

# PCS7-JÄRJESTELMÄN SOVELLUSUUNNITTELU

Toni Lakso

Opinnäytetyö  
Marraskuu 2013

Automaatiotekniikan koulutusohjelma  
Tekniikan ja liikenteen ala



JYVÄSKYLÄN AMMATTIKORKEAKOULU  
JAMK UNIVERSITY OF APPLIED SCIENCES



Tekijä(t) LAKSO, Toni	Julkaisun laji Opinnäytetyö	Päivämäärä 18.11.2013
	Sivumäärä 37	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty ( X )
Työn nimi PCS7-JÄRJESTELMÄN SOVELLUSUUNNITTELU		
Automaatiotekniikan koulutusohjelma		
Työn ohjaaja(t) RANTAPUSKA, Seppo SELOSMAA, Seppo		
Toimeksiantaja(t) JEEC Oy PEKKANEN, Jarmo		
Tiivistelmä <p>Opinnäytetyön tarkoituksena oli tehdä käytännönläheinen manuaali Siemensin PCS 7-järjestelmästä ja sovellussuunnittelusta. Tavoite oli opettaa kyseisen järjestelmän käyttöä mahdollisimman tehokkaasti aloittelevalle insinöörille tai aikaisemmin järjestelmään vihkiytymättömälle. Opinnäytetyön tilaaja oli JEEC Oy, joka toimii suunnittelu- ja konsultointitehtävissä automaatio- ja sähkötoimialoilla.</p> <p>Manuaalissa käytiin läpi sovellussuunnittelussa keskeisimmät ohjelmat/editorit läpi. Näitä olivat Simatic Manager, CFC- ja SFC-editori, WinCC-käyttöliittymä ja Import/Export assistant. Näitä ohjelmia käytetään projektin, käyttäjäohjelman kuin prosessin ohjaukseen tarvittavien prosessikuvien luomiseen. Myös Import/Export-assistentin läpikäyminen oli tärkeää, sillä kyseistä työkalua käytetään kaavioiden tekemiseen massana. Editoreiden käyttöliittymien esittelemisen lisäksi käsittelin niillä luotavia kaavioita ja kuvia, käsittelin keskeisiä käsitteitä kuten myös tärkeimpiä työvaiheita.</p> <p>Halusin tehdä manuaalin HTML-muotoon, sillä se mahdollistaa mm. kuvasarjojen käytön säästäten näin tilaa ja opettaen eri työvaiheet ja asiat selkeämmin. Toinen etu on, että manuaalin rakenne voidaan tehdä monipuolisemmaksi ja liikkuminen aiheesta toiseen on sujuvampaa kuin mitä word-tai pdf-muotoon tehdyssä manuaalissa. Aineistona käytin Siemensin manuaaleja sekä käytössäni olevaa järjestelmää ja sen helpejä. Eri työvaiheita selittäessäni käytin paljon ottamiani kuvia.</p> <p>Manuaalissa esitetyt käsitteet ja työvaiheet on esitetty selkeästi ja käytännönläheisesti, joten lukija pystyy perehtymään tiettyyn työvaiheeseen nopeasti sekä välttää myös sovellussuunnittelussa mahdollisesti tapahtuvia virheitä. Kehitettävää manuaalissa on mm. se, että se voisi käsitellä muitakin järjestelmän ja sovellussuunnittelun aihealueita.</p>		
Avainsanat (asiasanat) PCS7, Simatic Manager, CFC, SFC, WinCC, Import/Export-assistant		
Muut tiedot		



Author(s) LAKSO, Toni	Type of publication Bachelor's Thesis	Date 18.11.2013
	Pages 37	Language Finnish
		Permission for web publication ( X )
Title AUTOMATION ENGINEERING OF PCS7 SYSTEM		
Degree programme in Automation Technology		
Tutor(s) RANTAPUSKA, Seppo SELOSMAA, Seppo		
Assigned by JEEC Oy PEKKANEN, Jarmo		
Abstract <p>The purpose of this bachelor's thesis was to create a practical manual of the Siemens PCS7 system and application design. The goal was to teach the use of the system as efficiently as possible to people who have not used the system concerned. This thesis was ordered by a company, JEEC Oy which works in the field of automation and electrical designing and consulting.</p> <p>The manual goes through the most important programs/editors of the application design. These were Simatic Manager, CFC and SFC editors, WinCC and Import/Export assistant. These programs are used in creating a project, a user program and also process pictures which are used to control the process. Import/Export assistant was important also because it allows the user to create multiple diagrams at one time. The manual also discusses about diagrams and pictures, the most essential concepts and working phases.</p> <p>The purpose for making the manual in the form of HTML was to allow the use of series of pictures in order to save space and to teach different working phases more clearly. Another advantage was that the structure of the manual can be made more versatile and moving from one topic to another is easier compared to the word or pdf format. The materials used were other Siemens manuals and also the system itself and its helps. A lot of pictures were taken for explaining different working phases.</p> <p>The concepts and working phases in this manual were presented clearly and practically so the reader should be able to learn things more quickly and be able to avoid mistakes that could happen in the application design. This manual could include much more topics related to the system so that is one thing that could be improved.</p>		
Keywords PCS7, Simatic Manager, CFC, SFC, WinCC, Import/Export-assistant		
Miscellaneous		

## Sisältö

<b>1</b>	<b>Johdanto</b> .....	<b>5</b>
1.1	Opinnäytetyön tilaaja.....	5
1.2	Tarve ja tavoite .....	5
1.3	Henkilökohtaiset tavoitteet.....	5
1.4	Tehtäväkentän rajaus.....	5
<b>2</b>	<b>Tietoperusta</b> .....	<b>6</b>
2.1	PCS7-järjestelmä .....	6
2.1.1	Engineering station(ES) .....	7
2.1.2	Operator station(OS).....	7
2.1.3	Automation station(AS).....	8
2.2	Simatic Manager .....	9
2.3	CFC-editori .....	10
2.4	SFC.....	13
2.5	WinCC.....	14
2.6	Import/Export-assistant .....	16
<b>3</b>	<b>Toteutus</b> .....	<b>19</b>
3.1	Työkalut.....	19
3.1.1	Gimp.....	19
3.1.2	Programmer's notepad .....	20
3.1.3	JQuery .....	21
3.2	Työvaiheet .....	21
3.2.1	Rakenteen suunnittelu .....	21
3.2.2	Aineiston keruu.....	22
3.2.3	HTML, JQuery ja CSS .....	22

3.2.4	Käyttöliittymän luominen.....	22
<b>4</b>	<b>Sovellussuunnittelun ohjeistus .....</b>	<b>23</b>
4.1.1	Multiprojektin luominen .....	23
4.1.2	Kaavion kääntäminen lohkotyypiksi .....	24
4.1.3	Toimintojen lisääminen askeleisiin.....	25
4.1.4	WinCC-värit.....	26
4.1.5	Process tag typen luominen .....	27
<b>5</b>	<b>Tulokset .....</b>	<b>29</b>
<b>6</b>	<b>Pohdinnat .....</b>	<b>30</b>
	<b>Lähteet .....</b>	<b>32</b>
	<b>Liitteet .....</b>	<b>34</b>
	Liite 1. Lohkon parametointi .....	34
	Liite 2. Operointilohkot .....	35
	Liite 3. Instanssin luominen .....	36
	Liite 4. Import-filen luominen .....	37

## KUVIOT

KUVIO 1. Esimerkki PCS7-järjestelmästä peruskomponentteineen .....	7
KUVIO 2. Avattu multiproject Simatic Managerissa .....	9
KUVIO 3. CFC-editori .....	10
KUVIO 4. Analogiamittaus yksinkertaisimmillaan.....	11
KUVIO 5. Dataliikenne CFC:n, käyttöohjelman ja AS:n välillä .....	12
KUVIO 6. SFC-editori.....	13
KUVIO 7. WinCC:n ja AS:n välinen kommunikaatio .....	15
KUVIO 8. WinCC Explorer .....	16
KUVIO 9. Näkymä Import/Export-toolista.....	17
KUVIO 10. Gimp-kuvankäsittelyohjelma .....	19
KUVIO 11. HTML-sivu avattuna Programmer's notepadissa.....	20
KUVIO 12. HTML-manuaalin käyttöliittymä Opera-selaimessa avattuna .....	23
KUVIO 13. Ote aihealueesta multiprojektin luominen .....	24
KUVIO 14. Kaavion kääntäminen lohkotyypiksi.....	25
KUVIO 15. Toiminnon lisääminen .....	26
KUVIO 16. Palettivärien käytön havainnollistaminen .....	27
KUVIO 17. Process tag typen luominen .....	28

**Sanasto**

Simatic Manager	Pääohjelma, jolla hallitaan projektia ja josta pääsee käsiksi eri editoreihin.
WinCC	Käyttöliittymä, jolla operaattori ohjaa prosessia ja joka sisältää kaikki tarvittavat näytöntekotyökalut.
CFC	Continuous Function Chart. Editori, jolla tehdään CFC-kaavioita ja mahdollisesti koko ohjelma CPU:lle.
SFC	Sequential Function Chart. Editori, jolla luodaan sekvenssejä.
ES	Engineering station. Sisältää kaikki järjestelmätyökalut ja datan, joka sitten käännetään ja ladataan eri asemille.
OS	Operator station/server. Asemalla ohjataan ja monitoroidaan prosessia.
AS	Automation station. Sisältää käyttäjäohjelman. Prosessori dataa ja lähettää sitä niin kenttälaitteille kuin muille asemille.
IEA	Import/Export assistant. Sillä luodaan kaavioista mallipohjia, joista taas generoidaan kaavioita massana.
Process tag type	Kaaviosta tehty mallipohja.

# 1 Johdanto

## 1.1 Opinnäytetyön tilaaja

Opinnäytetyön tilasi JEEC Oy, joka toimii suunnittelu- ja konsultointitehtävissä automaatio- ja sähkötoimialoilla. Ydinosaaminen on automaatioissa ja prosessiautomaatiojärjestelmien suunnittelussa ja käyttöönnotossa. (JEEC Oy, 2013.) Pääpaino suunnittelussa on Metso DNA-järjestelmän käyttö. Tällä hetkellä yritys työllistää 14 henkilöä.

## 1.2 Tarve ja tavoite

Yritys pystyy tarjoamaan laajaa osaamista eri automaatiojärjestelmiin liittyen. Oppimisen nopeuttamiseksi yritys haluaa kehittää menetelmiä, joilla oppimiskynnystä saataisiin matalammaksi. Työharjoittelujaksolla pääsin tutustumaan Siemensin PCS7 automaatiojärjestelmään suunnittelun alusta FAT-testausvaiheeseen asti. Sen vuoksi luonnollisena jatkona opinnäytetyöni aiheeksi tehtäväkseni annettiin tehdä oppimista varten materiaali, jolla uusi suunnittelija voidaan helpommin saada tutustumaan järjestelmään. Opinnäytetyön toteutusmenetelmän sain valita itse.

Syy miksi päätin esittää opetusmateriaalin HTML-toteutuksena, oli se, että se mahdollistaa helpomman tavan liikkua eri aiheesta toiseen kuin mitä normaalin word- tai pdf-tiedoston selaaminen tekisi. Se mahdollistaa myös erilaisten toiminnallisuuksien luomisen kuten kuvasarjojen käytön. Kuvasarja on hyvä ja tiivis tapa opettaa jokin tietty toimintatapa/tehtävä.

## 1.3 Henkilökohtaiset tavoitteet

Pääsin tutustumaan PCS 7-järjestelmään harjoittelun aikana aina projektin sovellussuunnittelun päätökseen asti. Opinnäytetyö on hyvä lisä kasvattamaan omaa osaamistani ja tietämystäni kyseisestä järjestelmästä ja sen eri työkaluista. Jatkossa pystyn paremmin käyttämään eri työkaluja sovellussuunnittelussa ja antamaan myös neuvoa toisille.

## 1.4 Tehtäväkentän rajaus

Järjestelmän laajuudesta johtuen, manuaalin teko koko järjestelmästä on mahdoton tehtävä opinnäytetyölle varatun ajan puitteissa. Siksi aihealueet rajattiin koske-



maan ydinalueita, eikä aiheita käsitellä kovin syvällisesti. Tarkoituksena on opettaa asiat käytännönläheisesti ja nopeasti.

Aihealueet rajattiin karkeasti ottaen seuraavasti: Simatic managerin käyttöliittymän selittäminen; CFC-editori, lohkojen ja kaavioiden luonti; SFC eli sekvenssinohjausjärjestelmä; import/export-toolin käyttö; WinCC eli käyttöliittymä prosessikuvien tekemiseen ja prosessin valvontaan.

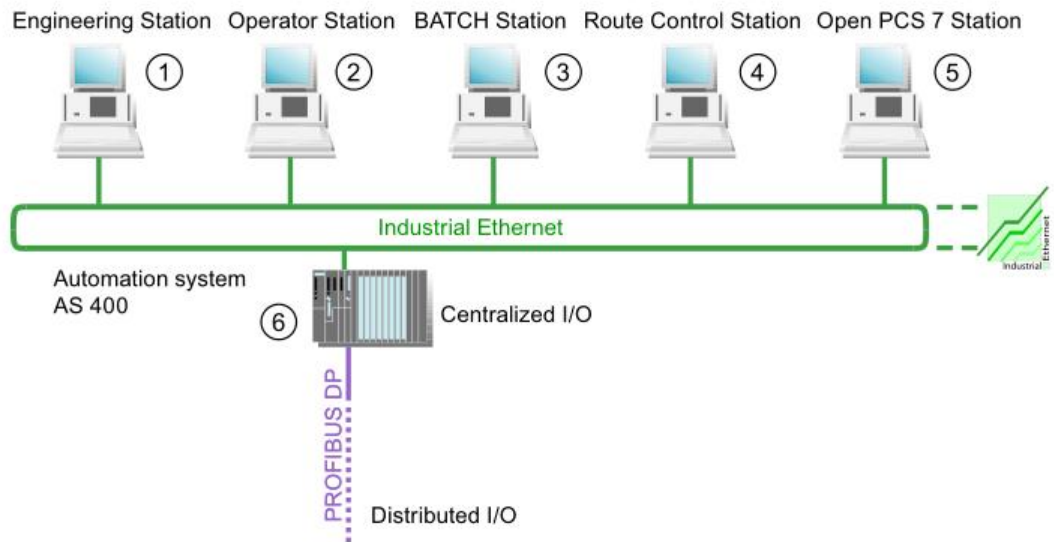
## **2 Tietoperusta**

### **2.1 PCS7-järjestelmä**

PCS7 on Siemensin tarjoama prosessinohjausjärjestelmä. Se sisältää hyvin monipuoliset työkalut, jolla haluttu järjestelmäkokonaisuus voitaisiin luoda. Se on myös hyvin laajennettavissa oleva järjestelmä mahdollistaen hyvin monipuolisia ja rakenteeltaan erilaisia järjestelmäkokonaisuuksia.

Järjestelmän rakenne voi poiketa tapauskohtaisesti hyvin paljon, mutta pääasiassa järjestelmä koostuu asemista kuten ES, OS, OS clients, Batch Station ym., väylistä kuten terminal bus-, plant bus- ja field bus-väylästä, IO-kaapeista ja kenttälaitteista.

Serverit ja clientit keskustelevat keskenään terminal bus-väylän välityksellä, serverit ja AS(PLC/logiikka) plant bus-väylän kautta ja AS keskustelee kenttälaitteiden kanssa kenttäväylän(field bus) välityksellä.(PCS 7 Engineering System, 21-22.) Alla oleva kuvio selventää järjestelmän rakennetta.



KUVIO 1. Esimerkki PCS7-järjestelmästä peruskomponentteineen. Eri asemat ovat yhteydessä toisiinsa ja AS:ään Industrial Ethernetin, joka toimii plant bus-väylänä, välityksellä. AS on taas yhteydessä kentälaitteisiin Profibus DP-väylän kautta.(PCS 7 Engineering system, 125.)

Ohjelmistopuolella järjestelmä on seuraavanlainen: ES sisältää järjestelmätyökalujen lisäksi projektikohtaisen datan, joka sitten ladataan eri asemille, OS sisältää prosessikuvat, joiden kautta operaattori ohjaa ja valvoo järjestelmää ja AS sisältää käyttäjäohjelman, jotta prosessin ohjaus on ylipäätään mahdollinen.

### 2.1.1 Engineering station(ES)

ES on keskustietokone, joka sisältää kaikki PCS7-järjestelmän työkalut. Sillä mm. konfiguroidaan järjestelmässä käytettävät laitteet ja niiden väliset yhteydet, luodaan käyttäjäohjelma(user program), joka käännetään ja ladataan AS:lle. Siellä luodaan myös laitoshierarkia, jolla on suora vaikutus siihen, miten eri alueet esitetään OS:llä. Myös käyttäjäohjelmaan luotujen moottori- ym. lohkojen lohkoikonit generoidaan OS:n prosessikuviin.

### 2.1.2 Operator station(OS)

Operaatioasemalla monitoroidaan ja ohjataan PCS7-järjestelmää prosessikuvien kautta. Faceplatet antavat tiedon yksittäisten komponenttien tilasta ja tekniikkafunktiosta. Se myös antaa muuta tietoa, mitä tarvitaan operaatio-ohjaukseen ja monito-

rointiin liittyen, kuten trendien, viesti ja hälytyslistojen ja arkistointitietojen muodossa.(PCS7 Operator Station, 17.)

Operaatioasema voi olla yksittäisasemana, jolloin kaikki monitorointi ja ohjaukset tapahtuvat sen kautta tai se voi toimia moniasemajärjestelmänä, jolloin se toimii serverinä ja sillä on useampia OS klienteja.

Moniasemajärjestelmässä OS-serveri välittää OS clienteleille dataa kuten projektidataa, prosessiarvoja, arkistointitietoja, hälytyksiä ja viestejä(PCS7 Operator Station, 19).

OS:lle tehdään kaikki prosessikuvat, joiden välityksellä sitten prosessia ohjataan ja monitoroidaan. Prosessikuvat tehdään WinCC:n Graphics Designer-työkalulla. OS kuviin myös generoidaan CFC-kaavioiden lohkojen lohkoikonit kuten moottori-, mitaus- ja säädinlohkot kuten myös SFC-kaavioiden sekvenssit OS käännöksen yhteydessä. Kyseiset lohkoikonit sisältävät valmiit faceplatet, joiden kautta prosessia voidaan ohjata ja monitoroida.

### **2.1.3 Automation station(AS)**

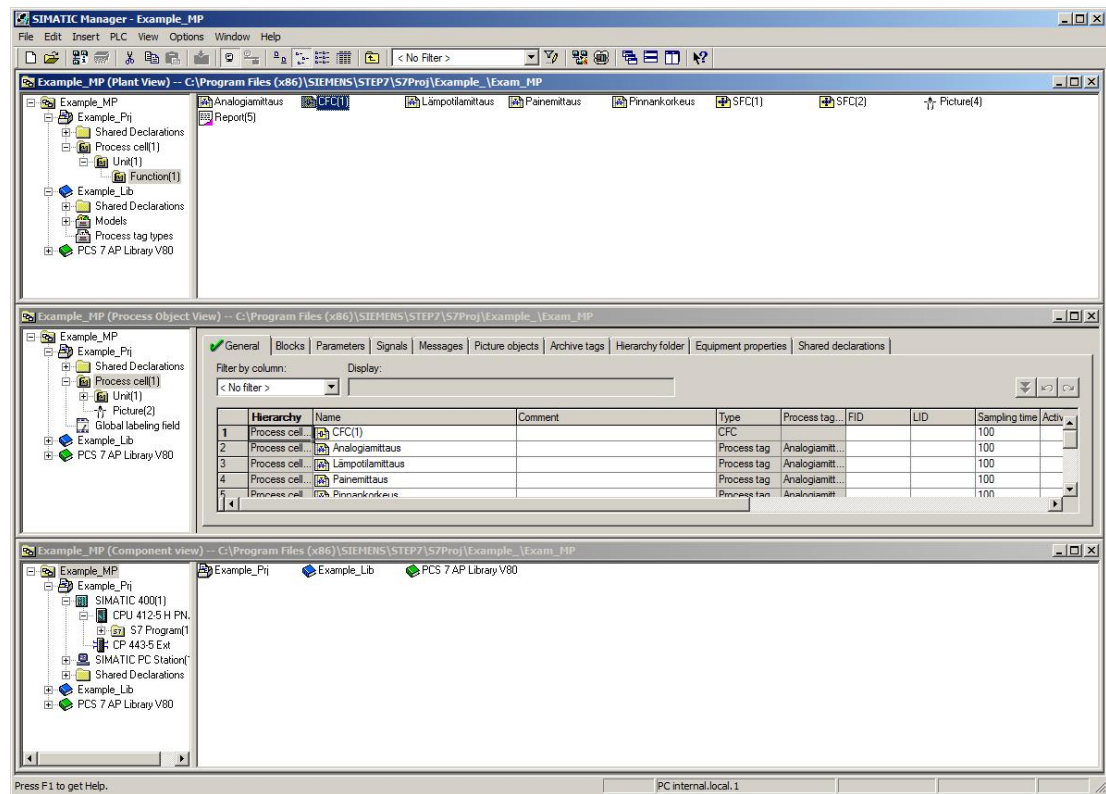
PCS7-järjestelmässä automaatioasemana toimii S7-400 sarjan logiikat. Automaatioasemaan voidaan viitata yleisesti käyttämällä nimityksiä CPU tai logiikka. Se on yhteydessä kenttälaitteisiin ja se prosessoi dataa ja lähettää tietoja OS:lle kuten myös ohjaussignaaleja takaisin kenttälaitteisiin.

AS hoitaa seuraavia tehtäviä: prosessimuuttujien hankkimisen, datan prosessoinnin käyttäjäohjelman ohjeiden mukaisesti, ohjaustietojen ja asetusarvojen lähettämisen prosessiin, datan toimittamisen OS:lle visualisointiin, OS:ltä tulevien operaattorin komentojen havaitseminen ja että ne ovat saapuneet prosessiin sekä suora yhteys reittiohjausasemaan.(PCS 7 Engineering System, 22.)

AS sisältää käyttäjäohjelman, jolla koko prosessia ohjataan tai osaprosessia riippuen kuinka monta AS:ää tehdas sisältää. Käyttäjäohjelma ladataan AS:lle ES:ltä, ja se sisältää mm. käyttäjän tehdyt CFC- ja SFC-kaaviot, FB:t, DB:t ym. datan.

## 2.2 Simatic Manager

Simatic Manager on paikka, josta pääsee eri Step 7-sovelluksiin/editoreihin. Sitä käytetään projektin hallintaan, arkistointiin ja dokumentointiin. (PCS 7 Engineering System, 197.)



KUVIO 2. Avattu multiproject Simatic Managerissa. Sama projekti kolmessa tärkeimmässä näkymässä: plant view, jossa lisätään ja poistetaan objekteja sekä luodaan hierarkiajärjestys; process object view, joka esittää objektit parametreineen taulukkomuodossa helppoa editointia varten; component view, jossa tehdään hardware konfiguraatio.

Simatic Managerissa luodaan uusi projekti/multiprojekti. Multiprojekti sisältää nimensä mukaan useampia projekteja, joista yksi voi olla esim. AS ja toinen OS. Managerissa luodaan laitoshierarkia eli esim. yläkansiona on tehdas, sen alla on eri tehdasalueet ja niiden alla taas omat alialueensa. Kaikestaan hierarkiatasoja voi olla 8. Simatic Managerissa myös luodaan eri objekteja. Ne voi olla muita projekteja, kuvia, kaavioita, sekvenssejä, asemia ja muita eri laitteita. Eri objektit lisätään oikeisiin paikkoihin niin, että esim. kuvat ja niihin liittyvät kaaviot ja sekvenssit ovat vastaavissa kansiossa, jotta OS-käännöksen yhteydessä lohko-ikonit generoituisivat oikeisiin kuviin.

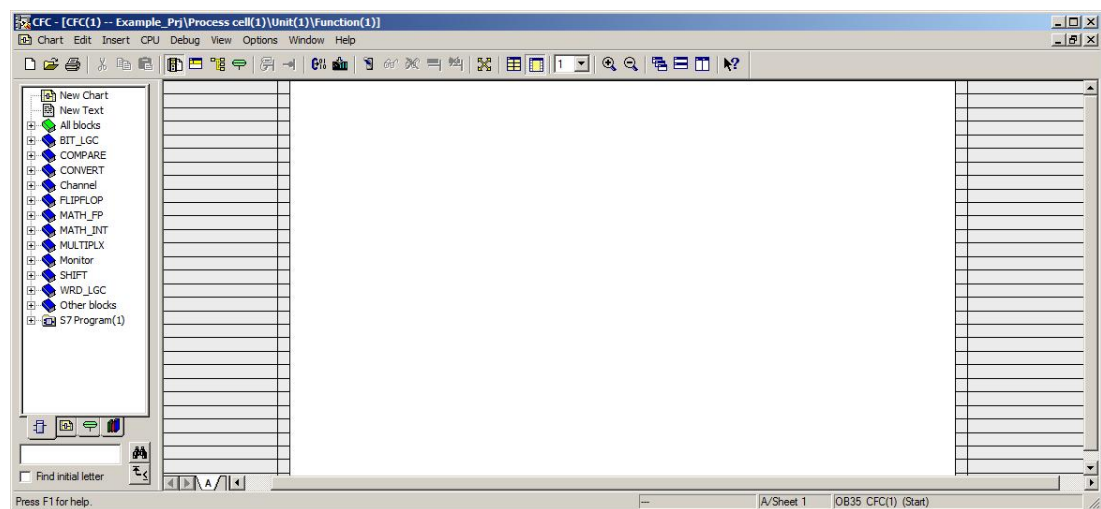
Manageri on myös keskuspaikka, josta pääsee eri editoreihin avaamalla halutun objektin, kuvan tapauksessa avautuu WinCC ja CFC-kaavion tapauksessa CFC-editori. Se sisältää myös muita tärkeitä työkaluja, joita tärkeimpänä Import/Export-assistant, jolla voidaan generoida kaavioita massana mallikaavion pohjalta.

Managerin kautta hoituu myös eri asemien konfigurointi, yhteysasetusten tekeminen kuin myös ohjelman kääntäminen ja lataaminen kyseisille asemille.

## 2.3 CFC-editori

CFC(Continuous Function Chart) on graafinen editori, jolla voidaan luoda koko ohjelma CPU:lle käyttäen valmiiksi konfiguroituja lohkoja(PCS 7 CFC for SIMATIC S7, 11).

Editorin saa auki joko luomalla ensin uuden kaavion Simatic Managerissa ja avaamalla sen tai sitten Windowsin käynnistä-valikosta Siemensin Step 7-työkaluista.



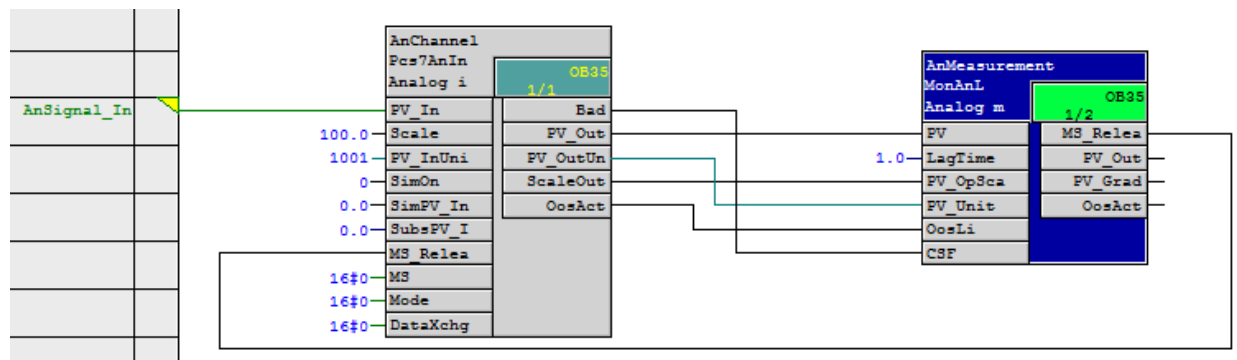
KUVIO 3. CFC-editori. Aloitusnäkyminä on tyhjä CFC-kaavio. Yläpuolella on editorin menuvalikot ja työkalurivi. Vasemmalla puolella on kirjastoikkuna(catalog), josta raahaamalla tuodaan lohkoja työtilaan. Työtilan vasemmalla puolella on tulot ja oikealla lähdöt.

Kyseisellä editorilla luodaan kaavioita(charts), jotka käännetään ja ladataan järjestelmään. Valmiista kaavioista voi myös tehdä lohkotyyppejä, joita edelleen voidaan käyttää muissa kaavioissa. Editori mahdollistaa myös kaavioiden sijoittamisen kaavioiden sisään, luoden näin selkeämpiä ohjelmakokonaisuuksia.

Kaavioiden toiminnallisuus rakennetaan lohkoista, esimerkkilohkoina voidaan mainita moottorilohko(MotL) ja analogiamittauslohko(MonAnL). Kyseiset lohkot/lohkotyytit löytyvät editorista siellä olevasta kirjastoikkunasta. Myös omia lohkoja voi tehdä ja

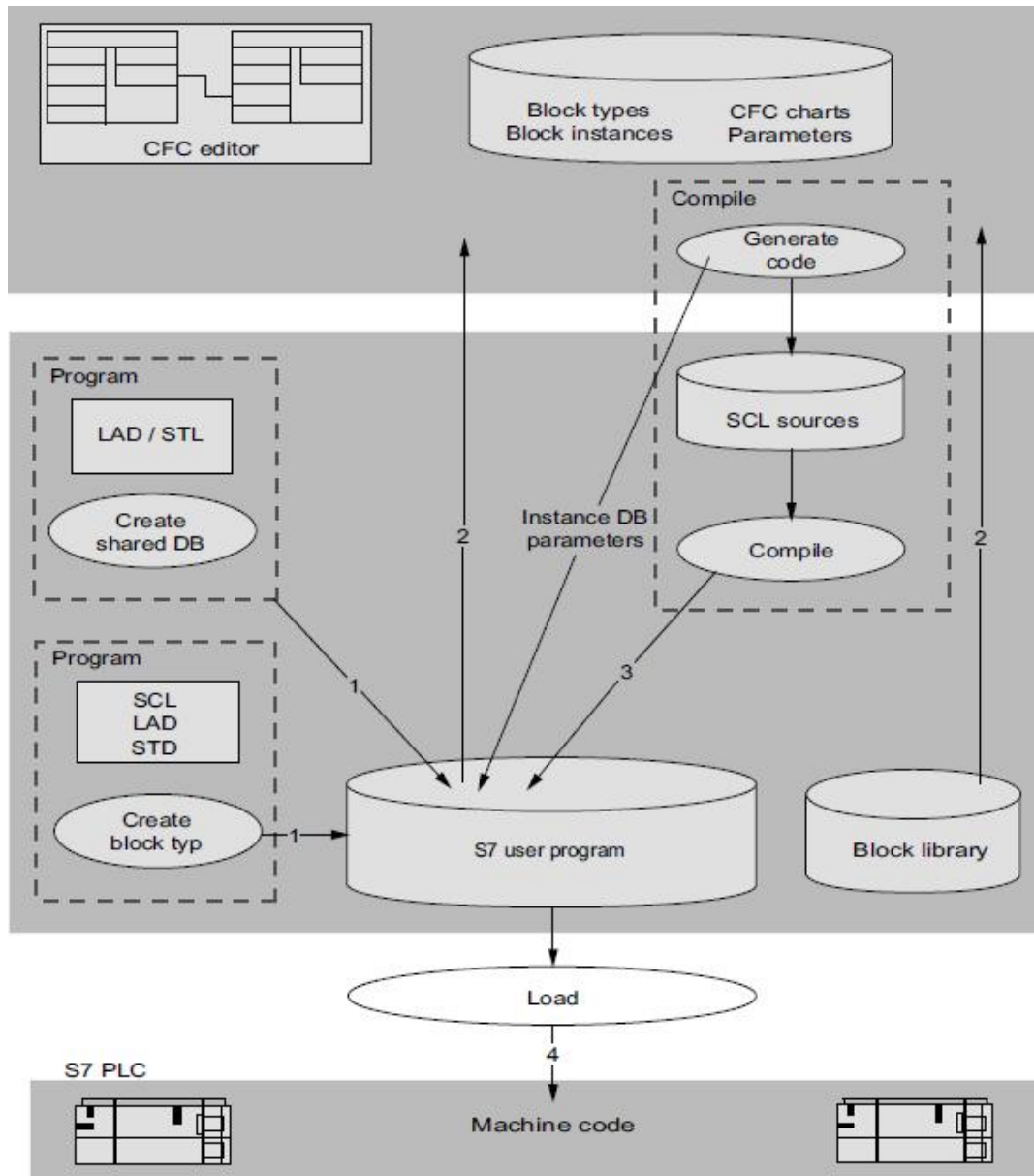
liittää edelleen kaavioihin käytettäviksi. Nämä lohkot parametroidaan ja kytketään toisiinsa muodostaen loogisia kokonaisuuksia.

Kaaviot voivat olla kooltaan varsin suuria. Kaavio koostuu arkeista/sivuista, jolle mahtuu useita eri lohkoja. Näitä sivuja on yhteensä 6 kpl:ta. Kaaviossa voi olla myös maksimillaan 26 osiota, joista kukin sisältää 6 arkkia, joten tila ei tule loppumaan. Käytännössä tehdyt kaaviot ovat toiminnoiltaan yksinkertaisia kokonaisuuksia, kuten moottorinohjaukseen tai mittaukseen tarkoitettu, eikä ne vie paljon tilaa. Alla oleva kuvio on kaaviosta, joka sisältää vain kaksi lohkoa.



KUVIO 4. Analogiamittaus yksinkertaisimmillaan. Kanavalohko lukee signaalia prosessiasemalta, josta signaali viedään edelleen mittauslohkoon. Kyseinen mittaus näkyy valvomossa mittauslohkosta generoidulla lohkoikonin faceplatella.

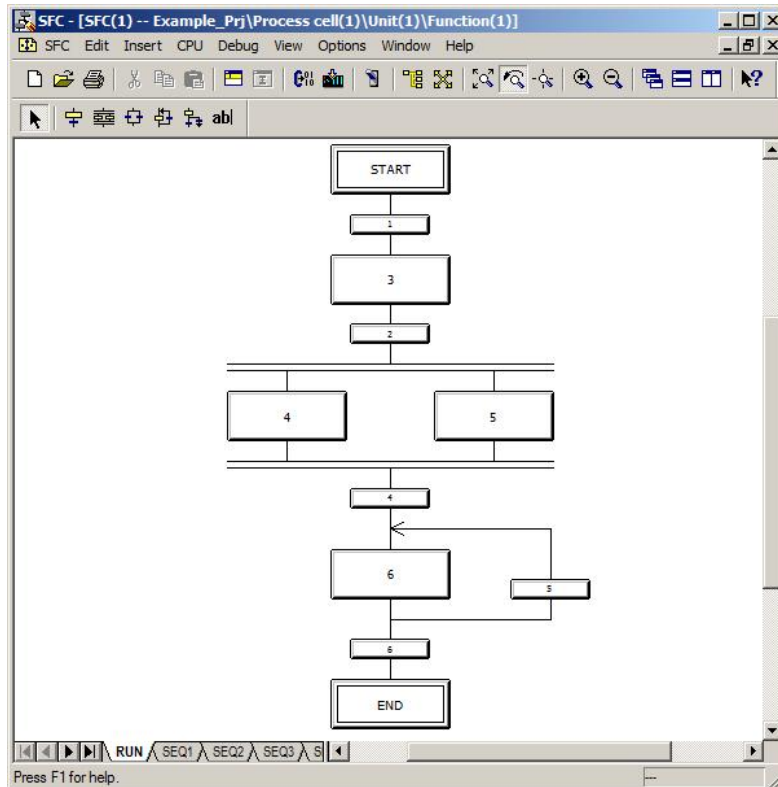
Valmiit kaaviot voidaan edelleen kääntää(compile) konekielelle, jonka jälkeen ne ladataan(download) PLC:lle(ks. kuvio5.).



KUVIO 5. Dataliikenne CFC:n, käyttöohjelman ja AS:n välillä. Yllä on CFC-editori, jolla kaaviot luodaan. Käyttöohjelma keskellä sisältää projektin kaiken datan ja josta CFC-editori saa käyttöönsä lohkotyyppejä joko lohkokirjastosta tai käyttöohjelmaan itse tehdyistä lohkotyypeistä. Lohkotyyppejä voi luoda kyseisellä editorilla tai muilla Siemensin Step 7-työkaluilla. Käännösvaiheessa kaaviot käännetään käyttöohjelmaan, josta ne voi edelleen ladata PLC:hen. (PCS 7 CFC for SIMATIC S7, 16.)

## 2.4 SFC

SFC(Sequential Function Chart) on käyttöliittymä, joka mahdollistaa sekvenssiohjausjärjestelmien luomisen(PCS 7 SFC for SIMATIC S7, 11).



KUVIO 6. SFC-editori. Käyttöliittymän ulkoasu on samantapainen kuin CFC-editorin menuvalikoiden ja ikonien osalta. Avattuna tyhjä sekvenssi, jossa havainnollistettu sekvenssin rakennetta.

SFC-editorilla luodaan sekvenssikaavioita, jotka voivat itsessään sisältää useampia eri sekvenssejä. Jokainen sekvenssi vie yhden osion, osioiden maksimimäärän ollessa normaalikaavioiden tapauksessa 8 ja sekvenssityyppien 32. Kyseisen sekvenssin päälle meneminen voi liittyä johonkin tapahtumaan/tapahtumiin eli vaikka moottorin käyntitilaan, tai sekvenssin omaan tilaan liittyen, esim. operaattori on pistänyt sekvenssin hold-tilaan.

Sekvenssin toiminta on varsin yksinkertainen. Se toimii askelperustaisesti eli askeleessa suoritetaan sille määrätty toimenpiteet, kuten joidenkin kaavioiden lohkojen parametrien pistäminen tiettyyn tilaan/arvoon. Askeleesta siirrytään seuraavaan, kun toimenpiteet on suoritettu ja siirtoehdot ovat toteutuneet eli tietyt parametrit ovat halutuissa arvoissa.

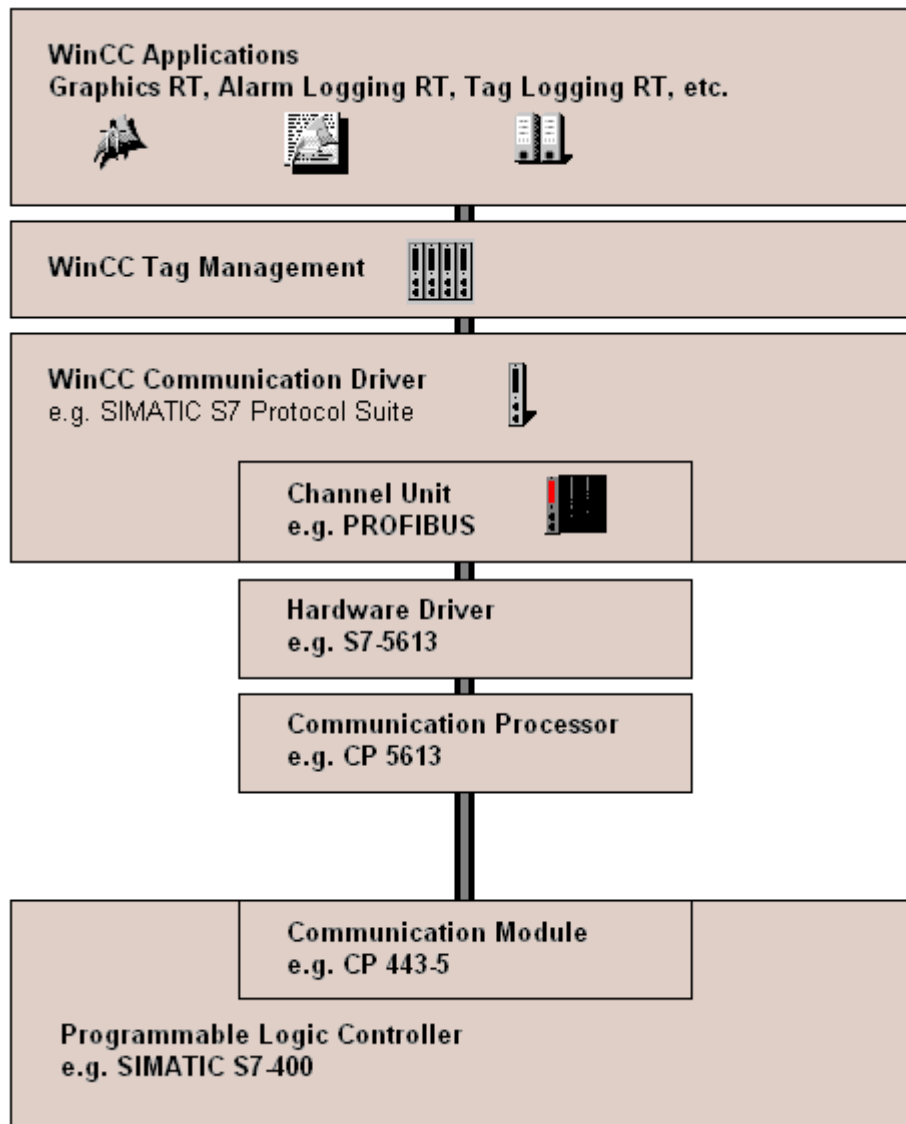


Sekvenssin rakenne taas voi muodostua normaaleista askeleista, rinnakkaisista tai vaihtoehtoisista haaroista, hyppyistä, luupeista ja tietenkin aloitus ja lopetusaskeleesta.

Valmiin sekvenssin voi kääntää ja ladata PLC:lle aivan samalla tavalla kuin CFC-kaavionkin.

## **2.5 WinCC**

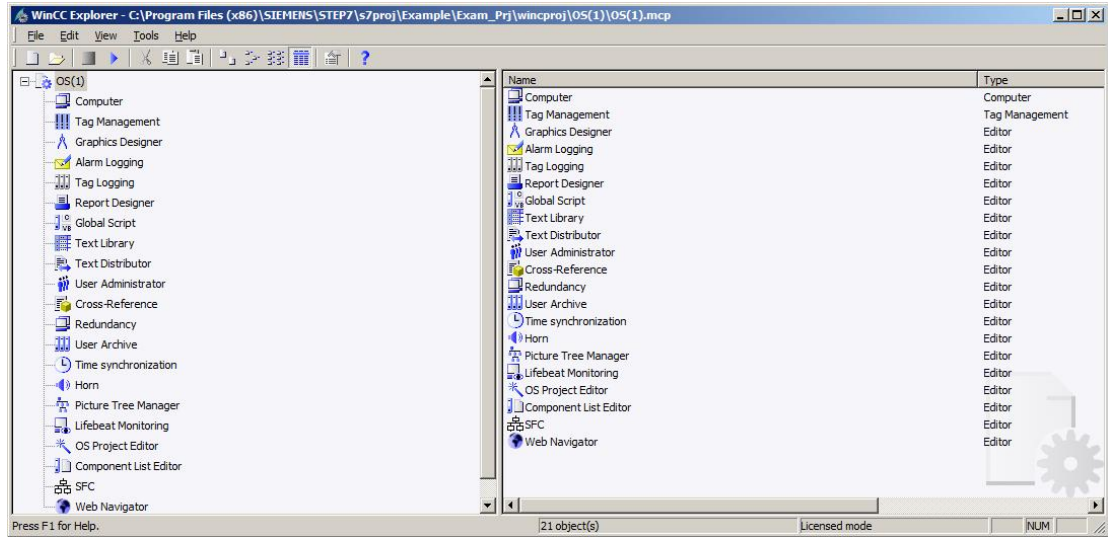
WinCC on käyttöliittymä, jolla operaattori on yhteydessä logiikkaan ja sitä kautta prosessiin valvoen ja operoiden sitä. Yhteys WinCC-serverin ja logiikan välillä tapahtuu tagien välityksellä. Tagit generoituvat WinCC:hen OS-käännöksen yhteydessä, kaavioiden sisältämien lohkojen parametreista. Yhteys OS:n ja AS:n välille konfiguroidaan Simatic Managerin puolella. Alla oleva kuvio selittää eri asemien välistä kommunikaatiota ja laitepuolta.



KUVIO 7. WinCC:n ja AS:n välinen kommunikaatio. Viestintä asemien välillä toimii tageilla tag managementin kautta. Väylästä riippumatta prosessiasema tarvitsee tietyn kommunikaatiomodulin, jota vastaava ajuri löytyy WinCC:n puolelta, jotta tiedonsiirto olisi mahdollinen. (WinCC Explorer help, 14.)

WinCC on laaja työkalu. Se itse sisältää monia eri editoreita, joista merkittävimpiä on Graphics Designer, jolla näyttökuvat luodaan. Näyttökuvuihin generoituu kaavioihin lisätyistä lohkoista lohkoikonit OS-käännöksen yhteydessä. Tämän jälkeen ikonit voidaan sijoittaa kuvassa oikeille paikoilleen. Ikonit ovat valmiita kokonaisuuksia tageineen. Esimerkki-ikoneita ovat moottori-, säädin- ja mittaus-ikonit valmiine faceplateineen, joilla voi runtimessa operaattori valvoa ja operoida lohkojen ja sitä kautta laitteiden toimintaa. Lisäksi kuviin voi tehdä muuta toimintaa käyttämällä esim. C- ja VBS-ohjelmointikieliä.

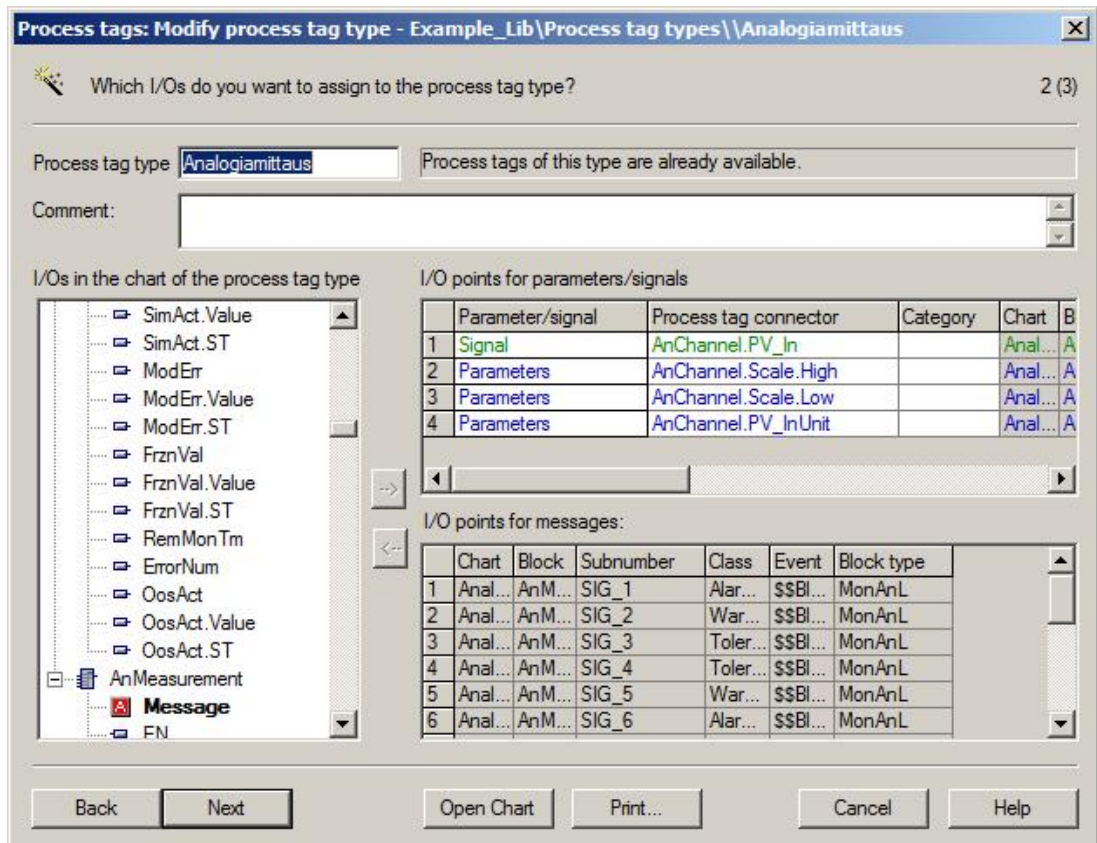
Kuvien lisäksi WinCC:ssä voi tehdä runtime-asetuksia, muokata esim. näyttöhierarkia, ottamalla kantaa sekvenssien visuaaliseen ulkonäköön, muokata hälytyksiä ja kieliasetuksia ym.



KUVIO 8. WinCC Explorer. Sisältää useita eri editoreita, joista esim. Graphics Designerissä tehdään projektiin liittyvät prosessikuvat.

## 2.6 Import/Export-assistant

Monesti on tilanteita, joissa voidaan käyttää samoja kaaviota suorittamaan samantyyppisiä toimintoja, helppona esimerkkinä mittauskaaviot, joissa toiminta on sama, mutta mitattava suure, rajat, hälytykset, ym. parametrit poikkeavat toisistaan. Tällaisissa tilanteissa kannattaa luoda kaavio, jota sitten käyttää mallipohjana muiden kaavioiden teossa.



KUVIO 9. Näkymä Import/Export-toolista. Toolilla luodaan/muokataan olemassa olevaa process tag typeä. Kyseisessä kuvassa määritetään parametrit, signaalit ja viestit, jotka otetaan mukaan process tag typeen ja sitä kautta tulevaan import-tiedostoon.

Import/Export-assistentilla kaaviosta luodaan mallipohja, toiselta nimeltään process tag type. Mallipohjaa luotaessa määritellään mitä parametreja otetaan mukaan importoitavaan Excel- tai IEA-tiedostoon. Mukaan kannattaa ottaa parametreja, jotka tulevat poikkeamaan eri kaavioiden kesken. Mallipohjana käytettävän kaavion muut parametrit kannattaa parametroida oikein, sillä ne ovat oletuksena samat kaikille generoiduille kaavioille. Syy, miksi mallipohjaan otetaan mukaan vain tarvittavat parametrit kaikkien sijaan, on se, että jo yksi lohko saattaa sisältää useita satoja parametreja eikä niitä ole järkevää ottaa mukaan luotavaan import-tiedostoon.

Kun kaaviosta on luotu process tag type, voidaan sen pohjalta luoda import-tiedosto. Se on tiedostotyyppistä riippumatta taulukkomuotoinen tiedosto, johon kirjoitetaan generoitavien kaavioiden eli toiselta nimeltään process tags tiedot. Kyseinen tiedosto määrätään process tag typeelle käytettäväksi, jotta uusien kaavioiden generoiminen olisi mahdollista.

Itse generoiminen tapahtuu Import-toiminnolla. Tuloksena luodaan uudet kaaviot omine tietoineen paikkaan, minne ne on määrätty generoitaviksi. Ne säilyttävät kuitenkin tietyn yhteyden mallipohjaan, josta ne on luotu. Mallipohjaan tehdyt muutokset eivät kuitenkaan päivity siitä luotuihin process tageihin.

Import/Export-assistantilla on myös export-toiminto, jolla voidaan ottaa talteen tiedot generoiduista process tageista, jos halutaan tehdä muutoksia tai jostain syystä alkuperäinen import-tiedosto on poistettu.

Mikäli esim. havaitaan, että generoidut kaaviot ovat toiminnoiltaan virheellisiä, voidaan niistä tiedot ottaa talteen export-komennolla, jolloin niistä luodaan vastaavanlainen tiedosto kuin import-tiedosto ja sitä voidaankin käyttää siihen tarkoitukseen. Nyt voidaan esim. tehdä tarvittavat muutokset mallipohjaan, määrittää sille talteen otettu export-tiedosto import-tiedostoksi ja tehdä siihen mahdolliset muutokset ja generoida kaaviot uudelleen. Näin säästetään se vaiva, että kaikki parametri-tiedot pitäisi syöttää käsin taulukkoon uudelleen tai että kävisi kädestä pitäen muuttamassa jokainen aikaisempi generoitu kaavio toiminnaltaan oikeaksi.

IEA-työkalulla voidaan myös tehdä mallipohjia suuremmista kokonaisuuksista kuten tietyistä prosessialueista, mikäli niiden toiminta on samantapainen. Tällaisia mallipohjia kutsutaan nimeltä models. Työkalu tarjoaa siis tällaisenkin toiminnon, vaikka sellaisen käytölle harvoin tulee tarvetta.

## 3 Toteutus

### 3.1 Työkalut

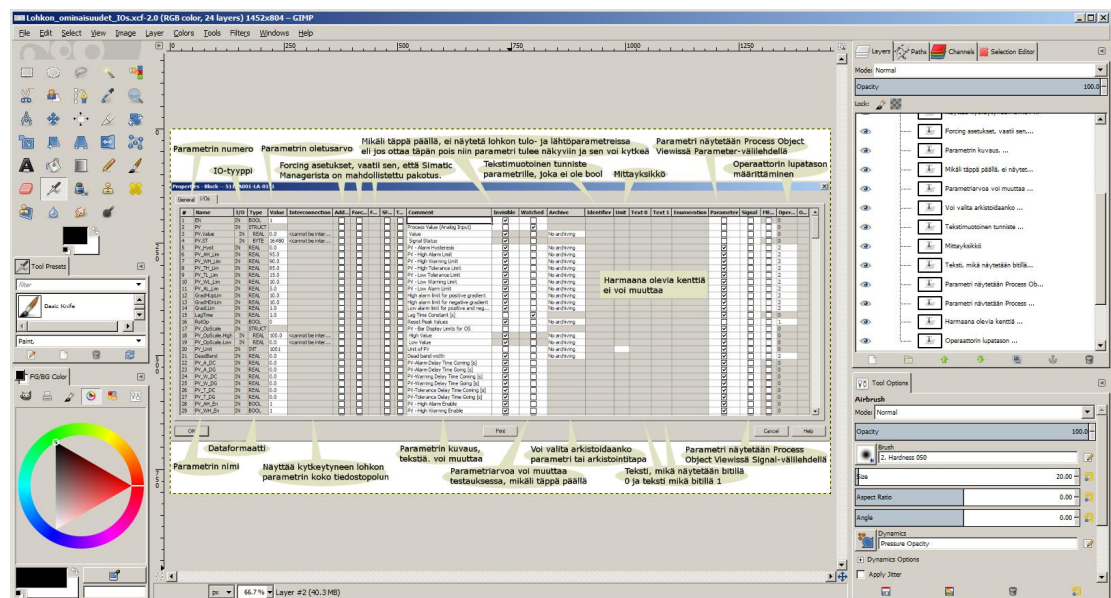
Opinnäytetyön ajan minulla oli käytössä PCS 7-järjestelmä. Pystyin sitä kautta keräämään aineistoa käyttämällä järjestelmän itsensä sisältämiä helpejä ja documentaatiota. Lisäksi aineistona käytin Siemensin sivuilta lataamiani manuaaleja.

Itse järjestelmästä pystyin ottamaan kuvakaappauksia eri työvaiheista, joita sitten tarpeen vaatiessa muokkasin GIMP-nimisellä kuvankäsittelyohjelmalla.

HTML-toteutuksessa käytin Programmer's Notepad-nimistä ohjelmaa, jolla tein html-sivuja keräämäni aineiston pohjalta. Lisäksi käytin JQuery-kirjastoa, jolla voi tehdä toiminnallisuutta sekä vaikuttaa html-sivujen ulkonäköön.

#### 3.1.1 Gimp

Gimp on akronyymi sanoista GNU Image Manipulation Program. Se on vapaasti jaettu ohjelma ja se on tarkoitettu tehtäviin kuten kuvan käsittelyyn, sommitteluun ja luomiseen (Gimp introduction, 2013).



Kuvio 10. Gimp-kuvankäsittelyohjelma.

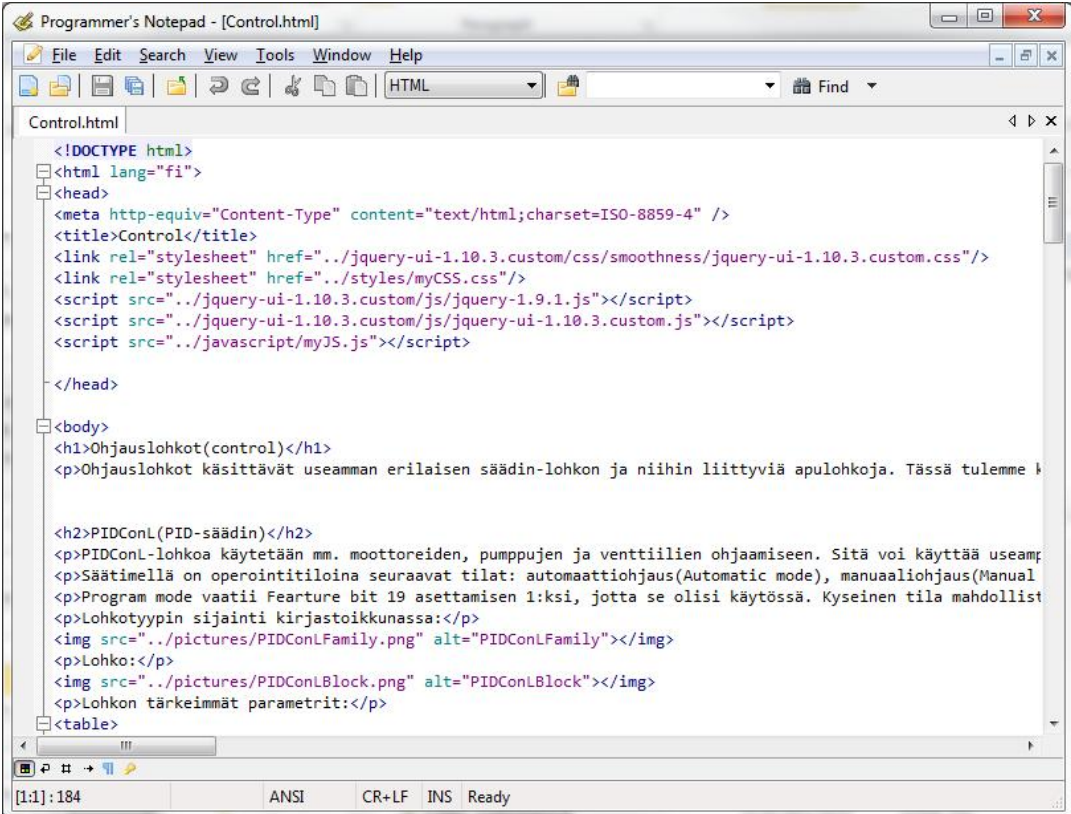
Mitä tein, oli, että otin PCS7-järjestelmästä kuvia eri objekteista kuten kaavioista, lohkoista, editoreista, ponnahdusikkunoista ja työn eri vaiheista. Näitä kuvia sitten käsittelin Gimpissä korostaen niiden tärkeimpiä kohtia, skaalaten niitä ja yhdistellen

niitä toisiin kuviin tarkoituksena esittää käsiteltävät asiat mahdollisimman selkeässä ja tiiviissä muodossa. Joissain kuvissa rajasin tärkeimmät kohdat, joihin lukijan halusi kiinnittävän huomiota ja joita sitten kommentoin tarkemmin.

### 3.1.2 Programmer's notepad

Programmer's notepad on avoimen lähdekoodin tekstieditori, joka on tarkoitettu käyttäjille, jotka työskentelevät lähdekoodin kanssa (Wikipedia, 2011).

Se tukee eri ohjelmointikielien syntaksia eli auttaa käyttäjää kirjoittamaan koodia oikein esim. korostaen virheellisesti kirjoitettuja kohtia. Se myös luo HTML-tagin lopetuksen automaattisesti, kun käyttäjä itse on kirjoittanut ensiksi tagin aloitusosan.



```

<!DOCTYPE html>
<html lang="fi">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-4" />
<title>Control</title>
<link rel="stylesheet" href="../jquery-ui-1.10.3.custom/css/smoothness/jquery-ui-1.10.3.custom.css"/>
<link rel="stylesheet" href="../styles/myCSS.css"/>
<script src="../jquery-ui-1.10.3.custom/js/jquery-1.9.1.js"></script>
<script src="../jquery-ui-1.10.3.custom/js/jquery-ui-1.10.3.custom.js"></script>
<script src="../javascript/myJS.js"></script>
</head>
<body>
<h1>Ohjauslohkot(control)</h1>
<p>Ohjauslohkot käsittävät useamman erilaisen säädin-lohkon ja niihin liittyviä apulohkoja. Tässä tulemm...</p>

<h2>PIDConL(PID-säädin)</h2>
<p>PIDConL-lohkoa käytetään mm. moottoreiden, pumppujen ja venttiilien ohjaamiseen. Sitä voi käyttää useamp...</p>
<p>Säätimellä on operointitiloina seuraavat tilat: automaattiohjaus(Automatic mode), manuaaliohjaus(Manual...</p>
<p>Program mode vaatii Feature bit 19 asettamisen 1:ksi, jotta se olisi käytössä. Kyseinen tila mahdollist...</p>
<p>Lohkotyyppi sijainti kirjastoikkunassa:</p>
</img>
<p>Lohko:</p>
</img>
<p>Lohkon tärkeimmät parametrit:</p>
<table>

```

Kuvio 11. HTML-sivu avattuna Programmer's notepadissa.

Käytin ohjelmaa kirjoittamalla HTML-sivut, niihin liittyvät tyylitiedostot eli CSS-tiedostot sekä toiminnallisuudesta vastaavat javascript-tiedostot. Kyseiset tiedostot olisin voinut kirjoittaa myös tavallisella notepadilla, mutta programmer's notepad, aikaisempien ominaisuuksiensa lisäksi, esittää kirjoitetun koodin selkeämmässä muodossa helpottaen näin tekemistä.

### 3.1.3 JQuery

JQuery on nopea, pieni ja ominaisuuksiltaan rikas JavaScript kirjasto. Se tekee asiat kuten html-dokumentissa kulkemisen, sen manipuloinnin, tapahtumien käsittelyn, animaatioiden ja Ajaxin paljon helpommaksi helppokäyttöisellä ohjelmointirajapinnalla, joka toimii monilla eri selaimilla.(jQuery What is jQuery?, 2013)

Kyseinen kirjasto lyhentää huomattavasti JavaScript-koodia, joten toiminnallisuksien luominen on helpompaa. Toimintojen lisäksi sen funktioilla on mahdollista myös muokata html-sivujen ulkonäköä ja luoda yhtenäinen teema sivuille.

Latasin JQueryyn sen kotisivuilta. Sillä sai muokattua nopeasti ja helposti PCS7-manuaalin HTML-sivujen ulkonäön. Käytin sen sisältämiä funktioita luoden manuaalin valikkorakenteen ja liikkumisen valikosta toiseen. Lisäksi tein mm. koodinpätkän, mikä mahdollisti kuvasarjojen käytön. Täten eri työvaiheiden opetus oli tehokkaampaa.

## 3.2 Työvaiheet

### 3.2.1 Rakenteen suunnittelu

Manuaalin sisältö päätettiin karkeasti opinnäytetyösuunnitelman yhteydessä. Käsiteltävät aihealueet olivat Simatic Manager, CFC, SFC, WinCC ja Import/Export-assistant. Sisältö tarkentui opinnäytetyön edetessä.

Haasteita rakenteen suunnitteluun tuottivat aikataulu ja varsinkin aiheen laajuus. Järjestelmä on niin suuri kokonaisuus, käsittäen monia eri työkaluja ja editoreita, hardware- ja yhteysasetuksia, joiden läpikäyminen opinnäytetyölle varatun ajan puitteissa ei ollut mahdollista. Käsiteltäviksi valitut asiat itsessään sisälsivät aihealueita, joiden katsoin tärkeiksi käsitellä, mutta jotka myös kasvattivat työn määrää. Olisin kyllä halunnut käsitellä monia muitakin aihealueita kuten järjestelmän simulointia, mutta opinnäytetyö olisi paisunut liian suureksi.

Rakennetta suunnitellessa piti miettiä mm. mitä asioita käsitelisi, mikä olisi paras suoritusjärjestys opettaa asiat ja miten jaata ja käsitellä asiat järkevissä kokonaisuuk- sissa. Esimerkkinä viimeisestä on se, että CFC-kaavioissa otetaan kantaa lohkoista



generoitavien lohkoikonien ulkoasuun, joka taas kuuluu hyvin kiinteästi WinCC:n aihealueeseen. Eli ongelmaksi muodostuu mihin vetää raja, jossa asian käsittely hoidetaan toisessa aihealueessa ja kuinka paljon samaa asiaa käsitellään molemmissa.

### **3.2.2 Aineiston keruu**

Aineistoa keräsin siten, että keskityin yleensä yhteen isoon kokonaisuuteen kuten CFC-editoriin ja siihen liittyviin lohkoihin ja kaavioihin, jonka jälkeen siirryin sekvensseihin jne.

Kirjoittaessani luin aiheeseeni liittyviä manuaaleja, käytin Siemensin helppiä ja testasin asioita järjestelmässä ottaen kuvakaappauksia eri työn vaiheista.

### **3.2.3 HTML, JQuery ja CSS**

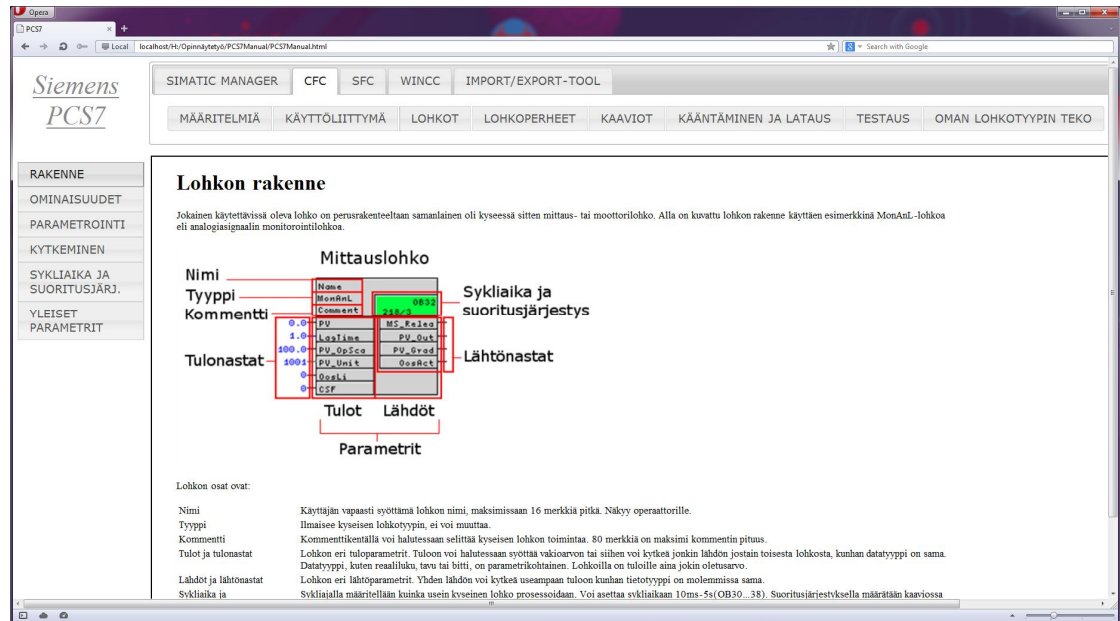
Ajatuksena esittää manuaali html-toteutuksena oli esittää asiat selkeästi, omina pieninä kokonaisuuksinaan ja joihin löytäisi helposti. Lisäksi tarkoituksena oli käyttää kuvasarjoja kuvatekstein, joilla jonkin asian havainnollistaminen tapahtuisi mahdollisimman tehokkaasti.

Opinnäytetyötä tehdessäni tutustuin html-kieleen ja eri kirjastojen käyttöön. Päätin käyttää työssä JQuery- nimistä kirjastoa, sillä se tarjosi yksinkertaisia funktiota toimintojen luomiseen sekä sillä sai helposti visuaalisen ilmeen luotua. Lisäksi tein omia CSS-tiedostoja, joilla pystyy muokkaamaan html-sivujen ulkonäköä.

### **3.2.4 Käyttöliittymän luominen**

Käyttöliittymää pystyin alkaa tekemään, kun aineistoa oli enemmän kertyneenä ja manuaalin rakenne paremmin hahmottuneena.

Yritin esittää asiat selkeästi niin, että käyttäjän ei tarvitsisi etsiä haluamaansa tietoa kovin pitkään. Päädyin esittämään asiat kolmessa tasossa niin, että ylimmäinen välilehti sisältäisi pääaihealueen kuten CFC:n, jota klikatessa sen alapuolelle avautuisivat siihen liittyvät aiheet kuten lohkot ja kaaviot. Esim. lohkot-kohtaa klikkaamalla taas avautuisi yleinen esittelysivu käsitellen lohkoja yleisesti ja antaen lukijalle yleissilmäyksen aiheesta. Samalla sivun vasemmalle puolelle ilmestyisi vielä lohkoihin liittyvät aiheet kuten lohkojen rakenne, ominaisuudet ym., joita klikkaamalla aiheeseen liittyvä sivu aukeaisi. Alla oleva kuvio hahmottaa käyttöliittymän rakennetta.



KUVIO 12. HTML-manuaalin käyttöliittymä Opera-selaimessa avattuna. Yläpalkissa on pääaiheet, sen alapuolella avattuna siihen liittyvät alialueet ja vasemmassa nurkassa vielä kyseinen aihe pilkottuna pienempiin kokonaisuuksiin. Käsiteltävä aihe on avattuna pääikkunassa.

## 4 Sovellussuunnittelun ohjeistus

Manuaalissa käsitellään siis aikaisemmin mainitut 5 pääaluetta niihin liittyvine aihealueineen. Ohjeistuksessa käytettiin materiaalina niin aikaisemmin tehdystä projektista saamaa materiaalia kuin myös itse tehtyä esimerkkiprojektia, jotta tietyt teko vaiheet voitiin selittää. Seuraavaksi tullaan käsittelemään kustakin pääalueesta esi-merkinomaisesti yksi sen sisältämistä kohdista.

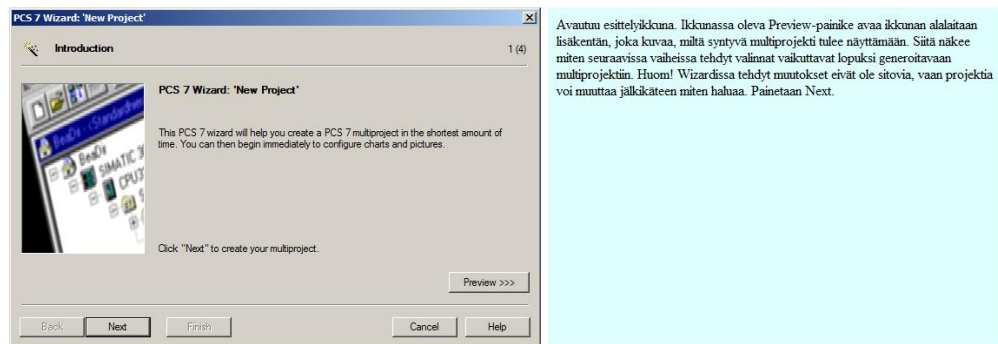
### 4.1.1 Multiprojektin luominen

Simatic manager-osion alla on kohta, jossa neuvotaan miten uusi multiprojekti luodaan. Projektin luonti tapahtuu avaamalla New Project Wizard- työkalun Simatic Managerin File-valikosta. Projektin luomisen yhteydessä määritetään mm. käytettävissä oleva CPU, kommunikaatiomodulien määrä, hierarkiatasojen määrä, oletuksena luotavat tyhjät kuvat, CFC- ja SFC-kaaviot, OS-asetat sekä tietysti projektin nimi ja tallennusosoite.(ks. kuvio 11.)

Osiassa näytetään luotu kuva oletusnäkyminen, jotka ovat plant ja component view sekä näytetään luodut objektit ja hierarkiakansiot, jotka luontivaiheessa oli määritetty. Pointtina mainitaan myös se, että aikaisemmin tehdyt valinnat eivät ole sitovia, vaan projektia voi muuttaa jälkikäteen miten haluaa.

## Multiprojektin luominen

Uuden multiprojektin luominen tapahtuu Simatic Managerin File-valikosta 'New Project' Wizard... alta. Seuraava kuvasarja selittää eri projektin luomisen vaiheet.



Edellinen Seuraava

Kun Simatic Manager on ladannut aikansa, avautuu kyseinen näkymä, missä on Component View- ja Plant View-näkymät.

KUVIO 13. Ote aihealueesta multiprojektin luominen. Esitettynä on New Project-wizard-ponnahdusikkuna ja sen oikealla puolella selventävä tekstikenttä. Painonapeista siirrytään kuvasarjassa eteen- ja taaksepäin selittäen eri Wizardin vaiheet ja valintojen merkitykset.

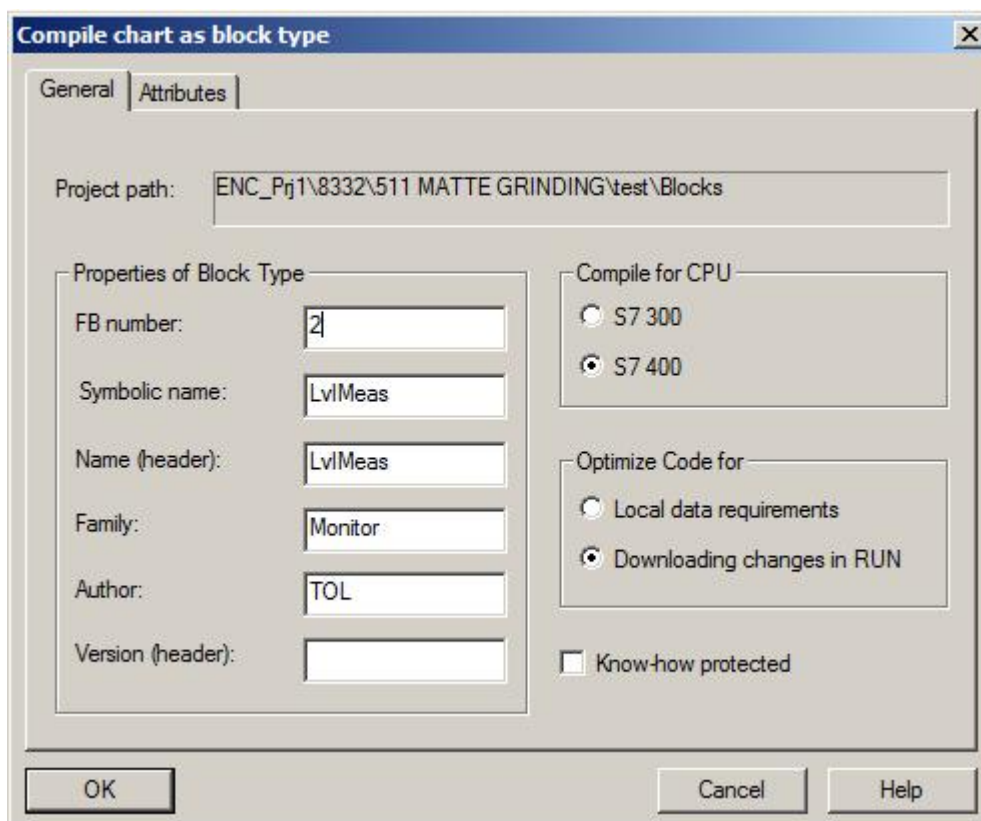
Lisäksi osiossa mainitaan miten luotuun multiprojektiin lisätään uusi kirjasto/projekti sekä selitetään erot tavallisen kirjaston ja master-kirjaston välillä.

### 4.1.2 Kaavion kääntäminen lohkotyypiksi

CFC-alueen alla on kohta "Oman lohkotyyppin teko" ja sillä vielä omat aihealueensa, joista käsittelemme yhden. Lohkotyyppin voi nimittäin tehdä kahdella eri tavalla ja toinen näistä tavoista on kaavion kääntäminen lohkotyypiksi.

Lohkotyyppin voi tehdä periaatteessa kaaviosta kuin kaaviosta. Rajoituksena on kuitenkin se, että kyseinen kaavio ei voi sisältää kaavion sisäisiä kaavioita (nested charts). Nämä ovat tavallisia kaavioita, toisen kaavion sisällä ja ovat ulkoasultaan kuka-takuinkin samanlaisia kuin tavalliset lohkot mutta poikkeavat kuitenkin tietyiltä ominaisuuksiltaan näistä. Esimerkiksi näiden "kaaviolohkojen" sisään on mahdollista päästä ja niitä voi muokata, kun taas lohkon tai kaavion, jonka on kääntänyt lohkotyypiksi, sisään ei enää pääse.

Lohkotyyppiä luotaessa on tärkeää määrittää sille IO:t eli tulot ja lähdöt, jotta siitä luotavaan lohkotyyppiin voi kytkeä tulo- ja lähtöparametreja tai signaaleja. Tämän jälkeen kaavion voi kääntää lohkotyypiksi valitsemalla menuvalikosta Chart ja sieltä Compile ja edelleen Chart as Block Type. Alla on kuvio avautuvasta ponnahdusikkunasta, jossa määritetään luotavalle lohkotyypille sen ominaisuudet.



KUVIO 14. Kaavion kääntäminen lohkotyyppiä. Luotavalle lohkotyyppiä määrätään mm. FB-numero, nimi ja lohkoerhe, johon se tullaan sijoittamaan. FB-numeron pitää olla vapaa ja sopia CPU:n rajoihin.

Kun tarvittavat tiedot on täytetty, kaavion voi kääntää lohkotyyppiä. Kääntämisen jälkeen luotu lohkotyyppi löytyy kirjastoikkunasta siitä lohkoerheestä johon se on määrätty ja siitä sen voi raahata käytettäväksi kaavioon kuin kaavioon.

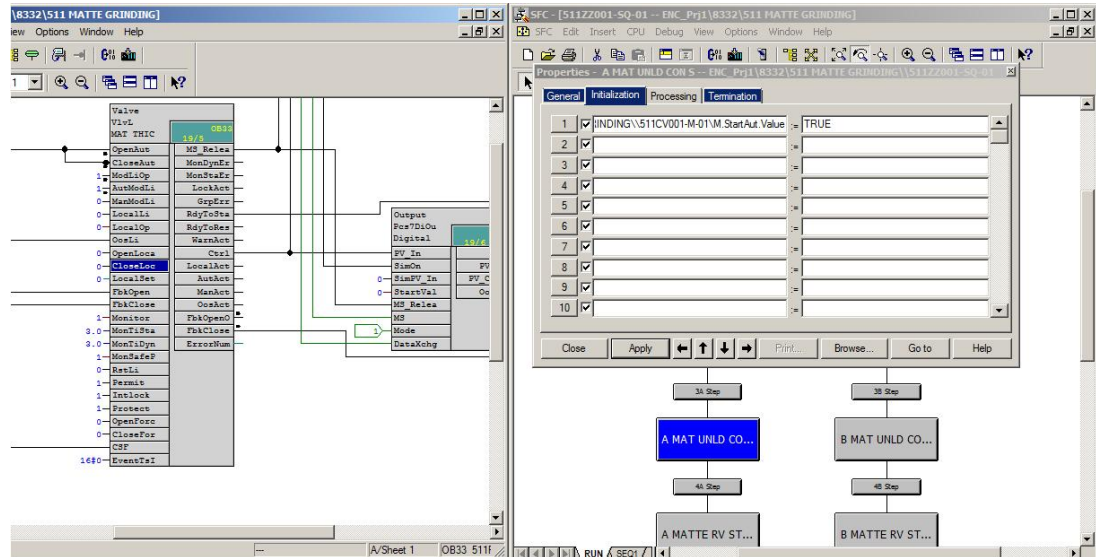
### 4.1.3 Toimintojen lisääminen askeleisiin

SFC-alueen alla on kohta "Sekvenssi", jolla itsellään on useita eri kohtia. Yksi näistä on toimintojen lisääminen askeleisiin. Kyseisessä kohdassa mainitaan 3 eri tapaa, miten toimintoja voi lisätä.

Toiminta on yksinkertaisuudessaan askeleeseen syötetty parametri ja sille määrätty arvo, joka syötetään kyseiselle parametrille askeleeseen tultaessa. Jotta ohjattaisiin oikeaa parametria, pitää parametrkenttään syöttää koko hierarkiakansoiden ketju, kaavion ja lohkon nimi sekä ohjattavan lohkon parametri. Syötettävä rivi on muodoltaan Tehdas\Alue\Alialue\...\kaavio\lohko.parametri.

Yksi tapa on kirjoittaa kyseinen rivi käsin, joka on aika virhealtis tapa. Toinen on klikata askeleen ominaisuuksissa olevaa Browse...-painiketta ja etsiä listalta kyseinen pa-

rametri. Kolmas tapa on avata kaavio, jossa kyseinen parametri on, klikata parametri aktiiviseksi ja raahata se askeleen parametririville. Alla oleva kuvio selventää kyseistä tapahtumaa.



KUVIO 15. Toiminnon lisääminen. Kuvassa on vasemmalla puolella avattuna CFC-kaavio, josta haluttu parametri on klikattu aktiiviseksi. Tämän jälkeen sen voi raahata hiirellä oikealla olevan SFC-kaavion askeleen ominaisuuksissa olevalle parametririville, jonka jälkeen sille voi syöttää ohjattava arvo.

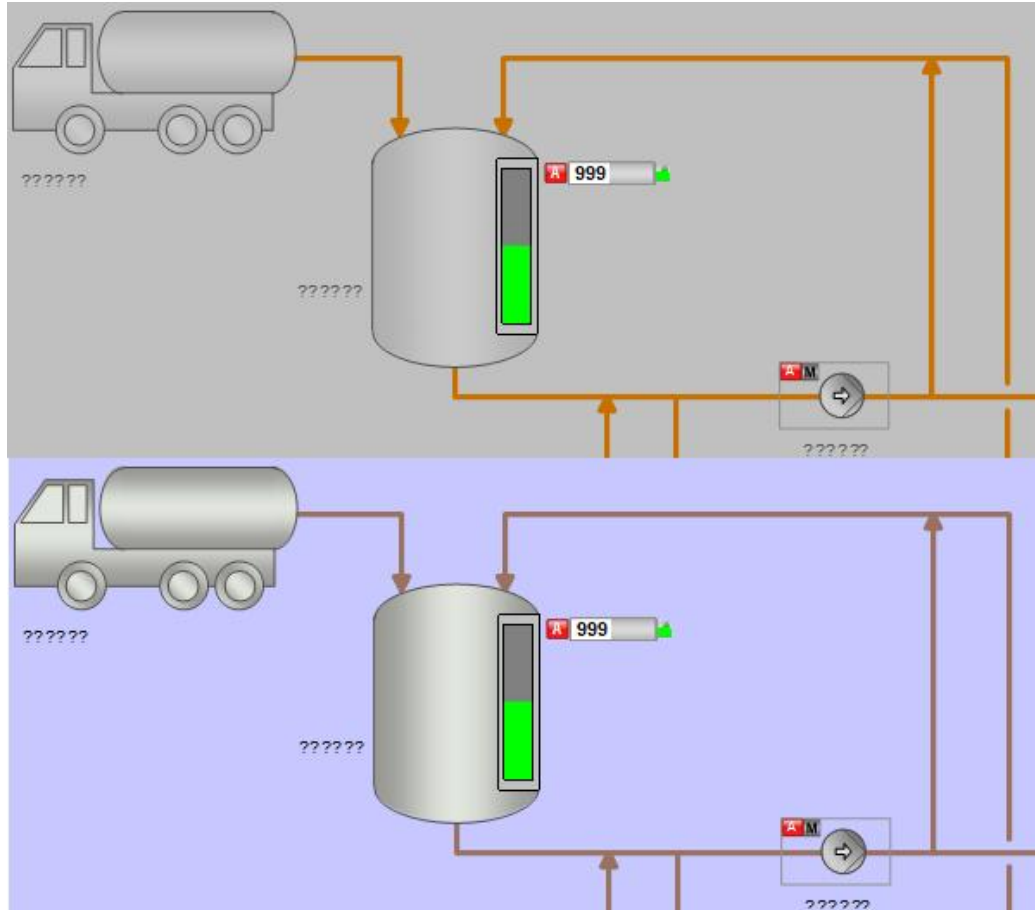
#### 4.1.4 WinCC-värit

WinCC-alueessa tärkeä käsiteltävä kohta on "Värit". Näyttökuvissa objektit käyttävät yleisesti globaalia väriteemaa. Väriteeman voi käydä muuttamassa WinCC:ssä avaamalla projektin ominaisuudet, menemällä välilehdelle User Interface and Design ja klikkaamalla sieltä Active Design-kondan Edit-painiketta. Avautuvasta ikkunasta voi valita jonkin oletusteeman tai luoda projektiin oman.

Objektien värejä voi näyttökuvissa muokata käsipelissäkin mikä monesti onkin hyvä ratkaisu, sillä globaali väriteema ei ole välttämättä tarpeeksi monipuolinen. Sellainen ongelma syntyy kuitenkin helposti, että halutaankin muuttaa käytettyjä värejä, jonka seurauksena pitäisi käydä läpi kaikki prosessikuvat ja niiden objektit ja muuttaa värit uudelleen käsipelissä. Ratkaisu tähän ongelmaan on käyttää paletteja.

Paletti on 20 värin tiedosto, jossa kullekin numerolle on määrätty tietty väri. Paletin voi tehdä samassa paikassa kuin mistä pääsee käsiksi globaaleihin väriteemoihin, klikkaa vain Central Color Palette-kohdan Edit-painiketta. Avautuvasta ikkunasta voi luoda omia paletteja, ottaa käyttöön olemassa olevia, muokata palettien värejä ym.

Nyt, jos tietyille prosessikuvissa oleville objekteille on väriksi määritetty palettiväri, ei väriä tarvitse käydä muuttamassa itse objektista vaan muuttaa kyseisen värin palettista. Tällöin projektin kaikkien prosessikuvien objektien väri vaihtuu uudeksi. Alla oleva kuvio havainnollistaa tätä toimintaa.



KUVIO 16. Palettivärien käytön havainnollistaminen. Prosessikuva ennen palettivärien muuttamista ja muutosten jälkeen. Kyseiset muutokset ovat tapahtuneet kaikkiin prosessikuviin. Palettien avulla saadaan siis helposti ja nopeasti muutettua kuvien väriteema halutunlaiseksi.

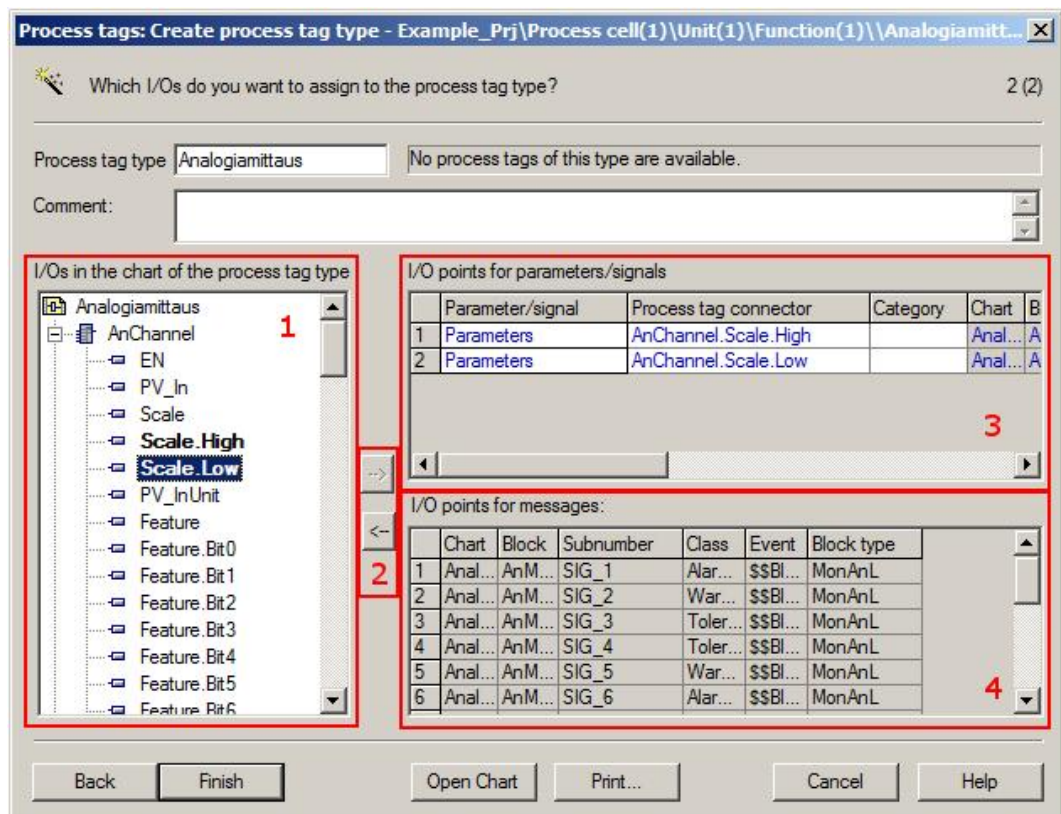
#### 4.1.5 Process tag typen luominen

Import/Export-assistentilla tai toolilla, joksi sitä kanssa kutsutaan, pystytään generoimaan kaavioita suuria määriä mallipohjan ja siitä luotavan import-filen pohjalta. Mallipohjaa kutsutaan process tag typeksi. Mallipohjan ideana on monistaa toiminnaltaan samanlaisia asioita kuten mittausta tai moottorinohjausta.

Mallipohjan voi luoda kaaviosta kuin kaaviosta. Mallipohjana käytettävän kaavion kannattaa kuitenkin tehdä toiminnaltaan mahdollisimman järkeväksi, kuten myös sen

parametointi, sillä siitä generoitavat kaaviot ovat niiden parametrien osalta identtisiä mallipohjan kanssa joita ei erikseen muuteta luotavassa import-filessa.

Mallipohjan luominen tapahtuu klikkaamalla olemassa oleva kaavio aktiiviseksi, klikkaamalla hiiren kakkospainiketta ja valitsemalla Process tags ja edelleen Create/Change Process Tag Type... Avautuu ponnahdusikkuna, jossa määritetään luotavalle mallipohjalle parametrit ja viestit, jotka otetaan mukaan import-fileen(ks. kuvio 15).



KUVIO 17. Process tag typen luominen. Ikkunassa määrätään mitkä parametrit otetaan mukaan luotavaan mallipohjaan. Näitä parametreja voidaan sitten muokata myöhemmin luotavassa import-filessa. Kuvassa on korostettu tärkeimmät alueet, jotka on sitten selitetty myöhemmin.

Kun tarvittavat parametrit on otettu mukaan, voidaan painaa Finish-painiketta, jolloin mallipohja luodaan projektin master-kirjastoon.

Mallipohjaa voi käydä myöhemmin muuttamassa, kuten myös import-fileen otettavia parametreja. Nämä muutokset eivät valitettavasti astu voimaan mallipohjasta aikaisemmin luotuihin kaavioihin ja import-fileen vaan import-fileen pitää tehdä käsipelissä tarvittavat parametrimuutokset ja generoida kaaviot uudelleen.

## 5 Tulokset

Opinnäytetyön tuloksena syntyi HTML-muotoinen manuaali, joka käsittää keskeiset sovellussuunnittelussa tarvittavat editorit, työvaiheet ja riittävän tietoperustan, jotta järjestelmään vihkiytymättömän on helpompi päästä järjestelmään käsiksi. Pääpaino on ollut käytännön esimerkeissä, jotka opastavat keskeisiin työvaiheisiin, näin nopeuttaen sovellussuunnitteluun ryhtymistä.

Manuaali on hyvin kuvapainotteinen, minkä lisäksi suurinta osaa kuvista on kommentoitu, jotta lukija ymmärtäisi paremmin keskeiset asiat. HTML on myös mahdollistanut kuvasarjojen käytön, joten työvaiheet on pystytty selittämään tiiviissä ja selkeässä muodossa. Kaikestaan manuaali sisältää kuvia yli 170 kpl:ta ja eri aihealueet ovat jaettu 45 eri HTML-sivulle.

Kuvien lisäksi teoriapohja on vahva, varsinkin CFC-editoria, kaaviota ja lohkoja käsiteltäessä. Kyseinen työkalu on sovellussuunnittelun kannalta erittäin oleellinen, joten kattava ymmärrys sen eri aihealueista on tärkeää. Lisäksi keskeiset käsitteet ja erot mm. samantapaisten objektien välillä on selitetty, jotta lukija ei menisi niissä sekaisin. Manuaali on kirjoitettu suomeksi, mutta monesti suomenkielistä vastinetta oleva objekti on mainittu suluissa englanniksi sekaannusten välttämiseksi.

Yksi Jeecin suunnittelijoista vilkaisi manuaalin läpi. Hänen mielestään manuaali vaikutti ihan pätevältä ohjeelta. Kritiikkiä tuli vain siitä, että pienimmät nippelitiedot olisi voinut jättää pois. Jos menee liian tarkaksi, niin lukijalta häviää ajatus, mistä oli tiin puhumassa.



## 6 Pohdinnat

Olen melko tyytyväinen tuottamaani manuaaliin. Haasteita sen tekemiselle on tietenkin ollut monia. Oma teoriapohjani ja käytännön kokemukseni kyseisestä järjestelmästä olivat olleet melko vähäiset, joten siihen tutustumiseen ja sitä koskevien manuaalien lukemiseen on mennyt oma aikansa. Lopputuloksena oma tietämykseni järjestelmästä on merkittävästi kasvanut, joten sovellussuunnittelu jatkossa tulee olemaan itselleni huomattavasti helpompaa. Myös kokemukseni HTML-kielestä ja kaikesta siihen liittyvästä olivat olleet varsin vähäiset, mutta hyödyn sen käyttämiselle katsoin merkittäväksi manuaalin teossa. Manuaaleitakaan ei ole tullut aikaisemmin kirjoitettua, mutta esimerkit olivat minusta melko hyvin esitetty ja kommentoitu. Uskon, että kynnyks opetella ainakin perusasiat ovat tätä manuaalia lukiessa helpompi omaksua kuin että lukisi monta sataa tai tuhatta sivuista manuaalia läpi ja joka ei sisällä kuvan kuvaa. Ja kun perusasiat on tullut opeteltua, tarkempien tietojen löytäminen kyseisistä manuaaleista on helpompaa.

Koska käsiteltävä järjestelmä on laaja kokonaisuus ja opinnäytetyö käsitteli vain osaa sen aihealueista, manuaalista löytyy kyllä kehityskohteita. Aihealueita mitä voisi käsitellä on esim. järjestelmän simulointi, eri Step 7-työkalujen käsittely, yhteysasetusten teko, hardware-konfiguraatiot ym. Lisäksi jo kirjoitetuista aihealueista löytyisi kyllä kohtia, joita voisi selittää syvällisemmin tai mukaan voisi ottaa uusia kohtia, joita tässä manuaalissa ei käsitelty.

Itse manuaalin rakennetta voi aina kehittää sujuvammaksi niin, että kaikki asiat loimituvat toisiinsa luonnollisemmin. Materiaalina oli käytetty siis jo olemassa olevaa projektia sekä itse tehtyä esimerkkiprojektia. En usko, että siitä oli mitään merkittävää haittaa, mutta tietenkin jos esimerkkiprojektissa itsessään olisi käyty koko sovellussuunnittelun eri vaiheet läpi, olisi kokonaisuus hieman nykyistä sujuvampi. Se olisi tosin vienyt hieman enemmän aikaa.

HTML mahdollistaa monia toimintoja, joten asioiden opettaminenkin voisi olla tehokkaampaa. Esimerkkikehityskohteina voisivat olla animaatioiden ja videotiedostojen käyttö. Ne tietenkin lisäisivät työn määrää ja vaatisivat toteutukselle suurempaa suunnitelmallisuutta, mutta hyötynä tietenkin olisi, että manuaalia lukeva oppisi ymmärtämään keskeiset käsitteet nopeammin ja varmemmin.

Merkittävin hyöty manuaalin käyttämisessä ja sen jatkokehitykselle on juuri siinä, että varsinkin aloitteleva insinööri tai järjestelmään aikaisemmin vihkiytymätön voitaisiin opettaa mahdollisimman nopeasti kyseiseen järjestelmään kiinni. Siinä säästettäisiin aikaa ja mahdollisten virheiden syntyminen sovellusta suunniteltaessa. Virheitä ja vääriä toimintatapoja voidaan ehkäistä antamalla tietoa ja luomalla ymmärrystä, jota uskon tämän manuaalin omalta osaltaan tarjoavan.

## Lähteet

Introduction to GIMP. 2013. <http://www.gimp.org/about/introduction.html>

JEEC Oy. Kotisivut. 2013. <http://jeec.fi/index.htm>

Programmer's Notepad. 12/2011.

[http://en.wikipedia.org/wiki/Programmer's\\_Notepad](http://en.wikipedia.org/wiki/Programmer's_Notepad)

Siemens Simatic. Process Control System PCS 7, CFC for SIMATIC S7(V8.0). Function Manual. 12/1012 . Document identification number: A5E03710059-04.

<https://support.automation.siemens.com/WW/llisapi.dll?query=PCS7+process+control+system&func=cslib.cssearch&content=adsearch%2Fadsearch.aspx%3Ftab%3D2&lang=en&siteid=csius&objaction=cssearch&searchinprim=searchinprim &nodeid=99&x=16&y=8&cssearchengine=NEW&x=-970&y=-121>

Siemens Simatic. Process Control System PCS 7, Engineering System(V8.0 SP1). Configuration Manual. 3/2013. Document identification number: A5E02779453-03.

<https://support.automation.siemens.com/WW/llisapi.dll?query=PCS7+process+control+system&func=cslib.cssearch&content=adsearch%2Fadsearch.aspx%3Ftab%3D2&lang=en&siteid=csius&objaction=cssearch&searchinprim=searchinprim &nodeid=99&x=16&y=8&cssearchengine=NEW&x=-970&y=-121>

Siemens Simatic. Process Control System PCS7, Operator Station(V8.0 SP1). Configuration Manual. 3/2013. Document identification number: A5E02779465-04.

<https://support.automation.siemens.com/WW/adsearch/resultset.aspx?region=WW&lang=en&netmode=internet&ui=NDAwMDAxNwAA&term=PCS7+process+control+system&ID=68157026&ehbid=68157026>

Siemens Simatic. Process Control System PCS 7, SFC for SIMATIC S7. Programming and Operating Manual. 12/2011. Document identification number: A5E03710074-01.


<https://support.automation.siemens.com/WW/llisapi.dll?query=PCS7+process+control+system&func=cslib.cssearch&content=adsearch%2Fadsearch.aspx%3Ftab%3D2&lang=en&siteid=csius&objaction=cssearch&searchinprim=searchinprim &nodeid=99&x=16&y=8&cssearchengine=NEW&x=-970&y=-121>

What is jQuery?. 2013. <http://jquery.com/>

WinCC Explorer. Help, Communication, WinCC process communication, Principle of WinCC communication

# Liitteet

## Liite 1. Lohkon parametointi



RAKENNE

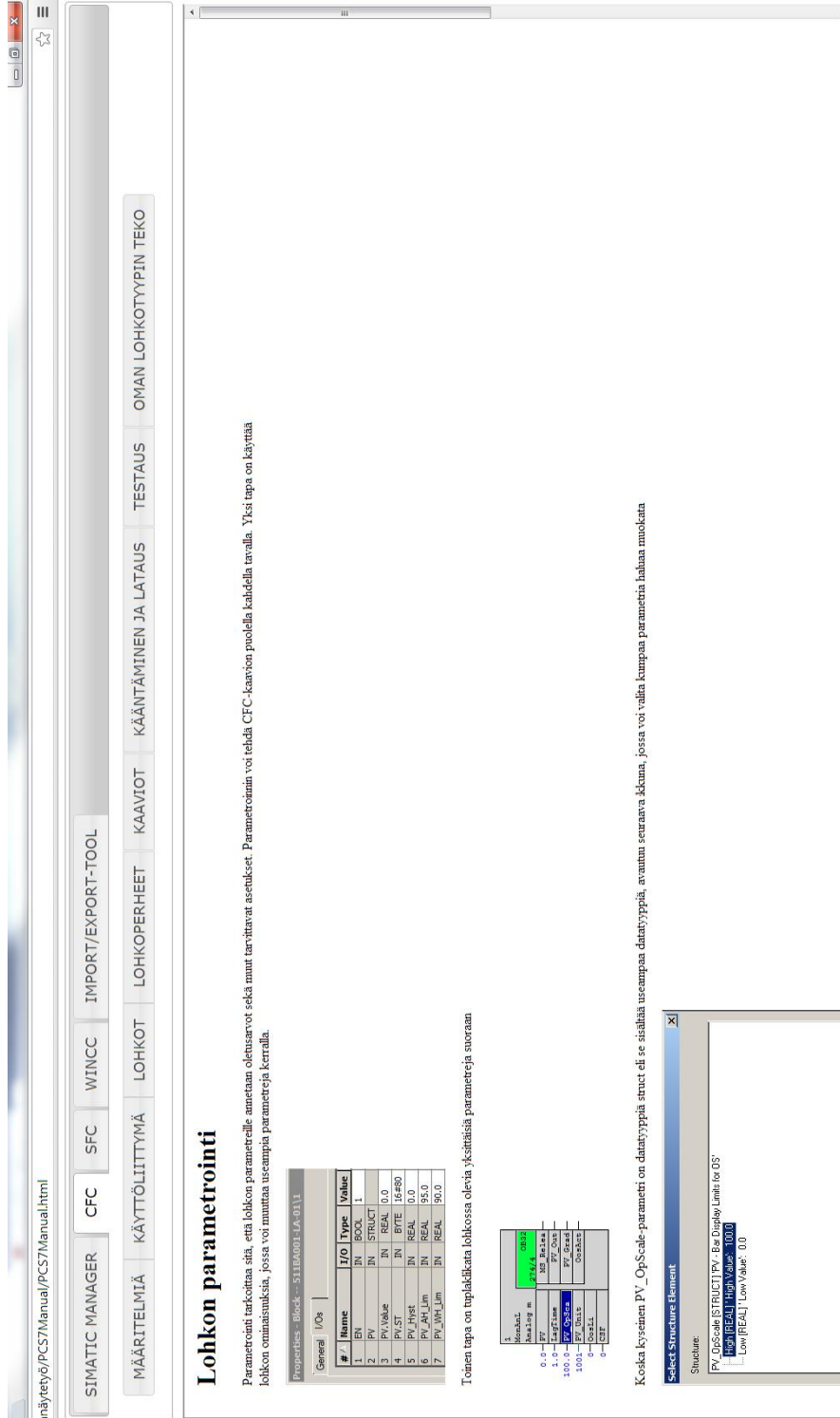
OMINAISUUDET

PARAMETROINTI


KYTKEMINEN

SYKLAIKA JA SUORITUSJÄRJ.

YLEISET PARAMETRIT



## Liite 2. Operointilohkot



[SIMATIC MANAGER](#)
[CFC](#)
[SFC](#)
[WINCC](#)
[IMPORT/EXPORT-TOOL](#)

[MÄÄRITELMIÄ](#)
[KÄYTTÖLIITYMÄ](#)
[LOHKOT](#)
[KAAVIOT](#)
[KÄÄNTÄMINEN JA LATAUS](#)
[TESTAUS](#)
[OMAN LOHKOTYYPIN TEKO](#)

**Operointilohkot(operate)**

Operointilohkoja käytetään signaalien tuottamiseen operaattorin toimesta. Treenää on tarkoituksena ohjata jokin tai asettaa jokin arvo. Operointilohkoista käsitellemme tässä vain lohkot OpAnL-, OpD01- ja Op-Ig-lohkot. Jätämme valittu näitä vastaavat suppeammat "small"-versiot, sekä OpStations-lohkon.

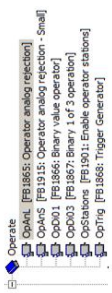
**OpAnL(analogiaoperointi)**

OpAnL-lohkoa käytetään analogiatulon tarkasteluun ja lähettämiseen. Analogiatulo tulee joko CFC/SFC:n kautta eli järjestelmän sisäpuolelta(internal) tai operaattorin syöttämästä(external). Lohkolla operaattori voi asettaa esim. säätimen asetusarvon haluttuun tasoon.

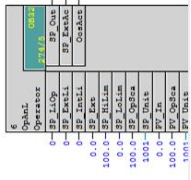
Lohkolla voi määrittellä suurimman ja pienimmän syötettävän arvon. Syötettyä asetusarvoa voi säätää halutessaan rampin kautta(ising setpoint ramp). Rampua voi käyttää aika-arvoa tai gradienttia. Gradientin suuruutta voi rajoittaa niin ylä- kuin alasmuuntaan eli muutosnopeus ei yltäisi asetettuihin maksimiin. Halutessaan saa tiedon, mikäli asetetut rajat tai gradientit on saavutettu tai ylitetty.

Lohkon parametreihin SetP1 ja SetP2 voi kytkeä jomkin toisen lohkon minkä tahansa parametrin, jolloin kyseiset lohkot saa araitua tämän lohkon faceplatesta. Lohkolla on kaksi muotoa, "On" ja "Out of service". Sillä voi generoida halutessaan neljä omaa viestiä(parameter EXNlsg1..4). Lohkosta on olemassa suppeampi versio OpAUS.


Lohkotyypin sijainti kirjastoikkunassa:



Lohko:



## Liite 3. Instanssin luominen



SFC TYPEN LUOMINEN

INSTANSSIN LUOMINEN

SIMATIC MANAGER

SFC

WINCC

IMPORT/EXPORT-TOOL

SEKVENSSI

SFC TYPE

KÄÄNTÄMINEN JA LATAUS

TESTAUS

### Instanssin luominen

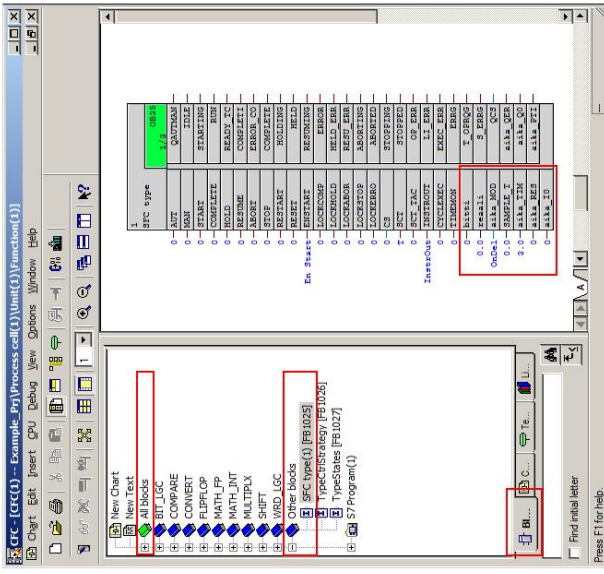
SFC instanssi luodaan aina CFC-kaavion. Instanssin luominen tapahtuu vastaavasti kuin millä tahansa lohkokäytöllä. SFC type on esimerkiksi lisätty CFC(I)-kaavioon sisään.

Ensin esitään SFC type, joka löytyy joko All blocks- tai Other blocks-kansioista kirjastokkaman Blocks-vallehdeltä. Sieltä lohkokäyttö siirretään raaka-annalla se CFC-kaavioon sisään. Oletuksena on nimi 1, jonka voi muuttaa lohkon ominaisuuksista.

SFC instanssi on ulkonäköltään miltei saman näköinen kuin tavallinen SFC-kaavio external tiivissä. Sillä on samat oletusparametrit näkyvissä kuin tavallisella SFC:llä. Poikkeuksen muodostavat kolme parametria, jotka luotiin aikaisemmin SFC tyyppiin: biti, reaali ja aika. Kyseiset parametrit saa näkyviin normaalisti lohkon ominaisuuksista meremällä I/Os-vallehdelle ja pistämällä kyseisiltä parametreilta tapaan Invisible pois päältä.

Toinen tapa on pistää kyseiset parametrit SFC tyyppiä näkymään lohkoinstanssin oletuksena. Alla on kuvaus ja sen tekemisestä.

Koska SFC instanssi on CFC-kaavioon sisällä, samaan kaavioon pystyy lisäämään muita lohkoja ja tekemään kytköksiä. SFC-kaaviossa pystyy tekemään vain kytköksiä.



## Liite 4. Import-filen luominen

Import-file luodaan klikkaamalla master-krjastossa haluttua process typeä, painamalla hiiren kakko-painiketta ja valitsemalla avautuvasta valikosta Process Tags->Assign Create Import File...

**Import-filen luominen**

Process Tags Assistant: Assign the Import File to the Process Tag Type

With the assistant, you can:

- Assign an import file to a process tag type.
- Check the assignment of the import file to the process tag type.
- Create a template of the import file for the process tag.

All process tag types are stored in the master data library

Master data library: Example.Lib