

Metropolia Ammattikorkeakoulu  
Automaatiotekniikka

**Niko Tuomas Jämsén**

**Jacket-kunnonvalvontajärjestelmän modernisointi**

Insinööriyö 22.10.2008

Ohjaaja: tuotepäällikkö Tomi Nurma

Ohjaava opettaja: yliopettaja Jouni Jokelainen

Tekijä Otsikko	Niko Tuomas Jämsén Jacket-kunnonvalvontajärjestelmän modernisointi
Sivumäärä Aika	47 sivua 22.10.2008
Koulutusohjelma	automaatiotekniikka
Tutkinto	insinööri (AMK)
Ohjaaja Ohjaava opettaja	tuotepäällikkö Tomi Nurma yliopettaja Jouni Jokelainen
<p>Opinnäytetyössä suunniteltiin ja toteutettiin pyörivien prosessiuunien Jacket-valvontajärjestelmän mittaustietojen keruun siirto teollisuus-PC:ltä ohjelmoitavalle logiikalle. Tavoitteena oli muokata järjestelmä luotettavammaksi ja modulaarisemmaksi. Lisäksi järjestelmässä käytetty infrapunaskannerityyppi päivitettiin uudempaan versioon, jolloin sarjaliikennöinnin rinnalle saatiin myös Ethernet-liikennöinti. Uutta järjestelmää on mahdollista käyttää myös vanhojen skannereiden kanssa. Projektissa tutkittiin myös mahdollisuuksia lisätä järjestelmään uusia ominaisuuksia päivityksen yhteydessä.</p> <p>Mittaustietojen keruu päädyttiin toteuttamaan Beckhoffin sulautetulla SoftPLC-logiikalla sen monipuolisuuden, joustavuuden ja kilpailukykyisen hintansa ansiosta. Valitun järjestelmän rakenne vastaa myös projektin yhtä tavoitetta luoda järjestelmästä modulaarisempi.</p> <p>Työssä valittiin projektiin sopivat komponentit ja toteutettiin prototyyppi mittaustietojenkeruujärjestelmästä. Tärkeänä osana projektia oli Beckhoffin konfigurointi- ja ohjelmointiympäristöön TwinCATiin ja sen ominaisuuksiin tutustuminen. Tämä osa järjestelmää tullaan liittämään valvontajärjestelmän palvelimeen, joka on yhteydessä muihin automaatiojärjestelmiin.</p> <p>Valittu kokoonpano saatiin toimimaan halutulla tavalla ja liikennöinti logiikan ja skannerin välillä toteutettua sekä sarja- että Ethernet-liikenteellä skannerien vaatiman liikennöinti-protokollan mukaisesti. Myös raakaversio mittaustietoja keräävästä ohjelmakoodista saatiin toimimaan.</p> <p>Uusi järjestelmä on toistaiseksi tarkoitettu käytettäväksi ainoastaan yhden skannerin kanssa, eikä sitä ole vielä yhdistetty Jacket-palvelimeen. Lisäksi tarvitaan laajempia testauksia ennen kuin järjestelmää voidaan käyttää kaupallisesti. Tämä opinnäytetyö antaa kuitenkin projektille vahvan pohjan ja osoittaa, että mittaustietojen keruu on mahdollista ja kannattavaa toteuttaa ohjelmoitavalla logiikalla.</p>	
Hakusanat	TwinCAT, SoftPLC, Beckhoff, Jacket, pyörivä prosessiuuni

Author Title	Niko Tuomas Jämsén Modernization of the Jacket condition monitoring system
Number of Pages Date	47 22 October 2008
Degree Programme	Automation Technology
Degree	Bachelor of Engineering
Instructor Supervisor	Tomi Nurma, Product Manager Jouni Jokelainen, Principal Lecturer
<p>The main aim of this final year project was to design and execute an update to the Jacket kiln shell temperature monitoring system. In this update, the collection of measurement data is moved from the Jacket-server to PLC logic. The main reason for this update was to make the system more modular and more reliable. The scanner manufacturer has also launched a new version of the infrared scanner with an Ethernet interface; therefore the updated system was to use Ethernet communication with new scanners and serial communication with old scanners. Another aim of the project was to examine opportunities to add new features to the system.</p> <p>Measurement data were collected with Beckhoff's embedded SoftPLC. The main reasons for selecting Beckhoff's product were its modularity, versatility, flexibility and a competitive price.</p> <p>The components for the prototype were chosen, and the first version of the PLC code was developed. An important part of the project was to become acquainted with Beckhoff's configuring and programming software TwinCAT and its features. The SoftPLC developed will be connected to the Jacket server that is connected to the other automation systems.</p> <p>Tests showed that the components selected for the prototype and the raw version of the program code function as they were designed to function. The logic can communicate with the scanner via a serial or an Ethernet interface in accordance with the requisite communicating protocol.</p> <p>The number of scanners is so far limited to one, and this system has not been connected to the Jacket server yet. Further tests are also required before this system can be used commercially. However, the results of this project give a good basis for the adoption of PLC logic in the Jacket condition monitoring system and shows that it is possible and feasible to collect measurement data with a PLC system.</p>	
Keywords	TwinCAT, SoftPLC, Beckhoff, Jacket, rotary kiln

# Sisällys

Tiivistelmä

Abstract

Käytetyt merkinnät ja lyhenteet

1 Johdanto	8
2 Pyörivät prosessiuunit	9
2.1 Toimintaperiaate	9
2.2 Pyörivien prosessiuunien lämpötilan valvonnan merkitys	10
3 Jacket-valvontajärjestelmä	11
4 Vanhan järjestelmän laitteisto	12
4.1 Infrapunaviivaskannerit	12
4.2 Jacket-palvelin	13
4.3 Ohjelmisto	13
4.4 Jacketin versiot	15
5 Järjestelmän päivitys	16
5.1 Nykyisen järjestelmän kehitettävät kohteet	16
5.2 Tiedonkeruun siirtäminen ohjelmoitavalle logiikalle	16
5.3 Skannerin päivitys	17
5.4 Luiston mittaus	18
5.5 Vakiomäärä mittaustuloksia kierrosta kohden	18
6 Ohjausjärjestelmä	19
6.1 Ohjausjärjestelmän vaatimukset	19
6.2 Ohjausjärjestelmän valinta	19

7 Sulautettu järjestelmä CX9000	21
7.1 Valittu perusosa CX9000-00001	21
7.2 Modulaariset lisäosat	22
8 TwinCAT	23
8.1 Kommunikointi Jacketin käyttöliittymän kanssa	23
8.2 SoftPLC:n reaaliaikaisuus	23
9 IEC 61131-3 -standardi	25
9.1 IEC 61131 yleisesti	25
9.2 IEC 61131-3	25
9.3 Ohjelmayksiköt	26
9.4 Muuttujat	26
9.5 Standardin mukaiset kuvauskielet	26
10 Laitteiston käyttöönotto ja konfigurointi	28
10.1 Kotelointi ja kaapelointi	28
10.2 Sarjaliikenteen toteutus	29
10.3 Ethernet-liikenteen toteutus	30
10.4 System Manager	31
11 Liikennöinti SoftPLC:n ja skannereiden välillä	32
11.1 Sarjaliikenne	32
11.2 Ethernet-liikenne	33
12 TwinCAT PLC Control -projekti	35
12.1 Käytetyt kuvauskielet	35
12.2 Muuttujat	36
13 Skannerien liikennöintiprotokolla	37

14 Ohjelman toteutus	39
14.1 Skannereiden parametrit	39
14.2 Mittausvälin ja luistokertoimen laskenta	39
14.3 Mittaustietojen vastaanotto ja tallennus sarjaliikenteellä	40
14.4 Mittaustietojen vastaanotto ja tallennus Ethernet-liikenteellä	41
15 Päivitysprojektin tila ja yhteenveto	42
Lähteet	43
Liitteet	
Liite 1: Esimerkki ohjelmakoodista	45
Liite 2: Vuokaavio - Parametrien asetus skannerille	46
Liite 3: Vuokaavio - Mittaustietojen keruu skannerilta	47

## Käytetyt merkinnät ja lyhenteet

ADS	Automation Device Specification. Beckhoffin kommunikointimetodi SoftPLC:n ja ulkoisten PC-ohjelmien välille.
BCC	Block Character Checksum. Viestin tarkistussummatavu.
DCS	Distributed Control System. Hajautettu automaatiojärjestelmä.
DIN	Deutsches Institut für Normung. Saksan standardi-instituutti.
DLL	Dynamic Link Library. Jaettu kirjasto. Tieto saadaan usean eri ohjelman käyttöön.
EtherCAT	Ethernet for Control Automation Technology. Beckhoffin luoma Ethernet-kehyksen sisällä toimiva kenttäväylä.
FTP	File Transfer Protocol. TCP-protokollaa hyödyntävä tiedonsiirtoprotokolla kahden PC:n välille.
I/O	Input/Output. Tulo- ja lähtötiedot.
IEC	International Electrotechnical Commission. Kansainvälinen sähköalan standardointiorganisaatio.
NDT	Non-Destructive Testing. Ainetta rikkomaton testaus.
OLE	Open Link and Embedding. Työkalu, joka mahdollistaa tietojen yhteisen käytön useiden ohjelmien välillä.
OPC	OLE for Process Control. Open connectivity via open standards. Teollisuuden käyttämä, avoimen tiedonsiirron standardi.
PLC	Programmable Logic Controller. Ohjelmoitava logiikka.
POU	Program Organisation Unit. PLC-ohjelman rakennuskomponentit.
SoftPLC	Software based PLC. Ohjelmallinen, ohjelmoitava logiikka.
TCP/IP	Transmission Control Protocol / Internet Protocol. Useamman tietoliikenteessä käytettävän tietoverkkoprotokollan yhdistelmä.

## 1 Johdanto

Jacket on Sintrol Oy:n kehittämä kunnonvalvontajärjestelmä, jolla valvotaan pyörivien prosessiuunien pintalämpötilaa. Lämpötilan mittaus tapahtuu infrapunaskannereilla. Lisäksi uunin pyörimistä mitataan induktiivisen anturin avulla ja järjestelmää voidaan laajentaa liittämällä siihen infrapunapyrometrejä tai uunin laakereiden valvontajärjestelmä. Järjestelmällä voidaan valvoa yhtä tai useampaa prosessiuunia. Pyöriviä uuneja käytetään maailmanlaajuisesti esimerkiksi sellu- ja sementtiteollisuudessa. [3, s. 263]

Jacket-järjestelmän keskusyksikkönä toimii tällä hetkellä teollisuus-PC, jossa ajetaan Jacketin mittaus- ja valvontaohjelmistoa. Vuonna 2008 aloitetun päivityksen myötä järjestelmää ollaan ohjauslaitteiston osalta päivittämässä ja samalla muokkaamassa modulaarisemmaksi. Järjestelmän modularisoinnilla järjestelmän hallinta helpottuu ja siihen on yksinkertaisempaa lisätä uusia ominaisuuksia. Näin järjestelmää voidaan myydä asiakkaalle entistä paremmin räätälöitynä, jolloin asiakas saa järjestelmältä juuri ne ominaisuudet, joita tarvitsee. [3, s. 263; 4]

Jacketin päivitysprojektia johtaa Jacketin tuotepäällikkö Tomi Nurma, joka on ollut luomassa järjestelmää alusta lähtien. Tämän työn osuus projektista on päivittää tiedonkeruu infrapunaskannereilta ohjelmoitavalle logiikalle, joka toimii itsenäisesti ja lähettää mittaustiedot pyydettäessä keskusyksikölle. Lisäksi työssä perehdytään tämän muutoksen tuomiin etuihin ja uusiin mahdollisuuksiin järjestelmän parantamiseksi.

Jacket-järjestelmän kehittänyt Sintrol Oy on vuonna 1975 perustettu yritys, jonka liikevaihto vuonna 2007 oli noin 9,1 miljoonaa euroa ja henkilöstömäärä 26 [1; 2]. Sintrol on erikoistunut prosessiteollisuuden mittauksiin, automaatioon sekä NDT- ja laboratoriolaitteisiin. Yritys myy useiden valmistajien tuotteita kotimaan markkinoilla. Sintrolin tekninen osasto tarjoaa huolto- ja käyttöönottopalveluita sekä koulutuksia. Lisäksi Sintrol Oy valmistaa ja myy hiukkaspitoisuusmittareita ja Jacket-lämpötilanvalvontajärjestelmiä maailmanlaajuisesti. [2]

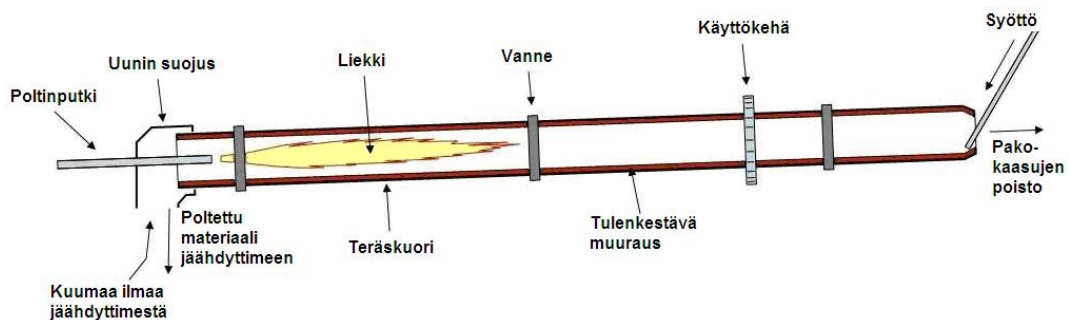


## 2 Pyörivät prosessiuunit

### 2.1 Toimintaperiaate

Pyöriviä prosessiuuneja käytetään maailmanlaajuisesti erityisesti sementti- ja selluteollisuudessa. Pyörivät prosessiuunit ovat teräsvaippaisia, yleensä hieman kaltevia ja noin 50-150 metriä pitkiä. Prosessiuunit pyörivät yleensä noin 30-250 kierrosta tunnissa. [4; 5; 6, s. 167]

Kuva 1 esittää pyörivän prosessiuunin toimintaperiaatetta. Materiaali syötetään uunin yläosasta, josta se valuu uunin kaltevuuden ja pyörimisen ansiosta kohti uunin alaosaa. Materiaalia lämmitetään kuumalla liekillä reaktiolämpötilaan. Materiaali liikkuu liekin ali saavuttaen maksimilämpötilansa uunin alapäässä juuri ennen siirtymistä jäähdyttimeen. Palamiseen vaadittava ilma syötetään jäähdyttimen kautta, jossa se jäähdyttää kuumaa materiaalia ja kuumenee samalla noin 400-800-asteiseksi. Kuuma ilma mahdollistaa välittömän ja nopean palamisen reagoidessaan polttoaineen kanssa vähentäen samalla uunin polttoaineen kulutusta. Lämpötila prosessiuunissa saattaa olla korkeimmillaan noin 1450 °C uunin sisällä ja noin 350 °C uunin ulkokuorella. [4; 5; 6, s. 167]



Kuva 1. Pyörivän prosessiuunin toimintaperiaate [5]

## 2.2 Pyörivien prosessiuunien lämpötilan valvonnan merkitys

Pyörivillä prosessiuuneilla on teollisuudessa hyvin korkea käyttöaste. Niitä käytetään normaalisti 24 tuntia vuorokaudessa, ja ne voivat tuottaa tänä aikana jopa 10 000 tonnia poltettua materiaalia. Pisimmillään prosessiuunia voidaan pyörittää jopa 18 kuukautta tauotta. [5]

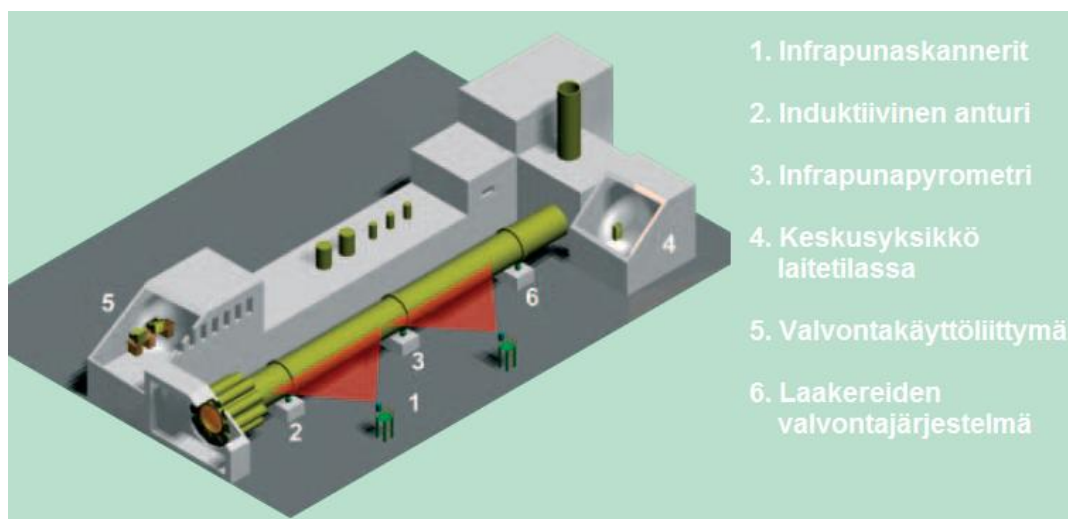
Korkean käyttöasteen vuoksi prosessiuunien kunnonvalvonta on erittäin tärkeää. Prosessissa voi muodostua ongelmatilanteita, kuten uunin reunoille kasaantuneita renkaita, uunin tukkeutumia tai kuumia pisteitä. Prosessiuuni saattaa myös vääntyä. Tällaisten ongelmien korjaaminen saattaa kestää päiviä tai jopa viikkoja, jolloin kustannukset voivat olla jopa miljoonia euroja. Myöskään inhimillisen virheen mahdollisuutta ei voida koskaan täysin sulkea pois. [3; 4]

Uunin lämpötilan valvonnalla yllä mainitut ongelmat voidaan ennaltaehkäistä ja näin saavuttaa suuria kustannussäästöjä pitkällä aikavälillä. Lisäksi lämpötilan valvonta mahdollistaa paremman prosessin hallinnan ja mittaustietojen tallennuksen myöhempää tarkastelua varten. Histogrammien avulla voidaan oppia tuntemaan prosessia paremmin ja välttämään vaaratilanteita tulevaisuudessa. [3; 4]

### 3 Jacket-valvontajärjestelmä

Jacket<sup>®</sup> on Sintrol Oy:n vuonna 1995 kehittämä pyörivien prosessiuunien lämpötilanvalvontajärjestelmä. Sintrol on myynyt Jacket-järjestelmiä useisiin maihin eri puolella maailmaa meesa- ja sementtiuunien valvontaan. Sintrol vastaa myös Jacket-järjestelmien valmistuksesta ja käyttöönotoista sekä järjestää Jacket Packet<sup>®</sup> -koulutuksia. Koulutuksen avulla tuotantolaitokset oppivat ottamaan prosessista kaiken hyödyn irti ja välttämään vaaratilanteita. [4]

Järjestelmän komponentit näkyvät kuvassa 2. Valvontajärjestelmä mittaa pyörivien uunien vaipan pintalämpötilaa infrapunaviivaskannereilla ja hälyttää, jos lämpötila nousee jossakin pisteessä liian korkeaksi. Järjestelmä myös helpottaa uunien ylläpitoa ja antaa lisäinformaatiota itse uuniprosessista. Järjestelmällä voidaan valvoa yhtä tai useampaa prosessiuunia, yhdellä tai useammalla infrapunaskannerilla. Lisäksi mittausaluetta voidaan täydentää lisäämällä hankaliin paikkoihin infrapunapyrometrejä. Jacketia voidaan laajentaa myös laakereiden kunnonvalvontajärjestelmällä. [3]

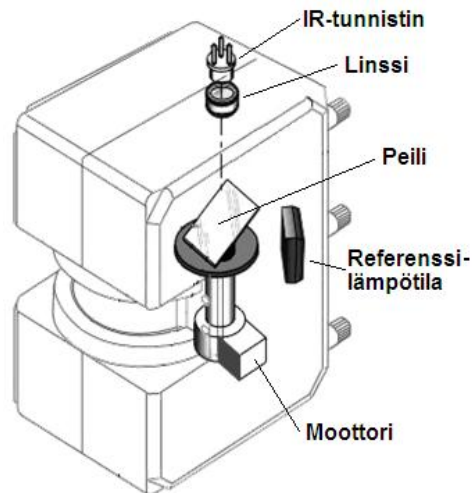


Kuva 2. Jacket-järjestelmän osat [3]

## 4 Vanhan järjestelmän laitteisto

### 4.1 Infrapunaviivaskannerit

Itse lämpötilanmittaus tapahtuu Raytekin MP50-infrapunaviivaskannereilla. Skannerin toimintaperiaate näkyy kuvassa 3. Skannerin peili pyörii 360° maksimissaan 48 kertaa sekunnissa. 90 asteen mittausalueelta kerätään 256 mittausarvoa, jotka muutetaan digitaalisiksi ja lähetetään eteenpäin kaksisuuntaisen RS-485-liitynnän avulla. Skanneri kalibroi itsensä joka kierroksella. [2; 3; 7]



Kuva 3. Skannerin toimintaperiaate [7]

Skannerit on asennettu haponkestävästä teräksestä valmistettuihin, paineistettuihin koteloihin (kuva 3), jotka suojaavat niitä pölyltä, kosteudelta ja muilta epäpuhtauksilta. Skannerin lisäksi kotelossa on teholähde, lämmitin ja ilmanhuoltolaitteet. Skannerin ikkunaan puhalletaan paineilmaa, jolloin laite pysyy puhtaana ja viileänä kuumissakin olosuhteissa. Lämmitin pitää kotelon lämpötilan riittävän korkealla myös kylmissä olosuhteissa. [3; 7]

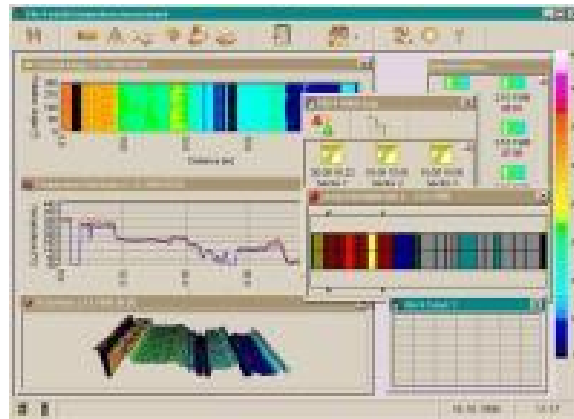
## 4.2 Jacket-palvelin

Mittaustiedot siirtyvät skannereilta Jacket-palvelimelle, joka on sijoitettu erilliseen teräslevykaappiin. Palvelimena toimii teollisuus-PC, joka kerää mittaustiedot ja luo niistä lämpötilaprofiilin Microsoft Windows NT/2000/XP -käyttöjärjestelmässä toimivan ohjelmiston avulla. Järjestelmän ohjelmisto pyörii käyttöjärjestelmän taustalla palveluna, jolloin sitä ei voida vahingossa kytkeä pois päältä. Mittaustulokset tallennetaan vähintään vuodeksi eteenpäin. Tähän palvelimeen voidaan liittyä Ethernet-verkon välityksellä ja laitekaappiin on mahdollista asentaa lisä-I/O-järjestelmä, jolla voidaan liittyä joko digitaalisesti tai analogisesti esimerkiksi DCS-järjestelmään. [3; 4]

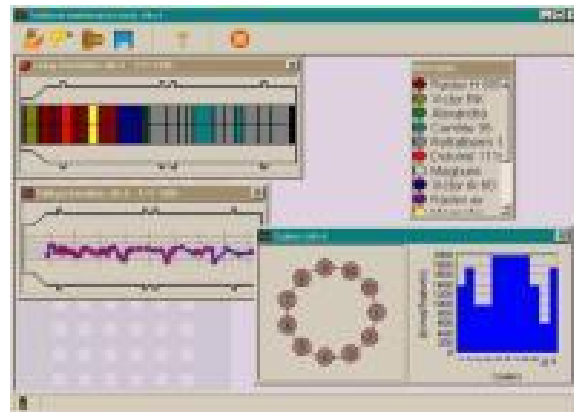
## 4.3 Ohjelmisto

Jacket-palvelin voidaan kytkeä mihin tahansa jo valvomossa olevaan työasemaan. Järjestelmän mukana toimitetaan uunin valvontakäyttöliittymä ja kunnossapitotyökalu. Ohjelmistojen käyttöliittymä muistuttaa perinteistä Windows-pohjaista käyttöliittymää. Millä tahansa työasemalla voidaan liittyä palvelimeen Ethernet-verkon välityksellä. [2; 3]

Kuvassa 4 näkyvästä valvontakäyttöliittymästä saadaan esiin yhden tai useamman uunin lämpötilakarttoja havainnollisina graafisina esityksinä. Ohjelmalla voidaan myös selata mittaushistoriaa vähintään vuosi taaksepäin. Ohjelmaan voidaan asettaa erilaisia hälytyksiä indikoimaan vaaratilanteita. Kuvassa 5 näkyvän kunnossapitotyökalun avulla voidaan hallita yhden tai useamman uunin muurauksia ja porauksia. Ohjelma sallii useamman käyttäjän yhtäaikaisen käytön. [2; 3]

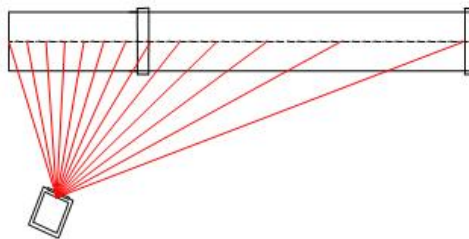


*Kuva 4. Valvontakäyttöliittymä [3]*



*Kuva 5. Kunnossapitotyökalu [3]*

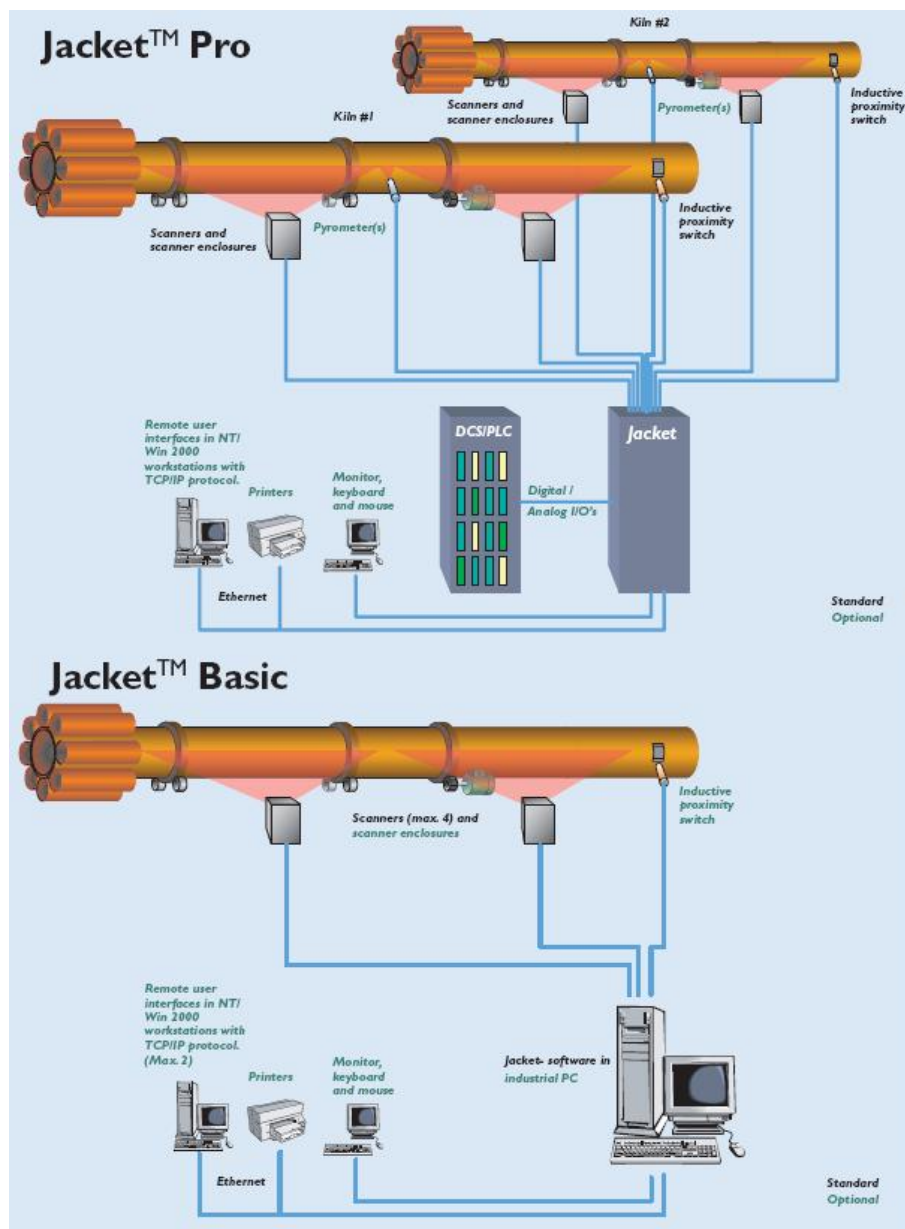
Ideaalista olisi, jos skannerit voitaisiin asentaa kohtisuoraan uuniin nähden. Näin voidaan harvoin kuitenkaan käytännössä tehdä, joten mittauspisteen leveys vaihtelee riippuen skannerin sijainnista ja asennosta uuniin nähden (ks. kuva 6). Leveyden vaihtelusta aiheutuva geometrinen vääristymä korjataan ohjelmallisesti Jacket-palvelimella järjestelmän käyttöönoton yhteydessä. [4]



*Kuva 6. Geometrinen virhe [7]*

#### 4.4 Jacketin versiot

Jacket-järjestelmästä on saatavana kahta erilaista versiota: Jacket<sup>®</sup> Pro ja Jacket<sup>®</sup> Basic. Näistä versioista Basic on tarkoitettu vain yhdelle uunille ja siinä on rajoitetummat ominaisuudet kuin Pro-versiossa. Versioiden erot sekä laitteiston kytkentäperiaate näkyvät kuvassa 7. [3]



Kuva 7. Jacket-versioiden erot ja kytkentäperiaate [3]

## **5 Järjestelmän päivitys**

Jacket-järjestelmä on pysynyt lähes muuttumattomana siitä lähtien, kun se on kehitetty vuonna 1995. Sen jälkeen järjestelmään on suunniteltu uusia ominaisuuksia ja parannuksia, jotka on tarkoitus toteuttaa päivityksen yhteydessä. Jacketissa käytettävä laitteisto on vanhentunut tekniikan kehityksen myötä. Lisäksi asiakkaiden toiveet ja parannusehdotukset on otettu huomioon päivitystä suunniteltaessa. Näiden tietojen pohjalta järjestelmää on nyt alettu päivittää uuteen versioon. [4]

### **5.1 Nykyisen järjestelmän kehitettävät kohteet**

Vanha järjestelmä on toteutettu keskitetysti teollisuus-PC:lle. PC:llä on kuitenkin ilmennyt ongelmia esimerkiksi tietoturvaohjelmistojen kanssa. Lisäksi joissakin tilanteissa PC:n on havaittu aiheuttavan viivettä mittauksiin. [4]

Jacket on ollut tähän asti yksi suuri kokonaisuus. Sen päivittäminen on työlästä, koska pienenkin muutoksen seurauksena koko ohjelmistoa on muutettava ja testattava. Päivityksen myötä järjestelmän modulaarisuus paranee, jolloin sen päivittäminen on jatkossa vähemmän työlästä ja asiakaskohtaisesti räätälöityjen järjestelmien tarjoaminen helpompaa. [4]

### **5.2 Tiedonkeruun siirtäminen ohjelmoitavalle logiikalle**

Tämän opinnäytetyön päätavoite on mittaustietojen keruun siirtäminen ohjelmoitavalle logiikalle. Kun logiikka kerää mittaustiedot itsenäisesti, vältytään Jacket-palvelimen ja infrapunaskannereiden väliseltä jatkuvalta liikennöinniltä, joka rasittaa palvelimen prosessoria turhaan. Tämä parantaa myös järjestelmän varmuutta, koska logiikka jatkaa mittausten keräämistä, vaikka yhteys palvelimeen hetkellisesti katkeaisikin. Unin viimeisimmän pyörähdysten mittaustiedot säilyvät logiikan muistissa ja logiikalta on mahdollista viedä hälytyksiä suoraan valvomoon palvelimesta riippumatta. [4]



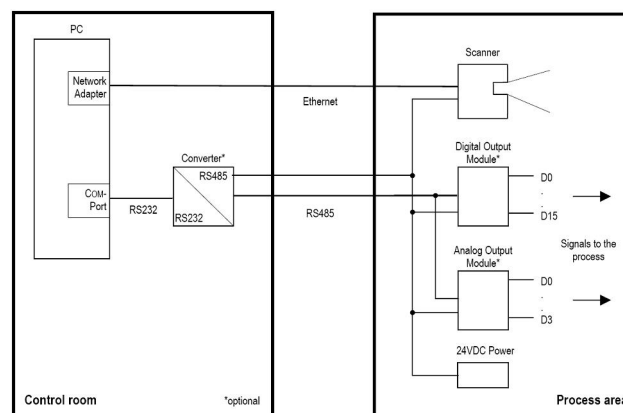
### 5.3 Skannerin päivitys

Vanhassa järjestelmässä käytetystä infrapunaskannerista MP50 on julkaistu uudempi malli MP150, joka on tarkoitettu ottaa järjestelmässä käyttöön päivityksen yhteydessä. Uudella skannerilla päästään parempaan mittaustarkkuuteen. Uuden skannerin tarkkuus on 1 024 mittauspistettä, kun vanhalla jouduttiin tyytymään 256 pisteeseen. [4; 8]

Uudessa skannerissa on myös Ethernet-liityntä, jolloin kalliita RS485-RS232-muuntimia ei enää tarvita. Käytännössä kuitenkin tässä sovelluksessa kaapelointietäisyydet ovat monesti niin pitkiä, että todennäköisesti joudutaan käyttämään esimerkiksi valokuitukaapelia pidemmillä matkoilla, jolloin tarvitaan taas erillisiä muuntimia. Käyttämällä skannerien läheisyydessä Ethernet-kytkintä päästään kenttäväylä-tyyliseen ratkaisuun. Tällöin säästetään kaapelointikuluissa, kun jokaiselle skannerille ei tarvitse viedä omia kaapelointeja Jacket-palvelimelta (vrt. kuva 7). [4; 8]

MP150-mallissa on asennusta helpottava sisäänrakennettu kohdistuslaser. Skannerin uudessa versiossa on myös parempi mittaustasaajuus. Se ei kuitenkaan tuo Jacket-järjestelmään käytännön etua, koska uunin kehä pyörii niin hitaasti, että vanhankin skannerin nopeus riittää tähän sovellukseen. [4; 8]

Päivitys toteutetaan siten, että myös vanhoja MP50-mallisia skannereita on mahdollista käyttää uudenkin järjestelmän kanssa. Tämä tarkoittaa käytännössä sitä, että uuteen järjestelmään on rakennettava kaksi rinnakkaista liikennöintimetodia (kuva 8). [4; 9, s. 3]



Kuva 8. Uuden skannerin liityntämahdollisuudet PC:hen [9, s.3]

#### **5.4 Luiston mittaus**

Prosessiuunin käyttökehä kiinnittyy uunin vaippaan ainoastaan käyttökehän hampaiden ja uunin vaipan väliin muodostuvan kitkan avulla. Tästä johtuen uuni luistaa hieman käyttökehään nähden. Asiakkaiden pyynnöstä Jacket-järjestelmään tullaan lisäämään luiston mittaus. Tämä tapahtuu mittaamalla induktiivisten kytkimien avulla käyttökehän ja uunin kierrosajat ja laskemalla logiikalla kerroin aikojen suhteesta. Tämän kertoimen avulla voidaan tarkkailla mm. uunin kuormitusta ja sen tasaisuutta. [4]

#### **5.5 Vakiomäärä mittaustuloksia kierrosta kohden**

Tällä hetkellä skannerit keräävät mittaustuloksia tietyin aikavälein riippumatta uunin pyörimisnopeudesta, jolloin mittaustulosten määrä uunin kierrosta kohden vaihtelee. Tämä aiheuttaa sen, että mittaustulosmatriisit ovat erikokoisia, jolloin niiden vertailu keskenään vaikeutuu. Lisäksi uunin pyöriessä hitaasti tällä menetelmällä saatetaan kerätä öturhaaö tietoa, koska mittauskohta ei ehdi olennaisesti muuttua. [4]

Päivityksen myötä logiikka mittaa uunin pyörähtämiseen kuluneen ajan ja laskee sen perusteella näytteenottovälin skannereille. Näin saadaan vakiomäärä mittaustuloksia pyörähdystä kohden riippumatta uunin nopeudesta. Vaikka pyörähdysaika onkin näin aina edelliseltä kierrokselta, se ei vaikuta oleellisesti näytteenottoväliin, koska uunin nopeudessa ei tapahdu nopeita muutoksia [4].

Laskennasta aiheutuvasta pyöristyksestä johtuen mittaustulosten määrä saattaa vaihdella yhdellä mittaustuloksella. Ongelma voidaan eliminoida käyttämällä mittaustulosten määrää, joka saadaan alaspäin pyöristämällä. Tällöin niillä kierroksilla, joilla mittaustuloksia tulee yksi enemmän, jätetään viimeinen mittaustulos huomioimatta.

## **6 Ohjausjärjestelmä**

### **6.1 Ohjausjärjestelmän vaatimukset**

Ohjausjärjestelmältä vaaditaan luotettavuutta, jotta järjestelmä pystyy toimimaan itsenäisesti, vaikka järjestelmän palvelin kaatuisikin. Myös ohjausjärjestelmän modulaarisuuteen ja joustavuuteen on syytä panostaa. Tällöin uusien ominaisuuksien lisääminen järjestelmään on helppoa. Lisäksi joustavuus on tärkeää, jotta järjestelmä toimii ongelmitta Jacket-palvelimen ohjelmistojen kanssa ja ohjausjärjestelmää on mahdollista hallita etänä muista järjestelmistä. Koska ohjausjärjestelmä tulee kaupalliseen sovellukseen, sen on oltava myös hinnaltaan kilpailukykyinen. Myös tietoturvaohjelmista aiheutuneista ongelmista ja PC:n aiheuttamista viiveistä haluttiin päästä eroon. [4]

### **6.2 Ohjausjärjestelmän valinta**

Mittaustietojen kerääminen päätettiin toteuttaa ohjelmoitavalla logiikalla, koska se kykenee toimimaan itsenäisesti toteuttamaan älyä vaativia toimintoja. Myös tietoturvaongelmat ja viiveet poistuvat, kun käytetään PLC-laitteistoa. Koska järjestelmän joustavuus oli tärkeä valintakriteeri, projektissa päädyttiin SoftPLC-järjestelmään. [10]

Mistä tahansa PC:stä voidaan tehdä SoftPLC asentamalla siihen automaatioväyläkortti ja SoftPLC-ohjelmisto. Tässä projektissa päätettiin kuitenkin käyttää sulautettua PC:tä sen paremman luotettavuuden, kompaktin kokonsa ja huokean hintansa ansiosta. Sulautetulla PC:llä tarkoitetaan tavallista PC:tä, joka on kehitetty jotakin tiettyä käyttökohdetta varten. Tässä tapauksessa sulautettu PC on tehty erityisen luotettavaksi ja kestäväksi teollisuuden vaatimia olosuhteita. Lisäksi sulautetut PC:t on teollisuudessa koteloitu niin, että ne on helppo sijoittaa esimerkiksi laitekaapin tai kotelon DIN-kiskoon. [10]

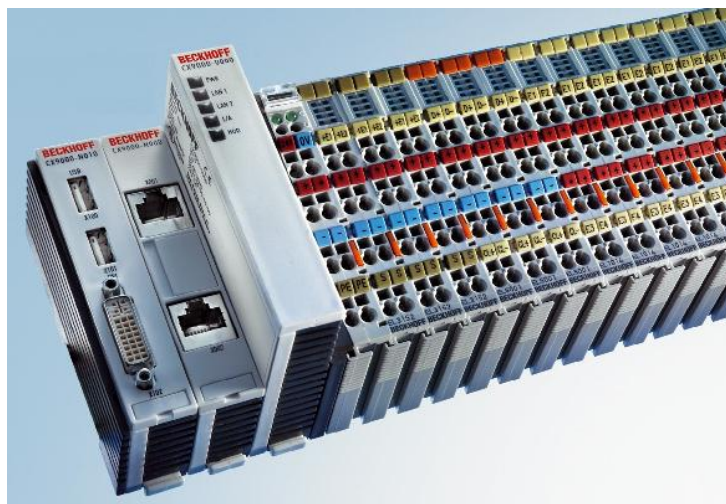
Järjestelmän alustaksi valittiin Beckhoffin valmistama sulautettu järjestelmä CX9000. Beckhoff on ehkä merkittävin SoftPLC-toimittaja, joten laitteistojen saatavuus pitäisi olla taattu myös tulevaisuudessa.

Beckhoffin TwinCAT-ohjelmisto tarjoaa valmiit kirjastot logiikan ja palvelimen käyttöliittymän väliseen kommunikointiin. Tällaista ominaisuutta ei löytynyt muilta valmistajilta. Valmiiden kirjastojen puuttuessa kommunikoinnin luominen Jacket-ohjelmiston kanssa olisi ollut selkeästi työläämpää, koska koko liikennöinti-protokolla olisi pitänyt rakentaa itse. Liikennöinti-protokollakirjastot skannerien ja logiikan välillä sekä sarjaliikenteenä että Ethernet-liikenteenä olivat projektin alkaessa saatavilla erikseen tilattaessa Beckhoffilta. Päivitysprojektin aikana nämä kirjastot tulivat myös ladattaviksi Beckhoffin internetsivuilta, eikä niitä tarvitse enää tilata erikseen. [4; 10]

Valintaan vaikuttivat monipuolisen TwinCAT-ohjelmiston lisäksi myös Beckhoffin kilpailukykyinen hinta ja joustavuus. Tulevaisuutta ajatellen myös Beckhoffin järjestelmien OPC-tuki voi olla hyödyllinen ominaisuus. [4; 10]

## 7 Sulautettu järjestelmä CX9000

CX9000 on Beckhoffin edullisin sulautettujen järjestelmien sarja, mutta ominaisuuksiltaan tähän sovellukseen täysin riittävä. Se on modulaarisesti DIN-kiskoon koottava järjestelmä. Kuten kuvasta 9 näkyy, CX9000 näyttää ulkoisesti PLC:n CPU-yksiköltä. [10] Modulaarisuuden etuna on joustavuus osia valittaessa. Kokonaisuuteen voidaan valita juuri ne osat, jotka tarvitaan. Näin järjestelmästä saadaan juuri halutunlainen ilman mitään ylimääräistä, jolloin hinta ja fyysinen koko pysyvät mahdollisimman pieninä. Modulaarista järjestelmää on helppo laajentaa myöhemmin, jos siihen on tarvetta.



*Kuva 9. Sulautettu PC CX9000 [10]*

### 7.1 Valittu perusosa CX9000-00001

Tähän projektiin valitun sulautetun järjestelmän tarkempi tyyppi on CX9000-0001. Valitussa mallissa on 16 Mt:n flash-muisti, 64 Mt:n keskusmuisti, E-väylä (EtherCAT) ja TwinCAT-PLC-Runtime -ohjelmisto. Tämän sulautetun järjestelmän perusosa käyttää Intelin IXP420 XScale® -teknologiaa ja 266 MHz:n kellotaajuutta. Käyttöjärjestelmänä toimii Microsoft Windows CE. Järjestelmä käyttää 24 voltin tasajännitettä, ja siinä on kaksi RJ-45-liitintä Ethernet-yhteyttä varten. [10]

## 7.2 Modulaariset lisäosat

Perusosan (CX9000-0001) lisäksi testausjärjestelmään valittiin seuraavat moduulit:

- digitaalinen tulomoduuli (EL1002)
- kolme relelähtömoduulia (3 x EL2602)
- analoginen lähtömoduuli (EL4024)
- sarjaporttiliityntämoduuli (EL6021)
- Ethernet-liityntämoduuli (EL6601)
- EtherCAT-väylän päättömoduuli (EL9011) [10].

Järjestelmään on mahdollista liittää K-väylällä toimivia sekä Beckhoffin omalla EtherCAT-väylällä toimivia moduuleita. E-väyläisiä moduuleja käytettäessä tietoliikenne kulkee samalla tiedonsiirtoprotokollalla myös moduulien välillä, toisin kuin perinteisillä kenttäväyläratkaisuilla. Lisäksi liikenne kulkee tavallisessa Ethernet-protokollassa Ethernet-kehysten sisällä. Koska tässä projektissa käytetään muutenkin Ethernet-liikennöintiä, oli EtherCAT luonnollinen valinta moduulien väliseksi väyläksi. EtherCAT on myös tällä hetkellä markkinoiden nopein kenttäväylä, mutta se ei ole tässä projektissa ratkaiseva ominaisuus, koska erityisen suuria tiedonsiirtonopeuksia ei tarvita. [11]

Koska osat ovat erillisiä moduuleja, niitä tullaan todennäköisesti tarjoamaan asiakkaille optioina. Tällöin asiakkaalle voidaan myydä juuri ne osat, jotka hän tarvitsee. Esimerkiksi asiakas, joka ostaa koko järjestelmän uutena, ei tule tarvitsemaan sarjaporttiliityntämoduulia, koska uudet skannerit käyttävät Ethernet-liikennöintiä. Vastaavasti vanhoille skannereille Ethernet-liikennöintiä ei tarvita.

Projektiin valitut osat ovat prototyypin testilaitteisto. Lopulliseen, päivitettyyn järjestelmään tulevat komponentit ja niiden määrät saattavat poiketa tästä kokoonpanosta.

## **8 TwinCAT**

TwinCAT on Beckhoffin sulautettu, IEC 61131-3 -pohjainen, Windows-ympäristössä (NT/2000/XP/Vista/XPE/CE) toimiva ohjausjärjestelmä. Ohjelmalla voidaan toteuttaa perinteinen PLC tai liikkeenohjaus ohjelmallisesti PC:llä. TwinCAT sisältää reaaliaikaisen ohjausjärjestelmän, ohjelmointiympäristön sekä diagnosointi- ja konfigurointityökalut. Koska järjestelmään valitussa mallissa CX9000 käytetään Windows CE:tä, siinä tulee mukana vain ohjausjärjestelmä. Ohjelmointi tapahtuu tällöin erilliseltä PC:ltä sarjakaapelilla tai lähiverkon välityksellä. [10] Tässä tapauksessa ohjelmointikoneena voidaan käyttää esimerkiksi Jacket-palvelinta tai huolto-PC:tä.

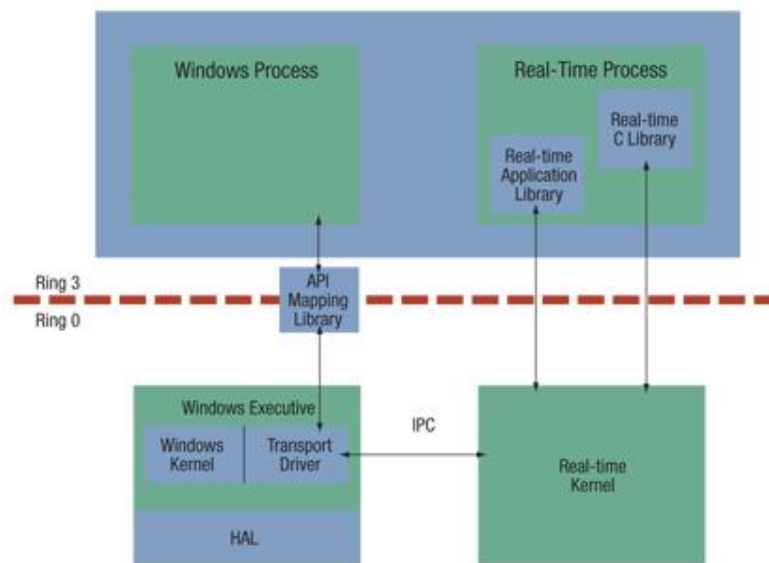
### **8.1 Kommunikointi Jacketin käyttöliittymän kanssa**

TwinCAT voi kommunikoida muiden sovellusten kanssa avoimien standardien avulla. Jacketissa tullaan käyttämään ohjelmiston mukana tulevaa valmista ADS.DLL-kirjastoa kommunikointiin SoftPLC:n ja Jacketin käyttöliittymän välillä. Myös OPC:n käyttö tulevaisuudessa on täysin mahdollista, jos Jacketista halutaan liittää muihin automaatiojärjestelmiin. [4; 10]

### **8.2 SoftPLC:n reaaliaikaisuus**

TwinCAT:n reaaliaikaohjaus on toteutettu siten, että vaikka käyttöjärjestelmä kaatuisi, TwinCAT:n suoritusjärjestelmä jää toimintakykyiseksi ja voi ajaa järjestelmän turvallisesti alas. Reaali-ohjauksen korkeasta prioriteetista johtuen sen pitäisi eliminoida vanhalla järjestelmällä muodostuneet viiveet. [10]

Todellisuudessa SoftPLC-järjestelmää ei voida pitää täysin reaaliaikaisena, koska Windows-käyttöjärjestelmä toimii priorisointimenetelmällä. Vaikka reaaliaikaiset toiminnot pyörivätkin erillisessä kernelin osassa korkealla prioriteetilla, on Windowsilla joidenkin toimintojen prioriteetti vielä korkeampi. Tämä tarkoittaa käytännössä sitä, että korkeamman prioriteetin omaavat toiminnot, joita on esimerkiksi hiiren liikutus, suoritetaan ennen SoftPLC:n prosessia. Kuva 10 selkeyttää reaaliaikaisen prosessin ja tavallisen Windows-prosessin eroa. [12]



Kuva 10. Reaaliaikaisten prosessien ajo omassa kernelin osassa [1]

Käytännössä tässä projektissa ei ole korkeamman prioriteetin toimintoja niin paljoa, että ne häiritsevät prosessia merkittävästi. Hyvin aikakriittisissä prosesseissa tämä saattaisi kuitenkin olla ongelma.



## **9 IEC 61131-3 -standardi**

### **9.1 IEC 61131 yleisesti**

IEC 61131-standardi on International Electrotechnical Commissionin kehittämä standardi PLC:iden laitteistoa ja ohjelmointia varten. Standardia on alettu kehitellä vuonna 1979, ja sen ensimmäinen versio on julkaistu 1990. Standardin osa 61131-3 käsittelee PLC-ohjelmointia. Sen ensimmäinen versio on julkaistu vuonna 1993, toinen vuonna 2001 ja kolmas versio on suunnitteilla. [13, s. 15; 14]

### **9.2 IEC 61131-3**

IEC 61131-3 ei ole käskykanta, jonka mukaan logiikkaohjelma tulee toteuttaa, vaan ennemminkin ohjekokoelma, jonka avulla ohjelma voidaan kuvata yksikäsitteisesti. Standardin tarkoituksena on ohjata jäsentelemään ohjelma selkeällä ja loogisella tavalla niin, että monimutkaisetkin asiat on esitetty mahdollisimman selkeästi. Toinen tärkeä tavoite on yhtenäistää eri yritysten ohjelmointimenetelmiä siten, että ohjelmat olisivat riippumattomia laitealustoista. [13, s. 12, 15; 14]

Standardin kolmanteen osaan (IEC 61131-3) on koottu yleisesti käytettyjä logiikkaohjelmoinnin kuvauskieliä. Kuvauskielistä kaksi on tekstipohjaisia ja kolme graafisia. Näistä kuvauskielistä voidaan ohjelmassa käyttää joustavasti yhtä tai useampaa. Lisäksi standardin kolmannessa osassa keskitytään PLC-ohjelman luomisen peruseräisiin. [13, s. 21, 95; 14]

### 9.3 Ohjelmayksiköt

PLC-ohjelma koostuu ohjelmayksiköistä, POU:sta (*Program Organisation Unit*), jotka vastaavat perinteisiä ohjelmoinnin lohkoja (*Blocks*). On olemassa kolme erilaista POU-tyyppiä:

- funktioita (*Function*, FUN)
- toimilohkoja (*Function block*, FB)
- ohjelmia (*Program*, PROG).

Funktion ja toimilohkon ero on, että funktio palauttaa samoilla parametreilla kutsuttuna aina saman arvon. Toimilohko taas sisältää ömuistiaö, joten se kykenee muistamaan esimerkiksi tilatietoja. Ohjelma pääsee käsiksi PLC:n I/O:hon ja mahdollistaa myös muiden POU:iden pääsyn sinne. [13, s. 30631]

### 9.4 Muuttujat

Logiikkaohjelmoinnissa käytetään muuttujia (*variables*) tiedon varastointiin ja käsittelyyn. Monissa matalan tason ohjelmointikielissä on ollut tapana määrittellä muistipaikan osoite fyysisesti (*flags*). Standardin mukaisessa ohjelmoinnissa tätä ei tarvitse tehdä manuaalisesti, vaan riittää, kun antaa muuttujalle nimen ja tyyppin. Ohjelmointijärjestelmä hoitaa muuttujan tallennuksen automaattisesti. [13, s. 22]

### 9.5 Standardin mukaiset kuvauskielet

Ensimmäinen standardin mukaisista ohjelmointikielistä on käskylista (IL, *Instruction List*). Käskylista on tekstipohjainen, tietokoneen assembler-ohjelmaa muistuttava, matalan tason ohjelmointikieli. Käskylistalla jokainen käsky sijoitetaan omalle rivilleen allekkain, ja käskyt toteutetaan järjestyksessä ylhäältä alas. Tästä järjestyksestä voidaan tarvittaessa poiketa hyppy-käskyillä. [13, s. 96; 14]

Rakenteellinen teksti (ST, *Structured Text*) muistuttaa syntaksiltaan Pascal-ohjelmointikieltä ja se on kehitetty ohjausjärjestelmiä varten. ST mahdollistaa myös valintojen käytön ohjelmassa (*if*-lauseet ja *case*-käskyt) ja iteraatiosilmukat (*for*, *while* ja *repeat*). [13, s. 1116112; 14] Rakenteellisella tekstillä kirjoitettu ohjelmakoodi on käskylistalla kirjoitettua koodia selvästi havainnollisempaa. Siksi sen käyttö onkin nykyisin yleisempää kuin käskylistan käyttö.

Toimilohkoja käytettäessä (FBD, *Function Block Diagram*) toiminnot kuvataan graafisesti symbolilaatikoilla. Näiden symbolilaatikoiden merkinnät on standardoitu. Toimilohkon vasemmalle puolelle tuodaan tulot, joiden perusteella toimilohkon oikealla puolella olevat lähdöt määräytyvät. Toimilohkoilla kuvattu ohjelma on selkeästi esitetty, ja sitä käytetään laajasti erityisesti Euroopassa. [14]

Tikapuukaavio (LD, *Ladder Diagram*) perustuu relekaavioihin. LD kuvaa ikään kuin releiden koskettimia. Tikapuukaavioilla voidaan kuvata pääsääntöisesti boolean-signaaleja (tosi ja epätosi). [13, s. 141; 14] Tikapuukaaviossa tuodaan kuvainnollisesti virta kaavion vasempaan laitaan, josta virta etenee oikealle releen yli, jos releen ehto täyttyy. Kaavion oikeassa päässä on lähtö, jonka tila riippuu siitä, saavuttaako övirtaö sen vai ei.

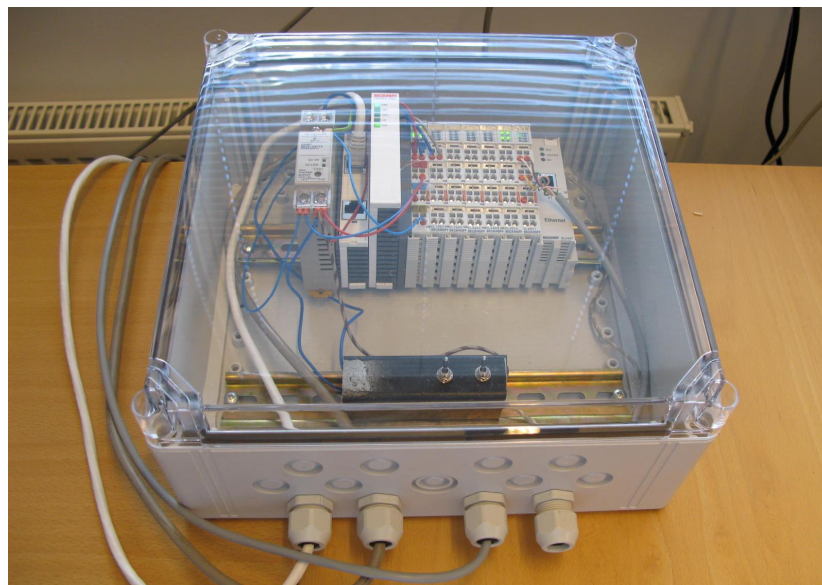
Sekvenssikaavio (SFC, *Sequential Function Chart*) koostuu askelista. Askeleiden välissä on ehtoja, joiden on toteuduttava ennen seuraavaan askeleeseen siirtymistä. Jokainen askel voi sisältää toimenpiteitä, jotka ovat voimassa niin kauan kuin askel on aktiivinen (kunnes seuraavat reunaehdot toteutuvat). Toimenpiteet ja ehdot voidaan kuvata millä tahansa standardin kuvauskielistä tai IEC-askelilla, joilla voidaan tehdä yksinkertaisia toimenpiteitä, kuten asettaa lippuja aktiivisiksi ja nollata niitä. Askeleet suoritetaan aina järjestyksessä ylhäältä alas. Rinnakkaiset ja vaihtoehdotiset sekvenssit ovat mahdollisia. [13, s. 164; 14] Sekvenssikaavio sopii hyvin sovelluksiin, jossa samat vaiheet toistuvat aina samassa järjestyksessä, kuten reseptipohjaisissa prosesseissa.

## 10 Laitteiston käyttöönotto ja konfigurointi

### 10.1 Kotelointi ja kaapelointi

Beckhoffin testilaitteisto ja Omronin 24 voltin virtalähde S8VS-03024 asennettiin Enston valmistamaan OPCM303013T-koteloon DIN-kiskoon kiinnitettynä. Kotelossa on valmiit läpiviennit kaapeloinneille. Virtalähteeseen kiinnitettiin pistotulpalla varustettu kaapeli, joka on mahdollista kytkeä tavalliseen pistorasiaan. Jännitelähteeltä vietiin 24 voltin jännite logiikalle ja EtherCAT-väylälle. Lisäksi väylän maadoitus otettiin jännitelähteeltä. Konfigurointia ja ohjelmointia varten logiikka kytkettiin Ethernet-verkkoon tavallisella Ethernet-kaapelilla.

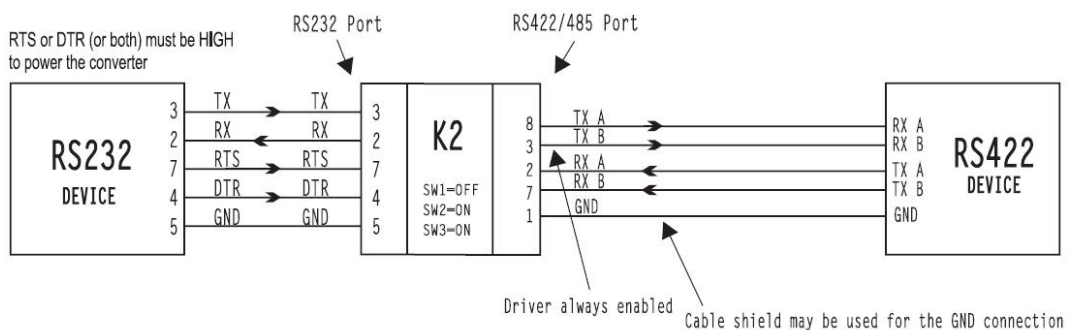
Testausta varten tulomoduuliin kytkettiin kaksi jousipalautteista kytkintä simuloimaan induktiivisia kytkimiä. Kytkimen yhteen liittimeen tuodaan 24 voltin jännite. Normaalisti kytkin on avonainen. Kun kytkimen vipua käännetään, kytkin sulkeutuu ja tulomoduulin vastaava paikka aktivoituu. Fyysisesti kytkimet asennettiin palaan onttoa, neliskulmaista tankoa niille porattuihin reikiin. Tanko asennettiin koteloon toiselle DIN-kiskolle. Kotelo asennusten jälkeen näkyy kuvassa 11.



*Kuva 11. Testilaitteisto koteloituna*

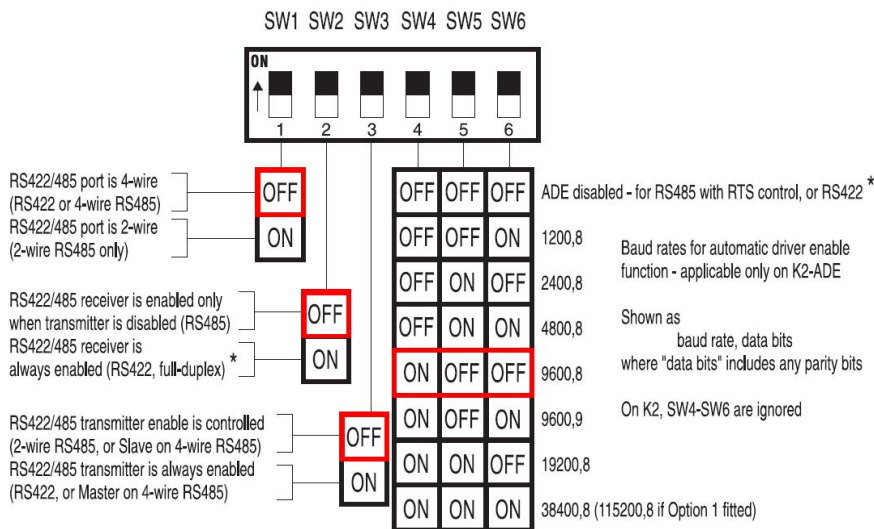
## 10.2 Sarjaliikenteen toteutus

Sarjaliikennettä simuloitiin ohjelmoinnin alkuvaiheessa kytkemällä sarjaliikennemoduuli PC:n sarjaporttiin ja lähettämällä ja vastaanottamalla viestejä EVS128-ScanReader-ohjelmalla. Koska projektissa käytetty moduuli on tarkoitettu RS422/RS485-liikenteelle ja PC:n sarjaportti RS232-liikenteelle, tarvittiin niiden välille vielä RS232-RS485-sovitin. Tässä työssä käytettiin KK Systemsin valmistamaan K2-ADE-sovitinta. Sovittimen kytkentä selviää kuvasta 12.



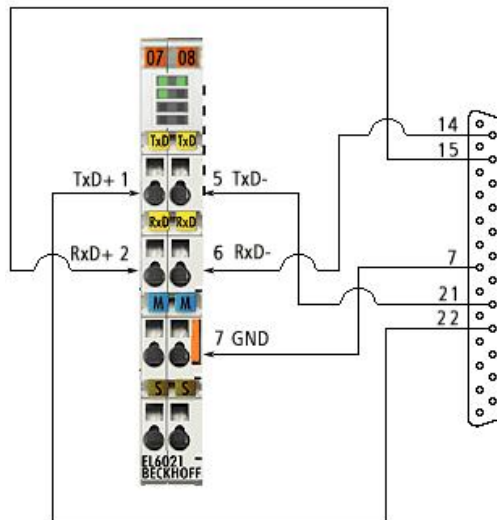
Kuva 12. K2-ADE-sovittimen kytkentä [15, s. 3]

Sovitin konfiguroitiin toimimaan samoilla asetuksilla kuin Jacketissa käytettävä skanneri ja PC:n sarjaportti. Nämä asetukset on merkitty kuvaan 13 punaisilla laatikoilla. Konfigurointi tapahtui muuntimen DIP-kytkimillä. [15]



Kuva 13. K2-ADE-sovittimen DIP-kytkinasetukset [15, s. 5]

Projektin myöhemmässä vaiheessa saatiin skanneri testikäyttöön. Skannerin mukana toimitetaan sarjaliikennekaapeli, jonka skannerin päässä on Raytekin oma teollisuusliitin ja toisessa päässä D25-naarasliitin. Naarasliittimen ja sarjaliikennemuodulin välille juotettiin kuvan 14 kytkennän mukainen kaapeli. Kaapelin toiseen päähän kytkettiin D25-urosliitin, joka kytkettiin Raytekin kaapelin D25-naarasliittimeen.



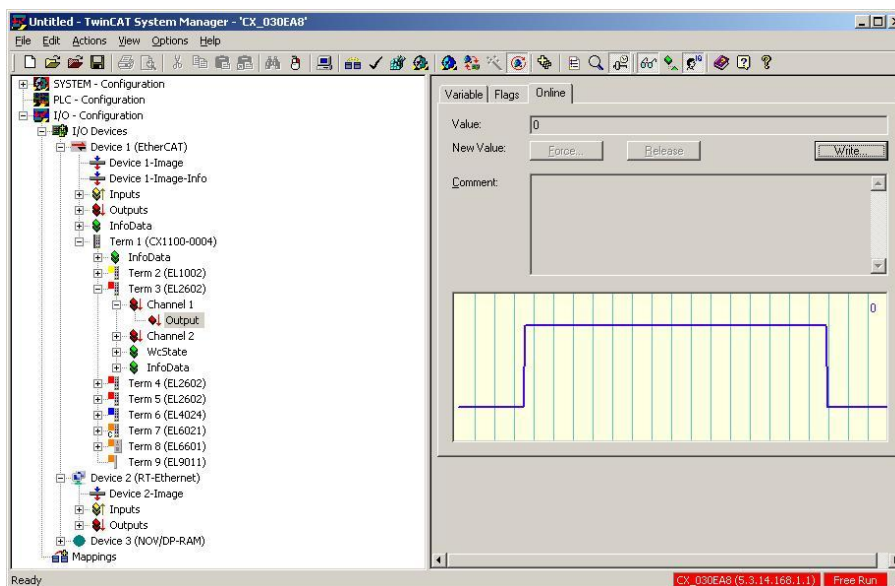
Kuva 14. Skannerin kytkentä sarjamoduuliin [10]

### 10.3 Ethernet-liikenteen toteutus

Jotta SoftPLC voidaan kytkeä useampaan skanneriin Ethernet-liikennettä käyttäen, tarvitaan SoftPLC:n ja skannereiden väliin Ethernet-kytkin. Yksi skanneri voitaisiin periaatteessa kytkeä ilman kytkintäkin, mutta silloin tarvittaisiin ristiinkytketty Ethernet-kaapeli. Sellaisen hankintaa ei katsottu järkeväksi, koska useampaa skanneria käytettäessä kytkin on joka tapauksessa välttämätön. Tässä projektissa päädyttiin käyttämään D-Linkin valmistamaa DES-1005D-kytkintä. Valintaan vaikuttivat D-Linkin edullinen hinta ja kompakti koko.

## 10.4 System Manager

Laitteiston konfigurointi tapahtui TwinCATin System Managerilla. System Managerilla etsittiin logiikoita lähiverkosta ja valittiin löytyneiden listalta oikea logiikka. Muut järjestelmän osat haettiin konfiguraatioon *Scan Sub Devices* -toiminnolla, jolloin ohjelmisto etsii logiikkaan liitettyjä laitteita. Laitteet on mahdollista hakea konfiguraatioon myös manuaalisesti, mutta se on automaattista hakua työläämpää. Lisäksi manuaalisesti haettaessa inhimillisen virheen mahdollisuus on olemassa. Kun konfigurointi on valmis, voidaan I/O:ta testata System Managerin *Free Run* -toiminnolla. Kuvassa 15 näkyy System Manager, joka on asetettu *Free Run* -tilaan. [10; 16]



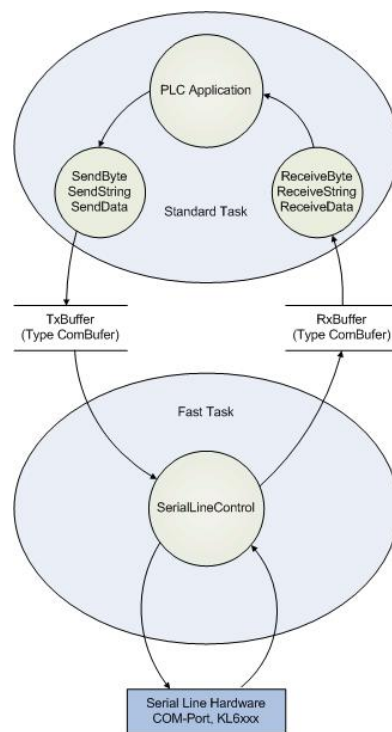
Kuva 15. System Manager Free Run -tilassa

TwinCATin ohjelmointiympäristössä, PLC Controllissa, ohjelmoija voi nimetä tulot, lähdöt ja muistipaikat haluamallaan tavalla ja jättää osoitteen määrittämisen ohjelmiston huoleksi. Tällöin osoitteeksi asetetaan esimerkiksi tuloille  $%I^*$  ja lähdöille  $%Q^*$ . Nämä tulot ja lähdöt linkitetään System Managerista I/O-moduulien fyysisiin tuloihin ja lähtöihin valitsemalla listalta haluttu moduulin paikka. [10; 16]

## 11 Liikennöinti SoftPLC:n ja skannereiden välillä

### 11.1 Sarjaliikenne

TwinCAT PLC Controliin on liitettävä erillinen kirjasto sarjaliikennöintiä varten. Kirjasto sisältää *SerialLineControl*-toimilohkon, joka hoitaa sarjaliikennöintiä. Yhteydenpitoon laitetason kanssa on luotava globaalit muuttujat (*ScannerInData* ja *ScannerOutData*), jotka on tarkoitettu käytettäväksi käytössä olevan sarjaliikennemuoduulin kanssa. Tässä tapauksessa nämä tyytit ovat *KL6inData22B* ja *KL6outData22B*. Tyyppi viittaa harhaanjohtavasti K-väylää käyttävään moduuliin, mutta on tarkoitettu käytettäväksi myös E-väylän laitteella (EL6021). *22B* tarkoittaa, että kerralla voidaan lähettää 22 tavua. Lisäksi tarvitaan muuttujat tiedon sisäiseen tallennukseen (*TxBuffer* ja *RxBuffer*). Näiden muuttujien tyyppi on *ComBuffer*. Kuva 16 selkeyttää tätä rakennetta. [10]



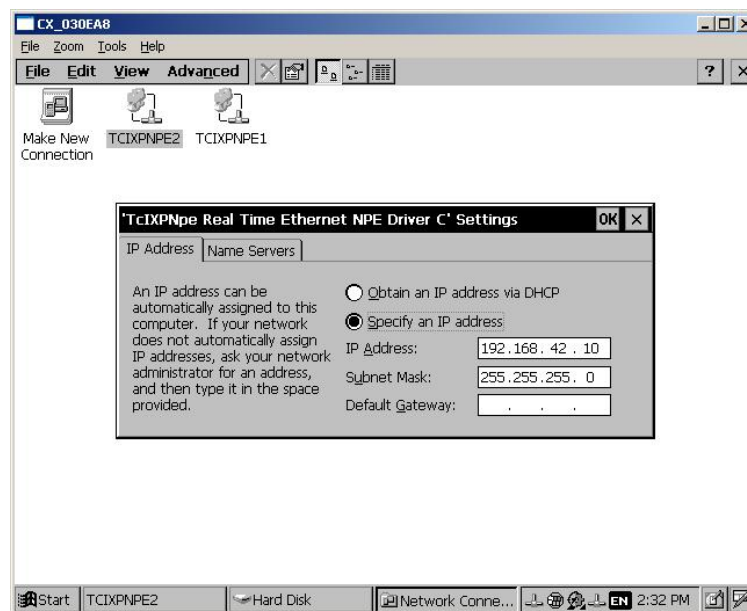
Kuva 16. Sarjaliikennöinnin periaatekuva [10]



Jotta sarjaportilla olisi mahdollista liikennöidä nopeudella 9600 baudia menettämättä dataa, on sarjaliikennöintiä hoitavan ohjelman kiertoajan oltava tarpeeksi nopea. Jos taas koko ohjelman kiertoajan asettaa hyvin nopeaksi, se kuormittaa SoftPLC:tä turhaan. Tämän vuoksi sarjaliikennöintiä varten on syytä luoda PLC Controliin erillinen tehtävä (*task*), jolla on lyhyempi kiertoaika kuin muut ohjelmat sisältävällä tehtävällä. [10]

## 11.2 Ethernet-liikenne

Ethernet-liikennöintiä varten TwinCATiin tarvitaan erillinen kirjasto, joka sisältää liikennöintiin tarvittavat toimilohkot. Lisäksi sulautetulle PC:lle pitää asentaa TCP/IP-yhteyspalvelin (*TCP/IP Connection Server*), joka tulee kirjaston mukana. Tässä käytettiin apuna CERHOST-etäkäyttöohjelmaa. CERHOST-ohjelmalla kirjaututtiin sulautetulle PC:lle ja kytkettiin FTP-tiedonsiirto-protokolla käyttöön, siirrettiin asennustiedosta logiikalle ja asennettiin se. Lisäksi Ethernet-kytkimen, johon skannerit liitetään, IP-osoite määriteltiin samaan aliverkkoon skannerin kanssa. Tämä tapahtui sulautetun PC:n lähiverkkoasetuksista (ks. kuva 17). Skannerit on syytä pitää eri aliverkossa kuin itse SoftPLC. [16]



Kuva 17. EL6601-moduulin IP-asetusten muuttaminen CERHOSTin avulla

TCP/IP-verkossa viesti saattaa pirstoutua siirron yhteydessä, jolloin viesti ei välttämättä tule sellaisenaan perille. Tämä täytyy huomioida mittaustietoja vastaanottaessa. Jos vastaanottokäske annetaan vain kerran yhden PLC-kierron aikana, saattaa osa datasta hävitä. Ongelmalta välttyään muodostamalla kahdesta ajastimesta (TON ja TOF) pulssigeneraattori, joka antaa 100 millisekunnin välein vastaanottokäskyn. Jokainen vastaanotettu viesti kirjoitetaan talteen matriisiin edellisten perään, ennen kuin uusi viesti vastaanotetaan puskurimatriisiin, jotta dataa ei katoa. Näin kaikki tieto saadaan talteen, vaikka viesti ei tulisikaan perille yhtenä pakettina. [10]

Haluttaessa kommunikoida skannerin kanssa on ensin avattava yhteys skannerille asetetun IP-osoitteen oikeaan porttiin. Oletuksena skannerin IP-osoitteeksi on asetettu 192.168.42.30 ja portiksi 2727. Yhteyden luonti tapahtuu toimilohkolla FB\_SocketConnect. Toimilohko luo yhteyden logiikan ja skannerin välille ja muodostaa tästä yhteydestä kahvan (*connection handle*). Tämän kahvan tunnisteen perusteella muut toimilohkot tietävät, mitä yhteyttä halutaan käyttää. Itse tiedonvaihto tapahtuu toimilohkoilla FB\_SocketSend ja FB\_SocketReceive. Lopuksi yhteys suljetaan toimilohkon FB\_SocketClose avulla. [10]

## 12 TwinCAT PLC Control -projekti

Yksi projektin tavoitteista oli tehdä ohjelmasta modulaarinen. Tämä onnistuikin TwinCAT-ohjelmistolla varsin hyvin. Yhteen TwinCAT-projektiin on helppo luoda useita ohjelmia. Näin ohjelmien koot saatiin pidettyä tarpeeksi pieninä, jotta ohjelmarakenne olisi selkeä.

### 12.1 Käytetyt kuvauskielet

Projekteissa on mahdollista käyttää yhtä tai useampaa kuvauskieltä. Eri kuvauskielet toimivat saumattomasti yhdessä, ja niitä yhdistämällä voidaan käyttää juuri kyseiseen tehtävään parhaiten soveltuvaa kieltä. Kuvauskielen valinta riippuu paljolti myös ohjelmoijan omista tottumuksista ja aiemmasta ohjelmointikokemuksesta. Harvoin onkaan olemassa yhtä ainoata kuvauskieltä tietyn asian toteuttamiseen, vaan todennäköisesti se on mahdollista toteuttaa myös jollain toisella kuvauskielellä.

Projektin alkaessa ohjelma oli tarkoitus toteuttaa sekvenssikaavioilla, joiden toiminnot ja ehdot on toteutettu toimilohkoilla. Tätä esitystapaa pidettiin luonnollisena valintana, koska useat toiminnot toistuivat aina samassa järjestyksessä. Ohjelmoinnin edetessä sekvenssikaavioista kuitenkin luovuttiin. Pääsyy tähän oli se, että toiminnot eivät pysy aktiivisina kuin sen aikaa, kun ohjelma on tietyssä askeleessa. Myös toimilohkojen käytöstä jouduttiin luopumaan, koska projektissa käsitellään mittaustuloksia ja komentoja merkkijonoina ja matriiseina, joiden käsittely toimilohkoilla on hankalaa.

Edellä mainituista syistä päädyttiin käyttämään rakenteellista tekstiä, jolla esimerkiksi merkkijonon tallentaminen muuttuukaan on yksinkertaista. Lisäksi monet toiminnot on TwinCATin sähköisen ohjeen, Information Systemsin, esimerkeissä toteutettu ainoastaan käskylistalla ja rakenteellisella tekstillä [10].

Jäsentelemällä rakenteellinen teksti loogisesti ja kirjoittamalla siihen riittävä määrä kommentteja ohjelmakoodista saatiin melko selkeästi luettavaa ja ymmärrettävää. Lisäksi kokonaisuudesta tulee selkeämpi, kun kaikki ohjelmat on kirjoitettu samalla ohjelmointikielellä. Rakenteellisen tekstin eduksi paljastui ohjelmaa testattaessa myös se, että sitä käytettäessä näytölle mahtuu selvästi enemmän informaatiota kerrallaan kuin muilla kuvauskielillä. Lyhyt esimerkki ST-kuvauskielestä on liitteenä 1.

## 12.2 Muuttujat

Projektissa päädyttiin käyttämään sekä paikallisia että globaaleja muuttujia. Jos kaikki muuttujat olisivat globaaleja, tulisi globaalien muuttujien listasta pitkä ja vaikealukuinen. Joitakin muuttujia tarvittiin kuitenkin useammassa ohjelmassa, joten niissä käytettiin globaaleja muuttujia. Toinen merkittävä syy tähän on, että kaikki prosessin kannalta oleelliset muuttujat halutaan viedä myös Jacket-palvelimelle, ja se onnistuu vaivattomammin globaaleja muuttujia käyttämällä.

Muuttujien nimeämisessä yritettiin käyttää mahdollisimman havainnollisia ja yhtenäisiä nimiä. Tämä on erittäin tärkeää erityisesti globaaleilla muuttujilla, jotta ne voidaan helposti erottaa toisistaan, vaikka muuttujalista on pitkä. Muuttujille kirjoitettiin lisäksi lisäinformaatiota antavat kommentit ja ne ryhmiteltiin selkeiksi kokonaisuuksiksi. Kaikki nimeämiset ja kommentoinnit tehtiin englanniksi, koska se on teknisissä projekteissa usein käytäntönä ja sitä ymmärretään maailmanlaajuisesti laajemmin kuin suomea. Muuttujien alkuun lisättiin lyhenne muuttujan tyyppistä. Esimerkiksi *GV\_DII* on globaali muuttuja (*Global Variable*) digitaaliselle tulolle (*Digital Input*) 1. TwinCAT ei salli välimerkkejä muuttujien ja ohjelmien nimissä, joten niiden tilalla käytettiin alaviivoja.

### 13 Skannerien liikennöinti-protokolla

Kommunikoidessa skannereiden kanssa on huomioitava skannereiden liikennöinti-protokolla. Liikennöinti-protokolla selviää Raytekin protokollamanuaalista. Tässä työssä käytettiin vanhan skannerin (MP50) protokollamanuaalia ja uuden version (MP150) protokollamanuaalin raakaversiota.

Lähetettäessä komentoa skannerille viestiin tulee ensimmäiseksi aloitustavu (SOH, 01H), sitten itse komento (esim. AR, 41H 52H), lopetustavu (EOT, 04H) ja BCC-tarkistussummatavu, joka on viestin muiden merkkien heksalukuarvojen summa (esim. komennolla AR BCC se on 98H). Komennon saatuaan skanneri vastaa joko ACK- (06H), NAK- (15H) tai ETB-komentomerkillä (17H). ACK tarkoittaa, että komento on ymmärretty ja hyväksytty. Jos vastaus on NAK, komennon syntaksi ei ole oikein tai tarkistussumma ei täsmää, jolloin komento ei aiheuta muutoksia skannerissa. Vastaus ETB puolestaan tarkoittaa skannerin sisäisen testin havainneen virheen, joka on korjattava, ennen kuin skannerille voidaan lähettää uusia käskyjä. [17, s. 365]

Mittausviesti skannerilta saadaan lähettämällä sille aloituskomento (STX, 02H). Tämä komento lähetetään skannerille sellaisenaan ilman aloitus-, lopetus- tai tarkistussummatavuja. Skanneri on mahdollista asettaa myös *Burst*-tilaan, jolloin se aloittaa aloituskomennolla mittauksen jatkuvan lähettämisen ja lopettaa vasta saadessaan lopetuskäskyn (ESC, 1BH). Tätä tilaa ei tässä projektissa kuitenkaan käytetä, vaan jokainen mittausviesti pyydetään skannerilta erikseen (*host*-tila). [17, s. 8, 11]

Vastaanotettaessa mittaus tietoja skannerilta mittaus tietoviesti alkaa *SYN*-merkillä, jota seuraa asetuksista riippuva määrä mittaus tavuja ja mahdollisesti skannerin sisäinen lämpötila ja skannerin lähtöjen senhetkiset tilat. Mittausviestissä ei lähetetä tarkistussummaa eikä viestin yhteydessä käytetä kuittausta (ACK, NAK tai ETB). [17, s. 8]

Mittaustietotavu saadaan muutetuksi lämpötilaksi kaavalla:

$$T(\text{pikseli}) = \frac{\text{Data}(\text{pikseli})}{255 * (T_{\max} - T_{\min})} + T_{\min}.$$

Kaavassa T on kyseisen pikselin lämpötila celsiusasteina, Data on mittaustiedon arvo (06255),  $T_{\max}$  on skannerille asetettu yläraja ja  $T_{\min}$  skannerille asetettu alaraja. [17, s. 869] Tämä muutos tapahtuu vasta Jacket-palvelimella lämpötilakarttaa luotaessa. Skannerin sisäistä lämpötilaa kuvaava tavu saadaan muutettua lämpötila-arvoksi muuttamalla heksaluku suoraan desimaaliluvuksi. [17, s. 9] Esimerkiksi 1AH tarkoittaa, että skannerin sisäinen lämpötila on 26 °C.

## 14 Ohjelman toteutus

Ohjelma koostuu pienistä ohjelmista ja funktioista. Taustalla pyörii myös nopeammalla kiertoaajalla toimiva sarjaliikennöinti-ohjelma. Parametrien asetusta skannereille ja mittaustietojen keräystä skannerilta kuvaavat vuokaaviot löytyvät tämän työn liitteistä 2 ja 3.

### 14.1 Skannereiden parametrit

Kierron alussa, skannerien käynnistyksen yhteydessä skannereille lähetetään tarvittavat parametrit omassa ohjelmassaan. Halutut parametrit syötetään tekstitiedostoon rivinvaihdolla erotettuina. SoftPLC-ohjelma avaa tekstitiedoston, lukee parametrit matriisiin tavu-muodossa ja sulkee tekstitiedoston. Tästä matriisista parametrit tallennetaan merkkijonomuodossa uuteen matriisiin siten, että havaittaessa rivinvaihto siirrytään seuraavaan matriisiin soluun. Näin merkkijonomatriisiin jokaiseen soluun tulee yksi komento.

Itse tiedonlähetytys tapahtuu funktion avulla, johon haluttu komento lähetetään parametrina. Funktio lisää komentoon alku- ja loppumerkit sekä laskee ja liittää siihen tarkistussummatavun. Tarkistussumman laskemista varten löytyy TwinCATista F\_CheckSum16-funktio. Valmis komento lähetetään skannerille ja tarkistetaan, että skanneri kuittaa ymmärtäneensä komennon.

### 14.2 Mittausvälin ja luistokertoimen laskenta

Taustalla pyörii kaksi ajastinohjelmaa, jotka mittaavat uunin kehän ja uunin kierrosaikoja. Näistä ajoista lasketaan seuraavan kierroksen mittausväli ja uunin luistokerroin. Mittausvälin ja luistokertoimen laskennassa käytettäviin jakolaskuihin on tehty tarkistus, että jakaja ei ole nolla. Ohjelma, jossa on toinen ajastin ja luistokertoimen laskenta, on liitteenä 1.

Jos jossakin ohjelmakierron vaiheessa jaettaisiin nollalla, SoftPLC jumiutuisi ja se pitäisi nollata, ennen kuin se olisi taas toimintakykyinen. Näin ei luonnollisesti saa missään tilanteessa päästä tapahtumaan ja laskutoimituksen ohitus on huomattavasti pienempi ongelma kuin logiikan jumiutuminen.

### **14.3 Mittaustietojen vastaanotto ja tallennus sarjaliikenteellä**

Kun laskettu mittaussväli on kulunut, lähetetään skannerille mittauskäsky (STX) SendByte-funktiolla. Mittaustiedot otetaan vastaan tavu kerrallaan ReceiveByte-funktion avulla. Tähän ratkaisuun päädyttiin, koska sarjaliikennöintipuskurit eivät olleet tarpeeksi suuria yli tuhannen tavun mittaustiedoille, jolloin vastaanotetusta viestistä luettavaksi jäi vain osa tiedoista.

Tavun vastaanotto täytyy toteuttaa silmukassa, koska ohjelma, jossa sarjaliikennekommunikointi ohjelman ja laitteiston välillä on toteutettu, toimii nopeammalla kiertojalla kuin muut ohjelmat. Silmukkaan mennään sisään, kun vastaanotto alkaa, ja silmukasta poistutaan vasta, kun tavu on vastaanotettu.

Uuden mittausviestin ensimmäinen tavu tallennetaan aina kaksiulotteisen mittausmatriisin vuorossa olevan rivin ensimmäiseen soluun. Kun tavu on vastaanotettu, tarkistetaan, onko vastaanotettu tavu SYN-merkki, joka ilmaisee mittausviestin alkua. Jos näin on, siirrytään matriisin seuraavaan tavuun, johon tallennetaan mittaustieto ja siirrytään seuraavaan soluun. Tätä jatketaan, kunnes saavutetaan palvelimelta asetettu pikseleiden määrä riviä kohti lisättynä kahdella (alkutavu ja sisäinen lämpötila). Jos rivin ensimmäinen tavu ei ole SYN-merkki, ohjelma ei siirry seuraavaan soluun vaan tallentaa seuraavan vastaanotetun tavun edellisen päälle, kunnes rivin ensimmäinen tavu on SYN-merkki. Näin jokaiselle riville tulee yksi mittausviesti.

Koska matriisin kokoa ei voida määritellä muuttujalla, on matriisi määritelty suurimman mahdollisen pikselimäärän mukaan. Pienempää pikselimäärää käytettäessä palvelimella jätetään vain ylimääräiset solut huomioimatta.



#### 14.4 Mittaustietojen vastaanotto ja tallennus Ethernet-liikenteellä

Ethernet-liikenteellä mittaustietojen vastaanotto on toteutettu *case*-rakenteella, joka alkaa kohdasta *case 0*. Kun ajastin-ohjelma antaa mittauskäskyn, muodostetaan yhteys (*case 1*) ja lähetetään STX-komentomerkki (*case 2*), jonka jälkeen siirrytään vastaanottokohtaan (*case 3*).

Tässä kohdassa on kahdesta ajastimesta muodostettu pulssigeneraattori, joka antaa vastaanottokäskyn 100 millisekunnin välein. Vastaanotettu data siirretään puskurimatriisista talletusmatriisiin. Seuraava viesti korvaa edellisen viestin puskurimatriisissa, mutta koska talletusmatriisin osoitinta ei nollata välillä, viesti tulee talletusmatriisissa edellisen perään. Tässäkin ehtona on, että rivin alkuun tulee SYN-merkki osoittamaan, että kyse on todella mittausviestistä.

Kun tavuja on vastaanotettu tällä menetelmällä mittausviestin verran, siirrytään seuraavaan kohtaan (*case 4*). Tässä kohdassa yhteys suljetaan, kirjoitetaan vastaanotettu mittausviesti talteen kaksiulotteisen mittausmatriisin seuraavalle vapaalle riville ja palataan ohjelmakierron alkuun odottamaan seuraavaa mittauskäskyä. Seuraavaan kohtaan siirtymisen ehtona kaikissa kohdissa on se, että edellinen kohta on suoritettu loppuun asti ilman virheitä.

## 15 Päivitysprojektin tila ja yhteenveto

Vaikka tämän työn tavoitteet täytyivät, ei koko päivitysprojekti ole vielä valmis. Palvelimelle tehtävät muutokset ovat vielä kesken, joten kaikkia logiikan toimintoja ei ollut vielä mahdollista toteuttaa (esim. virheilmoitusten jatkotoimenpiteet). Näitä öaukkojaö tullaan paikkailemaan sitä mukaan kuin palvelimen päivitys etenee.

Rajallisen testilaitteiston vuoksi useampaa skanneria ei voitu käyttää, joten ohjelmat on tällä hetkellä tehty ainoastaan yhdelle skannerille. Ohjelmia toteutettaessa on kuitenkin pidetty mielessä se tosiasia, että järjestelmään on oltava mahdollista kytkeä useampia skannereita. Siksi ohjelman rakenteesta on tehty sellainen, että skannereiden lisäyksen tulisi onnistua melko vähällä vaivalla ilman suurempia ongelmia.

Järjestelmälle testiympäristössä suoritettujen suppeat testitoimenpiteet eivät vielä anna täyttä varmuutta logiikkaohjelman toimivuudesta kaikissa mahdollisissa tilanteissa. Laajempia testejä tarvitaan vielä, ennen kuin päivitettyä versiota voidaan alkaa käyttää teollisuudessa vanhan järjestelmän tilalla.

Valittuun laitteistoon oltiin tyytyväisiä. Beckhoffin SoftPLC-järjestelmä vaikutti toimivan ajossa luotettavasti. Ohjelmoinnin yhteydessä SoftPLC osoittautui normaalia logiikkaa virheherkemmäksi. Ongelmia esiintyi kuitenkin ainoastaan silloin, kun järjestelmä joutui ei-toivottuun tilaan. Tällöin SoftPLC saattoi menettää toimintakykynsä täysin ja sen takaisin toimintakuntoon saaminen oli ajoittain turhan työlästä. Ohjemateriaalia TwinCAT-ohjelmistosta ja sillä ohjelmoinnista ei ole juurikaan tarjolla. Ohjelmoinnin ohjeistus on sähköisen, web-pohjaisen Information Systemsin varassa, josta kokonaisuuksien hahmottaminen on vaikeahkoa.

Kokonaisuutena tämä opinnäytetyö onnistui odotusten mukaisesti. Vaikka valmista tuotetta ei syntynyt, asetettuihin tavoitteisiin päästiin ja työ antaa hyvät lähtökohdat Jacket-päivitysprojektille. Työn edetessä myös TwinCAT-ympäristö on tullut tutummaksi ja TwinCAT-osaamista on mahdollista hyödyntää myös tulevilla vastaavanlaisissa projekteissa.

## Lähteet

- 1 Sintrol Oy - Tilinpäätös 30.9.2007. (Word-dokumentti.) Sintrol Oy. 17.12.2007. Luettu 28.5.2008.
- 2 Sintrol Oy. (WWW-dokumentti.) Sintrol Oy. <<http://www.sintrol.com>>. Luettu 26.5.2008.
- 3 Jackett . (pdf-dokumentti.) Sintrol Oy. <<http://www.sintrol.com/files/sintrol/productfiled/498file1Upload.pdf>>. Luettu 26.5.2008.
- 4 Nurma, Tomi. Tuotepäällikkö, Sintrol Oy, Lappeenranta. Palaveri 21.5.2008.
- 5 Rafiee, Ali. Rotary Kiln Modelling. (WWW-dokumentti.) <<http://rafiee77.googlepages.com/furnacemodelling>>. Luettu 27.5.2008.
- 6 Seppälä, Markku J., Klemetti, Ursula, Kortelainen, Veli-Antti, Lyytikäinen, Jorma, Siitonen, Heikki & Sironen, Raimo. Kemiallinen metsäteollisuus I, Paperimassan valmistus. Jyväskylä: Gummerts Kirjapaino Oy, 2001.
- 7 Jackett - Monitoring system for rotary kilns. (ppt-dokumentti.) Sintrol Oy. 11.10.2007. Luettu 29.9.2008.
- 8 MP150 Datasheet. (pdf-dokumentti.) Raytek A Fluke Company. <[http://support.fluke.com/raytek-sales/Download/Asset/9250378\\_ENG\\_A\\_W.PDF](http://support.fluke.com/raytek-sales/Download/Asset/9250378_ENG_A_W.PDF)>. 07/2008. Luettu 2.6.2008.
- 9 TF150/GS150/GS150LE Operator's Instructions. (pdf-dokumentti.) Raytek A Fluke Company. <[http://support.fluke.com/raytek-sales/Download/Asset/9250392\\_ENG\\_G\\_W.PDF](http://support.fluke.com/raytek-sales/Download/Asset/9250392_ENG_G_W.PDF)>. 07/2008. Luettu 25.8.2008.
- 10 Beckhoff Information System 04/2008. Beckhoff Automation Oy.
- 11 EtherCAT Technology Group. (WWW-dokumentti.) <<http://www.ethercat.org/>>. Luettu 11.7.2008
- 12 Christensen, Peter. Design Considerations for the Development of Real-Time Windows Systems. (WWW-dokumentti.) <<http://www.rtc magazine.com/home/article.php?id=100360>>. June/2005. Luettu 3.6.2008.
- 13 Karl-Heinz, John & Tiegkamp, Michael. IEC 61131-3: Programming Industrial Automation Systems. Berlin. Springer-Verlag. 2001.

- 14 Jokelainen, Jouni. Ohjaustekniikka. Luentomoniste. EVTEK-ammattikorkeakoulu, 2007.
- 15 K2-ADE RS232-RS485 Converter with ADE (pdf-dokumentti.) KK Systems LTD. <[http://www.kksystems.com/english/html\\_files/products.htm](http://www.kksystems.com/english/html_files/products.htm)>. 4.1.2002. Luettu 24.7.2008.
- 16 Piispanen, Jussi. Tekninen tuki, Beckhoff Automation Oy. TwinCAT-kurssi 12.6.2008.
- 17 Thermalert MP50Î Process Imager - Protocol Manual. (pdf-dokumentti) Raytek® Noncontact Temperature Measurement. Rev B 02/00. Luettu 18.8.2008.

```

PROGRAM PRG_Timer2
VAR
  V_Falling : F_TRIG;
  V_Rising  : R_TRIG;
  V_RS      : RS;
END_VAR

```

(\*When inductive switch is deactive, V\_Flip\_flop is set and the GV\_Timer2 is started. The falling edge of inductive switch resets the V\_Flipflop and GV\_Timer2 is reseted. TON-timer's output ET is elapsed time from last reset. Q comes TRUE, when the time that is set to PT is reached and Q changes to FALSE.\*)

```

V_Rising      ( CLK := GV_DI2 );
V_Falling     ( CLK := GV_DI2 );
V_RS          ( SET := NOT ( GV_DI2 ) , RESET1 := V_Falling.Q );
GV_Timer2     ( IN := V_RS.Q1 , PT := #3600s );

```

```

IF V_Rising.Q THEN
  (*The previous rotational time is stored to variable as integer*)
  GV_Rotational_time2 := TIME_TO_DINT( GV_Timer2.ET);
  (*Check that the time is not zero*)
  IF GV_Rotational_time1 <> 0 AND GV_Rotational_time2 <> 0 AND GV_Rotational_Time2 >= GV_Rotational_time1 THEN
    (*Slide is time1 per time2*)
    GV_SLIDE := DINT_TO_REAL(GV_Rotational_time2) / DINT_TO_REAL(GV_Rotational_time1);
  ELSE
    (*If can't be calculated, the slide is zero*)
    GV_SLIDE := 0 ;
  END_IF
END_IF

```

