

Tvorba výukových materiálů pro robotické rameno Stäubli

Martin Zicháček

Bakalářská práce
2025



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav automatizace a řídicí techniky

Akademický rok: 2024/2025

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Martin Zicháček
Osobní číslo: A22208
Studijní program: B0714A150006 Aplikovaná informatika v průmyslové automatizaci
Specializace: Inteligentní systémy s roboty
Forma studia: Prezenční
Téma práce: Tvorba výukových materiálů pro robotické rameno Stäubli
Téma práce anglicky: Creation of Teaching Materials for the Stäubli Robotic Arm

Zásady pro vypracování

1. Seznamte se s typy robotů v robotizované technologické lince, vývojovým prostředím a souvisejícím technickým vybavením.
2. Navrhněte jednotlivé vhodné úlohy pro účely výuky, využívající vybavení a možnosti konkrétního robota/pracoviště.
3. Navržené úlohy realizujte ve vývojovém prostředí.
4. Vytvořené úlohy otestujte ve vývojovém prostředí a na zvoleném reálném robotu.
5. K jednotlivým navrženým úlohám vytvořte manuály s podrobným postupem řešení jednotlivých navržených úloh.

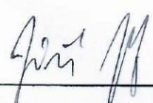
Forma zpracování bakalářské práce: **tisková/elektronická**

Seznam doporučené literatury:

1. SKAŘUPA, Jiří. *Průmyslové roboty a manipulátory*. Ostrava: VŠB – Technická univerzita, 2008. ISBN 978-80-248-1522-0.
2. BOTEK, Petr. *Tvorba výukových materiálů pro robotickou ruku Stäubli TX2-60*. 2022. Dostupné také z: <http://hdl.handle.net/10563/50435>
3. STÄUBLI, International. *CS9 Controller: Instruction manual*. D.280.940.04-K. Pfäffikon / Switzerland, 2023.
4. STÄUBLI, International. *STÄUBLI Robotics Suite 2022 (SRS): User's manual*. D.280.983.04-B. Pfäffikon / Switzerland, 2022.
5. STÄUBLI, International. *VAL 3 reference manual: Version 8*. D.280.935.04-H. Pfäffikon / Switzerland, 2022.
6. STÄUBLI, International. *Arm – TX2 series 60 family: Instruction manual*. D.280.942.04-G. Pfäffikon / Switzerland, 2022.
7. STÄUBLI, International. *Arm – TS series 40/60/80 family: Instruction manual*. C.285.036.04-E. Pfäffikon / Switzerland, 2020.

Vedoucí bakalářské práce: **Ing. Petr Navrátil, Ph.D.**
Ústav řízení procesů

Datum zadání bakalářské práce: **13. prosince 2024**
Termín odevzdání bakalářské práce: **27. května 2025**



doc. Ing. Jiří Vojtěšek, Ph.D.
děkan





prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Ve Zlíně dne 13. prosince 2024

PROHLÁŠENÍ AUTORA BAKALÁŘSKÉ PRÁCE

Beru na vědomí, že

- odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., v platném znění bez ohledu na výsledek obhajoby;
- bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému a bude dostupná k nahlédnutí;
- jedno vyhotovení bakalářské práce v listinné podobě bude ponecháno Univerzitě Tomáše Bati ve Zlíně k uložení;
- na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – bakalářskou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá; neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji, že

- jsem na bakalářské práci pracoval(a) samostatně a použitou literaturu jsem řádně citoval(a); v případě publikace výsledků budu uveden(a) jako spoluautor;
- odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.

Ve Zlíně, dne

.....

podpis autora

ABSTRAKT

Bakalářská práce se zabývá tvorbou výukových materiálů za účelem seznámení studentů s problematikou programování průmyslových robotů Stäubli. Jedná se o vytvoření vhodných ukázkových materiálů/příkladů v simulačním prostředí a jejich ověření na reálném robotu. Pro jednotlivé ukázkové příklady budou vytvořeny manuály, které krok za krokem budou vysvětlovat základy práce ve vývojovém prostředí Stäubli Robotics Suite a také s reálným robotem.

Klíčová slova: robot, aplikace, Stäubli, úloha,

ABSTRACT

The bachelor's thesis focuses on the development of educational materials aimed at introducing students to the programming of Stäubli industrial robots. It involves creating suitable demonstration materials/examples in a simulation environment and verifying them on a real robot. For each example, manuals will be prepared, providing step-by-step explanations of the basics of working in the Stäubli Robotics Suite development environment as well as with the real robot.

Keywords: robot, application, Stäubli, examples

Poděkování

Chtěl bych poděkovat svému vedoucímu bakalářské práce Ing. Petru Navrátilovi, Ph.D. za odborné vedení a ochotu poradit a navést. A chtěl bych také poděkovat mým rodičům za podporu při studiu.

OBSAH

ÚVOD.....	11
I TEORETICKÁ ČÁST	12
1 ROBOTIKA.....	13
1.1 HISTORIE ROBOTIKY	13
1.2 HISTORIE PRŮMYSLVÉ ROBOTIKY	14
2 ROBOT.....	15
2.1 DEFINICE.....	15
2.2 SLOŽENÍ ROBOTA	15
2.3 SENZORY.....	15
2.4 ŘÍDÍCÍ JEDNOTKA	16
2.6 AKČNÍ ČLENY	17
3 USPOŘÁDÁNÍ ROBOTŮ	18
3.1 KARTÉZSKÝ	18
3.2 VÁLCOVÝ	19
3.3 SFÉRICKÝ	20
3.4 SCARA.....	21
3.5 ANTROPOMORFNÍ	22
4 MOBILNÍ ROBOTY	23
4.1 KOLOVÉ ROBOTY	23
4.2 KRÁČEJÍCÍ ROBOTY	23
4.3 LÉTAJÍCÍ ROBOTY	23
5 MANIPULÁTORY	24
5.1 ROZDĚLENÍ MANIPULÁTORŮ	24
5.1.1 Podle stupňů volnosti	24
5.1.2 Podle kinematické struktury.....	24
5.1.3 Podle jejich pohonu.....	24
5.2 ČÁSTI MANIPULÁTORŮ	25
5.2.1 Smyčky manipulátoru	25
5.2.2 Polohovací ústrojí.....	25
5.2.3 Pojezdové ústrojí.....	25
5.2.4 Orientační ústrojí.....	25
5.3 KONCOVÉ EFEKTORY	26
5.3.1 Grippers	26
6 STÄUBLI(FIRMA)	27
7 STÄUBLI TS2-60	28

8	STÄUBLI TX2-60.....	29
9	CS9 CONTROLLER.....	30
II	PRAKTICKÁ ČÁST	32
10	PRÁCE VE STÄUBLI ROBOTICS SUITE	33
10.1	ZAPNUTÍ ROBOTICKÉ LINKY	33
10.2	ZALOŽENÍ NOVÉHO PROJEKTU V SRS	35
10.3	PRÁCE V PROSTŘEDÍ SRS	38
10.3.1	Karty v SRS.....	38
10.3.2	Založení nové aplikace	42
10.3.3	Vytvoření nástroje	43
10.3.4	Vložení CAD souboru	49
10.3.5	Vytvoření objektů.....	50
10.3.6	Vytvoření Bodů	50
10.3.7	Vlastní souřadný systém.....	55
10.3.8	Tvorba a využití knihoven.....	56
10.3.9	Založení vlastního uživatelského rozhraní a práce s ním.....	57
10.3.10	Vytvoření programu	59
10.3.11	Vytvoření podprogramu	61
10.3.12	Spuštění simulace	62
11	ÚLOHY PRO PROCVIČENÍ PRÁCE V SRS	64
11.1	ÚLOHA 1 – PRÁCE S POHYBEM MEZI BODY	64
11.1.1	Zadání úlohy 1	64
11.1.2	Provedení úlohy 1	64
11.2	ÚLOHA 2 – DEPALETIZACE	68
11.2.1	Zadání úlohy 2.....	68
11.2.2	Provedení úlohy 2.....	68
11.3	ÚLOHA 3 – STAVĚNÍ PYRAMIDY	71
11.3.1	Zadání úlohy 3	71
11.3.2	Provedení úlohy 3.....	71
11.4	ÚLOHA 4 – PRÁCE S GUI.....	73
11.4.1	Zadání úlohy 4.....	73
11.4.2	Provedení úlohy 4.....	73
12	VYZKOUŠENÍ VYTVOŘENÝCH MANUÁLŮ.....	79
	ZÁVĚR.....	80
	SEZNAM POUŽITÉ LITERATURY	81
	SEZNAM POŽITÝCH SYMBOLŮ A ZKRATEK	83
	SEZNAM OBRÁZKŮ	84
	SEZNAM PŘÍLOH.....	87

ÚVOD

V dnešní době se stále více pracovišť robotizuje a stále více se využívají pracovní roboti. V teoretické části bakalářské práce je popsána historie robotů a jejich oboru robotiky. Dále je popsán robot jako stroj, kde se zabýváme jeho složením, jeho jednotlivými komponenty a různými uspořádáními robotů. Dále jsou stručně rozepsány mobilní roboty a poté více dopodrobna rozepsány manipulátory a jejich rozdělení.

Praktická část bakalářské práce je zaměřená na návrh úloh a vypracování manuálů pro výuku studentů. První část praktické části je koncipována jako všeobecný návod pro práci v prostředí *Staubli Robotics Suite*. Tento návod obsahuje řešení základních operací v tomto vývojovém prostředí. Úlohy pro výuku jsou zpracovány formou zadání a většinou slovních manuálů, kde je krok po kroku vysvětlený postup a zdůvodněno proč je postup takový. V první úloze je cílem studenty seznámit s určováním bodů na robotovi, základními pohyby robota a jejich využití. V druhé úloze se studenti seznámí s prací s *Grippers* a jednoduchými programovacími smyčkami. Ve třetí úloze si studenti procvičí dovednosti získaných v předchozích dvou úlohách u trochu těžšího zadání. Čtvrtá úloha je zaměřená na práci s vlastním uživatelským prostředím, kde se studenti učí, jak propojit prvky uživatelského prostředí s veličinami a ty následně použít pro psaní kódu programu. Všechny úlohy jsou prvně simulovány a po žádoucích úpravách přesunuty na reálného robota. Díky tomuto postupu si studenti vyzkouší práci v obou prostředích.

I TEORETICKÁ ČÁST

1 ROBOTIKA

Robotika je věda zabývající se vývojem a využitím ro/botů.

1.1 Historie robotiky

Člověk jako organismus od počátku věků hledal způsoby, kterými by mohl provést činnosti potřebné k jeho přežití s vydáním nejmenší možné energie. Tato filozofie přetrvává i v dnešním moderním světě, kde jeden z jejich velice důležitých vynálezů je oblast robotiky, nebo např oblast umělé inteligence. Slovo „Robot“ jako takové se prvně objevilo v literárním díle od českého spisovatele Karla Čapka. Toto dílo neslo název R.U.R. (v celém znění Rossumovi univerzální roboti).

Význam robota jako mechanický stroj se objevil ve 40. letech 20. století kdy ruský spisovatel Isaac Asimov pojal robota jako stroj s lidskou podobou ale bez citů. V jedné z jeho povídek o robotovi, který dokázal číst myšlenky Asimov spolu s jeho redaktorem jménem John Wood Campbell vymysleli základní pravidla pro chování robota, tak aby byl lidstvu prospěšný. Tyto zákony jsou nazvány Asimovovy zákony a jsou následující:

- Robot nesmí ublížit člověku nebo svou nečinností dopustit, aby bylo člověku ublíženo
- Robot musí uposlechnout příkazů člověka, kromě případů, kdy jsou tyto příkazy v rozporu s prvním zákonem
- Robot musí chránit sám sebe před poškozením, kromě případů, kdy je tato ochrana v rozporu s prvním zákonem

Postupem času se přidaly další zákony jako třeba nultý zákon, který zní:

- Robot nesmí ublížit lidstvu nebo svou nečinností dopustit, aby mu bylo ublíženo.

Další autoři se pokusili o doplnění či změnu těchto zákonů [1].

1.2 Historie průmyslové robotiky

Přestože většina vizionářů si robota představovala jako humanoidní stroj, nejvíce užitečné využití robotů bylo v oblasti průmyslu výroby. První industriální roboty byly zavedeny v podobě robotického ramena, které závisle na jeho designu mělo daný počet stupňů volnosti. Díky zavedení oboru průmyslové robotiky se drasticky změnilo pracovní prostředí v mnoha oborech. Zavedení robotů do průmyslové výroby změnilo i produkční procesy, jelikož člověk byl nahrazen hlavně v repetitivních, nebezpečných či velmi únavných procesech. Roboti ve výrobě mají také spoustu pozitiv z pohledu produktivity, kvality výroby a přesnosti.

První komerčně dostupný průmyslový robot byl vyvinutý v roce 1954 vynálezcem Georgem Devolem, který ho pojmenoval “Unimate” [2]. Tento robot v navržené rameni s manipulátorem byl pohybován pomocí hydraulického systému a mohl provádět různé repetitivní operace. Měl 4 stupně volnosti, 3 rotační klouby a dokázal uzvednout těleso o váze až 34 kg. Tento robot byl zakoupen americkou firmou General Motors a v roce 1961 byl zaveden do výroby, kde jeho práce spočívala ve vyzvednutí odlitých dílů z dopravníku a následné sváření se zbytkem karoserie. Po této první adopci ve výrobě se instalování robotů do výrobních hal pro automobilový průmysl stalo nedílnou součástí výroby ve většině zemích. V Evropě v roce 1973 byly přivedeny na trh další industriální roboti od firem ABB a KUKA Robotics. V roce 2017 bylo již zaimplementováno 2.5 miliónu těchto průmyslových robotů po celém světě.

Další pokrok ve světě průmyslových robotů bylo představení takzvaných kolaborativních robotů (Cobot). Tito roboti byli navrženi tak, aby mohli spolupracovat v blízkosti s lidskými pracovníky. Museli tedy být zkonstruováni, tak aby neublížili lidem v jejich pracovním okolí. Filozofie kolaborativních robotů lpí na tom, že roboti by neměli nahradit lidské pracovníky, ale vylepšit jejich pracovní výkon pomocí jejich dovedností. Tato filozofie představuje zlom ve vnímání robotů jako pracovní síly. [3]

2 ROBOT

2.1 Definice

Robot je elektro-mechanické zařízení s více úhly volnosti, které se dá naprogramovat, aby provedlo různou řadu úkolů [4]. Podle Robotického Institutu Ameriky (RIA): „Robot je pře programovatelný multifunkční manipulátor navrhnutý na přesun materiálu, nástrojů, nebo speciálních zařízení skrz programování pomocí proměnných pro vykonávání více různých úkonů.“

Roboty můžeme primárně rozdělit na mobilní roboty a manipulátory.

2.2 Složení robota

Robot se jako systém skládá z: Pojezdového ústrojí, manipulátoru, zápěstí, koncového efektoru, akčních členů, snímačů, řídicího systému a softwaru. Tyto části v zapojení spolu dohromady umožňují pohyb robota a výkon mu zadaných úkolů. Mechanická část robota je sestavena z více článků spojených klouby a tvoří kinematický řetězec. Tyto články se mohou pohybovat vzhledem k ostatním článkům. Kloubů, kterými jsou pevné články propojeny mezi sebou je 6 typů: rotační kloub, translační kloub, cylindrický kloub, šroubový kloub, sférický kloub a planární kloub. U průmyslových robotů využívaných v průmyslu se nejčastěji využívají klouby rotační a translační. Spojením pomocí kloubů můžeme články popsat vzájemnou polohou proti jinému článku, který je s daným článkem spojen. Tato poloha je popsána kloubovými souřadnicemi. Tyto souřadnice jsou u každého typu kloubu jiné, například u translačního kloubu, jelikož se dva články pohybují po stejné ose jako kloubová souřadnice se považuje vzdálenost mezi články. Každý kloub obsažený v robotu mu přidává stupeň volnosti (DOF) a tím zlepšuje dosah a manévrovatelnost robota v prostoru. Po dosažení 6 DOF je robot schopný dosáhnout jakéhokoliv bodu v jeho pracovní oblasti, a proto přidávání dalších kloubů je kinematicky zbytečné. [12]

2.3 Senzory

Senzory jsou nesdílnou částí konstrukce robota, jelikož fungují jako smysly, podle kterých se robot řídí. Tyto senzory robotu umožňují sbírat informace nejenom z jeho okolí, ale také z jeho polohy, nebo nastavení kloubů. Pomocí nasbíraných informací, které jsou pak odeslány do řídicí jednotky, přesně víme v jaké poloze se robot nachází a jaké příkazy má řídicí jednotka vyslat.

Rozdělení senzorů:

- **Vidění** – Systém pro vidění se skládá z kamery, díky které robot vidí své okolí. Tyto senzory jsou často využívány při sestavování elektronických součástek pro přesné usazování čipů do jejich pozic.
- **Hlas** – Senzory pro rozpoznávání hlasu se využívají při ovládání robotů pomocí hlasu
- **Blížkost** – Tyto senzory robota upozorní, pokud je v blízké vzdálenosti od senzoru nějaký předmět. Jsou často využívány pro vyhýbání se kolizím u mobilních robotů.
- **Taktilní** – umožňuje detekci a měření kontaktu s povrchy těles. Často se používají, aby robot mohl „cítit“ své okolí a pro jemnou manipulaci s předměty
- **Úhlu náklonu** – měří pod jakým úhlem je robot nakloněn. V podstatě nutný senzor pro drony.
- **Koncové senzory** – často jsou nainstalované na okrajích pracovního prostoru robota, aby upozornil robota, že se dostal ze svého pracovního prostoru.

[4]

2.4 Řídící jednotka

Řídící jednotka zajišťuje řízení a koordinaci jednotlivých částí robota. Zahrnuje hardware i software, díky kterým robot zpracovává data získané senzory a vyhodnocuje další signály, tyto signály určují, jak se robot pohne. Tyto signály řídící jednotka dále pošle do akčních členů, aby se robot nastavil do správné polohy potřebné pro vykonání úkolu. Řídící jednotka také může komunikovat s externími zařízeními, nebo operátory. [13]

2.6 Akční členy

Akční členy jsou prvky, které provádějí fyzické akce podle příkazů, které vysílá řídicí jednotka. Zajišťují například pohyb robota jako třeba manipulace s objekty, nebo pohyb robota v prostoru. Akční členy se rozdělují podle pohonu, který práci vykonává. Pohony jsou elektrické, pneumatické či hydraulické kde každý pohon má své výhody i nevýhody. Elektrický pohon aktuátoru je velice často používán kvůli jejich přesnosti, jednoduchosti zapojení a tichosti. Elektrické pohony můžeme realizovat jak stejnosměrnými, tak střídavými motory. Mezi často používané elektrické motory patří například jednoduchý DC motor, servomotor nebo krokový motor. Pneumatické pohony pro pohyb robotem využívají stlačený vzduch, který vede v trubičkách v konstrukci robota. Tento systém je obvykle lehčí a rychlejší, než pohony pneumatické ale mají omezenou sílu. Často se využívají pro grippery. Hydraulické pohony využívají kapaliny pod tlakem (většinou olej). Tento systém dokáže zvládnout velké náklady a má vysokou přesnost v pohybu, ale je pomalejší a potřebuje složitější údržbu, jako například výměnu filtrů oleje. Jako aktuátor můžeme také využít piezoelektrické materiály, ale jelikož jejich velikost pohybu a jeho síla je velice malá nedává to z konstruktérského hlediska smysl. V šedesátých letech 20. století byl vynalezen tzv umělý sval který byl nazván McKibben Muscle. Tento akční člen sice v té době nebylo možné ovládat, ale v dnešní moderní době lze tento umělý sval využívat hlavně v konstrukci robotických paží nebo jiných lidských částí těla. [4]

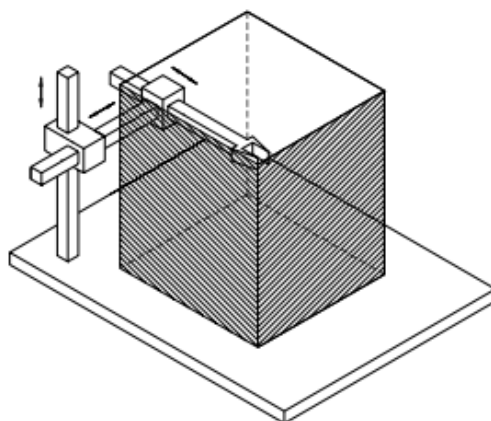


Obrázek 1 McKibben Muscle [5]

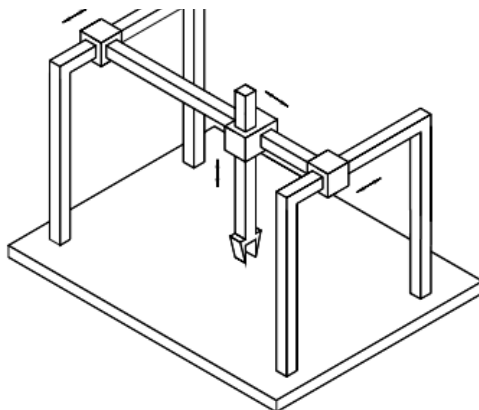
3 USPOŘÁDÁNÍ ROBOTŮ

3.1 Kartézský

Kartézská geometrie je realizována třemi posuvnými klouby, jejichž osy jsou obvykle vzájemně kolmé. Díky jednoduché geometrii tohoto uspořádání každý stupeň volnosti odpovídá jedné kartézské souřadnici, a proto je přirozené provádět přímočaré pohyby v prostoru. Kartézská struktura poskytuje velice dobrou mechanickou tuhost a přesnost polohování zápěstí je konstantní v celém pracovním prostoru. Tento prostor je vymezen kvádrem. Výhoda této struktury je že má vysokou přesnost. Jako jasná nevýhoda je nízká obratnost, protože všechny klouby jsou posuvné. Nejčastěji využívaná konstrukce této geometrie je portálová konstrukce. Tato konstrukce manipuluje s objektem přístupem shora a poskytuje tak velký pracovní prostor a umožňuje manipulaci s rozměrnými těžkými předměty. Motory pohánějící klouby kartézského manipulátoru jsou obvykle elektrické, výjimečně pneumatické. [1]



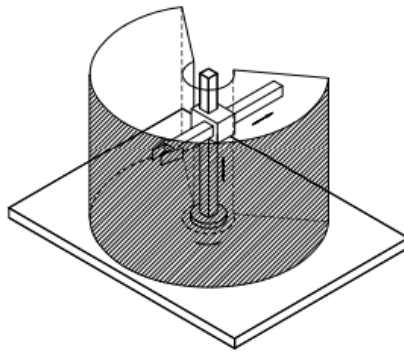
Obrázek 2 Kartézský manipulátor [1]



Obrázek 3 Portálová konstrukce [1]

3.2 Válcový

Válcová geometrie se od kartézské liší tím, že první translační kloub je nahrazen otočným kloubem. Pokud je úloha popsána s pomocí válcových souřadnic, každý stupeň volnosti také odpovídá jedné kartézské souřadnici. Válcová struktura poskytuje dobrou mechanickou tuhost, ale přesnost polohování zápěstí se snižuje s rostoucím vodorovným dosahem. Pracovní prostor tvoří část dutého válce. Vodorovný posuvný kloub umožňuje přístup do horizontálních dutin. Jelikož se válcové manipulátory převážně využívají pro přepravu předmětů je u nich vhodné použít spíše hydraulické motory než motory elektrické. [1]

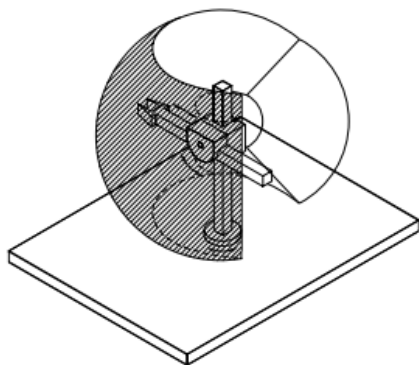


Obrázek 4 Válcový manipulátor [1]

3.3 Sfěrický

Sfěrická geometrie se od válcové liší tím, že druhý posuvný kloub je nahrazen otočným kloubem. Tato konstrukce bude mít tedy 2 klouby otočné a jeden posuvný. Každý stupeň volnosti odpovídá kartézské souřadnici. Mechanická tuhost je nižší, než u předchozích dvou geometrií a konstrukce je složitější. Přesnost polohování zápěstí opět klesá s narůstajícím radiálním dosahem. Pracovní prostor tvoří část duté koule a může zahrnovat i základnu manipulátoru, což umožňuje manipulaci s předměty na podlaze. Sfěrické manipulátory se často využívají při obrábění, přičemž klouby jsou obvykle poháněny elektrickými motory.

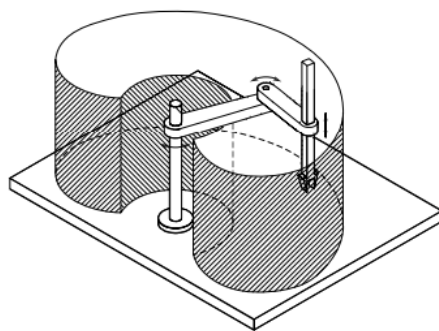
[1]



Obrázek 5 Sfěrický manipulátor [1]

3.4 SCARA

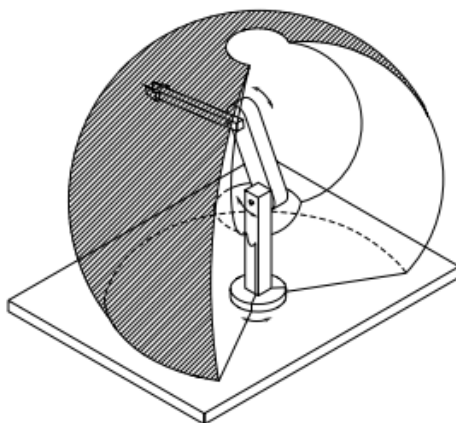
Speciální geometrie SCARA je dosažena pomocí dvou otočných kloubů a jednoho posuvného kloubu, jejichž osy pohybu jsou vzájemně rovnoběžné. Akronym SCARA v angličtině znamená Selective Compliance Assembly Robot Arm. Tato geometrie nabízí vysokou tuhost vůči vertikálním zatížením a poddajnost vůči horizontálním zatížením. Proto se tato geometrie velice hodí pro vertikální montážní operace. Vztah mezi stupni volnosti a kartézskými souřadnicemi tady platí pouze pro vertikální složku pohybu. Přesnost polohování zápěstí klesá se vzrůstající vzdáleností od osy prvního kloubu. Pracovní prostor je ilustrován na obrázku č.6. SCARA manipulátory jsou ideální k manipulaci malých objektů a jsou poháněny elektrickými motory. [1]



Obrázek 6 SCARA manipulátor [1]

3.5 Antropomorfní

Antropomorfní geometrie je realizovaná třemi otočnými klouby kde osa prvního kloubu je kolmá k osám dvou dalších kloubů. Tyto dva druhé otočné klouby jsou osami vzájemně rovnoběžné. Díky podobnosti s lidskou rukou se druhý kloub označuje jako ramenní a třetí jako loketní. Tato struktura je nejvíce obratná ze všech uspořádání, avšak ztrácí se přímá vazba mezi stupni volnosti a kartézskými souřadnicemi. To způsobuje nespolehlivou přesnost polohování zápěstí. Pracovní prostor má tvar výseče koule a je výrazně rozsáhlejší než fyzické rozměry manipulátoru. Manipulátory zkonstruované touto geometrií jsou využívány v široké škále průmyslových aplikací jako je například sváření, lakování či montáž. Pohon u těchto manipulátorů je typicky řešen elektrickými motory. [1]



Obrázek 7 Antropomorfní manipulátor [1]

4 MOBILNÍ ROBOTY

Hlavní schopností mobilních robotů je jejich možnost pohybovat se v prostředí a tím pádem vykonávat úkoly, které zahrnují navigaci, manipulaci nebo sběr dat. Tyto roboty mohou měnit svou polohu a své pracoviště často a bez problému měnit je-li to po nich požadováno. Tento typ robota je široce využíván v různých odvětvích od průmyslové automatizace až po vojenské účely.[6]

4.1 Kolové roboty

Tito roboti se pohybují pomocí různého počtu kol. Existují čtyři často používané typy kol a to jsou: Pevné, Řiditelné, Rejdovací a Mecanum kolo. Pevné kolo může rotovat kolem osy procházející středem kola a je kolmé k rovině kola. Kolo je pevně připojeno k podvozku, jehož orientace vůči kolu je konstantní. Otočné kolo má dvě osy rotace. První je shodná s pevným kolem, zatímco druhá je vertikální a prochází středem kola. To umožňuje měnit orientaci kola vůči podvozku. Rejdovací kolo má opět dvě osy rotace, ale vertikální osa neprochází středem kola – je od něj posunuta o konstantní vzdálenost. Toto uspořádání umožňuje automatické natočení kola, které se samo zarovná se směrem pohybu podvozku. [1]

4.2 Kráčející roboty

Kráčející mobilní roboti jsou tvořeni více pevnými tělesy, která jsou vzájemně propojeny posuvnými nebo otočnými klouby. Některá z těchto těles tvoří dolní končetiny, jejichž zakončení periodicky kontaktují zem, což umožňuje pohyb. Design těchto robotů je často inspirován studiem živých organismů. Konstrukce těchto robotů se liší podle počtu končetin, rozsahu pohyblivosti kloubů a způsobu interakce s terénem, přičemž klíčovou výzvou zůstává udržení stability a plynulosti pohybu v různých prostředích. [1]

4.3 Létající roboty

Tito roboti představují plod vědecké revoluce, která dozrála během 20. století. Tyto létající systémy již dnes operují v našem okolí a v budoucnu najdou uplatnění prakticky všude. Jako možné využití můžeme očekávat například kontrolu stavu napěťových vedení, postřik pesticidy, nebo široké využití v konfliktních zónách. Tito roboti vyžadují vytvoření mnoha dalších dílčích řešení založených na algoritmech umělé inteligence anebo metodách počítačového vidění. [7]

5 MANIPULÁTORY

Manipulátory jsou robotické systémy navržené pro vykonávání různých úkolů manipulace s výrobky či součástkami. Tyto roboty se obvykle skládají z robotického ramena, které má různé stupně volnosti (DOF). Manipulátory jsou využívány v širokém spektru průmyslových a vědeckých aplikací, včetně výroby, montáže, svařování a v oblastech, kde je vyžadována vysoká přesnost a jemná kontrola.[12]

5.1 Rozdělení manipulátorů

Máme spoustu typů rozdělení manipulátorů. Nejpoužívanější typy rozdělení si v této kapitole popíšeme.

5.1.1 Podle stupňů volnosti

Manipulátory můžeme rozdělit podle jejich stupňů volnosti a to na:

- Univerzální robot ($6^\circ V$, Kartézský SS)
- Redundantní robot ($>6^\circ V$)
- Deficitní robot ($<6^\circ V$)

5.1.2 Podle kinematické struktury

Manipulátory lze rozdělit také podle jejich kinematické struktury:

- Sériové (S otevřeným kinematickým řetězcem)
- Paralelní (S uzavřeným kinematickým řetězcem)
- Hybridní (Kombinující oba typy řetězců)

5.1.3 Podle jejich pohonu

- Elektrický pohon
- Pneumatický pohon
- Hydraulický pohon

5.2 Části manipulátorů

V následující kapitole jsou popsány jednotlivé části manipulátorů.

5.2.1 Smyčky manipulátoru

Jsou to cyklické struktury, které se využívají v robotických manipulátorech pro řízení jejich pohybu. Tyto smyčky slouží k propojení jednotlivých článků v rámci manipulátoru. Máme dva typy těchto smyček, a to jsou Kinematické smyčky a dynamické smyčky. Kinematické smyčky jsou užívány k popisu vzorů pohybu manipulátoru. Dynamické smyčky se využívají k analýze sil a momentů působících na různé části manipulátorů.

5.2.2 Polohovací ústrojí

Je to součást robotických manipulátorů, která umožňuje pohyb jednotlivých částí jako jsou například ramena, klouby, nebo koncové nástroje. Toto ústrojí zajišťuje přesné nastavení robota v prostoru pro vykonání požadovaných úloh. Možnost kombinací polohování závisí na stupních volnosti robota.

5.2.3 Pojezdové ústrojí

Slouží k posunu celého robota v rámci jeho pracoviště. Tento posun je brán jako posun základny robota a u manipulátorů často je prováděn pomocí připojení na pojezdné dráhy. U mobilních robotů má robot toto ústrojí zakomponované do jeho konstrukce.

5.2.4 Orientační ústrojí

Za účel má při zachování natočení robota dané polohovacím ústrojím provádět jemnější operace koncovým efektozem. Toto ústrojí umožňuje přesnou změnu úhlu natočení koncového efektoru pro vykonávání úkolů. [13]

5.3 Koncové efekторы

Koncový efektor je zařízení připojené k zápěstí robotického ramene, které umožňuje vykonávat specifické požadované úkoly. Toto zařízení představuje rozhraní mezi robotem a jeho pracovním prostředím, přičemž jeho podoba se odvíjí od požadovaného úkonu robota. Je to poslední článek kinematického řetězce robota. Základně je dělíme na grippery a nástroje. Grippery slouží k úchopu a manipulaci s různými součástkami a nástroje jsou už předem designované pro svůj daný úkol (svářečka). [13]

5.3.1 Grippery

Vývoj robotických gripperů je poháněn potřebou vykonávat konkrétní manuální úkoly nebo splnit specifické požadavky v manipulačních operacích. Grippery se specializovanými funkcemi se liší velikostí, přesností a mírou ovladatelnosti – od malých, přesných a vysoce ovladatelných (jako jsou chirurgické nástroje robota Da Vinci, určené pro minimálně invazivní zákroky ovládané člověkem během laparoskopických operací) až po větší, ale stále velmi přesné chytače, jako je Shadow Dexterous Hand, která napodobuje pohyby lidské ruky. Na druhé straně existují i méně přesně ovladatelné, ale přirozeněji fungující chytače, například iRobot-Harvard-Yale (iHY) Hand nebo IIT-Pisa SoftHand, které využívají biomechanicky inspirovaný design k intuitivnímu uchopování předmětů bez nutnosti složitého řízení. S rostoucí autonomií a rozšířením robotických systémů se stává stále důležitější zaměřit se na design, tvar a funkci robotických chytačů, aby co nejlépe odpovídaly požadavkům různých aplikací.

Přestože se robotické chytače liší designem, lze je rozdělit do skupin podle jejich principu fungování, a to na Antropomimetické, které se snaží napodobovat pohyby lidské ruky, Biomimetické, které čerpají inspiraci pro svou konstrukci z jiných biologických struktur jako jsou například chapadla, nebo klepeta a nebiometrické, které nejsou inspirovány přírodou.

Grippery můžeme také rozdělit podle jejich pohonu, a to na Pneumatické grippery, které jsou poháněné stlačeným vzduchem a jsou vhodné pro rychlé a lehké aplikace, Hydraulické grippery, které poskytují vysokou sílu, ale mají složitější údržbu, Elektrické grippery, které využívají možnosti pro přesné a programovatelné ovládání, a Vakuové grippery, využívající podtlaku pro uchopení hladkých a lehkých předmětů jako jsou například skla, nebo plechy.[8]

6 STÄUBLI(FIRMA)

Společnost Stäubli je mezinárodně působící výrobce průmyslových technologií se sídlem ve Švýcarsku. Firma, založená v roce 1892, se původně zaměřovala na textilní stroje, ale postupně rozšířila své portfolio o rychlospojky, robotiku a automatizační systémy. Dnes má Stäubli pobočky ve více než 50 zemích a dodává své produkty do různých odvětví, včetně automobilového průmyslu, letectví, elektroniky a zdravotnictví. Tato firma má současně přes 6000 zaměstnanců.

Pro roboty firmy Stäubli je také vyvinuto vlastní vývojové prostředí s možností simulace Stäubli Robotics Suite.[18]

Mezi klíčové produkty společnosti patří průmyslové roboty, elektrické a pneumatické konektory a textilní stroje. Stäubli klade důraz na inovace a dlouhodobou spolehlivost svých řešení, což mu umožňuje udržovat silnou pozici na globálním trhu. Firma spolupracuje s průmyslovými podniky a výzkumnými institucemi, aby dále rozvíjela automatizaci a moderní výrobní technologie. Její produkty jsou známé pro vysokou přesnost a odolnost, což je předurčuje pro náročné průmyslové aplikace.[19]

7 STÄUBLI TS2-60

Je to 4osý robot od švýcarské firmy Stäubli. Jedná se o robot typu SCARRA. Robot má široké využití v průmyslu. Lze jej použít ve všech prostředích jak ve špinavých, tak i ve sterilních. Robot disponuje dutými hřídelemi, a tak není potřeba kabeláže z vnější strany robota, ale kabely jsou vedeny těmito dutými hřídelemi. Celé rameno je voděodolné s ochrannou třídou IP65. Toto rameno lze namontovat jak na strop, tak na podlahu pracovního prostoru. Robot má velkou obratnost v malých prostorech díky kompaktnímu navržení motorů. Tento robot má také modulární design a sdílí komponenty s ramenem TX2-60. Chytrá převodovka JCS na ose 1 a 2 poskytuje vysokou dynamiku a přesnost trajektorie díky vyšší frekvenci pohonů. Disponuje také 19bitovými enkodéry, které umožňují, že robot je připraven k provozu bez nutnosti inicializace. Tento robot má jedny z nejmenších nároků na uvedení do provozu a jeho následnou údržbu.[11] [17]



Obrázek 8 Obě instalace robota TS2-60 [11]

8 STÄUBLI TX2-60

Je to 6osý industriální robot od firmy Stäubli s ideální kombinací rychlosti, velikosti, přesnosti a houževnatosti. Tento robot má široké využití ve všech možných odvětvích, a to od sváření až po využití ve sterilních oblastech. Umožňuje flexibilní montáž v rozsahu 360°, a to na podlahu, stůl, stěnu i strop. Robot disponuje integrovanými elektrickými a pneumatickými obvody, včetně rozhraní Cat5e pro připojení kamer a senzorů. Jeho konstrukce je lehká, avšak pevná, celé rameno je hygienicky zapouzďeno a díky použití dutých hřídelí nejsou potřeba vnější kabely.

Vyžaduje minimální údržbu – nemá baterie ani v rameni, ani v ovládací jednotce. Všechny komponenty jsou snadno přístupné a jednoduše vyměnitelné. Díky tomu jsou celkové náklady na vlastnictví nízké a návratnost investice rychlá.

Robot je vybaven všemi běžnými komunikačními protokoly, včetně těch pro reálný čas, a už z výroby obsahuje rozsáhlá bezpečnostní opatření.[9] [16]



Obrázek 9 Robot TX2-60 [9]

9 CS9 CONTROLLER

Díky svému promyšlenému designu a vysoké kapacitě zajišťuje řídicí jednotka nepřetržitou provozní dostupnost. Její otevřená architektura zajišťuje kompatibilitu s moderním výrobním prostředím a umožňuje rychlou a snadnou integraci. Bezpečnostní prvky jsou integrované pro každou možnou aplikaci.[15]

Tento controller slouží k řízení SCARA robotů, 6osých průmyslových robotů, kolaborativních POWER robotů i mobilních systémů. Je vybaven univerzálním napájecím zdrojem pro různé typy napětí a díky vyměnitelným panelům lze snadno měnit konektorové sady podle potřeby.

Součástí platformy je SP2 pendant – lehký a odolný ovládací panel s uživatelským rozhraním pro provoz, konfiguraci i vizualizaci. Nabízí integrované bezpečnostní prvky, včetně třípolohového tlačítka a nouzového zastavení.

Díky multitaskingovému jazyku VAL3 a možnosti rozšíření o knihovny uniVAL PLC je systém plně programovatelný v jazyce dle standardu IEC 61131. Ovladač uniVAL Drive umožňuje přesné řízení kloubového pohybu i přes externí kontroléry. [14]

Řídicí jednotka navíc rekuperuje energii při zpomalování pohybů a nabízí režimy úspory energie s potenciálem snížení spotřeby až o 40 %. Podporuje všechny běžné komunikační protokoly, včetně real-time Ethernetu. Vestavěný webový server založený na HTML poskytuje přístup k uživatelskému rozhraní z jakéhokoliv zařízení. Nechybí ani podpora EtherCat pro flexibilní připojení periférií. [10]



Obrázek 10 Řídicí jednotka CS9 [10]

II PRAKTICKÁ ČÁST

10 PRÁCE VE STÄUBLI ROBOTICS SUITE

10.1 Zapnutí robotické linky

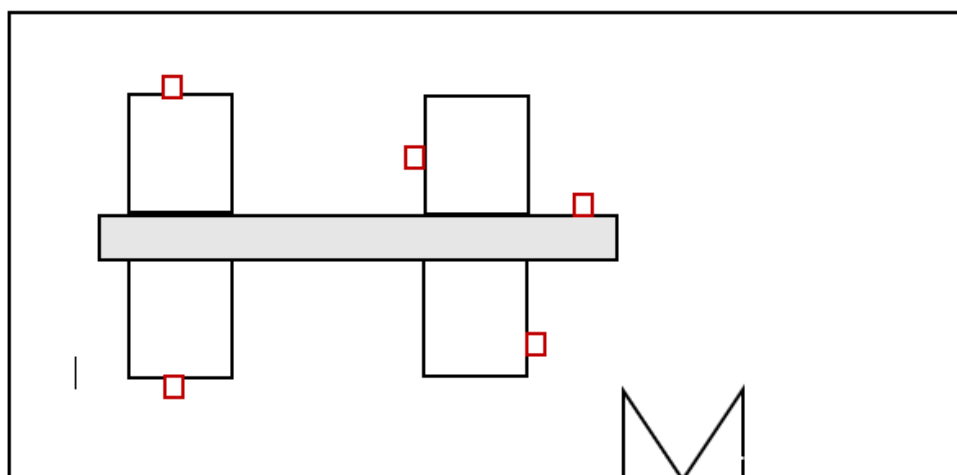
Abychom mohli pracovat s roboty v laboratoři musíme prvně zapnout celou robotickou linku. Ta se skládá ze 4 stanovišť s roboty propojené pásovým dopravníkem.

Linku zapneme tak, že pomocí vypínačů zapneme všechny jednotlivé stanoviště a dopravník.



Obrázek 11 Hlavní vypínač

Takových vypínačů musíme zapnout 5 a jejich lokace jsou následující.



Obrázek 12 Umístění vypínačů

Vypínač u robota ABB Yuumi je většinou zapnutý.

Po zapnutí všech vypínačů můžeme pracovat s ovládacím panelem jednotlivé robotické kóje. Pro tu máme 2 ovládací panely, první panel pro práci s kójí a druhý panel pro práci s robotem.



Obrázek 13 Panel pro práci s kójí

Pokud chceme zasahovat do prostoru kóje tak klíč musí být otočen do polohy *Servis*. Poté po zmáčknutí tlačítka *Reset bezpečnostních obvodů* se uvolní zámky a můžeme kóji otevřít.

Na panelu se také nachází tlačítka pro *Nouzové zastavení*, *Start cyklu*, *Stop cyklu* a *Provozní zastavení*.

Pokud chceme s robotem pohybovat musíme klíč nastavit do polohy *Provoz* a poté co se ujistíme, že všechny dveře kóje jsou zavřené zmáčkne tlačítka *Reset bezpečnostních obvodů*. Počkáme, dokud neuslyšíme jak se zámky zamknou a poté znovu zmáčkne tlačítka *Reset bezpečnostních obvodů*. Pokud je na obrazovce napsáno **press and release robot's resets** musíme se přesunout na druhý panel a tam zmáchnout tlačítka *Restart*.



Obrázek 14 Panel pro práci s robotem

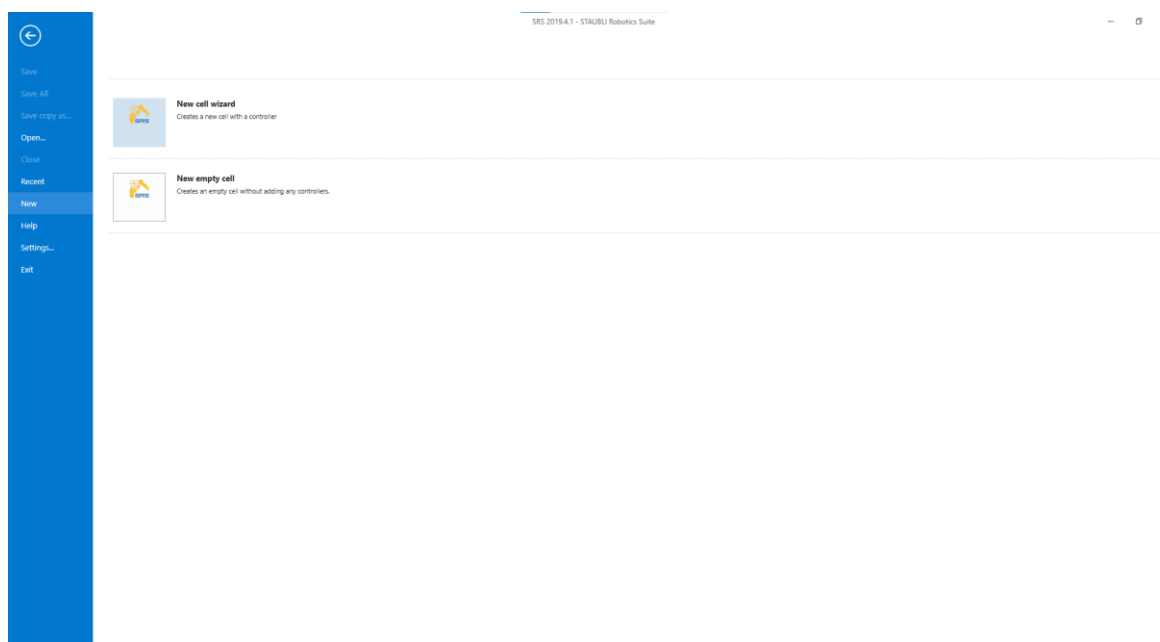
Na panelu se nachází tlačítka E-STOP, tlačítka pro restart robotické ruky a klíč, kde vybíráme mód robota. Vrchní možnost je robot v manuálním režimu, prostřední je robot v automatickém režimu a spodní volba je robot ve vzdáleném režimu.

Nyní by robot měl být připraven k používání.

10.2 Založení nového projektu v SRS

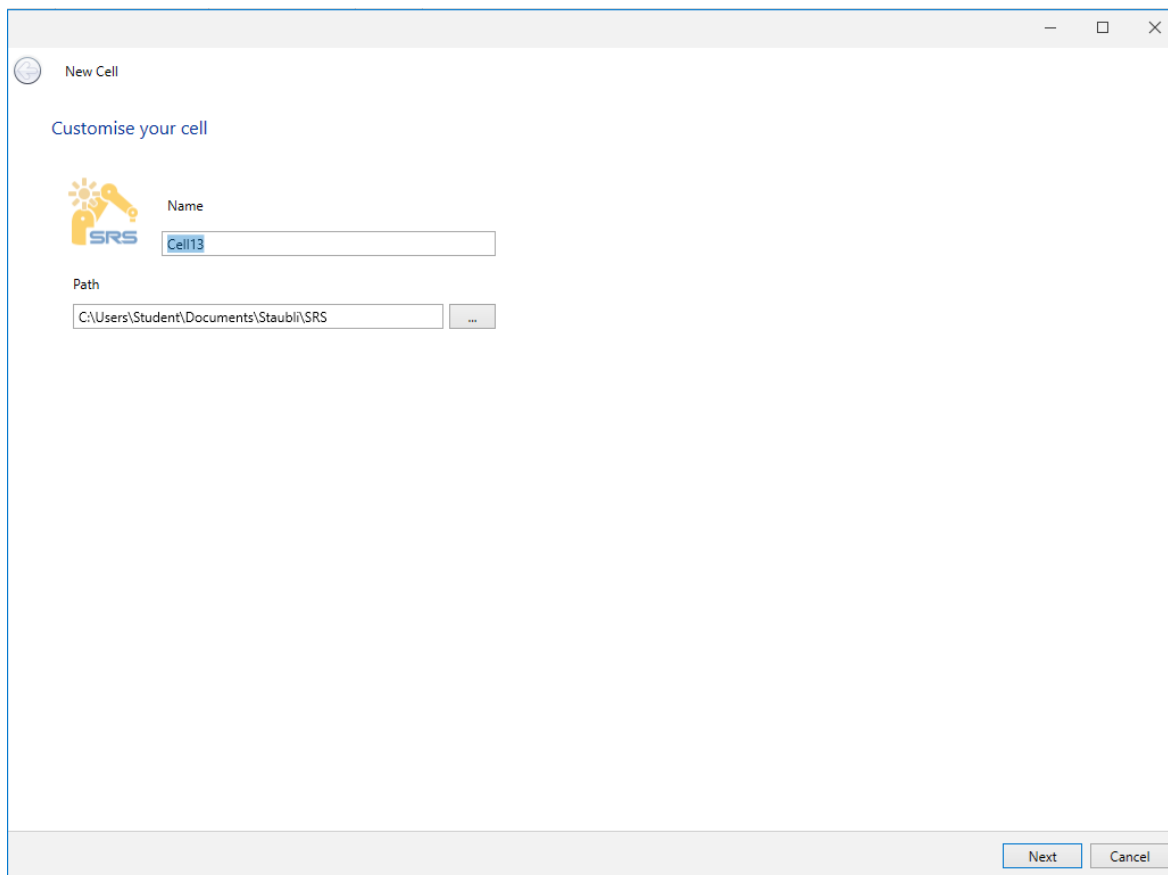
Pro práci s robotem prvně musíme založit nový projekt ve kterém budeme pracovat s robotem, u kterého pracujeme. Jeden projekt může mít spustu různých aplikací.

Pro založení nového projektu ihned po zapnutí programu SRS 2019 vybereme v levém bočním sloupci *New* a zvolíme *New cell wizard*.



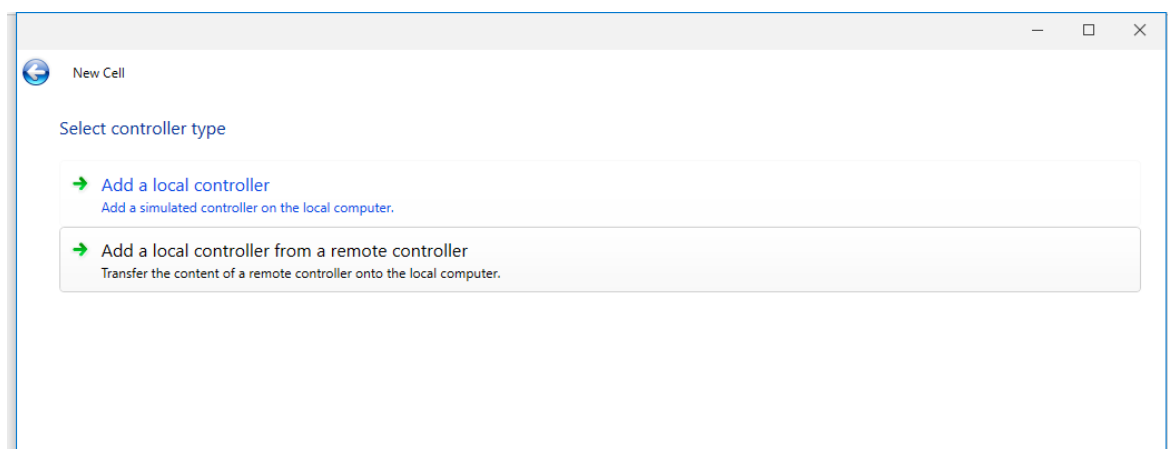
Obrázek 15 Nový průvodce vytvoření kóje

Poté si v tabulce *New Cell* vyplníme název a popřípadě změníme cestu pro přístup k projektu.



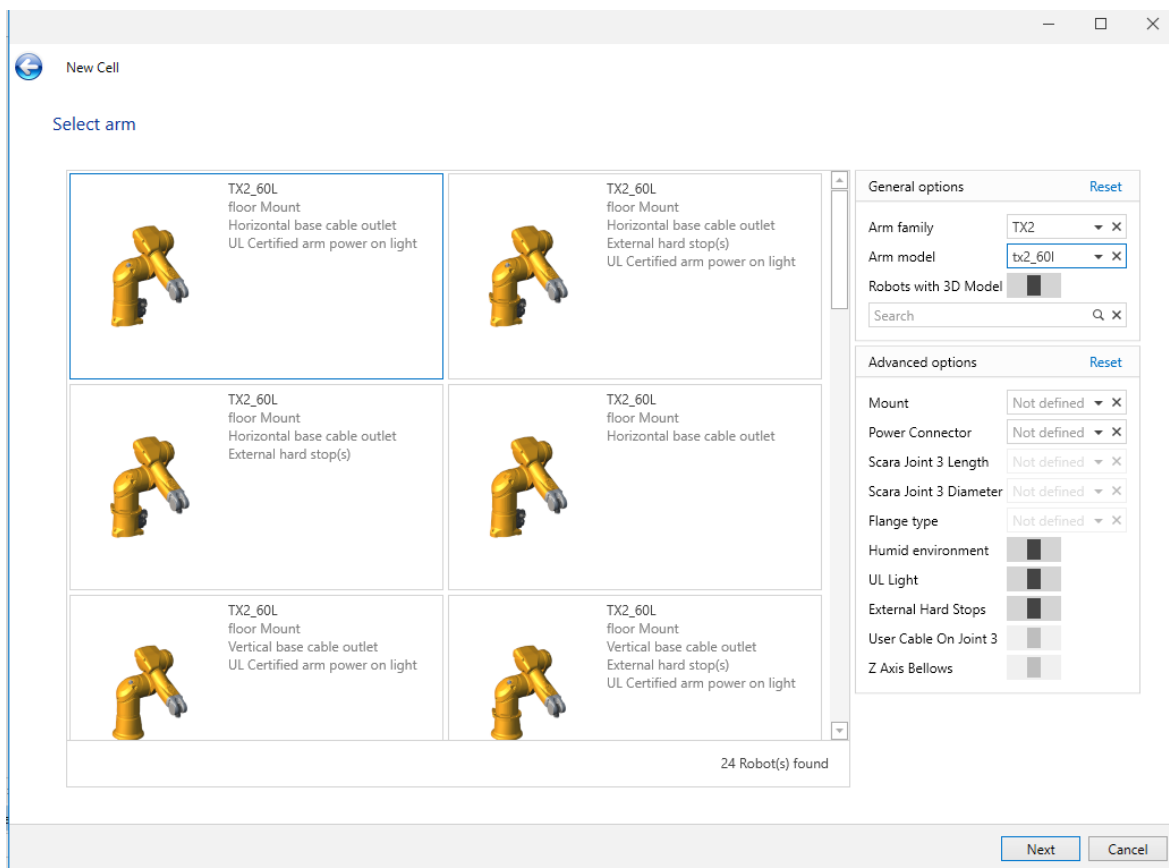
Obrázek 16 Přejmenování kóje

V dalším okně vybereme *Add a local controller*. Druhou možnost zvolíme pouze pokud máme reálného robota a chceme nahrát jeho controller.



Obrázek 17 Typ ovladače

Dále se přesuneme na výběr typu robota, kterého chceme připojit ke controlleru. V případě práce na robotu TX2-60 v *Arm family* volíme TX2 a v *Arm model* zvolíme tx2_60l.

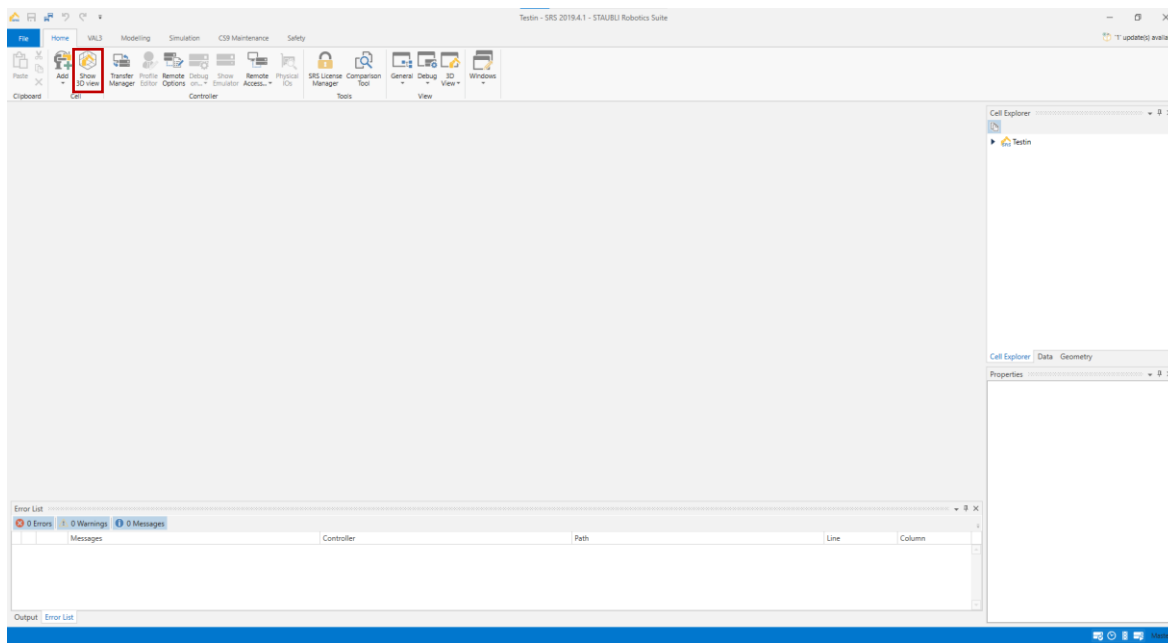


Obrázek 18 Výběr robota

Dále se otevře okno, kde pojmenujeme controller a můžeme změnit jeho verzi. Poté v sekci *Valves* vybereme *Bistable*. Zmáčknutím tlačítka next se přesuneme na nové okno, kde můžeme zkontrolovat správné nastavení. Po překontrolování, pomocí tlačítka Finish dokončíme zakládání projektu.

10.3 Práce v prostředí SRS

Po založení projektu musíme prvně zapnout 3D vizualizaci. To provedeme V sekci *Home* kliknutím na *Show 3D view*.



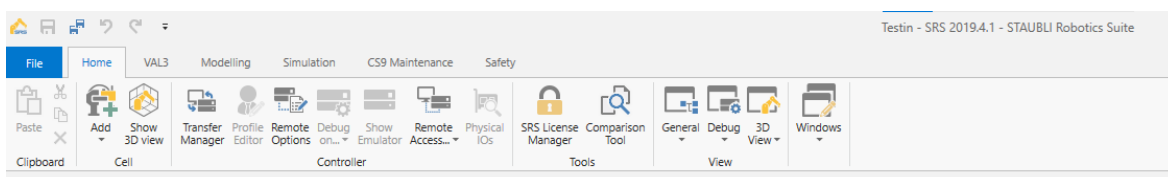
Obrázek 19 3D view

Po zobrazení 3D vizualizace se můžeme pohybovat prostorem:

- Podržením kolečka na myši a pohybem se jednoduše posouváme
- Točením kolečka se přibližujeme či oddalujeme
- Držením klávesy Alt a podržením levého tlačítka na myši a jeho pohybem si pohled rotujeme.

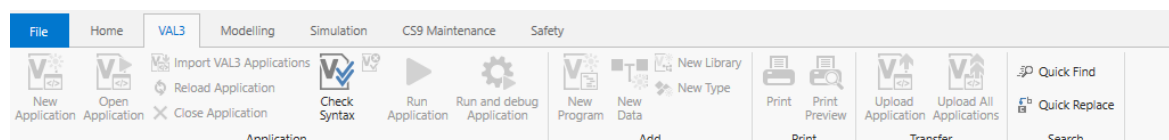
10.3.1 Karty v SRS

Jako první otevřená karta bude vždy *Home*. Na této kartě jsou základní možnosti pro práci s projektem. Jedna možnost je *Add*, kterou můžeme přidat druhého robota. Možnost *show 3D view*, kterou jsme již použili a *Transfer manager*, pomocí kterého můžeme buď nahrát program do reálného robota, nebo vytvořit zálohu robota.



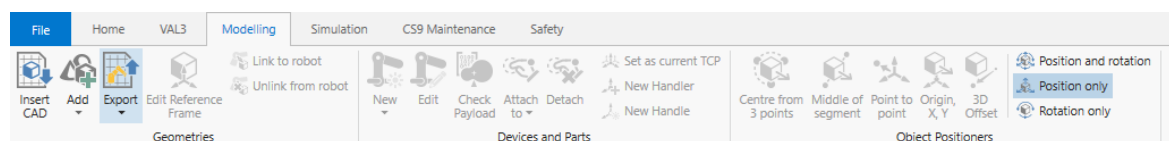
Obrázek 20 Karta Home

Další karta je VAL3, kde se pracuje s programem robota. Mezi funkce na této kartě patří vytvoření a otevření nové aplikace, kontrola syntaxe, spuštění aplikace a vytvoření nebo spuštění nového programu. Nachází se zde také funkce pro vytvoření nových knihoven.



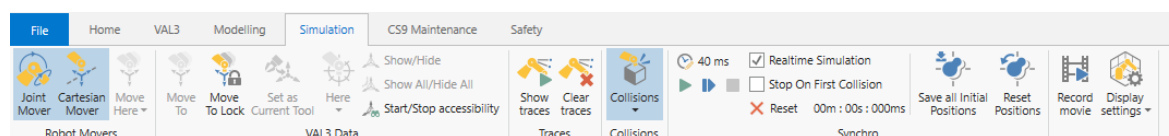
Obrázek 21 Karta VAL3

V kartě *Modelling* pracujeme s vytvářením nebo importováním různých objektů. Volbou *Add* můžeme vybrat geometrický obrazec (krychle, válec) a přidat jej do 3D prostoru. Výběrem *New* poté můžeme nastavit objekt jako součástku, nebo nástroj robota. Sloupec *Object Positioners* nám umožňuje polohovat body například na prostředek od 3 bodů.

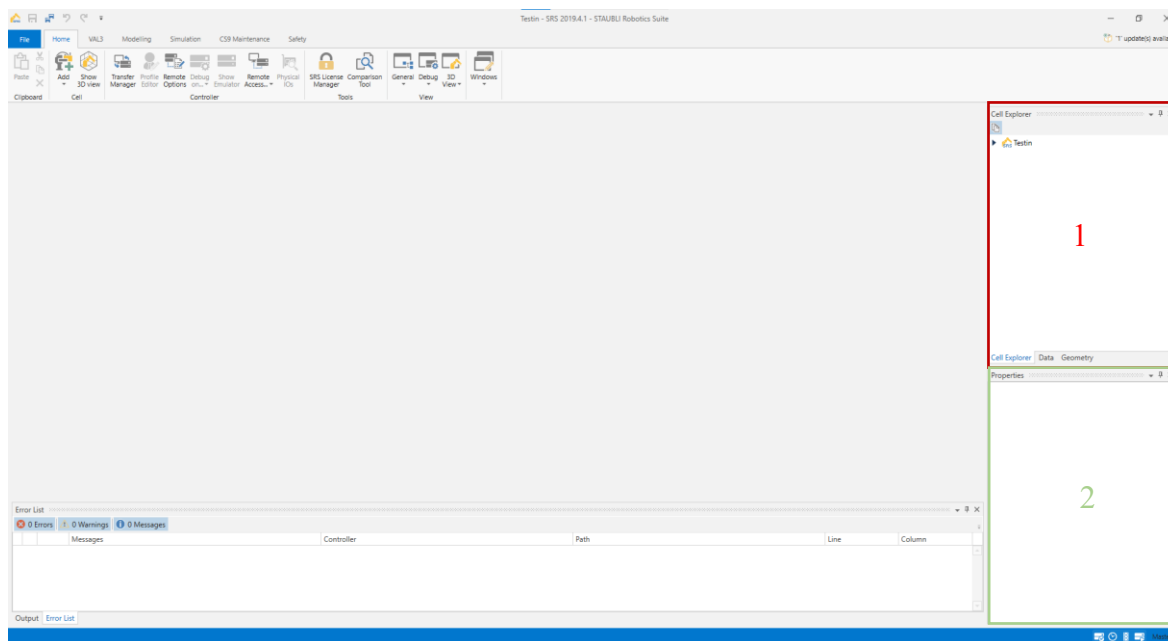


Obrázek 22 Karta Modelling

V kartě *Simulation* se nachází možnosti zobrazení typů pohybu buď pomocí kartézských souřadnic, nebo pomocí úhlového natočení kloubů robota. Ve sloupci *Synchro* lze zeleným tlačítkem spustit synchronizaci se spuštěným programem. *Save all initial Positions* a *Reset Positions* slouží k nastavení a resetování pozic objektů, kterými se v programu pohybuje.

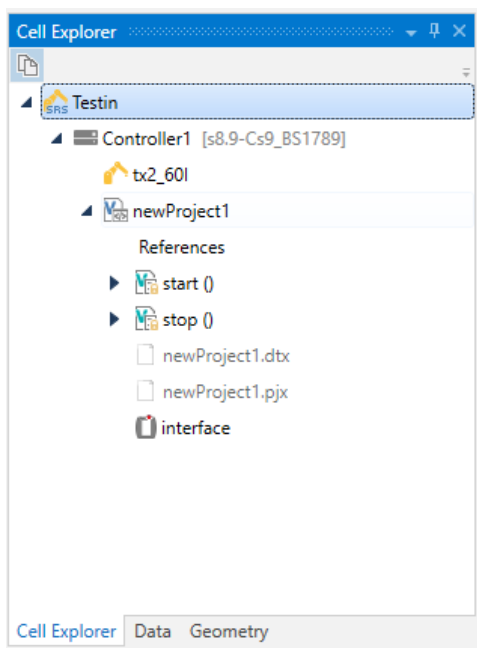


Obrázek 23 Karta Simulations

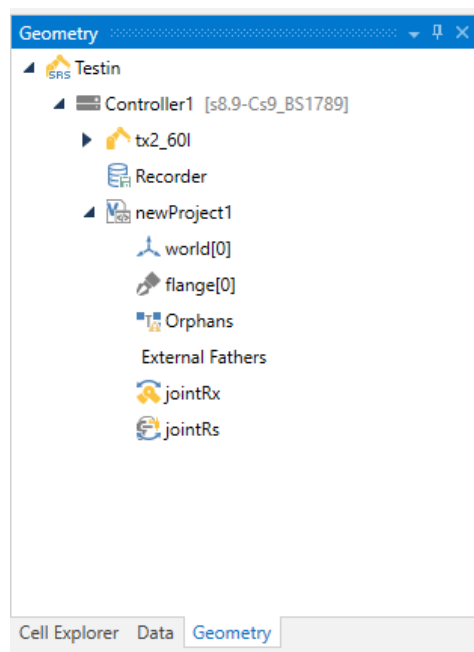


Obrázek 24 Boční panely

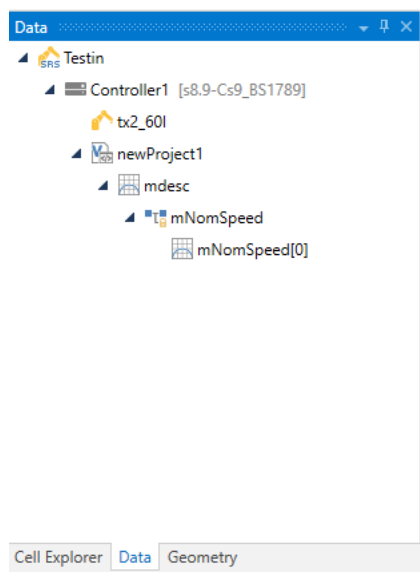
V panelu označeném 1 jsou 3 záložky. Záložka *Cell explorer* obsahuje knihovny, uživatelské aplikace a podprogramy. V záložce *Data* se nachází proměnné, které jsou využívány buď v programu nebo v interface. A v záložce *Geometry* najdeme uživatelem deklarované body a souřadné systémy.



Obrázek 26 Cell Explorer

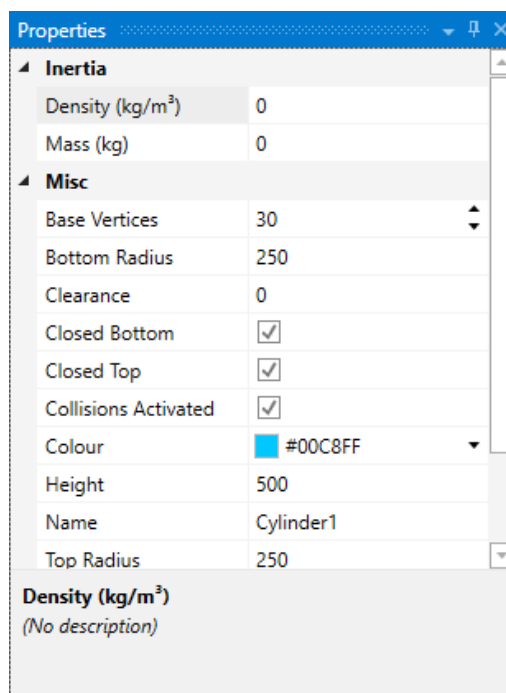


Obrázek 25 Geometry



Obrázek 27 Data

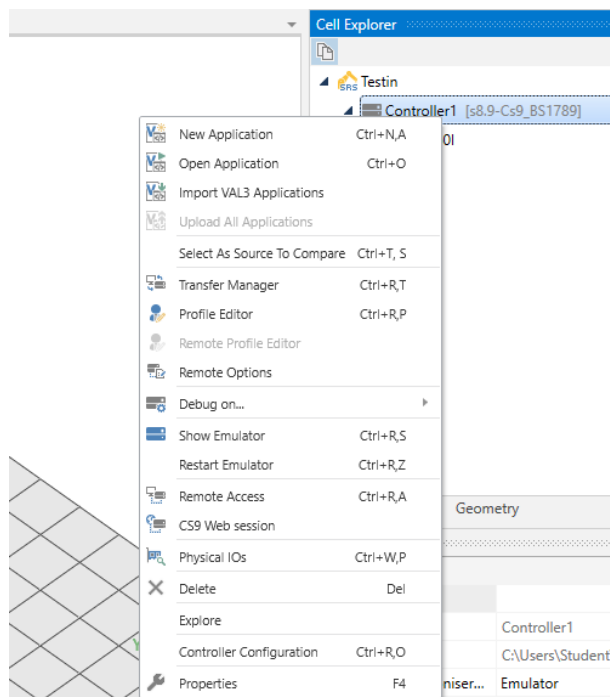
V panelu označeném 2 nalezneme vlastnosti označeného objektu.



Obrázek 28Vlastnosti

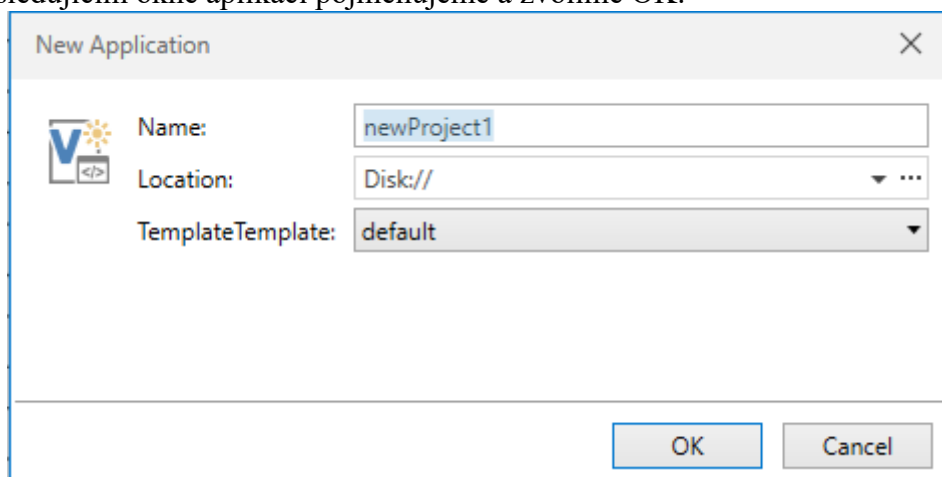
10.3.2 Založení nové aplikace

Novou aplikaci založíme kliknutím pravého tlačítka myši na Controller1 a vybereme *New Application*.



Obrázek 29 Založení nové aplikace

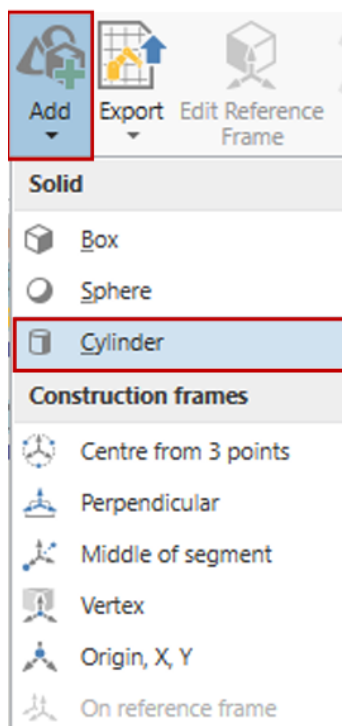
V následujícím okně aplikaci pojmenujeme a zvolíme OK.



Obrázek 30 Pojmenování nové aplikace

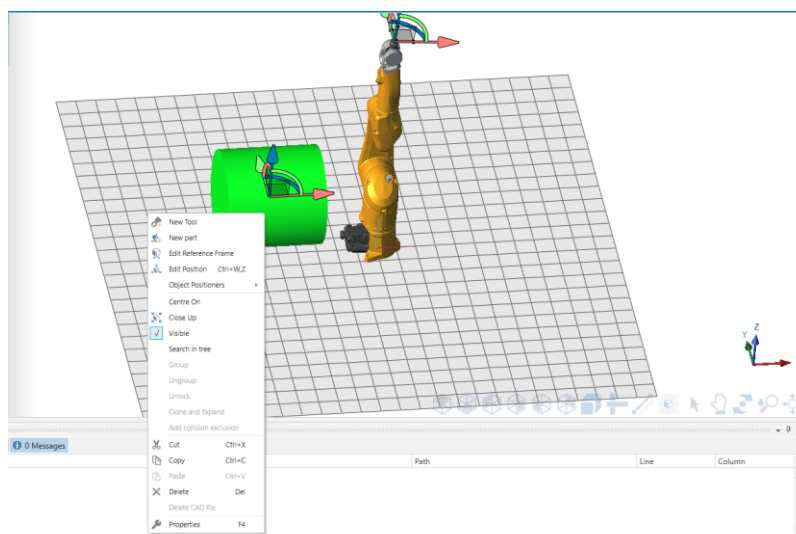
10.3.3 Vytvoření nástroje

V SRS máme 2 možnosti, jak přidat Nástroj. Pokud máme k dispozici 3D model nástroje přímo na robotu tak jej vložíme v kartě *Modelling* pomocí ikony *Insert CAD* anebo pokud model nemáme vytvoříme si náhradu pomocí *Add Cylinder*. Při obou verzích poté postupujeme stejně.



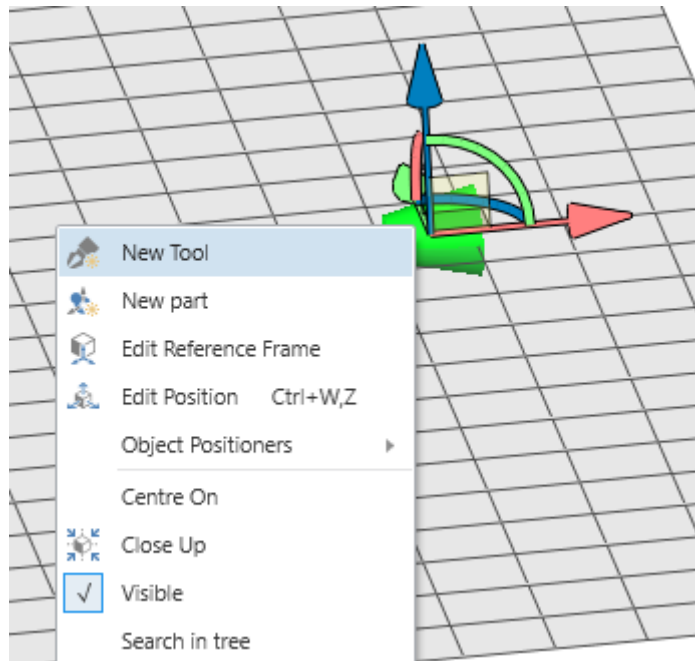
Obrázek 31 Vytvoření válce

Dále pomocí pravého kliknutí na válec se objeví menu s různými akcemi. Nyní zvolíme *Properties*. Tyto vlastnosti lze také najít v pravém dolním rohu obrazovky při kliknutí na válec.



Obrázek 32 Selektce při RCM na objekt

Poté v *Properties* změníme výšku válce na 100 mm, *Top Radius* na 50 mm a *Bottom Radius* na 25 mm. Dále pravým stisknutím myši se zobrazí nabídka a vybereme *New Tool*.



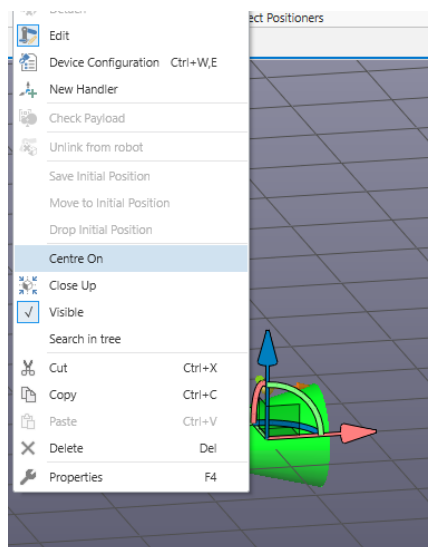
Obrázek 33 New Tool

Dále v levé části obrazovky najdeme strom, kde pod Tools najdeme náš nový nástroj. Nyní potřebujeme nastavit handle a handler.

Handle – Je to místo které propojuje robota s nástrojem

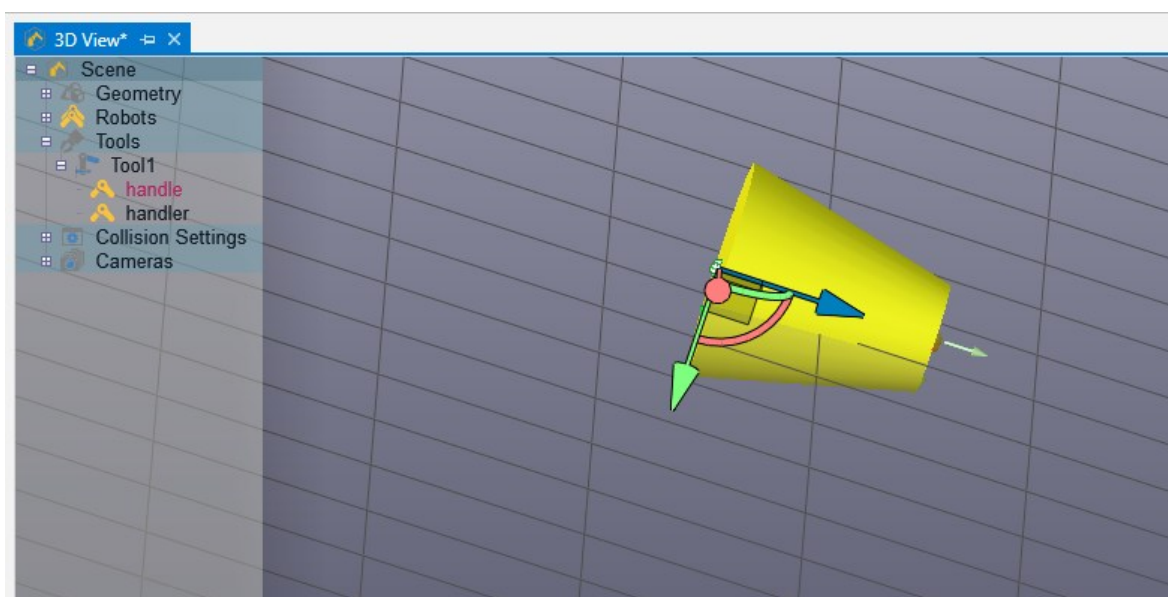
Handler – Je to místo, kde nástroj interaguje se součástkami.

Tyto části nastavíme tak, že si označíme nástroj a v kartě *Modelling* ve sloupci *Devices and Parts* vybereme *Edit*. Pro jednodušší otáčení si nastavíme náš nástroj jako střed rotace, a to pravým klikem a vybráním *Center On*.



Obrázek 34 Centre on

Nyní v levém stromě vybereme Handle a v kartě *Modelling* vybereme *Center from 3 points*.
Dále označíme 3 body po obvodu větší základny.

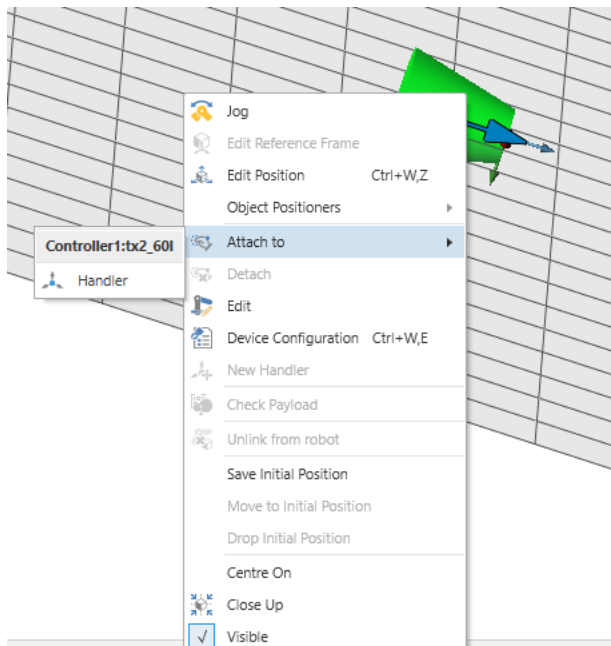


Obrázek 35 Handle

Pozor na to, aby modrá šipka (osa z) směřovala směrem do nástroje!

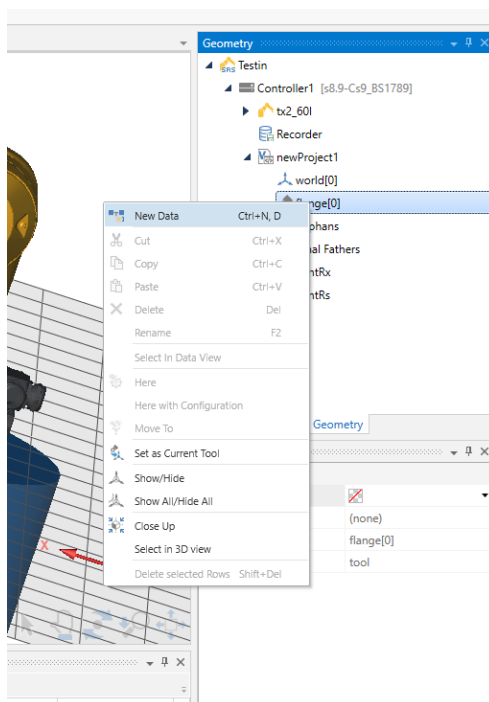
Dále označíme Handler a postupujeme stejně jako u Handle akorát označujeme body po obvodu menší základny. Tady by osa z měla směřovat od nástroje.

Poté již můžeme nástroj připevnit k robotovi stisknutím pravého tlačítka na nástroj a vybrat možnost *Attach to a Handler*



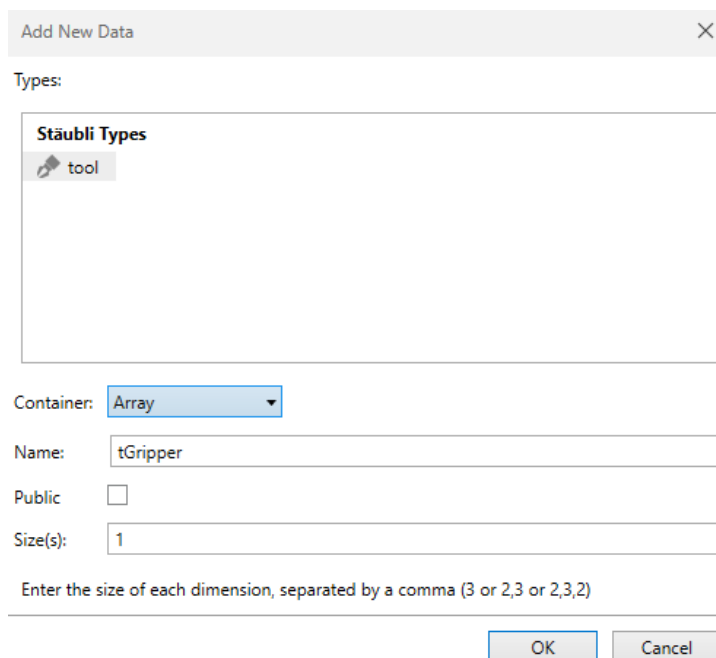
Obrázek 36 Připojení nástroje na robota

V kartě Geometry na levé straně obrazovky vytvoříme nový typ nástroje pravým kliknutím myši na *flange*, kde zvolíme *New Data*.



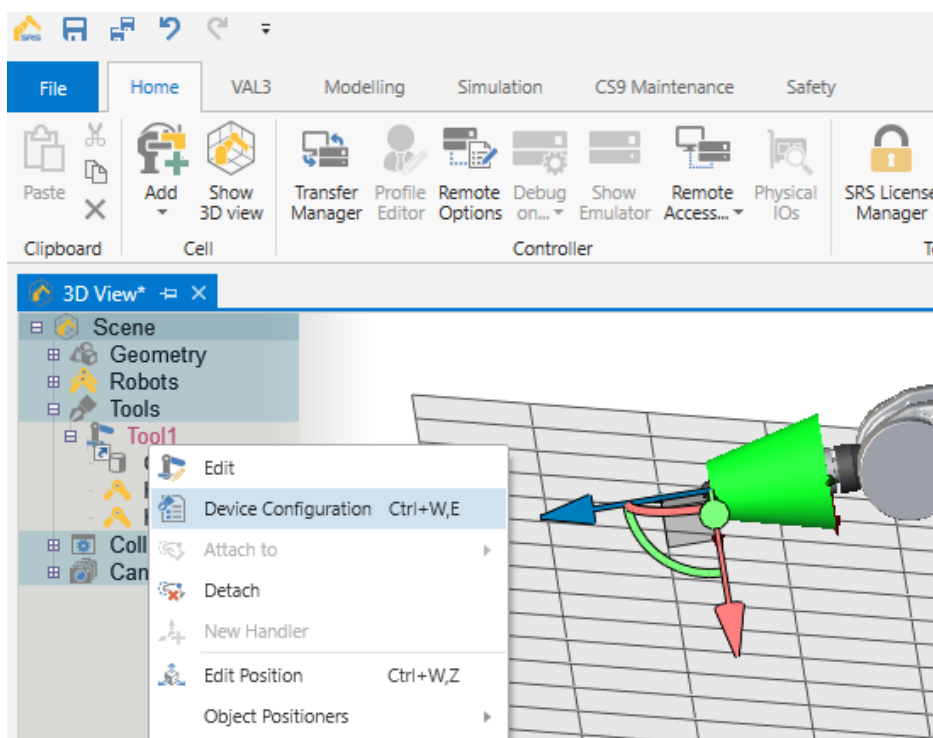
Obrázek 37 Nový typ nástroje

Dále si nástroj pojmenujeme. Důležité je, aby první písmeno názvu bylo malé t.



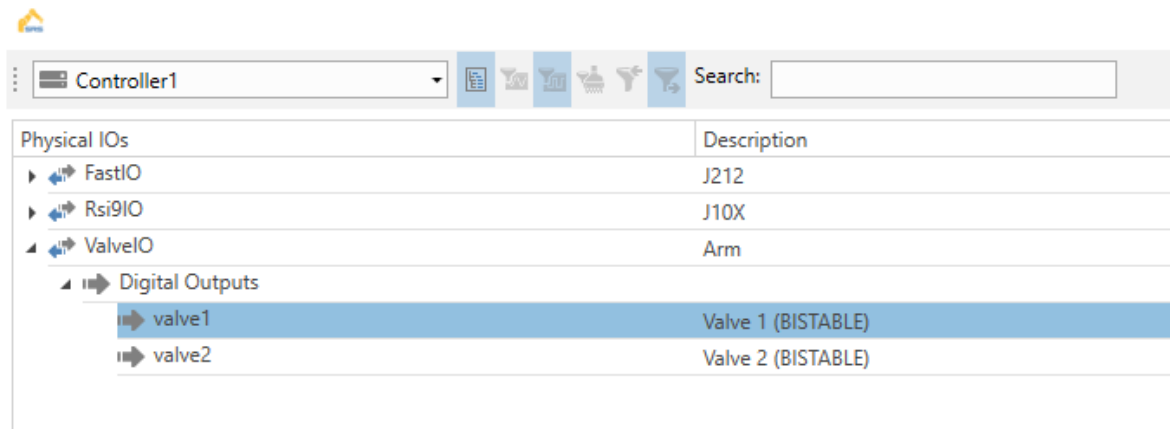
Obrázek 38 Pojmenování nástroje

Nyní v levé části obrazovky klikneme pravým tlačítkem myši na *Tool1* a vybereme *Device Configuration*.



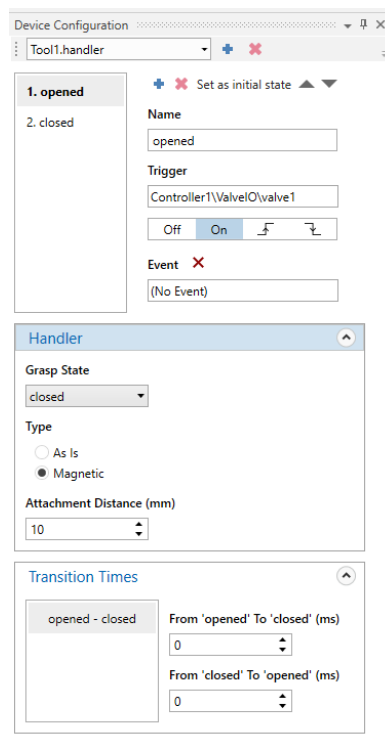
Obrázek 39 Device Configuration

V okně *Device Configuration* klikneme na + a otevře se nám okno kde nastavíme trigger nástroje. Do cesty zadáme/vybereme *Controller/ValveIO/Digital Outputs/valve1*.



Obrázek 40 Cesta Device Configuration

Dále se otevře další okno *Device Configuration*, kde pouze změníme *Type* na *As Magnetic*.



Obrázek 41 Zapnutí magnetického gripperu

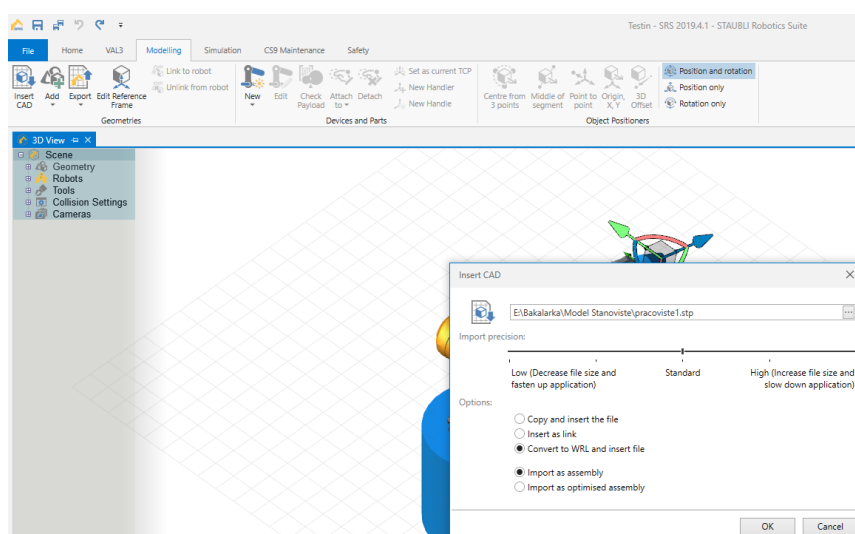
Poté vlevo ze stromu objektů rozklikneme *Tools* dále *Tool1* a pravým klikneme na *handler* a vybereme *Copy Tool Data*. Dále vpravo v záložce *Geometry* dvojklikem otevřeme nastavení nástroje a vložíme tam nakopírovaná data. Ve sloupci *Gripper* vybereme dvojklikem cestu k ovládání gripperu. V našem případě cesta bude *Controller/ValveIO/Digital Outputs/valve1*.

newProject1-tGripper Data* 3D View*									
Index	X	Y	Z	Rx	Ry	Rz	Gripper	OTime	CTime
0	0	0	100	0	0	0	ValveIOValve1	0	0

Obrázek 42 Nastavení rozměrů Gripperu

10.3.4 Vložení CAD souboru

Pokud máme CAD model, který bychom chtěli vložit do 3D prostoru, tak pouze v kartě *Modelling* vybereme možnost *Insert CAD* a najdeme jeho umístění na počítači. Dále máme možnost nastavení, s jakou přesností se model naimportuje a jakým způsobem se vloží.



Obrázek 43 Vložení CAD modelu

Některé modely se mohou skládat z více komponentů. Tyto jednotlivé komponenty lze uvolnit od zbytku 3D modelu pravým kliknutím na ně a zvolení možnosti *Unlock*.

10.3.5 Vytvoření objektů

Pro vytvoření součástek pro simulaci vybereme v kartě Modelling *Add* a *Cylinder*. V properties válce nastavíme oba jeho radiusy na 30 a výšku na 60. Dále jej natočíme tak, aby byl válec natočen svisle. Dále pravým kliknutím na válec vybereme *New* a dále možnost *Part*. Tím se válec nastaví jako součástka a automaticky se vytvoří její handle.

10.3.6 Vytvoření Bodů

V SRS jsou podle druhu robota použitého (řada TS nebo TX) využívány jiné typy bodů. Správné použití bodů je následující:

JointRX – kloubové souřadnice pro TX2 - 60

PointRX – kartézské souřadnice pro TX2 - 60

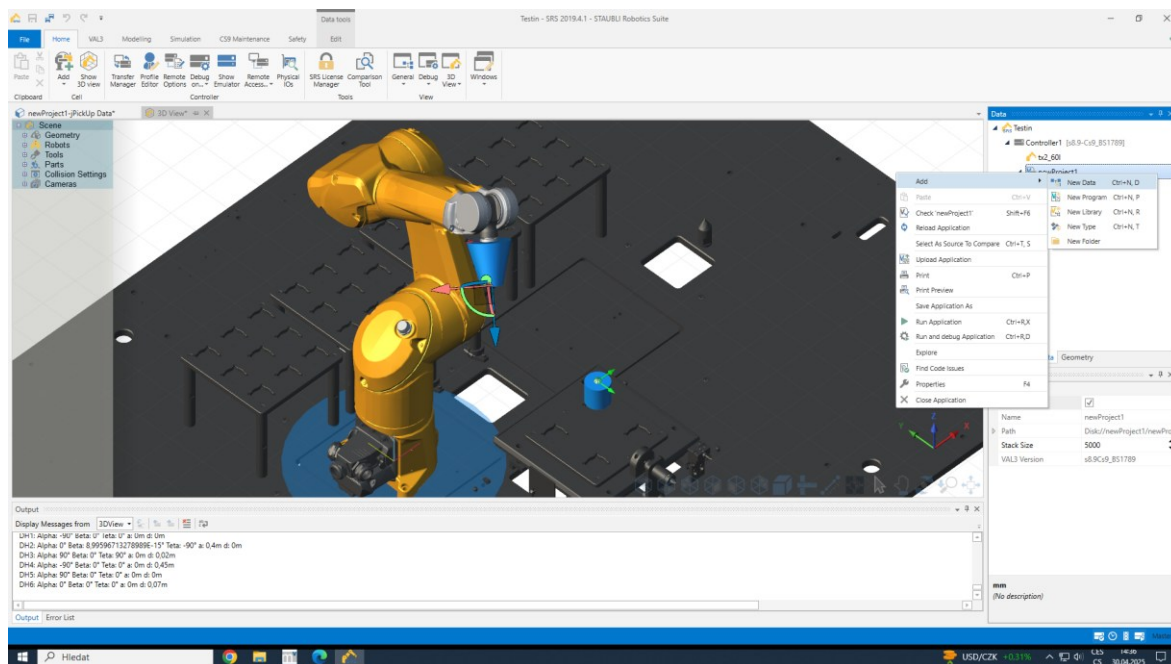
JointRS – kloubové souřadnice pro TS2 - 60

JointRX – kartézské souřadnice pro TS2 - 60

Vytvoření bodů můžeme realizovat dvěma způsoby, buď body nadefinujeme v simulačním prostředí, nebo body definujeme na reálném robotu.

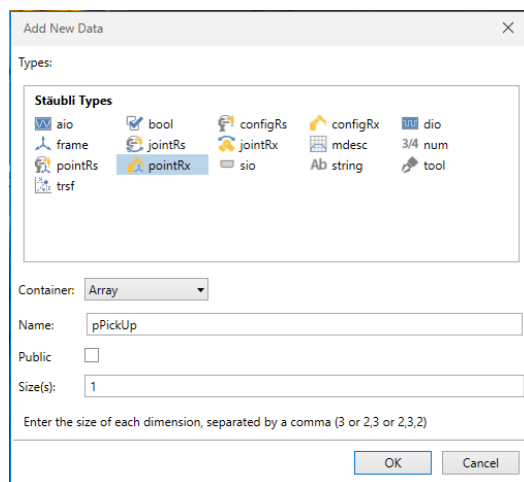
A) Body vytvořené v simulačním prostředí

Stisknutím pravým tlačítkem na naši Aplikaci (karta Cell Explorer na pravé straně obrazovky) dále vybereme *Add a New Data*.



Obrázek 44 Nový bod

Poté vybereme typ bodů, co zrovna potřebujeme. V tomhle případě použijeme PointRX a pojmenujeme jej. Jména bodů musí vždy začínat malým písmenem popisující typ bodů (malé p pro PointRX a malé j pro JointRX).

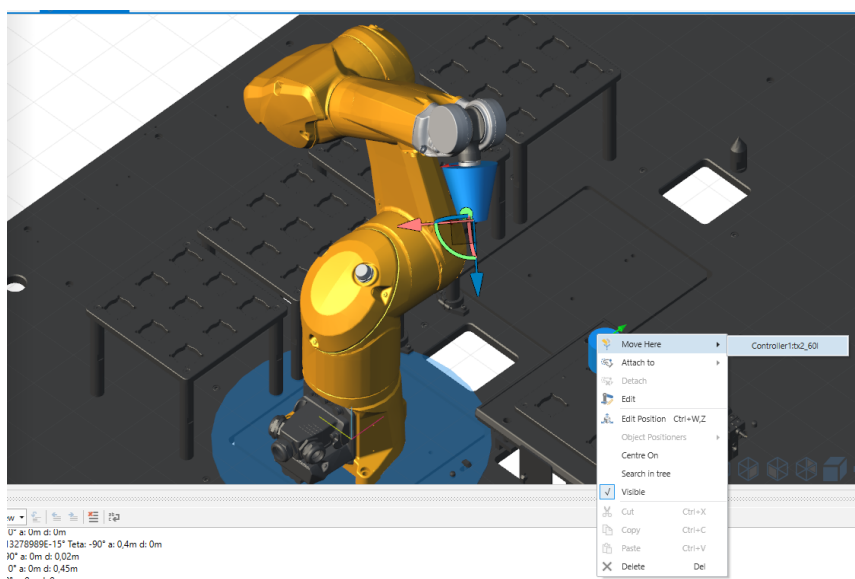


Obrázek 45 Pojmenování bodu

Po vytvoření pole bodů jej můžeme najít v pravo v kartě *Geometry* > *World* > *PointRX*, nebo v kartě *Data* > *Applikace* > *PointRX*.

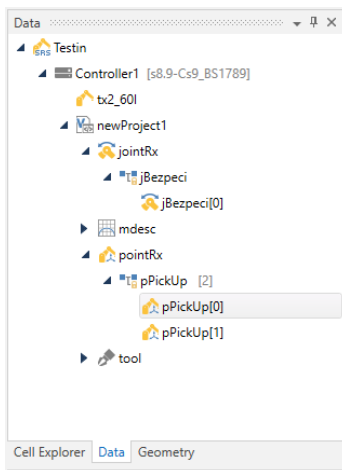
Jelikož je to pole proměnných, tak můžeme přidat pod jedno jméno více různých bodů. Poté v programu k nim budeme přistupovat pomocí hranatých závorek (pBod[0], pBod[1]).

Těd bodům nastavíme souřadnice. To lze opět udělat dvěma způsoby, a to buď přímým zápisem, nebo graficky pohneme robotickou rubou a pomocí tlačítka *Here* nastavíme souřadnice na bod handleru Gripperu. Pokud chceme, aby bod byl na handleru součástky, prvně robota přesuneme na součástku funkcí *Move Here*.



Obrázek 46 Move Here

Poté nastavíme souřadnice bodu pravým stisknutím na bod, co chceme a vybereme *Here*.

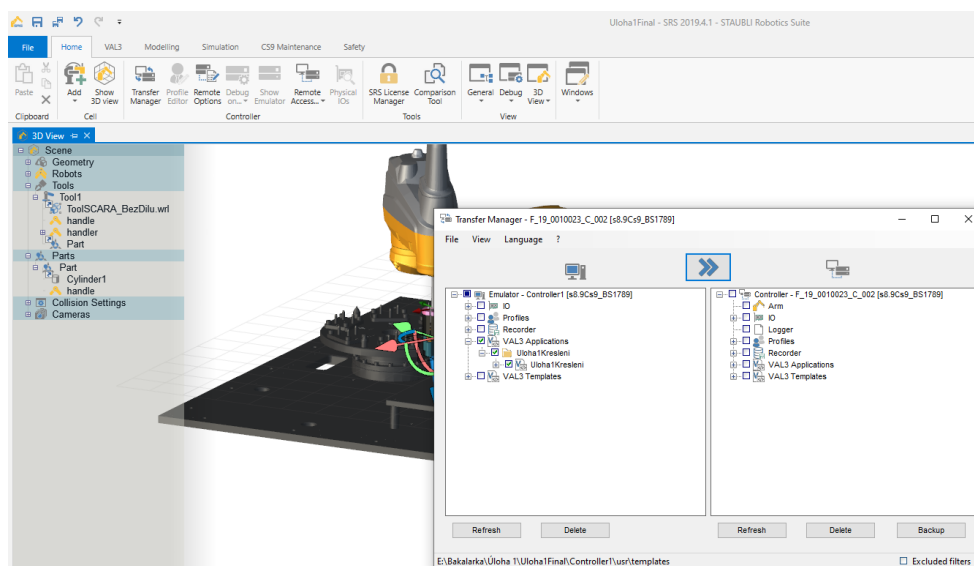


Obrázek 47 Nastavení bodu

B) Body vytvořené na reálném robotu

Pro definování souřadnic bodů na reálném robotovi prvně musíme mít založenou aplikaci a správně definovaný nástroj. Nástroj musí být definovaný, protože jinak by nesesedla výšková souřadnice Z.

Pokud máme založenou aplikaci a definovaný nástroj můžeme pomocí ikony *Transfer Manager* nahrát aplikaci do kontroléru robota. Robot musí být v manuálním režimu.



Obrázek 48 Transfer manager

Poté na flexpendatu robota postupujeme tak, že načteme naši aplikaci, a zmáčkne ikonu tužky (úprava programu). Dále buď vybereme, nebo vytvoříme bod, kterému chceme nadefinovat souřadnice a stiskneme ikonu pro pohyb robota.



Obrázek 49 Editování aplikace



Obrázek 50 Ikona pro pohyb

Otevře se nám stránka, kde můžeme pomocí tlačítek na flexpendantu pohybovat robotem. Vybereme správný typ pohybu robota (při Point to bude Frame a při Joint to bude Joint) a robota si nastavíme tam, kde chceme, aby byl bod a stiskneme ikonu *Here*. Pokud pracujeme s nástrojem musíme si dát pozor na správné nastavení nástroje (v poli pro nástroj musí být definovaný nástroj, ne flange).

Otevření a

zavření gripperu



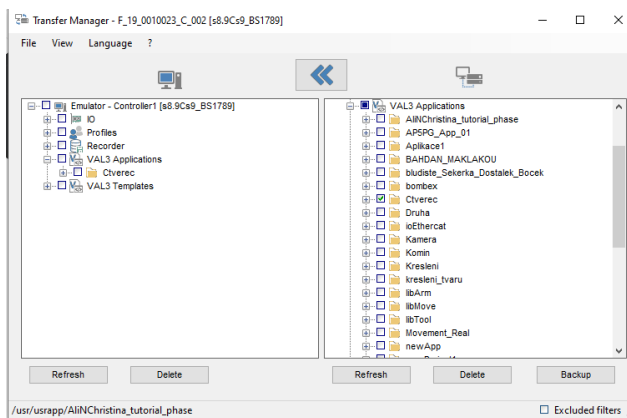
Pohyb robota

Obrázek 51 Nastavení bodu na robotovi

Poté co si nastavíme všechny body aplikaci uložíme a pomocí transfer manager ji načteme zpět do počítače nyní už s definovanými body. Abychom na počítači viděli body musíme znovu načíst aplikaci. (Stisknutí pravého tlačítka na aplikaci a Reload Application)



Obrázek 53 Uložení aplikace

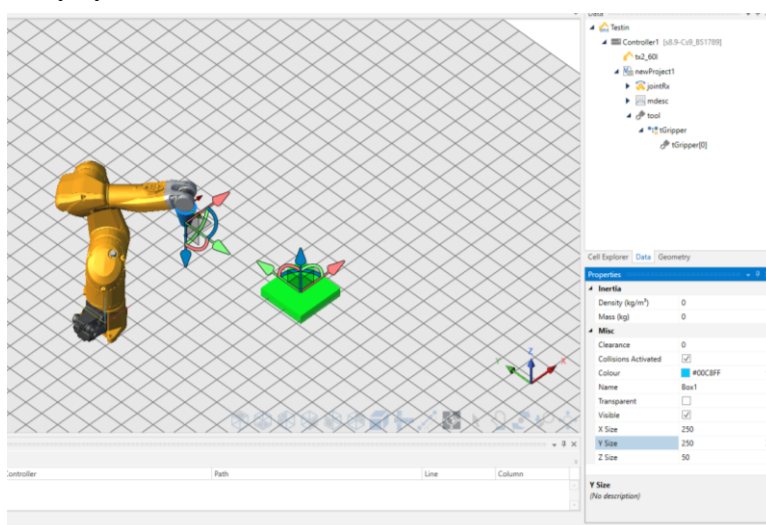


Obrázek 52 Transfer zpět

10.3.7 Vlastní souřadný systém

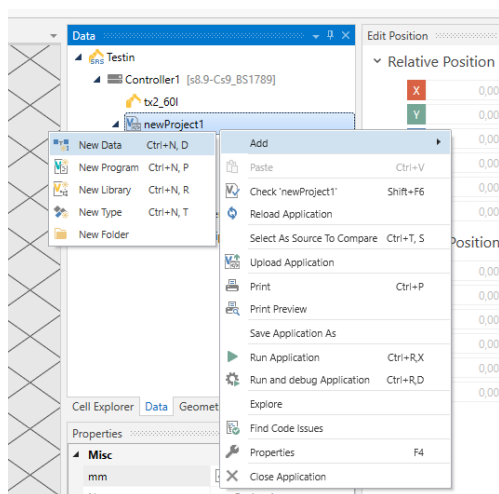
Vlastního souřadného systému využíváme pro snadnější pohyb například na paletě. Jelikož body jsou vztaženy na souřadný systém, pokud dojde k posunu nebo natočení palety stačí pouze předefinovat 3 body souřadného systému a robot bude jezdit znovu správně.

Prvně si namodelujeme paletu pomocí *Add* a upravíme vložený kvádr. Výška bude 50 mm, a obě strany budou mít délku 250 mm. Tuto paletu modelujeme pouze, abychom měli na co vlastní souřadný systém vztáhnout.



Obrázek 54 Paleta

Dále pravým kliknutím na naši aplikaci zvolíme *Add* a *New Data*. Z typů dat vybereme typ *frame* a pojmenujeme si jej.

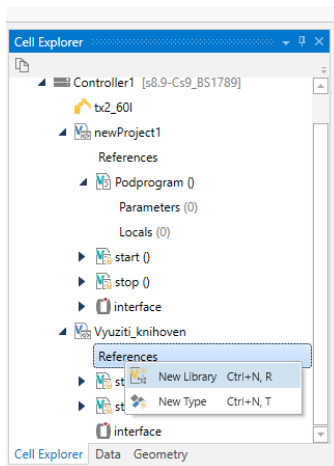


Obrázek 55 Přidání souřadného systému

Poté pravým kliknutím na vytvořený *frame* vybereme *Select in 3D View*. Poté pomocí ikony *Origin X,Y* v kartě *Modelling* nastavíme souřadnicový systém palety. Jakmile máme definovaný vlastní souřadný systém body do něj přidáme pravým kliknutím na nadefinovaný *frame*, zvolíme možnost *New Data* a vybereme typ bodu, který potřebujeme.

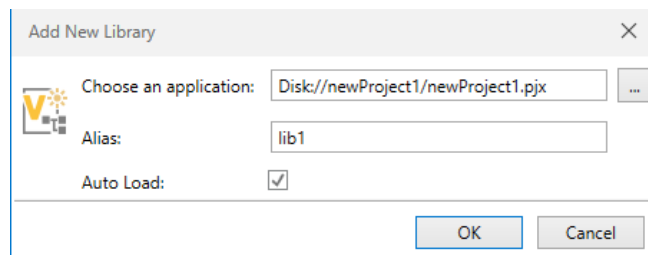
10.3.8 Tvorba a využití knihoven

Knihovna je vlastně nahrání všech dat z jiné aplikace, která jsou označena jako *Public*. Pro demonstraci využijeme předchozí ukázky vlastního souřadného systému. Tedy data, které chceme přenést jako knihovnu musíme označit jako *Public* a to provedeme pravým kliknutím na ně a volbou možnosti *Public*. Abychom mohli využít knihovnu musíme prvně založit novou aplikaci, kde knihovnu importujeme. Poté v nové aplikaci klikneme pravým tlačítkem na *References* a vybereme *New Library*.



Obrázek 56 Založení nové knihovny

Dále se otevře okno, kde vybereme cestu k aplikaci, kterou chceme nainportovat jako knihovnu a knihovnu pojmenujeme.



Obrázek 57 Pojmenování nové knihovny

Nyní již můžeme pracovat s daty přes novou knihovnu. V kódu mají tyto data svůj vlastní prefix který je název knihovny, ze které jsou.

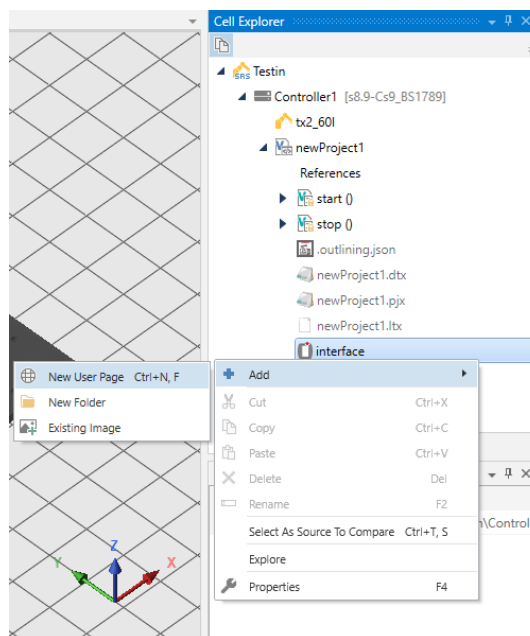
```
1 begin
2   movej(lib1:jBezpeci,lib1:tGripper,mNomSpeed)
3 end
```

Obrázek 58 Využití proměnných z knihovny

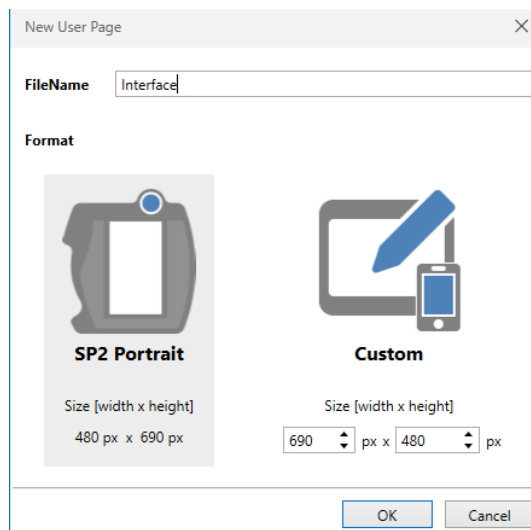
10.3.9 Založení vlastního uživatelského rozhraní a práce s ním

Pokud chceme, zasahovat do programu pomocí uživatelského prostředí musíme jej založit, navrhnout vzhled a pro ovládání veličin jej propojit s danými veličinami.

Nové uživatelské prostředí založíme tak, že v pravém okraji v kartě Cell Explorer klikneme pravým tlačítkem na Interface a zvolíme *Add New User Interface*. Dále vyskočí okno, kde si nové rozhraní můžeme pojmenovat a zvolit jeho velikost.

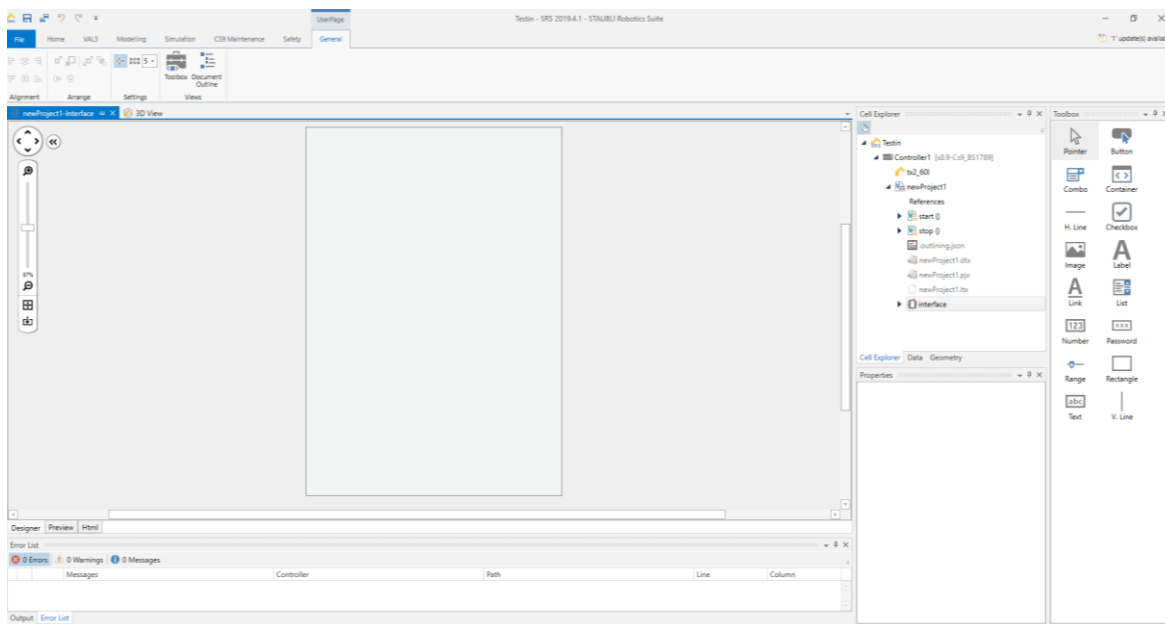


Obrázek 59 Založení nového uživatelského rozhraní



Obrázek 60 Volba velikosti a jména rozhraní

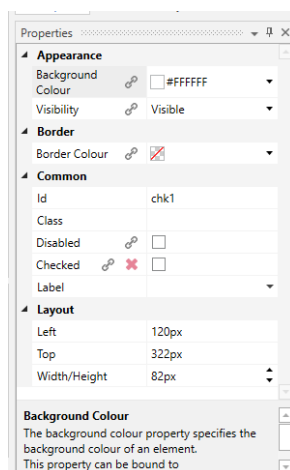
Po volbě rozhraní se v pravé části obrazovky zobrazí menu s prvky, které můžeme na rozhraní vložit.



Obrázek 61 Rozhraní a možnosti prvků

Každý z těchto prvků má své vlastnosti, kde můžeme změnit barvu nebo velikost. Některé prvky lze také spárovat s veličinami v projektu a to buď pouze jako pro čtení, tedy že bude číst hodnotu z programu a zobrazovat ji, nebo jako zadávací pole, kde zadáme hodnotu, se kterou bude program pracovat.

Pro propojení prvků slouží u propojitelných vlastností ikona řetězu, kterou když rozklidneme tak můžeme nastavit s jakou proměnnou je možný prvek propojit. Prvky lze také vypnout, aby se s nimi nemohlo pracovat, toho dosáhneme propojením proměnných s vlastností *Disabled* (prvek se vypne, když hodnota proměnné bude 1).



Obrázek 62 propojení prvků s proměnnými

10.3.10 Vytvoření programu

Program otevřeme tím, že v kartě *Cell Explorer* rozklikneme položku *start()*. Zde se píše kód co se má provést po startu aplikace.

Pro pohyb robota jsou podle druhu pohybu 3 základní funkce:

move – lineární pohyb z jedné kartézské souřadnice do druhé.

movej – rotační pohyb, kde se využívá kloubových souřadnic

movec – obloukový pohyb využívající kartézské souřadnice

Každý pohyb má své parametry, které se píšou do závorky za něj:

move(koncový bod, nástroj, rychlost pohybu)

movej(koncový bod, nástroj, rychlost pohybu)

movec(bod, kterým má projít, koncový bod, nástroj rychlost pohybu)

Pro zjednodušení práce a zkrácení kódu pracujeme s cykly a podmínkami:

- **Cyklus While**

```
While (podmínka)
<kód co se má provést>
endWhile
```

- **Cyklus For**

```
for index = 0 to 9 step 1
<kód co se má provést>
endFor
```

- **Cyklus Do .. until**

```
do
<kód co se má provést>
until (podmínka)
```

Dále se v programu využívají příkazy na ovládání gripperu:

```
Open(název Gripperu)
Close(název Gripperu)
```

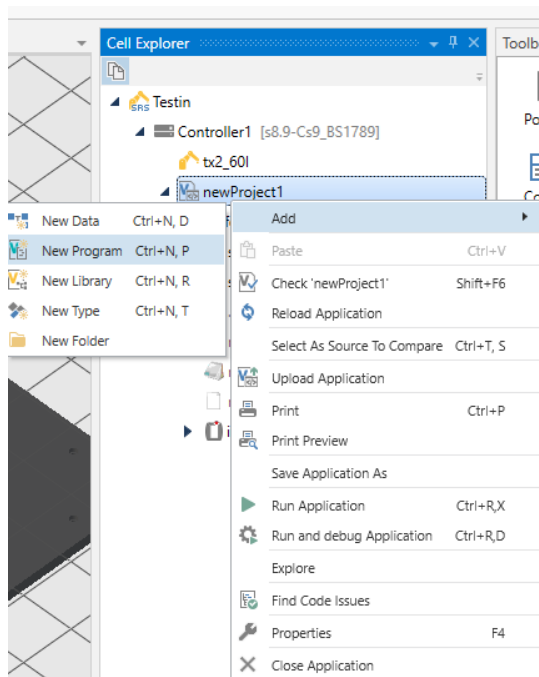
Pozor tyto příkazy jsou na reálném robotovi prohozeny, a proto se kód musí upravit před posláním na robota.

Dále pracujeme s dalšími příkazy jako:

```
WaitEndTask() - tento příkaz zajistí, že robot stihne dojet na bod
Delay(čas v sekundách) - tento příkaz slouží k čekání robota
v sekundách
```

10.3.11 Vytvoření podprogramu

Podprogram vytvoříme tak, že klikneme na Aplikaci pravým tlačítkem a zvolíme *Add* a *New Program* a nastavíme jej jako Public.



Obrázek 63 Nový podprogram

Do podprogramu můžeme napsat jaký kód chceme a poté jej můžeme zavolat v hlavním programu přes funkci *call()*.

```
1 begin
2 call Podprogram()
3 end
```

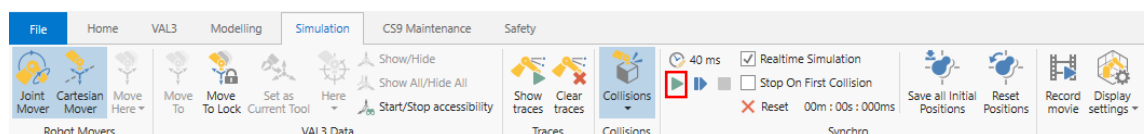
Obrázek 64 Zavolání podprogramu

10.3.12 Spuštění simulace

Pro demonstraci simulace vytvoříme jednoduchý program, který přesune robota do bezpečné pozice, z něj přijede, vyzvedne válec a poté jej znova položí. Na konci pohybu se robot vrátí do bezpečné pozice.

```
begin
  movej (jBezpeci, tGripper, mNomSpeed)
  open (tGripper)
  movel (pPickup[1], tGripper, mNomSpeed)
  movel (pPickup[0], tGripper, mNomSpeed)
  close (tGripper)
  movel (pPickup[1], tGripper, mNomSpeed)
  movel (pPickup[0], tGripper, mNomSpeed)
  open (tGripper)
  movel (pPickup[1], tGripper, mNomSpeed)
  movej (jBezpeci, tGripper, mNomSpeed)
  waitEndMove ()
end
```

Abychom spustili simulaci musíme prvně kód zkontrolovat, a to uděláme ikonou *Check Syntax* v kartě *VAL3*. Pokud nemáme žádné chyby tak můžeme spustit synchronizaci pomocí zelené šipky v kartě *Simulation*. Nad zelenou šipkou si také můžeme nastavit, jak plynule se simulace bude vykreslovat.



Obrázek 65 Zapnutí synchronizace

Poté spustíme aplikaci pravým kliknutím na naši aplikaci v záložce *Cell Explorer* a zvolením možnosti *Run Application*.

Po spuštění aplikace se otevře okno s emulací FlexPendantu:



Obrázek 66 FlexPendant

Pro spuštění simulace musíme zapnout motory robota, to se provede stisknutím tlačítka vpravo nahoře a že jsou motory zapnuty poznáme tím, že světlo pod tlačítkem začne modře blikat. Aby se robot hýbal musíme spustit aplikaci tlačítkem pod ním. Pokud je simulace v manuálním režimu tlačítko se musí držet a pokud je v automatickém režimu stačí jej pouze stisknout.

11 ÚLOHY PRO PROCVIČENÍ PRÁCE V SRS

11.1 Úloha 1 – práce s pohybem mezi body

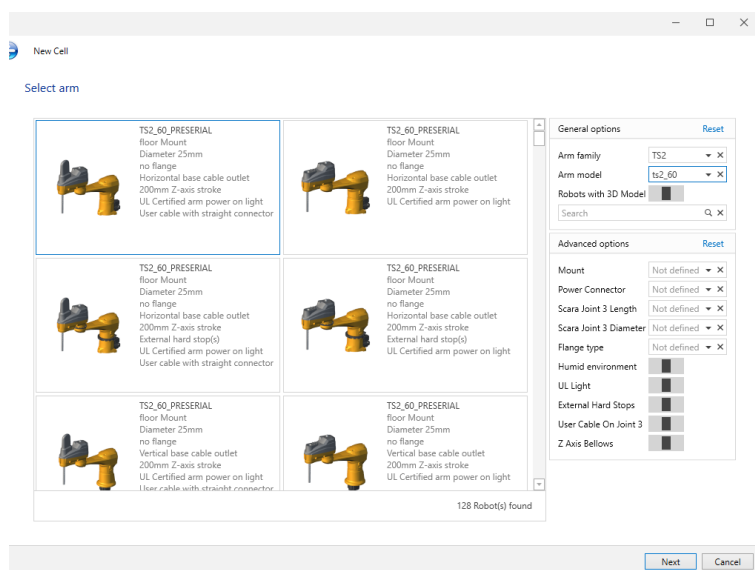
V úloze se budeme zabývat různými typy pohybu robota a jejich využitím.

11.1.1 Zadání úlohy 1

Na robotovi RS2-60 vytvořte program pro kreslení různých obrazců. Program by měl obsahovat lineární posun, posun po křivce a spojení obou posunů. Využijte nástroje v laboratoři. Body nastavte z robota. Prvně program simulujte a po konzultaci s vyučujícím následně vyzkoušejte na reálném robotovi.

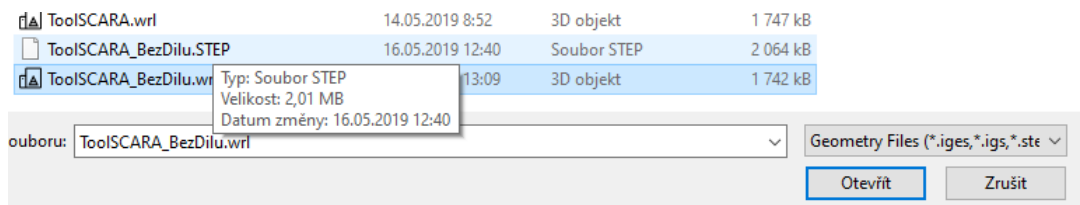
11.1.2 Provedení úlohy 1

Prvně si založíme novou Cell, kde jako robota vybereme RS2-60.



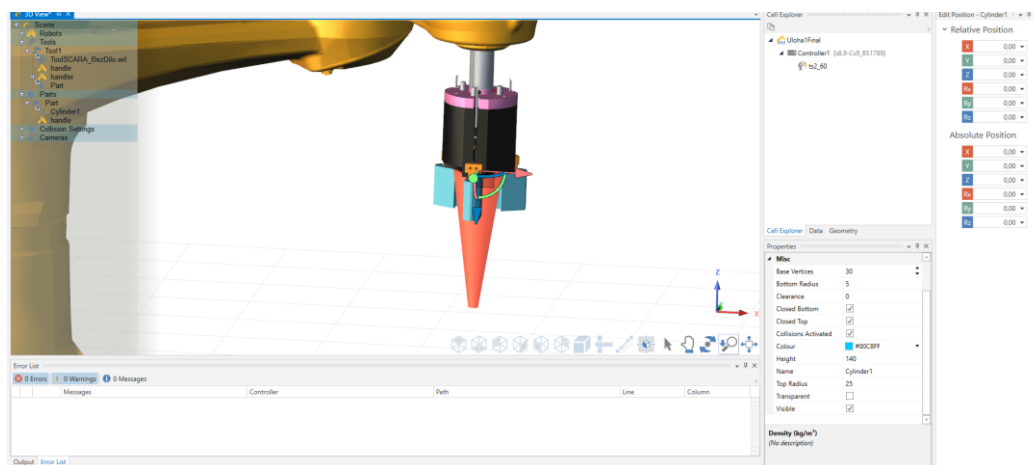
Obrázek 67 Výběr robota

Ihned založíme aplikaci. Poté si vložíme CAD model pracoviště. Přes Insert Cad vložíme také Gripper robota, který musíme správně nastavit (viz kapitola 10.3.3). Vybereme ToolSCARA_BezDilu.



Obrázek 68 Soubor Gripperu

Dále si musíme vytvořit simulaci součástky pro uchopení propisky, to uděláme tak, že si vložíme válec a upravíme jeho rozměry. Rozměry jsou 25mm *top radius*, 5mm *bottom radius* a výška 140mm. Poté jej nastavíme jako *Part* a využijeme *Attach to tool1 handler*.



Obrázek 69 Simulace součástky

Poté

si Aplikaci přeneseme na robota a nastavíme si body, které potřebujeme. (10.3.6.)

Budeme potřebovat 3 nájezdové body (každý pro jeden tvar), 1 bod pro bezpečnou pozici robota (jointRS) a body pro provedení tvarů. V našem případě 4 body pro obdélník, 4 body pro elipsu a 6 bodů pro tvar srdce.

Jakmile máme body na všechny obrazce přeneseme aplikaci zpět a můžeme začít psát program. Aby se body správně ukázaly na počítači musíme znovu načíst aplikaci pomocí tlačítka *Reload Application*.

Vytvoříme si tolik podprogramů, kolik chceme kreslit obrazců, tedy v našem případě 3. Programy pojmenujeme podle jejich tvarů jako Ctverec, Oblouk, Srdce.

Jednotlivé podprogramy budou vypadat následovně:

#program pro čtverec

```
begin
  movej (jBezpeci, tNastroj, mNomSpeed)
  movel (pNajezdy, tNastroj, mNomSpeed)
  movel (pCtverec[0], tNastroj, mNomSpeed)
  movel (pCtverec[1], tNastroj, mNomSpeed)
  movel (pCtverec[2], tNastroj, mNomSpeed)
  movel (pCtverec[3], tNastroj, mNomSpeed)
  movel (pCtverec[0], tNastroj, mNomSpeed)
  movel (pNajezdy, tNastroj, mNomSpeed)
  waitEndMove()
end
```

#program pro elipsu

```
begin
  movej (jBezpeci, tNastroj, mNomSpeed)
  movel (pNajezdy[1], tNastroj, mNomSpeed)
  movel (pOblouk[0], tNastroj, mNomSpeed)
  movec (pOblouk[1], pOblouk[2], tNastroj, mNomSpeed)
  movec (pOblouk[3], pOblouk[0], tNastroj, mNomSpeed)
  movel (pNajezdy[1], tNastroj, mNomSpeed)
  waitEndMove()
end
```

#program pro srdce

```
begin
  movej (jBezpeci, tNastroj, mNomSpeed)
  movel (pNajezdy[2], tNastroj, mNomSpeed)
  movel (pSrdce[0], tNastroj, mNomSpeed)
  movel (pSrdce[1], tNastroj, mNomSpeed)
  movec (pSrdce[2], pSrdce[3], tNastroj, mNomSpeed)
  movec (pSrdce[4], pSrdce[5], tNastroj, mNomSpeed)
  movel (pSrdce[0], tNastroj, mNomSpeed)
  movel (pNajezdy[2], tNastroj, mNomSpeed)
  waitEndMove()
end
```

Poté zavoláme v hlavním programu start() jeden podprogram po druhém:

```
begin
  call Ctverec()
  call Oblouk()
  call Srdce()
end
```

Nyní aplikaci vyzkoušejte v simulaci a po konzultaci s vyučujícím po přehození funkcí open() a close() ve všech podprogramech přehrajte přes Transfer Manager do ovladače robota.

11.2 Úloha 2 – Depaletizace

V této úloze se budeme zabývat jednoduchou depaletizací pomocí robota.

11.2.1 Zadání úlohy 2

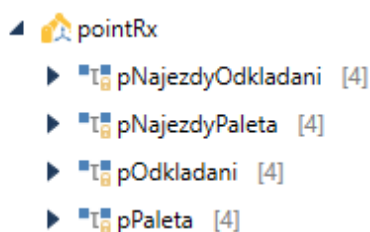
Vytvořte jednoduchý program pro depaletizaci pomocí robota. Pro kratší a přehlednější kód využijte funkce *For*. Body opět nastavte na reálném robotovi. Program prvně simulujte a po konzultaci s vyučujícím vyzkoušejte na robotovi. Na úlohu je možno využít obou druhů robotů v laboratoři.

11.2.2 Provedení úlohy 2

Založíme si novou Cell, kde vybereme robota, podle toho, na kterém pracovišti se nacházíme. Pro návod vybíráme robota RX2-60. Jediný rozdíl ve zpracování bude použití rozdílných typů bodů pro jednotlivé roboty. Tedy pointRX/jointRX pro robota RX2-60 a pointRS/jointRS pro robota RS2-60.

Poté si založíme novou aplikaci a buď vložíme gripper jako CAD soubor, nebo si uděláme náhradu gripperu v simulaci pomocí 3D objektu. (viz. 10.3.3)

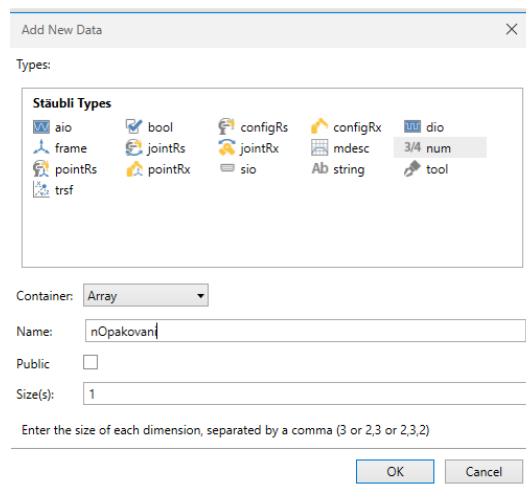
Jelikož máme založenou aplikaci i nastavený gripper můžeme pomocí ikony *Transfer Manager* nahrát aplikaci do ovladače. Ujistíme se, že robot je v manuálním režimu a poté ve FlexPendantu přidáme potřebné body a nastavíme jejich pozici. Bodů v našem případě potřebujeme 18. 16 bodů typu point, kde na uchopení součástek na paletě potřebujeme 4 a pro jejich nadejzdy také 4 (body nad uchopení, které nám zajistí, že při pohybu gripper nenarazí do součástky z boku). To stejné platí i pro odkládací plochu, kde robot bude odkládat součástky z palety. Zbývají jen 2 body typu joint, které budou sloužit pro nadejzd nad paletu a odkládací plochu. Pokud chceme můžeme nadefinovat ještě jeden bod jako bod bezpečí robota. Tyto body můžeme dát do jednoho pole pod název jTravel.



Obrázek 70 Potřebné body typu joint

Jakmile máme všechny body, co potřebujeme, Aplikaci na robotovi uložíme a pomocí ikony Transfer Manager jej přehrajeme zpět do počítače. Abychom viděli změny v softwaru musíme aplikaci znovu načíst pomocí pravého kliknutí na aplikaci a volby *Reload Application*.

Nyní již můžeme začít se psaním kódu. Pro funkci smyčky for musíme nadefinovat proměnnou typu num. To uděláme pravým kliknutím na aplikaci v kartě *Data* a zvolíme *Add New Data*. Dále zvolíme typ proměnné jako num a pojmenujeme ji jako nOpakovani.



Obrázek 71 Nová proměnná typu num

Kód pak bude vypadat následovně:

```
begin
  movej (jTravel[0], tGripper, mNomSpeed)
  open (tGripper)
  for nOpakovani=0 to 3 step 1
    movej (jTravel[0], tGripper, mNomSpeed)
    movel (pNajezdyPaleta[nOpakovani], tGripper, mNomSpeed)
    movel (pPaleta[nOpakovani], tGripper, mNomSpeed)
    waitEndMove ()
    close (tGripper)
    movel (pNajezdyPaleta[nOpakovani], tGripper, mNomSpeed)
    waitEndMove ()
    movej (jTravel[0], tGripper, mNomSpeed)
    movej (jTravel[1], tGripper, mNomSpeed)
    waitEndMove ()
    movel (pNajezdyOdkladani[nOpakovani], tGripper, mNomSpeed)
    movel (pOdkladani[nOpakovani], tGripper, mNomSpeed)
    waitEndMove ()
    open (tGripper)
    movel (pNajezdyOdkladani[nOpakovani], tGripper, mNomSpeed)
    waitEndMove ()
    movej (jTravel[1], tGripper, mNomSpeed)
  endFor
  movej (jTravel[0], tGripper, mNomSpeed)
end
```

Pro použití kódu na reálném robotovi musíme prohodit příkazy open() a close().

11.3 Úloha 3 – Stavění pyramidy

V této úloze budeme mít cílem poskládat pyramidu z předem dodaných válečků.

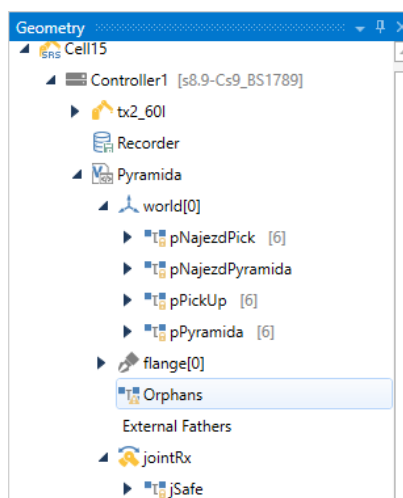
11.3.1 Zadání úlohy 3

Vytvořte jednoduchý program pro poskládání pyramidy z předem položených válečků. Body definujte na robotovi. Využijte smyčkových funkcí pro zjednodušení kódu. Program prvně simulujte a po konzultaci s vyučujícím vyzkoušejte na robotovi. Na úlohu se doporučuje použití robota RX2-60.

11.3.2 Provedení úlohy 3

Prvně si založíme novou Cell, pro kterou zvolíme robota RX2-60. Založíme novou aplikaci a pojmenujeme ji. Dále importujeme gripper jako CAD soubor, nebo si vytvoříme simulaci gripperu. Nástroj poté musíme nadefinovat jako Tool, aby správně pracoval (viz kapitola 10.3.3). Jakmile máme nastavený nástroj a založenou aplikaci opět nahrajeme aplikaci do kontroléru pomocí ikony *Transfer Manager*.

Na robotovi si opět nadefinujeme body, se kterými budeme pracovat. Potřebujeme 6 bodů typu point na vyzvednutí válečku a 6 na nadjezdy pro body. Poté potřebujeme 6 bodů typu point na poskládání pyramidy a jeden bod na nadjezd pro pyramidu. Dále potřebujeme jeden bod typu joint pro bezpečný výchozí bod robota.



Obrázek 72 Body pro úlohu
č.3

Po definování souřadnic bodů aplikaci uložíme a přeneseme zpět na počítač. Nyní můžeme začít psát kód.

```
begin
  movej (jSafe, tGripper, mNomSpeed)
  open (tGripper)
  for nCounter = 0 to 5 step 1
    move1 (pNajezdPick[nCounter], tGripper, mNomSpeed)
    move1 (pPickUp[nCounter], tGripper, mNomSpeed)
    waitEndMove()
    close (tGripper)
    move1 (pNajezdPick[nCounter], tGripper, mNomSpeed)
    move1 (pNajezdPyramida, tGripper, mNomSpeed)
    waitEndMove()
    move1 (pPyramida[nCounter], tGripper, mNomSpeed)
    waitEndMove()
    open (tGripper)
    move1 (pNajezdPyramida, tGripper, mNomSpeed)
    waitEndMove()
  endFor
end
```

11.4 Úloha 4 – Práce s GUI

V této úloze se budeme zabývat vlastním uživatelským rozhraním, kde budeme ovládat robota.

11.4.1 Zadání úlohy 4

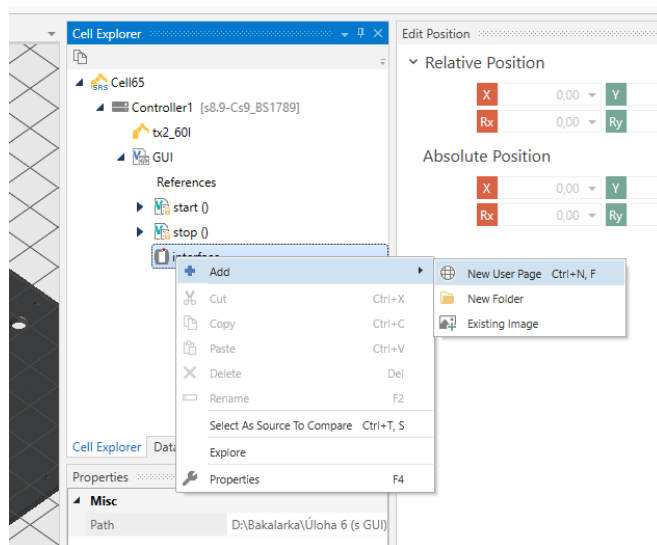
Vytvořte program s vlastním uživatelským rozhraním, kde před startem programu uživatel smí označit na jaké pozici palety se nacházejí válečky s dírou. Po startu programu, pro který by mělo být tlačítko na uživatelském rozhraní, robot rozdělí válečky s dírou a válečky bez díry. Body opět nastavte na reálném robotovi. Program prvně simulujte a po konzultaci s vyučujícím vyzkoušejte na robotovi. Na úlohu je možno využít obou druhů robotů v laboratoři.

11.4.2 Provedení úlohy 4

Jako první si založíme novou Cell a zvolíme robota podle pracoviště na kterém se nacházíme. V návodu budeme pracovat s robotem RX2-60. Založíme novou aplikaci a pojmenujeme ji. Vložíme nástroj jako CAD soubor, nebo si vytvoříme simulaci gripperu. Nástroj správně nastavíme a připojíme k robotovi (viz kapitola 10.3.3). Aplikaci si pomocí ikony Transfer Manager nahrajeme na robota, kde vytvoříme a nastavíme body. Pro aplikaci potřebujeme 24 bodů typu point a dva body typu joint. Čtyři body na vyzvednutí a body na nadejzdy. Čtyři body na položení válečků bez díry a body na nadejzdy. Čtyři body na položení válečků s dírou a body na nadejzdy. Body typu joint využíváme při pohybu mezi paletou a odkládacími plochami.

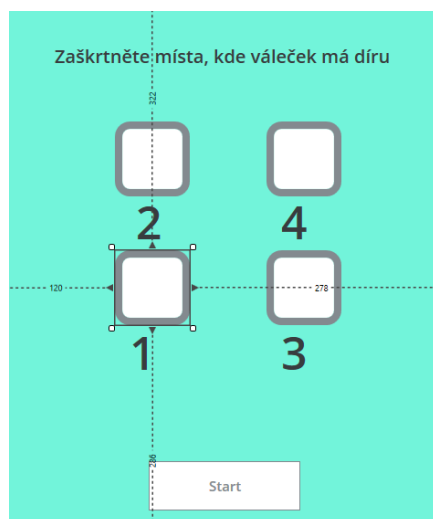
Po nastavení všech bodů aplikaci na FlexPendantu uložíme a pomocí *Transfer Manager* přeneseme novou aplikaci zpět na počítač.

Předtím než začneme psát kód založíme nové uživatelské rozhraní a to pravým kliknutím v kartě *Cell Explorer* na *Interface* kde zvolíme *New User Page*.



Obrázek 73 Založení nového GUI

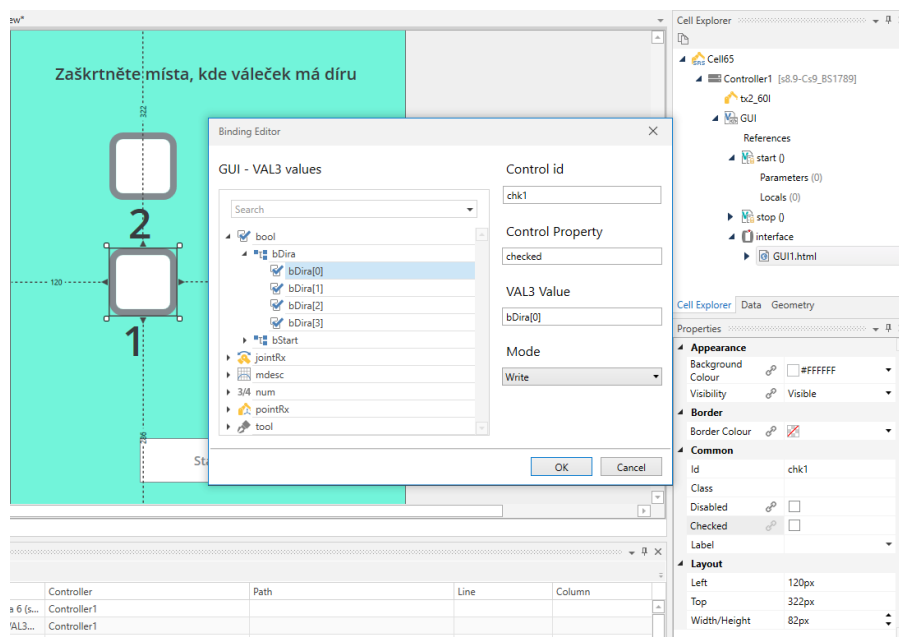
Po vybrání velikosti SP2 Portrait si na stránku vložíme všechny potřebné prvky pro správnou funkci programu. Pro program potřebujeme 4 zaškrťovací boxy (Checkbox), tlačítko na start (Button) a popisovací bloky (Label) na text. Dále budeme přidávat počítání válečků s dírou a bez ní a na to použijeme prvek Number.



Obrázek 74 Rozložení GUI

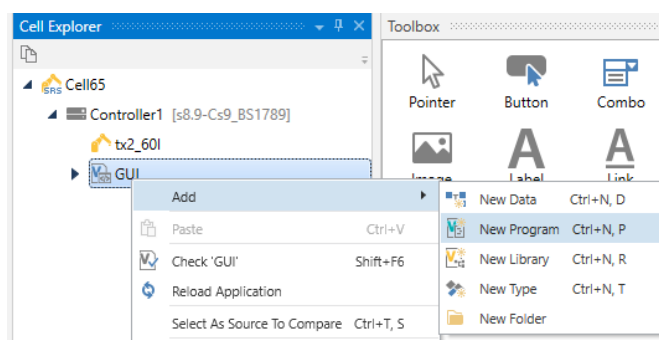
Pro propojení zaškrťovacího boxu potřebujeme založit proměnnou typu *bool*, kterou pojmenujeme *bDira* a pokud bude hodnota 1 budeme počítat s díravým válečkem.

Dále ve vlastnostech každého čtverce zvolíme ikonu řetězu u položky *Checked* a zde propojíme zaškrťovací box s proměnnými. Jako mód zvolíme *Write*. Dále si založíme proměnnou *bDisableCheck*, pomocí které vypneme možnost zaškrťvat box po startu programu. Tuto proměnnou propojíme s položkou *Disabled*.



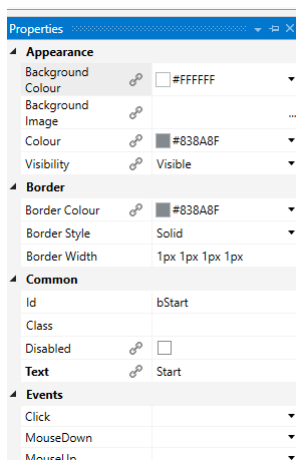
Obrázek 75 Nastavení zaškrťovacího boxu

Pro propojení tlačítka potřebujeme založit nový program, který provede kód když tlačítko stiskneme. Program pojmenujeme *clickStart*.



Obrázek 76 Vytvoření nového programu pro tlačítko

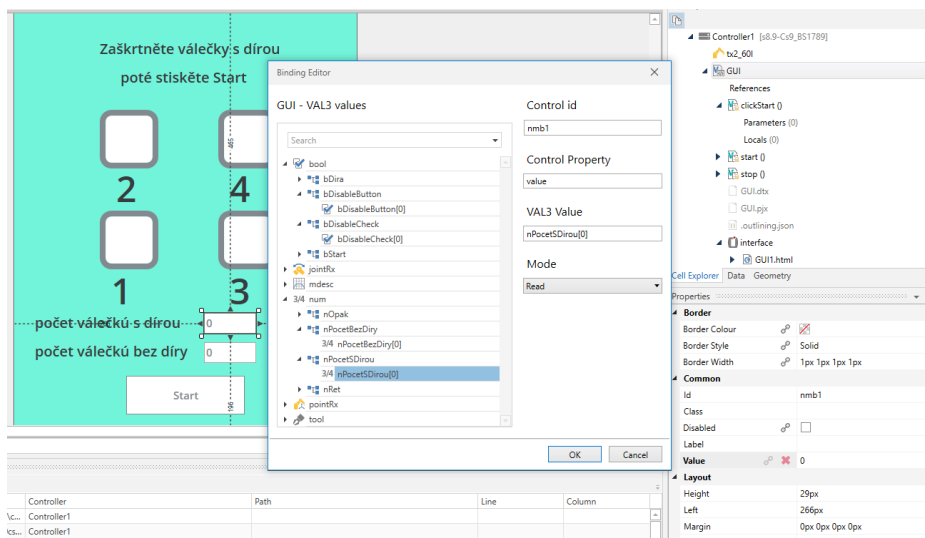
Pro propojení programu s tlačítkem ve vlastnostech tlačítka najdeme Složku *Events* a přidáme nový program do položky *Click*.



Obrázek 77 Nastavení tlačítka

Pro zobrazování počtu válečků s dírou a válečků bez díry musíme založit dvě nové proměnné typu *num*. Tyto proměnné pojmenujeme *nPocetSDirou* a *nPocetBezDiry*.

Tuto proměnnou propojíme s prvkem *Number* kliknutím na ikonu řetězu vedle položky *Value* ve vlastnostech prvku. Mód nastavíme na *Read*, aby uživatel neměl možnost čísla přepisovat.



Obrázek 78 Nastavení prvku Number

Jakmile máme všechny potřebné prvky propojené můžeme začít se psaním kódu. Jako první sepíšeme malý kód, který se spustí, když zmáčkne tlačítko. Tento kód spustí kód na robotovi a vypne možnosti zaškrtačacího boxu a stisk tlačítka. Pro start programu si založíme proměnnou typu *bool* a pojmenujeme ji *bStart*.

```
begin
  bStart = true
  bDisableCheck = true
  bDisableButton = true
end
```

Poté už jenom napíšeme hlavní program pro pohyb robota.

```
begin
  nPocetSDirou = 0
  nPocetBezDiry = 0
  bStart = false
  bDisableButton = false
  bDisableCheck = false
  movej(jSafe,tGripper,mNomSpeed)
  open(tGripper)
  userPage("GUI1")
  while !bStart
    wait(true)
  endwhile
  if bStart
    movej(jSafe[0],tGripper,mNomSpeed)
    open(tGripper)
    for nOpak = 0 to 3 step 1
      movel(pNajezdPick[nOpak],tGripper,mNomSpeed)
      movel(pPickup[nOpak],tGripper,mNomSpeed)
      waitEndMove()
      close(tGripper)
      movel(pNajezdPick[nOpak],tGripper,mNomSpeed)
      waitEndMove()
      movej(jSafe[1],tGripper,mNomSpeed)
      if !bDira[nOpak]
        movel(pNajezdBezDiry[nPocetBezDiry],tGripper,mNomSpeed)
        movel(pBezDiry[nPocetBezDiry],tGripper,mNomSpeed)
      end if
    end for
  end if
end
```

```
waitEndMove()
open(tGripper)
move1(pNajezdBezDiry[nPocetBezDiry],tGripper,mNomSpeed)
nPocetBezDiry = (nPocetBezDiry+1)
movej(jSafe[1],tGripper,mNomSpeed)
movej(jSafe[0],tGripper,mNomSpeed)
else
    move1(pNajezdDira[nPocetSDirou],tGripper,mNomSpeed)
    move1(pSdirou[nPocetSDirou],tGripper,mNomSpeed)
    waitEndMove()
    open(tGripper)
    move1(pNajezdDira[nPocetSDirou],tGripper,mNomSpeed)
    nPocetSDirou = nPocetSDirou+1
    movej(jSafe[1],tGripper,mNomSpeed)
    movej(jSafe[0],tGripper,mNomSpeed)
endIf
endFor
endIf
waitEndMove()
```

Nyní program otestujeme v simulaci a po přepsání všech instrukcí open() a close() program spustíme na reálném robotovi.

12 VYZKOUŠENÍ VYTVOŘENÝCH MANUÁLŮ

Všechny manuály vytvořené v rámci bakalářské práce byly poskytnuty testerům, aby manuály vyzkoušeli a zhodnotili jejich stručnost, množství informací a celkovou kvalitu. Při řešení úloh se většinou nenašly žádné problémy. Jediný problém byl u úlohy č.3 kde posazení bodu *pNajezdPyramida* bylo moc blízko bodů pyramidy a tím pádem při stavění došlo ke kolizi. Tato chyba se napravila buď posunutím bodu *pNajezdPyramida* nebo přidáním více bodů na nájezd. Testeři všechny úlohy zvládli a celkově se vyjádřili pozitivně.

ZÁVĚR

Cílem této bakalářské práce bylo vytvořit prakticky využitelné výukové materiály zaměřené na práci s robotickými rameny Stäubli a seznámit studenty s prostředím Stäubli Robotics Suite (SRS). V teoretické části byly popsány základní pojmy z oblasti robotiky, včetně historického vývoje, základních konstrukčních prvků robotů a přehledu typických robotických uspořádání a komponent. Samostatná pozornost byla věnována průmyslovým robotům Stäubli, zejména modelům TX2-60 a TS2-60, a jejich řídicí jednotce CS9.

V praktické části byly vytvořeny podrobné návody, které krok za krokem popisují, jak pracovat v prostředí SRS — od spuštění robotické linky, přes tvorbu nástrojů, objektů, bodů a vlastních uživatelských rozhraní, až po tvorbu simulačních programů. Dále byly navrženy čtyři výukové úlohy různé obtížnosti, jejichž prostřednictvím si studenti osvojí nejen základy ovládání a programování robotického ramene, ale i propojení simulace s reálným robotem.

Výsledné výukové materiály umožňují efektivní přechod mezi simulačním prostředím a reálnou aplikací, což přispívá k lepšímu pochopení fungování robotických systémů. Díky tomuto přístupu mají studenti možnost získat praktické zkušenosti, které jsou využitelné jak v akademickém prostředí, tak i v průmyslové praxi. Práce tak naplňuje svůj cíl a přináší přínos v oblasti technického vzdělávání v robotice.

SEZNAM POUŽITÉ LITERATURY

- [1] SICILIANO, Bruno, Lorenzo SCIAVICCO, Luigi VILLANI and Giuseppe ORIOLO, 2008. *Robotics* [online]. Dostupné z: doi:10.1007/978-1-84628-642-1
- [2] GASPARETTO, A. and L. SCALERA, 2018. From the Unimate to the Delta Robot: the early decades of industrial robotics. In: *History of mechanism and machine science/History of mechanism and machine sciense* [online]. p. 284–295. Dostupné z: doi:10.1007/978-3-030-03538-9_23
- [3] LI, Ming; MILOJEVIĆ, Andrija; HANDROOS, Heikki. Robotics in manufacturing—The past and the present. *Technical, Economic and Societal Effects of Manufacturing 4.0: Automation, Adaption and Manufacturing in Finland and Beyond*, 2020, 85-95.
- [4] STUDOCU, [b.r.]. Introduction to robotics - Bob Williams, williar4@ohio Mechanical Engineering, Ohio University EE/ME - Studocu. *Studocu* [online]. Dostupné z: <https://www.studocu.com/in/document/chandigarh-university/btech-cse/introduction-to-robotics/56433870>
- [5] VAN DAMME, Michael, [b.r.]. *Pneumatic artificial muscles (PAMs)* [online]. Dostupné z: http://mech.vub.ac.be/multibody/topics/pam_sub.htm
- [6] Jones, J.L., Seiger, B.A., & Flynn, A.M. (1998). *Mobile Robots: Inspiration to Implementation, Second Edition (2nd ed.)*. A K Peters/CRC Press. <https://doi.org/10.1201/9781439863985>
- [7] BABA, Abdullatif, 2023b. Flying robots for a smarter life. *SSRN Electronic Journal* [online]. 1 January 2023. DOI 10.2139/ssrn.4349634. Dostupné z: <https://doi.org/10.2139/ssrn.4349634>
- [8] CAIRNES, Mr Thomas J., Mr Christopher J. FORD, Dr Efi PSOMOPOULOU and Professor Nathan LEPORA, 2023. An overview of robotic grippers. *arXiv (Cornell University)* [online]. 1 January 2023. DOI 10.48550/arxiv.2304.14051. Dostupné z: <https://arxiv.org/abs/2304.14051>
- [9] Anon., [b.r.]. 6 axis industrial robots TX2-60 | Stäubli. *Staubli* [online]. Dostupné z: <https://www.staubli.com/global/en/robotics/products/industrial-robots/tx2-60.html#tabs-4005c1fe06-item-f29c000796-tab>

- [10] Anon., [b.r.]. CS9 robot controller. *Staubli* [online]. Dostupné z: <https://www.staubli.com/global/en/robotics/products/robot-controllers/cs9-robot-controller.html>
- [11] Anon., [b.r.]. TS2-60 SCARA industrial robots | Stäubli. *Staubli* [online]. Dostupné z: <https://www.staubli.com/global/en/robotics/products/industrial-robots/ts2-60.html#tabs-0dc10636b9-item-5f7e1cfeba-tab>
- [12] SKAŘUPA, Jiří, 2008. *Průmyslové roboty a manipulátory*.
- [13] BOTEK, Petr. *Tvorba výukových materiálů pro robotickou ruku Stäubli TX2-60*. 2022. Dostupné z: <http://hdl.handle.net/10563/50435>
- [14] STÄUBLI, International. *VAL 3 reference manual: Version 8*. D.280.935.04-H. Pfäffikon / Switzerland, 2022.
- [15] STÄUBLI, International. *CS9 Controller: Instruction manual*. D.280.940.04-K. Pfäffikon / Switzerland, 2023.
- [16] STÄUBLI, International. *Arm - TX2 series 60 family: Instruction manual*. D.280.942.04-G. Pfäffikon / Switzerland, 2022.
- [17] STÄUBLI, International. *Arm - TS series 40/60/80 family: Instruction manual*. C.285.036.04-E. Pfäffikon / Switzerland, 2020.
- [18] STÄUBLI, International. *STÄUBLI Robotics Suite 2022 (SRS): User's manual*. D.280.983.04-B. Pfäffikon / Switzerland, 2022.
- [19] Anon., [b.r.]. How we are. *Staubli* [online]. Dostupné z: <https://www.staubli.com/cz/en/about-us/how-we-are.html>

SEZNAM POŽITÝCH SYMBOLŮ A ZKRATEK

mm milimetr

SEZNAM OBRÁZKŮ

Obrázek 1 McKibben Muscle [5].....	17
Obrázek 2 Kartézský manipulátor [1]	18
Obrázek 3 Portálová konstrukce [1].....	18
Obrázek 4 Válcový manipulátor [1].....	19
Obrázek 5 Sfěrický manipulátor [1].....	20
Obrázek 6 SCARA manipulátor [1].....	21
Obrázek 7 Antropomorfní manipulátor [1]	22
Obrázek 8 Obě instalace robota TS2-60 [11].....	28
Obrázek 9 Robot TX2-60 [9]	29
Obrázek 10 Řídící jednotka CS9 [10]	30
Obrázek 11 Hlavní vypínač.....	33
Obrázek 12 Umístění vypínačů.....	33
Obrázek 13 Panel pro práci s kójí	34
Obrázek 14 Panel pro práci s robotem	34
Obrázek 15 Nový průvodce vytvoření kóje	35
Obrázek 16 Přejmenování kóje	36
Obrázek 17 Typ ovladače	36
Obrázek 18 Výběr robota.....	37
Obrázek 19 3D view	38
Obrázek 20 Karta Home	38
Obrázek 21 Karta VAL3	39
Obrázek 22 Karta Modelling.....	39
Obrázek 23 Karta Simulations	39
Obrázek 24 Boční panely	40
Obrázek 25 Geometry	40
Obrázek 26 Cell Explorer	40
Obrázek 27 Data.....	41
Obrázek 28 Vlastnosti.....	41
Obrázek 29 Založení nové aplikace	42
Obrázek 30 Pojmenování nové aplikace	42
Obrázek 31 Vytvoření válce	43
Obrázek 32 Selektce při RCM na objekt.....	43
Obrázek 33 New Tool.....	44
Obrázek 34 Centre on	45

Obrázek 35 Handle	45
Obrázek 36 Připojení nástroje na robota.....	46
Obrázek 37 Nový typ nástroje	46
Obrázek 38 Pojmenování nástroje	47
Obrázek 39 Device Configuration	47
Obrázek 40 Cesta Device Configuration	48
Obrázek 41 Zapnutí magnetického gripperu	48
Obrázek 42 Nastavení rozměrů Gripperu	49
Obrázek 43 Vložení CAD modelu.....	49
Obrázek 44 Nový bod	51
Obrázek 45 Pojmenování bodu.....	51
Obrázek 46 Move Here.....	52
Obrázek 47 Nastavení bodu.....	52
Obrázek 48 Transfer manager.....	53
Obrázek 49 Editování aplikace	53
Obrázek 50 Ikona pro pohyb	53
Obrázek 51 Nastavení bodu na robotovi.....	54
Obrázek 52 Transfer zpět.....	54
Obrázek 53 Uložení aplikace	54
Obrázek 54 Paleta	55
Obrázek 55 Přidání souřadného systému.....	55
Obrázek 56 Založení nové knihovny	56
Obrázek 57 Pojmenování nové knihovny	56
Obrázek 58 Využití proměnných z knihovny	57
Obrázek 59 Založení nového uživatelského rozhraní.....	57
Obrázek 60 Volba velikosti a jména rozhraní	58
Obrázek 61 Rozhraní a možnosti prvků	58
Obrázek 62 propojení prvků s proměnnými	59
Obrázek 63 Nový podprogram	61
Obrázek 64 Zavolání podprogramu	61
Obrázek 65 Zapnutí synchronizace.....	62
Obrázek 66 FlexPendant.....	63
Obrázek 67 Výběr robota.....	64
Obrázek 68 Soubor Gripperu.....	65
Obrázek 69 Simulace součástky	65

Obrázek 70 Potřebné body typu joint	68
Obrázek 71 Nová proměnná typu num	69
Obrázek 72 Body pro úlohu č.3	71
Obrázek 73 Založení nového GUI	74
Obrázek 74 Rozložení GUI.....	74
Obrázek 75 Nastavení zaškrtačacího boxu	75
Obrázek 76 Vytvoření nového programu pro tlačítko	75
Obrázek 77 Nastavení tlačítka	76
Obrázek 78 Nastavení prvku Number.....	76

SEZNAM PŘÍLOH