

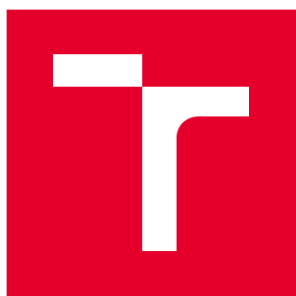
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2021

Jan Malý



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

VYUŽITÍ PROGRAMOVATELNÉHO LOGICKÉHO AUTOMATU VE VÝUCE ELEKTRICKÝCH POHONŮ

USE OF A PROGRAMMABLE LOGIC CONTROLLER IN TEACHING PROCESS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jan Malý

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Dalibor Červinka, Ph.D.

BRNO 2021

Bakalářská práce

bakalářský studijní program **Silnoproudá elektrotechnika a elektroenergetika**

Ústav výkonové elektrotechniky a elektroniky

Student: Jan Malý

ID: 203504

Ročník: 3

Akademický rok: 2020/21

NÁZEV TÉMATU:

Využití programovatelného logického automatu ve výuce elektrických pohonů

POKYNY PRO VYPRACOVÁNÍ:

1. Provedte přehled možností programování PLC v programovém prostředí SoMachine.
2. Realizujte demonstrační přípravek s motorkem, koncovými spínači a tlačítky, navržený v rámci SP.
3. Navrhněte a vytvořte s tímto přípravkem laboratorní úlohu s návodem na vytvoření ovládacího programu.

DOPORUČENÁ LITERATURA:

- [1] Dokumentace k PLC Modicon M241 a programovacímu prostředí EcoStruxure Machine Expert.
- [2] Koláčný J., Elektrické mikropohony, Vydání 1., skriptum FEKT VUT Brno, 2009.
- [3] Skalický J., Navrhování elektrických pohonů, Vydání 1., skriptum FEKT VUT Brno, 2002.

Termín zadání: 8.2.2021

Termín odevzdání: 27.5.2021

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

doc. 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.

Abstrakt

Bakalářská práce je rozdělena do čtyř částí. První část stručně uvádí práci popisem PLC. Druhá část se věnuje jazykům používaných pro programování PLC. Tyto jazyky jsou rozděleny na grafické a textové. Do grafických jazyků patří jazyk příčkového diagramu a jazyk funkčního blokového schématu. Do textových jazyků se pak řadí jazyk strukturovaného textu a jazyk mnemokódů. Jazyky jsou popsány ve svých podkapitolách. Třetí část se věnuje popisu laboratorní úlohy, která se věnuje zapínacím proudům. Dělí se dvě podkapitoly, jedna se věnuje postupu měření laboratorní úlohy, druhá se věnuje vytváření programu pro PLC. V poslední kapitole bude stručně popsán výrobek vyrobený pro účely laboratorní úlohy.

Klíčová slova

Laboratorní úloha, PLC, programování PLC, DC motor, elektrické pohony

Abstract

Bachelor thesis is divided into four parts. First briefly introduces the thesis by discussing PLC. Second part concentrates on programming languages used for programming a PLC. These are divided into graphical and text languages. Graphical languages consist of ladder diagram and function block diagram. Text languages are Structured text and instruction list. Each of these languages are described in subchapters. Third part deals with suggested laboratory task. It is divided into two subchapters, first concentrates on executing the measurement, second consists of suggestions for creating code for PLC. Last part briefly overviews creation of a product for the laboratory task.

Keywords

Laboratory task, PLC, programming PLC, DC motor, electric drives

Bibliografická citace

MALÝ, Jan. *Využití programovatelného logického automatu ve výuce elektrických pohonů*. Brno, 2021. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/131249>. Semestrální 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: *Jan Malý*

VUT ID studenta: *203504*

Typ práce: *Bakalářská práce*

Akademický rok: *2020/21*

Téma závěrečné práce: *Využití programovatelného logického automatu ve výuce elektrických pohonů*

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: 27. května 2021

podpis autora

Poděkování

Chtěl bych poděkovat svému vedoucímu bakalářské práce, Ing. Daliboru Červinkovi Ph.D. za užitečné rady při psaní bakalářské práce.

V Brně dne: 27. května 2021

podpis autora

Obsah

SEZNAM SYMBOLŮ A ZKRATEK	11
SEZNAM OBRÁZKŮ	12
SEZNAM TABULEK.....	13
ÚVOD	14
1. PROGRAMOVATELNÝ LOGICKÝ AUTOMAT	15
1.1 PROGRAMOVATELNÝ AUTOMAT.....	15
1.1.1 <i>Kompaktní PLC</i>	17
1.1.2 <i>Modulární PLC</i>	17
1.2 SCHNEIDER ELECTRIC – MODICON M241	18
2. PROGRAMOVÁNÍ LOGICKÝCH AUTOMATŮ.....	20
2.1 GRAFICKÉ JAZYKY	21
2.1.1 <i>Jazyk příčkového diagramu</i>	21
2.1.2 <i>Jazyk funkčního blokového schématu</i>	23
2.2 TEXTOVÉ JAZYKY.....	24
2.2.1 <i>Jazyk strukturovaného textu</i>	24
2.2.2 <i>Jazyk seznamu instrukcí</i>	26
2.3 ECOSTRUCTURE MACHINE EXPERT	28
2.4 SOMACHINE.....	31
3. LABORATORNÍ ÚLOHA	33
3.1 TEORETICKÁ ČÁST LABORATORNÍ ÚLOHY	33
3.1.1 <i>Zapínací proud stejnosměrného motoru</i>	33
3.1.2 <i>Omezení zapínacího proudu</i>	35
3.2 PRAKTICKÁ ČÁST	35
3.2.1 <i>Postup laboratorní úlohy:</i>	35
3.2.2 <i>Vytvoření programu pro laboratorní úlohu</i>	37
4. VÝROBA LABORATORNÍHO PŘÍPRAVKU	40
ZÁVĚR	42
LITERATURA.....	43

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

FEKT	Fakulta elektrotechniky a komunikačních technologií
VUT	Vysoké učení technické v Brně
DPS	Deska plošných spojů
PLC	Programovatelný logický automat

Symboly:

U	napětí	[V]
I	proud	[A]
L_a	<i>indukčnost obvodu kotvy</i>	[H]
B	součinitel viskózního tření	[-]
J	moment setrvačnosti	[kg m ²]
t	čas	[s]
ω	úhlová rychlost	[rad/s]
M	moment síly	[Nm]
n	otáčky	[1/min]
$c\varphi$	konstanta motoru	[Nm/A]

SEZNAM OBRÁZKŮ

Obrázek 1: PLC Tecomat TC700 [19].....	16
Obrázek 2: Vnitřní struktura PLC [8].....	17
Obrázek 3: Ilustrační obrázek kompaktního PLC [17].....	17
Obrázek 4: Ilustrační obrázek modulárního PLC [17].....	18
Obrázek 5: Modicon M241.....	18
Obrázek 6: Pohled na Modicon M241 zepředu [12].....	19
Obrázek 7: Jazyky pro programování řídicích jednotek specifikované v normě IEC EN 61131-3 [7].....	20
Obrázek 8: Symboly jazyka příčkového diagramu [5].....	22
Obrázek 9: Operátory jazyka mnemokódů [5].....	27
Obrázek 10: Prostředí EcoStructure Machine Expert při spuštění.....	28
Obrázek 11: Vytváření projektu.....	29
Obrázek 12: Prostředí EcoStructure Machine Expert.....	30
Obrázek 13: Připojení PLC před vytvořením projektu.....	31
Obrázek 14: Úvodní obrazovka prostředí SoMachine.....	31
Obrázek 15: Okno po připojení PLC k počítači.....	32
Obrázek 16: Náhradní schéma stejnosměrného motoru [14].....	33
Obrázek 17: Průběhy proudu a otáček při zapínání motoru.....	34
Obrázek 18: Zapojení laboratorní úlohy.....	36
Obrázek 19: Možné zapojení laboratorní úlohy.....	36
Obrázek 20: Vytváření POU v prostředí SoMachine.....	38
Obrázek 21: Seznam výstupů PLC s jejich adresami.....	39
Obrázek 22: Seznam vstupů PLC s jejich adresami.....	39
Obrázek 23: Relé Finder 40.61.....	40
Obrázek 24: Dokončený přípravek pro laboratorní úlohu.....	41

SEZNAM TABULEK

Tabulka 1: Tabulka datových typů v jazyce strukturovaného textu [1]	25
Tabulka 2: Tabulka pro jazyk funkcí strukturovaného textu [1]	26

ÚVOD

Téma této bakalářské práce je *Využití programovatelného logického automatu ve výuce elektrický pohonů*. Důraz bude kladen na využití programovatelného logického automatu ve výuce na vysoké škole, zejména pak pro studenty předmětu Elektrické pohony.

Teoretická část bakalářské práce bude uvedena textem a obrázky o programovatelných logických automatech. Další kapitola bude věnována programování PLC. Bude popsáno několik programovacích jazyků, které budou rozděleny na grafické a textové. Kapitola o programování bude zakončena popisem dvou programovacích prostředí pro PLC od firmy Schneider Electric, kterými jsou SoMachine a EcoStructure Machine Expert.

Další kapitola bude věnována samotné laboratorní úloze. Nejdříve se bude krátce zabývat problematikou, kterou má tato úloha ukázat. Dále se pak bude věnovat postupu měření úlohy. Součástí tohoto postupu bude i podkapitola věnována vytvoření programu pro tuto úlohu. Vytvoření programu je nedílnou součástí této úlohy.

Poslední kapitola se bude krátce věnovat přípravku, který byl pro tuto úlohu vyroben. Kapitola bude obsahovat stručný popis výroby.

Cílem této bakalářské práce je vytvořit úlohu, která bude sloužit studentům předmětu Elektrické pohony a bude využívat PLC.

1. PROGRAMOVATELNÝ LOGICKÝ AUTOMAT

Následující text se bude věnovat programovatelným logickým a automatům a informacím s nimi souvisejícími.

1.1 Programovatelný automat

Programovatelný automat je řídicí systém, programovaný uživatelem, určený k řízení průmyslových a technologických procesů nebo strojů. Často bývají označovány různými zkratkami například „PLC“ (Programmable Logic Controller), PC (Programmable Controller) nebo českou zkratkou PA (Programovatelný automat). [2, 4]

K lepšímu seznámením se s PLC využijeme citace Ladislava Šmejkal a Marie Martináskové, kteří ve své práci *PLC a automatizace* popisují PLC následovně: „*Programovatelný automat je uživatelný řídicí systém přizpůsobený pro řízení průmyslových a technologických procesů, mnohdy specializovaný na úlohy převážně logického typu (obzvláště u nejmenších systémů). Nejčastěji se označuje zkratkou PLC (Programmable Logic Controller)*“. Kromě označení PLC je rovněž odkazováno na německou literaturu, kde se můžeme setkat s označením Speicherprogrammierbare Steuerung (SPS). Dalším pojmenováním, se kterým se můžeme setkat je pojmenování „Programmable automation controller“ (PAC). Toto pojmenování bývá nově více používáno, neboť lépe vystihuje vlastnosti moderního programovatelného automatu. [2, 4, 20]

Časopis AUTOMA, který se zabývá automatizační technikou, popisuje ve svém článku *Programovatelné automaty – PLC, nebo PAC?* jehož autory jsou Ladislav Šmejkal a Luboš Urban současné vlastnosti PLC následovně: „*Pro současné PLC jsou typické: velký výpočetní výkon, značný rozsah paměti, schopnost komunikace a dostupnost vyspělých algoritmů založených na nejnovějších teoretických poznatcích. I s jednoduchými a levnými kompaktními PLC lze mnohdy realizovat poměrně složité a náročné způsoby řízení, regulace a technické diagnostiky, které si zaslouží módní označení smart (chytré) či intelligent.*“ [20]

Na následujícím obrázku můžeme vidět ukázkou jednoho ze starších PLC, a to



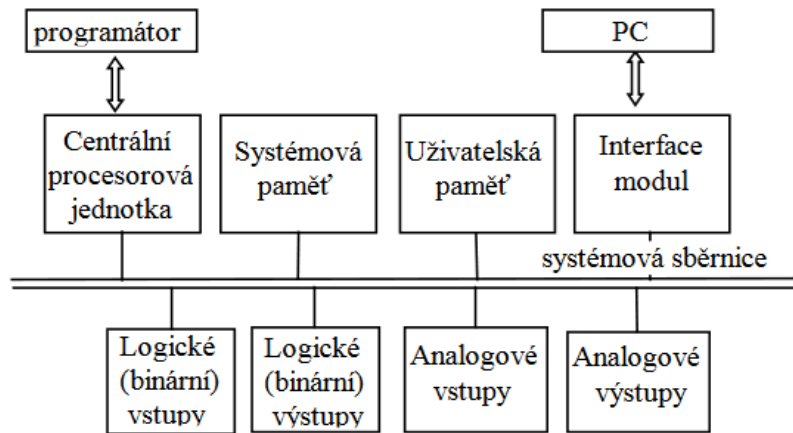
Obrázek 1: PLC Tecomat TC700 [19]

Tecomat TC 700 vyroben českou společností Teco a.s. [19]

Tradiční PLC v sobě obsahuje centrální procesovou jednotku, která je „inteligencí“ PLC. Tato jednotka zajišťuje chod instrukcí a systémových služeb, rovněž se stará komunikaci uvnitř PLC. Dalším komponentem obsaženým v PLC je systémová a uživatelská paměť. Zařízení PLC rovněž disponuje několika vstupy (digitální a binární/logické) a výstupy. Co se týče samotného programování PLC, důležitou součástí jsou programovací jazyky, které umožňují realizaci logických funkcí. O těchto jazycích se budeme dále bavit v kapitole 2 a jejích podkapitolách, přičemž teď si je pouze vyjmenujeme. Programovací jazyky, které slouží k realizaci logických funkcí jsou následující: [8]

- Jazyk strukturovaného textu
- Jazyk kontaktních (reléových) schémat
- Jazyk logických schémat
- Jazyk mnemokódů

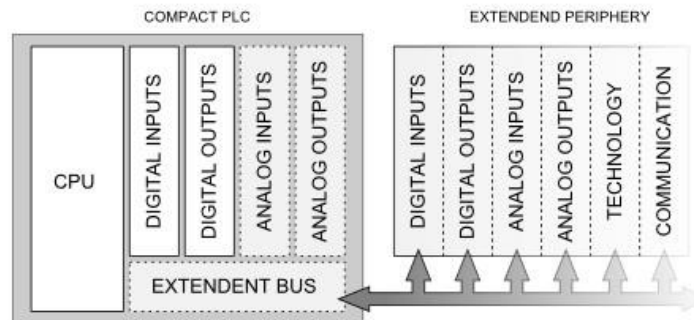
Na následujícím obrázku můžeme vidět vnitřní strukturu PLC popsanou Ivanem Švarcem v *Základech automatizace*. [8]



Obrázek 2: Vnitřní struktura PLC [8]

1.1.1 Kompaktní PLC

Je provedení PLC, ve kterém jsou obsaženy všechny nutné součásti. Výhodou je rychlý přístup k perifériím a cena. Kompaktní PLC má také omezený počet vstupů a výstupů, jejichž poměr je dán výrobcem. [17, 18]

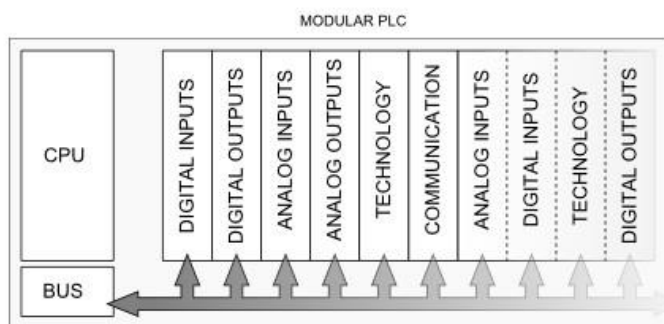


Obrázek 3: Ilustrační obrázek kompaktního PLC [17]

1.1.2 Modulární PLC

Modulární PLC, jak již název napovídá, je PLC složené z různých modulů, které určují např. počet vstupů a výstupů nebo. Tato modulárnost umožňuje poskládat PLC

z modulů, které jsou potřebné pro daný úkol. Zároveň má větší kapacitu pro program a data, jelikož se předpokládá zpracování většího množství signálů a dat. [17, 18]



Obrázek 4: Ilustrační obrázek modulárního PLC [17]

1.2 Schneider Electric – Modicon M241

Firma Schneider Electric, která se zabývá nejpokročilejšími energetickými technologiemi a jejich kombinací, automatizací v reálném čase, softwarem apod. nabízí řadu kontrolérů, jako je například logický kontrolér PLC Modicon M241, který můžeme vidět na následujícím obrázku. [12]



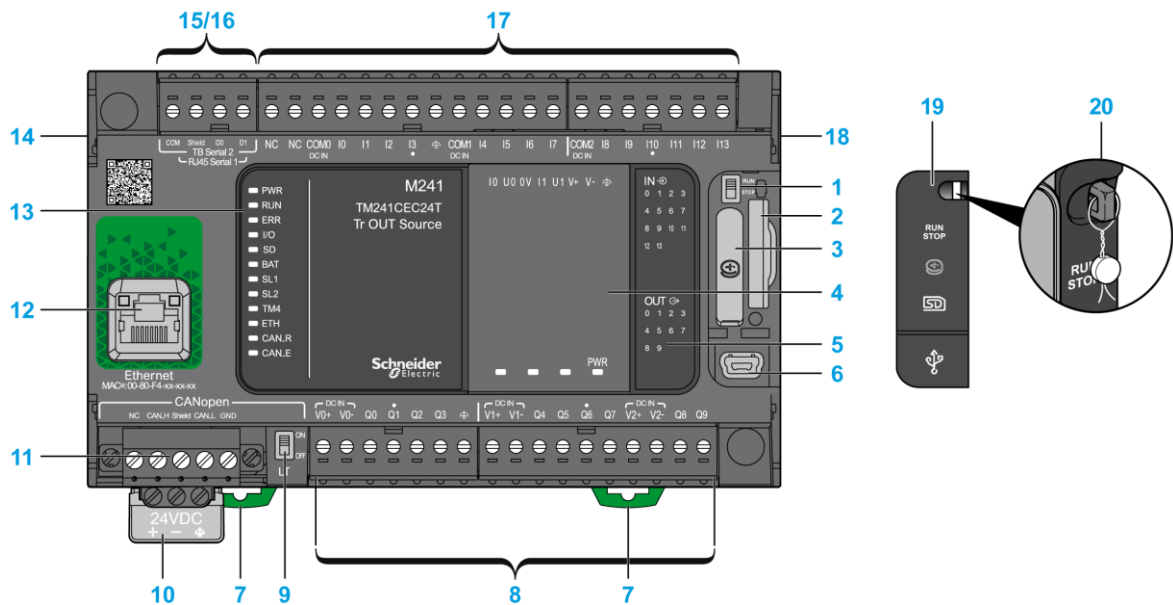
Obrázek 5: Modicon M241

Logický kontrolér PLC Modicon M241 (zobrazen na obrázku výše) je napájen napětím 24 V DC, obsahuje 14 diskretních vstupů, z toho je 8 rychlých vstupů podle normy IEC 61131-2. Výstupy tohoto logického kontroloru jsou rozděleny na 10 tranzistorových a 4 rychlé. Schneider Electric uvádí hodnoty diskretního napěťového vstupu 24 V, diskretní proudový vstup má pak 5 mA pro normální vstup a 10,7 mA pro rychlý vstup. Dále pak hodnoty diskretního napěťového výstupu uvádí firma u tohoto

zařízení 24 V DC pro tranzistorový výstup, proudový výstup s hodnotami 0,5 A pro tranzistorový výstup a 0,1 A pro rychlý výstup. Meze výstupního napětí se rovná 30 V DC. [12]

PLC Modicon M241 disponuje několika ochranami. Tento logický kontrolér je chráněn proti zkratu, proti přetížení a zkratu s automatickým resetem, proti přepólování pro rychlý výstup. Disponuje paměťovou kapacitou 64 MB RAM. Kromě paměti disponuje tento přístroj možností zálohování dat, a to vyhrazenými 128MB paměti vestavěnými v zařízení. Přístroj v sobě obsahuje řadu integrovaných typů připojení, jako jsou CANopen port, či Ethernet port, rovněž port USB a 2 sériové porty. [12]

Následující obrázky slouží k popsání složení logického kontroléru PLC Modicon M241. [12]



Obrázek 6: Pohled na Modicon M241 zepředu [12]

- | | |
|--------------------------------------|----------------------|
| 1. Tlačítko zapnout/vypnout | 10. 24 V DC zdroj |
| 2. Slot pro paměťovou kartu | 11. CANopen port |
| 3. Příhrádka pro baterii | 12. Ethernet port |
| 4. Cartridge | 13. Led označení |
| 5. LED označení I/O | 14. TM4 přípojka |
| 6. USB mini-B port | 15. Sériový port 1 |
| 7. Montážní držák pro 35 mm DIN rail | 16. Sériový port 2 |
| 8. Vestavěné výstupy | 17. Vestavěné vstupy |
| 9. Tlačítko CANopen zapnout/vypnout | 18. TM3/TM2 přípojka |
| | 19. Zajišťovací hák |

2. PROGRAMOVÁNÍ LOGICKÝCH AUTOMATŮ

V této kapitole se budeme věnovat programování logických automatů. Toto programování je standardizováno normou IEC 61131-3. Tato norma se skládá z několika částí, přičemž nás bude zajímat část třetí, která se věnuje programovacím jazykům. Důležité je zmínit, že norma IEC 61131-3 definuje několik základních pojmů, které je nutné vysvětlit předtím, než se začneme věnovat konkrétním jazykům. [1]

Mezi první pojmy patří programovací organizační jednotka. Programová organizační jednotka, anglicky „Program Organisation Unit“ (POU), je podle normy IEC 61131-3 základním pojmem při programování PLC a vyjadřuje nejmenší nezávislou část programu. Jako nejjednodušší programovou organizační jednotku považujeme „Funkci“. Její nejdůležitější vlastností je, že pokud zadáme stejné vstupní parametry, následný výsledek bude vždy stejný a bude odpovídat těmto vstupním parametrům. V případě, že se změní vstupní parametry, změní se i výsledek. Mimo funkci je nutné zmínit rovněž další programovou organizační jednotku, funkční blok. Na rozdíl od funkce je funkční blok schopen ukládat data do paměti. Tato data uložená v paměti pak mohou ovlivňovat výsledky funkčního bloku. Kromě možnosti ukládat data do paměti se v případě funkčního bloku nachází další vlastnost, a to možnost mít více než jeden výsledek bloku. [1]

Programovací jazyky, jak grafické, tak textové jsou specifikované v normě IEC EN 61131-3. „Jazyky podle normy IEC EN 61131-3 a jejich společné prvky jsou podstatně systematičtější a obecněji definovány, než tomu je u dosavadních jazyků. Navíc všechny navzájem sdílejí společné prvky. Přejechy mezi různými typy jazyků jsou tak snazší.“ [1]

Kodifikované označení v angličtině		Označení obvyklé v němčině		Vhodný název v češtině
zkratka	název	zkratka	název	
LD	Ladder Diagram	KOP	Kontaktplan	reléové schéma
FBD	Function Block Diagram	FUP	Funktionplan	jazyk funkčních bloků
IL	Instruction List	AWL	Anweisungsliste	jazyk mnemokódů
ST	Structured Text	ST	Strukturierter Text	strukturovaný text
SFC	Sequential Function Chart	AS	Ablaufsprache	jazyk sekvenčního programování

Norma IEC EN 61131-3

Obrázek 7: Jazyky pro programování řídicích jednotek specifikované v normě IEC EN 61131-3 [7]

Podle této normy jsou na následujícím obrázku rozděleny všechny jazyky (jak grafické, tak textové), se kterými je možno pracovat. Výjimku tvoří jazyk sekvenčního programování SFC. „*Je to prostředek pro strukturovaný popis sekvenčních úloh, nadřazený ostatním jazykům. Někdy je mylně nazýván jako jazyk Grafcet, který je sice velmi podobný SFC (SFC je od něj odvozen), ale je obecnější a je popsán vlastní normou.*“ [3]

2.1 Grafické jazyky

V následující kapitole se budeme věnovat grafickým jazykům. Grafické jazyky jsou spolu s textovými jazyky nedílnou součástí programování PLC. V nadcházejících podkapitolách se zmíním o jednotlivých grafických jazycích, které budou následně podrobněji rozepsány. Konkrétně se bude jednat o *Jazyk příčkového diagramu* a *jazyk funkčního blokového diagramu*. Každý z těchto jazyků bude krátce popsán. Popisy budou rovněž doprovázet obrázky, které se budou u každého z jazyků nacházet.

2.1.1 Jazyk příčkového diagramu

Prvním z grafických jazyků, který si představíme, je *Jazyk příručkového diagramu*. Tento jazyk slouží, stejně jako ostatní grafické či textové jazyky, které zmíním v dalších podkapitolách, pro programování PLC. Diagram lze zobrazit skládáním dvou svislých kolejnic. Mezi těmito kolejnicemi se nachází horizontální příčky, na kterých se nachází tzv. *symboly*. Program systematicky prochází jednotlivé příčky a symboly, které se na nich nacházejí. Každý symbol v programu je zkontrolován, případně je provedena operace závislá na konkrétní funkci daného symbolu. Kombinací těchto symbolů a změnou jejich zapojení se vytváří logika programu. [2, 4, 5, 11]

Funkce	Anglický název	Symbol
spínací kontakt	<i>open contact</i>	- -
rozpínací kontakt	<i>closed contact</i>	- / -
cívka s přímou funkcí	<i>coil</i>	-()-
cívka s negovanou funkcí	<i>negated coil</i>	-(/)-
cívka s funkcí <i>set</i>	<i>set coil</i>	-(S)-
cívka s funkcí <i>reset</i>	<i>reset coil</i>	-(R)-

Obrázek 8: Symboly jazyka příčkového diagramu [5]

Na obrázku 2 můžeme vidět několik symbolů typických pro jazyk příčkového diagramu, konkrétně *Spínací kontakt*, *rozpínací kontakt*, *cívku s přímou funkcí*, *cívku s negovanou funkcí*, *cívku s funkcí set* a *cívku s funkcí reset*. Každý z těchto symbolů má svoji funkci. Kromě své funkce obsahuje každý symbol také adresu bitu, se kterým pracuje. [12]

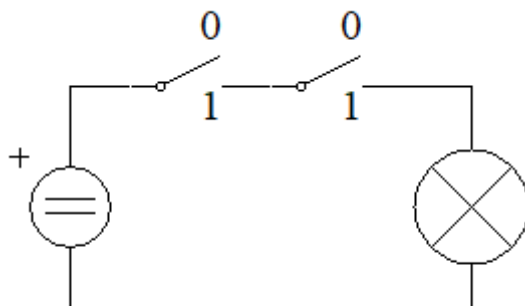
Prvními dvěma symboly jsou *spínací* a *rozpínací kontakt*. Tyto symboly pracují se vstupními bity. Jejich funkce je velmi podobná. Oba tyto symboly kontrolují svůj vstup a na jeho základě ovlivňují svůj výsledek. Výsledek spínacího kontaktu je stejná hodnota, jako je hodnota vstupu. Naopak výsledná hodnota rozpínacího kontaktu je opačná vůči hodnotě vstupní. [12]

Dalšími symboly jsou různé typy cívek. Všechny pracují s výstupy, ale každá pracuje jiným způsobem. *Cívka s přímou funkcí* mění svoji hodnotu podle vstupu vždy tak, aby hodnoty byly stejné. Pokud je na vstupu 1, výstup bude také 1. *Cívka s negovanou funkcí* opět mění svoji hodnotu pokaždé, když se změní hodnota vstupu. Výstup bude mít vždy opačnou hodnotu, než je hodnota vstupní. Pro příklad, pokud bude vstup 1, výstup bude roven 0. Posledními, ne však méně důležitými symboly, jsou *cívky s funkcemi set* a *reset*. Jejich funkce byla popsána v článku *Esperanto programátorů PLC: Programování podle normy IEC/EN 61131-3 (část 10)*, jehož autorem je Ladislav Šmejkal. Tyto symboly popsal následovně: „*Obě mění stav své výstupní proměnné, jen jestliže k nim zleva přichází pravdivý signál – sepnuto. V tom případě cívka Set nastavuje svůj výstup na hodnotu pravda (TRUE, sepnuto), zatímco cívka Reset svůj výstup nastavuje na hodnotu nepravda (FALSE, rozepnuto). V opačném případě (pokud k cívkám přichází nepravdivý signál – rozepnuto) nemění cívky stav své proměnné – pamatují si naposledy nastavenou hodnotu.*“ [6, 12]

2.1.2 Jazyk funkčního blokového schématu

Druhý z grafických jazyků se nazývá *Jazyk funkčního blokového schématu*. Jazyk funkčního blokového schématu je grafický jazyk používající logické operátory AND, NOT, OR a funkce z nich složené. Těmto funkcím je v programu přiřazen blok. Kombinací těchto funkcí nebo vytvářením vlastních, lze vytvářet další bloky, které jsou následně propojovány. Pro základní používání tohoto jazyka je nutné znát logické funkce zmíněné výše, a proto je v krátkosti vysvětlíme. [5]

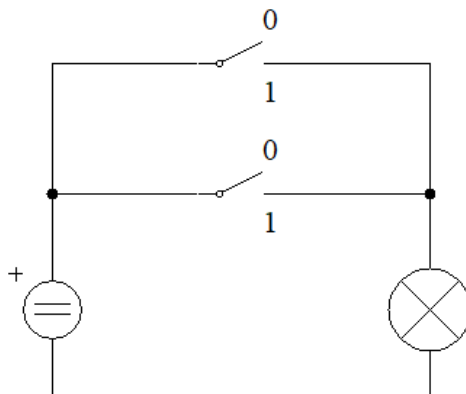
Prvním logická funkce je funkce AND. AND je funkce, jejímž výsledkem může být 1 nebo 0, respektive „true“ nebo „false“. Výsledek funkce závisí na vstupech. V případě, že jsou všechny stupy rovny 1, celá funkce je vyhodnocena výsledkem 1. V případě, že je alespoň jeden vstup roven 0, následný výsledek je rovněž 0. [10]



Obrázek 3: Zobrazení logické funkce AND pomocí elektrického obvodu

Jako další logickou funkci si vysvětlíme funkci NOT. NOT je logická funkce, která má za výsledek opak svého vstupu, to znamená, že pokud je vstupem logická 1, výsledek bude 0. [10]

Třetí logickou funkcí, kterou si vysvětlíme je funkce OR. OR je opět logická funkce, jejímž výsledkem, stejně jako u funkce AND, může být 1 nebo 0. V případě funkce OR



Obrázek 4: Zobrazení logické funkce OR pomocí elektrického obvodu

je mezi vstupy nutný pouze jeden vstup s logickou 1, aby byl výstup roven 1. Aby byl výsledek roven 0, musí být všechny vstupy 0. [10]

Kombinací dříve zmíněných logických funkcí mohou vznikat další funkce, například NAND, což je funkce složená z funkcí NOT a AND. Výsledek je vytvořen postupným aplikováním zmíněných funkcí. [10]

V neposlední řadě je nutno zmínit logickou funkci XOR. Tento logický operátor rovněž operuje s výsledky 1 a 0, přičemž pokud jsou oba vstupy stejné (tzn. Oba 1 nebo oba 0), výsledkem je vždy 0. [10]

2.2 Textové jazyky

Textové jazyky tvoří druhé křídlo programovacích jazyků PLC. Rovněž jsou to jazyky, které jsou specifikované v normě IEC EN 61131-3. Stejně tak, jak tomu bylo i v případě Grafických jazyků, popíšeme si podrobněji jazyky, které se do této kategorie řadí, a sice *Strukturovaný text* a *Jazyk seznamu instrukcí*. [1]

2.2.1 Jazyk strukturovaného textu

Strukturovaný text je, jak je z názvu zřejmé, textový programovací jazyk pro PLC. Mimo název *strukturovaný text* se můžeme rovněž setkat s anglickým názvem *Structured text*, či zkratkou *ST*. Tento jazyk je svojí strukturou velmi podobný jazykům jako je například C nebo PHP. Ladislav Šmejkal ve svém článku *Esperanto programátorů PLC: programování podle normy IEC/EN 61131-3 (část 4)* uvádí několik výhod strukturovaného jazyka. Mezi výhody tohoto jazyka uvedl například, že jazyk strukturovaného textu je výhodný pro řešení komplikovaných úloh logických, numerických i smíšených. Rovněž jako další výhody uvádí následující: „*Je zcela obecný, má bohatý aparát příkazů pro úlohy různých typů. Při systematickém používání a při užití jeho logických a aritmetických výrazů a struktur dat je jazyk ST velmi efektivní.*“ [1, 3, 4, 8]

Jak již bylo zmíněno výše, jazyk strukturovaného textu je velmi podobný jazykům jako C nebo PHP. Tuto podobnost je možno pozorovat jak při vytváření různých proměnných, tak při psaní logiky programu, která může obsahovat různé větvení a funkce pro opakování. [1, 8]

Klíčové slovo	Anglicky	Datový typ	Bitů	Rozsah hodnot
BOOL	Boolean	Boolovské číslo	1	0,1
SINT	Short integer	Krátké celé číslo	8	-128 až 127
INT	Integer	Celé číslo	16	-32 768 až 32 767
DINT	Double integer	Celé číslo, dvojnásobná délka	32	-2 147 483 648 až 2 147 483 647
USINT	Unsigned short integer	Krátké celé číslo bez znaménka	8	0 až 255
UINT	Unsigned integer	Celé číslo bez znaménka	16	0 až 65 535
UDINT	Unsigned double integer	Celé číslo bez znaménka, dvojnásobná délka	32	0 až 4 294 967 295
REAL	Real (single precision)	Číslo v pohyblivé řádové čárce (jednoduchá přesnost)	32	±2,9E-39 až ±3,4E+38 Podle IEC 559
LREAL	long real (double precision)	Číslo v pohyblivé řádové čárce (dvojnásobná přesnost)	64	Podle IEC 559
TIME	Duration	Trvání času	24d 20:31:23.647	
DATE	Date (only)	Datum	Od 1.1.1970 00:00:00	
TIME_OF_DAY nebo TOD	Time of day (only)	Denní čas	24d 20:31:23.647	
DATE_AND_TIME nebo DT	Date and time of day	"Absolutní čas"	Od 1.1.1970 00:00:00	
STRING	String	Řetězec	Max. 255 znaků	
BYTE	Byte (bit string of 8 bits)	Sekvence 8 bitů	8	Není deklarován rozsah
WORD	Word (bit string of 16 bits)	Sekvence 16 bitů	16	Není deklarován rozsah
DWORD	Double word (bit string of 32 bits)	Sekvence 32 bitů	32	Není deklarován rozsah

Tabulka 1: Tabulka datových typů v jazyce strukturovaného textu [1]

V tabulce 1 jsou datové typy používané pro vytváření proměnných v jazyce strukturovaného textu při programování PLC. Můžeme si všimnout některých podobností s datovými typy jazyka C. Například datový typ „INT“ je téměř totožný, možný rozdíl je počet bitů, který může být v jazyce C 16 nebo 32, dle systému. Další podobností může být možnost odebrat číslům jejich znaménko. V jazyce C je pro tuto možnost vyhrazeno

klíčové slovo „Unsigned“. V jazyce strukturovaného textu, jak je možno vidět v tabulce 1, se místo klíčového slova „Unsigned“ objevuje pouze „u“, ale funkce je stejná. [8, 9]

Jak je již zmíněno výše, Jazyk strukturovaného textu je podobný jazykům, jako C, i při psaní logiky. Stejně jako u jazyka C, je možno použít různá větvení a funkce pro opakování. V tabulce 2 je možno vidět tyto funkce. Většina těchto funkcí je stejná jako jazyce C, pouze syntax je jiný. [8]

Příkaz	Popis	Příklad	Poznámka
:=	Přiřazení	A := 22	Přiřazení hodnoty vypočtené na pravé straně do identifikátoru na levé straně
	Volání funkčního bloku	InstanceFB(par1 := 10, par2 := 20);	Volání funkčního bloku s předáváním parametrů
IF	Příkaz výběru	IF A > 0 THEN B := 100; ELSE B := 0; END_IF;	Výběr alternativy podmíněné výrazem BOOL
CASE	Příkaz výběru	CASE kod OF 1 : A := 11; 2 : A := 22; ELSE A:= 99; END_CASE	Výběr bloku příkazů podmíněný hodnotou "kod"
FOR	Iterační příkaz smyčka FOR	FOR i := 0 TO 10 BY 2 DO j := j + i; END_FOR;	Vícenásobná smyčka bloku příkazů s počáteční a koncovou podmínkou a hodnotou inkrementu
WHILE	Iterační příkaz smyčka WHILE	WHILE i > 0 DO n := n * 2; END_WHILE;	Vícenásobná smyčka bloku příkazů s podmínkou na ukončení smyčky na začátku
REPEAT	Iterační příkaz smyčka REPEAT	REPEAT k := k + i; UNTIL i < 20; END_REPEAT;	Vícenásobná smyčka bloku příkazů s podmínkou na ukončení smyčky na konci
EXIT	Ukončení smyčky	EXIT;	Předčasné ukončení iteračního příkazu
RETURN	Návrat	RETURN;	Opuštění právě vykonávané POU a návrat do volající POU
;	prázdný příkaz	::	

Tabulka 2: Tabulka pro jazyk funkcí strukturovaného textu [1]

2.2.2 Jazyk seznamu instrukcí

Jazyk mnemokódů, též známý jako Instructions list, ve zkratce IL je „*obdobou assembleru u počítačů, a je také strojově orientován. To znamená, že každé instrukci PLC systému odpovídá stejně pojmenovaný příkaz jazyka.*“ Jazyk mnemokódů je často

používán, hlavně pak profesionálními programátory, kterým tento jazyk dovoluje nejlépe přizpůsobit úlohu možnostem PLC. [4]

Operace v jazyku mnemokódů/jazyku seznamu instrukcí se provádí následovně: „Požadovaná operace se provádí s poslední hodnotou výsledku (aktuálním stavem střádače) a s programovaným operandem. Výsledek je opět uložen ve střádači. Pro realizaci logických úloh má jazyk IL k dispozici mnemokódy instrukcí pro různé typy operací.“ [5]

výsledek: = výsledek OPERATOR operand

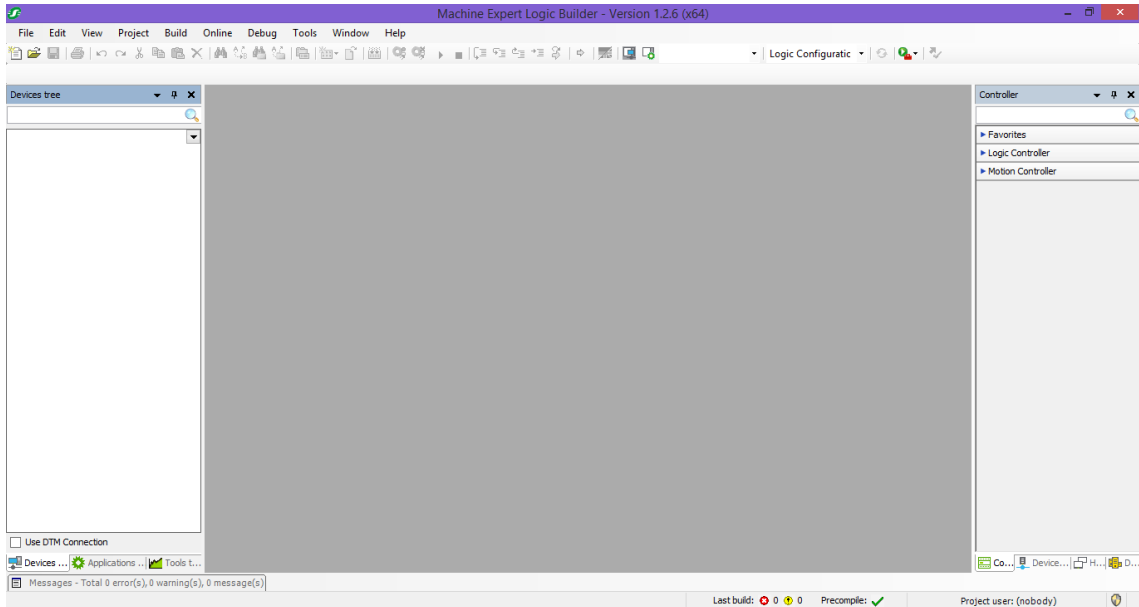
Na následujícím obrázku můžeme vidět operátory a modifikátory pro logické operace datového typu ANY_BIT, to znamená využití pro projekty typu BOOL, BYTE, WORD a DWORD. [5]

Operátor	Modifikátor	Popis funkce
LD	N	LD (<i>load</i> , <i>sejmi</i>) – nastaví výsledek podle aktuální hodnoty adresovaného operandu (<i>sejme</i> aktuální hodnotu adresovaného operandu do střádače)
AND	N, (booleovský součin AND
OR	N, (booleovský součet inkluzivní OR
XOR	N, (booleovský součet exkluzivní XOR
ST	N	ST (<i>store</i> , <i>ulož</i>) – uložení aktuálního výsledku do adresované proměnné
S	<i>set</i> (podmíněné nastavení)	nastaví programovanou proměnnou do log.1 (TRUE), jestliže je výsledek jedničkový, jinak nic
R	<i>reset</i> (podmíněné nulování)	nastaví programovanou proměnnou do log. 0 (FALSE), jestliže je výsledek jedničkový, jinak nic
)	vyhodnocení odložené (<i>deferred</i>) operace	uzavírací závorka výrazu, vyčíslení výrazu

Obrázek 9: Operátory jazyka mnemokódů [5]

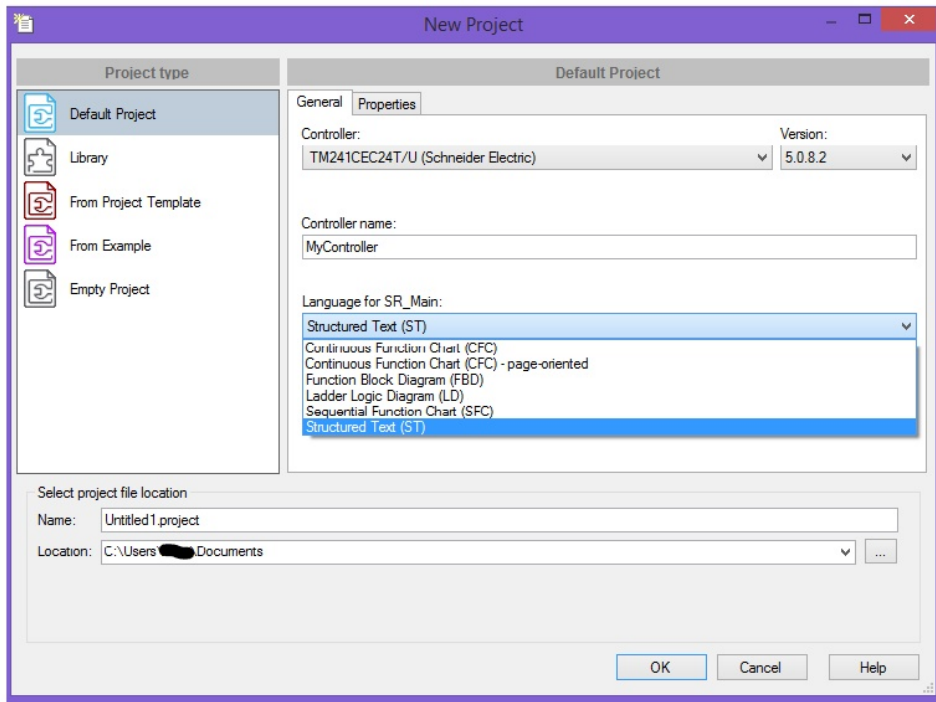
2.3 EcoStructure Machine Expert

Tato podkapitola se bude věnovat základnímu seznámení s programem Ecostructure Machine Expert, jako náhrady za SoMachine. Pro naše účely oba programy vykonávají stejnou funkci, avšak Machine Expert je novější.



Obrázek 10: Prostředí EcoStructure Machine Expert při spuštění

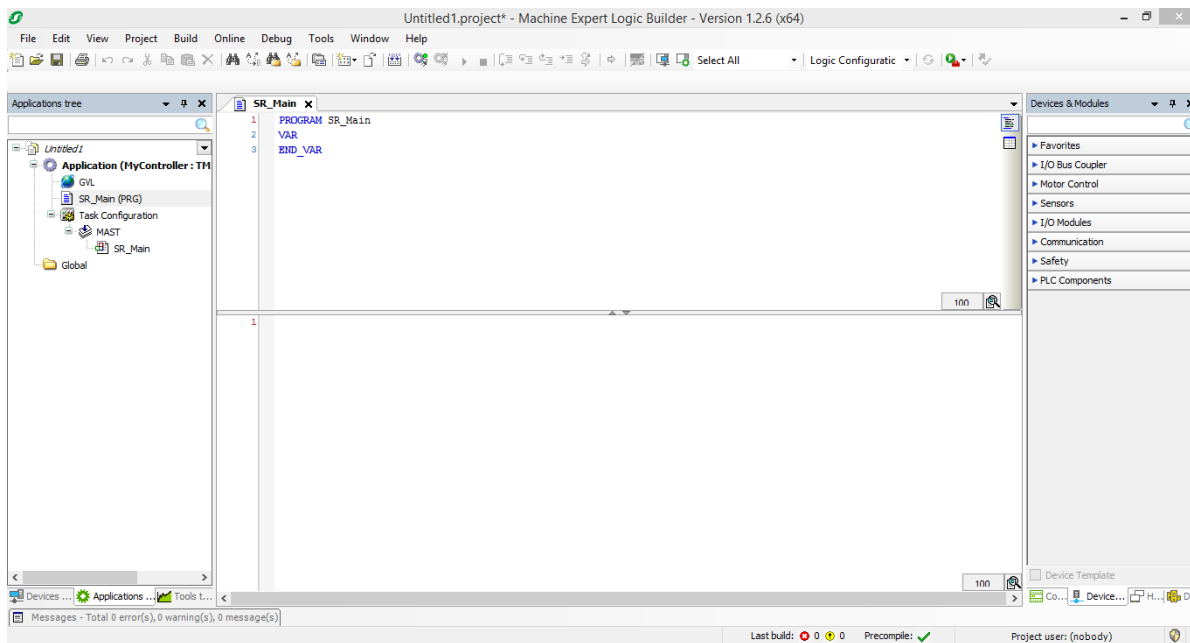
Na obrázku 4 je možné vidět, jak bude vypadat program po spuštění. Uprostřed se nachází část, ve které bude probíhat samotné programování, podle zvoleného jazyka. Vlevo nahoře jsou umístěny záložky „File“ apod. V této fázi je pro nás důležitá záložka „File“, neboť v ní se buď vytváří projekt nebo vybírá z projektů již vytvořených.



Obrázek 11: Vytváření projektu

Na obrázku 5 je vidět okno, které se zobrazí při vytváření projektu. Toto okno obsahuje nastavení jména projektu a místo jeho uložení. Mnohem důležitější jsou možnosti nastavení typu controlleru a programovacího jazyka. Controller je nutno nastavit správně, kvůli adresaci vstupů a výstupů. Pokud bude controller zvolen špatně, tyto adresy nebudou sedět a controller nebude vykonávat požadovanou funkci. Nastavení jazyka je důležitá pro samotné programování. Při zvolení špatného jazyka může dojít ke zpomalení vývoje programu nebo dokonce programátor nemusí být schopen program vytvořit. [13]

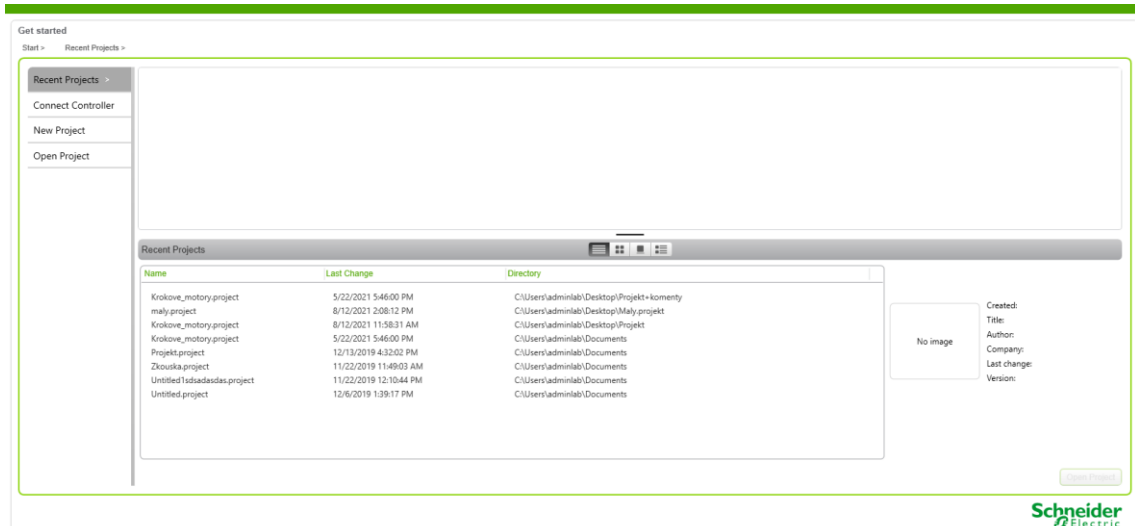
Obrázek 6 názorně ukazuje rozložení prvků na obrazovce. Prostředek je rozdělen na dvě části pro psaní kódu (v tomto případě ST). Vlevo je pak několik záložek, ze kterých důležitá záložka „Devices tree“, v níž se dají najít adresy konkrétních vstupů a výstupů, popřípadě adresy další prvků PLC.



Obrázek 12: Prostředí EcoStructure Machine Expert

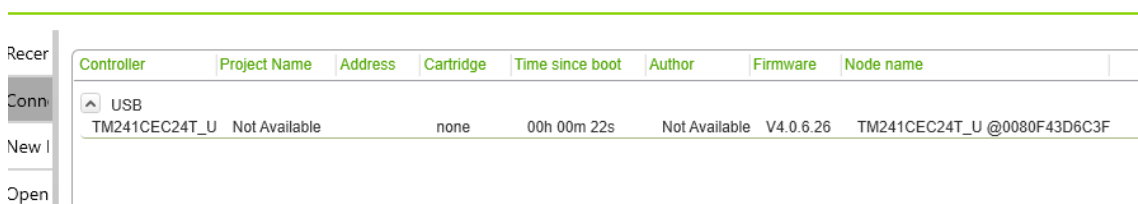
2.4 SoMachine

Tato kapitola bude věnována stručnému popsaní SoMachine, jelikož je stále možné se s ním setkat. Program vykonává stejnou funkci jako již výše zmíněný EcoStructure Machine Expert, je pouze starší verzí s jinak vypadajícím uživatelským rozhraním.



Obrázek 14: Úvodní obrazovka prostředí SoMachine

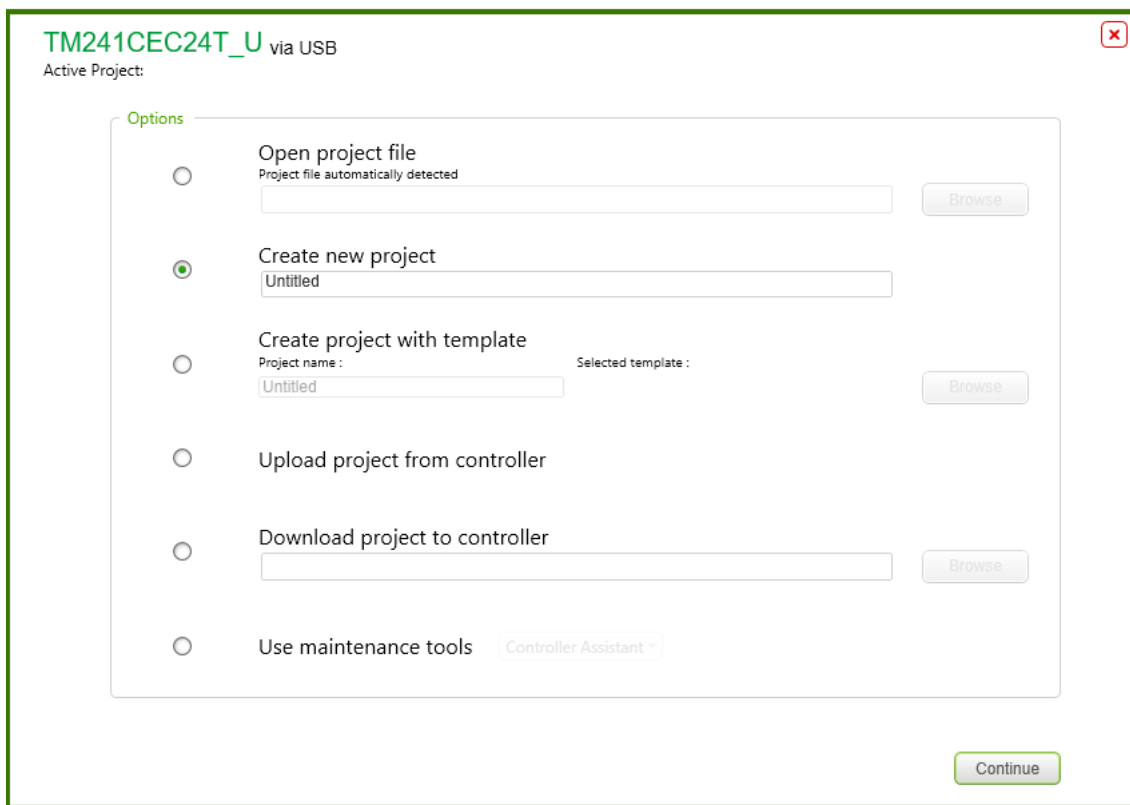
Obrázek 11 zobrazuje prostředí SoMachine hned po jeho zapnutí. Zde je možnost výběru dalšího postupu. V levé části je menu, kde je možnost vybrat mezi “Recent projects”, což je záložka zobrazující seznam posledních otevřených projektů, “New project” vytvoří nový projekt, “Open project” je záložka pro otevření konkrétního projektu, který nemusí být zobrazen v posledních projektech a poslední je “Connect controller” což je možnost kterou lze použít v případě, že PLC je již připojeno k počítači.



Obrázek 13: Připojení PLC před vytvořením projektu

Obrázek 12 ukazuje pokračování, v případě zvolení “Connect controller”. Zobrazí se okno, ve kterém je možno vidět všechny připojené PLC. Z nich se následně vybírá, ke kterému se připojí počítač. Po připojení se dále pokračuje zvolením nebo vytvořením projektu. Tyto možnosti lze dále vidět na obrázku 13. Kromě možností otevření projektu a vytvoření nového, lze taky do programu nahrát projekt, který je uložený v kontroleru,

samozřejmě za předpokladu, že v je kontroleru něco uložené. Poslední možností je provedení údržby na připojeném zařízení.



Obrázek 15: Okno po připojení PLC k počítači

3. LABORATORNÍ ÚLOHA

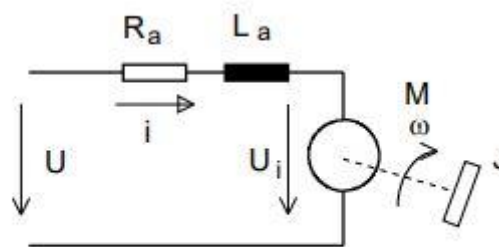
Kapitola *Laboratorní úloha* se bude zabývat vytvářením laboratorní úlohy, která bude součástí předmětu Elektrické pohony. Ke splnění laboratorní úlohy je potřeba aplikace základních znalostí z předmětu elektrické pohony a základní znalosti v oblasti programování PLC. Podrobný postup pro vytvoření programu bude popsán ve vlastní podkapitole.

3.1 Teoretická část laboratorní úlohy

Laboratorní úloha se bude zabývat problematikou zapínacích proudů. Zapínací proud jako jev se objevuje v každém elektrickém zařízení. Jeho průběh a vlastnosti jsou dány typem zařízení a jeho stavem. Následující textu se bude věnovat hlavně zapínacímu proudu elektrického motoru, protože cílem laboratorní úlohy bude změřit zapínací proud konkrétního stejnosměrného motoru.

3.1.1 Zapínací proud stejnosměrného motoru

Zapínací proud je jev, který se objevuje v případě zapnutí jakéhokoli elektrického zařízení. V našem případě se budeme zabývat rozběhem tzn. zapnutím stejnosměrného motoru. Pro popis použijeme obrázek číslo 11. [14]



Obrázek 16: Náhradní schéma stejnosměrného motoru [14]

Z obrázku lze odvodit následující rovnice:

$$U = R_a i + L_a \frac{di}{dt} + C\Phi\omega \quad (1)$$

$$C\Phi i = J \frac{d\omega}{dt} + B\omega + M_0 \quad (2)$$

Rovnice (1) a (2) popisují přechodový jev rozběhu stejnosměrného motoru. Rovnice (1) je rovnicí rovnováhy napětí v obvodu kotvy, rovnice (2) pak představuje rovnováhu momentů na hřídeli. Řešením soustavy těchto dvou rovnic bychom dostali matematické řešení průběhu proudu a změny otáček v čase. Samotné řešení této soustavy je náročné, a proto se zavádí určitá zjednodušení. [14]

Konkrétní zjednodušení, která se zavádí, jsou $L_a = 0$, $B = 0$, $M_0 = 0$. L_a je indukčnost obvodu kotvy, B je koeficient viskózního tření, M_0 je moment odporu. Po zjednodušení dostáváme následující rovnice:

$$U = R_a i + C\Phi\omega \quad (3)$$

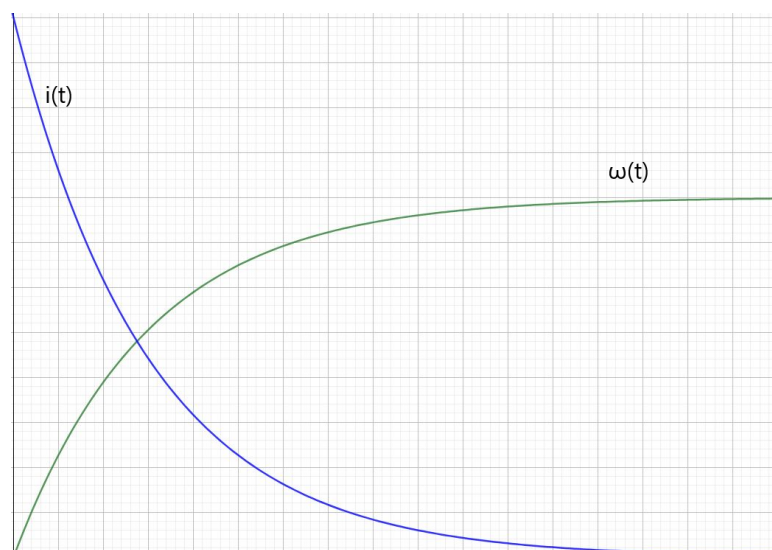
$$C\Phi i = J \frac{d\omega}{dt} \quad (4)$$

Řešením soustavy těchto diferenciálních dostáváme rovnice průběhu proudu (5) a otáček motoru (6).

$$\omega = \frac{U}{C\Phi} (1 - e^{-\frac{t}{\tau_m}}) \quad (5)$$

$$i = \frac{U}{R} e^{-\frac{t}{\tau_m}} \quad (6)$$

Z rovnic (5) a (6) pak vychází obrázek 15, na kterém jsou znázorněny průběhy proudu a otáček při rozběhu. Jejich tvar se může mírně lišit v závislosti na parametrech motoru. [14]



Obrázek 17: Průběhy proudu a otáček při zapínání motoru

3.1.2 Omezení zapínacího proudu

Vzhledem k velikostem zapínacích proudů je nutné počítat s tím, že bude nutné je omezit. Existuje několik metod pro omezení zapínacích proudů, které se používají v různých situacích. Způsob omezení proudu může být dán například typem zařízení nebo jeho výkonem. [13]

V případě elektrických motorů lze omezit zapínací proud několika způsoby dle typu a výkonu motoru:

- Softstartér
- Přepínač Y/D
- Frekvenční měnič
- Předřadný rezistor

Nás bude zajímat metoda předřadného rezistoru. Tato metoda je použitelná v případě stejnosměrného motoru s malým výkonem, což naše úloha splňuje. Tento způsob není vhodný pro dlouhodobé regulace, jelikož je ztrátový. [16]

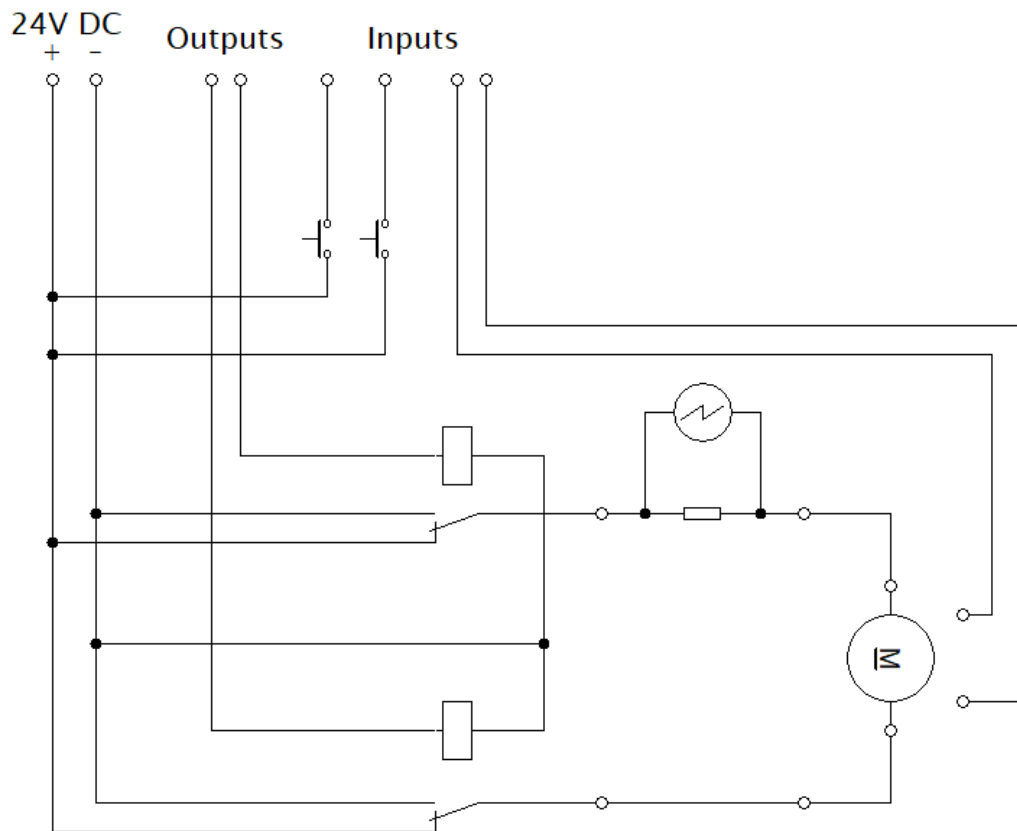
3.2 Praktická část

Tato část bude věnována praktické části laboratorní úlohy, tudíž postupu měření a v druhé části vytvoření programu pro PLC.

3.2.1 Postup laboratorní úlohy:

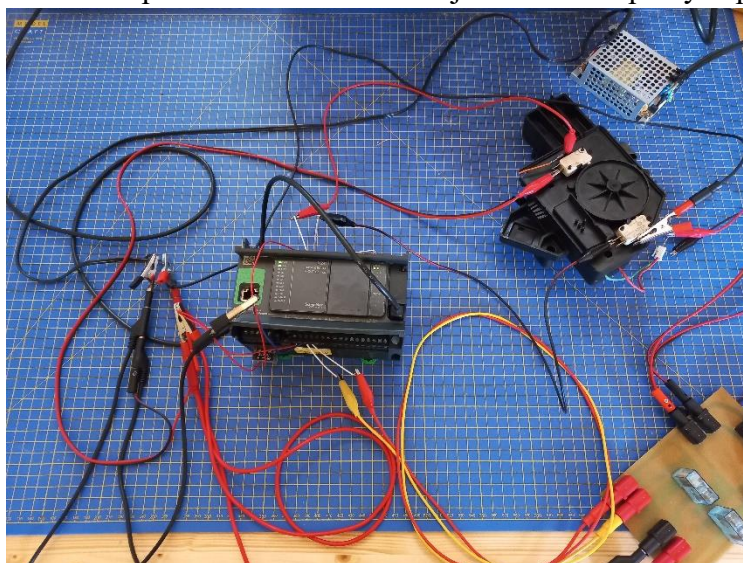
1. Po příchodu, kontrola pracoviště (přítomnost materiálu, jeho stavu apod.)
2. Zapojení úlohy dle schématu. Schéma se může měnit na základě dostupného hardwaru a způsobu měření. Největší změny může způsobit nepřítomnost koncových spínačů. V tomto případě je zapojeno méně vstupů do PLC a při vytváření programu i měření je nutno s touto situací počítat. Při zapojování PLC je také důležité pohlídat, že je zapojené napájení vstupů a výstupů.
3. Napsání programu pro PLC. Popsáno ve vlastní podkapitole. Při jeho psaní je opět nutno brát v potaz dostupné součástky.
4. Ověření funkce programu a popřípadě jeho úpravy. Vyzkoušet napsaný program a zkontrolovat, že dělá, co má.
5. Provedení měření pomocí osciloskopu. V této úloze je podstatná špička zapínacího proudu. Osciloskop měří napětí, hodnota proudu bude získána

přepočtem buď přes konstantu hallový sondy nebo hodnotu bočníku. Je také možno měřit čas ustálení.



Obrázek 18: Zapojení laboratorní úlohy

6. Zapojení předřadného rezistoru. Do obvodu bude připojen předřadný rezistor a měření bude provedeno znovu. Sledujeme změnu špičky zapínacího proudu.



Obrázek 19: Možné zapojení laboratorní úlohy

7. Vypnutí přístrojů
8. Úklid a kontrola pracoviště

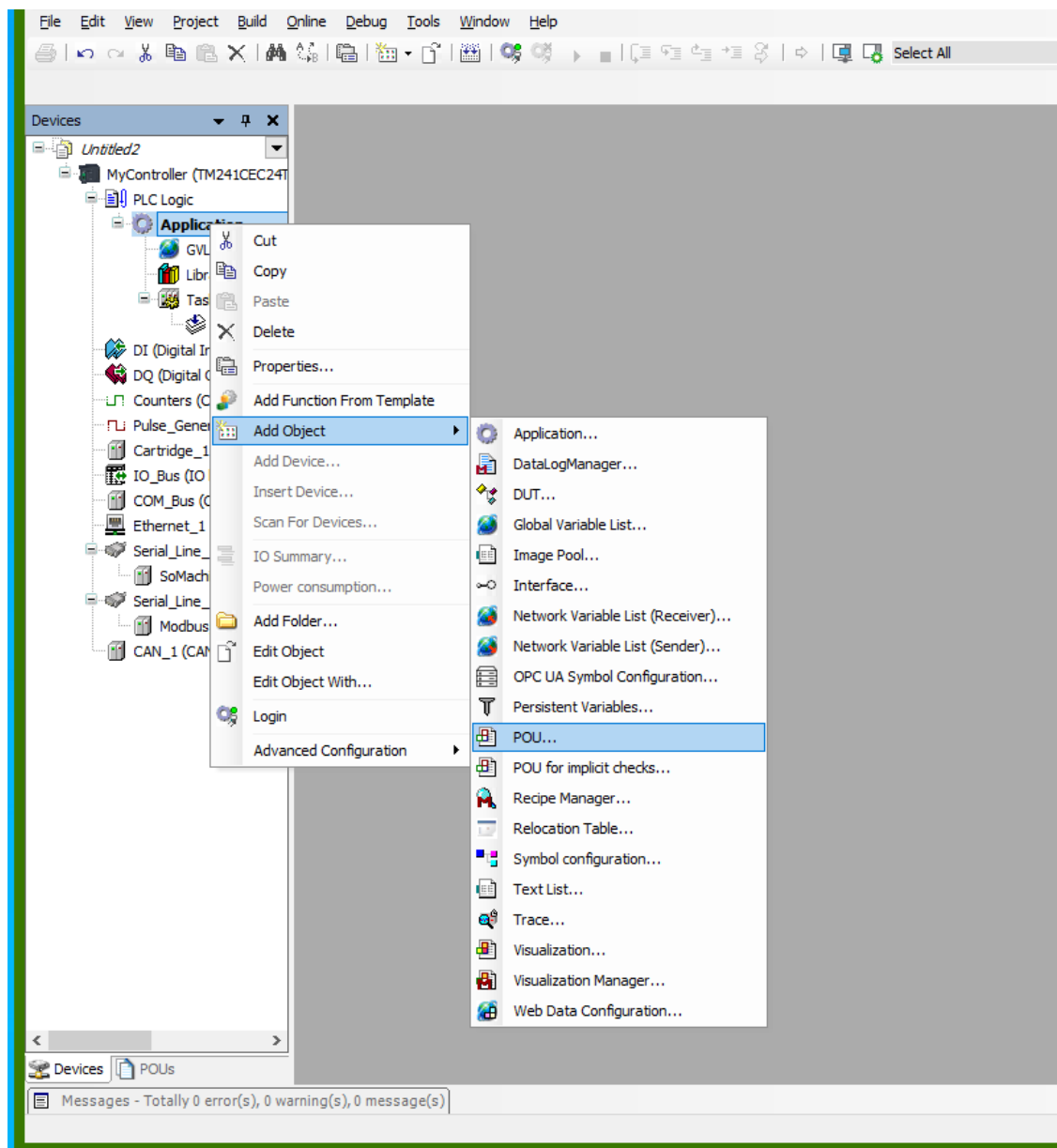
3.2.2 Vytvoření programu pro laboratorní úlohu

Tato podkapitola bude věnována vytváření programu pro laboratorní úlohu. Pro vytvoření programu je důležitý výběr programovacího jazyka. Pro účely této úlohy jsou nevhodnější jazyk strukturovaného textu a příčkového diagramu. Jazyk příčkového diagramu je vhodný svojí jednoduchostí a názorností, jelikož je to grafický jazyk. Jazyk strukturovaného textu je vhodný zejména díky svým podobnostem s jazykem C (viz. Kapitola 2.2), se kterým se studenti setkali během studia.

Pro programování PLC lze použít jak SoMachine, tak EcoStructure Machine Expert. Při vytváření programu v SoMachine je na začátku jeden krok navíc, kterým je nutnost vytvoření POU (Program Organisation Unit), do které je následně psán kód. Na obrázku

19 je názorně ukázáno vytváření POU v SoMachine. EcoStructure Machine Expert to provádí automaticky, takže po vytvoření projektu je okamžitě možno programovat.

Pro program je nejdůležitější ovládání relé, které ovládají spouštění a směr rotace



Obrázek 20: Vytváření POU v prostředí SoMachine

motoru. Ovládací cívky jsou připojeny na výstupy PLC, konkrétní čísla výstupů zaleží na pozici, kam jsou připojeny. Výstupy jsou označeny písmenem „Q“ a v programu lze

Variable	Mapping	Channel	Address	Type	Unit	Description
Outputs						
qwDQ_QW0		QW0	%QW0	WORD		
		Q0	%QX0.0	BOOL		Fast output,...
		Q1	%QX0.1	BOOL		Fast output,...
		Q2	%QX0.2	BOOL		Fast output,...
		Q3	%QX0.3	BOOL		Fast output,...
		Q4	%QX0.4	BOOL		Regular output
		Q5	%QX0.5	BOOL		Regular output
		Q6	%QX0.6	BOOL		Regular output
		Q7	%QX0.7	BOOL		Regular output
		Q8	%QX1.0	BOOL		Regular output
		Q9	%QX1.1	BOOL		Regular output

Obrázek 21: Seznam výstupů PLC s jejich adresami

dohledat konkrétní adresu pro jejich ovlivňování z programu. Tento systém platí i pro výstupy, rozdílem je jejich písmenné označení, pro které je použito písmeno „I“.

Variable	Mapping	Channel	Address	Type	Unit	Description
Inputs						
iwDI_IW0		IW0	%IW0	WORD		
		I0	%IX0.0	BOOL		Fast input, Si...
		I1	%IX0.1	BOOL		Fast input, Si...
		I2	%IX0.2	BOOL		Fast input, Si...
		I3	%IX0.3	BOOL		Fast input, Si...
		I4	%IX0.4	BOOL		Fast input, Si...
		I5	%IX0.5	BOOL		Fast input, Si...
		I6	%IX0.6	BOOL		Fast input, Si...
		I7	%IX0.7	BOOL		Fast input, Si...
		I8	%IX1.0	BOOL		Regular input...
		I9	%IX1.1	BOOL		Regular input...
		I10	%IX1.2	BOOL		Regular input...
		I11	%IX1.3	BOOL		Regular input...
		I12	%IX1.4	BOOL		Regular input...
		I13	%IX1.5	BOOL		Regular input...
ibDI_IB1		IB1	%IB2	BYTE		

Obrázek 22: Seznam vstupů PLC s jejich adresami

Logika programu záleží na způsobu měření a dostupném hardwaru. V případě přítomnosti koncových kontaktů motoru lze udělat kompletně automatický program ovládání motoru. V opačném případě je nutno vymyslet způsob měření např. jednotlivá měření ovládaná ručně.

4. VÝROBA LABORATORNÍHO PŘÍPRAVKU

Tato kapitola bude věnován stručnému popisu výroby přípravku, který byl vyrobený pro potřeby této úlohy. Jeho hlavní součástí je dvojice relé, která slouží k ovládání stejnosměrného motorku. Požadavky na toto relé byli dány použitím PLC a zapojením.

Konkrétně tedy:

- Přepínací kontakt – vzhledem k zapojení
- Cívka 24 – vzhledem k napájení
- Co nejvyšší maximální proud kontaktem – měření různých stejnosměrných motorků

Na základě těchto požadavků bylo vybráno relé značky Finder. Přepínací kontakt , s maximálním spínaným proudem 30 A a maximálním trvalým proudem 16 A, je více než dostačující. Výhodou také může být prohledný obal, který umožňuje poměrně snadnou kontrolu, zda relé přepnulo, a také brání kontakt před zašpiněním a tím narušení kontaktní plochy.



Obrázek 23: Relé Finder 40.61

Po vybrání relé byla vytvořena deska plošných spojů. Její leptání bylo provedeno chloridem železitým, kvůli snadnému postupu a relativní bezpečnosti. Deska byla dále osazena výše již zmíněnými relé a kontakty pro připojení dalších periferií a pro připojení k PLC a ke zdroji. Výsledný výrobek je možno vidět na obrázku 23.



Obrázek 24: Dokončený přípravek pro laboratorní úlohu

ZÁVĚR

Téma této bakalářské práce je *Využití programovatelného logického automatu ve výuce elektrický pohonů*. Důraz byl kladen na využití programovatelného logického automatu ve výuce na vysoké škole, zejména pak pro studenty předmětu Elektrické pohony.

Teoretická část bakalářské práce byla rozdělena do několika částí. První byla věnována krátkému úvodu do oblasti programovatelných logických automatů. Druhá část byla věnována programovacím jazykům pro PLC. Tato část byla ještě dále rozdělena grafické a textové jazyky. Do grafických jazyků patří jazyk příčkového diagramu, který využívá svoje symboly a jejich kombinaci k tvoření logiky programu, a jazyk funkčních bloků jehož hlavní přednost je vytváření vlastních bloků a to i v jiných jazycích. Do textových jazyků patří jazyk strukturovaného textu, který je velmi podobný jazyku C, což je jeho největší výhodou, a jazyk seznamu instrukcí, který je podobný assembleru.

Další část se zabývala popisem laboratorní úlohy věnující se zapínacím proudům. V její teoretické části jsme se věnovali zapínacímu proudu stejnosměrného motoru a vztahům, z nichž byl následně, při určitém zjednodušení, odvozen průběh zapínacího proudu a úhlové rychlosti motoru. Praktická část pak byla věnována postupu měření laboratorní úlohy a vytvoření programu pro tuto úlohu, jehož vytvoření je nedílnou součástí této úlohy.

Cíle bakalářské práce byly splněny, byla vymyšlena úloha, při které je využit programovatelný logický automat. Tyto úlohy mohou posloužit studentům vysokých škol v prohlubování jejich znalostí a dosavadních zkušeností.

LITERATURA

- [1] KOVÁŘ, Josef, Zuzana PROKOPOVÁ a Ladislav ŠMEJKAL. Programování dle normy IEC 61 131 [online]. [cit. 2021-01-05]. Dostupné z: https://www.pslib.cz/jaroslav.semerad/board/E4_2015_16/programovani_dle_normy_iec61131.pdf
- [2] MARTINÁSKOVÁ, Marie a Ladislav ŠMEJKAL. Řízení programovatelnými automaty. Vyd. 2. V Praze: Vydavatelství ČVUT, 2004. ISBN 80-01-02925-5.
- [3] MASLAR, Mark. PLC Standard Programming Languages: IEC 1131-3 [online]. 1996 [cit. 2021-01-05]. Dostupné z: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=535979>
- [4] ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace. Praha: BEN – technická literatura, 1999. ISBN 80-86056-58-9.
- [5] ŠMEJKAL, Ladislav. Esperanto programátorů PLC: programování podle normy IEC/EN 61131-3 (část 4). Automa [online]. 2011(12), 38–42 [cit. 2021-01-05]. ISSN 1210-9592. Dostupné z: https://automa.cz/Aton/FileRepository/pdf_articles/45413.pdf
- [6] ŠMEJKAL, Ladislav. Esperanto programátorů PLC: programování podle normy IEC/EN 61131-3 (část 10). Automa [online]. 2012(7), 44–45 [cit. 2021-01-05]. ISSN 1210-9592. Dostupné z: http://automa.cz/Aton/FileRepository/pdf_articles/9732.pdf
- [7] URBAN, Luboš. Programování PLC podle normy IEC EN 61131-3 – víc než jednotné jazyky. Automa [online]. 2005(2) [cit. 2021-01-05]. ISSN 1210-9592. Dostupné z: https://automa.cz/cz/casopis-clanky/programovani-plc-podle-normy-iec-en-61131-3-vic-nez-jednotne-jazyky-2005_02_30310_1237/
- [8] Structured Text Tutorial For PLC Programmers. Free PLC Training: Learn PLC Programming Online | PLC Academy [online]. Dostupné z: <https://www.plcademy.com/structured-text-tutorial/>
- [9] C - Data types. Tutorials Point [online]. [cit. 2021-01-05]. Dostupné z: https://www.tutorialspoint.com/cprogramming/c_data_types.html
- [10] Logic gate (AND, OR, XOR, NOT, NAND, NOR and XNOR) [online]. 2020 [cit. 2021-01-05]. Dostupné z: <https://whatis.techtarget.com/definition/logic-gate-AND-OR-XOR-NOT-NAND-NOR-and-XNOR>
- [11] Co by měl každý vědět o programovacích jazycích PLC [online]. 29.8.2017 [cit. 2021-01-05]. Dostupné z: <https://www.elektroprumysl.cz/software/co-by-mel-kazdy-vedet-o-programovacich-jazycich-plc>
- [12] PLC Ladder Logic Programming Tutorial (Basics) | PLC Academy. Free PLC Training: Learn PLC Programming Online | PLC Academy [online]. Dostupné z: <https://www.plcademy.com/ladder-logic-tutorial/>
- [13] Nejčastější metody připojení elektromotoru [online]. 19.7.2017 [cit. 2021-8-15]. Dostupné z: <https://www.elektroprumysl.cz/energetika/nejcastejsi-metody-pripojeni-elektromotoru-k-siti>

- [14] SKALICKÝ, Jiří. Elektrické servopohony [online]. Vyd. 2. Brno: Vysoké učení technické, 2001 [cit. 2021-5-27]. ISBN 80-214-1978-4. Dostupné z: https://www.vutbr.cz/www_base/priloha.php?dpid=33400
- [15] TM241CEC24T. Schneider Electric Česká republika [online]. [cit. 2021-5-27]. Dostupné z: <https://www.se.com/cz/cs/product/TM241CEC24T/plc-modicon-m241-24-io-poz.logika-ethernet-can-master/>
- [16] Řízení napětí předřadným odporem [online]. 2.2.2016 [cit. 2021-8-15]. Dostupné z: <https://elektrika.cz/data/clanky/rizeni-napeti-predradnym-odporem-nebo-delicem-napeti>
- [17] Sestava PLC. PLC Automatizace [online]. [cit. 2021-8-15]. Dostupné z: <http://plc-automatizace.cz/knihovna/plc/plc-hw-sestava.htm>
- [18] Co je PLC neboli Programovatelný Logický Automat. DreamLand-PLC [online]. [cit. 2021-8-15]. Dostupné z: <https://dreamland-plc.cz/plc-programovatelný-logický-automat/>
- [19] PLC Tecomat TC700 [online]. [cit. 2021-01-05]. Dostupné z: <https://www.tecomat.cz/products/cat/cz/plc-tecomat-tc700-1/>
- [20] URBAN, Luboš a Šmejkal LADISLAV. Programovatelné automaty – PLC, nebo PAC? Automa [online]. 2003(6) [cit. 2021-01-05]. ISSN 1210-9592. Dostupné z: https://automa.cz/cz/casopis-clanky/programovatelné-automaty-plc-nebo-pac-2003_06_28832_683/
- [21] Datasheet pro relé Finder. GM Electronic [online]. [cit. 2021-8-16]. Dostupné z: <https://www.gme.cz/data/attachments/dsh.634-039.1.pdf>