



TALLINNA TEHNIKAÜLIKOO

INSENERITEADUSKOND

Elektroenergeetika ja mehhatroonika instituut

# JUHENDID TÖÖKS TÖÖSTUSAUTOMAATIKA 3D MODELLEERIMISE PAKETIGA

GUIDES FOR WORKING WITH INDUSTRIAL AUTOMATION 3D MODELLING  
PACKAGE

BAKALAUREUSETÖÖ

Üliõpilane: Siim Kallisaar

Üliõpilaskood: 165009AAAB

Juhendaja: Madis Lehtla, dotsent

Tallinn, 2020

(Tiitellehe pöördel)

## AUTORIDEKLARATSIOON

Olen koostanud lõputöö iseseisvalt.

Lõputöö alusel ei ole varem kutse- või teaduskraadi või inseneridiplomit taotletud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on viidatud.

“.....” ..... 2020

Autor: .....

/ allkiri /

Töö vastab bakalaureusetöö/magistritööle esitatud nõuetele

“.....” ..... 2020

Juhendaja: .....

/ allkiri /

Kaitsmisele lubatud

“.....” .....2020 .

Kaitsmiskomisjoni esimees .....

/ nimi ja allkiri /

# LÕPUTÖÖ LÜHIKOKKUVÕTE

<i>Autor:</i> Siim Kallisaar	<i>Lõputöö liik:</i> Bakalaureusetöö
<i>Töö pealkiri:</i> Juhendid tööks tööstusautomaatika 3D modelleerimise paketiga	
<i>Kuupäev:</i> 18.05.2020	<i>46 lk (lõputöö lehekülgede arv koos lisadega)</i>
<i>Ülikool:</i> Tallinna Tehnikaülikool	
<i>Teaduskond:</i> Inseneriteaduskond	
<i>Instituut:</i> Elektroenergeetika ja mehhatroonika instituut	
<i>Töö juhendaja(d):</i> dotsent Madis Lehtla	
<i>Töö konsultant (konsultandid):</i> Margus Määr, Vahur Maask, Tarmo Korõtko	
<i>Sisu kirjeldus:</i>  Eesmärgiks oli koostada lühijuhendid algajaile süsteemide (riistvara ja tarkvara) modelleerimiseks koos konkreetsete näidetega sammude kaupa. Töö koosneb juhenditest koos lühinäidetega tööstusroboti, automaatikasüsteemi 3D mudeli juhtimiseks programmeeritavate kontrollritega. Töö annab lühiülevaate 3D-mudeli juhtimisest tööstusroboti mudeliga, tarkvaralise programmeeritava loogikakontrolleri sidevõimalustest OPC-protokolliga ja füüsilise programmeeritava kontrolleri sidevõimalustest MODBUS-protokolliga.	
<i>Märksõnad:</i> juhend, automatiseerimine, 3D-modelleerimine, tööstusrobot, tööstuskontroller, virtuaalreaalsus, imitatsioonimudel, digitaalne kaksik, visualiseerimine, andmeside, kasutajaliides.	

## ABSTRACT

<i>Author:</i> Siim Kallisaar	<i>Type of the work:</i> Bachelor
<i>Title:</i> Guides for working with Industrial Automation 3D Modelling Package	
<i>Date:</i> 18.05.2020	46 pages (the number of thesis pages including appendices)
<i>University:</i> Tallinn University of Technology	
<i>School:</i> School of Engineering	
<i>Department:</i> Department of Electrical Power Engineering and Mechatronics	
<i>Supervisor(s) of the thesis:</i> Associate Professor Madis Lehtla	
<i>Consultant(s):</i> Margus Müür, Vahur Maask, Tarmo Korõtko	
<i>Abstract:</i> <p>The purpose of this thesis was to put together step-by-step beginner guides for modelling hardware and software. The thesis consists of guides with brief examples of controlling a 3D model industrial robot automation system with programmable logic controllers. The thesis gives a brief overview of controlling 3D industrial robot models, software programmable logic controller communication capabilities with OPC protocol and communication capabilities with a physical programmable controller with MODBUS TCP (2) protocol.</p>	
<i>Keywords:</i> manual, robot, robotics, controller, modelling, automation, virtual reality, user interface, imitation model, visualization.	

# LÕPUTÖÖ ÜLESANNE

Lõputöö teema:	<b>Juhendid tööks tööstusautomaatika 3D modelleerimise paketiga</b>
Lõputöö teema inglise keeles:	<b>Guides for working with Industrial Automation 3D Modelling Package</b>
Üliõpilane:	<b>Siim Kallisaar, 165009AAAB</b>
Eriala:	<b>Elektrotehnika AAAB 02/09</b>
Lõputöö liik:	<b>bakalaureusetöö</b>
Lõputöö juhendaja:	<b>Madis Lehtla</b>
Lõputöö ülesande kehtivusaeg:	<b>2019/2020 kevadsemester</b>
Lõputöö esitamise tähtaeg:	<b>19.05.2020 kell 15:00</b>

---

Üliõpilane (allkiri)

---

Juhendaja (allkiri)

---

Õppekava juht (allkiri)

## 1. Teema põhjendus

Virtuaalreaalsus on tööstusautomaatikas üha enam kasutusel. Digitaalseid „kaksikuid“ luuakse erineval otstarbel – tarkvara katsetamiseks arendusprotsessis kui seda pole otstarbekas teha füüsilisel objektil, süsteemi töö visualiseerimiseks jms. Tallinna Tehnikaülikooli tootmise automatiseerimise ja robotitehnika laboratooriumites on olemas nii mitmeid tööstusautomaatika modelleerimisel kasutatavaid tarkvarapakette (sh Ciroso Studio ja ABB Robotstudio). Samad tarkvarapaketid on kasutatavad ka reaalsete tööstuses kasutatavate tootmiseseadmete modelleerimisel. Pakettide rakendamist praktikas takistab sageli asjakohaste või ajakohaste juhendite puudumine.

## 2. Töö eesmärk

Töö eesmärgiks on koostada lühijuhendid algajaile süsteemide (riistvara ja tarkvara) modelleerimiseks koos konkreetsete näidetega sammude kaupa.

## 3. Lahendamisele kuuluvate küsimuste loetelu:

- 1) Robotjaama komponentide 3D modelleerimine keskkonnas Ciroso Studio
- 2) Robotjaama komponentide juhtimise näide simulatsioonikeskkonna signaalide abil (nt haaratsi avamine, detaili tõstmine vms)
- 3) Tööstuskontrollerite tarkvara töö modelleerimine keskkonnas Ciroso Studio
- 4) Füüsilise tööstuskontrolleri liidestamine pakettis Ciroso Studio töötava simulatsiooniga

#### **4. Lähteandmed**

Eesmärkide lahendamiseks plaanin kasutada:

- 1) Robotitehnika õpetamise meetodid ja vahendid Mitsubishi ja Festo seadmete näitel / Margus Müür, Tallinna Tehnikaülikool, 2009
- 2) Varasema paketi „COSIMIR Professional“ juhendid
- 3) Ülesanded (tarkvara vanemale versioonile)
- 4) Komplekt varasemaid juhendeid ja ülesandeid õppejõult ZIP-failina

#### **5. Uurimismeetodid**

Selle eesmärgi saavutamiseks kasutan järgmisi uurimismeetodeid:

- 1) Ülevaate koostamine süsteemi komponentide modelleerimisest,
- 2) Juhendite koostamine (roboti, tööstuskontrolleri ja visualiseerimise osas),
- 3) Algajale vajaliku video salvestamine simulatsioonikeskkonnast ja selle varustamine eesti ning inglise keelsete subtiitritega.

#### **6. Graafiline osa**

Graafiline osa sisaldab: tarkvara plokkiskeeme, 3D mudelite vaateid, programmitekste, elektriskeeme.

#### **7. Töö struktuur**

Loetelu töö peatükkidest koos alapeatükkide, punktide ja alapunktidega.

- Sissejuhatus ja ülevaade
- Modelleerimine
- Katsetulemused
- Kokkuvõte
- Lisa: lühijuhendid sammude kaupa (4 tk) (sama asja kohta salvestada ka video)

#### **8. Kasutatud kirjanduse allikad**

Plaanin kasutada järgmisi allikaid: varem tehtud lõputöid, teadusartikleid, aruandeid, kasutusjuhendeid, raamatuid.

#### **9. Lõputöö konsultandid**

Tööl paketiga Ciro studio on vajadusel lisaks juhendajale kavas konsulteerida Tallinna Tehnikaülikooli töötajate Margus Müüri ja Vahur Maaskiga.

## 10. Töö etapid ja ajakava

Loetelu töö etappidest, mille taga tähtaeg.

Lähteandmete kogumine	06.03.2020
Simulatsioonitarkvara ja selle funktsioonide uurimine	09.03.2020 - 27.03.2020
Töötava mudeli loomine	27.03.2020 - 15.04.2020
Juhendi koostamine (koos algajale sobiliku lühivideoga) roboti modelleerimise näitel	15.04.2020 - 06.05.2020
Juhendi koostamine (koos algajale sobiliku lühivideoga) robotjaama komponentide 3D modelleerimise näitel	15.04.2020 - 06.05.2020
Juhendi koostamine (koos algajale sobiliku lühivideoga) tööstuskontrolleri tarkvara simuleerimise näitel	15.04.2020 - 06.05.2020
Juhendi koostamine (koos algajale sobiliku lühivideoga) füüsilise tööstuskontrolleri liidestamise näitel	15.04.2020 - 06.05.2020
Töö esimene versioon valmis ja juhendajale läbilugemiseks saatmine	08.05.2020
Paranduste sisseviimine ja juhendajale teiseks läbilugemiseks saatmine	13.05.2020
Töö lõplik versioon valmis	18.05.2020

# SISUKORD

LÕPUTÖÖ LÜHIKOKKUVÕTE.....	3
ABSTRACT .....	4
LÕPUTÖÖ ÜLESANNE .....	5
EESSÕNA.....	10
LÜHENDITE JA TÄHISTE LOETELU .....	11
SISSEJUHATUS .....	12
1. ROBOTJAAMA KOMPONENTIDE 3D MODELLEERIMINE .....	13
1.1 Mudeli alustamine .....	13
1.1.1 Vaadete muutmine modelleerimiskeskkonnas.....	14
1.2 Mudeli redigeerimine ja kohandamine.....	15
1.1.2 Mõõtmete muutmine.....	15
1.1.3 Liikumisulatuse seadistamine kinemaatikamudelis .....	16
1.1.4 Kolmemõõtmeliste objektide lisamine mudelisse ja nende suuruse muutmine .....	17
1.2 Roboti programmi sidumine simulatsioonis katsetamiseks .....	19
1.3 Füüsilise roboti andmesideühenduse seadistamine.....	25
1.3.1 Ühendumine TCP/IP protokolliga.....	25
1.3.2 Ühendumine jadapordi kaabliga .....	26
1.4 Failide avamine robotist .....	26
2. VIRTUAALSE ROBOTJAAMA KOMPONENTIDE JUHTIMINE PROGRAMMEERITAVATE KONTROLLERITEGA .....	28
2.1 Simulatsioonikeskkonna juhtimine tarkvaralise kontrolleriiga.....	28
2.2 Rakendusprogrammide liidestamine läbi OPC serveri.....	31
2.2.1 OPC serveri paigaldamine ja seadistamine .....	33
2.3 Tarkvaralise kontrolleri OPC-side virtuaalse robotjaamaga .....	35
2.4 Kontrolleri side virtuaalse robotjaamaga läbi andmeside-protokolli MODBUS TCP .....	41



KOKKUVÕTE .....	42
SUMMARY .....	44
KASUTATUD KIRJANDUS.....	45
LISAD .....	46

## EESSÕNA

Lõputöö teema püstitamisel liikus mõte juba alguses kontrollerite ning programmeerimise poole. Võtsin ühendust Tallinna Tehnikaülikooli elektroenergeetika ja mehhatroonika instituudi dotsendi Madis Lehtlaga, kellega koos arutasime lõputöö teema variante. Teemadest oli kaalumisel oli nii valgustuse juhtimise lahendused kui ka tööstusautomaatika simulatsioonid. Lõplik valik langes 3D keskkonnas kasutusjuhenditele, millest on kasu nii tudengitel kui ka inseneridel erinevates valdkondades (tootmise automatiseerimise, robotitehnika ja võibolla ka elektervalgustuse juhtimisel).

Lõputöö valmis riikliku eriolukorra tingimustes, sellegipoolest oli instituudi poolt võimaldatud ligipääs laboriruumidesse. Tööd lihtsustas asjaolu tarkvara sai kasutada ka väljaspool ülikooli ruume. Kogu lõputöö koostamise vältel oli juhendajaga tihe kontakt e-kirjade ja videosuhtluskeskkondade vahendusel.

Soovin avaldada tänu Madis Lehtlale, Vahur Maaskile ja Margus Müürile, kes aitasid minu lõputöö koostamiseks vajaliku informatsiooni ja andmetega. Lisaks soovin tänada Tallinna Tehnikaülikooli töötajaskonda, tänu kellele tekitati keerulisel ajal võimalus modelleerimistarkvaraga ülikooliväliselt töötada. Minu siiras tänu Teile kõigile!

## LÜHENDITE JA TÄHISTE LOETELU

FBD	funktsiooniplokiskeem ( <i>Function Block Diagram</i> )
FMS	paindootmissüsteem ( <i>Flexible Manufacturing System</i> )
IRL	Standardiga DIN 66312 määratud tööstusrobotite programmeerimiskeel ( <i>Industrial Robot Language</i> )
KRL	firma KUKA robotite programmeerimiskeel ( <i>KUKA Robot Language</i> )
LAD	kontaktaseskeem ( <i>Ladder logic</i> )
MPS	modulaarne tootmissüsteem ( <i>Modular Production System</i> )
OPC	avatud standardite komplekt rakendustevaheliseks sideks ( <i>Open Platform Communications</i> )
PLC	programmeeritav loogikakontroller
SFC	järjestatud funktsioonide kaart / sammprogramm ( <i>Sequential Function Chart</i> )
TCP (1)	tööstusroboti tööriista keskpunkt ( <i>Tool Centre Point</i> )
TCP (2)	edastusohje protokoll ( <i>Transmission Control Protocol</i> )

## SISSEJUHATUS

Virtuaalreaalsus leiab tööstusautomaatikas ja mujalgi üha enam kasutust. Võimalike kasutusvaldkondi on erinevaid. Üheks niisugusteks on ka tarkvara katsetamine arendusprotsessis ja süsteemi töö visualiseerimine. Luues esmalt digitaalne mudel planeeritavast süsteemiosast annab see võimaluse arendada selle tarkvara väiksema riskiga, see aitab vältida ekslike prototüüpide valmishitamist, aidates ehituse käigus välja tuua parandamist vajavaid kitsaskohti. Süsteemide tarkvara arendusprotsessis ja häälestamisel pole väga sageli võimalik teha katseid füüsilistel seadmetel, sest füüsiline katsetamine võib olla kas materjalikulukas, aeglane, ohtlik või ebatäpne. Seepärast rakendatakse üha enam virtuaalreaalsust. Selliselt säästetakse nii aega kui raha. Kui on tarvis tulevikus protsessi muuta (süsteemi tarkvara korrigeerimine, uus seade), on virtuaalsüsteem taaskord süsteemi korrigeerimisele abiks. Selliselt on võimalik planeeritavad muudatused enne visualiseerida, nähes saadud tulemit, tekkivaid probleeme ning muud sellist, samal ajal hoides töös juba töötavat lahendust. Väiksemate tarkvarakatsetuste jaoks ei pea süsteemi seiskama, katsed saab teha mudelil.

Tallinna Tehnikaülikooli tootmise automatiseerimise ja robotitehnika laboratooriumites on olemas mitmeid tööstusautomaatika modelleerimisel kasutatavaid tarkvarapakette (näiteks Ciro Studio), mida kasutatakse ka reaalsete tööstuses kasutatavate tootmiseseadmete modelleerimisel. Antud tarkvarapaketi rakendamist praktikas raskendab tudengitele asjaolu, et asjakohased ja ajakohased juhendid puuduvad. Autor soovib käesoleva lõputööga lihtsustada tulevaste tudengite tööstusautomaatika praktika läbiviimist, varustades neid põhjalike juhenditega.

Käesoleva lõputöö eesmärgiks on koostada täpsed ja asjakohased lühijuhendid algajatele Ciro Studio süsteemide (tarkvara ja riistvara) modelleerimiseks koos konkreetsete näidetega sammude kaupa. Autor soovib töö sisus kirjeldada protsesside olemust ning tagamaid. Lisaks kirjalikele juhenditele salvestab autor õpivideod. Samuti võib juhenditest olla abi tootmisprotsesside modelleerimisega tegelevatel inseneridel.

Lõputöö põhiosa on jaotatud kahte peatükki. Esimeses peatükis kirjeldatakse roboti modelleerimist Ciro Studio tarkvarapaketi koos mitmesuguste muude 3D mudelitega. Teises peatükis kirjeldatakse tööstuskontrollerite tarkvara simuleerimist ning füüsilist kontrolleri liidestamist. Selleks kasutab autor tarkvarapaketti CoDeSys. Lõputöö eesmärgi täitmiseks uurib autor varasemaid juhendeid varasema tarkvarapaketi kohta, tutvub praeguse tarkvarapaketiga, koostab ülevaate modelleerimisest, koostab juhendid roboti, tööstuskontrolleri ja visualiseerimise osas ning salvestab õpivideod simuleerimiskeskonnast. Õpivideode salvestamiseks kasutab autor Debut NCH ekraanisalvestusprogrammi.

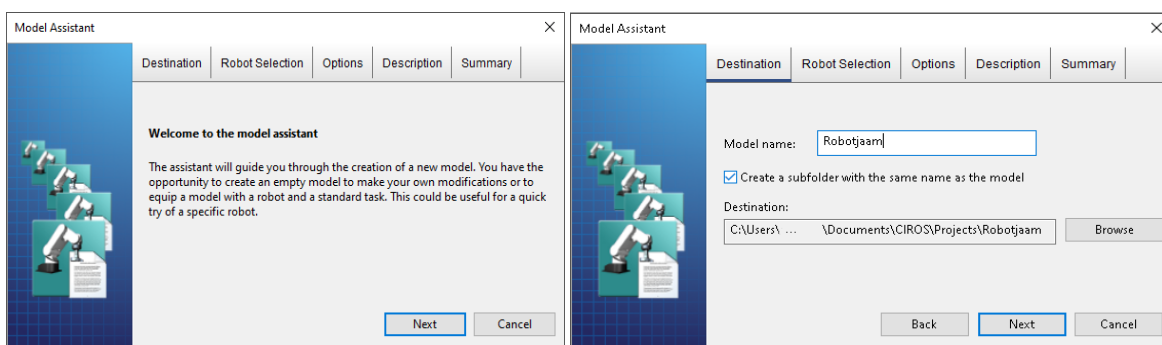
# 1. ROBOTJAAMA KOMPONENTIDE 3D MODELEERIMINE

## 1.1 Mudeli alustamine

Projekti võib alustada mitmel viisil. Võib luua tühja mudeli ja seejärel lisada sinna komponente või kasutada mudelabilist (*Model assistant*). Selles sisaldub komplekt enamkasutatavaid valmiskomponente robotjaamade mudelite koostamiseks.

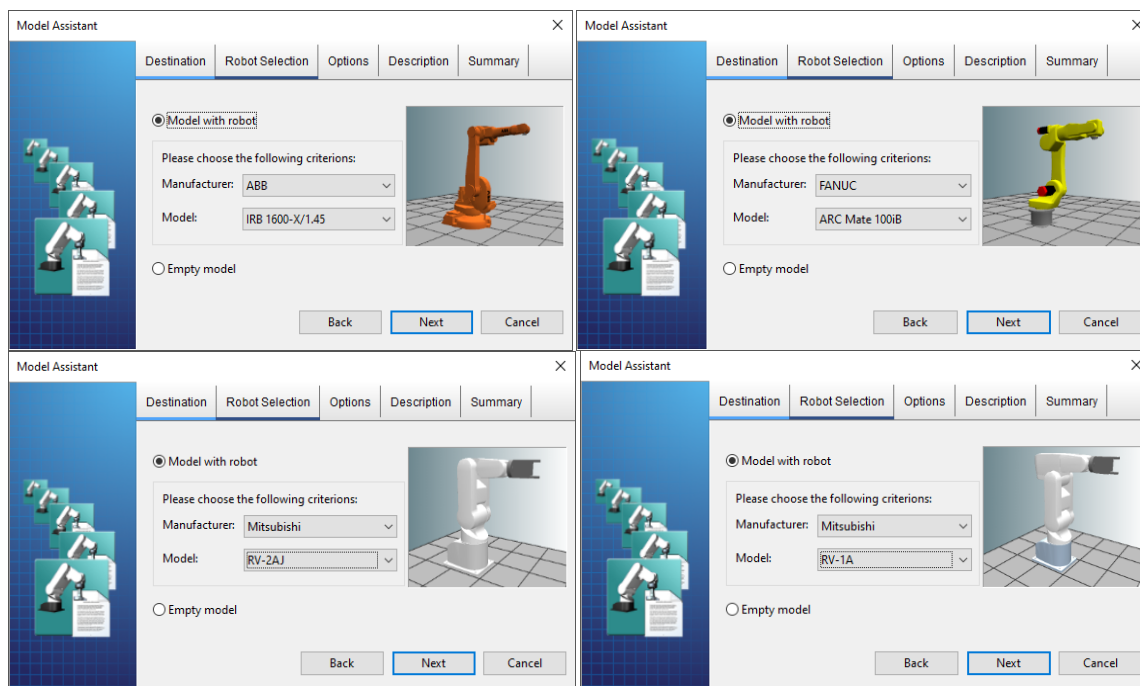
Mudelabiline käivitatakse failimenüüst, valides *File -> New -> Model assistant*

Seejärel avaneb dialoogiaken. Vajutage mudelabilise dialoogiakna „Model Assistant“ esilehel nuppu „Next“ ja seejärel sisestage mudeli nimi ja valige salvestamiskoht.



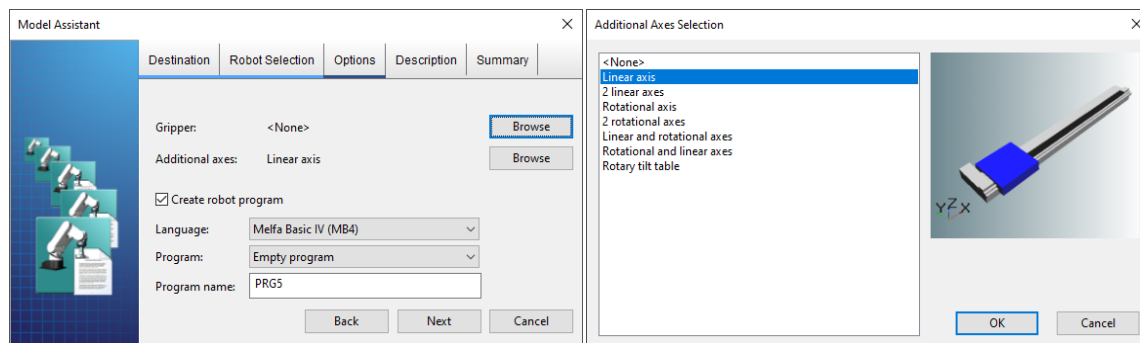
Joonis 1.1 Projekti loomine mudelabilise dialoogiaknas

Vajutades „Next“ avaneb järgnev leht, millel on võimalik valida roboti tootja ja roboti mudel.



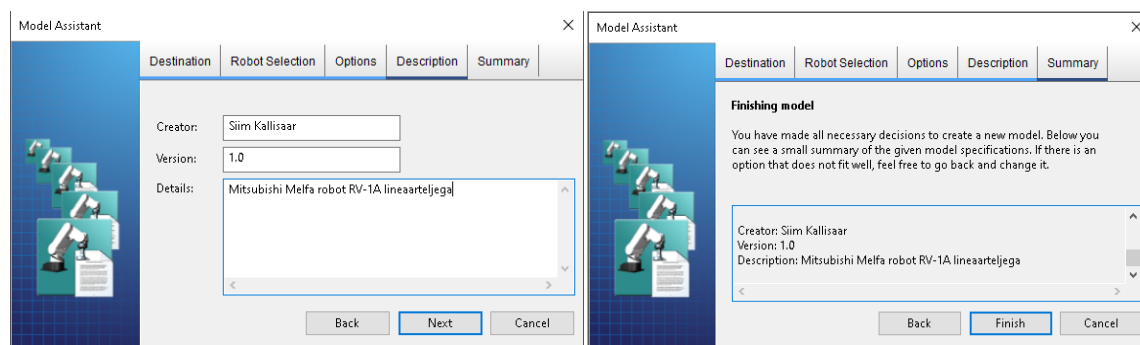
Joonis 1.2. Valik mitmesuguste tööstusrobotite mudeleid mis on ka Tallinna Tehnikaülikooli erinevates ruumides füüsiliselt olemas.

Roboti mudeli (nt Mitsubishi MELFA RV-1A) valiku järel tuleb vajutada nupule „Next“. Selle järel tekib võimalus roboti tööriistade nt haaratsite (*gripper*) ja lisatelgede (*additional axes*) valikuks. Võimalikeks lisatelgedeks on mitmesugused lineaarteljed (*linear axis*), pöörd- või kallutuslauad. Nende valikus tuleb kirja „additional axis“ juures vajutada nupule „Browse“. Antud näites valime vastavalt laboris olemasolevale süsteemile „Linear axis“. Lisaseadmete esialgse valiku järel sisestage roboti juhtimisprogrammi nimi (toodud näites PRG5) ja vajutage Next.



Joonis 1.3 Roboti lisatelje valik, juhtprogrammi keele ning nime valik.

Sisestage projekti autor, versioon ja kirjeldus, vajutage nupule „Next“. Summary ehk kokkuvõtte osas kontrollida, et valitud on õige robot, vajaminevad lisateljed ja programmeerimiskeel. Lõpetamiseks vajutage nupule „Finish“ [6].



Joonis 1.4 Autori ja versiooni lisamine koos kirjeldusega, kokkuvõtte enne roboti koostamise kinnitamist.

Seejärel loob programm vastavalt selle mudeliteekides (*model libraries*) olemasolevatele andmetele valitud roboti kolmemõõtmelise mudeli.

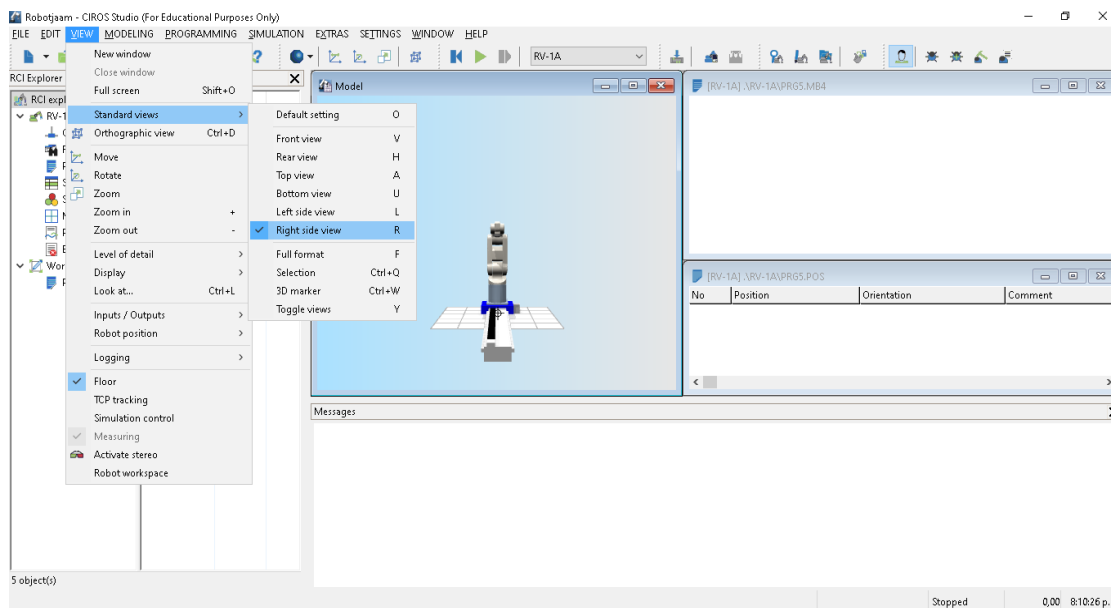
### 1.1.1 Vaadete muutmine modelleerimiskeskonnas

Olulisemad võtted mudelis navigeerimiseks on järgnevad.

- Mudeli liigutamiseks tuleb hoida all SHIFT klahvi ja liigutada hiirt
- Suurendamiseks ja vähendamiseks hoida all korraga CTRL ja SHIFT klahve ja liigutada hiirt või kasutada hiire rullikut.

Mudeli pööramiseks hoida all kas CTRL või ALT klahve ja liigutada hiirt.

Lineaarteljel paikneva roboti eestvaade on vaade mudelile paremalt „Right side view“ (*View -> Standard views -> Right side view*), sel juhul on roboti baaskoordinaatsüsteemi X-telg otse ette, Y-telg paremale ja Z-telg ülesse. Mudeli vaadete valikust annab ülevaate järgnev joonis.



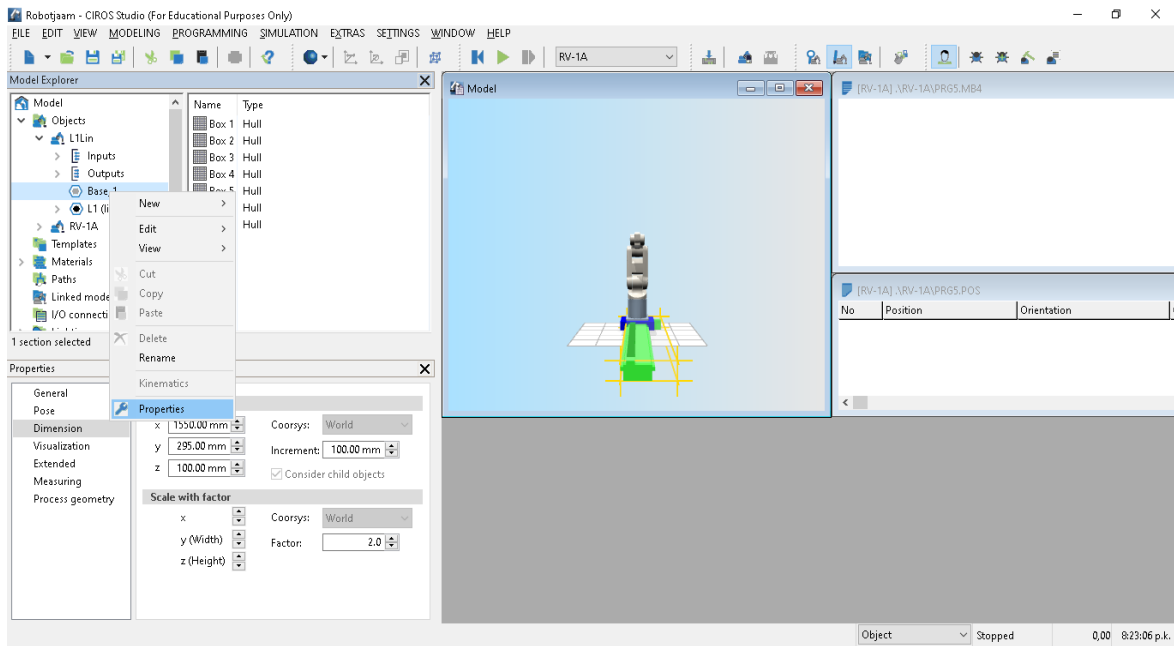
Joonis 1.5. Mudeli vaate valik

## 1.2 Mudeli redigeerimine ja kohandamine

### 1.1.2 Mõõtmete muutmine

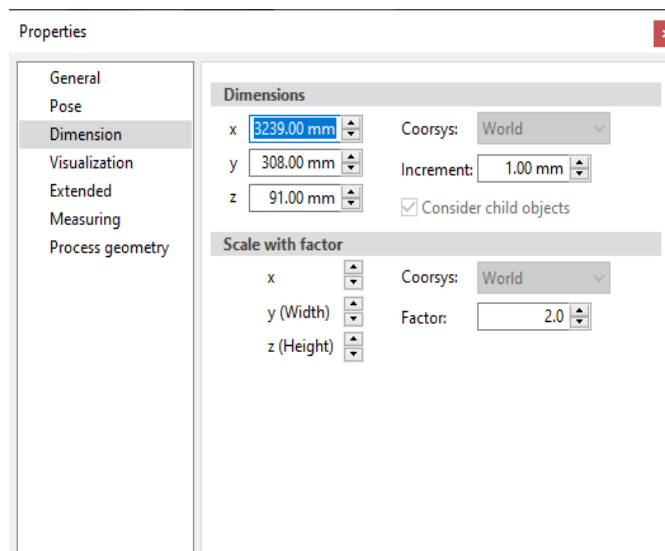
Lisades mudelisse objekte ei pruugi need kohe alguses olla õigete mõõtmetega. Näiteks lineaartelje pikkus mudelis tuleb muuta vastavaks tegeliku robotjaama lineaartelje pikkusega. Selleks tuleb teha järgnevad toimingud.

1. Aktiveerida Edit Mode (*Modeling -> Edit mode*)
2. Valida Model Explorer (*Modeling -> Model explorer*)
3. Valida mudelipuust aktiivseks lineaartelje alus Base, teha paremklõps, valida Properties



Joonis 1.6. Objektide mõõtmete muutmine mudelis

Parameetrite muutmise dialoogist tuleb valida „Dimension“. Antud juhul soovime muuta lisatelje pikkust vastavaks tegelikkusele. Näiteks Tallinna Tehnikaülikoolis õppelaboris NRG-423 on lineaartelje pikkus  $l_1=3239$  mm. Pikkuse muutmine on tarvilik selle jaoks, et praegu ehitatav mudel langeks kokku laboris oleva mudeliga, muutes seetõttu pärastise simulatsiooni võimalikult tõeseks ning võimaldab kasutada olemasolevat telge maksimaalselt.

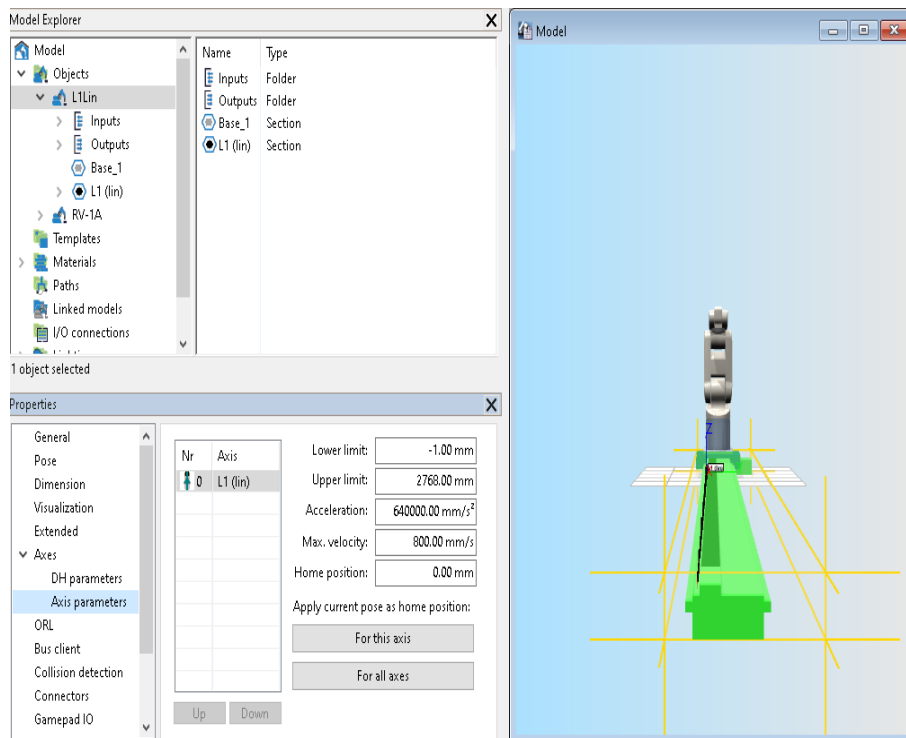


Joonis 1.7. Objekti mõõtmete muutmise dialoog

### 1.1.3 Liikumisulatus seadistamine kinemaatikamudelis

Seejärel valida aktiivseks lineaartelg ja selle alt „Axis parameters“ ja muuta lineaartelje lubatud liikumisulatus vastavaks tegeliku robotjaama lineaartelje liikumisulatusega.



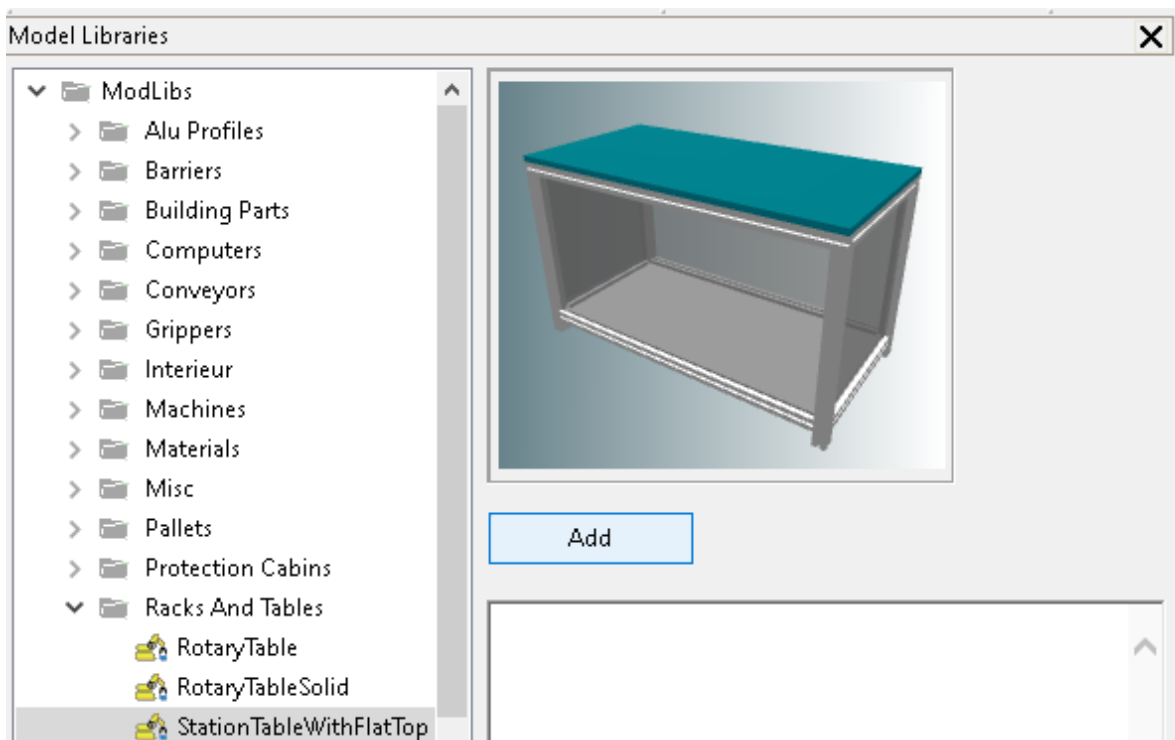


Joonis 1.8. Liikumisulatus muutmine kinemaatikamudeli telgede parameetrites

Laboris NRG-423 on kasutusel lineaartelj Festo DGE seeriast, mille liikumisulatus  $l_a = 2768$  mm ja telje kogupikkus  $l_i = 3239$  mm [4][5].

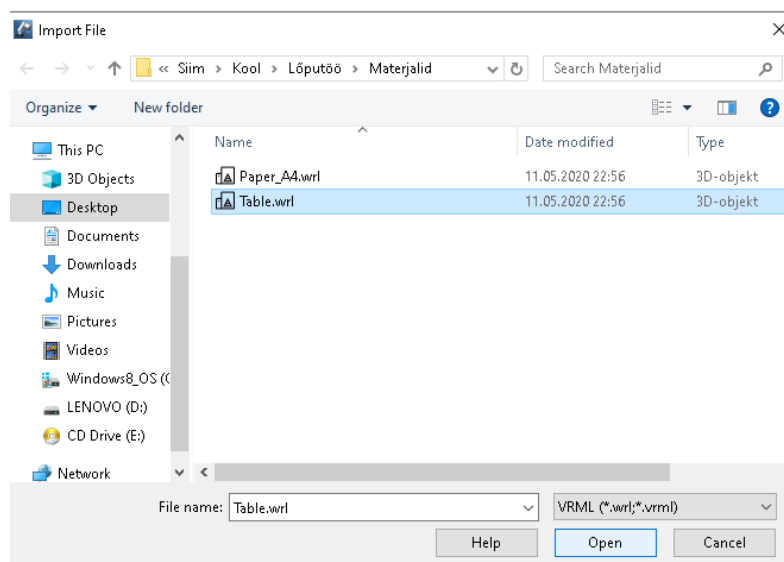
#### 1.1.4 Kolmemõõtmeliste objektide lisamine mudelisse ja nende suuruse muutmine

Selleks, et lisada mudelile juurde 3D objekte, tuleb valida Modeling → Model Libraries. Valime alammenüüst ModLibs → Racks and Tables 3D objekti StationTableWithFlatTop.



Joonis 1.9. Valmis 3D mudeli valimine

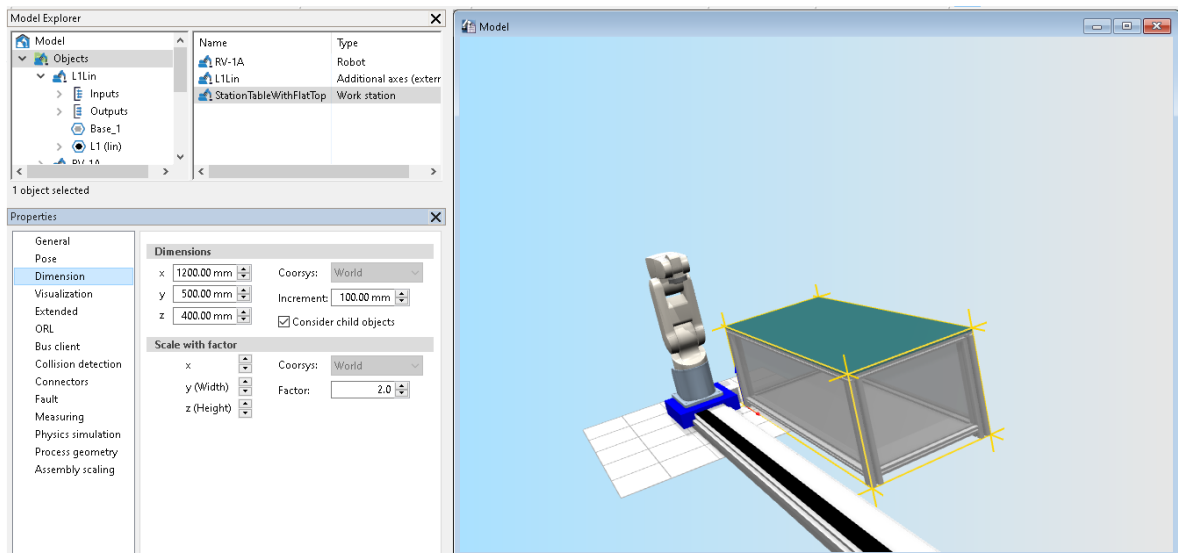
Lisaks valmis mudeli valimisele on võimalik ka erinevaid mudeleid importida. Selleks tuleb minna File → Import ja valida soovitud mudel.



Joonis 1.10. 3D mudelite importimine failidest

Laua mõõtmete ning koordinaatide muutmiseks tuleb valida Model Explorer ning teha parem hiireklõps StationTableWithFlatTop peal. Valida Properties.

Mõõte on võimalik muuta Dimension alamkategorias, asukoha koordinaate Pose alamkategorias [1].



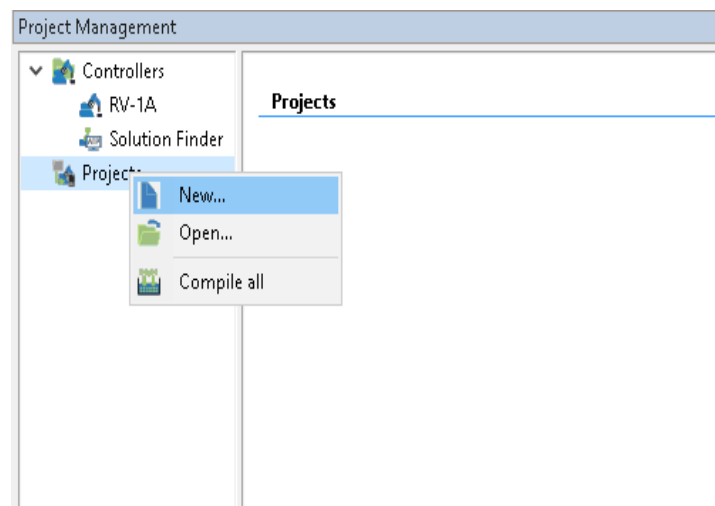
Joonis 1.11. 3D mudeli mõõtmete muutmise

## 1.2 Roboti programmi sidumine simulatsioonis katsetamiseks

Simulatsiooni kompileerimiseks tuleb lisada positsioonilist ja programmifaili projekti

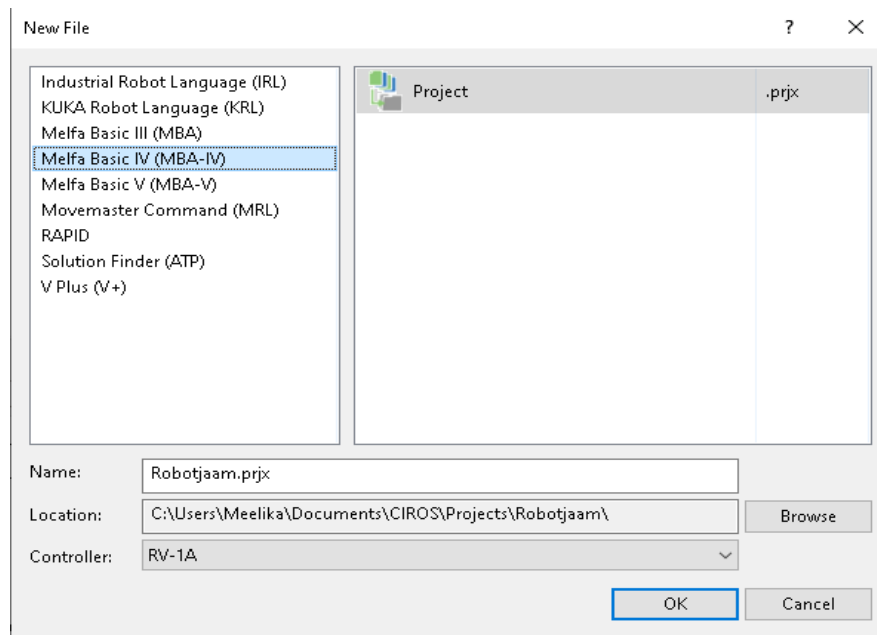
(*PROGRAMMING* -> *Project management*)

Juhul, kui Model Assistantis ei valitud roboti keelt, tuleb see praegu käsitsi teha. Selleks tuleb teha parem hiireklõps Projects'i peal ning valida New.



Joonis 1.12. Programmifailide projekti lisamine

Seejärel roboti keel (toodud näites Melfa Basic IV).



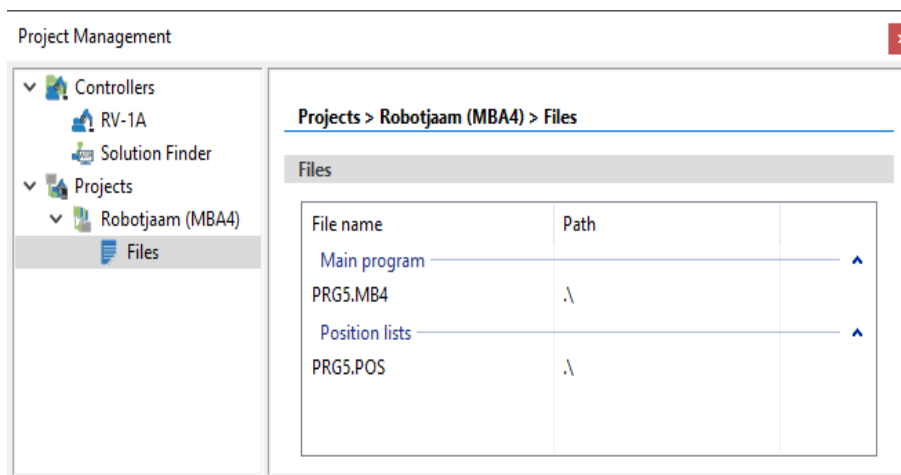
Joonis 1.13. Robotite programmeerimiskeelte valik

Pakett Ciros Studio pakub roboti programmeerimiskeeltena mitmesuguseid valikuid:

- IRL (*Industrial Robot Language*) on Saksa standardi järgi loodud Pascalil põhinev programmeerimiskeel, mis võimaldab mitme roboti ja seadme samaaegset programmeerimist. See on võimalik tänu paralleeltöö täpsustamisele ning sammprogrammide osadele.
- KRL (*KUKA Robot Language*) on mõeldud KUKA robotite programmeerimiseks ning sarnaneb samuti Pascalile.
- MBA (*Melfa Basic III*), MBA-IV (*Melfa Basic IV*) ja MBA-V (*Melfa Basic V*) ehk MELFA-BASIC on kõik loodud teatud Mitsubishi robotite programmeerimiseks (ka meie Melfa RV-1A). Rooma tähised tähendavad edasiarendatud programmi versiooni, kusjuures iga edasiarendusega on suurenenud võimalused ning programmi kasutaja jaoks lihtsustatud.
- MRL (*Movemaster Command*) on Mitsubishi robotite programmeerimiskeel, mis on vanem kui MELFA-BASIC keeled.
- RAPID on ABB robotite programmeerimiskeel, mis sarnaneb samuti Pascalile.
- APT (*Solution Finder*) on programmeerimiskeel, mis eelnes tänapäevastele CAM süsteemidele. Seda kasutatakse numbriliselt kontrollitavate tööriistade liigutamiseks, et luua keerukaid kujusid (sisuliselt CNC).
- V+ (*V Plus*) on mõeldud Stäubli ja Adept kontrollritega robotitele.

Kuigi paljud erinevad programmeerimiskeeled tuginevad programmeerimiskeeltele Pascal ja BASIC, on igal keelel oma iseärasused, mistõttu Pascali tundmine lihtsustab programmeerimiskeelte kasutamist, kuid enne programmi valmis kirjutamist tuleb tutvuda iga keelega eraldi.

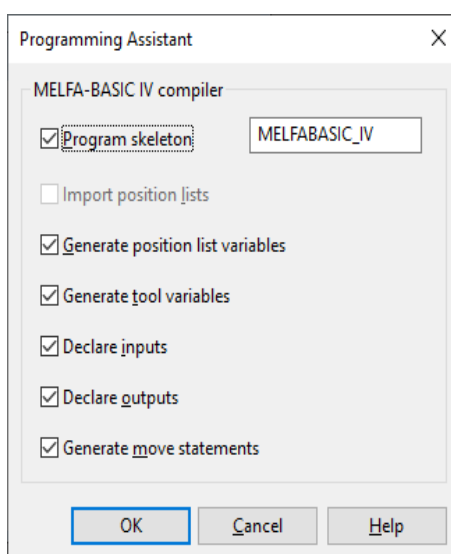
Positsioonilisti ja programmifaili saab projekti lisades tehes Files'i peal parem hiireklõps, valides New ning esmalt programmifaili, seejärel korrata sammu sama kohani ja valida positsioonilist.



Joonis 1.14. Projekti haldur koos roboti juhtimisprogrammi ja positsioonilistiga

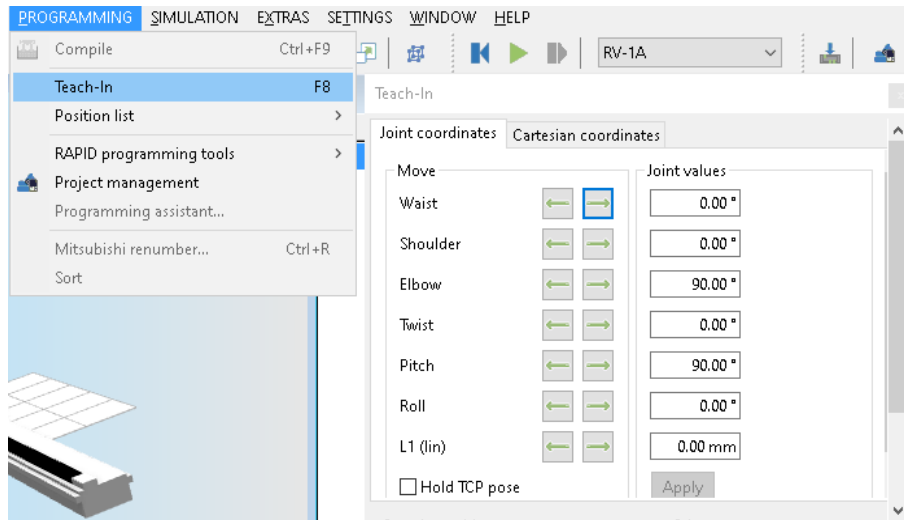
Simulatsiooni kompileerimiseks tuleb avada programmifail (toodud näites PRG5.MB4). Programmifail avaneb projekti haldurist hiire topeltvajutusega.

Programmi koostamist saab lihtsustada programmeerimisabilisega (*Programming Assistant*), mis genereerib esialgse programmikoodi. Selleks peab olema programmifaili aken aktiivne.



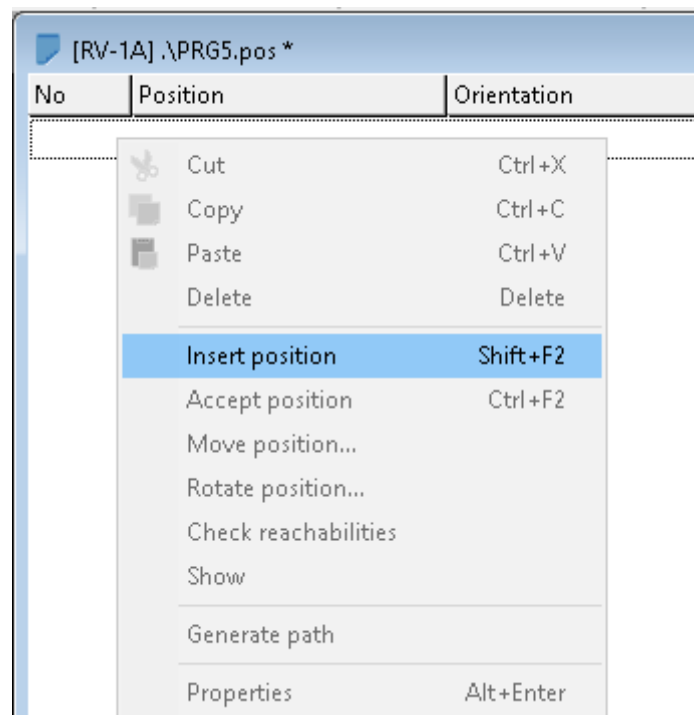
Joonis 1.15. Programmikoodi automaatgenereerimise võimalused

Valides Programming → Teach-In, saate robotit käsitsi liigutada. See on abiks roboti positsioonilisti täiendamiseks.



Joonis 1.16. Virtuaalse roboti liigutamine

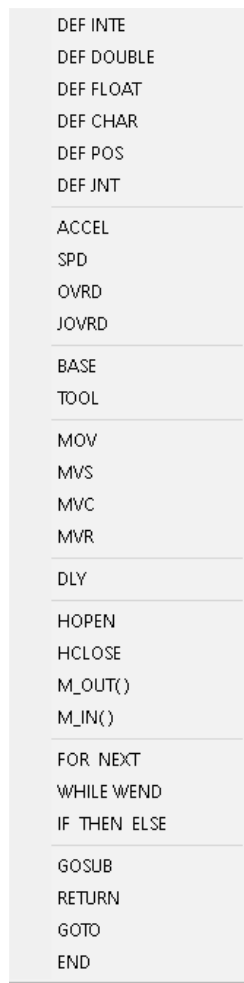
Positsioonilisti asendite lisamiseks on tarvis liigutada robot sellisesse asendisse, kui on tarvis, võtta ette positsioonilisti aken, teha parem hiireklõps ja vajutada Insert Position. Õpetage selgeks neli asendit.



Joonis 1.17. Positsiooni lisamine positsioonilisti

Programmikoodi kirjutamiseks peale programmeerimisassistendi kasutamist tuleb otsida rida, kus on kirjas lisa oma kood siia. Meie näitel rida 170.

Võimalike käskude nimekirja on näha, tehes programmifaili aknas parem hiireklõps.



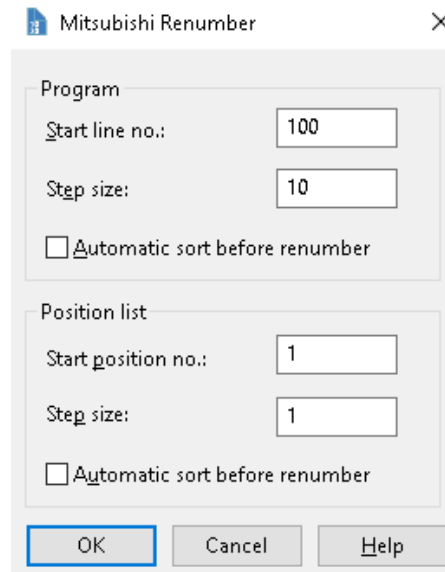
Joonis 1.18. Programmeerimiskeele käskude kiirvalikumenüü

Algajale on tähtsad käsud MOV (liigendliikumine), MVS (sirgliikumine), DLY (viivitus), HOPEN (haaratsi avamine, haaratsi number täpsustada), HCLOSE (haaratsi sulgemine, haaratsi number täpsustada) ja END (programmi lõpp) [2].

Liikumiskäskudel on võimalik lisada määratud punktist liikuda mööda Z-telge üles ja alla, näiteks.

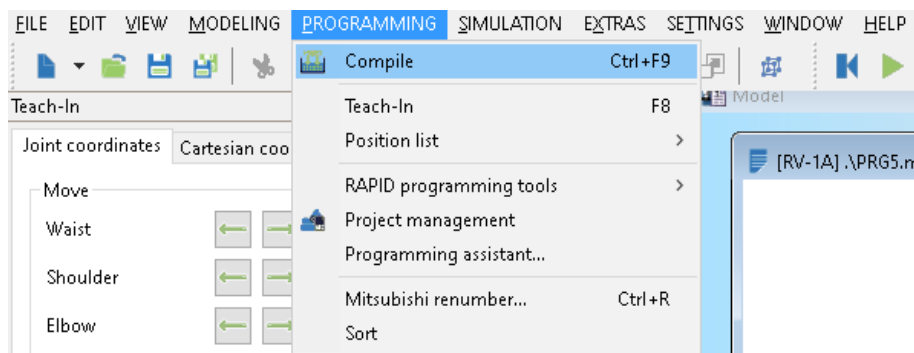
```
100 MOV P1, 50
```

Vaadeldava roboti programmeerimiskeele versioon eeldab programmiridade nummerdamist. Peale koodi kirjutamist tuleb seega read automaatselt nummerdada juhul kui ridasid pole enne käsitsi nummerdatud. Selleks vaalide menüüst „Programming“ → „Mitsubishi renumber“.



Joonis 1.19. BASIC-keele ridade automaatne ümbernummerdamine

Selleks, et simulatsioon töötaks, on vaja programm kompileerida. Kompileerimine toimib ainult siis, kui aktiivseks on tehtud programmifaili aken.



Joonis 1.20. Programmikoodi kompileerimine simulatsioonis kasutamiseks

Näidiskoodist oma kirjutatud koodilõik lisatud [3].

```

170 REM TODO: Add your code here
180 MOV P1
190 DLY 0.1
200 MOV P3
210 MVS P3, 100
220 MVS P3
230 MOV P4
240 MOV P5, -50
250 MVS P5
260 MVS P5, -50
270 MOV P1
280 END

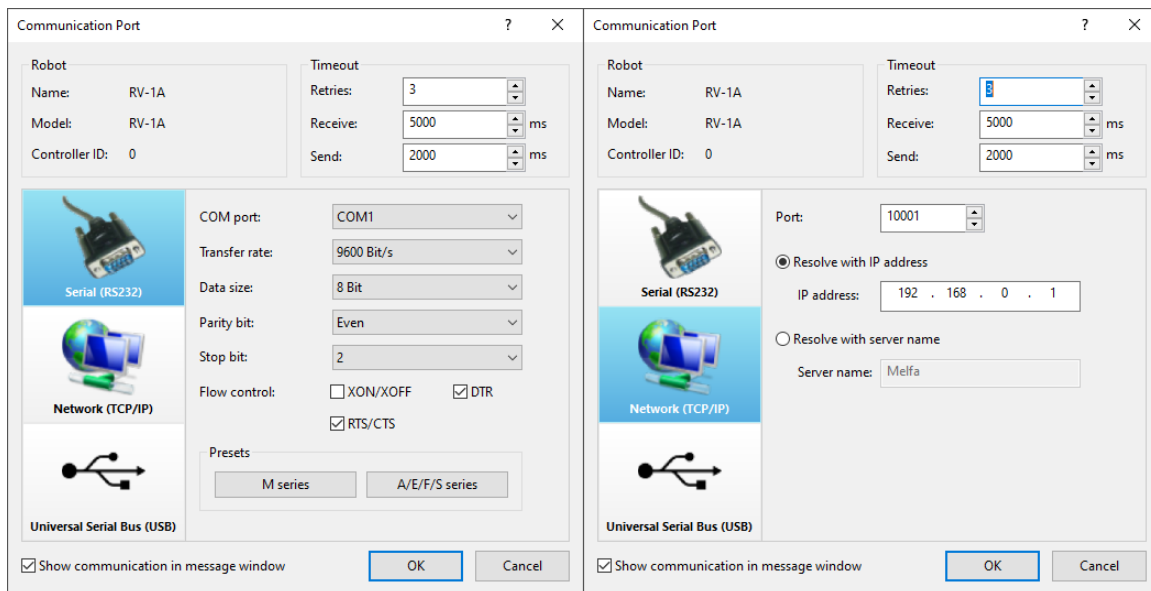
```

Joonis 1.21 Näidiskood



## 1.3 Füüsilise roboti andmesideühenduse seadistamine

Robotiga ühenduse loomiseks peab mudelis olema sama robot, millega ühendutakse, vastasel juhul kuvatakse veateade ja ühenduse loomine robotiga ei ole võimalik.



Joonis 1.22. Erinevad võimalused ühenduse loomiseks füüsilise robotiga

### 1.3.1 Ühendumine TCP/IP protokolliga

Robotist simulatsiooni koodi saatmiseks on vaja luua ühendus roboti ja arvuti vahel Ethernet LAN kaabli kaudu. See eeldab, et roboti kontrollerial on Etherneti kaart ning kontrollerial on samas alamvõrgus juhtarvutiga. Paketi CIROS Studio litsents kinnitatakse samuti Etherneti võrgu kaudu, seetõttu on arvutis vaja kahte Ethernet kaabli sisendit ja vajadusel tuleb lisada täiendav võrgukaart. Kui ühendus CIROS Studio litsentsiserveriga on loodud peab ühenduse looma ka robotiga [8].

Antud näites toodud roboti (Laboris NRG-423 lineaarteljel paikneva roboti RV-1A) aadress sisevõrgus on 192.168.10.10. Niisuguse aadressi poole pöördumiseks peab ka arvuti olema samas alamvõrgus so täiendava võrgukaardi seadistus või lisa IP-aadress alamvõrgus 192.168.10.x. Arvuti võrgukaardi enda IP aadressina oli antud näites kasutusel 192.168.10.35.

Seadistused CIROS Studio tarkvarapaketi on järgnevad:

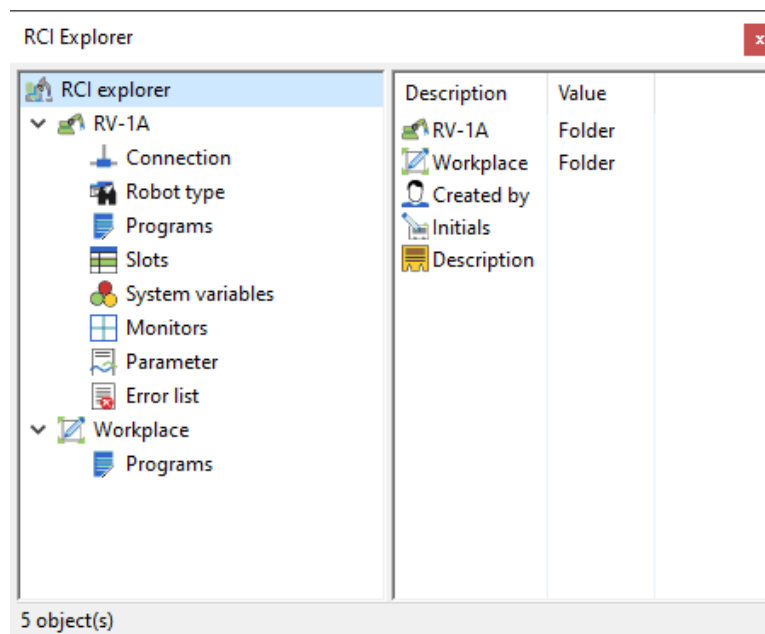
- *Settings -> Online Management-> Communication port*
- Valida ülevalt dialoogiaken TCP/IP
- Roboti IP aadressiks määrata 192.168.10.10
- Roboti TCP-Pordi numbriks 9000 või 10005

### 1.3.2 Ühendumine jadapordi kaabliga

NB! ainult ühel robotil laboris NRG-423 on Etherneti kaart (juhtseadme lisa), teisega ühendutatakse jadaliidese (COM port) kaudu. Jadaliidese ühendumisel on andmekiiruse maksimum 19200 bit/s. See tuleneb konkreetse roboti kontrolleri sidekiiruse maksimumist [7].

## 1.4 Failide avamine robotist

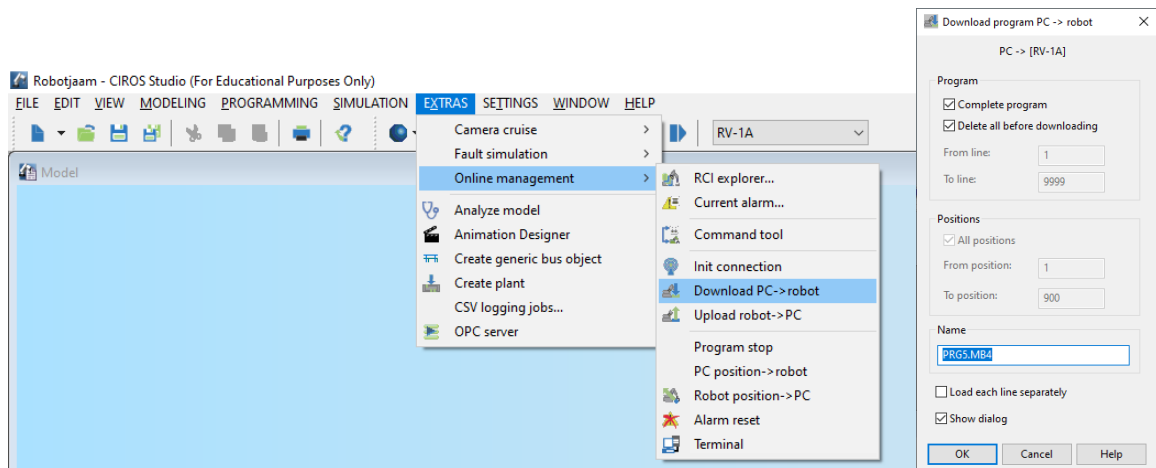
Roboti juhtseadme mälus olevate failide vaatamiseks ja avamiseks või redigeerimiseks saab kasutada paketi Ciro Studio liidest „RCI Explorer“. Kui liides „RCI Explorer“ oli eelnevalt avatud siis ühenduse uuesti loomisel robotiga on soovitatav sideprobleemide vältimiseks vana aken sulgeda ja seejärel uuesti avada.



Joonis 1.23. Füüsilise roboti liides pakettis Ciro Studio

Valides eelneval joonisel kujutatud liidesevastava roboti alt alajaotise „Programs“ saab vaadata roboti mälus olemasolevaid programme koos nende muutmise kuupäevadega. Seejärel saab robotist alla laadida vajaliku programmi koodi.

Paketi Ciro Studio mudelis olemasoleva programmi laadimiseks robotisse toimub alajaotisest *Extras -> Online Management -> Download PC -> robot* (vt. joonis).



Joonis 1.24. Programmikoodi laadimine personaalarvutist robotisse.

Programmi edukaks laadimiseks ja kasutamiseks füüsilises robotis tuleb arvestada mitmesuguste piirangutega. Näiteks roboti programmifaili ja liikumise sihtpunktide nimed peavad olema samad ja mitte pikemad kui vastava roboti juhtseade võimaldab (näiteks kuvamiseks roboti juhtseadme esipaneeli 7-segmendilisel kuvaril mitte üle 4 sümboli ja kasutades ainult 7-segmendilisel näidikul eristatavaid sümboleid). Simulatsiooniprogrammis failinimedele piiranguid ei ole [8].

## 2. VIRTUAALSE ROBOTJAAMA KOMPONENTIDE JUHTIMINE PROGRAMMEERITAVATE KONTROLLERITEGA

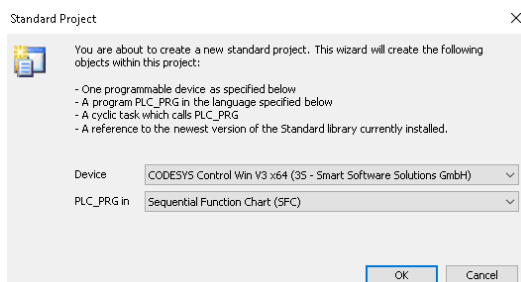
### 2.1 Simulatsioonikeskkonna juhtimine tarkvaralise kontrolleriiga

Virtuaalse robotjaama juhtimiseks saab kasutada tarkvaralist kontrolleri CoDeSys. Antud näites on valitud CoDeSys V3.5. Süsteemi seadistamisest ja programmi koostamisest keskkonnas CoDeSys 3.5 annab ülevaate järgnev lühijuhend [9].

Uue projekti alustamiseks CoDeSys keskkonnas tuleb:

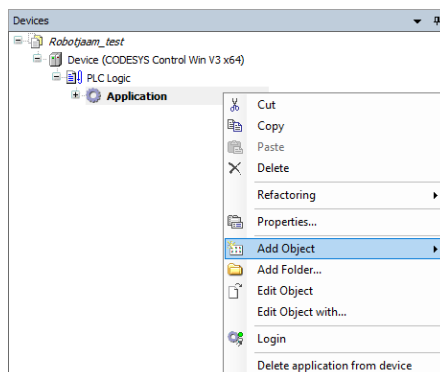
1. Käivitada tarkvaralise programmeeritava kontrolleri seadistusprogramm CodeSys V3.5.  
Seejärel valida File → New Project
2. Valida „Standard Project“ ning lisada pealkiri.
3. Valige seade (nt tarkvaralise kontrolleri tüüp), mille jaoks projekti loote ning millises programmeerimiskeeles soovite projekti koostada.

Käesolevas näites valime programmeerimiskeeleks SFC (*Sequential Function Chart*) kasutamiseks kontrolleriil CODESYS Control Win V3 x64 (vt joonis ).



Joonis 2.1 Projekti seadme ning programmeerimiskeele valik

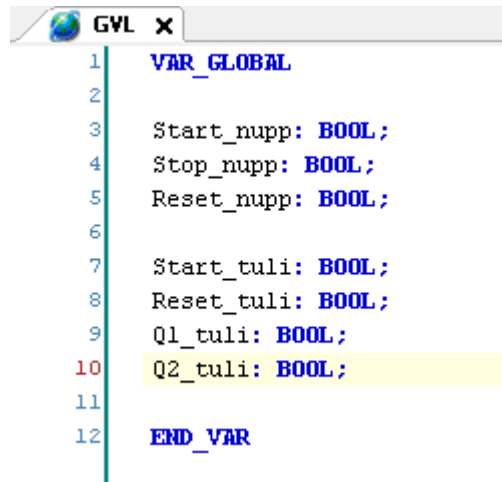
Rakenduse (*application*) loomise järel saab lisada sellesse sisu.



Joonis 2.2 Objektide lisamine rakendusse.

Järgnevalt loome muutujate nimekirja, vajutades sõnal „Application“ parema hiireklahviga ja valides „Add Object“ alt avanevast menüüst „Global Variable List“.

Kirjutame globaalmuutujate nimekirja muutujad kolme nupu oleku kontrollimiseks.



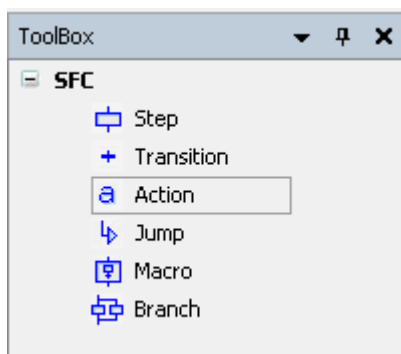
```
1  VAR_GLOBAL
2
3  Start_nupp: BOOL;
4  Stop_nupp:  BOOL;
5  Reset_nupp:  BOOL;
6
7  Start_tuli:  BOOL;
8  Reset_tuli:  BOOL;
9  Q1_tuli:    BOOL;
10 Q2_tuli:    BOOL;
11
12 END_VAR
```

Joonis 2.3 Globaalmuutujate nimekiri

Peale seda tuleb muutujate nimekiri üles ehitada, valides menüüribalt Build -> Rebuild.

Edasi tuleb koostada loogikakontrolleri programm.

SFC programmeerimiskeele puhul on kasutusel järgnevad lülid (vt joonis).



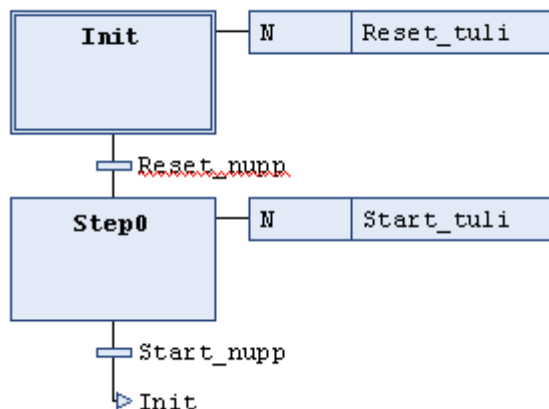
Sammu (*Step*) jooksul teostatakse programmis määratud tegevusi (*Action*, väljund) kuni täidetud on järgmise sammuni viiv tingimus (*Transition*, sisend).

Macro on alamprogrammid, mis võimaldavad teatud programmi osasid plokkidesse peita. Seda kasutatakse enamasti programmi lihtsustamiseks.

Jump lüli võimaldab liikuda tagasi või edasi teatud sammuni. Kasutades Hargnemisega (*Branch*) on võimalik programmi luua lisaharud, mida mööda on võimalik programmil edasi liikuda kasutades üht või teist teed.

Joonis 2.4 Järjestatud funktsioonikaardi (SFC) elemendid keskkonnas CoDeSys

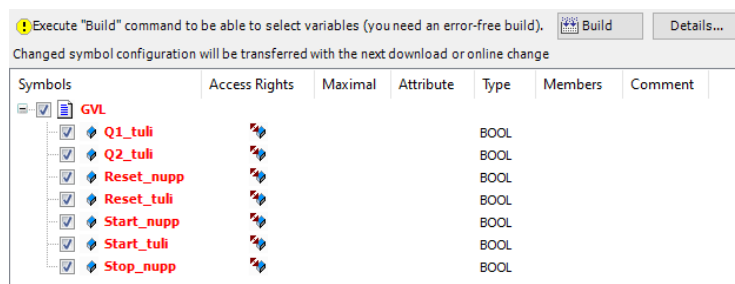
Toodud näites loome programmi, kus alustuseks põleb robotjaama kasutajapaneelil signaallamp „Reset“. Peale Reset nupu vajutamist lülitatakse signaallamp „Reset“ välja ning lülitatakse sisse signaallamp „Start“. Vajutades Start nuppu lülitatakse programmi abil signaallamp „Start“ välja ning programm algab otsast peale.



Joonis 2.5. Kasutajaliidese juhtimisprogramm järjestatud funktsioonide kaardina SFC

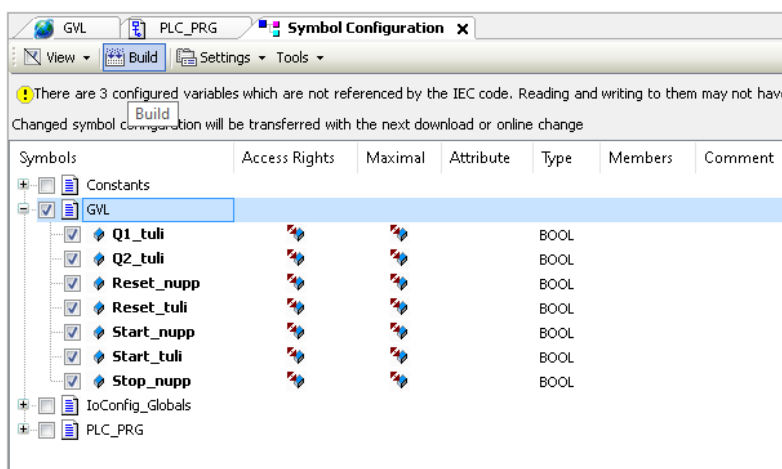
Programmis määratud sümbolitele tuleb määrata ka muutujatüübid ja määrata ka muud parameetrid. Muutujate kasutamiseks ja sidumiseks tuleb projekt eelnevalt edukalt kompilleerida.

Lisage Symbol Configuration, tehes parem hiireklõps Applicationi peal ning valides Add Object -> Symbol Configuration. Lisage linnuke Support OPC UA Features'ile ning valige Add.



Joonis 2.6 Muutujate nimekiri

Lisage Symbol Configurationis linnuke GVL ette ning seejärel kompilleerige programmikood valides Build -> Build või vajutades F11 klahvi.



Joonis 2.7 Sümbolite nimekiri pärast kompilleerimist

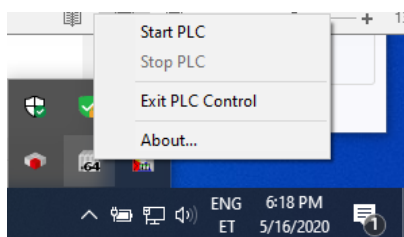
Muutujad, mis on määratud ligipääsuks OPC UA protokolliga on ligipääsetavad Windowsi keskkonna teistele rakendustele läbi nimetatud protokoll serveriprogrammi. Tarkvaralise kontrolleri seadistamist tööks läbi OPC UA protokoll on kirjeldatud järgnevas punktis.

## 2.2 Rakendusprogrammide liidestamine läbi OPC serveri

OPC protokoll on standard, mis defineerib erinevate tootjate seadmete andmesuhtluse. Antud protokoll loodi selleks, et Microsoft Windowsi baasil oleva tarkvara ning protsessi kontrolli riistvara ühendada.

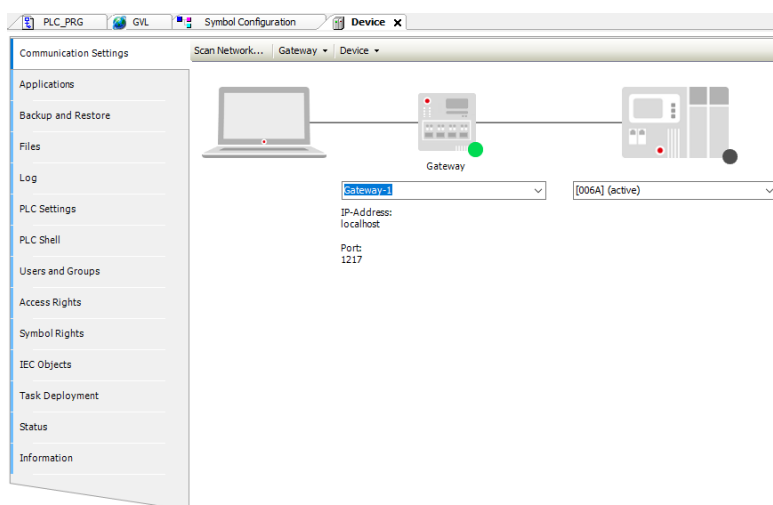
OPC protokoll töötab Server ja Client põhimõttel, kus Server edastab Clientile päringu, Client töötleb seda, vajadusel käivitab protsessi ning annab tagasisidet Serverile.

Esmalt tuleb käivitada seade, mille jaoks projekt loodi (CoDeSys Control Win V3 X64). Selleks vajutada Windows -> CODESYS -> CoDeSys ControlWin V3 - X64. Tarkvaralise kontrolleri käivitamiseks võib kasutada spetsiaalset menüüprogrammi „CoDeSys ControlWin V3 - X64 SysTray“.

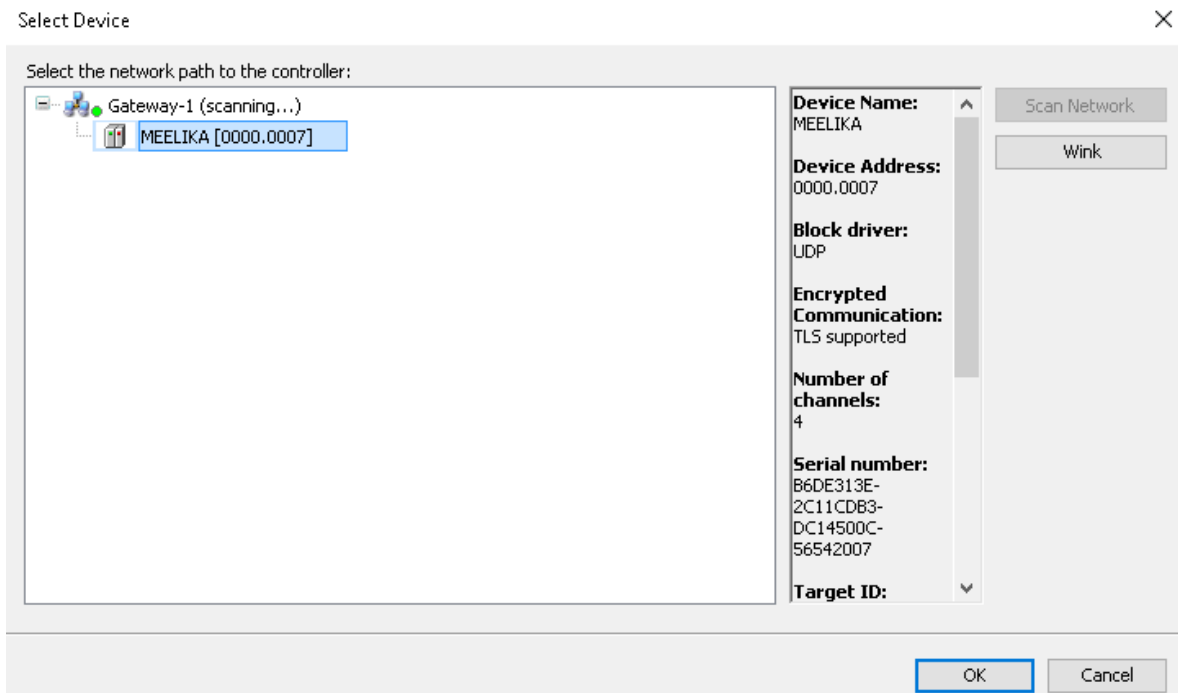


Joonis 2.8 Programmi „CODESYS Control Win V3 – x64“ käivitusmenüü

Avage CoDeSys programmis hiire topeltvajutusega alajaotisel „Device“ seadistusleht „Device“ ning valige „Communication Settings“ alt „Scan Network“.

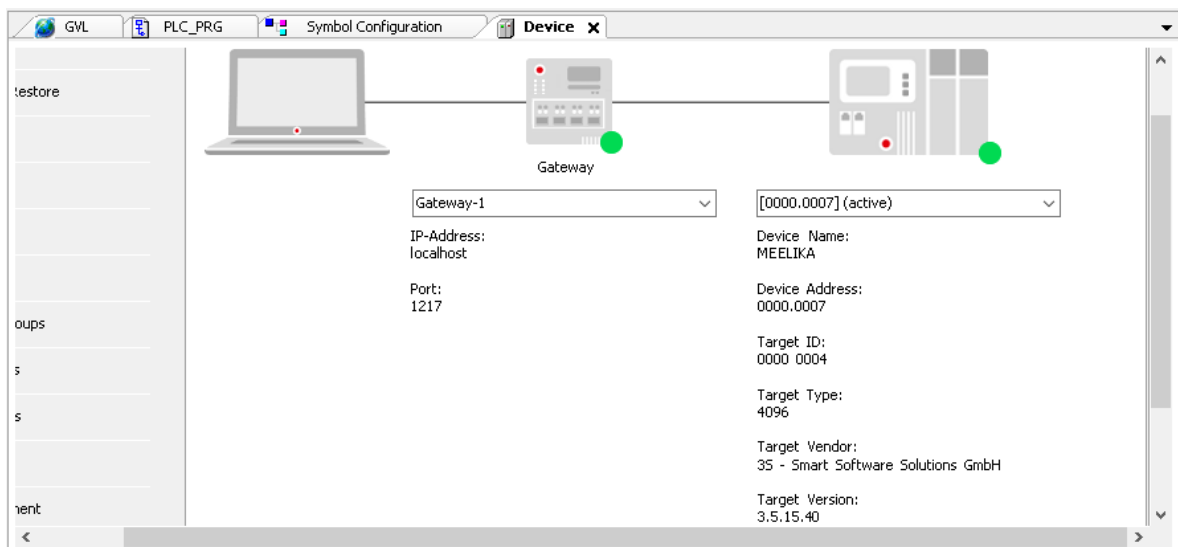


Joonis 2.9. Aken seadmete otsinguks võrgus



Joonis 2.10 Ühenduva seadme valik

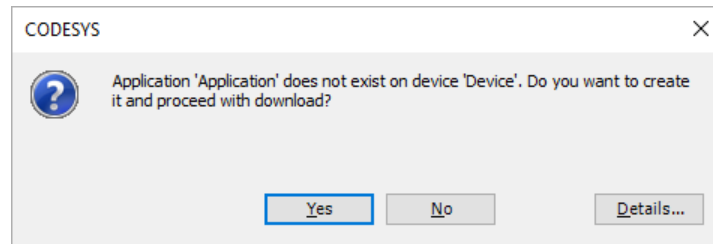
Õnnestunud ühenduse puhul on nii Gateway kui Device juures roheline märk.



Joonis 2.11 Õnnestunud ühendus pääsupunkti ning seadme vahel.

Peale õnnestunud ühenduse loomist tarkvaralise kontrolleri saate CoDeSys keskkonnas loodud programmi tarkvaralisse seadmesse laadida. Selleks tuleb valida menüüribalt Online -> Login.

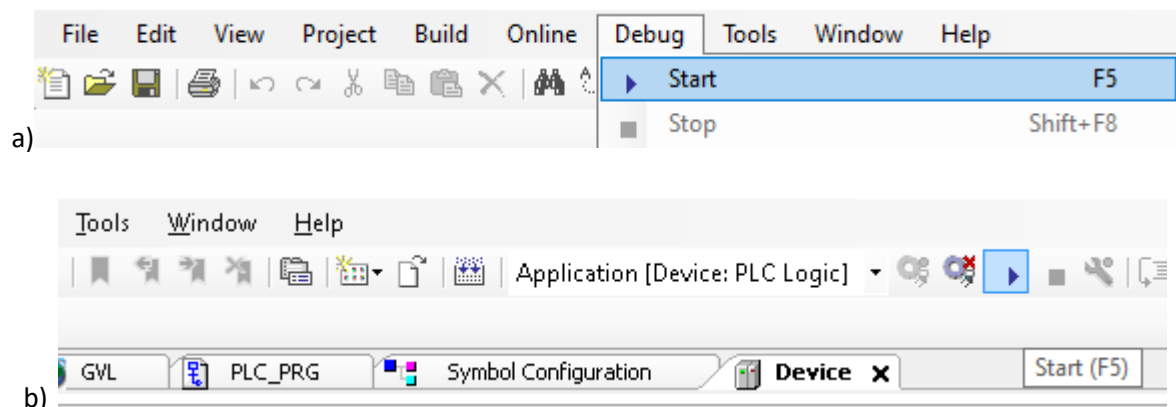




Joonis 2.12 Programmi laadimise dialoog

Avanenud dialoogiaknale vajutada Yes, et seadmesse laadida koostatud aplikatsioon.

Vajutage Start, et programm käivitada. Selleks on mitu võimalust.

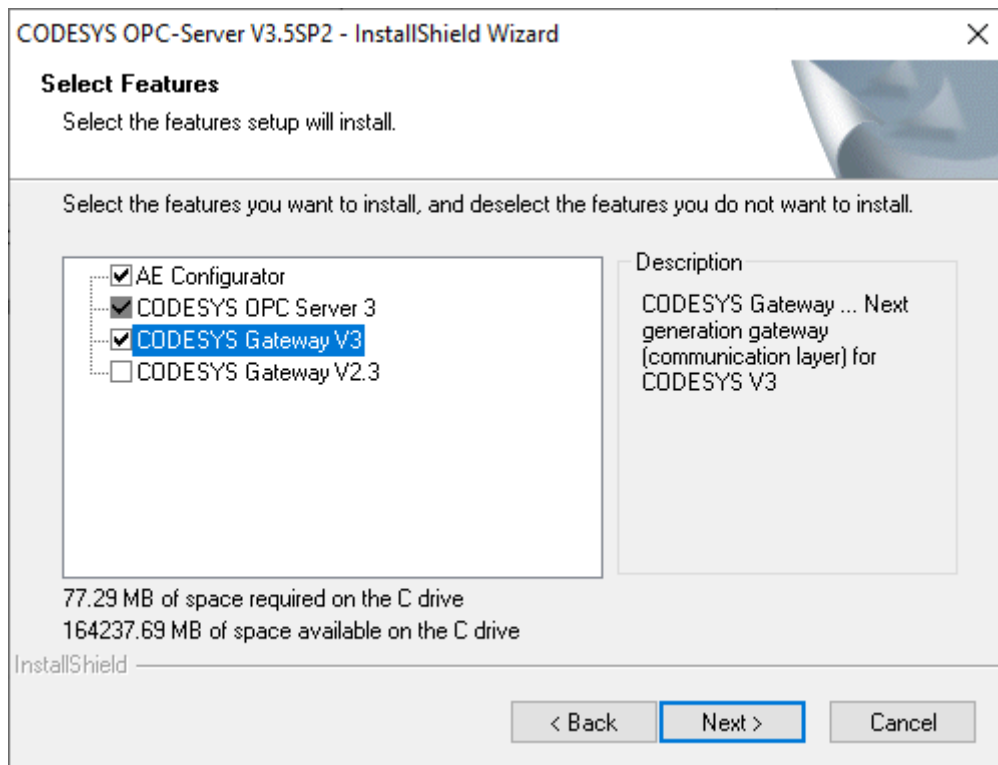


Joonis 2.13 Loodud programmi käivitamine

Akna alaservas olevale olekuribal asendub punane kiri STOP rohelise kirjaga RUN.

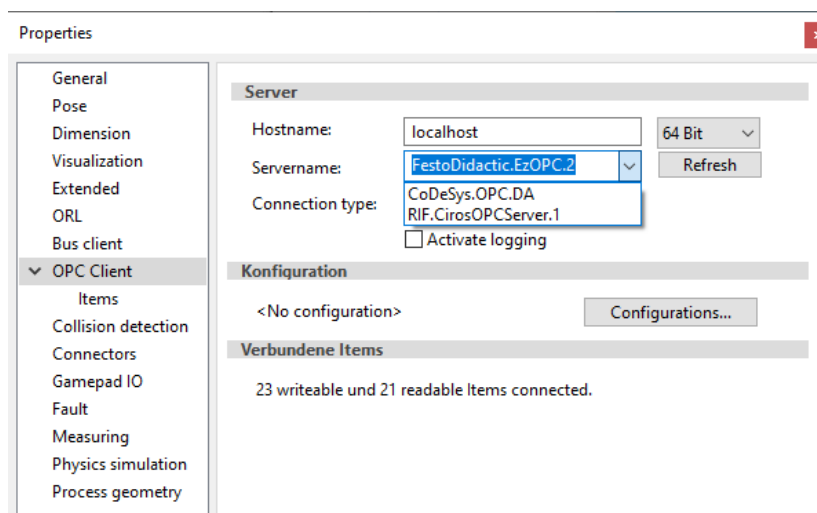
### 2.2.1 OPC serveri paigaldamine ja seadistamine

**Oluline on kasutada CoDeSysi versioonile sobivat OPC programmi.** Näiteks PLCWinNT, mis toimib CoDeSys versiooniga V2.3 ei pruugi toimida versiooniga V3.5. Sobib näiteks „CODESYS OPC DA Server SL“, katsetamiseks saab kasutada 30 päeva demoversiooni [<https://store.codesys.com/codesys-opc-da-server-sl-bundle.html>]



Joonis 2.14 CoDeSys OPC programmi moodulite installeerimine

Kui eelnevalt millegi ei eksitud siis on tarkvaralises kontrolloris määratud muutujad ligipääsetavad teistele Windows keskkonna rakendustele läbi OPC UA protokoll. Ciroso Studio rakenduses mis töötab OPC kliendina saab seejärel valida CoDeSys.OPC.DA serverit.



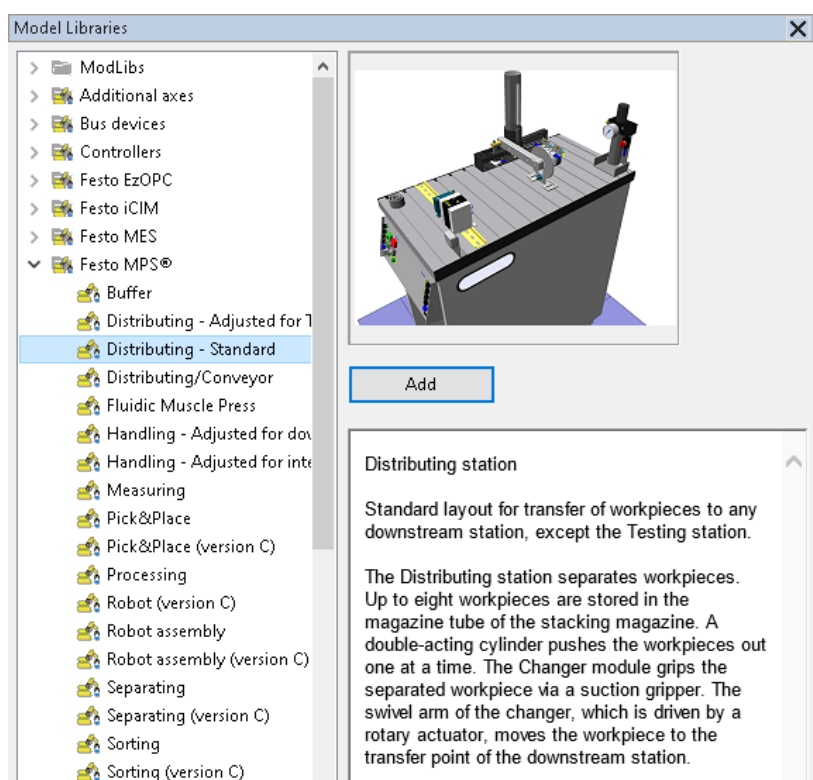
Joonis 2.15 OPC serveri valik Ciroso Studio keskkonna rakenduses

## 2.3 Tarkvaralise kontrolleri OPC-side virtuaalse robotjaamaga

Käivitage Ciro Studio 6.2 ning looge uus MPS (*Modular Production System*) süsteem (File -> New -> MPS System).

Käesolevas näites kasutame robotjaama Festo MPS -> Distributing - Standard.

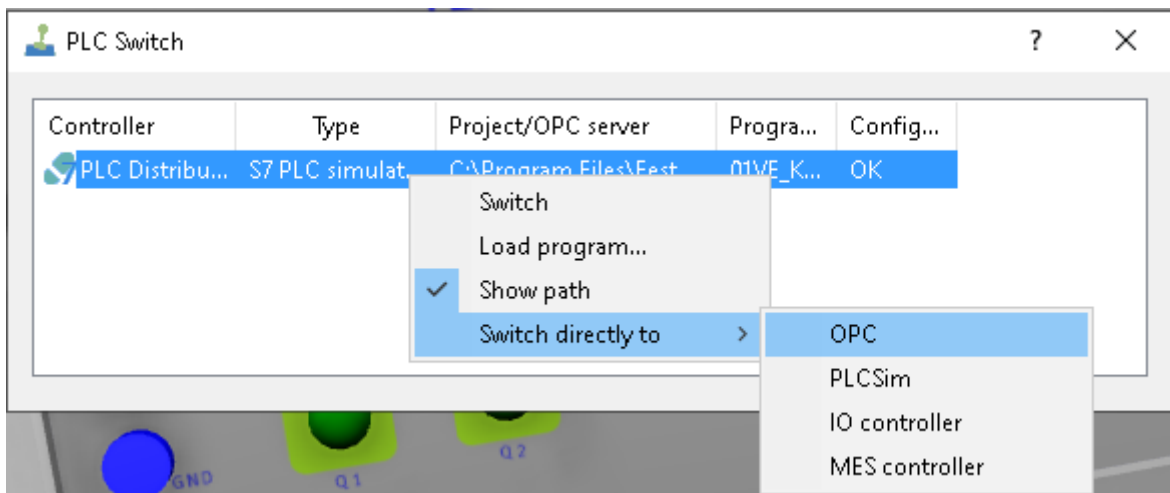
Antud jaotusjaam (*Distribution Station*) on mõeldud detailide eraldamiseks. Kuni kaheksa detaili on jaotusjaama sööturis (*Magazine*) ning kahtepidi liikuv töösilinder lükkab detailid ükshaaval välja. Pöördöla (*Swivel Arm*) küljes olev vaakumhaarats (*Vacuum Gripper*) haarab detaili ja tõstab selle järgmisele tööjaamale.



Joonis 2.16 Robotjaama valik

Ciros Studio keskkonnas valik PLC (*Programmable Logic Controller*) Switch võimaldab valida ja vahetada erinevaid programmeeritavaid kontrollereid eelnevalt koostatud visuaalsete objektide juhtimiseks.

Valige menüüribal Modeling -> PLC Switch. Tehes kontrollerial paremklõpsu ning liikudes Switch directly to peale saate valikud.



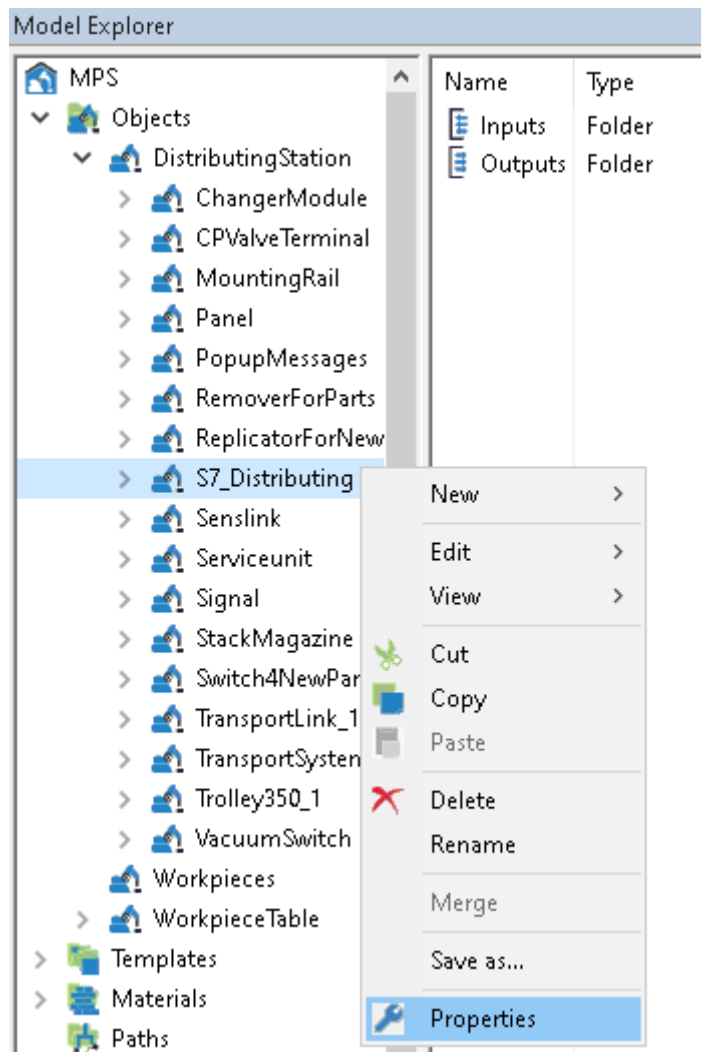
Joonis 2.17 Erinevate kontrolleri valikud

Hetkel on aktiivne Internal PLC, kuid valida on veel võimalik OPC, PLCSim, IO controller või MES controller.

- Internal PLC on Cirose enda sisemine loogikakontroller, mis on üles ehitatud Step7 kontrolleriile.
- OPC (*Open Platform Communications*) kasutab ühist ühendamisprotokolli erinevate tootjate vahel. Võimalik kasutada näiteks CoDeSys'is loodud programme.
- PLCSim on Step7 simuleerimiskontroller, mille programm on tarvis seadistada TIA Portalis.
- IO controller on lihtne sisendite ja väljundite kontroller.
- MES (*Manufacturing execution system*) controllerit kasutatakse tootmisüksustes jälgimaks ja dokumenteerimaks detailide tööprotsessi algtootest lõpp-produktini.

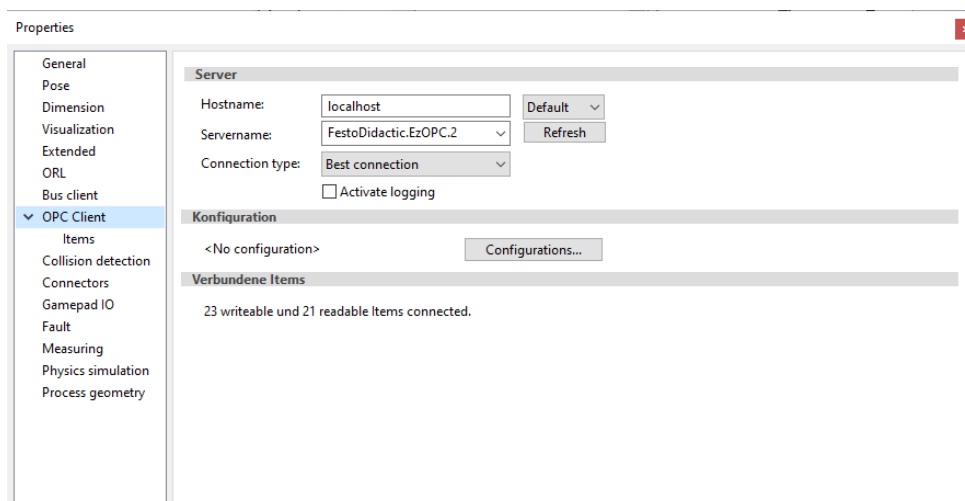
Eelnevalt kirjeldatud juhtimisprogramm töötab samas arvutis ja samas Windows operatsioonisüsteemis CoDeSys tarkvaralisel kontrolleriil. Seepärast kasutame loodud visuaalsete objektide juhtimiseks ja kontrollimiseks samas operatsioonisüsteemis töötavat tarkvaralist PLCd läbi OPC-protokolli.

Avage Model Explorer, valige Objects -> Sorting Station, tuvastage S7\_Distributing ning klõpsake avanenud menüüs jaotisel „Properties“.



Joonis 2.18 Robotjaama kontrolleri sätete valik

Liikuge valikus edasi OPC client peale, vajutage refresh ning valige CoDeSyS.OPC.DA.



**Server**

Hostname:   ▾

Servername:  ▾

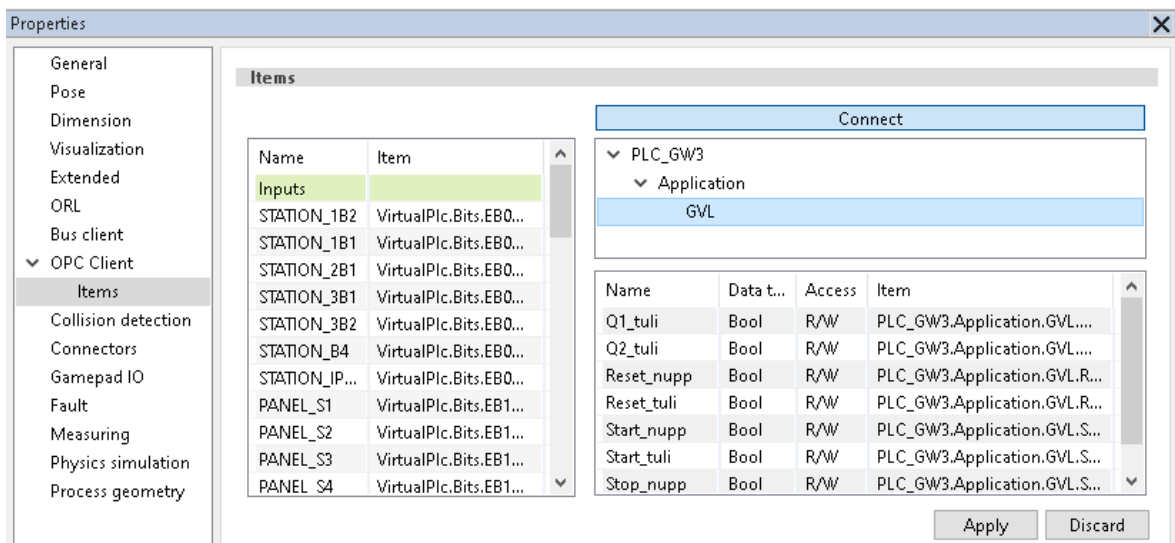
Connection type:  ▾

Activate logging

Joonis 2.19 OPC serveri valik

Liikuge OPC client alakategooria Items peale, vajutage Connect.

Avage tekkinud valiku PLC\_GW3 alamkategooria GVL ehk loodud muutujate nimekiri.

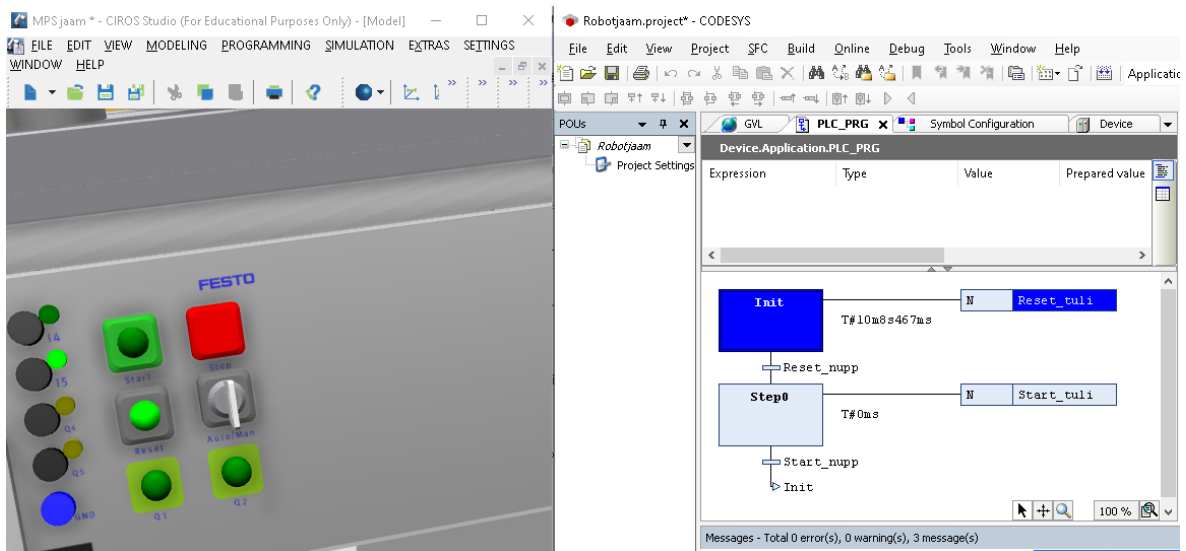


Joonis 2.20 Muutujate nimekiri ning OPC kliendi I/O tabel

1. Kustutage kõik I/O tabeli väärtused, märgistades need ära, tehes parem hiireklõps ja valides Remove Assignment.
2. Paneeli sisenditesse lisage Panel\_S1 = Start\_nupp, Panel\_S2 = Stop\_nupp, Panel\_S4 = Reset\_nupp.
3. Paneeli väljunditesse lisage Panel\_P1 = Start\_tuli, Panel\_P2 = Reset\_tuli, Panel\_P3 = Q1\_tuli, Panel\_P4 = Q2\_tuli.

Peale sisendite ning väljundite lisamist tuleb kindlasti vajutada nupule „Apply“.

Kui simulatsiooni käivitamise järel kõik õigesti toimib, siis peaks CiroS Studios ja CoDeSys keskkonnas olema süsteem piltide kujutatud olekus.



Joonis 2.21 Simulatsiooni ning virtuaalse robotjaama liidestus

Vajutades Reset nuppu, süttib Start nupu signaallamp. Vajutades Start nuppu süttib signaallamp nupul „Reset“.

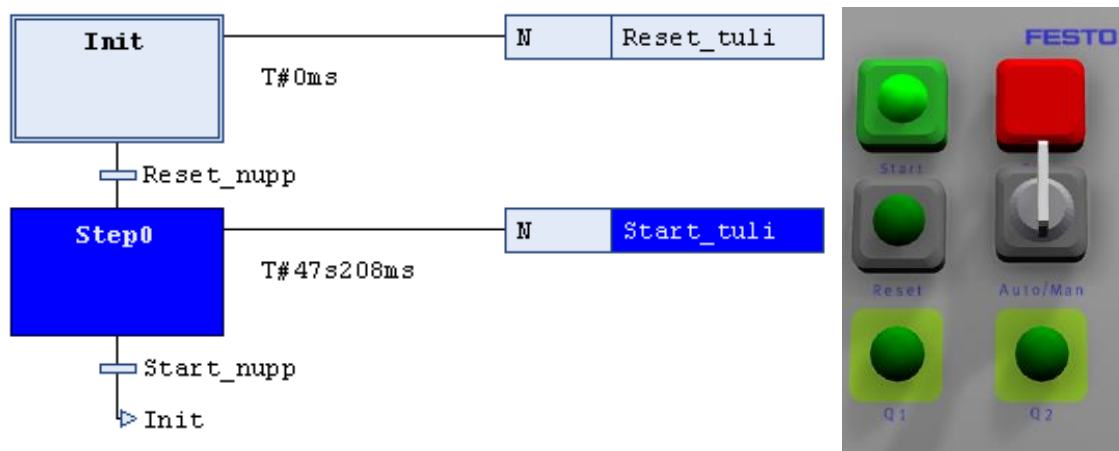
Tähtsamad märkused:

1. OPC server ja CiroS Studio peavad olema käivitatud samas arvutis.
2. Kasutada arusaadavate nimedega muutujaid liidestamise lihtsustamiseks.

OPC server ja kontrolleri või muu seade peavad olema rakenduse katsetamise ajaks ühenduses ning ühendus peab töötama. Muul juhul ei toimi muutujate väärtuse vahetus CiroS Studio ja OPC serveri vahel.

Eltoodud näite kontrollimiseks on võimalik kasutada MatrikonOPC'd, mille puhul on võimalik kontrollida OPC serveri muutujate väärtuseid.

Kui süsteem on käivitatud, ning Start signaallamp põleb (vt. Joonis 2.22), on MatrikonOPC Tag'idest näha, et PLC\_GW3.Application.GVL.Start\_tuli väärtus on True (vt. Joonis 2.23).



Joonis 2.22 CoDeSys ja CiroS Studio liidestuse näide, kui Start signaallamp on sisse lülitatud

MatrikonOPC Explorer - [Untitled\*]

File Server Group Item View Help

\$Group\_1

Localhost \MEEELIKA

- CoDeSys.OPC.02
- CoDeSys.OPC.AE.1
- CoDeSys.OPC.DA
- \$Group\_1
  - PestoDidactic.E5OPC.1
  - PestoDidactic.E4OPC.2
  - Matrikon.OPC.Modbus.1
  - Matrikon.OPC.Simulation.1
  - National Instruments.LookoutOPCServer.1
  - RIF.CrossOPCServer.1
- Network Neighborhood
- Other Network Computers

Contents of '\$Group\_1'

Item ID	Access Path	Value	Quality	Timestamp	Status
PLC_GW3.Application.GVL.Start_tuul		True	Good, non-specific	05.16.2020 10:42:56.755 AM	Active
PLC_GW3.Application.GVL.Start_nupp		False	Good, non-specific	05.16.2020 10:42:56.755 AM	Active
PLC_GW3.Application.GVL.Reset_tuul		False	Good, non-specific	05.16.2020 10:42:55.719 AM	Active
PLC_GW3.Application.GVL.Reset_nupp		False	Good, non-specific	05.16.2020 10:42:56.755 AM	Active

Server Info

Server: CoDeSys.OPC.DA

Connected: Yes  
 State: Running  
 Groups: 1  
 Total Items: 4  
 Current Local Time: 05.16.2020 10:43:14.674 AM  
 Update Local Time: 05.16.2020 10:42:56.895 AM

Did you know?  
 Explorer Tip #4  
 OPC Explorer can filter your browsable OPC items.  
 Click For Details.

MatrikonOPC

Group Info

Group: \$Group\_1

Connected (Async I/O): Yes (2.0)

Active: Yes  
 Items: 4  
 Current Update Rate: 1000 ms  
 Percent Deadband: 0.00%  
 Data Change Rate: 0.37 Items/Sec

Joonis 2.23 MatrikonOPC kaudu signaallampide ja nuppude väärtuste kontroll



## 2.4 Kontrolleri side virtuaalse robotjaamaga läbi andmeside-protokolli MODBUS TCP

Andmesideprotokoll MODBUS on tööstuslikel programmeeritavates kontrollerites, robotsüsteemides ja masinnägemise moodulites väga levinud. Seetõttu anname lühiülevaate selle sidumise võimalustest keskkonnas Ciroso Studio töötava 3D-rakendusega.

Andmeside edukaks toimimiseks on vajalik teha järgnevad sammud:

- 1) IP-aadresside ja alamvõrkude seadistamine kontrollerial ja selle seos arvuti võrgukaardi seadistuste ja OPC serveriga
- 2) OPC-serveri Modbus draiveri seadistamine (sh Modbus Device ID-de jms.)
- 3) OPC-märgendite (tag) seadistamine OPC serveris
- 4) OPC-kliendiks on meil Ciroso. OPC töövõimelisust saab kontrollida OPC-märgendite vaatamise programmiga MatrikonOPC ModBus OPC Server [10].

Kuna klient-tarkvara ühendub OPC serveriga, tuleb tarkvara valides pöörata tähelepanu, et klient võimaldaks OPC ühendust. Tehnilisi raskusi niisuguste ühenduste seadistamisel võib olla mitmesuguseid. Asjaolu et OPC-serveris määratud muutujate nimed on klientrakenduses nähtavad ei pruugi veel tähendada et ka edastusprotokollid omavahel sobivad ja andmed ka korrektselt kohale jõuavad. OPC-server ei ole üldiselt andmeside lõpp-punkt, seetõttu on tarvis täieliku ühilduvuse hindamiseks veenduda, et OPC klientprogramm ühildub ka OPC serveriga.

## KOKKUVÕTE

Üha enam leiab tööstusautomaatikas ning mujal kasutust virtuaalreaalsus. Kasutusvaldkondi on mitmeid. Üheks selliseks on ka tarkvara katsetamine arendusprotsessis ja süsteemi töö visualiseerimine. Luues planeeritava süsteemiosa digitaalse mudeli, annab see võimaluse arendada tarkvara väiksema riskiga. See aitab vältida ekslike prototüüpide väljaehitamist. Selliselt säästetakse nii aega kui raha. Siinkohal on võimalik ka hilisemaid süsteemi muudatusi eelnevalt katsetada.

Tallinna Tehnikaülikooli tootmise automatiseerimise ja robotitehnika laboratooriumites on olemas mitmeid tööstusautomaatika modelleerimisel kasutatavaid tarkvarapakette (näiteks Ciro Studio), mida kasutatakse ka reaalsete tööstuses kasutatavate tootmiseseadmete modelleerimisel. Antud tarkvarapaketi rakendamist praktikas raskendab asjaolu, et on vähe asjakohaseid ja asjakohaseid juhendeid ning mõnede teemade kohta juhendid puuduvad.

Lõputöö raames koostas autor mitmeid juhendeid algajatele, kes puutuvad kokku robotite ja robotjaamade 3D modelleerimisega, nende robotite programmeerimisega ning tarkvaraliste programmeeritavate kontrollritega simulatsioonikeskkonna signaalide juhtimisega.

Ilma juhenditeta on algajal keeruline ning ajakulukas 3D modelleerimise paketi põhifunktsioone selgeks saada. Juhendid on mõeldud olema abimaterjalideks robotitehnika ning tootmise automatiseerimise alaste õppeainete praktilises osas aga nendest võib olla abi ka tootmisettevõtete inseneridel.

Töö esimeses peatükis uuriti tehnilisi süsteeme ja töötati läbi kirjandusallikaid 3D modelleerimisest ning robotite programmeerimisest. Töö teises peatükis uuriti simulatsioonikeskkonna andmesidefunktsioone sh OPC-protokolli ja programmeeritavate kontrollrite rakendusi. Mõlema osa kohta valmisid nii kirjalikud kui ka videojuhendid.

Võib öelda, et suurel määral sai lõputöö eesmärk täidetud. Autori poolt koostatud juhendid koos lühivideotega võimaldavad nende kasutajatel jõuda samm-sammult sama tulemuseni, andes edasi antud tarkvara- ja simuleerimiskeskondade kasutuskogemuse. Juhendi testimine tudengite õpetamisel oleks võimalik edasiarendus lõputöö teemale. Samuti saaks 3D simulatsioonikeskkonda rakendada muudes valdkondades, nt valgustuse juhtimisel.

Autor jõudis lõputööd koostades järeldusele, et täpse, kuid lihtsustatud juhendi koostamine ei ole lihtne. Tähelepanu tuleb pöörata paljudele nüanssidele, et juhend oleks algajatele kasutamiseks sobilik. Lisaks nüanssidele tekkis autoril algselt videojuhendeid koostades mitmel juhul raskusi

tulemuste korratavusega, mis olid osaliselt seotud töö koostamiseks kasutatava arvuti nõrgemapoolsete parameetritega.

Tuleviku perspektiivi silmas pidades võib luua detailsemad juhendid edasijõudnutele, kus ühe võimalusena käsitletakse tarkvaraliste programmeeritavate kontrolleriite liidestamist päris kontrolleriite ja/või robotjaamadega.

## SUMMARY

The use of Virtual Reality is becoming more and more popular in industrial automation and other fields. It has many uses, one of them being the testing of software in development processes and visualisation of the system in action. When creating a digital model of a planned part of the system, virtual reality gives the opportunity to develop the software with lower risks. It helps to avoid building unnecessary prototypes, saving both time and money.

The industrial automation and robotics laboratories of Tallinn University of Technology have many software packages for modelling industrial automation. One of them is Ciro's Studio and all of them are used in actual industries when modelling manufacturing lines and appliances. The usage of Ciro's Studio in Tallinn University of Technology for students is complicated due to the lack of usable, up-to-date guides.

With this thesis the author created guides for beginners, who are obtaining knowledge of 3D modelling of robots and robot stations, programming these robots and controlling the simulation environment with the signals from a virtual controller.

Creating guides was chosen as the thesis subject due to the fact that without any guides it is very complicated and time-consuming for a beginner to learn the main functions of the programs. These guides are mainly created for future use as a teaching aid for the robotics and industrial automation students in Tallinn University of Technology.

The author feels confident to say that the thesis goal is achieved. This lies on the fact that the guides are created with step-by-step videos, that assist the user to reach the same result, giving them a better user experience in software and simulation environments. Further development of the thesis would be to use it for teaching new students.

While making this thesis, the author came to a conclusion, that creating a precise, but simplified guide is not an easy task. For the guides to be suitable for beginners, many things have to be noted.

As a future perspective it seems reasonable to create more detailed guides for an advanced level, where as one opportunity would be how to create a virtual controller interface with actual controllers and/or robot stations.

## KASUTATUD KIRJANDUS

1. Robotitehnika õpetamise meetodid ja vahendid Mitsubishi ja Festo seadmete näitel / Margus Müür, Õppematerjal, Tallinna Tehnikaülikooli elektriajamite ja jõuelektronika instituut, Tallinn, 2009  
[WWW]  
[http://www.ene.ttu.ee/elektriajamid/oppeinfo/materjal/AAR0070/Robotitehnika\\_opetamine\\_TA\\_laboris\\_2009.pdf](http://www.ene.ttu.ee/elektriajamid/oppeinfo/materjal/AAR0070/Robotitehnika_opetamine_TA_laboris_2009.pdf) (18.05.2020)
2. Mitsubishi RV-2AJ Industrial Robot Programming and Calibration Lab Notes, School of Computing, Communication and Electronics, The University of Plymouth, [WWW]  
[http://worldcolleges.info/sites/default/files/enggnotes/robot\\_programming.pdf](http://worldcolleges.info/sites/default/files/enggnotes/robot_programming.pdf) (18.05.2020)
3. CR1/CR2/CR3/CR4/CR7/CR8/CR9 controller instruction manual - Detailed explanations of functions and operations, BFP-A5992-P, Mitsubishi Electric Corporation, 10/2009
4. Linear axis DGE-ZR/-SP operating instructions, FESTO, 80751572017-07g[8075159], Festo 2017-07g [WWW]  
[https://www.festo.com/net/SupportPortal/Files/468225/DGE-ZR\\_-SP\\_2017-07g\\_8075159g1.pdf](https://www.festo.com/net/SupportPortal/Files/468225/DGE-ZR_-SP_2017-07g_8075159g1.pdf) (18.05.2020)
5. Passive guide axes FDG, without drive [WWW]  
[https://www.festo.com/cat/en-gb\\_gb/data/doc\\_ENUS/PDF/US/FDG\\_ENUS.PDF](https://www.festo.com/cat/en-gb_gb/data/doc_ENUS/PDF/US/FDG_ENUS.PDF) (18.05.2020)
6. MELFA Industrial robots Installation description RV-1A/2AJRP-1AH/3AH/5AH, Version A, MITSUBISHI ELECTRIC INDUSTRIAL AUTOMATION, 30/03/2010
7. Mitsubishi roboti juhtseadme CR1 kasutusjuhend (Mitsubishi CR1 Controller Operation Manual, BFP-A8054-J)
8. Pt4 COSIMIR PROFESSIONAL – Pt4\_COSIMIR\_Professional.pdf [WWW]  
[http://www.ene.ttu.ee/elektriajamid/oppeinfo/materjal/AAR0040/Pt4\\_COSIMIR\\_Professional.pdf](http://www.ene.ttu.ee/elektriajamid/oppeinfo/materjal/AAR0040/Pt4_COSIMIR_Professional.pdf) (15.04.2020)
9. CODESYS Store – CODESYS Development System V3 [WWW]  
<https://store.codesys.com/codesys.html> (18.05.2020)
10. Modbus OPC Server for Modbus devices [WWW]  
<https://www.matrikonopc.com/opc-drivers/opc-modbus/base-driver-details.aspx> , (18.05.2020)

## LISAD

### Lisa 1 Videod näidetega simulatsioonipaketi kasutamise kohta

Töö juurde on lisatud andmekandja videote ja subtiitrifailidega simulatsiooninäidete kohta:

1. „Juhend1\_Robotjaama\_Komponentide\_3D-modelleerimine.mp4“ , 6569 KB
2. „Juhend1\_Robotjaama\_Komponentide\_3D-modelleerimine\_EST.srt“ , 4 KB
3. „Juhend1\_Robotjaama\_Komponentide\_3D-modelleerimine\_ENG.srt“ , 4 KB
4. „Juhend2\_Virtuaalse\_robotjaama\_juhtimine\_programmeeritavate\_kontrolleritega.mp4“ ,  
5601 KB
5. „Juhend2\_Virtuaalse\_robotjaama\_juhtimine\_programmeeritavate\_kontrolleritega\_EST.srt“ ,  
3 KB
6. „Juhend2\_Virtuaalse\_robotjaama\_juhtimine\_programmeeritavate\_kontrolleritega\_ENG.srt“ ,  
3 KB