

Eri teollisuuslogiikoiden ohjelmointialustojen käyttö ja dokumentointi

Olli Huotari

OPINNÄYTETYÖ
Huhtikuu 2025

Sähkö- ja automaatiotekniikka
Automaatiotekniikka

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Sähkö- ja automaatiotekniikan tutkinto-ohjelma
Automaatiotekniikka

HUOTARI, OLLI:

Eri teollisuuslogiikoiden ohjelmointialustojen käyttö ja dokumentointi

Opinnäytetyö 72 sivua, joista liitteitä 43 sivua
Huhtikuu 2025

Tässä opinnäytetyössä laadittiin dokumentaatio neljälle JIS-Automation Oy:n käyttämälle logiikkaohjelmointialustalle. Dokumentaation tarkoitus oli tarjota sekä uusille että alustojen pariin palaaville työntekijöille selkeää ja kattava kuva logiikkaohjelmointialustojen perusominaisuuksista.

Työssä perehdytään logiikkaohjelmointiin liittyvään IEC 61131-standardiin, oppimisen teoriaan lyhyesti, sekä logiikkaohjelmointialustoihin, joille dokumentaatiot tehtiin. Lisäksi tarkastellaan dokumentaatioiden rakennetta ja työn toteutusprosessia. Valmiit dokumentaatiot ovat opinnäytetyön liitteinä.

Dokumentaation rakenteen suunnittelussa hyödynnettiin tilaajan edustajan haastattelun sekä ohjelmoijille suunnatun anonyymien kyselyn tuloksia. Lisäksi työhön saatiin henkilökohtainen näkökulma, sillä osaan ohjelmointialustoista perehdyttiin ensimmäistä kertaa opinnäytetyön aikana. Lopputuloksena saatiin neljä dokumentaatiota, jotka vastasivat niille asetettuja vaatimuksia ja joihin tilaaja oli tyytyväinen.

Asiasanat: Logiikkaohjelmointi, dokumentaatio, alusta

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Electrical and Automation Engineering
Automation Engineering

HUOTARI, OLLI:

The Use and Documentation of Various Industrial Logic Programming Platforms

Bachelor's thesis 72 pages, appendices 43 pages
April 2025

The purpose of this thesis work was to create documentation for four logic programming platforms used by JIS-Automation Oy. The purpose of the documentation was to provide a clear and comprehensive overview of the basic features of these logic programming platforms to new employees and those returning to these platforms.

The work delved into the IEC 61131 standard related to logic programming, briefly touched on learning theory and focused on the logic programming platforms which the documentation was created on. Additionally, it examined the structure of the documentation and the implementation process of the work. The completed documentation was included as attachments to the thesis.

The design of the documentation structure utilized results from an interview with the client's representative and an anonymous survey conducted with programmers in JIS-Automation Oy. Moreover, a personal perspective was incorporated into the work, as some of the programming platforms were explored for the first time during the thesis process. As a result, four pieces of documentation were produced meeting the set requirements and satisfying the client.

Key words: PLC programming, documentation, platform

SISÄLLYS

1	JOHDANTO	6
2	JIS-automation.....	7
3	IEC 61131 standardi	8
	3.1. Eri ohjelmointikielät	9
	3.1.1 LD-ohjelmointikielen perusteet	10
	3.1.2 FBD-ohjelmointikielen perusteet.....	11
	3.1.3 ST-ohjelmointikielen perusteet	12
4	ERI OHJELMISTOALUSTOJEN OPPIMISEEN LIITTYVÄT HAASTEET JA NÄKÖKULMIA OPPIMISTEORIOISTA.....	14
	4.1. Eri ohjelmistoalustojen oppimisen haasteet ja mahdollisuudet	14
	4.2. Oppimisen ja kertaamisen teoria	15
5	TYÖSSÄ KÄYTETYT OHJELMOINTIALUSTAT	16
	5.1. Omron Sysmac Studio	16
	5.2. Omron CX-Programmer	17
	5.3. Beckhoff TwinCAT 3	17
	5.4. Schneider Machine Expert	20
6	DOKUMENTAATIO.....	22
	6.1. Työssä tehtävien dokumentaatioiden lähtökohdat ja tavoitteet	22
	6.2. Dokumentaatioiden rakenne	22
7	TYÖN SUORITUS	24
	7.1. Aloitus ja haastattelut	24
	7.2. Dokumentaatioiden rakenne	25
	7.3. Dokumentaatioiden eroavaisuudet ja arviointi.....	27
8	POHDINTA	28
	LÄHTEET.....	29
	LIITTEET	30
	Liite 1. TwinCAT 3-dokumentaatio.....	30
	Liite 2. Omron CX-Programmer-dokumentaatio.....	43
	Liite 3. Omron Sysmac Studio-dokumentaatio.....	51
	Liite 4. Schneider Machine Expert-dokumentaatio.....	62

LYHENTEET JA TERMIT

FBD	Function Block Diagram eli lohkokaavio-ohjelmointi
IL	Instruction List eli käskyluettelo
IoT	Internet Of Things eli esineiden internet
LD	Ladder Diagram eli tikapuukaavio
PLC	Ohjelmoitava logiikka
SFC	Sequential Function Chart eli sekvenssikaavio
ST	Structured Text eli tekstimuotoinen ohjelmointikieli

1 JOHDANTO

Tämän opinnäytetyön tavoitteena on laatia JIS-Automation Oy:lle dokumentaatio, joka auttaa uusia työntekijöitä perehtymään sekä kokeneita työntekijöitä kertamaan eri logiikkaohjelmointialustojen perusominaisuuksia helposti ja tehokkaasti.

JIS-Automationilla on ollut käytössään laaja valikoima eri valmistajien logiikkaohjelmointialustoja. Työntekijöiden luonnollinen vaihtuvuus, yrityksen kasvu sekä työtehtävien monipuolistuminen johtavat tilanteisiin, joissa esimerkiksi vanhojen järjestelmien päivittämisen tullessa ajankohtaiseksi, saattaa työn tehdä tekijä, jolla ei ole tuoretta kosketuspintaa kyseiseen logiikkaohjelmointialustaan. Tällaisissa tapauksissa selkeä ja helposti saatavilla oleva dokumentaatio vähentää alustaan uudelleen perehtymiseen kuluvaan aikaa, ehkäisee virheitä ja parantaa työn tehokkuutta. Lisäksi uusien työntekijöiden perehdyttäminen kyseisiin logiikkaohjelmointialustoihin on yhtenäistä ja selkeää, kun ohjelmointialustojen perusominaisuudet ovat helposti selvitettävissä dokumentaatioiden avulla.

Tämä opinnäytetyö keskittyy kokoamaan ja jäsentämään olennaiset tiedot eri logiikkaohjelmointialustoista siten, että ne ovat helposti ymmärrettävissä ja hyödynnettävissä käytännön työssä. Dokumentaatio tehtiin neljälle tilaajayrityksessä laajimmin käytössä olevalle ohjelmointialustalle: TwinCAT 3:lle, Omron CX-Programmerille ja Sysmac Studiolle, sekä Schneider Machine Experille, ja niissä keskitytään ohjelmien perusominaisuuksiin, kuten online-muutosten tekoon, varsinaista ohjelmointia ei kuitenkaan niissä käsitellä. Dokumentaatioiden sisällön määrittämisessä käytettiin apuna haastatteluita tilaajan edustajan kanssa, sekä yrityksen ohjelmoijille teetettyä anonyymiä kyselyä dokumentaatioiden rakenteesta.

Lisäksi opinnäytetyössä esitellään tarkemmin siinä käsiteltävät logiikkaohjelmointialustat, tarkastellaan yleisimpiä ohjelmointikieliä ja standardeja sekä havainnollistetaan eri logiikkaohjelmointialustojen eroja selkeiden esimerkkien avulla.

2 JIS-Automation Oy

JIS-Automation Oy on Pirkkalassa sijaitseva vuonna 2003 perustettu osakeyhtiö. Yhtiö suunnittelee ja toteuttaa erinäisiä kiinteistöautomaation ja teollisuusautomaation järjestelmiä, muun muassa savunpoistojärjestelmiä ja ohjauskeskuksia. JIS-Automation työllistää noin 200 alansa ammattilaista kuudessa kaupungissa, Pirkkalan lisäksi toimipisteitä on Helsingissä, Turussa, Oulussa, Jyväskylässä, Rovaniemellä ja Lahdessa. Yhtiö on historiansa aikana toimittanut yli 2000 onnistunutta automaatioprojektia, muun muassa kauppakeskus Triplan ja Nokia Areenan savunpoistojärjestelmät sekä HUS-tammisairaalan leikkaussaliautomaatiojärjestelmät. (JIS-Automation Oy, 2024)

Yritys on kasvanut viime vuosina voimakkaasti, varsinkin teollisuusautomaation alalla, ja se onkin noussut alihankkijan asemasta täysiveriseksi automaatiourakoijaksi tällä saralla. Tätä siirtymää ovat vauhdittaneet strategisesti järkevät liiketoimien hankinnat ja integroinnit alan muiden toimijoiden kanssa. Vuonna 2024 JIS-Automation Oy ja Integratio Oy yhdistyivät ja heti tämän jälkeen JIS-Automation integroitui DDC-Tekniikka Oy:n kanssa. Vuoden 2025 alussa JIS-Automation Oy osti Lapin Automaatiopalvelut Oy:n saaden näin toimipisteen Pohjois-Suomen markkinoille. Myös talousluvut tukevat vahvaa kasvua, vuonna 2024 yrityksen liikevaihto kasvoi 20 milj. eurosta 29 milj. euroon (JIS-Automation Oy).

JIS-Automation Oy:n tulevaisuuden näkymät ovat positiiviset, ja näyttäisi siltä, että yritys jatkaa tulevana vuosina vahvaa kasvuaan. Yrityksen tarina on hieno esimerkki kuinka kahden hengen automaatioyrityksestä voi kasvaa menestyvä maanlaajuinen toimija teollisuusautomaation alalla.

3 IEC 61131 standardi

Automaatioalalla on käytössä useita eri ohjelmointialustoja, sillä lähes jokaisella PLC-toimittajilla on oma ohjelmointialustansa, jotka eroavat toisistaan ainakin jossain määrin. IEC 61131- standardin tavoitteena on luoda logiikkaohjelmointialustojen toimittajille yhteinen ohjeistus ja suuntaviivat, joiden avulla voidaan yhdenmukaistaa muun muassa ohjelmistojen rakenne, ohjelmointikielet sekä muuttujien käsittely. Tämä yhdenmukaistaminen helpottaa eri toimijoiden ohjelmointialustojen käyttöönottoa, keventää ohjelmoijien työtaakkaa sekä vähentää ohjelmointivirheitä, mikä puolestaan lyhentää vianhakuun vaadittavaa aikaa (John & Tiegelkamp, 12–13).

IEC 61131-standardi on 10:n vanhan kansainvälisen standardin (IEC 50, IEC559, IEC 617-12, IEC 617-13, IEC 848, ISO/ANFOR, ISO/IEC 646, ISO 8601, iso 7185 ja ISO 7498) yhdistelmä ja jatkaja. Jo vuodesta 1977 lähtien, jolloin PLC:t ja logiikkaohjelmointi alkoivat yleistyä teollisuudessa, alalla on ollut tarve yhtenäiselle standardille. IEC 61131 on ensimmäinen logiikkaohjelmointi koskeva standardi, joka on saanut laajan kansainvälisen hyväksynnän eri toimijoilta (John & Tiegelkamp 2010, 13–14).

IEC 61131-standardi on jaettu kahdeksaan osaan, joista tässä opinnäytetyössä keskitytään osaan kolme: Ohjelmointikielet. Eri osien sisällöt on esitetty taulukossa 1.

TAULUKKO 1. IEC 61131-standardin eri osat (John & Tiegelkamp, 14–15).

Standardin osa	Sisältö
1	Yleistä tietoa standardista: Sisältää yleisiä määritelmiä
2	Laitteistovaatimukset: Määritellään PLC-laitteiston vaatimukset
3	Ohjelmointikielet: Määritellään käytettävät ohjelmointikielet
4	Käyttöohjeet: Käyttöohjeita koko projektin elinkaarelle
5	Kommunikaatio: PLC-laitteiden ja muiden laitteiden välinen viestintä
6	Turvallisuuskriittiset PLC:t, ja niiden mukauttaminen muihin standardeihin (IEC 61508 ja IEC 62061)
7	Sumean logiikan ohjelmointikieli: Määrittelee standardin sumean logiikan sovelluksille ja niiden integroinnille.
8	Ohjelmointikielten käyttöä ja sovellusta koskevia ohjeistuksia

3.1. Eri ohjelmointikielet

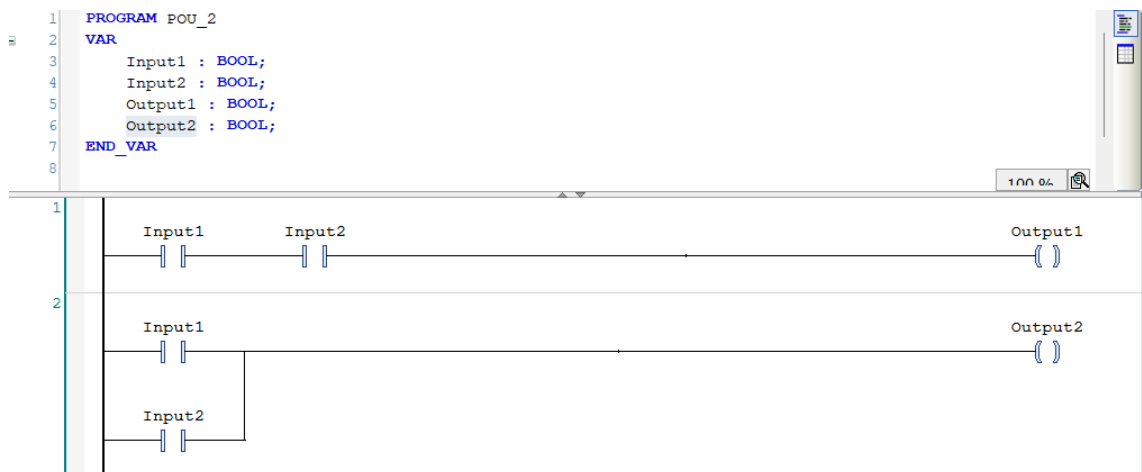
IEC 61131- standardi sisältää useita eri ohjelmointikieliä, joihin kuuluvat LD (Ladder Diagram), SFC (Sequential Function Chart), FBD (Function Block Diagram), IL (Instruction List) ja ST (Structured Text). Tässä opinnäytetyössä keskitytään näistä kolmeen yleisimmin käytettyyn: LD, FBD ja ST. Ohjelmointikielistä SFC on käytössä pääasiassa prosessiteollisuudessa, ja muissa sovelluksissa, joissa prosessi voidaan jakaa selkeisiin sekvensseihin. IL puolestaan on harvoin käytetty ohjelmointikieli, sillä sitä pidetään monimutkaisena ja vikaherkkyytensä vuoksi haasteellisena. Sen vahvuutena on kuitenkin koodin suorittamiseen vaadittava vähäinen muistin tarve ja nopea suoritus aika (White 2023, 21–22).

Ohjelmoija voi halutessaan yhdistää useita ohjelmointikieltä samaan ohjelmaan hyödyntäen kunkin kielen vahvuuksista ja välttämällä niiden heikkouksia. Esimerkiksi sekvenssiohjelma voi olla helpointa toteuttaa LD- tai FBD-kielillä, kun taas ehtolausekkeet voidaan kirjoittaa selkeämmin ST-kielillä. Ohjelmointikieltä voidaan yleisesti vaihtaa kesken ohjelmakierron tai joissain tapauksissa jopa kesken sekvenssin.

3.1.1 LD-ohjelmointikielen perusteet

LD on yleisimmin käytetty ohjelmointikieli logiikkaohjelmoinnissa. Sen suosio johtuu pitkälti ohjelmointikielen pitkästä historiasta, graafisesta esitystavasta sekä oppimisen helppoudesta verrattuna tekstipohjaisiin ohjelmointikieliin (Hansen 2015, 317). LD-ohjelmointikielen perusrakenne on esitetty kuvassa 1.

Graafisesti LD on suunniteltu muistuttamaan relekaaviota. Ohjelman rakenne koostuu logiikkariveistä, joita kutsutaan nimellä rung, ja ne suoritetaan järjestyksessä ylhäältä alaspäin ja vasemmalta oikealle. Jokainen logiikkarivi päättyy toimintoon, joka aktivoituu, jos asetetut ehdot täyttyvät. LD:lle erityispiirteenä verrattuna esimerkiksi FBD-ohjelmointikieleen on, että kaavio ei pidä sisällään AND- tai OR-piirejä, vaan vastaavat ehdot täytyy toteuttaa kontaktien avulla. Kuvassa 1 logiikkarivillä 1 on toteutettu AND-piiri ja logiikkarivillä 2 OR-piiri LD-ohjelmointikielellä. Ensimmäisellä logiikkarivillä oleva "Output 1" menee "TRUE"-tilaan kun kontaktit "Input 1" ja "Input 2" ovat molemmat aktivoituna, kun taas toisella logiikkarivillä oleva "Output 2" menee "TRUE"-tilaan kun toinen kontakteista on aktivoituneena.



KUVA 1. AND- ja OR-piirit tehtynä LD-ohjelmointikielellä.

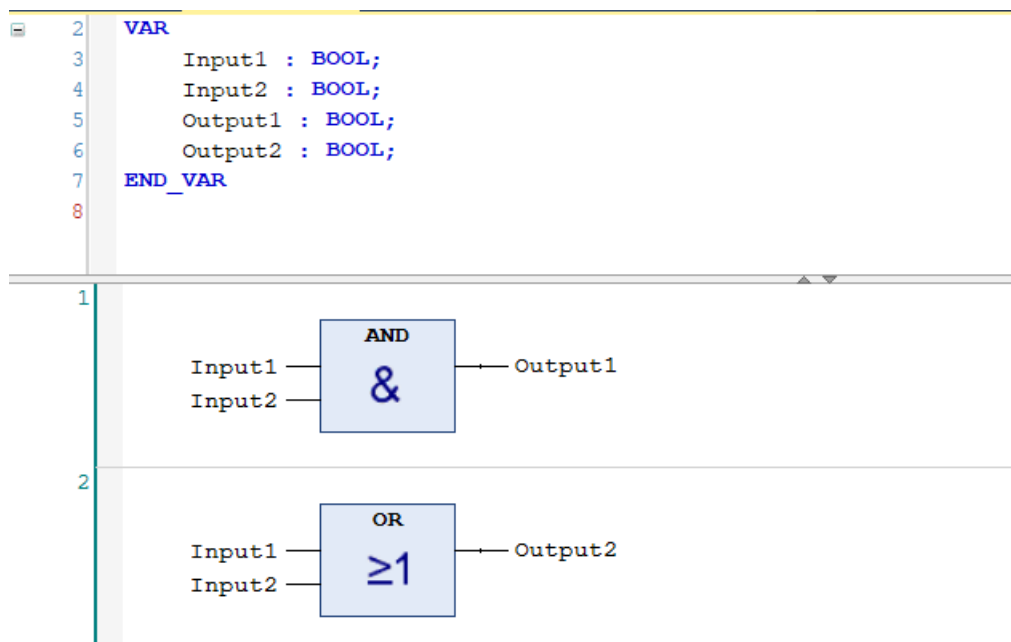
LD on erityisen hyödyllinen sovelluksissa, joissa tarvitaan yksinkertaisia ja helposti ymmärrettäviä ohjauslogiikoita, kuten koneteollisuudessa ja tuotantolinjojen hallinnassa. Vaikka LD on helppo oppia ja käyttää, sen graafinen rakenne voi

tehdä laajoista ohjelmista vaikeasti hallittavia. Varsinkin monimutkaisten loogisten ehtojen ja laskennan toteutus LD-kielellä johtaa usein sekavaan kokonaisuuteen, jota on vaikea ylläpitää.

3.1.2 FBD-ohjelmointikielen perusteet

FBD on graafinen ohjelmointikieli, jonka perustoiminnallisuus on hyvin samankaltainen kuin aikaisemmin esitetyllä LD-kielellä. Suurin ero LD- ja FBD-kielten välillä on niiden tapa käsitellä ehtoja ohjelmalogiikassa. LD-kielessä kontaktit toimivat kuten mekaaniset kytkimet, ja ohjelmarivin toiminta määräytyy sen mukaan, ovatko kontaktit auki vai kiinni. FBD-kielessä taas ehtoja käsitellään suoraan muuttujien avulla, jolloin ohjelmarivin sisääntuloon voidaan asettaa loogisia ehtoja ilman erillisiä kontakteja.

FBD soveltuu erityisesti monimutkaisten ohjausjärjestelmien ohjelmointiin, sillä se mahdollistaa lohkojen uudelleenkäytön ja selkeämmän visuaalisen esitystavan kuin esimerkiksi LD. Kuvassa 2 on esitetty AND-piiri sekä OR-piiri FBD-ohjelmointikielellä toteutettuna.



KUVA 2. AND- ja OR-piirit toteutettuna FBD-ohjelmointikielellä.

FBD:n vahvuutena on sen selkeä ja looginen rakenne, joka helpottaa monimutkaisten prosessien hahmottamista ja ohjelman toimintalogiikan visualisointia. Se

sopiikin erityisesti tilanteisiin, joissa ohjelmassa käytetään paljon matemaattisia operaatioita, loogisia ehtoja ja aikareleitä. Toisaalta FBD:n haasteena voi olla sen tilan tarve, suuret ohjelmat saattavat muuttua vaikeasti hallittavaksi, jos lohkot ja niiden yhteydet kasvavat liian monimutkaisiksi.

3.1.3 ST-ohjelmointikielen perusteet

ST eroaa muista logiikkaohjelmoinnissa käytettävistä ohjelmointikielistä siinä, että se perustuu tekstipohjaiseen syntaksiin eikä graafiseen esitystapaan. ST muistuttaa rakenteeltaan vanhojen BASIC- ja Ada-ohjelmointikielten yhdistelmää (White 2024, 175). Kuvassa 3 on esitetty AND- ja OR-piiri ST-kielillä toteutettuna.

```
1 PROGRAM POU
2 VAR
3     Input1 : BOOL;
4     Input2 : BOOL;
5     Output1 : BOOL;
6     Output2 : BOOL;
7 END_VAR
8
9
10 IF Input1=TRUE AND Input2=TRUE THEN
11     Output1:=TRUE;
12 END_IF
13
14 IF Input1=TRUE OR Input2=TRUE THEN
15     Output2:=TRUE;
16 END_IF
```

KUVA 3. AND- ja OR-piiri ST-ohjelmointikielillä toteutettuna.

ST:n etuna on sen luettavuus ja joustavuus monimutkaisissa ohjausjärjestelmissä. Graafisista ohjelmointikielistä poiketen ST mahdollistaa monimutkaisten laskentojen, ehtolausekkeiden ja toistorakenteiden tehokkaan toteutuksen. Lisäksi ohjelmoijan tarvitsee muistaa selkokielisiä avainsanoja symbolien sijaan, mikä parantaa koodin ymmärrettävyyttä ja helpottaa sen ylläpitoa.

ST:n erityinen vahvuus on sen samankaltaisuus korkean tason ohjelmointikielten, kuten Pythonin ja C:n kanssa. Tämä tekee siitä houkuttelevamman vaihtoehdon ohjelmoijille, joilla on aiempaa kokemusta perinteisestä ohjelmistokehityksestä.

Lisäksi ST tarjoaa muihin logiikkaohjelmointikieliin verrattuna laajan tuen matemaattisille operaatioille ja tietorakenteille, mikä mahdollistaa kehittyneiden algoritmien toteutuksen ilman tarvetta siirtyä erillisiin ohjelmointiympäristöihin.

Tulevaisuudessa ST:stä voi tulla logiikkaohjelmoinnin yleisimmin käytetty kieli, sillä modernit automaatiojärjestelmät sisältävät yhä useammin IoT-laitteita, tekoälyä ja muita edistyksellisiä komponentteja. ST:n kyky integroitua näihin teknologioihin tekee siitä erityisen soveltuvan tulevaisuuden automaatoratkaisuihin (White 2024, 177).

Vaikka ST tarjoaa monia etuja, sen haasteena voi olla korkea oppimiskynnys erityisesti graafisiin logiikkaohjelmointikieliin tottuneille käyttäjille. Lisäksi ST:n tekstipohjaisuudesta johtuen sen vianhaku voi olla haastavampaa kuin graafisissa ohjelmointikielissä ilman sopivia työkaluja.

4 ERI OHJELMISTOALUSTOJEN OPPIMISEEN LIITTYVÄT HAASTEET JA NÄKÖKULMIA OPPIMISTEORIOISTA

Logiikkaohjelmointi on keskeinen osa automaatiojärjestelmien suunnittelua ja käyttöönottoa, mutta sen ohjelmistoalustojen oppiminen voi olla haastavaa varsinkin uusille käyttäjille tai ohjelmistoalustan pariin palaaville ammattilaisille. Vaikka lähes kaikki ohjelmistoalustat pohjautuvat IEC-61131-standardiin, on niiden käyttöliittymissä ja perusominaisuuksissa eroavaisuuksia, jotka voivat aiheuttaa ristiriitoja muiden ohjelmistoalustojen kanssa. Lisäksi tilanne voi lisätä kognitiivista kuormaa erityisesti asiantuntijoille, joilla kognitiivinen kuorma työpäivän aikana on lähtökohtaisesti jo ennestään suuri.

4.1. Eri ohjelmistoalustojen oppimisen haasteet ja mahdollisuudet

Logiikkaohjelmointialustojen oppimisessa suurimpia haasteita ovat eri ohjelmistojen poikkeavat käyttöliittymät, ohjelmointimenetelmät ja toimintaperiaatteet. Vaikka IEC 61131-standardi yhtenäistää logiikkaohjelmointia tukemalla eri ohjelmointikieliä (mm. ST, LD ja FBD), kommunikaatioprotokollia, muuttujien nimeämiä sekä datamuotoja ja logiikkaohjelman rakennetta, käytännön toteutus vaihtelee ohjelmistosta toiseen. Ohjelmistoalustan opittavuus on kuitenkin keskeinen osa käyttökokemusta, ja sen merkitys kasvaa varsinkin pitkän aikavälin käyttäjätyytyväisyydessä ja ohjelmiston käytön tehokkuudessa (Heikkinen 2024)

Eri ohjelmointialustojen oppimisessa haasteita aiheuttaa eniten käyttöliittymien eroavaisuudet ja dokumentaation sekä ohjeistuksen saatavuus. Ohjelmistoalustoilla on lähtökohtaisesti aina erilainen tapa toteuttaa paljon käytettäviä, ja näin ollen tärkeitä ominaisuuksia. Tästä hyvä esimerkki on online-edit, joka mahdollistaa muutosten tekemisen ohjelmaan niin, että ohjelma on käynnissä, eli ohjelman pyörittämää prosessia ei tarvitse pysäyttää muutoksen teon ajaksi. Jos päivityksen tekijällä ei ole tuoreessa muistissa, kuinka online-edit tehdään käyttämällään ohjelmointialustalla, on virheen todennäköisyys suuri ja pahimmillaan päivitettävä prosessi pysähtyy. Myös dokumentaation sekä ohjeistuksen saatavuus on heikkoa logiikkaohjelmointialustoissa, tai vaikka sellainen olisikin saatavilla, se on usein joko epäselvää satunnaiselle käyttäjälle tai sieltä on erittäin vaikea löytää etsimänsä asia.

Vaikka ohjelmistoalustojen käytössä ja sen oppimisessa on haasteensa, on niissä myös monia seikkoja, joita voi pitää merkittävänä mahdollisuuksina. Itse logiikkaohjelman teko on standardoinnista johtuen usein eri alustoilla melko samankaltaista, joka mahdollistaa ohjelmoijan käyttävän useaa ohjelmointialustaa työssään rinnakkain sulavasti. Lisäksi nykyaikaiset alustat tarjoavat simulaattoreita ja ohjevideoita, jotka helpottavat ohjelman testausta ja sen vianhakua.

4.2. Oppimisen ja kertaamisen teoria

Oppimisen voidaan katsoa olevan monivaiheinen ilmiö, jota tapahtuu koko ihmisen elämän ajan. Aikuisten kohdalla oppimista tapahtuu ainakin neljällä tavalla, työssäoppimisella, jokapäiväisellä oppimisella, tietoisella oppimisella ja vapaa-ajan oppimisella (Kallio 2020, 118). Näistä jokapäiväinen oppiminen tapahtuu arjen ohessa, työssäoppiminen työelämässä, tietoinen oppiminen esimerkiksi koulussa, tai kun muuten tietoisesti opiskellaan jotain asiaa, ja vapaa-ajan oppiminen esimerkiksi harrastuksissa.

Oppiminen voidaan katsoa olevan myös ihmisen kahden muistijärjestelmän, lyhytaikaisen muistin ja pitkäaikaisen muistin, yhteistyötä ja vuoropuhelua. Jonkin asian muistaminen edellyttää sitä, että opittu asia on siirretty lyhytaikaisesta muistista, eli työmuistista pitkäaikaiseen muistiin, ja sitten palautettu pitkäaikaisesta muistista työmuistiin, kun opittua asiaa taas tarvitaan (Päivänsalo 2020, n.d).

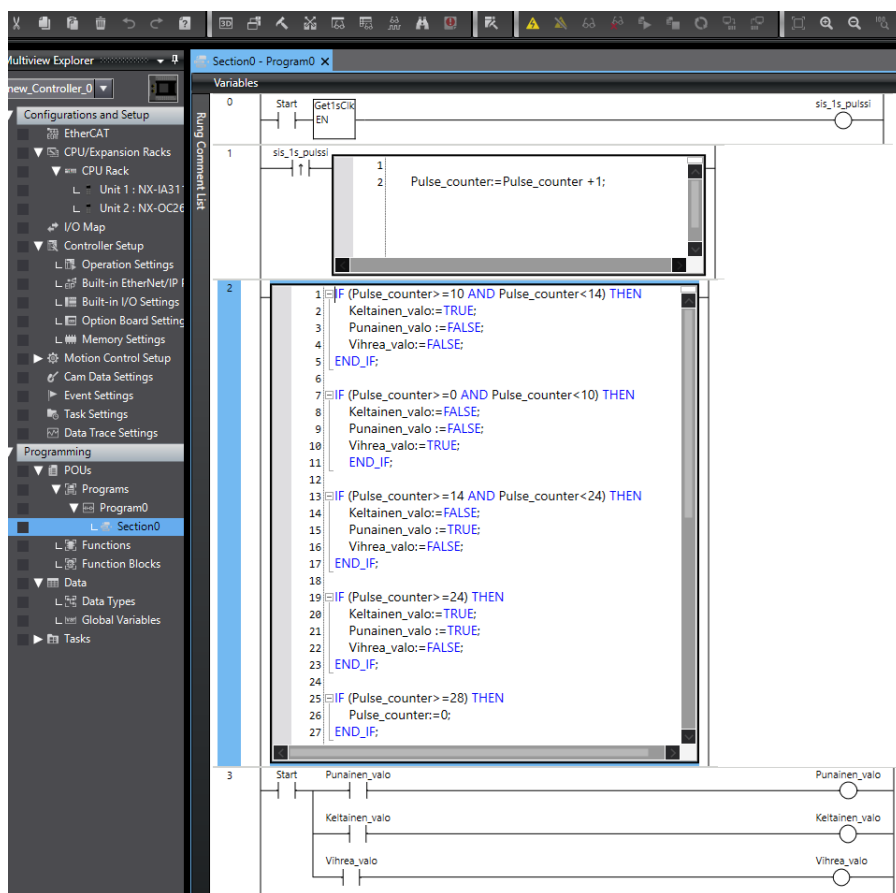
Eli vaikka jonkin asian olisi joskus osannut hyvin, on se palautettava pitkäaikaismuistista työmuistiin esimerkiksi kertaamalla. Opitun asian kertaamalla, esimerkiksi dokumentaatiosta tai havainnollistavien kuvien avulla voidaan paikata muistin aukkoja. Tämä ei ole opiskelumielessä tehokkain tapa oppia jokin asia mutta esimerkiksi jos työtehtävissä joutuu satunnaisesti käyttämään jotain logiikkaohjelmointialustaa, on sen perusominaisuuksien kertaaminen selkeästä dokumentaatiosta tehokas ja turvallinen tapa varmistaa, että ohjelman käyttö menee halumallaan tavalla.

5 TYÖSSÄ KÄYTETYT OHJELMOINTIALUSTAT

Tässä luvussa esitetään työssä käytetyt ohjelmointialustat, niiden perusominaisuudet ja mahdolliset erikoisuudet, ja esitetään kunkin ohjelmointialustan ulkonäköä ja toiminnallisuutta yksinkertaisella liikennevalosovelluksella.

5.1. Omron Sysmac Studio

Sysmac on Omronin uusi koneautomaatioympäristö, jonka ohjelmointiympäristönä toimii Sysmac Studio. Sysmac on ollut järjestelmänä käytössä jo vuodesta 1971. Sysmac Studioon on integroitu robotiikan sekä PLC- ja HMI-ohjelmointi, joten koko järjestelmän ohjelmointi hoituu yhdellä ohjelmalla (Omron n.d). Se tukee IEC 61131-standardin mukaisia ohjelmointikieliä, ja on suunniteltu erityisesti kone- ja liikeohjauksen sovelluksiin. Kuvassa 4 on esitetty Sysmac Studion ulkoasu esimerkiohjelman tehtäessä. Ohjelma toteutettiin LD-kielellä, mutta ehdot tehtiin ST-kielellä koodin yksinkertaistamiseksi.

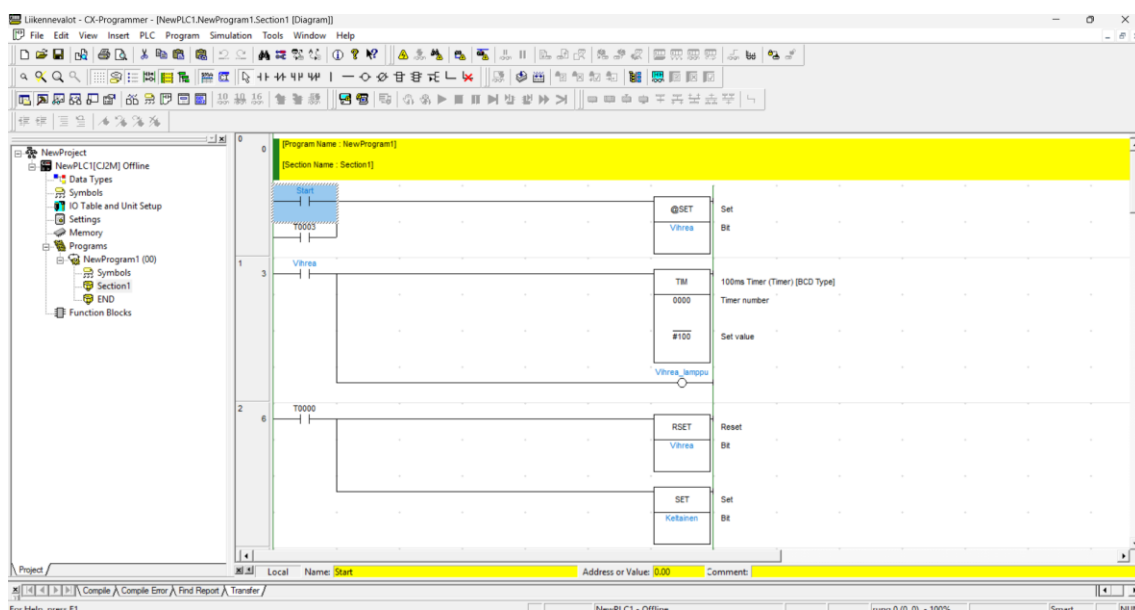


KUVA 4. Sysmac Studio.

5.2. Omron CX-Programmer

CX-Programmer on Omronin luoma ohjelmointityökalu, joka on osa CX-One ohjelmistopakettia. Se on suunniteltu erityisesti Omronin C ja CJ-sarjan PLC-logiikoille, mutta tukee myös muita Omronin logiikkaohjaimia. Ohjelmisto tukee IEC 31131-standardin mukaisia ohjelmointikieliä, ja ohjelmien teko on helpotettu drag & drop-toiminnolla (Omron n.d).

CX-Programmerin etuina voidaan pitää ohjelman keveyttä muihin ohjelmistoihin verrattuna, sen helppoa integrointia Omronin laitteisiin ja käyttäjäystävällistä käyttöliittymää. Toisaalta sen käytön rajoittuminen lähinnä Omronin tuotteisiin tekee siitä hieman marginaalisen ohjelmointiympäristön nykyajan automaatio suunnittelussa. Kuvassa 5 on esitetty CX-Programmerin ulkoasua esimerkkiohjelmaa tehtäessä.



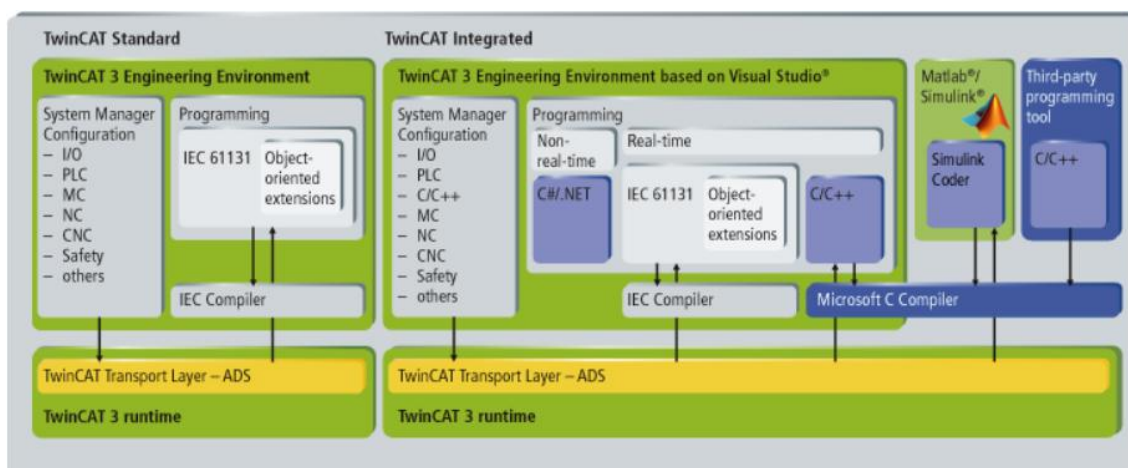
KUVA 5. CX-Programmer.

5.3. Beckhoff TwinCAT 3

TwinCAT 3 on Beckhoffin automaatiojärjestelmien ohjelmistokehityksen uusin versio. TwinCAT on lyhenne sanoista The Windows Control and Automation Technology, sen ensimmäinen versio julkaistiin jo vuonna 1996, kun taas TwinCAT 3 vuonna 2010 (Thompson 2022). Sen merkittävimpiä etuja kilpailijoihin nähden ovat yhteensopivuus yleisimpien kenttäväyläprotokollien kanssa, mikä

parantaa tuotteen käytettävyyttä huomattavasti, sekä ilmainen lisenssi suurimpaan osaan sen ominaisuuksista. TwinCAT 3 koostuu kahdesta pääosasta: XAE (eXtended Automation Engineering) ja XAR (eXtended Automation Runtime).

TwinCAT 3 XAE on automaatio-ohjelmistojen suunnitteluun tarkoitettu ohjelmisto, joka on integroitu Microsoft Visual Studioon. Tämä antaa käyttäjälle erittäin monipuolisen valikoiman erilaisia ohjelmointitapoja. Jos käyttäjällä ei ole Visual Studion täysversiota asennettuna, on hänellä käytössään kaikki IEC 61131-standardin ohjelmointikieliet. Visual Studion täysversion asennuksen jälkeen ohjelmistoympäristö laajenee entisestään, mahdollistaen myös Matlab Simulink- tai C/C++- ohjelmointiympäristöjen käytön automaatio-ohjelmistonsa tekemiseen, kuten kuvassa 6 on esitetty.

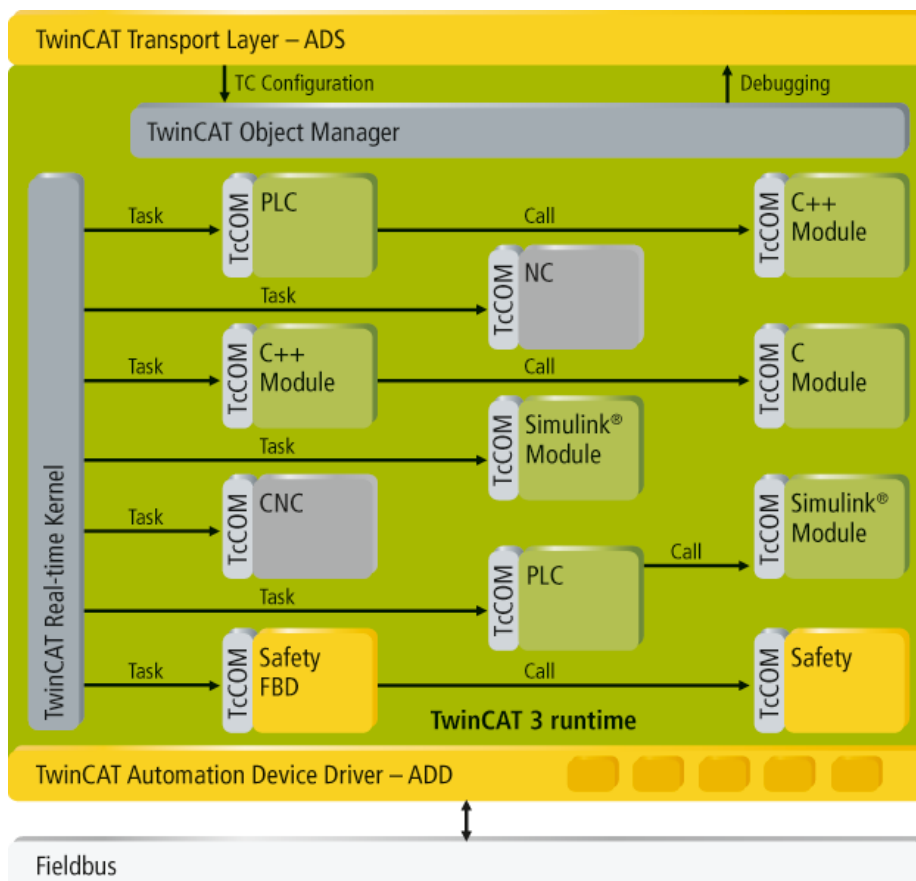


KUVA 6. TwinCAT 3:n ja Visual Studion integraatio (Beckhoff n.d).

Muita TwinCAT 3 XAE:n etuja on muun muassa sen reaaliaikaisuus ja suorituskyky. Järjestelmä pystyy hyödyntämään moniydinprosessoreita, mikä takaa sille tehokkaan suoritus- ja laskentakyvyn vaativissakin sovelluksissa. Lisäksi sen kyky integroitua IoT- ja teollisuus 4.0-järjestelmiin (Thompson 2022) tekee TwinCAT 3:sta erinomaisen työkalun nykyaikaisten automaatiojärjestelmien suunnitteluun.

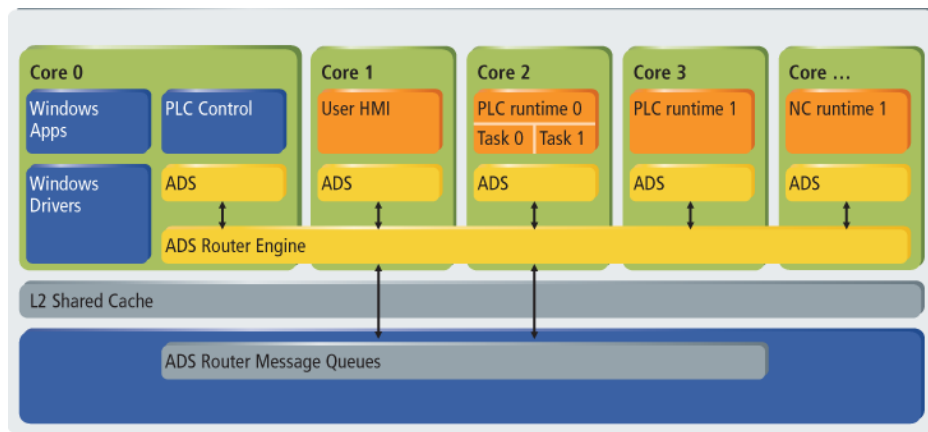
TwinCAT 3 XAR on reaaliaikainen ympäristö, johon TwinCAT-moduuleja voidaan ladata, pyörittää ja hallinnoida. Moduulit voivat olla eri ohjelmointikielillä kehitettyjä, ja niiden suoritusjärjestystä sekä -frekvenssiä voidaan säätää vapaasti.

Tämä tarjoaa merkittävää joustavuutta järjestelmän suunnitteluun ja toteutukseen. Kuvassa 7 on esitetty XAR:n arkkitehtuuria ja eri komponenttien vuorovaiutus. Kuvassa ADS on Beckhoffin kehittämä protokolla, joka mahdollistaa tiedonsiirron TwinCAT-komponenttien välillä. TwinCAT Object Manager taas on järjestelmän osa, joka hallinnoi ja koordinoi eri TwinCAT-komponenttien toimintaa, ja TwinCAT Real-Time Kernel vastaa tehtävien (Task) suorittamisesta deterministisellä ajoituksella.

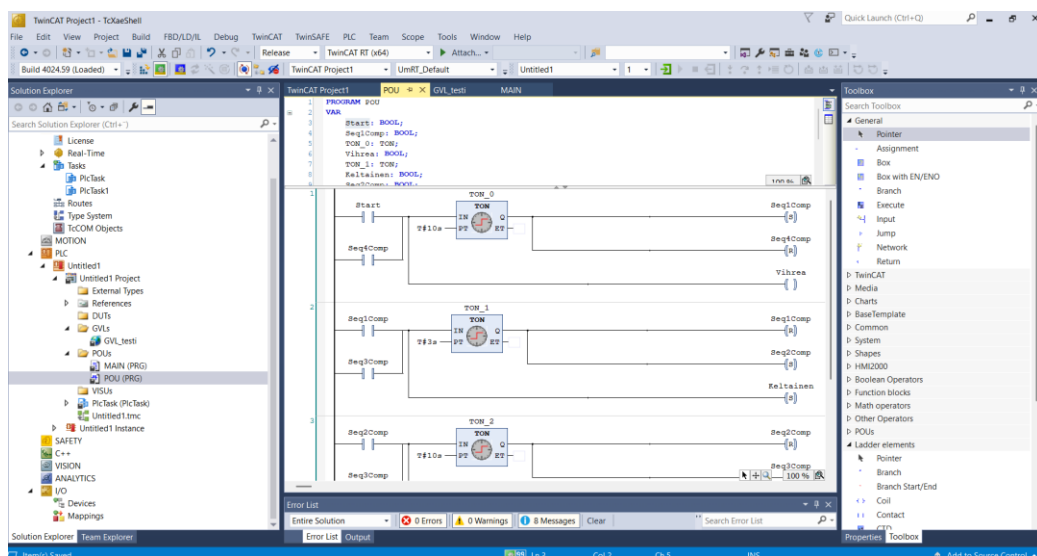


KUVA 7. TwinCAT 3 XAR arkkitehtuuri (Beckhoff n.d).

Lisäksi XAR:n vahvuuksiin kuuluu tietokoneen prosessiydinten tehokas hyödyntäminen. ADS-protokollan ansiosta moniydinprosessorin eri ytimet voidaan kohdistaa suorittamaan ohjelman eri osia, mikä mahdollistaa järjestelmän optimaalisen suorituskyvyn ja resurssien hallinnan. Kuvassa 8 havainnollistetaan, kuinka useat suorittavat komponentit jakautuvat eri prosessoriytimille järjestelmän tehokkuuden maksimoimiseksi. Kuvassa 9 on nähtävissä TwinCAT3:n ulkoasu esimerkkiohjelmassa tehtäessä.



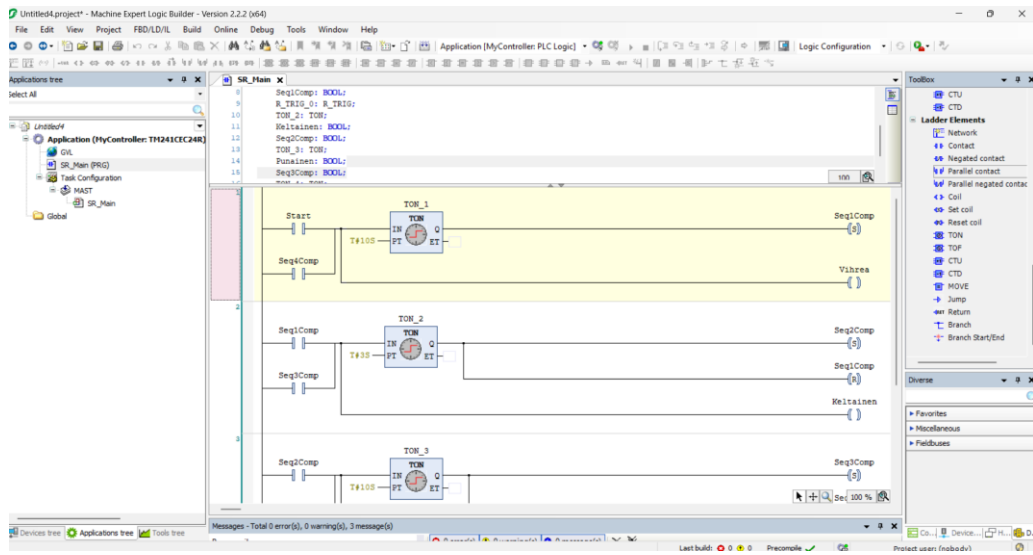
KUVA 8. Usean ytimen hyödyntäminen järjestelmässä (Beckhoff n.d).



KUVA 9. TwinCAT 3 käyttöliittymä.

5.4. Schneider Machine Expert

Machine Expert on ohjelmointityökalu, joka on suunniteltu erityisesti Schneider Electricin automaatiojärjestelmille. Se tarjoaa integroituja ratkaisuja koneiden ja tuotantolinjojen ohjelmointiin ja optimointiin, ja onkin tehokas työkalu Schneiderin laitteiden ohjaukseen ja konfigurointiin (Schneider n.d). Se tukee IEC61131-standardin mukaisia ohjelmointikieliä. Kuvassa 10 on nähtävissä Machine Expertin ulkoasu esimerkiohjelmassa tehtäessä. Kuten huomaamme, ulkoasu on hyvin samankaltainen edellä nähdyn TwinCAT3:n kanssa.



KUVA 10. Machine Expert.

6 DOKUMENTAATIO

6.1. Työssä tehtävien dokumentaatioiden lähtökohdat ja tavoitteet

JIS-Automation Oy:lla on käytössään seitsemän eri logiikkaohjelmointialustaa. Koska työntekijöiden tehtävät vaihtelevat monipuolisesti, voi syntyä tilanteita, joissa työntekijän on nopeasti perehdyttävä uudelleen ohjelmointialustaan, jota hän ei ole käyttänyt pitkään aikaan. Tällaisia tilanteita ovat esimerkiksi järjestelmän päivitys- ja vikatilanteet, tai projektin käyttöönotto silloin, kun alkuperäinen ohjelmistokehittäjä ei ole käytettävissä. Näitä tilanteita varten JIS-Automation halusi selkeän ja helposti ymmärrettävän dokumentaation, jonka avulla työntekijä voi nopeasti kerrata ohjelman perusominaisuudet.

Toinen tärkeä dokumentaatiolle on uuden työntekijän perehdytys. Hyvin laadittu ja selkeä dokumentaatio lyhentää merkittävästi uuden työntekijän perehdytykseen kuluvaan aikaan ja vähentää perehdyttäjän työmäärää. Dokumentaation lähtökohtana oli, ettei sellaista ollut aiemmin saatavilla.

Työn tavoitteena oli laatia dokumentaatio neljälle yrityksen käytetyimmälle logiikkaohjelmointialustalle:

- TwinCAT 3:lle
- Schneider Machine Expertille
- Omron Sysmac Studiolle ja CX-Programmerille.

Kolme harvemmin käytettyä ohjelmistovalmistajan, eli Wagon, Crouzetin ja Siemensin ohjelmointialustat jätettiin opinnäytetyön ulkopuolelle, sillä niiden sisällyttäminen olisi venyttänyt aikataulua merkittävästi ja lisännyt työmäärää liikaa.

6.2. Dokumentaatioiden rakenne

Dokumentaatioiden suunnittelu aloitettiin joulukuussa 2024. Lähtökohtana oli, ettei dokumentaatioiden tarkoitus ole opettaa käyttäjälle ohjelmointia tai logiikkaohjelman konfigurointia. Nämä aiheet rajattiin pois, koska logiikkaohjelmoinnin perusteet ovat laaja ja abstrakti kokonaisuus, jonka käsittely olisi tehnyt dokumentaatiosta liian laajan ja vaikeaselkoisen. Lisäksi jokaisella valmistajalla on

omat dokumentaationsa, jotka käsittelevät ohjelmointiin ja komponentteihin liittyviä yksityiskohtia (Puska 2024).

Dokumentaatioihin päätettiin sisällyttää alustavasti seuraavat osa-alueet:

- Ohjelman asennustiedoston sijainti ja lisenssikäytäntö
- Projektin luonti
- Ohjelman luonti
- Online-edit
- Laitteiden lisäys ja skannaus
- Ohjelman lataus laitteesta ohjelmaan ja ohjelmasta laitteeseen
- Teknisen tuen yhteystiedot

Yllä oleviin osa-alueisiin päädyttiin miettimällä tarkasti tilaajan työntekijöille vastaan tulevia tilanteita, joissa dokumentaatiota tarvitaan. Tärkeimmäksi tilanteeksi määriteltiin muutoksen teko valmiiseen ohjelmaan, varsinkin kiireisessä tilanteessa. Tässä skenaariossa erityisesti ohjelman lataus sekä online-edit ovat äärimmäisen tärkeitä ominaisuuksia omaksua. Lisäksi päätettiin tallentaa jokaisen ohjelmistoalustan valmistajan laatima käyttöopas JIS-Automationin pilvipalveluun. Dokumentaatioon lisätään verkkosijainti, josta oppaat löytyvät.

7 TYÖN SUORITUS

Dokumentaatiot laadittiin tammi- ja helmikuussa 2024. Koska eri ohjelmointialustat luonnostaan eroavat toisistaan, myös niiden dokumentaatiot muotoutuivat hie-
man erilaisiksi.

7.1. Aloitus ja haastattelut

Työ aloitettiin perehtymällä jokaiseen ohjelmointialustaan, joille dokumentaatioita laadittiin. Samalla teetettiin anonymi kysely yrityksen ohjelmoijille. Kyselyssä kartoitettiin mahdollisia ongelmakohtia ohjelmistoissa sekä asioita, joita dokumentaatioissa tulisi erityisesti käsitellä.

Kysely toteutettiin lähettämällä sähköpostitse kuudelle logiikkaohjelmointia teke-
välle työntekijälle Word-tiedosto, johon he saattoivat anonymisti kirjata vastauk-
sensa. Word-tiedoston saatteena olevassa viestissä esiteltiin kysely, sen tekijä ja
syy miksi kysely teetetään. Kyselyn toteutuksessa huomioitiin eettiset näkökul-
mat, kuten vastaajien anonymiteetti ja vapaaehtoisuus, jotta saatu palaute olisi
mahdollisimman rehellistä ja luotettavaa. Ehdotuksia saatiin 15 kappaletta, kyse-
lyn tulokset on esitelty kuvassa 11.

Beckhoff TwinCAT 3:

- Projektin luonti
- GVL:n ja muuttujien luonti sekä linkitys I/O-korttiin
- POU:n luonti ja sen kutsuminen pääohjelmassa
- I/O (Laitteiden skannaus ja lisäys manuaalisesti)
- Siirtyminen run-modeen sekä config-modeen
- Online-muutosten teko ja target systemin etsintä

Omron Sysmac Studio:

- Laitteen lisäys "rackiin"
- Muuttujan linkitys I/O-tuloon/lähtöön
- Asetukset
- Ohjelman luonti

Omron CX-programmer:

- I/O-korttien ja muiden laitteiden lisääminen
- Ohjelman luonti

Schneider Machine Expert:

- Ohjelman luonti
- Ohjelman lataaminen PLC:lle/PLC:itä
- Online-muutosten teko

KUVA 11. Ohjelmoijille lähetetyn kyselyn tulokset.

7.2. Dokumentaatioiden rakenne

Korpelan (1996) mukaan ohjeiden jäsentämisessä on tärkeää esittää toiminnot ajallisessa järjestyksessä. Tämän periaatteen mukaisesti dokumentaatioiden rakenne pyrittiin suunnittelemaan niin, että aiheet etenevät loogisessa järjestyksessä, vastaamaan työntekijän kohtaamia vaiheita projektin aikana. Tietoturvasyistä dokumentaatioihin ei tässä vaiheessa sisällytetty asennustiedostojen tai ohjeistuksen verkkoasemien osoitteita.

Ensimmäisenä laadittiin dokumentaatio TwinCAT 3-ohjelmointiympäristöön, valmiin dokumentaation sisällysluettelo on esitetty kuvassa 12. Seuraavaksi dokumentaatio tehtiin Omron CX-Programmeriin (kuva 13), sitten Omron Sysmac Studioon (kuva 14), ja lopuksi Schneider Machine Expertiin, jonka sisällysluettelo on nähtävissä kuvassa 15.

Sisällys

1. Ohjelman asennus ja uuden projektin luominen	3
1.1 Uuden TwinCAT-projektin luominen	3
1.2 Uuden PLC-projektin luominen	3
1.3 Uuden PLC-ohjelman luominen	4
2. Laitteiston lisäys ja tulojen linkitys muuttujiin	5
2.1 Laitteiston lisääminen skannaamalla	6
2.2 Laitteiston lisääminen käsin	6
2.3 Tulojen/lähtöjen linkitys muuttujiin	7
3. Ohjelman simulointi	8
3.1 Simulointi Windows 10 käyttöjärjestelmällä	8
3.2 Simulointi Windows 11 käyttöjärjestelmällä	9
4. Ohjelman lataaminen logiikalle/logiikalta ja sen muuttaminen	10
4.1 Ohjelman lataaminen logiikalle	10
4.2 Ohjelman lataaminen logiikalta TwinCAT:iin	11
4.3 Online-muutosten tekeminen	12
4.4 Toimenpiteet ohjelman muokkaamisen jälkeen	13
5. Tekninen tuki ja ohjekirja	13

KUVA 12. TwinCAT 3-dokumentaation sisällysluettelo.

Sisällys

1. Ohjelman asennus, uuden projektin luonti, uuden ohjelman luonti ja laitteiden lisäys	3
1.1. Projektin luonti	3
1.2. Ohjelman luonti	4
1.3. Laitteiden lisäys	5
2. Ohjelman lataus PLC:lle/PLC:ltä sekä online edit	6
2.1 Ohjelman lataus PLC:lle/PLC:ltä	6
2.2 Online edit	6
2.3 Toimenpiteet ohjelman muokkaamisen jälkeen	7
3. Tekninen tuki ja ohjekirja	7

KUVA 13. Omron CX-Programmer- dokumentaation sisällysluettelo.

Sisällys

1. Ohjelman luominen, projektin luominen, lataaminen PLC:lle ja online edit	3
1.1 Uuden projektin luominen	3
1.2 Projektin lataus PLC:ltä	3
1.3 Projektin lataaminen PLC:lle	4
1.4 Online edit	5
2. Ohjelman luominen, laitteiston lisäys sekä muuttujien lisääminen ja linkittäminen I/O:hon	5
2.1 Ohjelman luominen	6
2.2 Laitteiston lisäys	6
2.3 Muuttujien lisääminen ja linkittäminen I/O:hon	8
2.4 Toimenpiteet ohjelman muokkaamisen jälkeen	10
3. Tekninen tuki ja ohjekirja	10

KUVA 14. Omron Sysmac Studio-dokumentaation sisällysluettelo.

Sisällys

1.	Ohjelman asennus ja uuden projektin luominen	3
1.1	Uuden projektin luominen	3
1.2	Uuden PLC-ohjelman luominen	3
2.	Laitteiston lisäys ja tulojen linkitys muuttujiin	6
2.1	Uuden projektin luominen	7
2.2	Tulojen/lähtöjen linkitys muuttujiin	7
3.	Ohjelman lataus logiikalle/logiikalta ja online-muutosten teko	10
3.1	Ohjelman lataus logiikalle/logiikalta	10
3.2	Online-muutosten teko	10
3.3	Toimenpiteet ohjelman muokkaamisen jälkeen	11
4.	Tekninen tuki ja ohjekirja	11

KUVA 15. Schneider Machine Expert-dokumentaation sisällysluettelo.

Dokumentaatiot löytyvät kokonaisuutenaan opinnäytetyön liitteinä, TwinCAT 3:n dokumentaatio on liitteenä 1, Omron CX-Programmerin liitteenä 2, Sysmac Studio liitteenä 3 ja Schneider Machine expertin dokumentaatio liitteenä 4.

7.3. Dokumentaatioiden eroavaisuudet ja arviointi

Dokumentaatioiden perusrakenne on pitkälti samanlainen, ja myös sivumäärä asettuu 10–13 sivun välille. Poikkeuksena CX-programmer-dokumentaatio, jonka sivumäärä on hieman pienempi, sillä ohjelma on kevyempi verrattuna muihin. Kunkin ohjelmistoalustan dokumentaatioissa pyrittiin käsittelemään alkutiedoissa ja kyselyvaiheessa esiin nousseet seikat ja ominaisuudet. TwinCAT 3 erottuu dokumentaatioista selkeimmin. Dokumentaatiota laadittaessa kävi ilmi, että TwinCATin runtime-moduuli, joka mahdollistaa muun muassa ohjelman simuloinnin, ei ole suoraan yhteensopiva Windows 11 kanssa. Tämän vuoksi dokumentaatioon oli tehtävä oma lukunsa simuloinnille.

Dokumentaatiot käytiin valmistumisen jälkeen läpi tilaajan edustajan kanssa. Arviointikeskustelussa ei havaittu puutteita tai kehitystarpeita, vaan tilaaja oli erittäin tyytyväinen lopputulokseen. Dokumentaatiot luovutettiin tilaajan käyttöön, ja niitä saatetaan jatkossa laajentaa kattamaan myös ohjelmointi, jotta yritykselle voidaan luoda selkeät suuntaviivat ohjelmointityylille.

8 POHDINTA

Opinnäytetyön tavoitteena oli laatia kattava ja selkeä dokumentaatio JIS-Automation Oy:n käyttämiin logiikkaohjelmointialustoihin, jotta sekä uudet että kokeneemmat työntekijät voivat helposti perehtyä ohjelmistojen perusominaisuuksiin. Työssä kiinnitettiin erityistä huomiota dokumentaation rakenteeseen, selkeyteen ja visuaaliseen esitystapaan.

Yrityksellä ei aiemmin ollut käytössä kattavaa dokumentaatiota käyttämistään logiikkaohjelmointialustoista, mikä teki perehtymisestä ja aiempien projektien pariin palaamisesta haastavaa. Tämän vuoksi dokumentaation laatiminen koettiin tärkeäksi, ja sen odotetaan helpottavan alustojen parissa työskentelyä jatkossa merkittävästi.

Dokumentaation laatimisessa hyödynnettiin ohjelmistojen ohjekirjoja, ohjelmoijille teetetyn kyselyn tuloksia sekä omia kokemuksia ohjelmointialustoihin perehtymisestä. Lisäksi jatkuva vuoropuhelu tilaajan edustajan kanssa auttoi hahmottamaan dokumentaation laajuuden ja asiasisällön, varmistaen, että se vastasi yrityksen tarpeita ja käytännön työssä esiintyviä haasteita.

Kokonaisuudessaan opinnäytetyö tarjosi arvokasta kokemusta teknisen dokumentaation laatimisesta sekä ohjelmistoalustojen perusominaisuuksien ymmärtämisestä. Dokumentaation vastaanotto tilaajan taholta oli erittäin positiivinen, mikä osoittaa työn tarpeellisuuden ja hyödyn yritykselle. Jatkossa dokumentaation ylläpito ja päivittäminen ovat tärkeitä, jotta ne pysyvät ajan tasalla ja palvelevat käyttäjiä parhaalla mahdollisella tavalla. Lisäksi dokumentaatioiden laajentaminen koskemaan esimerkiksi ohjelmoinnin perusteita on helppoa hyvän pohjatyön ansiosta.

LÄHTEET

Beckhoff Automation. n.d. TwinCAT 3 Philosophy. Verkkosivu. Viitattu 24.2.2025. <https://infosys.beckhoff.com/>

Hansen, D.H. 2015. Programmable logic controllers: a practical approach to IEC 61131-3 using CODESYS. Chichester, England: Wiley.

Heikkinen, S. 2024. Opittavuuden merkitys käyttäjäkokemuksessa. LabOpen 28.10.2024. Viitattu 23.2.2025. <https://www.labopen.fi/lab-pro/opittavuuden-merkitys-kayttajakokemuksessa/>

JIS-automation Oy. n.d. JIS. Verkkosivu. Viitattu 20.12.2024. <http://www.jis.fi>

John, K. & Tiegelkamp, M. 2010. IEC 61131-3: Programming Industrial Automation Systems. Berlin, Germany: Springer.

Kallio, E. 2020. Development of adult thinking: interdisciplinary perspectives on cognitive development and adult learning. London, England: Routledge.

Korpela, J. 1996. Ohjeen kirjoittaminen. Verkkosivu. Viitattu 25.2.2025. <https://jkorpela.fi/kirj/7.7.html>

Omron Automation. n.d. CX-Programmer Programming and debugging your PLC has never been easier. Verkkosivu. Viitattu 24.2.2025. <https://industrial.omron.fi/fi/products/cx-programmer>

Omron Automation. n.d. Sysmac-automaatioympäristö. Verkkosivu. Viitattu 24.2.2025. <https://industrial.omron.fi/fi/products/sysmac-platform#why-sysmac>

Puska, J. osastopäällikkö. Sähköpostiviesti 16.12.2024.

Päivänsalo, T-M. 2020. Oppimiskoodi: Kuinka oppiminen onnistuu. Jyväskylä: PS-kustannus.

Schneider Electric. n.d. What is EcoStructure Machine Expert. Verkkosivu. Viitattu 24.2.2025. https://product-help.schneider-electric.com/Machine%20Expert/V1.1/en/SoGloRef/SoGloRef/Generals_2/Generals_2-2.htm

Thompson, D. 2022. TwinCAT Automation Software from Beckhoff Turns 25. Beckhoff 24.1.2022. Viitattu 24.2.2025. <https://www.blog.beckhoff-us.com/post/twincat-software-turns-25>

White, M.T. 2023. Mastering PLC Programming. Birmingham, England: Packt Publishing.

White, M.T. 2024. PLCs for Beginners: An Introductory Guide to Building Robust PLC Programs with the Structured Text Language. Birmingham, England: Packt Publishing.

LIITTEET

Liite 1. TwinCAT 3-dokumentaatio

1 (13)

**JIS-Automation Oy**

TwinCAT3



Sisälllys

1. Ohjelman asennus ja uuden projektin luominen	3
1.1 Uuden TwinCAT-projektin luominen	3
1.2 Uuden PLC-projektin luominen	4
1.3 Uuden PLC-ohjelman luominen	5
2. Laitteiston lisäys ja tulojen linkitys muuttujiin	6
2.1 Laitteiston lisääminen skannaamalla	6
2.2 Laitteiston lisääminen käsin	7
2.3 Tulojen/lähtöjen linkitys muuttujiin	7
3. Ohjelman simulointi	8
3.1 Simulointi Windows 10 käyttöjärjestelmällä	9
3.2 Simulointi Windows 11 käyttöjärjestelmällä	9
4. Ohjelman lataaminen logiikalle/logiikalta ja sen muuttaminen	10
4.1 Ohjelman lataaminen logiikalle	10
4.2 Ohjelman lataaminen logiikalta TwinCAT:iin	11
4.3 Online-muutosten tekeminen	12
4.4 Toimenpiteet ohjelman muokkaamisen jälkeen	13
5. Tekninen tuki ja ohjekirja	13

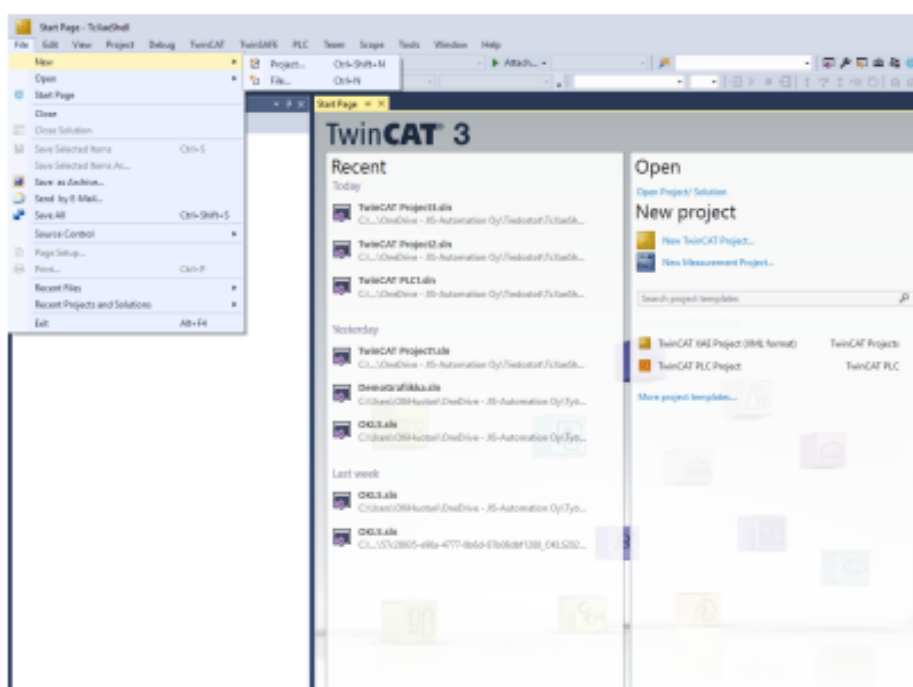


1. Ohjelman asennus ja uuden projektin luominen

TwinCAT3:n, ja muiden käytössä olevien Beckhoff-tuotteiden asennustiedostot löytyvät pilvipalvelusta.

1.1 Uuden TwinCAT-projektin luominen

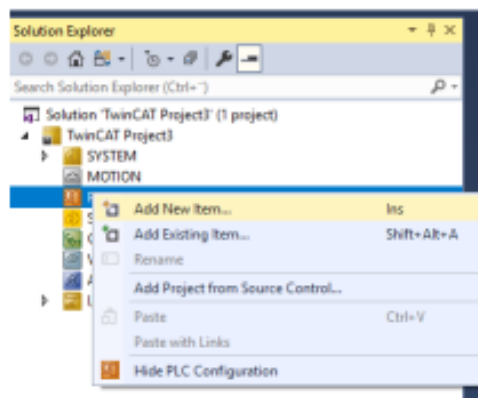
Uusi projekti luodaan TwinCAT3:ssa valitsemalla ohjelman käynnistyksen jälkeen New TwinCAT Project, valitsemalla tehtäväpalkista File --> New --> Project, tai painamalla Ctrl+Shift+N. Tämän jälkeen valitaan TwinCAT XAE Project, ja ohjelma luo uuden projektin.



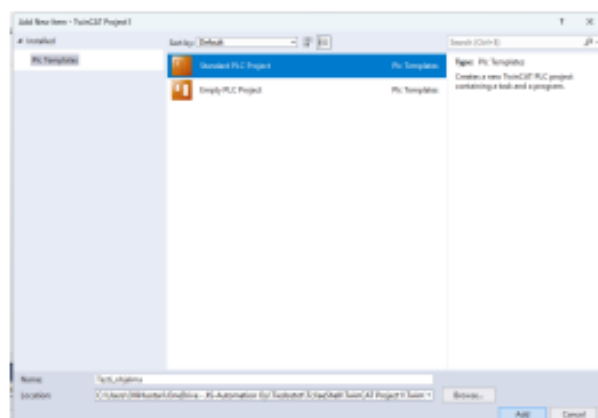


1.2 Uuden PLC-projektin luominen

Jotta voidaan aloittaa ohjelmointi, täytyy TwinCAT-projektiin lisätä vielä PLC-projekti. Tämä tapahtuu painamalla hiiren oikealla näppäimellä PLC-kohtaa projektin puunäkymässä ja valitsemalla "Add New Item".



Avautuvasta valikosta valitaan Standard PLC Project, projekti nimetään ja painetaan "Add", jonka jälkeen ohjelma lisää PLC-projektin puunäkymään.





1.3 Uuden PLC-ohjelman luominen

Uusi PLC-ohjelma luodaan painamalla hiiren oikealla näppäimellä projektipuusta PLC-->POUs ja valitsemalla "Add" ja "POU". Avautuvasta valikosta valitaan, luodaanko ohjelma vai Function Block sekä ohjelmointikieli.



Kun PLC-ohjelma on tehty, täytyy ohjelmaa kutsua vielä pääohjelmassa (MAIN). Esimerkiksi "Testi"-nimistä ohjelmaa kutsutaan pääohjelmassa kirjoittamalla ohjelmakutsuun Testi();. Jos tämän jälkeen tulee virheilmoitus "Functionblock Testi must be instantiated to be accessed" ohjelmaa suorittaessa täytyy Function Block instansoida esimerkiksi alla olevan kuvan mukaisesti.

```

MAIN*
1 PROGRAM MAIN
2 VAR
3     Testi : Testi;
4 END_VAR
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```



2. Laitteiston lisäys ja tulojen linkitys muuttujiin

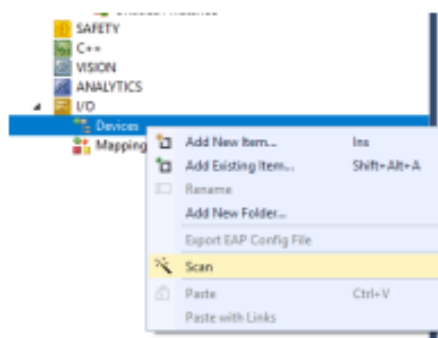
Käytettävä laitteisto voidaan lisätä TwinCAT-projektiin joko käsin tai skannaamalla. Jos mahdollista, kannattaa näistä vaihtoehdoista suosia skannaamista, sillä näin varmistut, että laitteet on oikein lisätty projektiin.

2.1 Laitteiston lisääminen skannaamalla

Jotta laitteiden skannaus onnistuu, täytyy TwinCAT-järjestelmässä olla Config Mode (sininen nappi alla olevassa kuvassa) valittuna sekä koneen pitää olla kytkettynä fyysisesti skannattavaan laiteryhmään. Tämän jälkeen valitaan alla näkyvän kuvan mukaisesta alavetovalikosta "Choose Target System", ja valitaan vaihtoehto Search (Ethernet). PLC:n IP-osoite voidaan syöttää myös käsin, jos se on tiedossa.



Kun PLC on löydetty ja se on valittu yllä olevasta alavetovalikosta, voidaan järjestelmän laitteet skannata painamalla oikealla hiiren painikkeella projektipuusta I/O--->Devices ja valitsemalla "Scan".



Tämän jälkeen järjestelmä näyttää laitteet, jotka se löysi ja ne voidaan lisätä projektipuuhun painamalla "OK"-nappia.

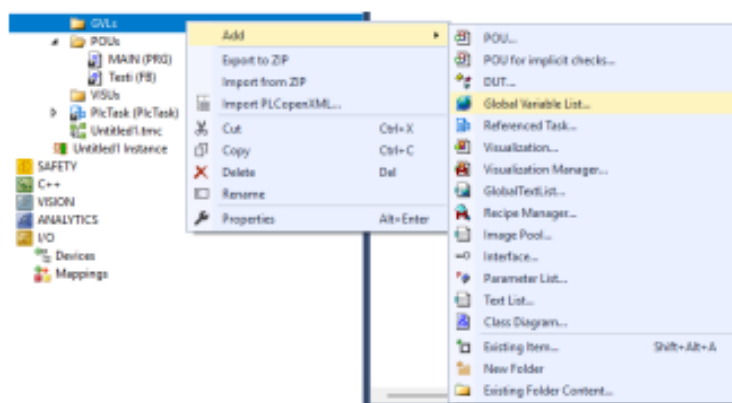


2.2 Laitteiston lisääminen käsin

Laitteiston syöttäminen käsin onnistuu painamalla hiiren oikealla näppäimellä projektipuusta I/O—> Devices ja valitsemalla "Add New Item". Tämän jälkeen täytyy ensin syöttää EtherCAT Master, tämän jälkeen Power Supply ja lopuksi muut laitteet.

2.3 Tulojen/lähtöjen linkitys muuttujiin

Projektiin saa luotua globaaleja muuttujia luomalla GLV:n (Global Variable listin) alla olevan kuvan mukaisesti.



Globaalin muuttujan voi linkittää tiettyyn tulo- tai lähtöporttiin lisäämällä muuttujan nimen ja sen tyypin väliin "AT %I*" tai "AT %Q*" riippuen onko kyseessä input vai output. Esimerkki alla.

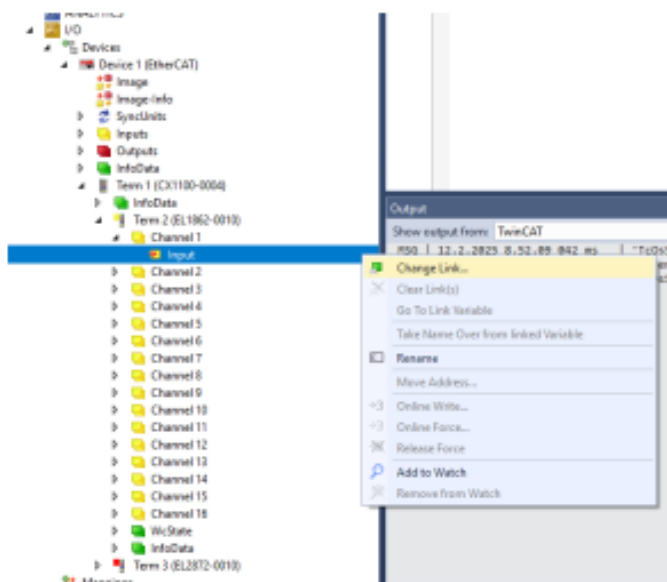
```

3VL_testi*  X  TwinCAT Project1  MAIN
1  {attribute 'qualified_only'}
2  VAR GLOBAL
3      Testi_in AT %I* : BOOL;
4      Testi_out AT %Q* : BOOL;
5  END_VAR

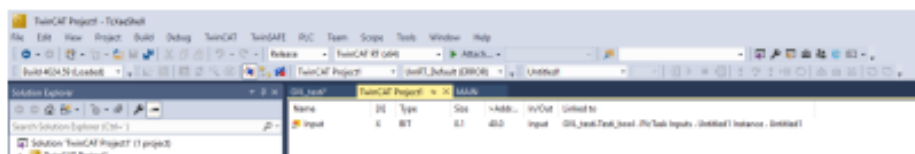
```



Tämän jälkeen muuttuja linkitetään tulo- tai lähtöporttiin painamalla haluttua porttia hiiren oikealla näppäimellä ja valitsemalla "Change link"



Avautuvasta ikkunasta valitaan haluttu muuttuja ja valitaan "OK". Nyt muuttujan pitäisi näkyä portin "Linked to"-kohdassa.



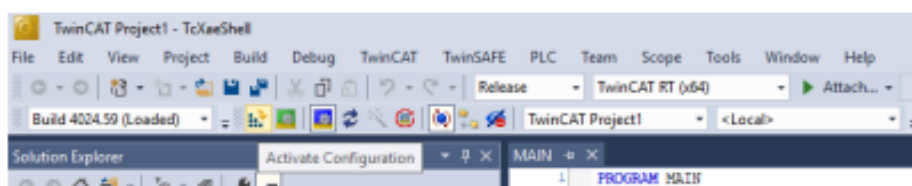
Muuttujan luonnin jälkeen ohjelma pitää "kääntää" ennen kuin se voidaan linkittää, tämä tapahtuu painamalla Ctrl+Shift+B tai valitsemalla valikkopalkista Build --> Build solution.

3. Ohjelman simulointi



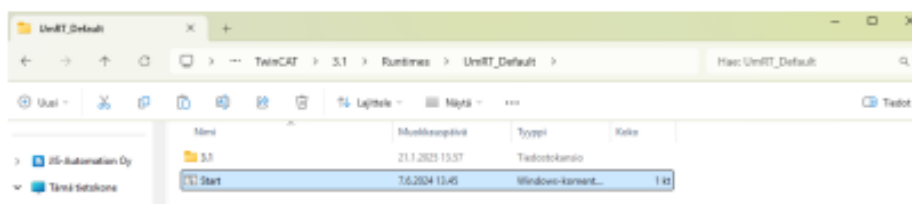
3.1 Simulointi Windows 10 käyttöjärjestelmällä

Ohjelman toimintaa voidaan tarkastella simuloimalla se TwinCAT:in runtime-sovellusta käyttäen. Ohjelman teon jälkeen valitaan Build-valikosta Build Solution, jolloin TwinCAT "kääntää" tehdyn ohjelman. Tämän jälkeen ohjelman konsoliin tulee mahdolliset virheilmoitukset, jotka on korjattava ennen kuin ohjelmaa voidaan käyttää. Tämän jälkeen valitaan target-valikosta <local>, painetaan "Activate Configuration", jos kaikki on ok, muuttuu "login"-nappi vihreäksi ja sitä painamalla saadaan aloitettua simulointi. Simulointi lopetetaan "logout"-napilla.



3.2 Simulointi Windows 11 käyttöjärjestelmällä

Windows 11 ei vielä ole virallisesti tuettu Runtimen kanssa, joten simulointi on tehtävä Usermode Runtimea käyttäen. Tätä varten on ennen simuloinnin aloittamista etsittävä TwinCAT --> 3.1 --> Runtimes --> UmRT_Default-kansio ja sieltä käynnistetään start.bat tiedosto.

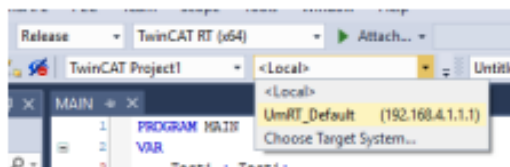


Tämän jälkeen aukeaa komentorivi, josta näemme kyseisen Usermode Runtimen IP-osoitteen:



```
C:\TwinCAT3_1\bin\user\umrt_default>C:\TwinCAT3_1\bin\user\bin\bu1d_4004_00\TcSystemService.exe -i 192.168.4.1.1.1
-> umrt_default -> "192.168.4.1.1.1"
TcSystemService started
Heap memory allocated 0000023980300000 size=00000000
TcSystemService started
Address: 192.168.4.1.1.1
TcSystemService started: canrtg
Press "c" for Reconfig TwinCAT System.
Press "r" for Restart TwinCAT system.
Press "s" to view current state.
Press "x" to exit TwinCAT system service.
```

Sama IP-osoite löytyy nyt TwinCAT-in target-valikosta:

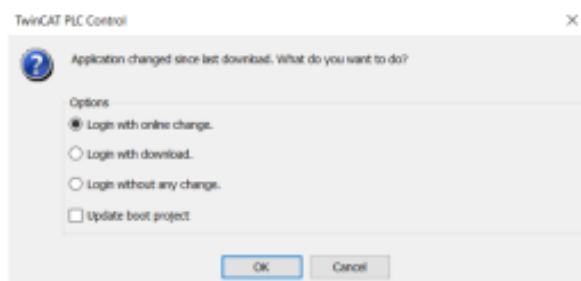


Tämän jälkeen simulointi tapahtuu samalla tavalla kuin Windows 10-käyttöjärjestelmällä.

4. Ohjelman lataaminen logiikalle/logiikalta ja sen muuttaminen

4.1 Ohjelman lataaminen logiikalle

Ohjelma ladataan logiikalle valitsemalla haluttu PLC target-valikosta, ja simuloinnin tapaan valitsemalla ensin "Build Solution" ja sen jälkeen "Activate Configuration". Tämän jälkeen, jos virheilmoituksia ei tule, ohjelma latautuu logiikalle, kun painetaan "login"-nappia. Kun kyseistä nappia painetaan, tulee ruudulle valintaikkuna:



Jos halutaan ladata muutettu ohjelma PLC:lle, valitaan ensimmäinen vaihtoehto, jos halutaan ladata ohjelma PLC:ltä tietokoneelle, valitaan toinen, ja kolmas jos halutaan kirjautua sisään tekemättä

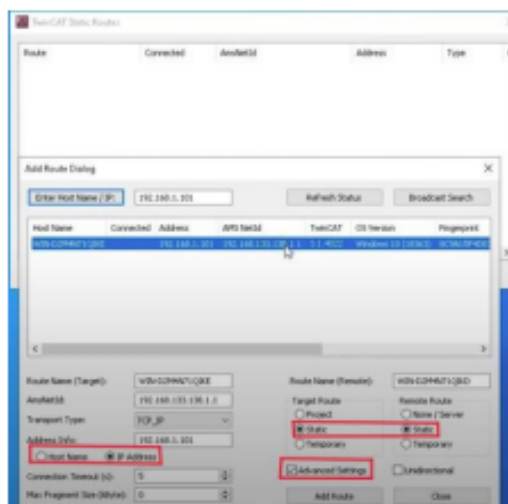


kumpaakaan edellisistä. Tässä tapauksessa siis valitaan ensimmäinen vaihtoehto. "Update boot project"-valinta jätetään valitsematta, ellei boot projectiin ole tehty muutoksia, jotka halutaan päivittää PLC:lle.

4.2 Ohjelman lataaminen logiikalta TwinCAT:iin

PLC-ohjelma voidaan ladata PLC:ltä TwinCAT:iin etenemällä edellisen kohdan mukaisesti ja valitsemalla viimeisessä kohdassa "Login with download". Jos tämä ei kuitenkaan jostain syystä onnistu, voidaan sama asia hoitaa seuraavalla tavalla:

Valitaan Windowsin "näytät piioitetut kuvakkeet"-painikkeen takaa TwinCAT Config Mode --> Router --> Edit Routes, valitaan "Add", syötetään PLC:n IP-osoite, johon on kytkeydytty ja painetaan "Enter Host Name/IP". Varmistetaan, että alla näkyvät asetukset on valittu kuvan mukaisesti:



Tämän jälkeen valitaan "Add Route", avautuvasta ikkunasta poistetaan "Secure ADS"-valinta ja kirjaudutaan PLC:lle sen administrator-tunnuksilla.





4.4 Toimenpiteet ohjelman muokkaamisen jälkeen

Kun ohjelmaa on muutettu online-editillä tai muuten, tallennetaan se toiminnallisuuden testauksen jälkeen pilvipalveluun JIS-Automation Oy --> JIS Suunnitteluosasto - Tiedostot --> Projektit-kansiosta löytyvään projektin Ohjelmointi-kansioon. Jos kansiossa on jo vanhoja ohjelmistoversioita, voidaan muokattu ohjelma tallentaa kansioon esimerkiksi "muokkauspäivämäärä_projektin nimi"-muodossa.

5. Tekninen tuki ja ohjekirja

Jos tarvitset teknistä tukea TwinCAT3:n suhteen, ota yhteyttä Beckhoffin tekniseen tukeen:

- +358 20 7423807
- support@beckhoff.fi

TwinCAT3:n ohjekirja on tallennettu pilvipalveluun.

Liite 2. Omron CX-Programmer-dokumentaatio

1 (8)

**JIS-Automation Oy**

Omron CX-Programmer



Sisällys

1. Ohjelman asennus, uuden projektin luonti, uuden ohjelman luonti ja laitteiden lisäys.....	3
1.1. Projektin luonti.....	3
1.2. Ohjelman luonti.....	4
1.3. Laitteiden lisäys.....	5
2. Ohjelman lataus PLC:lle/PLC:ltä sekä online edit.....	6
2.1 Ohjelman lataus PLC:lle/PLC:ltä.....	6
2.2 Online edit.....	6
2.3 Toimenpiteet ohjelman muokkaamisen jälkeen.....	7
3. Tekninen tuki ja ohjekirja.....	8

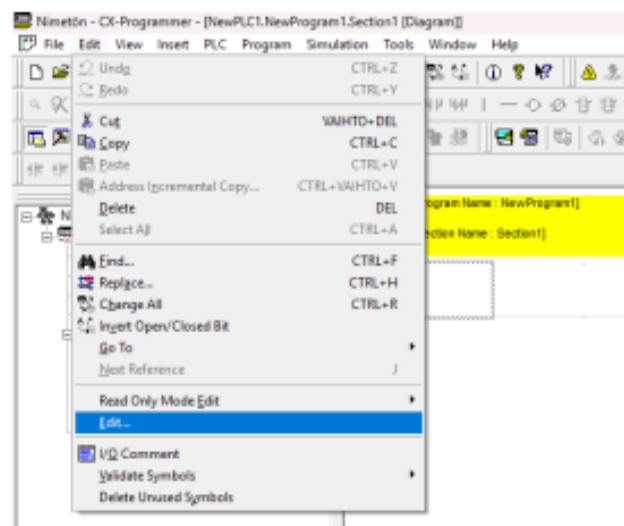


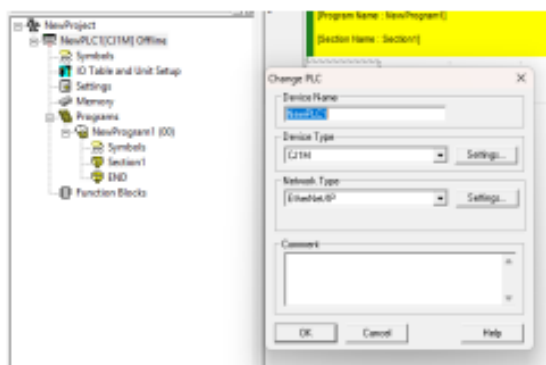
1. Ohjelman asennus, uuden projektin luonti, uuden ohjelman luonti ja laitteiden lisäys

CX-programmerin, ja muiden käytössä olevien Omron-tuotteiden asennustiedostot löytyvät pivipalvelusta. Ohjelman lisenssiavain löytyy samasta kansiosista asennustiedoston kanssa.

1.1. Projektin luonti

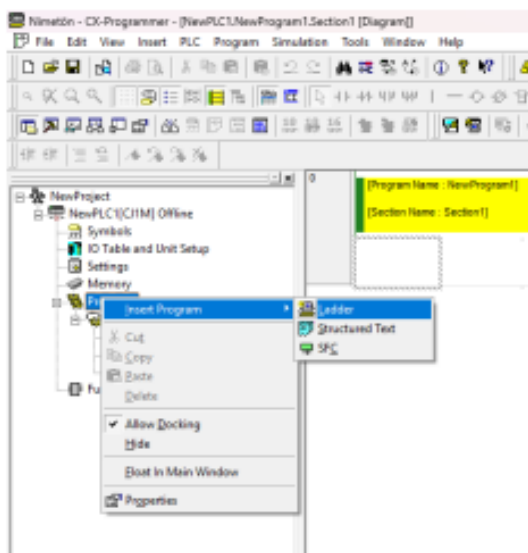
Uusi projekti luodaan valitsemalla valikkopalkista File --> New... tai painamalla Ctrl+N. Avautuvasta pop-up-ikkunasta valitaan PLC:n tyyppi, nimitään se ja valitaan tapa, jolla laitteeseen otetaan yhteys. Asetuksia voi muuttaa myös jälkikäteen valitsemalla PLC:n projektin puunäkymästä, ja valitsemalla valikkopalkista Edit --> Edit...





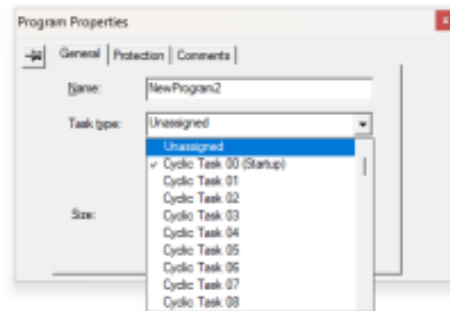
1.2. Ohjelman luonti

Projektiin luodaan uusi ohjelma painamalla projektin puunäkymästä joko PLC:tä tai "Programs"-kohtaa hiiren oikealla näppäimellä, valitsemalla "Insert program" ja valitsemalla haluttu ohjelmointikieli.



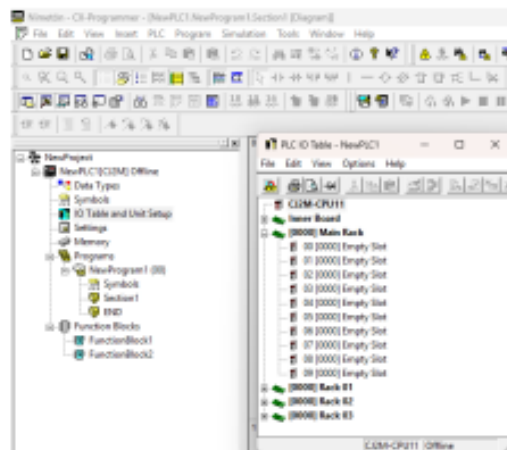


Valinnan jälkeen avautuu ikkuna, jossa ohjelma nimetään ja valitaan missä kohtaa ohjelmakiertoa ohjelma suoritetaan. HUOM! Vain yksi ohjelma voi olla merkittynä jokaiselle "cyclic task"-numerolle.



1.3. Laitteiden lisäys

CX-Programmerissa voidaan laitteita lisätä joko kytkeytymällä laitteistoon ja antaa ohjelman etsiä laitteiston osat, tai syöttämällä ne käsin. Molemmissa tapauksissa mennään projektin puunäkymässä kohtaan "IO Table and Unit Setup".

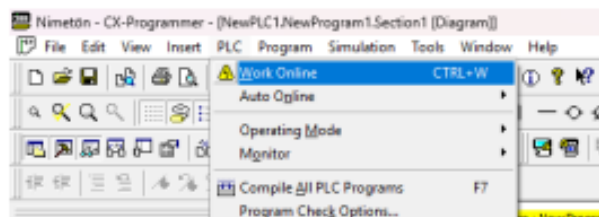




Jos on kytkeydytty laitteistoon, valitaan valikkopalkista Options -> Create, ja CX-programmer etsii laitteiston laitteen ja lisää ne listaan. Jos halutaan syöttää laitteet käsin, tuplaklikataan räkin paikkaa, johon laite halutaan lisätä ja valitaan laite listasta.

2. Ohjelman lataus PLC:lle/PLC:ltä sekä online edit

HUOM! Ohjelman pitää olla online-tilassa (Ctrl+W tai valikkopalkista PLC -> Work Online)

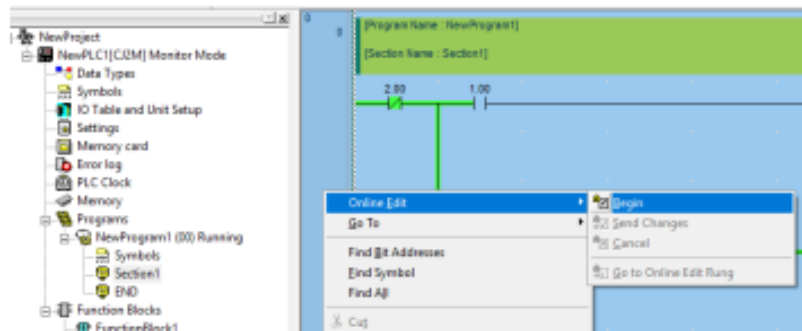


2.1 Ohjelman lataus PLC:lle/PLC:ltä

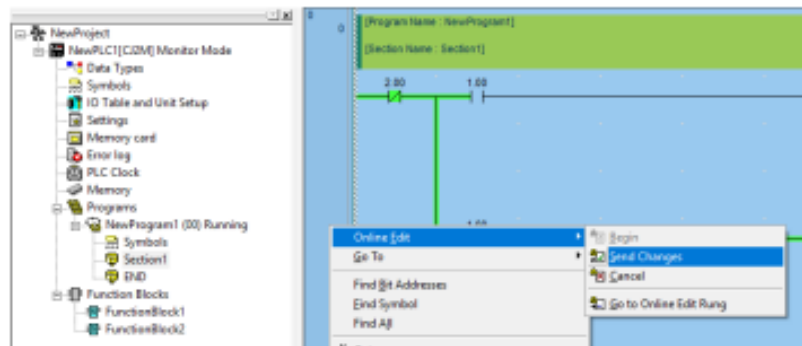
Valitse valikkopalkista PLC -> Transfer -> "To PLC" tai "From PLC", riippuen haluatko ladata ohjelman CX-programmerista PLC:lle vai päin vastoin.

2.2 Online edit

Luotua tai ladattua ohjelmaa voidaan muokata online-tilassa. Painetaan hiiren oikealla napilla sekvenssiä, jota halutaan muokata ja valitaan Online Edit -> Begin.



Kun muutokset on saatu tehtyä, valitaan samasta valikosta "Send Changes" tai "Cancel" jos halutaan perua online-muutokset. "Go to Online Edit Rung"-valinnalla pääsee hyppäämään siihen sekvenssiin, jolle online-muutosten teko on sillä hetkellä aktivoituna.



2.3 Toimenpiteet ohjelman muokkaamisen jälkeen

Kun ohjelmaa on muutettu online-editillä tai muuten, tallennetaan se toiminnallisuuden testausten jälkeen pilvipalveluun JIS-Automation Oy --> JIS Suunnitteluosasto - Tiedostot --> Projektit-kansioon löytyvään projektin Ohjelmointi-kansioon. Jos kansiossa on jo vanhoja ohjelmistoversioita, voidaan muokattu ohjelma tallentaa kansioon esimerkiksi "muokkauspäivämäärä_projektin nimi"-muodossa.



3. Tekninen tuki ja ohjekirja

Jos tarvitset teknistä tukea CX-Programmerin suhteen, ota yhteyttä Omronin tekniseen tukeen:

- Puhelin 0207 464 200
- Sähköpostiosoite: info_fi@omron.com

CX-Programmerin ohjekirja on tallennettu pilvipalveluun.

Liite 3. Omron Sysmac Studio-dokumentaatio

1 (11)

**JIS-Automation Oy**

Omron Sysmac Studio



Sisällys

1. Ohjelman luominen, projektin luominen, lataaminen PLC:lle ja online edit	3
1.1 Uuden projektin luominen	3
1.2 Projektin lataus PLC:ltä.....	4
1.3 Projektin lataaminen PLC:lle.....	4
1.4 Online edit	5
2. Ohjelman luominen, laitteiston lisäys sekä muuttujien lisääminen ja linkittäminen I/O:hon	6
2.1 Ohjelman luominen	6
2.2 Laitteiston lisäys	6
2.3 Muuttujien lisääminen ja linkittäminen I/O:hon.....	8
2.4 Toimenpiteet ohjelman muokkaamisen jälkeen	10
3. Tekninen tuki ja ohjekirja	11



1. Ohjelman luominen, projektin luominen, lataaminen PLC:lle ja online edit

CX-programmerin, ja muiden käytössä olevien Omron-tuotteiden asennustiedostot löytyvät pilvipalvelusta. Ohjelman lisenssiavain löytyy samasta kansioista asennustiedoston kanssa.

1.1 Uuden projektin luominen

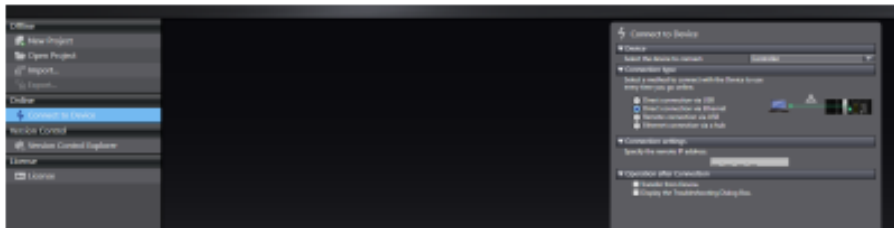
Uusi projekti luodaan Sysmac Studioissa valitsemalla New Project aloitussivulta, nimeämällä projekti, sekä valitsemalla sen PLC:n tyyppi jolle ohjelmaa ollaan tekemässä Select Device valikosta. PLC:n tyyppiä voi vaihtaa myös projektin luomisen jälkeen. Tämän jälkeen valitaan Create, ja Sysmac Studio luo uuden projektin. Jos haluat emuloida tarkemmin I/O-laitteiden sekä PLC:n toimintaa, valitse sivun vasemmasta alareunasta "Open in Emulation Mode"-valinta.





1.2 Projektin lataus PLC:itä

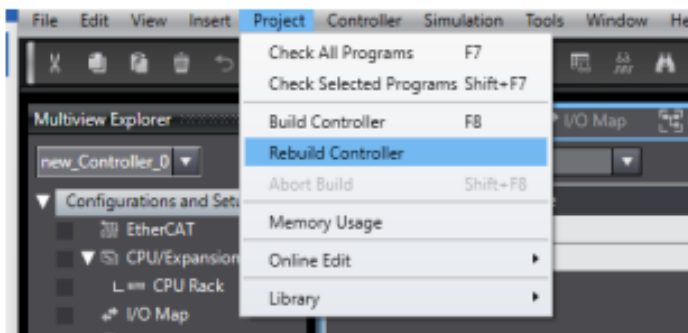
Jos halutaan ladata ohjelma PLC:itä esimerkiksi sen muokkaamista varten, valitaan aloitusvalikosta "Connect to Device", valitaan Controller ja haluttu yhteysvaihtoehto (yleensä Direct connection via Ethernet tai Ethernet connection via a hub). Tämän jälkeen laitetaan kohdelaitteen IP-osoite, Sysmac Studioon pitäisi löytää kohdelaite ilman IP-osoitteen syöttämistäkin, ja lopuksi valitaan Connect.



Kun edellinen vaihe on suoritettu onnistuneesti, valitaan valikkopalkista Controller --> Transfer --> From Controller ja tämän jälkeen pop-up-valikosta "Execute". HUOM! Latauksen jälkeen järjestelmä on runmodossa.

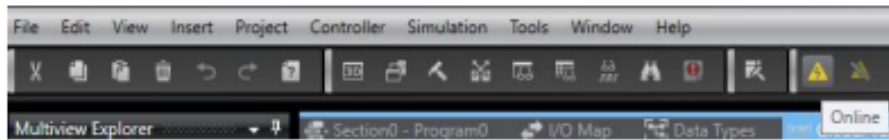
1.3 Projektin lataaminen PLC:lle

Valmis projekti saadaan ladattua PLC:lle valitsemalla valikkopalkista Project --> Build Controller tai Rebuild Controller.



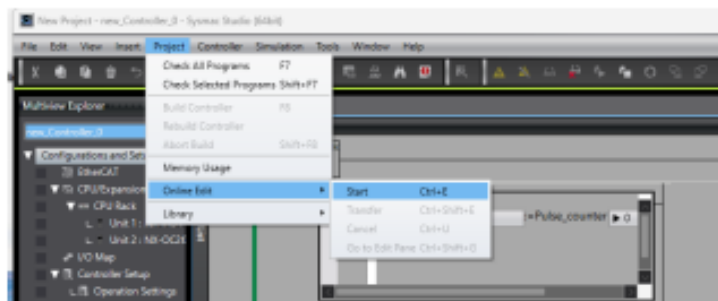


Tämän jälkeen mennään online-modeen, ja valikkopalkista Controller --> Transfer --> To Controller...

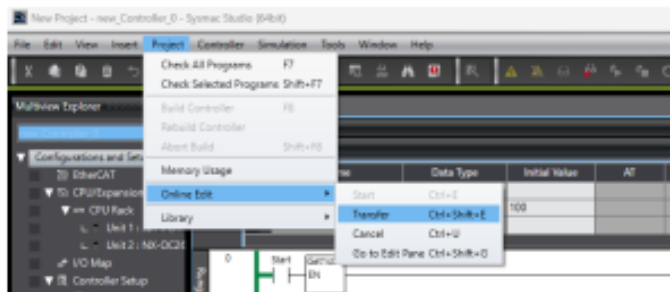


1.4 Online edit

Jos latauksen jälkeen halutaan tehdä ohjelmalle muutoksia online-tilassa (järjestelmää ei pysäytetä muutosten aikana), valitaan valikkopalkista Project --> Online edit --> Start tai painetaan Ctrl+E.



Tämän jälkeen ohjelmalle tehdään tarvittavat muutokset ja valitaan samasta valikosta "Transfer", tai painetaan Ctrl+Shift+E. Valitaan latauksen aikana tulevasta pop-up-ikkunasta "Yes", ja Sysmac Studio palaa online-tarkastelutilaan latauksen jälkeen.

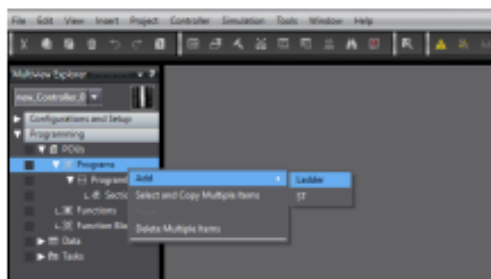




2. Ohjelman luominen, laitteiston lisäys sekä muuttujien lisääminen ja linkittäminen I/O:hon

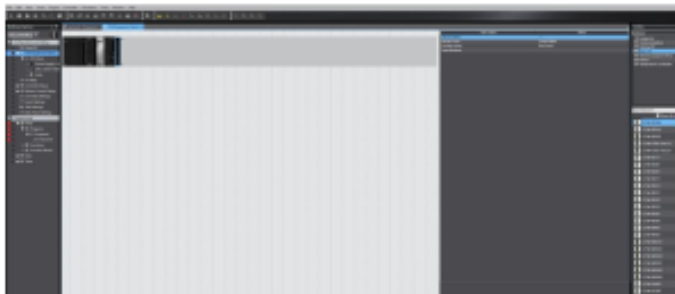
2.1 Ohjelman luominen

Kun halutaan luoda uusi ohjelma luotuaan projektiin, valitaan projektin puunäkymästä Programming --> POUs, painetaan hiiren oikealla näppäimellä "Programs" ja valitaan Add --> Ladder tai ST, riippuen kumpaa ohjelmointitapaa haluaa käyttää.



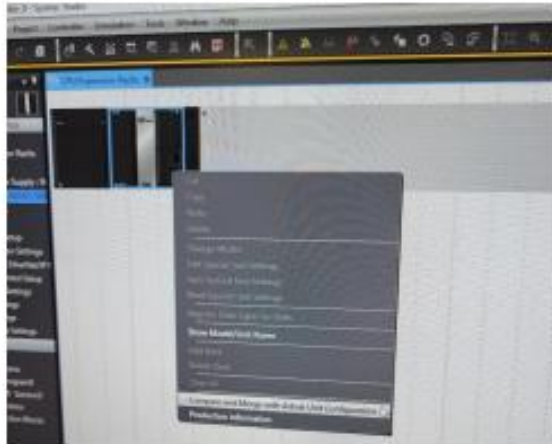
2.2 Laitteiston lisäys

Jos laitteistoa halutaan syöttää käsin, valitaan projektin puunäkymästä Configurations and Setup ja tuplakkikataan CPU/Expansion Racks. Tämän jälkeen lisätään halutut laitteet "räkkiin" sivun oikeassa laidassa olevista valikoista.

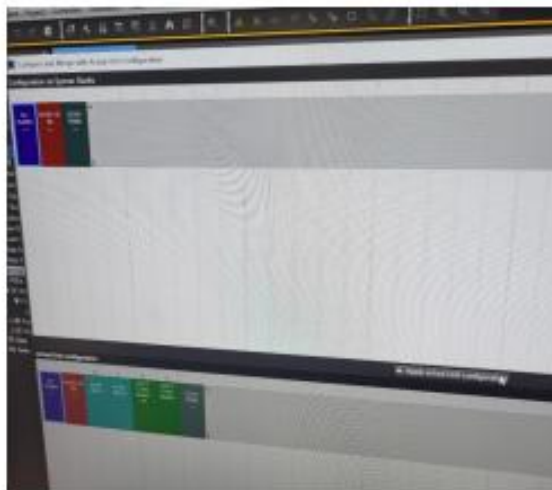




Laitteisto voidaan lisätä myös skannaamalla, tätä varten täytyy olla kytkeytyneenä PLC:hen sekä Sysmac Studio on oltava online-tilassa. Tätä varten tuplaklikataan projektin puunäkymästä CPU/Expansion Racks -> CPU Rack, painetaan hiiren oikealla näppäimellä CPU:n kuvaa ja valitaan "Compare and Merge with Actual Unit Configuration".

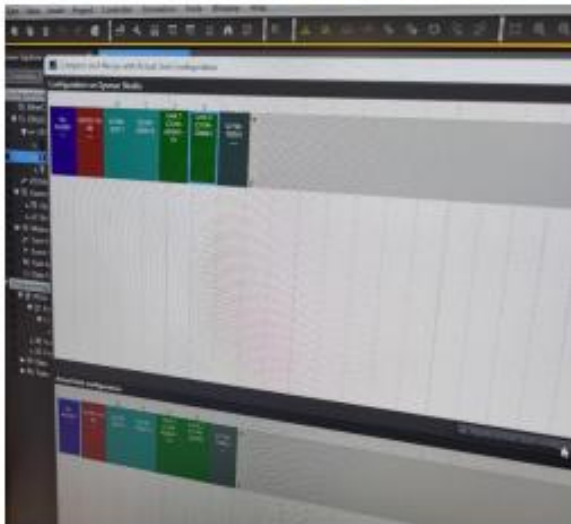


Hetken päästä Sysmac Studio näyttää löytämänsä laitteet verrattuna nykyisiin:





Valitaan "Apply actual Unit configuration" ja ohjelma korvaa edellisen laitekonfiguraation skannatulla:



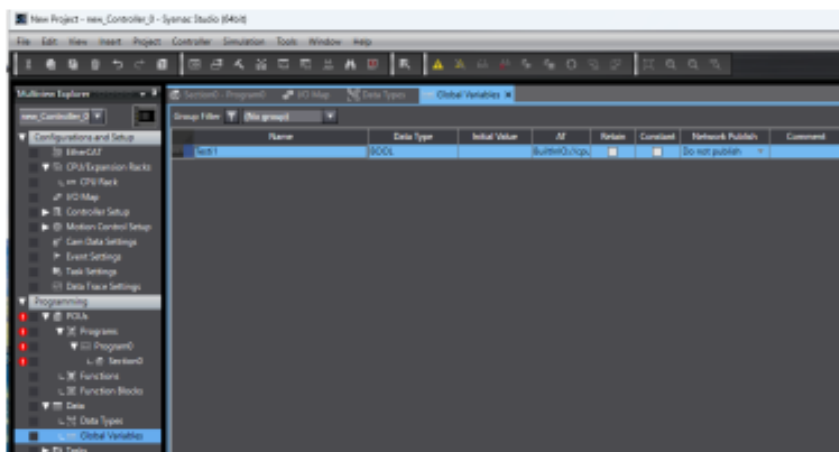
2.3 Muuttujien lisääminen ja linkittäminen I/O:hon

Järjestelmän I/O tuloja ja lähtöjä voidaan osoittaa joko globaaleiksi muuttujiksi tai jonkun ohjelman sisäiseksi muuttujiksi. Valitaan projektin puunäkymästä Configurations and Setup -> I/O Map, nimetään haluttu tulo tai lähtö ja valitaan "Variable Type"-kohdasta onko muuttuja globaali vai jonkun ohjelman sisäinen. Seuraavassa kuvassa on lisätty kaksi muuttujaa, toinen globaaliksi muuttujaksi ja toinen ohjelman sisäiseksi.

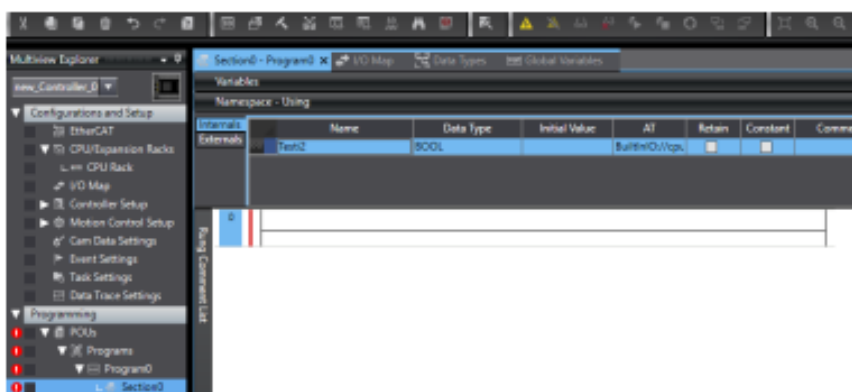
Position	Post	Description	I/O	Data Type	Variable	Variable Comment	Variable Type
Sub 01 01	Sub 01 01 Settings						
	Input BA 00	Input BA 00	F	BOOL	input		Global Variable
	Input BA 01	Input BA 01	F	BOOL	input		Program Variable
	Input BA 02	Input BA 02	F	BOOL			
	Input BA 03	Input BA 03	F	BOOL			
	Input BA 04	Input BA 04	F	BOOL			
	Input BA 05	Input BA 05	F	BOOL			
	Input BA 06	Input BA 06	F	BOOL			
	Input BA 07	Input BA 07	F	BOOL			
	Input BA 08	Input BA 08	F	BOOL			



Tämän jälkeen Testi1-niminen muuttuja löytyy globaaleista muuttujista projektin puunäkymässä Programming --> Data --> Global variables.



Ja testi2 taas Program0:n sisäisistä muuttujista.



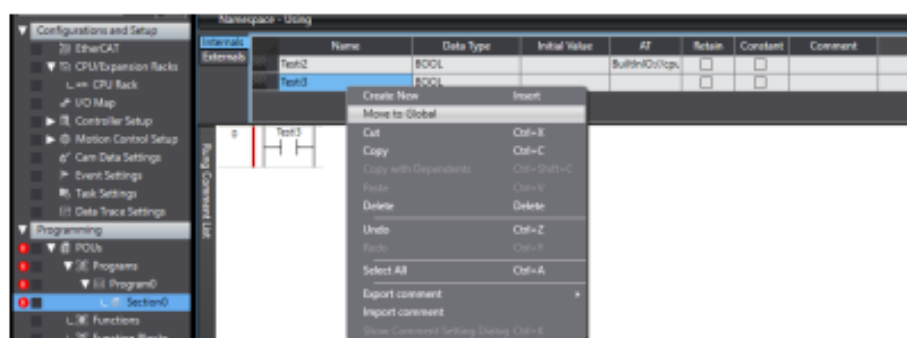


Ohjelmaa tehtäessä voidaan muuttuja luoda, muuttaa tarvittaessa globaaliksi ja linkata I/O-osoitteeseen seuraavalla tavalla:

Nimetään esimerkiksi input ohjelmassa Testi3-nimiseksi, jolloin järjestelmä luo Testi3-nimisen sisäisen muuttujan.



Jos halutaan muuttaa kyseinen muuttuja globaaliksi, klikataan Testi3:a ohjelman muuttujalistasta ensin hiiren oikealla ja sitten vasemmalla ja valitaan "move to global".



Muuttuja linkitetään haluttuun I/O-osoitteeseen Lisäämällä sen nimi halutun osoitteen "Variable"-kohtaan.

Name	Data Type	Initial Value	#I	Active	Constant	Comment
Input B1 00	BOOL		IO1	<input type="checkbox"/>	<input type="checkbox"/>	Global Variables
Input B1 01	BOOL		IO2	<input type="checkbox"/>	<input type="checkbox"/>	Program Internal
Input B1 02	BOOL		IO3	<input type="checkbox"/>	<input type="checkbox"/>	Global Variables
Input B1 03	BOOL		IO4	<input type="checkbox"/>	<input type="checkbox"/>	Global Variables

2.4 Toimenpiteet ohjelman muokkaamisen jälkeen

Kun ohjelmaa on muutettu online-editillä tai muuten, tallennetaan se toiminnallisuuden testauksen jälkeen pilvipalveluun JIS-Automation Oy --> JIS Suunnitteluosasto - Tiedostot --> Projektit-kansioon löytyvään projektin Ohjelmointi-kansioon. Jos kansiossa on jo vanhoja ohjelmistoversioita, voidaan muokattu ohjelma tallentaa kansioon esimerkiksi "muokkauspäivämäärä_projektin nimi"-muodossa.



3. Tekninen tuki ja ohjekirja

Jos tarvitset teknistä tukea CX-Programmerin suhteen, ota yhteyttä Omronin tekniseen tukeen:

- Puhelin 0207 464 200
- Sähköpostiosoite: info.fi@omron.com

CX-Programmerin ohjekirja on tallennettu pilvipalveluun.

Liite 4. Schneider Machine Expert-dokumentaatio

1 (11)

**JIS-Automation Oy**

Schneider Machine Expert



Sisällys

1. Ohjelman asennus ja uuden projektin luominen	3
1.1 Uuden projektin luominen	3
1.2 Uuden PLC-ohjelman luominen.....	3
2. Laitteiston lisäys ja tulojen linkitys muuttujiin	7
2.1 Uuden projektin luominen	7
2.2 Tulojen/lähtöjen linkitys muuttujiin	8
3. Ohjelman lataus logiikalle/logiikalta ja online-muutosten teko.....	10
3.1 Ohjelman lataus logiikalle/logiikalta	10
3.2 Online-muutosten teko	11
3.3 Toimenpiteet ohjelman muokkaamisen jälkeen	11
4. Tekninen tuki ja ohjekirja	11



1. Ohjelman asennus ja uuden projektin luominen

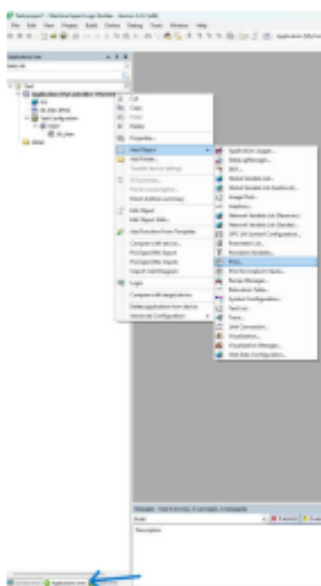
Machine Expertin, ja muiden käytössä olevien Schneider-tuotteiden asennustiedostot löytyvät pilvipalvelusta.

1.1 Uuden projektin luominen

Machine Expertissä tuodaan uusi projekti valitsemalla tehtäväpaikista File --> New Project tai painamalla Ctrl + N. Tämän jälkeen avautuvasta ikkunasta valitaan ohjaimen tyyppi sekä versio, haluttu ohjelmointikieli, sekä nimitään projekti.

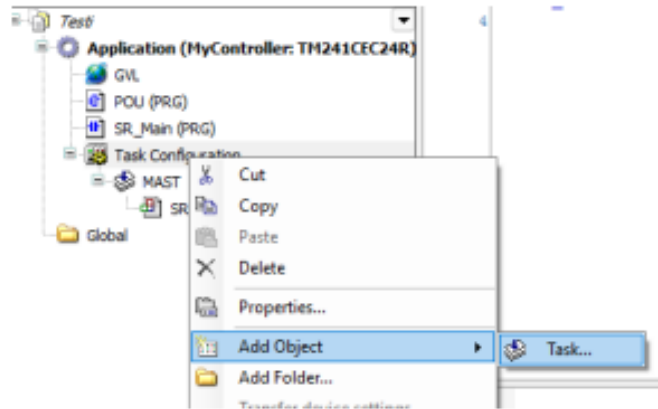
1.2 Uuden PLC-ohjelman luominen

Uusi ohjelma luodaan Applications-välilehdestä painamalla hiiren oikealla näppäimellä haluttua applikaatiota ja valitsemalla Add Object --> POU.

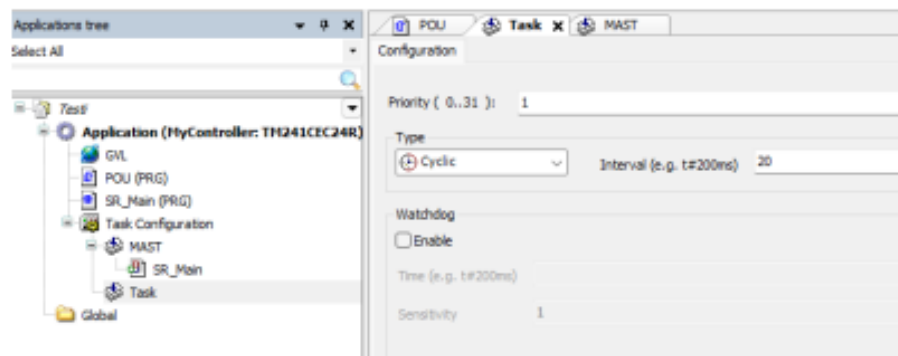




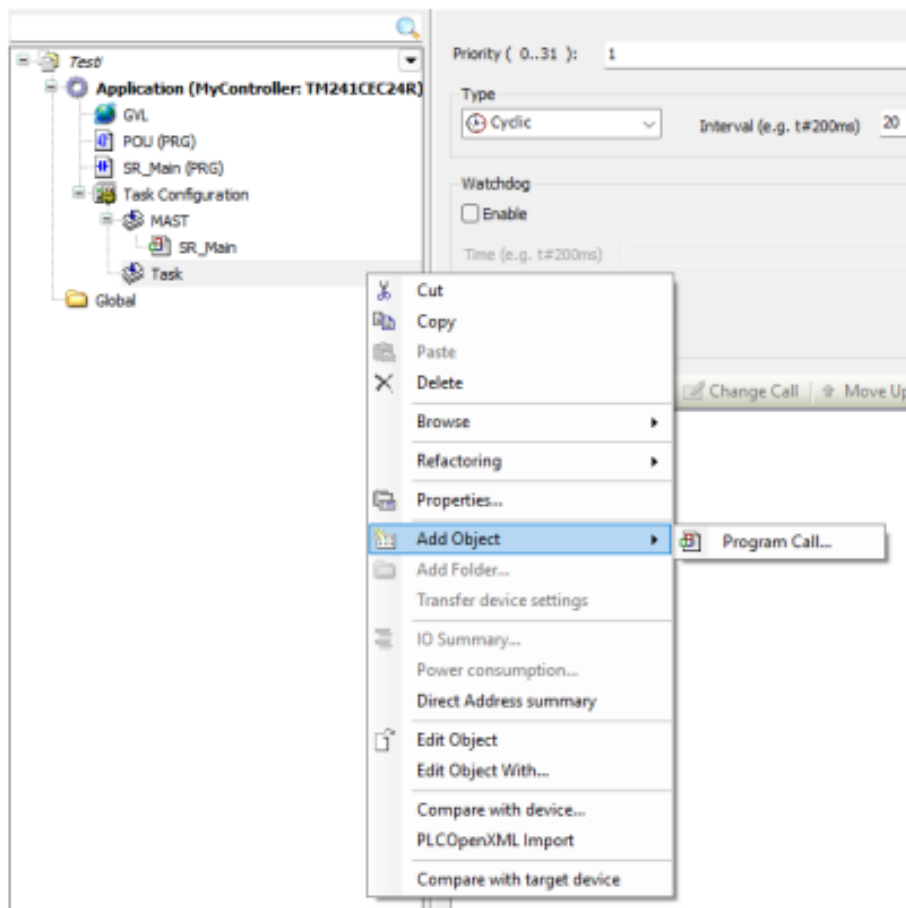
Avautuvasta ikkunasta valitaan, halutaanko luoda ohjelma, function block, vai funktio, nimetään POU ja valitaan haluttu ohjelmointikieli. POU:n luomisen jälkeen, on se vielä lisättävä johonkin ohjelman suoritusvaiheeseen. Jos halutaan luoda uusi suoritusvaihe, painetaan Task Configuration-kohtaa hiiren oikealla napilla ja valitaan Add Object --> Task.



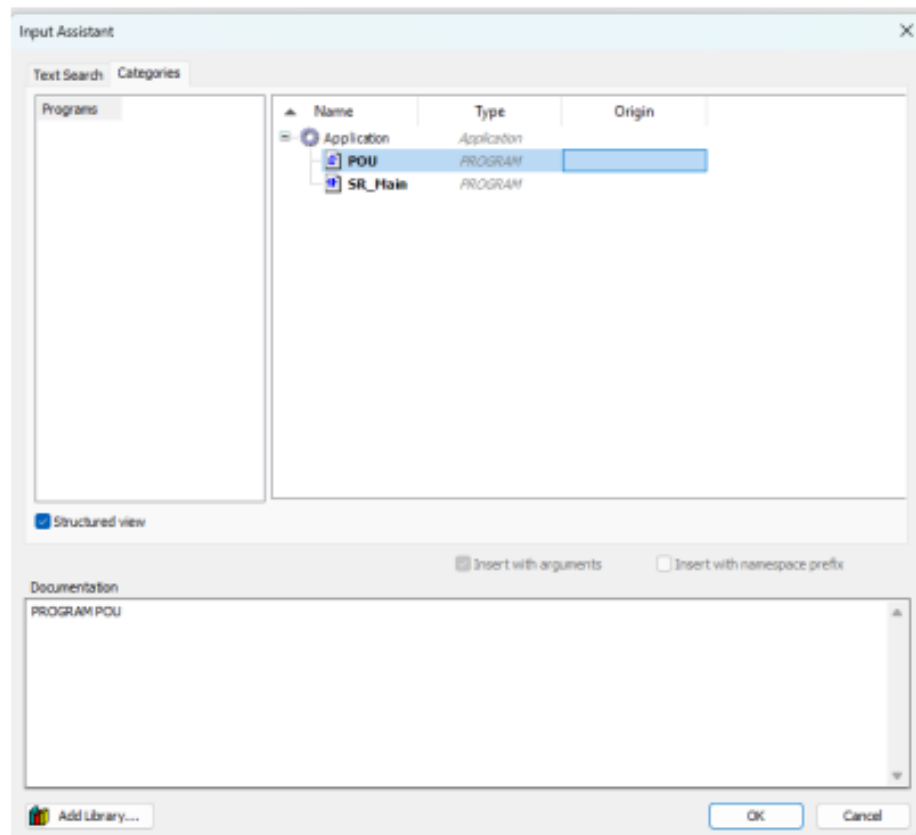
HUOM! Kahdella suoritusvaiheella (Task) ei voi olla sama prioriteetti (Priority). Jos luot uuden suoritusvaiheen, vaihda tarvittaessa vaiheen prioriteetti sen asetuksista. Prioriteetilla voi hallita missä järjestyksessä vaiheet suoritetaan. Suorittaminen aloitetaan prioriteetista nolla, ja edetään aina lukuun 31 saakka.



Lopuksi luotu POU vielä lisätään haluttuun suoritusvaiheeseen painamalla hiiren oikealla napilla suoritusvaihetta ja valitsemalla Add Object --> Program Call.



Haluttu POU lisätään vaiheeseen joko kirjoittamalla POU:n nimi "POU to call"-ikkunaan tai valitsemalla se "..."-nappia painamalla aukeavasta valikosta.

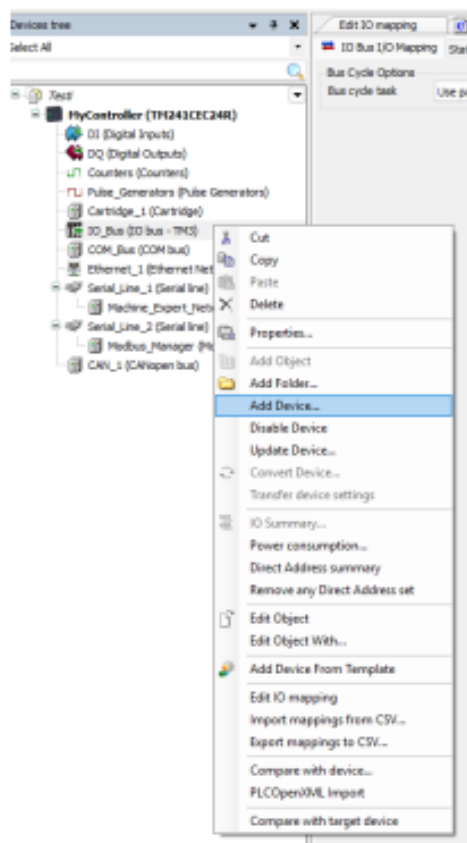




2. Laitteiston lisäys ja tulojen linkitys muuttujiin

2.1 Uuden projektin luominen

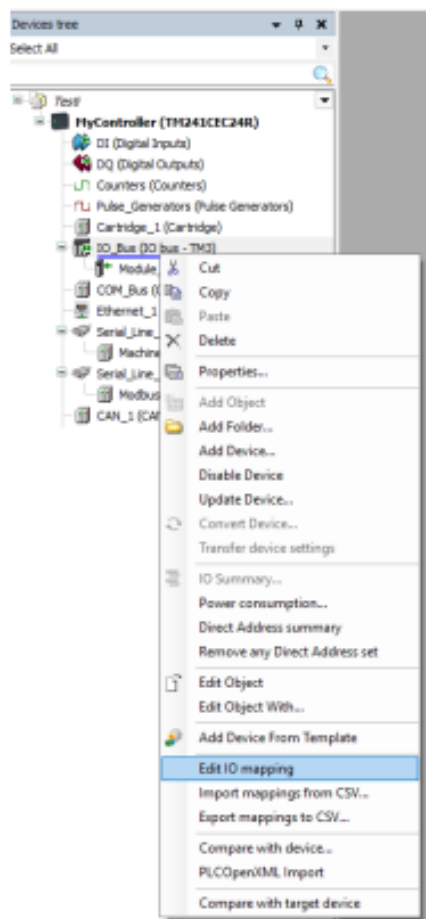
Machine Expertissä voi lisätä laitteistoa "Devices tree"-puunäkymässä painamalla hiiren oikealla näppäimellä esimerkiksi IO_Bus-kohtaa, jos haluaa lisätä IO-laitteen, ja valitsemalla valikosta "Add Device...". Tämän jälkeen avautuvasta valikosta valitaan lisättävä laite.





2.2 Tulojen/lähtöjen linkitys muuttujiin

Kun halutaan linkittää joku tulo tai lähtö jonkun ohjelman sisäiseen muuttujaan tai globaaliin muuttujaan, valitaan "Devices tree"-puunäkymästä hiiren oikealla napilla IO_Bus ja tämän jälkeen "Edit IO-mapping".





Tämän jälkeen valitaan haluttu tulo tai lähtö, siirrytään "..."-napilla valintaikkunaan...

Variable	Channel	Address	Type	D
IX3_0				
Module_3				
Inputs				
Module_3_00	00	%I0	BYTE	
I1	01	%I0.0	BOOL	
I2	02	%I0.1	BOOL	
I3	03	%I0.2	BOOL	
I4	04	%I0.3	BOOL	
I5	05	%I0.4	BOOL	
I6	06	%I0.5	BOOL	
I7	07	%I0.6	BOOL	
I8	08	%I0.7	BOOL	

...ja valitaan ohjelman muuttuja, joka halutaan linkittää kyseiseen tuloon tai lähtöön.

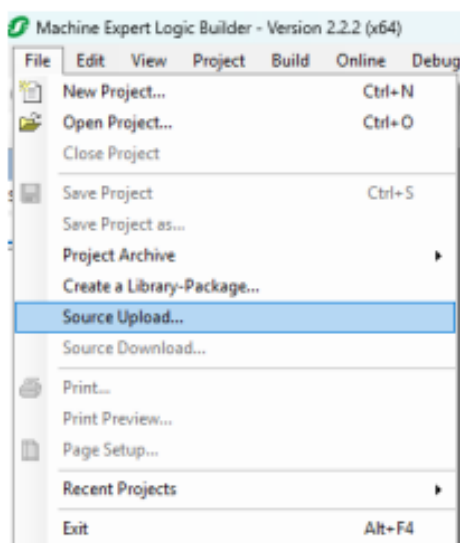
Yllä olevassa tapauksessa siis liitetään Input1-niminen muuttuja POU-nimisestä ohjelmasta tuloporttiin IX3.0.



3. Ohjelman lataus logiikalle/logiikalta ja online-muutosten teko

3.1 Ohjelman lataus logiikalle/logiikalta

Jos tiedetään PLC:n admin-tunnukset, ja on kytkeydytty PLC:hen esimerkiksi ethernet-johdolla, voidaan PLC:n logiikkaohjelma ladata Machine Expertiin valitsemalla valikkopalkista File --> Source Upload.



Tämän jälkeen valitaan ohjain, josta ohjelma ladataan, kirjaututaan ohjeimelle ja valitaan polku, johon ladattu ohjelma tallennetaan.

Ohjelman lataus tietokoneelta PLC:lle tapahtuu valitsemalla "Source Download" samasta kohdasta valikkopalkkia. Tämän jälkeen valitaan PLC, jolle ohjelma halutaan ladata ja kirjaututaan ohjaimen admin-tunnuksilla.



3.2 Online-muutosten teko

Jotta päästään tekemään online-muutoksia ohjelmalle, täytyy järjestelmään ensin kirjautua sisään valitsemalla valikkopalkista Online --> Login, ja kirjautumalla ohjaimelle sen admin-tunnuksilla. Tämän jälkeen painetaan kohdetta, jota halutaan muuttaa hiiren oikealla näppäimellä ja valitaan "Edit Object (Offline)", tehdään muutokset ja lopuksi valutaan valikkopalkista Online --> Online Change

3.3 Toimenpiteet ohjelman muokkaamisen jälkeen

Kun ohjelmaa on muutettu online-editillä tai muuten, tallennetaan se toiminnallisuuden testauksen jälkeen pilvipalveluun JIS-Automation Oy --> JIS Suunnitteluosasto - Tiedostot ---> Projektit-kansiosta löytyvään projektin Ohjelmointi-kansioon. Jos kansiossa on jo vanhoja ohjelmistoversioita, voidaan muokattu ohjelma tallentaa kansioon esimerkiksi "muokauspäivämäärä_projektin nimi"-muodossa.

4. Tekninen tuki ja ohjekirja

Jos tarvitset teknistä tukea Machine Expertin suhteen, ota yhteyttä Schneiderin tekniseen tukeen:

- Puhelin 09 8171 0510
- Sähköpostiosoite: asiakaspalvelu.fi@se.com

Machine Expertin ohjekirja on tallennettu pilvipalveluun.