

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2022

Danny Sonbol



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV VÝKONOVÉ ELEKTROTECHNIKY A ELEKTRONIKY

DEPARTMENT OF POWER ELECTRICAL AND ELECTRONIC ENGINEERING

PROGRAMOVATELNÉ LOGICKÉ AUTOMATY SCHNEIDER A JEJICH VYUŽITÍ VE VÝUCE

PROGRAMMABLE LOGIC CONTROLLERS AND THEIR USE IN TEACHING PROCESS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Danny Sonbol

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Dalibor Červinka, Ph.D.

BRNO 2022

Bakalářská práce

bakalářský studijní program **Silnoproudá elektrotechnika a elektroenergetika**

Ústav výkonové elektrotechniky a elektroniky

Student: Danny Sonbol

ID: 203344

Ročník: 3

Akademický rok: 2021/22

NÁZEV TÉMATU:

Programovatelné logické automaty Schneider a jejich využití ve výuce

POKyny PRO VYPRACOVÁNÍ:

1. Popište jednotlivé programovací jazyky programovatelných automatů Schneider a zhodnoťte jejich použitelnost ve výuce.
2. Navrhněte a realizujte jednoduché laboratorní úlohy na procvičení jednotlivých programovacích technik.
3. Vytvořte příslušné laboratorní návody.

DOPORUČENÁ LITERATURA:

- [1] Dokumentace k PLC Schneider electric.
- [2] Vorel P., Patocka M., Prumyslová elektronika, Vydání 1., skriptum FEKT VUT Brno, 2007
- [3] SKALICKÝ, J. Navrhování elektrických pohonů. Navrhování elektrických pohonů. Brno: VUT FEKT, 2002.

Termín zadání: 7.2.2022

Termín odevzdání: 31.5.2022

Vedoucí práce: Ing. Dalibor Červinka, Ph.D.

prof. Ing. Petr Toman, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRACT

This thesis deals with a programmable logic controller (PLC) that is used in automation, machine control or production lines in factories. The studied PLC type is Modicon M241. Its principle, construction, and programming languages are further described. In addition, a control program was developed in Machine Expert programming software and laboratory manual.

KEY WORDS:

PLC; Machine Expert; Shneider Electric; Modicon;

ABSTRAKT

Tato práce se zabývá programovatelným logickým automatem (PLC), který slouží pro automatizaci, řízení strojů nebo výrobních linek v továrnách. Cílem bylo prostudovat dokumentaci k PLC typu Modicon M241, co se týče principu činnosti, konstrukce, programovatelných jazyků a programovatelného softwaru Machine Expert. A vytvořit experimentální verzi ovládacího programu a laboratorní návod.

KLÍČOVÁ SLOVA:

PLC; Machine Expert; Shneider Electric; Modicon;

BIBLIOGRAFICKÁ CITACE

SONBOL, Danny. *Programovatelné logické automaty Schneider a jejich využití ve výuce* [online]. Brno, 2022 [cit. 2022-05-17]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/142713>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav výkonové elektrotechniky a elektroniky. Vedoucí práce Dalibor Červinka.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	Danny Sonbol
VUT ID studenta:	203344
Typ práce:	Bakalářská práce
Akademický rok:	2021/22
Téma závěrečné práce:	Programovatelné logické automaty Schneider a jejich využití ve výuce

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: 31. května 2022

podpis autora

Poděkování

Rád bych poděkoval svému vedoucímu bakalářské práce Ing. Daliborovi Červinkovi za odbornou a pedagogickou pomoc a rady při zpracování mé bakalářské práce.

V Brně dne: 31. května 2022

podpis autora

Obsah

Seznam Obrázku	8
Seznam tabulek	9
Seznam symbolů a zkratk	10
Úvod	11
1 Typy PLC	12
1.1 Mikro PLC	12
1.2 Kompaktní PLC	13
1.3 Modulární PLC	13
2 Vnitřní konstrukce PLC	14
3 Princip činnosti	16
3.1 Skenovací cyklus	16
4 Programovací jazyky	17
4.1 Instrukční list IL (Instruction List)	17
4.2 Strukturovaný text ST (Structured Text)	18
4.3 Reléová schémata LD (Ladder Diagram)	18
4.4 Funkční bloky FBD (Function Block Diagram)	19
4.5 Sekvenční programování SFC (Sequential Function Chart)	20
5 PLC Modicon M241 TM241CEC24T	21
6 Programovací prostředí Machine Expert	23

6.1	Vytváření nového projektu	24
6.2	Nahrání programu do PLC	26
7	Výuková rozšiřující deska pro PLC	28
7.1	Struktura rozšiřující desky	28
7.2	Realizace desky	30
8	Využití desek ve výuce	31
8.1	Realizace demonstrační úlohy Ovládání pohonu armatury	31
8.2	Řídicí program	31
8.2.1	Řídicí program realizovaný jazykem LD	32
8.2.2	Řídicí program realizovaný jazykem ST	33
9	Laboratorní návod	
	Ovládání pohonu armatury v JE Temelín	34
10	Závěr	37

Seznam obrázků

1-1	Příklad mikro PLC [2]	12
1-2	Příklad kompaktního PLC s moduly	13
1-3	Příklad modulárního PLC s moduly	13
2-1	Schéma PLC systému	15
2-2	Bloková struktura PLC systému [6]	15
3-1	Schéma skenovacího cyklu PLC systému [10]	16
4-1	Rozdělení PLC Programovacích Jazyků	17
4-2	Příklad programovacího jazyka IL [15]	17
4-3	Příklad programovacího jazyka ST [17]	18
4-4	Příklad programovacího jazyka LD [19]	19
4-5	Příklad programovacího jazyka FBD [14]	19
4-6	Příklad programovacího jazyka SFC [21]	20
5-1	Popis části PLC Modicon TM241CEC24T [22]	22
6-1	Centrální obrazovka softwaru Machine Expert	23
6-2	Postup při vytváření nového projektu	24
6-3	Ukázka vytváření POU	24
6-4	Ukázka propojování Vstupů	25
6-5	Ukázka Task Configuration	25
6-6	Ukázka varovného okénka	26
6-7	Ukázka nahrávání programu	26
6-8	Simulation v softwaru Machine Expert	27
7-1	Zapojení silové části pohonu	29
7-2	Rozložení jednotlivých prvků na desce	29

7-3	Hotové sety připravené na nasazení ve výuce	30
8-1	Mechanismus s DC motorem a koncovými spínači	31
8-2	Řídící program (programovací jazyk LD)	32
8-3	Řídící program (programovací jazyk ST)	33
9-1	Laboratorní pracoviště	34
9-2	Schéma zapojení laboratorní úlohy	35

Seznam tabulek

5-1	Hlavní parametry PLC Modicon TM241CEC24T [22]	21
5-2	Popis části PLC Modicon TM241CEC24T [22]	22
7-2	Seznam součástí na Obr. 7-2	30
8-2	Význam jednotlivých symbolů použitých v programu	32
8-3	Význam jednotlivých příkazů použitých v programu	33

Seznam symbolů a zkratek

Zkratka	Název
CPU	Centrální procesorová jednotka (central processing unit)
DC	Stejnoseměrný proud (Direct Current)
DI	Digitální vstupy (Digital inputs)
DQ	Digitální výstupy (Digital outputs)
FBD	Funkční bloky (Function Block Diagram)
HMI	Rozhraní mezi člověkem a strojem (Human machine interface)
IL	Instrukční list (Instruction List)
LD	Réleová schémata (Ladder Diagram)
PLC	Programovatelný automatizovaný kontrolér (Programmable Logic Controller)
SFC	Sekvenční programování (Sequential Function Chart)
ST	Strukturovaný text (Structured Text)
USB	Univerzální sériová sběrnice (Universal Serial Bus)
MB	megabajt (megabyte)

Úvod

Programovatelný logický automat je průmyslový počítač, který sleduje vstupy a výstupy, které posuzuje na základě vytvořeného programu. Snižuje namáhání obsluhy při rozhodování a tím dosáhneme vyšší efektivnosti a také snižuje náklady na různé typy fyzických zařízení (relé, časovače, tlačítka atd.), které lze nahradit interními funkcemi. Využívají se zejména v průmyslových systémech (elektrické stroje, osvětlení, výrobní procesy a stavební systémy), ale lze je využít i v civilních aplikacích (domácí spotřebiče, různé typy zařízení, řízení dopravních signálů a výtahů).

Hlavním cílem bakalářské práce je podrobné prostudování dokumentace pro PLC Modicon M241 TM241CEC24T a jeho programovací software Machine Expert, navrhnout a realizovat laboratorní úlohu pro výukové účely.

Práce je členěná do devíti kapitol. V první kapitole jsou uvedeny obecné typy PLC jednotek. Druhá kapitola se věnuje vnitřní konstrukci programovatelného automatu. Ve třetí kapitole je uveden princip činnosti programovatelných automatů. Čtvrtá kapitola se zabývá programovacími jazyky dle normy *IEC 61131-3*. Pátá kapitola se věnuje struktuře a hlavními parametry programovacího automatu Modicon M241 TM241CEC24T. Šestá kapitola se věnuje programovacímu softwaru Machine Expert, grafické znázornění a stručný návod pro programování v softwaru. V sedmé kapitole je znázorněna struktura a realizace výukové rozšiřující desky. Osmá kapitola se věnuje realizaci demonstrační úlohy a řídicímu programu úlohy. V deváté kapitole byl vytvořen laboratorní návod.

1 Typy PLC

V dnešní době technický průmysl vyžaduje automatické ovládání zařízení a různých aplikací. Pro tyto účely využíváme programovatelné logické automaty nebo také známé jako PLC. PLC je průmyslový počítačový řídicí systém, který přijímá informace ze vstupních signálů (senzory, tlačítka a spínače) a spouští výstupy na základě předem naprogramovatelném programu. PLC rozdělujeme na tři základní typy dle konstrukce a účelů:

- Mikro PLC
- Kompaktní PLC
- Modulární PLC

1.1 Mikro PLC

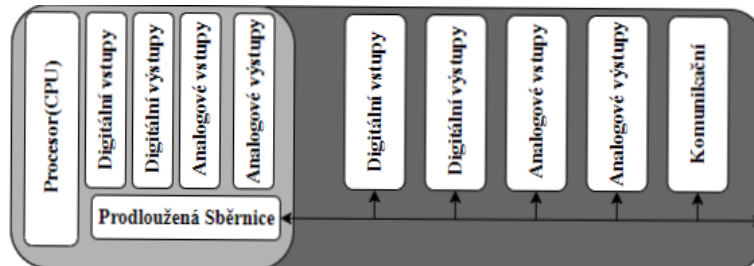
Mikro PLC jsou vhodné pro tvorbu jednodušších programů nebo rychlou změnu parametrů (většina typů). Změnu parametrů lze měnit přímo v terénu na integrovaném displeji bez nutnosti připojení počítače. Jedna z nevýhod je pevná sestava vstupů a výstupů a nelze je už rozšiřovat [1].



Obr. 1-1: Příklad mikro PLC [2]

1.2 Kompaktní PLC

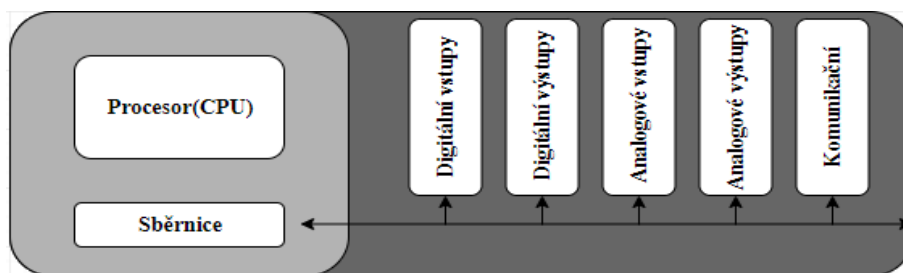
Kompaktní PLC má také jako mikro PLC pevnou sestavu vstupů a výstupů, ale n rozdíl od mikro PLC je možné přidat jiné moduly, ale rozšiřitelnost kompaktních PLC je omezená [3]. Tento typ PLC bude využit v rámci této bakalářské práce a bude níže podrobně popsán.



Obr. 1-2: Příklad kompaktního PLC s moduly

1.3 Modulární PLC

Modulární PLC je postaveno s několika moduly, které jsou zapojeny do společného stojanu nebo sběrnice s rozšiřitelnými možnostmi vstupu/výstupu. Stojan nebo sběrnice obsahuje napájecí modul, CPU a další vstupní/výstupní moduly (analogové nebo digitální) a moduly s komunikačními schopnostmi (Ethernet, Wifi), které mohou být od stejných výrobců nebo od jiných výrobců. Modulární PLC jsou dostupné v různých velikostech s variabilním napájením, výpočetními schopnostmi, vstupní/výstupní konektivitou [4].



Obr. 1-3: Příklad modulárního PLC s moduly

2 Vnitřní konstrukce PLC

Struktura PLC se skládá na pět hlavní části, jak je znázorněno na Obr.2-1.

- **Napájecí zdroj**

Poskytuje stejnosměrné nebo střídavé napájení vstupním/výstupním modulům, paměťovému systému a procesoru [5].

- **CPU (centrální jednotka)**

Ukládá a spouští softwarové programy, pomáhá provádět základní aritmetické, logické, ovládací a vstupní a výstupní operace uvedené v pokynech [6].

- **Paměť**

Je zodpovědná za získávání a ukládání dat a informací.

- **RAM (operační paměť s možností čtení i zápisu)**

Je dočasná paměť používá se při spouštění programu PLC tato paměť je pokaždé vymazána, když je napájení vypnuto. Proto nové PLC systémy mají baterii, která slouží k zálohování dat při výpadku napájení [7].

- **ROM (paměť pouze pro čtení)**

Je trvalá paměť používá se k uložení operačního systému PLC, který provádí na programovaný program [7].

- **Vstupní a Výstupní moduly**

Spojuje mozek PLC, CPU, s okolím. Vstupní modul detekuje vstupní signál (tlačítka, spínače, snímače teploty atd), výstupní modul řídí zařízení (relé, spouštěče motorů, světla atd.) [8].

- **Analogový**

Analogové Vstupy / Výstupy označují signály, které mají rozsah hodnot (například 0-10 VDC) [8].

- **Digitální**

Diskrétní nebo digitální Vstupy / Výstupy označují signály, které mohou být zapnuté nebo vypnuté (tlačítka nebo osvětlení) [8].

- **Komunikace**

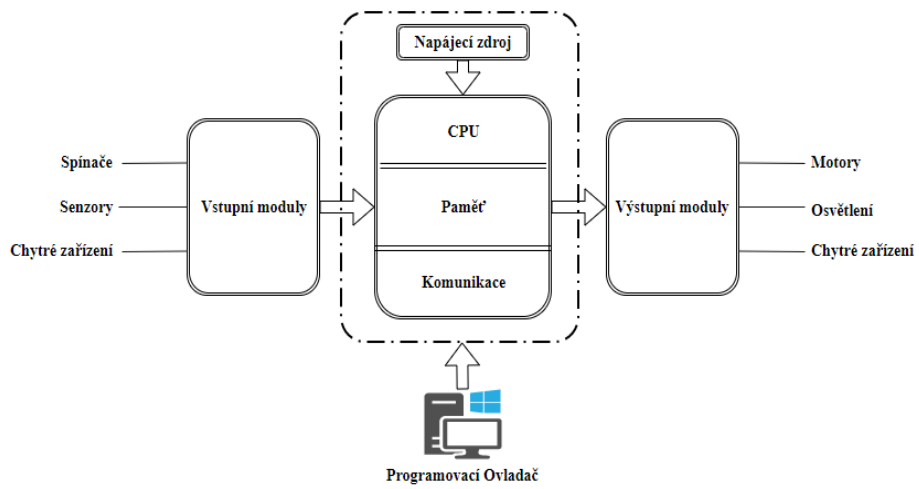
Slouží k přenosu dat z programovacího zařízení (PC nebo Smart zařízení) nebo pro ovládání PLC z dálky dvou (např. sériová linka, USB nebo ethernetu atd.) [6].

- **Programovací zařízení**

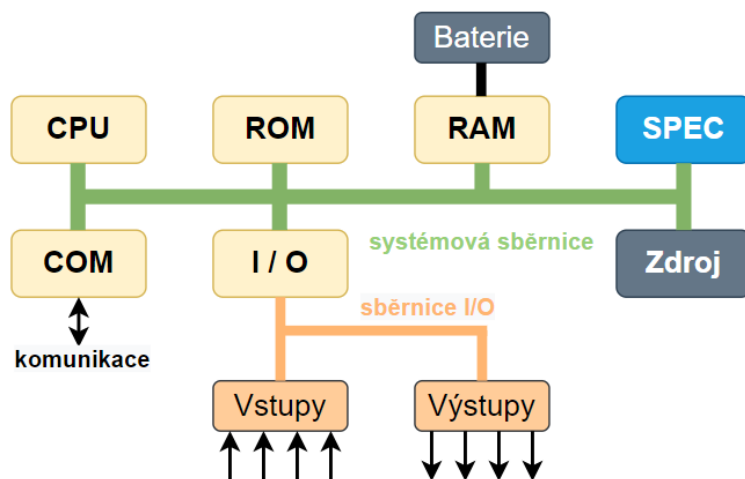
Programovací zařízení je hlavní částí pro PLC systémy, celý program se naprogramuje na tomto zařízení (PC nebo Smart zařízení), který se poté nahrává (přes USB nebo sériovou linku atd.) do PLC.

- **SPEC (speciální funkce)**

Některé typy PLC jsou vybaveny tak zvanými speciálními funkcemi, které obsahují modul hodin, časovače, čítače, sekvenční registry a případně algoritmy pro regulace a matematické funkce [6].



Obr. 2-1: Schéma PLC systému



Obr. 2-2: Blokovaná struktura PLC systému [6]

3 Princip činnosti

Všechny PLC systémy zahrnují jeden jednoduchý proces tzv. skenovací cyklus viz. na Obr.2-5. U některých PLC systému jsou přidány speciální hardwarové moduly, a proto jsou vyžadovány další skenovací cykly [9].

3.1 Skenovací cyklus

Když se PLC spustí provede se auto test, který provede kontrolu hardwaru a softwaru na závady. Poté jestli nedojde k problémům, PLC spustí skenovací cyklus, který se skládá ze tří kroků:

- **Vstupní skenování**

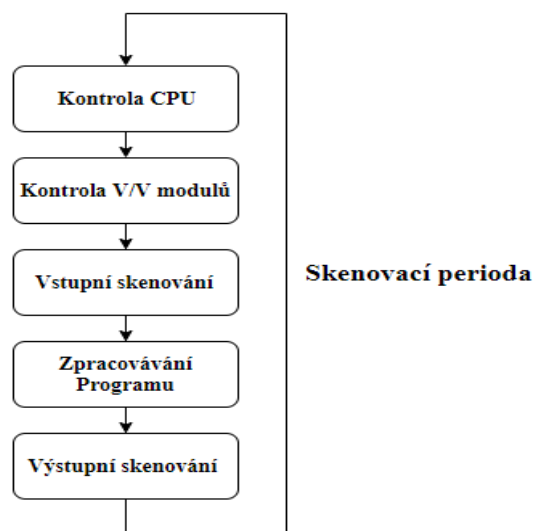
PLC prohlédne všechny vstupních perifériích signálů, aby určil jestli je zapnuto nebo vypnuto a uloží tyto informace do paměti, která se označuje obraz procesních vstupů (PII - Process Input Image) [10].

- **Zpracování programu**

Jakmile PLC přečte každý stav vstupního signálů (Zapnutý/Vypnutý), který he uložen ve vyhrazené oblasti paměti. Začne provádět všechny instrukce programu od adresy 0 [10].

- **Výstupní skenování**

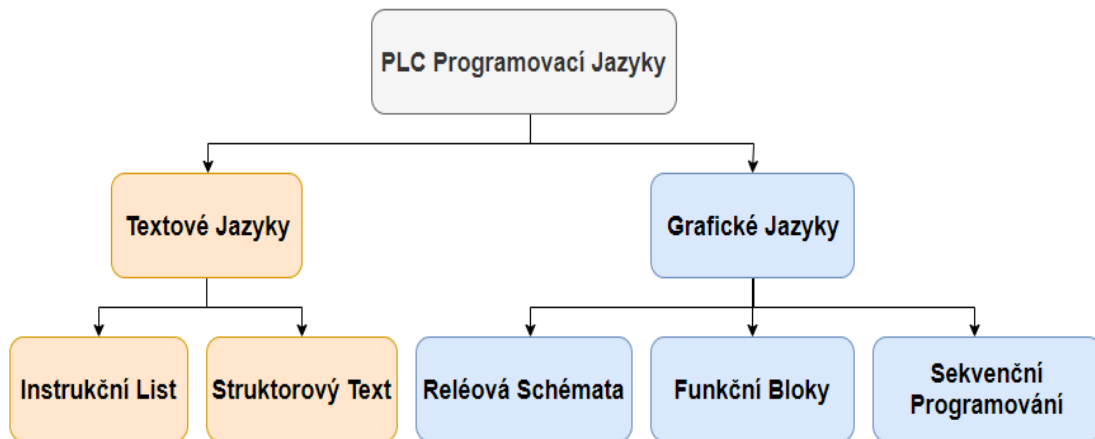
Při zpracovávání programu se už dle algoritmu vytvářejí výstupní signály. Stav signálů se neukládají přímo do výstupních periférií, ale do paměti zvané obraz procesních výstupů (POI – Process Output Image). Po dokončení programu, je stav výstupů zapsán na fyzické výstupy periférií [10].



Obr. 3-1: Schéma skenovacího cyklu PLC systému [10]

4 Programovací jazyky

Norma *IEC 61131-3* definuje 5 programovacích jazyků pro PLC, ale někteří výrobci umožňují i jiné jazyky. Programovací jazyky jsou rozdělené na dva druhy textové a grafické, jak je znázorněno na Obr.2-1. Mezi textové jazyky patří Instrukční list IL Instruction List a Strukturovaný text ST (Structured Text) a mezi grafické Funkční bloky FBD (Function Block Diagram), Reléová schémata LD (Ladder Diagram) a Sekvenční programování SFC (Sequential Function Chart) [11].



Obr. 4-1: Rozdělení PLC Programovacích Jazyků

4.1 Instrukční list IL (Instruction List)

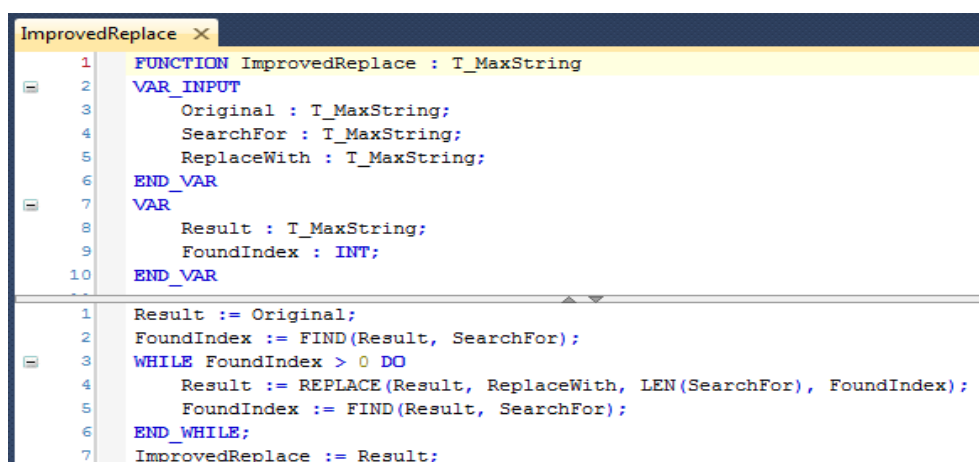
Instrukční list IL je programovací jazyk na nízké úrovni a velmi připomíná jazyk Assembler. Skládá se ze série instrukcí, každá instrukce začíná na novém řádku a obsahuje operátora v závislosti na typu operace, jeden nebo více operandů oddělených čárkami a můžeme připsat komentář [12]. Většina operací se provádí podle schématu: Výsledek: = výsledek OPERATOR operand komentář [13]. Výhodou je rychlost provádění programu a tendence zabírat méně paměti, nevýhodou je horší orientace a přehlednost programu [14].

1.1	LD SM 0.1	On for One Scan
1.2	MOVD #4000 , VD200	Put 4000 in address
1.3	LD SM 0.1	VD200
1.4	MOVD# 41 , VW10	Put 41 in address VW10
1.5	LD SM 0.1	
1.6	DIV VW10 , VD200	DIV Value in Address VD200
		On the Value VW10
1.7	MEND	put result in the address VW200

Obr. 4-2: Příklad programovacího jazyka IL [15]

4.2 Strukturovaný text ST (Structured Text)

Strukturovaný text ST je programovací jazyk na vyšší úrovni, který je velmi podobný programovacím jazykům pascal nebo C [14]. Zápis tvoří posloupnost symbolických instrukcí, jedna instrukce reprezentuje celou posloupnost základních instrukcí. Kód využívá příkazy, které jsou odděleny středníkem a tyto příkazy dále mění buď vstupy, výstupy nebo proměnné [16]. Vyhodnocení výrazu záleží na aplikování operátorů na operandy s uznáváním priorit operátorů. Nejprve jsou aplikovány operátory s nejvyšší prioritou, dále se postupně využívají operátory s menší prioritou, až je vyhodnocení výrazu dokončeno [13]. Strukturovaný text je velice vhodný ke složitým algoritmům, dlouhým matematickým funkcím a manipulaci s poli, nevýhodou je horší přehlednost zápisu logických operací [14].

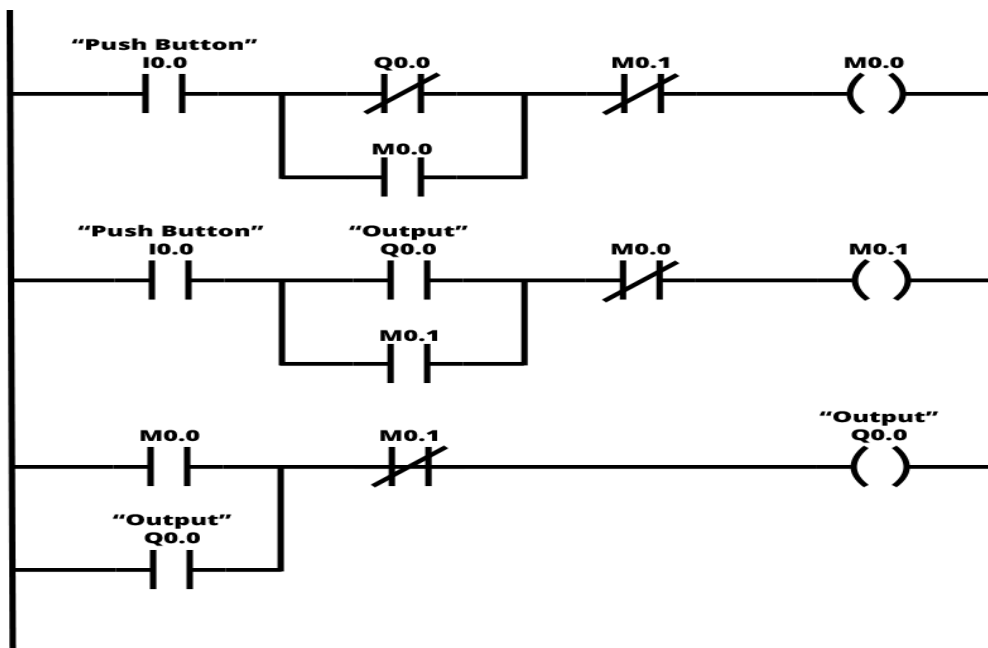


```
ImprovedReplace X
1  FUNCTION ImprovedReplace : T_MaxString
2  VAR_INPUT
3      Original : T_MaxString;
4      SearchFor : T_MaxString;
5      ReplaceWith : T_MaxString;
6  END_VAR
7  VAR
8      Result : T_MaxString;
9      FoundIndex : INT;
10 END_VAR
11
12 Result := Original;
13 FoundIndex := FIND(Result, SearchFor);
14 WHILE FoundIndex > 0 DO
15     Result := REPLACE(Result, ReplaceWith, LEN(SearchFor), FoundIndex);
16     FoundIndex := FIND(Result, SearchFor);
17 END_WHILE;
18 ImprovedReplace := Result;
```

Obr. 4-3: Příklad programovacího jazyka ST [17]

4.3 Reléová schémata LD (Ladder Diagram)

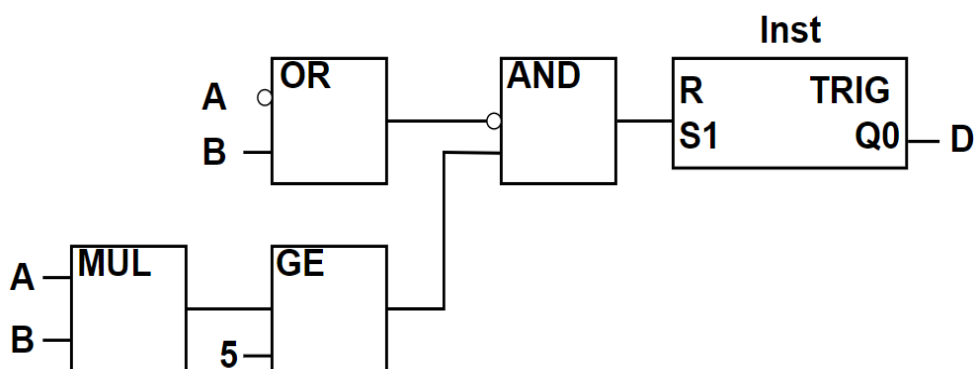
Tento programovací program vychází z reléové logiky a je velmi podobný liniovému schématu v elektrotechnice. Schéma má z obou stran dvě svislé čáry. Mezi dvěma čarami jsou příčky s kontakty nebo s cívkami relé nebo stykače a další prvky, příčky s kontakty mají mnoho typů a představují vstupy (přepínače, tlačítka a senzory atd.) i příčky s cívkami mají mnoha typů a představují výstupy (osvětlení, motory atd.) [18]. Výhodou je přehled zápisu programu a rychlé programování logických funkcí a nevýhodou je nepraktický zápis pro aritmetické operace a práce s daty [14].



Obr. 4-4: Příklad programovacího jazyka LD [19]

4.4 Funkční bloky FBD (Function Block Diagram)

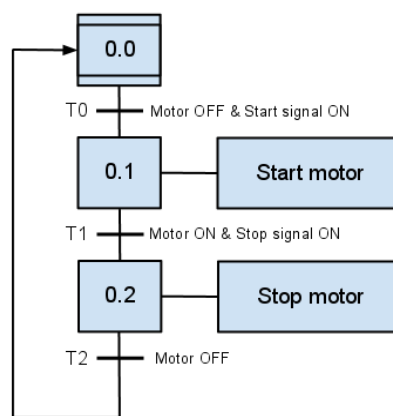
Programovací jazyk FBD popisuje funkci mezi vstupy a výstupy. Skládá se ze soustavy propojených bloků. Tyto bloky mohou být logické (OR, AND, XOR nebo XAND atd.), funkční (čítače, časovače atd.) nebo speciální funkce dle potřeby, bloky jsou propojeny spojnicemi [16]. Výhodou je přehledný zápis programu a možnost vzít mnoho řádků kódu a vložit je do jednoho bloku nebo několika funkčních bloků, nevýhodou je dezorganizace kódů, protože funkční bloky můžete umístit kdekoli na listu to může také ztížit řešení problémů [14].



Obr. 4-5: Příklad programovacího jazyka FBD [14]

4.5 Sekvenční programování SFC (Sequential Function Chart)

Sekvenční programovací jazyk je velmi podobný vývojovému diagramu (větvení na základě podmínek). Tento programovací jazyk používá kroky a přechody k dosažení konečných výsledků. Každý krok je příkaz nebo akce, která se provádí, když je krok aktivní, ale krok lze aktivovat, provést a deaktivovat. Přechod z jednoho kroku na druhý nastane při splnění nastavené podmínky [20]. Výhodou je přehledný zápis chování programu a vhodný pro realizaci sekvenční logiky, nevhodné pro programování složitých algoritmů [14].



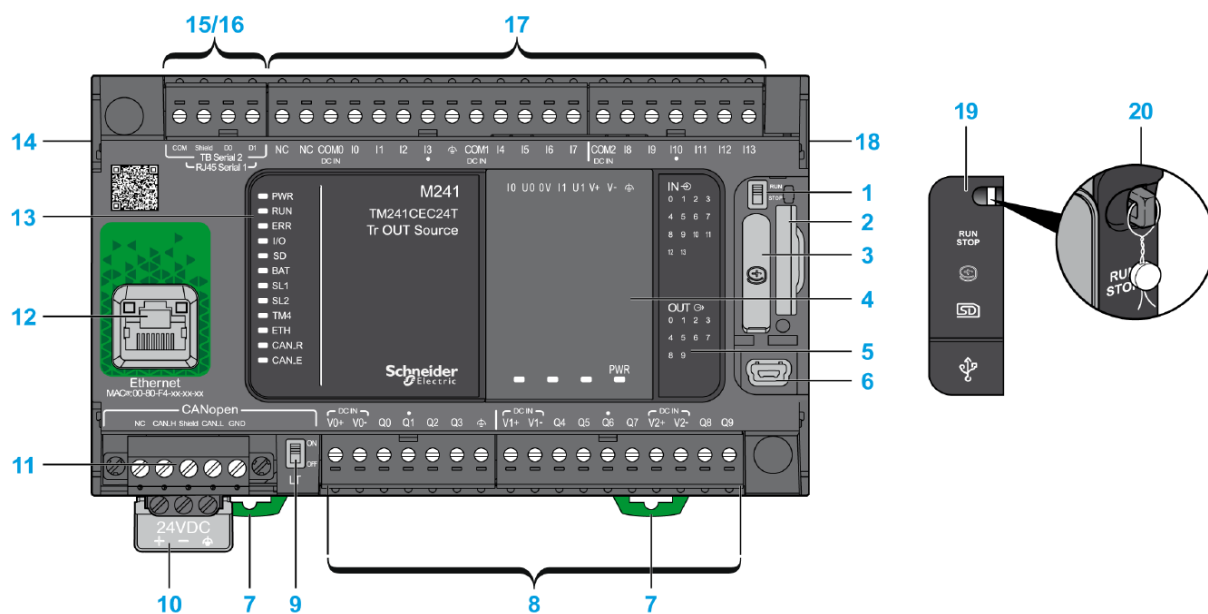
Obr. 4-6: Příklad programovacího jazyka SFC [21]

5 PLC Modicon M241 TM241CEC24T

PLC Modicon M241 TM241CEC24T je výrobek firmy Schneider Electric a je to kompaktní PLC. Programuje se v softwarovém prostředí Machine Expert, který je také výrobek firmy Schneider Electric. Napájí se zdrojem 24 V DC, má dva druhy paměti první druh je RAM 64 MB, z toho 8 MB je pro vykonávání aplikace a druhá je Flash 128 MB, která slouží pro ukládání dat a programu v případě výpadku napájení. Také obsahuje systém reálného času, zachová čas při vypnutém napájení, proto je nutná baterie. Hardwarový Run/Stop přepínač, který se využívá ke spuštění nebo zastavení aplikačního programu. Obsahuje 14 digitálních vstupů (4 normální a 8 rychlých), 10 digitálních výstupů (6 normálních a 4 rychlé) a 5 komunikačních portů (2 sériové porty, 1 Ethernet port, 1 CANopen port a 1 programovací port USB mini-B). Indikační LED display slouží k signalizaci stavu PLC (připojení, vstupů, výstupů, napájení a závad) [22].

Tab. 5-1: Hlavní parametry PLC Modicon TM241CEC24T [22]

Napájecí napětí	24 VDC
Výstupní napětí	24 VDC
Vstupní proud	5 mA (rychlý vstup 10.7 mA)
Výstupní proud	500 mA (rychlý výstup 100 mA)
Počet vstupů	14 (8 rychlé)
Počet výstupů	10 (4 rychlé)
Typ výstupu	Tranzistorový
Max. výstupní frekvence	1 kHz (rychlé výstupy až 100 kHz)
Paměť	RAM 64 MB (8 pro vykonávání aplikace) Flash 128MB



Obr. 5-1: Popis části PLC Modicon TM241CEC24T [22]

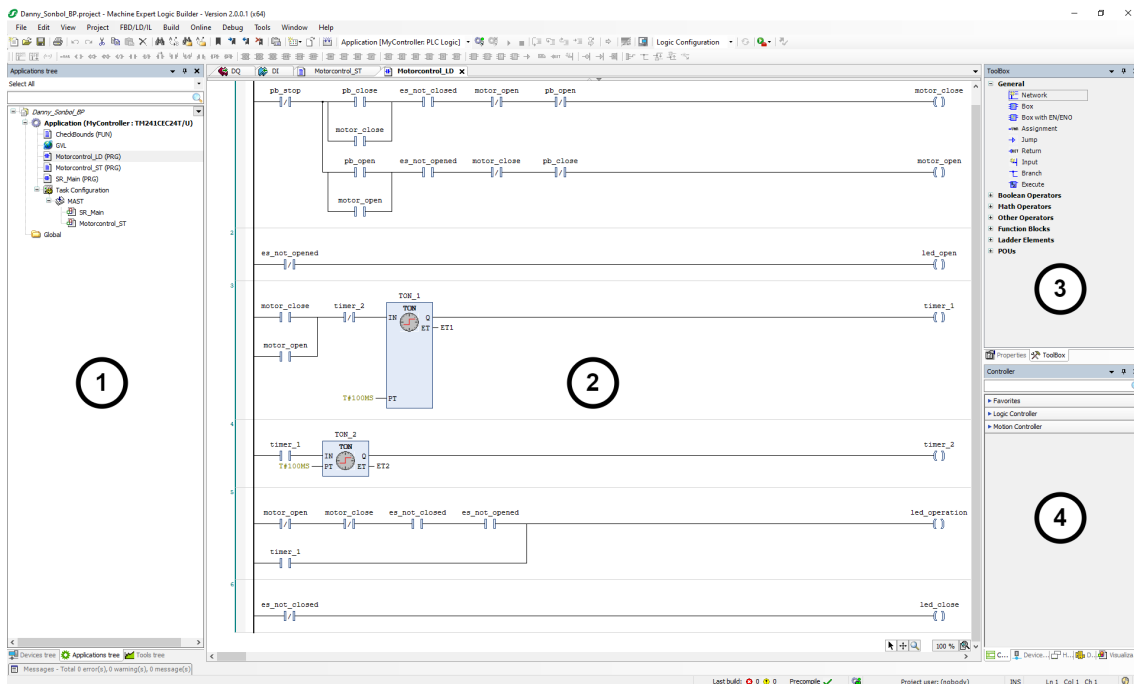
Tab. 5-2: Popis části PLC Modicon TM241CEC24T [22]

Číslo	Popis
1	Run/Stop přepínač
2	Slot na SD kartu
3	Slot na baterku pro RTC
4	Slot na cartridge
5	Indikační LED diody pro V/V stavy
6	USB mini-B port
7	Úchyt na DIN lištu
8	Svorkovnice výstupu
9	Přepínač CANopen
10	Napájení 24 VDC
11	CANopen port
12	Ethernetový port
13	LED diody pro status
14	Sběrnice pro připojení TM4 modulů
15	Sériová linka pro RS 232 nebo RS 485
16	Sériová linka pro RS 486
17	Svorkovnice vstupů
18	Sběrnice pro připojení pro TM3/TM2 moduly
19	Ochranný kryt
20	Úchyt pro zámek

6 Programovací prostředí Machine Expert

Machine Expert je programovací software od firmy Schneider Electric. Umožňuje programování, nastavování a řízení programovatelného automatu. Program využívá šest programovacích jazyků IL, ST, LD, FBD, SFC a volně propojované bloky CFC (Continuous Function Chart).

Při spuštění programu Machine Expert se objeví tzv. centrální obrazovka, jak lze vidět na Obr. 6-1.



Obr. 6-1: Centrální obrazovka softwaru Machine Expert

V oblasti 1 jsou tři záložky:

- *Devices tree*: Zde jsou uvedeny hardwarové komponenty, lze také konfigurovat hardwarovou komunikaci a přiřazovat aplikace.
- *Application tree*: V této záložce dochází k vytváření nových programů a konfigurace úloh.
- *Tools tree*: Zde lze přidávat a odebírat knihovny, které jsou využity v projektu.

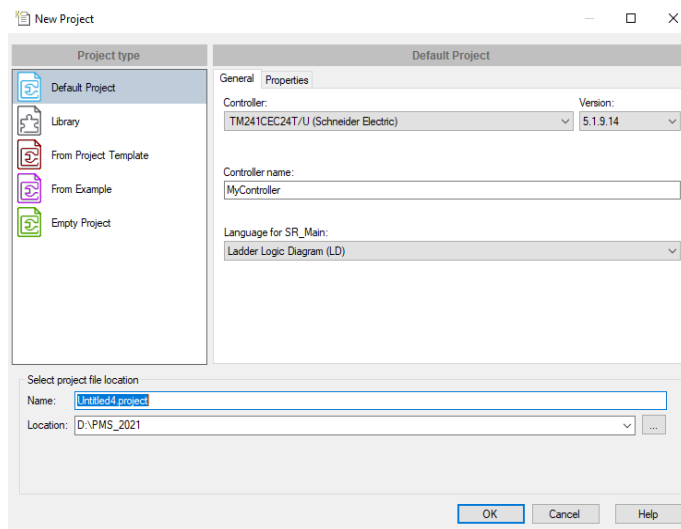
Oblast 2 (Pracovní oblast): V této oblasti dochází k zápisu programu, propojování vstupů a výstupů, vytváření globálních proměnných a další funkce.

Oblast 3 (*Toolbox*): Zde se nacházejí různé nástroje, funkce, elektrické a logické prvky, které poté přesuneme do pracovní oblasti.

Oblast 4 (*Controller*): Zde nalezneme rychlý přístup do seznamu oblíbených a často používaných prvků [23].

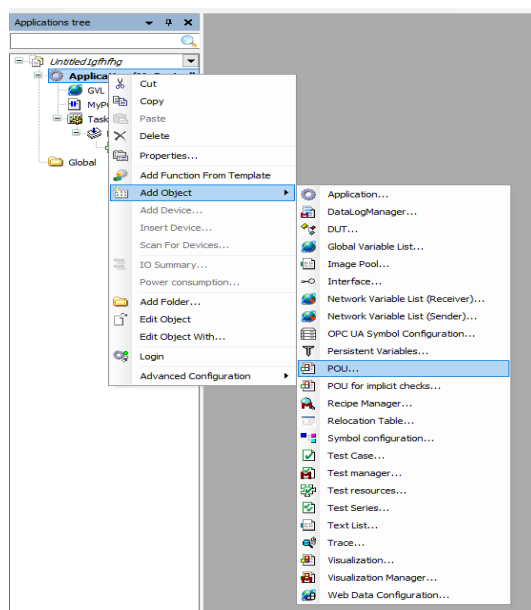
6.1 Vytváření nového projektu

Pro vytvoření nového projektu klikneme na hoře v levé straně *file* poté *new project*. Otevře se nové okno, jak lze vidět na Obr. 6-2, v novém oknu lze nastavit: typ plc, jméno projektu a typ programovacího jazyku. Také je možné nastavit různé šablony, šablony z projektů a knihovny.



Obr. 6-2: Postup při vytváření nového projektu

Na levé straně v *Application Tree Menu*, stiskneme pravým tlačítkem na *Application (My-Controller)* poté se vybere *Add object* a poté *POU* viz. na Obr. 7-6. Následně se zobrazí nové okno, ve kterém se mění jméno a nastavuje programovací jazyk po potvrzení se zobrazí programovací obrazovka, kde lze tvořit či psát program.



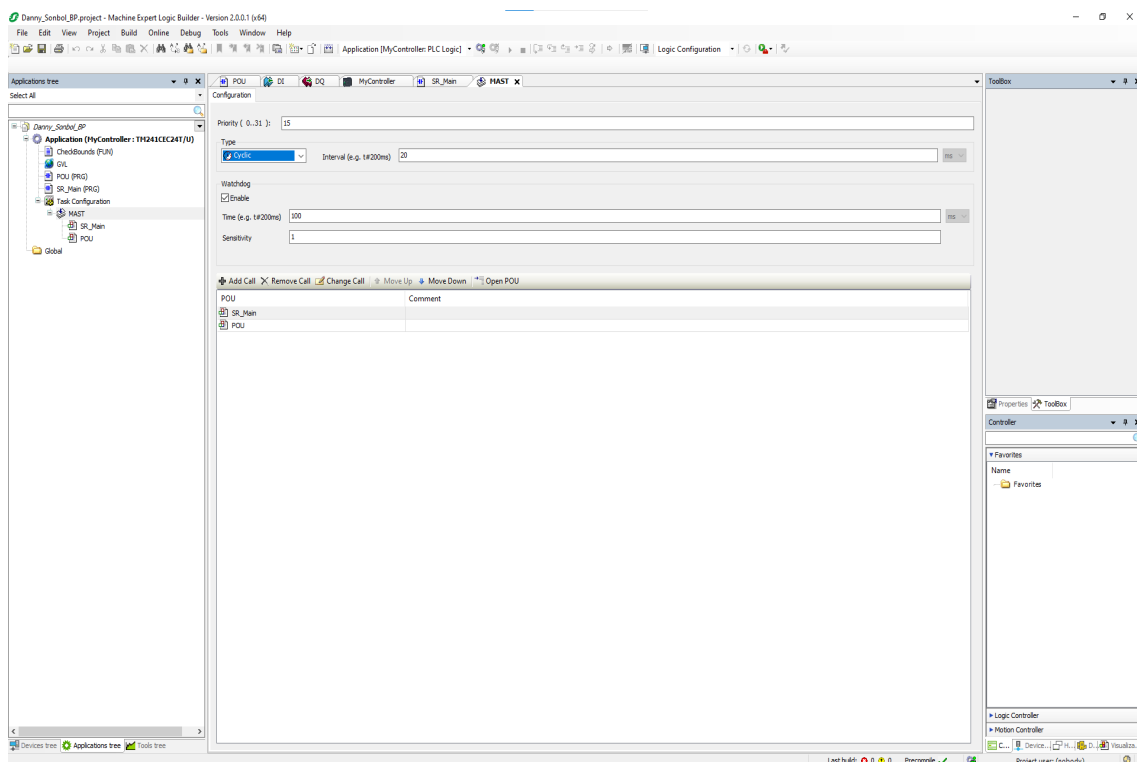
Obr. 6-3: Ukázka vytváření POU

Po dokončení zápisu programu je vhodné propojit vstupy a výstupy s proměnnými, jak je znázorněno na Obr. 6-4. V záložce *Device Tree menu* se klikne na *DI (digital inputs)* kde se propojí vstupy a *DQ (digital outputs)* kde se propojí výstupy. Dvakrát se klikne na sloupec *variable* objeví se nové okno, kde se vybere *application* a poté se vybere proměnná. Po dokončení programování a propojení vstupů a výstupů následuje kompilace, kterou lze spustit tlačítkem F11 nebo v horním *menu build*, aby se zjistilo, jestli v programu nejsou chyby.

Variable	Mapping	Channel	Address	Type	Default Value	Unit	Description
Inputs		IW0	%IW0	WORD			
		I0	%IX0.0	BOOL			Fast input, Sink/Source
		I1	%IX0.1	BOOL			Fast input, Sink/Source
stop		I2	%IX0.2	BOOL			Fast input, Sink/Source
pb 1		I3	%IX0.3	BOOL			Fast input, Sink/Source
pb		I4	%IX0.4	BOOL			Fast input, Sink/Source
aux		I5	%IX0.5	BOOL			Fast input, Sink/Source
aux 1		I6	%IX0.6	BOOL			Fast input, Sink/Source
		I7	%IX0.7	BOOL			Fast input, Sink/Source
		I8	%IX1.0	BOOL			Regular input, Sink/Source
		I9	%IX1.1	BOOL			Regular input, Sink/Source
		I10	%IX1.2	BOOL			Regular input, Sink/Source
		I11	%IX1.3	BOOL			Regular input, Sink/Source
		I12	%IX1.4	BOOL			Regular input, Sink/Source
		I13	%IX1.5	BOOL			Regular input, Sink/Source
		IB1	%IB2	BYTE			

Obr. 6-4: Ukázka propojování Vstupů

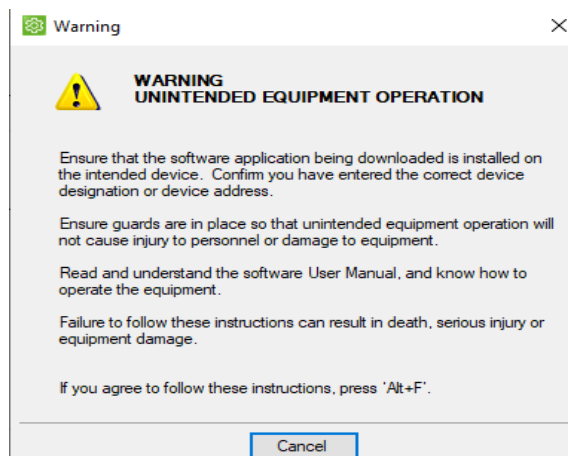
Před nahrání programu do programovacího automatu musíme v *Task Configuration* přidat vytvořené *POU*, a to kliknutím na *ADD Call* a vybrat vytvořené *POU*, jak lze vidět na Obr. 6-5. Tento krok je velmi důležitý, a to proto aby byl program načítán.



Obr. 6-5: Ukázka Task Configuration

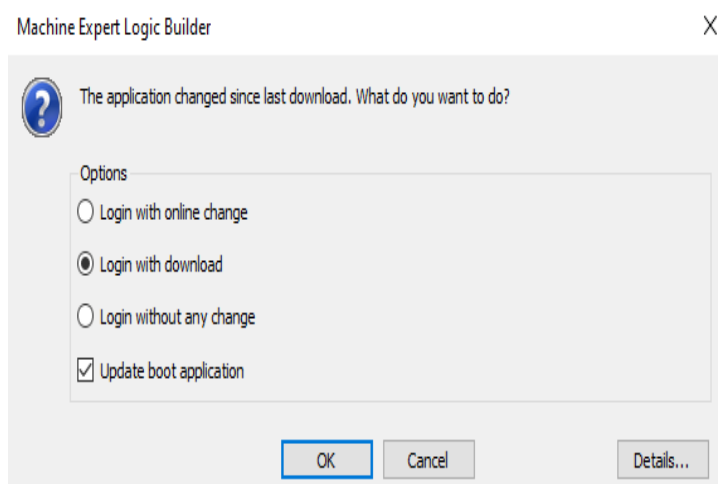
6.2 Nahrání programu do PLC

Pro nahrání programu do PLC, klikneme nahoře v panelu *Online* a poté *Login* nebo klávesovou zkratkou ALT+F8. Otevře se varovné okno, jak je znázorněno Obr.6-6, které upozorňuje na to, jestli je program nahrán do správného PLC nebo na PLC adresu, pro souhlas musíme stisknout klávesovou zkratku ALT+F.



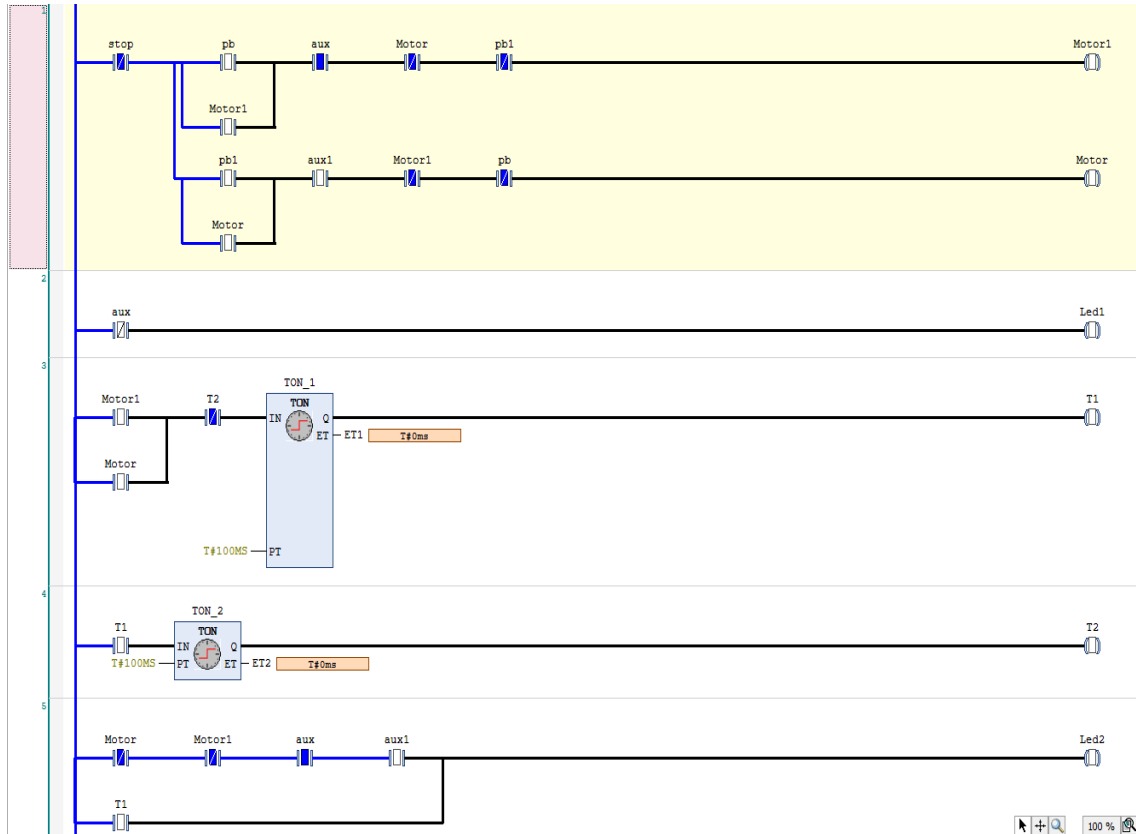
Obr. 6-6: Ukázka varovného okénka

Objeví se nové okno viz. Obr.6-7 ve kterém jsou tři možnosti, první možnost propojit program s PLC s online změnami, druhá propojit a nahrát program do PLC a třetí možnost je propojit ale bez změn.



Obr. 6-7: Ukázka nahrávání programu

Po nahrání programu do PLC se vytvořené *POU* změní na *Simulation* jak lze vidět na Obr.6-8, ve kterém lze libovolně měnit proměnné a vyzkoušet, jestli vše funguje dle požadavků.



Obr. 6-8: Simulation v softwaru Machine Expert

7 Výuková rozšiřující deska pro PLC

Pro účelné využití PLC Modicon ve výuce je vhodné vytvořit jednoduchou rozšiřující desku, která bude umožňovat snadné připojení základních periférií a realizovat tak jednoduché úlohy.

7.1 Struktura rozšiřující desky

Deska obsahuje:

Vstupní prvky:

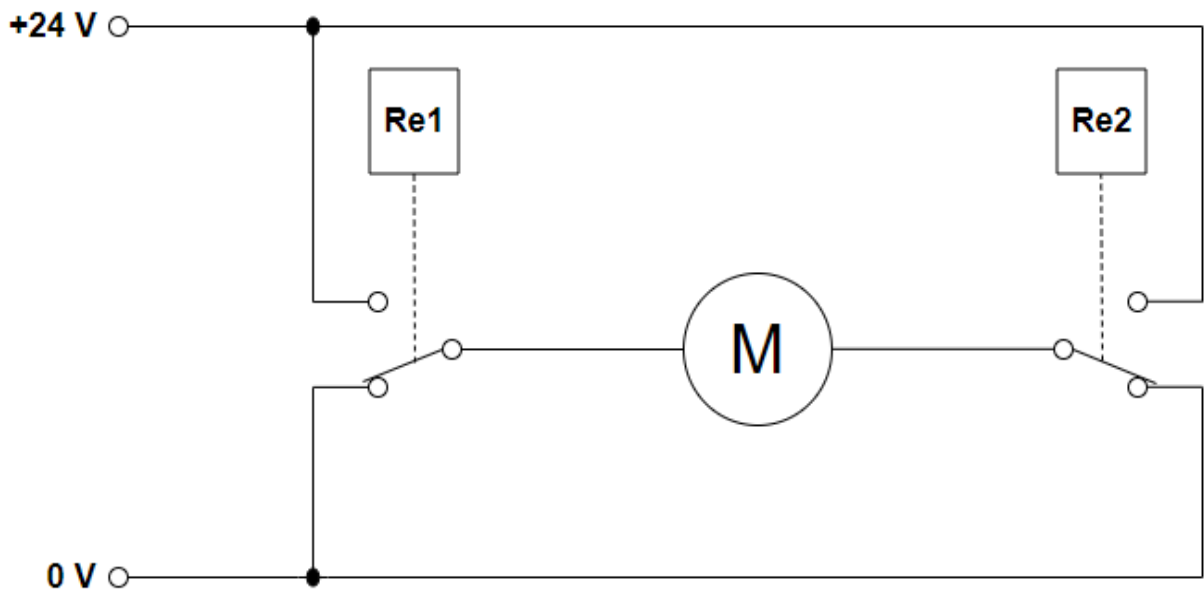
- Tři tlačítka Pu1, Pu2, Pu3
- Dva posuvné spínače Sw1, Sw2

Výstupní prvky:

- Tři LED Le1, Le2, Le3
- Dvě relé Re1, Re2
- Dva drivery pro krokové motory

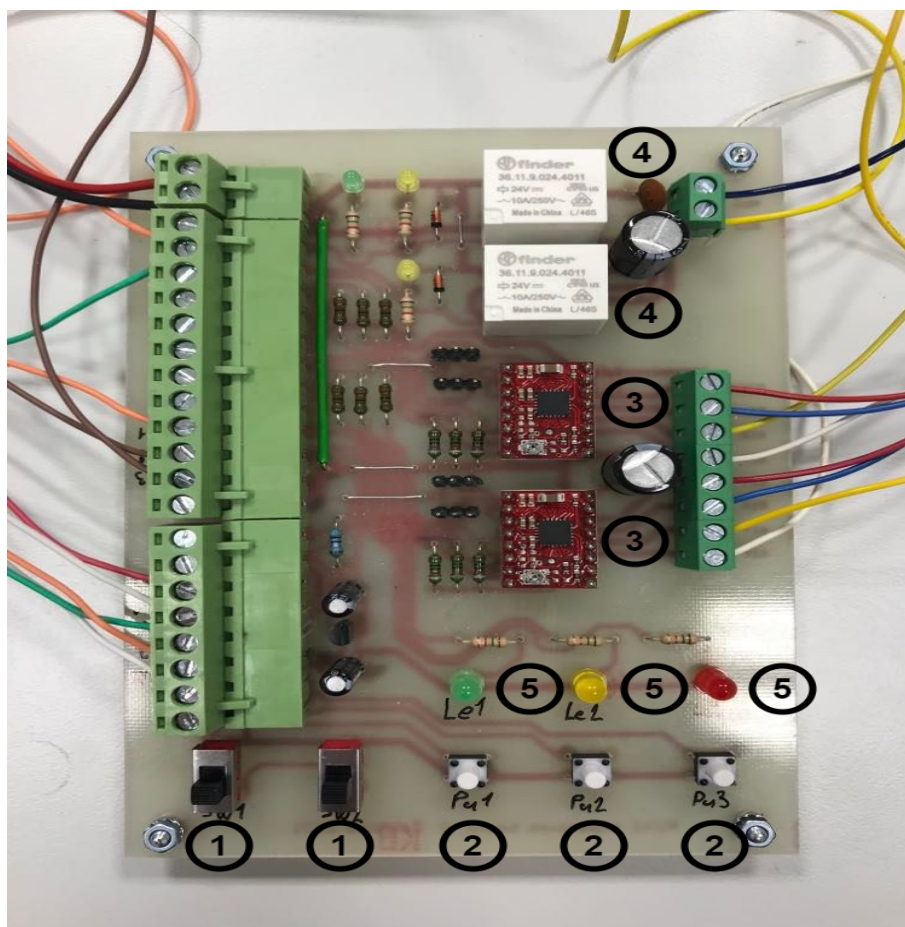
Deska je napájena ze společného zdroje 24 V jako samotné PLC a obsahuje ještě pomocné led signalizující přítomnost napájecího napětí a též sepnutí jednotlivých relé. Propojení s PLC je řešeno přes rychle odpojitelné svorkovnice a všechny vstupy i výstupy pracují s logikou 24 V, kde logická 1 představuje +24 V a logická 0 představuje 0 V.

Relé Re1 a Re2 jsou primárně určeny k spínání stejnosměrného motoru. Jejich přepínací kontakty jsou zapojeny tak, že v klidovém stavu je výstupní svorka pro připojení DC motoru připojena k zápornému pólu napájecího zdroje a po sepnutí relé se připojí ke kladnému pólu zdroje 24 V. Spínáním relé lze tedy motorem otáčet oběma směry a nebo jej zabrzdit zkratováním jeho kotvy.



Obr. 7-1: Zapojení silové části pohonu

Přítomnost driveru Pololu 2967 umožňuje k rozšiřující desce připojit dva hybridní krokové motory. Tato funkce je popsána a využita v rámci dřívější bakalářské práce [24]. Schéma zapojení rozšiřující desky je v příloze A



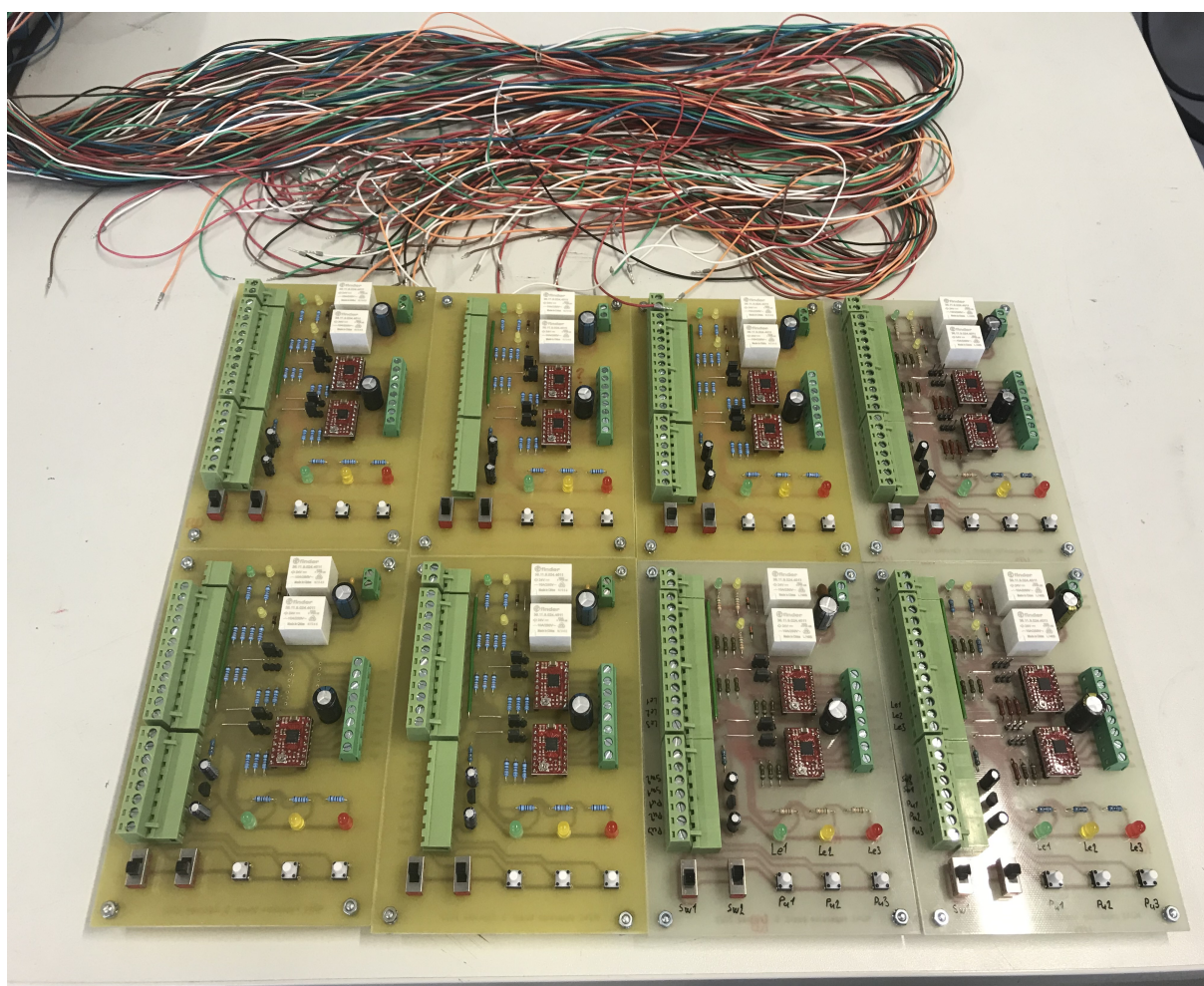
Obr. 7-2: Rozložení jednotlivých prvků na desce

Tab. 7-2: Seznam součástek na Obr. 7-2

č. součástky	Popis
1	Posuvné spínače Sw1, Sw2
2	Tlačítka Pu1, Pu2, Pu3
3	Drivery pro krokové motory
4	Relé Re1, Re2
5	LED Le1, Le2, Le3

7.2 Realizace desky

Dříve existující návrh desky dodaný vedoucím byl aktualizován a byly též opraveny všechny chyby a nedostatky. Následně bylo vyleptáno, osazeno a oživeno 8 kusů desek, pro kompletní zajištění potřeb výuky. Byly též vytvořeny sady propojovacích vodičů opatřených dutinkami. Vznikly tak naprosto funkční sady, které jsou připraveny na nasazení do výuky.



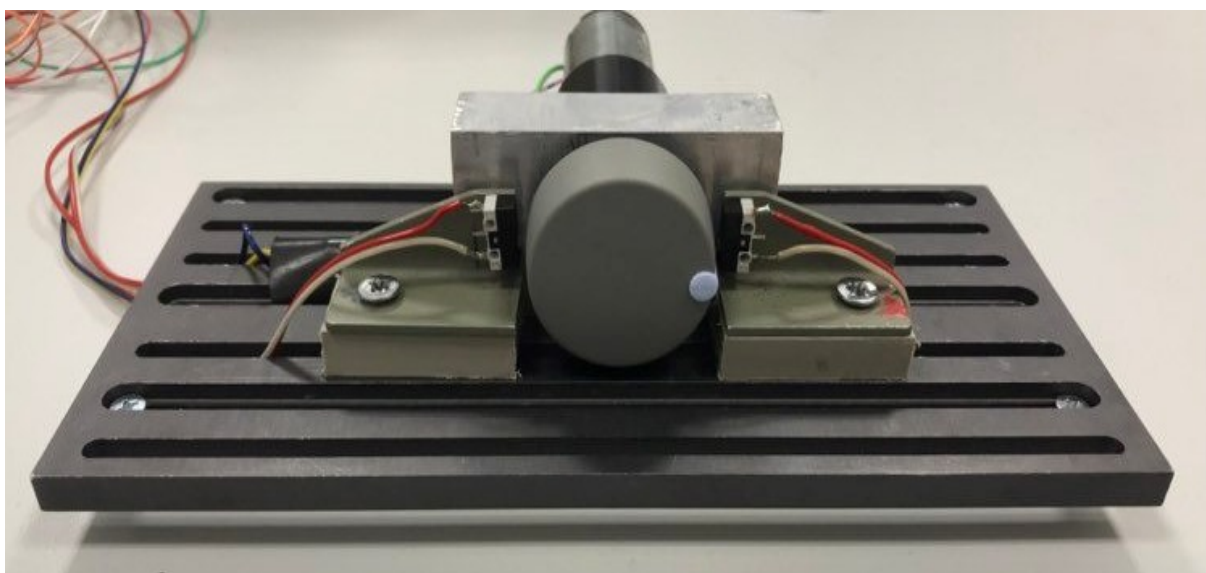
Obr. 7-3: Hotové sady připravené na nasazení ve výuce

8 Využití desek ve výuce

Jak již bylo uvedeno výše, desky byly navrženy tak, aby umožňovaly snadnou realizaci základních úloh průmyslové automatizace. Tyto úlohy budou vznikat postupně, dle požadavků výuky. Jako příklad jedné z výukových úloh zde uvádím jednoduché zařízení, které simuluje otevírání a zavírání průmyslové armatury pomocí stejnosměrného motoru. Úloha byla nejdříve realizována pomocí programovacího jazyku LD a následně též pomocí jazyku ST. Na těchto příkladech lze dobře demonstrovat výhody a nevýhody jednotlivých jazyků.

8.1 Realizace demonstrační úlohy Ovládání pohonu armatury

Úloha simuluje ovládání uzavírací armatury pomocí pohonu se stejnosměrným motorem a koncovými spínači. Pro účely úlohy byl vytvořen jednoduchý přípravek s motorkem a převodkou s vysokým převodovým poměrem, na jejímž výstupu je osazen kotouček ovládající dva koncové spínače.



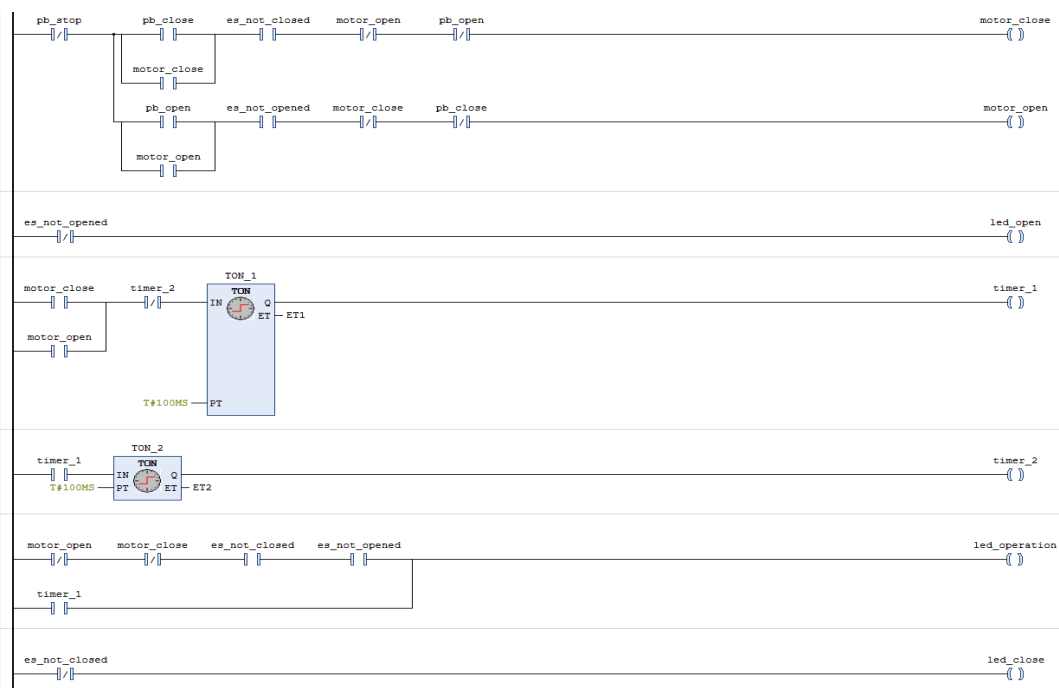
Obr. 8-1: Mechanismus s DC motorem a koncovými spínači

Smyslem úlohy je ovládání DC motoru prostřednictvím tří tlačítek se signalizací dosažení koncové polohy pomocí LED.

8.2 Řídicí program

Řídicí program byl naprogramován v softwaru Machine Expert za použití dvou programovacích jazyků LD (Ladder Diagram) a ST (Structured Text) pro dvě obtížnosti programování.

8.2.1 Řídicí program realizovaný jazykem LD



Obr. 8-2: Řídicí program (programovací jazyk LD)

Tab. 8-2: Význam jednotlivých symbolů použitých v programu

Symbol	Význam
pb_stop (negovaný)	Tlačítko pro zastavení motoru nebo signalizaci poruchy
pb_close, pb_open	Tlačítko pro otevření a uzavření armatury
es_not_closed, es_not_opened	Koncové spínače, které určují dosažení koncové polohy
motor_close, motor_open (kulaté závorky)	Výstupy pro zapojení PLC a relé na desce
motor_close, motor_open	Imaginární kontakty, které udržují motor v chodu při jednom stisknutí tlačítka
motor_close, motor_open (negovaný)	Imaginární kontakty, které chrání před zapnutím dvou funkcí na jednou
pb_close, pb_open (negovaný)	Imaginární kontakty, které umožňují přejít z jedné armatury na druhou při chodu jedné z armatur
led_close, led_open, led_operation (kruhové závorky)	Výstupy pro zapojení PLC a LED na desce
TON_1, TON_2	Časovače, které slouží k tomu, aby led_operation blikalo při chodu mezi armatury

8.2.2 Řídicí program realizovaný jazykem ST

```

// Funkce pro otevreni motoru
IF ((NOT pb_stop) AND ((pb_open OR motor_open) AND es_not_opened AND (NOT motor_close) AND (NOT pb_close)))
    THEN motor_open:= TRUE;
    ELSE motor_open:= FALSE;
END_IF
// Funkce pro zavereni motoru
IF ((NOT pb_stop) AND ((pb_close OR motor_close) AND es_not_closed AND (NOT motor_open) AND (NOT pb_open)))
    THEN motor_close:= TRUE;
    ELSE motor_close:= FALSE;
END_IF
// LED zapnuta pri zavreni
IF (NOT es_not_closed)
    THEN led_close:= TRUE;
    ELSE led_close:= FALSE;
END_IF
// LED zapnuta pri otevreni
IF (NOT es_not_opened)
    THEN led_open:= TRUE;
    ELSE led_open:= FALSE;
END_IF
// LED blikani pri procesu a zapnuta pri poruse
TIMER_1(IN:= (motor_open OR motor_close ) AND (NOT TM_2),PT:=T#100MS);
TM_1:=TIMER_1.Q;

TIMER_2(IN:=TM_1,PT:=T#100MS);
TM_2:=TIMER_2.Q;

IF (((NOT motor_close) AND (NOT motor_open) AND es_not_closed AND es_not_opened) OR TM_1)
    THEN led_operation:= TRUE;
    ELSE led_operation:= FALSE;
END_IF

```

Obr. 8-3: Řídicí program (programovací jazyk ST)

Tab. 8-3: Význam jednotlivých příkazů použitých v programu

Příkaz	Význam
Funkce pro otevření motoru	Po stisknutí tlačítka pb_open se motor otáčí do doby, než motor dosáhne do koncové polohy koncového spínače es_not_opened nebo při poruše (pb_stop) nebo při stisknutí tlačítka pb_close pro uzavření armatury
Funkce pro uzavření motoru	Po stisknutí tlačítka pb_close se motor otáčí do doby, než motor dosáhne do koncové polohy koncového spínače es_not_closed nebo při poruše (pb_stop) nebo při stisknutí tlačítka pb_open pro otevření armatury
Le1 zapnuto při uzavření	Když motor dosáhne ke koncovému spínači es_not_closed
Le3 zapnuto při otevření	Když motor dosáhne ke koncovému spínači es_not_opened
Le2 blikání při procesu a zapnuto při poruše	Při chodu jedné z armatur, jeden z časovačů zapíná a druhý vypíná a tím vznikne blikání. Druhá funkce, když nejsou v chodu armatury a také nedosáhly koncových spínačů Le2 signalizuje poruchu a je zapnutá.

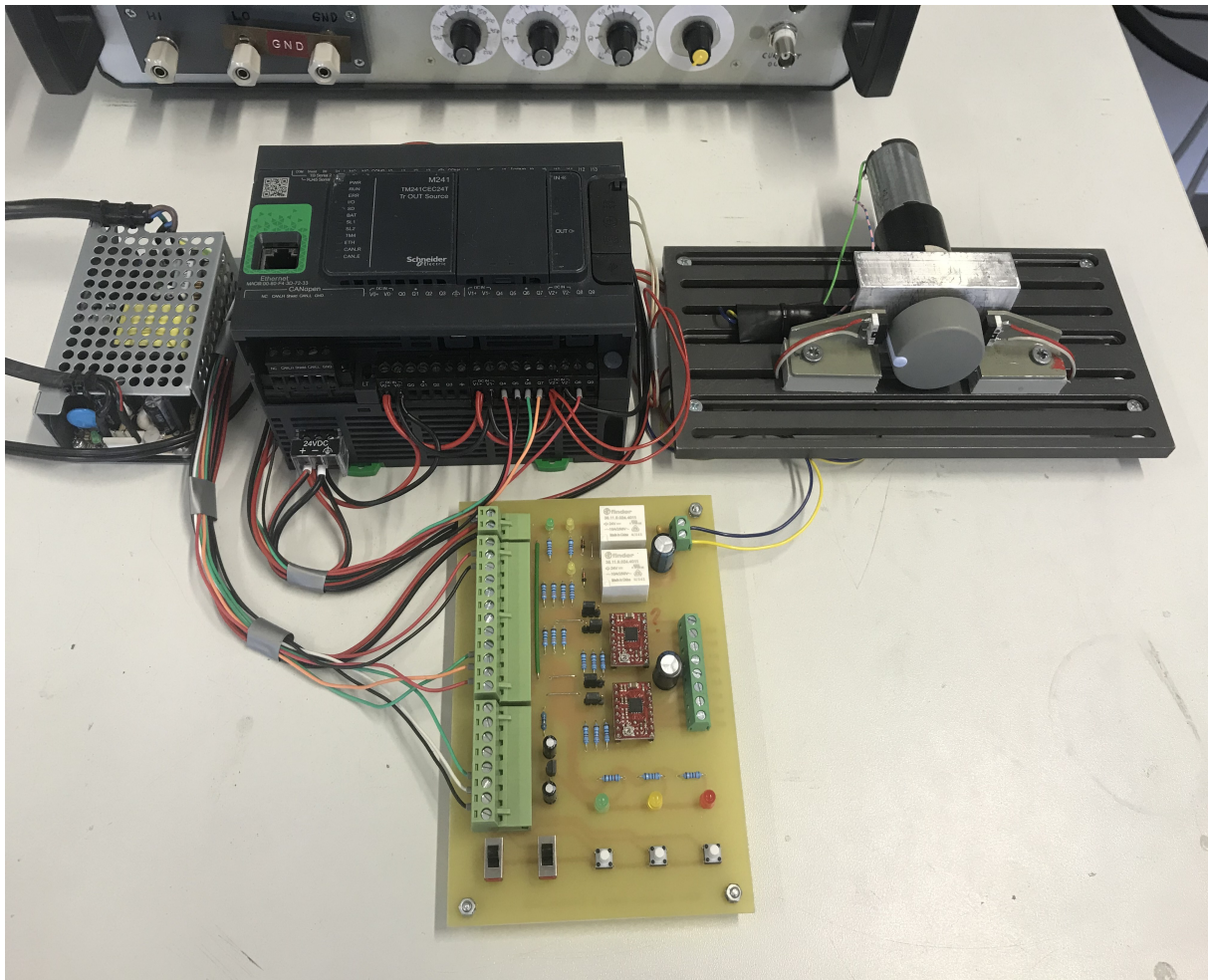
9 Laboratorní návod

Ovládání pohonu armatury v JE Temelín

Pohon mechanismu armatury

Pro ovládání průtoku pracovní látky potrubím se používají zařízení, které označujeme jako výstroj potrubí neboli armatury [25].

Elektrické armatury využívají pohonu elektromotoru, který je připojen pomocí převodovky, na jejímž výstupu je osazena momentová koncová a signalizační jednotka. Po umístění pohonu je velmi důležité seřízení momentových a koncových poloh, aby nedošlo ke zničení armatury. Tento problém je řešen mikropsínačem nainstalovaným na koncové jednotce. Servopohon vypne při dosažení nastavenému momentu nebo polohy mikropsínače pro daný typ armatury. [25].



Obr. 9-1: Laboratorní pracoviště

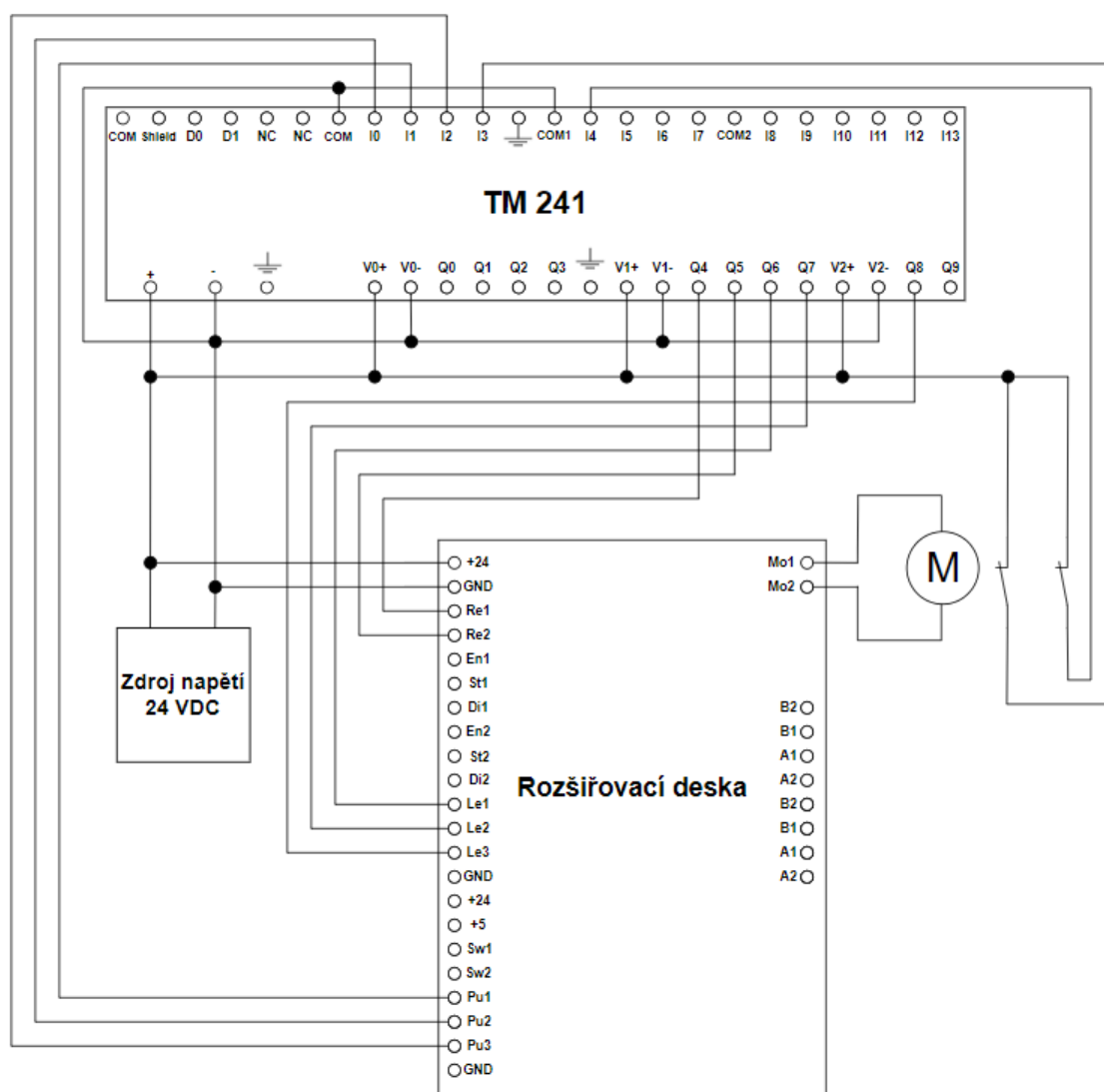
Zadání úkolu „Ovládání pohonu armatury“

Pro ovládání pohonu armatury budou použita tři tlačítka, jedno pro úplné otevření „Otevři“, jedno pro úplné uzavření „Zavři“ a jedno pro okamžité zastavení pohybu klapky „Stop“. Pro signalizaci uzavřeného či otevřeného stavu armatury budou použity kontrolky, z nichž zelená signalizuje stav „uzavřeno“ a červená stav „otevřeno“. Třetí, žlutá kontrolka bude při pohybu klapky blikat a trvale se rozsvítí jen při nouzovém zastavení a bude tak signalizovat nestandardní stav.

Pro realizaci úlohy využijete modul s DC motorem a koncovými spínači Ks1 a Ks2 a rozšiřující desku s ovládacími prvky. Na ní budou využity tyto prvky:

- Relé Re1 a Re2
- Tlačítka Pu1, Pu2, Pu3
- Kontrolky Le1, Le2, a Le3

Schéma zapojení:



Obr. 9-2: Schéma zapojení laboratorní úlohy

Postup řešení:

1. Zkontrolujte zapojení pracoviště
2. Otevřete Machine Expert a vytvořte *nový projekt*.
3. Pojmenujte projekt Ovládání pohonu armatury a využijte programovací jazyk LD
4. Založte nové *POU* a v *Application Tree Menu*, *Task configuration* přidejte nové *POU*
5. Vytvořte programovací program tak aby splnil tyto požadavky:
 - (a) Oba stavy armatur nesmí fungovat současně
 - (b) Při provozu jedné armatury lze přejít na druhou armaturu bez zastavení celého systému
 - (c) Tlačítka Pu1, Pu2 a Pu3 postačí jedno stisknutí tlačítka, aby přešel z jedné armatury na druhou (nedržet tlačítko celou dobu přechodu)
 - (d) Při provozu armatur lze kdykoliv zastavit celý systém (bude znázorňovat nestandardní stav)
 - (e) červená kontrolka signalizuje „otevřeno“, zelená signalizuje „uzavřeno“ a žlutá bude při pohybu klapky blikat, rozsvítí se jen při nouzovém zastavení a bude tak signalizovat nestandardní stav „Stop“
6. Pro určení koncové polohy armatury použijte koncové spínače
7. Pro blikání žluté kontrolky použijte dva funkční bloky TON (ON delay timer) jeden pro dobu zapnutí a druhý pro dobu vypnutí oba nastavte na $t=100$ ms
8. Po dokončení programu propojte vstupy a výstupy k proměnných
9. Zkontrolujte, jestli nejsou v programu chyby
10. Nahrajte program do PLC jednotky
11. Otestujte funkčnost programu

Laboratorní návod lze modifikovat i na programovacím jazyku strukturovaný text (ST) viz. kapitola 8.2.2 Řídící program realizovaný jazykem ST.

10 Závěr

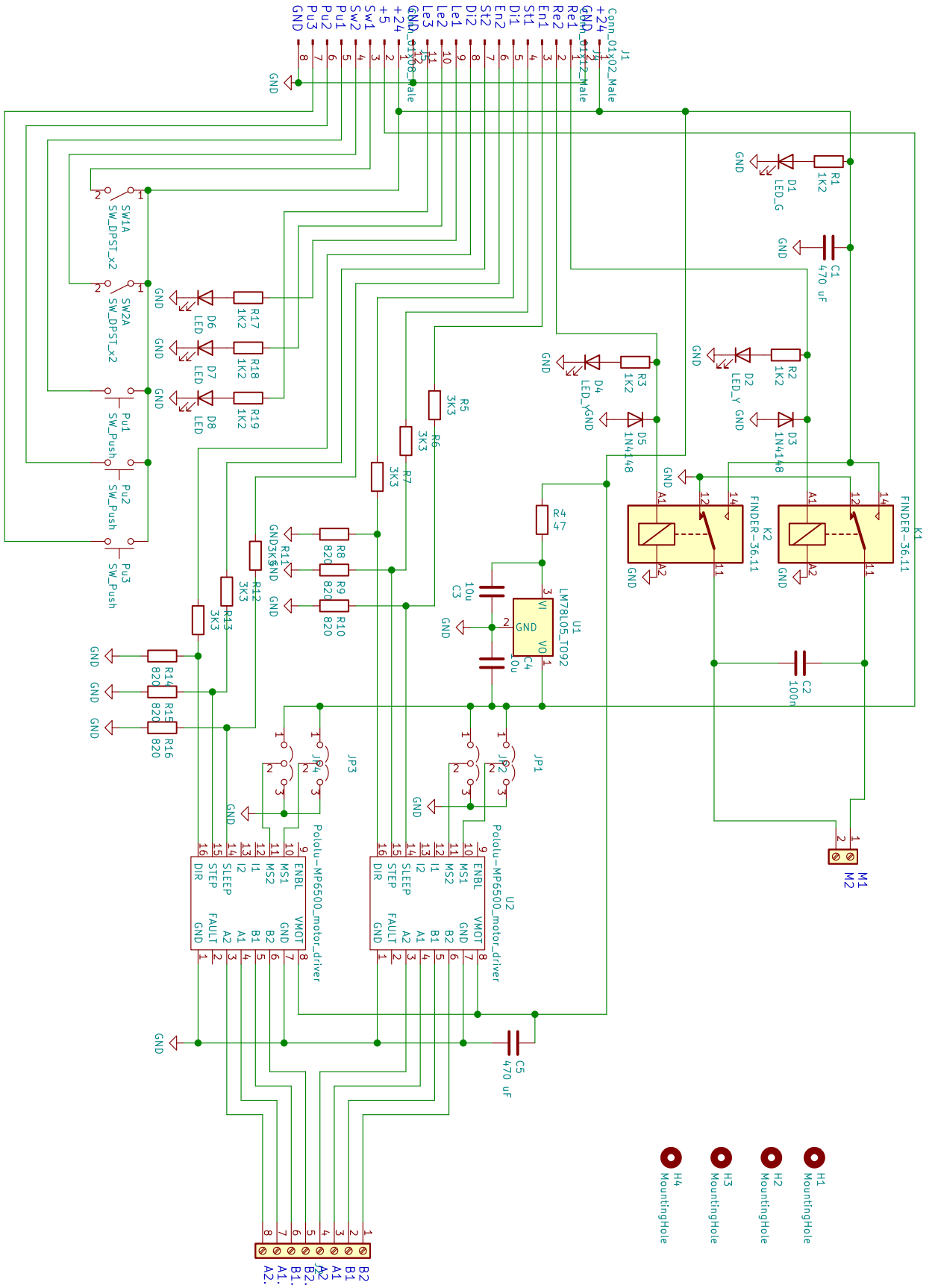
Bakalářská práce prezentuje stručný úvod do problematiky využití programovatelných logických automatů firmy Schneider Electric. V úvodu je popsána struktura a možnosti zvoleného PLC Modicon M241, dále je představeno programovací prostředí s vysvětlením postupu při vytváření projektu. V závěrečné části jsou popsány návrh a realizace sady jednoduchých rozšiřujících desek, umožňujících přímé nasazení PLC ve výuce. Tím bylo zahájeno vytváření různých výukových laboratorních úloh. První z těchto úloh byla již navržena a realizována a byl k ní vytvořen detailní laboratorní návod, který je možno přímo použít ve výuce. V budoucnu mohou být podobným způsobem vytvořeny další návody. Očekává se, že především úlohy se dvěma krokovými motory budou pro studenty zábavné a inspirativní.

Reference

- [1] Mikro PLC [online]. Česká Republika [cit. 2019-12-02]. Dostupné z: <https://eluc.kr-olomoucky.cz/verejne/lekce/973>
- [2] Micro810 Programmable Logic Controller Systems [online]. In: . Milwaukee: Allen-Bradley [cit. 2019-12-02]. Dostupné z: <https://ab.rockwellautomation.com/Programmable-Controllers/Micro810>
- [3] Programmable Logic Controllers (PLCs): Basics, Types Applications [online]. Electrical 4 U, 2019 [cit. 2019-11-30]. Dostupné z: <https://www.electrical4u.com/programmable-logic-controllers/>
- [4] What is a PLC System – Different Types of PLCs with Applications [online]. India: El-ProCus [cit. 2019-12-02]. Dostupné z: <https://www.elprocus.com/programmable-logic-controllers-and-types-of-plcs/>
- [5] HOW A PLC POWER SUPPLY WORKS [online]. Rotterdam: REALPARS [cit. 2019-12-02]. Dostupné z: https://realpars.com/plc_power_supply/
- [6] Programovatelné logické automaty [online]. Praha: Vysoká škola chemicko-technologická v Praze [cit. 2019-11-30]. Dostupné z: <http://uprt.vscht.cz/kminekm/mrt/F5/F5k53-PLC.htm>
- [7] PLC Hardware – Inside the PLC [online]. PLC Programming Automation Online, 2018 [cit. 2019-12-02]. Dostupné z: <https://www.plcademy.com/plc-hardware-components/>
- [8] A Beginner’s PLC Overview, Part 3 of 4: PLC Inputs and Outputs (I/O) [online]. My-PLCtraining, 2018 [cit. 2019-11-30]. Dostupné z: <https://www.automation.com/automation-news/article/a-beginners-plc-overview-part-3-of-4-plc-inputs-and-outputs-io>
- [9] CYKLUS PLC [online]. HAPESOFT [cit. 2019-12-02]. Dostupné z: <http://plc-automatizace.cz/knihovna/plc/plc-cyklus.htm>
- [10] What is PLC scan and types of Scan Cycles?. Engineerscommunity [online]. (01.08.2019), 2 [cit. 2022-05-11]. Dostupné z: <https://engineerscommunity.com/t/what-is-plc-scan-and-types-of-scan-cycles/8592>
- [11] IEC EN 61131-3 [online]. AUTOMA, 2005 [cit. 2019-12-02]. Dostupné z: https://automa.cz/cz/casopis-clanky/programovani-plc-podle-normy-iec-en-61131-3-vic-nez-jednotne-jazyky-2005_02_3_0310_1_237/
- [12] PLC learning series 8: Instruction List programming [online]. automationforum.co, 2019 [cit. 2019-12-02]. Dostupné z: <https://automationforum.co/plc-learning-series-8-instruction-list-programming/>
- [13] Esperanto programátorů PLC: programování podle normy IEC/EN 61131-3 (část 4) [online]. Ústí nad Labem: AUTOMA, 2011 [cit. 2019-11-30]. Dostupné z: <http://www.odbornecasopisy.cz/res/pdf/45413.pdf>
- [14] Programovací režimy pro PLC dle IEC 61131-3 (CoDeSys) [online]. Praha: HW server, 2011 [cit. 2019-11-30]. Dostupné z: <https://automatizace.hw.cz/programovaci-rezimy-pro-plc-dle-iec-61131-3-codesys>

-
- [15] Instrumentation and Control Engineering [online]. 2015 [cit. 2019-12-02]. Dostupné z: <http://instruzengg.blogspot.com/2015/09/plc-programming-examples.html>
- [16] WHAT ARE THE MOST POPULAR PLC PROGRAMMING LANGUAGES [online]. Rotterdam: REALPARS [cit. 2019-11-30]. Dostupné z: <https://realpars.com/plc-programming-languages/>
- [17] TwinCAT 3 Tutorial: Structured Text [online]. contactandcoil [cit. 2019-12-02]. Dostupné z: <http://www.contactandcoil.com/twincat-3-tutorial/structured-text/>
- [18] Ladder Logic Tutorial for Beginners [online]. PLC Programming Automation Online, 2017 [cit. 2019-12-02]. Dostupné z: <https://www.plcademy.com/ladder-logic-tutorial/>
- [19] Discover ideas about Logic Programming [online]. Pinterest [cit. 2019-12-02]. Dostupné z: <https://www.pinterest.com/pin/578782989582372742/>
- [20] PLC Programming Languages – IEC 61131-3 [online]. PLCGURUS.NET [cit. 2019-11-30]. Dostupné z: <https://www.plcgurus.net/plc-programming-languages/>
- [21] Sequential function chart [online]. San Francisco: wikipedia, 2019 [cit. 2019-12-02]. Dostupné z: https://it.wikipedia.org/wiki/Sequential_function_chart
- [22] Modicon M241 Logic Controller, Hardware Guide [online]. Francie: Schneider Electric, 2018 [cit. 2022-05-11]. Dostupné z: [https://download.schneider-electric.com/files?p_enDocType = User + guidep_FileName = EIO0000001456.07.pdfp_DOCRef = EIO0000001456](https://download.schneider-electric.com/files?p_enDocType=User+guidep_FileName=EIO0000001456.07.pdfp_DOCRef=EIO0000001456)
- [23] EcoStruxure Machine Expert [online]. Rueil-Malmaison, France [cit. 2022-05-06]. Dostupné z: [https://download.schneider-electric.com/files?p_enDocType = User + guidep_FileName = EIO0000002854.03.pdfp_DOCRef = EIO0000002854](https://download.schneider-electric.com/files?p_enDocType=User+guidep_FileName=EIO0000002854.03.pdfp_DOCRef=EIO0000002854)
- [24] JOCHEC, Jan. ŘÍZENÍ KROKOVÝCH MOTORŮ POMOCÍ PROGRAMOVATELNÉHO LOGICKÉHO AUTOMATU. Brno, 2021. BAKALÁŘSKÁ PRÁCE. VUT. Vedoucí práce Ing. Dalibor Červinka, Ph.D. Dostupné také z: [https://www.vut.cz/www_base/zavrace_soubor_verejne.php?file_id = 227172](https://www.vut.cz/www_base/zavrace_soubor_verejne.php?file_id=227172)
- [25] Diagnostika armatur na ETE metodou momentových křivek [online]. Praha, 2019 [cit. 2021-12-17]. Dostupné z: [https://dspace.cvut.cz/bitstream/handle/10467/83643/F2-BP-2019-Novotny-Lukas-Novotny_Bakalarska%20prace.pdf?sequence = -1isAllowed = y.BakalářskáPrace.ÈVUT](https://dspace.cvut.cz/bitstream/handle/10467/83643/F2-BP-2019-Novotny-Lukas-Novotny_Bakalarska%20prace.pdf?sequence=-1isAllowed=y.BakalářskáPrace.ÈVUT).

Příloha A - Schéma zapojení rozšiřující desky



- H1 MountingHole
- H2 MountingHole
- H3 MountingHole
- H4 MountingHole

- 1 B2
- 2 B1
- 3 A2
- 4 A1
- 5 B2
- 6 B1
- 7 A1
- 8 A2