

Janne Keskimölo

SIEMENS PCS 7 -PROJEKTIN OHJELMALOHKOJEN GENEROINTI IEA-TYÖKALULLA

SIEMENS PCS 7 -PROJEKTIN OHJELMALOHKOJEN GENEROINTI IEA-TYÖKALULLA

Janne Keskimölo
Opinnäytetyö
Syksy 2020
Automaatiotekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Sähkö- ja automaatiotekniikka, automaatiotekniikka

Tekijä: Janne Keskimölo

Opinnäytetyön nimi: Siemens PCS 7 -projektin ohjelmaloikojen generointi IEA-työkalulla

Opinnäytetyön nimi englanniksi:

Työn ohjaajat: Manne Tervaskanto (OAMK), Hannu Katajisto (PCS-Engineering Oy)

Työn valmistumislukukausi ja -vuosi: Syksy 2020

Sivumäärä: 52

Opinnäytetyössä käydään yksityiskohtaisesti läpi ohjelmaloikojen generointi Siemensin PCS 7 -projektiin. Työn alussa tutustutaan ensin yleisesti hajautettuihin automaatiojärjestelmiin teollisuudessa ja PCS 7:n sisältämään ohjelmistoon. Työn tavoitteena oli luoda selkeä työohje PCS 7:n Import/Export Assistant -työkaluun (IEA) ja samalla viedä työn alla olevan projektin sovellussuunnittelua eteenpäin.

Työhön lähdettiin PCS 7 -järjestelmän koulutuksen pikaoppimäärällä ja sen jälkeen syvennyttiin nopeasti IEA-työkalun käyttöön. Muutamien generointikokeilujen jälkeen siirryttiin tekemään ohjelmaa oikeaan projektiin. Tämä kattaa generoinnin vaatimien Import-tiedostojen luomisen ja täyttämisen projektin lähtötiedoilla sekä itse piirien tuomisen projektiin.

Lopputuloksena saatiin teoriaa hajautettujen automaatiojärjestelmien konstruktioista, osista ja sen käyttämisestä kommunikointitavoista. Siemensin PCS 7 -järjestelmän sisältämään ohjelmistoon perehdyttiin yleisellä tasolla, josta selviää sen ohjelman rakenne ja eri työkalujen käyttötarkoitukset. IEA-työkalun käyttämiseen luotiin yksityiskohtainen työohje, jota käyttämällä pystytään generoimaan piirejä PCS 7 -projektiin ja löytämään ratkaisuja mahdollisiin ongelmakohtiin.

Asiasanat: Siemens PCS 7, DCS, Import/Export Assistant, Process Tag Type, CFC

ABSTRACT

Oulu University of Applied Sciences
Electrical and Automation Engineering, Automation

Author: Janne Keskimölä

Title of thesis: Generating program charts for the Siemens PCS 7 project with IEA tool

Supervisors: Manne Tervaskanto (OAMK), Hannu Katajisto (PCS-Engineering Oy)

Term and year when the thesis was submitted: Fall 2020

Pages: 52

The objective of this thesis was to create detailed work instructions of Siemens PCS 7 Import/Export Assistant (IEA) tool and get superficially familiar with other parts of PCS 7 software. Instructions of the IEA tool was written during the automation project where software charts were generated for the program. In the first sections of the theoretical part the structure and parts of the distributed control systems are also described.

First, the PCS 7 system is briefly introduced and thereafter the aim quickly focuses on usage of the IEA tool after a few successful import test files. Instructions includes the creating, filling, and editing of IEA file as well as generating charts for the project.

The thesis consists of description of structure, parts, and communication methods of the distributed control system. The Siemens PCS 7 software was evaluated on a general level which revealed the structure of the software and the intended use of the different tools it offers. The usage of IEA tool is covered very precisely step by step and it can be used to generate charts for PCS 7 project. The guide also goes through a few problematic situations which can be encountered during use.

Keywords: Siemens PCS 7, DCS, Import/Export Assistant, Process Tag Type, CFC

ALKULAUSE

Haluan kiittää PCS-Engineering Oy:tä mahdollisuudesta suorittaa hyvin mielenkiintoinen opinnäytetyö asiakasprojektin lomassa. Siemensin PCS 7 tuli minulle uutena järjestelmänä, joten suuret kiitokset opinnäytetyöni valvojalle Hannu Kattajistolle opettamisesta ja kärsivällisyydestä.

Oulu 4.12.2020

Janne Keskimölo

SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
ALKULAUSE	5
SISÄLLYS	6
LYHENTEET	8
1 JOHDANTO	9
2 AUTOMAATIOJÄRJESTELMÄT TEOLLISUUDESSA	11
2.1 Toimintaperiaate	11
2.2 Käyttöliittymä	12
2.3 Hajautettu ohjausjärjestelmä (DCS)	13
2.3.1 DCS-järjestelmän laitteisto	14
2.3.2 Automaatiolaitteiston hierarkia	15
2.4 Kommunikointi automaatiassa	17
2.4.1 Verkkotopologia	17
2.4.2 Kenttäväyläprotokollat	19
2.4.3 OPC-rajapinta	20
3 SIEMENS PCS 7	21
3.1 Ohjelman osat	21
3.2 SIMATIC Manager	22
3.3 HW Config	23
3.4 WinCC Explorer	24
3.5 Graphics Designer	24
3.6 Import/Export Assistant (IEA)	25
3.7 Control Module Type (CMT)	26
3.8 Process Historian (PH)	28
3.9 Plant Device Management	28
3.10 SIMATIC BATCH	29
4 PIIRIEN GENEROINTI IEA-TYÖKALULLA	31
4.1 Process Tag Typen luominen	31
4.2 IEA-tiedoston muokkaaminen Excelissä	36
4.3 IEA File Editorin ja Excelin välinen soveltaminen	41

4.4 IEA-tiedoston tuominen projektiin	42
4.5 Generoitujen piirien päivittäminen	44
4.6 Ongelmatilanteita IEA-työkalun käytössä	46
4.6.1 Väärä listaerotusmerkki	46
4.6.2 Function Block -lohkojen vanhat versiot	48
5 YHTEENVETO	49
LÄHTEET	51

LYHENTEET

AS	Automation Station
CFC	Continuous Function Chart (PCS 7)
CPU	Central Processing Unit
DCS	Distributed Control System
ERP	Enterprise Resource Planning
ES	Engineering Station
FB	Function Block
HMI	Human Machine Interface
HW	Hardware
IEA	Import/Export Assistant (PCS 7)
I/O	Input/Output
MES	Manufacturing Execution System
OPC	Open Platform Communications
OS	Operator Station
PLC	Programmable Logic Controller
PTT	Process Tag Type (PCS 7)
RTU	Remote Terminal Unit
SFC	Sequential Function Chart (PCS 7)

1 JOHDANTO

Tässä opinnäytetyössä perehdytään hajautetun automaatiojärjestelmän rakenteeseen ja tutustutaan Siemensin PCS 7 -järjestelmän ohjelmointityökaluihin sekä tavoitteena on luoda yksityiskohtainen Import Export Assistant -työkalun työohje. Automaatiojärjestelmän sisältämistä laitteistokomponenteista kerrotaan aluksi yleisellä tasolla, jotta PCS 7:n eri ohjelmien käyttötarkoitukset voidaan selittää kohdennetummin. Työohjeessa käydään läpi IEA-työkalun käyttö vaiheittain samalla, kun automaatiojärjestelmän päivitysprojektin sovellussuunnittelua viedään eteenpäin generoimalla peruspiirit ohjelmaan.

Automaatiojärjestelmä vaatii toimiakseen ohjelman, joka on määritettynä käyttökohteensa mukaisesti. Kenttätasolla saattaa olla laitoksen koon mukaan satoja tai tuhansia piirejä, jotka mittaavat ja säätävät prosessisuureita. Jokainen piiri täytyy lisätä myös ohjelmaan, mikä olisi yksitellen hyvin työlästä ja aikaa vievää. IEA-työkalu on massaohjelmointiin suunniteltu PCS 7:n ohjelma, jolla samantyyppisiä piirejä voidaan generoida suuria määriä kerrallaan. Esimerkiksi projektin kaikkien säätöventtiilien positiot ja muut tarkemmat parametrit kerätään yhteen tiedostoon ja suoritetaan sen pohjalta säätöventtiilien ohjelmalohkojen generointi. Tällä säästetään huomattavasti aikaa, koska kaikkia ohjelman osia ei tarvitse luoda käsin yksitellen.

Työ suoritetaan PCS-Engineeringille, jolta on tilattu järjestelmäpäivitys biojalostustekniikkaa kehittäväille Oululaiselle Chempolis Oy:lle. Tämänhetkinen käytössä oleva ohjausjärjestelmä on ABB:n valmistama, joka päivitetään vastaavalla Siemensin laitteistolla ja ohjelmistolla. Nykyiset kenttälaitteet ja ABB:n I/O-moduulit tulevat jäämään paikalleen, joka täytyy ottaa huomioon ohjelmassa toimivan lopputuloksen kannalta.

PCS-Engineering Oy on insinööritoimisto, joka tarjoaa suunnittelu- ja konsultointipalveluita teollisuuden investointihankkeisiin. Yrityksen tarjoamiin palveluihin sisältyy muun muassa sähkö-, automaatio- ja instrumentointipuolen suunnittelu, asennusvalvonta, konsultointi ja projektinhoito. Sen vuosittainen liikevaihto on

keskimäärin noin viisi miljoonaa euroa. Yritys on perustettu 2004 Oulussa ja sillä on toinen toimipiste Jyväskylässä. (1.)

2 AUTOMAATIOJÄRJESTELMÄT TEOLLISUUDESSA

Erilaisia automaatiolla toimivia järjestelmiä on nykyään lähes kaikkialla teollisuudessa. Prosessit voivat olla mittakaavaltaan yksinkertaisia linjastoja tai osana suurempia tehtaita. Automaation tarkoituksena on vähentää tai poistaa kokonaan käsin tehdyn työn määrää ja jatkuvaa ihmisen valvontaa. Tällä päästään kohti tehokkaampaa ja vakaampaa tuotantoa sekä säästetään kustannuksissa.

Automaatiojärjestelmiä on useita eri tyyppisiä käyttötarkoituksesta ja järjestelmän laajuudesta riippuen. Yleisimmät teollisuudessa käytettävät automaatiojärjestelmätyypit ovat ohjelmoitavat logiikat (PLC), SCADA-järjestelmät ja hajautetut automaatiojärjestelmät (DCS). Logiikkaohjaimia käytetään pienemmissä automaatiototeutuksissa ja linjastoissa, joiden ohjaamiseen riittää yksi itsenäisesti toimiva ohjain. DCS- ja SCADA (Supervisory Control and Data Acquisition) -järjestelmät koostuvat useista PLC- ja RTU-ohjaimista, joilla prosessin ohjaus ja valvonta suoritetaan hajautetusti. Kaikki SCADA-järjestelmän ohjaimet ovat yhteydessä yhteiseen tiedonkeräys- ja käsittelyjärjestelmään ja ne voivat sijaita maantieteellisesti laajalla alueella. DCS-järjestelmä koostuu laitoksen tai tehtaan alueella paikallisesti sijaitsevista ohjaimista ja niitä hallitaan yhteisestä valvomosta. (2.)

2.1 Toimintaperiaate

Toiminnan perustana on prosessista, käyttäjältä tai vallitsevasta ympäristöstä saatava tieto, jonka perusteella suoritetaan määritellyjä toimenpiteitä. Mittalaitteet ovat kytkettynä tuloportteihin, joiden välityksellä informaatio tuodaan järjestelmään. Prosessin ohjaus toteutetaan toimilaitteilla, kuten venttiileillä, moottoreilla tai jollain muulla tavalla vaikuttamalla prosessisuureisiin järjestelmän lähtöjen kautta. Kaikki mitta- ja toimilaitteet ovat kytkettynä ohjausjärjestelmään, jonka kautta informaatiota käytetään ohjaukseen ja sitä voidaan tarvittaessa välittää ylemmäksi tuotannonohjausjärjestelmiin. Laitteiden välinen kommunikointi tapahtuu joko kaapeleita pitkin tai sitten langattomasti. (3, s. 10–12.)

Mittauksen ja säädön välillä on ohjelmoitava ohjausjärjestelmä, jolla analysoidaan ja käsitellään kerättyä tietoa. Lähes kaikki informaatio käsitellään nykyään

digitaalisesti, koska mikroprosessoreita on alettu käyttämään osana prosessiteollisuuden ohjausjärjestelmiä 1970-luvulta lähtien (4, s. 5). Ohjausjärjestelmien toiminta perustuu niille määritellyille ohjelmille, joita varten on kehitetty intuitiivisia ohjelmointikieliä. Tavallisimmat logiikkaohjelmointikielet ovat Ladder ja FB, jotka löytyvät lähes kaikkien laitevalmistajien ohjelmointityökaluista. Ladder eli tikapuu-logiikassa syntaksina toimii sähköpiirikaavioita muistuttavat vasemmalta oikealle kulkevat ”virtapiirit”, joita muokkaamalla ohjelman toiminnot määritetään. FB-kielellä sovelluksen funktio luodaan graafisia toimilohkoja yhdistämällä ja muokkaamalla. Käytettävissä ovat myös sanalliset ohjelmointikielet, joiden syntaksit vaihtelevan hieman eri valmistajien välillä.

Ohjelmassa tutkitaan mittalaitteilta saatavia tulosignaaleita, jotka ovat joko diskreettejä tai analogisia. Diskreetti signaali on binäärinen eli sen arvo on yksi tai nolla. Analoginen signaali on jollain tietyllä välillä vaihteleva reaalinen lukuarvo, joka saadaan virta- tai jänniteviestistä. Lähdöt voivat olla samalla tavalla diskreettejä tai analogisia. Yleinen teollisuudessa standardina oleva analoginen virtaviesti on välillä 4–20 mA. Vaikka väli onkin rajattu, se pystytään skaalaamaan ohjelmassa vastaamaan oikeita arvoja. Yksinkertaistettuna analogiana ohjelmassa voidaan tutkia esimerkiksi asioita A ja B ja suorittaa niiden perusteella toimenpide C (5, s. 14).

2.2 Käyttöliittymä

Automaatiojärjestelmän ja ihmisen välisenä rajapintana toimii käyttöliittymä eli HMI, jonka avulla prosessia hallitaan ja valvotaan. Käyttöliittymän operointikuvat sisältävät usein graafisen yleiskuvan prosessin jostain osa-alueesta ja siihen kuuluvista kentälaitteista. Sen kautta saadaan reaaliaikaista informaatiota ja käytetään mahdollisesti useampia eri prosessin osia samasta paikasta. Päätelaitteena toimivat valvomossa olevat tietokoneet tai kentällä sijaitsevat kosketusnäyttöpaneelit (Kuva 1).

Valvomosovellukset ovat olennaisia työkaluja prosessin ajamisessa, kunnossapidossa ja vikojen diagnosoinnissa. Digitaalisille näytöille on helppoa ohjelmoida lisää painikkeita, mittareita ja taulukoita eikä kentälle tarvitse alkaa asentamaan uusia fyysisiä painikkeita. Käyttöliittymään tulee hälytyksiä, jotka voivat vaatia

käyttäjiltä toimenpiteitä ja kuittauksen prosessin jatkamiseksi. Sen kautta voidaan seurata trendejä ja historiatietoja, jotka ovat apuna kunnossapidon diagnostiikassa ja tuotannon tehokkuuden parantamisessa.

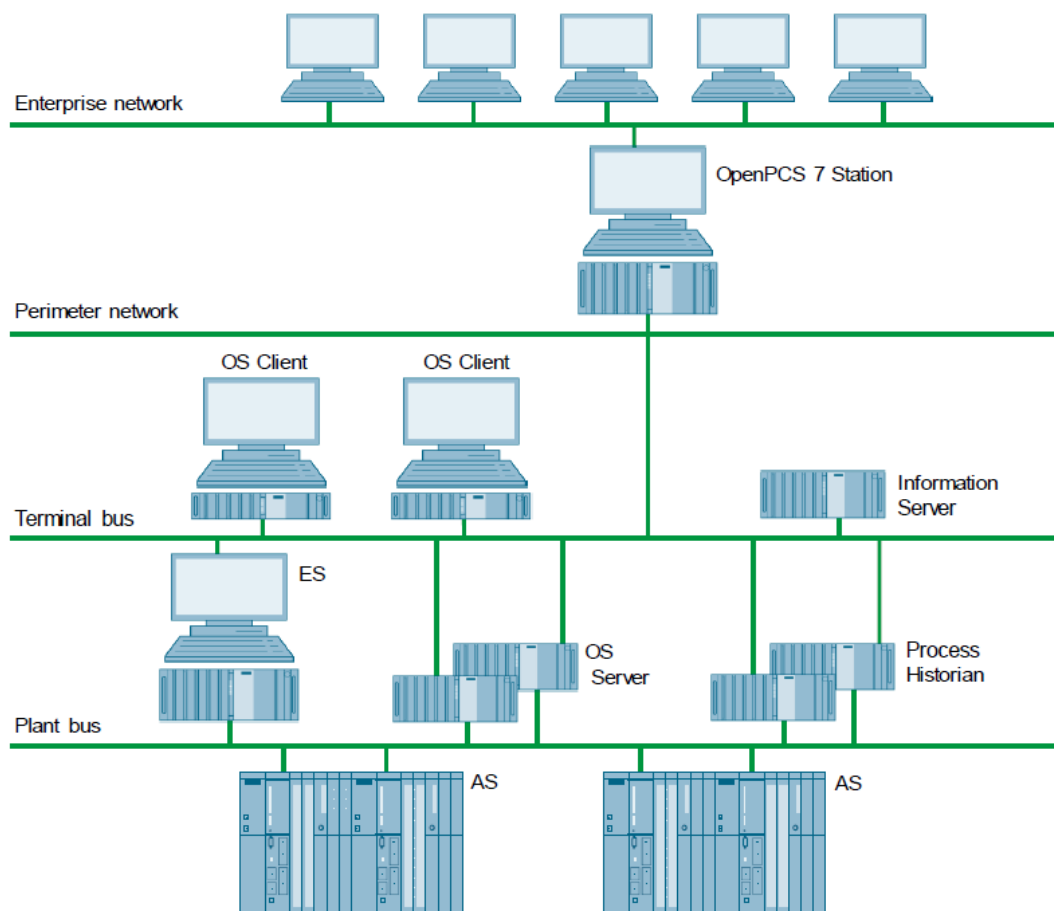


KUVA 1. Siemensin SIMATIC HMI kosketusnäyttöpaneeli (6).

2.3 Hajautettu ohjausjärjestelmä (DCS)

Isoissa laitoksissa I/O-määrät ovat hyvin suuria ja prosessin osa-alueet voivat olla hajautettuna laajalle alueelle. Niiden vaatimat lukuisat ohjelmalohkot vaativat paljon laskentatehoa, joten redundanttisen toiminnan kannalta kaikkia automaatiojärjestelmän osia ei kannata jättää yhden aseman varaan. Järjestelmän toimintoja ja kuormaa hajautetaan siis useammalle erilaiselle asemalle ja samalla lähemmäksi fyysisiä kenttälaitteita.

Hajautettu ohjausjärjestelmä koostuu autonomisista prosessiasemista, jotka ovat yhteydessä toisiinsa nopealla tiedonsiirtoyhteydellä (Kuva 2). Prosessiasemat ja hajautetut etä-IO-moduulit pyritään sijoittamaan lähelle kenttälaitteita, jotta kaapeloinnit pysyvät suhteellisen lyhyinä. Hajautetulle ohjausjärjestelmälle on ominaista, että prosessilaitteille on yksi yhdistävä hallintatyökalu, jonka avulla järjestelmää konfiguroidaan, valvotaan ja ohjataan (7, s. 4).



KUVA 2. DCS-järjestelmän automaatioasemien sijainti (8).

2.3.1 DCS-järjestelmän laitteisto

Hajautetun ohjausjärjestelmän älynä toimivat automaatioasemat (AS), joita kutsutaan myös prosessiasemiksi. Niillä kerätään informaatiota prosessista ja suoritetaan laskutoimintoja ja ohjauksia. Prosessoreissa on kattavat liitännät I/O-moduuleita ja kommunikointia varten.

Valvomoasemat (OS) ovat Windows-pohjaisia tietokoneita valvomosovelluksella ja niihin on tavallisesti liitetty useita näyttöjä käytön helpottamiseksi. Valvomonäyttöillä näkyvät piirikohtaiset hälytykset, mittausarvot, trendikaaviot sekä suuremmissa mittakaavassa prosessikaaviot ja sekvenssien tilanne ja ohjaus. DCS-järjestelmän valvomosta pystytään hallitsemaan koko tehdasalueen kaikkia prosessialueita, koska kaikki automaatioasemat ovat integroituna samaan järjestelmään.

Ohjelmointitietokoneilla (ES) suoritetaan päivityksiä laitekonfiguraatioon, automaatio-ohjelman toimintaan ja valvomonäyttöihin. Ne ovat tavallisesti sijoitettuna valvomosta erilleen automaatio- tai sähkötiloihin.

AS:n ja OS:n välisenä rajapintana toimivat serveriasemat, joita on usein redundanttisen toiminnan kannalta kaksi kappaletta. Ne vastaavat informaation keräämisestä, säilyttämisestä ja jakamisesta.

Kenttälaitteet ovat yhteydessä prosessiasemiin I/O-moduulien kautta. Ne voivat olla kytkettynä suoraan modulaariseen prosessiasemaan tai sitten hajautettuihin etä-I/O-yksiköihin. Oikean tyyppiset I/O-moduulit valitaan prosessin kenttälaitteiden mukaisesti. Suurin osa I/O:sta koostuu digitaalisista ja analogisista tuloista ja lähdöistä, mutta erilaisia moduuleita tarvitaan esimerkiksi nopeiden pulssien mittausta tai turvatoimintoja varten.

2.3.2 Automaatiolaitteiston hierarkia

DCS-järjestelmä perustuu client-server-arkkitehtuuriin, jossa dataa säilytetään serverillä ja luetaan client-laitteella. Ilman serveriä automaatiojärjestelmä kykenee toimimaan itsenäisesti, mutta valvomoon ei saada tuotua lainkaan informaatiota eikä historiatietoja saada tallennettua. Client-asetat toimivat puolestaan päätelaitteina valvomosovelluksille, joita käytetään datan lukemiseen ja kirjoittamiseen.

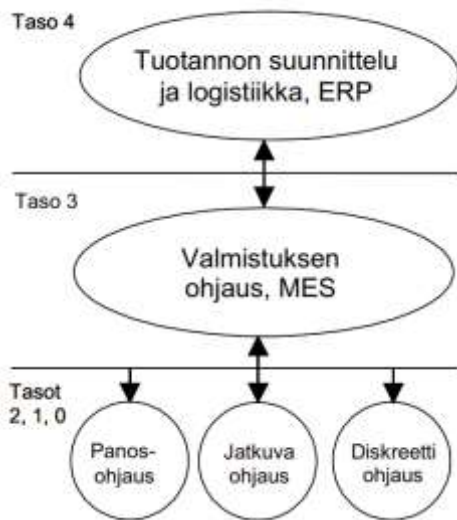
Hajautetun ohjausjärjestelmän laitteet voidaan jakaa karkeasti kolmelle eri tasolle niiden toiminnan ja automaatiolaitteiston hierarkkisen sijainnin perusteella. Olenaisia termejä tässä ovat horisontaalinen ja vertikaalinen integraatio. Horisontaalisessa integraatiossa informaatio on kaikkien saman hierarkkisen tason laitteiden käytössä. Vertikaalisessa integraatiossa informaatio välittyy tasolta toiselle, kuten esimerkiksi kenttälaitteelta säätimelle. (7, s. 2–3.)

Alimmalla eli hierarkkisen rakenteen perustasolla ovat kenttälaitteet, joilla vaikutetaan suoraan prosessiin. Ne eivät ole yhteydessä toisiinsa suoraan, mutta horisontaalinen integraatio tällä hierarkiatasolla on toteutettavissa välittämällä tietoa I/O:n kautta järjestelmästä toiseen, mikä on kuitenkin melko harvinaista.

Toisella tasolla kenttälaitteiden yläpuolella ovat prosessiasemat, ohjaimet ja I/O-moduulit. Prosessiasemat ovat yhteydessä toisiinsa, jolloin koko tehtaan alueen kenttälaitteet ovat ohjattavissa yhteisen valvomon kautta.

Ylemmällä tasolla tapahtuu prosessin hallinta, valvonta ja kehittäminen. Tähän kuuluvat valvomotietokoneet (OS), joissa käyttäjät ajavat prosessia näytöiltä ja toiminnanohjausjärjestelmästä saatavan informaation perusteella. Valvomotietokoneiden rinnalla on myös ohjelmointitietokone (ES), jolla suoritetaan muutoksia laitekonfiguraatioon, automaatio-ohjelmaan ja valvomonäyttöihin. Vertikaalissa integraatiossa kaikki säätimet ovat yhteydessä informaatiotasoon ja kenttälaitteisiin, joten informaatio kulkee eri hierarkiatasolta toiselle. Näin kasvatetaan hajautuksen astetta, joka parantaa järjestelmän tehokkuutta, redundanttisuutta ja lisää modulaarisuutta. (7, s. 4–8.)

Näiden kolmen hierarkiatason yläpuolella ovat tuotannon- ja toiminnanohjausjärjestelmät, jotka eivät suoranaisesti ole välttämättömiä prosessin ohjauksen kannalta (Kuva 3). Tuotannonohjausjärjestelmällä (ERP) hallitaan talous- ja tuotantojärjestelmiä yritystasolla. ERP-järjestelmä yhdistää samaan tietokantaan yrityksen valmistamat tuotteet, työntekijät, asiakkaat ja muut resurssit. ERP:llä voidaan määrittää tuotteen resepti, kuinka paljon sitä tuotetaan ja millä aikataululla. Toiminnanohjausjärjestelmä (MES) toimii porttina ERP- ja automaatiojärjestelmän välillä, jotta ERP-tasolla suunnitellut toimet saadaan toteutettua käytännössä. MES-järjestelmä voidaan integroida osaksi automaatiojärjestelmää, jolloin sen kautta on mahdollista seurata ja ohjata prosessia, sekä hallinnoida tuotantomääriä ja aikataulutusta. (9, s. 3–5.)



KUVA 3. Automaatiojärjestelmän tuotannonhallinnan kaavio (3, s. 11, Kuva 5).

2.4 Kommunikointi automaatiossa

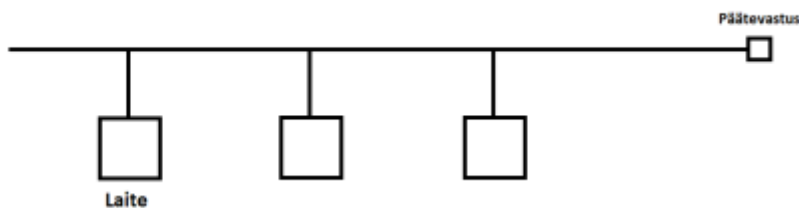
Hajautetuissa ohjausjärjestelmissä integraatio on keskeisessä osassa ja tiedon pitää olla ennen kaikkea reaaliaikaista. Automaatiolaitteiden valmistajia on suuri määrä, eivätkä kaikki tehtaan laitteet aina ole samalta valmistajalta, joten toimivaan tiedonsiirtoon tarvitaan yhteisiä standardisoituja protokollia. Protokollalla tarkoitetaan laitteiden väliseen digitaaliseen tiedonsiirtoon käytettävää metodia (10, s. 3). Automaatioverkossa kommunikointi oli aluksi sidottuna vain tietyn valmistajan laitteiden välillä, mutta nykyään käytetään pitkälti vakiintuneita kommunikointiprotokollia.

2.4.1 Verkkotopologia

Automaatiolaitteet on mahdollista kytkeä toisiinsa nähden usealla eri tavalla, mikä vaikuttaa kommunikointiominaisuuksiin. Tiedonsiirto voi tapahtua suoraan laiteelta laiteelle (point-to-point) tai sitten esimerkiksi serverin tai kytkimen välityksellä. Automaatioverkko pidetään erillään toimistoverkosta turvallisuuden ja suorituskyvyn varmistamiseksi.

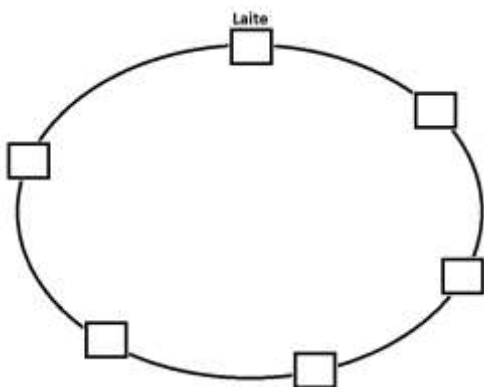
Yksinkertaisin tapa yhdistää verkon laitteita on väylätopologisesti, millä kommunikointi tapahtuu laiteelta laiteelle. Kaikki väylän kautta siirtyvä data kulkee jo-

kaisen siihen kytketyn laitteen eli solmukohdan kautta (Kuva 4). Koska väyläkaapeli ei johda takaisin järjestelmään, sille tarvitaan päätevastus väylän loppuun, jotta signaali ei heijastu takaisin ja aiheuta häiriöitä ja ylimääräistä kuormaa. Vastus valitaan vastaamaan piirin impedanssia. Väylätopologian huonona puolena on, että väylän mennessä poikki jostakin kohtaa kaikki kommunikointi loppuu. (10, s. 7.)



KUVA 4. Väylä päätevastuksella.

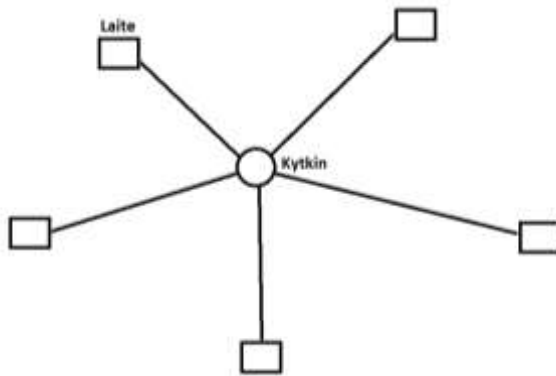
Toisena tapana on yhdistää verkon laitteet suljetuksi renkaaksi (Kuva 5). Rengastopologian rakenne on hyvin yksinkertainen, mutta tiedonsiirto on nopeampaa kuin väylämuotoisessa rakenteessa. Jokainen laite osaa lähettää ja vastaanottaa dataa sille määritellyn osoitteen mukaisesti. Jos datan lähetyksen kohdelaitteelle epäonnistuu, lähetetään se uudelleen toiseen suuntaan.



KUVA 5. Rengastopologinen kytkentä.

Verkon luotettavuutta voidaan parantaa muuttamalla sen rakennetta, jolloin myös tietoliikenteen kuormaa saadaan jaettua tasaisemmin. Tähtikytkennässä kaikki verkon laitteet ovat yhteydessä keskellä olevaan kytkimeen, joka reitittää yhteyden verkon muille laitteille (Kuva 6). Yhden laitteen väyläyhteyden katketessa

verkko ei lamaannu, koska muut laitteet ovat yhä yhteydessä kytkimen kautta. Ainoana heikkona kohtana tähtitopologiassa on keskellä oleva kytkin. Suurimassa osassa laajempia automaatiototeutuksia käytetään useampia verkkotopologisia ratkaisuja samassa, sekä rinnakkaisissa verkoissa. (10, s. 7–8.)



KUVA 6. Tähtitopologia yhdistävällä kytkimellä.

Valvomo- ja ohjelmointiasemat sijaitsevat usein kauempana prosessiasemista, joten niiden välinen kommunikointi tapahtuu serverin kautta. Serveri luo kaikille automaatioverkon laitteille verkon, jonka kautta tietoliikenne kulkee. Serveritietokoneet ovat usein tavallista tietokonetta luotettavampia ja nopeampia käsittelemään dataa sekä niissä on suurempi muisti. Redundanttisuuden varmistamiseksi rinnalla voi olla toinen varaserveri, joka otetaan käyttöön vikatilanteessa. (10, s. 9.)

2.4.2 Kenttäväyläprotokollat

Kenttäväyliä käytetään erityisesti nopeaan ja deterministiseen kommunikointiin logiikkaohjaimien ja kenttälaitteiden välillä. Kenttäväyläprotokollia on alettu käyttämään 1970-luvulta alkaen helpottamaan automaatiolaitteiden kytkemistä toisiinsa. Samaa väylää voidaan käyttää mittaamiseen ja säätämiseen, mikä vähentää kaapeloinnin määrää huomattavasti.

Profibus DP - ja Modbus RTU -protokollat käyttävät tiedonsiirtoon RS-485-sarjaliikenneväylää, jossa tiedonsiirto tapahtuu fyysisellä kerroksella 7-tasoisessa OSI-mallissa (Open Systems Interconnection Model - OSI-mallia käytetään kuvaamaan tietoliikenteen eri kerrosten välistä tiedonsiirtoa). RS-485 on nopeampi

tiedonsiirtoväylä kuin RS-232. Sen nopeus ylittää yhteen megabittiin sekunnissa 100 metrin kaapelissa. Toisin kuin RS-232:n point-to-point-kommunikoinnissa, RS-485-väylään voidaan liittää sarjaan useampia slave-laitteita, jotka keskustelevat master-laitteen kanssa väylää pitkin. OSI-mallin data- ja sovellustasoilla Profibus DP ja Modbus RTU käyttävät omia protokolliaan. (10, s. 16–17.)

Profinet on Ethernet-yhteyteen perustuva teollinen kommunikointistandardi, joka tarjoaa asyklisen tiedonsiirron automaatiolaitteiden välille. Sen ovat kehittäneet yhteistyössä PROFIBUS, PROFINET International ja Siemens (11, s. 13). Toisin kuin väylämuotoiset kommunikointitekniikat, Profinet on verkkomuotoinen ja sen väyläkapasiteetti on lähes rajaton, koska jokaiselle siihen liitetyille laitteelle määritellään looginen IP-osoite.

2.4.3 OPC-rajapinta

OPC (Open Platform Communications) on teollisuudessa käytetty kommunikointistandardi, joka kehiteltiin Microsoftin ohjelmien käyttämästä OLE-rajapinnasta (Object Linking and Embedding) HMI-ohjelmien ja prosessiasemien väliseen tiedonsiirtoon. OPC-yhteensopiva valvomo-ohjelma pystyy siis keskustelemaan lähes minkä tahansa valmistajan logiikkaohjaimen kanssa. Toiminta perustuu ajureihin, jotka muodostavat rajapinnan HMI-ohjelman ja prosessorin välille ja valitaan liitettävän laitteen kommunikointimetodin mukaisesti. (10, s. 29–30.) Informaatiota hallitaan OPC-serverin kautta, joka on asennettuna paikalliselle tietokoneelle. Sen kautta data tuodaan käyttöliittymään, joka toimii OPC Client -asiakasohjelmana.

OPC DA (Data Access) on vanhempi OPC-versio, joka käyttää vielä Microsoftin COM/DCOM2-standardeja (Component Object Model/Distributed COM). **OPC UA** (Unified Architecture) on kehittyneempi versio DA:sta eikä se ole enää riippuvainen Microsoftin ohjelmistosta. Toisin kuin vanhemmassa OPC:ssa, OPC UA:ta ei ole rajoitettu millekään tietylle alustalle ja OPC Serverin ei ole enää pakko olla asennettuna Windows-tietokoneelle. Se käyttää tiedonsiirtoon TCP-protokollaa (Transmission Control Protocol), mikä tarjoaa laajemmat mahdollisuudet datan käyttöön. Informaatiota voidaan tuoda kentältä ERP-tasolle ja se on käytettävissä internetin ylitse. (12, s. 11.)

3 SIEMENS PCS 7

SIMATIC PCS 7 on Siemensin kehittänyt hajautettu ohjausjärjestelmä, joka on kansainvälisesti laajassa käytössä prosessiteollisuuden sovellutuksissa. Järjestelmä kuuluu Siemensin TIA-konseptiin (Totally Integrated Automation), joka integroi Siemensin tuotteita kaikilla automaatiohierarkian tasoilla. PCS 7 tukee tavallisimpia kommunikointiprotokollia, kuten PROFINET-protokollaa, PROFIBUS-protokollaa ja Foundation Fieldbus -protokollaa. (13, Luku 9, s. 4–5.)

Automaatioasemina PCS 7 -järjestelmä käyttää Siemensin modulaarisia S7-400-sarjan logiikkaohjaimia PROFINET- ja PROFIBUS-liittimillä standardisoitua kommunikointia varten. Väylän kautta logiikoihin liitetään erillisiä IO-moduuleita ja slave-tason kenttälaitteita. PROFINET-protokollan välityksellä logiikalle on liitettävissä Siemensin ET200M ja ET200SP etä-IO-asemia. Etäasemilla on lähes samat ominaisuudet kuin logiikoilla, joten ne voivat toimia kenttälaitteille master-laitteina. (13, Luku 8, s. 4–5.)

3.1 Ohjelman osat

Function block (FB) eli toimilohko on graafisessa muodossa oleva ohjelman osa, joka sisältää valmiiksi ohjelmoituja toimintoja. FB koostuu tulo- ja lähtöpuolesta, joiden I/O-pisteiden kautta lohko lukee ja antaa ulos informaatiota. Toimilohkoja on määriteltynä valmiiksi Siemensin kirjastoihin yksinkertaisia ja monimutkaisempia toimintoja varten. Niitä on esimerkiksi laskentaa, vertailua ja skalauslausta varten. Siemensillä on tarjolla erilaisia symbolipaketteja, kuten PCS7 Industry Library ja Advanced Process Library (APL). APL-paketin mukana tulee muun muassa standardisoituja toimilohkoja venttiilien ja moottoreiden ohjaamiseen.

Organization block (OB) -ohjelmalohkot suorittavat toimintoja CPU:n ja ohjelman välillä. OB-ohjelmalohkot määrittävät suoritusprioriteettinsa mukaisesti eri ryhmiin. Ne voivat toimia syklisesti, CPU:n käynnistyksen yhteydessä, laitteistovian tai ohjelmavirheen ilmetessä. CPU kutsuu siis OB-lohkoja prioriteettinsa mukaisesti ja suorittaa toimenpiteitä, joita OB kutsuu.

CFC-kaavioissa (Continuos Function Chart) eli charteissa suoritetaan automaatiojärjestelmän jatkuvasti taustalla pyörivät toiminnot, kuten mittaukset ja säätölaskennat. Ohjelmarakenne koostuu kaavioon sijoitetuista toimilohkoista, joiden välisillä johdotuksilla ja sisäisillä parametreilla toiminnallisuus luodaan. CFC Editorissa on oma katalogi Siemensin määritellyillä toimilohkoilla. CFC-kaavion vasemmassa laidassa sijaitsevat tulevien signaalien viittaukset ja oikeassa laidassa lähtevien. Viittauksilla voidaan kommunikoida eri CFC-kaavioiden välillä, sekä lukea ja kirjoittaa fyysisiä tuloja ja lähtöjä.

Luodessa uutta CFC-kaaviota, ohjelma luo automaattisesti hakemistoon uuden ryhmän, jonka alle kaikki CFC-kaavioon lisättävät toimilohkot ja niiden instanssit tulevat. Toimilohkojen muuttujien dataa säilövät instanssit generoituvat automaattisesti aina kun kaavioon lisätään uusi lohko.

SFC:t (Sequential Function Chart) ovat sekvenssikaavioita. SFC-Editorilla luodaan sekvenssiohjauksia, joiden syntaksi perustuu askeleisiin, joista Siemens käyttää nimeä Sequencer. Sekvenssejä käytetään prosessin ylös- ja alasajamiseen sekä suorittamaan toimenpiteitä esimerkiksi vian ilmetessä. Yksinkertaistettuna jonkin prosessin osa on jaettuna useisiin peräkkäisiin askeleisiin, jotka sisältävät erilaisia ehtoja ja toimenpiteitä. Yhdessä askeleessa voidaan tarkastella jonkin ehdon eli tulon tai muuttujan tilaa, jonka toteutuessa suoritetaan ennalta määriteltyjä toimenpiteitä.

SFC-kaavioita on kahta eri tyyppiä: SFC Chart ja Type. SFC Chartilla voidaan suorittaa toimenpiteitä koko tehtaan tai laitoksen tasolla CFC-kaavioilla eikä sitä suoriteta kuin kerran. Sequencereiden eli askeleiden määrä on rajoitettuna kahdeksaan. SFC Typeä käytetään pienemmissä prosessin osissa, koska se pystyy kirjoittamaan arvoja vain SFC-instansseihin. SFC Typeen on mahdollista lisätä 32 askelta ja se voidaan suorittaa rajattoman monta kertaa. (13, Luku 4, s. 8.)

3.2 SIMATIC Manager

SIMATIC Manager on PCS 7 -järjestelmän ohjauskeskus, johon on integroituna kaikki työkalut, mitä projektin määrittämiseen ja suunnitteluun tarvitaan. Nämä ohjelmat ovat asennettuna ES:lle, jonka kautta projektin konfigurointi hoidetaan.

SIMATIC Managerin pohjana on hakemisto, jonka kautta löytyvät projektin prosessitagit, ohjelmalohkot ja siihen lisätyt laitteet. Hakemiston kansiot määrittelevät projektin hierarkkisen rakenteen ja myös mille asemalle ne ovat tallennettuna. SIMATIC Managerin kautta avautuu erillisiä integroituja ohjelmia objektin tyyppin mukaan, kuten CFC Editor, Graphics Designer tai HW Config.

SIMATIC Managerissa on neljä erilaista projektinäkömää eri käyttötarkoituksia varten. Component view -näkyvässä projektin objektit ovat näkyvässä fyysisen tallennuspaikan mukaisesti eri prosessiasemilla. Tämän kautta suoritetaan siis kaikki laitteiston lisäykset ja määrittelyt. Plant view -näkyvän kautta nähdään projektin kansioden kautta todellisuutta vastaava hierarkkinen rakenne, joka koostuu CFC- ja SFC-ohjelmalohkoista. Process object view on taulukkomuotoinen näkyvä ohjelman parametrien, kommenttien ja muiden attribuuttien muokkaamista varten. Sen kautta voidaan tehdä helposti ja nopeasti muutoksia yksittäisiin tai useampiin attribuutteihin rajaamalla piirejä suodattimilla. Technological list editor on tullut uutena lisäyksenä vasta PCS 7 V9.0 versioon Control Module Typeä (CMT) varten. Control Module Typen toimintaperiaate käydään tarkemmin läpi sille varatussa luvussa.

Minkä tahansa PCS 7 -ohjelman kautta voidaan avata Help-kirjasto, josta löytyy tietoa ja apua eri ohjelmalohkojen toiminnasta. Se avataan valitsemalla haluttu objekti ja painamalla näppäimistön F1-funktionappia. Help-kirjastoissa on myös hakutoiminto, mutta sitä käyttäessä kannattaa huomioida, mistä kirjasto on avattu. CFC Editorin kautta avattu Help eroaa esimerkiksi SIMATIC Managerin kirjastosta.

3.3 HW Config

HW Configilla (Hardware Config) eli laitteiston konfigurointityökalulla määritetään projektiin lisättävät laitteet, kuten CPU:t, kenttäväylät ja IO-moduulit. Käytettävissä on Siemensin kirjasto, josta lisättävät laitteet tuodaan projektiin. IO-kanavien osoitteita vastaavat symboliset nimet määritellään HW Configin puolella Edit Symbolic Names -työkalun kautta.

3.4 WinCC Explorer

Siemensin HMI-järjestelmä on nimeltään WinCC eli Windows Control Center. Sitä käytetään automaatiojärjestelmien ohjaukseen ja valvontaan Microsoft Windows -pohjaisella tietokoneella, koska sen toiminta perustuu sen ohjelmistokomponentteihin. WinCC Explorer on OS:n määrittämiseen tarkoitettu työkalu, jolla suoritetaan muun muassa näyttöjen, tagien, käyttäjien ja hälytyksien hallinta erilaisilla editoreilla.

3.5 Graphics Designer

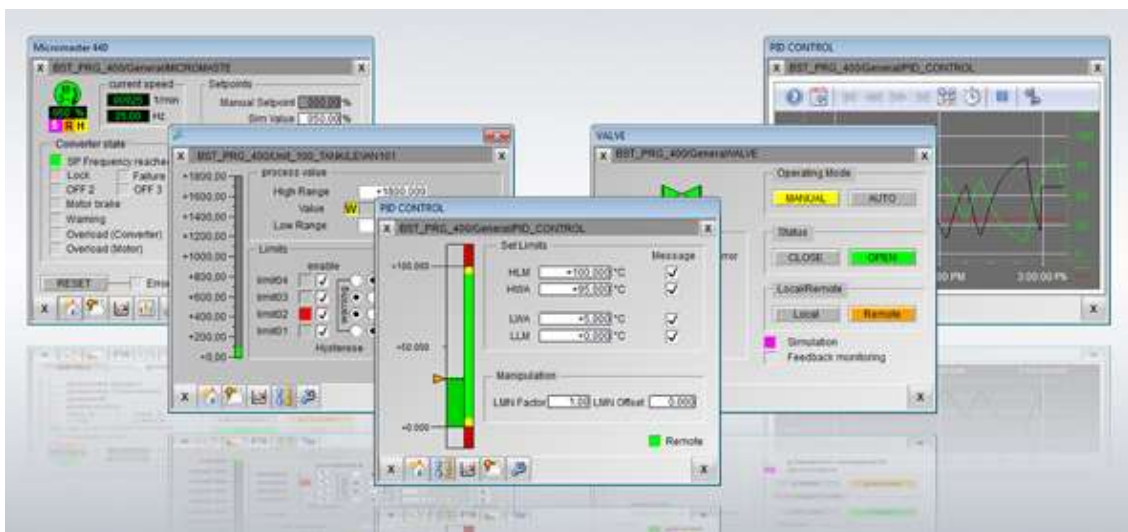
PDL-tiedostomuodossa olevat käyttöliittymän operointi- ja valvomonäytöt luodaan Graphics Designer -sovelluksessa. Se voidaan aukaista WinCC Explorerin tai SIMATIC Managerin Component View -näkyvässä OS:n alta.

Näyttöjen tekeminen aloitetaan tyhjältä ruudulta, johon piirretään prosessin ajokaavio tai muu sitä kuvaava mallinnus. Prosessilaitteet ja ympäristö piirretään viivoja ja geometrisia kappaleita käyttämällä sekä lisäämällä näytölle valmiita symboleita. Siemensin HMI Symbol Librarysta löytyy yleisimpiä venttiili-, säiliö-, kuljetin- ja pumpputyyppejä, jolloin kaikkia näytön osia ei tarvitse luoda itse tyhjästä. Kaikki näytön objektit ovat laajasti editoitavissa havainnollisia ja intuitiivisia valvomonäyttöjä ajatellen. Niitä voidaan manipuloida vapaasti prosessin muuttujien mukaan, jolloin esimerkiksi jokin symboli voidaan täyttää värillä tai piilottaa.

Prosessin seuraamista ja operointia varten kuvaan pystytään lisäämään erilaisia indikaattoreita, painikkeita tai toimilohkojen faceplate-objekteja. Graafiset mittausnäytöt ja palkit ovat selkeitä prosessin tilannetta kuvaavia indikaattoreita, jotka seuraavat mittauksen tai säädön arvoa. Prosessiin pystytään vaikuttamaan painikkeilla ja toimilohkoista luotujen block icon -objektien avulla. Block iconit ovat tarvittaessa automaattisesti generoitavia objekteja, jotka sijoitetaan hierarkkisen alueen mukaisesti valmiiksi oikeaan pdl-tiedostoon. Generointia varten CFC Editorissa toimilohkon asetuksista täytyy asettaa valinta kohtaan "Create block icon". Symboli generoidaan automaattisesti valvomokuvaan esimerkiksi venttiiliksi tai moottoriksi ja se saa toimilohkon parametrien mukaisen position, kommentin ja

parametrit. Valvomonäytöllä Block icon -kuvaketta klikkaamalla avataan faceplate-ikkuna operointia ja valvontaa varten.

Faceplatet toimivat yhden prosessilaitteen hallintapaneelina, jossa sen tärkeimmät mitta- ja säätöarvot ovat näkyvillä (Kuva 7). Faceplatet generoidaan fpt-tiedostomuotoiselta pohjalta ja ne määritellään WinCC Explorerin kautta Faceplate Editor -työkalulla. Yhdenlaista pohjaa voidaan käyttää esimerkiksi samanlaisille taajuusmuuttajaohjatuille moottoreille, jolloin painikkeet ja mittausikkunat ovat yhdenmukaisia.



KUVA 7. Siemensin valmiita faceplate-objekteja (14).

3.6 Import/Export Assistant (IEA)

Import/Export Assistant on massaohjelmointiin tarkoitettu työkalu, jolla voidaan generoida useita yksilöllisiä piirejä yhden tyyppiin pohjalta. IEA-työkalu sopii erityisesti suurempiin kokonaisuuksiin, joissa on paljon samanlaisia piirejä. Esimerkiksi mittauksien toimintaperiaate on kaikilla piireillä lähes sama, mutta yksilölliset parametrit, kuten mittausalue, hälytys- ja lukitusrajat vaihtelevat. Pohjana eli tyyppiin pohjalta luotaville ohjelmaloikoille toimii CFC-kaavio, johon toiminnallisuudet ovat määriteltynä toimilohkoilla ja johdotuksilla. IEA-työkalu generoi uudet piirit tämän tyyppiin pohjalta halutuilla yksilöllisillä piirien positiolla ja parametreillä.

SIMATIC Managerissa luodaan tyyppiin toimivan CFC-kaavion pohjalta ensin eli Process Tag Type (PTT), johon muutettavaksi halutut parametrit valitaan. Valmiista Process Tag Typeestä tehdään IEA-tiedosto, johon generoitavat piirit lisätään. IEA-tiedostoa voidaan muokata suoraan Siemensin IEA File Editorilla tai sitten kätevämmiin Microsoft Excelissä. Taulukkomuotoisessa IEA-tiedostossa rivit vastaavat piirejä ja pystysarakkeet yksilöllisiä parametreja. Taulukkoon päivitetään kaikkien piirien parametrit lähtötietojen mukaisesti, jonka jälkeen se voidaan tuoda takaisin projektiin. Piirit generoidaan Import-toiminnolla, jolloin niistä luodaan Process Tajeja syötettyjen lähtötietojen mukaisesti. CFC ja Process Tag ovat käytännössä täysin sama asia, mutta Process Tag on luotu IEA-työkälulla. Kytöksen tyyppiin eli Process Tag Typeen voi poistaa kaavion Object properties -asetuksista, jonka jälkeen se näkyy projektihakemistossa CFC:nä.

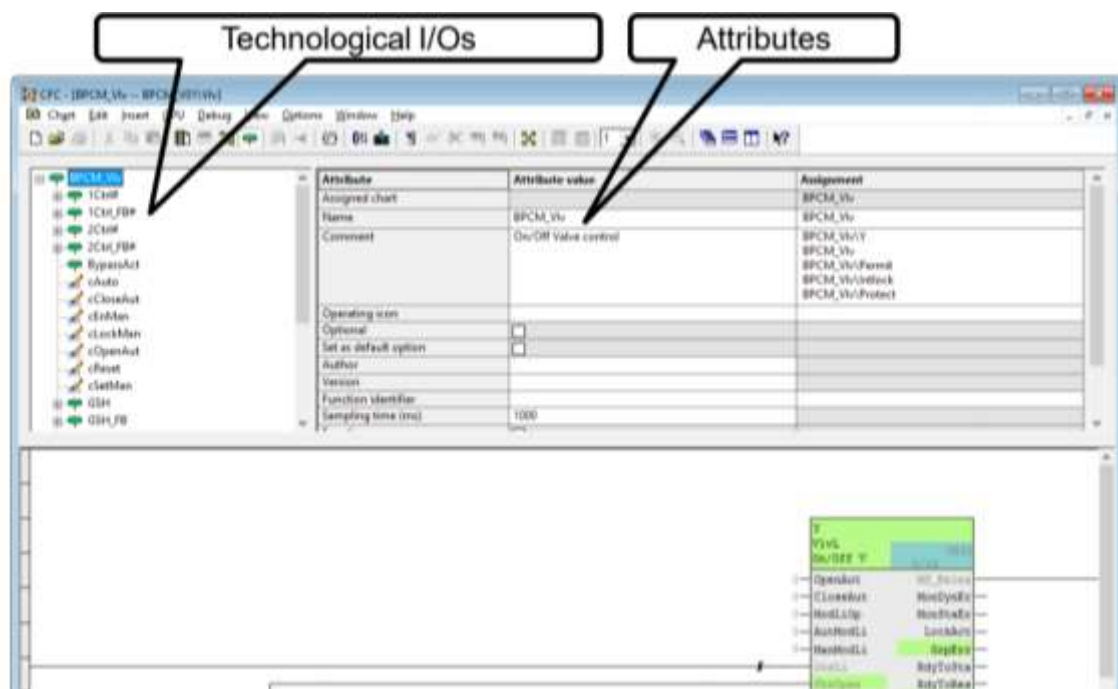
IEA-työkälun huono puoli on, että ohjelmakaavioihin aikaisemmin tehdyt muutokset kirjoitetaan yli, jos Process Tag Typeen tai IEA-tiedostoon tarvitsee tehdä päivityksiä ja piirit generoidaan uudestaan. Tästä syystä joissakin tapauksissa on järkevämpää jakaa piiri kahteen erilliseen CFC:hen, jolloin tyyppiin kautta generoitavat perustoiminnot ovat omassa CFC:ssä ja piirin yksilöllinen toiminnallisuus kokonaan erillisessä kaaviossa. Tällöin toiminnallinen osa säilyy samana, vaikka päivityksien myötä piirien generointi tehtäisiinkin uudestaan.

3.7 Control Module Type (CMT)

Control Module Type on uudempi ja monikäyttöisempi tapa useiden piirien generointiin ja muokkaamiseen. IEA-työkälulla voidaan vaikuttaa vain toimilohkoihin ja niiden I/O-pisteiden attribuutteihin, mutta CMT tukee laajemmin toimilohkojen välisiä yhteyksiä ja parametreja. Siemensin SIMATIC PCS 7 V9.0 -ohjelmistossa ja sitä uudemmissa versioissa Control Modulea varten on lisätty taulukkomuotoinen *Technological list editor* -projektinäkö, joka mahdollistaa sujuvan massaohjelmoinnin. Myös aikaisemmat IEA-työkälussa käytettävät Process Tag Tyypet ovat mahdollista kääntää CMT:ksi SIMATIC Managerissa *Create Control Module Type from Process Tag Type* -toiminnolla.

Control Module Typen toiminta perustuu ohjelmallisiin ”Technological I/O” -rajapintoihin, joiden avulla kommunikointi usean eri Control Modulen välillä on mahdollista. CMT tyyppiä luodessa halutut toimilohkot voidaan vetää hiirellä suoraan Technological IO -alueelle, jolloin ne määritellään automaattisesti Control Moduleen. Technological IO:ssa kaikki lisätyt FB:t näkyvät allekkain ja niiden alta valitaan muokattavat I/O-pisteet (Kuva 8).

Yhdestä Control Module Type tyyppiä on mahdollista luoda useampia variaatioita, joissa niiden toiminnallisuus vaihtelee. Tämä tulee erittäin hyödylliseksi ominaisuudeksi, jos esimerkiksi projektissa on moottoriäiriä erilaisilla lukituslokoilla ja kaikissa tapauksissa niitä ei tarvita. Toimilohkoja voidaan siis jättää piirikohtaisesti pois piilottamalla niitä Optional-valinnalla attribuuttien asetuksissa. (15, s. 9–10.)



KUVA 8. CFC-kaavion Technological IO:t, joihin lisätyt attribuutit näkyvät toimilohkossa vihreällä korostusvärillä (15, s.18).

Control Moduleessa on synkronointiominaisuus, jolloin piirejä generoitaessa tai päivittäessä muutokset rajataan tiettyihin Control Module Type -kaavioihin tai toimilohkojen I/O-pisteisiin. Synkronointi päivittää vain Control Moduleen valitut Technological IO -attribuutit, jolloin ohjelmalohkon muita yksilöllisiä johdotuksia

tai parametreja ei kirjoiteta ylitse. Jos yksittäiseen Control Moduleen tehdään muutoksia, niitä ei tarvitse välttämättä päivittää pohjana käytettävään Control Module Typeen. (15, s. 19.)

3.8 Process Historian (PH)

Process Historian -ohjelmaa käytetään prosessidatan pitkäaikaiseen arkistointiin. Tallennetut mittaus- ja säätöparametrien arvot ovat olennaisessa osassa prosessin optimoinnissa ja kunnossapidossa. Vian ilmetessä nähdään tarkka ajankohta ja sen hetkiset prosessin olosuhteet, jolloin diagnosointi helpottuu. Process Historian ei ole kuitenkaan välttämätön automaatiojärjestelmän osa, koska OS Serverille tallennetaan prosessidataa lyhytaikaisesti yhdestä neljään viikkoon ja viestien osalta kahdeksi kuukaudeksi. Process Historianin etuna on tietojen säilytys pitkäaikaisesti ja tallennuspaikkana voidaan käyttää vapaammin esimerkiksi verkkolevyä. (13, Luku 6, s. 2.)

Process Historian -sovellutuksen käyttöä varten ohjelmaan määritellään, mitä tageja halutaan tallentaa. Kaikkea projektin dataa ei kannata arkistoida, koska se kuormittaa järjestelmää turhaan ja muisti olisi hyvin äkkiä täynnä. SIMATIC Managerin Process Object View -näkyvässä projektin tagit tallennetaan ”Long-term archiving” -valinnalla, kunhan projektille on määriteltynä Process Historian. Tagien arvot siirretään OS:ltä ja Batch-serveriltä Microsofti SQL -serverin kautta Process Historian -tietokantaan, jossa ne ovat luettavana graafisessa ja taulukkomuodossa. Data tallennetaan fyysiselle kiintolevyille Windows Server -pohjaiselle tietokoneelle. (13, Luku 6, s. 2.)

3.9 Plant Device Management

SIMATIC Process Device Manager (PDM) on älykkäiden automaatiolaitteiden ja komponenttien konfigurointiin ja käyttöönottoon tarkoitettu ohjelma. SIMATIC PDM:stä on saatavilla eri versioita, joissa ominaisuuksien määrä vaihtelee. Ohjelman saa asennettuna erillisenä tai sitten PCS 7 -järjestelmään integroituna, jossa toimintoja on kattavammin. Se tukee yli 3 500 Siemensin laitetta ja yli 200

muiden valmistajien laitteita. Näihin kuuluvat etä-IO:t, kompaktit säätimet, kenttälaitteet ja muut automaatiojärjestelmän alla olevat komponentit. (13, Luku 7, s. 2.)

SIMATIC Maintenance Station (MS) toimii ohjausjärjestelmän rinnalla mahdollistaen ennakoivan kunnossapidon sekä laitteiden diagnostiikan. Analysoimalla mekaanisten kenttälaitteiden toiminnan ja kunnan heikkenemistä voidaan ennakoita yllättäviä laitteiden rikkoutumisia. Kenttälaitteille, joilla ei ole sisäistä älyä, on käytettävissä AssetMon-toimilohko integroiduilla diagnostiikkatoiminnolla, joka analysoi mekaanisen laitteen kuntoa, tilaa ja mahdollisia vikoja. SIMATIC MS:n kautta kunnossapidon henkilöstö voi aikatauluttaa huoltoja, asettaa kenttälaitteita huoltotilaan ja kuitata huoltoja tehdyiksi. Diagnostiikkadataa voidaan viedä käsiteltäväksi tarvittaessa ylemmille MES- ja ERP-tasolle XML-tiedosto-muodossa. (13, Luku 7, s. 14–17.)

SIMATIC PDM Maintenance Station (PDM MS) toimii täysin erillisenä asemana riippumattomana PCS 7 -ohjausjärjestelmästä ja se käyttää hyödykseen PDM:n kommunikointirajapintoja. PDM Maintenance Station soveltuu erityisesti ratkaisuihin, joissa se halutaan jälkiasentaa vanhempaan PCS 7 -projektiin tai jonkin muun valmistajan automaatiojärjestelmään. PDM MS:n avulla voidaan myös kasvattaa tuettavien laitteiden määrää konfiguroimalla Maintenance Station -ohjelmia rinnakkain. (13, Luku 7, s. 19.)

3.10 SIMATIC BATCH

SIMATIC BATCH on integroituna PCS 7 -järjestelmään ja sitä käytetään eräprosessien käyttämiseen ja valvomiseen. Batch process eli eräprosessissa valmistetaan jotain tuotetta sarjoina eikä tuotanto ole jatkuvaa. Samaa automaatiolinjastoa saatetaan käyttää useamman tuotetyypin valmistukseen, jolloin prosessin ohjausjärjestelmä tarvitsee muutoksia sen säätömenetelmiin. Jokaiselle tuotetyypille määritellään oma resepti, jota vaihdellaan tuotteen mukaan. SIMATIC BATCH:issa resepteillä vaikutetaan prosessin ajamiseen muun muassa muuttamalla SFC-kaavioiden rakennetta, instansseja ja asetusarvoja.

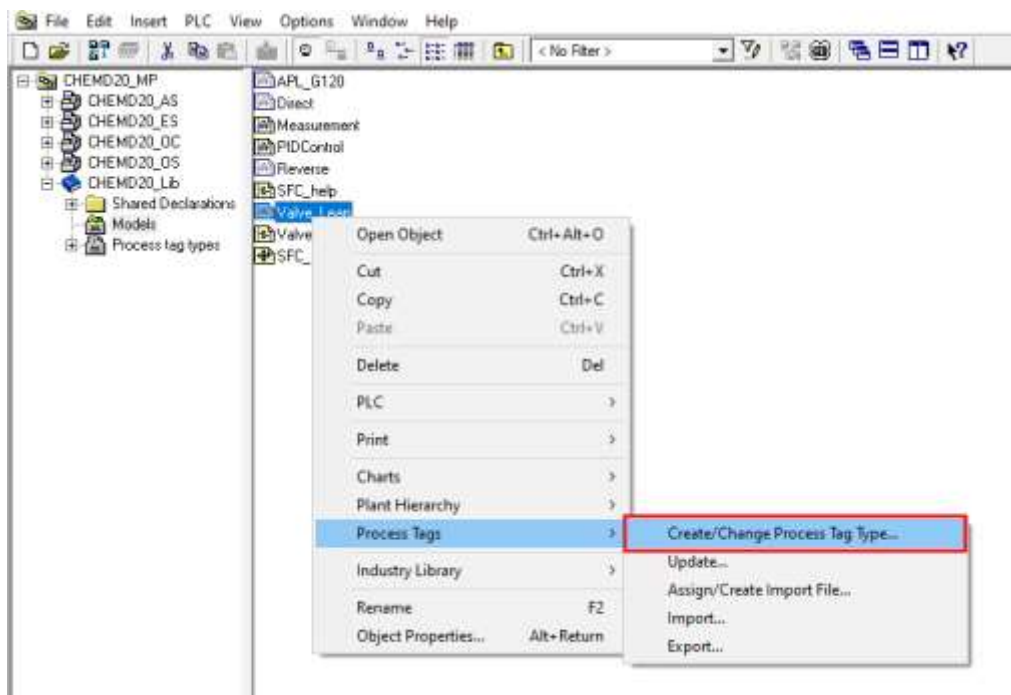
Kaikki eräprosesseihin liittyvä toiminta hoidetaan Batch Control Center -ohjelman (BatchCC) kautta, joka on SIMATIC BATCH:in ohjauskeskus. Sen graafisessa käyttöliittymässä hallinnoidaan muun muassa reseptejä, kaavoja, materiaaleja ja datan käsittelyä. BatchCC:n välityksellä avataan eräprosessien tarkempaa konfigurointia varten olevat Repice Editor-, Batch OS Control- ja Batch planning -työkalut. (13, Luku 12, s. 6–8.)

4 PIIRIEN GENEROINTI IEA-TYÖKALULLA

Projektissa on suuria määriä taajuusmuuttaja- ja SIMOCODE-ohjattuja moottoreita, säätöventtiileitä ja mittauksia, joten sovellusohjelmointiin on järkevintä käyttää IEA-työkalua. Samanlaisista piireistä on luotuna tyyppiipiirit, joita käytetään pohjana generoinnille. Kun peruspiirit saadaan tehtyä nopeasti IEA-työkalun avulla, säästetään tässä huomattavasti arvokasta aikaa, jota voidaan käyttää tarkempien toiminnallisuuksien ohjelmointiin. Seuraavaksi käydään läpi On-Off-venttiilien piirien generointi ohjelmaan tyyppiipiirin pohjalta. Muihin projektin piireihin sovellettiin samaa toimintaperiaatetta, mutta niiden Process Tag Typet ja IEA-tiedostot poikkeavat sisällöltään ja rakenteeltaan piirikohtaisesti.

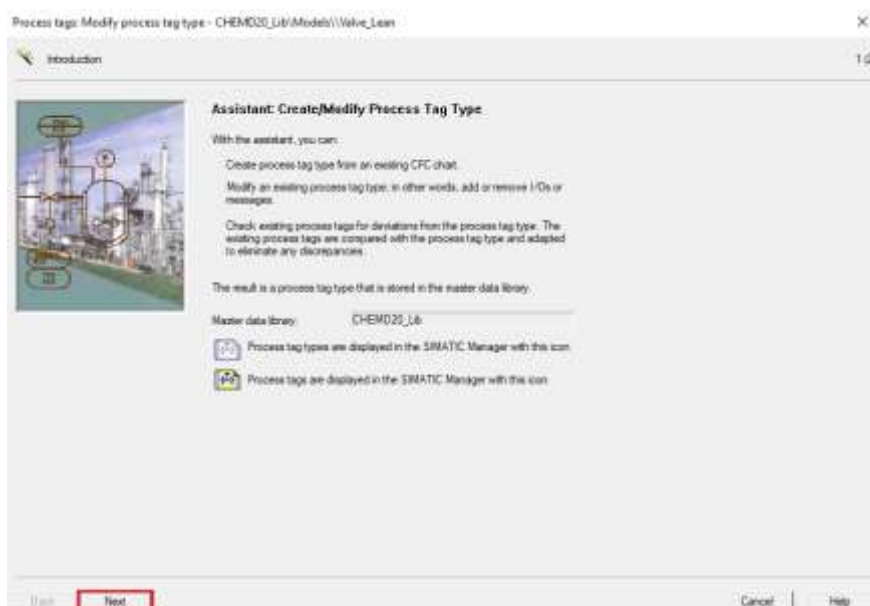
4.1 Process Tag Typen luominen

Piirien generoinnin ensimmäisenä askeleena CFC-muotoisesta tyyppiipiiristä tehdään Process Tag Type. Aluksi navigoidaan Simatic Managerin Plant View -näkyvässä pohjana käytettävän tyyppiipiirin kohdalle. Tähän esimerkkiin käytettiin Siemensin valmista On-Off-venttiilien ohjaukseen tarkoitettua Valve_Lean-tyyppiipiiriä pienillä muutoksilla. Pohjana toimivaa CFC-kaaviota klikataan hiiren oikealla painikkeella, josta valitaan *Create/Change Process Tag Type...* (Kuva 9).



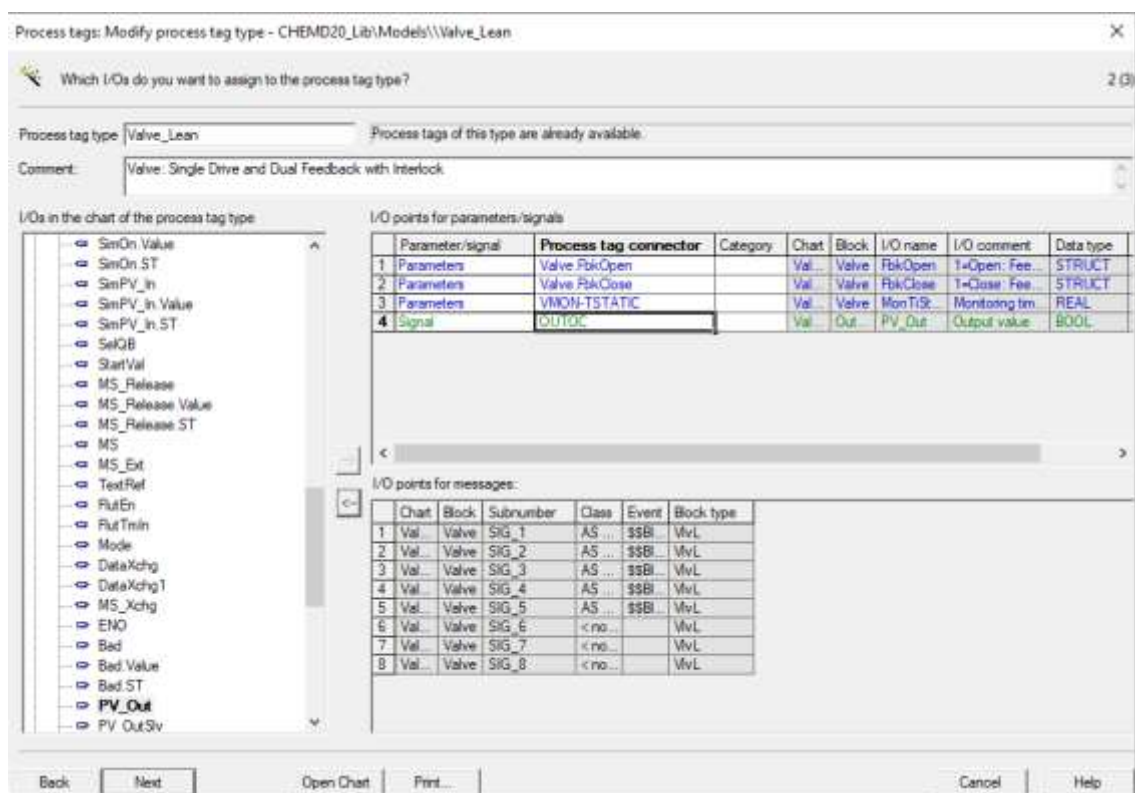
KUVA 9. Process Tag Typen luominen ja muokkaaminen.

Avautuvassa ikkunassa näkyy käytettävä Master data library, jonne Process Tag Type tallennetaan. Tästä jatketaan klikkaamalla *Next*-painiketta (Kuva 10). Jos tyyppiin on ennalta olemassa Process Tag Type poikkeavalla nimellä, niin ohjelma avaa ponnahdusikkunan, jossa se varoittaa muokattavan Process Tag Typen nimeksi vaihdettavan nykyisen tyyppiin nimen.



KUVA 10. Create/Modify Process Tag Typen käyttämä Master data library.

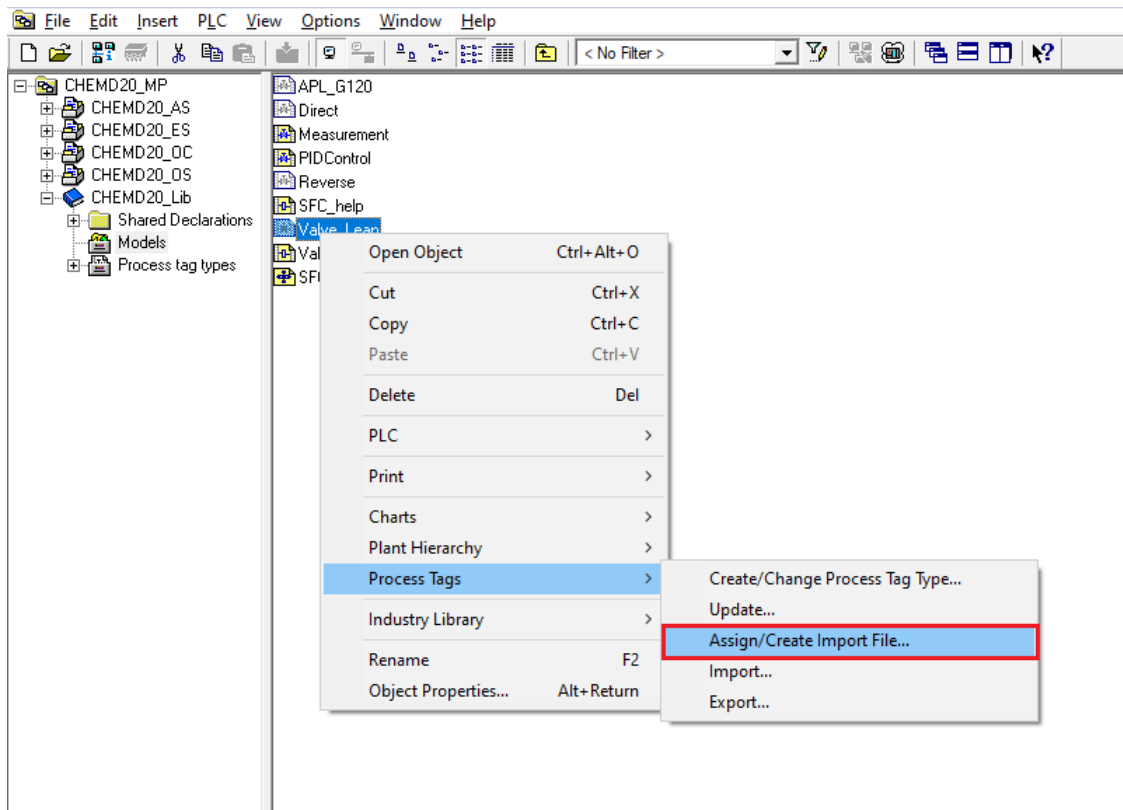
Seuraavaksi valitaan mitkä toimilohkot ja niiden nastat eli I/O-pisteet halutaan ottaa mukaan IEA-tiedostoon muokattaviksi sarakkeiksi. Vasemmassa laidassa näkyvät tyyppiin sisältyvät toimilohkot, joiden alla ovat niiden tulo- ja lähtöpuolien I/O-pisteet (Kuva 11). Kaksoisklikkaamalla tai käyttämällä keskellä olevia nuolinäppäimiä objekteja voidaan siirtää oikealle puolelle, jolloin ne ovat valittuna IEA-tiedostoa varten. Alemmassa taulukossa näkyvät hälytystekstit, jotka lisäävät IEA-tiedostoon etsimällä vasemmanpuoleisesta hakemistosta kyseisen lohkon Messages-objekti ja valitsemalla se. Hälytystekstejä voidaan muokata myöhemmin IEA-tiedostossa.



KUVA 11. Process Tag Typen muokkausnäkyvä. Oikealla ylhäällä ovat valitut parametrit ja signaalit sekä alempana ovat hälytykset.

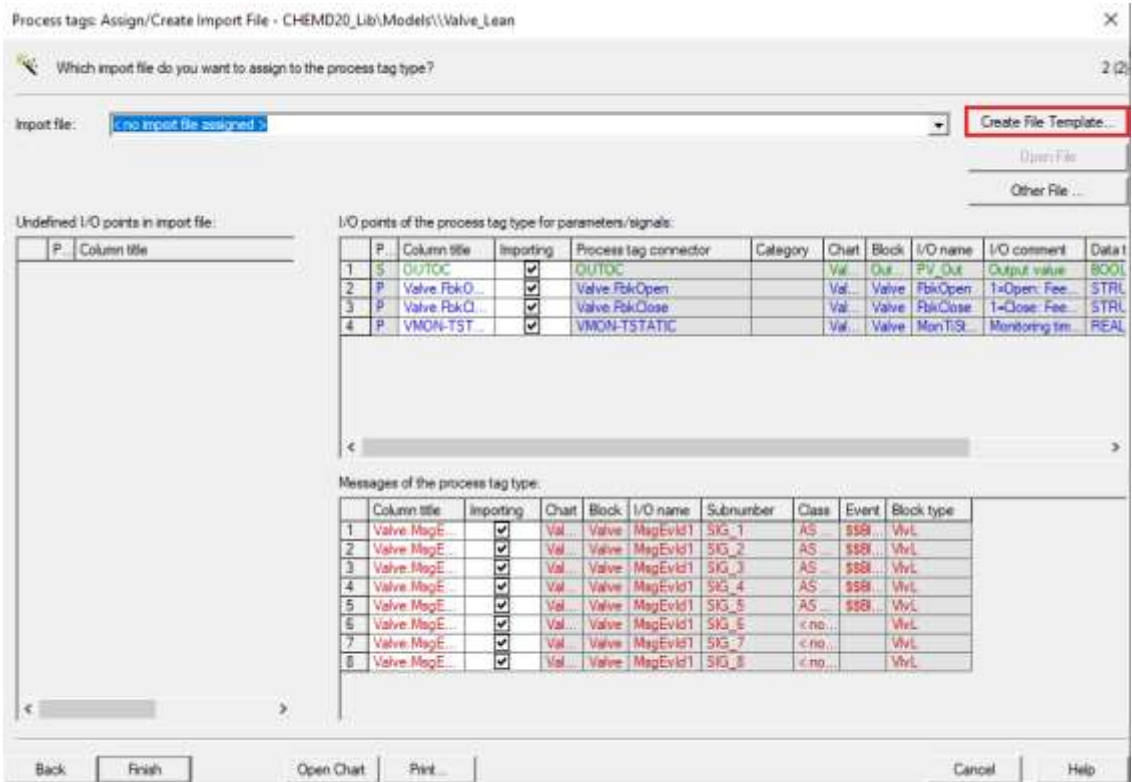
Kun halutut I/O-pisteet ovat valittuna, klikataan *Finish*. Jos toiminto on onnistunut, ikkunan voi sulkea Exit-painikkeesta, jonka jälkeen Process Tag Type on määriteltynä. Muutoksia PTT:n valinnoille voi palata tekemään myöhemmin samalla tavalla. Nyt voidaan myös huomata, että Valve_Lea-tyyppiin tiedostomuoto on muuttunut hakemistossa CFC:stä Process Tag Typeksi.

Määrittelystä Process Tag Typestä on nyt mahdollista tehdä IEA-tiedosto, joka on vapaasti käsiteltävissä ja siirrettävissä. IEA-tiedosto tuodaan projektista klikkaamalla kuvan 12 mukaista *Assign/Create Import File...* -painiketta PTT:n oikean hiiren painikkeen valinnoista.



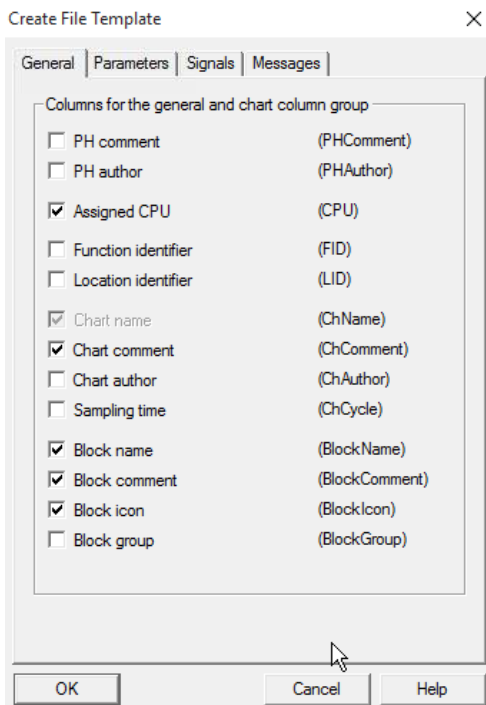
KUVA 12. Luodaan IEA-tiedosto.

Työkalua käynnistäessä avautuu vastaavanlainen informaatioikkuna kuin Process Tag Typeä luodessa, joka ilmoittaa käytettävän Master data libraryn. Ikkunasta klikataan *Next*-painiketta, jonka jälkeen avautuu kuvan 13 mukainen näkymä.



KUVA 13. IEA-tiedoston I/O-pisteiden ja hälytyksien valinta, sekä Create File Template... -painikkeella tiedoston luominen ja tallennus.

Ikkunassa näkyvät kaikki aikaisemmin Process tag tyyppiin valitut I/O-pisteet. Importing-sarakkeesta kannattaa tarkastaa, että kaikkien haluttujen I/O-pisteiden kohdalla on valinta, jotta ne tulevat mukaan IEA-tiedostoon. Seuraavaksi Create File Template -painiketta klikkaamalla IEA-tiedostolle valitaan nimi ja tallennusjainti. Tallennuksen jälkeen ohjelma avaa Create File Template -ponnahdusikkunan toimilohkokohtaisten sarakkeiden näkymää varten (Kuva 14).



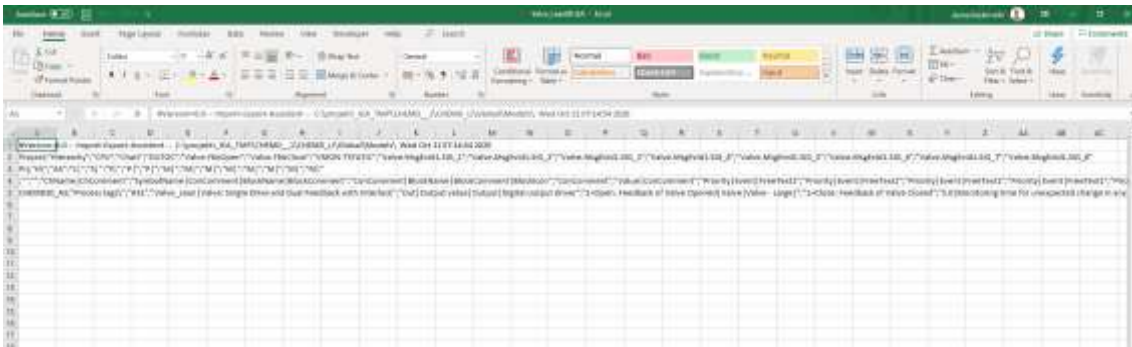
KUVA 14. Create File Template -ikkunasta valitaan toimilohkoille ja I/O-pisteille näkyville tulevat sarakkeet.

Aikaisemmin Process Tag Typeä luodessa valittiin, mitkä tyyppiin toimilohkojen I/O-pisteet tulevat IEA-tiedostoon muokattavaksi. Nyt kaikille toimilohkoille määritellään yhteisesti, että mitkä sarakkeet tulevat näkyviin. Esimerkiksi tässä projektissa valitsin oletusvalintojen lisäksi General-välilehdeltä Block Namen ja Block Iconin tulevan näkyviin IEA-tiedoston taulukkoon erillisinä sarakkeina. Block Name on toimilohkon nimi ja Block Icon määrittelee siitä automaattisesti generoitavan kuvakkeen tyyppin valvomonäytöille. Messages-välilehdeltä valitsin hälytyksiä varten tulevaisuuden varalle sarakkeet Priority, Event ja Free Text 1. Klikkaamalla OK luodaan IEA-tiedosto ja nyt se on löydettävissä aikaisemmin määritellystä sijainnista.

4.2 IEA-tiedoston muokkaaminen Excelissä

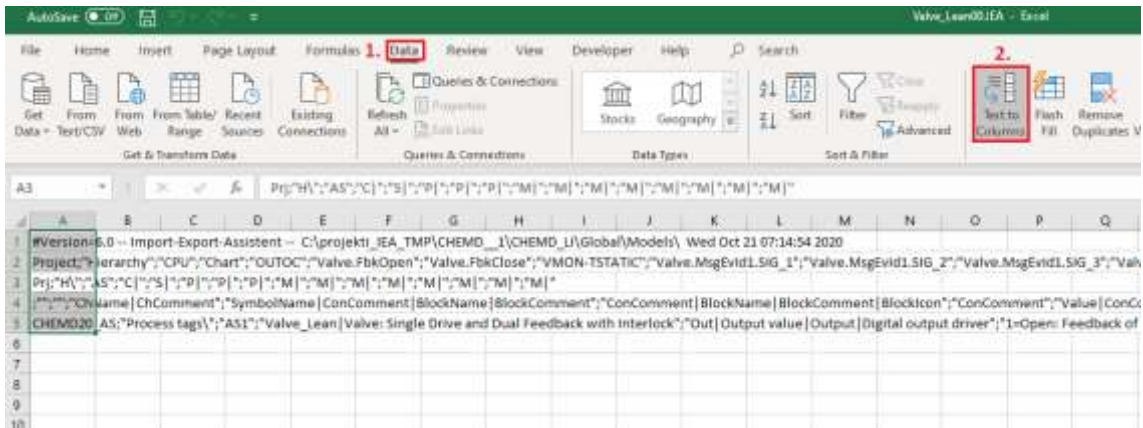
Projektista tuodun IEA-tiedoston voi avata muokkausta varten Siemensin IEA File Editorilla, jonka kautta taulukko on automaattisesti näkyvillä oikein. Laajempien kokonaisuuksien muokkaaminen onnistuu kuitenkin helpommin Excelissä.

IEA-tiedosto ei avaudu oletuksena Excelissä, vaan Windows yrittää käyttää tähän IEA File Explorer -ohjelmaa, jos se on asennettuna. Tiedosto avataan Excelissä klikkaamalla sitä hiiren oikealla painikkeella, valitsemalla *Open with...*, ja hakeamalla tähän käytettäväksi ohjelmaksi Excelin. Kun IEA-tiedosto saadaan avattua Excelissä, huomataan sen rakenteen olevan hyvin sekava ja hankalasti luettava muodossa (Kuva 15). Taulukon koko sisältö on pakattuna vasemmanpuoleisiin soluihin yhtenä merkkijonona. Näkymä saadaan korjattua jakamalla A-sarakkeen tekstit erillisiksi soluiksi listaerotusmerkkiä käyttäen. Merkkijonoa tutkimalla huomataan nopeasti toistuva erotusmerkki, joka on tässä tapauksessa puolipiste. Listaerotusmerkkinä voi olla myös pilkku riippuen Windowsin asetuksista. Tätä tarkastellaan tarkemmin myöhemmässä luvussa.



KUVA 15. IEA-tiedosto Excelissä avattuna.

Aluksi valitaan kaikki tekstiä sisältävät solut, mikä onnistuu helpoiten klikkaamalla yhtä A-sarakkeen solua ja käyttämällä näppäinyhdistelmää *Ctrl + A*. Kun solut ovat valittuna, mennään Excelissä Data-välilehdelle, josta valitaan *Text to Columns*-työkalu (Kuva 16).



KUVA 16. Valitut solut ja Text to Columns -työkalun avaus.

Ensimmäisestä ikkunasta valitaan *Delimited*, joka sopii IEA-tiedoston tekstin sisältötyyppiin (Kuva 17). IEA-tiedoston listaerotusmerkinä huomataan olevan puolipiste (semicolon), joten tämä valitaan seuraavassa avautuvassa ikkunassa. Jos IEA-tiedoston erotusmerkinä on pilkku (comma), valitaan tämä puolipisteen sijasta. Kolmannessa ikkunassa klikataan *Finish*, jonka jälkeen IEA-tiedoston sarakkeet näkyvät oikeassa formaatissa (Kuva 18).



KUVA 17. Datan sisältötyypin ja listaerotusmerkin valinta.



KUVA 18. IEA-tiedosto muutettuna oikeaan muotoon.

Sarakkeiden ollessa eroteltuna voidaan jatkaa IEA-tiedoston täyttämistä projektin lähtötietoaineiston piirien yksilöllisillä tiedoilla. *Prj*-sarakkeeseen tulee kohteena

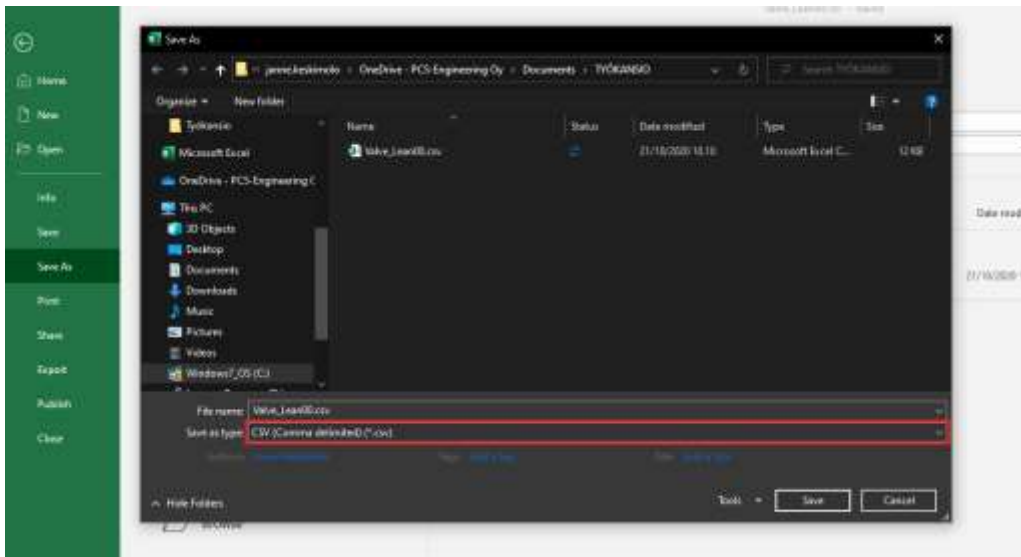
olevan projektin nimi ja *Hierarchy*-sarakkeeseen piirien hierarkkinen alue, joka pitää laittaa täysin oikein vastaamaan projektialuetta. Jos alueen nimi poikkeaa projektin olemassa olevista kansioista, piirejä generoitaessa ohjelma luo virheelisellä nimellä olevan uuden kansion ja sijoittaa luotavat piirit sinne.

Jos piiriin täytyy sijaita toisen kansion sisällä, hierarkkisen nimen tulee olla muotoa "Taso_1_kansio\Taso_2_kansio". Kuvassa 19 näkyy täytetty IEA-tiedosto, jossa osa piireistä on sijoitettuna D20-Cooking_1-kansion alle.

Component ID	Component Name	Description	Feedback Loop
1	PIIRED	PIIRED	PIIRED
2	PIIRED	PIIRED	PIIRED
3	PIIRED	PIIRED	PIIRED
4	PIIRED	PIIRED	PIIRED
5	PIIRED	PIIRED	PIIRED
6	PIIRED	PIIRED	PIIRED
7	PIIRED	PIIRED	PIIRED
8	PIIRED	PIIRED	PIIRED
9	PIIRED	PIIRED	PIIRED
10	PIIRED	PIIRED	PIIRED
11	PIIRED	PIIRED	PIIRED
12	PIIRED	PIIRED	PIIRED
13	PIIRED	PIIRED	PIIRED
14	PIIRED	PIIRED	PIIRED
15	PIIRED	PIIRED	PIIRED
16	PIIRED	PIIRED	PIIRED
17	PIIRED	PIIRED	PIIRED
18	PIIRED	PIIRED	PIIRED
19	PIIRED	PIIRED	PIIRED
20	PIIRED	PIIRED	PIIRED
21	PIIRED	PIIRED	PIIRED
22	PIIRED	PIIRED	PIIRED
23	PIIRED	PIIRED	PIIRED
24	PIIRED	PIIRED	PIIRED
25	PIIRED	PIIRED	PIIRED
26	PIIRED	PIIRED	PIIRED
27	PIIRED	PIIRED	PIIRED
28	PIIRED	PIIRED	PIIRED
29	PIIRED	PIIRED	PIIRED
30	PIIRED	PIIRED	PIIRED
31	PIIRED	PIIRED	PIIRED
32	PIIRED	PIIRED	PIIRED
33	PIIRED	PIIRED	PIIRED
34	PIIRED	PIIRED	PIIRED
35	PIIRED	PIIRED	PIIRED
36	PIIRED	PIIRED	PIIRED
37	PIIRED	PIIRED	PIIRED
38	PIIRED	PIIRED	PIIRED
39	PIIRED	PIIRED	PIIRED
40	PIIRED	PIIRED	PIIRED
41	PIIRED	PIIRED	PIIRED
42	PIIRED	PIIRED	PIIRED
43	PIIRED	PIIRED	PIIRED
44	PIIRED	PIIRED	PIIRED
45	PIIRED	PIIRED	PIIRED
46	PIIRED	PIIRED	PIIRED
47	PIIRED	PIIRED	PIIRED
48	PIIRED	PIIRED	PIIRED
49	PIIRED	PIIRED	PIIRED
50	PIIRED	PIIRED	PIIRED
51	PIIRED	PIIRED	PIIRED
52	PIIRED	PIIRED	PIIRED
53	PIIRED	PIIRED	PIIRED
54	PIIRED	PIIRED	PIIRED
55	PIIRED	PIIRED	PIIRED
56	PIIRED	PIIRED	PIIRED
57	PIIRED	PIIRED	PIIRED
58	PIIRED	PIIRED	PIIRED
59	PIIRED	PIIRED	PIIRED
60	PIIRED	PIIRED	PIIRED
61	PIIRED	PIIRED	PIIRED
62	PIIRED	PIIRED	PIIRED
63	PIIRED	PIIRED	PIIRED
64	PIIRED	PIIRED	PIIRED
65	PIIRED	PIIRED	PIIRED
66	PIIRED	PIIRED	PIIRED
67	PIIRED	PIIRED	PIIRED
68	PIIRED	PIIRED	PIIRED
69	PIIRED	PIIRED	PIIRED
70	PIIRED	PIIRED	PIIRED
71	PIIRED	PIIRED	PIIRED
72	PIIRED	PIIRED	PIIRED
73	PIIRED	PIIRED	PIIRED
74	PIIRED	PIIRED	PIIRED
75	PIIRED	PIIRED	PIIRED
76	PIIRED	PIIRED	PIIRED
77	PIIRED	PIIRED	PIIRED
78	PIIRED	PIIRED	PIIRED
79	PIIRED	PIIRED	PIIRED
80	PIIRED	PIIRED	PIIRED
81	PIIRED	PIIRED	PIIRED
82	PIIRED	PIIRED	PIIRED
83	PIIRED	PIIRED	PIIRED
84	PIIRED	PIIRED	PIIRED
85	PIIRED	PIIRED	PIIRED
86	PIIRED	PIIRED	PIIRED
87	PIIRED	PIIRED	PIIRED
88	PIIRED	PIIRED	PIIRED
89	PIIRED	PIIRED	PIIRED
90	PIIRED	PIIRED	PIIRED
91	PIIRED	PIIRED	PIIRED
92	PIIRED	PIIRED	PIIRED
93	PIIRED	PIIRED	PIIRED
94	PIIRED	PIIRED	PIIRED
95	PIIRED	PIIRED	PIIRED
96	PIIRED	PIIRED	PIIRED
97	PIIRED	PIIRED	PIIRED
98	PIIRED	PIIRED	PIIRED
99	PIIRED	PIIRED	PIIRED
100	PIIRED	PIIRED	PIIRED

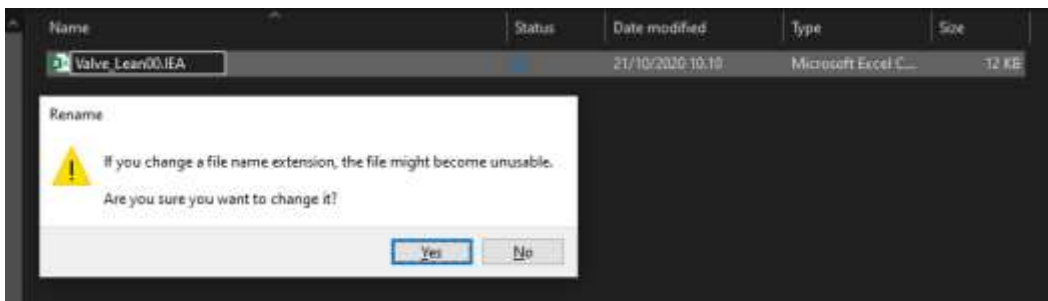
KUVA 19. IEA-tiedosto täytettynä Excelissä.

Text to Columns -työkalulla IEA-tiedoston sisältö muutettiin helpommin luettavaksi, mutta tällaisenaan sitä ei voida tuoda takaisin projektiin, koska ohjelma ei osaa lukea taulukkoa tässä muodossa. IEA-tiedosto täytyy muuttaa alkuperäiseen formaattiin tallentamalla se nimellä ja vaihtamalla sen tiedostotyyppiä .CSV (Comma Delimited) (Kuva 20).



KUVA 20. IEA-tiedoston tallennus CSV-muodossa.

Kuvan 20 mukaisen tallennuksen jälkeen IEA-tiedoston päätteeksi on luonnollisesti nyt ".csv", joka korjataan muotoon ".IEA". Tiedostopäätettä muuttaessa avautuu varoitusikkuna, jossa varoitetaan tiedoston mahdollisesta korruptoitumisesta. Klikataan siitä huolimatta Yes, jolloin lopulliseksi tiedostonimeksi tulee Lean_Valve00.IEA (Kuva 21).



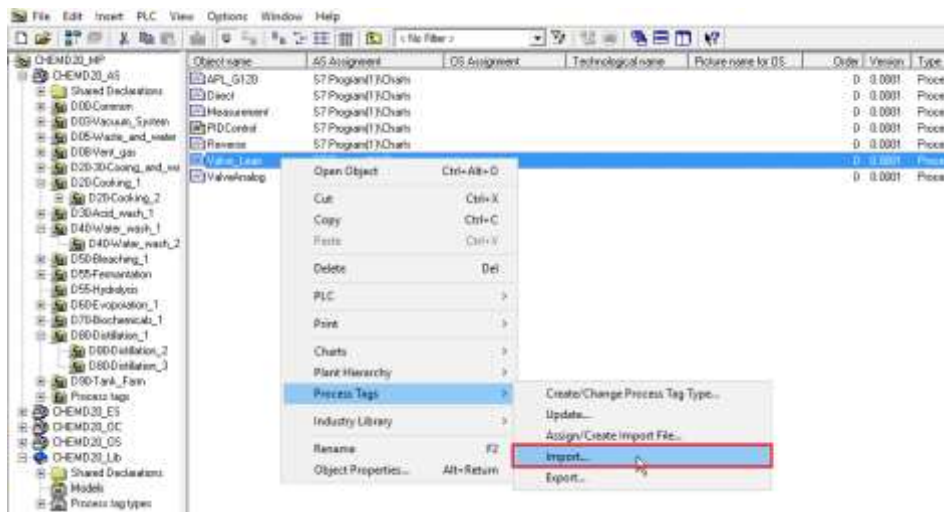
KUVA 21. Tiedostopäätteen muuttaminen .csv:stä .IEA:ksi.

IEA-tiedosto on nyt oikeassa formaatissa ja se voidaan tarkastaa avaamalla Valve_Lean.IEA-tiedosto IEA File Editorissa. Tiedoston pitäisi avautua huomautuksitta ja tehtyjen piirien tietojen pitäisivät olla jaoteltuna oikein (Kuva 22). IEA-tiedosto on nyt valmiina projektiin takaisin tuomista varten ja piirien generointi voidaan aloittaa.

4.4 IEA-tiedoston tuominen projektiin

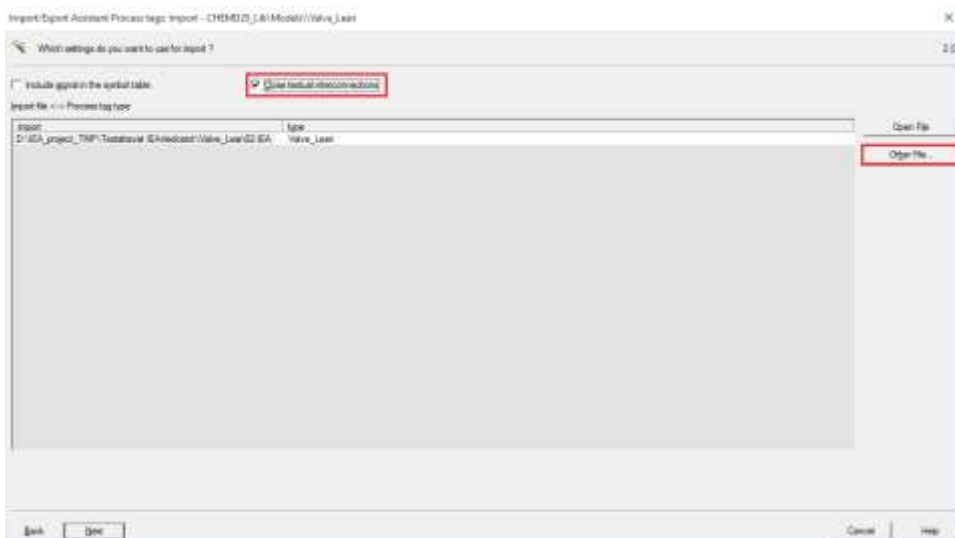
Ennen IEA-tiedoston tuomista projektiin on suositeltavaa ottaa varmuuskopio, jos generoinnissa ilmenee ongelmia. IEA-tiedostossa voi olla satojakin piirejä ja virheellisen importtauksen seurauksena näitä on hyvin työlästä hakea ja poistaa yksitellen projektihakemistosta. Importtaus on suositeltavaa koeajaa ensin pienemmällä määrällä testipiirejä, ennen kuin koko paketti ajetaan läpi. IEA-tiedostosta voidaan ottaa esimerkiksi kopio ja tehdä siitä kokeiluversio. Tästä poistetaan suurin osa piireistä ja jätetään jäljelle vain muutama piiri. Kokeiluversion importtauksen onnistuttua ja halutunlaisen lopputuloksen varmistuttua voidaan projektiin generoida loputkin piirit suhteellisen turvallisesti.

Muokatun IEA-tiedoston importtaus aloitetaan samasta paikasta kuin Process Tag Typen luominen tapahtuu. Navigoidaan Plant View -näkyvässä IEA-tiedostoa vastaavaan tyyppiin kohdalle ja klikataan hiiren oikean painikkeen Process Tag -valikon kautta *Import* (Kuva 23).



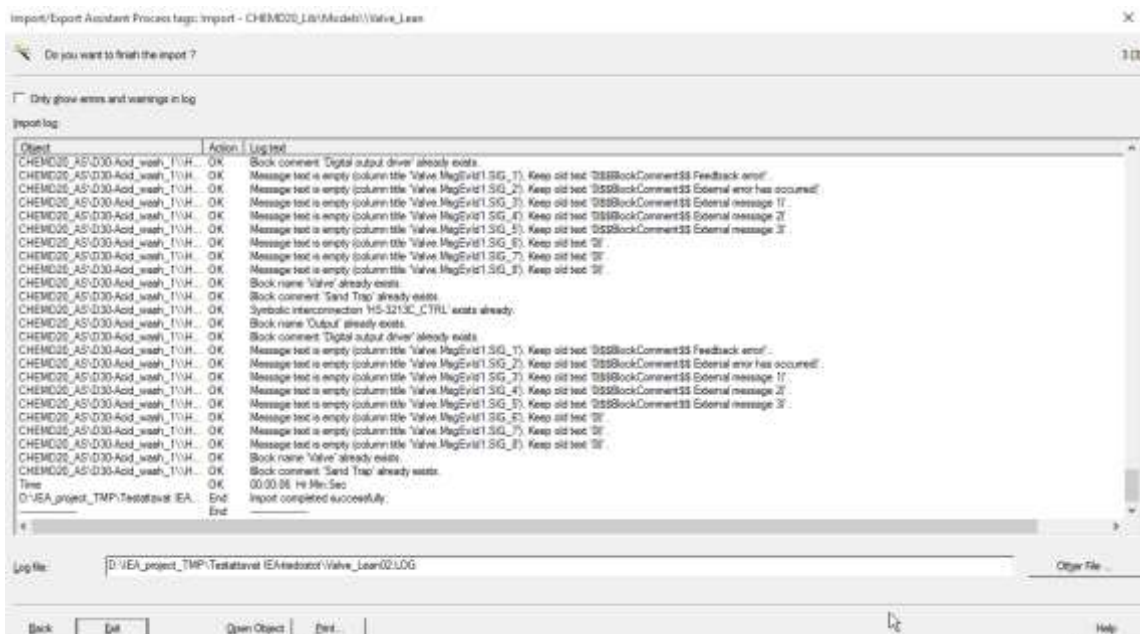
KUVA 23. IEA tiedoston importtaus.

Avautuvassa ikkunassa nähdään oletuksena kyseisestä Process Tag Typestä luotu IEA-tiedosto. Jos sen nimi on vaihdettu muokkausvaiheessa tai tiedostoa on siirretty, täytetty IEA-tiedosto valitaan *Other file...* -painikkeesta (Kuva 24). Kohtaan "Close textual interconnections" laitetaan valinta, jotta mahdolliset piirien yksilölliset toimilohkojen viittaukset johdotetaan automaattisesti.



KUVA 24. Import-työkalun asetukset ja käytettävän IEA-tiedoston valinta.

Oikean IEA-tiedoston ollessa valittuna klikataan *Next* ja sitten *Finish*, jolloin piirien generointi alkaa. Importin lokia seuraamalla pitäisi nähdä, että piirejä generoidaan eikä siihen ilmesty virheilmoituksia. Toimenpiteen onnistuttua lokin loppuun tulee teksti "Import completed successfully" ja Import-ikkunan voi sulkea (Kuva 25).



KUVA 25. Import-työkalun lokitiedot.

Importin lokin varoitukset eivät vaikuta suoraan muiden piirien generointiin, mutta ne kannattaa käydä huolellisesti läpi ja suorittaa korjaukset tarvittaessa. Yleinen

virheilmoitus syntyy väärästä desimaalierottimesta, joka riippuu Windowsin asetuksista. Jos IEA-tiedostossa ilmenee virheitä, jokaisesta generoitavasta piiristä aukeaa erillinen virheestä kertova ponnahdusikkuna. Tämän takia IEA-tiedoston toimivuus kannattaa kokeilla ensin vain muutamalla piirillä. Yhdestäkin virheilmoituksesta (Errorista) generointi perutaan, jolloin yhtään uutta piiriä ei luoda.

4.5 Generoitujen piirien päivittäminen

Ensimmäisellä generointikerralla projektin piirien lähtötiedot tai tyyppipiirien toiminnot eivät välttämättä ole vielä lopulliset, jolloin näitä joudutaan päivittämään jälkikäteen. Jos muutoksia tai lisäyksiä tulee vain IEA-tiedostoon, tiedosto tallennetaan ja Import-vaihe suoritetaan uudestaan. Tässä kannattaa huomioida, että ohjelmakaavioihin ensimmäisen generoinnin jälkeen tehdyt muutokset kirjoitetaan generoinnissa yli.

IEA-työkalun huonona puolena on, että kun tyyppipiiriin tehdään ohjelmamuutoksia nykyisen Process Tag Typen sisältämiin toimilohkoihin, silloin se pitää määrittää uudestaan alusta lähtien. Process Tag Typen päivityksen jälkeen aikaisemmin tehty IEA-tiedosto ei enää toimi uuden PTT:n kanssa, jolloin IEA-tiedostokin täytyy luoda uudestaan. Uuden import-tiedoston täyttäminen onnistuu nopeasti, kun vanhasta tiedostosta kopioidaan projektin piirien sarakkeet oikeisiin kohtiin.

Tämän projektin ohjelmointivaiheessa nähtiin järkeväksi suorittaa kaikkien generoitavien piirien poistaminen varmuuden vuoksi, ennen kuin ne tuodaan uudelleen projektiin. IEA-tiedoston yläosaan kopioidaan jo tiedostossa olemassa olevat piirit ja ne määritetään poistettaviksi. Kun tiedosto on avattuna IEA File Editorissa, taulukkoon lisätään uusi ImportMode-sarake klikkaamalla taulukkoa hiiren oikealla painikkeella ja valitsemalla sieltä "Expand Column Group". Avautuvasta ikkunasta valitaan "Import mode" ja painetaan OK (Kuva 26).

1	Project	Hierarchy	CPU	ChName
2				
3	Pj	H\	AS	
4	CHEMD20_AS	D20-Cooking_1\	AS1	HS-2349
5	CHEMD20_AS	D20-Cooking_1\D20-Cooking_2\	AS1	HS-2451
6	CHEMD20_AS	D20-Cooking_1\D20-Cooking_2\	AS1	HS-2452
7	CHEMD			HS-2551
8	CHEMD			HS-2560
9	CHEMD			HS-2643
10	CHEMD			HS-2661
11	CHEMD			HS-2662
12	CHEMD			HS-2665
13	CHEMD			HS-2666
14	CHEMD			HS-2671
15	CHEMD			HS-2672
16	CHEMD			HS-2749
17	CHEMD			HS-2821
18	CHEMD			KS-2439A
19	CHEMD			KS-2439B
20	CHEMD			KS-2439C
21	CHEMD20_AS	D30-Acid_wash_1\	AS1	HS-3213A
22	CHEMD20_AS	D30-Acid_wash_1\	AS1	HS-3213B
23	CHEMD20_AS	D30-Acid_wash_1\	AS1	HS-3213C

Insert General Column Groups

General columns per model/process tag

2. Import mode (ImportMode)

Project name (Project)

PH comment (PHComment)

PH author (PHAuthor)

Assigned CPU (CPU)

Assigned OS (OS)

Function identifier (FID)

Location identifier (LID)

OK Cancel Help

KUVA 26. ImportMode-sarakkeen lisääminen IEA-tiedostoon.

ImportMode-sarakkeen ollessa näkyvässä otetaan koko IEA-tiedoston sisällöstä kopio, joka sijoitetaan tyhjille riveille esimerkiksi nykyisten alapuolelle. IEA-tiedostossa on nyt kaksi kertaa samat piirit, joista ylemmät tulevat toimimaan poistosuutena. ImportMode-saraketta klikkaamalla aukeaa vetolaatikko, jossa on valinnat *Delete* ja *Ignore*. Asetetaan jokin rivi poistettavaksi Deletellä ja kopioidaan se kaikkiin ylempiin poistettaviin piireihin (Kuva 27). Alemmaksi jäävät samat piirit, mutta ne tullaan generoimaan uudestaan importauksessa. Menetelmä on toimiva, koska poistettaviksi merkatut piirit ovat ohjelman suoritusjärjestyksessä ensin. Mahdollisuutena on myös jättää rivejä huomioimatta "Ignore"-asetuksella, jos niitä ei esimerkiksi haluta kirjoittaa generoinnissa ylitse. Tällöin pitää huomioida, ettei niitä poisteta tahattomasti ylempien rivien mukana. Kuvassa 27 kaksi piiriä jätettiin pois generoinnista Ignore-valinnalla ja ne poistettiin samalla Delete-osuudesta.

1	ImportMode	Project	Hierarchy	CPU	ChName	ChComment	SymbolName	ConComment	BlockName
2	IM	Pj	H1	AS		Chart			OUTOC
3						C1			S1
4	Delete	CHEMD20_AS	020-Cooking_1\	AS1	HS-2349	Compressed air to screw feeder	HS-2349_CTRL	Output value	Output
5	Delete	CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2451	On-Off valve in COO line to Heating Tank 20 T 264	HS-2451_CTRL	Output value	Output
6	Delete	CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2452	On-Off valve in COO rise line to Digester 20 T 274	HS-2452_CTRL	Output value	Output
7	Delete	CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2551	On-Off valve in COO from pump 20 P 244	HS-2551_CTRL	Output value	Output
8	Delete	CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2560	On-Off valve in COO line after Pump 20 P 338	HS-2560_CTRL	Output value	Output
9	Delete	CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2661	On-Off valve in COO line after heating tank 20 T 264	HS-2661_CTRL	Output value	Output
10	Delete	CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2662	On-Off valve in recycling line of Digester before pump 20 P 266	HS-2662_CTRL	Output value	Output
11	Delete	CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2655	On-Off valve 1 in COO line before Heater 20 E 222	HS-2655_CTRL	Output value	Output
12	Delete	CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2656	On-Off valve in COO bypass line of Heater 20 E 222	HS-2656_CTRL	Output value	Output
13	Delete	CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2672	On-Off valve in displaced line of used cooked acid from digester	HS-2672_CTRL	Output value	Output
14	Delete	CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2749	Blow valve of Digester 20 T274	HS-2749_CTRL	Output value	Output
15	Delete	CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2821	On-Off valve of Digester 20 T 274 Emergency venting	HS-2821_CTRL	Output value	Output
16	Delete	CHEMD20_AS	020-Cooking_1\	AS1	KS-2439A	On-Off valve	KS-2439A_CTRL	Output value	Output
17	Delete	CHEMD20_AS	020-Cooking_1\	AS1	KS-2439B	On-Off valve	KS-2439B_CTRL	Output value	Output
18	Delete	CHEMD20_AS	020-Cooking_1\	AS1	KS-2439C	On-Off valve	KS-2439C_CTRL	Output value	Output
19	Delete	CHEMD20_AS	030-Acid_wash_1\	AS1	HS-3213A	Sand Trap	HS-3213A_CTRL	Output value	Output
20	Delete	CHEMD20_AS	030-Acid_wash_1\	AS1	HS-3213B	Sand Trap	HS-3213B_CTRL	Output value	Output
21	Delete	CHEMD20_AS	030-Acid_wash_1\	AS1	HS-3213C	Sand Trap	HS-3213C_CTRL	Output value	Output
22		CHEMD20_AS	020-Cooking_1\	AS1	HS-2349	Compressed air to screw feeder	HS-2349_CTRL	Output value	Output
23		CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2451	On-Off valve in COO line to Heating Tank 20 T 264	HS-2451_CTRL	Output value	Output
24		CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2452	On-Off valve in COO rise line to Digester 20 T 274	HS-2452_CTRL	Output value	Output
25		CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2551	On-Off valve in COO from pump 20 P 244	HS-2551_CTRL	Output value	Output
26		CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2560	On-Off valve in COO line after Pump 20 P 338	HS-2560_CTRL	Output value	Output
27	Ignore	CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2643	On-Off valve of COO Heating Tank 20 T 264 Emergency venting	HS-2643_CTRL	Output value	Output
28		CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2661	On-Off valve in COO line after heating tank 20 T 264	HS-2661_CTRL	Output value	Output
29		CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2662	On-Off valve in recycling line of Digester before pump 20 P 266	HS-2662_CTRL	Output value	Output
30		CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2655	On-Off valve 1 in COO line before Heater 20 E 222	HS-2655_CTRL	Output value	Output
31		CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2656	On-Off valve in COO bypass line of Heater 20 E 222	HS-2656_CTRL	Output value	Output
32	Ignore	CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2671	On-Off valve 2 in COO line before Heater 20 E 222	HS-2671_CTRL	Output value	Output
33		CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2672	On-Off valve in displaced line of used cooked acid from digester	HS-2672_CTRL	Output value	Output
34		CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2749	Blow valve of Digester 20 T274	HS-2749_CTRL	Output value	Output
35		CHEMD20_AS	020-Cooking_1\020-Cooking_2\	AS1	HS-2821	On-Off valve of Digester 20 T 274 Emergency venting	HS-2821_CTRL	Output value	Output
36		CHEMD20_AS	020-Cooking_1\	AS1	KS-2439A	On-Off valve	KS-2439A_CTRL	Output value	Output
37		CHEMD20_AS	020-Cooking_1\	AS1	KS-2439B	On-Off valve	KS-2439B_CTRL	Output value	Output
38		CHEMD20_AS	020-Cooking_1\	AS1	KS-2439C	On-Off valve	KS-2439C_CTRL	Output value	Output
39		CHEMD20_AS	030-Acid_wash_1\	AS1	HS-3213A	Sand Trap	HS-3213A_CTRL	Output value	Output
40		CHEMD20_AS	030-Acid_wash_1\	AS1	HS-3213B	Sand Trap	HS-3213B_CTRL	Output value	Output
41		CHEMD20_AS	030-Acid_wash_1\	AS1	HS-3213C	Sand Trap	HS-3213C_CTRL	Output value	Output

KUVA 27. IEA-tiedoston ImportMode-sarakkeen valinnat.

4.6 Ongelmatilanteita IEA-työkalun käytössä

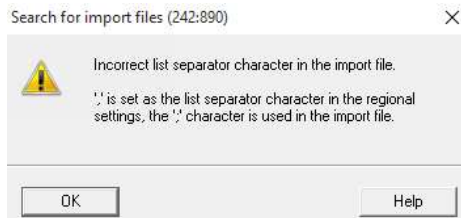
Seuraavassa luvussa käydään läpi pari ongelmatilannetta, jotka työkalun käyttämisessä ilmeni. Ensimmäisessä kohdassa ongelmia tuli Windows-käyttöjärjestelmän ja IEA-tiedoston käyttämissä listaerotusmerkeissä. Toisena ongelmakohdana oli FB-lohkojen eri versioiden väliset ristiriidat.

4.6.1 Väärä listaerotusmerkki

Process Tag Typet ja IEA-tiedostot määriteltiin ES-aseamalla, jonka jälkeen IEA-tiedosto tuotiin erilliselle tietokoneelle, koska projektissa käytettävällä ES:illä ei ollut Exceliä asennettuna. Joitakin muutoksia tiedostoon tehtiin toisen tietokoneen IEA File Editorissa, jolloin sen listaerotusmerkki vaihtui. Tilanne ei välttämättä ole kovin yleinen, mutta siinä ilmi tulleen ongelman ratkaisua voidaan hyödyntää muissakin tilanteissa.

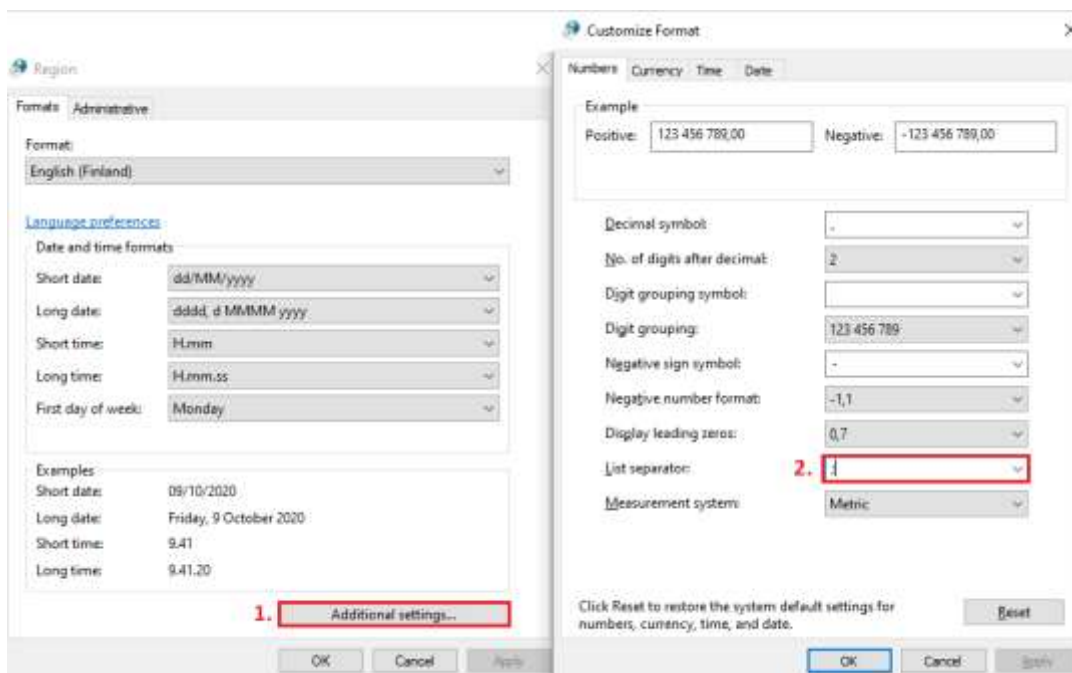
IEA-tiedoston sarakkeet erotellaan käyttäen tiettyä listaerotusmerkkiä, joten toiminnan kannalta on olennaista, että Windowsin Regional asetukset vastaavat käytettävää erotusmerkkiä. Maasta riippuen listojen erotusmerkinä käytetään usein pilkkua tai puolipistettä. Esimerkiksi Yhdysvalloissa ja Isossa-Britanniassa

käytetään erotusmerkinä pilkkua, kun taas yleisesti monessa Euroopan maassa käytetään puolipistettä. Kun IEA-tiedostoa muokattiin toisella tietokoneella, muutui samalla myös sen listaerotusmerkki. Tämä huomattiin Import-vaiheessa avautuneesta ponnahdusikkunasta, jossa sanotaan list separatorin olevan väärä eikä importtausta voida suorittaa (Kuva 28).



KUVA 28. Väärän erotusmerkin ponnahdusikkuna.

ES-tietokoneen listaerotusmerkinä oli pilkku, joka vaihdettiin puolipisteeksi vasaamaan IEA-tiedostoa ja Suomessa yleisemmin käytettävää erotusmerkkiä (Kuva 29). Tämän jälkeen importtaus toimi ongelmitta, koska ohjelma osaa lukea tiedostoa oikein. Ongelman olisi voinut korjata myös vaihtamalla IEA-tiedoston listaerotusmerkin puolipisteeksi Excelissä Find and Replace -työkalulla. Windowsin Regional-asetuksista voidaan vaihtaa myös tietokoneen käyttämä desimaalierotin.

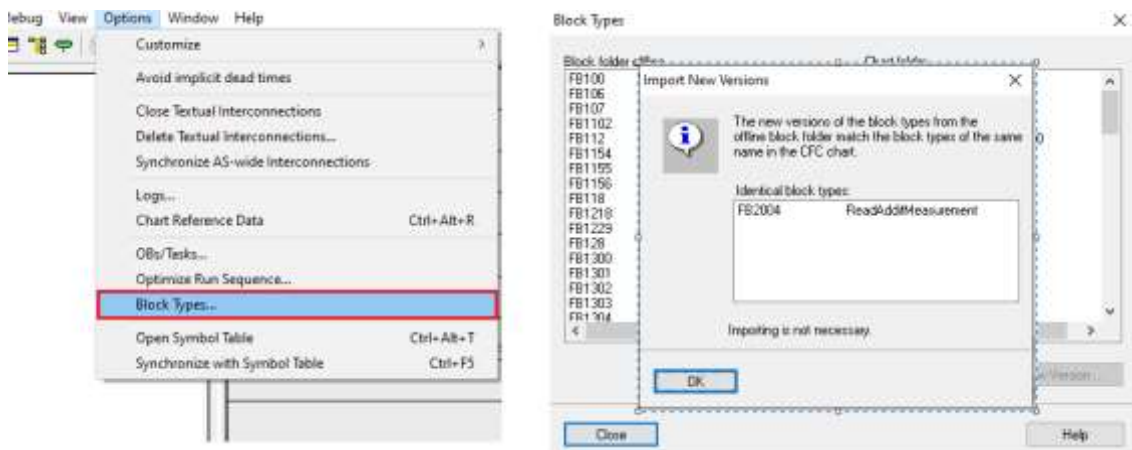


KUVA 29. Erotusmerkin valinta Windowsin Region -asetuksista.

4.6.2 Function Block -lohkojen vanhat versiot

Toisena ongelmien aiheuttajina olivat ristiriidat tyyppiin käyttämien FB (Function Block) versioiden välillä. Projektissa ohjelmointia suoritti useampi henkilö yhtäaikaan, minkä vuoksi alustavia IEA:n vaatimia työvaiheita testattiin varmuuskopioprojektin puolella. Varmuuskopioprojekti ei ollut kuin muutaman päivän vanha, mutta sinä aikana tyyppiin käyttämiin toimilohkoihin oli kerennyt tulla muutoksia.

Importtausta aloittaessa ilmestyi ponnahdusikkuna, jonka mukaan tietyn tyyppiin käyttämän FB:n Block Call ei toimi. Block Types -työkalulla tarkastamalla huomattiin toimilohkojen olevan identtiset kirjastossa ja tyyppiin käytössä olevien versioiden välillä. Block Type -työkalu saadaan avattua CFC Editorin kautta ja sillä voidaan suorittaa toimilohkoversioiden päivitys (Kuva 30).



KUVA 30. Block Types -työkalun avaaminen CFC Editorissa.

Process Tag Type- ja IEA-tiedosto olivat luotu varmuuskopioprojektin puolella, joten importauksessa ohjelma huomasi eroavaisuudet IEA-tiedoston alkuperäisen ja uuden tyyppiin FB-lohkojen välillä. Vertailemalla niitä ohjelman ja kirjaston välillä huomattiin niiden muokkauspäivämäärien olevan poikkeavat. Ratkaisuna oli kopioida sama FB varmuuskopioprojektista uuden projektin ohjelma- puolelle ja kirjastoon. Tässä täytyy varmistaa, että kopioitavaan FB-ohjelmalohkoon on tehtynä tuoreimmat muutokset eikä niitä vahingossa kirjoiteta yli.

5 YHTEENVETO

Työn tarkoituksena oli perehtyä pinnallisesti teollisuuden automaatiojärjestelmiin, tutustua Siemensin PCS 7 -järjestelmän ohjelmistoon sekä luoda yksityiskohtainen IEA-työkalun työohje ja samalla viedä automaatiojärjestelmän päivitysprojektin sovellussuunnittelua eteenpäin. Ennen työn aloittamista minulla ei ollut yhtään kokemusta PCS 7 -järjestelmistä, joten tutustuminen siihen lähti täysin puhtaalta pöydältä.

Alussa projektinäkömää SIMATIC Managerissa selaillessa kokonaisuus vaikutti hyvin monimutkaiselta ja kysymyksiä nousi enemmän kuin kerkesin sisäistää uutta tietoa. Jostakin oli aloitettava, joten kävin aluksi läpi Siemensin manuaaleja ja aloin tutustumaan IEA-työkalun käyttämiseen. Perusperiaate selvisi suhteellisen nopeasti, koska minulla oli vähän aiempaa kokemusta vastaavasta sovelluksesta Siemensin TIA Portalin puolelta. Ensimmäisiä piirejä generoitaessa PCS 7 -ohjelman rakenteesta oli vielä paljon opittavaa, enkä ollut vielä täysin tietoinen kaikkien työvaiheiden tarkoituksesta. Aloitin projektin eri prosessialueiden piirien generoinnin, jolloin IEA-työkalun käyttäminen tuli tutuksi lukuisien toistojen yhteydessä. Hiljalleen aloin oivaltamaan alussa mietityttäneitä asioita ja PCS 7 käyttäminen muuttui luontevammaksi.

IEA-työkalun suoritusohjeen koen olevan suhteellisen kattava ja uskon sen toimivan hyvänä käyttöohjeena jollekin, joka tulee tarvitsemaan sitä PCS 7 -sovellusohjelmoinnissa. Kirjoitin työvaiheet järjestelmää ennalta tuntematta, joten uskon ottaneeni huomioon asioita, joita kokenut käyttäjä olisi voinut pitää itsestään selvyytenä. Mitä PCS 7:n muuhun ohjelmistoon tulee, niin työn teoriaosuus antaa asiaan vihkiytymättömälle käyttäjälle jonkinlaisen yleiskuvan, mutta koko PCS 7 -ohjelmiston käyttämiseen kirjoittamani teoria ei tule riittämään. Piirsin samaan projektin kaikki valvomonäytöt, joten osaaminen olisi riittänyt Graphics Designerin ja WinCC:n ominaisuuksista kertovaan materiaaliin. Lisäksi olisin halunnut opetella ja kirjoittaa Control Modulesta, joka on IEA-työkalua vastaava kehittyneempi työkalu, joka tulee varmasti olemaan tulevaisuudessa olennaisessa osassa projektien ohjelmointityötä. Aikaa eikä ohjelmalisenssiä tähän perehtymiseen ei löytynyt, joten se jää myöhemmälle ajalle. Kaiken kaikkiaan sain PCS 7 -

sovellusohjelmoinnille hyvän pohjan ja tästä on helpompaa jatkaa kohti tulevaisuuden haasteita ja projekteja.

LÄHTEET

1. PCS-Engineering 2020. Palvelut. Saatavissa: <https://pcs-engineering.fi/palvelut/> Hakupäivä: 03.09.2020.
2. Realpars. What are the differences between DCS and SCADA. Verkkoartikkeli. Saatavissa: <https://realpars.com/dcs-vs-scada/> Hakupäivä:14.11.2020
3. Automaatiosuunnittelun prosessimalli 2007. Suomen Automaatioseura ry Saatavissa: https://www.automaatioseura.fi/site/assets/files/1426/automaatiosuunnittelun_prosessimalli.pdf Hakupäivä: 09.09.2020
4. Aalto, Otto - Harju, Timo – Koskinen, Kari 2018. Automaatioseura. Automaatio ennen nyt ja tulevaisuudessa. Saatavissa: https://www.automaatioseura.fi/site/assets/files/1380/automaatio_ennen_nyt_ja_tulevaisuudessa_av_artikkelisarja_2018.pdf Hakupäivä: 09.09.2020
5. Bolton, W 2006. Programmable Logic Controllers. Fourth Edition 2006. Saatavissa: [https://www.etf.ues.rs.ba/~slubura/Procesni%20racunari/Programmable%20Logic%20Controllers%204th%20Edition%20\(W%20Bolton\).pdf](https://www.etf.ues.rs.ba/~slubura/Procesni%20racunari/Programmable%20Logic%20Controllers%204th%20Edition%20(W%20Bolton).pdf) Hakupäivä: 09.09.2020
6. Siemens AG. SIMATIC Industrial Monitors and Thin Clients. Saatavissa: <https://new.siemens.com/global/en/products/automation/pc-based/simatic-ifp-itc.html#SIMATICIndustrialFlatPanels> Hakupäivä: 10.09.2020
7. Automaatio 1, ELEC-C1210, 3. Automaatiojärjestelmien rakenne. 2015. Saatavissa: https://mycourses.aalto.fi/pluginfile.php/293727/mod_resource/content/2/Automaatioj%C3%A4rjestelmien%20rakakenne.pdf Hakupäivä: 10.09.2020
8. Business for you CO. Saatavissa: <https://business4u.ae/dcs> Hakupäivä [14.11.2020](https://business4u.ae/dcs)

9. McClellan, Michael 2001. Introduction to Manufacturing Execution systems. MES Solutions Incorporated. Saatavissa: <https://www.cosyninc.com/papers/3.pdf> Hakupäivä: 16.09.2020
10. Hamill, Michael J. 2016. Industrial Communications and Control Protocols. PDH Center. Saatavissa: <https://pdhonline.com/courses/e497/e497content.pdf> Hakupäivä: 21.09.2020
11. Nieminen, Ismo 2019. Profinet prosessiteollisuuden automaatio suunnittelussa. Opinnäytetyö. Energiatekniikka. Kaakkois-Suomen ammattikorkeakoulu. Saatavissa: https://www.theseus.fi/bitstream/handle/10024/159772/Nieminen_Ismo.pdf?sequence=1&isAllowed=y Hakupäivä: 18.11.2020
12. Schneider Electric. 06/2019. OPC DA, User Guide. Saatavissa: https://download.schneider-electric.com/files?p_enDocType=User+guide&p_File_Name=EIO0000003723.00.pdf&p_Doc_Ref=EIO0000003723 Hakupäivä: 18.11.2020
13. Siemens AG 2017. Catalog. SIMATIC PCS 7 Process Control System. Saatavissa: https://www.automation.siemens.com/w2/efiles/pcs7/pdf/76/KG_STPCS7_2017-V_en_Web.pdf Hakupäivä: 25.09.2020
14. Siemens AG 2014. Industry Online Support. Kuva. Saatavissa: [https://support.industry.siemens.com/cs/document/68679830/example-blocks-for-wincc-v7-and-step-7-v5-\(for-s7-300-with-scl-and-s7-400-with-cfc-scl\)?dti=0&lc=en-BD](https://support.industry.siemens.com/cs/document/68679830/example-blocks-for-wincc-v7-and-step-7-v5-(for-s7-300-with-scl-and-s7-400-with-cfc-scl)?dti=0&lc=en-BD) Hakupäivä: 05.11.2020
15. Siemens AG 2020. Control Module (CM) Technology - Efficient Engineering in SIMATIC PCS 7. Kuva. Saatavissa: https://cache.industry.siemens.com/dl/files/748/109475748/att_977011/v3/109475748_PCS7_CMT_Engineering_DOC_PCS7V90_en.pdf Hakupäivä: 07.10.2020