



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A ROBOTIKY

INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND ROBOTICS

VÝVOJ PROGRAMOVÉHO VYBAVENÍ PRO LABORATORNÍ MODEL CNC MANIPULÁTORU

DEVELOPMENT OF SOFTWARE FOR LABORATORY MODEL OF CNC MANIPULATOR

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JANA PIVOŇKOVÁ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. FRANTIŠEK BRADÁČ, Ph.D.

BRNO 2022

Zadání bakalářské práce

Ústav: Ústav výrobních strojů, systémů a robotiky
Studentka: **Jana Pivoňková**
Studijní program: Strojírenství
Studijní obor: Stavba strojů a zařízení
Vedoucí práce: **Ing. František Bradáč, Ph.D.**
Akademický rok: 2021/22

Ředitel ústavu Vám v souladu se zákonem č.1111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Vývoj programového vybavení pro laboratorní model CNC manipulátoru

Stručná charakteristika problematiky úkolu:

Úlohou práce je příprava aplikačního a řídicího softwaru pro laboratorní model CNC manipulátoru za využití vývojového prostředí EcoStruxure Machine Expert. Součástí řešení musí být i praktické odladění programu včetně konfigurace a nastavení pohonů.

Cíle bakalářské práce:

Popište současný stav vědy a techniky u řešené problematiky.

Proveďte systémový rozbor problematiky.

Navrhněte a zdůvodněte zvolený způsob řešení zadaného úkolu.

Navrhněte variantní řešení SW vybavení pro laboratorní model a zdůvodněte výběr varianty.

Proveďte zpracování vybraných modulů SW vybavené pro zvolenou variantu.

Definujte vlastní závěry a/nebo doporučení vzhledem k využití v praxi.

Seznam doporučené literatury:

Hlavní kategorie řady Modicon [online]. Praha: Schneider Electric, 2021 [cit. 2021-10-21].
Dostupné z: <https://www.se.com/cz/cs/work/products/master-ranges/modicon/>

BOLTON, W. Programmable logic controllers. Sixth edition. Amsterdam: Newnes, 2015. ISBN 978-0-12-802929-9.

BENEŠ, Pavel. Automatizace a automatizační technika: prostředky automatizační techniky. 5., rozš. a aktualiz. vyd. Brno: Computer Press, 2014. ISBN 978-80-251-3747-5.

MAREK, Jiří. Konstrukce CNC obráběcích strojů III. Praha: MM publishing, 2014. MM speciál. ISBN 978-802-6067-801.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2021/22

V Brně, dne

L. S.

doc. Ing. Petr Blecha, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

Tato bakalářská práce se zabývá tvorbou softwarového vybavení pro laboratorní model CNC manipulátoru. V rešeršní části je rozebrána problematika robotických manipulátorů a jejich řídicích systémů. Dále se zaměřuje na výběr jak konstrukčního, tak i softwarového řešení, která jsou dále rozpracována do finálního návrhu. Jako poslední se práce zabývá realizací výsledného řešení a jeho praktického odladění.

Summary

This bachelors thesis focuses on development of software for laboratory model of CNC manipulator. Research part discuss the problematics of robotic manipulators and their control systems. Furthermore, the thesis focuses on the selection of both design and software solutions, which are further developed into the final design. Lastly, the thesis deals with the implementation of the final solution and its practical debugging.

Klíčová slova

CNC manipulátor, Řídicí systém, CNC řízení, EcoStruxure Machine Expert

Keywords

CNC manipulator, Control system, CNC control, EcoStruxure Machine Expert

PIVOŇKOVÁ, J. *Vývoj programového vybavení pro laboratorní model CNC manipulátoru.*
Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2022. 45 s. Vedoucí
Ing. František Bradáč, Ph.D.

Čestné prohlášení

Prohlašuji, že tato práce je mým původním dílem, zpracovala jsem ji samostatně pod vedením Ing. Františka Bradáče, Ph.D. a s použitím literatury uvedené v seznamu.

V Brně dne 20. 5. 2022

Jana Pivoňková

Poděkování

Tímto bych ráda poděkovala Ing. Františku Bradáčovi, Ph.D. za cenné rady a pomoc během tvorby této práce. Dále bych chtěla poděkovat společnosti Schneider Electric, s.r.o., především Ing. Juriji Tomisovi, za poskytnuté rady a podporu. Poděkování patří také mojí rodině a blízkým, kteří mě během celého mého studia podporovali.

Obsah

1	Úvod	3
2	Současný stav vědy a techniky	4
2.1	Manipulátory	4
2.2	Kinematická struktura manipulátorů	5
2.2.1	Sériová	5
2.2.2	Paralelní	6
2.2.3	Kartézské manipulátory	6
2.3	Řídící systém	7
2.3.1	PLC	7
3	Systémový rozbor	8
3.1	S1 - Mechanická konfigurace	8
3.2	S2 - Softwarové řešení	8
3.3	Výběr hodnotících kritérií	9
4	Návrh variantních řešení zadaného problému	10
4.1	Navrhované mechanické konfigurace	10
4.1.1	Dvouosý kartézský manipulátor	10
4.1.2	Tříosý kartézský manipulátor	10
4.1.3	Tříosý delta robot	10
4.1.4	Výběr mechanické konfigurace	11
4.2	S2 - Navrhovaná softwarová řešení	11
4.2.1	CNC řízení	12
4.2.2	Výběr softwarového řešení	14
5	Tvorba řídicího systému a odladění mechaniky	16
5.1	Mechanická konfigurace	16
5.1.1	PLC	16
5.1.2	Osy manipulátoru MAXR12R	17
5.1.3	Operátorský panel	17
5.2	Řídící program	17
5.2.1	Použité rozšiřující knihovny	18
5.2.2	Struktura POU v programu	19
5.2.3	Operátorský panel	24
5.3	Odladění programu a mechaniky	27
5.3.1	Nastavení kontrolní smyčky pohonů	28
5.3.2	Odladění programu	28
6	Závěr	29
A	POU PowerON	31
B	POU PowerON - Initialization	33

OBSAH

C	POU PowerON - Home	34
D	POU PowerON - ManualMovement	35
E	POU PowerON - NCFile	36
F	POU PowerON - OutQueue	37
G	POU CNCTask	38
H	POU OutQueueMode	39
I	POU NCFileMode	41
J	POU Axis_DiagnosticMessages	43
K	Adresování	44
L	Vývojový diagram	45

1. Úvod

Průmyslová automatizace je rychle se rozvíjející odvětví, které má za cíl co nejvíce nahradit a ušetřit lidskou práci. Do tohoto odvětví spadají mimo jiné i manipulátory, které usnadňují manipulaci, ale také mají i další využití.

Úlohou této práce je příprava aplikačního a řídicího softwaru pro laboratorní model CNC manipulátoru, u kterého se předpokládá budoucí využití jako vykreslovací plotr. Jeho řízení bude zajišťovat PLC Modicon M262. Aby tento řídicí systém mohl být aplikován a mohla být ověřena jeho funkčnost, je třeba vybrat také mechanickou konfiguraci manipulátoru. Toto bude zajištěno pomocí vícekritériální analýzy, jež má za úkol porovnat navržené možnosti, a dle hodnocení vybrat nejoptimálnější řešení.

Následná tvorba softwaru proběhne pomocí vývojového prostředí EcoStruxure Machine Expert od společnosti Schneider Electric. Řešení se také bude věnovat praktickému odladění programu včetně konfigurace a nastavení pohonů. Dále práce zahrne i zhodnocení a doporučení pro budoucí vylepšení modelu.

2. Současný stav vědy a techniky

S postupně vyvíjejícími se technologiemi v různých odvětvích techniky vyvstala potřeba po strojích, které ulehčí práci při manipulační činnosti. Proto se v 60. letech 20. století začaly rozvíjet robotické manipulátory. Zařízení, které nahrazují funkce lidských rukou - úchop, držení a manipulaci. Postupným vývojem byly manipulátory doplněny senzory, které zajišťují zpětnou vazbu, což umožňuje jejich nezávislou operaci. [1]

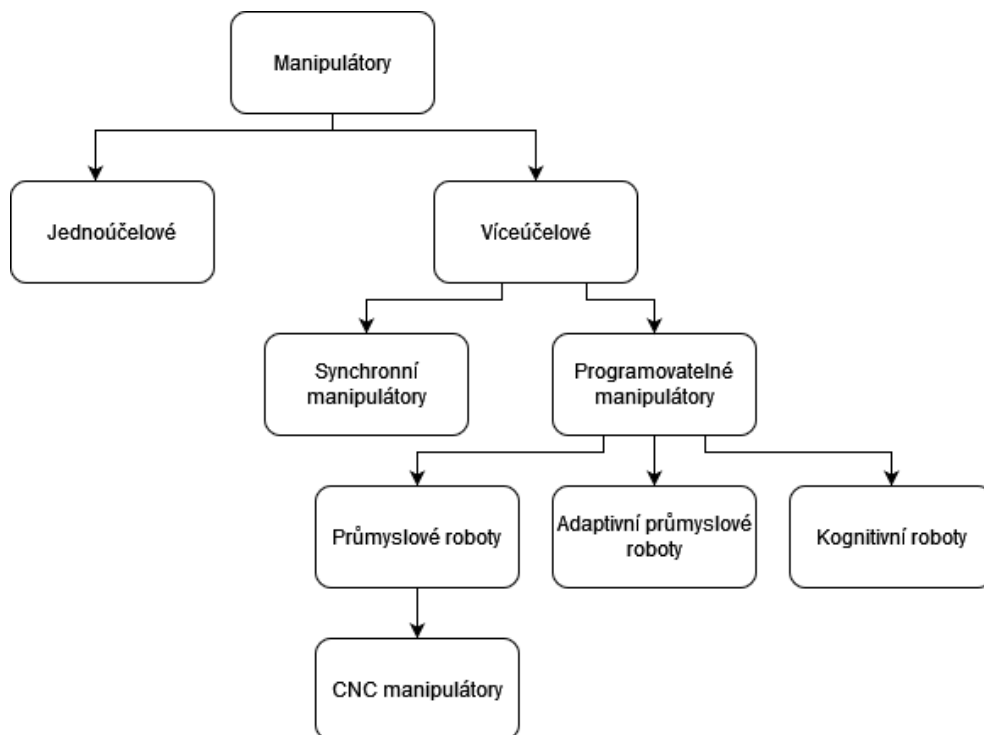
Kromě mechanické konfigurace je důležitou součástí manipulatorů i řídicí systém, který zpracovává uživatelské instrukce, informace ze sensorů a jednotlivých zařízení. [2]

2.1. Manipulátory

S postupným vývojem manipulatorů vznikalo i jejich rozdělení (obr. 2.1). Jedny z prvních manipulatorů byly jednoúčelové roboty, které sloužily pouze k jedné určité činnosti a nešlo jejich funkci měnit. Později se začaly objevovat víceúčelové manipulátory, které umožňují změnu jejich funkce. [3]

Synchronní manipulátory jsou ovládány pracovníkem, pro něž představují zesilovací ústrojí jeho schopností. Mohou být umístěny přímo na jeho těle nebo mohou být ovládány na dálku. Programovatelné manipulátory mají automatický řídicí systém a změna operačního programu je jednoduchá. [3]

Průmyslové roboty jsou manipulátory s pružným programem, zde spadají i CNC manipulátory, kterým se blíže věnuji v práci. Adaptivní průmyslové roboty mají schopnost reakce na změnu pracovní scény. Kognitivní roboty disponují jistou mírou umělé inteligence, díky čemuž stačí zadat cíl úkonu a robot si určí, jak tento úkon provede. [3, 4]



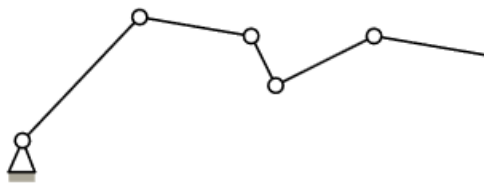
Obrázek 2.1: Dělení manipulatorů [3]

2.2. Kinematická struktura manipulátorů

Jedním z hlavních rozlišovacích prvků manipulátorů je kinematická struktura, která určuje operační prostor, ve kterém se může koncový efektor pohybovat. Skládá se z jednotlivých řetězců, jež tvoří členy a vazby mezi nimi (kinematické dvojice). [3, 4]

2.2.1. Sériová

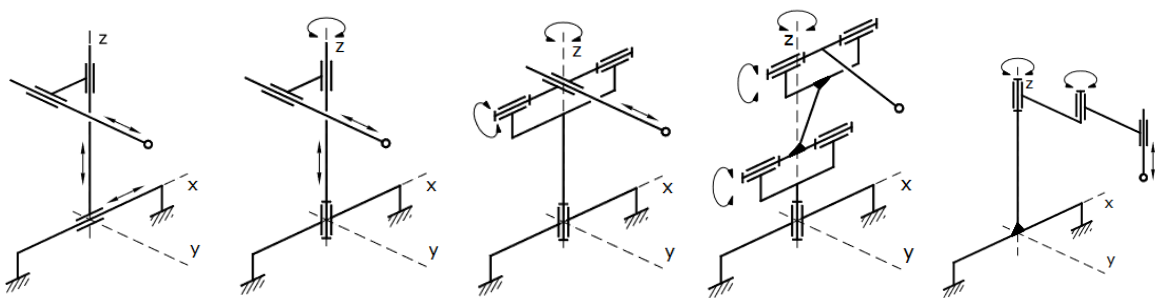
Tuto strukturu tvoří otevřený kinematický řetězec (obr. 2.2), který se skládá z ramen spojených posuvnými nebo rotačními vazbami. Z principu funkčnosti otevřeného řetězce vyplývá, že disponuje nižší tuhostí než uzavřený kin. řetězec. Tudíž nedokáže zajistit až takovou přesnost. Proto je sériová kinematika převážně využívána na manipulační úkony, kde je přínosem univerzálnost použití. [5]



Obrázek 2.2: Otevřený kinematický řetězec [4]

Tuto strukturu lze dělit na základě geometrie pracovního prostoru (obr. 2.3):

- Kartézská
- Cylindrická
- Sférická
- Angulární
- SCARA

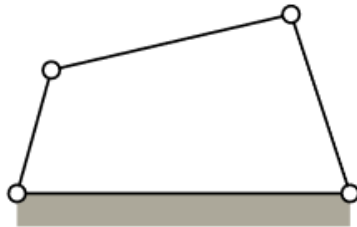


Obrázek 2.3: Geometrie pracovního prostoru sférické struktury - zleva Kartézská, Cylindrická, Sférická, Angulární, SCARA [4]

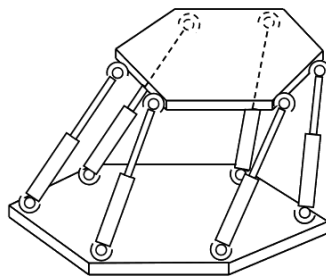
2.2. KINEMATICKÁ STRUKTURA MANIPULÁTORŮ

2.2.2. Paralelní

Tato struktura je tvořena uzavřeným kinematickým řetězcem (obr. 2.4), který může reprezentovat pohyblivá příhradová konstrukce nebo pohyblivá základna podepíraná několika paralelními rameny (obr. 2.5). Díky umístění pohonů přímo na rámu stroje, má struktura vyšší tuhost, z čehož vyplývá, že má vyšší přesnost polohování. [5]



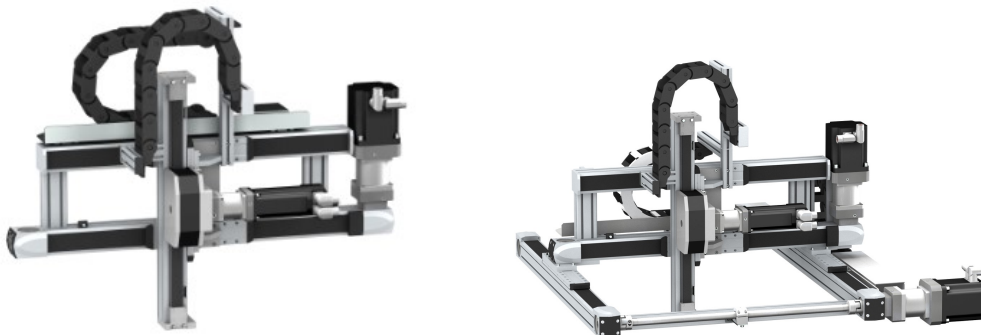
Obrázek 2.4: Uzavřený kinematický řetězec [4]



Obrázek 2.5: Paralelní kinematická struktura [5]

2.2.3. Kartézské manipulátory

Jelikož je dále v práci využít tento manipulátor, je třeba informace více rozvinout. Tyto manipulátory mohou být jak dvouosé, tak i tříosé (obr. 2.6), což rozhoduje o jejich operačním rozsahu. Geometrie pracovního prostoru dvouosého manipulátoru má tvar roviny, která je určena osami XY, XZ nebo YZ, v závislosti na dané aplikaci. Geometrie pracovního prostoru tříosého manipulátoru má tvar hranolu a je určena třemi vzájemně kolmými osami XYZ. Vzhledem ke konstrukci lineárních pohybových ústrojí jsou kartézské manipulátory většinou těžší v porovnání s ostatními zařízeními se sériovou kin. strukturou, což umožňuje vyšší tuhost a z toho vyplývající přesnost polohování. Konstrukce těchto manipulátorů umožňuje i větší pracovní rozsahy nebo využití stavebnicové konstrukce. [6]



Obrázek 2.6: Kartézské manipulátory - zleva dvouosý, tříosý [7]

2.3. Řídící systém

Chod manipulátorů je ovládán pomocí řídicího systému, který může být buď řídicí počítač nebo programovatelný automat (PLC). Liší se tím, že řídicí počítače využívají univerzální výpočetní prostředky, zatímco PLC jsou speciální systémy určené pro průmyslové použití za účelem řízení strojů. [2, 8]

2.3.1. PLC

Pro svůj chod využívá programovatelnou paměť, do které ukládá uživatelské interní instrukce, které ovládají analogové a digitální vstupy a výstupy, pomocí nichž řídí stroj. Běh PLC probíhá v periodicky se opakujícím cyklu - scanning period. Kdy na začátku si načte hodnoty jednotlivých vstupů, ty uloží do paměti. Poté provede program, během něhož vyhodnocuje jednotlivé podmínky, z kterých se vypisují výstupní hodnoty, které jsou po ukončení jednoho cyklu přes výstupy komunikovány dále. [2, 8]

PLC lze z konstrukčního hlediska dělit na modulární a kompaktní. Modulární PLC se skládá z nosného rámu a základní jednotky, k nimž jsou připojeny rozšiřující moduly v závislosti na potřebách daného stroje. Modulární provedení umožňuje variabilně měnit konfiguraci PLC v závislosti na rozsahu řešeného problému. Dále také na postupném vývoji a rozšiřování možností stroje. Kompaktní PLC mají pevně danou konfiguraci vstupů a výstupů, která se většinou používá při řízení menších systémů. [2]

Řídící program PLC je tvořený pomocí vývojového prostředí, jenž umožňuje tvorbu v textových nebo grafických programovacích jazycích dle mezinárodního standardu IEC 61131-3. [8]

3. Systémový rozbor

V současné době můžeme najít nepřehledné množství technických řešení manipulátorů, proto je třeba se během tvorby této práce zaměřit i na volbu variantního řešení, díky kterému lze získat co nejvýhodnější výsledek. Pro výběr je použito několik hodnotících kritérií (systém podstatných veličin), kterým se přiřadí váha dle jejich důležitosti. Tento systém bude rozdělen na dvě skupiny - mechanickou (S1) a softwarovou (S2), jelikož parametry softwarového řešení odvíjející se od mechanické konfigurace neovlivní výsledek natolik, aby patřily mezi hodnotící kritéria, ale ve výsledném návrhu se promítnou.

Úkolem této práce je návrh a tvorba softwarového vybavení pro CNC manipulátor, což je značně široké zadání a vyvstalo by velké množství variantních řešení. Jelikož je tato práce tvořena ve spolupráci se společností Schneider Electric (SE), bylo zadání doplněno tak, aby šlo o CNC řízený manipulátor, který má předpoklad budoucího využití jako plotr pro vykreslování. Na jeho sestavení má být využito produktů společnosti.

3.1. S1 - Mechanická konfigurace

Při výběru hodnotících kritérií mechanické konfigurace ze zadání vyplynula čtyři kritéria, která nejlépe definují výběr výsledné varianty (tab. 3.1).

- **K1 - Cena manipulátoru** - Vzhledem k tomu, že jde o návrh laboratorního modelu manipulátoru, je žádoucí držet pořizovací náklady co nejnižší
- **K2 - Přesnost pohybu** - Z předpokladu využití manipulátoru pro funkci plotru, vyplývá, že je třeba do určité míry brát ohled na přesnost
- **K3 - Pracovní rozsah** - Jak velký prostor zvládne manipulátor pokrýt
- **K4 - Zástavbový prostor** - Kolik prostoru zabírá konstrukce manipulátoru vzhledem k pracovnímu rozsahu

3.2. S2 - Softwarové řešení

Vzhledem k principu funkce řídicího programu (kap. 4.2.1) je třeba zvolit pouze kompilační mód, kterým bude zpracován G-kód, zbytek programu je pro všechny tři režimy identický, díky čemuž vyvstala pouze dvě rozhodovací kritéria (tab. 3.1).

- **K5 - Možnost jednoduché aplikace libovolného kódu bez nutnosti změny řídicího programu** - Pro zjednodušení používání manipulátoru a urychlení práce je žádoucí umožnit změnu G-kódu, aniž by bylo třeba měnit ovládací program
- **K6 - Možnost použití G-kódu bez předzpracování** - Pro aplikace vyžadující přesné zachování původního G-kódu bez úprav např. dynamiky nebo cesty, je nutné zohlednit, jakým způsobem fungují jednotlivé kompilační módy

3.3. Výběr hodnotících kritérií

Pro určení samotných kritérií a jejich vah existuje několik metod, které jsou velice nápomocné. V tomto případě je však postačující jejich určení dle priority na základě konzultace s odborníky z oboru.

Skupina	Kritérium	Název	Váha
S1	K1	Cena manipulátoru	0,5
	K2	Přesnost pohybu	0,1
	K3	Pracovní rozsah	0,2
	K4	Zástavbový prostor	0,2
S2	K5	Možnost jednoduché aplikace libovolného kódu bez nutnosti změny řídicího programu	0,6
	K6	Možnost použití G-kódu bez předzpracování	0,4

Tabulka 3.1: Systém podstatných veličin

4. Návrh variantních řešení zadaného problému

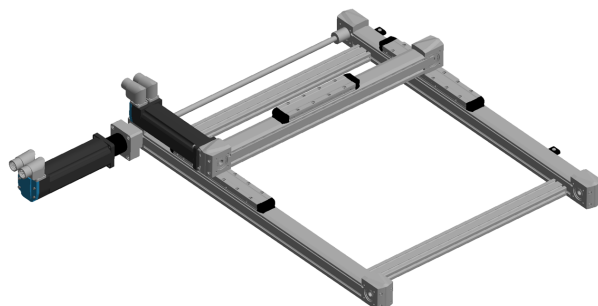
Dle zadání (kap. 3) je třeba určit několik variantních řešení manipulátoru. I nadále budou rozděleny návrhy variantních řešení na mechanickou a softwarovou část pro jejich intuitivnější výběr.

Pro přiřazení hodnot jednotlivých kritérií je využito klasifikace 1 - 3, kdy 1 značí neoptimálnější volbu a 3 volbu nejméně optimální vzhledem k definici daného kritéria. Poté tyto hodnoty budou porovnány napříč řešeními s ohledem na váhy a tím se získá nejvíce vyhovující řešení.

4.1. Navrhované mechanické konfigurace

4.1.1. Dvouosý kartézský manipulátor

První z nabízejících se řešení je využití dvouosého kartézského manipulátoru MAXR12, MAXR22, MAXR32 nebo MAXR42. Tento manipulátor umožňuje velký pracovní rozsah při zachování přesnosti polohování. U tohoto řešení je třeba brát v úvahu nutnost připojení koncového efektoru, který zajistí držení a přítlak pera (např. pomocí elektromagnetu). To z důvodu dodržení předpokladu jeho využití pro funkci plotru. Tento koncový efektor může mít negativní vliv na přesnost polohování. [7]



Obrázek 4.1: Dvouosý kartézský manipulátor MAXR12R [7]

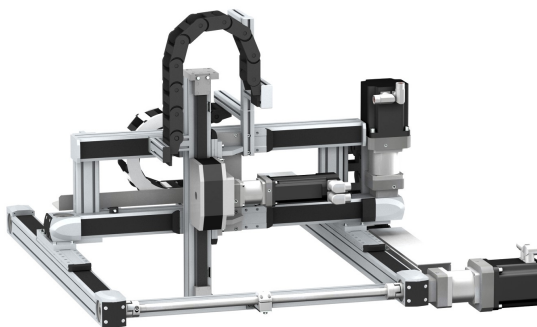
4.1.2. Tříosý kartézský manipulátor

Další z nabízených řešení je využití tříosého kartézského manipulátoru MAXR13, MAXR23, MAXR33 nebo MAXR43, kde oproti první variantě přibude navíc osa Z zajišťující přítlak per, z čehož vyplývá, že připojený koncový efektor bude zajišťovat pouze držení pera. Toto umožní velký pracovní rozsah a zachování přesnosti polohování. [7]

4.1.3. Tříosý delta robot

Narozdíl od prvních dvou řešení, tento návrh využívá tříosý delta robot Lexium P Delta 3, který má díky využití paralelní kinematické struktury vysokou přesnost PTP (Point to Point) pohybů. Pro lineární pohyby tato přesnost tak vysoká není. Delta robot operuje v malém pracovním rozsahu s vysokým zástavbovým prostorem. [7]

4. NÁVRH VARIANTNÍCH ŘEŠENÍ ZADANÉHO PROBLÉMU



Obrázek 4.2: Tříosý kartézský manipulátor [7]



Obrázek 4.3: Tříosý delta robot [7]

4.1.4. Výběr mechanické konfigurace

Kritérium (Váha)	K1 (0,5)	K2 (0,1)	K3 (0,2)	K4 (0,2)	Výsledná hodnota
Dvouosý kartézský manipulátor	1	2	1	1	1,1
Tříosý kartézský manipulátor	2	1	1	2	1,7
Tříosý delta robot	3	3	2	3	2,8

Tabulka 4.1: S1 - Porovnání návrhových řešení mechanické konfigurace

Z výsledků porovnání jednotlivých variantních řešení (tab. 4.1) vyplývá, že pro zadání této práce je nejvíce vyhovující dvouosý kartézský manipulátor (kap. 4.1.1), který je zvolen pro další vypracování.

4.2. S2 - Navrhovaná softwarová řešení

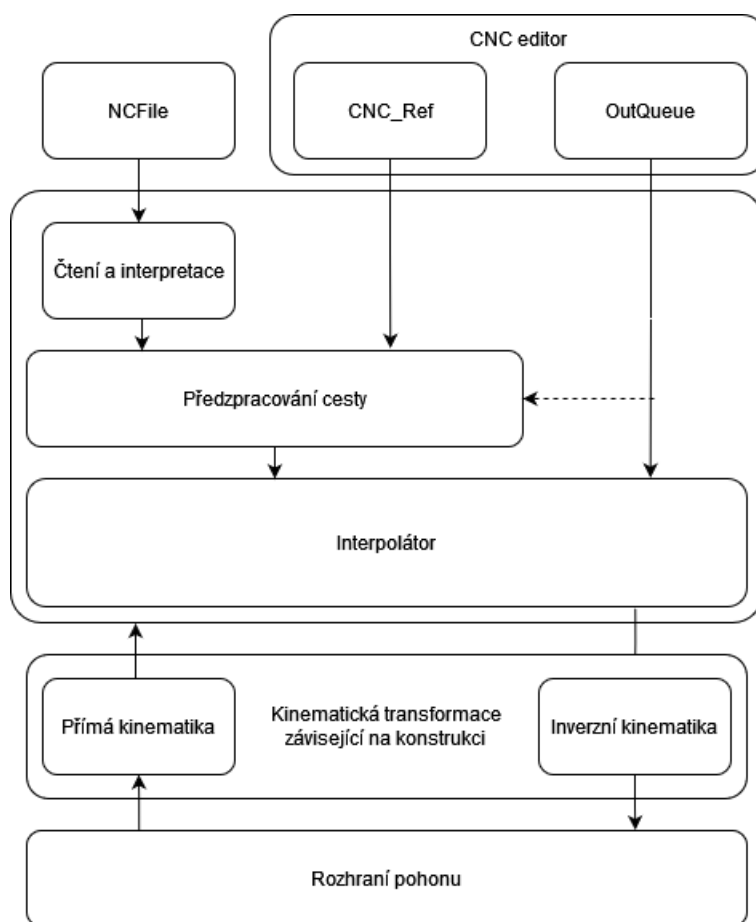
Před výběrem softwarového řešení je třeba rozebrat, jak CNC řízení probíhá konkrétně u zvoleného PLC a také jaké jsou možnosti vývojového prostředí v tomto ohledu.

4.2. S2 - NAVRHOVANÁ SOFTWAREVÁ ŘEŠENÍ

4.2.1. CNC řízení

Tvorba řídicího programu probíhá pomocí softwarového nástroje CODESYS SoftMotion, který rozšiřuje možnosti vývojového prostředí pro řízení CNC (Computer Numerical Control), PTP (Point to Point) a CP (Controlled Path) pohybů. Samotné CNC řízení probíhá pomocí funkčních bloků doplňkových knihoven, které umožňují manipulaci s G-kódem, jeho zpracování a komunikaci s pohony. Principiální schéma CNC řízení je zobrazeno na obr. 4.4. [9]

Řídicí G-kód je možné vytvářet v CNC editoru před nahráním řídicí aplikace do PLC nebo je možné jej do PLC nahrát jako soubor ASCII. Další zpracování G-kódu probíhá pomocí tří kompilačních módů - NCFfile, CNC-Ref a OutQueue. Získané informace jsou poté v PLC zpracovány a je z nich vytvořena interpolovaná cesta, z níž jsou kinematickou transformací určeny žádané polohy pro jednotlivé pohony. Ty jsou následně přeneseny do jejich rozhraní. [9]



Obrázek 4.4: Princip CNC řízení [9]

Kompilační módy

NCFfile - Tento mód zpracovává program ze souboru vloženého do kontroléru, který je čtený a zpracovaný krok po kroku. Je ideální pro velké programy, jež nemůžou být celé uloženy v paměti PLC a nebo programy, které byly vytvořeny po kompilaci řídicí aplikace. Pro načtení programu v PLC jsou použity funkční bloky pro čtení a interpretaci G-

4. NÁVRH VARIANTNÍCH ŘEŠENÍ ZADANÉHO PROBLÉMU

-kódu, čímž se získá datová struktura, která je dále upravena pomocí funkčních bloků pro předzpracování cesty a vložena do interpolátoru. [9]

CNC-Ref - Řídící G-kód je vytvořen v CNC editoru, po kompilaci programu jsou na datovou strukturu použity funkční bloky pro předzpracování cesty a následně je struktura vložena do interpolátoru. V tomto módu lze v programu používat proměnné, jejichž hodnotu lze změnit během restartu dekodéru. [9]

OutQueue - Stejně jako CNC-Ref i v tomto módu je G-kód vytvářen v CNC editoru. Vzniklou datovou strukturu je možné vložit buď přímo do interpolátoru nebo je možné použít i funkční bloky pro předzpracování cesty. [9]

Zpracování cesty

Po kompilaci G-kódu vzniká datová struktura, která je v PLC dále zpracovávána pomocí funkčních bloků pro předzpracování cesty. Mezi tyto funkční bloky patří např. [10]:

- Check velocities - Monitoruje rychlosti v ostrých ohybech cesty a zde je omezuje
- Limit Dynamics - Omezuje rychlosti a zrychlení dle maximálních zadaných hodnot
- Avoid Loop - Zkontroluje cestu a pokud se na ní nachází nějaké křížení se smyčkami, je cesta v překřížení oříznuta a smyčka odstraněna

Takto upravená datová struktura je dále vkládána do interpolátoru, kde se z ní tvoří cesta a zároveň se zohledňují výrazy G-kódu určující rychlosti, zrychlení a další doplňující informace. Interpolátor pracuje v cyklech, na konci každého z nich je vygenerována žádaná pozice a informace o žádané rychlosti. [10]

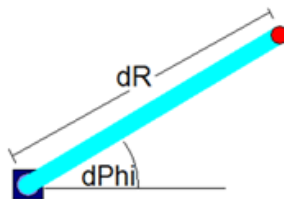
Kinematická transformace

Žádaná pozice z interpolátoru je pomocí kinematické transformace upravena, aby každá osa měla svoji hodnotu zvlášť. Tato transformace probíhá pomocí funkčního bloku Trafo, který principiálně pracuje stejně, ale liší se v závislosti na dané mechanické konfiguraci a způsobu výpočtu polohy. Tento výpočet může probíhat dvěma způsoby [10]:

- Přímá transformace - Poloha v prostoru je určena z natočení a polohy os
- Inverzní transformace - Natočení a polohy os jsou určeny z polohy v prostoru

Možné mechanické konfigurace dle kinematické struktury jsou [10]:

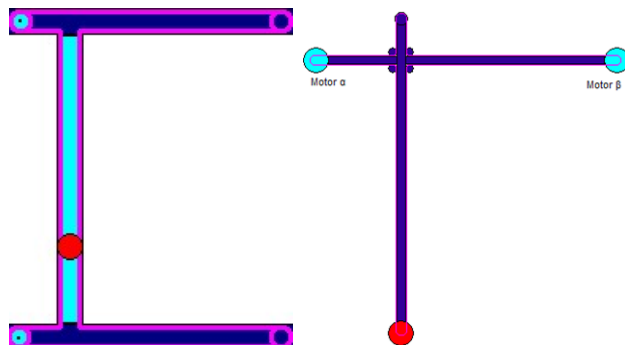
- Polární systém - Skládá se z otočné a lineární osy, kdy otočná osa udává směr a lineární osa vzdálenost (obr. 4.5)



Obrázek 4.5: Polární systém [9]

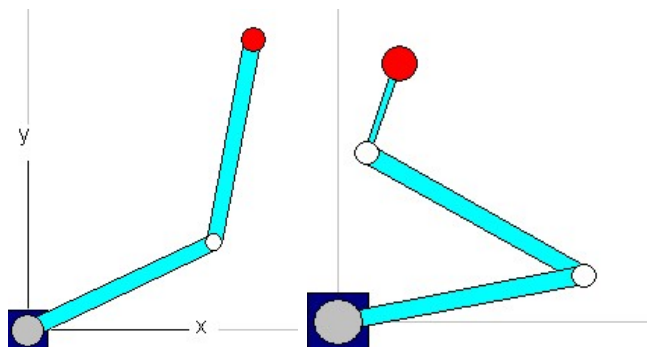
4.2. S2 - NAVRHOVANÁ SOFTWAREVÁ ŘEŠENÍ

- Portálový robot - Jsou to dvouosé nebo tříosé systémy, kde osy mezi sebou svírají pravé úhly, možné variace jsou s otočnou osou, s odsazením nástroje, tvaru H a tvaru T (obr. 4.6)



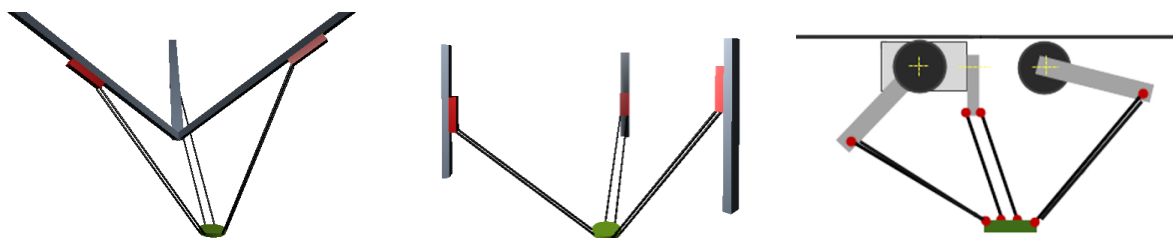
Obrázek 4.6: Portálové roboty [9]

- SCARA roboty - Dvoj nebo tříramenné roboty, které svým pohybem napodobují lidskou paži (obr. 4.7)



Obrázek 4.7: SCARA roboty [9]

- Paralelní roboty - Skládají se ze tří os (lineárních orientovaných pod úhlem, lineárních orientovaných vertikálně, otočných) spojených dle přesně definované mechaniky, umožňující pohyb v prostoru (obr. 4.8)



Obrázek 4.8: Paralelní roboty [9]

4.2.2. Výběr softwarového řešení

Z informací o průběhu CNC řízení lze určit, že pro výběr softwarového řešení je třeba vybrat jeden ze tří kompilačních módů, které udávají jakým způsobem je G-kód zpracován.

4. NÁVRH VARIANTNÍCH ŘEŠENÍ ZADANÉHO PROBLÉMU

Kritérium (Váha)	K5 (0,6)	K6 (0,4)	Výsledná hodnota
OutQueue	2	1	1,6
CNC_Ref	2	2	2
NCFfile	1	3	1,8

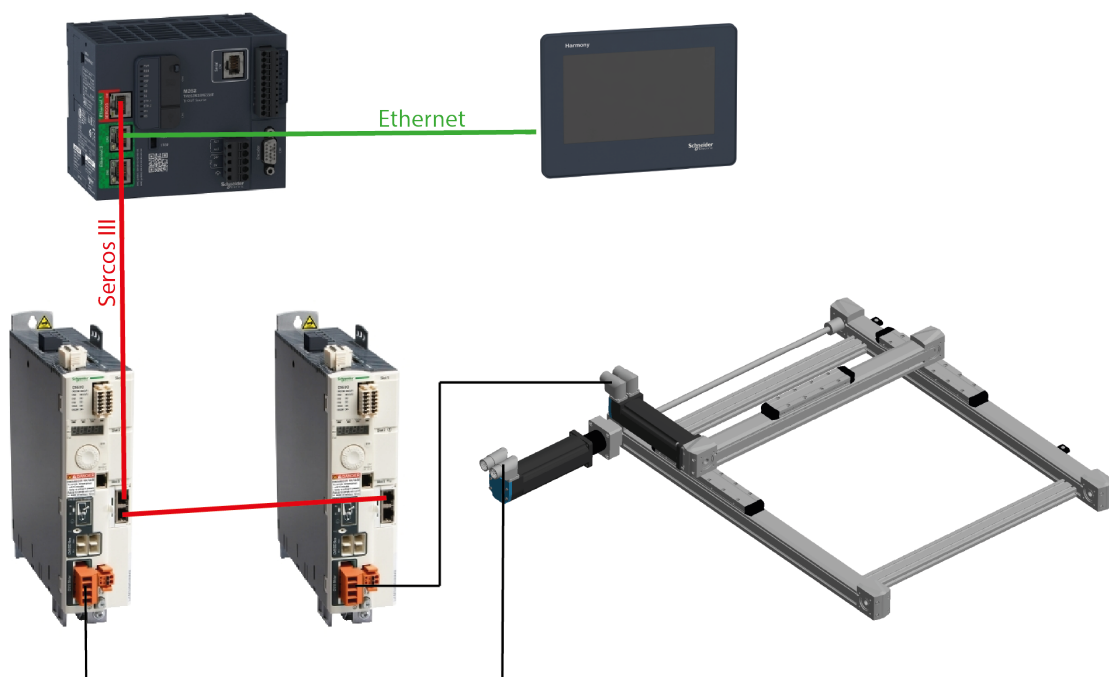
Tabulka 4.2: Porovnání návrhových softwarových řešení

Ačkoliv z výsledků porovnání jednotlivých variantních řešení (tab. 4.2) vyplývá, že pro zadání této práce je nejvíce vyhovující kompilační mód OutQueue (kap. 4.2.1). Nesplňuje však naplno kritérium K5 - Možnost jednoduché aplikace libovolného kódu bez nutnosti změny řídicího programu. Pro zavedení nového programu je třeba jej vytvořit ve vývojovém prostředí pomocí CNC editoru. Z toho důvodu jsou vybrány kompilační módy dva - OutQueue a NCFfile (kap. 4.2.1 a 4.2.1), které se liší natolik, aby bylo jejich využití opodstatněné.

5. Tvorba řídicího systému a odladění mechaniky

5.1. Mechanická konfigurace

Zvolená mechanická konfigurace se skládá z několika zařízení (obr. 5.1). Základ tvoří PLC M262, které ovládá dva servoměniče Lexium 32S pomocí komunikačního rozhraní Sercos III. K servoměničům je připojen manipulátor MAXR12R, jehož osy jsou poháněny servomotory BSH. PLC je doplněno operátorským panelem Magelis, kde fyzické rozhraní komunikace tvoří Ethernet. Komunikace probíhá pomocí protokolu Modbus TCP/IP.



Obrázek 5.1: Mechanická konfigurace

5.1.1. PLC

Modicon M262M35MESS8T

O řízení manipulátoru se stará PLC M262, které je uzpůsobeno pro realizaci synchronního pohybu, což je zajištěno pomocí implementace komunikačního rozhraní Sercos III přímo do kontroléru. Toto PLC má také zabudované protokoly pro použití průmyslového internetu věcí (Industrial Internet of Things - IIoT), který dává prostor pro budoucí rozvoj tohoto laboratorního modelu. [7]

5.1.2. Osy manipulátoru MAXR12R

Lineární osy - PAS41BR0300A pro osu X a PAS41BR0350A pro osu Y

Jak osa X, tak i osy Y disponují stejnými parametry. Hlavním rozdílem je, že osa Y je nosná pro osu X, proto je využito zdvojených os. Ty jsou vzájemně spojeny pomocí synchronizační hřídele. Dalším rozdílem je pracovní délka os, která je v případě osy X 300 mm a v případě os Y je 350 mm. [7]

Osy tvoří profil o rozměru 40 x 40 mm (velikost 1). Převod rotačního pohybu na lineární je realizován pomocí ozubeného řemenu napnutého na řemenicích umístěných v koncových blocích. Vedení je zajištěno pomocí opěrných kladek. [7] Pro vymezení referenční polohy os jsou použity indukční koncové snímače. Ty jsou připojeny pomocí pozitivní logiky (PNP kontakty) s vyhodnocením signálu jako normálně uzavřené kontakty. [12]

Servomotory - BSH0551T01F2A pro osu X a BSH0553T01F2A pro osu Y

Osa X je poháněna synchronním servomotorem BSH0551T01F2A, jehož polohovou zpětnou vazbu zajišťuje jednotáčkový enkodér. Servomotor je vybaven brzdou pro zamezení pohybu při odpojení motoru od přívodu elektrického proudu. Motor je schopný vyvinout kroutící moment o velikosti $M_o = 0,5$ Nm. [7]

Osa Y je poháněna synchronním servomotorem BSH0553T01F2A, jenž stejně jako motor osy X disponuje jednotáčkovým enkodérem a brzdou. Tento motor je schopný vyvinout kroutící moment o velikosti $M_o = 1,2$ Nm. [7]

Servoměniče - Lexium LXM32SD18M2

Oba servomotory jsou řízeny pomocí servoměničů Lexium 32S, které jsou uzpůsobeny pro použití komunikace pomocí protokolu Sercos III. [12]

5.1.3. Operátorský panel

HMIS5T

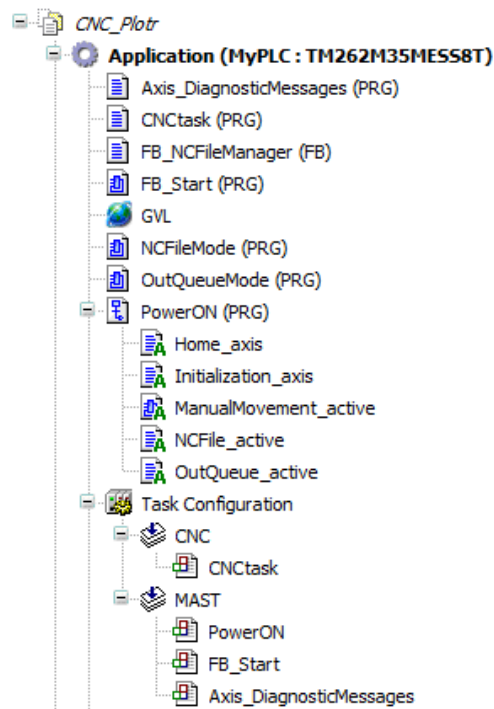
Pro pohodlné ovládání manipulátoru je využit operátorský panel HMIS5T, jež s kontrolérem komunikuje pomocí protokolu Modbus TCP/IP. [7]

5.2. Řídící program

Dle návrhového řešení bylo určeno, že řídicí program využije dvou kompilačních módů pro CNC řízení. Dále bylo rozhodnuto doplnit tyto módy o manuální ovládání pohybu, kterým docílíme ručního polohování.

Běh programu je rozdělen do dvou cyklických činností - tasků (obr. 5.2). První z nich je task MAST s dobou trvání 10 ms, ve kterém běží POU zajišťující chod programu - PowerON a FB_Start. Pro CNC řízení je separátní task CNC, který trvá 1 ms a zajišťuje chod interpolátoru pro zvolený ovládací mód. Samotný program je řízen dle vývojového diagramu (příloha L).

5.2. ŘÍDÍCÍ PROGRAM



Obrázek 5.2: Rozdělení aplikace a činnosti PLC

5.2.1. Použité rozšiřující knihovny

SM3_CNC

Tato knihovna umožňuje číst, interpretovat a interpolovat G-kód. [10]

CNCExtension

Tato knihovna je určena pro použití specifických funkcí z knihovny SM3_CNC s kontrolérem M262. [10]

CAA File

Tato knihovna slouží pro přístup k souborům a manipulaci s nimi. [10]

SercosMaster

Knihovna poskytující enumerace, struktury a funkce pro ovládání a monitorování komunikace Sercos na kontroléru M262. Knihovna umožňuje číst a psát parametry komunikace, resetovat chybové hlášky a získat informace o jejím nastavení. [10]

V programu je využita v POU PowerON.Initialization, kde zajišťuje inicializaci komunikace Sercos do Phase 4, což je plně funkční komunikace. [10]

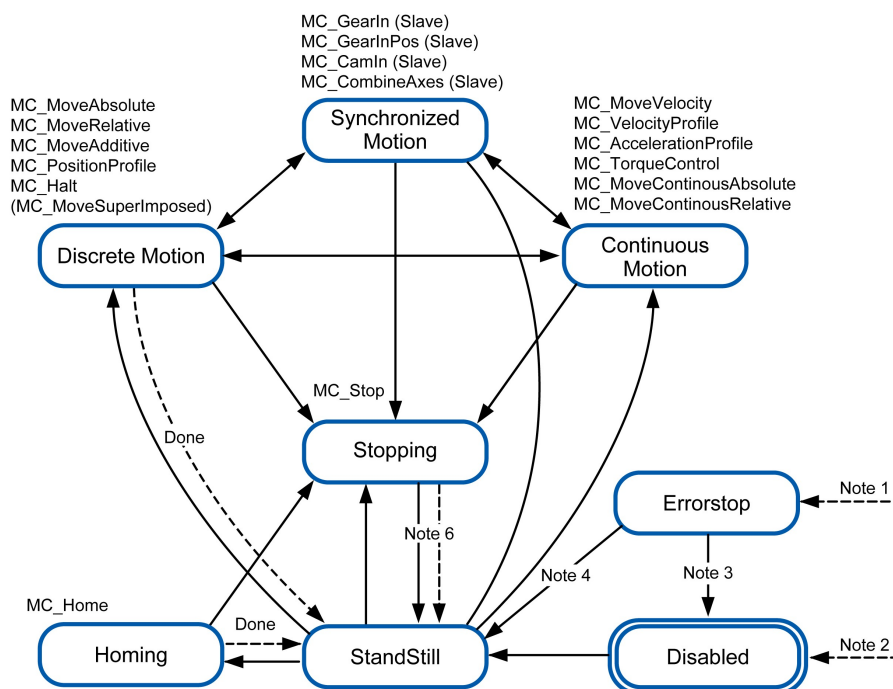
MotionInterface

Tato knihovna se skládá z datových typů a enumerací reference osy IF_Axis, což slouží jako vstup funkčních bloků z knihovny PLC Open MC. [10]

PLCopen MC

Nezávislá knihovna, která obsahuje funkční bloky pro pohyb jednotlivých os a jejich správu. Tuto knihovnu je možné nahradit knihovnou Sercos3PlcOpenMc. Ta má na rozdíl od PLCopen MC rozšířené diagnostické možnosti a je přímo uzpůsobená pro použití se Sercos. [10]

Výsledkem použití FB této knihovny je vždy jeden ze stavů osy v diagramu na obr. 5.3, kde jsou znázorněny i možné přechody mezi jednotlivými stavy. V jedné chvíli se může osa nacházet pouze v jednom z těchto stavů. [10]



Note 1 An error has been detected. (Transition from any state).

Note 2 The input Enable of the function block MC_Power is set to FALSE and no error has been detected (transition from any state).

Note 3 MC_Reset and MC_Power.Status = FALSE.

Note 4 MC_Reset and MC_Power.Status = TRUE and MC_Power.Enable = TRUE.

Note 5 MC_Power.Enable = TRUE and MC_Power.Status = TRUE.

Note 6 MC_Stop.Done = TRUE and MC_Stop.Execute = FALSE.

Obrázek 5.3: Stavový diagram osy dle knihovny PLCopen MC [9]

5.2.2. Struktura POU v programu

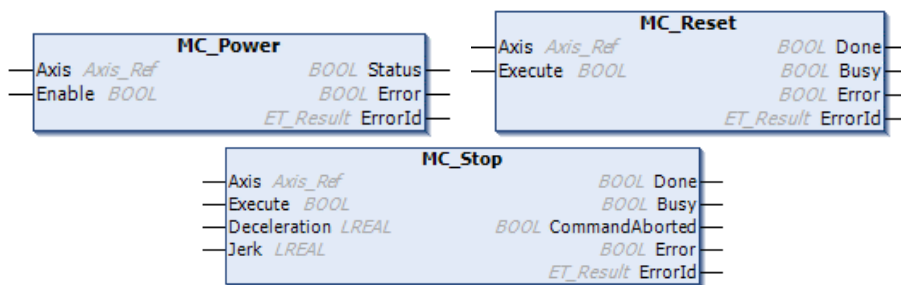
FB_Start

Toto POU cyklicky provádí funkční bloky pro ovládání os a zajišťuje čtení a zápis jejich aktuální polohy do proměnných.

Použité funkční bloky MC_Power, MC_Reset a MC_Stop (obr. 5.4) jsou z knihovny PLCopen MC a zajišťují základní ovládání os X a Y. Dále je zde použita i konverzní

5.2. ŘÍDÍCÍ PROGRAM

funkce to REAL pro hodnotu aktuální polohy jednotlivých os, jelikož operátorský panel nepracuje s datovým typem LREAL.



Obrázek 5.4: Funkční bloky MC_Power, MC_Reset a MC_Stop [10]

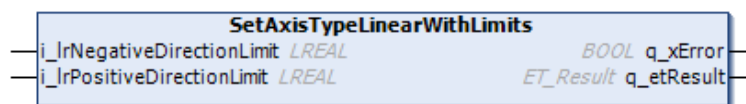
PowerON

POU PowerON (příloha A) je základem celého programu, zajišťuje inicializaci komunikace, nájezd os do referenční polohy, výběr operačního módu a jeho následné provedení. Skládá se z několika kroků, které znázorňuje vývojový diagram (příloha L).

PowerON - Initialization

V tomto kroku (příloha B) je zajištěna inicializace komunikace Sercos z Phase 0 do Phase 4, což značí plně funkční komunikaci.

Je zde použitý funkční blok SetAxisTypeLinearWithLimits (obr. 5.5) z knihovny MotionInterface, který je využíván pro inicializaci os. V tomto konkrétním případě pro nastavení limitů lineárních os. Neboť je u každé osy použitý pouze jeden koncový snímač v negativním směru pohybu vymezující referenční polohu.

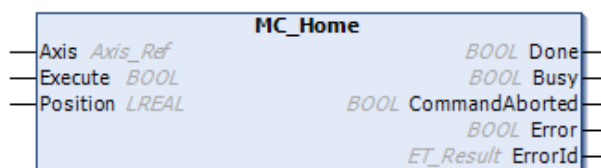


Obrázek 5.5: Funkční blok SetAxisType [10]

PowerON - Home

Tento krok (příloha C) zajišťuje nájezd do referenční polohy, pro její správné vymezení.

Využívá funkční blok MC_Home (obr. 5.6) zprostředkující nájezd os do referenční polohy, čímž zajistí stejnou polohu os při každém zapnutí manipulátoru nebo po vyvolání tohoto FB. Určení referenční polohy probíhá nájezdem vozíku osy na koncový snímač v negativním směru pohybu a krátkým odjetím do kladného směru, kde tuto polohu nastaví.

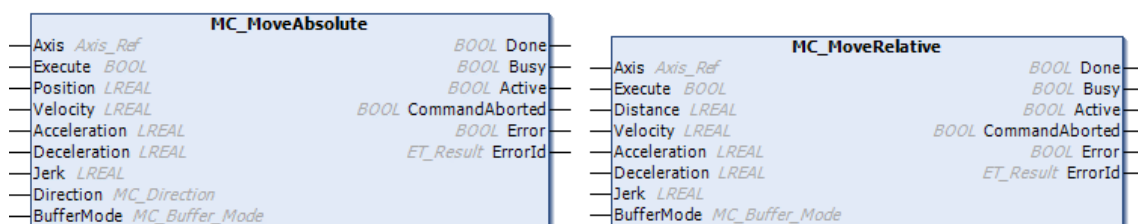


Obrázek 5.6: Funkční blok MC_Home [10]

PowerON - ManualMovement

V tomto kroku (příloha D) je realizován manuální ovládací mód. Je možné vykonávat buď relativní nebo absolutní pohyb. Pomocí panelu je možné nastavit vzdálenost, rychlost i zrychlení daného pohybu.

Jsou zde použity konverzní funkce, které zajišťují kompatibilitu mezi datovými typy, se kterými pracuje operátorský panel a datovými typy jednotlivých proměnných ve FB. V manuálním módu zajišťují absolutní a relativní pohyb os do žádané polohy funkční bloky MC_MoveAbsolute a MC_MoveRelative (obr. 5.7) z knihovny PLCOpen MC.

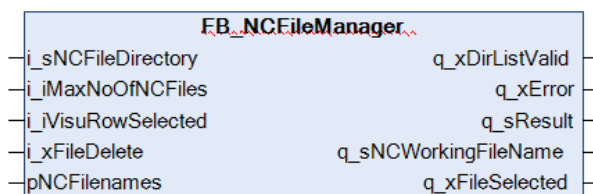


Obrázek 5.7: Funkční bloky MC_MoveAbsolute a MC_MoveRelative [10]

PowerON - NCFfile

V tomto kroku (příloha E) probíhá čtení NC souborů ze složky kontroléru, do které jsou nahrány soubory z SD karty. Také se zde přiřazují názvy těchto souborů do proměnných, které mají vlastní fyzickou adresu. To slouží pro ovládání pomocí operátorského panelu, kde lze vybrat soubor, který se provede. Tento zvolený soubor slouží jako vstupní proměnná pro FB SMC_ReadNCFile v POU NCFfileMode. Dále zde probíhá vypisování chybového hlášení interpolátoru pro zobrazení na operátorském panelu.

Použitý funkční blok FB_NCFFileManager (obr. 5.8) byl vyvinut nezávisle a tudíž nespadá pod žádnou knihovnu. Pro jeho funkci je využíváno prostředků knihovny CAA File. Pomocí něj lze otevřít úložiště se soubory, jež byly do kontroléru vloženy pomocí SD karty. Následně je možné v tomto úložišti provést výběr a odstranění CNC souborů.



Obrázek 5.8: Funkční blok NCFFileManager [10]

PowerON - OutQueue

Tento krok (příloha F) zahrnuje výběr prováděného G-kódu v módu OutQueue, který používá soubory předem nahrané v kontroléru a upravené v CNC editoru. Výběr probíhá pomocí operátorského panelu.

5.2. ŘÍDÍCÍ PROGRAM

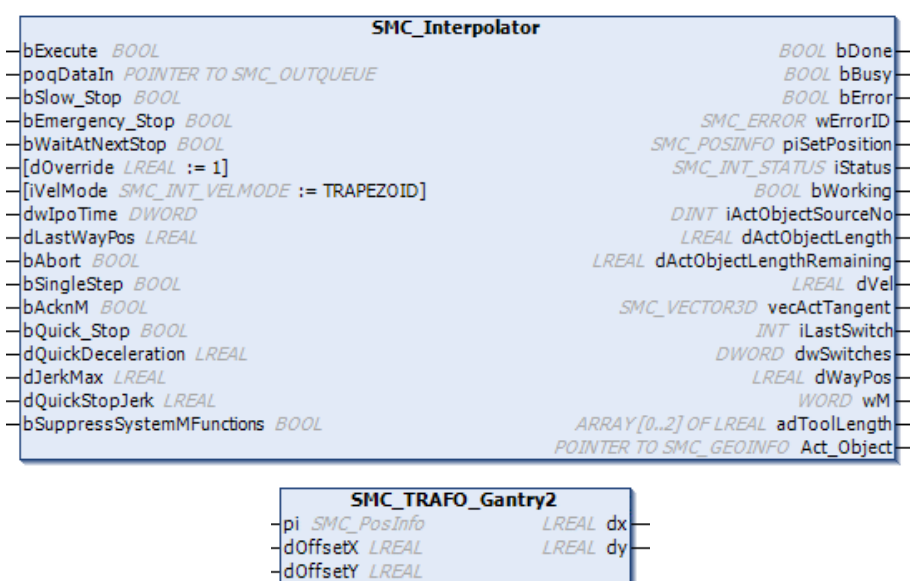
CNCtask

Na rozdíl od předchozích, toto POU (příloha G) běží v tasku CNC, který se stará o CNC řízení. Probíhá zde volání POU OutQueueMode a NCFileMode v závislosti na zvoleném ovládacím módu.

OutQueueMode

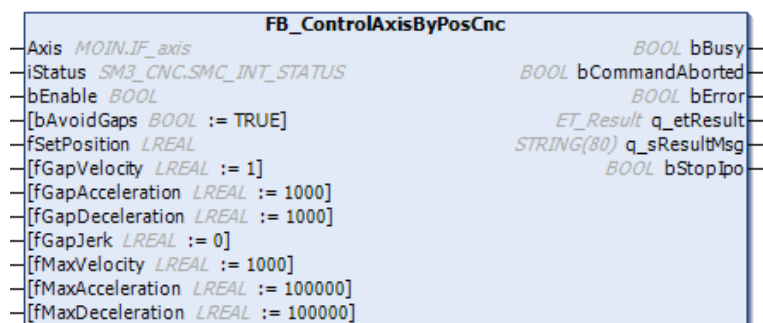
Toto POU (příloha H) slouží pro realizaci CNC řízení v ovládacím módu OutQueue.

Funkční bloky SMC_Interpolator a SMC_TrafoGantry2 (obr. 5.9) slouží ke zpracování cesty a získání žádaných poloh pro jednotlivé osy na konci každého cyklu interpolátoru. Délka jednoho cyklu je 1 ms (parametr dwIpoTime), což odpovídá hodnotě délky cyklu komunikace Sercos. Rychlostní profil je zvolený Quadratic_smooth, což je profil trapezoidního tvaru s omezeným trháním (jerk) a bez skoků dle hodnoty Maximum jerk.



Obrázek 5.9: Funkční bloky SMC_Interpolator a SMC_TrafoGantry2 [10]

Funkční blok FB_ControlAxisByPosCnc z knihovny CNCExtension zajišťuje vpisování žádaných hodnot pozice do rozhraní os na konci každého cyklu interpolátoru. Zde se nastavují hodnoty rychlosti a zrychlení pro vyplňování mezer v cestě a také limitní hodnoty rychlosti a zrychlení pro danou osu.

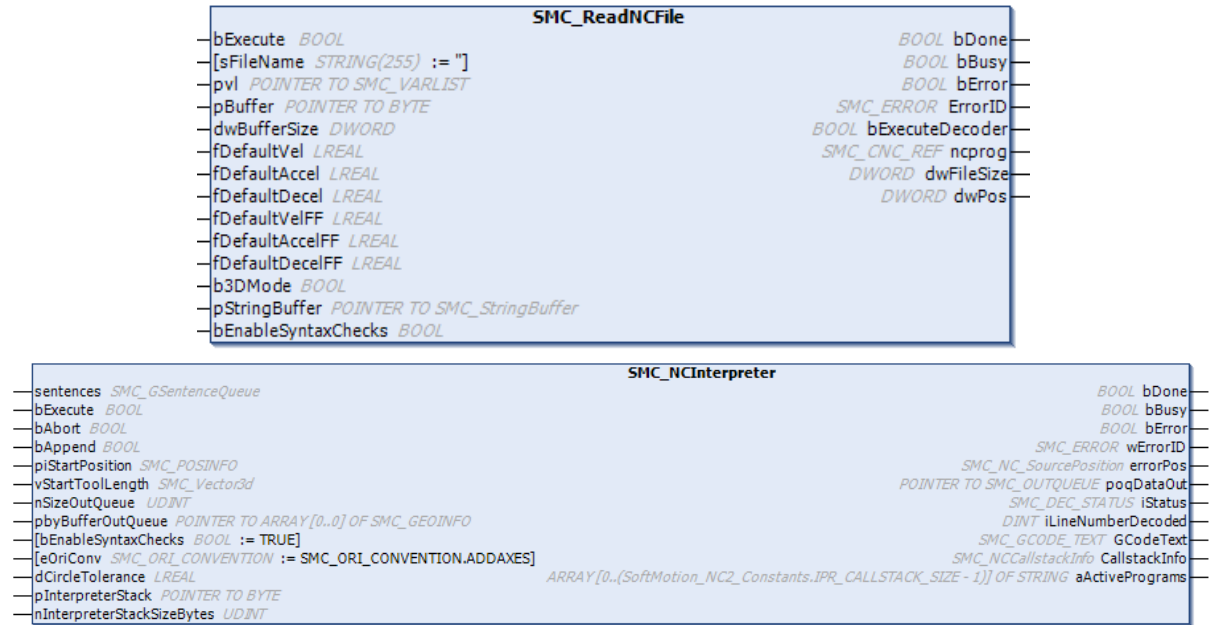


Obrázek 5.10: Funkční blok ControlAxisByPosCnc [10]

NCFileMode

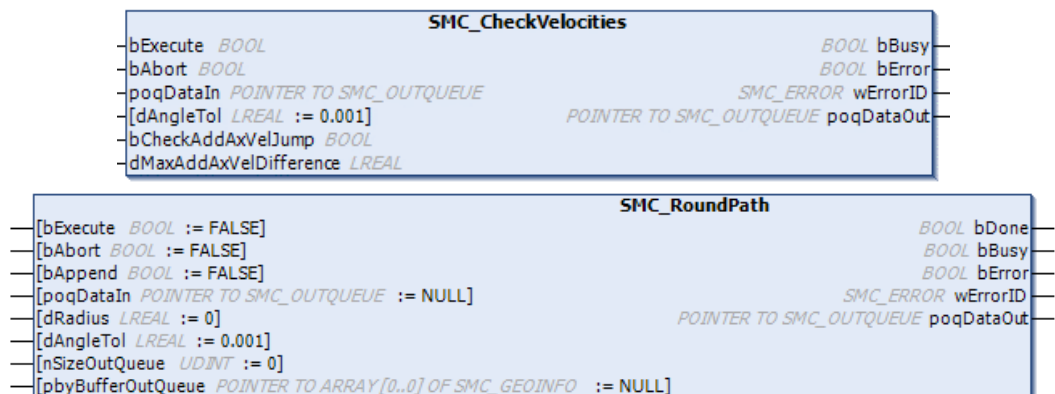
Toto POU (příloha I) zahrnuje ovládací mód NCFile, který čte G-kód ze souborů nahraných do kontroléru pomocí SD karty. Toto čtení souborů a jejich převod do struktury rozeznatelné interpolátorem zajišťují FB knihovny SM3_CNC.

Ve FB SMC_ReadNCFile2 (obr. 5.11) jsou nastavené výchozí hodnoty rychlosti a zrychlení pro normální a rychlý pohyb, pokud nejsou přímo definované v kódu. Tento nastavený a zpracovaný kód je poté v SMC_NCInterpreter upravený na strukturu OutQueue.



Obrázek 5.11: Funkční bloky SMC_ReadNCFile2, SMC_NCInterpreter [10]

Struktura OutQueue jde přes FB SMC_RoundPath a SMC_CheckVelocities (obr. 5.12), kde jsou upraveny rychlosti v ostrých rozích cesty. Dále do SMC_Interpolator a je dále zpracována jako v módu OutQueue. Hodnoty parametrů pro FB RoundPath a CheckVelocities se upravují buď přímo v dané instanci nebo globálně v CNC settings, stejně jako pro ostatní FB předzpracovávající cestu.



Obrázek 5.12: Funkční bloky SMC_RoundPath a SMC_CheckVelocities [10]

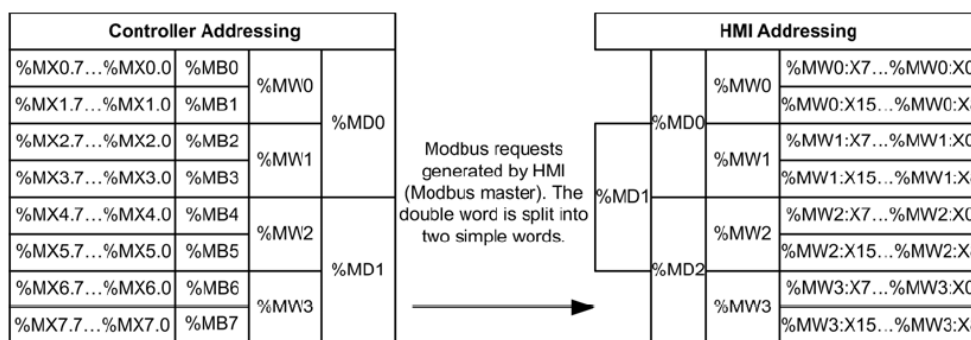
5.2. ŘÍDÍCÍ PROGRAM

Axis_DiagnosticMessages

Toto POU (příloha J) zajišťuje vepsání aktuálního stavu os do proměnných s fyzickou adresou, aby mohl aktuální stav os být zobrazen v operátorském panelu. Také zde probíhá přiřazení slovního vyjádření k číselné hodnotě enumerace etAxisState.

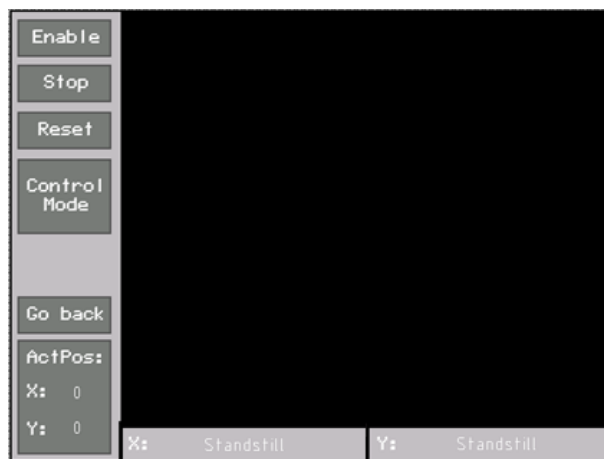
5.2.3. Operátorský panel

Pro ovládání manipulátoru je využit operátorský panel, který s kontrolérem komunikuje pomocí Modbus. Je třeba dát si pozor na zápis jejich adresy vzhledem k rozdílu v adresování proměnných v kontroléru a panelu, jak je vidět na obrázku 5.13. V tabulce K.1 (příloha K) je vidět, jaké fyzické adresy mají přiřazené jednotlivé proměnné dle adresování kontroléru a panelu.



Obrázek 5.13: Rozdíl adresová PLC a operátorského panelu [11]

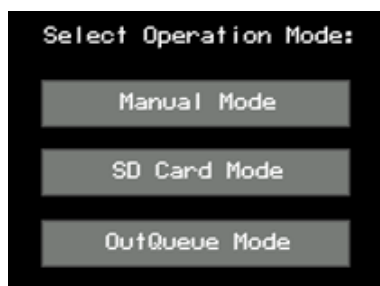
Uživatelské rozhraní se skládá ze 6 hlavních panelů a jednoho vyskakovacího. Společným prvkem je postranní lišta, na které se nachází ovládání os manipulátoru, tlačítko pro výběr operačního módu, aktuální poloha os a jejich stav na spodní liště (obr. 5.14).



Obrázek 5.14: Základní okno op. panelu

Po stlačení tlačítka Control Mode, vyběhne vyskakovací okno (obr. 5.15), za pomoci kterého proběhne výběr operačního módu.

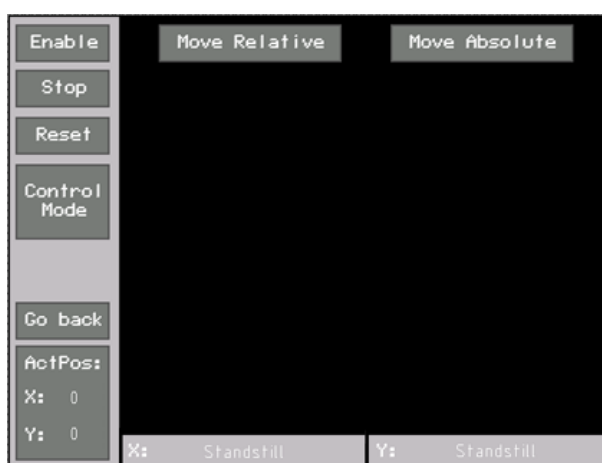
5. TVORBA ŘÍDÍCÍHO SYSTÉMU A ODLADĚNÍ MECHANIKY



Obrázek 5.15: Vyskakovací okno op. panelu

Manual Mode

Manuální ovládání probíhá ve dvou režimech – Move Absolute, Move Relative. Po výběru manuálního módu se zobrazí obrazovka (obr. 5.16), na které se vybere jeden z těchto režimů.



Obrázek 5.16: Výběr režimu během manuálního ovládání

V režimu Move Relative (obr. 5.17) se osa posune vždy o danou vzdálenost z aktuální polohy, je možné nastavit různou vzdálenost pro každou osu. Dalšími parametry, které jdou změnit, jsou rychlost a zrychlení. Obrazovka obsahuje i tlačítko Start, kterým se pohyb spustí. Pokud je třeba pohyb v jeho průběhu zastavit, využije se tlačítko Stop v postranní liště.



Obrázek 5.17: Manuální režim Move Relative

5.2. ŘÍDÍCÍ PROGRAM

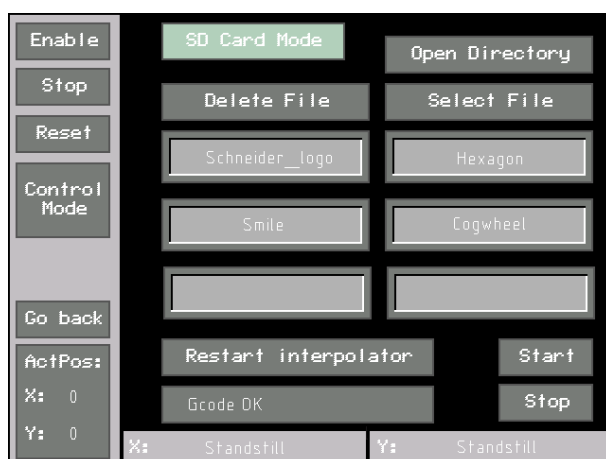
Režim Move Absolute (obr. 5.18) se liší tím, že zadávaná hodnota je vzdálenost od předem definovaného nulového bodu a osa do této vzdálenosti dojde z jakékoliv výchozí polohy. Stejně jako v předchozím režimu je možné nastavit rychlost, zrychlení pohybu a spustit jej.



Obrázek 5.18: Manuální režim Move Absolute

SD Card Mode

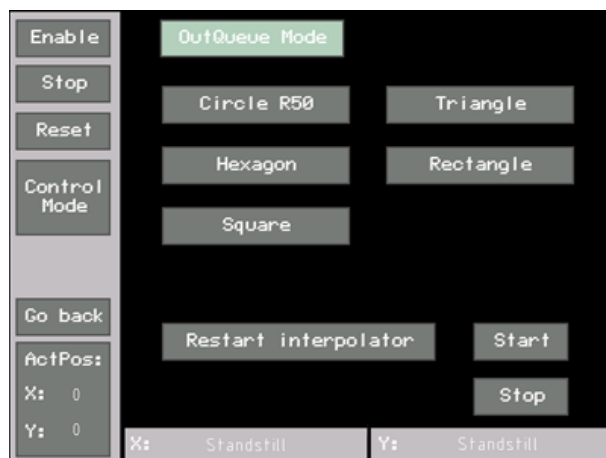
Ovládací mód SD card Mode (obr. 5.19) spočívá ve výběru souboru nahraného pomocí SD karty a následného spuštění interpolátoru. Maximální počet CNC souborů je šest, ze kterých je možné vybrat vždy jeden. Výběr probíhá stlačením tlačítka Open Directory, následným zvolením požadovaného souboru a stlačením tlačítka Select File. Při výběru jiného programu je třeba toto tlačítko opětovně zmáčknout. Je zde i tlačítko Delete File, které umožňuje odstranění nepotřebných souborů. Běh interpolátoru je spuštěn pomocí tlačítka Start. Pokud je třeba běh interpolátoru pozastavit, je možné použít tlačítko Stop, po jeho opětovném stlačení bude pohyb pokračovat. Úplně zastavit interpolátor je možné pomocí stlačení tlačítka Stop a Restart interpolator, což běh zastaví a vrátí do výchozího stavu, po opětovném stlačení Stop, nebude pohyb pokračovat.



Obrázek 5.19: CNC řízení - SD Card Mode

OutQueue Mode

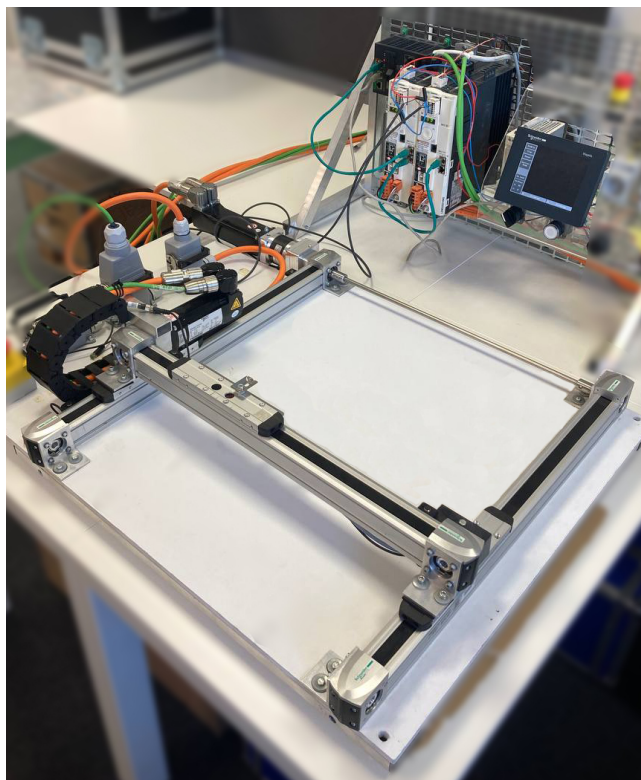
V tomto ovládacím módu (obr. 5.20) probíhá ovládání interpolátoru stejně jako v SD Card Mode. Rozdílem je zde možnost výběru z několika přednastavených cest, které může manipulátor vykonávat. Jejich výběr je možný stiskem příslušného tlačítka.



Obrázek 5.20: CNC řízení - OutQueue Mode

5.3. Odladění programu a mechaniky

Pro správné fungování programu je třeba propojit mechanickou konfiguraci (obr.5.21) s řídicím programem a to pomocí odladění.



Obrázek 5.21: Mechanická konfigurace CNC manipulátoru

5.3. ODLADĚNÍ PROGRAMU A MECHANIKY

5.3.1. Nastavení kontrolní smyčky pohonů

Servopohony jsou řízeny pomocí regulační struktury, která se skládá ze tří smyček - proudové, rychlostní a polohové. Tyto smyčky jsou ovládány parametry, kontrolujícími činnost motoru. Pro produkty firmy Schneider Electric probíhá nastavení těchto parametrů pomocí programu SoMove, kde se v režimu expert tuning upravují jednotlivé hodnoty pro dosažení ideálního nastavení, k tomuto pomáhá vykreslování daných parametrů osciloskopem.

Nastavení rychlostní smyčky má za cíl, aby aktuální rychlost otáčení co nejvíce odpovídala referenční rychlosti. Toto je ovlivněno hodnotou proporciálního zesílení (parametr KPn), integrační akční doby (TNn) a filtru hodnoty referenční rychlosti (TAUnref).

Polohovou smyčku je ideální nastavit tak, aby aktuální poloha dosáhla hodnoty referenční polohy co nejrychleji, ale zároveň by nemělo dojít k jejímu překročení. Nastavení je kontrolováno proporcionálním zesílením polohové smyčky (KPP).

Proudová regulační smyčka má předdefinované hodnoty dle daného motoru, tudíž není třeba její úprava.

5.3.2. Odladění programu

V rámci odladění programu bylo třeba nastavit škálování jednotek polohy. Víme, že enkodér má rozlišení 131072 inkrementů na jednu otáčku a z technických údajů lineární osy lze zjistit, že jedna otáčka odpovídá délkovým 84 mm. Čímž lze dosáhnout výsledného polohování v milimetrech.

Pro správnou funkčnost CNC řízení bylo třeba čas cyklu interpolátoru nastavit, aby korespondoval s délkou cyklu komunikace Sercos, což odpovídá 1 ms. V závislosti na tomto nastavení byl rozdělen běh PLC do dvou činností, aby interpolátor pracoval s intervalem 1 ms a zbytek programu s plně postačujícími 10 ms.

Dále byly zjištěny limitní hodnoty rychlosti a zrychlení mechaniky, které byly uloženy do proměnných, jež vstupují do funkčních bloků zajišťujících ovládání os během CNC řízení.

6. Závěr

Práce měla za cíl návrh řídicího systému laboratorního modelu CNC manipulátoru. Nejdříve se v rešeršní části shrnuly poznatky aktuálního stavu poznání nejen z oboru manipulátorů a průmyslových robotů, ale také z oblasti řídicích systémů a jejich CNC řízení.

Z těchto poznatků byly vybrány tři návrhy mechanické konfigurace a funkce řídicího systému. Po konzultaci s odborníky z oboru byla stanovena hodnotící kritéria, dle kterých proběhl výběr finální mechanické konfigurace a podoby řídicího systému, aby výsledek co nejlépe odpovídal zadání. Jelikož jde o laboratorní model, největší váhu mezi rozhodovacími kritérii měla cena. Proto byl jako základ zvolen dvouosý manipulátor, který z vybraných variant v tomto ohledu vychází nejlépe. Osy manipulátoru jsou řízeny za pomoci servomotorů s inkrementálními enkodéry, tudíž pro nájezd do referenční polohy bylo nutné osy vybavit koncovými snímači. Pro ovládání celého manipulátoru byl zvolen operátorský panel, který umožňuje uživatelsky přívětivé ovládání a nabízí široké možnosti.

Dle výběru funkcí řídicího systému byly zvoleny dvě varianty CNC řízení. Jedna zajišťuje zpracování G-kódu bez nutnosti jeho předzpracování a druhá umožňuje čtení G-kódů nahraných do kontroléru zapomocí SD karty. Pro uživatelské polohování byl řídicí systém doplněn o manuální režim, který umožňuje nájezd do cílové polohy v režimu Move Absolute nebo Move Relative. Tvorba řídicího systému proběhla ve vývojovém prostředí Ecostruxure Machine Expert. Tyto módy jsou ovládány za pomoci operátorského panelu, který byl zpracován společně s řídicím systémem. Práce se také zaměřila na praktické odladění programu a nastavení pohonů.

Během výsledného testování manipulátoru bylo ověřeno, že manipulátor zvládá zpracování jak jednodušších G-kódů, tak i složitějších. Také zvládá pohyby jak při nízkých, tak i vysokých rychlostech. Pro jeho budoucí vylepšení je možné využít IIoT (Industrial Internet of Things) funkce PLC M262 a tím celý manipulátor zasadit do konceptu Průmyslu 4.0. Z mechanického hlediska je možné vylepšení na straně použitých motorů, u kterých se nabízí možnost zvolit absolutní enkodéry, tudíž by se eliminovala nutnost použití koncových snímačů. Zvolené osy manipulátoru jsou poháněny pomocí ozubeného řemenu. Pokud by byla vyžadována větší přesnost a tuhost, bylo by možné zvážit využití os s kuličkovým šroubem, což by ovšem negativně ovlivnilo cenu. Možné vylepšení řídicího systému spočívá ve využití posledního kompilačního módu, který dle multikritériálního hodnocení neuspěl a tudíž nebyl zpracován, jeho možná výhoda spočívá v použití proměnných uvnitř G-kódu.

Tvorba této práce mi pomohla rozšířit si znalosti z oblasti programování PLC, speciálně CNC řízení, které nabízí nepřeberné možnosti. Mimo jiné mi tato práce dala i možnost, získat praktické zkušenosti s odladěním mechanické konfigurace.

Literatura

- [1] BILLARD, Aude a Danica KRAGIC. Trends and challenges in robot manipulation. *Science* [online]. 2019, **364**(6446) [cit. 2022-04-27]. ISSN 0036-8075. Dostupné z: doi:10.1126/science.aat8414
- [2] BENEŠ, Pavel. *Automatizace a automatizační technika: prostředky automatizační techniky*. 5., rozš. a aktualiz. vyd. Brno: Computer Press, 2014. ISBN 978-80-251-3747-5.
- [3] ADAMEC, Jaromír. *Technologie automatizovaných výroby* [online]. Ostrava: VŠB - Technická univerzita, 2006 [cit. 2022-04-27]. ISBN 80-248-0871-4.
- [4] SKAŘUPA, Jiří. *Průmyslové roboty a manipulátory* [online]. Ostrava: Vysoká škola báňská - Technická univerzita, [2008] [cit. 2022-04-27]. ISBN 978-80-248-1522-0.
- [5] BELDA, KVĚTOSLAV. Paralelní robotické struktury a moderní metody jejich řízení. *Automatizace* [online]. **52**(5, strany 296-300) [cit. 2022-04-28]. Dostupné z: <http://library.utia.cas.cz/separaty/historie/belda-0325559.pdf>
- [6] NOVOTNÝ, František, Vlastimil HOTAŘ, Marcel HORÁK, aj. *ÚVOD DO AUTOMATIZACE A ROBOTIZACE VE STROJÍRENSTVÍ* [online]. Liberec: Technická univerzita v Liberci, 2020 [cit. 2022-04-28]. ISBN 978-80-7494-545-8. Dostupné z: <https://publi.cz/download/publication/1275?online=1>
- [7] *Schneider Electric Global* [online]. 2022 [cit. 2022-04-28]. Dostupné z: <https://www.se.com/ww/en/>
- [8] BOLTON, W. *Programmable logic controllers*. Sixth edition. Amsterdam: Newnes, 2015. ISBN 978-0-12-802929-9.
- [9] *Schneider Electric EcoStruxure Machine Expert product help* [online]. 2022 [cit. 2022-04-29]. Dostupné z: <https://product-help.schneider-electric.com/Machine%20Expert/V2.0/LandingPages/en/index.html>
- [10] *CODESYS SoftMotion SM3_CNC product help* [online]. 2022 [cit. 2022-04-29]. Dostupné z: https://product-help.schneider-electric.com/Machine%20Expert/V2.0/en/SM3_CNC/index.htm#t=SM3_CNC%2Fmodules%2Fidx-SM3_CNC-lib.html
- [11] SCHNEIDER ELECTRIC. *Modicon M262 Logic/Motion Controller: User Guide* [online]. 2022. Dostupné také z: https://download.schneider-electric.com/files?p_enDocType=User+guide&p_File_Name=EIO0000004285.02.pdf&p_Doc_Ref=EIO0000004285
- [12] SCHNEIDER ELECTRIC. *Lexium 32S: User Guide* [online]. 2021. Dostupné také z: https://download.schneider-electric.com/files?p_enDocType=User+guide&p_File_Name=0198441114060_03.pdf&p_Doc_Ref=0198441114060-EN

A. POU PowerON

```

PROGRAM PowerON
VAR_INPUT
    NumberOfGCode           : INT;
END_VAR
VAR
    // Initialization
    iState                   : INT;

    //Home
    bHomeDone                : BOOL;
    bHome                    : BOOL;
    lrHomePosition           : LREAL;
    bHomeDoneX               : BOOL;
    bHomeDoneY               : BOOL;
    HomeX                    : PLCO.MC_Home;
    HomeY                    : PLCO.MC_Home;
    iHomeState               : INT;

    //Manual Movement
    MC_MoveRelative_X        : PLCO.MC_MoveRelative;
    MC_MoveRelative_Y        : PLCO.MC_MoveRelative;
    DistanceXMRel AT %MW8    : INT;
    DistanceYMRel AT %MW9    : INT;
    VelocityMRel AT %MW10    : INT;
    bExecuteMRel AT %MX0.6   : BOOL;

    MC_MoveAbsolute_X        : PLCO.MC_MoveAbsolute;
    MC_MoveAbsolute_Y        : PLCO.MC_MoveAbsolute;
    VelocityMAbs AT %MW13    : INT;
    PositionXMAbs AT %MW11    : INT;
    PositionYMAbs AT %MW12    : INT;
    bExecuteMAbs AT %MX0.7   : BOOL;

    AccDccManMovement AT %MW80 : INT := 150;

    //NCFile
    FileName                  : STRING(255);
    FB_NCFileManager_0        : FB_NCFileManager;
    FileNamesArray            : ARRAY [0..6] OF STRING;
    sSelectCNCFileNCFile1 AT %MB32 : STRING;
    sSelectCNCFileNCFile2 AT %MB48 : STRING;
    sSelectCNCFileNCFile3 AT %MB64 : STRING;
    sSelectCNCFileNCFile4 AT %MB96 : STRING;
    sSelectCNCFileNCFile5 AT %MB112 : STRING;

```

```

sSelectCNCFileNCFile6 AT %MB128 : STRING;
iSelectCNCFileNCFile AT %MW15 : INT;
DeleteNCFile AT %MX1.5 : BOOL;
sIpoError AT %MB192 : STRING;
iIpoErrorLine AT %MW105 : INT;
bIpoError AT %MX1.6 : BOOL;

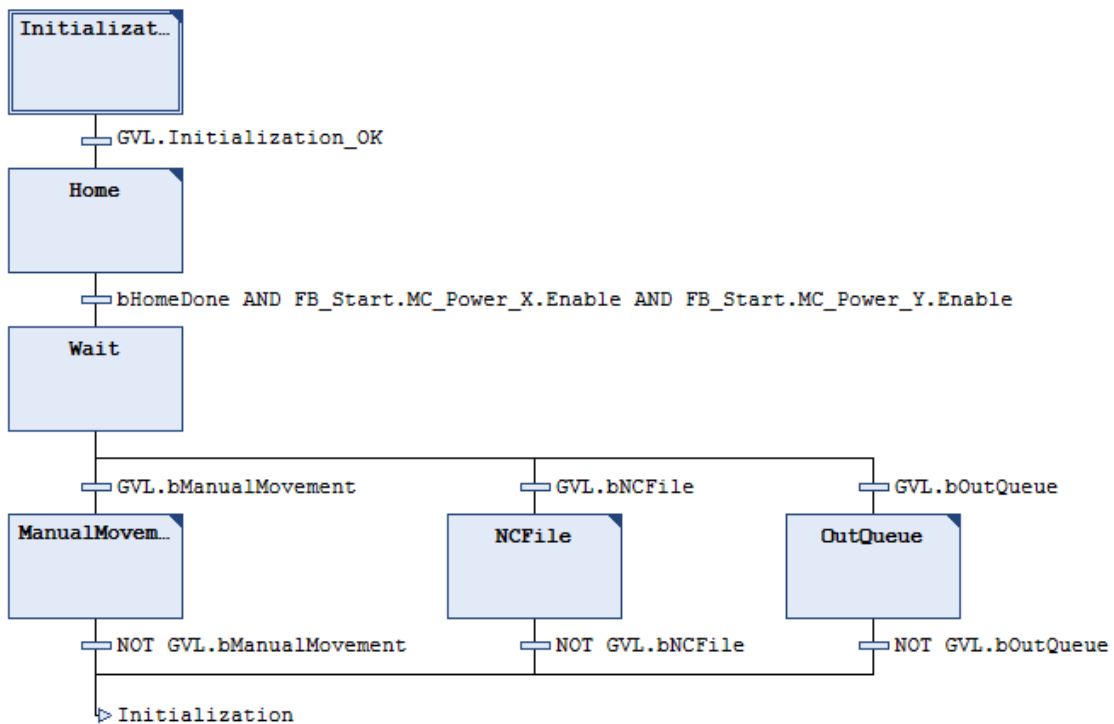
```

```

//OutQueue
iSelectCNCFileOutQueue AT %MW14 : INT;

```

END_VAR



Obrázek A.1: POU PowerON

B. POU PowerON - Initialization

```

IF NOT GVL.Initialization_OK THEN
    SercosMaster.SercosPhaseChanger.DesiredPhase:=
    S3M.ET_SercosPhase.NRT;
    CASE iState OF //Setting SW limits of the axis and setting
                    Sercos Phase after setup
0: // Sercos Phase = NRT (-1) – Setup of the communication ,
    first stage after power on or during error
        GVL.Initialization_OK:= FALSE;
        GVL.gSercosPhaseSet:= S3M.ET_SercosPhase.NRT;
        IF SercosMaster.SercosPhaseChanger.ActualState =
        S3M.ET_SercosState.NRT THEN
            iState:= 10;
        END_IF

10: //Setting SW limits of the axis
    Axis_X_Lexium32S.Axis.SetAxisTypeLinearWithLimits
    (i_lrNegativeDirectionLimit:= 0,
    i_lrPositiveDirectionLimit:= 270);
    Axis_Y_Lexium32S.Axis.SetAxisTypeLinearWithLimits
    (i_lrNegativeDirectionLimit:= 0,
    i_lrPositiveDirectionLimit:= 320);
        iState:= 20;

20: // Setting Sercos Phase = CP4 – fully
functional communication
        SercosMaster.SercosPhaseChanger.DesiredPhase:=
        S3M.ET_SercosPhase.Phase4;
        GVL.gSercosPhaseSet := S3M.ET_SercosPhase.Phase4;
        IF SercosMaster.SercosPhaseChanger.ActualState =
        S3M.ET_SercosState.Phase4 THEN
            GVL.Initialization_OK :=TRUE;
            iState:= 30;
        END_IF

30: // Checkup if the setup of the communication was
    sucessfull , if not the setup will go through again
        IF NOT GVL.Initialization_OK THEN
            iState:= 0;
        END_IF
END_CASE
END_IF

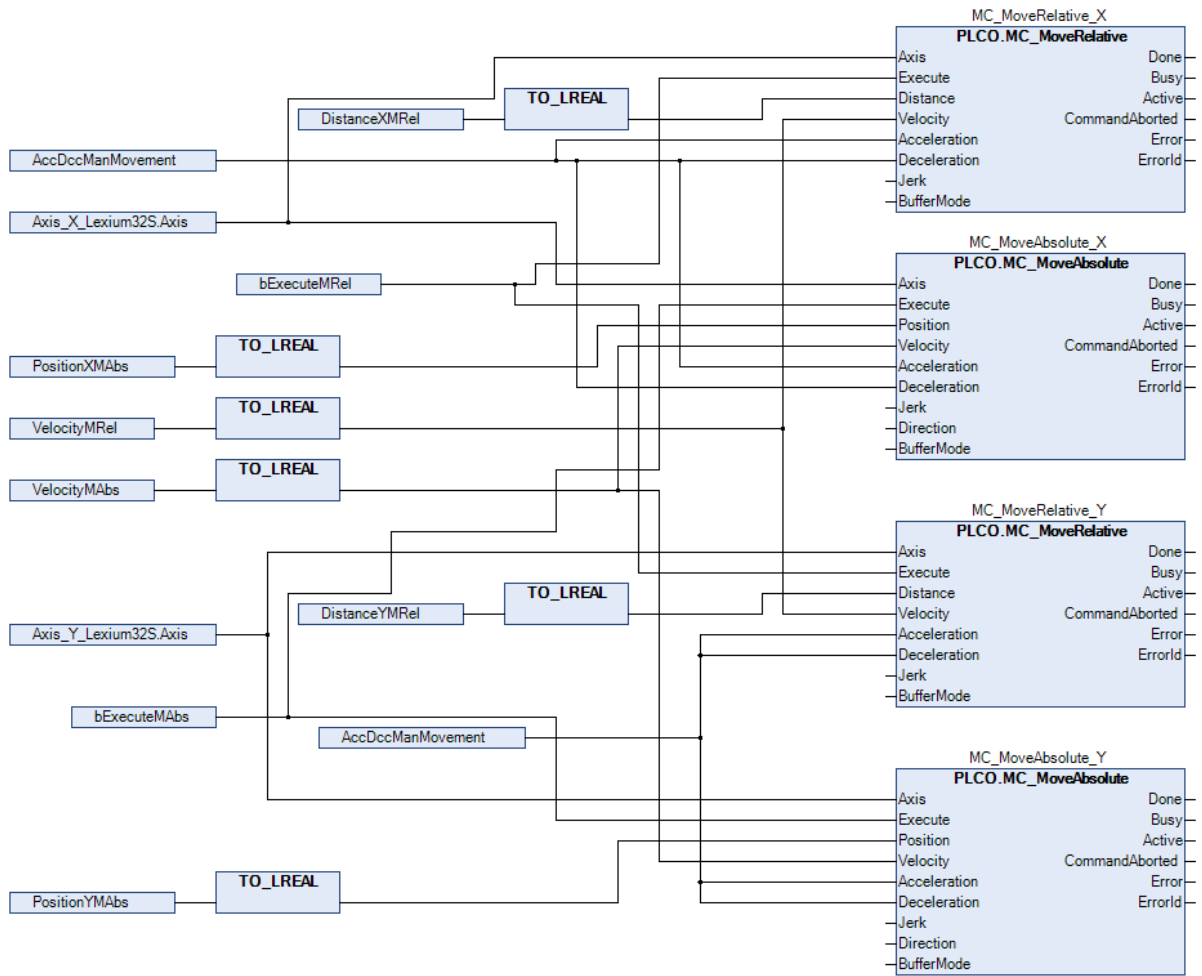
```

C. POU PowerON - Home

```
//Homing of the axis

IF NOT bHomeDone THEN
  CASE iHomeState OF
    0:
      bHome:= TRUE;
      FB_Start.bEnable := TRUE;
      IF Axis_X_Lexium32S.Axis.etAxisState = 2 AND
      Axis_Y_Lexium32S.Axis.etAxisState = 2 THEN
        iHomeState:=1;
      END_IF
    1:
      HomeX(Axis:= Axis_X_Lexium32S.Axis, Execute:= bHome,
      Position:= 0, Done=> bHomeDoneX);
      IF bHomeDoneX THEN
        iHomeState:= 2;
      END_IF
    2:
      HomeY(Axis:= Axis_Y_Lexium32S.Axis, Execute:= bHome,
      Position:= 0, Done=> bHomeDoneY);
      IF bHomeDoneY THEN
        iHomeState:= 3;
      END_IF
    3:
      IF bHomeDoneX AND bHomeDoneY THEN
        bHomeDone:= TRUE;
        bHome:= FALSE;
        FB_Start.bEnable := FALSE;
      END_IF
      IF bHomeDone THEN
        iHomeState:= 0;
      END_IF
  END_CASE
END_IF
```

D. POU PowerON - ManualMovement



Obrázek D.1: POU PowerON - ManualMovement

E. POU PowerON - NCFile

```
//Writing File names array values to the variables for
the selection in HMI
sSelectCNCFileNCFile1:= PowerON.FileNameArray [1];
sSelectCNCFileNCFile2:= PowerON.FileNameArray [2];
sSelectCNCFileNCFile3:= PowerON.FileNameArray [3];
sSelectCNCFileNCFile4:= PowerON.FileNameArray [4];
sSelectCNCFileNCFile5:= PowerON.FileNameArray [5];
sSelectCNCFileNCFile6:= PowerON.FileNameArray [6];

//Error notification for HMI
CASE NCFileMode.SMC_NCInterpreter_0.iStatus OF
0:
    sIpoError:= 'Waiting';
    bIpoError:= FALSE;
1,2:
CASE NCFileMode.SMC_NCInterpreter_0.wErrorID OF
0:
    sIpoError:= 'Gcode OK';
    bIpoError:= FALSE;
1204:
    sIpoError:= 'Error at';
    bIpoError:= TRUE;
END_CASE
END_CASE

iIpoErrorLine :=
TO_INT (NCFileMode.SMC_NCInterpreter_0.iLineNumberDecoded) + 1;

//Choosing which G-code to execute
CASE iSelectCNCFileNCFile OF
0: PowerON.NumberOfGCode:= 1;
1: PowerON.NumberOfGCode:= 2;
2: PowerON.NumberOfGCode:= 3;
3: PowerON.NumberOfGCode:= 4;
4: PowerON.NumberOfGCode:= 5;
5: PowerON.NumberOfGCode:= 6;
END_CASE

FB_NCFileManager_0(i_sNCFileDirectory:= '\usr\NCFiles\',
    i_iMaxNoOfNCFiles:= 6,
    i_iVisuRowSelected:= NumberOfGcode,
    i_xFileDelete:= DeleteNCFile,
    pNCFileNames:= ADR(FileNameArray),
    q_sNCWorkingFileName=> FileName);
```

F. POU PowerON - OutQueue

```
//Choosing which G-code to execute
```

```
CASE iSelectCNCFileOutQueue OF
0:
    OutQueueMode.fbSMC_Interpolator.poqDataIn:= ADR(Circle_R50);
1:
    OutQueueMode.fbSMC_Interpolator.poqDataIn:= ADR(Triangle);
2:
    OutQueueMode.fbSMC_Interpolator.poqDataIn:= ADR(Hexagon);
3:
    OutQueueMode.fbSMC_Interpolator.poqDataIn:= ADR(Rectangle);
4:
    OutQueueMode.fbSMC_Interpolator.poqDataIn:= ADR(Square);
END_CASE
```

G. POU CNCTask

//This POU is designed to call POU's of different control modes in separate

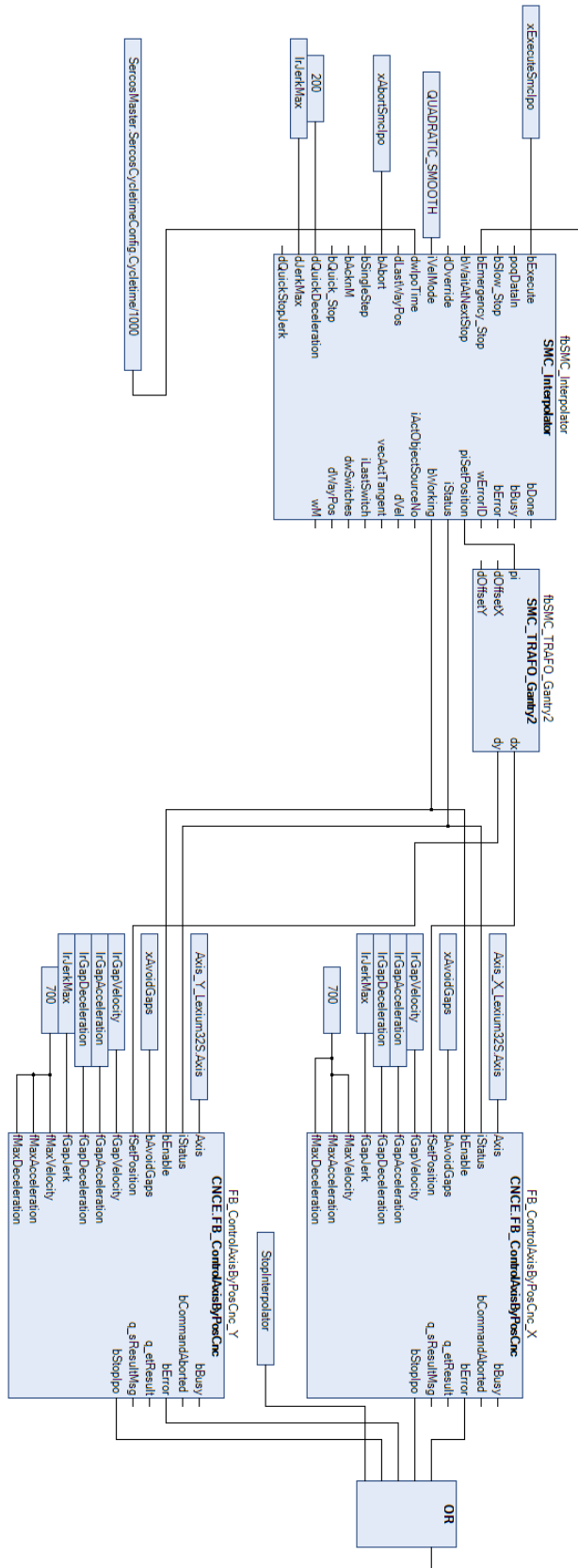
```
IF GVL.bOutQueue THEN
    OutQueueMode ();
END_IF
```

```
IF GVL.bNCFile THEN
    NCFileMode ();
END_IF
```

H. POU OutQueueMode

```
PROGRAM OutQueueMode
VAR_INPUT
    fbSMC_Interpolator          : SM3_CNC.SMC_Interpolator;
END_VAR

VAR
    xExecuteSmcIpo AT %MX1.1    : BOOL;
    xAbortSmcIpo AT %MX1.0      : BOOL;
    xAvoidGaps                 : BOOL := TRUE;
    lrGapVelocity               : LREAL := 20;
    lrGapAcceleration           : LREAL := 50;
    lrGapDeceleration           : LREAL := 50;
    FB_ControlAxisByPosCnc_X    : CNCE.FB_ControlAxisByPosCnc;
    FB_ControlAxisByPosCnc_Y    : CNCE.FB_ControlAxisByPosCnc;
    fbSMC_TRAFO_Gantry2        : SM3_CNC.SMC_TRAFO_Gantry2;
    StopInterpolator AT %MX1.2   : BOOL;
    lrJerkMax                   : LREAL := 400;
END_VAR
```



Obrázek H.1: POU OutQueueMode

I. POU NCFileMode

```
PROGRAM NCFileMode
```

```
VAR_INPUT
```

```
    fbSMC_Interpolator          : SMC_CNC.SMC_Interpolator;
    SMC_LimitDynamics_0        : SMC_LimitDynamics;
    SMC_CheckVelocities_0      : SMC_CheckVelocities;
```

```
END_VAR
```

```
VAR
```

```
    xExecuteSmcIpo AT %MX1.1    : BOOL;
    xAbortSmcIpo AT %MX1.0      : BOOL;
    xAvoidGaps                   : BOOL := TRUE;
    lrGapVelocity                : LREAL := 20;
    lrGapAcceleration            : LREAL := 50;
    lrGapDeceleration            : LREAL := 50;
    FB_ControlAxisByPosCnc_X     : CNCE.FB_ControlAxisByPosCnc;
    FB_ControlAxisByPosCnc_Y     : CNCE.FB_ControlAxisByPosCnc;
    fbSMC_TRAFO_Gantry2          : SMC_CNC.SMC_TRAFO_Gantry2;
    StopInterpolator AT %MX1.2   : BOOL;
    lrJerkMax                    : LREAL := 400;
    SMC_ReadNCFile2_0            : SMC_ReadNCFile2;
    FileName                     : STRING(255);
    F_TRIG_0                     : F_TRIG;
    SMC_NCInterpreter_0         : SMC_NCInterpreter;
    astGCodeBuffer : ARRAY [0..199] OF SMC_CNC.SMC_GCODE_WORD;
    fbSMC_InterpolatorNCFile     : SMC_Interpolator;
    SMC_NCDecoder_0             : SMC_NCDecoder;
    FB_NCFileManager_0          : FB_NCFileManager;
    FileNamesArray              : ARRAY [0..6] OF STRING;
    stPosInfoStartPos           : SMC_CNC.SMC_POSINFO;
    SMC_CheckVelocities_1       : SMC_CheckVelocities;
    SMC_SmoothPath_0            : SMC_SmoothPath;
```

```
END_VAR
```

Na obrázku [I.1](#) je vidět první část POU NCFile s funkčními bloky pro zpracování CNC programu, zbytek POU není znázorněn, jelikož je totožný jako u OutQueueMode.

J. POU Axis__DiagnosticMessages

```

PROGRAM Axis_DiagnosticMessages
VAR
    AxisStateX AT %MB144          : STRING;
    AxisStateY AT %MB176          : STRING;
END_VAR

CASE Axis_X_Lexium32S.Axis.etAxisState OF
    0: AxisStateX:= 'ErrorStop';
    1: AxisStateX:= 'Disabled';
    2: AxisStateX:= 'Standstill';
    3: AxisStateX:= 'Stopping';
    4: AxisStateX:= 'Homing';
    5: AxisStateX:= 'Motion';
    6: AxisStateX:= 'Motion';
    7: AxisStateX:= 'Motion';
ELSE
    AxisStateX:= 'Unknown';
END_CASE

CASE Axis_Y_Lexium32S.Axis.etAxisState OF
    0: AxisStateY:= 'ErrorStop';
    1: AxisStateY:= 'Disabled';
    2: AxisStateY:= 'Standstill';
    3: AxisStateY:= 'Stopping';
    4: AxisStateY:= 'Homing';
    5: AxisStateY:= 'Motion';
    6: AxisStateY:= 'Motion';
    7: AxisStateY:= 'Motion';
ELSE
    AxisStateY:= 'Unknown';
END_CASE

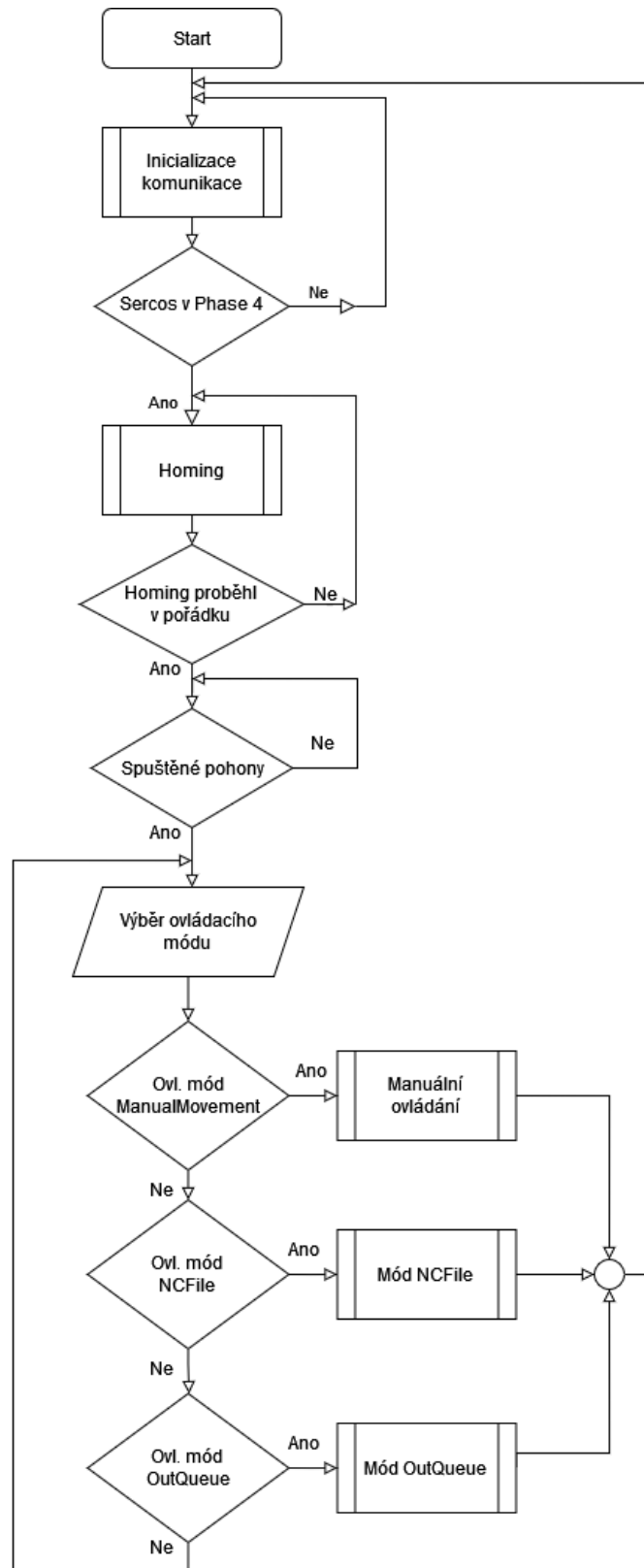
```

K. Adresování

Proměnná	PLC	HMI
bEnable	%MX0.0	%MW0:X0
bStop	%MX0.1	%MW0:X1
bReset	%MX0.2	%MW0:X2
bManualMovement	%MX0.3	%MW0:X3
bNCFile	%MX0.4	%MW0:X4
bOutQueue	%MX0.5	%MW0:X5
bExecuteMRel	%MX0.6	%MW0:X6
bExecuteMAbs	%MX0.7	%MW0:X7
bAbortIpo	%MX1.0	%MW0:X8
bExecuteIpo	%MX1.1	%MW0:X9
bStopIpo	%MX1.2	%MW0:X10
bOpenDir	%MX1.3	%MW0:X11
bSelectFile	%MX1.4	%MW0:X12
bDeleteNCFile	%MX1.5	%MW0:X13
bIpoError	%MX1.6	%MW0:X14
rActPosX	%MD1	%MW2
rActPosY	%MD2	%MW4
iDistanceXMRel	%MW8	%MW8
iDistanceYMRel	%MW9	%MW9
iVelocityMRel	%MW10	%MW10
iTargetPosXMAbs	%MW11	%MW11
iTargetPosYMAbs	%MW12	%MW12
IVelocityMAbs	%MW13	%MW13
iSelectCNCFileOutQueue	%MW14	%MW14
iSelectCNCFileNCFile	%MW15	%MW15
SelectCNCFileNCFile1	%MB32	%MW16
SelectCNCFileNCFile2	%MB48	%MW24
SelectCNCFileNCFile3	%MB64	%MW32
SelectCNCFileNCFile4	%MB96	%MW48
SelectCNCFileNCFile5	%MB112	%MW56
SelectCNCFileNCFile6	%MB128	%MW64
sAxisStateX	%MB144	%MW72
iAccDccManMovement	%MW80	%MW80
sAxisStateY	%MB176	%MW88
sIpoError	%MB192	%MW96
iIpoErrorLine	%MB105	%MW105

Tabulka K.1: Přiřazení proměných fyzickým adresám PLC a operátorského panelu

L. Vývojový diagram



Obrázek L.1: Vývojový diagram programu