnejte Softwarové vybavení použijte Mosaic od firmy Teco a. s.
3. Jako podporu využijte simulační prostředí vývojového programu Mosaic od firmy Teco a. s.
4. Absolventskou práci vypracujte problémově ve struktuře odpovídající vědecké práci

Doporučená literatura:

- [1] ŠMEJKAL, L., A MARTINÁSKOVÁ, M., *PLC a automatizace*, 2003, ISBN 80-86056-58-9.
- [2] ŠMEJKAL, L., *PLC a automatizace 2*, 2005, ISBN 80-7300-0878-3.

Vedoucí práce: **Ing. Václav Šedivý, VOŠ, SŠ, COP, Sezimovo Ústí**
Odborný konzultant práce: **Ing. Ladislav Šmejkal, Csc., Teco Kolín a. s.**
Oponent práce: **Ing. Alexej Salzman, VOŠ, SŠ, COP, Sezimovo Ústí**

Datum zadání absolventské práce: **1.9.2011**

Datum odevzdání absolventské práce: **30.9.2012**


Ing. Václav Šedivý
(vedoucí práce)




Ing. František Kamlach
(ředitel školy)

Prohlášení:

Prohlašuji, že jsem tuto absolventskou práci vypracoval samostatně, s použitím odborné literatury uvedené v seznamu, který je nedílnou součástí této absolventské práce.

V Táboře _____

podpis _____

Poděkování:

Chtěl bych tímto hlavně poděkovat za odborné vedení Ing. Václavu Šedivému, cenné rady týkající se struktury práce Mgr. Miloši Blechovi, odborné konzultace při zpracování absolventské práce a jejího obsahu panu Ing. Ladislavovi Šmejkalovi, CSc. z firmy Teco a.s.

Anotace

Tato absolventská práce se zabývá vytvořením učební pomůcky k programování programovatelného automatu Tecomat Foxtrot v integrovaném vývojovém prostředí Mosaic v jazyce reléových schémat. Bude zde popis tohoto vývojového prostředí, dále zde budou i úlohy k řešení a k vyzkoušení nabytých znalostí.

Annotation

Diese Diplomarbeit beschäftigt sich mit Erstellung Lehrmitteln für Programmierung von Speicher Programmierbare Steuerung Tecomat Foxtrot in der Integrierte Entwicklungsumgebung Mosaic in der Sprache des Relais Schemata. Es wird eine Beschreibung der Entwicklungsumgebung, dann wird auch eine Lösung der Aufgabe werden, und zum Testen der erworbenen Kenntnisse.

Obsah

1	ÚVOD	- 1 -
2	ZÁKLADNÍ POJMY	- 3 -
2.1	Popis PLC	- 3 -
2.2	Historie PLC	- 3 -
2.3	Základní požadavky na PLC	- 4 -
2.4	Dělení programovatelných automatů.....	- 4 -
2.5	Technické (HW) vybavení PLC	- 8 -
2.6	Zpracovávání programu PLC.....	- 14 -
2.6.1	Základní pojmy programování.....	- 14 -
2.6.2	Zpracování programu PLC.....	- 15 -
2.6.3	Řešení úlohy řízení.....	- 16 -
2.7	Programování dle normy IEC 1131 – 3.....	- 20 -
2.7.1	Rozdělení programovacích jazyků	- 21 -
2.7.2	Programovací jazyky	- 22 -
3	Tecomat Foxtrot.....	- 27 -
3.1	Popis programovatelného automatu Tecomat Foxtrot.....	- 27 -
3.1.1	Výkon systému.....	- 28 -
3.1.2	Komunikace	- 29 -
3.1.3	Programování	- 30 -
4	Mosaic.....	- 31 -
4.1	Začínáme v programu Mosaic	- 33 -
4.1.1	Spuštění vývojového prostředí Mosaic.....	- 33 -
4.1.2	Založení nové skupiny projektů.....	- 34 -
4.1.3	Základní popis prostředí Mosaic	- 36 -
4.1.4	Hlavní menu prostředí Mosaic.....	- 37 -
4.1.5	Nastavení vstupů a výstupů.....	- 39 -
4.2	Programování v Mosaic v jazyce reléových schémat (LD).....	- 41 -
4.3	Ovládací prvky v Editoru LD	- 42 -
4.4	Úlohy k řešení	- 44 -
5	ZÁVĚR.....	- 46 -
6	SEZNAM LITERATURY	- 47 -
7	OBSAH CD.....	- 48 -
8	PŘÍLOHY	- 49 -

Seznam obrázků

Obrázek 1 Tecomat TC600	- 5 -
Obrázek 2 Tecoreg TR050	- 6 -
Obrázek 3 Tecomat Foxtrot	- 7 -
Obrázek 4 TC700	- 7 -
Obrázek 5 Blokové schéma PLC	- 8 -
Obrázek 6 Univerzální zapojení DI (pro střídavé, i stejnosměrné napětí).....	- 9 -
Obrázek 7 Zapojení DO	- 10 -
Obrázek 8 Zpracování logických signálů PLC	- 14 -
Obrázek 9 Rozšíření jednotlivých jazyků	- 21 -
Obrázek 10 Jazyk mnemokódů	- 22 -
Obrázek 11 Jazyk strukturovaného textu	- 23 -
Obrázek 12 Jazyk reléových schémat	- 24 -
Obrázek 13 Jazyk blokových schémat	- 25 -
Obrázek 14 Foxtrot CP-1005	- 27 -
Obrázek 15 Rozměry základního modulu CP-1005.....	- 28 -
Obrázek 16 Komunikace mezi moduly pomocí sběrnice TCL2.....	- 29 -
Obrázek 17 Spouštění vývojového prostředí Mosaic.....	- 33 -
Obrázek 18 Založení nové skupiny projektů	- 34 -
Obrázek 19 Okno - Základní výběr řídicího systému	- 34 -
Obrázek 20 Okno - Deklarace POU.....	- 35 -
Obrázek 21 Mosaic	- 35 -
Obrázek 22 Hlavní okno programovacího prostředí.....	- 36 -
Obrázek 23 Hlavní nabídka a nástrojová lišta s ikonami	- 37 -
Obrázek 24 Nástroj nastavení V/V	- 39 -
Obrázek 25 Vytvoření nového obvodu	- 41 -
Obrázek 26 Ovládací prvky v editoru LD.....	- 42 -

Seznam tabulek

Tabulka 1 Unifikované signály	- 12 -
Tabulka 2 Programovací jazyky.....	- 21 -
Tabulka 3 Význam ikon v grafické nástrojové liště hlavního menu.....	- 38 -
Tabulka 4 Informace o stavu PLC	- 38 -
Tabulka 5 Ovládací prvky Editoru LD.....	- 43 -

1 ÚVOD

PLC (Programmable Logic Controller) je číslicově pracující elektronický systém konstruovaný pro použití v průmyslovém prostředí, využívající programovatelnou paměť pro interní ukládání uživatelsky orientovaných instrukcí pro provádění specifických funkcí (logickým, sekvenčních, časovacích, čítacích, komunikačních apod.), za účelem řízení strojů či procesů, a to prostřednictvím digitálních, nebo analogových vstupů a výstupů.

Pomocí programovatelných logických automatů, je řešena naprostá většina průmyslových řídicích aplikací. V automatizační technice se programovatelné automaty používají zhruba od r.1970. Původně byly určeny

pro řízení strojů, jako náhrada za pevnou reléovou logiku. Postupně se jejich možnosti rozšiřovaly a dnes se s nimi můžeme setkat v nejrůznějších oborech, kde mnohdy vytlačují dříve používané přístroje. Jsou to nejenom tradiční strojírenské výrobní technologie včetně manipulační a dopravní techniky, ale i energetika (regulace v elektrárnách, v kotelnách, v klimatizačních jednotkách i chladicích zařízeních). Uplatnění mají programovatelné automaty rovněž i v chemických výrobách, farmacii, v zemědělských výrobnách apod.

Proto jsem si jako moji absolventskou práci vybral vytvoření učební pomůcky pro modulární programovatelný automat Tecomat Foxtrot od české firmy Teco a. s. z Kolína. V širším vymezení se práce zabývá nejen programovatelný automatem Tecomat Foxtrot,



ale i všeobecně programovatelnými automaty. Kapitola Základní pojmy se zabývá základním rozdělení PLC, popisem jednotlivých periférií PLC, programovacími jazyky...

Část Tecomat Foxtrot se zabývá samotným programovatelným automatem Tecomat Foxtrot od české firmy Teco a. s. Zde bude stručný popis tohoto PLC (výkon systému, komunikace s jeho moduly, či jeho programování).

Další kapitola se již zabývá programováním PLC Tecomat Foxtrot v integrovaném vývojovém prostředí Mosaic. Zde bude popsán Mosaic jako takový. Dále zde naleznete návod na založení nového projektu a programování v jazyce reléových schémat. Nakonec zde bude několik úloh pro studenty, jejichž řešení bude k dispozici v příloze.

2 ZÁKLADNÍ POJMY

2.1 Popis PLC

Nejčastější zkratkou je PLC (z anglického názvu Programmable Logic Controller), v německém jazyce pak SPS (Speicher Programmierbare Steuerung) a konečně v českém jazyce PA (programovatelný automat). Programovatelný automat (dále jen PLC) je číslicově řídicí systém určený pro řízení pracovních strojů a procesů v průmyslovém prostředí. PLC prostřednictvím číslicových, nebo analogových vstupů a výstupů získává a zároveň předává informace „z a do“ řízeného prostředí. Algoritmy řízení jsou uloženy v paměti uživatelského programu, který je cyklicky vykonáván.

2.2 Historie PLC

Vývoj PLC začal po roce 1968 na základě požadavků amerických automobilek. První programovatelné automaty byly ve výrobě použity v roce 1969 a počátkem 70. let se začínají PLC rozšiřovat do celého světa.

První programovatelné automaty a jejich programovací přístroje byly realizovány na bázi pevné logiky. Od 70. let 20. století se k jejich konstrukci používaly mikroprocesorové řadiče a později mikroprocesory. V počátcích éry PLC disponovaly omezeným počtem instrukcí (obvykle byly obdobnou funkcí relé), pomocí kterých se mohly sestavovat programy převážně logického typu.

Nejdříve se k zápisu a ladění programu PLC používaly pouze specializované programovací přístroje v podobě kufříků, nebo kalkulátorů, případně čelních panelů. Dnes jsou tyto přístroje jako alternativa ponechány jen u nejjednodušších logických modulů.

V 80. letech se vývoj PLC zaměřuje na minimalizaci rozměrů, standardizaci komunikačního protokolu a symbolické způsoby programování. S rozvojem PLC se začaly také množit různé komunikační protokoly, se kterými automaty pracovaly a tak v 90. letech sílí snaha o jejich redukci a sjednocení. Od 90. let 20. století přebírají úlohu programovacích přístrojů ve významné míře osobní počítače a to jak stolní, tak i přenosné, což se projevilo j jejich koncepcí. Základem se stal specializovaný vývojový diagram,

nebo-li počítačový program. Posledním používaný standardem pro programování automatů se stala norma IEC 1131-3, která ujednotila programovací jazyky a vnesla do narůstajícího chaosu řád [1].

2.3 Základní požadavky na PLC

Robustnost – hlavním požadavkem na PLC je spolehlivost, tak aby byl zajištěn bezpečný provoz řízené technologie. Jsou konstruovány tak, aby mohly pracovat i v těch nejobtížnějších provozních podmínkách v těsné návaznosti na řízenou technologii. To klade vysoké nároky na jejich odolnost vůči okolním vlivům (vlivům okolního prostředí) jako například teplota, prašnost, otřesy, vlhkost apod.

Programování – Nastavení základního programu často umožňuje „programovací panel“ připojitelný k základnímu řídicímu bloku. NORMA

Diagnostika – provoz PLC vyžaduje co nejsnazší a nejrychlejší možnost identifikovat a následně odstranit závadu. Funkčnost jednotlivých komponentů je proto zřetelně indikována (LED diody). Díky této indikaci může kvalifikovaná obsluha určit chybový stav.

Paměť – při poruše řídicího systému (výpadek proudu, chyba systému) si musí PLC na rozdíl od PC zapamatovat poslední stav, od něhož bude po obnovení funkce pokračovat dál.

Cena - PLC jsou z hlediska HW i SW jednodušší než ostatní způsoby řízení, proto i jejich cena bývá nižší

2.4 Dělení programovatelných automatů

Programovatelné automaty se vyrábějí v různých variantách. Podle velikosti je můžeme rozdělit do tří různých skupin:

- Kompaktní PLC
- Mikro PLC
- Modulární PLC

Kompaktní PLC

Kompaktní PLC jsou přístroje vše v jednom, které se vyznačují již v základní výbavě množstvím hardwarových a softwarových funkcí a jsou vhodné pro mnoho případů použití v oblasti řízení, regulace a měření. Pokud integrované funkce nestačí, je možné přístroje lokálně, nebo přes síť jednoduše rozšířit. Všechna řízení je možné zapojovat do sítě a programovat je přes provozní sběrnici. Do této skupiny patří například Tecomat TC600 firmy TECO a. s.



Obrázek 1 Tecomat TC600

Zdroj: <http://tecomat.com>

Mikro PLC

Nejmenší a také nejlevnější kompaktní PLC, neboli mikro PLC, nabízejí uživateli pevnou sestavu vstupů a výstupů. Použitím nejmenších PLC (mikro PLC) je realizace logické výbavy jednoduchých strojů a mechanismů, která se dřív řešila pevnou reléovou logikou. Velkým rozdílem jsou ceny ovládacích prvků (relé, stykačů) a je zjevné, že použití PLC je účelné již u nejprostších aplikací, kde se nahrazuje jen část relé.

K základnímu modulu lze připojit jeden, nebo několik přídatných modulů z omezeného sortimentu s pevnou kombinací vstupů a výstupů, např. modul s 8 binárním vstupy a 8 binárními výstupy, analogový vstupní a výstupní modul apod. Některé kompaktní systémy se navíc vyznačují ještě vnitřními modulárností, kdy konfiguraci základního modulu lze sestavit osazením základní desky zásuvnými moduly vhodného typu. . Do této skupiny patří například Tecomat TR050 firmy TECO a. s.



Obrázek 2 Tecoreg TR050

Zdroj: <http://tecomat.com>

Modulární PLC

Modulární, nebo také stavebnicové PLC se sestavují z jednotlivých modulů. Je tedy volnost a možnost sestavit individuálně pro určitou aplikaci. Skládá se z těchto modulů: centrální procesorové jednotky (CPU), systémové a uživatelské paměti, modulu napájení a záložního zdroje, vstupních a výstupních jednotek (pro připojení řízeného systému), komunikačních jednotek (pro komunikaci se souřadnými i nadřazenými řídicími systémy) a dalších speciálních jednotek. Moduly jde podle potřeby přidat, vyměnit apod. Do této skupiny patří například Tecomat Foxtrot , nebo TC700 firmy TECO a. s.



Obrázek 3 Tecomat Foxtrot

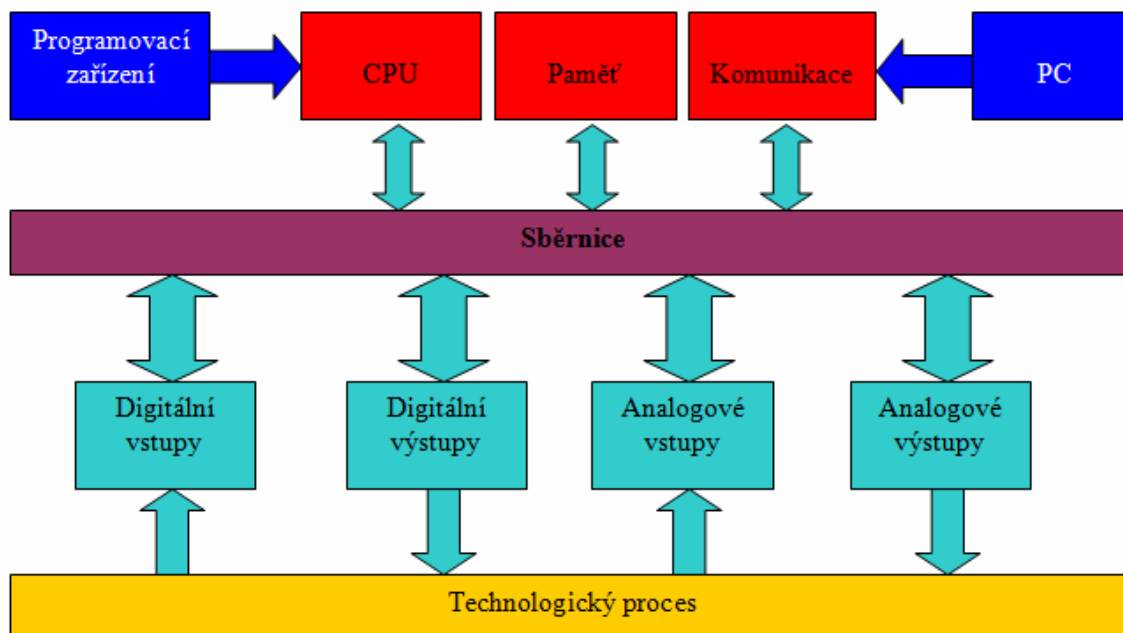
Zdroj: <http://tecomat.com>



Obrázek 4 TC700

Zdroj: <http://tecomat.com>

2.5 Technické (HW) vybavení PLC



Obrázek 5 Blokové schéma PLC

Centrální procesorová jednotka (CPU) a paměť

Centrální procesorová jednotka (CPU – Central Procesor Unit) je základem celého PCL která určuje jeho výkonnost. Může být jednoprocesorová, nebo víceprocesorová (matematické koprocesory, komunikační procesory, vstupně-výstupní procesory, rychlý bitový procesor apod.). Důležitým charakteristickým parametrem je operační rychlost posuzovaná podle doby cyklu nazýváme SCAN (základní parametr je doba zpracování 1000 logických instrukcí). Výrobci nabízejí pro dané typy různé CPU lišící se rychlostí, velikostí pamětí a tím i samozřejmě cenou.

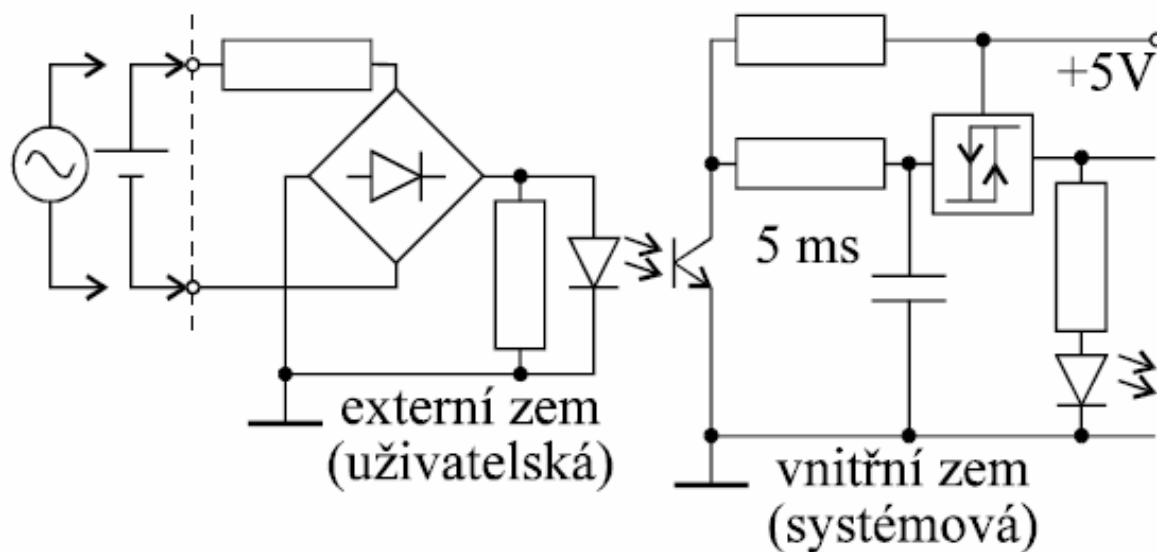
Paměť se může dělit na uživatelskou, systémovou, či paměť dat apod. Do uživatelské paměti se ukládá uživatelský program, bývá ty EPROM, nebo EEPROM a mívá kapacitu v desítkách kB, někdy i v jednotkách MB. V systémové paměti je umístěn systémový program. Tato paměť bývá též typu EPROM. V samostatné jednotce je umístěná přídatná paměť, která je uživatelská. Tato paměť je typu RAM (RWM). Jsou v ní umístěny pro uživatele dostupné registry, zápisníkové registry (flagy), čítače, časovače, či vyrovnávací registry pro obrazy vstupů a výstupů. Počet registrů výrazně ovlivňuje

možnosti PLC. Adresovatelný prostor vymezený pro vstupy a výstupy omezuje počet připojitelných periferních jednotek.

Dalším důležitým parametrem jsou rozsahy čítačů a časovačů. Na CPU bývá také umístěn jeden, nebo i více sériových komunikačních kanálů. Většina automatů dnes disponuje hodinami reálného času, případně i kalendářem. To znamená, že lze tyto údaje využívat při tvorbě programu (ovládání zařízení v určitou hodinu, či v určitý den).

Binární vstupy (DI)

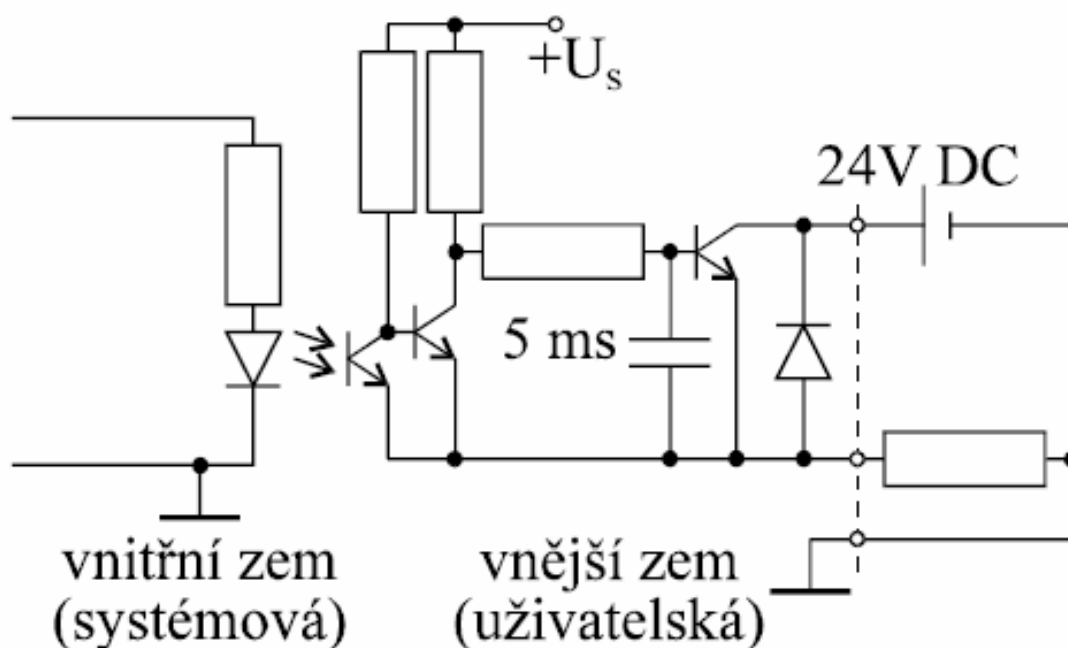
Slouží k připojování tlačítek, přepínačů, koncových spínačů a jiných snímačů s dvouhodnotovým charakterem signálu (např. dvouhodnotové snímače tlaku, teploty, či hladiny). Bývají v univerzálním provedení (pro stejnosměrné, či střídavé napětí). Kromě galvanického oddělení jsou vybaveny filtrem, pro odfiltrování poruchových signálů a diodami pro ochranu proti napěťovým špičkám.



Obrázek 6 Univerzální zapojení DI (pro střídavé, i stejnosměrné napětí)

Binární výstupy (DO)

Jsou určeny k buzení cívek relé, stykačů, elektromagnetický spojek, pneumatických a hydraulických převodníků, k ovládání signálek, ale i ke stupňovitému řízení pohonu a frekvenčních měničů. Digitální výstupy mohou být buď v nevýkonovém provedení s tranzistorem do max. stovek mA, ve výkonovém s tyristorem, nebo reléové pro spínání větších výkonů stejnosměrných, nebo střídavých. Pro stejnosměrné napětí jsou pro 5, 12, 24 a 48V, pro střídavé napětí pro 24, 48, 115 a 230V (unifikované signály).



Obrázek 7 Zapojení DO

Analogové vstupy (AI)

Slouží k připojení například snímačů teploty (obvykle odporových, mohou být i speciální pro zapojení polovodičových snímačů, či termočlánků), vlhkosti, hladiny, síly, polohy, rychlosti, ale i většinu inteligentních přístrojů s analogovými výstupy. Analogové vstupy můžeme dělit na napěťové, nebo na proudové. Nezbytnou částí je A/D převodník (šířka 8, nebo 10 nebo 14 bitů) podle přesnosti převodu. Někdy je galvanické oddělení pro ochranu PLC, tato ochrana se pak ale výrazně projeví na ceně PLC.

Analogové výstupy (AO)

Slouží k ovládní různých akčních členů, či zařízení se spojitým charakterem vstupního signálu, jako jsou např. spojitě servopohony, frekvenční měniče, ale i ručičkové měřicí přístroje apod. Nezbytnou částí je D/A převodník (šířka 8, nebo 10 bitů) podle přesnosti převodu. Analogové výstupy jsou napěťové, nebo proudové. Proudové výstupy se pak dělí na aktivní, nebo na pasivní. U aktivních nemusí mít akční člen zapojený svůj zdroj proudu a je napájen přímo z analogového proudového výstupu PLC. Tento připojený akční člen má v tomto případě omezený jeho max. odpor. U pasivních musí mít akční člen vlastní zdroj proudu.

Prvky pro tvorbu vstupních signálů

Těmi se rozumí ovládací prvky a snímače, které se nacházejí přímo na řízených strojích a zařízeních. Jsou to:

Kontaktní – tlačítka, spínače, koncové spínače, přepínače apod.

Bezkontaktní – elektronické, kapacitní, indukční a optické snímače.

Ostatní – snímače tlaku, síly, rychlosti, hladiny, průtoku, kouřových plynů, pasivní infrasnímače (PIR) apod.

Prvky pro realizaci výstupních signálů

Těmi se rozumí členy, které zesilují elektrické signály z PLC, nebo je přeměňují na jinou fyzikální veličinu. Jsou to:

Signalizační prvky - Akustické (houkačka, siréna, zvonek...)

- Optické (kontrolka, majáček, segmentovka...)

Spínací prvky - Relé, stykače, polovodičové spínací prvky

Unifikované signály

Jedná se o signály, které mají určité předem definované hodnoty. Požívají se z důvodu kompatibility měřicích a řídicích zařízení od různých výrobců.

Druh signálu	Hodnota
Proudový signál	0 – 20mA 4 – 20mA
Napěťový signál stejnosměrný	0 – 5V 0 – 10V -15 – 15V
Napěťový signál střídavý	24, 48, 110, 230V (logická 1)
Napěťový signál stejnosměrný	24, 48, 110, 230V (logická 1)
Digitální signál	Pro RS-232, RS-485

Tabulka 1 Unifikované signály

Komunikační jednotky

Důležitou vlastností PLC je schopnost komunikovat se vzdálenými moduly vstupů a výstupů, s podsystémy i nadřizenými systémy, s operátorskými panely a s jinými inteligentními přístroji, s počítači a s jejich sítěmi a tak vytvářet distribuované systémy. Komunikační jednotky většinou rozšiřují počet asynchronních sériových komunikačních kanálů. U některých systému jsou k dispozici i jednotky dálkových přenosů umožňující dálkové přenosy dat přes modem, nebo přes různé druhy sběrnice (např. ASI, PROFIBUS, RS 232, RS 422, RS 485).

Speciální jednotky

Obsahuje modul hodin, časovače, čítače sekvenční registry, nebo specializované moduly pro řešení regulačních úloh (např. PID regulátor), případně algoritmy pro regulace matematické funkce, nebo moduly pro řešení úloh fuzzy logiky a fuzzy regulace.

Časovače – odměřují délku časových intervalů, spouštějí a zastavují se binárním signálem. Výstup je binární (po proběhnutí času)

Čítače – Čítají vstupní pulsy, nebo vysílají pulsy na výstup. Tyto pulsy se vyskytují především u odměřování dráhy, rychlosti pohyblivých částí strojů a manipulačních mechanismů. Ovládají se binárním signálem a výstup je také binární (po odpočítání zadaného počtu pulsů)

Sekvenční registry – Mají za úkol posloupnost bitů, každý je adresovatelný. Po vložení binární informace na vstup, způsobí posun celé posloupnosti, obsah posledního bitu se ztrácí (dávají možnost vložit informaci a po určitém počtu kroků ji zase vyjmout a zpracovat).

Počítačové jednotky

Je to počítačový modul, který je kompatibilní s počítačem. Lze v něm standardními počítačovými prostředky řešit úlohy, které nejsou pro PLC typické, např. složité a rychlé výpočetní algoritmy, grafické a geometrické úlohy, zpracovávání a archivace velkého množství dat, databázové úlohy, výkonné komunikace, napojení PLC do počítačových sítí, připojení standardních počítačových periférií, paměťových karet apod. [2]

2.6 Zpracovávání programu PLC

2.6.1 Základní pojmy programování

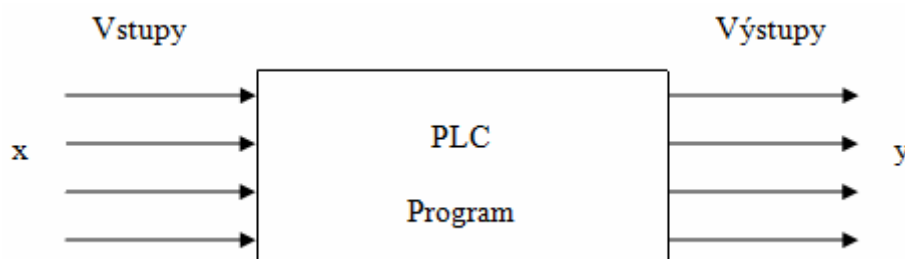
Řídící program – Skládá se z instrukcí, které řeší dané algoritmy zapsané ve formě srozumitelné pro daný řídicí systém.

Algoritmus – Je to metoda postupu řešení daného problému. Sestavení algoritmu je základní a také rozhodující tvůrčí činností při programování. Pokud se tedy sestavený algoritmus správně přepíše do srozumitelného pro řídicí systém, vznikne program. Algoritmus musí být srozumitelný a jednoznačný předpis, co je potřeba udělat, aby byla požadovaná úloha správně provedena.

Algoritmizace – je to hlavní, nikoli jediná etapa programování. Před vlastní algoritmizací úlohy je třeba danou úlohu vhodně a jednoznačně formulovat, následně vytvořit vhodný model a zvolit vhodnou metodu řešení problému. Po sestavení algoritmu je nutné ho přepsat do zvoleného programovacího jazyka.

Program obsahuje posloupnost instrukcí, kterou procesor vykonává cyklicky. Chování PLC je v podstatě dáno zaměnitelným programem, kdežto u reléových systémů bylo určeno strukturován zapojením, které bylo nezaměnitelné.

PLC zpracovává vstupní logické signály pomocí programu na výstupní viz Obr. 8.



Obrázek 8 Zpracování logických signálů PLC

Popis činnosti PLC lze také popsat vztahem: $Y = f(X)$,

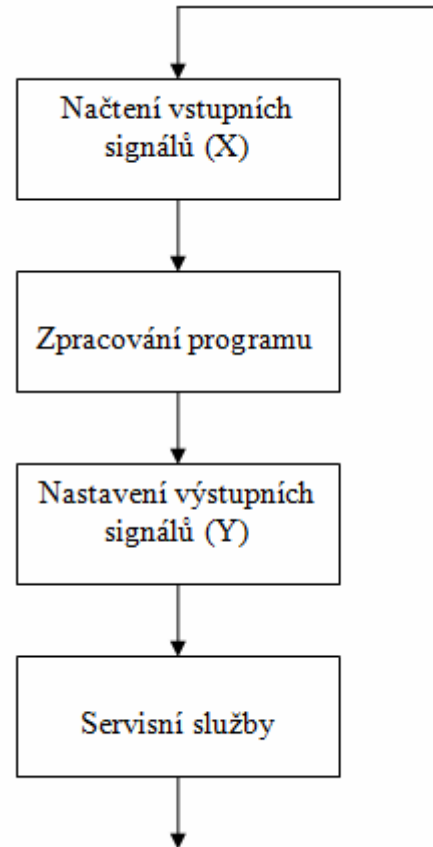
kde hodnota funkce f je dána vlastním programem řešící daný problém.

2.6.2 Zpracování programu PLC

PLC vykonává program v periodicky se opakujícím uzavřeném cyklu, nebo-li **Scanu**. Nejprve dojde k načtení vstupních veličin, dále dojde k sériovému zpracování jednotlivých instrukcí programu. Po jejich zpracování se data paralelně předají na výstup.

Vzhledem k rychlosti zpracování instrukcí se PLC navenek jeví jako paralelní systém (vše se zpracovává navzájem), ve skutečnosti se však data zpracovávají sériově. Za běhu programu je však nutné eliminovat účinek změn na vstupech a výstupech. Na základě různého stavu vstupů a výstupů a jejich vyhodnocení na začátku a na konci programu by se mohli objevit nežádoucí reakce PLC. Proto PLC pracuje s pamětí, do které jsou uloženy před každým zpracováním

programu stavy vstupů a výstupů. Proto si PLC po dobu jednou oběhu programu (cyklu) udržuje hodnoty vstupů a výstupů konstantní bez ohledu na právě probíhající děj a případné změny vstupů a výstupů.



Cyklická činnost PLC může být rozdělena na čtyři úseky:

- Načtení všech vstupů a zápis hodnot do paměti
- Vyhodnocení jednotlivých podmínek na základě vstupních, výstupních a vnitřních proměnných. Po vyhodnocení se zapíše do paměti výstupů nové hodnoty výstupních proměnných.
- Podle uložených výstupních hodnot se po ukončení běhu programu jednorázově nastaví výstupní veličiny, které pak aktivují akční členy.
- Na závěr scanu se vyhodnotí stavový soubor a nastaví se jeho aktualizované parametry, případně se i vyřídí komunikace s ostatními zařízeními v síti, čímž je PLC připraven na další scan.

2.6.3 Řešení úlohy řízení**Řešení úlohy řízení lze rozdělit na několik etap:**

1. Formulace úlohy
2. Analýza úlohy
3. Algoritmizace úlohy
4. Napsání programu
5. Provedení programu
6. Ladění programu
7. Zpracovávání dokumentace programu
8. Instalace řídicího systému u zákazníka a zkušební provoz
9. Předání systému do provozu

1) Formulace úlohy

První etapa má velký význam pro úspěšné zahájení řešení problému. Musí se v ní stanovit s jakými vstupními daty se bude pracovat a jakých hodnot mohou nabývat. Dále se musí formulovat problémy, které s těmito vstupními daty máme řešit a stanovit cíle, kterých má být dosaženo. Zpracovává se slovně. Zadání zpravidla definuje budoucí uživatel s někým, kdo rozumí dané problematice, tak možnostem počítače. Je bezpodmínečně nutné, aby zadání bylo srozumitelné, úplné a jednoznačně formulované.

2) Analýza úlohy

Tato etapa analyzuje úlohu, ověřuje se její řešitelnost a vytváří se první postupy řešení úlohy. Na závěr této etapy by mělo být jasné, zda je úloha řešitelná na daném počítači. Výsledkem je formulace postupu obecného řešení úlohy – vytvoření matematického, nebo logického modelu úlohy.

Obecné principy analýzy úlohy

Abstrakce – Proces je abstrakcí reality, zanedbání, vyloučení vlastností, které nejsou podstatné z hlediska sledovaného účelu.

Dekompozice – Rozklad problému na dílčí jednotky a prvky. Ty se pak řeší odděleně, důležité je určit úroveň rozkladu.

Hierarchie – Zachycení souvislostí, tzn. co se má řešit v jakém pořadí.

Modularita – Dekompozice řešení a menší části. Moduly vykazují jistou samostatnost a mají minimum vazeb na své okolí.

3) Algoritmizace úlohy

V této etapě se zpracuje podrobný postup řešení úlohy na daném řídicím systému, tzn. sestaví se algoritmus řešení. Správně sestavený algoritmus je základním předpokladem pro úspěšné řešení úlohy. Zápis algoritmu se provádí nejprve ve slovní formě. Vytvoří se posloupnost všech dílčích operací, které má řídicí systém vykonat.

Často bývá pro řešení zadané úlohy k dispozici hned několik algoritmů. Potom se volí ten, který je v nějakém smyslu výhodný (obsazení paměti, rychlost výpočtu apod.). Slovní popis algoritmu je zejména u složitějších úloh nepřehledný a zdlouhavý. Používají se proto stručnější přehlednější formy zápisu v grafickém tvaru.

4) Napsání programu

Tato etapa spočívá v přepisu algoritmu řešení do jazyka vhodného ke zpracování na počítači. Takovému jazyku říkáme programovací jazyk. Algoritmus napsaný v programovacím jazyku se nazývá zdrojový program.

5) Provedení programu

Napsaný program je přenesen do řídicího systému obvykle prostřednictvím klávesnice, nebo nějakého druhu komunikační sběrnice. Při provádění programu jsou vykonávány všechny příkazy a na základě zadaných vstupních dat obdržíme výstupní data (výsledky). Během zpracování se ale mohou vyskytnout chyby, které musíme opravit, což je náplní další etapy.

6) Ladění programu

Cílem této etapy je prokázat, že je program správný, tj., že za všech okolností (se všemi vstupními daty) realizuje výpočetní proces algoritmu řešení. Ladění programu se skládá ze dvou etap:

- Odstranění formálních nedostatků v programu (syntaktické chyby)
- Ověření, zda má program správnou funkci

Etapa ladění je ze všech etap nejpracnější, jelikož se musí najít a opravit všechny chyby a hrozí nebezpečí vzniku dalších chyb při opravách těch předchozích. Časová náročnost je taktéž velká. Obecně bylo při programování výzkumy dokázáno, že i odladění program obsahuje chyby (cca 2 chyby na 1000 instrukcí/řádků programu).

7) Zpracování dokumentace programu

Jsme-li přesvědčeny, že program je správný, je potřebné vypracovat přehlednou a srozumitelnou uživatelskou a technickou dokumentaci (tzn. návod k použití programu). Dokumentace se obvykle skládá ze dvou částí:

- **Uživatelská dokumentace**
 - Název programu a jméno autora (opis zadání)
 - Popis funkce programu, přesnosti
 - Návod k obsluze
 - Způsob přípravy a tvar vstupních dat
 - Ovládání programu (pokyn pro uživatele)
 - Tvar výstupních dat
 - Nároky na vybavení počítače

- **Technická dokumentace**

Může obsahovat obdobné údaje jako uživatelská dokumentace, ale musí obsahovat i některé další údaje:

- Použité metody řešení a hrubí popis algoritmu
- Opis zdrojového programu a jeho grafické vyjádření
- Soubory zkušebních vstupních dat a získané výsledky

Tuto etapu nelze podceňovat, protože kvalitativně zpracovaná uživatelská dokumentace může sehrát velkou roli v průběhu využití systému uživatelem. Význam má samozřejmě i kvalitně zpracovaná technická dokumentace. Do programu se snadněji zavádějí změny, pokud máme z čeho vycházet a nemusíme luštit zdrojové kódy. To má zejména význam, pokud změny provádí jiný programátor než autor, nebo když se k němu vrátíme po delší době.

8) Instalace řídicího systému u zákazníka a zkušební provoz

Jedná se o důležitou etapu sloužící k doladění systému v reálném provozu za dohledu instalující firmy a zákazníka. V rámci možností reálného řízeného systému by měl být systém vyzkoušen v běžném provozu, v mezních stavech a měli by být simulovány i předvídatelné chybové stavy. Pokud zákazník v této fázi požaduje úpravy funkčnosti řídicího systému, tak by tyto úpravy měly být řešeny v rámci víceprací.

9) Předání do provozu

Finální část projektu, kdy je fungující zařízení předáno do provozu zákazníkovi včetně dokumentaci a je provedeno finální zaškolení obsluhujícího personálu. O předání do provozu je sepsán protokol [2].

2.7 Programování dle normy IEC 1131 – 3

Dokud řešily PLC jen základní úlohy, vystačily jen s několika jednoduchými instrukcemi v jazyku kontaktních schémat, nebo mnemokódů. Takže pro programátora nebyl takový problém se i v cizích programech zorientovat. Nešlo sice o kompatibilitu a už vůbec ne o přenositelnost programů, ale podobnost systémů umožňovala porozumění a snadnou adaptaci programátorů. Lze to přirovnat k podobnosti nářečí, nebo blízkých jazyků (např. jako čeština se slovenštinou apod.). S rostoucími požadavky na PLC si každá firma jazyk nějakým způsobem obohatila a jazyky se začaly stále více odlišovat. Jazyky měli sice určitou podobnost, ale bylo potřeba se na naučit každý jazyk zvlášť a tak postupně v používání PLC nastalo zmatení jazyků. Proto bylo potřeba vytvoření normy která by sjednotila požadavky na provedení PLC a na jejich programování. V roce 1993 bylo publikováno první vydání mezinárodní normy IEC 1131, která sjednocuje požadavky na PLC. Později byla uvedená norma přijatá i jako evropská IEC/EN 61131-3. Následně byla přijata, nikoli přeložena i jako česká norma pod označením ČSN EN 61131-3 Programovatelné řídicí jednotky – Část 3: Programovací jazyky. Tato norma se zabývá obecným programováním automatizačních prostředků. Tuto normu přijalo také jako směrnici většina významných výrobců PLC.

Norma IEC 1131-3 definuje následující požadavky na vlastnosti programového vybavení:

1. Nezávislost na technickém vybavení (přenositelnost programů)
2. Jednotný programátorský přístup (stačí jedno školení pracovníků na jeden typ PLC)
3. Možnost ladění softwaru již ve fázi návrhu (chybu programu je žádoucí odhalit již před testováním v provozu při uvádění technologie do výroby)
4. Strukturovanost a modularita (logický systém bývá začleněn do složitějšího celku, ve kterém je nutné udržovat přehled

Tato část normy (3. část) doporučuje 4 programovací jazyky s přesně definovanou syntaxí a sémantikou: LD, FBD, IL, ST. Jako pátý programovací jazyk se často uvádí SFC, který však v normě zařazen mezi jazyky, ale mezi tzv. společnými prvky, neboť tvoří jakousi nadstavbu pro strukturování celé aplikace [3].

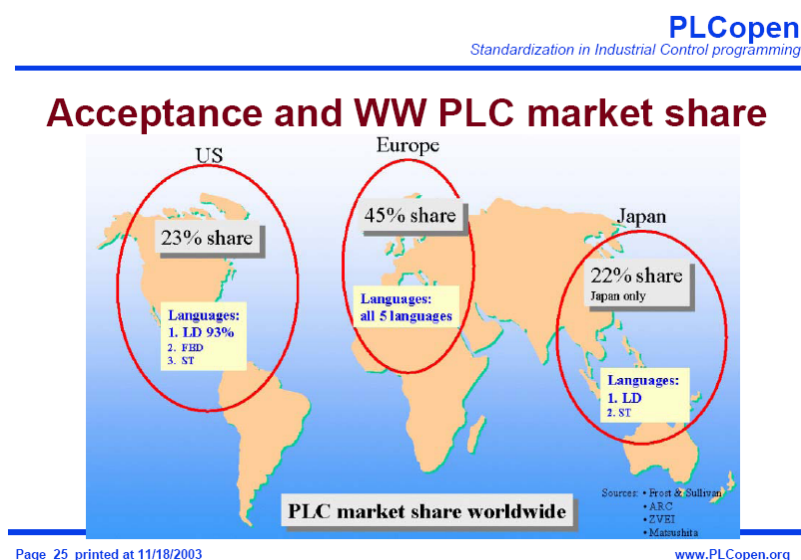
2.7.1 Rozdělení programovacích jazyků

Programovací jazyky lze rozdělit do dvě skupin:

1. Textové jazyky
 - a. Jazyk mnemokódů (IL)
 - b. Jazyk strukturovaného textu (ST)
2. Grafické jazyky
 - a. Jazyk kontaktních (reléových) schémat (LD)
 - b. Jazyk blokových schémat (FBD)
 - c. Jazyk sekvenčního programování (SFC)

Kodifikované označení v angličtině		Označení obvyklé v němčině		Název v češtině	
Zkratka	Název	Zkratka	Název	Zkratka	Název
IL	Instruction List	AWL	Anweisungsliste	IL	Jazyk mnemokódů
ST	Structured Text	ST	Strukturierter Text	ST	Strukturovaný text
LD	Ladder diagram	KOP	Kontaktplan	LD	Reléové schéma
FBD	Function Block Diagram	FUP	Funktionplan	FBD	Jazyk blokových schémat
SFC	Sequential Function Chart	AS	Ablaufsprache	SFC	Jazyk sekvenčního programování

Tabulka 2 Programovací jazyky



Obrázek 9 Rozšíření jednotlivých jazyků

Zdroj: <http://automatizace.cz>

2.7.2 Programovací jazyky

Jazyk mnemokódů (IL)

Jazyk mnemokódů patří do skupiny textových jazyků, je označován také jako jazyk pokynů (povelů) je obdobou assembleru u počítačů. To znamená, že každé instrukci PLC systému odpovídá stejně pojmenovaný příkaz jazyka. Tento jazyk poskytuje obvyklý „asemblerový komfort“, to je aparát symbolického označení návěští pro cíle skoků a volání, symbolická jména pro číselné hodnoty, pro pojmenování vstupní, výstupních a vnitřních proměnných a jiných objektů programu (datových tabulek a bloků, struktur a jejich prvků), pro automatické přidělování paměti pro uživatelské registry a pro jiné datové objekty, pro jejich inicializaci (zadání počátečního obsahu), pro zadávání číselných hodnot v různých číselných soustavách.

Každá organizační jednotka je složena ze sekvence instrukcí, z nichž každá začíná na novém řádku a může obsahovat návěští, které je nepovinné, ukončené dvojtečkou, operátor (např. LD, AND, ADD, CAL apod.), který může být doplněn modifikátorem, operátor, a někdy také komentářem, který je nepovinný. Pomocí modifikátorů se vyjadřují negace, podmíněnost a nepodmíněnost instrukce skoků, volání, návratů a priorit. Tyto jazyky jsou často používány profesionálními programátory.

```
0 LDN %I0.1
1 AND( %I0.0
2 OR %Q0.0
3 )
4 ANDN %Q0.1
5 ANDN %M1
6 ST %Q0.0
7 LDN %I0.1
8 AND( %I0.2
9 OR %Q0.1
10 )
11 ANDN %Q0.0
12 ANDN %M1
13 ST %Q0.1
14 BLK %TMO
12 LD %I0.1
13 ST %Q0.1
14 BLK %TMO
15 LD %I0.1
16 IN
17 OUT. BLK|
18 LD Q
19 ST %M1
20 END_BLK
```

Obrázek 10 Jazyk mnemokódů

Zdroj: <http://automa.cz>

Jazyk strukturovaného textu (ST)

Jazyk strukturovaného textu patří do skupiny textových jazyků a je obdobou vyšších programovacích jazyků pro PC (Pascal, nebo C). Umožňuje úsporný a názorný zápis algoritmů. Syntaxe je dána povolenými výrazy a příkazy. Vyhodnocením výrazu vyjde hodnota v některém z definovaných datových typů. Takový výraz se skládá z operátorů a operandů. Operandem může být například nějaká konstanta, funkce, proměnná, či jiný výraz. Operátory pro tento jazyk jsou nadefinovány pro 17 typů operací (booleovské funkce OR, NOR, XOR, AND, NAND, vyhodnocení funkce, násobení apod.). Příkazů pro tento jazyk je nadefinováno 10 (přiřazení, vyvolání funkce, návrat, výběr, větvení, cyklus apod.). Jednotlivé příkazy jsou odděleny středníkem a na jednom řádku jich může být i více. Jazyk strukturovaného textu je vhodným nástrojem pro definování komplexních funkčních bloků, které pak mohou být použity v libovolném programovacím jazyku.

```
! (*Hnací válec pro plnění boxu 1 a 2 *)  
  
SET %M4;  
RESET %M5;  
  
IF %M1 and %M3 THEN SET %M6;  
ELSIF %M1 AND %M2 THEN %M6:=%M2 OR %M3;  
ELSE RESET %M6;  
END_IF;  
  
WHILE %M1 DO %MW1:=%MW1+1;  
END_WHILE;  
  
REPEAT %MW2:=%MW2-1;  
UNTIL %MW2>3;  
END_REPEAT;  
  
FOR %MW5:=1 TO 50 DO %MW3:=%MW3+1;  
END_FOR;
```

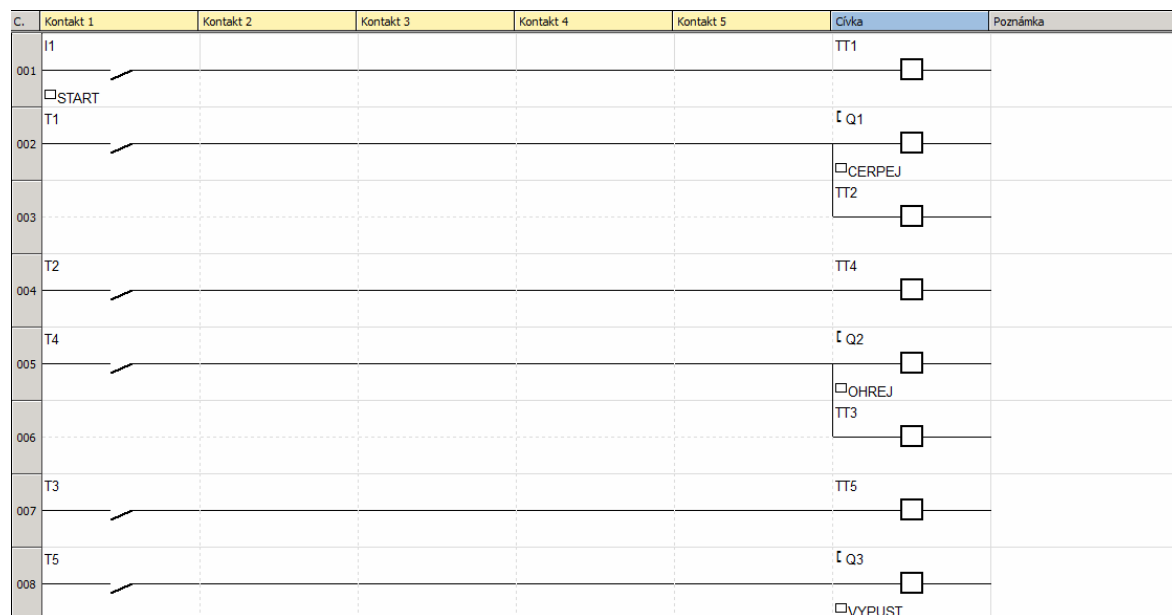
Obrázek 11 Jazyk strukturovaného textu

Zdroj: <http://automatizace.cz>

Jazyk reléových schémat (LD)

Jazyk reléových schémat, nebo-li kontaktních schémat patří do grafických jazyků. Je založen na reprezentaci reléové logiky. Svou grafickou podobou může připomínat žebřík, odtud anglický název Ladder. Organizační jednotka programu je vyjádřena sítí propojených grafických prvků. Tato síť je zleva, i zprava ohraničena svislými čarami, které se nazývají jako pravá a levá napájecí sběrnice. Program se základními logickými operacemi zobrazuje schéma ve formě obvyklé pro kreslení schémat při práci reléovými a kontaktními prvky. Kontakty a cívky se znázorňují zjednodušeně, spínací kontakt vypadá jako dvojice svislých čárek, rozpínací kontakt je navíc přeškrtnut a cívky jsou se označují dvojicí závorek. Funkční bloky, jako například čítače, nebo časovače mají obdélníkové značky.

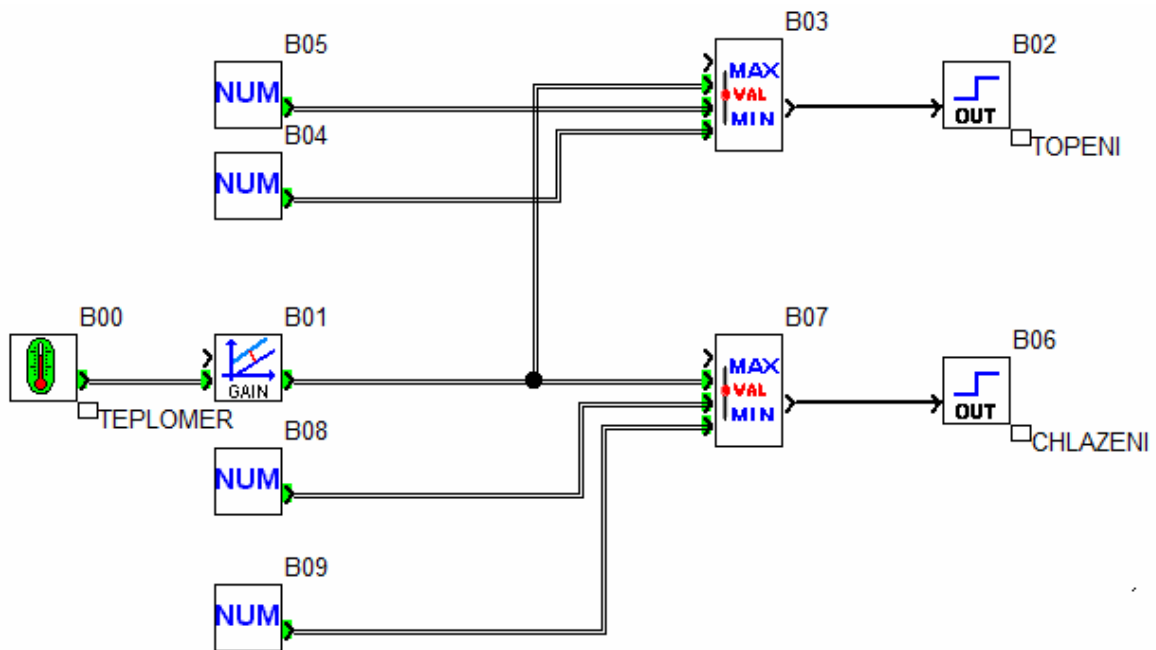
Tento jazyk je výhodný při programování nejjednodušších logických příkladů a v případech, kdy s ním pracuje personál, který nezná tradiční počítačové programování. Je nejvýhodnější při požadavku rychlého servisu, obzvláště pokud ladící prostředky dovolují zvýraznit na schématu „vodivou cestu“. Pak je nalezení závady na přístroje otázkou několika minut. Pokud v programu převažují složitější instrukce, pak je kontaktní schéma už méně vhodné, protože ztrácí svou názornost.



Obrázek 12 Jazyk reléových schémat

Jazyk blokových schémat (FBD)

Jazyk blokových schémat patří do skupiny grafických jazyků, který základní logické operace popisuje obdélníkovými značkami. Tyto jednotlivé bloky jsou navzájem propojeny do sekvence, která se dá velice snadno sledovat. Používá stejné instrukce jako jazyk reléových schémat, ale vizuálně je ještě o něco přehlednější. Výhodou je stejně jako u předchozího jazyku jednoduchost nalezení závady (stačí jednoduše sledovat cestu). Tento jazyk je ideální pro jednodušší programy tvořené digitálními vstupy (např. jako vstupy fotoelektrické snímače a jako výstupy rozváděcí ventily). Nicméně je méně vhodný pro velké programy a pro programy využívající speciální vstupy a výstupy. Nevhodné jsou pro velký požadovaný prostor na obrazovce a program se tak může stát za chvíli těžkopádný a nepřehledný. Výběr tohoto jazyku vyžaduje větší předběžnou přípravu pro pochopení programu a jeho toku ještě před samotným psaním jakéhokoli programového kódu, protože později už může být velice obtížné provádět korekce programu.



Obrázek 13 Jazyk blokových schémat

Jazyk sekvenčního programování (SFC)

Tento jazyk patří také do skupiny grafických schémat a popisuje sekvenční chování programu. V normě sice není označen jako jazyk, přesto jej ale lze považovat za prostředek vhodný jak k popisu činnosti programu již v počátečních fázích analýzy a zadávání požadavků na řídicí program, tak i k zápisu algoritmu řízení technologické soustavy. Kdyby si tento nepřiliš komplikovaný grafický program osvojili již konstruktéři a projektanti, mohli by své požadavky na řízení formulovat podstatně přesněji a komplexněji než při dosud nejobvyklejším slovním popisu.

Je odvozen ze symboliky Petřino sítě, ale liší se tím, že grafická reprezentace se zde převádí přímo do souborů výkonných řídicích prvků. Umožňuje rozložit úlohu řízení na zvládnutelné části a zachovat při tom přehled o chování celku. Sekvenční funkční diagram se skládá z kroků a přechodů. Každý krok reprezentuje stav řízeného systému a má k sobě přiřazen blok akcí. Přechod je spojen s podmínkami, které musí být splněny, aby mohl být deaktivován krok, který přechodu předchází, a naopak aktivován krok, který následuje. Každý prvek, tzn. přechod i blok akcí, může být naprogramován v libovolném jazyku definovaném v normě, včetně vlastního SFC. Jazyk umožňuje i větvení programu se spojením alternativních větví a paralelní souběh více větví s jejich následnou synchronizací [3].

3 Tecomat Foxtrot

3.1 Popis programovatelného automatu Tecomat Foxtrot

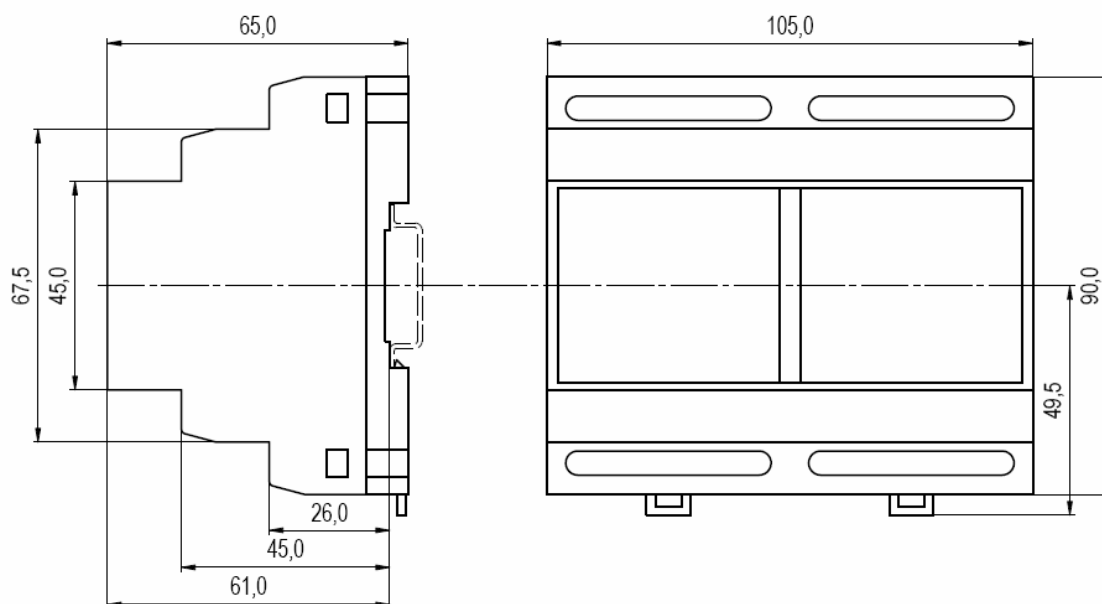
Programovatelné automaty Tecomat Foxtrot jsou malé kompaktní automaty s možností modulárního rozšíření od české firmy Teco a. s. Spojují tak výhody kompaktních automatů co do velikosti a modulárních automatů co do rozšiřitelnosti a variability.

Tyto automaty jsou určeny pro řízení v různých oblastech průmyslu, ale i v jiných odvětvích (vytápění, klimatizace, řízení osvětlení, fotovoltaické elektrárny, inteligentní budovy, výrobní linky apod.). Jednotlivé moduly jsou uzavřeny v plastových ochranných pouzdrech a se svým desingem s malými rozměry se montují přímo do rozvaděče na tzv. U-lištu. Díky tomu s nimi lze manipulovat bez nebezpečí poškození citlivých součástek.



Obrázek 14 Foxtrot CP-1005

Zdroj: <http://tecomat.com>



Obrázek 15 Rozměry základního modulu CP-1005

3.1.1 Výkon systému

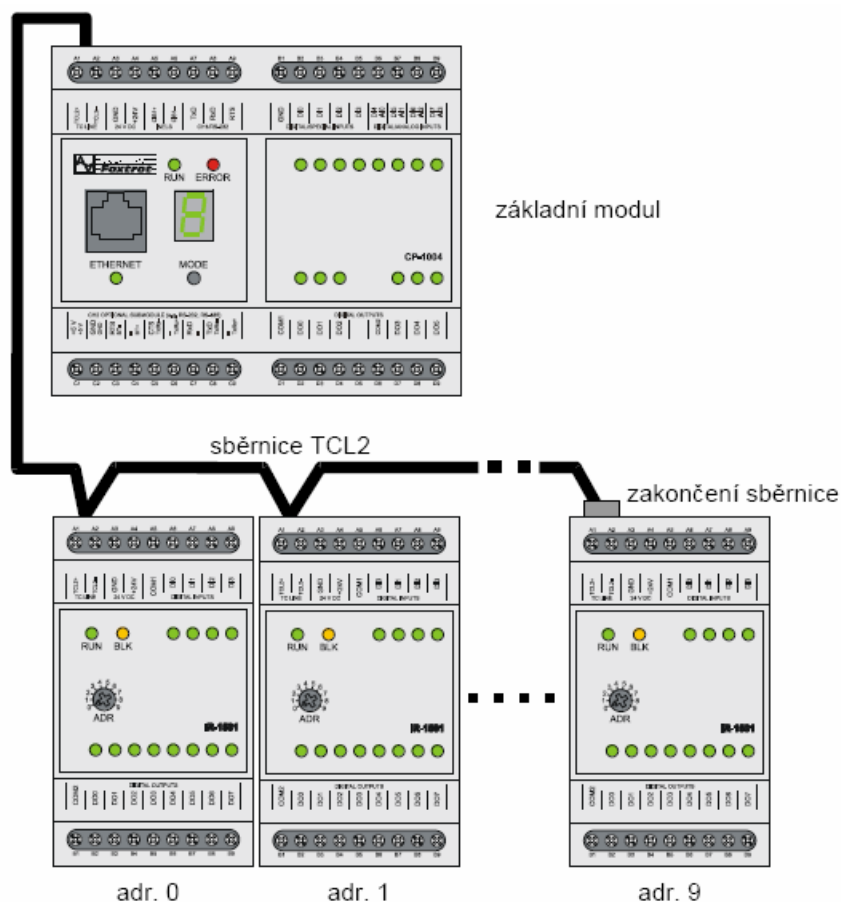
Tecomat Foxtrot má výkonnou procesorovou jednotku, kterou zajišťuje 32 bitový procesor RISC s frekvencí 166MHz, díky němuž dosahuje doby cyklu 0,2 ms na 1000logických instrukcí. Paměť pro samotný program je 192 kB a 64 kB pro tabulky. Paměť programu i tabulek je zálohována lithium-iontovým akumulátorem. Pro ukládání dat lze využít přídavné paměti DataBox, nebo je jej možné osadit kartou MMC/SD.

Tecomat Foxtrot díky svému vysokému výkonu zvládá řešení různorodých úloh průmyslové automatizace, ale je zároveň vhodný pro malé lokální úlohy s několika vstupy a výstupy. Dále je vhodný vedle průmyslových aplikací také v oblasti řídicích systémů v budovách, jako je řízení energetických zdrojů, monitoring a optimalizace spotřeby energií a také zajišťuje řízení jednotlivých místností, např. vytápění, klimatizace, řízení osvětlení, nebo spínání, či odpínání zásuvkových okruhů v závislosti na zadaném časovém programu, přítomnosti osob a dalších jiných parametrech.

3.1.2 Komunikace

Moduly řídicího systému jsou propojeny sběrnicí TCL2. tato sběrnice je propojena metalickými kabeli odpovídá rozhraní RS-485 a musí být na obou koncích ukončena. Základní modul obsahuje její zakončení a vždy musí být na jejím konci. Na druhém konci musí být sběrnice k poslednímu modulu připojena společně se zakončovacím členem KB-0290. Pomocí této sběrnice lze systému připojit až 10 rozšiřujících modulů I/O modulů.

Základní modul dokáže komunikovat s okolím pomocí integrovaných rozhraní jako jsou např. Ethernet, RS-232, nebo jiného volitelného rozhraní, které je realizováno osazením příslušného submodulu do základního modulu. Tecomat Foxtrot dále obsahuje webový server, který dává možnost vytvořit pro každý modul vlastní webovou stránku v jazyce XML. Prostřednictvím této stránky lze daný proces nejen monitorovat, ale i řídit z libovolného místa s připojením na internet [4].



Obrázek 16 Komunikace mezi moduly pomocí sběrnice TCL2

Zdroj: <http://tecomat.com>

3.1.3 Programování

Jako programovací nástroj lze požit počítač na kterém musí být nainstalováno vývojové prostředí Mosaic, který bude popsán v následující kapitole. Tecomat Foxtrot podporuje programování podle normy IEC EN 1131-3 (viz.Kap 2.7 Programování dle normy IEC 1131 – 3), což znamená, že při programování lze využít jak textových (jazyk mnemokódů(IL) a jazyk strukturovaného textu (ST)), tak i grafických jazyků (jazyk reléových schémat (LD), jazyk blokových schémat (FBD)).

4 Mosaic

Mosaic je integrované vývojové prostředí pro tvorbu a ladění programů pro PLC firmy Tecomat a Tecoreg firmy Teco a. s a je zdarma k dispozici na internetové stránce www.tecomat.cz. Kolín. Mosaic je dodáván od roku 2000 a je ve shodě s mezinárodní platnou normou IEC EN 61131-3, která definuje strukturu programu programovací jazyky pro PLC. Tento program je dodáván stylem „all in one“, což znamená, že instalace již obsahuje všechny nástroje, které jsou v tu dobu k dispozici. Pokud nemáme hardwarový klíč, Mosaic funguje ve verzi „Lite“, která umožňuje programovat bez omezení menší řady PLC Tecomat, i Tecoreg. Ve verzi „Lite“ jsou všechny nástroje plně funkční, pokud bychom chtěli programovat větší typy PLC, budeme potřebovat hardwarový klíč, který umožní deklarování většího počtu I/O modulů.

Jak už bylo řečeno, program Mosaic umožňuje programovat podle normy IEC EN 61131-3 jak v textových jazycích (jazyku instrukcí – IL, a strukturovaného textu – ST), tak i v grafických jazycích (reléových schématach – LD, funkčních bloků – FBD). Program v jazyce se podle normy IEC EN 61131-3 skládá z jednotlivých elementů, které se nazývají programové organizační jednotky POU (anglická zkratka z Program Organisation Unit). Těmito jednotkami jsou funkce, funkční bloky a tou nejvyšší je samotný program. Při psaní programu je možné jednotlivé jazyky kombinovat, ale jakmile je však zvolen pro zápis POU určitý jazyk, není možné ho již měnit. Ale další POU už může být zapsán v jiném jazyce. To nám umožní si rozčlenit program například na 2 části. Jedna část řídicí logiky může být v jazyce LD a druhá část, kde se budou provádět matematické výpočty a složité výrazy v jazyce ST.

Deklarační část programu je všem jazykům společná. Podporovány jsou všechny datové typy definované výše uvedenou normou včetně datových typů pro práci s časem, datem, nebo řetězcem. Deklarace vlastních datových typů včetně struktur a polí je podporována stejně jako deklarace všech typů POU. Uživatel Mosaic má rovněž možnost vytvářet vlastní knihovny POU.

Jedinečnost programu Mosaic spočívá především v mnoha dalších nástrojích, které jsou také jeho součástí a jejichž úkolem je zjednodušit a zrychlit programování typických situací, respektive poskytnout další informace ve fázi ladění programu, které nám pomáhají při analýze činnosti programu. Například to mohou být tyto nástroje:

- **Inspektor POU**

Tento nástroj umožňuje sledovat chování jednotlivých POU. U zobrazené POU jsou zobrazeny stavy jak vstupních, tak i výstupních proměnných. U grafických editorů je barevně označena aktivní linie a zobrazeny hodnoty proměnných u vstupů a výstupů jednotlivých bloků.

- **SimPLC – simulátor PLC**

Umožňuje ladit programy bez nutnosti připojení reálného PLC. Tento nástroj dovoluje odsimulovat všechny vyráběné PLC Tecomat. Unikátní je možnost komunikace mezi prostředím Mosaic a vizualizačním systémem. Lze zde tedy ladit nejen program pro PLC, ale i vizualizaci a jejich vzájemnou interakci, přičemž oba programy mohou běžet na jednu PC.

- **GraphMaker – nástroj pro sledování průběhů proměnných**

Tento nástroj se chová jako šestikanálový osciloskop s možností sledovat průběhy všech typů proměnných v reálném čase. Nasnímané průběhy si lze ukládat na pevný disk, dále pak tisknout a exportovat do dalších programů (např. Excel), nebo je přímo analyzovat.

- **PIDMaker – nástroj pro definici a sledování regulačních smyček**

Tento nástroj je určen ke snadné implementaci a ladění regulačních algoritmů. Nástroj automaticky generuje kód algoritmu, který je zařazen do programu uživatele. Dokáže také simulovat jednoduché procesy bez nutnosti jejich implementace v programu uživatele. V nabídce jsou nástroje lineární soustavy do 3. řádu s možností přidání dopravního zpoždění [5].

- **WEBMaker**

Tento nástroj je určený primárně pro tvorbu webových stránek pro systémy Tecomat s webovým serverem. Druhá funkce tohoto nástroje je možnost je ho využít pro vizualizaci v prostředí Mosaic. Této funkce je možné využít i pro systémy bez webového serveru, pokud jsou programované dle normy IEC EN 61131-3. Pro

využití funkce webserveru v PLC řady Foxtrot a TC700 je potřeba paměťová karta pro uložení souborů webových stránek.

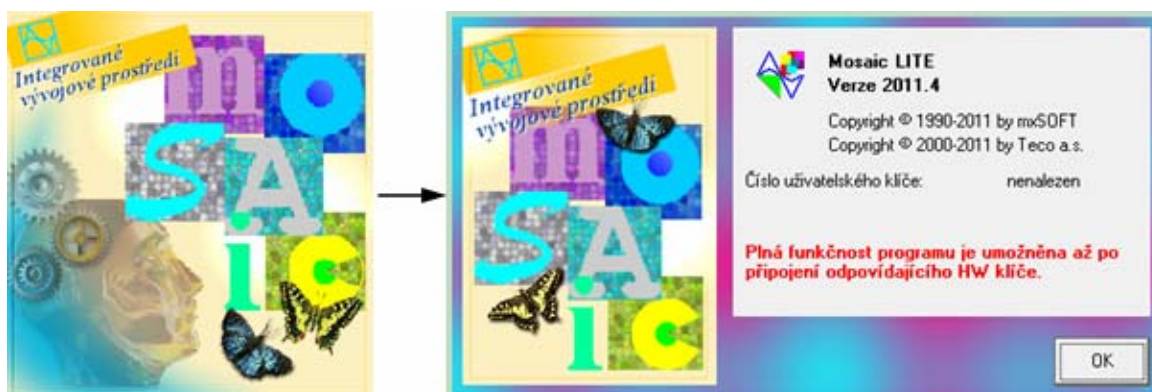
4.1 Začínáme v programu Mosaic

Projektem v Mosaic je myšlen program pro PLC, včetně všech souvisejících souborů. Programy pro řídicí systémy se skládají z jednotlivých souborů. Některé si vytváří sám programátor a jiné jsou tvořené automaticky.

Další důležitá věc je, že každý projekt pro PLC ve vývojovém prostředí Mosaic musí být součástí skupiny projektů. Skupina projektů obsahuje nejméně jeden, nebo více projektů které jsou částí celé sítě řídicího systému. Projekty ve skupině mezi sebou mohou mít komunikační vazby a vytváří tak celek. Každý projekt je tvořen samostatnou složkou, ve které najdeme všechny potřebné soubory a informace pro programování jednoho řídicího systému.

4.1.1 Spuštění vývojového prostředí Mosaic

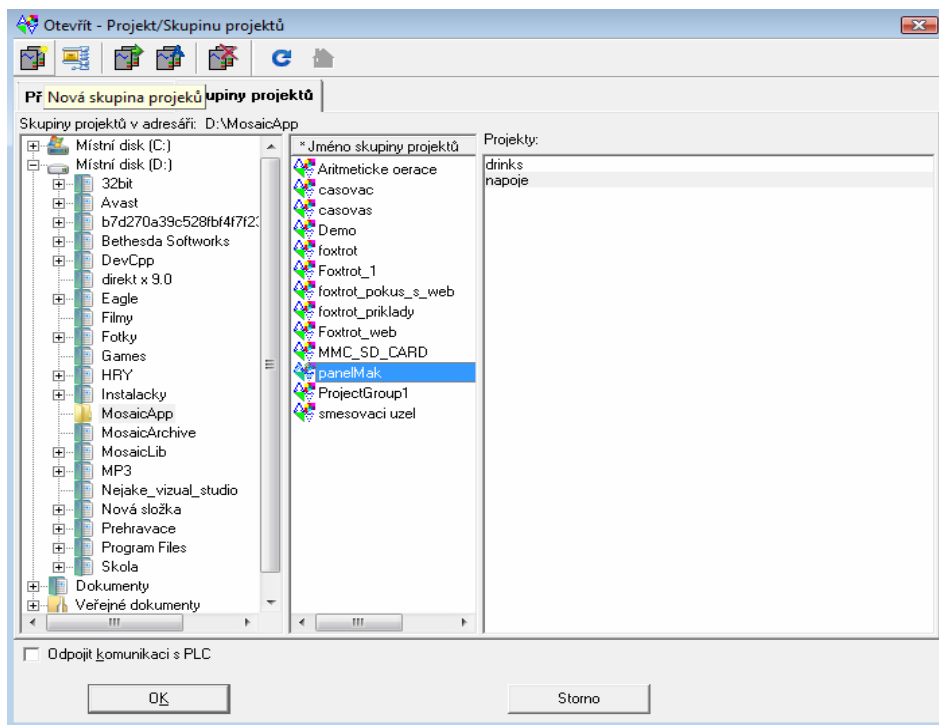
Zde si popíšeme vytvoření nového projektu první program pro náš systém (Tecomat Foxtrot). Po kliknutí na ikonu Mosaic naběhnou dvě úvodní obrazovky zobrazené na Obrázku č.17. Po otevření těchto oken se zobrazí dialog pro otevření, či založení nového projektu.



Obrázek 17 Spuštění vývojového prostředí Mosaic

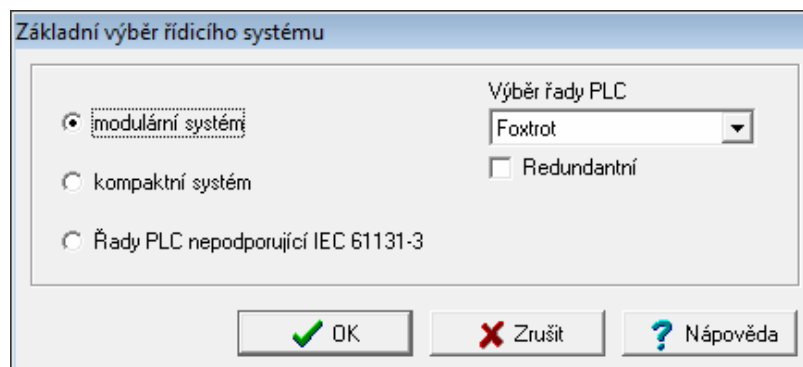
4.1.2 Založení nové skupiny projektů

Po stisku tlačítka „OK“ se nám otevře okno, kde nalezneme již existující skupiny projektů, a projekty, nebo si zde můžeme založit novou skupinu projektů. Dále následuje posloupnost oken, která nás povede k vytvoření nového projektu.



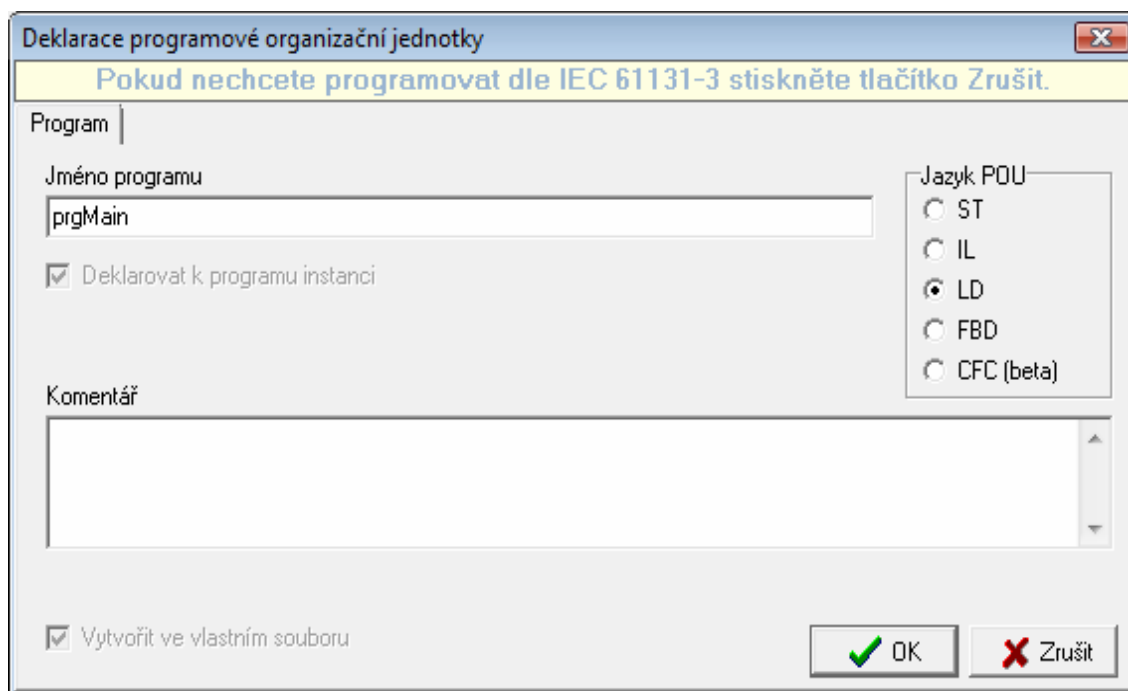
Obrázek 18 Založení nové skupiny projektů

Dále bude následovat okno *Základní výběr řídicího systému*. Toto okno slouží k určení typu PLC produkce Teco na němž program cílově poběží. Starší řady (NS950, TC400...), které nepodporují programování podle normy IEC EN 61131-3, umožní Mosaic programovat původním mnemokódem.

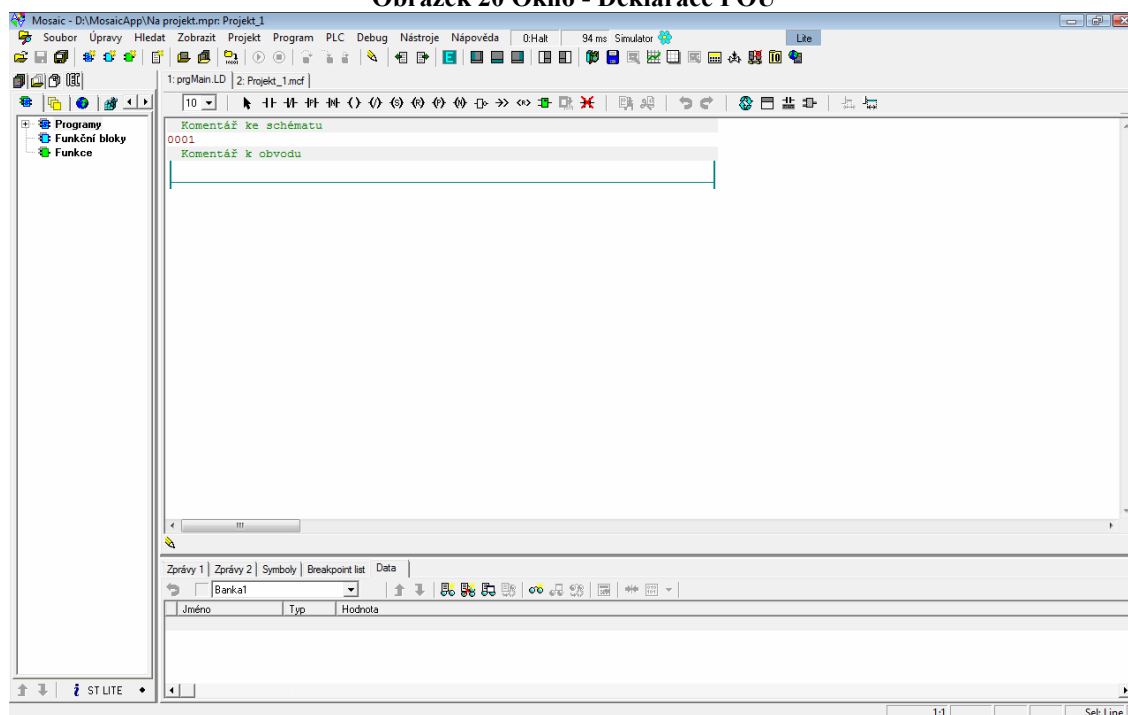


Obrázek 19 Okno - Základní výběr řídicího systému

Zde si vyberme možnost, jako je na obrázku *modulární systém* a výběr řady PLC *Foxtrot* a stiskneme OK. Dále nám vyběhne další okno *Deklarace programové organizační jednotky*. Zde zadáme jméno programu a je zde na výběr jazyků, kterými můžeme programovat. My si zde vybereme LD (Ladder diagram). Pak stiskneme OK, pak znova OK a už nám naběhne Mosaic pro samotné programování (viz. Obrázek 21).



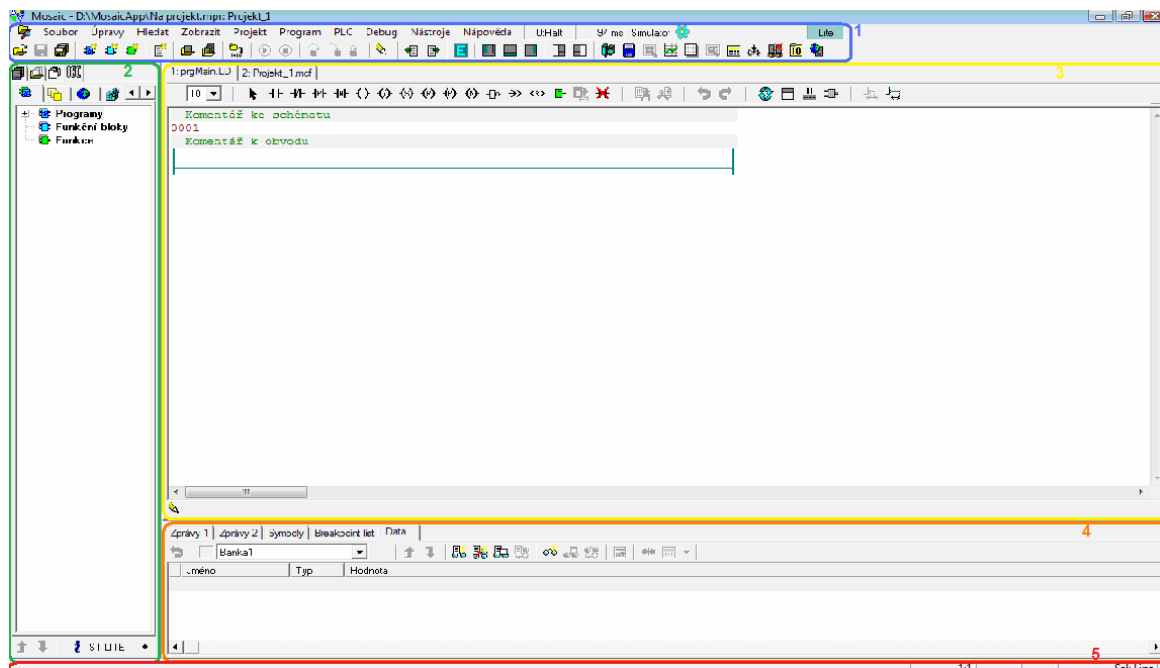
Obrázek 20 Okno - Deklarace POU



Obrázek 21 Mosaic

4.1.3 Základní popis prostředí Mosaic

Zde si popíšeme základní uspořádání hlavního okna programovacího prostředí Mosaic.

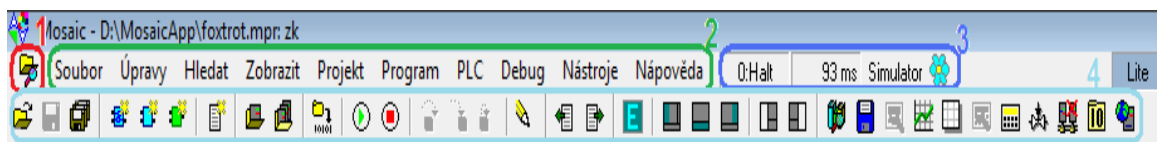


Obrázek 22 Hlavní okno programovacího prostředí

1. Hlavní nabídka, v ní najdeme textové menu a pod ním je nástrojová lišta s grafickými ikonami
2. V levé části je panel, kde obvykle najdeme okna:
 - Skupiny projektů
 - Soubory projektu
 - Seznam otevřených souborů
 - IEC manager
3. Ve střední části nalezneme panel, kde je okno, v němž provádíme samotné programování. V horní části jsou záložky se jmény otevřených souborů.
4. V dolní části obvykle nalezneme panel, kde jsou otevřena okna informačních nástrojů. Např.:
 - Zprávy1
 - Zprávy2
 - Seznam ladicích bodů (breakpoints)
 - Data
5. V nejnižším řádku hlavního okna je informační řádek, kde se zobrazují informační texty a v pravé části informace z aktivního editoru, jako je číslo řádku, sloupec a pracovní režimy editoru

















4.1.4 Hlavní menu prostředí Mosaic

V levém horním rohu hlavní nabídky se nalézá ikona manažera projektu (1), dále obsahuje hlavní menu (2), vlevo nahoře jsou informace o stavu PLC (3) dole se nalézá grafická nástrojová lišta(4).










Obrázek 23 Hlavní nabídka a nástrojová lišta s ikonami

Význam ikon v grafické nástrojové liště hlavního menu		
Ikona	Význam ikony	Klávesová zkratka
	Manažer projektu	Ctrl+Alt+F11
	Otevřít soubor	Ctrl+O
	Uložit	Ctrl+S
	Uložit všechny modifikované soubory	
	Nový program	
	Nový funkční blok	
	Nová funkce	
	Nový soubor	Ctrl+N
	Seznam projektů	Shift+Ctrl+F12
	Otevřít skupinu projektů	Ctrl+F11
	Přeložit	F9
	Run (spuštění programů)	Ctrl+F9
	Halt (zastavení provádění programu)	Ctrl+F2
	Režim editoru „edice“	Alt+F6
	Režim editoru „ladění“	Alt+F6
	Předchozí editor	Ctrl+F6
	Následující editor	F6


	Hlavní panel	Ctrl+Alt+Up
	Levý panel (zap/vyp)	Ctrl+Alt+Left
	Dolní panel (zap/vyp)	Ctrl+Alt+Down
	Pravý panel (zap/vyp)	Ctrl+Alt+Right
	Levý panel (snížit/zvýšit)	
	Pravý panel (snížit/zvýšit)	
	Průzkumník knihoven	
	Datalogger	
	GraphMaker	
	Mapa uživatelských registrů	
	PanelMaker	
	Simulátor panelu	
	PIDMaker	
	PLCnet Manažer	
	Nastavení vstupů/výstupů	
	WebMaker	

Tabulka 3 Význam ikon v grafické nástrojové liště hlavního menu

Informace o stavu PLC		
Stav		Význam
		PLC běží, výstupy PLC odblokovány, program v PLC je shodný s otevřeným projektem
		PLC stojí, výstupy PLC zablokovány, program v PLC je shodný s otevřeným projektem
		PLC stojí, výstupy PLC zablokovány, program v PLC je odlišný s otevřeným projektem
		PLC stojí, program je odlišný od aktuálního projektu
		Komunikace s PLC, nebo se simulátorem je odpojena
		Navazování komunikace s PLC
		Chyba během komunikace s PLC

Tabulka 4 Informace o stavu PLC

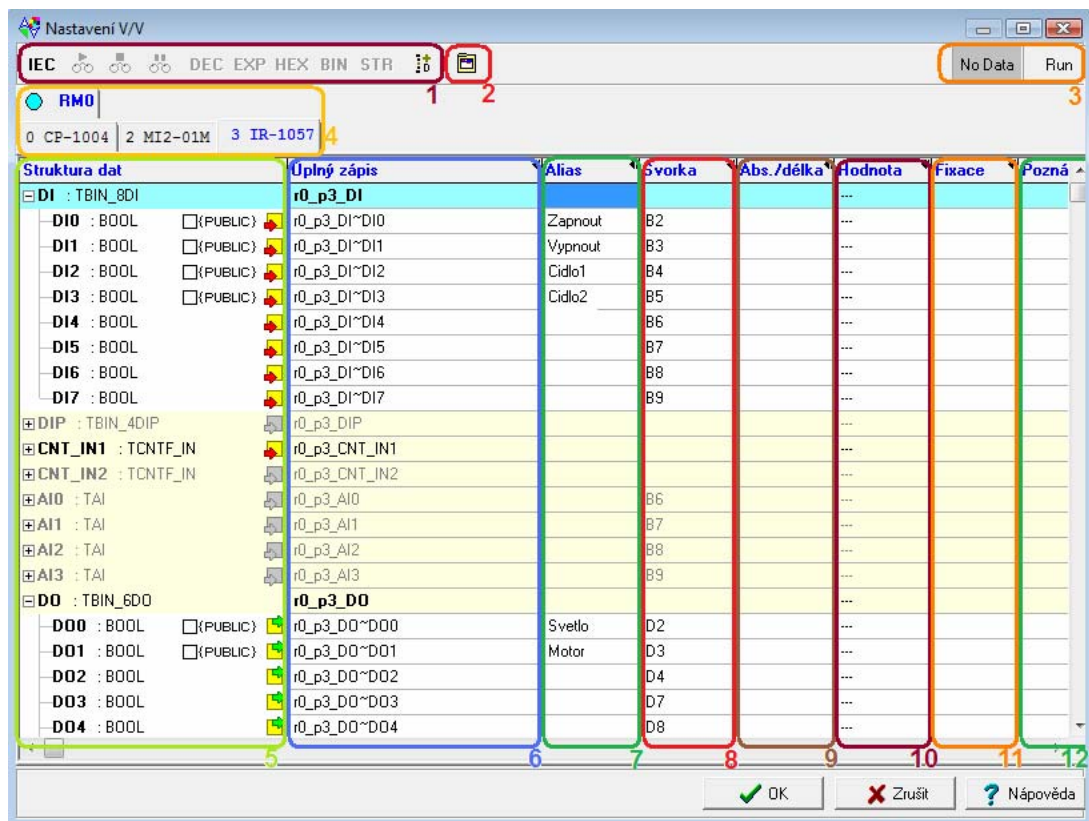
4.1.5 Nastavení vstupů a výstupů

Klikneme-li na ikonu , která se nalézá v horní části v grafické nástrojové liště, otevře se nástroj pro nastavení a správu V/V. Otevře se nové okno v modálním režimu (tzn. okno se musí po ukončení práce zavřít). Tento nástroj má 2 základní funkce:

Zobrazuje datovou strukturu periferních modulů a dovoluje přiřadit k jednotlivým tzv. aliasy (Každý vstup si může programátor pojmenovat vlastní názvem), se kterými bude programátor dále pracovat.

Druhá funkce je , že je-li PLC v režimu Run, tak zobrazuje aktuální hodnoty všech V/V proměnných. V případě potřeby umožňuje během ladění fixovat jejich hodnoty do zvoleného stavu

Chce-li programátor používat daný V/V, tak si do Alias napíše jméno a pod tímto jménem s V/V dále pracuje. S těmito Alias (jmény) lze pracovat až po překladu (tzn. že po pojmenování všech potřebných Aliasů zmáčkneme OK a v hlavním okně pro programování zmáčkneme tlačítko Přeložit (F9)).





Obrázek 24 Nástroj nastavení V/V

1. Ovládací lišta
2. Mapa osazení V/V
3. Stavové informace
4. Záložky vyjadřující strukturu sestavy ve více rámech a jejich osazení. Slouží pro vybírání periferního modulu podle jeho pozice v konkrétním rámu.
5. Struktura dat – stromová struktura dat dostupných na zvoleném modulu. Jedná se nejenom o vstupy a výstupy, ale i o stavové informace, řídicí slova, údaje o rozsazích apod.
6. Úplný zápis – automaticky přidělené systémové jméno proměnné ve struktuře: rám_pozice_vstup/výstup
7. Alias – Pojmenování proměnné programátorem přiřazené konkrétnímu V/V vlastním jménem. Změny v Alias jsou akceptovány až po překladu programu zapsání do PLC.
8. Svorka – označení svorky na konektoru modulu
9. Abs./délka – absolutní adresa proměnné/délka proměnné v bytech
10. Hodnota – aktuální hodnota V/V připojeného, nebo simulovaného PLC
11. Fixace – zafixování hodnoty proměnné během odlaďování algoritmů. „ze zde nastavit:
 - Výstupy – nastavení hodnot, které nejsou ovlivněny skutečným stavem na vstupu modulu
 - Výstupy – nastavení hodnot, které nejsou ovlivněny programem
12. Poznámky – zde si může zapsat libovolné poznámky o V/V

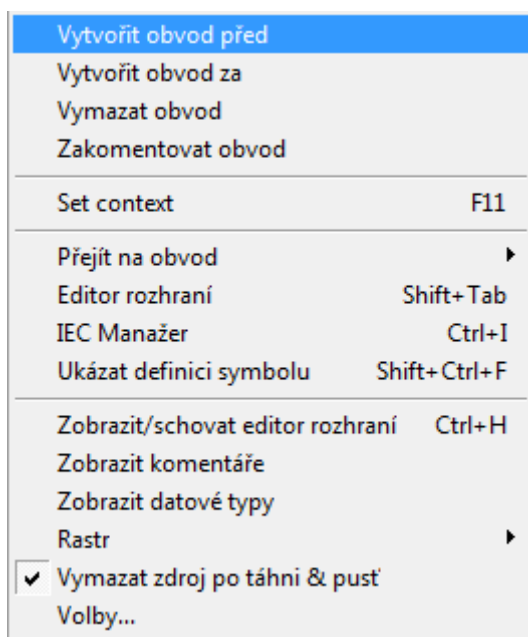
4.2 Programování v Mosaic v jazyce reléových schémat (LD)

Zde bude vysvětleno programování v jazyce reléových schémat v integrovaném vývojovém prostředí Mosaic.

Prázdný obvod je tvořen počáteční a konečnou sběrnicí a mezi nimi je základní příčka obvodu, na kterou se umísťují jednotlivé prvky obvodu. Prvky jsou vkládány v pevném rastru o minimální šířce 12 znaků. Při použití většího počtu rastrů se prostor mezi sběrnicemi rozšiřuje směrem doprava. Rozteč si lze měnit tlačítky Plus/Mínus, nebo ikonami  pro zmenšení rastru, nebo  pro zvětšení rastru.

Při vytváření samotného programu je pro přehlednost vhodné, netvořit zbytečně široké obvody a raději si vytvořit tzv. pomocnou vnitřní cívku (vnitřní proměnnou) a pokračovat níže v dalším obvodu.

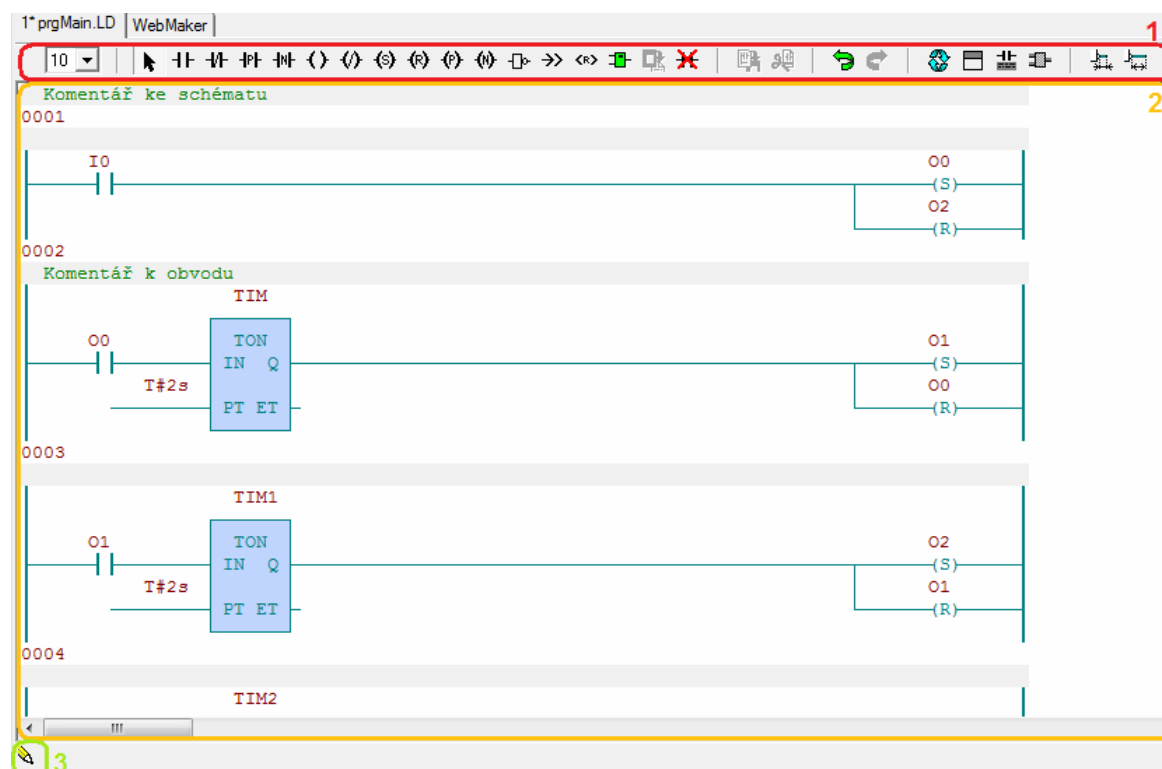
Nové prázdné obvody se přidávají kliknutím pravého tlačítka myši na ploše editoru, kde klikneme na: *Vytvořit obvod před*, nebo *Vytvořit obvod za*.



Obrázek 25 Vytvoření nového obvodu

Jednotlivé obvody jsou číslovány (0001 – 9999). Dvojklikem na číslo obvodu se otevře okno Návěští. Toto okno slouží pro tvorbu, nebo k editaci jména návěští, které dále slouží jako bod pro příkazy skoků. Komentář obvodu vyvoláme dvojklikem na šedý řádek, který se vždy nalézá pod číslem obvodu.


















4.3 Ovládací prvky v Editoru LD



Obrázek 26 Ovládací prvky v editoru LD

1. Ovládací prvky
2. Pracovní plocha
3. Ikona signalizující „Editaci“, nebo „Ladění“

Ovládací prvky Editoru LD	
Ikona	Význam ikony
	Velikost fontu
	Kurzor (konec editace)
	Kontakt (položít na značku: sériový , paralelní)
	Negovaný kontakt (položít na značku: sériový , paralelní)
	Kontakt citlivý na náběžnou hranu (položít na značku: sériový , paralelní)
	Kontakt citlivý na sestupnou hranu (položít na značku: sériový , paralelní)
	Cívka (položít na značku:)
	Negovaná cívka (položít na značku:)

	Cívka SET (položít na značku: <input type="checkbox"/>)
	Cívka RESET (položít na značku: <input type="checkbox"/>)
	Cívka citlivá na náběžnou hranu (položít na značku: <input type="checkbox"/>)
	Cívka citlivá na sestupnou hranu (položít na značku: <input type="checkbox"/>)
	NOT (negace) (položít na značku: <input type="checkbox"/> , negace kontaktu, cívky, signálu boxu)
	Podmíněný skok na návěští (položít na značku: <input type="checkbox"/>)
	Podmíněný návrat z POU (položít na značku: <input type="checkbox"/>)
	Vložit box POU (časovače, čítače, matematické a logické funkce apod.) (položít na značku: <input type="checkbox"/>)
	Zrušit objekt
	Zpět
	Vrátit
	Obnovit obrazovku
	Editor rozhraní
	Zobrazit komentáře
	Zobrazit datové typy
	Zmenšení rastru
	Zvětšení rastru

Tabulka 5 Ovládací prvky Editoru LD

4.4 Úlohy k řešení

Úloha 1: Při sepnutí tlačítka se žárovka rozsvítí, je-li tlačítko rozepnuté, žárovka nesvítí.

Úloha 2: Žárovka se rozsvítí, bude-li sepnuto tlačítko 1, nebo tlačítko 2, nebo budou-li sepnuty obě tlačítka najednou.

Úloha 3: Chceme-li roztočit motor, musí být zároveň sepnuty tlačítko 1 a tlačítko 2. Bude-li sepnuto jen jedno, motor se neroztočí.

Úloha 4: Žárovka se rozsvítí, bude-li sepnuto tlačítko 1, nebo tlačítko 2, budou-li sepnuty, nebo rozepnuty obě tlačítka najednou, žárovka nesvítí.

Úloha 5: Motor se roztočí, jen když bude sepnuto tlačítko 1 a tlačítko 2, nebo když budou obě tlačítka rozepnutá.

Úloha 6: Motor se roztočí jen za podmínky, že budou sepnutá 2 libovolná tlačítka ze 3.

Úloha 7: Žárovka se rozsvítí až po 5-ti stisknutí tlačítka.

Úloha 8: Při stisku vypínače se rozsvítí světlo a zároveň se rozběhne i větrák. Při vypnutí vypínače zhasne světlo, ale větrák poběží ještě dalších 15 sekund.

Úloha 9: Žárovka se rozsvítí až po 5-ti sekundách po sepnutí tlačítka.

Úloha 10: Sepnu-li tlačítko, světlo se rozsvítí, po rozepnutí tlačítka bude světlo svítit ještě 10 sekund.

Úloha 11: Vytvořte simulaci semaforu. Při sepnutí tlačítka 1 se na 5 sekund rozsvítí červené světlo, jakmile červené světlo zhasne, rozsvítí se na 5 sekund světlo oranžové. Po zhasnutí oranžového světla se rozsvítí světlo zelené.

Úloha 12: Vytvořte simulaci semaforu s tím, že se bude cyklus stále opakovat. Při sepnutí tlačítka 1 se na 5 sekund rozsvítí červené světlo, jakmile červené světlo zhasne, rozsvítí se na 5 sekund světlo oranžové. Po zhasnutí oranžového světla se rozsvítí světlo zelené. Při zhasnutí zeleného světla se na 5 sekund rozsvítí červené světlo...

5 ZÁVĚR

Cílem této práce bylo vytvořit co nejnázornější učební pomůcku k předmětu PGS na bázi k programovatelným automatům Tecomat Foxtrot a jejich programování v integrovaném vývojovém prostředí Mosaic v jazyce reléových schémat. Tyto cíle se v rámci absolventské práce podařilo splnit a naplnit tak požadavky zadání. Sám jsem si to vyzkoušel na mé přítelkyni, která nemá žádné technické vzdělání a podle této práce téměř bez problémů zadané úlohy zvládla.

Tvorba absolventské práce mi přinesla mnoho prohloubení stávajících znalostí a dovedností ve SW (MS Word, Adobe Photoshop, Mosaic), které byly využívány pro tvorbu této práce. Tato práce mi ukázala jak veliký význam mají dnes programovatelné automaty v řízení technologií v průmyslu, kde mají dnes nezastupitelné místo.

Tvorba této práce pro mě měla veliký přínos a doufám, že získané znalosti využiji i v budoucnu, jelikož bych si chtěl najít zaměstnání v tomto, nebo podobném oboru.

6 SEZNAM LITERATURY

- [1] KOCHANÍČEK, Ludvík. Historický vývoj programovatelných automatů.
In: <http://coptel.coptkm.cz>. [online]. Dostupné z:
<http://coptel.coptkm.cz/index.php?action=2&doc=2547&docGroup=179&cmd=0&instance=1> COPTEL
- [2] VANČURA. T. et.al. Výukové texty k mechatronickým sestavám. Lanškroun. TG TISK s. r. o., červen 2007.
- [3] MARTINÁSKOVÁ. M. Programovací jazyky pro PLC. In: <http://automatizace.cz>. [online]. Dostupné z: <http://www.automatizace.cz/article.php?a=142>
- [4] PROGRAMOVATELNÉ AUTOMATY TECOMAT FOXTROT.
In: <http://tecomat.com>. [online]. 18. vydání. leden 2012. Dostupné z:
http://www.tecomat.com/wpimages/other/DOCS/cze/TXV00410_01_General_Foxtrot.pdf
- [5] NEMEŠKAL. V. Mosaic – integrované prostředí pro PLC Tecomat.
In: <http://automatizace.cz>. [online]. Dostupné z:
<http://www.automatizace.cz/article.php?a=1695>

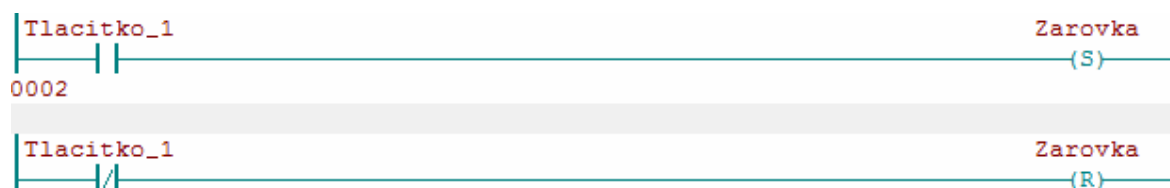
7 OBSAH CD

- Absolventská práce ve formátu PDF
- Mosaic

8 PŘÍLOHY

I. Řešení úloh

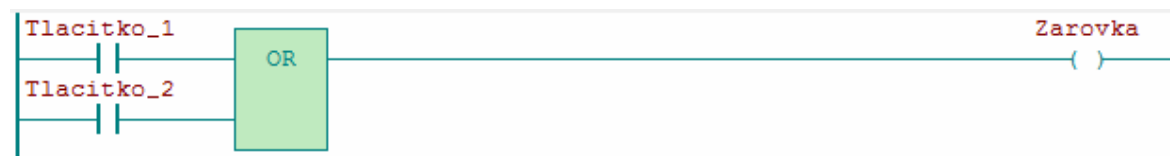
Úloha 1: Při sepnutí tlačítka se žárovka rozsvítí, je-li tlačítko rozepnuté, žárovka nesvítí.



nebo a jednodušeji



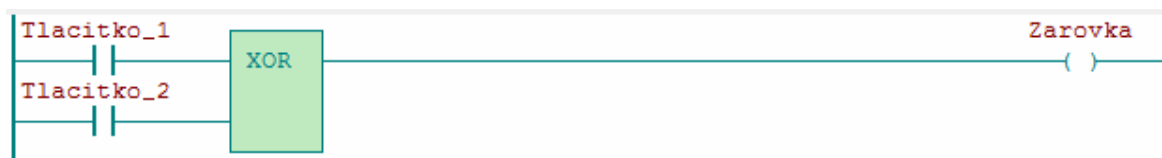
Úloha 2: Žárovka se rozsvítí, bude-li sepnuto tlačítko 1, nebo tlačítko 2, nebo budou-li sepnuty obě tlačítka najednou.



Úloha 3: Chceme-li roztočit motor, musí být zároveň sepnuty tlačítko 1 a tlačítko 2. Bude-li sepnuto jen jedno, motor se neroztočí.



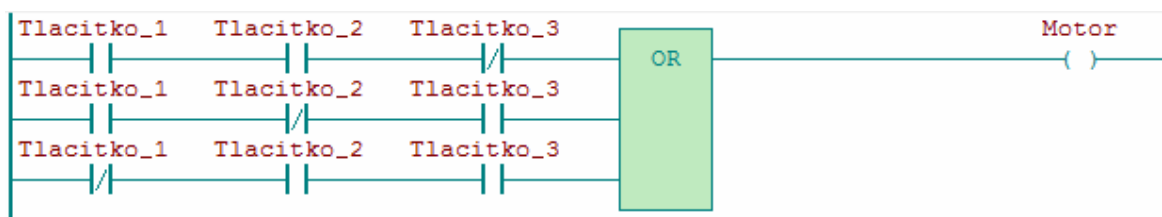
Úloha 4: Žárovka se rozsvítí, bude-li sepnuto tlačítko 1, nebo tlačítko 2, budou-li sepnuty, nebo rozepnuty obě tlačítka najednou, žárovka nesvítí.



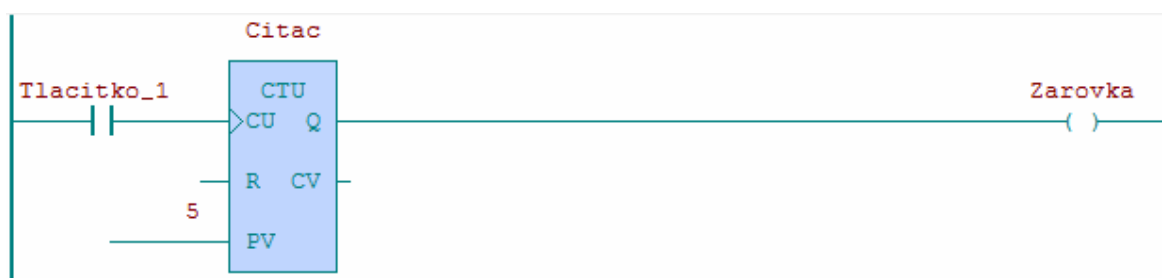
Úloha 5: Motor se roztočí, jen když bude sepnuto tlačítko 1 a tlačítko 2, nebo když budou obě tlačítka rozepnutá.



Úloha 6: Motor se roztočí jen za podmínky, že budou sepnutá 2 libovolná tlačítka ze 3.

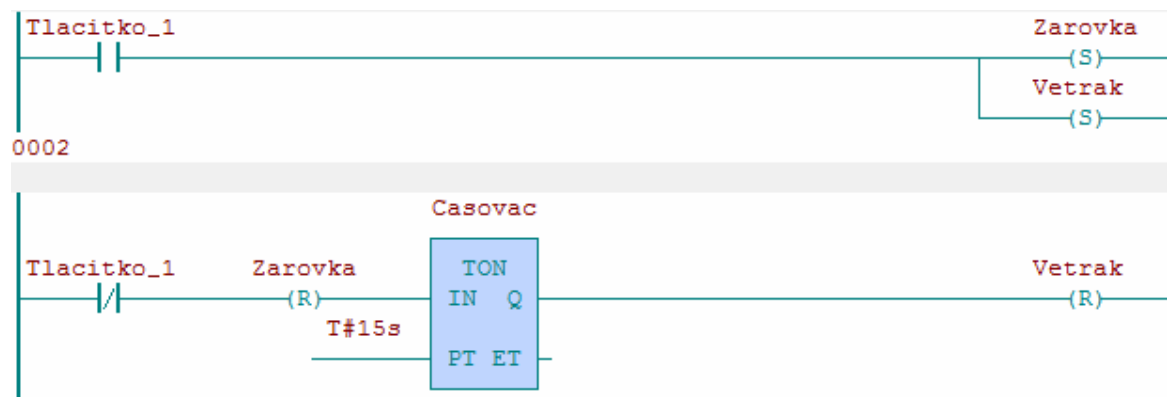


Úloha 7: Žárovka bude svítit až po 5-ti stisknutí tlačítka.



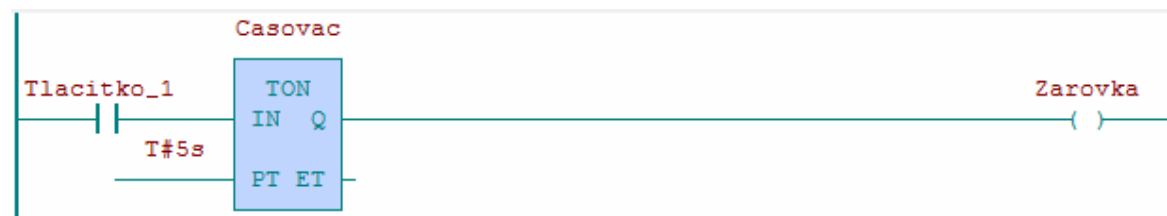
Po rozkliknutí kontaktu PV se do Jména napíše počet stisků (v tomto případě 5).

Úloha 8: Při stisku vypínače se rozsvítí světlo a zároveň se rozběhne i větrák. Při vypnutí vypínače zhasne světlo, ale větrák poběží ještě dalších 15 sekund.



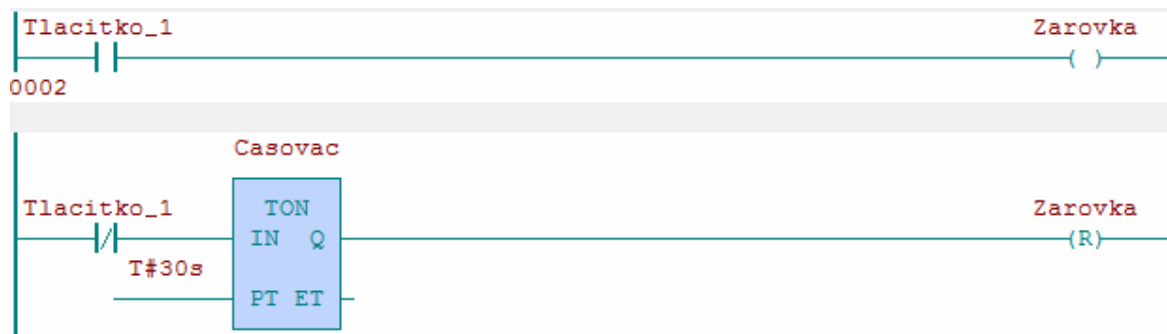
Po rozkliknutí kontaktu PT se do Jména napíše: T#CHTĚNÝ ČASs (v tomto případě: T#15s).

Úloha 9: Žárovka se rozsvítí až po 5-ti sekundách po sepnutí tlačítka.



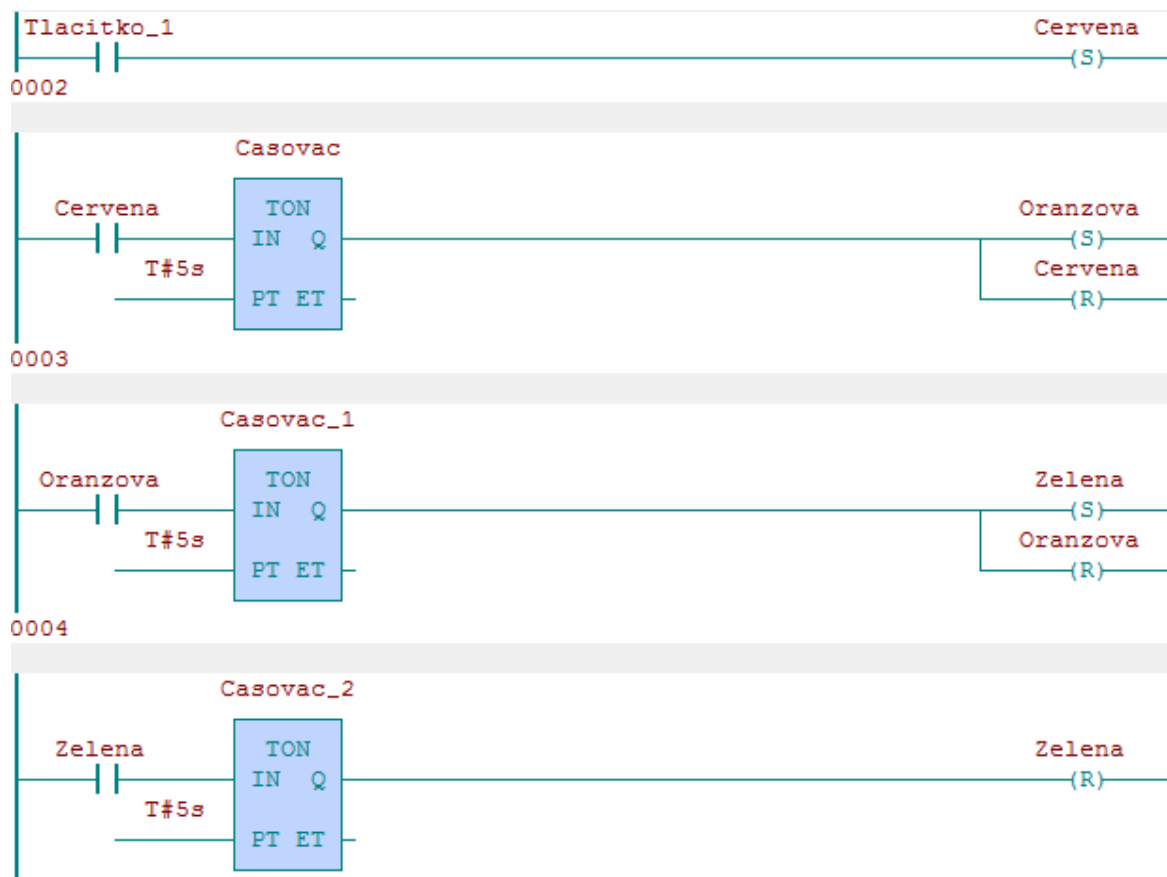
Po rozkliknutí kontaktu PT se do Jména napíše: T#CHTĚNÝ ČASs (v tomto případě: T#5s).

Úloha 10: Sepnu-li tlačítko, světlo se rozsvítí, po rozeptnutí tlačítka bude světlo svítit ještě 30 sekund.

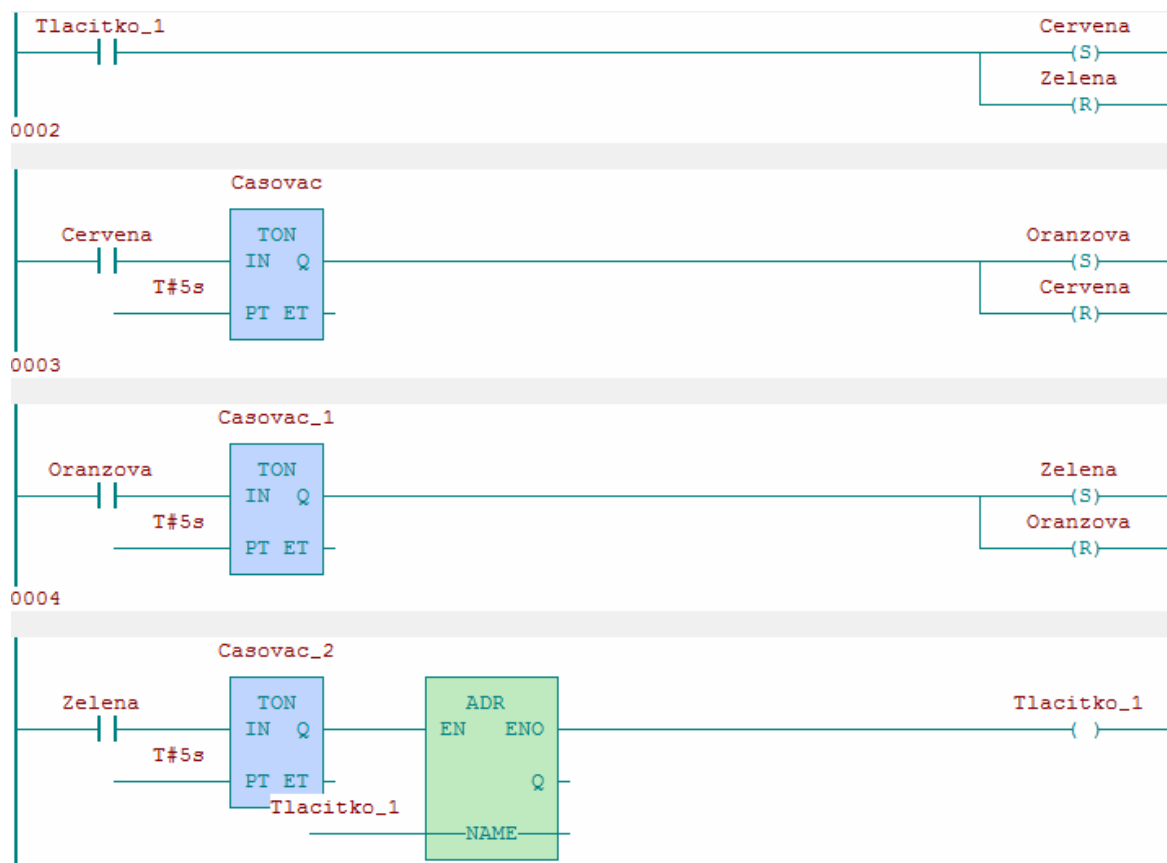


Po rozkliknutí kontaktu PT se do Jména napíše: T#CHTĚNÝ ČASs (v tomto případě: T#30s).

Úloha 11: Vytvořte simulaci semaforu. Při sepnutí tlačítka se na 5 sekund rozsvítí červené světlo, jakmile červené světlo zhasne, rozsvítí se na 5 sekund světlo oranžové. Po zhasnutí oranžového světla se rozsvítí světlo zelené.



Úloha 12: Vytvořte simulaci semaforu s tím, že se bude cyklus stále opakovat. Při sepnutí tlačítka 1 se na 5 sekund rozsvítí červené světlo, jakmile červené světlo zhasne, rozsvítí se na 5 sekund světlo oranžové. Po zhasnutí oranžového světla se rozsvítí světlo zelené. Při zhasnutí zeleného světla se na 5 sekund rozsvítí červené světlo...

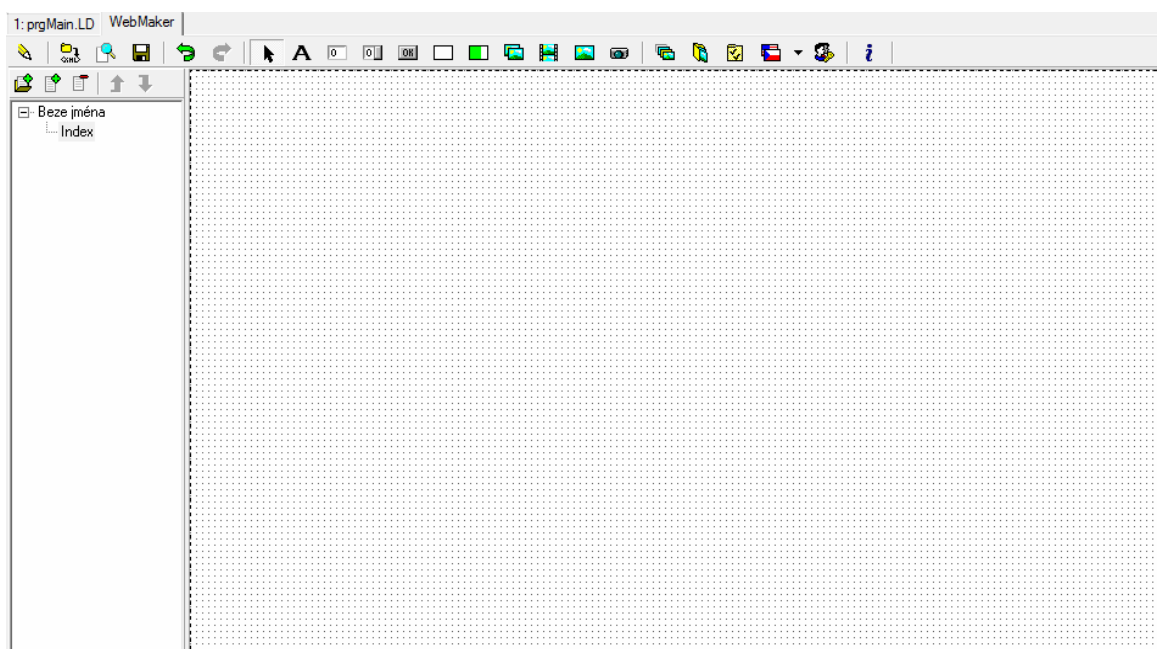


II. WebMaker


Tento nástroj je určený primárně pro tvorbu webových stránek pro systémy Tecomat s webovým serverem. Druhá funkce tohoto nástroje je možnost je ho využít pro vizualizaci v prostředí Mosaic. Této funkce je možné využít i pro systémy bez webového serveru, pokud jsou programované dle normy IEC EN 61131-3.

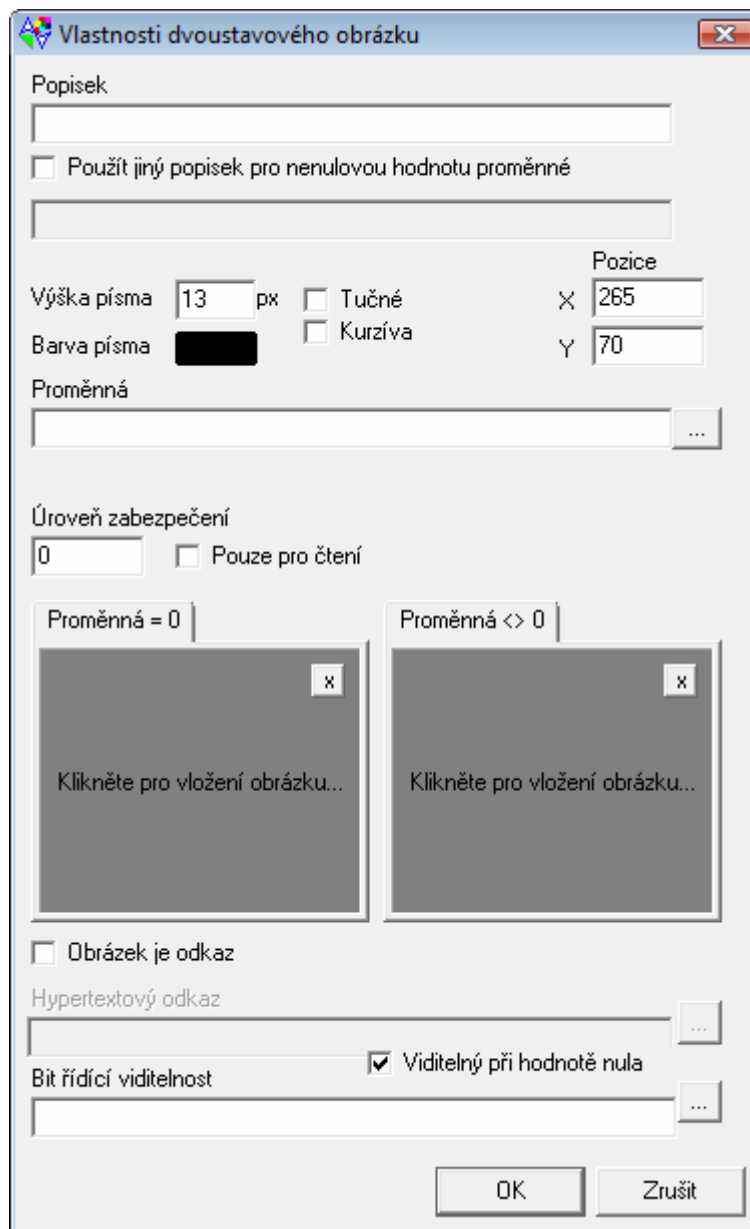
Zde bude návod, jak WebMaker využít jako vizualizačním nástroj v prostředí Mosaic.

V hlavní nabídce klikneme na ikonu  a spustí se nástroj WebMaker.



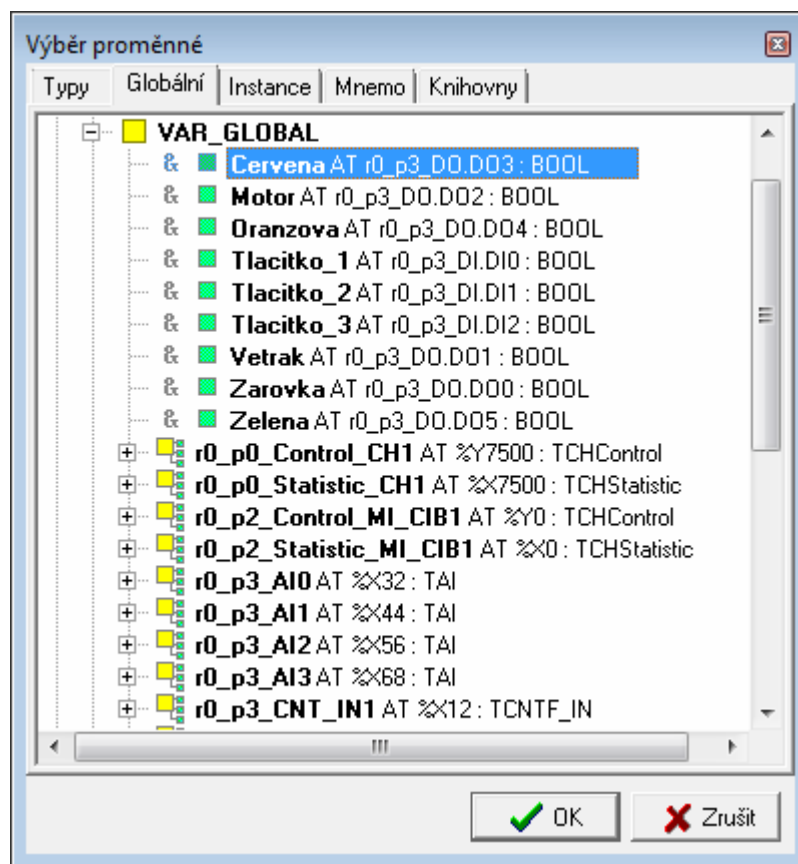
WebMaker

Klikneme na ikonu: Dvoustavový obrázek  a vyběhne následující okno.

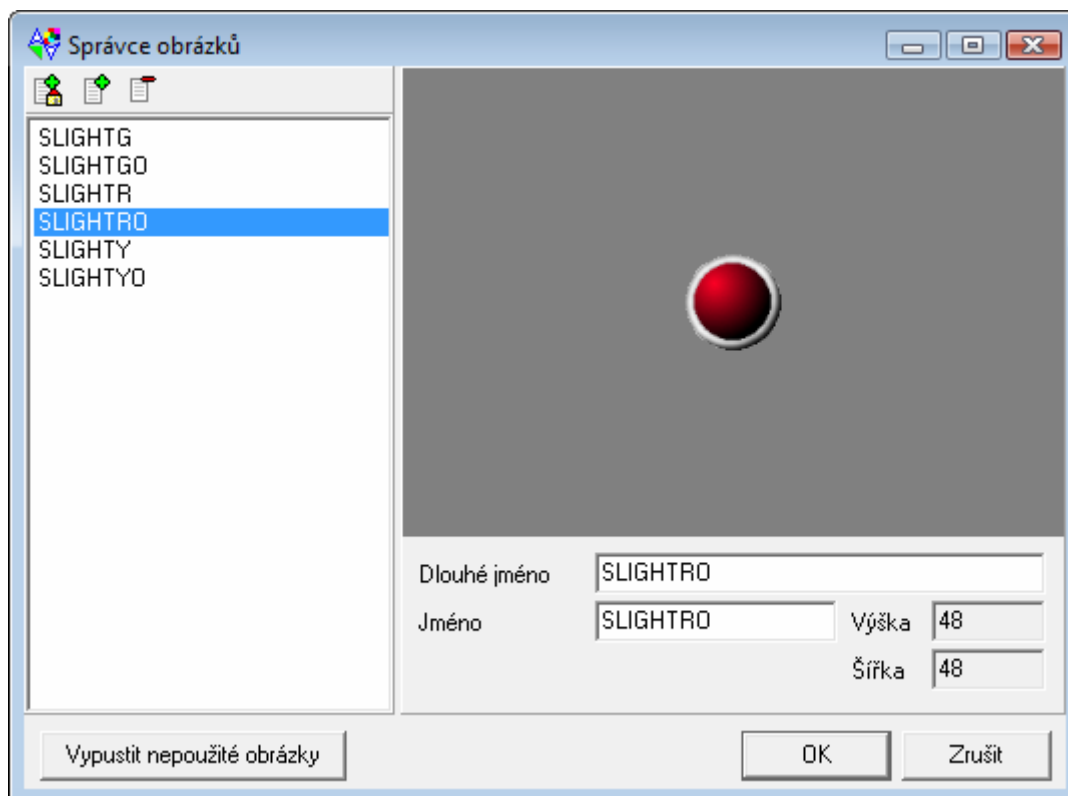


Nastavení dvoustavového obrázku

Do okénka „Popisek“ si můžeme napsat Jméno znázorňované proměnné. Po rozkliknutí políčka „Proměnná“ si vybereme ze vstupů, nebo výstupů chtěnou znázorňovanou proměnnou. Další krok je vložení obrázků, které si musím nejdříve načíst, a budou danou proměnnou znázorňovat. To se provede následovně. Klikneme na obrázek, nad kterým je „Proměnná = 0“, to znamená, že tento obrázek bude simulovat danou proměnnou ve stavu logické 0. pak provedeme to samé s obrázkem, nad kterým je „Proměnná <>0“. Tento obrázek bude danou proměnnou simulovat ve stavu logické 1. Nakonec zaškrtneme okénko „Viditelný při hodnotě nula“. To aby byl obrázek vidět, když je znázorňovaná proměnná ve stavu logické 0.



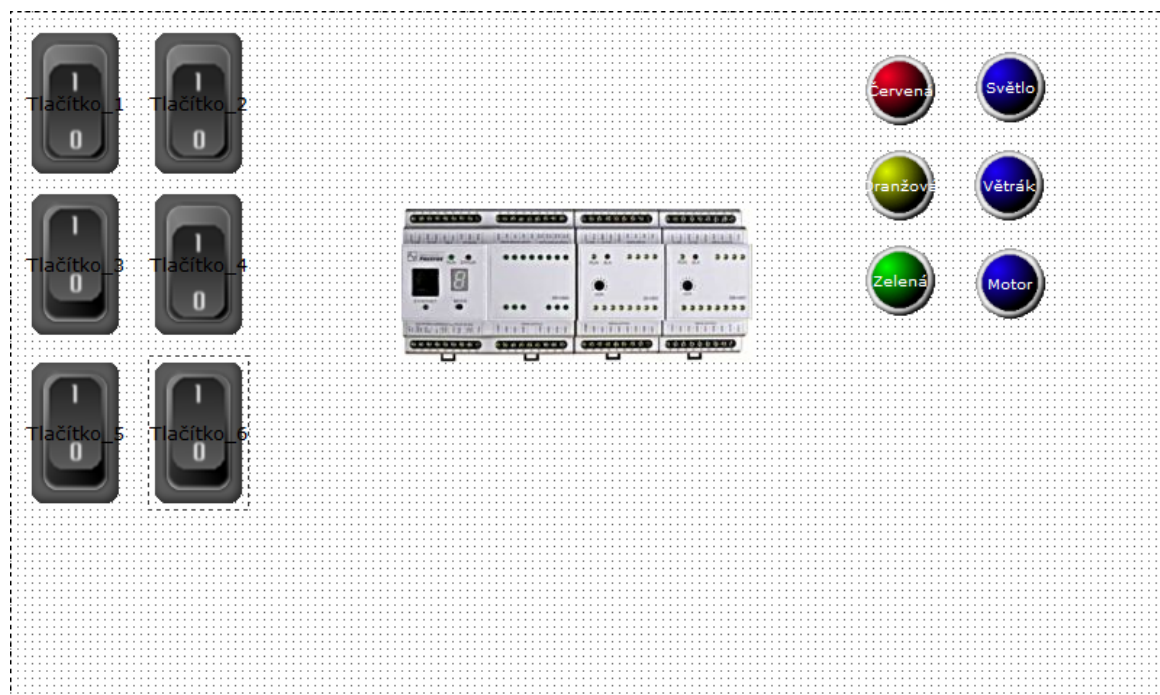
Výběr znázorňované proměnné



Výběr obrázku simulované proměnné



Nastavení vlastností dvoustavového obrázku



Příklad vizualizace pomocí nástroje WebMaker