



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Petri Rinta-Jaskari

TEOLLISUUSROBOTIN ETÄOHJEL-
MOINTIYMPÄRISTÖN KEHITYS VISU-
AL COMPONENTS 4.0 -OHJELMALLA

Tekniikka
2018

TIIVISTELMÄ

Tekijä	Petri Rinta-Jaskari
Opinnäytetyön nimi	Teollisuusrobotin etäympäristön kehitys Visual Components 4.0 -ohjelmalla
Vuosi	2018
Kieli	suomi
Sivumäärä	32+ 4 liitettä
Ohjaaja	Mika Billing

Opinnäytetyön aiheena oli tutkia Visual Components 4.0 -robottiohjelman ominaisuuksia ja kehittää etäohjelmointiympäristö Vaasan ammattikorkeakoulun konelaboratorion valmiista teollisuusrobottisolusta. Tehtävänä oli luoda esimerkki robottiohjelmaa Visual Components 4.0 -ohjelmalla ja testata niiden toimivuus todellisella robotilla.

Aluksi perehdyttiin ohjelmiston ominaisuuksiin, luotiin testietäympäristösolu ja testirobottiohjelma, jotta nähtäisiin luodun ohjelman yhteensopivuus todellisen robotin ohjelmiston kanssa. Varsinaiseen työtehtävään etäohjelmointiympäristöjen ja robottiohjelmien lisäksi suunniteltiin ja valmistettiin työkomponentit, joiden avulla selvitettiin Visual Components 4.0 -ohjelman käyttöä erilaisten robottiohjelmien suunnitteluun.

Valmiiksi saatujen robottiohjelmien testaustulokset todellisella robotilla osoittivat Visual Components 4.0 -ohjelman käytettävyyden robottiohjelmana. Ohjelmiston käyttö robotiikan opetuksessa on hyödyksi Vaasan ammattikorkeakoululle. Lisäksi ohjelmiston käyttö voisi tarjota yhteistyötä muiden oppilaitosten sekä alan yritysten kanssa tulevaisuudessa.

ABSTRACT

Author	Petri Rinta-Jaskari
Title	Development of Offline Programming Environment for an Industrial Robot with Visual Components 4.0
Year	2018
Language	Finnish
Pages	32 + 4 Appendices
Name of Supervisor	Mika Billing

The subject of this thesis was to study the features of the Visual Components 4.0 robotics and to develop an Offline Programming Environment from the completed industrial robot block at the mechanical engineering laboratory of Vaasa Universal of Applied Sciences. The task was to create an example of robotic programs with Visual Components 4.0 and test their functionality on the real robot.

Initially the features of the software were familiarised with, a test Offline Programming Environment and a test robot program were created to see the compatibility of the created program with the actual robot software. In addition to Offline Programming Environments and robot programs, the actual task was to design and manufacture work components to help determine the use of Visual Components 4.0 for designing various robotic programs.

The testing results of pre-made robotic programs with real robot demonstrated the usability of Visual Components 4.0 as a robot program. The use of the software in robotics teaching is useful for Vaasa Universal of Applied Sciences. In addition, the use of software could provide co-operation with other educational institutions and companies in the field in the future.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO.....	8
1.1	Työn tavoitteet	8
1.2	VAMK Oy ja Technobothnia.....	8
2	TEOLLISUUSROBOTIIKKA	11
2.1	Yleistä	11
2.2	Robottityypit	11
3	ROBOTIN OHJELMOINTI.....	14
3.1	Yleistä	14
3.2	Online-ohjelmointi	14
3.3	Offline-ohjelmointi	15
4	VISUAL COMPONENTS 4.0	17
4.1	Yritys.....	17
4.2	Ohjelmisto.....	17
5	TYÖN LÄHTÖKOHTA	18
6	TYÖN TOTEUTUS	19
6.1	Tutustuminen ohjelmistoon	19
6.2	Solun rakennus ja kalibrointi	20
6.3	Ohjelmointi ja verifiointi	20
6.4	Ohjelman testaus robotilla	22
6.5	Harjoitustöiden luonti ja testaus.....	23
7	YHTEENVETO	31
	LÄHTEET.....	32
	LIITTEET	

KUVALUETTELO

Kuva 1. Vaasan ammattikorkeakoulun Palosaaren kampusrakennus.	9
Kuva 2. Opetus- ja tutkimuslaboratorio Technibothnia.	10
Kuva 3. Yleisimpien robottityyppien rakenne, kinematiikka ja työaluekuvia.	13
Kuva 4. ABB Yumi ja Universal Robots-mallit UR3, UR5 ja UR10.	15
Kuva 5. Visual Components Premium 4.0 -aloitusnäkökuvana.	17
Kuva 6. 3D-Automate 2014 ja Visual Components Premium 4.0 -ohjelmien toimintavalikot.	19
Kuva 7. Help -välilehti.	20
Kuva 8. Esimerkinäyte Visual Components robottiohjelmasta.	21
Kuva 9. Program-välilehti.	22
Kuva 10. Export-toiminto.	22
Kuva 11. ABB IRB-120 -robottisolu.	23
Kuva 12. ABB IRB-120 -robotin monitoimityökalu.	23
Kuva 13. ABB IRB-120 -robottia ohjaava tietokone.	24
Kuva 14. Nimmariharjoitustyön robottisolu.	25
Kuva 15. Monitoimityökalun t_pencil-työkalupiste.	25
Kuva 16. Onnistunut nimikirjoitus.	26
Kuva 17. Nelikulmiosiirtokappaleita.	27
Kuva 18. Siirtokappaleiden reikäalusta.	27
Kuva 19. Siirto_2 ohjelmassa käytetyt reikäalustat.	27
Kuva 20. Monitoimityökalun t_Imukuppi-työkalupiste.	28
Kuva 21. Siirto_1-ohjelman robottisolu.	29
Kuva 22. Siirto_2-ohjelman robottisolu.	29

LIITELUETTELO

LIITE 1. Visual Components Premium 4.0 -käyttöohje.

LIITE 2. Harjoitusohjelman 1 robottiohjelmakoodi.

LIITE 3. Harjoitusohjelman 2 robottiohjelmakoodi.

LIITE 4. Harjoitusohjelman 3 robottiohjelmakoodi.

KÄYTETYT LYHENTEET JA TERMIT

DOF	Vapausaste (Degree Of Freedom).
Etäohjelmointiympäristö	Todellista robottisolua vastaava ympäristö.
CAD	Tietokoneavusteinen suunnittelu (Computer Aided Design).
Layout	Etäohjelmassa rakennettu robottisoluympäristön pohjapiirustus.
STEP	Tiedostoformaatti 3D-malleille (Science and Technology for Environmental Protection).

1 JOHDANTO

1.1 Työn tavoitteet

Työn tavoite on tutustua Visual Components Premium 4.0 -ohjelman toimintaan, luoda etäohjelmointiympäristö Technobothnian konetekniikan laboratorion ABB IRB-120 -robottisolusta ja tehdä yksinkertaiset robottiohjelmat työkappaleen siirtämisestä ja kynällä piirtämisestä. Robottiohjelmien liikeratojen toiminta simuloidaan ja niistä tehdään videotiedostot simuloinnin havainnollistamiseksi. Lopuksi luotu ohjelmakoodi otetaan ulos ja siirretään oikealle robotille testattavaksi. Kaikista edellä mainitusta tehdään työn liitteeksi yksinkertaiset kirjalliset ohjeet Visual Components 4.0 -ohjelman käytöstä.

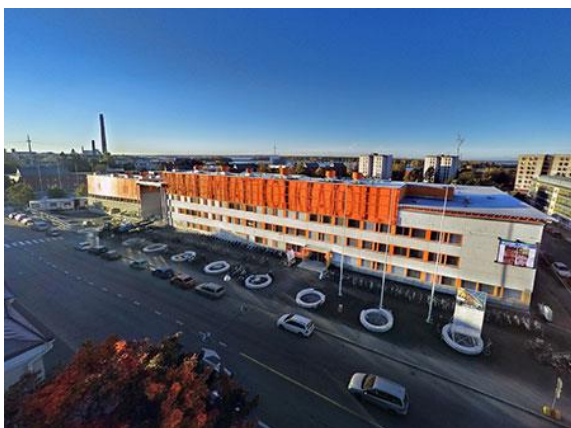
Korkeakoulujen kansainvälisen yhteistyön edistämiseksi luodut robottisolu-merkit voidaan myöhemmin lähettää Vaasan ammattikorkeakoulun yhteistyöoppilaitokselle, jossa alan opiskelijat voivat tehdä ohjelman ja testata etäohjelmointiympäristön toimivuutta.

1.2 VAMK Oy ja Technobothnia

VAMK Oy eli Vaasan ammattikorkeakoulu on vuonna 1996 toimintansa aloittanut ja vuonna 2010 osakeyhtiöksi muutettu Vaasan kaupungin Palosaaren kaupungin osassa sijaitseva kansainvälinen ammattikorkeakoulu (Kuva 1), joka kouluttaa insinöörejä, tradenomeja, sairaanhoitajia, terveydenhoitajia ja sosionomeja suomen ja englannin kielellä niin paikallisen kuin kansainvälisenkin työelämän vaativiin tarpeisiin.

Opiskelijoita on yhteensä 3800, monimuoto-opiskelijat mukaan laskettuna, ja henkilöstöä noin 200. Koulutusaloja ovat tekniikka, liiketalous ja sosiaali- ja terveysala ja suoritettavan tutkinnon kesto, riippuen tutkinnosta, on 3,5 - 4 vuotta tai jopa vain 3 vuotta. Tutkinto koostuu opiskelualan perus- ja ammattiopinnoista, suuntautuvista opinnoista, vapaasti valittavista opinnoista sekä opinnäytetyöstä ja työharjoittelusta. Opinnäytetyön sekä työharjoittelun opiskelijan on usein pyrittävä

vä tekemään opiskelemansa alan vastaavassa yrityksessä, mutta tarpeen tullen, ainakin insinöörien kohdalla, myös ammattikorkeakoulu pystyy tarjoamaan mahdollisuutta niiden suorittamiseen.



Kuva 1. Vaasan ammattikorkeakoulun Palosaaren kampusrakennus.

Opetus- ja tutkimuslaboratorio Technobothnia on perustettu lisäämään paikallista yhteistyötä Vaasan insinöörikoulutuksen ja Vaasan seudun teknologiaan keskittyvien yritysten välillä. Technobothnia on Vaasan ammattikorkeakoulun, Vaasan Yliopiston ja Yrkeshögskolan Novian yhteiskäytössä oleva Vaasan Puuvilla Oy:n entisen tehdasrakennuksen tiloissa sijaitseva laboratoriokokonaisuus, jossa insinööri- sekä myös diplomi-insinööriopiskelijat suorittavat laboratorioharjoituksia kaikista teknillisistä opetusaineista apunaan monipuoliset laboratoriolaitteet ja asianmukaiset laboratoriotilat. Technobothnia (Kuva 2) otettiin käyttöön vuonna 1996 laboratoriotilojen valmistuttua ja jo yli 20 vuoden ajan se on tarjonnut paikalliselle teollisuudelle kuin myös elinkeinoelämälle tutkimus- ja kehittämishankkeiden lisäksi testaus-, mittaus- ja koepalveluita sekä myös koulutusmahdollisuuksia tarpeen mukaan.



Kuva 2. Opetus- ja tutkimuslaboratorio Technobothnia.

2 TEOLLISUUSROBOTIIKKA

Ammattimielessä teollisuusroboteilla tarkoitetaan väsymätöntä työkonetta, joka työskentelee tehtaissa tai varastoissa ihmisen valvonnassa tai ei. Ensimmäiset teollisuusrobotit syntyivät 1960-luvulla ja ovat siitä asti kehittyneet ja jakautuneet eri teollisuusrobottiluokkiin ominaisuuksiensa mukaan.

Erilaisten teollisuusrobottien käyttö on yleistynyt ympäri maailmaa, koska tänä päivänä robotille annetaan ne ihmiselle hyvin yksitoikkoiset sekä pitkäkestoiset, joskus myös nopeutta ja tarkkuuttakin vaativat työtehtävät sekä esim. vaarallisissa tiloissa tapahtuvat työtehtävät mikä säästää ihmiseltä paljon aikaa sekä parantaa tehtävän työn laatua huomattavasti.

2.1 Yleistä

Robottistandardin ISO 8373 mukaan teollisuusrobotti on automaattinen, joko sähköisesti, paineilmalla tai hydraulisesti liikuteltavissa oleva, työkaluja tai siirrettäviä kappaleita käsittelevä, uudelleen ohjelmoitava monitoimilaite, jolla on vähintään 3 kääntyvää vapausastetta (DOF) eli mekaanista niveltä ja, joka uudelleen muutettavan liikeratansa ansiosta, on suunniteltu hyvin erilaisiin yksinkertaisiin työtehtäviin joita tuotantoteollisuudessa riittää. /3, s. 13; 4; 5, s. 259/

2.2 Robottityypit

Suorakulmaisen robotin eli portaalirobotin kolme ensimmäistä vapausastetta ovat suorakulmaisessa koordinaatistossa, mutta siitä käytettävä työkalu, yleensä tarttuja, voi olla nivelöity, mikä lisää robotin työalueen ulottuvuutta. Yksinkertaisen rakenteensa ansiosta portaalirobotti kestää hyvin kuormituksia sekä näiden vaihte-
luja. Portaalirobottia käytetään niin raskaissa logistiikka- ja varastointisovelluksissa kuin myös kevyemmässä työstössä, esimerkiksi työstökoneissa, sekä tänä päivänä yleistyvissä materiaalia lisäävässä valmistus menetelmässä ts. 3D-tulostimien rakenteessa. /4; 5, s.259/

Napakoordinaatistorobottilla on yksi lineaarinivel ja kaksi kiertyvää käsivarren niveltä, jotka muodostavat napakoordinaatiston, näin ollen robotin työskentelyalue voidaan nähdä kokonaisuudessaan lähes pallonmuotoisena (Kuva 3). Napakoordinaatistorobottia käytetään mm. erilaisissa hitsaustyöissä. Nykyään käytetään enemmän kiertyvänivelistä robottia, jonka työalue vastaa napakoordinaatistorobotin omaa. /3/

Kiertyvänivelisellä robotilla eli nivelvarsirobotilla on 4 - 6 vapausastetta eli mekaanista niveltä. Ne ovat yleisimmin teollisuudessa käytettyjä robotteja, koska käyttömahdollisuudet ovat todella laajat niiden joustavuuden, monikäyttöisyyden sekä suuren ulottuvuusalueen ansiosta. Robotin käyttökohteita ovat mm. erilaiset kokoonpano-, pakkaus-, hitsaus- ja maalaustehtävät sekä erilaisten työstökoneiden kanssa työskentely. /4, 5/

Scara-robotti (Selective Compliance Assembly Robot Arm) on nivelvarsirobotin erikoissovellus, jolla on kaksi tai kolme samassa tasossa kiertyvää niveltä ja yksi pystysuuntainen lineaarinivel, ja jota suurien liikenopeuksiensa, hyvän liiketarkkuutensa ja pienen kokonsa ansiosta käytetään mm. elektroniikkateollisuuden kevyissä kokoonpanotehtävissä. /4, 5/

Rinnakkaisrakenteinen robotti on suuria voimia sekä nopeutta ja kestävyyttä omaava teollisuusrobotti, jonka työkalu kolmen lineaarisen nivelen varassa muodostaa robotille suljetun kinemaattisen rakenteen jättäen robotin työalueen kuitenkin varsin pieneksi. Robotin sijoituspaikka onkin pienen työalueen yläpuolella ja siksi sitä käytetään erilaisissa työkohteen yläpuolelta tapahtuvissa, tarkkuutta vaativissa, poiminta- ja työstötehtävissä. /4/

Kuvassa 3 on esitetty kaikkien edellä mainittujen robottityyppien rakenteen, kinematiikan ja työalueen kuvaukset.

Nimitys pääakseleiden mukaan	Rakenne	Kinemaattinen kaavio	Työalue
Suorakulmainen robotti			
Sylinterirobotti			
Napa-koordinaatisto-robotti			
Scara-robotti			
Kiertyvänivelinen robotti			
Rinnakkaisrakenteinen robotti			

Kuva 3. Yleisimpien robottityyppien rakenne-, kinematiikka- ja työaluekuvia. /3/

3 ROBOTIN OHJELMOINTI

3.1 Yleistä

Kiertyvä nivelisen teollisuusrobotin ohjelmointi on robottikäsi­varren liikkeiden toiminnan suorittamista, jossa toteutetaan annettuja liike­käskyjä. Robotin ohjel­moinnin tärkeimmät tehtävät ovat laatia liike­käskyjen toimintajärjestys robotin työkalun liikkeiden toteuttamiseksi, määrittää robotin toiminta mahdollisissa vir­hetilanteissa ja huomioida robottisolun toiminta mahdollisiin ympärille oleviin laitteisiin ts. antaa tarvittaessa signaalitietoja ympärillä sijaitseviin koneisiin ja laitteisiin. Robottiohjelmaan luotujen liikeratojen lisäksi ohjelmoinnissa tulee huomioida käytettävän robotin ominaisuudet, liikerajoitteet sekä työympäristö. Esimerkiksi työympäristön mallinnus auttaa määrittämään robotin törmäysriskit sekä ulottuvuus ja eri työasento mahdollisuudet.

3.2 Online-ohjelmointi

Online-ohjelmointi tarkoittaa sekä johdattamalla ohjelmointia, että opettamalla ohjelmointia. Johdattavassa ohjelmoinnissa vapautetaan robotin liikeakselit ja näin ohjelmoija pystyy lihasvoimin liikuttamaan robottia valitsemaansa liikerataa pitkin ja viemään robotin työkalun haluttuun paikoituspisteeseen. Liikerata lisätään robottiohjaimen muistiin ja pystytään toistamaan jälkeensä. Johdattavan ohjelmoinnin huonoina puolina ovat epätarkat liikeradat sekä niiden että robotin nivelasentojen hankala muunneltavuus. Universal Robots-robotit sekä ABB Yumi-yhteistyörobotti (Kuva 4) ovat tänä päivänä hyviä esimerkkejä roboteista, jotka on suunniteltu johdattamalla ohjelmoitaviksi.



Kuva 4. ABB Yumi ja Universal Robots-mallit UR3, UR5 ja UR10.

Opettamalla ohjelmointi on tavallisesti enemmän käytössä kuin johdattamalla ohjelmointi. Tässä ohjelmointitavassa robottia ohjataan käsiohjaimen avulla ja samalla määritetyt paikoituspisteet tallentuvat ohjaimen muistiin. Määritetyille paikoituspisteille annetaan liikekomennot halutun liikkumistavan ja valitun nopeuden mukaan, jotka auttavat robottia seuraavaan paikoituspisteeseen siirryttäessä. Robotin työkalusta riippuen paikoituspisteissä voidaan antaa myös I/O-käskyjä tarpeen mukaan.

3.3 Offline-ohjelmointi

Offline-ohjelmointi eli etäohjelmointi tarkoittaa robotin ohjelmointia ulkoisen 3D-graafisen tietokoneohjelmiston avulla, jossa robottisolun layout on rakennettu CAD-suunnittelujärjestelmän avulla vastaamaan oikeaa solu ympäristöä, ja joka on tänä päivänä myös yleisin teollisuusrobottien ohjelmointitapa, koska oikeaan robottiin ei tarvitse olla yhteydessä, eikä tuotantoprosessia siksi tarvitse pysäyttää ohjelman muutosten ajaksi. Etäohjelmoinnin toteuttamista varten tarvitaan työkohteesta riippuen riittävän tehokas ohjelmointilaitteisto ja laadukas ohjelmointikieli.

Tämän päivän offline-ohjelmointi on yleensä mallipohjaista ohjelmointia, jossa robotin liikeradat perustuvat luotuihin paikoituspisteisiin kuten edellä mainituissa onlineohjelmointitavoissa. Mallipohjainen ohjelmointijärjestelmä käyttää ohjelmistossa sekä ohjelmiston ulkopuolella mallinnettujen komponenttien 3D-

muototietoa hyväksi ohjelmaa luodessa. Itse roboteista ja käytettävistä oheislaitteista järjestelmä sisältää laajat valmismallikirjastot. Toimivan robottisolun saa helposti rakennettua näiden mallien avulla ja ohjelman saa luotua komponenttimallien geometrisia ominaisuuksia hyödyntäen. Ennen ohjelman luontia robottisolun layout tulee kalibroida eli mitoittaa vastaamaan todellisen robottisolun ympäristöä. Valmiin robottiohjelman toimivuus tarkastetaan simuloimalla sen liikerradan toiminta, mikä on tärkeää mahdollisten ongelmien välttämiseksi. Lopulta luotu robottiohjelma siirretään oikealle robotille, esimerkiksi paikallisverkon välityksellä, jonka jälkeen robotin kotipisteen paikka ja liikerradan toiminta on ensin vielä tarkastettava ilman työkappaletta.

4 VISUAL COMPONENTS 4.0

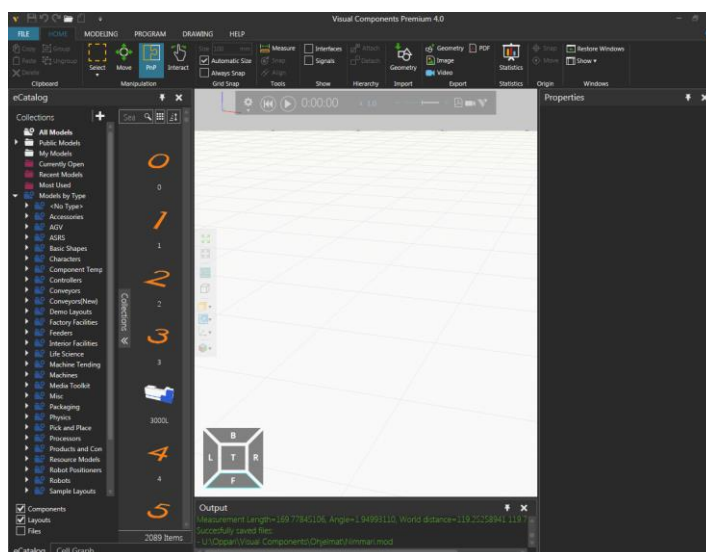
Visual Components 4.0 on uuden ajan valmistussuunnittelu- ja simulointitekniikkaan perustuva 3D-graafinen robottiohjelmisto, jota käyttävät monet maailmanlaajuiset yritykset ja organisaatiot mm. ohjelmiston tehokkuuden, joustavuuden ja skaalattavuuden ansiosta. /8/

4.1 Yritys

Vuonna 1999 toimintansa toiminimenä aloittanut Visual Components Oy on Helsingissä toimiva teollisuuden IT-ohjelmistoihin ja IT-sovelluksiin perehtynyt osakeyhtiö, joka on perustamisestaan asti kehittänyt kuluttajille helppokäyttöisiä ja helposti käyttöönotettavia valmistussuunnittelu- ja simulointitekniikkasovellusohjelmistoja. /6, 7/

4.2 Ohjelmisto

Vuonna 2016 ilmestynyt Visual Components 4.0-ohjelmistoperhe sisältää kolme eri ohjelmaversiota, jotka ovat Essentials, Professional ja Premium. Premium (Kuva 5) on ainoa versio, jossa on etäohjelmointiominaisuus ja pitkälle edistynyt robotiikan soveltaminen. /8/



Kuva 5. Visual Components Premium 4.0 -aloitusnäkömääkkuna.

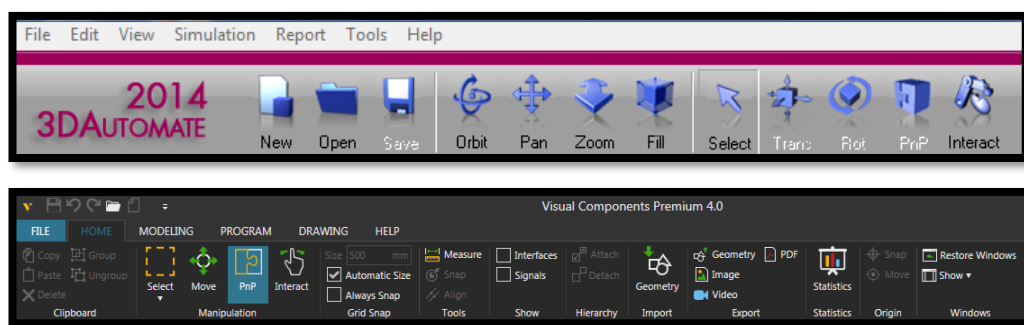
5 TYÖN LÄHTÖKOHTA

Opinnäytetyön alkaessa oli tarkoitus suunnitella Visual Components Premium 4.0 -ohjelmalla robottisolulayout Technobothnian konelaboratoriossa sijaitsevan ABB IRB-120 -teollisuusrobotin solu ympäristöstä, tehdä erilaisia robotin toimintoja suorittavia robottiohjelmia, simuloida ohjelmien toiminta ja korjata tarvittaessa ja siirtää valmiiden ohjelmien ohjelmakoodit oikealle robotille. Ohjelmakoodien siirtämisessä tuli käyttää ABB RobotStudio -ohjelman Rapid-ohjelmointikieltä, joka havaitsisi ohjelmassa tai ohjelman siirrossa satunnaisia ongelmia. Ennen ohjelman ajoa tulisi ensin varmistaa, että robotin työkalun alkuasento niin oikeassa ympäristössä kuin etäympäristössä vastaisivat toisiaan, jos näin ei olisi tulisi etäohjelmassa työkalun asentoa muuttaa ja ohjelmakoodi päivittää. Lopulta kun ongelmia ei enää olisi, niin robottiohjelmien ajot suoritettaisiin ja tarvittaessa niitä vielä muutettaisiin ajon aikana. Muutoksia saattaisivat olla liikekäskyjen joustavuuden muokkaus, jos se auttaisi paremmin liikekäskyn suorittamisessa.

6 TYÖN TOTEUTUS

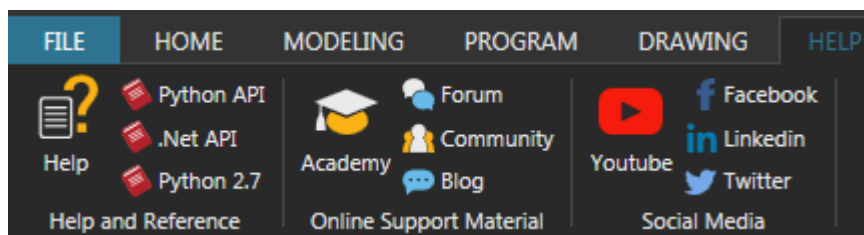
6.1 Tutustuminen ohjelmistoon

Visual Components Premium 4.0 on uudempi versio Visual Components Oy:n 3D-Automate 2014 -ohjelmasta, jota Vaasan ammattikorkeakoulu on myös käyttänyt robotiikan opetusohjelmistona. Näiden kahden ohjelmiston toimintavarauudet sekä robotti- ja oheislaittekirjastot muistuttavat toisiaan, mutta esimerkiksi Visual Components 4.0 -toimintavalikko on lajiteltu viiteen välilehtiin ja jokainen sisältää hieman laajemmat ominaisuudet verrattuna 3D-Automate 2014 -toimintavalikon välilehtiin (Kuva 6).



Kuva 6. 3D-Automate 2014 ja Visual Components Premium 4.0 -ohjelmien toimintavalikot.

Ohjelmistoon tutustuminen alkoi Visual Components Oy:n websivujen videoluentoja seuraamalla ja niissä tapahtunutta matkimalla. Omia robottiohjelmiä tehdessä tarvittiin välillä yksityiskohtaisempaa tietoa, mm. luotujen paikoituspisteiden konfiguraation muuttamisesta yhteensopivaksi todellisen robotin paikoituspisteiden kanssa, jotta työkalun toiminta toteutuisi. Apua löytyi Help -välilehden (Kuva 7) sisältämistä Help -dokumentista, Visual Components Oy:n yhteydenpitopalvelimista sekä yrityksen sosiaalisen median kanavista.



Kuva 7. Help – välilehti.

6.2 Solun rakennus ja kalibrointi

Työn kohteeksi valittu ABB IRB-120 -nivelsiirrobottisolu rakennettiin solun komponenttien jo valmiina olevista STEP-tiedostoista, jotka saatiin ohjaajalta. Ulkopuolelta tuodut STEP-tiedostot saatiin tuotua ohjelmaan vain Import-sisääntulotoiminnolla, josta kerrotaan lisää Visual Components 4.0 -käyttöohjeessa (LIITE 1). Etäympäristöjä luotiin useampi samanlainen eri työstöissä käytettävien komponenttien takia, koska ohjelmien erilaisten liikeratojen toiminta olisi vain sotkenut koko ohjelmointiprosessin.

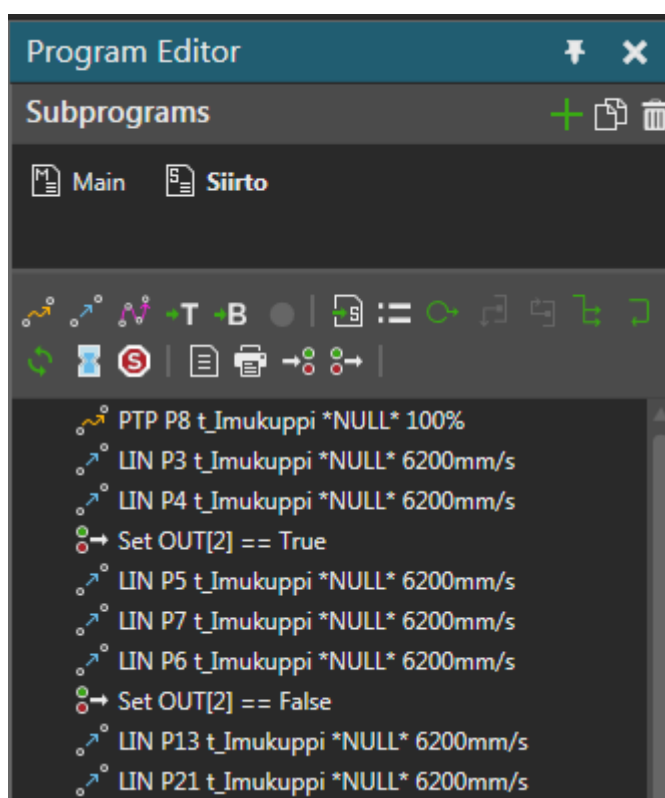
Robottisolun kalibroinnilla tarkoitetaan solun koon ja käytettyjen komponenttien etäisyyksien mitoittamista, jotta solu vastaisi todellista robottisoluympäristöä. Robotin kalibroinnilla taas tarkoitetaan etäympäristön robotin aseman asettamista vastaamaan todellisen robotin asemaa eli nivelten ja työkalun asentoa.

6.3 Ohjelmointi ja verifiointi

Ohjelmointi ja ohjelman muokkaus Visual Components 4.0 -ohjelmalla on helppoa. Ennen ohjelmoinnin aloittamista tuli robotin työkalulle lisätä ja nimetä työkalupiste, jonka mukaan robotti ohjelman suorittaa, tässä tapauksessa robotin työkalulle luotiin kaksi eri työkalupistettä sen vastakkaisille puolille, niitä ei kuitenkaan käytetty samassa robottiohjelmassa.

Vielä ennen ohjelmoinnin aloittamista pohditaan, sijoitetaanko ohjelman liikekäsyt robotin pääohjelmaan (main) vai tehdäänkö aliohjelma, jota kutsutaan pääohjelmassa toiminnolla Call ”aliohjelma 1”. Valitun sijoituspaikka jälkeen ohjel-

mointiosuus voidaan aloittaa esimerkiksi luomalla peräkkäin niin joustavia (joint) kuin suoraviivaisia (line) liikekäskyjä, lisäksi ohjelmalle suunnitellusta toiminnasta ja käytettävästä työkalusta riippuen liikekäskyjen väliin pystyy lisäämään esimerkiksi I/O eli Input/Output -käskytoimintoja (Kuva 8) tai tarvittaessa säädettäviä odotusaikatoimintoja.



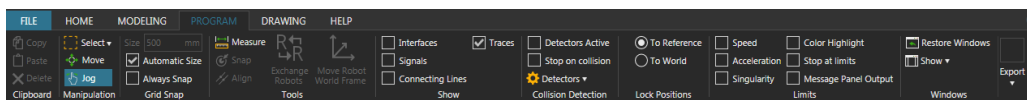
Kuva 8. Esimerkinäyte Visual Components robottiohjelmasta.

Verifiointi tarkoittaa ohjelman toimivuuden tarkastusta ennen oikealle robotille siirtämistä. Tarkastus eli robottiohjelman simulointi toteutetaan ajamalla ohjelman luoto liikerata etäohjelmassa nopeasti läpi, näin huomataan heti ohjelman liikerradan mahdolliset ongelmat joita voivat olla paikoituspisteen asento eli konfiguraatio suhteessa robotin työkalun asentoon ja lopulta koko robotin liian hankalaan työasentoon, jossa robotin akselit ovat liian lähellä ääriasentoa. Siksi etäohjelmassa luotujen paikoituspisteiden konfiguraatiota pystyy muuttamaan joustavamman-

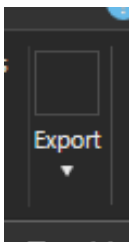
si, jotta liikerata voidaan suorittaa oikean robotin akselien ääriasennoista huolimatta, mutta aina sekään ei välttämättä auta.

6.4 Ohjelman testaus robotilla

Ohjelmien siirtämiseen käytetään Program-väkilehden (Kuva 9) oikean reunan Export-ulosvientitoimintoa (Kuva 10), jonka Post Process -jälkikäsitteilytoiminto muuttaa ohjelmakoodin WordPad-tekstiedostomuotoon, joka on yhteensopiva monien tunnettujen teollisuusrobotivalmistajien kanssa, joita ovat mm. ABB, FANUC ja KUKA. Varsinaisessa ohjelmakoodin siirtämisessä, esimerkiksi ABB robotille, jonka käyttämä RobotStudio-ohjelman Rapid-ohjelmointikieli esittää ohjelmakoodin WordPad-tiedoston kaltaisena, mutta ilmoittaa heti mahdollisista koodivirheistä. Yleisimpiä virheitä ovat saman nimisen työkalun esiintyminen ohjelmakoodissa, joka löytyy Rapid-työkaluluettelosta sekä ohjelmakoodissa esiintyvien I/O-käskyjen toiminnot eli tila-asennot, jotka ovat joko 0 tai 1, voivat olla ristiriidassa Rapidin omien I/O-käskyjen kanssa. Nämä virheet ovat helposti korjattavissa ja ohjelma on valmis testattavaksi robotilla, mutta testiajon aikanakin saattaa esiintyä vielä ongelmatilanteita, jotka kuitenkin pystyy korjaamaan testiajon aikana tai sitten ohjelman toimivuus pitää tarkistaa etäohjelmassa, jos on jäänyt huomaamatta jokin niistä ongelmista jotka mainittiin verifiointikappaleessa.



Kuva 9. Program-välilehti.



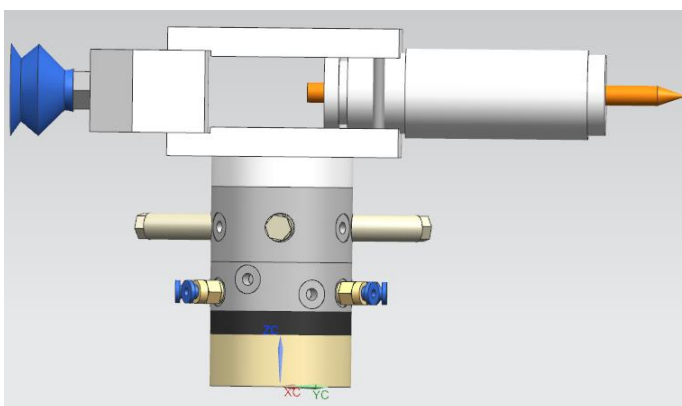
Kuva 10. Export-toiminto.

6.5 Harjoitustöiden luonti ja testaus

Harjoitustöiden aiheita olivat robotin kynätyökalulla nimikirjoituksen kirjoitus ja imukupityökalulla kappaleen siirtäminen paikasta toiseen. Käytettäväksi robotiksi valittiin konelaboratorion ABB IRB-120 -robottisolu (Kuva 11), jonka käyttämä monitoimityökalu (Kuva 12) koostui sekä kynästä että imukupista, ja joka oli huollon takia hetken poissa käytöstä, mikä ei kuitenkaan juuri hidastanut töiden etenemistä.



Kuva 11. ABB IRB-120 -robottisolu.



Kuva 12. ABB IRB-120 -robotin monitoimityökalu.

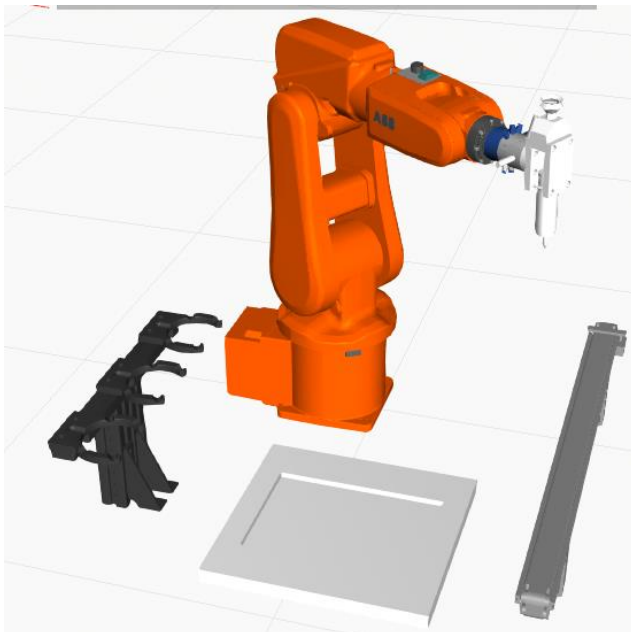
Ennen harjoitustöiden aloittamista molempien periaatetta kokeiltiin useammalla testirobotiohjelmalla, jotta nähtäisiin, miten ne saadaan onnistumaan. Ohjelmat

siirrettiin muistitikulla IRB-120 -robottia ohjaavaan kannettavaan tietokoneeseen (Kuva 13), jossa robottiohjelmaa tarkastettiin ja tarvittaessa muokattiin ABB:n robottien ohjaukseen käytettävän RobotStudio-ohjelman avulla.

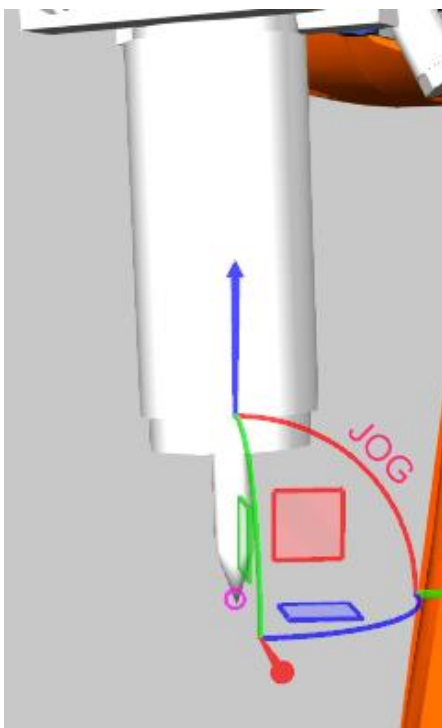


Kuva 13. ABB IRB-120 -robottia ohjaava tietokone.

Ensimmäinen harjoitustyö on nimeltään Nimhari (Kuva 14), jossa tarkoitus oli kirjoittaa nimikirjaimet IRB-120 -robotin kanssa aiemmin käytetylle piirto- ja kirjoituslustalle. Työkaluna käytettiin IRB-120 -robotin monitoimityökalun kynäosaa, jolle tuli luoda työkalupiste kärkeen ja pisteelle annettiin nimeksi t_pencil (Kuva 15). Ohjelman teossa ei ollut ongelmia, mutta ohjelmaa siirrettäessä oikealle robotille RobotStudion kautta, ilmoitti Rapid-ohjelmakieli virhetiedoksi työkalupisteen nimen, koska saman niminen työkalu löytyi myös robotin omista työkaluissa. Tänä virhetieto korjattiin helposti muuttamalla ohjelmakoodin työkalutieto kommentitiedoksi.

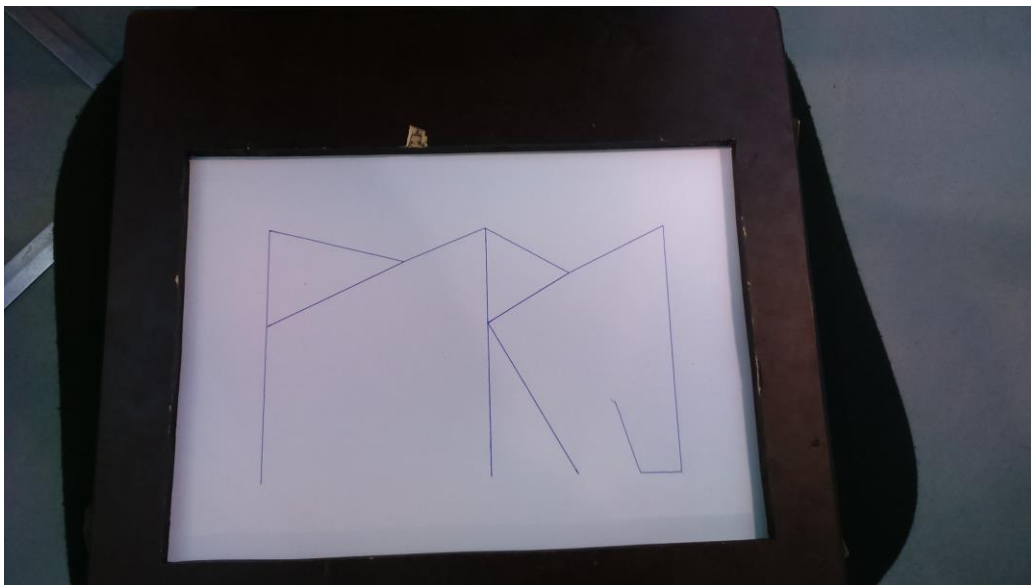


Kuva 14. Nimmariharjoitustyön robottisolu.



Kuva 15. Monitoimityökalun t_pencil-työkalupiste.

Työntulokseksi saatiin onnistunut nimikirjoitus (Kuva 16), joka vastasi etäohjelmassa luotua paikoituspisteiden liikerataa yksityiskohtaisesti, samalla todistettiin Visual Components Premium 4.0 -ohjelman yhteensopivuus Vaasan ammattikorkeakoulun konelaboratorion teollisuusrobottien käyttämien ohjelmien kanssa.



Kuva 16. Onnistunut nimikirjoitus.

Toinen harjoitusohjelma oli nimeltään Siirto, jossa oli tarkoitus viedä nelikulmio (Kuva 17) sille suunniteltuun saman muotoiseen syvennykseen, joka sijaitsi suorakulmaisessa reikäalustassa (Kuva 18), palapelin tapaisesti ja takaisin alkupisteeseen. Nämä työkappaleet ovat akryylilevyä ja ne valmistettiin konelaboratorion laserleikkurilla. Tämän harjoitustyön etäympäristöstä sekä ohjelmasta tehtiin kaksi eri versiota nimeltään Siirto_1 ja Siirto_2, koska ensimmäisen version siirtokappaleet sijaitsivat päällekkäin aloituspisteissään ja olivat ympäriltään tukematomat ja siirryttäessä seuraavaan pisteeseen kappaleen asento ei välttämättä ollut kohtisuorassa niin kuin oli suunniteltu. Toisessa versiossa nelikulmioiden alkupisteet oli sijoitettu toisen reikäalustan (Kuva 19) syvennyksiin, josta ne siirrettiin toisiin syvennyksiin ja takaisin. Siirtotyökaluna käytettiin ABB IRB-120 -robotin monitoimityökalun imukuppiosaa, jolle nimettiin t_Imukuppi-niminen työkalupiste (Kuva 20).



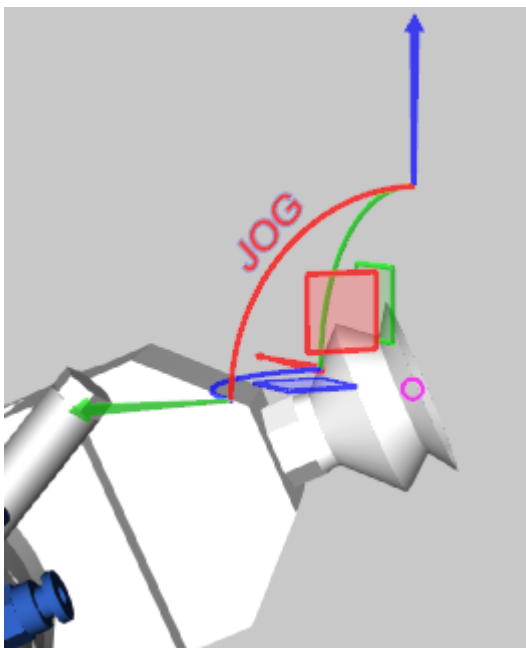
Kuva 17. Nelikulmiosiirtokappaleita.



Kuva 18. Siirtokappaleiden reikäalusta.

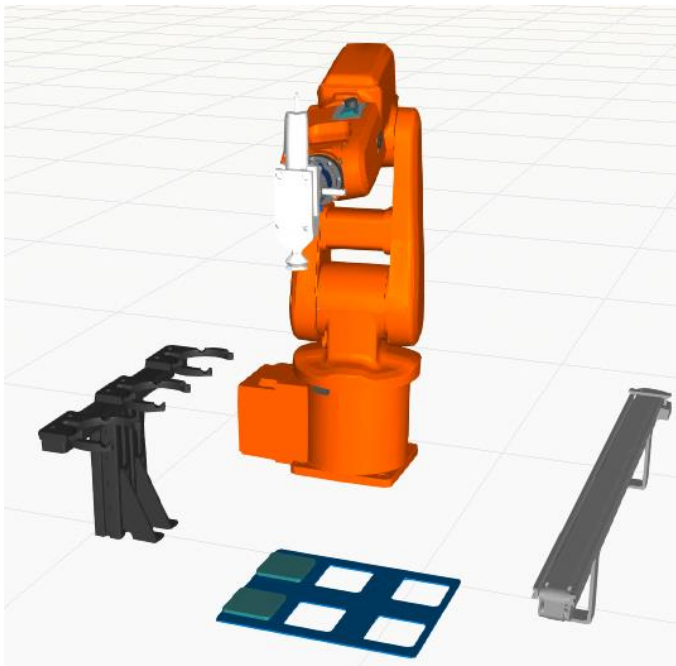


Kuva 19. Siirto_2-ohjelmassa käytetyt reikäalustat.

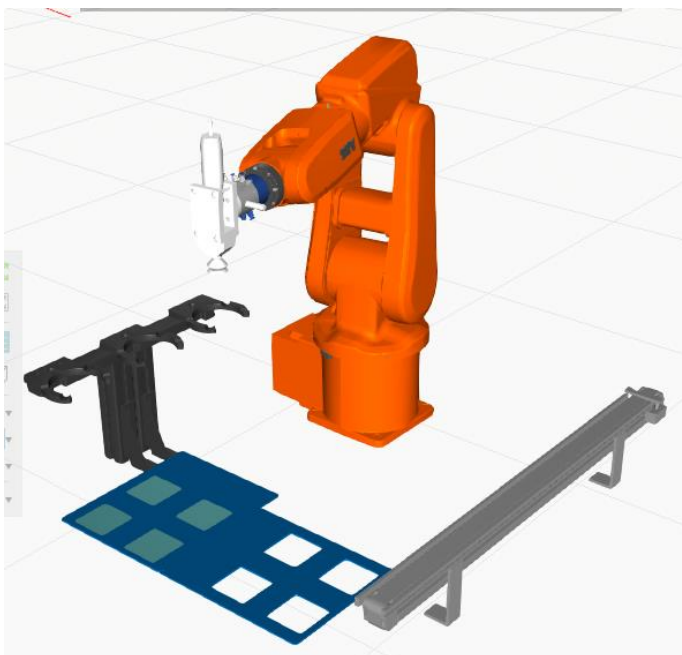


Kuva 20. Monitoimityökalun t_Imukuppi-työkalupiste.

Molempien siirto-ohjelmaversioiden periaate oli sama ja siksi Siirto_1-ohjelman (Kuva 21) liikeradat olivatkin suurimmilta osin kopioitu Siirto_2-ohjelmaan (Kuva 22), tämä aiheutti myös ongelmia, jotka myös näkyivät molemmissa ohjelmissa. Ongelmaksi muodostui käytetyn robotin 4. ja 6. akselien ääriasennot työkierron aikana, mikä myös näkyi etäohjelmointiympäristössä, mutta ei vaikuttanut ohjelmien työkierron suorittamiseen. Oikealla robotilla kuitenkin siirtyminen seuraavaan paikoituspisteeseen ei suoraan onnistunut ääriasentojen takia, vaan ohjelman suorittaminen onnistui vain siirtymällä piste kerrallaan seuraavaan eli askelittain liikkuen. Siirto-ohjelmien testaus ei siis aivan onnistunut oikealla robotilla mutta näiden ohjelmien periaate oli kuitenkin mahdollista toteuttaa.



Kuva 21. Siirto_1-ohjelman robottisolu.



Kuva 22. Siirto_2-ohjelman robottisolu.

Kaikkien harjoitustöiden simuloinneista etäympäristöissä on lisäksi luotu videotiedostot, joista ulkopuolisetkin helposti näkevät ja ymmärtäisivät robotin tekemän työkierron toimintaperiaatteen. Tällaiset videot voisivat tarjota Vaasan ammattikorkeakoululle myös yhteistyömahdollisuuksia paikallisen teollisuusyritysten ja toisten sekä kansainvälisten korkeakoulujen kanssa.

7 YHTEENVETO

Työn ensimmäisenä ajatuksena oli luoda robottiohjelma, joka lähetettäisiin yhteistyöoppilaiselle Saksaan, mutta tämä osuus jätettiin pois suunniteltujen aikataulujen vuoksi, koska tehtävä työ oli tärkeämpi. Visual Components Premium 4.0 tarjosi aiheena uuden ja mielenkiintoisen työkokonaisuuden ja antoi uusia näkemyksiä mihin teollisuusrobotiikka on menossa. Vaikka työn tavoitteet robottiohjelmien osalta ei täysin toteutunut, koska niitä joutui muokkaamaan myös testausvaiheessa, voidaan kuitenkin olla tyytyväisiä siihen mitä käytetyn ohjelmiston ominaisuuksista ja toiminnoista saatiin selville.

Jos jotain olisi vielä voinut tehdä toisin siirto-ohjelmien suhteen, niin työkappaleiden korkeampi sijainti robottiin nähden olisi varmasti toiminut, mutta helpompaa olisi lopulta ollut vain luoda uusi siirto-ohjelma, jossa heti olisi pyritty korjaamaan akselien liian hankalat asennot.

Loppujen lopuksi suurin osa työntavoitteista onnistui niin, että voin olla tyytyväinen ja toivonkin että Visual Components -ohjelmistojen käyttö lisääntyisi vieläkin, koska suomalaiset ohjelmistot ja ohjelmointiosaaminen on ensiluokkaista.

LÄHTEET

- /1/ VAMK. 2018. Vaasan ammattikorkeakoulu. Viitattu 9.5.2018.
<http://www.puv.fi/fi/about/>
- /2/ Technobothnia. 2018. Technobothnian kotisivu. Viitattu 9.5.2018.
<http://www.technobothnia.fi/>
- /3/ Kuivanen, R. 1999. Robotiikka. Vantaa. Talentum Oyj.
- /4/ Lahden ammattikorkeakoulu. 2016. Automaatio ja robotiikka. Viitattu 15.2.2018.
http://miniweb.lpt.fi/automaatio/opetus/luennot/pdf_tiedostot/Robotiikka_yleinen.pdf
- /5/ Keinänen, T., Kärkkäinen, P., Lähetkangas, M., Sumujärvi, M. 2010. Automaatiojärjestelmien logiikat ja ohjaustekniikat. Helsinki. WSOYpro Oy
- /6/ YTJ. 2018. Visual Components Oy:n yritystiedot. Helsinki. Viitattu 15.5.2018.
<https://tietopalvelu.ytj.fi/yritystiedot.aspx?yavain=523783&tarkiste=117ABDE5EA7F9D0192FB58B2DB50C50043EFE102>
- /7/ Visual Components Oy. 2018. Visual Components Oy:n yritysesittely. Viitattu 15.5.2018. <https://www.visualcomponents.com/about-us/>
- /8/ Visual Components Oy. 2018. Visual Components 4.0 -ohjelmiston esittely. Viitattu 17.5.2018. <https://www.visualcomponents.com/products/visual-components-4-0/>

LIITE 1.

Visual Components Premium 4.0 -käyttöohje

Tässä ohjeessa käydään läpi Visual Components Premium 4.0 -ohjelman ominaisuuksia, kerrotaan robottisoluympäristön ja robottiohjelman luomisvaiheista ja kerrotaan, miten ohjelmakoodi viedään ulos ohjelmistosta ja suoritetaan todellisella robotilla.

SISÄLLYS

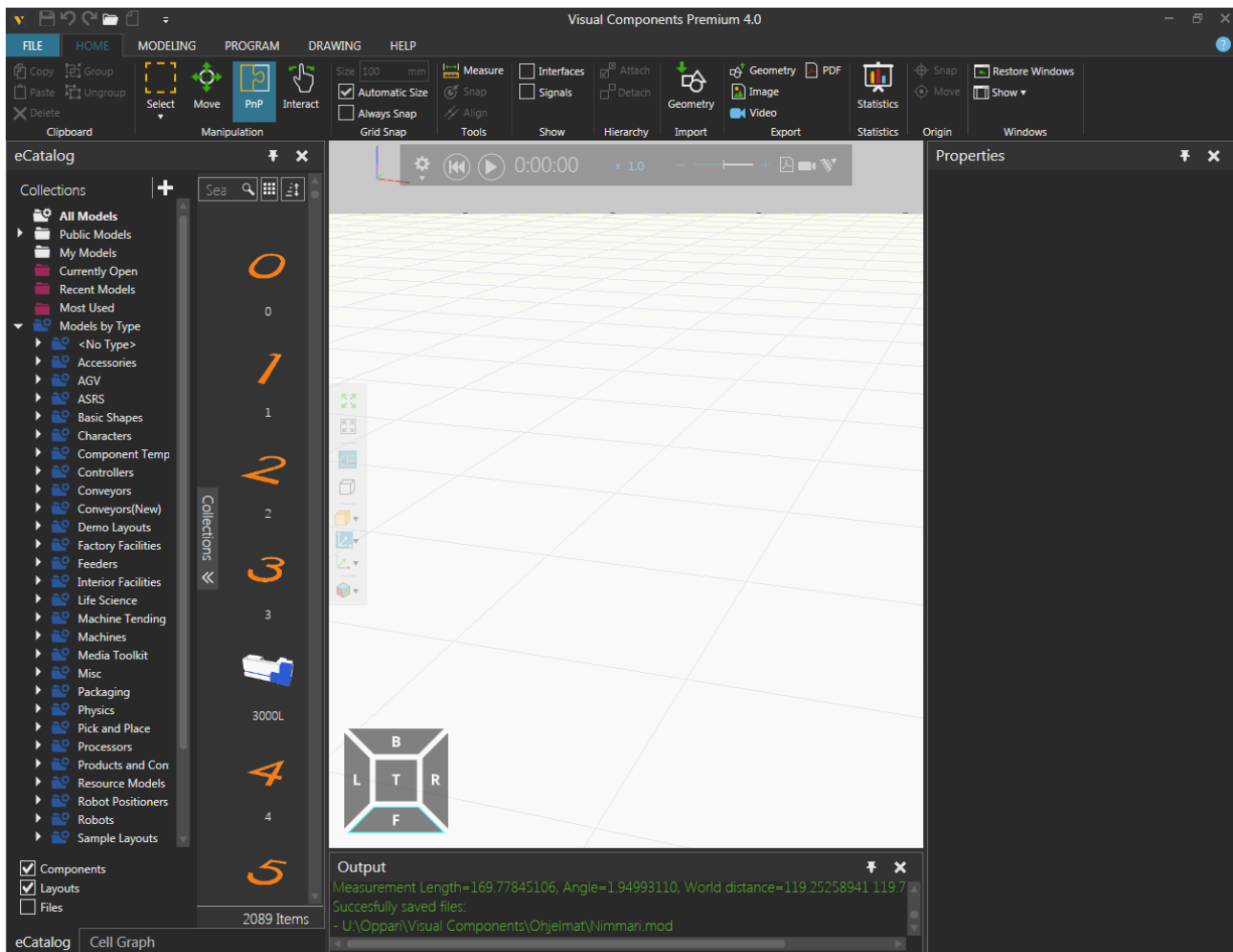
1. Visual Components Premium 4.0 -ohjelman käynnistys
2. Uuden robottisolun rakentaminen
3. Uuden robottiohjelman luominen
4. Ohjelman simulointi
5. Ohjelman ulosvienti ja käyttö todellisella robotilla.

1. Visual Components Premium 4.0 -ohjelman käynnistys

Visual Components Premium 4.0 käynnistyy työpöydän kuvakkeesta (Kuva 1), latautuu hetken ja avaa aloitusikkunan (Kuva 2).



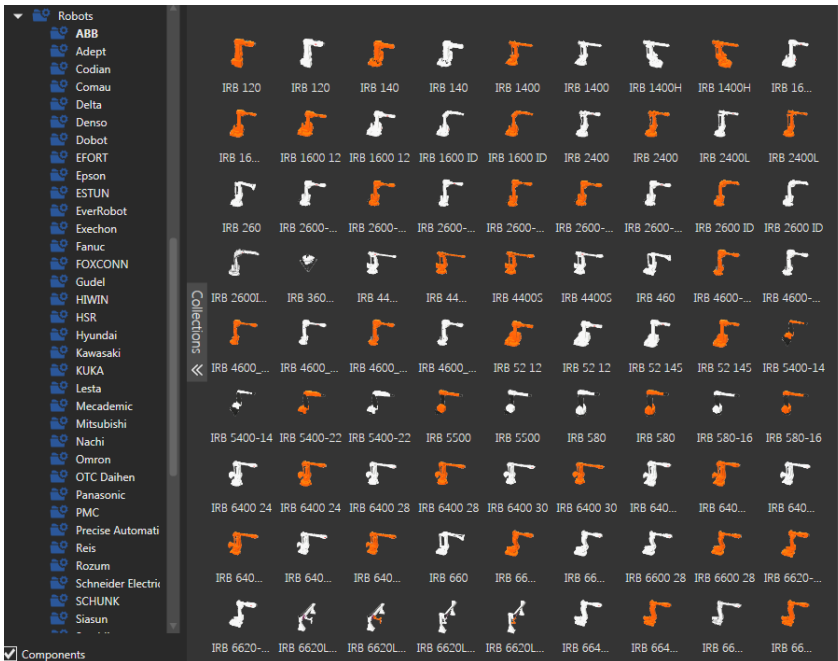
Kuva 1. Visual Components Premium 4.0 työpöytäkuva.



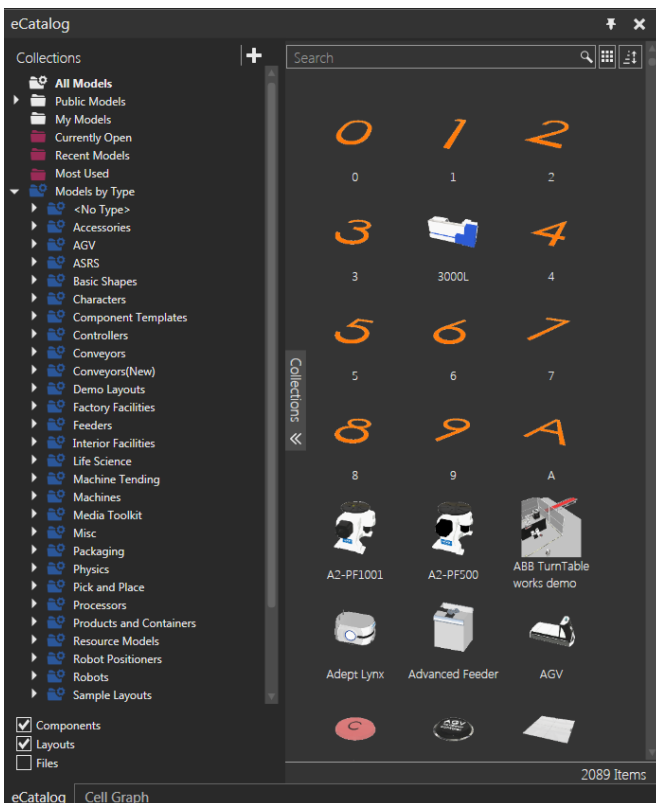
Kuva 2. Visual Components Premium 4.0 aloitusikkuna.

2. Uuden robottisolun rakentaminen

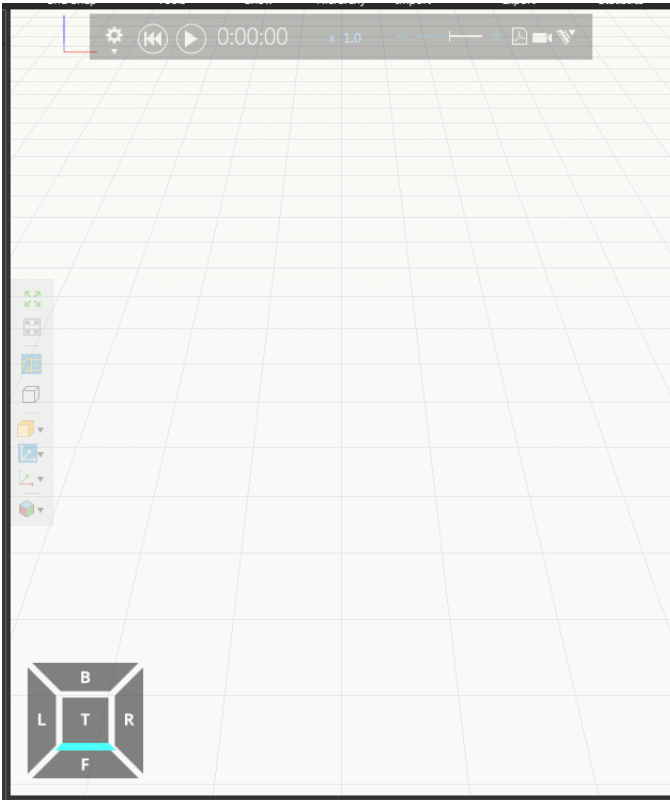
Uuden robottisolun rakentaminen on helppoa. Ensin valitaan eCatalog -ikkunasta (Kuva 3) robotin malli ja muut tarpeelliset komponentit (Kuva 4), jotka sitten ilmestyvät toiminta-avaruus-ikkunan (Kuva 5) origopisteeseen (Kuva 6).



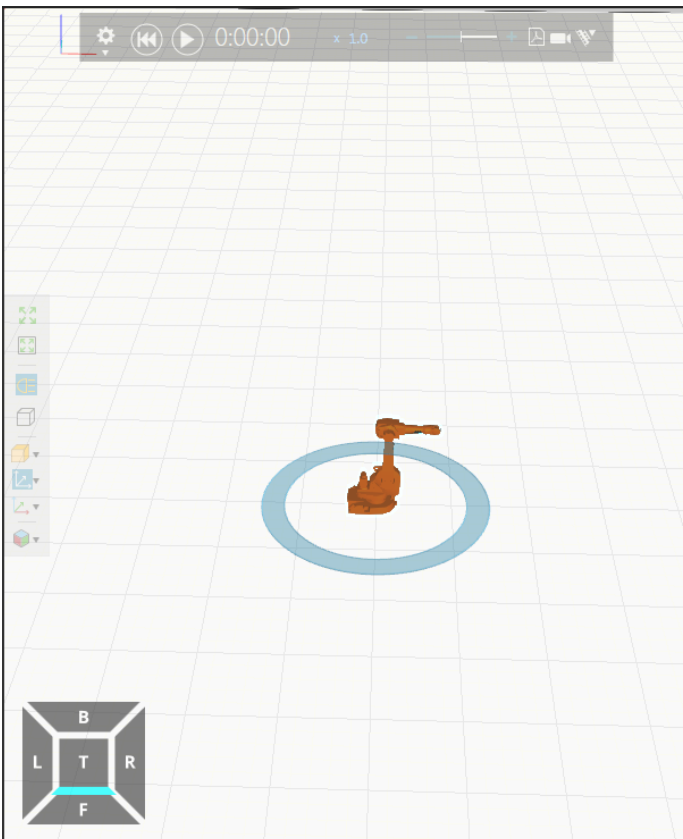
Kuva 3. Robottimalleja (ABB).



Kuva 4. E-Catalog-ikkuna

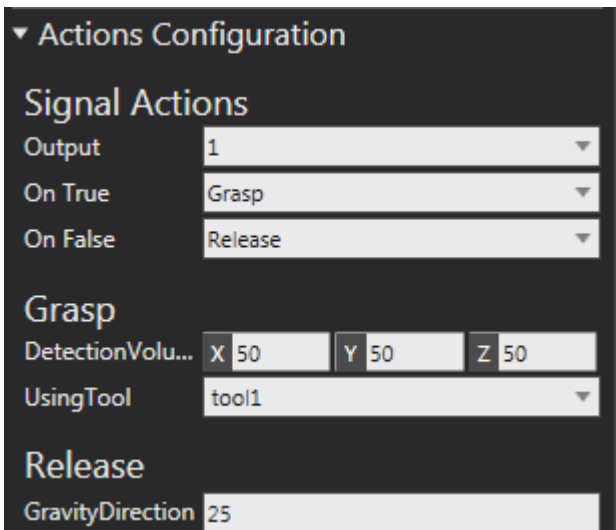


Kuva 5. Ohjelman tyhjä toimintaikkuna.



Kuva 6. Valittu robottimalli origopisteessä.

Ennen robottisolun tekoa tulee huomioida, jos esimerkiksi suunniteltavassa robottiohjelmassa robotin työkalulla halutaan esimerkiksi siirtää työstettävää kappaletta, niin silloin tulee robotille määrittää Output-toimintakäskyt heti ensimmäisenä uuden robottisolun rakennusvaiheessa. Output-käskyjen asetukset löytyvät valitun robotin Component Properties-ikkunan alareunassa sijaitsevasta Actions Configuration-valikosta (Kuva 7), jossa valitaan esimerkiksi ensimmäiseen ryhmään kuuluva Grasp eli tarttujatyypinen Output-signaali, jonka vastakkainen käsky on Release eli irrotus.



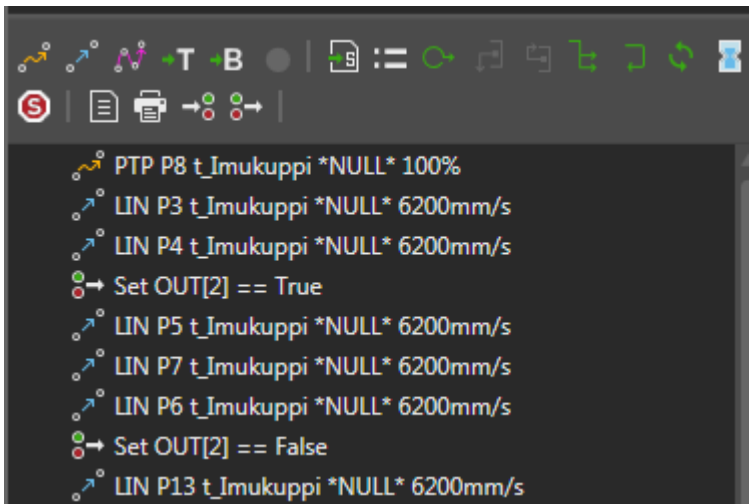
Kuva 7. Output-signaalien valintaikkuna.

Output-signaalit on jaettu neljään eri tyyppiin ryhmiin (Kuva 8), joilla kaikilla on sekä aktivointi että lopetuskäskyt, ja jotka ovat nimeltään Grasp/Release, Trace Tool On/Trace Tool Off, Mount Tool/Unmount Tool ja Trace External Tool On/Trace External Tool Off, joista kolme ensimmäistä sisältää 16 signaalipaikkaa ja neljäs 32.

Output #	Action On True	Action On False
1...16	Grasp	Release
17...32	Trace Tool On	Trace Tool Off
33...48	Mount Tool	Unmount Tool
49...80	Trace External Tool On	Trace External Tool Off

Kuva 8. Output-signaalitaulukko.

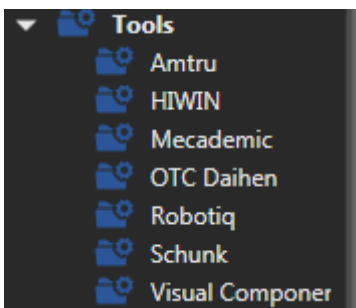
Määritettyä Output-signaalia kutsutaan robottiohjelmassa (Kuva 9) Set Binary Output Statement-komennolla, jossa Output-signaalin suorittamiseksi valitaan joko True tai False-toiminto.



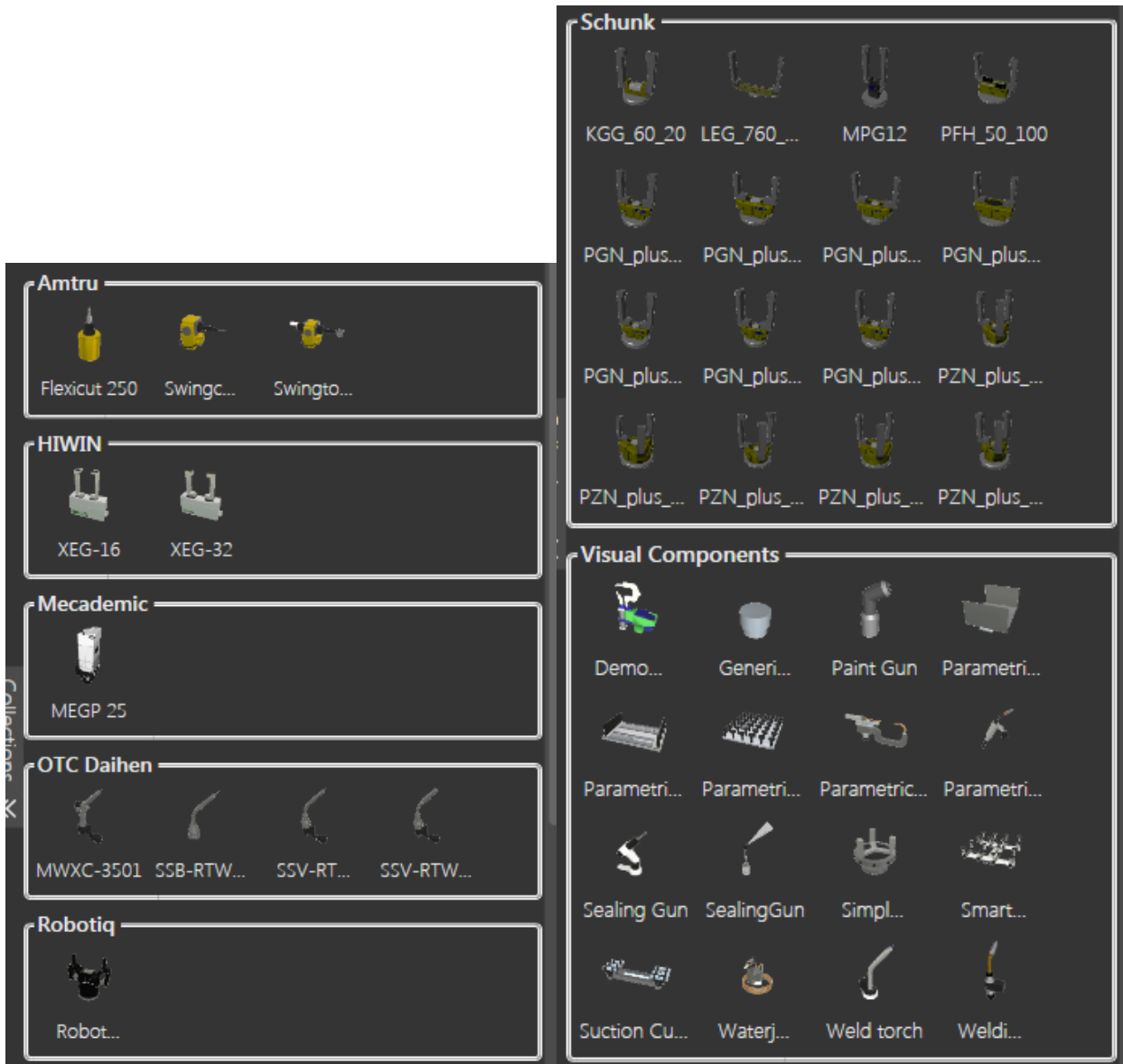
Kuva 9. Esimerkki Output-signaalien käytöstä robottiohjelmassa.

Output-käskeyjen asettamisen jälkeen voidaan siirtyä robotin työkalun valitsemiseen.

Robotissa käytettäviä työkaluja löytyy e-Catalogin Tools-luettelosta (Kuva 10) monelta eri valmistajalta ja niistäkin on mallikuvat (Kuva 11) kuten roboteista ja muista oheislaitteista. Valittu työkalu ilmestyy robotin käsivarren päähän ja pysyy kiinni ilman erillistä kiinnitystoimintoa. Joidenkin yksinkertaisten työkalujen kokoa pystyy tarvittaessa muuttamaan sopivamman kokoisiksi.

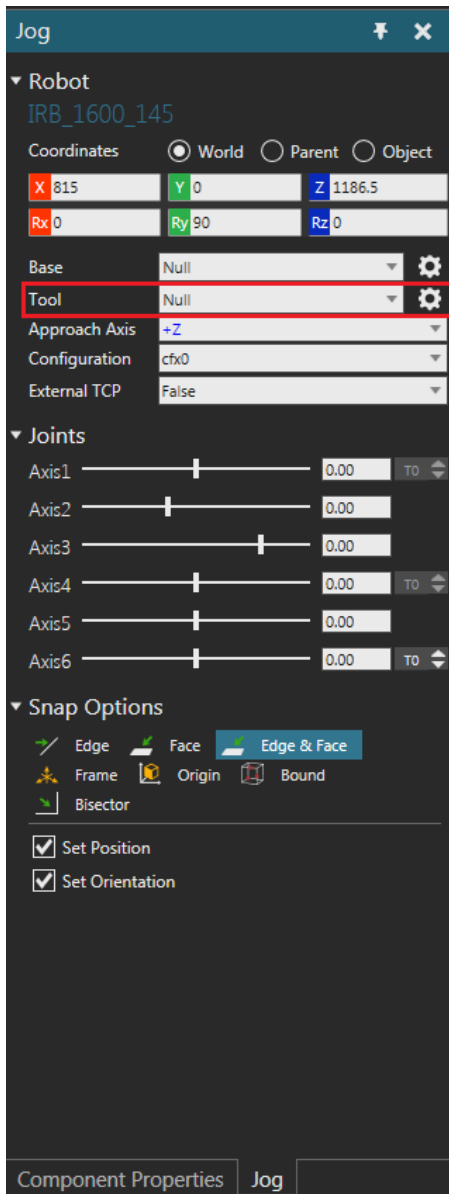


Kuva 10. E-Catalogin Tools-luettelo.

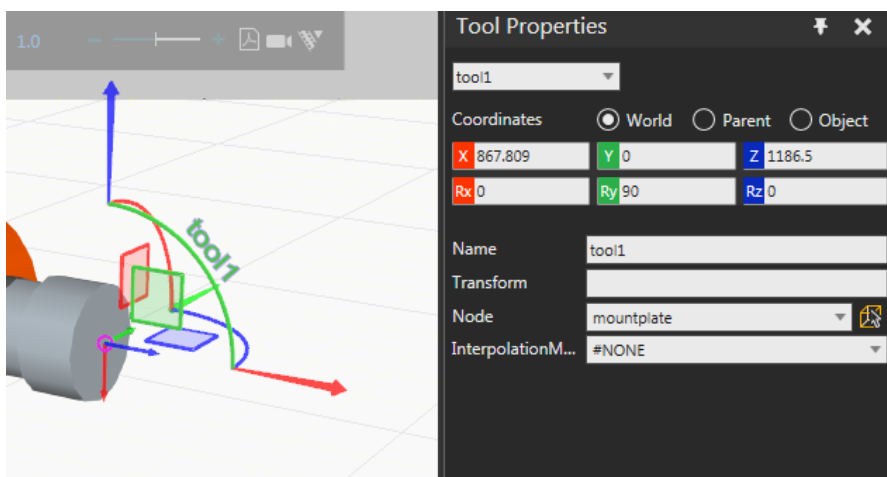


Kuva 11. Robotin erilaisia työkaluja.

Valitulle työkalulle annetaan nimi ja luodaan työkalupiste Program-välilehdessä aktivoimalla ensin robotti tai robotin työkalu, sitten viereen ilmestyvästä ikkunasta valitaan Jog-välilehti (Kuva 12), jossa valitaan työkalulle nimi valitsemalla Tool-kentän työkalunimiluettelosta yksi valmiista työkalunimistä, joita ovat tool1, tool2 jne. Työkalun nimen valitsemisen jälkeen valitaan viereinen hammasrataskuvake, jonka jälkeen Tool Properties-välilehti (Kuva 13) avautuu ja työkalun nimen voi vaihtaa sekä työkalupisteen paikan määrittää haluttuun paikkaan.

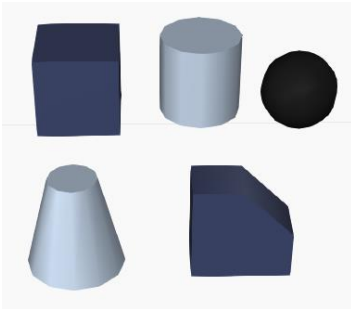


Kuva 12. Jog-välilehti.

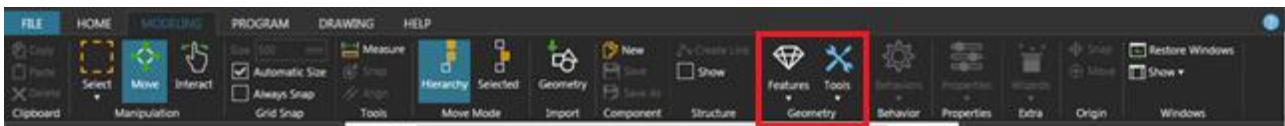


Kuva 13. Tool Properties-ikkuna ja työkalupisteen paikan määrittäminen.

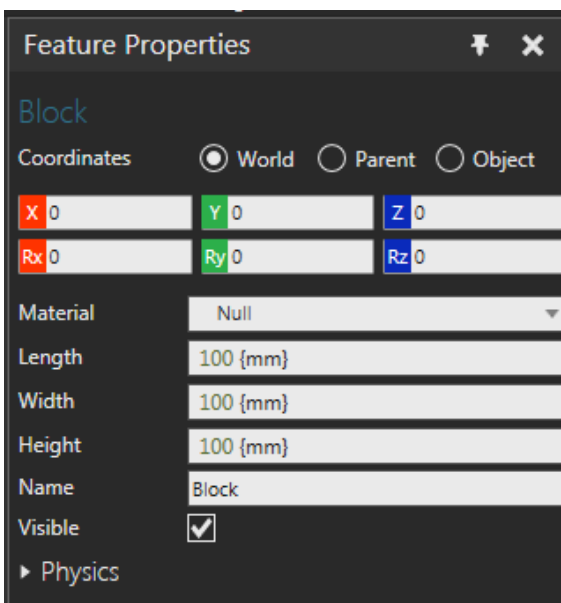
Robottisoluun on mahdollista mallintaa tarvittaessa myös yksinkertaisen muotoisia komponentteja, kuten kuutioita tai ympyrälieriöitä (Kuva 14), joita pystyy lisäämään Modeling-välilehden Geometry-osan Feature-valikosta (Kuva 15), jonka jälkeen valitaan haluttu muoto, joka sitten ilmestyy solun origopisteeseen ja jolle avautuu samalla mitoitusikkuna (Kuva 16).



Kuva 14. Yksinkertaisia komponenttimalleja.

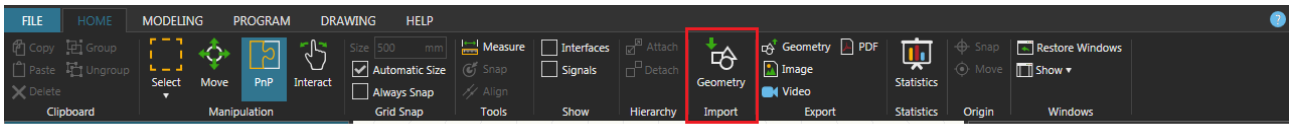


Kuva 15. Modeling -välilehti.

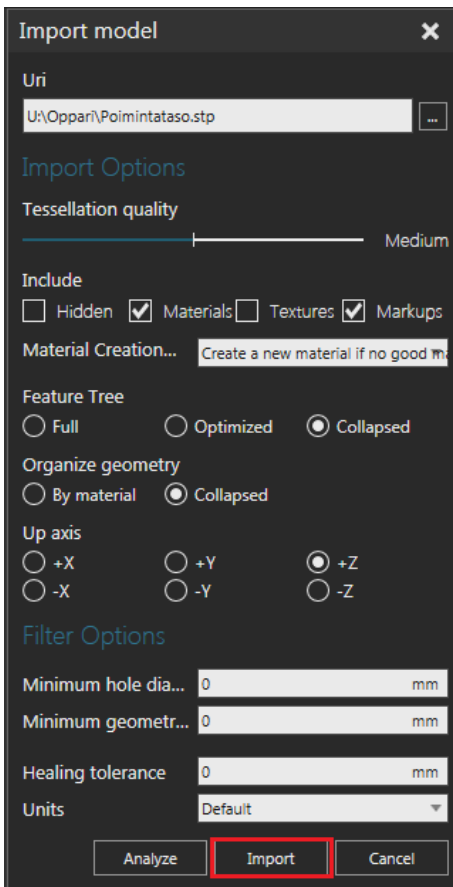


Kuva 16. Yksinkertaisen kappaleen (kuutio) mitoitusikkuna.

Ohjelman ulkopuolella mallinnettujen yksittäisten komponenttien, ja pienien kokoonpanojen, tuonti Visual Components-ohjelmaan onnistuu Home-välilehden Import-osasta (Kuva 17), jonka jälkeen haetaan haluttu STEP-tiedostotyyppinen komponentti, jonka jälkeen ilmestyy Import model-ikkuna (Kuva 18), jolla komponentti sijoitetaan halutulle tasolle ja lopuksi tuodaan robottisoluun Import-painikkeella.

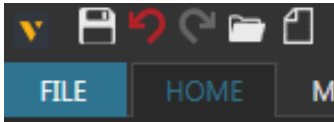


Kuva 17. Home -välilehti, johon on merkitty Import -osio.

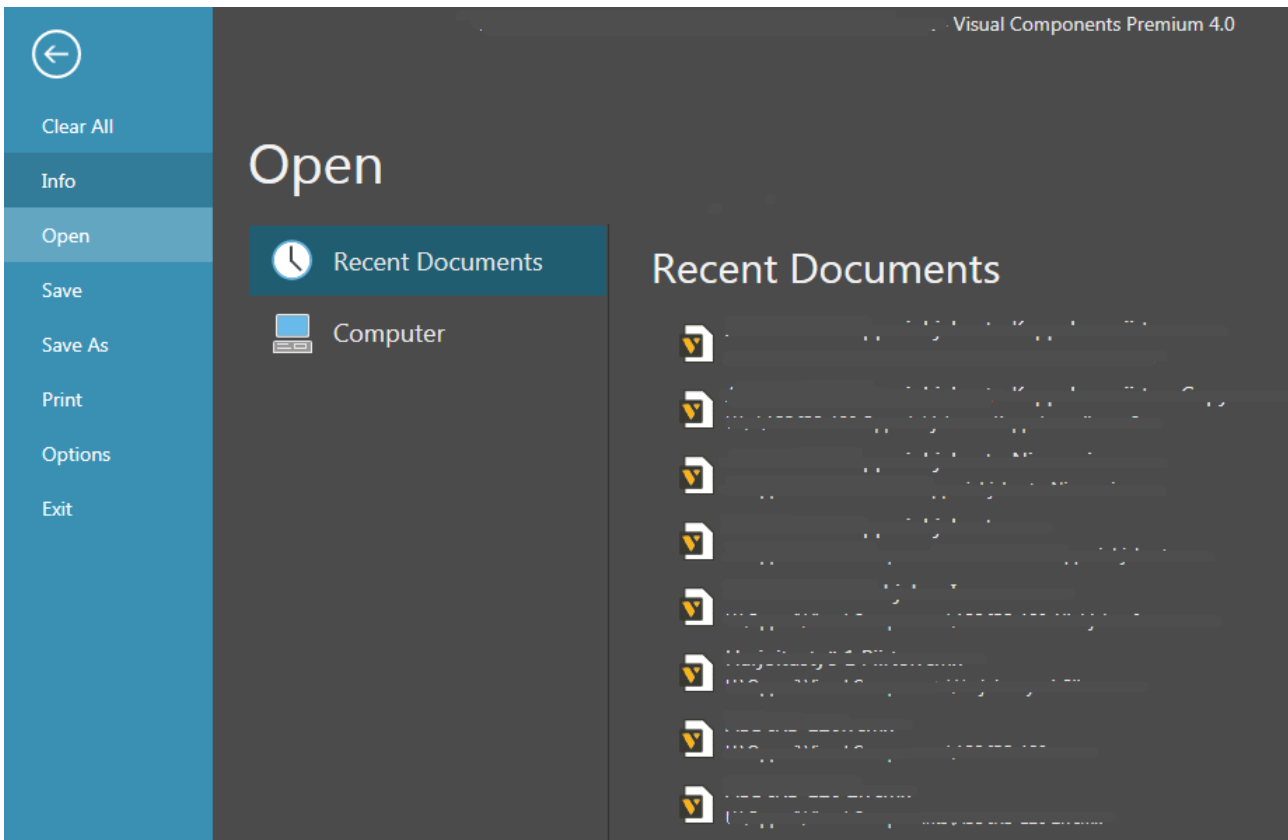


Kuva 18. Import model-ikkuna ja merkitty Import-painike.

Aiemmin luotu robottisolu saadaan näkyviin valitsemalla ensin File-valikko (Kuva 19), jonka jälkeen valitaan Open-valikko (Kuva 20), jolloin aiempi robottisolu on valittavissa Recent Documents-luettelosta sekä myös erikseen haettavissa tietokoneen muista dokumenttikansioista.



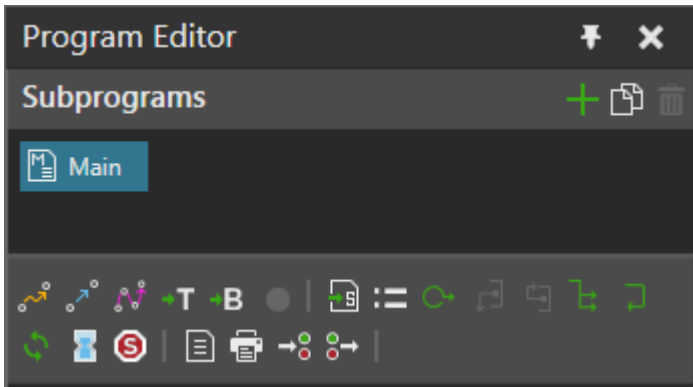
Kuva 19. File-valikko.



Kuva 20. Open-valikko.

3. Uuden robottiohjelman luominen

Uusi robottiohjelma luodaan Program-valikossa, jossa ensin aktivoidaan robotti, jolloin robotin vasemmalle puolelle ilmestyy Program Editor-ikkuna (Kuva 21.), jossa robottiohjelma rakennetaan.



Kuva 21. Program Editor-ikkuna ja Toolbar-valikko.

Robottiohjelman alussa määritetään ensin kotipiste, johon yleensä palataan ohjelman lopussa. Kotipisteen jälkeen robotille on mahdollista, riippuen työkalusta ja työkalupisteen sijainnista, tehdä millainen ohjelma tahansa sekä määrittää ohjelman sijainti joko pääohjelmaan (Main) tai aliohjelmaan (Subroutine). Aliohjelma saadaan käyttöön kutsumalla sitä pääohjelmassa Call ”Aliohjelma” -komennolla.

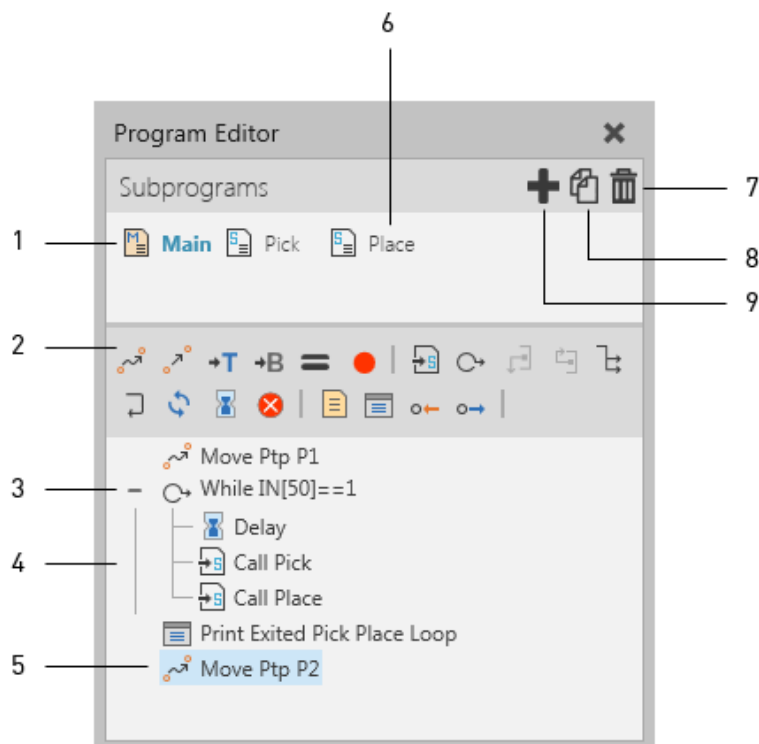
Robottiohjelman tekoon käytettävän Toolbar-valikon kuvioiden nimet ja toiminnot on kerrottu järjestyksessä seuraavasti Taulukossa 1:

<u>Point-to-Point Motion Statement:</u> Executes a point-to-point motion to a specified target.
<u>Linear Motion Statement:</u> Executes a linear motion to a specified target.
<u>Path Statement:</u> Executes motion along a specified path.
<u>Define Tool Statement:</u> Changes the position and orientation of a tool frame.
<u>Define Base Statement:</u> Changes the position and orientation of a base frame.
<u>Touch-up:</u> Updates and saves the motion statement with the current base/tool robot configuration.
<u>Call Sequence Statement:</u> Executes a subroutine in a robot's program.
<u>Assign Variable Statement:</u> Assign a value to a variable.
<u>While Statement:</u> While Statement.
<u>Break Statement:</u> Break Statement.
<u>Continue Statement:</u> Continue Statement.
<u>If Statement:</u> If Statement.
<u>Return Statement:</u> Return Statement.

<u>Program Synchronize Statement:</u> Program Synchronize Statement.
<u>Delay Statement:</u> Executes a time delay in seconds.
<u>Halt Statement:</u> Stops a simulation and can also be used to reset a simulation.
<u>Comment Statement:</u> Comment Statement.
<u>Print Statement:</u> Prints a message in the application's Message Panel.
<u>Wait for Binary Input Statement:</u> pauses a robot's program and listens for a connected signal's value to match an inputted value. Once completed, the execution of the robot's program resumes.
<u>Set Binary Output Statement:</u> Sends out a signal and its value to connected behaviors. It is also used to execute signal actions like grasp, release and trace.

Taulukko 1. Program Editor.

Kuvassa 22 on vielä kerrottu lyhyesti robottiohjelman Program Editor-ikkunan toiminnoista.



1. Main Routine
2. Statements Toolbar
3. Conditional Statement
4. Nested Statement
5. Motion Type Statement
6. Subroutine
7. Delete Selected Routine
8. Duplicate Selected Routine
9. Add Subroutine

Kuva 22. Program Editor-ikkunan toimintojen kuvaus.

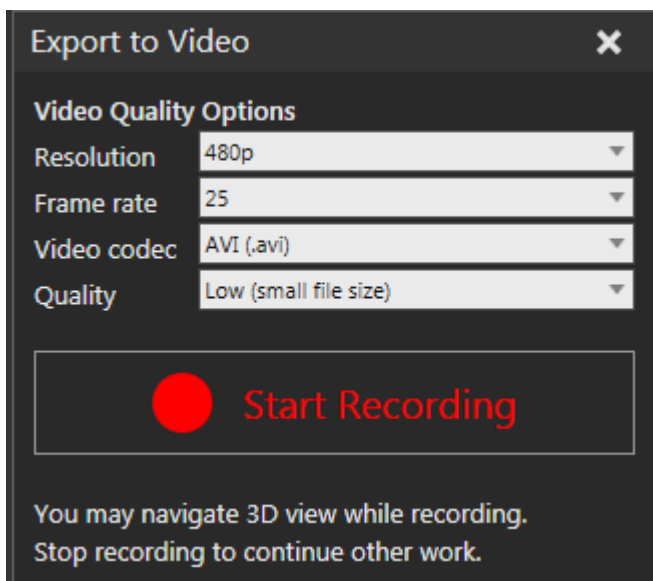
4. Ohjelman simulointi

Robottiohjelman simulointiohjain (Kuva 23) löytyy toiminta-avaruus-ikkunan yläreunasta ja sillä pystyy ohjelman simuloinnin lisäksi esimerkiksi säätämään ohjelman toistonopeutta, tekemään simuloinnista PDF-mallin sekä videotiedoston.



Kuva 23. Simulointiohjain.

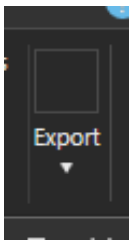
Kuvassa 24 kuvataan simulointivideon tekoon muutettavissa olevista ominaisuuksista, joita ovat resoluutio, kehys, tiedostotyyppi ja laatu. Simuloinnin kuvaus käynnistyy Start Recording-toiminnolla. Videon nauhoituksen aikana on vielä mahdollista muuttaa kuvakulmaa, kuvan etäisyyttä sekä toistonopeutta tarvittaessa.



Kuva 24. Simulointivideon tekoikkuna.

5. Ohjelman ulosvienti ja käyttö todellisella robotilla

Robottiohjelman ulosvienti Visual Components 4.0 -ohjelmistosta onnistuu vain ohjelmiston Program-välilehden Export-valikon Post Process-toiminnolla. Ohjelmakoodin tiedostotyyppi muuttuu mod-tiedostoksi, joka on yhteensopiva muiden etäohjelmointiohjelmistojen kanssa. Ohjelmakoodi pystytään ulosviennin jälkeen vielä tarkistamaan avaamalla se WordPad-tiedostoksi.



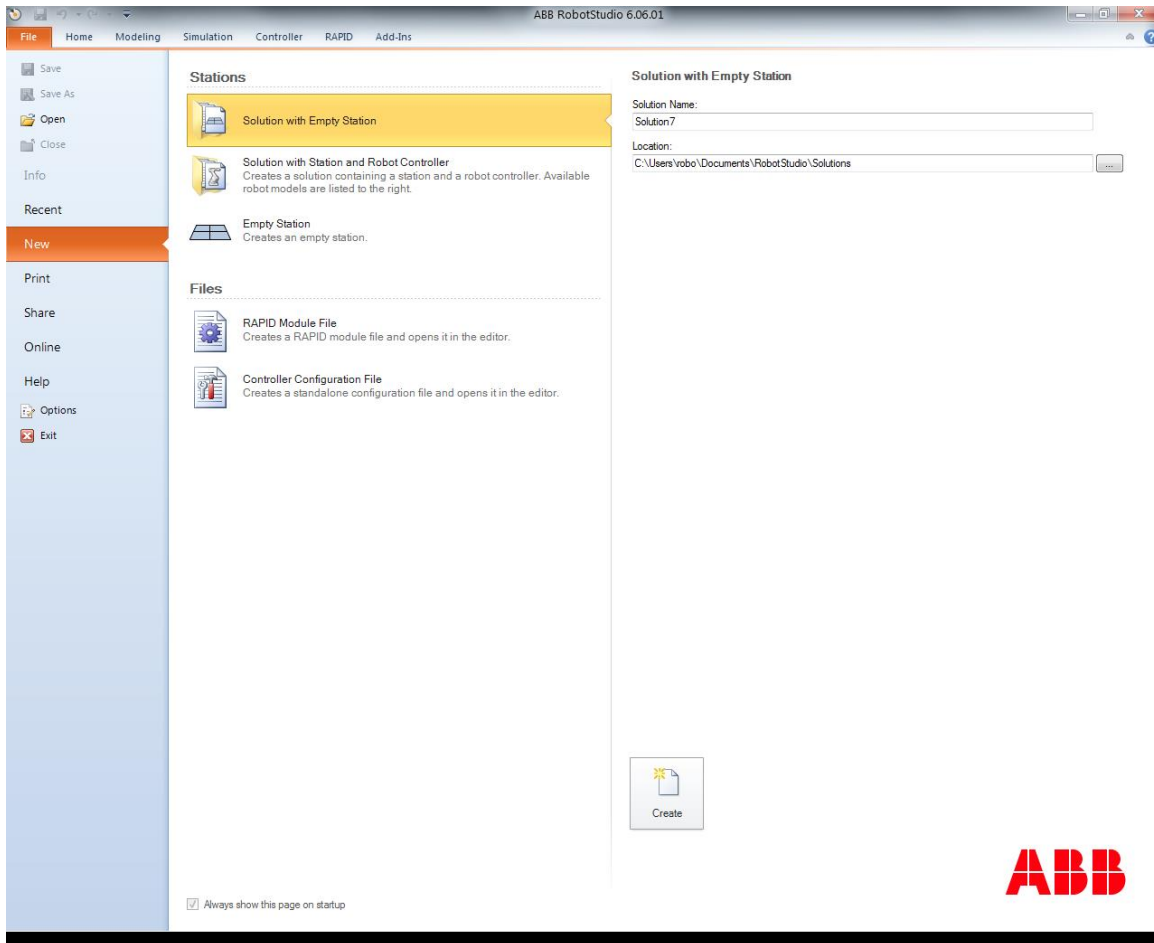
Kuva 25. Export-valikko.

Jos luodussa robottiohjelmassa on käytetty esimerkiksi ABB tai FANUC-robottia käytetään ohjelman siirtoon näiden teollisuusrobottimallien etäohjelmointiohjelmistoa eli RobotStudiota (Kuva 26).



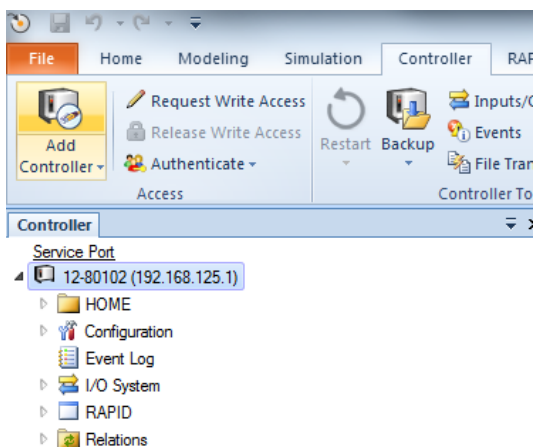
Kuva 26. RobotStudion työpöytäkuva.

RobotStudion aloitusikkunan (Kuva 27) avauduttua valitaan vasemmalta Open-kuvake ja haetaan Visual Components 4.0 -ohjelmassa luotu robottiohjelmakooditiedosto.



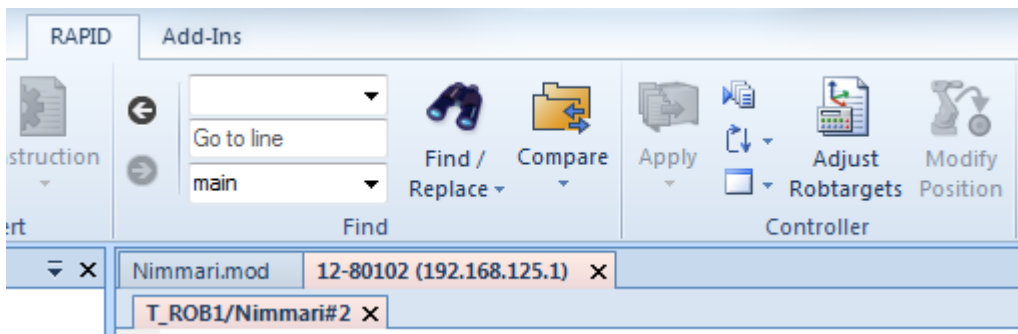
Kuva 27. RobotStudion aloitusikkuna.

Robottiohjelmakoodin ilmestyttyä näkyviin RobotStudiassa luodaan yhteys todelliseen robottiin siirtymällä Controller-välilehteen ja valitsemalla Add Controller-toiminto, joka ehdottaa valittavissa olevia robotin ohjausyksiköitä, jotka ilmoitetaan IP-osoitteina (Kuva 28).

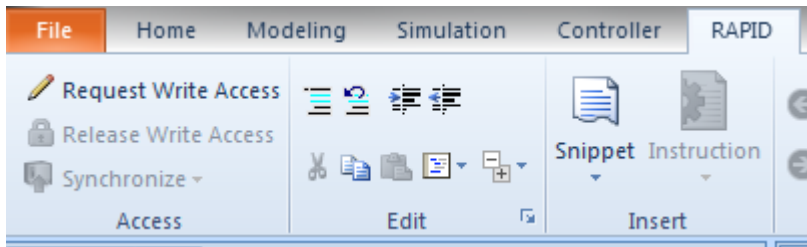


Kuva 28. Controller-välilehti, jossa on luotu yhteys todelliselle robotille.

Kun yhteys todelliseen robottiin on luotu ilmestyy robottiohjelma RobotStudiossa todellisen robotin käytössä olevaksi ohjelmaksi (Kuva 29), joka voidaan ottaa käyttöön samantien. Jos ohjelmakoodissa näkyy punaisella alleviivattuja kohtia ilmoittaa Rapid-ohjelmakieli nämä kohdat virhetiedoiksi, jotka vielä estävät ohjelman käytön todellisella robotilla. Jos virhetietoja löytyy valitaan Rapid-välilehden Access-valikosta Request Write Access-toiminto (Kuva 30), niin virheet pystytään korjaamaan manuaalisesti. Ennen ohjelmakoodin korjausta tulee vielä kuitata sallintalupa koodiin tehtävistä muutoksista robotin käsiohjaimessa, jotta todellinen robottin ohjausyksikkö saa muutetun ohjelmakoodin käyttöönsä.



Kuva 29. Luotuetäohjelma todellisen robotin käyttöohjelmana.



Kuva 30. Rapid-välilehden Access-valikko.

Kun ohjelmakoodin tarvittavat muutokset on tehty ja koodin tarkastettu vielä virheiden osalta päivitetään uusi ohjelmakoodi todelliselle robotille Rapid-välilehden Controller-valikon Apply-komennolla ja sitten robottiohjelma voidaan suorittaa todellisella robotilla.

LIITE 2.

Nimikirjoitus ohjelmakoodi

%%%

VERSION: 1

LANGUAGE: ENGLISH

%%%

MODULE Nimmari

```
PERS tooldata t_pencil:=[TRUE,[[54.64,127.44,99.68],[0.594425,-
0.400004,0.587072,0.376832]], [5, [0,0,0.001], [1,0,0,0],0,0,0]];
```

```
PERS robtarget P1:=[[385.088,-
223.389,474.821],[0.0101665,0.253529,0.966963,0.0245456],[-1,-1,-
1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P2:=[[36.3121,-342.999,329.451],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P3:=[[36.3121,-457.839,56.096],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P4:=[[36.3121,-457.839,20.495],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P5:=[[36.3121,-331.605,20.495],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P6:=[[104.064,-348.871,20.495],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P7:=[[36.3121,-381.605,20.495],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P8:=[[104.064,-348.871,20.495],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P9:=[[145.863,-331.605,20.495],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P10:=[[145.863,-457.839,20.495],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P11:=[[145.863,-331.605,20.495],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P12:=[[188.555,-356.064,20.495],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P13:=[[145.863,-381.605,20.495],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P21:=[[188.555,-457.839,20.495],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P22:=[[145.863,-381.605,20.495],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P23:=[[188.555,-356.064,20.495],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P24:=[[239.022,-331.605,20.495],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P25:=[[239.022,-457.839,20.495],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P29:=[[219.022,-457.839,20.495],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```

PERS robtarget P26:=[[209.022,-423.586,20.495],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P27:=[[36.3121,-342.999,329.451],[0.0642884,-0.643249,-
0.761862,-0.0407863],[-1,-1,-1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P28:=[[385.088,-
223.389,474.821],[0.0101665,0.253529,0.966963,0.0245456],[-1,-1,-
1,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];

```

```

PROC main()
  Nimmari;
ENDPROC

```

```

PROC Nimmari()
  MoveJ P1,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P2,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P3,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P4,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P5,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P6,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P7,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P8,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P9,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P10,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P11,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P12,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P13,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P21,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P22,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P23,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P24,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P25,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P29,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P26,vmax,fine,t_pencil\WObj:=wobj0;
  MoveL P27,vmax,fine,t_pencil\WObj:=wobj0;
  MoveJ P28,vmax,fine,t_pencil\WObj:=wobj0;
ENDPROC

```

```

ENDMODULE

```

LIITE 3.

Kappaleen siirto 1 ohjelmakoodi

%%%

VERSION: 1

LANGUAGE: ENGLISH

%%%

MODULE Siirto_1

```
PERS tooldata t_Imukuppi:=[TRUE, [[-37.8373,-
84.3117,99.7472],[0.273114,-0.0102264,-
0.707033,0.652234]], [5, [0,0,0.001], [1,0,0,0],0,0,0]];
```

```
PERS robtarget P1:=[[66.554,-453.738,469], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-1,-1,-
3,1], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P3:=[[66.554,-453.738,169], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-1,-1,-
3,1], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P4:=[[39.9432,-546.043,2.6], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-1,0,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P5:=[[39.9432,-546.043,19], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-1,0,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P7:=[[129.899,-546.043,10.5233], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,1,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P6:=[[129.899,-546.043,0], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,1,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P13:=[[129.899,-546.043,10.5233], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,1,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P21:=[[66.554,-453.738,169], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-1,-1,-
3,1], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P22:=[[39.9432,-546.043,2.6], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-1,0,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P23:=[[39.9432,-546.043,19], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-1,0,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P24:=[[244.899,-546.043,10.5233], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,-2,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P25:=[[244.899,-546.043,-2.6], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,-2,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```

PERS robtarget P26:=[[244.899,-546.043,10.5233],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,-2,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P27:=[[66.554,-453.738,169],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,-1,-
3,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P28:=[[39.8432,-426.143,2.6],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P29:=[[39.8432,-426.143,19],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P30:=[[129.399,-426.143,10.5233],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,1,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P31:=[[129.799,-426.143,0],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,1,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P32:=[[129.399,-426.143,10.5233],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,1,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P33:=[[66.554,-453.738,169],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,-1,-
3,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P34:=[[39.8432,-426.143,0],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P35:=[[39.8432,-426.143,19],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P36:=[[244.499,-426.143,10.5233],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,-2,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P37:=[[244.899,-426.143,-5.2],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,-2,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P38:=[[244.499,-426.143,10.5233],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,-2,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P39:=[[244.899,-426.143,-5.2],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,-2,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P40:=[[244.499,-426.143,10.5233],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,-2,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P41:=[[39.8432,-426.143,19],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P42:=[[39.8432,-426.143,0],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P43:=[[39.8432,-426.143,19],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];

```

```

PERS robtarget P44:=[[129.399,-426.143,10.5233],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,1,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P45:=[[129.799,-426.143,0],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,1,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P46:=[[129.399,-426.143,10.5233],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,1,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P47:=[[39.8432,-426.143,19],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P48:=[[39.8432,-426.143,2.6],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P49:=[[39.8432,-426.143,19],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P50:=[[66.554,-453.738,169],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,-1,-
3,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P51:=[[243.499,-546.043,10.5233],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,-2,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P52:=[[243.499,-546.043,-2.6],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,-2,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P53:=[[243.499,-546.043,10.5233],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,-2,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P54:=[[39.9432,-546.043,19],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P55:=[[39.9432,-546.043,2.6],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P56:=[[39.9432,-546.043,19],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P57:=[[129.899,-546.043,10.5233],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,1,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P58:=[[129.899,-546.043,0],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,1,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P59:=[[129.899,-546.043,10.5233],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,1,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P60:=[[39.9432,-546.043,19],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P61:=[[39.9432,-546.043,2.6],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];

```

```

PERS robtarget P62:=[[39.9432,-546.043,19],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P63:=[[66.554,-453.738,169],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,-1,-
3,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P8:=[[66.554,-453.738,469],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,-1,-
3,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];

```

```

PROC main()
  Siirto;
ENDPROC

```

```

PROC Siirto()
  MoveJ P1,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P3,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P4,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,1;
  MoveL P5,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P7,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P6,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,0;
  MoveL P13,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P21,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P22,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,1;
  MoveL P23,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P24,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P25,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,0;
  MoveL P26,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P27,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P28,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,1;
  MoveL P29,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P30,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P31,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,0;
  MoveL P32,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P33,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P34,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,1;
  MoveL P35,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P36,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P37,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,0;
  MoveL P38,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P39,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,1;
  MoveL P40,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P41,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P42,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,0;
  MoveL P43,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P44,vmax,fine,t_Imukuppi\WObj:=wobj0;

```

```
MoveL P45,vmax,fine,t_Imukuppi\WObj:=wobj0;
SetDO do2,1;
MoveL P46,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P47,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P48,vmax,fine,t_Imukuppi\WObj:=wobj0;
SetDO do2,0;
MoveL P49,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P50,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P51,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P52,vmax,fine,t_Imukuppi\WObj:=wobj0;
SetDO do2,1;
MoveL P53,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P54,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P55,vmax,fine,t_Imukuppi\WObj:=wobj0;
SetDO do2,0;
MoveL P56,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P57,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P58,vmax,fine,t_Imukuppi\WObj:=wobj0;
SetDO do2,1;
MoveL P59,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P60,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P61,vmax,fine,t_Imukuppi\WObj:=wobj0;
SetDO do2,0;
MoveL P62,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P63,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveJ P8,vmax,fine,t_Imukuppi\WObj:=wobj0;
ENDPROC

ENDMODULE
```


LIITE 4.

Kappaleen siirto 2 Ohjelmakoodi

%%%

VERSION: 1

LANGUAGE: ENGLISH

%%%

MODULE Siirto_2

```
PERS tooldata t_Imukuppi:=[TRUE, [[-37.8373,-
84.3117,99.7472],[0.273114,-0.0102264,-
0.707033,0.652234]], [5, [0,0,0.001], [1,0,0,0],0,0,0]];
```

```
PERS robtarget P8:=[[66.554,-453.738,469], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-1,1,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P3:=[[66.554,-453.738,169], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-1,-1,-
3,1], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P4:=[[-124.357,-558.643,2.6], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-2,0,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P5:=[[-124.357,-558.643,19], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-2,0,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P7:=[[178.099,-546.043,10.5233], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,1,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P6:=[[178.099,-546.043,2.6], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,1,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P13:=[[178.099,-546.043,10.5233], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,1,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P21:=[[66.554,-453.738,169], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-1,-1,-
3,1], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P22:=[[-4.35678,-558.543,2.6], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-2,0,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P23:=[[-4.35678,-558.543,19], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-2,0,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P24:=[[293.099,-546.043,10.5233], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,-2,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```
PERS robtarget P25:=[[293.099,-546.043,2.6], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,-2,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
```

```

PERS robtarget P26:=[ [293.099,-546.043,10.5233], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,-2,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P27:=[ [66.554,-453.738,169], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-1,-1,-
3,1], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P28:=[ [-124.457,-443.643,2.6], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-2,0,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P29:=[ [-124.457,-443.643,19], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-2,0,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P30:=[ [177.599,-426.143,10.5233], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,-2,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P31:=[ [177.999,-426.143,2.6], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,-2,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P32:=[ [177.599,-426.143,10.5233], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,-2,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P33:=[ [66.554,-453.738,169], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-1,-1,-
3,1], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P34:=[ [-4.45678,-443.643,0], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-2,0,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P35:=[ [-4.45678,-443.643,19], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-2,0,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P36:=[ [292.699,-426.143,10.5233], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,-2,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P37:=[ [293.099,-426.143,0], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,-2,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P38:=[ [292.699,-426.143,10.5233], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,-2,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P39:=[ [293.099,-426.143,0], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,-2,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P40:=[ [292.699,-426.143,10.5233], [-1.53908e-17,-
0.25136,0.967894,5.92644e-17], [-1,-2,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P41:=[ [-4.45678,-443.643,19], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-2,0,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P42:=[ [-4.45678,-443.643,0], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-2,0,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P43:=[ [-4.45678,-443.643,19], [-1.5391e-17,-
0.251362,0.967893,5.92644e-17], [-2,0,-
1,0], [9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];

```

```

PERS robtarget P44:=[[177.599,-426.143,10.5233],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,-2,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P45:=[[177.999,-426.143,2.6],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,-2,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P46:=[[177.599,-426.143,10.5233],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,-2,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P47:=[[-124.457,-443.643,19],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-2,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P48:=[[-124.457,-443.643,2.6],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-2,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P49:=[[-124.457,-443.643,19],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-2,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P50:=[[66.554,-453.738,169],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,-1,-
3,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P51:=[[293.099,-546.043,10.5233],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,-2,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P52:=[[293.099,-546.043,-2.6],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,-2,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P53:=[[293.099,-546.043,10.5233],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,-2,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P54:=[[-4.35678,-558.543,19],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-2,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P55:=[[-4.35678,-558.543,-2.6],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-2,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P56:=[[-4.35678,-558.543,19],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-2,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P57:=[[178.099,-546.043,10.5233],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,1,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P58:=[[178.099,-546.043,0],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,1,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P59:=[[178.099,-546.043,10.5233],[-1.53908e-17,-
0.25136,0.967894,5.92644e-17],[-1,1,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P60:=[[-124.357,-558.643,19],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-2,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P61:=[[-124.357,-558.643,0],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-2,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];

```

```

PERS robtarget P62:=[[-124.357,-558.643,19],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-2,0,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P63:=[[66.554,-453.738,169],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,-1,-
3,1],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];
PERS robtarget P9:=[[66.554,-453.738,469],[-1.5391e-17,-
0.251362,0.967893,5.92644e-17],[-1,1,-
1,0],[9e+09,9e+09,9e+09,9e+09,9e+09,9e+09]];

```

```

PROC main()
  Siirto;
ENDPROC

```

```

PROC Siirto()
  MoveJ P8,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P3,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P4,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,1;
  MoveL P5,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P7,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P6,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,0;
  MoveL P13,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P21,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P22,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,1;
  MoveL P23,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P24,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P25,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,0;
  MoveL P26,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P27,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P28,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,1;
  MoveL P29,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P30,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P31,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,0;
  MoveL P32,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P33,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P34,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,1;
  MoveL P35,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P36,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P37,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,0;
  MoveL P38,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P39,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,1;
  MoveL P40,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P41,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P42,vmax,fine,t_Imukuppi\WObj:=wobj0;
  SetDO do2,0;
  MoveL P43,vmax,fine,t_Imukuppi\WObj:=wobj0;
  MoveL P44,vmax,fine,t_Imukuppi\WObj:=wobj0;

```

```
MoveL P45,vmax,fine,t_Imukuppi\WObj:=wobj0;
SetDO do2,1;
MoveL P46,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P47,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P48,vmax,fine,t_Imukuppi\WObj:=wobj0;
SetDO do2,0;
MoveL P49,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P50,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P51,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P52,vmax,fine,t_Imukuppi\WObj:=wobj0;
SetDO do2,1;
MoveL P53,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P54,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P55,vmax,fine,t_Imukuppi\WObj:=wobj0;
SetDO do2,0;
MoveL P56,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P57,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P58,vmax,fine,t_Imukuppi\WObj:=wobj0;
SetDO do2,1;
MoveL P59,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P60,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P61,vmax,fine,t_Imukuppi\WObj:=wobj0;
SetDO do2,0;
MoveL P62,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveL P63,vmax,fine,t_Imukuppi\WObj:=wobj0;
MoveJ P9,vmax,fine,t_Imukuppi\WObj:=wobj0;
ENDPROC

ENDMODULE
```