

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

VYUŽITÍ PROGRAMOVATELNÝCH LOGICKÝCH AUTOMATŮ VE VÝUCE

USE OF PROGRAMMABLE LOGIC CONTROLLERS IN THE TEACHING PROCESS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Jan Fojtášek

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 Fojtášek

ID: 203220

Ročník: 3

Akademický rok: 2020/21

NÁZEV TÉMATU:

Využití programovatelných logických automatů ve výuce

POKYNY PRO VYPRACOVÁNÍ:

1. Prostudujte a popište strukturu a vlastností programovatelných logických automatů Schneider a odpovídajících programovacích jazyků.
2. Navrhněte laboratorní úlohu s krokovými motory, vhodnou pro výukové účely.
3. Úlohu realizujte a optimalizujte dle pokynů vedoucího.
4. Vytvořet příslušný laboratorní návod.

DOPORUČENÁ LITERATURA:

- [1] Dokumentace k PLC Schneider electric.
- [2] Vorel P., Patocka M., Průmyslová elektronika, Vydání 1., skriptum FEKT VUT Brno, 2007
- [3] SKALICKÝ, J. Navrhování elektrických pohonů. Navrhování elektrických pohonů. Brno: VUT FEKT, 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 se zaměřuje na programovatelné logické automaty ve výuce. Tato bakalářská práce je rozdělena do šesti kapitol. První kapitola se zabývá historií, technickým vybavením a přednostmi těchto automatů. V druhé kapitole je sepsáno provedení a typy jednotlivých automatů. Třetí kapitola je zaměřena na programovací jazyky s jejich grafickým znázorněním. Čtvrtá a pátá kapitola popisuje typy programovatelných logických automatů Modicon a programovatelné prostředí SoMachine od firmy Schneider Electric. Závěrečnou část této práce tvoří návrh a realizace laboratorní úlohy s krokovým motorem, kterou lze využít ve výuce. Pro tuto úlohu byla vytvořena jednoduchá vizualizace a laboratorní návod pro řídicí program.

Klíčová slova

M241, Modicon, PLC, Programovatelný logický kontrolér, Schneider, SoMachine, Krokový motor.

Abstract

The bachelor thesis focuses on programmable logic controllers in teaching. This bachelor thesis is divided into six chapters. The first chapter deals with the history, technical equipment and merits of these machines. In the second chapter, the design and types of individual automats are written. The third chapter focuses on programming languages with their graphic representation. The fourth and fifth chapters describe the types of Modicon's programmable logic machines and Schneider Electric's SoMachine programmable environment. The final part of this work consists of the design and implementation of a step-engine laboratory role that can be used in teaching. A simple visualisation and laboratory guide for the management programme have been developed for this role.

Keywords

M241, Modicon, PLC, Programmable logic controller, Schneider, SoMachine, Modicon, Stepper motor.

Bibliografická citace

FOJTÁŠEK, Jan. *Využití programovatelných logických automatů ve výuce* [online]. Brno, 2021 [cit. 2021-05-18]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/134608>. 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 Fojtášek*

VUT ID studenta: *203220*

Typ práce: *Bakalářská práce*

Akademický rok: *2020/21*

Téma závěrečné práce: *Využití programovatelných logických automatů ve výuce*

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

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

V Brně dne: 27. května 2021

podpis autora

Poděkování

Děkuji mému vedoucímu bakalářské práce Ing. Daliborovi Červinkovi, Ph.D. a panu Ing. Janu Knoblochovi, Ph.D. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé bakalářské práce. Na závěr bych chtěl vyjádřit velké díky mé rodině a přátelům, kteří mi byli podporou po celou dobu mého studia.

V Brně dne: 27. května 2021

podpis autora

Obsah

SEZNAM OBRÁZKŮ	9
SEZNAM TABULEK.....	10
ÚVOD	11
1. PROGRAMOVATELNÉ LOGICKÉ AUTOMATY	12
1.1 TECHNICKÉ VYBAVENÍ PLC	12
1.2 HISTORIE PLC.....	13
1.3 PŘEDNOSTI PLC	14
2. PROVEDENÍ PLC	15
2.1 MODULÁRNÍ PLC.....	16
2.2 KOMPAKTNÍ PLC	16
3. PROGRAMOVACÍ JAZYKY PLC	17
3.1 JAZYK STRUKTUROVANÉHO TEXTU (ST)	17
3.2 JAZYK SEZNAMU INSTRUKCÍ (IL)	18
3.3 JAZYK PŘÍČKOVÉHO DIAGRAMU (LD)	18
3.4 JAZYK FUNKČNÍHO BLOKOVÉHO SCHÉMATU (FBD)	19
3.5 JAZYK SEKVENČNÍCH FUNKČNÍCH DIAGRAMŮ (SFC)	20
4. PLC FIRMY SCHNEIDER ELECTRIC	21
4.1 PLC ŘADY MODICON M241	22
5. PROGRAMOVACÍ PROSTŘEDÍ SOMACHINE	25
5.1 LOGIC BUILDER.....	27
5.1.1 Vytvoření a nahrání projektu do PLC.....	29
5.2 VIJEQ DESIGNER	30
6. REALIZACE LABORATORNÍ ÚLOHY	32
6.1 LABORATORNÍ ÚLOHA S KROKOVÝM MOTOREM	32
6.2 KROKOVÝ MOTOR HY 200 1713	33
6.3 INTEGROVANÝ DRIVER KROKOVÉHO MOTORU A4988.....	33
6.4 HLAVNÍ PROGRAM V PROSTŘEDÍ SoMACHINE	35
6.4.1 Program pro integrovaný obvod A4988	35
6.4.2 Program pro povolení pohybu motoru.....	36
6.4.3 Program pro čtení a vynulování pozice motoru.....	38
6.4.4 Program pro mikrokrokování motoru	38
6.4.5 Program pro najetí motoru do dané pozice	39
6.4.6 Program pro vykonání určité vzdálenosti	40
6.4.7 Program pro stálé otáčení motoru	43
6.5 VIZUALIZACE	47
6.6 NAMĚŘENÉ PRŮBĚHY KROKOVÉHO MOTORU.....	48
7. ZÁVĚR.....	54
LITERATURA.....	55

SEZNAM SYMBOLŮ A ZKRATEK	57
SEZNAM PŘÍLOH.....	58

SEZNAM OBRÁZKŮ

1.1	Cyklus PLC	12
1.2	Bloková struktura PLC.....	13
1.3	První PLC MODICON 084 [4]	14
2.1	Modulární PLC.....	16
2.2	Kompaktní PLC	16
3.1	Programovací jazyky PLC.....	17
3.2	Příklad jazyka ST [8].....	18
3.3	Příklad jazyka IL	18
3.4	Příklad jazyka LD [9].....	19
3.5	Příklad jazyka FBD [9]	19
3.6	Příklad jazyka SFC [9]	20
4.1	PLC model TM241CEC24T [18].....	22
5.1	Primární obrazovka programu SoMachine.....	25
5.2	Obrazovka pracovního prostoru projektu	27
5.3	Výběr PLC typu M241	27
5.4	Pracovní prostor Logic Builderu	28
5.5	Přidání objektu POU	29
5.6	Grafické prostředí Vijeo-Designer	30
6.1	Model pro laboratorní úlohu.....	32
6.2	Krokový motor HY 200	33
6.3	Schéma zapojení s integrovaným obvodem A4988.....	34
6.4	DPS s tlačítky a integrovaným obvodem A4988.....	35
6.5	Program pro integrovaný obvod A4988	36
6.6	Program pro povolení pohybu motoru	36
6.7	Homing Mode, Reset a Stop pohybu motoru	37
6.8	Čtení aktuální pozice a vynulování pozice	38
6.9	Program pro krokování motoru	39
6.10	Program pro najetí motoru do dané pozice.....	40
6.11	Vykonání určité vzdálenosti vpřed při zvolené rychlosti	41
6.12	Vykonání určité vzdálenosti zpět a vpřed při dané rychlosti.....	41
6.13	Vykonání zvolené vzdálenosti zpět a vpřed při dané rychlosti	42
6.14	Vykonání dané vzdálenosti zpět a vpřed při dané rychlosti	43
6.15	Stálé otáčení motoru vlevo a vpravo	44
6.16	Bloky pro stálé otáčení motoru ve vizualizaci.....	45
6.17	Funkce při dané rychlosti a zvoleném čase	46
6.18	Funkce při zvoleném čase i rychlosti pohybu	47
6.19	Vizualizace v programovacím prostředí SoMachine.....	48
6.20	Režim rozpadu plného a polovičního kroku [23]	49
6.21	Průběh proudu motoru pro plný krok	49
6.22	Průběh proudu motoru pro poloviční krok	50
6.23	Detail průběhu proudu motoru pro plný krok.....	51
6.24	Detail průběhu proudu motoru pro poloviční krok.....	51
6.25	Přechod ze záporné polaroty proudu do kladné pro plný krok	52
6.26	Průběh napětí proudu motoru pro poloviční krok	52
6.27	Detail průběhu proudu a napětí motoru pro poloviční krok	53
6.28	Detail průběhu proudu a napětí při zastavení motoru.....	53

SEZNAM TABULEK

4.1	Popis PLC modelu TM241CEC24T.....	23
4.2	Parametry PLC modelu TM241CEC24T	24
5.1	Legenda k primární obrazovce programu SoMachine.....	26
5.2	Legenda pro pracovní prostor Logic Builderu	28
5.3	Legenda pro grafické prostředí Vijeo-Designeru	30
6.1	Tabulka krokových režimů motoru	38

ÚVOD

V dnešní době se navrhují stále složitější aplikace a procesy, kdy jsou zapotřebí řídicí jednotky s větším počtem vstupů a výstupů. Programovatelné logické automaty jsou dnes hojně využívány zejména v automatizační technice díky nízkým nákladům a univerzálnosti. Cílem této bakalářské práce je seznámení a využití těchto programovatelných logických automatů pro výukové účely.

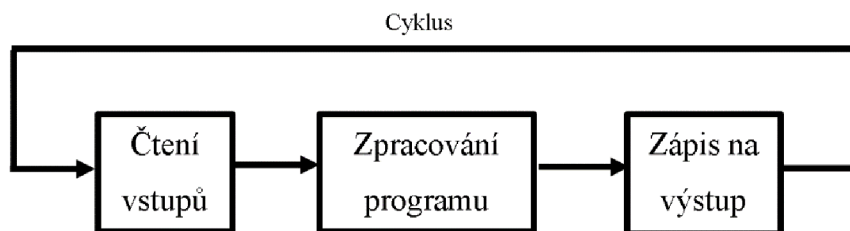
Hlavním účelem bakalářské práce je seznámení se s programovatelnými logickými automaty Schneider a následné vytvoření laboratorní úlohy. Ve spojitosti s tímto účelem je porovnání jednotlivých PLC výrobků od firmy Schneider Electric a vybrat nejvhodnější typ PLC, pomocí kterého bude navržena laboratorní úloha s krokovým motorem. Při realizaci této úlohy se spolu s navrženým programem pro řízení krokového motoru vytvoří příslušný laboratorní návod a vizualizace pro tuto úlohu.

Tato bakalářská práce je členěna do 6 kapitol. V první kapitole je seznámení s danou problematikou, historií a vybavení těchto automatů získaných z odborné literatury. Dále je zde struktura a přednosti PLC. Rozdělení, obecné typy a provedení těchto automatů je sepsáno v druhé kapitole. Ve třetí kapitole jsou sepsány a zobrazeny příklady programovacích jazyků, které jsou členěny dle normy do pěti skupin. Tato norma zahrnuje jazyk strukturovaného textu, jazyk seznamu instrukcí, jazyk příčkového diagramu, dále jazyk funkčního blokového schématu a sekvenční funkční diagram. Typy PLC a programovací prostředí do firmy Schneider Electric je podrobně ve čtvrté a páté kapitole, kde je mimo jiné i grafické znázornění tohoto prostředí a návod pro programování v něm. Závěrečná část této práce obsahuje samotný návrh a realizace laboratorní úlohy s krokovým motorem, sloužící pro lineární polohování tiskové hlavy. Dále jde zde zobrazen řídicí program s příslušným laboratorním návodem, vizualizace a naměřené průběhy proudu a napětí, které jsou v kapitole šesté.

1. PROGRAMOVATELNÉ LOGICKÉ AUTOMATY

Programovatelné logické automaty (Programmable Logic Controllers) jsou nejčastěji označovány jako PLC. Jsou to uživatelsky programovatelné řídicí systémy přizpůsobené k řízení technologických a průmyslových systémů a procesů. Typické PLC řeší úlohy logického typu, ale mohou sloužit také k řízení spojitéch veličin pomocí PID regulátorů [1].

Obecně automat, jak už napovídá název, má za úkol vykonávat předem danou úlohu samočinně. PLC vykonává automaticky program, který je vykonávám v periodách neboli v cyklech. Na začátku každého cyklu jsou načteny vstupy. Následně dojde ke zpracování a proběhnutí celého programu a po dokončení se nastaví výstupy. Tento proces se stále opakuje, dokud nedojde k zastavení nebo poruše [1]. Jednoduchý cyklus PLC lze vidět na obrázku 1.1.



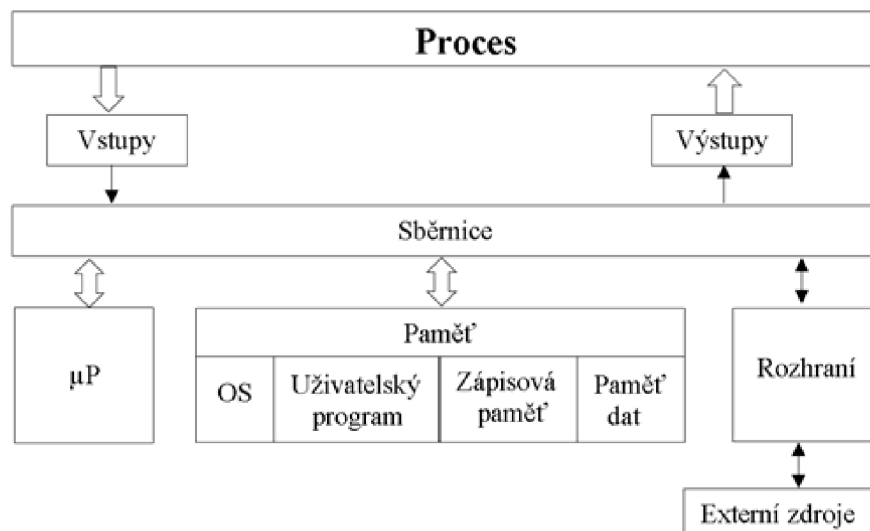
Obrázek 1.1 Cyklus PLC

1.1 Technické vybavení PLC

PLC obsahuje modulové jednotky, jako je centrální procesorová jednotka, systémová a uživatelská paměť, interface pro spojení s PC, a množství modulů pro vstupy a výstupy. Jádrem celého PLC je centrální procesorová jednotka (CPU), která obsahuje mikroprocesor pro rychlou realizaci souboru instrukcí (např. pro logické operace, aritmetiku a operaci s čísly) a systémových služeb. Systémové služby jsou prostředky poskytnuté od PLC nad rámec instrukčního souboru [2].

Paměťový prostor obsahuje paměť systémovou a paměť uživatelskou. Systémová paměť slouží pro čítače a časovače, uživatelské registry a proměnné (komunikační, časové a uživatelské). Do uživatelské paměti se ukládá uživatelský program, který se během vykonávání programu nemění [2].

Na vstupy PLC se přivádí binární (digitální) signály s dvouhodnotovým charakterem (např. tlačítka, spínače a přepínače) nebo signály analogové (např. snímače tlaku a teploty). Na výstupy jsou přiváděny také signály binární nebo analogové [2].



Obrázek 1.2 Bloková struktura PLC

1.2 Historie PLC

Koncem 60. let 19. stolní chtěla firma General Motors nahradit reléová zařízení a bezkontaktní logiku takovým zařízením, které by zpracovávalo binární informace. Zařízení mělo být spolehlivější, mechanicky odolnější a byla by možnost jej upravovat dle požadavků. Takto vzniklo první PLC na světě, pojmenované MODICON 084, které lze vidět na obrázku 1.3. Obsahovalo 256 vstupů/výstupů a ferritovou paměť o velikosti 32 kB, určenou pro operační systém a vlastní program [3].

Postupem času se rozjela velkovýroba PLC. Jako jádro se používaly bitové procesory, které byly velice rychlé oproti procesorům s 8 nebo 16 ti bitovým slovem. Proto se kladly na architekturu PLC následující nároky [3]:

- bitově orientovaná CPU a paměť dat
- slovně orientovaná paměť programu
- rozhraní na programovací přístroj
- instrukční soubor pro zpracování
- systém speciálních funkcí (časovače, čítače a další)

Takto vyrobené PLC se nezachovalo dodnes. U prvních PLC byla paměť programu oddělena od paměti dat a pro data paměti se používala jiná organizace paměti. U dnešních PLC je použita pouze jedna operační paměť, která obsahuje vstupní data, proměnné a paměťový prostor pro vlastní program. Dále jsou v paměti uloženy funkční bloky a veškeré funkce [3][4].



Obrázek 1.3 První PLC MODICON 084 [4]

1.3 Přednosti PLC

PLC jsou velice spolehlivé systémy i v extrémních podmínkách. Jsou navrženy tak, aby odolaly poruchám či rušení různého druhu. Jsou vybaveny diagnostickými funkcemi, které systém kontrolují (hlavně jeho činnost) a při lokalizaci závady usnadňují její odstranění. Hlavní předností PLC je rychlá realizace, protože stačí PLC sestavit s pomocí vhodných modulů, vytvořit program a vše realizovat. Pokud je jakýkoliv řídicí program chybný, tak použitím PLC stačí program upravit, změnit či rozšířit program. Toto je velký problém u řídicích systémů s pevnou logikou, jako je třeba relé. Častým problémem je nedostatek vstupů nebo výstupů. Je možnost využít rezervy v konfiguraci, nebo doplnit potřebné moduly či další PLC [1].

Programovatelné logické automaty mají dvě zásadní nevýhody. Zatímco u řídicích systémů s pevnou logikou je odezva na změnu vstupů v řádech nanosekund až mikrosekund, tak u PLC odezva bývá řádově jednotek až desítek milisekund. Odezva je dána dobou průchodu programu a závisí na rychlosti procesoru. Další nevýhodou je časová nespojitost zpracování, kdy algoritmus je vykonáván cyklicky jen v určitých okamžicích. V intervalu mezi okamžiky aktivace systém nereaguje na změny vstupních hodnot [1].

2. PROVEDENÍ PLC

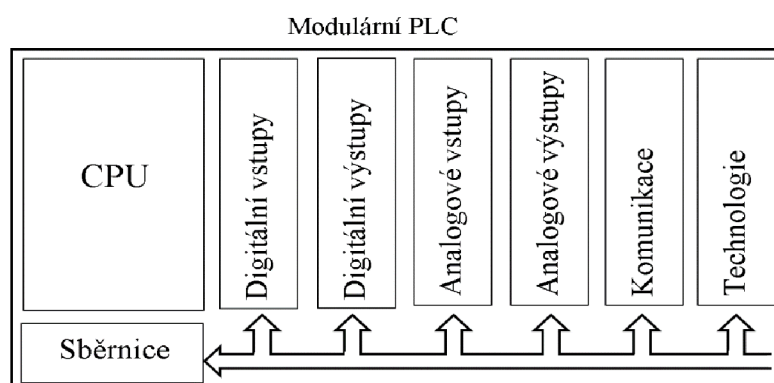
Při návrhu a výrobě PLC je možnost se setkat s různým provedením. Mezi hlavní provedení se řadí velikost, typ konstrukce a kolik procesorů dané PLC obsahuje. Menší systémy jsou konstruovány jako kompaktní, kde je možnost PLC rozšiřovat o další moduly. Větší systémy jsou řešeny jako modulární, kde je vše v jednom pouzdře a k rozšíření o další moduly dochází jen zřídka. Z hlediska počtu procesorů dělíme PLC na jednoprocesorové nebo víceprocesorové. PLC obsahující více procesorů může vykonávat několik procesů zároveň a většinou se konstruuje jako modulární [1].

Z hlediska velikosti se PLC dělí na čtyři typy [1][5]:

- Mikro PLC – je to nejmenší a nejlevnější kompaktní PLC, obvykle s pevnou soustavou binárních vstupů a výstupů. Používá se pro realizaci jednoduchých strojů a mechanismů.
- Malé PLC – je navrženo jako kompaktní jednotka, umístěná vedle ovládaného zařízení. Používá se k nahrazení pevně zapojené reléové logiky, čítačů a časovačů.
- Střední PLC – používají se v průmyslu, kde je mnoho modulů namontovaných na základní desce systému. Jsou schopny zpracovávat stovky vstupních a výstupních signálů.
- Velké PLC – kapacita těchto PLC je z hlediska paměti, programovacích jazyků, vstupů/výstupů a komunikačních modulů mnohem vyšší než u středních PLC. Používají se pro nejnáročnější aplikace.

2.1 Modulární PLC

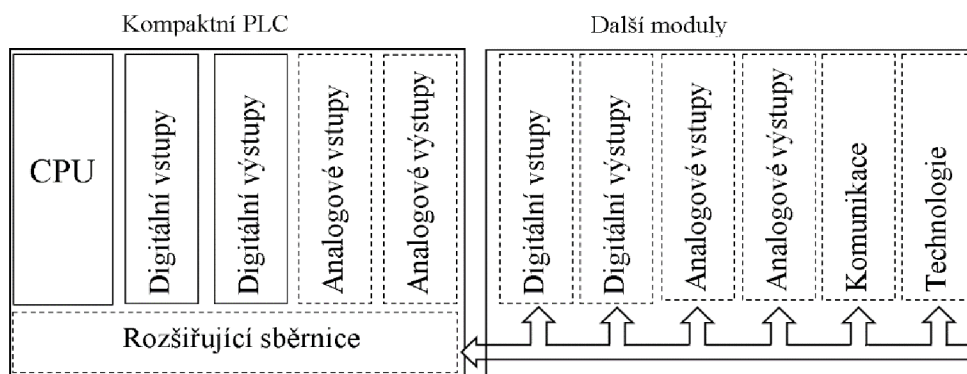
Modulární PLC je zařízení tvořeno několika moduly, které jsou sestaveny do jednoho celku. Základním modulem je CPU, ke kterému jsou dle potřeby připojeny další moduly, jako jsou vstupy a výstupy, technologické a komunikační schopnosti nebo speciálními moduly a další. CPU je vybaveno velkou kapacitou paměti pro program a data, protože se předpokládá zpracování většího počtu signálů. Pokud je potřeba změna, lze modulární systém snadno modifikovat [6].



Obrázek 2.1 Modulární PLC

2.2 Kompaktní PLC

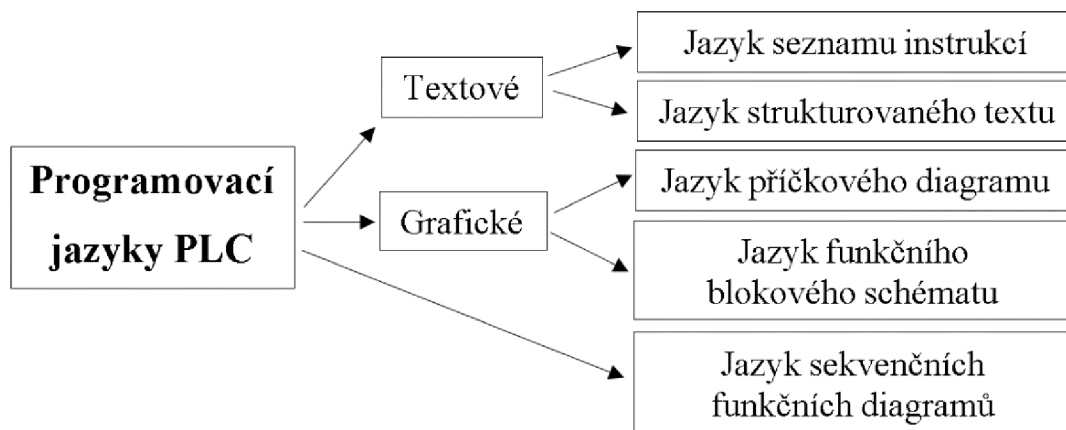
Kompaktní PLC obsahuje v jednom modulu CPU, veškeré vstupy a výstupy a základní komunikační rozhraní. Jsou určeny pro malé řídicí systémy a mají menší kapacitu paměti pro program i data oproti modulárním PLC. Další rozšíření, tedy připojení dalších modulů je omezené, kdy záleží hlavně na typu a výrobci. I přes toto omezení jsou v dnešní době hojně využívána. Hlavní důvod je nízká pořizovací cena a rychlost cyklu, kdy signály nemusí procházet přes řadič sběrnice [6].



Obrázek 2.2 Kompaktní PLC

3. PROGRAMOVACÍ JAZYKY PLC

PLC lze programovat mnoha způsoby v různých prostředí. Dle normy IEC EN 61131-3 je k dispozici pět programovacích jazyků, jak je patrné na obrázku 3.1. Tyto programovací jazyky se dělí na textové a grafické. Grafické rozhraní obsahuje bloky, kontakty, propojení a další útvary, kdežto textové rozhraní obsahuje obecné a speciální symboly. Mezi textové jazyky patří jazyk seznamu instrukcí (IL) a jazyk strukturovaného textu (ST). Mezi grafické jazyky patří jazyk příčkového diagramu (LD) a jazyk funkčního blokového schématu (FBD). Jako pátý se uvádí jazyk sekvenčních funkčních diagramů (SFC) [7].



Obrázek 3.1 Programovací jazyky PLC

3.1 Jazyk strukturovaného textu (ST)

Textový jazyk ST (Structured Text) je výkonný vyšší programovací jazyk. Je vhodný pro definování komplexních funkčních bloků, které pak mohou být použity v libovolném programovacím jazyku. Skladba jazyka je dána povolenými výrazy a příkazy. Výraz se skládá z operátorů a operandů. Operandem může být konstanta, proměnná, funkce nebo jiný výraz. Příkazů může být více na jednom řádku a jsou odděleny středníkem [7].

```

IF %M0 THEN
  FOR %MW99:=0 TO 31 DO
    IF %MW100[%MW99]<>0 THEN
      %MW10:=%MW100[%MW99];
      %MW11:=%MW99;
      %M1:=TRUE;
      EXIT; (* exit the loop *)
    ELSE
      %M1:=FALSE;
    END_IF;
  END_FOR;
ELSE
  %M1:=FALSE;
END_IF;

```

Obrázek 3.2 Příklad jazyka ST [8]

3.2 Jazyk seznamu instrukcí (IL)

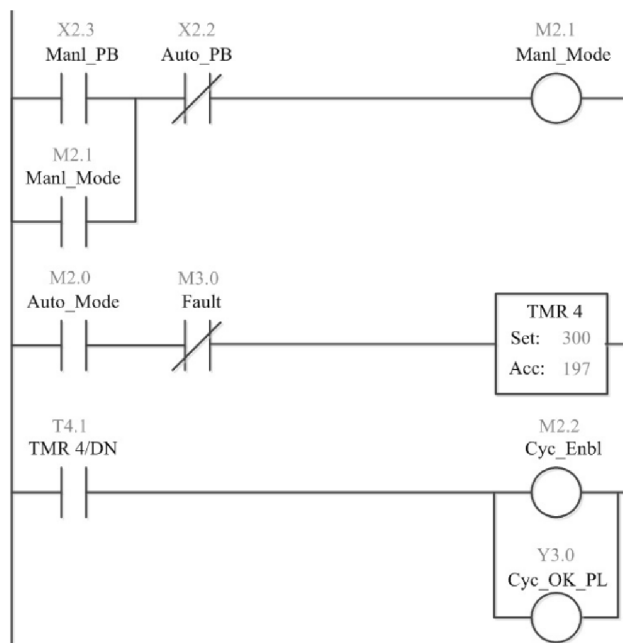
Textový jazyk IL (Instruction List) je nejzákladnější jazyk pro popis programu podobný assembleru. Program je skládán posloupností jednoduchých instrukcí. Každá instrukce může obsahovat návěští, operátor, operand a komentář. Operátor může být doplněn modifikátorem, který vyjadřuje podmíněnost, skoky, volání a návraty, prioritu a další [7].

U	" S11"
S	"Mode_A_S"
O	"_S15"
O	
U	"_S14"
UN	"_S11"
R	"Mode_A_S"
U	"Mode_A_S"
=	"Mode_A_S"

Obrázek 3.3 Příklad jazyka IL

3.3 Jazyk příčkového diagramu (LD)

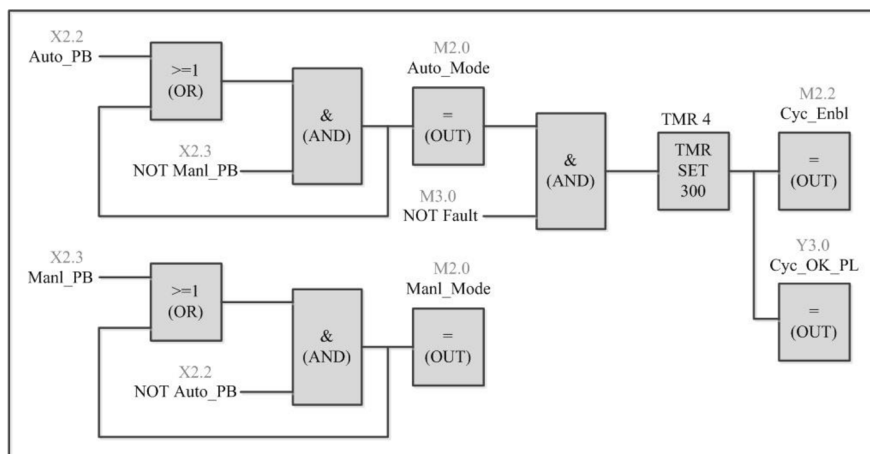
Grafický jazyk LD (Ladder Diagram), jinak označován jazykem kontaktních schémat, znázorňuje reléovou logiku. Organizační jednotka programu je vyjádřena sítí propojených grafických prvků. Síť je zleva i zprava ohraničena čarami (napájecími sběrnicemi). Příčka mezi sběrnicemi je vodorovná nebo svislá a může se větvit. Do příček se vkládají kontakty, cívky, funkce a funkční bloky [7].



Obrázek 3.4 Příklad jazyka LD [9]

3.4 Jazyk funkčního blokového schématu (FBD)

Grafický jazyk FBD (Function Block Diagram) je soubor vzájemně spojených grafických bloků, který zpracovává signály. Používají se funkční bloky, jako jsou čítače, časovače, komunikační bloky a jiné. Dle potřeby se mohou doplnit speciální bloky. Každý firma používá odlišný soubor bloků [7].

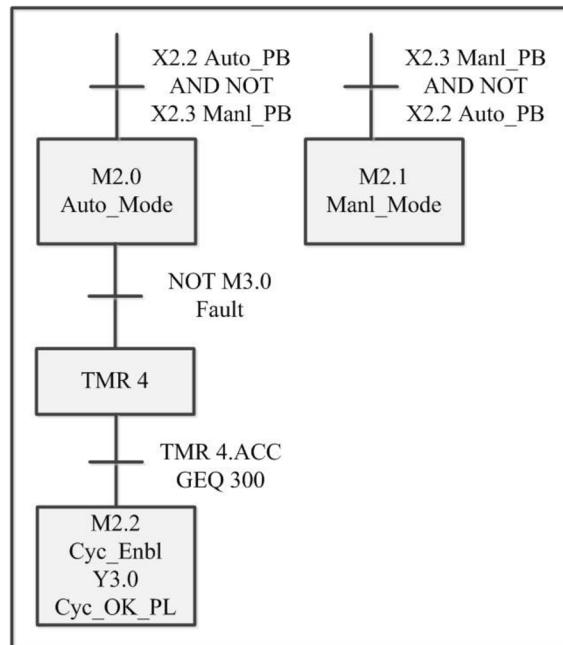


Obrázek 3.5 Příklad jazyka FBD [9]

3.5 Jazyk sekvenčních funkčních diagramů (SFC)

Jazyk sekvenčních funkčních diagramů SFC (Sequential Function Chart) popisuje sekvenční chování řídicího programu. Je odvozen ze symboliky Petriho sítí, ale grafické prvky se zde převádí přímo do souboru řídicích prvků. Tento jazyk vytváří struktury programu a umožňuje rozložit úlohu řízení na zvládnutelné části a zachovat přitom přehled o chování celku [7].

Sekvenční funkční diagram se skládá z kroků a přechodů. Každý krok reprezentuje stav řízeného systému. Přechod je spojen s podmínkami, které musí být splněny, aby mohl být deaktivován předcházející krok a aktivován následující krok. K základním strukturám patří lineární sekvence, alternativní větvení programu a souběh více větví s jejich následnou synchronizací [7].



Obrázek 3.6 Příklad jazyka SFC [9]

4. PLC FIRMY SCHNEIDER ELECTRIC

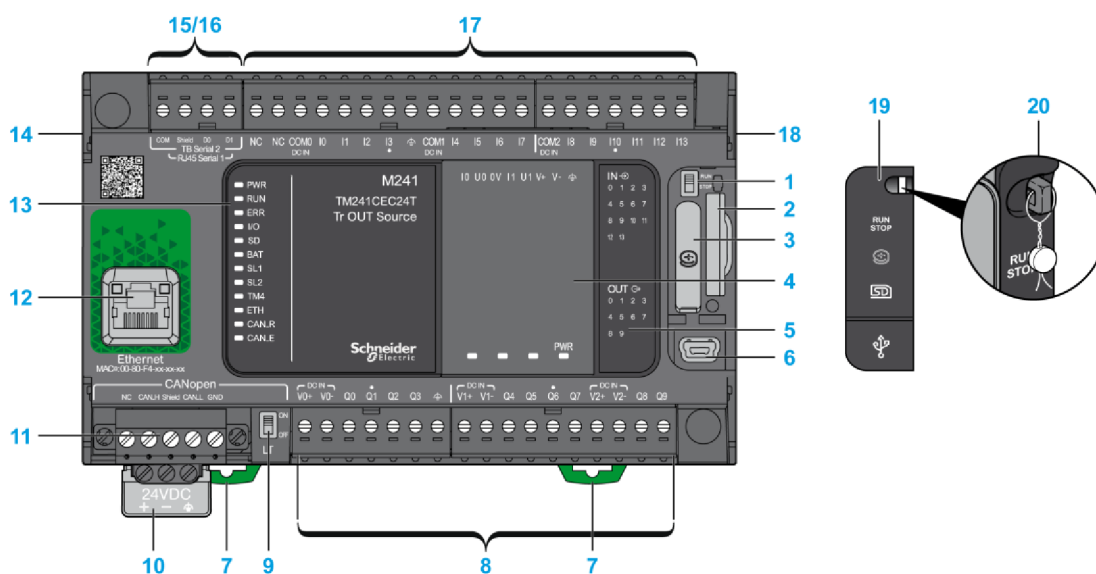
Firma Schneider Electric se zabývá výrobou zařízení pro průmyslovou automatizaci a pro přenos a rozvod elektrické energie. Nabízí celkem 7 různých programovatelných zařízení pro průmyslové stroje [10]:

- Zelio Logic – programovatelné relé určené pro řízení malých automatizačních systémů. Používá se v průmyslovém odvětví a automatizaci budov. Díky malým rozměrům a snadné instalaci nahrazuje reléovou techniku [11].
- Modicon M238 – kompaktní PLC vhodné pro řízení strojů s funkcemi rychlých čítačů. Je možné jej rozšířit až o 7 modulů a nabízí širokou konektivitu. Výhodou je snadná instalace připojením signálů na svorkovnice modulů nebo na inteligentní externí svorkovnici. Komunikace probíhá přes CANopen port nebo pomocí USB [12].
- Modicon M221 – vyrábí se v modulárním provedení nebo v provedení „book“, které šetří místo v rozvaděči. Má velký výkon a šetří náklady na provoz celého zařízení díky komunikačnímu rozhraní Ethernet [13].
- Modicon M251 – PLC malých rozměrů, které bylo vyvinuto na snížení nákladů a zvýšení výkonu. Je možné jej rozšířit až o 14 modulů. Díky dvoujádrovému procesoru pracuje s velmi krátkou dobou cyklu. Může komunikovat pomocí CANopen port, Ethernetu nebo Wi-Fi [14].
- Modicon M262 – kontrolér pro výkonné řízení servopohonů a logické řízení. Obsahuje protokoly IIoT s přenosem dat do cloudu včetně zabezpečení. K dispozici má 8 rychlých vstupů/výstupů. Komunikace probíhá přes CANopen port nebo přes Ethernet, kdy je možnost připojení až třech zařízení [15].
- Modicon M258 – kompaktní a velmi výkonné PLC navrženo k dosažení snadné a rychlé instalace a dalšího rozšíření. Lze ho použít pro jednoúčelové, dopravníkové a balící stroje, nebo k diskrétnímu řízení [16].
- Modicon M241 – blíže popsán v kapitole 4.1.

Pro tyto zařízení byly vyvinuty dvě programovací prostředí, a to Zelio Soft 2 a EcoStruxure Machine, nazývané jako SoMachine. Pro programovací prostředí Zelio Soft je navrženo programovatelné relé Zelio Logic. V programovacím prostředí SoMachine lze programovat veškeré zbylé kontroléry Modicon [10].

4.1 PLC řady Modicon M241

Posledním typem od firmy Schneider Electric je PLC řady Modicon M241, konkrétně model TM241CEC24T, které lze vidět na obrázku 4.1. Toto PLC bylo vybráno a použito v této práci. Jako jediné ve své třídě nabízí pět komunikačních portů. Stejně jako PLC typu M251 bylo navrženo na zvýšení výkonu a ke snížení nákladů. Obsahuje nejmodernější technické vlastnosti a přes CANopen port lze řídit až 63 zařízení. Výkon dvoujádrového procesoru umožňuje PLC pracovat s velmi krátkou dobou cyklu [17].



Obrázek 4.1 PLC model TM241CEC24T [18]

Pro komunikaci obsahuje PLC dvě sériové linky, CANopen port, rychlejší komunikační port mini USB (umístěno pod ochranným obalem), nebo komunikační port Ethernet. Indikační LED diody slouží pro detekci stavů vstupů a výstupů. Stavové LED diody signalizují stav PLC, závady, připojení a další. Pod ochranným obalem je dále přepínač Run/Stop, slot pro SD kartu o kapacitě do 16 GB a integrovaná baterie [19]. Popis komponentů PLC je znázorněn v tabulce 4.1.

Tabulka 4.1 Popis PLC modelu TM241CEC24T

Označení	Popis
1	Přepínač Run/Stop
2	Slot pro SD kartu
3	Držák baterie
4	Slot kazety
5	Indikační LED diody vstupů/výstupů
6	USB mini-B port pro připojení k PC
7	Zámek k připnutí na DIN lištu
8	Výstupy PLC
9	Přepínač CANopen portu
10	Připojení k napájení
11	CANopen port
12	Ethernet port
13	Stavové LED diody
14	Konektor sběrnice typu TM4
15/16	Sériový port linky
17	Vstupy PLC
18	Konektor sběrnice typu TM3/TM2
19	Ochranný kryt
20	Úchyt pro zámek

PLC je napájeno jmenovitým stejnosměrným napětím 24 V. Obsahuje 24 diskretních vstupů/výstupů. Napěťový signál přivedený na vstupy je vnímán jako logická jedna od hodnoty 15 V do hodnoty 30 V. PLC je vybaveno ochranou proti zkratu a přetížení [19]. Parametry tohoto PLC jsou sepsány v tabulce 4.2.

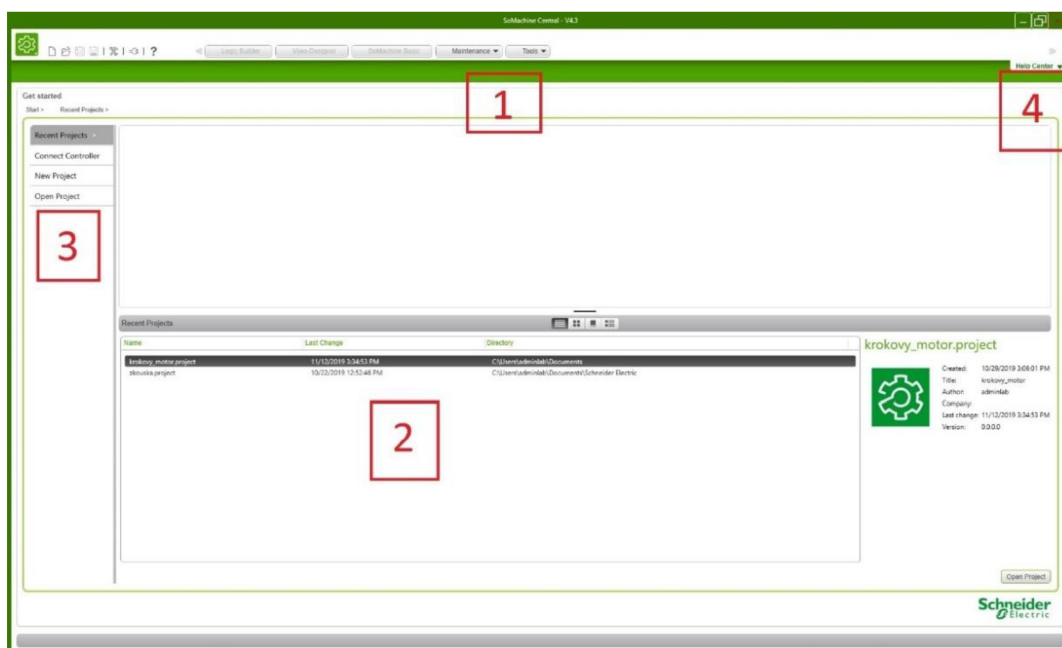
Tabulka 4.2 Parametry PLC modelu TM241CEC24T

Napájecí napětí	24 V DC
Počet vstupů	14, z toho 8 rychlých
Počet výstupů	10, z toho 4 rychlé
Typ výstupu	Tranzistorový
Výstupní napětí	24 V DC
Výstupní proud	500 mA, pro rychlý výstup 100 mA
Vstupní proud	5 mA, pro rychlý vstup 10,7 mA
Zapínací proud	50 A
Maximální výstupní frekvence	1 kHz, pro rychlé výstupy až 100 kHz
Kapacita paměti	8 MB pro program
	64 MB na paměti RAM
	128 MB pro zálohování

5. PROGRAMOVACÍ PROSTŘEDÍ SOMACHINE

SoMachine je profesionální a efektivní programovací prostředí od firmy Schneider Electric, které konfiguruje a uvádí do provozu celý stroj v jediném prostředí. Je to intuitivní řešení pro automatizaci strojů se všemi funkcemi, které jsou potřeba pro sestavení strojů. SoMachine je schopen použít šest standardních jazyků. Jedná se o jazyky SFC, LD, FBD, ST, IL a jazyk volně propojovaných bloků CFC (Continuous Function Chart). Tento program se skládá ze dvou částí. První část tvoří software Logic Builder, ve kterém se může naprogramovat samotné PLC. Druhou část tvoří software Vijeo Designer, který slouží k vytváření vizualizace a ke konfiguraci pro panely HMI. Součástí tohoto softwaru je další programovací prostředí SoMachine Basic, které je obdobné jako Logic Builder. Ovšem toto prostředí je určené pouze pro PLC řady Modicon 221 od firmy Schneider Electric [20].

Při spuštění programu SoMachine se zobrazí primární obrazovka s funkcemi pro řízení projektu, jak je patrné na obrázku 5.1. Popis jednotlivých oblastí je sepsán v tabulce 5.1.



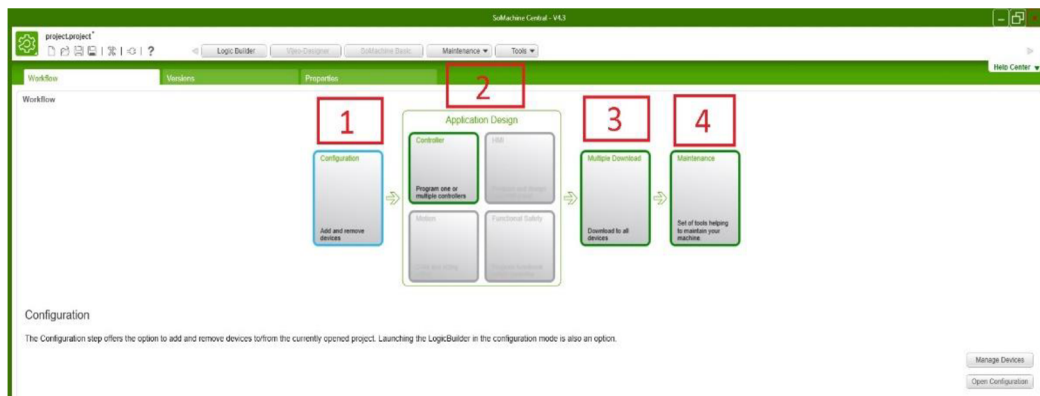
Obrázek 5.1 Primární obrazovka programu SoMachine

Tabulka 5.1 Legenda k primární obrazovce programu SoMachine

Oblast	Popis
1	Domovská stránka
2	Seznam předchozích projektů
3	Dostupné operace
4	Centrum nápovědy

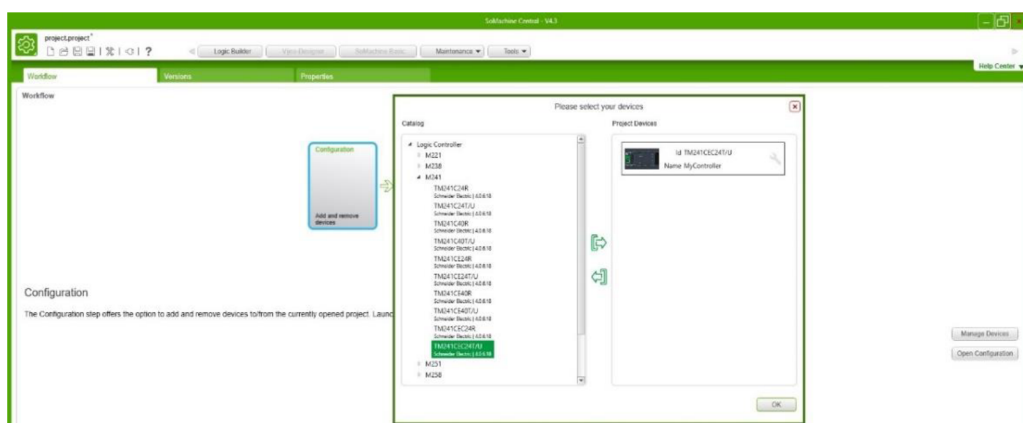
V poli 3 je k dispozici výběr ze 4 operací. Při výběru *Recent Projects* se zobrazí přechozí projekty. Po volbě operace *Connect Controller* je možnost se snadno připojit k PLC nebo HMI, pokud ještě není připojen žádný kontrolér. Dále je na výběr vytvořit nový projekt kliknutím na *New Project*, nebo otevřít existující projekt výběrem *Open Project*. Při vytvoření nového projektu je uživatel přesměrován do tzv. pracovního prostoru projektu, který je znázorněn na obrázku 5.2. Obrazovka pracovního postupu poskytuje grafické znázornění celého projektu. Zde je několik důležitých částí [20]:

1. *Configuration* – možnost výběru nebo odebrání zařízení
 - Po otevření této sekce se zobrazí katalog aktuálních zařízení. Jestliže je přidáno zařízení, dojde k nahrání důležitých knihoven a program bude moci pracovat s tímto zařízením.
2. *Application Design* – pro vytvoření celého projektu
 - V sekci *Controller* se vytváří samotný program v prostředí Logic Builder. V sekci *HMI* se vytváří vizualizace v prostředí Vijeo Designer. *Motion* a *Functional Safety* jsou nástroje jen pro určité typy kontrolérů.
3. *Multiple download* – zde dochází k nahrání programu do všech připojených zařízení.
4. *Maintenance* – zde jsou nástroje pro správu PLC.



Obrázek 5.2 Obrazovka pracovního prostoru projektu

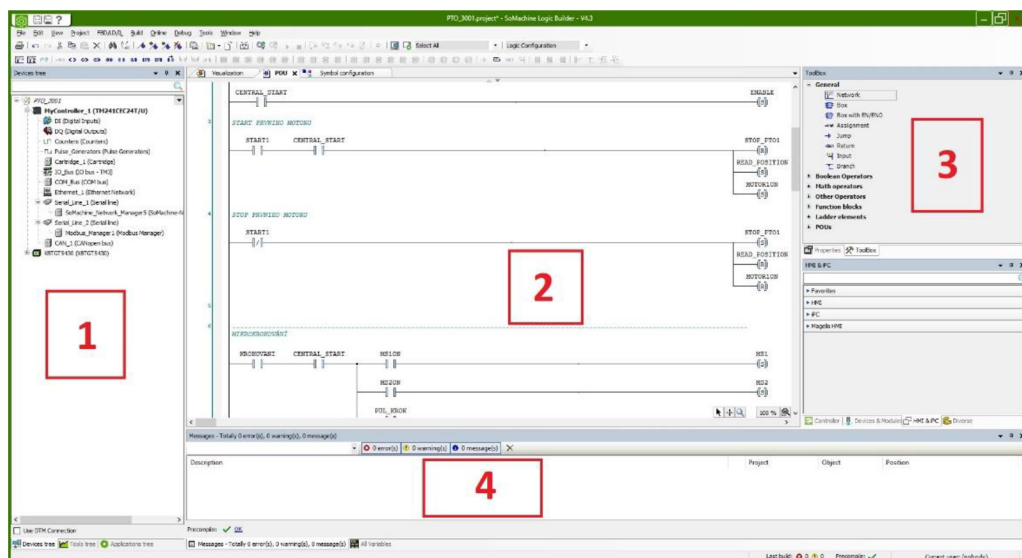
V sekci *Configuration* se vybere požadované zařízení. Na obrázku 5.3 lze vidět kontrolér, který byl použit v této páci, tedy PLC řady M241 typ TM241CEC24T. Po přidání tohoto kontroléru lze přejít na samotné programování v prostředí Logic Builder výběrem sekce *Controller*.



Obrázek 5.3 Výběr PLC typu M241

5.1 Logic Builder

Jak už bylo zmíněno, programovací prostředí Logic Builder slouží pro naprogramování samotného PLC. Je určeno pro programování všech PLC řady Modicon. Zde se píše vlastní program, který je po vytvoření poslán do PLC [20]. Po otevření tohoto prostředí vyskočí pracovní prostor Logic Builderu, který lze vidět na obrázku 5.4 a popis pracovního prostoru je sepsán v tabulce 5.2.



Obrázek 5.4 Pracovní prostor Logic Builderu

Tabulka 5.2 Legenda pro pracovní prostor Logic Builderu

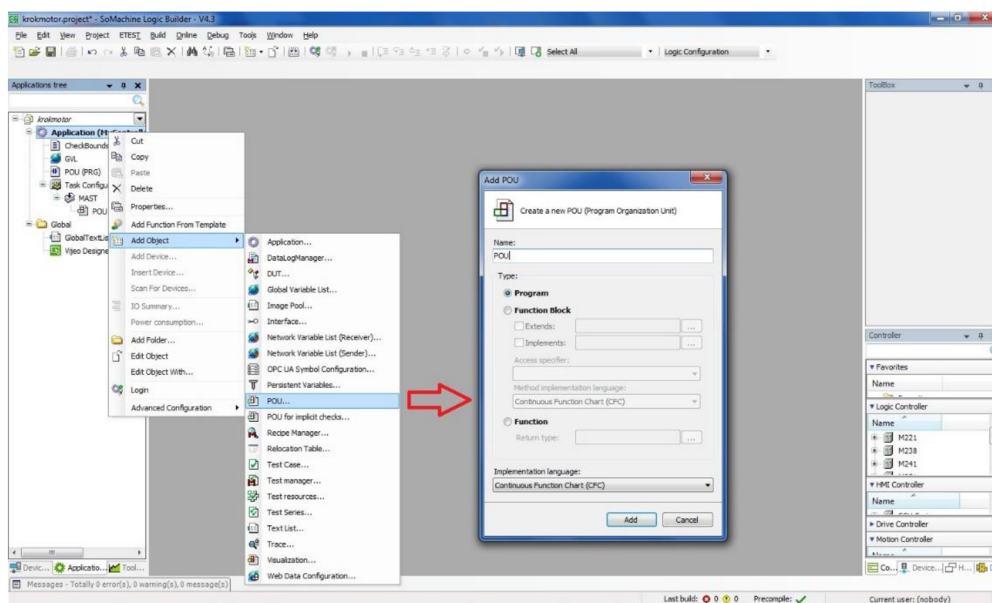
Oblast	Popis
1	Záložky pro správu PLC a programu
2	Pracovní oblast
3	Prvky pro přesun do pracovního prostoru
4	Zprávy upozornění

V oblasti 1 jsou tři důležité záložky. *Devices Tree* neboli strom zařízení, který slouží ke konfiguraci hardwaru aktuálního řadiče. Nachází se zde veškerá zařízení, která byla přiřazena do projektu. V této záložce se přiřazují vstupy a výstupy zařízení, čítače, pulzní generátor a další. Možností je i nakonfigurovat mapování vstupů/výstupů. Další záložkou je *Application Tree* neboli strom aplikací. Zde dochází k vytvoření samotného programu. Poslední záložkou je *Tools Tree* neboli strom nástrojů, kde jsou k dispozici správce knihovny a informace o projektu. Zde se zobrazují knihovny, které jsou použity v projektu a můžou se přidávat i další knihovny [20].

Oblast 3 tvoří tzv. *ToolBox* neboli sada nástrojů. Ta obsahuje různé funkce a prvky, které lze přesunout do pracovní oblasti a přidat do vlastního programu. Je zde na výběr z několika kategorií funkcí, jako jsou funkční bloky, časovače a čítače, matematické operátory a další. V kartě *Messages*, jak ukazuje oblast 4, se vypisují zprávy upozorňující na nějakou chybu [20].

5.1.1 Vytvoření a nahrání projektu do PLC

Při vytvoření programu se nejdříve musí přidat objekt *POU*, jak lze vidět na obrázku 5.5. Po přidání tohoto objektu se zvolí název, typ (Program, Funkční blok nebo Funkce) a programovací jazyk dle vlastního uvážení. Po potvrzení se přechází do pracovní oblasti, kde dochází k vytváření proměnných a psaní vlastního programu. Po napsání programu je vhodné propojit nově vytvořené proměnné se vstupy a výstupy v záložce *Devices Tree*. Jestliže je program vytvořen, následuje kompilace projektu výběrem *Build - Build* v horní části menu. Následně se musí vložit objekt *POU* do *MAST* (Master Task). To se provede jednoduchým přetažením [20].



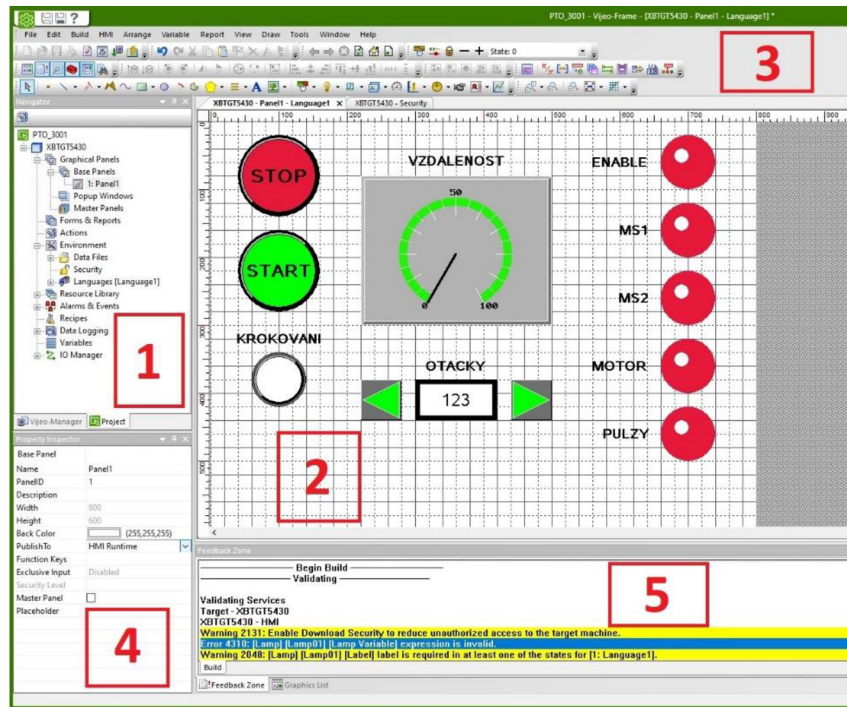
Obrázek 5.5 Přidání objektu POU

Pokud se splnily předchozí kroky, může se program vyzkoušet v simulaci. Simulace je užitečná, pokud se chtějí sledovat stavy proměnných. Před nahráním programu do PLC je nutné vybrat zařízení, které je programováno, v záložce *Device Tree* a potvrdit výběr tohoto zařízení. Simulace se provede výběrem *Simulation* v horním menu *Online* a v dolní liště se zobrazí červený nápis *Simulation* [20].

Dále se vybere ze stejného menu *Login*, kdy dojde k nahrání programu do PLC. Další způsob nahrání programu je výběr *Multiple Download* ze stejného menu, kdy dojde k nahrání programu do všech připojených zařízení. Nakonec se spustí program v sekci *Debug* a tlačítkem *Start* nebo klávesou F5 [20].

5.2 Vijeo Designer

Vijeo Designer je grafické prostředí k vytváření vizualizace a konfiguraci pro panely HMI. Jednotlivé proměnné je nutné propojit s grafickými prvky v tomto softwaru a je možnost sledovat chování těchto proměnných při chodu programu. Tedy proměnné mohou daný prvek řídit, nebo naopak, daný prvek může řídit proměnnou (SET, RESET) [21]. Toto grafické prostředí je zobrazeno na obrázku 5.6. Jednotlivé oblasti Vijeo Designeru jsou sepsány v tabulce 5.3.



Obrázek 5.6 Grafické prostředí Vijeo-Designer

Tabulka 5.3 Legenda pro grafické prostředí Vijeo-Designeru

Oblast	Popis
1	Strukturální strom projektu
2	Pracovní prostor
3	Hlavní menu a nástroje
4	Menu pro výběr komponentů
5	Nastavení vlastností komponentů
6	Zpětná vazba a grafický seznam

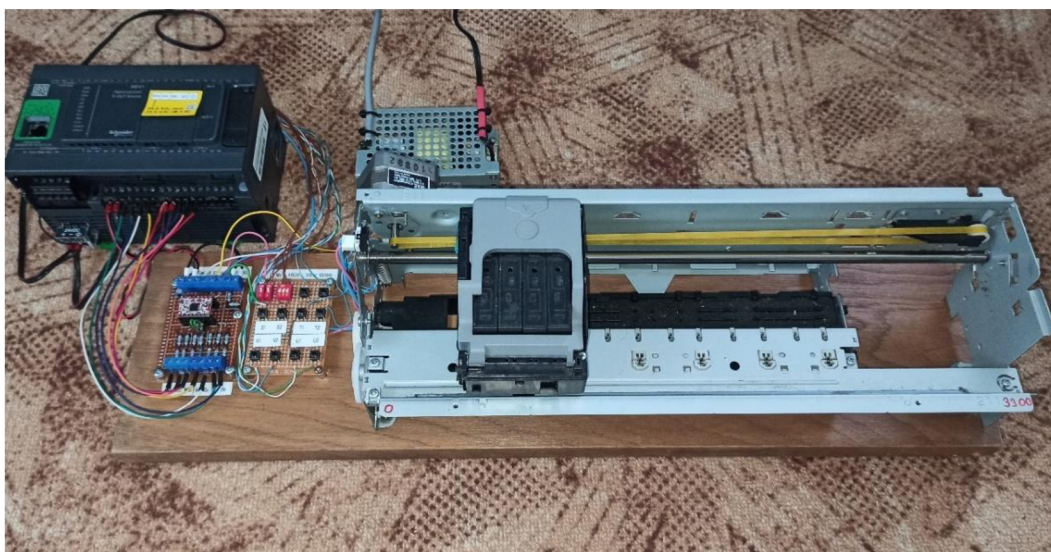
V oblasti 1 jsou veškeré vlastnosti HMI panelu, které mohou být konfigurovány. Oblast 2 představuje hlavní pracovní prostor, kam jsou nahrávány vizualizační prvky (přepínač, žárovka, numerický displej a další). Tyto vizualizační prvky a další prvky pro práci s grafikou panelu jsou na výběr v sekci 3. Vlastnosti vložených komponentů do pracovního prostoru, které mohou být libovolně upravovány, jsou znázorněny v oblasti 4. V oblasti 5 se zobrazuje zpětná vazba, která dává informaci po kompilaci projektu, nebo grafický seznam všech použitých komponentů [21].

Jestliže je do projektu připojen HMI panel, je důležité při použití tohoto prostředí nahrát proměnné vytvořené v projektu do HMI panelu. V záložce *Tools Tree* přidáme do *Application* objekt s názvem *Symbol Configuration* a následně se provede kompilace vybraných proměnných tlačítkem *Build*. Po vytvoření vizualizace je důležité vyplnit jména zařízení v sekci *IO Manager*, aby došlo ke spárování HMI panelu s PLC. Nahrání projektu do panelu se provede stejným způsobem, jak bylo popsáno v kapitole 5.1.1, tedy jakmile dojde k nahrání programu do PLC [21].

6. REALIZACE LABORATORNÍ ÚLOHY

Součástí této bakalářské práce bylo navrhnout laboratorní úlohu s krokovými motory, která by se využila ve výuce. Cílem této práce je seznámení se s programovatelnými logickými automaty Schneider a programovacím prostředím SoMachine. Současně bude vytvořen příslušný laboratorní návod a vizualizace pro tuto úlohu.

Při návrhu této úlohy byla použita tisková hlava z tiskárny, integrovaný driver krokového motoru A4988 a dvě vyrobené desky plošných spojů, které slouží pro ovládání úlohy. Kompletní model pro laboratorní úlohu lze vidět na obrázku 6.1. Pro polohování tiskové hlavy je použit krokový motor MAE HY 200.



Obrázek 6.1 Model pro laboratorní úlohu

6.1 Laboratorní úloha s krokovým motorem

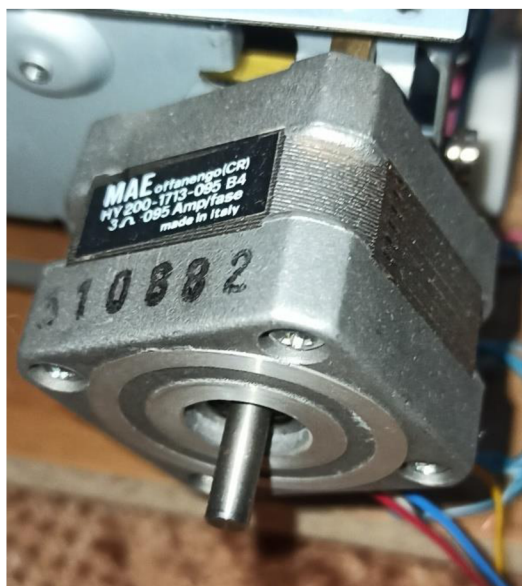
Laboratorní úloha je realizována jako polohovací lineární systém pro tiskovou hlavu tiskárny. Tisková hlava je uložena na plechové konstrukci se stabilizační tyčí. K této konstrukci je rovněž připevněn krokový motor. Na tiskové hlavě je umístěn gumový řemínek, který pomocí krokového motoru a ozubeného kolečka slouží pro pohyb této tiskové hlavy směrem vlevo a vpravo.

Cílem při realizaci této úlohy bylo lineární polohování pro tiskovou hlavu. Za tímto účelem byl vytvořen program v programovacím prostředí SoMachine, kterým je ovládán pohyb krokového motoru a zároveň i pohyb tiskové hlavy. Tisková hlava na povel vykoná dané funkce, jako je najetí do přesně dané pozice při zvolené rychlosti, ujetí určité vzdálenosti, pohyb po zvolenou dobu, nebo pohyb v různém režimu krokování motoru.

Tyto funkce lze spouštět přes jednoduché vizualizační prostředí, nebo přes externí tlačítka umístěná na desce plošných spojů. Ve vizualizačním prostředí se nastavuje například pomocí virtuálního potenciometru požadovaná rychlost pohybu motoru, dále doba pohybu a vzdálenost pro pohyb motoru. Funkce pro pohyb tiskové hlavy a vizualizační prostředí jsou znázorněny v dalších kapitolách. Kompletní návod pro tuto laboratorní úlohu je sepsán v Příloze C.

6.2 Krokový motor HY 200 1713

Pro realizaci této úlohy byl zvolen hybridní krokový motor od firmy MAE, který je zobrazen na obrázku 6.2. Jedná se o dvoufázový bipolární motor typu HY 200 1713, který má impedanci 3Ω a maximální proud $0,95 \text{ A}$ na fázi. Úhel jednoho kroku tohoto motoru je $1,8^\circ$, což odpovídá 200 krokům na otáčku [22]. Detailní informace pro tento motor jsou k dispozici v Příloze A.



Obrázek 6.2 Krokový motor HY 200

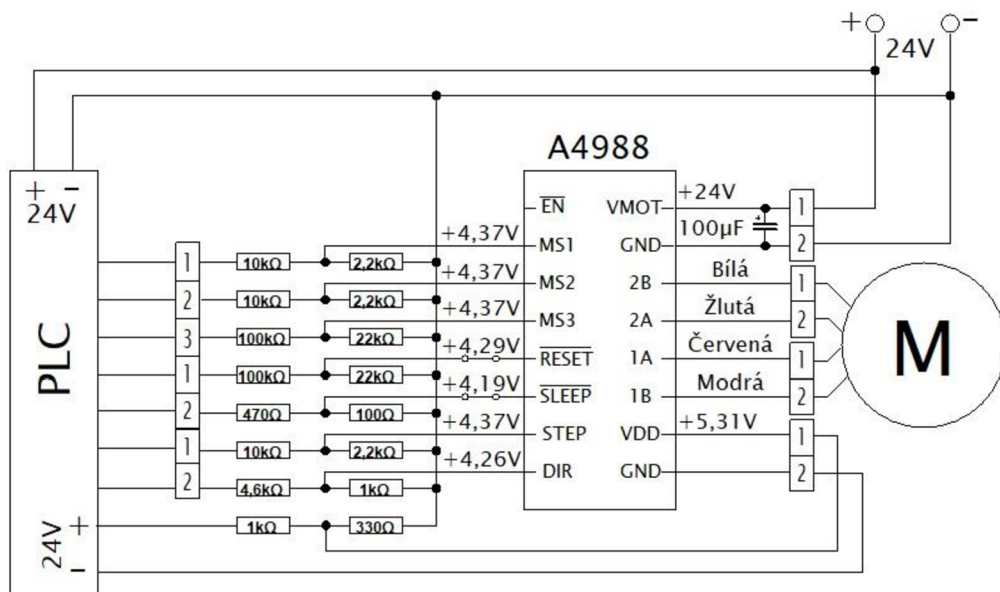
6.3 Integrovaný driver krokového motoru A4988

Pro ovládání krokového motoru byl vybrán integrovaný driver A4988 od firmy Allegro, který je určen pro bipolární řízení krokových motorů. Aby byl tento driver v provozu, potřebuje mít stejnosměrné napětí v rozmezí $3 - 5,5 \text{ V}$ pro logickou část. Pro napájení motoru je potřeba stejnosměrné napětí v rozmezí $8 - 35 \text{ V}$, kdy do motoru je obvod schopen dodat proud až $\pm 2 \text{ A}$ [23]. Detailnější informace k tomuto driveru jsou k dispozici v Příloze B.

Driver je vybaven tepelnou a proudovou ochranou, regulátorem proudu a potenciometrem, kterým se nastavuje efektivní hodnota proudu motorem. Umožňuje řízení motoru v pěti krokových režimech. Integrovaný driver obsahuje celkem 8 logických vstupů [23]:

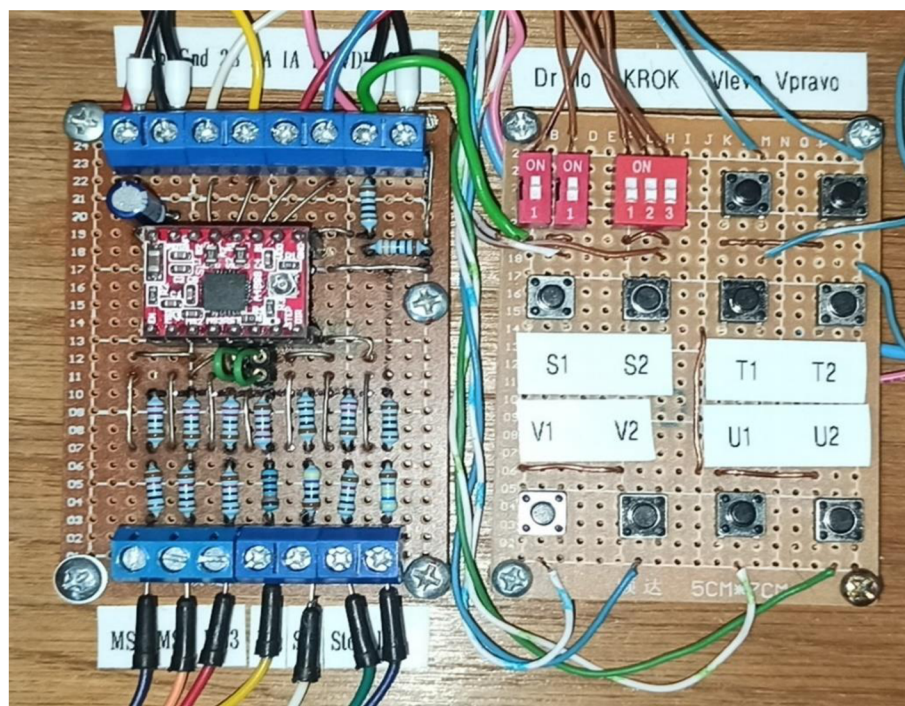
- $\overline{\text{Enable}}$ = Vypíná nebo zapíná tranzistorové FET výstupy, aktivní v logické nule.
- $\overline{\text{MS1}}$, $\overline{\text{MS2}}$, $\overline{\text{MS3}}$ = Mikrokrokování motoru, viz. Tabulka 6.1.
- $\overline{\text{Reset}}$ = Vypíná výstupní FET tranzistory a provádí reset překladače, Aktivní v logické nule.
- $\overline{\text{Sleep}}$ = Režim spánku, minimalizuje spotřebu energie při neaktivním motoru. Vypíná výstupní FET tranzistory a proudový regulátor, aktivní v logické nule.
- Step = S každou nástupnou hranou na tomto vstupu vykoná motor jeden krok.
- Dir = Určuje směr otáčení motoru.

Protože napájecí napětí a napětí na výstupech PLC je 24 V DC, byly použity děliče napětí, které přivedou na vstupy integrovaného driveru napětí v požadovaném rozsahu 3 – 5,5 V. Schéma zapojení DPS s integrovaným driverem A4988 a PLC je na obrázku 6.3.



Obrázek 6.3 Schéma zapojení s integrovaným obvodem A4988

Jak už bylo zmíněno, byly vytvořené dvě DPS pro ovládání krokového motoru. První deska obsahuje tlačítka, která jsou připojena na vstupy PLC a pomocí nich se ovládají různé proměnné obsažené v programu. Druhá deska je osazena právě tímto integrovaným driverem. Tyto dvě desky plošných spojů jsou zobrazeny na obrázku 6.4.



Obrázek 6.4 DPS s tlačítky a integrovaným obvodem A4988

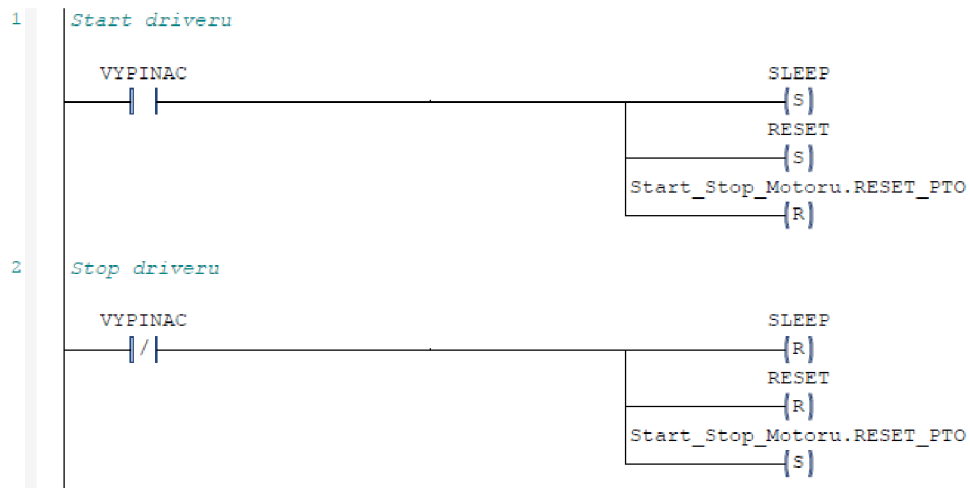
6.4 Hlavní program v prostředí SoMachine

Při realizaci programu pro krokový motor byla zvolena funkce PTO pulzního generátoru. Takto jsou posílány na jeden výstup PLC pulzy a na druhý výstup informace o směru otáčení pro motor. Následně bylo navrženo sedm programů, které jsou psány v jazyce LD a spolu tvoří hlavní program pro řízení krokového motoru a tiskové hlavy. Jedná se o programy pro povolení driveru A4988, povolení pohybu motoru, krokování motoru a čtení aktuální pozice motoru.

Pro pohyb motoru a směr otáčení slouží blok pro zasílání neomezeného počtu pulzů do motoru (Move Velocity), blok pro vykonání určité vzdálenosti motoru (Move Relative) a blok pro najetí motoru do dané pozice (Move Absolute). Pro všechny tyto bloky byly vytvořeny zbylé programy s různými funkcemi, které jsou sepsány v následujících kapitolách.

6.4.1 Program pro integrovaný obvod A4988

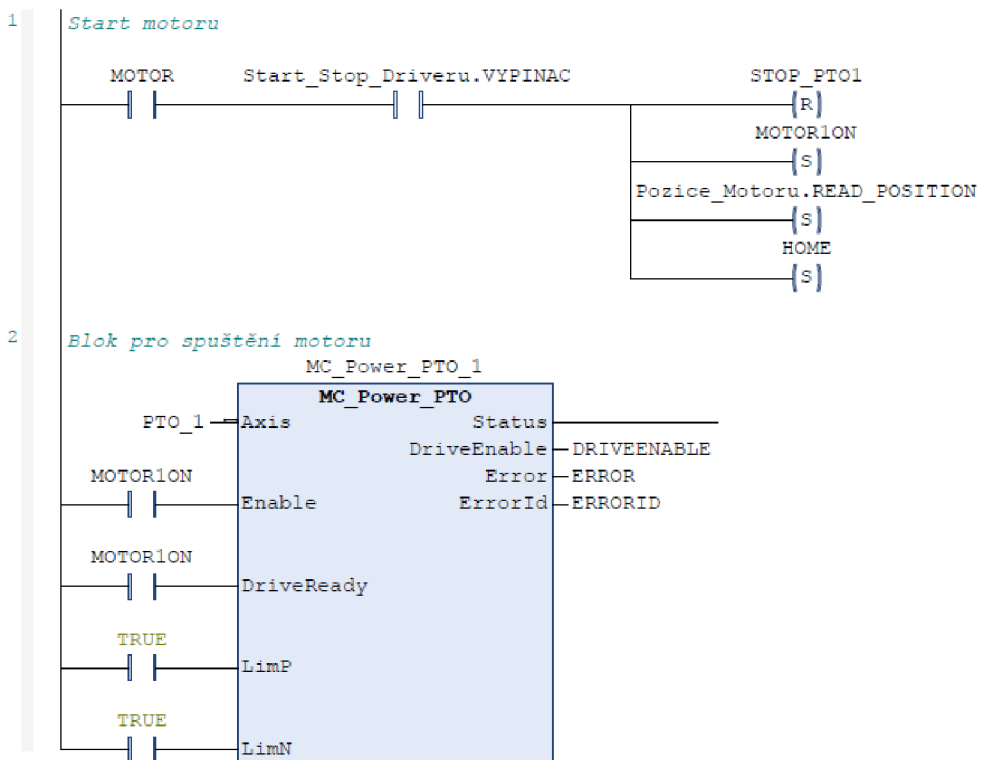
Aby bylo možné používat tento obvod, je nutné nastavit jeho logické výstupy \overline{SLEEP} a \overline{RESET} na hodnotu logická 1, aby byl v aktivním stavu. V programovacím prostředí proto bylo vytvořeno tlačítko *VYPINAC*, který povolí tyto dva výstupy a provede reset pro případ, kdyby se vyskytla chyba v programu. Toto tlačítko, na DPS označeno jako *Dr*, je připojeno na vstup PLC.



Obrázek 6.5 Program pro integrovaný obvod A4988

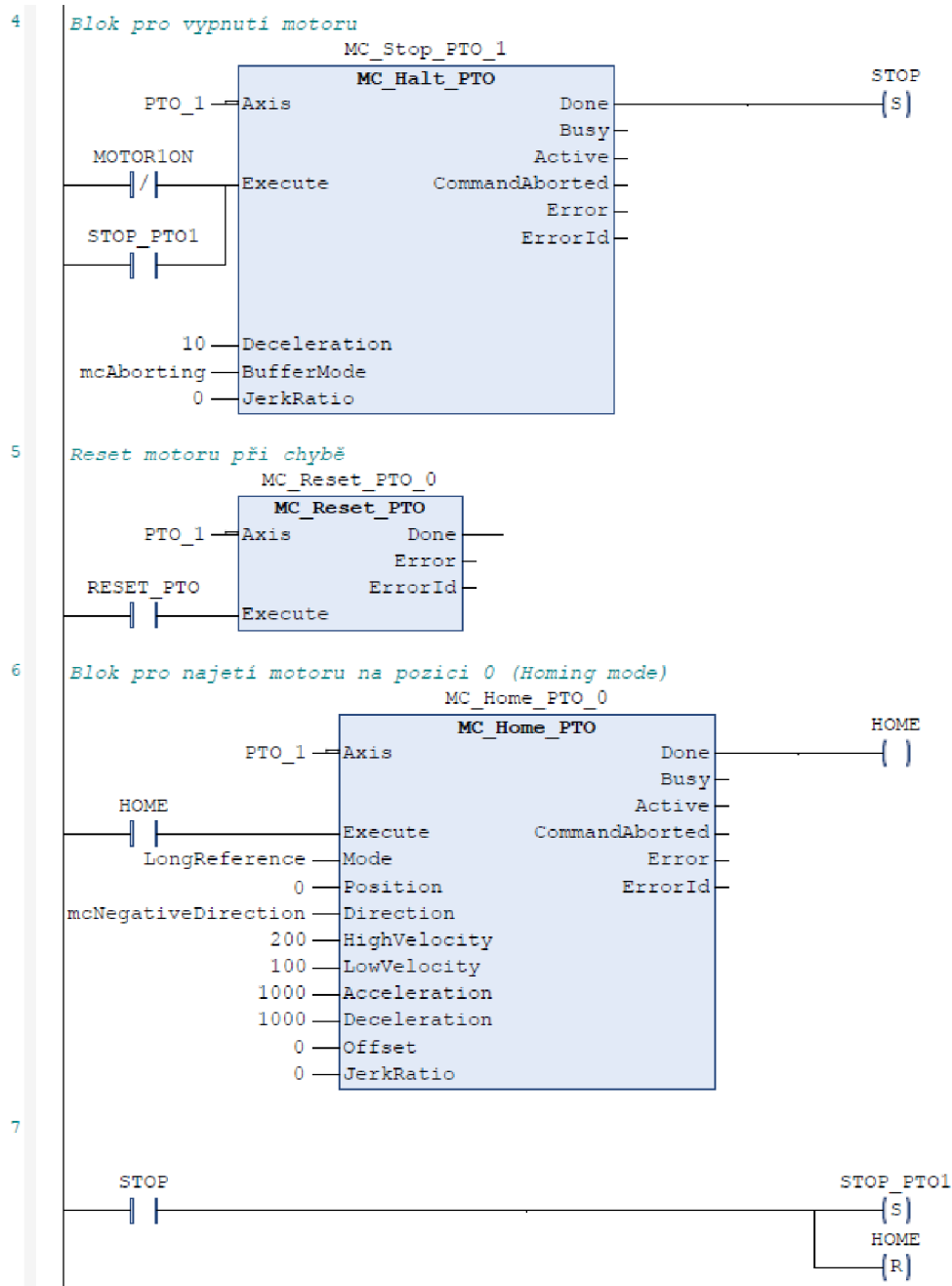
6.4.2 Program pro povolení pohybu motoru

V programovacím prostředí existuje blok, nazývaný *MC_Power_PTO*, který slouží pro povolení pohybu motoru po ose. Při zapnutí tlačítka *MOTOR*, které je taktéž připojeno na vstup PLC a na DPS označeno jako *Mo*, dojde k aktivaci tohoto bloku a tím i k povolení pohybu motoru po ose.



Obrázek 6.6 Program pro povolení pohybu motoru

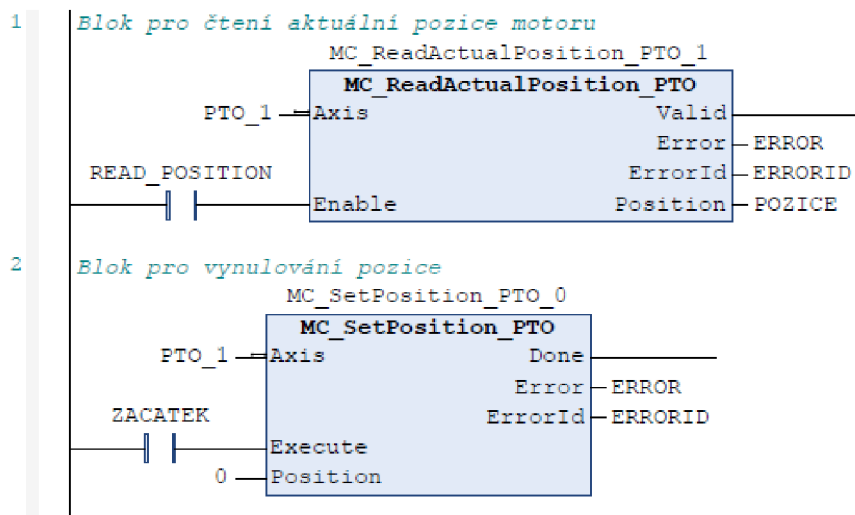
Současně při zapnutí tohoto tlačítka se povolí blok pro čtení pozice motoru, který je zobrazen v kapitole 6.4.3. Dále se spustí tzv. *Homing Mode*, kdy otáčením motoru najede tisková hlava směrem ke spínači, který představuje proměnnou STOP. Po sepnutí tohoto spínače se aktivuje blok pro zastavení motoru a tím tisková hlava dojde na nulovou pozici, ať se nachází v jakékoliv pozici. Pro vypnutí pohybu motoru je v prostředí určen blok *MC_Halt_PTO*.



Obrázek 6.7 Homing Mode, Reset a Stop pohybu motoru

6.4.3 Program pro čtení a vynulování pozice motoru

Jak bylo zmíněno v předchozí kapitole, tak existuje blok pro čtení aktuální pozice osy motoru. Do vizualizace se proto přidal měřič, který zobrazuje aktuální pozici motoru. Na ukázkou je v programu přidán blok, kdy při stisknutí tlačítka ZACATEK dojde k vynulování pozice, jak naznačuje obrázek 6.8.



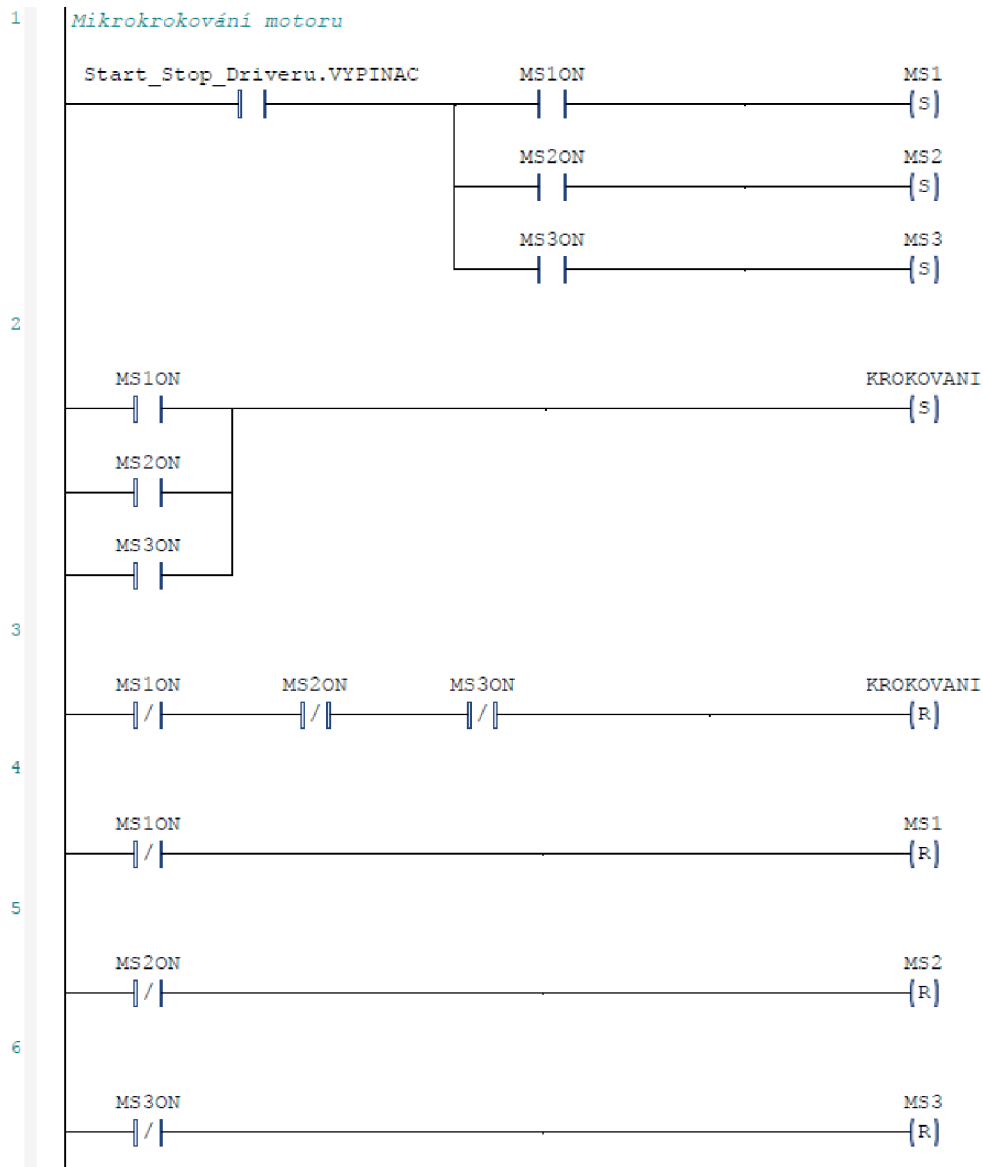
Obrázek 6.8 Čtení aktuální pozice a vynulování pozice

6.4.4 Program pro mikrokrokování motoru

Integrovaný obvod je vybaven třemi logickými výstupy pro řízení krokování motoru. Proto byly do programu přidány proměnné MS1, MS2 a MS3, které jsou namapovány na výstup PLC a propojeny s logickými výstupy obvodu. Pro jejich ovládání slouží v programu tlačítka MS1ON, MS2ON a MS3ON, které jsou propojeny s tlačítky na DPS pod označením KROK. Ve vizualizaci aktivní stav jednotlivých výstupů pro krokování motoru indikují žluté LED diody. Kombinace těchto režimů krokování je v tabulce 6.1.

Tabulka 6.1 Tabulka krokových režimů motoru

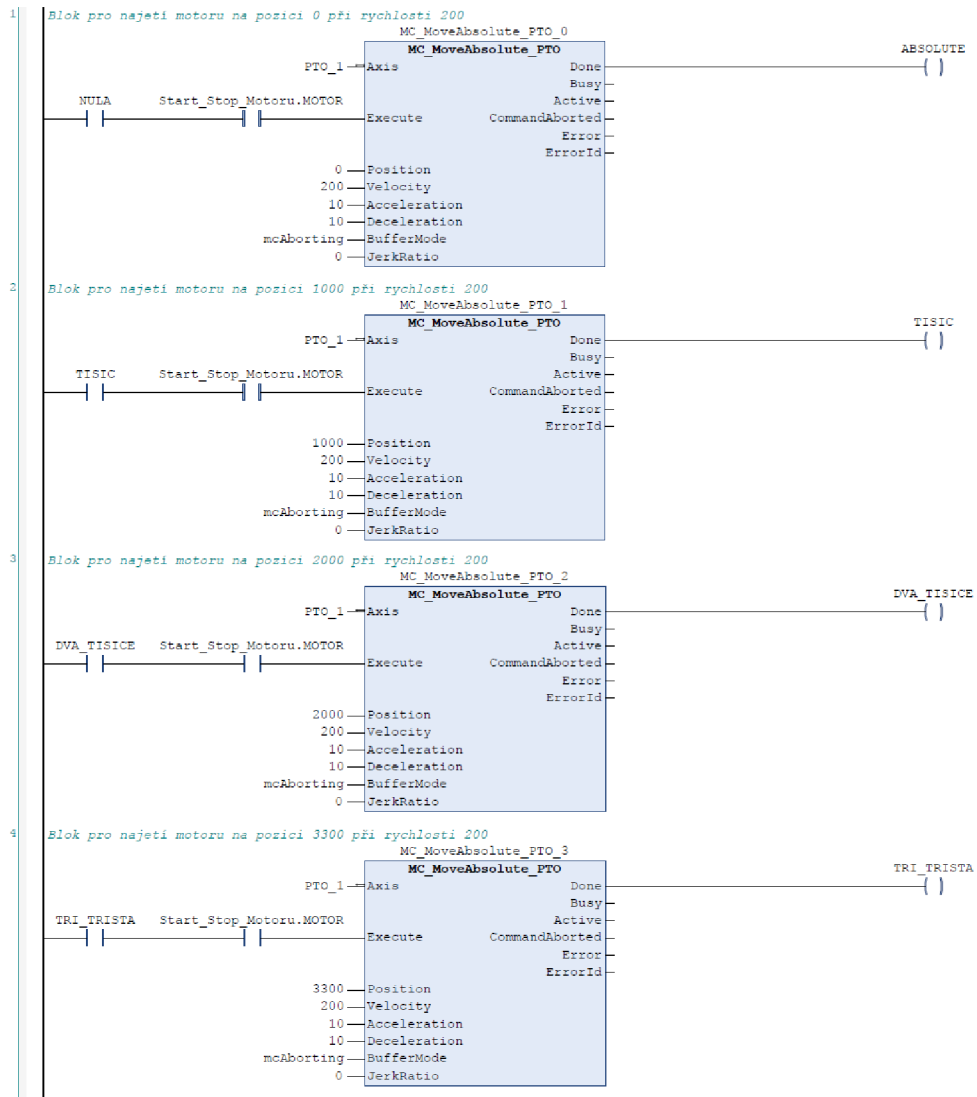
MS1	MS2	MS3	Režim kroku
0	0	0	Plný krok
1	0	0	1/2 krok
0	1	0	1/4 krok
1	1	0	1/8 krok
1	1	1	1/16 krok



Obrázek 6.9 Program pro krokování motoru

6.4.5 Program pro najetí motoru do dané pozice

Pro přesun tiskové hlavy pomocí motoru na přesně danou pozici existuje v programovacím prostředí blok *MC_MoveAbsolute_PTO*. Stačí zadat, s jakou rychlostí se má motor otáčet a na jakou pozici má tisková hlava najet. Do této práce se přidaly čtyři tyto bloky, které ovládají čtyři tlačítka na DPS. Tato tlačítka jsou připojena na vstupy PLC a při jejich stisknutí tisková hlava najede na předem nastavenou pozici při rychlosti 200. Pro najetí na pozici 0 slouží tlačítko *V1*, na pozici 1000 tlačítko *V2*, na pozici 2000 tlačítko *U1* a na konečnou pozici 3300 tlačítko *U2*.

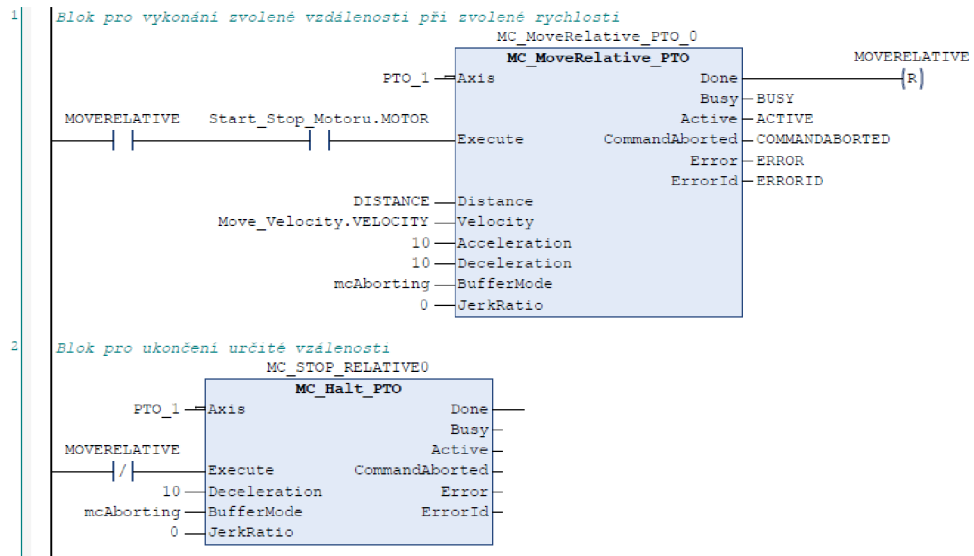


Obrázek 6.10 Program pro najetí motoru do dané pozice

6.4.6 Program pro vykonání určité vzdálenosti

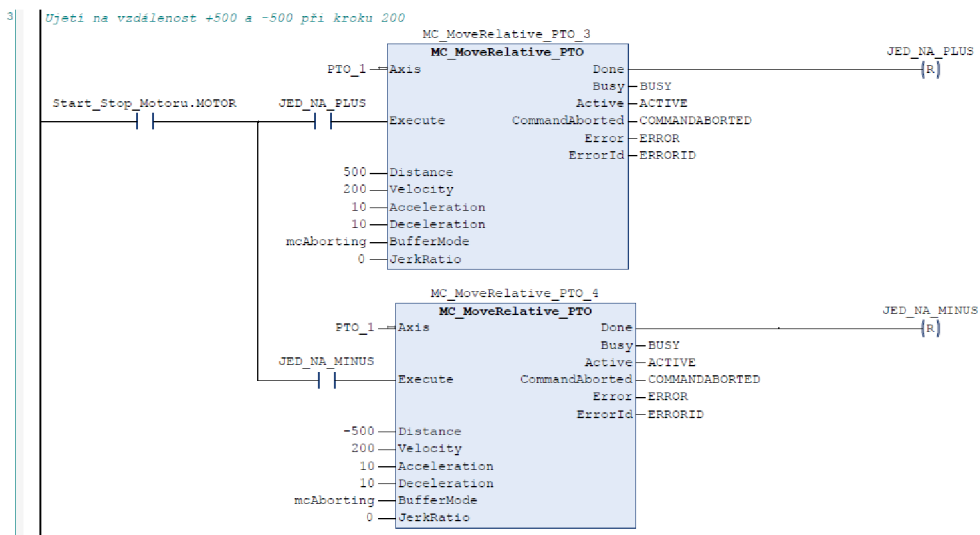
Pro pohyb tiskové hlavy se může zadat, aby vykonala určitou vzdálenost směrem doprava nebo doleva. Pro takový pokyn slouží blok *MC_MoveRelative_PTO*. V tomto bloku opět stačí zadat rychlost otáčení motoru a o jakou vzdálenost se má tisková hlava posunout.

V této práci se vytvořily celkem 4 funkce pro ukázkou funkčnosti tohoto bloku. První funkce slouží pro najetí tiskové hlavy vpřed o vzdálenost a s rychlostí, kterou si uživatel sám zvolí. Tato funkce se provede přes vizualizaci, kdy si uživatel nejdříve zvolí rychlost motoru a vzdálenost, kterou tisková hlava má vykonat. Poté se funkce provede stisknutím tlačítka *S7*, které představuje v programu proměnnou *MOVERELATIVE*.



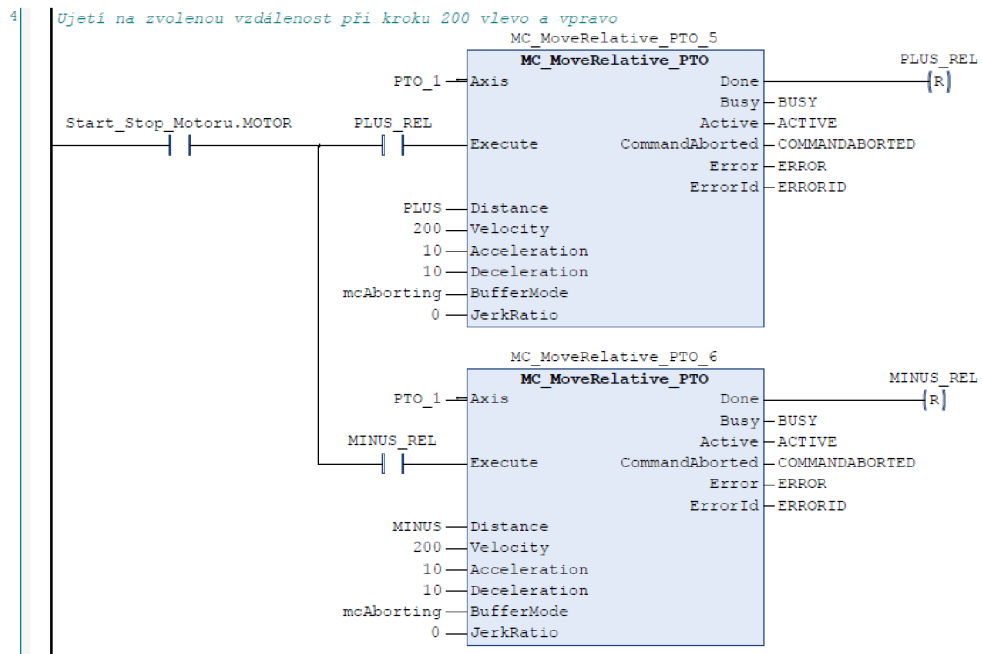
Obrázek 6.11 Vykonání určité vzdálenosti vpřed při zvolené rychlosti

Při stisku tlačítka *S1* na DPS, které v programu představuje proměnnou *JED_NA_MINUS*, se tisková hlava přesune zpět o 500 kroků. Naopak při stisku tlačítka *S2* na DPS, v programu proměnná *JED_NA_PLUS*, se tisková hlava přesune o 500 kroků vpřed. Motor se bude otáčet rychlostí 200 v případě obou těchto funkcí, jak ukazuje obrázek 6.12.



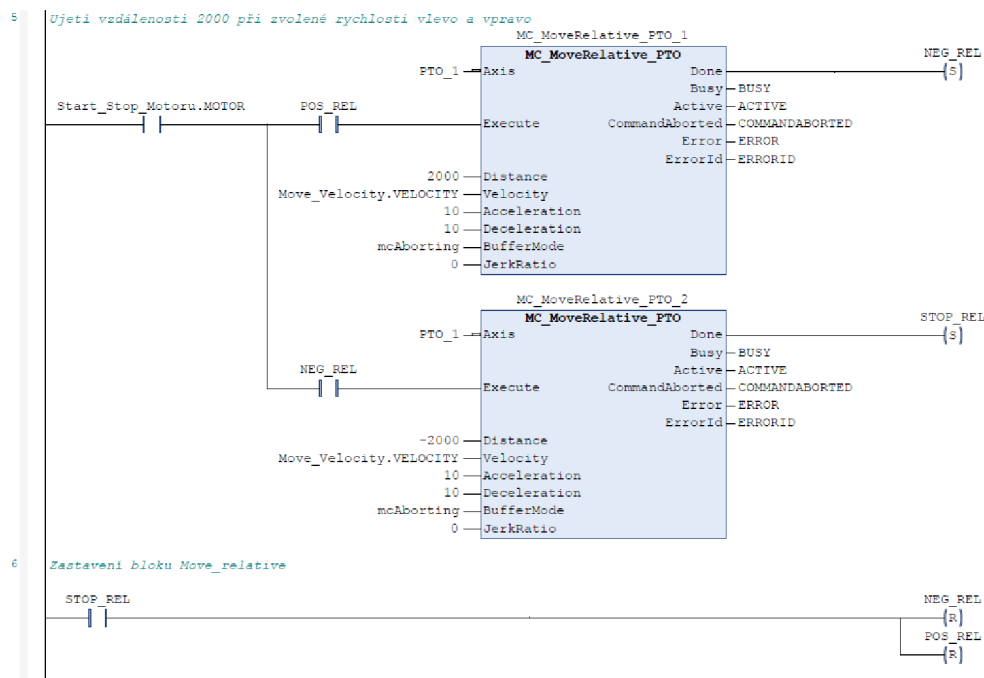
Obrázek 6.12 Vykonání určité vzdálenosti zpět a vpřed při dané rychlosti

Následující funkce je téměř totožná jako předchozí, pouze s tím rozdílem, že vzdálenost se zde volí přes vizualizaci. Při nastavené rychlosti motoru 200 si uživatel sám zvolí, o jakou vzdálenost a jakým směrem se má tisková hlava posunout. Pro posun vpravo slouží tlačítko *S5*, představující proměnnou *PLUS_REL* a pro posun vlevo tlačítko *S6*, představující proměnnou *MINUS_REL*, jak ukazuje obrázek 6.13.



Obrázek 6.13 Vykonání zvolené vzdálenosti zpět a vpřed při dané rychlosti

Poslední funkce slouží pro posun tiskové hlavy o předem danou vzdálenost 2000. Uživatel si ve vizualizaci zvolí rychlost, jakou se má tisková hlava pohybovat. Při stisku tlačítka *S9*, představující proměnnou *POS_REL*, urazí tisková hlava vzdálenost 2000. Po jejím dosažení se tisková hlava vrátí zpět a zastaví se.

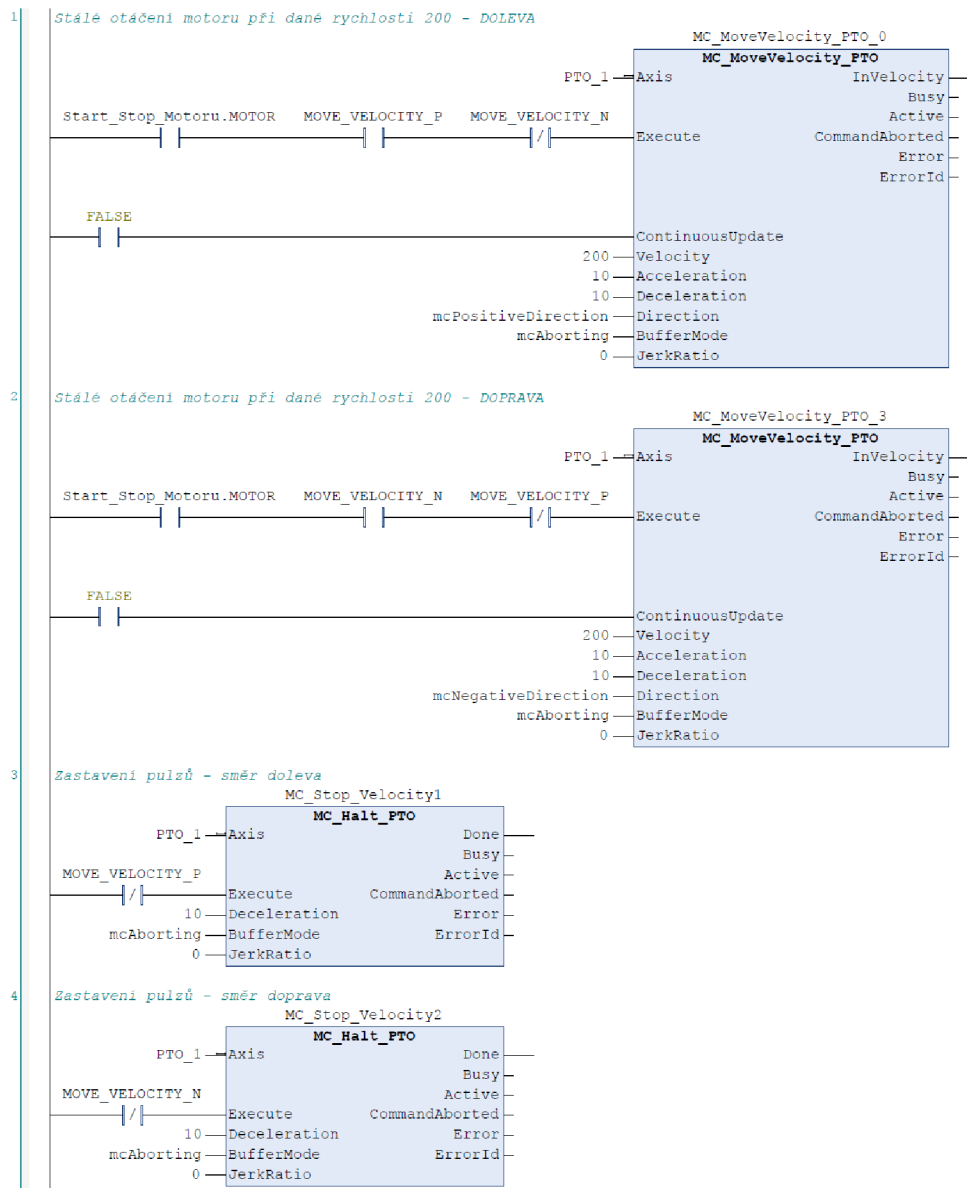


Obrázek 6.14 Vykonání dané vzdálenosti zpět a vpřed při dané rychlosti

6.4.7 Program pro stálé otáčení motoru

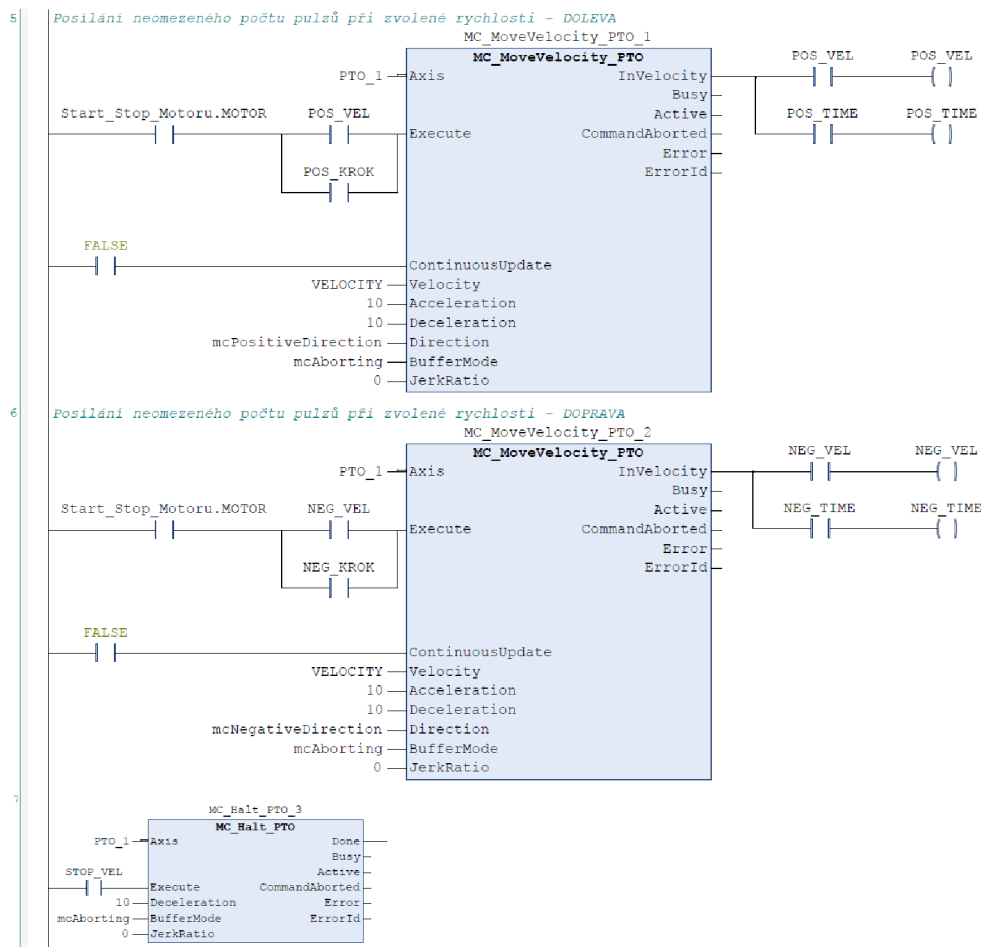
Jestliže je požadavek, aby se motor stále otáčel bez zadané vzdálenosti, tak je k dispozici blok *MC_MoveVelocity_PTO*. Při spuštění tohoto bloku dochází k posílání pulzů z PLC do motoru a tím ke stálému pohybu tiskové hlavy. Do tohoto bloku je nutné zadat rychlost otáčení motoru a směr, kterým se má motor otáčet.

Jako příklady použití tohoto bloku jsou v této práci vypracovány 3 funkce. První funkce, kterou lze vidět na obrázku 6.15, slouží pro stálé otáčení motoru konstantní rychlostí. První blok slouží pro pohyb motoru směrem doleva a druhý blok pro pohyb motoru směrem doprava. Proměnná *MOVE_VELOCITY_P* a *MOVE_VELOCITY_N* jsou spouštěny pomocí tlačítek *Vlevo* a *Vpravo* na DPS. To znamená, že při stisku a držení tlačítka *Vlevo* se tisková hlava bude posouvat doleva do té doby, než uživatel pustí tlačítko. Totéž platí pro tlačítko *Vpravo*, ale tisková hlava se bude pohybovat opačným směrem. Po puštění tlačítek se aktivují bloky *MC_Halt_PTO*, které dají povel k zastavení motoru.



Obrázek 6.15 Stálé otáčení motoru vlevo a vpravo

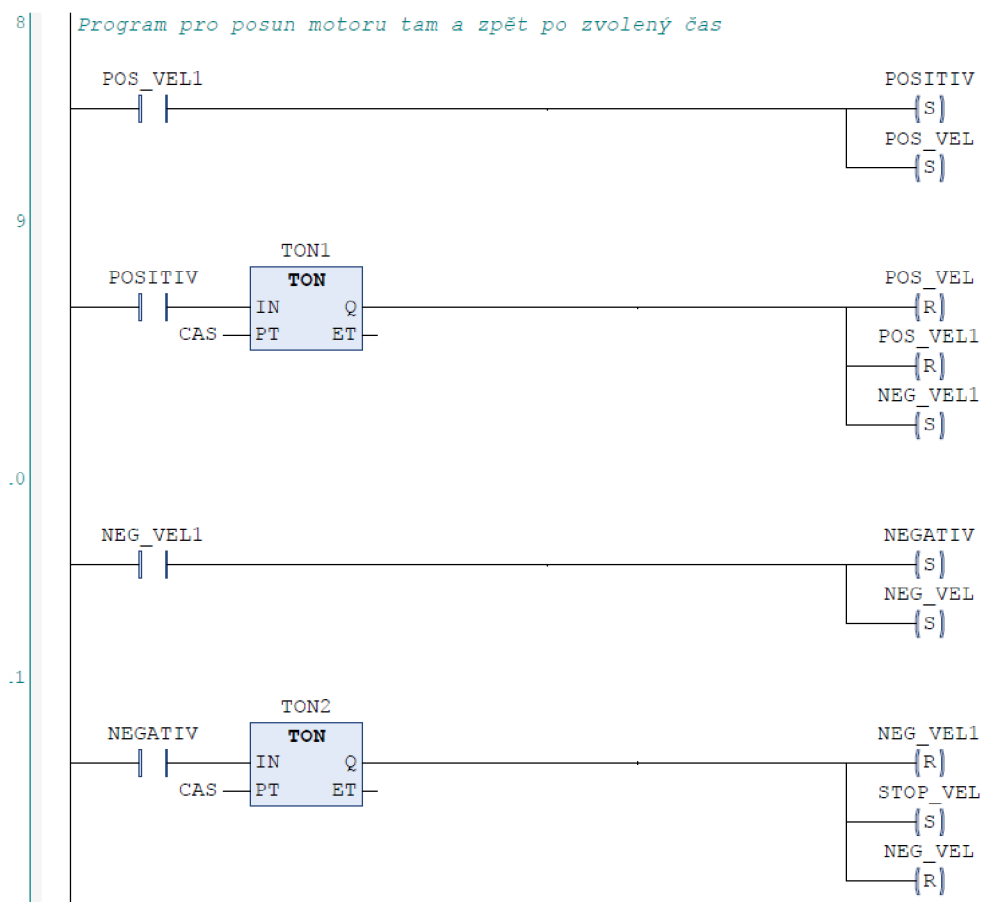
Pro zbylé 2 funkce jsou společné bloky pro posílání pulzů *MC_MoveVelocity_PTO* a blok pro zastavení pulzů *MC_Halt_PTO*, které lze vidět na obrázku 6.16. Při těchto funkcích byly použity časovače, kterým se zadá doba, po kterou má motor vykonávat svůj pohyb. Tyto bloky se spouští tlačítky, které se ovládají přes vizualizaci.



Obrázek 6.16 Bloky pro stálé otáčení motoru ve vizualizaci

První funkce, zobrazena na obrázku 6.17, slouží pro posun tiskové hlavy při dané rychlosti po dobu, která je zvolena ve vizualizaci. Uživatel ve vizualiaci zvolí čas posunu a funkce se spustí stisknutím tlačítka S8, které reprezentuje proměnnou *POS_VEL1*. Při stisku tohoto tlačítka se aktivuje blok pro posun tiskové hlavy směrem doprava a zároveň začne běžet zvolený čas posunu hlavy. Po dosažení tohoto času dojde k vypnutí bloku pro pohyb tiskové hlavy doprava.

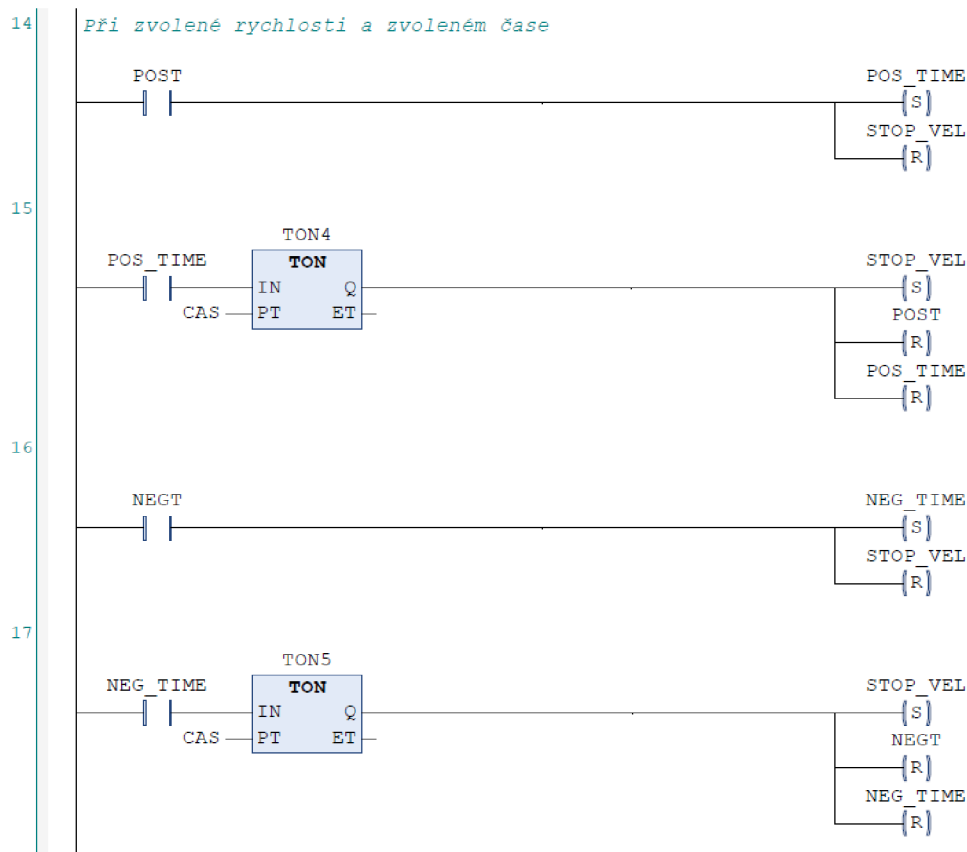
V tentýž moment se aktivuje blok pro pohyb směrem doleva a opět běží vybraný čas posunu. Při docílení nastaveného času dojde k deaktivaci bloku pro pohyb tiskové hlavy směrem vlevo a aktivuje se proměnná *STOP_VEL*, která aktivuje blok pro zastavení vysílání pulzů do motoru, který byl viděn na obrázku 6.16.



Obrázek 6.17 Funkce při dané rychlosti a zvoleném čase

Druhá funkce slouží pro pohyb tiskové hlavy směrem vlevo nebo vpravo, kdy se nejdříve zvolí rychlost, jakou se má tisková hlava pohybovat a čas, jak dlouho se bude tisková hlava pohybovat. Poté se stiskem tlačítek ve vizualizaci vybere směr pohybu.

Proměnnou POST, která je ve vizualizaci znázorněna jako tlačítko S3, se aktivuje blok pro pohyb tiskové hlavy směrem vpravo a zároveň začne běžet doba pohybu. Naopak, při stisku tlačítka S4, představující proměnnou NEG, se spustí blok pro pohyb tiskové hlavy směrem doprava po zvolený čas. Po dosažení zvolené doby pohybu se aktivací proměnné STOP_VEL sepne blok pro zastavení pulzů a dojde k zastavení tiskové hlavy. Tato funkce je na obrázku 6.18.



Obrázek 6.18 Funkce při zvoleném čase i rychlosti pohybu

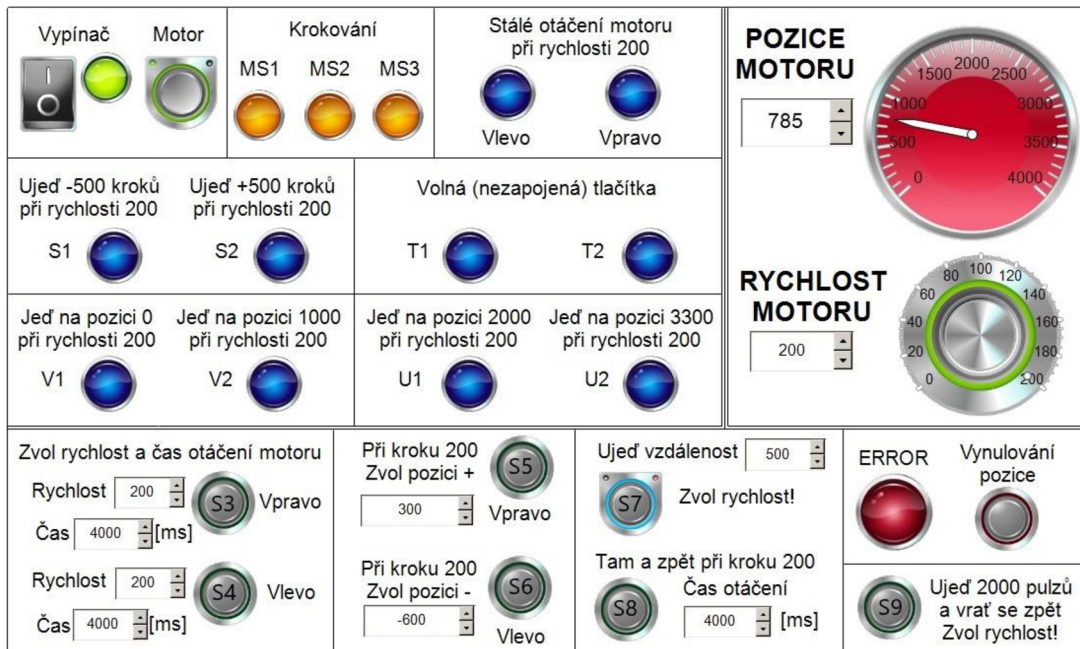
6.5 Vizualizace

Součástí návrhu této laboratorní úlohy bylo vytvořit jednoduché vizualizační rozhraní. Tato vizualizace částečně slouží k řízení navržené laboratorní úlohy, ale také jako ukázka, s jakými funkcemi a nástroji se dá taková vizualizace realizovat. Touto vizualizací se současně ověřuje funkčnost navrženého programu.

Pro DPS s tlačítka jsou zde použity signalizační LED diody, které indikují aktivitu stisknutého tlačítka. Při zapnutí tlačítka *Vypínač* a *Motor* se rozsvítí dioda zelené barvy. Pokud je aktivní režim krokování motoru, rozsvítí se žlutá dioda pro daný logický výstup pro krokování motoru. Pro tlačítka usazené na DPS, která vykonávají daný úkol pro pohyb tiskové hlavy, slouží modré diody.

Pro zobrazení aktuální pozice motoru je zde použit měřič pozice. Pro nastavení rychlosti pohybu motoru ve vizualizaci slouží virtuální potenciometr. Tato rychlost se dá případně nastavit v okénku s šipkami vedle potenciometru. Rychlost je udávána v Hz, tedy při nastavené rychlosti 200 Hz vykoná motor 200 kroků na otáčku v režimu plného kroku motoru. Pro zbytek funkce programu, které jsou sepsány v předchozích kapitolách, zde slouží virtuální tlačítka pro spuštění těchto funkcí.

Volba pozice, rychlosti motoru a času se stejně jako pozice motoru zobrazuje v okénkách u tlačítek. Pokud nastane chyba v programu, rozsvítí se červená LED dioda *ERROR*.



Obrázek 6.19 Vizualizace v programovacím prostředí SoMachine

6.6 Naměřené průběhy krokového motoru

Pro tuto úlohu byly naměřeny průběhy proudu a napětí krokového motoru, kdy motor poháněl tiskovou hlavu. Na cívky krokového motoru byly posílány periodické pulzní signály. Poté se pomocí osciloskopu, Hallovy a napěťové sondy změřily průběhy proudu a napětí na jedné fázi krokového motoru. Tyto průběhy byly změřeny, kdy se krokový motor otáčel v režimu plného a polovičního kroku.

Porovnání režimu rozpadu pro plný a poloviční krok lze vidět na obrázku 6.20. Pro aktivní krokování motoru se spínají logické vstupy driveru MS1, MS2 a MS3 dle tabulky 6.1. Tyto vstupy mají 100 k Ω nebo 50 k Ω stahovací odpory. Během krokování si driver automaticky vybere režim rozpadu, a to smíšený nebo pomalý rozpad. Pro plný krok driver vybere pomalý režim rozpadu, pro poloviční oba režimy rozpadu. Při smíšeném režimu rozpadu dochází ke zvýšení přesnosti kroků, snížení ztrátového výkonu a ke snížení hluku motoru [23].

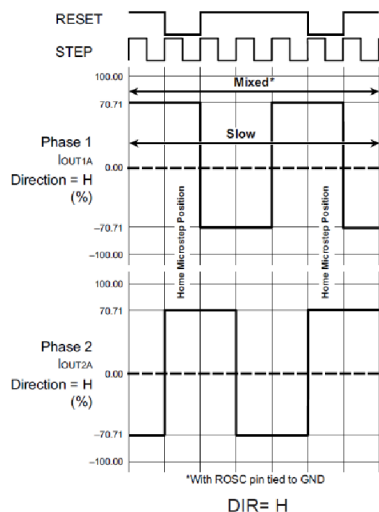


Figure 9: Decay Mode for Full-Step Increments

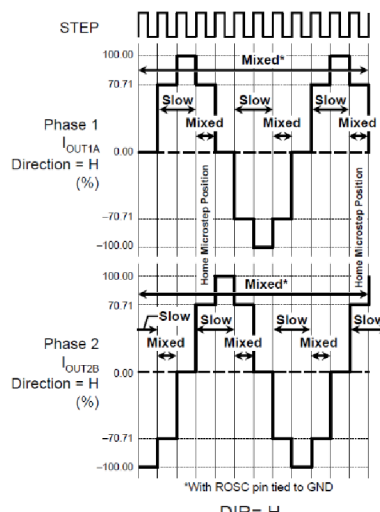
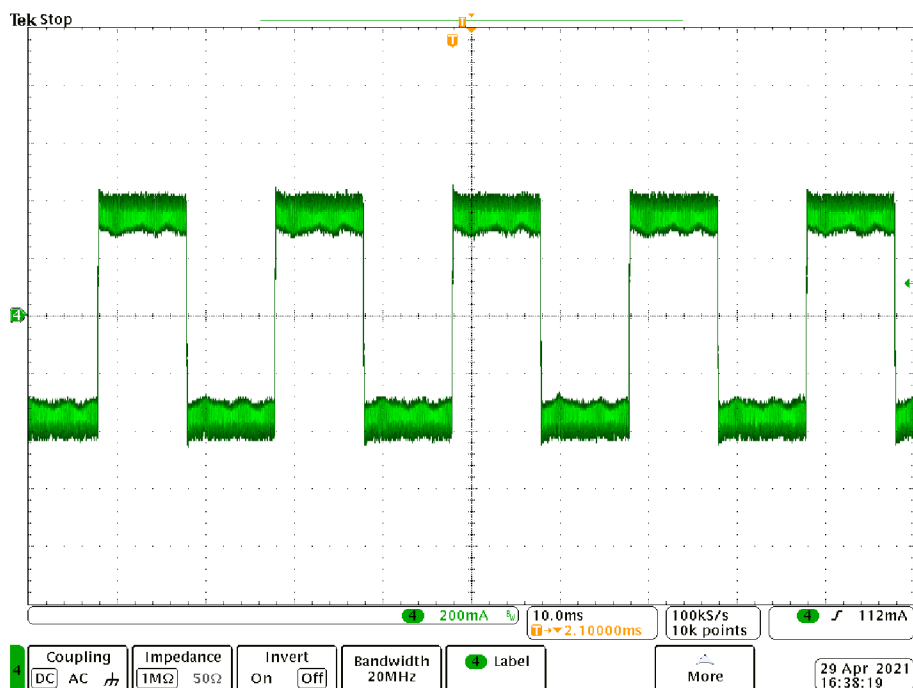


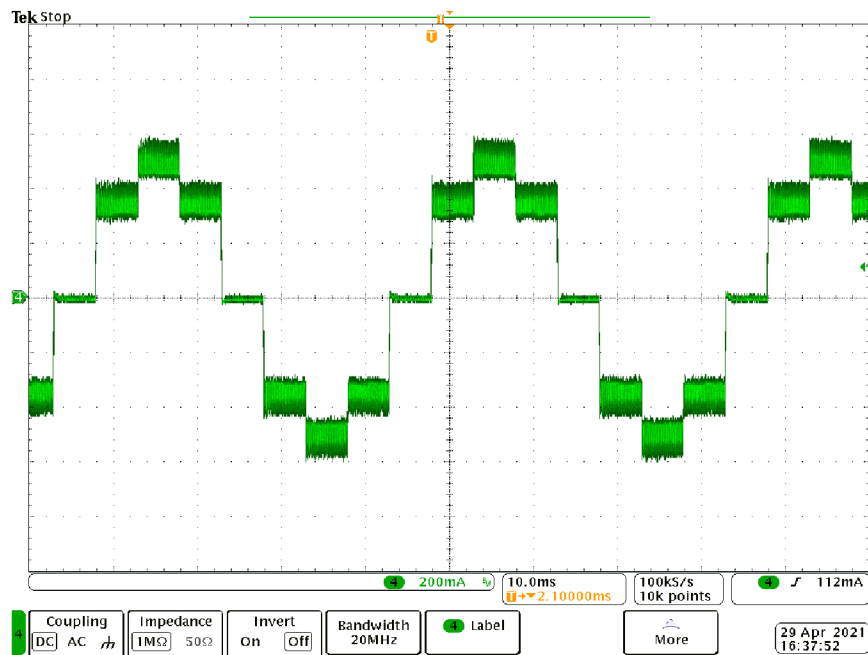
Figure 10: Decay Modes for Half-Step Increments

Obrázek 6.20 Režim rozpadu plného a polovičního kroku [23]

V režimu plného kroku má průběh proudu jedné fáze motoru tvar obdélníkového signálu. Průběh proudu, kdy byl motor v režimu polovičního kroku, se podobá sinusovému průběhu. Tyto dva průběhy a jejich porovnání lze vidět na obrázku 6.21 a obrázku 6.22. Použitím menšího kroku získáme větší přesnost polohování motoru. Motor rotuje při nižších otáčkách plynuleji a může dosáhnout většího počtu otáček než v režimu plného kroku.



Obrázek 6.21 Průběh proudu motoru pro plný krok



Obrázek 6.22 Průběh proudu motoru pro poloviční krok

Na obrázku 6.23 a obrázku 6.24 lze vidět detail průběhu proudu fáze motoru v režimu plném a polovičním kroku, který je pro oba režimy totožný. Pro režim plného kroku byla pomocí kurzorů změřena časová konstanta proudu, která byla 35 μs . Pomocí této časové konstanty lze spočítat frekvenci pulzního signálu.

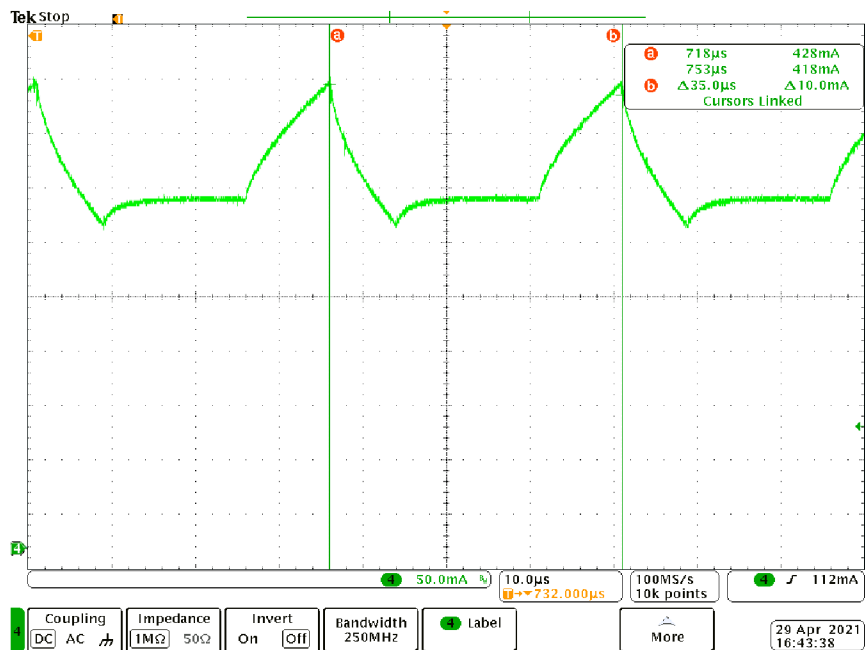
Frekvence je definována vztahem

$$f = \frac{1}{T}, \quad (6.1)$$

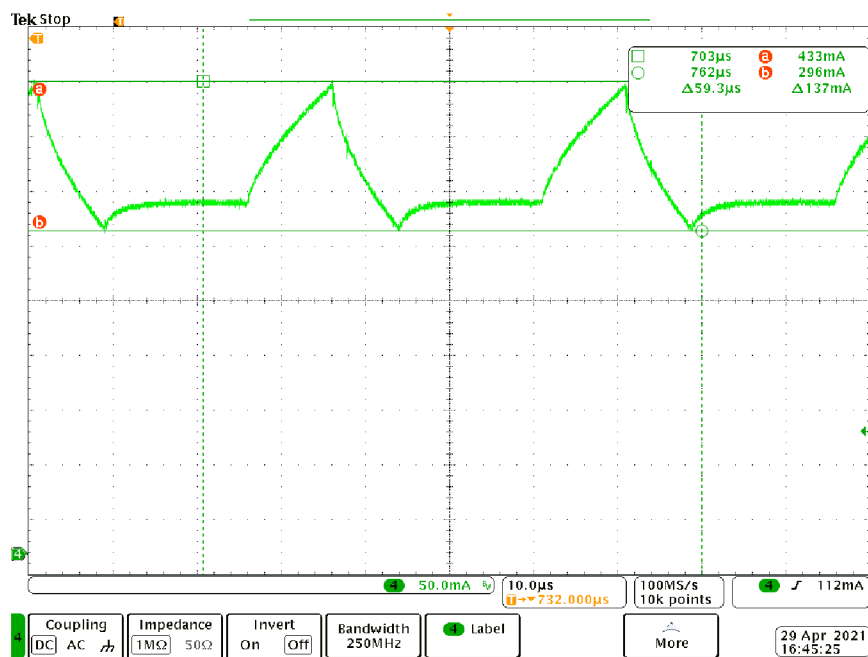
kde f je frekvence pulzního signálu a T je časová konstanta.

V režimu plného kroku je tedy frekvence pulzního signálu

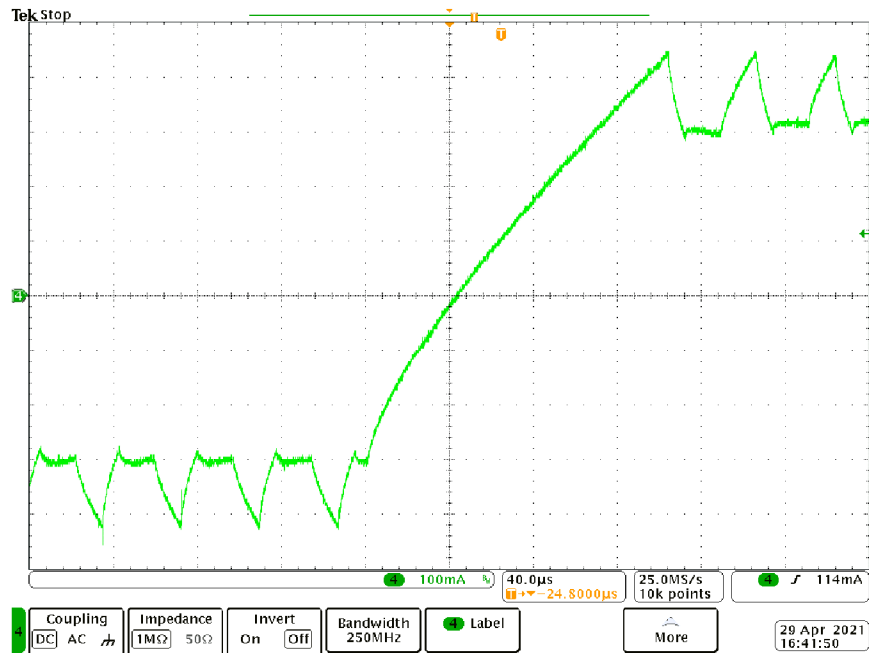
$$f = \frac{1}{T} = \frac{1}{35 \cdot 10^{-6}} = 28,6 \text{ kHz} \quad (6.2)$$



Obrázek 6.23 Detail průběhu proudu motoru pro plný krok

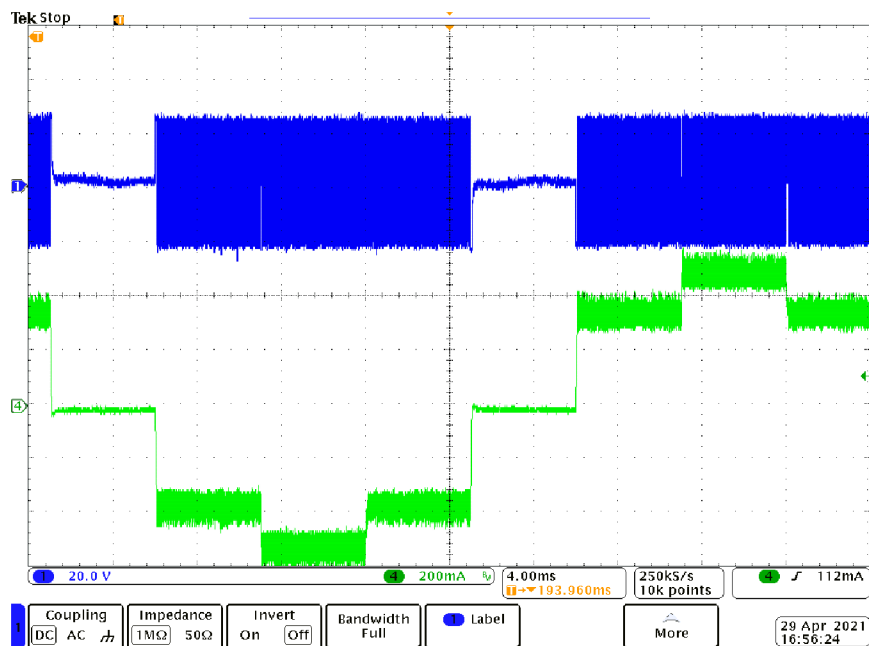


Obrázek 6.24 Detail průběhu proudu motoru pro poloviční krok

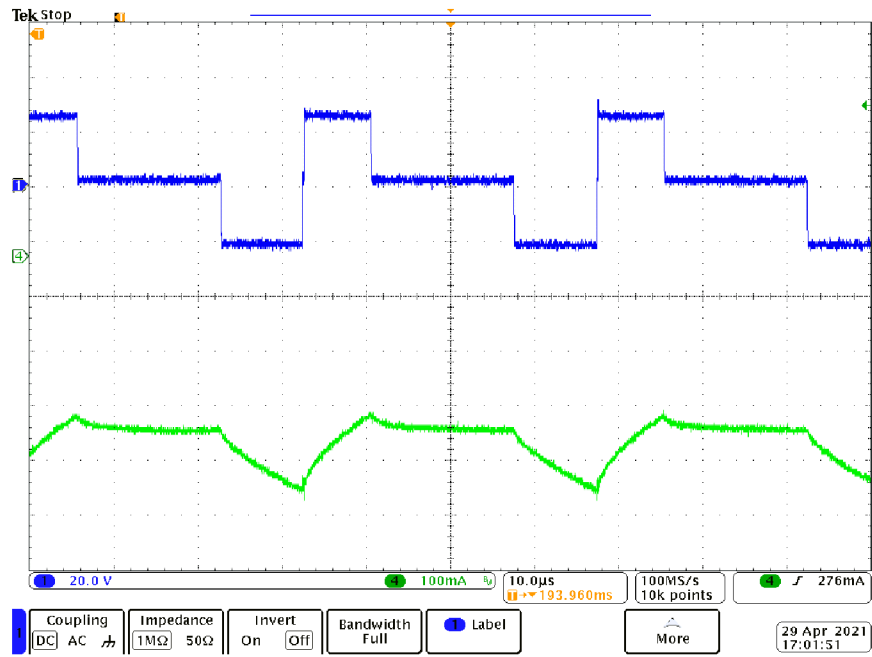


Obrázek 6.25 Přechod ze záporné polarity proudu do kladné pro plný krok

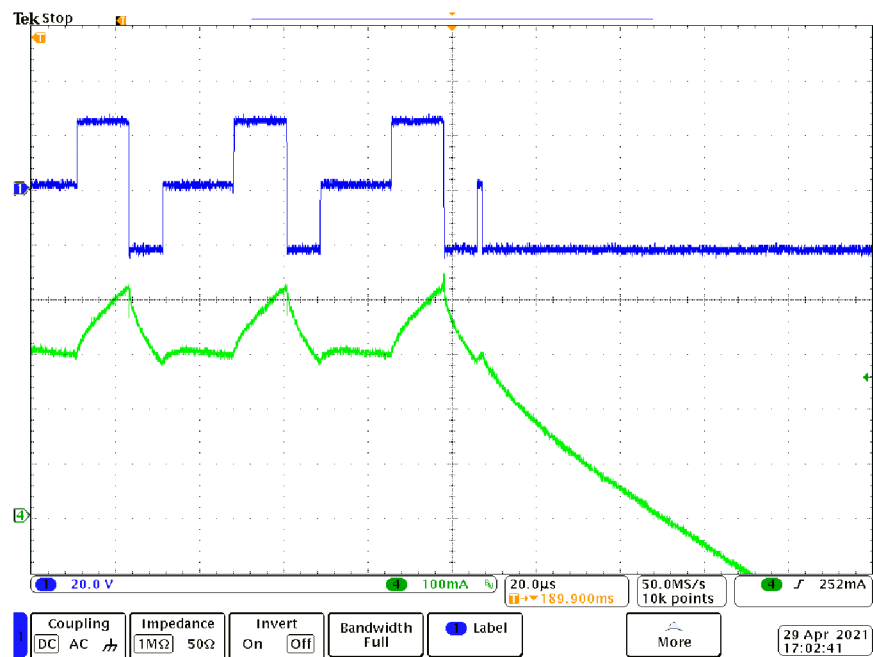
Následně se připojila k osciloskopu i napěťová sonda, která zaznamenávala průběh napětí na fázi motoru při polovičním kroku. Napětí je zaznamenáno na kanálu 1 a proud na kanálu 4. Z obrázku 6.26 lze vidět, že pulzy mají napětí kolem 24 V, což je jmenovité napětí zdroje.



Obrázek 6.26 Průběh napětí proudu motoru pro poloviční krok



Obrázek 6.27 Detail průběhu proudu a napětí motoru pro poloviční krok



Obrázek 6.28 Detail průběhu proudu a napětí při zastavení motoru

7. ZÁVĚR

Bakalářská práce se zabývala využitím programovatelnými logickými automaty ve výuce. Hlavním cílem bylo seznámení se s programovatelnými logickými automaty Schneider, návržení laboratorní úlohy s krokovým motorem a její následnou realizací. V první části byl úvod do problematiky a teorie pro programovatelné logické automaty. Dále byly popsány typy PLC, jejich přednosti a využití. V další části byly popsány programovací jazyky s jejich grafickým popisem. Následoval stručný popis produktů PLC od firmy Schneider Electric, jejich porovnání a výběr nejvhodnějšího typu PLC pro tuto práci. Práce obsahovala podrobný popis programovacího prostředí SoMachine s ukázkou tohoto prostředí i vývojového prostředí pro vizualizaci a stručným návodem pro programování.

V závěrečné části této bakalářské práce byla navržena a realizována laboratorní úloha s krokovým motorem, která bude použita pro výukové účely. Tato laboratorní úloha slouží pro lineární polohování tiskové hlavy z tiskárny, která je poháněna krokovým motorem HY 200. Pro řízení motoru byl použit integrovaný driver A4988 od výrobce Allegro, který je umístěn na jedné z desek plošných spojů. Druhá deska je složena z tlačítek pro řízení několika funkcí vytvořeného řídicího programu v programovacím prostředí SoMachine, které jsou zobrazeny v této práci. Zároveň byl sepsán laboratorní návod k tomuto řídicímu programu. Pro tuto úlohu byla také zrealizována přehledná a prostá vizualizace, která částečně slouží pro řízení krokového motoru. Při pohybu krokového motoru s plným a polovičním krokovým režimem byl zaznamenáván průběh proudu a napětí na jedné z fází motoru. Průběhy proudu fáze motoru jsou v porovnání s průběhy od výrobce integrovaného driveru A4988 téměř totožné.

V navazující diplomové práci lze na této laboratorní úloze vytvářet další funkce pro řízení krokového motoru a přidávat další komponenty pro vylepšení funkčnosti této bakalářské práce.

LITERATURA

- [1] ŠMEJKAL, Ladislav a Marie MARTINÁSKOVÁ. PLC a automatizace. Praha: BEN - technická literatura, 1999. ISBN 978-80-86056-58-6.
- [2] ŠVARC, Ivan. Automatizace: automatické řízení. Vyd. 2., dopl. Brno: Akademické nakladatelství CERM, 2005. ISBN 80-214-2943-7.
- [3] ZEŽULKA, František. Automatizační prostředky. Vyd. 2. Brno: Z. Novotný, 2000. Učební texty vysokých škol. ISBN 80-214-1739-0.
- [4] KŘENA, Michal. Řídicí systémy Modicon psaly historii, vynikají v současnosti a tvoří budoucnost [online]. 2015 [cit. 2019-11-23]. Dostupné z: [http://www.controlengcesko.com/index.php?id=47&no_cache=1&tx_ttnews\[tt_news\]=5697&cHash=d8c0e6b395&type=98](http://www.controlengcesko.com/index.php?id=47&no_cache=1&tx_ttnews[tt_news]=5697&cHash=d8c0e6b395&type=98).
- [5] What is a PLC System – Different Types of PLCs with Applications. ElProCus - Electronic Projects for Engineering Students [online]. 2013 [cit. 2019-11-15]. Dostupné z: <https://www.elprocus.com/programmable-logic-controllers-and-types-of-plcs/>.
- [6] SESTAVA PLC. PLC AUTOMATIZACE [online]. [cit. 2019-11-15]. Dostupné z: <http://plc-automatizace.cz/knihovna/plc/plc-hw-sestava.htm>.
- [7] LUKÁŠ, Jiří. PLC - programovací jazyky [online]. In: . 2005 [cit. 2019-11-17]. Dostupné z: <http://www.zandruvsvet.opatnet.cz/view.php?cisloclanku=2005051001>.
- [8] OLIVEIRA, Paulo Jorge. Industrial Automation: PLC Programming languages [online]. 2010 [cit. 2019-11-24]. Dostupné z: http://users.isr.ist.utl.pt/~jag/courses/api13/docs/API_I_C3_3_ST.pdf.
- [9] LAMB, Frank. Other PLC programming languages [online]. In: . 2019 [cit. 2019-11-24]. Dostupné z: <https://www.controleng.com/articles/other-plc-programming-languages/>.
- [10] Schneider Electric [online]. 2018 [cit. 2019-12-01]. Dostupné z: <https://www.se.com/cz/cs/about-us/company-profile/>.
- [11] Zelio Logic: Malý řídicí systém. Schneider Electric [online]. 2018 [cit. 2019-12-01]. Dostupné z: <https://www.se.com/cz/cs/product-range/531-zelio-logic/?parent-category-id=3900&parent-subcategory-id=3910&filter=business-1-prumyslova-automatizace-a-řízení>.
- [12] Modicon M238: Řídicí systém pro výrobce strojů. Schneider Electric [online]. 2018 [cit. 2019-12-01]. Dostupné z: <https://www.se.com/cz/cs/product-range-presentation/2335-modicon-m238/?parent-category-id=3900&parent-subcategory-id=3910&filter=business-1-prumyslova-automatizace-a-řízení>.
- [13] Modicon M221: Řešení pro komunikaci. Schneider Electric [online]. 2018 [cit. 2019-12-01]. Dostupné z: <https://www.se.com/cz/cs/product-range-presentation/62128-modicon-m221/?parent-category-id=3900&parent-subcategory-id=3910&filter=business-1-prumyslova-automatizace-a-řízení#tabstop>.

- [14] Modicon M251: Modulární řešení a distribuovaná architektura. Schneider Electric [online]. 2018 [cit. 2019-12-01]. Dostupné z: <https://www.se.com/cz/cs/product-range-presentation/62130-modicon-m251/?parent-category-id=3900&parent-subcategory-id=3910&filter=business-1-průmyslová-automatizace-a-řízení#tabs-top>.
- [15] Kontrolér pro logické řízení a servopohony Modicon M262. Schneider Electric [online]. 2018 [cit. 2019-12-01]. Dostupné z: <https://www.se.com/cz/cs/product-range-presentation/65771-kontrolér-pro-logické-řízení-a-servopohony-modicon-m262/?parent-category-id=3900&parent-subcategory-id=3910&filter=business-1-průmyslová-automatizace-a-řízení#tabs-top>.
- [16] Modicon M258: Řídicí systém pro výrobce strojů. Schneider Electric [online]. 2018 [cit. 2019-12-01]. Dostupné z: <https://www.se.com/cz/cs/product-range-presentation/2730-modicon-m258/?parent-category-id=3900&parent-subcategory-id=3910&filter=business-1-průmyslová-automatizace-a-řízení#tabs-top>.
- [17] Modicon M241 Logic controllers: Catalog. Schneider Electric [online]. 2019 [cit. 2019-11-25]. Dostupné z: https://download.schneider-electric.com/files?p_enDocType=Catalog&p_File_Name=DIA3ED2140107EN.pdf&p_Doc_Ref=DIA3ED2140107EN.
- [18] TM241CEC24T. Schneider Electric [online]. 2019 [cit. 2019-11-25]. Dostupné z: <https://product-help.schneider-electric.com/Machine%20Expert/V1.1/en/m241hw/m241hw/TM241CEC24T/TM241CEC24T.htm>.
- [19] TM241CEC24T: Product data sheet [online]. Schneider Electric, 2019 [cit. 2019-11-16]. Dostupné z: <https://www.se.com/ww/en/product/download-pdf/TM241CEC24T>.
- [20] HOWLETT, Bruce. Getting Started With SoMachine: Self Study Manual [online]. Schneider Electric, 2014 [cit. 2019-11-25]. Dostupné z: https://www.se.com/ww/resources/sites/SCHNEIDER_ELECTRIC/content/live/FAQS/288000/FA288633/en_US/Training%20Manual%20EN.pdf.
- [21] Vijeo Designer: Training Manual [online]. Schneider Electric, 2012 [cit. 2019-12-08]. Dostupné z: http://kalatech-bg.com/upload/VijeoDesigner61/VJD61_Training_v12.pdf
- [22] MAE Stepper Motors. Elektronik Lavpris [online]. 2003 [cit. 2021-5-15]. Dostupné z: https://elektronik-lavpris.dk/files/user_uploaded/142498_1524729373.pdf
- [23] A4988: DMOS Microstepping Driver with Translator And Overcurrent Protection. POLOLU [online]. 2009 [cit. 2021-4-24]. Dostupné z: https://www.pololu.com/file/0J450/a4988_DMOS_microstepping_driver_with_translator.pdf

SEZNAM SYMBOLŮ A ZKRATEK

Symboly:

U	napětí	(V)
f	frekvence	(Hz)
I	proud	(A)
R	odpor	(Ω)

Zkratky:

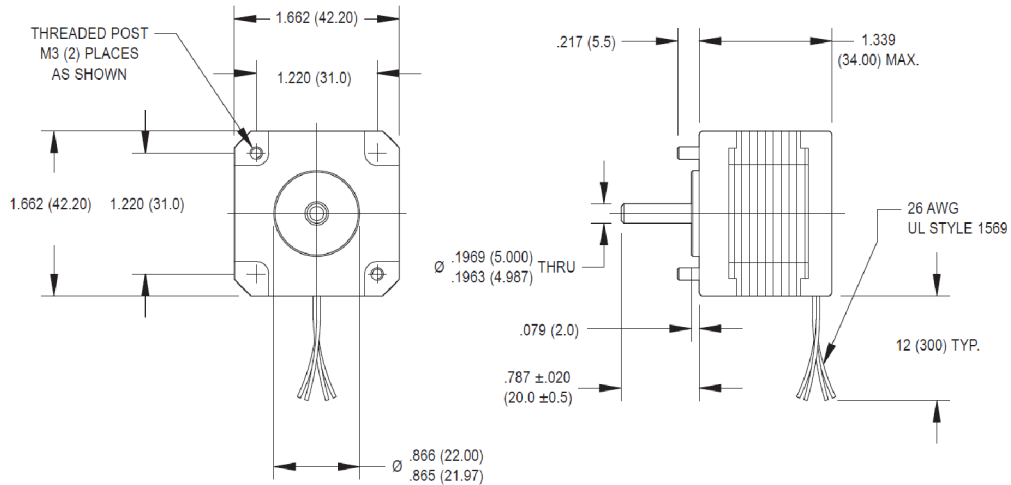
CAN	Datová sběrnice místní sítě (Controller Area Network)
CFC	Nepřetržitý funkční diagram (Continuous Function Chart)
CPU	Centrální procesorová jednotka (Central Processing Unit)
DC	Stejnoseměrné napětí (Direct Current)
DPS	Deska plošných spojů (Printed Circuit Board)
FBD	Diagram funkčních bloků (Function Block Diagram)
HMI	Rozhraní mezi člověkem a strojem (Human-Machine Interface)
IIoT	Průmyslový internet věcí (Industrial Internet of Things)
IL	Instrukční list (Instruction List)
LD	Příčkový diagram (Ladder Diagram)
LED	Světlo emitující dioda (Light Emitting Diode)
OS	Operační systém (Operation System)
PC	Osobní počítač (Personal Computer)
PID	Proporcionální, Integrovaný, Derivační (Proportional, Integrative, Derivative)
PLC	Programovatelný logický kontrolér (Programmable Logic Controller)
POU	Organizační jednotka programu (Program Organisation Unit)
PTO	Výstup pulzního vlaku (Pulse Train Output)
PWM	Pulzně šířková modulace (Pulse Width Modulation)
RAM	Paměť s náhodným přístupem (Random Access Memory)
SFC	Sekvenční funkční diagram (Sequential Function Chart)
ST	Strukturovaný text (Structured Text)
USB	Universální sériová sběrnice (Universal Serial Bus)
μ P	Mikroprocesor (Microprocessor)

SEZNAM PŘÍLOH

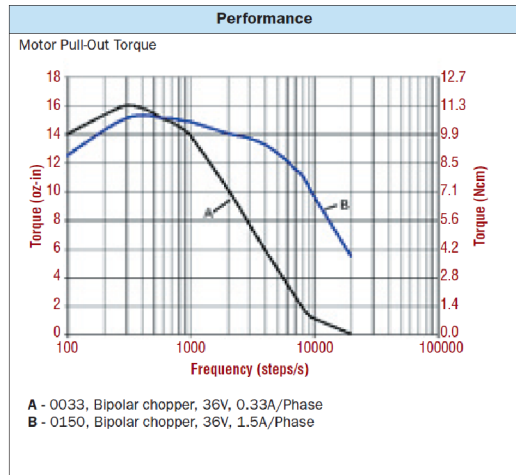
PŘÍLOHA A - DATASHEET MOTORU HY 200 1713 [22]	59
PŘÍLOHA B - DATASHEET DRIVERU A4988 [23]	60
PŘÍLOHA C - LABORATORNÍ NÁVOD	61

Příloha A - Datasheet motoru HY 200 1713 [22]

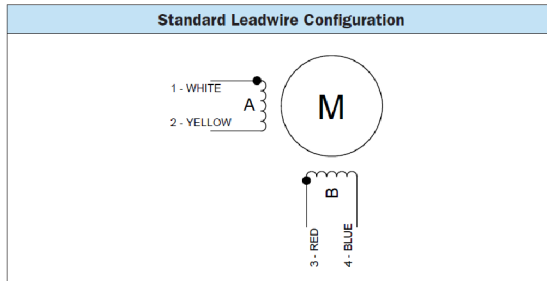
SIZE 17 STEPPER MOTOR DATA



Specification	Units	HY 200 1713	
		0033	0150
Rated Phase Current	A	0.33	1.50
Phase Resistance	Ω	23.9	1.0
Phase Inductance	mH	28.9	1.2
Holding Torque Unipolar	oz-in Ncm	—	—
Holding Torque Bipolar	oz-in Ncm	19.4 13.7	18.4 13.0
Detent Torque	oz-in Ncm	2.4 1.7	2.4 1.7
Rotor Inertia	oz-in-s ² $\times 10^{-4}$ g-cm ²	2.5 18	2.5 18
Motor Weight (Mass)	lb kg	0.4 0.2	0.4 0.2
Maximum Voltage	V	40	40
Std. No. of Leads	—	4	4



Available through the MotionExpress program.



- Standard Features**
- Step angle: 1.8°
 - Step angle accuracy: 5%
 - Insulation class: B (130°C)
 - NEMA 17 mounting configuration
 - Neodymium magnets
 - Additional windings and customization options available

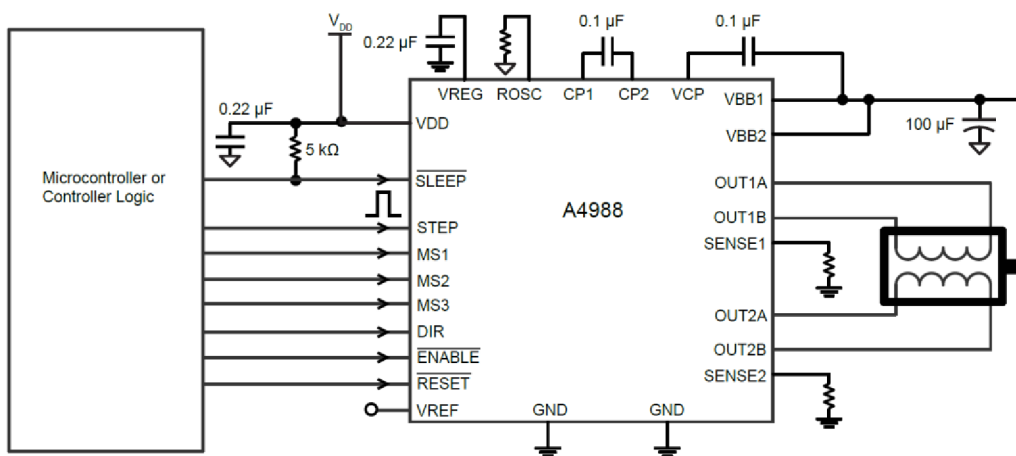
- Complementary Products (See Bulletin CO)**
- Gearboxes
 - Encoders

Příloha B - Datasheet driveru A4988 [23]

ELECTRICAL CHARACTERISTICS¹ at $T_A = 25^\circ\text{C}$, $V_{BB} = 35\text{ V}$ (unless otherwise noted)

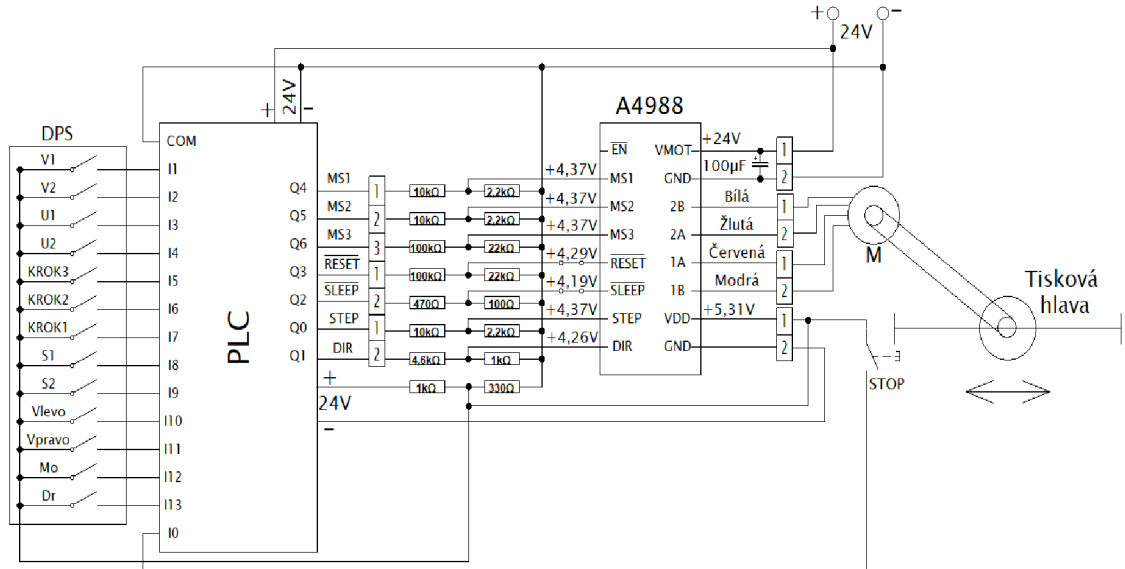
Characteristics	Symbol	Test Conditions	Min.	Typ. ²	Max.	Units
Output Drivers						
Load Supply Voltage Range	V_{BB}	Operating	8	–	35	V
Logic Supply Voltage Range	V_{DD}	Operating	3.0	–	5.5	V
Output On Resistance	R_{DSON}	Source Driver, $I_{OUT} = -1.5\text{ A}$	–	320	430	m Ω
		Sink Driver, $I_{OUT} = 1.5\text{ A}$	–	320	430	m Ω
Body Diode Forward Voltage	V_F	Source Diode, $I_F = -1.5\text{ A}$	–	–	1.2	V
		Sink Diode, $I_F = 1.5\text{ A}$	–	–	1.2	V
Motor Supply Current	I_{BB}	$f_{PWM} < 50\text{ kHz}$	–	–	4	mA
		Operating, outputs disabled	–	–	2	mA
Logic Supply Current	I_{DD}	$f_{PWM} < 50\text{ kHz}$	–	–	8	mA
		Outputs off	–	–	5	mA
Control Logic						
Logic Input Voltage	$V_{IN(1)}$		$V_{DD} \times 0.7$	–	–	V
	$V_{IN(0)}$		–	–	$V_{DD} \times 0.3$	V
Logic Input Current	$I_{IN(1)}$	$V_{IN} = V_{DD} \times 0.7$	–20	<1.0	20	μA
	$I_{IN(0)}$	$V_{IN} = V_{DD} \times 0.3$	–20	<1.0	20	μA
Microstep Select	R_{MS1}	MS1 pin	–	100	–	k Ω
	R_{MS2}	MS2 pin	–	50	–	k Ω
	R_{MS3}	MS3 pin	–	100	–	k Ω
Logic Input Hysteresis	$V_{HYS(IN)}$	As a % of V_{DD}	5	11	19	%
Blank Time	t_{BLANK}		0.7	1	1.3	μs
Fixed Off-Time	t_{OFF}	OSC = V_{DD} or GND	20	30	40	μs
		$R_{OSC} = 25\text{ k}\Omega$	23	30	37	μs
Reference Input Voltage Range	V_{REF}		0	–	4	V
Reference Input Current	I_{REF}		–3	0	3	μA
Current Trip-Level Error ³	err_i	$V_{REF} = 2\text{ V}$, % $I_{TRIPMAX} = 38.27\%$	–	–	± 15	%
		$V_{REF} = 2\text{ V}$, % $I_{TRIPMAX} = 70.71\%$	–	–	± 5	%
		$V_{REF} < 2\text{ V}$, % $I_{TRIPMAX} = 100.00\%$	–	–	± 5	%
Crossover Dead Time	t_{DT}		100	475	800	ns
Protection						
Overcurrent Protection Threshold ⁴	I_{OCPST}		2.1	–	–	A
Thermal Shutdown Temperature	T_{TSD}		–	165	–	$^\circ\text{C}$
Thermal Shutdown Hysteresis	T_{TSDHYS}		–	15	–	$^\circ\text{C}$
VDD Undervoltage Lockout	V_{DDUVLO}	V_{DD} rising	2.7	2.8	2.9	V
VDD Undervoltage Hysteresis	$V_{DDUVLOHYS}$		–	90	–	mV

Typical Application Diagram



Příloha C - Laboratorní návod

- 1) Zapojte laboratorní úlohu dle uvedeného schématu zapojení



- 2) Spusťte program SoMachine, připojte zvolený kontrolér a založte nový projekt s vhodným programovacím jazykem.
- 3) Povolte pulzní generátor PTO, ve kterém nastavte pro vstupní pin STEP výstup Q0 a pro pin DIR výstup Q1. Dále povolte Homing mode pro najetí motoru na nulovou pozici, dokud nesespe spínač STOP připojený na rychlý vstup IO PLC.
- 4) Přidejte objekty POU pro start driveru a motoru, krokování, pozici motoru a tři druhy pohybu motoru (MoveAbsolute, MoveRelative a MoveVelocity).
- 5) Vytvořte přehlednou a vizualizaci, do které budou vloženy tlačítka, měřič pozice, potenciometr pro rychlost motoru, LED diody pro indikaci spuštěných funkcí a okénka, do kterých se bude zapisovat vzdálenost, rychlost a čas otáčení motoru.
- 6) Pro start driveru vložte vstupní proměnnou VYPINAC, který namapujte na vstup I13 a tlačítko Dr na DPS připojte na tento vstup. Při stisku tlačítka se povolí piny SLEEP a RESET driveru, které jsou připojeny na výstupy Q2 a Q3 a zároveň se provede blok pro reset. Ve vizualizaci vložte LED diodu pro indikaci aktivní proměnné.
- 7) Pro start motoru vložte vstupní proměnnou MOTOR, HOME, MOTOR1ON a STOP_PTO1. Aktivní proměnná MOTOR, namapována na vstup I12 a připojená na tlačítko Mo na DPS, aktivuje blok pro povolení pohybu motoru Power_PTO. Ve vizualizaci vložte LED diodu nebo tlačítko s diodou pro indikaci aktivní proměnné. Dále se aktivuje proměnná HOME, kdy se spustí Homing mode a motor najede na pozici 0. V tu chvíli sepne spínač STOP a dojde k zastavení motoru.

- 8) Pro čtení aktuální pozice motoru vložte vstupní proměnnou `READ_POSITION`, která aktivuje blok `ReadActualPosition_PTO` pro čtení aktuální pozice. Pro tuto proměnnou vytvořte ve vizualizaci měřič, který bude zobrazovat aktuální pozici motoru. Dále vložte vstupní proměnnou `ZACATEK`, která sepne blok `SetPosition_PTO`. Do tohoto bloku vložte pozici 0 a povolením této proměnné se vynuluje pozice. Ve vizualizaci vložte tlačítko pro nulování pozice.
- 9) Pro krokování motoru vložte vstupní proměnné `MS1ON`, `MS2ON` a `MS3ON`, které budou namapovány na tlačítka `KROK` na `DPS`. Tyto proměnné budou sloužit pro povolení výstupních pinů driveru `MS1` (výstup `Q4`), `MS2` (výstup `Q5`) a `MS3` (výstup `Q6`), které jsou připojeny na rychlé výstupy `PLC`. Ve vizualizaci vložte `LED` diody pro indikaci povolených výstupů.
- 10) Pro najetí motoru do předem dané pozice vytvořte čtyři vstupní proměnné, a to `NULA`, `TISIC`, `DVA_TISICE` a `TRI_TRISTA`, které budou spouštět blok `MoveAbsolute_PTO`. Do těchto bloků zadejte rychlost otáčení 200 a pozici 0, 1000, 2000 a 3300. Tyto proměnné namapujte na vstupy `I1`, `I2`, `I3` a `I4`. Tlačítka `V1`, `V2`, `U1` a `U2` propojte s těmito vstupy a ve vizualizaci pro každé tlačítko vložte `LED` diodu pro signalizaci jejich aktivity. Při stisku těchto tlačítek motor i tisková hlava najede do nastavené pozice s krokem 200 na otáčku.
- 11) Pro vykonání určité vzdálenosti motoru využijte blok `MoveRelative_PTO`. Do bloku vložte proměnné `DISTANCE` a `VELOCITY`, které budou sloužit pro zadání vzdálenosti a rychlosti motoru ve vizualizaci. Pro spuštění bloku bude sloužit vstupní proměnná `MOVERELATIVE`, která bude vložena ve vizualizaci jako tlačítko. Při aktivní proměnné najede motor do zvolené pozice při zvolené rychlosti. Dále vložte blok `Halt_PTO`, který po vykonání funkce zastaví pohyb motoru.
- 12) Vytvořte vstupní proměnné `JED_NA_PLUS` a `JED_NA_MINUS`. Tyto proměnné namapujte na vstupy `I8` a `I9` a propojte s tlačítky `S1` a `S2` na `DPS`. Ve vizualizaci pro tyto tlačítka vložte `LED` diody pro jejich signalizaci. Do bloků `MoveRelative_PTO` zapište vzdálenosti 500 nebo -500 a rychlost 200. Při stisku těchto tlačítek motor vykoná 500 kroků vlevo nebo vpravo.
- 13) Podobně vytvořte vstupní proměnné `PLUS_REL` a `MINUS_REL`, které spolu s proměnnými `PLUS` a `MINUS` vepsané do bloku budou vloženy do vizualizace. Do bloku vepište rychlost otáčení motoru 200. Ve vizualizaci vytvořte okénka pro proměnné `PLUS` a `MINUS`, do kterých se zapiše zvolená vzdálenost a vložte tlačítka `S5` a `S6` představující vstupní proměnné, kdy při jejich povolení motor vykoná zvolenou vzdálenost.
- 14) Pro poslední funkci tohoto typu vytvořte vstupní proměnné `POS_REL` a `NEG_REL`. Do bloku vepište vzdálenost 2000 a -2000. Ve vizualizaci vytvořte pro proměnnou `POS_REL` tlačítko `S9`. Při povolení tohoto tlačítka motor urazí 2000 kroků a poté se vrátí o 2000 kroků zpět a zastaví se.

- 15) Pro stálé otáčení motoru použijte blok MoveVelocity_PTO, do kterého vložte rychlost motoru 200 a směr otáčení. Vytvořte vstupní proměnné MOVE_VELOCITY_P pro pohyb motoru vlevo a MOVE_VELOCITY_N pro pohyb vpravo. Tyto proměnné namapujte na vstupy I10 a I11 a tlačítka Vlevo a Vpravo na DPS připojte na tyto vstupy. Do vizualizaci pro tlačítka vložte LED diodu pro signalizaci, že jsou tlačítka stisknutá. Při stisku a držení těchto tlačítek se bude motor a tisková hlava pohybovat daným směrem. Po puštění tlačítek se sepne blok Halt_PTO a dojde k zastavení motoru.
- 16) Vytvořte vstupní proměnnou POS_VEL1, NEG_VEL1 a STOP_VEL. Proměnnou POS_VEL1 vložte do vizualizace jako tlačítka S8. Dále vložte časovače TON1 a TON2, do kterým se zapíše proměnná CAS. Pro tuto proměnnou vytvořte ve vizualizaci okénko pro zapsání zvoleného času otáčení. Při stisku tlačítka S8 se spustí blok pro pohyb motoru vlevo a poběží zvolená doba pohybu. Po dosažení času se povolí proměnná NEG_VEL1 a stejnou dobu se motor bude otáčet směrem vpravo. Po dosažení nastaveného času se sepne proměnná STOP_VEL, která dá povel k zastavení motoru.
- 17) Podobně vytvořte vstupní proměnné POST a NEG_T spolu s časovačem TON4 a TON5. Ve vizualizaci pro vstupní proměnné vložte tlačítka S3 a S4 a vložte okénka pro zvolení rychlosti a času otáčení. Při stisku těchto tlačítek se motor bude pohybovat směrem vpravo nebo vlevo při zvolené rychlosti a času ve vizualizaci. Po dosažení nastaveného času se sepne proměnná STOP_VEL, která dá povel k zastavení motoru.
- 18) Nakonec nahrajte vytvořený program do PLC.