

VŠB – Technická univerzita Ostrava

Fakulta strojní

Katedra robotiky

Kolaborativní robotika

Collaborative Robotics

Student:

Tomáš Hrachovec

Vedoucí diplomové práce:

prof. Dr. Ing. Petr Novák

Ostrava 2017

Zadání diplomové práce

Student: **Bc. Tomáš Hrachovec**

Studijní program: N2301 Strojní inženýrství

Studijní obor: 2301T013 Robotika

Téma: **► Kolaborativní robotika**
Collaborative Robotics

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Proveďte rešerši stavu kolaborativních robotů.
2. Popište bezpečnostní požadavky, normy pro oblast kolaborativních robotů.
3. Podle zadání zprovozněte výukové a demonstrační pracoviště s kolaborativním robotem.
4. Navrhněte a vytvořte sadu výukových úloh.

Seznam doporučené odborné literatury:

Vysocký, A., Novák, P. Human-Robot collaboration in industry. *MM Science Journal*. 2016, June 2016, s. 903-906, DOI : 10. 17973/MMSJ. 2016_06_201611 (Scopus), WoS, ISSN 1805-0476.

Matthias B., Industrial Safety Requirements for Collaborative Robots and Applications. *ERF 2014 – Workshop: Workspace Safety in Industrial Robotics: trends, integration and standards [online]*. Available from <https://eu-robotics.net/cms/upload/euRobotics_Forum/ERF2014_presentations/day_2/Industrial_HRC_-_ERF2014.pdf>

ROBOTIQ, *Collaborative robots e-books* [online] Available from <<http://robotiq.com/resource-center/ebooks/>>

norma ISO/TS 15066:2016 *Robots and robotic devices -- Collaborative robots*, 2016

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **prof. Dr. Ing. Petr Novák**

Datum zadání: 09.12.2016

Datum odevzdání: 15.05.2017



prof. Dr. Ing. Petr Novák
vedoucí katedry



doc. Ing. Ivo Hlavatý, Ph.D.
děkan fakulty



Místopřísežné prohlášení studenta

Prohlašuji, že jsem celou diplomovou práci včetně příloh vypracoval samostatně pod vedením vedoucího diplomové práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě 15.5.17



podpis studenta

Prohlašuji, že

- jsem byl seznámen s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo.
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen „VŠB-TUO“) má právo nevýdělečně ke své vnitřní potřebě diplomovou práci užít (§ 35 odst. 3).
- souhlasím s tím, že diplomová práce bude v elektronické podobě• uložena v Ústřední knihovně VŠB-TUO k nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že údaje o kvalifikační práci budou zveřejněny v informačním systému VŠB-TUO.
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona.
- bylo sjednáno, že užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).
- beru na vědomí, že odevzdáním své práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, bez ohledu na výsledek její obhajoby.

V Ostravě : 15. 5. 17



.....
podpis

Jméno a příjmení autora práce:

Tomáš Hrachovec

Adresa trvalého pobytu autora práce:

Přespolní 318, Horní Datyně 739 32

ANOTACE DIPLOMOVÉ PRÁCE

HRACHOVEC, T. *Kolaborativní robotika*. Ostrava: VŠB – Technická univerzita Ostrava, Fakulta strojní, Katedra robotiky, 2017, 111s., Vedoucí práce: prof. Dr. Ing. Petr Novák.

Diplomová práce se zabývá kolaborativní robotikou. V úvodní části je provedena analýze současného stavu dostupných kolaborativních robotů. Následuje popsání kladených požadavků na různé metody spolupráce robotů, vytvoření výukových úloh a na základě znalostí z nich je vytvořena demonstrační úloha.

ANNOTATION OF MASTER THESIS

HRACHOVEC, T. *Collaborative Robotics*. Ostrava: VŠB – Technical university of Ostrava, Faculty of Mechanical engineering, Department of Robotics, 2017, 111p., Thesis head: prof. Dr. Ing. Petr Novák.

The thesis deals with collaborative robotics. The first part is an analysis of the current state of available collaborative robots. The analysis is followed by a description of requirements of collaboration methods. Creation of teaching tasks and based on these teaching tasks creation of demonstrative example.

Obsah

0	Úvod.....	9
1	Rešerše.....	10
1.1	Kolaborativní robotika.....	10
1.1.1	Využití kolaborativních robotů.....	10
1.1.2	Vhodné úkoly.....	10
1.2	Kolaborativní roboty.....	11
1.2.1	ABB – YUMI.....	11
1.2.2	ABB – ROBERTA.....	11
1.2.3	BOSCH – APAS.....	12
1.2.4	Fanuc CR – 35iA.....	12
1.2.5	KUKA – IIWA.....	13
1.2.6	AuBo i5.....	13
1.2.7	Franka Emika.....	14
1.2.8	Universal Robots.....	15
1.2.9	Rethink – Sawyer.....	16
1.2.10	Rethink – Boxter.....	17
1.2.11	Precise Automation – PF400.....	18
1.2.12	Precise Automation – PP100.....	19
1.2.13	Precise Automation – PAVP6.....	19
1.2.14	Yaskava Motoman – HC10.....	20
1.2.15	Nachi MZ04E.....	21
1.2.16	MRK-SYSTEME GMBH.....	21
2	Kolaborativní provoz.....	22
2.1	Bezpečnostní monitorované zastavení.....	22
2.1.1	Požadavky na robotický systém.....	22
2.2	Ruční vedení.....	23
2.2.1	Požadavky.....	24

2.2.2	Vodící zařízení.....	24
2.2.3	Přechod mezi ručním vedením a dalšími typy operací.....	24
2.2.4	Zhodnocení rizik.....	25
2.3	Monitorování rychlosti a polohy.....	25
2.3.1	Požadavky:.....	25
2.4	Konstantní a proměnná rychlost a oddělovací vzdálenost.....	26
2.4.1	Udržování dostatečné odstupu.....	26
2.5	Omezení síly a příkonu vlastní konstrukce nebo ovládním.....	30
2.5.1	Kontaktní situace:.....	30
2.5.2	Snižování rizik pro možný kontakt mezi robotem a obsluhou.....	31
2.5.3	Pasivní a aktivní opatření pro snížení rizik.....	32
2.5.4	Kontrola maximální síly, výkonu a rychlosti.....	32
3	Výukové úlohy.....	33
3.1	Základní operace.....	33
3.1.1	Kalibrace robotu YuMi.....	33
3.1.2	Spuštění kalibrační metody využívajících hallových sond a počítání otáček.	33
3.1.3	Kalibrace čelistí efektorů.....	38
3.1.4	Vytvoření koordinačního systému nástroje.....	39
3.1.5	Základní programování.....	44
3.2	Využití kamery.....	49
3.3	Synchronizace ramen.....	54
4	Demonstrační úloha.....	58
4.1	Vytvoření cesty.....	58
4.2	Poloha papíru.....	59
5	Závěr.....	61
6	Použité zdroje.....	63
7	Seznam příloh.....	65
8	Přílohy.....	66

0 Úvod

V současné době se kolaborativní roboty velice rychle rozšiřují, ať už za účelem pokrytí nedostatku kvalifikovaného personálu, nebo k usnadnění práce zaměstnancům. V dnešním globálním světě je pro konkurenceschopnost firem zapotřebí, aby rychle reagovaly na požadavky trhu a vyráběly s profitem i různorodé malosériové výrobky. K tomu lze využít kolaborativní roboty, neboť díky jejich snadnému programování a možnosti využívat je bez oplocení, lze tyto roboty rychle přestavět k výrobě odlišných výrobků.

Cílem diplomové práce je vytvoření výukových úloh k programování kolaborativního robotu od firmy ABB YuMi a vytvoření demonstrační úlohy, která bude demonstrovat jeho činnost.

1 Rešerše

1.1 Kolaborativní robotika

Roboty, jak je většinou známe, představují velká a robustní zařízení, která jsou navržena k opakované práci při řešení stále stejného úkolu. Tato zařízení musí být obklopena bezpečnostními ploty jasné barvy, které zajišťují bezpečnost zaměstnanců. Jejich zavedení bylo náročné a vyžadovalo vysokou programátorskou úroveň pracovníků. [1], [2]

Kolaborativní roboty jsou do jisté míry opakem předcházejících průmyslových robotů. Lze je využít pro různorodé úkoly, nepotřebují oplocení, mají až sedm stupňů volnosti, to je více, než je zapotřebí pro jejich univerzálnost, což jim umožňuje lepší konfiguraci ramen a tím mohou být zručnější a flexibilnější. Mohou ještě lépe vypomáhat lidem v pracovních procesech. [1],[2]

Kolaborativní roboty kombinují některé nebo všechny následující vlastnosti: [1]

- Jsou navrženy tak, aby byly bezpečné pro zaměstnance v jejich okolí. Využívají silové senzory, které zamezují zranění lidí při kontaktu nebo senzory, které kontaktu předcházejí.
- Jsou relativně lehké a mohou být přemísťovány z jedné úlohy na jinou podle potřeby.
- Nevyžadují vysokou programátorskou úroveň.
- Jsou obecně jednodušší než tradiční roboty, což je dělá levnějšími a méně náročnými na údržbu.

1.1.1 Využití kolaborativních robotů

Kolaborativní roboty mohou být využity především tam, kde jsou nestrukturované úlohy, u kterých by bylo těžké nebo příliš nákladné použít tradiční průmyslové roboty. Jejich využití je vhodné zejména pro malé nebo střední společnosti při vykonávání malých a krátkodobých zakázek.[3]

1.1.2 Úkoly vhodné pro využití kolaborativních robotů

- Volně strukturované zadání
- Krátkodobé úlohy
- Opakující se úlohy nebo úlohy vyžadující vysokou zručnost
- Balení
- Výroba
- Konečná montáž [3]

1.2 Přehled kolaborativních robotů

1.2.1 ABB – YUMI

YuMi je dvouramenný robot od společnosti ABB určený pro montáž malých součástí. Má 14 stupňů volnosti, (7 pro každé rameno). Může být vybaven kamerou pro určování pozic součástí. Robot je snadno přenosný a upevnitelný na různé pracoviště. K identifikaci kolize robotu se využívá elektrický proud. [4]

Tab. 1 Parametry robotu Yumi

Nosnost	0,5 kg (na rameno)
Dosah	559 mm
Přesnost	0,02 mm
Hmotnost	38 kg
IP krytí	30



Obr. 1 Robot YuMi [4]

1.2.2 ABB – ROBERTA

Další zástupce od společnosti ABB (Gomtec) má 6 stupňů volnosti bez bodu singularity. Efektor je vybavený kamerou a silovými senzory. Vyrábí se ve třech velikostech s nosností 4 kg, 8 kg a 12 kg. [5], [6]

Tab. 2 Parametry robotů Roberta

Parametry	P40	P80	P120
Nosnost	4 kg	8 kg	12 kg
Dosah	600 mm	800 mm	1200 mm
Přesnost	±0,1 mm	±0,1 mm	±0,1 mm
Hmotnost	14,5 kg	19,5 kg	30,5 kg
IP krytí	54	54	54



Obr. 2 Robot Roberta [5]

1.2.3 BOSCH – APAS

Robot má integrované kamery a může být dodán s 2D nebo 3D vidícím systémem. Je vybaven tříprstým chapadlem. Rameno má 6 stupňů volnosti s dosahem 911 mm a nosností 2 kg.[7]



Obr. 3 Robot Bosch - APAS [7]

1.2.4 Fanuc CR – 35iA

Robot vychází z robotu M20iA/35M s dodatečným gumovým opláštěním pro zakrytí ostrých bodů. Na rozdíl od ostatních kolaborativních robotů není snadno přenosný a musí být upevněn na podstavci připevněném k zemi. Má 6 stupňů volnosti s nosností 35 kg a dosahem 1813 mm. [8]



Obr. 4 Robot Fanuc CR - 35iA [8]

1.2.5 KUKA - IIWA

Design je navržený na základě lidské ruky. Má 7 stupňů volnosti s nosností 7 nebo 14 kg. Každý kloub je vybaven senzory kontrolujícími jejich polohu. [9]

Tab. 3 Parametry robotů IIWA

Parametry	LBR IIWA 7	LBR IIWA 14
Nosnost	7 kg	14 kg
Dosah	800 mm	820 mm
Přesnost	±0,1 mm	±0,15 mm
Hmotnost	22 kg	30 kg
IP krytí	54	54



Obr. 5 KUKA IIWA [9]

1.2.6 AuBo i5

Je založený na "open source" softwarové architektuře umožňující integraci existujících algoritmů. Robot má 6 stupňů volnosti a jeho nosnost je 5 kg. Může být vybaven kamerovým systémem. [10]

Tab. 4 Parametry robotu AuBo i5

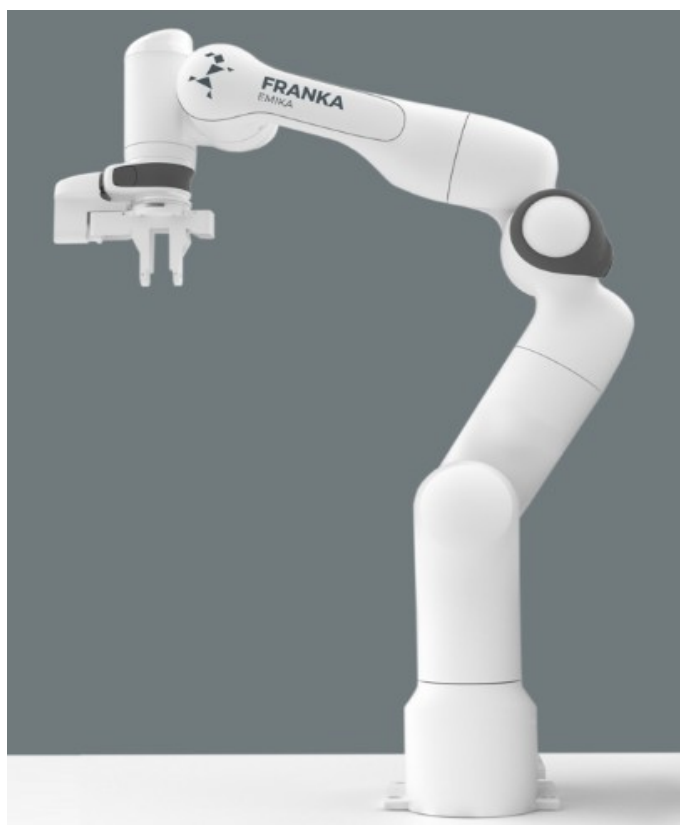
Nosnost	5 kg
Dosah	924,5 mm
Přesnost	±0,1 mm
Hmotnost	24 kg
IP krytí	54



Obr. 6 AuBo i5 [10]

1.2.7 Franka Emika

Kolaborativní robot s dosahem 800 mm. Jeho nosnost je 3 kg a má 7 stupňů volnosti. Je vybaven senzory krouticího momentu ve všech sedmi osách. Hmotnost robotu je 18,5 kg. Programování robotu může být provedeno pomocí předpřipravených aplikací stažitelných z internetu. [11]



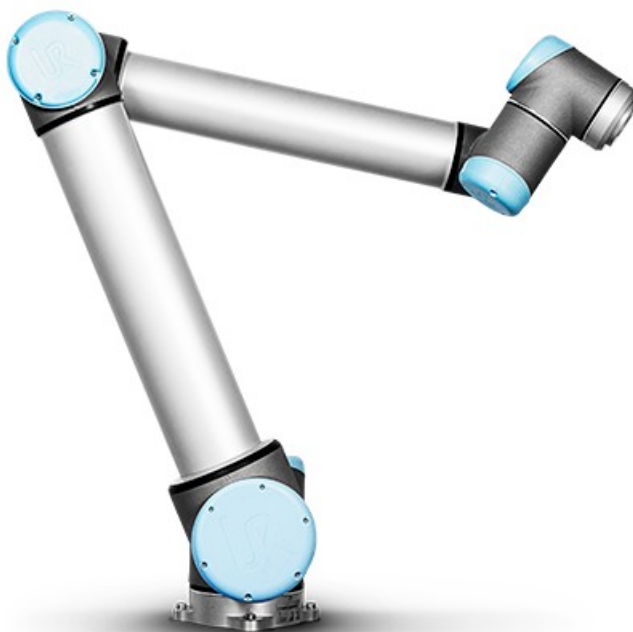
Obr. 7 Robot Franka [11]

1.2.8 Universal Robots

Universal Robots nabízí tři kolaborativní roboty UR3, UR5 a UR10 s nosností 3 kg, 5 kg a 10 kg. Všechny tři varianty mají 6 stupňů volnosti, ale pouze varianta UR3 má nekonečný rozsah na šestém kloubu. [12]

Tab. 5 Parametry robotů UR

Parametry	UR3	UR5	UR10
Nosnost	3 kg	5 kg	10 kg
Dosah	500 mm	850 mm	1300 mm
Přesnost	±0,1 mm	±0,1 mm	±0,1 mm
Hmotnost	11 kg	18,4 kg	28,9 kg
IP krytí	64	54	54



Obr. 8 Robot UR [12]

1.2.9 Rethink – Sawyer

Kolaborativní robot se 7 stupni volnosti. Nosnost ramene je 4 kg. Je vybaven kamerou v zápěstí a širokoúhlou kamerou v hlavě. Může být umístěn na mobilním podstavci pro jeho snadné přemísťování. [13]

Tab. 6 Parametry robotu Sawyer

Nosnost	4 kg
Dosah	1260 mm
Přesnost	$\pm 0,1$ mm
IP krytí	54



Obr. 9 Rethink Sawyer [13]

1.2.10 Rethink – Boxter

Další zástupce od společnosti Rethink. Skládá se ze dvou ramen. Každé rameno má 7 stupňů volnosti a nosnosti 2,2 kg. Ramena mohou pracovat na dvou odlišných operacích ve stejný okamžik. Robot může být taky umístěn na mobilním podstavci. [14]

Tab. 7 Parametry robotu Boxter

Nosnost	2,2 kg (na rameno)
Dosah	1210 mm
Hmotnost	74,8 kg bez podstavce 138,8 kg s podstavcem
IP krytí	50



Obr. 10 Rethink Boxtter [14]

1.2.11 Precise Automation – PF400

První kolaborativní robot typu scara. Je dostupný ve dvou variantách s dosahem 576 mm a 731 mm, které mohou být vyrobeny ještě ve třech variantách s posuvem v ose Z 400 mm, 750 mm nebo 1160 mm. Nosnost všech těchto robotů je 3 kg bez efektoru. [15]



Obr. 11 Robot typu scara [15]

1.2.12 Precise Automation – PP100



Obr. 12 Robot kartézského typu [16]

Jedná se o robot s kartézskou strukturou vyráběný ve dvou rozměrových variantách (500 mm a 685 mm). Nosnost robotu je 2 kg pokud je vybavený efektoem. V opačném případě je jeho nosnost 3 kg. [16]

1.2.13 Precise Automation – PAVP6

Tentokrát se jedná o zástupce angulárního robotu se 6 stupni volnosti. Nosnost robotu je 2,5 kg za předpokladu, že efektor směřuje dolů při maximálním úhlu 45° , při překročení tohoto úhlu nosnost robotu klesá na 2 kg. Veškerá řídicí elektronika je vestavěna v podstavci robotu. [17]



Obr. 13 Robot PAV6 [17]

1.2.14 Yaskawa Motoman - HC10

Robot se 6 stupni volnosti a dosahem 1,2 m. Maximální zatížení je 10 kg. Jedná se zatím o prototyp, který bude certifikovaný podle technické specifikace ISO TS 15066 [18]



.Obr. 14 Kolaborativní robot HC10 [18]

1.2.15 Nachi MZ04E

Jedná se o robot MZ04, ale na rozdíl od něj má snížený výkon. Maximální výkon motoru v robotu je 80 W. Robot využívá takzvanou technologii „Safe Robot“, kde robot může být využit i bez oplocení. [19]



Obr. 15 Kolaborativní robot od Nachi [19]

1.2.16 MRK-SYSTEME GMBH

Firma vyrábí bezpečnostní sadu na robot KR 5 ARC HW od firmy KUKA, která se skládá z pěnového obalu na robot. Obal je vybavený taktilními spínači a senzory přiblížení. Dále vyrábějí systémy ručního vedení na roboty KUKA s nosností v rozsahu 5 až 500 kg. [20]

2 Kolaborativní provoz

Kolaborativním provozem se zabývá norma ISO 10218-2 a dále ji rozšiřuje technická specifikace ISO/TS 15066.

Kolaborativní provoz může využívat jednu, nebo více z následujících metod:

1. Bezpečnostní monitorované zastavení
2. Ruční vedení
3. Monitorování rychlosti a polohy
4. Omezení síly a příkonu vlastní konstrukce nebo ovládním [21],[22]

2.1 Bezpečnostní monitorované zastavení

Tento způsob se využívá k zastavení robotu ve společném pracovním prostoru, před vstupem člověka do pracovního prostoru a spoluprací s robotem. Když se ve společném pracovním prostoru nevyskytuje žádný člověk, robot může vykonávat úkoly jako nekolaborativní robot. Když se robot nachází ve společném pracovním prostředí a je přítomná obsluha, stroj se musí zastavit. Pohyb přístroje bez dodatečného zásahu může začít, až pracovník opustí společný pracovní prostor. [21], [22]

2.1.1 Požadavky na robotický systém

Fungování bezpečnostního monitorovaného zastavení zobrazuje obr. 16. Robotický systém tak může pracovat nekolaborativním způsobem i v kolaborativním prostoru. [22]

Robot motion or stop function		Operator's proximity to collaborative workspace	
		Outside	Inside
Robot's proximity to collaborative workspace	Outside	Continue	Continue
	Inside and moving	Continue	Protective stop
	Inside, at Safety - Rated Monitored Stop	Continue	Continue

Obr. 16 Pohyb robotu nebo jeho zastavení na základě pozice operátora [22]

Kolaborativní prostor musí být vytvořen na základě normy ISO 13855, která udává požadované vzdálenosti. Robotický systém musí být vybaven bezpečnostními vyhodnocovacími zařízeními, které určují přítomnost obsluhy v kolaborativním prostoru. V souladu s vyhodnocováním rizik musí být zamezeno přístupu do nekolaborativního prostoru. Obsluze je umožněno vstoupit do pracovního prostoru pouze v souladu následujícími podmínkami:

- a) Robotický systém nebo další riziko se nevyskytuje v pracovním prostoru.
- b) Robotický systém se nachází v kolaborativním prostoru a je zastaven bezpečnostním monitorovaným zastavením podle normy ISO 10218-1:2011, 5.4, bezpečnostní monitorované zastavení musí zůstat aktivní po celou dobu přítomnosti obsluhy v kolaborativním prostoru. [22]

Jakmile operátor opustí spolupracující prostor s robotem, funkce bezpečného monitorovaného zastavení se může deaktivovat a robotický systém může pokračovat v práci. [22]

2.2 Ruční vedení

U této metody provozu využívá obsluha ručního vedení k vykonání pohybů robotu. Před začátkem ručního vedení robotu se robot zastaví na základě

bezpečnostního monitorovaného zastavení. Úkol je dále vykonáván pomocí ručního řízení nacházejícího se na konci nebo poblíž koncového bodu. [21], [22]

2.2.1 Požadavky na ruční vedení

Robotický systém využívá bezpečnostního monitorovaného zastavení. Rychlostní limit pro bezpečné monitorované zastavení se určí na základě zhodnocení rizik. V případě, že bezpečnost pracovníka je zajištěna omezeným rozsahem pohybu robotu, využívají se měkké osy podle ISO 10218-1. Jestliže dojde k narušení společného pracovního prostoru obsluhou dříve, než je robotický systém připraven k ručnímu vedení, musí dojít k bezpečnostnímu monitorovanému zastavení. Přístup mimo společnou pracovní oblast musí být zabezpečen na základě zhodnocení pracovních rizik. [22]

2.2.2 Vodící zařízení

Robot musí být vybaven vodícím zařízením, které umožňuje nouzové zastavení, a souhlasným povelovým zařízením podle normy ISO 10218-1. [22]

Vodící zařízení musí být umístěno podle následujících požadavků:

- a) musí být umístěno v blízkosti obsluhy tak, aby byla obsluha schopna sledovat přímo pohyb robotu a nemohlo dojít k žádnému nebezpečí při pohybu robotu.
- b) pozice obsluhy musí být taková, aby nevedla k další nebezpečné situaci (např. nesmí stát pod ramenem robotu, nebo pod těžkým břemenem).
- c) postavení obsluhy musí být navrženo tak, aby obsluze nic nepřekáželo ve výhledu na kolaborativní prostor a měla o něm přehled v případě, že by do něj vstoupil někdo další. [22]

Směr pohybu robotu a efektoru musí být ovladatelný ze zařízení pro ruční vedení a musí být intuitivní. [22]

2.2.3 Přejít mezi ručním vedením a dalšími typy operací

Přejít mezi ručně vedenými operacemi a nekolaborativně vykonávanými operacemi nebo dalšími typy kolaborativních operací nesmí představovat dodatečné nebezpečí. Přejít se uskuteční na úmyslný podnět obsluhy. [22]

Je nezbytné dodržet: Specifické hlediska k uvážení v těchto případech jsou:

- a) při přechodu z ručního vedení do bezpečnostního monitorovaného zastavení nesmí dojít k žádnému dodatečnému riziku;
- b) při přechodu z bezpečnostního monitorovaného zastavení do ručního vedení nesmí dojít k žádnému neočekávanému pohybu;
- c) při přechodu z ručního vedení do nekolaborativních operací musí robotický systém vyčkat až veškerá obsluha opustí společný pracovní prostor;
- d) přechod z nekolaborativních operací na ruční vedení nesmí představovat dodatečné nebezpečí. [22]

2.2.4 Zhodnocení rizik

Ke snížení rizik přispívá kombinace přímé kontroly pohybu robotu obsluhou a bezpečnostní omezení rychlosti a pozice. Při hodnocení rizik je třeba zohlednit:

- a) bezpečně vyhodnocovanou rychlost, která umožňuje obsluze kontrolovat robot a závažné nebezpečí pro něj;
- b) čas a vzdálenost, které jsou potřebné pro bezpečnostní zastavení robotu;
- c) nebezpečí způsobené obrobkem, efektozem, perifériemi nebo samotným zařízením. [22]

2.3 Monitorování rychlosti a polohy

V tomto případě se robot a pracovník mohou pohybovat ve společném pracovním prostoru ve stejnou chvíli. Bezpečnost je zajišťována dodržováním minimální bezpečnostní vzdálenosti mezi pracovníkem a robotem po celou dobu. Robotický systém se nikdy nemůže dostat k pracovníkovi blíže než na danou bezpečnou vzdálenost, v opačném případě musí dojít k okamžitému zastavení systému až do doby, kdy se pracovník posune do minimální vzdálenosti od robotu. Minimální vzdálenost je závislá na rychlosti pohybu stroje. [21], [22]

2.3.1 Požadavky na monitorování rychlosti a polohy:

Robot musí být vybaven systémem bezpečného monitorovaného zastavení a bezpečnostního monitorování rychlosti. V případě, že je bezpečnost obsluhy zajišťována omezujícím rozsahem pohybu robotu, robot musí být vybaven bezpečnostními měkkými osami. Podmínky rychlosti a vzdálenosti k robotu musí být splněny vůči všem přítomným lidem ve společném pracovním prostoru. V případě, že

bezpečnostní měření je omezeno počtem lidí ve společném pracovním prostoru s robotem a dojde k překročení maximálního počtu pracovníků na daném místě, robot se z bezpečnostních důvodů musí zastavit. [22]

Jestliže dojde ke zmenšení bezpečnostní vzdálenosti mezi obsluhou a nebezpečnou částí robotického systému, robotický systém musí provést bezpečnostní zastavení a provedení dalších bezpečnostních opatření v souladu s normou ISO 10218-2 (např. vypnutí všech nebezpečných nástrojů). Jakmile dojde k opětovnému dosažení bezpečné vzdálenosti, robot se může opět uvést do pohybu. [22]

Robotický systém se může vyhnout narušení bezpečné vzdálenosti robotu od obsluhy zejména:

- a) snížením rychlosti, s možným celkovým uvedením robotu do bezpečnostního monitorovaného zastavení;
- b) provedením pohybu po další možné trajektorii, u které nedojde k narušení bezpečné vzdálenosti mezi robotem a pracovníkem a nemusí tedy dojít ke snížení rychlosti. [22]

2.3.2 Konstantní a proměnná rychlost a oddělovací vzdálenost

Maximální dovolená rychlost a minimální bezpečnostní oddělovací vzdálenost může být jak konstantní tak i proměnná. U proměnných hodnot maximální dovolené rychlosti a bezpečnostní vzdálenosti mohou být hodnoty neustále upravovány na základě relativní rychlosti a vzdálenosti mezi operátorem a robotickým systémem. U konstantních hodnot (maximální dovolená rychlost a bezpečnostní vzdálenost) musí být hodnoty určeny z vyhodnocení z odhadu bezpečnostních rizik pro nejhorší možný případ. [22]

2.3.3 Udržování dostatečného odstupu

Během automatických operací se nebezpečné části robotického systému nesmí nikdy dostat blíže k obsluze, než je bezpečnostní vzdálenost. Tato vzdálenost může být spočtená na základě výpočtu minimální vzdálenosti z normy ISO 13855, ale je třeba vzít v úvahu přidružené nebezpečí spojené s rychlostí pohybu. [22]

- a) U konstantní rychlosti se využívá nejhorší možný případ pro získání hodnoty pro bezpečnostně monitorované zastavení. Tato hodnota je závislá na aplikaci a je

ověřena vyhodnocením rizik. Limitní hodnota je nastavena pro bezpečnostní monitorovanou rychlost a nesmí být překročena. [22]

b) U proměnných rychlostí jsou k určení bezpečnostní vzdálenosti použity rychlost robotického systému a rychlost obsluhy. Tato hodnota se neustále vyhodnocuje. Alternativním řešením je určování maximální rychlosti, kterou se robot může pohybovat, určené na základě rychlosti pohybu obsluhy a aktuální vzdálenosti mezi robotem a obsluhou. [22]

c) Bezpečnostní oddělovací vzdálenost může být popsána následujícím vzorcem:

$$S_p(t_0) = S_h + S_r + S_s + C + Z_d + Z_r \quad (1)$$

kde

S_t je bezpečnostní oddělovací vzdálenost v čase t_0

t_0 je přítomný čas

S_h je přídavná vzdálenost pro změnu pozice operátora

S_r je přidaná vzdálenost pro reakční dobu robotického systému

S_s je přidaná vzdálenost pro brzdovou vzdálenost robotu

C je předepsaná vzdálenost podle ISO 13855

Z_d je poziční nejistota operátora v kolaborativním prostoru, měřená přítomnostním snímacím zařízením a jeho měřicí tolerance

Z_r je poziční nejistota robotického systému, vyplývající z přesnosti měření pozice robotu

Hodnota $S_p(t_0)$ umožňuje vypočítávat bezpečnou oddělovací vzdálenost dynamicky, umožňuje, aby se hodnota vzdálenosti lišila v závislosti na rychlosti, ale může být použita i pro fixní hodnotu pro nejhorší možný případ. [22]

Vzorec 1 bere v úvahu kombinaci pohybujících se částí robotu a pracovníka ve společném pracovním prostoru. Nejbližší část robotu a obsluha se mohou pohybovat od sebe, zatím co jiná část robotického systému se může přibližovat k obsluze. [22]

Přidaná bezpečnostní vzdálenost pro změnu pozice operátora S_h je vyjádřena vzorcem:

$$S_h = \int_{t_0}^{t_0 + T_r + T_s} v_h(t) dt \quad (2)$$

kde:

T_r je reakční čas robotického systému zahrnující čas vyžadovaný pro detekci pozice operátora, zpracování tohoto signálu, aktivování zastavení robotu, ale samotný čas zastavení se nezapočítává

T_s je čas potřebný k zastavení robotu od aktivování signálu pro zastavení. T_s není konstanta, ale funkce konfigurace robotu, plánovaného pohybu, rychlosti a zatížení

v_h je rychlost obsluhy včetně směru vůči robotu v kolaborativním prostoru. Rychlost může být kladná nebo záporná v závislosti na tom, zda se oddělovací vzdálenost zvětšuje nebo zmenšuje.

t je integrační proměnná

Konstantní hodnota pro S_h využívající odhadovanou rychlost člověka $1,6 \text{ m} \cdot \text{s}^{-1}$ může být odhadovaná na základě vzorce:

$$S_h = 1,6 \times (T_r + T_s) \quad (3)$$

Přidaná bezpečnostní vzdálenost pro reakční čas robotického systému S_r je vyjádřena vzorcem:

$$S_r = \int_{t_0}^{t_0 + T_r} v_r(t) dt \quad (4)$$

Kde v_r je rychlost robotu včetně jeho směru vůči pracovníkovi ve společném pracovním prostoru a může být pozitivní v případě, že se přibližuje k obsluze, nebo záporná, jestliže se od obsluhy oddaluje.

Řídicí systém odpovídající za v_r musí být navržen tak, aby ověřoval způsoby nejvíce ovlivňující zmenšení bezpečné vzdálenosti:

- a) Jestliže rychlost robotu není monitorovaná, uvažuje se, že rychlost robotu je maximální.

- b) Jestliže rychlost robotu je monitorovaná, může být využita, ale musí se počítat s možností akcelerace robotu ve směru nejvíce zmenšující bezpečnou vzdálenost. [22]

Konstantní hodnota pro S_r může být odhadnuta pomocí vzorce:

$$S_r = v \cdot (t_0) \times T_r \quad (5)$$

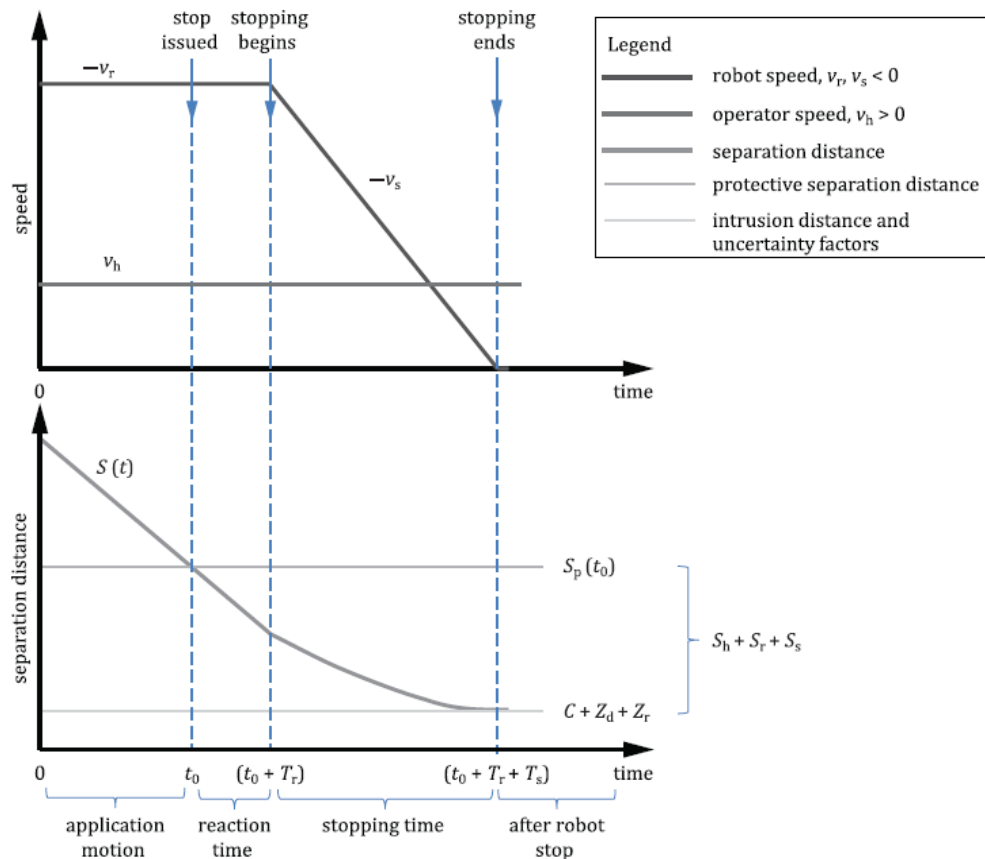
Přidaná bezpečnostní vzdálenost pro brzdou dráhu je daná vzorcem:

$$S_s = \int_{t_0 + T_r}^{t_0 + T_r + T_s} v_s(t) dt \quad (6)$$

Kde v_s je rychlost robotu na dráze jeho zastavení od okamžiku aktivace příkazu k zastavení do úplného zastavení.

Řídicí systém odpovídající za v_s musí být navržen tak, aby ověřoval způsoby nejvíce ovlivňující zmenšení bezpečné vzdálenosti:

- a) Pokud rychlost robotu není monitorovaná, uvažuje se, že pohyb probíhá ve směru nejvíce zmenšující bezpečnou vzdálenost.
- b) Jestliže je rychlost robotu monitorovaná, systém může využít danou rychlost pro potřebnou vzdálenost k zastavení, ale aplikuje se ve směru nejvíce zmenšující bezpečný prostor. [22]



Obr. 17 Graf zastavení robotu na základě jeho rychlosti a vzdálenosti k osluzě [22]

2.4 Omezení síly a příkonu vlastní konstrukce nebo ovládáním

V tomto způsobu spolupráce může dojít ke kontaktu mezi robotickým systémem a obsluhou ať už úmyslně, nebo neúmyslně. Roboty vyžadují speciálně navržené systémy pro spolupráci. K zajištění bezpečnosti se využívají bezpečnostní prostředky samotných robotů a bezpečnostní prostředky jejich řídicích systémů, které udržují hodnotu pod mezní hodnotou získanou při zhodnocování rizik. [21], [22]

2.4.1 Kontaktní situace:

U kolaborativních operací využívajících metody omezení síly a příkonu dochází ke kontaktu mezi operátorem a robotem samotným. Kontaktní situace můžou nastávat z následujících příčin:

- Zamýšlená kontaktní situace, která je součástí samotné aplikace;
- Náhodná kontaktní situace, která může vyplynout z nedodržování pracovního postupu, ale bez technické závady;
- Selhání režimů, které vede ke kontaktní situaci. [22]

Možné typy kontaktu mezi pohybujícím se robotem a částmi lidského těla:

- a) Téměř statický kontakt: situace, při kterých dojde k chycení části těla mezi pohybující se částí robotického systému a nehybné objekty. V takových situacích robotický systém působí silou na chycenou část těla delší dobu, než dojde k jeho uvolnění.
- b) Krátkodobý kontakt: neboli dynamický náraz, představuje situaci, kde do části těla narazí pohybující se část robotického systému a tělo se může odrazit od robotu bez jeho zachycení. Přejímový kontakt závisí na kombinaci setrvačnosti robotu, setrvačnosti části těla a jejich relativní rychlosti. [22]

2.4.2 Snižování rizik pro možný kontakt mezi robotem a obsluhou

Pro snížení rizik z kontaktu robotu s jeho obsluhou a předcházení poranění je třeba:

- a) identifikovat podmínky, ve kterých by takový kontakt nastal;
- b) zhodnotit potenciální rizika pro takový kontakt;
- c) navrhnout robotický systém a společný pracovní prostor, kde by k takovému kontaktu docházelo zřídka a dalo by se mu předcházet;
- d) uplatňovat opatření ke snížení rizik k držení kontaktních situací pod mezní hodnotou. [22]

U odhadu rizik z kontaktních situací se musí brát v úvahu i to, že obsluha není chráněná žádným způsobem pro snížení rizika, včetně osobních ochranných pomůcek, a je třeba sledovat tyto aspekty: [22]

- odhalená část těla
- původ kontaktu, tj. jestli byl kontakt zamýšlený, nebo naopak
- pravděpodobnost, nebo frekvence výskytu této situace
- typ kontaktu (téměř statický, krátkodobý)
- oblasti kontaktu, rychlosti, síly a další veličiny charakterizující fyzický kontakt.

Pro snížení rizika nesmí být v kontaktních oblastech objekty s ostrými a špičatými hranami, jako jsou např. nože, jehly apod.

2.4.3 Pasivní a aktivní opatření pro snížení rizik

Vztahují se jak na krátkodobý kontakt, tak i na téměř statický. Pasivní prvky se vztahují na mechanický návrh robotu, zatímco aktivní se vztahují na řídicí systém robotu. [22]

Pasivní bezpečnostní prvky zahrnují: [22]

a) plochu kontaktní oblasti:

- zakulacené hrany a rohy
- hladké povrchy

b) prvky absorbující energii, prodlužující přenosný čas energie, nebo zmenšující sílu nárazu

- deformační komponenty
- poddajné klouby

Aktivní bezpečnostní prvky zahrnují: [22]

- omezení síly nebo krouticího momentu
- omezení rychlosti pohyblivých částí
- omezení hybnosti, mechanické síly nebo energie jako funkci hmoty a rychlosti
- použití měkkých os a prostorového omezení
- využití funkce bezpečnostního monitorovaného zastavení
- využití senzorů detekujících kontakt.

2.4.4 Kontrola maximální síly, výkonu a rychlosti

Robotický systém musí být navržen tak, aby neumožnil překročení daných limitních hodnot v působení síly na obsluhu a to jak pro dynamické, tak statické kontakty. Maximální povolená rychlost pohybu části robotického systému je závislá na momentu setrvačnosti a minimální ploše robotu, která může přijít do kontaktu s operátorem. [22]

3 Výukové úlohy

3.1 Základní operace

Úloha se zabývá základními operacemi pro přípravu robotu k použití.

3.1.1 Kalibrace robotu YuMi

Kalibrace se musí provést, když dojde k jednomu z následujících případů:

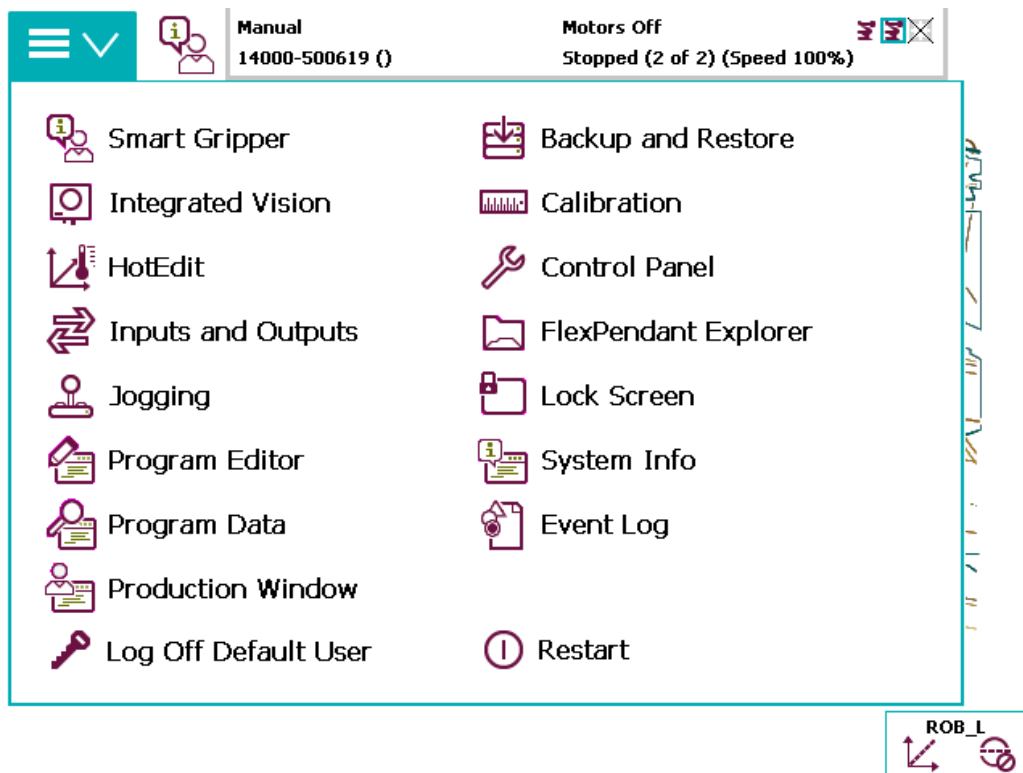
- změní se hodnoty dekodéru;
- ztratí se paměť počítačové otáček;
- dojde k přestavbě robotu.

V situaci, kdy je robot přestavěn, je zapotřebí ho nechat zkalibrovat odborníky pomocí pendula, protože robot je vybavený absolutní přesností. V případě, kdy dojde ke změně hodnot dekodéru, je vhodné, ale není to nezbytně nutné, provést kalibraci na absolutní přesnost. Může se použít kalibrační metoda s využitím Hallových sond.

K provedení kalibrací je zapotřebí nejprve natočit jednotlivé stupně volnosti ramen do kalibračních poloh, které jsou označeny ryskami. Daných poloh je možné dosáhnout natáčením jednotlivých os s využitím FlexPendantu, nebo povolením "ručního průchodu" a nastavením jednotlivých poloh ručně.

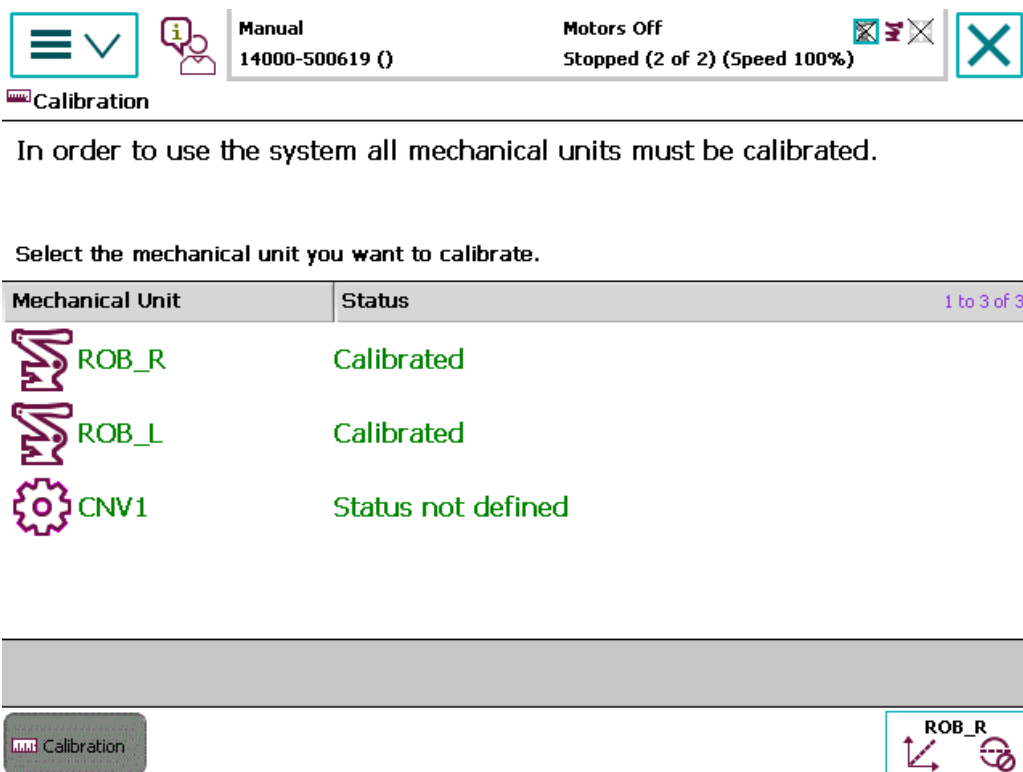
3.1.2 Spuštění kalibrační metody využívající Hallových sond a počítání otáček.

Robot pro provedení kalibrace musí být přepnut do manuálního režimu. Následně se na úvodním panelu FlexPendantu vybere kalibrace obr. 18.



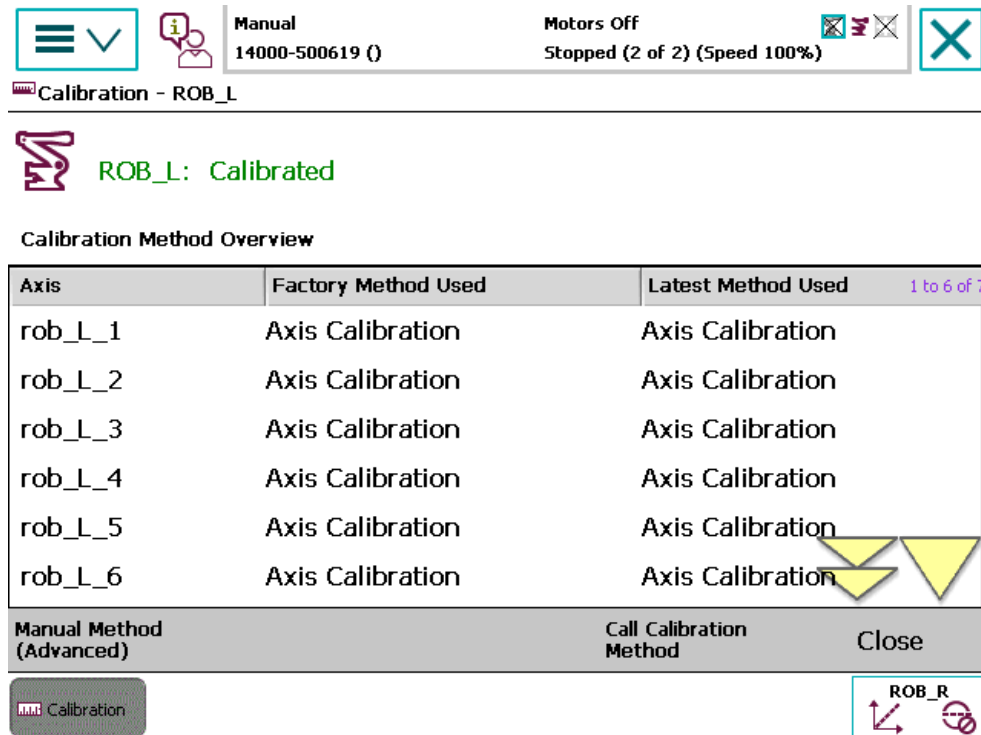
Obr. 18 Úvodní panel FlexPendantu

Na dalším okně se vybere rameno robotu, které chceme kalibrovat obr. 19.



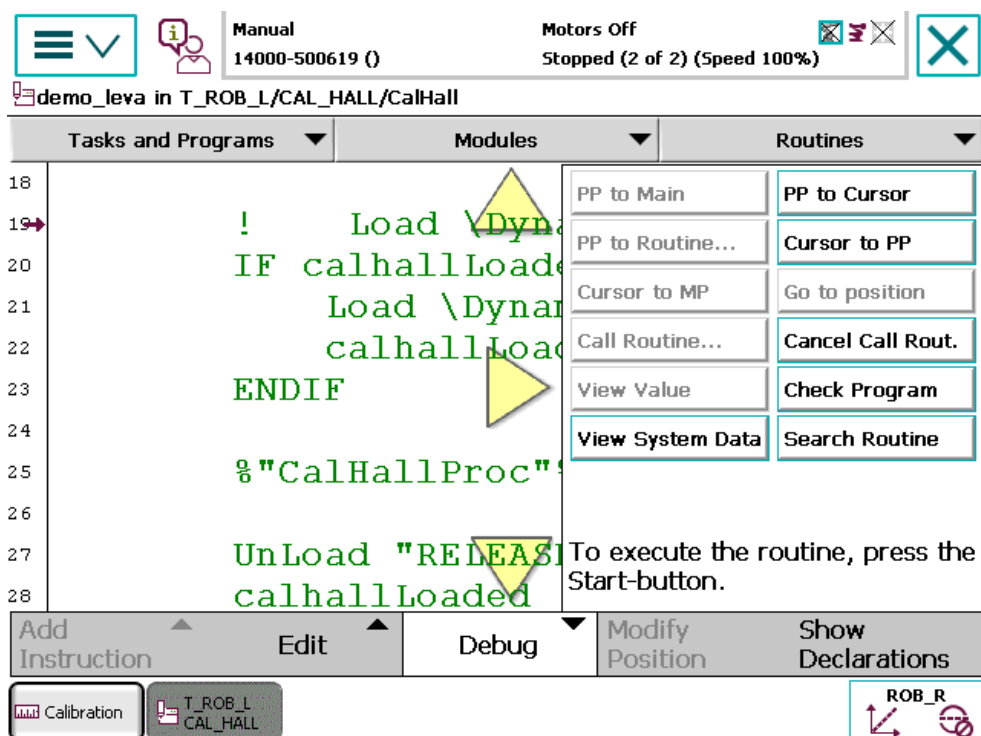
Obr. 19 Výběr ramene ke kalibraci

Ve spodní části vybereme možnost volat kalibrační metodu obr. 20.



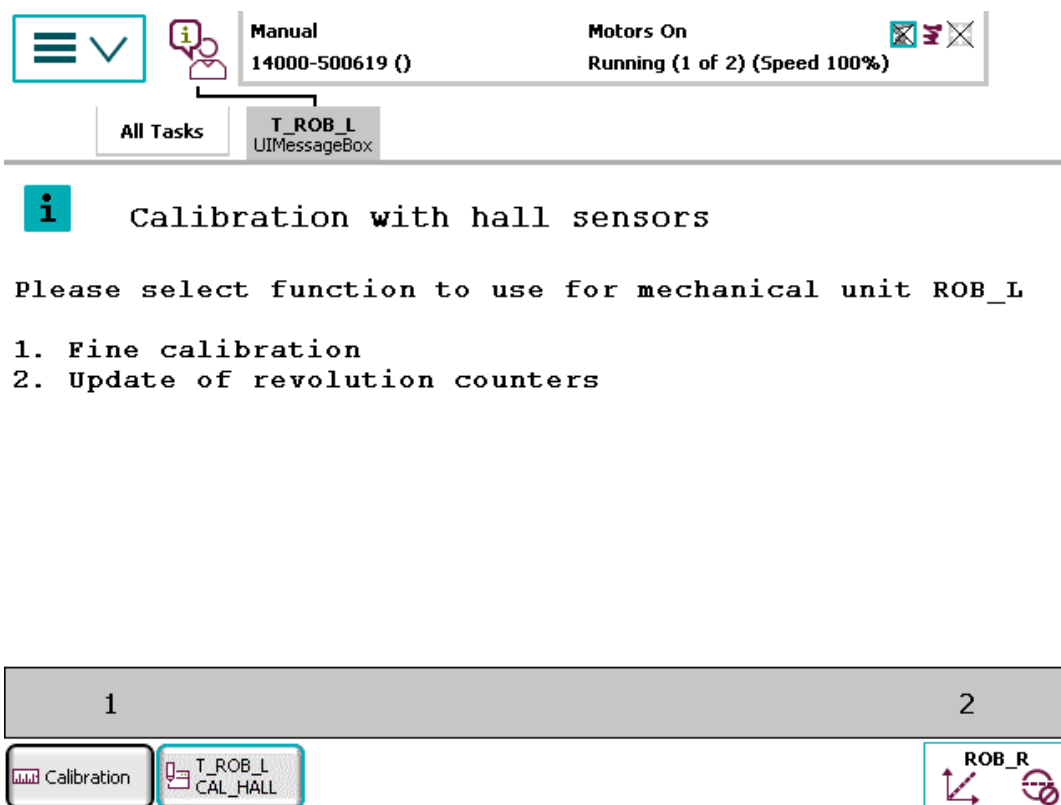
Obr. 20 Zavolání kalibrační funkce

Po stisknutí možnosti volat kalibrační metodu se načte kalibrační funkce obr. 21, která se spustí tlačítkem start. V robotu musí být nahrán nějaký program, v případě, že v robotu není nic nahráno, nabídne se možnost vytvoření nového programu.



Obr. 21 Spuštění kalibrační funkce

Na dalším okně se zobrazí možnost výběru kalibrace, nebo aktualizování počítadel otáček obr. 22.



Obr. 22 Výběr kalibrace nebo aktualizace otáček

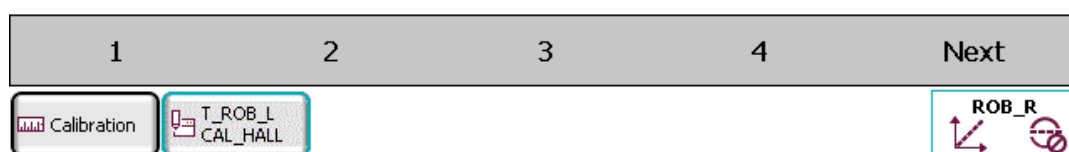
Při volbě kalibrace se ještě objeví dotaz, jestli opravdu chceme provést kalibraci. Na další obrazovce se vyberou osy obr. 23, které se mají kalibrovat a kalibrace se spustí tlačítkem OK obr. 24. Ke kalibraci je možné vybrat všechny osy naráz, ale jednotlivé osy se kalibrují postupně.



i selection of joint(s) to update

Choose joint(s) to fine calibrate ROB_L

1. []
2. []
3. []
4. []
5. []
6. []
7. []



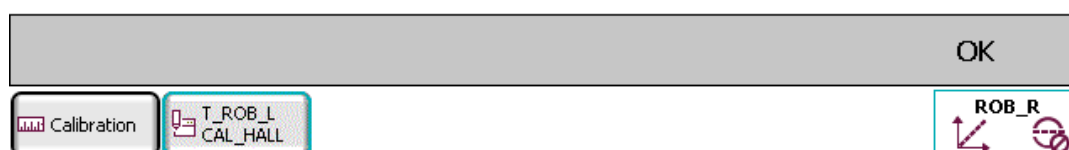
Obr. 23 Výběr os ke kalibraci



i Sampling of Hall sensor starting

This operation may take several minutes to complete.

Press OK to start!



Obr. 24 Spuštění kalibrace

Při volbě aktualizace otáček se postupuje stejným způsobem, jako při kalibrování pomocí Halloových sond (výběr os ke kalibraci a spuštění tlačítkem OK).

Kalibrační metodu lze také spustit výběrem editace programu na hlavním panelu, vybráním ramene s programem (popřípadě vytvoření nového programu), následně se pointer umístí do hlavní funkce, vybere se volat rutinu a CallHall.

K ověření, že je robot správně zkalibrován, je vhodné využít pohybový příkaz MoveAbsJ, do kterého se zadají jednotlivé stupně natočení os z tabulky 8.

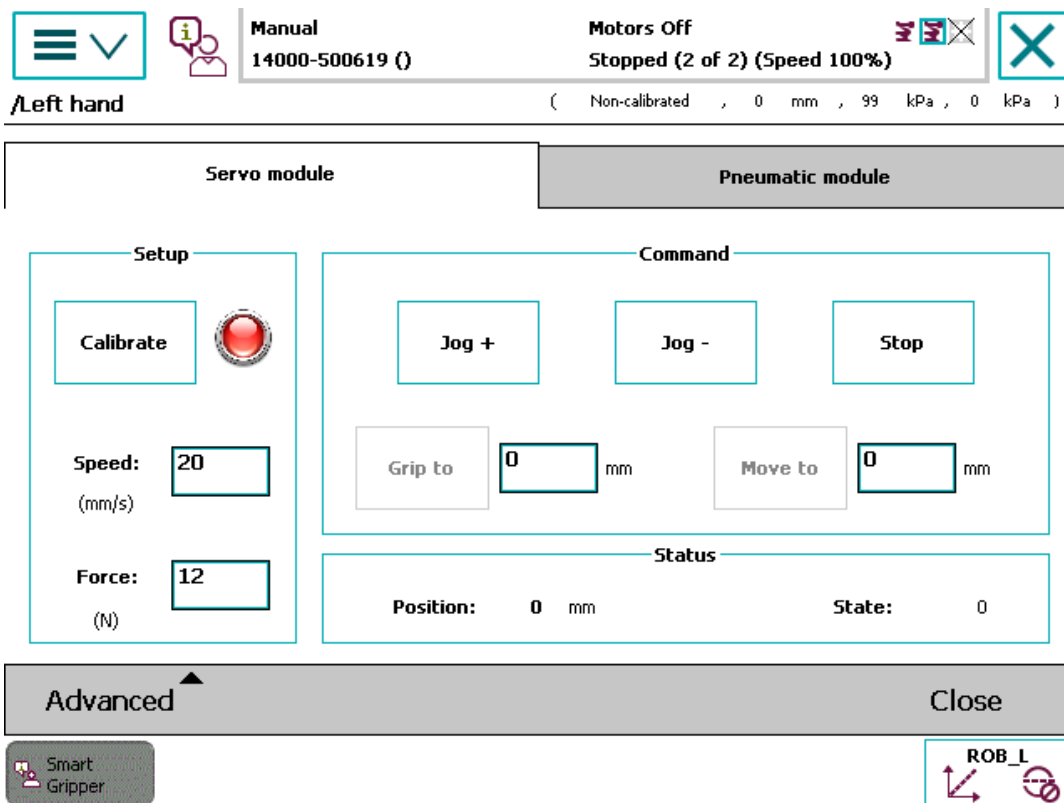
Tab. 8 Stupně natočení jednotlivých os v kalibrační poloze

Osa	Pravé rameno	Levé rameno
	[°]	[°]
1	0	0
2	-130	-130
3	30	30
4	0	0
5	40	40
6	0	0
7	-135	135

3.1.3 Kalibrace čelistí efektorů.

Čelistí efektorů je potřeba kalibrovat po každém spuštění robotu, jelikož si robot nepamatuje jejich polohu.

Na hlavním panelu se vybere záložka Smart Gripper, kde se následně objeví okno s výběrem levého a pravého efektoru. Čelisti je potřeba maximálně sevřít pomocí tlačítka "Jog –" a v dalším kroku zvolit možnost kalibrovat obr. 25.



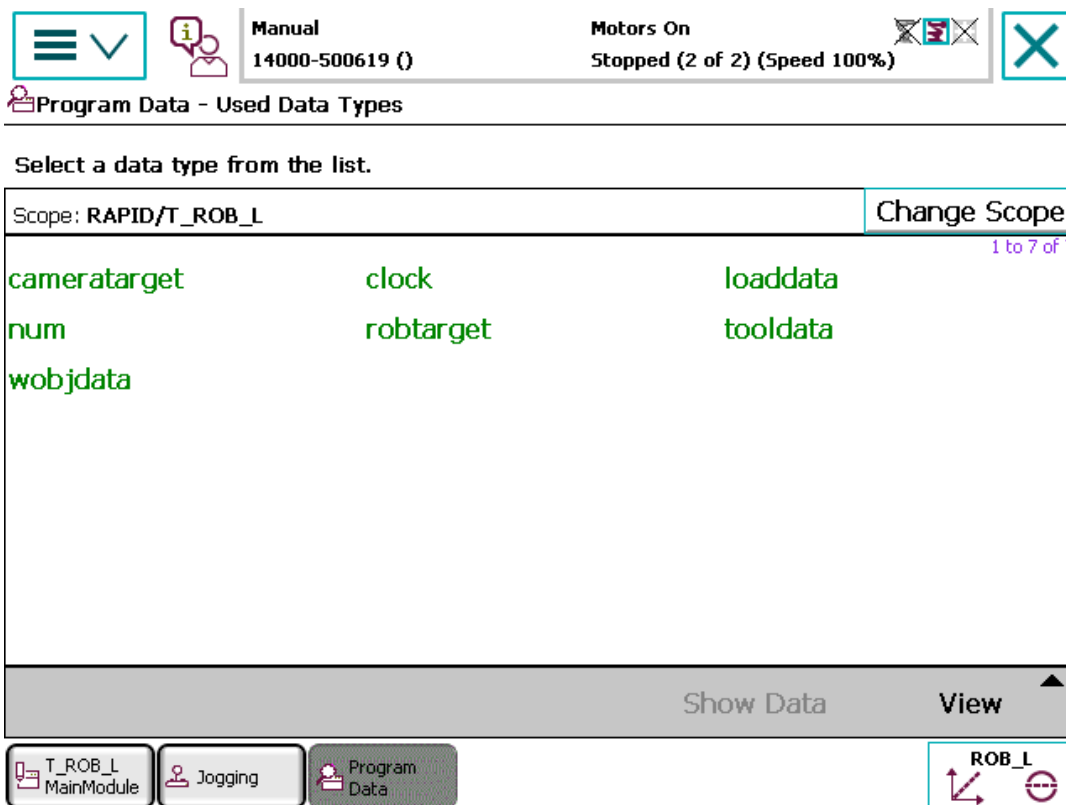
Obr. 25 Kalibrace čelistí

Variantním řešením provedení kalibrace je vložení příkazu pro kalibraci (uveden v příkazech pro serva v odd. 3.1.5.1) do kódu programu.

3.1.4 Vytvoření koordinačního systému nástroje.

Koordinační systém nástroje je vhodné vytvořit pro snazší nastavování poloh nástroje (zejména při pootočení nástroje od koordinačního systému příruby).

Pro vytvoření koordinačního systému nástroje se musí na úvodním okně FlexPendantu vybrat data programu. Na dalším otevřeném okně se vybere možnost tooldata obr. 26.



Obr. 26 Programové data robotu

Tato možnost otevře okno s vytvořenými nástroji. Ve spodní části okna je možnost "nový", která slouží pro vytvoření nového nástroje. Při vytváření nástroje je zapotřebí jej pojmenovat, určit typ uložení (trvalá, proměnná a konstantní), dostupnost nástroje (globální, lokální, nebo úloha) a zvolit, do kterého modulu chceme data uložit obr. 27.

Manual 14000-500619 () Motors On Stopped (2 of 2) (Speed 100%)

New Data Declaration

Data type: tooldata Current Task: T_ROB_L

Name: prisavkaLeva

Scope: Task

Storage type: Persistent

Task: T_ROB_L

Module: user

Routine: <None>

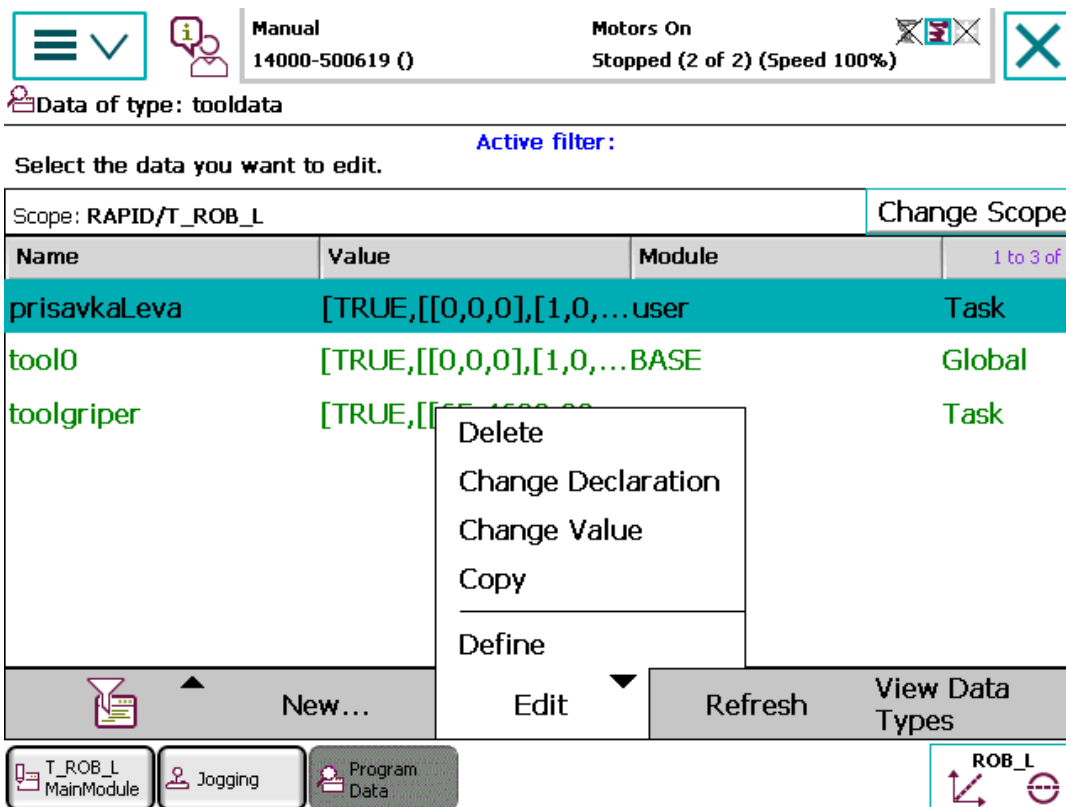
Dimension: <None>

Initial Value OK Cancel

T_ROB_L MainModule Jogging Program Data ROB_L

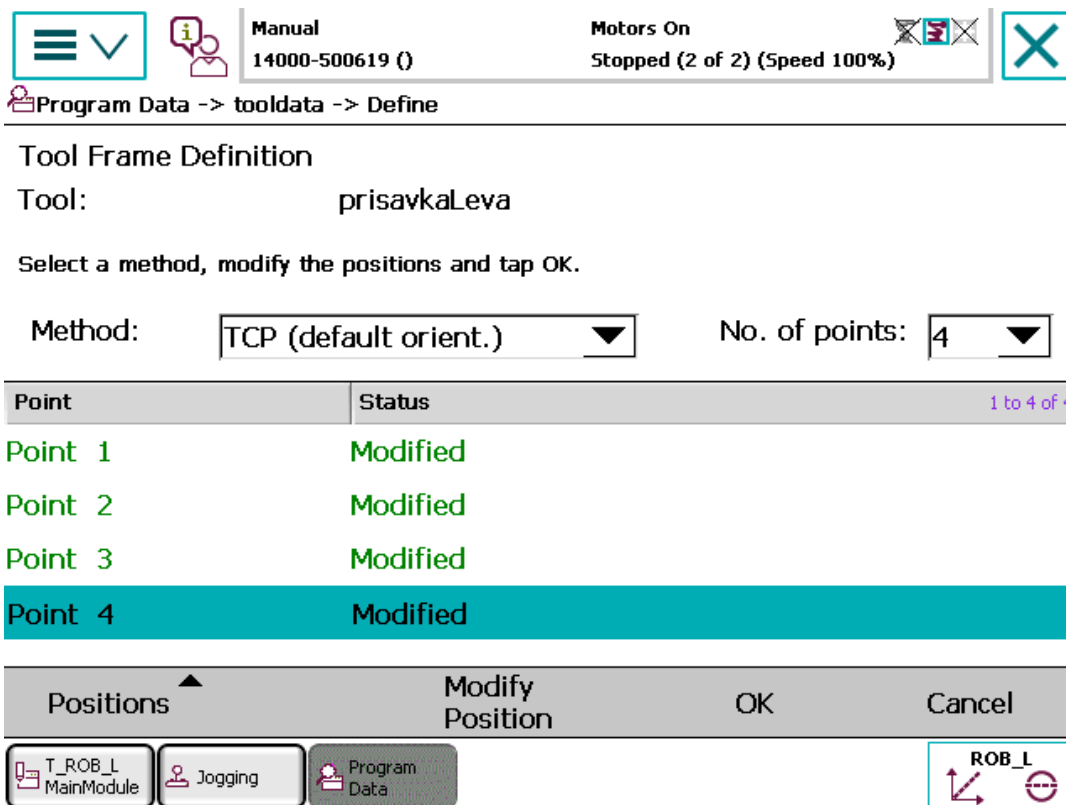
Obr. 27 Vytvoření nového nástroje

Po vyplnění dat se tlačítkem OK vytvoří nový nástroj, ale je zapotřebí ještě nadefinovat jeho koordinační systém. Na okně se seznamem nástrojů se vybere nově vytvořený nástroj a pod záložkou upravit je možnost definovat obr. 28.

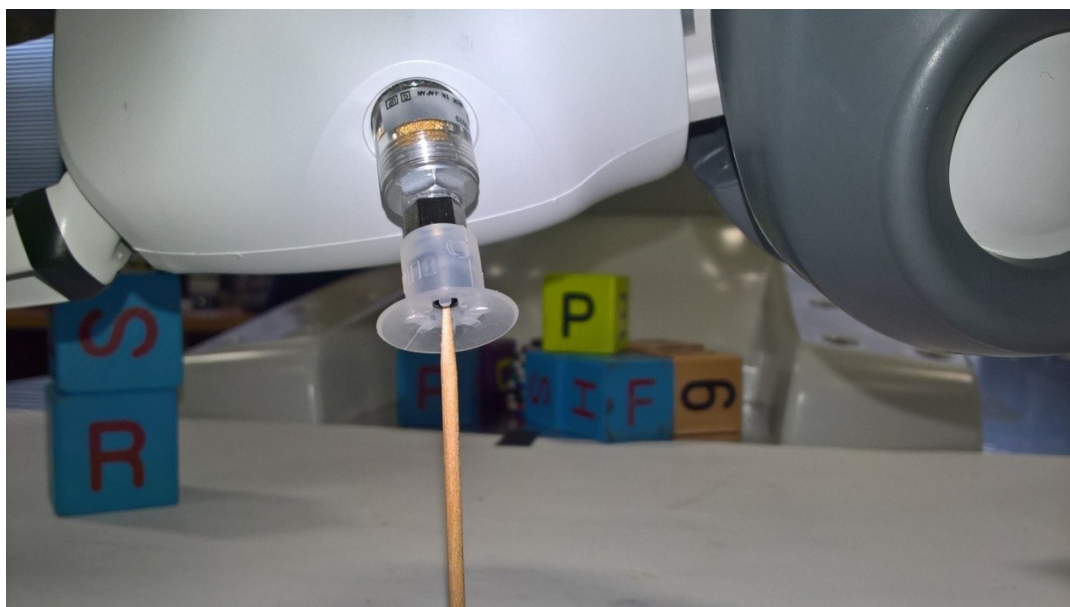


Obr. 28 Okno s výběrem nástrojů

Nejprve je třeba zvolit libovolný bod v prostoru. Definování koordinačního systému nástroje se provádí opakovaným najetím nástroje na tento zvolený bod v prostoru (obr. 30), pokaždé z jiného úhlu a kliknutím na tlačítko upravit pozici, které uloží danou pozici pod jednotlivé body (Point 1 až Point 4 v obr. 29). Výchozí počet pozic - bodů (různých poloh najetí na bod v prostoru) je nastaven na čtyři, může být nastaven v rozmezí tří až devíti.



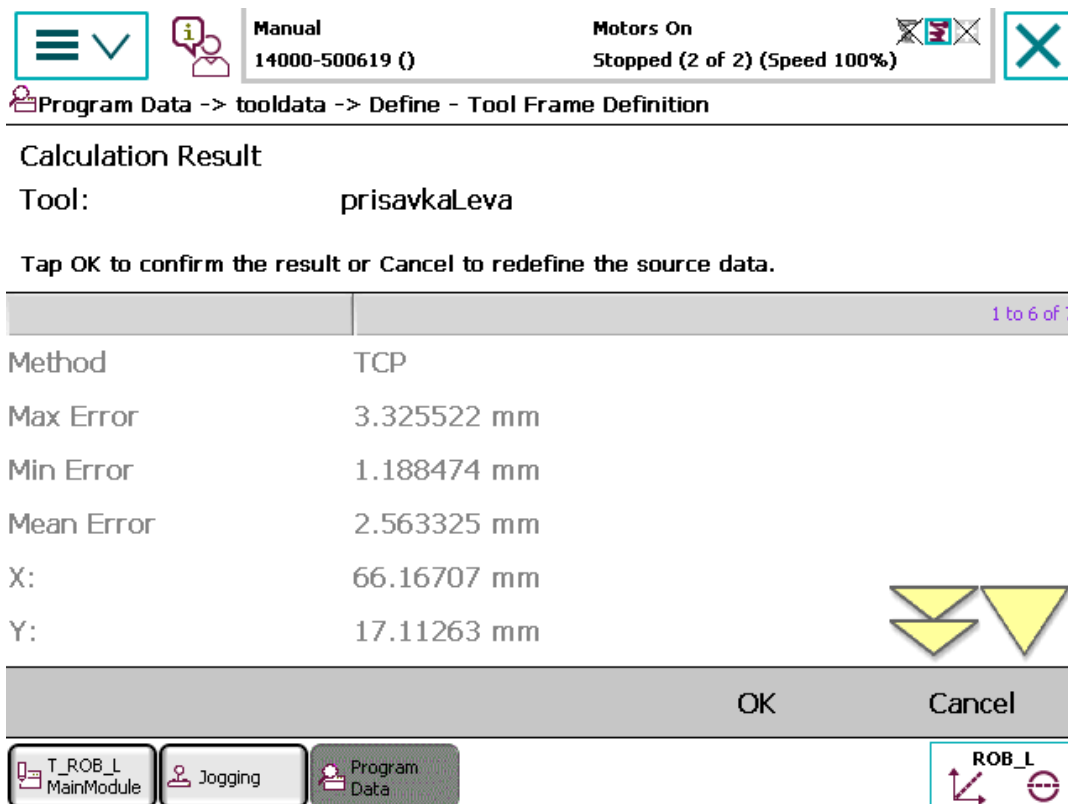
Obr. 29 definování koordinačního systému nástroje



Obr. 30 Najíždění nástrojem na bod k vytvoření jeho koordinačního systému

Kliknutím na tlačítko OK po doplnění pozic se provede výpočet polohy koordinačního systému nástroje. Maximální chyba v posunutí souřadného systému je

10 mm, pokud dojde k překročení maximální hodnoty, musí se znovu a přesněji definovat pozice najetí na zvolený bod nebo přidat další bod najetí.



Obr. 31 Okno s vypočteným koordinačním systémem

Aby nástroj mohl být použit, je u něj zapotřebí nastavit jeho hmotnost. Na okně s výběrem nástrojů označíme nástroj, zvolíme upravit, klikneme na změnit hodnotu a v novém okně doplníme hmotnost nástroje.

3.1.5 Základní programování

K programování je možné využít jak software RobotStudio, tak i FlexPendant. Cokoliv se naprogramuje v prostředí RobotStudia je taky možné naprogramovat i na FlexPendantu. YuMi umožňuje uvolnit jeho klouby a nastavit jeho ramena do požadované pozice a orientace ručně, proto je FlexPendant vhodný zejména na vytvoření Robtargetů. Pro vytvoření celého programu je lepší využít RobotStudio, které má výhodu v přehlednosti.

3.1.5.1 Základní příkazy:

Robot YuMi je vybaven dvěma originálními efekty obsahujícími přísavky, kameru a serva. Jsou pro ně vytvořené speciální příkazy:

Příkazy pro serva [24]

Hand_Initialize

Používá se k inicializaci čelistí s využitím volitelně zvolených hodnot.

Zápis: `Hand_Initialize [\maxSpd], [\holdForce], [\phyLimit], [\Calibrate];`

Příklad: `Hand_Initialize \maxSpd :=20, \holdForce:=5, \phyLimit:=15, \Calibrate;`

Argumenty:

`[\maxSpd]` je datového typu num. Určuje maximální rychlost pohybu serva efektoru v mm/s. Pokud argument není použit, využije se základní maximální hodnota 25 mm/s.

`[\holdForce]` je datového typu num. Nastavuje velikost síly, která má působit na objekt pomocí čelistí efektoru. Zadává se v Newtonech. Pokud není argument použit, budou čelisti vyvozovat sílu 10 N.

`[\phyLimit]` datový typ num. Použitím tohoto argumentu se nastavuje maximální rozsah otevření čelisti, kde hodnoty mohou být od 0 – 25 mm. Když argument není použit, využívá se základní hodnota 25 mm.

`[\Calibrate]` datový typ switch. Když je tento argument uveden, čelisti se zavřou a provede se kalibrace.

Hand_JogInward

Tento příkaz se využívá k zavření čelistí.

Hand_JogOutward

Příkaz otevírá čelisti efektoru.

Hand_MoveTo

Příkaz se používá k nastavení pozice, do které mají čelisti najet. Před použitím musí být efektor kalibrovaný.

Zápis: `Hand_MoveTo [targetPos], [\NoWait];`

Příklad: Hand_MoveTo 15, \NoWait;

Argumenty:

[targetPos] je datového typu num. Nastavení vzdálenosti pohybu od 0. Hodnota by měla být v rozsahu 0 – 25 mm. Pokud je pohyb moc malý (méně než 0,2 mm), čelisti se nepohnou.

[\NoWait] datový typ switch. Uvedení argumentu v příkaze způsobí, že při běhu programu se čelisti otevírají, nebo zavírají, a nečeká se na dosažení určité pozice.

Hand_GripInward

Určuje, že čelisti budou svírat objekt. Před použitím toho příkazu musí být servo zkalibrované.

Zápis: Hand_GripInward [\holdForce], [targetPos], [posAllowance], [\NoWait];

Příklad: Hand_GripInward \holdForce:=15, targetPos := 10, posAllowance := 1;

Argumenty:

[\holdForce] síla, jakou by měl být objekt sevřen. Rozmezí síly je 0 – 20 N. V případě, že argument není použit, využívá se maximální síla 20 N.

[targetPos] pozice, ve které by měl být objekt sevřen. Rozmezí se nastavuje od 0 po 25 mm. Pokud čelisti nesevřou objekt na dané pozici, dojde k chybě.

[\posAllowance] určuje možnou chybu, kde by mělo dojít k sevření objektu. Jestli [targetPos] není uveden, [\posAllowance] se ignoruje. V případě, že není uveden, použije se výchozí hodnota 2.

Hand_GripOutward

Funguje stejně jako předešlý příkaz, na rozdíl od svírání čelistí se čelisti rozevírají.

Hand_DoCalibrate;

Příkaz sloužící ke kalibraci čelistí

Zápis: Hand_DoCalibrate \Jog;

Argument:

[Jog] datový typ je swich. Když je argument použit, prsty efektoru se zavřou a v daném bodě se nastaví hodnota 0. Pokud argument není použit, hodnota 0 je nastavená v aktuální pozici čelistí.

Příkazy pro přísavky [24]

Příkazy ovládající přísavky obsahují číslo, určující přísavku, která se má použít.

Hand_TurnOnBlow a Hand_TurnOffBlow.

Jsou příkazy spouštějící a vypínající foukání z přísavek.

Příklad: Hand_TurnOnBlow1;

Hand_TurnOnVacuum a Hand_TurnOffVacuum

Těmito příkazy se provádí přísání objektu k přísavce a jeho uvolnění.

Příklad: Hand_TurnOffVacuum2;

3.1.5.2 Základní programování v Rapidu

Mezi základní programovací příkazy patří MoveL a MoveJ. MoveL se využívá k pohybu koordinačního systému nástroje po přímce k zadanému místu. MoveJ se využívá k rychlému pohybu ramene robotu z jednoho bodu do dalšího, kde pohyb nemusí být vykonán po přímce.

Zápis: MoveL [\Conc], DoBodu [ID], rychlost [\V], [\T], zóna [\Z], [\Inpos], nástroj [\Wobj] [\Corr] [\TLoad]

Příklad: MoveL p10, v1000, z50, PrisavkaLeva\Wobj:=wobj1;

Argumenty:

Argumenty, které nejsou v zápise uvedeny v hranatých závorkách, musí být vždy zadány, aby byl příkaz napsaný správně.

Argument DoBodu určuje, do které pozice má rameno najet. Může se zadávat buď celá informace (pozice X, Y a Z a konfigurace ramene), nebo název konstanty robtargetu, obsahující tuto informaci. [ID] je využíván pro synchronizaci pohybu. Pomocí [\Wobj] se určuje, ve kterém koordinačním systému se má robot pohybovat, v případě, že není uveden, pohybuje se v základním souřadném systému. Zóna určuje,

v jaké vzdálenosti od bodu se může přejít na další příkaz. To má za následek, že koordinační systém nástroje nedojede až do daného bodu. V případě, že je třeba dojet na požadované místo a až potom může rameno pokračovat do dalšího bodu, uvádí se slovo "fine" na rozdíl od z50 jak je uvedeno v příkladu. [Z] upřesňuje poziční přesnost nástroje. Rychlost pohybu robotu v příkladu je uvedena pomocí v1000. Může být upřesněna argumentem [\V] určujícím pohyb v mm/s nebo [\T], který stanoví, jak dlouho se má pohyb vykonávat.

Program se píše do modulu. Začíná slovem MODULE a končí slovem ENDMODULE. Na stejném řádku jako je MODULE je napsaný jeho název. Pod slovem MODULE mohou být uvedeny různé proměnné. Pod nimi musí být hlavní funkce "main" vytvořená slovy PROC main () a ukončená slovem ENDPROC. Do ní se většinou vkládají jenom názvy vytvořených dalších funkcí, aby došlo k jejich zavolání. Další vytvořené funkce musí být až za slovem ENDPROC pro hlavní funkci. K vytvoření komentáře se před každý komentovaný řádek vkládá „!“.

Příklad:

```
MODULE Module1

  CONST VAR a;

  PROC main ()

    Inicializace;

  ENDPROC

  PROC Inicializace ()

    ! Nastavit hodnotu „a“ na 5

    a:=5;

    Hand_DoCalibrate \Jog;

  ENDPROC

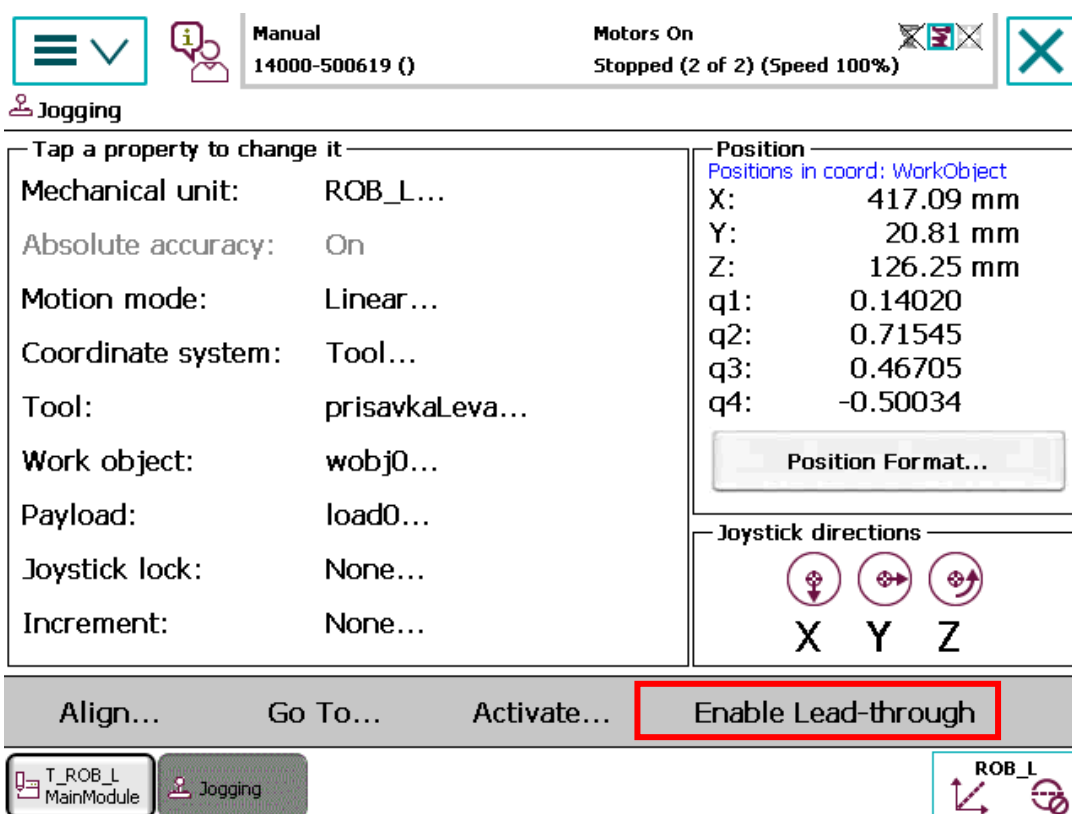
ENDMODULE
```


3.2 Využití kamery.

V této úloze se využívá pouze levé rameno robotu Yumi s efektořem se zabudovanou kamerou. Pravé rameno robotu umístíme bokem, aby nám nebránilo ve snímání ploše a pohybu levého ramene. Pro práci s kamerou je zapotřebí spustit 32bitovou verzi Robotstudia.

V úloze se využije kamera k nalezení pozic kostek ve směru osy X a osy Y a natočení kolem osy Z. Dalším krokem bude určení barvy kostky, zdali se jedná o kostku modrou, nebo zelenou a na základě jejich barvy budou kostky přemístěny na dvě zvolená místa.

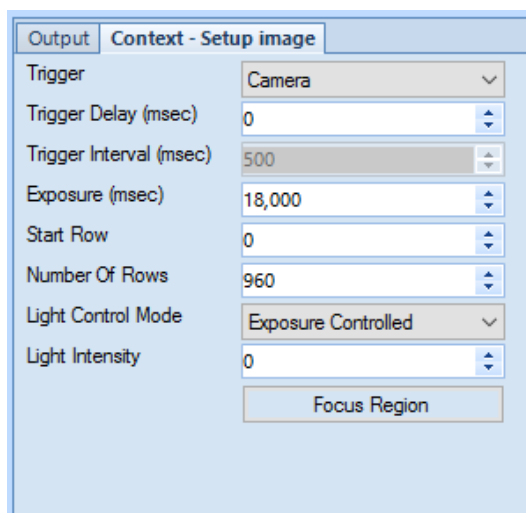
Pro vytvoření přehledového snímku pracovní plochy je zapotřebí nastavit rameno robotu do polohy, z níž bude pokaždé snímek pořizován. Rameno nastavíme do polohy povolením "ručního průchodu" na flexpendantu pod záložkou ruční přestavení obr. 32.



Obr. 32 Okno k pohybování s robotem

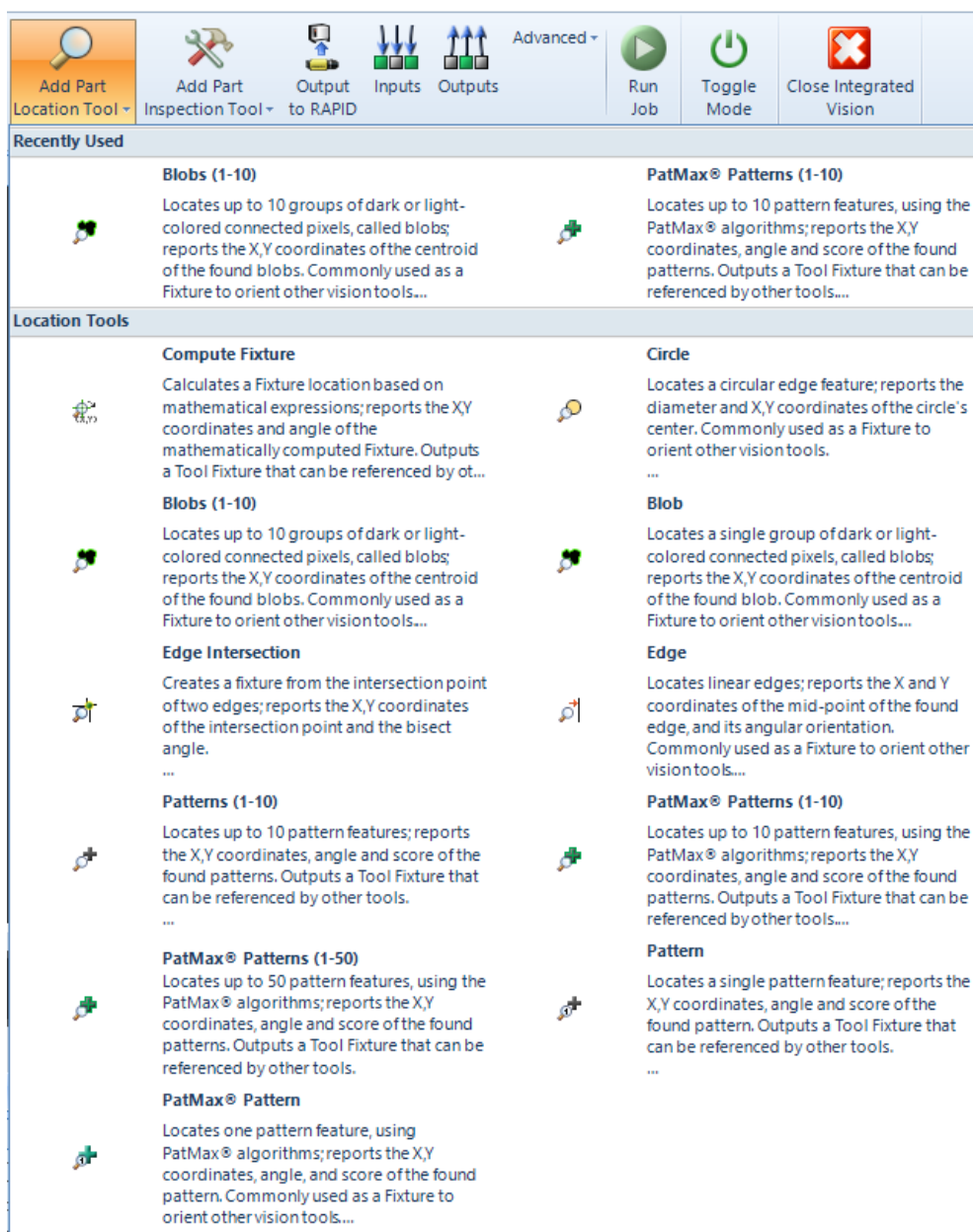
V liště kontrolér nebo rapid je ve stromu kontrolér záložka Vision systém, pomocí které se zviditelní lišta Vision, kde nastavujeme naši kameru na robotu. V liště Vision si vytvoříme pracovní úlohu pro hledání kostek, tak zvaný job. V záložce nastavení

obrázku si nastavíme parametry pro vytvoření snímku. Především je třeba nastavit dobu otevření clony a intenzitu světla. Při zvyšování hodnoty světla dochází k zapínání diody, která přisvětluje snímek obr. 33. (Pro job vyhledávající pozice a natočení kostek v této konkrétní úloze, jsem použil hodnotu světla nulovou a dlouhou dobu otevření clony, aby na kostkách nedocházelo k odlesku a byl jsem schopen najít vyhledávaný vzor.)



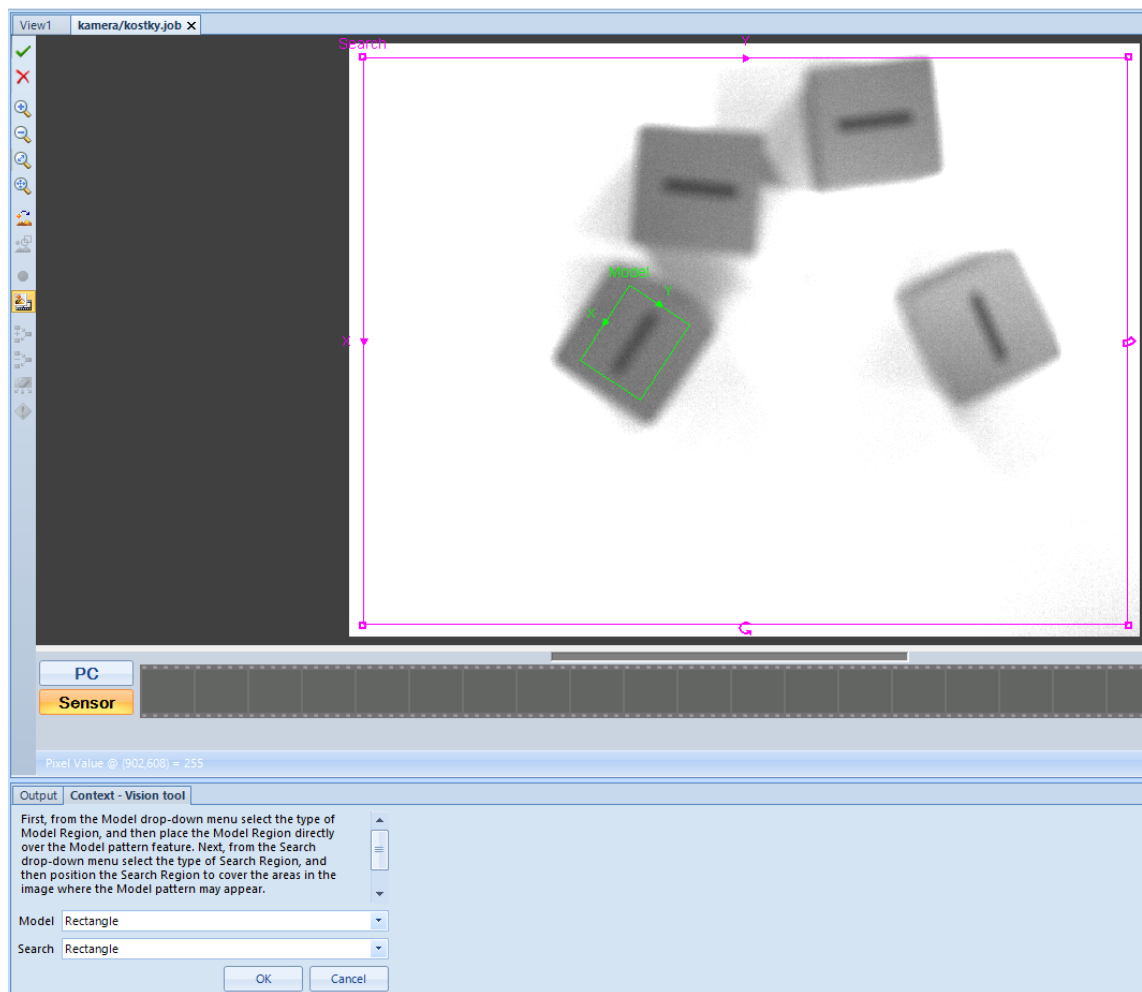
Obr. 33 Nastavení kamery

Nastavení vyhledávání vyhledávaných pozic objektů se provádí pod záložkou přidat vyhledávací nástroj. Zde je několik možností pro nastavení, podle čeho se díly budou vyhledávat a jestli bude vyhledáván pouze jeden nebo více dílů. Maximálně jich může být vyhledáváno najednou 10, kromě možnosti PatMax, která umožňuje vyhledávat až 50 dílů obr. 34.



Obr. 34 Volba vyhledávací funkce

K vyhledání pozic kostek v této úloze jsem využil možnost PatMax do 10 dílů. Na pracovní ploše musí být nachystány vyhledávané díly, protože kamera pořídí snímek pracovní plochy. Na snímku si vybereme hledaný vzor, v tomto konkrétním případě jsem hledal kostky podle písmene I, a dále můžeme upravit prohledávanou velikost snímku obr. 35.



Obr. 35 Nastavení vyhledávaného vzoru

Dále se nastavuje maximální počet vzorů, který může být nalezen. Je zapotřebí nastavit rotaci, o kolik stupňů může být hledaný vzor otočen oproti původní orientaci vzoru.

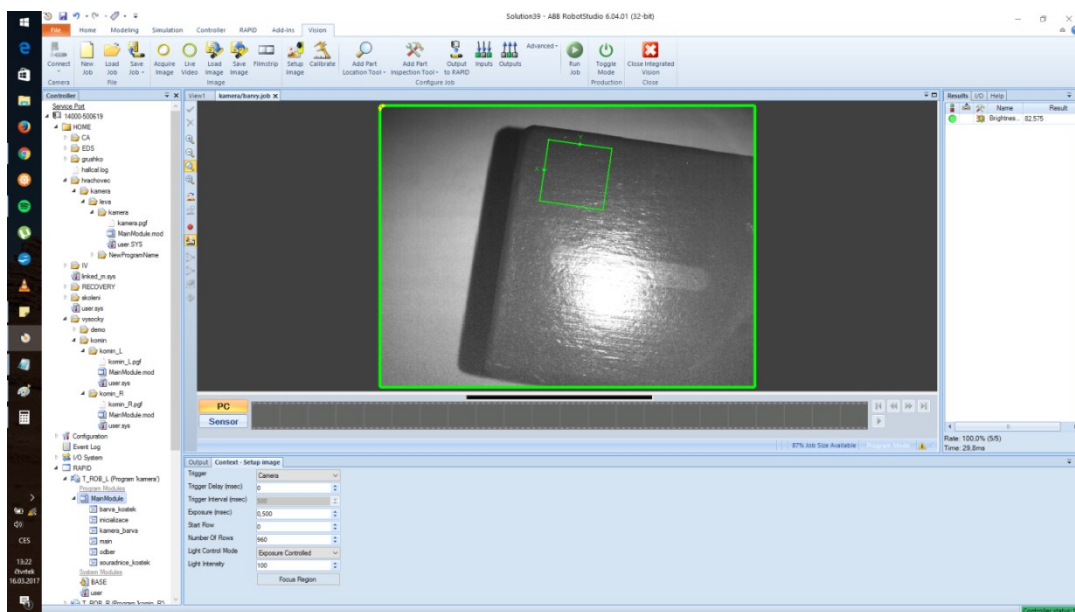
Poslední věc potřebná v nastavení Vision je nastavení výstupních veličin do rapidu, aby mohly být použity k programování. Nastavení se provádí pod záložkou výstup do rapidu. V úloze jsem nastavil výstup souřadnic dílů ve směru os kamery X a Y a rotaci kolem osy Z a dále počet nalezených dílů obr. 36.

Camera target inputs for selected item type					Resulting camera target	
Component	Data type	Group	Result	Value	Component	Value
Position x	num	Pattern_1	Feature.X [D-3]	854.382 1034.845 571.013 456.422	x	854.382 1034.845 571.013 456.422
Position y	num	Pattern_1	Feature.Y [D-3]	127.272 464.263 235.488 488.682	y	127.272 464.263 235.488 488.682
Rotation z	num	Pattern_1	Feature.Angle [D-3]	-13.005 59.506 -0.210 -243.183	z	0
Value 1	num	Pattern_1	Number_Found	4,000	anglex	0
Value 2	num	Constant	0	0	angley	0
Value 3	num	Constant	0	0	anglez	-13.005 59.506 -0.210 -243.183
Value 4	num	Constant	0	0	val1	4,000
Value 5	num	Constant	0	0	val2	0
String 1	string	Constant	0	0	val3	0
String 2	string	Constant	0	0	val4	0

Obr. 36 Nastavení výstupu do Rapidu

V rapidu se připraví proměnná typu camererarget, do které jsou zapisovány výstupy z kamery. Jelikož v úloze byla kamera při tvoření přehledového snímku příliš vysoko a nebyla schopna zaostřit kalibrační mřížku kamery a tedy výstupní hodnoty posunutí v osách X a Y, neodpovídaly hodnoty velikostem v pracovním systému, a bylo zapotřebí je přepočítávat měřítkem, které se muselo odměřit pro obě osy. Nulová hodnota souřadného systému kamery se nenachází v nulovém bodu pracovního prostoru robotu, je třeba nastavit offset jejich os. Osy souřadného systému kamery jsou orientované tak, že osa Y kamery odpovídá ose X pracovního souřadného systému a osa X kamery opačnému směru osy Y pracovní plochy.

Pro zjišťování barev kostek, je nutné si vytvořit další job pro kameru. Zde už se nehledá jejich pozice, ale provádí se jejich inspekce. Kamera je jenom černobílá, a tak rozeznávání barvy je provedeno podle funkce světlosti, která se nachází pod záložkou přidat inspekční nástroj dílu. U tohoto jobu je taky zapotřebí nastavit, z které části snímku bude inspekce prováděna, nastavit kameru (čas otevření clony a světlost) obr. 37 a výstupní hodnotu pro rapid.



Obr. 37 Nastavení kamery pro určení barvy kostky

K využití kamery je třeba v Rapidu vytvořit kód obsahující následující příkazy:

CamSetProgramMode kamera;

CamStartLoadJob kamera, "kostky";

CamWaitLoadJob kamera;

CamSetRunMode kamera;

CamReqImage kamera;

CamGetResult kamera, cameratarget1;

První příkaz určuje, která kamera se má nastavit do programovacího módu, aby se mohl nahrát job (kamera na efektoru YuMiho je pojmenovaná kamera). Druhý příkaz určuje, který job bude pro kterou kameru nahrán. Třetí příkaz pouze zajistí, že se nebude pokračovat v programu dříve, než bude daný job nahrán. Další příkaz slouží k nastavení kamery do provozního režimu, aby mohly být pořizovány snímky. Předposlední příkaz vytvoří snímek a posledním příkazem je daný snímek uložen do proměnné cameratarget1;

Příklad, který demonstruje, jak může vypadat celý programový kód nalezení poloh kostek, určení jejich barvy a přemístění na jiné místo podle jejich barvy, je uveden v příloze A. Video, ukazující tuto úlohu, je v příloze B.

3.3 Synchronizace ramen

Robot YuMi je vybaven dvěma robotickými rameny, každé robotické rameno představuje jeden robot, připojený na společnou řídicí jednotku. Je proto vhodné se seznámit se synchronizací ramen.

Synchronizace ramen může představovat buď synchronizování úloh, kdy dochází k čekání jednoho ramene v nastavené synchronizační poloze do doby, než se dostane druhé rameno do části programového kódu se synchronizačním bodem, nebo synchronizování pohybu dvou ramen. Tato funkce je využívána převážně na svařujícím pracovišti se dvěma roboty svařujícími ve stejnou chvíli a polohovadlem otáčejícím svařovanou součást. Synchronizovaný pohyb může být závislý na jednom rameni (jedno rameno vede druhé), nebo každé rameno vykonává jiný pohyb.

3.3.1 Odložení dvou kostek na určené místo a následná manipulace s nimi oběma rameny:

Na začátku obou programů (levého ramene a pravého ramene) je zapotřebí vytvořit proměnnou typu pers, do níž se vytvoří datový typ tasks s počtem synchronizovaných mechanických jednotek. Dále se přiřadí názvy mechanických jednotek.

```
PERS tasks Task_list{2} := [{"T_ROB_L"}, {"T_ROB_R"}];
```

Task_list je vytvořený název pro datový typ tasks a jeho pomocí se na danou proměnnou odvolává. T_ROB_L a T_ROB_R jsou jednotlivá ramena robotu.

Dále musí být vytvořeny proměnné VAR s datovým typem syncident, který má funkci synchronizačních bodů.

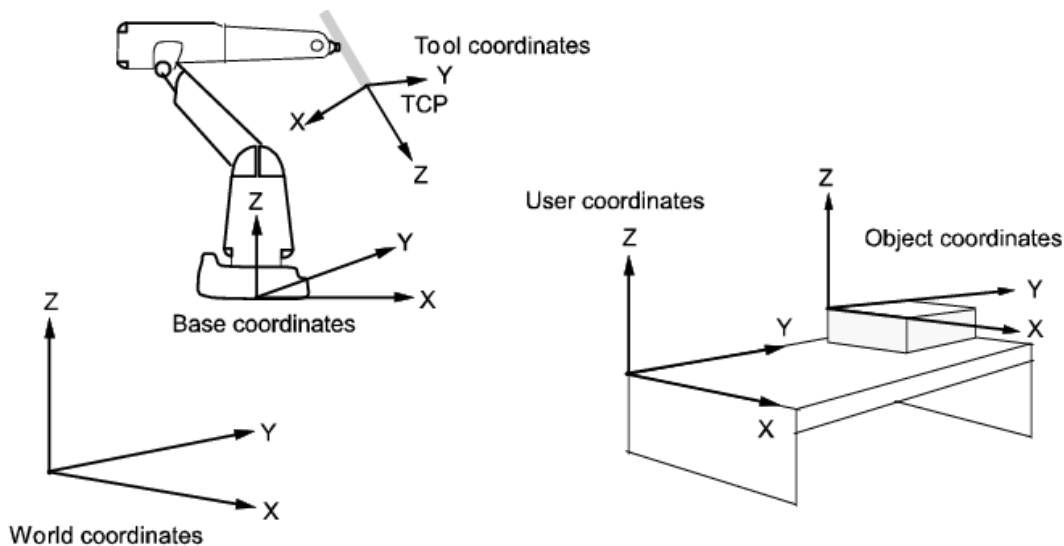
VAR syncident Sync1;

Sync1 je název proměnné syncident, a jakmile jedno rameno robotu dojde v kódu k dané proměnné, čeká, než ke stejné proměnné dojde i druhé rameno robotu.

Pro synchronizaci pohybu, v případech, kdy jsou pohyby závislé na koordinačním systému, je zapotřebí ještě vytvořit pracovní objekt.

PERS wobjdata dynam := [FALSE, FALSE, "ROB_R", [[0,0,0],[1,0,0,0]], [[0,0,0],[1,0,0,0]]];

Dynam je názvem vytvořeného pracovního objektu. Prvním FALSE ve vytvořeném pracovním objektu patří pro "robhold", jež je datového typu "bool", a určuje, zdali robot drží pracovní objekt a využívá se stacionární nástroj, nebo robot drží nástroj a nedrží pracovní objekt. Druhé FALSE patří pro ufprog, které určuje, jestli se jedná o nehybný, nebo pohyblivý koordinační systém. Pro můj případ, kdy u synchronizovaného pohybu sleduje jedno rameno druhé, musí být nastaveno, že se nejedná o nehybný koordinační systém. V tomto případě musí být vyplněná i hodnota, do které se nastavuje název mechanické jednotky, ke které bude připnut a s ní se hýbat. První hranatá závorka s číselnými hodnotami patří translačním hodnotám "uframe" (uživatelského koordinačního systému), druhá závorka určuje jeho natočení ve kvaternionech, ve třetí závorce jsou translační hodnoty "oframe" (objektu) a čtvrtá závorka představuje jeho natočení. Pro jasnější pochopení jsou koordinační systémy znázorněny na obr. 38. Hodnoty uframe a oframe nemusíme přesně nastavovat a můžeme zadat hodnoty světového koordinačního systému, jelikož se připe k jednomu ramenu a s tím se bude pohybovat.



Obr. 38 Ukázka koordinačních systémů [23]

Zápis pro čekání jednotlivých ramen je naprosto jednoduchý. Slouží pro to funkce `WaitSyncTask`, kde se zapíše synchronizační proměnná a proměnná se synchronizovanými roboty.

```
WaitSyncTask Sync1, Task_list;
```

Stejný zápis se vloží do obou kódu robotu.

Synchronizace pohybu se spouští příkazem `SyncMoveOn` s uvedením synchronizační polohy a zvolením synchronizačního tasku. Ukončení synchronizačního pohybu je provedeno příkazem `SyncMoveOff` a synchronizační pozicí.

```
SyncMoveOn Sync2, Task_list;
```

Každý pohyb musí být spojen s identifikačním číslem, aby bylo jasné, které pohyby se synchronizují. Nastavené zóny v pohybu musí být shodné. U sledujícího robotu se musí nastavit, že nástroj se pohybuje ve vytvořeném pohyblivém pracovním objektu.

```
ROB_L:
```

```
SyncMoveOn Sync2, Task_list;
```

```
MoveL Pozice10\ID:=10,v500,z50,too_YuMiGripper_S_C_VI\WObj:=Dynam;
```

```
MoveL Pozice10\ID:=20,v500,z50,too_YuMiGripper_S_C_VI\WObj:=Dynam;
```


SyncMoveOff Sync3;

ROB_R:

SyncMoveOn Sync2, Task_list;

MoveL Pozice10\ID:=10,v500,z50,too_YuMiGripper_S_C_V1;

MoveL Pozice20\ID:=20,v500,z50,too_YuMiGripper_S_C_V1;

SyncMoveOff Sync3;

Stejným způsobem se vytváří kód, kde jedno rameno hýbe s objektem a druhé na něm vykonává operaci. Nežadává se pořád stejná pozice, jak je uvedeno v příkladu levého ramene robotu, kdy rameno jenom sleduje pohyb druhého ramene, ale zadávají se jednotlivé požadované pohyby.

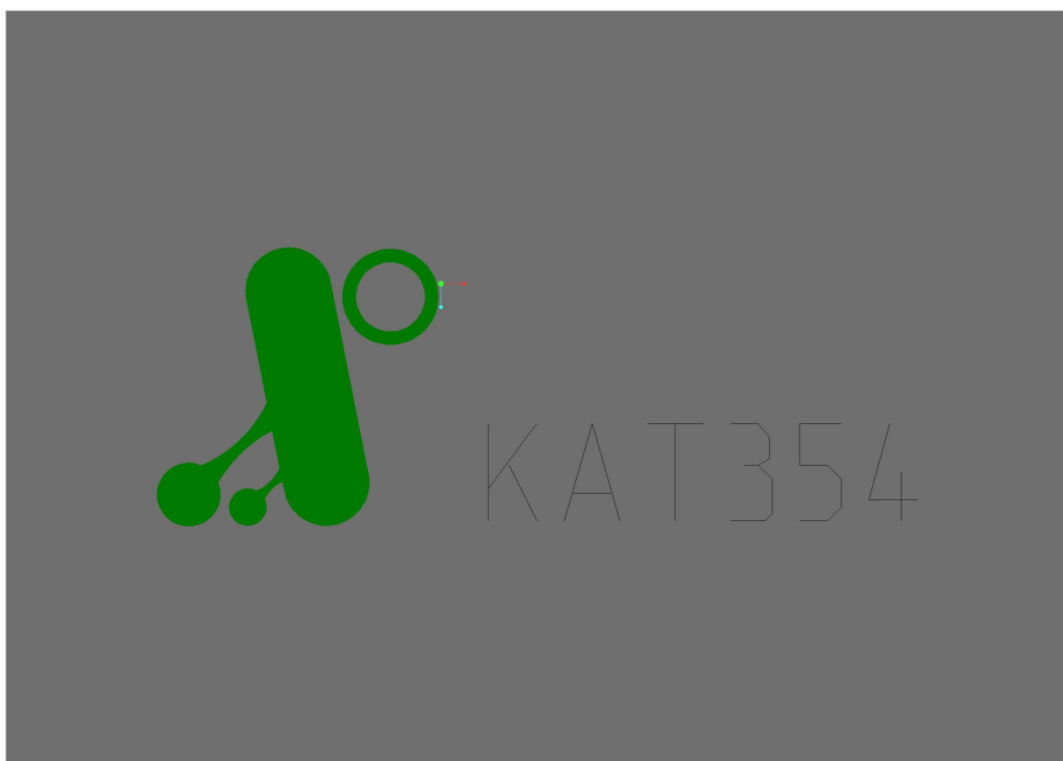
Kód pro vytvořenou celou úlohu je uveden v příloze C, video, ukazující úlohu, je v příloze D.

4 Demonstrační úloha

Jako demonstrační úlohu jsem vytvořil program, ve kterém robot uchopí fix umístěný na konkrétním místě, pomocí druhého ramene robotu sundá z fixu víčko a následně nakreslí na papír znak katedry Robotiky. Poloha papíru je určena pomocí kamery, kterou je robot vybaven. Variantním řešením této úlohy, které se dá průmyslově využít, může být nanášení lepidla robotem na součásti, v rámci kooperace s pracovníkem.

4.1 Vytvoření cesty

Cesta, kterou robot opisuje, aby nakreslil znak katedry, je vytvořena pomocí 3D modelu, vymodelovaného v software Creo Parametric 3. Body cesty znaku byly vytvořeny ve virtuální simulaci pomocí příkazu Automatická cesta. Souřadnice jednotlivě vytvořených bodů byly použity k vypočtení vzdálenosti k dalšímu bodu v osách X a Y. Funkcí MoveL a Offs, kde jsem zadával vypočtené vzdálenosti, jsem najížděl do jednotlivých bodů a vytvářel jsem robtargety funkcí CRobT. Hodnoty vytvořených robtargetů jsem kopíroval v RobotStudios ze záložky sledující hodnoty proměnných do jednotlivých bodů, aby robot nemusel vždy před kreslením projíždět všechny body a aby byly známy jejich hodnoty.



Obr. 39 3D model pro vytvoření cesty

Nápis je vytvořen pomocí skici v Creu se zvoleným typem písma ISO. Nápis nelze přenést do RobotStudia tak, aby podle něj byly vygenerované body, proto bylo nutno odměřit vzdálenosti všech bodů v Creu a zadat je do funkce Offs při pohybu MoveL během vytváření bodů jednotlivých částí písmen.

4.2 Poloha papíru

Na papíru je nakreslené písmeno L. Musí být nakresleno v přesně zadané části papíru. Poloha písmene je hledaná pomocí vyhledávací funkce PatMax, kde výstupem je poloha X a Y a natočení kolem osy Z. Všechny souřadnice jsou určeny v souřadném systému pracovního objektu kamery.

K určení polohy papíru byl vytvořen pomocný pracovní objekt, který má souřadnice X a Y stejné jako jsou souřadnice pracovního objektu kamery. Tyto souřadnice jsem určil vytvořením snímku s písmenem L, ze kterého jsem získal souřadnice písmene v souřadnicovém systému pracovního objektu kamery. Najel jsem robotem na písmeno L a pomocí MoveL a Offs s hodnotami polohy písmene jsem robot dostal na místo, kde se nachází počátek pracovního objektu kamery. Souřadnice aktuální polohy ramene v globálním souřadnicovém systému jsem použil pro souřadnice pomocného pracovního objektu.

Poloha papíru je přepočítávána vůči pomocnému pracovnímu objektu následující vytvořenou funkcí:

PROC prepocetni_wobj()

L{1,1}:=cameratargetL.cframe.trans.x;

L{2,1}:=cameratargetL.cframe.trans.y;

L{3,1}:=EulerZYX(\Z,cameratargetL.cframe.rot);

anglex:=EulerZYX(\X,kamera_0.uframe.rot);

angley:=EulerZYX(\Y,kamera_0.uframe.rot);

anglez:=EulerZYX(\Z,kamera_0.uframe.rot);

anglez_nove:=anglez+L{3,1};

papir.uframe.rot:=OrientZYX(anglez_nove,angley,anglex);

papir.uframe.trans.x:=kamera_0.uframe.trans.x-L{1,1}+200;

```
papir.uframe.trans.y:=kamera_0.uframe.trans.y-L{2,I}+100;
```

```
papir.uframe.trans.z:=kamera_0.uframe.trans.z;
```

ENDPROC

V této funkci první tři řádky ukládají polohu X a Y a natočení kolem osy Z do proměnné L. Další tři řádky převádějí hodnoty natočení pomocného pracovního objektu z kvaternionů do stupňů. Následující řádek vypočítává otočení papíru kolem osy Z. Osmý řádek určuje, jak je papír orientován a poslední tři řádky určují polohu papíru.

Celý kód je v příloze E a video úlohy v příloze F.

5 Závěr

V úvodní části diplomové práce byl vysvětlen pojem kolaborativních robotů, byly uvedeny možné oblasti pro jejich využití a byla provedena analýza stavů kolaborativních robotů, které jsou v současné době na trhu. Dále byly zjištěny metody kolaborativních operací a důkladně popsány požadavky na ně.

Další část diplomové práce byla věnována vytvoření výukových úloh, které byly zaměřeny na popis základních operací s robotem YuMu. První úloha byla zaměřená na vysvětlení způsobu kalibrace robotu YuMi, vytvoření koordinačního systému nástroje a na základ programování pomocí programovacího jazyka Rapid. Druhá úloha vysvětlovala využití kamery integrované v efektoru robotu a třetí úloha se věnovala vysvětlení synchronizace ramen robotu.

V poslední části diplomové práce byl popsán postup při vytvoření demonstrační úlohy, v rámci které byl vytvořen programový kód pro robot YuMi, na základě kterého robot nakreslil logo katedry Robotiky. Celá demonstrační úloha byla také zdokumentována na videu, které je součástí přílohy diplomové práce. Demonstrační úloha byla zvolena na základě snadné dostupnosti potřebných nástrojů a možnosti snadné interpretace pro využití v průmyslovém nasazení robotu.

Poděkování:

Děkuji prof. Dr. Ing. Petru Novákovi za odborné vedení mé diplomové práce a také Ing. Aleši Vysockému za cenné rady při zpracování praktické části mé diplomové práce.

6 Použité zdroje

- [1] Packaging Digest [Online]. [cit. 2016-11-11]
<<http://www.packagingdigest.com/robotics/what-are-collaborative-robots-and-why-should-you-care1505>>
- [2] World economic forumu [Online]. [cit. 2016-11-11]
<<https://www.weforum.org/agenda/2015/06/what-is-collaborative-robotics/>>
- [3] Génération Robot [Online]. [cit. 2016-11-11]
<<http://www.generationrobots.com/blog/wp-content/uploads/2014/01/Lightweight-robots-and-Collaborative-Robotics.pdf>>
- [4] ABB [Online]. [cit. 2016-11-11] <<http://new.abb.com/products/robotics/industrial-robots/yumi>>
- [5] ROS.org [Online]. [cit. 2016-11-11] <<http://wiki.ros.org/Robots/Roberta>>
- [6] Robotiq [Online]. [cit. 2016-11-11] <<http://blog.robotiq.com/bid/71048/New-Collaborative-Robot-called-Roberta-Your-New-Best-Friend>>
- [7] The robot report [Online]. [cit. 2016-11-11]
<<https://www.therobotreport.com/news/co-bots-on-the-move>>
- [8] Fanuc [Online]. [cit.2016-11-11] <<http://www.fanuc.eu/cz/en/robots/robot-filter-page/collaborative-robots/collaborative-cr35ia>>
- [9] Kuka [Online]. [cit. 2016-11-11] <<https://www.kuka.com/cs-cz/produkty.-slu%C5%BEby/robotick%C3%A9-syst%C3%A9my/pr%C5%AFmyslov%C3%A9-roboty/lbr%C2%A0iiwa>>
- [10] ROS COMPONENTS [Online]. [cit. 2016-11-11]
<https://www.roscomponents.com/en/robotic-arms/92-our-i-5.html#/our_i5_teach_pendant-no/aubo_i5_control_box-no>
- [11] Franka emika [Online]. [cit. 2016-11-11] <<https://www.franka.de/>>
- [12] Universal robots [Online]. [cit. 2016-11-11] <<https://www.universal-robots.com/products/>>

- [13] Rethink robotics [Online]. [cit. 2016-11-11]
<<http://www.rethinkrobotics.com/sawyer/>>
- [14] Rethink robotics [Online]. [cit. 2016-11-11]
<<http://www.rethinkrobotics.com/baxter/>>
- [15] Precise automatik [Online]. [cit. 2016-11-11]
<<http://preciseautomation.com/PF400.html>>
- [16] Precise automatik [Online]. [cit. 2016-11-11]
<<http://preciseautomation.com/PP100.html>>
- [17] Precise automatik [Online]. [cit. 2016-11-11]
<<http://preciseautomation.com/PAVP6.html>>
- [18] Yaskawa [Online]. [cit. 2016-11-11] <http://www.yaskawa.eu.com/nl/news-events/news/article/news/motoman-hc10-collaborative-robot-safe-and-flexible-interaction/?tx_news_pi1%5Bcontroller%5D=News&tx_news_pi1%5Baction%5D=detail&cHash=e91562b0cf2f02dac3884111cfacf6dc>
- [19] Nachi robotic systems inc [Online]. [cit. 2016-11-11]
<<http://www.nachirobotics.com/mz04>>
- [20] MKR-SYSTEME GMBH [Online]. [cit. 2016-11-11] <http://www.mrk-systeme.de/images/downloads/prospekt_mrk_en.pdf>
- [21] ČSN EN ISO 10218-2:2011 Roboty a robotická zařízení – Požadavky na bezpečnost průmyslových robotů- Část 2: Systémy robotů a integrace, Praha: Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2011. 68 s.
- [22] ISO TS 15066 Robots and robotic device – Collaborative robots, Geneva: ISO copyright office, 2016. 33 s.
- [23] Návod RobotStudio 6.04.01 32-bit [Software] 16. Prosince 2016
- [24] Product manual – Grippers for IRB 14000, Revision: D

7 Seznam příloh

Příloha A – Kód pro výukovou úlohu s kamerou

Příloha B – Video výukové úlohy s kamerou

Příloha C – Kód pro výukovou úlohu využívající obou ramen

Příloha D – Video výukové úlohy využívající obou ramen

Příloha E – Kód pro vytvořenou demonstrační úlohu

Příloha F – Video demonstrační úlohy

8 Přílohy

Příloha A

MODULE MainModule

```
CONST robtargt scan:=[[402.09,60.53,358.92],[0.0583795,0.0121557,-  
0.688084,0.723177],[-1,1,-2,5],[-159.023,9E+09,9E+09,9E+09,9E+09]];
```

```
CONST robtargt nula:=[[239.67,33.82,128.45],[0.0143715,0.0412622,0.682041,-  
0.730007],[-2,1,-1,5],[-161.676,9E+09,9E+09,9E+09,9E+09]];
```

```
CONST robtargt nula10:=[[239.88,33.63,226.58],[0.0338073,-0.483059,0.874926,-  
0.00399792],[-2,-1,0,5],[-162.266,9E+09,9E+09,9E+09,9E+09]];
```

```
CONST robtargt modra10:=[[270.76,-98.79,241.99],[0.0148382,-  
0.337404,0.941186,0.0103576],[-2,-1,0,5],[-  
132.487,9E+09,9E+09,9E+09,9E+09]];
```

```
CONST robtargt  
zelena10:=[[274.73,221.89,229.08],[0.00472298,0.964075,0.265416,0.00959932],[-1,-  
1,-2,5],[-152.63,9E+09,9E+09,9E+09,9E+09]];
```

```
VAR cameratarget cameratarget1 {4};
```

```
VAR cameratarget cameratarget2;
```

```
VAR num pocet;
```

```
VAR num modra:=0;
```

```
VAR num zelena:=0;
```

```
VAR num vyska_kostky:=35;
```

```
VAR num kostky{4,4};
```

```
CONST num scalex:=6;
```

```
CONST num scaley:=5.77;
```

```
CONST num offsx:=867/6;  
CONST num offsy:=1218/5.77;  
CONST num scale_natoceni:=21;
```

```
PROC main()  
    inicializace;  
    souradnice_kostek;  
    kamera_barva;  
    FOR i FROM 1 TO pocet DO  
        barva_kostek(i);  
    ENDFOR  
    FOR i FROM 1 TO pocet DO  
        odber(i);  
    ENDFOR  
ENDPROC
```

```
PROC inicializace()  
    Hand_DoCalibrate;  
    Hand_GripOutward;  
    modra:=0;  
    zelena:=0;  
ENDPROC
```

```
PROC souradnice_kostek()
```

```

CamSetProgramMode kamera;

CamStartLoadJob kamera,"kostky";

CamWaitLoadJob kamera;

CamSetRunMode kamera;

MoveJ scan,v500,z50,tool_YuMiGripper_S_C_V1;

WaitRob\InPos;

CamReqImage kamera;

CamGetResult kamera,cameratarget1 {1};

pocet:=cameratarget1 {1}.val1;

IF pocet=0 THEN

    TPWrite "Kostky nenalezeny";

ELSE

    FOR i FROM 2 TO pocet DO

        CamGetResult kamera,cameratarget1 {i};

    ENDFOR

ENDIF

ENDPROC

PROC kamera_barva()

    CamSetProgramMode kamera;

    CamStartLoadJob kamera,"barvy";

    CamWaitLoadJob kamera;

    CamSetRunMode kamera;

```

```

    CamReqImage kamera;

    CamGetResult kamera,cameratarget2;

ENDPROC

PROC barva_kostek(num kostky_id)

    kostky{kostky_id,2}:=(cameratarget1 {kostky_id}.cframe.trans.x/scalex)*(-
1)+offsx;

    kostky{kostky_id,1}:=(cameratarget1 {kostky_id}.cframe.trans.y/scaley)*(-
1)+offsy;

    kostky{kostky_id,3}:=EulerZYX(\Z,cameratarget1 {kostky_id}.cframe.rot);

    MoveJ
Offs(nula,kostky{kostky_id,1},kostky{kostky_id,2},0),v200,z50,tool_YuMiGripper_S_
C_V1;

    WaitRob\InPos;

    CamReqImage kamera;

    CamGetResult kamera,cameratarget2;

    kostky{kostky_id,4}:=cameratarget2.val1;

ENDPROC

PROC odber(num kostky_id)

    MoveJ Offs(RelTool(nula10,0,0,0\Rz:=kostky{kostky_id,3}-
scale_natoceni),kostky{kostky_id,1}+7.85,kostky{kostky_id,2}-
39.19,0),v500,z50,tool_YuMiGripper_S_V1;

    WaitRob\InPos;

```

```
MoveL Offs(RelTool(nula10,0,0,0\Rz:=kostky{kostky_id,3}-
scale_natoceni),kostky{kostky_id,1}+7.85,kostky{kostky_id,2}-39.19,-
81.81),v200,fine,tool_YuMiGripper_S_V1;
```

```
Hand_GripInward;
```

```
MoveL Offs(RelTool(nula10,0,0,0\Rz:=kostky{kostky_id,3}-
scale_natoceni),kostky{kostky_id,1}+7.85,kostky{kostky_id,2}-
39.19,0),v200,fine,tool_YuMiGripper_S_V1;
```

```
IF kostky{kostky_id,4}<100 THEN
```

```
MoveJ modra10,v200,fine,tool_YuMiGripper_S_V1;
```

```
MoveL Offs(modra10,0,0,-
95.74+vyska_kostky*modra),v200,fine,tool_YuMiGripper_S_V1;
```

```
WaitRob\InPos;
```

```
Hand_GripOutward;
```

```
modra:=modra+1;
```

```
MoveL modra10,v200,z50,tool_YuMiGripper_S_V1;
```

```
ELSE
```

```
MoveJ zelena10,v200,fine,tool_YuMiGripper_S_V1;
```

```
MoveL Offs(zelena10,0,0,-
81.93+vyska_kostky*zelena),v200,fine,tool_YuMiGripper_S_V1;
```

```
WaitRob\InPos;
```

```
Hand_GripOutward;
```

```
zelena:=zelena+1;
```

```
MoveL zelena10,v200,z50,tool_YuMiGripper_S_V1;
```

```
ENDIF
```

ENDPROC
ENDMODULE

Příloha C

Levé rameno

MODULE MainModule

PERS wobjdata dynam:=
[FALSE,false,"ROB_R",[[0,0,0],[1,0,0,0]],[[0,0,0],[1,0,0,0]]];

PERS tasks task_list{2}:=["T_ROB_L"],["T_ROB_R"];

VAR syncident sync1;

VAR syncident sync2;

VAR syncident sync3;

VAR syncident sync4;

VAR syncident sync5;

VAR syncident sync6;

PROC main()

inicializace;

Premisteni;

WaitSyncTask sync3,task_list;

SpolecnyPohyb;

Hand_GripOutward;

MoveL [[414.70,165.57,84.52],[0.706503,0.707223,0.00857353,-
0.0248034],[-2,-3,0,5],[-145.074,9E+09,9E+09,9E+09,9E+09]], v500, z50,
tool_YuMiGripper_S_C_V1;

ENDPROC

PROC inicializace()

Hand_DoCalibrate;

Hand_GripOutward;

MoveJ [[298.45,173.11,222.13],[0.112711,0.991988,-0.0511144,-0.0253739],[-1,-3,1,4],[-144.708,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z50,tool_YuMiGripper_S_C_V1;

ENDPROC

PROC Premisteni()

MoveJ [[272.05,105.24,221.77],[0.0637859,0.997579,0.0271927,0.00539245],[-2,-3,0,4],[-116.897,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z50,tool_YuMiGripper_S_C_V1;

MoveL [[272.21,105.04,141.62],[0.0641541,0.997559,0.0269969,0.00569134],[-2,-3,0,4],[-117.044,9E+09,9E+09,9E+09,9E+09,9E+09]],v200,fine,tool_YuMiGripper_S_C_V1;

Hand_GripInward;

MoveL [[272.06,105.54,221.93],[0.0638005,0.997578,0.0271795,0.00538607],[-2,-3,0,4],[-116.93,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z50,tool_YuMiGripper_S_C_V1;

MoveJ [[413.73,-0.54,275.67],[0.0839266,0.99601,0.0299796,-0.00469113],[-2,-3,0,4],[-138.602,9E+09,9E+09,9E+09,9E+09,9E+09]], v500, z50, tool_YuMiGripper_S_C_V1;

MoveL [[413.60,-0.10,177.14],[0.0833775,0.996052,0.0300675,-0.00491723],[-2,-3,0,4],[-138.786,9E+09,9E+09,9E+09,9E+09,9E+09]], v200, fine, tool_YuMiGripper_S_C_V1;

Hand_GripOutward;

MoveL [[413.73,-0.54,275.67],[0.0839266,0.99601,0.0299796,-0.00469113],[-2,-3,0,4],[-138.602,9E+09,9E+09,9E+09,9E+09,9E+09]], v500, z50, tool_YuMiGripper_S_C_V1;

```
MoveJ [[298.45,173.11,222.13],[0.112711,0.991988,-0.0511144,-0.0253739],[-1,-  
3,1,4],[-  
144.708,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z50,tool_YuMiGripper_S_C_V1;
```

```
WaitSyncTask sync1,task_list;
```

```
WaitSyncTask sync2,task_list;
```

```
MoveJ [[414.70,165.57,84.52],[0.706503,0.707223,0.00857353,-0.0248034],[-2,-  
3,0,5],[-145.074,9E+09,9E+09,9E+09,9E+09,9E+09]], v500, z50,  
tool_YuMiGripper_S_C_V1;
```

```
MoveL [[414.76,139.15,78.55],[0.706392,0.705865,-0.0444187,0.0281568],[-2,-  
3,0,5],[-145.741,9E+09,9E+09,9E+09,9E+09,9E+09]], v500, fine,  
tool_YuMiGripper_S_C_V1;
```

```
Hand_GripInward;
```

```
ENDPROC
```

```
PROC SpolecnyPohyb()
```

```
SyncMoveOn sync4, task_list;
```

```
MoveL [[-5.98,8.72,290.61],[0.0272889,0.0804531,0.996367,-0.00593502],[-2,-  
3,0,5],[-145.776,9E+09,9E+09,9E+09,9E+09,9E+09]]\ID:=10, v300, fine,  
tool_YuMiGripper_S_C_V1\WObj:=dynam;
```

```
MoveL [[-5.98,8.72,290.61],[0.0272889,0.0804531,0.996367,-0.00593502],[-2,-  
3,0,5],[-145.776,9E+09,9E+09,9E+09,9E+09,9E+09]]\ID:=20, v300, fine,  
tool_YuMiGripper_S_C_V1\WObj:=dynam;
```

```
MoveL [[-5.98,8.72,290.61],[0.0272889,0.0804531,0.996367,-0.00593502],[-2,-  
3,0,5],[-145.776,9E+09,9E+09,9E+09,9E+09,9E+09]]\ID:=30, v300, fine,  
tool_YuMiGripper_S_C_V1\WObj:=dynam;
```

```
MoveL [[-5.98,8.72,290.61],[0.0272889,0.0804531,0.996367,-0.00593502],[-2,-3,0,5],[-145.776,9E+09,9E+09,9E+09,9E+09,9E+09]]\MD:=40, v300, fine,  
tool_YuMiGripper_S_C_V1\WObj:=dynam;
```

```
MoveL [[-5.98,8.72,290.61],[0.0272889,0.0804531,0.996367,-0.00593502],[-2,-3,0,5],[-145.776,9E+09,9E+09,9E+09,9E+09,9E+09]]\MD:=50, v300, fine,  
tool_YuMiGripper_S_C_V1\WObj:=dynam;
```

```
SyncMoveOff sync5;
```

```
ENDPROC
```

```
ENDMODULE
```

Pravé rameno

```
MODULE MainModule
```

```
PERS wobjdata wobR:= [FALSE,true,"",[[0,0,0],[1,0,0,0]],[[413,-147.90,52.25],[1,0,0,0]]];
```

```
PERS tasks task_list{2}:=["T_ROB_L"],["T_ROB_R"];
```

```
VAR syncident sync1;
```

```
VAR syncident sync2;
```

```
VAR syncident sync3;
```

```
VAR syncident sync4;
```

```
VAR syncident sync5;
```

```
VAR syncident sync6;
```

```
PROC main()
```

```
inicializace;
```

```
premistení;
```

```

WaitSyncTask sync3,task_list;

SpolecnyPohyb;

Hand_GripOutward;

    MoveI [[414.70,-186.29,84.52],[0.0321325,-0.0100786,-0.71174,-
0.701636],[1,-1,-1,4],[177.797,9E+09,9E+09,9E+09,9E+09,9E+09]], v500, fine,
tool_YuMiGripper_S_C_V1;

ENDPROC

PROC inicializace()

    Hand_DoCalibrate;

    Hand_GripOutward;

    MoveJ [[307.02,-388.88,207.35],[0.187463,-0.559903,-0.711624,-0.380732],[0,0,-
1,4],[157.008,9E+09,9E+09,9E+09,9E+09,9E+09]], v500, z50,
tool_YuMiGripper_S_C_V1;

ENDPROC

PROC premisteni()

    WaitSyncTask sync1,task_list;

    MoveJ [[279.18,-62.17,220.41],[0.0228832,0.000971335,-
0.999734,0.00270841],[0,0,-
1,1],[86.7473,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,z50,tool_YuMiGripper_S_C_
V1;

    MoveL [[280.60,-62.47,140.82],[0.019759,-0.00061257,-0.999794,0.00464251],[-
1,0,-
1,1],[86.2559,9E+09,9E+09,9E+09,9E+09,9E+09]],v200,fine,tool_YuMiGripper_S_C_
V1;

    Hand_GripInward;

```

MoveL [[279.22,-62.50,220.86],[0.0229025,0.000970398,-
0.999734,0.00272177],[0,0,-
1,1],[86.7146,9E+09,9E+09,9E+09,9E+09,9E+09]],v500,fine,tool_YuMiGripper_S_C_
V1;

MoveJ [[414.74,-12.29,241.86],[0.00171603,-0.0315768,-0.999482,-
0.00600237],[0,0,0,0],[81.1032,9E+09,9E+09,9E+09,9E+09,9E+09]], v500, z50,
tool_YuMiGripper_S_C_V1;

MoveL [[414.74,-12.29,211.04],[0.00172103,-0.0315705,-0.999482,-
0.00599703],[0,-1,0,0],[80.7834,9E+09,9E+09,9E+09,9E+09,9E+09]], v200, fine,
tool_YuMiGripper_S_C_V1;

Hand_GripOutward;

MoveL [[414.74,-12.29,241.86],[0.00171603,-0.0315768,-0.999482,-
0.00600237],[0,0,0,0],[81.1032,9E+09,9E+09,9E+09,9E+09,9E+09]], v500, z50,
tool_YuMiGripper_S_C_V1;

WaitSyncTask sync2,task_list;

MoveJ [[414.70,-186.29,84.52],[0.0321325,-0.0100786,-0.71174,-0.701636],[1,-1,-
1,4],[177.797,9E+09,9E+09,9E+09,9E+09,9E+09]], v500, fine,
tool_YuMiGripper_S_C_V1;

MoveL [[416.85,-150.87,73.37],[0.0321448,-0.0100801,-0.711742,-0.701633],[1,-
1,-1,4],[178.42,9E+09,9E+09,9E+09,9E+09,9E+09]], v500, fine,
tool_YuMiGripper_S_C_V1;

Hand_GripInward;

ENDPROC

PROC SpolecnyPohyb()

SyncMoveOn sync4, task_list;

MoveL [[416.85,-151.32,157.18],[0.0321548,-0.010079,-0.711745,-0.70163],[0,-
1,-1,4],[178.527,9E+09,9E+09,9E+09,9E+09,9E+09]]\id:=10, v300, fine,
tool_YuMiGripper_S_C_V1;

MoveL [[336.57,-212.73,157.18],[0.0321679,-0.0100817,-0.711734,-0.701639],[0,-1,-1,4],[178.695,9E+09,9E+09,9E+09,9E+09,9E+09]]MD:=20, v300, fine,
tool_YuMiGripper_S_C_V1;

MoveL [[336.57,-49.91,157.19],[0.0321652,-0.0100853,-0.711738,-0.701636],[0,-1,-1,4],[178.732,9E+09,9E+09,9E+09,9E+09,9E+09]]MD:=30, v300, fine,
tool_YuMiGripper_S_C_V1;

MoveL [[416.85,-151.32,157.18],[0.0321548,-0.010079,-0.711745,-0.70163],[0,-1,-1,4],[178.527,9E+09,9E+09,9E+09,9E+09,9E+09]]id:=40, v300, fine,
tool_YuMiGripper_S_C_V1;

MoveL [[416.85,-150.87,73.37],[0.0321448,-0.0100801,-0.711742,-0.701633],[1,-1,-1,4],[178.42,9E+09,9E+09,9E+09,9E+09,9E+09]]id:=50, v300, fine,
tool_YuMiGripper_S_C_V1;

SyncMoveOff sync5;

ENDPROC

PROC Routine1()

MoveJ [[414.74,-12.29,241.86],[0.00171603,-0.0315768,-0.999482,-0.00600237],[0,0,0,0],[81.1032,9E+09,9E+09,9E+09,9E+09,9E+09]], v1000, z50,
tool_YuMiGripper_S_C_V1;

MoveL [[414.74,-12.29,211.04],[0.00172103,-0.0315705,-0.999482,-0.00599703],[0,-1,0,0],[80.7834,9E+09,9E+09,9E+09,9E+09,9E+09]], v1000, z50,
tool_YuMiGripper_S_C_V1;

MoveJ [[417.71,-186.29,68.92],[0.0321325,-0.0100786,-0.71174,-0.701636],[1,-1,-1,4],[177.797,9E+09,9E+09,9E+09,9E+09,9E+09]], v1000, z50,
tool_YuMiGripper_S_C_V1;

MoveL [[416.85,-150.87,73.37],[0.0321448,-0.0100801,-0.711742,-0.701633],[1,-1,-1,4],[178.42,9E+09,9E+09,9E+09,9E+09,9E+09]], v1000, z50,
tool_YuMiGripper_S_C_V1;

ENDPROC

PROC Routine2()

MoveL [[416.85,-151.32,157.18],[0.0321548,-0.010079,-0.711745,-
0.70163],[0,-1,-1,4],[178.527,9E+09,9E+09,9E+09,9E+09,9E+09]], v1000, z50,
tool_YuMiGripper_S_C_V1;

MoveL [[336.57,-212.73,157.18],[0.0321679,-0.0100817,-0.711734,-
0.701639],[0,-1,-1,4],[178.695,9E+09,9E+09,9E+09,9E+09,9E+09]], v1000, z50,
tool_YuMiGripper_S_C_V1;

MoveL [[336.57,-49.91,157.19],[0.0321652,-0.0100853,-0.711738,-
0.701636],[0,-1,-1,4],[178.732,9E+09,9E+09,9E+09,9E+09,9E+09]], v1000, z50,
tool_YuMiGripper_S_C_V1;

ENDPROC

ENDMODULE

Příloha E

Pravé rameno:

MODULE MainModule

```
TASK PERS wobjdata drevo:=[FALSE,TRUE,"",[[409.698,-  
176.834,114.441],[0.707071,0.00110076,0.00139579,0.70714]],[[0,0,0],[1,0,0,0]]];
```

```
PERS wobjdata  
papir:=[FALSE,TRUE,"",[[459.305,164.726,89.62],[0.997464,0,0,0.0711668]],[[0,0,0],  
[1,0,0,0]]];
```

```
CONST jointtarget JointTarget_2:=[[0,-130,30,0,40,0],[-  
135,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
CONST robtarget pred_zelenou:=[[-41.54,114.94,-  
38.42],[0.716729,0.695903,0.0256958,0.0368519],[0,0,-1,4],[-  
101.046,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
CONST robtarget zelena:=[[-40.54,76.98,-  
38.28],[0.716731,0.695901,0.0256969,0.0368548],[0,0,-1,4],[-  
100.862,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
CONST robtarget nad_zelenou:=[[-  
40.55,77.16,13.56],[0.716728,0.695904,0.0256958,0.0368677],[0,0,-1,4],[-  
102.15,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
CONST robtarget sundani_vicka:=[[418.91,1.75,283.86],[0.580527,-  
0.403219,0.588454,0.392586],[1,-1,-1,5],[-  
174.736,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
CONST robtarget sundani_vicka10:=[[475.69,-21.10,284.26],[0.580496,-  
0.403239,0.588471,0.392587],[1,-2,0,5],[-  
174.547,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
CONST robtarget doma:=[[-9.00,-180.57,200.53],[0.0671613,-0.841008,-0.108406,-  
0.525779],[0,0,0,4],[-102.018,9E+09,9E+09,9E+09,9E+09,9E+09]];
```


CONST robtarget poloha_1:=[[308.49,32.91,89.24],[0.648315,-0.751212,-
0.0941714,0.0806227],[1,-2,-1,5],[158.285,9E+09,9E+09,9E+09,9E+09,9E+09]];

!CONST robtarget poloha_kamera_0:=[[416.16,61.85,89.62],[0.64831,-0.751217,-
0.0941649,0.0806269],[1,-2,-1,5],[158.291,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget poloha_kamera_0:=[[0.07,-0.40,0.38],[0.648304,-0.751222,-
0.0941568,0.0806328],[1,-2,-1,5],[158.297,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget pruchozi_bod:=[[532.37,-317.76,327.22],[0.251351,-0.820207,-
0.101034,-0.50386],[0,-3,-1,5],[-179.57,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget prechod:=[[355.69,-135.15,225.83],[0.680292,-
0.722843,0.0851922,-0.0862772],[0,-2,-1,5],[-
177.484,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak_nahore:=[[130.222,-230.927,10.7595],[0.648221,-0.751301,-
0.094116,0.0806141],[0,-2,-1,5],[158.333,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget znak1:=[[130.07,-231.69,-2],[0.648287,-0.751238,-
0.0941534,0.0806312],[0,-2,-1,5],[158.654,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget znak2:=[[100.492,-226.273,-2],[0.648343,-0.751182,-
0.0942056,0.0806353],[1,-2,-1,5],[158.697,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak3:=[[90.0089,-234.224,-2],[0.648336,-0.751184,-
0.0942495,0.0806236],[1,-2,-1,5],[158.738,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak4:=[[82.9436,-245.411,-2],[0.648319,-0.751198,-
0.0942481,0.0806345],[1,-2,-1,5],[158.782,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak5:=[[68.5524,-255.595,-2],[0.648314,-0.751199,-
0.0942665,0.0806362],[1,-2,-1,5],[158.83,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak6:=[[77.9602,-241.486,-2],[0.648267,-0.751244,-
0.0942089,0.0806648],[1,-2,-1,5],[158.873,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak7:=[[86.2605,-235.382,-2],[0.648255,-0.751257,-
0.0941884,0.0806674],[1,-2,-1,5],[158.915,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak8:=[[-92.4063,-227.175,-2],[0.648268,-0.751249,-0.0941601,0.0806637],[1,-2,-1,5],[158.959,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak9:=[[-81.9004,-225.506,-2],[0.648291,-0.751226,-0.0941824,0.0806728],[1,-2,-1,5],[159.002,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak10:=[[-75.4349,-232.327,-2],[0.648283,-0.751231,-0.0942011,0.0806661],[1,-2,-1,5],[159.047,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak11:=[[-75.4966,-237.495,-2],[0.648224,-0.751283,-0.0941862,0.080676],[1,-2,-1,5],[159.089,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak12:=[[-65.7743,-237.279,-2],[0.64827,-0.75124,-0.0942819,0.0805933],[1,-2,-1,5],[159.139,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak13:=[[-72.9252,-231.388,-2],[0.648239,-0.751268,-0.0942706,0.0805941],[1,-2,-1,5],[159.183,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak14:=[[-75.6235,-229.647,-2],[0.648218,-0.751284,-0.0942784,0.0806028],[1,-2,-1,5],[159.227,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak15:=[[-77.5612,-227.177,-2],[0.648178,-0.751322,-0.094218,0.0806397],[1,-2,-1,5],[159.269,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak16:=[[-65.2569,-215.617,-2],[0.648236,-0.751275,-0.0942361,0.0805952],[1,-2,-1,5],[159.318,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak17:=[[-79.0308,-203.867,-2],[0.648214,-0.751298,-0.0941805,0.080619],[1,-2,-1,5],[159.358,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak18:=[[-133.439,-214.993,-2],[0.648158,-0.75135,-0.0940776,0.0807108],[0,-2,-1,5],[159.389,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak19:=[[-142.515,-229.389,-2],[0.648125,-0.751374,-0.0940018,0.0808401],[0,-2,-1,5],[159.419,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak20:=[[-128.873,-239.336,-2],[0.64815,-0.751352,-0.0939903,0.0808604],[0,-2,-1,5],[159.458,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak_mezi_1:=[[-79.7237,-256.591,-2],[0.648305,-0.751211,-0.0942401,0.0806396],[1,-2,-1,5],[158.829,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak_mezi_2:=[[-67.2739,-244.4,-2],[0.648339,-0.751178,-0.0942843,0.080613],[1,-2,-1,5],[158.828,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak_mezi_3:=[[-73.9478,-239.308,-2],[0.648287,-0.751223,-0.0942272,0.0806784],[1,-2,-1,5],[159.091,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS robtarget znak_mezi_4:=[[-66.5881,-232.119,-2],[0.648366,-0.751157,-0.0942291,0.0806552],[1,-2,-1,5],[159.096,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget k_1:=[[-106.422,-155.5089,-2],[0.648144,-0.751363,-0.0939801,0.0808194],[1,-2,-1,5],[158.392,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget k_2:=[[-80.3142,-155.9709,-2],[0.648108,-0.751387,-0.093863,0.0810182],[1,-2,-1,5],[158.422,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget k_3:=[[-87.6647,-155.9709,-2],[0.648947,-0.750629,-0.0941589,0.0809836],[1,-2,-1,5],[158.515,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget k_4:=[[-106.422,-144.743,-2],[0.64885,-0.750724,-0.094184,0.0808523],[1,-2,-1,5],[158.55,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget k_5:=[[-93.7532,-151.9889,-2],[0.648957,-0.750624,-0.0941786,0.0809322],[1,-2,-1,5],[158.615,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget k_6:=[[-80.3142,-144.743,-2],[0.648823,-0.750734,-0.0940903,0.0810855],[1,-2,-1,5],[158.633,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget k_nahore_1:=[[-106.422,-160.873,11],[0.648061,-0.751433,-0.0939898,0.0808138],[1,-2,-1,5],[158.353,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget k_nahore_2:=[[-87.6647,-155.9709,11],[0.648929,-0.750648,-0.0941585,0.0809486],[1,-2,-1,5],[158.582,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget k_nahore_3:=[[-93.7532,-151.9889,11],[0.648929,-0.750648,-0.0941585,0.0809486],[1,-2,-1,5],[158.582,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget k_nahore_4:=[[-80.3142,-144.743,11],[0.648823,-0.750734,-0.0940903,0.0810855],[1,-2,-1,5],[158.633,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget a_1:=[[-80.1585,-137.876,-2],[0.648757,-0.750802,-
0.0940053,0.0810786],[1,-2,-1,5],[158.702,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget a_2:=[[-106.925,-130.569,-2],[0.648738,-0.750823,-
0.0940729,0.0809602],[1,-2,-1,5],[158.737,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget a_3:=[[-80.1481,-123.379,-2],[0.648687,-0.750852,-
0.0939034,0.081295],[1,-2,-1,5],[158.755,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget a_4:=[[-87.6024,-136.906,-2],[0.648783,-0.750768,-
0.0940182,0.0811706],[1,-2,-1,5],[158.834,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget a_5:=[[-87.5707,-127.197,-2],[0.648621,-0.750905,-
0.0939858,0.0812354],[1,-2,-1,5],[158.856,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget a_nahore_1:=[[-80.2984,-137.51,11],[0.648715,-0.750834,-
0.0939885,0.0811438],[1,-2,-1,5],[158.664,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget a_nahore_2:=[[-87.7408,-136.617,11],[0.648721,-0.75082,-
0.0939448,0.0812725],[1,-2,-1,5],[158.791,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget a_nahore_3:=[[-87.5635,-127.709,11],[0.648555,-0.750953,-
0.0939547,0.0813499],[1,-2,-1,5],[158.879,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget t_1:=[[-106.422,-118.266,-2],[0.648498,-0.75103,-
0.0940192,0.0810217],[1,-2,-1,5],[158.953,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget t_2:=[[-106.431,-103.398,-2],[0.648333,-0.751168,-
0.0939439,0.0811507],[1,-2,-1,5],[158.968,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget t_3:=[[-106.108,-111.882,-2],[0.648347,-0.751162,-
0.0940409,0.0809831],[1,-2,-1,5],[159.036,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget t_4:=[[-79.2342,-112.345,-2],[0.648289,-0.751201,-
0.0939637,0.0811788],[1,-2,-1,5],[159.056,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget tri_1:=[[-105.949,-82.7007,-2],[0.648212,-0.75129,-
0.0938545,0.0810915],[1,-2,-1,5],[159.154,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget tri_2:=[[-105.936,-75.5639,-2],[0.648081,-0.751401,-
0.0937893,0.0811884],[1,-2,-1,5],[159.172,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget tri_3:=[[-102.213,-72.9146,-2],[0.647912,-0.751539,-
0.0936386,0.0814272],[1,-2,-1,5],[159.183,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget tri_4:=[[-96.6314,-73.3876,-2],[0.647848,-0.751577,-
0.093515,0.0817325],[1,-2,-1,5],[159.193,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget tri_5:=[[-94.6633,-76.8448,-2],[0.64793,-0.7515,-
0.0935442,0.0817515],[1,-2,-1,5],[159.226,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget tri_6:=[[-94.5148,-80.9582,-2],[0.648025,-0.75143,-
0.0935745,0.0816121],[1,-2,-1,5],[159.274,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget tri_7:=[[-90.6914,-73.7563,-2],[0.647934,-0.751492,-
0.0935623,0.0817833],[1,-2,-1,5],[159.286,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget tri_8:=[[-81.2204,-74.2075,-2],[0.647888,-0.751519,-
0.0934358,0.0820364],[1,-2,-1,5],[159.304,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget tri_9:=[[-79.3454,-76.253,-2],[0.648108,-0.751306,-
0.0935123,0.0821588],[1,-2,-1,5],[159.339,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget tri_10:=[[-79.26,-86.1353,-2],[0.648202,-0.751224,-
0.0935392,0.0821402],[1,-2,-1,5],[159.38,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget pet_1:=[[-105.898,-56.5432,-2],[0.648086,-0.751338,-
0.0935562,0.0819887],[1,-2,-1,5],[159.474,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget pet_2:=[[-105.787,-68.3345,-2],[0.648231,-0.751215,-
0.0935934,0.0819346],[1,-2,-1,5],[159.517,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget pet_3:=[[-94.3595,-68.7951,-2],[0.648185,-0.751241,-
0.0934906,0.082172],[1,-2,-1,5],[159.535,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget pet_4:=[[-94.3046,-61.622,-2],[0.648054,-0.75135,-
0.0934686,0.0822362],[1,-2,-1,5],[159.552,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget pet_5:=[[-90.5389,-58.3067,-2],[0.647896,-0.751477,-
0.0933513,0.0824488],[1,-2,-1,5],[159.563,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget pet_6:=[[-83.0612,-58.7323,-2],[0.647928,-0.751427,-
0.093233,0.0827918],[1,-2,-1,5],[159.579,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget pet_7:=[[-79.2667,-62.8199,-2],[0.648,-0.751356,-
0.0932066,0.0829014],[1,-2,-1,5],[159.609,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget pet_8:=[[-79.1947,-70.8163,-2],[0.648047,-0.751316,-
0.093203,0.0829018],[1,-2,-1,5],[159.646,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget ctyri_1:=[[-105.876,-47.0139,-2],[0.647929,-0.751432,-
0.0931748,0.0828033],[1,-2,-1,5],[159.738,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget ctyri_2:=[[-84.8425,-53.1978,-2],[0.647912,-0.751435,-
0.0930409,0.0830658],[1,-2,-1,5],[159.758,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget ctyri_3:=[[-84.8175,-40.2289,-2],[0.647797,-0.751528,-
0.092998,0.0831664],[1,-2,-1,5],[159.774,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget ctyri_4:=[[-92.26,-45.4579,-2],[0.647862,-0.751481,-
0.0930645,0.0830065],[1,-2,-1,5],[159.846,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget ctyri_5:=[[-78.8885,-45.9129,-2],[0.6478,-0.751523,-
0.0929424,0.0832495],[1,-2,-1,5],[159.864,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget t_nahore_1:=[[-106.517,-117.852,11],[0.648492,-0.751029,-
0.0940193,0.0810832],[1,-2,-1,5],[158.92,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget t_nahore_2:=[[-106.194,-111.477,11],[0.648333,-0.751175,-
0.0940275,0.0809914],[1,-2,-1,5],[159.006,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget t_nahore_3:=[[-79.3174,-97.3654,11],[0.648143,-0.75134,-
0.0937274,0.0813247],[1,-2,-1,5],[159.083,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget tri_nahore_1:=[[-106.069,-82.3596,11],[0.648125,-0.751368,-0.0937841,0.0811519],[1,-2,-1,5],[159.12,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget tri_nahore_2:=[[-79.2378,-86.5347,11],[0.648146,-0.751274,-0.0934917,0.0821738],[1,-2,-1,5],[159.412,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget pet_nahore_1:=[[-106.026,-56.2065,11],[0.648032,-0.751382,-0.0935202,0.0820615],[1,-2,-1,5],[159.44,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget pet_nahore_2:=[[-79.187,-71.239,11],[0.647973,-0.751386,-0.0930994,0.0829653],[1,-2,-1,5],[159.678,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget ctyri_nahore_1:=[[-105.989,-46.6545,11],[0.647891,-0.751463,-0.0931375,0.0828603],[1,-2,-1,5],[159.707,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget ctyri_nahore_2:=[[-92.4329,-45.0822,11],[0.6478,-0.751527,-0.0930102,0.0831363],[1,-2,-1,5],[159.808,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget ctyri_nahore_3:=[[-78.8475,-46.315,11],[0.647684,-0.751629,-0.0928679,0.0832719],[1,-2,-1,5],[159.891,9E+09,9E+09,9E+09,9E+09,9E+09]];

var robtarget znak_nahore_kruh:=[[-130.1,-204.327,11],[0.648326,-0.75121,-0.0942254,0.0804952],[1,-2,-1,5],[158.376,9E+09,9E+09,9E+09,9E+09,9E+09]];

var robtarget znak_nahore_kruh_m:=[[-130.161,-181.943,11],[0.648136,-0.75137,-0.0940964,0.0806754],[1,-2,-1,5],[158.397,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget kruh1:=[[-129.983,-204.757,-2],[0.64835,-0.751183,-0.0942766,0.0804837],[1,-2,-1,5],[158.406,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget kruh2:=[[-143.098,-191.924,-2],[0.648225,-0.751297,-0.0943321,0.0803652],[1,-2,-1,5],[158.437,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget kruh3:=[[-129.751,-179.035,-2],[0.64818,-0.751337,-0.094255,0.0804447],[1,-2,-1,5],[158.466,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR robtarget kruh4:=[[-116.485,-192.592,-2],[0.648297,-0.751227,-0.0942272,0.0805646],[1,-2,-1,5],[158.503,9E+09,9E+09,9E+09,9E+09,9E+09]];

```
VAR robtarget kruh5:=[[-129.662,-206.251,-2],[0.64835,-0.751182,-  
0.0943066,0.0804639],[1,-2,-1,5],[158.544,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
VAR robtarget kruh1_m:=[[-130.114,-182.315,-2],[0.648201,-0.751312,-  
0.0940863,0.0807104],[1,-2,-1,5],[158.43,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
VAR robtarget kruh2_m:=[[-139.482,-192.165,-2],[0.648318,-0.751211,-  
0.0942647,0.0804946],[1,-2,-1,5],[158.472,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
VAR robtarget kruh3_m:=[[-129.918,-202.078,-2],[0.648395,-0.75114,-  
0.0942197,0.0805917],[1,-2,-1,5],[158.509,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
VAR robtarget kruh4_m:=[[-120.425,-193.022,-2],[0.648277,-0.75124,-  
0.0940293,0.0808334],[1,-2,-1,5],[158.533,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
VAR robtarget kruh5_m:=[[-129.851,-183.908,-2],[0.64817,-0.751335,-  
0.0940728,0.0807534],[1,-2,-1,5],[158.564,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

```
PERS tasks task_list{2}:=["T_ROB_R"],["T_ROB_L"];
```

```
VAR syncident sync1;
```

```
VAR syncident sync2;
```

```
VAR syncident sync3;
```

```
VAR syncident sync4;
```

```
VAR syncident sync5;
```

```
VAR syncident sync6;
```

```
VAR syncident sync7;
```

```
PROC main()
```



```

!MoveAbsJ JointTarget_2,v300,fine,tool_YuMiGripper_S_C_V1;

!inicializace;

!MoveJ pruchozi_bod,v300,z50,tool_YuMiGripper_S_C_V1;

MoveJ doma,v300,z50,tool_YuMiGripper_S_C_V1;

WaitSyncTask sync3,task_list;

WaitSyncTask sync4,task_list;

chyceni_zelene_fixy;

sundani_vicka_fce;

!znak_vytvoreni_cesty;

Znak_1;

Znak_kruh;

Znak_kruh_m;

!MoveL znak1,v100,fine,tool_YuMiGripper_S_C_V1\WObj:=papir;

napis;

!napis_souradnice_t;

zavreni_vicka;

odlozeni_fixy;

MoveJ doma,v300,z50,tool_YuMiGripper_S_C_V1;

ENDPROC

PROC odlozeni_fixy()

MoveJ nad_zelenou,v300,fine,tool_YuMiGripper_S_C_V1\WObj:=drevo;

```

```
MoveL zelena,v100,fine,tool_YuMiGripper_S_C_V1\WObj:=drevo;  
  
WaitRob\InPos;  
  
Hand_MoveTo 7.5;  
  
MoveL pred_zelenou,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=drevo;  
  
ENDPROC
```

```
PROC zavreni_vicka()  
  
MoveJ sundani_vicka10,v300,fine,tool_YuMiGripper_S_C_V1;  
  
WaitSyncTask sync5,task_list;  
  
MoveL sundani_vicka,v200,fine,tool_YuMiGripper_S_C_V1;  
  
WaitSyncTask sync6,task_list;  
  
WaitSyncTask sync7,task_list;  
  
ENDPROC
```

```
PROC Znak_kruh()  
  
MoveJ znak_nahore_kruh,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;  
  
MoveL kruh1,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;  
  
MoveC kruh2,kruh3,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;  
  
MoveC kruh4,kruh1,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;  
  
MoveL znak_nahore_kruh,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;  
  
ENDPROC
```

```
PROC Znak_kruh_m()  
  
MoveJ znak_nahore_kruh_m,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```

MoveL kruh1_m,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveC kruh2_m,kruh3_m,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveC kruh4_m,kruh1_m,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL
znak_nahore_kruh_m,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;

ENDPROC

PROC kruh_m()

MoveL
offs(znak_nahore_kruh,0,22.98897289,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj
:=papier;

znak_nahore_kruh_m:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL
offs(znak_nahore_kruh_m,0,0,z1),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

kruh1_m:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL offs(kruh1_m,-9.646637779,-
9.64663778,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

kruh2_m:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL offs(kruh2_m,9.646637779,-
9.64663778,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

kruh3_m:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

```

```
MoveL  
offs(kruh3_m,9.64663778,9.64663778,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:  
=papier;
```

```
kruh4_m:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
MoveL offs(kruh4_m,-  
9.64663778,9.64663778,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
kruh5_m:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
ENDPROC
```

```
PROC kruh()
```

```
MoveL offs(znak_nahore,-  
0.05569778,26.85928684,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
znak_nahore_kruh:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
MoveL  
offs(znak_nahore_kruh,0,0,z1),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
kruh1:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
MoveL offs(kruh1,-  
13.34233511,13.34233511,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
kruh2:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
MoveL  
offs(kruh2,13.34233511,13.34233511,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:  
=papier;
```

```
kruh3:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
MoveL offs(kruh3,13.34233511,-  
13.34233511,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
kruh4:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
MoveL offs(kruh4,-13.34233511,-  
13.34233511,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
kruh5:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
ENDPROC
```

```
PROC napis_souradnice_4()
```

```
moveL offs(pet_nahore_2,-  
27,25.072,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
ctyri_nahore_1:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
MoveL  
offs(ctyri_nahore_1,0,0,z1),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
ctyri_1:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
MoveL offs(ctyri_1,21.214,-  
5.786,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
ctyri_2:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
MoveL
```

```
offs(ctyri_2,0,13.5,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
ctyri_3:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
MoveL offs(ctyri_3,-7.714,-
```

```
4.5,z2),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
ctyri_nahore_2:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
MoveL
```

```
offs(ctyri_nahore_2,0,0,z1),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
ctyri_4:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
MoveL
```

```
offs(ctyri_4,13.5,0,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
ctyri_5:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
MoveL offs(ctyri_5,0,0,z2),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
ctyri_nahore_3:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
ENDPROC
```

```
PROC napis_souradnice_5()
```

```
MoveL offs(tri_nahore_2,-
```

```
27,30.857,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

pet_nahore_1:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL

offs(pet_nahore_1,0,0,z1),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

pet_1:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL offs(pet_1,0,-

11.5714,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

pet_2:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL

offs(pet_2,11.5714,0,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

pet_3:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL

offs(pet_3,0,7.71429,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

pet_4:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL

offs(pet_4,3.857,3.857,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

pet_5:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL

offs(pet_5,7.71429,0,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

pet_6:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

```

MoveL offs(pet_6,3.857,-
3.857,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

pet_7:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL offs(pet_7,0,-
7.71429,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

pet_8:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL offs(pet_8,0,0,z2),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

pet_nahore_2:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

ENDPROC

```

```

PROC napis_souradnice_3()

```

```

MoveL offs(t_nahore_3,-
27,15.429,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

tri_nahore_1:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

```

```

MoveL
offs(tri_nahore_1,0,0,z1),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

tri_1:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

```

```

MoveL
offs(tri_1,0,7.71429,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

tri_2:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

```


MoveL

offs(tri_2,3.857,3.214,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

tri_3:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL

offs(tri_3,5.78571,0,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

tri_4:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL offs(tri_4,1.928,-

3.214,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

tri_5:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL offs(tri_5,0,-

3.85715,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

tri_6:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL

offs(tri_6,3.857,7.714,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

tri_7:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL

offs(tri_7,9.64286,0,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

tri_8:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL offs(tri_8,1.929,-

1.929,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

```
tri_9:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
MoveL offs(tri_9,0,-  
9.64286,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
tri_10:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
MoveL offs(tri_10,0,0,z2),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
tri_nahore_2:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
ENDPROC
```

```
PROC napis_souradnice_t()
```

```
!MoveL offs(a_nahore_3,-  
19.286,10.286,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
!t_nahore_1 := CRobT(\Tool:=tool_YuMiGripper_S_C_V1 \WObj:=papier);
```

```
!MoveL  
offs(t_nahore_1,0,0,z1),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
!t_1 := CRobT(\Tool:=tool_YuMiGripper_S_C_V1 \WObj:=papier);
```

```
! MoveL  
offs(t_1,0,15.4286,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
!t_2 := CRobT(\Tool:=tool_YuMiGripper_S_C_V1 \WObj:=papier);
```

```
MoveL offs(t_2,0,-  
7.71428,13),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

t_nahore_2:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL offs(t_nahore_2,0,0,-
13),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

t_3:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

MoveL offs(t_3,27,0,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

t_4:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);

!MoveL offs(t_4,0,0,z2),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

!t_nahore_3 := CRobT(\Tool:=tool_YuMiGripper_S_C_V1 \WObj:=papier);

ENDPROC

PROC napis()

!pismo K

MoveJ k_nahore_1,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL k_1,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL k_2,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveJ k_nahore_2,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL k_3,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL k_4,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveJ k_nahore_3,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL k_5,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL k_6,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

!písmeno A

MoveJ a_nahore_1,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL a_1,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL a_2,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL a_3,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveJ a_nahore_2,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL a_4,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL a_5,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

!písmeno T

MoveJ t_nahore_1,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL t_1,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL t_2,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveJ t_nahore_2,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL t_3,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL t_4,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

!číslo 3

MoveJ tri_nahore_1,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL tri_1,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL tri_2,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL tri_3,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL tri_4,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL tri_5,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL tri_6,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL tri_5,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL tri_7,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL tri_8,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL tri_9,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL tri_10,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

!cislo 5

MoveJ pet_nahore_1,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL pet_1,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL pet_2,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL pet_3,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL pet_4,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL pet_5,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL pet_6,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL pet_7,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL pet_8,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

!cislo 4

MoveJ ctyri_nahore_1,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL ctyri_1,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL ctyri_2,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL ctyri_3,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

```
MoveJ ctyri_nahore_2,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;  
MoveL ctyri_4,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;  
MoveL ctyri_5,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;  
MoveL ctyri_nahore_3,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;  
ENDPROC
```

```
PROC napis_souradnice_a()
```

```
MoveL  
offs(k_nahore_4,0,7.71429,0),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;  
a_nahore_1:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);  
  
movel offs(a_3,-7.714,-  
12.857,z2),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;  
a_nahore_2:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);
```

```
MoveL offs(a_5,0,0,z2),v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;  
a_nahore_3:=CRobT(\Tool:=tool_YuMiGripper_S_C_V1\WObj:=papier);  
ENDPROC
```

```
PROC cesta_a()
```

```
MoveJ a_nahore_1,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;  
MoveL a_1,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```

MoveL a_2,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL a_3,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveJ a_nahore_2,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL a_4,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL a_5,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
ENDPROC

```

```

PROC sundani_vicka_fce()

```

```

MoveJ sundani_vicka,v300,fine,tool_YuMiGripper_S_C_V1;
WaitSyncTask sync1,task_list;
WaitSyncTask sync2,task_list;
MoveL sundani_vicka10,v200,fine,tool_YuMiGripper_S_C_V1;
ENDPROC

```

```

PROC Znak_1()

```

```

MoveJ znak_nahore,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL znak1,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL znak2,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveC znak3,znak4,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveC znak_mezi_1,znak5,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveC znak_mezi_2,znak6,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveC znak7,znak8,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL znak9,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;
MoveL znak10,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

```

MoveC znak_mezi_3,znak12,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveC znak_mezi_4,znak13,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL znak14,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL znak15,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveC znak16,znak17,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL znak18,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveC znak19,znak1,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=papier;

MoveL znak_nahore,v200,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;

ENDPROC

PROC inicializace()

Hand_DoCalibrate\Jog;

ENDPROC

PROC chyceni_zelene_fixy()

Hand_MoveTo 7.5;

!WaitTime 5;

MoveJ pred_zelenou,v300,z50,tool_YuMiGripper_S_C_V1\WObj:=drevo;

MoveL zelena,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=drevo;

Hand_GripInward;

WaitRob\InPos;

MoveL nad_zelenou,v200,fine,tool_YuMiGripper_S_C_V1\WObj:=drevo;

ENDPROC

PROC Routine1()

MoveJ [[-41.54,114.94,-38.42],[0.716729,0.695903,0.0256958,0.0368519],[0,0,-1,4],[-101.046,9E+09,9E+09,9E+09,9E+09,9E+09]],v1000,z50,tool_YuMiGripper_S_C_V1\
WObj:=drevo;

MoveL [[-40.54,76.98,-38.28],[0.716731,0.695901,0.0256969,0.0368548],[0,0,-1,4],[-100.862,9E+09,9E+09,9E+09,9E+09,9E+09]],v1000,z50,tool_YuMiGripper_S_C_V1\
WObj:=drevo;

MoveL [[-40.55,77.16,13.56],[0.716728,0.695904,0.0256958,0.0368677],[0,0,-1,4],[-102.15,9E+09,9E+09,9E+09,9E+09,9E+09]],v1000,z50,tool_YuMiGripper_S_C_V1\
WObj:=drevo;

MoveJ [[-9.00,-180.57,200.53],[0.0671619,-0.841008,-0.108405,-0.525779],[0,0,0,4],[-102.018,9E+09,9E+09,9E+09,9E+09,9E+09]],v1000,z50,tool_YuMiGripper_S_C_V1;

MoveJ [[308.49,32.91,89.24],[0.648315,-0.751212,-0.0941714,0.0806227],[1,-2,-1,5],[158.285,9E+09,9E+09,9E+09,9E+09,9E+09]],v1000,z50,tool_YuMiGripper_S_C_V1;

MoveJ [[416.16,61.85,89.62],[0.64831,-0.751217,-0.0941649,0.0806269],[1,-2,-1,5],[158.291,9E+09,9E+09,9E+09,9E+09,9E+09]],v1000,z50,tool_YuMiGripper_S_C_V1;

MoveJ [[0.07,-0.40,0.38],[0.648304,-0.751222,-0.0941568,0.0806328],[1,-2,-1,5],[158.297,9E+09,9E+09,9E+09,9E+09,9E+09]],v1000,z50,tool_YuMiGripper_S_C_V1\
WObj:=papier;

MoveJ [[-133.50,-237.13,88.42],[0.693597,-0.711235,0.114073,0.00741721],[0,-2,-1,5],[176.62,9E+09,9E+09,9E+09,9E+09,9E+09]],v1000,z50,tool_YuMiGripper_S_C_V1\
WObj:=papier;

```
MoveJ [[355.69,-135.15,225.83],[0.680292,-0.722843,0.0851922,-0.0862772],[0,-2,-1,5],[-177.484,9E+09,9E+09,9E+09,9E+09,9E+09]],v1000,z50,tool_YuMiGripper_S_C_V1;
```

```
MoveL [[-130.07,-231.69,-3.25],[0.648287,-0.751238,-0.0941534,0.0806312],[0,-2,-1,5],[158.654,9E+09,9E+09,9E+09,9E+09,9E+09]],v1000,z50,tool_YuMiGripper_S_C_V1\WObj:=papier;
```

```
MoveJ [[532.37,-317.76,327.22],[0.251351,-0.820207,-0.101034,-0.50386],[0,-3,-1,5],[-179.57,9E+09,9E+09,9E+09,9E+09,9E+09]],v1000,z50,tool_YuMiGripper_S_C_V1;
```

```
MoveJ [[298.15,69.91,77.08],[0.695273,-0.718188,-0.0212816,-0.0186567],[1,-2,-2,5],[168.336,9E+09,9E+09,9E+09,9E+09,9E+09]],v1000,z50,tool_YuMiGripper_S_C_V1;
```

```
MoveL [[440.36,81.31,89.62],[0.64831,-0.751217,-0.0941646,0.0806253],[1,-2,-1,5],[158.279,9E+09,9E+09,9E+09,9E+09,9E+09]],v1000,z50,tool_YuMiGripper_S_C_V1;
```

```
MoveJ [[-9.00,-180.57,200.53],[0.0671613,-0.841008,-0.108406,-0.525779],[0,0,0,4],[-102.018,9E+09,9E+09,9E+09,9E+09,9E+09]],v1000,z50,tool_YuMiGripper_S_C_V1;
```

```
ENDPROC
```

```
ENDMODULE
```

Levé rameno:

```
MODULE MainModule
```

```
CONST robtarget doma_L:=[[-6.32,181.53,198.13],[0.0688427,0.844408,-0.111914,0.519337],[-1,0,0,4],[102.21,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

CONST robtarget Pred_vickem10:=[[363.87,399.63,445.67],[0.152829,-
0.450079,0.763075,-0.437937],[-1,-1,0,4],[-
139.892,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Pred_vickem20:=[[337.25,290.85,270.22],[0.361018,0.613822,-
0.347324,0.610127],[-1,-1,-1,4],[-147.482,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Foto:=[[304.27,109.49,386.14],[0.0159576,0.0112773,0.743878,-
0.66803],[-1,0,0,4],[-174.506,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget chyceni_vicka:=[[360.67,260.30,279.09],[0.376272,0.599524,-
0.412886,0.573162],[-1,-1,-1,4],[-143.514,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget pruchozi:=[[295.88,343.52,257.92],[0.193217,-
0.677891,0.476889,-0.525079],[-1,-
1,0,4],[143.092,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS wobjdata

kamera_0:=[FALSE,TRUE,"",[[440.36,81.31,89.62],[1,0,0,0]],[[0,0,0],[1,0,0,0]]];

PERS wobjdata

papir:=[FALSE,TRUE,"",[[459.305,164.726,89.62],[0.997464,0,0,0.0711668]],[[0,0,0],
[1,0,0,0]]];

PERS tasks task_list{2}:=[["T_ROB_R"],["T_ROB_L"]];

VAR syncident sync1;

VAR syncident sync2;

VAR syncident sync3;

VAR syncident sync4;

VAR syncident sync5;

VAR syncident sync6;

VAR syncident sync7;

VAR cameratarget cameratargetL;

VAR num L{3,1};

VAR num anglex;

VAR num angley;

VAR num anglez;

VAR num transx;

VAR num transy;

VAR num transz;

VAR num anglez_nove;

PROC main()

!inicializace;

MoveJ doma_L,v300,z50,tool_YuMiGripper_S_C_V1;

WaitSyncTask sync3,task_list;

camera;

prepocetni_wobj;

WaitSyncTask sync4,task_list;

sundani_vicka_fce_l;

```
!MoveAbsJ [[0,-  
130,30,0,40,0],[135,9E+09,9E+09,9E+09,9E+09,9E+09]],v100,fine,tool_YuMiGripper  
_S_C_V1;
```

```
zavreni_vicka_L;
```

```
MoveJ Pred_vickem10,v300,z50,tool_YuMiGripper_S_C_V1;
```

```
MoveJ doma_L,v300,z50,tool_YuMiGripper_S_C_V1;
```

```
ENDPROC
```

```
PROC zavreni_vicka_L()
```

```
WaitSyncTask sync5,task_list;
```

```
MoveL chyceni_vicka,v200,fine,tool_YuMiGripper_S_C_V1;
```

```
WaitSyncTask sync6,task_list;
```

```
Hand_GripOutward;
```

```
WaitSyncTask sync7,task_list;
```

```
ENDPROC
```

```
PROC camera()
```

```
CamSetProgramMode kamera;
```

```
MoveJ pruchozi,v300,z50,tool_YuMiGripper_S_C_V1;
```

```
MoveJ Foto,v100,fine,tool_YuMiGripper_S_C_V1;
```

```
CamSetProgramMode kamera;
```

```
CamStartLoadJob kamera,"poloha_L.job";
```

```
CamWaitLoadJob kamera;
```

```
CamSetRunMode kamera;
```

```

CamReqImage kamera;

CamGetResult kamera,cameratargetL;

ENDPROC

PROC prepocetni_wobj()

L{1,1}:=cameratargetL.cframe.trans.x;

L{2,1}:=cameratargetL.cframe.trans.y;

L{3,1}:=EulerZYX(\Z,cameratargetL.cframe.rot);

anglex:=EulerZYX(\X,kamera_0.uframe.rot);

angley:=EulerZYX(\Y,kamera_0.uframe.rot);

anglez:=EulerZYX(\Z,kamera_0.uframe.rot);

anglez_nove:=anglez+L{3,1};

papir.uframe.rot:=OrientZYX(anglez_nove,angley,anglex);

papir.uframe.trans.x:=kamera_0.uframe.trans.x-L{1,1}+200;

papir.uframe.trans.y:=kamera_0.uframe.trans.y-L{2,1}+100;

papir.uframe.trans.z:=kamera_0.uframe.trans.z;

ENDPROC

PROC sundani_vicka_fce_l()

Hand_MoveTo 7.5;

MoveJ Pred_vickem10,v300,z50,tool_YuMiGripper_S_C_V1;

MoveJ Pred_vickem20,v300,z50,tool_YuMiGripper_S_C_V1;

```

```
WaitSyncTask sync1,task_list;
```

```
MoveL chyceni_vicka,v200,fine,tool_YuMiGripper_S_C_V1;
```

```
Hand_GripInward;
```

```
WaitSyncTask sync2,task_list;
```

```
ENDPROC
```

```
PROC inicializace()
```

```
Hand_DoCalibrate\Jog;
```

```
ENDMODULE
```