

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] Kolářský 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 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. Druhá část se věnuje popisu tří laboratorních úloh. Každá z úloh obsahuje schéma zapojení a diagram, jako vodičko pro vytvoření programu pro danou úlohu. Třetí část je věnována přípravě výroby desky plošných spojů a výběru součástek. Poslední část obsahuje obrázky a poznámky ohledně výroby přípravku pro laboratorní úlohy.

## **Klíčová slova**

Laboratorní úloha, PLC, programování PLC, výroba desky plošných spojů, elektrické pohony

## **Abstract**

Bachelor thesis is divided into four parts. First 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. Second part deals with description of three laboratory tasks. The last part contains pictures a information about laboratory task's made equipment.

## **Keywords**

Laboratory task, PLC, programming PLC, production of printed circuit boards, electric drive



## **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 pohonných*

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

<b>SEZNAM SYMBOLŮ A ZKRATEK .....</b>	<b>8</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>9</b>
<b>SEZNAM TABULEK.....</b>	<b>10</b>
<b>ÚVOD .....</b>	<b>11</b>
<b>1. PROGRAMOVÁNÍ LOGICKÝCH AUTOMATŮ .....</b>	<b>12</b>
1.1 GRAFICKÉ JAZYKY .....	13
1.1.1 <i>Jazyk příčkového diagramu .....</i>	<i>13</i>
1.1.2 <i>Jazyk funkčního blokového schématu.....</i>	<i>14</i>
1.2 TEXTOVÉ JAZYKY.....	16
1.2.1 <i>Jazyk strukturovaného textu.....</i>	<i>16</i>
1.2.2 <i>Jazyk seznamu instrukcí.....</i>	<i>18</i>
1.3 ECOSTRUCTURE MACHINE EXPERT .....	20
<b>2. LABORATORNÍ ÚLOHY .....</b>	<b>23</b>
2.1. LABORATORNÍ ÚLOHA 1.....	24
2.2. LABORATORNÍ ÚLOHA 2.....	27
2.3. LABORATORNÍ ÚLOHA 3.....	28
<b>3. PŘÍPRAVA SCHÉMATU, VÝBĚR SOUČÁSTEK.....</b>	<b>30</b>
<b>4. VÝROBA DPS A OSAZENÍ SOUČÁSTEK.....</b>	<b>33</b>
<b>5. ZÁVĚR.....</b>	<b>38</b>
<b>LITERATURA.....</b>	<b>39</b>

# SEZNAM SYMBOLŮ A ZKRATEK

## Zkratky:

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

## Symboly:

$U$	napětí	[V]
$I$	proud	[A]
$\eta$	účinnost	[-]
$\omega$	úhlová rychlost	[rad/s]
$M$	moment síly	[Nm]
$P$	Výkon	[W]
$n$	otáčky	[1/min]
$c\varphi$	konstanta motoru	[Nm/A]

## SEZNAM OBRÁZKŮ

Obrázek 1: Jazyky pro programování řídicích jednotek specifikované v normě IEC EN 61131-3 [7].....	12
Obrázek 2: Symboly jazyka příčkového diagramu [5].....	13
Obrázek 3: Operátory jazyka mnemokódů [5].....	19
Obrázek 4: Prostředí EcoStructure Machine Expert při spuštění.....	20
Obrázek 5: Vytváření projektu.....	21
Obrázek 6: Prostředí EcoStructure Machine Expert .....	22
Obrázek 7: Zapojení úlohy č. 1.....	24
Obrázek 8: Jednoduchý vývojový diagram pro úlohy č. 1 a 2 .....	25
Obrázek 9: Zapojení pro úlohu č. 2 .....	27
Obrázek 10: Zapojení pro úlohu č. 3 .....	28
Obrázek 11: Vývojový diagram pro úlohu č. 3.....	29
Obrázek 12: Relé FINDER 40.61 [13].....	30
Obrázek 13: Zdířky s průměrem 4 mm (červená a černá) [13] .....	31
Obrázek 14: Schéma zapojení před úpravou pro DPS .....	31
Obrázek 15: Schéma zapojení upravené pro DPS (pohled ze strany součástek).....	32
Obrázek 16: Schéma zakreslené na desce plošných spojů .....	33
Obrázek 17: DPS ponořená do chloridu železitého .....	34
Obrázek 18: Výsledek po vyleptání DPS chloridem železitým .....	34
Obrázek 19: DPS po odstranění ikoustu .....	35
Obrázek 20: DPS po vyvrtání dírek .....	36
Obrázek 21: DPS osazená součástkami .....	37

## SEZNAM TABULEK

Tabulka 1: Tabulka datových typů v jazyce strukturovaného textu [1] .....	17
Tabulka 2: Tabulka pro jazyk funkcí strukturovaného textu [1] .....	18

# ÚVOD

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

V teoretické části bakalářské práce budou představeny kapitoly týkající se Programování logických automatů, kde důležitou součástí budou programovací jazyky. Budou zmíněny jak grafické jazyky, tak textové jazyky. Grafické jazyky jsou spolu s textovými jazyky nedílnou součástí programování PLC.

Teoreticky i prakticky bude představena kapitola týkající se laboratorních úloh, kde dojde k z vytváření laboratorních úloh, které budou dále využity a které budou rovněž cílem této bakalářské práce.

Cílem této bakalářské práce je vytvořit úlohy, které budou sloužit studentům předmětu Elektrické pohony. Aby bylo možné dosáhnout tohoto cíle, bude potřeba vytvořit laboratorní úlohy, vytvořit schéma, vybrat součástky a nakonec vyrobit desku plošných spojů.



# 1. 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 systematictější a obecněji definovány, než tomu je u dosavadních jazyků. Navíc všechny navzájem sdílejí společné prvky. Přejechy mezi různými typy jazyků jsou tak snazší.*“ [1]

Podle této normy jsou na následujícím obrázku rozděleny všechny jazyky (jak grafické, tak textové), se kterými je možno pracovat. Výjimku tvoří jazyk sekvenčního

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

Obrázek 1: Jazyky pro programování řídicích jednotek specifikované v normě IEC EN 61131-3 [7]

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]

## 1.1 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.

### 1.1.1 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 set	<i>set coil</i>	(S)-
cívka s funkcí reset	<i>reset coil</i>	(R)-

Obrázek 2: 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]

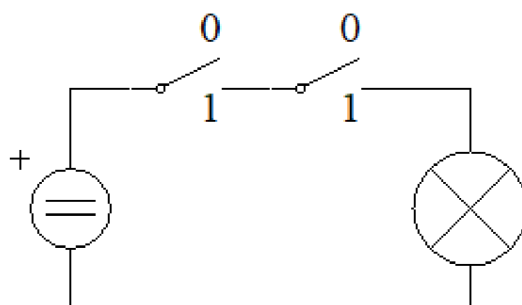
Prvními dvěma symboly jsou *spínací* a *rozpínací kontakt*. Tyto symboly pracují se vstupními bity. Jejich funkce je velmi podobná. Oba tyto symboly kontrolují svůj vstup a na jeho základě ovlivňují svůj výsledek. Výsledek spínacího kontaktu je stejná hodnota, jako je hodnota vstupu. Naopak výsledná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

### **1.1.2 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]

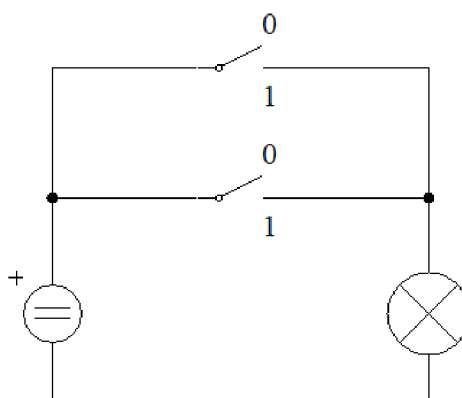
Prvním logická funkce je funkce AND. AND je funkce, jejímž výsledkem může být 1 nebo 0, respektive „true“ nebo „false“. Výsledek funkce závisí na vstupech. V případě, že jsou všechny stupy rovny 1, celá funkce je vyhodnocena výsledkem 1. V případě, že je alespoň jeden vstup roven 0, následný výsledek 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 je mezi vstupy nutný pouze jeden vstup s logickou 1, aby byl výstup roven 1. Aby byl



Obrázek 4: Zobrazení logické funkce OR pomocí elektrického obvodu

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]

## 1.2 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]

### 1.2.1 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]

### 1.2.2 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]

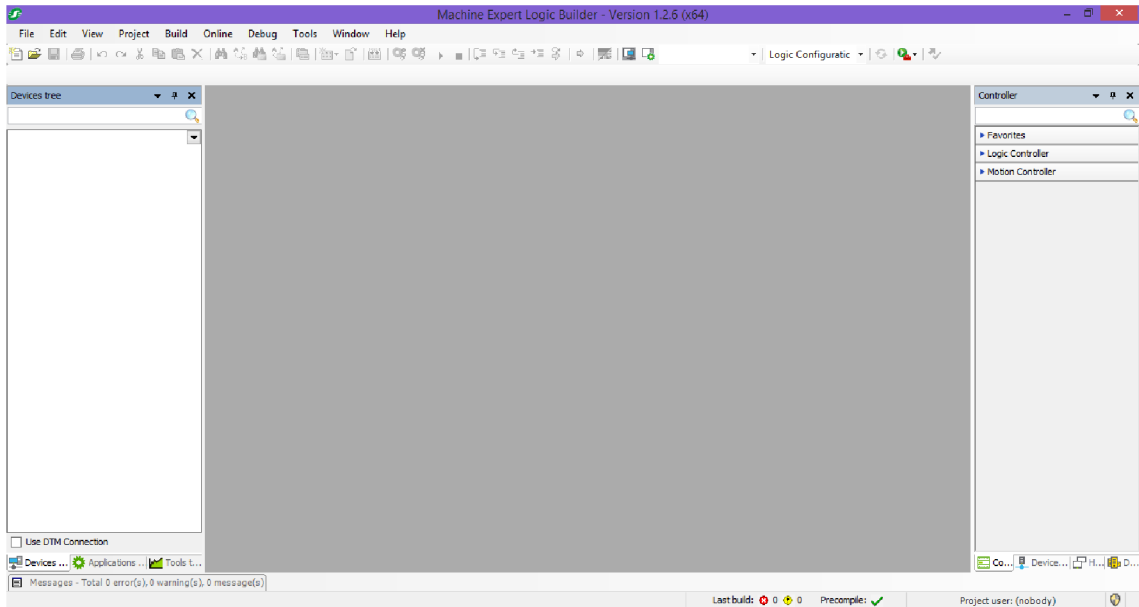
Operátor	Modifikátor	Popis funkce
LD	N	LD ( <i>load</i> , <i>sejmi</i> ) – nastaví výsledek podle aktuální hodnoty adresovaného operandu ( <i>sejme</i> aktuální hodnotu adresovaného operandu do střádače)
AND	N, (	booleovský součin AND
OR	N, (	booleovský součet inkluzivní OR
XOR	N, (	booleovský součet 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 3: Operátory jazyka mnemokódů [5]



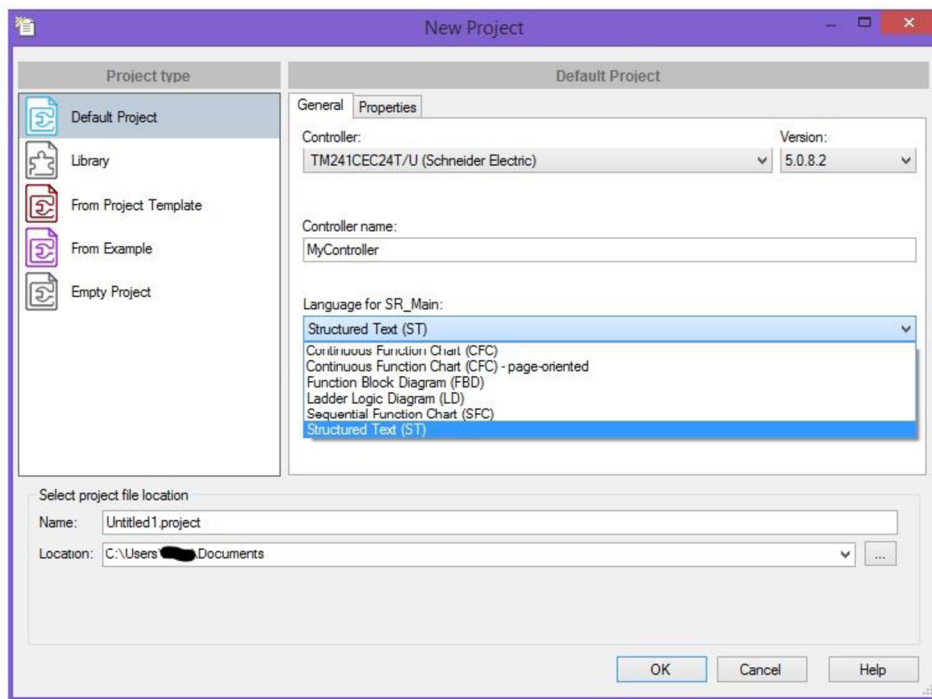
### 1.3 EcoStructure Machine Expert

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 4: Prostředí EcoStructure Machine Expert při spuštění

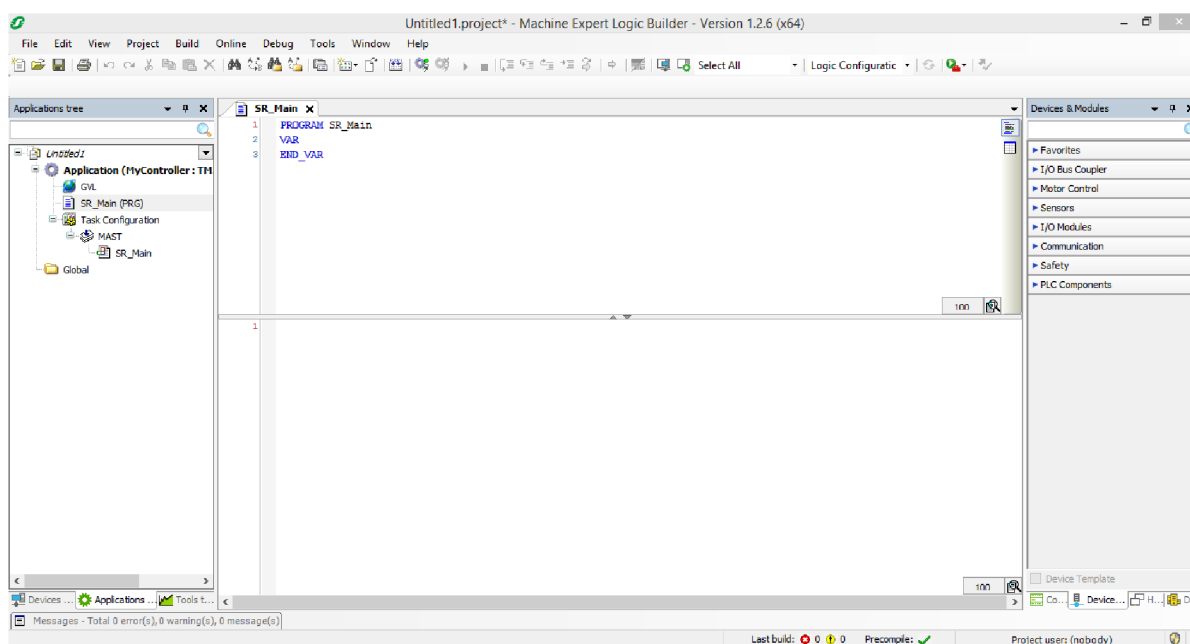
Na obrázku 4 je možné vidět, jak bude vypadat program po spuštění. Uprostřed se nachází část, ve které bude probíhat samotné programování, podle zvoleného jazyka. Vlevo nahoře jsou umístěny záložky „File“ apod. V této fázi je pro nás důležitá záložka „File“, neboť v ní se buď vytváří projekt nebo vybírá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



Obrázek 5: 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 6: Prostředí EcoStructure Machine Expert

## 2. LABORATORNÍ ÚLOHY

Kapitola *Laboratorní úlohy* se bude zabývat vytvářením laboratorních úloh, které budou sloužit studentům předmětu Elektrické pohony. V následujících podkapitolách budou představeny úlohy, které budou vyžadovat znalosti ze zmíněného předmětu. Ke splnění laboratorních úloh je potřeba aplikace základních znalostí spojené s prací s elektrickými pohony. Tyto úlohy budou testovat aplikaci základních znalostí, zejména půjde o aplikaci znalostí z předmětu Elektrické pohony. Úlohy jsou demonstrací teoretických znalostí získaných na přednáškách a v seminářích.

Celkem budou v následujících podkapitolách představeny 3 úlohy. Tyto úlohy budou blíže popsány, budou rovněž graficky znázorněny. Cílem studentů bude aplikace poznatků v praxi, aby bylo možné tyto úlohy vyřešit. Pro všechny úlohy by měl být použit motor s konstantním magnetickým tokem.

Obecný postup aplikovatelný pro všechny úlohy:

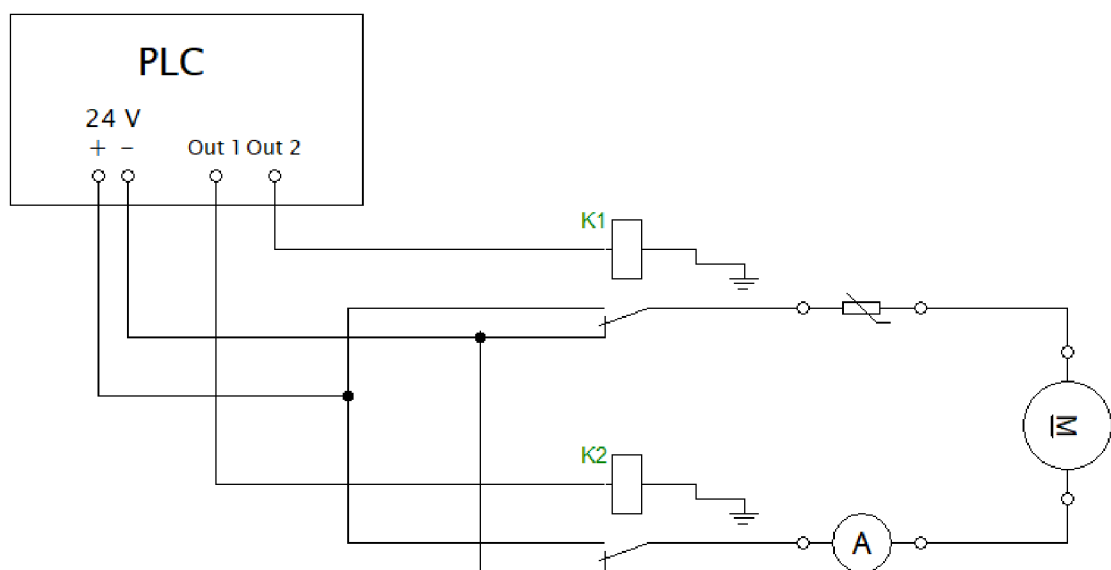
Studenti přijdou na pracoviště, které zkontrolují. Kontrolovat se bude stav zařízení, která se budou používat (stav materiálu, kontrolu drátů, například narušená izolace, utržená koncovka). Případně nalezené problémy budou neprodleně hlásit svému vyučujícímu. Po kontrole materiálu dochází k zapojení úlohy (vyučující zkontroluje správné zapojení úlohy, aby se předešlo zbytečným chybám). Po zapojení úlohy dojde k psaní programu. Studenti budou mít možnost nastudovat program dopředu (příčkový diagram, strukturovaný text), aby základům programu rozuměli. Na pracovišti by měl být přítomen seznam funkcí/symbolů, který by studenti mohli v průběhu své úlohy využít v případě potřeby. Napsaný program by měl být zkontrolován a schválen vyučujícím. V případě správně napsaného programu pokračují studenti v dalším kroku a začínají měřit. Po skončení měření jsou hodnoty vyučujícím podepsány a schváleny. Studenti zkontrolují stav zařízení, vypnout je a odchází z uklizeného pracoviště.

## 2.1. Laboratorní úloha 1

Cílem této úlohy je věnovat se základním parametrům pohonu (elektrické, mechanické) a ukázat jednoduchý, nýbrž neefektivní způsob řízení motoru.. Úloha slouží k ukázce řízení motoru proměnným odporem, přičemž dochází k ukázce nedokonalosti takového řízení. Řízení není ideální, protože snižuje účinnost pohonu jako takového. Odebíraná energie ze zdroje, která normálně putuje do motoru (mechanická energie), se na rezistoru přeměňuje na teplo, což je neúčinná energie a není primárním účelem takovou energii vyrábět. Tato úloha ukazuje neefektivní, ale jednoduché řízení motoru pomocí proměnného rezistoru.

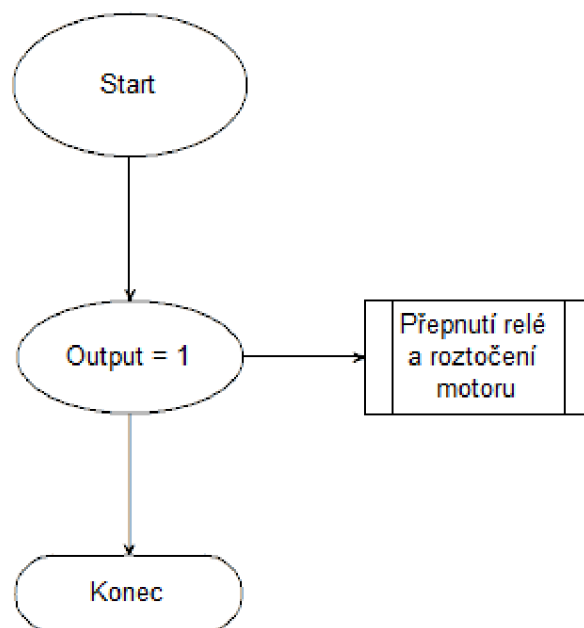
### Postup:

1. Po příchodu, kontrola pracoviště
2. Zapojení úlohy dle schématu



Obrázek 7: Zapojení úlohy č. 1

3. Napsání programu – nastavení jednoho výstupu ovládající jedno relé



Obrázek 8: Jednoduchý vývojový diagram pro úlohy č. 1 a 2

4. Ověření funkce programu, případně jeho úpravy
5. Provedení měření, zápis hodnot proudu, otáček a momentu
6. Opakování měření pro různé hodnoty odporu
7. Vypnutí přístrojů
8. Úklid a kontrola pracoviště

Studenti vytvoří protokol, který bude náležitě formulován. Podstatnou část budou tvořit zpracovaná data z měření a hodnoty z nich vypočtené, které mohou být zapsány do tabulky níže.

Tabulka 3: Příklad tabulky pro úlohu číslo 1

No.	I [A]	M [Nm]	n [ot/min]	$c\phi$ [Nm/A]	$\omega$ [rad/s]	$U_i$ [V]	$P_{el}$ [W]	$P_{mech}$ [W]	$\eta$ [-]
1									
2									
3									
4									
5									
6									
7									
8									
9									
10									

Tabulky č. 1 může být inspirací pro zpracování dat. Levá část tabulky bude vyplněna hodnotami z měření. Pravá část pak bude výsledky výpočtů z následujících rovnic.

$$\omega = \frac{2 \cdot \pi \cdot n}{60} \quad (1)$$

Rovnice XY slouží k přepočtu otáček na úhlovou rychlost. Úhlová rychlost bude použita pro další výpočty (např. mechanický výkon). [14]

$$c\phi = \frac{M}{I} \quad (2)$$

Další rovnice slouží k výpočtu „ $c\phi$ “. „ $c$ “ je konstanta motoru (napěťová, momentová),  $\phi$  je magnetický tok. Tato hodnota by měla pro všechna měření zůstat přibližně stejná (pouze chyba způsobená nepřesností měření). [14]

$$U_i = c\phi \cdot \omega \quad (3)$$

Rovnice (3) slouží k výpočtu napětí, které indukuje uvnitř motoru.

$$P_{el} = U_a \cdot I \quad (4)$$

Z rovnice (4) lze vypočítat výkon odebíraný ze zdroje, který bude následně použit pro výpočet účinnosti. [14]

$$P_{Mech} = M \cdot \omega \quad (5)$$

Rovnice (5) je používána pro výpočet mechanického výkonu, který bude stejně jako elektrický výkon, použit pro výpočet účinnosti.

$$\eta = \frac{P_{mech}}{P_{el}} \quad (6)$$

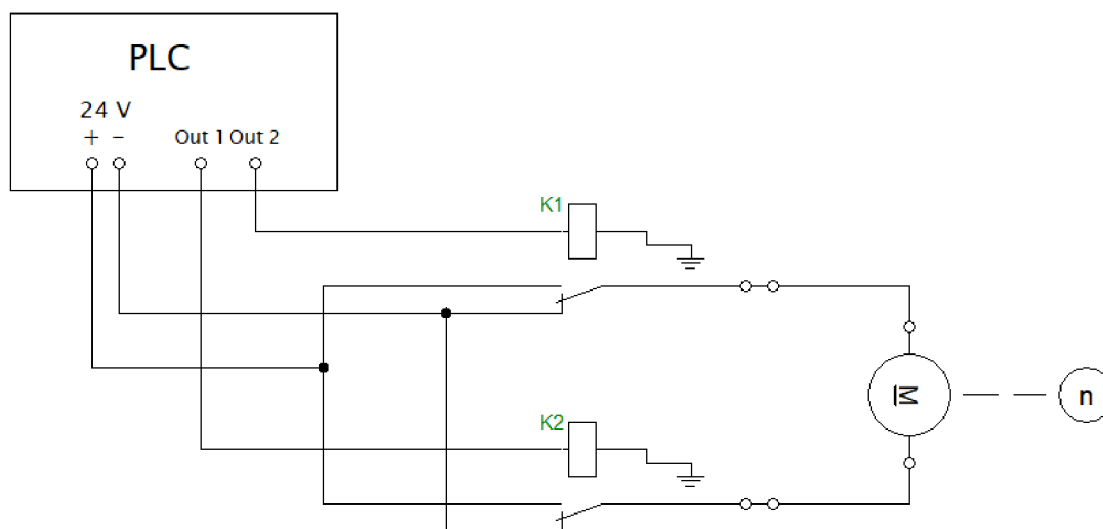
Poslední rovnicí je určena pro výpočet účinnosti. Účinnost může být následně uváděna i v procentech.

## 2.2. Laboratorní úloha 2

Cílem této úlohy je provést měření zatěžovací charakteristiky. Zatěžovací charakteristika je graf, který popisuje otáčky motoru podle zátěže na hřídeli. Charakteristika je pro různé stejnosměrné motory jiná. Studenti budou mít za úkol nastavit moment, zapsat si jeho hodnotu a následně zapsat hodnotu otáček pro daný moment. Měření by mělo být provedeno alespoň desetkrát, přičemž součástí vytvořeného protokolu bude vyneseny graf.

### Postup:

1. Po příchodu, kontrola pracoviště
2. Zapojení úlohy dle schématu (pouze elektrická část)



Obrázek 9: Zapojení pro úlohu č. 2

3. Napsání programu – nastavení jednoho výstupu ovládající jedno z relé
4. Ověření funkce programu, případně jeho úpravy
5. Provedení měření – nastavování momentu a měření otáček, zápis obou hodnot
6. Opakování měření – alespoň 10x
7. Vypnutí přístrojů
8. Úklid a kontrola pracoviště

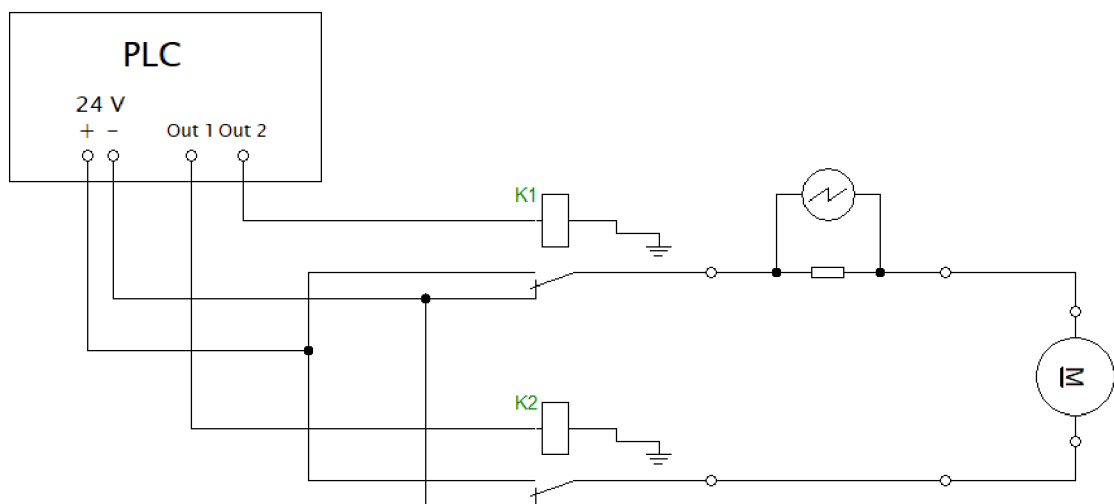


## 2.3. Laboratorní úloha 3

Laboratorní úloha č. 3 se bude věnovat zapínacím proudům pohonu. V této úloze studenti vytvoří program pro PLC a následně změří a zaznamenají zapínací proud pomocí osciloskopu. Úloha 3. je oproti dvěma předchozím úlohám těžší, protože se zde objevuje těžší kód a samotné zapojení je rovněž složitější. Způsob měření může být rovněž proveden více způsoby. Prvním způsobem je měření pomocí bočníku, kdy se převádí hodnota napětí na proud pomocí odporu. Druhý způsob spočívá v možnosti využití Hallovy sondy, kde se měří hallovo napětí, které je přímo úměrné proudu. Studenti během měření změří zapínací proud a jeho průběh. Následně provedou měření znovu s tím rozdílem, že tentokrát bude do obvodu zařazen další rezistor, který bude mít za následek změny maximální hodnoty proudu. Maximální hodnota proudu je právě tou veličinou, která studenty zajímá.

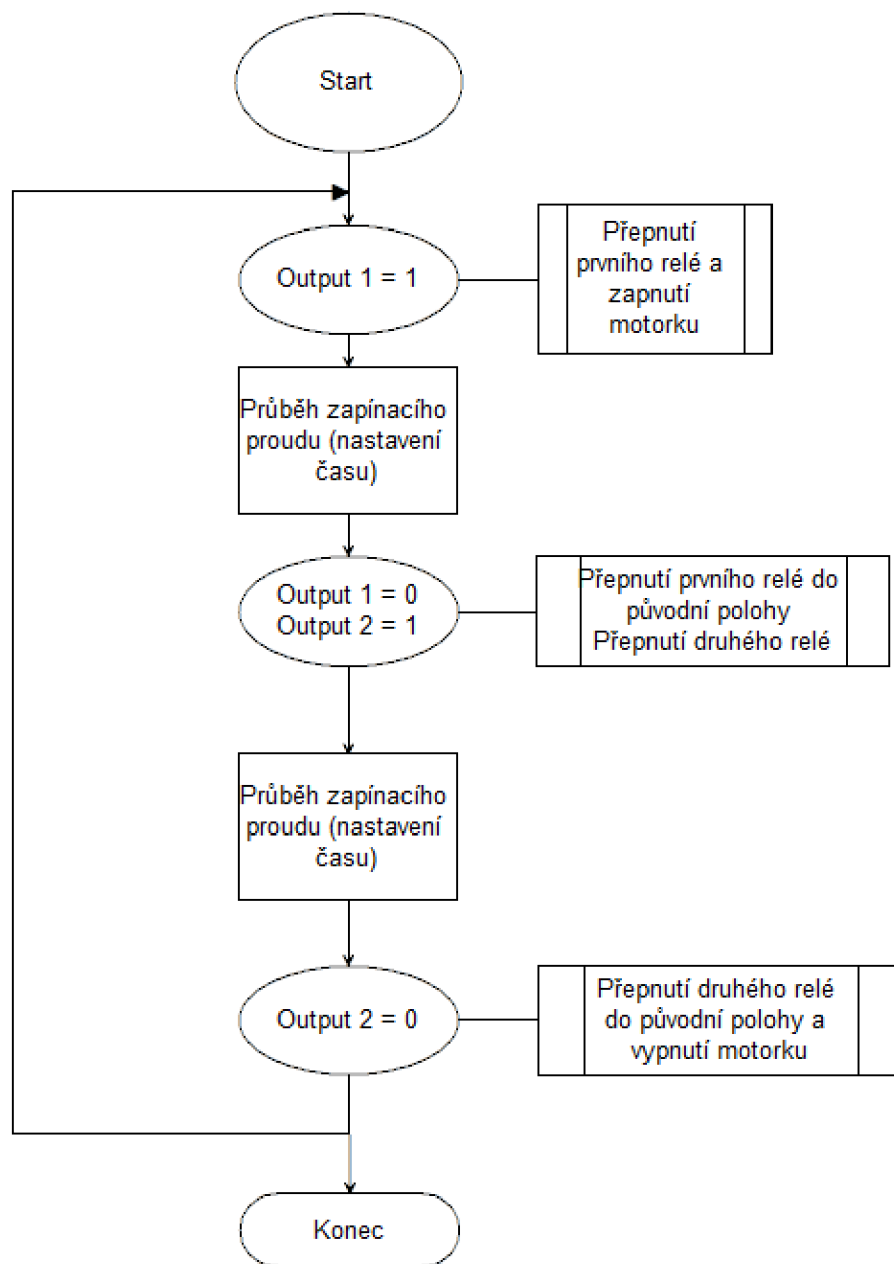
### Postup laboratorní úlohy:

1. Po příchodu, kontrola pracoviště (přítomnost materiálu, jeho stavu apod.)
2. Zapojení úlohy dle schématu (schéma se může měnit, v závislosti na způsobu připojení osciloskopu – Hallova sonda nebo bočník)



Obrázek 10: Zapojení pro úlohu č. 3

3. Napsání programu pro ovládání relé – studenti si mohou vymyslet vlastní logiku nebo se mohou řídit podle diagramu na obrázku 11



Obrázek 11: Vývojový diagram pro úlohu č. 3

4. Ověření funkce programu a popřípadě jeho úpravy
5. Provedení měření pomocí osciloskopu – maximální hodnoty proud, popřípadě času do ustálení proud (uložení záznamu na flash disk)
6. Opakování měření (alespoň dvě různé hodnoty předřadného odporu)
7. Vypnutí přístrojů
8. Úklid a kontrola pracoviště

### 3. PŘÍPRAVA SCHÉMATU, VÝBĚR SOUČÁSTEK

Kapitola se bude zabývat přípravou schématu pro DPS a výběrem součástek. Objeví se zde součástky, které byly použity pro osazení DPS. Tyto součástky zde budou blíže popsány a bude rovněž odůvodněn jejich výběr. Tyto součástky jsou velmi důležitou částí projektu, který je blíže popsán v kapitole 4.

Mezi první důležitou součástkou, která byla vybrána na projekt je relé. Tato součástka je nejdůležitější součástkou v samotném zapojení. Pro můj projekt bylo vybráno relé FINDER 40.61. Toto relé bylo vybráno z následujících důvodů [13]:

- typ kontaktu (přepínací kontakt)
- cívka 24 V
- uzavřené relátko (bezpečnostní důvod, aby nebylo možno se do relé dostat. Průhledné relátko je ideální, neboť do něj studenti mohou vidět a vidět, jak pracuje.)
- spínaný proud (co největší)

Při práci se zmíněným relém je potřeba počítat s několika jeho parametry. První parametrem je čas odpadu (3 ms), který je třeba zohlednit při vytváření programu. Druhým parametrem je maximální spínaný proud, který omezuje výběr motorků pro úlohy zmíněné v kapitole 2. [13]

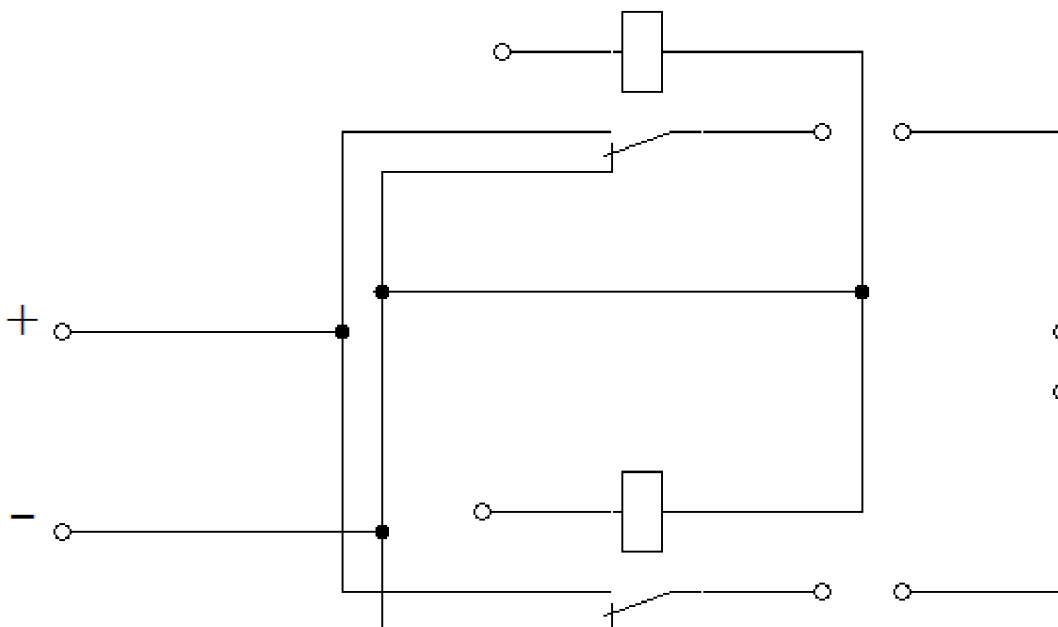


Obrázek 12: Relé FINDER 40.61 [13]

Druhou důležitou součástí potřebnou k vytvoření laboratorní úlohy jsou banánky. Tyto součástky byly vybrány díky své dostupnosti. Rovněž byly vybrány z důvodu snadné práce v laboratořích. Jsou také univerzálními přípojkami a pro práci v laboratořích ideální. V případě mé laboratorní úlohy byly použity červené a černé banánky z důvodu rozlišení polaritu na zdroji. Červené banánky rovněž značí vstup do cívky relátka. Pro laboratorní práci jsem vybral zdířky 4 mm do desky plošných spojů 24.245.1. [13]

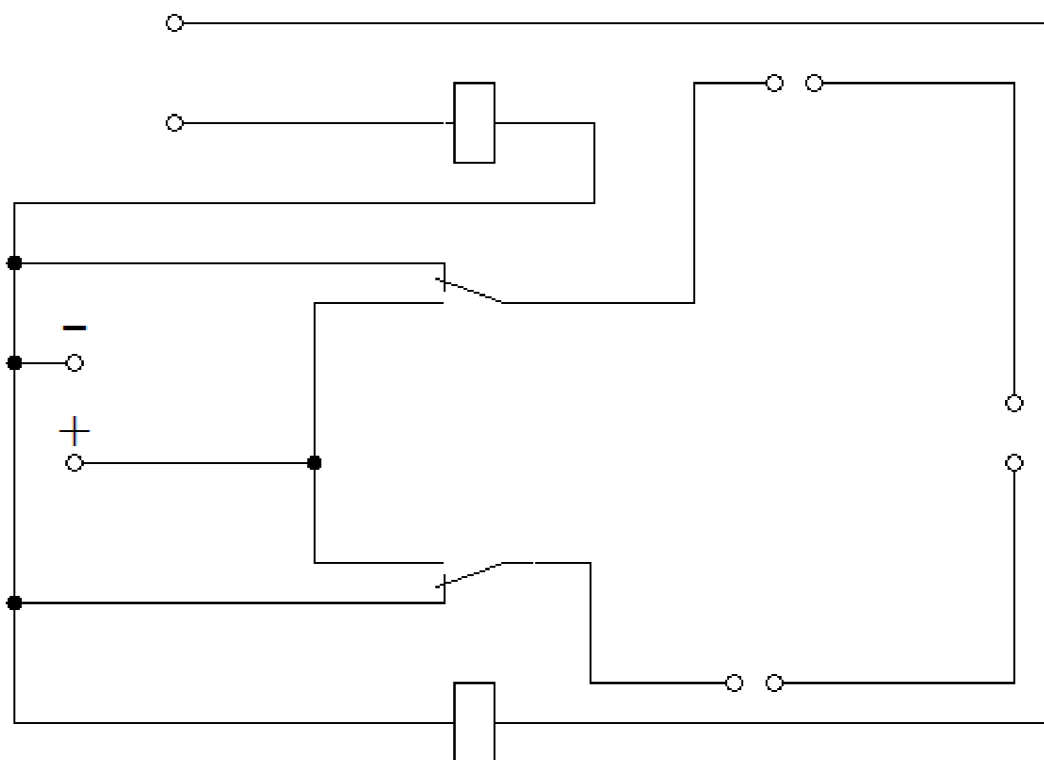


Obrázek 13: Zdířky s průměrem 4 mm (červená a černá) [13]



Obrázek 14: Schéma zapojení před úpravou pro DPS

Součástí přípravy byl návrh schématu. Schéma je založeno na potřebách úloh zmíněných v kapitole 2. Na jejich základě bylo navrženo schéma ukázané na obrázku 14. Cílem bylo vytvořit jednotný přípravek pro všechny tři úlohy s možností obměny součástek pro potřeby jednotlivých úloh. Společnými komponenty jsou pouze relé, které budou osazeny přímo na desce, a zdiřky, které umožňují obměnu dalších komponentů.



Obrázek 15: Schéma zapojení upravené pro DPS (pohled ze strany součástek)

Schéma zobrazené na obrázku 15 je upravené pro načrtnutí na DPS. Cílem bylo schéma zakrestlit tak, aby se jednotlivé cesty nepřekrývaly a nebylo nutné použít propojky, které by se mohli stát slabým místem. Tyto úpravy lze provádět ručně nebo lze použít specializovaný software (např. Eagle). Vzhledem k poměrně jednoduchému zapojení byly úpravy provedeny ručně.

## 4. VÝROBA DPS A OSAZENÍ SOUČÁSTEK

V této kapitole se budeme věnovat výrobě přípravku pro laboratorní úlohy s PLC zmíněné v kapitole 2. Cílem tohoto přípravku je možnost ovládní směru proudu. Tento přípravek bude zapojen k samotnému PLC. Bude obsahovat několik zdířek do desky plošných zdrojů, ke kterým bude možno připojit další zařízení, například měřicí přístroje (ampérmetr) a pasivní součástky (rezistory). Dále se zde objevují dvě přepínací relé (v kombinaci slouží k přepínání směru proudu).

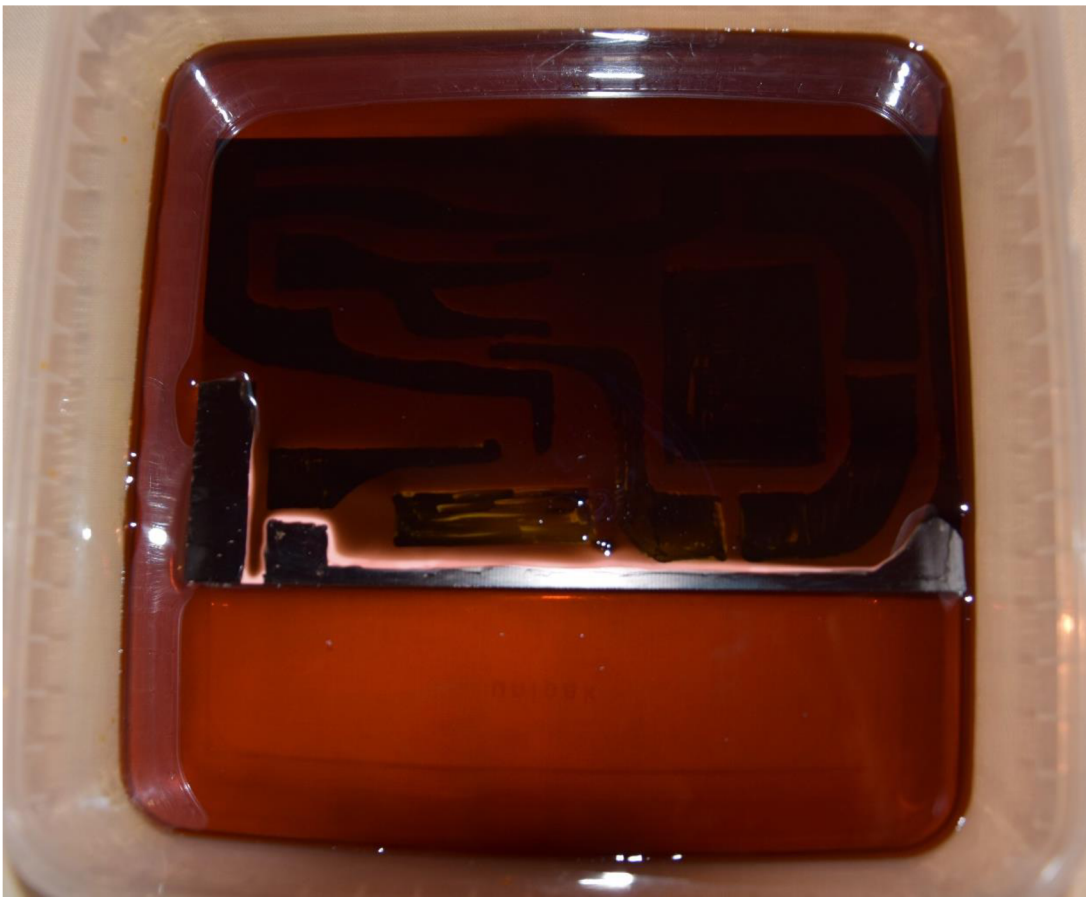
Na obrázku 16 se nachází deska plošných spojů, na které je nakresleno schéma zapojení, upraveno pro desku plošných spojů. Náčrt je proveden lihovým fixem, protože nereaguje s chloridem železitým, čímž chrání měď pod inkoustem.



Obrázek 16: Schéma zakreslené na desce plošných spojů

Následující obrázek ukazuje desku plošných spojů ponořenou do latoru s chloridem železitým ( $\text{FeCl}_3$ ). Z desky byla působením chloridu železitého odstraněna nezakrytá měď, díky čemuž zůstalo přítomno pouze samotné zapojení.





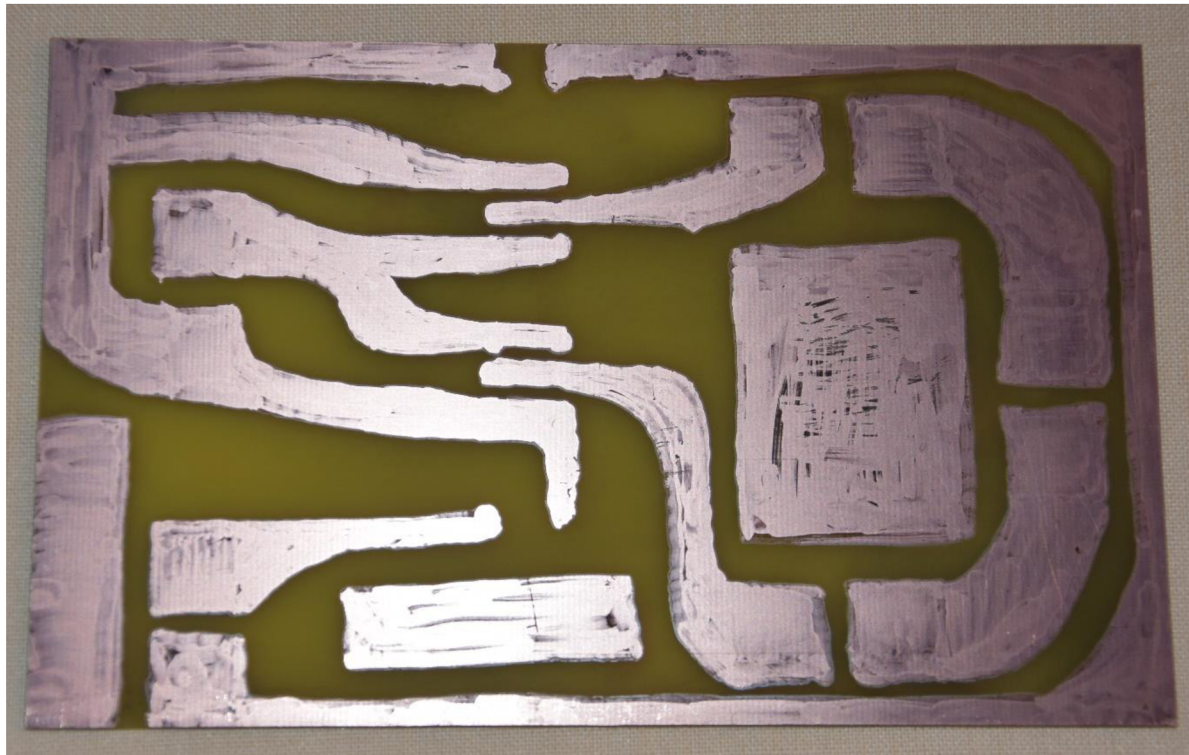
Obrázek 17: DPS ponořená do chloridu železitého

Leptání DPS trvalo přibližně 25 minut a jeho výsledek je možno vidět na obrázku 18. Pro leptání byl použit chlorid železitý, jak již bylo zmíněno výše. Chlorid železitý je



Obrázek 18: Výsledek po vyleptání DPS chloridem železitým

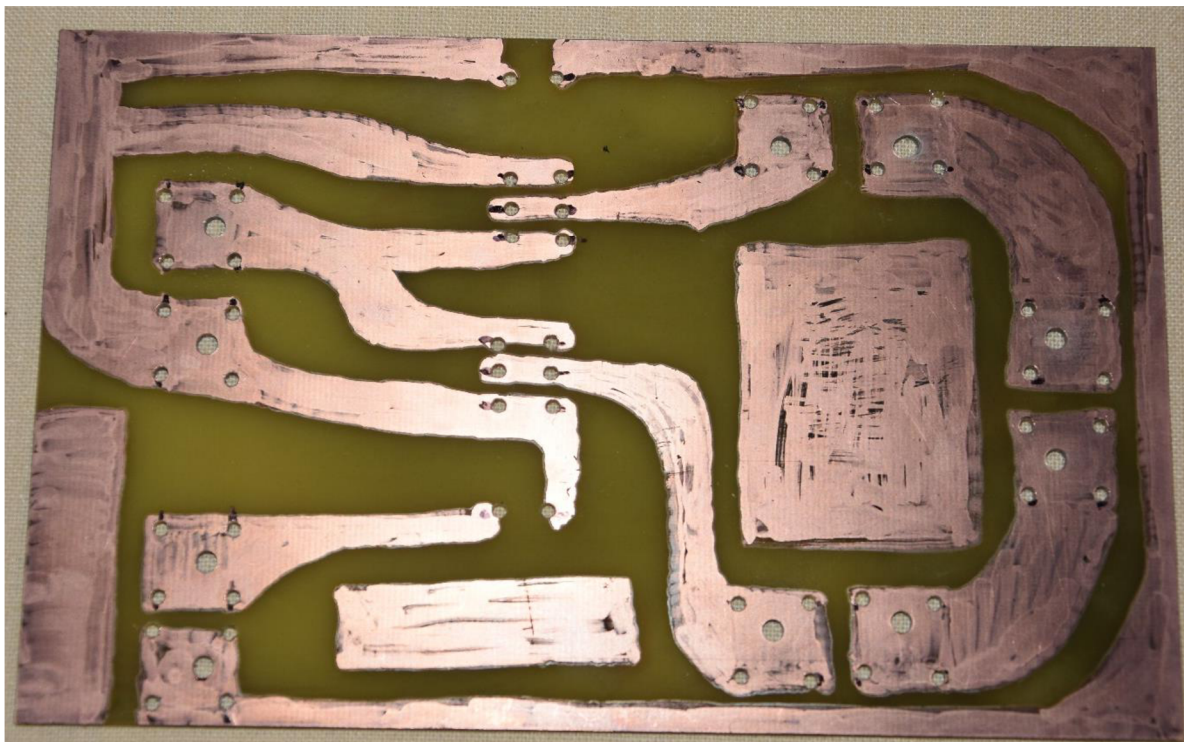
žiravá látka, která odstraní nezakrytou měď z desky. Tento roztok je vhodný pro použití amatéry kvůli snadnému použití a relativní bezpečnosti (krátkodobý styk s kůží se neprojeví).



Obrázek 19: DPS po odstranění inkoustu

Následující obrázek zobrazuje desku po působení chloridem železitým, přičemž byl použit technický líh pro odstranění inkoustu. Na desce je možno vidět nedokonalosti způsobené nerovnoměrným rozptřením inkoustu.





Obrázek 20: DPS po vyvrtání dírek

Na obrázku 20 je možné vidět desku po vyvrtání děr sloužících pro budoucí osazení součástek. Vyvrtané dírky mají průměr 1,5 a 2 mm. Tyto průměry byly zvolené vzhledem k velikosti kontaktů jednotlivých součástek a také kvůli nepřesnostem způsobené lidským faktorem.

Obrázek 21 ukazuje finální stav desky. Deska je osazena dvěma relé a deseti zdičkami pro připojení PLC a dalších prvků, které mohou být potřeba pro danou úlohu. Těla součástek jsou osazena na opačné straně, aby nebránili pájení kontaktů.



Obrázek 21: DPS osazená součástkami

## 5. 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.

V teoretické části bakalářské byly představeny kapitoly týkající se programování logických automatů. Zejména důležitou součástí tvořily programovací jazyky. Byly představeny textové i grafické jazyky, které slouží k vytváření programu PLC. Díky těmto jazykům mohou studenti vytvářet programy ve svých laboratorních úlohách.

Teoreticky a prakticky byly představeny 3 laboratorní úlohy, přičemž tyto úlohy slouží k prohloubení znalostí studentů, kteří na nich budou pracovat. První úloha spočívala v seznámení studentů s elektrickými a mechanickými veličinami souvisejícími s pohonem. Byl zde rovněž ukázan způsob řízení motoru se svým kladem i zápolem.

Druhá úloha spočívala v měření zatěžovací charakteristiky, kdy se veličiny měnily v závislosti na zatížení motoru. První i druhá úloha spočívala v zpracování dat, měření hodnot a nanášení grafů.

Třetí úloha se jako jediná věnovala dynamickému procesu průběhu proudu. Tato úloha měla hlavně demonstrativní účel, přičemž ukazovala potřebu řešit příliš velké hodnoty zapínacích proudů.

Poslední dvě kapitoly sloužily k přípravě výroby a výrobě přípravku, který zpracovává dříve zmíněné 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](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](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](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/](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](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] GM Electronic [online]. [cit. 2021-5-27]. Dostupné z: <https://www.gme.cz/>
- [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](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/>