TwinCAT 3 Opas

Juhana Jauhiainen

2022

1 Mikä on TwinCAT?

TwinCAT (The Windows Control and Automation Technology) on Beckhoffin automaatiojärjestelmien ohjelmointiin, ja ohjelmien ajamiseen, käytettävä ohjelmisto.

TwinCAT on Windows PC-pohjainen ohjelmisto, joka muuttaa minkä tahansa PC:n tehokkaaksi automaatiojärjestelmien ohjaimeksi.

TwinCAT 3 on ohjelmiston uusin versio, jonka avulla automaatiohjelmointia voidaan tehdä IEC 61131-3 standardin mukaisten kielten lisäksi C ja C++ ohjelmointikielillä. Twincat 3 on rakennettu Microsoftin Visual Studio ohjelmointiympäristön päälle.

TwinCat on jaettu komponentteihin, joista **TC3 Engineering** sisältää logiikoiden konfigurointi, ohjelmointi ja debuggaus ominaisuudet. Se on edelleen jaetta pienempiin komponentteihin, kute **TE1000** (ohjelmointi) ja **TE2000,** jonka avulla voidaan luoda web-selaimella toimivia käyttöliittymiä PLC-projekteille.

TC3 Engineering komponenttien lisäksi **Functions** komponenttien avulla kehitysympäristöön voidaan lisätä ominaisuuksia kuten analytiikka työkaluja, robotiikkaan liittyviä työkaluja tai esimerkiksi tuki **OPC UA** protokollalle.

Lisätietoa eri TwinCAT komponenteista löytyy Beckhoffin sivuilta. https://www.beckhoff.com/english.asp?twincat/twincat-3.htm

2 Asennus ja ensimmäinen käynnistys

2.1 Vaatimukset

TwinCATin viralliset vaatimukset eivät ole kovin kummoiset, mutta kannattaa ottaa huomioon, että Visual Studio on suhteellisen raskas ohjelmisto, joten käytettävän koneen tulisi olla tehokas. Beckhoffin antamat viralliset vaatimukset on listattu alla.

- Windows 8 tai uudempi
- Prosessori 1.6Ghz
- 2GB RAM
- 3GB kiintolevytilaa

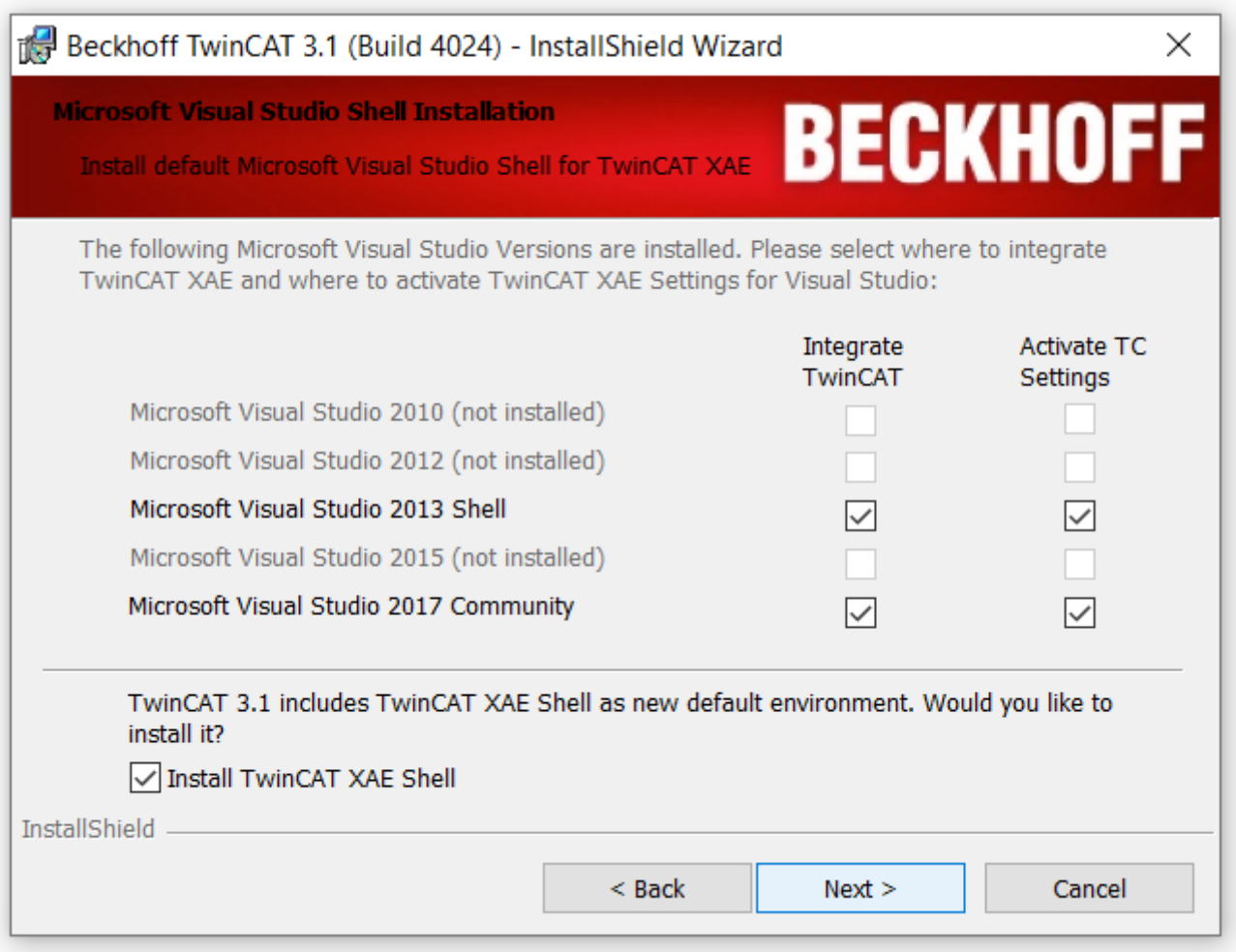
2.2 Asennus

Koska TwinCAT toimii osana Microsoftin Visual Studio ohjelmointiympäristöä, täytyy ensin asentaa Visual Studio. Oppaan kirjoitushetkellä Beckhoffin web-sivujen mukaan uusin Visual Studio version, jota TwinCAT 3 tukee on Visual Studio 2017.

Visual Studion voi ladata Microsoftin sivuilta osoitteesta https://visualstudio.microsoft.com/ ja sen Community version lataus ja käyttö on täysin ilmaista. Tässä oppaassa käytetään Visual Studion versiota 2017.

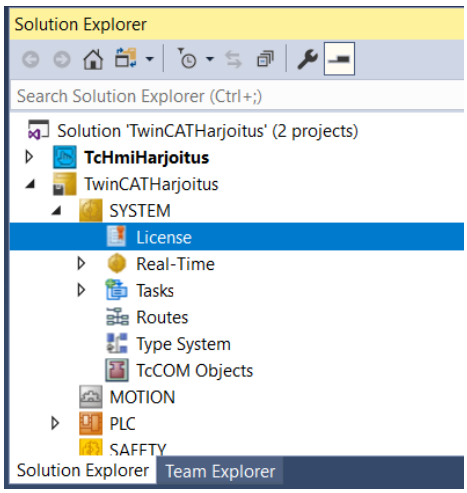
Kun Visual Studio on asennettu, voidaan asentaa halutut TwinCAT komponentit. **TE1000 XAE** on ainut vaadittu komponentti, jos halutaan vain kehittää automaatio-ohjelmia, mutta käyttöliittymien kehittämiseen kannattaa asentaa myös **TE2000 HMI**. Molempien lataus onnistuu Beckhoffin sivuilta https://www.beckhoff.com/english.asp?download/default.htm

TE1000 XAE asennuksen aikana voidaan valita minkä Visual Studio asennusten kanssa TwinCAT integroidaan.

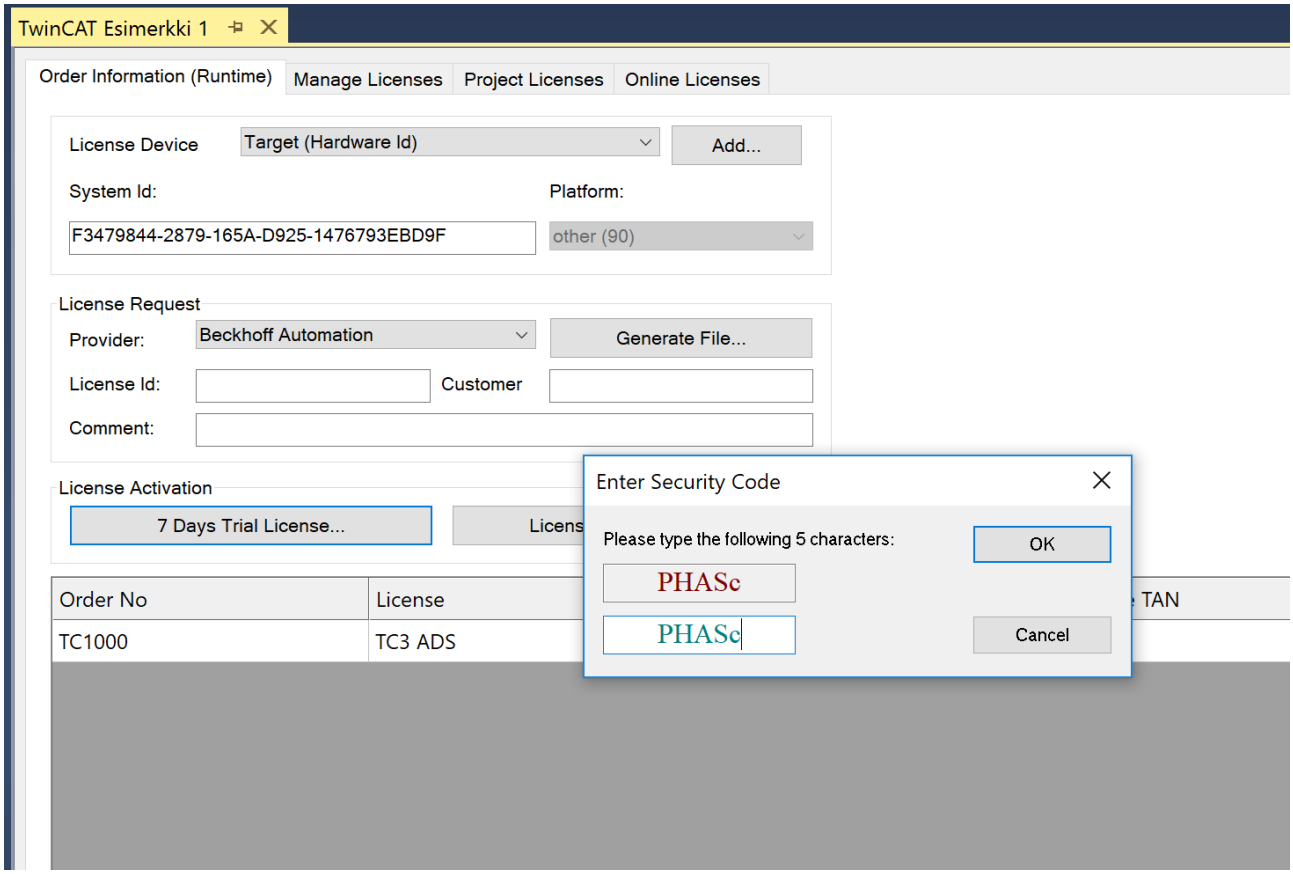


2.3 Lisenssi

TwinCAT 3 kehitysympäristön lisenssi on täysin ilmainen mutta se on aktivoitava uudelleen 7 päivän välein. Lisenssin aktivointi tapahtuu TwinCat projektin SYSTEM kohdan alla olevasta License kohdasta.



Kuva 2.3.1 Lisenssin uusinta tapahtuu License kohdasta SYSTEM osion oalta



Kuva 2.3.2 Lisenssi uusitaan tapahtuu syöttämällä annettu 5-merkin mittaunen koodi

Kehityskoneen lisenssin voi aktivoida klikkaamalla **Activate 7 Days Trial License…** ja täyttämällä kysytty koodi.

TwinCAT ehdottaa lisenssin uudelleen aktivointia myös aina käynnistettäessä, jos lisenssi on ehtinyt vanhentua.

Beckhoffin automaatiolaitteet käyttävät samaa TwinCAT XAR (eXtended Automation Runtime) ympäristöä kuin kehitysympäristökin, mutta niille täytyy lisenssi ostaa laitteen hankinnan yhteydessä tai lisenssi täytyy uusia seitsemän päivän välein.

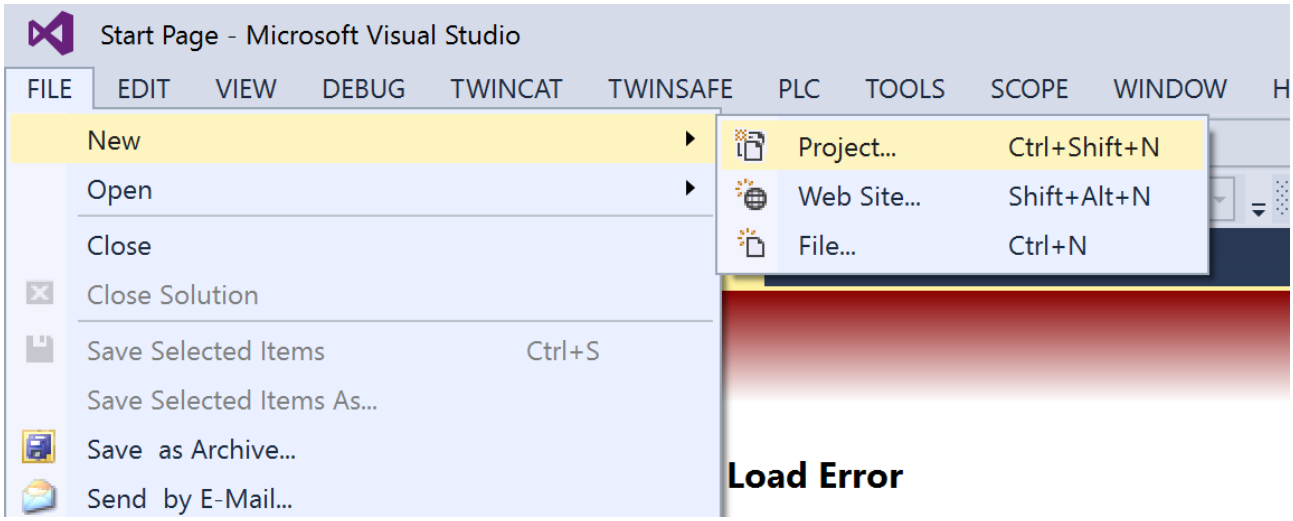
3 Ensimmäinen PLC projekti

3.1 TwinCAT projekti

TwinCAT projekti on kokonaisuus, joka voi sisältää TwinCATin eri komponenteilla tehtyjä aliprojekteja (PLC, HMI jne.), jotka liittyvät samaan kokonaisuuteen.

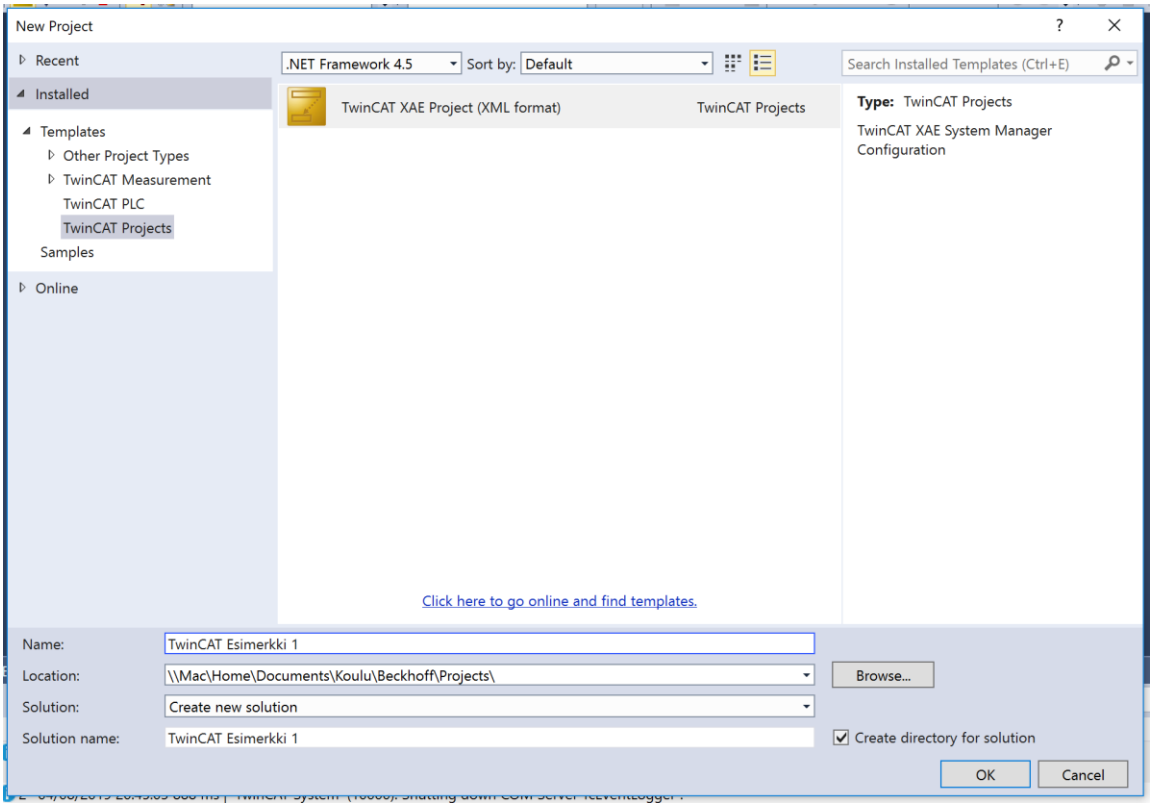
3.2 Uusi projekti

Uusi TwinCAT projekti aloitetaan valitsemalla Visual Studio File valikosta New Project.



Kuva 3.2.1 Uuden TwinCAT projektin aloittaminen

Sen jälkeen valitaan Template valikosta TwinCAT Project XAE, annetaan projektille nimi ja sijainti ja luodaan projekti klikkaamalla OK.

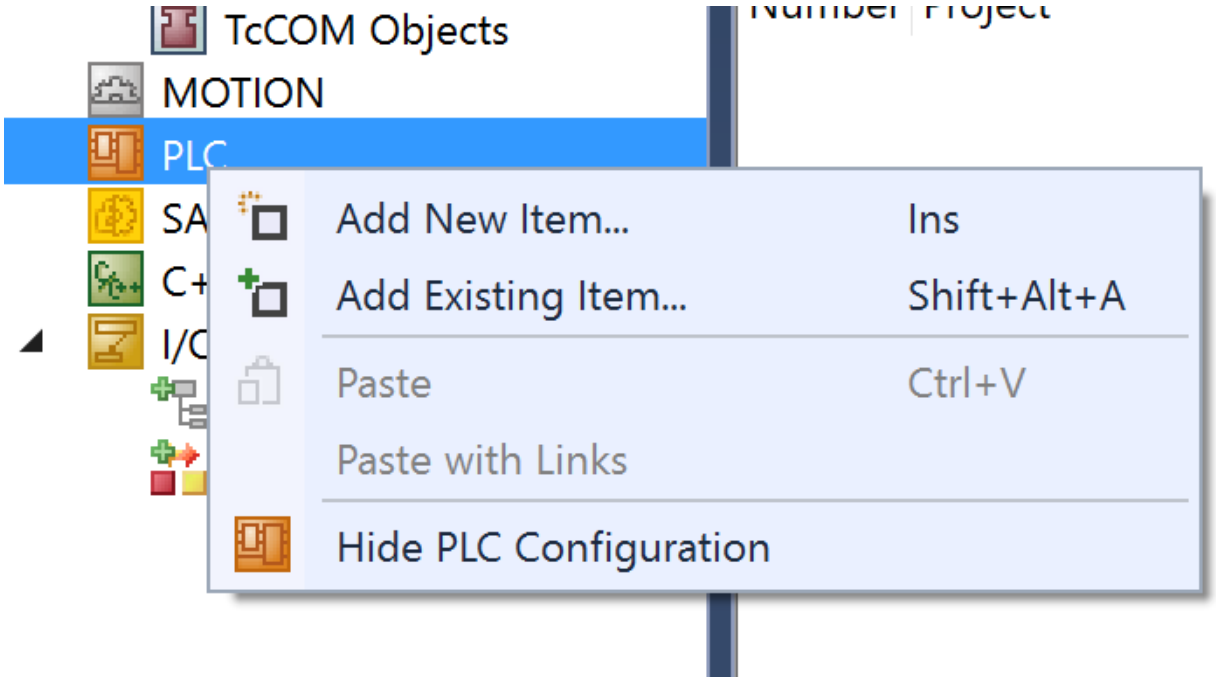


3.3 PLC projekti

Kun TwinCat projekti on luotu, täytyy siihen vielä lisätä PLC projekti, jotta päästään ohjelmoimaan. TwinCatissä jokainen PLC projekti vastaa "virtuaalista" logiikkakontrolleria. PLC-projekti sisältää logiikalle ladattavat ohjelmat, muuttujat ja omat tietotyypit.

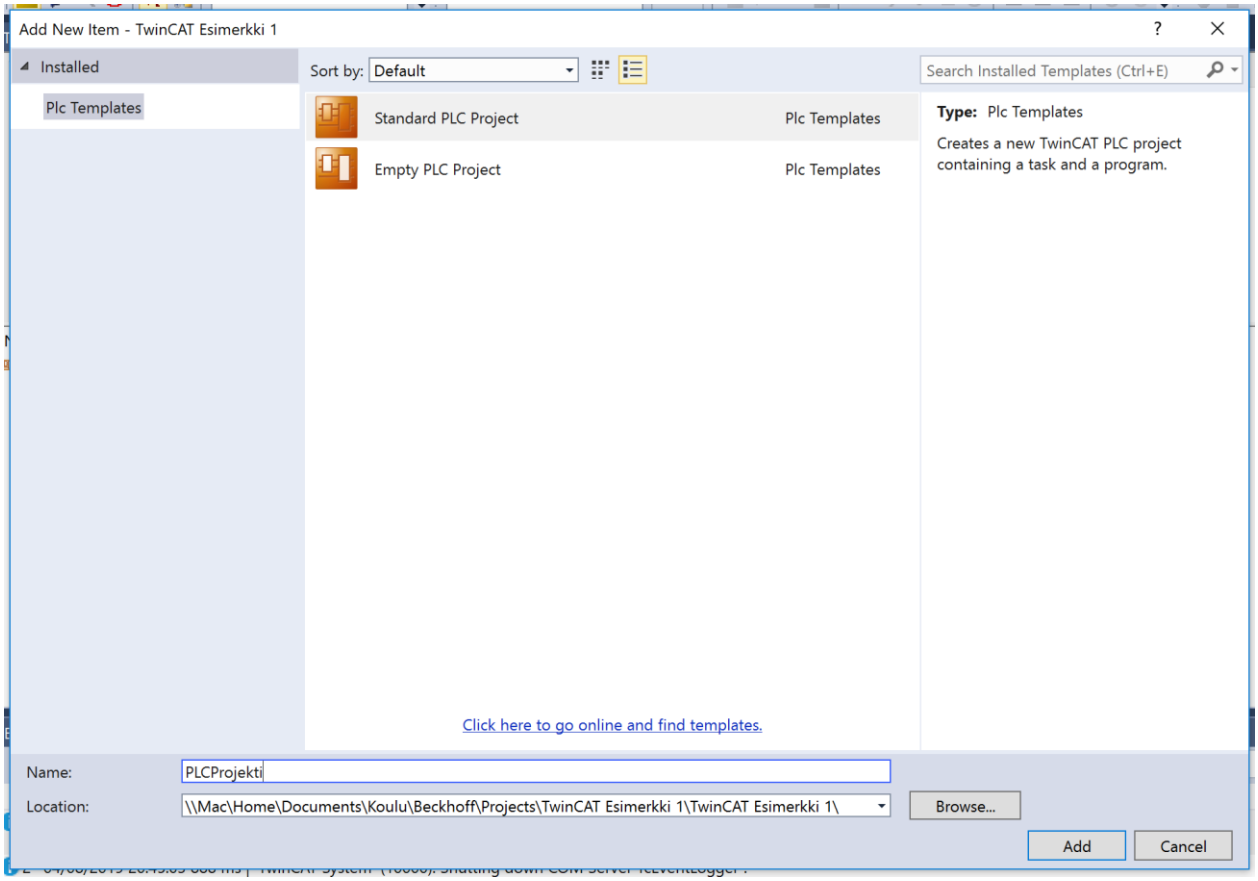
PLC-projektin lisäys tapahtuu klikkaamalla hiiren oikealla puunäkymän kohtaa PLC, ja valitsemalla **Add New Item.**

Kuva 3.2.2 Uuden TwinCAT projektin nimeäminen



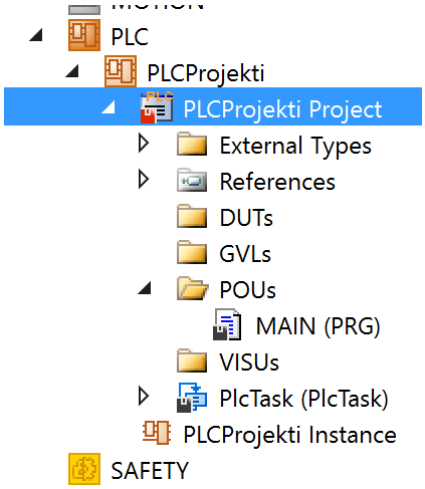
Kuva 3.3.1 PLC-projektin lisäys TwinCAT projektiin

Avautuvasta ikkunasta valitaan projektityypiksi **Standard PLC Project**, annetaan projektille nimi ja klikataan OK.



Kuva 3.3.2 Projektin tyypin valinta

PLC-projektin lisäyksen jälkeen projektipuuhun ilmestyy PLC-projekti annetulla nimellä ja sen alta löytyy PLC-projektin oletusrakenne.



Kuva 3.3.3 Projektin puunäkymään PLC-kohdan alle ilmestyy luotu projekti tiedostoineen

3.4 Kansiorakenne

PLC projektin alle luodaan automaattisesti kansiorakenne, joista tärkeimmät ovat

- **References** Ulkoisten kirjastojen lisääminen.
- **DUTs** Data Unit Types, tietotyyppien määrittelyt.
- **GVL** Globaalien muuttujien määrittely. Globaalit muuttujat ovat käytettävissä kaikissa logiikkaohjelmissa ja käyttöliittymässä.
- **POU** Program Organization Unit, sisältää logiikkaohjelmat ja niihin liittyvät paikalliset muuttujat.
- **VISUs** Käyttöliittymien määrittely ja luominen.

VISUs kansion alle voidaan tehdä käyttöliittymä Beckhoffin vanhemmalla visualisointi menetelmällä. Tässä oppaassa sitä ei kuitenkaan käsitellä vaan käyttöliittymät tehdään uudemmalla TwinCAT TE2000 HMI komponentilla. Lisätietoa löytyy Beckhoffin dokumentaatiosta kohdasta **TwinCAT 3 → TE1000 XAE → PLC → Creating a visualization**

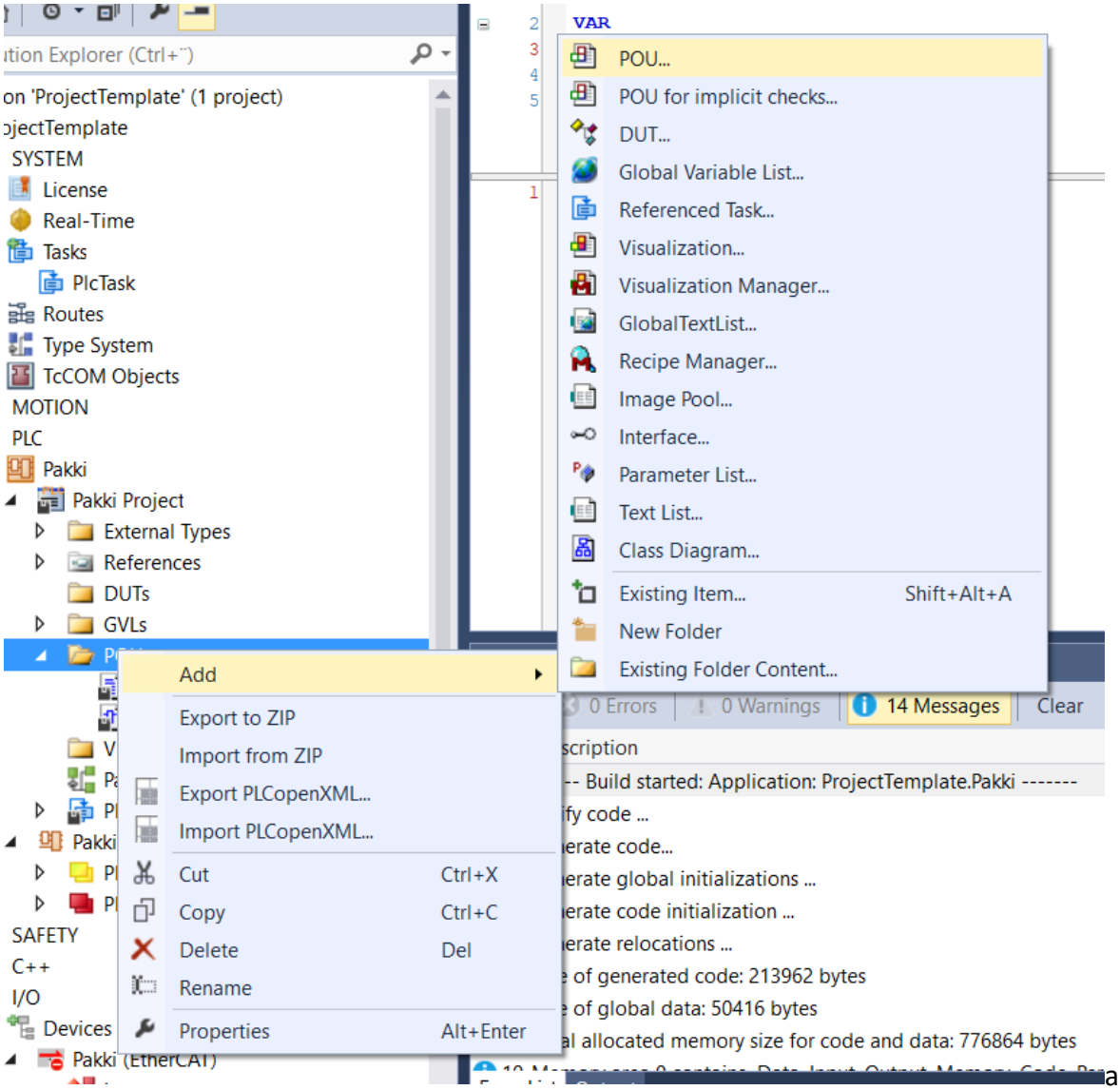
4 Ohjelmointi

TwinCat 3 tukee kaikkia **IEC 61131-3** standardin mukaisia ohjelmointimenetelmiä, eli LD (Ladder diagram), FBD (Function Block Diagram), ST (Structured Text), IL (Instruction List) ja SFC (Sequential Function Chart). Tässä ohjeessa käydään läpi lyhyesti LD, FBD ja ST ohjelmointimenetelmien käyttö TwinCat 3:ssa. Näistä ST on tekstipohjainen ja muistuttaa eniten korkeamman tason ohjelmointikieliä kuten (PHP, Python tai C). ST on myös Beckhoffin suosittelema PLC ohjelmointitapa.

4.1 PLC ohjelman lisäys

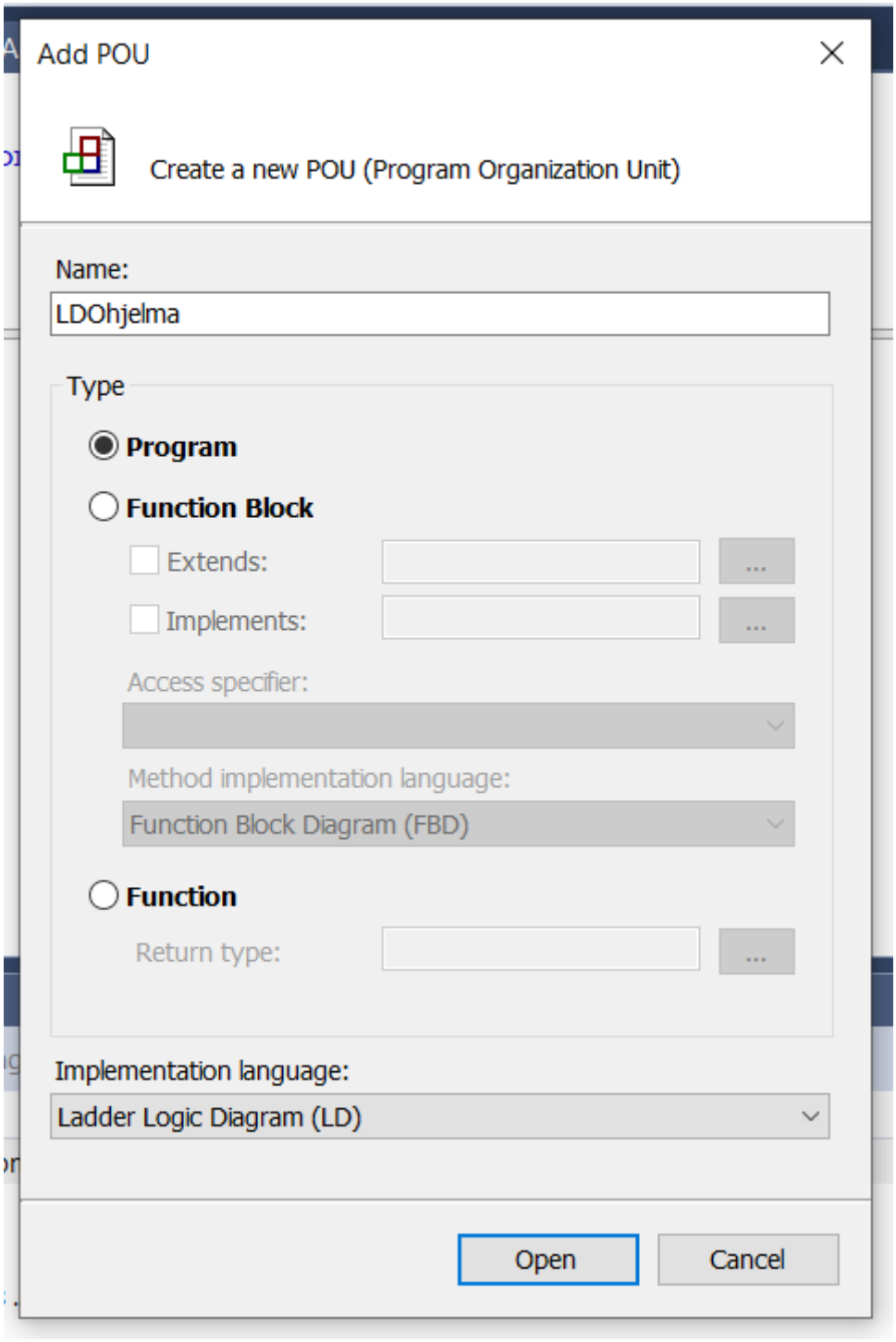
Kun TwinCAT projektiin lisää PLC projektin, luo TwinCat automaattisesti MAIN-ohjelman PLC-projektin kansioon POU. Automaattisesti luotu MAIN-ohjelma on ST-ohjelma, joka on oletuksena tyhjä. Yksinkertaisen ohjelman voi toteuttaa halutessaan suoraan MAIN-tiedostoon, mutta monimutkaisemmat logiikkaohjelmat kannattaa jakaa useampaan tiedostoon.

POU-kansioon voidaan lisätä eri ohjelmointimenetelmillä toteutettuja ohjelmia eri klikkaamalla kansiota hiiren oikealla ja valitsemalla **Add** ja **POU**. Avautuvasta ikkunasta voidaan valita ohjelmointimenetelmä ja antaa ohjelm



alle nimi.

Kuva 4.1.1 Uuden ohjelman lisääminen POU kansioon



Kuva 4.1.2 Ohjelman lisäämiseen käytettävä ikkuna

Uudet ohjelmat eivät automaattisesti ole "käytössä" vaan ne pitää erikseen määritellä ajettavaksi. Tämä voidaan tehdä esimerkiksi suorittamalla uusi ohjelma MAIN-ohjelmasta käsin.

```
UusiOhjelma();
```

4.2 Muuttujat

Muuttujia voidaan TwinCatissä määrittää joko paikallisesti tai globaalisti. Paikalliset muuttujat ovat yhden logiikkaohjelman käytössä, eikä niiden arvoihin ole pääsyä muista ohjelmista. Globaaleja muuttujia taas voidaan käyttää kaikista PLC-projektin ohjelmista ja käyttöliittymistä. Globaalit muuttuja ovat myös mahdollista liittää johonkin fyysiseen tuloon tai lähtöön.

Eri muuttujatyyppejä ovat mm. BOOL, INT, UINT, REAL, TIME, DATE, ARRAY jne. Myös ajastimet ja triggerit määritetään muuttujina, joilla on omat tyyppinsä.

4.3 Ohjelman muuttujat

PLC-ohjelman muuttujat voidaan asettaa ohjelman muokkausikkunan yläosassa olevasta editorista. Määrittely voidaan tehdä joko tekstipohjaisesti, taulukkona tai **Auto Declare** ikkunan avulla.

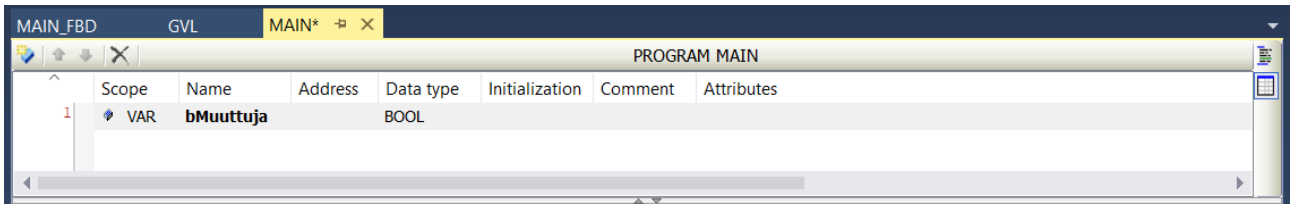
Tekstipohjainen muuttujien määrittely tapahtuu formaatissa:

```
/muuttujanNimi/ : /tyyppi/;
```

Esimerkiksi **BOOL**-tyyppinen muuttuja, jonka nimi on **muuttuja1**, esiteltäisiin näin.

```
muuttuja1 : BOOL;
```

Kuva 4.3.1 BOOL muuttuja esiteltynä editorilla



Kuva 4.3.2 BOOL muuttuja esiteltynä taulukon avulla

4.4 Globaalit muuttujat

Globaalien muuttujien avulla logiikkalaitteiston tulot ja lähdöt saadaan helposti liitettyä ohjelmissa käytettäviin muuttujiin. Muuttujien määrittely onnistuu lisäämällä **GVL** kansioon lista, joka sisältää halutut muuttuja. Muuttujien lisäys listaan tapahtuu samalla tavalla kuin paikallisten muuttujien lisäys ohjelmiin.

Muuttujaan voidaan määrittää myös erityinen rekisteriosoite, joka voidaan myöhemmin liittää johonkin fyysiseen tuloon tai lähtöön. Määrittäminen tapahtuu lisäämällä muuttujan nimen perään AT ja halutun rekisterin tyyppi ja osoite %-merkin jälkeen. Rekisterin tyyppi ja osoite erotetaan X-kirjaimella. Esim. input rekisteri 0.0

```
/muuttujanNimi/ AT %IX0.0 : /tyyppi/;
```

Asettamalla osoitteen tilalle tähti (*), asetetaan osoite automaattisesti.

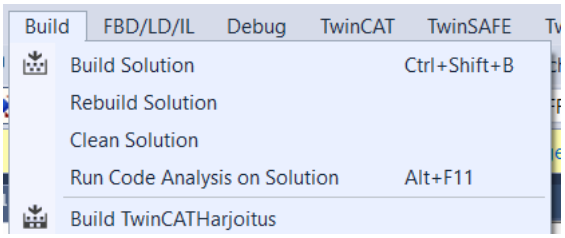
```
/muuttujanNimi/ AT %I* : /tyyppi/;
```

Muuttujat voidaan linkittää johonkin lähtöön tai tuloon I/O valikon alta, kun laitteet on lisätty.

Output rekisterit määritetään korvaamalla I-kirjain Q:lla.

```
/muuttujanNimi/ AT %QX0.0 : /tyyppi/;
```

Kun projekti käännetään muuttujien määrittämisen jälkeen (**Build Solution**), ilmestyy PLC-projektin alla olevaan virtuaaliseen logiikkaohjaimeen (**PLC Instance**) globaaleja muuttujia vastaavat lähdöt ja tulot.



Kuva 4.4.1 Build Solution toiminto löytyy Build valikon alta

Kun globaaleja muuttujia halutaan käyttää ohjelmissa, täytyy muuttujan nimen eteen lisätä sen tiedoston nimi jossa muuttuja on määritelty. Esim **GVL.muuttujanNimi**

4.5 Ohjelmointiesimerkit

Seuraavaksi esitellään LD, FBD ja ST ohjelmointia yksinkertaisen esimerkin avulla. Ennen esimerkkien tekemistä tulisi määrittää globaalit **BOOL** muuttujat *bMerkkivalo*, *bKytkin* ja *bPainonappi* GVL-tiedostoon.

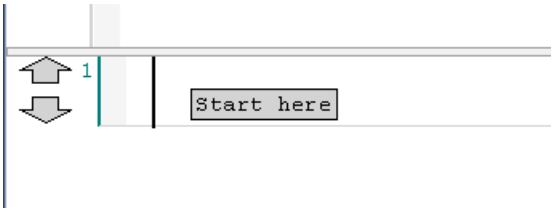
```
VAR_GLOBAL
  bKytkin AT %I* : BOOL;
  bPainonappi AT %I* : BOOL;
  bMerkkivalo AT %Q* : BOOL;
END_VAR
```

bMerkkivalo on globaali BOOL muuttuja, joka on liitetty digitaalisen lähtöön. **bPainonappi** on globaali BOOL muuttuja, joka on liitetty digitaaliseen tuloon. **bKytkin** on globaali BOOL muuttuja, joka on liitetty digitaaliseen tuloon. Kytkimen voi tässä tapauksessa ajatella olevan kiertokytkin, joka on kytketty siten että asennossa 1, muuttuja **bKytkin** saa arvon tosi.

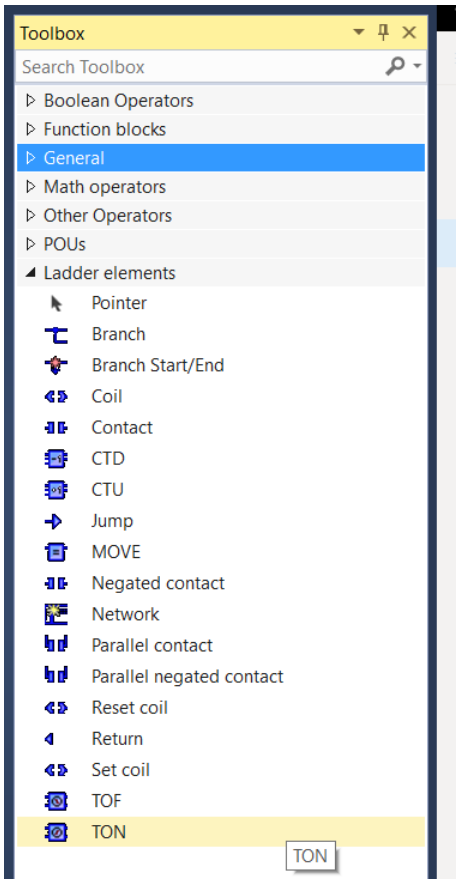
4.5.1 Ladder diagram -ohjelmointi

LD-ohjelmointia varten lisätään PLC-projektiin uusi ohjelma, jonka tyypiksi **(Implementation language)** määritetään **Ladder Diagram**.

Sovelluksen oikeaan reunaan avautuu **ToolBox**, jonka alta LD-ohjelmointiin käytettävät työkalut löytyvät. LD-komponentteja voidaan lisätä ohjelmaan raahaamalla niitä **ToolBox**-ikkunasta, ja tiputtamalla ne sopivaan kohtaan ohjelmassa. Ensimmäistä komponenttia raahatessa, ilmestyy editoriin **Start Here** -laatikko, johon komponentti voidaan pudottaa.

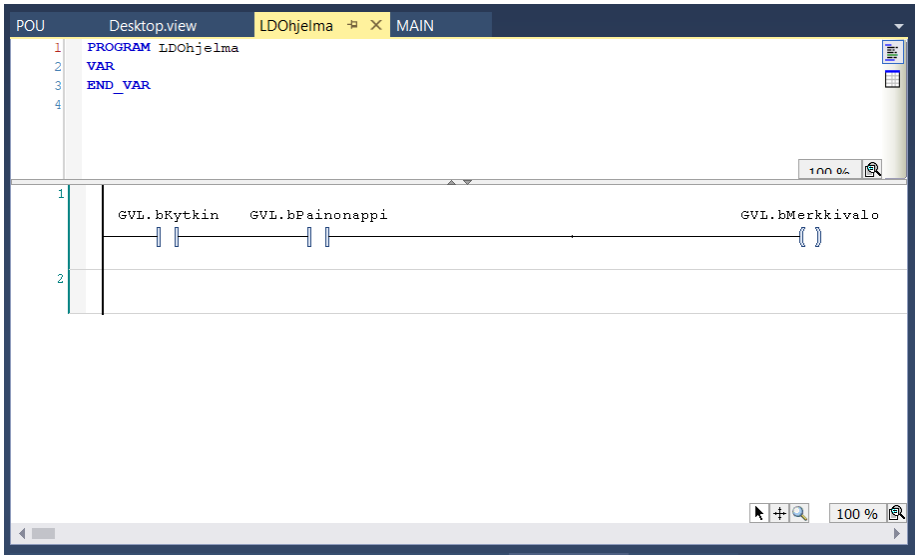


Kuva 4.5.1 Start Here laatikko ilmestyy kun ladder komponenttia alkaa raahaamaan



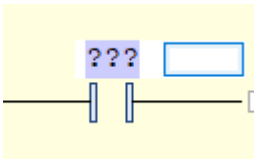
Kuva 4.5.2 Ladder ohjelmointiin käytettäviä työkaluja

Tehdään seuraavaksi yksinkertainen AND-ohjelma, joka sytyttää merkkivalon, kun kiertokytkin on asennossa 1, ja painonappia painetaan. Tätä varten tarvitaan kaksi **Contact**-komponenttia ja yksi **Coil**-komponentti.



Kuva 4.5.3 Valmis ladderohjelma

Muuttujat liitetään komponentteihin kirjoittamalla muuttuja nimi komponentin vieressä olevaan **???**-kenttään tai klikkaamalla sen vieressä olevaan laatikkoa, jolloin muuttujan voi valita listasta.



Kuva 4.5.4 Contact komponentti ennen muuttujan määrittämistä

Kun ohjelma on valmis, MAIN-ohjelmaan lisätään vielä komento, jotta ohjelmaa ajetaan PLC:llä.

```
LDOhjelma();
```

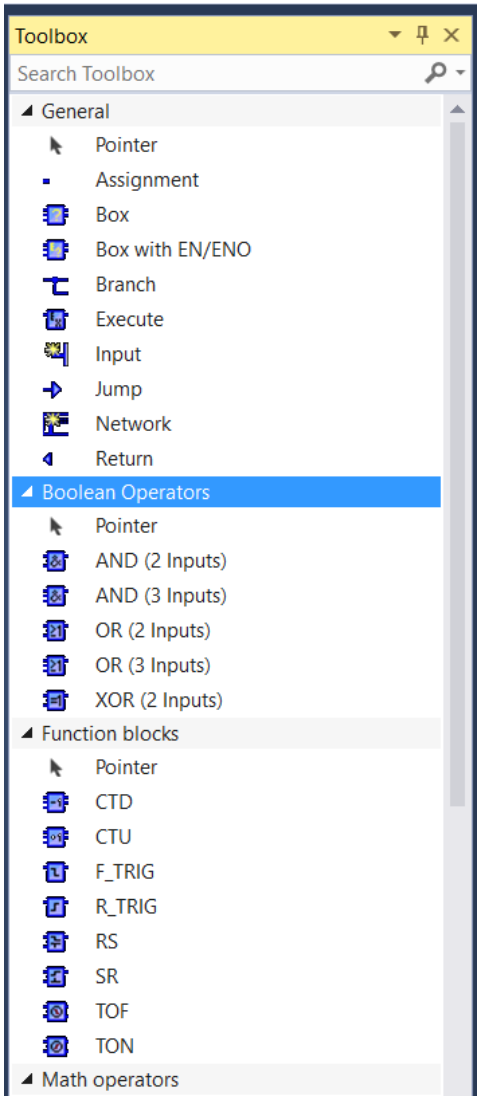
LDOhjelma on tässä ohjelman tiedostonimi.

Tässä vaiheessa projekti kannatta kääntää (ylävalikosta **Build Solution**), jotta mahdolliset virheet tulevat näkyviin.

4.5.2 Function Block Diagram -ohjelmointi

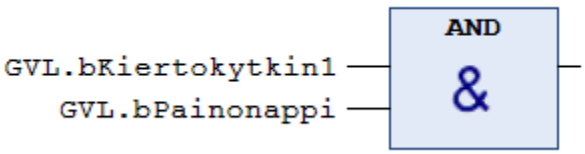
FBD-ohjelmointimenetelmällä voidaan logiikka ohjelmoida erilaisten funktiolohkojen avulla. PLC-projektiin lisätään ohjelmointia varten uusi ohjelma, valiten tällä kertaa tyypiksi **Funcion Block Diagram**.

Toolbox-ikkunasta löytyvät FBD-ohjelmointiin tarvittavat lohkot ja niitä voi lisätä ohjelmaan raahaamalla ne ohjelmanmuokkausnäkymään.



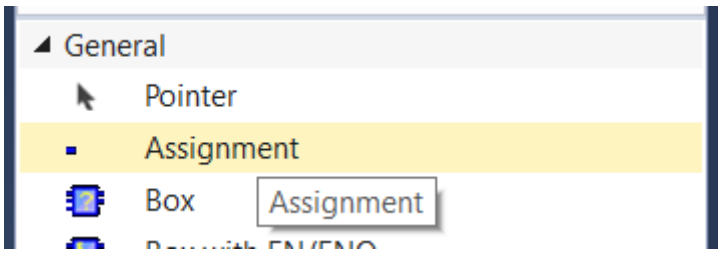
Kuva 4.5.5 FBD-ohjelmointilohkot

Merkkivalon sytyttämiseen käytettävä ohjelma voidaan toteuttaa käyttämällä **AND**-lohkoa, jossa on 2-tuloa. Tuloihin määritetään muuttujat **bKytkin** ja **bPainonappi** samaan tapaan kuin LD-ohjelmassa.



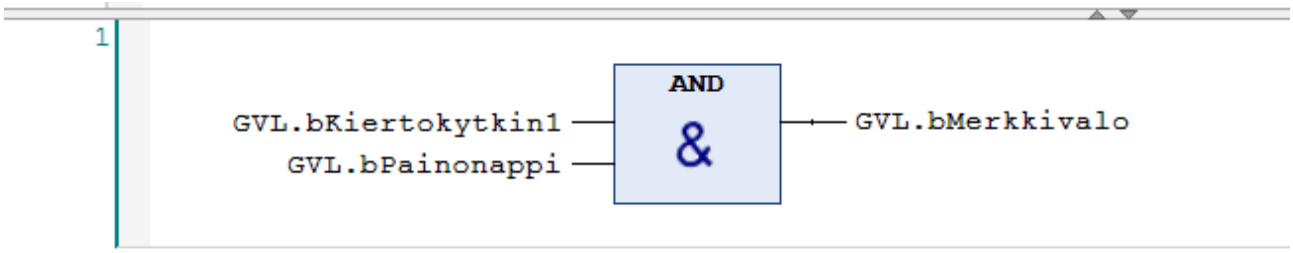
Kuva 4.5.6 AND-lohko johon on asetettu muuttujat

AND-lohkon lähdöllä saadaan määritettyä **bMerkkivalo**-muuttujan tila sijoittamalla **Assignment**-lohko **AND**-lohkon lähtöön.



Kuva 4.5.7 Assignment-lohkolla voidaan asettaa muuttujan tila

Tämän jälkeen voidaan muuttuja, johon lähdön tila sijoitetaan, asettaa normaalisti.



Kuva 4.5.8 AND-lohko jonka perässä on Assignment-lohko

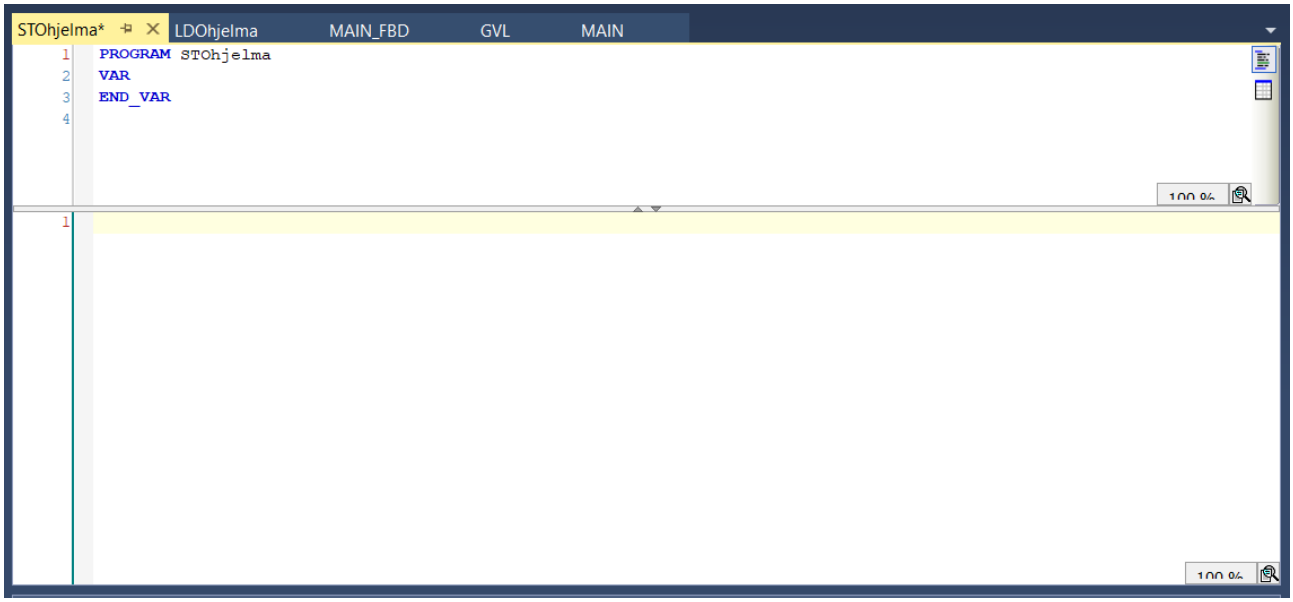
Lisätään vielä MAIN-ohjelmaan komento, jolla FBD-ohjelma suoritetaan.

```
FBDOhjelma();
```

4.5.3 Structured Text -ohjelmointi

ST-ohjelmointimenetelmä sisältää paljon samoja rakenteita kuin korkeamman tason ohjelmointikielet. Esimerkiksi kontrollirakenteet IF, FOR, WHILE ja CASE ovat käytössä myös ST-ohjelmoinnissa.

ST-ohjelma lisätään samoin kuin muutkin ohjelmat, mutta valitaan tyypiksi **Structured Text**. Ohjelman editorina on tekstialue johon ohjelmakoodin voi kirjoittaa.



Kuva 4.5.9 ST-ohjelman editori ennen ohjelman tekoa

Merkkivalon sytyttäminen kiertokytkimen tilan ja painonapin avulla toteutettaisiin ST-ohjelmoinnissa muuttujaan sijoituksella ja AND-operaattorilla.

```
GVL.bMerkkivalo := GVL.bNappi AND GVL.bKytkin;
```

MAIN-ohjelmaan lisätään vielä komento, jolla ohjelma ajetaan.

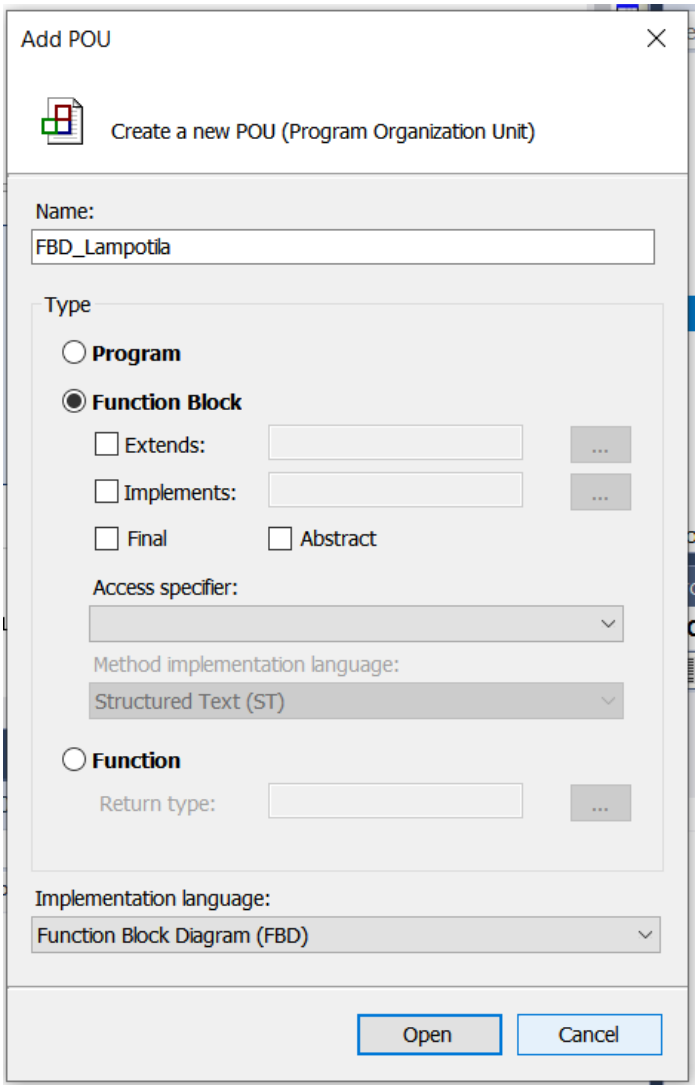
```
STOhjelma();
```

Looppien (FOR, WHILE, REPEAT) käytössä on huomioitava, että PLC ohjelma itsessään on jo eräänlainen looppi, jota ajetaan asetuksiin määritetyllä skannausnopeudella. Looppaaminen odottaen jonkin tulon arvon muuttumista ei myöskään toimi, koska tällöin ohjelma jumiutuu looppin.

4.5.4 Omien funktiolohkojen teko ja käyttö

TwinCAT kehitysympäristössä on PLC-ohjelmien lisäksi mahdollista tehdä myös kokonaan omia funktiolohkoja, joita voi sen jälkeen käyttää osana ohjelmia. Funktiolohkojen tekeminen on hyvä tapa paloitella ohjelmat pienemmiksi ja yksinkertaisemmiksi palasiksi, jotta niiden toimintaa on helpompi ymmärtää.

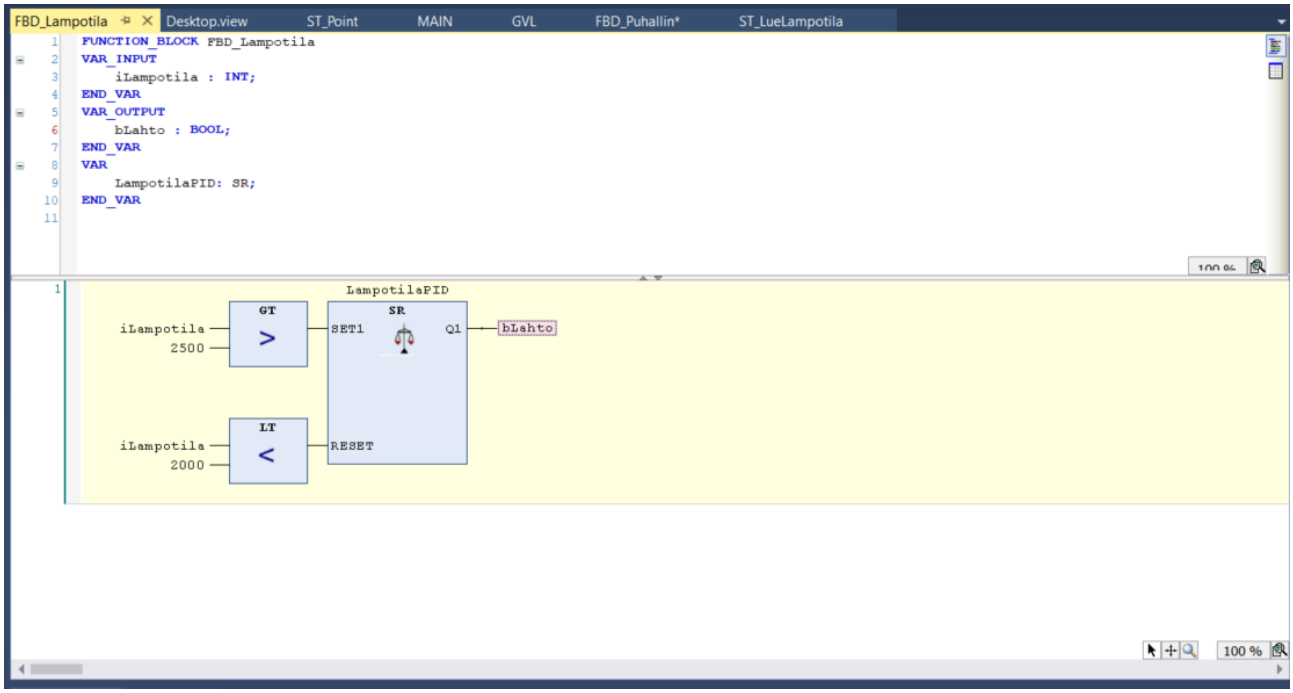
Funktiolohkoja lisätään samaan tapaan kuin uusia ohjelmiakin, mutta valitaan tyypiksi **Function Block**.



Kuva 4.5.10 Funktiolohko tehdään samasta ikkunasta kuin ohjelma

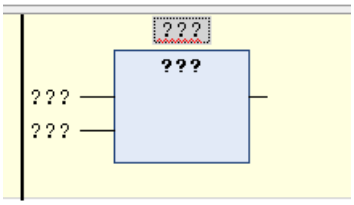
Tämän jälkeen määritetään lohkon tulot ja lähdöt muuttujina ja tehdään lohkon toiminnallisuus samaan tapaan kuin minkä tahansa ohjelman.

Esimerkkinä ohjelma, joka lukee lämpötila tietoa **INT** tyyppiseen muuttujaan **iLampotila**, ja asettaa **BOOL** tyyppisen lähdön tilan. Tässä lähtö asetetaan arvoon **TRUE** jos lämpötila nousee yli 25 asteen ja arvoon **FALSE**, kun lämpötila on jälleen laskenut alle 20 asteen.

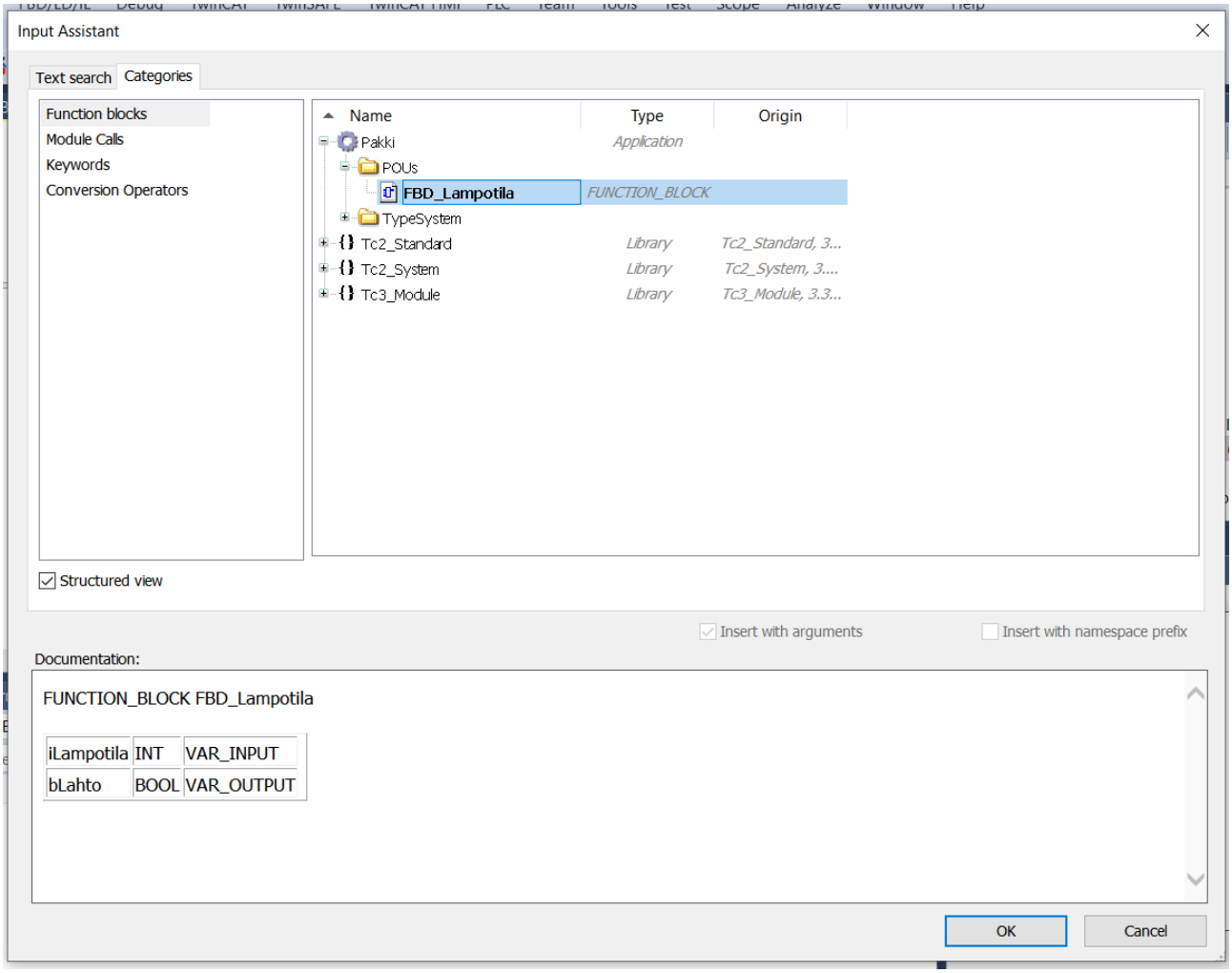


Kuva 4.5.11 Oman funktiolohkon käyttö ohjelmassa

Uutta funktiolohkoa käytetään esim. FDB-ohjelmoinnissa lisäämällä ensin **Box**-tyyppinen funktiolohko, ja valitsemalla sen tyypiksi luotu funktiolohko. Tyypin valinta onnistuu tuplaklikkaamalla **Box**-lohkon yläpuolella olevia kolmea kysymysmerkkiä.

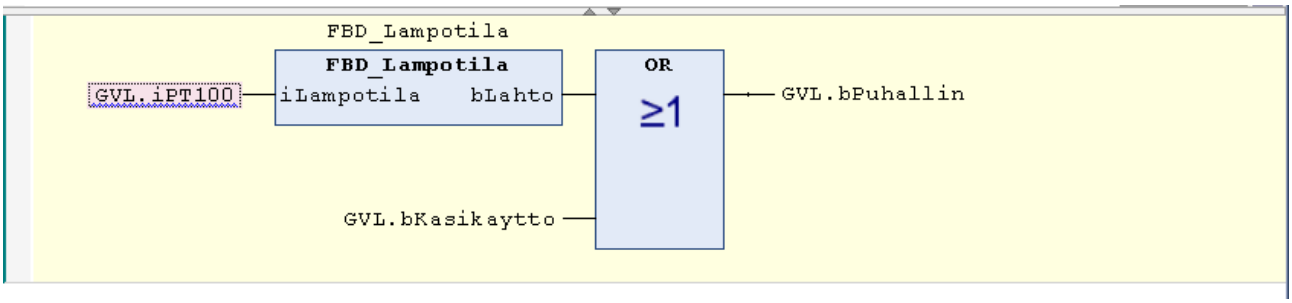


Kuva 4.5.12 Box-lohko ennen sen tyypin valintaa



Kuva 4.5.13 Funktiolohkon tyypin valinta Box-lohkoon

Tämän jälkeen funktiolohkon tulot ja lähdöt täydentyvät ohjelmointinäkymään sitä voidaan käyttää kuten mitä tahansa muuta funktiolohkoa.



Kuva 4.5.14 Oma funktiolohko ohjelmassa

5 Simulointi

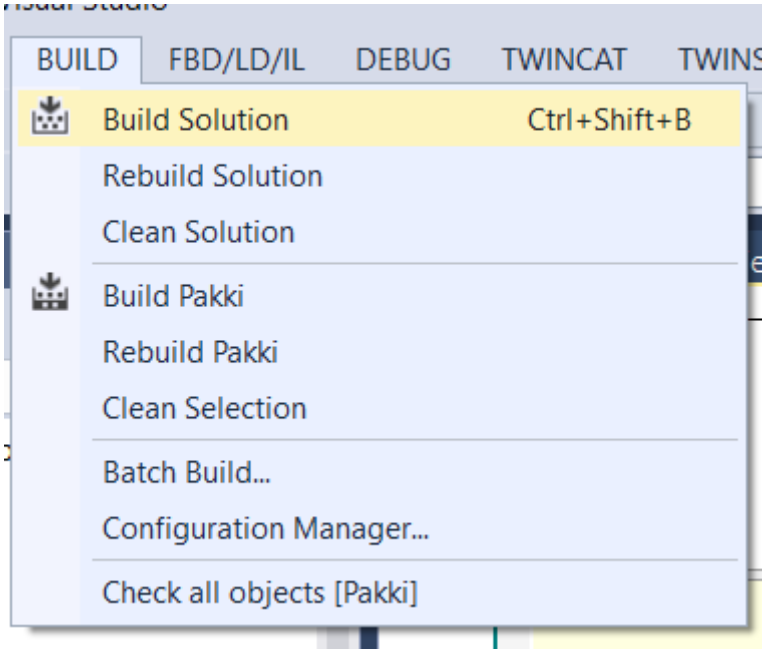
5.1 Projektin simuloiminen TwinCat runtimella

PLC-ohjelmia voidaan testata TwinCatissä simuloimalla logiikka PC:llä. Simuloinnissa PC:llä ajetaan samaa TwinCat Runtime, jota myös logiikat käyttävät, joten ohjelmien tulisi käyttäytyä realistisesti.

Seuraavaksi kuvataan vaihe vaiheelta, miten simulointi tehdään.

1. **Varmista että sovellus kääntyy**

Ennnen simuloinnin yrittämistä kannattaa varmistaa, että sovellus kääntyy ja kaikki on kunnossa simulointia varten. Kääntäminen tapahtuu **Build**-valikon **Build Solution** komennolla.



Kuva 5.1.1 Käännetään projekti jotta varmistutaan että ohjelmat ovat oikein tehdyt

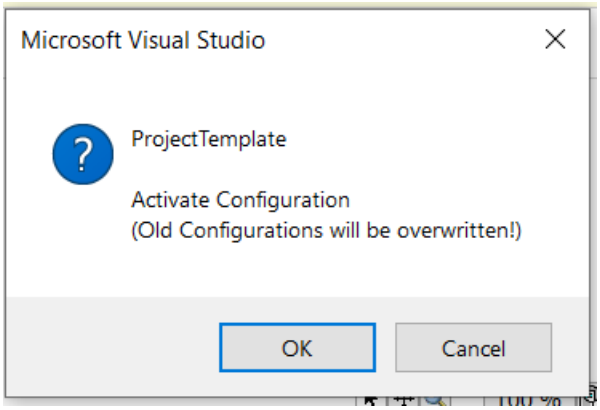
Konsoliin tulostuu tietoja käännöksestä mukaanlukien mahdolliset virheet, jotka on korjattava ennen kuin sovellusta voi simuloida tai ajaa

2. **Aktivoi konfiguraatio**

Onnistuneen kääntämisen jälkeen tämänhetkinen konfiguraatio täytyy aktivoida, jolloin kaikki PLC-projektin tiedostot siirretään kohdelaitteistolle (tässä tapauksessa simuloidulle PLC:lle). Konfiguraation aktivointi tapahtuu työkalupalkin vasemmassa reunassa olevasta napista.

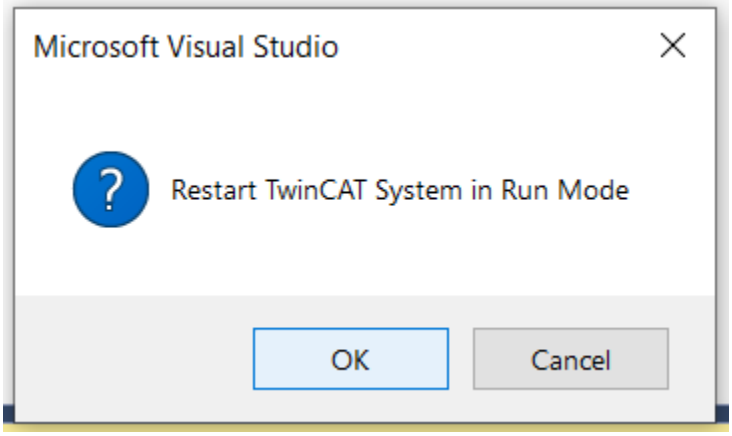
Kuva 5.1.2 Konfiguraation aktivointi tapahtuu sinisiä neliöitä sisältävästä napista

TwinCat kysyy, korvataanko olemassa oleva konfiguraatio.



Kuva 5.1.3 Aktivoidaan uusi konfiguraatio

TwinCat kysyy, siirrytäänkö Run-moodiin vai pysytäänkö Config-moodissa. Vastataan tähän **OK**.

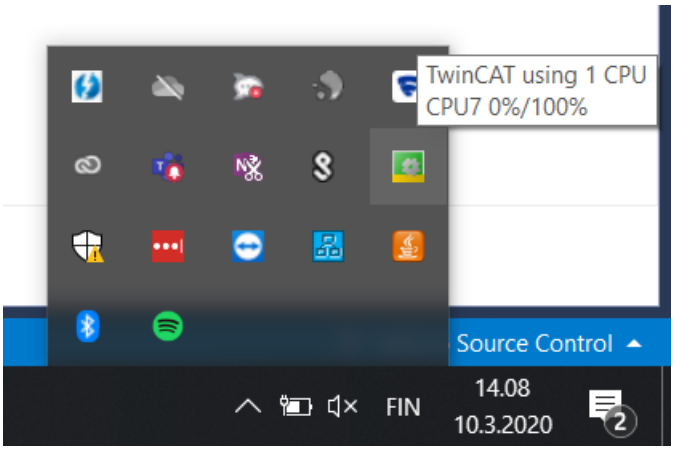


Kuva 5.1.4 Käynnistetään logiikka Run-moodissa

Tämän jälkeen PLC-projekti on kopioitu simuloidulle logiikalle ja on **Run Mode** -tilassa, jos Visual Studion tilarivillä näkyy vihreä ikoni.

Kuva 5.1.5 Vihreä ikoni hammasrattaalla tarkoittaa että PLC on Run Mode -tilassa

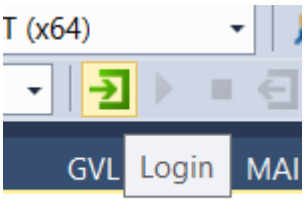
Simuloidun logiikan tila on nähtävissä myös Windowsin alapalkista löytyvän valikon alta.



Kuva 5.1.6 Simuloidun logiikan tila on nähtävissä myös Windowsin työkalupalkin ikonin kautta

3. **Login**

Seuraavaksi suoritetaan **Login** klikkaamalla työkalupalkin ikonia. Loginin jälkeen kehitysympäristö on reaaliaikaisessa yhteydessä logiikkaan.



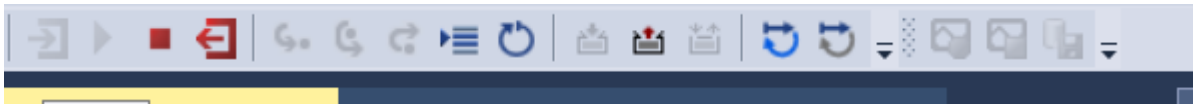
Kuva 5.1.7 Login ikoni työkalupalkissa

TwinCat kysyy, halutaanko luoda ja ladata sovellus logiikalle. Vastataan **Yes**.

4. **Start**

Kun sovellus on ladattu logiikalle, klikataan työkalupalkin vihreää "play"-ikonia, jolloin sovellus käynnistetään logiikalla.

Kuva 5.1.8 Ohjelman ajamiseen liittyvä työkalurivi



Kuva 5.1.9 Käynnissä olevan ohjelman hallintaan on työkalurivissä eri toimintoja

5.2 Ohjelman debuggaus ajon aikana

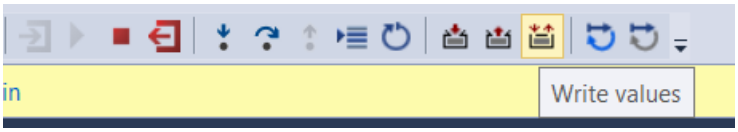
Kun ohjelma on käynnissä PLC:llä, voidaan sen toimintaa tutkia ja muokata ajon aikana. Seuraavaksi käydään läpi muuttujien arvon muuttaminen ajona aikana sekä ohjelman muuttaminen ja päivittäminen laitteelle ***lennosta***.

Kun ohjelma on käynnissä PLC:llä voidaan muuttujien ja ohjelmien tilaa tarkastella avaamalla jokin ohjelma. Muuttujien arvoja voidaan muuttaa tuplaklikkaamalla ohjelman muokkausnäkymässä jotain muuttujaa. BOOL tyyppiset muuttujat vaihtavat arvonsa vastakkaiseksi ja esim. numeromuuttujien arvon pystyy kirjoittamaan kenttään.

Kokeillaan muuttaa edellisessä luvussa tehdyn LD-ohjelman sisältämien muuttujien arvoja. Kun muuttujaa kaksoisnapsautetaan, ilmestyy sen viereen teksti **<TRUE>** tai **<FALSE>** riippuen siitä, mikä muuttujan arvo oli aikaisemmin.

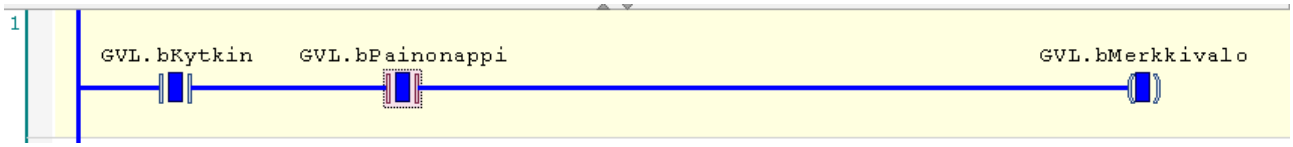
Kuva 5.2.1 Muuttujan arvon muuttaminen

Muuttujan arvon muutos ei kuitenkaan ole vielä voimassa, vaan muutetut arvon täytyy ensin kirjoittaa PLC:n muistiin. Klikataan **Write values**-kuvaketta työkalupalkin oikeasta reunasta.



Kuva 5.2.2 Write Values kirjoittaa muutokset PLC:n muistiin

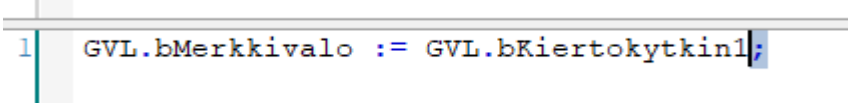
Kun sama on tehty ohjelman molemmille tulo-muuttujille (**bKytkin** ja **bPainonappi**), muuttuu lähtö **bMerkkivalo** arvoon **TRUE** ja merkkivalo syttyy.



Kuva 5.2.3 Kun bKytkin ja bPainonappi ovat TRUE, saa myös bMerkkivalo arvon TRUE

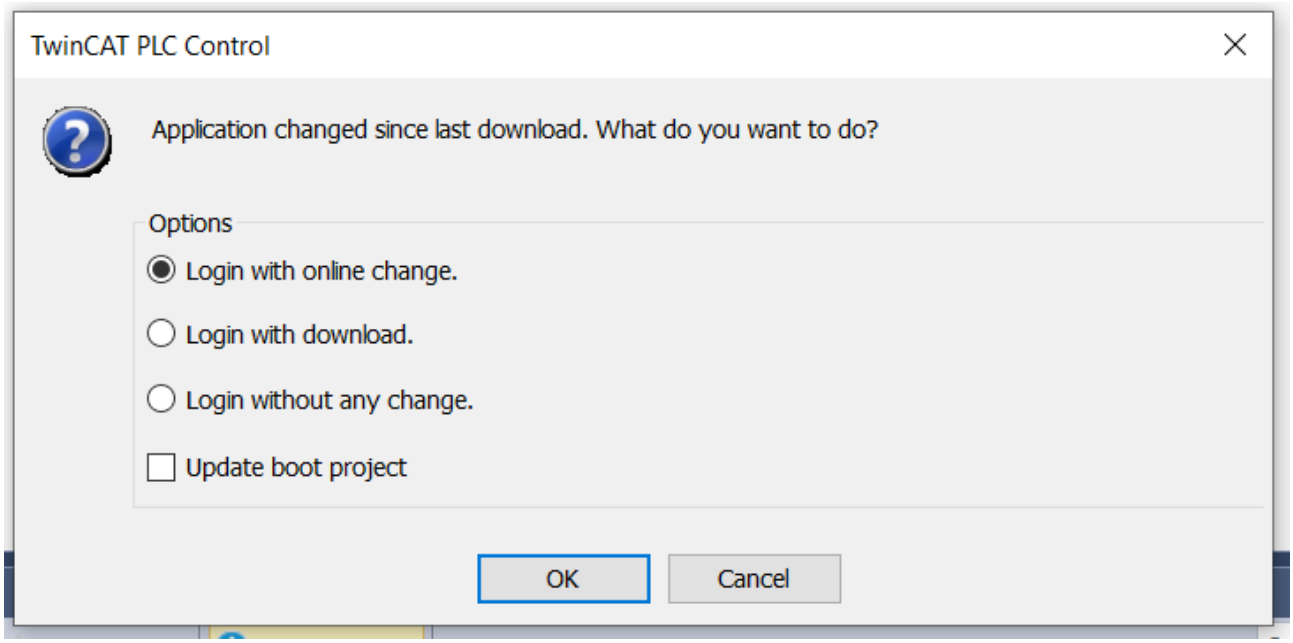
Myös itse käynnissä olevaa ohjelmaa voidaan muokata ilman että PLC:tä täytyy käynnistää uudelleen tai konfiguraatiota tarvitsee ladata uudestaan.

Kokeillaan käynnissä olevan ohjelman muuttamista muokkaamalla edellisessä luvussa tehtyä ST-ohjelmaa. Ennen muutosten tekoa täytyy TwinCat ***kirjata ulos*** PLC:stä työkalupalkin **Logout** kuvaketta klikkaamalla.



Kuva 5.2.4 Muokattu ST-ohjelma

Tämän jälkeen muokataan ohjelmaa siten että merkkivalon sytyttämiseen vaaditaan vain, että kiertokytkin on asennossa 1.



Kuva 5.2.5 TwinCAT kysyy halutaanko muutokset ladata laitteelle

Kun muutos on tehty, klikataan jälleen **Login** kuvaketta, jolloin TwinCAT kysyy, halutaanko kirjautuminen tehdä lataamalla muutokset laitteelle (**Login with online change**), lataamalla PLC:llä oleva ohjelma TwinCatiin (**Login with download**) vai ilman muutoksia (**Login without any change**). Valitaan **Login with online change**, tällöin ohjelmaan tehdyt muutokset kopioidaan PLC:lle.

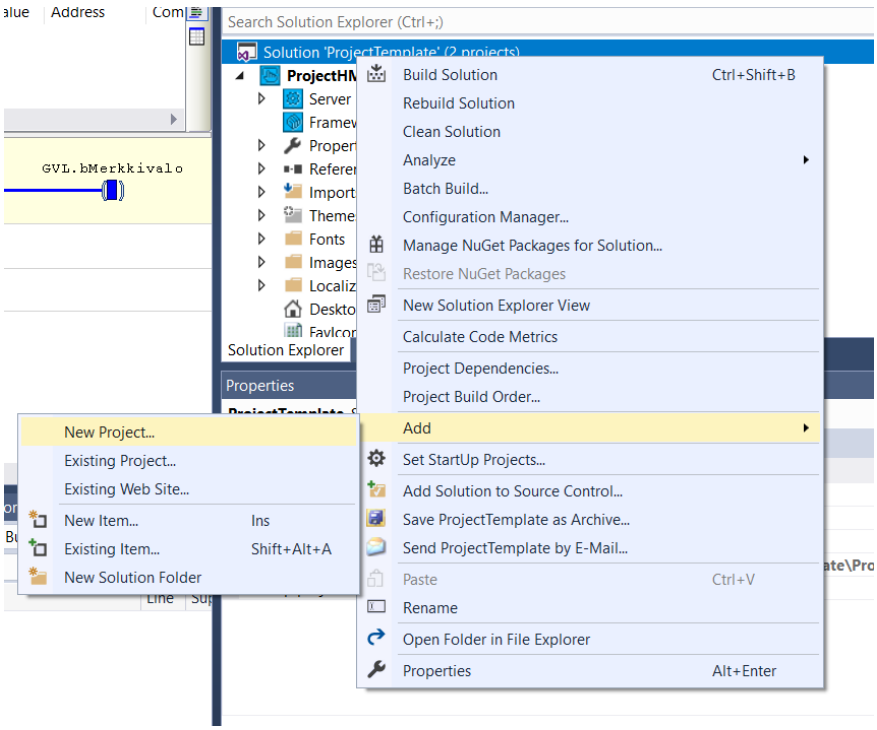
6 TwinCat 3 HMI

TwinCat 3 HMI on Beckhoffin työkalu modernien PLC-käyttöliittymien luomiseen. HMI:llä tehdyt käyttöliittymät ovat HTML5-pohjaisia ja niitä voi käyttää normaalilla Internet-selaimella. Käyttöliittymää varten luodaan oma HMI projektinsa, joka voidaan halutessa luoda PLC-projektin rinnalle tai omaan kansioonsa.

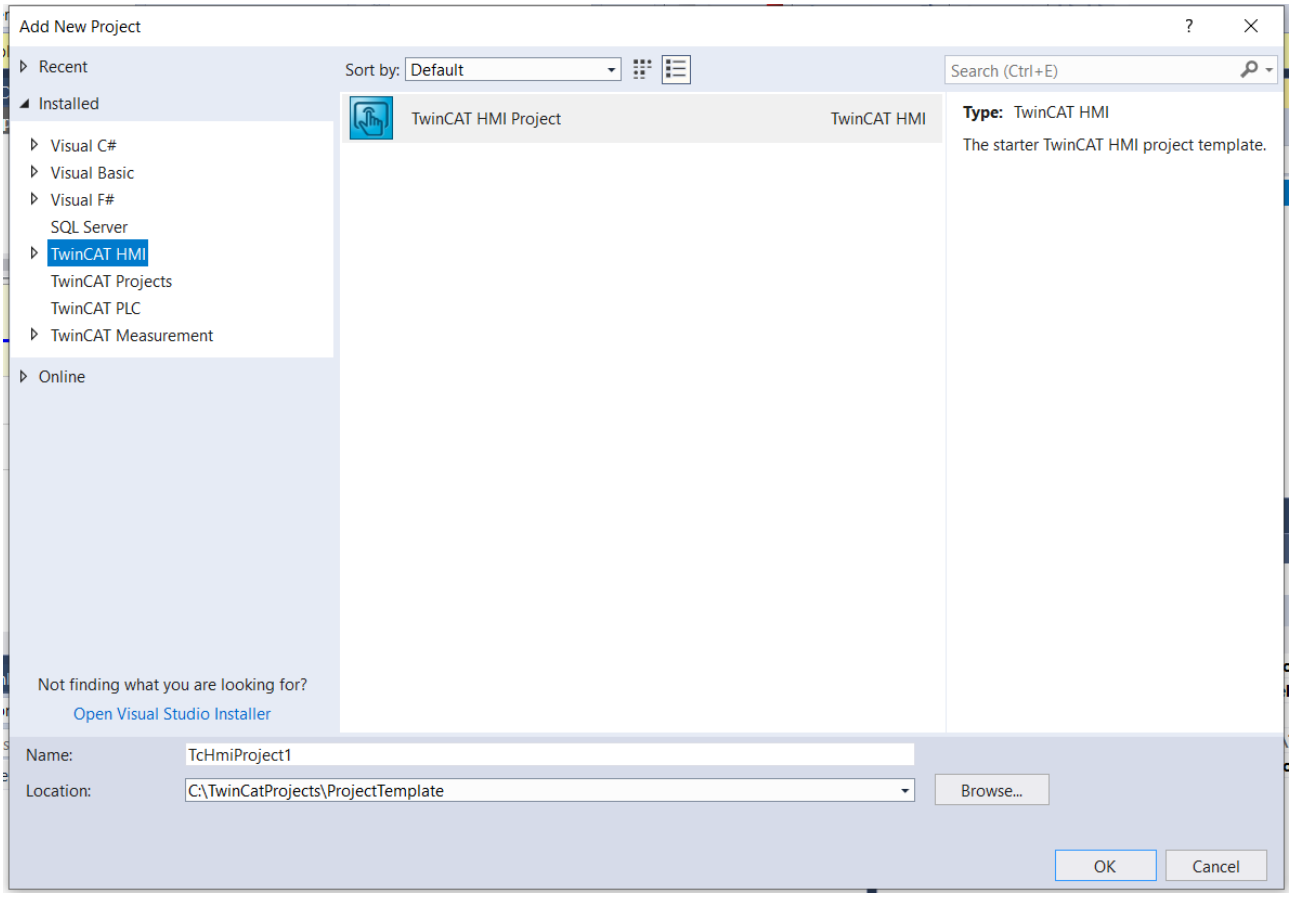
Ennen TwinCAT HMI projektin luomista täytyy TwinCAT 3 HMI Engineering komponentti olla asennettuna.

6.1 HMI projektin luominen ja muokkaaminen

Avoinna olevaan TwinCat projektiin (solution) voidaan lisätä HMI-projekti klikkaamalla oikealla ja valitsemalla **Add→ New Project**. Avautuvasta ikkunasta valitaan TwinCat HMI ja HMI Project.



Kuva 6.1.1 Uuden HMI projektin lisäys



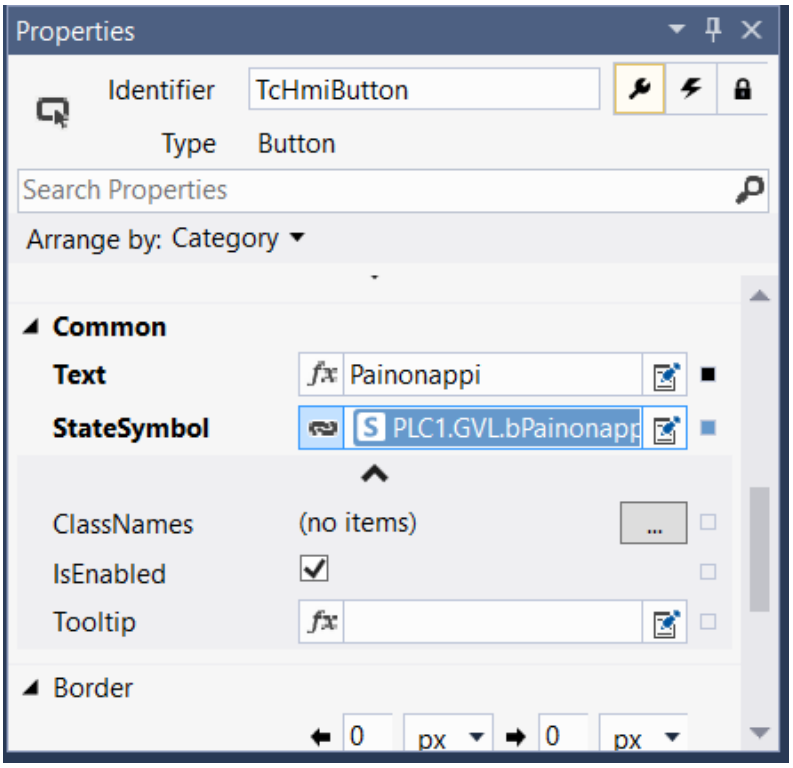
Kuva 6.1.2 Valitaan projektin tyypiksi TwinCAT HMI Project

Käyttöliittymä voidaan muokata joko käyttäen ns. WYSIWYG (What You See Is What You Get) editoria tai suoraan muokkaamalla HTML koodia. Käyttöliittymän luominen peruskomponenteilla on helppoa ja seuraavaksi käymme läpi käyttöliittymän luomisen edellä tehdyille PLC-ohjelmille.

6.2 Käyttöliittymän komponentit

Käyttöliittymän peruskomponentit löytyvät **Toolbox** ikkunasta (ikkunan saa tarvittaessa näkyviin **View** valikon alta). TwinCat HMI sisältää noin parikymmentä komponenttia yksinkertaisista tekstikentistä aina monimutkaisempiin datan visualisointikomponentteihin asti.

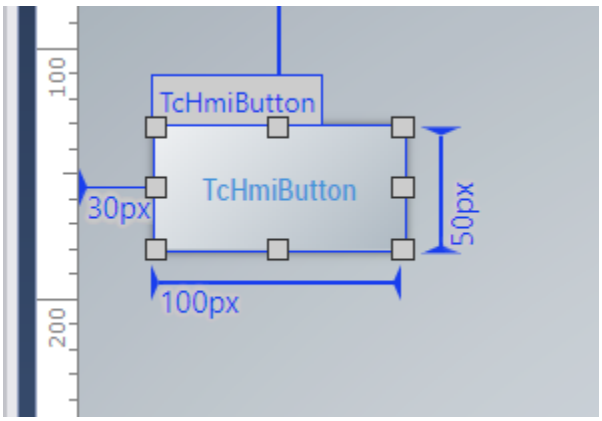
Kun jokin komponentti on valittuna voidaan sen ominaisuuksia muuttaa **Properties** ikkunan alta. **Properties** ikkunan **Events** välilehdeltä (salamaikoni) voidaan PLC-ohjelman muuttujat linkittää käyttöliittymään.



Kuva 6.2.1 Painonapin properties ikkuna

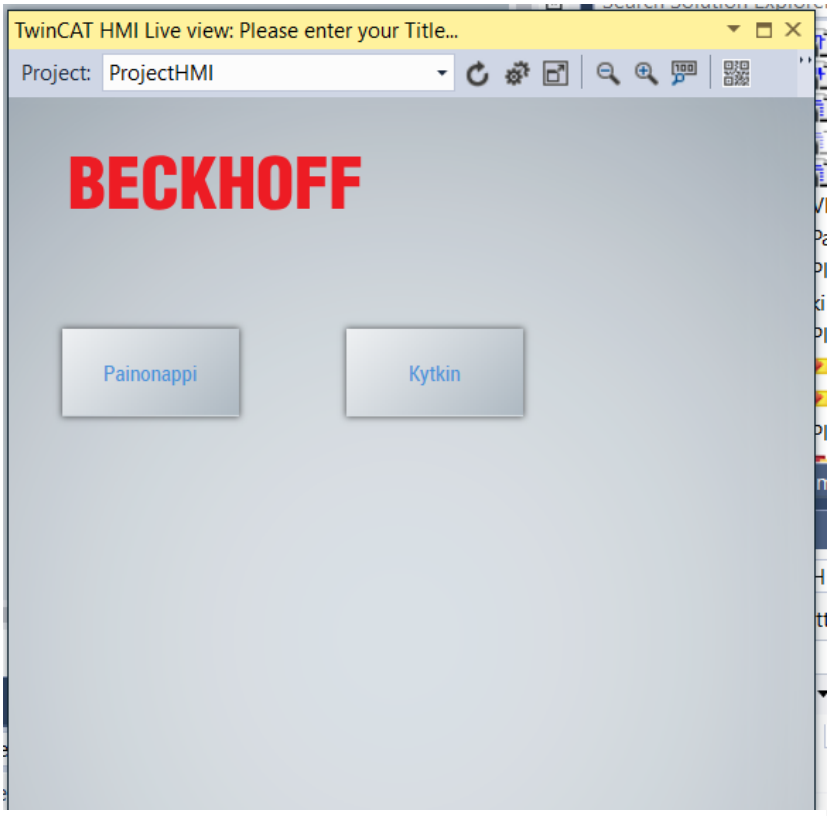
Seuraavaksi käsitellään peruskomponenttien **Button**, **ToggleButton** ja **Ellipse** käyttämistä edellisessä luvussa tehtyjen ohjelmien käyttöliittymän tekemisessä.

Ensimmäisenä lisätään painonappi, jota käytetään **bPainonappi** muuttujan arvon asettamiseen. Nappi lisätään raahaamalla se **ToolBox**-ikkunasta käyttöliittymän muokkausnäkymään. Elementtien tarkka asemointi on helppoa editorin apuviivojen avulla.



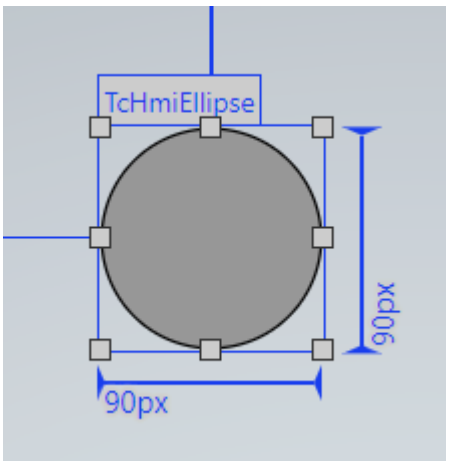
Kuva 6.2.2 Button elementin asemointi

Seuraavaksi lisätään käyttöliittymään **ToggleButton,** jolla asetetaan **bKytkin** muuttujan arvo.



Kuva 6.2.3 Button ja ToggleButton elementit

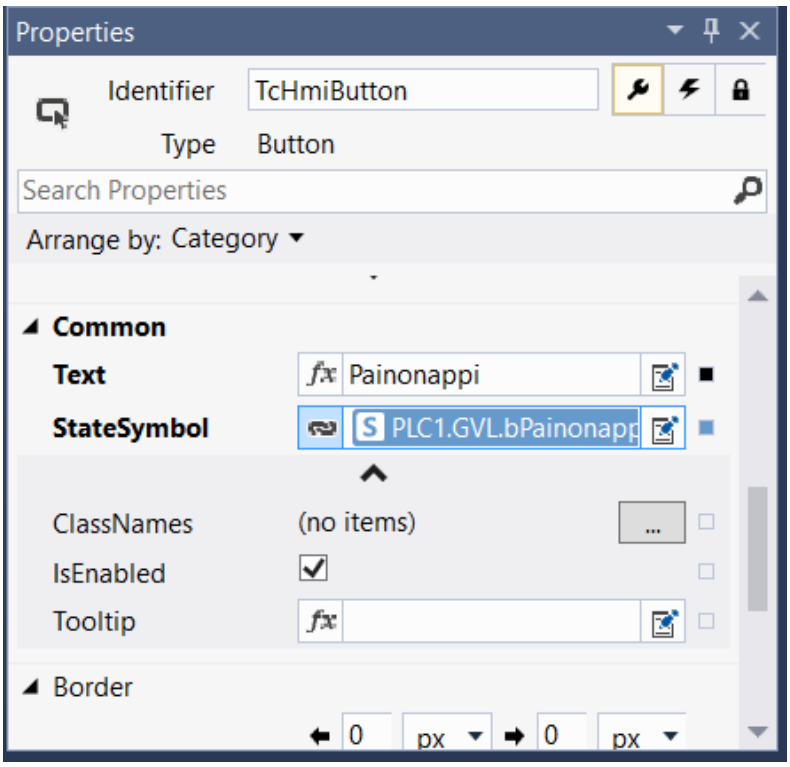
Viimeisenä käyttöliittymään lisätään **Ellipse**, eli ellipsi. Ellipsin täyttöväriä voidaan käyttää **bMerk-kilamppu**-muuttujan arvon indikoimiseen.



Kuva 6.2.4 Ellipse elementti

6.3 Linkitys PLC-projektin muuttujiin

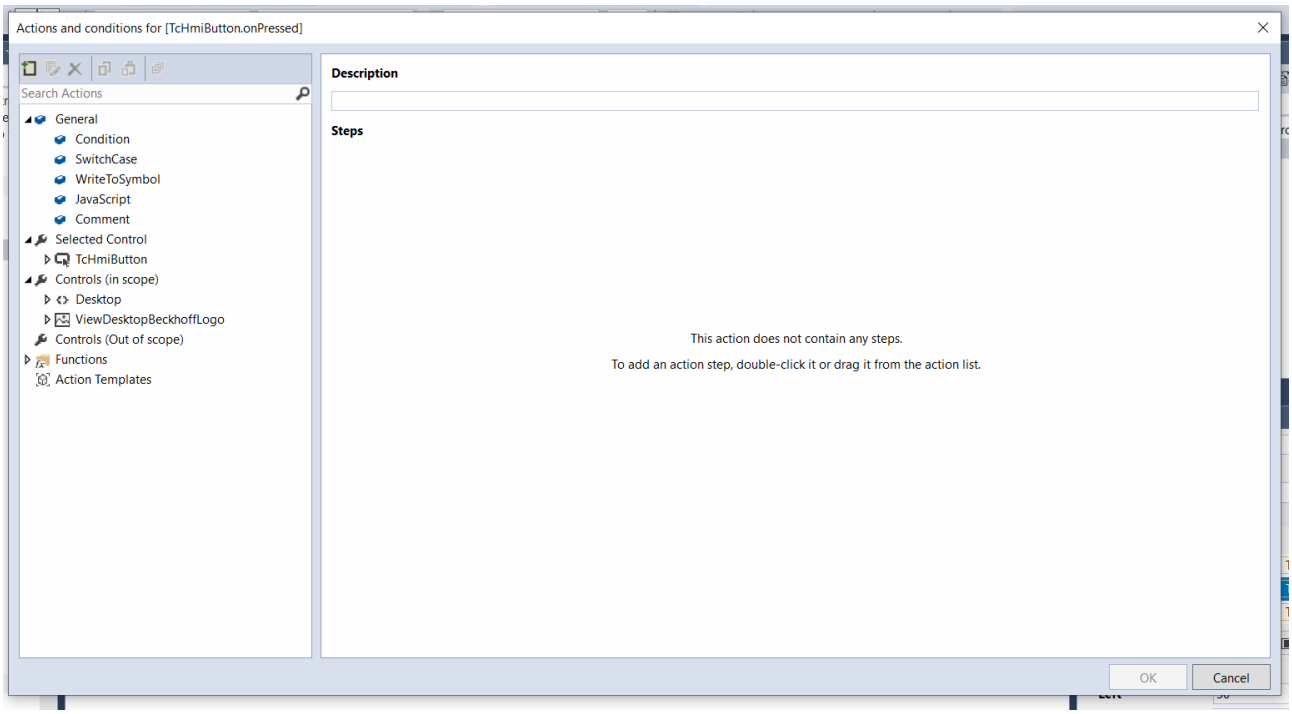
Nappi linkitetään PLC-ohjelman muuttujaan **Properties**-ikkunan olevan kentän **StateSymbol** avulla. Kentän arvo on tosi, aina kun nappi on alas painettuna, ja epätosi muulloin. Kun **StateSymbol** linki-



Kuva 6.3.1 Napin linkitys StateSymbol kentän avulla

tetään PLC-projektin globaaliin **bPainonappi** muuttujaan, vaikuttaa napin painaminen PLC-ohjelmien toimintaan.

Seuraavaksi linkitetään ToggleButton muuttujaan **bKytkin**. Tämä tapahtuu kaksoisnapsauttamalla nappia, jolloin avautuu ***Actions and conditions for [TcHmiToggleButton.onPressed]***-ikkuna.

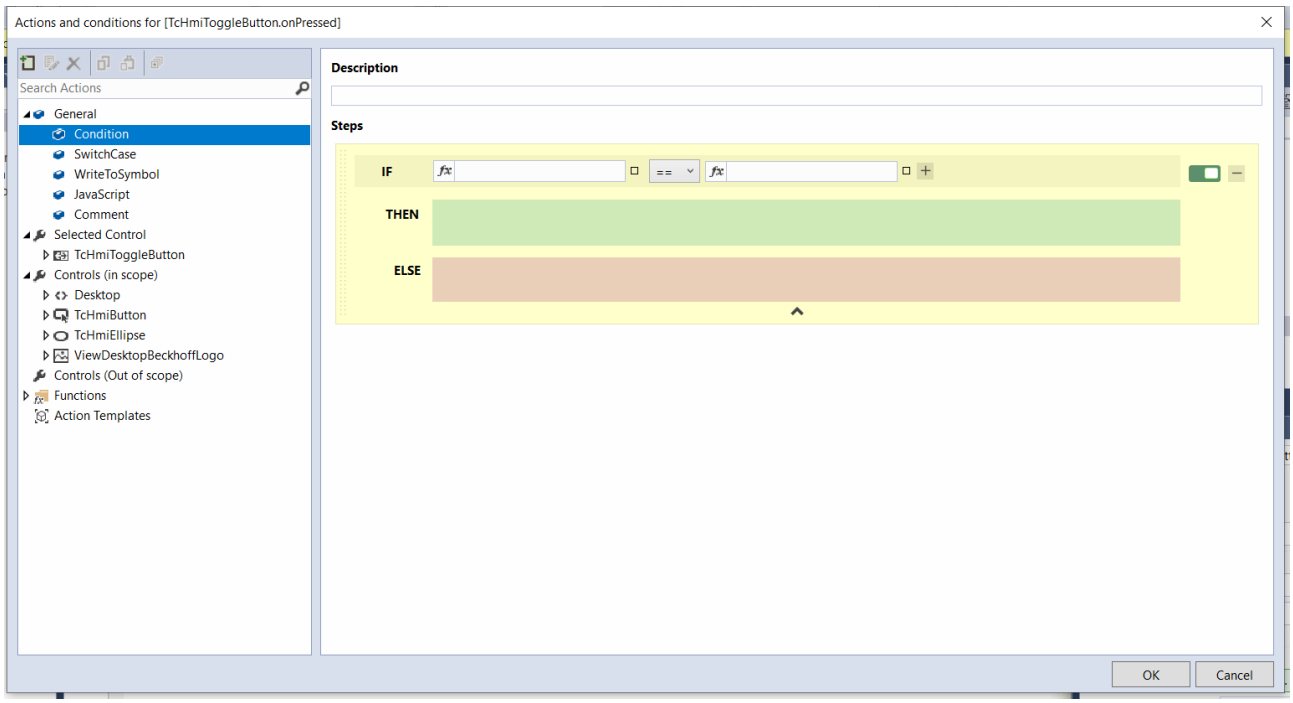


Kuva 6.3.2 Actions and conditions ikkuna

Actions and Conditions-ikkunasta voidaa käyttöliittymäkontrolleille lisätä toiminnallisuuksia, jotka suoritetaan jonkin eventin yhteydessä. Tässä tapauksessa eventti on **onPressed**, eli kun nappia painetaan.

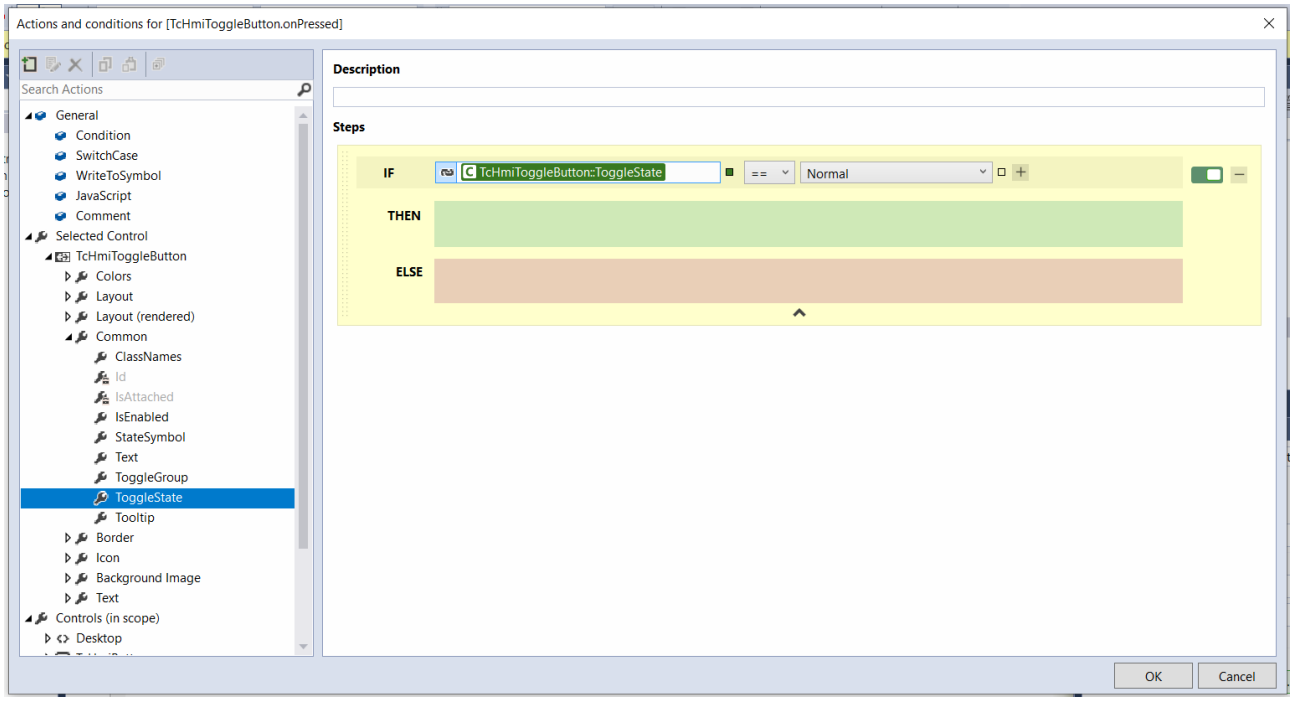
Ikkunan vasemmassa laidassa on **Action**-steppejä, joita voidaan lisätä suoritettavaksi. Lista sisältää myös muuttujia ja kontrolleja, joita voidaan käyttää actioneissa. ToggleButton-kontrollilla on arvo **ToggleState**, jota voimme käyttää **bKytkin**-muuttujan arvon asettamiseen. Sen arvo voi olla joko **Normal** tai **Active**.

Lisätään ensin ehto, jonka perusteella tila asetetaan. Valitaan vasemmasta valikosta **Condition** ja raahataan se oikealla olevaan tyhjään tilaan. **Condition**-actioni luo if-ehdon, jolla voidaan tarkistaa muuttujan arvo ja sen jälkeen suorittaa toisia actioneita arvosta riippuen.



Kuva 6.3.3 Condition ehto lisättynä

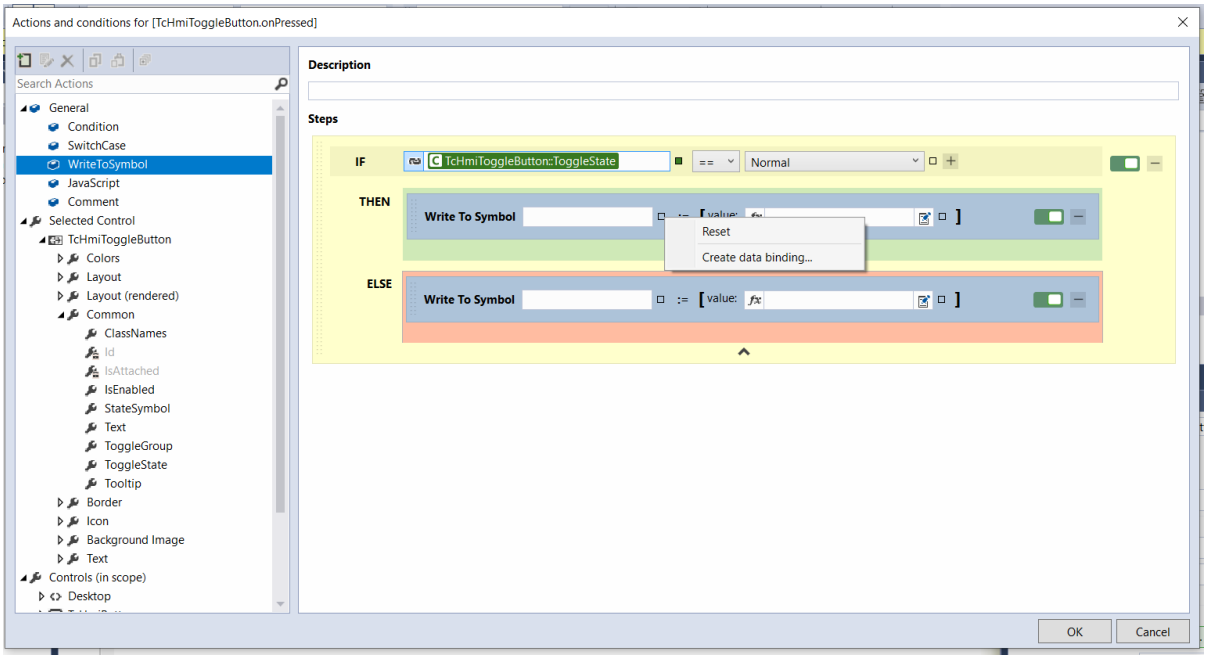
Tämän jälkeen valitaan vasemmasta listasta ToggleButtonin **ToggleState** arvo, ja raahataan se IF-ehdon ensimmäiseen kenttään. Kolmanteen kenttään valitaan arvoksi **Normal**



Kuva 6.3.4 Napin ToggleState arvo IF ehdossa

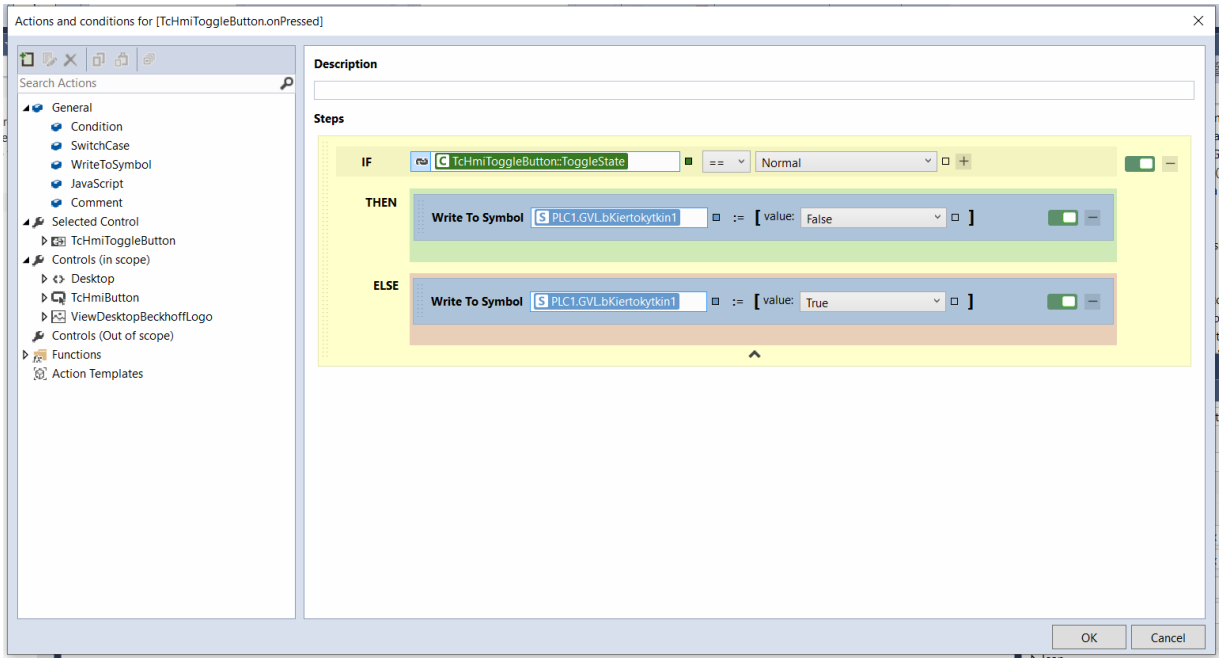
Nyt voidaan määrittää mitä tapahtuu, kun käyttäjä painaa **ToggleButton**-nappia ja napin tila on joko **Normal** tai **Active**. Raahataan vasemmasta valikosta **WriteToSymbol** sekä IF-ehdon THEN-kenttään, että ELSE-lenttään.

Tehdään molempien ***Write To Symbol***-actionien ensimmäiseen kenttään linkitys **bKytkin** muuttujaan klikkaamalla kentän vieressä olevaa neliötä, valitsemalla ***Create data binding*** ja etsimällä muuttuja.



Kuva 6.3.5 Linkitetään muuttuja Create data binding toiminnolla

Tämän jälkeen valitaan ELSE-ehdon ***Write to Symbol***-actionin toiseen kenttään (value) arvoksi

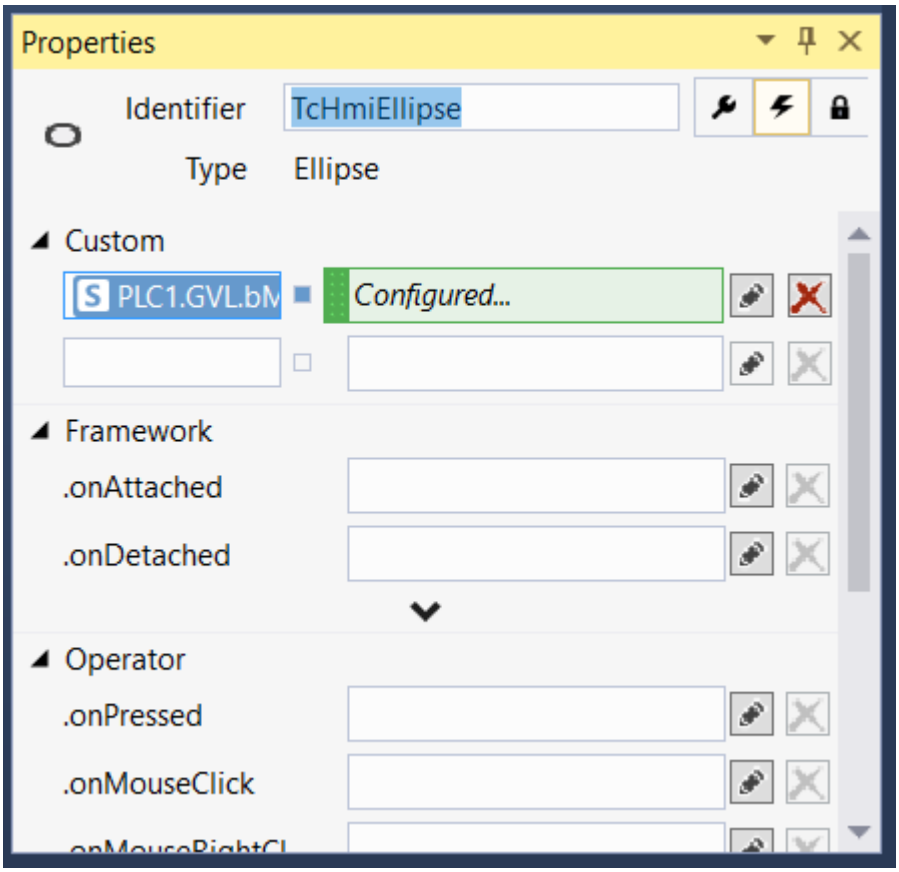


Kuva 6.3.6 Kytkimen actionit määritettyinä

True

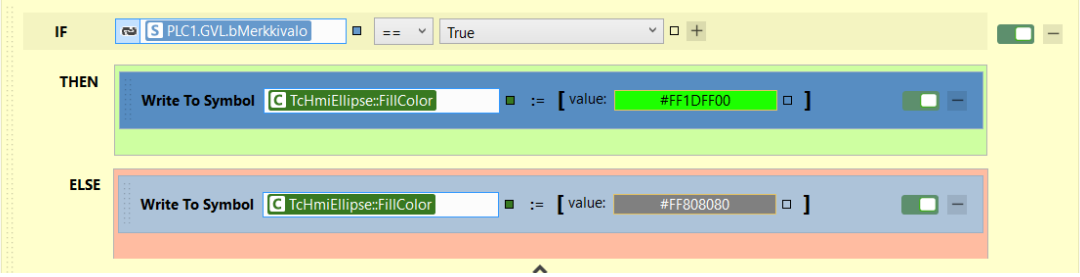
Viimeisenä tehdään linkitys **bMerkkivalo** muuttujan ja ellipsin täyttövärin välille. Ellipsin tapauksessa linkitys PLC-projektin muuttujaan on erilainen, koska sen sijaan että käyttöliittymästä muutettaisiin muuttujan tilaa, muutetaan käyttöliittymää, kun muuttujan arvo muuttuu.

Ensimmäisenä täytyy lisätä ellipsin ***Properties***-ikkunan ***Show Events***-välilehdelle linkitys **bMerkkivalo**-muuttujaan. ***Show Events***-välilehti löytyy salamaikonin takaa. Linkitys tehdään kohtaan ***Custom*** valitsemalla ***Create data binding*** ensimmäisen kentän vieressä olevasta neliöstä aukeavasta valikosta.



Kuva 6.3.7 Ellipsin linkitys bMerkkivalo muuttujaan

Nyt voidaan muuttujan arvon muuttumiseen lisätä toiminto. Se tapahtuu avaamalla ***Actions and Conditions***-ikkuna kynäikonista. Toiminnot määritellään samaan tapaan kuin ToggeButton-napin tapauksessa mutta nyt muutetaankin **TciHmiEllipse::FillColor**-arvoa.

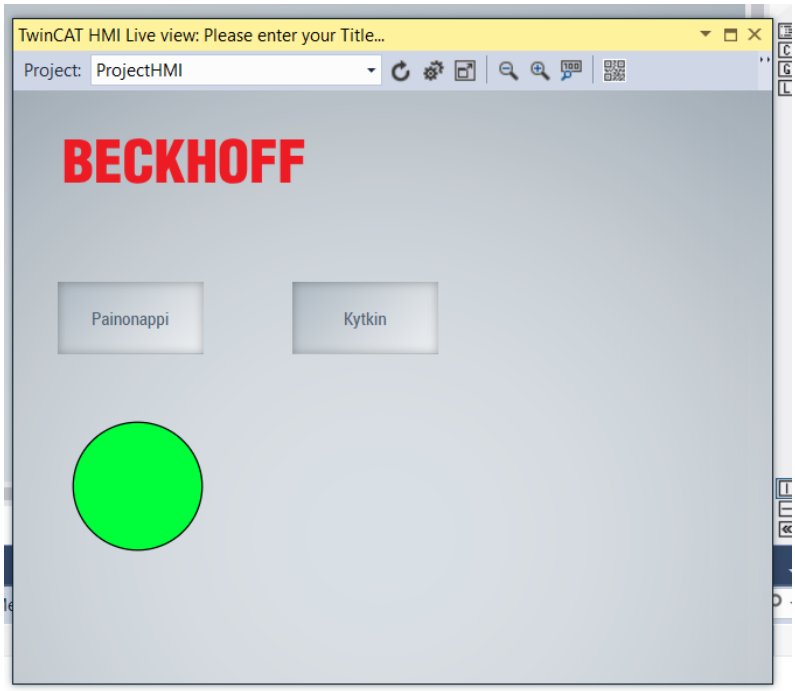


Kuva 6.3.8 Ellipsin värin muuttaminen bMerkkivalo muuttujan arvon mukaan

6.4 Käyttöliittymän testaus

Käyttöliittymän testaus onnistuu **Live View** ominaisuuden avulla. Ensin varmistetaan, että PLC-projekti on käynnissä kohdan ***Simulointi*** ohjeiden mukaan. Tämän jälkeen klikataan **L**-ikonia käyttöliittymäeditorin oikeassa laidassa.

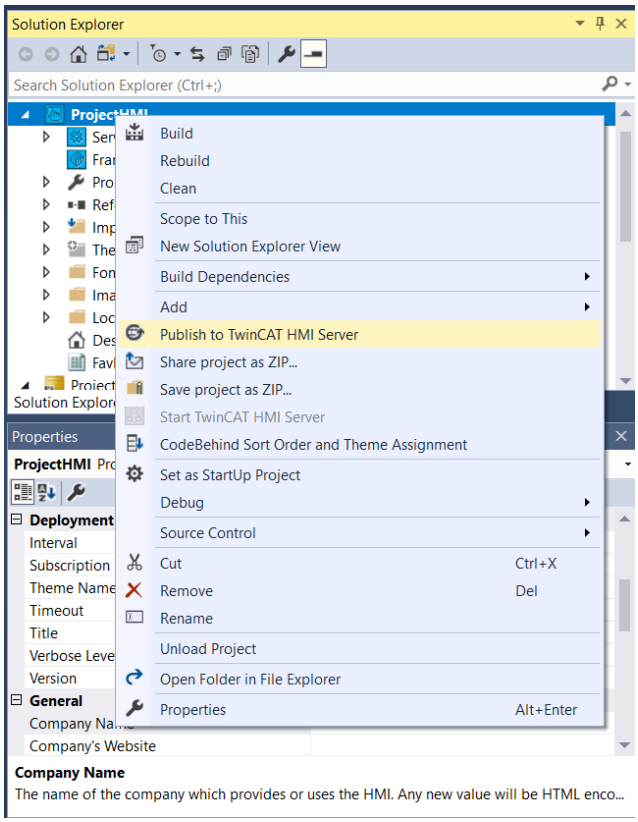
Käyttöliittymä aukeaa uuteen ikkunaan jossa sitä voidaan testata. Kun **Kytkin**-nappi painetaan ensin ala-asentoon, ja sen jälkeen painetaan **Painonappi**-nappia, muuttuu ellipsin väri vihreäksi.



Kuva 6.4.1 Käyttöliittymän testaus käynnissä olevan PLC-ohjelman kanssa

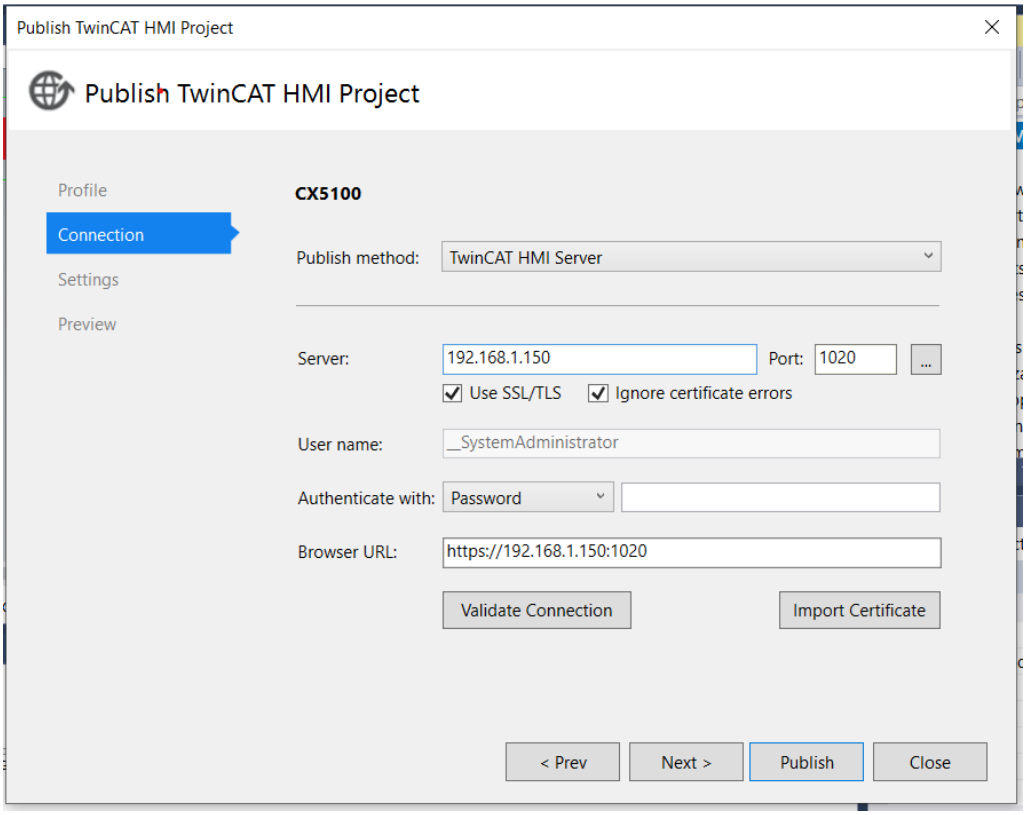
6.5 Käyttöliittymän julkaisu

TwinCAT HMI käyttöliittymän voi julkaista, eli siirtää kohdelaitteistolle, valitsemalla ***Publish to TwinCAT HMI Server*** HMI projektin valikosta.



Kuva 6.5.1 Käyttöliittymän julkaisu

Kun julkaisua tehdään ensimmäistä kertaa, tehdään julkaisuprofiili, joka sisältää julkaisuun tarvittavat asetukset. Profiiliin tallennetaan kohdelaitteen IP-osoite ja portti, jotka voidaan syöttää joko manuaalisesti tai hakea automaattisesti, jos kohdelaite on lähiverkossa. IP-osoitteen ja portin lisäksi profiiliin voidaan tarvittaessa määrittää käyttäjänimi ja salasana sekä HTTPS-asetukset.



Kuva 6.5.2 Käyttöliittymän julkaisun asetukset

Kun profiili on luotu, voidaan käyttöliittymä julkaista HMI palvelimelle klikkaamalla ***Publish***.

7 Testaus PLC-laitteistolla

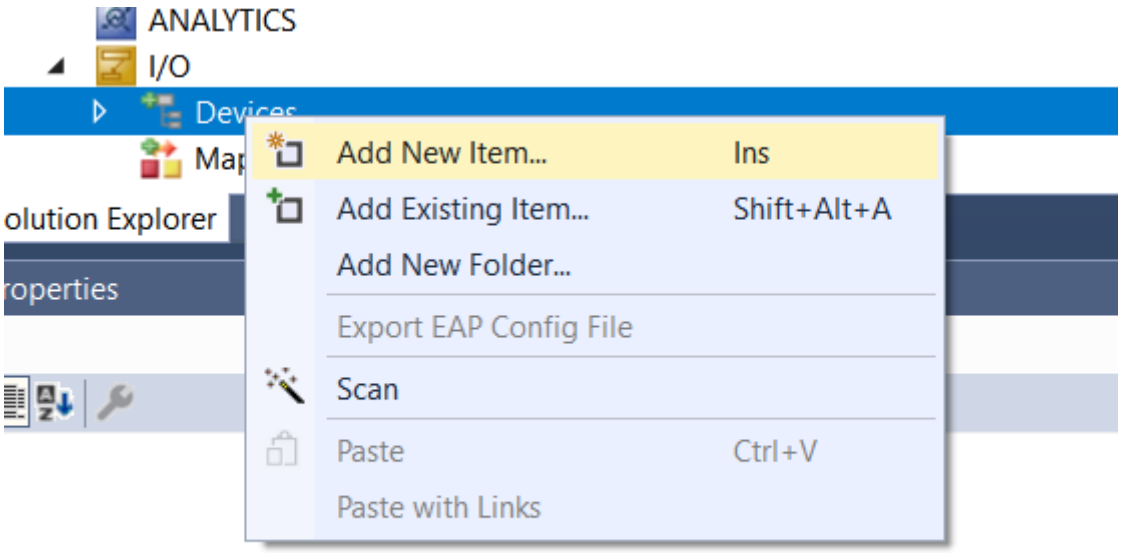
Kun PLC-projektia halutaan testata varsinaisella laitteistolla, tai kun projekti halutaan viedä tuotantojärjestelmään, täytyy laitteisto ensin lisätä TwinCAT projektiin. Laitteiston voi lisätä joko manuaalisesti tai skannaamalla.

7.1 Laitteiston lisäys manuaalisesti TwinCAT projektiin

7.1.1 EtherCAT master laitteen lisääminen

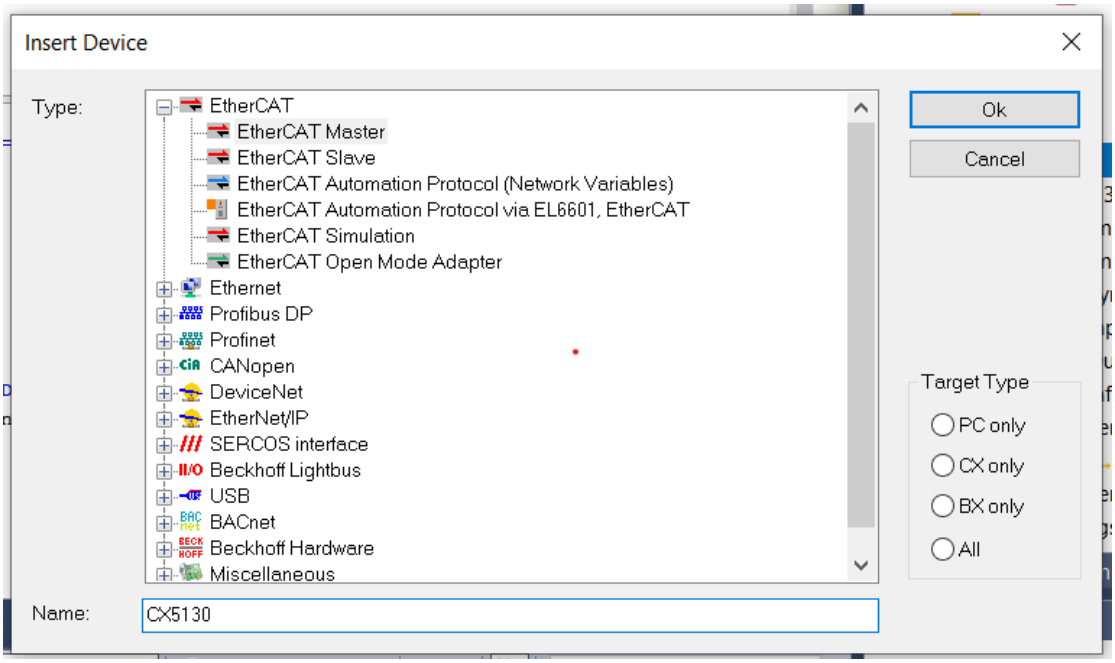
Ensimmäisenä tulee laitteistoon lisätä EtherCAT master, jonka alle muut laitteet lisätään. EtherCAT masterin voidaan ajatella vastaavan järjestelmän PLC-laitetta.

Aloitetaan klikkaamalla projektipuussa I/O osion alla olevaan **Device** kohtaa hiiren oikealla, ja valitsemalla **Add New Item**.



Kuva 7.1.1 Lisätään uusi laite

Avautauvasta ikkunasta valitaan **EtherCAT master** ja annetaan laitteelle sopiva nimi. Tässä tapauksessa nimeksi annettiin CX5130.

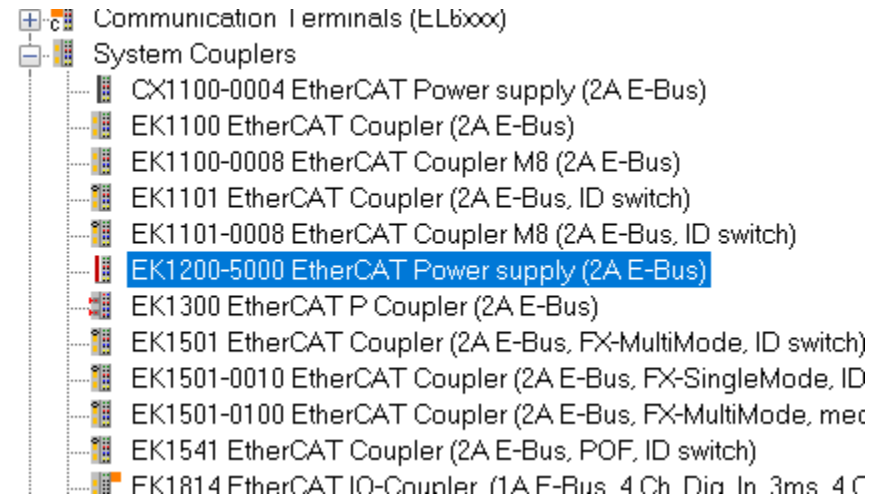


Kuva 7.1.2 Valitaan laitteen tyyppi ja annetaan sille nimi

Tämän jälkeen aukeaa ikkuna **Device Found At**, johon voidaan vastata **Cancel**

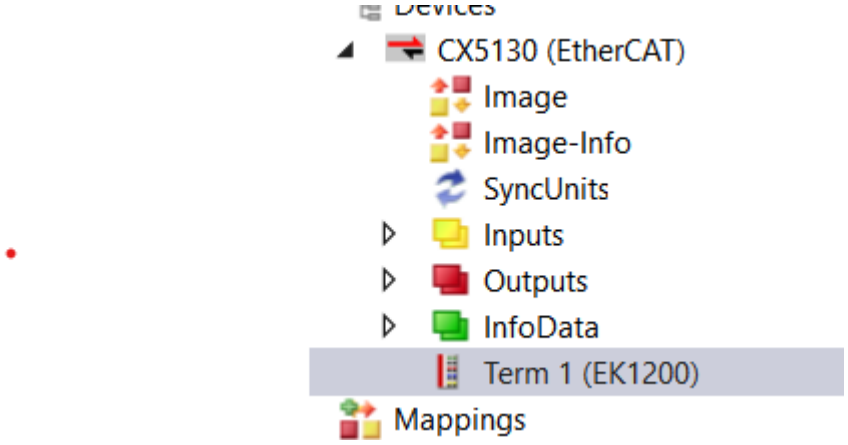
Kuva 7.1.3 Klikataan Cancel Device At ikkunassa

Ennen terminaalimoduulien lisäämistä täytyy **EtherCAT master**-laitteen alle lisätä vielä **EK1200** ennen kuin muita terminaalimoduuleja pystytään lisäämään. Lisääminen tapahtuu klikkaamalla lisättyä **EtherCAT master**-laitetta hiiren oikealla, valitsemalla **Add New Item** ja valitsemalla avautuvasta valikosta **EK1200-5000 EtherCAT Power Supply (2A E-Bus)**



Kuva 7.1.4 Lisätään EtherCAT masterin alle EK1200

Tämän jälkeen laitepuu sisältää **EtherCAT master**-laitteen ja sen alla terminaalimoduulin **EK1200**.



Kuva 7.1.5 CX5130 master ja sen alla oleva EK1200 terminaalimoduuli

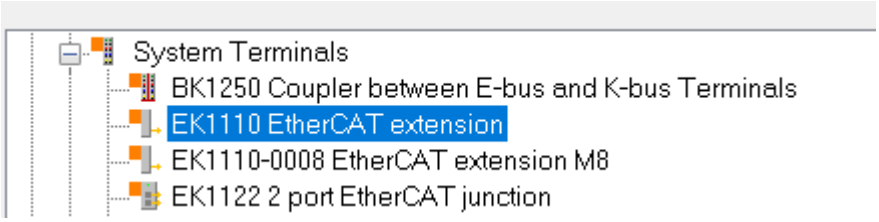
Muiden moduulien lisääminen nyt onnistuu samaan tapaan.

7.1.2 Terminaalimoduulien lisääminen

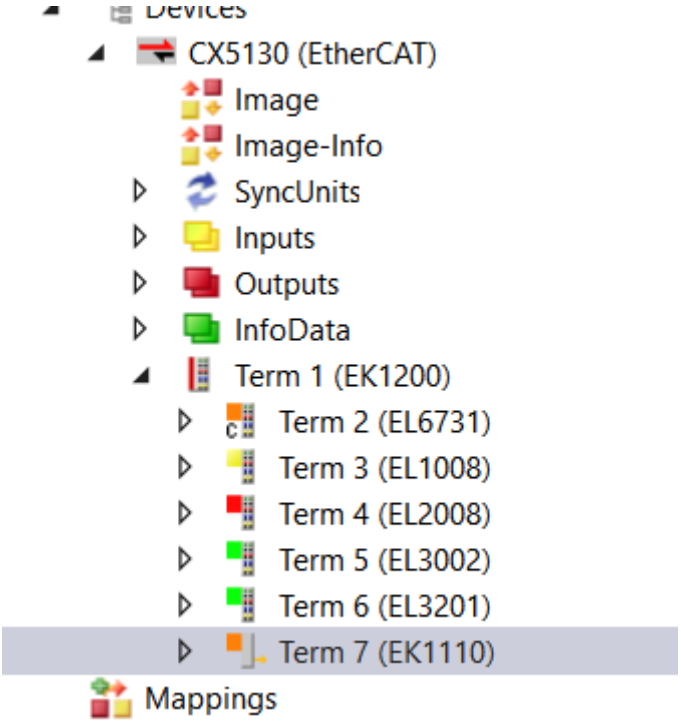
Lisätään seuraavaksi terminaalimoduulit

- EL6731
- EL1008
- EL2008
- EL3002
- EL3201

Lopuksi lisätään vielä **EK1110**-moduuli, eli väyläcoupler. Sen avulla voidaan järjestelmää jatkaa EtherCAT-kaapelilla.



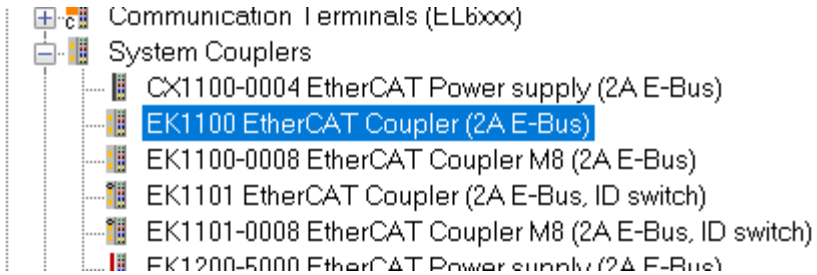
Kuva 7.1.6 Lisätään väyläcoupler EK1110



Kuva 7.1.7 Laitepuu väyläcouplerin lisäyksen jälkeen

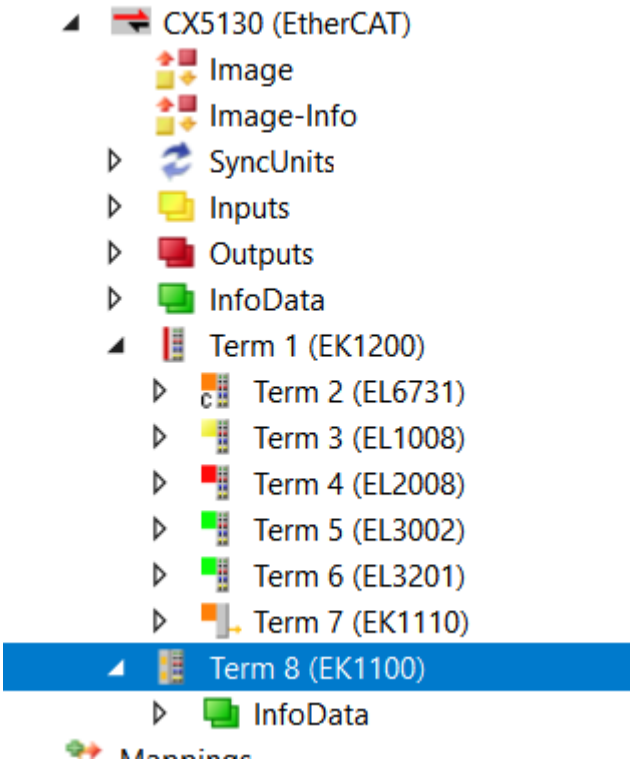
Kun **EK1200** terminaalimoduulin alle on lisätty väyläcoupler **EK1110**, voidaan nyt **EtherCAT master** laitteen alle lisätä väyläcoupler johon EtherCAT-kaapeli viedään.

Lisätään siis **EtherCAT master** laitteen alle **EK1100**.



Kuva 7.1.8 Lisätään väyläcoupler EK1100

Tämän jälkeen laitepuun tulisi näyttää ensin **EtherCAT master**-laite ja sen alla terminaalimoduulit **EK1200** ja **EK1100**

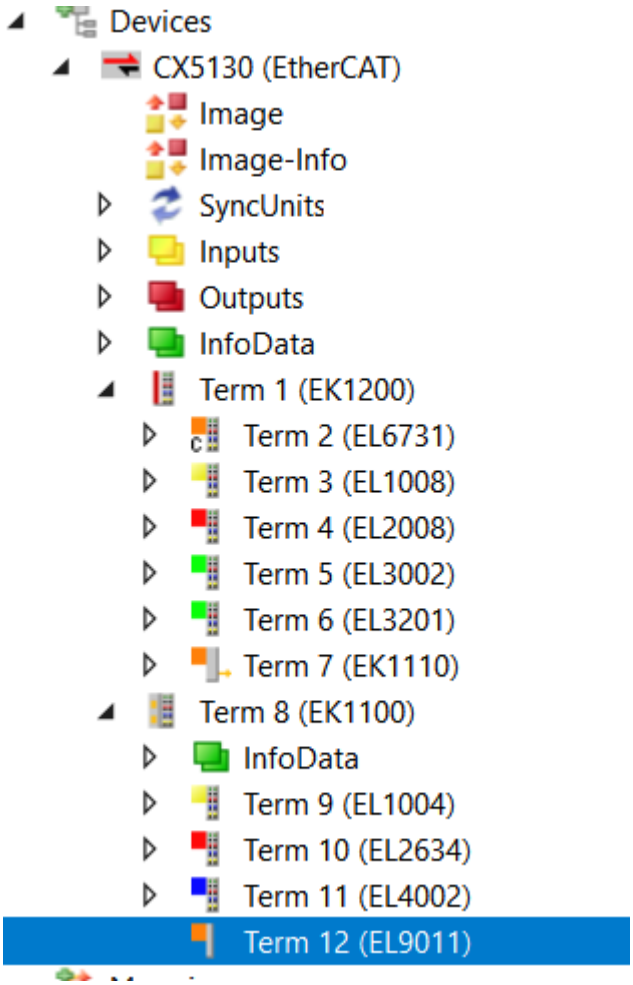


Kuva 7.1.9 EtherCAT master CX5130 ja sen alla EK1200 ja EK1100

EK1100-moduulin alle lisätään sen jälkeen loput I/O moduulit

- EL1004
- EL2634
- EL4002

Viimeiseksi lisätään vielä **EL9011 end cap** jollainen tulee laitta aina järjestelmän viimeiseksi moduuliksi.



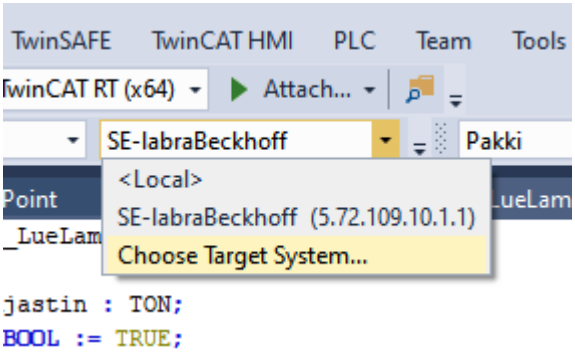
Kuva 7.1.10 Laitepuu kaikkien moduulien lisäyksen jälkeen

7.2 Laitteiston lisäys skannaamalla

I/O-laitteisto voidaan lisätä myös automaattisesti skannaamalla verkkoon liitetty PLC-laitteisto. Skannaamalla saadaan kaikki laitteet lisättyä kerralla, ja varmasti oikein, joten jos mahdollista kannattaa sitä käyttää. Skannauksen onnistumiseksi täytyy PLC-laitteiston olla samassa lähiverkossa kuin käytettävä tietokone.

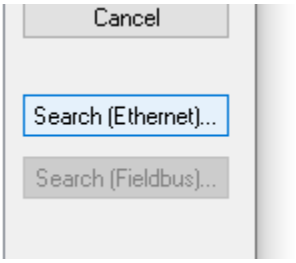
Skannaaminen tapahtuu asettamalla TwinCAT Runtime ensin konfiguraatiotilaan klikkaamalla työkalupalkista **Configuration Mode** ikonia.

Sen jälkeen täytyy PLC-laitteseen muodostaa yhteys. Ensin valitaan yläpalkin **target** alasvetovalikosta **Choose Target System**, jolloin aukeaa laitteiden hakuun käytettävä ikkuna.



Kuva 7.2.1 Valitaan kohdelaitteisto

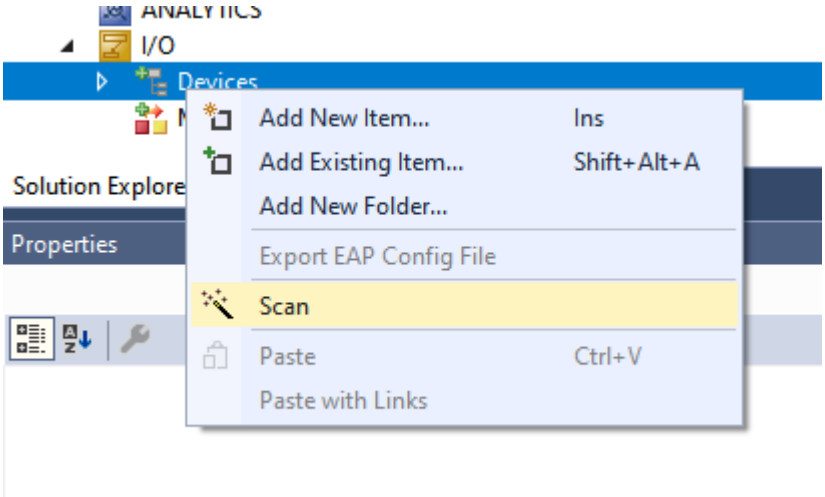
Laitteiden hakuikkunasta valitaan **Search (Ethernet)**, jos PLC-laitteen IP-osoite ei ole tiedossa. Jos IP-osoite tiedetään, voidaan se syöttää suoraan.



Kuva 7.2.2 Etsitään laitteita lähiverkosta

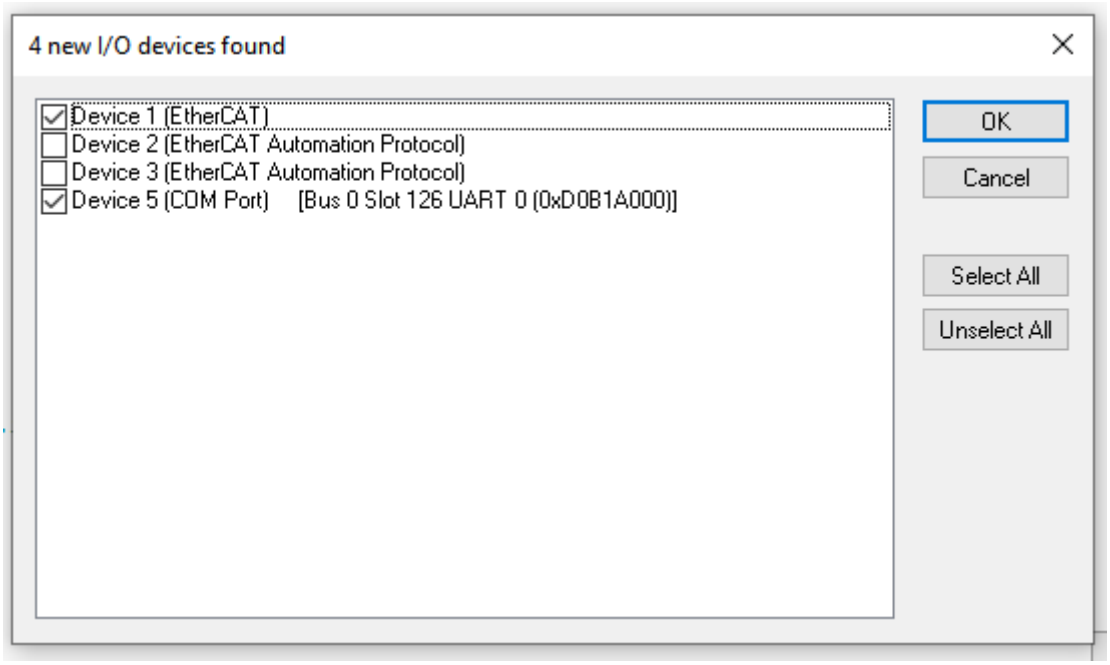
Kun PLC-laite on löydetty verkosta, ja se on valittu **target** valikosta, voidaan se skannata.

Skannaus aloitetaan klikkaamalla projektipuun **I/O** kohdan alla kohtaa **Devices** hiiren oikealla ja valitsemalla **Scan**.



Kuva 7.2.3 Skannataan moduulit löydetystä PLC-laitteesta

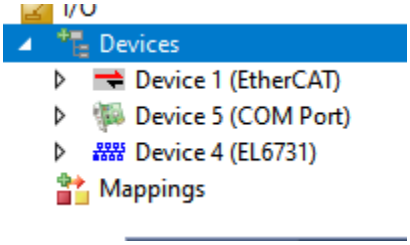
Kun on tehty, näyttää TwinCAT mitä laitteita se löysi. Koulun laitteiston tapauksessa löydetään neljä laitetta.



Kuva 7.2.4 Koulun laittestosta löytyy neljä laitetta

Device 1 on CX5130, eli PLC laite, jolla ohjelmat ajetaan. **Device 2** on EK1110, eli EtherCAT coupler jonka avulla järjestelmää voidaan jatkaa EtherCAT kaapelilla. **Device 3** on EK1100, eli EtherCAT coupler jota käytetään EtherCAT kaapelin toisessa päässä järjestelmää jatkettaessa. **Device 5** on EL6731, eli ProfiBUS liitäntä moduuli, jolla järjestelmää voidaan jatkaa ProfiBUS väylään.

Näistä lisätään TwinCAT projektiin laitteet 1 ja 5 valitsemalla ne ja klikkaamalla **OK**. Vastataan vielä **OK,** kun TwinCAT kysyy, halutaanko myös moduulit skannata (kuva?). Tämän jälkeen projektipuuhun, kohtaan **Devices**, ilmestyy kolme laitetta.



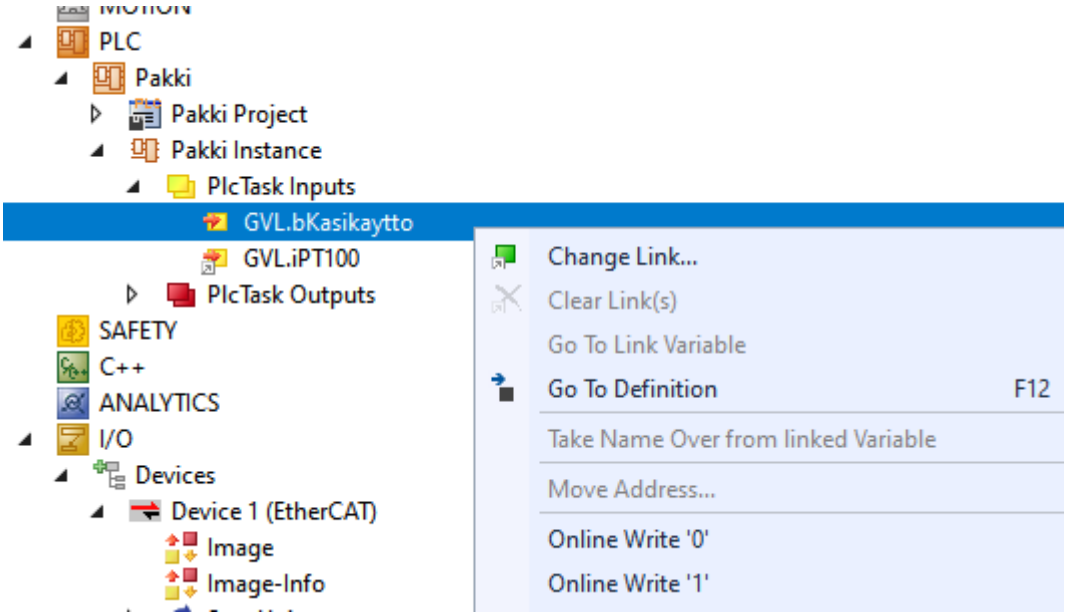
Kuva 7.2.5 Laitepuu kun skannatut laitteet on lisätty

EL6731 ProfiBUS moduulin COM-portti näkyy siis erillisenä laitteena.

7.3 Muuttujien linkittäminen tuloihin ja lähtöihin

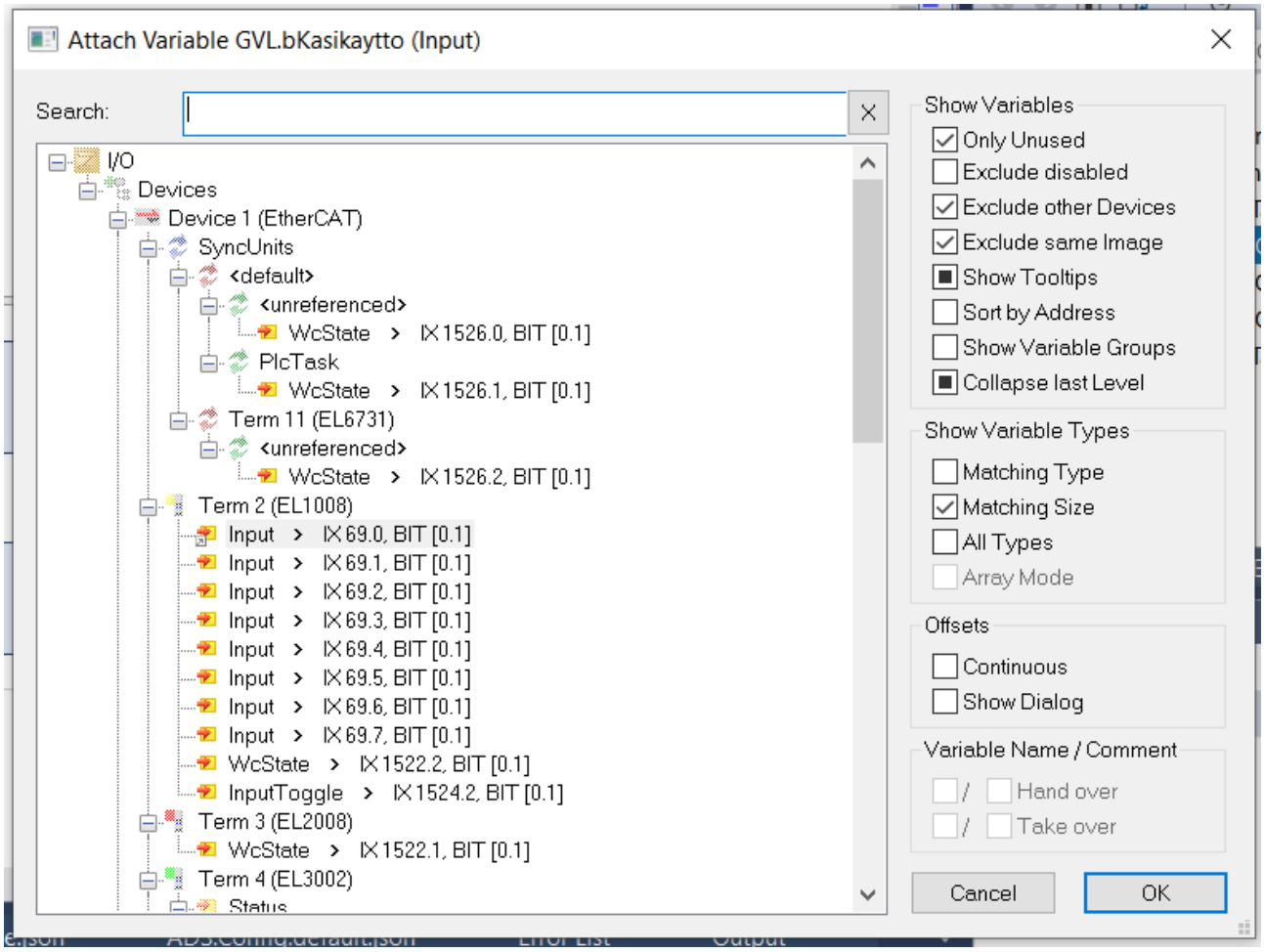
Kun laitteisto on lisätty projektiin, voidaan ohjelmissa käytetyt muuttujat linkittää laitteiston tuloihin ja lähtöihin. Linkittäminen voidaan tehdä joko moduulin tietojan kautta laitepuusta, tai PLC-projektin alta.

PLC-projektin alta linkittäminen tapahtuu aukaisemalla **Instance** kohta, ja edelleen sen alta joko **PlcTask Inputs** tai **PlcTask Outputs** riippuen siitä ollaanko linkittämässä tuloa vai lähtöä. Sen jälkeen klikataan haluttua muuttujaa oikealla ja valitaan **Change Link**



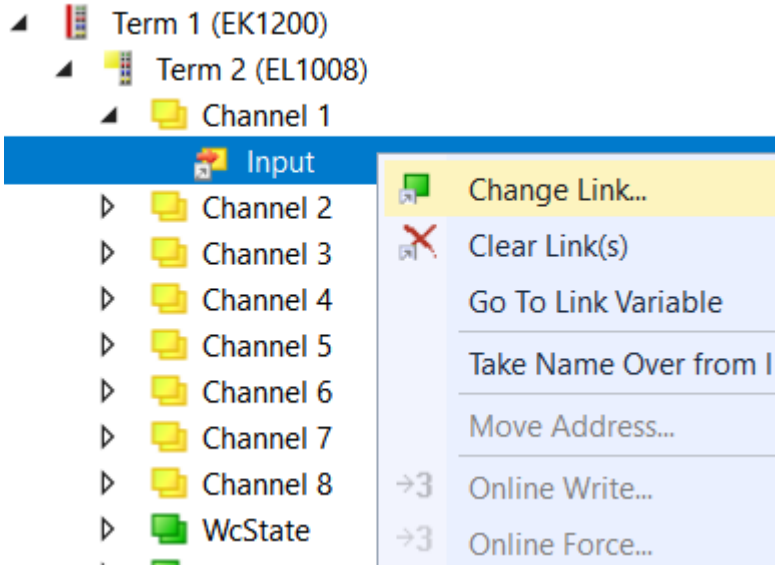
Kuva 7.3.1 Muuttujan linkitys PLC-projektin inputtien kautta

Seuraavaksi aukeaa ikkuna, josta voidaan valita tulo tai lähtö, johon muuttuja linkitetään.



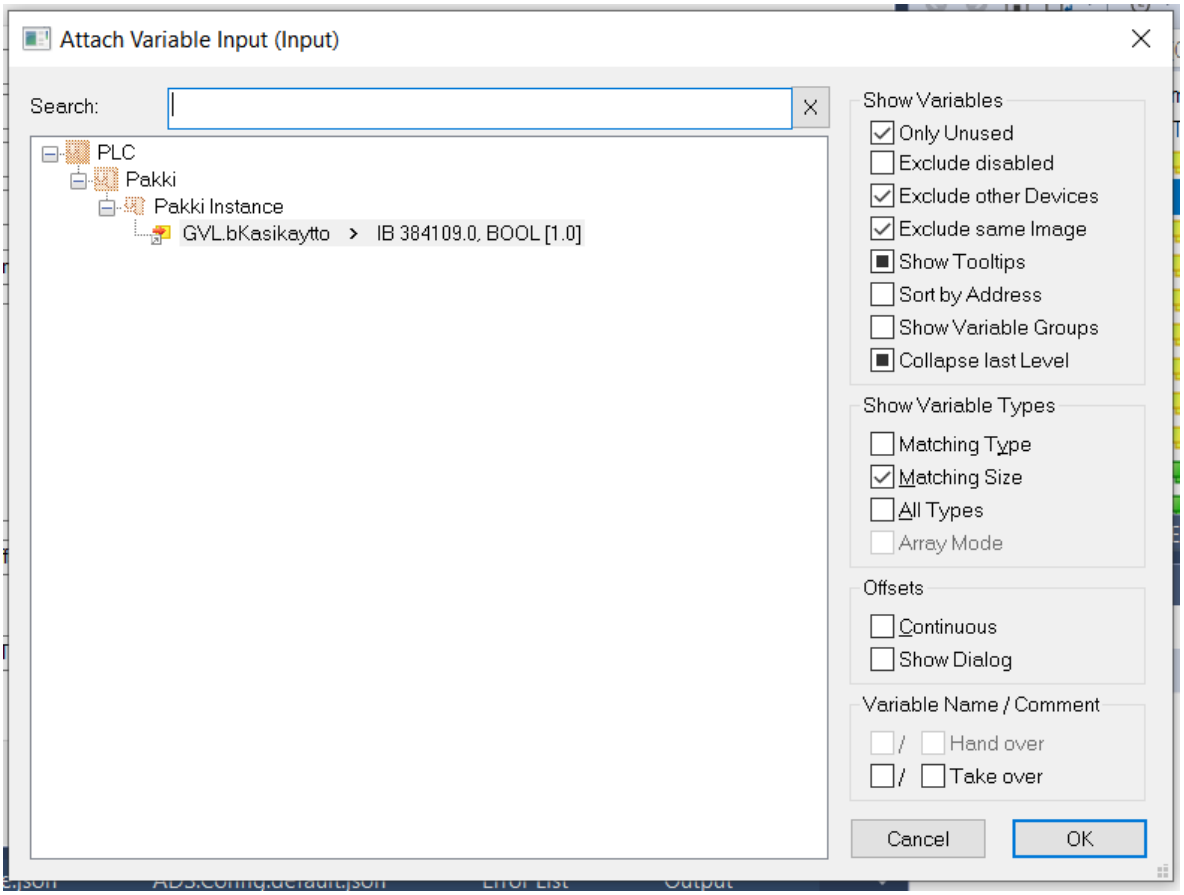
Kuva 7.3.2 Valitaan tulo/lähtö johon muuttuja linkitetään

Laitepuusta linkittäminen tapahtuu aukaisemalla halutun moduulin alta ensin tulo/lähtö (esim. Channel 1) ja sen jälkeen klikkaamalla **Output** tai **Input** kohtaa oikealla ja valitsemalla **Change Link**.



Kuva 7.3.3 Muuttujan linkitys laitepuussa olevan moduulin kautta

Aukeavasta ikkunasta nähdään valittuun tuloon tai lähtöön sopivat muuttujat, joihin linkitys voidaan tehdä.



Kuva 7.3.4 Linkitettävän muuttujan valinta on helppoa koska vain sopivat muuttujat näytetään

8 Ongelmatilanteet

8.1 Virtuaalikoneet

Koska TwinCat ajaa PLC:tä virtuaalisesti PC:llä saattaa se aiheuttaa ongelmia, jos samaan aikaan on käytössä muita virtuaalikoneita. TwinCat projektia ei tästä syystä esimerkiksi pysty asettamaan Run-moodiin, jos samaan aikaan PC:llä on käynnissä esimerkiksi VirtualBox tai jokin muu virtualisointijärjestelmä.

9 Linkkejä

Structured Text ohjelmointiopas

https://www.plcacademy.com/structured-text-tutorial.pdf

TwinCAT 3 esittely

https://download.beckhoff.com/download/document/catalog/TwinCAT_3_Booklet.pdf

TwinCAT 3 tutoriaali

http://www.contactandcoil.com/twincat-3-tutorial/quick-start/