

TAMPEREEN AMMATTIKORKEAKOULU  
Kone- ja tuotantotekniikan koulutusohjelma  
Kone- ja laiteautomaatio

Opinnäytetyö

Petteri Kuusisto

**CAN-VÄYLÄ TwinCAT-YMPÄRISTÖSSÄ**

Työn ohjaaja  
Työn teettäjä  
Tampere 2008

DI Leo Sutinen  
Beckhoff Automation Oy

TAMPEREEN AMMATTIKORKEAKOULU

Kone- ja tuotantotekniikan koulutusohjelma

Kone- ja laiteautomaatio

Petteri Kuusisto

CAN-Väylä TwinCAT ympäristössä

Opinnäytetyö

57 sivua + 5 liitesivua

Työn ohjaaja

DI Leo Sutinen

Työn teettäjä

Beckhoff Automation Oy

Heinäkuu 2008

Hakusanat

twincat, can-väylä, logiikka, ohjekirja, ohjelmointi

## TIIVISTELMÄ

TwinCAT on Windows-pohjainen tietokoneohjelma. Se sisältää sekä reaaliaikaisen ohjausjärjestelmän että ohjelmointiympäristön ja diagnosointi- ja konfigurointityökalut. Sen avulla voidaan suunnitella ja luoda ohjelmia. TwinCATilla luotuja ohjelmia voidaan käyttää PC:illä ja väyläterminaalien ohjaimissa. Väyläterminaalien ohjaimet liitetään sarjaliitäntöjen tai kenttäväylien avulla.

Tämä opinnäytetyö tehtiin opetuskäyttöä varten Tampereen ammattikorkeakoulun kone- ja laiteautomaation suuntautumisvaihtoehtoa varten. Aiheena oli laatia ohjekirja siitä, kuinka TwinCATilla luodun logiikkaohjelman saa liitettyä CANopen-kenttäväylään. Ohjekirjasta löytyy myös tietoa ohjelmasta, sen käytöstä ja käytettävistä operaatioista. Siinä on myös lyhyt ohjeistus ohjelmoimisesta sekä CANopen-väylän pystyttämisestä.

TAMPERE POLYTECHNIC

Mechanical and Production Engineering

Machine Automation

Petteri Kuusisto

Using CAN-Bus in TwinCAT Environments

Engineering Thesis

57 pages + 5 appendices

Thesis Supervisor

M.Sc. Leo Sutinen

Commissioning Company

Beckhoff Automati GmbH

July 2008

Keywords

twincat, can-bus, logic, manual, programming

## **ABSTRACT**

TwinCAT is the Windows control computer program. It consists of run-time systems that execute control programs in real-time and the development environments for programming, diagnostics and configuration. It is used for creating and designing special programs. With TwinCAT created programs can run on the PCs or on Beckhoff bus terminal controllers. Bus terminal controllers are integrated with via serial interfaces or fieldbuses.

This engineering research report was made for educational use to Tampere Polytechnic. The purpose of this report was make devise manual that informs the reader how to use the program. It also includes instructions for programming and basic information about the program, for example its user interface and functions it contains. There is also a chapter for CAN-bus.

## **ALKUSANAT**

Kiitän Beckhoff Automationia ja Beckhoffin aluemyyntipäällikkö Aki Kalajaista haastavasta ja mielenkiintoisesta opinnäytetyöaiheesta. Erityiskiitokset laboratorioinsinööri Seppo Mäkelälle ja työni ohjaajalle Leo Sutiselle, joilta olen saanut asiantuntevia neuvoja ja apua koko opinnäytetyöni laatimisen ajan.

Tampereella 29.10.2008

Petteri Kuusisto



## SISÄLLYSLUETTELO

TIIVISTELMÄ .....	2
ABSTRACT .....	3
ALKUSANAT .....	4
LYHENTEET .....	8
1 JOHDANTO .....	10
2 CAN-VÄYLÄ.....	11
2.1 Mikä on CAN? .....	11
2.2 Perusominaisuudet .....	11
2.3 Tiedonsiirron periaate .....	12
2.4 Kehysrakenne.....	13
2.5 Kilpavaraus .....	14
2.6 Virheiden käsittely .....	14
2.7 CANopen .....	15
2.7.1 Rakenne.....	15
2.7.2 Tiedonsiirto .....	15
3 BECKHOFF.....	15
3.1 Beckhoff Suomessa.....	16
3.2 Beckhoff maailmalla.....	16
3.3 Tuotteet ja palvelut .....	16
4 TwinCAT- LOGIIKKAOHJELMA.....	17
4.1 PLC ja liikkeenohjaus PC:ssä .....	17
4.2 TwinCAT-arkkitehtuuri .....	17
4.3 Ohjelmistoratkaisu .....	17
4.4 Käynnistys ja pysäytys.....	18
4.5 Uudelleenkäynnistys ja tietojen suojaus .....	18
4.6 TwinCAT ja "sininen ruutu" .....	19
4.7 Maailmanlaajuiset yhteydet Message Routing -toiminnolla.....	19
4.8 Maailmanlaajuinen tiedonsiirto .....	19
5 OHJEKIRJA: TwinCAT-logiikkaohjelman käyttö.....	20
5.1 System Manager.....	20
5.2 I/O-määrittely.....	21

5.2.1 Skannaus .....	21
5.2.2 Käsien määrittely .....	22
5.2.3 Määritellyn I/O:n testaus.....	25
5.3 TwinCat PLC-projektin luominen .....	28
5.3.1 Ohjelmointikielet .....	30
5.3.1.1 IL.....	30
5.3.1.2 LD .....	31
5.3.1.3 FBD.....	31
5.3.1.4 SFC .....	32
5.3.1.5 ST.....	33
5.3.1.6 CFC.....	33
5.4 Ohjelmointiympäristö .....	34
5.4.1 Pikakuvakkeet .....	35
5.4.2 Välilehdet.....	35
5.5 Ohjelmointi .....	39
5.5.1 Pikanäppäimet FBD- ohjelmoinnissa .....	40
5.5.2 Avustavat toiminnot.....	40
5.5.3 Aputoiminto .....	41
5.6 Operaatiot.....	41
5.6.1 Peruskäskyt .....	42
5.6.2 Vertailijat .....	43
5.6.3 Ajastimet .....	44
5.6.4 Laskurit .....	46
5.6.5 Aritmetiikkakäskyt.....	47
5.7 Ohjelman lataaminen järjestelmään.....	48
5.8 PLC-projektin liittäminen konfiguraatioon .....	50
5.9 PLC-muuttujien linkittäminen .....	52
6 TULOKSET.....	55
7 TULOSTEN TARKASTELU.....	56

LÄHTEET

LIITTEET

## LYHENTEET

ACK	Acknowledge, ACK-bitti kertoo lähettäjälle, onko vastaanottaja saanut viestin
BSOD	Blue Screen of Death, suom. sininen ruutu
CFC	Continuous Function Chart
CNC	Computerized Numerical Control, tietokoneistettu numeerinen ohjaus
CRC	Cyclic Redundancy Check, tarkisteavaimen luontiin tarkoitettu tiivistealgoritmi
DLC	Data Length Code, ilmaisee viestissä olevien datatavujen lukumäärän
EOF	End Of Frame, ilmaisee kehyksen loppumisen
FB	Funktion Block, suom. toimilohko
FBD	Function Block Diagram, suom. toimilohkokaavio
HTTP	Hypertext Transfer Protocol
I/O	Input/Output, suom. tulot/ lähdöt
IDE	I Dentifier Ex-tension, ilmaisee kehyksen pituuden
IFS	Intermission Frame Space
IL	Instruction List, suom. käskylista
kbps	Tiedonsiirtonopeus, kilobittiä sekunnissa
LD	Ladder Diagram, suom. relekaavio
Mbit/s	Tiedonsiirtonopeus, megabittiä sekunnissa
NC	Numerical Control, suom. numeerinen ohjaus
OPC	Object Linking and Embedding for Process Control, teollisuusstandardi
PLC	Programmable Logic Cotrollers, suom. Ohjelmoitavat logiikat
POU	Program Organisation Unit, suom. ohjelmalohko
RTR	Remote Transmission Request, ilmaisee onko kyseessä tieto- vai kyselykehys
SCADA	Supervisory Control And Data Acquistion, valvomo-ohjelmisto
SFC	Sequential Flow Chart, suom. Sekvenssikaavio
SOAP	Simple Object Access Protocol, tietoliikenneprotokolla
SOF	Star Of Frame, ilmaisee viestikehyksen alkavan
ST	Structured Text, suom. strukturoitu teksti



TCP/IP	Transmission Control Protocol/Internet Protocol, Internetin peruskommunikointikieli
UPS	Uninterruptible power supply, keskeytymätön virransyöttölaite

## 1 JOHDANTO

Beckhoffin automaatiotekniikka yhdistettynä TwinCAT -automaatio-ohjelmiston liikkeenohjausteknologiaan tarjoavat tarjoaa monipuolisen automaatiojärjestelmän. Beckhoffin PC-pohjainen ohjaustekniikka soveltuu erinomaisesti korkeaa dynamiikkaa vaativiin yksi- tai moniakselisiin asemointisovelluksiin. Myös Tampereen ammattikorkeakoulun kone- ja laiteautomaatin suuntautumisvaihtoehto tulee ottamaan TwinCATin opetuskäyttöön.

TwinCAT on kehittynyt ja erittäin monipuolinen logiikkaohjelma, joka vaatii käyttäjältään perehtymistä ohjelman käyttöön. Tämän opinnäytetyön tavoitteena on luoda perusteet sisältävä ohjekirja TwinCAT:in käytöstä, sekä antaa ohjeistus siihen kuinka logiikkaohjelma liitetään väyläterminaaliin.

## 2 CAN-VÄYLÄ

### 2.1 Mikä on CAN?

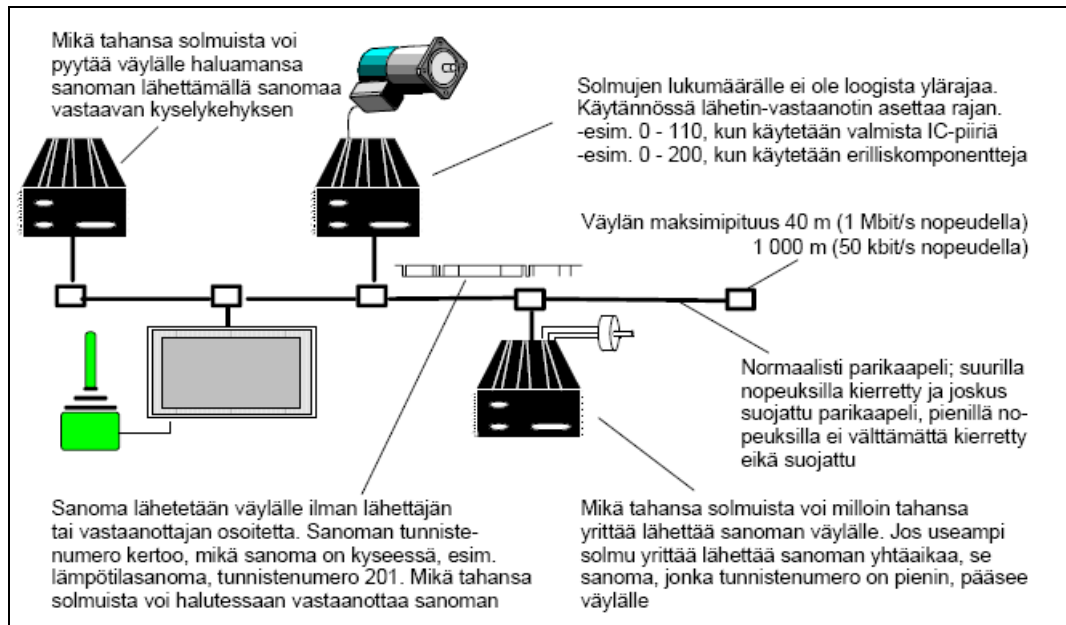
CAN (Controller Area Network) suunniteltiin alun perin autoteollisuuden tarpeisiin. Tästä suunnittelulähtökohdasta johtuu, että erityisesti häiriön sietokyky ja tiedonsiirron varmuus olivat keskeisiä tekijöitä. CAN-väylää sovelletaan kuitenkin laajasti myös muissa tuotteissa, kuten busseissa, hisseissä, maatalouskoneissa, lääketieteellisissä laitteissa, kappaletavara-automaatiossa, roboteissa, mittausjärjestelmissä ja ohjelmoitavissa logiikoissa. CAN sopii siis periaatteessa mihin tahansa koneeseen, jossa on kysymys lyhyistä tiedonsiirtoetäisyyksistä ja tarpeena on reaaliaikainen kontrollereiden välinen kommunikointi, jossa sanomat ovat lyhyitä.

Jokaiseen CAN-viestiin sisältyy tarkistussumma, jolla varmistetaan CAN-viestin perille meno muuttumattomana. Lisäksi CAN-väylä on digitaalinen sarjaliikenneväylä, mikä lisää dataliikenteen häiriöttömyyttä. Jokaisessa CAN -viestissä on ID-kenttä, jonka perusteella voidaan tunnistaa esim. eri laitteille tarkoitetut komennot tai signaalit. Väylällä liikkuvat viestit ovat kaikkien kytkettyjen laitteiden luettavissa, eli samaa tietoa ei tarvitse lähettää useaan kertaan. /2;4;6;8;9/

### 2.2 Perusominaisuudet

CAN-väylä on luonteeltaan usean masterin väylä (multi-master bus), jossa jokainen moduuli voi oma-aloitteisesti lähettää väylälle sanoman. Yhdessä sanomassa voi lähettää korkeintaan 8 tavua (64 bittiä) tietoa. Sanoma lähetetään yleisesti kaikille laitteille vastaanotettavaksi (broadcasting), ja ne solmut, jotka tarvitsevat kyseisen sanoman sisältämää tietoa, ottavat sanoman vastaan. Koska CAN-kommunikoinnissa ei käytetä lähettäjän tai vastaanottajan osoitetta, solmuja voi väylällä olla periaatteessa rajaton määrä; käytännössä kuitenkin lähetin-vastaanotin-piirit asettavat solmujen määrälle ylärajan. Enimmäisnopeus on 1 Mbit/s (Megabittiä sekunnissa),

jolloin väylän maksimipituus on noin 40 m. Perusominaisuudet on kuvattu kuvaan 1.  
/2;4;6;8;9/



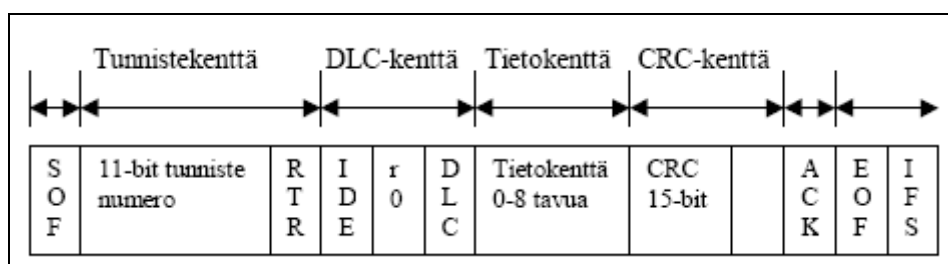
Kuva 1 CAN-väylän perusominaisuudet

### 2.3 Tiedonsiirron periaate

CAN-väylä käyttää objektiorientoitunutta tiedonvälitystä, joka tarkoittaa sitä, että asemilla ei ole varsinaisia osoitteita, vaan sanomat numeroidaan ja lähetetään väylälle yleisesti vastaanotettavaksi. Esimerkiksi CAN-väylään liitettävä painemittari voisi lähettää paineviestiä vaikkapa tunniste numerolla 200. Tällöin kaikki solmut, jotka tarvitsevat paineviestiä, sieppaavat väylältä sanoman, jonka tunniste on 200. Muiden solmujen on kuitenkin tiedettävä, että paineviestin tunniste on 200. Kukaan muu ei saa lähettää samalla tunnisteella, sillä jos kaksi solmua lähettää saman tunnisteista sanomaa samaan aikaan, sanomat törmäävät tavalla, jota CAN-protokolla ei salli. Tunnistenumeron pituus on 11 bittiä standardiformaatissa. /2;4;6;8;9/

## 2.4 Kehysrakenne

CAN-protokolla määrittelee neljä erilaista kehystä: sanomakehys, kyselykehys, virhekehys ja ylikuormituskehys. Varsinaisen sanomakehyksen pääosat ovat tunnistekenttä, jossa on sanoman tunnistenumero, DLC-kenttä (Data Length Code), joka ilmaisee tietokentän pituuden sekä tietotavujen kenttä, jonka maksimipituus on kahdeksan tavua. Muut kentät on esitelty kuvassa 2.

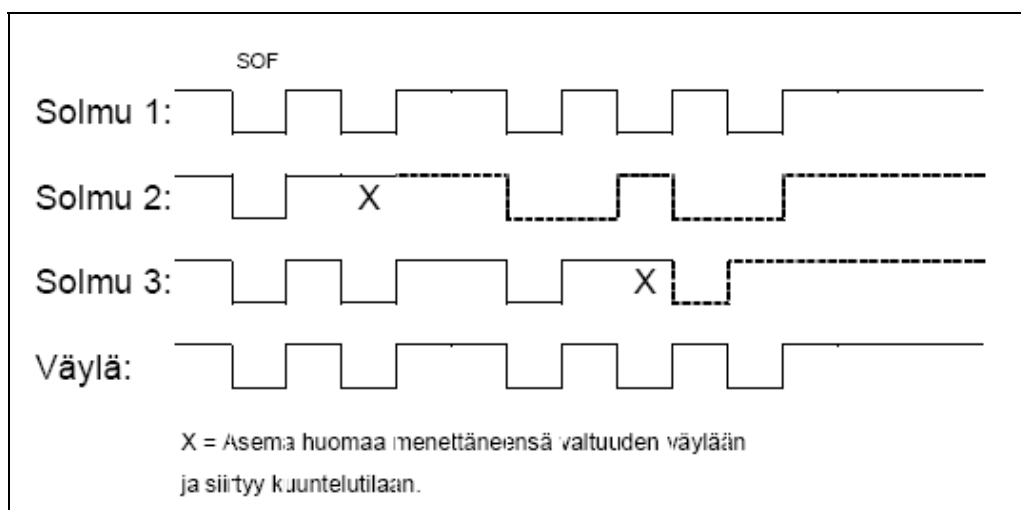


Kuva 2 Sanomakehyksen rakenne

CAN-viesti alkaa SOF-bitillä (Star Of Frame), joka ilmaisee dominoivalla bitillä viestikehyksen alkavan. SOF- bittiä seuraa tunnistekenttä, joka ilmaisee viestin prioriteetin ja yksilöi viestin. RTR-bitti (Remote Transmission Request) kertoo, onko kyseessä tieto- vai kyselykehys. IDE-bitti (Identifier Extension) ilmaisee, onko kyseessä standardi pituinen vai laajennettu kehys. r0-bitti on varattu tulevia sovelluksia varten. DLC-osa ilmaisee tavumäärän neljällä bitillä. CRC-kenttä (Cyclic Redundancy Check) ilmaisee viestin oikeellisuuden 15-bittisellä tarkistussummalla. ACK- bitti (Acknowledge) kertoo lähettäjälle, onko vastaanottaja saanut viestin. ACK on resessiivinen bitti lähetettäessä, mutta vastaanottaja kirjoittaa siihen dominantin bitin, tällöin lähettäjä tietää onko viesti vastaanotettu. EOF-kenttä (End Of Frame) ilmaisee kehyksen loppumisen seitsemällä peräkkäisellä bitillä. Kehysten välillä pitää olla vähintään kolme resessiivistä bittiä, ennen kuin väylä on vapaa, IFS-kenttä (Intermission Frame Space) sisältää nämä kolme bittiä. /2;4;6;8;9/

## 2.5 Kilpavaraus

Kuvasta 3 selviää, kuinka pienimmän tunnistenumeron omaava asema saa etuoikeuden väylälle. Alemman prioriteetin viestiä lähettävät asemat siirtyvät vastaanottajiksi ja odottavat väylän vapautumista ennen kuin yrittävät lähettää viestiä uudestaan.



Kuva 3 Kilpavarauksen periaate

CAN-väylällä törmäyksen sattuessa dominoiva asema jatkaa lähetystä. CAN-väylän ennustettava käytös mahdollistaa periaatteessa 100 % kaistan käytön, mutta tätä ei kuitenkaan suositella. Periaate on tarpeellinen ylikuormitustapauksissa, jolloin taataan tärkeimpien viestien perille meno. /2;4;6;8;9/

## 2.6 Virheiden käsittely

Virheiden tarkistukseen käytetään yleisimmin CRC-summaa. Virheitä voidaan myös tarkastella seuraavilla tarkistusmenetelmillä:

- Lähettäjä monitoroi, onko väylällä sama bitti kuin lähetetty.
- Bit- stuffing -säännön rikkominen, eli kehykseen kuuluvia kiinteitä bittejä tarkkaillaan.
- Viestikehyksien rakennetta vertaillaan kiinteään mallikehykseen.

## 2.7 CANopen

CAN-väylällä voidaan käyttää myös jotain monista ylemmän kerroksen protokollista tiedonsiirron hallitsemiseksi. Yksi yleisesti käytetty on CANopen-protokolla. Se on moderni, yleiskäyttöinen ylemmän kerroksen kommunikointiprotokolla, jonka lähtökohtana on alun perin ollut anturien, toimilaitteiden ja ohjelmoitavien logiikoiden tarpeet, mutta joka on nykyisin käytössä monissa laajemmissa sovelluskohteissa (mm. bussit ja trukit). CANopen on avoin protokolla, jota tukee laaja valikoima erilaisia toimilaitteita ja kehitysympäristöjä. /9/

### 2.7.1 Rakenne

CANopen-järjestelmä perustuu väylärakenteeseen, johon voidaan liittää 127 asemaa. Samaan väylään voidaan liittää useita master-asemia. Väylä päätetään päätevastuksella. Väylän suurin sallittu pituus on 1 200 metriä. Pituutta voidaan lisätä vahvistimen avulla enintään 5 000 metriin tiedonsiirtonopeuden ollessa 10 kbps (kilobittinä sekunnissa). /8;9/

### 2.7.2 Tiedonsiirto

CANopen kommunikoi häiriöitä sietävän protokollan avulla. Virheellisten tietojen ja viollisten asemien havaitsemiseen käytetään useita menetelmiä, kuten 15-bittistä CRC-tarkastusta. Tietoa voidaan lähettää jatkuvasti niin sanotussa broadcast-tilassa tai tapahtumakohtaisesti. /2;4 /

## 3 BECKHOFF

Beckhoff toimittaa automaatiojärjestelmiä, jotka pohjautuvat PC-pohjaiseen ohjaustekniikkaan. Beckhoffin tuotevalikoimaan kuuluvat mm. kenttäväyläkomponentit, liikkeenohjaustuotteet, teollisuus-PC:t ja ohjauspaneelit sekä automaatio-sovelluksien ohjelmistot. Eri ryhmien tuotteita voidaan käyttää erillisinä komponentteina, tai ne voidaan integroida yhteen ohjausjärjestelmiksi.

Beckhoffin tuotteita ja järjestelmäratkaisuja käytetään maailmanlaajuisesti monenlaisissa sovelluksissa nopeista työstökeskuksista aina älykkääseen rakennusautomaatioon. /7/

### **3.1 Beckhoff Suomessa**

Beckhoff Suomen pääkonttori sijaitsee Hyvinkäällä. Haarakonttorit sijaitsevat Tampereella ja Seinäjoella. Kaikissa Beckhoffin Suomen konttoreissa on myynti, tekninen tuki, koulutus, tuotekehitys, sovellukset ja huolto edustettuna. Näiden lisäksi Hyvinkäällä sijaitsee Beckhoff komponenttien ja tuotteiden varasto. Beckhoff on toiminut Suomessa vuodesta 1986, ensin edustajan välityksellä ja vuodesta 2000 Beckhoff yhtiönä. /7/

### **3.2 Beckhoff maailmalla**

Beckhoff Automation GmbH:n keskuspaikka sijaitsee Saksan Verlissä. Keskuspaikka käsittää kaikki tärkeimmät ja keskeisimmät osastot, kuten kehityksen, tuotannon, hallinnon, myynnin, markkinoinnin, tuotetuen ja asiakaspalvelun. Tytäryritykset edustavat Beckhoffia kansainvälisillä markkinoilla Belgiassa, Tanskassa, Suomessa, Ranskassa, Italiassa, Itävallassa, Puolassa, Venäjällä, Ruotsissa, Sveitsissä ja Espanjassa sekä Australiassa, Brasiliassa, Kiinassa, Intiassa, Kanadassa, Etelä-Afrikassa, Yhdysvalloissa ja Arabiemiraattien liitossa. Tämän lisäksi Beckhoffin yhteistyökumppanit edustavat yli 60 maassa ympäri maailmaa. /7/

### **3.3 Tuotteet ja palvelut**

Tuotekehityksen tuloksena syntyneet tuotteet ja ratkaisut ovat muodostaneet pohjan Beckhoffin toiminnalle. Beckhoffin PC-pohjainen ohjaustekniikka yhdessä Lightbus järjestelmän, väyläterminaalien ja TwinCAT-automaatio-ohjelmiston kanssa edustavat kulmakiviä nykyaikaisessa automaatiotekniikassa ja ovat tulleet yleisesti vaihtoehtoiksi perinteiselle ohjaustekniikalle. Reaaliaikainen Ethernet ratkaisu,



EtherCAT mahdollistaa korkean suorituskyvyn ja edistyksellisyyden uuden sukupolven ohjaustekniikoille. /7/

## **4 TwinCAT- LOGIIKKAOHJELMA**

### **4.1 PLC ja liikkeenohjaus PC:ssä**

TwinCAT-logiikkaohjelma on Windows-pohjainen ohjausjärjestelmä, jonka avulla voidaan PC:stä tehdä usean PLC:n (Programmable Logic Cotrollers, suom. ohjelmoitavat logiikat) reaaliaikainen ohjausjärjestelmä. Reaaliaikalaajennus on toteutettu integroimalla reaaliaikaominaisuudet suoraan Windows NT/2000/XP/Vistaan. Tiedonsiirto on mahdollista käyttöliittymään tai ulkoisiin ohjelmiin Microsoftin avoimien rajapintojen avulla. TwinCAT-ohjausohjelmisto on IEC61131-3-standardin mukainen avoin, yhteensopiva PC-laitteisto. /5;7/

### **4.2 TwinCAT-arkkitehtuuri**

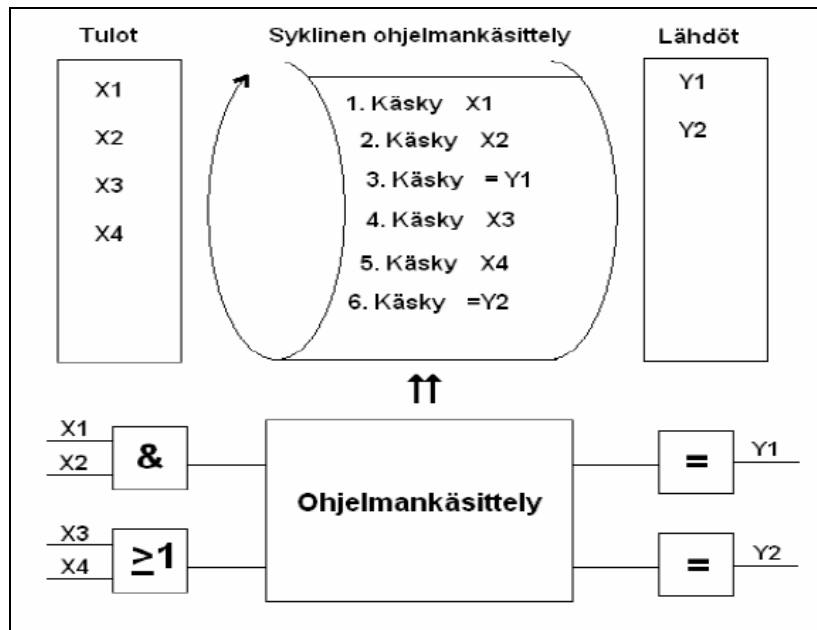
TwinCAT-järjestelmä sisältää reaaliaikaisen ohjausjärjestelmän, ohjelmointiympäristön ja diagnosointi- ja konfigurointityökalut. Kaikki Windows-ohjelmat (esim. visualisointi- tai Office-ohjelmistot) voivat hakea tietoja tai suorittaa komentoja TwinCATissa Microsoftin standardi-liityntärajapintojen kautta. /5;7/

### **4.3 Ohjelmistoratkaisu**

TwinCATissa ohjelman käsittely tapahtuu deterministisesti täsmällisesti ajastetuissa sykleissä riippumatta muista prosessorin tehtävistä (kuva 4). Reaaliaikaohjauksen aiheuttaman kuormituksen määrää yhdelle PC:lle voidaan helposti säätää TwinCATin avulla. Näin pystytään saavuttamaan haluttu suorituskyky. TwinCAT näyttää kaikkien käynnissä olevien ohjelmien kuormituksen. Järjestelmään voidaan asettaa

kuormitusrajoitus, joka varmistaa, että käyttöjärjestelmällä on riittävästi kapasiteettia käytössään. Jos asetettu kuormitusrajoitus ylitetään, järjestelmä ilmoittaa virheestä.

/5;7/



Kuva 4 Syklinen ohjelmankäsittely

#### 4.4 Käynnistys ja pysäytys

TwinCAT voidaan käynnistää ja pysäyttää asetuksista riippuen joko manuaalisesti tai automaattisesti. TwinCAT on integroitu Windows NT/2000/XP/Vista-käyttöjärjestelmään, järjestelmän käynnistämiseen ei tarvita erillisiä käyttäjän toimenpiteitä – virran päälle kytkeminen riittää. /5;7/

#### 4.5 Uudelleenkäynnistys ja tietojen suojaus

Käynnistettäessä tai uudelleenkäynnistettäessä TwinCAT lataa ohjelmat ja tallennetut tiedot automaattisesti. Tietojen turvaamisen ja Windows NT/2000/XP/Vistan turvallisen sulkemisen kannalta olisi suositeltavaa käyttää UPS:ia (uninterruptible power supply). /5;7/

#### 4.6 TwinCAT ja "sininen ruutu"

TwinCAT-järjestelmä pystytään konfiguroimaan siten, että reaaliaikaohjaus jää toimintakykyiseksi, vaikka käyttöjärjestelmä kaatuisi BSOD:n (Blue Screen of Death, suom. sininen ruutu) vuoksi. Reaaliaikatoiminnot, kuten PLC ja NC (Numerical Control), voivat sen ansiosta jatkaa toimintaansa loppuun ja saattaa ohjaamansa prosessin turvalliseen tilaan. On kuitenkin täysin ohjelmoijan päätettävissä, käyttääkö hän tätä ominaisuutta vai ei, ottaen huomioon sen, että tiedot ja ohjelmat voivat jolla vioittuneet BSOD:n vuoksi. /5;7/

#### 4.7 Maailmanlaajuiset yhteydet Message Routing -toiminnolla

TwinCATin ohjelmistoa voidaan tarvittaessa jakaa käyttölaitteille: TwinCAT PLC-ohjelmia voi käyttää PC:illä ja Beckhoffin väyläterminaalien ohjaimissa. Niin kutsuttu "Message Router" hallinnoi ja jakelee viestit järjestelmän sisällä ja TCP/IP-protokollan (Transmission Control Protocol/Internet Protocol, suom. Internetin peruskommunikointikieli) avulla. PC-järjestelmät voidaan liittää toisiinsa TCP/IP:n avulla. Väyläterminaalien ohjaimet liitetään sarjaliitäntöjen tai kenttäväylien (Lightbus, PROFIBUS DP, CANopen, RS232, RS485, Ethernet TCP/IP) avulla. /7/

#### 4.8 Maailmanlaajuinen tiedonsiirto

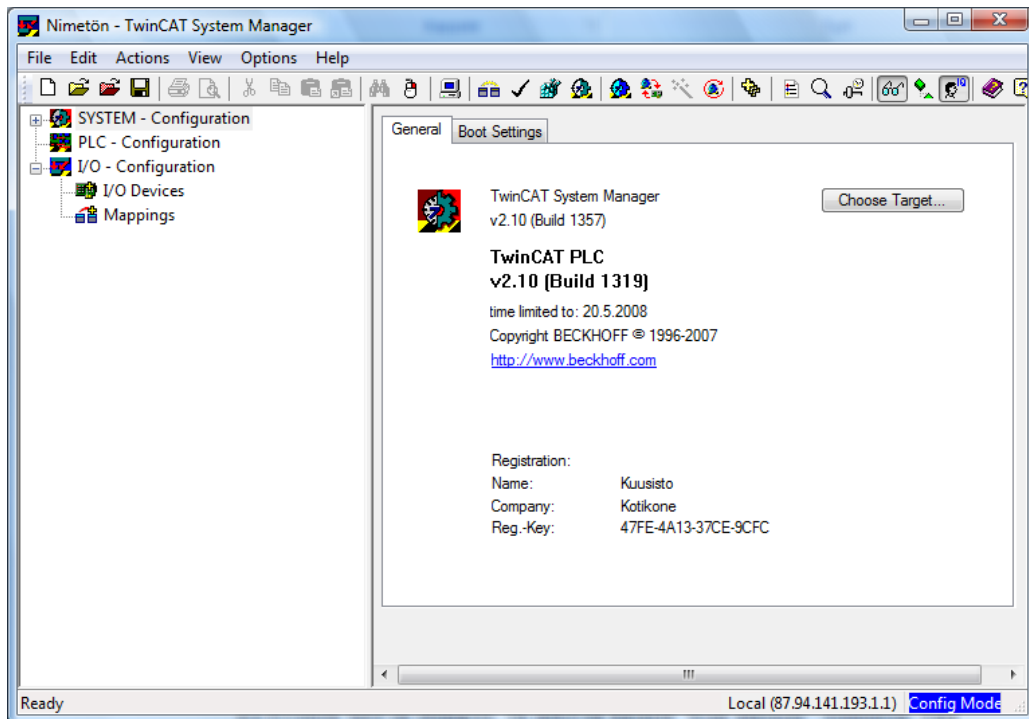
Koska järjestelmässä voidaan käyttää Windows NT/2000/XP/Vistan perus-TCP/IP-protokollia, on tiedonsiirto mahdollista kaikkialle maailmaan. Järjestelmä tarjoaa skaalattavan tiedonsiirtokapasiteetin ja aikavalvontakatkaisun tiedonsiirron valvontaan. OPC:n (Object Linking and Embedding for Process Control) avulla voidaan päästä käsittelemään monia erilaisia SCADA (Supervisory Control And Data Acquisition) -paketteja. SOAP:n (Simple Object Access Protocol) avulla voidaan kaksi tietokonetta yhdistää internetin kautta perus-HTTP-protokollan (Hypertext

Transfer Protocol) avulla. TwinCATissa on tähän tarkoitukseen erillinen komponentti. /7/

## 5 OHJEKIRJA: TwinCAT-logiikkaohjelman käyttö

### 5.1 System Manager

System Manageria käytetään laitteen kuvauksen luomiseen, eli sillä voidaan määrittellä, minkälaista IO:ta (Input/Output, suom. tulot/lähdöt) ja akseleita laitteesta löytyy. System Managerissa pystytään myös luomaan NC -akselit ja kanavat sekä voidaan liittää haluttu PLC -projekti laitteeseen. Sen jälkeen kun PLC -projekti on liitetty, voidaan PLC:ssä käytetyt tulo- ja lähtömuuttujat linkittää fyysisiin tuloihin ja lähtöihin. Kuvassa 5 on esiteltyä System Managerin aloitusikkuna. /1;5/



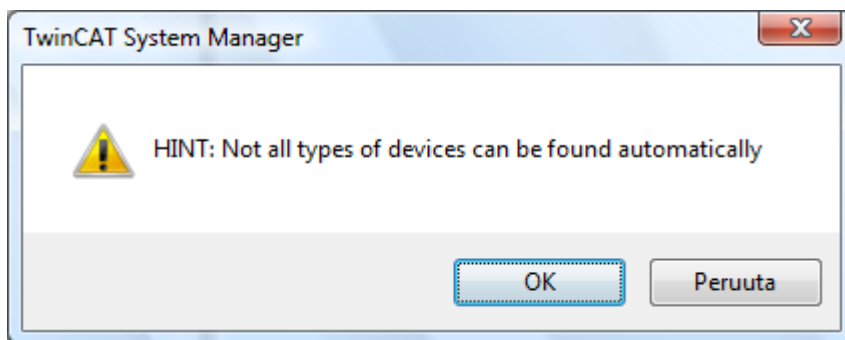
Kuva 5 System Managerin perusnäky

## 5.2 I/O-määrittely

I/O -määrittely voidaan tehdä joko käsin tai skannaamalla. Mikäli fyysistä laitetta ei ole, määrittely tarvitsee tehdä käsin, mutta jos fyysinen laite on olemassa, voidaan käyttää skannausta. /1;5/

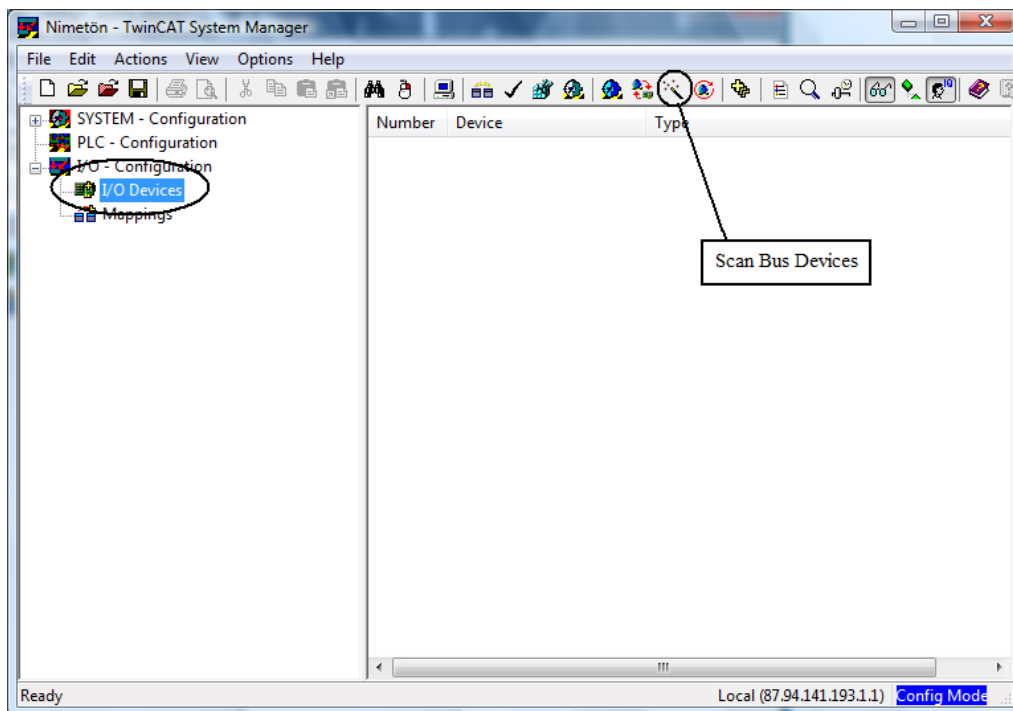
### 5.2.1 Skannaus

Jos fyysinen laite on olemassa, on järkevää käyttää *Scan Devices* -toimintoa. Tätä toimintoa käyttämällä säästyy aikaa sekä välttyy inhimillisiltä virheiltiltä. Skannaus löytää käytännössä kaikki laitteet, vaikka ohjelma varoittaaakin, että kaikkia laitteita ei välttämättä löydetä automaattisesti (kuva 6). /1;5/



**Kuva 6** Varoitusilmoitus

Skannaus on mahdollista vain silloin kun laite on konfigurointitilassa (Config mode). Aloita skannaus aktivoimalla hiiren painikkeella vasemmanpuoleisesta sarakkeesta *I/O Devices* -painike, tämän jälkeen klikkaa *Scan Bus Devices* -painiketta (kuva 7). Skannauksen voi myös suorittaa valitsemalla hiiren oikealla painikkeella *I/O Devices* -painike ja valitsemalla *Scan Devices*.... /1;5/

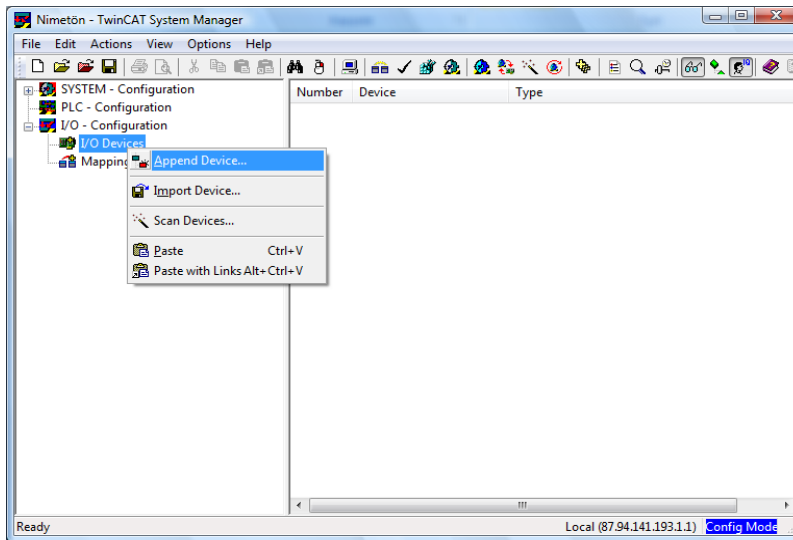


**Kuva 7** Skannauksen aloitus

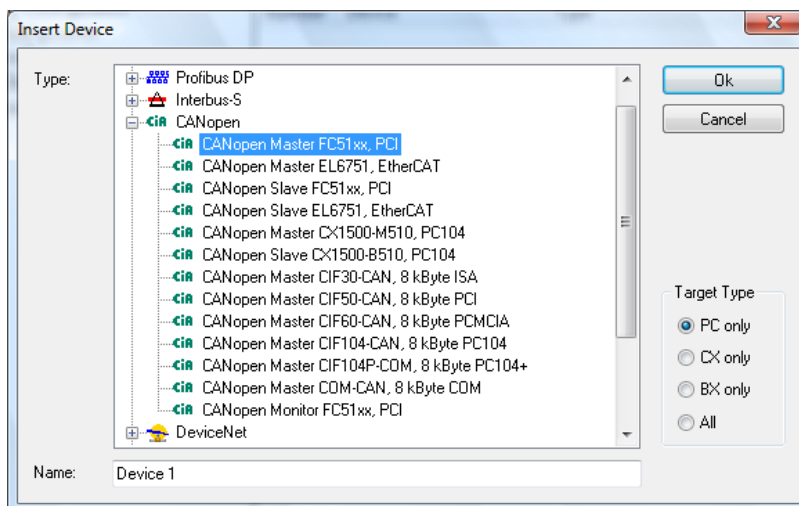
Kun laitteet on skannattu, System Manager kysyy, halutaanko skannata myös väyläterminaalit *Scan for boxes?* Tähän täytyy vastata *Yes*, jos haluaa, että kaikki väyläterminaalitkin etsitään. /1;5/

### 5.2.2 Käsin määrittely

Silloin kun fyysistä laitetta ole käytössä, I/O-määrittely tarvitsee tehdä käsin. Määrittely aloitetaan tekemällä ylimmän tason laite I/O:lle *Append Device...* - valinnalla (kuva 8). Ylimmän tason laitteilla tarkoitetaan esimerkiksi CanOpen master FC5100, Profibus master kortti FC31xx jne, eli laitteita jotka voivat olla jonkin kenttäväylän kommunikointimasterina (kuva 9). /1;5/

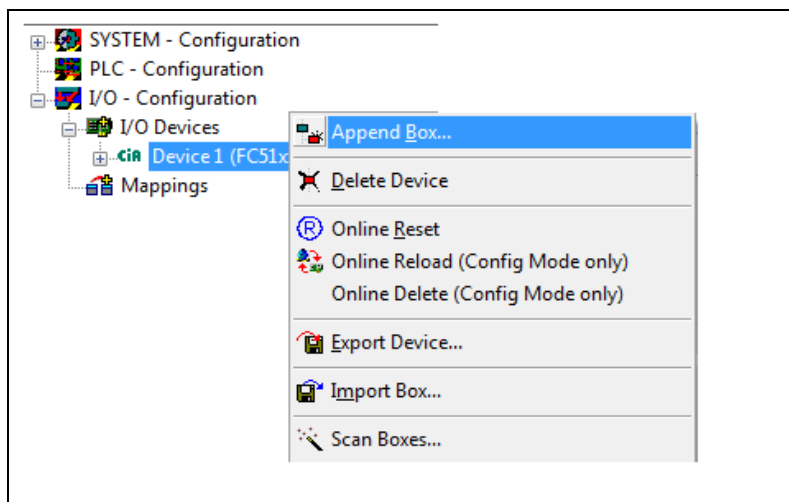


Kuva 8 I/O:n määrittely käsin

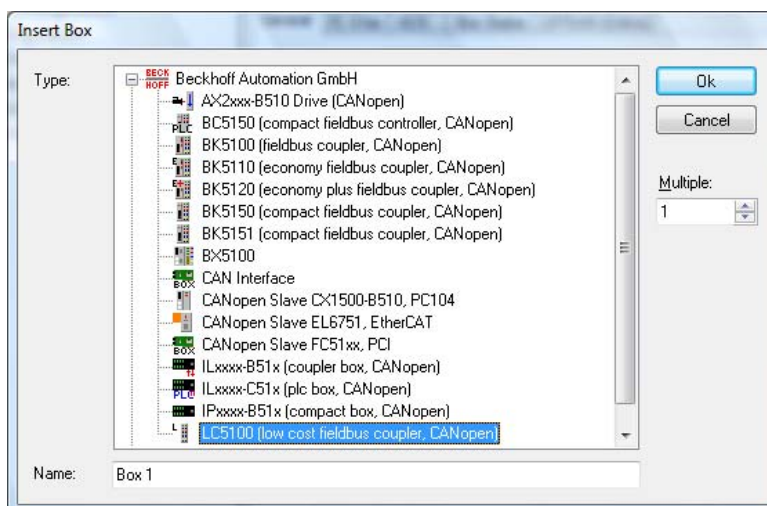


Kuva 9 Ylimmäntason laitteen valinta

Seuraavaksi luodaan väyläterminaali valitsemalla *Append Box...* (kuva 10). Väyläterminaalilla tarkoitetaan sitä laitetta jonka perässä I/O:t sijaitsevat. System Manager poistaa valittavissa olevia väyläterminaaleja sen mukaan, minkä laitteen alle terminaalia ollaan luomassa. Eli CANopen masterin FC5100 alle pystyy luomaan ainoastaan CANopen-väyläterminaaleja tai kontrollereita (kuva 11).



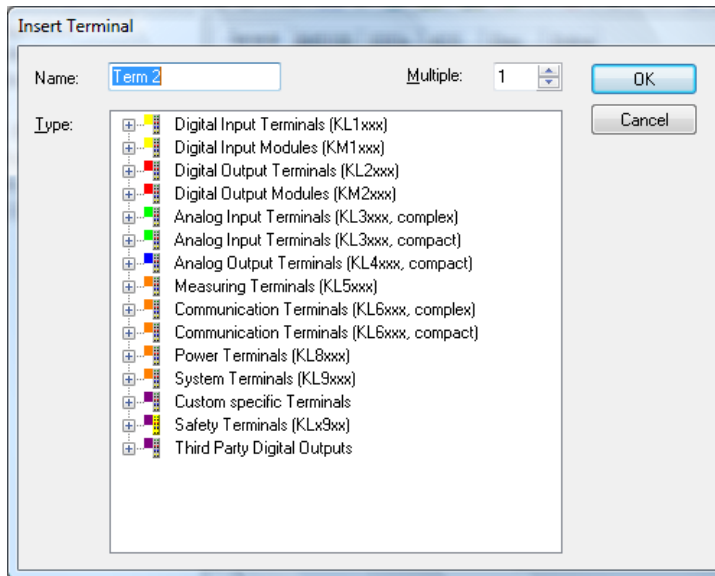
Kuva 10 Terminaalin luonti



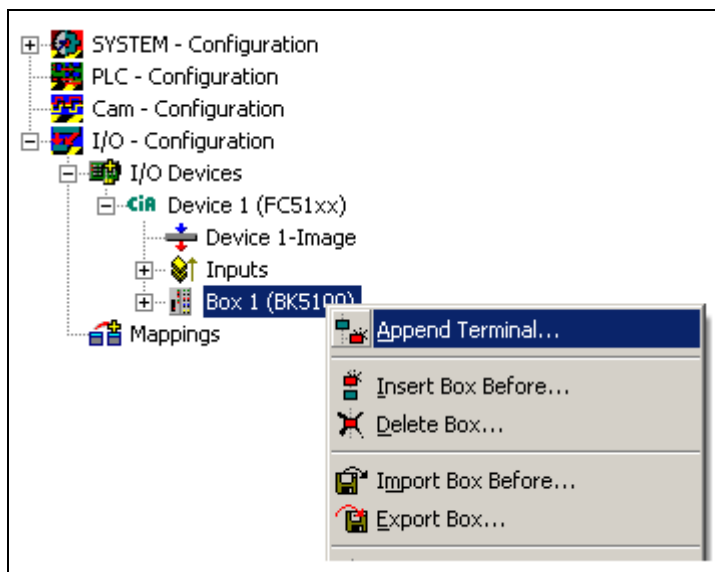
Kuva 11 Väyläterminaalin tai masterin valinta

Väyläterminaalin perään lisätään laitteesta löytyvät I/O:t. ”*Insert Terminal*”-ikkunasta (kuva 11) valitaan, minkä tyyppin kortteja halutaan lisätä. *Multiple*-valinnalla voidaan samantyyppisiä kortteja lisätä useita kerrallaan. Kortin nimen voi muuttaa haluamukseen *Name*-kentässä. *Insert Terminal*-ikkunan (kuva 12) saa auki valitsemalla hiiren oikealla painikkeella valitun väyläterminaalin tai kontrollerin ja valitsemalla *Append Terminal...* (kuva 13). /1;5/





Kuva 12 Insert Terminal

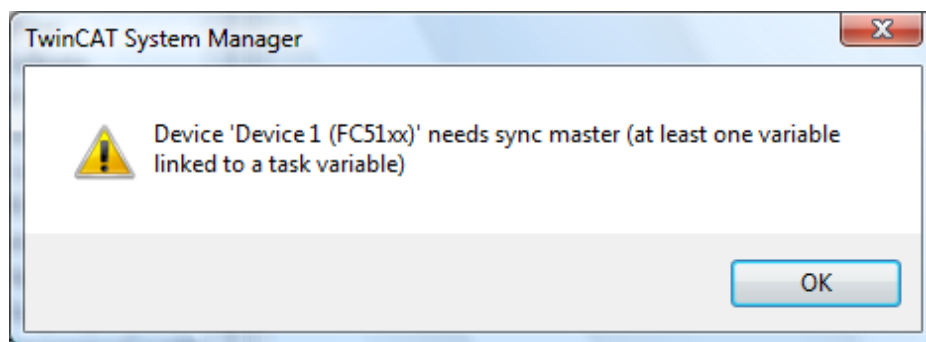


Kuva 13 Append Terminal

### 5.2.3 Määritellyn I/O:n testaus

Kun I/O-määrittely on tehty ja fyysinen laite on kytketty, voidaan laitteen I/O-kortteja testata. Tehty määrittely kannattaa tallentaa *Save*-painikkeella. Tallennuksen jälkeen System Manager -konfiguraatio aktivoidaan *Active Configuration* -painikkeella. /1;5/

Tämän jälkeen System Manager antaa virheilmoituksen (kuva 14), mikä on aivan normaalia, koska I/O tarvitsee synkronointiin jonkin synkronointimasterin eikä PLC projektia ole liitetty konfiguraatioon. Eli jokaiseen I/O -määrittelyssä olevan laitteen alla olevaan I/O:hon täytyy olla vähintään yksi linkitetty PLC –muuttuja, jotta I/O:t päivittyvät.



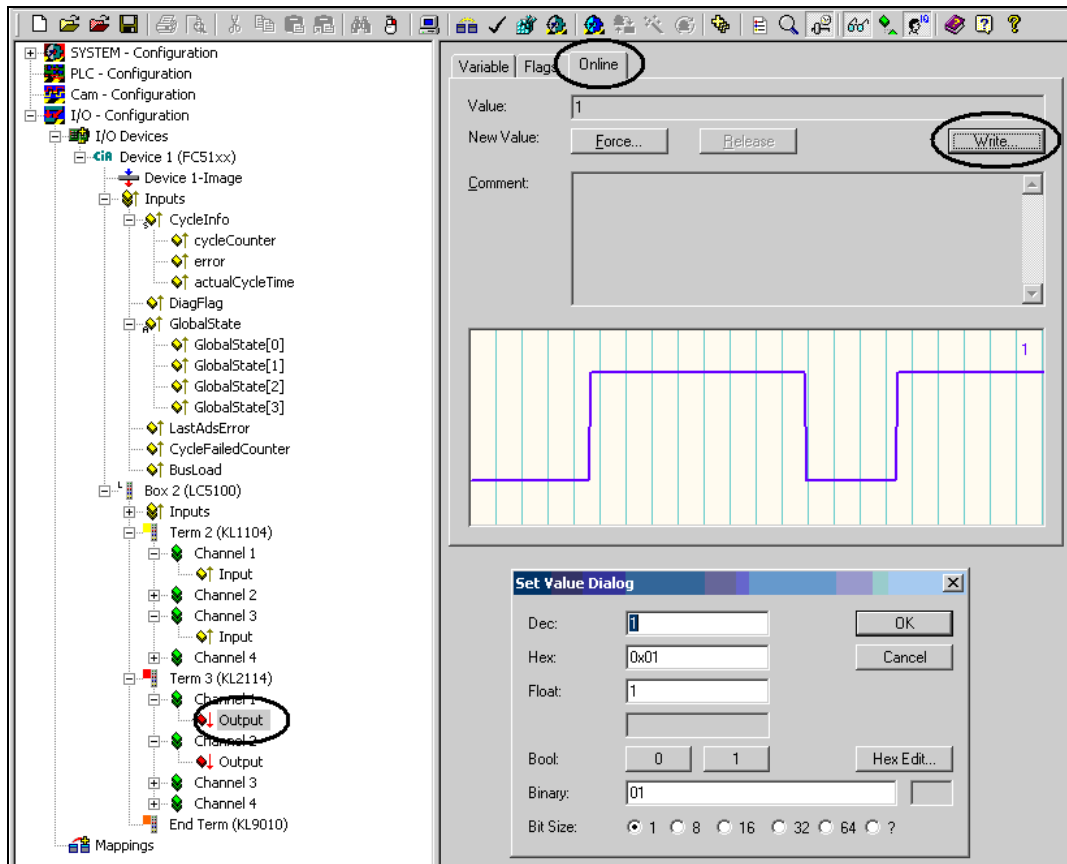
**Kuva 14** System Managerin virheilmoitus

Tämän jälkeen System Manager kysyy, käynnistetäänkö järjestelmä *Run* tilaan. Koska PLC -projektia ei ole vielä tehty, eikä liitetty konfiguraatioon, ei *Run* tilan käynnistämisestä ole hyötyä, eli valitaan *Peruuta*.

I/O-laitteilla ei ole vielä synkronointimasteria, eivät I/O:n tilat päivity. System Managerissa on kuitenkin *Free Run* -toiminto, jolloin System Manager toimii synkronointimasterina. Free Run -toiminnon ollessa päällä oikeassa alareunassa vilkku vuorotellen **Config Mode** ja **Free Run**. I/O-korttien tiloja pystytään nyt tutkimaan ja syöttämään päälle suoraan System Mangerissa.

Vasemmanpuoleisesta sarakkeesta valitaan lähtö, jota halutaan tutkia (kuva 15), jolloin oikeanpuoleiselle sarakkeelle aukeaa näkymä, josta valitaan *Online* -välilehti (kuva 15). Välilehdeltä avautuu aikajana, josta näkee lähdön tilan. *Write* -nappia painamalla avautuu *Set Value Dialog* –ikkuna, josta voidaan syöttää lähtö päälle.

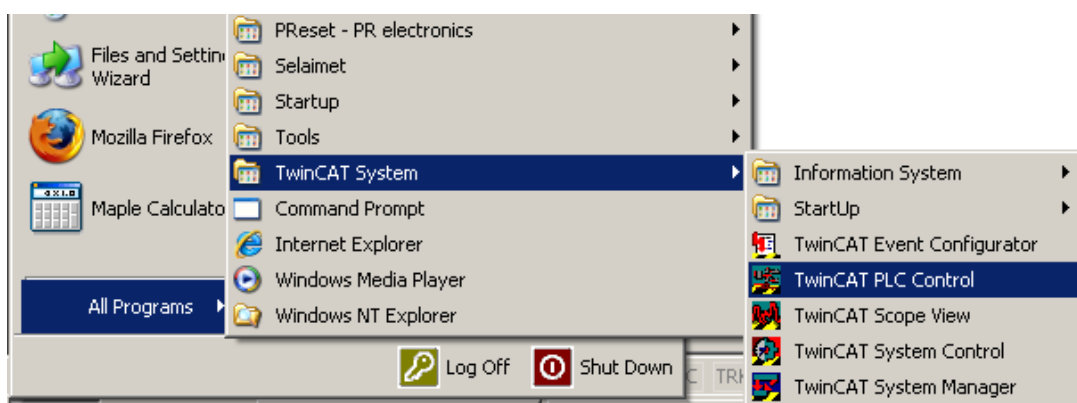
/1;5/



Kuva 15 I/O:n testaus

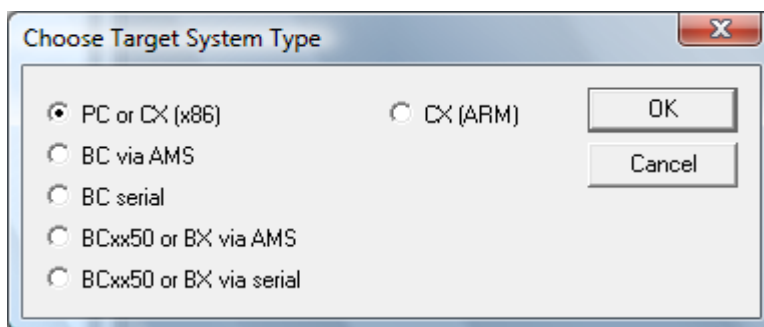
### 5.3 TwinCat PLC-projektin luominen

Uusi PLC-projekti luodaan TwinCAT PLC Control -ohjelmalla (kuva 16).



**Kuva 16** TwinCAT PLC Controlin käynnistys

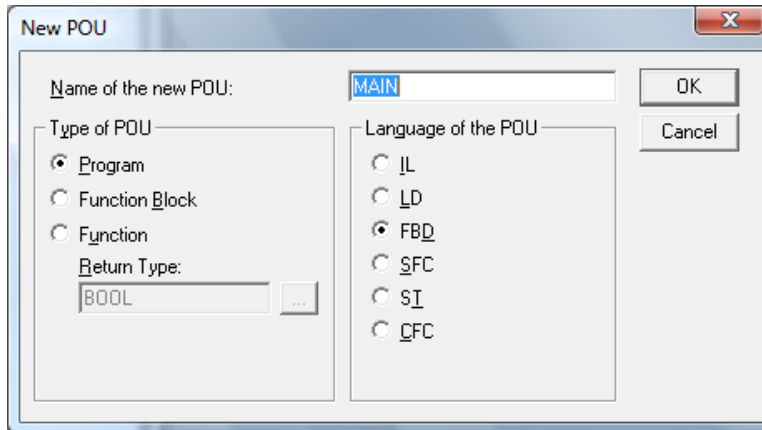
Kun aletaan luoda uutta PLC- projektia, täytyy ensimmäisenä kertoa, mikä on kohdeympäristö, jossa PLC- ohjelmaa tullaan suorittamaan (kuva 17).



**Kuva 17** Kohdeympäristön määrittäminen

Teollisuus-PC:ssä ja sulautetuissa laitteissa valinta on PC or CX (x86). CX9000-sarjan laitteille valitaan CX(ARM). BC- ja BX-väyläkontrolleissa, liitytäänkö niihin sarjakaapelilla (via serial) vai väylän kautta (via AMS).

Seuraavaksi PLC Control ehdottaa POU:n (Program Organisation Unit, suom. ohjelmalohko) luomista (kuva 18).



**Kuva 18** Ohjelmalohkon luominen

*New POU* -ikkunassa annetaan ohjelmalohkolle nimi ja valitaan, millä IEC61131-1 -standardin mukaisista kielistä koodi halutaan kirjoittaa. Vasemmasta reunasta valitaan haluttu lohkotyyppi.

Program- Function Block- ja Function-lohkotyyppien ero:

- Program on pääohjelmatasolla käytetty. Program on globaali ja sitä voi kutsua muista ohjelmalohkoista ilman esittelyä. Program säilyttää paikalliset muuttujat PLC kiertojen välillä. Program:lla ei yleensä ole VAR\_INPUT- VAR\_OUTPUT- tai VAR\_IN\_OUT-tyyppisiä tulo/lähtömuuttujia mutta voi kuitenkin olla. Program voi kutsua toisia Programeja, FB:ejä (Funktion Block, suom. toimilohko) tai Funktioita.
- FB:ta käytetään etenkin sellaisissa ohjelmalohkoissa, joissa samalla koodilla voidaan ohjata useata samankaltaista laitetta. FB täytyy esitellä ja siitä luodaan ilmentymä kutsuvassa ohjelmalohkossa. Function blockille tulee VAR\_INPUT-muuttujina sisäänsyöttömuuttujia, sieltä tulee paluuarvona VAR\_OUTPUT-muuttujia sekä sinne voi tulla VAR\_IN\_OUT-muuttujia. VAR\_IN\_OUT-muuttujat välitetään funktioihin referenssinä (ByRef). Etenkin suuret taulukot tai käyttäjän määrittelemät struktuurit kannattaa viedä FB:iin VAR\_IN\_OUT-tyyppisinä nopeuden ja muistin säästämisen takia.

FB:sta luodut ilmentymät säilyttävät jokainen omat paikalliset muuttujansa PLC kiertojen välissä. FB voi kutsua toisia FB:eja, mutta ei ohjelmia.

- Function on globaali ja sitä voi kutsua sekä Program- että FB-lohkoista ilman esittelyä. Function on tarkoitus käyttää esim. laskentafunktioina joissa VAR\_INPUT-muuttujilla annetaan sisäänsyöttömuuttujat ja funktion paluuarvona palautetaan laskettu arvo. Functionilla ei voi olla VAR\_OUTPUT- tai VAR\_IN\_OUT-tyyppisiä muuttujia. Function ei säilytä paikallisten muuttujien tiloja PLC-kiertojen välissä, ja näin ollen esimerkiksi ajastimia ja laskureita ei voi ohjelmoida function-lohkoihin. Funktio voi kutsua toisia funktioita, mutta ei FB:eja tai Programeja. /1;5/

### 5.3.1 Ohjelmointikielet

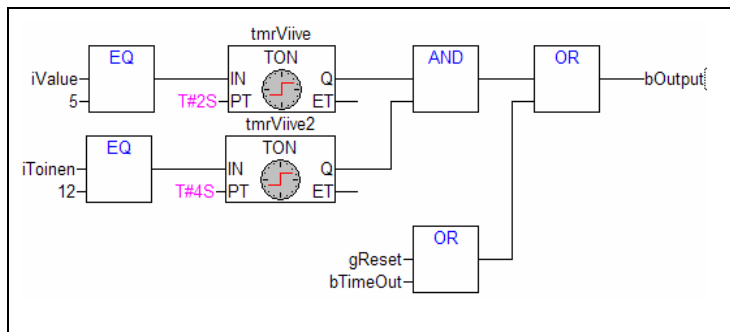
Ohjelman voi kirjoittaa kuudella eri ohjelmointikielellä: IL (Instruction List, suom. käskylista), LD (Ladder Diagram, suom. relekaavio), FBD (Function Block Diagram, suom. toimilohkokaavio), SFC (Sequential Flow Chart, suom. sekvenssikaavio), ST (Structured Text, suom. strukturoitu teksti) ja CFC (Continuous Function Chart).  
/1;5/

#### 5.3.1.1 IL

IL on yksinkertain, mutta rajoittunut ohjelmointikieli. SET/RESET -logiikkaohjaukset ja laskennat pystyy tekemään tiiviissä muodossa, mutta monimutkaisemmat ohjelmat ovat haastavia tehdä. /1;5/



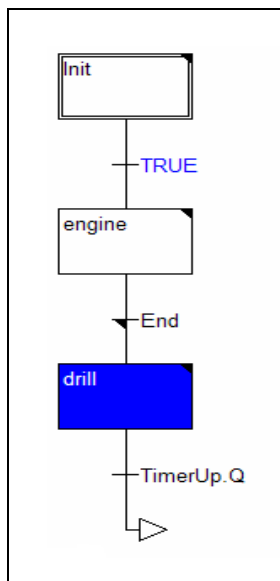
on monesti hyödyllistä PLC-ohjelman ”ylätasolla” eli siellä, missä aliohjelmaa kutsutaan. Etu on myös se, että yhdellä silmäyksellä näkee, mitkä ovat funktiolohkojen sisään menevät ja ulos tulevat muuttujat ja mitkä ovat em. muuttujien arvot. /1;5/



Kuva 21 FBD /1/

#### 5.3.1.4 SFC

Vuokaavio-ohjelmointi on erittäin havainnollinen etenkin askelketjumaisissa toiminnoissa. Askel, jossa koodin suoritus on menossa, käy selkeästi esille sinisestä väristä, samoin kuin siirtymäehto askeleesta toiseen. /1;5/



Kuva 22 SFC /1/



### 5.3.1.5 ST

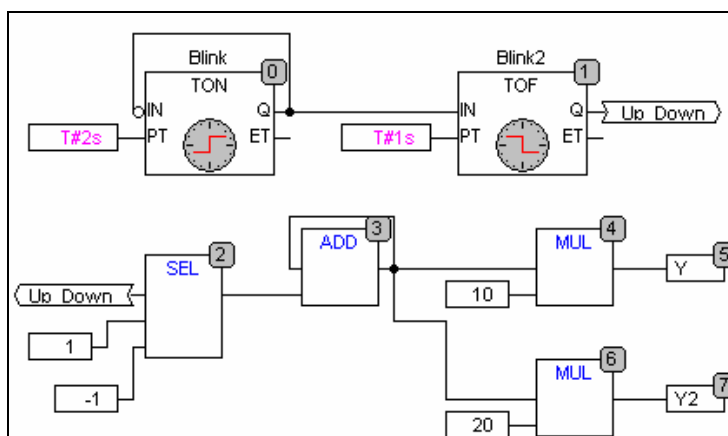
ST on pascal/basic-sukuinen ohjelmointikieli. Tämä on ehdottomasti joustavin ja monelle ohjelmoijalle entuudestaan tuttu ohjelmointikieli. Yksi ST-kielen suurin etu on se, että kommentteja on mahdollisuus kirjoittaa suoraan koodin jokaiselle riville.  
/1;5/

```
(* Erilliset IF ja ELSE lauseet *)  
IF i_VaunuSisaan AND (NOT i_VaunuSisalla) THEN  
  o_VaunuSisaan:=TRUE;  
ELSE  
  o_VaunuSisaan:=FALSE;  
END_IF  
  
(* Saman voi tehdä myös suoraan sijoittamalla *)  
o_VaunuUlos:=i_VaunuUlos AND (NOT i_VaunuUlkona);  
  
(* Vilkkuvalon välkyttämiseen käytetään globaalia vilkkubittia jota hoidetaan FB_FlashByte funktiosta *)  
(* Huomaa, että byte muuttujan yksittäisiin bitteihin voi viitata BYTE.x tyyllillä *)  
IF (o_VaunuSisaan OR o_VaunuUlos) AND gFlashByte.2 THEN  
  o_VilkkuValo:=TRUE;  
ELSE  
  o_VilkkuValo:=FALSE;  
END_IF
```

Kuva 23 ST /1/

### 5.3.1.6 CFC

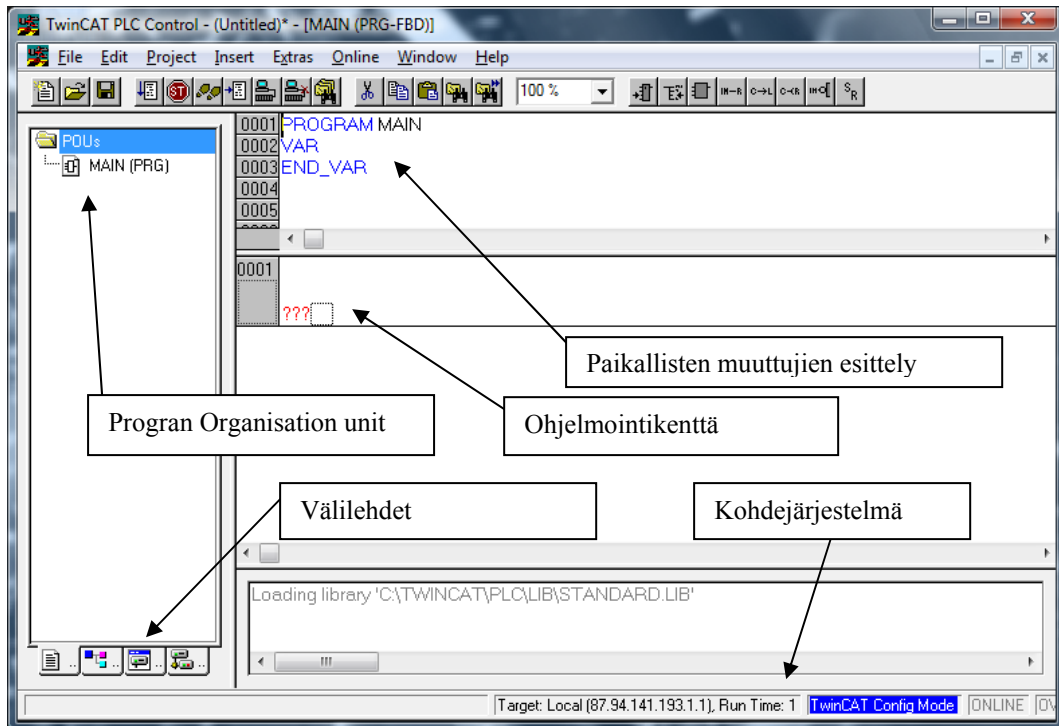
CFC on laajennus FBD:hen. Lohkot eivät ole sidottu gridiin, vaan ne voivat leijua vapaasti ohjelmointi-ikkunassa. Lohkoja voidaan yhdistellä hiirellä raahaamalla. /1;5/



Kuva 24 CFC /1/

## 5.4 Ohjelmointiympäristö

Kuvassa 25 on esiteltyä ohjelmointiympäristöä.













Kuva 25 Ohjelmointiympäristö

#### 5.4.1 Pikakuvakkeet

Taulukkoon 1 on kuvattuna pikakuvakkeet ja niiden toiminnot, jotka ovat aina näkyvissä PLC -ohjelmaa tehdessä.

**Taulukko 1** Pikakuvakkeet ja niiden kuvaus /1/

Pikakuvake	Kuvaus
	Luo uuden projektin
	Avaa tallennetun projektin
	Tallettaa projektin
	Käynnistää PLC -ohjelman (ennen käynnistystä PLC Control oltava Login -tilassa). Pikanäppäin F5
	Pysäyttää PLC -ohjelman. Pikanäppäin Shift + F8
	Siirtää ohjelman suorittamista askeleen eteenpäin ja jää odottamaan seuraavaa painallusta
	Merkitsee valitun ohjelmarivin Breakpoint listaan
	Login, eli luo yhteyden kohdejärjestelmään. Pikanäppäin F11
	Logout, eli katkaisee yhteyden kohdejärjestelmään. Pikanäppäin F12
	Global search, eli etsii määritetyn merkkijonon projektista

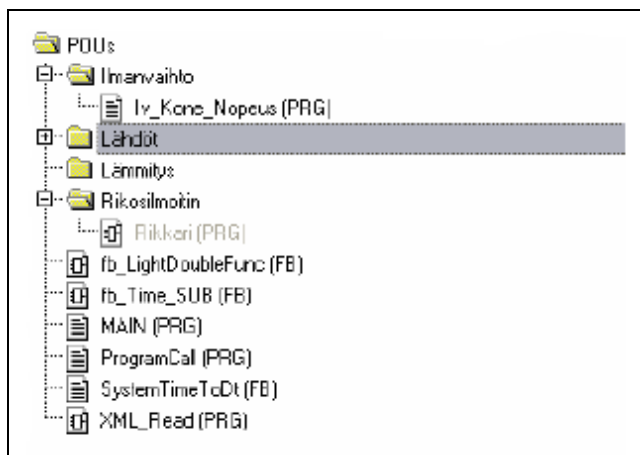
#### 5.4.2 Välilehdet

Ohjelmointiympäristössä on neljä erilaista lohkoa. Lohkojen välillä siirtyminen onnistuu välilehtipainikkeiden avulla (kuva 26).



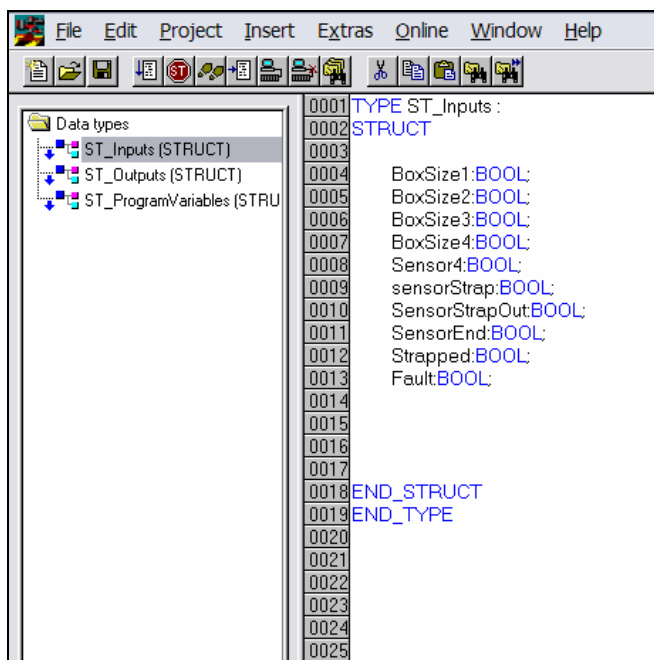
Kuva 26 Välilehtipainikkeet

POU-välilehdeltä näkyvät luodut ohjelmat, funktiot ja FB:t. Ohjelmia voidaan siirtää eri kansioihin, mikä selkeyttää rakennetta. Aliohjelmien sijainti eri kansioissa ei vaikuta suoritusjärjestykseen (kuva 27). /1;5/



Kuva 27 POU

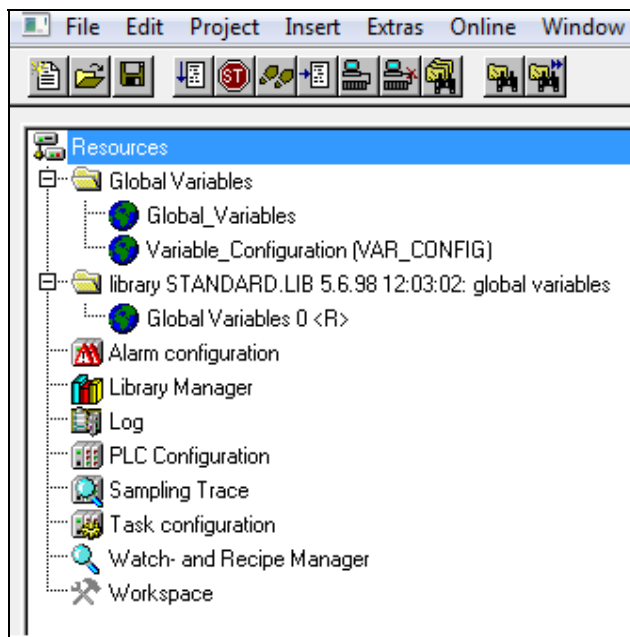
Data Types -kenttään (kuva 28) voi määrittellä omia tietotyyppejä.



Kuva 28 Data Types

Vizualizations on TwinCat PLC -ohjelmistoon sisältyvä käyttöliittymäsovellus.

Resources-välilehdeltä (kuva 29) hallitaan seuraavia asioita:



**Kuva 29** Resources-välilehti

- Global variables-kansiosta löytyvät projektin globaalit muuttujat. Global\_Variables listaa käyttäjä voi halutessaan muokata. Sen lisäksi kansiossa näkyvät listat TwinCAT\_Configuration ja Variable\_Configuration, joita TwinCAT päivittää automaattisesti silloin kuin ohjelmaa käännetään. Global\_Variables-listan lisäksi käyttäjä voi halutessaan luoda omia globaaleja muuttujalistoja.
- Alarm configuration toimii yhdessä Visualizationin Alarm Tab - ominaisuuden kanssa. Työkalulla voidaan määritetää hälytysluokat, prioriteetit, raja-arvot yms.
- Library Manager ottaa käyttöön tai poistaa työkalukirjastoja. Uusien kirjastojen lisääminen tehdään *Insert*-valinnalla. Painallus avaa valikon, jossa näkyvät käytössä olevat työkalukirjastot.

Kirjastoja on kolmea eri tyyppiä, jotka erotetaan tiedostopäätteellä toisistaan. .lb6 -kirjastot ovat BC-laitteille, .lbx-kirjastot ovat BX-laitteille ja .lib-kirjastot ovat PC-pohjaisille järjestelmille, kuten CX-laitteille ja teollisuustietokoneille.

- Log- tiedostoon tallentuvat käyttäjän toimenpiteet PLC ohjelman ollessa Run-tilassa. Logitiedoston kirjoitus asetetaan päälle *Project* → *Log* → *Options* → *Activate Logging*
- PLC Configurations-asetuksella (kuva 17) voidaan muuttaa käytettävää kohdelaitetta. Mikäli kohdelaitetta muutetaan kesken ohjelmoinnin, on myös muistettava tehdä tarvittavat muutokset kirjastoihin.
- Sampling Trace on skooppi, jolla voidaan seurata muuttujia.
- Task Configuration. PLC-ohjelmaan voidaan luoda eri nopeuksilla päivittyviä taskeja. Uusien taskien perustaminen ja vanhojen muokkaaminen tehdään Task Configuration lehdeltä. Eri nopeudella toimivien taskien luominen kuormittaa suoritinta huomattavasti vähemmän. Tehoa jää niille ohjelmille, jotka sitä oikeasti tarvitsevat.

Uusi taski luodaan puunäkymässä *Task Configuration* -otsikon päällä painamalla hiiren oikeaa näppäintä ja valitaan valikosta *Append Task*. Käyttäjä määrittelee itse taskin nimen, prioriteetin sekä sykliajan.

Taskin alle täytyy vielä lisätä ohjelmakutsu. Ohjelmakutsu luodaan painamalla hiiren oikeaa näppäintä uuden taskin päällä ja valitaan *Append program call*. Kutsutusta ohjelmasta tulee pääohjelma ja kaikki pääohjelmassa kutsutut aliohjelmat suoritetaan taskiin määritetyn sykliajan mukaisesti.

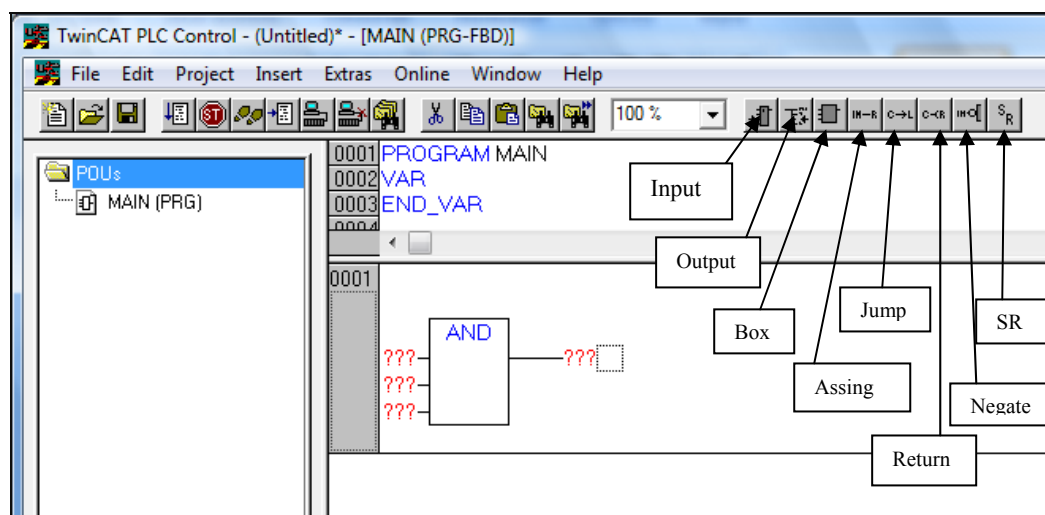
Watch- and Recipe Manager on toiminto, jolla pystytään pakoittamaan useampi muuttuja kerralla päälle. Valmiita muuttujalistoja voidaan tallentaa tiedostoksi.

Toiminto on hyvänä apua PLC-ohjelmaa testatessa, jolloin saadaan useamman muuttujan arvoa vaihdettua kerralla ja monitoroitua eri muuttujia samanaikaisesti.

/1;5/

## 5.5 Ohjelmointi

Ohjelmointitilassa (kuva 30) näyttö jakaantuu kahteen kenttään. Ylemmästä kentästä löytyvät ohjelmakohtaiset muuttujamäärittelyt, ja alemmassa kentässä on varsinainen ohjelma.



Kuva 30 Ohjelmointitila

*Box*-painike luo uuden operaation, joka on oletuksena AND. Painamalla F2-painiketta saadaan käyttöön *Input Assistant*-toiminto (kappale 6.5.2), jolla voidaan valita haluttu operaatio. Peräkkäisiä operaatioita saadaan luotua viemällä kursori tulon/lähdön päälle ja valitsemalla *Box* (kuva 30). Ohjelma luo automaattisesti operaatioiden väliset yhteydet. Painamalla ohjelmointialueella hiiren oikeata nappia ja valitsemalla *Network (After)* (Pikanäppäin Ctrl + T) saadaan luotua uusi virtapiiri edellisen alle. Vastaavasti valinnalla *Networ (Before)* saadaan luotua virtapiiri edellisen yläpuolelle. Tulojen ja lähtöjen muuttaminen negaatioksi tapahtuu viemällä kursori tulon/lähdön päälle ja valitsemalla *Negate* (kuva 30). /1;5/

### 5.5.1 Pikanäppäimet FBD- ohjelmoinnissa

*Ctrl + välilyönti* avaa valikon käytössä olevista muuttujista.

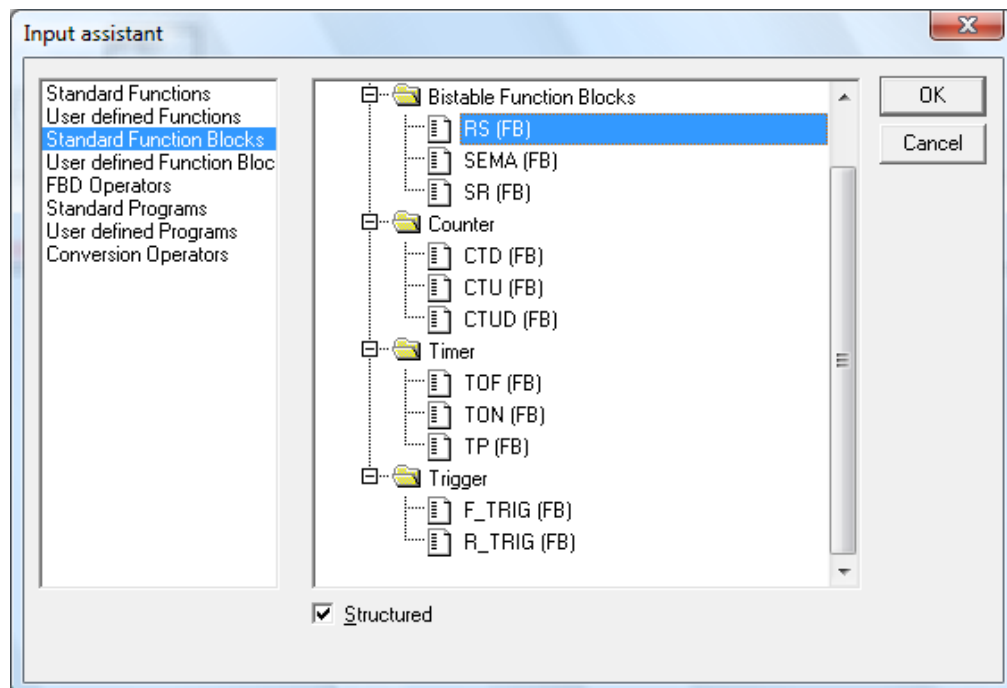
*Ctrl + B* luo uuden ”boxin”.

*Ctrl + A* luo muuttujasijoituksen.

*Ctrl + N* luo negaation.

### 5.5.2 Avustavat toiminnot

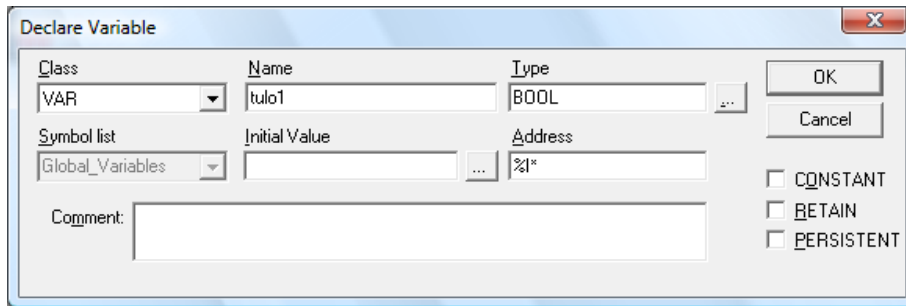
Input Assistant on ominaisuus, joka avustaa muuttujien ja ohjelmalohkojen lisäämisessä. Toiminto näyttää kaikki käytettävissä olevat muuttujat, ohjelmat, funktiot, yms. Toiminnon saa käyttöön *F2* -painikkeella. *Input Assistant* -ikkuna jakaantuu kahteen kenttään (kuva 31). Vasemman puoleisesta valikosta valitaan haluttu toimintoryhmä ja oikeanpuoleisesta valitaan haluttu toiminto. /1;5/



**Kuva 31** Input Assistant

Declare Variable (kuva 32) -ohjelma tunnistaa automaattisesti muuttujan, jota ei ole vielä esitelty. Käyttäjä voi muuttaa määrittämiä, jonka jälkeen muuttuja lisätään muuttuja kenttään. /1;5/





**Kuva 32** Declare Variable

*Class*-kentässä määritellään muuttujaluokka.

*Name*-kenttään kirjoitetaan muuttujalle haluttu nimi.

*Type*-kentässä määritellään muuttujatyyppi.

*Symbol list*-kenttään määritellään mihin listaan globaali muuttuja lisätään. Oletuksena on, että käytössä on vain yksi globaali lista.

*Intial Value*-kentässä määritellään muuttujalle kiinteä arvo.

*Adress*-kentässä määritellään muuttujan osoite, jos se halutaan liittää fyysiseen tuloon tai lähtöön. Automaattista muistiosoitteen määrittelyä käytettäessä osoite annetaan esim. muodossa *%I\**. Jos muistiosoitte määritellään käsin, voidaan se antaa esim. muodossa *%IO*.

### 5.5.3 Aputoiminto

Ongelmatilanteita varten on olemassa kattava Help-aputoiminto, joka saadaan käyttöön *Help*-painikkeella. Aputoiminnosta löytyy tietoa käskyistä, ohjelman luomisesta, konfiguroinnista, ohjelmointikielistä, datatyypeistä ym. toiminnoista. Myös kaikista käytettävissä olevista funktioista löytyy tarkat kuvaukset. /1;5/

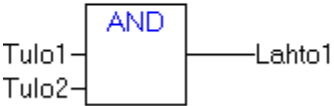
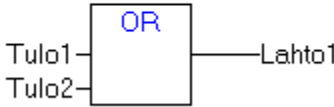

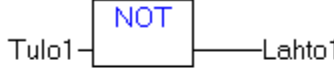
## 5.6 Operaatiot

TwinCat PLC Control -ohjelma luodaan operaatioiden avulla. Ohjelma sisältää runsaasti käskyjä eri tarkoituksiin, tähän lukuun koottu vain yleisimmät ja käytetyimmät operaatiot. Lisätietoa erilaisista käskyistä ja operaatioista löytyy Help-aputoiminnosta (kappale 5.5.3). /1;5/

### 5.6.1 Peruskäskyt

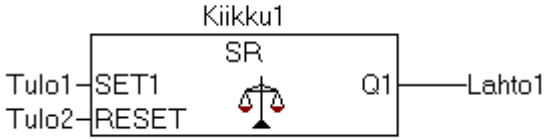
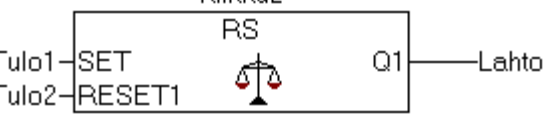
Perusoperaatiot (taulukko 2) löytyvät helpoiten Input Assistant -toiminnolla ohjelmaa luodessa (kappale 6.5.2).

**Taulukko 2** Perusoperaatiot ja niiden kuvaus

Käsky	Kuvaus															
<p>JÄ-käsky</p> 	<table border="1"> <thead> <tr> <th>Tulo1</th> <th>Tulo2</th> <th>Lähtö1</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	Tulo1	Tulo2	Lähtö1	1	1	1	1	0	0	0	1	0	0	0	0
Tulo1	Tulo2	Lähtö1														
1	1	1														
1	0	0														
0	1	0														
0	0	0														
<p>TAI-käsky</p> 	<table border="1"> <thead> <tr> <th>Tulo1</th> <th>Tulo2</th> <th>Lähtö1</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	Tulo1	Tulo2	Lähtö1	1	1	1	1	0	1	0	1	1	0	0	0
Tulo1	Tulo2	Lähtö1														
1	1	1														
1	0	1														
0	1	1														
0	0	0														
<p>EHDOTON TAI-käsky</p> 	<table border="1"> <thead> <tr> <th>Tulo1</th> <th>Tulo2</th> <th>Lähtö1</th> </tr> </thead> <tbody> <tr><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	Tulo1	Tulo2	Lähtö1	1	1	0	1	0	1	0	1	1	0	0	0
Tulo1	Tulo2	Lähtö1														
1	1	0														
1	0	1														
0	1	1														
0	0	0														
<p>EI-käsky</p> 	<table border="1"> <thead> <tr> <th>Tulo1</th> <th>Lähtö1</th> </tr> </thead> <tbody> <tr><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td></tr> </tbody> </table>	Tulo1	Lähtö1	1	0	0	1									
Tulo1	Lähtö1															
1	0															
0	1															

Muistitoiminnot (taulukko 3) toteutetaan muistipiirien avulla. Muistipiirin muisti asetetaan päälle käskyllä SET ja nollataan nollauskäskyllä RESET. Jos sekä SET ja RESET ovat päällä, muistipiirin toiminta määräytyy sen tyyppin mukaan. SR-muistipiirillä on dominoiva asetus, ja vastaavasti RS-kiikulla on dominoiva nollaus. Muistikäskyt löytyvät *Input Assistant* -aputoiminnosta (kappale 6.5.2). /1;5/

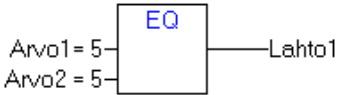

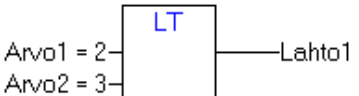
**Taulukko 3** Muistipiirit ja niiden kuvaus /3/

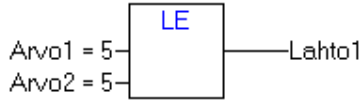
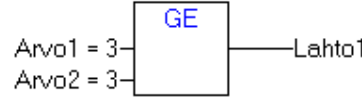
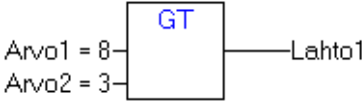
Käsky	Kuvaus
<p>SR-kiikku</p> 	<p>Kun molemmat tulot ovat tilassa 1, lähtö on signaalitilassa 1. Kun pelkkä RESET on tilassa 1, lähtö on signaalitilassa 0.</p>
<p>RS-kiikku</p> 	<p>Kun molemmat tulot ovat tilassa 1, lähtö on signaalitilassa 0. Kun pelkkä RESET on tilassa 1, lähtö on signaalitilassa 1.</p>

### 5.6.2 Vertailijat

Vertailijat (taulukko 4) löytyvät *Input Assistant* -aputoiminnolla (kappale 6.5.2).

**Taulukko 4** Vertailukäskyt ja niiden kuvaus /3/

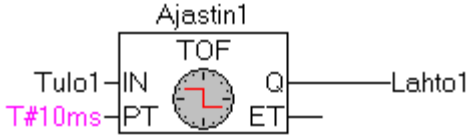
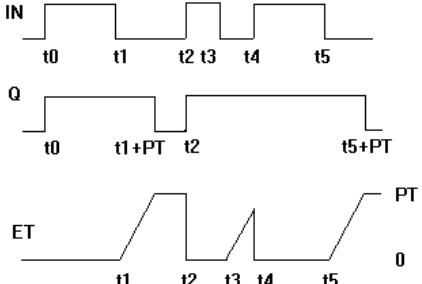
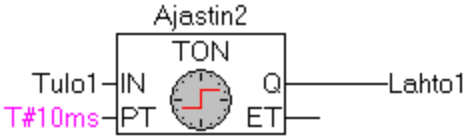
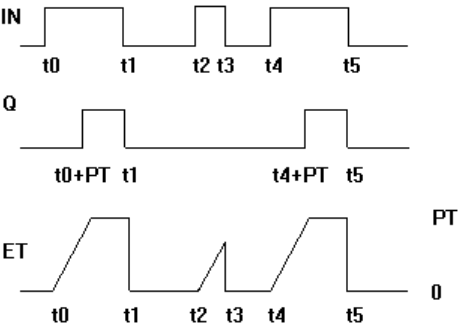
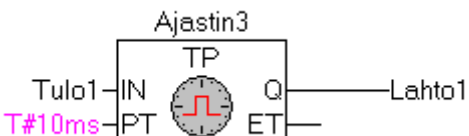
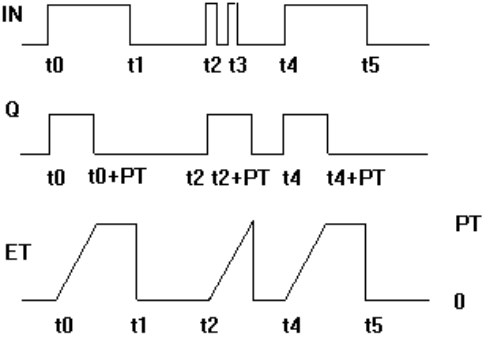
Käsky	Kuvaus
	<p>Yhtäsuuri-käsky                      Arvojen 1 ja 2 yhtäsuuret lähtö 1 on päällä</p>
	<p>Erisuuri-käsky                      Arvojen 1 ja 2 erisuuret lähtö 1 on päällä</p>
	<p>Pienempi kuin-käsky                      Kun arvo 1 on pienempi kuin arvo 2, lähtö 1 on päällä</p>

	<p>Pienempi tai yhtäsuuri kuin-käskey</p> <p>Kun arvo 1 on pienempi tai yhtäsuuri kuin arvo 2, lähtö 1 on päällä</p>
	<p>Suurempi tai yhtäsuuri kuin-käskey</p> <p>Kun arvo 1 on suurempi kuin arvo 2, lähtö 1 on päällä</p>
	<p>Suurempi kuin-käskey</p> <p>Kun arvo 1 on suurempi kuin arvo 2, lähtö 1 on päällä</p>

### 5.6.3 Ajastimet

Ajastimet (taulukko 5) löytyvät *Input Assistant*-tominnolla (kappale 6.5.2) muiden operaatioiden tavoin.

**Taulukko 5** Ajastintyypit /3/

Ajastin	Kuvaus
<p style="text-align: center;">Ajastin1</p> 	<p>IN = ajastimen käynnistys                  PT = ajastettava arvo                  ET = näyttää ajan                  Q = ilmoittaa kun aika on kulunut</p> 
<p style="text-align: center;">Ajastin2</p> 	<p>IN = ajastimen käynnistys                  PT = ajastettava arvo                  ET = näyttää ajan                  Q = ilmoittaa kun aika on kulunut</p> 
<p style="text-align: center;">Ajastin3</p> 	<p>IN = ajastimen käynnistys                  PT = ajastettava arvo                  ET = näyttää ajan                  Q = ilmoittaa kun aika on kulunut</p> 

5.6.4 Laskurit

Taulukkoon 6 on kuvattuna erilaisia laskureita. Laskurit 1 ja 2 eivät tarvitse erillistä resetointikäskyä kuten laskuri 3.

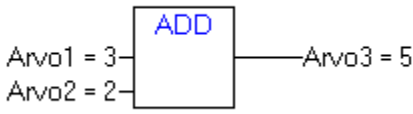
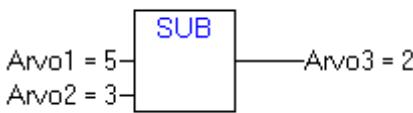

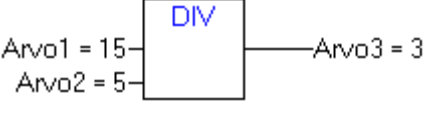
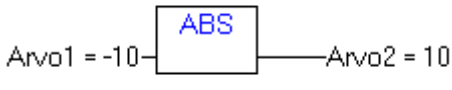
**Taulukko 6** Laskurityypit ja niiden kuvaus /3/

Laskuri	Kuvaus
<p style="text-align: center;">Laskuri1</p>	<p>Alaspäin laskeva laskuri. Laskuriarvo (CV) pienenee tulolla 1. Lähtö1 on päällä, kun laskuriarvo pienempi tai yhtäsuuri kuin nolla. Esiasetusarvolla (PV) saadaan laskuri laskemaan halutusta arvosta alaspäin. Esiasetusarvo saadaan ladattua tulolla 2, tulon 1 ollessa päällä.</p>
<p style="text-align: center;">Laskuri2</p>	<p>Ylöspäin laskeva laskuri. Laskuriarvo (CV) kasvaa tulolla 1. Lähtö1 on päällä, kun laskuriarvo on saavuttanut esiasetusarvon (PV). Laskuri resetoidaan tulolla 2.</p>
<p style="text-align: center;">Laskuri3</p>	<p>Laskuriarvo (CV) kasvaa tulolla 1, ja pienenee tulolla 2. Lähtö1 on päällä, kun laskuriarvo on suurempi tai yhtäsuuri kuin Esiasetusarvo (PV). Lähtö2 on päällä, kun laskuriarvo on pienempi tai yhtäsuuri kuin nolla. Esiasetusarvo ladataan tulolla 4, tuojen 1 ja 2 ollessa päällä. Laskuriarvo resetoidaan tulolla 3.</p>

### 5.6.5 Aritmetiikkakäskyt

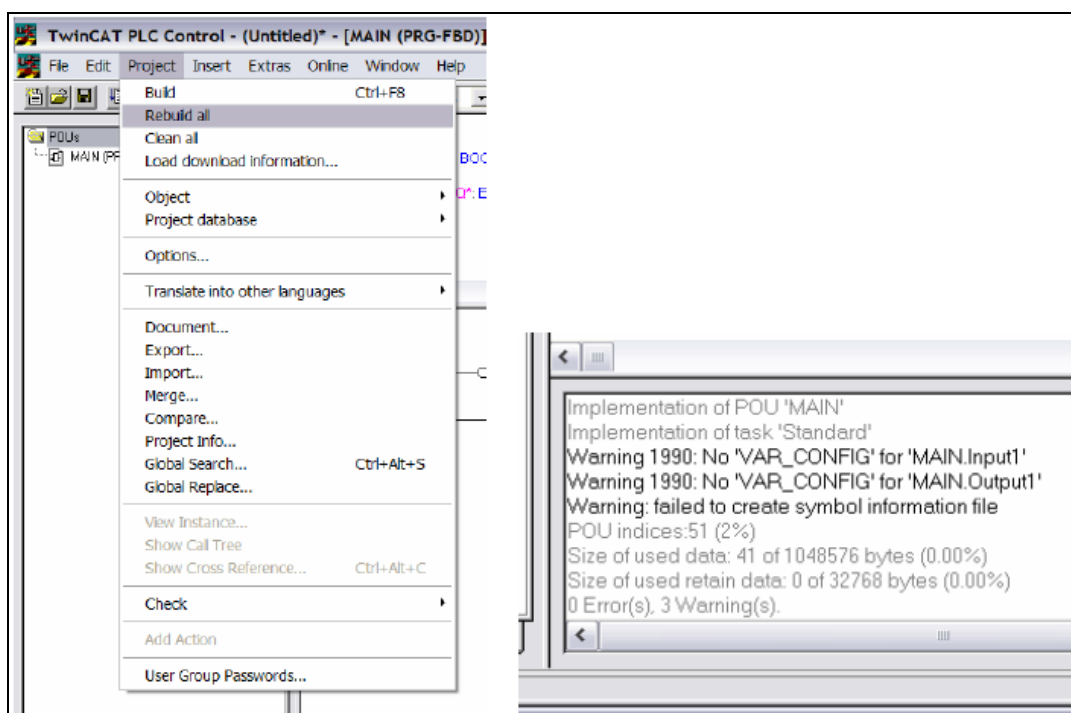
Aritmetiikkakäskyt eli laskentakäskyt (taulukko 7) löytyvät myös *Input Assistant*-aputoiminnolla (kappale 6.5.2).

**Taulukko 7** Laskentakäskyt /3/

Käsky	Kuvaus
	ADD-operaatio suorittaa arvojen 1 ja 2 yhteenlaskun. Tulos näkyy arvona 3
	SUB-operaatio suorittaa arvojen 1 ja 2 vähennyslaskun. Tulos näkyy arvona 3
	MUL-operaatio suorittaa arvojen 1 ja 2 kertolaskun. Tulos näkyy arvona 3
	DIV-operaatio suorittaa arvojen 1 ja 2 jakolaskun. Tulos näkyy arvona 3
	ABS-operaatio muuttaa arvon 1 positiiviseksi arvoksi.

## 5.7 Ohjelman lataaminen järjestelmään

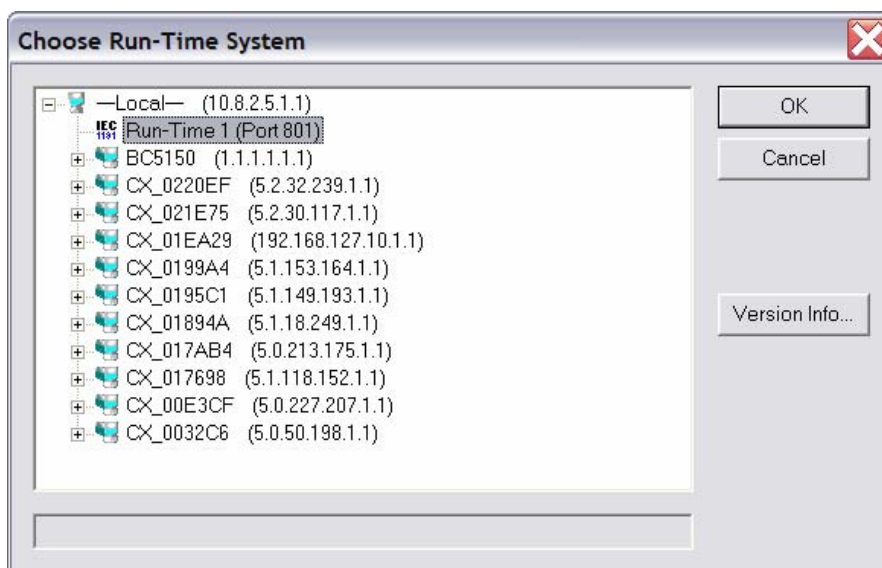
Kun PLC-ohjelmaa ensimmäistä kertaa ladataan järjestelmään, pitää PLC -koodi kääntää. Ennen käännöksen aloittamista ohjelman pitää olla tallennettuna, mieluiten paikkaan, josta se on helppo löytää. Ohjelman käännös tapahtuu *Project*-valikon *Rebuild all* -toiminnolla (kuva 33). Jos käännös onnistuu virheettömästi, ohjelma luo tyyppipäätteisen tiedoston samaan kansioon, johon ohjelma tallennettiin. *Warning(s)* -tekstin (kuva 33) ilmoittamien varoitusten määrä ei estä ohjelman liittämistä System Manageriin. Varoitukset poistuvat heti, kun ohjelma on saanut System Managerilta muistiosoitteet. Muistiosoitteiden haku tapahtuu automaattisesti. /1;5/



Kuva 33 Ohjelman kääntäminen

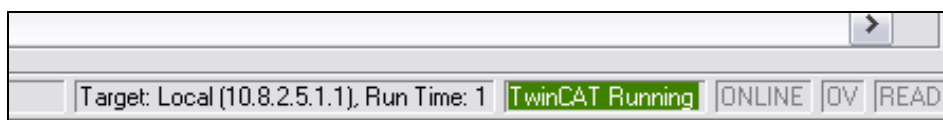
Kääntämisen jälkeen valitaan kohdejärjestelmä, johon ohjelma halutaan ladata. Tässä vaiheessa System Manager tarvitsee kääntää RUN -tilaan. Haluttu kohdejärjestelmä valitaan *Online* -valikosta löytyvällä *Choose Run-Time System...* -valinnalla. Avautuvasta ikkunasta (kuva 34) valitaan haluttu kohdejärjestelmä. Valintaa ei tarvitse tehdä kuin ensimmäisellä kerralla ohjelmaa ladattaessa. /1;5/





**Kuva 34** Kohde järjestelmän valitseminen

Valittu järjestelmä näkyy sivun oikeassa alalaidassa olevassa palkissa (kuva 35). **TwinCAT Running** -tekstistä ilmenee System Managerin tila. Tässä tapauksessa (kuva 35) System Manager on Run tilassa, jossa ohjelma pystytään lataamaan järjestelmään. /1;5/



**Kuva 35** System Managerin tila

Ohjelma ladataan järjestelmään *Online*-valikosta löytyvällä *Login*-painikkeella (pikanäppäin F11). Samanaikaisesti PLC Control siirtyy myös Online-tilaan, jolloin ohjelman kulkua pystytään helposti monitoroimaan. PLC-ohjelma ei ole vielä tässä vaiheessa RUN –tilassa, vaan ohjelma pitää käynnistää *Online*-valikosta löytyvällä *Run*-painikkeella (pikanäppäin F5). Kun ohjelma on siirretty RUN-tilaan ja PLC-Control on Online-tilassa, alapalkissa näkyvät kuvan 36 mukaiset merkinnät. /1;5/

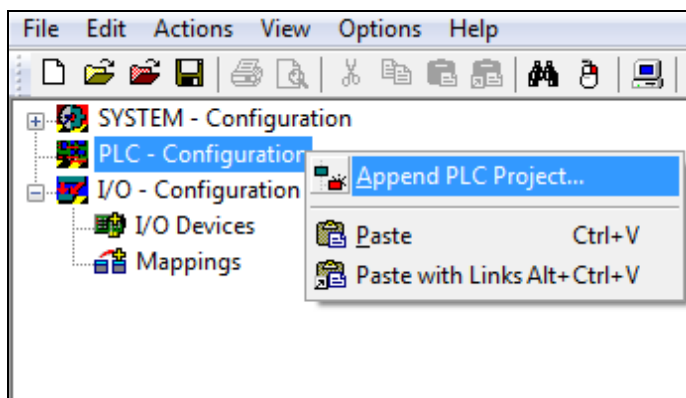


**Kuva 36** Alapalkki

Jos ohjelmaa halutaan muuttaa sen ollessa RUN-tilassa, pitää *Online*-valikosta valita *Logout* (pikanäppäin F12). Näin toimimalla ohjelma jää kuitenkin RUN-tilaan ja haluttuja muutoksia pystytään tekemään. Muutosten lataaminen järjestelmään tapahtuu jälleen valitsemalla *Online*-tila.

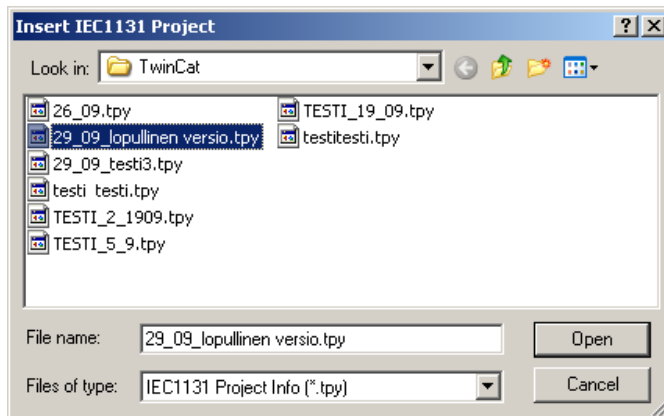
## 5.8 PLC-projektin liittäminen konfiguraatioon

Kun PLC-projekti on luotu, tallennettu ja kertaalleen käännetty TwinCAT PLC Controlilla, voidaan PLC-projekti liittää osaksi System Manager -konfiguraatiota. PLC-projektin liittäminen aloitetaan luomalla ensin PLC-projekti *Append PLC Project...* -valinnalla (kuva 37). /1;5/



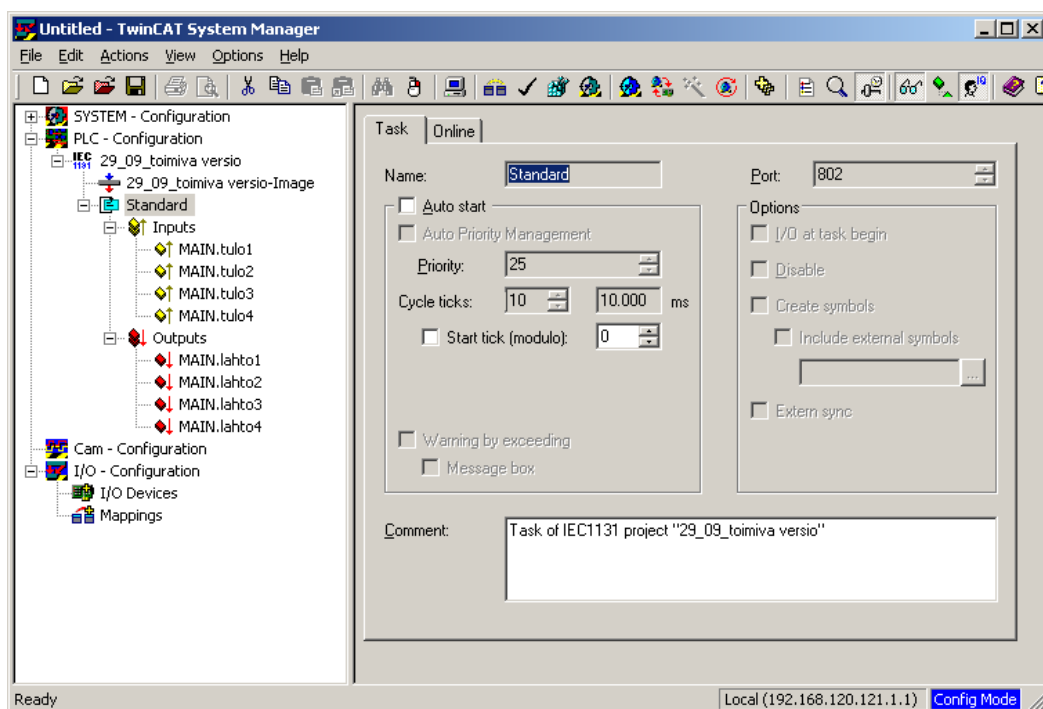
**Kuva 37** PLC -projektin liittäminen konfiguraatioon

PLC -projektiksi valitaan se *tpy* -päätteinen tiedosto siitä kansioista, johon PLC-projekti TwinCAT PLC Controlissa on tallennettu (kuva 38). *Tpy*-päätteinen tiedosto syntyy itsestään PLC-projektista käänösvaiheessa (kappale 6.7).



Kuva 38 Projektin valinta

Riippuen luodusta PLC-projektista näkyviin tulee joko yksi tai useampia PLC-taskeja. Esimerkkikuvassa 39 on yksi taski.

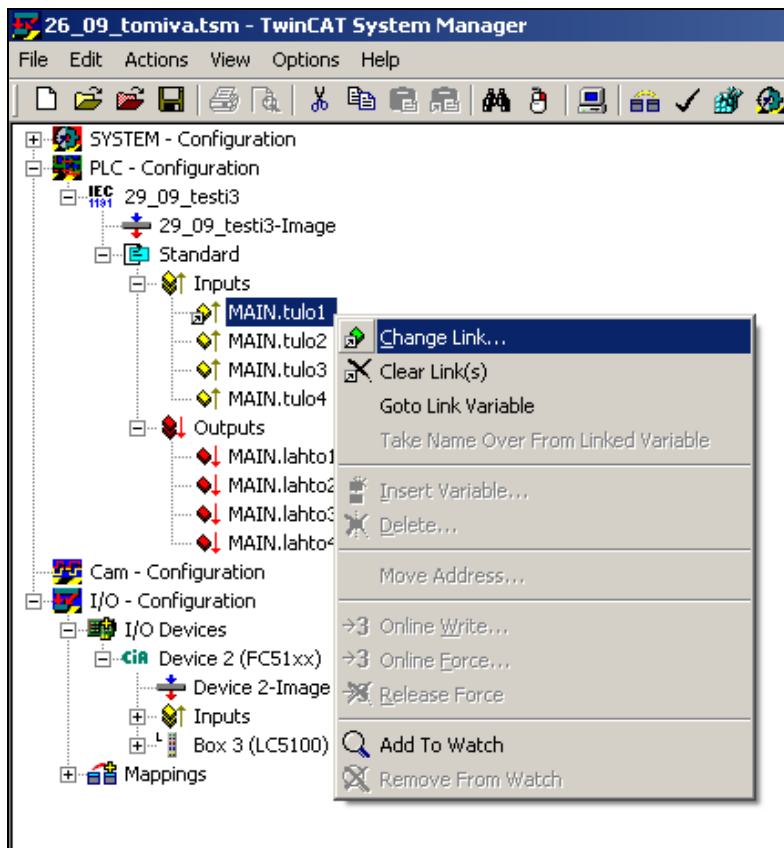


Kuva 39 PLC-taskit

Jos PLC-projektissa halutaan käyttää nopeata taskia jonkin asian hoitamiseen, pitää nopean taskin tarvitsemat I/O:t siirtää ”Drag&Drop” -toiminnolla *Standard*-taskista nopeaan taskiin. Jos siirtoa ei tehtäisi, I/O:t päivittyisivät vakiotaskin mukaan, eikä nopeasta taskista olisi varsinaisesti mitään hyötyä. /1;5/

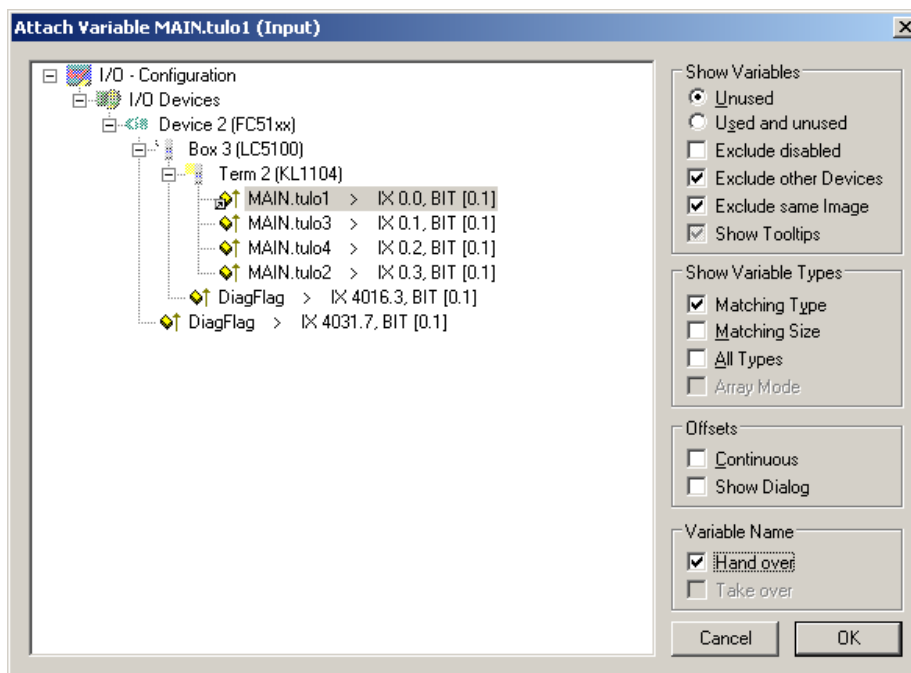
## 5.9 PLC-muuttujien linkittäminen

Jotta tulot saadaan päivitettyksi PLC-muuttujiin ja että PLC:n lähdöt kirjautuisivat oikeisiin lähtöihin, täytyy PLC:n muuttujat linkittää fyysisiin tuloihin ja lähtöihin. Linkitys voidaan tehdä joko I/O-konfiguraatiosta PLC:hen tai PLC:stä I/O-konfiguraatioon päin. Suunnalla ei ole mitään merkitystä. /1;5/



Kuva 40 Muuttujien linkittäminen

Linkitys tehdään joko klikkaamalla haluttua muuttujaa hiiren oikealla painikkeella ja valitsemalla *Change Link...*, tai klikkaamalla oikeanpuoleisessa ikkunassa *Linked to...* -painiketta. Olemassa olevia ja luotuja linkityksiä pääsee muuttamaan ja korjaamaan samaa kautta. Seuraavaksi avautuvasta ikkunasta valitaan linkitettävä I/O (kuva 41). /1;5/



**Kuva 41** Linkitettävä I/O

Ikkunan oikeaan reunaan ilmestyy erilaisia suodattimia, joilla voidaan rajoittaa muuttujien tai I/O:den määrää vasemman puoleisessa ikkunassa (kuva 41):

- *Unused* -valinnalla listassa näytetään vain ne muuttujat tai I/O:t joita ei ole vielä linkitetty mihinkään tuloon tai lähtöön.
- *Used and unused* -valinnalla listassa näytetään sekä jo linkitetty että vapaina olevat muuttujat.
- *Exclude disabled* -valinnalla listasta deletoidaan ne I/O:t, jotka on disabloitu I/O konfiguraatiosta.
- *Exclude other Devices* -valinnalla listasta poistetaan muut kuin I/O laitteet.
- *Exclude same Image* -valinnalla listasta poistetaan se laite (PLC-taski), josta ollaan linkitystä tekemässä.
- *Matching Type* -valinnalla näytetään vain sopivaa tyyppiä olevat I/O:t tai muuttujat.
- *Matching Size* -valinnalla näytetään vain sopivaa kokoa olevat I/O:t tai muuttujat.
- *All Types* -valinnalla näytetään kaiken tyyppiset I/O:t tai muuttujat.
- *Hand Over* -valinnalla PLC muuttujan nimi kirjoitetaan myös I/O:ssa olevan tulon nimeksi silloin kun linkitetään PLC:stä I/O:hon päin.

- *Take Over* -valinta on päin vastainen kuin *Hand Over*, eli kun linkitetään I/O:sta PLC-muuttujiin päin.

Jos linkitettävä PLC-muuttuja on esimerkiksi tavun kokoinen ja tulokortit ovat kaksikanavaisia, voidaan valita listasta neljä tulokorttia ja laittaa ruksi kohtaan *Offsets Continuous*, jolloin linkitettävän tavun bitit linkitetään järjetyksessä tulokortteihin. Jos offsetteja halutaan itse muuttaa, voidaan valita kohta *Show Dialog*, jolloin avautuu ikkuna, jossa offsetit pystytään määrittämään. /1;5/

## 6 TULOKSET

Tämän työn tuloksena syntyi kappaleessa 5 esitetty ohjekirja, jonka nimi on TwinCAT-logiikkaohjelman käyttö. Ohjekirja sisältää käyttöohjeet automaatio-ohjelmalle, sekä antaa tietoa sen soveltamisesta CANopen-kenttäväylään. Opinnäytetyö antaa myös lukijalleen jonkin verran tietoa Beckhoffista, ohjelmoitavista logiikoista sekä muista automaatioon ja väyläohjaimiin liittyvistä asioista.

Opinnäytetyössä kuvattua kenttäväylän liittämistä automaatio-ohjelmaan testattiin käytännössä ja se todettiin toimivaksi. Toivottavasti opinnäytetyö auttaa tulevaisuudessa Beckhoff Oy:n asiakkaita ja Tampereen ammattikorkeakoulun opettajia ja opiskelijoita TwinCATin ja väyläohjainten käytössä.

## 7 TULOSTEN TARKASTELU

Opinnäytetyön tavoitteena oli luoda perusteet sisältävä ohjekirja TwinCATin käytöstä sekä antaa ohjeistus siihen, kuinka logiikkaohjelma liitetään väyläterminaaliin. Opinnäytetyö on täyttänyt ne tavoitteet, jotka sille asetettiin. Se on tiivis ja helposti omaksuttava kokonaisuus aiheesta. Asioiden oppimista edesauttavat monipuoliset kuvat ja esimerkit. Ohjekirjan toimivuus on testattu koekäytössä, mutta varsinaiselta kohderyhmältä palautetta ohjekirjan toimivuudesta ei luonnollisestikaan vielä ole.

Ohjekirjan kirjoittaminen sujui ilman suurempia ongelmia. Koska sivumäärä on rajallinen, ihan kaikkia aihepiirejä ei voitu esitellä kattavasti. Eniten vaikeuksia tuottikin monimutkaisten asioiden selittäminen lyhyesti, mutta kuitenkin ymmärrettävästi. Ehkä tulevaisuudessa tälle työlle tulee jatko-osa, jossa syvennyttään tarkemmin logiikkaohjelman käyttöön.

Käsitellyt ovat siinä järjestyksessä kuin niiden oletetaan olevan hyödyllisiä ohjelman käytön kannalta. Turhia toistoja työssä on pyritty välttämään, jos sama aihe on tullut esille myöhemmissä osioissa, tekstissä on viittaukset kohtiin, joissa aihe on selitetty ensimmäistä kertaa.



## LÄHDELUETTELO

### Painetut lähteet

- 1 Beckhoff Automation Oy. TwinCAT peruskurssi. [kurssimoniste]. Hyvinkää 2008 44 s.
- 2 Nordvall, Seppo, Opinnäytetyö, CAN-väylän tutkiminen. Tampere, 2003 43s
- 3 Heinonen, Anu, Opinnäytetyö, GX IEC DEVELOPER FX-Ohjekirja ohjelmointiin ja ohjelman testaaminen. Tampere, 2006 62s

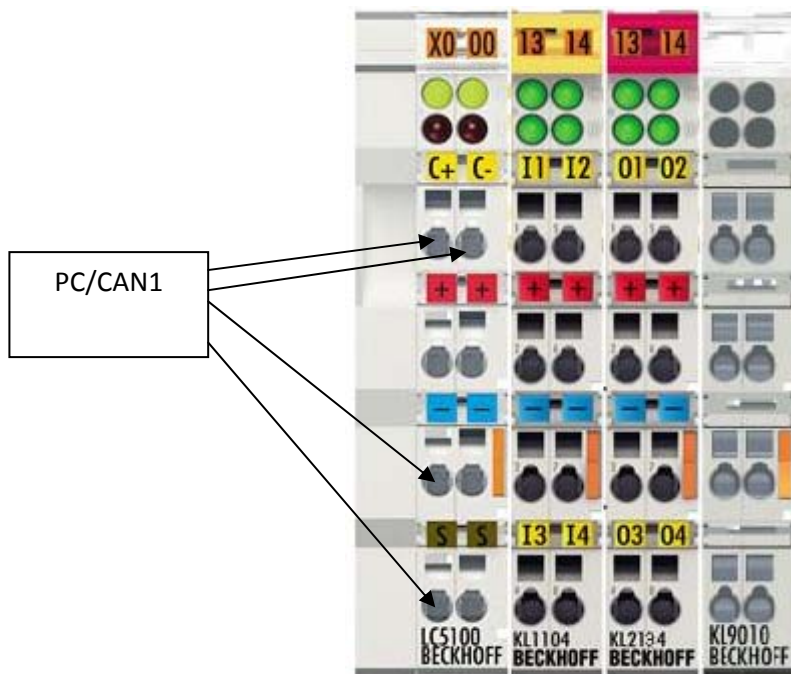
### Painamattomat lähteet

- 4 VTT Automaatio, Koneautomaatio, Jarmo Alanen CAN- ajoneuvojen ja koneiden sisäinen paikallisväylä [PDF-dokumentti]. Tampere, 2000 17s
- 5 Kalajainen, Aki, TwinCAT -ohjeistus [keskustelu]. Tampereen ammattikorkeakoulu 24.10.2008 ja 19.9.2008

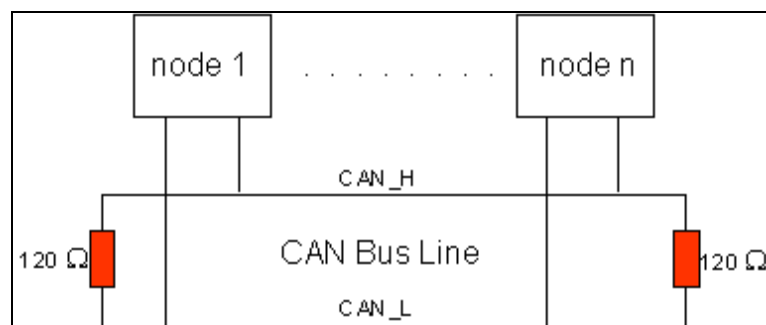
### Sähköiset lähteet

- 6 CAN in Automation (CiA) [www-sivu]. Saatavissa: <http://www.can-cia.org/can/>
- 7 Beckhoff Automation GmbH. [www-sivu]. Saatavissa: [www.beckhoff.com](http://www.beckhoff.com)
- 8 Teho, Valle, CAN-väylämateriaali. [sähköpostiviesti]. 23.6.2008
- 9 Mäkelä, Seppo, CAN-väylämateriaali. [sähköpostiviesti]. 15.5.2008

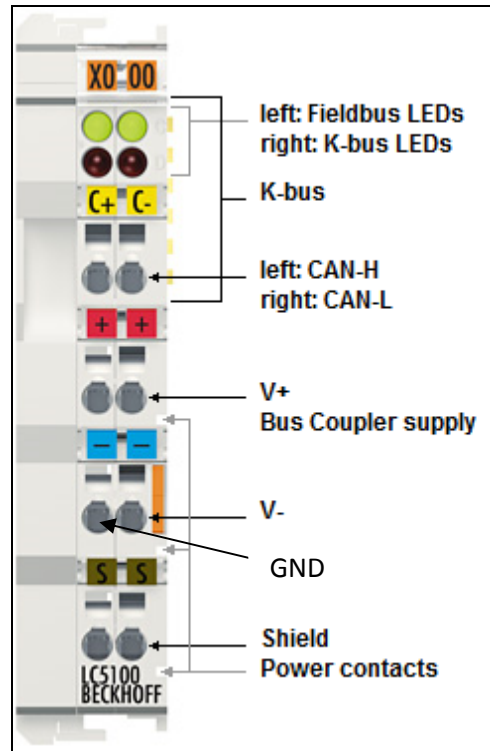
### CANopen väyläliitin



Kuvasta poiketen CAN-H:n ja CAN-L:n välille on kytkettävä 120 ohmin päätevastus.



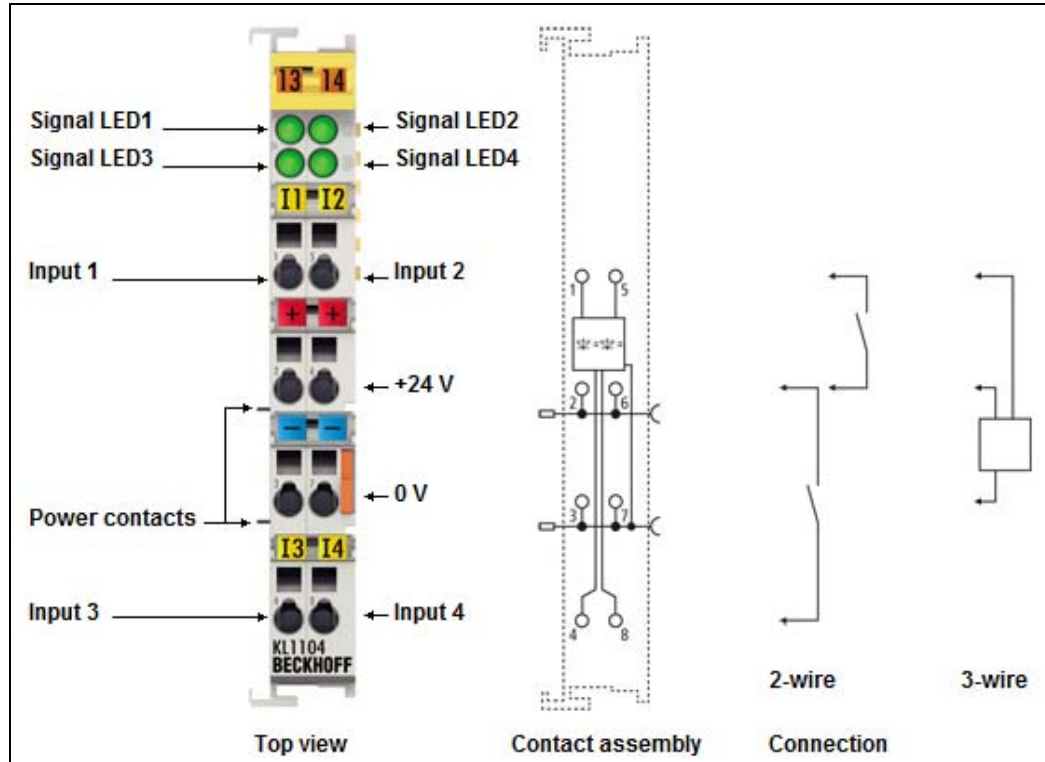
LC5100



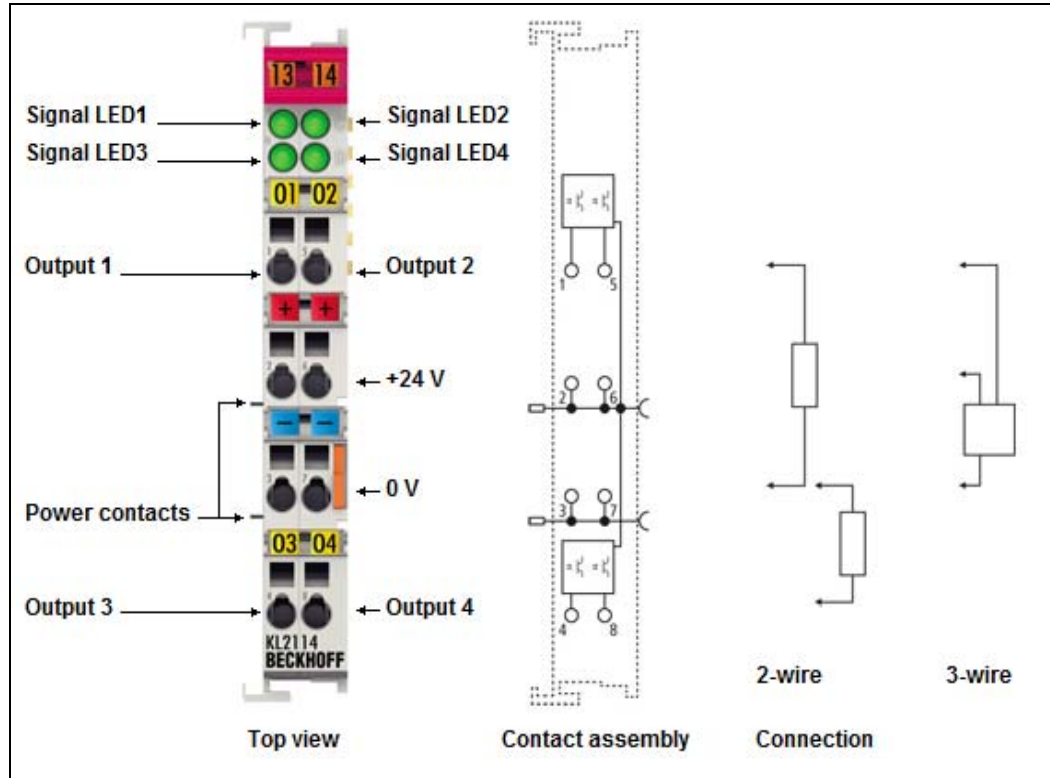
Osoitteen valitsin ja konfigurointi



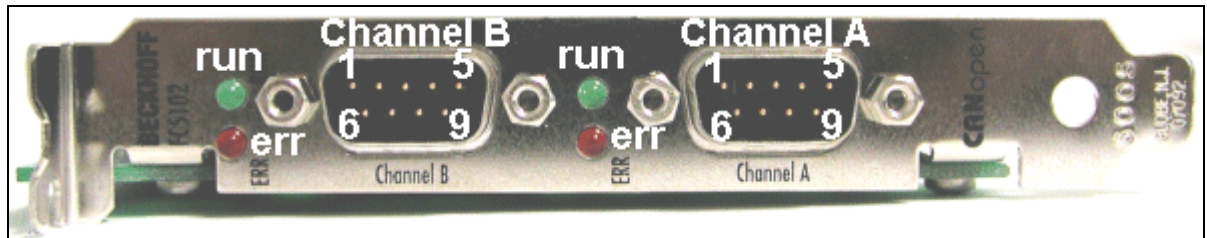
KL1104



KL2134/KL2114



## PC-kortti



## Pinnit

Pinni	Tehtävä	LC5100
2	CAN low (CAN-)	Right: CAN-L
3	CAN ground (sisäisesti liitetty pinniin 6)	GND
6	CAN ground (sisäisesti liitetty pinniin 3)	
5	Screen	Shield
7	CAN high (CAN+)	Left: CAN-H