

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií



DIPLOMOVÁ PRÁCE

Liberec 2010

Bc. Michal Tvrzník

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: N2612 – Elektrotechnika a informatika

Studijní obor: Informační technologie

Realizace mini-robotického centra pomocí produktů firmy RHINO

Realization of mini-robotic center from components of RHINO robots

Diplomová práce

Autor:	Bc. Michal Tvrzník
Vedoucí práce:	doc. Mgr. Ing. Václav Záda CSc.
Konzultant:	Ing. David Lindr

Rozsah práce a příloh

Počet stran textu:	74
Počet příloh:	1
Počet obrázků:	38
Počet tabulek:	12

V Liberci 04.05.2010

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Ústav mechatroniky a technické informatiky

Akademický rok: 2009/2010

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Michal Tvrzník**

Studijní program: **N2612 – Elektrotechnika a informatika**

Studijní obor: **Informační technologie**

Název tématu: **Realizace mini-robotického centra pomocí produktů
firmy RHINO**

Zásady pro vypracování:

1. Seznamte se s roboty firmy RHINO a příslušným softwarovým vybavením.
2. Prozkoumejte možnosti komunikace PC s řídicími systémy Mark III a Mark IV, propojenými s periferními zařízeními.
3. V programovacím jazyku C nebo Java vytvořte aplikaci pro snadné programování a ovládání pohybů robotů pomocí PC.
4. Na základě požadavků vedoucího DP realizujte sadu výukových úloh v laboratoři robotiky, vhodných pro předměty Základy robotiky a Robotika.
5. Vyhodnoťte užitečnost realizovaného systému z hlediska výukových potřeb.

Rozsah grafických prací: dle potřeby dokumentace
Rozsah pracovní zprávy: cca 40–50 stran
Forma zpracování diplomové práce: tištěná/elektronická

Seznam odborné literatury:

- [1] **RHINO: RoboTalk for Windows Manual. USA 2000**
- [2] **RHINO: Owner's Manual – XR-3, XR-4 and SCARA. USA 1995**
- [3] **RHINO: An introduction to robotics – Student Workbook 1. USA 2000**

Vedoucí diplomové práce: doc. Mgr. Ing. Václav Záda CSc.
Ústav mechatroniky a technické informatiky
Konzultant diplomové práce: Ing. David Lindr
Ústav mechatroniky a technické informatiky
Datum zadání diplomové práce: 16. října 2009
Termín odevzdání diplomové práce: 21. května 2010

L.S.

prof. Ing. Václav Kopecký, CSc.
děkan

doc. Ing. Petr Tůma, CSc.
vedoucí ústavu

V Liberci dne 16. října 2009

Prohlášení

Byl jsem seznámen s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

Datum

Podpis

Poděkování

Velmi rád bych chtěl poděkovat vedoucímu mé diplomvé práce, doc. Mgr. Ing. Václavu Zádovi CSc. a konzultantovi, Ing. Davidu Lindrovi, za jejich přístup, poskytnuté znalosti a cenné rady.

Děkuji!

Abstrakt

Cílem diplomové práce bylo vytvořit aplikaci pro řízení robotů rhino a úlohy vhodné k výuce v Laboratoři inteligentních robotů.

Nejprve bylo nutné nastudovat a seznámit se s možnostmi robotických systémů od firmy Rhino Robotics a možnostmi jejich komunikace s počítačem pomocí sériového rozhraní RS-232C. Specificky jde o řídicí systémy Mark III a Mark IV, roboty XR-4 a SCARA a jejich periferní zařízení. Mezi periferní zařízení patří rotační karusel, pásový dopravník, optický snímač a mikrosnímač pro detekci objektu.

Na základě těchto poznatků byla vytvořena aplikace pro snadné ovládání těchto systémů, která je schopná nahradit originální softwarové vybavení od firmy Rhino Robotics. Aplikace byla napsána ve vývojovém prostředí Borland C++ Builder. Součástí aplikace je i vlastní řešení výpočtu pozice a orientace nástroje robota XR-4 pomocí přímé úlohy kinematiky za použití Denavit-Hartenbergovi matice.

Pomocí vytvořené aplikace, ale i ovládacích panelů řídicích systémů, byla vytvořena sada 5 úloh. Úlohy jsou koncipovány tak, aby se na jejich základě bylo možné naučit a uplatnit všechny základní možnosti a funkce robotů Rhino a jejich periférií.

Všechny úlohy jsou podrobně zdokumentovány a zapsány i s postupy při jejich tvorbě v závěrečné práci.

Klíčová slova

Robotika

Rhino Robotics

C++ Builder

RS-232C

Přímá úloha kinematiky

Abstract

The aim of the diploma thesis was to create an application for operation of Rhino robots and make excersises which could be suitable for using in Laboratory of intelligent robots.

First of all it was necessary to work up and get to know with a possibilities of the robotic systems of RHINO Robotics company and with their communication posibility with computer thru serial port RS-232C. Specifcly it is the controller Mark III and Mark IV, robots XR-4 and SCARA and their periferal equipment. Periferal equipements are rotary carusel, belt conveyoer, optical sensor and microswitcher for object detection.

On the base of this knowledge it was created the application for easy operating this systems, which is able to displace original software from the RHINO Robotics. The application was written by using the development tool Borland C++ Builder. One part of the application is the calculation of the pozition and orientation of XR-4 robot's tool. This is achieved by direct excersise of kinematics and Denavit-Hartenberg matrix.

Combining created application and the teachpendants of control systems was created a package of 5. excersises. Excersises are concieved for complete familiarization with all functions of RHINO robots and their periferal equipments.

All details about the excersises can be found in the documentation. The exercises are written down with the process of their creating.

Keywords

Robotics

Rhino Robotics

C++ Builder

RS-232C

Direct exercise of kinematics

Obsah

Prohlášení	3
Poděkování	4
Abstrakt	5
Abstract	6
Seznam zkratk	12
1 Úvod	13
2 RHINO Robotics	15
2.1 Řídící systém Mark III	15
2.2 Řídící systém Mark IV	17
2.3 Robot XR-4	20
2.4 Robot SCARA	21
2.5 Motory a enkodéry	23
2.6 Příslušenství robotů RHINO	23
2.6.1 Pásový dopravník	23
2.6.2 Rotační karusel	24
2.6.3 Optický snímač	24
2.6.4 Mikrospínač	24
2.7 Programovací jazyk RoboTalk	24
2.7.1 Základní instrukce	25
2.7.2 Problémy s aplikací RoboTalk pro Windows	26
2.8 Sériový port RS-232C a připojení řídicích systémů	27
3 Kinematika robotů	29
3.1 Přímá úloha kinematiky	29
3.1.1 Brátova metoda	29
3.1.2 Denavit-Hartenbergova (DH) metoda	30
3.2 Vlastní výpočet kinematiky	31
4 Vývoj vlastní aplikace – RhinoTalk	36
4.1 C++ Builder	36
4.2 Technické parametry aplikace	37

4.2.1	Nastavení sériového portu	37
4.2.2	Testování nečinnosti při běhu programu	39
4.2.3	Běh programu ve vláknu	40
4.2.4	Změna z pohybu jednotlivých os na kartézské souřadnice	40
4.3	Připojení řídicího systému Mark IV k PC	41
4.4	Online ovládání robota	41
4.4.1	Ovládání jednotlivých os	41
4.4.2	Ovládání v kartézských souřadnicích (XYZ)	41
4.5	Ovládání vstupních a výstupních portů	42
4.6	Tvorba kompletních programů	43
4.7	Řízení běhu programu	43
4.8	Zobrazení polohy a orientace nástroje	44
4.9	Kalibrace posuvné základny robota	45
5	Výukové úlohy	47
5.1	Základní manipulace pomocí softwaru RhinoTalk	47
5.2	Základy ovládání pomocí ovládacího panelu	50
5.3	Ovládání vstupů a výstupů	53
5.3.1	Zapojení vstupů a výstupů	53
5.3.2	Mark IV a robot XR-4	54
5.3.3	Mark III a robot SCARA	55
5.3.4	Záloha programu pro Mark III	58
5.4	Detekce objektů na pohybujícím se dopravníku	60
5.4.1	Zapojení dopravníku a optického snímače	60
5.4.2	Nastavení robota SCARA	61
5.4.3	MarkIV a robot XR-4	62
5.4.4	MarkIII a robot SCARA	63
5.5	Mikrospínač jako detektor objektu	65
5.5.1	Připojení rotačního karuselu	65
5.5.2	Zapojení a upevnění mikrospínače	66
5.5.3	Využití přepínačů	67
6	Závěr	69
	Literatura	72
	Seznam příloh	74

Seznam obrázků

1	Řídicí systém Mark III	15
2	Řídicí systém Mark IV	17
3	Robot XR-4	20
4	Robot SCARA	22
5	Uživatelské rozhraní aplikace RoboTalk	25
6	Chyba komunikace s řídicím systémem Mark III	26
7	Symbolické značení rotačních kloubů	29
8	Symbolické značení translačních kloubů	29
9	Kinematika robota XR-4	31
10	Pohyb os při změně natočení os E a D	34
11	Vývojové prostředí C++ Builder	37
12	Modul ovládání jednotlivých os	41
13	Modul ovládání v kartézských souřadnicích	42
14	Modul ovládání vstupů a výstupů	42
15	Modul řízení běhu programu	43
16	Modul polohy nástroje a robota	44
17	Matrice polohy	45
18	Dřevěné kostky	47
19	Přehled směrů pohybů os robota XR-4	48
20	Ovládací panel – Mark IV	50
21	Plastová tyčka a kolečko	52
22	Zapojení výstupu na vstup pro Mark IV	53
23	Zapojení vstupů a výstupů mezi Mark IV a Mark III	54
24	Pozice objektu určeného k přesunu	54
25	Ovládací panel – Mark III	55
26	Přehled směrů pohybů os robota SCARA	56
27	Mark3 Utility – uživatelské rozhraní	58
28	Mark3 Utility – ukládání programu do PC	58
29	Mark3 Utility – nahrávání programu do řídicího systému	59
30	Zapojení optického snímače - oba moduly	60
31	Propojení výstupu snímače s řídicím systémem Mark IV	61
32	Výškové nastavení robota SCARA	62
33	Přemístění detekovaného objektu z dopravníku	64
34	Zapojení motoru karuselu v režimu „rychlosti“	65

35	Upevnění mikrospínače k dopravníku	66
36	Propojení mikrospínače s řídicím systémem Mark IV	66
37	Přepínač na řídicím systému Mark IV	67
38	Detekce objektů na rotačním karuselu	68

Seznam tabulek

1	Technická specifikace řídicího systému Mark III	16
2	Technická specifikace řídicího systému Mark IV	18
3	Technická specifikace robota RHINO XR-4	21
4	Technická specifikace robota RHINO SCARA	22
5	Zapojení RS-232C, PC (9 pinů) → Mark III (25 pinů)	27
6	Zapojení RS-232C, PC (9 pinů) → Mark IV (25 pinů)	28
7	Proměnné DH matice	32
8	Konstantní hodnoty pro DH matici	32
9	Převod z jednotek enkodéru na stupně a mm	34
10	Kroky programu 2. úlohy pro Mark IV	51
11	Kroky programu 3. úlohy pro Mark III	57
12	Kroky programu 4. úlohy pro Mark III	63

Seznam zkratek

CNC	Computer Numerical Control
DC	Stejnoseměrný proud
DCB	Device Control Block
DH	Denavit-Hartenberg
EEPROM	Electrically Erasable Programmable Read-Only Memory
MTI	Mechatronika a Technická Informatika
PC	Personal Computer
PID	Proporčně, Integračně, Derivační
PM	Pulse Modulation
PWM	Pulse Width Modulation
SCARA	Selective Compliance Assembly Robot Arm
TCP	Tool Center Point
TP	Teach Pendant
TTL	Transistor-Transistor Logic, Tranzistorově-Tranzistorová Logika
WYSIWYG	What You See Is What You Get

1 Úvod

Téma diplomové práce bylo vybráno na základě potřeby dále rozšiřovat rozsah úloh prováděných při výuce v rámci Laboratoře inteligentních robotů. Při hledání nových možností a rozmyšlení nad moderními a drahými technologiemi byla nakonec dána přednost relativně starému vybavení od americké firmy RHINO Robotics. Jedná se o set dvou řídicích systémů a robotů, rozšířený o další motorově poháněné příslušenství a součásti určené k detekci objektů.

Dále bylo nutné navrhnout koncepci celé úlohy diplomové práce. Vzhledem k tomu, že originální software pro řízení řídicích systémů pomocí počítače je závislý na hardwarovém klíči, kterého je pouze jediný exemplář, je jednou z hlavních částí práce tvorba vlastní aplikace pro řízení robotického systému od firmy RHINO Robotics. Vznikne tak možnost ovládat oba řídicí systémy současně a realizovat spolupráci obou robotů.

Před samotnou realizací řídicí aplikace musí dojít k podrobnému prozkoumání možností a vlastností celého robotického systému tak, aby jich aplikace využila v co největším rozsahu.

Aplikace by měla být velice jednoduchá z hlediska ovládání, aby její základy pochopil i téměř úplný začátečník při výuce robotiky. Důležitou částí aplikace bude ovládání pohybu robota v reálném čase, tedy možnosti zadávat povely k přímému pohybu. Další neopomenutelnou součástí aplikace musí být také textový editor pro tvorbu a editaci výsledného programu pro řízení robotů a jejich periferních zařízení. Vhodné bude se také zamyslet nad využitím úloh kinematiky pro zjišťování aktuální pozice a orientace nástroje robota. Celá aplikace bude vytvářena pomocí vývojového prostředí C++ Builder od firmy Borland.

Další důležitou částí diplomové práce je tvorba výukových úloh v nově vytvořené aplikaci. Úlohy by také měly být koncipovány se zaměřením na snadnou pochopitelnost a měly by seznámit s většinou základních možností a funkcí robotických systému RHINO i nově vytvářené aplikace.

Součástí zadání je také vyhodnotit užitečnost realizovaného systému z hlediska výukových potřeb. Proto bude v práci uvedeno krátké zamyšlení nad tím, jakým způsobem přispělo řešení celé diplomové práce k rozšíření a zlepšení výuky v rámci laboratoře a předmětů věnujících se robotice.

Celá práce je rozdělena do několika logických celků. V první části jsou teoreticky rozebrány všechny součásti robotického systému RHINO a to po stránce hardwaru

i softwaru. Jedná se především o oba roboty XR-4 a SCARA a jejich řídicí systémy Mark III a Mark IV.

Druhá kapitola se věnuje kinematice robotů. Nejdříve je probrána teoreticky i s metodami její řešitelnosti. V druhé části této kapitoly je praktické řešení Denavit-Hartenbergovi metody přímé úlohy kinematiky. Výsledky výpočtů jsou využity pro zjištění aktuální polohy a orientace nástroje robota v prostoru a jsou přístupné ve vytvořené aplikaci pojmenované RhinoTalk.

Tvorbou této aplikace, pomocí vývojového prostředí C++ Builder, se zabývá třetí část práce. Je zde rozebráno technické řešení některých důležitých částí aplikace a také představena většina možností výsledného produktu.

Ve čtvrté části je série výukových úloh. Všechny úlohy jsou přehledně sepsány i s postupem jejich realizace. V popisu není opomenuto ani nastavení všech hardwarových komponent.

V závěru celé práce jsou zhodnoceny dosažené výsledky a také podrobně rozebrán přínos celé práce a vytvořených úloh pro potřeby výuky.

2 RHINO Robotics

RHINO Robotics s.r.o. je americká firma nabízející kompletní řadu robotů, CNC (Computer Numerical Control) soustruhy, CNC frézy a příslušenství pro výukové a vývojové aplikace.

Na fakultě mechatroniky a mezioborových inženýrských studií (FM) jsou k dispozici dva řídicí systémy Mark III a Mark IV a roboty XR-4 a SCARA. Oba typy řídicích systémů jsou popsány v kapitolách 2.1 a 2.2 a popis robotů je umístěn v kapitole 2.3 a 2.4. Dále je možné využít i některé doplňky jako je například posuvná základna a rotační karusel. Těmto doplňkům se věnuje kapitola č. 2.6.

2.1 Řídicí systém Mark III

Řídicí systém Mark III je základní systém s možností ovládat 8 os, reagovat na 8 TTL vstupů a ovládat 8 TTL výstupů. Mezi jeho další schopnosti patří také ovládání dvou AUX portů, které poskytují až 20 V při 1,5 A. Tyto porty mohou být použity například k ovládání motorů nebo relé. Systém je schopný pracovat s roboty RHINO XR-3 a SCARA.

Pro zjednodušení ovládání TTL vstupů je běžně na všech hodnota „High“ a změnu docílíme uzemněním daného vstupu. Není tedy nutné žádné další napájení.



Obrázek 1: Řídicí systém Mark III

Mark III disponuje schopností komunikace, mikroprocesorovou logikou, podporou ovládacího panelu, vstupními a výstupními porty a softwarovým jazykem.

Ovládací panel užívá 32 kláves a displej se sedmisegmentovým zobrazováním. Umožňuje kompletní řízení celého řídicího systému a robota.

V tabulce 1 jsou uvedeny všechny důležité parametry řídicího systému Mark III.

Konfigurace	8 portů pro ovládání opticky řízených motorů robotů XR-3 a SCARA a dalších doplňků. Dva AUX porty pro doplňkové motory s PWM podporou.
Vstupy a výstupy	8 vstupů + 8 vnitřních vstupů 8 výstupů, TTL Port pro teachpendant
Mikroprocesor	16bitový hlavní procesor 6502
Komunikace	Rozhraní RS-232C
Kompatibilita	Lze ovládat z jakéhokoliv PC s rozhraním RS-232C
Příkazy	14 ovládacích příkazů
Napájení	120 nebo 240 V, 50 nebo 60 Hz, jednofázové
Rozměry	45,72 cm x 33,02 cm x 15,24 cm
Hmotnost	10,9 kg

Tabulka 1: Technická specifikace řídicího systému Mark III

Řídící příkazy pro Mark III

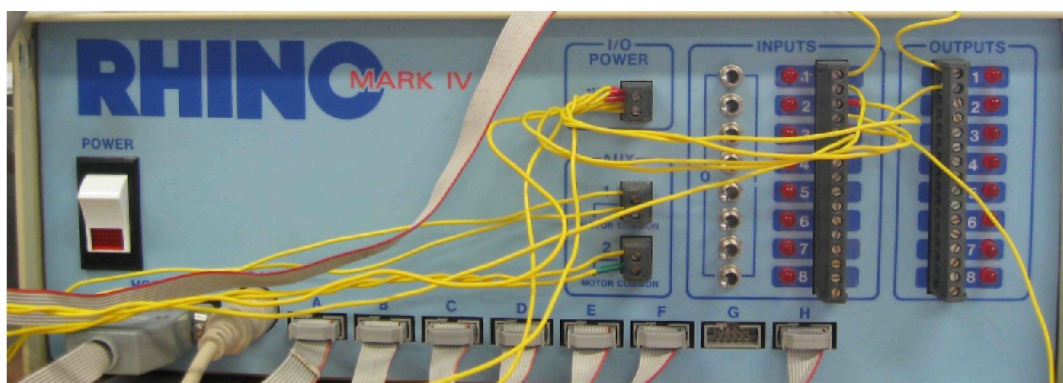
Seznam 14 příkazů softwarového jazyku řídicího systému Mark III, které umožňují kompletní kontrolu řídicího systému při komunikaci přes sériový port. Tyto základní ovládací povely se používají při tvorbě vyšších programovacích jazyků pro řízení robotů RHINO. Např. jazyky RoboTalk a Rhino-Val.

<return>	Vykonání pohybu.
?	Vrátí zbývající vzdálenost.
A–H	Nastaví aktuální hodnotu polohy osy.
I	Načte stav koncových spínačů C–H.
J	Načte stav koncových spínačů A–B a vstupních portů.
K	Načte stav vstupů.
L	Zapne port AUX 1.
M	Vypne port AUX 1.
N	Zapne port AUX 2.
O	Vypne port AUX 2.
P	Nastavení výstupních portů do hodnoty logické 1.
Q	Reset řídicího systému.

- R** Nastavení výstupních portů do hodnoty logické 0.
X Zastaví pohyb motoru.

2.2 Řídící systém Mark IV

Řídící systém RHINO Mark IV a jeho ovládací panel jsou plně kompatibilní s XR sérií robotů (XR-3, XR-4) i SCARA a mohou být použity jako rozšiřující náhrada pro řídicí systém Mark III.



Obrázek 2: Řídící systém Mark IV

Řídící systém podporuje řízení rychlosti všech motorů a také posílání PWM (Pulse Width Modulation) signálů.

Pro běh řídicího systému i ovládacího panelu jsou použity dva 8bitové a jeden 16bitový mikroprocesor. Ovládací panel má svůj vlastní 8bitový mikroprocesor a plně programovatelný alfanumerický displej se dvěma řádky po 16 znacích. Druhý 8bitový procesor zajišťuje fungování osmi polohových enkodérů motorů a PWM signálů. 16bitový mikroprocesor obsluhuje celkovou koordinaci řízení systému.

V softwarovém jazyku je podporováno více jako 100 příkazů pro ovládání systému.

Programy vytvořené pomocí ovládacího panelu je možné ukládat v paměti EEPROM (Electrically Erasable Programmable Read-Only Memory). Pro tento typ paměti není nutné používat žádnou bateriovou zálohu.

Pro komunikaci po sériovém portu s hostitelským počítačem i ovládacím panelem je implementován Full Handshake.

Všechny motorové obvody mají zabudovanou automatickou ochranu proti přetížení. To zabraňuje zničení motorových obvodů a tranzistorů.

Mark IV na rozdíl od Mark III podporuje pohyb nejen kloubových, ale i v kartézských souřadnicích jako základní součást ovládacích instrukcí řídicího systému.

V tabulce 2 jsou uvedeny všechny důležité parametry řídicího systému Mark IV.

Konfigurace	8 portů pro ovládání opticky řízených motorů robotů série XR a SCARA a dalších doplňků. Každý port má plnou PID podporu. Dva AUX porty pro doplňkové motory s PWM podporou.
Vstupy a výstupy	8 vstupů 8 vstupních spínačů 8 výstupů Port pro teachpendant
Mikroprocesor	16bitový hlavní procesor 8bitový procesor kontroleru motorů 8bitový procesor kontroleru teachpendantu
Komunikace	Rozhraní RS-232C
Kompatibilita	Lze ovládat z jakéhokoliv PC s rozhraním RS-232C
Příkazy	Přes 100 ovládacích příkazů
Napájení	120 nebo 240 V, 50 nebo 60 Hz, jednofázové
Rozměry	38,1 cm x 45,72 cm x 15,24 cm
Hmotnost	15,9 kg

Tabulka 2: Technická specifikace řídicího systému Mark IV

Řídicí příkazy pro Mark IV

Seznam důležitých příkazů systému Mark IV s jejich stručným popisem. Tyto příkazy jsou využity pro ovládání systému při komunikaci z počítače přes sériový port. Jsou důležité i pro programování pomocí vytvářené aplikace RhinoTalk, o které se více dozvíme v kapitole 4.

Ovládání systému

TH Ovládání systému pomocí PC.

TX Ovládání systému pomocí ovládacího panelu.

Ovládání vstupů a výstupů

OB,b,s	Nastaví specifikovaný výstupní port na požadovanou hodnotu. <i>b...port (1-8), s...hodnota (0 - Off, 1 - On)</i>
OP,d	Nastaví specifikované výstupní porty. Zadání v hexadecimálním tvaru. <i>d...porty (0-255)</i>
WA	Zruší všechna čekání na vstup.
WI,b,s	Detekuje, kdy port nebo přepínač dosáhne požadované hodnoty. <i>b...port (1-8), přepínač (9-16), s...hodnota (0 - Off, 1 - On)</i>
XS,p,d	Nastaví výstupní hodnotu portu AUX. <i>p...port (1 nebo 2), d...velikost napětí (-100 až +100)</i>

Ovládání motorů a pohybu

AC,m	Vymaže aktuální pozici motoru. <i>m...motor (A,B,C,D,E,F,G nebo H)</i>
GC	Zavře čelisti.
GO	Otevře čelisti.
AS,d	Nastaví akceleraci celého systému. <i>d...procenta (0-100)</i>
HA	Nastaví všechny motory do počáteční pozice (Hard Home).
HG	Nastaví všechny motory do uživatelské počáteční pozice (Soft Home).
HH	Vyhledá počáteční pozici robota pomocí pohybu přes pozice koncových spínačů. Pro všechny motory.
HL,m	Vyhledá počáteční pozici pomocí pohybu přes pozice koncových spínačů. Pouze pro daný motor. <i>m...motor (A,B,C,D,E,F,G nebo H)</i>
HS	Nastaví aktuální pozici jako uživatelskou počáteční pozici (Soft Home).
MA	Zastaví všechny motory a vypne AUX porty.
MC	Spustí koordinovaný pohyb. Všechny motory začnou i ukončí pohyb ve stejném okamžiku.
MI	Spustí okamžitý pohyb. Všechny motory začnou vykonávat pohyb ve stejný okamžik, ale je ukončen v závislosti k rychlosti a dráze.
MM,m	Zastaví daný motor. <i>m...motor (A,B,C,D,E,F,G nebo H)</i>
MS,m	Spustí daný motor. <i>m...motor (A,B,C,D,E,F,G nebo H)</i>
MX	Spustí pohyb v souřadnicích XYZ.
PD,m,d	Nastaví absolutní cílovou pozici daného motoru. <i>m...motor (A,B,C,D,E,F,G nebo H) d...velikost (-32767 až +32767)</i>
PR,m,d	Nastaví relativní cílovou pozici daného motoru. <i>m...motor (A,B,C,D,E,F,G nebo H) d...velikost (-32767 až +32767)</i>

- PX,m,f** Nastaví absolutní cílovou pozici daného motoru v souřadnicích XYZ.
m...osa (X,Y,Z, A nebo T) f...velikost (-1000,00 až +1000,00) v mm,
pro osy A a T je velikost ve stupních
- PY,m,f** Nastaví relativní cílovou pozici daného motoru v souřadnicích XYZ.
m...osa (X,Y,Z, A nebo T) f...velikost (-1000,00 až +1000,00) v mm,
pro osy A a T je velikost ve stupních
- VG,d** Nastaví rychlost systému. Platí pro všechny motory.
d...procenta (0-100)
- VS,m,d** Nastaví rychlost dílčího motoru v procentech rychlosti systému (VG,d).
d...procenta (-100 až +100)

Kompletní seznam všech příkazů je možné najít v oficiální dokumentaci k řídicímu systému Mark IV [3].

2.3 Robot XR-4

XR-4 je robotické rameno s 5 rotačními osami a motorem ovládanými chapadly. Místo chapadla je možné využít i další nástroje (magnetické, vakuové), které bohužel nejsou součástí vybavení naší laboratoře robotiky.



Obrázek 3: Robot XR-4

Tělo ramene je sestrojeno z 0,32 cm a 0,64 cm tlustého hliníkové plechu. Robot je kompatibilní pouze s řídicím systémem Mark IV. Všechny osy jsou řízeny stejnosměrnými (DC) servomotory za použití inkrementálních enkodérů pro zpětnou vazbu a disponují také koncovými spínači pro nastavení robota do počáteční pozice (Hard Home).

Všechny důležité parametry robota XR-4 jsou uvedeny v tabulce 3.

Konfigurace	5 os a chapadlo všechny osy zcela nezávislé a ovladatelné současně	
Pohony	6 PMDC servomotorů s integrovanou převodovkou a inkrementálním optickým snímačem.	
Řídicí systém	Mark IV	
Nosnost	2 kg	
Rychlosti	Chapadlo – Otevření 1 s, Zavření 2 s Osy – max. 40 stupňů/s, programovatelné	
Hmotnost	11,4 kg tělo robota, 4,5 kg základna	
Dosah	60,96 cm	
Rozsah pohybu os	F – 350 stupňů D – 180 stupňů B – 7 otáček	E – 150 stupňů C – 250 stupňů

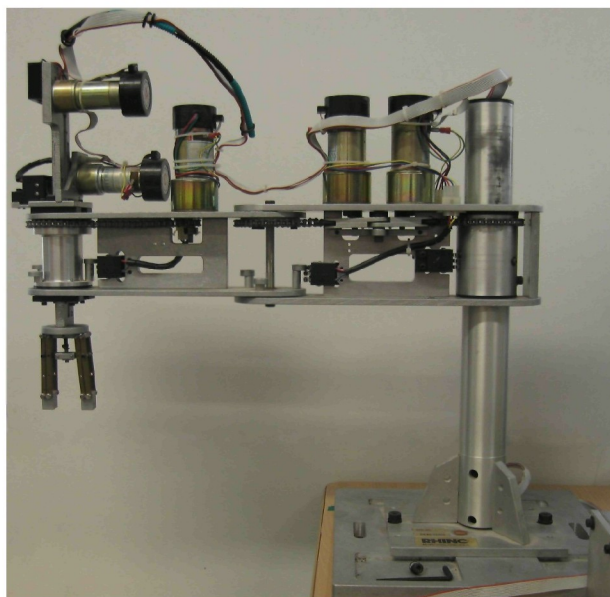
Tabulka 3: Technická specifikace robota RHINO XR-4

Pro realizaci úloh diplomové práce je robot XR-4 rozšířen o translační osu pomocí posuvné základny. Tato posuvná základna, kterou je možné vidět s robotem XR-4 na obrázku č. 3, je originálním příslušenstvím robotů RHINO a pro pohyb využívá standardní motory a konektory jako ostatní vybavení. Podrobnější informace o příslušenstvích a motorech jsou uvedeny v kapitolách 2.6 a 2.5.

2.4 Robot SCARA

SCARA (Selective Compliance Assembly Robot Arm) je robotické rameno se 4 osami a motorem ovládanými chapadly. 3 osy jsou rotačního a jedna translačního typu. Robot je kompatibilní s řídicími systémy Mark III i Mark IV. Maximální dosah ramene robota je 45,72 cm a maximální váha zátěže 0,45 kg. Osa z je manuálně

nastavitelná při velikosti kroku 1 cm. Programovatelný rozsah osy z je 8 cm.



Obrázek 4: Robot SCARA

Konfigurace	4 osy a chapadlo všechny osy zcela nezávislé a ovladatelné současně
Pohony	5 PMDC servomotorů s integrovanou převodovkou a inkrementálním optickým snímačem.
Řídící systém	Mark III a Mark IV
Nosnost	1 kg
Rychlosti	Chapadlo – Otevření 1 s, Zavření 2 s Osy – rychlosti jsou různé, programovatelné
Hmotnost	7,7 kg tělo robota, 4,5 kg základna
Dosah	45,72 cm
Rozsah pohybu	Vnější poloměr – 45,72 cm Vnitřní poloměr – 22,86 cm E – 220 stupňů D – 220 stupňů C – 9,53 cm B – 275 stupňů

Tabulka 4: Technická specifikace robota RHINO SCARA

Všechny osy jsou řízeny DC servomotory za použití inkrementálních enkodérů

pro zpětnou vazbu a disponují také koncovými spínači pro nastavení robota do počáteční pozice (Hard Home).

2.5 Motory a enkodéry

Motory, použité pro roboty a příslušenství, se mohou lišit v převodu, počtu kroků enkodéru a velikosti napájení, ale zapojení, vedení a funkčnost jsou pro všechny stejné. Každý blok motoru se skládá z vlastního motoru, převodovky, optického enkodéru a kabelu. Kabel zajišťuje napájení, informace enkodéru a signály pro koncové spínače.

Použité motory jsou stejnosměrné pro +12 V. Maximální proud při zastavení je obecně pod 2 A. Standardní převodový poměr převodovky je 66.1 na 1.

Všechny motory využívají desetižilový kabel pro připojení k řídicímu systému. Černý konektor, na jedné ze stran kabelu, je určen k připojení k motoru. Konektor není polarizovaný, proto je nutné dbát na správné připojení. Po připojení by měl kabel z konektoru směřovat podél těla motoru. Šedý konektor je naopak určen k připojení k řídicímu systému a je polarizovaný.

Enkodéry jsou inkrementální a skládají se z hliníkového disku se světlými a tmavými segmenty. Z jedné strany je disk osvětlen a světlo je jím odráženo k optickým senzorům. Optické senzory jsou navzájem posunuty od 90 stupňů a tím je zajištěna možnost snímat pozici motoru. Při otáčení vznikají dva posunuté signály, pokud signál „A” předchází „B” otáčí se motor jedním směrem a naopak. Rychlost se snadno vypočítá poměrem času a cyklu signálu.

2.6 Příslušenství robotů RHINO

V této sekci je uveden popis motorově poháněného příslušenství robotů RHINO a také příslušenství sloužícího pro detekci a lokalizaci objektů. Všechno uváděné vybavení je možné použít s oběma druhy řídicích systémů (Mark III, Mark IV).

2.6.1 Pásový dopravník

Jedná se o dopravník s kovovou konstrukcí opatřený gumovým pásem. K pohonu pásu slouží stejnosměrný 12V motor, který je možné připojit ve dvou režimech. První je režim s nastavitelnou pozicí, připojený ke standardnímu portu motoru, kdy je pozice zadávána v jednotlivých krocích enkodéru motoru a je tedy možné dosáhnout přesného posunu dopravníku o stanovenou vzdálenost. U druhého režimu

je dopravník připojen k výstupu AUX, u kterého je možné měnit velikost výstupního napětí a tím i rychlost pohybu dopravníku (Pro zjednodušení bude nadále použito označení jako režim „pozice“ a „rychlosti“). Nevýhodou režimu „pozice“ je nutnost stálé kontroly dopravníku programem, při režimu „rychlosti“ je pouze nastavena rychlost a dále není nutné běh dopravníku hlídat.

2.6.2 Rotační karusel

Karusel může být opět připojen pro režim „pozice“ s přesně danou pozicí otočení nebo se bude otáčet určenou rychlostí v režimu „rychlosti“.

Karusel je vybavený koncovým spínačem, který poskytuje možnost detekce počáteční pozice. Je tedy umožněno nastavit karusel do známé pozice před započítáním dalších operací.

2.6.3 Optický snímač

Je primárně určen k použití na pásovém dopravníku, ale je možné i využití u jiných aplikací. Každá strana snímače kombinuje vysílač a přijímač. Vysílačem je infračervená dioda (Infrared Emitting Diode) a přijímačem fototranzistor. Senzorové moduly je možné pomocí pinů (JP1) nastavit pro napájení 5 V nebo 12 V. Při spojení pinů 1 a 2 je modul nastaven na 5 V a je jej možné připojit k řídicímu systému Mark III. Nastavení pro řídicí systém Mark IV, tedy 12 V, se provede propojením pinů 2 a 3. Pozor na nesprávné nastavení, při připojení k systému Mark IV s nastavením pro 5 V, může dojít k poškození odporů a diod. Přítomnost objektu lze zjistit ze stavu výstupu snímače. Přerušování paprsku a tedy přítomnost objektu je vyjádřena hodnotou „Low“ na výstupu a naopak při nepřítomnosti objektu hodnotou „High“. Je však možné využít i negovaného výstupu, který je přítomen na modulu.

2.6.4 Mikrospínač

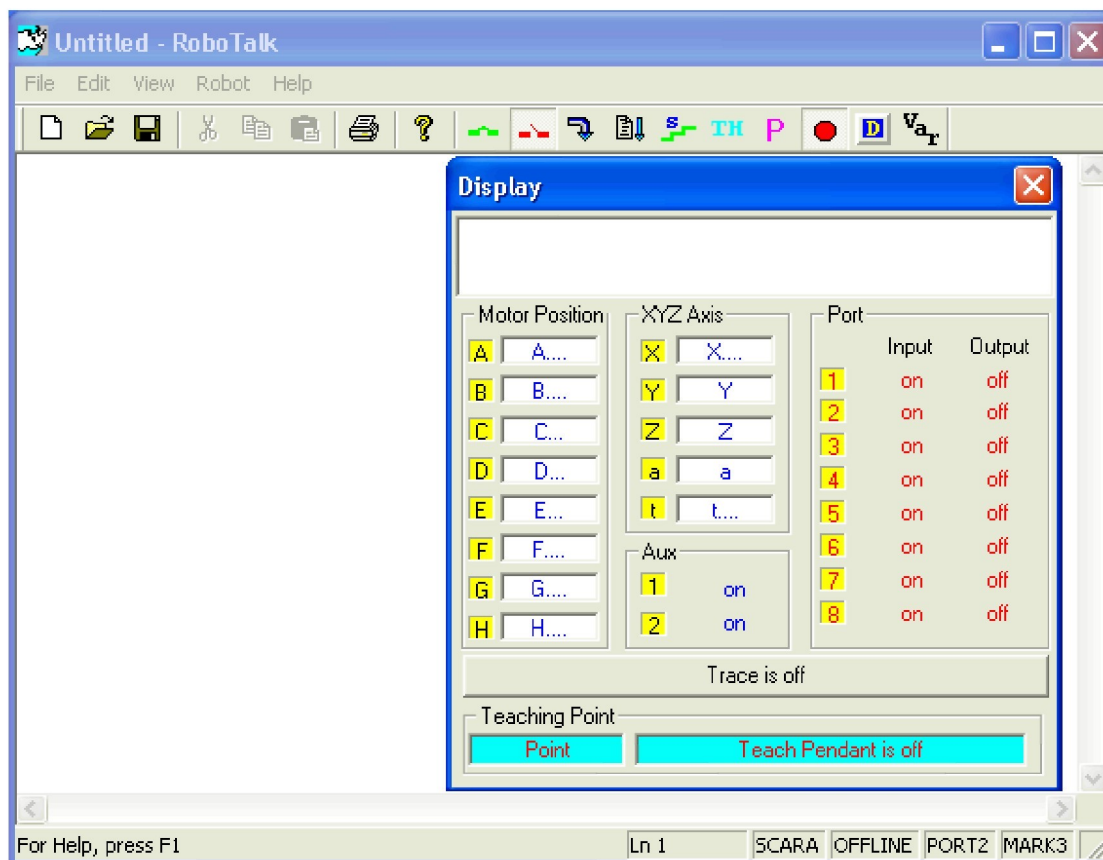
Mikrospínač a sadu připevňovacích kovových tyček je možné připevnit například k pásovému dopravníku. K detekci objektu dochází při jeho kolizi s mikrospínačem.

2.7 Programovací jazyk RoboTalk

RoboTalk je programovací jazyk a *RoboTalk pro Windows* aplikace pro ovládání robotů vytvořená speciálně pro roboty série XR a SCARA a také jejich příslušenství. Je možná kompletní kontrola všech funkcí jmenovaných robotů a řídicích systémů

Mark III a Mark IV. Jazyk má jen malé množství příkazů a díky tomu je velice snadné se jej naučit.

Aplikace *RoboTalk pro Windows* poskytuje interaktivní výukové prostředí, které obsahuje rozsáhlou nápovědu a zabudovaný debugger. Obsahy všech proměnných, pozic motorů, vstupní a výstupní signály mohou být v reálném čase zobrazovány na monitoru. Na následujícím obrázku je ukázka uživatelského prostředí aplikace.



Obrázek 5: Uživatelské rozhraní aplikace RoboTalk

2.7.1 Základní instrukce

Přehled několika základních instrukcí jazyka RoboTalk.

HARDHOME	Nalezne počáteční pozici robota pomocí koncových spínačů.
HOME	Pohyb do počáteční pozice.
OPEN	Otevře čelisti.
CLOSE	Zavře čelisti.
MOVE B,C,D,E,F	Relativní pohyb jednotlivých os.

MOVE TO B,C,D,E,F	Absolutní pohyb jednotlivých os.
MOVEX X,Y,Z,A,T	Relativní pohyb v kartézských souřadnicích.
MOVEX TO X,Y,Z,A,T	Absolutní pohyb v kartézských souřadnicích.

Ostatní instrukce a jejich podrobný popis použití je možné najít v oficiální dokumentaci k programovacímu jazyku RoboTalk [4].

2.7.2 Problémy s aplikací RoboTalk pro Windows

Se systémem Mark III je úspěšně navázána komunikace mezi hostitelským počítačem a řídicím systémem pomocí sériového rozhraní. Ovšem při vykonávání příkazů dochází v náhodných intervalech k „zamrznutí“ řídicího systému. Intervaly jsou někdy velmi krátké a k „zamrznutí“ dochází většinou v řadě jednotek instrukcí odeslaných do řídicího systému. Jediným východiskem z této situace je hardwarový reset celého systému. Při odhalování příčiny těchto problémů bylo testováno několik možností, ale žádná nevedla ke zlepšení situace nebo přímému odhalení příčiny problému.

- Kontrola kompatibility aplikace s použitou verzí operačního systému Windows XP. Podle oficiální dokumentace je aplikace plně podporovaná systémy Windows 9x, Windows ME, Windows NT, Windows 2000 a Windows XP.
- Výměna PC použitého pro ovládání řídicího systému.
- Kontrola a přeměření všech použitých kabelů a konektorů pro připojení řídicího systému.

S přihlédnutím ke všem okolnostem se musí s největší pravděpodobností jednat o hardwarový problém samotného řídicího systému.



Obrázek 6: Chyba komunikace s řídicím systémem Mark III

Systém Mark IV není schopen komunikovat s originální plnou verzí dodanou k řídicímu systému. Pokus o navázání komunikace skončí chybou o ztrátě spojení s řídicím systémem. Ovšem po stažení jiné verze aplikace z oficiálních stránek výrobce se

podařilo spojení navázat bez jakýchkoliv problémů. Bohužel se jedná o demoverzi a vzhledem k několika omezením při použití není vhodná pro vytváření komplexních úloh a slouží spíše pro prezentování produktu. Mezi výrazná omezení patří nemožnost uložení programu, což je možné vyřešit například kopírováním do vlastního souboru. Bohužel další omezení se týká nemožnosti spouštět rozsáhlejší programy s omezením na pár řádků.

2.8 Sériový port RS-232C a připojení řídicích systémů

RS-232C je standard pro sériový přenos dat, který je použit v počítačových sériových portech.

Komunikace mezi řídicím systémem a počítačem je možná pouze pomocí sériového portu. Řídicí systémy jsou vybaveny 25pinovou verzí portů, který již není přítomen na běžných PC, proto je nutné použít redukci na 9 pinů nebo si vytvořit vlastní kabel. Zapojení jednotlivých pinů je uvedeno ve dvou tabulkách pro oba typy řídicích systémů.

Konfigurace pro Mark III

Mark III používá ASCII kód, baudrate 9600, 7 data bitů, 2 stop bity, sudou paritu.

PC		Mark III
9 pinů – „samice“		25 pinů – „samec“
2	→	2
3	→	3
7	→	4
8	→	5
6	→	6
5	→	7
1	→	8

Tabulka 5: Zapojení RS-232C, PC (9 pinů) → Mark III (25 pinů)

Konfigurace pro Mark IV

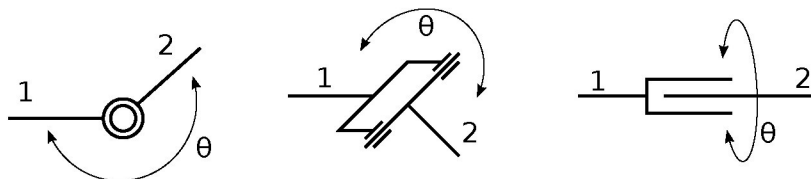
Mark IV používá ASCII kód, baudrate 9600, 7 data bitů, 2 stop bity, lichou paritu.

PC		Mark IV	
9 pinů – „samice“		25 pinů – „samec“	
3	→	2	
2	→	3	
7	→	4	
8	→	5	
6	→	6	
5	→	7	
1	→	8	

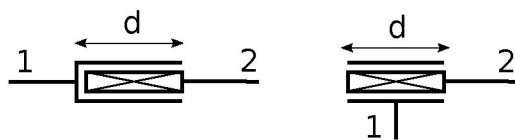
Tabulka 6: Zapojení RS-232C, PC (9 pinů) → Mark IV (25 pinů)

3 Kinematika robotů

Kinematika je vztah mezi pozicí, rychlostí a akcelerací kinematických členů manipulátoru (ramene) bez ohledu na fyzické příčiny pohybu. Dva spojené členy, které vykonávají vzájemný pohyb, se nazývají kinematická dvojice a spoj mezi nimi kloub. Nejběžněji používané typy kloubů jsou rotační a translační. Symbolické označení těchto kloubů je znázorněno na obrázcích 7 a 8.



Obrázek 7: Symbolické značení rotačních kloubů



Obrázek 8: Symbolické značení translačních kloubů

V kinematické analýze pozice manipulátoru jsou dva rozdílné problémy: Přímá úloha a inverzní úloha. Inverzní úloha se zabývá nalezením hodnoty kloubových souřadnic na základě znalosti polohy a orientace nástroje manipulátoru. Tato úloha je složitější a to i vzhledem k tomu, že nalezené řešení nemusí být jednoznačné. Řešení inverzní úlohy není předmětem diplomové práce a proto se dále budeme věnovat popisu a řešení přímé úlohy kinematiky.

3.1 Přímá úloha kinematiky

Přímá úloha kinematiky se využívá k popisu statické pozice a orientace kinematických dvojic manipulátoru. Existují dva způsoby výpočtu a popisu transformačních matic.

3.1.1 Brátova metoda

Tato metoda využívá kartézské souřadnice, které se skládají z pozice (x,y,z) a rotace reprezentované příslušnou transformační maticí. Matice se vypočítají pro každou

kinematickou dvojici a výsledná matice pro transformaci pozice mezi základním souřadným systémem (Base Frame) a souřadným systémem nástroje (Hand Frame) se získá součinem základních matic orientace a translace.

Podrobnější informace o této metodě jsou dostupné v připravovaných skriptech Robotiky od F. Novotného [7].

3.1.2 Denavit-Hartenbergova (DH) metoda

Zde dochází k zavádění souřadných systémů do každé kinematické dvojice dle pevně stanovených pravidel. K popsání vzájemného vztahu dvou souřadných systémů pak stačí jedna transformační matice.

Pravidla zavádění souřadných systémů

- osa z_{i-1} leží v ose pohybu i -té kinematické dvojice
- osa x_i je normálou k ose z_{i-1} , ve směru od osy z_{i-1}
- osu y_i je možné zvolit libovolně jen s podmínkou dokončení pravotočivého systému souřadnic

Po zavedení souřadných systému získáme čtyři následující parametry. Při jejich odečítání a výpočtu je nutné dodržet pořadí uvedené v tomto postupu.

Parametry DH matice

1. ε_i - úhel mezi x_{i-1} a x_i okolo osy z_{i-1}
2. s_i - vzdálenost mezi x_{i-1} a x_i ve směru osy z_{i-1}
3. a_i - vzdálenost mezi z_{i-1} a z_i ve směru osy x_i
4. α_i - úhel mezi z_{i-1} a z_i okolo osy x_i

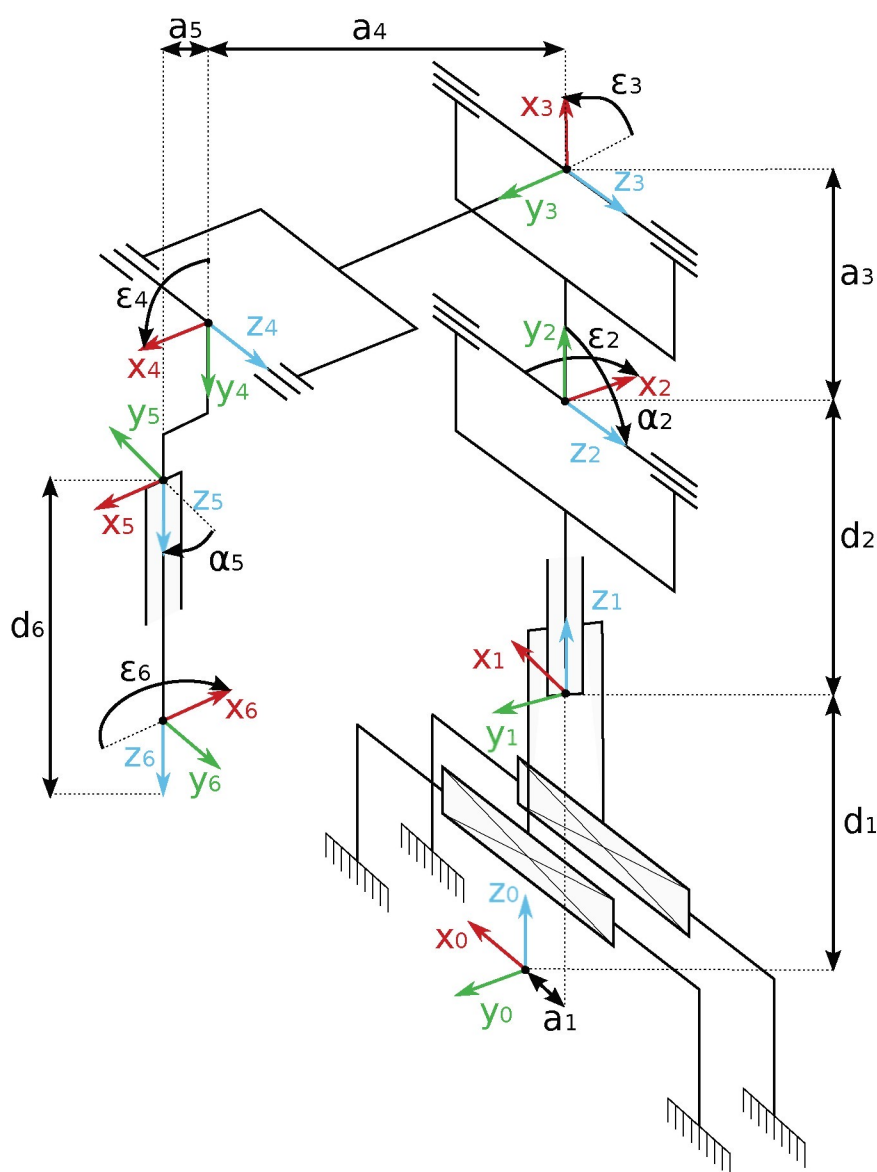
Pro každou kinematickou dvojici stačí jedna DH transformační matice ze vzorce č. 1, do které dosadíme příslušné parametry.

$$\mathbf{T}_i = \begin{vmatrix} \cos \cdot \varepsilon_i & -\sin \cdot \varepsilon_i \cdot \cos \cdot \alpha_i & \sin \cdot \varepsilon_i \cdot \sin \cdot \alpha_i & a_i \cdot \cos \cdot \varepsilon_i \\ \sin \cdot \varepsilon_i & \cos \cdot \varepsilon_i \cdot \cos \cdot \alpha_i & -\cos \cdot \varepsilon_i \cdot \sin \cdot \alpha_i & a_i \cdot \sin \cdot \varepsilon_i \\ 0 & \sin \cdot \alpha_i & \cos \cdot \alpha_i & s_i \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (1)$$

3.2 Vlastní výpočet kinematiky

Systém Mark IV disponuje integrovanou schopností pohybu robota v kartézském souřadném systému, ale při různém natočení jednotlivých os robota není schopen tuto polohu vrátit uživateli. Díky vlastnímu výpočtu je tak možné zjistit polohou a orientaci koncového bodu nástroje robota v jakémkoliv okamžiku.

Pro výpočet kinematiky robota XR-4 byla zvolena metoda pomocí DH matice. Do výpočtu je zahrnuta i translační osa, o kterou je robot rozšířen pomocí pohyblivé základny. Celkem se tedy jedná o výpočet pro 6 os. Pro přehlednost a snadnější orientaci poslouží následující náčrt celého systému i se všemi zavedenými souřadnými systémy a parametry nutnými pro výpočet všech matic kinematické úlohy.



Obrázek 9: Kinematika robota XR-4

Všechny souřadné systémy jsou zavedeny dle pevně daných pravidel pro zavádění souřadných systémů při využití DH matice.

Celkový počet souřadných systémů je 7 a pro přechody mezi nimi je tedy nutné použít 6 DH matic. Vzájemným roznásobením těchto 6 základních matic získáme výslednou matici pro převedení polohy a orientace od základního souřadného systému s číslem 0 až po souřadný systém nástroje robota s číslem 6.

V následující tabulce jsou vypsané všechny parametry, které určují podobu matic pro každou kinematickou dvojici. Hodnoty v prvním sloupci (kinematická dvojice) jsou složeny z čísel souřadných systémů. Tedy první číslice udává souřadný systém, ze kterého vycházíme, a druhá číslice výsledný souřadný systém.

Kinematická dvojice	ε_i	s_i	a_i	α_i
01	0	d_1	a_1	0
12	ε_2	d_2	0	α_2
23	ε_3	0	a_3	0
34	ε_4	0	a_4	0
45	0	0	a_5	α_5
56	ε_6	d_6	0	0

Tabulka 7: Proměnné DH matice

Další tabulka udává rozměry jednotlivých konstantních vzdáleností mezi klouby robota.

Konstanta	Hodnota
d_1	267,10 mm
d_2	97,63 mm
a_3	228,60 mm
a_4	228,60 mm
a_5	9 mm
d_6	169,86 mm

Tabulka 8: Konstantní hodnoty pro DH matici

Po dosazení všech konstant a upravení výrazů dostaneme následujících 6 matic polohy a orientace.

$$\mathbf{T}_{01} = \begin{vmatrix} 1 & 0 & 0 & a_1 \cdot 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 267,10 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (2)$$

$$\mathbf{T}_{12} = \begin{vmatrix} \cos \cdot \varepsilon_i & 0 & \sin \cdot \varepsilon_i & 0 \\ \sin \cdot \varepsilon_i & 0 & -\cos \cdot \varepsilon_i & 0 \\ 0 & 1 & 0 & 97,63 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (3)$$

$$\mathbf{T}_{23} = \begin{vmatrix} \cos \cdot \varepsilon_i & -\sin \cdot \varepsilon_i & 0 & 228,6 \cdot \cos \cdot \varepsilon_i \\ \sin \cdot \varepsilon_i & \cos \cdot \varepsilon_i & 0 & 228,6 \cdot \sin \cdot \varepsilon_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4)$$

$$\mathbf{T}_{34} = \begin{vmatrix} \cos \cdot \varepsilon_i & -\sin \cdot \varepsilon_i & 0 & 228,6 \cdot \cos \cdot \varepsilon_i \\ \sin \cdot \varepsilon_i & \cos \cdot \varepsilon_i & 0 & 228,6 \cdot \sin \cdot \varepsilon_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (5)$$

$$\mathbf{T}_{45} = \begin{vmatrix} \cos \cdot \varepsilon_i & 0 & \sin \cdot \varepsilon_i & 9 \cdot \cos \cdot \varepsilon_i \\ \sin \cdot \varepsilon_i & 0 & -\cos \cdot \varepsilon_i & 9 \cdot \sin \cdot \varepsilon_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (6)$$

$$\mathbf{T}_{56} = \begin{vmatrix} \cos \cdot \varepsilon_i & -\sin \cdot \varepsilon_i & 0 & 0 \\ \sin \cdot \varepsilon_i & \cos \cdot \varepsilon_i & 0 & 0 \\ 0 & 0 & 1 & 158,75 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (7)$$

Výsledná poloha se vypočítá dosazením aktuálních hodnot proměnných a vynásobením všech 6 matic.

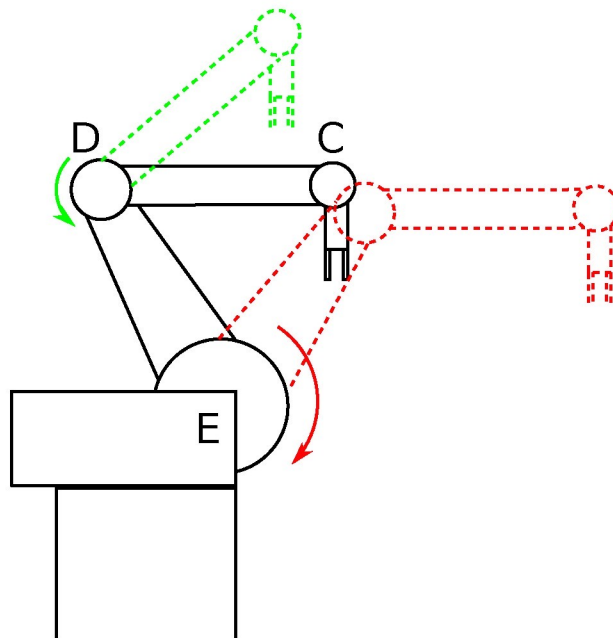
$$\mathbf{T}_{06} = T_{01} \cdot T_{12} \cdot T_{23} \cdot T_{34} \cdot T_{45} \cdot T_{56} \quad (8)$$

Aktuální pozici jednotlivých os získáme odečtením pozice motoru a převedením hodnoty z jednotek enkodéru na stupně dle převodní tabulky č. 9.

osa	dvojice	proměnná	převodní poměr	jednotky
H	01	a_1	27.899 / 1	mm
F	12	ε_2	17.637 / 1	stupně
E	23	ε_3	35.274 / 1	stupně
D	34	ε_4	35.274 / 1	stupně
C	45	ε_5	35.274 / 1	stupně
B	56	ε_6	12.800 / 1	stupně

Tabulka 9: Převod z jednotek enkodéru na stupně a mm

Vzhledem k řešení převodů u robota XR-4 je zapotřebí ještě několik úprav při výsledném výpočtu na reálném robotu. Při pohybu osy E dochází k automatické rotaci v ose D o stejný počet stupňů. To samé platí pro pohyb osy D a na ní závislé pootočení osy C. Na obrázku 10 jsou červenou barvou znázorněny pohyby robota při natočení osy E a zelenou barvou při natočení osy D.



Obrázek 10: Pohyb os při změně natočení os E a D

Ovšem k těmto závislým pohybům dochází díky řetězovým převodům a nedojde k pootočení samotného motoru dané osy. Proto není možné toto natočení zjistit odečtením hodnoty enkodéru motoru. Řešením je k pohybu osy D přičíst pohyb osy E a k pohybu osy C pohyb osy D.

Pokud vše symbolicky znázorníme bude mít závislost pro výpočet natočení osy D a C takovýto tvar:

$$\begin{aligned}\text{rotD} &= \text{motD} / 35.274 + \text{motE} / 35.274; \\ \text{rotC} &= \text{motC} / 35.274 + \text{motD} / 35.274;\end{aligned}$$

Proměnné *mot* obsahují hodnotu natočení motoru v jednotkách enkodéru načtenou z řídicího systému. Po jejím vydělení převodním poměrem z tabulky 9 získáme natočení ve stupních. Po sečtení příslušných hodnot získáme reálné natočení os D a C.

4 Vývoj vlastní aplikace – RhinoTalk

Pro vlastní realizaci úlohy použitelné k ovládání robotických systémů RHINO Robotics bylo nejdříve zapotřebí prostudovat a otestovat možnosti spolupráce těchto systémů s počítačem.

Jak bylo zmíněno v sekci 2.7.2, dochází při komunikaci s řídicím systémem Mark III k výrazným problémům. Tyto problémy se bohužel netýkají pouze aplikace *RhinoTalk pro Windows*, ale dochází k nim i u ostatních aplikací a pokusů o komunikaci. Z tohoto důvodu bylo na základě konzultace s vedoucím a konzultantem diplomové práce rozhodnuto opustit od realizace aplikace pro tento řídicí systém. Aby však nebyla diplomová práce ochuzena o spolupráci obou robotů a řídicích systému, bude systém Mark III ovládán a programován pomocí ovládacího panelu.

Celá aplikace je tedy určena pro řídicí systém Mark IV s robotem XR-4, který nevykazoval problémy se stabilitou při připojení k řídicímu systému. Komunikační problém tohoto řídicího systému s aplikací *RoboTalk pro Windows*, také popsáný v sekci 2.7.2, je softwarového rázu, a proto nemůže ovlivnit komunikaci pomocí nově vytvořené aplikace.

Celá aplikace byla vyvíjena v prostředí Borland C++ Builder a důvody, proč bylo vybráno právě toto prostředí, jsou shrnuty v následující kapitole 4.1.

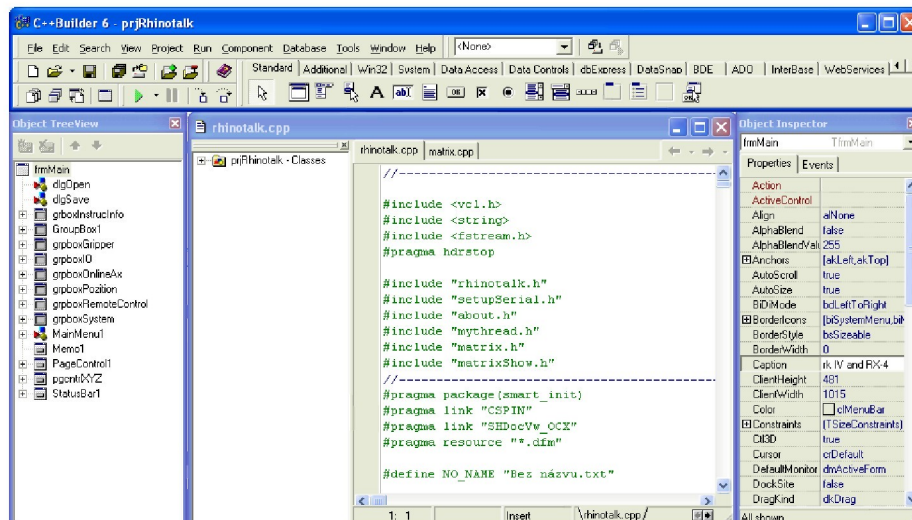
4.1 C++ Builder

C++ Builder je grafické vývojové prostředí, vyvinuté společností Borland a od roku 2009 vlastněné společností Embarcadero Technologies, pro vývoj programů v programovacím jazyku C++. C++ Builder je kombinací VCL (Visual Component Library), rozhraní vytvořeného v Delphi a C++ kompilátoru. VCL je vizuální komponentový objektově orientovaný framework pro vývoj aplikací pro MS Windows.

Vývojové prostředí je vybaveno funkcí WYSIWYG (What You See Is What You Get) a je tedy možné pomocí vizuálních komponent vytvářet rozhraní aplikace pouhým přetažením z nabízené palety.

Vývojové prostředí bylo zvoleno především na základě jednoduchosti návrhu uživatelského rozhraní a velkého množství dostupné dokumentace v podobě internetových článků i odborných knih.

Mezi nevýhody tohoto řešení může patřit například nižší optimalizace výsledného kódu a možnost běhu vytvořené aplikace pouze na operačních systémech MS Windows.



Obrázek 11: Vývojové prostředí C++ Builder

4.2 Technické parametry aplikace

Popis jakým způsobem byly technicky řešeny důležité části aplikace. Cílem této sekce není popsat řešení celé aplikace, ale jedná se pouze o některé části, které jsou nějakým způsobem důležité a zajímavé.

4.2.1 Nastavení sériového portu

Pro vytvoření sériového portu jsou využity funkce WIN32 API. K otevření portu COM je využita funkce *CreateFile*. Proměnná *comName* obsahuje jméno portu („COM1“, „COM2“,...). Pokud se otevření portu nezdaří zobrazí se se chyba s upozorněním „Nelze otevřít COM“.

```
frmMain->hComm = CreateFile( frmMain->comName,
    GENERIC_READ | GENERIC_WRITE,
    0,
    0,
    OPEN_EXISTING,
    0,
    0);

if (frmMain->hComm == INVALID_HANDLE_VALUE) {
    showError(„Nelze otevřít COM“);
    return 0;
}
```

Pomocí struktury Device Control Block (DCB) nakonfigurujeme přenosovou rychlost, paritu, počet datových bitů a stop bitů. Struktura DCB disponuje ještě dalšími parametry, které však nejsou v tomto použití důležité a proto se jimi nebudeme zabývat.

```
GetCommState(frmMain->hComm, &dcbSerialParams.dcb);  
dcbSerialParams.dcb.DCBlength = sizeof(dcbSerialParams.dcb);  
  
dcbSerialParams.dcb.BaudRate = frmMain->baudRate;  
dcbSerialParams.dcb.ByteSize = frmMain->byteSize;  
dcbSerialParams.dcb.StopBits = frmMain->stopBits;  
dcbSerialParams.dcb.Parity = frmMain->parity;
```

Ošetření běhu programu pokud dojde k nějaké chybě při nastavování parametrů sériového portu. Při neúspěchu zobrazení zprávy „Nepodařilo se nastavit COM“.

```
if(!SetCommState(frmMain->hComm, &dcbSerialParams.dcb)) {  
    CloseHandle(frmMain->hComm);  
    frmMain->hComm = NULL;  
    showError(„Nepodařilo se nastavit COM“);  
    return 0;  
}
```

Po úspěšném nastavení sériového portu a všech parametrů je možné začít odesílat instrukce do řídicího systému pomocí funkce *WriteFile*.

```
WriteFile(hComm, data, StrLen(data), &BytesWrite, NULL);
```

Parametry funkce *WriteFile*:

1. Identifikátor (handl) sériového portu.
2. Proměnná data je pole typu char a obsahuje příkaz ukončený znakem odřádkování \n.
3. Počet bytů k odeslání.
4. Ukazatel na počet odeslaných bytů.
5. Ukazatel na strukturu pro overlapped I/O.

Samozřejmě je také možné data z řídicího systému přijímat. K tomu slouží funkce *ReadFile*.

```
ReadFile(hComm, received, sizeof(received), &BytesRead, NULL );
```

Parametry funkce *ReadFile*:

1. Identifikátor (handl) sériového portu.
2. Proměnná *received* je pole typu *char*.
3. Počet bytů k přečtení.
4. Ukazatel na počet přijatých bytů.
5. Ukazatel na strukturu pro overlapped I/O.

4.2.2 Testování nečinnosti při běhu programu

Pro vykonání pohybu robota je nejdříve nutné zadat příslušné parametry. U pohybu jednotlivých os se jedná především o hodnotu polohy jednotlivých motorů. Pohyb v kartézských souřadnicích je definován souřadnicemi X,Y a Z, nakloněním A a rotací T. Pro oba druhy pohybů je samozřejmě možné nastavit např. rychlost a akceleraci pohybu.

Řídicí systém disponuje pro každou hodnotu pouze jedním registrem. Vzhledem k tomu není možné uložit hodnoty aktuálního pohybu a během jeho vykonávání odesílat řídicímu systému hodnoty pro další pohyb. Musí se tedy testovat dokončení aktuálního pohybu a po jeho dokončení začít s odesíláním konfigurace pro následující pohyb robota. Tímto postupem bohužel vzniká prodleva po každém pohybu způsobená čekáním na dokončení konfigurace pohybu.

Funkce pro zjištění stavu řídicího systému odesláním instrukce *SS*.

```
int TfrmMain::systemState (void){
    frmMain->send(„SS\n");           //dotaz na stav systému
    char received[4] = „”;
    ReadFile(frmMain->hComm, received, sizeof(received),
    &frmMain->BytesRead, NULL );      //načti odpověď na dotaz
    if(received[0] == '\\0') return 1; //pokud nedostane odpověď
    int i = atoi(received);           //konverze z char[] na int
    return i;                         //vrať i
}
```

Odpovědí systému je hodnota v dekadickém tvaru (0-255). Po jejím převedení na binární vyjádření označují jednotlivé bity aktuální stav a konfiguraci. Bit číslo 7 oznamuje zda je stále vykonáván nějaký pohyb (aktivita některého motoru). Proto tento bit otestujeme funkcí *moveFinished*.

```
int TfrmMain::moveFinished (void){
    int i = systemState(); //funkce pro zjištění stavu systému
    if (i == 1) return 1;  //pokud nepřišla odpověď, vrať 1
    i = i & 128;           //bit 7 - indikace aktivity motoru
    return i;              //0 - nepohybuje se, 1 - pohybuje se
}
```

4.2.3 Běh programu ve vláknu

Při spuštění vykonávání programu pro řízení robota je zapotřebí, aby uživatel měl možnost program zastavit nebo přerušit. Pokud by vykonávání programu běželo v hlavním vlákně aplikace, nebyla by možnost sledovat aktivitu uživatele, např. stisk tlačítka, protože by došlo k zablokování hlavního formuláře aplikace běžícím procesem. Z tohoto důvodu bylo vytvořeno samostatné vlákno, které řídí běh programu a zároveň je možné v hlavním vlákně testovat přerušení běhu programu uživatelem. Po přerušení běhu se vlákno ukončí, ale dojde k uložení potřebných informací k navázání běhu programu.

4.2.4 Změna z pohybu jednotlivých os na kartézské souřadnice

Při střídání pohybu jednotlivých os a pohybu v kartézských souřadnicích je nutné vykonávat příkaz *CC,d* pro transformaci hodnot v registrech pozice.

- [*d* = 0] Do registrů absolutní pozice budou uloženy hodnoty odpovídající aktuální pozici v kartézských souřadnicích (XYZ).
Vykonat při změně z kartézských souřadnic (*MX*) na pohyb jednotlivých os (*MI*, *MC*).
- [*d* = 1] Do registrů XYZ pozice budou uloženy hodnoty odpovídající aktuálnímu pozicím enkodéru.
Vykonat při změně z pohybu jednotlivých os (*MI*, *MC*) na kartézské souřadnice (*MX*).

Při vykonávání online pohybu robota jsou tyto konverze v aplikaci automaticky ošetřeny a je možné oba typy pohybů libovolně střídat. Ve vlastním programu pro

řízení robota je ovšem nutné, aby tyto změny ohlídal sám uživatel a konverzi hodnot provedl, kdy je nutné.

Co vše tedy nakonec aplikace umožňuje a jak vypadá její uživatelské rozhraní, si popíšeme v následující části.

4.3 Připojení řídicího systému Mark IV k PC

Navázání spojení pomocí sériového rozhraní RS-232C včetně nastavení jeho parametrů. Parametry jsou ukládány v souboru a jsou tedy dostupné v nezměněné podobě i po dalším spuštění aplikace.

4.4 Online ovládání robota

Online nebo-li okamžitý pohyb robota, který ihned reaguje na stisk tlačítka nebo zadání parametru hodnotou.

4.4.1 Ovládání jednotlivých os

V tomto online režimu je každá osa robota ovládána zvlášť. Osy B až F jsou rotační a velikost kroku je udávána ve stupních pootočení. Osa G je z neznámé příčiny nefunkční a k ose H je připojena posuvná základna, proto je velikost kroku této osy udávána v mm.



Obrázek 12: Modul ovládání jednotlivých os

4.4.2 Ovládání v kartézských souřadnicích (XYZ)

Režim, ve kterém se využívá schopnosti řídicího systému Mark IV pohybovat robotem dle zadaných kartézských souřadnic. Tento způsob ovládání je rozdělen na

další dva režimy a to je relativní posun v jednotlivých směrech, nebo možnost zadat přesnou pozici robota ve všech souřadnicích současně.

Obrázek 13: Modul ovládání v kartézských souřadnicích

4.5 Ovládání vstupních a výstupních portů

Řídící systém Mark IV disponuje 8 přepínači a 8 vstupy, u kterých je možné sledovat aktuální stav znázorněný ve formě červených diod.

Switch	IN	OUT
1	☐	☐ On
2	☒	☒ On
3	☐	☐ On
4	☐	☐ On
5	☐	☐ On
6	☐	☐ On
7	☐	☐ On
8	☐	☐ On

Obrázek 14: Modul ovládání vstupů a výstupů

Osm výstupních portů je také možné libovolně nastavit pouhým kliknutím na přepínač **ON** u daného výstupu. Hodnota výstupu se opět znázorní v podobě zhasnuté nebo rozsvícené červené diody.

4.6 Tvorba kompletních programů

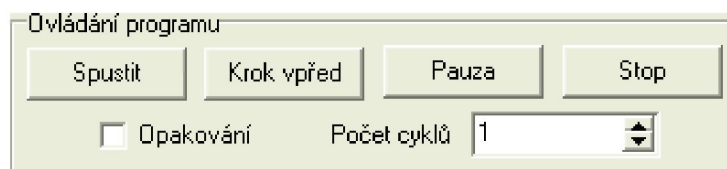
Další nedílnou součástí aplikace je jednoduchý editor programu, jehož součástí je seznam instrukcí rozdělený do 3 kategorií.

- **Nastavení** - ovládání a nastavení řídicího systému
- **Pohyb** - ovládání a nastavení pohybu robota
- **I/O** - ovládání výstupů a testování přepínačů a vstupů

K ulehčení programování pohybů slouží tlačítko **Generovat kód**. Po jeho stisku jsou do programu zapsány všechny instrukce potřebné k provedení pohybu do daného umístění.

4.7 Řízení běhu programu

K řízení běhu programu vytvořeného pomocí editoru slouží čtveřice tlačítek umístěných pod textovým editorem. Je také možnost nastavit cyklické opakování programu a počet cyklů.



Obrázek 15: Modul řízení běhu programu

- **Spustit** - Spustí celý program od začátku. V případě pozastavení programu tlačítkem **Pauza** pokračuje vykonávání od místa přerušení.
- **Krok vpřed** - Při každém stisku provádí vždy jen jeden řádek programu, aktuální řádek je zvýrazněn. Návrat na první řádek programu je možné provést stiskem tlačítka **Stop**.
- **Pauza** - Dokončí právě prováděný pohyb robota, a poté pozastaví vykonávání programu. V programu je možné pokračovat z místa přerušení pomocí opětovného spuštění tlačítkem **Spustit**.

- **Stop** - Ihned ukončí vykonávání programu i prováděný pohyb. Není možné pokračovat ve vykonávání programu z místa přerušení, ale pouze od začátku stiskem tlačítka **Spustit**.
- **Opakování** - Zapnutí nebo vypnutí opakování provádění programu.
- **Počet cyklů** - Nastaví počet opakování programu.

4.8 Zobrazení polohy a orientace nástroje

Pro zobrazení aktuální polohy nástroje robota je využita přímá úloha kinematiky vypočtená pomocí DH matice. Teoretický rozbor i praktický výpočet této metody byl uveden v kapitole č. 3 - Kinematika robotů.

Aktuální poloha všech os se zobrazuje v modulu, v pravé části rozhraní aplikace, nazvaném „Pozice robota a nástroje“.

The screenshot shows a window titled "Pozice robota a nástroje" with a light green background. It displays the following data:

B:	0	stupňů
C:	0	stupňů
D:	0	stupňů
E:	0	stupňů
F:	0	stupňů
G:	-	mm
H:	0	mm
X:	0	mm
Y:	109,769	mm
Z:	384,388	mm

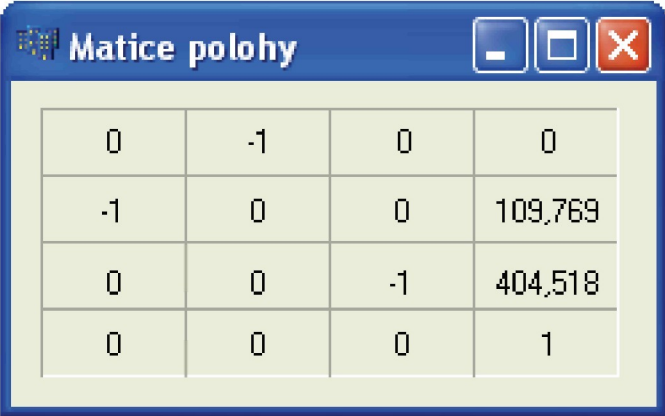
At the bottom of the window is a button labeled "Obnovit".

Obrázek 16: Modul polohy nástroje a robota

Zobrazení kompletní matice polohy vypočítané pomocí DH matice je možné nalézt v menu.

Robot → Matice polohy

Po zvolení této položky se otevře nové okno s grafickým zobrazením celé matice (viz. obrázek č. 17 na další straně).



0	-1	0	0
-1	0	0	109,769
0	0	-1	404,518
0	0	0	1

Obrázek 17: Matice polohy

4.9 Kalibrace posuvné základny robota

Posuvná základna umožňující rozšíření pohybu robota v ose x není originální součástí robota, proto při vykonání standardních funkcí pro nalezení počáteční pozice robota nedojde k její kalibraci.

Posuvná základna je vybavena koncovým spínačem, a proto je možná její kalibrace do počáteční pozice. Pro nalezení této pozice je možné využít standardní funkci řídicího systému $HL,H,1$. Písmenem H je specifikován motor a číslo (0 nebo 1) na konci udává směr pohybu motoru při hledání koncového spínače. Po nalezení počáteční polohy vynulujeme enkodér motoru pomocí příkazu AC,H .

Pokud je vykonáno nalezení počáteční pozice robota pomocí menu v aplikaci je ošetřeno i automatické vykonání kalibrace osy H s posuvnou základnou.

Robot → Nastavit Hard Home

Stejně tak je ošetřeno nastavení **SoftHome**, které uloží aktuální polohu robota i s polohou motoru H . Položky **Přejít do Hard Home** a **Přejít do Soft Home** samozřejmě také obsahují i pohyb osy H do dané pozice.

Při vykonávání instrukcí počáteční pozice (Hard Home) a uživatelské počáteční pozice (Soft Home) při běhu programu je nutné osu H ošetřit dle vlastních potřeb. Vyhledání počáteční pozice může vypadat například:

HH	- počáteční pozice robota (Hard Home)
PR,H,-750	- posun osy H záporným směrem
HL,H,1	- nalezení počáteční pozice v pozitivním směru
AC,H	- nulování enkodéru motoru H

Pohyb do počáteční pozice:

HA	- pohyb robota do počáteční pozice
PD,H,0	- pohyb motoru H do počáteční pozice

V průběhu vývoje aplikace se průběžně přidávaly nové funkce i odstraňovaly nedostatky a chyby. Velkou měrou a zpětnou vazbou se podíleli vedoucí a konzultant, se kterými byla aplikace pravidelně kontrolována a testována. Spousta dalších nedostatků a chyb byla také odhalena díky realizaci sady výukových úloh.

Ovládání vytvořené aplikace RhinoTalk a její použití při reálném řízení robota je podrobně popsáno u výukových aplikací v kapitole č. 5.

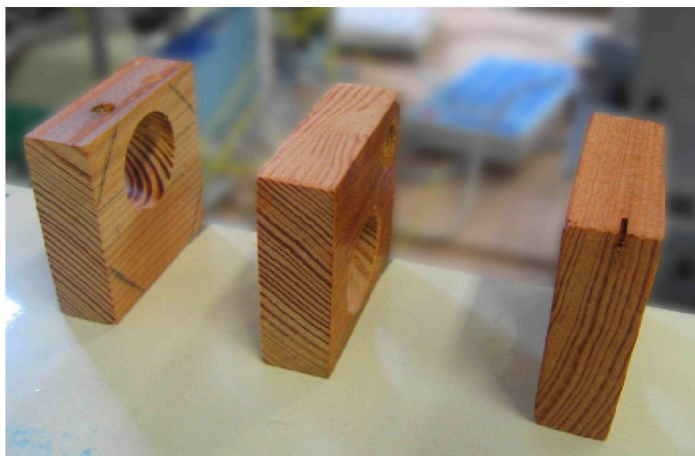
5 Výukové úlohy

Kapitola výukové úlohy obsahuje popis tvorby sady rozdílných úloh. V úlohách je využito především vytvořené aplikace RhinoTalk, ale nedílnou součástí je i programování pomocí ovládacích panelů robotů. Typy úloh byly sestaveny a vybrány tak, aby došlo k seznámení se všemi důležitými možnostmi řízení robotických systémů RHINO včetně všech dostupných periferních zařízení.

5.1 Základní manipulace pomocí softwaru RhinoTalk

V této úloze budou představeny základní možnosti ovládání a programování pomocí softwaru RhinoTalk pro řídicí systém Mark IV a robota XR-4. Úkolem je přenést tři dřevěné kostky a postavit je na sebe.

Dřevěné kostky rozestavím například na plochu rotačního karuselu s dostatečným rozestupem, aby bylo možné ke každé kostce přistoupit pomocí nástroje robota.



Obrázek 18: Dřevěné kostky

Zkontrolujeme zapojení řídicího systému a robota. Řídicí systém musí být připojen sériovým kabelem s počítačem. Do portů na řídicím systému připojíme odpovídající přívodní kabely všech motorů robota a pojezd do portu H.

Spustíme software RhinoTalk a v menu zvolíme:

Nastavení → Navázat spojení

Při dotazu na vykonání počáteční pozice (Hard Home) odpovíme **Ano** a necháme robota najít tuto polohu, pokud je připojen i pojezd robota, bude nalezena i počáteční pozice osy H s pojezdem. Po nalezení počáteční pozice je robot připraven k pohybu.

Občas dochází k situaci, kdy robot nereaguje na žádné příkazy pohybu. V tomto případě je možné resetovat nastavení pro daný typ robota (XR-4).

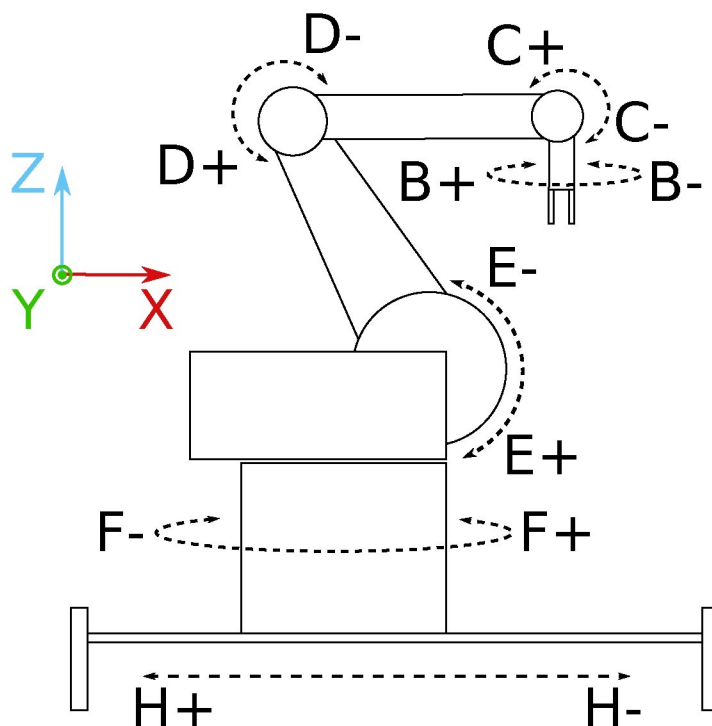
Řídící systém → Reset nastavení

Po vykonání tohoto příkazu zkusíme zadat znovu nalezení počáteční pozice a vše by mělo být v pořádku.

Robot → Nastavit Hard Home

Pohyb robota můžeme vykonávat pomocí pohybu jednotlivými osami nebo pohybem v kartézských souřadnicích (XYZ). Ovládání pomocí souřadnic XYZ je rychlejší a jednodušší, ale například pokud je zapotřebí dosáhnout nějaké složitější pozice, bude výhodné využít natáčení jednotlivých os.

Přehled směrů otáčení všech os a orientace kartézského souřadného systému si ukážeme na přehledném obrázku 19. U každé osy robota je znázorněna šipka pohybu s popisem s kladným nebo záporným znaménkem. Kladný symbol určuje, jakým směrem se bude osa pohybovat, pokud bude motorem otáčeno v kladném směru, tedy bude docházet k přičítání jednotek enkodéru daného motoru. U záporného znaménka je tomu logicky naopak. V aplikaci RhinoTalk je směr pohybu dané osy také znázorněn kladným nebo záporným znaménkem.



Obrázek 19: Přehled směrů pohybů os robota XR-4

Pro začátek, než si zvykneme na obsluhu robota, bude výhodné nastavit rychlost a akceleraci systému na nižší hodnoty než je přednastavených 50 %. Při neuváženém pohybu tak nedojde k větším škodám na okolním vybavení nebo robotu samotném. V aplikaci RhinoTalk, v modulu **Globální nastavení systému**, nastavíme rychlost a akceleraci v procentech maximálních hodnot a potvrdíme tlačítkem **Odeslat nastavení**.

„Robota“ přemístíme nad první z připravených kostek. Stiskneme tlačítko **Generovat kód** a do editoru programu nám přibude nastavení všech os i s příkazem pro vykonání pohybu. Kód přidáný do programu bude vypadat například:

```
PD,B,-1558
PD,C,0
PD,D,-650
PD,E,2921
PD,F,-2173
PD,H,0
MC
```

Otevřeme čelisti pomocí tlačítka **A-Otevřít**. Kód *GO* pro otevření čelistí také přidáme do našeho programu. Najedeme robotem až do místa, kde budou čelisti svírat mezi prsty připravenou kostku. Opět vygenerujeme kód pohybu pomocí tlačítka **Generovat kód** a zavřeme čelisti tlačítkem **A-Zavřít**. Instrukci *GC* pro zavření čelistí přidáme do programu. Uchycenou kostku zvedneme výš nad ostatní tak, abychom nezavadili při dalším pohybu. Přesuneme se nad požadovanou pozici, vygenerujeme kód, a poté se spustíme níž až k ploše podkladu, na který kostku přesouváme. Vygenerujeme další bod pohybu do programu a přidáme instrukci otevření čelistí. Opustíme místo s přesunutou kostku, opět nejlépe nejdříve provedeme pohyb v ose z tak, aby jsme o přesunutou kostku nezavadili při dalším pohybu. Přesun zbylých dvou kostek provedeme téměř analogicky jako přesun první.

Celý program je možné najít na přiloženém CD k diplomové práci. Z programu je zřejmý každý krok pro vykonání této úlohy, a proto není nutné uvádět přehled všech kroků jednotlivě.

Nakonec můžeme program spustit tlačítkem **Spustit** nebo krokovat pomocí **Krok vpřed**. Při krokování se aktuální instrukce zvýrazní označením. Odpojení od řídicího systému při ukončení práce lze provést v hlavním menu.

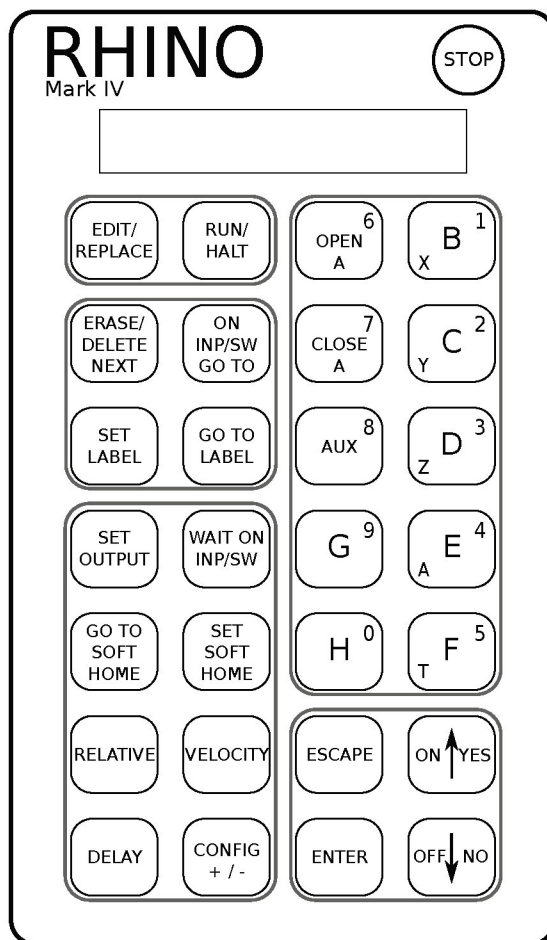
Nastavení → Odpojit

5.2 Základy ovládání pomocí ovládacího panelu

Jednoduchá úloha pro osvojení ovládání řídicího systému Mark IV a robota XR-4 pomocí standardního ovládacího panelu.

Cílem této úlohy je uchopení plastové tyčky ze stavebnice lego a její následné vsunutí do otvoru v plastovém kolečku taktéž ze stavebnice lego.

Na obrázku č. 20 je zobrazen ovládací panel řídicího systému Mark IV, který bude využit pro splnění této úlohy.



Obrázek 20: Ovládací panel – Mark IV

Rozložení tlačítek není příliš ergonomické a na první pohled nebude asi zřejmé jak panel použít k ovládání robota. I přesto však stačí jen několik málo minut a je možné porozumět většině funkcí, které ovládací panel poskytuje.

Po zapnutí systému tlačítkem na čelním panelu přejde systém automaticky do režimu *Play*, to znamená, že je možné řízení pomocí ovládacího panelu. Aby bylo možné s robotem správně manipulovat a došlo ke zpřístupnění všech možností ovládání, vykonáme příkaz pro nalezení počáteční pozice robota (Hard Home). Stisk-

neme tlačítko **CONFIG** $+/-$, šipkami v pravém dolním rohu najdeme položku **GO TO HARD HOME** a potvrdíme klávesou **ENTER**. Řídící systém spustí rutinu pro nalezení počáteční pozice robota.

Je vhodné zkontrolovat nastavení rychlosti pohybu robota, to provedeme pomocí tlačítka **VELOCITY**. Po stisku se zobrazí aktuální rychlost v rozmezí 0–100 %. Šipkami nastavíme novou hodnotu (50 %) a potvrdíme klávesou **ENTER**.

Nyní se pustíme do vytváření programu. Ten se vytváří v editačním režimu, do kterého se přepneme prvním tlačítkem v levém horním rohu – **EDIT/REPLACE**. Aktivace tohoto režimu je indikovaná slovem *EDIT* v pravém horním rohu displeje ovládacího panelu. Je možné přidávat jednotlivé kroky do programu, aktuální počet kroků programu se zobrazuje v pravém horním rohu displeje.

Přejdeme k prvním pohybům robota. Pro pohyb osy F stiskneme tlačítko **F** a šipkami můžeme pohybovat osou v obou směrech. Nastavím hodnotu -1209 . Stejným způsobem přepneme na osu E a nastavíme hodnotu 2476 a osu D zastavíme na hodnotě -600 . Po nastavení všech tří pohybů uložíme všechny do jednoho kroku programu stiskem klávesy **ENTER**. Instrukce kroku se uloží do programu a můžeme pokračovat dalšími pohyby. Hodnoty všech následujících kroků jsou přehledně utříděny v tabulce č. 10.

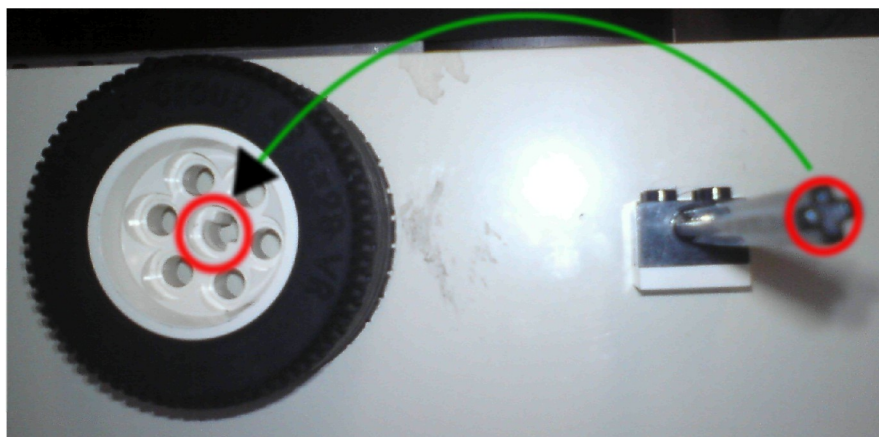
V prvním sloupci jsou jména os a v dalších sloupcích hodnota natočení dané osy v jednotkách enkodéru, které používá ovládací panel. Proškrtnutí hodnoty znamená, že v daném kroku se hodnota natočení osy nemění a zůstane tedy stejná jako v kroku předchozím. Osa A jsou čelisti robota proto není hodnota uvedena v jednotkách, ale slovně udává jestli mají být čelisti otevřeny nebo zavřeny.

Na startu programu počítáme s tím, že robot je uveden do počáteční pozice.

kroky programu									
osa	1	2	3	4	5	6	7	8	9
A	-	-	zavřít	-	-	-	-	otevřít	-
B	-	-	-	-	-	-	-	-	-
C	-	-	-	-	-	-	-	-	-
D	600	-808	-	-331	-699	-813	-909	-	0
E	2476	2552	-	2280	-	2326	2374	-	0
F	-1209	-	-	-	1394	-	-	-	0
H	-	-	-	-	-	-	-	-	-

Tabulka 10: Kroky programu 2. úlohy pro Mark IV

Krokem č. 2 se dostaneme čelistmi robota nad připravenou tyčku a bude nutné jí uchytit. Stiskneme **CLOSE A** a potvrdíme klávesou **ENTER**. Dojde k uzavření čelistí a přidání dalšího kroku s pořadovým číslem 3 do našeho programu.



Obrázek 21: Plastová tyčka a kolečko

Dvěma dalšími kroky č. 4 a 5 se přesuneme s uchycenou tyčkou až těsně nad otvor v kolečku.

Abychom nyní dokázali vsunout plastovou tyčku do otvoru v kolečku musíme postupovat po malých krocích a střídat posun v ose E a D. Budeme se tedy přepínat mezi oběma osami a vždy vykonáme jen malý pohyb směrem dolů do doby než dosáhneme hodnoty 2374 pro osu E a -909 v ose D. Celý pohyb můžeme pro jistotu rozdělit přibližně v polovině pohybu stisknutím tlačítka **ENTER**.

Po vsunutí tyčky do kolečka otevřeme čelisti pomocí **OPEN A** a vrátíme robota do počáteční pozice klávesou **GO TO SOFT HOME**. V tabulce je tato pozice znázorněna nulováním polohy všech os, se kterými bylo v průběhu vykonávání programu pohybováno.

Editační režim opustíme stiskem **ESCAPE** a tlačítkem **RUN/HALT** spustíme vytvořený program.

Program je možné uchovat i po vypnutí systému uložením do paměti EEPROM. Vstoupíme do nastavení systému stiskem **CONFIG +/-** a zvolíme položku **TRANSFER MENU**, kde je možné uložit nebo nahrát program pomocí **SAVE USER** a **LOAD USER**.

5.3 Ovládání vstupů a výstupů

Následující úloha rozšíří znalosti z předešlých úloh o použití vstupů a výstupů. Po přenesení objektu robotem XR-4 s řídicím systémem Mark IV bude vyslána informace o provedení této činnosti, na kterou bude reagovat druhý řídicí systém Mark III s robotem SCARA. Jako mezipřechod mezi oběma roboty bude využito rotačního karuselu ovládaného z řídicího systému Mark III.

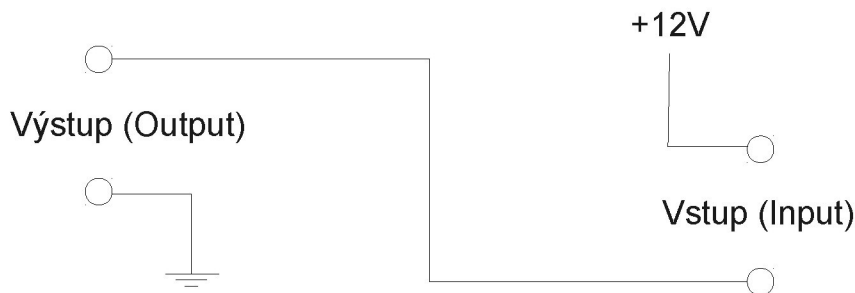
5.3.1 Zapojení vstupů a výstupů

Zapojení vstupů a výstupů se liší dle verze řídicího systému.

Mark III má 8 výstupů, které je možné programově přepínat mezi hodnotami 0 a 5 V a tím předávat signály externímu vybavení. Pro příjem signálu se využívá 8 vstupů, které rozlišují mezi hodnotami 0 a 5 V. Zapojení je velmi jednoduché, každý vstup nebo výstup obsahuje jednu svorku, do které se připojí vodič pro vstupní nebo výstupní informaci.

Mark IV disponuje 8 výstupy a 8 vstupy a každý z nich je složen ze dvou svorek. Vstup rozeznává hodnoty 0–2,5 V jako logické 0 (Low) a hodnoty nad 2,5 V jako logické 1 (High). Hodnota napětí na výstupu závisí na zapojení výstupu. Například je možné použít 12 V z čelního panelu řídicího systému nebo využít pouze propojení se zemí.

Zapojení systému Mark IV pro ovládání vstupu výstupem může vypadat takto:

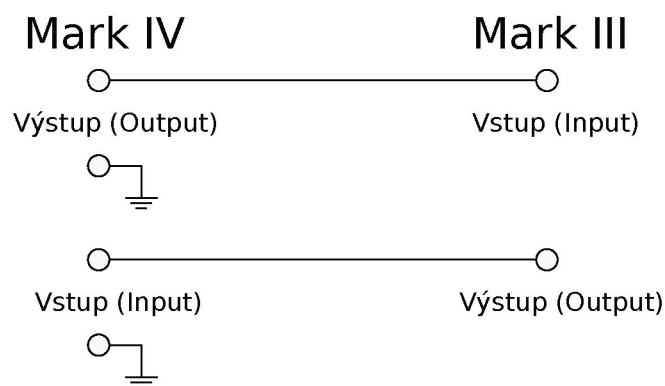


Obrázek 22: Zapojení výstupu na vstup pro Mark IV

Po sepnutí výstupu dojde k uzemnění a následnému sepnutí vstupu jehož druhá svorka je připojena k +12 V.

Pro předávání signálů mezi řídicím systémem Mark III a Mark IV použijeme zapojení dle obrázku č. 23 na další straně.

Rotační karusel připojíme standardním způsobem pomocí motorového konektoru například jako osu G. Jeho řízení bude tedy probíhat v režimu „polohy“.

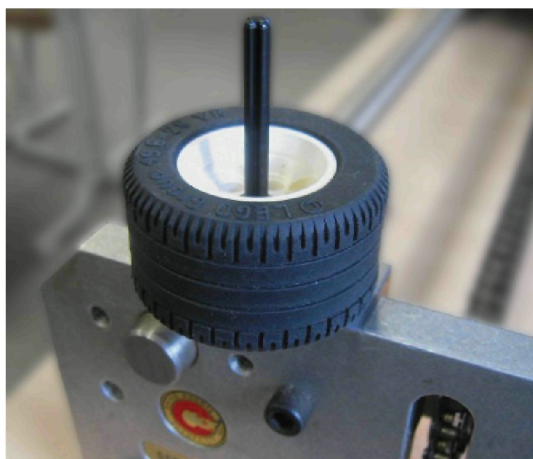


Obrázek 23: Zapojení vstupů a výstupů mezi Mark IV a Mark III

Nyní už je zřejmé jakým způsobem fungují vstupy a výstupy i jak propojit oba řídicí systémy a karusel a můžeme se vrhnout na programování pohybů obou robotů a karuselu.

5.3.2 Mark IV a robot XR-4

Ovládání robota XR-4 pomocí aplikace RhinoTalk bylo již popsáno v první výukové úloze. Proto již využitě operace a příkazy nebudou v této úloze popsány do takových detailů. Zaměříme se především na nové funkce a možnosti.



Obrázek 24: Pozice objektu určeného k přesunu

Kolečko s upevněnou tyčkou usadíme na roh základny tak, jak je zobrazeno na obrázku č. 24. Přemístíme se nad kolečko, vygenerujeme kód tohoto umístění, posuneme se až k tyčce z kolečka, vygenerujeme další bod, uzavřeme čelisti, kolečko nadzvedneme a přemístíme na karusel.

Po přesunu kolečka na karusel musíme nastavit výstup pro předání informace

řídícímu systému Mark III, který již bude na informaci čekat, aby mohl zahájit vykonávání pohybu. Nastavení výstupu provedeme instrukcí *OB*.

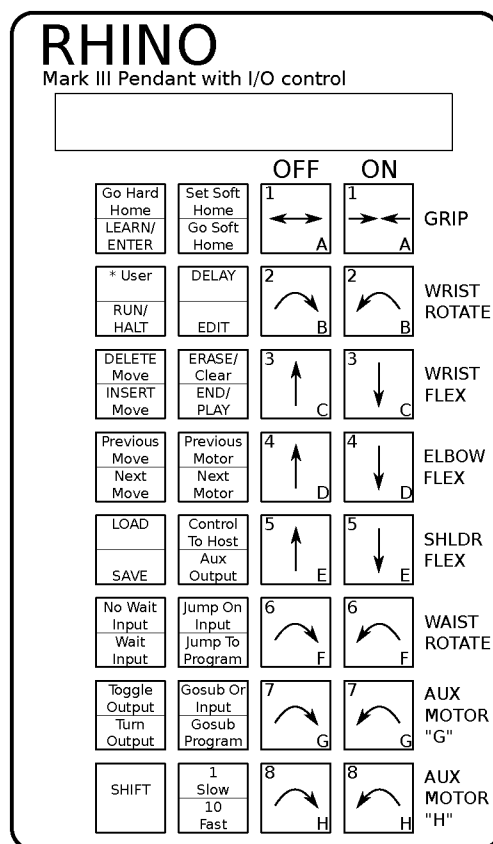
OB,1,1	- zapnutí výstupního portu 1
HA	- přesun robota do počáteční pozice
OB,1,0	- vypnutí výstupního portu 1

Zapnutím výstupu č. 1 dojde k propojení portů se zemí a tím snížení napětí na vstupním portu systému Mark III, který je v běžném stavu v hodnotě 5 V. Dojde tedy k překlopení stavu do hodnoty na kterou systém Mark III bude čekat.

Po předání této informace bude další manipulaci zajišťovat robot SCARA ovládaný pomocí ovládacího panelu systému Mark III.

5.3.3 Mark III a robot SCARA

Po spuštění řídícího systému Mark III se automaticky vykoná inicializace všech připojených motorů. Poté je nutné vyhledat počáteční pozici všech os robota (Hard Home). To se provede stisknutím tlačítka **Go Hard Home** na ovládacím panelu.

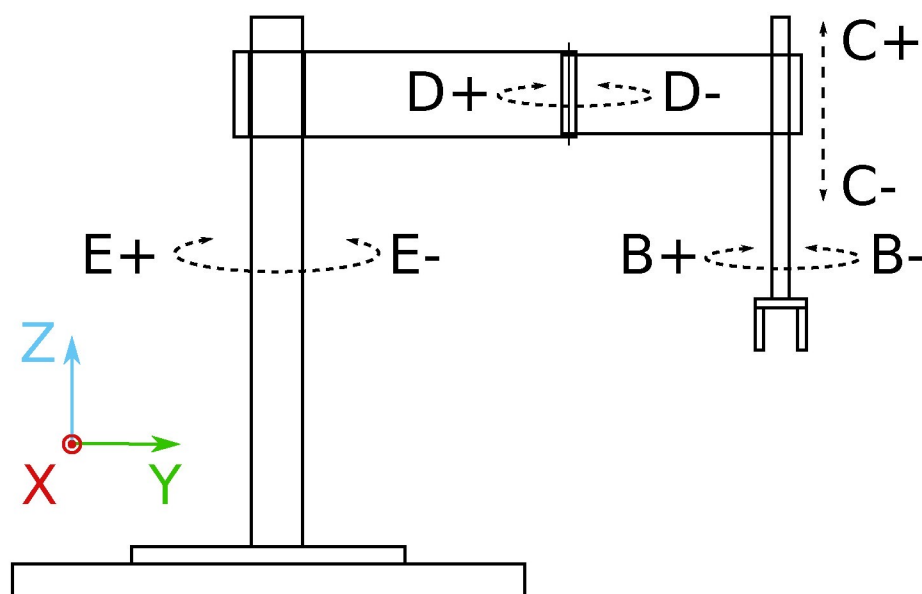


Obrázek 25: Ovládací panel – Mark III

U kláves se dvěma funkcemi se funkce uváděná v horní části tlačítka vykoná pomocí současného stisku klávesy **Shift**. Funkce ve spodní části se vykoná při běžném stisku samotného tlačítka.

Po nalezení počáteční pozice se můžeme přepnout do režimu tvorby programu a to tlačítkem **Learn/Enter**. Ovládání všech os se provádí osmy tlačítka v pravé části ovládacího panelu (A–H). Pro každou osu jsou dvě tlačítka pro kladný a záporný směr pohybu. Celý program bude velice jednoduchý a vystačíme s 6 kroky. Pro uložení jednotlivých kroků se používá tlačítko **Learn/Enter**. Jeden krok se může skládat z více pohybů a příkazů, např. současný pohyb os D a C i s testováním vstupu na určitou hodnotu. Pokud dochází ke kombinování příkazů pohybu a testování vstupů, čeká se nejdříve na dosažení požadovaného vstupu, a až poté se vykoná daný pohyb robota.

Dále si opět, stejně jako u robota XR-4, ukážeme směr pohybu všech os a orientaci kartézského souřadného systému na obrázku č. 26.



Obrázek 26: Přehled směrů pohybů os robota SCARA

To by mělo být dostačující na úvodní seznámení s řídicím systémem Mark III i robotem SCARA a nyní se pustíme do tvorby vlastního programu.

První krok bude obsahovat testování vstupu kvůli zajištění synchronizace pohybu obou robotů. Zatímco robot XR-4 bude přesouvat objekt na rotační karusel, řídicí systém Mark III čeká na dokončení této operace testováním hodnoty na vstupním portu. Po dokončení pohybu odešle řídicí systém Mark IV signál nastavením výstupního portu. Příkaz pro testování vstupu zadáme stiskem **Wait On Input** a

nastavením příslušných parametrů. Parametry se aktivují tlačítka **1–8** rozdělených do dvou sloupců podle hodnoty, na kterou se má čekat. Levý sloupec s nadpisem **OFF** znamená čekání na hodnotu logické 1 a pravý s nadpisem **ON** na hodnotu logické 0. Hodnota na vstupním portu č. 1 přejde po dokončení operace robota XR-4 do logické 1, proto nastavíme čekání na tuto hodnotu stiskem klávesy **1** z řady **OFF**. Operaci potvrdíme stiskem **Learn/Enter** a tím dojde k uložení prvního kroku programu.

Další kroky se již budou skládat pouze z instrukcí pohybu robota a karuselu, který je připojený k řídicímu systému Mark III. Karusel je připojen ke standardnímu portu motoru a tedy v režimu „pozice“, proto mu budou parametry pohybu zadávány stejným způsobem jako ostatním osám robota. Pohyb jednotlivých os se provádí stiskem příslušného tlačítka v pravé části ovládacího panelu. Rozlišení směru jednoduše výběrem tlačítka z řady **OFF** nebo **ON**. Karusel otočíme přibližně o 180° na hodnotu 790. Další osy a kroky jsou uvedeny v tabulce:

osa	kroky programu				
	2	3	4	5	6
A	-	-	zavřít	-	-
B	-	-	-	-	-
C	110	-	-	160	-
D	-	154	-	-	0
E	390	427	-	-	0
G	790	-	-	-	0

Tabulka 11: Kroky programu 3. úlohy pro Mark III

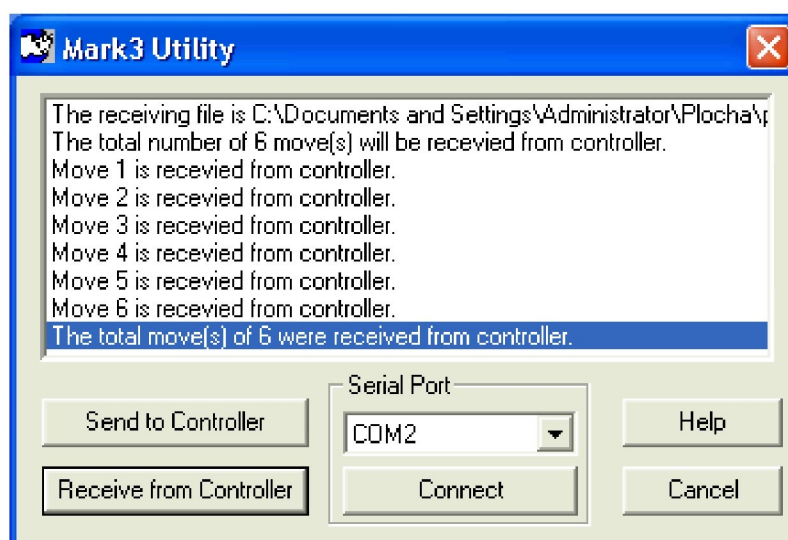
Tabulka je opět setříděna dle jednotlivých kroků, u kterých je pro každou osu udána hodnota natočení. A v případě osy A otevření a zavření čelistí.

Po nastavení pozice všech os pro jednotlivé kroky opět stačí stisknout klávesu **Learn/Enter** a dojde k uložení celého kroku do programu. Po uložení všech 6 kroků opustíme režim programování stiskem **End/Play**.

Při vykonávání úlohy nejdříve spustíme program na řídicím systému Mark III pomocí klávesy **Run/Halt**. Systém přejde ke kroku čekání na signál z řídicího systému Mark IV. Poté můžeme spustit i program určený pro systém Mark IV vytvořený pomocí aplikace RhinoTalk.

5.3.4 Záloha programu pro Mark III

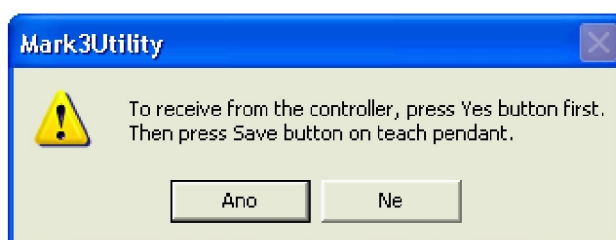
K ukládání programu z řídicího systému Mark III je k dispozici aplikace Mark3 Utility. S touto aplikací je možné program uložený v řídicím systému přenést přes sériový port do hostitelského počítače.



Obrázek 27: Mark3 Utility – uživatelské rozhraní

K navázání komunikace stačí nastavit správný port COM a stisknout tlačítko **Connect**.

Uložení programu do PC se provede tlačítkem **Receive from Controller**. Otevře se dialog, kde zadáme vlastní umístění a jméno pro nově vytvořený soubor, který přijmeme z řídicího systému.

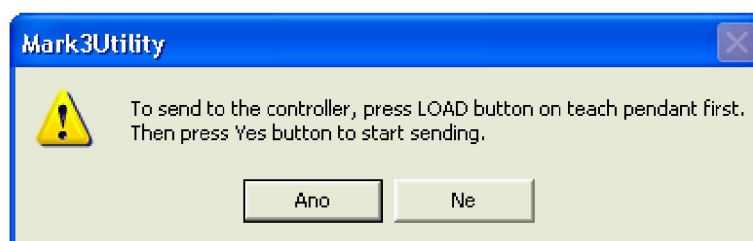


Obrázek 28: Mark3 Utility – ukládání programu do PC

Dále budeme navigování k tomu, abychom stiskli tlačítko **Ano** a následně klávesu **SAVE** na ovládacím panelu.

Nahrání programu do řídicího systému se vykoná tlačítkem **Send to Controller**. Po otevření dialogu vybereme soubor, který požadujeme nahrát do řídicího systému. V dalším kroku bude nutné nejdříve stisknout klávesu **LOAD** na ovládacím panelu

a poté potvrdit otevřené okno tlačítkem **Ano**. Všechny instrukce daného programu



Obrázek 29: Mark3 Utility – nahrávání programu do řídicího systému

budou postupně nahrány do řídicího systému.

Problém s komunikací s řídicím systémem Mark III bohužel ovlivňuje i tuto aplikaci, proto je možné, že občas dojde k zaseknutí řídicího systému. Pokud se tak stane u přenosu z počítače do řídicího systému nejedná se o významný problém. Systém restartujeme a program zkusíme přenést znovu. Ovšem při situaci, že program nahráváme ze systému do počítače a dojde k zaseknutí před dokončením přenosu, ztratíme vytvořený program, jelikož při resetování systému jsou všechna data ztracena.

5.4 Detekce objektů na pohybujícím se dopravníku

Čtvrtá úloha je zaměřena na využití pásového dopravníku a optického snímače. Na pás dopravníku bude přemístěn objekt pomocí robota XR-4 a dojde ke spuštění dopravníku. Optický snímač připevněný k dopravníku bude předávat informace řídicímu systému Mark IV. Při přerušení paprsku snímače předmětem dojde k zastavení dopravníku a detekovaný objekt bude přemístěn robotem SCARA.

5.4.1 Zapojení dopravníku a optického snímače

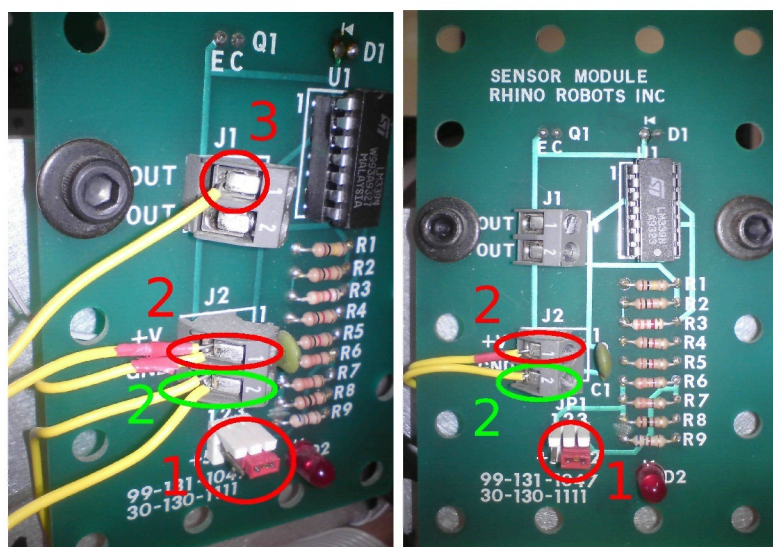
Nejprve je nutné správně připojit pásový dopravník a také optický snímač pro detekci pohybujícího se objektu.

Dopravník připojíme do režimu „rychlosti“ a to pomocí výstupu AUX. Ze svorek portu AUX-1 vyvedeme vodiče a připojíme na svorky motoru dopravníku. Po připojení je možné řídit rychlost pohybu dopravníku změnou napětí na výstupu AUX-1. Změna napětí se provádí příkazem *XS*. Parametry jsou číslo portu a velikost napětí v procentech maximálního napětí, které je 20 V.

XS,1,40 - AUX-1, 8V (40 % z 20V)

XS,2,-50 - AUX-2, -10V (50 % z -20V)

Optický snímač bude připojen k systému Mark IV, proto svorky pro změnu napájení nastavíme do polohy pro napájení 12 V. Nastavení přepínače je zvýrazněno červeným kroužkem s číslem 1 na obou částech obrázku č. 30.



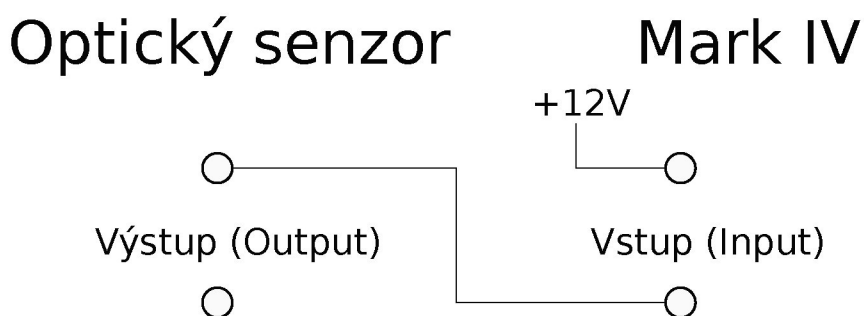
Obrázek 30: Zapojení optického snímače - oba moduly

Na levé části obrázku je první modul snímače do kterého je přivedeno napájení z řídicího systému.

- 2 zelená – uzemnění
- 2 červená – +12 V

Se stejným označením je provedeno i připojení druhého modulu. Napájení můžeme vyvést z první části modulu, takže nemusí být tažen vodič až z řídicího systému.

Číslem 3 je označen vývod výstupu, který připojíme na vstupní port číslo 2 řídicího systému Mark IV. Druhou svorku vstupního portu připojíme na +12 V. Při



Obrázek 31: Propojení výstupu snímače s řídicím systémem Mark IV

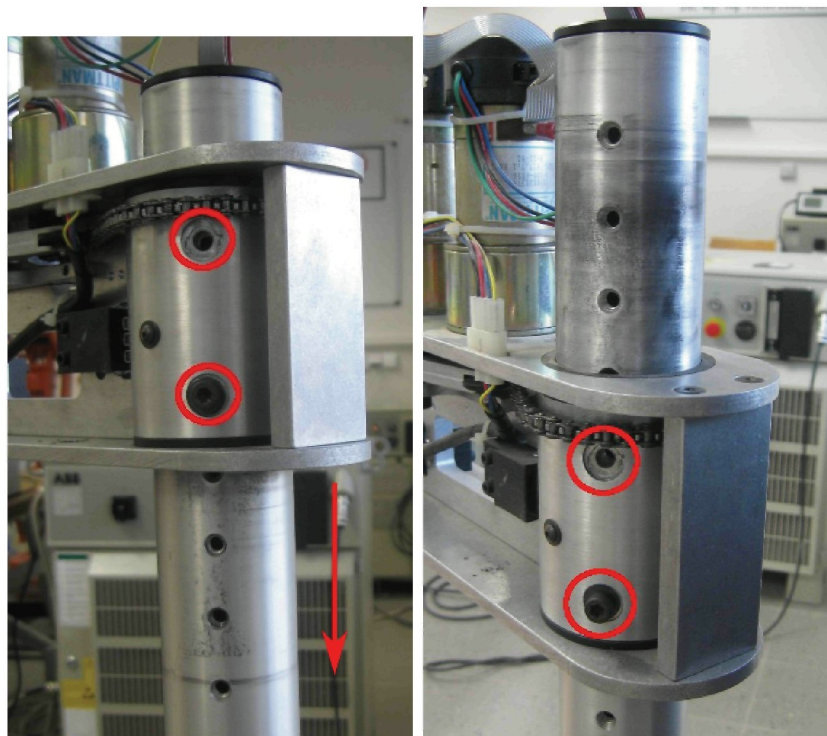
detekci objektu dojde k překlopení výstupu optického snímače do logické 0 a tím pádem také k přepnutí vstupu řídicího systému do hodnoty „High“.

5.4.2 Nastavení robota SCARA

Rameno robota SCARA při pohybu nedisponuje velikým rozsahem v ose z (8 cm), a proto je vybaveno manuálním vertikálním nastavením. Dosud jsme ve všech úlohách vystačili s původním nastavením, ale pás dopravníku je podstatně níž než plocha rotačního karuselu a proto budeme muset rameno přizpůsobit našim požadavkům.

Nastavení ramene je velice jednoduché a zabere jen pár okamžiků. Nejprve je nutné nastavit osu E do takové polohy, aby se odkryly dva imbusové šroubky v zadní části robota. Šroubky povolíme a tím dojde k uvolnění celého ramene. Lehkým přitlačením a popřípadě zakroužením ramenem nastavíme požadovanou polohu a šroubky opět dotáhneme.

Na levé části obrázku č. 32 je zobrazena původní poloha ramene. Vrchní šroub je připevněn do druhého závitu od vrchu.



Obrázek 32: Výškové nastavení robota SCARA

V pravé části je zobrazeno rameno po přesunutí o 3 polohy níž, vrchní šroub je v pátém závitu.

Nastavení všech zařízení v této úloze zabralo více času, ale nyní by vše mělo být připraveno pro zahájení operací s roboty.

5.4.3 MarkIV a robot XR-4

Robotem XR-4 přeneseme objekt (lego kolečko) doprostřed pásu dopravníku. Výchozí polohu předmětu můžeme zvolit například totožně s umístěním v úloze z kapitoly 5.3, dle obrázku č. 24. Po přenesení kolečka na dopravník otevřeme čelisti a kolečko upustíme. Robota kousek přesuneme, aby nedošlo ke shození objektu při jeho pohybu po dopravníku.

Přejdeme ke spuštění pohybu dopravníku řízeného výstupem AUX-1. Výstupní napětí zvolíme například 50 % (10 V).

XS,1,50

OB,1,1

Po spuštění dopravníku necháme systém čekat na detekci objektu optickým snímačem, který je připojený ke vstupu č. 2. Přítomnost objektu se projeví hodnotou logické 1 na tomto vstupu.

WI,2,1

Ihned po detekci objektu dopravník zastavíme udáním nulové hodnoty výstupního napětí.

XS,1,0

Informaci o detekovaném objektu předáme řídicímu systému Mark III nastavením výstupu č. 1 do hodnoty logické 1.

OB,1,1

Po předání informace přemístíme robota do počáteční pozice. Výstup vrátíme zpět do výchozí polohy, tedy logické 0.

OB,1,0

Tím jsou veškeré operace pro řídicí systém Mark IV a robota XR-4 vykonány a přemístění detekovaného objektu z dopravníku zajistíme pomocí robota SCARA a řídicího systému Mark III.

5.4.4 MarkIII a robot SCARA

Všechny pohybové operace robota SCARA jsou vypsány po jednotlivých krocích v tabulce č. 12. Programování opět provádíme pomocí ovládacího panelu, které už známe z předchozí úlohy.

osa	kroky programu											
	1	2	3	4	5	6	7	8	9	10	11	12
A	-	-	-	-	Zavřít	-	-	-	-	Otevřít	-	0
B	40	-	195	-	-	-	201	-	-	-	156	0
C	100	-	100	-57	-	82	-	-	-23	-	-	0
D	-393	-	-176	-	-	-	-306	-487	-	-	-	0
E	660	-	780	-	-	-	700	656	-	-	-	0

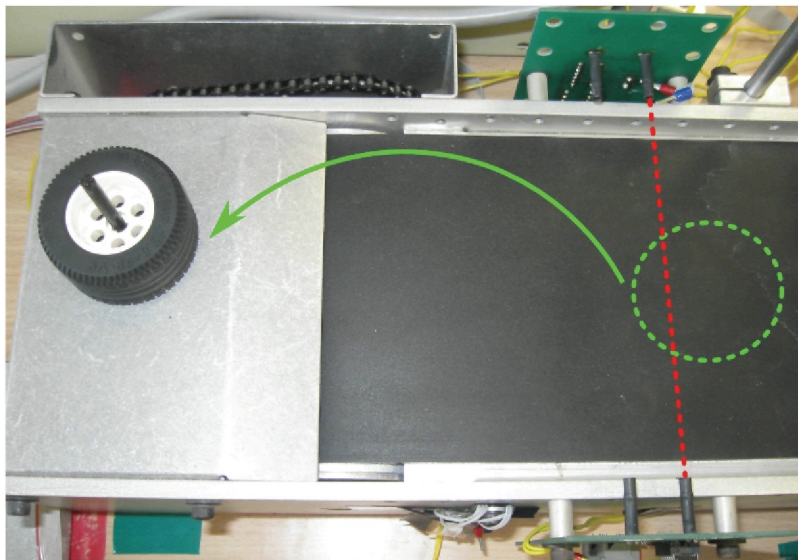
Tabulka 12: Kroky programu 4. úlohy pro Mark III

V prvním kroku z počáteční pozice přejdeme blíž směrem k dopravníku.

Druhý krok neobsahuje žádný pohybový příkaz, ale zadáme instrukci pro čekání na určitou hodnotu vstupu č. 1, na který nám přijde informace z druhého řídicího

systému. Stiskneme tedy nejprve klávesu **Wait Input** a následně **1 – ON**. Systém bude čekat na hodnotu vstupu *ON* (Low), která nastane uzemněním po zapnutí výstupu na řídicím systému Mark IV.

Po informaci o detekci objektu pokračujeme ve vykonávání dalších kroků dle tabulky, které zajistí přesunutí dopraveného objektu mimo pás dopravníku. Objekt bude umístěn například na nepohybující se plochu dopravníku.



Obrázek 33: Přemístění detekovaného objektu z dopravníku

Při spouštění úlohy opět nejdříve spustíme řídicí systém Mark III, který bude čekat na signál od systému Mark IV.

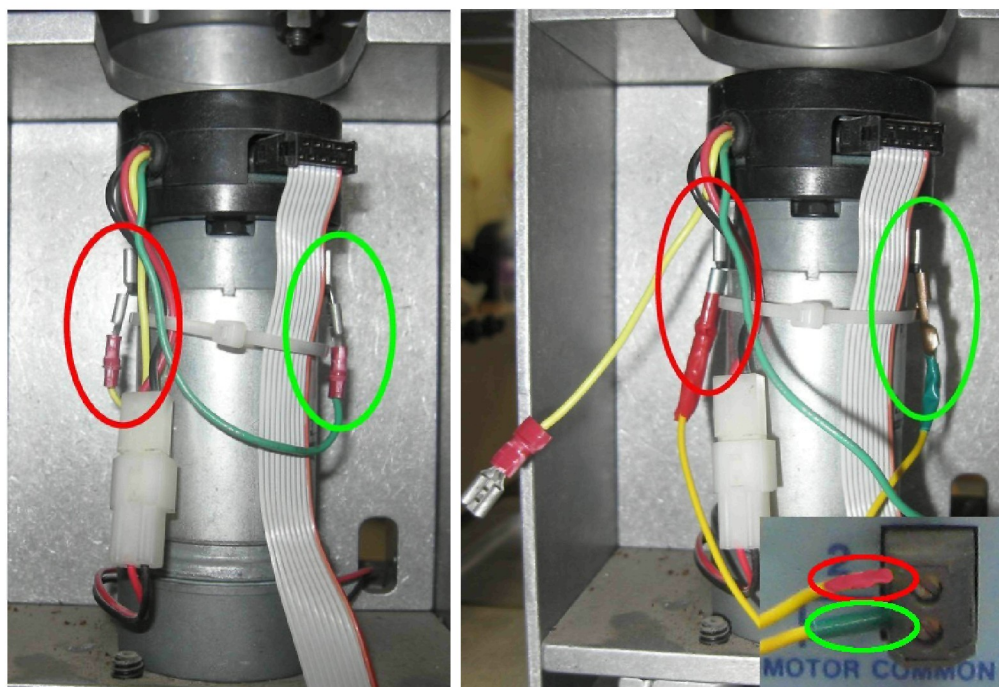
5.5 Mikrospínač jako detektor objektu

V této úloze budeme detekovat objekty na rotačním karuselu pomocí mikrospínače. Úloha bude rozšířena o ovládání pomocí přepínače, který je součástí I/O modulu řídicího systému Mark IV. Vhodně využijeme i funkci cyklu celého programu ve vytvořené aplikaci RhinoTalk.

Všechny náležitosti týkající se nových funkcí a příslušenství si opět popíšeme před samotným zahájením programování robota.

5.5.1 Připojení rotačního karuselu

V této úloze bude vhodnější řídit rotaci karuselu pouze změnou napětí, tedy je možný neustálý pohyb pouze ponecháním určité hodnoty napájecího napětí z portu AUX-2. Bude se tedy jednat o již zmiňovaný režim „rychlosti“. Od motoru karuselu odpojíme stávající napájení, které je vyvedeno z enkodéru a umožňuje docílení přesné polohy (režim „polohy“) jak bylo ukázáno v úloze ovládání vstupů a výstupů v kap. 5.3. Na vývody napájení nasuneme vodiče s připojením k portu AUX-2. Oba vodiče jsou vybaveny konektory pro snadnou změnu typu napájení.

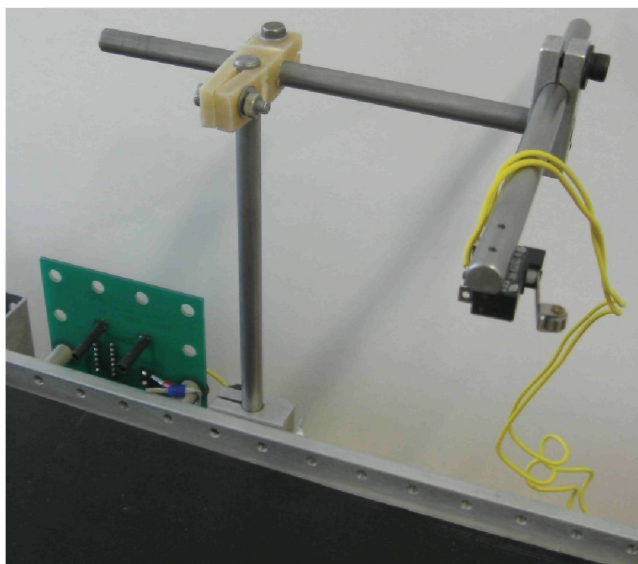


Obrázek 34: Zapojení motoru karuselu v režimu „rychlosti“

Na levé části obrázku 34 je původní zapojení motoru v režimu „polohy“. V druhé polovině obrázku je změněno napájení na režim „rychlosti“, vodiče jsou připojeny k výstupu AUX-2.

5.5.2 Zapojení a upevnění mikrospínače

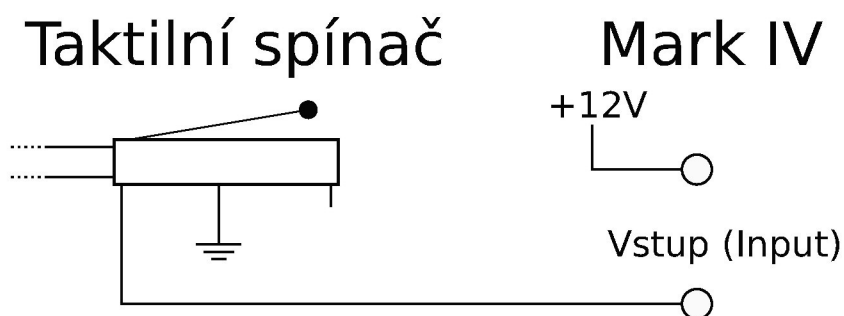
Mikrospínač a sadu připevňovacích kovových tyček připevníme k pásovému dopravníku. Způsob upevnění je patrný z obrázku.



Obrázek 35: Upevnění mikrospínače k dopravníku

Takto připevněnou bránu s mikrospínačem využijeme pro detekci objektu na rotujícím karuselu.

Připojení mikrospínače na vstupní port řídicího systému je velice snadné. Prostřední svorku mikrospínače připojíme na zemnicí vodič a levou svorku na vstup řídicího systému, jehož druhou část připojíme na napájení +12 V. Při sepnutí mikrospínače tak dojde k vyhodnocení vstupu jako „High“, tedy signalizuje přítomnost objektu. Na obrázku 36 je vidět schéma zapojení.



Obrázek 36: Propojení mikrospínače s řídicím systémem Mark IV

5.5.3 Využití přepínačů

Poslední novou funkcí využitou v této úloze je řízení běhu programu pomocí přepínačů. Testování změny stavu těchto přepínačů se téměř neliší od testování vstupů. Pouze v příkazu pro testování jsou adresovány hodnotou 9–16.

WI,9,1 - čeká na přepnutí 1. přepínače do hodnoty „Zapnuto“

- Vypnuto – páčka v poloze vlevo
- Zapnuto – páčka v poloze vpravo



Obrázek 37: Přepínač na řídicím systému Mark IV

Na čelním panelu jsou polohy přepínačů označeny symboly 0 (Vypnuto) a 1 (Zapnuto).

To je vše k seznámení s zařízením využitým v této výukové úloze a opět můžeme přejít k tvorbě vlastního programu pro řízení robota a periferních zařízení. V této úloze vystačíme pouze s řídicím systémem Mark IV a robotem XR-4 s ovládáním pomocí vytvořené aplikace RhinoTalk.

Na rotační karusel umístíme dva objekty (lego kolečka s tyčkou). Ty umístíme tak, aby při rotaci karuselu došlo k jejich kontaktu s připraveným mikropřepínačem.

Prvními instrukcemi v našem programu bude spuštění rotace karuselu a následně testování sepnutí spínače při jeho kolizi s objektem. Ihned po detekci objektu karusel zastavíme.

XS,2,25

WI,3,1

XS,2,0

Rychlost rotace volíme záměrně velmi nízkou. Testování probíhá v cyklu a dochází k určité časové prodlevě při vykonávání instrukcí. Pokud by byla zvolena větší rychlost došlo by vlivem prodlevy před zastavením karuselu například k posunutí celé brány nebo nepožadovanému odsunutí detekovaného objektu.

Objekt uchytíme pomocí chapadla robota a přemístíme do požadované polohy na dosah obsluhy robota. V této poloze ve které je možné objekt manuálně odejmout budeme testovat přepnutí přepínače č. 1.

WI,9,1

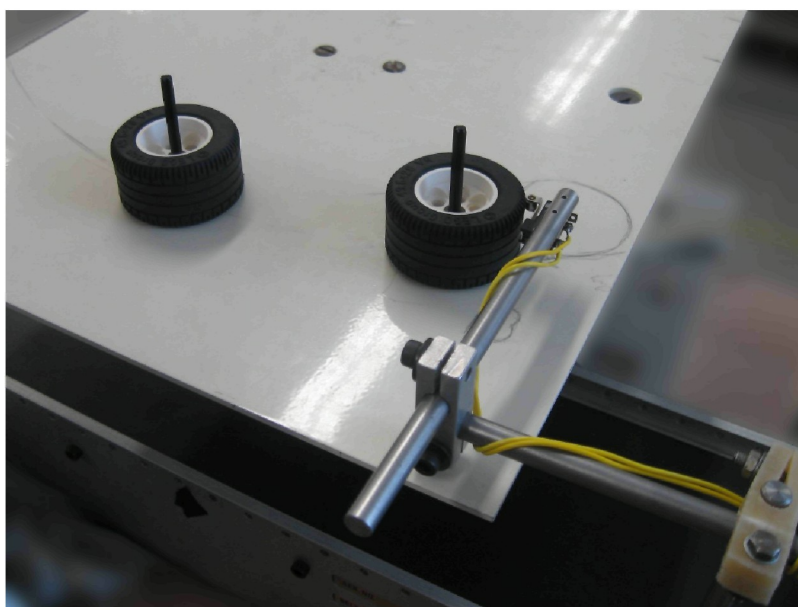
Po přepnutí přepínače do polohy *zapnuto* upustíme předmět otevřením čelisti.

GO

Po krátké prodlevě dojde k upuštění předmětu, například do připravené dlaně obsluhy robota. Následně přepneme přepínač do polohy *vypnuto* a robota vrátíme do počáteční pozice.

WI,9,0

HA



Obrázek 38: Detekce objektů na rotačním karuselu

Celá úloha je připravena tak, aby jí bylo možné spustit v cyklu. Proto pokud před spuštěním programu zaškrtneme políčko možnosti **Opakování** a nastavíme počet na hodnotu 2, dojde k opakování úlohy a detekování dalšího objektu. Samozřejmě můžeme cyklů nastavit i více, dle počtů objektů, nebo je po přenesení vracet zpět na rotační karusel.

6 Závěr

Před vytvářením vlastní aplikace pro řízení robotických systémů RHINO musela být, vzhledem ke stáří těchto zařízení, otestována jejich funkčnost. Ta bohužel není úplně ideální a v průběhu realizace muselo být řešeno několik technických problémů.

Mezi problémy řídicího systému Mark IV patří nefunkčnost jednoho z 8 portů pro připojení motorů. Naštěstí se nejedná o jeden ze základních portů motorů A–E, které jsou využity pro řízení motorů robota. Jde o port G, který spolu s portem H slouží k připojení volitelných externích zařízení se standardním motorem robotických systémů RHINO. Další problém, který byl v práci rozepsán se týká nedostatků při komunikaci s aplikací *RoboTalk pro Windows*.

Systém Mark III bohužel trpí větším technickým problémem. Při komunikaci s počítačem dochází k nečekanému zaseknutí řídicího systému a to v různých časových intervalech. Intervaly jsou velice malé a jedná se v řádu o jednotky provedených příkazů. Problém musí být s největší pravděpodobností hardwarového rázu. Chyby s komunikací se totiž objevují i u všech originálních aplikací dodávaných k systému.

Na základě všech těchto poznatků a domluvy s vedoucím a konzultantem diplomové práce byla vytvářená aplikace zaměřena pouze na řídicí systém Mark IV a robota XR-4. Ovládání řídicího systému Mark III je nahrazeno ovládáním pomocí teachpendantu, při kterém systém funguje bez jakýchkoliv problémů. Nedošlo tedy k omezení výukových úloh o spolupráci obou robotů.

Pro realizaci vlastní aplikace RhinoTalk bylo zvoleno vývojové prostředí C++ Builder a tedy programovací jazyk C++. Všechny důležité části realizace a výsledné schopnosti, které aplikace poskytuje pro ovládání a programování robota, jsou v diplomové práci podrobně rozebrány. Při realizaci aplikace bylo také rozhodnuto o vlastní implementaci výpočtu kinematiky pomocí přímé úlohy kinematiky a Denavit-Hartenbergovi metody. Důvodem je, že systém Mark IV sice disponuje integrovanou schopností pohybu robota v kartézském souřadném systému, ale při různém natočení jednotlivých os robota není schopen tuto polohu vrátit uživateli. Díky vlastnímu výpočtu je tak možné zjistit polohou koncového bodu nástroje robota v jakémkoliv okamžiku.

Po vytvoření aplikace RhinoTalk byla navržena a realizována sada výukových úloh. Každá úloha má specifické zaměření a v souhrnu se úlohy snaží vysvětlit a seznámit uživatele se všemi možnostmi a funkcemi robotických systémů RHINO.

- První úloha je zaměřená na základní možnosti ovládání a programování pomocí vytvořeného softwaru RhinoTalk.

- Druhá úloha seznamuje uživatele s ovládáním řídicího systému Mark IV a robota XR-4 pomocí standardního ovládacího panelu.
- Úloha číslo 3 rozebírá možnosti řízení programu pomocí vstupních a výstupních portů a také ovládání řídicího systému Mark III a robota SCARA.
- Čtvrtá výuková úloha využívá pásového dopravníku a optického snímače k detekci přepravovaného objektu.
- Poslední, pátá úloha, představuje použití mikropínače k detekci pohybujícího se objektu, využití přepínače na čelním panelu řídicího systému Mark IV k ovládání běhu programu a také funkci opakování programu ve vytvořené aplikaci RhinoTalk.

Všechny postupy tvorby úloh jsou podrobně rozepsány v kapitole 5. Každý postup vždy obsahuje nutné informace k zapojení všech využitých periférií a jejich ovládání.

Všechny úlohy jsou vydány i samostatně za účelem lepší dostupnosti pro libovolného uživatele robotických systémů RHINO v Laboratoři inteligentních robotů. Úlohy jsou dále doplněny o stručný manuál k obsluze, který se snaží v krátkém časovém horizontu seznámit uživatele s možnostmi ovládání robotů RHINO. Součástí tohoto manuálu je nejen ovládání pomocí aplikace RhinoTalk, ale i pomocí ovládacích panelů řídicích systémů. Uživatel je tak schopen během krátké doby proniknout do kompletní správy celého mini-robotického centra a ušetří si spoustu času oproti pročítání velkého množství oficiálních dokumentací. To je velice výhodné především vzhledem k omezeným časovým možnostem při absolvování jednotlivých výukových kurzů robotiky.

Při programování v aplikaci RhinoTalk jsou využity základní instrukce řídicího systému Mark IV. Programování pomocí těchto instrukcí je na velmi nízké úrovni. Například při pohybu na určitou pozici je nutné provést jednotlivé instrukce pro nastavení všech motorů, rychlosti, akcelerace apod. a až poté zadat instrukci pro samotné vykonání pohybu. Studentovi je tak umožněno lépe poznat všechny aspekty nutné k pohybu robota.

Celé mini-robotické centrum může vhodně rozšířit možnosti výuky v rámci laboratoře. Pro studenty je vždy výhodnější pracovat na více pracovištích v menších skupinách než hromadně obsluhovat jedno zařízení. Celá skupina studentů se tedy

může vhodně rozdělit například mezi práci s průmyslovými roboty firmy ABB, technickou stavebnicí LEGO a nově vzniklé pracoviště s robotickými systémy RHINO.

Po seznámení se základními možnostmi ovládání pomocí manuálu a výukových úloh mohou studenti projevit svou kreativitu při dalším zdokonalování těchto úloh nebo vytváření úloh vlastních.

Samozřejmě by existovala spousta možností jak celou aplikaci RhinoTalk vylepšit. Například textový editor by bylo vhodné rozšířit o zvýrazňování syntaxe, možnost přidávání komentářů nebo zabudovaný debugger pro odhalování chyb v kódu. Na další vylepšování a zefektivňování bohužel nebyl dostatek času i vzhledem k technickým problémům, které musely být v průběhu práce řešeny.

I přes veškeré technické problémy se povedlo splnit všechny body zadání diplomové práce i s možností využít výsledky k výuce v Laboratoři inteligentních robotů.

Literatura

- [1] RHINO: AN INTRODUCTION TO ROBOTICS – STUDENT WORKBOOK 1. *Úvod do robotiky pro studenty, USA 2000.*
- [2] RHINO: OWNER’S MANUAL – MARK III 8 AXIS CONTROLLER *Uživatelská dokumentace k řídicímu systému Mark III, USA 2000*
- [3] RHINO: OWNER’S MANUAL – MARK IV 8 AXIS CONTROLLER *Uživatelská dokumentace k řídicímu systému Mark IV, USA 1989*
- [4] RHINO: ROBOTALK FOR WINDOWS MANUAL. *Dokumentace k programu RoboTalk pro Windows , USA 2000.*
- [5] RHINO: OWNER’S MANUAL – XR-3, XR-4 AND SCARA. *Uživatelská dokumentace k robotům XR-3, XR-4 a SCARA, USA 1995*
- [6] RHINO: OWNER’S MANUAL – ACESSORIES FOR ROBOTS AND WORK CELLS. *Uživatelská dokumentace vybavení a příslušenství robotů Rhino, USA 1995*
- [7] NOVOTNÝ F. *Separáty připravovaných skript Robotiky, Fakulta strojní, TU Liberec*
- [8] KADLEC VÁCLAV. *Učíme se programovat v Borland C++ Builder a jazyce C++. 1. vydání. Praha, Computer Press, 2002. 80-7226-550-4*
- [9] HOLAN TOMÁŠ, NERUDA ROMAN. *C++ Builder v příkladech. 1. vydání. Praha, BEN, 2002. 80-7300-042-3*

- [10] SIMPLE SERIAL COMMUNICATION WITH BORLAND TURBO C++ EXPLORER [ONLINE]. *Nastavení sériového portu pomocí C++*. [cit. 2010-05-04]. URL:<<http://simpleserial.com/>>

Seznam příloh

1. CD

- Aplikace RhinoTalk
 - Výsledná aplikace
 - Zdrojové kódy
- Výuková úloha č.1
 - Soubory programu
 - Postup realizace
 - Video simulace
- Výuková úloha č.2
 - Postup realizace
 - Video simulace
- Výuková úloha č.3
 - Soubory programu
 - Postup realizace
 - Video simulace
- Výuková úloha č.4
 - Soubory programu
 - Postup realizace
 - Video simulace
- Výuková úloha č.5
 - Soubory programu
 - Postup realizace
 - Video simulace

1. Základní manipulace pomocí softwaru

RhinoTalk

V této úloze budou představeny základní možnosti ovládání a programování pomocí softwaru RhinoTalk pro řídicí systém Mark IV a robota XR-4. Úkolem je přenést tři dřevěné kostky a postavit je na sebe.

Dřevěné kostky rozestavím například na plochu rotačního karuselu s dostatečným rozstupem, aby bylo možné ke každé kostce přistoupit pomocí nástroje robota.



Obrázek 1: Dřevěné kostky

Zkontrolujeme zapojení řídicího systému a robota. Řídicí systém musí být připojen sériovým kabelem s počítačem. Do portů na řídicím systému připojíme odpovídající přívodní kabely všech motorů robota a pojezd do portu H.

Spustíme software RhinoTalk a v menu zvolíme:

Nastavení → Navázat spojení

Při dotazu na vykonání počáteční pozice (Hard Home) odpovíme **Ano** a necháme robota najít tuto polohu, pokud je připojen i pojezd robota, bude nalezena i počáteční pozice osy H s pojezdem. Po nalezení počáteční pozice je robot připraven k pohybu.

Občas dochází k situaci, kdy robot nereaguje na žádné příkazy pohybu. V tomto případě je možné resetovat nastavení pro daný typ robota (XR-4).

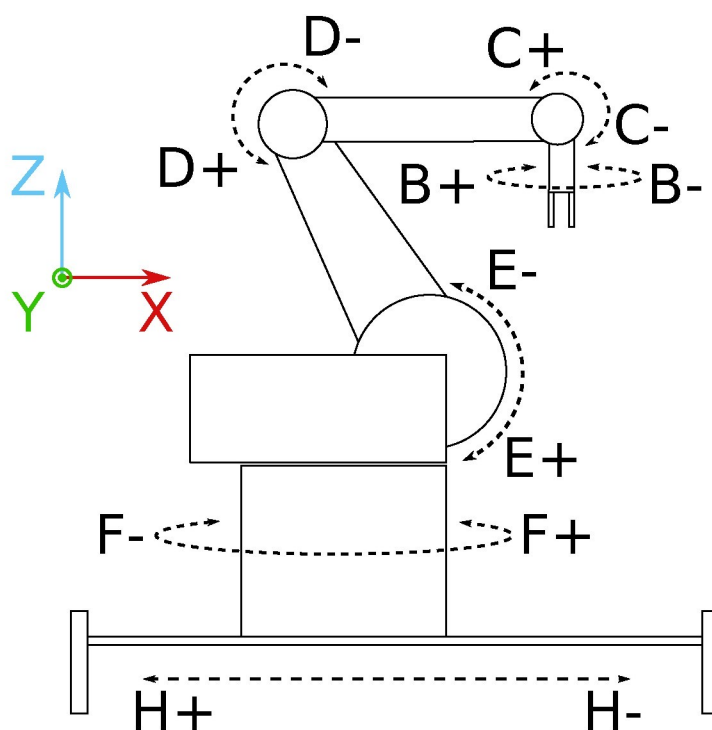
Řídicí systém → Reset nastavení

Po vykonání tohoto příkazu zkusíme zadat znovu nalezení počáteční pozice a vše by mělo být v pořádku.

Robot → Nastavit Hard Home

Pohyb robota můžeme vykonávat pomocí pohybu jednotlivými osami nebo pohybem v kartézských souřadnicích (XYZ). Ovládání pomocí souřadnic XYZ je rychlejší a jednodušší, ale například pokud je zapotřebí dosáhnout nějaké složitější pozice, bude výhodné využít natáčení jednotlivých os.

Přehled směrů otáčení všech os a orientace kartézského souřadného systému si ukážeme na přehledném obrázku 2. U každé osy robota je znázorněna šipka pohybu s popisem s kladným nebo záporným znaménkem. Kladný symbol určuje, jakým směrem se bude osa pohybovat, pokud bude motorem otáčeno v kladném směru, tedy bude docházet k přičítání jednotek enkodéru daného motoru. U záporného znaménka je tomu logicky naopak. V aplikaci RhinoTalk je směr pohybu dané osy také znázorněn kladným nebo záporným znaménkem.



Obrázek 2: Přehled směrů pohybů os robota XR-4

Pro začátek, než si zvykneme na obsluhu robota, bude výhodné nastavit rychlost a akceleraci systému na nižší hodnoty než je přednastavených 50 %. Při neuváženém pohybu tak nedojde k větším škodám na okolním vybavení nebo robotu samotném. V aplikaci RhinoTalk, v modulu **Globální nastavení systému**, nastavíme rychlost a akceleraci v procentech maximálních hodnot a potvrdíme tlačítkem **Odeslat nastavení**.

„Robota“ přemístíme nad první z připravených kostek. Stiskneme tlačítko **Generovat kód** a do editoru programu nám přibude nastavení všech os i s příkazem pro vykonání pohybu. Kód přidáný do programu bude vypadat například:

PD,B,-1558

PD,C,0

PD,D,-650

PD,E,2921

PD,F,-2173

PD,H,0

MC

Otevřeme čelisti pomocí tlačítka **A-Otevřít**. Kód *GO* pro otevření čelistí také přidáme do našeho programu. Najedeme robotem až do místa, kde budou čelisti svírat mezi prsty připravenou kostku. Opět vygenerujeme kód pohybu pomocí tlačítka **Generovat kód** a zavřeme čelisti tlačítkem **A-Zavřít**. Instrukci *GC* pro zavření čelistí přidáme do programu. Uchycenou kostku zvedneme výš nad ostatní tak, abychom nezavadili při dalším pohybu. Přesuneme se nad požadovanou pozici, vygenerujeme kód, a poté se spustíme níž až k ploše podkladu, na který kostku přesouváme. Vygenerujeme další bod pohybu do programu a přidáme instrukci otevření čelistí. Opustíme místo s přesunutou kostku, opět nejlépe nejdříve provedeme pohyb v ose z tak, aby jsme o přesunutou kostku nezavadili při dalším pohybu. Přesun zbylých dvou kostek provedeme téměř analogicky jako přesun první.

Nakonec můžeme program spustit tlačítkem **Spustit** nebo krokovat pomocí **Krok vpřed**. Při krokování se aktuální instrukce zvýrazní označením. Odpojení od řídicího systému při ukončení práce lze provést v hlavním menu.

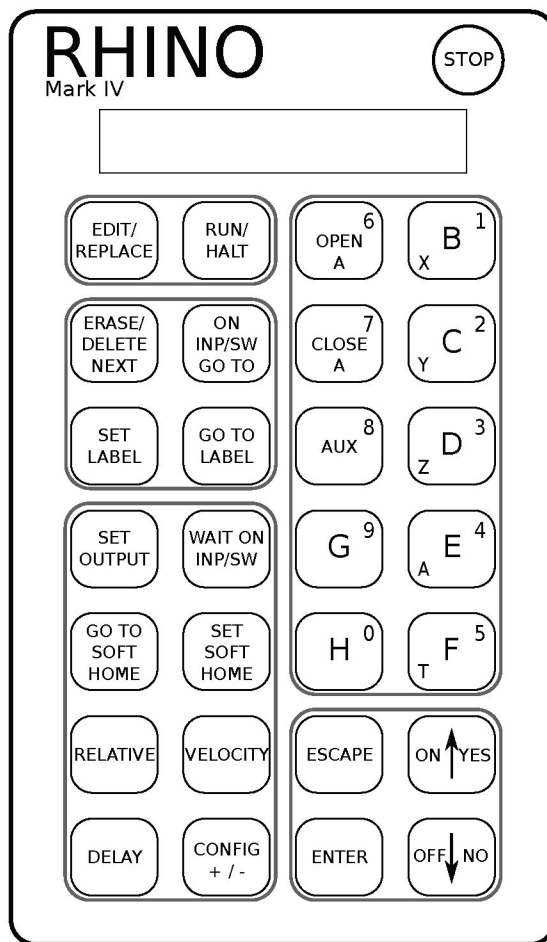
Nastavení → Odpojit

2. Základy ovládání pomocí ovládacího panelu

Jednoduchá úloha pro osvojení ovládání řídicího systému Mark IV a robota XR-4 pomocí standardního ovládacího panelu.

Cílem této úlohy je uchopení plastové tyčky ze stavebnice lego a její následné vsunutí do otvoru v plastovém kolečku taktéž ze stavebnice lego.

Na obrázku č. 1 je zobrazen ovládací panel řídicího systému Mark IV, který bude využit pro splnění této úlohy.



Obrázek 1: Ovládací panel – Mark IV

Rozložení tlačítek není příliš ergonomické a na první pohled nebude asi zřejmé jak panel použít k ovládání robota. I přesto však stačí jen několik málo minut a je možné porozumět většině funkcí, které ovládací panel poskytuje.

Po zapnutí systému tlačítkem na čelním panelu přejde systém automaticky do režimu *Play*, to znamená, že je možné řízení pomocí ovládacího panelu. Aby bylo možné s robotem správně manipulovat a došlo ke zpřístupnění všech možností ovlá-

dání, vykonáme příkaz pro nalezení počáteční pozice robota (Hard Home). Stiskneme tlačítko **CONFIG** +/–, šipkami v pravém dolním rohu najdeme položku **GO TO HARD HOME** a potvrdíme klávesou **ENTER**. Řídící systém spustí rutinu pro nalezení počáteční pozice robota.

Je vhodné zkontrolovat nastavení rychlosti pohybu robota, to provedeme pomocí tlačítka **VELOCITY**. Po stisku se zobrazí aktuální rychlost v rozmezí 0–100 %. Šipkami nastavíme novou hodnotu (50 %) a potvrdíme klávesou **ENTER**.

Nyní se pustíme do vytváření programu. Ten se vytváří v editačním režimu, do kterého se přepneme prvním tlačítkem v levém horním rohu – **EDIT/REPLACE**. Aktivace tohoto režimu je indikovaná slovem *EDIT* v pravém horním rohu displeje ovládacího panelu. Je možné přidávat jednotlivé kroky do programu, aktuální počet kroků programu se zobrazuje v pravém horním rohu displeje.

Přejdeme k prvním pohybům robota. Pro pohyb osy F stiskneme tlačítko **F** a šipkami můžeme pohybovat osou v obou směrech. Nastavím hodnotu –1209. Stejným způsobem přepneme na osu E a nastavíme hodnotu 2476 a osu D zastavíme na hodnotě –600. Po nastavení všech tří pohybů uložíme všechny do jednoho kroku programu stiskem klávesy **ENTER**. Instrukce kroku se uloží do programu a můžeme pokračovat dalšími pohyby. Hodnoty všech následujících kroků jsou přehledně utříděny v tabulce č. 1.

V prvním sloupci jsou jména os a v dalších sloupcích hodnota natočení dané osy v jednotkách enkodéru, které používá ovládací panel. Proškrtnutí hodnoty znamená, že v daném kroku se hodnota natočení osy nemění a zůstane tedy stejná jako v kroku předchozím. Osa A jsou čelisti robota proto není hodnota uvedena v jednotkách, ale slovně udává jestli mají být čelisti otevřeny nebo zavřeny.

Na startu programu počítáme s tím, že robot je uveden do počáteční pozice.

Krokem č. 2 se dostaneme čelistmi robota nad připravenou tyčku a bude nutné jí uchytit. Stiskneme **CLOSE A** a potvrdíme klávesou **ENTER**. Dojde k uzavření čelistí a přidání dalšího kroku s pořadovým číslem 3 do našeho programu.

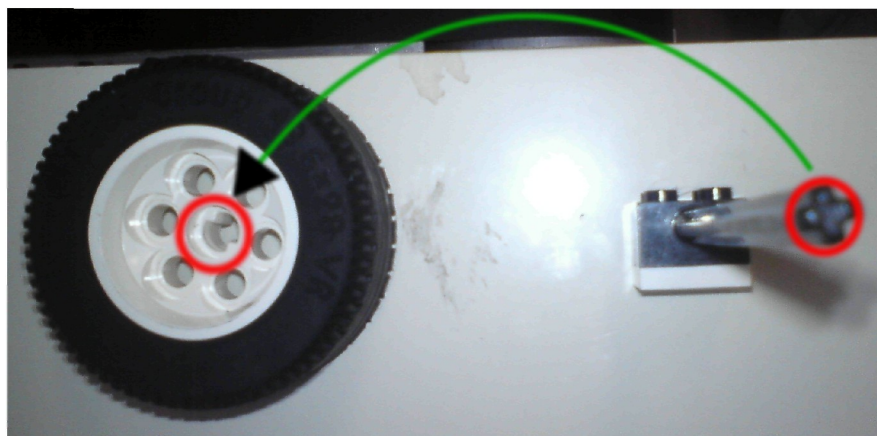
Dvěma dalšími kroky č. 4 a 5 se přesuneme s uchycenou tyčkou až těsně nad otvor v kolečku.

Abychom nyní dokázali vsunout plastovou tyčku do otvoru v kolečku musíme postupovat po malých krocích a střídat posun v ose E a D. Budeme se tedy přepínat mezi oběma osami a vždy vykonáme jen malý pohyb směrem dolů do doby než dosáhneme hodnoty 2374 pro osu E a –909 v ose D. Celý pohyb můžeme pro jistotu rozdělit přibližně v polovině pohybu stisknutím tlačítka **ENTER**.

Po vsunutí tyčky do kolečka otevřeme čelisti pomocí **OPEN A** a vrátíme robota

osa	kroky programu								
	1	2	3	4	5	6	7	8	9
A	-	-	zavřít	-	-	-	-	otevřít	-
B	-	-	-	-	-	-	-	-	-
C	-	-	-	-	-	-	-	-	-
D	600	-808	-	-331	-699	-813	-909	-	0
E	2476	2552	-	2280	-	2326	2374	-	0
F	-1209	-	-	-	1394	-	-	-	0
H	-	-	-	-	-	-	-	-	-

Tabulka 1: Kroky programu 2. úlohy pro Mark IV



Obrázek 2: Plastová tyčka a kolečko

do počáteční pozice klávesou **GO TO SOFT HOME**. V tabulce je tato pozice znázorněna nulováním polohy všech os, se kterými bylo v průběhu vykonávání programu pohybováno.

Editační režim opustíme stiskem **ESCAPE** a tlačítkem **RUN/HALT** spustíme vytvořený program.

Program je možné uchovat i po vypnutí systému uložením do paměti EEPROM. Vstoupíme do nastavení systému stiskem **CONFIG +/-** a zvolíme položku **TRANSFER MENU**, kde je možné uložit nebo nahrát program pomocí **SAVE USER** a **LOAD USER**.

3. Ovládání vstupů a výstupů

Následující úloha rozšíří znalosti z předešlých úloh o použití vstupů a výstupů. Po přenesení objektu robotem XR-4 s řídicím systémem Mark IV bude vyslána informace o provedení této činnosti, na kterou bude reagovat druhý řídicí systém Mark III s robotem SCARA. Jako mezičlánek mezi oběma roboty bude využito rotačního karuselu ovládaného z řídicího systému Mark III.

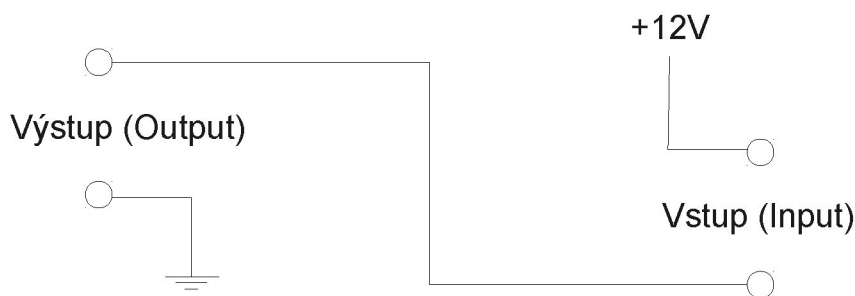
Zapojení vstupů a výstupů

Zapojení vstupů a výstupů se liší dle verze řídicího systému.

Mark III má 8 výstupů, které je možné programově přepínat mezi hodnotami 0 a 5 V a tím předávat signály externímu vybavení. Pro příjem signálu se využívá 8 vstupů, které rozlišují mezi hodnotami 0 a 5 V. Zapojení je velmi jednoduché, každý vstup nebo výstup obsahuje jednu svorku, do které se připojí vodič pro vstupní nebo výstupní informaci.

Mark IV disponuje 8 výstupy a 8 vstupy a každý z nich je složen ze dvou svorek. Vstup rozeznává hodnoty 0–2,5 V jako logické 0 (Low) a hodnoty nad 2,5 V jako logické 1 (High). Hodnota napětí na výstupu závisí na zapojení výstupu. Například je možné použít 12 V z čelního panelu řídicího systému nebo využít pouze propojení se zemí.

Zapojení systému Mark IV pro ovládání vstupu výstupem může vypadat takto:

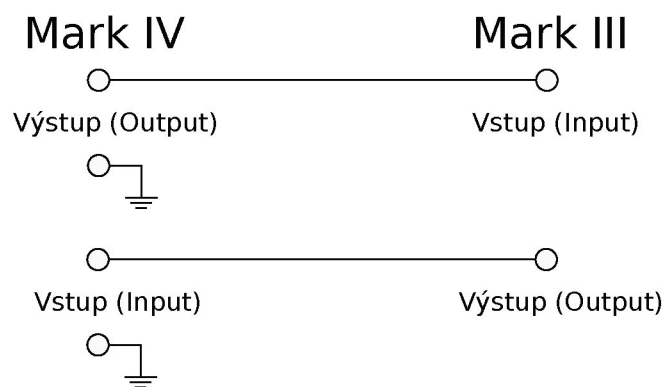


Obrázek 1: Zapojení výstupu na vstup pro Mark IV

Po sepnutí výstupu dojde k uzemnění a následnému sepnutí vstupu jehož druhá svorka je připojena k +12 V.

Pro předávání signálů mezi řídicím systémem Mark III a Mark IV použijeme zapojení dle obrázku č. 2 na další straně.

Rotační karusel připojíme standardním způsobem pomocí motorového konektoru například jako osu G. Jeho řízení bude tedy probíhat v režimu „polohy“.

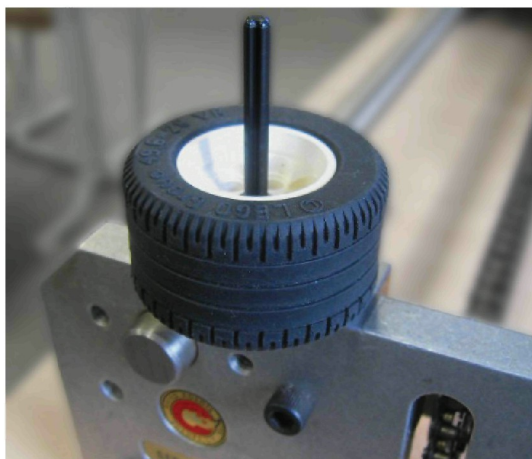


Obrázek 2: Zapojení vstupů a výstupů mezi Mark IV a Mark III

Nyní už je zřejmé jakým způsobem fungují vstupy a výstupy i jak propojit oba řídicí systémy a karusel a můžeme se vrhnout na programování pohybů obou robotů a karuselu.

Mark IV a robot XR-4

Ovládání robota XR-4 pomocí aplikace RhinoTalk bylo již popsáno v první výukové úloze. Proto již využití operace a příkazy nebudou v této úloze popsány do takových detailů. Zaměříme se především na nové funkce a možnosti.



Obrázek 3: Pozice objektu určeného k přesunu

Kolečko s upevněnou tyčkou usadíme na roh základny tak, jak je zobrazeno na obrázku č. 3. Přemístíme se nad kolečko, vygenerujeme kód tohoto umístění, posuneme se až k tyčce z kolečka, vygenerujeme další bod, uzavřeme čelisti, kolečko nadzvedneme a přemístíme na karusel.

Po přesunu kolečka na karusel musíme nastavit výstup pro předání informace řídicímu systému Mark III, který již bude na informaci čekat, aby mohl zahájit vykonávání pohybu. Nastavení výstupu provedeme instrukcí *OB*.

OB,1,1	- zapnutí výstupního portu 1
HA	- přesun robota do počáteční pozice
OB,1,0	- vypnutí výstupního portu 1

Zapnutím výstupu č. 1 dojde k propojení portů se zemí a tím snížení napětí na vstupním portu systému Mark III, který je v běžném stavu v hodnotě 5 V. Dojde tedy k překlopení stavu do hodnoty na kterou systém Mark III bude čekat.

Po předání této informace bude další manipulaci zajišťovat robot SCARA ovládaný pomocí ovládacího panelu systému Mark III.

Mark III a robot SCARA

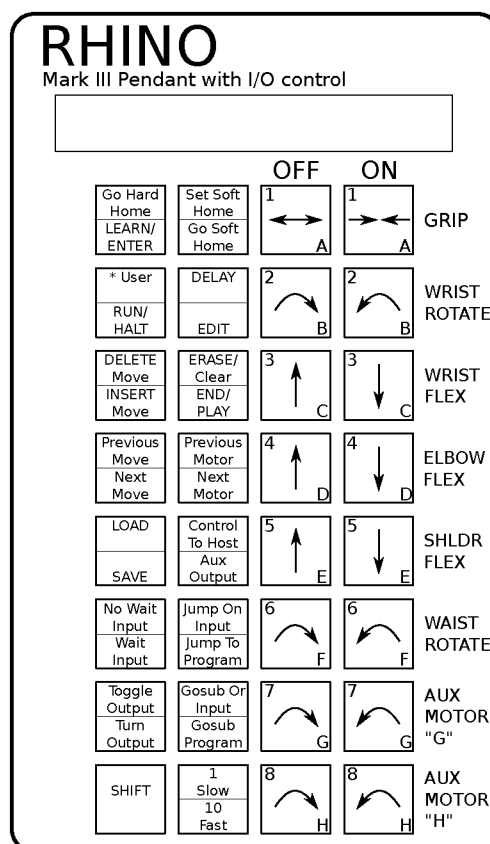
Po spuštění řídicího systému Mark III se automaticky vykoná inicializace všech připojených motorů. Poté je nutné vyhledat počáteční pozici všech os robota (Hard Home). To se provede stisknutím tlačítka **Go Hard Home** na ovládacím panelu. U kláves se dvěma funkcemi se funkce uváděná v horní části tlačítka vykoná pomocí současného stisku klávesy **Shift**. Funkce ve spodní části se vykoná při běžném stisku samotného tlačítka.

Po nalezení počáteční pozice se můžeme přepnout do režimu tvorby programu a to tlačítkem **Learn/Enter**. Ovládání všech os se provádí osmy tlačítky v pravé části ovládacího panelu (A–H). Pro každou osu jsou dvě tlačítka pro kladný a záporný směr pohybu. Celý program bude velice jednoduchý a vystačíme s 6 kroky. Pro uložení jednotlivých kroků se používá tlačítko **Learn/Enter**. Jeden krok se může skládat z více pohybů a příkazů, např. současný pohyb os D a C i s testováním vstupu na určitou hodnotu. Pokud dochází ke kombinování příkazů pohybu a testování vstupů, čeká se nejdříve na dosažení požadovaného vstupu, a až poté se vykoná daný pohyb robota.

Dále si opět, stejně jako u robota XR-4, ukážeme směr pohybu všech os a orientaci kartézského souřadného systému na obrázku č. 5.

To by mělo být dostačující na úvodní seznámení s řídicím systémem Mark III i robotem SCARA a nyní se pustíme do tvorby vlastního programu.

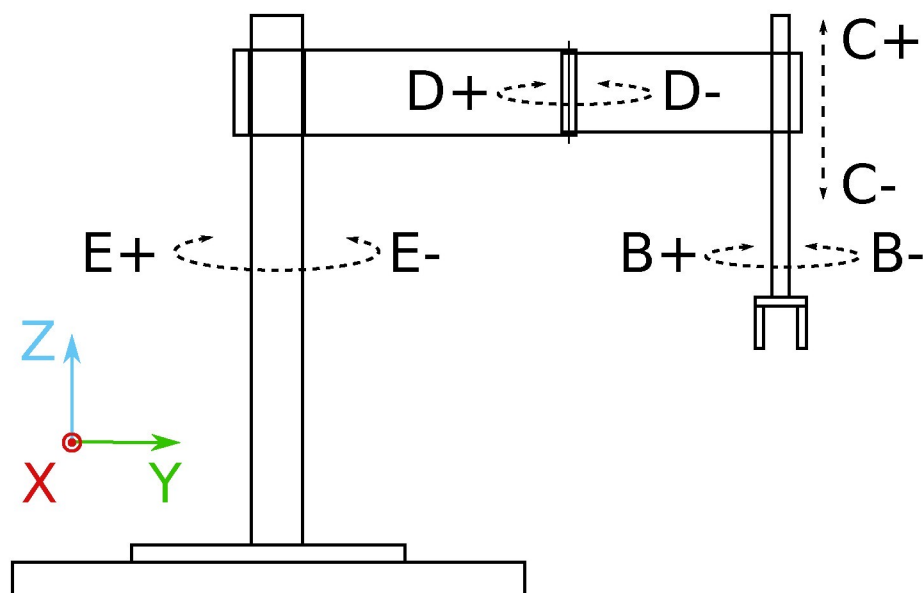
První krok bude obsahovat testování vstupu kvůli zajištění synchronizace pohybu obou robotů. Zatímco robot XR-4 bude přesouvat objekt na rotační karusel,



Obrázek 4: Ovládací panel – Mark III

řídící systém Mark III čeká na dokončení této operace testováním hodnoty na vstupním portu. Po dokončení pohybu odešle řídící systém Mark IV signál nastavením výstupního portu. Příkaz pro testování vstupu zadáme stiskem **Wait On Input** a nastavením příslušných parametrů. Parametry se aktivují tlačítka 1–8 rozdělených do dvou sloupců podle hodnoty, na kterou se má čekat. Levý sloupec s nadpisem **OFF** znamená čekání na hodnotu logické 1 a pravý s nadpisem **ON** na hodnotu logické 0. Hodnota na vstupním portu č. 1 přejde po dokončení operace robota XR-4 do logické 1, proto nastavíme čekání na tuto hodnotu stiskem klávesy 1 z řady **OFF**. Operaci potvrdíme stiskem **Learn/Enter** a tím dojde k uložení prvního kroku programu.

Další kroky se již budou skládat pouze z instrukcí pohybu robota a karuselu, který je připojený k řídícímu systému Mark III. Karusel je připojen ke standardnímu portu motoru a tedy v režimu „pozice“, proto mu budou parametry pohybu zadávány stejným způsobem jako ostatním osám robota. Pohyb jednotlivých os se provádí stiskem příslušného tlačítka v pravé části ovládacího panelu. Rozlišení směru jednoduše výběrem tlačítka z řady **OFF** nebo **ON**. Karusel otočíme přibližně o 180°



Obrázek 5: Přehled směrů pohybů os robota SCARA

na hodnotu 790. Další osy a kroky jsou uvedeny v tabulce:

osa	kroky programu				
	2	3	4	5	6
A	-	-	zavřít	-	-
B	-	-	-	-	-
C	110	-	-	160	-
D	-	154	-	-	0
E	390	427	-	-	0
G	790	-	-	-	0

Tabulka 1: Kroky programu 3. úlohy pro Mark III

Tabulka je opět seříděna dle jednotlivých kroků, u kterých je pro každou osu udána hodnota natočení. A v případě osy A otevření a zavření čelistí.

Po nastavení pozice všech os pro jednotlivé kroky opět stačí stisknout klávesu **Learn/Enter** a dojde k uložení celého kroku do programu. Po uložení všech 6 kroků opustíme režim programování stiskem **End/Play**.

Při vykonávání úlohy nejdříve spustíme program na řídicím systému Mark III pomocí klávesy **Run/Halt**. Systém přejde ke kroku čekání na signál z řídicího systému Mark IV. Poté můžeme spustit i program určený pro systém Mark IV vytvořený pomocí aplikace RhinoTalk.

4. Detekce objektů na pohybujícím se dopravníku

Čtvrtá úloha je zaměřena na využití pásového dopravníku a optického snímače. Na pás dopravníku bude přemístěn objekt pomocí robota XR-4 a dojde ke spuštění dopravníku. Optický snímač připevněný k dopravníku bude předávat informace řídicímu systému Mark IV. Při přerušení paprsku snímače předmětem dojde k zastavení dopravníku a detekovaný objekt bude přemístěn robotem SCARA.

Zapojení dopravníku a optického snímače

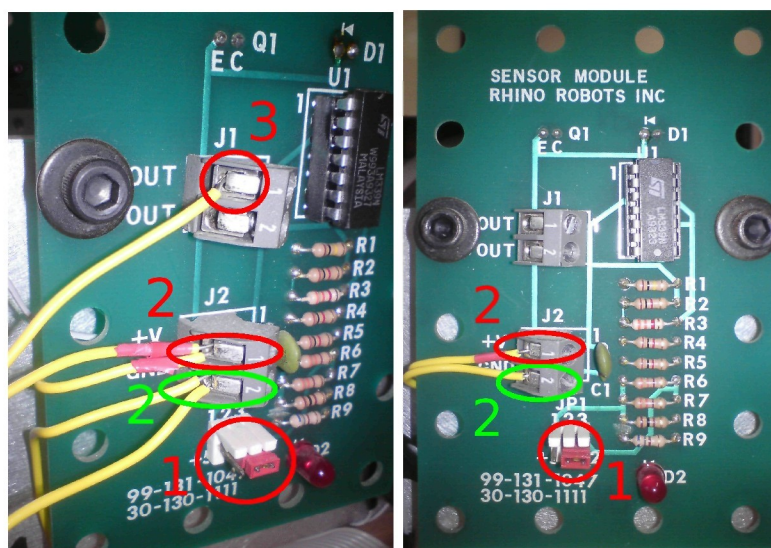
Nejprve je nutné správně připojit pásový dopravník a také optický snímač pro detekci pohybujícího se objektu.

Dopravník připojíme do režimu „rychlosti“ a to pomocí výstupu AUX. Ze svorek portu AUX-1 vyvedeme vodiče a připojíme na svorky motoru dopravníku. Po připojení je možné řídit rychlost pohybu dopravníku změnou napětí na výstupu AUX-1. Změna napětí se provádí příkazem XS. Parametry jsou číslo portu a velikost napětí v procentech maximálního napětí, které je 20 V.

XS,1,40 - AUX-1, 8V (40 % z 20V)

XS,2,-50 - AUX-2, -10V (50 % z -20V)

Optický snímač bude připojen k systému Mark IV, proto svorky pro změnu napájení nastavíme do polohy pro napájení 12 V. Nastavení přepínače je zvýrazněno červeným kroužkem s číslem 1 na obou částech obrázku č. 1.



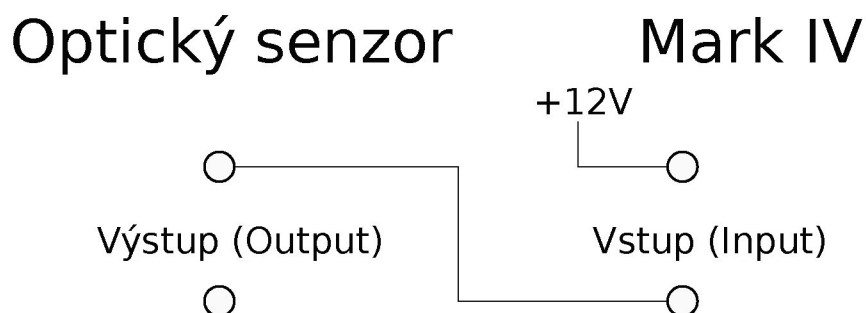
Obrázek 1: Zapojení optického snímače - oba moduly

Na levé části obrázku je první modul snímače do kterého je přivedeno napájení z řídicího systému.

- 2 zelená – uzemnění
- 2 červená – +12 V

Se stejným označením je provedeno i připojení druhého modulu. Napájení můžeme vyvést z první části modulu, takže nemusí být tažen vodič až z řídicího systému.

Číslem 3 je označen vývod výstupu, který připojíme na vstupní port číslo 2 řídicího systému Mark IV. Druhou svorku vstupního portu připojíme na +12 V. Při



Obrázek 2: Propojení výstupu snímače s řídicím systémem Mark IV

detekci objektu dojde k překlopení výstupu optického snímače do logické 0 a tím pádem také k přepnutí vstupu řídicího systému do hodnoty „High“.

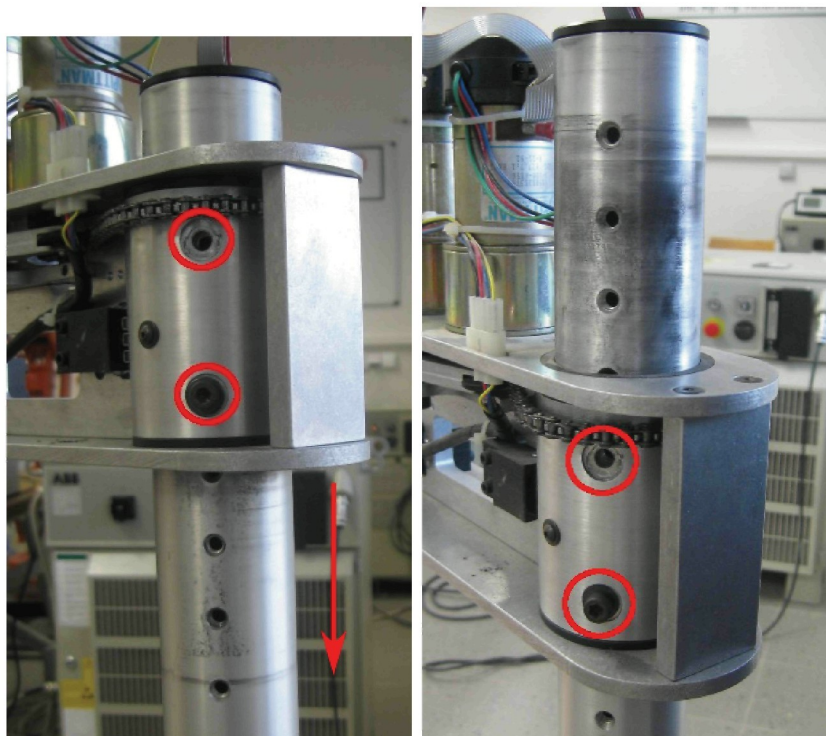
Nastavení robota SCARA

Rameno robota SCARA při pohybu nedisponuje velikým rozsahem v ose z (8 cm), a proto je vybaveno manuálním vertikálním nastavením. Dosud jsme ve všech úlohách vystačili s původním nastavením, ale pás dopravníku je podstatně níž než plocha rotačního karuselu a proto budeme muset rameno přizpůsobit našim požadavkům.

Nastavení ramene je velice jednoduché a zabere jen pár okamžiků. Nejprve je nutné nastavit osu E do takové polohy, aby se odkryly dva imbusové šroubky v zadní části robota. Šroubky povolíme a tím dojde k uvolnění celého ramene. Lehkým přitlačením a popřípadě zakroužením ramenem nastavíme požadovanou polohu a šroubky opět dotáhneme.

Na levé části obrázku č. 3 je zobrazena původní poloha ramene. Vrchní šroub je připevněn do druhého závitu od vrchu.

V pravé části je zobrazeno rameno po přesunutí o 3 polohy níž, vrchní šroub je v pátém závitu.



Obrázek 3: Výškové nastavení robota SCARA

Nastavení všech zařízení v této úloze zabralo více času, ale nyní by vše mělo být připraveno pro zahájení operací s roboty.

MarkIV a robot XR-4

Robotem XR-4 přeneseme objekt (lego kolečko) doprostřed pásu dopravníku. Výchozí polohu předmětu můžeme zvolit například totožně s umístěním v úloze z kapitoly ??, dle obrázku č. ??. Po přenesení kolečka na dopravník otevřeme čelisti a kolečko upustíme. Robota kousek přesuneme, aby nedošlo ke shoení objektu při jeho pohybu po dopravníku.

Přejdeme ke spuštění pohybu dopravníku řízeného výstupem AUX-1. Výstupní napětí zvolíme například 50 % (10 V).

XS,1,50

OB,1,1

Po spuštění dopravníku necháme systém čekat na detekci objektu optickým snímačem, který je připojený ke vstupu č. 2. Přítomnost objektu se projeví hodnotou logické 1 na tomto vstupu.

WI,2,1

Ihned po detekci objektu dopravník zastavíme udáním nulové hodnoty výstupního napětí.

XS,1,0

Informaci o detekovaném objektu předáme řídicímu systému Mark III nastavením výstupu č. 1 do hodnoty logické 1.

OB,1,1

Po předání informace přemístíme robota do počáteční pozice. Výstup vrátíme zpět do výchozí polohy, tedy logické 0.

OB,1,0

Tím jsou veškeré operace pro řídicí systém Mark IV a robota XR-4 vykonány a přemístění detekovaného objektu z dopravníku zajistíme pomocí robota SCARA a řídicího systému Mark III.

MarkIII a robot SCARA

Všechny pohybové operace robota SCARA jsou vypsány po jednotlivých krocích v tabulce č. 1. Programování opět provádíme pomocí ovládacího panelu, které už známe z předchozí úlohy.

osa	kroky programu											
	1	2	3	4	5	6	7	8	9	10	11	12
A	-	-	-	-	Zavřít	-	-	-	-	Otevřít	-	0
B	40	-	195	-	-	-	201	-	-	-	156	0
C	100	-	100	-57	-	82	-	-	-23	-	-	0
D	-393	-	-176	-	-	-	-306	-487	-	-	-	0
E	660	-	780	-	-	-	700	656	-	-	-	0

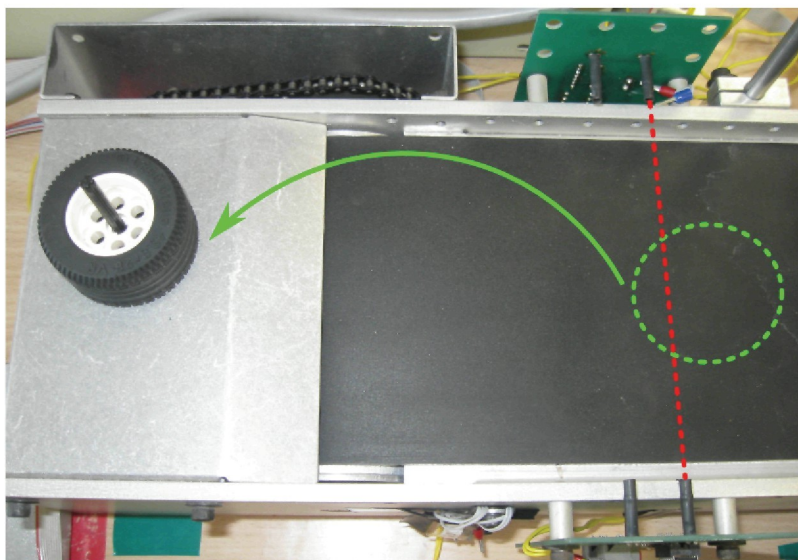
Tabulka 1: Kroky programu 4. úlohy pro Mark III

V prvním kroku z počáteční pozice přejdeme blíž směrem k dopravníku.

Druhý krok neobsahuje žádný pohybový příkaz, ale zadáme instrukci pro čekání na určitou hodnotu vstupu č. 1, na který nám přijde informace z druhého řídicího systému. Stiskneme tedy nejprve klávesu **Wait Input** a následně **1 – ON**. Systém

bude čekat na hodnotu vstupu *ON* (Low), která nastane uzemněním po zapnutí výstupu na řídicím systému Mark IV.

Po informaci o detekci objektu pokračujeme ve vykonávání dalších kroků dle tabulky, které zajistí přesunutí dopraveného objektu mimo pás dopravníku. Objekt bude umístěn například na nepohybující se plochu dopravníku.



Obrázek 4: Přemístění detekovaného objektu z dopravníku

Při spouštění úlohy opět nejdříve spustíme řídicí systém Mark III, který bude čekat na signál od systému Mark IV.

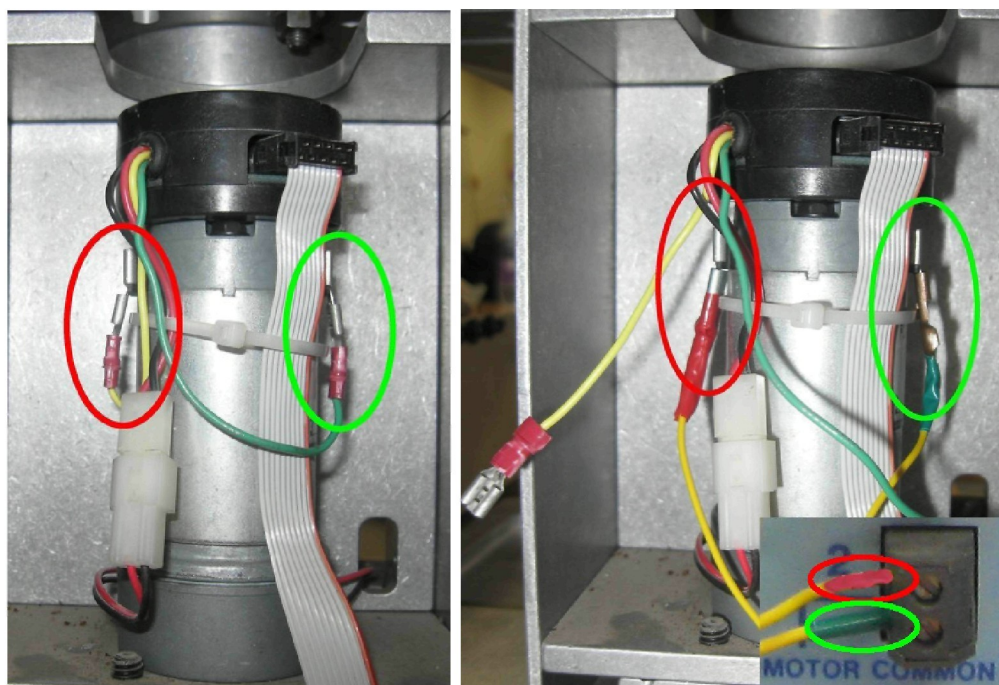
5. Mikrospínač jako detektor objektu

V této úloze budeme detekovat objekty na rotačním karuselu pomocí mikrospínače. Úloha bude rozšířena o ovládání pomocí přepínače, který je součástí I/O modulu řídicího systému Mark IV. Vhodně využijeme i funkci cyklu celého programu ve vytvořené aplikaci RhinoTalk.

Všechny náležitosti týkající se nových funkcí a příslušenství si opět popíšeme před samotným zahájením programování robota.

Připojení rotačního karuselu

V této úloze bude vhodnější řídit rotaci karuselu pouze změnou napětí, tedy je možný neustálý pohyb pouze ponecháním určité hodnoty napájecího napětí z portu AUX-2. Bude se tedy jednat o již zmiňovaný režim „rychlosti“. Od motoru karuselu odpojíme stávající napájení, které je vyvedeno z enkodéru. Na vývody napájení nasuneme vodiče s připojením k portu AUX-2. Oba vodiče jsou vybaveny konektory pro snadnou změnu typu napájení.

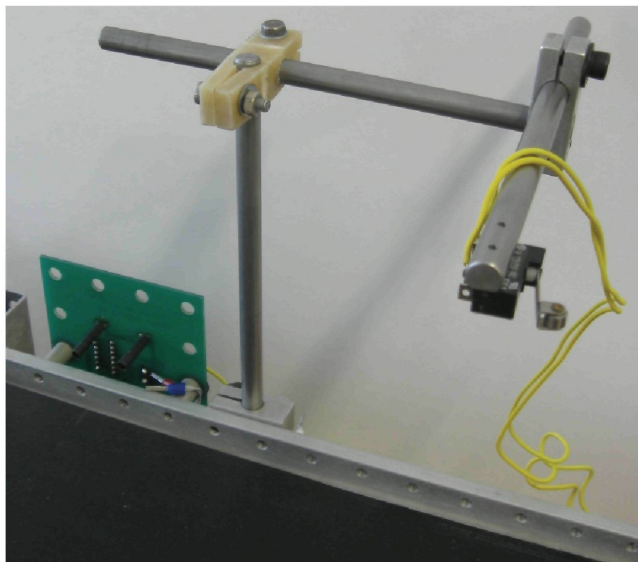


Obrázek 1: Zapojení motoru karuselu v režimu „rychlosti“

Na levé části obrázku 1 je původní zapojení motoru v režimu „polohy“. V druhé polovině obrázku je změněno napájení na režim „rychlosti“, vodiče jsou připojeny k výstupu AUX-2.

Zapojení a upevnění mikrospínače

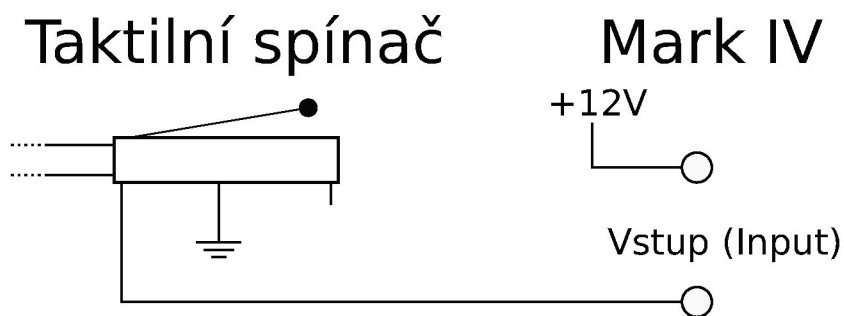
Mikrospínač a sadu připevňovacích kovových tyček připevníme k pásovému dopravníku. Způsob upevnění je patrný z obrázku.



Obrázek 2: Upevnění mikrospínače k dopravníku

Takto připevněnou bránu s mikrospínačem využijeme pro detekci objektu na rotujícím karuselu.

Připojení mikrospínače na vstupní port řídicího systému je velice snadné. Prostřední svorku mikrospínače připojíme na zemnicí vodič a levou svorku na vstup řídicího systému, jehož druhou část připojíme na napájení +12 V. Při sepnutí mikrospínače tak dojde k vyhodnocení vstupu jako „High“, tedy signalizuje přítomnost objektu. Na obrázku 3 je vidět schéma zapojení.



Obrázek 3: Propojení mikrospínače s řídicím systémem Mark IV

Využití přepínačů

Poslední novou funkcí využitou v této úloze je řízení běhu programu pomocí přepínačů. Testování změny stavu těchto přepínačů se téměř neliší od testování vstupů. Pouze v příkazu pro testování jsou adresovány hodnotou 9–16.

WI,9,1 - čeká na přepnutí 1. přepínače do hodnoty „Zapnuto“

- Vypnuto – páčka v poloze vlevo
- Zapnuto – páčka v poloze vpravo



Obrázek 4: Přepínač na řídicím systému Mark IV

Na čelním panelu jsou polohy přepínačů označeny symboly 0 (Vypnuto) a 1 (Zapnuto).

To je vše k seznámení s zařízením využitým v této výukové úloze a opět můžeme přejít k tvorbě vlastního programu pro řízení robota a periferních zařízení. V této úloze vystačíme pouze s řídicím systémem Mark IV a robotem XR-4 s ovládáním pomocí vytvořené aplikace RhinoTalk.

Na rotační karusel umístíme dva objekty (lego kolečka s tyčkou). Ty umístíme tak, aby při rotaci karuselu došlo k jejich kontaktu s připraveným mikropřepínačem.

Prvními instrukcemi v našem programu bude spuštění rotace karuselu a následně testování sepnutí spínače při jeho kolizi s objektem. Ihned po detekci objektu karusel zastavíme.

XS,2,25

WI,3,1

XS,2,0

Rychlost rotace volíme záměrně velmi nízkou. Testování probíhá v cyklu a dochází k určité časové prodlevě při vykonávání instrukcí. Pokud by byla zvolena větší rychlost došlo by vlivem prodlevy před zastavením karuselu například k posunutí celé brány nebo nepožadovanému odsunutí detekovaného objektu.

Objekt uchytíme pomocí chapadla robota a přemístíme do požadované polohy na dosah obsluhy robota. V této poloze ve které je možné objekt manuálně odejmout budeme testovat přepnutí přepínače č. 1.

WI,9,1

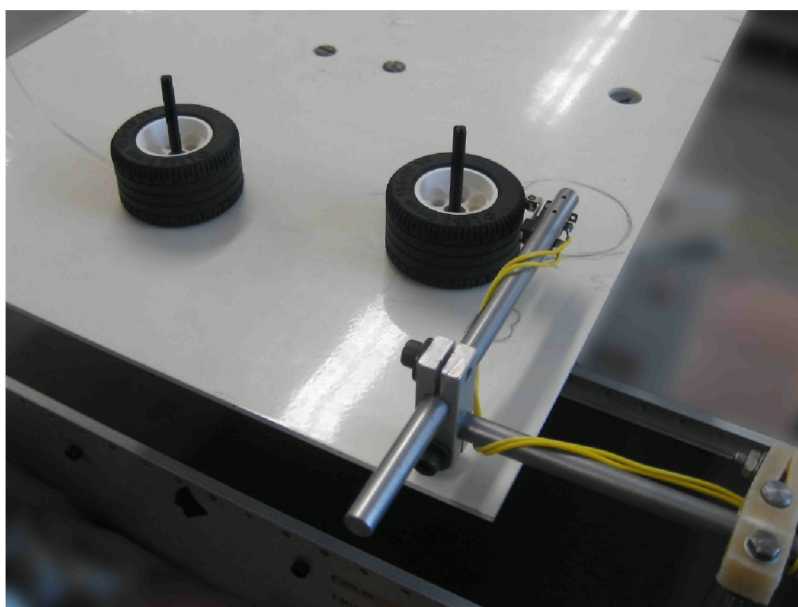
Po přepnutí přepínače do polohy *zapnuto* upustíme předmět otevřením čelisti.

GO

Po krátké prodlevě dojde k upuštění předmětu, například do připravené dlaně obsluhy robota. Následně přepneme přepínač do polohy *vypnuto* a robota vrátíme do počáteční pozice.

WI,9,0

HA



Obrázek 5: Detekce objektů na rotačním karuselu

Celá úloha je připravena tak, aby jí bylo možné spustit v cyklu. Proto pokud před spuštěním programu zaškrtneme políčko možnosti **Opakování** a nastavíme počet na hodnotu 2, dojde k opakování úlohy a detekování dalšího objektu. Samozřejmě můžeme cyklů nastavit i více, dle počtů objektů, nebo je po přenesení vracet zpět na rotační karusel.