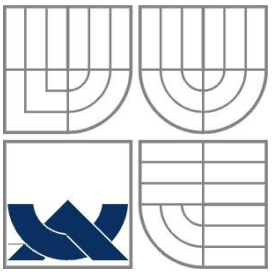
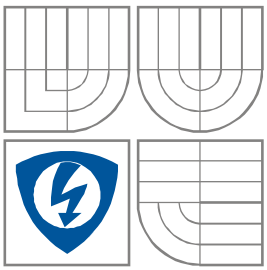


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

PÍSMO PRO 3D ZAPISOVAČ

LETTERING FOR 3D PLOTTER

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — JAROMÍR POLÁK
AUTHOR

VEDOUCÍ PRÁCE — Ing. RADEK ŠTOHL, Ph.D.
SUPERVISOR

BRNO 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav automatizace a měřicí techniky

Bakalářská práce

bakalářský studijní obor
Automatizační a měřicí technika

Student:	Jaromir Polák	***ID:***	115259
Ročník:	3	***Akademický rok:***	2010/2011

NÁZEV TÉMATU:

Písmo pro 3D zapisovač

POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s instrumentaci firmy Rockwell Automation a modelem manipulátoru-zapisovače.
2. Navrhněte a realizujte program pro řízení 3 os modelu zapisovače pro vykreslování české abecedy podle normovaného písma.
3. Parametry písma budou velikost písma a směr kreslení
4. Ověřte funkčnost programu na modelové úloze.

DOPORUČENÁ LITERATURA:

Kinetix 2000 Multi-axis Servo Drive. (User Manual) Milwaukee: Rockwell Automation, Inc. 2007. 226 s.

Dle vlastního literárního průzkumu a doporučení vedoucího práce.

Termín zadání:	7.2.2011	***Termín odevzdání:***	30.5.2011

Vedoucí práce: Ing. Radek Štohl, Ph.D.

prof. Ing. Pavel Jura, CSc.
Předseda oborové rady

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 zabývá řízením 3 os manipulátoru. Manipulátor má znázorňovat plotter, který má umožnit uživateli psaní celé české abecedy. U písmen můžeme měnit velikost písma, natočení kolem os a posunutí. V práci je i zahrnuta softwarová kontrola bezpečnosti a ukázka naprogramování jednoho písmene.

Klíčová slova

PLC, řízení, technické písmo, plotter, manipulátor, bezpečnost

Abstract

Bachelor thesis deals with the control of the 3 axes manipulator. The manipulator is represented as a plotter, to enable the user to write all of the Czech alphabet. The letters can change the font size, the rotation around the axis and move. The work also includes security and control tasks. An example of one letter shows programming features of the manipulator.

Keywords

PLC, control, technical font, plotter, manipulator, safety task

Bibliografická citace:

POLÁK, J. *Písmo pro 3D zapisovač.* Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2011. 42 s. Vedoucí bakalářské práce Ing. Radek Štohl, Ph.D..

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma Písmo pro 3D zapisovač jsem vypracoval samostatně pod vedením vedoucího bakalářské 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 bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské 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: **25. května 2011** ……
podpis autora

Poděkování

Děkuji vedoucímu bakalářské práce Ing. Radku Štohlovi, Ph.D. za odbornou pomoc a další cenné rady při zpracování mé bakalářské práce.

V Brně dne: **25. května 2011**

……
podpis autora

Obsah

1 Seznam obrázků 8
2 Seznam tabulek 9
3 Úvod 10
4 Rockwell Automation 11
4.1 Softwarová část 11
4.2 Hardwarová část 11
5 Manipulátor-zapisovač 13
5.1 Bezpečnost 14
5.2 Bezpečnostní tlačítko 14
5.3 Hardwarové limity 15
5.4 Softwarové limity 15
6 Hardwarová konfigurace 16
7 Programování 17
7.1 Vývojový diagram 17
7.2 Vytvořené instrukce 20
7.2.1 Nastavení logiky instrukce 21
7.2.2 Instrukce PNSP 21
7.2.3 Instrukce Cp 22
7.3 Instrukce MCT 23
7.4 Safety task 24
8 Návrh písmen 25
9 Ovládání MAnipulátoru 28
10 Závěr 31
11 Literatura 32
12 Seznam příloh 33

1 SEZNAM OBRÁZKŮ

Obrázek 1: PLC manipulátoru 11
Obrázek 2: PLC pro komunikaci se vstupy a výstupy 12
Obrázek 3: Optická závora 12
Obrázek 4: Servomotor 12
Obrázek 5: Manipulátor 13
Obrázek 7: Bezpečnostní tlačítko 14
Obrázek 6: Optická závora manipulátoru 14
Obrázek 8: Homing snímače (žluté) a spínače (červené) pro hardwarové limity 15
Obrázek 9: Maximální a minimální hodnoty softwarových limitů 15
Obrázek 10: Hardwarová konfigurace 16
Obrázek 11: Vývojový diagram programu 18
Obrázek 12: Vývojový diagram písmene 19
Obrázek 13: Nová instrukce 20
Obrázek 14: Instrukce PSNP 21
Obrázek 15: Instrukce Cp 22
Obrázek 16: Instrukce MCT 23
Obrázek 17: Otočení kolem osy Z 23
Obrázek 18: Reálné a virtuální osy, koordinační systém 23
Obrázek 19: Instrukce pro safety task 24
Obrázek 20: Písmo obkreslené z technické normy 25

2 SEZNAM TABULEK

Tabulka 1: Písmena 1. skupiny 26
Tabulka 2: Písmena 2. skupiny 27
Tabulka 3: ASCII kódy 28
Tabulka 4: Signalizace konfigurace, poruch 29
Tabulka 5: Funkce jednotlivých přepínačů 29

3 ÚVOD

Úkolem mé bakalářské práce je seznámit se s přístroji firmy Rockwell Automation pro model manipulátoru-zapisovače. Manipulátor neboli zapisovač je vhodný nástroj pro praktickou ukázku a práci s PLC.

Pod pojmem manipulátor si můžeme představit portálový jeřáb, který je určen k manipulaci se zařízeními v jeho dosahu. Pod pojmem zapisovač si můžeme představit přístroj, který pracuje na podobném principu jako plotter.

Právě tato představa je tématem mé bakalářské práce. Zapisovač má k dispozici pohyb ve 3 osách. K ose Z je upevněno pero, kterým budu demonstrovat funkci plotteru.

Práce bude směřována na ukázku funkčnosti zápisu písmene do paměti FIFO, dále k jejímu čtení a následnému „vytisknutí“. Požadavkem je možná změna velikosti písma, změna natočení směru psaní a psaní celé české abecedy. Je nutné navrhnout program, který bude řešit problematiku, která může nastat. A to:

- zajistit softwarovou bezpečnost, jelikož manipulátor je nemalých rozměrů (řádově stovky centimetrů) viz Obrázek 5Obrázek 5: Manipulátor a hmotnosti přesahující hmotnost dospělého člověka. Při nezabezpečení manipulátoru je zřejmé, že by mohl vážně ublížit na zdraví.
- zadávání souřadnic písmene, neboť tyto souřadnice se musí měnit v závislosti na velikosti písmene a natočení
- kontrola zda se písmeno vejde na řádek
- způsob zápisu písmen do paměti FIFO a jejich následné čtení
- způsob psaní diakritických znamének

4 ROCKWELL AUTOMATION

4.1 Softwarová část

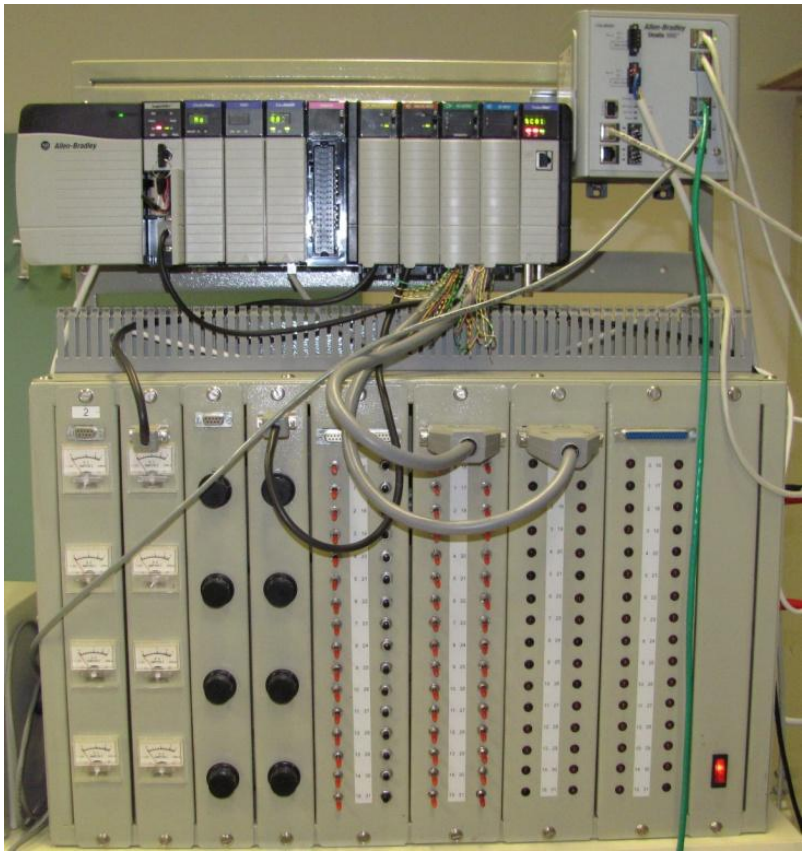
Od firmy Rockwell automation je použit software pro ovládání manipulátoru s názvem RSlogix 5000 v tomto programovacím prostředí byl vytvořen program v ladder diagramu, který byl nahrán do PLC. Dalším programem byl použit RSlinx, který zajišťoval komunikaci různých programu od firmy Rockwell Automation.

4.2 Hardwarová část

Z hardwarových prostředků, které byly použity na bakalářské práci: PLC na manipulátoru,
PLC pro komunikaci s výstupy a vstupy, optické závory, motory.

Obrázek 1: PLC manipulátoru

Obrázek 2: Optická závora



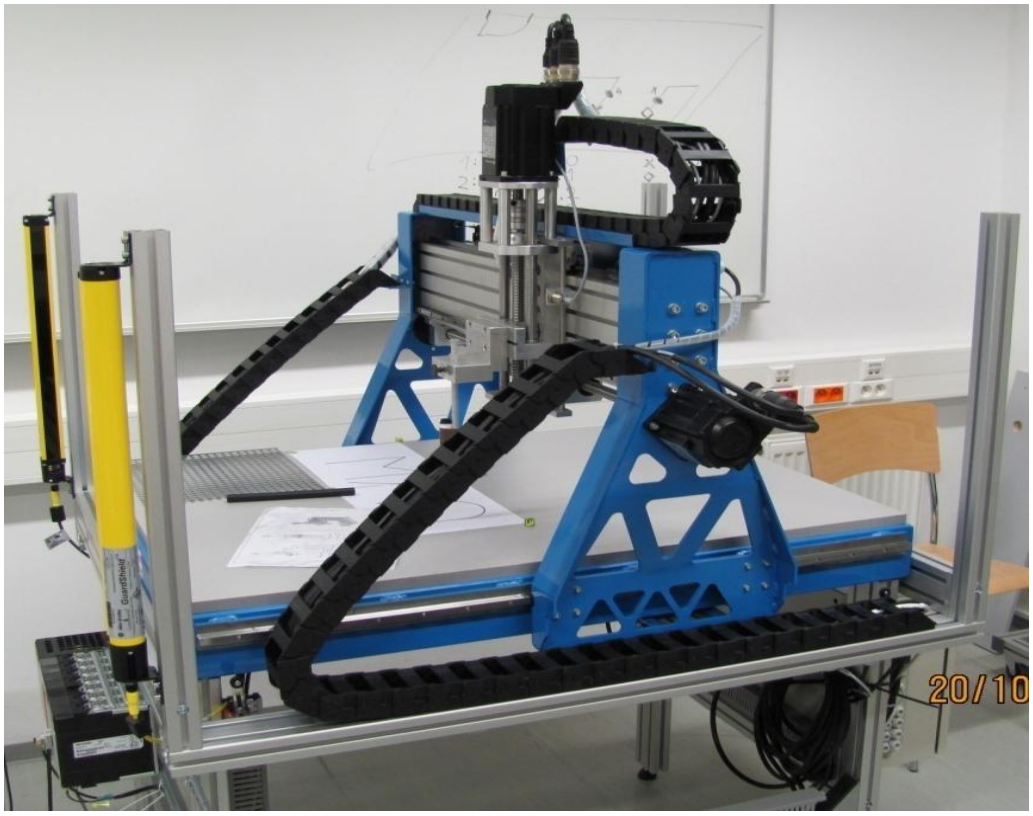
Obrázek 3: PLC pro komunikaci se vstupy a výstupy

Celkově jsou na manipulátoru tři motory. Každý motor pro jednu osu psaní.

Jeden motor je vybavený brzdou, kterou je možný aktivovat v případě vypnutí.Tento motor je na ose Z slouží k tomu, aby gravitační síla na něj nepůsobila tedy, aby motor po vypnutí nesjel dolů. Motor MPL-A310F-HK24AA má kroutící moment 1,58 Newton metra, 3000 otáček za minutu, digitální enkodér, brzdu, napájení na 230 V a váží 4,1 kg. Ostatní dva motory MPL-A310F-HK22AA jsou stejné, ale nemají brzdu.

Obrázek 4: Servomotor

5 MANIPULÁTOR-ZAPISOVAČ



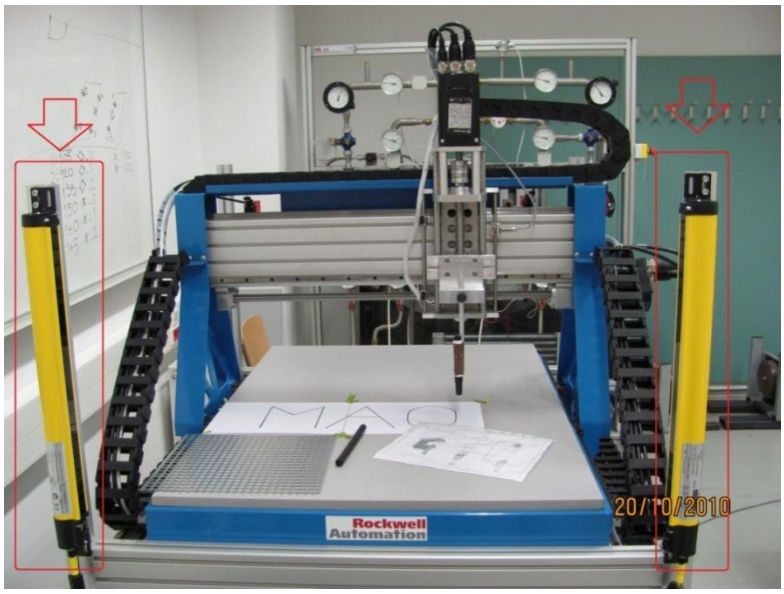
Obrázek 5: Manipulátor

Manipulátor je schopný pohybovat se ve 3 osách. Každá osa má dané softwarové a hardwarové limity. Dané limity mají za úkol uhlídat, aby se manipulátor nedostal za maximální nebo minimální hranici pojezdu. Jestliže se manipulátor přiblíží k hardwarové hranici, sepne se spínač a polohovací systém signalizuje chybu v programu PLC motor v daném směru se vypne. Na každém motoru je snímač polohy natočení hřídele, který nám umožňuje nastavit již zmíněné softwarové limity pomocí složitých přepočtů. V přepočtu je brán zřetel na Homing pozici, stoupání šroubu osy a polohy natočení hřídele.

Důležitá funkce manipulátoru je homing, bez které by softwarové limity nefungovaly. Funkce homing nastavuje polohu os manipulátoru pomocí indukčnostních snímačů polohy. Jakmile jsou na všech osách aktivní tyto snímače, je poloha zkalibrovaná na [0,0,0].

5.1 Bezpečnost

Optická závora je jeden z bezpečnostních prvků. Při přerušení světelného paprsku dojde k okamžitému softwarovému zastavení všech os při jakémkoliv procesu. Jestli-že se paprsek přeruší v průběhu psaní písmene je možné pomocí tlačítek resetovat manipulátor a tak pokračovat v psaní tam, kde jsme přestali. Podobně manipulátor zareaguje, zmáčkneme-li bezpečnostní tlačítko. Naše PLC je bezpečnostní a má Safety task, který kontroluje bezpečnost manipulátoru.



Obrázek 6: Optická závora manipulátoru

Typ závor: 440L-T4J0480YD
Výrobce: Rockwell Automation
Popis: bezpečnostní závora s 48 paprsky

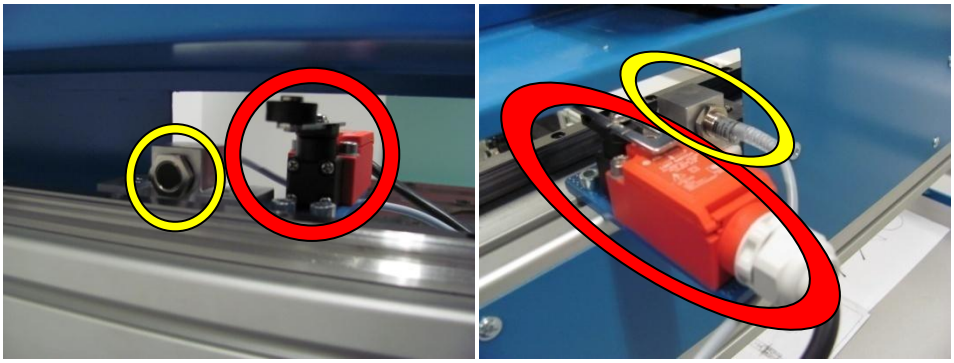
5.2 Bezpečnostní tlačítko

Slouží k okamžitému zastavení všech os motoru, z jakéhokoliv místa v programu.

Obrázek 7: Bezpečnostní tlačítko

5.3 Hardwarové limity

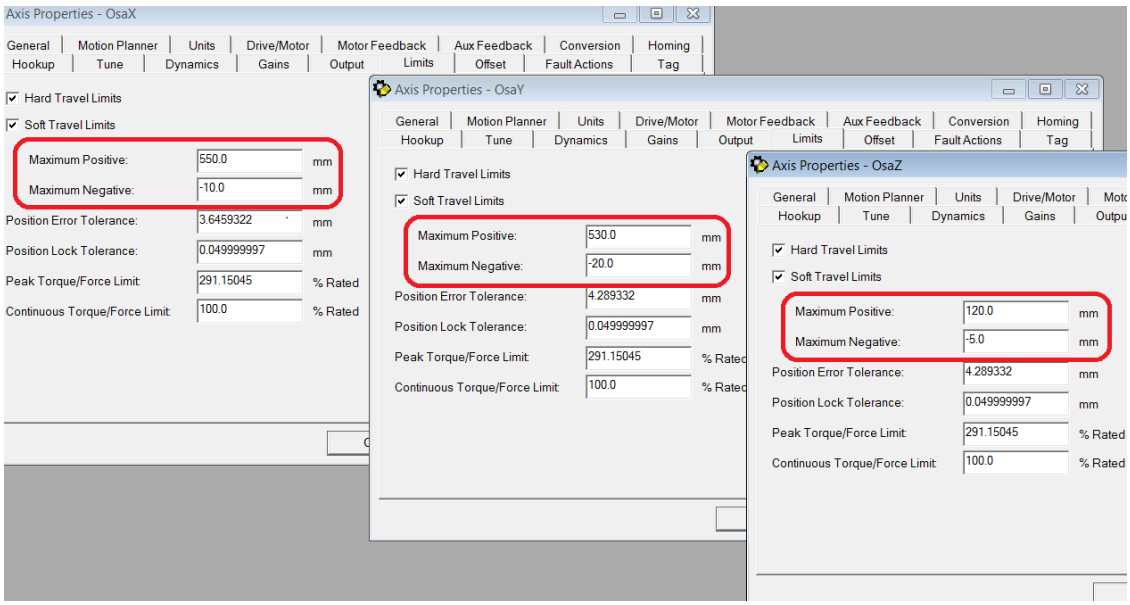
Hardwarové limity nám hlídají bezpečnost zařízení, které by mohlo vyjet z mezí os, a tím se vážně poškodit. Hardwarový limit nám osu, která sepne spínač, vypne a na dané ose se objeví chyba. Tato chyba se musí pomocí tlačítek odstranit



Obrázek 8: Homing snímače (žluté) a spínače (červené) pro hardwarové limity

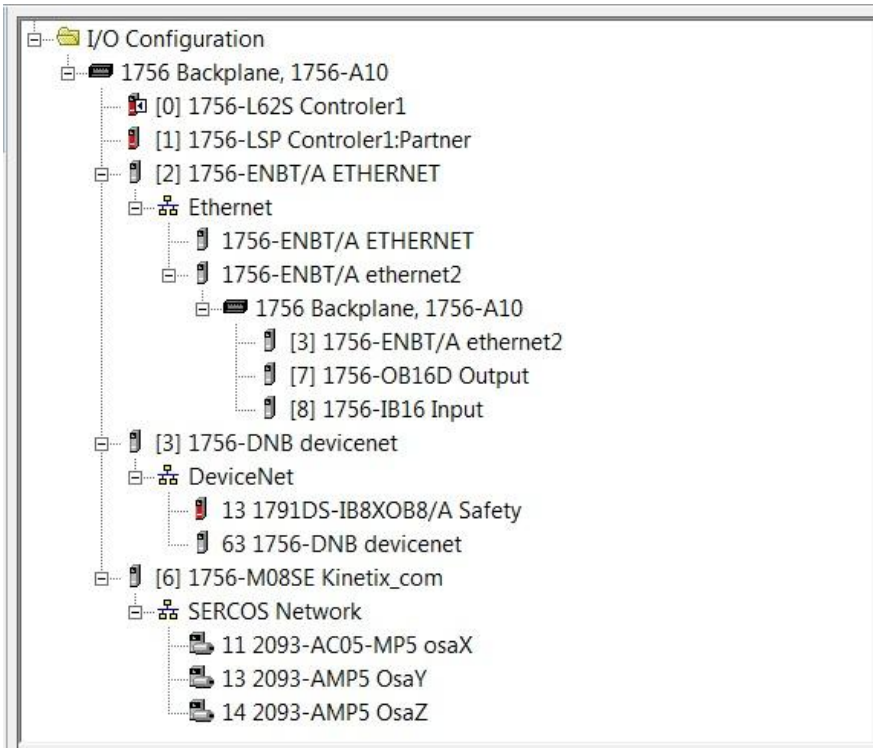
5.4 Softwarové limity

Softwarové limity hlídají osy, aby se nepřiblížili k hardwarovým limitům a tím nedošlo k vypnutí osy. Jak můžeme vidět na obrázku, jsou zde maximální a minimální hodnoty, které může manipulátor nabývat v dané ose.



Obrázek 9: Maximální a minimální hodnoty softwarových limitů

6 HARDWAROVÁ KONFIGURACE



Obrázek 10: Hardwarová konfigurace

Na obrázku č. 10 můžeme vidět konfiguraci PLC, kde najdeme kartu s digitálními výstupy ([7] 1756-OB16D Output), pomocí ní lze zobrazovat různé stavy za použití ledek viz Tabulka 4: Signalizace konfigurace, poruch. Další důležitá karta je digitální vstupy ([8] 1756-IB16 Input). Vstupy, které používáme jsou popsány v Tabulka 5: Funkce jednotlivých přepínačů. Další důležité karty jsou například bezpečnostní (ty jsou zobrazeny červeně), komunikační.

Bakalářská práce je inspirována bakalářskou prací pana Matějíčka [1]. Pro lepší seznámení jsem použil jeho vytvořený projekt v programovacím softwaru od firmy Rockwell Automation RSLogix5000, který jsem postupně upravoval podle svých představ a požadavků.

Hardwarové nastavení jsem použil opět od pana Matějíčka, jelikož tato část nebyla mým zadáním.

7 PROGRAMOVÁNÍ

Manipulátor se programuje v softwaru RSLogix5000. Je to vývojové prostředí, kde si můžeme vybrat ze 4 způsobů programování: strukturovaný text, žebříčkový diagram, sekvenční funkční schéma, blokové schéma.

Zvolil jsem si ladder (žebříčkový) diagram kvůli návaznosti na semestrální projekt. Žebříčkový diagram je rozdělen na jednotlivé řádky. Stejně jako má žebřík příčky. Na každý řádek můžeme vkládat instrukce dvojího typu:

- Input - tyto instrukce porovnávají, testují podmínky vstupů z PLC
- Output - vykonávají námi zadané procesy, úlohy

7.1 Vývojový diagram

Vývojový diagram se skládá ze 4 částí.

V první části se čeká, dokud se nenakonfigurují osy manipulátoru. Jakmile jsou nastaveny, tak pomocí tlačítka 0 viz Tabulka 5 se osy zinicializují.

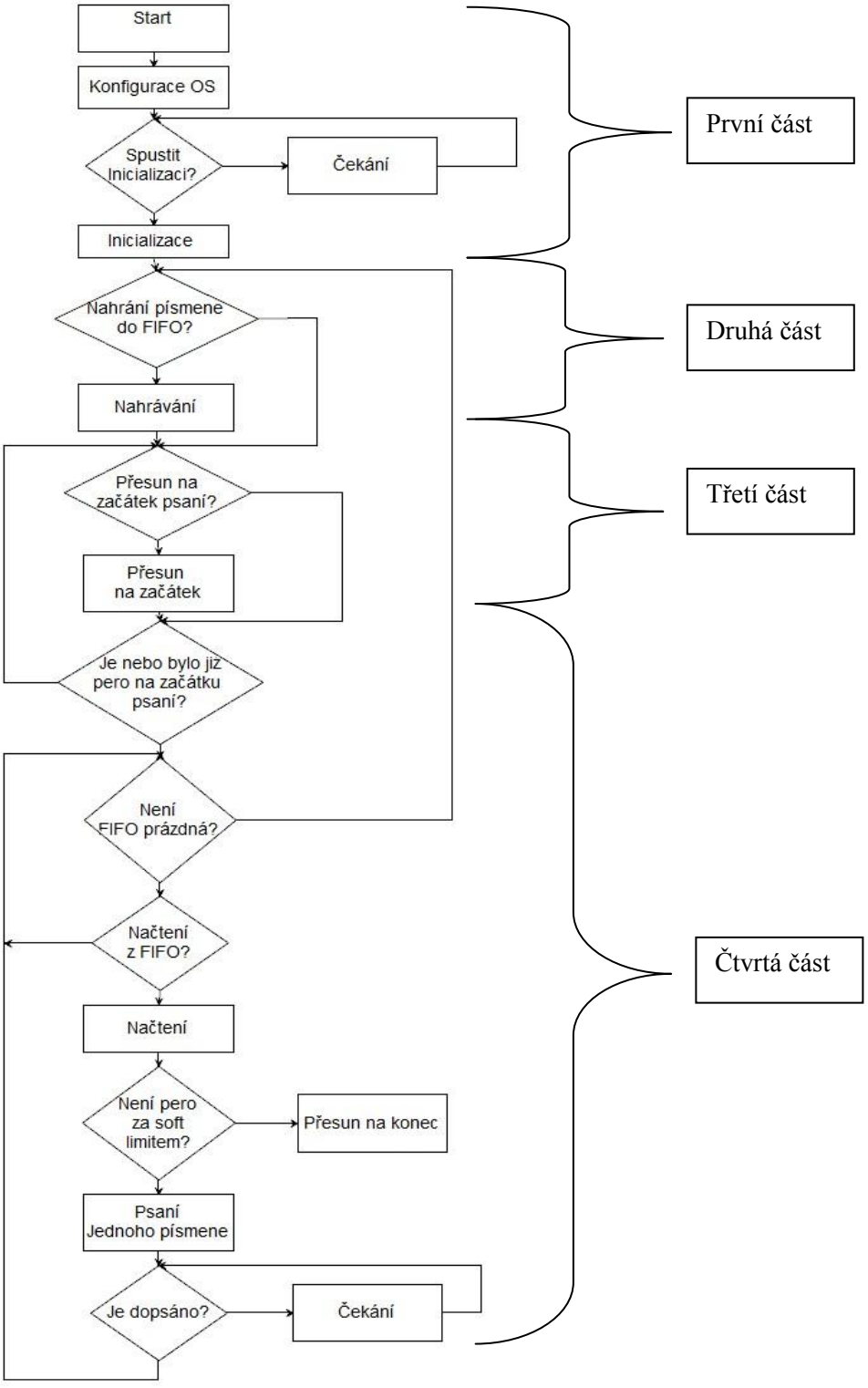
Druhá část se skládá z nahrání písmene do FIFO. V našem případě pomocí tlačítka 15 viz Tabulka 5. A ze zápisu písmene do programu. To je popsáno v kapitole číslo 10.

Během třetí části se pero pomocí signálu od tlačítka 1 viz Tabulka 5 přesune na začátek psaní a poslední část nám kontroluje, zda vše bylo provedeno správně:

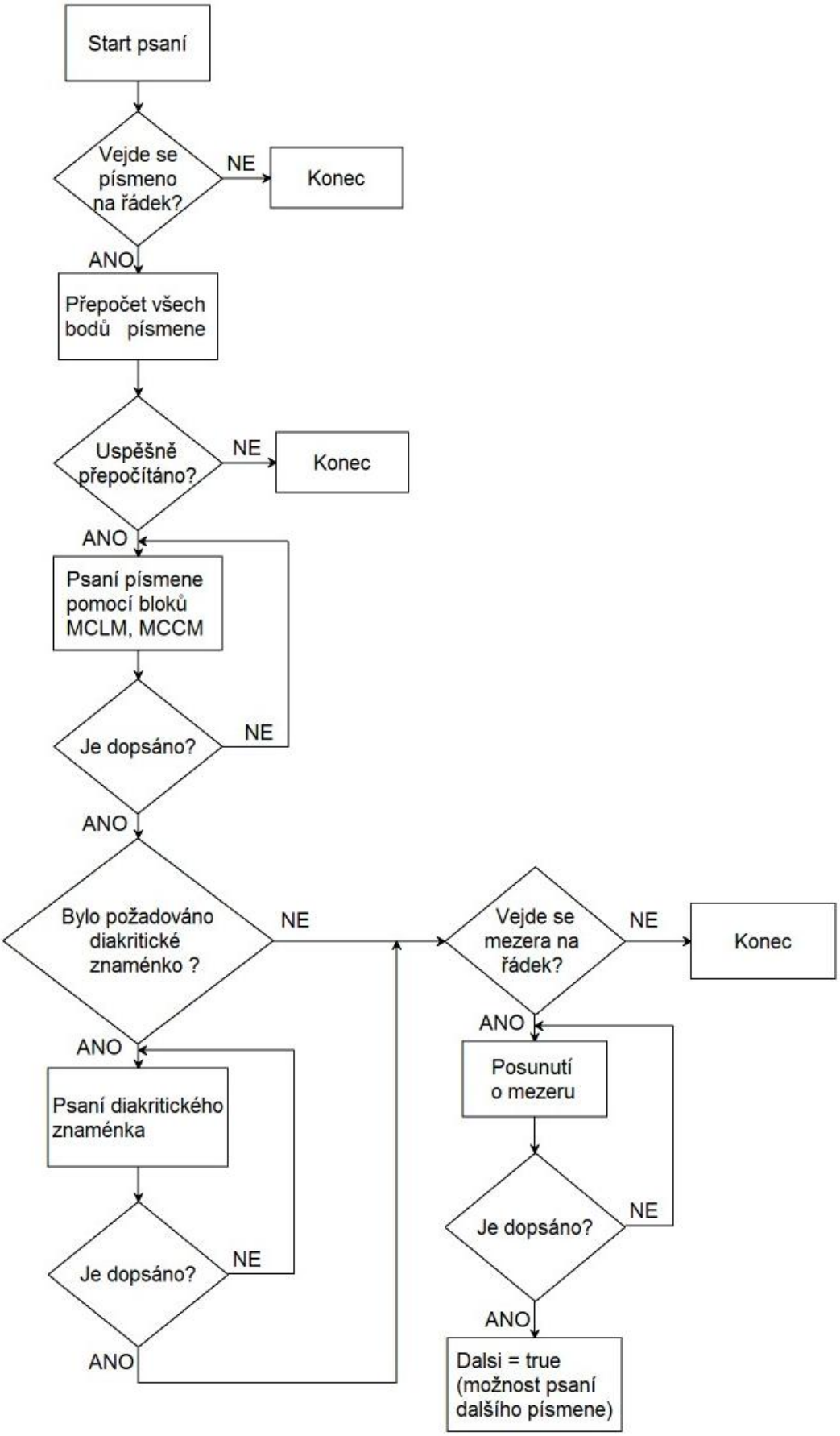
- jestli je pero na začátku nebo začátkem už prošlo, tedy už nepíšeme první písmeno
- přepočet písmene (natočení, posunutí) tlačítko 10 viz Tabulka 5
- abychom měli ve FIFO uloženo písmeno, aby bylo co psát
- kontrolu nad přesažení softwarového limitu koncovým bodem písmene

Jakmile jsou všechny tyto podmínky splněny, program si načte písmeno z FIFO a vytiskne ho na papír. Při psaní se v programu čeká až na dokončení písmene, tedy se během psaní neprovádí nic jiného krom SafetyTask. Jakmile je písmeno dopsáno, opět se vše zkontroluje, a buď se píše dál, nebo se čeká na uložení písmene do FIFO.

Čtvrtá část je proto popsána v dalším vývojovém diagramu pro svou obsáhlost viz Obrázek 12



Obrázek 11: Vývojový diagram programu



Obrázek 12: Vývojový diagram písmene

7.2 Vytvořené instrukce

Novou instrukci vytvoříme viz Obrázek 13**Chyba! Nenalezen zdroj odkazů.**, klikneme pravým tlačítkem myši a zvolíme „New Add-On Instruction…“ dále vyplníme tabulku, která nám nabídne:

- Name: jméno nové instrukce
- Type: typ programovacího jazyka
- Dále můžeme nastavit revisi, tedy kolikátá je to vývojová řada instrukce, např. verze 1.4 - můžeme v ní popsat co se změnilo

  Pokračujeme stisknutím OK

  Objeví se další tabulka, kde je nutno nastavit v jednotlivých záložkách:

- Parameters: parametry, které budou do instrukce vstupovat i vystupovat.

  Zadáme: Name

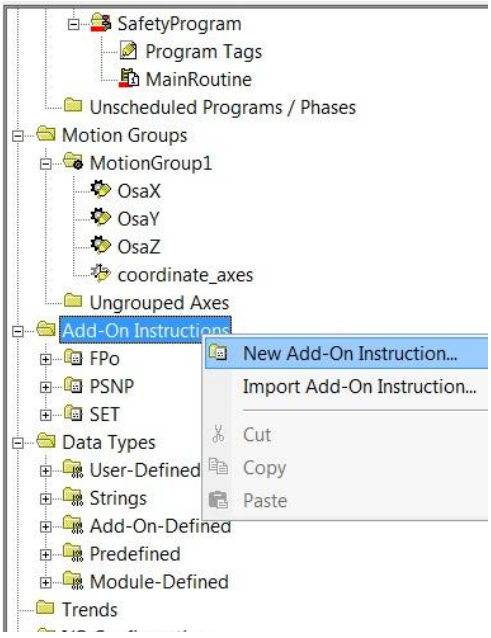
  Usage: jestli bude vstupovat nebo vystupovat

  Data Type: datový typ proměnné, se kterou budeme dále pracovat

  Default: hodnota proměnné, která bude výchozí a dvě důležité kolonky pro zatržení: Re a Vis

  Re neboli Required – znamená, jestli bude tato hodnota pro správnost chodu programu požadovaná nebo nikoliv

  Vis neboli Visibility – znamená, zda bude tato proměnná vidět na bloku instrukce
- Local Tags: jsou lokální proměnné. Tyto proměnné můžeme používat pouze v dané instrukci, nikde jinde ne.
- Další záložky nejsou pro naši potřebu již důležité.



Obrázek 13: Nová instrukce

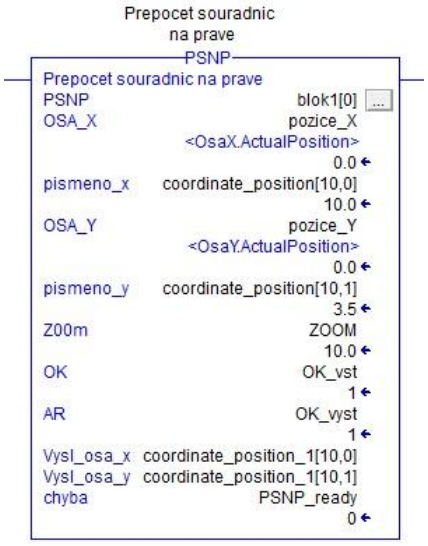
7.2.1 Nastavení logiky instrukce

Jestliže se nám podařilo vytvořit novou instrukci, musíme ji dále vytvořit logiku - co má provádět. Přidáme nový (Rung), novou příčku a zde vložíme naši novou instrukci, kterou najdeme v záložce Add-On. Klikneme na ni pravým tlačítkem myši a zvolíme „Open Instruction Logic“. Zde naprogramujeme potřebné podmínky, příkazy.

7.2.2 Instrukce PNSP

Funkce instrukce PNSP je přepočítávání souřadnice každého bodu písmene do souřadnic, ve kterých se právě pohybujeme. Uvedu příklad: každé písmeno je nadefinováno vlastním souřadnicovým systémem, kde začínám v [0,0]. Pokračuje například písmeno A bodem [10, 3.5] a pak další body.

Pokud bychom se právě pohybovali v souřadnicích [250,250] manipulator bude začínat v souřadnicích [250,250] a pokračovat v [260,253.5]. Pro výpočet bereme v potaz pouze dvě souřadnice, z důvodu toho, že třetí souřadnice může být nadefinována dopředu. Nezáleží na velikosti nebo na správných souřadnicích systému. Osa Z se bude zvedat a spouštět pořád stejně.



Obrázek 14: Instrukce PSNP

Dalším výpočtem je ZOOM. Zoom násobí souřadnici daného písmene a poté ji přičte k aktuálním souřadnicím. Podmínkou k provedení této instrukce je, aby výstup z instrukce Cp (tato instrukce je popsána v další kapitole) byla TRUE (logická 1). Je to proměnná OK_vst. Aby program mohl pokračovat dále, je v této instrukci výstup OK_vyst, který se dále kontroluje.

Popis:

- PSNP: zde zadáme (jestliže nemáme pouze jeden blok) nejlépe pole proměnné typu PSNP. Do naší proměnné blok1[0] se ukládají data pro tento konkrétní blok.
- OSA_X, OSA_Y: proměnná, kde je uložena aktuální pozice dané osy
- Písmeno_x, Písmeno_y: souřadnice x a y pro dané písmeno
- ZOOM: tato proměnná mění velikost písmene
- OK: vstup, který je popsán výše
- AR: výstup, který je popsán výše

- Výsl_osa_x, Výsl_osa_y: do těchto proměnných se uloží přepočítaná hodnota souřadnic, která se později načítá do pohybových instrukcí
- Chyba: tato proměnná je typu BOOL - zapne nám žárovku signalizující chybu výpočtu.

7.2.3 Instrukce Cp

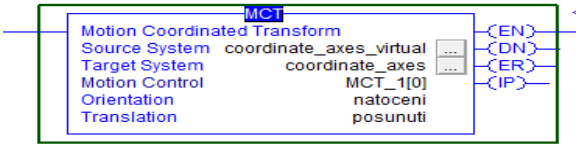
Instrukce Cp počítá cílovou pozici nebo pozici nejvzdálenější po ose kreslení. Tuto pozici srovnává se softwarovými limity. Pokud je výsledná pozice menší, je vše v pořádku a pokračujeme dále.

Popis:

- Cp: proměnná, kam se ukládají hodnoty jednotlivých bloků
- OSA_Y : aktuální pozice osy Y
- Finish_pos: nejvzdálenější bod písmene
- ZOOM: velikost písmene
- AR: proměnná, která nám umožňuje pokračování v programu
- Obnovení: vstupní podmínka, bez které by Cp nepočítalo. Defaultně je hodnota FALSE (logická 0).
- Spočítáno: podmínka pro pokračování program

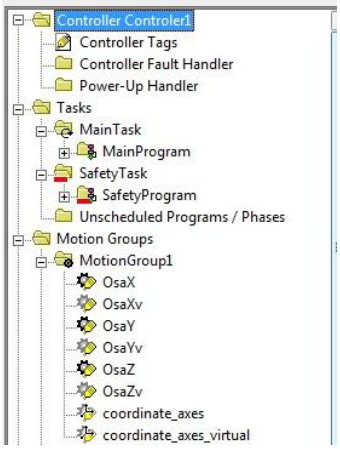
Obrázek 15: Instrukce Cp

7.3 Instrukce MCT

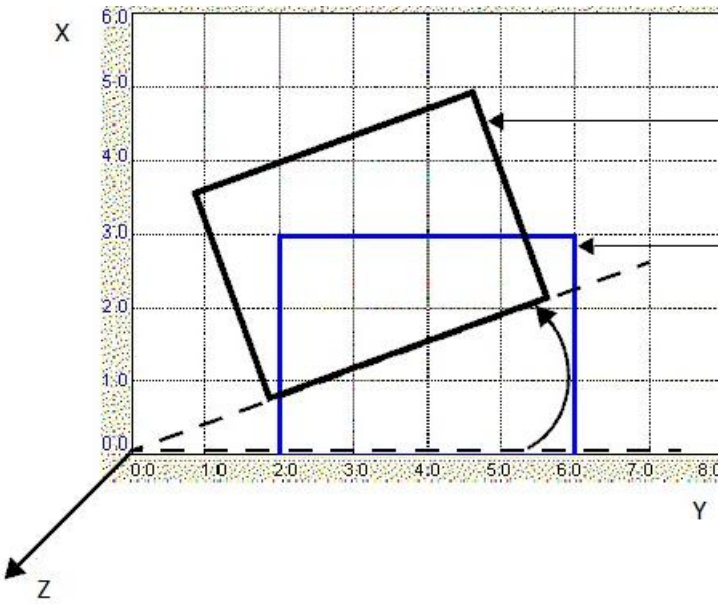


Obrázek 16: Instrukce MCT

Instrukce MCT slouží k transformaci souřadnicového systému. Source systém je zdrojový systém. Sem dosadíme virtuální koordinační systém, který budeme později dosazovat do pohybových instrukcí. Pro virtuální koordinační systém jsou potřeba vytvořit i virtuální osy X, Y, Z. Další proměnná je target systém zde přiřadíme koordinační systém reálný: systém manipulátoru. Motion kontrol-proměnná pro ukládání veškerých výpočtů. Orientation pole tří prvku typu real, tři prvky znázorňují osy x,y,z tedy jestliže pole prvků bude vypadat takto 0,20,0 tak se souřadnicový reálný systém pootočí kolem osy Y o 20°. Na podobném principu pracuje i proměnná translation, avšak jedná se o posunutí ne o natočení. Opět je to pole tří prvků. Posunutí reálného systému je v jednotkách, které jsou navoleny tedy, jestliže zadáme 20,0,0 reálný systém bude posunut po ose X o 20mm.



Obrázek 18: Reálné a virtuální osy, koordinační systém

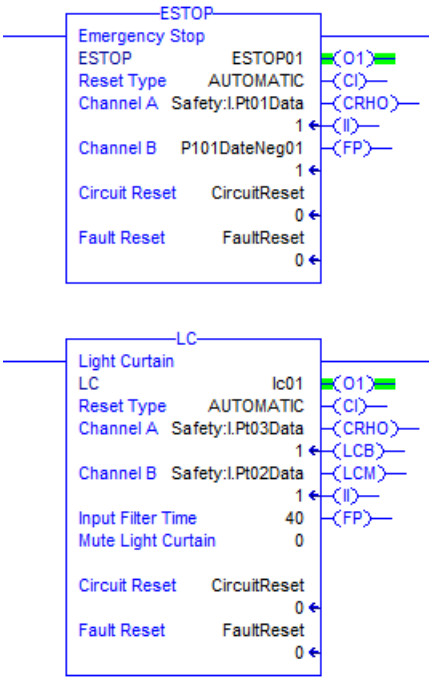


Obrázek 17: Otočení kolem osy Z

7.4 Safety task

Safety task, neboli bezpečnostní program, nám obsluhuje bezpečnostní prvky manipulátoru. Hlavním úkolem, je hardwarové zastavení os. Při narušení bezpečnosti se odpojí zdroje napájení.

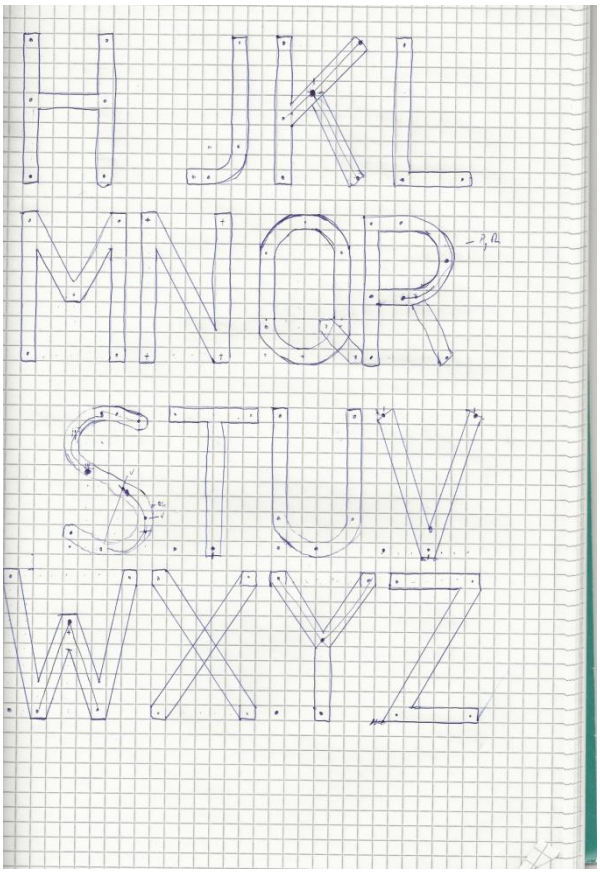
Přidal jsem softwarové nastavení, kdy při narušeni jsou všechny osy zastaveny. Optickou závoru jsem nastavil tak, aby kdykoliv se přeruší optické paprsky, se manipulátor zastavil a neprováděl žádný pohyb, dokud se nevypne přepínač, kterým se chyba optické závory vymaže. Po vymazání chyby se musí program spustit od začátku. Bezpečnostní tlačítko funguje na podobném principu. Na obrázku č. 19 můžeme vidět dvě instrukce, které se používají v safety tasku. Pomocí těchto instrukcí se hlídá již zmíněná optická závora (LC) a bezpečnostní tlačítko (ESTOP)



Obrázek 19: Instrukce pro safety task

8 NÁVRH PÍSMEN

Pro lepší orientaci v programu a pro budoucí práci s ním jsem písmena rozdělil na dvě skupiny. První skupina byla písmena, která potřebují k vykreslení pouze přímky, to jsou tedy písmena: A,E,F,H,I,K,L,M,NT,V,W,X,Z,Y. Druhou skupinu tvořila zbývající řada písmen: B,C,D,G,O,P,Q,R,S,U. Pro každou skupinu jsem vytvořil proměnné CP_1, do které se ukládaly výpočty cílových pozic (zda se písmeno vejde do softwarových limitů) dále pak PSNP_1 i zde byli ukládány výpočty a výsledky výpočtu, které nám vrátili hodnoty souřadnic spočítané v závislosti na natočení na velikosti písmene. Další proměnou je coordinate, zde se ukládali výpočty z bloku MCML (blok pro ovládání pohybu) a poslední proměnou je STEP_1, zde jsou jednotlivé kroky písmene výpočet, kreslení. Poslední proměnná se týká jen druhé skupiny, coordinate_o_3. Do této proměnné se ukládají výpočty z bloku MCCM pohybu po kružnici.



Obrázek 20: Písmo obkreslené z technické normy

Tabulky, které jsou zde uvedeny, slouží pro lepší orientaci v programu. Tedy když bych potřeboval něco změnit, či se na něco podívat pomocí těchto dvou tabulek si mohu každé písmeno najít. U každé proměnné, která je použita u daného písmene jsem úmyslně nechal rezervy pro případné změny.

Tabulka 1: Písmena 1. skupiny

Písmeno	CP_1	PSNP	COORDINATE_1	STEP_1
A	0	0-9	0-9	6-30
E	2	11-22	13-26	31-56
F	3	22-31	26-36	57-78
H	4	32-45	37-50	79-106
I	5	46-51	51-56	107-118
K	6	52-64	57-69	119-145
L	7	65-73	70-78	146-163
M	8	74-81	79-86	164-179
N	9	82-89	87-94	180-195
T	10	90-100	95-105	196-217
V	11	101-108	106-113	218-233
W	12	109-118	114-123	234-253
X	13	119-127	124-132	254-271
Y	14	128-139	133-144	272-295
Z	15	140-147	145-152	296-311
MEZERA	1	10,148-150	12,154,155	-
´	-	160-166	170-176	320-333
ˇ	-	167-174	177-184	334-349
°	-	175-181	185-190	350-362
START POZICE	-	-	153	1-2

Program jsem se snažil navrhnout co možná nejjednodušeji, tedy jsem psaní písmen s háčkem, čárkou nebo kroužkem napsal zvlášť. Vytvořil jsem samotné podprogramy těchto tří znaků. Na konci každého písmene se zkontroluje, jestli nebyl požadavek na napsání písmene se znakem. Jestliže byl, zavolá se daný podprogram a k písmenu se dopíše znak. V podprogramu těchto znaků se zkontroluje šířka písmene a podle této šířky se teprve začne psát znak tak, aby byl napsán podle normy. Jakmile se znak dopíše, vrátí proměnou, která je porovnána v programu písmene, jestliže je tedy písmeno napsáno, diakritické znaménko je napsáno a vrátí proměnnou v true tak se z paměti FIFO načte další písmeno.

Písmo, které jsem použil ke psaní manipulátoru je technické písmo kolmé [3]. Spočítal jsem souřadnice pro každé písmeno pomocí obrázku č. 12. Tento obrázek jsem překreslil z technické normy, která nemůže být kopírována jinak než ručně opsána.

Například pro písmeno M jsou souřadnice: počátek [0,0], následující bod je vrchol [10,0], dále pak [6,3.5] druhý vrchol [10,7] a konec [0,7]. Manipulátor pracuje v metrické soustavě tedy, jestliže necháme ZOOM na hodnotě 10 bude písmo veliké 10cm. Velikost proměnné ZOOM je nastaveno na hodnotu 10 záměrně, neboť pero, které je připevněno k ose Z, má tloušťku čáry cca 2mm, proto není vhodné psát písmo příliš malé.

Spočítané souřadnice jsem uložil do proměnných, s kterými dále pracuji pro výpočet souřadnic, které jsou aktuální.

Tabulka 2: Písmena 2. skupiny

Písmeno	CP_2	PSNP_2	COORDINATE_2	STEP_2	COORDINATE_O_3
O	0	0-11	0-7	0-20	0-2
J	2	12-20	8-15	21-37	3
B	3	21-33	16-25	38-63	4-6
C	4	34-45	26-33	64-85	7-9
D	5	46-58	34-42	86-109	10-12
G	6	59-73	43-53	110-137	13-15
Q	7	74-88	54-64	138-166	16-18
P	8	89-98	65-72	167-185	19-20
R	9	99-109	73-81	186-207	21-22
S	10	110-122	82-90	208-232	23-25
U	11	123-133	91-99	233-252	26-27
°	-	-	-	-	28-29

9 OVLÁDÁNÍ MANIPULÁTORU

Po zapnutí napájení, vyčkáme na to, až se osy zkalibrují. To můžeme vidět na signalizaci ledky č. 0. Jakmile zhasne je vše v pořádku a můžeme pokračovat.
Následně musíme zinicializovat osy. To se provede přepínačem č. 0. Poté v programu do proměnné „ZOOM“ zadáme hodnotu velikosti písmene v centimetrech, natočení písmene ve stupních do proměnné „natoceni“, posunutí písmene v milimetrech do proměnné „posunutí“, dále pak můžeme pokračovat nahráním písmen do paměti FIFO. Zadávání hodnot do proměnných natočení a posunutí je popsáno v kapitole 237.3.

Nahrávání se provádí zadáním písmene do proměnné FFL. Sem napíšeme jedno velké písmeno z abecedy, jestliže chceme písmeno z diakritickým znaménkem, musíme zadat toto písmeno v ascii kódu, tedy například Ž nahrajeme pomocí čísla 142.

Tabulka 3: ASCII kódy

ASCII kód	138	141	142	193	200	201	204
znak	Š	Ť	Ž	Á	Č	É	Ě
205	207	210	211	216	217	218	221
Í	Ď	Ň	Ó	Ř	Ů	Ú	Ý

Přepínačem č. 15 ho nahrajeme do FIFO. Stejným způsobem můžeme nahrát více písmen, např. 10. Jakmile je nahrávání u konce, přejdeme na psaní.

Nejprve přesuneme pero do pozice, ze které se začíná psaní. Toto provedeme přepínačem č. 1. Jakmile se manipulátor zastaví, přepínačem č.10 se zapne a vypne a tím se dá impulz pro přepočet písmene z virtuální osy do reálné. Poté můžeme přepínač č. 12 použit jako tlačítko (zapnout vypnout). Tím dáme prvotní signál pro začátek psaní. Pokud by přepínač zůstal v zapnuté poloze, stále by opakoval písmeno, které je zrovna v proměnné FFU. Tato proměnná čte z FIFO. Jestliže je v paměti FIFO uloženo více písmen než je manipulátor schopen napsat, dostane se za softwarové limity. Tím pádem se přesune do polohy [0,0,0] a rozsvítí se ledka č. 4.

Tabulka 4: Signalizace konfigurace, poruch

Číslo ledky	Signalizace:
0	Konfigurace os manipulátoru
1	Aktivní optická závora nebo tlačítko
2	Chyba na motorech
3	Špatné spočítané hodnoty z bloku PNSP
4	Cílová pozice písmene je za softwarovými limity

Tabulka 5: Funkce jednotlivých přepínačů

Číslo přepínače:	Funkce:
0	Inicializace os manipulátoru
1	Přesun na začátek psaní
2	Vypne motory
3	Vymaže chyby na motorech
4	Vymaže chybu vzniklou optickou závorou
5	Vymaže chybu při přepočtu, nebo je cílová pozice písmene větší než softwarové limity
7	Reset písmene
8	Nulování Proměnné STEP, STEP_1
10	Přepínač pro zapnutí přepočtu virtuální osy na reálnou
12	Načítání písmen z FIFO
15	Ukládání písmen do FIFO

Pokud nastane chyba, a rozsvítí se ledka č. 2, je nutné zkontrolovat, zda není manipulátor již za hardwarovými limity. Jestliže není, tak se pokusíme vymazat chybu přepínačem č. 3. Pokud chyba i nadále přetrvává, musíme vypnout motory a nechat zapnutý přepínač pro mazání chyby. Poté by již měl manipulátor najet do požadované polohy. Nejčastěji se problém vyskytne při konfiguraci a následné inicializaci.
Další problém, který může nastat je, že se při psaní přeruší světelný paprsek závor.
Tento stav vypne všechny motory. Abychom mohli opět pokračovat v psaní, musíme nejprve přepnout přepínač č. 4 pro vymazání chyby (alarmu) na optické závoře. Jakmile ledka č.1 zhasne, můžeme přepínač opět vypnout. Poté se zapne a následně vypne přepínač č.8, ten nám vynuluje proměnné STEP, STEP_1 abychom mohli dát opět náběžnou hranou prvotní impulz pohybovým instrukcím. Posledním krokem je zapnutí a vypnutí přepínače číslo 7, který nahraje poslední známou hodnotu proměnných STEP a STEP_1 právě do těchto proměnných. Tím jsme dali opět impulz pro psaní a tedy tam, kde psaní skončilo, tam pokračuje.

10 ZÁVĚR

V bakalářské práci jsem se seznámil s přístroji od firmy Rockwell Automation. Naprogramoval jsem manipulátoru tak, aby fungoval jako plotter, který píše celou českou abecedu. Při psaní se dá změnit velikost písma. Jednotlivá písmena nebo celá slova lze natáčet ve třech osách (teoreticky) prakticky jen ve dvou, za pomocí udání hodnoty natočení ve stupních. Posunutí písmen či slov lze provést ve třech osách při nastavení hodnoty v milimetrech. A také je možné libovolně měnit text přes paměť FIFO. Natáčet písmo sice lze teoreticky ve 3 osách, ale prakticky jen ve 2.

V programu jsou řešeny chybové stavy, které mohou nastat. Softwarově jsem zprovoznil optickou závoru, bezpečnostní tlačítko. V programu jsem myslel i na možné stavy typu: cílová pozice písmene je za softwarovým limitem, špatný přepočet souřadnic písmene na aktuální pozici. Jestliže je při psaní písmene porušena bezpečnost, zastaví se všechny tři osy. A čeká se na vypnutí obsluhou. Pokud je chyba restartována pokračuje se v psaní tam, kde se skončilo.

Tuto bakalářskou práci je možné rozšířit o:

- při překročení cílové pozice za softwarový limit se přesunout na další řádek a pokračovat dál.
- psaní velkých a malých písmen.
- psaní různými styly písma
- vizualizaci

Možnosti manipulátoru nejsou zdaleka vyčerpány. Jediná nevýhoda je, že při práci musí být student u manipulátoru.

V bakalářské práci je uvedeno psaní písmene O (viz příloha č.1), není zde uveden celý program pro svou obsáhlost (cca 320 stránek), proto je přiložen v příloze. Popisuji zde hlavně algoritmus uložení do FIFO, důležité instrukce pro přepočty a natočení, následné vytištění.

11 LITERATURA

[1] MATĚJÍČEK, Miroslav. Řízení 3 os manipulátoru*. Brno, 2010. 61 s. Bakalářská práce. Vysoké učení technické, fakulta elektrotechniky a komunikačních technologií.*

[2] CHLAD, Petr. Vizualizace manuálního ovládání modelu manipulátoru*. Brno, 2010. 45 s. Bakalářská práce. Vysoké učení technické, fakulta elektrotechniky a komunikačních technologií.*

[3] DRASLÍK, František. Normativně technická dokumentace: pravidla tvorby a používání*. Ostrava: Montanex, 1998. 287 s. ISBN 80-85780-91-7.*

[4] User Manual *[online]. 2008 [cit. 15. 12. 2010]. Logix 5000 Controllers Motion Instruction . Dostupné z WWW: <http://literature.rockwellautomation.com/idc/groups/literature/docum ents/rm/1756-rm007_-en-p.pdf >.*

[5] On Instructions *[online]. 2009 [cit. 15. 12. 2010]. Logix5000 Controllers Add. Dostupné z WWW: <http://literature.rockwellautomation.com/idc/groups/literature/documents/pm/1756-pm010_-en-p.pdf>.*

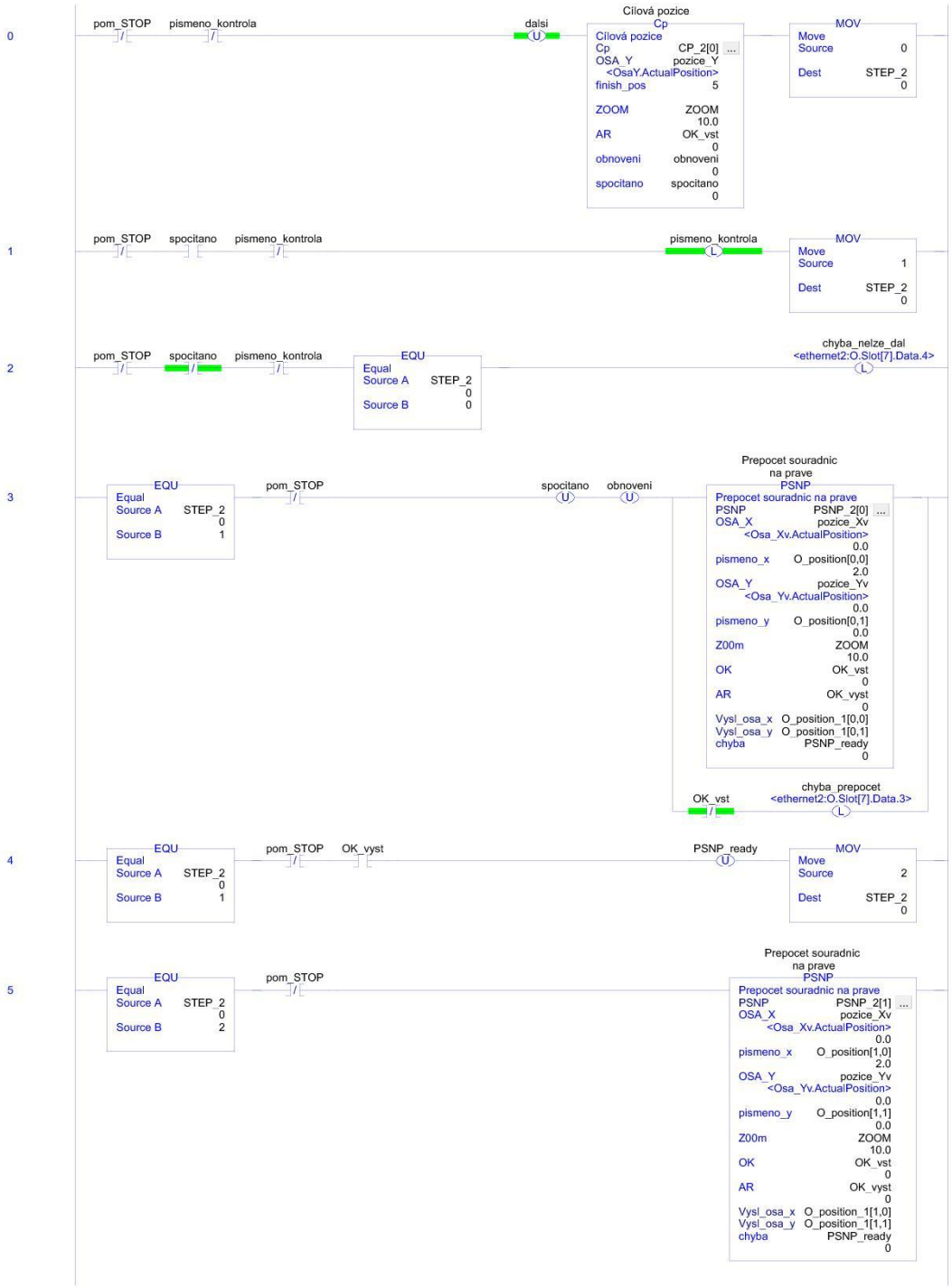
[6] KLETEČKA, Jaroslav; FOŘT, Petr. Technické kreslení*. [s.l.] : Computer Press, 2005. 252 s. ISBN 978-80-251-1887-0.*

12 SEZNAM PŘÍLOH

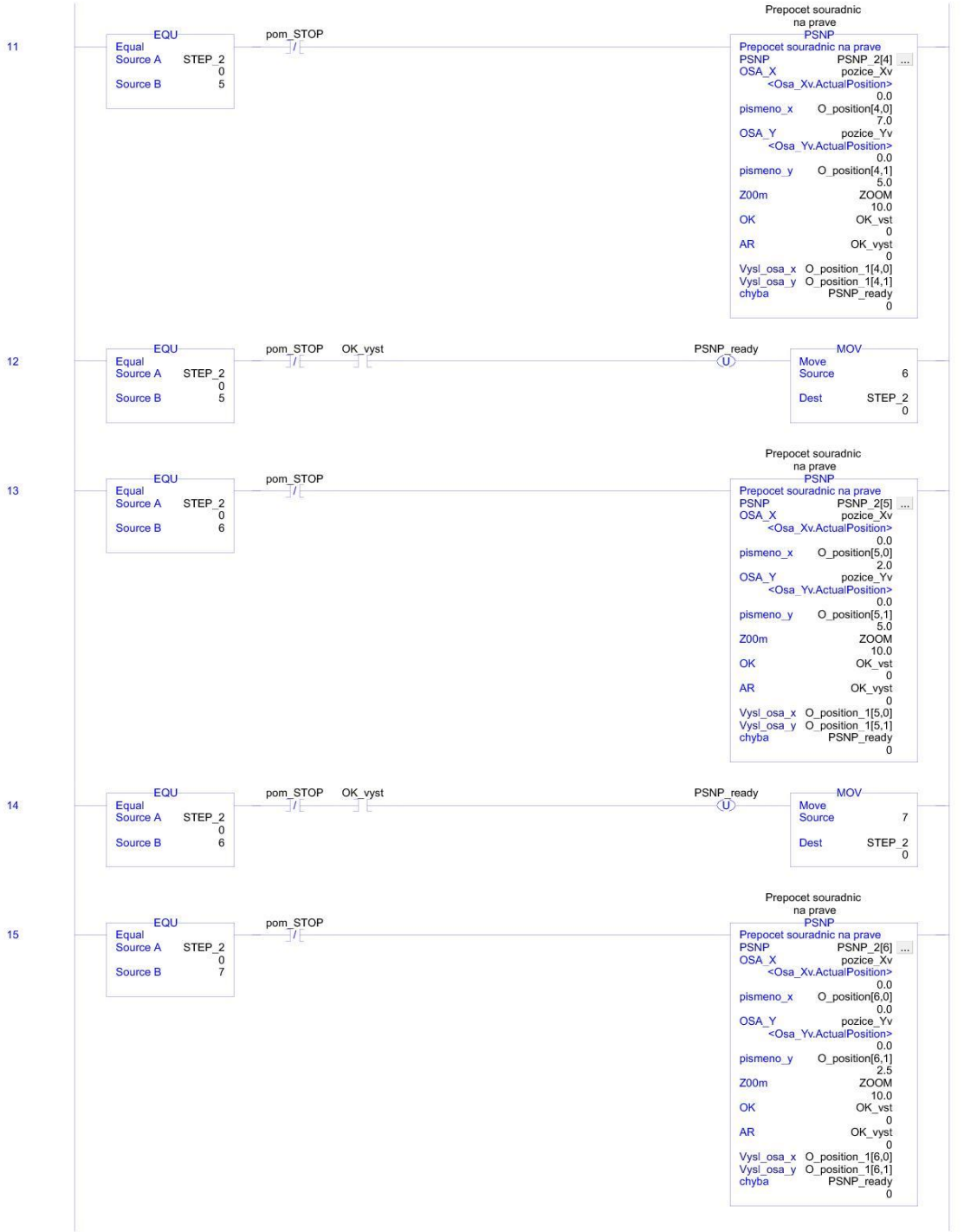
- Příloha č.1 ukázka programu písmene O

- CD, které obsahuje:
  - Report programu ve formátu pdf
  - Funkční program
  - Dokumentace v elektronické podobě

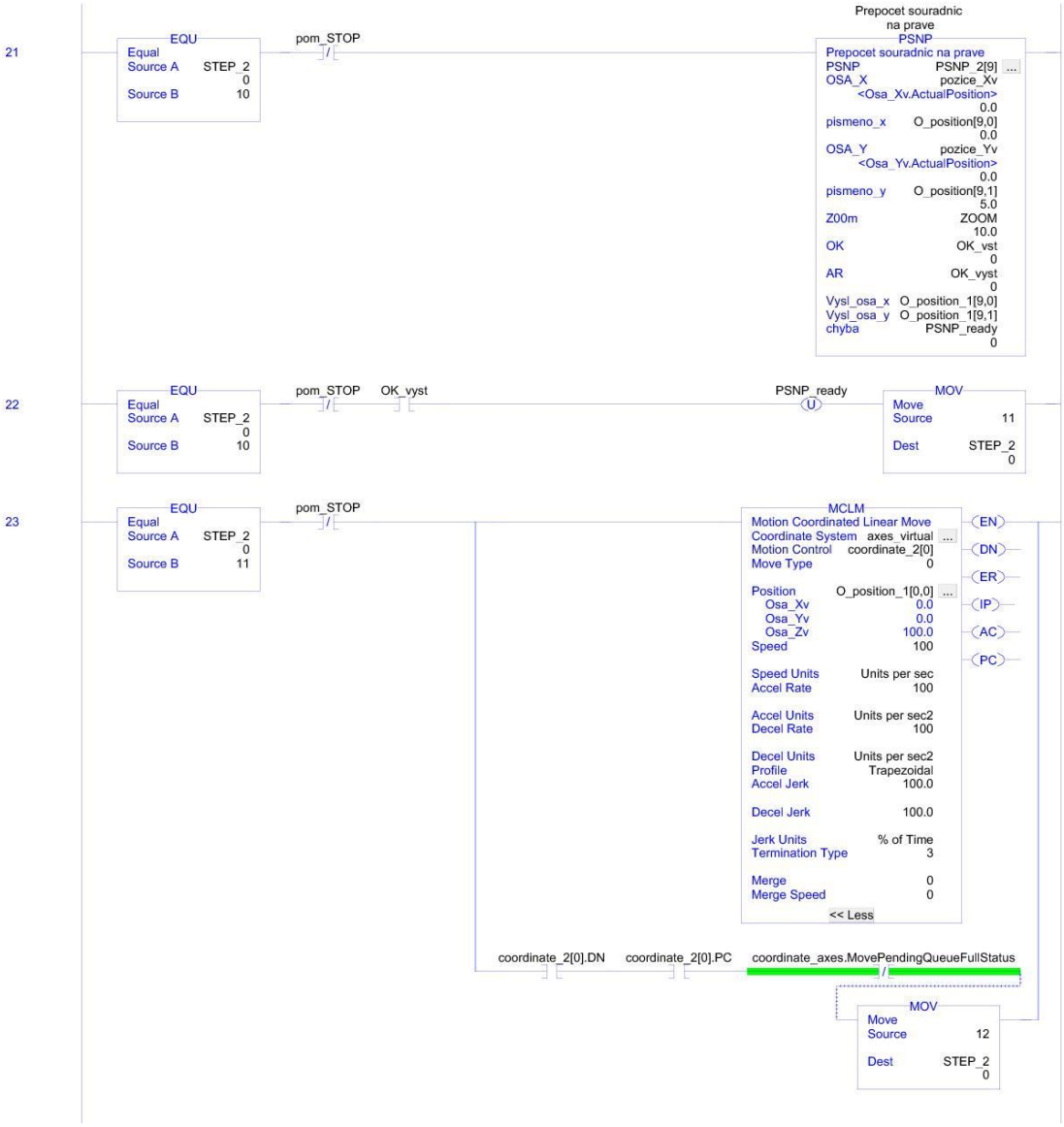
PŘÍLOHA Č.1


0
pom_STOP
pismeno_kontrola
dalsi
Cílová pozice
Cp
Cílová pozice
Cp CP_2[0]
OSA_Y pozice_Y
<OsaY.ActualPosition>
finish_pos 5
ZOOM ZOOM
10.0
AR OK_vst
0
obnoveni obnoveni
0
spocitano spocitano
0
MOV
Move
Source 0
Dest STEP_2
0
1
pom_STOP
spocitano
pismeno_kontrola
pismeno_kontrola
MOV
Move
Source 1
Dest STEP_2
0
2
pom_STOP
spocitano
pismeno_kontrola
EQU
Equal
Source A STEP_2
0
Source B 0
chyba_nelze_dal
<ethernet2:O.Slot[7].Data.4>
3
EQU
Equal
Source A STEP_2
0
Source B 1
pom_STOP
spocitano
obnoveni
Prepocet souradnic
na prave
PSNP
Prepocet souradnic na prave
PSNP PSNP_2[0]
OSA_X pozice_Xv
<Osa_Xv.ActualPosition>
0.0
pismeno_x O_position[0,0]
2.0
OSA_Y pozice_Yv
<Osa_Yv.ActualPosition>
0.0
pismeno_y O_position[0,1]
0.0
Z00m ZOOM
10.0
OK OK_vst
0
AR OK_vyst
0
Vysl_osa_x O_position_1[0,0]
Vysl_osa_y O_position_1[0,1]
chyba PSNP_ready
0
OK_vst
chyba_prepocet
<ethernet2:O.Slot[7].Data.3>
4
EQU
Equal
Source A STEP_2
0
Source B 1
pom_STOP
OK_vyst
PSNP_ready
MOV
Move
Source 2
Dest STEP_2
0
5
EQU
Equal
Source A STEP_2
0
Source B 2
pom_STOP
Prepocet souradnic
na prave
PSNP
Prepocet souradnic na prave
PSNP PSNP_2[1]
OSA_X pozice_Xv
<Osa_Xv.ActualPosition>
0.0
pismeno_x O_position[1,0]
2.0
OSA_Y pozice_Yv
<Osa_Yv.ActualPosition>
0.0
pismeno_y O_position[1,1]
0.0
Z00m ZOOM
10.0
OK OK_vst
0
AR OK_vyst
0
Vysl_osa_x O_position_1[1,0]
Vysl_osa_y O_position_1[1,1]
chyba PSNP_ready
0

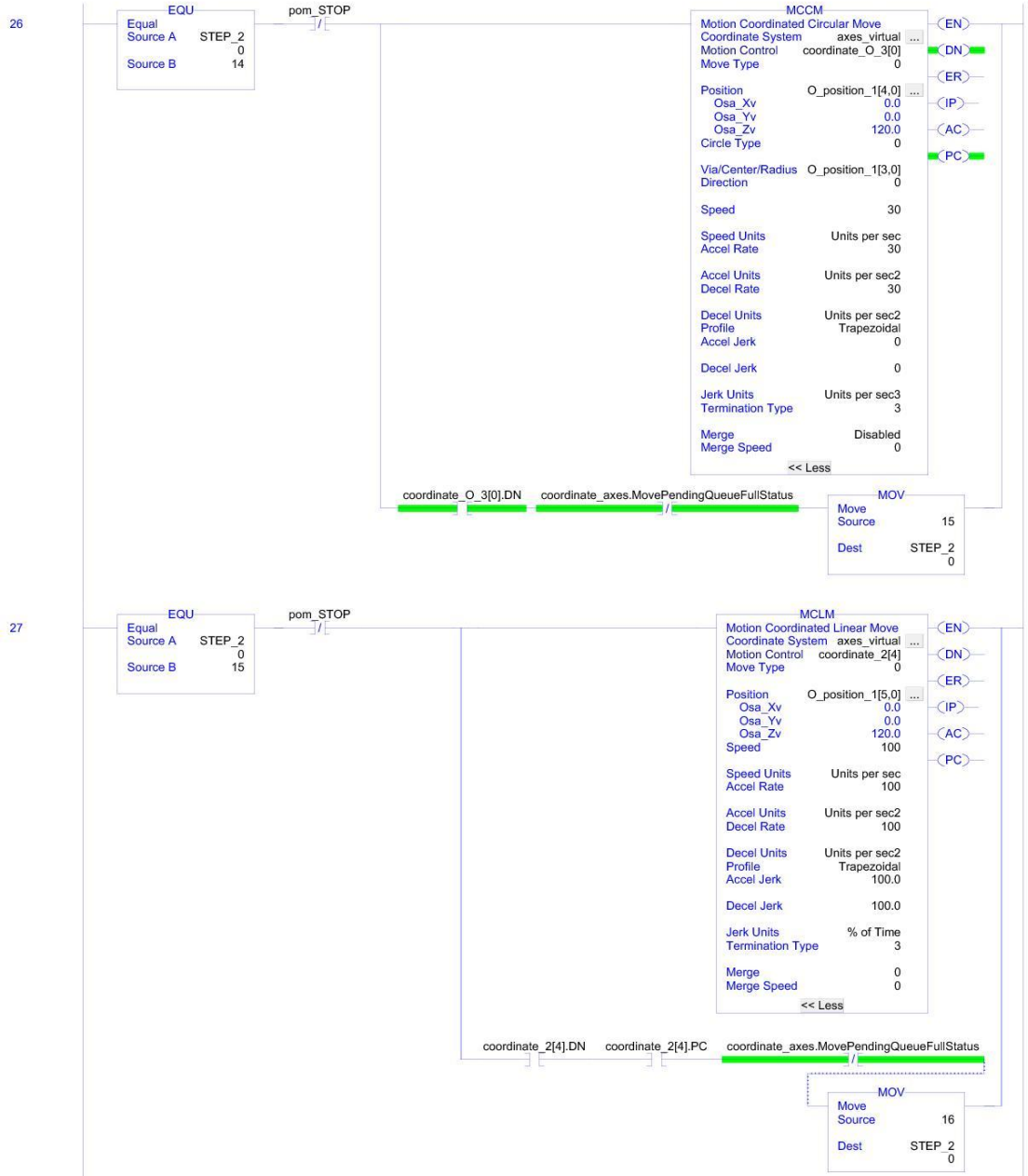
6
EQU
Equal
Source A STEP_2 0
Source B 2
pom_STOP
OK_vyst
PSNP_ready
U
MOV
Move
Source 3
Dest STEP_2 0
7
EQU
Equal
Source A STEP_2 0
Source B 3
pom_STOP
Prepocet souradnic na prave
PSNP
Prepocet souradnic na prave
PSNP PSNP_2[2]
OSA_X pozice_Xv
<Osa_Xv.ActualPosition> 0.0
pismeno_x O_position[2,0] 7.0
OSA_Y pozice_Yv
<Osa_Yv.ActualPosition> 0.0
pismeno_y O_position[2,1] 0.0
Z00m ZOOM 10.0
OK OK_vst 0
AR OK_vyst 0
Vysl_osa_x O_position_1[2,0]
Vysl_osa_y O_position_1[2,1]
chyba PSNP_ready 0
8
EQU
Equal
Source A STEP_2 0
Source B 3
pom_STOP
OK_vyst
PSNP_ready
U
MOV
Move
Source 4
Dest STEP_2 0
9
EQU
Equal
Source A STEP_2 0
Source B 4
pom_STOP
Prepocet souradnic na prave
PSNP
Prepocet souradnic na prave
PSNP PSNP_2[3]
OSA_X pozice_Xv
<Osa_Xv.ActualPosition> 0.0
pismeno_x O_position[3,0] 9.0
OSA_Y pozice_Yv
<Osa_Yv.ActualPosition> 0.0
pismeno_y O_position[3,1] 2.5
Z00m ZOOM 10.0
OK OK_vst 0
AR OK_vyst 0
Vysl_osa_x O_position_1[3,0]
Vysl_osa_y O_position_1[3,1]
chyba PSNP_ready 0
10
EQU
Equal
Source A STEP_2 0
Source B 4
pom_STOP
OK_vyst
PSNP_ready
U
MOV
Move
Source 5
Dest STEP_2 0


11
EQU
Equal
Source A STEP_2 0
Source B 5
pom_STOP
Prepocet souradnic na prave
PSNP
Prepocet souradnic na prave
PSNP PSNP_2[4]
OSA_X pozice_Xv
<Osa_Xv.ActualPosition> 0.0
pismeno_x O_position[4,0] 7.0
OSA_Y pozice_Yv
<Osa_Yv.ActualPosition> 0.0
pismeno_y O_position[4,1] 5.0
Z00m ZOOM 10.0
OK OK_vst 0
AR OK_vyst 0
Vysl_osa_x O_position_1[4,0]
Vysl_osa_y O_position_1[4,1]
chyba PSNP_ready 0
12
EQU
Equal
Source A STEP_2 0
Source B 5
pom_STOP
OK_vyst
PSNP_ready
U
MOV
Move
Source 6
Dest STEP_2 0
13
EQU
Equal
Source A STEP_2 0
Source B 6
pom_STOP
Prepocet souradnic na prave
PSNP
Prepocet souradnic na prave
PSNP PSNP_2[5]
OSA_X pozice_Xv
<Osa_Xv.ActualPosition> 0.0
pismeno_x O_position[5,0] 2.0
OSA_Y pozice_Yv
<Osa_Yv.ActualPosition> 0.0
pismeno_y O_position[5,1] 5.0
Z00m ZOOM 10.0
OK OK_vst 0
AR OK_vyst 0
Vysl_osa_x O_position_1[5,0]
Vysl_osa_y O_position_1[5,1]
chyba PSNP_ready 0
14
EQU
Equal
Source A STEP_2 0
Source B 6
pom_STOP
OK_vyst
PSNP_ready
U
MOV
Move
Source 7
Dest STEP_2 0
15
EQU
Equal
Source A STEP_2 0
Source B 7
pom_STOP
Prepocet souradnic na prave
PSNP
Prepocet souradnic na prave
PSNP PSNP_2[6]
OSA_X pozice_Xv
<Osa_Xv.ActualPosition> 0.0
pismeno_x O_position[6,0] 0.0
OSA_Y pozice_Yv
<Osa_Yv.ActualPosition> 0.0
pismeno_y O_position[6,1] 2.5
Z00m ZOOM 10.0
OK OK_vst 0
AR OK_vyst 0
Vysl_osa_x O_position_1[6,0]
Vysl_osa_y O_position_1[6,1]
chyba PSNP_ready 0

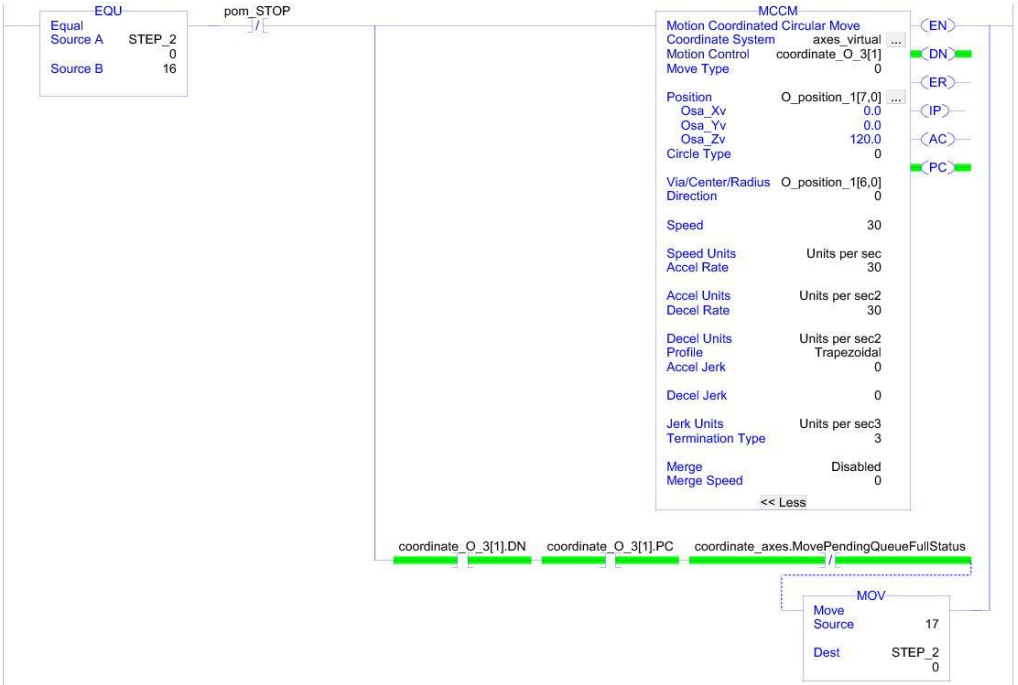
16
EQU
Equal
Source A STEP_2 0
Source B 7
pom_STOP
OK_vyst
PSNP_ready
U
MOV
Move
Source 8
Dest STEP_2 0
17
EQU
Equal
Source A STEP_2 0
Source B 8
pom_STOP
Prepocet souradnic na prave
PSNP
Prepocet souradnic na prave
PSNP PSNP_2[7]
OSA_X pozice_Xv <Osa_Xv.ActualPosition> 0.0
pismeno_x O_position[7,0] 2.0
OSA_Y pozice_Yv <Osa_Yv.ActualPosition> 0.0
pismeno_y O_position[7,1] 0.0
Z00m ZOOM 10.0
OK OK_vst 0
AR OK_vyst 0
Vysl_osa_x O_position_1[7,0]
Vysl_osa_y O_position_1[7,1]
chyba PSNP_ready 0
18
EQU
Equal
Source A STEP_2 0
Source B 8
pom_STOP
OK_vyst
PSNP_ready
U
MOV
Move
Source 9
Dest STEP_2 0
19
EQU
Equal
Source A STEP_2 0
Source B 9
pom_STOP
Prepocet souradnic na prave
PSNP
Prepocet souradnic na prave
PSNP PSNP_2[8]
OSA_X pozice_Xv <Osa_Xv.ActualPosition> 0.0
pismeno_x O_position[8,0] 2.0
OSA_Y pozice_Yv <Osa_Yv.ActualPosition> 0.0
pismeno_y O_position[8,1] 0.0
Z00m ZOOM 10.0
OK OK_vst 0
AR OK_vyst 0
Vysl_osa_x O_position_1[8,0]
Vysl_osa_y O_position_1[8,1]
chyba PSNP_ready 0
20
EQU
Equal
Source A STEP_2 0
Source B 9
pom_STOP
OK_vyst
PSNP_ready
U
MOV
Move
Source 10
Dest STEP_2 0


21
EQU
Equal
Source A STEP_2 0
Source B 10
pom_STOP
Prepocet souradnic na prave
PSNP
Prepocet souradnic na prave
PSNP PSNP_2[9]
OSA_X pozice_Xv
<Osa_Xv.ActualPosition> 0.0
pismeno_x O_position[9,0] 0.0
OSA_Y pozice_Yv
<Osa_Yv.ActualPosition> 0.0
pismeno_y O_position[9,1] 5.0
Z00m ZOOM 10.0
OK OK_vst 0
AR OK_vyst 0
Vysl_osa_x O_position_1[9,0]
Vysl_osa_y O_position_1[9,1]
chyba PSNP_ready 0
22
EQU
Equal
Source A STEP_2 0
Source B 10
pom_STOP
OK_vyst
PSNP_ready
MOV
Move
Source 11
Dest STEP_2 0
23
EQU
Equal
Source A STEP_2 0
Source B 11
pom_STOP
MCLM
Motion Coordinated Linear Move
Coordinate System axes_virtual
Motion Control coordinate_2[0]
Move Type 0
Position O_position_1[0,0]
Osa_Xv 0.0
Osa_Yv 0.0
Osa_Zv 100.0
Speed 100
Speed Units Units per sec
Accel Rate 100
Accel Units Units per sec2
Decel Rate 100
Decel Units Units per sec2
Profile Trapezoidal
Accel Jerk 100.0
Decel Jerk 100.0
Jerk Units % of Time
Termination Type 3
Merge 0
Merge Speed 0
<< Less
EN
DN
ER
IP
AC
PC
coordinate_2[0].DN
coordinate_2[0].PC
coordinate_axes.MovePendingQueueFullStatus
MOV
Move
Source 12
Dest STEP_2 0

24
EQU
Equal
Source A STEP_2 0
Source B 12
pom_STOP
MCLM
Motion Coordinated Linear Move
Coordinate System axes_virtual
Motion Control coordinate_2[1]
Move Type 0
Position O_position_1[1,0]
Osa_Xv 0.0
Osa_Yv 0.0
Osa_Zv 120.0
Speed 100
Speed Units Units per sec
Accel Rate 100
Accel Units Units per sec2
Decel Rate 100
Decel Units Units per sec2
Profile Trapezoidal
Accel Jerk 100.0
Decel Jerk 100.0
Jerk Units % of Time
Termination Type 3
Merge 0
Merge Speed 0
<< Less
EN
DN
ER
IP
AC
PC
coordinate_2[1].DN
coordinate_axes.MovePendingQueueFullStatus
MOV
Move
Source 13
Dest STEP_2 0
25
EQU
Equal
Source A STEP_2 0
Source B 13
pom_STOP
MCLM
Motion Coordinated Linear Move
Coordinate System axes_virtual
Motion Control coordinate_2[2]
Move Type 0
Position O_position_1[2,0]
Osa_Xv 0.0
Osa_Yv 0.0
Osa_Zv 120.0
Speed 100
Speed Units Units per sec
Accel Rate 100
Accel Units Units per sec2
Decel Rate 100
Decel Units Units per sec2
Profile Trapezoidal
Accel Jerk 100.0
Decel Jerk 100.0
Jerk Units % of Time
Termination Type 3
Merge 0
Merge Speed 0
<< Less
EN
DN
ER
IP
AC
PC
coordinate_2[2].PC
coordinate_2[2].DN
coordinate_axes.MovePendingQueueFullStatus
MOV
Move
Source 14
Dest STEP_2 0

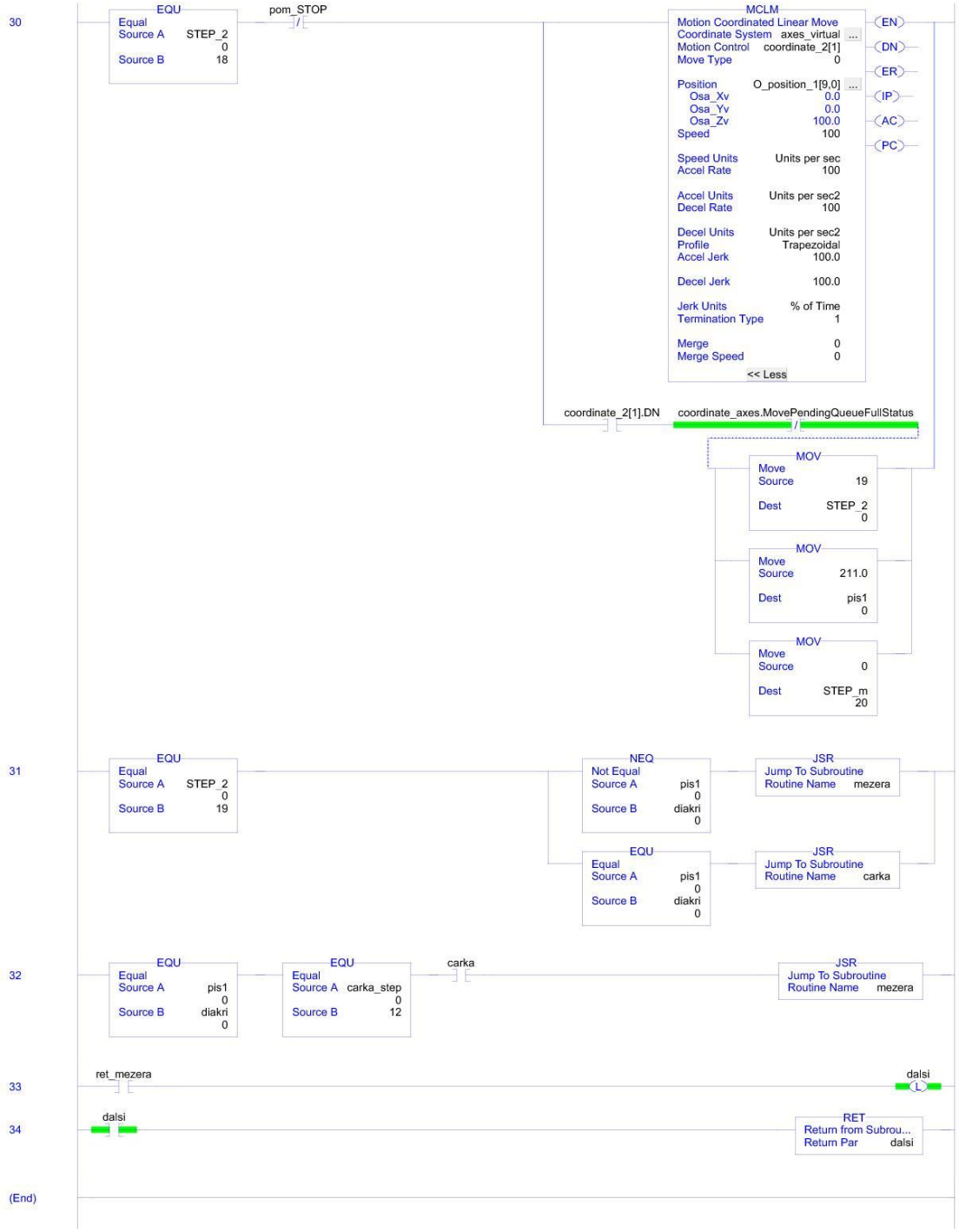

26
EQU
Equal
Source A STEP_2 0
Source B 14
pom_STOP
MCCM
Motion Coordinated Circular Move
Coordinate System axes_virtual
Motion Control coordinate_O_3[0]
Move Type 0
Position O_position_1[4,0]
Osa_Xv 0.0
Osa_Yv 0.0
Osa_Zv 120.0
Circle Type 0
Via/Center/Radius O_position_1[3,0]
Direction 0
Speed 30
Speed Units Units per sec
Accel Rate 30
Accel Units Units per sec2
Decel Rate 30
Decel Units Units per sec2
Profile Trapezoidal
Accel Jerk 0
Decel Jerk 0
Jerk Units Units per sec3
Termination Type 3
Merge Disabled
Merge Speed 0
<< Less
EN
DN
ER
IP
AC
PC
coordinate_O_3[0].DN
coordinate_axes.MovePendingQueueFullStatus
MOV
Move
Source 15
Dest STEP_2 0
27
EQU
Equal
Source A STEP_2 0
Source B 15
pom_STOP
MCLM
Motion Coordinated Linear Move
Coordinate System axes_virtual
Motion Control coordinate_2[4]
Move Type 0
Position O_position_1[5,0]
Osa_Xv 0.0
Osa_Yv 0.0
Osa_Zv 120.0
Speed 100
Speed Units Units per sec
Accel Rate 100
Accel Units Units per sec2
Decel Rate 100
Decel Units Units per sec2
Profile Trapezoidal
Accel Jerk 100.0
Decel Jerk 100.0
Jerk Units % of Time
Termination Type 3
Merge 0
Merge Speed 0
<< Less
EN
DN
ER
IP
AC
PC
coordinate_2[4].DN
coordinate_2[4].PC
coordinate_axes.MovePendingQueueFullStatus
MOV
Move
Source 16
Dest STEP_2 0

28


EQU
Equal
Source A STEP_2
0
Source B 16
pom_STOP
MCCM
Motion Coordinated Circular Move
Coordinate System axes_virtual
Motion Control coordinate_O_3[1]
Move Type 0
Position O_position_1[7,0]
Osa_Xv 0.0
Osa_Yv 0.0
Osa_Zv 120.0
Circle Type 0
Via/Center/Radius O_position_1[6,0]
Direction 0
Speed 30
Speed Units Units per sec
Accel Rate 30
Accel Units Units per sec2
Decel Rate 30
Decel Units Units per sec2
Profile Trapezoidal
Accel Jerk 0
Decel Jerk 0
Jerk Units Units per sec3
Termination Type 3
Merge Disabled
Merge Speed 0
<< Less
EN
DN
ER
IP
AC
PC
coordinate_O_3[1].DN
coordinate_O_3[1].PC
coordinate_axes.MovePendingQueueFullStatus
MOV
Move
Source 17
Dest STEP_2
0

29

EQU
Equal
Source A STEP_2
0
Source B 17
pom_STOP
MCLM
Motion Coordinated Linear Move
Coordinate System axes_virtual
Motion Control coordinate_2[1]
Move Type 0
Position O_position_1[8,0]
Osa_Xv 0.0
Osa_Yv 0.0
Osa_Zv 100.0
Speed 100
Speed Units Units per sec
Accel Rate 100
Accel Units Units per sec2
Decel Rate 100
Decel Units Units per sec2
Profile Trapezoidal
Accel Jerk 100.0
Decel Jerk 100.0
Jerk Units % of Time
Termination Type 1
Merge 0
Merge Speed 0
<< Less
EN
DN
ER
IP
AC
PC
coordinate_2[1].DN
coordinate_axes.MovePendingQueueFullStatus
MOV
Move
Source 18
Dest STEP_2
0


30
EQU
Equal
Source A STEP_2 0
Source B 18
pom_STOP
MCLM
Motion Coordinated Linear Move
Coordinate System axes_virtual
Motion Control coordinate_2[1]
Move Type 0
Position O_position_1[9,0]
Osa_Xv 0.0
Osa_Yv 0.0
Osa_Zv 100.0
Speed 100
Speed Units Units per sec
Accel Rate 100
Accel Units Units per sec2
Decel Rate 100
Decel Units Units per sec2
Profile Trapezoidal
Accel Jerk 100.0
Decel Jerk 100.0
Jerk Units % of Time
Termination Type 1
Merge 0
Merge Speed 0
<< Less
EN
DN
ER
IP
AC
PC
coordinate_2[1].DN
coordinate_axes.MovePendingQueueFullStatus
MOV
Move
Source 19
Dest STEP_2 0
MOV
Move
Source 211.0
Dest pis1 0
MOV
Move
Source 0
Dest STEP_m 20
31
EQU
Equal
Source A STEP_2 0
Source B 19
NEQ
Not Equal
Source A pis1 0
Source B diakri 0
JSR
Jump To Subroutine
Routine Name mezera
EQU
Equal
Source A pis1 0
Source B diakri 0
JSR
Jump To Subroutine
Routine Name carka
32
EQU
Equal
Source A pis1 0
Source B diakri 0
EQU
Equal
Source A carka_step 0
Source B 12
carka
JSR
Jump To Subroutine
Routine Name mezera
33
ret_mezera
dalsi
34
dalsi
RET
Return from Subrou...
Return Par dalsi
(End)