



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

DEPARTMENT OF CONTROL AND INSTRUMENTATION

ADD-ON INSTRUKCE A FACEPLATE PRO DOBOT

ADD-ON INSTRUCTIONS AND FACEPLATES FOR DOBOT

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Radim Novák

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Radek Štohl, Ph.D.

BRNO 2023

Diplomová práce

magisterský navazující studijní program **Kybernetika, automatizace a měření**

Ústav automatizace a měřicí techniky

Student: Bc. Radim Novák

ID: 211166

Ročník: 2

Akademický rok: 2022/23

NÁZEV TÉMATU:

Add-On instrukce a Faceplate pro Dobot

POKYNY PRO VYPRACOVÁNÍ:

- 1) Popište práci s Dobotem a PLC řady CompactLOGIX.
- 2) Vytvořte knihovnu Add-On instrukcí pro práci s robotem.
- 3) Vytvořte FacePlate pro vizualizaci dat v FT View Studiu podle standardu ANSI/ISA-101.01-2015.
- 4) Realizujte demonstrační úlohu.
- 5) Ověřte své řešení.

DOPORUČENÁ LITERATURA:

Logix 5000 Controllers Tasks, Programs, and Routines. Programming Manual. Rockwell Automation. 1756-PM005H-EN-P. 2018.

Dobot Magician Communication Protocol. Shenzhen Yuejiang Technology Co.,Ltd. 2019.

<https://download.dobot.cc/product-manual/dobot-magician/pdf/en/Dobot-Communication-Protocol-V1.1.5.pdf>.

Termín zadání: 6.2.2023

Termín odevzdání: 17.5.2023

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

doc. Ing. Petr Fiedler, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato práce se zabývá průmyslovým stacionárním robotem Dobot Magician. První část práce se věnuje teoretickým informacím o těchto robotech, seznamuje čtenáře s různými druhy robotů, s jejich výhodami, nevýhodami a jejich uplatněním v průmyslu. Druhá část práce je věnována praktickým výstupům této práce, tedy hardwarovým návrhem, softwarovým řešením, návrhem vizualizace a demonstrační úloze.

KLÍČOVÁ SLOVA

Dobot, průmyslový robot, statický robot, add-on instrukce, faceplate, zpráva, interface, komunikace, příkaz, header, payload, obrazovka.

ABSTRACT

This master thesis deals with industrial static robot Dobot Magician. First part of this work is dedicated to a theoretical information about such robots, advantages and disadvantages of each type and its use. The second part of this thesis is devoted to the practical outputs of this work, hardware design, software solution, visualization and demonstration of those parts.

KEYWORDS

Dobot, industrial robot, static robot, add-on instruction, faceplate, message, interface, communication, command, header, payload, screen.

NOVÁK, Radim. *Add-On instrukce a Faceplate pro Dobot*. Brno: Brno University of Technology, Fakulta elektrotechniky a komunikačních technologií, Ústav automatizace a měřicí techniky, 2023, 72 s. Diplomová práce. Vedoucí práce: Ing. Radek Štohl, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora: Bc. Radim Novák
VUT ID autora: 211166
Typ práce: Diplomová práce
Akademický rok: 2022/23
Téma závěrečné práce: Add-On instrukce a Faceplate pro Dobot

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

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Radku Štohlvi Ph.D., za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod	13
1 Průmysloví roboti	14
1.1 Stacionární roboti	14
1.2 Kloubový robot	14
1.2.1 Výhody kloubových robotů	14
1.2.2 Nevýhody kloubových robotů	14
1.3 Kartézský robot	15
1.3.1 Výhody kartézských robotů	16
1.3.2 Nevýhody kartézských robotů	16
1.4 SCARA robot	17
1.4.1 Výhody SCARA robotů	17
1.4.2 Nevýhody SCARA robotů	17
1.5 Cylindrický robot	17
1.5.1 Výhody cylindrických robotů	18
1.5.2 Nevýhody cylindrických robotů	18
1.6 Polární robot	18
1.6.1 Výhody polárních robotů	19
1.6.2 Nevýhody polárních robotů	19
1.7 Delta robot	19
1.7.1 Výhody delta robotů:	20
1.7.2 Nevýhody delta robotů	20
1.8 Cobot	20
2 Dobot Magician	23
2.1 Základy práce s Dobotem	23
2.2 Souřadnicové systémy Dobot	24
2.2.1 Kloubový souřadnicový systém	25
2.2.2 Kartézský souřadnicový systém	25
2.3 Ovládání pohybu	26
2.3.1 Jogging mode	26
2.3.2 Point to Point	27
2.3.3 ARC	27
2.4 Parametry Dobot	28
2.5 LED indikace Dobot	29
2.6 Komunikační rozhraní Dobot	29
2.6.1 USB	30

2.6.2	Wi-Fi	30
2.6.3	BLE	30
2.6.4	TTL	30
2.6.5	Interface Dobota	30
3	Zapojení Dobota	33
4	Ovládání Dobota	35
4.1	Formát zpráv odesílaných do Dobota	35
4.2	Příklady příkazů z knihovny Dobota	35
4.2.1	Set Device Name	35
4.2.2	Get Device Name	35
4.2.3	Get Device Version	37
4.2.4	GetPose	37
4.2.5	Get Alarm State	38
4.2.6	Clear All Alarms State	38
4.2.7	Set HOME Cmd	38
4.2.8	Set JOGcmd	38
4.2.9	Set Jog Params	38
4.2.10	Set PTP Cmd	40
4.2.11	Set End Effector Suction Cup	40
4.3	Použité Add-on instrukce	43
4.3.1	Checksum	43
4.3.2	HEX_TO_STRING	44
4.3.3	STRING_TO_HEX	45
4.3.4	TX_V2	46
4.3.5	READ_POSE	47
4.4	Použité rutiny	48
4.4.1	INIT	49
4.4.2	INIT_CHANNEL	52
4.4.3	MainRoutine	52
4.5	Alarmy	54
5	Vizualizace	56
5.1	Main okno vizualizace	56
5.2	Ovládání pohybu	56
5.3	Nastavení JOG parametrů	58
5.4	Nastavení HOME pozice	58
5.5	Nastavení PTP parametrů	59
5.6	PTP help	60

5.7	Ovládání koncového efektoru	60
5.8	Okno alarmů	61
5.9	Komunikace	61
5.10	Nastavení	62
6	Popis demonstrační úlohy	68
	Závěr	69
	Literatura	70
A	Přílohy	72
A.1	Obsah přiloženého CD	72

Seznam obrázků

1.1	Kloubový robot	15
1.2	Portálový jeřáb	16
1.3	Portálový jeřáb	16
1.4	SCARA robot	17
1.5	SCARA robot	18
1.6	Polární robot	19
1.7	Delta robot	20
1.8	Cobot	22
2.1	Dobot	23
2.2	Pracovní prostor Dobota	24
2.3	Pracovní prostor Dobota	25
2.4	Kloubový souřadnicový systém Dobota	26
2.5	Kartézský souřadnicový systém Dobota	26
2.6	MOVJ a MOVL trajektorie	27
2.7	JUMP trajektorie	28
2.8	ARC trajektorie	28
2.9	Interface Dobota	31
2.10	Pinout interfacu	32
3.1	Schéma zapojení	34
4.1	Add-on instrukce pro výpočet checksum	44
4.2	Ukázka převodu v add-on instrukci HEX_TO_STRING	45
4.3	Ukázka převodu v add-on instrukci STRING_TO_HEX	46
4.4	Ukázka add-on instrukce TX_V2	47
4.5	Zpracování polohy Dobota	49
4.6	Ukázka kódu add-on instrukce READ_POSE	50
4.7	Princip jednotlivých částí init rutiny	51
4.8	Ukázka části kódu rutiny INIT_CHANNEL	53
5.1	Main obrazovka vizualizace	57
5.2	Obrazovka s ovládáním pohybů	58
5.3	Pop-up okno s nastavením parametrů JOG pohybu	59
5.4	Pop-up okno s nastavením HOME pozice	60
5.5	Obrazovka s nastavením PTP parametrů	61
5.6	Obrazovka s vysvětláním PTP režimů	62
5.7	Obrazovka s ovládáním koncového efektoru	63
5.8	Obrazovka s alarmy	64
5.9	Obrazovka komunikace	65
5.10	Inicializační stav	66

5.11	Obrazovka Nastavení	67
------	-------------------------------	----

Seznam tabulek

2.1	LED indikace Dobota	29
2.2	Popis pinů komunikačního rozhraní	32
4.1	Příkaz pro nastavení názvu Dobota	36
4.2	Návratová instrukce po nastavení názvu Dobota	36
4.3	Instrukce pro vyčtení názvu Dobota	36
4.4	Návratová instrukce po vyčtení názvu Dobota	36
4.5	Obecný tvar instrukce pro vyčtení verze fw	37
4.6	Obecný tvar odpovědi na instrukci pro vyčtení verze fw	37
4.7	Instrukce pro vyčtení polohy koncového efektoru	37
4.8	Návratová instrukce po vyčtení polohy koncového efektoru	39
4.9	Instrukce pro vyčtení chybových hlášení	39
4.10	Návratová instrukce chybového hlášení	39
4.11	Instrukce sloužící pro smazání chyb	39
4.12	HOME instrukce	39
4.13	Návratová instrukce příkazu HomeCmd	40
4.14	Ukázka instrukce pro nastavení JOG pohybu	41
4.15	Příklad instrukce pro JOG pohyb ve směru osy X+	41
4.16	Ukázka instrukce pro nastavení JOG parametrů	41
4.17	Vyplněná instrukce pro nastavení JOG parametrů	41
4.18	Tabulka JOG parametrů (XYZR)	41
4.19	Ukázka PTP instrukce	42
4.20	Vyplněná PTP instrukce	42
4.21	Souřadnice PTP příkazu	42
4.22	Příklad instrukce pro ovládání suction cup	43
4.23	Vyplněná instrukce pro ovládání suction cup	43
4.24	Příklad převodu souřadnic	49
4.25	Přehled alarmů	55
4.26	Limitní úhly kloubů	55

Úvod

Průmysloví roboti jsou v dnešním průmyslu využíváni čím dál častěji pro jejich obrovské výhody. Ať už se jedná o jejich rychlost, kvalitu jejich práce, zvýšenou bezpečnost pro personál, který se díky nim vyhne potenciálně nebezpečné práci nebo z dlouhodobého hlediska jejich ekonomickou výhodou.

Jedním z druhů těchto robotů jsou takzvaní stacionární roboti. Ti mají svou základnu pevně připevněnou k podložce a pohyb je prováděn pomocí robotického ramena. Stacionární roboty je dále možné dělit na základě uchycení a pohybu ramen. Hovoříme například o kloubových, kartézských, cylindrických, SCARA, polárních a delta robotech.

S rozšiřujícím se trhem robotů se rozšířil i trh zaměřený na školy a studenty, vznikl například právě robot s názvem Dobot Magician, který je využit v této práci. Jedná se o minimalistický model plně funkčního robota, který lze využít na edukativní účely. Nabízí pohyb ve 4 osách, možnost upevnění koncového efektoru pro manipulaci s předměty, 3D tisk a laserové gravírování.

Tato práce by měla čtenáře seznámit s průmyslovými stacionárními roboty, vysvětlit výhody, nevýhody a použití jednotlivých výše popsaných typů robotů. Představit princip komunikace mezi Dobotem a PLC. Objasnit strukturu příkazů a stavbu těchto příkazů. Představit softwarové řešení ovládání Dobotu a představit uživateli vizualizaci sloužící k ovládání Dobotu.

1 Průmysloví roboti

Dobot, jimž se zabývá tato práce je jedním z průmyslových robotů, konkrétněji stacionárních robotů. V této kapitole bude objasněn pojem stacionární robot a popsáno 7 nejčastěji používaných typů robotů.

1.1 Stacionární roboti

Stacionárními roboty myslíme roboty, kteří jsou připevněni k podlaze, stropu nebo jinému povrchu - nejsou mobilní. Obecně jsou stacionární roboti robotická ramena určená pro úlohy, jako je přemísťování objektů, třídění, montáž a svařování.

1.2 Kloubový robot

V průmyslovém prostředí jsou nejběžnějším typem robotů klouboví roboti. Robotické rameno s rotačními klouby je známé jako kloubové robotické rameno. V terminologii robotiky jsou tyto klouby označovány jako osy. Tito roboti jsou obvykle poháněni servomotory a mohou být stejně zásadní jako dvouosá konfigurace nebo stejně složité jako deset nebo více os. Čtyři až šest os jsou v průmyslové robotice běžné, přičemž šest os je nejběžnější.

Tito roboti nabízejí flexibilitu díky široké škále úkolů, které mohou dělat. Obloukové svařování, zpracování materiálu, montáž, přechod komponent, výběr a umístění, balení, plnění systému a paletizace jsou jen některé z dostupných aplikací. Příklad takového robota je na obrázku 1.1. [1]

1.2.1 Výhody kloubových robotů

Šestiosé kloubové roboty lze snadno srovnat do různých rovin, snadno ovládat a spravovat, rychle znovu nasadit pro automatizaci, vstřikování plastů, a to na různých typech a velikostech vstřikovacích strojů. [1]

1.2.2 Nevýhody kloubových robotů

Rychlost těchto robotů je jednou z jejich nevýhod. Nejsou tak efektivní jako jiné druhy robotů, kteří mohou provádět úkoly vysokou rychlostí. Kvůli jejich různým kloubům a stupňům volnosti vyžadují tito roboti pro řízení svého pohybu složitou kinematiku. Mají také vyšší hustotu složek, což vytváří setrvačnou bariéru, kterou je třeba vyřešit jakýmkoli přechodem směru. Pokud je ve studii nákladů a přínosů

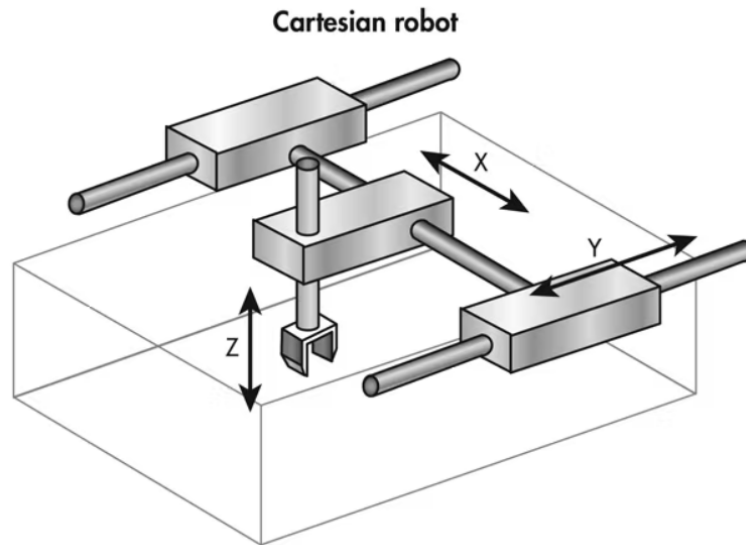
továrny zásadní úvaha o rychlosti, tak tento typ robotů nemusí být tou nejlepší volbou. [1]



Obr. 1.1: Kloubový robot [2]

1.3 Kartézský robot

Kartézský robot, jinak nazývaný také přímočarý nebo portálový, má nejméně tři lineární osy řízení a velmi často se používá při těžkých a precizních operacích. Zvládne přenášení materiálu a hravě si poradí i s výrobou detailních povrchových vzorů. Disponuje třemi lineárními klouby, které používají kartézský souřadný systém (X, Y a Z). Pohyb probíhá po lineárních vodících lištách. Právě díky jejich lineárnímu pohybu se vyznačuje vysokou přesností pohybu, řádově v mikrometrech. Princip robota je na obrázku 1.2 a příklad kartézského robota je na obrázku 1.3. [1]



Obr. 1.2: Obecný kartézský robot [4]

1.3.1 Výhody kartézských robotů

Vysoká přesnost s jednoduchými pohyby, lehké programovatelné, lze využít pro mnoho aplikací, relativně levné. [1]

1.3.2 Nevýhody kartézských robotů

Vyžadují velký pracovní a montážní prostor, složitá montáž, pohyb možný maximálně v jednom směru naráz. [1]



Obr. 1.3: Portálový jeřáb[4]

1.4 SCARA robot

Několikramenné roboty SCARA jsou kompaktní, rychlé, přesné a umožňují precizní manipulaci. V posledních letech se s oblibou využívají k montážním pracím v továrnách. Precizní robotická ruka si hravě poradí s vysokorychlostní montáží, kovááním, balením a jemnou manipulací s různými materiály a nástroji. Roboti SCARA bývají rychlejší než kartézští a mají i menší půdorys. Nejnovější typy poskytují i přes malou velikost a nízkou váhu skvělý výkon, pokročilé řízení trajektorie a perfektní přesnost. Příklad SCARA robota je na obrázku 1.4. [3]

1.4.1 Výhody SCARA robotů

vysoká rychlost, výborná opakovatelnost, na svou velikost je schopný pracovat ve velkém prostoru. [3]

1.4.2 Nevýhody SCARA robotů

vyžadují speciální driver, těžko programovatelní offline, omezen na rovinné plochy. [3]



Obr. 1.4: SCARA robot[3]

1.5 Cylindrický robot

Cylindrický robot je pevně ukotven, nachází se na něm otočný kloub a na jeho rameni je alespoň ještě jeden prizmatický kloub, který umožňuje pohyb pouze v jedné ose. Celkový pohyb robota je lineární a rotační okolo vertikální osy. Jednoduchý design robota umožňuje dostání se do úzkého pracovního prostoru bez nutnosti zpomalení.

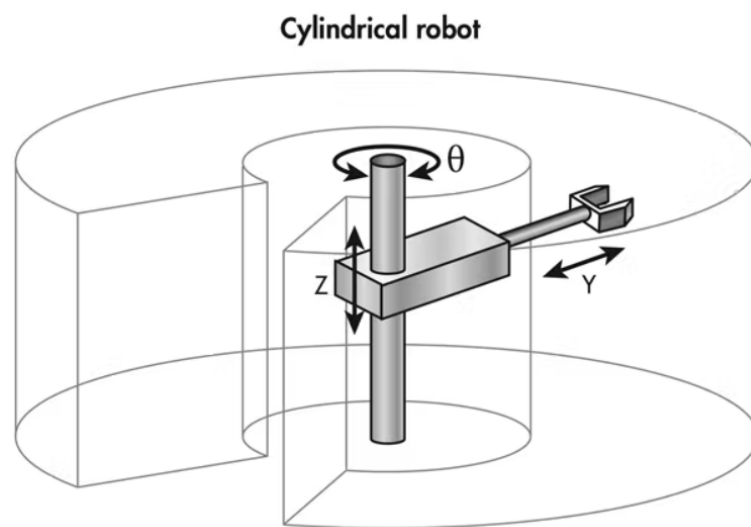
Tento typ robota se nejčastěji využívá v aplikacích, kde je potřeba přemísťovat věci v rámci rotačních pohybů. Příklad cylindrického robota je na obrázku 1.5. [3]

1.5.1 Výhody cylindrických robotů

Jednoduché operace a instalace - minimální montáž, pracovní prostor 360°, zvládá vysoké zatížení. [3]

1.5.2 Nevýhody cylindrických robotů

Vyžadují speciální driver, složité programování, složitá kinematika robota. [3]



Obr. 1.5: SCARA robot[3]

1.6 Polární robot

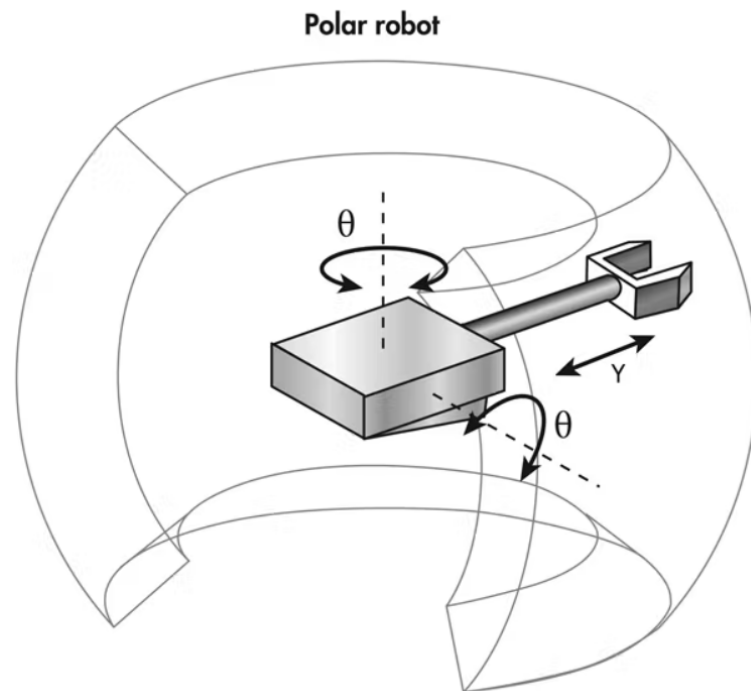
Tento typ robotů je historicky nejstarší, ovšem i přes jeho stáří se dnes stále používá a je vcelku populární. Skládá ze dvou rotačních kloubů a jednoho lineárního. Osy robota jsou navrženy tak, aby tvořily polární souřadnicový systém, díky tomu má robot sférický pracovní prostor. Tento typ robota je hojně využíván při vstřikování, svařování a manipulaci s materiálem. Příklad takového robota je na obrázku 1.6. [3]

1.6.1 Výhody polárních robotů

Rozsah pohybu je 360° , velká škála využití, jeho základna zabírá minimální prostor. [3]

1.6.2 Nevýhody polárních robotů

Malý vertikální dosah, nízká přesnost, nízká opakovatelnost ve směru rotačního pohybu, vyžadují sofistikovaný řídicí systém. [3]



Obr. 1.6: Polární robot[3]

1.7 Delta robot

Delta roboti se běžně skládají ze tří až čtyř ramen, která jsou uprostřed své délky vybaveny kloubem, díky němuž mohou upravovat vzdálenost od těla robota. Vzhledem k jejich designu se někdy nazývají pavoučí roboti. Na konci ramen je upevněný koncový efektor. Ten se pohybuje v osách x , y , z , čímž vytváří cylindrický pracovní prostor. Tento typ robotů se vyznačuje vysokou rychlostí a zrychlením, toho je dosaženo tím, že motory ovládající jednotlivá ramena jsou umístěny v těle robota a ramena si tedy zachovávají svou nízkou hmotnost. Využívají se pro montáže, dávkování, manipulaci s materiálem, který by ovšem měl být lehký. Využití tohoto

robota pro přemísťování těžkého materiálu by znemožnilo využití jeho předností. Příklad delta robota je na obrázku 1.7. [3]

1.7.1 Výhody delta robotů:

Vysoká rychlost, dobré zrychlení, vysoká přesnost. [3]

1.7.2 Nevýhody delta robotů

Jsou navrženi spíše pro lehké pohyby, takže se složitějšími pohyby mají problém, je vyžadován sofistikovaný řídicí systém. [3]



Obr. 1.7: Delta robot[6]

1.8 Cobot

Klasický průmyslový robot je samostatná jednotka, která provádí svou práci na základě pevného programu, bez ohledu na lidi, kteří kolem něj pracují. Svůj pracovní

prostor mají běžně ohraničený ploty, aby lidé do jeho pracovního prostoru nemohli vstoupit a nedošlo tak ke zranění.

Takzvaný kolaborující robot neboli cobot je však speciálně navržen tak, aby spolupracoval s lidským protějškem. Místo toho, aby byl v kleci, pracuje v kooperativním prostředí a pomáhá se složitými úkoly, které nemohou být plně automatizovány. Například mohou odevzdat komponenty lidským spolupracovníkům, kteří provádějí přesnější úkoly při sestavování nebo kontrole kvality.

Mezi výhody cobotů patří obvykle lehká a flexibilní konstrukce. Lze ho snadno přeprogramovat, aby řešil nové úkoly. To umožňuje při výrobě krátké pohyby cobota se zvýšenou účinností. Cobot bývá obvykle navržen tak, aby opakovaně a přesně vykonával pohyb naučený pracovníky. Jejich nejlepší využití spočívá v sestavování více komponent, při sestavování součástí a balení křehkých materiálů. Coboti také mohou plnit úkoly, které by mohly být pro lidi nebezpečné, jako je například doprava ostrých, špičatých nebo horkých obrobků nebo nebezpečné práce se šroubováním. Nasazení cobotů místo klasických robotů má za následek méně nehod a zanechává prostor lidským technikům, aby se zaměřili na méně náročné aspekty výroby.

Tradiční šestiosý průmyslový robot umožňuje v plně automatizovaných aplikacích pracovat rychleji a také zpracovávat těžší předměty. Také tito roboti mohou obvykle pracovat nepřetržitě 24 hodin, 7 dní v týdnu téměř celý rok a plnit tak úkoly, které by na lidského pracovníka kladly velké fyzické nároky. Spolupracující coboti se sice obvykle pohybují pomaleji a mají menší užitečné zatížení než klasický průmyslový robot, ale to je také činí vhodnějšími pro práci s jemnějšími materiály s nižší hmotností.

Být „spolupracující“ znamená pro okolní pracovníky nejen možnost být vedle robota bez ochranných pomůcek, ale také možnost snadné a rychlé optimalizace práce. Proto jsou systémy cobotů běžně vybavovány jednoduchými dotykovými obrazovkami nebo systémem učení založeným na vedením ramene robota, aby se ukázal požadovaný pohyb na obrazovce. Cobot se pak s lidskou obsluhou nijak „nepřetlačuje“, ale podvolí se. Ukázka kolaborativního robota je na obrázku 1.8. [8]



Obr. 1.8: Cobot[9]

2 Dobot Magician

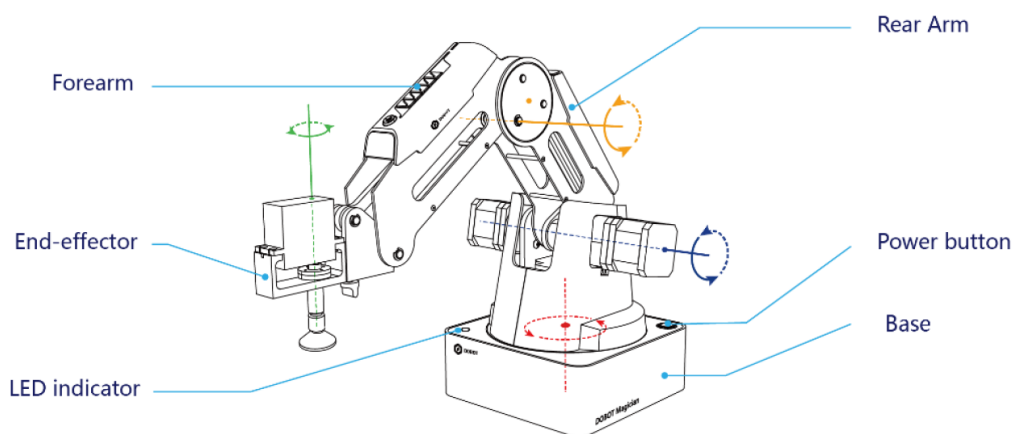
V této kapitole bude popsán Dobot Magician, který bude využit v této práci.

Jedná se o multifunkční robotické rameno navržené převážně pro akademické účely. Lze jej využít například pro 3D tisk, laserové gravírování, psaní, kreslení a přesouvání věcí. Lze jej rozšířit o další I/O zařízení a tím se zvyšují možnosti využití.

Dobot se skládá z několika částí:

- Základny, která je pevně umístěna na pracovní ploše
- Dvou ramen, označované jako rear arm a forearm
- Koncového efektoru

Popis částí Doboty je na obrázku 2.1 a jeho pracovní prostor je v obrázcích 2.1 a 2.3. [10]

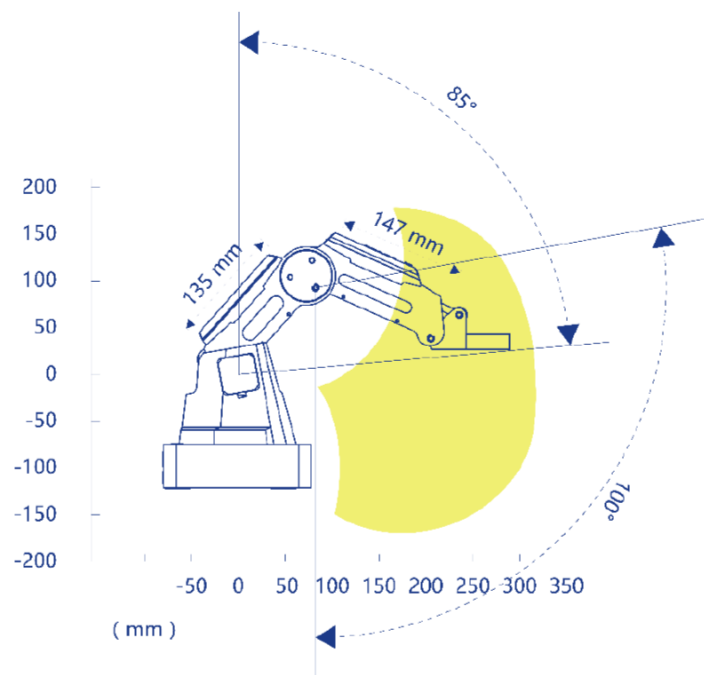


Obr. 2.1: Dobot[10]

2.1 Základy práce s Dobotem

Před uvedením Doboty do provozu je nutné dodržet jisté zásady stanovené výrobcem.

- Před zapnutím musí být koncový efektor v pracovním prostoru mimo krajní pozice. Rear arm a forearm by tedy v ideálním případě měly svírat úhel 45°. Pokud se koncový efektor nachází v krajní poloze je rozsvícená červená LED na základně Doboty.
- Po vypnutí se Dobot vrátí do výchozí polohy, proto je možné zasahovat do pracovního prostoru robota až po zhasnutí LED indikace.
- Odpojení Doboty od přívodu energie je také možné až po zhasnutí LED indikace.



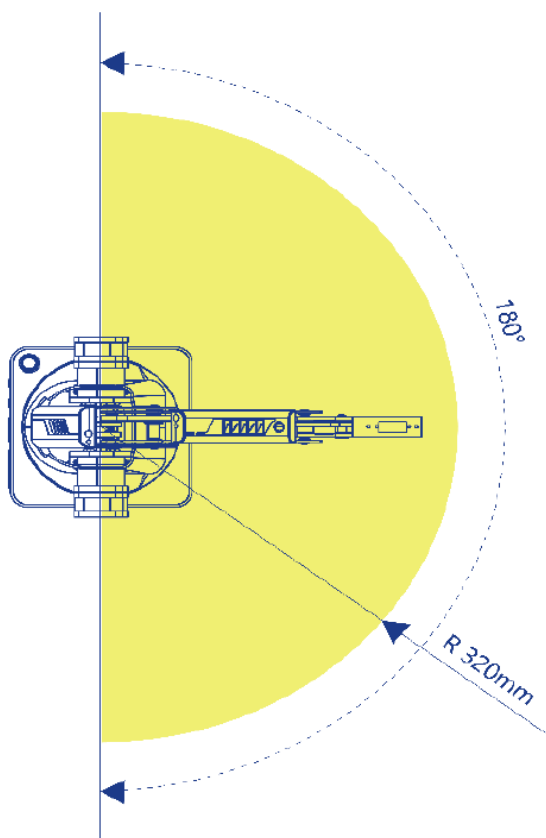
Obr. 2.2: Pracovní prostor Dobot[10]

- Pokud jsou souřadnice Dobot v programu DobotStudio abnormální, zmáčkněte resetovací tlačítko na základně Dobot nebo klikněte na „Home“ v DobotStudio.
- V průběhu resetování se Dobot odpojí od PC a LED indikátor se rozsvítí žlutě. V případě, že reset proběhl úspěšně, tak se zhruba po pěti sekundách LED indikace změní na zelenou.
- Při procesu houmování se Dobot bude otáčet po směru hodinových ručiček do mezních poloh a poté se sám vrátí do houmovací pozice, zároveň bude LED indikace blikat modře. Pokud se po houmování ozve pípnutí a indikátor se změní na zelenou, tak houmování proběhlo v pořádku.
- Před připojením, případně odpojením externího vybavení Dobot vypněte.
- Při používání laseru využijte vhodné bezpečnostní pomůcky.
- Hrot při 3D tisku má teplotu 250 °C.

[10]

2.2 Souřadnicové systémy Dobot

Dobot využívá dva souřadnicové systémy - kloubový a kartézský.



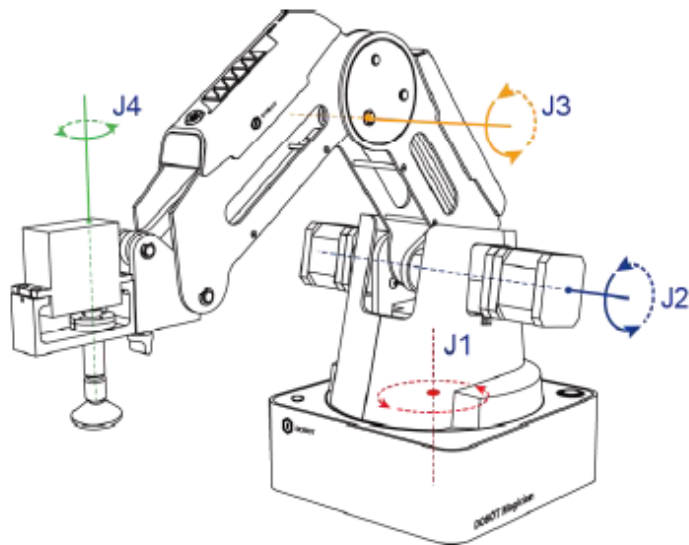
Obr. 2.3: Pracovní prostor Dobot[10]

2.2.1 Kloubový souřadnicový systém

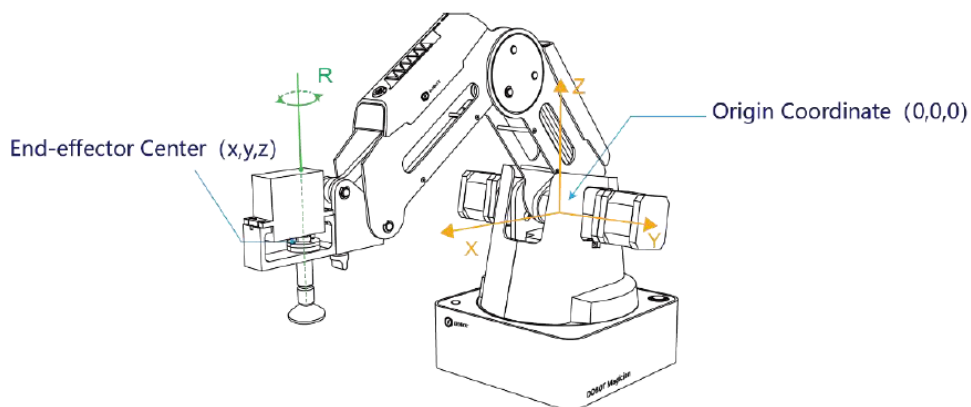
Souřadnice jsou určeny na základě pohybu kloubů. Pokud není připojený koncový efektor má Dobot tři klouby: J1, J2, J3. Kladný směr otáčení těchto kloubů je proti směru hodinových ručiček. Kloubový souřadnicový systém Dobot je na obrázku 2.4. [10]

2.2.2 Kartézský souřadnicový systém

Kartézský souřadnicový systém Dobot je orientován podle základny. Osa X je kolmá na základnu směrem dopředu. Osa Y je kolmá na základnu směrem doleva a osa Z je orientována nahoru. Kartézský souřadnicový systém Dobot je na obrázku 2.5. [10]



Obr. 2.4: Kloubový souřadnicový systém Dobotu [10]



Obr. 2.5: Kartézský souřadnicový systém Dobotu [10]

2.3 Ovládání pohybu

Pohyby Dobotu je možné ovládat ve třech režimech - Jogging, Point to Point, ARC. [10]

2.3.1 Jogging mode

Jedná se o režim, ve kterém se stiskem příslušných tlačítek aktivuje vybraný krokový motor po dobu stisku tlačítka, je tedy vhodný pro manuální polohování. V tomto

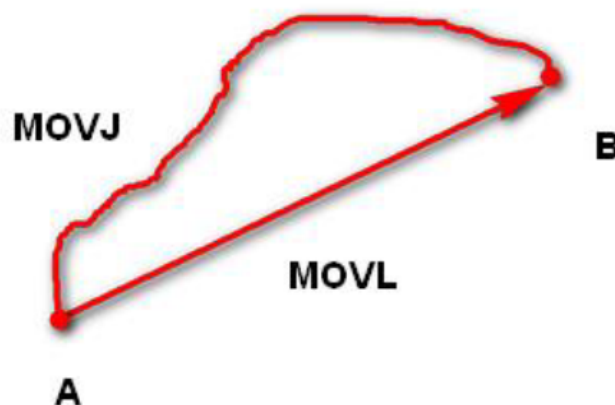
režimu je možné využívat jak kartézský, tak kloubový souřadnicový systém. [10]

2.3.2 Point to Point

Režim Point to Point (PTP) podporuje instrukce MOVJ, MOVL a JUMP, které zajišťují pohyb z výchozí polohy do koncové, trajektorie pohybu poté záleží na vybrané instrukci. [10]

MOVJ Každý kloub vykoná takový pohyb, aby byl v požadovaném úhlu bez ohledu na trajektorii. Využívá se v aplikacích, kde nejsou kladeny požadavky na trajektorii, ale na rychlost. Trajektorie při využití instrukce MOVJ je na obrázku 2.6. [10]

MOVL Vykonává přímočarý pohyb, přesun probíhá po přímkách. Využívá se při požadavcích na přímočarý přesun. Trajektorie při využití instrukce MOVL je na obrázku 2.6. [10]



Obr. 2.6: MOVJ a MOVL trajektorie[10]

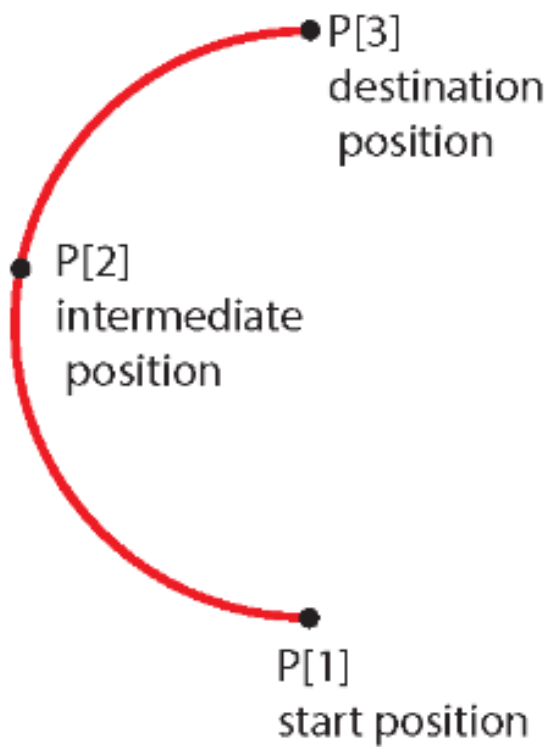
JUMP Pohyb z bodu A do B se vykoná podobně jako je tomu u MOVJ, ovšem postupně, jakoby obepisoval obdélník. Využívá se v aplikacích, kde je potřeba zvedat předměty. Příklad trajektorie při využití příkazu JUMP je na obrázku 2.7. [10]

2.3.3 ARC

Pohyb v ARC režimu je dán třemi body, počátečním, koncovým a libovolným bodem. Trajektorie tedy odpovídá části kružnice. Ukázka ARC trajektorie je na obrázku 2.8. [10]



Obr. 2.7: JUMP trajektorie [10]



Obr. 2.8: ARC trajektorie [11]

2.4 Parametry Dobota

V této kapitole budou popsány technické parametry Dobota Magician

- Maximální zatížení: 500 g
- Maximální dosah: 320 mm
- Rozsah kloubu v základně: -90° - $+90^{\circ}$
- Rozsah kloubu Rear Arm: 0° - $+85^{\circ}$
- Rozsah kloubu Forearm: -10° - $+90^{\circ}$
- Rozsah koncového efektoru: -90° - $+90^{\circ}$
- Maximální rychlost (při zatížení 250 g): $320^{\circ}/s$ pro klouby v ramenu
- Maximální rychlost (při zatížení 250 g): $480^{\circ}/s$ pro kloub v základně
- Opakovatelnost: 0,2 mm
- Napájecí zdroj: 100 V - 240 V AC, 50/60 Hz
- Napájení: 12 V/7 A DC
- Komunikační rozhraní: USB, WIFI, Bluetooth
- I/O: 20 rozšiřitelných rozhraní
- Provozní teplota: -10°C - $+60^{\circ}\text{C}$

[10]

2.5 LED indikace Dobot

V následující tabulce 2.1 budou vyjmenovány jednotlivé stavy LED indikace.

Tab. 2.1: LED indikace Dobot[10]

Status	Popis
Svítil zelená	Dobot pracuje dle zadání
Svítil žlutá	Dobot se zapíná
Svítil modrá	Dobot je v offline režimu
Bliká modrá	Homovací proces
Svítil červená	Dobot je v krajní poloze
	Nevyřešený error
	Připojení příslušenství pro 3D tisk je špatné

2.6 Komunikační rozhraní Dobot

Dobot disponuje čtyřmi možnostmi komunikačních možností.

- USB
- Wi-Fi

- BLE
- TTL

[10]

2.6.1 USB

V této podkapitole budou popsány parametry pro USB komunikaci.

- Baud rate: 115 200 bps
- Data bits: 8-Bit
- Stop bit: 1-Bit
- Parity bit: None

[10]

2.6.2 Wi-Fi

V této podkapitole budou popsány parametry pro komunikaci pomocí Wi-Fi.

- IP: IP pomocí směrování v síti
- COM port: 8899

[10]

2.6.3 BLE

V této kapitole bude popsána komunikace pomocí BLE.

- Service UUID0003CDD0-0000-1000-8000-00805F9B0131.
- UUID adresa portu pro čtení: 0003CDD1-0000-1000-8000- 00805F9B0131.
- UUID adresa portu pro zápis: 0003CDD2-0000-1000-8000- 00805F9B0131.

[10]

2.6.4 TTL

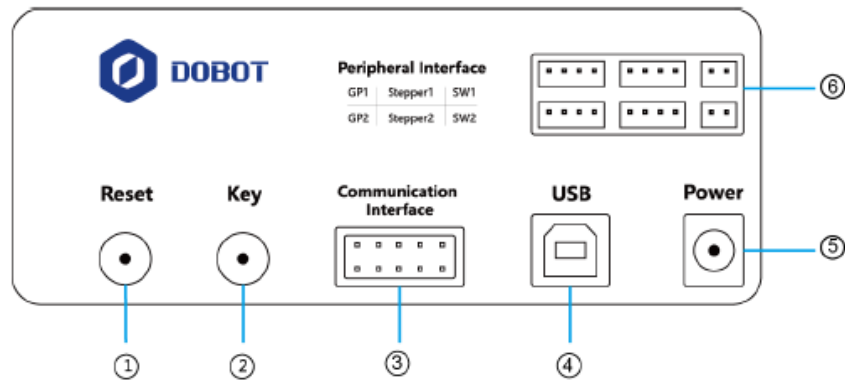
V této podkapitole budou popsány parametry pro TTL komunikaci s dobotem.

- Baud rate: 115 200 bps
- Data bits: 8-Bit
- Stop bit: 1-Bit
- Parity bit: None

[10]

2.6.5 Interface Dobot

V této kapitole bude popsán interface Dobot, který se nachází na jeho základně.

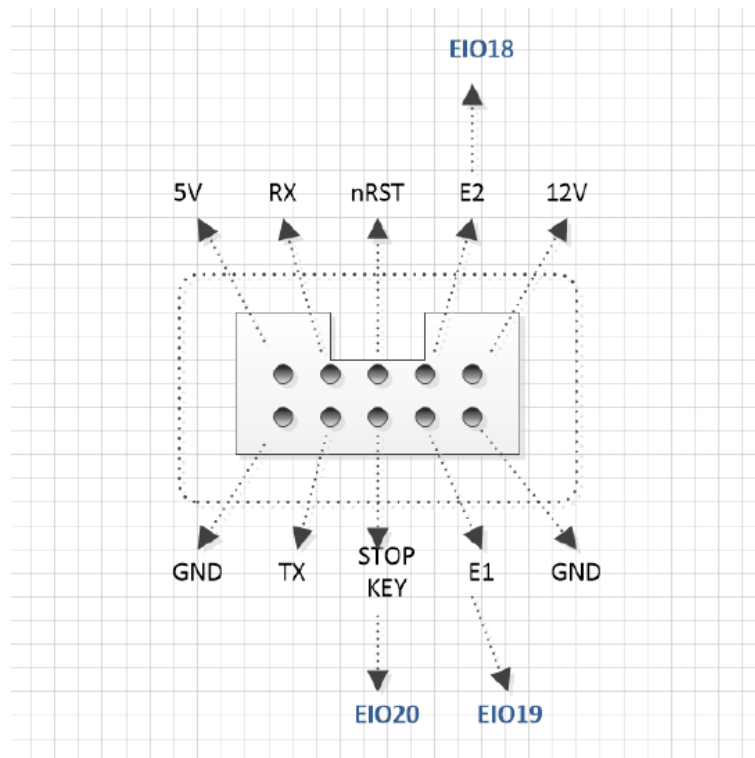


Obr. 2.9: Interface Dobot [10]

1. Reset tlačítko: slouží k restartování mikrokontroléru Dobot. V průběhu restartování se LED indikátor na základně Dobot rozsvítí žlutě na dobu zhruba 5 sekund. Pokud se poté LED indikace změní na zelenou, značí to úspěšný reset.
2. Ovládací tlačítko: krátké stisknutí spustí nahraný program, delší podržení (po dobu 2 sekund) spustí homovací proces.
3. Komunikační rozhraní.
4. USB rozhraní pro připojení k PC.
5. Napájecí port: slouží k připojení napájecího zdroje Dobot.

[10]

Popis jednotlivých pinů v komunikačním rozhraní je na obrázku 2.10 a jejich popis v tabulce 2.2.



Obr. 2.10: Pinout interfacu [10]

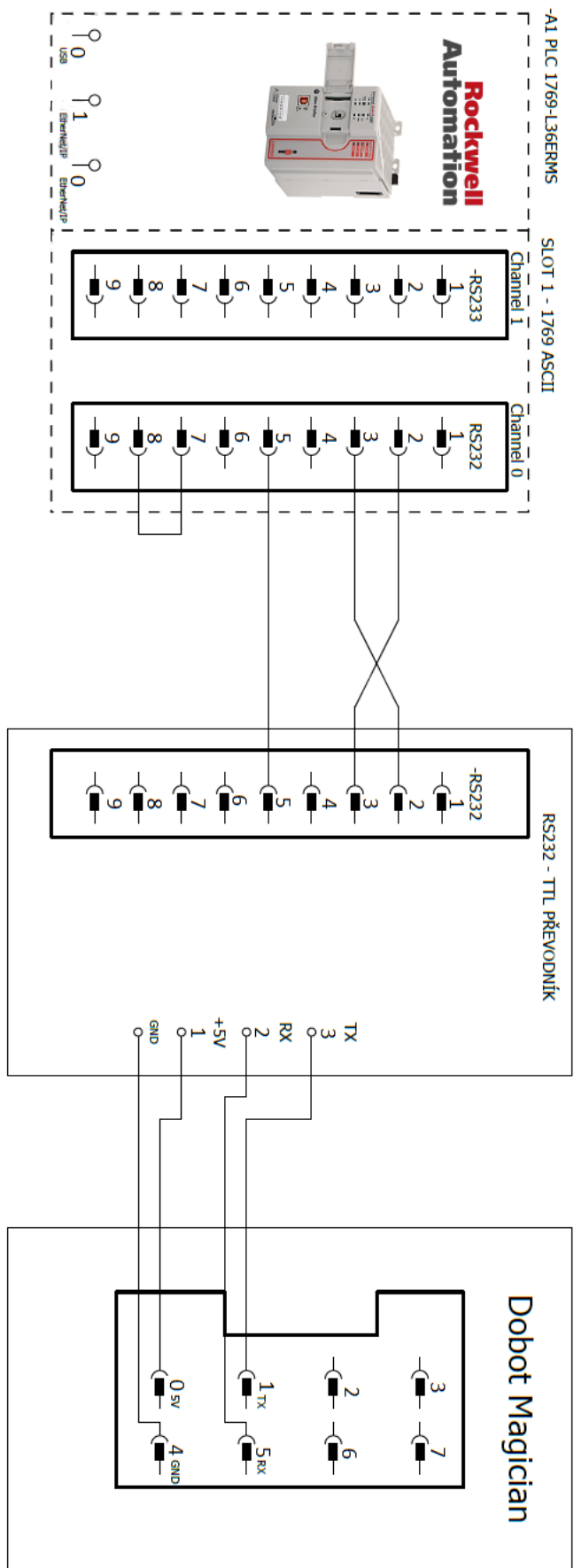
Tab. 2.2: Popis pinů komunikačního rozhraní

Pin	Popis	Výstupní hodnota	Vstupní hodnota
5 V	Napájení	5 V 1 A	-
GND	Zem	-	-
E2	-	3,3 V 20 mA	-
E1	-	-	3,3 V / 5 V 20 mA
nRST	HW Reset	-	3,3 V 20 mA
Stop Key		-	3,3 V / 5 V 20 mA
RX	UART příjem	-	3,3 V / 5 V 20 mA
TX	UART odesílání	3,3 V / 5 V 20 mA	-
12 V	Napájení	12 V 1 A	-
GND	Zem	-	-

3 Zapojení Dobota

Pro tuto práci jsem se rozhodl využít UART (TTL) komunikaci Dobota. Použil jsem komunikační kartu 1769 - ASCII, která je připojená k PLC 1769-L36ERMS ALLEN-BRADLEY. Tato komunikační karta umožňuje odesílání a příjem zpráv rychlostí 115 200 baudů, kterou vyžaduje Dobot.

Modul disponuje dvěma komunikačními porty s konektorem RS232. Aby port fungoval správně, je zapotřebí propojit piny 7 a 8. Pokud toto není provedeno, karta data neodesílá a ani nepřijímá. Jelikož výsledné rozhraní má být UART, je mezi Dobota a komunikační kartu zapojen RS232 - TTL převodník. Jelikož spojení pinů 7 a 8 na straně převodníku způsobuje problémy, je nutné propojit pouze potřebné piny pro komunikaci: RX, TX a GND. Převodník je poté připojen přímo na Dobota, přičemž piny převodníku se na straně Dobota nekříží, RX převodníku je připojeno na RX Dobota, stejně tak TX na TX. Samotné zapojení je na obrázku 3.1.



Obr. 3.1: Schéma zapojení

4 Ovládání Dobota

Ovládání Dobota se provádí pomocí přesně definovaných zpráv. Tyto zprávy jsou odesílány komunikační kartou 1769 - ASCII a generovány jsou pomocí PLC.

4.1 Formát zpráv odesílaných do Dobota

V této kapitole bude popsán pevný formát příkazů sloužící k ovládání Dobota. První část tvoří Header, tím začíná každá odeslaná i přijatá zpráva a jeho obsah je 0xAA 0xAA. Druhou částí je Payload length. Tato část udává počet bajtů, které následují po této části bez započítání checksum. Následuje část zvaná Payload, která se skládá z více částí: ID, CTRL a Params.

Příkazy jsou rozděleny do několika skupin podle toho, čeho se týkají, a právě tyto kategorie jsou označeny příslušným ID. RW označuje, zda chceme do Dobota zapisovat, nebo z něj vyčítat. Pokud chceme Dobotovi zadat například homovací pozici, zvolíme bajt RW jako 1, pokud mu pouze chceme dát příkaz, aby se dostal do homovací pozice, volíme 0.

Bit isQueued se používá v případě, že do Dobota bylo vysláno více příkazů, které zatím nestihl zpracovat, a je nutné další příkazy zařadit do fronty (1 zařazuje do fronty).

Další částí jsou parametry. V případě dotazovacích zpráv zůstává tato část prázdná, v opačném případě jsou v ní vyplněné potřebné parametry pro funkci, kterou chceme vykonat.

Poslední částí je PayloadChecksum, ta je popsána v kapitole zaměřené na add-on instrukci pro jeho spočítání 4.3.2.

4.2 Příklady příkazů z knihovny Dobota

V této kapitole budou uvedeny některé příkazy, které jsou zahrnuty v knihovně Dobota. Veškeré příkazy je možné dohledat v příslušné dokumentaci [12].

4.2.1 Set Device Name

Tyto příkazy se používají k nastavení jména Dobota a formát je v tabulce 4.1. Tvar návratové zprávy je v tabulce 4.2.

4.2.2 Get Device Name

Tento příkaz slouží k vyčtení jména Dobota 4.3.

Tab. 4.1: Příkaz pro nastavení názvu Dobota

Header	Len	Payload				Checksum
		ID	CTRL		Params	
			RW	isqueued		
0xAA 0xAA	Payload lenght	1	1	0	char* DeviceName	Payload Checksum

Tab. 4.2: Návratová instrukce po nastavení názvu Dobota

Header	Len	Payload				Checksum
		ID	CTRL		Params	
			RW	isqueued		
0xAA 0xAA	2+0	1	1	0	Empty	Payload Checksum

Tab. 4.3: Instrukce pro vyčtení názvu Dobota

Header	Len	Payload				Checksum
		ID	CTRL		Params	
			RW	isqueued		
0xAA 0xAA	2+0	1	0	0	Empty	Payload Checksum

Tab. 4.4: Návratová instrukce po vyčtení názvu Dobota

Header	Len	Payload				Checksum
		ID	CTRL		Params	
			RW	isqueued		
0xAA 0xAA	Payload length	1	0	0	char* DeviceName	Payload Checksum

4.2.3 Get Device Version

Tento příkaz slouží pro vyčtení aktuální verze firmwaru, který je nahrán v Dobotovi. Obecný tvar příkazu je v tabulce 4.5. Odpověď je poté v tabulce 4.6.

Tab. 4.5: Obecný tvar instrukce pro vyčtení verze fw

Header	Len	Payload					Checksum
		ID	CTRL		Params		
			RW	isqueued			
0xAA 0xAA	2+0	2	0	0	empty	Payload Checksum	

Tab. 4.6: Obecný tvar odpovědi na instrukci pro vyčtení verze fw

Header	Len	Payload							Checksum
		ID	CTRL		Params				
			RW	isqueued	uint8_t majorVersion	uint8_t MinorVersion	uint8_t revision		
0xAA 0xAA	2+3	2	0	0	uint8_t majorVersion	uint8_t MinorVersion	uint8_t revision	Checksum	

4.2.4 GetPose

Příkaz GetPose slouží k vyčtení polohy koncového efektoru. Ukázka instrukce je v tabulce 4.7 a odpověď je v tabulce 4.8

Tab. 4.7: Instrukce pro vyčtení polohy koncového efektoru

Header	Len	Payload					Checksum
		ID	CTRL		Params		
			RW	isqueued			
0xAA 0xAA	2+0	10	0	0	empty	Payload Checksum	

Struktura tagPose se skládá z parametrů:

- float x; vrací X souřadnici
- float y; vrací Y souřadnici
- float z; vrací Z souřadnici
- float jointAngle[4]; vrací úhly, ve kterých se nacházejí jednotlivé klouby doboty v pořadí: úhel základny, rear arm, forearm, koncový efektor

4.2.5 Get Alarm State

Příkaz sloužící k vyčtení aktuálního chybového hlášení.

Každý bajt seskupuje alarmy, které jsou způsobeny stejnou příčinou a bity v nich signalizují konkrétní alarm.

4.2.6 Clear All Alarms State

Instrukce slouží pro smazání chyb, v tomto případě má stejný formát a parametry i návratová instrukce.

4.2.7 Set HOME Cmd

Po odeslání tohoto příkazu se rameno dobota nastaví do HOME pozice. Struktura HOMEcmd byla vytvořena výrobcem pro budoucí použití, v aktuální verzi je prázdná, je tedy možné jí nahradit nulou.

4.2.8 Set JOGcmd

Tímto příkazem se nastavují pohyby v JOG režimu. V části payload se nastavuje jaký souřadný systém se má použít, zda kartézský, nebo kloubový a poté se volí mezi příkazem pro zastavení a pohyby v osách.

Struktura příkazu je v tabulce 4.14. Skládá z částí nazvaných „isJoint“, kde 0 znamená, že bude použit kartézský souřadný systém a 1 značí kloubový souřadný systém. Druhou částí je „cmd“, ta nabývá hodnot 0 až 8, podle toho, jaký pohyb chceme realizovat. 0 je příkaz pro zastavení, 1 je pro pohyb ve směru X+, respektive J1+, 2 je pro pohyby ve stejných osách, pouze do záporného směru a takto dále se čísla zvyšují přes pohyby pro osu Y, Z a R. Příklad vyplněného příkazu, kde chceme realizovat pohyb v ose X+ je v tabulce 4.15.

4.2.9 Set Jog Params

Tato instrukce slouží pro nastavení parametru rychlosti a zrychlení ve všech kloubech Dobota. Oba parametry se nastavují v procentech a zadávají se v hexadecimální reprezentaci, přičemž každý z údajů reprezentují 4 bajty. Obecný přepis instrukce je v tabulce 4.16. Vyplněná instrukce je poté v tabulce 4.17 a její rozepsané parametry v tabulce 4.18.

Tab. 4.8: Návrátová instrukce po vyčtení polohy koncového efektoru

Header	Len	Payload				Checksum
		ID	CTRL		Params	
			RW	isqueued		
0xAA 0xAA	2+32	10	0	0	typedef struct tagPose	Checksum

Tab. 4.9: Instrukce pro vyčtení chybových hlášení

Header	Len	Payload				Checksum
		ID	CTRL		Params	
			RW	isqueued		
0xAA 0xAA	2+0	20	0	0	empty	Payload Checksum

Tab. 4.10: Návrátová instrukce chybového hlášení

Header	Len	Payload				Checksum
		ID	CTRL		Params	
			RW	isqueued		
0xAA 0xAA	2+16	20	0	0	uint8_t[16]:alarmsState	Checksum

Tab. 4.11: Instrukce sloužící pro smazání chyb

Header	Len	Payload				Checksum
		ID	CTRL		Params	
			RW	isqueued		
0xAA 0xAA	2+0	21	1	0	Empty	Payload Checksum

Tab. 4.12: HOME instrukce

Header	Len	Payload				Checksum
		ID	CTRL		Params	
			RW	isqueued		
0xAA 0xAA	2+1	31	1	1	HOMECmd	Payload Checksum

Tab. 4.13: Návrátová instrukce příkazu HomeCmd

Header	Len	Payload				Checksum
		ID	CTRL		Params	
			RW	isqueued		
0xAA 0xAA	2+8	31	1	1	uint64_t: quededCmdIndex	Checksum

4.2.10 Set PTP Cmd

Tato instrukce slouží pro nastavení PTP pohybu, část payload se skládá z PTP mode a 4 souřadnic / úhlů / přírůstků, podle toho, jaký režim (mod) pohybu je vybrán. Vzhledem k tomu, že Dobot je očekává v hexadecimálním čtyř bajtovém formátu je nutné je na tento formát převést. Režimů PTP pohybu je celkem 10, programově jsou reprezentovány čísly 0 až 9, jedná se o:

0. JUMP_XYZ
1. MOVJ_XYZ
2. MOVL_XYZ
3. JUMP_ANGLE
4. MOVJ_ANGLE
5. MOVL_ANGLE
6. MOVJ_INC
7. MOVL_INC
8. MOVJ_XYZ_INC
9. JUMP_MOVL_XYZ

Princip JUMP, MOVJ A MOVL byl vysvětlen v kapitole 2.3. Pohyby, které mají v názvu „XYZ“ vykonají přesun na zadané souřadnice. V případě těch, které mají v názvu „ANGLE“ se zadává požadovaný úhel kloubů. Poslední skupinou jsou pohyby s „INC“, to značení inkrement, zadává se tedy posunutí vůči aktuálním souřadnicím. Ukázka příkazu je v tabulce 4.19 .

Příklad vyplněného příkazu pro pohyb v režimu JUMP_XYZ je v tabulce 4.20. Vzhledem k tomu, že položka coordinates je příliš dlouhá, je rozepsána ve vlastní tabulce 4.21, v příkazu by byla vyplněna postupně od X po R, s tím, že data by se zapisovala od nejvyššího indexu po nejmenší.

4.2.11 Set End Effector Suction Cup

Tento příkaz slouží pro ovládání suction cupu, což je jeden z druhů koncového efektoru. Ukázka obecného přepisu instrukce je v tabulce 4.22, poté vyplněný příkaz

Tab. 4.14: Ukázka instrukce pro nastavení JOG pohybu

Header	Len	Payload					Checksum
		ID	CTRL		Params		
			RW	isqueued			
0xAA 0xAA	2+2	73	1	1	uint8_t isJoint	uint8_t JOGCmd	Checksum

Tab. 4.15: Příklad instrukce pro JOG pohyb ve směru osy X+

Popis částí	Header		Len	Payload				Checksum	
				ID	CTRL		Params		
					RW	isqueued			
Index	0	1	2	3	4	5	6	7	
Data	0xAA	0xAA	0x04	0x49	0x03	0x00	0x01	0xB6	

Tab. 4.16: Ukázka instrukce pro nastavení JOG parametrů

Header	Len	Payload					Checksum
		ID	CTRL		Params		
			RW	isqueued			
0xAA 0xAA	2+8	72	1	0/1	float Velocity	float Acceleration	Checksum

Tab. 4.17: Vyplněná instrukce pro nastavení JOG parametrů

Popis částí	Header		Len	Payload				Checksum	
				ID	CTRL		Params		
					RW	isqueued			
Index	0	1	2	3	4	5 - 12	13		
Data	0xAA	0xAA	0x0A	0x48	0x03	Params[4.18]	0x20		

Tab. 4.18: Tabulka JOG parametrů(XYZR)

Parametr	Uživatелеm zadaná hodnota [%]	Data 0	Data 1	Data 2	Data 3
Velocity	80	0x42	0xA0	0x00	0x00
Acceleration	60	0x42	0x70	0x00	0x00

Tab. 4.19: Ukázka PTP instrukce

Header	Len	Payload					Checksum
		ID	CTRL		Params		
			RW	isqueued			
0xAA 0xAA	2+17	84	1	1	ptpMode	coordinates	Checksum

Tab. 4.20: Vyplněná PTP instrukce

Popis částí	Header		Len	Payload				Checksum	
				ID	CTRL		Params		
					RW	isqueued			
Index	0	1	2	3	4	5	6-21	22	
Data	0xAA	0xAA	0x13	0x54	0x03	0x00	XYZR[4.21]	0x8D	

Tab. 4.21: Souřadnice PTP příkazu

Část Payload	Hodnota zadaná uživatelem	Data 0	Data 1	Data 2	Data 3
X	174,23	0x43	0x2E	0x70	0x1B
Y	24,24	0x41	0xC1	0xEB	0x90
Z	-8,72	0xC1	0x0B	0x87	0x18
R	11,59	0x41	0x39	0x78	0xD6

pro aktivaci suction cup je v tabulce 4.23. Pro aktivaci se do části „issucked“ zadá 1, pro deaktivaci 0.

Tab. 4.22: Příklad instrukce pro ovládání suction cup

Header	Len	Payload					Checksum
		ID	CTRL		Params		
			RW	isqueued			
0xAA 0xAA	2+2	62	1	0/1	isCtrlEnabled	issucked	Checksum

Tab. 4.23: Vyplněná instrukce pro ovládání suction cup

Popis částí	Header		Len	Payload				Checksum	
				ID	CTRL		Params		
					RW	isqueued			
Index	0	1	2	3	4	5	6	7	
Data	0xAA	0xAA	0x04	0x3E	0x03	0x01	0x01	0xBD	

4.3 Použité Add-on instrukce

V této kapitole budou popsány add-on instrukce, které jsou v práci použity pro práci s Dobotem.

4.3.1 Checksum

Tato add-on instrukce realizuje spočítání checksumu pro odchozí a příchozí zprávy. Jejimi vstupními parametry jsou vstupní data, která se skládají pouze z části ze které se checksum počítá a délka těchto dat. Výstupním parametrem je spočítaný checksum.

Checksum se počítá tak, že v rámci for cyklu se sčítají hodnoty vstupních dat. Tento for cyklus se provádí dokud index, který je využíván na procházení vstupního pole je menší než délka dat mínus jedna. Jednička se odečítá z důvodu, že indexování je od nuly. Po projití for cyklu se od complementu, který odpovídá hodnotě 256, odečte výsledek součtu z for cyklu. Získaný výsledek odpovídá checksumu. Kód add-on instrukce je na obrázku 4.1.

```

3  if index < Data_IN_lenght-1 then
4
5      for index := 0 to Data_IN_lenght-1 do
6          R := R + DATA_IN[index];
7          IF index = Data_IN_lenght then
8              exit;
9
10             end_if;
11         end_for;
12 end_if;
13
14     if index = Data_IN_lenght then
15
16         Complement := 256;
17         Checksum:= Complement - R;
18
19         R:= 0;
20
21         index := 0;
22
23
24     end_if;
25

```

Obr. 4.1: Add-on instrukce pro výpočet checksum

4.3.2 HEX_TO_STRING

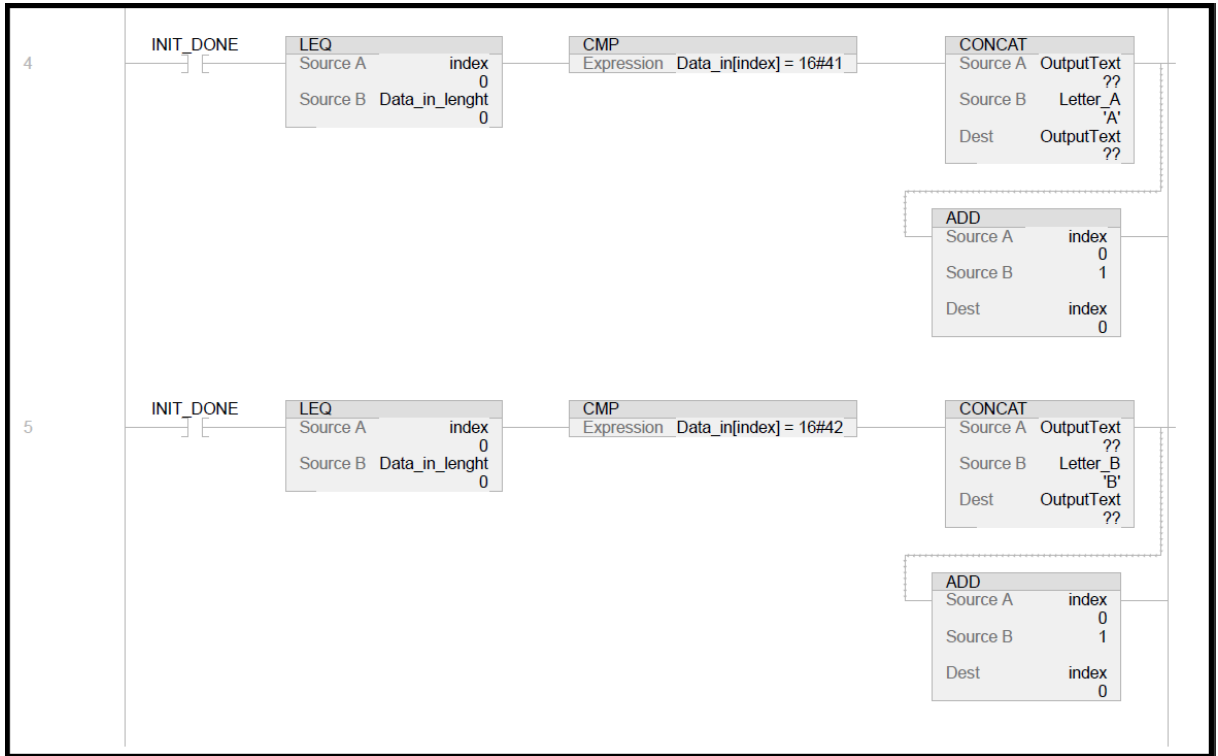
Add-on instrukce HEX_TO_STRING slouží primárně k převodu odpovědi z Dobota obsahující jeho jméno z hexadecimálního tvaru na ascii string. Add-on instrukce má jako vstupní data pole s znaky, jeho délku a bit sloužící k odstartování převodu.

Na nástupnou hranu povolovacího signálu se nastaví první stav stavového automatu, který provede inicializaci indexu sloužícího k procházení pole a smaže obsah výstupní proměnné, do které se ukládá jméno. Následuje proces porovnávání znaků ve vstupním poli s zadanými hodnotami, odpovídajícími příslušným znakům. V případě, že dojde ke shodě je do výstupního stringu přiřazeno odpovídající písmeno a je inkrementován index pro procházení pole. Při každé shodě se nově určený znak sloučí s doposud nalezenými a postupně se složí jméno dobota, dokud index neodpovídá délce vstupních dat.

Příklad porovnávací části add-on instrukce HEX_TO_STRING je na obrázku 4.2. Porovnávání se začíná provádět po tom, co je provedena inicializace proměnných. Jedná se o index pro procházení vstupních dat a je vymazán obsah výstupního

stringu, aby nedošlo k tomu, že nové jméno je kratší než předchozí a některé znaky nebudou přepsány a zůstanou uloženy ve stringu.

Poté začíná proces porovnávání, kdy se hodnota na aktuální pozici porovnává s hexadecimální hodnotou ascii znaku, který je v případě shody přidán do výstupního textu a je inkrementován index.



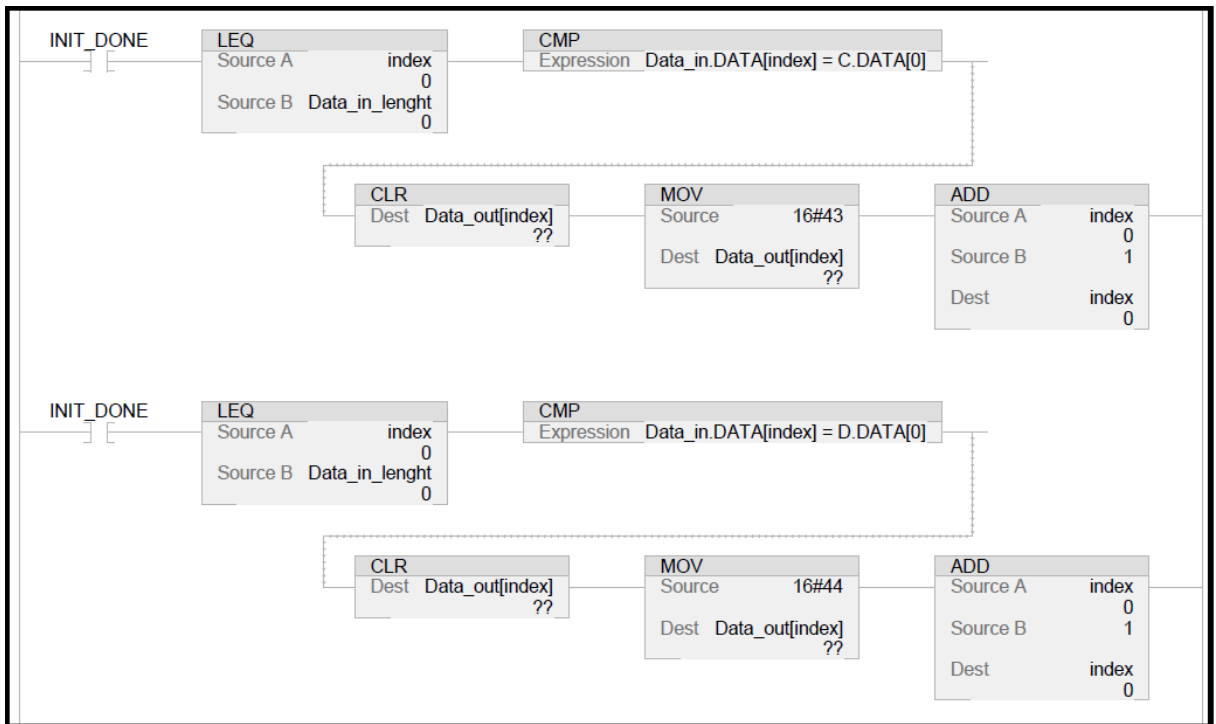
Obr. 4.2: Ukázka převodu v add-on instrukci `HEX_TO_STRING`

4.3.3 `STRING_TO_HEX`

Aby bylo možné nastavit Dobotovi nové jméno je potřeba add-on instrukce pro převod textu datového typu string na hexadecimální znaky. Realizace je obdobná, jako u opačné add-on instrukce `HEX_TO_STRING` 4.3.2, data ve vstupním string řetězci jsou porovnávána s ascii znaky a na základě shody se do výstupního pole nastaví odpovídající hexadecimální hodnota.

Ukázka části kódu je na obrázku 4.3. Opět je podmínkou pro spuštění částí převodu hotová inicializace, zároveň musí být aktuální index menší než délka dat. Poté je pomocí instrukce „`CMP`“ provedeno porovnání. Tato instrukce umožňuje porovnávání proměnných, není možné do ní přímo vložit hodnotu, proto byla vytvořena série proměnných reprezentující ascii znaky s jejich hodnotou. Hodnota je uložena na

multém bajtu v části data. Pro jistotu je po nalezení shody smazán obsah výstupního pole na příslušném indexu a poté vložena nová hodnota a provedena inkrementace indexu.



Obr. 4.3: Ukázka převodu v add-on instrukci STRING_TO_HEX

4.3.4 TX_V2

Add-on instrukce s názvem TX_V2 realizuje odesílání dat do Dobota. Veškeré příkazy do Dobota jsou uloženy v polích o stejné velikosti, z důvodu univerzality, a to velikosti 100. Příkaz ale obsahuje informaci o délce jeho payload části, ta může mít proměnnou délku, zatímco ostatní části zprávy, které nejsou započítány mají vždy stejnou délku, je tedy možné na základě informace o délce části payload dopočítat celkovou délku zprávy.

K délce je přičtena hodnota 4, což je počet bajtů obsazený headrem (2 bajty), právě hodnotu délky části payload (1 bajt) a checksum (1 bajt). Takto vypočtená délka se v rámci inicializace nastaví do proměnné TX_Length, která určuje počet bajtů ve výstupním poli, který má být odeslán do Dobota.

Dále se v rámci inicializace nastavení aktuální hodnota TX_ID do proměnné

„Puvodni_TX“. Je to z toho důvodu, aby v rámci jednoho vykonání instrukce nedošlo k odeslání dat dvakrát.

Po provedení inicializace se začne vykonávat for cyklus, který přesune data z vstupního pole do výstupního. Pokud jsou všechna vstupní data přesunuta, což se pozná dle porovnání indexu a délky příkazu, je nastavený bit pro signalizaci naplnění pole. Pokud je tento bit aktivní a zároveň se rovnají hodnoty v proměnných TX_ID a Puvodni_TX je provedena inkrementace TX_ID a tím je realizováno odeslání zprávy do Dobotu. Ukázka části add-on instrukce TX_V2 je na obrázku 4.4.

```
24
25 // PROJDU CELÉ POLE A DATA NAČTU DO VÝSTUPNÍHO POLE
26
27
28 if index < Command_lenght-1 then
29   for index := 0 to Command_lenght-1 do
30     Command_out[index] := Command_in[index];
31
32     IF index = Command_lenght then
33       exit;
34
35     end_if;
36
37   end_for;
38
39 end_if;
40
41
42
43 if index = Command_lenght AND Puvodni_TX = Tx_index then
44   Output_array_filled := 1;
45
46
47 else
48   Output_array_filled := 0;
49
50 end_if;
51
52
53 // NA NÁSTUPNOU HRANU NAPLNĚNÍ ZVÝŠÍM INDEX A TÍM ODEŠLU DATA
54 if OSRI1.OutputBit then
55   Tx_index := Tx_index +1;
56
57   DONE:=1;
58 else
```

Obr. 4.4: Ukázka add-on instrukce TX_V2

4.3.5 READ_POSE

Jak název napovídá add-on instrukce READ_POSE realizuje vyčítání polohy, ne samotné odeslání příkazu pro návrat polohy, ale zpracování návratové zprávy. Vstupními parametry do této instrukce je povolovací bit, pole obsahující odpověď na dotaz o poloze a výstup tvoří kartézské souřadnice X,Y,Z a kloubové J1,J2,J3 doplněné o příznaky DONE a ERROR.

Při nástupné hraně povolovacího bitu nastavíme stavový automat do prvního stavu, ve kterém inicializuje indexy a resetuje bity značící dokončení převodu polohy. Tím, že zprávy mají přesně daný tvar, víme, na jakém indexu očekáváme jaká identifikační data. Zpracování dat probíhá způsobem popsáním na obrázku 4.5. Po splnění podmínek popsanych na obrázku 4.5 se nastaví indexy, na kterých začínají data odpovídající jednotlivým souřadnicím. Ty Dobot odesílá jako čtveřici bajtů, které reprezentují hodnotu typu REAL. PLC realizuje konverzi z čtyřech bajtů na jednu float hodnotu pomocí instrukce COP a pokud proběhne úspěšně pro všechny souřadnice, tak se nastaví výstupní signál Done. Pokud se instrukci nepodaří najít očekávaná data, což znamená, že index pro procházení pole bude mít větší hodnotu než 85, nastaví se stavový automat do error stavu, kde se nastaví ERROR bit.

Příklad zpracování polohy

Obdržená zpráva z Dobot:

- Header: 0xAA 0xAA
- Len: 0x22
- ID: 0x0A
- CTRL: 0x00
- Payload: 0x1B 0x70 0x2E 0x43 0x90 0xEB 0xc1 0x41 0xC1 0x41 0x18 0x87 0x0B 0xC1 0xD6 0x78 0x39 0x41 0x8F 0x28 0xFD 0x40 0x60 0x00 0xBA 0x41 0xC7 0x23 0x81 0x42 0xCA 0x92 0x6B 0x40
- Checksum: D7

Každá čtveřice části Payload vyjadřuje hodnotu jedné souřadnice v pořadí X,Y,Z,R následované úhly kloubů J1,J2,J3,J4.

Převod z hexadecimální hodnoty na real typu float se provádí jako Little Endian - na paměťové místo s nejnižší adresou uloží nejméně významný bajt (LSB) a za něj se ukládají ostatní bajty až po nejvíce významný bajt (MSB).

Data jsou tedy přeskládána a při převodu se berou od konce, pro X jsou tedy v pořadí 0x43 0x2e 0x70 0x1b. Výsledek převodu je v tabule 4.24 .

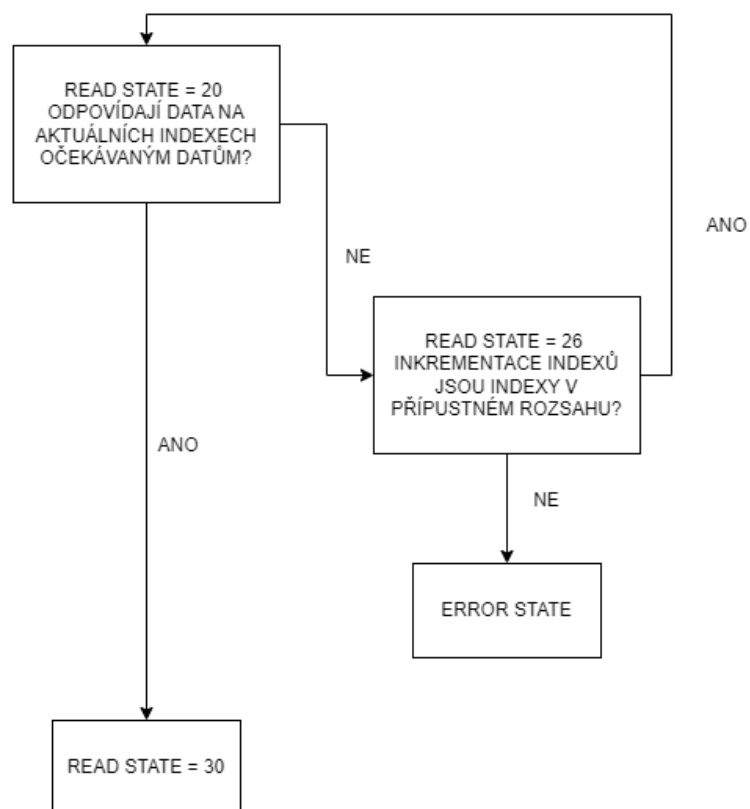
Ukázka části kódu add-on instrukce READ_POSE je na obrázku 4.6. V tomto obrázku můžeme vidět přiřazení částí vstupní zprávy do proměnných pro výpočet konkrétních souřadnic, na dalších řádcích je poté realizovaná konverze na správný datový typ.

4.4 Použité rutiny

Tato kapitola popisuje funkcionalitu použitých rutin v programu.

Tab. 4.24: Příklad převodu souřadnic

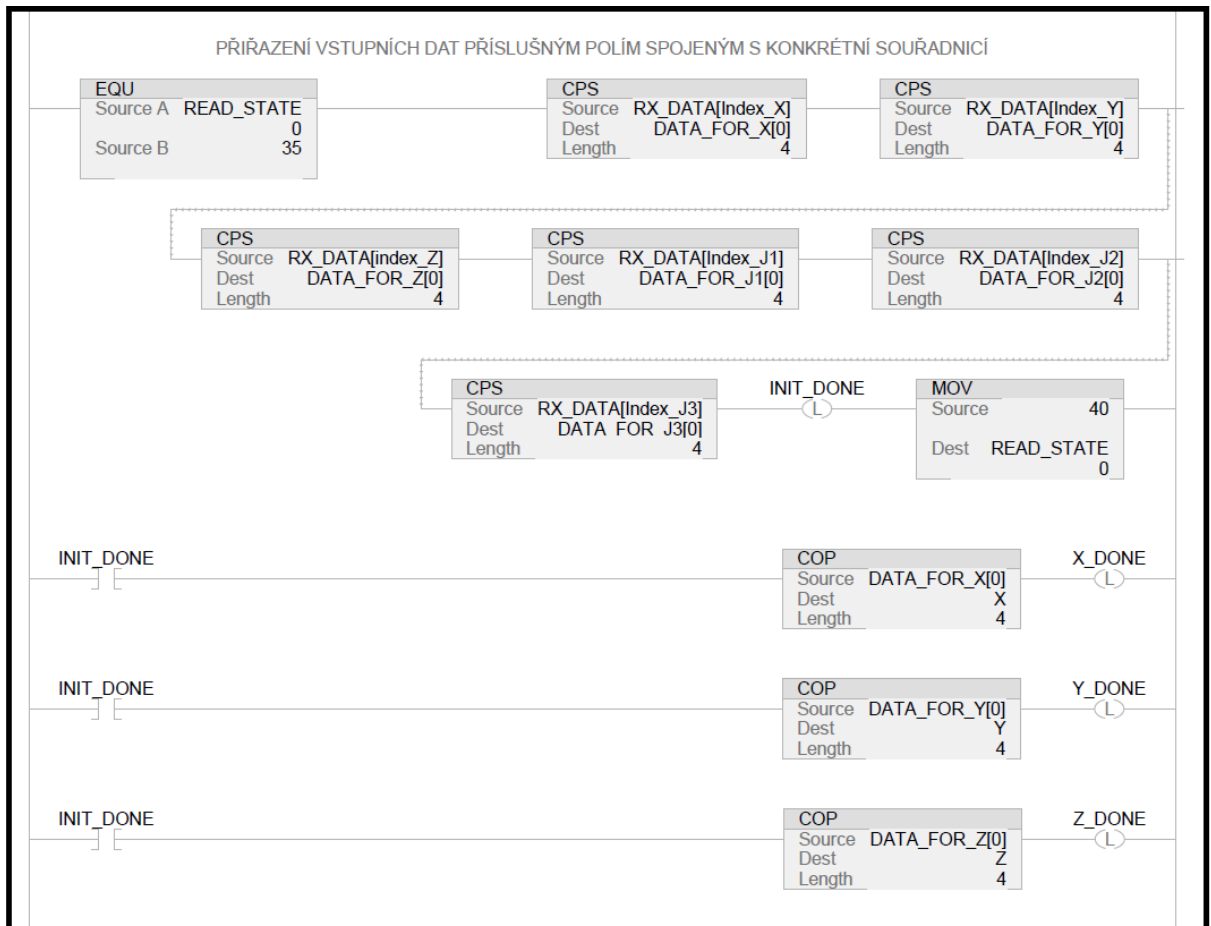
Souřadnice	Data 0	Data 1	Data 2	Data 3	Výsledek převodu
X	0x43	0x2E	0x70	0x1B	174,43
Y	0x41	0xC1	0xEB	0x90	24,24
Z	0xC1	0x0B	0x87	0x18	-8,72
R	0x41	0x39	0x78	0xD6	11,59
J1	0x40	0xFD	0x28	0x8F	7,9
J2	0x41	0xBA	0x00	0x60	23,25
J3	0x42	0x81	0x23	0xC7	64,56
J4	0x40	0x6B	0x92	0x3A	3,68



Obr. 4.5: Zpracování polohy Dobota

4.4.1 INIT

Rutina INIT slouží pro nastavení komunikačního kanálu a načtení dat z Dobota do programu. skládá se z několika kroků, přičemž se jedná o:



Obr. 4.6: Ukázka kódu add-on instrukce READ_POSE

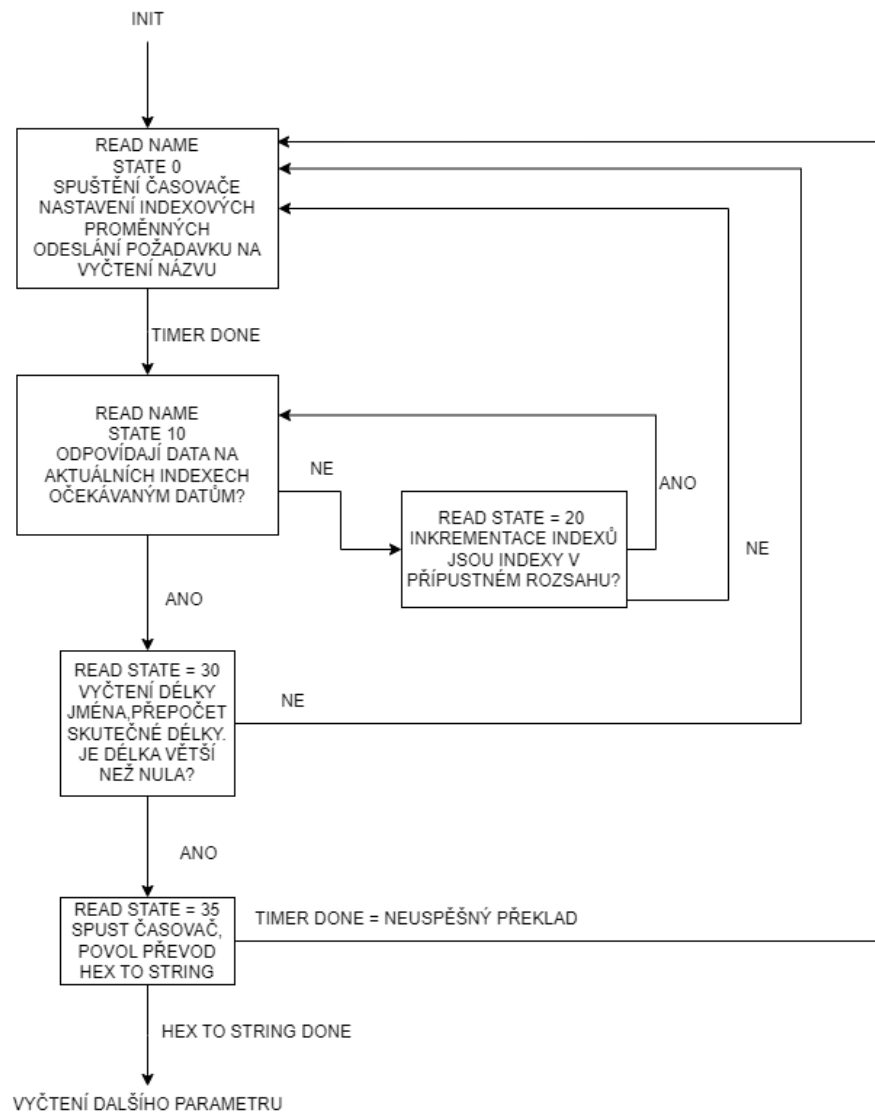
1. nastavení komunikačního kanálu
2. vyčtení jména
3. vyčtení verze firmwaru
4. home poloha
5. parametry jog pohybu
6. parametry ptp pohybu
7. parametry ptp jump pohybu
8. aktuální polohu

Princip vyčítání těchto parametrů je shodný s tím, který je popsán na obrázku 4.5. Přejde se do stavu, kde se inicializují indexy a spustí časovač, je to kvůli tomu, aby nedocházelo k rychlému přepínání povolovacího bitu, které nebude zaznamenáno. Po uplynutí času v časovači se nastaví povolovací bit pro odeslání dotazu do Dobotu.

Následuje prohledávání odpovědi, zda se v ní nacházejí očekávaná data (header

a ID příslušného dotazu), pokud ne, provede se inkrementace indexu a hledání dat v přijaté zprávě pokračuje dál. Kontroluje se, zda index není větší, než hodnota, která by dávala smysl při procházení pole o 100 prvcích. Pokud je tato hodnota překročena, proběhne návrat do stavu, kdy se posílá dotaz do Dobota. Pokud je index menší než nastavená mez, tak se program vrátí k porovnání. Pokud tentokrát data na aktuálních indexech odpovídají, přechází se do dalšího stavu, kde se přes pomocné proměnné, které slouží k určení začátku dat tyto data uloží a proběhne jejich zpracování.

Pokud je celý tento proces úspěšný, tak se posunu o položku k vyčtení dál. Ukázka stavového automatu pro část vyčítání jména je na obrázku 4.7.



Obr. 4.7: Princip jednotlivých částí init rutiny

4.4.2 INIT_CHANNEL

Tato rutina slouží pro inicializaci příslušného kanálu, nastaví na požadovaný kanál baudrate 115 200 baudů. Tato hodnota je zastoupena decimální hodnotou 7 a je přiřazena do proměnné ChxBaudRate (x nahrazené číslem kanálu). Dále je zakázán ByteSwapMode nastavením hodnoty 0 do proměnné ChxRxByteSwapMode. Další parametry se týkají ukončovacích znaků pro komunikaci (EndDelimiter).

Jak bylo zmíněno v předchozí kapitole, formát zpráv pro Dobotu má pevný tvar a ukončovací znaky se nepoužívají, proto je hodnota proměnné ChxRxEndDelimiter nastavena na hodnotu 39, což značí, že se mají ukončovací znaky v přijatých zprávách ignorovat a do odesílaných zpráv nepřidávat. Stejný důvod pro nastavení vynechání ukončovacího znaku je i u proměnné ChxRxEndMode. Předchozí nastavení zůstává stejné pro totožné proměnné pro TX data.

Důležitým parametrem pro příjem zpráv je Handshake mode, ten je nastavený na 1, čímž je povolen a zajišťuje, že při každém příjmu nových dat bude inkrementována proměnná RxID. Aby karta poznala, že příjem dat je hotový, je nutné použít parametr MessageTimeout. Protože zprávy nemají ukončovací znak, karta by nikdy nepoznala, že přenos dat je ukončen a k inkrementaci RxID by nedošlo.

MessageTimeout slouží pro nastavení času od příjmu prvního bajtu po ukončení příjmu. Tento čas byl vyzkoušen i s nejdelšími možnými příkazy a inicializační blok ho nastaví na dobu 1 ms. Karta umožňuje nastavení ještě několika parametrů, ale ty není potřeba nastavovat, jelikož byly zakázány jiným provedeným nastavením. Ukázka kódu rutiny INIT_CHANNEL je na obrázku 4.8.

4.4.3 MainRoutine

Samotné ovládání Dobotu se provádí z MainRoutine. Je zde realizováno přiřazování TX a RX proměnných komunikační karty podle vybraného kanálu. Příjem dat se skládá ze dvou částí, v první se inkrementuje proměnná RxID. Jakmile k tomu dojde, víme, že v RxData jsou nová data a v RxDataLenght je uloženo, kolik bajtů tato zpráva má. Kvůli parametru MessageTimeout, který byl popsán v kapitole 4.4.1, nemusí délka přijaté zprávy odpovídat délce zprávy odeslané Dobotem. Díky tomu, že zpráva z Dobotu v sobě skrývá i informaci o její délce, je možné vzít pouze užitečnou část přijaté zprávy.

Z MainRoutine se také po první spuštění plc nebo po pokynu pro inicializaci na základě vybraného kanálu spustí rutina Init, která dále spustí rutinu INIT_CHANNEL. V průběhu inicializace je ostatní ovládání Dobotu pozastaveno a není možné posílat příkazy, jejich fungování je popsáno v kapitolách 4.4.1 a 4.4.2.

Provádí se zde převod odeslaných a přijatých dat na string, který je dále využit ve vizualizaci, aby měl uživatel přehled o tom, jaká data odesílá a jaká přijímá.

```

1 SBR(Channel);
2
3
4 if Channel =0 then
5
6 Ch0BaudRate:= 7;
7 Ch0RxByteSwapMode:=0;
8 Ch0RxEndDelimiter:=39;
9 Ch0RxEndMode:=0;
10 Ch0RxHandShakeMode:=1;
11 Ch0RxMessageTimeout := 1;
12 Ch0RxPadChar:=39;
13 Ch0RxStartDelimiter:=39;
14 Ch0TxByteSwapMode :=0;
15 Ch0TxEndDelimiter:=39;
16 Ch0RxEndMode:=0;
17
18 TX_ID:=Local:1:0.Ch0TxID;
19
20 RX_ID:= Local:1:I.Ch0RxID;
21 RX_ID_Old:= Local:1:I.Ch0RxID;
22
23 end_if ;
24

```

Obr. 4.8: Ukázka části kódu rutiny INIT_CHANNEL

Ovládání pohybu je spouštěno z vizualizace, proměnné, které se využívají ve vizualizaci mají ve jméně „Visu“ pro větší přehlednost. Po stisku tlačítka pro příslušný pohyb se nastaví povolovací bit do TX add-on instrukce, která do Dobotu odešle příkaz pro zahájení pohybu. Jedná se o ovládání hlavně JOGovacího pohybu, takže dokud je tlačítka stisknuto Dobot vykonává pohyb. Toho je docíleno hlídáním sestupné hrany tlačítka z vizualizace a jakmile je puštěno, tak se do Dobotu posílá příkaz pro zastavení pohybu.

Začátkem pohybu se také spouští proces vyčítání polohy Dobotu. Tato podmínka platí jak pro JOG režim, tak pro PTP. Při JOG režimu ukončuje proces vyčítání polohy uvolnění tlačítka a u PTP pohybu je proces vyčítání polohy ukončen přesunem do nastavené polohy. V případě JOG režimu je ještě po ukončení pohybu jednou zopakováno vyčtení, aby poloha opravdu odpovídala současné poloze.

Dále se zde realizuje sestavování příkazu, v programu označované jako „CMD“, jedná se o příkazy, jejichž parametry se nastavují ve vizualizaci. Parametry z vizualizace jsou převedeny do správného formátu a je spočítán checksum, který je do zprávy přidán. Takto sestavený příkaz je připraven k odeslání do Dobotu, což se opět realizuje z vizualizace.

Probíhá zde proces nastavení nového jména, který se skládá právě ze složení příkazu, jeho odesláním a následným vyčtením jména pro kontrolu a jeho nastavení do správných proměnných.

Dobot má tu nevýhodu, že v případě erroru o tom sám nezašle informaci, ale

je potřeba jí vyčíst. Proto je v MainRoutině realizováno cyklické vyčítání alarmů a jejich zpracování. Je zde možné také odeslat příkaz pro smazání errorů, pokud jejich příčiny byly odstraněny.

4.5 Alarmy

Součástí práce je také řešení chybových stavů Dobota. Jejich seznam a popis je k nalezení v tabulce 4.25. Programově chybový stav nezamezí běhu programu, je to z důvodu, že většina chyb vyžaduje jejich nápravu skrze program. Jedná se buď o chybně nastavené parametry, které je v případě chyby nutné nastavit znovu nebo pohyb mimo pracovní prostor Dobota a řešením je tedy pohyb v opačném směru.

Dobot sám od sebe neodešle informace o chybě v případě, že se chybový stav objeví. Proto je v programu realizováno cyklické vyčítání alarmů. Po obdržení návratové instrukce na dotaz o alarmech je opět použita logika popsaná na obrázku 4.5. Hledá se header a ID zprávy, po nalezení jsou dopočítány indexy, na kterých se nachází část zprávy payload. Payload se skládá z osmi bajtů, přičemž každý bit jednotlivých bajtů signalizuje jeden error. Při jakémkoliv z alarmů je nastavená hlavní proměnná signalizující chybu do stavu true. Ve vizualizaci se alarmy okamžitě zobrazují v příslušném okně, aby o nich byl uživatel informován.

Popis některých chyb:

Neznámá instrukce pro Dobota: Zpráva obsahuje header, má správný tvar, pouze neznáme ID.

Koncový bod pro pohyb je abnormální: Možné příčiny chyby

- Cílový bod v režimu MOVL je zadaný chybně
- Cílový nebo prostřední bod v režimu ARC je zadaný chybně
- Cílový bod v režimu CP je zadaný chybně
- Cílový bod v režimu JUMP_MOVL je zadaný chybně

Chyby související s krajními polohami: krajní polohy jednotlivých úhlů jsou popsány v tabulce 4.26 .

Chybné JUMP parametry: Zadaná výška je záporná

Chyba polohování krokového motoru v kloubech: tato chyba je způsobena například nárazem Dobota do předmětu.

Po odstranění příčiny erroru dá uživatel z vizualizace dá příkaz pro odeslání příkazu clear alarm. Pokud již není aktivní příčina vzniku chyby, jsou alarmy vyresetovány.

Tab. 4.25: Přehled alarmů [13]

Index hex	Index dec	Adresa chyby	Popis chyby	Řešení chyby
0x01	1	Data[0].1	Neznámá instrukce pro Dobota	Odeslání správné instrukce + reset
0x02	2	Data[0].2	Systémová chyba	Restart Dobota
0x03	3	Data[0].3	Chyba komunikace mezi MCU a FPGA	Restart Dobota
0x04	4	Data[0].4	Chyba úhlového senzoru	Restart Dobota
0x10	16	Data[2].0	Koncový bod pro pohyb je abnormální	Odeslání správného bodu + reset
0x11	17	Data[2].1	Cílový bod je mimo pracovní prostor	Odeslání správného bodu + reset
0x12	18	Data[2].2	Inverzní úloha nemá řešení	Odeslání správného bodu + reset
0x13	19	Data[2].3	Opakující se body v ARC nebo JUMP_MOVEL pohybu	Odeslání správných bodů + reset
0x15	21	Data[2].5	Chybné JUMP parametry	Nastavení platných parametrů + reset
0x20	32	Data[4].0	Singularita	Pohyb v kloubovém režimu mimo singularitu
0x21	33	Data[4].1	Požadovaný pohyb vede mimo pracovní prostor	Nastavení platného pohybu + reset
0x22	34	Data[4].2	Pohyb do koncového bodu není možný žádným druhem pohybu	Nastavení platného koncového bodu + reset
0x30 - 0x33	48- 51	Data[6].0 - Data[6].3	Příliš vysoká rychlost kloubu (0x30 = J1, 0x33 = J4)	Nastavení platné rychlosti + reset
0x40 - 0x47	64- 71	Data[8].0 - Data[8].7	Pohyb je v limitní poloze (0x40=J1 v +maximu, 0x41 J1 v -maximu)	Opačný pohyb
0x48 - 0x49	72- 73	Data[9].0 - Data[9].1	Chyba paralelogramu	Zmenšit velikost kloubových souřadnic
0x50 - 0x53	80- 83	Data[10].0 - Data[10].3	Chyba polohování krokového motoru v kloubech (0x50 = J1, 0x53 = J4)	Proces homování

Tab. 4.26: Limitní úhly kloubů

Kloub	Maximální záporný úhel	Maximální kladný úhel
J1	-90°	90°
J2	0°	85°
J3	-10°	90°
J4	-90°	90°

5 Vizualizace

Jedním z cílů práce bylo vytvořit vizualizaci, která by umožňovala ovládání Dobota, jak jeho pohyby, tak nastavování parametrů. Za tímto účelem byla vytvořena vizualizace, skládající se z několika obrazovek, které budou popsány v následujících kapitolách.

V obrazovkách je použit jednotný koncept návrhu, po stránce designu a volby barev prvků. Šedé podbarvení mají aktivní tlačítka, která vyvolávají funkce při kliknutí na ně. Bílá textová pole mimo horní menu, polohu koncového efektoru a nadpisy v jednotlivých sekcích slouží k zadávání parametrů uživatelem. Žlutě podbarvená tlačítka signalizují nápovědu.

5.1 Main okno vizualizace

Při zapnutí vizualizace se načte okno s názvem main, slouží jako domovská obrazovka, ze které je možné přecházet na další obrazovky, podle toho, jakou akci chce uživatel provést.

Obrazovka se skládá z několika částí, označených na obrázku 5.1. Význam jednotlivých částí:

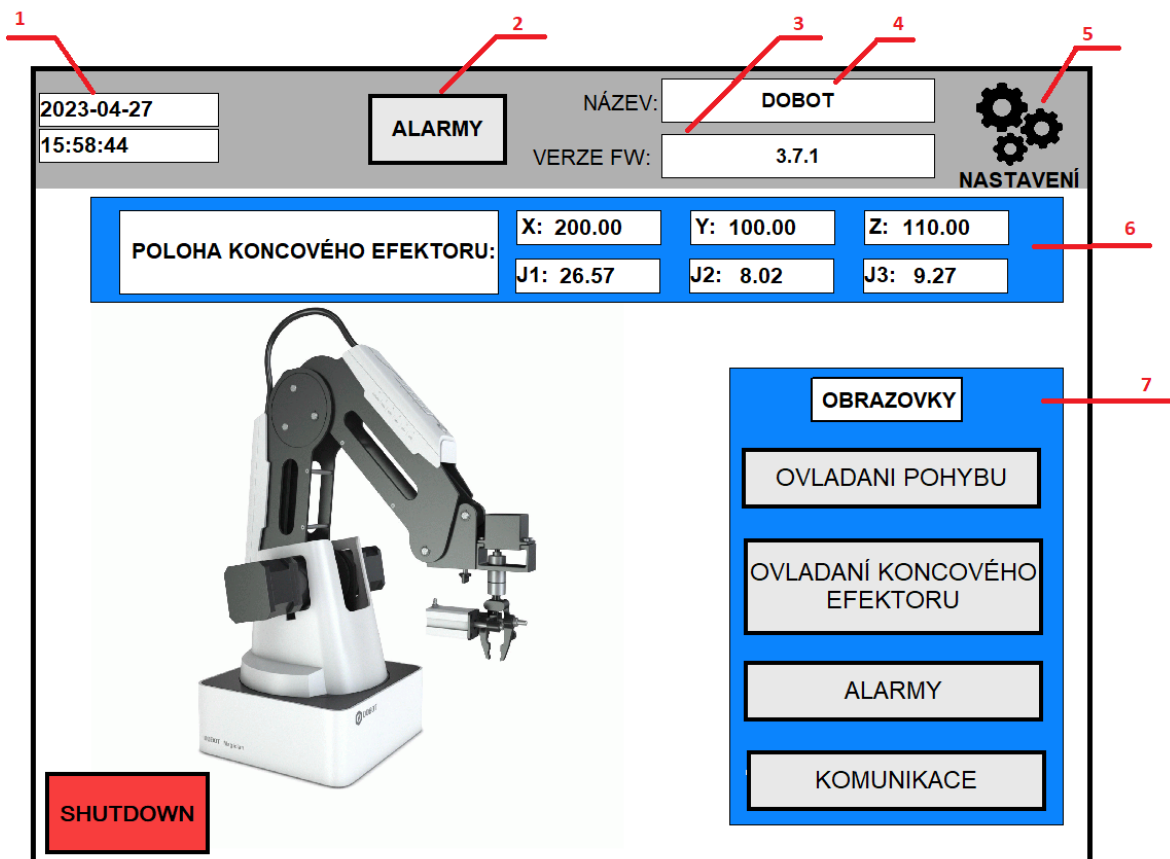
1. Aktuální čas
2. Přechod na obrazovku alarmů
3. Verze aktuální firmwaru nahraného v Dobotovi
4. Název Dobotu
5. Přechod do nastavení Dobotu
6. Aktuální poloha, jak v kartézských souřadnicích (X,Y,Z), tak kloubových (J1,J2,J3)
7. Tlačítka pro přechod na další obrazovky

5.2 Ovládání pohybu

Tato obrazovka slouží k ovládání pohybů Dobotu, najdeme na ní ovládání pro JOG režim, PTP režim, ovládání pro příkazy spojené s home pozicí Dobotu a části již popsané v kapitole 5.1.

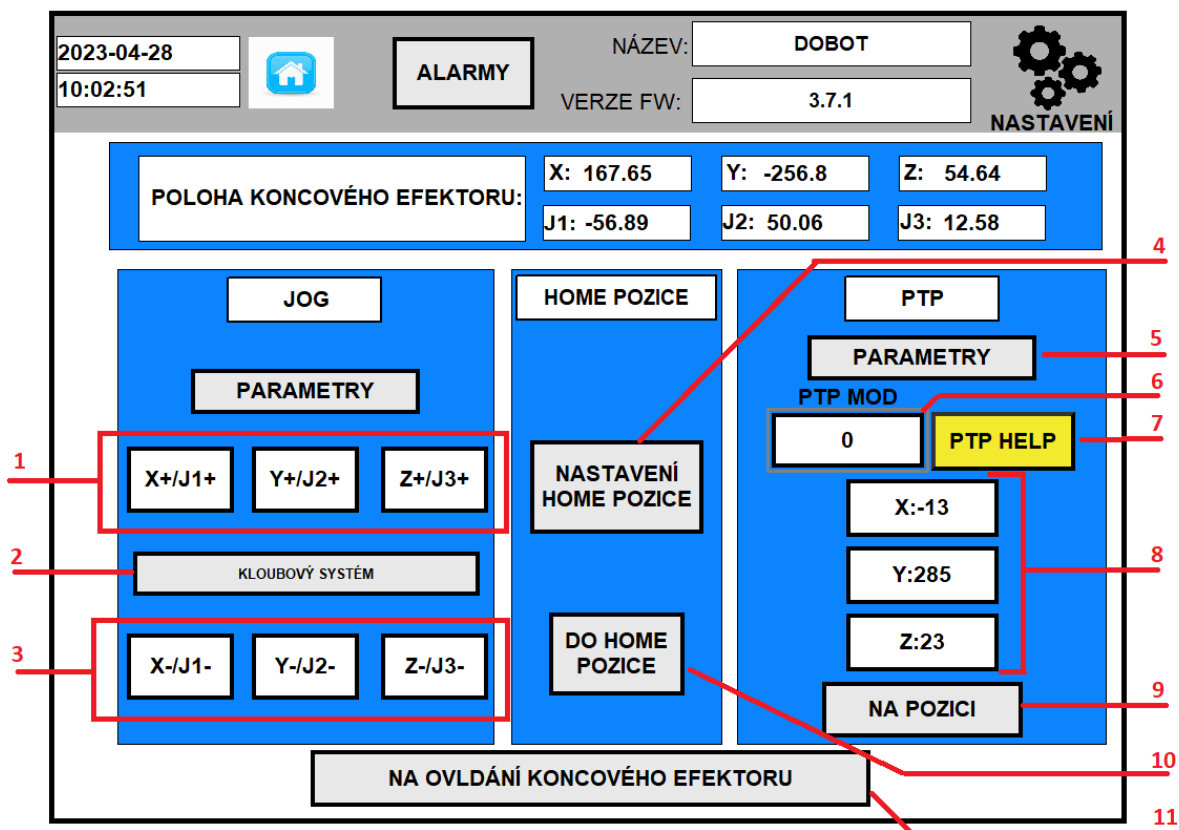
Vzhledem k tomu, že pokud se bude využívat pohyb Dobotu, s největší pravděpodobností se bude ovládat i koncový efektor, proto je možné přecházet z obrazovky pro ovládání pohybů na obrazovku pro ovládání koncového efektoru.

Obrazovka je aktualizována nejčastěji, jak jen to factory talk view umožňuje, aby souřadnice koncového efektoru při pohybu byly co nejaktuálnější. Popis prvků na obrazovce vzhledem k obrázku 5.2:



Obr. 5.1: Main obrazovka vizualizace

1. Tlačítka pro ovládání pohybu v kladném směru os, pokud je nastavený kartézský souřadný systém odesílají se příkazy korespondující s nápisy X+, Y+, Z+, v případě kloubového souřadnicového systému zase J1+, J2+, J3+.
2. Tlačítko pro přepínání mezi souřadnicovými systémy - kartézský / kloubový
3. Tlačítka pro ovládání pohybu v záporném směru os, pokud je nastavený kartézský souřadný systém odesílají se příkazy korespondující s nápisy X-, Y-, Z-, v případě kloubového souřadnicového systému zase J1-, J2-, J3-.
4. Přejít na obrazovku pro nastavení souřadnic home pozice
5. Přejít na obrazovku s parametry PTP pohybu
6. Výběr PTP režimu pohybu
7. Přejít na obrazovku s vysvětlením jednotlivých režimů PTP pohybu
8. Souřadnice pro PTP režim
9. Tlačítko pro odstartování PTP pohybu
10. Tlačítko pro odstartování přesunu do home pozice
11. Přejít na obrazovku s ovládáním koncového efektoru



Obr. 5.2: Obrazovka s ovládáním pohybů

5.3 Nastavení JOG parametrů

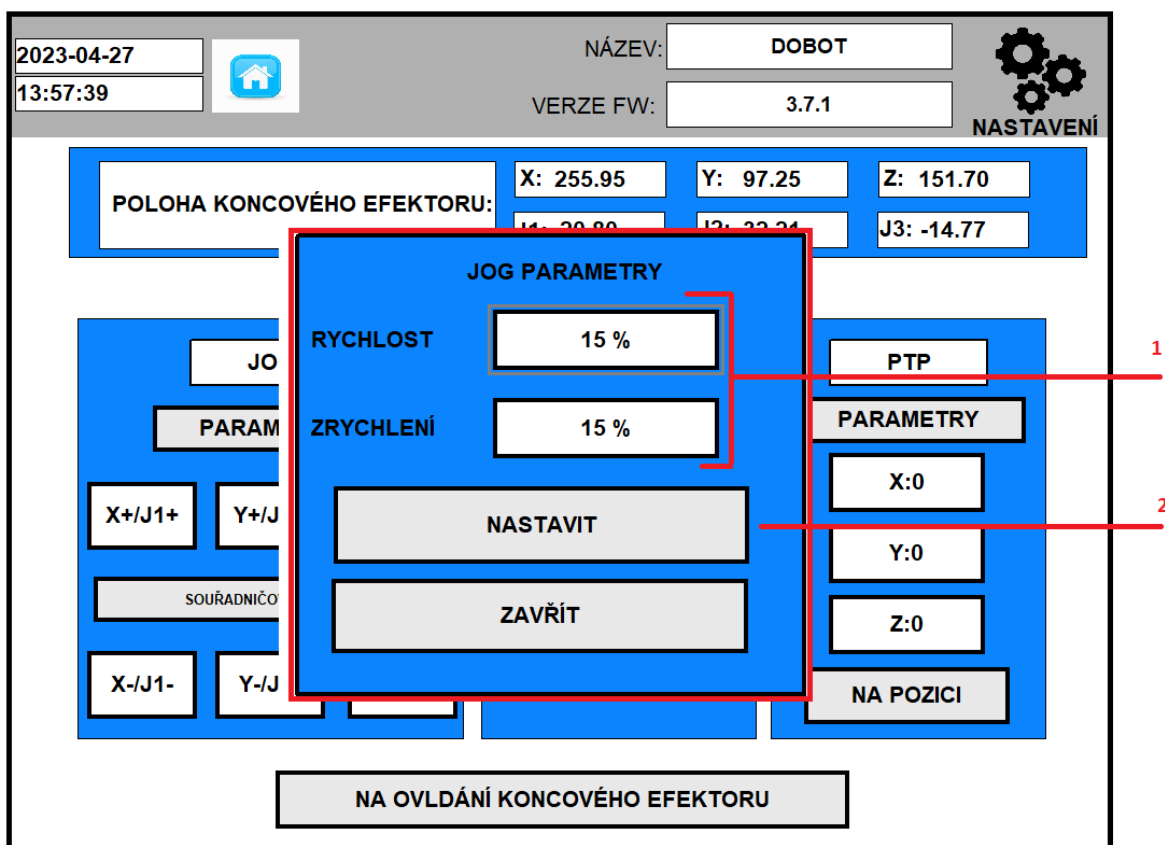
Na tuto obrazovku je možné přejít přímo z obrazovky pro ovládání pohybů, je zde umožněno nastavovat rychlost a zrychlení pohybu v JOG režimu. Oba tyto parametry se nastavují v procentech, je tedy možné zadávat hodnoty 0 až 100. Rychlost a zrychlení je společná pro všechny klouby.

Náhled obrazovky je na obrázku 5.3. Parametry se nastavují v části označené jedničkou, nestačí je pouze zadat, pro jejich nastavení je potřeba ještě stisknout tlačítko „Nastavit“(2).

5.4 Nastavení HOME pozice

Dalším nastavením, do kterého je možné se dostat z okna pro ovládání pohybů je nastavení souřadnic home pozice. Náhled obrazovky je v obrázku 5.4. Popis obrázku:

1. Samotné pop-up okno nastavení home pozice
2. Nastavení souřadnic home pozice s aktuálním nastavením



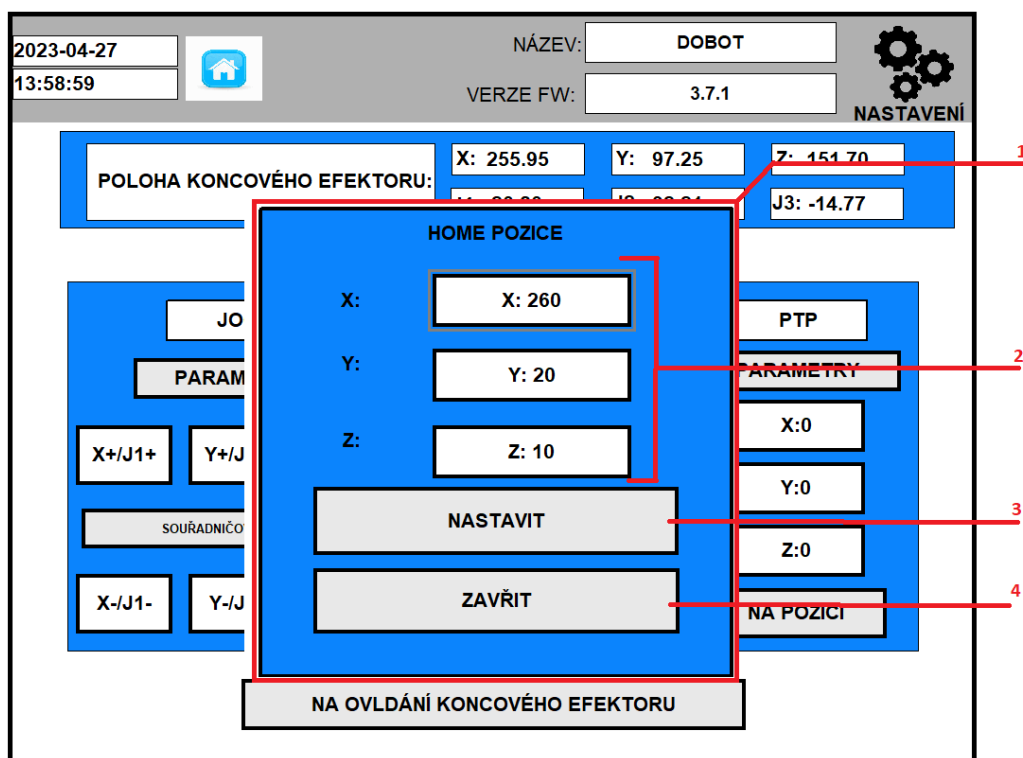
Obr. 5.3: Pop-up okno s nastavením parametrů JOG pohybu

3. Tlačítko pro nastavení nových souřadnic
4. Tlačítko pro zavření pop-up okna a návrat na obrazovku s ovládáním pohybu.

5.5 Nastavení PTP parametrů

Z obrazovky ovládající pohyb je možné přejít do dvou dalších, které se týkají PTP pohybů, nastavení PTP parametrů je jedna z nich. Slouží k nastavení parametrů rychlosti a zrychlení, stejně jako v případě JOG pohybů v procentech.

Dále je možné nastavovat parametry PTP jump pohybu, kdy se rameno prvně zvedne o definovanou výšku - výška zdvihu a poté také maximální zdvih - Z limit. Náhled obrazovky je na obrázku 5.5.



Obr. 5.4: Pop-up okno s nastavením HOME pozice

5.6 PTP help

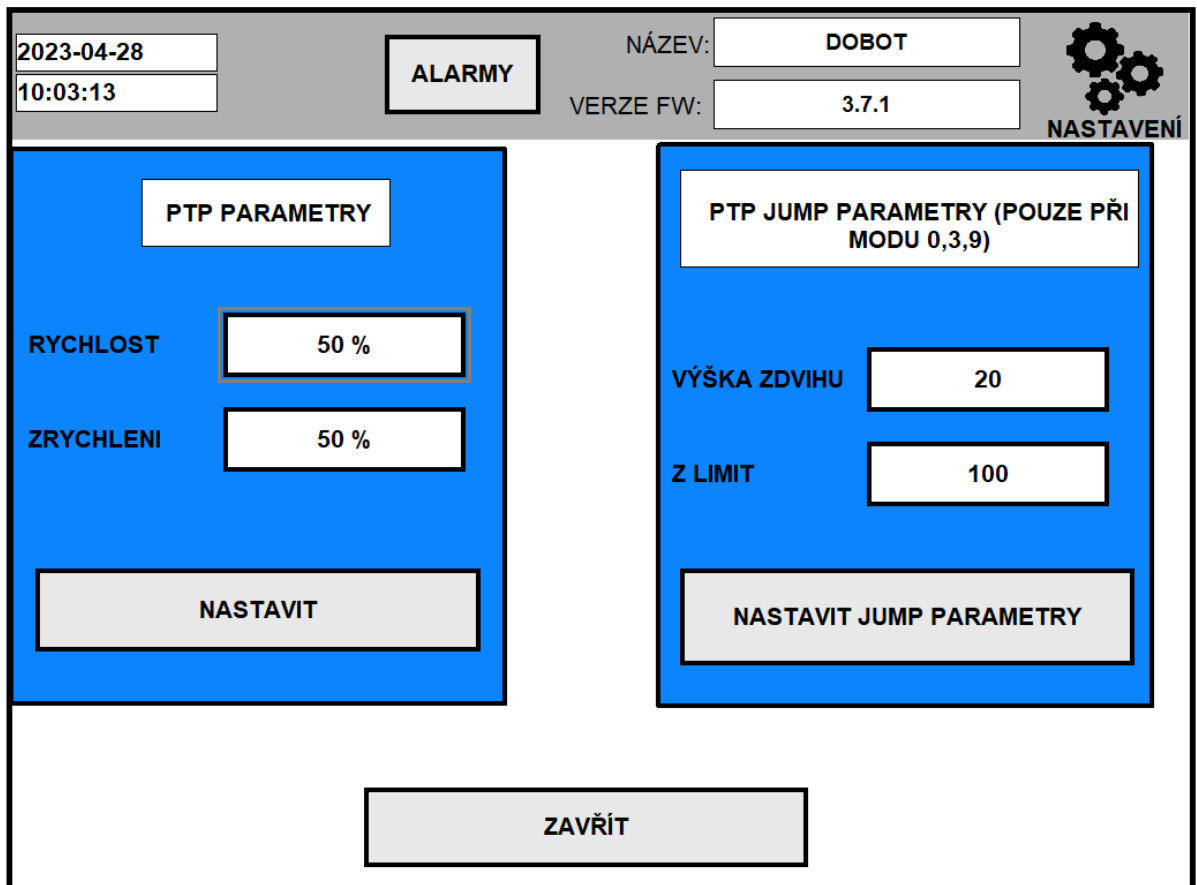
Vzhledem k většímu počtu možností nastavení PTP pohybu se z ovládání pohybu dá přejít také na obrazovku „PTP help“, na této obrazovce jsou graficky vysvětleny režimy pohybů a vysvětleny další možnosti nastavení s jejich číselným označením pro nastavení režimu. Náhled obrazovky je na obrázku 5.6.

5.7 Ovládání koncového efektoru

S obrazovkou ovládání pohybů je úzce spojené také ovládání koncového efektoru, na které se dá přejít právě ze zmiňované obrazovky pro ovládání pohybu. Je možné ovládat dva typy koncového efektoru.

Gripper, který předmět sevře mezi dvě ramena a suction cup, který podtlakem přichytí předmět. Dále je zde možné ovládat kloub J4, pro otáčení koncovým efektozem. Náhled obrazovky je na obrázku 5.7. Popis obrázku:

1. Část pro ovládání gripperu
2. Ovládání kloubu koncového efektoru ve směrech R+ a R-



Obr. 5.5: Obrazovka s nastavením PTP parametrů

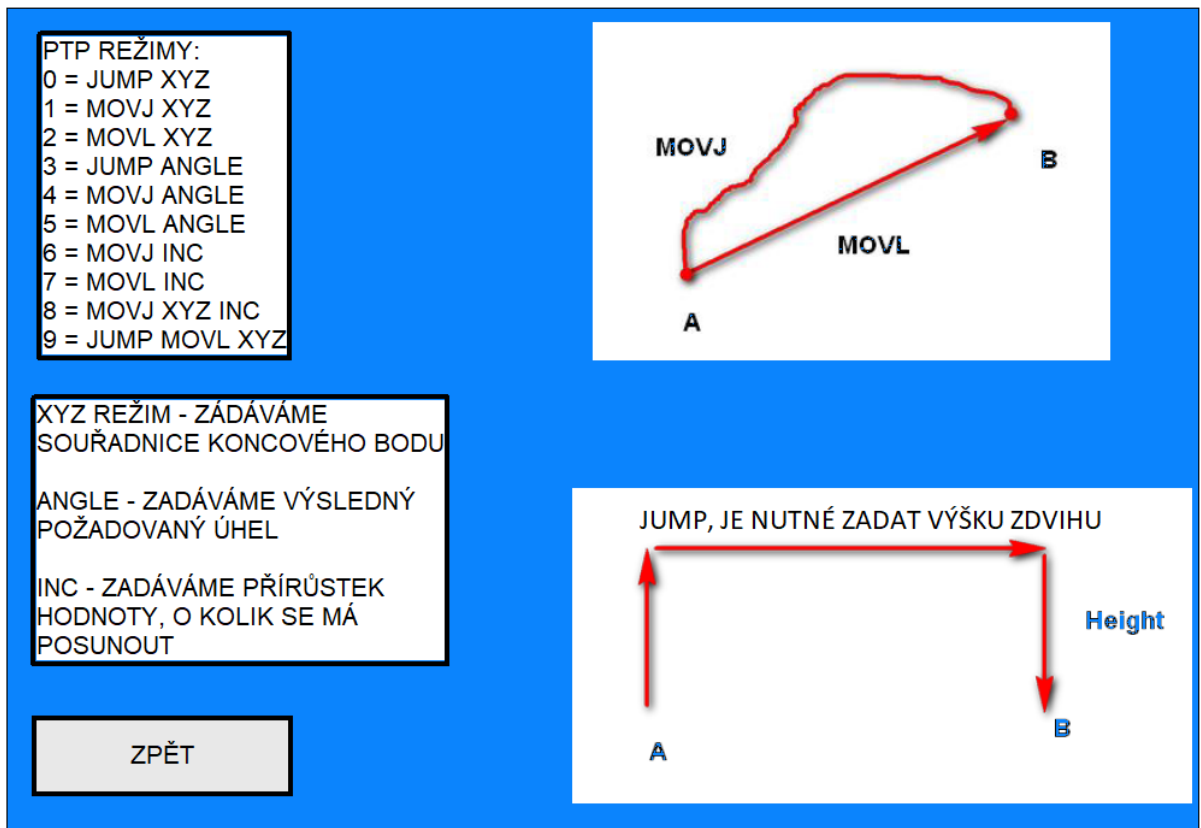
3. Ovládání suction cup
4. Přechod na obrazovku pro ovládání pohybů

5.8 Okno alarmů

V případě, že nastane nějaký z chybových stavů Dobotu, ve vizualizaci se otevře okno, kde jsou vypsány aktuální alarmy. Z tohoto okna je možné smazat výpis alarmů, odeslat příkaz pro resetování chybového stavu a provést acknowledge alarmu. Náhled obrazovky je na obrázku 5.8.

5.9 Komunikace

Jelikož je práce s dobotem založena na funkční komunikaci, je pro ní vytvořeno vlastní okno, kde je možné nastavit komunikační kanál, provést jeho inicializaci a je



Obr. 5.6: Obrazovka s vysvětlným PTP režimů

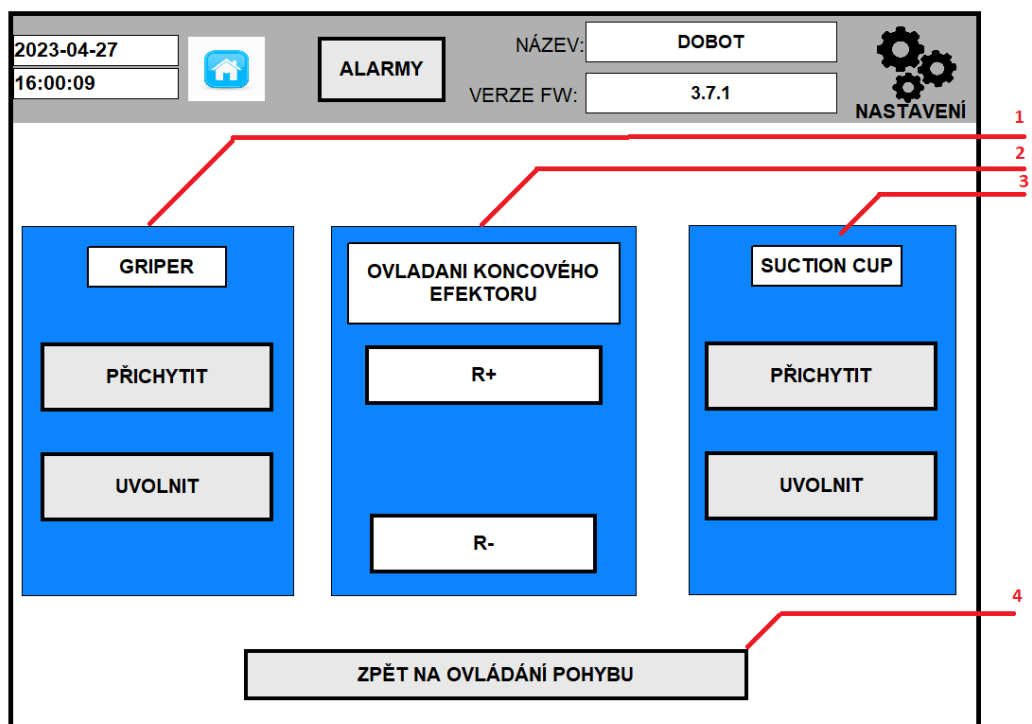
zde také přehled v podobě poslední odeslané a přijaté zprávy. Náhled obrazovky je na obrázku 5.9. Popis obrázku:

1. tlačítko sloužící pro přepínání mezi kanálem 0 a 1, stiskem tlačítka „NASTAVIT“ se provede inicializace vybraného kanálu
2. Poslední přijatá zpráva z Dobota
3. Poslední odeslaná zpráva do Dobota

Pokud uživatel odstartuje inicializaci, je po dobu tohoto procesu zobrazeno informační hlášení o tom, že inicializace probíhá, náhled takového stavu je na obrázku 5.10.

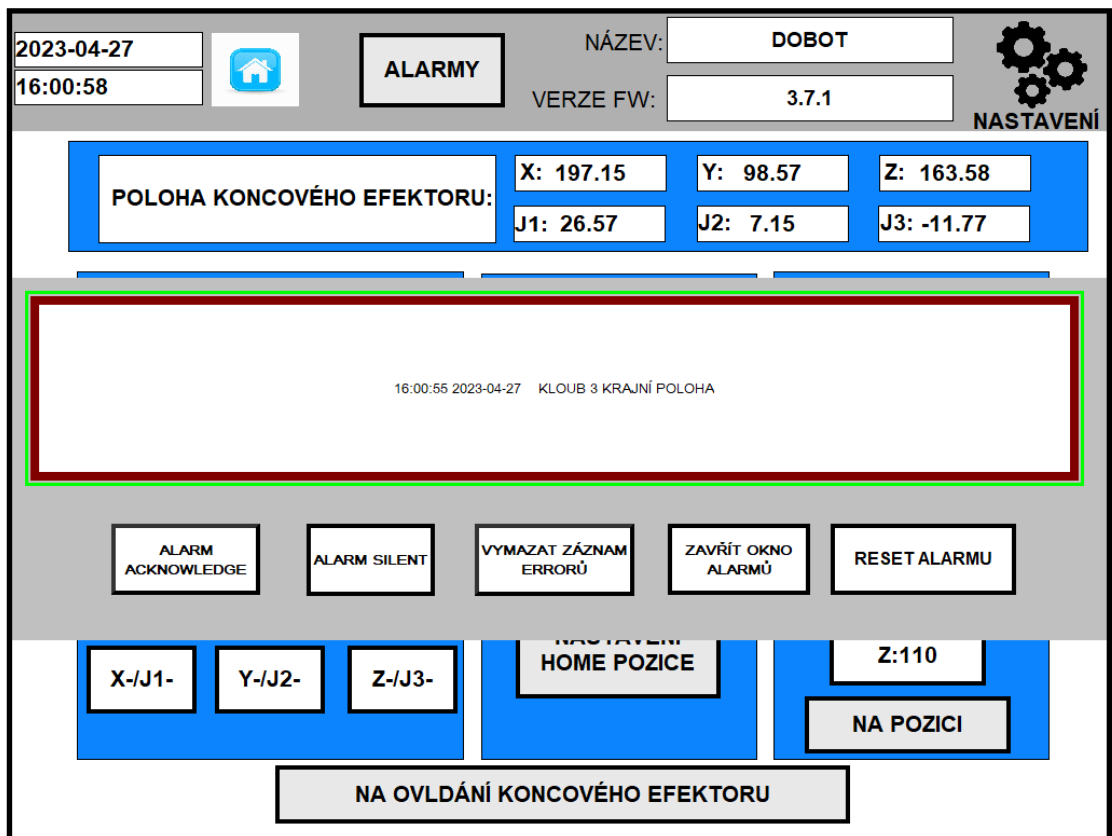
5.10 Nastavení

Kliknutím na ozubené kolečko v horní části obrazovek se přejde do nastavení Dobota. Kde se zobrazí jeho název, verze jeho firmwaru a nastavení home pozice. Jméno a home pozice je možné z této obrazovky editovat kliknutím na ikonu tužky vedle příslušné kolonky. Náhled obrazovky je na obrázku 5.11. Popis obrázku:

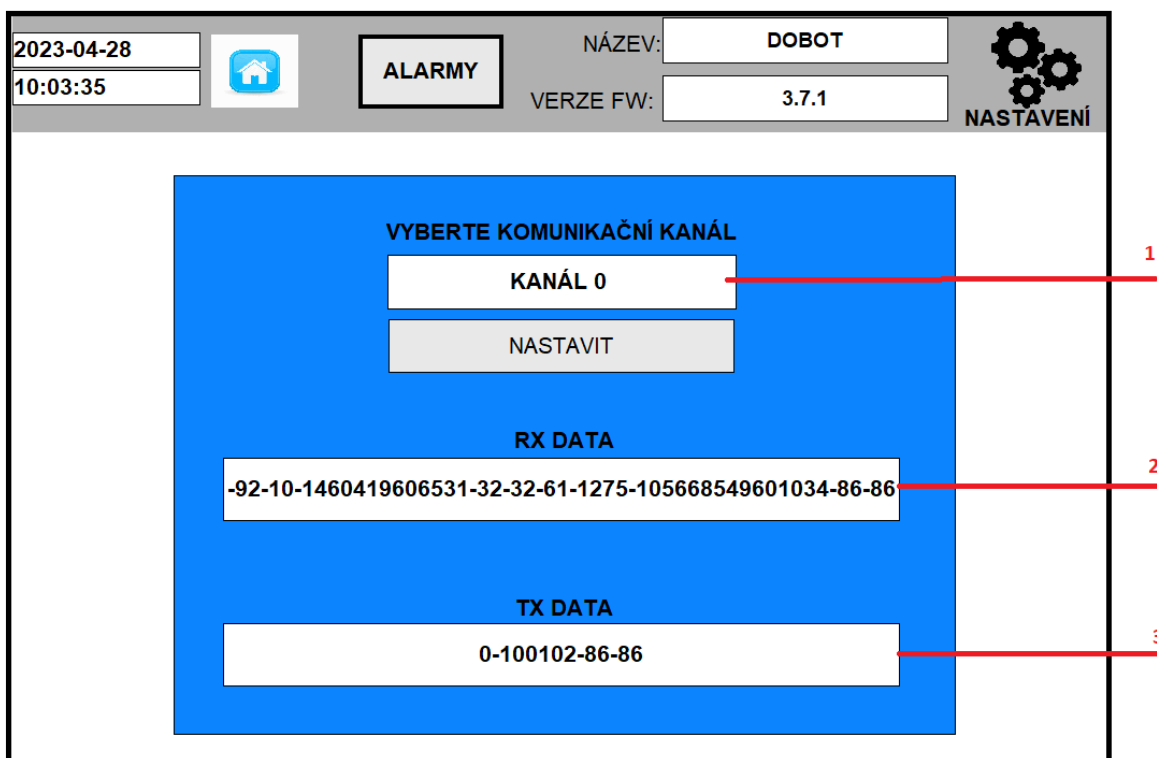


Obr. 5.7: Obrazovka s ovládáním koncového efektoru

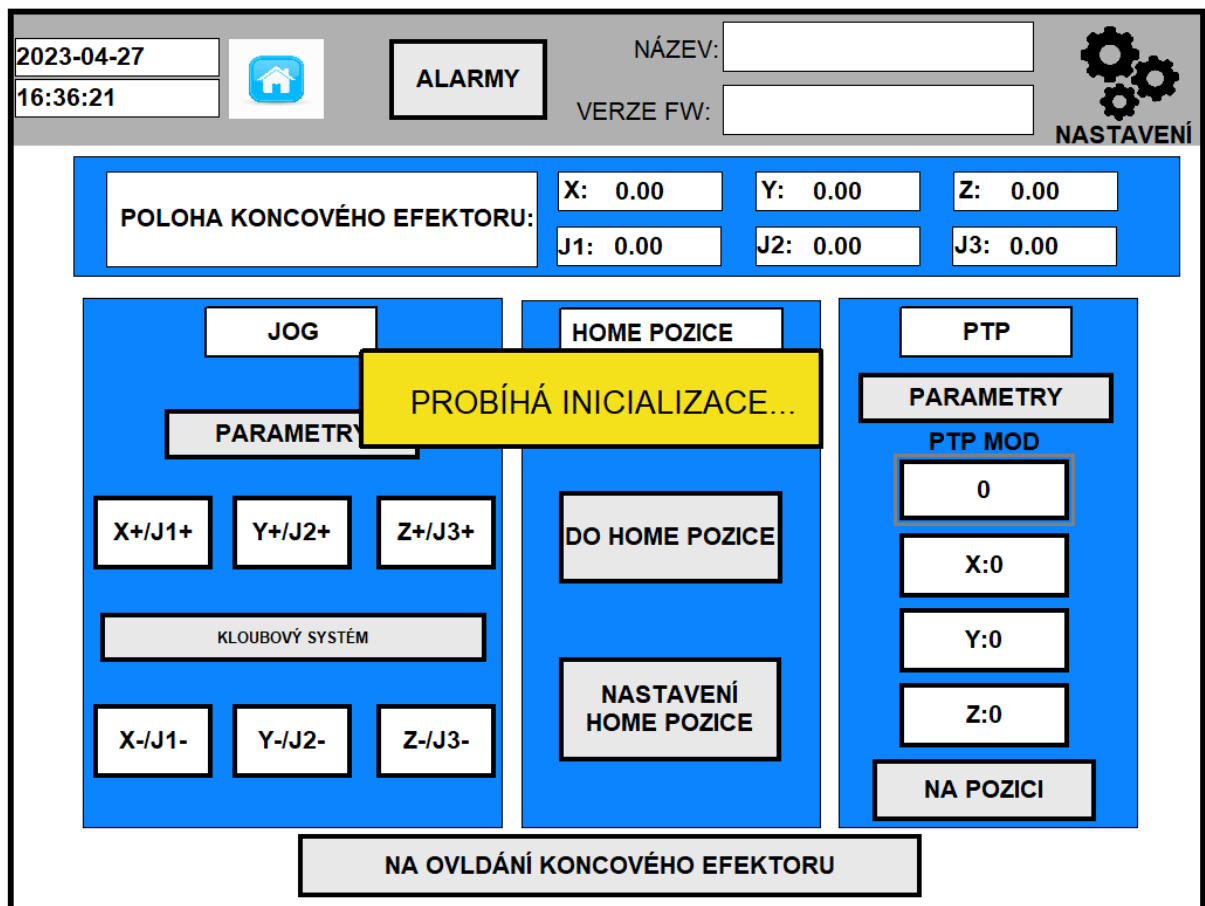
1. Aktuální název Dobot
2. Tlačítko pro otevření editačního menu názvu Dobot
3. Verze firmwaru
4. Souřadnice home pozice




Obr. 5.8: Obrazovka s alarmy





Obr. 5.9: Obrazovka komunikace



Obr. 5.10: Inicializační stav

2023-04-27		NÁZEV: DOBOT
13:59:55		VERZE FW: 3.7.1

NÁZEV DOBOTA		
DOBOT		
FW VERZE		
3.7.1		
HOME POZICE		
X:	Y:	Z:
260	20	10.00001
		

Red lines with numbers 1, 2, 3, and 4 point to the top header, the DOBOT name field, the FW VERZE field, and the HOME POZICE section respectively.

Obr. 5.11: Obrazovka Nastavení

6 Popis demonstrační úlohy

V rámci demonstrační úlohy je využit Dobot pro přesouvání molitanových krychliček, cílem je demonstrovat funkčnost ovládání pohybů a jejich nastavení. Cílem je přesunout rameno Dobot z jednoho konce dopravníku, na kterém jsou krychličky, na druhý, pomocí suction cup přichytit krychličku a umístit jí na jinou krychličku. Přesunout se opět do výchozí pozice a PTP pohybem se přesunout opět na druhou stranu dopravníku, kde se chytí další krychlička a přesune se na začátek dopravníku. Demonstrační úloha bude zakončena nastavením HOME pozice a přesunem do ní.

V rámci této demonstrační úlohy je využita a předvedena funkce inicializace, pohyby v JOG režimu a jejich nastavení. Následně práce s PTP režimem, nastavením home pozice a přesunem do ní. Popsaná demonstrační úloha je k vidění na videu v přílohách.

Závěr

Tato diplomová práce je rozdělena do teoretické a praktické části.

V teoretické části byl objasněn pojem stacionární roboti, jelikož do této skupiny patří právě Dobot Magician, kterému se věnuje tato práce. Byly popsány hlavní druhy stacionárních robotů, čtenář by měl být obeznámen s výhodami, nevýhodami a typickým využitím jednotlivých robotů.

Teoretická část dále popisuje samotného Dobota. Je popsán jak po stránce konstrukční, přes všechny části, ze kterých se skládá, tak i jeho komunikační rozhraní. Na základě tohoto popisu jsem pro práci vybral komunikační rozhraní TTL.

Praktická část se věnuje návrhu samotného zapojení, popisuje nezbytné věci, které je třeba udělat pro funkční komunikaci po hardwarové stránce. Dále popisuje, jakým způsobem je komunikace realizována po softwarové stránce, jsou vysvětleny nezbytné kroky inicializace komunikační karty 1769 - ASCII a objasněn způsob odesílání a přijímání přesně předdefinovaných zpráv pro ovládání Dobota.

V rámci této práce je navržena a otestována komunikace mezi PLC 1769-L36ERMS a Dobotem Magician. Aby byla komunikace možná po hardwarové stránce, je využit RS232 - TTL převodník. V práci čtenář najde obecné přepisy příkazů a i jejich vyplněné vzory, kterými se Dobot ovládá.

Dále jsou zde popsány veškeré použité add-on instrukce a objasněna struktura programu, která je rozdělena do rutin. Následuje popis vizualizace, ze které je možné Dobotu ovládat, na závěr práce je popsána demonstrační úloha.

Je možné tedy konstatovat, že všechny cíle zadání práce jsou úspěšně splněny. Je navrženo funkční zapojení, úspěšně zprovozněná komunikace a ovládání Dobota. Při řešení ovládání jsem se potýkal s neodpovídající dokumentací od výrobce, ale po testování se mi podařilo i tento problém vyřešit. PLC provádí kontrolu všech přijatých dat, nespolehá se jenom na to, že by mělo obdržet správnou odpověď. Vizualizace umožňuje nastavení parametrů, ovládání Dobota i správu chyb, které mohou při práci nastat. Funkcionalita je předvedena v rámci demonstrační úlohy, které bylo natočeno a video je k nalezení v přílohách práce.

Literatura

- [1] SIX TYPES OF INDUSTRIAL ROBOTS *SIX TYPES OF INDUSTRIAL ROBOTS* [cit. 2022-06-10] Dostupné z URL:
<https://diy-robotics.com/blog/top-six-types-industrial-robots-2020/>
- [2] KLOUBOVÝ ROBOT SÉRIE M-900 *KLOUBOVÝ ROBOT SÉRIE M-900* [cit. 2022-06-10] Dostupné z URL:
<http://ppkovanda.cz/fanuc-m-900-kloubovy-robot-serie-m-900>
- [3] SCARA robot *SCARA robot* [cit. 2022-06-10] Dostupné z URL:
<https://www.howtorobot.com/expert-insight/scara-robots/>
- [4] Sick Portálový jeřáb *Sick Portálový jeřáb* [cit. 2022-06-10] Dostupné z URL:
<https://www.sick.com/cz/cs/odvetvi/pristav/kontejnerovy-pristav/kontejnerovy-sklad/portalovy-jerab/c/g3660596.10.2022/>
- [5] Kloubový robot *Kloubový robot* [cit. 2022-06-10] Dostupné z URL:
<http://ppkovanda.cz/fanuc-lr-mate-kloubovy-robot-serie-lr-mate/>
- [6] Kloubový robot fanuc *Kloubový robot fanuc* [cit. 2022-06-10] Dostupné z URL:
<https://www.fanuc.eu/cz/cs/roboty/str%C3%A1nka-filtru-robot%C5%AF/delta-robots/%C5%99ada-m3/m-3ia-6a/>
- [7] Delta robot *Delta robot* [cit. 2022-06-10] Dostupné z URL:
<https://robotsdoneright.com/Articles/what-is-a-delta-robot.html/>
- [8] Cobot *Cobot* [cit. 2022-06-10] Dostupné z URL:
<https://automatizace.hw.cz/robot-vs-cobot.html/>
- [9] Cobot foto *Universal Robots Release their New Generation of Collaborative Robots* [cit. 2023-05-04] Dostupné z URL:
<https://blog.robotiq.com/bid/71494/>
- [10] Dobot Magician *Dobot Magician* Dostupné z URL:
<https://www.generationrobots.com/media/Dobot-Magician-User-Manual-V1.2.4.pdf>
- [11] Fanuc motion instructions *Fanuc motion instructions* Dostupné z URL:
<https://aleksandarhaber.com/fanuc-motion-instructions/>
- [12] Dobot Communication *Dobot Communication* Dostupné z URL:
<http://www.dobot.it/wp-content/uploads/2018/03/Dobot-Communication-Protocol-V1.0.4.pdf>

- [13] Dobot Magician Alarm description *Dobot Magician Alarm description* Dostupné z URL:
<https://download.dobot.cc/product-manual/dobot-magician/pdf/en/Dobot-Magician-ALARM-Description.pdf>

A Přílohy

A.1 Obsah příloženého CD

- | Demonstrační video
 - |_ Dobot_demonstrace.mp4
- | Dokumentace PLC
 - |_ PLC_REPORT.pdf
 - |_ Dobot.ACD
- | Dokumentace vizualizace
 - |_ Dobot_VISU_FINAL_VERSION.apa
- | Textové dokumenty
 - |_ DP-DOBOT-NOVAK.pdf