

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



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

ŘÍZENÍ KROKOVÝCH MOTORŮ POMOCÍ PROGRAMOVATELNÉHO LOGICKÉHO AUTOMATU

CONTROL OF STEPPER MOTORS USING A PROGRAMMABLE LOGIC CONTROLLER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jan Johec

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 Johech

ID: 211113

Ročník: 3

Akademický rok: 2020/21

NÁZEV TÉMATU:

Řízení krokových motorů pomocí programovatelného logického automatu

POKYNY PRO VYPRACOVÁNÍ:

1. Proveďte přehled programovacích možností PLC Schneider Electric v prostředí SoMachine.
2. Realizujte přídatný modul s obvodem MP6500 pro řízení krokového motoru, navržený v rámci SP.
3. Vytvořte laboratorní návod na vytvoření ovládacího software realizujícího zvolené funkce.

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

Tato bakalářská práce se zabývá krokovými motory a jejich řízením pomocí programovatelných automatů. Cílem práce je seznámit se s krokovými motory a dodaným programovatelným automatem, navrhnout a vyrobit přídatný modul potřebný pro ovládání krokových motorů, navrhnout výukovou úlohu a následně vytvořit experimentální verzi ovládacího programu.

Klíčová slova

Krokový motor, PLC, řízení pomocí PLC, SoMachine

Abstract

This bachelor thesis deals with stepper motors and their control using programmable logic controllers. The aim of the work is to get acquainted with stepper motors and the supplied programmable controller, to design and make an additional equipment necessary for the control of stepper motors, to design a teaching task and subsequently to create an experimental version of the control program.

Keywords

Stepper motor, PLC, PLC control, SoMachine

Bibliografická citace

JOCHEC, Jan. *Řízení krokových motorů pomocí programovatelného logického automatu*. Brno, 2021. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/133070>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav výkonové elektrotechniky a elektroniky. Vedoucí práce Dalibor Červinka.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta: Jan Johec

VUT ID studenta: 211113

Typ práce: Bakalářská práce

Akademický rok: 2020/21

Téma závěrečné práce: Řízení krokových motorů pomocí programovatelného logického automatu

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

Děkuji vedoucímu bakalářské práce Ing. Daliboru Červinkovi, Ph.D. za odbornou pomoc a rady při zpracování mé bakalářské práce.

V Brně dne: 27. května 2021

podpis autora

Obsah

SEZNAM OBRÁZKŮ	9
SEZNAM TABULEK.....	10
ÚVOD	11
1. KROKOVÉ MOTORY.....	12
1.1 STAVBA KROKOVÉHO MOTORU	12
1.2 TYPY KROKOVÝCH MOTORŮ	13
1.2.1 Pasivní krokové motory.....	13
1.2.2 Aktivní krokové motory	13
1.2.3 Hybridní krokové motory.....	14
1.2.4 Lineární krokové motory.....	14
1.3 ZPŮSOBY ŘÍZENÍ KROKOVÝCH MOTORŮ	15
1.3.1 Unipolární jednofázové řízení.....	15
1.3.2 Unipolární dvoufázové řízení.....	15
1.3.3 Bipolární jednofázové řízení	16
1.3.4 Bipolární dvoufázové řízení	16
1.3.5 Řízení s polovičním krokem.....	16
2. PROGRAMOVATELNÉ LOGICKÉ AUTOMATY.....	18
2.1 ROZDĚLENÍ PLC	18
2.1.1 Kompaktní PLC.....	18
2.1.2 Modulární PLC.....	19
2.2 ZPŮSOBY PROGRAMOVÁNÍ PLC	20
2.2.1 Jazyk strukturovaného textu (ST).....	20
2.2.2 Jazyk funkčního blokového schématu (FBD)	20
2.2.3 Sekvenční funkční diagram (SFC).....	21
2.2.4 Jazyk seznamu instrukcí (IL).....	21
2.2.5 Jazyk příčkového diagramu (LD).....	22
3. NÁVRH ZAŘÍZENÍ PRO LABORATORNÍ ÚLOHU.....	23
3.1 POUŽITÉ SOUČÁSTKY	23
3.1.1 PLC řady Modicon M241	23
3.1.2 Vývojová destička Pololu-2967.....	23
3.1.3 Krokový motor HY-200-1713-095 B4	24
3.1.4 Zdroj napětí EPS-45-24-C	25
3.2 ZAPOJENÍ ÚLOHY.....	26
4. NÁVRH LABORATORNÍ ÚLOHY.....	28
4.1 BLOKOVÉ SCHÉMA OVLÁDACÍHO PROGRAMU	29
5. VÝVOJOVÉ PROSTŘEDÍ SOMACHINE	30
5.1 SOMACHINE BASIC	31
5.2 VIJE0 DESIGNER	31
5.3 LOGIC BUILDER.....	32
.....	32

6.	DEMONSTRAČNÍ PŘÍPRAVEK K LABORATORNÍ ÚLOZE	34
6.1	DEMONSTRAČNÍ PŘÍPRAVEK.....	34
6.2	PŘÍDAVNÝ MODUL S OBVODEM MP6500.....	35
6.3	ZAPOJENÍ PRACOVIŠTĚ PRO LABORATORNÍ ÚLOHU	36
	37
7.	MĚŘENÍ NA LABORATORNÍM PŘÍPRAVKU	38
7.1	NASTAVENÍ PROUDOVÉHO OMEZENÍ	38
7.2	MĚŘENÍ PRŮBĚHŮ PROUDU.....	38
	39
	39
	40
8.	SOFTWAREVÁ ČÁST LABORATORNÍ ÚLOHY	41
8.1	FUNKCE PROGRAMU	41
8.2	LABORATORNÍ NÁVOD	43
9.	ZÁVĚR.....	46
	LITERATURA.....	47
	SEZNAM SYMBOLŮ A ZKRATEK	49
	SEZNAM PŘÍLOH.....	50

SEZNAM OBRÁZKŮ

1.1	Rotor krokového motoru [3].....	12
1.2	Stator krokového motoru [3]	12
1.3	Krokový motor jako celek [3]	13
1.4	Řez pasivním krokovým motorem [4].....	13
1.5	Řez aktivním krokovým motorem [4].....	14
1.6	Řez hybridním krokovým motorem [4].....	14
1.7	Lineární krokový motor, upraveno z [5]	15
1.8	Kroky při unipolárním jednofázovém řízení [3].....	15
1.9	Kroky při unipolárním dvoufázovém řízení [3]	16
1.10	Kroky při bipolárním jednofázovém řízení [3].....	16
1.11	Kroky při bipolárním dvoufázovém řízení [3]	16
2.1	Princip uspořádání kompaktního PLC [7].....	19
2.2	Princip uspořádání modulárního PLC [7].....	19
2.3	Zápis jazyku ST [7].....	20
2.4	Zápis jazyku FBD [7].....	21
2.5	Zápis jazyku SFC [7].....	21
2.6	Zápis jazyku IL [7].....	22
2.7	Zápis jazyku LD [7]	22
3.1	PLC TM241CEC24T [9].....	23
3.2	Jednotka řízení motoru Pololu-2967 [10].....	24
3.3	Vnitřní schéma destičky Pololu-2967 [10].....	24
3.4	Motor HY-200-1713-095 B4.....	25
3.5	Zdroj EPS-45-24-C [11].....	25
3.6	Uspořádání komponentů	26
3.7	Schéma zapojení.....	27
4.1	Mechanická sestava demonstračního přípravku	28
4.2	Blokové schéma programu	29
5.1	SoMachine Central	30
5.2	Otevřený projekt.....	31
5.3	Logic Builder	32
6.1	Optická závora [17].....	34
6.2	Demonstrační přípravek	35
6.3	Prototyp přídatného modulu	36
6.4	Pracoviště pro laboratorní úlohu.	37
7.1	Proud fází motoru.....	39
7.2	Náběžná hrana proudového pulzu	39
7.3	Detail zvlnění proudu	40
8.1	Příklad signálů generovaných PTO [13]	41
7.2	Funkční blok MC_MoveVelocity_PTO	42
8.3	Schéma zapojení laboratorní úlohy	44

SEZNAM TABULEK

6.1	Režimy krokování	35
-----	------------------------	----

ÚVOD

V dnešní době jsou elektrické motory jedny z nejpoužívanějších strojů našich životů. Můžeme je najít v mnoha zařízeních kolem nás. Od aut a vlaků až po tiskárnu nebo třeba zubní kartáček.

Elektrické motory mají mnoho různých podob, z nichž každá je vhodná pro jinou aplikaci. Jedním druhem jsou právě krokové motory, které se uplatňují zejména v případech, kdy potřebujeme přesně provádět změnu polohy.

K řízení krokových motorů se často používají jednotky PLC. Jsou to v podstatě malé počítače s mnoha vstupy a výstupy, které je snadné naprogramovat pro mnoho různých způsobů použití.

Práce je členěná do osmi kapitol, z nichž prvních pět je spíše teoretických a poslední tři jsou praktické a věnují se realizaci návrhů z předešlých kapitol. První kapitola se zabývá krokovými motory. Popisuje jejich funkci stavbu, různé druhy provedení a také možnosti jejich řízení. Druhá kapitola se pak věnuje PLC jednotkám, jejich rozdělení a způsobům programování.

Třetí a čtvrtá kapitola je zaměřena na návrh laboratorní úlohy a demonstračním přípravkům pro úlohu. Třetí kapitola je věnována doplnění PLC jednotky Modicon M241 o přídatné zařízení a návrhu přípravku pro laboratorní úlohu. Čtvrtá kapitola už se pak zabývá přímo návrhem laboratorní úlohy a struktury programu pro následné řízení motorů.

Pátá kapitola popisuje programovací možnosti vývojového prostředí SoMachine. Jsou zde stručně popsány jednotlivé softwary z balíčku SoMachine. Podrobněji je zde popsána část Logic Builder z balíčku SoMachine, která byla použita pro vytvoření ovládacího programu k laboratorní úloze.

Šestou kapitolu tvoří realizace hardwarového zařízení pro laboratorní úlohu. Je zde popsán demonstrační přípravek s pohyblivým ramenem a také přídatný modul s obvodem MP6500 pro řízení krokových motorů. V sedmé kapitole jsou pak zobrazeny výsledky měření proudů na demonstračním přípravku.

V poslední osmé kapitole je pak popsána funkce řídicího programu pro demonstrační přípravek a samotný návod pro laboratorní úlohu. Ten obsahuje zadání úlohy, schéma jejího zapojení a také podrobný postup pro její vypracování.

1. KROKOVÉ MOTORY

Krokový motor je impulzně napájený synchronní stroj, jehož pohyb není spojitý, ale probíhá v jednotlivých krocích. Při napájení cívek ve statoru se vytvoří elektromagnetické pole, které přitáhne opačný pól magnetu na rotoru a tím motor vykoná jeden krok. Krokové motory nacházejí využití v případech, kde potřebujeme přesně nastavovat polohu anebo v ní setrvat i v případě působení vnějších sil.

Díky principu, jakým krokové motory fungují, v nich dochází k minimálnímu opotřebování a tím pádem jsou velmi jednoduché na údržbu, jsou mechanicky odolné a mají dlouhou životnost. Mezi nevýhody patří stálý odběr elektrické energie i v případě, že se rotor neotáčí. Rychlost krokových motorů musí být také omezena, aby nedocházelo k takzvanému ztrácení kroků. Ke ztrátě kroku může také docházet při nadměrném zatížení krokového motoru. [1]

1.1 Stavba krokového motoru

Krokové motory jsou tvořeny dvěma hlavními částmi – rotorem a státorem. Na obrázku 1.1 můžeme vidět rotor krokového motoru. Ten je tvořen permanentními magnety připevněnými na hřídeli. Počet magnetů udává počet kroků, který motor vykoná na jednu otáčku. Celý rotor je pak uložen v kuličkových ložiskách, které umožňují jeho otáčení. Tyto ložiska jsou jediné místo, v kterém u krokových motorů dochází k mechanickému opotřebování.

Další část krokového motoru tvoří stator, který je zobrazen na obrázku 1.2. Stator je tvořen izolovanými plechy, na kterých jsou navinuté cívky. Pólové nástavce statoru jsou vroubkovány se stejnou roztečí, s jakou jsou rozmístěny magnety na rotoru.

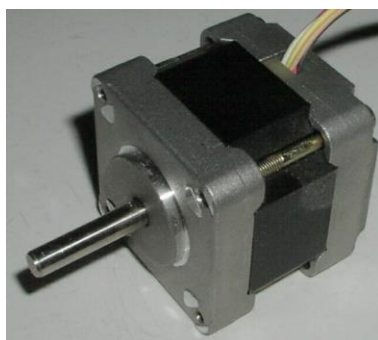
Na obrázku 1.3 pak můžeme vidět už zkompletovaný krokový motor. Krom rotoru a statoru zde přibyló uložení pro rotor a všechny části jsou k sobě přichyceny čtyřmi šrouby. [1] [3]



Obrázek 1.1 Rotor krokového motoru [3]



Obrázek 1.2 Stator krokového motoru [3]

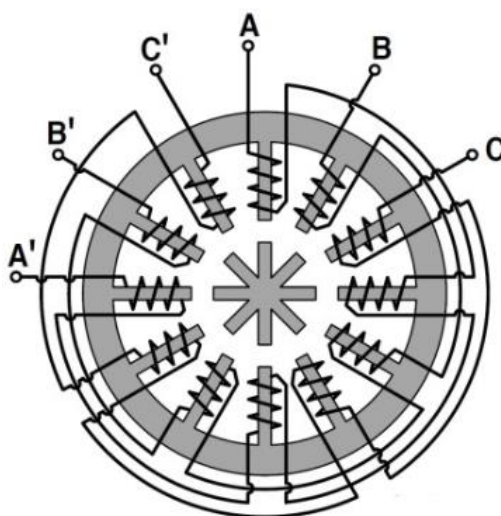


Obrázek 1.3 Krokový motor jako celek [3]

1.2 Typy krokových motorů

1.2.1 Pasivní krokové motory

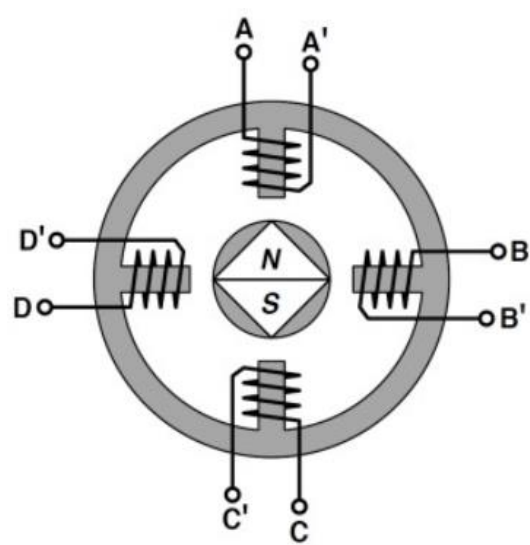
Tento typ krokových motorů pracuje na principu rozdílné magnetické vodivosti rotoru a statoru, které jsou ve většině případů složeny z ocelových plechů. Nejsou zde tedy použity žádné permanentní magnety. Nutnou podmínkou pro jejich funkci je rozdílný počet zubů na rotoru a statoru. [1] [4]



Obrázek 1.4 Řez pasivním krokovým motorem [4]

1.2.2 Aktivní krokové motory

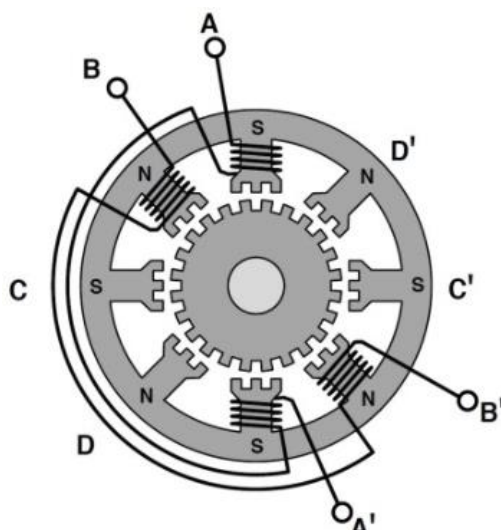
Jsou nejběžnějším typem krokových motorů. Jejich rotor se skládá z permanentních magnetů. Magnety rotoru se přitahují nebo odpuzují s vytvořeným magnetickým polem cívky s pólovými nastavci. [1] [4]



Obrázek 1.5 Řez aktivním krokovým motorem [4]

1.2.3 Hybridní krokové motory

Jak už název napovídá, jedná se o kombinaci aktivního a pasivního krokového motoru. Jejich rotor je tvořen axiálně uloženými permanentními magnety na jejichž pólech jsou umístěny feromagnetické pólové nástavce. Hybridní krokové motory mají pouze základní krok, ale zato mají oproti jiným typům krokových motorů větší točivý i přídržný moment. [1] [4]



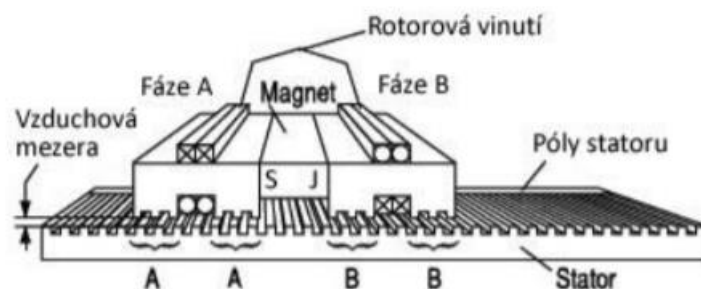
Obrázek 1.6 Řez hybridním krokovým motorem [4]

1.2.4 Lineární krokové motory

Tento typ krokových motorů se od ostatních typů odlišuje tím, že může konat pouze

neperiodický posuvný pohyb na jednu nebo druhou stranu a má omezenou dráhu pohybu.

Stator je rozložen do úsečky a je tvořen permanentními magnety. Stator tedy slouží jako dráha a nad ním se posouvá rotor s cívkami. [1] [5]



Obrázek 1.7 Lineární krokový motor, upraveno z [5]

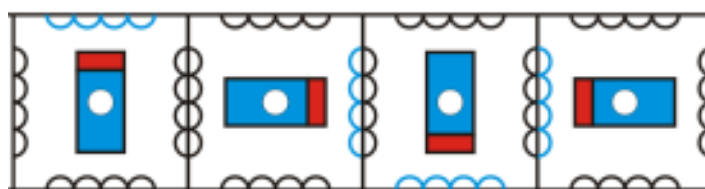
1.3 Způsoby řízení krokových motorů

Existuje mnoho způsobů řízení krokových motorů. Základními způsoby jsou jednofázové/dvoufázové řízení a unipolární/bipolární řízení. Pro detailní pochopení řízení krokových motorů jsou v následujících kapitolách popsány jednotlivé kombinace řízení.

Pro názornost je uvažován zjednodušený krokový motor se čtyřmi cívkami na statoru a jedním permanentním magnetem s dvěma póly na rotoru. Motor má tedy na otáčku pouze čtyři kroky. Cívka, kterou prochází proud je vždy na obrázku vyznačena červeně nebo modře podle polarit magnetického pole, kterým působí na rotor. [2] [3]

1.3.1 Unipolární jednofázové řízení

U unipolárního řízení prochází proud vždy jen jednou cívkou statoru v daný okamžik. Motor má při tomto způsobu nejmenší spotřebu, ale také nejmenší kroučící moment.

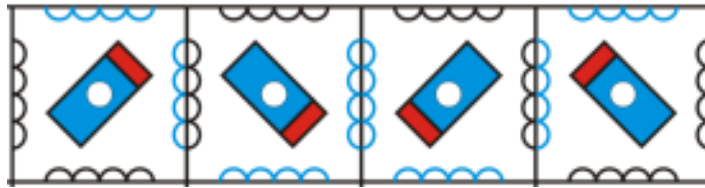


Obrázek 1.8 Kroky při unipolárním jednofázovém řízení [3]

1.3.2 Unipolární dvoufázové řízení

Při unipolárním dvoufázovém řízení vytvářejí vždy dvě sousední cívky souhlasně

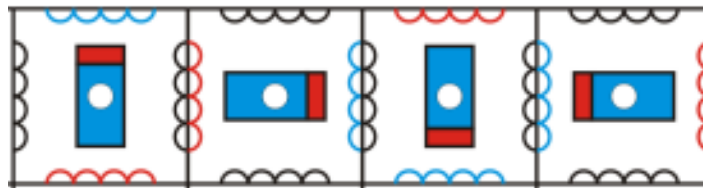
orientované magnetické pole. Díky tomu dosáhneme vyššího kroučícího momentu krokového motoru, avšak za cenu dvojnásobné spotřeby elektrické energie.



Obrázek 1.9 Kroky při unipolárním dvoufázovém řízení [3]

1.3.3 Bipolární jednofázové řízení

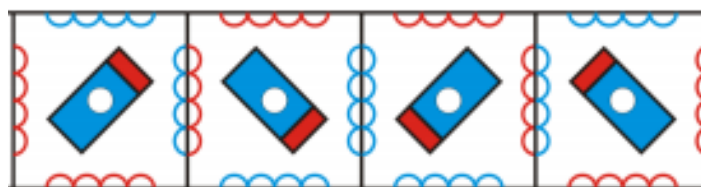
V tomto případě prochází proud v daný okamžik vždy dvěma protilehlými cívkami. Ty jsou zapojeny tak, aby generovaly navzájem opačně orientované magnetické pole. Tento způsob umožňuje zase o něco vyšší kroučící moment, ale spotřeba elektrické energie stoupá.



Obrázek 1.10 Kroky při bipolárním jednofázovém řízení [3]

1.3.4 Bipolární dvoufázové řízení

U tohoto typu řízení vždy proud prochází čtyřmi cívkami zároveň. Dvě sousední cívkami vytváří shodně orientované magnetické pole a dvě protilehlé cívkami k nim v ten samý okamžik tvoří vzájemně opačně orientované magnetické pole. Tím pádem dosáhneme ještě většího kroučícího momentu, ale také ještě větší spotřeby elektrické energie.



Obrázek 1.11 Kroky při bipolárním dvoufázovém řízení [3]

1.3.5 Řízení s polovičním krokem

Zvýšení přesnosti polohování krokového motoru můžeme docílit použitím polovičních

kroků namísto celých kroků. Jak můžeme už z názvu tušit, krokový motor pak vykoná na jednu otáčku dvakrát více kroků. Tím pádem jej můžeme polohovat s větší přesností.

Polovičního kroku dosáhneme tak, že při řízení střídáme jednofázové a dvoufázové řízení krokového motoru. Tímto způsobem lze řídit jak unipolární, tak i bipolární krokové motory, přičemž platí, že bipolární budou dosahovat většího kroutícího momentu.

Je-li potřeba ještě větší přesnosti při polohování, je možno využít mikrokrokování. V tomto případě se kroky dělí na ještě menší díly. Toho dosáhneme tak, že v jedné fázi postupně snižujeme proud a v druhé jej zvyšujeme. Pro regulaci proudu pro mikrokrokování se využívá pulzní šířková modulace. [2] [3]

2. PROGRAMOVATELNÉ LOGICKÉ AUTOMATY

V dnešním světě se čím dál více klade důraz na automatické ovládání zařízení a aplikací. Právě pro tyto účely nám slouží krom počítačů malé řídicí jednotky s různými funkcemi označované jako PLC.

Programovatelné logické automaty též známé jako PLC (z anglického Programmable Logic Controller) jsou malé, lehce programovatelné řídicí jednotky s mnoha vstupy a výstupy. Díky tomu je snadné k nim připojit i několik senzorů, displejů, tlačítek, spínačů, motorů a dalších zařízení. Cílem PLC je nahradit původní reléové automaty. Je u nich totiž jednoduché v případě potřeby kdykoliv upravit řídicí program, což byl u reléových systémů často velký a někdy i neřešitelný problém.

Funkce celého PLC i ovládání připojených prvků je řízeno uloženým programem, který lze většinou snadno vytvořit ve výrobcem dodávaném grafickém vývojovém softwaru, který umožňuje i grafickou simulaci naprogramovaného programu a po připojení PLC i reálné testování. Některé druhy PLC dokonce umožňují programování přímo na displeji samotného zařízení.

Uvedení PLC do provozu tedy vyžaduje naprogramování řídicího programu, což je ta nejsložitější část. Po nahrání programu do PLC už jen stačí nasadit modul na DIN lištu, připojit vodiče do svorek pro vstupy a výstupy a vše je připraveno k použití. Některé dnešní PLC pak umí připojená zařízení ovládat i bezdrátově. [6] [12]

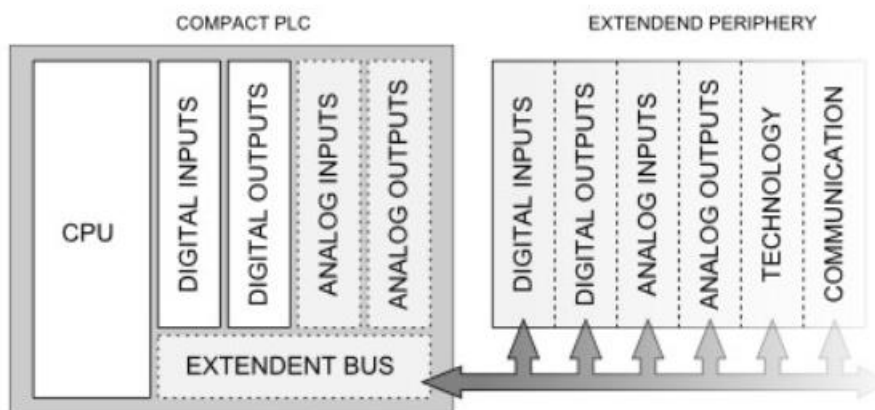
2.1 Rozdělení PLC

Jednotky PLC jsou vyráběny ve dvou základních variantách, a to modulární a kompaktní.

Rozdíly mezi nimi jsou popsány v následujících podkapitolách.

2.1.1 Kompaktní PLC

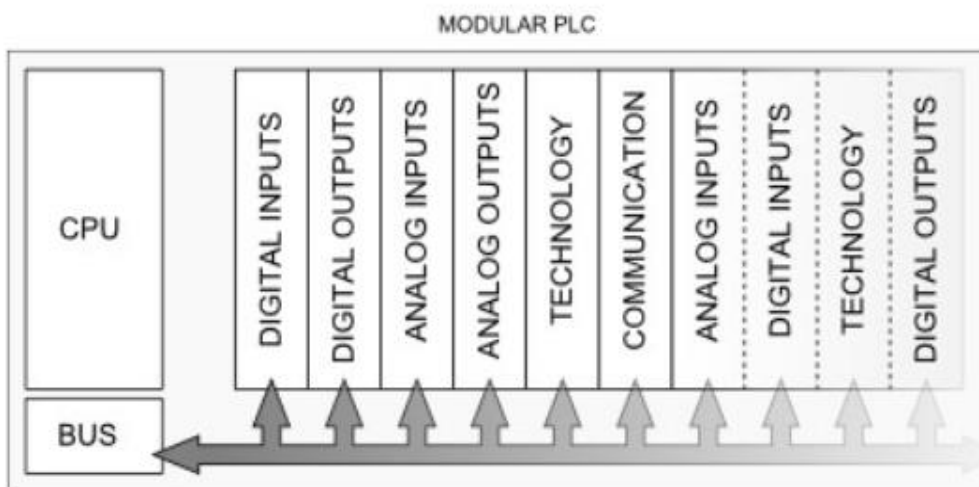
Tento typ PLC obsahuje v jednom modulu vše potřebné k jeho provozu. Základem je CPU, dále pak analogové a digitální vstupy a výstupy, podpora pro komunikaci a někdy i zdroj. Rozšiřitelnost kompaktních systémů je většinou omezena na pouhé přidávání modulů s pevným počtem vstupů a výstupů. Některé mohou obsahovat i displej, na kterém lze monitorovat stav programu nebo ho i celý naprogramovat. Výhodou oproti modulárním systémům je nízká cena a menší velikost. [7] [12]



Obrázek 2.1 Princip uspořádání kompaktního PLC [7]

2.1.2 Modulární PLC

Jak už název napovídá, modulární PLC se sestavují z různých modulů s rozlišnými funkcemi. Základním modulem bývá modul s CPU a k němu se pak volí další moduly potřebné pro požadovanou aplikaci. Díky modulárnosti je tedy možné poskládat PLC jednotku přímo na míru pro řízený systém. Ceny modulárních PLC se liší podle jejich funkce a velikosti. Cena sestavené jednotky se pak může klidně vyšplhat i na stovky tisíc Kč. [7] [12]



Obrázek 2.2 Princip uspořádání modulárního PLC [7]

2.2 Způsoby programování PLC

Jak už bylo zmíněno v předchozích kapitolách, existuje několik způsobů, kterými je možné jednotky PLC programovat. Od jednoduchých až po složité a komplexnější metody. Každý uživatel si tedy může vybrat metodu vhodnou pro druh jeho aplikace a také pro jeho programovací dovednosti.

Programy pro řízení PLC se nejdříve naprogramují na počítači a poté nahrají do PLC. Pokud se tedy nejedná o druh PLC, která lze programovat přímo pomocí tlačítek na řídicím panelu.

Před započítím programování je vždy dobré si nejdříve definovat úkol, který má soustava plnit. Poté se definují vstupy a výstupy a teprve pak započne samotné programování. Ne všechny PLC jednotky podporují každý programovací jazyk. Je tedy vždy nutné si napřed zjistit, které jazyky PLC podporuje a teprve poté začít programovat. [8] [12]

2.2.1 Jazyk strukturovaného textu (ST)

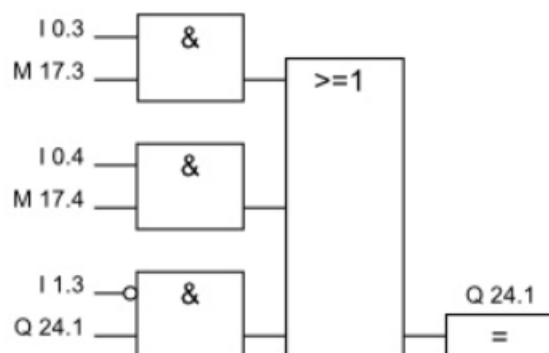
Jedná se o programovací jazyk na vysoké úrovni, který je velmi podobný programování v Pascalu. K definování úkonů PLC se používají výrazy.

```
(* simple state machine *)
TxtState := STATES[StateMachine];
CASE StateMachine OF
  1: ClosingValve();
ELSE
  ;; BadCase();
END_CASE;
```

Obrázek 2.3 Zápis jazyku ST [7]

2.2.2 Jazyk funkčního blokového schématu (FBD)

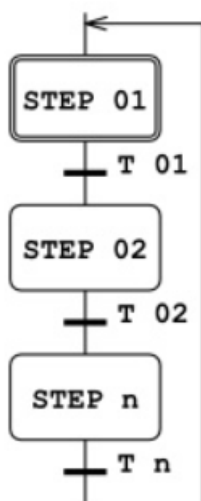
Jazyk FBD funguje na principu grafického znázornění logických bran AND, NAND, OR, NOR a dalších. Jednotlivé brány se pak spojí v požadovaném pořadí a díky tomu je popsána funkce mezi vstupními a výstupními proměnnými.



Obrázek 2.4 Zápís jazyku FBD [7]

2.2.3 Sekvenční funkční diagram (SFC)

Jde o grafický programovací jazyk. Postupuje se zde mezi příkazovými kroky po splnění přechodové podmínky. Program vždy setrvává v konkrétním kroku a opakuje ho do té doby, dokud není splněna přechodová podmínka.



Obrázek 2.5 Zápís jazyku SFC [7]

2.2.4 Jazyk seznamu instrukcí (IL)

Řádkově orientovaný jazyk typu assembler. Jak už z názvu můžeme tušit, výsledný program se skládá ze sekvence instrukcí, z nichž se každá nachází na novém řádku.

```

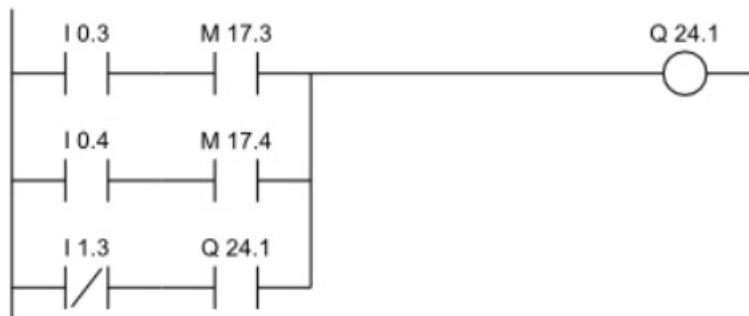
A   I   0.3
A   M   17.3
O
A   I   0.4
A   M   17.4
O
AN  I   1.3
A   Q   24.1
=   Q   24.1

```

Obrázek 2.6 Zázpis jazyku IL [7]

2.2.5 Jazyk příčkového diagramu (LD)

LD je nejpopulárnější programovací jazyk pro PLC. Snaží se napodobit mechanická relé v panelu, které PLC nahrazují. Má dvě svislé kolejnice a mezi nimi řadu horizontálních příček. Ovladač pak takhle naprogramovaný program skenuje zleva doprava a shora dolů.



Obrázek 2.7 Zázpis jazyku LD [7]

3. NÁVRH ZAŘÍZENÍ PRO LABORATORNÍ ÚLOHU

K řízení krokových motorů pomocí PLC je nutné použít také ovládací moduly vhodné pro daný typ krokového motoru. Jelikož cílem této práce je navrhnout laboratorní úlohu na krokových motory, tak zařízení pro úlohu bude také vybaveno několika tlačítky a vypínači, které bude možné naprogramovat na různé funkce pro ovládání krokových motorů.

3.1 Použité součástky

3.1.1 PLC řady Modicon M241

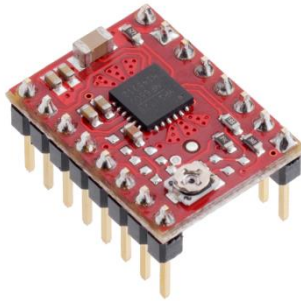
Pro řízení ovladačů krokových motorů bude použita PLC jednotka od výrobce Schneider Electric. Konkrétně model TM241CEC24T. Jedná se o PLC menšího rozměru napájené 24 V DC. Disponuje 14 vstupy a 10 výstupy. Na vstupy je nutné přivést minimální napětí 15 V pro sepnutí na logickou 1. Napětí na výstupech je 24 V DC a jsou schopny poskytnout proud až 0,5 A. [9]



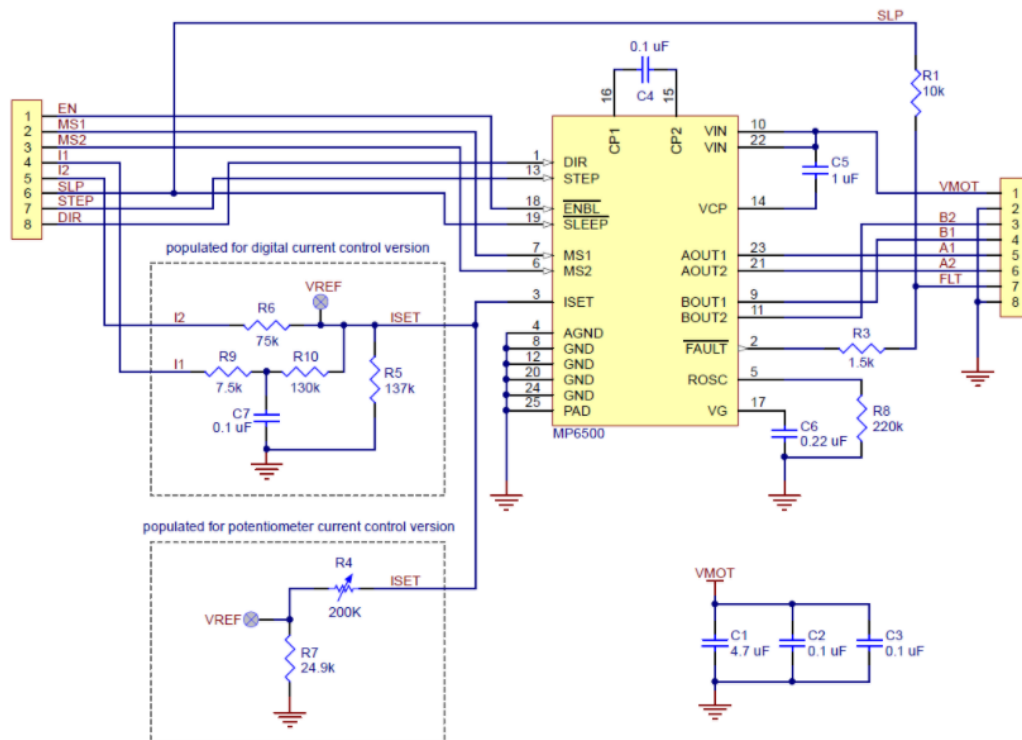
Obrázek 3.1 PLC TM241CEC24T [9]

3.1.2 Vývojová destička Pololu-2967

Krokové motory budou řízeny pomocí Pololu-2967, které je osazeno integrovaným obvodem pro řízení bipolárních krokových motorů MP6500. Je schopen pracovat při napájecím napětí od 4,5 V až do 35 V a dokáže poskytnout proud do motoru až 2,5 A. Jeho největšími výhodami jsou malé rozměry a nízká cena.



Obrázek 3.2 Jednotka řízení motoru Pololu-2967 [10]



Obrázek 3.3 Vnitřní schéma destičky Pololu-2967 [10]

Na obrázku 3.3 je vnitřní schéma destičky Pololu. Její hlavní částí je integrovaný obvod MP6500, který je pro svou správnou funkci doplněn o několik odporů a kondenzátorů. Nachází se zde také potenciometr, kterým se nastavuje výstupní proud do cívky krokového motoru. Je nutné jej správně nastavit, aby nedošlo k proudovému přetížení cívek a jejich následovnému poškození. Integrovaný obvod umožňuje řízení krokových motorů v režimu s celým krokem, tak i v režimu s polovičním, čtvrtinovým i osminovým krokem. [10]

3.1.3 Krokový motor HY-200-1713-095 B4

Krokové motory použité pro výukovou úlohu jsou od výrobce MAE. Jde o bipolární krokové motory, které mají úhel kroku $1,8^\circ$, takže pro vykonání jedné otáčky je

potřeba, aby motor udělal 200 kroků. Odpor jednoho vinutí je 3Ω a každé vinutí je dimenzováno na proud až 0,95 A. Přídržný moment motoru při bipolárním napájení je 23,4 Ncm.



Obrázek 3.4 Motor HY-200-1713-095 B4

3.1.4 Zdroj napětí EPS-45-24-C

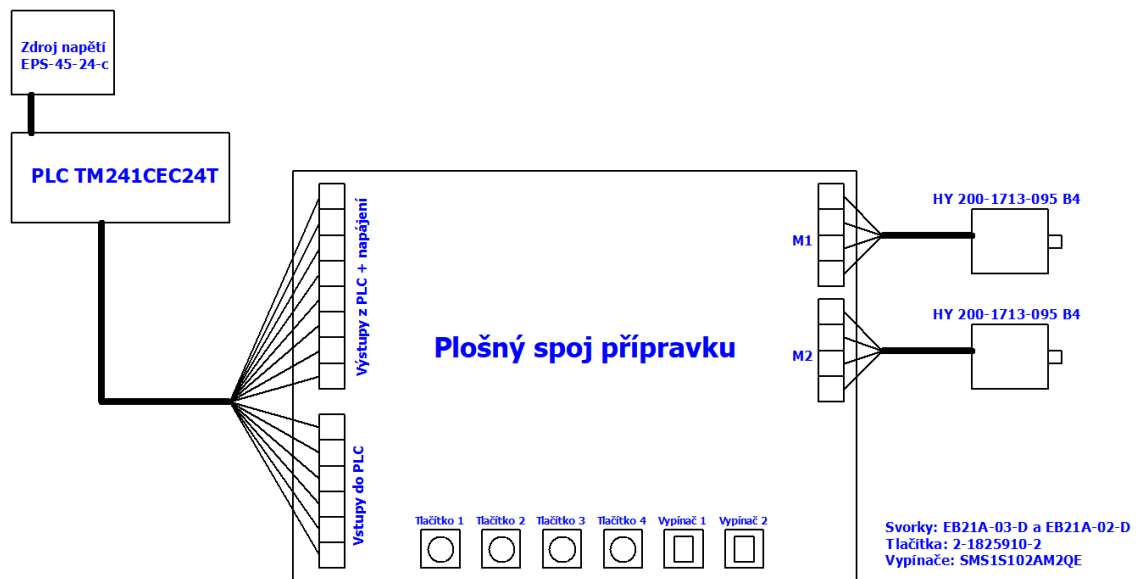
Celá soustava bude napájena ze spínaného zdroje napětí s výstupním napětím regulovatelným od 21,6 V do 27 V DC. Zdroj je schopen poskytnout proud až 1,9 A a jeho skutečný výkon je 45,6 W. [11]



Obrázek 3.5 Zdroj EPS-45-24-C [11]

3.2 Zapojení úlohy

Na obrázku 3.6 je zobrazeno pravděpodobné zapojení laboratorní úlohy. PLC jednotka bude napájena z 24 V zdroje napětí. PLC pak bude propojeno kabely s deskou plošných spojů, na které budou osazeny destičky Pololu-2967 spolu s tlačítky a vypínači. Pomocí tlačítek a vypínačů se budou posílat signály na vstupy PLC jednotky. Ta je vyhodotí a podle nastavení programu pak pošle jednotlivými výstupy signály na destičky Pololu-2967, které budou ovládat krokové motory

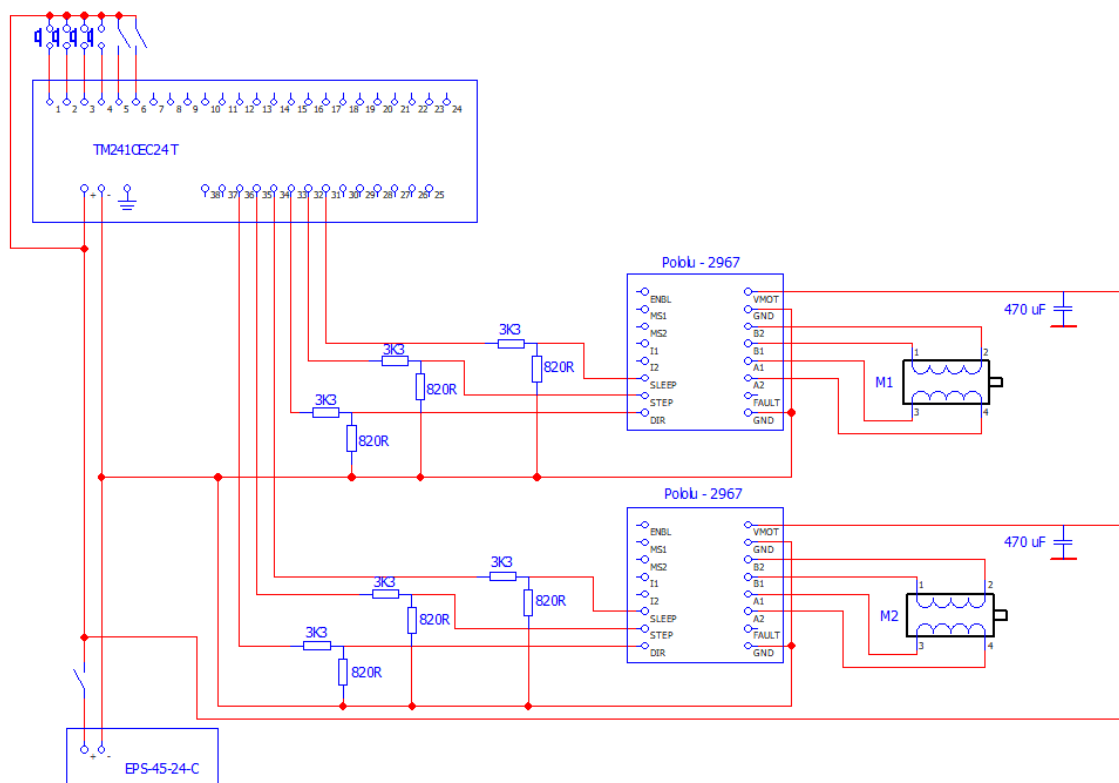


Obrázek 3.6 Uspořádání komponentů

Na obrázku 3.7 už je zobrazeno samotné zapojení komponentů pro laboratorní úlohu. Jelikož PLC jednotka poskytuje na výstupu 24 V a logické piny na destičce Pololu-2967 jsou určeny pro vstupní napětí 5 V, bylo potřeba použít dělič napětí pro výstupní signály z PLC. Ten je realizován dvěma odpory o hodnotách 3300 Ω a 820 Ω . Hodnoty odporů byly určovány tak, aby napětí na vstupech do Pololu-2967 nebylo vyšší než 6 V a zároveň nižší než 2,1 V. Při překročení napěťového rozmezí by totiž mohlo dojít k poškození destičky a při nižším napětí než 2,1 V by zase destička nereagovala na vstupní signál. Zároveň musel být také brán ohled na to, aby přes odpory netekl moc velký proud a ty se zbytečně nepřehřívaly. Při takto navrženém děliči půjde na vstupy destičky napětí 4,78 V a proud 5,8 mA.

Zdroj napětí je společný jak pro PLC jednotku, tak pro motory. Z PLC jednotky jsou přivedeny signály na tři piny destičky Pololu-2967. Signál na pin SLEEP určuje, zda je destička zapnutá či vypnutá. Pokud by tento pin nebyl nastaven na logickou 1, tak by celá destička nebyla schopna provozu. Pin DIR je určen pro udání směru otáčení krokového motoru. Poslední pin STEP pak slouží pro určení, kolik kroků motor vykoná. Bude na něj posílán PWM signál, jehož každá náběžná hrana znamená jeden krok motoru.

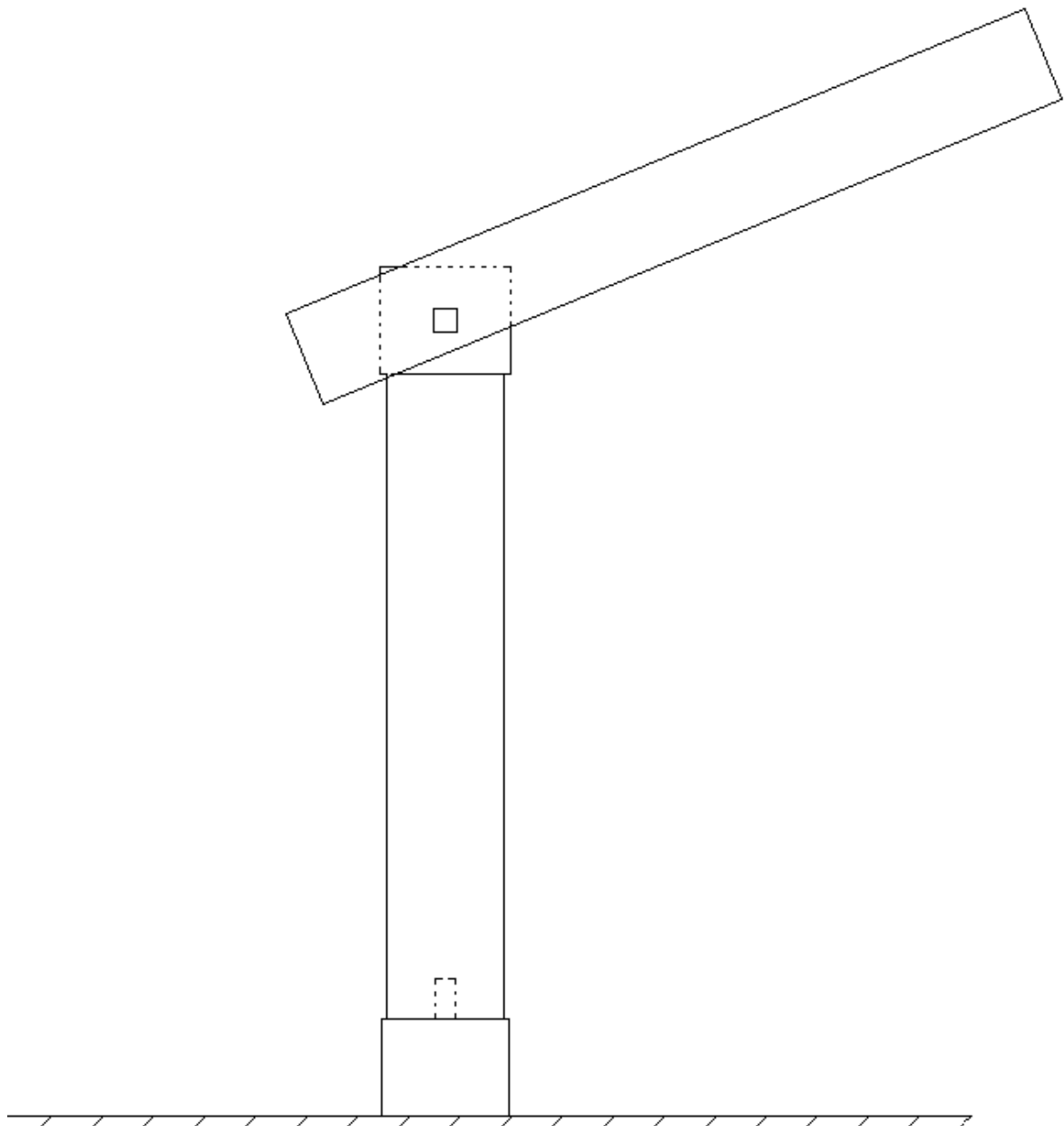
Napětí přivedené na destičku zajišťuje jak napájení destičky samotné, tak i napájení motoru a musí být uzemněno keramickým kondenzátorem.



Obrázek 3.7 Schéma zapojení

4. NÁVRH LABORATORNÍ ÚLOHY

Předmětem laboratorní úlohy bude ovládání krokových motorů díky správnému naprogramování PLC jednotky. Krokové motory budou využity k polohování ramene, jak můžeme vidět na obrázku 4.1. Po naprogramování PLC jednotky bude možno díky tlačítkům a vypínačům umístěným na přípravku navrženém v předchozí kapitole otáčet ramenem kolem základny a pohybovat s ním nahoru a dolů.

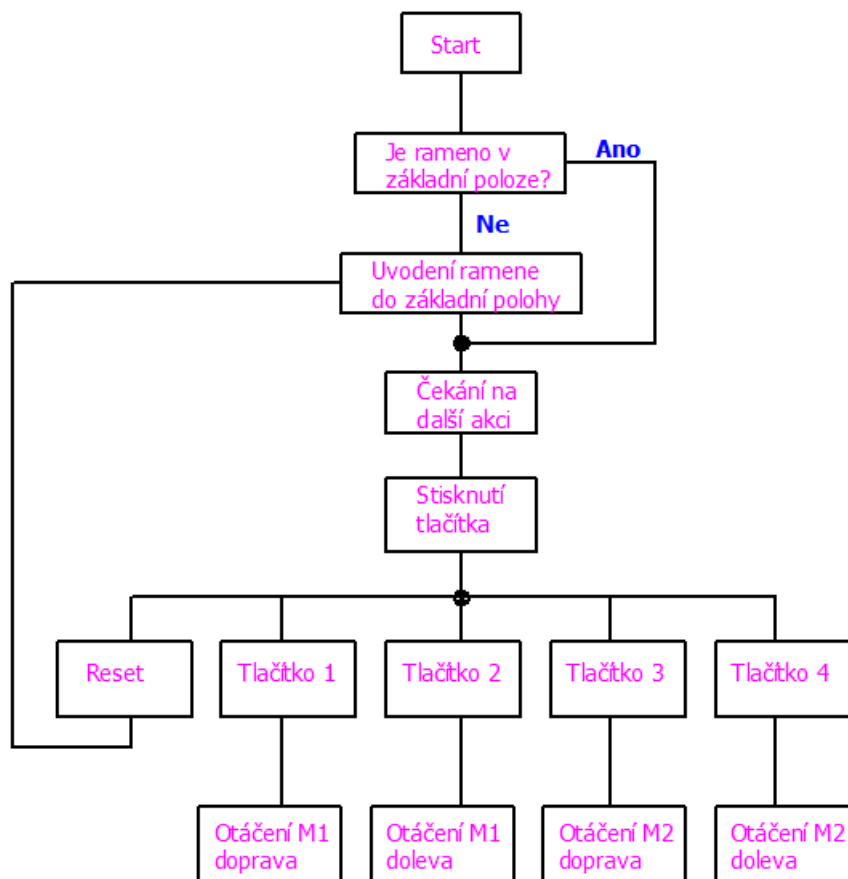


Obrázek 4.1 Mechanická sestava demonstračního přípravku

4.1 Blokové schéma ovládacího programu

Ovládací program nebude příliš složitý. Po zapnutí zjistí, jestli je rameno v základní poloze a pokud není, tak ho do ní nastaví. Poté bude program vyčkávat na požadavky uživatele. Ten bude moci ovládat rameno pomocí stisknutí a přidržení tlačítek na navrženém přípravku. Pomocí tlačítek 1-4 bude moci otáčet motory a tím kontrolovat rameno. Přidržení každého tlačítka vždy způsobí, že PLC jednotka pošle do řídicího obvodu signál na pin DIR pro určení směru otáčení a pulzní obdélníkový signál na pin STEP.

Motor M1 bude použit pro otáčení ramene kolem své osy a jeho rozsah pohybu bude omezen pouze na 360 °, aby nedocházelo k omotání přívodního kabelu k motoru M2 na rameno.

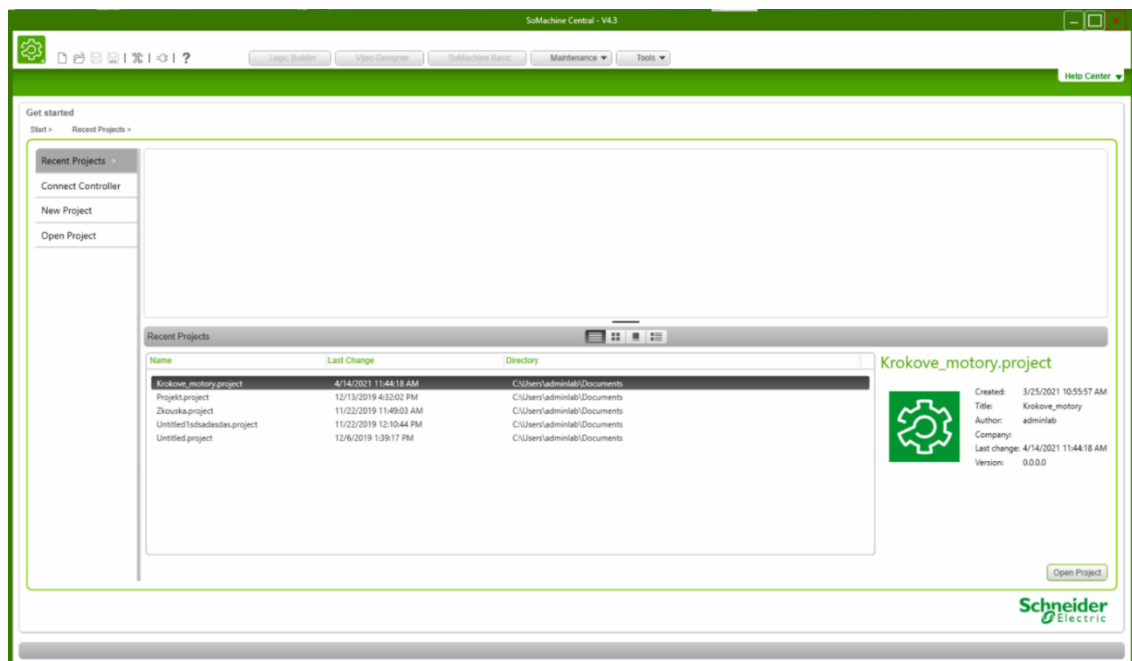


Obrázek 4.2 Blokové schéma programu

5. VÝVOJOVÉ PROSTŘEDÍ SOMACHINE

Vývojové prostředí SoMachine bylo vytvořeno firmou Schneider Electric pro programování jejich produktů. Skládá se z několika dalších dílčích softwarů, které společně slouží pro kontrolu a řízení PLC. Pro účely této práce byla použita konkrétně verze SoMachine V4.3.

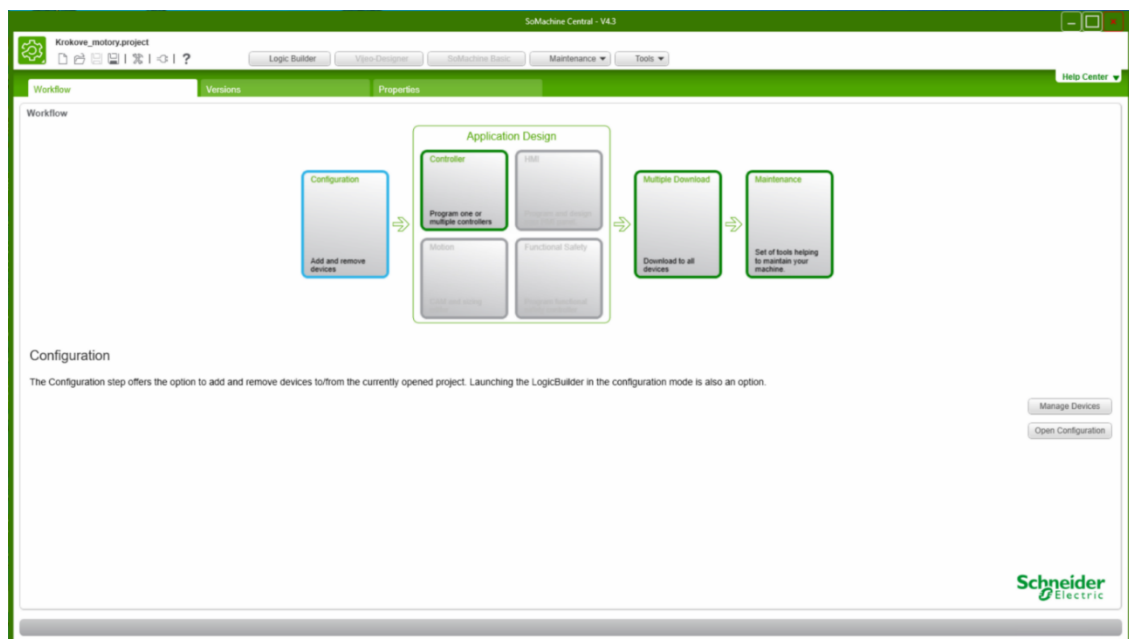
Po spuštění programu se nejprve otevře SoMachine Central, kde si uživatel zvolí, na kterém projektu chce pracovat. V prostřední části obrazovky se zobrazují nedávno otevřené projekty pro jednodušší přístup. Vlevo se pak nachází menu, se čtyřmi možnostmi. První jsou již zmiňované nedávné projekty. Další možností je *Connect Controller*, kde lze zjistit, jestli je k počítači připojeno nějaké PLC. Pod záložkou *New Project* se nachází založení nového projektu. Zde si uživatel může vybrat, jestli chce začít čistý projekt, použít asistenta pro tvorbu projektu nebo začít již z nějaké přednastavené šablony. Poslední položkou v menu je možnost *Open Project*, čímž se zvolí otevření projektu uloženého na počítači.



Obrázek 5.1 SoMachine Central

Po vybrání libovolného programu se objeví obrazovka, kterou můžeme vidět na obrázku 5.2. Zde může uživatel nakonfigurovat výběr komponentů a také nahrát program do všech připojených zařízení. Na horní liště se nachází tlačítka pro spuštění jednotlivých softwarů SoMachine. Vedle záložky *Workflow* můžeme vidět ještě záložku *Versions*, která poskytuje informace o různých verzích uživatelem vytvořeného programu a také záložku *Properties*, ve které můžeme zobrazit a spravovat vlastnosti

zvoleného projektu. [13] [14]



Obrázek 5.2 Otevřený projekt

5.1 SoMachine Basic

SoMachine Basic je jeden ze softwarů balíčku SoMachine, určený k tvorbě programů pro PLC jednotky. Byl vytvořen pro jednoduché a rychlé konfigurování PLC jednotek. Tím pádem se hodí hlavně pro nové uživatele s méně zkušenostmi. Je však omezen pouze na konfigurování PLC jednotky Modicon M221 a modulů Modicon TM2 a TM3. Jelikož se v této práci používá PLC řady Modicon M241, nebudeme se dále tímto softwarem zabývat, jelikož není se zvolenou jednotkou kompatibilní. [15]

5.2 Vijeo Designer

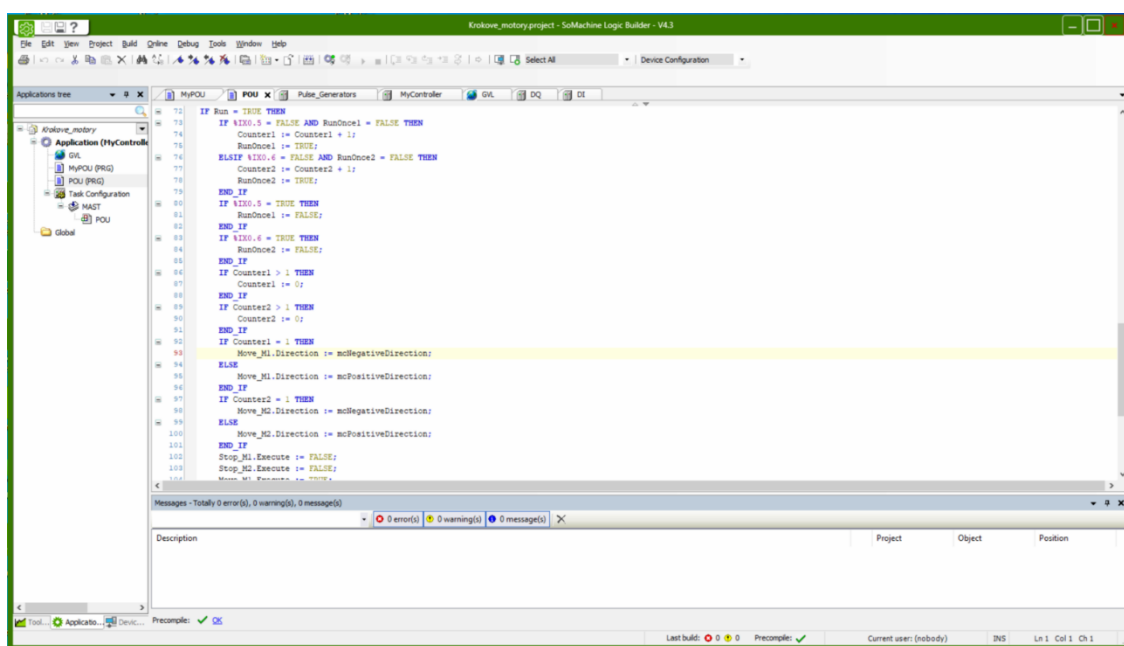
Vijeo Designer je další software z balíčku SoMachine. Slouží k nastavení HMI, tedy rozhraní mezi člověkem a strojem. Díky němu lze vytvořit vizualizaci pro řízení a monitorování programu bez nutnosti použít hardwarové součástky jako displeje a tlačítka. To je velmi užitečné právě v případě, že nejsou k dispozici fyzická tlačítka a spínače, které by bylo možné připojit na vstupy PLC. Vizualizaci je pak možné využít pro jednoduché a přehledné ovládání programu při jeho testování, bez nutnosti použití hardwaru.

Uprostřed prostředí Vijeo Designer se nachází pracovní plocha, do které jsou umístovány jednotlivé vizualizační prvky. Může se jednat například o vypínače, tlačítka, displeje, otočné knoflíky, signalizační kontrolky a další. Tyto prvky lze pak

nastavit tak, aby jimi byly řízeny programové proměnné anebo naopak, aby byly vizualizační prvky řízeny proměnnými. [16]

5.3 Logic Builder

Logic Builder je určen pro psaní programů pro PLC jednotky, stejně jako výše zmiňovaný SoMachine Basic. Avšak s tím rozdílem, že disponuje mnoho více funkcemi a možnostmi programování a není specifikován na programování pouze jednoho typu PLC. Díky tomu je vhodnější pro zkušenější uživatele, které by prostředí SoMachine Basic mohlo limitovat.



Obrázek 5.3 Logic Builder

Na obrázku 5.3 je zobrazeno prostředí Logic Builder. Úplně vlevo je sloupec *Applications tree*, který zobrazuje všechny vytvořené programy a jejich případné zařazení do konfigurace PLC. Vytváření nových programů nebo funkcí se provádí právě v této kartě. Je nutné kliknout pravým tlačítkem na zvolený logický kontrolér a vybrat možnost *Add Object – POU*. V tomto stejném okně lze také pomocí spodních dvou záložek zobrazovat další karty. Jednou z nich je karta *Devices tree*, kde jsou uvedeny všechny komponenty, které byly uživatelem přiřazeny do projektu. Pokud jsou některé z nich připojeny k počítači, tak jsou označeny zeleným rámečkem. Dále zde jde zobrazit karta *Tools tree*, v které je možné přidávat a odebírat knihovny funkcí, které jsou v projektu používány.

Uprostřed se nachází okno s hlavními kartami. Může se jednat o karty pro vytváření programů, ale také třeba karty s adresami digitálních vstupů, globálních proměnných a další. Pro psaní programu v Logic Builderu lze použít všechny standartní jazyky pro

PLC uvedené v kapitole 2.2. Ve spodní straně okna Logic Builderu se nachází karta *Messages*, ve které jsou zobrazovány chyby, varování a zprávy.

Rozvržení zobrazení vývojového prostředí není pevně dané a každý uživatel si ho může upravit podle svých potřeb. Jednotlivé karty lze dokonce zobrazovat i v samostatných oknech, nejen v hlavním okně prostředí. [14]

6. DEMONSTRAČNÍ PŘÍPRAVEK K LABORATORNÍ ÚLOZE

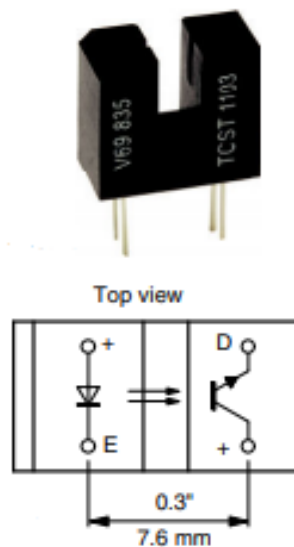
Pro potřeby laboratorní úlohy byl vytvořen přídatný modul s obvodem MP6500 navržený v kapitole 3 a také pohyblivé rameno ovládané krokovými motory, které bylo navrženo v kapitole 4.

6.1 Demonstrační přípravek

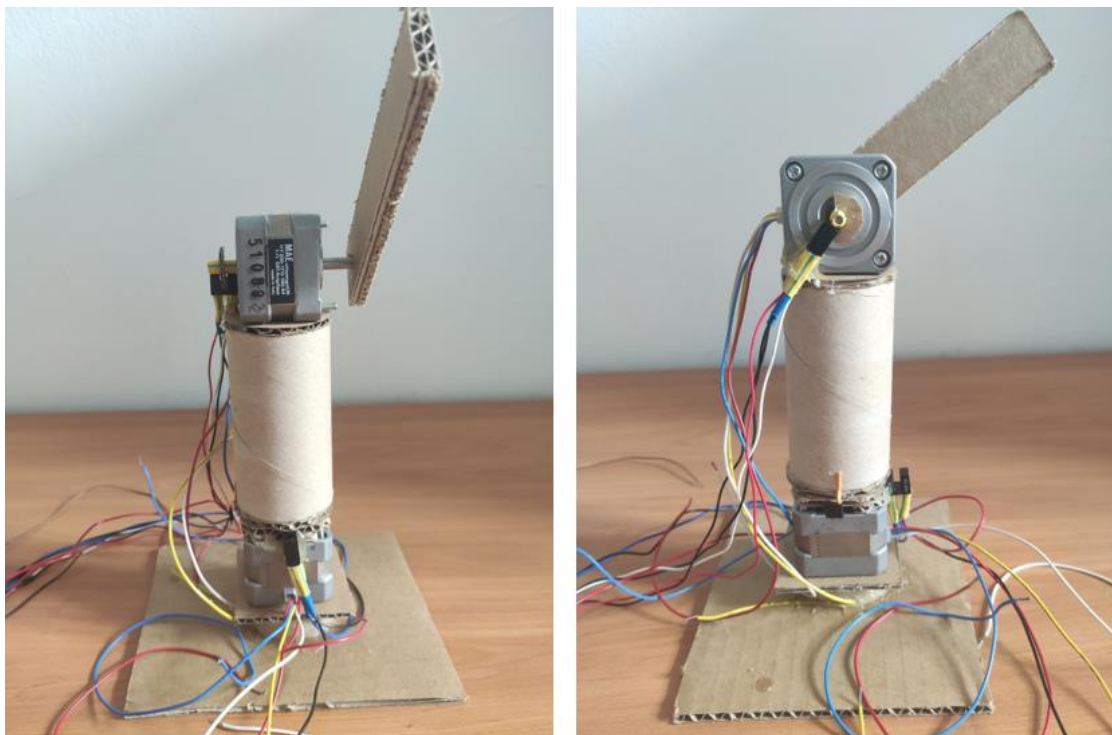
Demonstrační přípravek byl vytvořen podle návrhu na obrázku 4.1. Byl však doplněn ještě o dvojici optických závor, které slouží jako signalizace, že rameno dosáhlo své koncové polohy.

Díky nim bude možné realizovat funkci pro uvedení ramene do základní pozice a také omezit rozsah pohybu ramene, aby nedocházelo k nežádoucímu zamotávání přívodních kabelů k motorům a sensorům.

Jako optická závora byla použita součástka TCST 1103. Jedná se o senzor tvořený fototranzistorem na jedné straně a infračervenou diodou na straně druhé. Obě tyto části jsou umístěny ve společném pouzdře, které blokuje viditelné světlo.



Obrázek 6.1 Optická závora [17]



Obrázek 6.2 Demonstrační přípravek

6.2 Přídavný modul s obvodem MP6500

Přídavný modul byl vyroben podle schématu zapojení na obrázku 3.7, ale s pár drobnými změnami. Hlavní změnou je přidání části pro řízení stejnosměrného motoru. Ta však nebude pro účely této laboratorní úlohy využita. Byla přidána z důvodu možnosti využití jednoho přídavného modulu pro laboratorní úlohy s krokovými motory a zároveň i pro úlohy se stejnosměrnými motory.

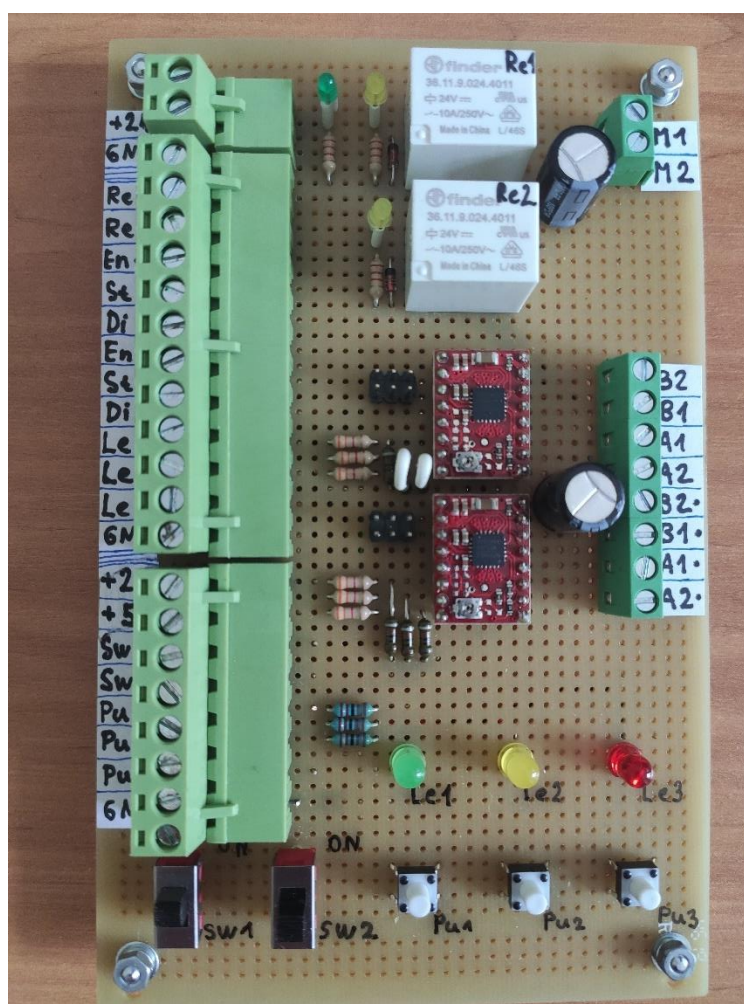
Další změnou je přidání pinů, které jsou napojeny na piny MS1 a MS2 destičky Pololu-2967. Díky nim bude v případě potřeby možné měnit v jakém krokovacím režimu motory pracují. Různými konfiguracemi zapojení pinů bude možné volit mezi režimy krokování zobrazenými v tabulce 6.1.

Tabulka 6.1 Režimy krokování

MS2	MS1	Režim krokování
L	L	Plný krok
L	H	Poloviční krok
H	L	Čtvrtinový krok
H	H	Osminový krok

Poslední změnou oproti původnímu návrhu je přidání tří barevných led diod, které bude možné používat při laboratorních úlohách. Realizovaný přídatný modul je pouze prototyp, který sloužil k otestování správné funkčnosti a bylo jej jednoduché upravovat podle potřeb. Přídatné moduly pro laboratorní úlohy už pak budou osazeny na deskách plošných spojů.

Kompletní schéma zapojení přídatného modulu lze najít v příloze Příloha A -



Obrázek 6.3 Prototyp přídatného modulu

6.3 Zapojení pracoviště pro laboratorní úlohu

Laboratorní úloha je zapojena stejně jak bylo navrženo na obrázku 3.6. Jedinou změnou bylo přidání optických závor pro detekci polohy ramene. Optické závory jsou ještě doplněny odpory o hodnotě 1200 Ω pro diodu i pro tranzistor, aby byl omezen proud, který jimi teče pouze na hodnotu 20 mA. Omezení proudů přidanými odpory bylo nutné ke splnění provozních parametrů komponentů. Schéma zapojení úlohy je k dispozici

v laboratorním návodu na obrázku 8.3 a obrázek 6.4 zobrazuje pracoviště pro laboratorní úlohu.



Obrázek 6.4 Pracoviště pro laboratorní úlohu.

7. MĚŘENÍ NA LABORATORNÍM PŘÍPRAVKU

Tato kapitola se bude zabývat průběhy proudů krokovými motory a kontrolou funkčnosti regulace proudu pomocí desky Pololu-2967.

Krokové motory jsou řízeny pomocí proudových impulsů puštěných do cívek motoru. Při vysokých otáčkách však proud nemusí stihnout dosáhnout své přednastavené hodnoty a motor pak pracuje s nižším momentem, než by měl. Proto je nutné ověřit, jestli motor nepracuje v příliš vysokých otáčkách a proud dosahuje své maximální nastavené hodnoty. Proud motorem se na desce Pololu-2967 reguluje pomocí potenciometru a požadovaná hodnota lze nastavit dvěma různými metodami.

První možností je nastavení proudového omezení pomocí měření proudu protékajícího cívkou motoru v režimu plného kroku, bez aktivace pinu STEP. Změřený proud se pak rovná 0,7násobku proudového limitu. Je to dáno tím, že při plném kroku jsou obě cívky motoru sepnuté a prochází jimi zhruba 70 % proudu, který je nastaven jako maximální.

Další možností je změřit referenční napětí pro nastavení proudového omezení a následně z něj pomocí následující rovnice vypočítat proudové omezení.

$$I_{limit} = U_{ref} \times 3,5 \quad (7.1)$$

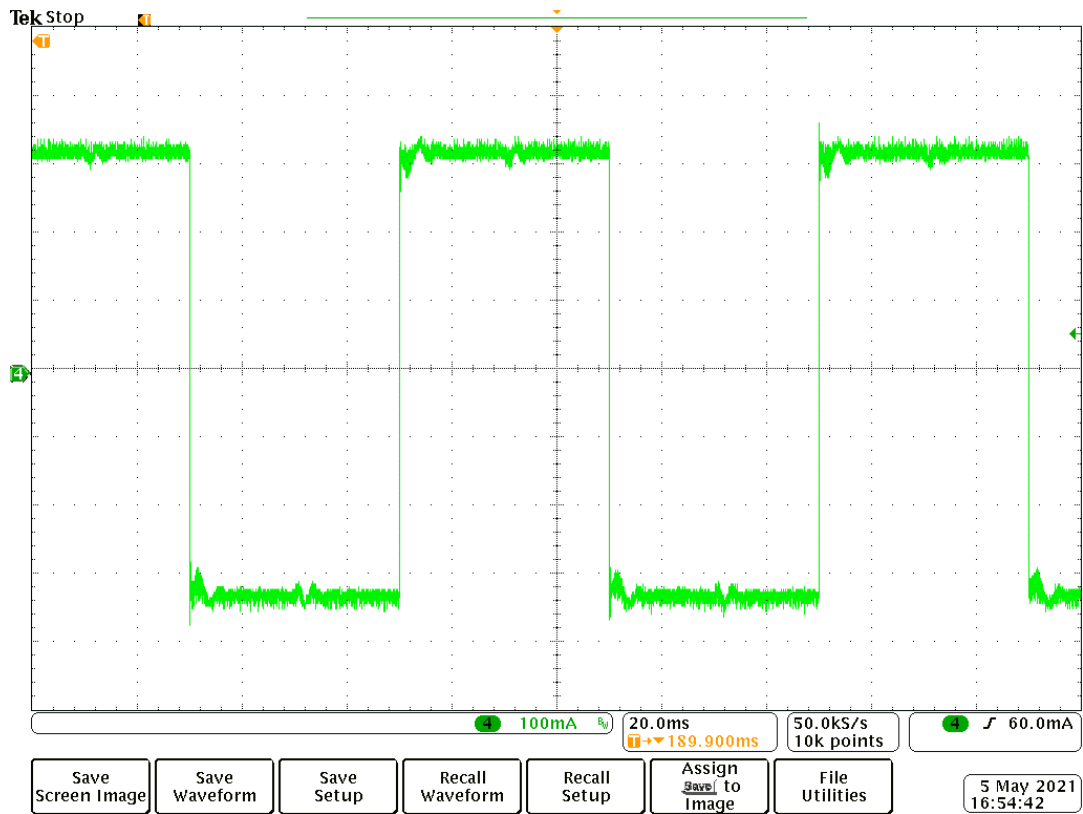
Jak můžeme vidět z rovnice (7.1, proudové omezení v ampérech se rovná 3,5násobku referenčního napětí ve voltech.

7.1 Nastavení proudového omezení

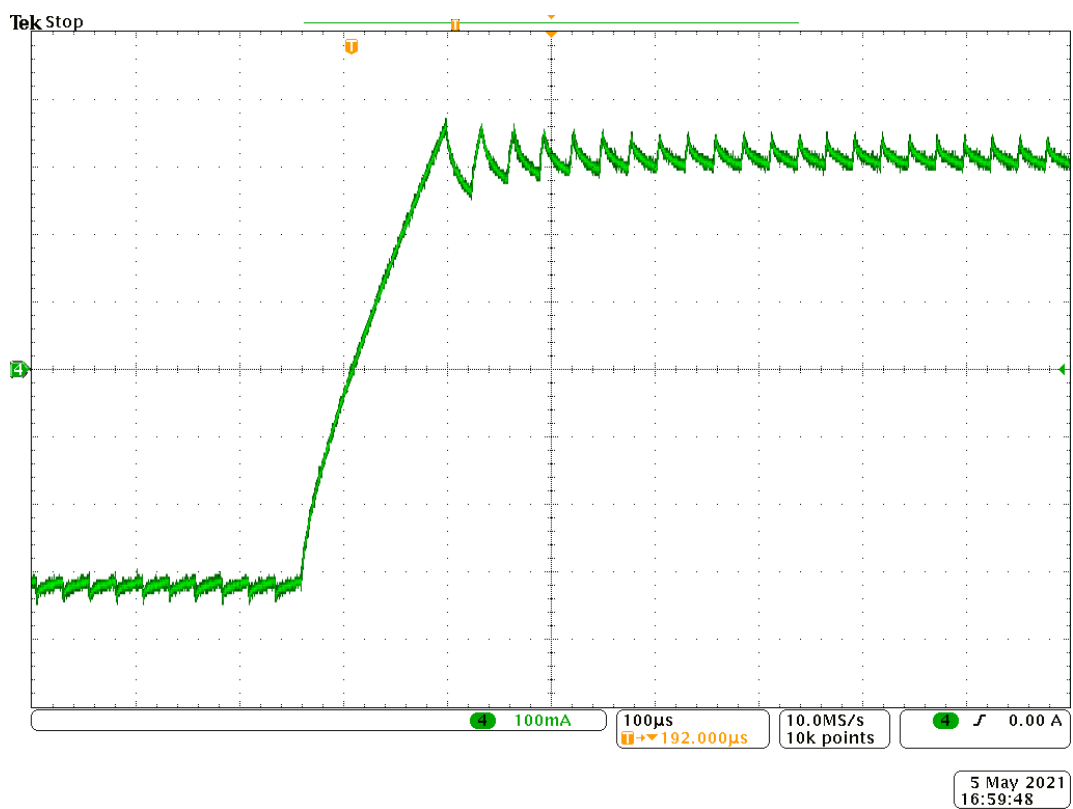
Proudové omezení bylo nastaveno pomocí metody referenčního napětí. Maximální povolený proud cívkou motoru je 0,95 A. Pro účely laboratorní úlohy však není potřeba, aby motor pracoval se svým maximálním momentem. Proudové omezení tak může být nastaveno na menší hodnotu, aby motor zbytečně nepracoval na hranici svých možností. Referenční napětí tedy bylo nastaveno na hodnotu 0,15 V. Po dosazení do rovnice (7.1 dostaneme hodnotu nastaveného proudového omezení, která je 0,525 A.

7.2 Měření průběhů proudu

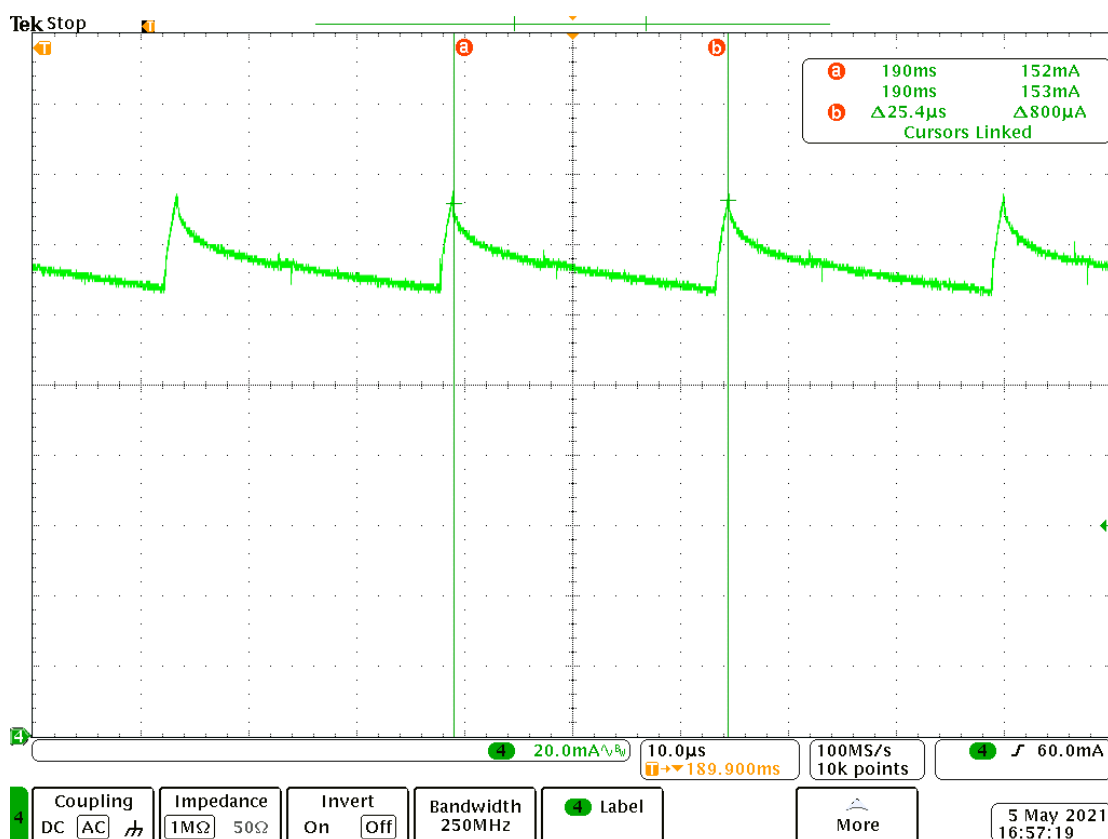
Průběhy proudu byly změřeny pomocí osciloskopu a proudové sondy. Předpokládaný proud cívkou motoru je zhruba 367 mA. Proudové omezení je sice nastaveno na 525 mA, avšak musíme mít na paměti, že pokud motor pracuje s plnými kroky, protéká jeho cívkami pouze 70 % nastaveného proudu.



Obrázek 7.1 Proud fází motoru



Obrázek 7.2 Náběžná hrana proudového pulzu



Obrázek 7.3 Detail zvlnění proudu

Na obrázku 7.1 můžeme vidět průběh proudových pulzů motorem, z nichž každý impulz znamená otočení motoru o jeden krok. Obrázek 7.2 zobrazuje náběžnou hranu jednoho pulzu proudu. Díky tomuto obrázku můžeme říci, že proud, který byl pro motor nastaven, jím opravdu teče. Předpokládaný proud byl 367 mA. Při měřítku 100 mA na dílek, můžeme z grafu vyčíst, že proud dosahuje hodnot zhruba 340 mA, což odpovídá nastavené hodnotě. Odchylna může být způsobena nepřesností měření nebo také ne úplně přesnému nastavení referenčního napětí na 0,15 V. Obrázek 7.3 zobrazuje zvlnění proudu kolem jeho nastavené hodnoty. Z grafu můžeme odečíst, že zvlnění proudu je zhruba 25 mA.

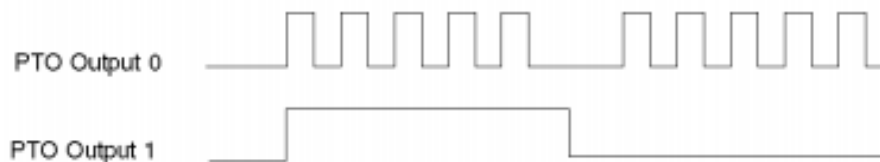
8. SOFTWAREVÁ ČÁST LABORATORNÍ ÚLOHY

Laboratorní úloha byla realizována na základě návrhu v kapitole 4. Funkce programu ramene však byla pozměněna, aby byly využity optické závory sloužící k omezení rozsahu pohybu ramene. Řídící program byl napsán v softwaru Logic Builder za použití jazyka ST. Kompletní kód programu je uveden v příloze Příloha B -

8.1 Funkce programu

Dva vypínače umístěné na přídavném modulu slouží k aktivaci řídicích obvodů motorů. Jedno z naprogramovaných tlačítek slouží k uvedení ramene do základní pozice, kdykoliv se rameno nachází v klidovém režimu. Zbylé dvě tlačítka pak obstarávají spuštění a zastavení pohybu ramene.

Proudové impulzy jsou generovány pomocí PTO funkce PLC jednotky. Pro řízení krokového motoru je výhodnější upřednostnit řízení pomocí PTO před PWM. Díky PTO totiž rovnou můžeme kontrolovat i směr otáčení krokového motoru a to tak, že je nastaven do módu, kdy na jeden výstup PLC je posílán obdélníkový signál pro krokování motoru a na druhý také obdélníkový signál, tentokrát určený pro definování směru otáčení motoru.

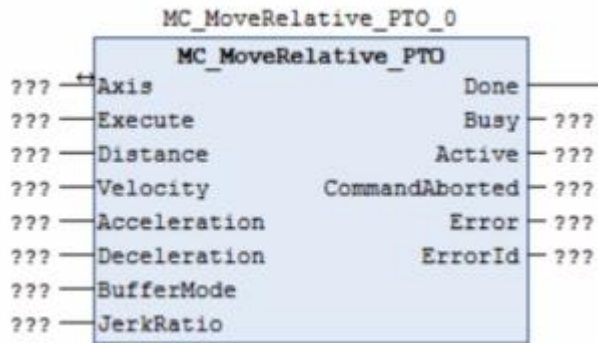


Obrázek 8.1 Příklad signálů generovaných PTO [13]

Rameno je naprogramováno tak, aby se po stisknutí tlačítka START začalo otáčet kolem své osy a zároveň se také pohybovala jeho vrchní část. Jakmile kterákoliv z jeho částí dosáhne své krajní polohy, vymezené optickými závorami a zářkami, dojde ke změně směru otáčení motoru. Tento cyklus se opakuje pořád dokola, dokud není zmáčknuto tlačítko STOP. Spuštění a zastavení běhu programu může být prováděno opakovaně. Motory budou po spuštění programu pokračovat v otáčení ve stejném směru, ve kterém skončily. Pokud dojde k zastavení programu ve stavu, kdy je některá část ramene ve svojí krajní poloze, tak se motor po spuštění bude otáčet takovým směrem, aby nedošlo k zamotání kabelů.

Celý program je naprogramován za využití struktur IF, jelikož je to pro studenty pochopitelnější a jednodušejší realizovatelné řešení než například použití struktury SWITCH.

Pro ovládání PLC funkce PTO je nutné volat funkční bloky. To se provede jejich definováním v globálních proměnných a následným použitím v jazyce ST. Pro aktivaci PTO slouží blok *MC_Power_PTO*, pro deaktivaci *MC_STOP_PTO*. Pro vyslání signálu pro motory byl použit blok *MC_MoveVelocity_PTO* a pro zastavení vyslání signálu blok *MC_HALT_PTO*.



Obrázek 8.2 Funkční blok *MC_MoveVelocity_PTO*

Na obrázku 8.2 se nachází funkční blok *MC_MoveVelocity_PTO*, který způsobí stálé otáčení motoru. Pro jeho správnou funkci nepotřebuje mít definovány všechny vstupy a výstupy. Hlavním vstupem, který musí být vždy definován je vstup *Axis*. Na něm se definuje, s kterým předdefinovaným PTO bude blok pracovat. Dále je potřeba definovat, kolik impulzů za sekundu bude PTO vysílat. To se nastaví na vstupu *Velocity*. Vstupy *Acceleration* a *Deceleration* pak určují v milisekundách, jak rychle motor dosáhne nastavené rychlosti *Velocity*. Posledním vstupem, který musíme nastavit je vstup *Direction*, který určuje, jakým směrem se bude motor točit. Jakmile jsou nastaveny všechny tyto potřebné vstupní parametry, lze funkční blok spustit nastavením logické 1 na vstupu *Execute*. Na stejném principu fungují i všechny ostatní použité funkční bloky.

V následujícím úryvku kódu jsou definovány všechny potřebné bloky pro ovládání PTO pro první motor. Jsou zde definovány pouze jejich parametry, které se nemění. Ostatní jsou přepisovány při běhu programu.

```

On_M1: MC_Power_PTO;
Off_M1: MC_STOP_PTO;
Move_M1: MC_MoveVelocity_PTO;
Stop_M1: MC_HALT_PTO;

On_M1(Axis := PTO_0, DriveReady := TRUE);
Move_M1(Axis := PTO_0, ContinuousUpdate := TRUE, Velocity := 50,
Acceleration := 10, Deceleration := 10);
Stop_M1(Axis := PTO_0, Deceleration := 10);
Off_M1(Axis := PTO_0, Deceleration := 10);

```

8.2 Laboratorní návod

Zadání:

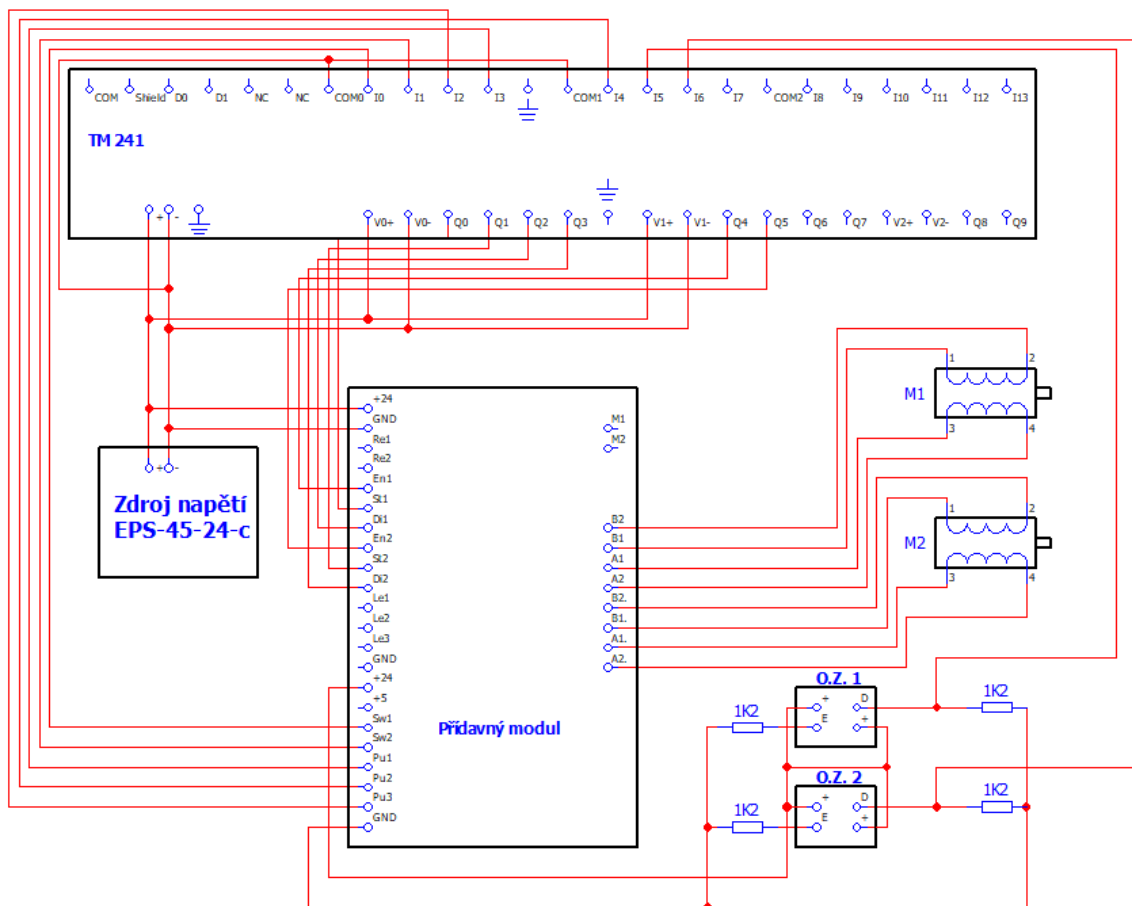
Vytvořte řídicí program v jazyce ST pro pohyb ramene poháněného krokovými motory. Funkce ramene bude taková, že se po spuštění programu bude otáčet a rameno se bude kývat ze strany na stranu. Vždy když dorazí do své koncové polohy, tak se změni směr otáčení.

Dva vypínače na přídatném modulu budou sloužit pro aktivaci řídicích obvodů motoru. Jedno tlačítko bude vyhrazeno na spuštění běhu programu a druhé na zastavení programu. Poslední tlačítko naprogramujte tak, aby se po jeho stisknutí rameno vrátilo do původní polohy. Pro generování signálů pro motory využijte integrovanou funkci PLC jednotek PTO, která bude umí do motoru posílat signál pro krokování a zároveň i signál pro směr otáčení.

Před začátkem programování se ujistěte, že je pracoviště zapojeno podle následujícího schématu a že je vhodně nastaveno proudové omezení pro motory, změřením referenčního napětí na potenciometrech přídatného modulu. Vhodné proudové omezení vypočítejte podle následujícího vztahu:

$$I_{limit} = U_{ref} \times 3,5$$

Schéma zapojení:



Obrázek 8.3 Schéma zapojení laboratorní úlohy

Postup řešení:

- 1) Zkontrolujte zapojení pracoviště a nastavte vhodně vypočtené proudové omezení motorů.
- 2) Spusťte SoMachine a založte nový projekt. Ten následně otevřete v programu Logic Builder.
- 3) Zde založte nový program v kartě *Applications tree* a zrovna ho přiřadte do master tasku.
- 4) V kartě *Devices tree* vyberte možnost *Pulse_Generators* a v ní nakonfigurujte dvě PTO pro řízení motorů. *Output mode* pro PTO zvolte *A Pulse/B Direction* a nastavte správné výstupy, podle zapojení úlohy.

- 5) Pro ovládání PTO použijte funkční bloky, které si zadefinujete v kartě globálních proměnných. Pro aktivaci PTO slouží blok *MC_Power_PTO*, pro deaktivaci *MC_STOP_PTO*. Pro spuštění signálu využijte blok *MC_MoveVelocity_PTO* a pro zastavení *MC_HALT_PTO*. V globálních proměnných bloky definujte následujícím způsobem `On_M1:MC_Power_PTO`. Názvy si zvolte podle sebe.
- 6) V kartě s kódem programu nejprve definujte parametry bloků pro obě PTO. Zápis vypadá následovně `On_M1(Axis := Název PTO, DriveReady := TRUE);`. U bloku *MC_MoveVelocity_PTO* definujte bit *Axis*, *ContinuousUpdate* nastavte na TRUE, *Velocity* nastavte na 50 kroků za sekundu a *Acceleration* a *Deceleration* nastavte na 10 ms. Pro *MC_HALT_PTO* a *MC_STOP_PTO* nastavte jen *Axis* a poté *Deceleration* zase na 10 ms.
- 7) Nejprve naprogramujte funkci vypínačů pro řídicí obvody. Pro označení vstupů a výstupů můžete použít přímo jejich adresy, které najdete v *Devices tree* pod položkami *DI* a *DQ*. Zároveň na vypínač nastavte i aktivaci a deaktivaci PTO pomocí vhodného nastavení bitů *Enable* pro *MC_Power_PTO* a *Execute* pro *MC_STOP_PTO*.
- 8) Dále nakonfigurujte tlačítko pro resetování ramene do základní polohy. Pro každý motor můžete použít samostatný příkaz IF. Pro určení směru otáčení je nutné nastavit bit *Direction* bloku *MC_MoveVelocity_PTO*. Směry otáčení jsou označeny *mcNegativeDirection* a *mcPositiveDirection*. Spouštění a vypínání signálů provedete vhodným nastavením bitů *Execute* u bloků *MC_MoveVelocity_PTO* a *MC_HALT_PTO*.
- 9) Tlačítko START nastavte tak, aby se po jeho stisknutí spustil běh programu a rameno konalo požadovaný pohyb. Pro změnu směru otáčení využijte interakci zářezek s optickými závory (počet interakcí můžete například používat pro nastavování hodnoty nějaké proměnné, která bude sloužit jako počítadlo, a podle její hodnoty nastavovat směr otáčení).
- 10) Následně nastavte tlačítko STOP, které zastaví pohyb ramene a vypne signály z PTO.
- 11) Hotový program nahrajte do PLC jednotky a otestujte jeho funkčnost. V případě neočekávaného chování programu buďte připraveni vypnout vypínače k řídicím obvodům motorů.

9. ZÁVĚR

Cílem práce bylo seznámit se s funkcí a stavbou krokových motorů a s programovatelnými logickými automaty, a to konkrétně s jednotkou Modicon M241 od výrobce Schneider Electric a také vývojovým prostředím SoMachine, které je rovněž od firmy Schneider Electric. Dalším cílem pak bylo doplnit PLC jednotku o přídatné zařízení a navrhnout laboratorní úlohu a experimentální verzi programu pro ovládání krokových motorů pomocí PLC.

Úvodní část se věnuje krokovým motorům. Je zde popsána jejich funkce, rozdělení a možnosti jejich řízení.

Druhá část rozebírá programovatelné logické automaty. Je zde popsána jejich funkce, rozdělení a také způsoby, kterými lze PLC jednotky programovat.

Třetí část se zabývá návrhem přípravku pro laboratorní úlohu. Jsou zde popsány funkce a vlastnosti přídatných zařízení i PLC jednotky. Dále je zde zpracováno schéma zapojení navrhovaného přípravku a také uspořádání laboratorního přípravku.

Čtvrtá část je pak věnována návrhu náplně laboratorní úlohy, která by měla spočívat v ovládní ramene pomocí řízení krokových motorů. Je zde také uvedeno blokové schéma programu, který bude použit pro řízení motorů.

V páté části je popsáno vývojové prostředí SoMachine s jednotlivými jeho softwary a programovacími možnostmi, které poskytují.

Šestá část popisuje realizaci demonstračního přípravku a přídatného modulu navržených v třetí a čtvrté kapitole. Byly zde popsány i úpravy a změny oproti původním návrhům.

V sedmé části jsou zobrazeny výsledky měření proudů na demonstračním přípravku. Díky měření byla ověřena funkčnost vyrobeného přídatného modulu a také správná funkce proudového omezení, které poskytuje deska Pololu-2967, která je součástí přídatného modulu.

Poslední část byla věnována programu pro laboratorní úlohu. Funkce programu byla oproti původnímu návrhu pozměněna, aby bylo možné využít i optické závory na demonstračním přípravku. Je zde podrobně popsána funkce programu a také způsoby, jakými byl program napsán. Nakonec pak byl napsán laboratorní návod, který bude použit pro výuku programování PLC.

Jako pokračování práce by mohlo být vytvoření přídatného modulu pro řízení motorů na DPS a vytvoření dalších laboratorních úloh a demonstračních přípravků.

LITERATURA

- [1] RYDLO, P. Krokové motory a jejich řízení. Liberec: Technická univerzita v Liberci, 2000.
- [2] KOLÁČNÝ, J. Elektrické mikropohony. Brno: Fakulta elektrotechniky a komunikačních technologií Vysoké učení technické v Brně, 2009.
- [3] ŘEZÁČ, K. robotika.cz [online]. 2002 [cit. 2020-12-7]. Dostupné z: <http://robotika.cz/articles/steppers/cs>
- [4] Robodoupě. Krokové motory 1 – typy motorů. [online]. 2013 [cit. 2020-12-7]. Dostupné z: <http://robodoupe.cz/2013/krokove-motory-1-typy-motoru/>
- [5] Linear Step Motor Systems. In: Motion Control Systems [online]. Charlotte, North Carolina: Parker Hannifin Corporation, b.r. [cit. 2020-12-7]. Dostupné z: http://www.compumotor.com/catalog/c75_78.pdf
- [6] Vojáček, A. Co se skrývá pod označením PLC?. [online]. 2007 [cit. 2020-12-13]. Dostupné z: <https://automatizace.hw.cz/co-se-skryva-pod-oznaceni-plc>
- [7] PLC-automatizace. PLC-programovatelné logické automaty. [online] [cit. 2020-12-13] Dostupné z: http://plc-automatizace.cz/index_plc.htm
- [8] Redakce časopisu. Co by měl každý vědět o programovacích jazycích PLC. [online] 2017 [cit. 2020-12-13] Dostupné z: <https://www.elektroprumysl.cz/software/co-by-mel-kazdy-vedet-o-programovacich-jazycich-plc>
- [9] Schneider Electric. PLC Modicon M241. [online]. [cit. 2021-1-28]. Dostupné z: <https://www.se.com/cz/cs/product/TM241CEC24T/plc-modicon-m241-24-io-poz.logika-ethernet-can-master/#>
- [10] Pololu Corporation. MP6500 Stepper Motor Driver Carrier. [online]. [cit. 2021-1-28]. Dostupné z: <https://www.pololu.com/product/2967/pictures>
- [11] Meanwell. EPS-45-24-C. [online]. [cit. 2021-1-28]. Dostupné z: <https://www.czech-meanwell.cz/meanwell/EPS-45-24-C-Meanwell-Zdroj-na-PCB-45W-24V-d3856.htm>
- [12] Petruzella, F. D. Programmable logic controllers. 4th ed. McGraw-Hill., New York 2010
- [13] Schneider Electric. Modicon M241 Logic Controller User Guide. [online]. [cit. 2021-4-25]. Dostupné z: https://download.schneider-electric.com/files?p_enDocType=User+guide&p_File_Name=EIO0000004267.00.pdf&p_Doc_Ref=EIO0000004267
- [14] HOWLETT, Bruce. Getting Started With SoMachine: Self Study Manual [online]. Schneider Electric Inc., 2014 [cit. 2021-4-25]. Dostupné z: https://download.schneider-electric.com/files?p_enDocType=Training+material&p_File_Name=Getting+Started+With+SoMachine+Self+Study+4.1.1.pdf&p_Doc_Ref=SoMachineTutorialManual

- [15] IRONS, Richard. SoMachine Basic: Training Manual. [online]. Schneider Electric Inc., 2014 [cit. 2021-4-25]. Dostupné z: https://download.schneider-electric.com/files?p_enDocType=Training+material&p_File_Name=SoMachine+Basic_v1.1_Training_Manual.pdf&p_Doc_Ref=SoMachineBasicTutorial_Manual
- [16] Schneider Electric. Vijeo Designer Tutorial. [online]. Schneider Electric Inc., 2014 [cit. 2021-4-25]. Dostupné z: https://download.schneider-electric.com/files?p_enDocType=User+guide&p_File_Name=Vijeo-Designer-Starting-guide-English.pdf&p_Doc_Ref=VD-userguide-V6.2
- [17] Vishay Intertechnology, Transmissive Optical Sensor with Phototransistor Output. [online] [cit. 2021-4-25]. Dostupné z: <https://www.gme.cz/data/attachments/dsh.523-174.1.pdf>

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

PLC	Programovatelný logický automat
ST	Strukturovaný text
FBD	Funkční blokový diagram
SFC	Sekvenční funkční diagram
IL	Seznam instrukcí
LD	Příčkový diagram
A	Ampér
V	Volt
DC	Stejnoseměrný proud
Ncm	Newton centimetr
nF	Nanofarad
Ω	Ohm
PWM	Pulzně šířková modulace
mA	Miliampér
ms	Milisekunda
PTO	Pulzní výstup
DPS	Deska plošných spojů

Symbyly:

U_{ref}	Referenční napětí	(V)
I_{limit}	Proudové omezení	(A)

SEZNAM PŘÍLOH

PŘÍLOHA A - SCHÉMA ZAPOJENÍ PŘÍDAVNÉHO MODULU	51
PŘÍLOHA B - ZDROJOVÝ KÓD PROGRAMU PRO LABORATORNÍ ÚLOHU	52

Příloha B - Zdrojový kód programu pro laboratorní úlohu

```
VAR_GLOBAL
    RunOnce1, RunOnce2, CalibrationDone_M1, CalibrationDone_M2, Run
: BOOL;
    Counter1, Counter2 : INT;
    On_M1: MC_Power_PTO;
    On_M2: MC_Power_PTO;
    Off_M1: MC_STOP_PTO;
    Off_M2: MC_STOP_PTO;
    Move_M1: MC_MoveVelocity_PTO;
    Move_M2: MC_MoveVelocity_PTO;
    Stop_M1: MC_HALT_PTO;
    Stop_M2: MC_HALT_PTO;
END_VAR

// Definování PTO
On_M1(Axis := PTO_0, DriveReady := TRUE);
Move_M1(Axis := PTO_0, ContinuousUpdate := TRUE, Velocity := 50,
Acceleration := 10, Deceleration := 10);
Stop_M1(Axis := PTO_0, Deceleration := 10);
Off_M1(Axis := PTO_0, Deceleration := 10);

On_M2(Axis := PTO_1, DriveReady := TRUE);
Move_M2(Axis := PTO_1, ContinuousUpdate := TRUE, Velocity := 50,
Acceleration := 10, Deceleration := 10);
Stop_M2(Axis := PTO_1, Deceleration := 10);
Off_M2(Axis := PTO_1, Deceleration := 10);

// Zapnutí řídicího obvodu Motoru 1
IF %IX0.0 = TRUE THEN
    %QX0.4 := TRUE;
    Off_M1.Execute := FALSE;
    On_M1.Enable := TRUE;
ELSE
    %QX0.4 := FALSE;
    On_M1.Enable := FALSE;
    Off_M1.Execute := TRUE;
END_IF

// Zapnutí řídicího obvodu Motoru 2
IF %IX0.1 = TRUE THEN
    %QX0.5 := TRUE;
    Off_M2.Execute := FALSE;
    On_M2.Enable := TRUE;
ELSE
    %QX0.5 := FALSE;
    On_M2.Enable := FALSE;
    Off_M2.Execute := TRUE;
END_IF

// Kalibrace Motoru 1
IF %IX0.4 = TRUE AND CalibrationDone_M1 = FALSE THEN
    Move_M1.Direction := mcNegativeDirection;
    Stop_M1.Execute := FALSE;
    Move_M1.Execute := TRUE;
```

```

Counter1 := 1; //Proměnné Counter
slouží jako počítadlo interakcí s optickou závorou a podle její
hodnoty se nastavuje směr otáčení motoru
ELSIF %IX0.5 = FALSE THEN
    Stop_M1.Execute := TRUE;
    CalibrationDone_M1 := TRUE; //Indikuje, zda proběhla
kalibrace a brní jejímu vícenásobnému spuštění, v případě, že už
proběhla rameno je v základní poloze
ELSE
    Move_M1.Execute := FALSE;
    Stop_M1.Execute := FALSE;
END_IF

// Kalibrace Motoru 2
IF %IX0.4 = TRUE AND CalibrationDone_M2 = FALSE THEN
    Move_M2.Direction := mcNegativeDirection;
    Stop_M2.Execute := FALSE;
    Move_M2.Execute := TRUE;
    Counter2 := 1;
ELSIF %IX0.6 = FALSE THEN
    Stop_M2.Execute := TRUE;
    CalibrationDone_M2 := TRUE;
ELSE
    Move_M2.Execute := FALSE;
    Stop_M2.Execute := FALSE;
END_IF

// Tlačítko START
IF %IX0.3 = TRUE THEN
    Run := TRUE;
    CalibrationDone_M1 := FALSE;
    CalibrationDone_M2 := FALSE;
END_IF

// Běh cyklu
IF Run = TRUE THEN
    IF %IX0.5 = FALSE AND RunOnce1 = FALSE THEN
        Counter1 := Counter1 + 1;
        RunOnce1 := TRUE; //Proměnné RunOnce zajišťují
pouze jedno přičtení counteru při interakci s optickou závorou
ELSIF %IX0.6 = FALSE AND RunOnce2 = FALSE THEN
        Counter2 := Counter2 + 1;
        RunOnce2 := TRUE;
END_IF
    IF %IX0.5 = TRUE THEN // Resetování proměnné RunOnce po
skončení interakce s optickou závorou
        RunOnce1 := FALSE;
END_IF
    IF %IX0.6 = TRUE THEN
        RunOnce2 := FALSE;
END_IF
    IF Counter1 > 1 THEN
        Counter1 := 0;
END_IF
    IF Counter2 > 1 THEN
        Counter2 := 0;
END_IF
    IF Counter1 = 1 THEN //Nastavení směru otáčení podle
hodnoty counteru

```

```

        Move_M1.Direction := mcNegativeDirection;
    ELSE
        Move_M1.Direction := mcPositiveDirection;
    END_IF
    IF Counter2 = 1 THEN
        Move_M2.Direction := mcNegativeDirection;
    ELSE
        Move_M2.Direction := mcPositiveDirection;
    END_IF
    Stop_M1.Execute := FALSE;
    Stop_M2.Execute := FALSE;
    Move_M1.Execute := TRUE;
    Move_M2.Execute := TRUE;
END_IF

// Tlačítko STOP
IF %IX0.2 = TRUE THEN
    Run := FALSE;
    Stop_M1.Execute := TRUE;
    Stop_M2.Execute := TRUE;
    Move_M1.Execute := FALSE;
    Move_M2.Execute := FALSE;
    CalibrationDone_M1 := FALSE;
    CalibrationDone_M2 := FALSE;
END_IF

```