

# TYYPPILOHKOT ÄLYKKÄILLE MOOTTORINOHJAIMILLE TIA-PORTAL - YMPÄRISTÖÖN

Juha-Matti Tammiranta

Opinnäytetyö  
Lokakuu 2013

Automaatiotekniikan koulutusohjelma  
Tekniikan ja liikenteen ala





Tekijä(t) TAMMIRANTA, Juha-Matti	Julkaisun laji Opinnäytetyö	Päivämäärä 14.10.2013
	Sivumäärä 53	Julkaisun kieli Suomi
		Verkkojulkaisulupa myönnetty (X)
Työn nimi TYYPPILOHKOT ÄLYKKÄILLE MOOTTORINOHJAIMILLE TIA PORTAL YMPÄRISTÖÖN		
Koulutusohjelma Automaatiotekniikan koulutusohjelma		
Työn ohjaaja(t) SELOSMAA, Seppo RANTAPUSKA, Seppo		
Toimeksiantaja(t) JEEC Oy PEKKANEN, Jarmo		
Tiivistelmä <p>Opinnäytetyön tavoitteena oli luoda väyläkommunikoinnin sisältävät moottorinohjauksen tyyppilohkot Siemens Simocode –moottorinohjaimelle sekä ABB -taajuusmuuttajalle sovellussuunnittelun tehostamiseksi. Lisäksi työhön kuului HMI-objektien luominen moottorin piiriinäyttöön väyläinformaation seuraamista varten. Työn toimeksiantajana oli jyvaskyläläinen automaatioalan suunnitteluyritys JEEC Oy.</p> <p>Työ koostui karkeasti suunnittelu- ja testausvaiheesta. Suunnitteluvaihe jakautui vielä kahteen osaan: lohkojen suunnittelu ja käyttöliittymän suunnittelu. Lohkojen valmistuttua alkoi testausvaihe, joka suoritettiin Simocoden osalta JEEC Oy:n toimistolla olleella Simocode demosalkulla. Taajuusmuuttajan lohko testattiin erään projektin käyttöönottilanteessa, jossa se myös otettiin välittömästi käyttöön.</p> <p>Tyyppilohkot sekä käyttöliittymä todettiin testausvaiheessa toimiviksi ja helppokäyttöisiksi. Valmiin väyläkommunikoinnin ansiosta turha lohkon ulkopuolisten tiedonsiirtofunktioiden käyttö loppui kokonaan ja ajankäyttö tehostui. Myös käyttöönottilanteessa tärkeä informaatio, kuten vikadiagnostiikka on nyt myös paremmin saatavilla.</p> <p>Tyyppilohkojen sovellusrakenne on suunniteltu niin että lohkojen toimintaa on tulevaisuudessa helppo laajentaa ja monipuolistaa. Esimerkiksi laajemmalle vikadiagnostiikalle on varattu sovelluksesta jo tilaa. Hyvin kommentoitujen toimintojen ansiosta myös isommat muutokset lohkoissa on helppo toteuttaa.</p>		
Avainsanat (asiasanat) tyyppilohko, simocode, taajuusmuuttaja, tia portal, profibus		
Muut tiedot Liitteenä simocoden ja taajuusmuuttajan paremetrointiohjeet, 25 sivua.		



Author(s) TAMMIRANTA, Juha-Matti	Type of publication Bachelor's Thesis	Date 14.10.2013
	Pages 53	Language Finnish
		Permission for web publication (X)
Title FUNCTION BLOCKS FOR INTELLIGENT MOTOR CONTROLLERS FOR TIA PORTAL ENVIRONMENT		
Degree Programme Automation technology		
Tutor(s) SELOSMÄÄ, Seppo RANTAPUSKA, Seppo		
Assigned by JEEC Ltd. PEKKANEN, Jarmo		
Abstract <p>The goal of bachelor's thesis was to create function blocks with integrated bus communication for Siemens Simocode motor controller and ABB frequency converter to make software engineering more efficient. In addition, the work contained HMI objects for bus communication monitoring. The work was commissioned by automation engineering firm JEEC Ltd. from Jyväskylä.</p> <p>The work consisted of planning and testing phase. The planning phase was divided into two parts: function block- and user interface planning. As soon as the blocks were ready, the testing phase began in JEEC Ltd's office with Simocode demo-controller. The frequency controller block was tested in actual project, and it was harnessed immediately to use.</p> <p>Function blocks, as well the user interface was found to be valid and easy to use. Usage of external data transfer functions was completely ceased and planning efficiency increased due to integrated bus communication. Also important information for commissioning such as fault diagnosis, is now easily available.</p> <p>Function blocks are designed so that the functionality of the blocks is easy to expand and diversify in future. For example, there are extra space reserved for diverse fault diagnostics in application. Thoroughly commented features make the blocks also easy-to-update for bigger changes.</p>		
Keywords function block, simocode, frequency converter, tia portal, profibus		
Miscellaneous Simocode and frequency converter parametrization instructions attached, length 25 pages.		

# SISÄLTÖ

SANASTO .....	3
1. LÄHTÖKOHDAT .....	4
1.1. Toimeksiantaja .....	4
1.2. Toiminnalliset tavoitteet .....	4
1.3. Henkilökohtaiset tavoitteet .....	5
2. TIETOPERUSTA .....	5
2.1. Sovellussuunnittelu .....	5
2.2. TIA Portal .....	6
2.3. Simocode pro .....	11
2.4. Taajuusmuuttaja (ABB) .....	12
2.4.1. Yleistä .....	12
2.4.2. ABB F-sarjan väyläsovitin .....	12
2.5. Kenttäväylä .....	12
2.5.1. Yleistä .....	12
2.5.2. Profibus .....	13
3. TYÖN TOTEUTTAMINEN .....	14
3.1. Toteutusprosessi .....	14
3.2. Perusmoottorin tyyppilohko ja sen hyödyntäminen .....	14
3.3. Simocoden tyyppilohko .....	15
3.4. ABB:n taajuusmuuttajan tyyppilohko .....	18
3.5. HMI-objektit .....	20
3.6. Testaus .....	22
4. TULOKSET .....	24
5. POHDINTA .....	25
LÄHTEET .....	26
LIITTEET .....	28
LIITE 1. Simocode perusparametrit suunnanvaihtokäynnistimelle .....	28
LIITE 2. FBPA-01 perusparametrit .....	28

## KUVIOT

KUVIO 1 Automaation elinkaarimalli (Laatu automaatiossa, parhaat käytännöt. 2001, 17.).....	6
KUVIO 2 STEP7 -ohjelmointityökalu .....	7
KUVIO 3 WinCC -ohjelmointityökalu.....	7
KUVIO 4 Valokuvamainen HW-konfiguraatio .....	8
KUVIO 5 TIA Portal pääikkuna .....	9
KUVIO 6 Projektipuun rakenne .....	9
KUVIO 7 PLC:n asetusruutu .....	10
KUVIO 8 Näyttöeditorin työkaluikkuna .....	10
KUVIO 9 Perusmoottorin toimilohko .....	15
KUVIO 10 Binäärisen ohjauksen siirto lähtöön .....	15
KUVIO 11 Ohjauksanan lähetys Simocodelle .....	16
KUVIO 12 Simocode oletusohjauksanan muodostuminen .....	16
KUVIO 13 Simocode Data Records .....	17
KUVIO 14 Simocode Data Record –alueen lukeminen .....	17
KUVIO 15 ABB Ohjaus- ja tilasanan muodostuminen .....	19
KUVIO 16 ABB Datasanomatyypien pituudet .....	20
KUVIO 17 Simocode HMI-näyttö.....	21
KUVIO 18 ABB HMI-näyttö .....	21
KUVIO 19 Multiplex-tagin määrittely.....	22
KUVIO 20 Simocode testiympäristö .....	23
KUVIO 21 Testilaitteiston HW-konfiguraatio.....	23
KUVIO 22 Simocode ES -parametrinti ohjelma .....	24

## SANASTO

Tyypilohko	Mallipohja jonkin toimilaitteen ohjelmalohkolle. Pohjaan tehdyt muutokset päivittyvät automaattisesti kaikkiin sitä käyttäviin laitteisiin.
HMI	Human-Machine Interface. Operaattorin ja järjestelmän välinen käyttöliittymä.
DB	Data Block. Paikka jossa sovellus säilöö tietoa. Datablockeja voi olla globaaleja tai johonkin tiettyyn lohkoon viittaavia.
HEX	Heksadesimaali. 16 –kantainen lukujärjestelmä. Heksadesimaalijärjestelmä käyttää numeroiden lisäksi kirjaimia A-F.
INT	Integer- tietotyyppi. Kokonaisluku välillä -32768 ... 32767.
PLC	Programmable Logic Controller. Ohjelmoitava logiikka.
HW-konfiguraatio	Konfiguraatio joka sisältää PLC:n ja siihen liittyvien laitteiden määrittelyt, sekä myös järjestelmän fyysisen rakenteen.

# 1. LÄHTÖKOHDAT

## 1.1. Toimeksiantaja

Opinnäytetyön toimeksiantajana toimi jyväskyläläinen vuonna 2009 perustettu JEEC Oy, joka tarjoaa suunnittelu- ja konsultointipalveluita automaatio- ja sähkötoimialoilla. Yrityksen ydinsaaminen painottuu automaatioon kattaen palvelut aina esisuunnittelusta ylläpitotehtäviin. JEEC Oy on tehnyt projekteja seuraavilla prosessialueilla:

- Selluteollisuus
- Paperiteollisuus
- Energia
- Perusmetalli
- Kemia
- Ympäristöteknologia

(JEEC Oy, 2013.)

## 1.2. Toiminnalliset tavoitteet

Opinnäytetyön päätavoitteena oli toteuttaa moottorin tyyppilohkot, jotka sisältäisivät väyläkommunikaation Simocode -moottorinohjaimen sekä ABB:n taajuusmuuttajan kanssa Profibus –väylää hyödyntäen. Kommunikointiosiot sovitettaisiin jo olemassa olevaan moottoritoimilohkon pohjaan. Uudet lohkot nopeuttaisivat suunnittelijan työtä, koska eri moottorinohjausten kommunikointiprotokollat olisivat integroituna lohkoon, eikä niitä tarvitsisi enää erikseen määritellä. Jokaiselle moottorityypille tulee oma lohko.

Lisäksi työhön kuului väyläviestien seurannan lisääminen moottorin operoinnin piiriin näyttöön. Seurantaikkunasta näkisi käytössä olevan moottorityypin sekä tietoja väylästä, kuten ohjauksen tila, moottorin todellinen tila, mittausarvoja sekä mahdollisia virheviestejä. Ikkunan kautta pystyisi tarpeen tullen myös lähettämään ohjausviestejä suoraan väylään.

Moottorinohjainten tarvitsemista parametreista tuli myös koota parametrintiohje, joka nopeuttaisi laitteiston käyttökuntoon saamista käyttöönottilanteessa. Etusijalla ovat väyläkommunikoinnin parametrit.

### **1.3. Henkilökohtaiset tavoitteet**

Ensisijaisena tavoitteenani oli tutustua sovellussuunnittelijan työhön ja oppia erilaisen moottorinohjainten väyläkommunikoinnista ja yleisistä ohjausperiaatteista. Erittäin tärkeänä pidin myös eri työkalujen, kuten TIA Portalin ja Simocode ES –parametrintiohjelman tutuksi tulemistä sekä testausvaiheessa saatavia käytännön kokemuksia profibus –väylästä ja erilaisista älykkäistä moottorinohjaimista.

## **2. TIETOPERUSTA**

### **2.1. Sovellussuunnittelu**

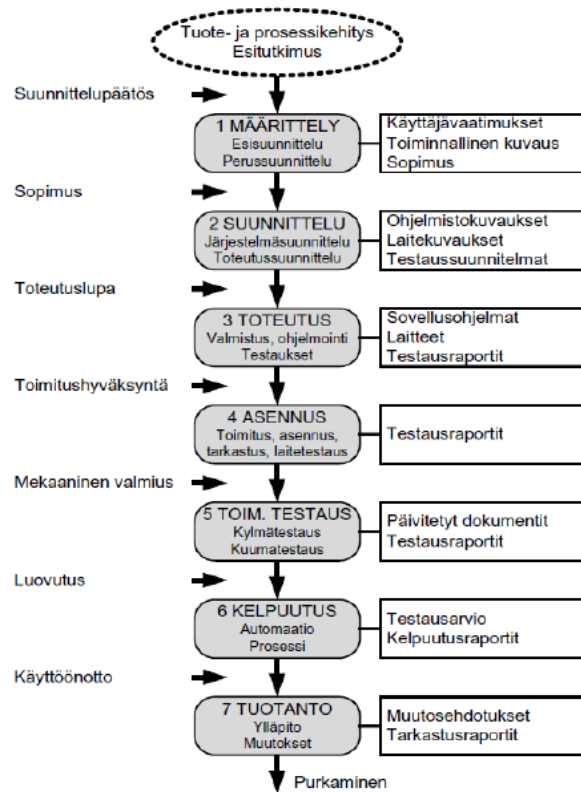
Automaation elinkaarimallissa sovellussuunnittelu sijoittuu suunnittelu- ja toteutusvaiheeseen. (ks. kuvio 1.) Automaation sovellussuunnittelu sisältää logiikan sekä käyttöliittymän sovelluksen suunnittelua. Sovellussuunnittelu on paljolti riippuvainen laitesuunnittelun etenemisestä ja ne kulkevatkin usein rinnan. (Laatu automaatiossa, parhaat käytännöt. 2001, 50.)

Logiikkasovelluksen suunnittelun ensimmäisessä vaiheessa määritellään rakenne, tietokannat ja kaikki vakioratkaisut, kuten tyyppiipiirin käyttö. Sen jälkeen kun ensimmäisessä vaiheessa tehdyt ratkaisut hyväksytään, voi ohjelmistomoduulien kuten säätöpiirien tai sekvenssien yksityiskohtainen suunnittelu alkaa. Logiikkasovelluksen suunnittelussa tulee hyödyntää mahdollisimman paljon jo toimivaksi todettuja, valmiita tyyppilohkoja. (Laatu automaatiossa, parhaat käytännöt. 2001, 50.)

Käyttöliittymän suunnittelu aloitetaan samaan aikaan logiikkasovelluksen suunnittelun kanssa. (Laatu automaatiossa, parhaat käytännöt. 2001, 51.) Näyttöjärjestelmän, yksittäisten näyttöjen sekä hälytysten suunnittelu ovat käyttöliittymäsuunnittelun pääkohteet. Yksittäisten näyttösivujen selkeys ja informatiivisuus ovat erittäin tärkeässä osassa käyttöliittymäsuunnittelussa, mutta myös näyttöjen hallinta sekä käyttö-



liittymän yleinen käytettävyys on otettava huomioon. Yleensä yksittäisten näyttösi-  
vujen informaatio jaetaan erilaisten toiminnallisten kokonaisuuksien mukaan, kuiten-  
kin siten, että yleisvalvonnankin asettamat vaatimukset täyttyvät. (Valvomo, suunnit-  
telun periaatteet ja käytännöt. 2010, 170.)

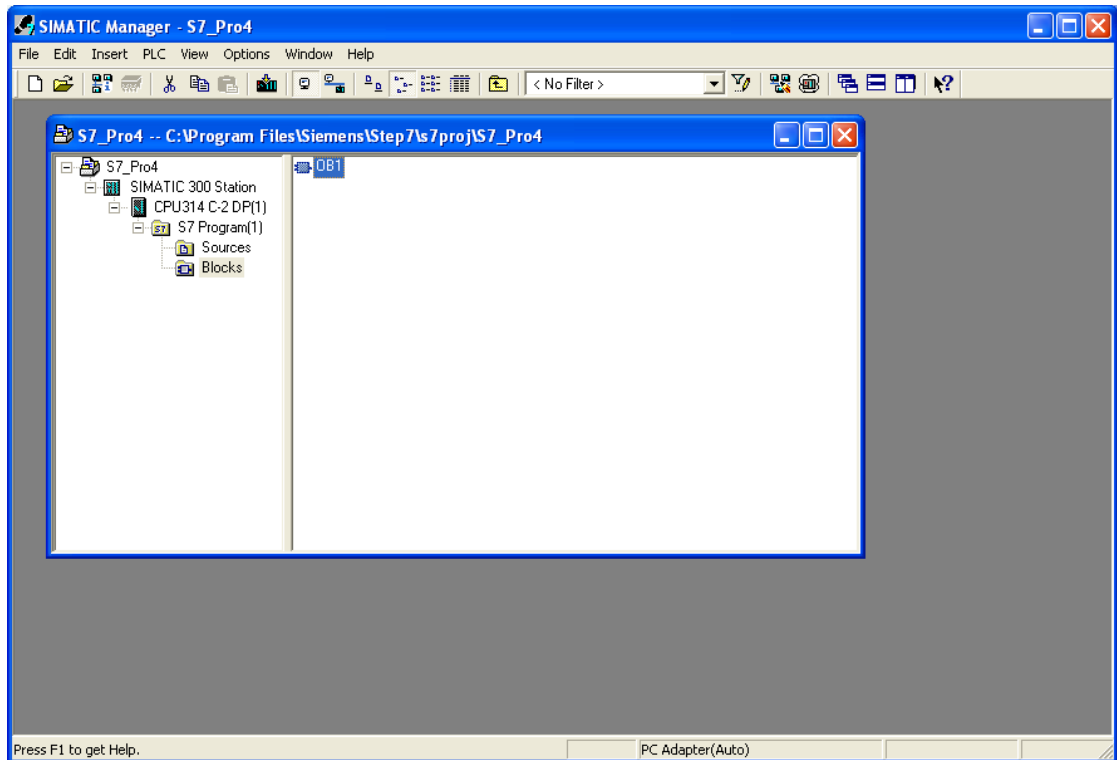


KUVIO 1 Automaation elinkaarimalli (Laatu automaatiassa, parhaat käytännöt. 2001, 17.)

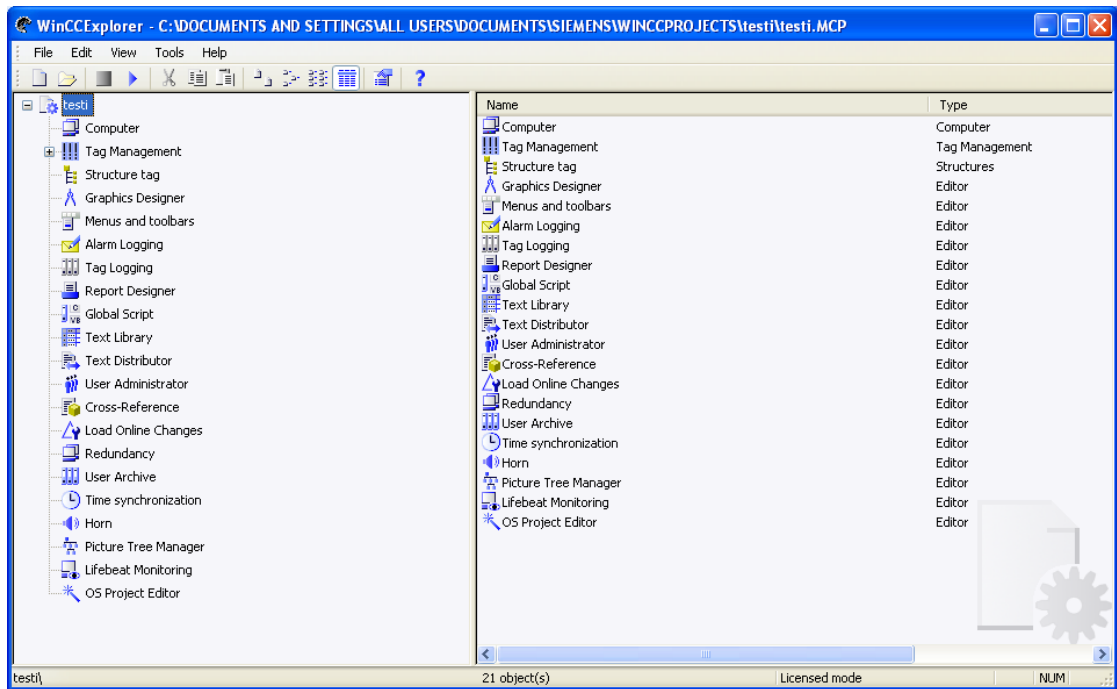
## 2.2. TIA Portal

TIA Portal on Siemensin ns. uuden sukupolven ohjelmointityökalu. Lyhenne TIA muodostuu sanoista Totally Integrated Automation joka tarkoittaa eri sovellustyökalujen integroimista samaan pakettiin. TIA Portal sisältää työkalut Siemensin uusimpien logiikoiden ohjelmointiin, näyttöjen visualisointiin ja turvatekniikkaan. (TIA Portal, 2012.)

Käytännössä TIA Portal yhdistää Siemensin kaksi edellisen sukupolven suunnittelu-  
työkalua: Step7 ja WinCC. Step7 on logiikkaohjelmointiin käytetty perustyökalu (ks.  
kuvio 2) ja WinCC taas puolestaan on HMI –suunnitteluun tarkoitettu työkalu. (ks. ku-  
vio 3)



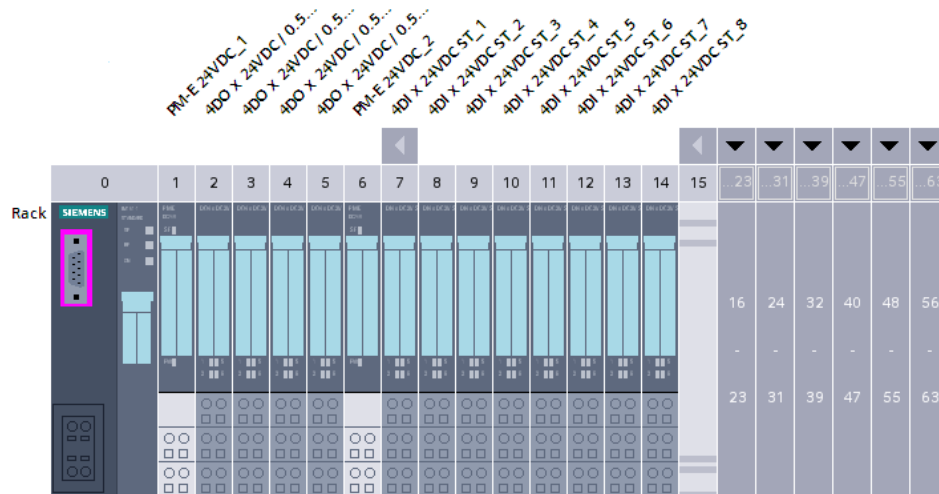
KUVIO 2 STEP7 -ohjelmointitökalu



KUVIO 3 WinCC -ohjelmointityökalu

Vanhoihin työkaluihin verrattuna TIA Portalin etuja ovat helppokäyttöisyys, helposti opittavuus sekä suunnittelun tehokkuus. Ohjelmiston merkittävimpiä uudistuksina

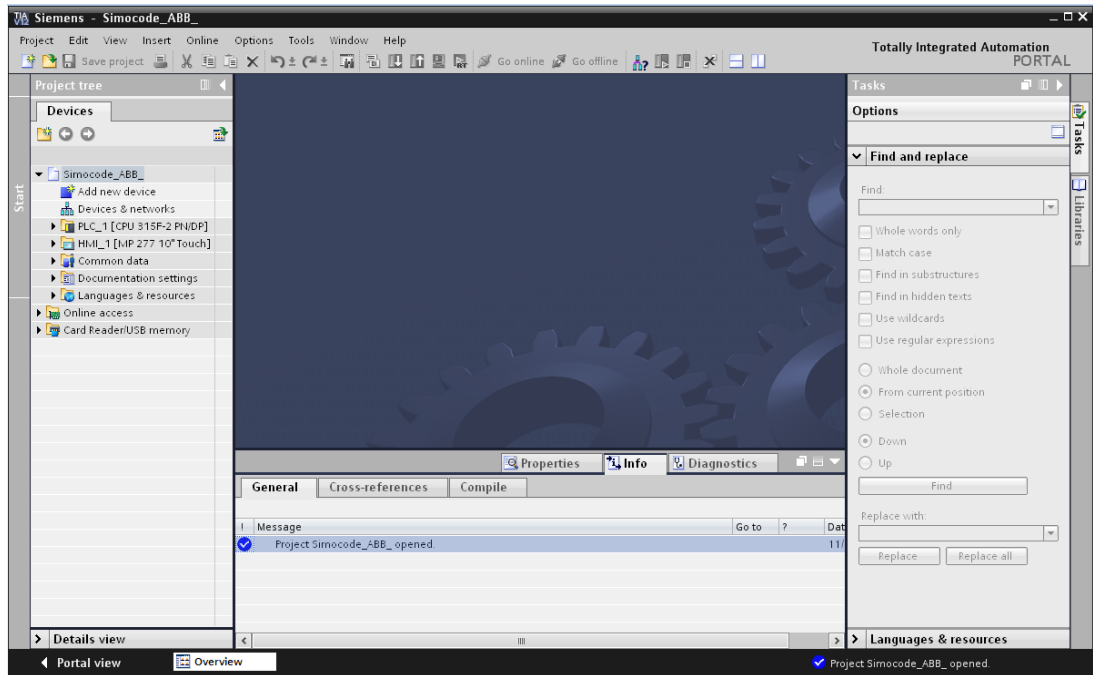
ovat drag/drop -ominaisuus sovellus- ja näyttöeditoreiden välillä sekä valokuvamainen HW-konfiguraatio (ks. kuvio 4.), josta näkee järjestelmän fyysisen rakenteen. (TIA Portal, 2012.) Drag/drop –ominaisuus toimii myös täysin erillisten projektien välillä, joten esimerkiksi ohjelmalohkojen kopiointi toisesta projektista helpottuu huomattavasti.



KUVIO 4 Valokuvamainen HW-konfiguraatio

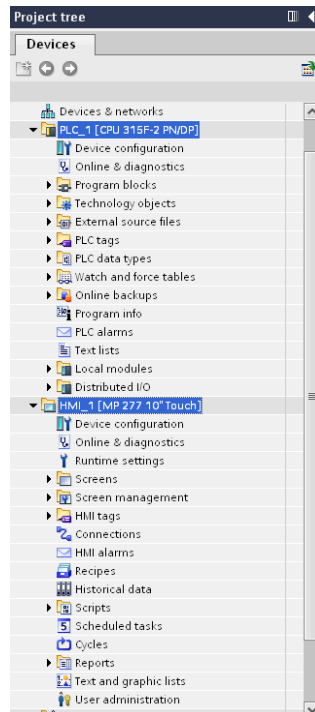
TIA Portal koostuu yhdestä pääikkunasta, joka pitää sisällään kaikki suunnitteeseen tarvittavat toiminnot. Toisin kuin edeltäjänsä, se avaa kaikki erilliset editorit myös tähän samaan ikkunaan.

Pääikkuna koostuu karkeasti jaotellen neljästä osasta: projektipuu vasemmalla, työkalut ja toiminnot oikealla, asetukset ja inforuutu alaosassa, sekä editoritila keskellä. (ks. kuvio 5) Ikkunan sisältö on vapaasti järjesteltävissä uudelleen ja editoritilan voi myös tarvittaessa eriyttää pääikkunasta vaikkapa toiselle näytölle. Editoritila on myös mahdollista jakaa useaan osaan split editor space –toiminnolla. Tällöin siihen saadaan esimerkiksi PLC:n sovellus sekä HMI-näyttö samaan tilaan.



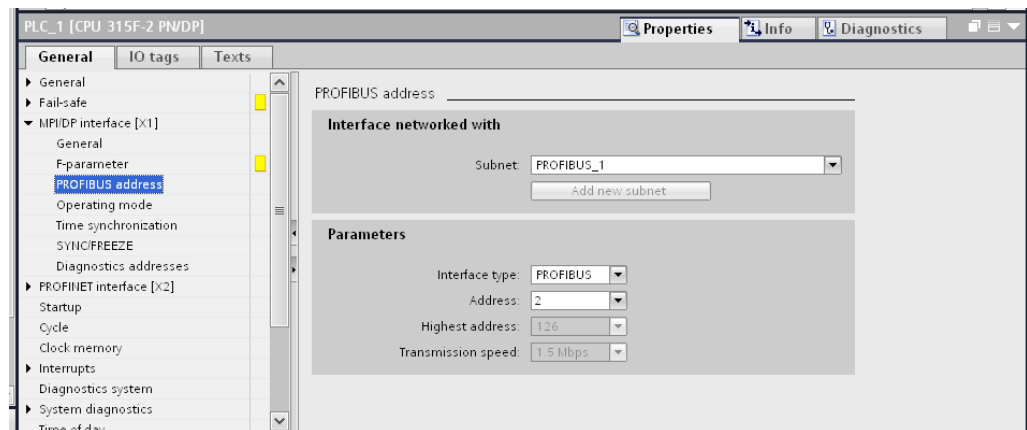
*KUVIO 5 TIA Portal pääikkuna*

Projektipuu on pääikkunan ainoa osa, joka pysyy aina muuttumattomana. Siinä näkyvät projektin sisältämät PLC:t ja näyttöpaneelit, sekä kaikki projektin hallintaan liittyvät valikot. (ks. kuvio 6)



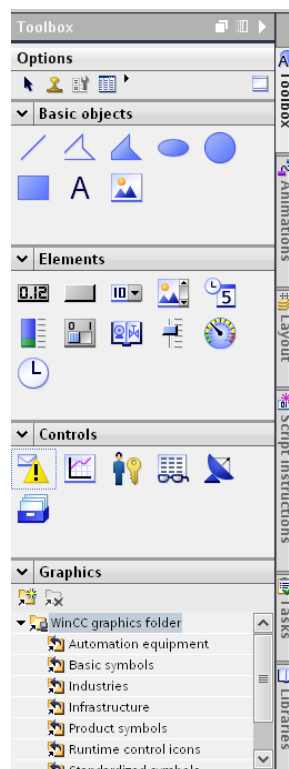
*KUVIO 6 Projektipuun rakenne*

Alaosassa sijaitseva ikkuna sisältää erillisten objektien asetusvalikon sekä yleiset inforuudut mm. virheilmoituksista. Ikkunan sisältö muuttuu käynnissä olevan toiminnon mukaan. Aktivoidessa jokin objekti pääikkunalta, aukeaa sen asetukset alaikkunaan. (ks. kuvio 7) Jos järjestelmästä aktivoidaan esimerkiksi sovelluksen lataus PLC:lle tai näyttöpaneelille, avautuu käänösprosessin inforuutu alaikkunaan asetuksen tilalle.



KUVIO 7 PLC:n asetusruutu

Oikeassa reunassa sijaitseva ikkuna sisältää erilaisia työkaluja ja kirjastoja avoinna olevasta editorista riippuen. Esimerkiksi näyttöeditorin ollessa avoinna se sisältää erilaisia piirtotyökaluja sekä symbolikirjastoja. (ks. kuvio 8)



KUVIO 8 Näyttöeditorin työkaluikkuna

TIA Portal on vielä melko uusi työkalu ja se sisältää myös vikoja. Yleinen kaatuilu on yksi näistä. Se vaivaa pääasiallisesti käynnös vaiheessa tai näyttöeditorissa. Tällöin sovellus sulkeutuu äkillisesti ja kaikki tallentamattomat muutokset katoavat. Toinen iso ongelma on online-tila. PLC:tä monitoroidessa online –tilassa järjestelmä hidastuu merkittävästi ja hidastuminen pahenee mitä pidempään monitorointi kestää. Lopulta sovellus täytyy käynnistää uudelleen, jotta työskentely sujuisi taas normaalisti. TIA Portalin kehitys jatkuu kuitenkin koko ajan, ja uudet päivitykset sisältävät korjausten ohella aina uusia ominaisuuksia.

### **2.3. Simocode pro**

Simocode pro on Siemensin modulaarinen ohjausjärjestelmä pienjännitemoottoreille. Nimi muodostuu sanoista SIRIUS Motor Management and Control Devices. Se sisältää kaikki moottorin suojaamisen, valvontaan ja ohjaamiseen tarvittavat toiminnot. Simocoden etuja perinteisiin ohjaustapoihin verrattuna ovat mm. pienemmät suunnittelu- ja ylläpitokustannukset, väyläliityntä, diagnostiikka sekä monipuoliset suojaustoiminnot. Laitteita löytyy kahta eri tasoa: pro C sekä pro V. (Simocode, 2013.)

Pro C on malleista yksinkertaisempi ja sisältää vain suorakäynnistimen ja suunnanvaihtokäynnistimen sekä neljä binäärituloa ja kolme relelähtöä. Myös binäärinen termistoriliitäntä löytyy. Pro V taas sisältää edellä mainittujen ominaisuuksien lisäksi tähti-kolmiokäynnistimen, moninopeuksisten moottorien käynnistimen, venttiilien ohjausmahdollisuuden sekä pehmokäynnistimen etukoje –toiminnon. Lisäksi pro V –yksikköön on saatavana erilaisia laajennusyksiköitä, kuten moottorin laakerien lämpötilan mittaus. Molemmat mallit sisältävät profibus dp –liitännän ja pro V mallia saa myös profinet –väyläliitynnällä varustettuna. (Simocode, 2013.)

## **2.4. Taajuusmuuttaja (ABB)**

### **2.4.1. Yleistä**

Taajuusmuuttaja on laite jolla pystytään säätelemään moottorin pyörimisnopeutta sekä vääntömomenttia. Merkittävimpänä etuna taajuusmuuttajassa on täysin portaaton nopeudensäätö. Sen avulla taajuusmuuttajalla pystytään toteuttamaan esimerkiksi hyvin tarkkoja virtauksen säätöjä. Sillä saadaan myös merkittävät energiasäästöt verrattuna kaksinopeusmoottoreihin tai esimerkiksi sammuttamalla tavallinen suorakäynnistysmoottori väliajoin. Taajuusmuuttajissa on myös pitkälle kehittyneet seuranta- ja turvaominaisuudet. Erilaiset vikatilanteet havaitaan nopeasti ja näin estetään mahdolliset laiterikot. (ABB Taajuusmuuttajat, 2008.)

### **2.4.2. ABB F-sarjan väyläsovitin**

ABB F-sarjan väyläsovitin on tarkoitettu nimensä mukaisesti liittämään taajuusmuuttaja kenttäväylään. Sovitin hoitaa liikennöinnin taajuusmuuttajan ja ohjausjärjestelmän välillä siirtäen ohjaus- ja tilaviestit sekä parametrit. F-sarjan sovitin toimii monessa ACS-sarjan taajuusmuuttajassa ja sen saa kaikille yleisimmille väylätyypeille. Se osaa myös automaattisesti tunnistaa joitakin parametreja, kuten väylän nopeus sekä sanomatyyppi. (ABB F-sovitin, 2013)

## **2.5. Kenttäväylä**

### **2.5.1. Yleistä**

Kenttäväyläteknikat lähtivät kehittymään 80-luvun puolivälin jälkeen. Ne ovat nimensä mukaan väylätyyppisiä. Tiedonsiirto on täysin digitaalista ja kaksisuuntaista. (Mäkelä, M. 2005.) Kenttäväylä yhdistää kentällä olevat laitteet kuten mittaukset ja moottorinohjaukset ohjausjärjestelmän kanssa samaan verkkoon. Se sallii älykkäät, omalla prosessorilla varustetut kenttälaitteet ja näin esimerkiksi mittausten tarkkuus, konfiguroitavuus ja vikadiagnostiikka paranevat huomattavasti. Väyläteknikka vähentää myös huomattavasti kaapeloinnin tarvetta ja samalla myös kustannukset laskevat. (Automaation digitaaliset kenttäväylät, nd)

### 2.5.2. Profibus

Profibus -väylän kehitys alkoi 80-luvun lopulla Saksassa hallituksen sekä alan johtavien yritysten painostuksesta. Tavoitteena oli kehittää sarjaliikennepohjainen standardisoitu kenttäväylä. Profibus DP on nykyisin laajalti käytössä oleva protokolla ja se noudattaa avointen järjestelmien OSI -mallia. Malli koostuu seitsemästä kerroksesta josta Profibus DP käyttää tasoja yksi ja kaksi. (Profibus, 2009) Taso yksi eli fyysinen kerros määrittää nimensä mukaan kaikki fyysiset asiat kuten esimerkiksi tiedonsiirtotavan ja nopeuden. Taso kaksi eli siirtoyhteyseros hoitaa kaikki yhteyteen liittyvät asiat kuten sen luomisen, purkamisen sekä virheiden korjaamisen. (OSI-malli, 2002) Profibus PA -väylä on kuten DP, mutta se sisältää tehonsyötön kenttälaitteille samassa kaapelissa. DP ja PA -väylät voidaan yhdistää helposti ns. segment couplerilla. (Weigmann, J., Kilian, G. 2000, 15.)

Profibus DP -protokollasta on nykyään käytössä kolme eri alaluokka: DPV0, DPV1 ja DPV2, joista kahta ensimmäistä käytetään Master-Slave -kommunikointiin. DPV0 ja DPV1 eroavat toisistaan ainoastaan siten, että DPV1 pystyy syklisen datankäsittelyn lisäksi asykliseen datankäsittelyyn sekä hälytysten käsittelyyn. (Profibus DP, Introduction, 2010)

Profibus -segmentin maksimipituus riippuu käytettävästä tiedonsiirtonopeudesta. Nopeuden voi valita väliltä 9,6 ... 12000 kbit/sec, jolloin maksimipituus muuttuu välillä 1000 ... 100m. Eli mitä suurempi nopeus, sen lyhyempi segmentti voi olla. Yhdessä Profibus DP -segmentissä voi olla maksimissaan 32 laitetta. Segmentit voidaan liittää toisiinsa toistimilla. Toistimia voi olla maksimissaan 9 kappaletta sarjassa. Näin ollen toistimia käyttäen koko väylän maksimipituus on kymmenkertainen yhden segmentin pituuteen nähden. Jokainen segmentin molemmissa päissä tulee olla päätevastus. (Weigmann, J., Kilian, G. 2000, 27.) Yhdessä väylässä voi olla maksimissaan 127 laitetta. Jokaisella laitteella on oma numeerinen osoite jolla se tunnistautuu väylässä. Osoitteet on jaettu väliltä 0 – 126. (Weigmann, J., Kilian, G. 2000, 31.)



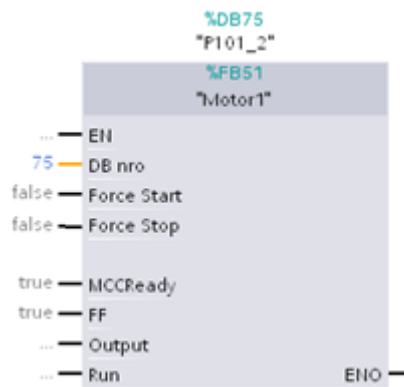
## 3. TYÖN TOTEUTTAMINEN

### 3.1. Toteutusprosessi

Opinnäytetyön toteutus alkoi tutustumalla Simocode- moottorinohjaimen sekä ABB:n taajuusmuuttajan tiedonsiirtoperiaatteisiin. Sen jälkeen määritettiin mitä tietoa halutaan välittää järjestelmän sekä laitteen välillä. Molemmat moottorinohjaimet tarjoavat paljon erilaista seuranta- ja diagnostiikkadataa joten niiden sovittaminen vanhaan moottoritoimilohkoon oli myös mietittävä tarkkaan. Näiden tietojen perusteella varsinaisen tyyppilohkon rakentaminen saattoi alkaa. Seuraava vaihe oli HMI-objektin suunnittelu ja toteutus sen perusteella mitä informaatiota näytöllä halutaan näkyvän. Kun sovelluspuolen toteutus oli lohkojen ja näyttöobjektien osalta valmis, alkoi testausvaihe. Testausvaihe koostui kahdesta eri osasta: Testilaitteiston toimintakuntoon saattamisesta ja parametroinnista sekä sen jälkeen lohkojen testauksesta valmiissa järjestelmässä. Simocoden testaus tapahtui JEEC Oy:n toimistolla olevalla demosalkulla ja taajuusmuuttajan testaus eräässä projektissa, jossa se myös otettiin heti käyttöön. Lohkojen toimintaa testattiin kaikilla mahdollisilla käyttötavoilla ja myös erilaisia vikatilanteita kuten väylän vikaantumista simuloitiin. Työn päätteeksi valmiit toimivaksi todetut lohkot sovitettiin JEEC Oy:n omaan kirjastoon ja niistä tehtiin tarvittava dokumentaatio käyttämisen tueksi.

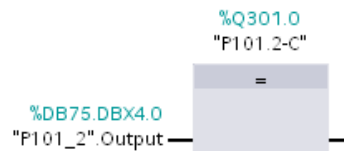
### 3.2. Perusmoottorin tyyppilohko ja sen hyödyntäminen

Perusmoottorin toimilohko sisältää kaiken moottorinohjaukseen tarvittavan logiikan: ohjauksen, tilatiedot sekä lukitusten ja hälytysten käsittelyn. Jokaisella moottorilla on toimilohkoon liittyvä instanssi-DB, joka sisältää kaikki lohkon käyttämät tiedot. Uudet, väyläkommunikoinnin sisältävät lohkot on tarkoitus tehdä vanhan perusmoottorilohkon pohjalta. Vanhasta lohkoista ohjaus saatiin binäärisessä muodossa suoraan output –parametristä. (ks. kuvio 9.)



KUVIO 9 Perusmoottorin toimilohko

Tästä ohjaussignaali piti erikseen siirtää moottorilohkon DB:stä haluttuun lähtöön mistä moottoria ohjattiin. (ks. kuvio 10.)

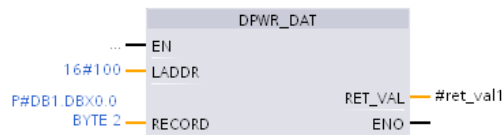


KUVIO 10 Binäärisen ohjauksen siirto lähtöön

Älykkäät moottorinohjaimet kuten Simocode sekä taajuusmuuttajat vaativat ohjaukseen yleensä 16 bittisen ohjaussanan, joten pelkkä suora binäärinen ohjaus ei riitä. Samoin tilatiedot saapuvat väylästä sanamuotoisina. Tästä syystä perusmoottorin lohko ei sellaisenaan kelpaa hyvin väyläohjaukseen vaan se vaatii sisälleen oman osion väyläkommunikoinnille missä lohkon binääriset signaalit muutetaan tavuiksi tai sanoiksi käytettävän moottorinohjaimen mukaan ja lähetetään väylään. Väylästä pyydetty tilatiedot taas puretaan tavu- tai sanamuodosta binäärisiksi signaaleiksi joita moottorilohkon toiminnot sitten hyödyntävät.

### 3.3. Simocoden tyyppilohko

Simocoden ohjaus tapahtuu siirtämällä ohjaussana oikeaan lähtöön käyttäen DPWR\_DAT –lohkoa. Lohkoon määritellään haluttu periferiaosoite HEX-muodossa LADDR –tuloon sekä siirrettävän tiedon sijainti sekä pituus RECORD –tuloon. (ks. kuvio 11.)



*KUVIO 11 Ohjaussanan lähetys Simocodelle*

Simocodeen on määritelty tietynlainen ohjaussana oletuksena (ks. kuvio 12), mutta sekin on täysin parametroitavissa uudelleen Simocode ES –ohjelman avulla. Oletuksena Simocode ei tarvitsisi toimiakseen kuin yhden tavun mittaisen ”ohjaussanan”, mutta mahdollisten lisätoiminnallisuuksien vuoksi sille lähetetään kokonainen 16 bit-tinen sana. Sana muodostetaan moottorin datablokkiin kuvion 6 mukaisesti.

Byte.Bit	Specification	Presetting (also see parameters)	Type
0.0	Cyclic receive - Bit 0.0	Control station - PLC/DCS [DP] ON<	Bit
0.1	Cyclic receive - Bit 0.1	Control station - PLC/DCS [DP] OFF	Bit
0.2	Cyclic receive - Bit 0.2	Control station - PLC/DCS [DP] ON>	Bit
0.3	Cyclic receive - Bit 0.3	Test 1	Bit
0.4	Cyclic receive - Bit 0.4	Motor protection - Emergency start	Bit
0.5	Cyclic receive - Bit 0.5	Mode selector S1	Bit
0.6	Cyclic receive - Bit 0.6	Reset 1	Bit
0.7	Cyclic receive - Bit 0.7	Not assigned	Bit
1.0	Cyclic receive - Bit 1.0	Not assigned	Bit
1.1	Cyclic receive - Bit 1.1	Not assigned	Bit
1.2	Cyclic receive - Bit 1.2	Not assigned	Bit
1.3	Cyclic receive - Bit 1.3	Not assigned	Bit
1.4	Cyclic receive - Bit 1.4	Not assigned	Bit
1.5	Cyclic receive - Bit 1.5	Not assigned	Bit
1.6	Cyclic receive - Bit 1.6	Not assigned	Bit
1.7	Cyclic receive - Bit 1.7	Not assigned	Bit

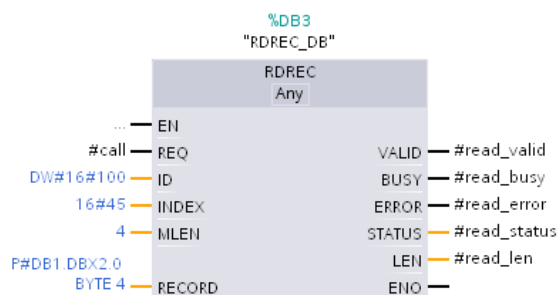
*KUVIO 12 Simocode oletusohjaussanan muodostuminen*

Tilatietojen lukeminen Simocodelta tapahtuu hieman eri lailla kuin ohjauksen kirjoittaminen. Simocode moottorinohjain sisältää erityisen Data Records –alueen minne on jaoteltu erilaista dataa osastoittain. Jokaisella osastolla on oma numero minkä avulla tarvittava data saadaan luettua.(ks. kuvio 13.)

Data record No.	Description	Read/write
1	S7 system diagnostics	Read
63	Recording analog values	Read
67	Process image of the outputs	Read
69	Process image of the inputs	Read
72	Fault memory	Read
92	Device diagnostics (faults, warnings, events)	Read
94	Measured values	Read
95	Service/statistical data	Read/write
130	Basic unit parameter 1 (BU1 BU2)	Read/write
131	Basic unit parameter 2 (BU1 BU2)	Read/write
132	Extended device parameter 1 (BU2)	Read/write
133	Extended device parameter 2 (BU2)	Read/write
139	Marking	Read/write
160	Communication parameters	Read/write
165	Designation	Read/write
202	Acyclic receive	Read/write
203	Acyclic send	Read
224	Password protection	Write

KUVIO 13 Simocode Data Records

Data Records –alueelta lukeminen tapahtuu RDREC –lohkolla. Simocoden periferiaosoite tulee lohkon ID –tuloon HEX –muodossa, luettavan alueen numero INDEX –tuloon HEX –muodossa, luettavan alueen maksimipituus tavuina INDEX -tuloon INT –muodossa sekä tietojen tallennuspaikka ja pituus RECORD -tuloon. RDREC –lohkoa kutsutaan sen REQ-tulosta. (ks. kuvio 14.) Kaikki tarvittavat tiedot tallennetaan moottorin datablokkiin, josta niitä voi lukea tarpeen vaatiessa joko yksittäisinä bitteinä tai myös isompina paketteina.



KUVIO 14 Simocode Data Record –alueen lukeminen

Simocoden lohkon tulee perusmoottoripohjaan verrattuna kaksi uutta ulkoista tuloa, periferiatulo-osoitteen sekä periferiälähtöosoitteen määrittely. Näiden avulla saadaan lähetys- sekä lukutoiminnot toimimaan oikean moottorin kanssa. Moottorilohkon ohjaussignaali kytketään datablokissa oikeisiin bitteihin. Vanhasta perusmoottorin lohkoista puuttuville biteille, kuten suunnanvaihdolle, resetille sekä modelle luodaan uudet muistipaikat moottorin datablokkiin ja ne tulevat näkymään myös HMI – piirinäytössä. Näytöllä näkyvinä tilatietoina Data Record –alueilta tuodaan moottorin käyntitieto, moottorin virran analogiarvo, sekä moottorivian ja väylävian ilmaisevat bitit. Moottorin datablokkiin tuodaan lisäksi myös System Diagnostics- sekä Device Diagnostics-alueen kaikki vikatiedot jos jossain projektissa tarvitaan tarkempaa diagnostiikkatietoa.

### **3.4. ABB:n taajuusmuuttajan tyyppilohko**

Taajuusmuuttajan tyyppilohkossa käytetään myös perusmoottorin lohkoa pohjana. Uuteen lohkon tulee Simocoden lohkon tapaan ulkoiset parametrit periferiatulo-osoitteelle ja periferiälähtöosoitteelle joilla liikenne kohdennetaan oikealle väylälaitteelle. Taajuusmuuttajan data lähetetään myös DPWR\_DAT –lohkoa käyttäen, kuten Simocodellakin (ks. kuvio 11.) Taajuusmuuttajan lähetettävä data muodostuu 16 bittisestä ohjaussanasta ja 16 bittisestä INT-tyyppisestä taajuusohjeesta ( $\pm 16384$  dec). Taajuusmuuttajalta lukeminen tapahtuu taas DPRD\_DAT –lohkoa käyttäen suoraan periferiaosoitteesta, toisin kuin Simocodella. Taajuusmuuttajalta saatava data muodostuu 16 bittisestä tilasanasta ja 16 bittisestä INT-tyyppisestä nopeustiedosta ( $\pm 16384$  dec). (ks. kuvio 15.) Taajuusmuuttajallakin sekä ohjaus että tilatiedot ovat täysin parametroitavissa ja edellä esitetyt mallit ovat oletusasetuksia. Nopeustiedon lisäksi voidaan saada esimerkiksi 32 bittinen REAL-tyyppinen luku moottorin tehosta, virrasta tai vaikkapa momentista. Datasanomien pituus riippuu käytettävästä PPO-tyypistä ja suurimmillaan se voi olla 12 x 16 bittiä (ks. kuvio 16.).

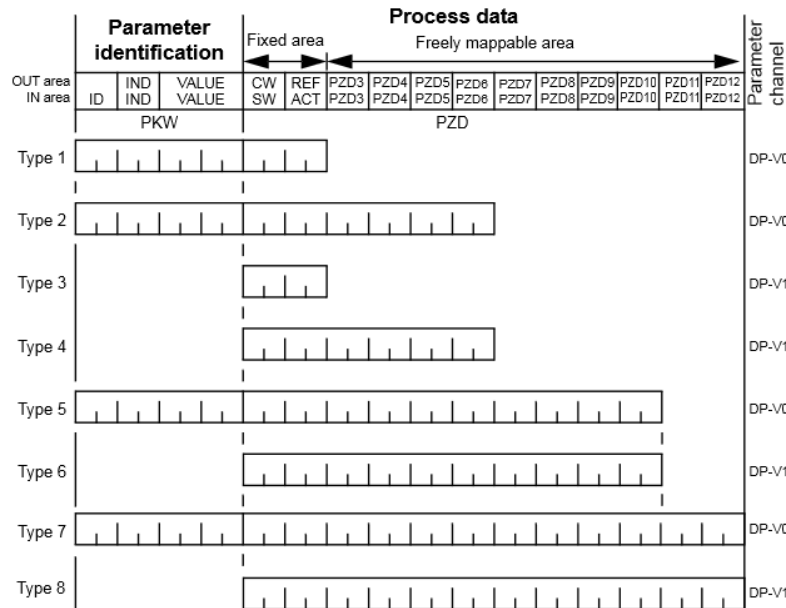
## Control word contents

Bit	Name	Value	STATE/Description
0	OFF1_CONTROL	1	Proceed to <b>READY TO OPERATE</b> .
		0	Stop along currently active deceleration ramp. Proceed to <b>OFF1 ACTIVE</b> ; proceed to <b>READY TO SWITCH ON</b> unless other interlocks (OFF2, OFF3) are active.
1	OFF2_CONTROL	1	Continue operation (OFF2 inactive).
		0	Emergency OFF, coast to stop. Proceed to <b>OFF2 ACTIVE</b> , proceed to <b>SWITCH-ON INHIBITED</b> .
2	OFF3_CONTROL	1	Continue operation (OFF3 inactive).
		0	Emergency stop, stop within time defined by drive parameter. Proceed to <b>OFF3 ACTIVE</b> ; proceed to <b>SWITCH-ON INHIBITED</b> . <b>Warning:</b> Ensure that motor and driven machine can be stopped using this stop mode.
3	INHIBIT_OPERATION	1	Proceed to <b>OPERATION ENABLED</b> . <b>Note:</b> Run enable signal must be active; see drive documentation. If the drive is set to receive the Run enable signal from the fieldbus, this bit activates the signal.
		0	Inhibit operation. Proceed to <b>OPERATION INHIBITED</b> .
4	RAMP_OUT_ZERO	1	Normal operation. Proceed to <b>RAMP FUNCTION GENERATOR: OUTPUT ENABLED</b> .
		0	Force Ramp Function Generator output to zero. Drive ramps to stop (current and DC voltage limits in force).
5	RAMP_HOLD	1	Enable ramp function. Proceed to <b>RAMP FUNCTION GENERATOR: ACCELERATOR ENABLED</b> .
		0	Halt ramping (Ramp Function Generator output held).
6	RAMP_IN_ZERO	1	Normal operation. Proceed to <b>OPERATING</b> . <b>Note:</b> This bit is effective only if the fieldbus interface is set as the source for this signal by drive parameters.
		0	Force Ramp Function Generator input to zero.
7	RESET	0 → 1	Fault reset if an active fault exists. Proceed to <b>SWITCH-ON INHIBITED</b> . <b>Note:</b> This bit is effective only if the fieldbus interface is set as the source for this signal by drive parameters.
		0	Continue normal operation.
8...9	Reserved		
10	REMOTE_CMD	1	Fieldbus control enabled
		0	Control word and reference not getting through to the drive, except for CW bits OFF1, OFF2 and OFF3.
11	EXT_CTRL_LOC	1	Select External Control Location EXT2. Effective if control location parameterized to be selected from fieldbus.
		0	Select External Control Location EXT1. Effective if control location parameterized to be selected from fieldbus.
12...15	Drive-specific (For information, see the drive documentation.)		

## Status word contents

Bit	Name	Value	STATE/Description
0	RDY_ON	1	<b>READY TO SWITCH ON</b>
		0	<b>NOT READY TO SWITCH ON</b>
1	RDY_RUN	1	<b>READY TO OPERATE</b>
		0	<b>OFF1 ACTIVE</b>
2	RDY_REF	1	<b>OPERATION ENABLED</b>
		0	<b>OPERATION INHIBITED</b>
3	TRIPPED	1	<b>FAULT</b>
		0	No fault
4	OFF_2_STA	1	OFF2 inactive
		0	<b>OFF2 ACTIVE</b>
5	OFF_3_STA	1	OFF3 inactive
		0	<b>OFF3 ACTIVE</b> .
6	SWC_ON_INHIB	1	<b>SWITCH-ON INHIBITED</b>
		0	–
7	ALARM	1	Warning/Alarm
		0	No warning/alarm
8	AT_SETPOINT	1	<b>OPERATING</b> . Actual value equals reference (= is within tolerance limits, ie, in speed control, speed error is 10% max. of nominal motor speed).
		0	Actual value differs from reference (= is outside tolerance limits).
9	REMOTE	1	Drive control location: REMOTE (EXT1 or EXT2)
		0	Drive control location: LOCAL
10	ABOVE_LIMIT	1	Actual frequency or speed equals or exceeds supervision limit (set by drive parameter). Valid in both directions of rotation.
		0	Actual frequency or speed within supervision limit
11	EXT_CTRL_LOC	1	External Control Location EXT2 selected
		0	External Control Location EXT1 selected
12	EXT_RUN_ENABLE	1	External Run Enable signal received
		0	No External Run Enable signal received
13...14	Drive-specific (For information, see the drive documentation.)		
15	FBA_ERROR	1	Communication error detected by fieldbus adapter module
		0	Fieldbus adapter communication OK

## ■ PPO types



KUVIO 16 ABB Datasanomatyypien pituudet

Ohjaus- ja tilasalle, nopeusohjeelle, todelliselle nopeudelle sekä kaikelle muulle datalle luodaan moottorin datablokkiin paikka mistä niiden luku ja kirjoitus tapahtuu.

### 3.5. HMI-objektit

Molemmille ohjaintyypeille tarvittiin HMI-objektit joista kävisi ilmi kaikki oleellinen tieto moottorinohjaimen ja väylän tilasta. Objektien tulisi olla popup –tyyppisiä ikkunoita jotka saisi moottorin piirinäytöstä auki. Niiden tulisi myös olla selkeitä ja kompaktin kokoisia näytöllä. Kaikesta informaatiosta täytyi siis valita vain oleellisimmat moottorin operoinnin ja seurannan kannalta jotta ikkuna pysyisi kompaktina. Varsinainen piirinäyttö sisälsi moottorin käynnistys- ja pysäytyspainikkeet, joten niitä ei tarvinnut sovittaa em. ikkunoihin lainkaan.

Simocoden ohjaustoimintoihin päätyi pyörimissuunnan muuttaminen, operointitilan muuttaminen sekä resetointi. Tilasta taas kertoo virtinäyttö, running/stopped –näyttö sekä erilaiset vikatyypit ja –koodit. (ks. kuvio 17.)



KUVIO 17 Simocode HMI-näyttö

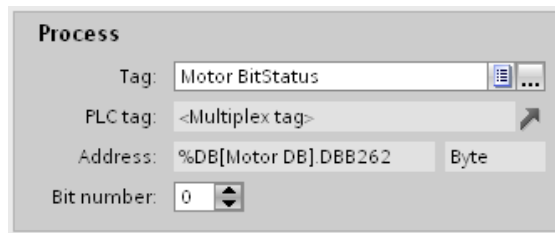
Taajuusmuuttajan ikkuna on hyvin samankaltainen Simocoden ikkunan kanssa muutamaa poikkeusta lukuun ottamatta. Suunnanvaihtotoiminto sekä vikakoodin näyttö jäivät pois koska ne eivät kuuluneet taajuusmuuttajan oletusviesteihin. Lisäyksenä näytölle tuli hätäpysäytyspainike sekä moottorin pyörimisnopeus ja virta. (ks. kuvio 18.)



KUVIO 18 ABB HMI-näyttö

HMI-objektit kirjoittavat ja lukevat tietoa suoraan logiikalla olevista datablokeista. Yhdeksi ongelmaksi muodostui tiedon osoittaminen moottorin mukaan vaihtuvan näytön tageista juuri oikean moottorin datablokkiin. Tähän löytyi ratkaisu TIA Portalin multiplex –tyyppisestä tagista jolla voidaan osoittaa yhden lisämuuttujan avulla oikeaan datablokkiin. (ks. kuvio 19.)





*KUVIO 19 Multiplex-tagin määrittely*

Moottorin piirinäyttöä aukaistaessa operointipaneelilta tämän kyseisen moottorin datablokin numero tallentuu tilapäisesti Motor DB –nimiseen muuttujaan. Piirinäytöllä jokin tietty, vaikkapa operointiin liittyvä tagi ns. multiplexataan tällä Motor DB –muuttujalla. Kun tagia operoidaan, tieto välittyy Motor DB –muuttujan osoittamaan datablockiin.

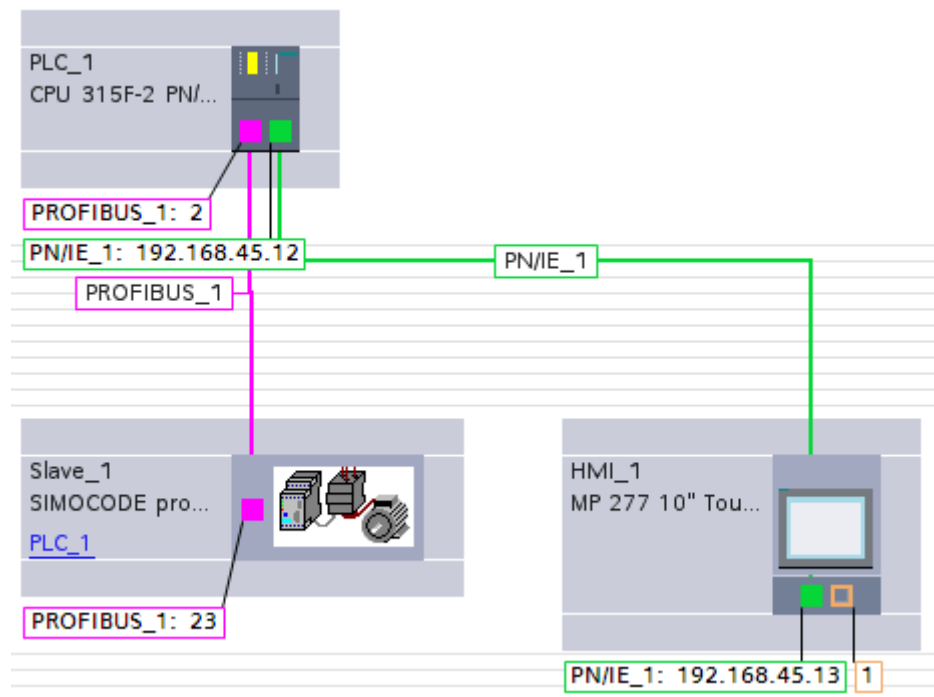
### 3.6. Testaus

Testaus alkoi tarvittavan laitteiston hankkimisella ja kokoamisella. Työssä käytetyt Siemens 315-F PN/DP –logiikan ja MP277 –operointipaneelin sain lainaksi Jyväskylän ammattikorkeakoululta, Simocode demosalkun suoraan Siemensiltä. Logiikka, paneeli sekä PG toimivat samassa ethernet-verkossa. Simocode kytkeytyi logiikkaan profibus-väylän kautta. (ks. kuvio 20 ja 21.)

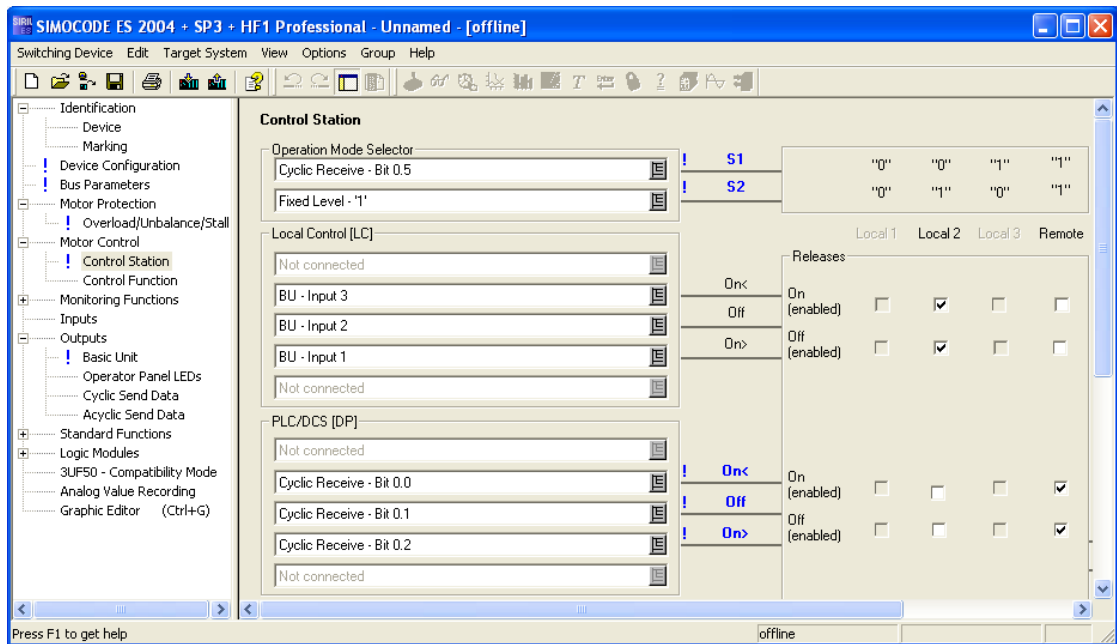
Ensimmäisenä testaukseen pääsi Simocoden tyyppilohko. Testausta varten logiikkaan ladattiin sovellus joka sisälsi kaiken tarvittavan HW-konfiguraatiosta itse lohkoihin. Operointipaneelille ladattiin myös yksi piirinäyttö. Simocodeen päivitettiin suunnanvaihdolla varustetun suorakäynnistimen uudet parametrit käyttäen Simocode ES –ohjelmaa (ks. kuvio 22.) ja tietokoneen sarjaporttiin kytkeytyvää ohjelmointikaapelia. Laitteiston ollessa käyttökunnossa Simocodella testattiin kaikki mahdolliset käyttötilanteet ja vikatilanteita siinä määrin mikä tällä laitteistolla oli mahdollista, mm. väylä- vika ja ylikuormitus. Taajuusmuuttajan lohkoa testattiin erään projektin käyttöön- otossa vastaavanlaisilla kokeilla ja se todettiin myös toimivaksi.



KUVIO 20 Simocode testiympäristö



KUVIO 21 Testilaitteiston HW-konfiguraatio



*KUVIO 22 Simocode ES -parametrinti ohjelma*

## 4. TULOKSET

Opinnäytetyön tuloksena syntyneet tyyppilohkot otettiin heti käyttöön uusissa projekteissa. Opinnäytetyön valmistumishetkellä taajuusmuuttajan toimilohko oli jo käytössä toimivassa laitoksessa. Käyttöönoton perusteella se todettiin toimivaksi ratkaisuksi ja helpoksi parametroida. Simocoden lohko todettiin myös toimivaksi testausvaiheessa monipuolisilla, oikeita tilanteita simuloivilla kokeilla, mutta sen todellinen käytettävyys ja toiminta käyvät ilmi vasta sitten kun se on käyttöönotettu jossain laitteistossa. Monipuolisempaa tietoa lohkojen toimivuudesta ja käytettävyydestä saadaan vasta pitkällä aikajänteellä.

Väyläkommunikaation sisältävät lohkot vähensivät sovellussuunnittelijan taakkaa moottorien osalta ja niin myös suunnittelu tehostui. Lohkojen myötä myös kaikki älykkäiden moottorinohjainten tarjoama monipuolinen informaatio tuli helpommin saataville ja paremmin hyödynnettyä, kun normaalisti laitteilta olisi luettu vain kaikki välttämättömimmät tiedot ja spesifimpi data olisi saattanut jäädä huomiotta. Moottorinohjainten tarjoamaa dataa perus-tilatietojen lisäksi ovat esimerkiksi moottorin virta tai momentti. Laitteiden vikadiagnostiikka tehostui myös ja nyt väylästä saadaan muutaman vikabitin sijaan hyvinkin tarkkoja tietoja mistä eri vikatilat johtuvat. Tätä vikadataa voi hyödyntää esimerkiksi kehitettäessä parempia suojausjärjestelmiä moottorin sovelluksessa.

Lohkot sisälsivät kaiken tarvittavan perustoiminnan moottorinohjaukseen väylän kautta, mutta kehityskohteitakin jäi. Monipuolisten diagnostiikkatietojen täydellinen hyödyntäminen jäi kokonaan pois. Tiedot kyllä luetaan moottorin datablokkiin säilöön mutta järjestelmä ei varsinaisesti käytä niitä mihinkään. Kyseisten tietojen hyödyntäminen on tulevaisuudessa helppoa, koska lähes kaikki tieto on olemassa moottorin datablokissa. Ohjausviestien lähettäminen ns. raakadatana HMI-objektin kautta jätettiin lopullisesta versiosta pois, koska sillä ei olisi saavutettu mitään merkittävää hyötyä normaalissa käyttö- tai edes käyttöönottilanteessa. Siitä ominaisuudesta olisi saattanut olla enemmän haittaa jos joku asiaan perehtymätön olisi päässyt käskyjä lähettämään. Lohkojen runko on toteutettu niin, että sen kehittäminen ja monipuolistaminen olisi tulevaisuudessa mahdollisimman helppoa. Ohjelman rakenne on selkeä ja yksinkertainen sekä hyvin kommentoitu. Esimerkiksi lohkojen muuttaminen yhteensopiviksi muiden väylätyyppien, kuten ProfiNet:in kanssa, on mahdollista hyvin pienillä muutoksilla.

## 5. POHDINTA

Tämä opinnäytetyö toimi hyvänä porttina sovellussuunnittelijan arkeen ja toimintatapoihin. Se tarjosi minulle mahdollisuuden oppia paljon uutta erilaisista laitteista ja tekniikasta, sovellustyökaluista sekä ihan tavallisista käytänteistä sovellussuunnittelussa. Sain samalla tietenkin myös arvokasta työkokemusta tulevaisuutta ajatellen.

Siemensin TIA Portal –työkalu tuli erittäin tutuksi opinnäytetyöprosessin aikana. Sitä pidän myös ehkä yhtenä tärkeimmistä asioista mitä olen tänä aikana oppinut. Se on erittäin yleinen työkalu sovellussuunnittelussa ja sen osaamista arvostetaan. Tutuksi tulivat myös Simocode, ABB taajuusmuuttajat sekä väylätekniikka yleensä. Ne ovat myös tärkeitä asioita joihin tulee törmäämään varmasti tällä alalla. Opinnäytetyöprosessin aikana opin myös sovellussuunnittelijan normaalia työrutiinia.

Mielestäni saavutin hyvin omat henkilökohtaiset tavoitteeni ja myöskin työn toiminnalliset tavoitteet. Lopputulokseen saatiin lähestulkoon kaikki ne toiminnot mitä alussa oli määriteltykin. Kaikki lohkot läpäisivät niille suoritettut testit täysin ja toimeksiantaja oli tyytyväinen valmiiseen lohkokirjastoon.

## LÄHTEET

JEEC Oy. Kotisivut. 2013. Viitattu 4.10.2013. <http://jeec.fi/>

ABB F-sovitin. 2013. ABB F-series fieldbus adapter modules. Viitattu 25.9.2013

[http://www05.abb.com/global/scot/scot201.nsf/veritydisplay/3ce813b6546db577c1257b470039e13a/\\$file/F-Series fieldbus adapter modules flyer EN 3AUA0000082386 RevB.pdf](http://www05.abb.com/global/scot/scot201.nsf/veritydisplay/3ce813b6546db577c1257b470039e13a/$file/F-Series%20fieldbus%20adapter%20modules%20flyer%20EN%203AUA0000082386%20RevB.pdf)

Automaation digitaaliset kenttäväylät, nd. Tampereen teknillinen yliopisto, automaation digitaaliset kenttäväylät –luento. Viitattu 1.10.2013

[http://www.ac.tut.fi/aci/courses/7601000/pdf/Averkot\\_2\\_fieldbus\\_2p.pdf](http://www.ac.tut.fi/aci/courses/7601000/pdf/Averkot_2_fieldbus_2p.pdf)

ABB Taajuusmuuttajat. 2008. Mikä taajuusmuuttaja on?. Viitattu 25.9.2013

<http://www.abb.fi/cawp/db0003db002698/d5b664f5dd909412c1257291003ef7cc.aspx>

Simocode. 2013. Älykäs SIMOCODE-moottorisuojaus. Viitattu 18.9.2013

[http://www.siemens.fi/fi/industry/teollisuuden tuotteet ja ratkaisut/tuotesivut/pienjannitekojeet/kytkenta suojaus ja ohjaus/simocode alykas moottorisuojaus.htm](http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/pienjannitekojeet/kytkenta_suojaus_ja_ohjaus/simocode_alykas_moottorisuojaus.htm)

TIA Portal. 2012. TIA Portal Suunnittelun uusi aikakausi –site. Viitattu 18.9.2013

[http://www.siemens.fi/pool/products/industry/iadt is/tuotteet/tia\\_portal/tia\\_portal\\_esite.pdf](http://www.siemens.fi/pool/products/industry/iadt_is/tuotteet/tia_portal/tia_portal_esite.pdf)

Profibus. 2009. RTA Inc, Comprehensive Protocol Overview. Viitattu 1.10.2013

<http://www.rtaautomation.com/profibus/>

OSI-malli. 2002. Tampereen teknillisen yliopiston tietotekniikan luento. Viitattu 1.10.2013

<http://www.cs.tut.fi/etaopetus/titepk/luku19/OSI.html>

Mäkelä, M. 2005. Kenttäväylät, johdanto. Luentomateriaali. JAMK

Weigmann, J., Kilian, G. 2000. Decentralization with PROFIBUS-DP. MCD Verlag

Profibus DP, Introduction. 2010. ABB FieldBusPlug Technical description. Viitattu 14.10.2013

[http://www05.abb.com/global/scot/scot209.nsf/veritydisplay/336f4961e2377c3dc12576ea0029d7a4/\\$file/2cdc192001d0209.pdf](http://www05.abb.com/global/scot/scot209.nsf/veritydisplay/336f4961e2377c3dc12576ea0029d7a4/$file/2cdc192001d0209.pdf)

Weigmann, J., Kilian, G. 2000. Decentralization with PROFIBUS-DP. MCD Verlag

Laatu automaatiassa, parhaat käytännöt. 2001. 1.p. Helsinki: Suomen Automaatioseura ry

Valvomo, suunnittelun periaatteet ja käytännöt. 2010.Helsinki: Suomen Automaatioseura ry

## **LIITTEET**

**LIITE 1. Simocode perusparametrit suunnanvaihtokäynnistimelle**

**LIITE 2. FBPA-01 perusparametrit**