



Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Niko-Pekka Rötökö

Ohjaus- ja servotekniikan PLC-harjoitusten suunnittelutyö

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Konetekniikka

Insinööriytyö

25.10.2019

Tekijä Otsikko	Niko-Pekka Rötkö Ohjaus- ja servotekniikan PLC-harjoitusten suunnittelutyö
Sivumäärä Aika	31 sivua + 3 liitettä 25.10.2019
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Konetekniikka
Ammatillinen pääaine	Koneautomaatio
Ohjaaja	Lehtori Antti Liljaniemi
<p>Insinööriyössä keskityttiin ohjaus- ja servotekniikkapohjaisten PLC-harjoitusten suunnitteluun sekä niiden toteuttamiseen. Toimeksiantajana annetulle työlle toimi Metropolia Ammattikorkeakoulu. Insinööriyö oli myös osa Metropolian laajempaa tutkimus- ja kehityshanketta, jonka tavoitteena on tarjota virtuaalinen opetusympäristö Myyrmäen kampuksen automaatiolaboratorion mekatronisista harjoituslaitteista.</p> <p>Ohjausjärjestelmät ja väylät-kurssin PC-harjoituslaitteistoista luoduista 3D-malleista rakennettiin digitaaliset kaksoset. Virtuaalimallien simulointi testattiin Siemens NX MCD-ohjelmistolla, jolloin niihin saatiin luotua kinematiikka ja dynamiikka mukaan. Virtuaalinen käyttöönotto suoritettiin varsinaisella ohjelmointiohjelmalla CODESYS PLC-editorilla. Jokaisesta neljästä virtuaalimallista tehtiin PC-harjoitustyöt sekä yksi esimerkillinen malliratkaisu.</p> <p>Loppuratkaisuna kehkeytyi kattava harjoituspaketti, joka sisältää digitaaliset kaksoset neljästä erilaisesta mekatronisesta laitteesta. Insinööriyö sisältää teoreettisen osuuden, ja liitteisiin on lisätty yhdestä virtuaalimallista yksityiskohtaisemmin läpikäyty PLC-malliratkaisu. Insinööriyön tuloksia on tarkoitus hyödyntää Metropolian insinöörikoulutuksessa, ja samalla työ edistää koulun tulevaisuuden hanketta.</p>	
Avainsanat	DigitalTwin, PLC, opetuskäyttö

Author Title	Niko-Pekka Rötö Planning of Control and Servo PLC Exercises
Number of Pages Date	31 pages + 3 appendices 25 October 2019
Degree	Bachelor of Engineering
Degree Programme	Mechanical Engineering
Professional Major	Machine Automation
Instructor	Antti Liljaniemi, Senior Lecturer
<p>The objective of this Bachelor's thesis was to design and implement PC exercises for control and servo technology. The study was commissioned by Metropolia University of Applied Sciences. The thesis is also part of Metropolia's larger project, which aims to create a virtual learning environment using the mechatronic training equipment in the Myyrmäki campus automation laboratory.</p> <p>3D models created from PC training equipment in the Control Systems and Buses course were used to create digital twins. The simulation was performed with the Siemens NX.MCD software and was designed with kinematics and dynamics. The virtual setup was done with the actual programming software using the CODESYS PLC editor. Each of the four virtual models were made into a PC exercise, and one model solutions was created for each virtual model.</p> <p>The final result was to create a comprehensive training package that includes digital duplicates of four different mechatronic devices. The thesis contains a theoretical part and also a virtual model, a more detailed PLC model solution, which is included in the appendices. The results of this thesis are intended to be used in the engineering studies of Metropolia UAS and also to contribute to the university's future projects.</p>	
Keywords	DigitalTwin, PLC, educational

Sisällys

Lyhenteet

1	Johdanto	1
2	Työn teoreettinen viitekehys	2
2.1	Asetetut tavoitteet	2
2.2	Virtuaalimallin suunnittelu	3
2.3	Digitaalinen kaksonen	5
2.4	Käytetyt ohjelmistot	7
2.4.1	CATIA V5 ja SolidWorks	7
2.4.2	Siemens NX ja MCD-lisäosa	9
2.4.3	NX MCD-moduulin käyttö	12
2.4.4	Yhteys ohjelmitavaan logiikkaan	14
2.4.5	CODESYS ja TwinCAT	16
2.5	Virtuaalimallit	17
2.5.1	Kuljetinlajittelija	18
2.5.2	Smart Factory	19
2.5.3	SCARA-robotti	20
2.5.4	XYZ-pienoisnosturi	21
3	Työn suunnittelu ja toteutus	22
3.1	Taustatiedot	22
3.2	PLC-harjoitukset	22
3.2.1	Työohjeet	23
3.2.2	PLC/ PC-malliharjoitus	23
3.2.3	Signaalien luominen	24
4	Tuloksien ja tavoitteiden saavuttaminen	25
5	Yhteenveto	26
	Lähteet	28

Liitteet

Liite 1. PLC/ PC-työohje, kuljetinlajittelija

Liite 2. PLC/ PC-signaalien luominen ja yhteyden luominen PLC-editoriin

Liite 3. PLC/ PC-malliharjoitus, kuljetinlajittelija

Lyhenteet

CAD	Computer Aided Design. Tietokoneavusteinen suunnittelu.
CAE	Computer Aided Engineering. Tietokoneavusteinen suunnittelu ja analysointi.
CAM	Computer Aided Manufacturing. Tietokoneavusteinen valmistus.
CATIA	Ransk. Conception Assistée Tridimensionnelle Interactive Appliquée. Suunnittelussa käytettävä 3D-mallinnusohjelmisto.
CODESYS	Controlled Development Systems. Ohjauslaitteiden, kuten ohjelmoitava logiikka, ja sulautettujen järjestelmien ohjelmointiin tarkoitettu kansainvälisen IEC 61131-3-normin mukainen ohjelmointiympäristö.
DigitalTwin	Digitaalinen kaksonen on tarkka virtuaalinen kopio fyysisistä prosesseista, ihmisistä, ympäristöistä, järjestelmistä tai laitteista, joita käytetään datan keräämiseen ja käsittelyyn.
NX MCD	Mechatronics Concept Designer. Siemens NX-ohjelmiston lisäosa, jolla saadaan simuloitua kappaleiden ja mallinnettujen kokonaisuuksien fyysisiä ominaisuuksia virtuaalisesti ja reaaliaikaisesti.
OPC	Open Platform Communications. Teollisuuden yhteentoimivuus-standardi.
PLC	Programmable Logic Controller. Ohjelmoitava logiikka.
PLM	Product Lifecycle Management. Tuotteen elinkaaren hallinta.

SolidWorks

Ranskalaisen Dassault Systemes kehittämä parametrinen 3D-mekaniikkasuunnitteluohjelmisto, joka sisältää tilavuus- ja pintamallinnustyökalut.

TwinCAT

The Windows Control and Automation Technology. Beckhoffin PC-pohjainen logiikka ohjelmisto.

1 Johdanto

Tarkastelun lähtökohdaksi insinööriyössä on otettu ohjaus- ja servotekniikan kurssiin sisältyvät PC-harjoitukset, joiden suunnittelutyössä keskitytään toteuttamisen tarkempiin yksityiskohtiin. Virtuaalitekniikka on tällä hetkellä hyvässä nousujohteessa IT-alalla, ja sitä hyödynnetään varsinkin tekoälyssä ja IoT-ratkaisuissa kuin myös analytiikassa. Tulevaisuudessa yleistyvät yhä mutkikkaammat IoT-järjestelmät, jotka tuottavat paljon myös dataa. Virtualisaation kautta voidaan optimoida erilaisten älykkäiden järjestelmien käyttöä sekä käyttäytymistä, tämän lisäksi voidaan myös simuloida vaihtoehtoisia tilanteita. On myös arvioitu, että jo vuoteen 2020 mennessä olisi yhdistetty yli 20 miljardia IoT-laitetta, joissa juuri tätä tekniikkaa olisi hyödynnetty. Insinööriyön toimeksiantajana toimi Metropolia Ammattikorkeakoulu, ja insinööriyö on myös osa suurempaa AuNe (Automation in Network)-hanketta. Hankkeen tavoitteena on luoda automaatioalan ja kone-tekniikan opiskelijoille sekä yritysten henkilökunnalle nykyaikainen digitaalinen verkko-oppimisympäristö FILE ja verkkolaboratorioiden konsepti FIINLab.

Tutkimustyön pääpainopiste on luoda fyysisesti jo olemassa olevista mekatronisista laitteista digitaaliset virtuaalimallit, joiden integroimista ja käyttöönottoa voidaan opettaa sekä harjoitella erillisessä ohjelmointiympäristössä. Mekatroniset laitteet, joista harjoitukset tehdään, ovat osa "Ohjausjärjestelmät ja väylät"-kurssin sisältöä. Laitteista on CAD-ohjelmistolla luotu 3D-mallit, jotka tuodaan erilliseen virtuaaliympäristöön. Valmiita digitaalisia malleja voidaan simuloida virtuaaliympäristössä, sillä niihin luotu kinematiikka ja dynamiikka tekevät sen mahdolliseksi. Mainitulla menettelytavalla niistä saadaan rakennettua niin kutsuttu digitaalinen kaksonen. Kaikista neljästä eri virtuaalimallista tehdään lopuksi myös kaksivaiheiset PC-työohjeet ja malliohjelmat PLC-editoria hyödyntäen. Insinööriyön tavoitteena on tehdä opiskelijoille omatoiminen PLC-ohjelmointiharjoituspaketti, jossa voidaan harjoitella virtuaalimallin rakentamista ja ohjelmointia. Tällä tavoin harjoituksen yhteydessä opitaan myös ohjelmoitavan logiikkaohjelman käytön periaatteet. Työn tuloksia voidaan tulevaisuudessa myös hyödyntää osana kurssisisältöä. Insinööriyön teoriaosuudessa avataan aiheeseen liittyvää termistöä ja käsitteitä. Työssä

esitellään käytettyjä ohjelmistoja sekä harjoituksiin liittyvät virtuaalimallit. Lopussa on vähän pohdintaa saavutetuista tuloksista ja niiden tuomista mahdollisuuksista opetuskäytössä samoin kuin myös jatkokehittymismahdollisuuksista.

2 Työn teoreettinen viitekehys

2.1 Asetetut tavoitteet

Tavoitteena tälle insinööriyölle on luoda neljä erilaista virtuaalimallia, joiden esikuvina ovat aidot mekatroniset laitteistot. CAD-suunnitteluohjelmistojen kuten CATIA V5 ja SolidWorks avulla malleihin tehdään tarvittavia muutoksia ja parannuksia, simuloinnin mahdollistamiseksi. Valmiit 3D-mallit viedään Siemens NX MCD-ympäristöön, jossa ne saadaan rakennettua lopulliseen muotoonsa. NX MCD-moduulilla ansiosta mallien rakennesille saadaan tehtyä välttämättömät määritykset, jotta ne voidaan myöhemmässä vaiheessa yhdistää onnistuneesti erilliseen PLC-editoriin. NX MCD- ja CODESYS-ohjelman välillä olevan yhteyden ansiosta sekä tarvittavien signaalien muodostamisen ja niiden linkittämisen jälkeen malleja voidaan ohjelmoida ja testata. Ohjelmoinnin näkyvät vaikutukset ovat heti havaittavissa MCD-ympäristössä.

Työskentelyprosessi on jaettu pienempiin osatavoitteisiin, jotka yhdessä muodostavat koko työn tavoitteellisen rakenteen. Kirjallisen osuuden lisäksi laaditaan kolme liitettä, jotka pitävät sisällään harjoitustyöt ja esimerkkimallin sekä lyhyen ohjeen, joka perehdyttää NX MCD- ja CODESYS-ohjelmistojen käyttöön. Kaikki liitteet yhdessä osoittavat, miten koko virtuaalisen käyttöönoton työprosessi oikein tapahtuu. Jokaiselle neljälle mekatronisen laitteen virtuaalimallille tehdään suurin piirtein samat toimenpiteet mutta laitekohtaiset eroavaisuudet niissä huomioiden. Näkökulmaksi tälle työlle on otettu aiheen koulutustarve. Mekatronisiin malleihin pohjautuviin PC-harjoituksiin sisällytetään kaksivaiheinen tehtäväosio, joka sisältää oppitunnilla tehtäviä perusharjoituksia. Insinööriyössä esitellään hieman yksityiskohtaisemmin yhden virtuaalimallin harjoitustyö sekä siihen liittyvä malliesimerkki kokonaisuudessaan läpikäytynä. Kirjallisen osuuden tavoitteena on käydä läpi työssä esiintyvät keskeisimmät ohjelmat ja käsitteet, jotka siihen on sisällytetty.

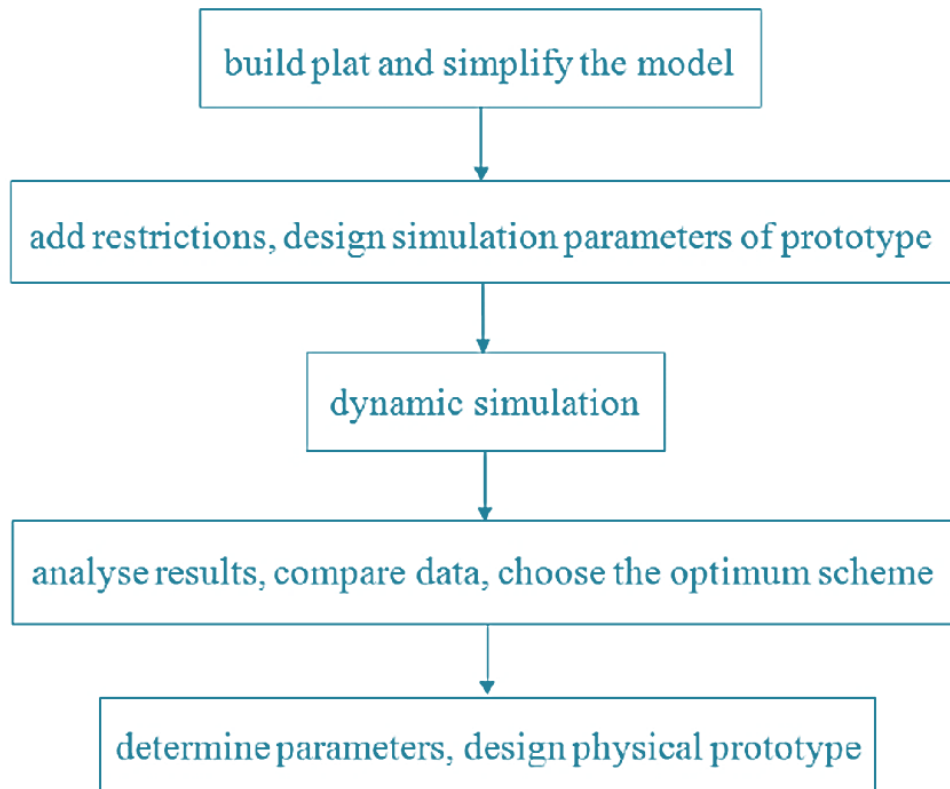
2.2 Virtuaalimallin suunnittelu

Virtuaalisuunnittelu perustuu esisijaisesti fyysisten ilmiöiden sekä erilaisten teknisten ratkaisujen mallintamiseen ja simulointiin. Ideoiminen sekä innovaatioiden suunnittelu ennen varsinaista toteutusvaihetta on näin ollen helpompaa, kun työssä käytetään siihen soveltuvia CAD-ohjelmia apuna. Virtuaalimallin tarkoitus on olla yleensä hieman yksinkertaistettumpi, kuin sen todellisuudessa oleva esikuva tai lopullinen tuote. Tämä asettaa myös mallintamiseen kohdistuvia rajoituksia, jotka on hyvä huomioida jo suunnitteluvaiheessa. Kolmiulotteisen virtuaalimallin parhaimpiin puoliin lukeutuu se, että jo luonnostelun aikana voidaan havainnollistaa tulevan tuotteen ominaisuuksia. Näitä ominaisuuksia tarkasteltaessa voidaan havaita myös mahdolliset suunnitteluvirheet sekä ilmi tulleet puutteet. Käytettävien ohjelmien ansiosta myös tarvittavat muutokset ja parannukset ovat tehtävissä virtuaalimalliin suhteellisen nopeasti ja helposti. Uudet suunnitteluohjelmat tekevät nykypäivänä työstä moninkertaisesti joustavampaa ja nopeampaa sekä ennen kaikkea kustannustehokkaampaa. Nykypäivänä ei ole enää välttämätöntä valmistaa kallista prototyyppiä ja testata sen toimintaa reaaliaikaisessa maailmassa. [8].

Virtuaalimallin parhaat puolet tulevat esille juuri silloin, kun fyysiset rajoitteet tulevat vastaan. Aina ei ole mahdollista edes valmistaa oikeaa prototyyppiä, ja syitä tähän voi olla useampia. Prototyyppi voi olla esimerkiksi monimutkainen valmistaa, jolloin syy voi olla kustannuskysymys, tai yksinkertaisesti sen fyysinen koko, kuten esimerkiksi maisemointi tai kaupunkisuunnittelutyö. Teknologiateollisuudessa juuri kustannustehokkuudella onkin nykyään kasvava painoarvo, mikä tarkoittaa sitä, kuinka paljon kyetään tuottamaan tai saadaan aikaan vaikutuksia suunnitelluilla budjeteilla sekä työpanoksen määrällä.

Kuten kuvassa 1 on esitelty, virtuaalisuunnittelu on hyvin johdonmukaista alusta loppuun saakka. Mallin rakentaminen aloitetaan niin, että vähitellen se alkaa lähestyä lopullista toteuttamista vaille olevaa valmista tuotetta. Menetelmää kutsutaan modulaariseksi mallintamiseksi, jossa virtuaalimalliin voidaan lisätä tai siitä voidaan myös poistaa osia, jos se rakentuu useammasta eri osa-alueen komponentista. Suunnittelu koostuu monista eri vaiheista, jolloin työ on hyvä aloittaa käytettävien materiaalien ominaisuuksien tutkimisesta ja valinnasta. Työhön soveltuvien käyttöohjelmien valinta on hyvin keskeistä, koska ohjelmiston valmistajia on olemassa useampia. Jotkut ohjelmistot ovat hyvinkin yhteensopivat. Ohjelmistojen valinnan jälkeen aloitetaan rakentamaan yksinkertainen malli, jolle

määrittellään halutut parametrit. Tätä ennen voidaan tehdä muutamia luonnoksia. Mallin dynaamisessa simuloinnissa tutkitaan annettujen parametrien vaikutusta itse malliin mahdollisimman tarkasti erilaisissa tilanteissa sekä olosuhteissa. Tämän jälkeen voidaan analysoida saavutettuja tuloksia ja suorittaa tarvittavia muutoksia kyseiseen malliin. Lopuksi työstä syntyy valmis virtuaalimalli, jonka kautta tehdään johtopäätökset tarvittaessa oikean prototyypin tai valmiin tuotteen valmistamiseksi. [12].



Kuva 1. Virtuaalimallin suunnitteluprosessi. [25].

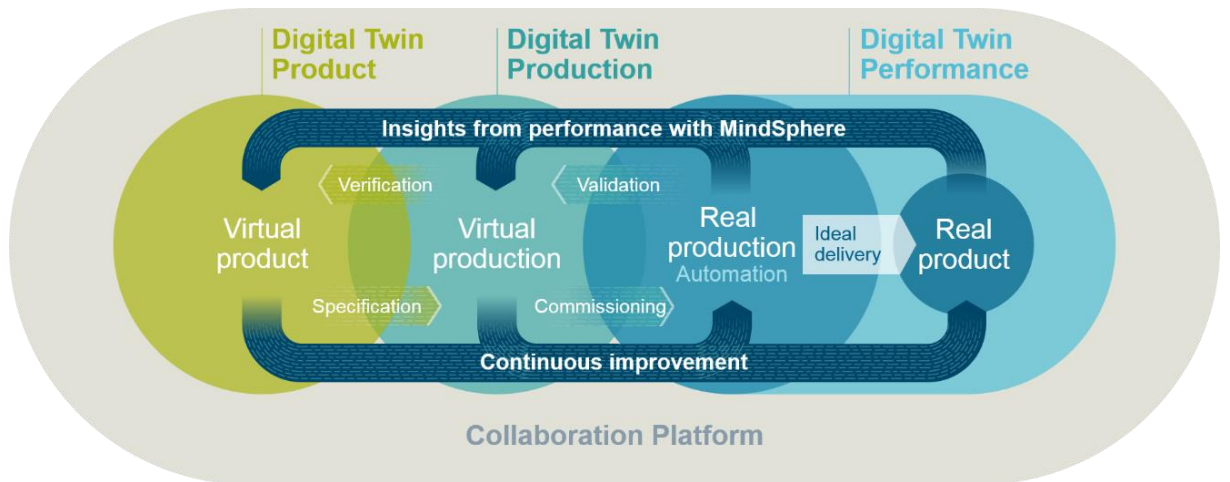
2.3 Digitaalinen kaksonen

Nykypäivänä elektroniikan valmistus on halpaa ja datan käsittely on tehostunut merkittävästi. Uutena asiana on tullut koneoppiminen, joka on mahdollistanut täysin uudenlaisen tavan lähestyä laitesuunnittelua. Nouseva teknologian trendi on digitaalinen kaksonen (**Digital Twin**) tai (**Cyber Twin**), jolla tavallisemmin puhekielessä tarkoitetaan fyysisesti olemassa olevan laitteen tai jopa kokonaisen teollisen tuotantolinjaston virtuaalista vastinetta, joka vastaa fyysistä kaksostaan sekä sijaitsee digitaalisessa ympäristössä. Tämän uuden teknologian parissa työskentelevät isot teknologiayhtiöt, esimerkiksi IBM Watson IoT ja GE Digital sekä Siemens PLM. Suomessa ehkä tunnetuin on Mevea, joka on simulointiohjelmia ja -tuotteita tarjoava yritys. [21, s. 23;22].

Digitaalinen kaksonen on käytännössä dataan perustuva matemaattinen malli. Tarve tämän kaltaiselle mallille syntyy silloin, kun tuotteen kehittämiseen suunniteltu aikataulu on tiukka ja resurssit rajalliset. Kaksosen luomiseksi käytetään CAD pohjaisia 3D-ohjelmistoja, joiden avulla kyetään simuloimaan mallin toimintaa. Uutta tuotetta suunniteltaessa voidaan verrata perinteistä käyttöönottoprosessia nykypäivänä yleistyneeseen virtuaaliseen käyttöönottoon, jonka etu piilee siinä, että suunnittelun aloitusvaiheessa pyritään todentamaan tuotteen kaikki mahdolliset ominaisuudet ja tarkastelemaan siitä kerättyä dataa tuotteen jalostamisvaiheessa. Tämän kaltaisen menettelytavan johdosta voidaan hyvissä ajoin saada varmistusta siitä, kuinka toimiva tuote tai linjasto olisi valmiina lopputuotteena, ennen kuin sitä on edes ehditty valmistamaan yhtäkään kappaletta. Digitalisaation ansiosta tuotekehityskin tehostuu, kuten ei tarvita resursseja kalliiden prototyyppien valmistamiseen ja niiden testaaminen myös investoinnin riskit pienenevät. [18].

Kuva 2 kertoo hieman digitaalisen kaksosen periaatteesta: kehittelyyn kuuluu mallin jatkuva parantaminen. Suunnittelussa tarkistetaan jatkuvasti, että prosessin kohde täyttää sille asetetut tietyt kriteerit. Digital Twinin ideassa pyritään myös ensisijaisesti suunnittelutyön turvallisuuteen toteutusvaiheessa, sillä esimerkiksi tuotantolinjaston toimintaa voidaan testata käyttämättä lainkaan oikeita ohjainlaitteita. Menetelmällä on suuria vaikutuksia myös investointeihin ja kulujen minimointiin, sillä mahdolliset toimintahäiriöt tai jopa ohjelmointivirheet kyetään seulomaan pois jo suunnittelutyön aikana. Ohjelmoitavan simulaation avulla voidaan myös ennakoita tietyissä asioissa, kuten esimerkiksi

kouluttaa henkilöstöä käyttämään laitteistoja edeltäpäin, hyödyntäen siis jo valmiiksi luotua virtuaalista ohjausjärjestelmää. Kerran luotua digitaalista mallia voidaan kehittää koko sen elinkaaren ajan, joka voidaan ulottaa aina kuluttajatasoon saakka. [14].



Kuva 2. Digital Twinin periaatekaavio. [26].

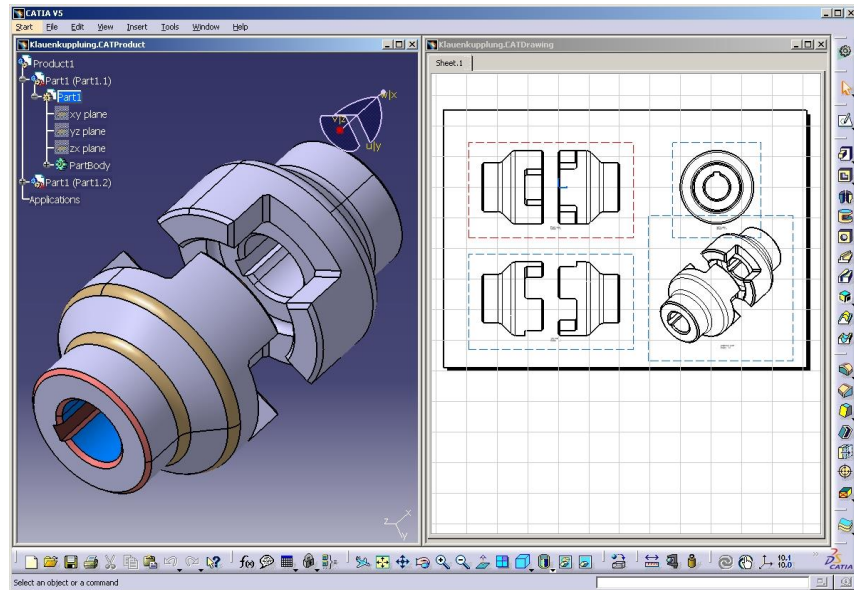
Digitaalisen mallin ohjaamista voidaan testata erillisellä PLC-editorilla, jolloin mallin käyttäytymistä voidaan ohjata ja havainnollistaa hyvinkin reaaliaikaisesti. Digitaaliselle mallille määritetyt fyysiset ominaisuudet sekä funktionaaliset rajoitteet saavat sen myös käyttäytymään, niin kuin se olisi todellisessa toimintaympäristössä. Toimiakseen juuri kuvatulla tavalla, vaatii se myös nopeaa informaation kulkua ohjelmistojen välillä eli toisin sanoen niiden on kyettävä ”keskustelemaan” lähes ongelmitta. Simuloinnin aikana on myös tarkoitus saada kerättyä dataa, jonka kautta voidaan tehdä arviointeja mallin ominaisuuksista.

2.4 Käytetyt ohjelmistot

Työskentelyn aloittamisen kannalta välttämättömäksi tulee hallita muutamia ohjelmistoja, joiden avulla on tarkoitus päästä asetettuihin tavoitteisiin sekä haluttuun lopputulokseen. Ohjaus- ja servotekniikan kurssilla rakennetut 3D-mallit toimivat työn kulmakivenä. Mallit on pääsääntöisesti suunniteltu käyttäen kahta CAD-ohjelmaa CATIA V5 ja SolidWorks. Vaikka laitteiden kokoonpanot olisi rakennettu kokonaan CAD-ohjelmalla, niitä ei voida kuitenkaan suoraan vielä ohjelmoida. Mallit on siirrettävä erilliseen simulaatioympäristöön, joka voidaan yhdistää erilliseen ohjelmoitavaan logiikkaan. Työssä käytetään CODESYS- ja TwinCAT PLC-editoria. Näitä samoja ohjelmistoja käytetään myös Metropolian insinööriopetuksessa. Ne ovat myös suosittuja yhtä lailla pienten, kuin myös suurtenkin ajoneuvo- ja laitevalmistajien keskuudessa. Kyseiset ohjelmistot soveltuvat erittäin hyvin juuri suunnittelun, tuotekehityksen sekä valmistuksen työkaluiksi.

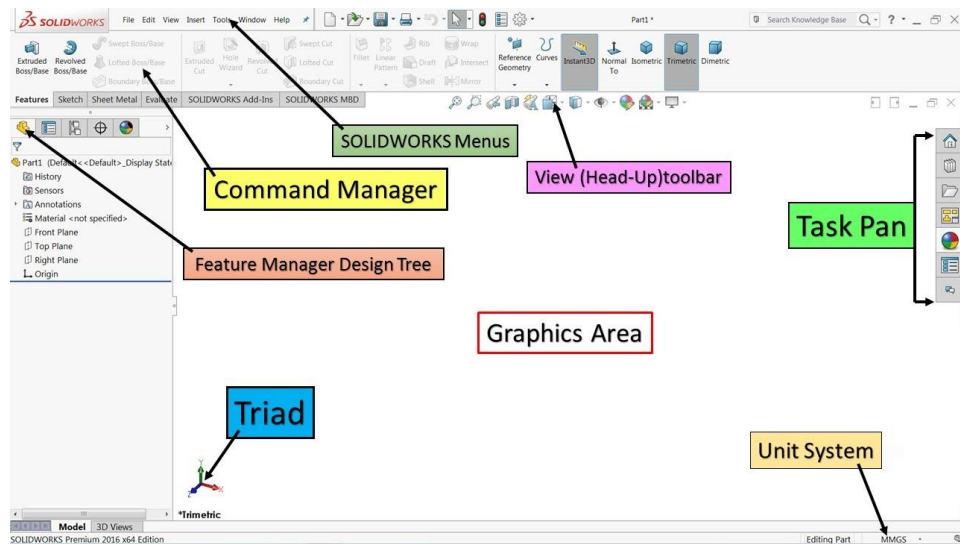
2.4.1 CATIA V5 ja SolidWorks

Virtuaalisten mallien luomisessa on käytetty kahta CAD-ohjelmaa, CATIA V5 ja SolidWorks. CATIA V5-ohjelma (kuva 3) on tietokoneavusteinen kolmiulotteinen interaktiivinen sovellus, joka on yksi monista CAD-mallinnusohjelmistoista. Ohjelmaa käytetään pääasiassa kolmiulotteisten mittatarkkojen kappaleiden mallintamiseen suunnittelussa. Ohjelmisto on Dassault Systemen kehittämä ja on alun perin ilmailunteollisuuden käyttöön suunniteltu, mutta nykyisin se on laajalti käytössä erityisesti lentokone-, auto- ja matkapuhelin- sekä konepajateollisuudessa. Ohjelmiston hyviin ominaisuuksiin kuuluu kappaleen parametrisointi sekä laaja valikoima erilaisia pintamallinnustyökaluja. CATIA V5 sisältää yhdeksän perusmoduulia, jotka sopivat suunnittelun eri osa-alueisiin. Moduuleita ovat muun muassa **Assembly Design**, jonka avulla voidaan rakentaa kolmiulotteisia kokonpanoja. **DMU-Navigator** ja **DMU-Space Analysis**-työkaluilla suoritetaan 3D -mallin parametriset paikannukset ja analysoidaan niitä mallista. **Drafting**-ominaisuudella voidaan luoda 2D-konepiirustuksia. **Generative Shape Designilla** voidaan muotoilla yksinkertainen mallin nopeasti esimerkiksi rautalankamallin pohjalta. **Part Designia** käytetään perustyökaluna kappaleiden mallinnuksessa. **DMU-Kinematics** on nimensä mukaisesti kinemaattinen mallinnustyökalu, kun taas **Knowledge Advisorin** kautta voidaan lisätä kaavoja tai sääntöjä, joka tarkistelee annettuja parametreja. **Generative Structural Analysis** toiminnolla tehdään huoliteltuja rakenne analyysseja. [24].



Kuva 3. CATIA V5-näkymä. [27].

SolidWorks (Kuva 4) on peräisin samalta ohjelmistojen kehittäväältä yhtiöltä kuin CATIA V5. Sen ohella se on myös yksi käytetyimmistä CAD-ohjelmistoista, joita hyödynnetään tietokoneavusteisessa suunnittelussa. Ohjelmiston soveltamismahdollisuudet ovat hyvinkin laajat, sillä 3D-kappaleiden mallintamisen ohella voidaan tehdä myös kokoonpanoja tai profiilirakenteista aina ohutlevymalleihin saakka. Ohjelmalla voidaan piirtää 2D-konepiirustuksia niin kuin CATIA-ohjelmistolla. Ohjelmistosta löytyy **eDrawings**-niminen työkalu, joka on suunniteltu CAD-tiedostojen tarkastelun ja jakamisen helpottamiseksi. Tämän työn kannalta käytössä oli suuri määrä valmiiksi mallinnettuja erilaisia koneiden rakenneosia sekä laitteistojen kokoonpanorakenteita, joita voitiin hyödyntää lopullisissa virtuaalisissa kaksosessa. Rakenneosat ja kokoonpanot on luotu pääsääntöisesti käyttäen CATIA- sekä SolidWorks-ohjelmistoja, koska ohjelmia käytetään myös opetuskäytössä Metropolia Ammattikorkeakoulussa. [2;3].



Kuva 4. SolidWorks näkymä. [28].

2.4.2 Siemens NX ja MCD-lisäosa

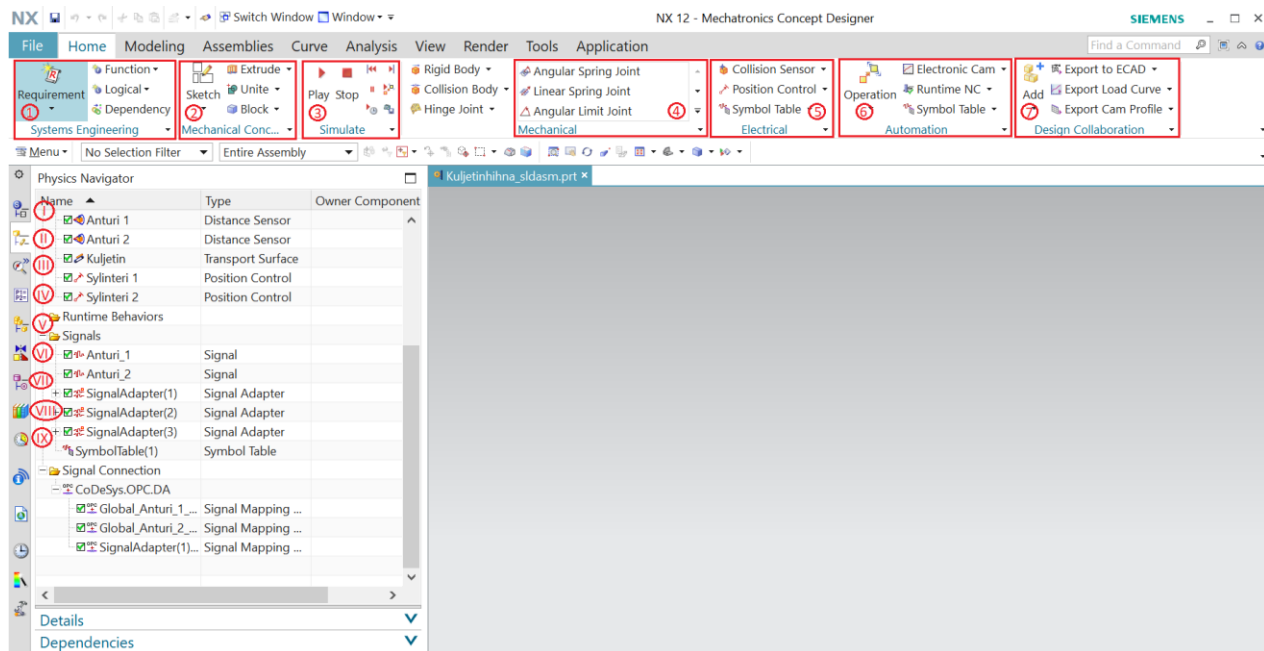
Siemens NX-ohjelmisto on hyvin laajalti käytössä teollisuusalaalla ympäri maailmaa. NX-ohjelmisto jaetaan usein karkeasti kolmeen eri osioon: suunnittelu, simulointi ja valmistus. Suunnitteluosiioon on sisällytetty tuotteen mallinnus sekä konseptointi. Ohjelmalla voidaan simuloida vaikkapa kokoonpanorakenteita tai sen erillisiä osia ja niiden yhteensopivuutta sekä mekaanista liikettä ylipäättänsä. Kaikki nämä osa-alueet on yhdistetty siten, että ohjelmalla voidaan luoda virtuaalimalli, jota kytetään myös simuloimaan. Ohjelmisto tukee myös muilla eri mallinnusohjelmilla tuotettuja 3D-mallitiedostoja, kuten esimerkiksi CATIA ja SolidWorks. [2].

Mechatronics Concept Designer (MCD) on NX-ohjelmiston CAD-tyyppinen lisäosa. Konseptisuunnittelu ja virtuaalinen käyttöönotto on huomattavasti joustavampaa juuri tämän lisäosan ansiosta. Se on hyvin monipuolinen ohjelmisto, koska sillä pystytään tekemään esimerkiksi CAD-, CAM-, CAE- tai PLM-toimintoja. Virtuaalimallissa on yleensä erilaisia liikkuvia osia, joiden halutaan liikkuvan juuri tietyllä tavalla sekä vaikuttavan toisiinsa. Mekatronisessa laitteessa näitä liikkuvia osia voivat olla esimerkiksi liukuhihna, moottori, työsylinteri ja ”työstettävä” tai ”lajiteltava” kappale. Kaikille keskenään vuorovaikutuksessa oleville kappaleille ja osille täytyy tehdä määritelmä eli miten sen halutaan käyttäytyvän ja onko se mahdollisesti vuorovaikutuksessa muihin kappaleisiin. [4].

MCD-simulaatiossa hyödynnetään Nvidia PhysX-fysiikkamoottoria, joka mahdollistaa sen, että kappaleiden väliset törmäykset ja niiden käyttäytyminen noudattelevat yleisesti fysiikan lakeja. Fyysisiin ominaisuuksiin sisältyy mallin koordinaatisto, massa, gravitaation suunta, nopeus ja kiihtyvyys. Nämä ovat mahdollista määrittellä simuloinnin virtuaalitodellisuudessa, jolloin se saadaan vastaamaan melkeinpä todellista ympäristöä. Näiden ominaisuuksien määrittämisen avulla malli myös käyttäytyy mahdollisimman aidosti sen omassa simulaatioympäristössä. Simulaatio ei myöskään noteeraa niitä rakenteita, joista tarvittavat fyysiset ominaisuudet puuttuvat kokonaan. [10].

NX MCD-lisäosa (Kuva 5) sisältää seuraavat tärkeimmät työkalut.

- (1) **Systems Engineering**; sisältää ominaisuuksia, joita voidaan hyödyntää toiminnallisen sekä johdonmukaisen mallinnuksen tekemiseen.
- (2) **Mechanical Concept**; osiosta löytyy työkalut CAD-mallin tekemiseen.
- (3) **Simulate**; on osio, josta simulaatiota voidaan ohjata tai jopa nauhoittaa video mallin toiminnasta.
- (4) **Mechanical**; pitää sisällään malliin sovellettavat fysikaaliset ominaisuudet, kuten nivel- ja törmäysominaisuudet.
- (5) **Electrical**; tästä osiosta saa anturien sekä toimilaitteiden määrittelemiseen tarvittavien ja siinä käytettävien **Signal Adapter**- ja **Collision Sensor**-työkalut.
- (6) **Automation**; työkalun avulla voidaan kontrolloida simulaation kinematiikkaa.
- (7) **Design Collaboration**; yksi tärkeimmistä työkaluista, jonka ansiosta voidaan simulaatioon tuoda osia toisista ohjelmista kuten CATIA tai SolidWorks.



Kuva 5. NX MCD-lisäosan yleisnäkymä.

- (I) **System Navigator**; tämän kautta voidaan tarkastella mallin funktionaalisia sekä loogisia ominaisuuksia.
- (II) **Physics Navigator**; näyttää luodut fyysiset rakenneosat, liitokset ja nivelet sekä sensorit ja signaalit.
- (III) **Runtime Inspector**; helpottaa testaamaan simulaation aikana mallin fyysisiä ominaisuuksia ja muuttaa niitä tai kytkeä jotain toimintoja pois.
- (IV) **Runtime Expression**; sisältää ominaisuuden tarkastella mallin fyysisten ominaisuuksien välisiä parametreja ja kaavoja.
- (V) **Assembly Navigator**; osio kertoo mallissa käytettävien osat ja näyttää kokonpanomäärittet.
- (VI) **Constraint Navigator**; on tarkoitettu siihen, että on helpompaa muuttaa joitakin kokonpanorajoituksia. Mitä enemmän rajoituksia on kokoonpanossa, sitä kauemmin NX:n lataaminen tai päivittäminen voi kestää. Tästä syystä on tärkeää säännöllisesti puhdistaa rajoituksia, joita ei enää käytetä kokoonpanossa.

(VII) **Part Navigator**; on oletusasetus, joka mahdollistaa näyttö- ja piilotustoiminnon. Kokoonpanon suunnittelu, rakentaminen ja tarkastelu ovat näin ollen sujuvampaa.

(VIII) **Reuse library**; toimii osakirjastona, eli useasti käytettyjä standardisoituja osia voidaan tallentaa tänne ja hakea aina uuteen kokoonpanon tarvittaessa.

(IX) **Sequence Editor**; tämän kautta hallinnoidaan aika- ja tapahtumapohjaisia operatioita simulaatiossa. [13].

2.4.3 NX MCD-moduulin käyttö

Seuraavaksi otetaan tarkastelun kohteeksi NX MCD:n käytön kannalta keskeisimmät työkalut ja niiden funktio tämän insinööriyön näkökulmasta. Nämä tiedot perustuvat Metropolian ”simulointi käyttöjen ohjauksen suunnittelussa”-kurssin NX MCD-opiskelumateriaaliin. Simulaation aloittamista varten on tehtävä muutamia alustavia toimenpiteitä virtuaalimallille. Silloin kun tiedetään ne kappaleen osat, jotka halutaan määrittää niin sanotusti ”liikkuviksi kiinteiksi” materiaaleiksi, tarvitaan siihen **Rigid Body**-työkalu. Valinta tehdään **Select Object**-toiminnon kautta, sillä se mahdollistaa samalla myös massan ja kappaleeseen tarvittavien voimien määrittämisen. Joskus saattaa syntyä tarve luoda kuljetinpintoja, on siihen myös käytettävä **Transport Surface**-toimintoa. Valitaan tasopinta, joka määritetään ”kuljettimeksi”, tämän jälkeen annetaan sille nopeus- ja suuntavektori. Hihnalla kulkevaan kappaleeseen käytetään **Collision Body** ja **Rigid Body**-työkalua, jolloin se liikkuu kuljetinhihnan pinnalla eikä siten ”putoa” sen lävitse. Tuotantolinjaston simulointia varten kätevin työkalu on **Object Source**, sen avulla voidaan tehdä valmiiksi määritettyjä kappaleita ja lisätä kuljetin pinnalle tasaisin väliajoin. Kappaleiden lisääminen voidaan suorittaa myös erillisen käskyn kautta, jolloin uuden kappaleen tuloa liukupinnalle voidaan määrittää määrällisesti sekä ajallisesti sopivaksi. Silloin kun **Object Source** määrittää kappaleet samankaltaisiksi, voidaan **Collision-** ja **Rigid body**-työkalulla luoda törmäyspinnat. MCD kykenee laskemaan tällöin myös jokaiselle kappaleelle törmäystilanteen sekunnilleen, mikä aiheuttaa myös sen, että simulaatiovaihe voi alkaa käymään turhan raskaanpuoleiseksi eli ohjelmien käyttö vaatii myös jonkinlaista laskentatehoa PC-laitteistolta. Simuloinnista voidaan myös saada **Object Source**-toiminnolla luodut kappaleet katoamaan, käyttämällä **Object Sink**-työkalua. Määritetään haluttu kosketuspinta, johon törmätessä kappale katoaa. Upotus-toiminnolla voidaan poistaa

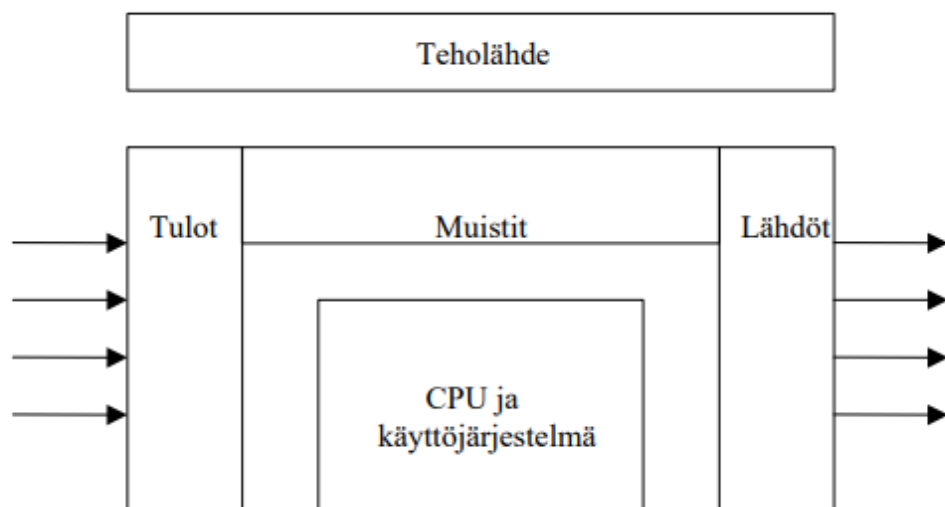
osia näkyvistä itse simulaatiomallista, jolloin samalla saada vapautettua hieman laskentatehoa tarvittaessa.

Collision Body-työkalulla kyetään kappaleelle määrittelemään haluttu törmäyspinta ja hallita sen liikesuuntaa. Kappaleelle täytyy siis määrittää törmäyspinta tai se putoaa liukupinnana lävitse ja saadaan aikaiseksi kappaleiden välistä vuorovaikutusta, on määrittäykset tehtävä oikein. On siis suositeltavaa luoda mahdollisimman yksinkertaisia törmäysmuotoja simulaation suorituskyvyn kannalta. Törmäysmuodon tarkkuuden voi määrittellä eri tarkkuudella, helposti käy niin että jos geometrinen tarkkuus on liian suuri, kappale saattaa altistua värinälle tai muille simulaatiovirheille. Törmäystarkkuutta voidaan muuttaa käyttämällä **Covex Factor**-toimintoa. Simuloinnin luonnin aikana tulee eteen myös tilanne, missä mallille täytyy luoda akseliliitoksia. Liitosten määrittäminen kappaleiden välille estää niiden vapaan putoamisen. Esimerkkinä tästä voidaan käyttää **Hinge Joint** eli saranaliitosta, jossa liike on rajoitettu vain tiettyjen akseleiden suuntaisesti, jolloin liikkeet rajoittuvat muuhun kuin akselin pituussuuntaiseen liikkeeseen. **Ball Joint** eli pallonivel toimii niin, että kappaleen liike on kokonaan lukittu yhden pisteen ympärille. **Fixed Joint** eli pitoliitos on hyvä vaihtoehto, jos halutaan että kappaleet pysyvät kiinni toisissaan ja eikä näiden välille saisi syntyä liikettä mihinkään suuntaan. Liitoksessa liikkuva osa on merkitty (Attachment), akselin vektori (Axis Vector) ja kiinnepiste (Anchor Point) sekä kiinteä osa tai runko (Base).

Anturoinnin toteutuksella voidaan simulaatiossa ohjata eri toimilaitteiden toimintaa, samalla tavalla kuin todellisessa toimintaympäristössä. Käytävissä on kaksi erilaista sensorointi mahdollisuutta, **Collision Sensor** havaitsee kaikki lähestyvät kappaleet. Toiminnon avulla kyetään tunnistamaan erilaisia muotoja sekä eri kokoisia kappaleita. Signaaliyhteyden avulla voidaan määrittää, mitä kuljettimen pinnalla liukuvalla kappaleella tapahtuu, kun anturi tunnistaa sen, eli pysähtyykö kappale vai jatkaako se matkaa. Jos anturimallin halutaan olevan tietyn tyyppinen, se voidaan muuttaa **Change Material**-toiminnon avulla voidaan valita ne kappaleet, joiden halutaan olevan vuorovaikutuksessa keskenään.

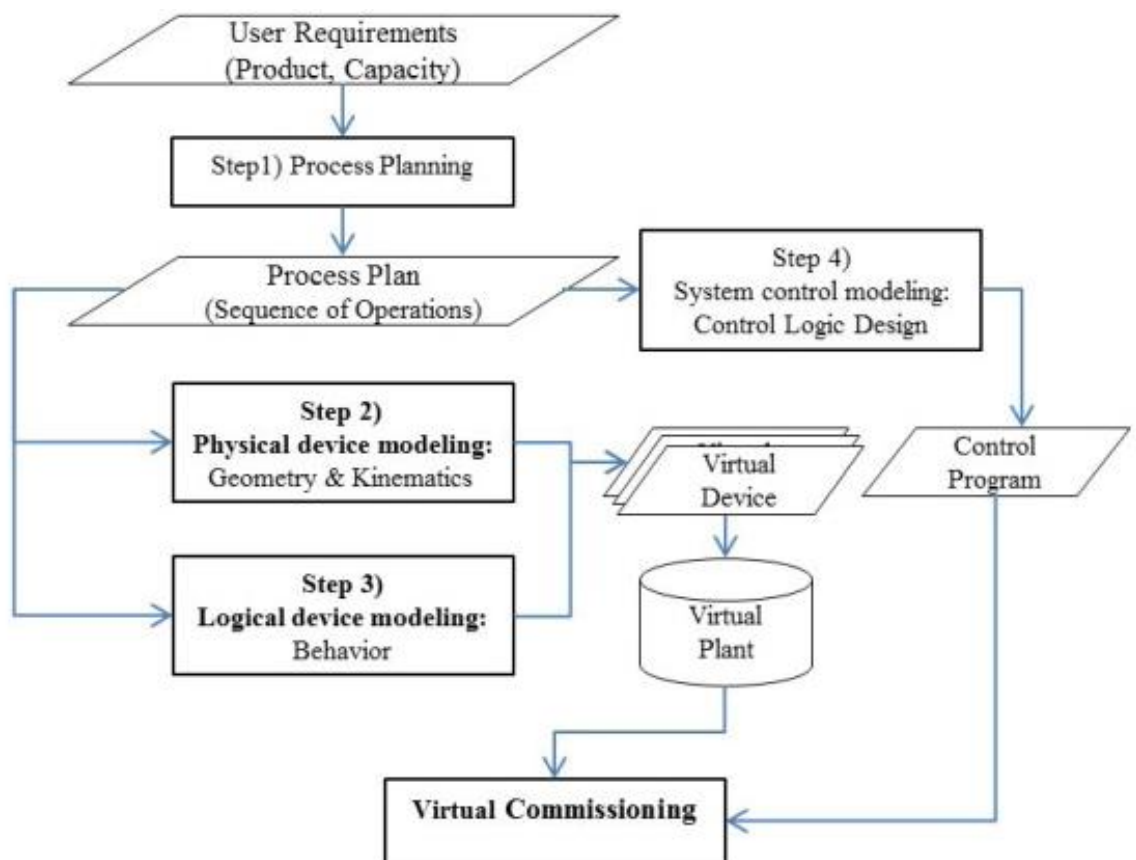
2.4.4 Yhteys ohjelmoitavaan logiikkaan

Ohjelmoitava logiikka (Kuva 6) on teollisuudessa käytetty automaation perustyökalu. Logiikalla ohjataan yksinkertaisia automaatio toimintoja kuten esimerkiksi kuljettimia ja prosessilaitteita. Ohjelmoitavilla logiikoilla kuten esimerkiksi PLC tai PC tarkoitetaan yleensä yhdellä tai useammalla mikroprosessorilla varustettua laitetta, joka kykenee ohjaamaan käytössä olevien laitteiden logiikoiden siirräntää (I/O) tai sitten väylän kautta muistissa olevien ohjelmien ja parametrien avulla. Mikroprosessori ohjaa logiikan sisäisiä toimintoja sekä käyttöjärjestelmää, se ohjaa myös viestiliikennettä logiikan sekä ohjelmointilaitteiden välillä. NX MCD sisältää ominaisuuden mitä muilta vastaavilta ohjelmistoilla ei ole, nimittäin sen kautta voidaan muodostaa yhteys ulkoiseen logiikkajärjestelmään eli tehdään niin sanottu virtuaalinen käyttöönotto. Yhteyden luomiseen käytetään OPC-yhteyttä, eli avointa liitettävyyttä avoimilla standardeilla. Standardi määrittää reaaliaikaisen tiedonvälityksen eri valmistajien järjestelmistä eli yhteisen rajapinnan. Yhteys ulkoiseen ohjelmoitavaan logiikkaan tehdään juuri OPC-serverin kautta. [5; 7; 15, s.14].



Kuva 6. Yksinkertaistettu logiikan lohkoakaavio. [29, s. 6].

NX MCD-ohjelmasta löytyy tähän tarkoitukseen suunniteltu toiminto **Signal Mapping**, tämä toiminto yhdistää ulkoisen logiikan omat signaalit NX MCD-ympäristössä luotuihin signaaleihin, mikä tarkoittaa tuloja sekä lähtöjä. **Signal Adapterissa** määritettyjen ohjaussignaalien ja toimilaitteiden välille muodostetaan linkki. Samasta valikkopaneelista yhdistetään ulkoisen logiikanjärjestelmän signaalit simuloituihin signaaleihin. Yhdistäminen tapahtuu valitsemalla halutut tunnisteet OPC-palvelimelta ja **Signal Adapterin** kautta. Menemällä **Signal Mapping**-valikkoon saadaan sieltä valittua **Client Parameters**. Sen avulla voidaan etsiä oikea OPC -palvelin (Kuva 7).



Kuva 7. Etenemisketju virtuaalisen käyttöönotossa. [30].

2.4.5 CODESYS ja TwinCAT

Kun mallin fyysiset ominaisuudet ja kappalekohtaiset määrytykset on tehty, voidaan alkaa ohjelmoida halutut liikkeet ja nopeudet siihen. Virtuaalisen mallin ohjelmointiin tarvitaan myös oma ohjelmistonsa. Käytössä on kaksi eri ohjelmaa, jotka toimivat samalla periaatteella, CODESYS sekä TwinCAT. Mallille määritellään oikeat signaalit, jotka tulevat virtuaalimallilta NX:n kautta OPC:hen. Kun signaaleja tehdään, voi niissä helposti syntyä sekaannusta ja tämän vuoksi ne kannattaa nimetä selkeästi ja oikein heti alkuvaiheessa, jolloin vältytään turhalta lisätyöltä.

Tällaista erillistä OPC-ohjelmaa käytetään usein lukemaan ohjelmoitavan logiikan tietoliikennettä ja rekistereitä, CODESYS (Controller Development System) on ohjelmoitavan logiikan ja sulautettujen järjestelmien ohjelmointiin tarkoitettu kansainvälisen IEC 61131-3 normin mukainen ohjelmointiympäristö automaatio-, logiikka- ja liikkeenohjaussovelluksiin. Ohjelman on suunnitellut saksalaisomisteinen ohjelmistoyritys 3S-SmartSoftware Solutions. PLC-laitteistoille (Kuva 8) tarkoitettujen demolisenssien avulla käyttäjä pääsee tutustumaan erilaisiin sovelluksiin ja ratkaisuihin täysmääräisesti. Kyseinen ohjelmointiympäristö yhdistää monia teollisuusautomaation erilaisia ohjelmia samassa ympäristössä, sillä CODESYS on kehitysympäristö, väyläkomponenttien ohjausjärjestelmä, I/O-konfigurointityökalu, visualisointi ja liikkeenohjaus, jossa on omana osana CNC, SoftPLC. Ohjelmistoa on myös mahdollista laajentaa erikseen ladattavilla lisämoduuleilla sekä kirjastotiedostoilla. [1; 15, s. 8; 18].

TwinCAT-ohjelmisto on rakennettu CODESYS-ympäristön pohjalle ja niin ollen sen osaamisella päästään hyvin nopeasti CODESYS ohjelmoinnin ytimeen. TwinCAT on hyvin samankaltainen myös toiminnoiltaan, mutta ohjelmien syntaksi poikkeaa hieman toisistaan. TwinCAT- ja CODESYS-ohjelmien hallitseminen tukevat toisiaan oppimisen kannalta. TwinCAT-ohjelmistossa on järjestelmiä, joilla ohjaukset voidaan suorittaa reaaliajassa, kuten myös ohjelmointi ja diagnostiikka. TwinCAT-ohjelmistolla on myös mahdollista synkronoida lähestulkoon mikä tahansa PC-tyyppinen järjestelmä yhteensopivaksi erillisen ohjelmoitavan PLC-logiikan kanssa. TwinCAT-ohjelma käyttää samaa standardin mukaista ohjelmointikieltä kuin CODESYS, ja sen ohella voidaan käyttää myös Beckhoffin luomaa CFC-kieltä (Continous Function Chart). [6; 22].

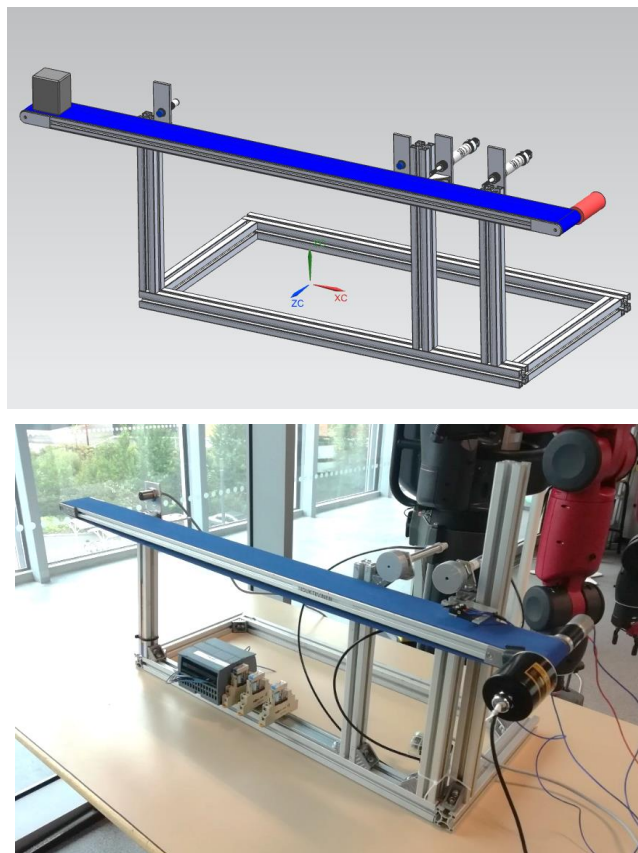
OPC on laajalle hyväksytty teollisuuden kommunikoinnin standardi laitteiden ja ohjelmistojen välillä. OPC-yhteys mahdollistaa jatkuvan yhteyden PLC:n, laitteiden, HMI:n (käyttöliittymien) ja ohjelmistojen välille. Yhteyden voi luoda eri valmistajien laitteiden ja ohjelmistojen välille. OPC-standardissa on monia ohjeita loppukäyttäjille ja ohjelmoijille. Nämä standardit tarkastelevat ohjelmistojen sekä palvelimien rajapintojen välistä yhteyttä. Standardi sisältää myös kahden tai useamman palvelimenvälisen yhteyden luomisen. Näiden yhteyksien välillä voidaan välittää ja seurata dataa reaaliaikaisesti. Yhteydellä voidaan myös seurata laitteiden välistä dataa. Tähän kuuluu myös tapahtumahistorian ja muiden ohjelmistojen seuraaminen. Vuonna 1996 OPC-standardi julkaistiin ensimmäisen kerran, ennen tätä laitteistojen ja ohjelmistojen välille ollut vielä olemassa minkäänlaista standardisoitua yhteyttä. Ennen standardin julkaisua jokaiselle laitteelle ja ohjelmistolle luotiin aina erillinen yhteys. Ongelmana oli, että luotu yhteys ei aina välttämättä toiminut enää saman laitteen ja ohjelmiston välillä, ja tämän kaltainen järjestely tuotti paljon ylimääräistä työtä onnistuneen yhteyden aikaansaamiseksi. [7].

2.5 Virtuaalimallit

Virtuaalimallilla tarkoitetaan tietokoneella suunniteltua kolmiulotteista mallia, jossa kappaletta pystytään tarkastelemaan eri kuvakulmista reaaliajassa. Insinööriä varten mallinnettavia mekatronisia laitteita on kaiken kaikkiaan yhteensä neljä erilaista, jotka löytyvät myös Metropolian Myyrmäen kampuksen koneautomaatiolaboratoriosta. Aikaisempien vuosikurssien aikana niistä on tehty useampia eri versioita, joista oli tarkoitus etsiä ja valita sopivimmat juuri tähän työhön. Pienillä muutoksilla niistä saadaan räätälöityä PC-harjoituksiin juuri sopivat, joihin luodaan yhteys myös CODESYS-logiikalla. Tarkoitus niissä on harjoitella 3D-mallintamista, simulointia, logiikkaohjauksen tekemistä sekä virtuaalista käyttöönottoa. [16].

2.5.1 Kuljetinlajittelija

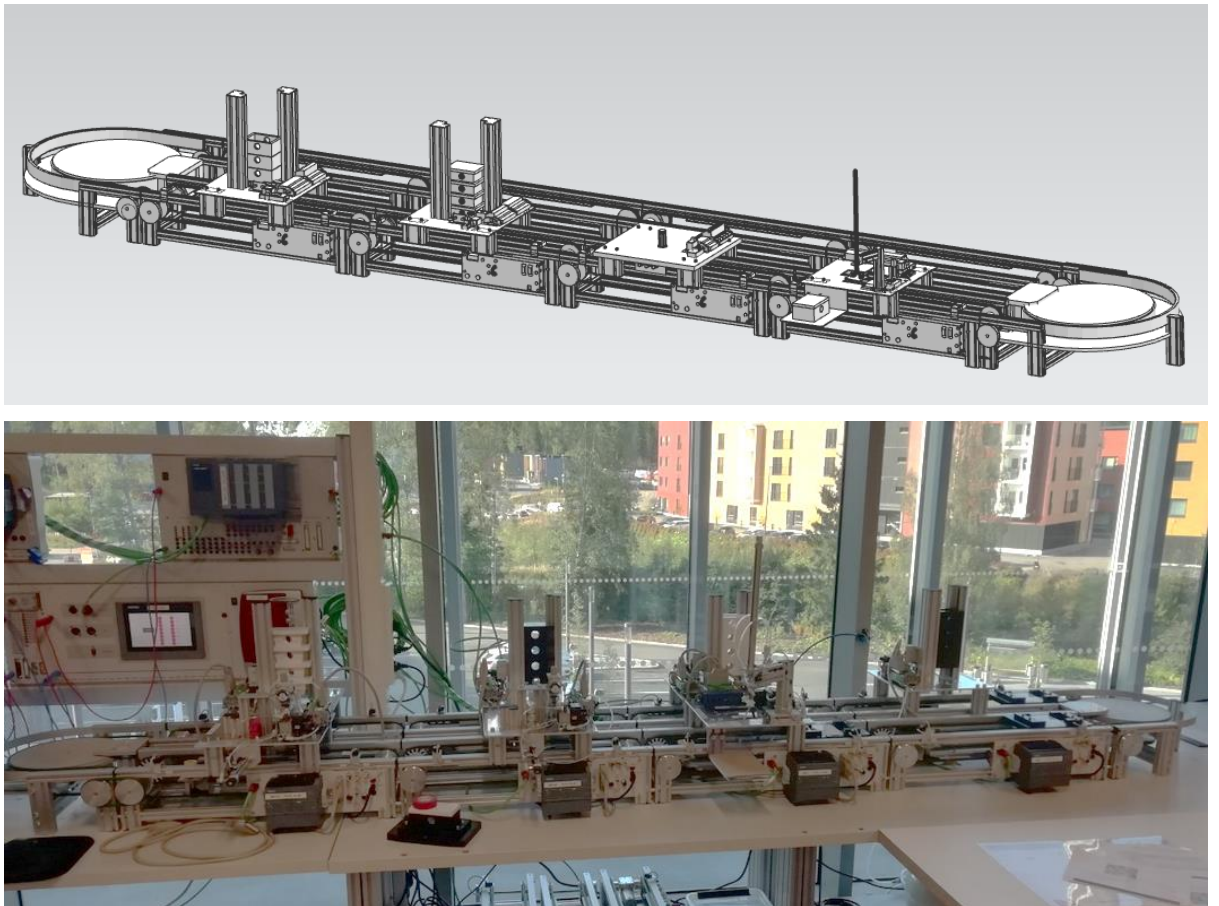
Yhdeksi virtuaalimallin pohjaksi valikoitui lajittelija (Kuva 8). Se on toteutettu niin, että se soveltuu erityisesti kappaleiden liikutteluun ja lajitteluun siinä olevan liukuhihnarakenteen vuoksi. Kuljettimessa on kaksi työsylinteriä sekä kaksi kappaletta tunnistelevaa anturia. Hihnalle tehdään virtuaalinen käyttöönotto, käyttäen Siemens NX-ohjelmaa. Mallinnus on tehty osittain SolidWorks- ja CATIA-ohjelmistolla. Alun perin mallin luomisessa on hyödynnetty traceparts.com -sivustoa, jolla on tarjolla ilmaiseksi ladattavia CAD-part tiedostoja. Valmiiden osien avulla saa helposti luotua hyvin autenttisen näköisen mallin, lisäämällä siihen taidokkaasti tehdyt anturit ja sylinterit. Varsinainen työvaihe NX-ohjelmistolla on luoda mallille liikeradat eli simuloida kaikki halutut toiminnot. Lajittelijan osalta liikkuviksi kappaleiksi määritellään työsylinterit ja liukuhihna sekä lajiteltava kappale. Kuljettimen antureille määritetään signaalit. Liukuvia liitoskohtia kuljettimessa ovat myös työsylinterit, koska ne koostuvat kahdesta eri osasta. Kuljettimen toimintaperiaate on, että käyttäjä voi valita lajittelun periaatteeksi joko muodon tai värin, joita halutaan linjaston lajittelevan.



Kuva 8. Kuljetinlajittelija.

2.5.2 Smart Factory

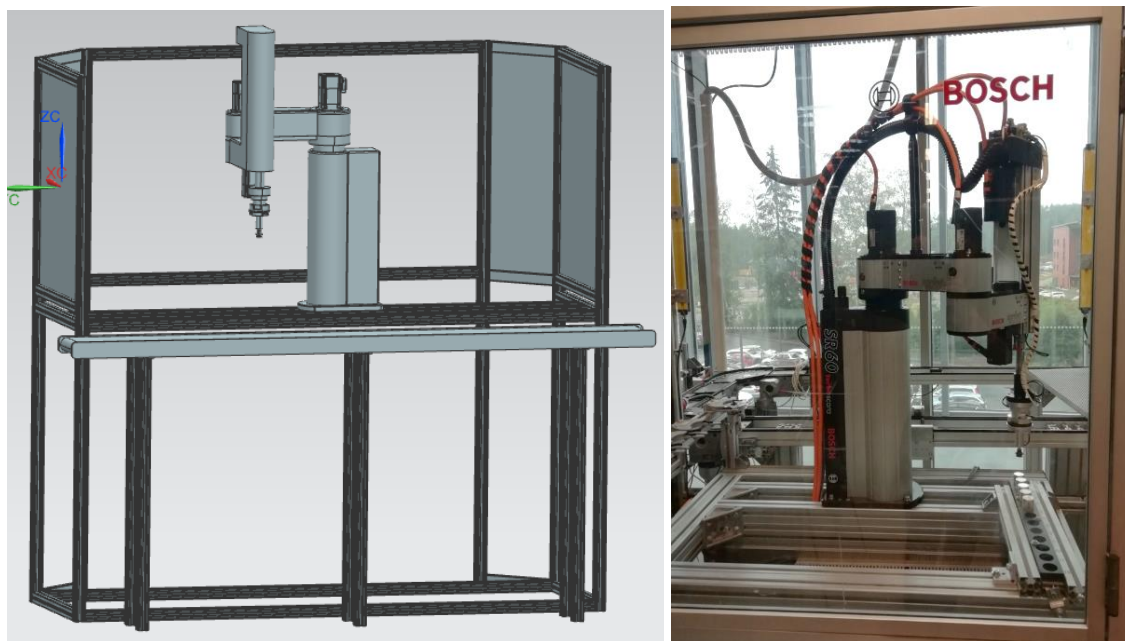
Mallipohjana toiselle digitaaliselle kaksoselle on Lucas-Nüllen säätiön koulutustarkoitukseen suunniteltu SmartFactory, niin sanottu ”älykätehdas” eli kokoonpanolinjaston jäljitelmä (Kuva 9). Malli on pienoiversio modernista ja älykkästä sekä automatisoidusta kokoonpanolinjastosta, jossa on kaksi eri kuljetinta yhteen liitettynä. Mallissa on myös käytetty valmiiksi mallinnettuja CAD-part-tiedostoja lopullisessa kokoonpanossa. Osien digitaaliset mallit on ladattu valmistajan omilta kotisivuilta. SmartFactoryn liikkuvia komponentteja ovat muun muassa linjastolla kulkeva kelkka, kansia syöttävän makasiinin pidikekynnet, työsylinteri ja liukuhihna. Linjastolla koottavan kappaleen osia ovat kansi, runko ja rungon läpi pujotettava sylinterimäinen tappi. Linjastolla voidaan koota kappale kolmesta eri osasta, joiden väriä tunnistelee kaksi optista anturia.



Kuva 9. SmartFactory-kokoonpanolinjasto.

2.5.3 SCARA-robotti

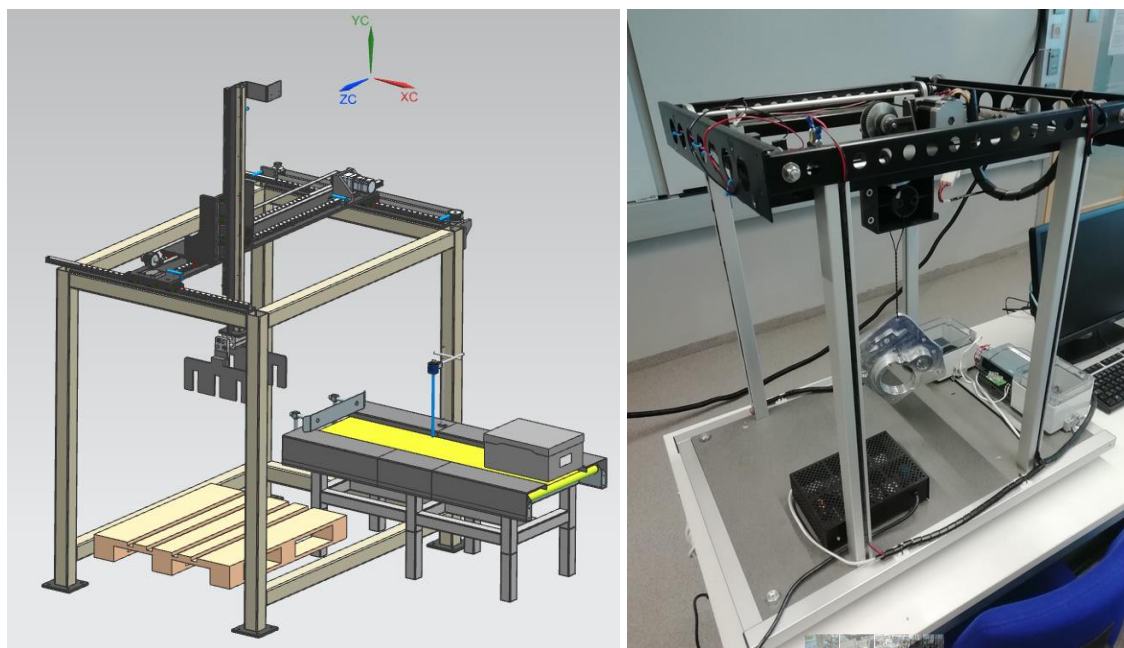
Metropolian automaatiolaboratoriosta löytyy myös SCARA-robotti (Selective Compliant Assembly Robot Arm) (kuva 10), jonka pohjalta luodaan myös virtuaalimalli. SCARAn tarkoitus on jäljitellä ihmisen käsivartta, mutta nivelet on suunniteltu siten, että olka- ja kyynärvarsi kääntyvät ainoastaan horisontaalisessa suunnassa. Kyynärvarren ulommaisessa päässä on lisäksi vertikaalisessa suunnassa liikkuva erillinen työkaluinstrumentti. SCARAn ohjaus vaatii käänteistä kinematiikkaa, jos halutaan simuloida myös lineaarista liikettä. Koulun laboratorion Bosch SR60-tyyppiselle SCARA-robotille oli ennuudestaan olemassa valmis virtuaalimalli aikaisemman kouluprojektin jäljiltä. SCARAn ympärille oli mallinnettu erillinen työsolu sekä pieni osa kuljetinhihnastoa. SCARAn osat on mallinnettu pääsääntöisesti CATIAlla. NX MCD-ohjelmalla määritellään liikkuvat osat, kuten esimerkiksi nivelet ja työkalukärki sekä liukuhihna. Toimielimille määritetään OPC-signaalit sekä törmäyspinnat. Liukuhihnalle on lisätty ”työstettävä” kappale sekä sitä tunnistava anturi.



Kuva 10. SCARA-robotti.

2.5.4 XYZ-pienoisnosturi

Mekatronisista laitteista viimeisin on XYZ-pienoisnosturi (Kuva 11), jolla on nimensä mukaisesti X-, Y- ja Z-suuntaiset liikeradat. Nosturin osat on mallinnettu CATIalla esikuvansa mukaisesti ja muuntaminen virtuaalimalliksi tehdään samalla tavalla kuin aikaisimmissa malleissa. Nosturissa oleva nostinkelkkaa liikutellaan askelmoottoreilla liikuteltavassa kehikossa X- ja Y-suunnassa, nostovinssi liikkuu mallissa Z-suuntaisesti eli ylös tai alas. Moottoreille luodaan OPC-signaalit ja ohjaus tapahtuvat PLC-ohjelmalla. Liikeradoilla kuten X ja Y on alkuasetat sekä päätyrajat. NX-ohjelmistolla mallin osat määritellään **Rigid Bodyksi**, jolloin niihin vaikuttavat ulkoiset voimat kuten "gravitaatio". Tarpeelliset pintakohdat määritettiin **Collision Bodyksi**, jolloin osat eivät pääse liikkumaan toistensa läpi eli muuttuvat konkreettisiksi toisilleen. Digital twin toteutetaan olemassa olevan CAD-mallin pohjalta. NX MCD-ympäristössä nosturin kelkkaa voidaan liikuttaa erillisellä paikkakomennolla, mutta CODESYS-ohjelman kautta avataan linkki PLC_PRG ja asetetaan **False/True**-toiminto päälle. Nosturissa olevat rajakytkimet antavat päätyjen rajasignaaleit.



Kuva 11. XYZ-siltanosturi.

3 Työn suunnittelu ja toteutus

3.1 Taustatiedot

Insinööriyön alkuvaiheessa on tärkeää määritellä työlle asetetut tavoitteet ja mitä toimintoja niiden saavuttamiseksi lopulta tarvitaan. Tärkeää on tutustua tarvittaviin ohjelmiin ja ymmärtää, miten OPC-signaalit luodaan ja miten CODESYS-ympäristö toimii. Tarkasteltavia virtuaalimallipohjia on tarjolla paljon. Jonkinlaista kartoitusta on tehtävä ja jokainen niistä on katsottava läpi ja tutkittava, mitkä niistä mahdollisesti vastaisivat työn luonnetta vastaavaa käyttötarkoitusta, joista voidaan rakentaa tarkat digitaaliset kaksoset. Etsinnän ja seulontavaiheen jälkeen on tutkittava malleilla olevat toiminnot ja mahdolliset puutteet, jotta voidaan arvioida, mitä on suunniteltava kunkin eri mekatronisen mallin kohdalla erikseen. Jokaisesta mallista löytyy kaikki tarvittavat perusrakenneosat. Mallien isoimmat puutteet ovat pääsääntöisesti anturit ja signaalien määritykset, jotka on tarkoitus luoda uudelleen ja testata niiden toiminta. Signaalien määritysten jälkeen ne voidaan testata ja mallin toimivuutta voidaan simuloida ja tehdä siitä halutunlainen.

Signaalit luodaan MCD-ympäristössä ja ne täytyy siirtää CODESYS-ympäristöön, OPC-yhteyttä käyttämällä. Signaaleja luodessa täytyy myös miettiä malliharjoituksen oppimistavoitteita eli mitä halutaan oppia ja miten se kannattaa esitellä. Harjoitustöiden suunnittelu ja pohdinta toimii siis koko ajan ajatuksena taustalla siinäkin vaiheessa, kun mallien anturointeja ja muita toimintoja luodaan. Lopullisessa digitaalimallissa voidaan CODESYS-ohjelmalla ikään kuin "ajaa" haluttu toiminta suoraan digitaaliselle kaksoselle.

3.2 PLC-harjoitukset

Yksi tärkeimmistä työn tavoitteista on siis suunnitella kaikista aikaisemmin esitellyistä mekatronisista laitteista kaksiosainen harjoitustyö, johon kuuluu selkeä työohje sekä kyseiseen tehtävään myös malliratkaisu. Harjoitukset on suunniteltu niin, että niiden suorittamiseen käytetään aikaa noin yhdestä kahteen tuntiin. Työohjeesta käy ilmi, mitä harjoituksessa on tarkoitus oppia, mitä tehdään kyseisen harjoituslaitteen kohdalla, sekä mitä harjoituksen kokonaissuorittamiseen vaaditaan. Harjoituksessa on siis kaksi erillistä tehtävää, joihin sisältyy perusharjoitus ja vähän vaativampi työ. Työohje sisältää yleisen

osion tehtävästä harjoitteesta ja työohjeen. Jokaiseen neljään työohjeeseen on olemassa myös malliratkaisu, jonka tarkoitus on selventää, kuinka harjoitus tulisi tehdä aina alusta loppuun saakka. Insinööriyön osalta yhdestä harjoitustyöstä tehdään PLC-ohje ja se esitellään kokonaisuudessaan ja kattavasti, muista harjoitteista tehdään lyhyempi tiivistelmä, miten ne on tehty, sillä periaate on kaikissa kuitenkin samalainen.

3.2.1 Työohjeet

Liite 1 käsittää työohjeet kaikista neljästä laitteistosta. Työohjeessa on pyritty noudattamaan selkeää esitystapaa, josta ilmenee kaikki tarvittava tieto työn suorittamiseksi. Työohjeessa esitellään, mitä harjoituksessa on tarkoitus opetella ja mitä harjoitukseen sisältyy, tämän tarkoitus on avata tehtävän luonnetta ja tavoitteita. Ohjeesta käy myös ilmi, millä ohjelmilla työssä operoidaan, sekä esitellään kaksiosainen tehtävä. Harjoituksessa käytetään valmiita virtuaalimalleja, joihin luodaan yksinkertainen logiikkasovellus. Jokaisessa ohjeessa esitellään lyhyesti myös harjoituksen kohteena oleva virtuaalimalli, miten se toimii ja mistä se rakentuu.

Työohjeessa avataan logiikkasuunnittelua siltä osin, että mitä lohkoavioon tarvitaan, kuten esimerkiksi (I/O) lähdöt on valmiiksi nimetty sekä tarvittavat muistipaikat. Harjoitukset on suunniteltu niin, että ne pystyttäisiin tekemään aina yhden oppitunnin aikana. Tehtäväosiot on esitetty niin, että työn eteneminen olisi hyvin johdonmukaista ”Step by step”-tyyppisesti. SmartFactory-työohje toteutetaan miniprojektina, sillä se sisältää eniten ohjelmitavia toimintoja kuin muut harjoitteet. Kaikissa harjoitteissa käytetään pääsääntöisesti CODESYS PLC-editoria, mutta SCARA-robotin logiikkasovelluksessa hyödynnetään poikkeuksellisesti TwinCat-ohjelmistoa.

3.2.2 PLC/ PC-malliharjoitus

Liite 3 on esitetty malliratkaisu liitteessä 1 olevaan työohjeeseen, jossa esitellään tehtäväosiot vaiheittain läpi. Malliharjoituksessa on tarkoitus auttaa ymmärtämään, kuinka kyseinen harjoitustyö tehdään ja mitä asioita sen tekemisessä on otettava huomioon. Malliratkaisu on jaettu työohjeessa olevien tehtävien mukaisesti kahteen osaan, jonka tarkoitus esitellä selkeästi ja hyvin loogisesti. Tehtäväosioden yhteydessä mainitaan aina, mitä ohjelmaa käytetään missäkin vaiheessa.

Malliharjoitus aloitetaan yksinkertaistetulla kuvalla MCD:stä löytyvästä **Physics Navigator**-kuvakkeesta ja myös sillä, miten ohjaussignaalit luodaan tiettyihin rakenneosiin. Malliharjoitukseen on liitetty runsaasti välivaihekuvia havainnollistamisen parantamiseksi. Signaalien luontien jälkeen käydään läpi, miten niitä käsitellään, esimerkiksi niiden siirtäminen **Inspector**-näkömään. Onnistuneen signaalien muodostuksen jälkeen suoritetaan vaiheittain ohjelmistojen yhteyden muodostus. MCD-ympäristössä olevan malli ja CODESYS-logiikka yhdistetään seuraavaksi. Malliesimerkin johdonmukaisuuden säilyttämiseksi ja ymmärrettävyyden helpottamiseksi siinä on selkeämmin mallin fyysiset osat, sensorit ja toimilaitteet sekä signaalit. Harjoituksessa tarvittavat signaaliyhteydet on myös merkitty sekä raja-arvot annettu kullekin erikseen. Malliesimerkin lopussa on yhteenveto harjoituksen sisällöstä ja kertaamisesta sekä mahdollisista ongelmatilanteista.

3.2.3 Signaalien luominen

Liite 2 täydentää malliharjoitusta itsessään, sillä ennen kuin varsinainen logiikkaharjoitus aloitetaan, on tehtävä tarvittavat signaalien määrittäykset, jotta digitaalista kaksosta voidaan edes ohjelmoida. Signaalien luomista varten on tehty oma ohje, jossa käydään lävitse harjoituksessa tarvittavat signaalityypit, miten niitä käytetään ja mitä pitää ottaa huomioon niitä tehdessä. Signaalien tekeminen on pyritty esittämään hyvin yksiselitteisesti. Ohjeessa on eritelty NX MCD- ja CODESYS-ohjelmien käyttö erikseen, niin että ohjeessa edetään kronologisesti eteenpäin, jotta signaalit ja yhteyden muodostus PLC-editoriin tehdään mahdolliseksi. Ohje sisältää runsaasti havainnekuvia, joita on täydennetty erikseen erillisellä selostuksella.

4 Tuloksien ja tavoitteiden saavuttaminen

Kaiken kaikkiaan insinööriyössä asetettuja välitavoitteita oli suhteellisen monta. Näistä tavoitteista saavutettiin muun muassa se, että saatiin ensimmäiseksi suoritettua seurlontaa, jonka tuloksena löytyivät sopivimmat virtuaalimallit PLC-harjoituksia varten. Malleista on tehty selkeät omat kansiot, joissa jokaisesta laitteesta löytyy myös työohje ja malliharjoitus pohja. Työohjeiden toteutus oli seuraava askel työssä. Virtuaalimalleista täytyi selvittää niiden ominaisuuksia, jolloin voitiin myös arvioida ja suunnitella laitekoh- taiset työohjeet. Tärkeää oli myös luoda NX MCD- ja CODESYS-ohjelmien käytölle käyt- töohje, josta käy yksiselitteisesti ilmi miten digitaaliset mallit lopulta viimeistellään. Käyt- töohje on suunniteltu siten, että se tukee varsinaista malliratkaisua, joka auttaa ymmär- tämään harjoitustyön rakenteen sekä työvaiheet. Työssä avataan muutamia ammattikä- sitteitä, jotta saadaan aiheesta selkeä yleiskuva ja ymmärrettävyys helpottuu.

Tehdyn työn perusteella aloitettiin luomaan kattavaa harjoituspakettia neljästä erilaisesta mekatronisesta laitteesta. Liitteistä löytyy työn toteutusohjeen esimerkki PC-harjoitusta varten. Tämän työn tuloksena myös Metropolia saa tulevaisuuden projektiaan eteenpäin. Jälkikäteen tarkasteltuna voidaan todeta, että työn aloituspalaverissa esille tulleiden ta- voitteiden täyttäminen vaati huomattavasti enemmän ymmärrystä itse aiheesta sekä käytettävistä ohjelmistoista, kuin saattoi etukäteen kuvitella. Haasteellisinta oli myös lo- giikkalohkojen suunnittelu ohjelmointia varten, joka vaati hieman opiskelua aiheesta, koska lohkot eivät olleet enää muistissa. Haastetta toivat myös ohjelmistot, joista ei ollut entuudestaan minkäläistä kokemusta. Työssä ilmi tulleet ongelmat kyettiin kuitenkin rat- kaisemaan, minkä ohella tapahtui myös paljon opiskelua ja uuden oppimista koko ajan.

Insinööriyön kautta saavutetut asiat toimivat hyvänä teoriapohjana harjoitustöiden aloit- tamiselle. Työstä syntyi erittäin hyvä pohja harjoitusten suorittamista varten. Harjoitus- töiden suunnittelussa liikkeelle on lähdetty siitä olettamuksesta, että harjoitusten suorit- tamiseksi ei tarvitse olla entuudestaan asiaan liittyvää tietotaitoa. Samaa näkökulmaa on pyritty pitämään myös insinööriyön kirjoitusasussa ja sen selkeässä sekä johdonmukai- sessa ulkoasussa. Kielessä on myös pyritty välttämään liiallista ammattisanaston käyt- töä, jotta ymmärrettävyys säilyisi.

5 Yhteenveto

Insinööriyössä käsiteltiin ohjaus- ja servotekniikan PC-harjoitusten suunnittelua ja toteutusta, niin teoriassa kuin myös käytännössä. Työn aloittaminen oli suhteellisen johdonmukaista, mutta haasteita toi työssä tarvittavien ohjelmien käyttö, sillä kaikkea tarvittavaa tietotaitoa ei ollut valmiina taustalla. Uuden oppimista on siis tapahtunut jatkuvasti työn alusta aina loppuun saakka. Aikaisempaa kokemusta löytyi jonkin verran NX MCD-ohjelmistosta, mutta CODESYS tuli aivan uudeksi asiakki, jolloin aikaa jouduttiin käyttämään myös selvitystyöhön ja ohjelmien perustoimintojen harjoitteluun. Teoriaa ja lähtötietoja piti kartuttaa, jotta kynnys madaltuisi itse varsinaisen työn aloittamiselle. Ohjelmistojen käyttöön ja toimintoihin täytyi siis perehtyä syvällisemmin, jotta asetetut tavoitteet voitaisiin saavuttaa. Työlle piti ajoittaa pieniä välitavoitteita, jotta aikataulussa pysyminen olisi mahdollista.

Virtuaalisen 3D-mallin toimintaympäristö luo uusia mahdollisuuksia automaatiotekniikan opiskelun sisältöön, ja se on myös askel eteenpäin, sillä virtuaalimallien käyttö on arkipäivää tämän päivän teollisuudessa. Virtuaalisen mallin käyttö mahdollistaa sen, ettei laitetta ole pakko olla fyysisesti olemassa. Tämä mahdollistaa myös sen, että opiskelijat voivat testata ja kokeilla laitetta virtuaalisessa ympäristössä omalla koneellaan ja turvallisessa ympäristössä, milloin vain. Virtuaalimalli tarjoaa myös edullisen tavan harjoitella laitteen suunnittelua ja ohjelmointia, mutta se ei kuitenkaan koskaan tule täysin korvaamaan oikeita laitteistoja opetuskäytössä. Vaikka digitaaliselle mallille voidaankin tehdä erilaisia törmäys- sekä tarttumapintoja, se ei kuitenkaan kykene saavuttamaan samantyyppistä realistista tuntumaa kuin se olisi todellisessa maailmassa. Ohjelmistot on kuitenkin kyettävä pitämään helppokäyttöisinä, ja se vaatii aina parempaa suorituskykyä myös itse käyttölaiteistolta.

Työstä syntyi myös jonkinlainen jatkokehitysidea, jossa voitaisiin yhdistää insinööriyössä esitetyt mekatroniset laitteet samaan virtuaaliympäristöön. Yhdessä ne muodostaisivat eräänlaisen ”pienoistehtaan”, jossa jokaisella laitesolulla olisi oma määritelty työvaiheensa. Tämän kautta voitaisiin harjoitella tuotantotekniikkaan liittyviä kunnossapidollisia asioita ja seurata virtuaalitehtaan läpimenoaikoja sekä tuotannollista suorituskykyä. Ideassa olisi tarkoitus yhdistää Metropolian konetekniikan kolmea eri suuntausta eli ko-

nesuunnittelu, koneautomaatio- ja tuotantotekniikkaa. Tähän aiheeseen pohjautuva laajempi kurssi voitaisiin suunnitella yhteiseksi konetekniikan alalle, jossa jokainen suuntautumisala pääsisi näyttämään ja syventämään osaamistaan yhteisessä hankkeessa.

Lähteet

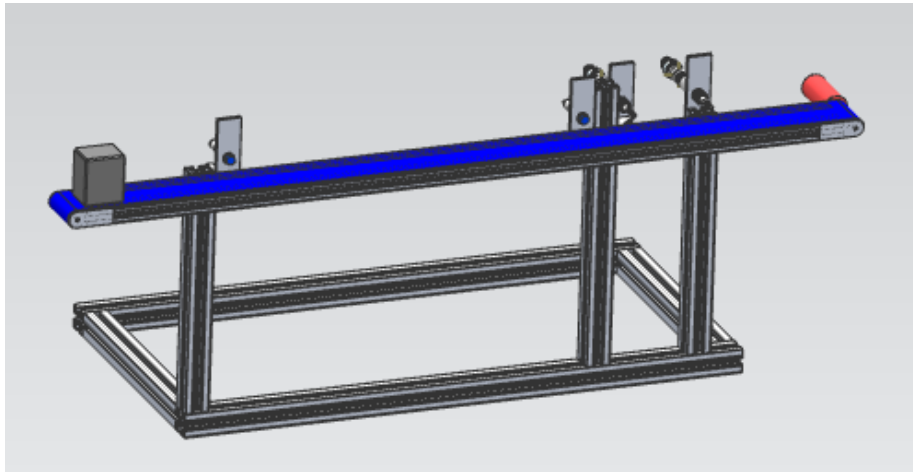
- 1 CODESYS Control. Verkkoaineisto. <https://www.codesys.com/products/codesys-runtime/control.html>. Luettu 16.05.2019.
- 2 Howling Pixel. Verkkoaineisto. <https://howlingpixel.com/i-fi/Pro/Engineer>. Luettu 19.05.2019.
- 3 Tuotteet. Verkkoaineisto. https://www.solidworks.fi/sw/docs/3DS_2017_SWK_Differences2017vs2016_Datasheet_A4_FOR-WEB.pdf. Luettu 20.05.2019.
- 4 NX for Manufacturing. Verkkoaineisto. <https://community.sw.siemens.com/s/article/mechatronics-concept-designer-mcd-player>. Luettu 26.05.2019.
- 5 Alander Jarmo T. 2018. PLC/Ohjelmoitava logiikka. Verkkoaineisto. <http://lipas.uwasa.fi/~TAU/AUTO1010/slides.php?Mode=Printer&File=9050PLC.txt>. Luettu 27.05.2019.
- 6 Beckhoff. Rakennusautomaatio. Verkkoaineisto. https://infosys.beckhoff.com/english.php?content=../content/1033/tcplccontrol/html/TcPlcCtrl_EditorCFC.htm&id=. Luettu 04.06.2019.
- 7 Metropolia. OPC-Open Process Control. Verkkoaineisto. <https://wiki.metropolia.fi/display/koneautomaatio/OPC++Open+Process+Control>. Luettu 8.06.2019.
- 8 Opiskelisinko CAD-suunnittelua? Verkkoaineisto. <https://www.aike.fi/opiskelisinko-cad-suunnittelua/>. Luettu 15.06.2019.
- 9 Tietokoneavusteisen valmistuksen (CAD/CAM) ohjelmistot. Verkkoaineisto. <https://www.hexagonmi.com/fi-FI/products/computer-aided-manufacturing-cad-cam-software>. Luettu 21.06.2019.

- 10 Suunnittelu, arkkitehtuuri ja rakentaminen. Verkkoaineisto. <https://www.lenovo.com/fi/fi/workstations/engineering-architecture-and-construction-industry/c/engineering-architecture-industry>. Luettu 28.06.2019.
- 11 Tuotteen elinkaaren hallinta (PLM). Verkkoaineisto. <https://www.e21.fi/toimialaratkaisut/tuotetiedon-hallinta/product-lifecycle-management-plm>. Luettu 01.07.2019.
- 12 Ahola. Jari M. 2011. Moniteknisen tuotteen virtuaalisuunnittelun konsepti. VTT. Verkkoaineisto. <http://videonet.fi/web/tekes/20110330/9/ahola.pdf>. Luettu 12.07.2019.
- 13 MCD-koulutusmateriaali.PDF. Metropolia Ammattikorkeakoulun Intranet.
- 14 FSG Siemens Digital Twin Engineering Excellence Award. 2018. Verkkoaineisto. <https://www.formulastudent.de/pr/news/details/article/fsg-siemens-digital-twin-engineering-excellence-award/>. Luettu 14.07.2019.
- 15 Bolton W. Programmable Logic Controller's. 2006. Verkkoaineisto. [https://www.ef.ues.rs.ba/~slubura/Procesni%20racunari/Programmable%20Logic%20Controllers%204th%20Edition%20\(W%20Bolton\).pdf](https://www.ef.ues.rs.ba/~slubura/Procesni%20racunari/Programmable%20Logic%20Controllers%204th%20Edition%20(W%20Bolton).pdf). Luettu 15.07.2019.
- 16 IGI-Global disseminator of knowledge. Verkkoaineisto. <https://www.igi-global.com/dictionary/virtual-modeling-prototyping-collaborative-product/31727>. Luettu 17.07.2019.
- 17 CODESYS. The Comprehensive Software Suite for Automation. Technology. Verkkoaineisto. <http://www.codesys.com/the-system.html>. Luettu 18.07.2019.
- 18 CODESYS. Codesys Development System. Verkkoaineisto. <https://www.codesys.com/products/codesys-engineering/development-system.html>. Luettu 21.07.2019.

- 19 Etteplan. Digitaalinen kaksonen mullistaa nyt teollisuuden tuotekehitystä. Verkkoaineisto. <https://www.etteplan.com/fi/artikkelit/digitaalinen-kaksonen-mullistaa-nyt-teollisuuden-tuotekehitysta>. Luettu 09.08.2019.
- 20 Verkkoaineisto. <https://sites.google.com/site/plcohjelmointi/tervetuloa/fbd-pikaopas/timers>. Luettu 10.08.2019.
- 21 Armendia Mikel. Twin Control. 2018. Verkkoaineisto. <https://link.springer.com/content/pdf/10.1007%2f978-3-030-02203-7.pdf>. Luettu 14.08.2019.
- 22 Pervilä Markku. Digitaaliset kaksoset ovat it:ssä nyt kuuminta uutta. 2019. Verkkoaineisto. <https://www.tivi.fi/uutiset/digitaaliset-kaksoset-ovat-itssa-nyt-kuuminta-uutta/40f1f3c3-0fe7-3825-b22b-418cc028f990>. Luettu 20.08.2019.
- 23 Scott Whitlock. The TwinCAT 3 Review. 2016. Verkkoaineisto. <http://www.contactandcoil.com/automation/industrial-automation/the-twincat-3-review/>. Luettu 21.08.2019.
- 24 Dannana Sundar. CATIA V5 Modules list. 2017. Verkkoaineisto. <https://extrudesign.com/catia-v5-modules-list/>. Luettu 24.08.2019.
- 25 Hao Lina. 2011. researchgate. Verkkoluettu. https://www.researchgate.net/figure/Design-Process-of-Virtual-Prototype_fig1_269344639. Luettu 27.08.2019
- 26 Formula student. 2018.FSG Siemens Digital Twin Engineering Excellence Award. Verkkoluettu. <https://www.formulastudent.de/pr/news/details/article/fsg-siemens-digital-twin-engineering-excellence-award/>. Luettu 30.08.2019.
- 27 CATIA Reviews & Product Details. Verkkoluettu. <https://www.g2.com/products/catia/reviews>. Luettu 03.09.2019

- 28 YouTube. Solidworks tutorial | How to Set Standard Views in Command Manager. Verkkoluettu. <https://www.youtube.com/watch?v=ijTERzFkhXY>. Luettu 5.09.2019.
- 29 SWIN-Ohjelmointi peruskoulutus. Verkkoluettu. http://www.tekniikka.oamk.fi/~timohei/TL602Z/aineisto/syswin_peruskoulutusmateriaali.PDF. Luettu 08.09.2019.
- 30 Lee Chi G. & Park Sang C. 2014. Survey on the virtual commissioning of manufacturing systems. Verkkoluettu. <https://www.sciencedirect.com/science/article/pii/S2288430014500292>. Luettu 11.09.2019.

PLC/ PC-harjoitus, kuljetinlajittelija



TYÖOHJE

Tehtävänä on suunnitella logiikkaohjelma automaatiolaboratorion virtuaaliselle kuljetinhihnan mallille. Tarkoituksena on harjoitella simulaatiomallin virtuaalista käyttöönottoa. Harjoitus koostuu kahdesta osasta, jotka on tarkoitus suorittaa oppituntien aikana. Ensimmäisessä osuudessa tehdään yksinkertainen logiikkasovellus, toisessa vaativampi logiikkasovellus samalle DigitalTwin mallille. Työt tehdään käyttäen NX MCD- ja CODESYS/ TwinCat-ohjelmistoa. Tee logiikkaohjelma tehtäväselostuksen mukaisesti.

VIRTUAALIMALLIN OSAT

- Kuljetinhihna
- Työsylinteri 2 kpl
- Anturi 2 kpl
- Sähkömoottori
- Kappale

TULOT JA LÄHDÖT

VAR_GLOBAL

```
// Sisäinen muistipaikka
```

```
bStart:BOOL;
```

```
// Tulot
```

```
bAnturi1:BOOL;
```

```
bAnturi2:BOOL;
```

```
// Lähdöt
```

```
bHihna:BOOL;
```

```
bSylinteri1:BOOL;
```

```
bSylinteri2:BOOL;
```

END_VAR

TEHTÄVÄ

OSIO 1

Simulaatiomallin yksinkertaisen toiminnan suorittaminen tehdään CODESYS-ohjelmalla.

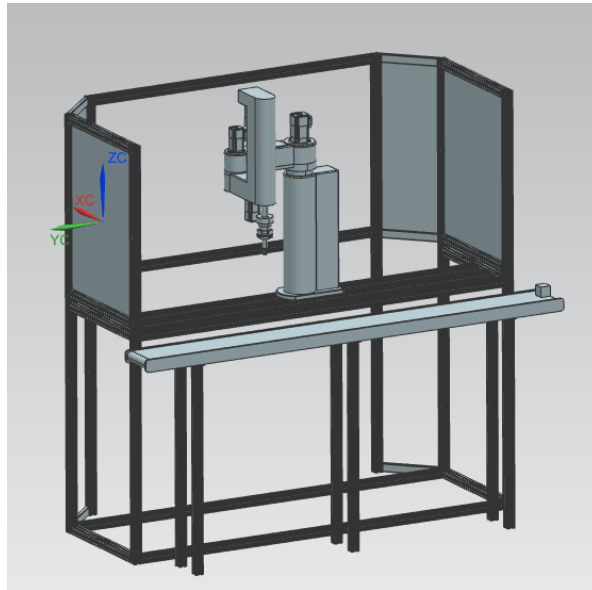
1. *START- kuljetinhihna lähtee liikkeelle.*
2. *Kuljettimella oleva kappale saavuttaa Anturi 2 kohdan.*
3. *Anturi tunnistaa kappaleen ja liike pysähtyy.*

OSIO 2

Vaativampi simulaatioharjoitus, jossa otetaan anturin lisäksi työsylinteri käyttöön.

1. *START- kuljetinhihna lähtee liikkeelle.*
2. *Kuljettimella oleva kappale saavuttaa Anturi 2 kohdan.*
3. *Anturi tunnistaa kappaleen laskuri käynnistyy.*
4. *Sylinteri 2 pudottaa kappaleen hihnalta.*

SERVO/ PC-harjoitus, SCARA-robotti



TYÖOHJE

Tehtävänä on suunnitella logiikkaohjelma automaatiolaboratorion virtuaaliselle SCARA-robotin mallille. Tarkoituksena on harjoitella simulaatiomallin virtuaalista käyttöönottoa. Harjoitus koostuu kahdesta osasta, jotka on tarkoitus suorittaa oppituntien aikana. Ensimmäisessä osuudessa tehdään yksinkertainen logiikkasovellus, toisessa vaativampi logiikkasovellus samalle DigitalTwin-mallille. Työt tehdään käyttäen NX MCD- ja CO-DESYS/ TwinCat-ohjelmistoa. Tee logiikkaohjelma tehtäväselostuksen mukaisesti.

VIRTUAALIMALLIN OSAT

- Kuljetinhihna
- Suoja/ tukikehikko
- Kappale
- Bosch SR60 tyyppinen SCARA-yksikkö (kolme pyörivää niveltä ja yksi pystysuuntainen liike)

TULOT JA LÄHDÖT**VAR_GLOBAL**

```
// Sisäinen muistipaikka
```

```
Start: BOOL;
```

```
// Tulot
```

```
bAnturi: BOOL;
```

```
// Lähdöt
```

```
bHihna: BOOL;
```

```
bInstrumentti: BOOL;
```

```
bKyynervarsi: BOOL;
```

```
bLiukuhihna: BOOL;
```

```
bOlkavarsi: BOOL;
```

END_VAR**TEHTÄVÄ****OSIO 1**

Simulaatiomallin yksinkertaisen toiminnan suorittaminen tehdään TwinCat-ohjelmalla.

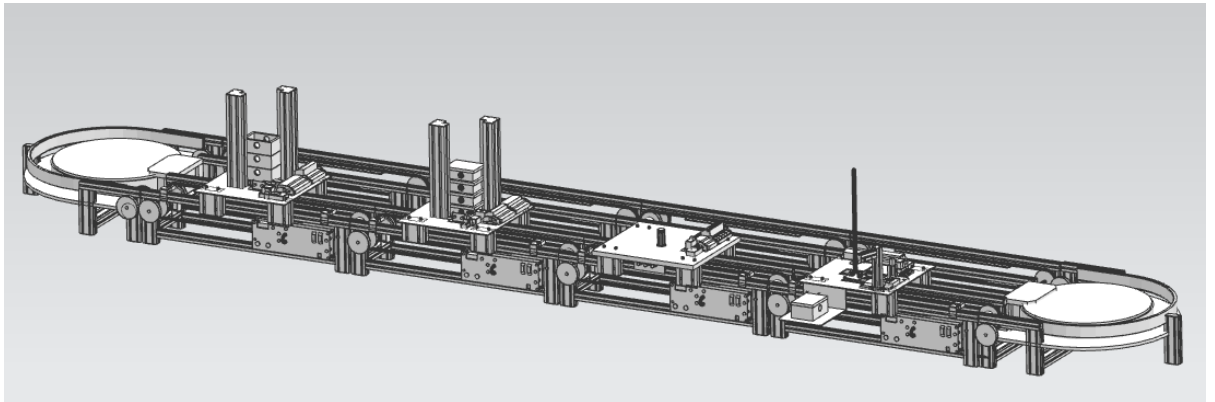
1. *START- servomoottorit käynnistyvät.*
2. *Ohjataan SCARA (A) koordinaatista (B) koordinaattiin.*
3. *Konfiguroi robotin liikkeen ohjaus erillisen ohjeen mukaisesti.*

OSIO 2

Vaativampi simulaatioharjoitus, jossa kappaleen lajittelu tapahtuu erilliselle tasolle.

1. *START- servomoottorit käynnistyvät.*
2. *Ohjataan SCARA (A) koordinaatista (B) koordinaattiin.*
3. *Konfiguroi robotin liikkeen ohjaus erillisen ohjeen mukaisesti.*
4. *Syötä kappale liukuhihnalle ja poimi se.*

PLC-harjoitus, SmartFactory-miniprojekti



TYÖOHJE

Tehtävänä on suunnitella logiikkaohjelma automaatiolaboratorion virtuaaliselle SmartFactory mallille. Tarkoituksena on harjoitella simulaatiomallin virtuaalista käyttöönottoa. Harjoitus on miniprojekti, jotka on tarkoitus suorittaa oppituntien aikana. Ensimmäisessä osuudessa tehdään yksinkertainen logiikkasovellus, toisessa vaativampi logiikkasovellus samalle DigitalTwin-mallille. Työt tehdään käyttäen NX MCD- ja CODESYS/ TwinCat -ohjelmistoa. Tee logiikkaohjelma tehtäväselostuksen mukaisesti.

VIRTUAALIMALLIN OSAT

- Sähkömoottori
- Työsylinteri
- Kappale (3 osaa)
- Kelkka
- Kuljetinhihna
- Kokoonpanomakasiini 2 kpl

TULOT JA LÄHDÖT

VAR_GLOBAL

```
// Sisäinen muistipaikka
```

```
Start: BOOL;
```

```
// Tulot
```

```
Paikka1: BOOL;
```

```
Paikka2: BOOL;
```

```
Paikka3: BOOL;
```

```
Paikka4: BOOL;
```

```
// Lähdöt
```

```
Mantalliike: DOUBLE;
```

```
Mantalliike: DOUBLE;
```

```
Metallitappi: DOUBLE;
```

```
Pohjaliike2: DOUBLE;
```

```
Stopperi1: DOUBLE;
```

```
Stopperi2: DOUBLE;
```

```
Stopperi3: DOUBLE;
```

```
Stopperi4: DOUBLE;
```

```
Stopperi5: DOUBLE;
```

```
Stopperi6: DOUBLE;
```

```
Stopperi7: DOUBLE;
```

```
Stopperi8: DOUBLE;
```

```
Tapinliike: DOUBLE;
```

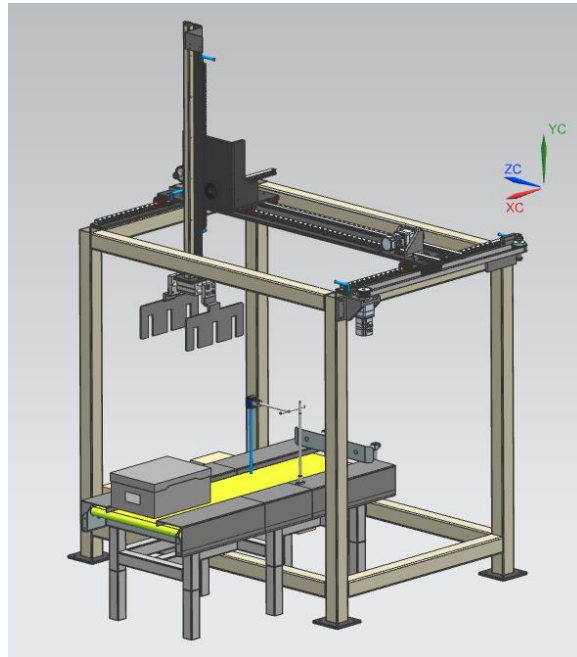
END_VAR

TEHTÄVÄ

Simulaatiomallin yksinkertaisen toiminnan suorittaminen CODESYS-ohjelmalla. Simulaatioharjoitus, jossa otetaan anturit ja kokoonpanolinjan työsylinterit käyttöön. Kappale kootaan pienoislinjastolla.

1. *Tilaus-painike, kuljetinkelkka liikkuu pohjamakasiinille.*
2. *Tuote jatkaa matkaa kansimakasiinille ja lopuksi pulttiasemalle.*
3. *Lopuksi valmis kappale liikkuu poiminta-asemalle.*

SERVO/ PC-harjoitus, XYZ-siltanosturi



TYÖOHJE

Tehtävänä on suunnitella logiikkaohjelma automaatiolaboratorion virtuaaliselle XYZ-siltanosturin mallille. Tarkoituksena on harjoitella simulaatiomallin virtuaalista käyttöä. Harjoitus koostuu kahdesta osasta, jotka on tarkoitus suorittaa oppituntien aikana. Ensimmäisessä osuudessa tehdään yksinkertainen logiikkasovellus, toisessa vaativampi logiikka sovellus samalle DigitalTwin-mallille. Työt tehdään käyttäen NX MCD- ja CODESYS/ TwinCat-ohjelmistoa. Tee logiikkaohjelma tehtäväselostuksen mukaisesti.

VIRTUAALIMALLIN OSAT

- Sähkömoottori
- Tukikehikosta
- Tarttuja
- Liukuhihna
- Kappale
- Jalusta

TULOT JA LÄHDÖT

VAR_GLOBAL

```
// Sisäinen muistipaikka
```

```
Start: BOOL;
```

```
Stop: BOOL;
```

```
// Tulot
```

```
Z_paikka2: DOUBLE;
```

```
Y_paikka2: DOUBLE;
```

```
X_paikka: DOUBLE;
```

```
Linjasto_anturi: BOOL;
```

```
// Lähdöt
```

```
1X+: BOOL;
```

```
1X-: BOOL;
```

```
1Y+: BOOL;
```

```
1Y-: BOOL;
```

END_VAR

TEHTÄVÄ

OSIO 1

Simulaatiomallin yksinkertaisen toiminnan suorittaminen CODESYS-ohjelmalla, esimerkiksi; nosturin tarttujaa liikutetaan kehikossa X-Y-suunnassa, Z-suunta on vinssin ylös- alas liike. X ja Y paikkaohje.

1. *START -> Nosturin käynnistys.*
2. *Mitataan matka X ja Y koordinaatista toiseen X ja Y koordinaattiin.*

OSIO 2

Vaativammassa simulaatioharjoituksessa tarttujalla poimitaan kappale liukuhihnalta, siten että joudutaan käyttämään kaikkia X-, Y-, ja Z-koordinaatteja.

1. *START-> Nosturin käynnistys.*
2. *Syötä kappale liukuhihnalle.*
3. *Kappale siirtyy poiminta paikkaan, josta se tarttujalla siirretään erilliselle alustalle.*

PLC/ PC-signaalien luominen ja yhteyden luominen PLC-editoriin

SISÄLLYSLUETTELO

1. NX.MCD

1.1 Ohjaussignaalien luominen

1.2. Valmiit ohjaussignaalit

2. CODESYS

2.1 NX & CODESYS-välinen yhteys

2.2. External signal configuration

2.3 Signal Mapping

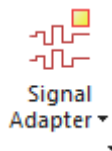
1. NX MCD

Aukaise NX MCD ohjelmisto ja etsi kuljetinhihnan malli

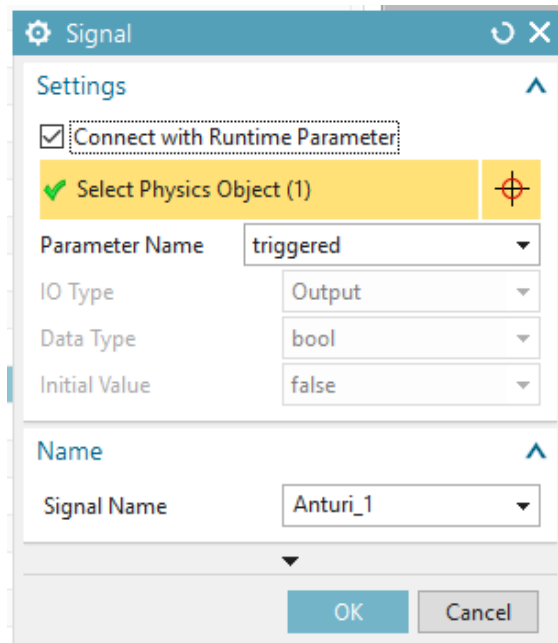
- Open -> T(T:) -> oppilas -> Digi_Lab -> Kuljetinhihna_sldasm.prt

1.1 Ohjaus-signaalin luominen

Anturi elementeille voidaan luoda signaalin suoraan, malliesimerkissä on olemassa kaksi anturia, joille luodaan signaalit samalla tavalla. Jotta Signal/ Signal Adapter toimisi, täytyy elementti olla määritetty joko Control tai Sensor-tyyppiseksi. Signal Adapteria käytetään, kun halutaan vaikkapa muuttaa ohjaus -signaalia. Signaalit löytyvät ylävalikosta kuvakkeen alta.



- Valitaan signals ylävalikosta.
- Valitaan anturielementti, johon signaali luodaan.
- Rastita **Connect with Runtime Parameter** -kohta.
- Parametriksi **Triggered**.
- Nimetään signaali **Anturi_1** ja OK.



Kuva 1. Tavallisten signaalien luominen

Luodaan Signal Adapter liukuhihnan ja moottorin välille, jolloin sitä voidaan ohjata ON /OFF-periaatteella. Myös työsylinterille, jolla on rajat esim. kiinni/ auki, voidaan luoda samanlainen signaalityyppi.

(1)

- Valitaan kuljettimen hihna aktiiviseksi
- Painetaan Add
- Rastitetaan Parameter_1

(2)

- Nimitään signaali
- Lähdeksi valitaan Input
- Painetaan Add

(3)

- Paina Parameter_1 ja alakumassa oleva Insert Conditionalm joka muuttuu aktiiviseksi, sieltä löytyy määrittely Conditional Builder.

(4)

- Nimeä Signal Adapteri

Kuva 2. Signal Adapter-valintaikkuna

Conditional Builder

If

Then

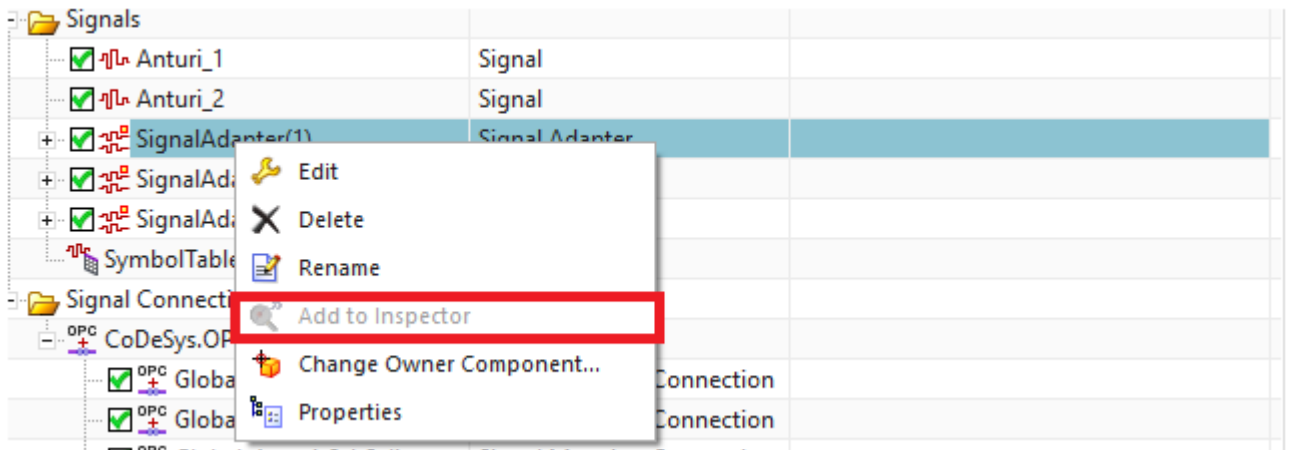
Else

OK Back Cancel

Kuva 3. Conditional Builder-toiminto

1.2 Valmiit ohjaussignaalit

Kun kaikki tarvittavat ohjaussignaalit on saatu luotua, siirretään kaikki yksitel-
len siten, että klikkaa hiiren kakkospainikkeella valintaikkunaa ja valitse ”**Add
Inspector**”. Tämän jälkeen mallia voidaan ohjata NX-ympäristössä.



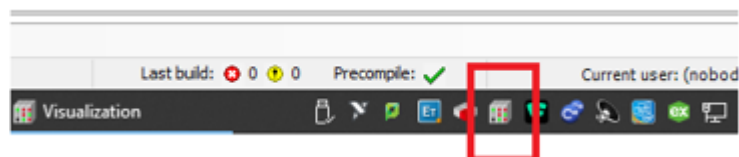
Kuva 4. Signaalien lisäys Inspectoriin.

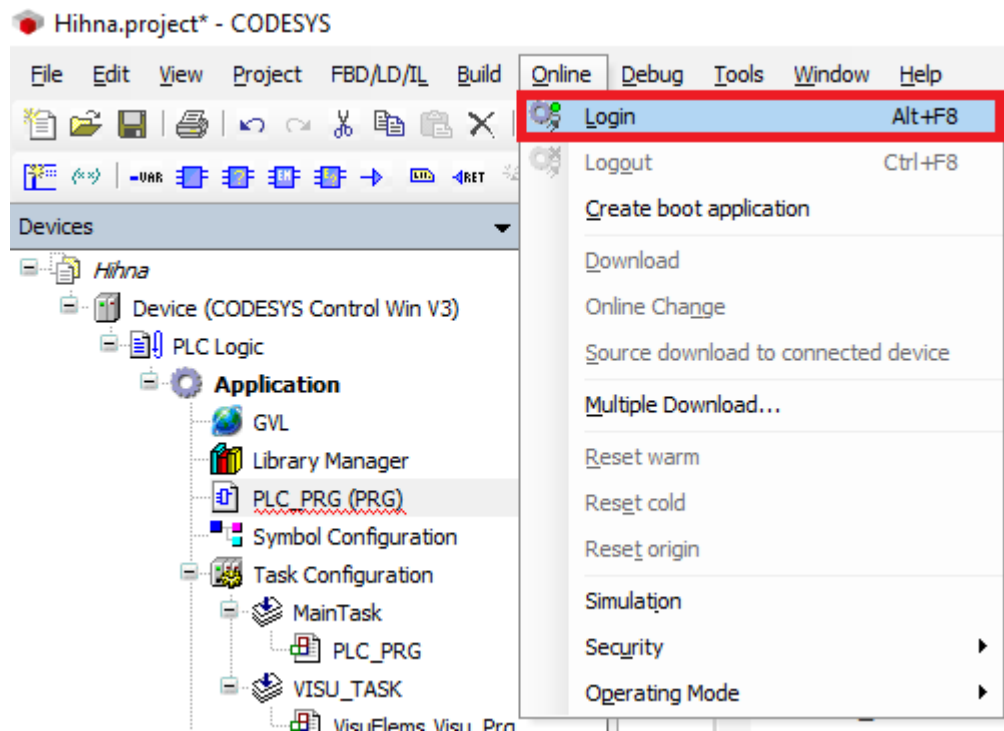
2. CODESSYS

2.1 NX & CODESYS välinen yhteys

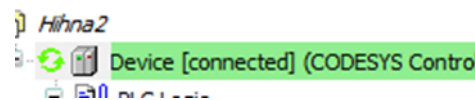
Aukaise CODESYS PLC -editori ja etsi Hihna2.project

- Open -> T(T:) -> oppilas -> Digi_Lab -> Codesys -> **Hihna2.project**
- Ota yhteys PLC:n alariviltä, klikkaa hiiren kakkospainiketta ja valitse **Start PLC**.



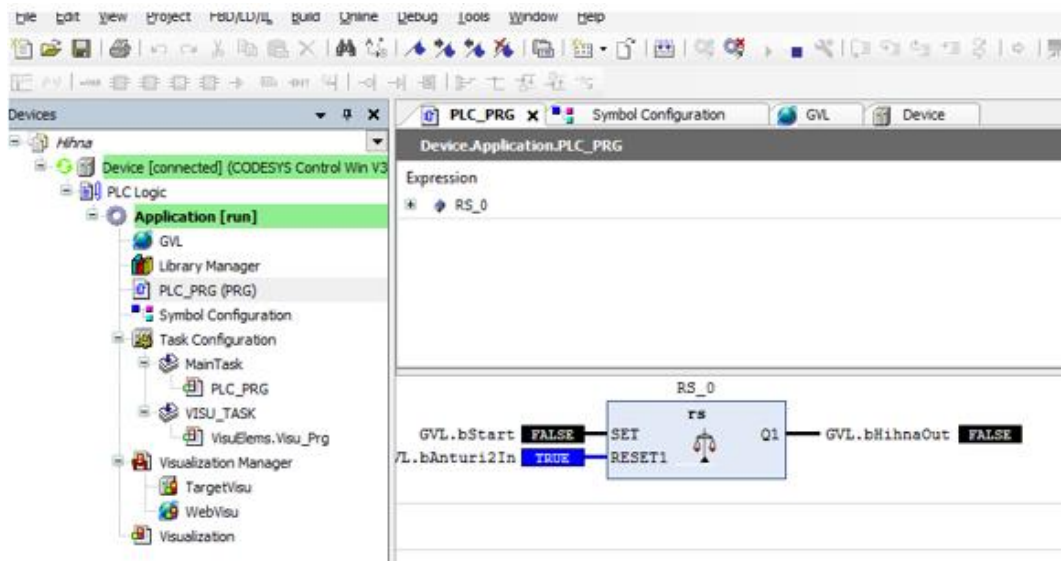


Kuva 5. Yläriviltä voidaan asettaa yhteys OPC:n, klikkaa -> Online.



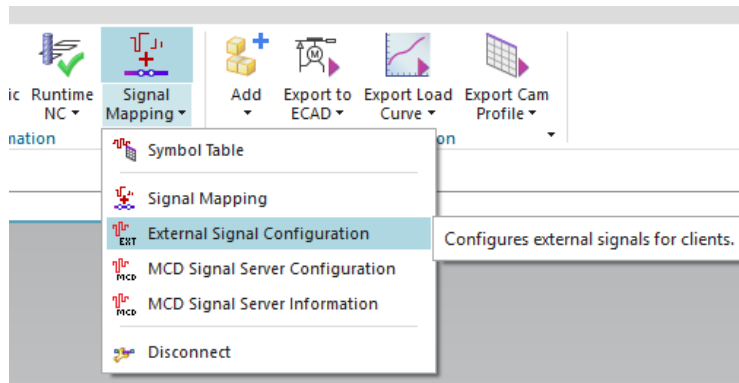
Kuva 6. yhteys muodostettu

Device muuttuu vihreäksi, kun yhteys on onnistunut. Jos **Mode**-toiminto lähtee käyntiin (ruudulle ilmestyy ohjainpainikkeet) Windows-kuvapainikkeella pääsee pois tilasta.



Kuva 6. Esimerkkiohjelman tulot sekä lähdöt.

2.2 External signal Configuration



(1)
- Vainaan CobDeSys-OPC.DA
- Server ProgID on sama

(2)
- Rasittaa tulot sekä lähdöt
-OK

2.3 Signal Mapping

- MCD signals -> Testi
- Luodaan Linkki -> Map Signal
- External signal -> Moottori

The screenshot shows the 'Signal Mapping' window with the following components:

- External Signal Type:** Set to 'OPC DA Servers'.
- External Signal (1):** A table listing MCD signals:

Name	Adapter Name	ID Type	Data Type	Mapping Count
Aiemu_1	Global	Output	bool	1
Aiemu_2	Global	Output	bool	2
Moottori	SignalAdapter(1)	Input	bool	1
System_1	SignalAdapter(2)	Input	bool	0
System_2	SignalAdapter(3)	Input	bool	1
- External Signal (12):** A table listing external signals:

Name	ID Type	Data Type	Mapping Count
CommonOpc	Output	bool	0
bahuninOpc	Input/Output	bool	1
bahuninIn	Input/Output	bool	1
bahuninOut	Input/Output	bool	1
bahunin	Input/Output	bool	1
bahuninOpc	Input/Output	bool	1
Map Signal	Input/Output	bool	1
bjylienOpc	Input/Output	bool	2
bjylienOut	Input/Output	bool	2
- Map Signal:** A table showing the mapping between the two external signal lists:

Map Signal	Input/Output	bool	Mapping Count
bjylienOpc	Input/Output	bool	1
bjylienOut	Input/Output	bool	2
- Mapped Signals:** A table showing the connection names, MCD signal names, directions, and external signal names:

Connection Name	MCD Signal Name	Direction	External Signal Name	Owner Component	Message
OPC DA,ColSys,OPC DA	Moottori	←	bahuninOpc		
SignalAdapter(1),Moottori,bahunin	Aiemu_1	→	bahuninIn		
Global,Aiemu_1,bahuninIn	Aiemu_2	→	bahuninIn		
SignalAdapter(1),Moottori,bahunin	System_1	→	bjylienOpc		
Global,Aiemu_2,bahuninIn	System_2	→	bjylienOpc		
SignalAdapter(3),System_2,bjylienOpc	Aiemu_2	→	bjylienOpc		
Global,Aiemu_2,bjylienOpc					

Annotations in the image:

- (11) - tarkista että signaali on tyhji
- tarkista että signaali on tyhji
- tarkista että signaali on tyhji
- (12) - Valitaan kahden toisen ja lähtee, jotta
- Valitaan kahden toisen ja lähtee, jotta
- Valitaan kahden toisen ja lähtee, jotta

Kun yhteys on valmis, mennään NX-ohjelmaan ja painetaan -> Play, kun taas CODESYS ohjelmassa avataan linkki PLC_PRG -> False/True (muutetaan painamalla **Ctrl+F7**)
Kuljetinhihna alkaa liikkua NX-ikkunassa.

PLC/ PC-malliharjoitus, kuljetinlajittelija

NX MCD-ohjelmisto

Yksinkertaistettu kuva Physics Navigator-osiosta.

Fyysiset osat

- 1Varsi
- 2varsi
- kappale
- Kuljetin

Sensorit ja toimilaitteet

- Anturi 1
- Anturi 2
- Kuljetin
- Sylinteri 1
- Sylinteri 2

Signaalit

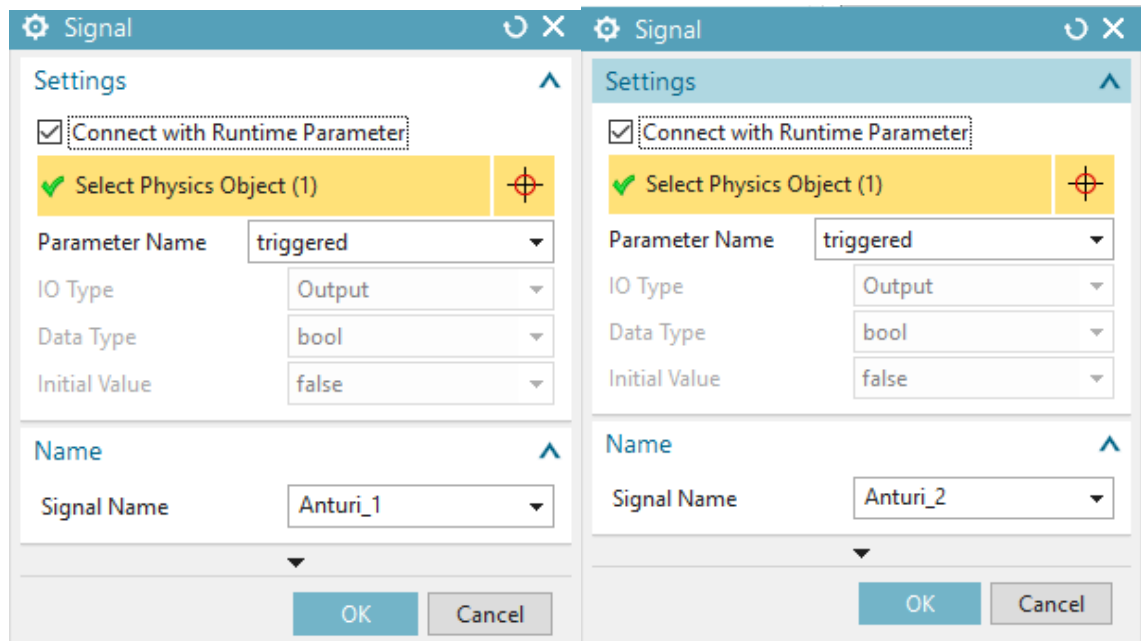
- Anturi 1
- Anturi 2
- SignaAdapter (1)
- SignaAdapter (2)
- SignaAdapter (3)

Tarvittavat signaaliyhteydet (Control tai Sensor elementti)

- Kuljetin -> Moottori (Signal Adapter)
- Anturi 1-> Lajittelija 1 (signal)
- Anturi 2-> Lajittelija 2 (signal)
- Kappale -> Anturi 1 & 2
- Lajittelija 1 (Signal Adapter 1/2 = 0tms/100tms)
- Lajittelija 2 (Signal Adapter 1/2 = 0tms/100tms)

Anturi elementit

Signal-toiminto



Anturien signaalien määrittäminen tapahtuu niin, että valitaan laitteesta anturielementit aktiiviseksi, joihin signaalit luodaan sekä nimetään ne tunnistamisen helpottamiseksi.

Signal Adapter-toiminto

Signal Adapter

Parameters

Select Physics Object (0)

Parameter Name

Add Parameter

A	Alias	Object	Object T...	Parameter	Value	Unit	Data Type	Read/Write
<input checked="" type="checkbox"/>	Parameter_1	Kuljetin	Transport ...	parallel speed	40,000000	mm/s	double	W

Signals

A	Name	Data Type	Input/Out...	Initial Value	Measure	Unit
	Moottori	bool	Input	false		

Formulas

Assign to	Formula	Comment
Parameter_1	If (Moottori = false) Then (0) Else (10)	

Formula

Name

SignalAdapter(1)

OK Cancel

Signal Adapter
✕

Parameters

✖ Select Physics Object (0)

Parameter Name

Add Parameter

A	Alias	Object	Object T...	Parameter	Value	Unit	Data Type	Read/Write
<input checked="" type="checkbox"/>	Parameter_1	Sylinteri 1	Position ...	speed	0.000000	mm/s	double	W
<input checked="" type="checkbox"/>	Parameter_3	Sylinteri 1	Position ...	position	0.000000	mm	double	W

Signals

A	Name	Data Type	Input/Out...	Initial Value	Measure	Unit
	Sylinteri_1	bool	Input	false		

Formulas

Assign to	Formula	Comment
Parameter_1	If (Signal_0 = false) Then (80) Else If (Signal_0 = true) Then (80) Else (...	
Parameter_3	If (Signal_0 = false) Then (0) Else If (Signal_0 = true) Then (50) Else (0)	

Formula

Name

SignalAdapter(2)

OK Cancel

Signal Adapter
✕

Parameters

✖ Select Physics Object (0)

Parameter Name

Add Parameter

A	Alias	Object	Object T ...	Parameter	Value	Unit	Data Type	Read/Write
<input checked="" type="checkbox"/>	Parameter_1	Syinteri 2	Position ...	speed	0.000000	mm/s	double	W
<input checked="" type="checkbox"/>	Parameter_2	Syinteri 2	Position ...	position	0.000000	mm	double	W

Signals

A	Name	Data Type	Input/Out...	Initial Value	Measure	Unit
	Syinteri_2	bool	Input	false		

Formulas

Assign to	Formula	Comment
Parameter_1	If (Signal_1 = true) Then (80) Else (80)	
Parameter_2	If (Signal_1 = true) Then (50) Else (0)	

Formula

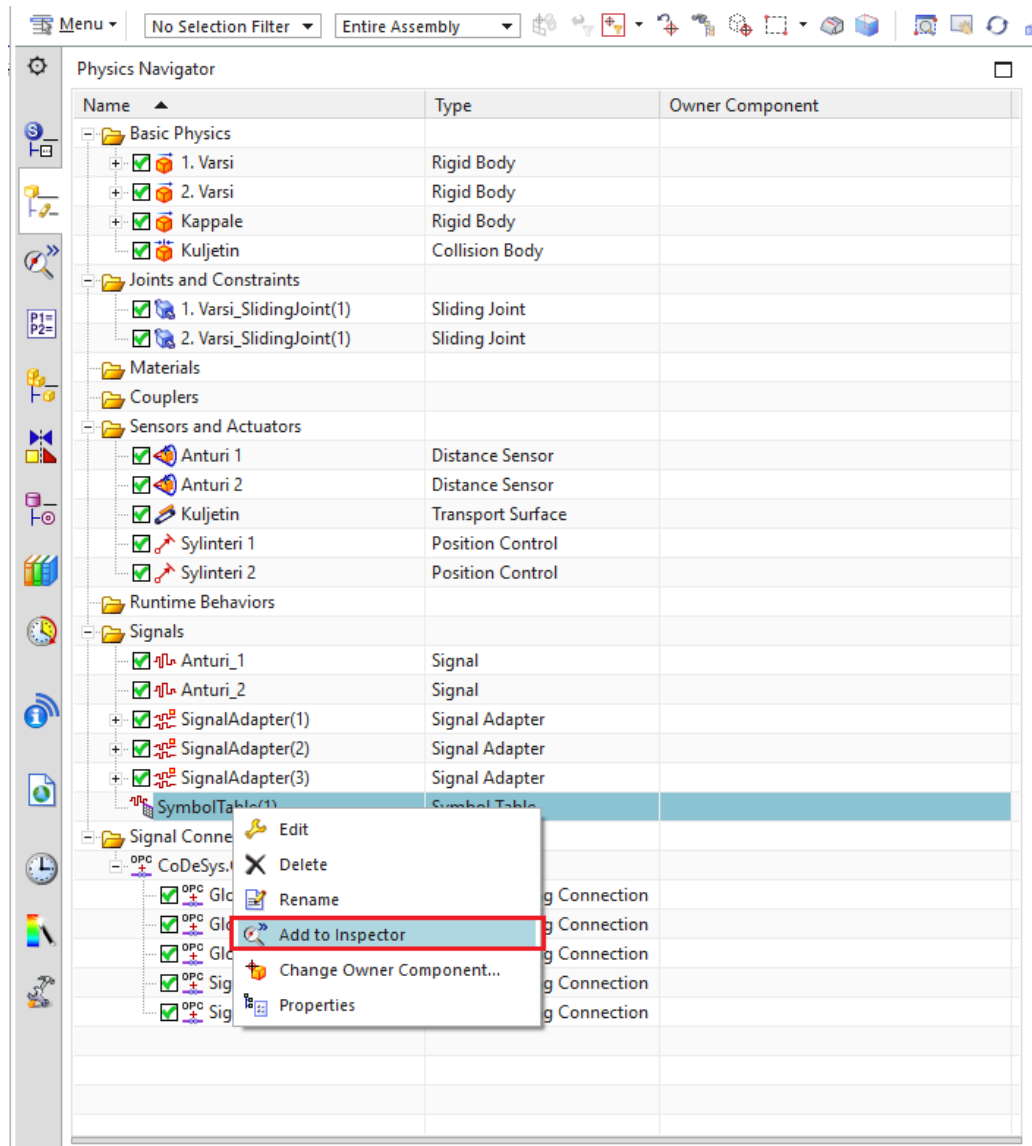
Name

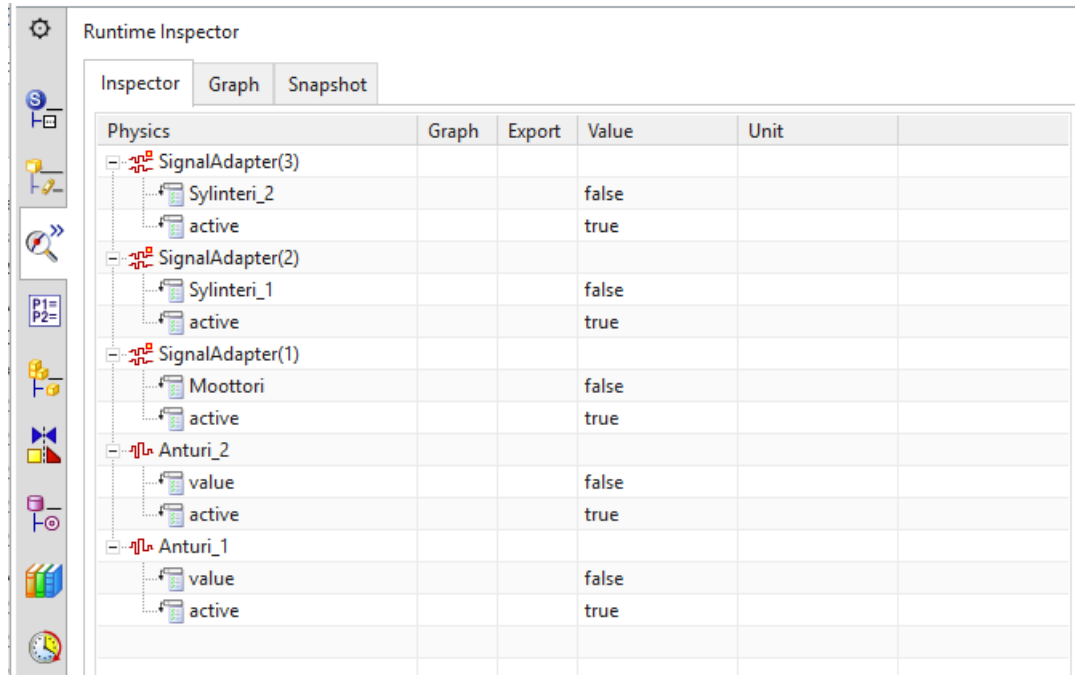
SignalAdapter(3)

OK Cancel

Signaalien lisääminen suoritettaviin toimintoihin

Runtime Inspector





Runtime Inspector

Inspector Graph Snapshot

Physics	Graph	Export	Value	Unit	
SignalAdapter(3)					
Sylinteri_2			false		
active			true		
SignalAdapter(2)					
Sylinteri_1			false		
active			true		
SignalAdapter(1)					
Moottori			false		
active			true		
Anturi_2					
value			false		
active			true		
Anturi_1					
value			false		
active			true		

Linkin rakentaminen

Signal Mapping-toiminto

Signal Mapping

External Signal Type
Type
OPC DA Servers

OPC DA
CodeSys:OPC.DA

Signals

MCD Signals (5)

Name	Adapter Name	IO Type	Data Type	Mapping Count
Anturi_1	Global	Output	bool	1
Anturi_2	Global	Output	bool	2
Moottori	SignalAdapter(1)	Input	bool	1
Sylinteri_1	SignalAdapter(2)	Input	bool	0
Sylinteri_2	SignalAdapter(3)	Input	bool	1

External Signals (12)

Name	IO Type	Data Type	Mapping Count
_CommState	Output	int	0
_CommState	Output	int	0
_CommStateOK	Output	bool	0
_CommStateOK	Output	bool	0
bAnturi1In	Input/Output	bool	1
bAnturi1In	Input/Output	bool	1
bAnturi2In	Input/Output	bool	1
bAnturi2In	Input/Output	bool	1
bHihnaOut	Input/Output	bool	1

Do Auto Mapping

Mapped Signals

Connection Name	MCD Signal Name	Direction	External Signal Name	Owner Component	Message
OPC DA.CodeSys:OPC.DA					
SignalAdapter(1)_Moottori_bHihnaOut	Moottori	←	bHihnaOut		
Global_Anturi_1_bAnturi1In	Anturi_1	→	bAnturi1In		
Global_Anturi_2_bAnturi2In	Anturi_2	→	bAnturi2In		
SignalAdapter(3)_Sylinteri_2_bsylinteri2Out	Sylinteri_2	←	bsylinteri2Out		
Global_Anturi_2_bsylinteri2Out	Anturi_2	→	bsylinteri2Out		

Check for N->1 Mapping

OK Cancel

Signal Mapping
OPC DA
CodeSys:OPC.DA

External Signal Type

OPC DA Servers

Signals

MCD Signals (5)

Name	Adapter Name	IO Type	Data Type	Mapping Count
Anturi_1	Global	Output	bool	1
Anturi_2	Global	Output	bool	2
Moottori	SignalAdapter(1)	Input	bool	1
Sylinteri_1	SignalAdapter(2)	Input	bool	0
Sylinteri_2	SignalAdapter(3)	Input	bool	1

External Signals (12)

Name	IO Type	Data Type	Mapping Count
__CommState	Output	int	0
__CommStateOK	Output	bool	0
__CommStateOK	Output	bool	0
bAnturi1In	Input/Output	bool	1
bAnturi1In	Input/Output	bool	1
bAnturi2In	Input/Output	bool	1
bAnturi2In	Input/Output	bool	1
bHihnaOut	Input/Output	bool	1

Do Auto Mapping

Mapped Signals

Connection Name	MCD Signal Name	Direction	External Signal Name	Owner Component	Message
OPC DA:CodeSys:OPC.DA	Moottori	←	bHihnaOut		
	Anturi_1	→	bAnturi1In		
	Anturi_2	→	bAnturi2In		
	Sylinteri_2	←	bsylinteri2Out		
	Anturi_2	→	bsylinteri2Out		

Check for N->1 Mapping

OK
Cancel

Signal Mapping

External Signal Type: OPC DA Servers

Type: OPC DA

CodeSys:OPC.DA

Signals

MCD Signals (5)

Find: Match Case Match Whole Word

Name	Adapter Name	IO Type	Data Type	Mapping Count
Anturi_1	Global	Output	bool	1
Anturi_2	Global	Output	bool	2
Moottori	SignalAdapter(1)	Input	bool	1
Sylinteri_1	SignalAdapter(2)	Input	bool	0
Sylinteri_2	SignalAdapter(3)	Input	bool	1

External Signals (12)

Find: Match Case Match Whole Word

Name	IO Type	Data Type	Mapping Count
__CommStateOK	Output	bool	0
bAnturi1In	Input/Output	bool	1
bAnturi1In	Input/Output	bool	1
bAnturi2In	Input/Output	bool	1
bAnturi2In	Input/Output	bool	1
bHilmaOut	Input/Output	bool	1
bHilmaOut	Input/Output	bool	1
bSylinteri2Out	Input/Output	bool	2
bSylinteri2Out	Input/Output	bool	2

Do Auto Mapping

Mapped Signals

Connection Name	MCD Signal Name	Direction	External Signal Name	Owner Component	Message
OPC DA.CoDesys.OPC.DA	Moottori	←	bHilmaOut		
SignalAdapter(1)_Moottori_bHilma...	Anturi_1	→	bAnturi1In		
Global_Anturi_1_bAnturi1In	Anturi_2	→	bAnturi2In		
Global_Anturi_2_bAnturi2In	Sylinteri_2	→	bSylinteri2Out		
SignalAdapter(3)_Sylinteri_2_bSyl...	Anturi_2	→	bSylinteri2Out		
Global_Anturi_2_bSylinteri2Out					

Check for N->1 Mapping

OK Cancel

Signal Mapping

External Signal Type: OPC DA
 Type: OPC DA Servers
 CodeSys:OPC.DA

Signals

MCD Signals (5)

Name	Adapter Name	IO Type	Data Type	Mapping Count
Anturi_1	Global	Output	bool	1
Anturi_2	Global	Output	bool	2
Moottori	SignalAdapter(1)	Input	bool	1
Sylinteri_1	SignalAdapter(2)	Input	bool	0
Sylinteri_2	SignalAdapter(3)	Input	bool	1

External Signals (12)

Name	IO Type	Data Type	Mapping Count
_CommState	Output	int	0
_CommStateOK	Output	bool	0
_CommStateOK	Output	bool	0
bAnturi1In	Input/Output	bool	1
bAnturi1In	Input/Output	bool	1
bAnturi2In	Input/Output	bool	1
bAnturi2In	Input/Output	bool	1
bHilmaOut	Input/Output	bool	1
bHilmaOut	Input/Output	bool	1

Do Auto Mapping

Mapped Signals

Connection Name	MCD Signal Name	Direction	External Signal Name	Owner Component	Message
OPC DA,CodeSys:OPC.DA	Moottori	→	bHilmaOut		
	Global_Anturi_1_bAnturi1In	→	bAnturi1In		
	Global_Anturi_2_bAnturi2In	→	bAnturi2In		
	SignalAdapter(3)_Sylinteri_2_bsylinteri_2Out	→	bsylinteri2Out		
	Global_Anturi_2_bsylinteri2Out	→	bsylinteri2Out		

Check for N-> 1 Mapping

OK Cancel

Signal Mapping

External Signal Type: OPC DA
 Type: OPC DA Servers
 CodeSys:OPC.DA

Signals

MCD Signals (5)

Name	Adapter Name	IO Type	Data Type	Mapping Count
Anturi_1	Global	Output	bool	1
Anturi_2	Global	Output	bool	2
Moottori	SignalAdapter(1)	Input	bool	1
Sylinteri_1	SignalAdapter(2)	Input	bool	0
Sylinteri_2	SignalAdapter(3)	Input	bool	1

External Signals (12)

Name	IO Type	Data Type	Mapping Count
__CommStateOK	Output	bool	0
bAnturi1In	Input/Output	bool	1
bAnturi1In	Input/Output	bool	1
bAnturi2In	Input/Output	bool	1
bAnturi2In	Input/Output	bool	1
bAnturi2In	Input/Output	bool	1
bHihnaOut	Input/Output	bool	1
bHihnaOut	Input/Output	bool	1
bsylinteri2Out	Input/Output	bool	2
bsylinteri2Out	Input/Output	bool	2

Do Auto Mapping

Mapped Signals

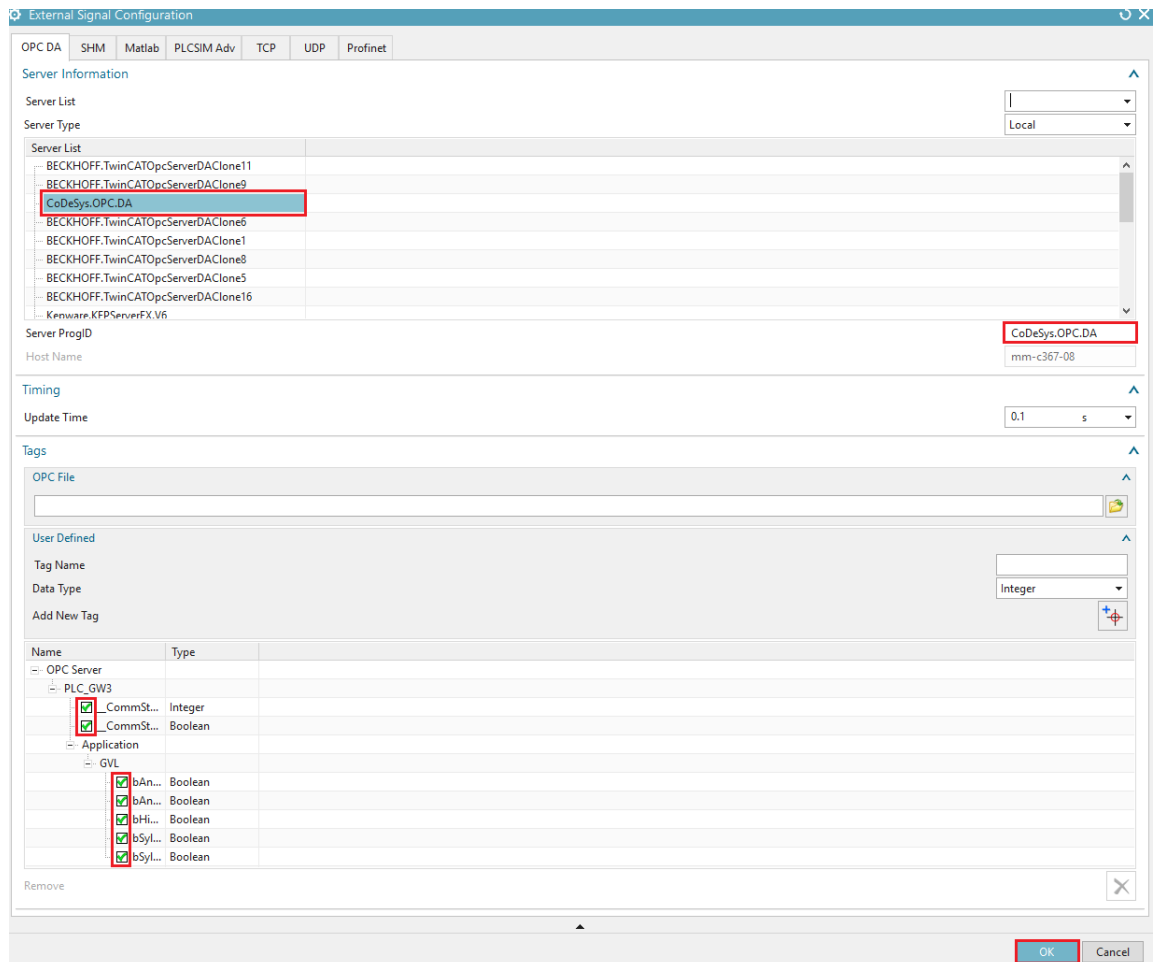
Connection Name	MCD Signal Name	Direction	External Signal Name	Owner Component	Message
OPC DA,CodeSys:OPC.DA					
SignalAdapter(1)_Moottori_bHihn...	Moottori	←	bHihnaOut		
Global_Anturi_1_bAnturi1In	Anturi_1	→	bAnturi1In		
Global_Anturi_2_bAnturi2In	Anturi_2	→	bAnturi2In		
SignalAdapter(3)_Sylinteri_2_bSyl...	Sylinteri_2	←	bsylinteri2Out		
Global_Anturi_2_bSylinteri2Out	Anturi_2	→	bsylinteri2Out		

Check for N->1 Mapping

OK Cancel

Signaalien lisääminen CODESYS ympäristöön

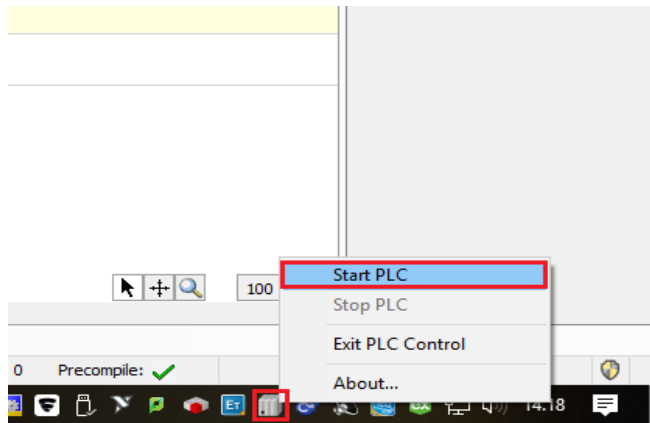
External Signal Adapter-toiminto



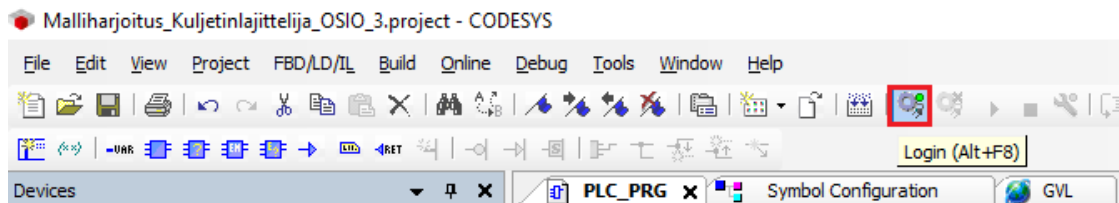
CODESYS-ohjelmisto

Ohjelman käynnistys

1. Käynnistä CODESYS **Start PLC**.

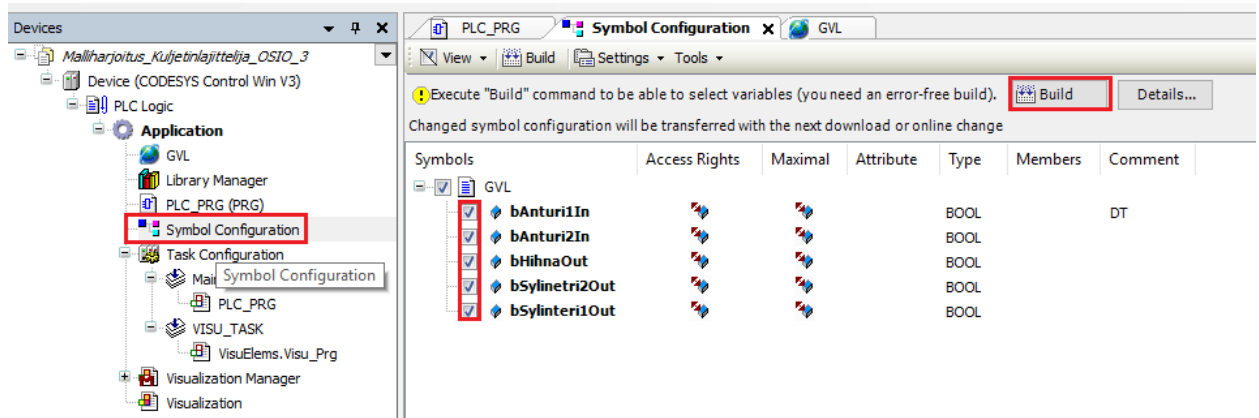


2. Kirjaudu sisään **Login**.



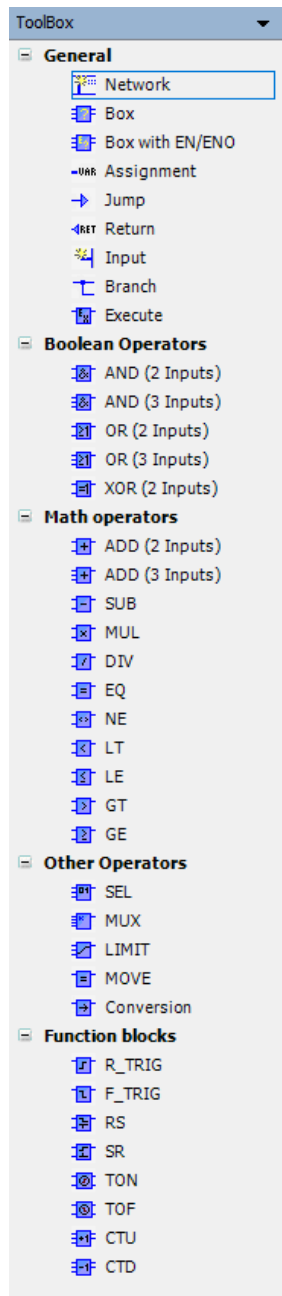
Signal Configuration-toiminto

Valitaan kaikki luodut signaalit ja hyväksytään ne **Build**-painikkeesta.

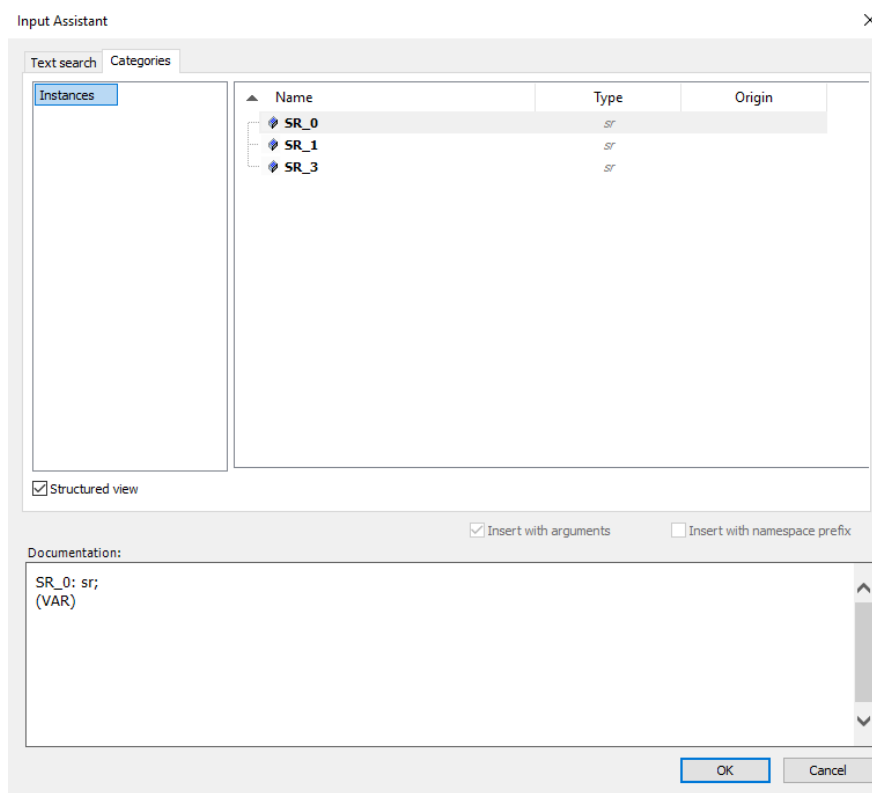
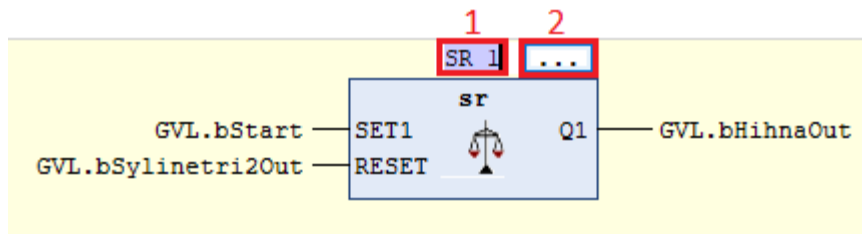


Lohkokaaviot

Lohkokaavioiden rakentamista varten löytyy **ToolBox**-kuvakkeen alta tarvittavat työvälineet niiden rakentamiseksi.

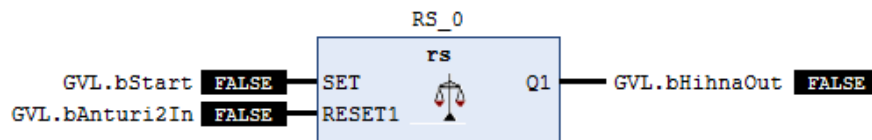


Lohkoja voidaan myös muokata klikkaamalla nimikettä, jolloin ilmestyy lisävalikko sen viereen. Samaa toimintoa voidaan käyttää myös muistipaikkojen kuin myös I/O uudelleen määrittämisessä.



TEHTÄVÄ OSIO 1.

SR ja RS ovat Function-blokkeja, joten ne tarvitsevat muuttujat. SR:ssä SET on dominoiva, kun taas RS:ssä RESET on dominoiva blokki.

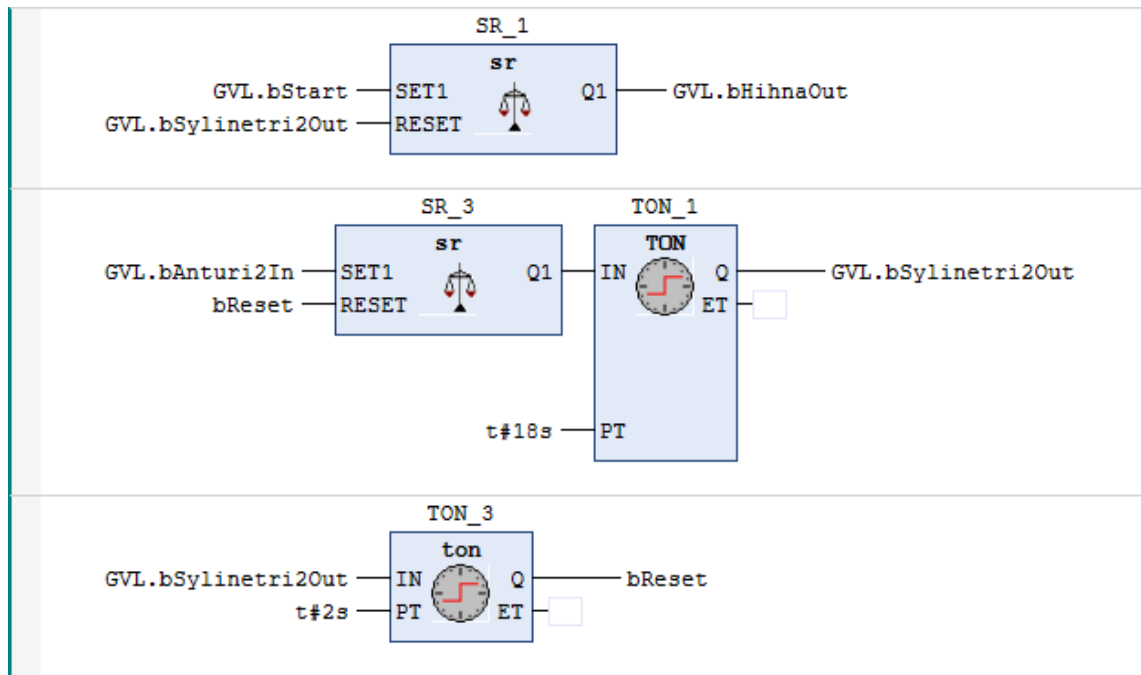


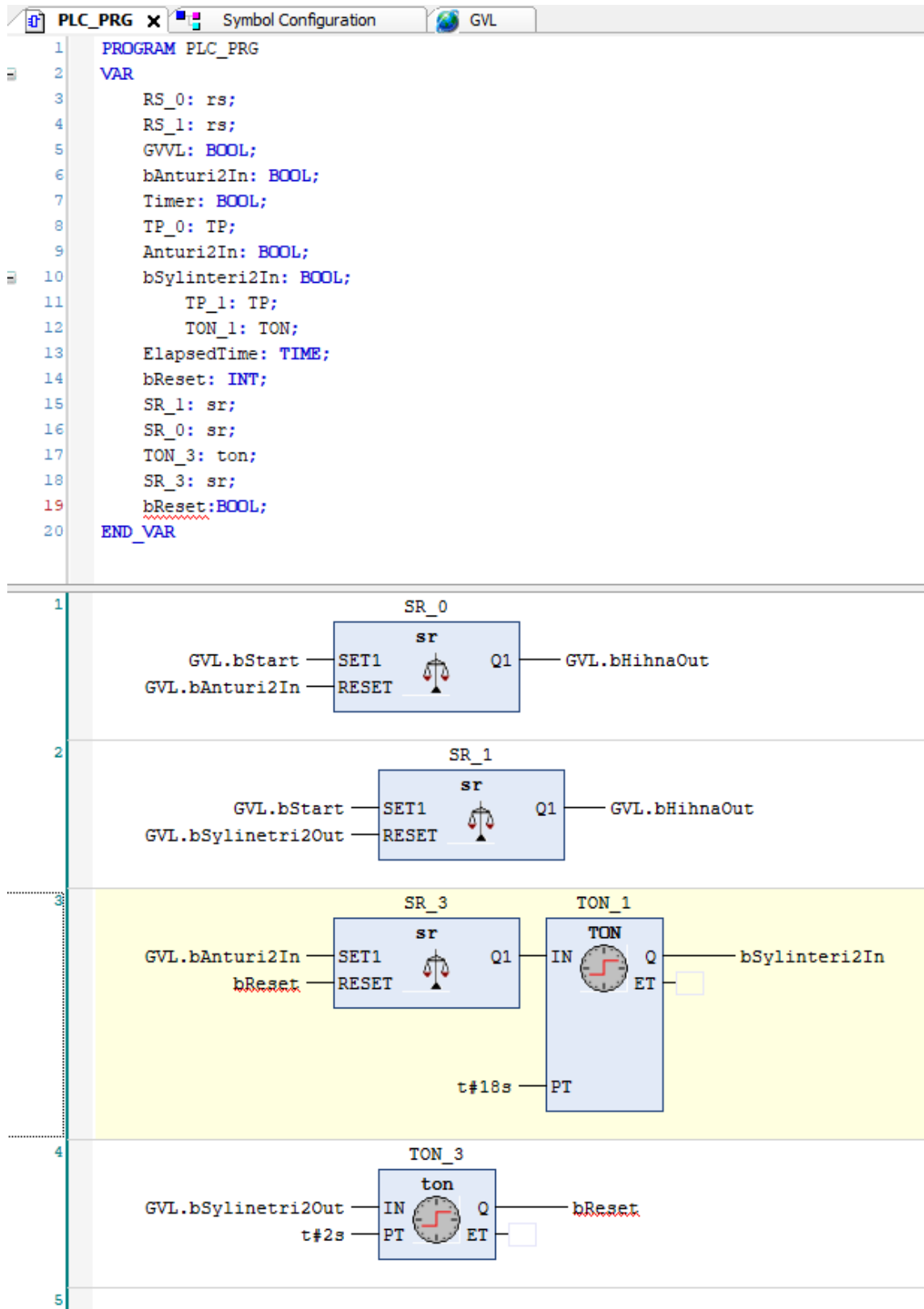
TEHTÄVÄ OSIO 2.

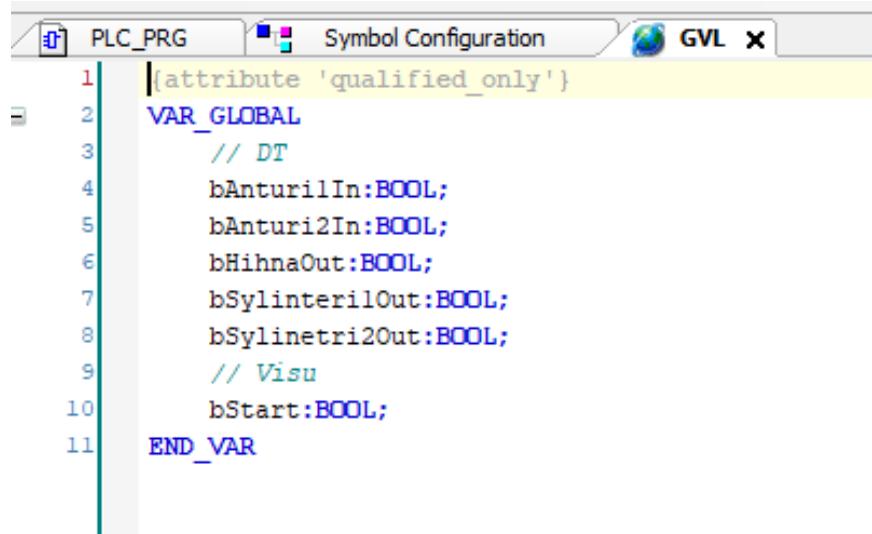
Silloin kun kappale saavuttaa bAnturi2-kohdan lähtee lohkoista SR_2 TRUE impulssi, joka kulkeutuu TON_1-ajastimelle ja alkaa 18 sekuntia kestävä laskenta. Tässä ajassa kappale on saavuttanut bSylinteri2, joka liikkuu asentoon 1 ja pudottaa kappaleen pois hihnalta. Tämän jälkeen TON_2 -ajastin alkaa laskea 2 sekuntia, jolloin bSylinteri2 palautuu 0-asentoon.

CODESYS-ohjelmisto

```
PLC_PRG x Symbol Configuration GVL
1 PROGRAM PLC_PRG
2 VAR
3     RS_0: rs;
4     RS_1: rs;
5     GVVL: BOOL;
6     bAnturi2In: BOOL;
7     Timer: BOOL;
8     TP_0: TP;
9     Anturi2In: BOOL;
10    bSylinteri2In: BOOL;
11        TP_1: TP;
12        TON_1: TON;
13    ElapsedTime: TIME;
14    SR_1: sr;
15    SR_0: sr;
16    TON_3: ton;
17    SR_3: sr;
18    bReset:BOOL;
19 END_VAR
```







```
1  |[attribute 'qualified_only']
2  VAR_GLOBAL
3      // DT
4      bAnturilIn:BOOL;
5      bAnturi2In:BOOL;
6      bHihnaOut:BOOL;
7      bSylinteri1Out:BOOL;
8      bSylinteri2Out:BOOL;
9      // Visu
10     bStart:BOOL;
11 END_VAR
```

Kirjautuminen sisään **Log in**.

The screenshot displays the SIMATIC Manager interface for a project named 'Maailmanpöytä_kuljetinjärjestelmä_OSIO_3'. The main window shows a ladder logic program for 'PROGRAM PLC_PRG'. The program consists of several rungs (1-5) and variable declarations (VAR).

Program Code:

```
1 PROGRAM PLC_PRG
2 VAR
3   RS_0: rs;
4   RS_1: rs;
5   GV1: BOOL;
6   hänturiz2in: BOOL;
7   timer: BOOL;
8   TP_0: TP;
9   Anturiz2in: BOOL;
10  bsylineeriz2in: BOOL;
11  TP_1: TP;
12  TON_1: TON;
13  ElapsedTime: TIME;
14  SR_1: sr;
15  SR_0: sr;
16  TON_3: ton;
17  SR_3: sr;
18  Breset:BOOL;
19 END_VAR
```

Ladder Logic Rungs:

- Rung 1: SR_0 coil, set by GV1.bStart (SETI) and reset by GV1.bAnturiz2in (RRESST).
- Rung 2: SR_1 coil, set by GV1.bStart (SETI) and reset by GV1.bSylineeriz2out (RRESST).
- Rung 3: SR_3 coil, set by GV1.bAnturiz2in (SETI) and reset by BReset (RRESST). This rung is connected to a TON_1 timer (TON) with a preset time of t#18s. The timer output Q is connected to GV1.bSylineeriz2out.
- Rung 4: TON_3 timer (TON) with a preset time of t#2s. The timer output Q is connected to BReset.
- Rung 5: BReset coil, set by GV1.bSylineeriz2out (IN) and reset by t#2s (ET).

A dialog box titled 'CODESYS' is overlaid on the program, displaying the message: 'Application changed since last download. What do you want to do?'. The 'Options' section is expanded, showing the following choices:

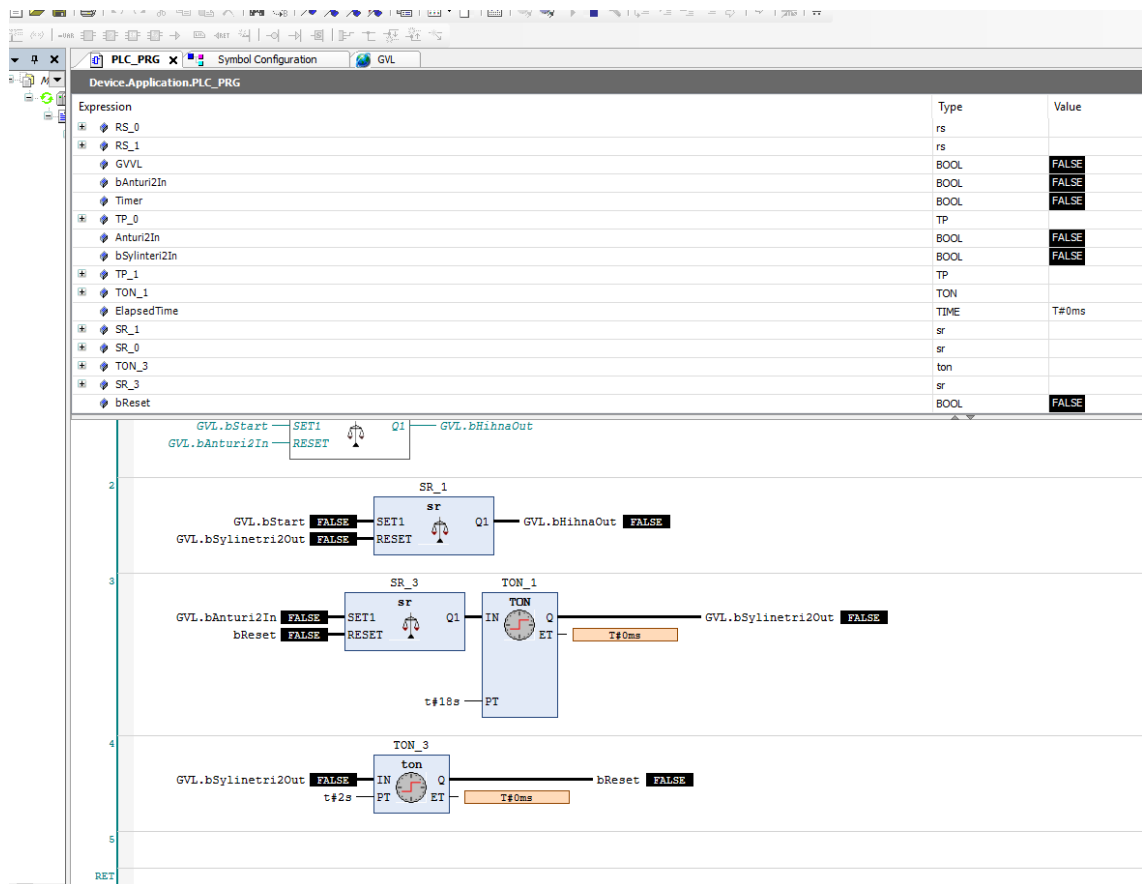
- Login with online change.
- Login with download.
- Login without any change.
- Update bootproject

The 'OK' button is highlighted with a red rectangle.

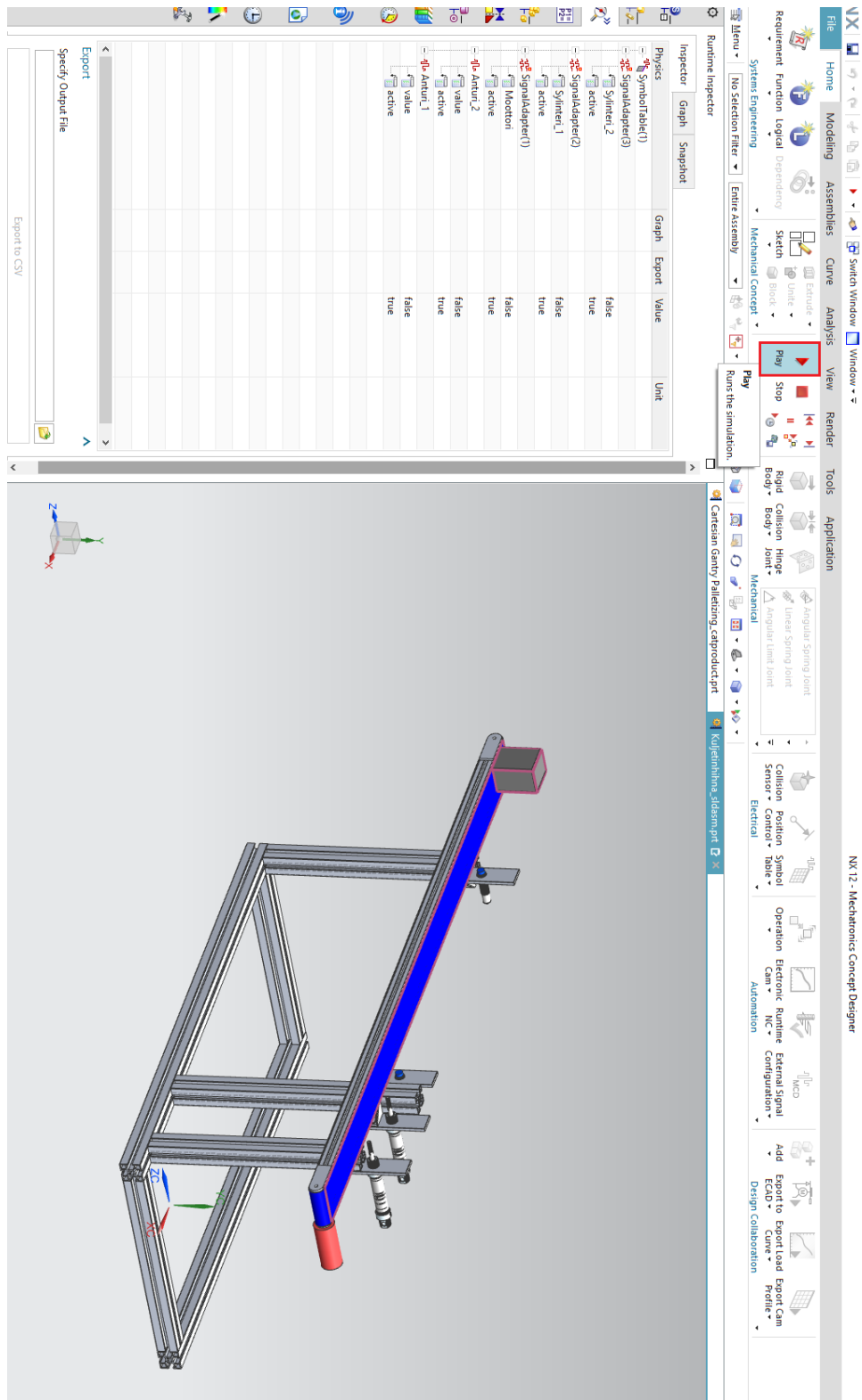
Start -> käynnistys.



Lohko muuttuu aktiiviseksi, merkiksi taustaväri muuttuu vihreäksi rakennepuussa.

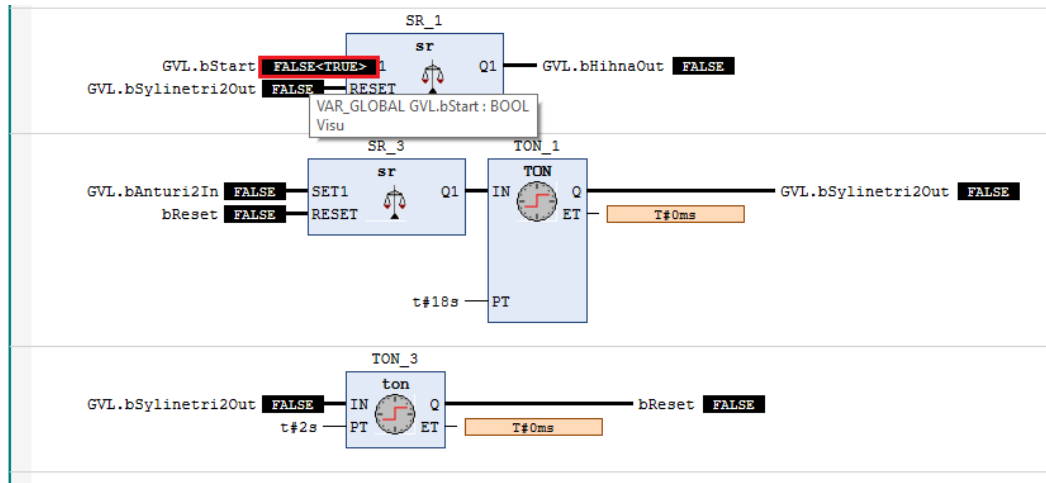


NX.MCD-ohjelmisto



Painetaan simulaatio päälle **Play**-painikkeesta ja linjastolle ilmestyy ääriiviivat merkiksi sen aktivoitumisesta.

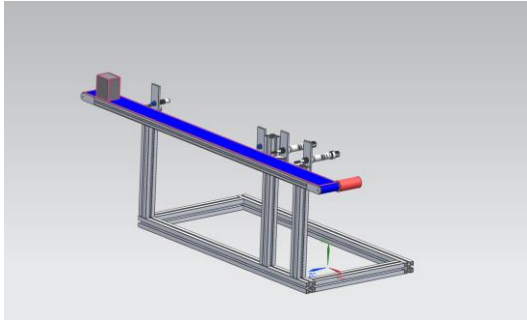
CODESYS-ohjelmisto



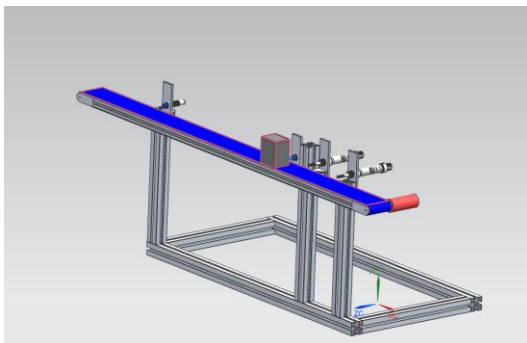
Valitse **GVL.bStart** klikkaamalla kahdesti ja paina **Ctrl + F7**.

NX.MCD-ohjelmisto

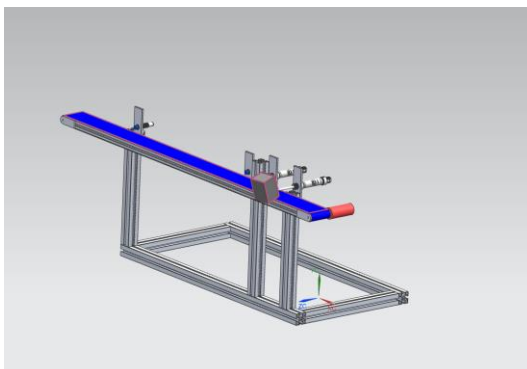
Simulaation lähtee liikkeelle heti kun **START** on painettu.



Kappale lähtee liikkeelle linjaston päässä.



Kakkosanturi tunnistaa kappaleen ja laskuri lähtee käyntiin.



18 sekunnin jälkeen sylinteri aktivoituu ja työntää kappaleen pois hihnalta ja pysähtyy.