

Tvorba výukových materiálů pro robotickou ruku Stäubli TX2-60

Petr Botek

Bakalářská práce
2022



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

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

Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Petr Botek**
Osobní číslo: **A19196**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Inteligentní systémy s roboty**
Forma studia: **Prezenční**
Téma práce: **Tvorba výukových materiálů pro robotickou ruku Stäubli TX2-60**
Téma práce anglicky: **Creation of Teaching Materials for the Stäubli TX2-60 Robotic Arm**

Zásady pro vypracování

1. Seznamte se s konkrétním typem robota, vývojovým prostředím, jeho začleněním v robotické lince a souvisejícím technickým vybavením.
2. Navrhněte několik úloh s různou složitostí, využívajících vybavení a možnosti konkrétního robota/pracoviště.
3. Implementujte úlohy do vývojového prostředí robota, včetně 3D modelů rozšiřujících přípravků, v případě, že jste nějaké použili.
4. Provedte testy jak v 3D simulaci ve vývojovém prostředí, tak na reálném robotu.
5. Vytvořte ke každé úloze podrobný a ucelený manuál s názornými vizualizacemi.
6. Ověřte manuály v praxi a vyhodnoťte jejich použitelnost.

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

Seznam doporučené literatury:

1. SKAŘUPA, Jiří. Průmyslové roboty a manipulátory. Ostrava: Vysoká škola báňská – Technická univerzita, 2008. ISBN 978-80-248-1522-0.
2. JAZAR, Reza N. Theory of applied robotics: kinematics, dynamics, and control. 2nd ed. New York: Springer, ©2010. ISBN 1441917497.
3. CATSOULIS, John. Designing embedded hardware. 2nd ed. Sebastopol: O'Reilly, 2005. ISBN 9780596007553.
4. STÄUBLI. Arm – TX2 series 60 family Instruction manual: TX2-60 – Master. D.280.942.04-C. ©2018, 104 s.
5. STÄUBLI. CS9 Controller Instruction Manual: CS9 – Master. D.280.940.04-C. ©2018, 170 s.
6. PAGÁČ, Marek. Učebnice SolidWorks. V Brně: Vydavatelství Nová média, [2020]. ISBN 978-80-270-8730-3.

Vedoucí bakalářské práce: **Ing. Jiří Zátopek**
Ústav automatizace a řídicí techniky

Datum zadání bakalářské práce: **15. ledna 2022**
Termín odevzdání bakalářské práce: **20. května 2022**

doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan



prof. Ing. Vladimír Vašek, CSc. v.r.
ředitel ústavu

Ve Zlíně dne 15. ledna 2022

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské 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 právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že 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;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že 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 samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne

Petr Botek v.r.
.....
podpis studenta

ABSTRAKT

Účelem bakalářské práce je seznámit studenty s problematikou programování robotů takovým způsobem, aby byli schopni samostatně vytvořit funkční aplikaci jak simulačně, tak i na reálném robotu. K tomu budou sloužit 3 vytvořené úlohy s pečlivě sepsaným manuálem, který bude krok po kroku vysvětlovat práci s vývojovým prostředím *Stäubli Robotics Suite* a také na reálném robotu.

Klíčová slova: robot, aplikace, Stäubli, koncový efektor, kloub, úloha

ABSTRACT

The purpose of the bachelor thesis is to acquaint students with the problems of robot programming in such a way that they are able to independently create a functional application both simulation and even on a real robot. This will be done using 3 created tasks with a carefully written manual, which will explain step by step how to work with the *Stäubli Robotics Suite* development environment and also on a real robot.

Keywords: robot, application, Stäubli, end effector, joint, task

Poděkování

Chtěl bych velmi poděkovat vedoucímu bakalářské práce Ing. Jiřímu Zátopkovi za jeho trpělivost, skvělou komunikaci a ochotu poradit a pomoci. Taktéž bych chtěl poděkovat Petru Navrátilovi, který byl ochoten mi pomoci při řešení problému. Dále chci poděkovat firmě Navi, především Davidu Zatloukalovi, který mi pomáhal při návrhu přídavné paletky. Na závěr bych chtěl také poděkovat mým rodičům, kteří mi byli oporou po celou dobu mého studia.

OBSAH

| | |
|---|-----------|
| ÚVOD | 9 |
| I TEORETICKÁ ČÁST | 10 |
| 1 ROBOTIKA | 11 |
| 1.1 OBECNÁ ROBOTIKA | 11 |
| 1.2 PRŮMYSLOVÁ ROBOTIKA – HISTORICKÝ VÝVOJ | 11 |
| 2 ROBOT | 13 |
| 2.1 DEFINICE | 13 |
| 2.2 SLOŽENÍ A KLOUBY | 13 |
| 2.3 AKČNÍ ČLENY..... | 14 |
| 2.4 SENZORY | 14 |
| 2.5 CONTROLLER – ŘÍDÍCÍ JEDNOTKA..... | 15 |
| 3 MOBILNÍ ROBOTY | 16 |
| 3.1 DĚLENÍ MOBILNÍCH ROBOTŮ..... | 16 |
| 3.1.1 Kolové roboty..... | 16 |
| 3.1.2 Kráčeující roboty..... | 17 |
| 3.1.3 Létající roboty | 18 |
| 3.1.4 Polní roboty..... | 20 |
| 3.1.5 Servisní roboty | 20 |
| 4 MANIPULÁTORY | 22 |
| 4.1 SMYČKY MANIPULÁTORU | 23 |
| 4.2 POJEZDOVÉ ÚSTROJÍ | 24 |
| 4.3 POLOHOVACÍ ÚSTROJÍ..... | 24 |
| 4.4 ORIENTAČNÍ ÚSTROJÍ (ZÁPĚSTÍ) | 25 |
| 4.5 KONCOVÝ EFEKTOR..... | 26 |
| 4.5.1 Vakuová chapadla | 26 |
| 4.5.2 Pneumatické chapadla..... | 27 |
| 4.5.3 Hydraulická chapadla | 27 |
| 4.5.4 Elektrická chapadla..... | 27 |
| 4.6 PRACOVNÍ PROSTOR | 28 |
| 5 USPOŘÁDÁNÍ ROBOTŮ | 30 |
| 5.1 SCARA (SELECTIVE COMPLIANT ARM FOR ROBOTIC ASSEMBLY)..... | 30 |
| 5.2 ANTROPOMORFNÍ..... | 31 |
| 5.3 SFÉRICKÝ..... | 32 |
| 5.4 CYLINDRICKÝ | 32 |
| 5.5 KARTÉZSKÝ | 33 |
| 6 STÄUBLI TX2-60 | 36 |
| 7 CS9 CONTROLLER | 38 |
| 8 ÚLOHY NA PROCVIČENÍ | 40 |
| II PRAKTICKÁ ČÁST | 41 |
| 9 PRÁCE VE STÄUBLI ROBOTICS SUITE | 42 |

| | | |
|-----------|---|------------|
| 9.1 | ZALOŽENÍ NOVÉHO PROJEKTU..... | 42 |
| 9.2 | ZÁKLADNÍ OVLÁDACÍ PRVKY..... | 45 |
| 9.3 | ZALOŽENÍ NOVÉ APLIKACE A VYTVOŘENÍ OBJEKTŮ | 47 |
| 9.4 | VYTVOŘENÍ NÁSTROJE A SOUČÁSTEK..... | 53 |
| 9.5 | VLOŽENÍ OBJEKTU..... | 59 |
| 9.6 | VYTVOŘENÍ BODU A VLASTNÍHO SOUŘADNÉHO SYSTÉMU | 61 |
| 9.7 | VYTVOŘENÍ PROGRAMU A SPUŠTĚNÍ SIMULACE | 68 |
| 9.8 | POUŽITÍ PODPROGRAMŮ | 73 |
| 9.9 | POUŽITÍ KNIHOVEN..... | 74 |
| 9.10 | VYTVOŘENÍ UŽIVATELSKÉHO ROZHŘANÍ..... | 78 |
| 10 | ZPRACOVÁNÍ ÚLOH PRO STUDENTY | 85 |
| 10.1 | ÚLOHA 1..... | 85 |
| 10.1.1 | Zadání | 85 |
| 10.1.2 | Vložení a přemístění válečků | 85 |
| 10.1.3 | Převod válečků součásti | 87 |
| 10.1.4 | Vytvoření nástroje | 88 |
| 10.1.5 | Vytváření bodů..... | 91 |
| 10.1.6 | Vytvoření programu | 94 |
| 10.1.7 | Verze A | 94 |
| 10.1.8 | Verze B | 96 |
| 10.1.9 | Verze C | 97 |
| 10.1.10 | Program pro reálného robota..... | 97 |
| 10.1.11 | Připojení k reálnému robotu..... | 98 |
| 10.2 | ÚLOHA 2..... | 99 |
| 10.2.1 | Zadání | 99 |
| 10.2.2 | Vložení CAD objektu | 99 |
| 10.2.3 | Převod válečku na součástku | 100 |
| 10.2.4 | Vložení nástroje..... | 101 |
| 10.2.5 | Definování bodů..... | 103 |
| 10.2.6 | Programování pohybu..... | 105 |
| 10.2.7 | Převod na reálného robota..... | 108 |
| 10.2.8 | Připojení k reálnému robotu..... | 109 |
| 10.3 | ÚLOHA 3..... | 110 |
| 10.3.1 | Zadání | 110 |
| 10.3.2 | Vložení CAD objektu | 110 |
| 10.3.3 | Převod válečků na součástky | 111 |
| 10.3.4 | Vložení nástroje..... | 113 |
| 10.3.5 | Definice bodů | 116 |
| 10.3.6 | Vytvoření vlastního souřadného systému | 118 |
| 10.3.7 | Vytvoření programu | 122 |
| 10.3.8 | Přesunutí paletky | 124 |
| 10.3.9 | Převod na reálného robota | 125 |
| 10.3.10 | Připojení na reálného robota | 126 |
| 11 | OVĚŘENÍ MANUÁLŮ VYTVOŘENÝCH ÚLOH..... | 128 |
| | ZÁVĚR | 129 |
| | SEZNAM POUŽITÉ LITERATURY | 130 |

| | |
|--|------------|
| SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK..... | 133 |
| SEZNAM OBRÁZKŮ | 134 |
| SEZNAM PŘÍLOH | 142 |

ÚVOD

V dnešní době se stále více využívá robotů v širokém odvětví aplikací, kvůli výhodám, které jsou roboty schopné poskytnout. V teoretické části bakalářské práce bude přiblížený vývoj robotů z hlediska robustnosti, struktur a aplikací. Pod pojmem struktura si můžeme představit uspořádání kloubů robota, jaké akční členy obsahuje a zda je připojeno zápěstí nebo koncový efektor. Rozebráno bude i dělení robotů v ohledu na možnosti pohybu. Mluvit se bude také o typech koncových efektorů, pracovních prostorech jednotlivých uspořádání a také o aplikačních možnostech.

Primárním cílem praktické části bakalářské práce je navrhnout výukové úlohy pro studenty, kde budou srozuměni s problematikou programování robotů. Část praktické části je věnována obecnému manuálu, který slouží jako návod pro základní práci a fungování ve vývojovém prostředí *Stäubli Robotics Suite*. Výukové úlohy byly zpracovány formou manuálu, kde je krok za krokem vysvětlený postup. Každá úloha je zaměřená na jinou problematiku. V první úloze se studenti mají naučit práci s body, správné používání pohybových funkcí, aproximaci pohybu a práci s cykly. V druhé úloze je primárním úkolem se naučit pokročilé polohování koncového efektoru. Pro uskutečnění této úlohy byla navržena paleta, se kterou se pracuje ve 2 úlohách. Třetí úloha má znázornit výhodu využívání vlastních souřadných systému. Všechny úlohy budou zpracovány nejprve simulačně a následně převedeny na reálného robota. Studenti by po projití těmito úlohami měli být schopni samostatné práce jak ve vývojovém prostředí, tak i práce na reálném robotu Stäubli TX2-60L. Ověření, zda jsou manuály k jednotlivým úlohám pochopitelné a pečlivě zpracovány, bude prováděno na vybraných studentech, jejichž hodnocení bude bráno jako zpětná vazba, z které budou čerpány informace o věcech, které jsou potřeba doplnit do manuálů nebo potřeba upravit něco na reálném robotu či zkonstruované paletě.

I. TEORETICKÁ ČÁST

1 ROBOTIKA

1.1 Obecná robotika

V průběhu staletí se lidské bytosti neustále pokoušely hledat náhražky, které by byly schopny napodobit lidské chování v různých případech interakce s okolním životním prostředím. Několik motivací inspirovalo toto neustálé hledání odkazující na filozofické, ekonomické, sociální a vědecké principy. Slovo „Robot“ prvně zaznělo v dramatu R.U.R. (Rossumovi univerzální roboti) od Karla Čapka. Obraz robota jako mechanického artefaktu začíná ve 40. letech 20. století, kdy Rus Isaac Asimov pojal robota jako automat lidského vzhledu, ale prostý citů. Jeho chování bylo diktováno „pozitronickým“ mozkiem naprogramovaným lidskou bytostí tak, aby splňovala určitá pravidla etiky chování. Termín robotika pak Asimov zavedl jako vědu věnovanou studiu robotů, která byla založena na třech základních zákonech:

1. Robot nesmí zranit člověka nebo svou nečinností dovolit člověku přijít k újmě.
2. Robot musí uposlechnout příkazů, které mu dají lidské bytosti, kromě případů, kdy by byly příkazy v rozporu s prvním zákonem.
3. Robot musí chránit svou vlastní existenci, dokud taková ochrana není v rozporu s prvními dvěma zákony.

Tyto zákony stanovily pravidla chování, která je třeba považovat za specifikace pro design robota, který od té doby dosáhl transformace na průmyslový výrobek navržený inženýry nebo specializovanými techniky. Podle vědecké interpretace scénáře sci-fi je robot vnímán jako stroj, který je nezávisle na svém exteriéru schopen upravovat prostředí, ve kterém působí. Toho se dosáhne provedením jednání, která jsou podmíněna určitými pravidly chování stroje a také některými údaji, které robot získává o jeho stavu a o životním prostředí. Ve skutečnosti je robotika běžně definována jako věda, která studuje inteligentní spojení mezi vnímáním a jednáním. [2]

1.2 Průmyslová robotika – historický vývoj

Základní součástí robota je mechanický systém vybavený, obecně pohybovým aparátem (kola, pásy, mechanické nohy) a manipulačním aparátem (mechanická ramena, koncové efektory, umělé ruce). Realizace takového systému odkazuje na návrh kloubových mechanických systémů a výběru materiálu tak, aby byl schopen vykonat jak pohyb, tak manipulaci.

To je zajištěno ovládacím systémem, který oživuje mechanické součásti robota. Takový systém potřebuje řízení pohybu, zabývající se servopohony, pohony a převodovkami. Schopnost vnímání je svěřena systému, který může získat údaje o vnitřním stavu mechanického systému (snímače pohybu a polohy) a také o stavu prostředí (např. kamery). Dále odkazuje na souhrn vlastností materiálů, úprav a zpracování signálu, dat a vyhledávání informací. Schopnost propojit akci s vnímáním inteligentním způsobem poskytuje řídicí systém, který může nařídit provedení akce s ohledem na cíle stanovené technikou plánování úkolů, stejně jako omezení kladených robotem a prostředím. Realizace takového systému se řídí stejným principem zpětné vazby věnovaným řízení funkcí lidského těla, případně s využitím popisu komponentů robotického systému. Kontextem je kybernetika, zabývající se řízením a dohledem nad pohybem robota a umělou inteligencí. Závisí ovšem na výpočetní architektuře a programovacím prostředí. Proto lze uznat, že robotika je interdisciplinární předmět týkající se kulturních oblastí mechaniky, řízení, výpočetní techniky a elektroniky.[2] První zařízení s řízením polohy vzniklo roku 1938 pro stříkání barev. V oblasti výzkumu byl pravděpodobně prvním automatickým strojem stroj Grege Waltera, který se jmenoval "Mechanická želva". Prvního programovatelného robota navrhl George Devol roku 1954. UNIMATE byl první průmyslový robot umístěný na trh. Roku 1962 byla založena první robotická firma Unimation. V roce 1955 byla vytvořena Denavit a Hartenbergova teorie, která zkonsolidovala popis rozvinuté kinematiky robotických manipulátorů. První robot s elektrickými pohony ASEA IRB-6 (1973). První robot řízený mikro počítačem Cincinnati Milacron T3(1974). Hirata AR-300 (1978) byl první SCARA robot. Unimation PUMA 560 (1978) 6R byl první robot, který měl uspořádání podobné lidské paži. Je možné říct, že průmyslové roboty byly použity po roce 1960 a jejich raketový rozvoj nastal v 80-tých letech. Důsledkem tohoto byly velké investice do automobilového průmyslu. Robot je spojením 2 technologií: teleoperátory, které se používaly pro manipulaci s radioaktivními materiály a CNC (computer numeric control) obráběcích strojů, které zvyšují požadovanou přesnost. V dnešní době se klade čím dál tím větší důraz na širší schopnost pohybu a na pokročilé senzorické vnímání. [3][4][14]

2 ROBOT

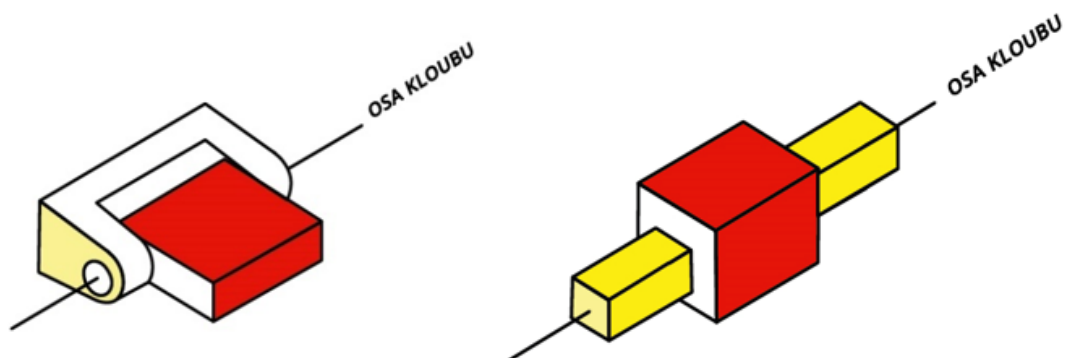
2.1 Definice

Robot je soustava ideálně tuhých (i pružných) mechanických těles vázanými mezi sebou navzájem kinematickými dvojicemi umožňující jejich vzájemný přímo řízený nebo vazbami vyvolaný cílevědomý pohyb za účelem přemístění celého robota nebo jeho částí v prostoru do definovaného místa případně definovanou rychlostí nebo s definovaným zrychlením. Na základě definice Robotického institutu Ameriky (RIA): „Robot je přeprogramovatelný multifunkční manipulátor určený k pohybu materiálu, dílů, nástrojů nebo specializovaných zařízení prostřednictvím proměnných naprogramovaných pohybu pro plnění různých úkolů.“ [5]

2.2 Složení a klouby

Robot se jako systém skládá z: Pojezdového ústrojí, manipulátoru, zápěstí, koncového efektoru, akčních členů, snímačů, řídicího systému a softwaru. Mechanická část robota je vytvořena jako kinematické spojení článků spojených v kloubech a tak vytvoří kinematický řetězec. Samostatná tuhá tělesa robota tvoří tzv. článek. Tento článek se může pohybovat vzhledem k ostatním článkům. Dva články jsou spojeny v kloubu, který umožňuje jejich vzájemný pohyb vyjádřitelný jednou souřadnicí. Existuje celkem 6 různých kloubů – rotační, translační, cylindrický, šroubový, sférický a planární. Rotační a translační jsou nejběžněji používané klouby u sériových průmyslových manipulátorů

- Translační kloub umožňuje vzájemný pohyb dvou článků po stejné ose.



Obrázek 1 Nejběžněji používané klouby [4]

Tyto osy necht' jsou nazvány osou kloubu. Vzájemná poloha dvou spojených článků je popsána kloubovými souřadnicemi. U rotačního kloubu je to úhel natočení a u translačního

kloubu se jedná o vzdálenost. Rotační kloub umožňuje relativní rotaci dvou spojených článku okolo vzájemné osy. Translační kloub zase umožňuje vzájemný pohyb po společné ose. Souřadnice aktivního kloubu je řízena akčním členem. Každý kloub přidává stupeň volnosti (DOF). Typicky 3 polohovací DOF a 3 pro orientaci. Manipulátor s více jak 6 DOF je kinematicky redundantní. Poloha volného tělesa v rovině je určena třemi na sobě nezávislými souřadnicemi – 2 souřadnice bodu a úhel svírající s osou x souřadnicové soustavy. Tyto 3 souřadnice nejsou vázány k základnímu rámu. Rámem se označuje základní nepohyblivé těleso. Roboty můžeme primárně rozdělit na **Mobilní roboty** a **Manipulátory**. [2][4][13]

2.3 Akční členy

Jsou to řídicí prvky, které působí jako sval robota. Jejich primární úkol je změna konfigurace. Poskytují energii pro působení na mechanickou konstrukci proti gravitaci, setrvačnosti a dalším vnějším silám, které by způsobily změnu geometrického tvaru umístění ruky robota. Akční členy jsou elektrické, hydraulické nebo pneumatické, ale je požadováno jejich ovládní. Elektrické akční členy pohání robota střídavými nebo stejnosměrnými motory a jsou považovány za nejpříjemnější volbu na pohon robota. Jsou čistší, tišší a přesnější vůči hydraulickým a pneumatickým. Elektromotory jsou účinné při vysoké rychlosti, tudíž je zapotřebí převodovka s vysokým převodovým poměrem. Vysoký převodový poměr zajistí samobrzdní. Pokud je vyžadována vysoká rychlost a potřeba vysoké nosnosti, výhodnější jsou hydraulické motory, které vyhovují kvůli poměru točivý moment/hmotnost nebo výkon/hmotnost. Hydraulické motory mají sklony k netěsnostem, hlučnosti a je zapotřebí čerpadla a jiného hardwaru. Pneumatické motory jsou levné a jednoduché, ale chybí možnost přesného řízení.[4]

2.4 Senzory

Díky sensorům jsme schopni zjistit a shromáždit informace o vnitřních stavech robota, ale také i o prostředí. Tomu se rozumí např. poloha kloubu, rychlost, zrychlení a síla. Zmíněné veličiny jsou nejdůležitější informace, které je potřeba vysnímat. Senzory integrované v robota odesílají informace do řídicí jednotky, která určí konfiguraci robota. [4]

- **Vidění** – počítačem řízená kamera poskytuje robotu možnost vidět jeho prostředí. Díky tomu může adekvátně upravovat svůj pohyb.

- **Síla/tlak** – umožňuje zjištění nesouososti, respektive nepravidelné zatížení na geometrii. Může také měřit kroutící momenty nebo momenty, které jsou síly působící na dálku.
- **Blížkost** -umožňují detekci objektů, které jsou velmi blízko, než se jich paže dotkne. Jsou použity pro možnost předcházení kolizím.
- **Koncové spínače** – mohou být instalovány na konci pohybu v pracovním prostoru. Mohou robota zastavit nebo obrátit jeho směr. Také slouží pro předcházení kolizí

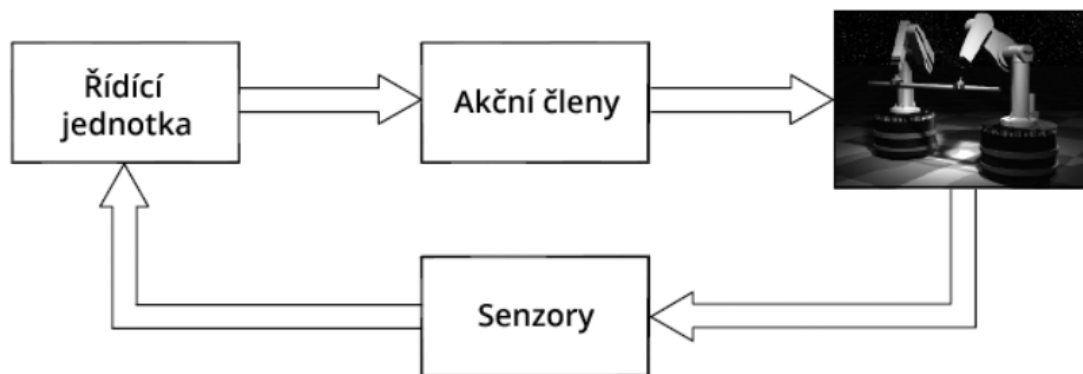
[1]

2.5 Controller – řídicí jednotka

Poskytuje nezbytnou inteligenci k ovládání robota. Řídicí jednotka má 3 základní funkce:

- **Informační role** – sbírá a zpracovává informace ze senzorů robota.
- **Rozhodovací role** – plánuje geometrický pohyb struktury robota.
- **Komunikační role** – organizuje informace mezi prostředím a robotem.

Součástí řídicí jednotky jsou procesor a software. [4][5]



Obrázek 2 Komponenty robotického systému [2]

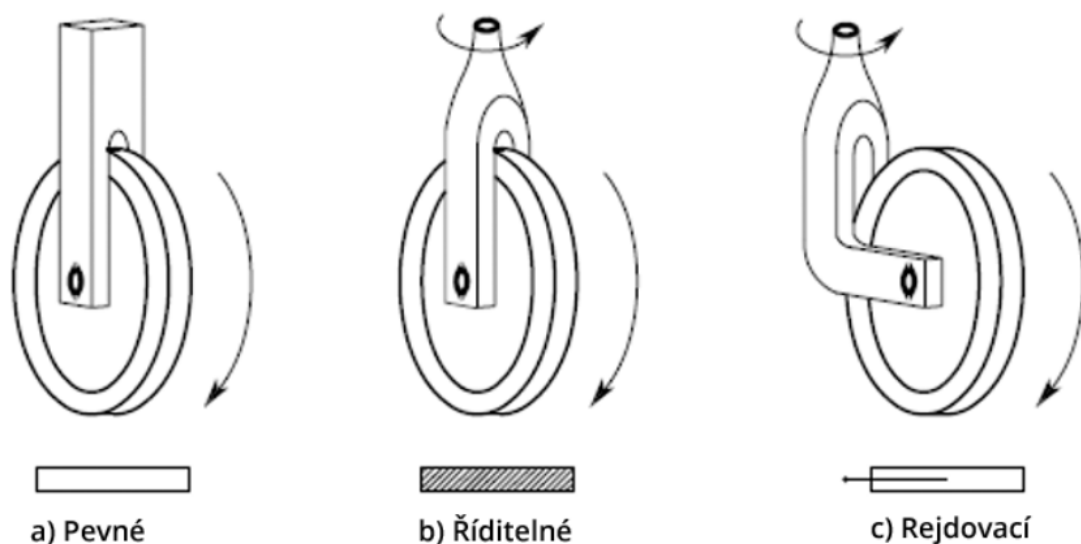
3 MOBILNÍ ROBOTY

Hlavní vlastností je přítomnost pohyblivé základny, která umožňuje robotu se volně pohybovat v prostředí. Tyto roboty jsou nejčastěji používány v servisních aplikacích, kde je potřeba rozsáhlých a autonomních pohybů. [2]

3.1 Dělení mobilních robotů

3.1.1 Kolové roboty

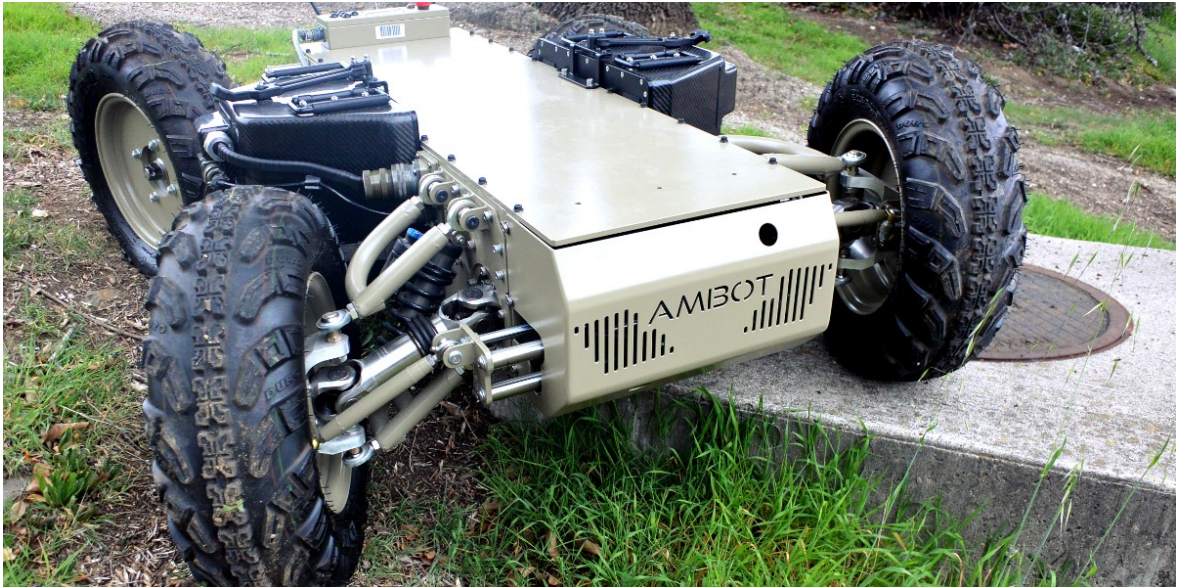
Základním mechanickým elementem robota je kolo. Existují 3 typy tradičních kol: Pevné, Říditelné, Rejdovací kolečko.



Obrázek 3 Typy kol kolových robotů [2]

- **Pevné (a)** - může rotovat kolo své osy procházející středem, která je kolmá na rovinu kola. Je pevně spojeno s podvozkem, jehož orientace vzhledem na kolo je konstantní.
- **Říditelné (b)** - má 2 osy rotace. Jedna prochází opět středem kola a druhá je vertikální procházející se středem kola, což umožňuje změnu orientace vzhledem k podvozku.
- **Rejdovací (c)** - má 2 osy rotace, ale vertikální osa nevede přes střed kola, ale je posunutá. To způsobí automatické otočení kola do směru pohybu podvozku. Použití pro poskytnutí statické rovnováhy.

Dělení kolových robotů závisí na počtu a uspořádání kol. Varianty kinematických struktur jsou široké a jsou dány kombinacemi těchto 3 kol. [2]



Obrázek 4 Kolový robot – AMBOT GRP 4400 [16]

3.1.2 Kráčejíci roboty

Dělení podle počtu nohou. Také se rozlišuje způsob chůze. [15]

- **Výhody:** překonávání vysokých překážek, pohyb po schodech, pohyb po extrémně členitém povrchu, hladký pohyb po nerovném povrchu díky nastavitelné výšce těla.
- **Nevýhody:** vyšší počet nezávisle řízených stupňů volnosti, vyšší počet akčních členů, složitější řízení z hlediska HW i SW, malá energetická účinnost, konstrukční složitost, výrobní náročnost, složitost ožívování, nejistota plnění funkce.



Obrázek 5 Kráčejíci robot – Laikago [17]

3.1.3 Létající roboty

Neobydlené letadlo je definováno jako UAV (uninhabited aerial vehicle), ROA (remotely operated aircraft) nebo RPV (remotely piloted vehicle). Řízení, jako je dynamický vztlak a tah na základech aerodynamiky, je prováděno autonomní navigací nebo pomocí dálkového ovládání. Do kategorie létajících robotů nepatří rakety, balistické střely a ani vzducholodě napuštěné plynem. Hlavními aplikacemi jsou: inspekce terénu, potrubí, budov, dále se používá při katastrofách pro pátrání a záchranu, pro letecké mapování a meteorologii. Nejpočetnější nasazení mají ovšem pro sledování a vojenskou obranu. [5] Dělí se na:

- **UAV s pevnými křídly** - potřebují dráhu ke vzletu nebo přistání. Mají vysokou vytrvalost a vysokou cestovní rychlost. [5]



Obrázek 6 UAV s pevnými křídly - MQ-9 Reaper [18]

- **UAV s rotačním křídlem** - nebo také **rotorová letadla**, používají vertikální vzlet a přistání. Mají vysoké manévrovací schopnosti, které jsou užitečné zejména v civilních aplikacích. [5]



Obrázek 7 UAV s rotačním křídlem – DJI Mavic 2 Pro

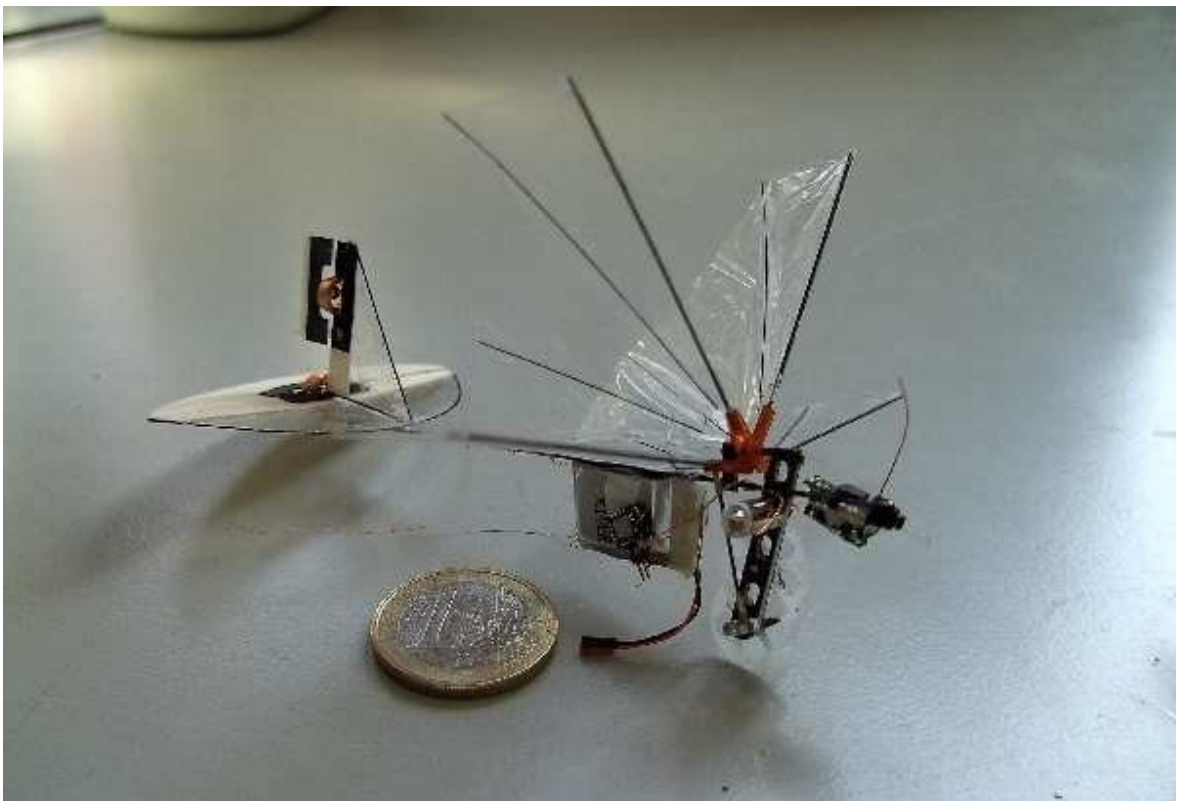
[19]

- **Vzducholoďe a balóny** - létají nízkou rychlostí a objemově jsou velké. [5]



Obrázek 8 Robotická vzducholoď [20]

- **UAV s mávajícím křídly** se inspirovaly flexibilními a tvarovatelnými křídly ptáků a létajícího hmyzu. [5]



Obrázek 9 UAV s mávajícím křídly – DelFly MAV [21]

3.1.4 Polní roboty

Tyto roboty se nasazují v oblastech, kde by člověk nemusel přežít nebo byl vystaven nepříznivým podmínkám. Primárním úkolem je provedení průzkumu a následné hlášení užitečných dat o prostředí. K této činnosti používají vhodné palubní senzory. Typické nasazení je v prostředí, kde se vyskytují vysoké teploty, toxický plyn nebo radiace. Dále se používají pro hloubkový. Nejnázornější ukázkou polního robota je Mars rover společnosti NASA, který na Marsu, který je částečně autonomní a částečně řízen ze Země. Nasazení je převážně pro záchranu lidských životů při katastrofách, kdy je nebezpečí dalšího požáru, exploze nebo nebezpečného plynu.[2]



Obrázek 10 Polní robot - Mars Rover [22]

3.1.5 Servisní roboty

Autonomní vozidla jsou také užita pro civilní aplikace jako součást inteligentních dopravních systémů věnované řízení dopravy v městských oblastech. Technologie se používá pro pomoc starším a postiženým lidem v každodenním životě pomocí autonomních invalidních vozíků, zvedače pomůcek, krmení a rehabilitace. Jedná se i o nosiče a zdravotní systémy. V lékařství se používají tam, kde je vyžadována vysoká přesnost polohování nástroje. Tento robot bývá ovládán pomocí 3D obrazu a dálkového zařízení, který zaznamenává hmatový

kontakt. V lékařství se také používají pro diagnostiku dutin lidského těla. V rehabilitačních systémech jsou servisní roboty reprezentovány pomocí Exoskeletu, který aktivně udržuje a koriguje pohyby podle naplánovaného programu fyzioterapeuta.[2]



Obrázek 11 Servisní robot – Stanley [23]

4 MANIPULÁTORY

Robot je automaticky řízený, přeprogramovatelný víceúčelový manipulátor navržený pro pohyb materiálů, dílů, nástrojů nebo speciálních zařízení pomocí proměnného programovaného pohybu, schopný vykonávat rozličné pohyby, přičemž přijímá informace ze svého okolí a na základě toho se pohybuje. Může být buď fixovaný na jednom místě nebo mobilní, použitelný v průmyslových automatizačních aplikacích. Manipulátor se sestává z článků, kloubů a dalších strukturálních prvků a stává se robotem, je-li připojeno zápěstí a koncový efektor a je-li zaveden řídicí systém. Efektor může být například úchopná hlavice. Design a řízení robota záleží na aplikaci v průmyslu. Robot pro průmyslové aplikace pracuje ve strukturovaném prostředí, kde známe geometrické a fyzické vlastnosti. Pohyb článků zajišťuje umělý sval, který představuje nový druh pohonu, založený na kombinaci pružného poddajného materiálu a elektronických prvků, který se podobá lidskému svalu. Hlavní faktor zavádění robotů v průmyslu je zvyšující se rozsah aplikací, snížení výrobních nákladů, zvýšení produktivity, zvýšení a stabilizace kvality výrobků a v první řadě eliminace škodlivých nebo nebezpečných úloh pro člověka. Termín automatizace se používá pro proces, kdy se nahrazuje člověk za robota, jde-li o fyzickou práci a inteligentní zpracování informace o stavu procesu. Automatizace se dá popsat 3 stupni.

- Pevná automatizace se zabývá hromadnou výrobou produktů stejného typu.
- Programovatelná automatizace se zabývá výrobou středně velkých šarží výrobků stejného typu. Tento stupeň umožňuje snadnou změnu operace za účelem změny výrobního produktu.
- Flexibilní automatizace se snaží umožnit variaci produktů při co nejmenší časové ztrátě pro přeprogramování operací. Vyžaduje silnou integraci počítače a průmyslovou technologii.

Průmyslový robot je tedy stroj s flexibilitou a všestranností. Průmyslový robot je typickou součástí programovatelných automatických systémů a představuje 3 základní funkce, které jsou užitečné pro výrobu: manipulace s materiálem, obecná manipulace a měření. Typické aplikace zahrnují: paletizaci, práce ve skladišti, třídění součástek, balení, obsluha fréz a obráběcích strojů. Schopnost robota pracovat se surovým materiálem nebo montáž více částí do jednoho celku je další důležitou vlastností, díky kterým se průmyslové roboty nasazují do výrobních procesů. [2][3]

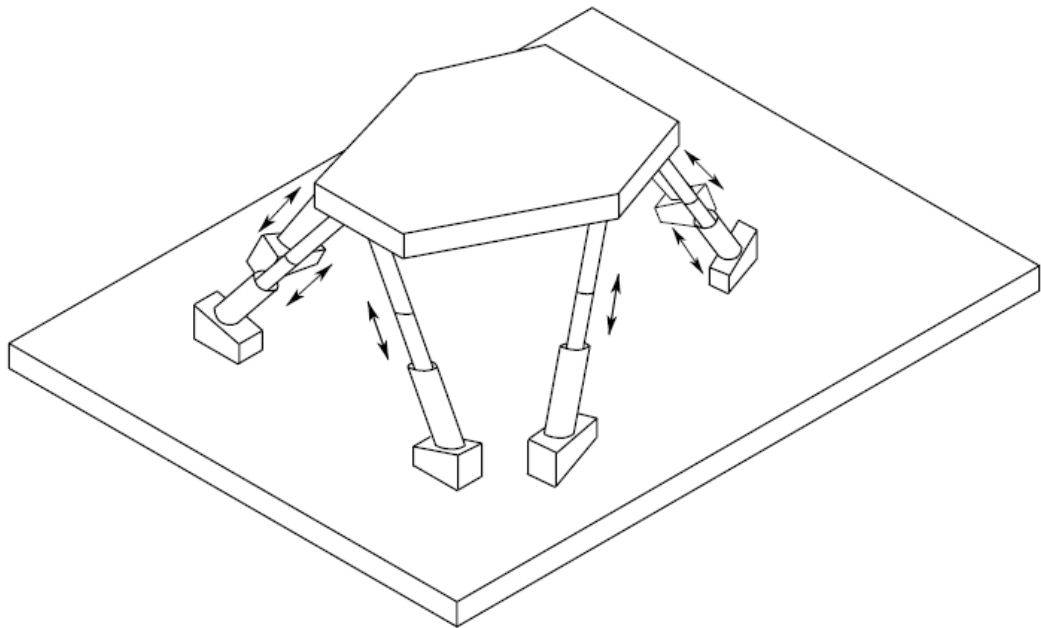
Typickými aplikacemi je:

- Sváření
- Lakování
- Řezání laserem nebo vodním paprskem
- Odjehlování a broušení
- Montáž mechanických i elektrických prvků
- Manipulace s materiálem (paletizace, balení, práce ve skladišti)

4.1 Smyčky manipulátoru

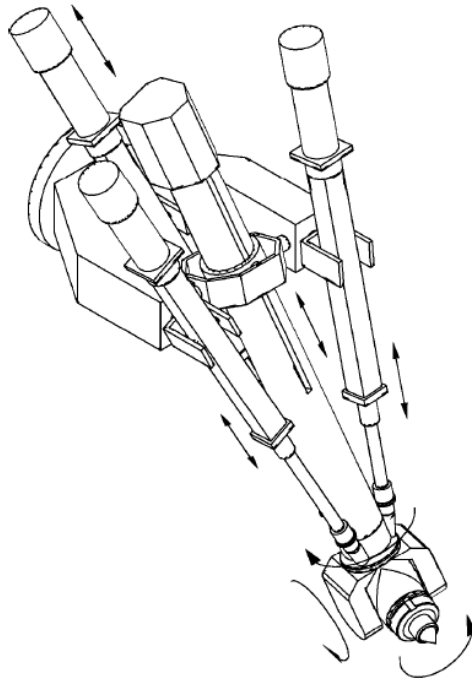
Sériový manipulátor – v otevřené smyčce – nevytváří se v řetězci smyčka

Paralelní manipulátor – v uzavřené smyčce – kinematická struktura vytváří v řetězci smyčku



Obrázek 12 Paralelní manipulátor [2]

Hybridní manipulátor – struktura obojího typu



Obrázek 13 Hybridní manipulátor [2]

4.2 Pojezdové ústrojí

Slouží k přesouvání manipulátorů a průmyslových robotů mezi pracovišti, nachází se buď na podlaze, na stropu nebo v úrovni manipulovaných pracovišť, roboty a manipulátory jsou na pojezdové ústrojí pouze připojeny, servisní roboty mají vlastní pojezdové ústrojí. [10]

4.3 Polohovací ústrojí

Slouží k přemísťování objektů, využití rotačních a translačních kloubů. U polohovacího ústrojí jsou možnosti spojení translačních a rotačních článků, jejich možnosti kombinací závisí na stupních volnosti, tzn. na jejich struktuře.[10]

1° DOF – 1 posuvná dvojice P nebo 1 posuvná dvojice R

2° DOF – PP, RP, PR, RR

3° DOF – PPP, RPP, PRP, PPR, RRP, PRR, RPR, RRR

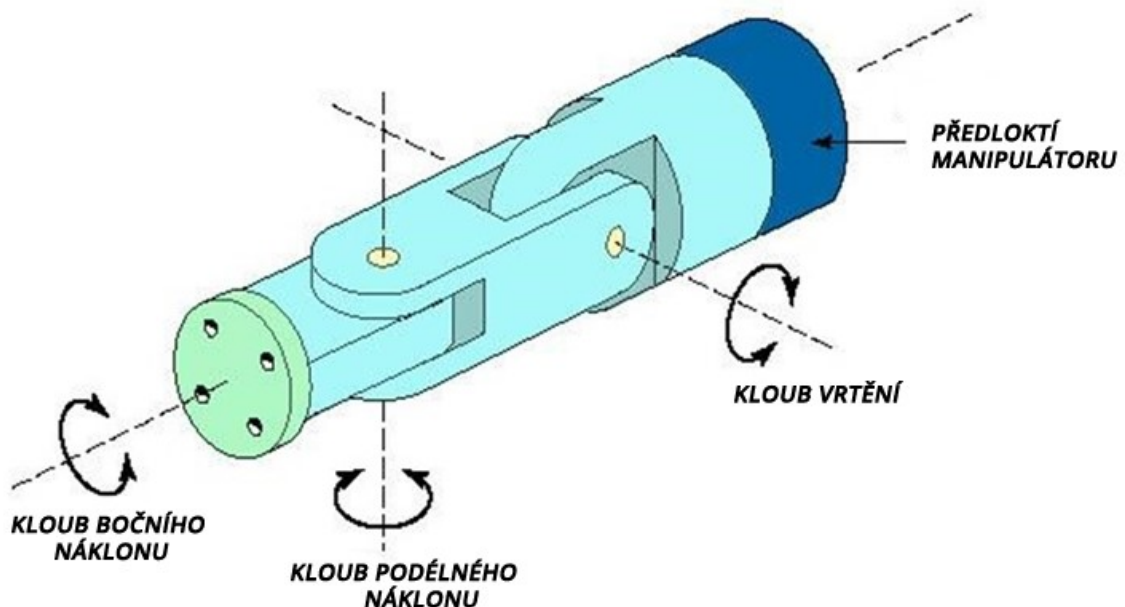
4.4 Orientační ústrojí (zápěstí)

Zachová nebo mění orientaci manipulovaného objektu pomocí koncového efektoru vůči pracovnímu prostředí. Nachází se mezi předloktím a koncovým efektozem robota, je zvykem navrhovat sférická zápěstí, tj. 3 rotační klouby s osami protínajícími se v 1 bodě. Klouby umožňují podélný náklon pro pohyb nahoru/dolů, boční náklon pro pohyb po směru nebo protisměru hodinových ručiček a vrtění doprava nebo doleva. Boční náklon zajišťuje rotaci okolo osy x. Rotaci kolem osy y zajišťuje podélný náklon. Pohyb okolo osy z nazýváme vrtění. Zápěstí vytvořené ze třech rotačních kloubů tvořících rotaci, podélný a boční náklon. Většina zápěstí má rotační klouby oddělené. Počet DOF má vliv na činnost zápěstí, měl by být vybírán opatrně obzvláště u 1DOF a 2DOF. Konstrukční nedostatek je, že se neprotínají osy kloubů v jednom bodě. [11]

Zápěstí s 1DOF – vrtění

Zápěstí s 2DOF – vrtění a boční náklon

Zápěstí s 3DOF – vrtění, podélný a boční náklon



Obrázek 14 Schématická ukázka 3DOF zápěstí [24]

Kinematické řetězce – rameno polohovacího mechanismu resp. referenční bod vykonává pohyb po přímce nebo křivce. Průmyslový robot se tedy skládá ze zařízení pro polohování a pro orientaci. [10][11]

4.5 Koncový efektor

Koncovým efektozem zasahuje robot do svého okolí. Koncovým efektozem se nazývá každé zařízení připojené na konec ramene robota. V případě manipulace s předměty se bavíme o úchopných hlavicích (gripper). V jistém smyslu je užitečné je chápat jako podobu lidské ruky. Možnosti použití chapadel závisí na konstrukčních prvcích jak manipulovaného předmětu, tak i z konstrukčního hlediska úchopné hlavice. Chapadla mohou mít 2,3 nebo také i 5 prstů. Chapadla mohou být i ve tvaru drápů nebo velké přísavky. Koncový efektor není pouze chapadlo. Jako koncový efektor můžeme brát chytrou kameru, svařovací nástroj a spousta dalších zařízení. [9,12]

4.5.1 Vakuová chapadla

Pracují na principu rozdílu atmosférického tlaku a vakuem ke zvedání a přemístění předmětů. Vakuum je většinou vytvořeno elektromechanickým čerpadlem nebo čerpadlem poháněným stlačeným vzduchem. Proud vakua musí být nepřerušovaný. Jsou 4 - 10 krát výkonnější, než elektromechanická chapadla. Z tohoto hlediska se hodí pro zvedání těžkých břemen. Nejpoužívanější aplikací těchto chapadel je balení a paletizace. Jsou schopni manipulovat se širokou škálou objektů. Nevýhodou je cena nákladů pro provoz a citlivost na prašné prostředí. [9]



Obrázek 15 Vakuové chapadlo od firmy
ROBOTIQ [25]

4.5.2 Pneumatické chapadla

Využívá stlačeného vzduchu a pístů k ovládnání čelistí. Nejčastější konfigurace je se 2 až 3 prsty. Jsou univerzální a mají široké pole využití. Výhodou je nízká cena, velký rozsah síly uchopení, práce ve stísněných prostorách a rychlá odezva. Nevýhodou je, že se více hodí na aplikace, kde se manipuluje stále se stejnými produkty z objemového a tvarového hlediska. [9]



Obrázek 16 Pneumatická chapadla [26]

4.5.3 Hydraulická chapadla

Jsou poháněna hydraulickými kapalinami, které poskytují větší uchopovací sílu než pneumatická chapadla. Jejich výhodou je vysoká uchopovací síla, ale s tím jsou i nevýhody. Například složitost spojená s manipulací oleje, potřebou nádrže a čerpadla. S tím jsou spojené velké náklady na údržbu. [9]

4.5.4 Elektrická chapadla

Jsou velmi oblíbenou volbou pro mnoho aplikací s kolaborativními roboty, ale i v aplikacích s obsluhou strojů a PICK and PLACE. Nenabízejí stejnou sílu jako pneumatické a hydraulické chapadla. Elektrická chapadla poskytují vysokou rychlost a lehkou/střední uchopovací sílu. Většinou v konfiguraci se 2 – 3 prsty pro manipulaci s kulatými/válcovými předměty. Charakteristickým rysem je ovládnání. Elektrická chapadla jsou dodávána s mikroprocesory,

které umožňují změnu uchopovací síly a rychlosti. Přidání snímače síly přidává možnost snadné manipulace s různými typy předmětů. Hlavní nevýhodou je nižší uchopovací síla a vysoká cena. [9]



Obrázek 17 Elektrický gripper
Zimmer Group GEP2000 [27]

4.6 Pracovní prostor

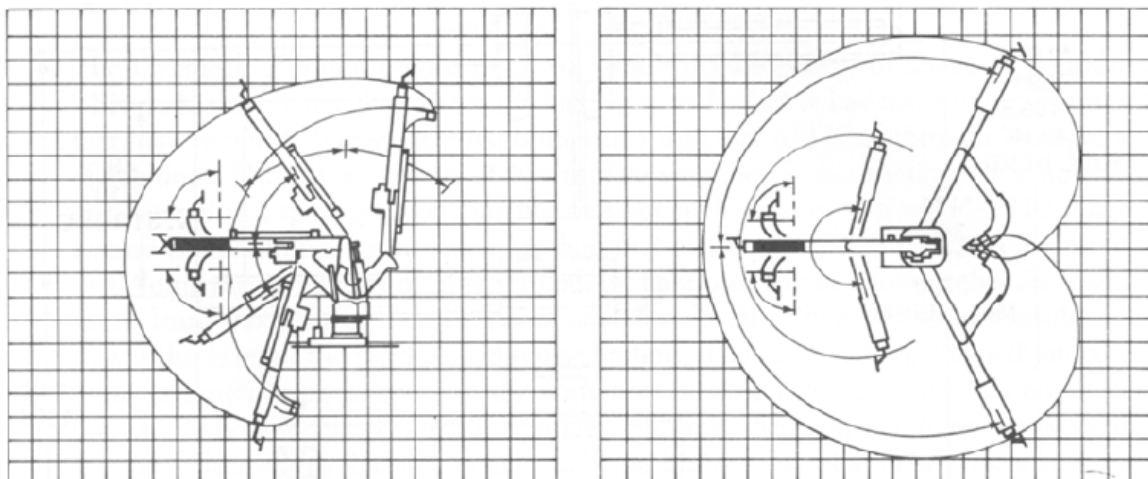
Celkový objem prostoru, ve kterém může robot dané konfigurace pracovat. Pracovní prostor se dělí na dosažitelná a pohotovostní prostor. [1,10]

Dosažitelný prostor

Je to objem pracovního prostoru, ve kterém je každý bod dosažitelný koncovým efektem alespoň při jedné jeho orientaci. [10]

Pohotovostní prostor

Je to objem pracovního prostoru, ve kterém je každý bod dosažitelný koncovým efektem při všech orientacích efektoru. [10]

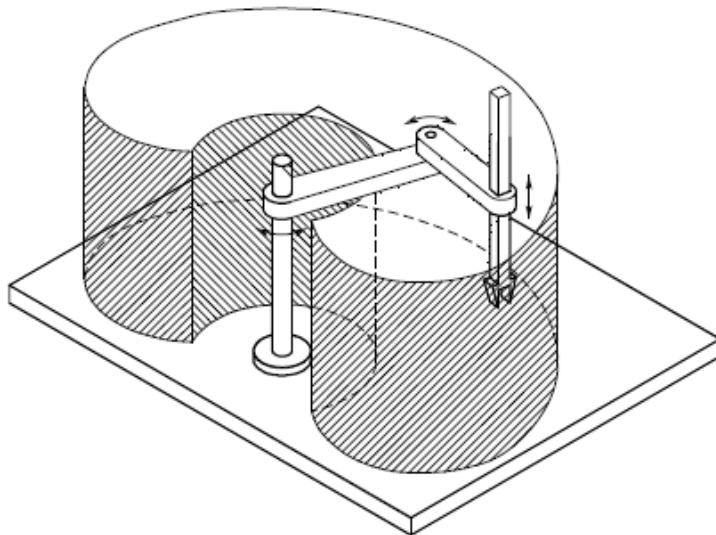


Obrázek 18 Znáornění pracovního prostoru [1]

5 USPOŘÁDÁNÍ ROBOTŮ

5.1 SCARA (Selective Compliant Arm for Robotic Assembly)

Vytvořen za montážním účelem, jeden z nejpoužívanějších. Všechny osy pohybu jsou paralelní. Mechanické rysy struktury nabízejí vysokou tuhost pro svislou zátěž a vysoký soulad pro horizontální zatížení. Tato struktura je vhodná pro vertikální montáž. Přesnost polohování zápěstí se snižuje se zvyšováním vzdálenosti zápěstí od 1. kloubu. Vhodný pro manipulaci s malými objekty. 4 stupně volnosti – 3 rotační klouby a 1 posuvný kloub. Pro horizontální práci, mikromontáž a Pick and Place tam, kde je vyžadována vysoká rychlost a vysoká přesnost. [2]



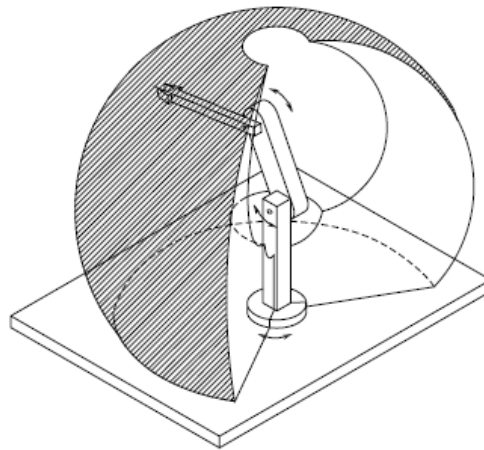
Obrázek 19 Pracovní prostor SCARA robota [2]



Obrázek 20 SCARA robot – Stäubli TS2-60 [28]

5.2 Antropomorfní

Pracovním prostorem je koule, roboty s touto kinematikou mající trojdílné rameno se třemi otočnými klouby zabírají ve srovnání s velikostí kulového pracovního prostoru nejméně místa ze všech robotů a potřebují k určitému přemístění při stejném čase ve srovnání s ostatními roboty nejmenší zrychlující síly, transformace souřadnic je ovšem nesmírně náročná. Vzhledem k podobnosti s lidskou rukou se druhý kloub nazývá rameno a třetí kloub loket. Spojuje nadloktí s předloktím. Nejobratnější a energeticky nejméně náročná. Přesnost polohy zápěstí se uvnitř pracovního prostoru mění. Akční členy elektrické. Široká oblast použití. [2]



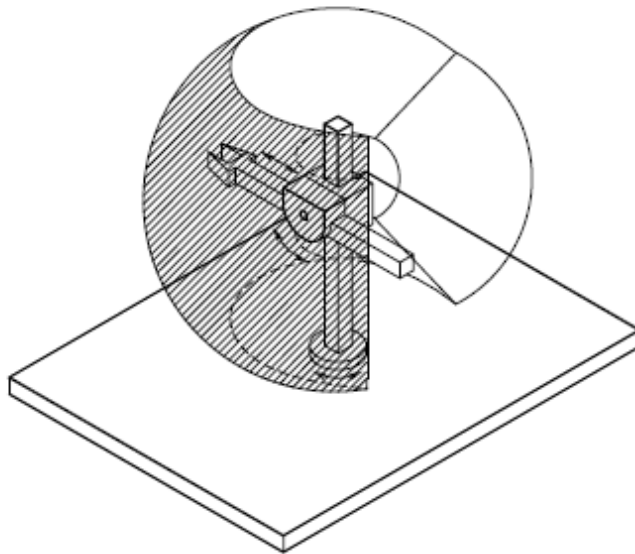
Obrázek 21 Pracovní prostor antropomorfního robota [2]



Obrázek 22 Antropomorfní robot – ABB IRB 2400 [29]

5.3 Sférický

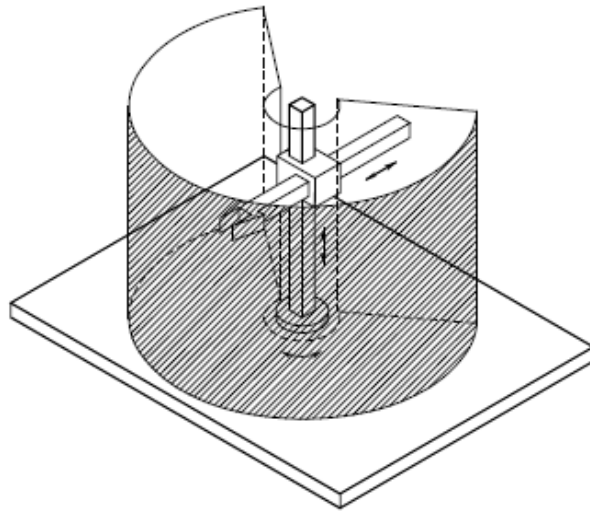
Tato konfigurace vyhovuje malým robotům. Sférický je odvozen od skutečnosti, že polohu efektoru udávají sférické souřadnice. Pracovním prostorem je dutá koule. Menší mechanická tuhost, komplexnější mechanické provedení. Přesnost polohování zápěstí se snižuje se zvyšujícím se radiálním zdvihem. Je-li přítomna podpůrná základna, manipulátor může operovat i s objekty na podlaze. Použití pro strojové obrábění. Poháněny elektrickými motory. [2]



Obrázek 23 Pracovní prostor sférického robota [2]

5.4 Cylindrický

Konfigurace vhodná pro střední zátěž efektoru. První kloub je rotační pro rotaci okolo báze a zbylé dva jsou translační. Pracovním prostorem je část dutého válce. Dobrá mechanická tuhost. Přesnost polohování zápěstí se snižuje se zvětšením horizontálního zdvihu. Jsou hlavně používány pro přepravování velkorozměrových předmětů a z tohoto důvodu je preferované použití hydraulických motorů. [2]



Obrázek 24 Pracovní prostor cylindrického robota [2]

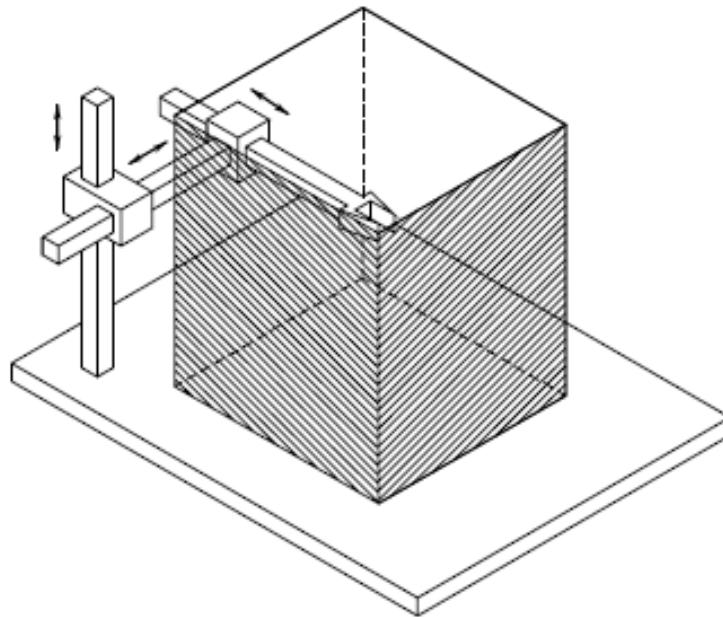


Obrázek 25 Cylindrický robot – GMF M100 [30]

5.5 Kartézský

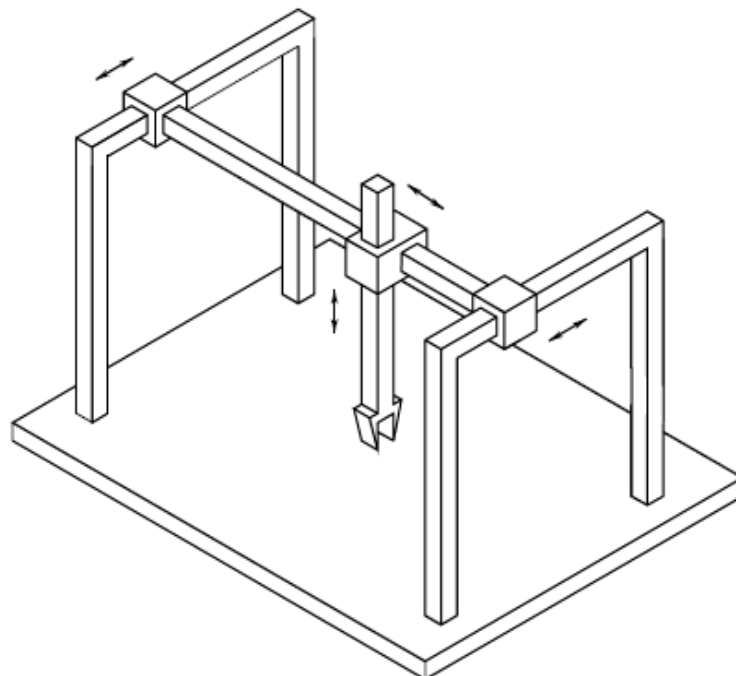
Tvořena 3 translačními klouby. Vysoká přesnost, ale nízká obratnost. Vhodné pro těžká břemena díky vysoké tuhosti, ale za cenu nízké obratnosti. Souřadnice jsou vzhledem k bázi v kartézských souřadnicích. Kinematický popis nejjednodušší. Užitečný pro aplikace montáže a portálový pro manipulaci s nákladem. Pracovním prostorem je kvádr. Přesnost polohování zápěstí je stejná v každém bodě pracovního prostoru. Při přístupu z vrchu můžeme

použít portálovou konstrukci, která je vhodnější pro těžší břemena. Při přístupu z boku se s více vysunutými články snižuje přesnost, když se manipuluje s těžkými předměty.[2]



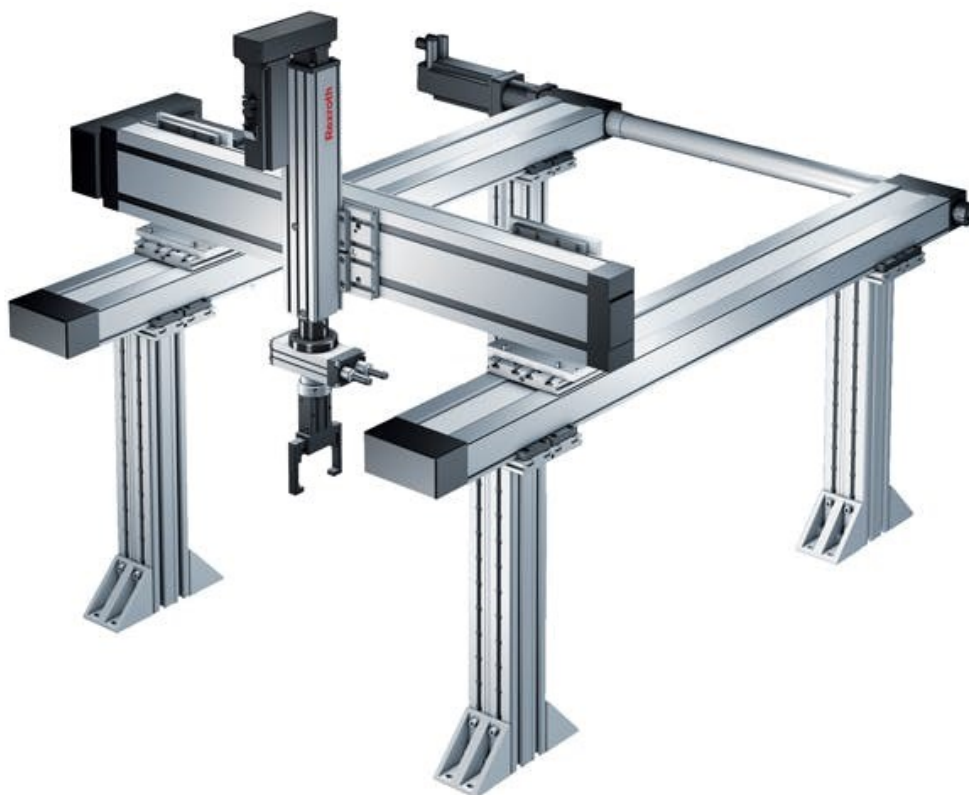
Obrázek 26 Pracovní prostor kartézského robota – přístup z boku

[2]



Obrázek 27 Schéma kartézského (portálového) robota – přístup z

vrchu [2]



Obrázek 28 Portálový robot [31]

6 STÄUBLI TX2-60

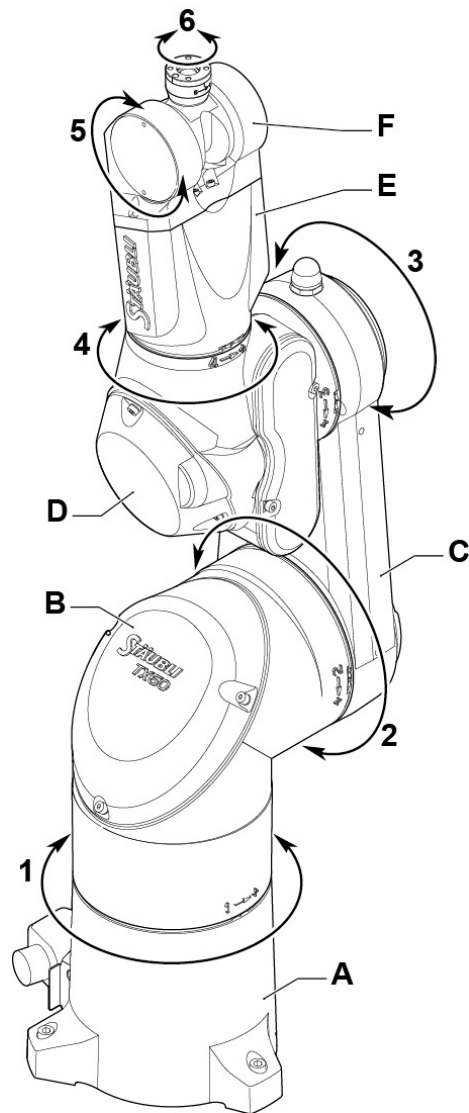
Jedná se o 6-osého průmyslového robota od švýcarské firmy Stäubli. Poskytuje ideální kombinaci rychlosti, tuhosti, rozměrů a přesnosti. Dají se aplikovat ve všech odvětvích průmyslu. Je použitelný ve vše prostředích, tj. od drsných a špinavých až po sterilní. Možnost 360° montáže pro flexibilní integraci (montáž na podlahu, stůl, stěnu nebo na strop). Má integrovaný elektrický a pneumatický uživatelský obvod včetně Cat5e (pro kameru a senzory). Lehká a pevná konstrukce, kde celé rameno je hygienicky zapouzdřeno a díky pohonům s dutou hřídelí není potřeba externích kabelů. Nízké požadavky na údržbu, žádná baterie na rameni ani v ovladači. Všechny součásti jsou snadno přístupné i vyměnitelné. Celkové náklady na vlastnictví jsou nízké a návratnost investic poměrně rychlá. Má všechny standardizované komunikační protokoly, včetně reálného času. Z výroby má tento robot implementován spoustu bezpečnostních opatření. [6]

Základní vlastnosti Stäubli TX2-60:

- Stupně volnosti - 6
- Dosah zápěstí - 920 mm
- Maximální zatížení – 3,7 kg
- Opakovatelnost - $\pm 0,03$ mm
- Stäubli controller – CS9



Obrázek 29 Stäubli TX2-60L [6]



Obrázek 30 Popis části robota

Staubli TX2-6L [7]

Rameno se skládá z prvků, které jsou vzájemně spojených klouby. Tyto prvky ramena jsou:

- Základna (A)
- Rameno (B)
- Ruka (C)
- Loket (D)
- Předloktí (E)
- Zápěstí (F)

7 CS9 CONTROLLER

Se svým designem a kapacitou nabízí stálou dostupnost. Díky otevřené architektuře je kompatibilní s produkčním světem a umožňuje rychlejší integraci. Bezpečnostní funkce jsou zabudovány pro každou aplikaci. Tímto controllerem ovládáme SCARA roboty, 6osé roboty, POWER coboty i mobilní robotické systémy. Univerzální napájecí zdroj pro více druhů napětí. Díky vyměnitelným panelům je možné měnit sady konektorů. K této platformě se dodává SP2 pendant, který představuje uživatelské rozhraní. Pendant kombinuje lehký design a robustnost s vynikající funkčností: Provoz, konfiguraci a vizualizaci pro různé případy použití (nastavení, údržba). Má integrované bezpečnostní funkce (3-polohové tlačítko, nouzová STOP). Díky multitaskingu jazyka VAL3 jsou možnosti rozšíření na knihovnách uniVAL PLC nabízející jazykovou podporu IEC 61131 PLC drive uniVAL pro plné ovládání pohybu kloubů přes externí ovladač. Rekuperuje energii při každém společném zpomalení. Má různé režimy úspory energie, které přinášejí až 40% úsporu. Všechny standardizované komunikační protokoly včetně Ethernetu v reálném čase. Vestavěný webový server založený na html pro uživatelský systém a rozhraní, která jsou dostupná na jakémkoli zařízení. K dispozici EtherCat pro modulární připojení k periferním zařízením. [7][8]



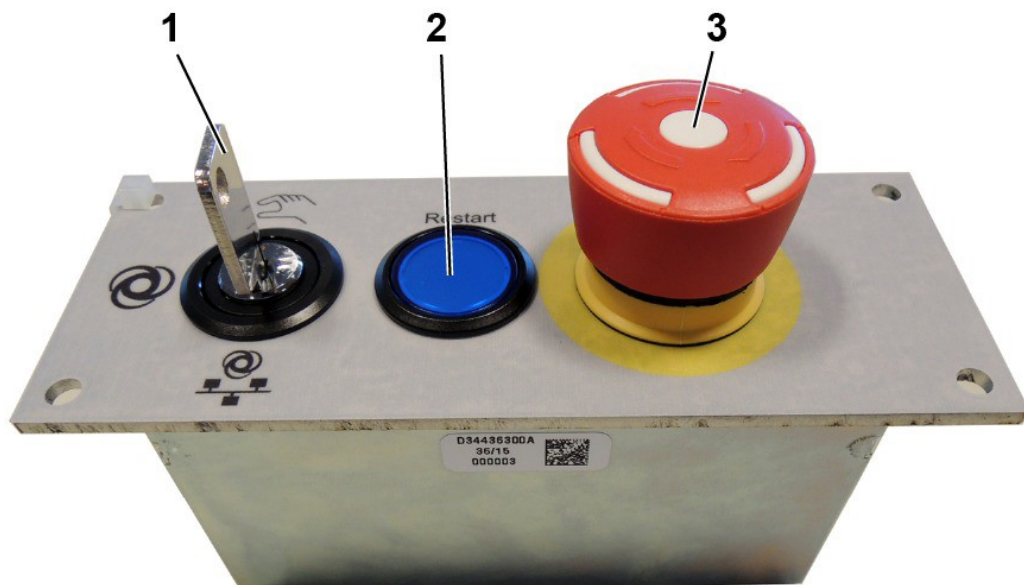
Obrázek 31 Controller CS9 [7][8]



Obrázek 32 Pendant [7]

1 – Nouzové tlačítko STOP – donutí robota přejít do bezpečného stavu.

2 – 3 polohové tlačítko, které je potřeba držet v manuálním režimu, aby byl robot schopen opustit bezpečný stav.



Obrázek 33 Ovládací panel Controlleru [7]

1 - 3 polohový přepínač na klíč – změnu pracovního módu je možné pouze se zasunutým klíčem.

2 – Resetovací tlačítko pro povolení restartu po bezpečnostním zastavení

3 – Nouzové tlačítko STOP

8 ÚLOHY NA PROCVIČENÍ

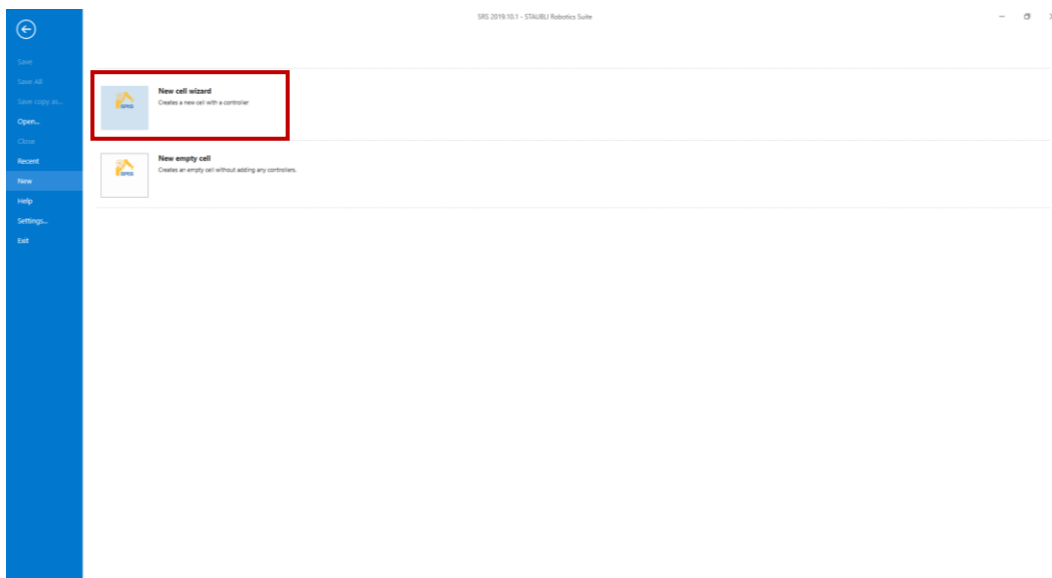
Úlohy jsou navrženy, aby se studenti naučili pracovat ve vývojovém prostředí *Stäubli Robotics suite*. Dále se naučí pracovat s 3D vizualizací prostředí, díky které se dále naučí základní principy fungování robotu, cyklů a uživatelských programů. Každá z úloh je zaměřena na určitý typ problému, který je následně pečlivě popsán krok za krokem v manuálu. Studenti po získání dovedností se simulací budou obeznámeni s prací na reálném robotu, ke které vytvořené úlohy spějí.

II. PRAKTICKÁ ČÁST

9 PRÁCE VE STÄUBLI ROBOTICS SUITE

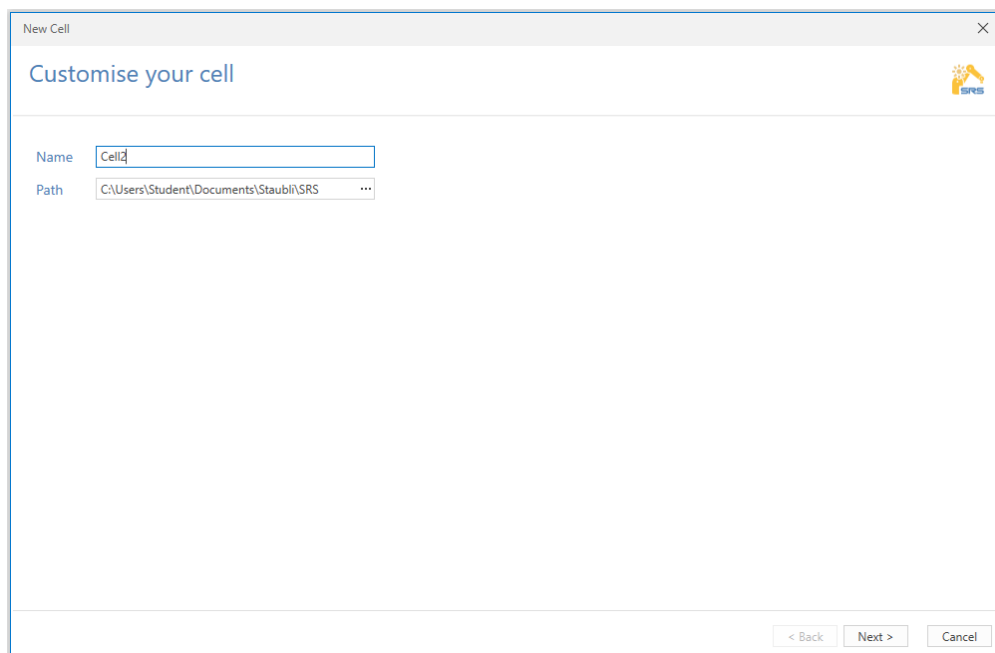
9.1 Založení nového projektu

Pro založení nového projektu vybereme záložku *New* a dále zvolíme *New cell wizard*.



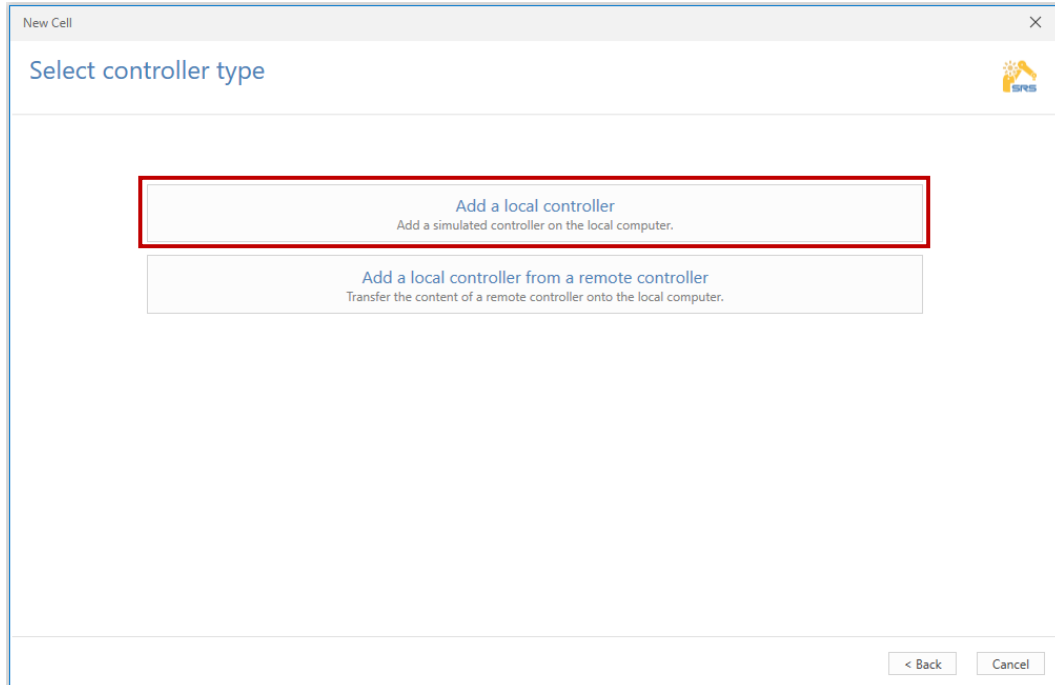
Obrázek 34 Vytvoření nového projektu

Do následujícího okna vyplníme název a uvedeme cestu, kam se bude přistupovat k projektu.



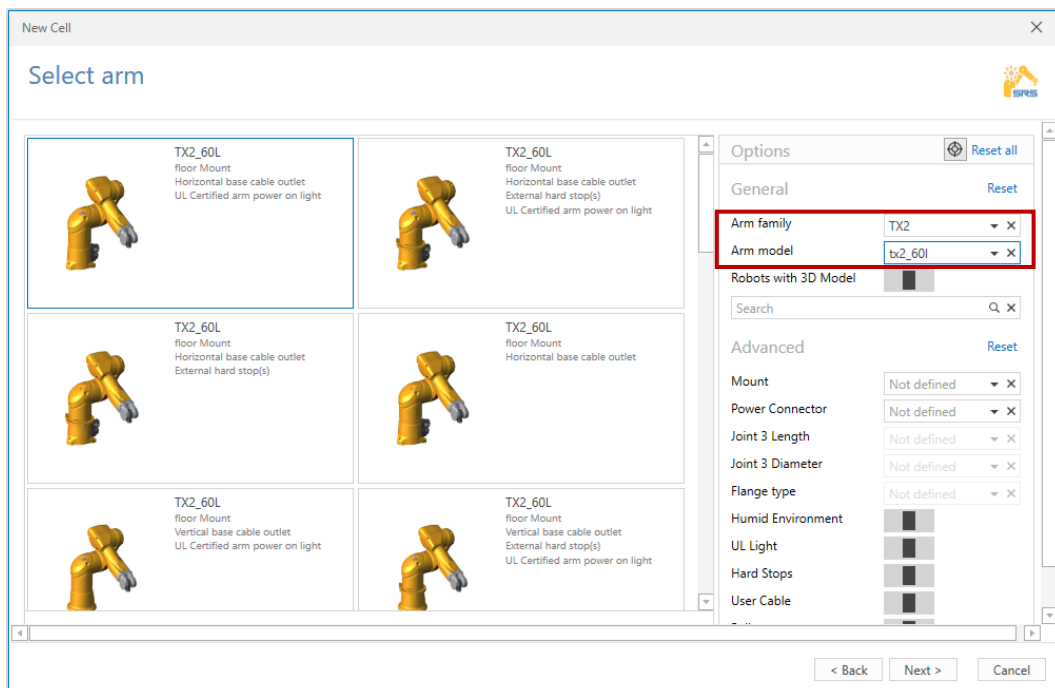
Obrázek 35 Pojmenování nového projektu

Zde volíme mezi možnostmi přidání controlleru manuálně nebo zda bude nahrán z již existujícího controlleru. Druhá možnost se používá, když máme reálného robota a chceme nahrát právě jeho controller.



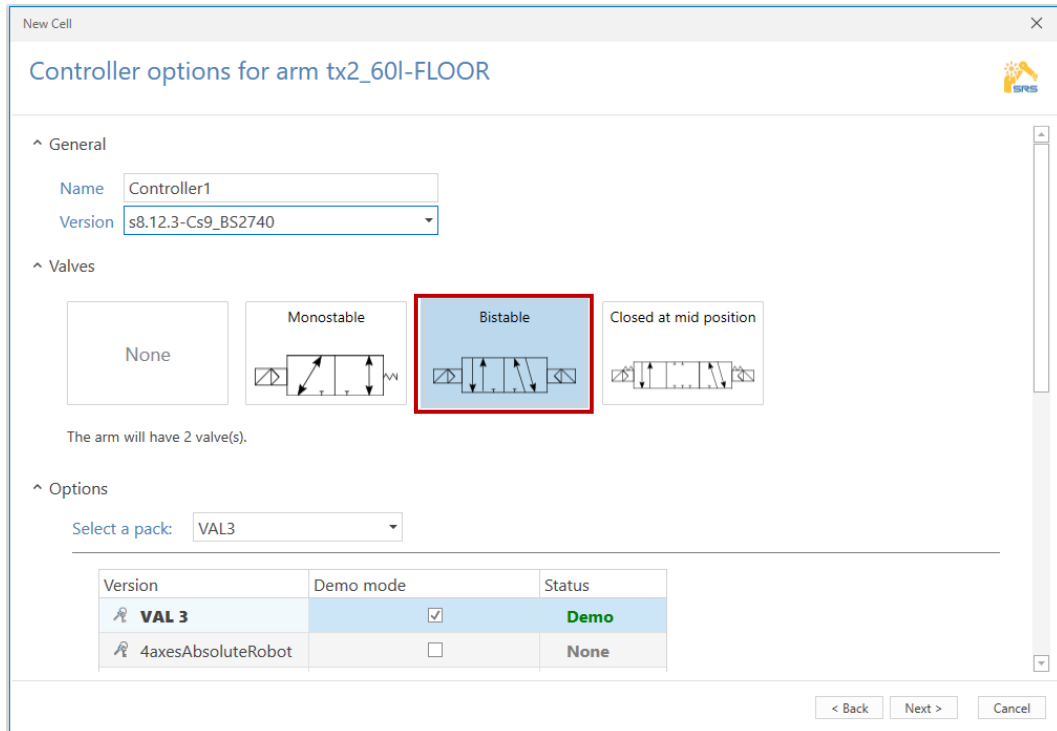
Obrázek 36 Přidání controlleru

Nyní volíme typ robota, kterého chceme přiřadit ke controlleru. V *Arm family* zvolíme *TX2* a v *Arm model* volíme *tx2_60l*.



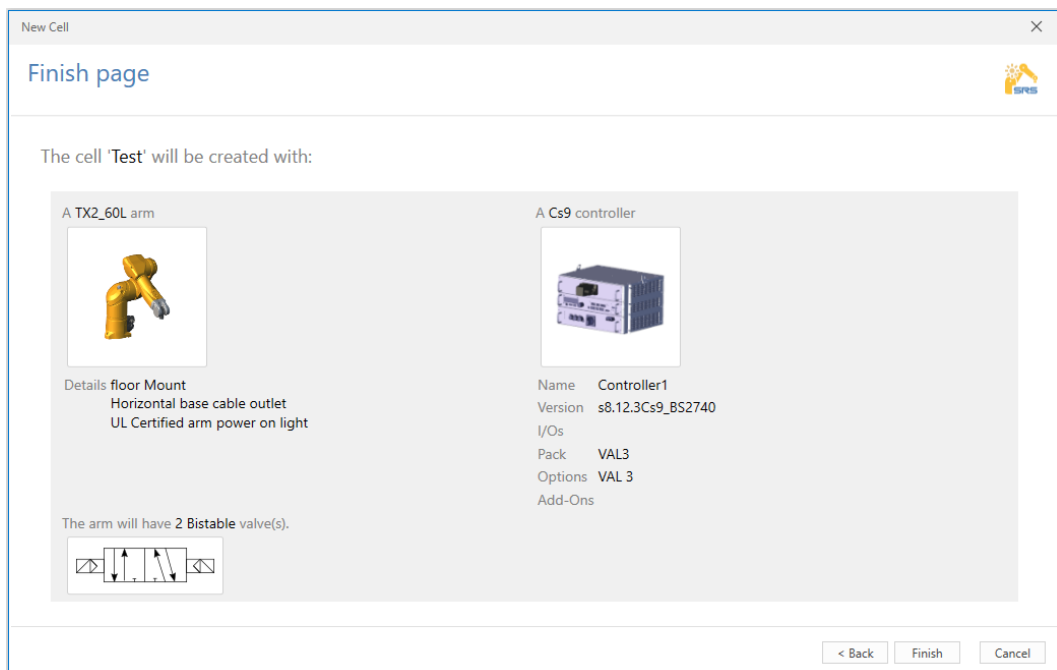
Obrázek 37 Volení správného typu robota

Zde pojmenujeme controller a zvolíme jeho verzi. Verze může být změněna i v samotném projektu, ale je lepší volit verzi, která je na reálném robotu. V sekci **Valves** zvolíme **Bistable**. V **Options** zaškrtneme pouze **VAL3**.



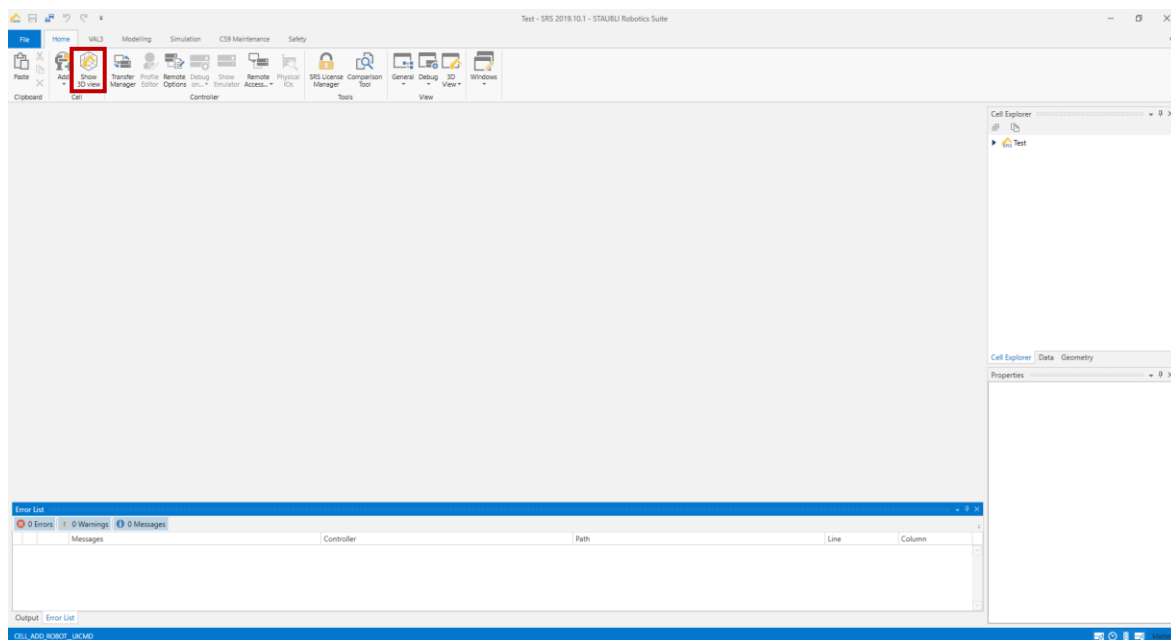
Obrázek 38 Rozšířené nastavení controlleru

Nyní si můžeme překontrolovat zvolené možnosti.



Obrázek 39 Kontrola navoleného projektu

Nyní máme připravený projekt. Pomocí **Show 3D View** v záložce **Home** a odstavci **Cell** zapneme 3D vizualizaci.



Obrázek 40 Zapnutí 3D vizualizace

9.2 Základní ovládací prvky

V bloku č.1 jsou nejdůležitější možnosti **Add**, **Show 3D view**, **3D view** a **Transfer manager**.

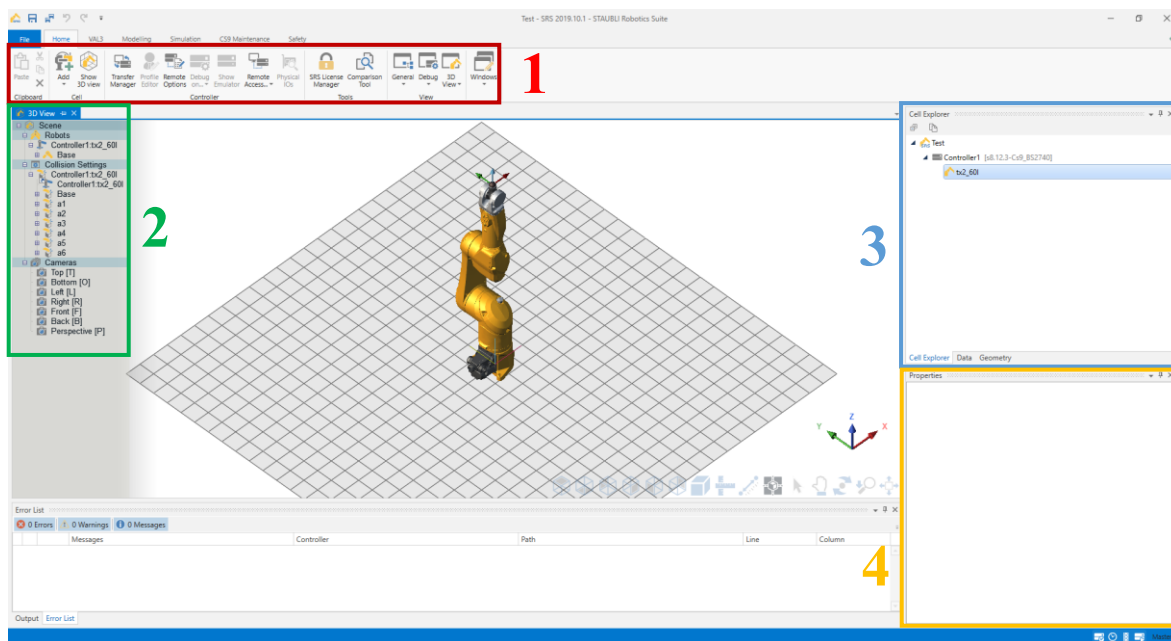
- **Add** – pro přidání nového robota.
- **Show 3D view** – zobrazí vizualizaci pracovního prostředí.
- **3D view** – najdeme zde **Edit Position**, **Device Configuration** a **Jog**.
- **Transfer manager** – poskytuje možnost vytvořit **Backup** robota nebo nahrání programu do robota.

V bloku č.2 je strom aktuálně používaných robotů, těles, kolizí a předvoleb kamery. Po přidání nějakého objektu jej nalezneme právě v tomto stromě.

V bloku č.3 máme 3 záložky:

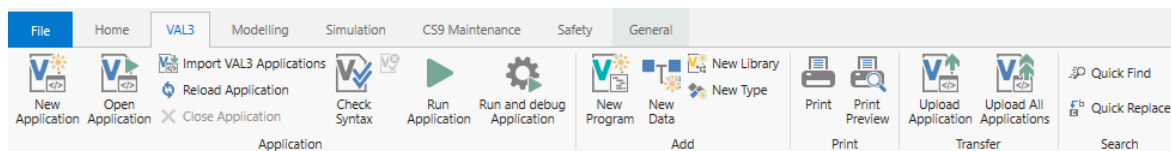
- **Cell Explorer** – zde najdeme uživatelské aplikace, knihovny, podprogramy a interface.
- **Data** – zde najdeme proměnné, které se využívají buď v programu nebo v interface.
- **Geometry** – zde nalezneme deklarované body, souřadné systémy a rozměry nástroje.

V bloku č. 4 najdeme vlastnosti vybraného objektu.



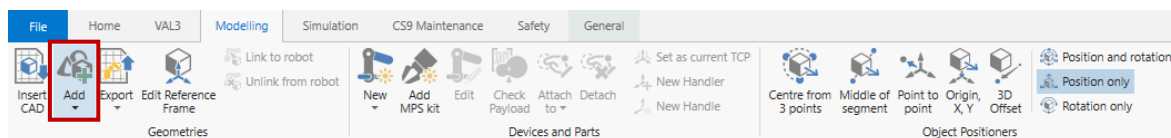
Obrázek 41 Znárodnění hlavních ovládacích prvků

V kartě **VAL3** se budeme nacházet, když budeme psát program pro pohyb robotu. Jsou nabízeny funkce vytvoření/otevření aplikace, kontroly syntaxu, spuštění aplikace, vytvoření nového programu, knihovny či proměnné.



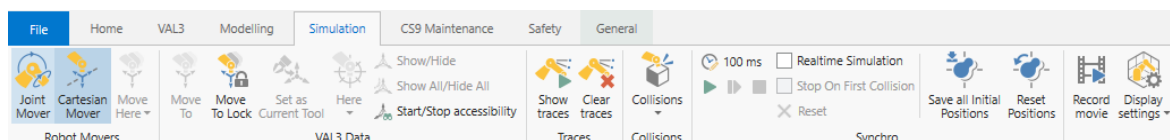
Obrázek 42 Karta VAL3

V kartě **Modelling** můžeme vložit CAD objekt, který nám importuje již vymodelovaný objekt. Dále můžeme pomocí **Add** přidávat základní objekty, jimiž jsou válec, kostka a koule. Dále můžeme vyexportovat vymodelovaný objekt nebo upravit jeho referenční frame, který je výchozím bodem objektu. Pomocí **New** převedeme objekt na nový nástroj nebo součást. Odstavec **Object Positioners** nabízí polohování pro pozicování bodů, framů nebo objektů.



Obrázek 43 Karta Modelling

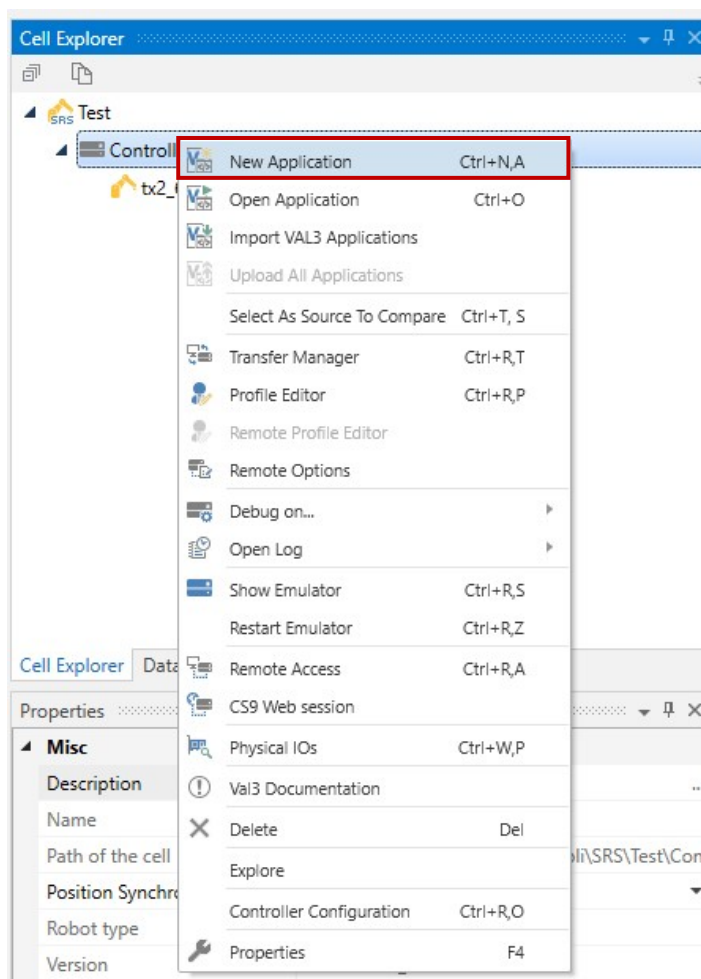
V kartě *Simulation* najdeme možnosti zobrazení typů pohybu pomocí kartézských souřadnic (*Cartesian mover*) nebo pomocí úhlového pootočení (*Joint mover*). Dále v odstavci *Synchro* pomocí zeleného tlačítka *Start synchro* zapneme synchronizaci se spuštěným programem. *Save all Initial Positios* a *Reset Positions* slouží pro uložení a resetování pozic součástí, se kterými se v programu pracuje a pohybuje.



Obrázek 44 Karta Simulation

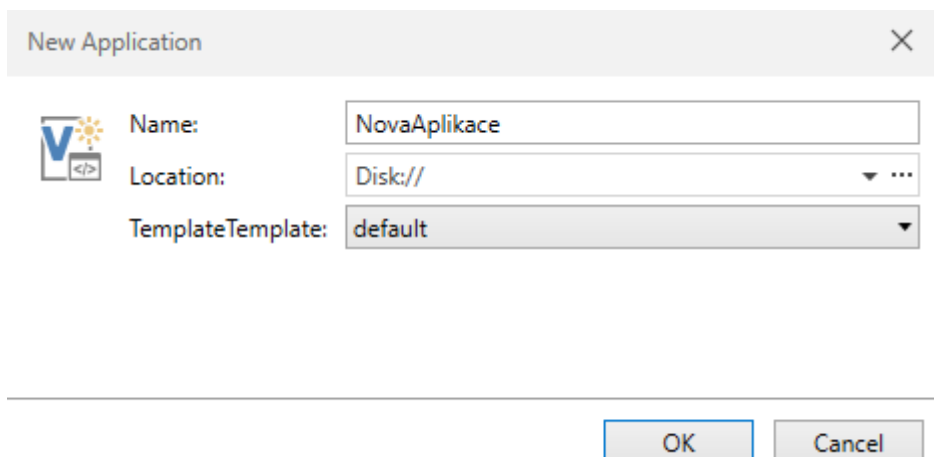
9.3 Založení nové aplikace a vytvoření objektů

Novou aplikaci vytvoříme pomocí pravého kliknutí na *Controller1* a zvolením *New Application*.



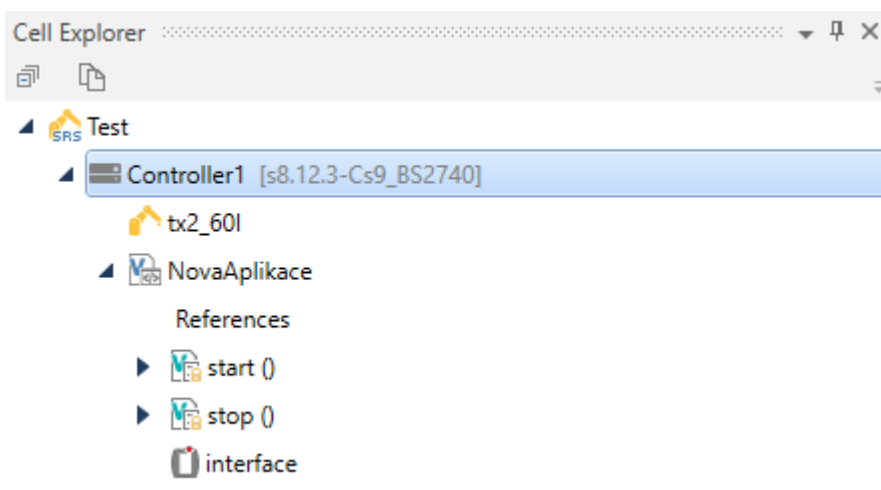
Obrázek 45 Vytvoření nové aplikace VAL3

Nyní svou aplikaci pojmenujeme a zvolíme **OK**

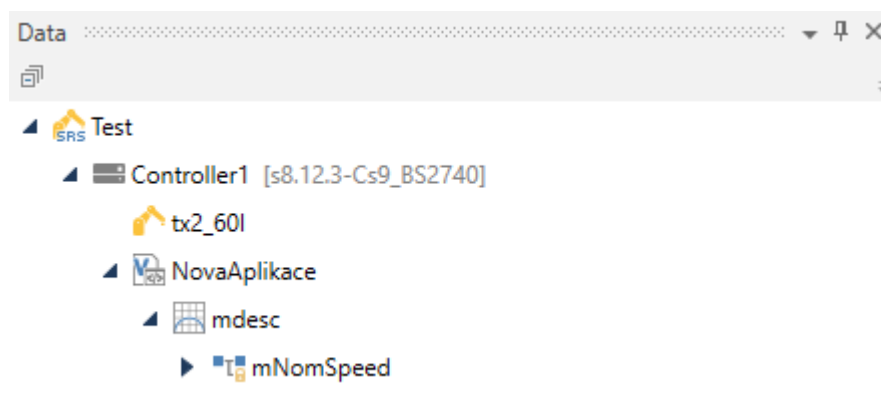


Obrázek 46 Pojmenování nové aplikace

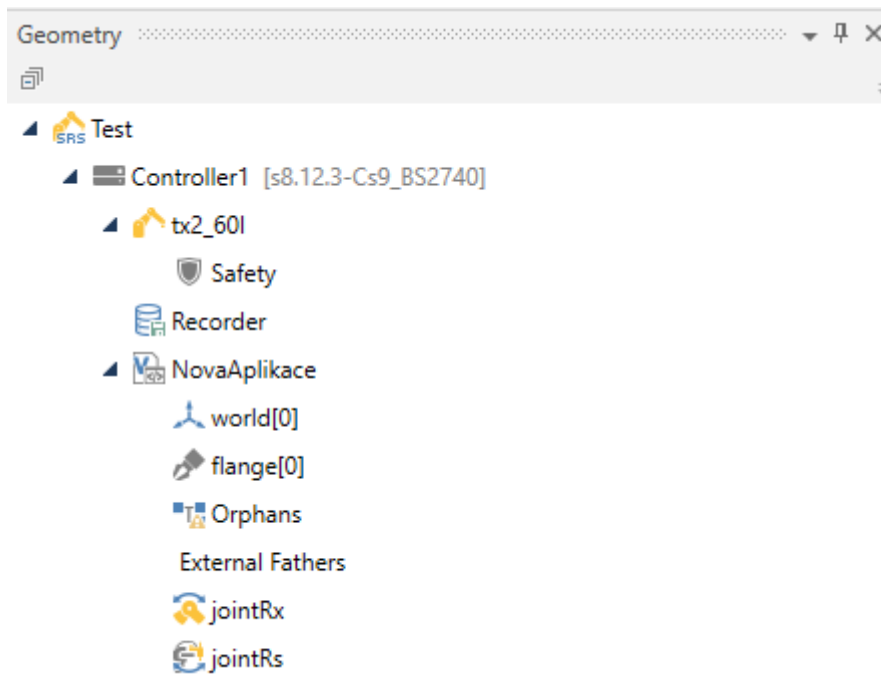
Po vytvoření aplikace můžeme v kartách **Cell Explorer**, **Data** a **Geometry** najít již zmíněné programy, proměnné a pole bodů.



Obrázek 47 Nová aplikace – karta Cell Explorer

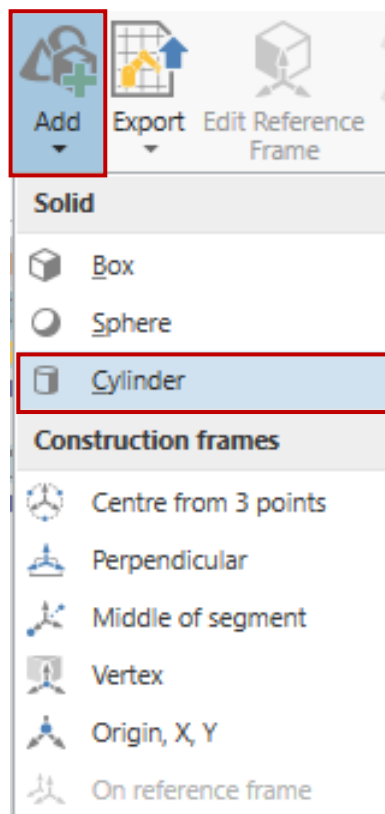


Obrázek 48 Nová aplikace – karta Data



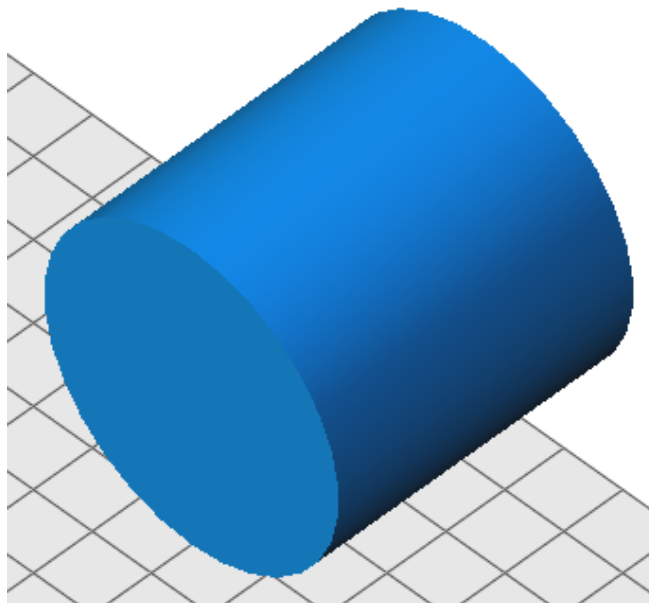
Obrázek 49 Nová aplikace – karta Geometry

Pro vytvoření nového objektu přejdeme do záložky **Modelling** a přes možnost **Add** a zvolíme si **Cylinder**.



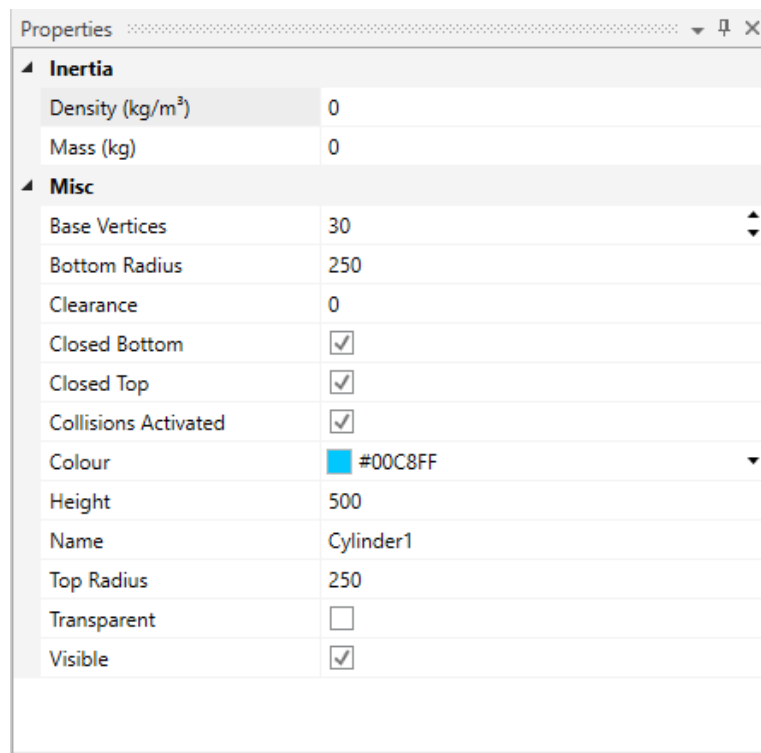
Obrázek 50 Vytvoření nového Cylinderu

Vytvoří se válec, který má oba rádiusy defaultně na 250 mm a výšku 500 mm s modrou barvou.



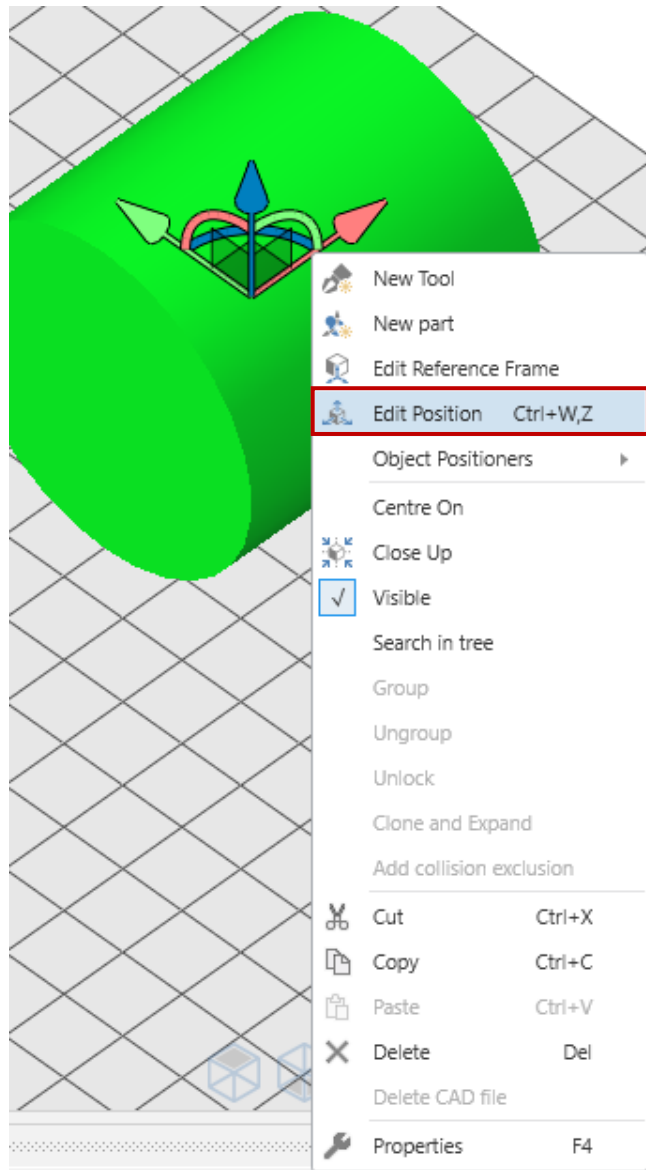
Obrázek 51 Vytvořený objekt – Cylinder

Každý vytvořený objekt má své vlastnosti, které se dají měnit pomocí karty **Properties**. Tělesu můžeme přiřadit jeho hmotnost a hustotu. Dále můžeme měnit jeho rozměry, barvu, název, průhlednost a viditelnost.



Obrázek 52 Vlastnosti Cylinderu

Při označení objektu a stisknutím pravého tlačítka dostaneme možnosti, které můžeme využít pro daný objekt. Nyní zvolíme *Edit Position*.



Obrázek 53 Lišta možnost objektu - Edit Position

Dostaneme okno pro editaci pozice zvoleného objektu. Vidíme 2 tabulky:

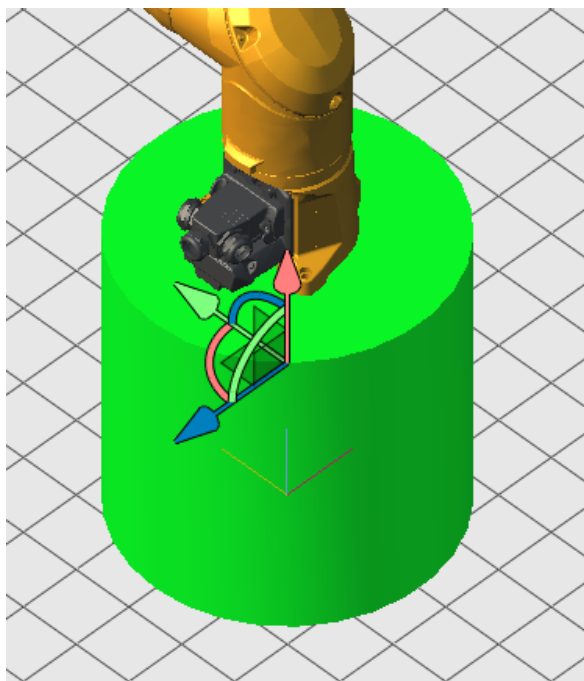
- *Relative position* – relativní posunutí/pootočení od bodu, kde jsme objekt označili.
- *Absolute position* – absolutní posunutí/pootočení od počátku souřadnic.

X , Y , Z jsou lineární posuny po svých osách, R_x , R_y a R_z jsou pootočení ve stupních okolo své osy.

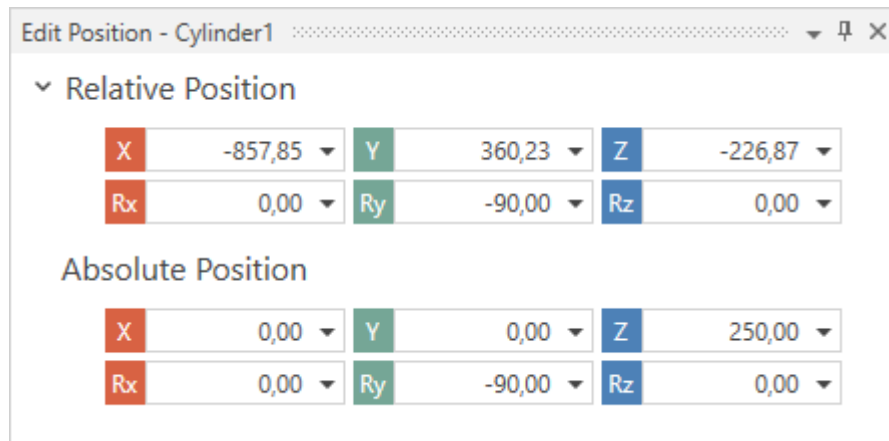


Obrázek 54 Okno editace pozice objektů

Pro ukázkou přesuneme tento válec tak, aby byl vlastní osou v počátku a aby byl na válci položen robot. Nejprve válec přesuneme do počátku. Toho nejjednodušeji dosáhneme zápisem do X , Y , Z souřadnice 0,0,0. Poté je potřeba otočit válec podle osy Y o -90 stupňů. Válec má manipulační bod ve středu válce, tudíž jej posuneme o 250 mm v ose Z .



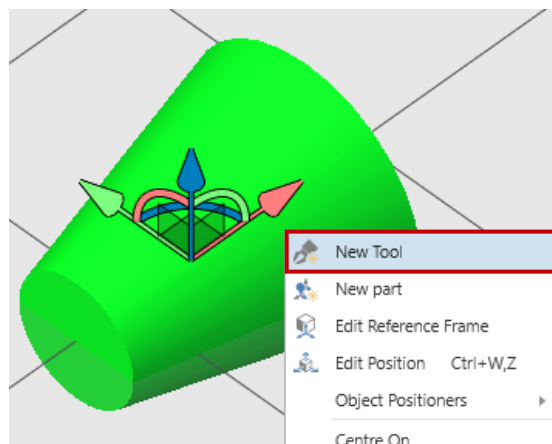
Obrázek 55 Posunutý válec ve 3D vizualizaci



Obrázek 56 Posunutí válce – okno Edit Position

9.4 Vytvoření nástroje a součástek

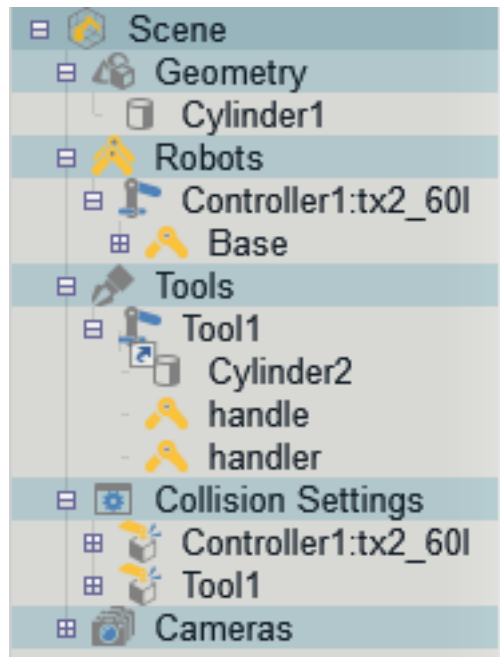
Stäubli Robotics Suite nenabízí širokou škálu možností na simulování funkčnosti úchopných hlav, proto vytvoříme imitaci nástroje. To provedeme vložením nového *Cylinderu*, kterému nastavíme výšku 100 mm, *Top Radius* na 50 mm a *Bottom Radius* na 25 mm. Pravým kliknutím myši se ukáže nabídka a vybereme možnost *New Tool*.



Obrázek 57 Převedení objektu na nástroj

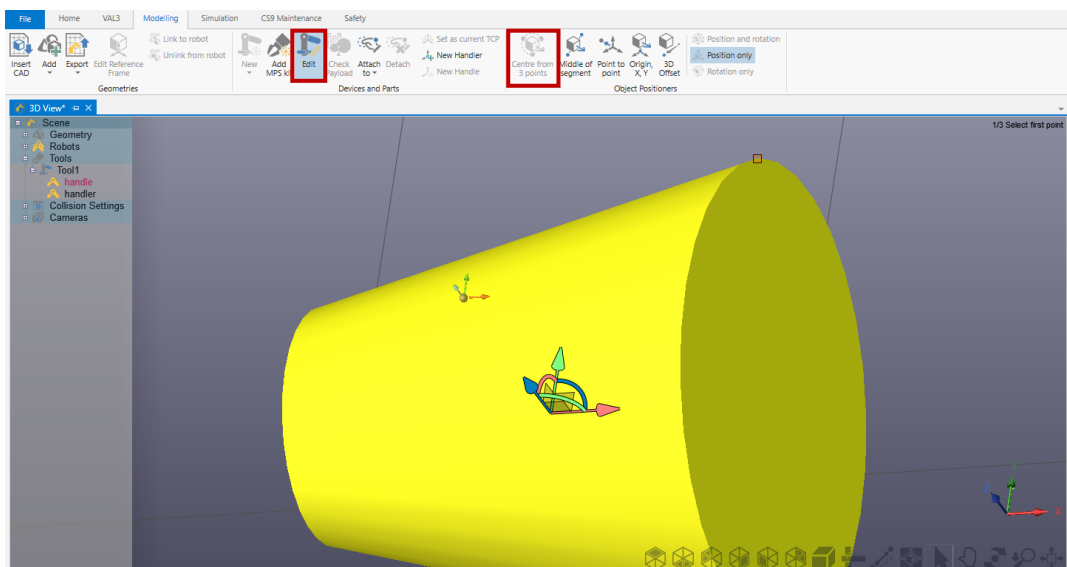
Ve stromu objektů vizualizace se nám objeví složka *Tools*, v ní *Tool1* a v ní *handler* a *handle*.

- *handler* – místo, kde se uchycuje nástroj nebo součást.
- *handle* – místo, kde se součást nebo nástroj připojuje k robotu nebo k dalšímu nástroji.



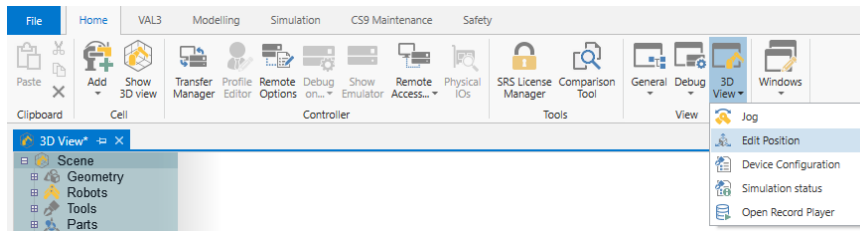
Obrázek 58 Strom vizualizace

Nyní musíme novému nástroji nastavit *handle* a *handler*. Označíme si vytvořený nástroj a v kartě *Modelling* v odstavci *Devices and Parts* zvolíme *Edit*. Ve stromě označíme *handle* a v odstavci *Object Positioners* vybereme *Center from 3 points*. Pomocné čtverečky budou ukazovat nejbližší vrchol, který lze použít jako bod. V pravém horním rohu vizualizace nalezneme našeptávač.

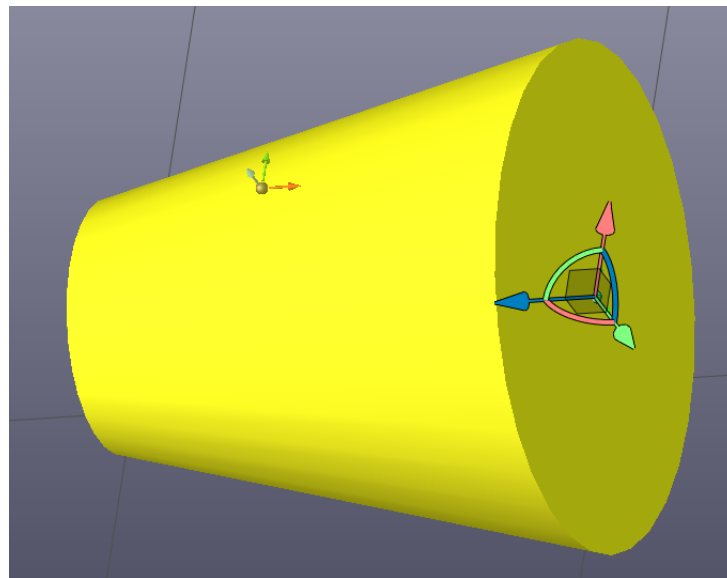


Obrázek 59 Editace nástroje – nastavení handle

Po umístění všechny 3 bodů po kružnici se nám *handle* přesune a přeorientuje do požadované orientace pomocí *Edit Position*. Stejný proces opakujeme s *handler*, akorát na opačném konci. Změny potvrdíme kliknutím na *Edit*.

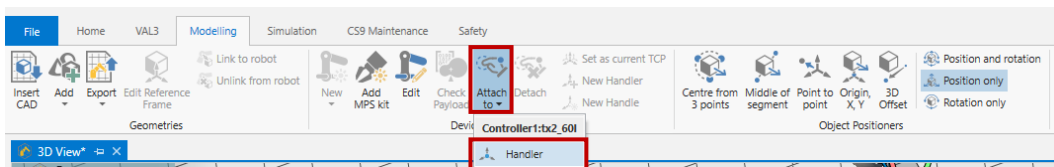


Obrázek 60 Zapnutí Edit Position

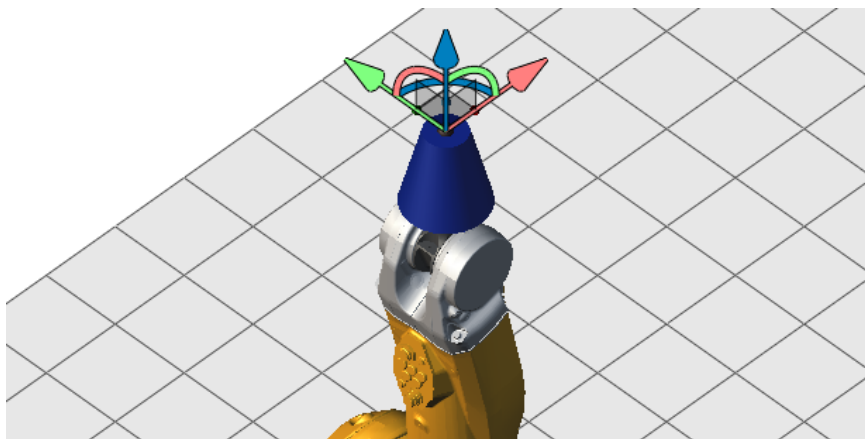


Obrázek 61 Správná orientace handle

Nástroj můžeme k robotu připevnit přes funkci *Attach to* a volíme *handler*.

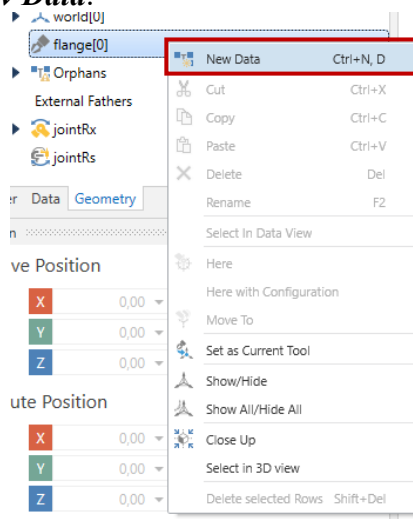


Obrázek 62 Připevnění nástroje k robotu



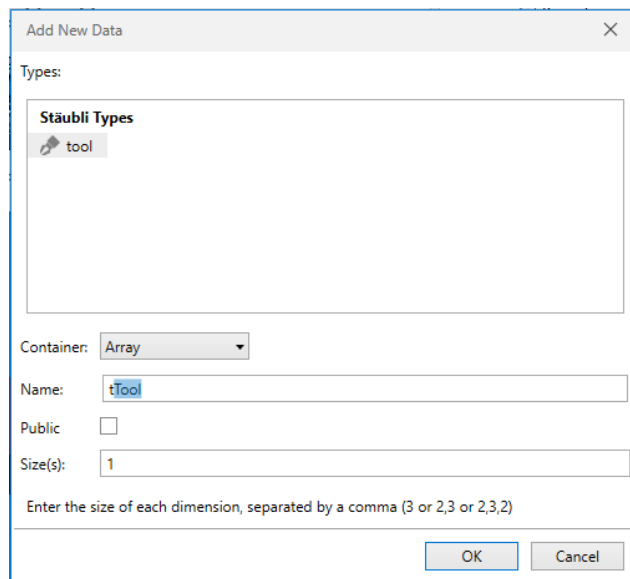
Obrázek 63 Nástroj připevněný k robotu

Nyní musíme v kartě **Geometry** vytvořit datový typ nástroje. To provedeme pravým kliknutím na **flange** a zvolíme **New Data**.



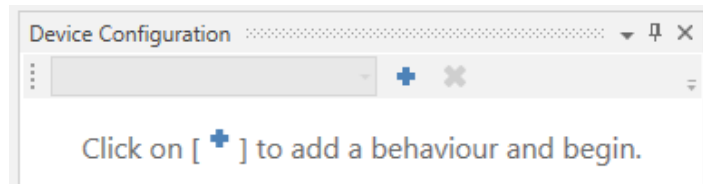
Obrázek 64 Vytvoření nových dat do položky flange

Nyní vybereme datový typ **tool** a pojmenujeme jej. Je potřeba jej pojmenovat s prvním písmenem **t** jako **tTool**.

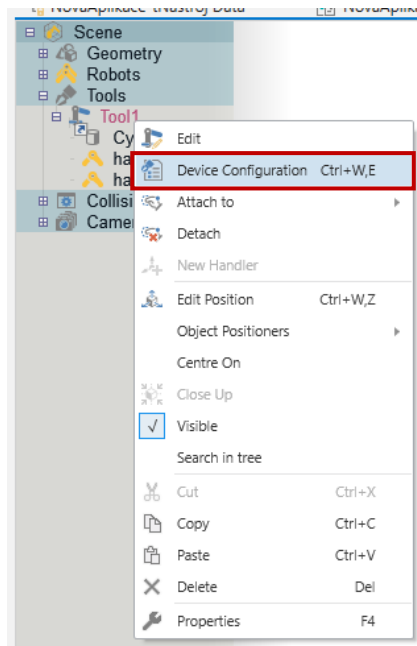


Obrázek 65 Tabulka pro vytvoření nástroje

Nyní ve stromě objektů najdeme **Tool1** a přes pravé kliknutí zvolíme **Device Configuration**. Nejprve klikneme na modré plus, poté vyskočí další okno.

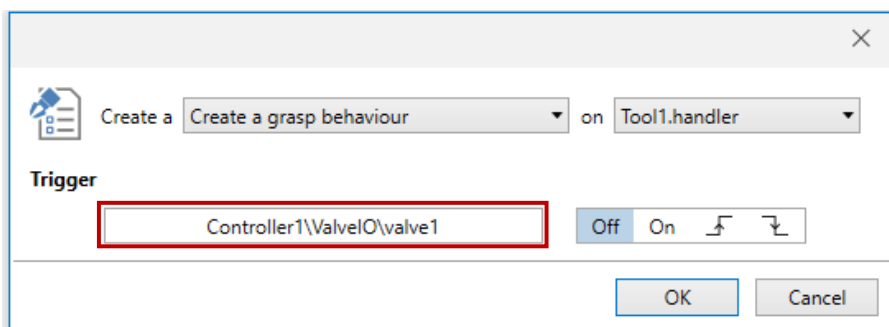


Obrázek 66 Přidání chování nástroje



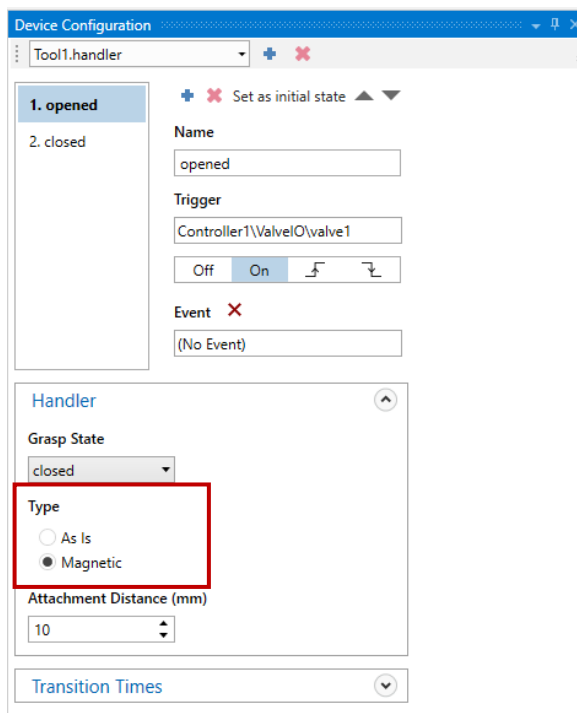
Obrázek 67 Otevření konfigurace nástroje

Vyskočí okno, kde je potřeba zkontrolovat, zda je zvolen správný nástroj. Dále je potřeba nastavit tzv. **Trigger** nástroje. To znamená, co gripper ve skutečnosti ovládá. Cesta k zařízení je **Controller1\ValveIO\Digital Outputs\valve1**.



Obrázek 68 Tabulka konfigurace

Nyní vyskočí další okno **Device Configuration**. Nejprve Zde je potřeba změnit **Type** z **As Is** na **Magnetic**.



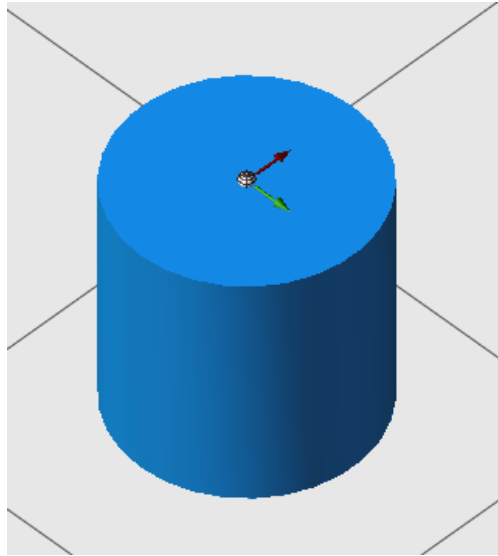
Obrázek 69 Poslední okno konfigurace nástroje

Nyní v **Geometry** dvojklikem na nástroj otevřeme vlastnosti nástroje. Ze stromu objektů v záložce **Tools** volíme **Tool1** a pravým tlačítkem klikneme na **handler** a vybereme **Copy Tool Data**. Nakopírovaná data nyní vložíme do otevřené záložky našeho vytvořeného nástroje. V sekci **Gripper** je nutno zadat cestu k ovládacímu prvku gripperu. Tu najdeme ve **Controller1\ValveIO\Digital Outputs\valve1**.

| Index | X | Y | Z | Rx | Ry | Rz | Gripper | OTime | CTime |
|-------|---|---|-----|----|----|----|----------------|-------|-------|
| 0 | 0 | 0 | 100 | 0 | 0 | 0 | ValveIO\valve1 | 0 | 0 |

Obrázek 70 Nastavení rozměrů a ovládání nástroje

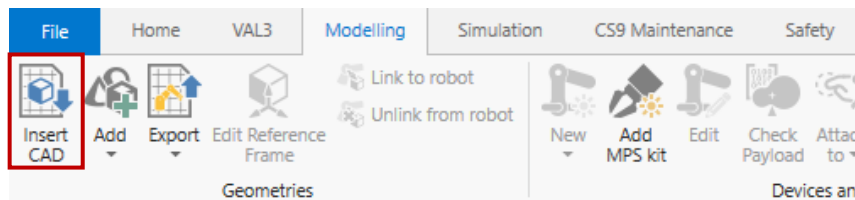
Pro vytvoření nové součástky můžeme objekt, ze kterého bude součástka vytvořena, buď naimportovat nebo vytvořit. Byl vytvořen válec s rádiusy 30, výškou 60 a jeho orientace byla změněna tak, aby válec byl orientován na výšku. Nyní v kartě **Modelling** v odstavci **Devices and Parts** zvolíme **New** a vybereme možnost **Part**. Vývojové prostředí automaticky změní objekt na součástku a automaticky vygeneruje orientaci **Handle**.



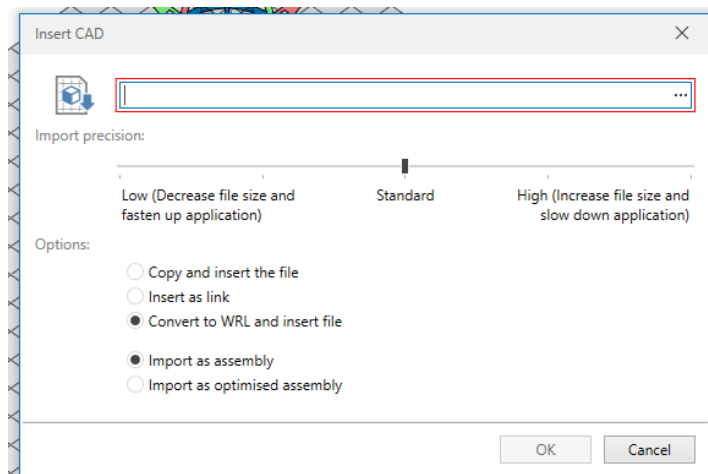
Obrázek 71 Vytvořený Part

9.5 Vložení objektu

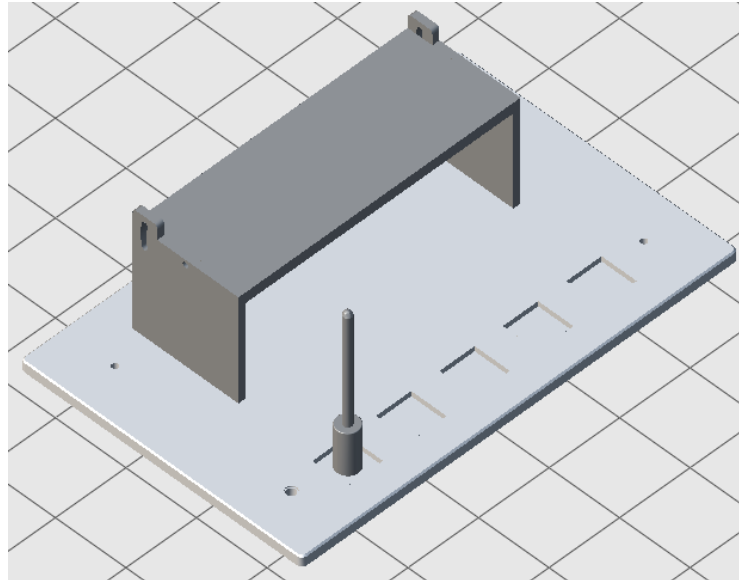
V kartě *Modelling* v odstavci *Geometries* volíme *Insert CAD*. Následovně vyskočí tabulka, kde je potřeba vyplnit cestu k požadovanému souboru. Poté je volba z možností ohledně velikosti souboru, jakým způsobem se objekt vloží a zda se každá část objektu vloží jako jednotlivý komponent nebo jestli se některé komponenty sjednotí.



Obrázek 72 Využití možnosti vložení CAD objektu

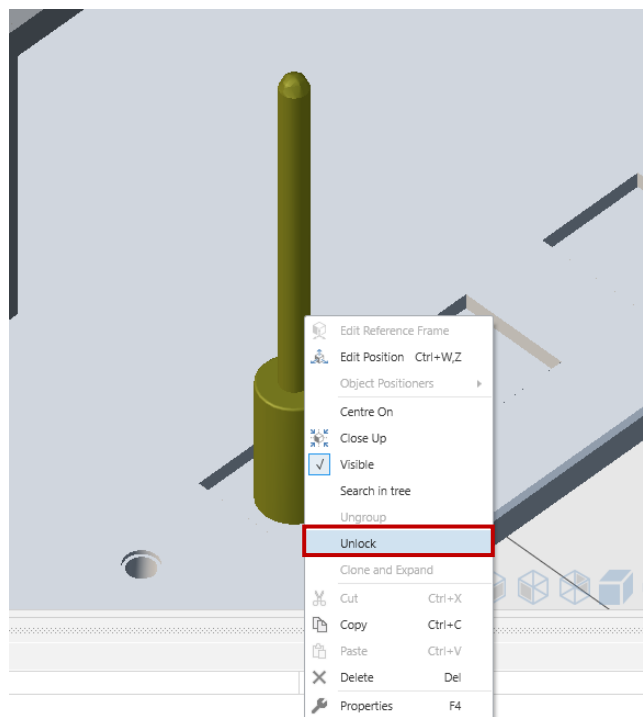


Obrázek 73 Tabulka pro vložení CAD objektu



Obrázek 74 Vložený CADovský objekt

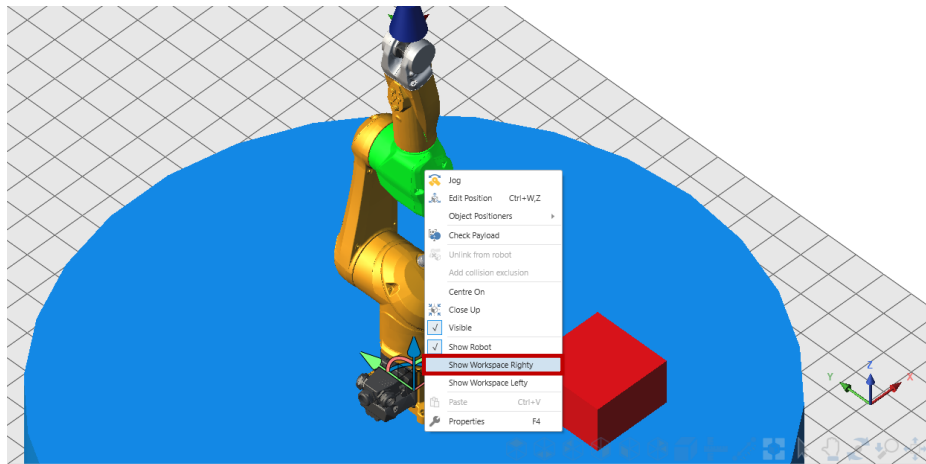
Některé objekty se mohou skládat z více komponentů. Tyto komponenty lze vybrat buď dvojklikem nebo vybráním ve stromě objektů. Komponenty je možné ze sestavy objektu odebrat nebo uvolnit. Uvolnění (**Unlock**) umožní samostatnou manipulaci s komponentem.



Obrázek 75 Označený komponent a volba uvolnění komponentu ze sestavy

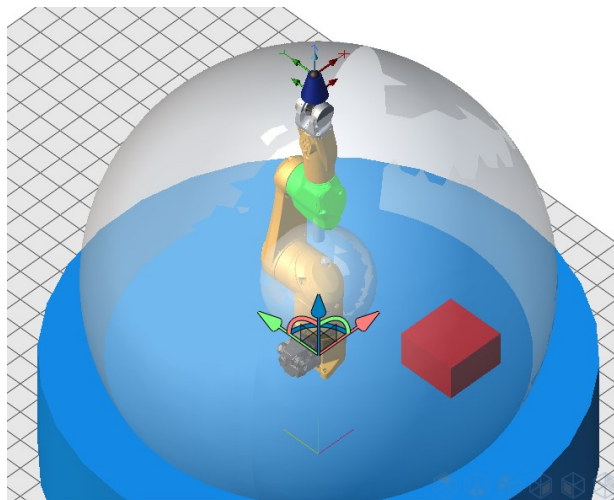
9.6 Vytvoření bodu a vlastního souřadného systému

Mějme vytvořené pracovní prostředí, kde se nachází robot, pracovní stůl a kvádr, jehož rohy budeme momentálně brát za body trajektorie. Pro ověření, že je kvádr v pracovním prostoru robotu, je zde možnost zobrazení pracovního prostoru. To provedeme označením a pravým kliknutím na robota a zvolíme *Show Workspace Rightly* nebo *Show Workspace Lefty*.



Obrázek 76 Volba zobrazení pracovního prostoru

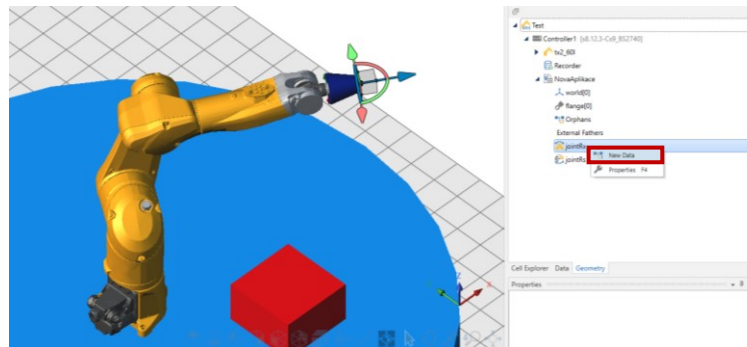
Na následujícím obrázku vidíme pracovní prostor a ujistíme se, že se kvádr nachází v pracovním prostoru robotu.



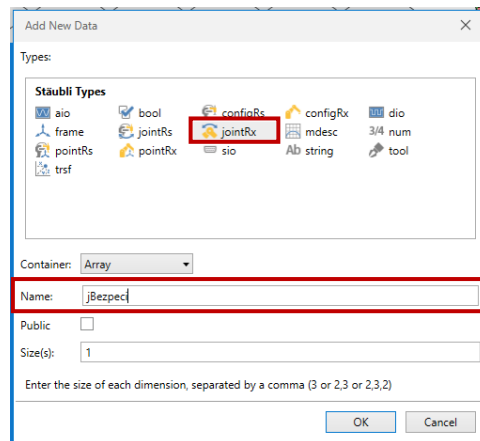
Obrázek 77 Zobrazení pracovního prostoru robota

Nyní si vytvoříme výchozí bod robota, které se bude považovat za bezpečné místo. Nejprve ale je potřeba napolohovat robota tak, aby jeho koncový efektor byl na místě, kde chceme mít bod bezpečí. V kartě *Geometry* ve zvolené aplikaci najedeme na *jointRx* (*jointRs* je pro

SCARA roboty). Joint je pro rotační pohyb. Do názvu bodu je zvykem dát první písmeno jako označení, o který pohyb/proměnnou se jedná.

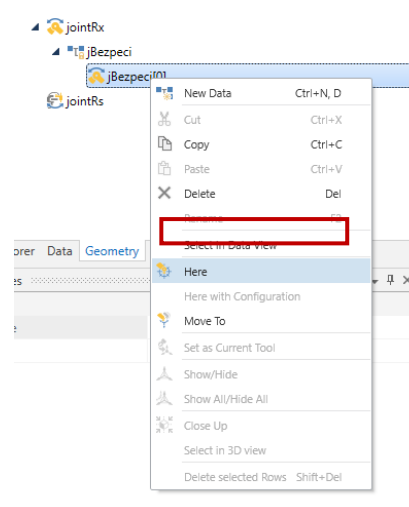


Obrázek 78 Vytvoření nového bodu v jointRx



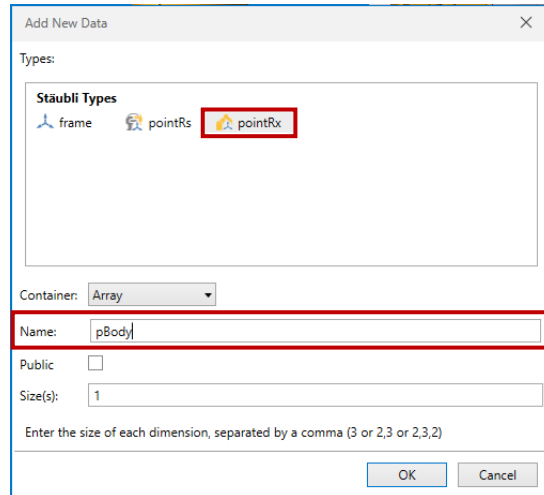
Obrázek 79 Tabulka pro vytvoření nového bodu

Nově vytvořený bod můžeme přenaučit pomocí pravého tlačítka a zvolení *Here*.



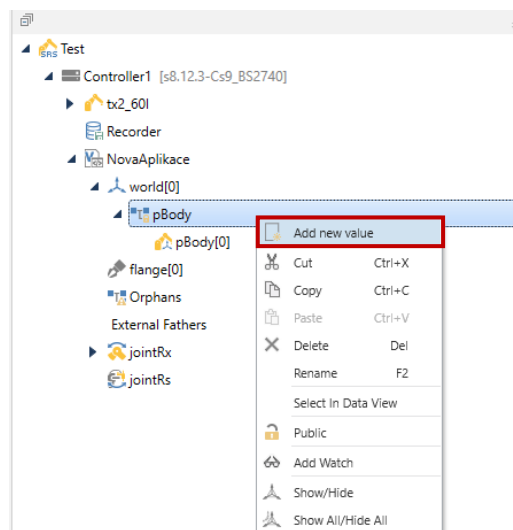
Obrázek 80 Naučení bodu jBezpeci

Pro vytvoření bodů na kvádru musíme vytvořit pole bodů v souřadném systému *world*. Opět přes pravé tlačítko na *world* a volíme *New data*. Zvolíme typ *pointRx* a opět před název dáme první písmeno typu bodu. V kolonce *Size(s)* volíme rozměr pole.



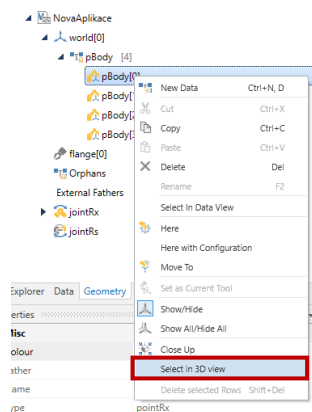
Obrázek 81 Vytvoření pole bodů

Pokud je potřeba přidat body do pole, tak přes pravé tlačítko zvolíme *Add new value*.



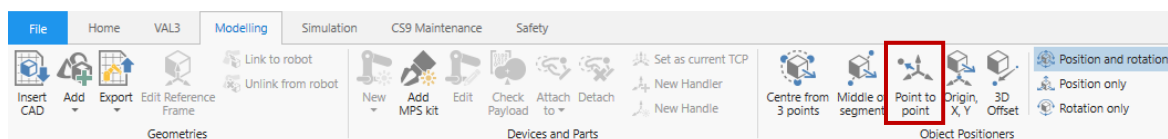
Obrázek 82 Přidání nové hodnoty do pole bodů

Pro zvolení bodu, za účelem manipulace ve 3D vizualizaci, musíme přes pravé tlačítko v možnostech zvolit *Select in 3D View*.

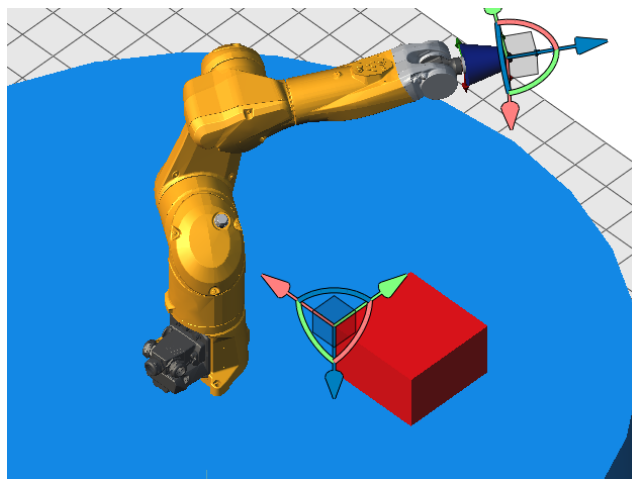


Obrázek 83 Zvolení bodu ve 3D vizualizaci

V kartě **Modelling** v odstavci **Object positioners** zvolíme **Point to point** a zvolíme jeden z rohů kvádrů. Bod ještě musíme následně zarotovat tak, aby osa Z směřovala směrem dolů, nebo-li do protisměru Z osy světového souřadného systému. To provedeme přes **Edit Position**. Správně naorientovaný bod vypadá následovně:

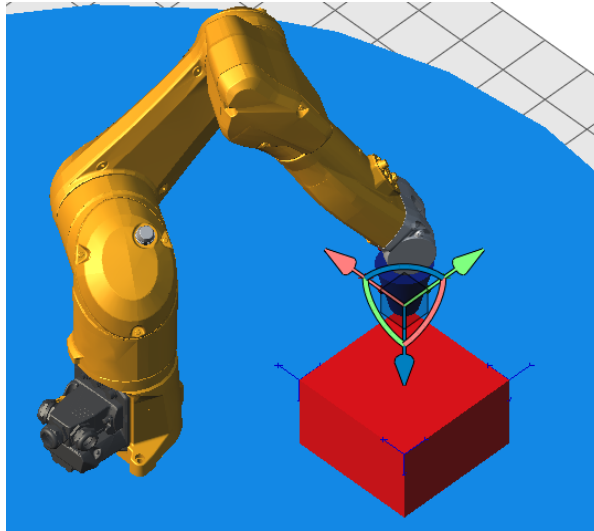


Obrázek 84 Výběr možnosti přesunutí bodu



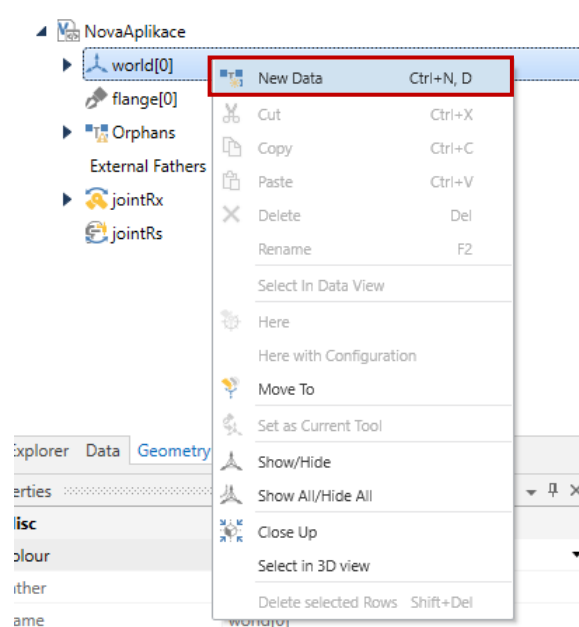
Obrázek 85 Správně naorientovaný bod na rohu kvádrů

Nyní celý proces zopakujeme pro zbylé 3 body. Pro ověření dostupnosti bodu koncovým efektořem robota můžeme koncový efektoř robota přemístit na daný bod. Přes pravé tlačítko zvolíme z možností **Move to** a pokud nenastane žádný problém, robot se do tohoto bodu dostane.



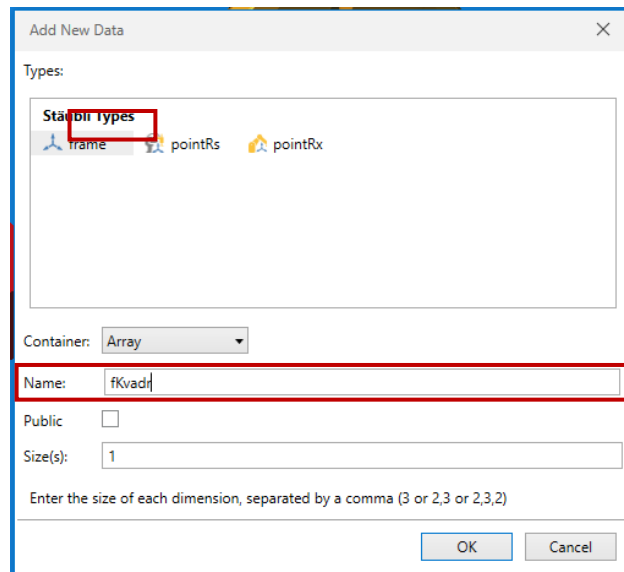
Obrázek 86 Ověření dostupnosti bodu koncovým efekto-
vým efekto-rem

Vytvoření vlastního souřadného systému má výhody obzvláště tehdy, kdy se pohne s paletou nebo je potřeba ji jinak položit a místo přeučování více bodů stačí přeučit pouze počátek a orientaci souřadného systému. Vytvoříme jej přes kliknutí pravým tlačítkem na *world* a zvolíme možnost *New Data*.



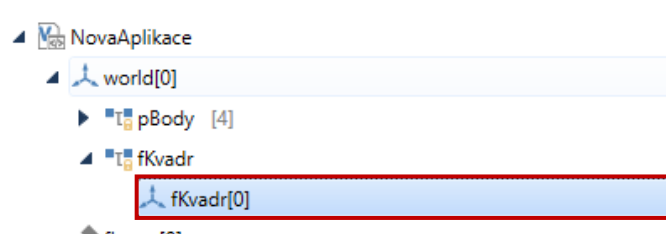
Obrázek 87 Vytvoření nového framu – při-
dání nových dat

Vyskočí tabulka, kde volíme typ *frame* a v kolonce *Name* svůj frame pojmenujeme.



Obrázek 88 Tabulka vytvoření nového framu

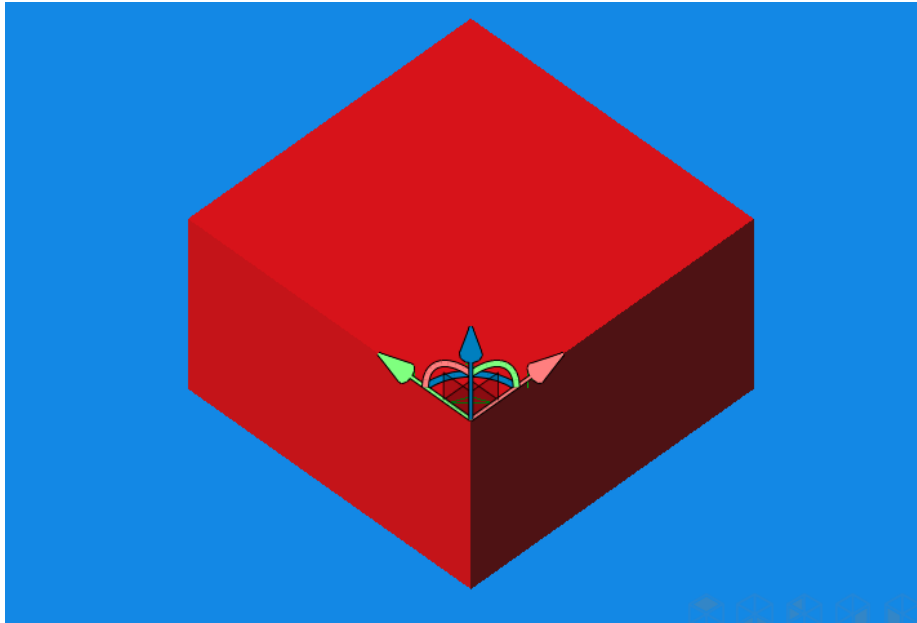
Nyní máme vytvořená data datového typu *frame* pojmenovaným jako *fKvadr*. Tento frame přes pravé tlačítko vybereme ve 3D vizualizaci pomocí *Select in 3D view* a umístíme jej na druhý kvádr.



Obrázek 89 Vytvořený frame ve stromu dat

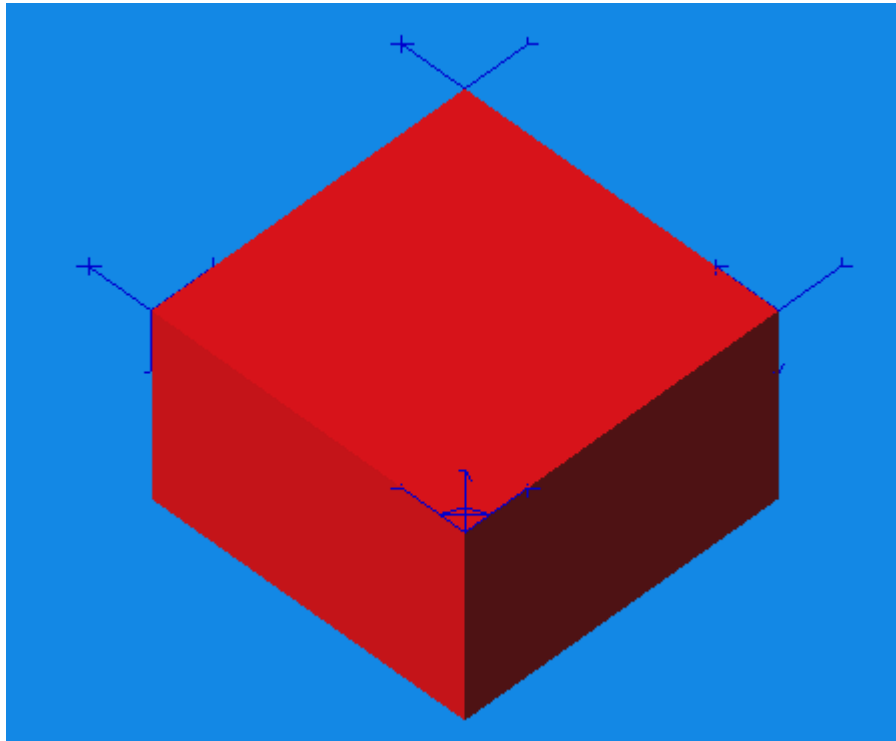
Nyní umístíme počáteční bod souřadného systému a jeho orientaci. V kartě *Modelling* v odstavci *Object Positioners* zvolíme *Origin X,Y* a je potřeba navolit 3 body:

- 1. bod – počáteční bod souřadnic
- 2. bod – bod na ose X
- 3. bod – bod na ose Y



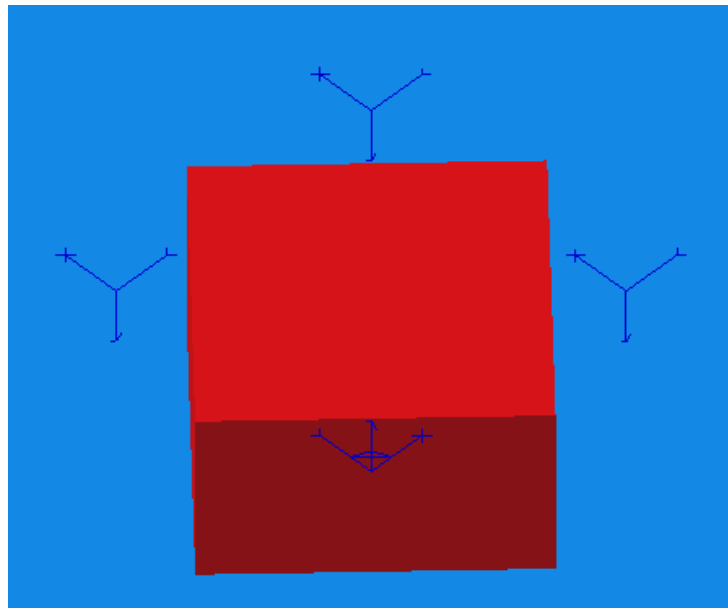
Obrázek 90 Nadefinovaný souřadný systém fKvadr

Nyní vytvoříme 3 body ve vytvořeném *fKvadr* a umístíme je do rohů kvádr. Tyto 3 body vytvoříme stejně jako tomu bylo výše. Poté s celým kvádrem pootočíme a tím demonstrováme funkčnost framu.

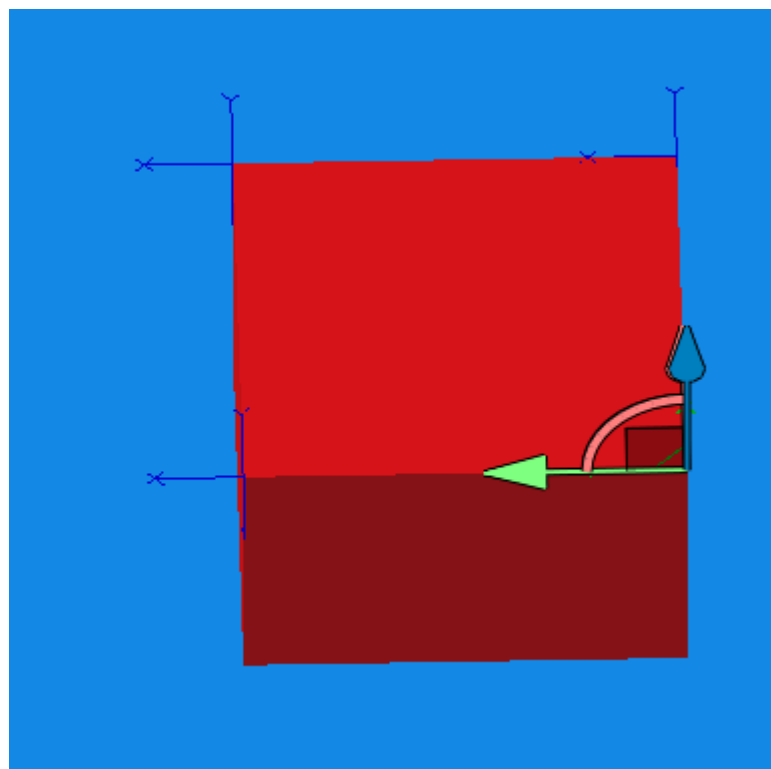


Obrázek 91 Frame s vlastními body

Nyní tento kvádr otočíme a přeúčíme souřadný systém.



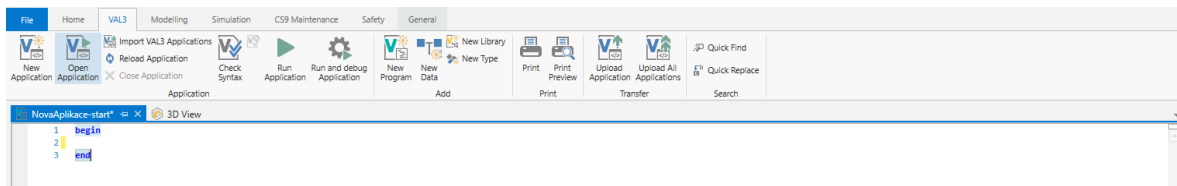
Obrázek 92 Otočený kvádr



Obrázek 93 Přenaucený frame na otočeném rámu

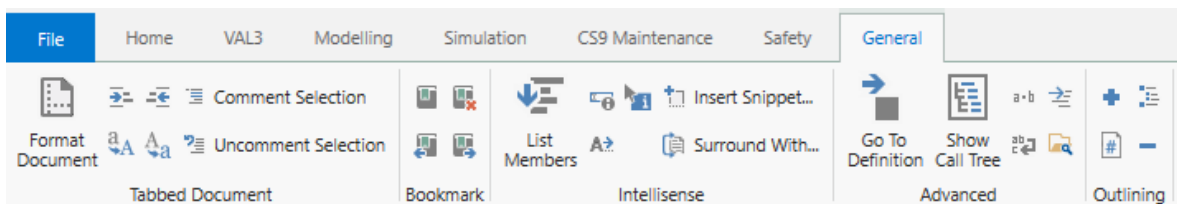
9.7 Vytvoření programu a spuštění simulace

V *Cell Explorer* otevřeme *start()*. Vyskočí nám okno, kde se programuje pohyb robotické ruky pomocí jazyka VAL3. Program začíná *begin* a končí *end*.



Obrázek 94 Programovací okno

V kartě **General** najdeme spoustu užitečných funkcí pro upravování kódu, jako je formátování dokumentu, komentování a spousta dalších.



Obrázek 95 Možnosti úpravy programovacího okna

Máme 3 základní pohyby:

- **movec** (koncový bod, bod, kterým má projít, nástroj, rychlost) – pohyb po kružnici
- **moveI** (bod, nástroj, rychlost) – lineární pohyb za pomoci kartézských souřadnic
- **movej** (bod, nástroj, rychlost) – jedná se o rotační pohyb, kdy se rameno nepohybuje pomocí kartézských souřadnic, ale pomocí úhlů natočení

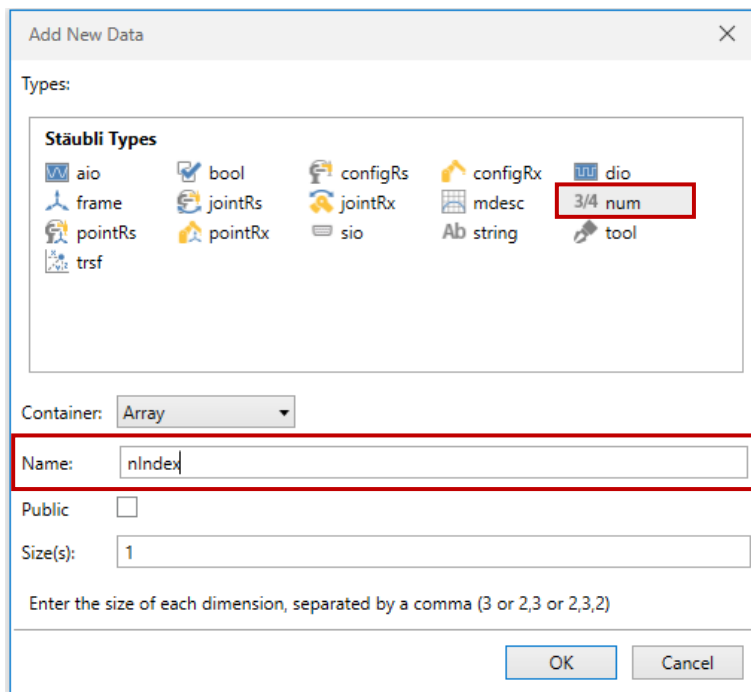
Pro práci s poli bodů slouží cykly:

- Cyklus **For**
for proměnná = num **to** num **step** num
 <kód>
endFor
- Cyklus **While**
while (podmínka)
 <kód>
endWhile
- Cyklus **Do until**
do
 <kód>
 Until (podmínka)

Dále se používá spousta dalších příkazů :

- **appro**(bod, transformace - {x,y,z,Rx,Ry,Rz})
- **close**(nástroj) – uzavře/sevře nástroj
- **open**(nástroj) – otevře nástroj
- **delay**(čas v sekundách) – funkce pro čekání v sekundách
- **waitEndMove()** – funkce, díky které se počká na dokončení pohybu, aby se nepřešlo hned k další instrukci

Vytvoříme nyní jednoduchý program, který přesune robota do bezpečného bodu *jBezpeci*, následně projede všechny 4 body na prvním kvádru a opětovně se vrátí do *jBezpeci*. Pro vytvoření proměnné na použití cyklu musíme v kartě **Data** pravým kliknout na aplikaci, zvolit **Add a New Data**. Zvolíme datový typ *num* a patřičně jej pojmenujeme.



Obrázek 96 Vytvoření nové proměnné

Nyní jsme schopni vytvořit program pro pohyb robota viz následující kód:

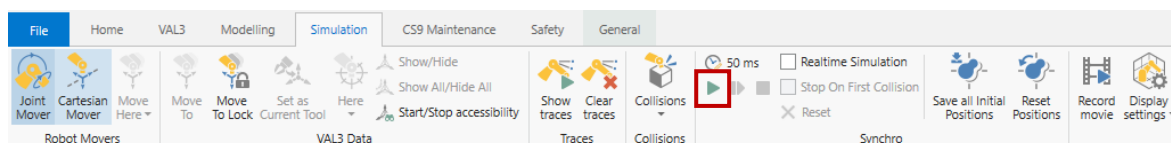
```

1  begin
2  movej(jBezpeci,tNastroj,mNomSpeed)
3  for nIndex = 0 to 3
4  move1(pBody[nIndex],tNastroj,mNomSpeed)
5  waitEndMove()
6  delay(0.5)
7  endFor
8  move1(pBody[0],tNastroj,mNomSpeed)
9  movej(jBezpeci,tNastroj,mNomSpeed)
10 waitEndMove()
11 end

```

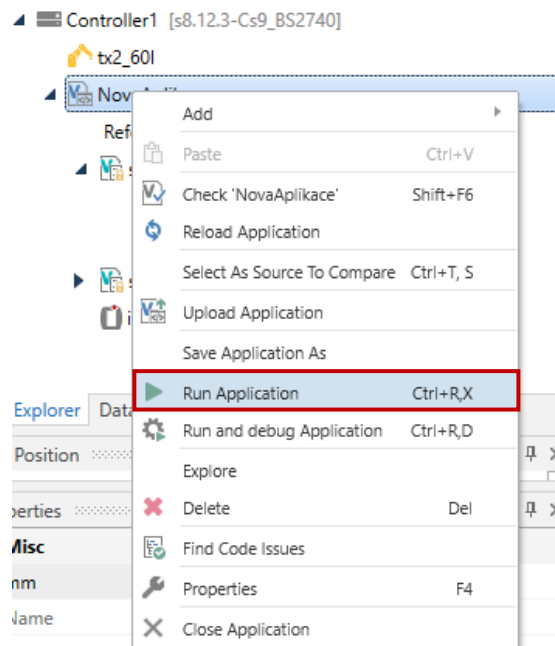
Obrázek 97 Naprogramovaný pohyb robota

Pro spuštění simulace aplikace je nejprve potřeba zapnout synchronizaci přes zelené tlačítko **Start synchro**.



Obrázek 98 Zapnutí synchronizace prostředí a programu

Následně spustíme aplikaci pomocí pravého tlačítka na aplikaci, kde z listu zvolíme **Run Application**. Aplikace se dá spustit i z karty VAL3.



Obrázek 99 Spuštění aplikace

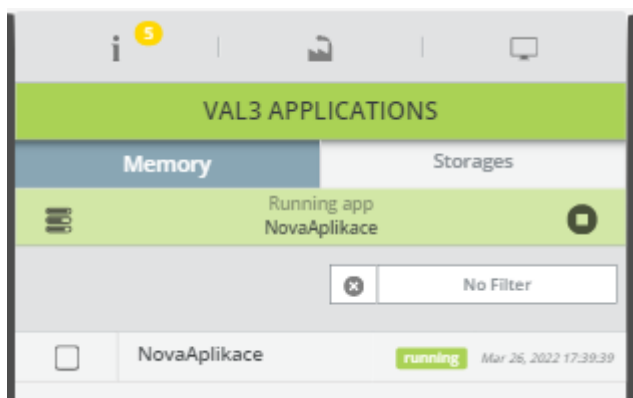
Po spuštění aplikace se otevře emulátor TeachPendantu. Ten obsahuje:

1. **JOG** – nabízí možnost manuálně naučit body nebo hýbat s robotem
2. **VAL3** – zde najdeme nahrané programy a odsud je taky můžeme spouštět/zavírat
3. **LOGGER** – najdeme zde chybové nebo informační hlášení
4. **IO** – nastavení komunikace, kontrola signálů
5. **ROBOT** – nastavení kalibrace, ovládání brzd, nastavení limitů a informace o robotu
6. **SETTINGS** – nastavení jazyka, sítě, času atd.
7. **OVLÁDACÍ TLAČÍTKA**
 - a) Zapnutí/vypnutí napájení motorů
 - b) Domovské tlačítko
 - c) Zpětné tlačítko
 - d) Zastavení/spuštění aplikace
 - e) Zvýšení/snížení rychlosti pohybu
 - f) Ovládací tlačítka pro manuální ovládání robota
 - g) Přepínání módů



Obrázek 100 Emulátor TeachPendantu v SRS

Přechodem na *VAL3* obrazovku si můžeme překontrolovat, kterou aplikaci jsme spustili a popřípadě ji můžeme vypnout a pustit jinou.

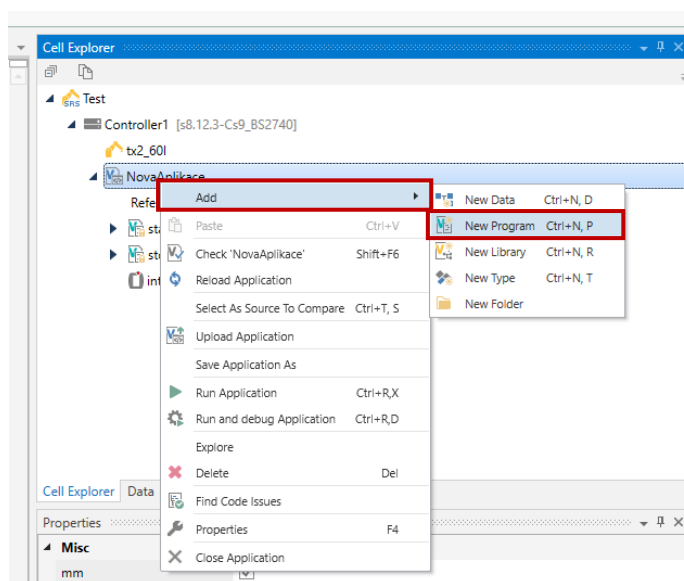


Obrázek 101 VAL3 obrazovka

Nyní stačí pouze zapnout motory. To poznáme tak, že LED pod tlačítkem začne modře blikat. Pro spuštění pohybu robotu stačí držet tlačítko pro zastavení/spuštění aplikace. To se v manuálním módu musí držet, ale v automatu ho stačí jen zmáčknout.

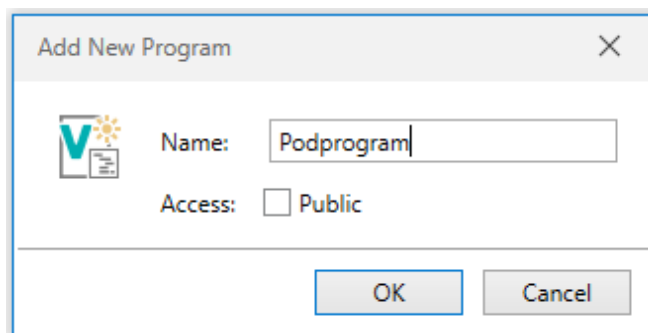
9.8 Použití podprogramů

Podprogramy se používají pro rozdělení programu do dílčích programů, které se potom zavolají v hlavním programu *start()*. Podprogram vytvoříme pomocí pravého tlačítka na zvolenou aplikaci a vybíráme *Add* a potom *New Program*.



Obrázek 102 Založení podprogramu

V následujícím okně podprogram pojmenujeme.



Obrázek 103 Pojmenování podprogramu

Nyní do podprogramu můžeme například nakopírovat cyklus *For*, který jsme výše vytvořili. Volání podprogramu je primitivní: pouze se napíše funkce call a jeho název. Na následujících 2 obrázcích bude toto volání znázorněno.

```

1 begin
2   for nIndex = 0 to 3
3     move1(pBody[nIndex],tNastroj,mNomSpeed)
4     waitEndMove()
5     delay(0.5)
6   endFor
7 end

```

Obrázek 104 Struktura kódu podprogramu

```

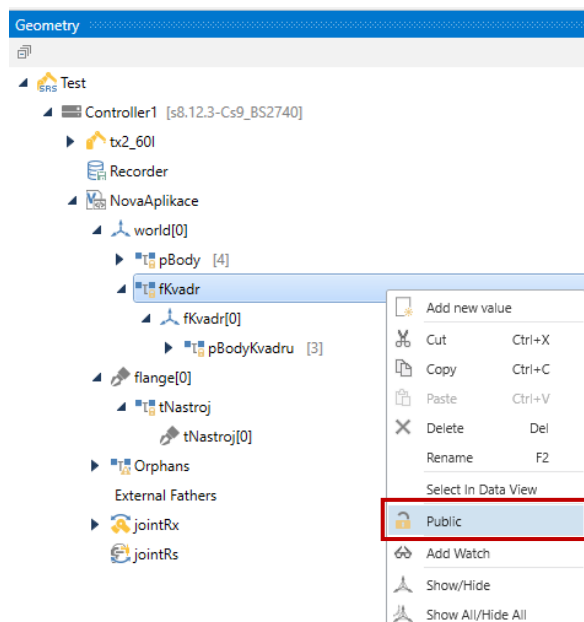
1 begin
2   movej(jBezpeci,tNastroj,mNomSpeed)
3   call Podprogram()
4   move1(pBody[0],tNastroj,mNomSpeed)
5   movej(jBezpeci,tNastroj,mNomSpeed)
6   waitEndMove()
7 end

```

Obrázek 105 Struktura kódu startu

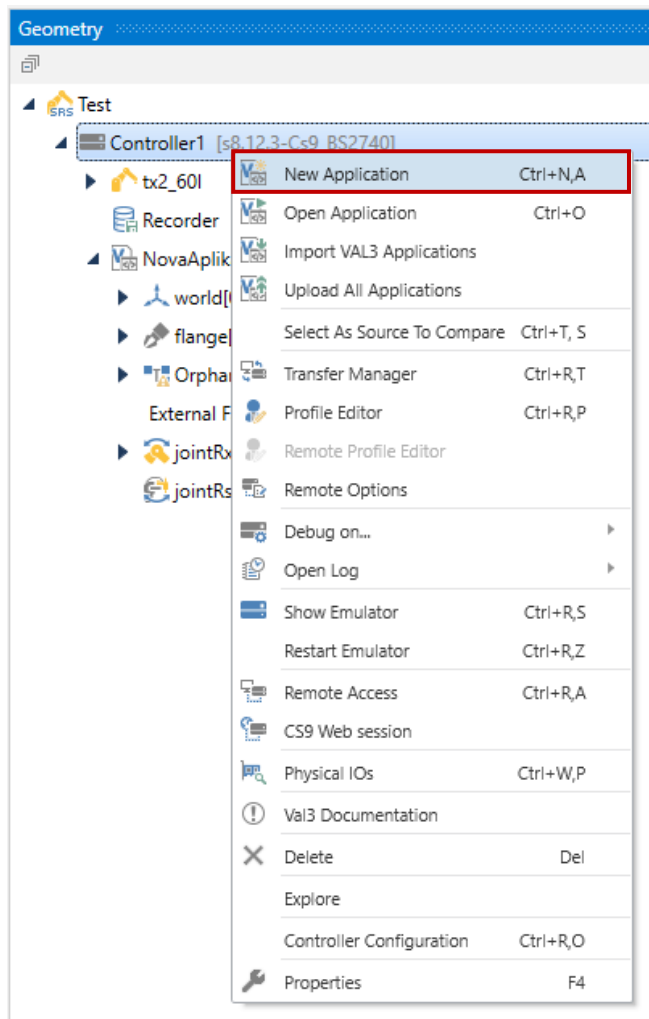
9.9 Použití knihoven

Knihovna je načtena jako aplikace, načteme-li knihovnu, tak máme přístup k veřejným bodům, proměnným a úlohám aplikace, která je považovaná za knihovnu. Nejprve začneme tím, že v aplikaci *NovaAplikace* v kartě *Geometry* pravým klikneme *fKvadr* a zvolíme *Public*. To samé provedeme s *tNastroj* a *pBodyKvadr*.

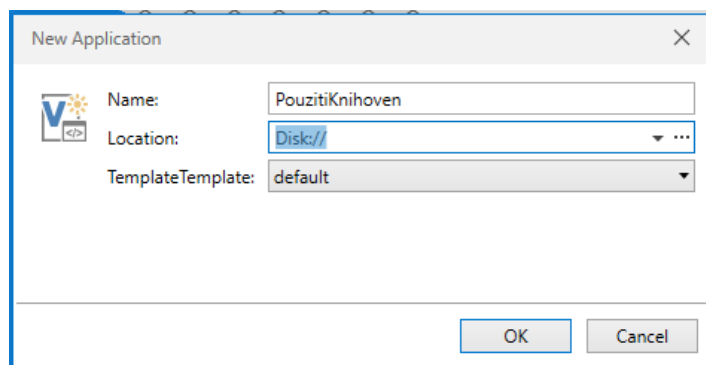


Obrázek 106 Převedení dat na Public

Nyní vytvoříme novou aplikaci, kterou pojmenujeme jako *PouzitiKnihoven*. To provedeme pravým kliknutím na *Controller1* a volíme *New Application* a v následujícím vyskakujícím okně ji pojmenujeme.

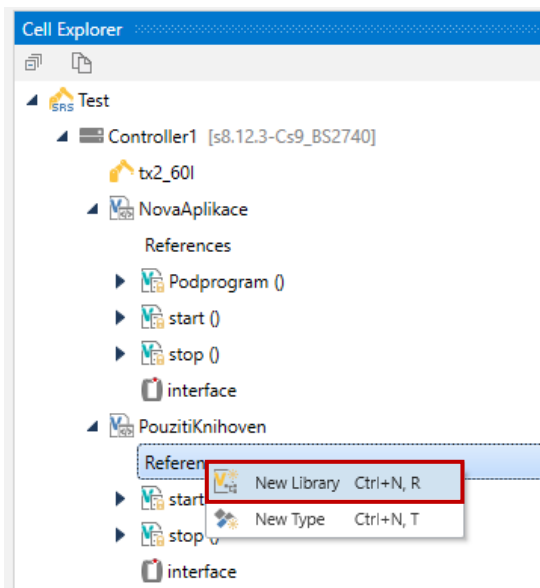


Obrázek 107 Vytvoření nové aplikace

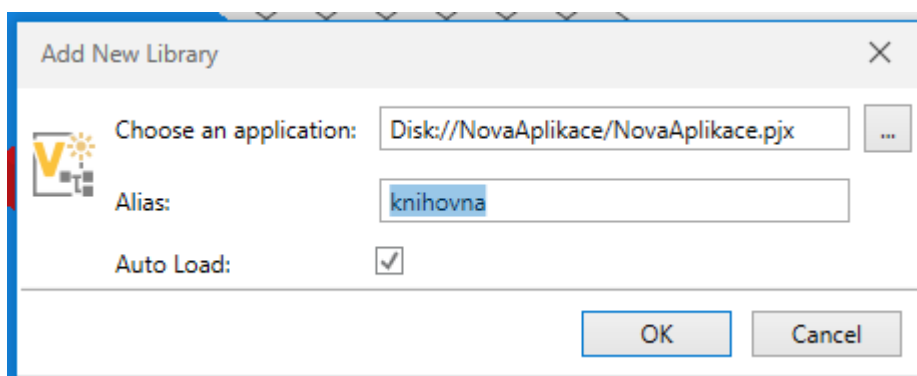


Obrázek 108 Pojmenování nové aplikace

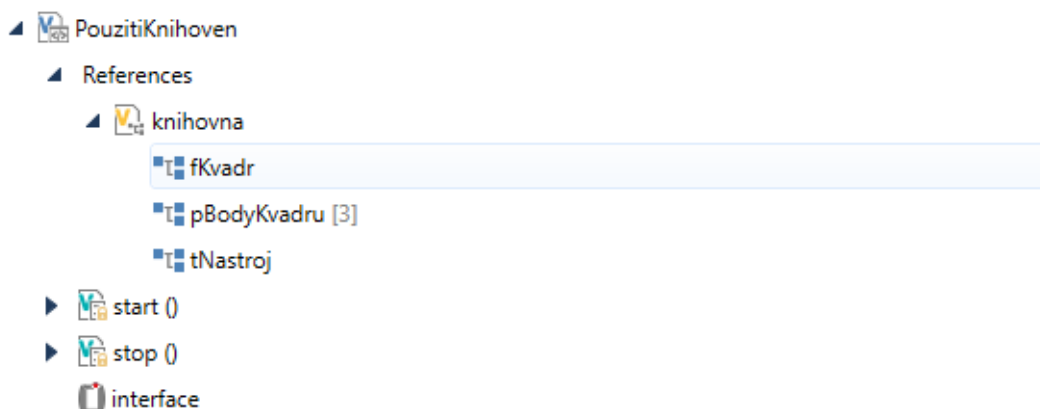
Nyní v kartě *Cell Explorer* klikneme pravým na *References* aplikace *PouzitiKnihoven* a volíme *New Library*. Do aplikace zvolíme cestu k souboru *NovaAplikace.pjx* a knihovnu pojmenujeme třeba jako *knihovna*.



Obrázek 109 Vytvoření nové knihovny



Obrázek 110 Pojmenování a vybrání nové knihovny

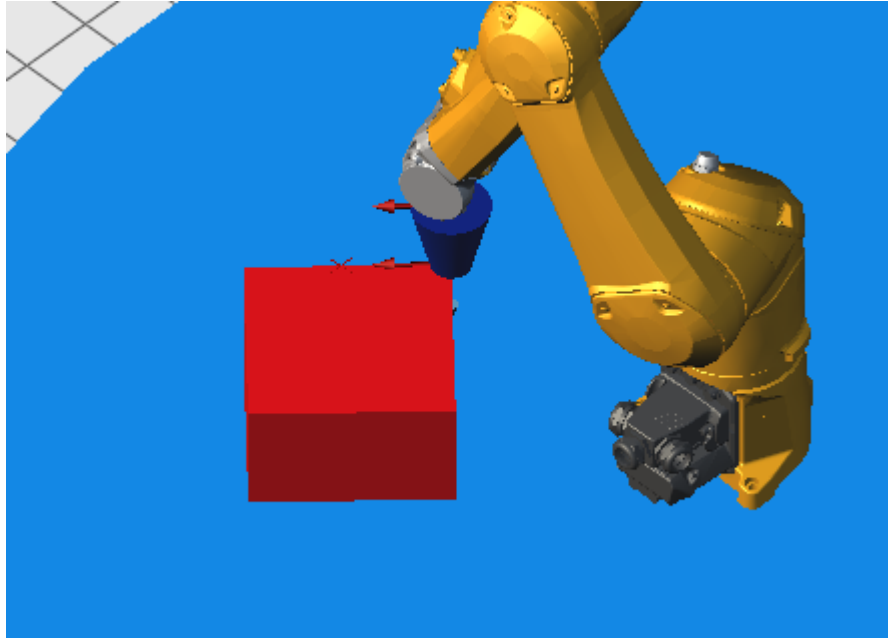


Obrázek 111 Obsah knihovny

Nyní si vytvoříme ve *start()* triviální program, který přesune robota na bod z knihovny

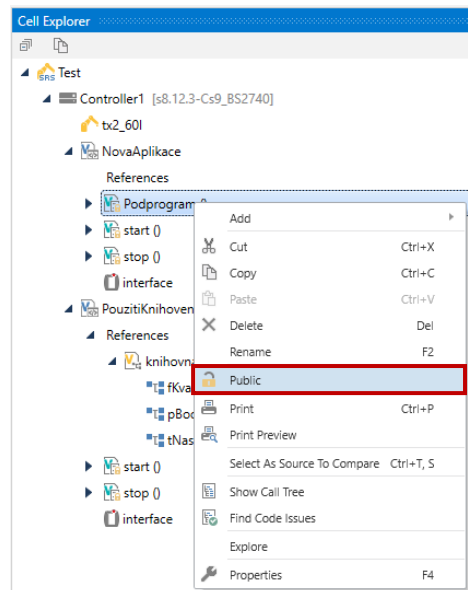
```
1 begin
2   movej(knihovna:pBodyKvadr[u],knihovna:tNastroj,mNomSpeed)
3   waitEndMove()
4 end
```

Obrázek 112 Presun robotu na bod z knihovny



Obrázek 113 Výsledek volání pohybu na bod z knihovny

Z knihovny můžeme volat i programy jiných aplikací, tudíž si *Podprogram()* z *NovaAplikace* převedeme na *Public*.



Obrázek 114 Převedení Podprogram na public

```
1 begin
2   movej(knihovna:pBodyKvadru[0], knihovna:tNastroj, mNomSpeed)
3   call knihovna:Podprogram()
4   waitEndMove()
5 end
```

Obrázek 115 Volání podprogramu z knihovny

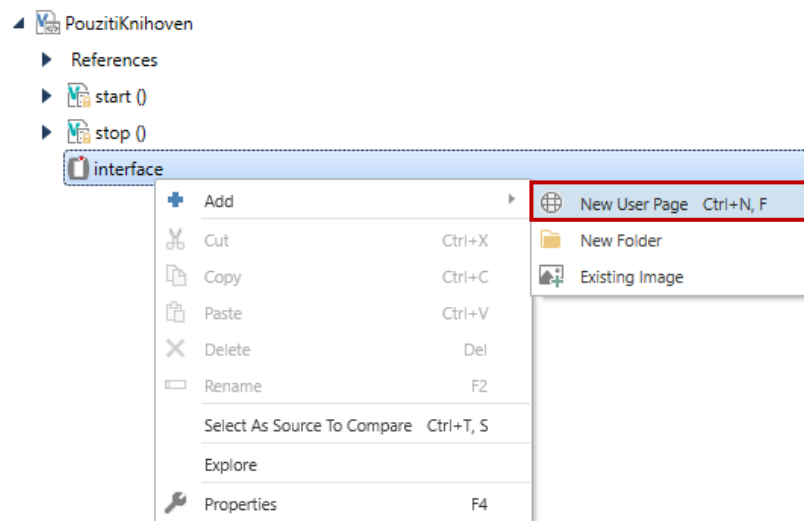
Dále se dají knihovny upravovat přes *libSave* a *libLoad*. Je potřeba mít vytvořenou numerickou proměnnou a vědět identifikátor.

nNum = knihovna:libLoad(“Cesta knihovny” nebo název knihovny)

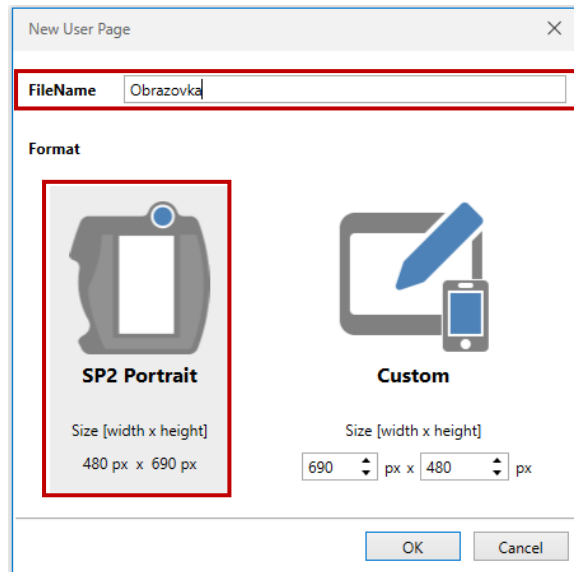
nNum = knihovna:libSave(“Cesta knihovny” nebo název knihovny”)

9.10 Vytvoření uživatelského rozhraní

Staubli Robotics Suite nabízí možnost vytvoření uživatelského rozhraní, nebo-li obrazovky pro pendant. Tu vytvoříme přes pravé tlačítko na *interface* a *New User Page*. Ve vyskakovajícím okně obrazovku pojmenujeme a vybereme velikost

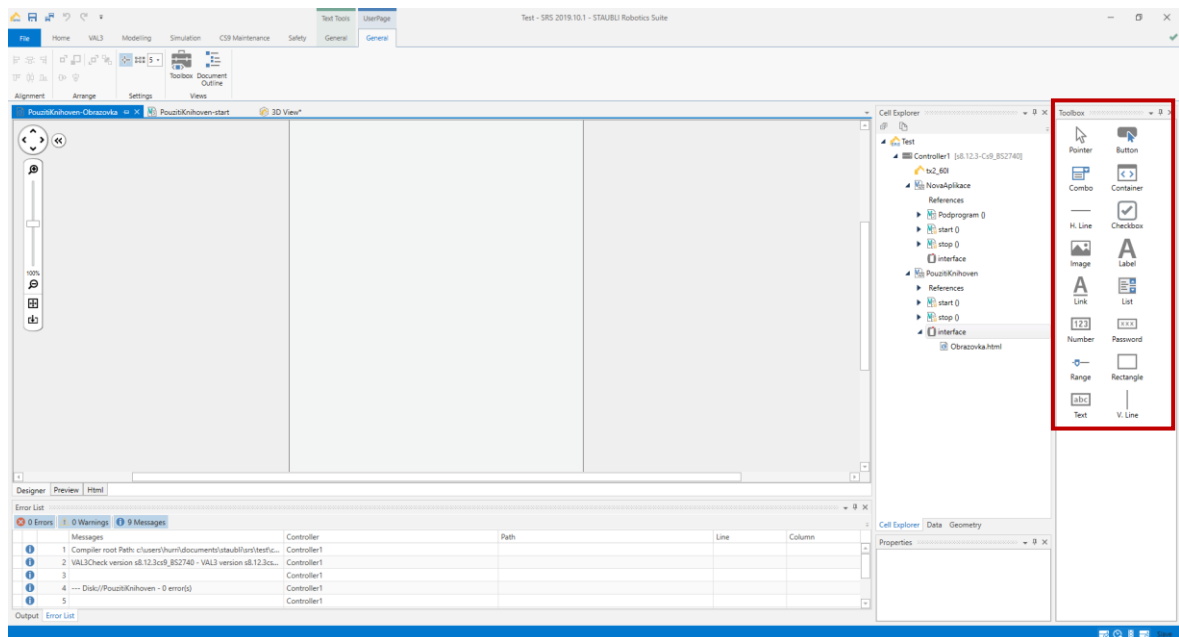


Obrázek 116 Vytvoření uživatelského rozhraní



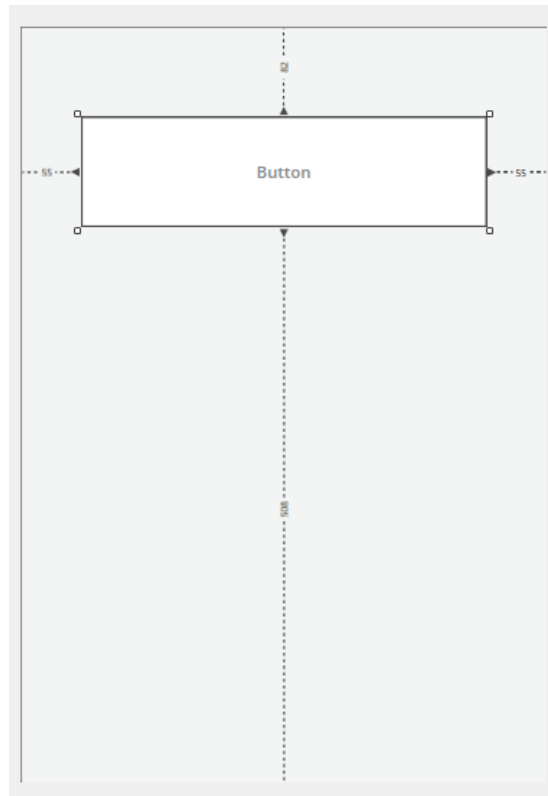
Obrázek 117 Definice obrazovky

Otevře se nám okno editace obrazovky a **Toolbox**, kde nalezneme dostupné prvky, které lze na obrazovku umístit



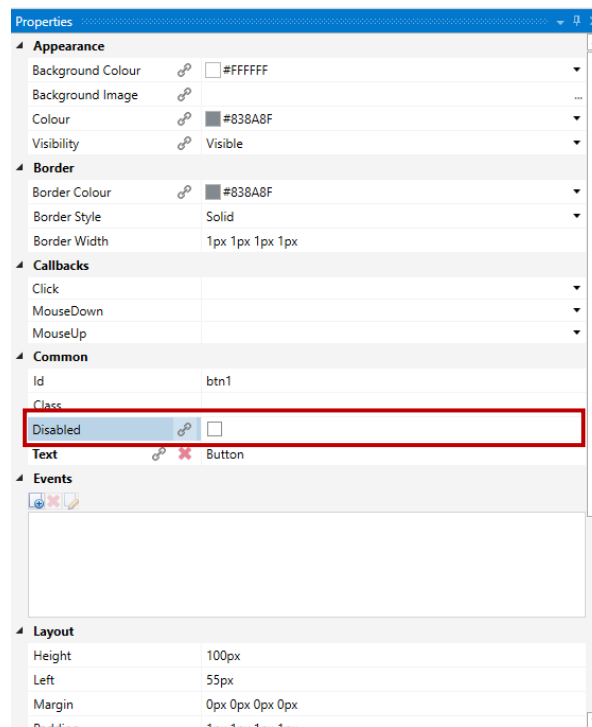
Obrázek 118 Toolbox a obrazovka editace UI

Vložíme-li tlačítko **Button**, tak můžeme vidět našeptávače vzdáleností od krajů obrazovky.



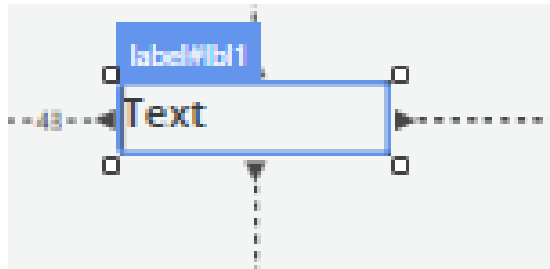
Obrázek 119 Vložený Button

Každý prvek má své vlastnosti, kde se dá měnit barva, ohraničení a funkce. **Id** je jeho jméno, díky kterému se bude používat v programu. U **Disabled** se používá připojená funkce, která když je **true**, tak se na tlačítko nedá kliknout. Tato funkce se k tlačítku připojí přes **link**.

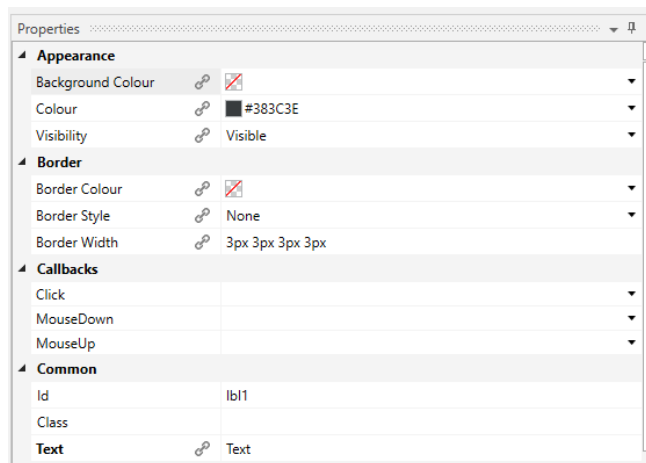


Obrázek 120 Vlastnosti tlačítka

Dále je zde prvek **Label**, který slouží pro pevný text. V možnostech můžeme měnit styl textu, velikost, ohraničení a barvy.

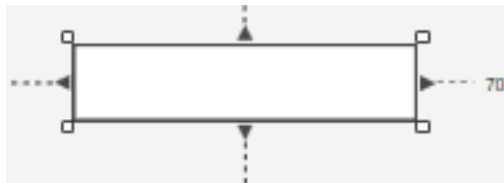


Obrázek 121 Prvek Label



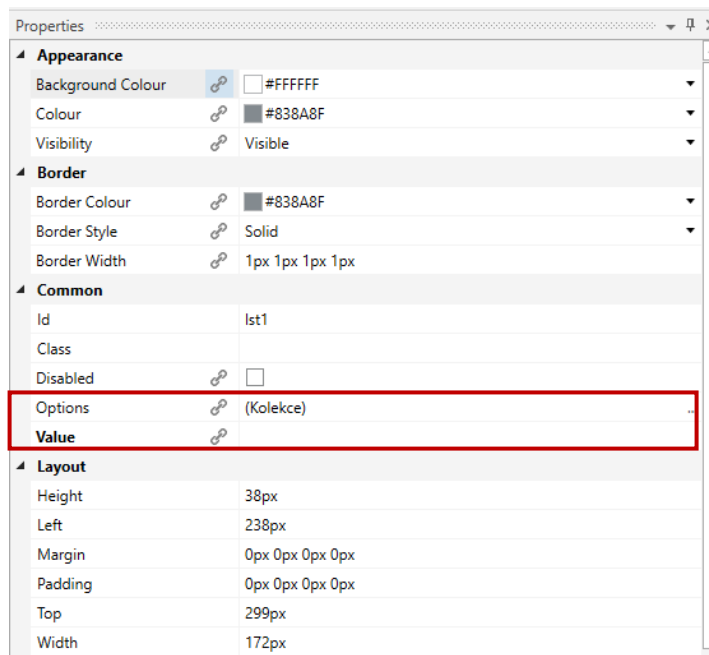
Obrázek 122 Vlastnosti prvku Label

Dále je zde prvek **List**, který slouží pro výběr z více možností.

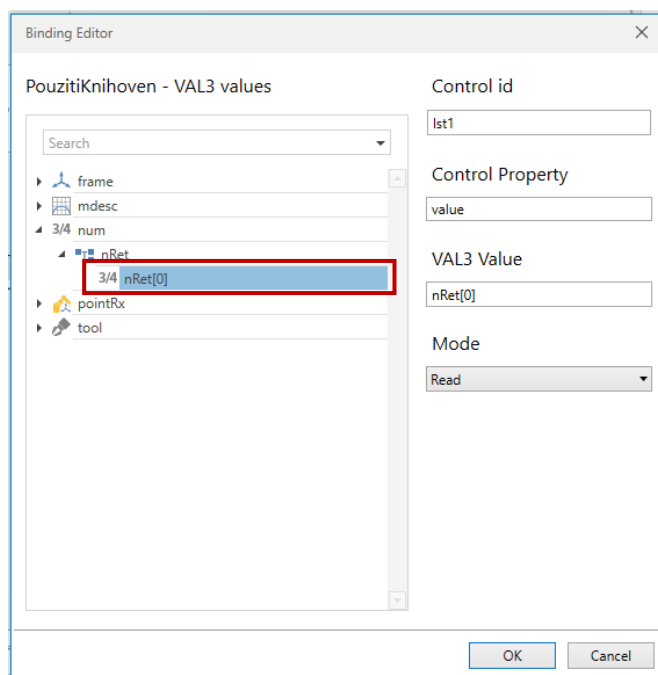


Obrázek 123 Prvek List

Do kolonky **Value** přes **link** připojíme proměnnou. Možnosti, které je možné vybrat, se přidávají v kolonce **Options**.

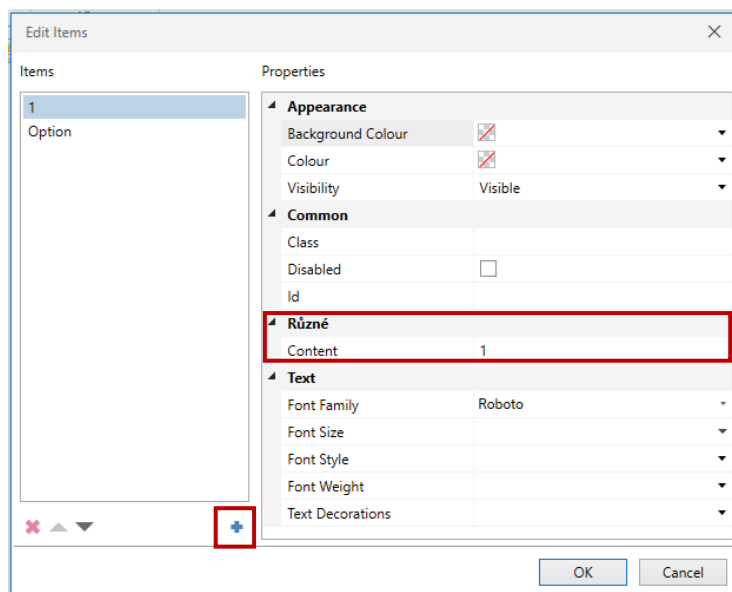


Obrázek 124 Vlastnosti prvku List



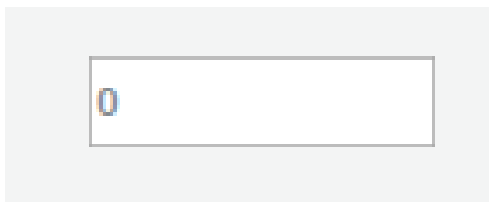
Obrázek 125 Linknutí listu k proměnné

V přidávání možností můžeme přidat možnost pomocí modrého tlačítka. Jejich hodnotu poté měníme v *Různé* a *Content*.



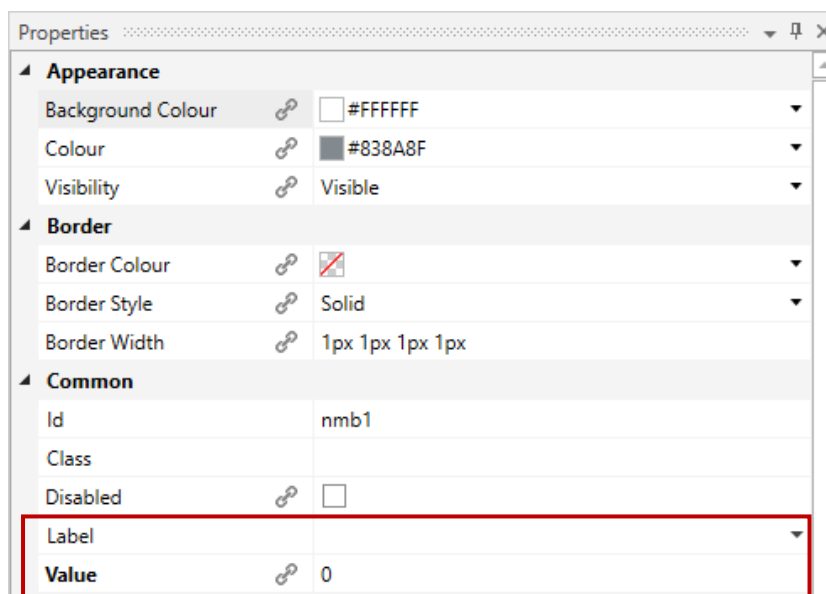
Obrázek 126 Přidání nových možností výběru

Dále je zde prvek **Text**, který zobrazuje jak statický, tak i proměnlivý text.



Obrázek 127 Prvek Text

Vlastnost **Label** slouží pro výchozí text, který se zobrazí, když není připojená k prvku proměnná, která se připojuje k **Value**.



Obrázek 128 Vlastnosti prvku Text

Pro zavolání uživatelského rozhraní musíme mít vytvořenou numerickou proměnnou. Dále pouze použijeme příkaz *userPage*.

```
1 begin
2   nRet=userPage("Obrazovka")
3   movej(knihovna:pBodyKvadr[u],knihovna:tNastroj,mNomSpeed)
4   call knihovna:Podprogram()
5   waitEndMove()
6 end
```

Obrázek 129 Volání UI



Obrázek 130 Vzhled UI

10 ZPRACOVÁNÍ ÚLOH PRO STUDENTY

10.1 Úloha 1

10.1.1 Zadání

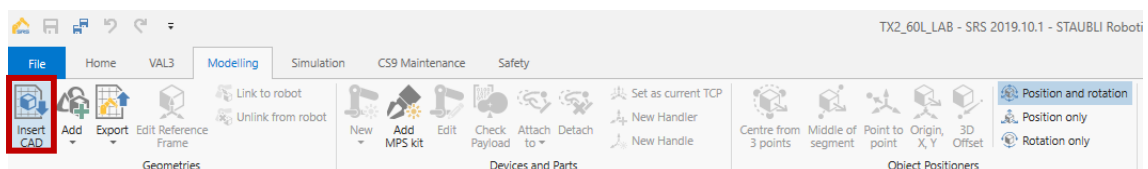
Vytvořte za pomoci 5 válečků sloup, který bude mít základnu v prvním sloupci na prvním řádku palety. Nejprve proved'te simulaci zadané úlohy a následně úlohu po konzultaci s vyučujícím převed'te na reálného robota a ověřte funkčnost programu. Při programování vytvořte 4 verze programu:

- Program neobsahuje cykly ani podprogramy, je potřeba vytvořit více bodů pro uchopení, pro přejezd, tak i pro umístění.
- Použitím cyklů a pole bodů minimalizujte počet použitých bodů
- Využijte funkci *apro*, díky které se potřebný počet bodů ještě více minimalizuje
- Přepracujte jednu z úloh tak, aby byla použitelná na reálného robota.

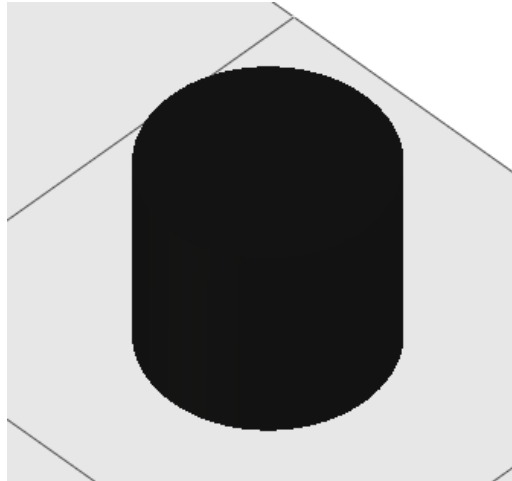
Jednotlivé verze zpracujte do podprogramů, které následně budete volat do *start()* programu. Při simulaci si vytvořte vlastní gripper, ale při přechodu na reálného robota vložte již vytvořený gripper, který odpovídá tomu na reálném robotu. Simulace prostředí je pouze orientační z hlediska vzdáleností od robota. Pro reálnou aplikaci je doporučeno kombinovat verzi B a verzi C.

10.1.2 Vložení a přemístění válečků

V kartě *Modelling* klikneme na *Insert CAD* a v následujícím okně vybereme soubor *Valecek_45x40_mm*.

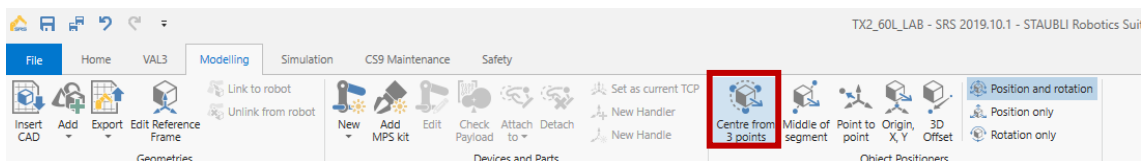


Obrázek 131 Vložení CAD objektu - váleček



Obrázek 132 Vložený váleček

Ten díky odstavci *Object Positioners* umístíme do drážky pomocí *Centre from 3 points*. Váleček následně posuneme v ose Z o 20 mm. To provedeme pravým kliknutím na váleček a vybereme funkci *Edit position*.



Obrázek 133 Výběr metody přemístění válečku



Obrázek 134 Umístění válečku do drážky

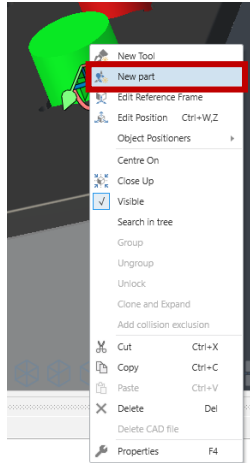
| Relative Position | | | | | |
|-------------------|--------|----|---------|----|---------|
| X | 0,00 | Y | 0,00 | Z | -20,00 |
| Rx | 0,00 | Ry | 0,00 | Rz | 0,00 |
| Absolute Position | | | | | |
| X | 180,00 | Y | -419,88 | Z | 1055,00 |
| Rx | 180,00 | Ry | 0,00 | Rz | -90,00 |

Obrázek 135 Tabulka Edit Position

Postup opakujeme pro další 4 válečky, dokud nemáme 5 válečků umístěných na paletce

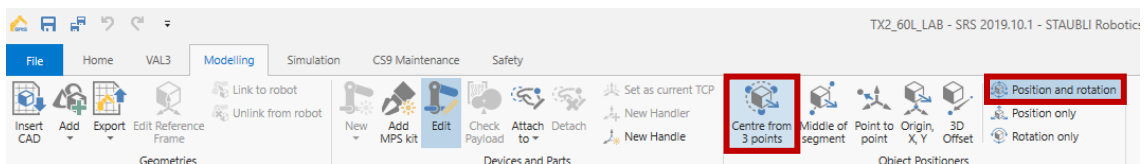
10.1.3 Převedení válečků součásti

Abychom umožnili pohyb s těmito válečky, musíme je změnit na typ **Part**. Pravým kliknutím na váleček a zvolením **New part** převedeme váleček na součást.

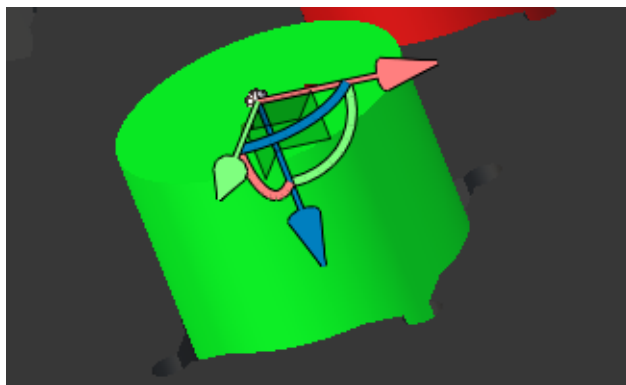


Obrázek 136 Převod válečku na součástku

Každá součástka má svůj **handle**, který slouží jako úchopový bod. Ten pomocí **Object Positioners** a **Centre from 3 points** přemístíme na vrchol válečku.



Obrázek 137 Přemístění handlu

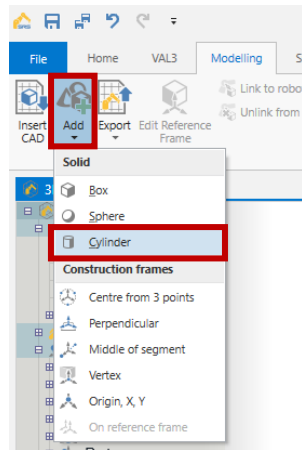


Obrázek 138 Správné umístění Handlu

Postup opakujeme pro zbývajících 4 válečky. Je zde ovšem také možnost tuto součást nakopírovat. To znamená, že se váleček nakopíruje i s jeho nastaveným **handlem**.

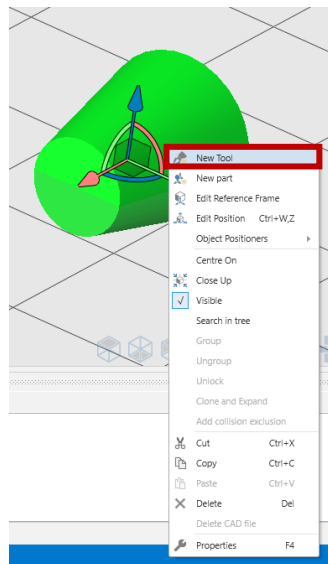
10.1.4 Vytvoření nástroje

Nyní si vytvoříme nástroj, se kterým budeme v simulaci pracovat. V kartě **Modelling** přes **Add** vložíme nový **Cylinder**.



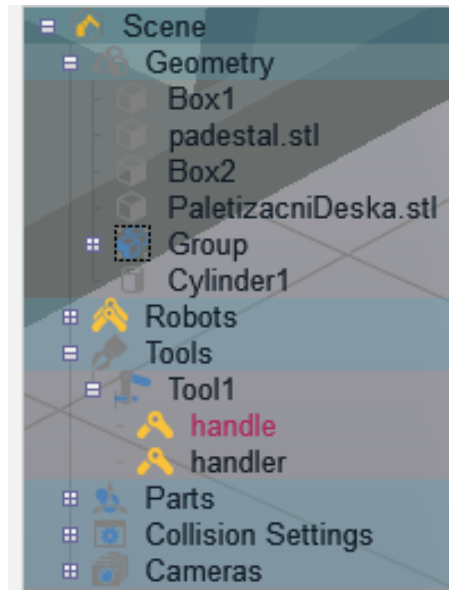
Obrázek 139 Vložení nového objektu

Tento vložený **Cylinder** si označíme a pozměníme jeho vlastnosti. To budou zejména jeho průměry, výška a barva. **Top Radius** změníme na 30, **Bottom Radius** na 50 a **Height** na 100. Barvu si zvolíme dle libosti. Tento **Cylinder** poté změníme na nástroj. Označíme jej a přes pravé kliknutí vybereme **New tool**.



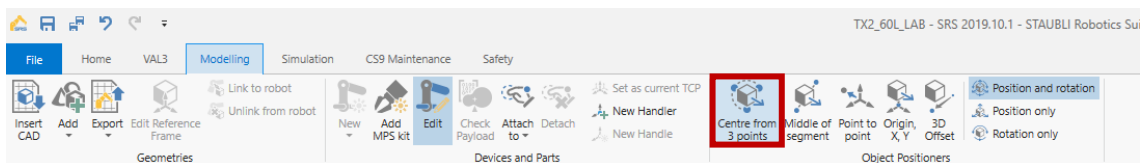
Obrázek 140 Převod Cylinderu na nástroj

Ve stromě scény si rozklikneme složku **Tools** a podsložku **Tool1** a jako první umístíme **handle** a to na spodní stěnu (**Bottom Radius**).

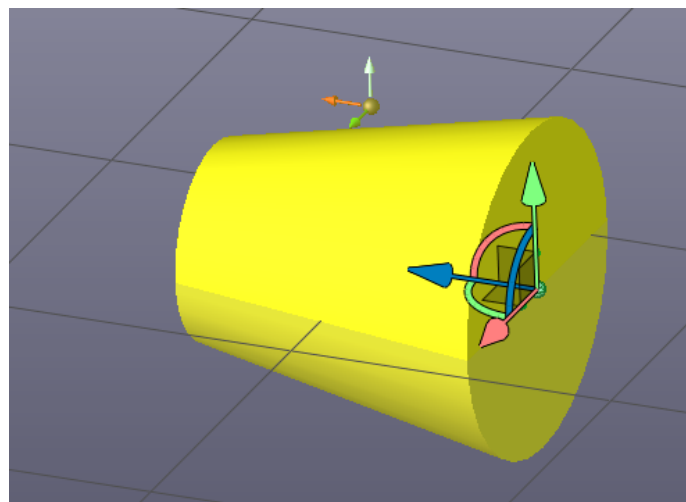


Obrázek 141 Strom scény a zvolení handlu

Pro umístění *handlu* použijeme *Object Positioners* a *Centre from 3 points*. Po umístění je důležité, aby osa Z směřovala do vnitřku nástroje.



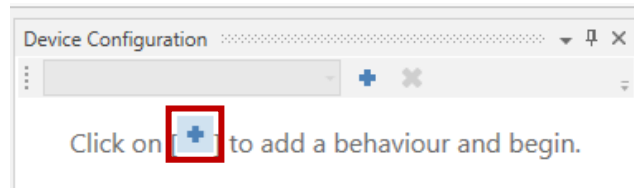
Obrázek 142 Zvolení možnosti pro umístění handlu



Obrázek 143 Správně umístěný handle

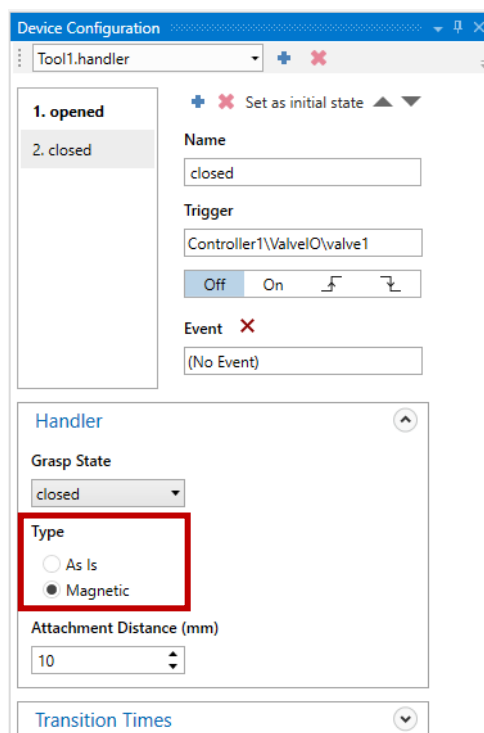
Stejný postup provedeme pro *handler*. Následně nastavíme chování nástroje přes *Device Configuration*.

V *Device Configuration* klikneme na modré plus a vyskočí okno, kde se nastavuje chování



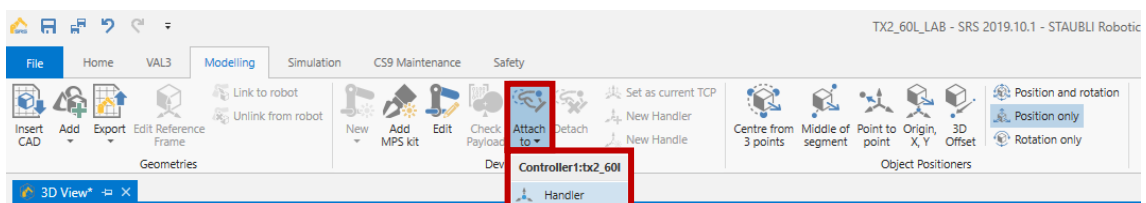
Obrázek 144 Nastavení nástroje
a přidání nového chování

Otevře se bližší nastavení, kde je potřeba v *Handler* změnit *Type* z *As is* na *Magnetic*.



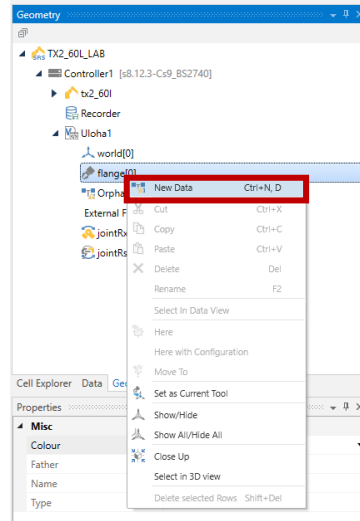
Obrázek 145 Bližší nastavení

Nyní můžeme nástroj připevnit k robotu. Provedeme to pomocí *Attach to* a zvolíme *Handler* robota.



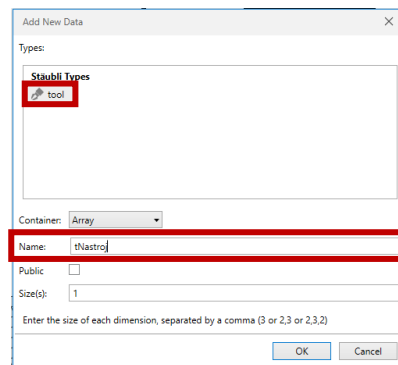
Obrázek 146 Připnutí nástroje k robotu

Pro práci s nástrojem je potřeba mu vytvořit data o jeho geometrii. Přepneme se tedy v panelu do karty *Geometry* a pravým kliknutím na *flange* a zvolíme *New Data*.



Obrázek 147 Vytvoření nové proměnné nástroje

Vyskočí tabulka, kde zvolíme datový typ **Tool** a v kolonce **Name**: jej pojmenujeme **tNastroj**.



Obrázek 148 Pojmenování proměnné pro nástroj

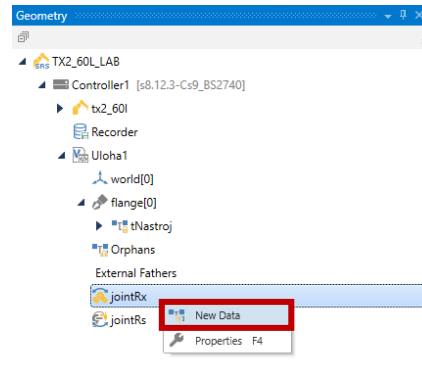
Po rozkliknutí dat vidíme, že je potřeba doplnit **Gripper**. Ještě je možnost nastavení času otevření/zavření pomocí **OTime/CTime**.

| Index | X | Y | Z | Rx | Ry | Rz | Gripper | OTime | CTime |
|-------|---|---|-----|----|----|----|---------|-------|-------|
| 0 | 0 | 0 | 100 | 0 | 0 | 0 | | 0 | 0 |

Obrázek 149 Data nástroje

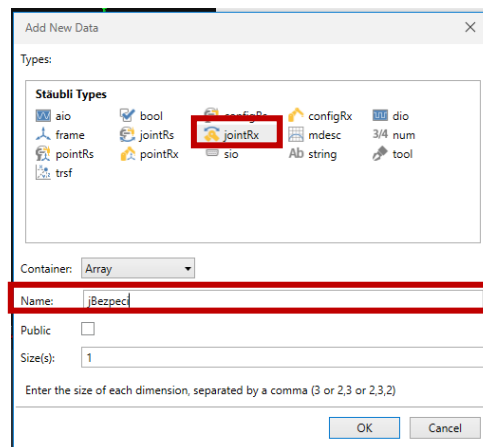
10.1.5 Vytváření bodů

Nyní můžeme začít vytvářet body. Začneme s bodem, který budeme považovat za bezpečný. Prvně si vytvoříme jeho data. To provedeme pravým kliknutím na **jointRx** v kartě **Geometry** a vybereme **New Data**.



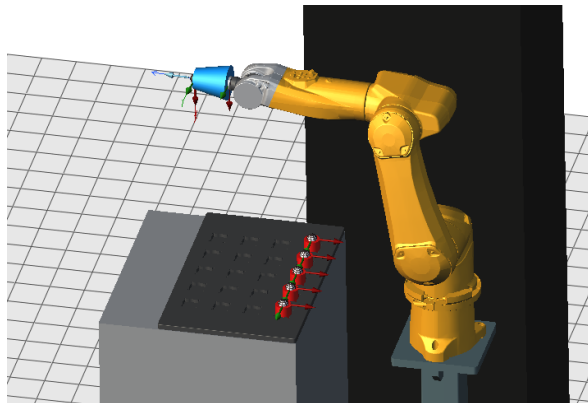
Obrázek 150 Vytvoření nového jointového bodu

Následně vyskočí tabulka, zvolíme datový typ *jointRx* a pojmenujeme bod *jBezpeci*.

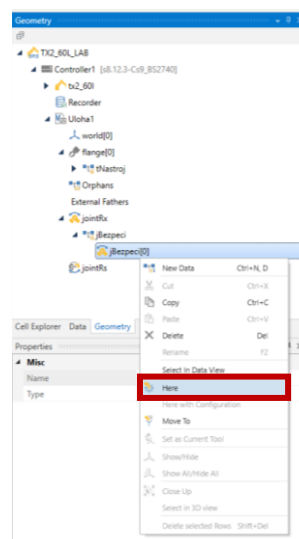


Obrázek 151 Volba datového typu a pojmenování jointového bodu

Následně si napolohujeme robota tak, aby *handler* nástroje byl tam, kde chceme bod *jBezpeci* mít umístěný. Máme-li napolohováno, tak po pravém kliknutí na bod *jBezpeci* vybereme z nabídky *Here*. Tím předáme bodu souřadnice, kde se nachází *handler* nástroje.

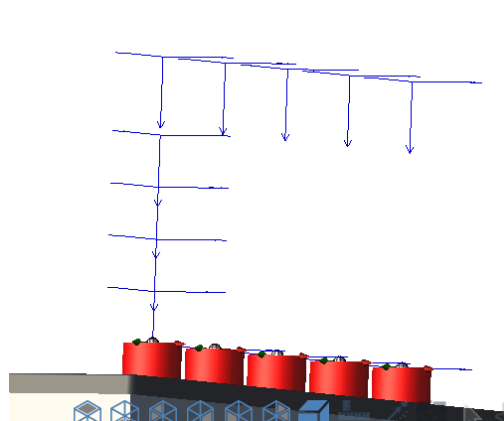


Obrázek 152 Napoložený robot pro naučení jBezpeci



Obrázek 153 Naučení bodu jBezpeci

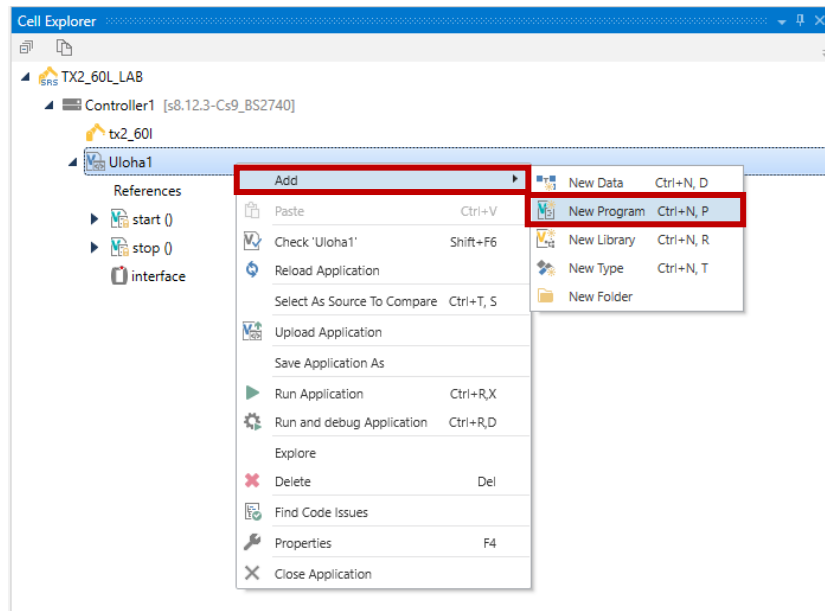
Dále si vytvoříme pole *pPick*, *pPlace* a *pPrejezd*.



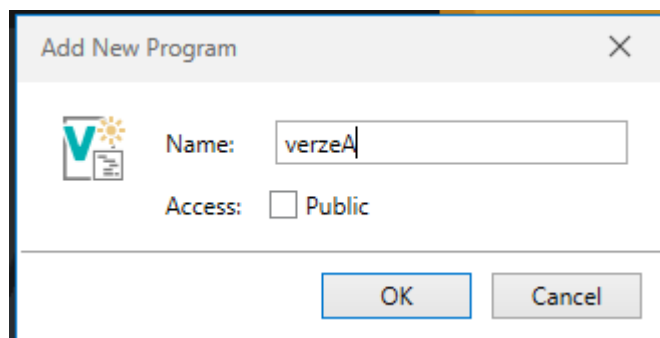
Obrázek 154 Správně napoložené body

10.1.6 Vytvoření programu

Nyní máme všechny potřebné body pro naprogramování verze úlohy a) a b). Výhodné bude vytvořit si nové programy, které budeme považovat za podprogramy, které budeme volat ve *start()*. Vytvoříme tedy *verzeA*, *verzeB* a *verzeC*.



Obrázek 155 Vytvoření podprogramu

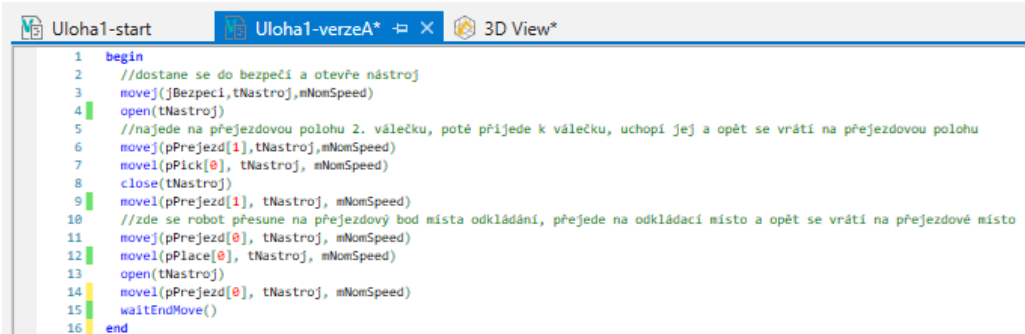


Obrázek 156 Pojmenování podprogramu

10.1.7 Verze A

Jako první je potřeba dostat robota do bezpečné pozice *jBezpeci*. To provedeme funkcí *movej()*. Jedná se o funkci, kdy se robot pohybuje pomocí pootočení kloubů. Tuto funkci je lepší používat převážně na přejezd, protože na napolohování není přesná tak, jako *movel()*. Do těchto funkcí se uvádí (cílový bod, nástroj, rychlost). Další důležitou věcí na začátku je otevření nástroje pomocí *open()*. Následovně můžeme začít vymýšlet pohyb robota na přejezdovou pozici nad válečkem, jeho uchopení a následné položení na sloupec. Nejprve přejezdem na *pPřejezd[1]* (bod z pole se určuje indexem), následně lineárním pohybem *movel()*

najedeme k válečku, uchopíme jej *close()* a opět pomocí lineárního pohybu dostaneme do přejezdového bodu.



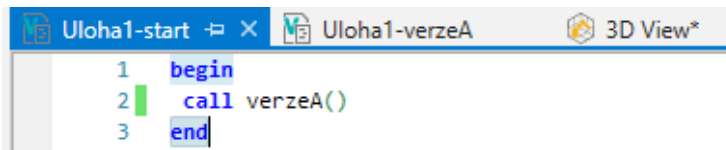
```

1 begin
2 //dostane se do bezpečí a otevře nástroj
3 movej(jBezpeci,tNastroj,mNomSpeed)
4 open(tNastroj)
5 //najede na přejezdovou polohu 2. válečku, poté přijede k válečku, uchopí jej a opět se vrátí na přejezdovou polohu
6 movej(pPřejezd[1],tNastroj,mNomSpeed)
7 movel(pPick[0], tNastroj, mNomSpeed)
8 close(tNastroj)
9 movel(pPřejezd[1], tNastroj, mNomSpeed)
10 //zde se robot přesune na přejezdový bod místa odkládání, přejeje na odkládací místo a opět se vrátí na přejezdové místo
11 movej(pPřejezd[0], tNastroj, mNomSpeed)
12 movel(pPlace[0], tNastroj, mNomSpeed)
13 open(tNastroj)
14 movel(pPřejezd[0], tNastroj, mNomSpeed)
15 waitEndMove()
16 end

```

Obrázek 157 Část naprogramovaného kódu

Nyní stejným způsobem postupujeme s programováním pohybu pro zbylé 3 válečky. Po přesunu posledního válečku se robot přesune opět do *jBezpeci* a také je důležité mít jako poslední řádek funkci *waitEndMove()*. Nyní spustíme simulaci a vyzkoušíme chod programu. Ve *start()* musíme pomocí *call* zavolat verziA.



```

1 begin
2 call verzeA()
3 end

```

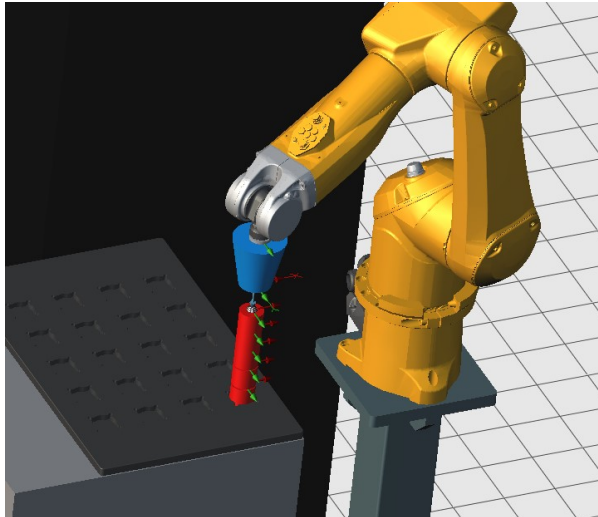
Obrázek 158 Volání verzeA

```

1 begin
2 //dostane se do bezpečí a otevře nástroj
3 movej(jBezpeci,tNastroj,mNomSpeed)
4 open(tNastroj)
5 //najede na přejezdovou polohu 2. válečku, poté přijede k válečku, uchopí jej a opět se vrátí na přejezdovou polohu
6 movej(pPřejezd[1],tNastroj,mNomSpeed)
7 movel(pPick[0], tNastroj, mNomSpeed)
8 close(tNastroj)
9 movel(pPřejezd[1], tNastroj, mNomSpeed)
10 //zde se robot přesune na přejezdový bod místa odkládání, přejeje na odkládací místo a opět se vrátí na přejezdové místo
11 movej(pPřejezd[0], tNastroj, mNomSpeed)
12 movel(pPlace[0], tNastroj, mNomSpeed)
13 open(tNastroj)
14 movel(pPřejezd[0], tNastroj, mNomSpeed)
15 //najede na přejezdovou polohu 3. válečku, poté přijede k válečku, uchopí jej a opět se vrátí na přejezdovou polohu
16 movej(pPřejezd[2],tNastroj,mNomSpeed)
17 movel(pPick[1], tNastroj, mNomSpeed)
18 close(tNastroj)
19 movel(pPřejezd[2], tNastroj, mNomSpeed)
20 //zde se robot přesune na přejezdový bod místa odkládání, přejeje na odkládací místo a opět se vrátí na přejezdové místo
21 movej(pPřejezd[0], tNastroj, mNomSpeed)
22 movel(pPlace[1], tNastroj, mNomSpeed)
23 open(tNastroj)
24 movel(pPřejezd[0], tNastroj, mNomSpeed)
25 //najede na přejezdovou polohu 4. válečku, poté přijede k válečku, uchopí jej a opět se vrátí na přejezdovou polohu
26 movej(pPřejezd[3],tNastroj,mNomSpeed)
27 movel(pPick[2], tNastroj, mNomSpeed)
28 close(tNastroj)
29 movel(pPřejezd[3], tNastroj, mNomSpeed)
30 //zde se robot přesune na přejezdový bod místa odkládání, přejeje na odkládací místo a opět se vrátí na přejezdové místo
31 movej(pPřejezd[0], tNastroj, mNomSpeed)
32 movel(pPlace[2], tNastroj, mNomSpeed)
33 open(tNastroj)
34 movel(pPřejezd[0], tNastroj, mNomSpeed)
35 //najede na přejezdovou polohu 5. válečku, poté přijede k válečku, uchopí jej a opět se vrátí na přejezdovou polohu
36 movej(pPřejezd[4],tNastroj,mNomSpeed)
37 movel(pPick[3], tNastroj, mNomSpeed)
38 close(tNastroj)
39 movel(pPřejezd[4], tNastroj, mNomSpeed)
40 //zde se robot přesune na přejezdový bod místa odkládání, přejeje na odkládací místo a opět se vrátí na přejezdové místo
41 movej(pPřejezd[0], tNastroj, mNomSpeed)
42 movel(pPlace[3], tNastroj, mNomSpeed)
43 open(tNastroj)
44 movel(pPřejezd[0], tNastroj, mNomSpeed)
45 waitEndMove()
46 end

```

Obrázek 159 Kompletní program verzeA



Obrázek 160 Výsledek kompletního programu
verzeA

10.1.8 Verze B

Nyní můžeme začít programovat další verzi programu, kde budeme využívat cyklus *For*. Program začne úplně stejně jako předchozí verze: Nejprve robot přejede do *jBezpeci* a otevře nástroj. Dále následuje již cyklus, který bude ve smyčce *For*. Tato smyčka má následující strukturu: **for** proměnná = DolníMez to HorníMez step krok. Část smyčky, kde je step není povinná. Ve *start()* zakomentujeme volání *verzeA()* a napíšeme řádek *call verzeB()*. Kompletní program vypadá následovně:

```
Uloha1-start  Uloha1-verzeB  3D View
1  begin
2  // call verzeA()
3  call verzeB()
4  end
```

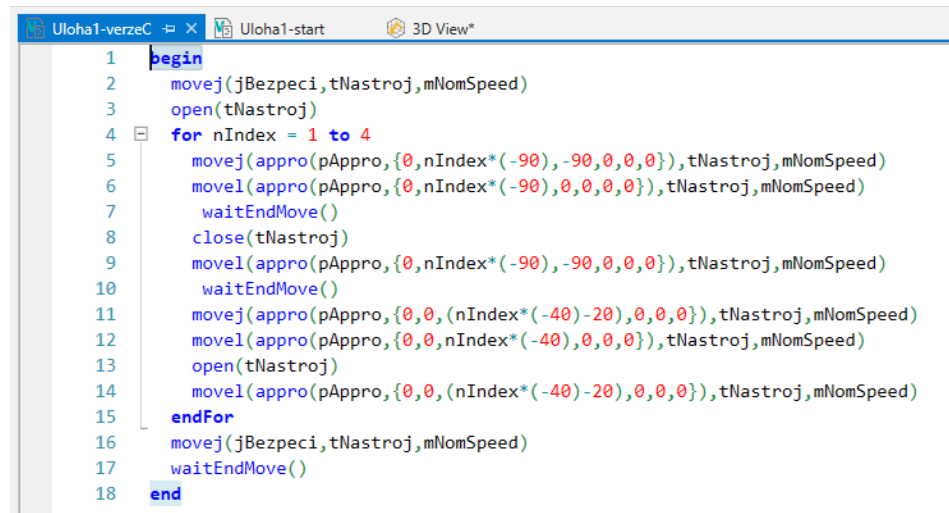
Obrázek 161 Hlavní program a volání verzeB

```
Uloha1-start  Uloha1-verzeB  3D View
1  begin
2  //dostane se do bezpeci a otevře nástroj
3  movej(jBezpeci,tNastroj,mNomSpeed)
4  open(tNastroj)
5  for nIndex = 0 to 3
6  movej(pPrejezd[(nIndex+1)],tNastroj,mNomSpeed)
7  movej(pPick[nIndex], tNastroj, mNomSpeed)
8  close(tNastroj)
9  movej(pPrejezd[(nIndex+1)], tNastroj, mNomSpeed)
10
11  movej(pPrejezd[0], tNastroj, mNomSpeed)
12  movej(pPlace[nIndex], tNastroj, mNomSpeed)
13  open(tNastroj)
14  movej(pPrejezd[0], tNastroj, mNomSpeed)
15  waitEndMove()
16  endFor
17  movej(jBezpeci,tNastroj,mNomSpeed)
18  waitEndMove()
19  end
```

Obrázek 162 Kompletní program verzeB

10.1.9 Verze C

V této verzi budeme využívat funkce *apro(bod,transfer)*, kdy *transfer* je: $\{X,Y,Z,Rx,Ry,Rz\}$. Tato funkce umožňuje přístup bodu se změněnými souřadnicemi nebo pootočením. To znamená, že bude stačit jeden bod k provedení požadavku jako ve verzích *A* a *B*.



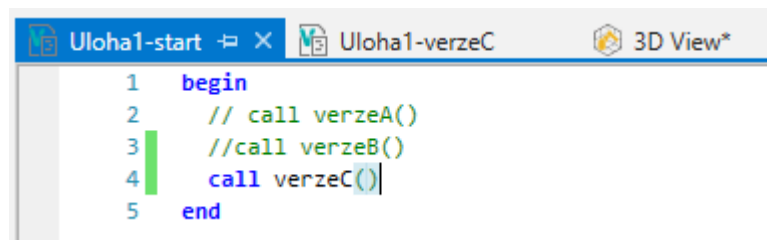
```

1 begin
2   movej(jBezpeci,tNastroj,mNomSpeed)
3   open(tNastroj)
4   for nIndex = 1 to 4
5     movej(apro(pAppro,{0,nIndex*(-90),-90,0,0,0}),tNastroj,mNomSpeed)
6     movel(apro(pAppro,{0,nIndex*(-90),0,0,0,0}),tNastroj,mNomSpeed)
7     waitEndMove()
8     close(tNastroj)
9     movel(apro(pAppro,{0,nIndex*(-90),-90,0,0,0}),tNastroj,mNomSpeed)
10    waitEndMove()
11    movej(apro(pAppro,{0,0,(nIndex*(-40)-20),0,0,0}),tNastroj,mNomSpeed)
12    movel(apro(pAppro,{0,0,nIndex*(-40),0,0,0}),tNastroj,mNomSpeed)
13    open(tNastroj)
14    movel(apro(pAppro,{0,0,(nIndex*(-40)-20),0,0,0}),tNastroj,mNomSpeed)
15  endFor
16  movej(jBezpeci,tNastroj,mNomSpeed)
17  waitEndMove()
18 end

```

Obrázek 163 Kompletní program verzeC

Ve *start()* zakomentujeme obě předchozí verze a zavoláme poslední verzi: *call verzeC()*. Následně ve *verziC* začneme přejetím do *jBezpeci*, otevřením nástroje a následně můžeme přes cyklus *For* a *apro* začít opět s programováním pohybu.



```

1 begin
2   // call verzeA()
3   //call verzeB()
4   call verzeC()
5 end

```

Obrázek 164 Volání verzeC

10.1.10 Program pro reálného robota

Program musíme upravit, je potřeba funkce nástroje *open(tNastroj)* a *close(tNastroj)* prohodit. To znamená, že místo *open(tNastroj)* bude *close(tNastroj)* a naopak.

```

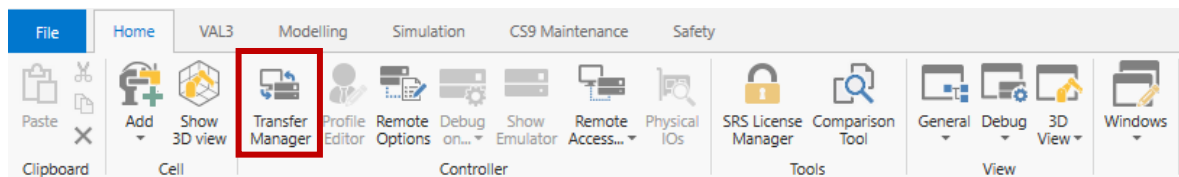
1  begin
2  //dostane se do bezpečí a otevře nástroj
3  movej(jBezpeci,tNastroj,mNomSpeed)
4  close(tNastroj)
5  for nIndex = 0 to 3
6  //pick
7  movej(appro(pPick[nIndex],{0,0,-220,0,0,0}),tNastroj,mNomSpeed)
8  movel(pPick[nIndex], tNastroj, mNomSpeed)
9  waitEndMove()
10 open(tNastroj)
11 movel(appro(pPick[nIndex],{0,0,-220,0,0,0}),tNastroj,mNomSpeed)
12 //place
13 movej(appro(pPlace[0],{0,0,-220,0,0,0}), tNastroj, mNomSpeed)
14 movel(appro(pPlace[0],{0,0,nIndex*(-40),0,0,0}), tNastroj, mNomSpeed)
15 waitEndMove()
16 close(tNastroj)
17 movej(appro(pPlace[0],{0,0,-220,0,0,0}), tNastroj, mNomSpeed)
18 waitEndMove()
19 endFor
20 movej(jBezpeci,tNastroj,mNomSpeed)
21 waitEndMove()
22 end

```

Obrázek 165 Upravený program

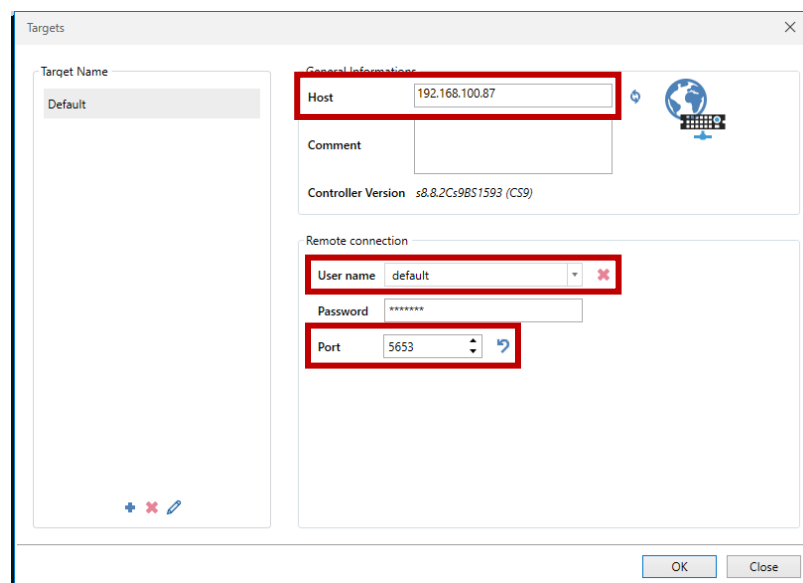
10.1.11 Připojení k reálnému robotu

Přes *Transfer Manager* v kartě *Home* se připojíme k reálnému robotu.



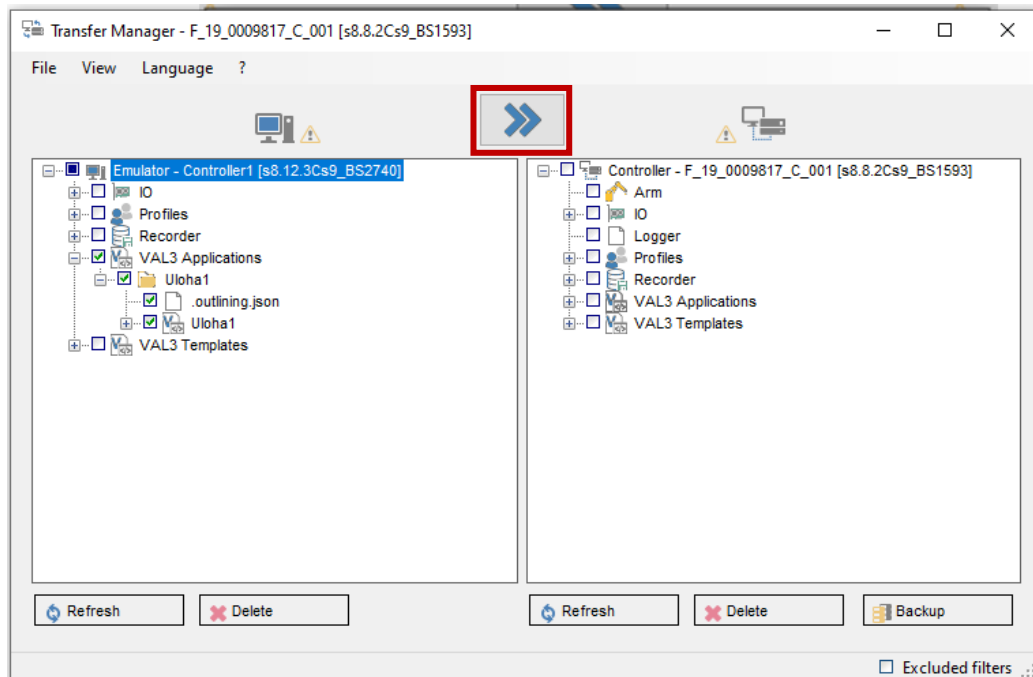
Obrázek 166 Transefer Manager

Objeví se okno, kde si překontrolujeme, že *Host* je *192.168.100.87* a *User name* *Default* a *Port* je *5693*.



Obrázek 167 Tabulka pro připojení k robotu

Pro nahrání aplikace do robota si rozklikneme *VAL3 Applications* a volíme *Uloha1*. Následně klikneme na nahrání a počkáme, než se úloha nahraje.



Obrázek 168 Nahrání aplikace do robota

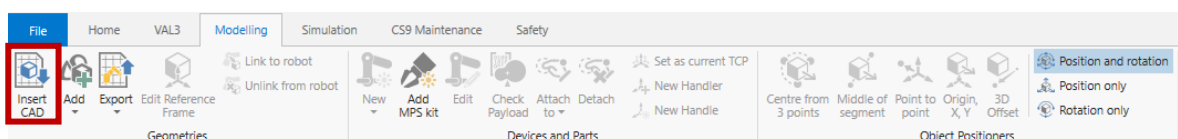
10.2 Úloha 2

10.2.1 Zadání

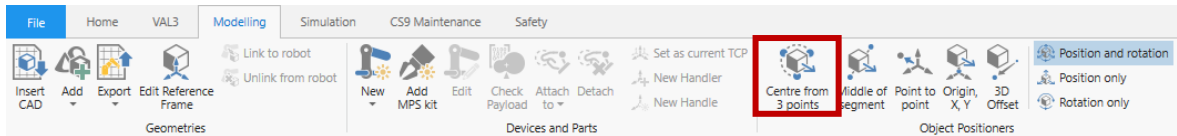
Přesuňte 3 válečky se středovou dírou nejprve z paletky na hřídel a následně z hřídele na poličku. Odkládání válečků na poličku je libovolné a není tedy pevně dané. Přejezdy robota pečlivě zvažte a podle toho volte správné umístění přejezdových bodů. Primárním cílem této úlohy je seznámení se s polohováním úchopné hlavy, aby byl robot schopen válečky vzít z hřídele. Nejprve vytvořte simulační aplikaci, kde si připravíte potřebné body. Následně tuto aplikaci upravte pro reálného robota s reálným nástrojem a ověřte funkčnost aplikace.

10.2.2 Vložení CAD objektu

Vložíme si válečky, se kterými budeme pracovat. V kartě *Modelling* vybereme *Insert CAD* a následovně zvolíme *Valecek_Stred.stl*.

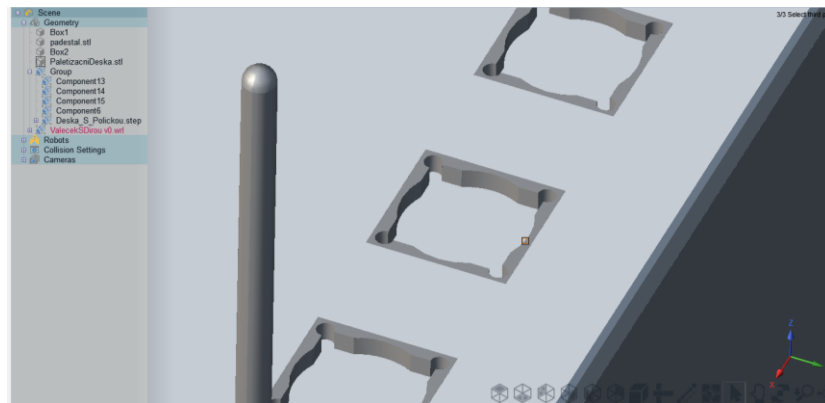


Obrázek 169 Vložení válečků

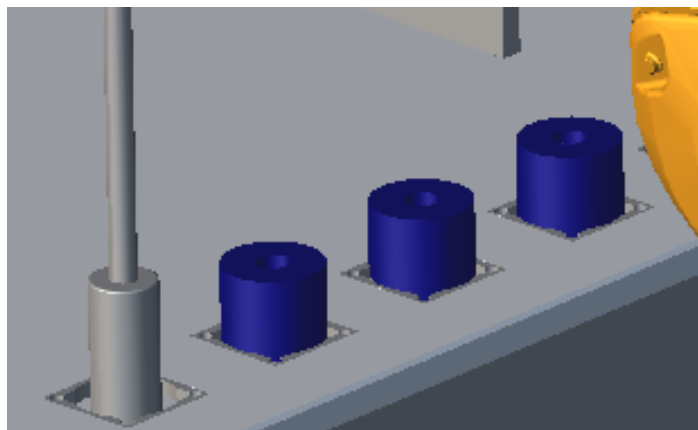


Obrázek 170 Výběr možnosti přesunu objektu

Pomocí *Object Positioners* a *Centre from 3 points* jej umístíme do drážky.



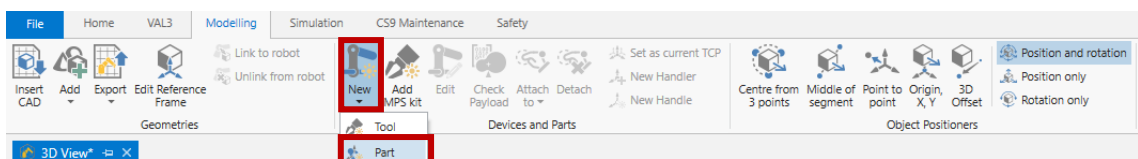
Obrázek 171 Našeptávač umístění



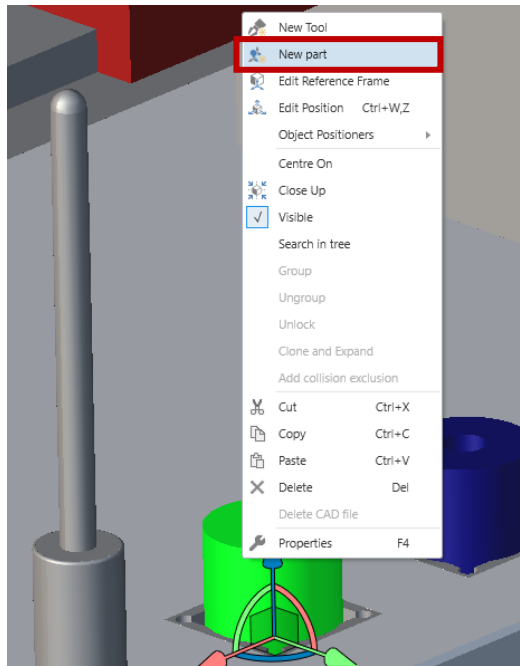
Obrázek 172 3 válečky umístěné v drážkách

10.2.3 Převedení válečku na součástku

Nyní máme 3 válečky a musíme je změnit na součástky. To provedeme přes pravé tlačítko a zvolíme *New Part* nebo přes kartu *Modelling*, *New* a *Part*.

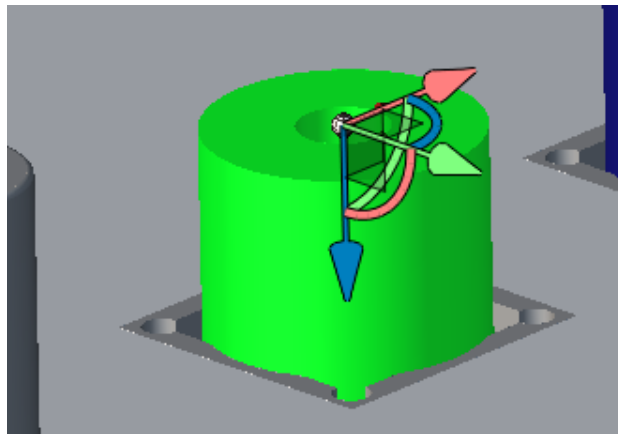


Obrázek 173 Převedení objektu na součástku



Obrázek 174 Převedení objektu na součástku

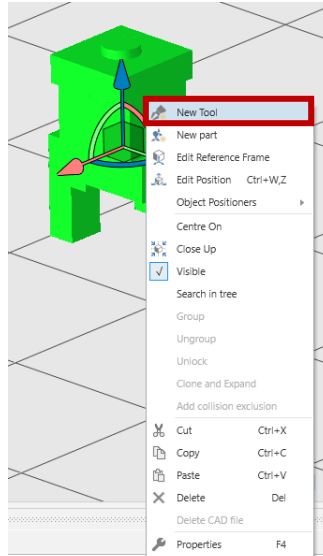
Handle součástky se automaticky naorientuje. Pokud ale není správně naorientovaný, tak jej přes *Edit Position* přeorientujeme.



Obrázek 175 Správně umístěný handle

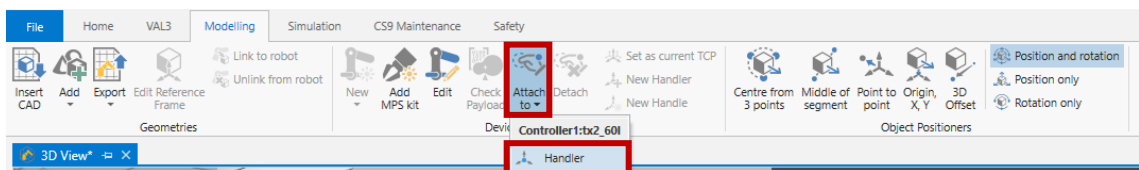
10.2.4 Vložení nástroje

Máme-li všechny 3 válečky změněné na součástky, tak je potřeba vložit si nástroj. Opět přes kartu *Modelling* a *Insert CAD* vložíme soubor *Gripper_2prst.step*. Ten následovně převedeme na nástroj.



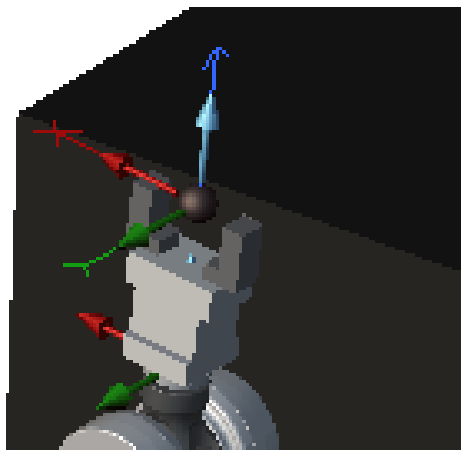
Obrázek 176 Převedení
objektu na nástroj

Nástroj můžeme připevnit k *handleru* robota pomocí *Attach to*.



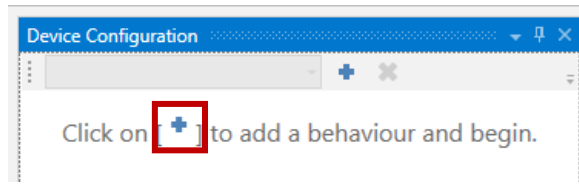
Obrázek 177 Připevnění nástroje k robotu

Po připevnění by měl nástroj mít stejně orientované osy jako *flange* robota.



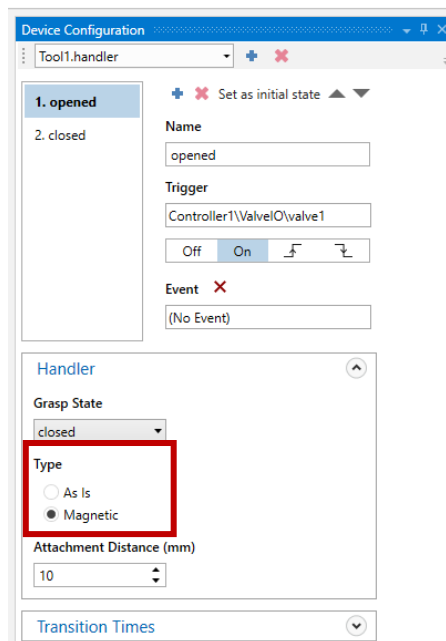
Obrázek 178 Kontrola orientace nástroje

Pravým kliknutím na nástroj ve vizualizaci se zobrazí nabídka, ze které vybereme *Device Configuration*. Vyskočí okno kde klikneme na modré plusko, čímž přidáme nové chování nástroje



Obrázek 179 Přidání nového chování nástroje

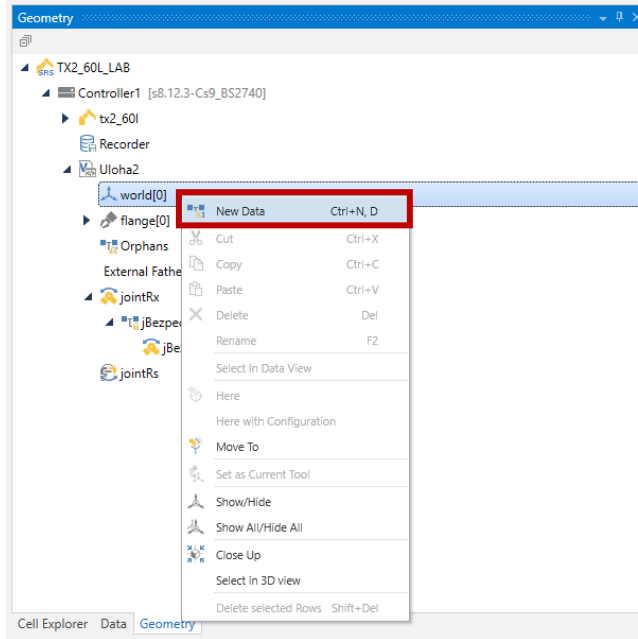
V následující tabulce změníme *Type* z *As Is* na *Magnetic*. Následovně ve stromu scény na *handler* nástroje klikneme pravým tlačítkem a dáme *Copy Tool Data*.



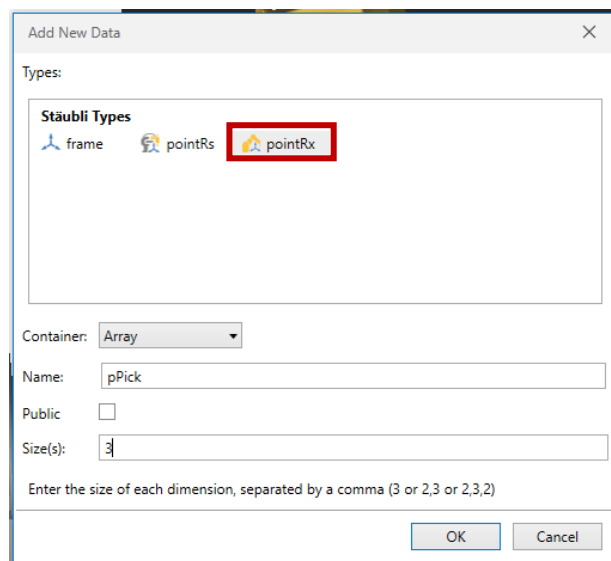
Obrázek 180 Nastavení chování

10.2.5 Definování bodů

vytvoříme pole bodů pro uchopení válečků. Pravým kliknutím na *world* a následným vybráním *New data* se objeví tabulka, kde vybereme typ bodů, pojmenujeme *pPick* a jako rozměr dáme 3 body.

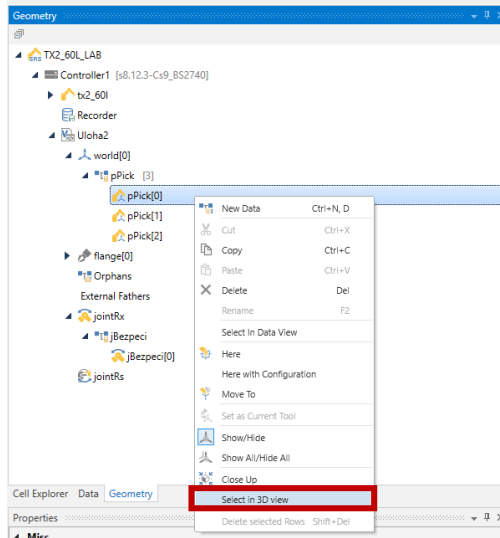


Obrázek 181 Vytvoření nového pole

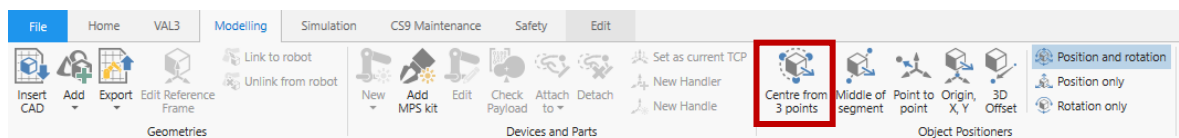


Obrázek 182 Definice pole pPick

Poté klikneme pravým na bod *pPick[0]* a zvolíme *Select in 3D view*. Přes kartu *Modelling* a *Object Positioners* použijeme *Centre from 3 points* a přemístíme bod na vršek prvního válečku. První váleček bude brán jako první zprava.

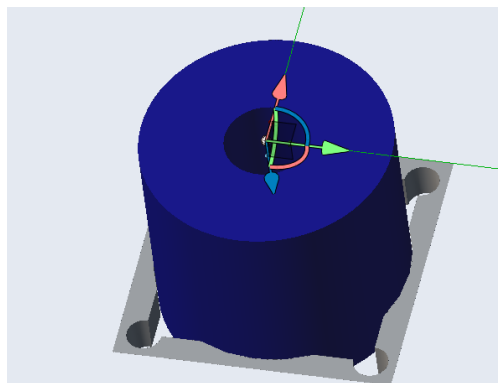


Obrázek 183 Výběr bodu pPick[0] do 3D vizualizace



Obrázek 184 Přesunutí bodu pomocí Object Positioners

Je-li potřeba upravit orientaci bodu, tak použijeme *Edit Position*, které najdeme v kartě *Home*, odstavci *View*, dále *3D view* a poté *Edit Position*.



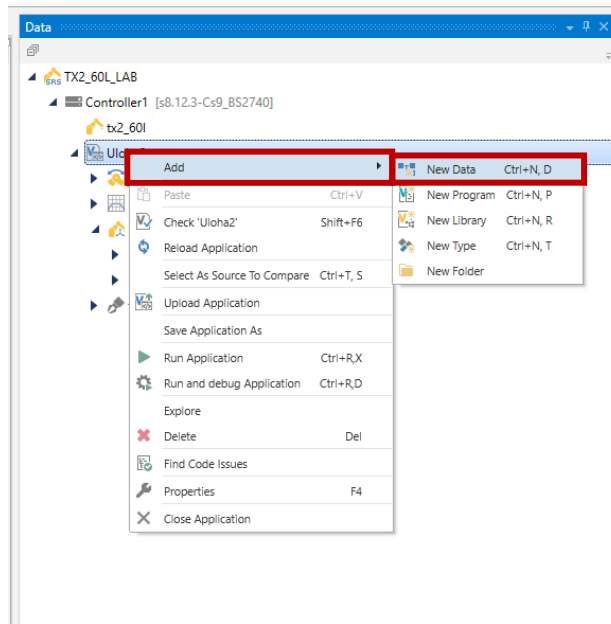
Obrázek 185 Správně umístěný bod pPick[0]

Stejný postup opakujeme pro zbylé 2 válečky. Dále vytvoříme pole *pPlace*, *pPrejez*, *pOdklad* a *pOsa*.

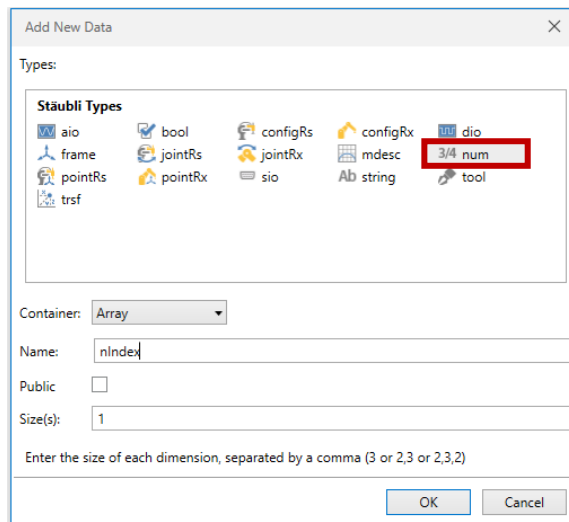
10.2.6 Programování pohybu

Budeme využívat cyklus *For*, tudíž je potřeba vytvořit číselnou proměnnou.

V kartě **Data** klikneme pravým na **Uloha2**, **Add a New data**. V následujícím okně volíme typ **num** a pojmenujeme proměnou **nIndex**.

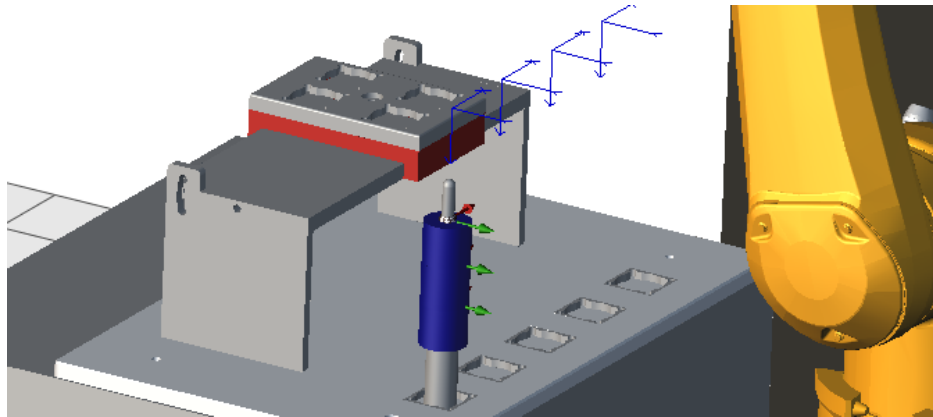


Obrázek 186 Vytvoření číselné proměnné



Obrázek 187 Definice nIndex

Nyní vytvoříme program nejprve pro přesun válečků z drážky na hřídel. Ten otestujeme a následně přepíšeme druhou část programu na odkládání válečků na polici.



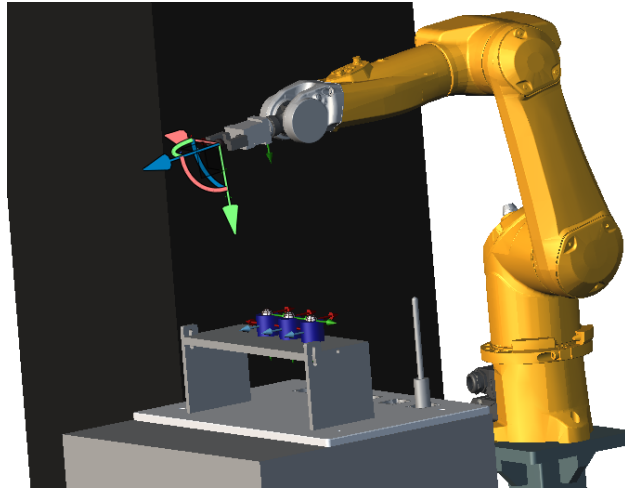
Obrázek 188 Výsledek první části programu

```

1  begin
2  movej(jBezpeci, tNastroj, mNomSpeed)
3  open(tNastroj)
4  for nIndex = 0 to 2
5      //náběr válečku
6      movej(pPrejezd[nIndex], tNastroj, mNomSpeed)
7      movel(appro(pPick[nIndex], {0,0,-30,0,0,0}), tNastroj, mNomSpeed)
8      movel(pPick[nIndex], tNastroj, mNomSpeed)
9      waitEndMove()
10     close(tNastroj)
11     movej(pPrejezd[nIndex], tNastroj, mNomSpeed)
12     //přejezd nad osu
13     movej(pPrejezd[3], tNastroj, mNomSpeed)
14     movel(appro(pPrejezd[3], {0,0,40,0,0,0}), tNastroj, mNomSpeed)
15     open(tNastroj)
16     movel(pPrejezd[3], tNastroj, mNomSpeed)
17     waitEndMove()
18 endFor
19 movej(jBezpeci, tNastroj, mNomSpeed)
20 waitEndMove()
21 for nIndex2 = 1 to 3
22     //přejezd nad osu
23     movej(pOsa[0], tNastroj, mNomSpeed)
24     waitEndMove()
25     movel(pOsa[nIndex2], tNastroj, mNomSpeed)
26     close(tNastroj)
27     movej(pOsa[0], tNastroj, mNomSpeed)
28     //přejezd nad poličku
29     movej(appro(pOdklad[nIndex2-1], {0,-60,0,0,0,0}), tNastroj, mNomSpeed)
30     movel(pOdklad[nIndex2-1], tNastroj, mNomSpeed)
31     open(tNastroj)
32     movel(appro(pOdklad[nIndex2-1], {0,-60,0,0,0,0}), tNastroj, mNomSpeed)
33     waitEndMove()
34 endFor
35 movej(jBezpeci, tNastroj, mNomSpeed)
36 waitEndMove()
37 end

```

Obrázek 189 Kompletní program



Obrázek 190 Výsledek programu

10.2.7 Převod na reálného robota

Prvně upravíme nástroj tak, že jeho *handler* přesuneme o 10 mm dovnitř, což můžeme udělat přes data nástroje. Dále upravíme program tak, že funkce nástroje *open(tNastroj)* a *close(tNastroj)* prohodíme. To znamená, že místo *open(tNastroj)* bude *close(tNastroj)* a naopak.

```

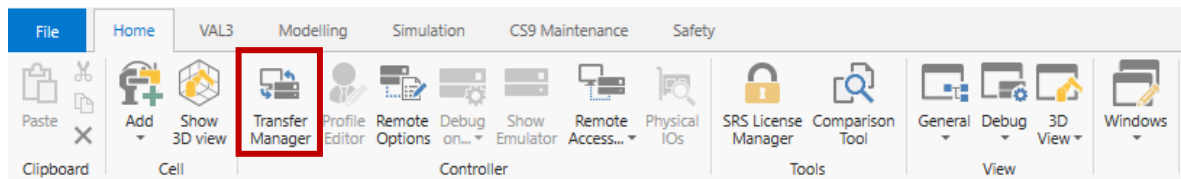
Uloha2-start  x  Uloha2-tNastroj Data  3D View*
1  begin
2  movej(jBezpeci, tNastroj, mNomSpeed)
3  close(tNastroj)
4  for nIndex = 0 to 2
5  //náběr válečku
6  movej(pPrejezd[nIndex], tNastroj, mNomSpeed)
7  movel(appro(pPick[nIndex], {0, 0, -30, 0, 0, 0}), tNastroj, mNomSpeed)
8  movel(pPick[nIndex], tNastroj, mNomSpeed)
9  waitEndMove()
10 open(tNastroj)
11 movel(pPrejezd[nIndex], tNastroj, mNomSpeed)
12 //přejezd nad osu
13 movej(pPrejezd[3], tNastroj, mNomSpeed)
14 movel(appro(pPrejezd[3], {0, 0, 40, 0, 0, 0}), tNastroj, mNomSpeed)
15 close(tNastroj)
16 movel(pPrejezd[3], tNastroj, mNomSpeed)
17 waitEndMove()
18 endFor
19 movej(jBezpeci, tNastroj, mNomSpeed)
20 waitEndMove()
21 for nIndex2 = 1 to 3
22 //přejezd nad osu
23 movej(pOsa[0], tNastroj, mNomSpeed)
24 waitEndMove()
25 movel(pOsa[nIndex2], tNastroj, mNomSpeed)
26 open(tNastroj)
27 movel(pOsa[0], tNastroj, mNomSpeed)
28 //přejezd nad poličku
29 movej(appro(pOdklad[nIndex2-1], {0, -60, 0, 0, 0, 0}), tNastroj, mNomSpeed)
30 movel(pOdklad[nIndex2-1], tNastroj, mNomSpeed)
31 close(tNastroj)
32 movel(appro(pOdklad[nIndex2-1], {0, -60, 0, 0, 0, 0}), tNastroj, mNomSpeed)
33 waitEndMove()
34 endFor
35 movej(jBezpeci, tNastroj, mNomSpeed)
36 waitEndMove()
37 end

```

Obrázek 191 Upravený program na reálného robota

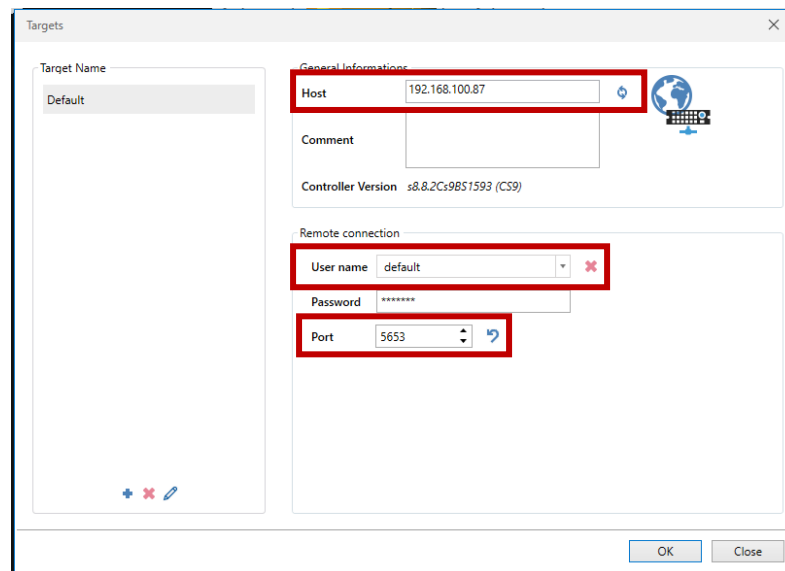
10.2.8 Připojení k reálnému robotu

Přes *Transfer Manager* v kartě *Home* se připojíme k reálnému robotu.



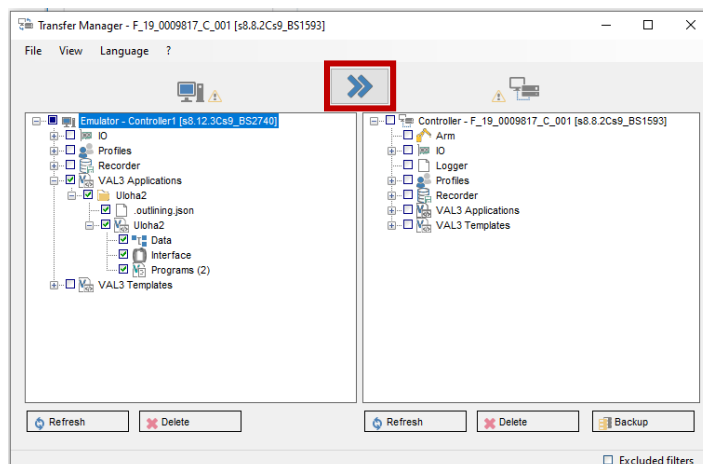
Obrázek 192 Transefer Manager

Objeví se okno, kde si přezkontrolujeme, že *Host* je **192.168.100.87** a *User name* **Default** a *Port* je **5693**.



Obrázek 193 Tabulka pro připojení k robotu

Pro nahrání aplikace do robotu si rozklikneme *VAL3 Applications* a volíme *Uloha2*. Následně klikneme na nahrání a počkáme, než se úloha nahraje.



Obrázek 194 Nahrání aplikace do robotu

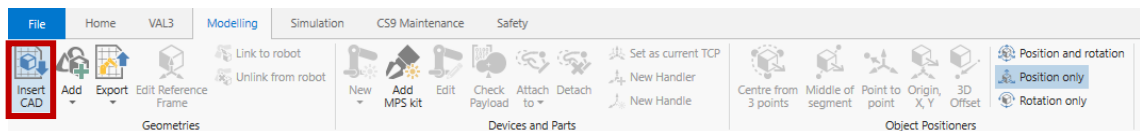
10.3 Úloha 3

10.3.1 Zadání

Vytvořte simulaci, kde se budou přemísťovat 4 válečky z palety do paletky na poličce. Tuto paletku na poličce poté přemístíte a proces opakujete. Využijte možnosti vytvořit si svůj vlastní souřadný systém, ve kterém definujete body paletky na poličce. Zhodnoťte využití vlastního souřadného systému při přemísťování oproti jednotlivému přeučení každého bodu na paletce. Dále převed'te simulaci na reálného robota a zhodnocení ověřte. Využijte možnost náklonu jak poličky, tak i paletky.

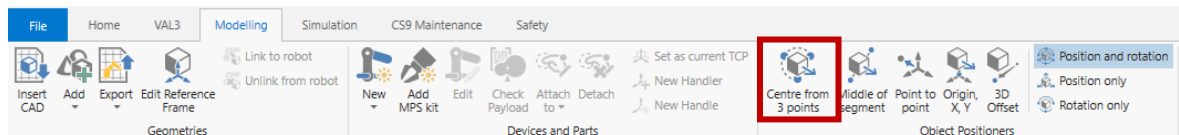
10.3.2 Vložení CAD objektu

Vložíme 4 válečky. V kartě **Modelling** klikneme na **Insert CAD**. V následující tabulce najdeme **Valecek_45x40_mm** a vložíme jej.

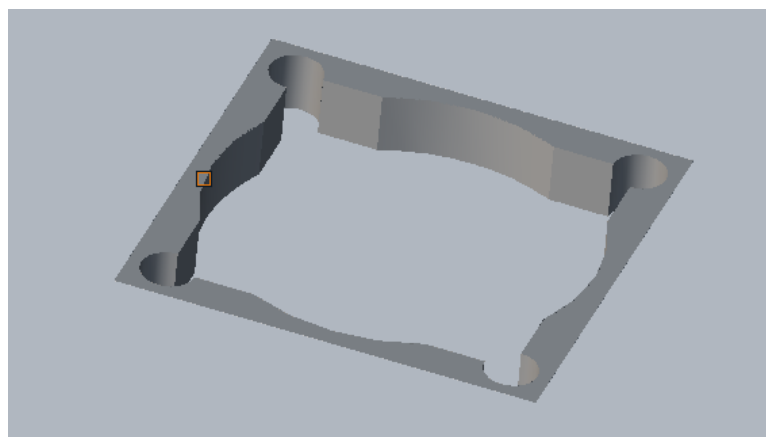


Obrázek 195 Výběr možnosti Insert CAD

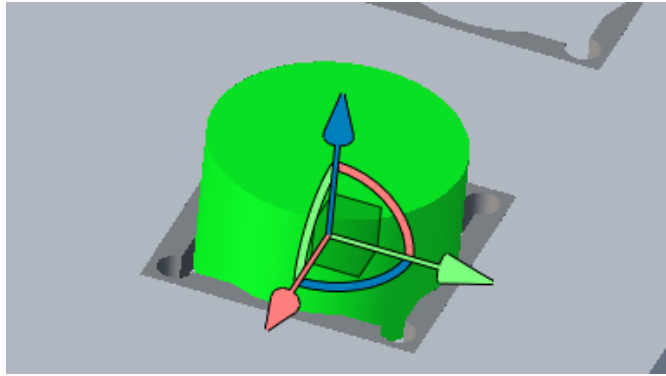
Váleček pomocí **Object Positioners** a **Centre from 3 points** přesuneme do drážky na desce.



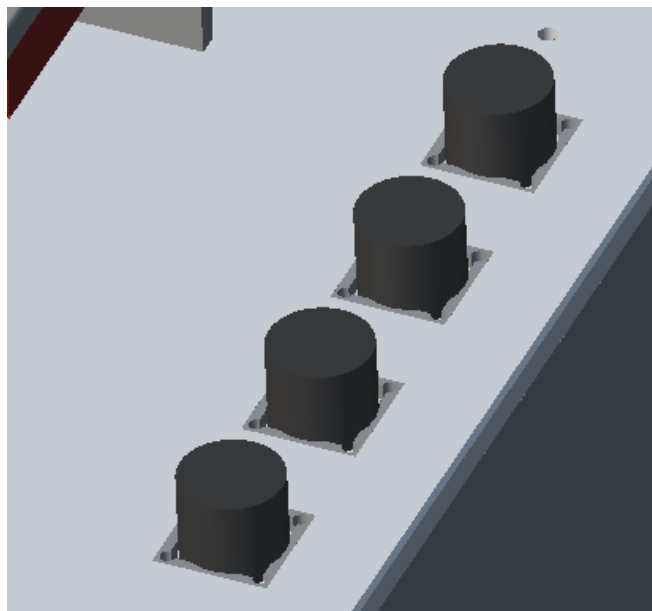
Obrázek 196 Zvolení množnosti přesunu válečku



Obrázek 197 Našeptávač označení hrany



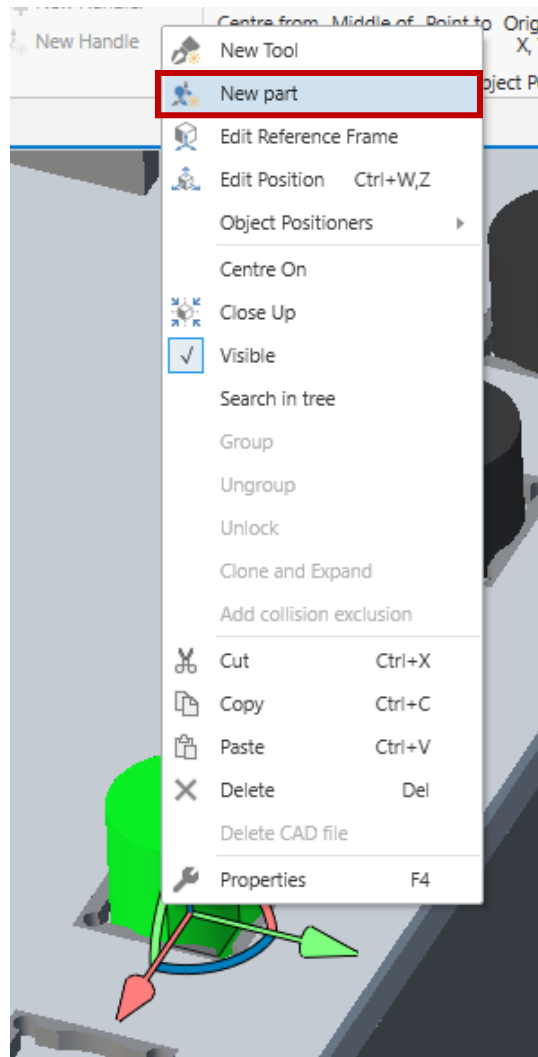
Obrázek 198 Umístěný váleček



Obrázek 199 4 válečky na hlavní desce

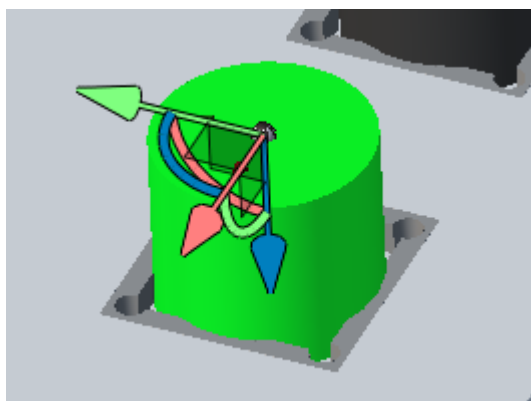
10.3.3 Převod válečků na součástky

Nyní válečky musíme převést na součástky. Máme-li označený váleček, tak přes pravé tlačítko a zvolíme *New Part*.

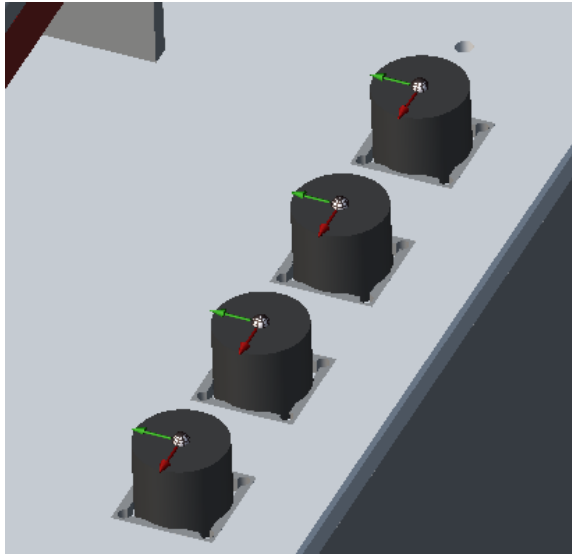


Obrázek 200 Převedení válečku na součást

Tímto se váleček přemění na součást a automaticky se vygeneruje *handle* a automaticky se umístí viz. následující obrázek č.19.



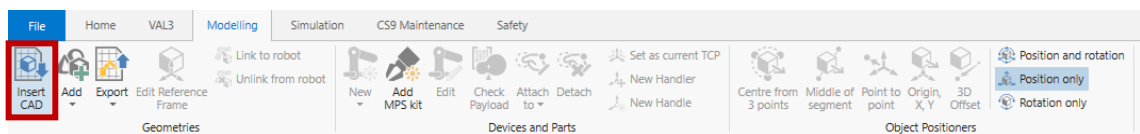
Obrázek 201 Umístěný handle



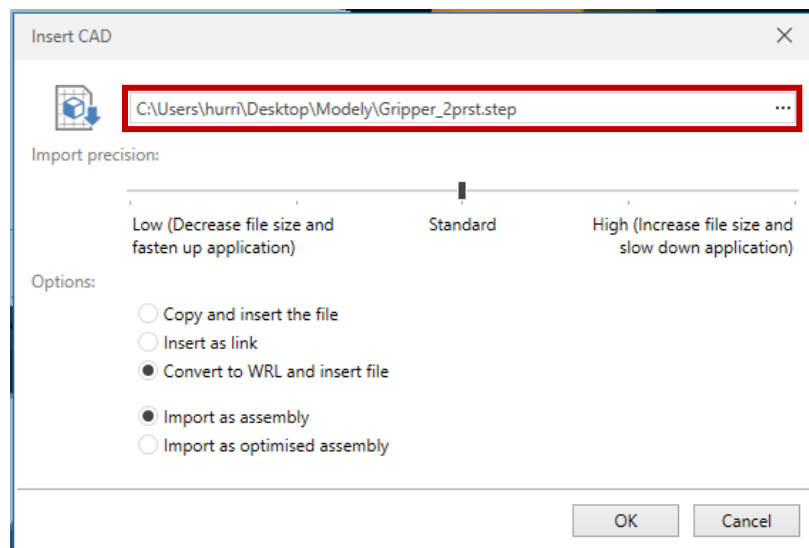
Obrázek 202 4 součástky

10.3.4 Vložení nástroje

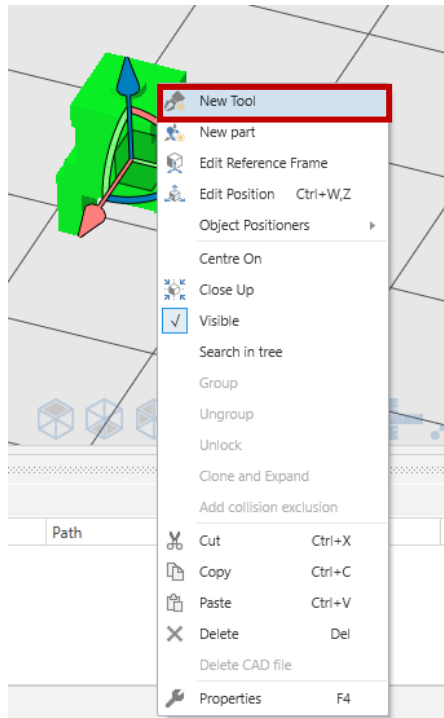
V kartě *Modelling* zvolíme *Insert CAD* a vložíme soubor *Gripper_2prst.step*. Ten pomocí pravého kliknutí a volby *New Tool* převedeme na nástroj.



Obrázek 203 Vložení nástroje

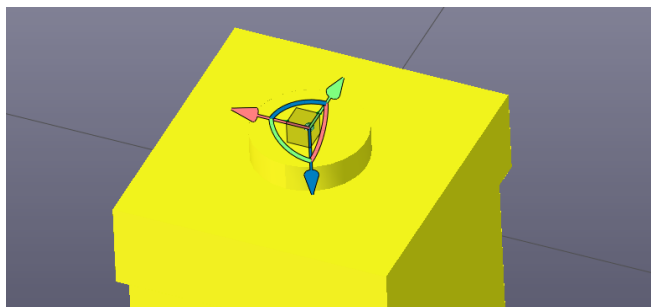


Obrázek 204 Zvolení souboru pro vložení



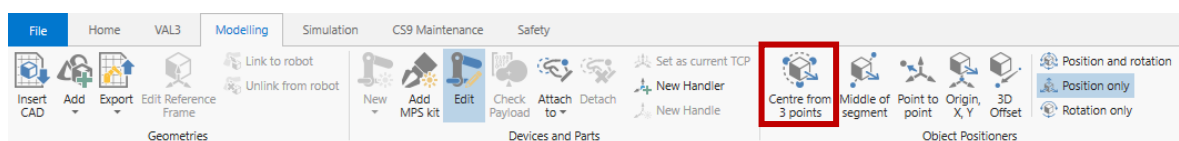
Obrázek 205 Převedení objektu na nástroj

Ve stromě scény si rozklikneme *Tools*, dále *Tool1* a označíme si *handle*. Ten přemístíme na vrchol objektu přes *Object Positioners* a *Centre from 3 points*. Bude-li potřeba *handle* přeorientovat, použijeme *Edit Position*.

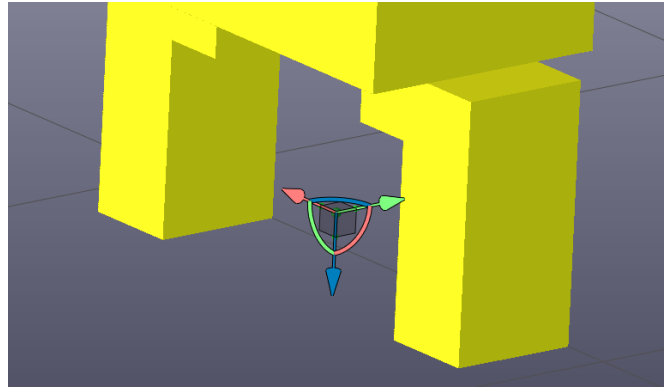


Obrázek 206 Umístěný handle nástroje

Nyní si označíme *handler* a ten umístíme mezi kleště nástroje. To provedeme zvolením *Centre from 3 points* z *Object positioners*. *Handler* následně posuneme o - 10 mm v ose Z. Výsledné umístění bude viditelné na obrázku č.28.

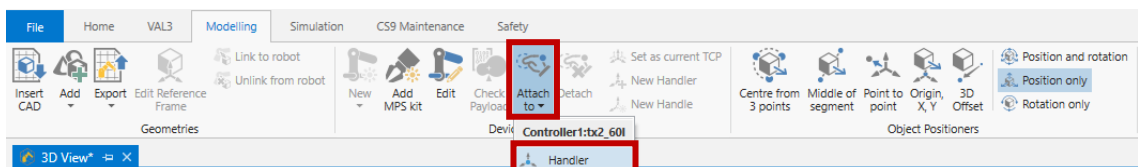


Obrázek 207 Výběr možnosti přesunu handleru

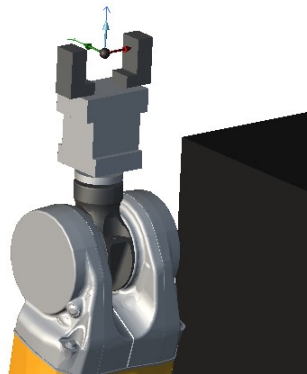


Obrázek 208 Správně umístěný handler

Nyní nástroj můžeme připevnit přes *Attach to* a zvolíme *handler* robota.

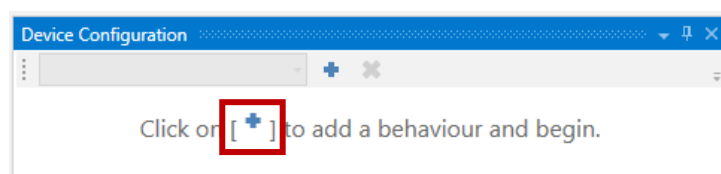


Obrázek 209 Připevnění nástroje k robotu

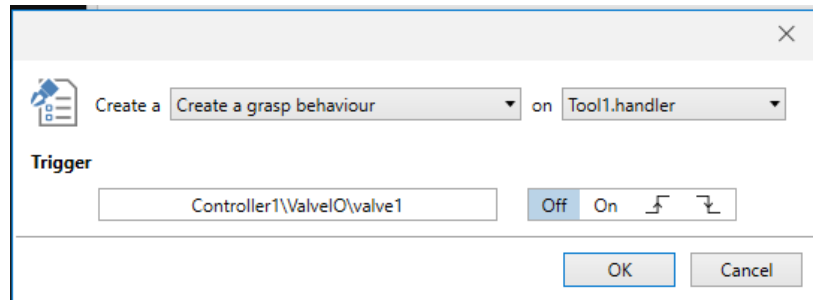


Obrázek 210 Připevněný nástroj

Teď je potřeba nastavit tomuto nástroji chování. Po pravém kliknutí na nástroj se zobrazí list možností, ze kterého vybereme možnost *Device Configuration*, která nám otevře nové okno, kde definujeme chování nástroje. V tomto okně klikneme na modré plus. Následovně vyskočí okno, kde vybereme nástroje, kterému chceme přidělit chování a následně nastavíme cestu k fyzickému zařízení.

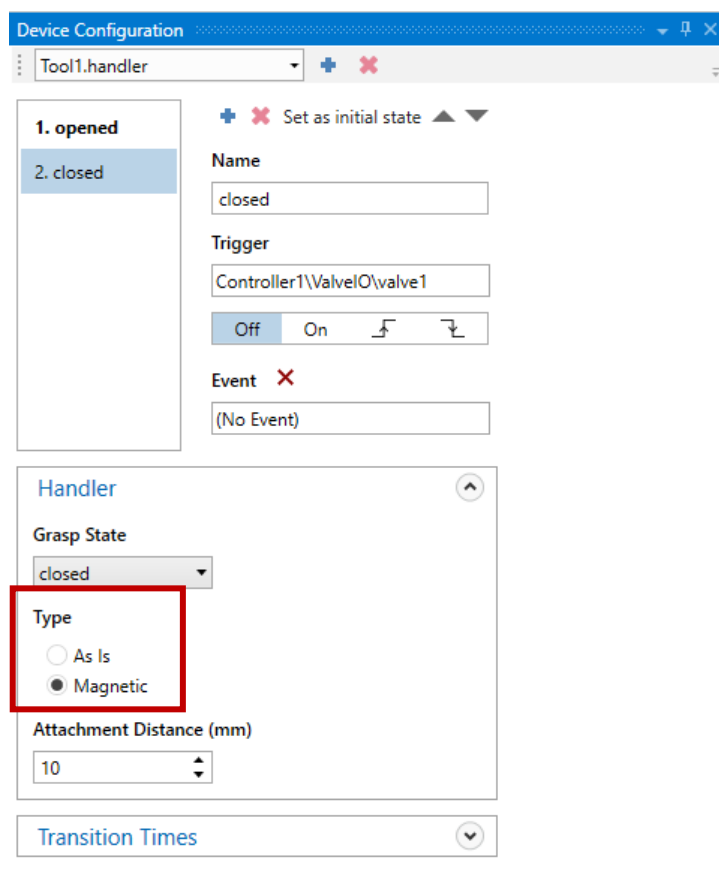


Obrázek 211 Přidání chování nástroje



Obrázek 212 Nastavení cesty k fyzickému zařízení

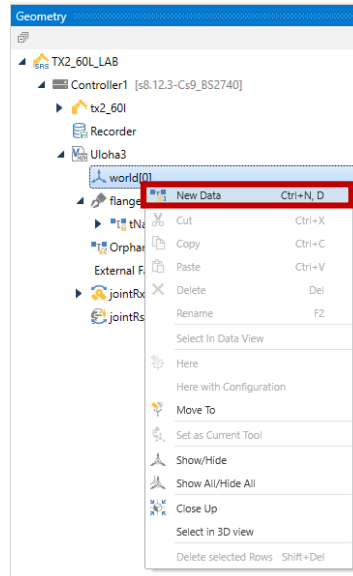
Znovu se nám otevře okno *Device Configuration*, kde změníme v *Handler Type* z *As Is* na *Magnetic*.



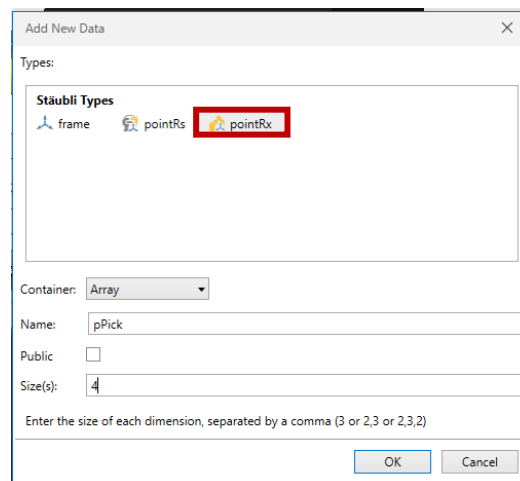
Obrázek 213 Nastavení chování nástroje

10.3.5 Definice bodů

Vytvoříme si pole bodů pro nabírání válečků. Pravým kliknutím na *world* z listu vybereme *New Data*. V dalším okně vybereme typ *pointRx*, pojmenujeme jej *pPick* a velikost pole bude 4.

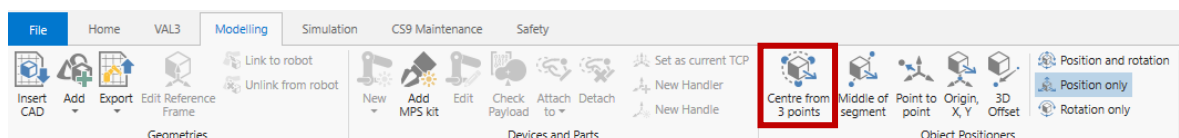


Obrázek 214 Vytvoření nového pole bodů



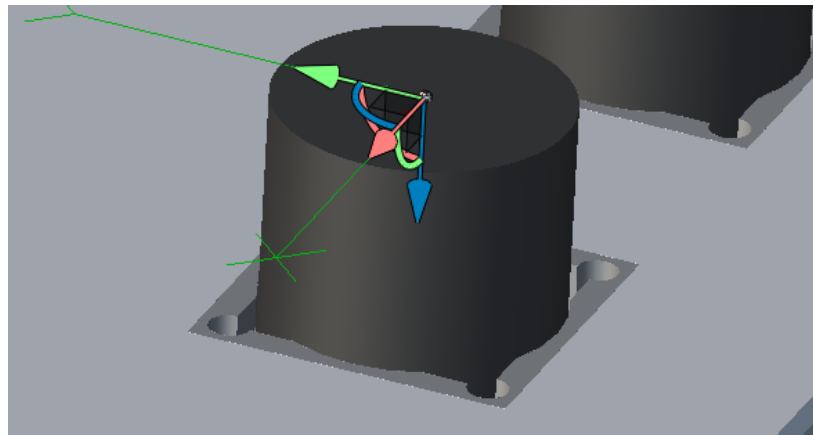
Obrázek 215 Definice pole pPick

Nyní tyto body přemístíme. Pravým kliknutím na **pPick[0]** a výběrem **Select in 3D view** vybereme bod v 3D vizualizaci a nyní jej můžeme pomocí **Object Positioners** a **Centre from 3 points** přesunout na vrchol prvního válečku.

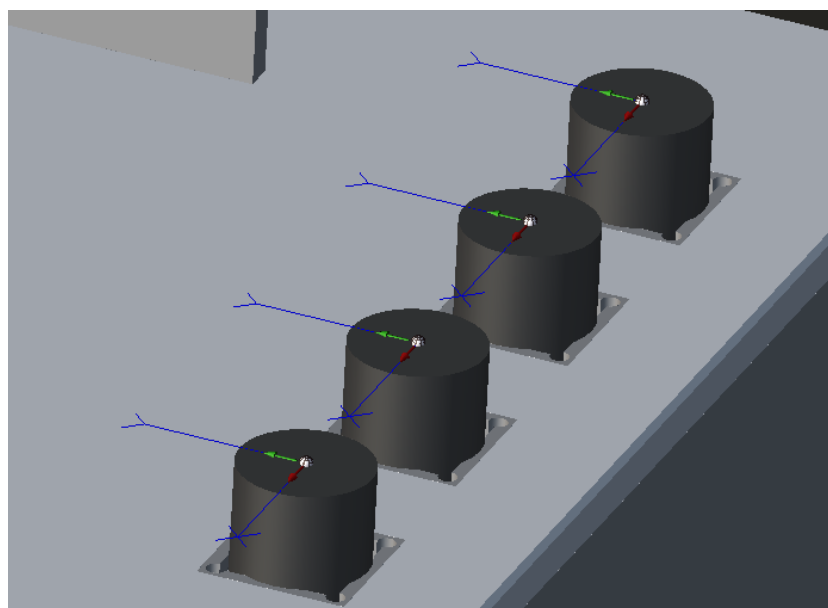


Obrázek 216 Výběr možnosti přesunutí bodu

Bude-li bod potřeba přeorientovat, tak akci provedeme pomocí **Edit Position**.



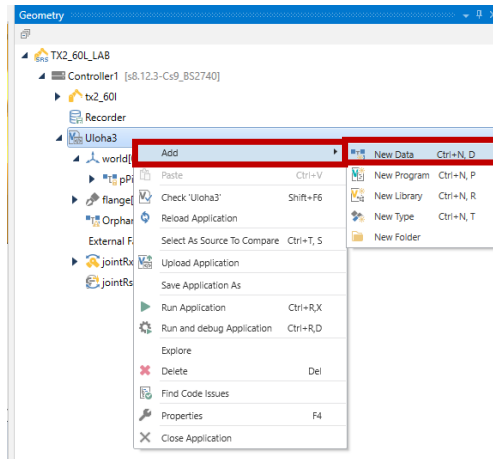
Obrázek 217 Správně umístěný bod pPick[0]



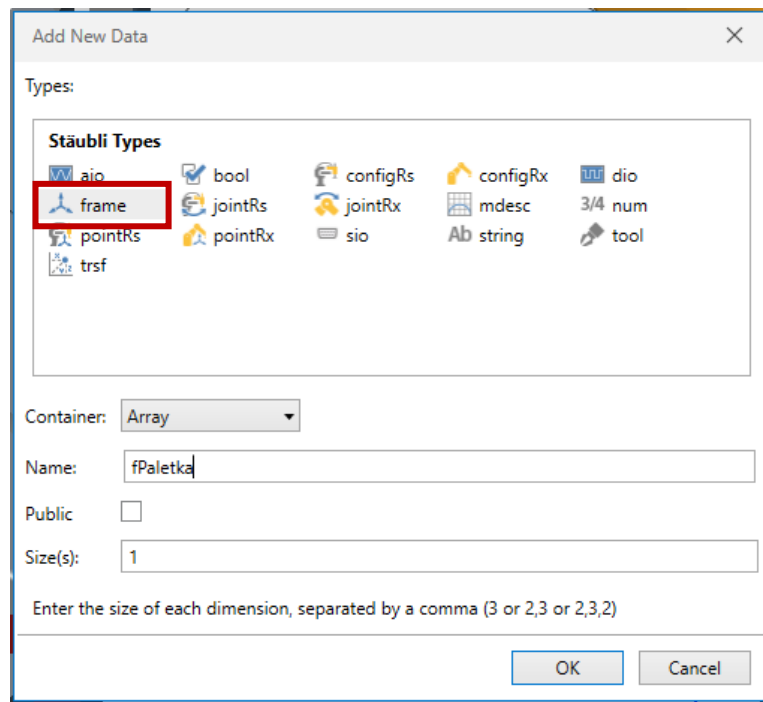
Obrázek 218 Umístění bodů z pole pPick

10.3.6 Vytvoření vlastního souřadného systému

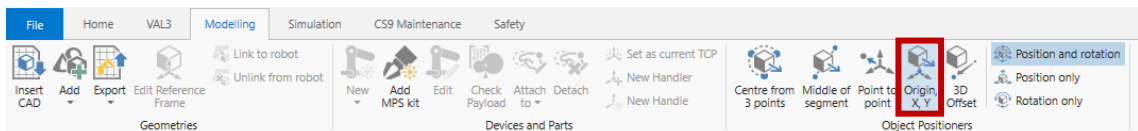
Nyní si vytvoříme vlastní souřadný systém, který bude položen na paletce. Pravým tlačítkem na *Uloha3* a následným výběrem *Add* a *New Data* dostaneme vyskakovací okno, kde zvolíme typ *frame*. Ten pojmenujeme jako *fPaletka*



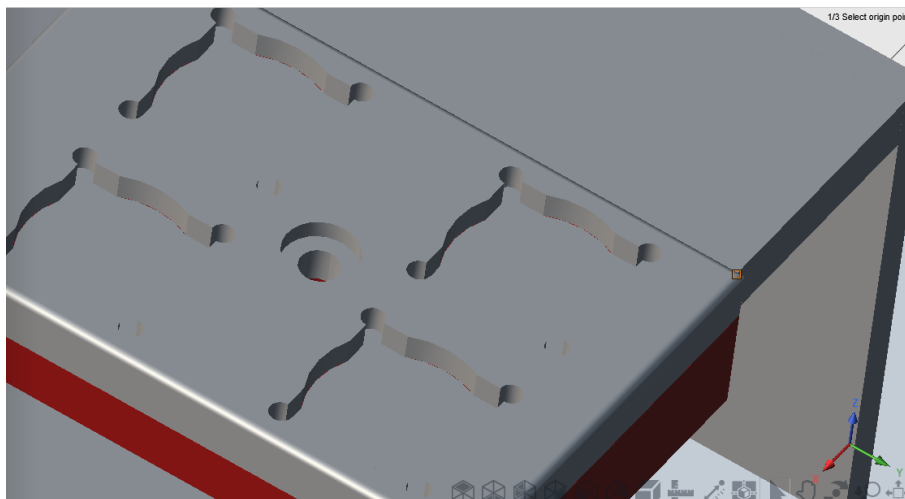
Obrázek 219 Vytvoření
nového souřadného systému



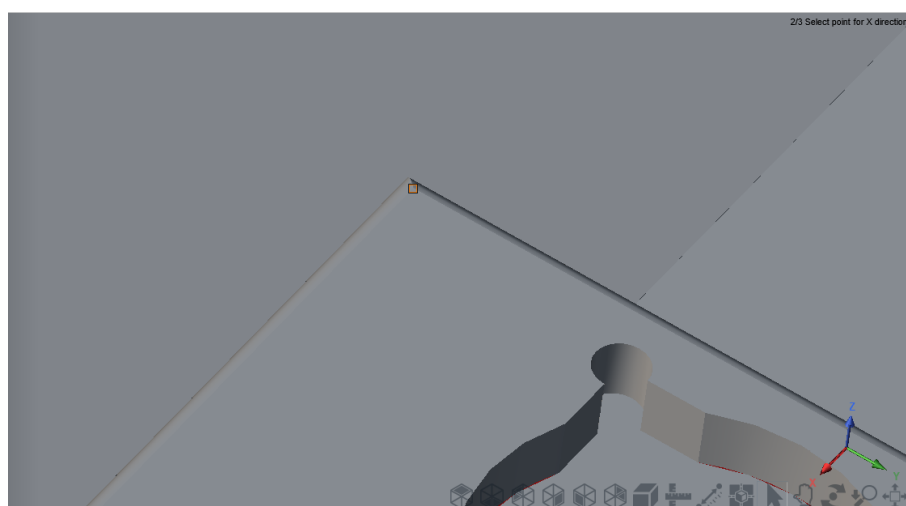
Obrázek 220 Definice souřadného systému



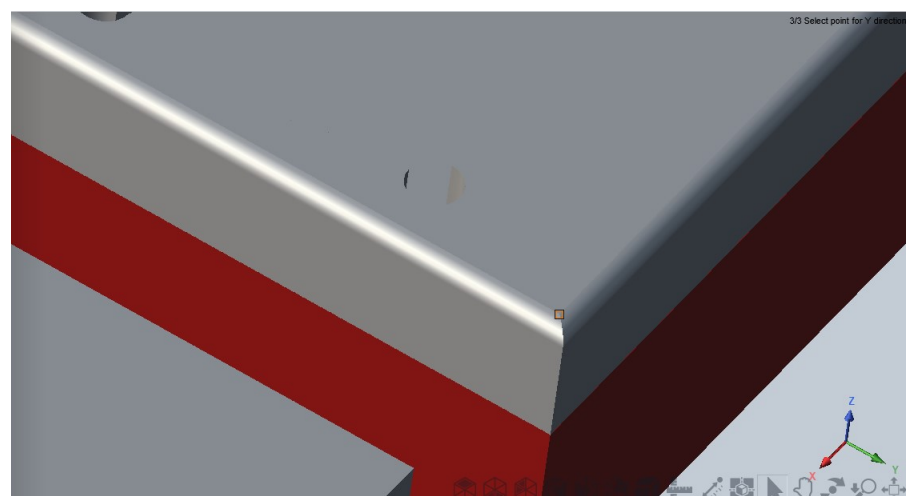
Obrázek 221 Naučení nového souřadného systému přes Origin X,Y



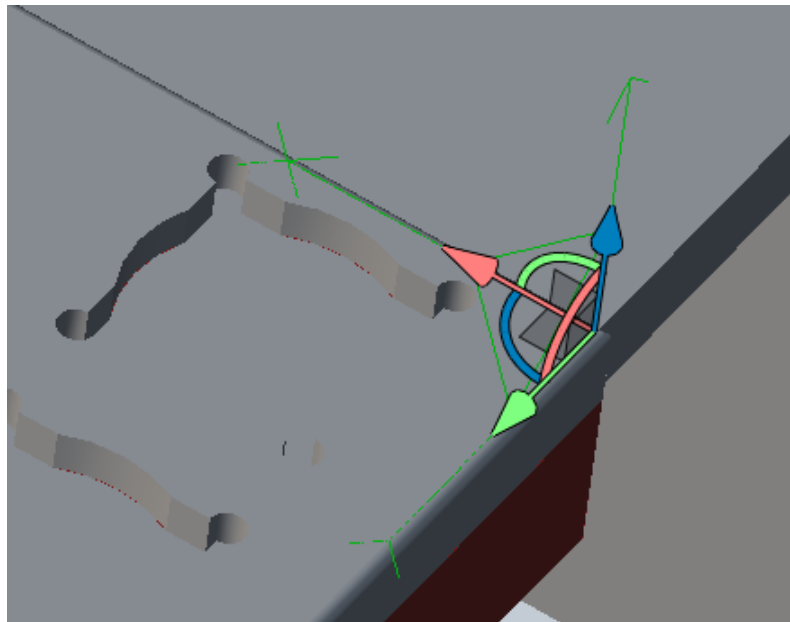
Obrázek 222 Umístění počátku souřadného systému



Obrázek 223 Vybrání bodu na ose X

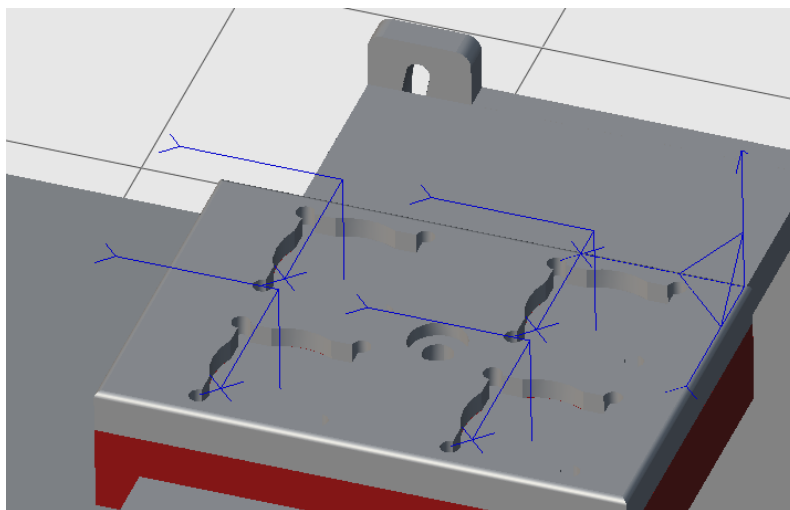


Obrázek 224 Vybrání bodu na ose Y



Obrázek 225 Umístěný souřadný systém

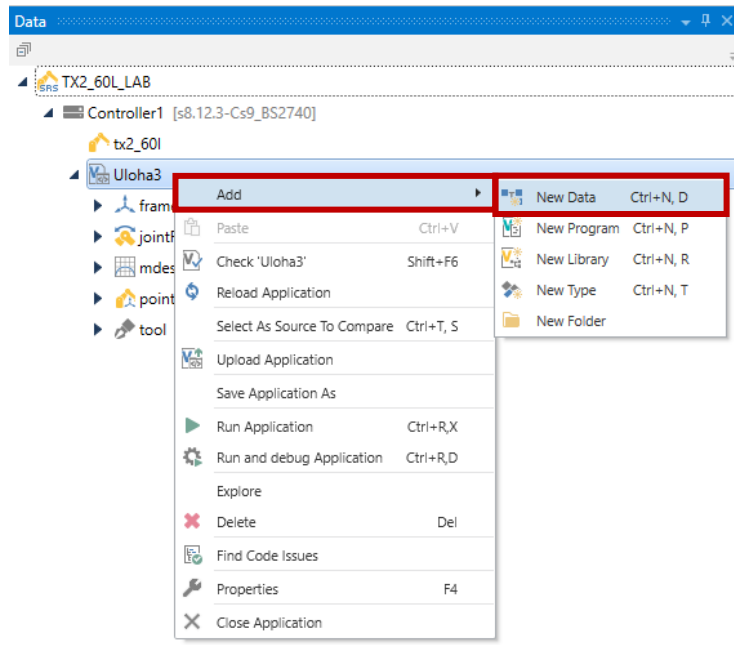
Nyní si v tomto souřadném systému vytvoříme pole bodů *pPlace*. Pole bude typu *pointRx* a bude obsahovat 4 body.



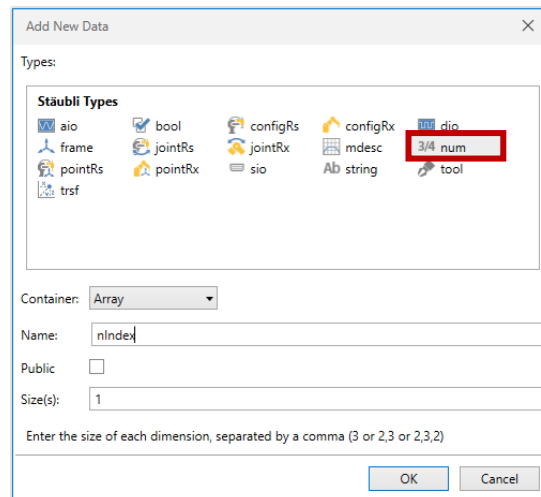
Obrázek 226 Umístěné body pole pPlace

10.3.7 Vytvoření programu

Pravým kliknutím na *Uloha3* a vybráním z listu *Add* a *New Data* dostaneme vyskakující okno, kde zvolíme datový typ *num* a proměnnou pojmenujeme *nIndex*.



Obrázek 227 Vytvoření číselné proměnné

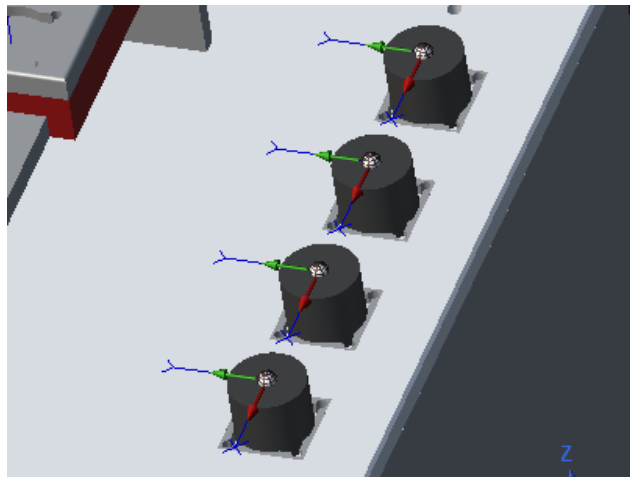


Obrázek 228 Definice nIndex

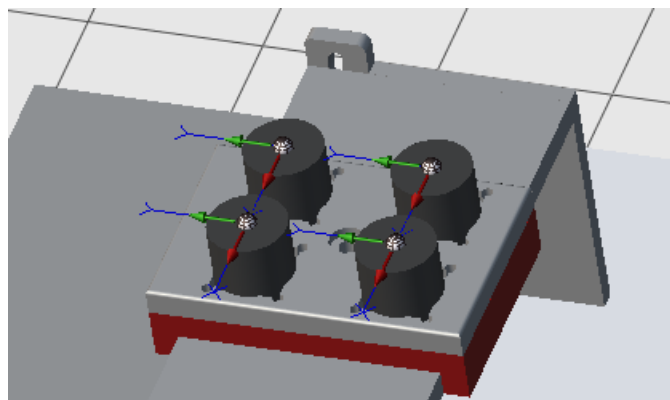
Nyní máme vše co je potřeba pro naprogramování pohybu robota.

```
Uloha3-start 3D View*
1 begin
2   movej(jBezpeci,tNastroj,mNomSpeed)
3   open(tNastroj)
4   for nIndex= 0 to 3
5     //njetí nad váleček + uchopení
6     movej(appro(pPick[nIndex],{0,0,-200,0,0,0}),tNastroj,mNomSpeed)
7     movel(pPick[nIndex],tNastroj,mNomSpeed)
8     waitEndMove()
9     close(tNastroj)
10    movej(appro(pPick[nIndex],{0,0,-200,0,0,0}),tNastroj,mNomSpeed)
11    //njetí nad paletku na policiče
12    movej(appro(pPlace[nIndex],{0,0,-70,0,0,0}),tNastroj,mNomSpeed)
13    movel(pPlace[nIndex],tNastroj,mNomSpeed)
14    waitEndMove()
15    open(tNastroj)
16    movej(appro(pPlace[nIndex],{0,0,-70,0,0,0}),tNastroj,mNomSpeed)
17  endFor
18  movej(jBezpeci,tNastroj,mNomSpeed)
19  waitEndMove()
20 end
```

Obrázek 229 Kompletní program

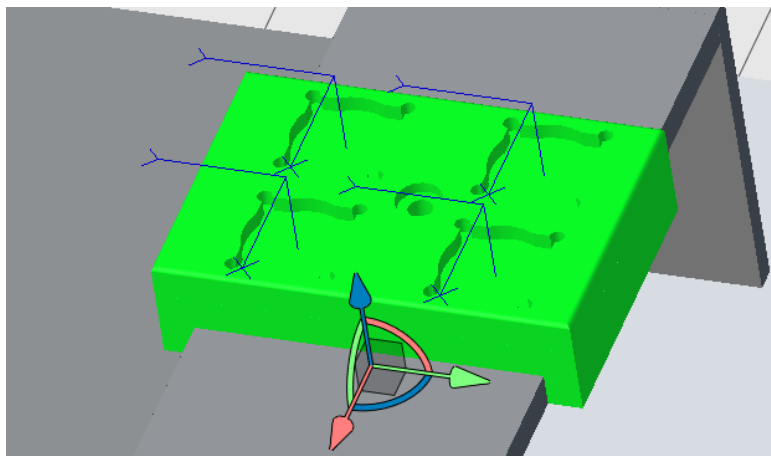


Obrázek 230 Počáteční pozice válečků

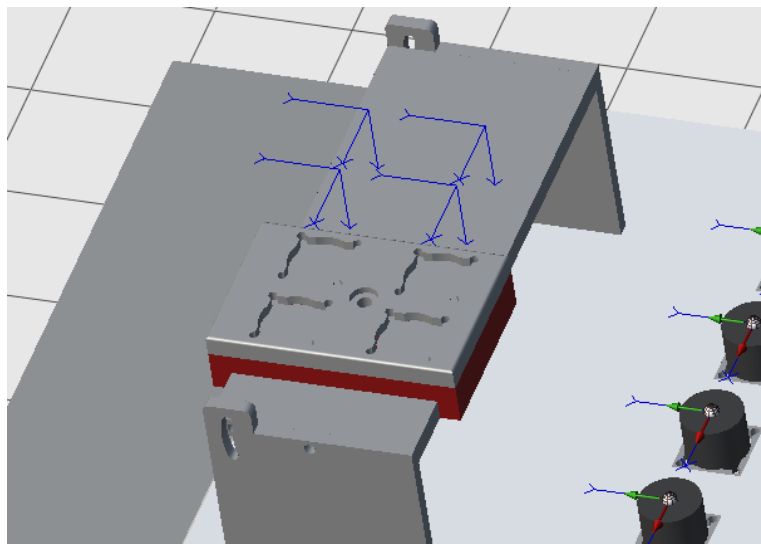


Obrázek 231 Výsledek programu

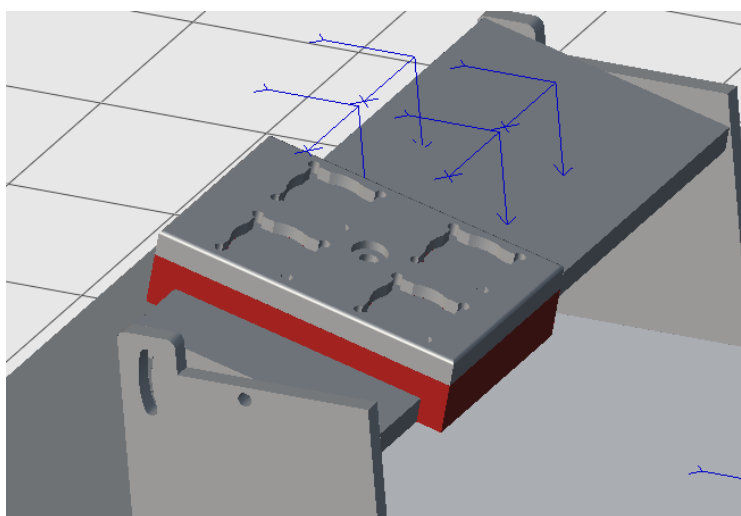
10.3.8 Přesunutí paletky



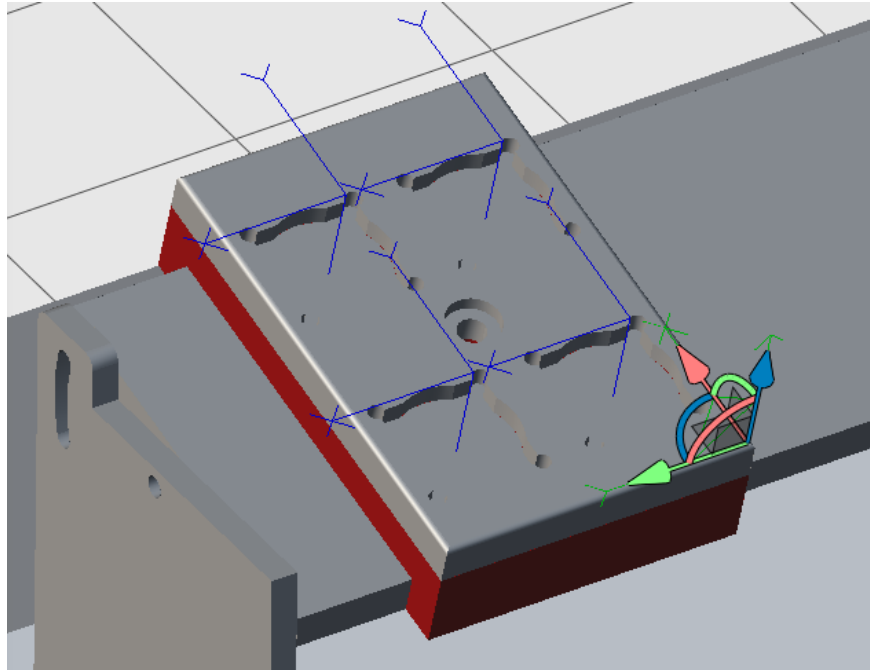
Obrázek 232 Vybraná paletka



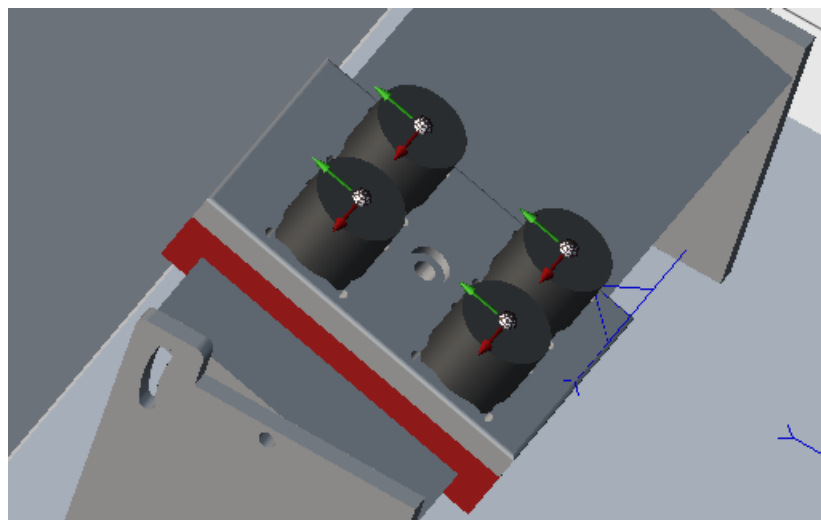
Obrázek 233 Přesunutá paletka



Obrázek 234 Nakloněné komponenty



Obrázek 235 Výsledek přemístění fPaletka



Obrázek 236 Výsledek programu po změnění fPaletka

10.3.9 Převedení na reálného robota

Máme-li odzkoušenou simulaci, tak hlavní program upravíme tak, aby byl funkční pro reálného robota. V této úloze je potřeba pouze funkce nástroje *open(tNastroj)* a *close(tNastroj)* prohodit. To znamená, že místo *open(tNastroj)* bude *close(tNastroj)* a naopak.

```

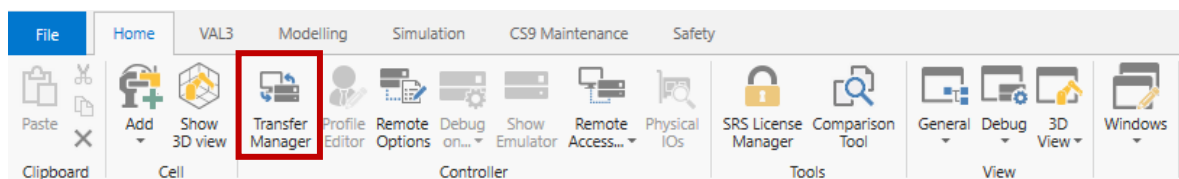
Uloha3-start 3D View*
1 begin
2   movej(jBezpeci,tNastroj,mNomSpeed)
3   close(tNastroj)
4   for nIndex= 0 to 3
5     //njetí nad váleček + uchopení
6     movej(appro(pPick[nIndex],{0,0,-200,0,0}),tNastroj,mNomSpeed)
7     movel(pPick[nIndex],tNastroj,mNomSpeed)
8     waitEndMove()
9     open(tNastroj)
10    movej(appro(pPick[nIndex],{0,0,-200,0,0}),tNastroj,mNomSpeed)
11    //njetí nad paletku na poličce
12    movej(appro(pPlace[nIndex],{0,0,-70,0,0}),tNastroj,mNomSpeed)
13    movel(pPlace[nIndex],tNastroj,mNomSpeed)
14    waitEndMove()
15    close(tNastroj)
16    movej(appro(pPlace[nIndex],{0,0,-70,0,0}),tNastroj,mNomSpeed)
17  endFor
18  movej(jBezpeci,tNastroj,mNomSpeed)
19  waitEndMove()
20 end

```

Obrázek 237 Upravená verze na reál

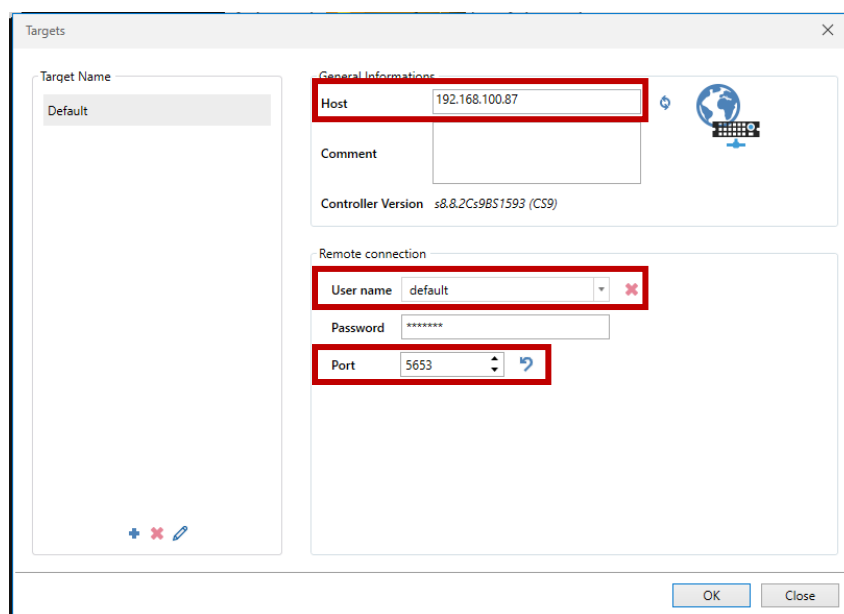
10.3.10 Připojení na reálného robota

Přes *Transfer Manager* v kartě *Home* se připojíme k reálnému robotu.



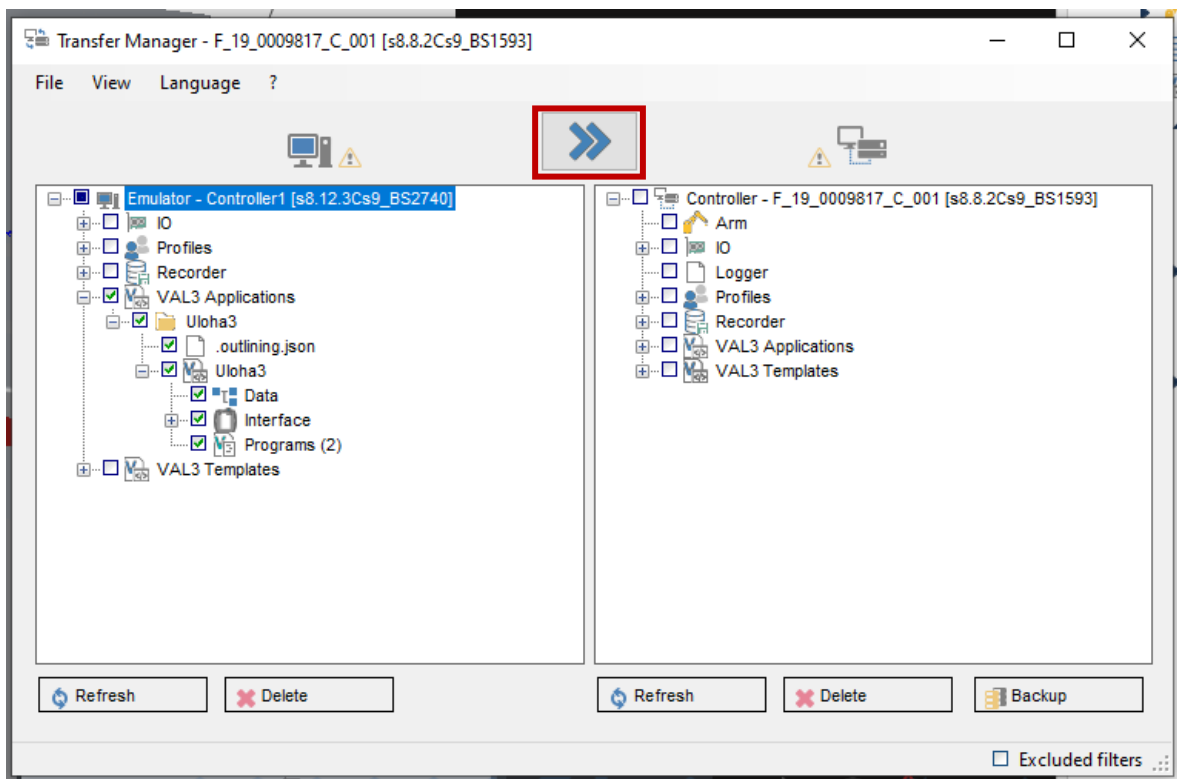
Obrázek 238 Transfer Manager

Objeví se okno, kde si překontrolujeme, že *Host* je **192.168.100.87** a *User name* **Default** a *Port* je **5693**.



Obrázek 239 Tabulka pro připojení k robotu

Pro nahrání aplikace do robota si rozklikneme *VAL3 Applications* a volíme *Uloha2*. Následně klikneme na nahrání a počkáme, než se úloha nahraje.



Obrázek 240 Nahrání aplikace do robota

11 OVĚŘENÍ MANUÁLŮ VYTVOŘENÝCH ÚLOH

Vytvořené manuály pro jednotlivé úlohy byly poskytnuty testerům, kteří si tyto manuály vyzkoušeli, zhodnotili a následně mi byly vyložili jejich případné připomínky a možnosti zlepšení manuálů. Testeři byli pozorováni v průběhu jejich testování a z jejich postupu byly vytvořeny poznámky. V první úloze se nenašel žádný problém ani v simulaci, ani na reálném robotu. V druhé úloze v simulaci také nebyl problém. Ovšem na reálném robotu se problém našel. Pootočený nástroj na reálném robotu zapříčinil vyšší složitost přeučení bodů, ale také při využití funkce *apro* při odkládání válečků, kdy tato funkce nefungovala podle představ. Tudíž v simulaci byly přidány další přejezdové body, které funkci *apro* nahradily. V třetí úloze v simulaci také nebyl. Při převodu na reálné zařízení se však našla chyba v konstrukci pojízdné paletky, která nedisponuje žádným upínacím způsobem. Tudíž při přeučení vlastního souřadného systému nebo bodu na paletce se právě paletka hýbala a znemožnila tím přesné přeučení bodů nebo souřadného systému. Tento problém byl vyřešen změnou návrhu paletky, která již má možnost připevnění k polici. Testeři se k zpracování manuálů vyjádřili kladně a podotkli zvýraznění některých částí manuálu.

ZÁVĚR

Teoretická část byla rozdělena do více bodů. Jako první byla představena obecná robotika, kde bylo rozebráno, čím se robotika zabývá a jak vznikala. Následně byla popsána krátká historie vývoje průmyslové robotiky. Pokračovalo se definicí robota a dále byly uvedeny informace, z čeho se robot skládá. Byly popsány klouby, akční členy, senzory a také řídicí jednotka robota. Dále zde bylo objasněno, co je to mobilní robot a jak se dělí. Ke každé větvi mobilního robota bylo zmíněno pár informací o jeho možnostech pohybu a základních aplikacích. Jako další přišlo na řadu popisování manipulátorů a průmyslových robotů. Bylo rozebráno, co takové průmyslové roboty obsahují. Byly rozebrány druhy smyček, pojezdové ústrojí, polohovací ústrojí, orientační ústrojí, koncový efektor a definice pracovních prostorů. Dále byly zmíněny jednotlivé struktury průmyslových robotů, kde byl také popsán jejich pracovní prostor, ukázka robota s takovou strukturou a také byly zmíněny jednotlivé aplikace dané struktury. Dále v teoretické části byla zmínka o robotu a jeho controlleru, se kterými budeme pracovat v teoretické části. Zmíněny byly především vlastnosti robota. Jako poslední byl vysvětlen základní důvod tvoření úloh pro studenty.

V praktické části byl vytvořen obecný manuál pro práci se *Stäubli Robotics Suite*. Hlavním úkolem praktické části bylo vytvořit úlohy pro studenty, které budou krok po kroku zpracovány do manuálů. Jako první byly vymyšlena orientační zadání, která se upravovala při zpracovávání manuálu tak, aby studentům bylo jasné, co je úkolem dané úlohy a co si z úlohy mohou odnést. První úloha slouží především pro pochopení rozdílu mezi lineárním a rotačním pohybem robota. Také se zde naučí používat cykly a podprogramy, aby se celkový počet řádků programu minimalizoval. V této úloze se studenti také setkají s funkcí *appro*, která slouží k aproximaci pohybu. Druhá úloha je zaměřena na práci s orientací nástroje. Pro tuto úlohu byla navržena a následně zpracována přídatná paleta, která má možnost náklonné poličky a také přimontování hřídele, která se právě používá v druhé úloze. V třetí úloze bylo ukázáno práce s vlastními souřadnými systémy a jejich výhoda při přečtení souřadného systému oproti přečtení všech bodů. Tento manuál byl ověřen studenty, kteří poskytli zpětnou vazbu, díky které se upravily úlohy. V druhé úloze bylo potřeba upravit odkládání na poličku, jelikož souřadný systém nástroje na skutečném robotu je pootočen a pomocí funkce *appro* neplnil program svůj účel. Z toho důvodu zde byly přidány body pro přejezd, které původně měla nahradit funkce *appro*. V třetí úloze se naskytl problém s tím, že pojízdná polička na paletě se nedala upevnit a kvůli tomu se při přečování bodů hýbala. Tento problém byl vyřešen předěláním návrhu této poličky a již je možno pojízdnou poličku připevnit.

SEZNAM POUŽITÉ LITERATURY

- [1] KAPILA, Vikram. Introduction to Robotics. [online] 2013. [cit. 13.05.2022]. Dostupné z: http://engineering.nyu.edu/mechatronics/smart/Archive/intro_to_robot/Intro2Robotics.pdf
- [2] SICILIANO, Bruno et al. *Robotics*. London: Springer London, 2009. Advanced Textbooks in Control and Signal Processing, ED. . ISBN 978-1-84628-641-4. DOI: 10.1007/978-1-84628-642-1
- [3] WILLIAMS, Bob. An Introduction to Robotics. . 2021, s. 46. [cit. 13.05.2022]
- [4] JAZAR, Reza N. *Theory of Applied Robotics*. Boston, MA: Springer US, 2010. ISBN 978-1-4419-1749-2. DOI: 10.1007/978-1-4419-1750-8
- [5] NONAMI, Kenzo et al. *Autonomous Flying Robots*. Tokyo: Springer Japan, 2010. ISBN 978-4-431-53855-4. DOI: 10.1007/978-4-431-53856-1
- [6] 6 axis industrial robots TX2-60 | Staubli. In: *Staubli* [online]. 2022 [cit. 13.05.2022]. Dostupné z: <https://www.staubli.com/tw/en/robotics/products/industrial-robots/tx2-60.html>
- [7] STAUBLI. Rameno - Skupina TX Řada 60. . 2018, s. 90. [cit. 13.05.2022]
- [8] CS9 robot controller. In: *Staubli* [online]. 2022 [cit. 13.05.2022]. Dostupné z: <https://www.staubli.com/global/en/robotics/products/robot-controllers/cs9-robot-controller.html>
- [9] UNIVERSAL ROBOTS. *Types of Grippers used in manufacturing* [online]. 2022. [cit. 13.05.2022]. Dostupné z: <https://www.universal-robots.com/blog/types-of-grippers-used-in-manufacturing/>
- [10] ÚŘEDNÍČEK, Zdeněk. *Konstrukce robotů_1*. Univerzita Tomáše Bati ve Zlíně, Fakulta Aplikované Informatiky, 2021. [cit. 13.05.2022]
- [11] ÚŘEDNÍČEK, Zdeněk. *Konstrukce robotů_2*. Univerzita Tomáše Bati ve Zlíně, Fakulta Aplikované Informatiky, 2021. [cit. 13.05.2022]
- [12] ÚŘEDNÍČEK, Zdeněk. *Konstrukce robotů_4*. Univerzita Tomáše Bati ve Zlíně, Fakulta Aplikované Informatiky, 2021. [cit. 13.05.2022]
- [13] ÚŘEDNÍČEK, Zdeněk. *Základní části robotů*. Univerzita Tomáše Bati ve Zlíně, Fakulta Aplikované Informatiky, 2021. [cit. 13.05.2022]

- [14] ÚŘEDNÍČEK, Zdeněk. *MTS a Robotika*. Univerzita Tomáše Bati ve Zlíně, Fakulta Aplikované Informatiky, 2021.
- [15] ÚŘEDNÍČEK, Zdeněk. *Robotika – krácející roboty*. Univerzita Tomáše Bati ve Zlíně, Fakulta Aplikované Informatiky, 2021.
- [16] AMERICAN ROBOT COMPANY. *GRP 4400* [online]. 2015 [cit. 13.05.2022]. Dostupné z: <http://www.ambot.com/ip-wheel.shtml>
- [17] UNITREE LAIKAGO. *Laikago* [online]. 2017 [cit. 13.05.2022]. Dostupné z: <http://www.unitree.cc/e/action/ShowInfo.php?classid=6&id=1>
- [18] MILITARY.COM. *MQ-9 Reaper* [online]. 2022 [cit. 13.05.2022]. Dostupné z: <https://www.military.com/equipment/mq-9-reaper>
- [19] DJI. *DJI Mavic 2 Pro* [online]. 2022 [cit. 13.05.2022]. Dostupné z: <https://dronpro.cz/dji-mavic-2-pro>
- [20] ANONYMOUS. *Robotic airship* [online]. 2022 [cit. 13.05.2022]. Dostupné z: https://www.researchgate.net/figure/Robotic-airship-performing-its-first-flight_fig1_334478919
- [21] TECHNICAL UNIVERSITY OF DELFT. *DelFly* [online]. 2022 [cit. 13.05.2022]. Dostupné z: https://www.researchgate.net/figure/Flapping-wings-civilian-UAV-a-Technical-University-of-Delfts-DelFly-b-FESTOs_fig12_278745381
- [22] NASA. *Mars Rover* [online]. 2019 [cit. 13.05.2022]. Dostupné z: <https://mars.nasa.gov/mer/>
- [23] STANFORD RACING REAM. *Stanley* [online]. 2006 [cit. 13.05.2022]. Dostupné z: <https://robotika.cz/competitions/grandchallenge/2005/cs>
- [24] TTHK. *Robot end of arm tooling* [online] [cit. 13.05.2022]. Dostupné z: <https://www.tthk.ee/inlearcs/7-robot-end-of-arm-tooling/>
- [25] ROBOTIQ. *AIRPICK VACUUM GRIPPER* [online]. 2020 [cit. 13.05.2022]. Dostupné z: <https://www.amtech-robotics.cz/roboty/vyrobciprislusenstviur/robotiq/airpick-vacuum-gripper/>
- [26] FESTO. *Gripper* [online] [cit. 13.05.2022]. Dostupné z: <http://pneumaticshydraulic.com/gripper/>

- [27] ZIMMER GROUP. *Zimmer Group GEP2000* [online]. 2022 [cit. 13.05.2022]. Dostupné z: <https://www.zimmer-group.com/en/technologies-components/components/handling-technology/grippers/electric/2-jaw-parallel-grippers/series-gep2000>
- [28] STÄUBLI. *Stäubli TS2-60* [online]. 2022 [cit. 13.05.2022]. Dostupné z: <https://www.staubli.com/global/en/robotics/products/industrial-robots/ts2-60.html>
- [29] ABB. *ABB IRB 2400* [online] [cit. 13.05.2022]. Dostupné z: <https://www.campetella.com/en/industrial-robots/>
- [30] GMF. *GMF M100* [online]. 2007 [cit. 13.05.2022]. Dostupné z: https://www.industrial-electronics.com/industrial-electricity-com/4_Cylindrical_Robots.html
- [31] ANONYMOUS. *Gantry robot* [online]. 2022 [cit. 13.05.2022]. Dostupné z: <https://www.machinedesign.com/mechanical-motion-systems/article/21831692/the-difference-between-cartesian-sixaxis-and-scara-robots>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

mm milimetr

SEZNAM OBRÁZKŮ

| | |
|--|----|
| Obrázek 1 Nejběžněji používané klouby [4] | 13 |
| Obrázek 2 Komponenty robotického systému [2] | 15 |
| Obrázek 3 Typy kol kolových robotů [2] | 16 |
| Obrázek 4 Kolový robot – AMBOT GRP 4400 [16]..... | 17 |
| Obrázek 5 Kráčející robot – Laikago [17] | 17 |
| Obrázek 6 UAV s pevnými křídly - MQ-9 Reaper [18]..... | 18 |
| Obrázek 7 UAV s rotačním křídlem – DJI Mavic 2 Pro [19]..... | 18 |
| Obrázek 8 Robotická vzducholoď [20] | 19 |
| Obrázek 9 UAV s mávajícími křídly – DelFly MAV [21]..... | 19 |
| Obrázek 10 Polní robot - Mars Rover [22]..... | 20 |
| Obrázek 11 Servisní robot – Stanley [23] | 21 |
| Obrázek 12 Paralelní manipulátor [2] | 23 |
| Obrázek 13 Hybridní manipulátor [2]..... | 24 |
| Obrázek 14 Schématická ukázka 3DOF zápěstí [24]..... | 25 |
| Obrázek 15 Vakuové chapadlo od firmy ROBOTIQ [25] | 26 |
| Obrázek 16 Pneumatická chapadla [26] | 27 |
| Obrázek 17 Elektrický gripper Zimmer Group GEP2000 [27] | 28 |
| Obrázek 18 Znázornění pracovního prostoru [1]..... | 29 |
| Obrázek 19 Pracovní prostor SCARA robota [2] | 30 |
| Obrázek 20 SCARA robot – Stäubli TS2-60 [28] | 30 |
| Obrázek 21 Pracovní prostor antropomorfního robota [2] | 31 |
| Obrázek 22 Antropomorfní robot – ABB IRB 2400 [29] | 31 |
| Obrázek 23 Pracovní prostor sférického robota [2] | 32 |
| Obrázek 24 Pracovní prostor cylindrického robota [2]..... | 33 |
| Obrázek 25 Cylindrický robot – GMF M100 [30]..... | 33 |
| Obrázek 26 Pracovní prostor kartézského robota – přístup z boku [2]..... | 34 |
| Obrázek 27 Schéma kartézského (portálového) robota – přístup z vrchu [2] | 34 |
| Obrázek 28 Portálový robot [31] | 35 |
| Obrázek 29 Stäubli TX2-60L [6]..... | 36 |
| Obrázek 30 Popis části robota Stäubli TX2-6L [7]..... | 37 |
| Obrázek 31 Controller CS9 [7][8]..... | 38 |
| Obrázek 32 Pendant [7]..... | 39 |

| | |
|--|----|
| Obrázek 33 Ovládací panel Controlleru [7]..... | 39 |
| Obrázek 34 Vytvoření nového projektu | 42 |
| Obrázek 35 Pojmenování nového projektu..... | 42 |
| Obrázek 36 Přidání controlleru | 43 |
| Obrázek 37 Volení správného typu robota | 43 |
| Obrázek 38 Rozšířené nastavení controlleru | 44 |
| Obrázek 39 Kontrola navoleného projektu..... | 44 |
| Obrázek 40 Zapnutí 3D vizualizace | 45 |
| Obrázek 41 Znázornění hlavních ovládacích prvků..... | 46 |
| Obrázek 42 Karta VAL3..... | 46 |
| Obrázek 43 Karta Modelling | 46 |
| Obrázek 44 Karta Simulation..... | 47 |
| Obrázek 45 Vytvoření nové aplikace VAL3 | 47 |
| Obrázek 46 Pojmenování nové aplikace | 48 |
| Obrázek 47 Nová aplikace – karta Cell Explorer | 48 |
| Obrázek 48 Nová aplikace – karta Data | 48 |
| Obrázek 49 Nová aplikace – karta Geometry | 49 |
| Obrázek 50 Vytvoření nového Cylinderu..... | 49 |
| Obrázek 51 Vytvořený objekt – Cylinder..... | 50 |
| Obrázek 52 Vlastnosti Cylinderu | 50 |
| Obrázek 53 Lišta možnost objektu - Edit Position..... | 51 |
| Obrázek 54 Okno editace pozice objektů | 52 |
| Obrázek 55 Posunutý válec ve 3D vizualizaci..... | 52 |
| Obrázek 56 Posunutí válce – okno Edit Position | 53 |
| Obrázek 57 Převedení objektu na nástroj | 53 |
| Obrázek 58 Strom vizualizace | 54 |
| Obrázek 59 Editace nástroje – nastavení handle..... | 54 |
| Obrázek 60 Zapnutí Edit Position | 55 |
| Obrázek 61 Správná orientace handle | 55 |
| Obrázek 62 Připevnění nástroje k robotu | 55 |
| Obrázek 63 Nástroj připevněný k robotu | 55 |
| Obrázek 64 Vytvoření nových dat do položky flange..... | 56 |
| Obrázek 65 Tabulka pro vytvoření nástroje | 56 |
| Obrázek 66 Přidání chování nástroje..... | 57 |

| | |
|--|----|
| Obrázek 67 Otevření konfigurace nástroje | 57 |
| Obrázek 68 Tabulka konfigurace | 57 |
| Obrázek 69 Poslední okno konfigurace nástroje..... | 58 |
| Obrázek 70 Nastavení rozměrů a ovládání nástroje..... | 58 |
| Obrázek 71 Vytvořený Part | 59 |
| Obrázek 72 Využití možnosti vložení CAD objektu | 59 |
| Obrázek 73 Tabulka pro vložení CAD objektu | 59 |
| Obrázek 74 Vložený CADovský objekt | 60 |
| Obrázek 75 Označený komponent a volba uvolnění komponentu ze sestavy | 60 |
| Obrázek 76 Volba zobrazení pracovního prostoru..... | 61 |
| Obrázek 77 Zobrazení pracovního prostoru robota | 61 |
| Obrázek 78 Vytvoření nového bodu v jointRx | 62 |
| Obrázek 79 Tabulka pro vytvoření nového bodu..... | 62 |
| Obrázek 80 Naučení bodu jBezpeci | 62 |
| Obrázek 81 Vytvoření pole bodů | 63 |
| Obrázek 82 Přidání nové hodnoty do pole bodů..... | 63 |
| Obrázek 83 Zvolení bodu ve 3D vizualizaci | 64 |
| Obrázek 84 Výběr možnosti přesunutí bodu | 64 |
| Obrázek 85 Správně naorientovaný bod na rohu kvádru | 64 |
| Obrázek 86 Ověření dostupnosti bodu koncovým efektozem | 65 |
| Obrázek 87 Vytvoření nového framu – přidání nových dat | 65 |
| Obrázek 88 Tabulka vytvoření nového framu | 66 |
| Obrázek 89 Vytvořený frame ve stromu dat..... | 66 |
| Obrázek 90 Nadefinovaný souřadný systém fKvadr..... | 67 |
| Obrázek 91 Frame s vlastními body..... | 67 |
| Obrázek 92 Otočený kvádr | 68 |
| Obrázek 93 Přenaučený frame na otočeném rámu..... | 68 |
| Obrázek 94 Programovací okno..... | 69 |
| Obrázek 95 Možnosti úpravy programovacího okna | 69 |
| Obrázek 96 Vytvoření nové proměnné..... | 70 |
| Obrázek 97 Naprogramovaný pohyb robota..... | 70 |
| Obrázek 98 Zapnutí synchronizace prostředí a programu..... | 70 |
| Obrázek 99 Spuštění aplikace | 71 |
| Obrázek 100 Emulátor TeachPendantu v SRS | 72 |

| | |
|--|----|
| Obrázek 101 VAL3 obrazovka | 72 |
| Obrázek 102 Založení podprogramu..... | 73 |
| Obrázek 103 Pojmenování podprogramu | 73 |
| Obrázek 104 Struktura kódu podprogramu | 74 |
| Obrázek 105 Struktura kódu startu..... | 74 |
| Obrázek 106 Převedení dat na Public..... | 74 |
| Obrázek 107 Vytvoření nové aplikace | 75 |
| Obrázek 108 Pojmenování nové aplikace | 75 |
| Obrázek 109 Vytvoření nové knihovny | 76 |
| Obrázek 110 Pojmenování a vybrání nové knihovny | 76 |
| Obrázek 111 Obsah knihovny..... | 76 |
| Obrázek 112 Presun robotu na bod z knihovny | 77 |
| Obrázek 113 Výsledek volání pohybu na bod z knihovny | 77 |
| Obrázek 114 Převedení Podprogram na public | 77 |
| Obrázek 115 Volání podprogramu z knihovny..... | 78 |
| Obrázek 116 Vytvoření uživatelského rozhraní..... | 78 |
| Obrázek 117 Definice obrazovky..... | 79 |
| Obrázek 118 Toolbox a obrazovka editace UI | 79 |
| Obrázek 119 Vložený Button..... | 80 |
| Obrázek 120 Vlastnosti tlačítka | 80 |
| Obrázek 121 Prvek Label | 81 |
| Obrázek 122 Vlastnosti prvku Label..... | 81 |
| Obrázek 123 Prvek List | 81 |
| Obrázek 124 Vlastnosti prvku List..... | 82 |
| Obrázek 125 Linknutí listu k proměnné..... | 82 |
| Obrázek 126 Přidání nových možností výběru | 83 |
| Obrázek 127 Prvek Text..... | 83 |
| Obrázek 128 Vlastnosti prvku Text | 83 |
| Obrázek 129 Volání UI | 84 |
| Obrázek 130 Vzhled UI..... | 84 |
| Obrázek 131 Vložení CAD objektu - váleček | 85 |
| Obrázek 132 Vložený váleček | 86 |
| Obrázek 133 Výběr metody přemístění válečku..... | 86 |
| Obrázek 134 Umístění válečku do drážky..... | 86 |

| | |
|---|----|
| Obrázek 135 Tabulka Edit Position | 86 |
| Obrázek 136 Převod válečku na součástku | 87 |
| Obrázek 137 Přemístění handlu | 87 |
| Obrázek 138 Správné umístění Handlu | 87 |
| Obrázek 139 Vložení nového objektu | 88 |
| Obrázek 140 Převod Cylinderu na nástroj | 88 |
| Obrázek 141 Strom scény a zvolení handlu | 89 |
| Obrázek 142 Zvolení možnosti pro umístění handlu | 89 |
| Obrázek 143 Správně umístěný handle | 89 |
| Obrázek 144 Nastavení nástroje | 90 |
| Obrázek 145 Bližší nastavení | 90 |
| Obrázek 146 Připnutí nástroje k robotu | 90 |
| Obrázek 147 Vytvoření nové proměnné nástroje | 91 |
| Obrázek 148 Pojmenování proměnné pro nástroj | 91 |
| Obrázek 149 Data nástroje | 91 |
| Obrázek 150 Vytvoření nového jointového bodu | 92 |
| Obrázek 151 Volba datového typu a pojmenování jointového bodu | 92 |
| Obrázek 152 Napolohovaný robot pro naučení jBezpeci | 93 |
| Obrázek 153 Naučení bodu jBezpeci | 93 |
| Obrázek 154 Správně napolohované body | 93 |
| Obrázek 155 Vytvoření podprogramu | 94 |
| Obrázek 156 Pojmenování podprogramu | 94 |
| Obrázek 157 Část naprogramovaného kódu | 95 |
| Obrázek 158 Volání verzeA | 95 |
| Obrázek 159 Kompletní program verzeA | 95 |
| Obrázek 160 Výsledek kompletního programu verzeA | 96 |
| Obrázek 161 Hlavní program a volání verzeB | 96 |
| Obrázek 162 Kompletní program verzeB | 96 |
| Obrázek 163 Kompletní program verzeC | 97 |
| Obrázek 164 Volání verzeC | 97 |
| Obrázek 165 Upravený program | 98 |
| Obrázek 166 Transefer Manager | 98 |
| Obrázek 167 Tabulka pro připojení k robotu | 98 |
| Obrázek 168 Nahrání aplikace do robotu | 99 |

| | |
|---|-----|
| Obrázek 169 Vložení válečků..... | 99 |
| Obrázek 170 Výběr možnosti přesunu objektu..... | 100 |
| Obrázek 171 Našeptávač umístění..... | 100 |
| Obrázek 172 3 válečky umístěné v drážkách..... | 100 |
| Obrázek 173 Převedení objektu na součástku..... | 100 |
| Obrázek 174 Převedení objektu na součástku..... | 101 |
| Obrázek 175 Správně umístěný handle..... | 101 |
| Obrázek 176 Převedení objektu na nástroj..... | 102 |
| Obrázek 177 Připevnění nástroje k robotu..... | 102 |
| Obrázek 178 Kontrola orientace nástroje..... | 102 |
| Obrázek 179 Přidání nového chování nástroje..... | 103 |
| Obrázek 180 Nastavení chování..... | 103 |
| Obrázek 181 Vytvoření nového pole..... | 104 |
| Obrázek 182 Definice pole pPick..... | 104 |
| Obrázek 183 Výběr bodu pPick[0] do 3D vizualizace..... | 105 |
| Obrázek 184 Přesunutí bodu pomocí Object Positioners..... | 105 |
| Obrázek 185 Správně umístěný bod pPick[0]..... | 105 |
| Obrázek 186 Vytvoření číselné proměnné..... | 106 |
| Obrázek 187 Definice nIndex..... | 106 |
| Obrázek 188 Výsledek první části programu..... | 107 |
| Obrázek 189 Kompletní program..... | 107 |
| Obrázek 190 Výsledek programu..... | 108 |
| Obrázek 191 Upravený program na reálného robota..... | 108 |
| Obrázek 192 Transefer Manager..... | 109 |
| Obrázek 193 Tabulka pro připojení k robotu..... | 109 |
| Obrázek 194 Nahrání aplikace do robotu..... | 109 |
| Obrázek 195 Výběr možnosti Insert CAD..... | 110 |
| Obrázek 196 Zvolení množnosti přesunu válečku..... | 110 |
| Obrázek 197 Našeptávač označení hrany..... | 110 |
| Obrázek 198 Umístěný váleček..... | 111 |
| Obrázek 199 4 válečky na hlavní desce..... | 111 |
| Obrázek 200 Převedení válečku na součást..... | 112 |
| Obrázek 201 Umístěný handle..... | 112 |
| Obrázek 202 4 součástky..... | 113 |

| | |
|--|-----|
| Obrázek 203 Vložení nástroje..... | 113 |
| Obrázek 204 Zvolení souboru pro vložení | 113 |
| Obrázek 205 Převedení objektu na nástroj | 114 |
| Obrázek 206 Umístěný handle nástroje..... | 114 |
| Obrázek 207 Výběr možnosti přesunu handleru..... | 114 |
| Obrázek 208 Správně umístěný handler..... | 115 |
| Obrázek 209 Připevnění nástroje k robotu | 115 |
| Obrázek 210 Připevněný nástroj | 115 |
| Obrázek 211 Přidání chování nástroje..... | 115 |
| Obrázek 212 Nastavení cesty k fyzickému zařízení..... | 116 |
| Obrázek 213 Nastavení chování nástroje | 116 |
| Obrázek 214 Vytvoření nového pole bodů..... | 117 |
| Obrázek 215 Definice pole pPick | 117 |
| Obrázek 216 Výběr možnosti přesunutí bodu | 117 |
| Obrázek 217 Správně umístěný bod pPick[0] | 118 |
| Obrázek 218 Umístění bodů z pole pPick | 118 |
| Obrázek 219 Vytvoření nového souřadného systému..... | 119 |
| Obrázek 220 Definice souřadného systému | 119 |
| Obrázek 221 Naučení nového souřadného systému přes Origin X,Y..... | 119 |
| Obrázek 222 Umístění počátku souřadného systému | 120 |
| Obrázek 223 Vybrání bodu na ose X | 120 |
| Obrázek 224 Vybrání bodu na ose Y | 120 |
| Obrázek 225 Umístěný souřadný systém | 121 |
| Obrázek 226 Umístěné body pole pPlace..... | 121 |
| Obrázek 227 Vytvoření číselné proměnné | 122 |
| Obrázek 228 Definice nIndex | 122 |
| Obrázek 229 Kompletní program..... | 123 |
| Obrázek 230 Počáteční pozice válečků | 123 |
| Obrázek 231 Výsledek programu..... | 123 |
| Obrázek 232 Vybraná paletka..... | 124 |
| Obrázek 233 Přesunutá paletka..... | 124 |
| Obrázek 234 Nakloněné komponenty | 124 |
| Obrázek 235 Výsledek přemístění fPaletka..... | 125 |
| Obrázek 236 Výsledek programu po změně fPaletka | 125 |

| | |
|---|-----|
| Obrázek 237 Upravená verze na reál | 126 |
| Obrázek 238 Transfer Manager | 126 |
| Obrázek 239 Tabulka pro připojení k robotu..... | 126 |
| Obrázek 240 Nahrání aplikace do robotu..... | 127 |

SEZNAM PŘÍLOH

MANUAL_Uloha1.pdf

MANUAL_Uloha2.pdf

MANUAL_Uloha3.pdf

Složka: **Modely**

- **Deska_S_Polickou.step**
- **Gripper_2prst.step**
- **Padestal.stl**
- **PaletizacniDeska.stl**
- **Valcek_45x40_mm.stl**
- **Valecek_Stred.stl**

Složka s výchozím projektem: **TX2_60L_LAB**