

Rauno Kykkänen

Malmigranulaattikuivaimen jakopään servojen ohjaus

Metropolia Ammattikorkeakoulu
Insinööri AMK
Automaatiotekniikka
Opinnäytetyö
10.5.2014

Alkusanat

Kiitokseni Outotecin Risto Huttuselle mahdollisuudesta suorittaa tämä työ. Eplan tuesta kiitän Etteplan Oyj:n Jari Saurasta. Siemens logiikka-ohjelmointituesta kiitän Etteplan Oyj:n Sylvain Moreauta ja Outotecin Mikko Härköstä. Mikolle kiitos myös käyttöönottotuesta.

Kiitokseni myös Markku Inkiselle, joka valvoi työtäni Metropolian toimesta.

Tekijä(t) Otsikko	Rauno Kykkänen Malmigranulaattikuivaimen jakopään servojen ohjaus
Sivumäärä Aika	75 sivua + 19 liitettä 10.5.2014
Tutkinto	Insinööri AMK
Koulutusohjelma	Automaatiotekniikka
Suuntautumisvaihtoehto	Kappaletavara-automaatio
Ohjaaja(t)	Lehtori Markku Inkinen Department manager automation Risto Huttunen
<p>Työn tarkoituksena oli tuottaa servo-ohjausjärjestelmän laitteistosuunnittelu, ohjelmointi ja laitteiston sekä sovellusohjelman testaus asiakkaan konepajalla.</p> <p>Kohdelaitteisto on malmigranulikuivaimen jakopää. Jakopää koostuu purkajaluukusta, jonka kummassakin päässä on servomoottori purkausvaaputusliikkeen toteuttamiseksi sekä purkajaluukun kummallakin reunalla olevista sektoriluukuista, joilla säädetään purettavan materiaalin määrää. Sektoriluukuilla on kummallakin omat servomoottorinsa.</p> <p>Toteutus tehtiin Siemens logiikalla, servo-ohjaimella, servovahvistimilla ja -moottoreilla.</p> <p>Keinoina työn toteuttamiseksi käytettiin työkaluohjelmien käyttöohjeita, kollegoiden tietämyksen hyödyntämistä, Siemensin Suomen tuen palveluita ja konepajalla tehtyjä käytännön testejä.</p> <p>Ohjelmoinneissa hyödynnettiin valmista ohjelmapohjaa yhtenäisemmän ilmeen ja lähestymistavan luomiseksi samoin kuin muissa Outotecin liikkeenohjaussovelluksissa. Tämä idea osoittautuikin toimivaksi ja on kannatettava verrattuna tapaan, jossa jokainen ohjelmoija toteuttaisi vain omaa tapansa.</p> <p>Työn myöhempää käyttömahdollisuutta esimerkiksi tulevassa laitteiston käyttöönotossa loppuasiakkaan luona palvee luku, jossa kuvataan signaalin kulku operaattorilta toimilaitteelle.</p>	
Avainsanat	laitteisto- / ohjelmistosuunnittelu, testaus, servo-ohjaus, paikkasäätö

Author(s) Title	Rauno Kykkänen Metal Granulate Dryer Distributor End Servo Control
Number of Pages Date	75 pages + 19 appendices 10 May 2014
Degree	Bachelor of Engineering
Degree Programme	Automation Engineering (AMK)
Specialisation option	Manufacturing automation
Instructor(s)	Markku Inkinen, Lecturer Risto Huttunen, Department Manager Automation
<p>The objective of this Bachelor's thesis was to produce servo control system hardware design, programming and testing for both at customer's work shop.</p> <p>The target hardware is the ore dryer's distributor. The distributor consists of a distributor hatch with servomotors attached at the both end of the hatch to produce movement for unloading, and of the sector hatches mounted at the both sides of the distributor hatch for controlling the material flow to unload. Both sector hatches have their own servomotors.</p> <p>Implementation was done with Siemens logic, servocontroller, servoamplifiers and motors.</p> <p>As a means to complete this task tool program's user manuals, the knowledge of colleague's, Siemens Finland's support assistance services and practical tests at work shop were used.</p> <p>A ready made program base was utilized in programming in order to produce an integral appearance and approach, as in other Outotec motion control applications. This idea proved to be practical compared to if every programmer carried out programming in their own way.</p> <p>A later use for this thesis, for example in future commissioning, serves a chapter where signal pass is described from operator to actuator.</p>	
Keywords	hardware design, software design, testing, servo control, position control

Sisällys

1	Johdanto	1
1.1	Työn taustaa	2
1.2	Aloituspalaveri	4
2	Laitteistosuunnittelu	5
2.1	Piirikaavio	5
2.1.1	Jäähdytyskoneen mitoitus	6
2.1.2	Tehon syöttö	6
2.1.3	Sähkökäyttöjen johdotus ja kaapelointi	6
2.1.4	Kenttäväylät	7
2.1.5	IO	8
2.2	Ohjauskaapin osasijoittelu	8
3	Ohjelmointi	9
3.1	Ohjelmatyökierron spesifikaatio	9
3.2	Siemens Simatic Step 7	9
3.3	HW konfiguraatio	13
3.4	WinCC flexible työpöytä	16
3.5	Käyttöliittymä, HMI	17
3.5.1	HMI, avaussivu	17
3.5.2	HMI, pääsivu	18
3.5.3	HMI, käsiajosivu	21
3.5.4	HMI, työparametrisivu	22
3.5.5	HMI, akseliparametrit	22
3.5.6	HMI, työaskeltaulu	23
3.5.7	HMI, raja-arvoskaalaussivu	24
3.6	Muuttujien liittäminen käyttöliittymän objekteihin	25
3.7	Step 7 ohjelmointi	27
3.7.1	Step 7-ohjelmalohkot	28
3.7.2	Step 7 datatyypit	30
3.7.3	Projektin ohjelmarakenne ja lohkot	32
3.7.4	Paikoitustoimintolohko	33
3.7.5	Servo-ohjaimien paikkalaskurien skaalaus liikealueelle	34

3.7.6	Servoakseleiden loogiset toimintolohkot	34
3.7.7	Purkajaluukun servojen yhteiskäyttö	35
3.7.8	Työsekvenssi	36
3.7.9	Työkierron toimintolohko	41
3.7.10	Lisäykset rakennetoimintolohkoon (OB)	41
3.7.11	Signaalin kulku operaattorilta toimilaitteelle	41
4	Käyttöönotto konepajalla	48
4.1	Siemens STARTER käyttöönotto työkaluohjelma	50
4.2	STARTER projektin luominen	51
4.2.1	Sähkökäyttöyksikön konfigurointi	55
4.2.2	Sähkökäyttöjen käyttöönotto	65
4.3	Siemens Simatic ET200S logiikan käyttöönotto	68
4.4	Sovellusohjelman käyttöönotto	70
5	Yhteenveto	73
	Lähteet	75
	Liitteet	
	Liite 1. Projektin aloituspalaverimuistio	
	Liite 2. Siemens Sinamics sähkökäyttöjen data	
	Liite 3. Eplan	
	Liite 4. Simatic Step 7 ohjelmalistaus, purkajaluukun servo 1:n DB	
	Liite 5. Simatic Step 7 ohjelmalistaus, purkajaluukun servo 1:n LogicFB	
	Liite 6. Simatic Step 7 ohjelmalistaus, purkajaluukun servo 2:n DB	
	Liite 7. Simatic Step 7 ohjelmalistaus, purkajaluukun servo 2:n LogicFB	
	Liite 8. Simatic Step 7 ohjelmalistaus, sektoriluukku 1 servon DB	
	Liite 9 Simatic Step 7 ohjelmalistaus, sektoriluukku 1 servon LogicFB	
	Liite 10. Simatic Step 7 ohjelmalistaus, sektoriluukku 2 servon DB	
	Liite 11. Simatic Step 7 ohjelmalistaus, sektoriluukku 2 servon LogicFB	
	Liite 12. Simatic Step 7 ohjelmalistaus, paikkalaskurin skaalaus FB	
	Liite 13. Simatic Step 7 ohjelmalistaus, työsekvenssi FB Seq01	
	Liite 14. Simatic Step 7 ohjelmalistaus, purkajan servojen paikkaero	
	Liite 15. Simatic Step 7 ohjelmalistaus, työkierron DB	
	Liite 16. Simatic Step 7 ohjelmalistaus, työkierron LogicFB	
	Liite 17. Sinamic Step 7 ohjelmalistaus, UDT Telegram 111	
	Liite 18. Sinamic Step 7 ohjelmalistaus, paikoitus FB Sinamics Position	

Liite 19. Sinamic Step 7 ohjelmalistaus, rakennelohko OB1

Lyhenteet

CPU	C entral P rocessing U nit. Keskusyksikkö. Prosessori.
DB	Siemens STEP 7 D ata B lock, käyttäjädata
DRIVE-CLiQ	Siemensin Ethernet-pohjainen tiedonsiirtoprotokolla moottorin takaisinkytkennälle SINAMICS-ohjausjärjestelmässä.
EMC	E lectro M agnetic C ompatib l ity. Sähkölaitteiden sähkömagneettinen yhteensopivuus.
Enkooderi	Paikkatakaisinkytkennän anturi. Kertoo moottorin paikan.
Eplan	Sähkösuunnitteluohjelma
EPOS	Siemens Simatic Basic positioner. Siemens sähkökäyttöjen moottorien paikoitustoiminnallisuus.
EU	E uroopan U nioni
FB	Siemens STEP 7 F unction B lock, toimintolohko
FBD	F unction B lock D iagram. Toimintolohkodiagrammi ohjelmointikieli
FC	Siemens STEP 7 F unction C ode, tehtävä
GUI	G rafical U ser I nterface. Graafinen käyttöliittymä
HMI	H uman M achine I nterface. Käyttöliittymä
HW	H ard W are. Kovo. Laitteisto
IEC	The I nternational E lectrotechnical C ommission eli kansainvälinen sähköalan standardointiorganisaatio
<i>kursiivi</i>	Tässä raportissa kursiviin käyttö tarkoittaa tekstiä työkaluohjelmassa.
LAD	L ADder logic. Tikapuulogiikka ohjelmointikieli
LED	L ight E mitting D iode. Ledilamppu.
LPC	L aptop P ersonal C omputer eli kannettava tietokone.
OB	Siemens STEP 7 O rganization B lock, rakennetoimintolohko
PG/PC	Siemens ohjelmointilaite. Käytännössä kannettava tietokone.
SCL	S tructured C ontrol L anguage. PASCALiin perustuva korkean tason tekstimuotoinen ohjelmointikieli joka pitää sisällään myös ohjelmoitaville logiikoille tyypilliset elementit kuten tulot, lähdöt, ajastimet, bittimuistit, lohkokutsut ja niin edelleen.
STL	S Tatement L ist. Käskylista ohjelmointikieli
SW	S oft W are. Ohjelmisto, ohjelma.
UDT	U ser D ata T ype. Käyttäjän luoma datatyyppi.

Sanastoa

Distributor	Kuivaimen purkaja. Sisältää kaikki purkajan servomoottorit
Distributing device	Purkajaluukku. Kahden servomoottorin liikuttama mekanismi materiaalipurkauksen tuottamiseksi
Distributing device 1	Purkajaluukun servo 1
Distributing device 2	Purkajaluukun servo 2
Radial Gate 1	Purkajan sektoriluukku 1
Radial Gate 2	Purkajan sektoriluukku 2
Layout	Osasijoittelu
Online	yhteydessä eli kommunikointiliityntä laitteiden välillä aktiivinen
Offline	yhteydetön eli kommunikointiliityntä laitteiden välillä estetty
Toolbar	Työkalupalkki
Zoom	Suurentaa näkymää
Wizard	Jonkin toiminnon avustaja. Ohjelma joka opastaa asetuksissa.

1 Johdanto

Opinnäytetyönä toteutin Outotecille malmigranulikuivaimen jakopään Siemens servo-ohjauksen laitteistosuunnittelun ja ohjelmoinnin [1]. Lisäksi suoritin käyttöönoton konepajalla. Laitteistosuunnittelun aloitin Outotecin Espoon automaation suunnittelutoimistolla Outotecin toimintajärjestelmän mukaisen aloituspalaverin jälkeen. Suunnittelun toteutin Eplan-ohjelmalla. Tämän jälkeen jatkoin Siemens-ohjelmoinneilla. Aloitin käyttöliittymästä ja jatkoin siitä servoliikkeen toteutukseen. Käyttöönottoon Outotecin Turulan konepajalla pääsin, kun kaappivalmistaja oli toimittanut ohjausjärjestelmäkaapin konepajalle [2, 3].

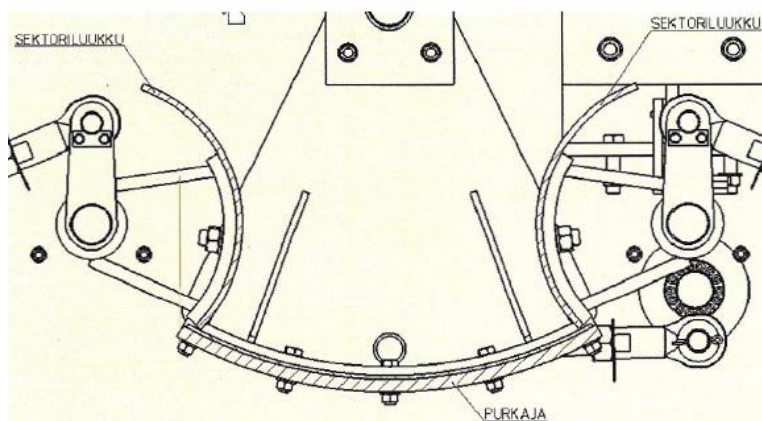
Mielenkiintoiseksi tämän työn teki sen toteutus Siemensillä, josta minulla ei ollut aiempaa kokemusta. Valinnan Siemensistä sovellettavana laitteistoina sekä sen mitoitukset oli Outotec tehnyt jo ennen tämän työn aloittamista (liite 2). Näin ollen syitä komponenttivalinnoille ei tässä työssä pohdita. Myös Eplan laitteistosuunnittelun työkaluna on mielenkiintoinen, koska aiemmat kokemukseni tällaisesta työkalusta liittyvät KymDatan CADSiin ja Autodeskin AutoCADiin.

Tavoitteenani oli saada laitteisto toteuttamaan Outotecin määrittelemää työkiertoa, samalla tavoitteeni oli myös omaksua Siemensin laite- ja ohjelmointiympäristöjä siinä laajuudessa, että voisin tulevissa töissäni ottaa vastaan samanlaisia toimeksiantoja sekä luoda perusta jolta edelleen syventää Siemens osaamista. Keinoina käytin kollegoiden tietämyksen hyödyntämistä, Siemensin käyttöohjeita, Siemensin Suomen tuen palveluita sekä tietenkin konepajalla tehtäviä käytännön testejä.

1.1 Työn taustaa

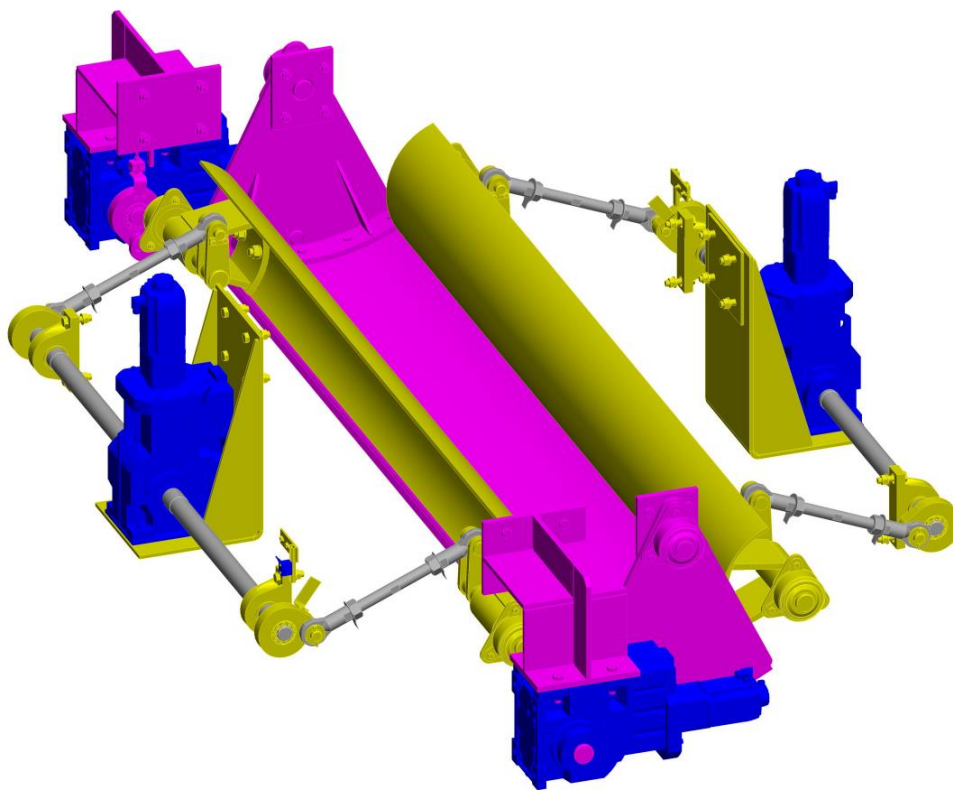
Purkaja on osa metalligranulin valmistuslinjaa. Outotec toteuttaa linjan ensimmäistä kertaa, joten kyse on prototyypistä. Purkaja sijoittuu fyysisesti granulikuivaustornin alapäähän, jossa sen on tarkoitus jakaa granulaattia tasaisesti jatkoprosessointia varten. Loppuasikas on Saudi-Arabiassa.

Kuivaimen purkaja on servo-ohjattu mekanismi, jolla pyritään luukkujen liikkeitä säätämällä saamaan aikaan tasainen materiaalivirtaus ulos purkajaluukun kummaltakin puolelta. Purkaja koostuu purkajaluukusta, jonka kummassakin päässä on servomoottori purkausvaaputusliikkeen toteuttamiseksi sekä purkajaluukun kummallakin reunalla olevista sektoriluukuista, joilla säädetään purettavan materiaalin määrää (Kuvio 1). Sektoriluukuilla on kummallakin omat servomoottorinsa. Purkajaluukun vaaputtavaa liikettä voidaan säätää välillä $\pm 7.45^\circ$:ta (n. ± 70 mm:ä nollapaikasta) ja sektoriluukkujen liike voidaan säätää välillä $0 - 60^\circ$:ta (n. 190 mm:ä nollapaikasta). Sektoriluukkujen ollessa ala-asennossa on kulma 0° :ta. Purkajaluukku on keskiasennossa, kun kulma on 0° :ta. Tällöin purkaja on suljettu, eikä materiaalivirtausta tapahdu.



Kuvio 1. Purkajan luukut

Kuviossa 2 violetilla värillä havaittavissa purkajaluukku, keltaisella värillä sektoriluukut ja sinisellä värillä vaihde/servomoottoriyhdistelmät.



Kuvio 2. Purkaja

1.2 Aloituspalaveri

Työn aloituspalaveri pidettiin Espoossa, Outotecin Safiiri-nimisellä toimistolla. Palaverin kulussa noudatettiin Outotecin toimintajärjestelmän mukaista asialistaa johon myös asiat kirjattiin (liite 1). Palaverin puheenjohtaja Panu Kinnunen esitteli prosessin, projektin dokumenttikansion sijainnin Outotecin toimintajärjestelmän kovalevyrakenteessa ja kuvasi sanallisesti itse kohdelaitteiston toimintaa. Lisäksi kävi ilmi, että laitteistoja toteutettaisiin kaksi kappaletta. Ohjaukset ovat keskenään identtisiä, joten ensimmäisen toteutuksen jälkeen toinen voidaan tuottaa kopioimalla ensimmäinen. Erityisesti huomionarvoista on, että ohjauskaappi sijoitetaan ulkoilmaan. Määäävinä tekijöinä huomioon otettavia ovat: ulkoilman maksimilämpötila +46 °C ja minimilämpötila +2 °C, mahdolliset hiekkamyrskyt sekä sadeveden sisään pääsyn esto (Kuvio 3).

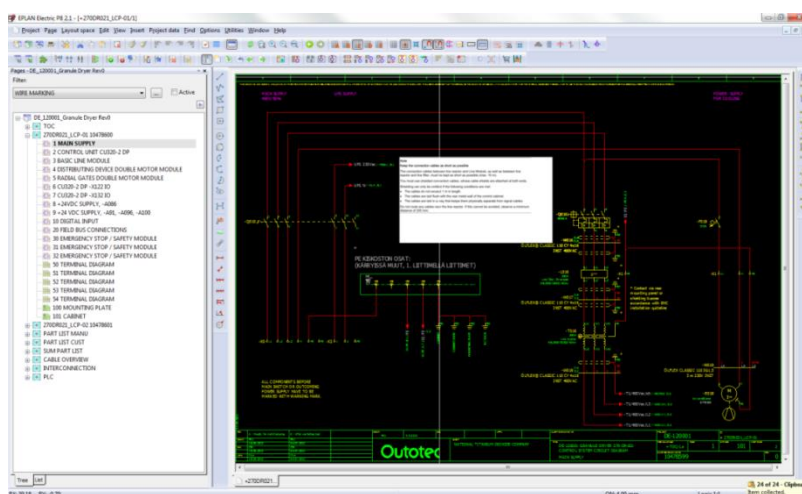
Temperatures	
Max. ambient air temperature	+46 °C
Min. ambient air temperature	+2 °C
Avg. ambient air temperature	+30 °C
Wet bulb design temperature (summer)	+31 °C
Wind	
For the purposes of plant design the following will apply:	
Main wind direction	South
Basic wind speed	UBC-97, Exposure D, $v=80 \text{ m/hr}=128,8 \text{ km/h}=35,8 \text{ m/s}$
Heights from the ground level	Pressure basis, kN/m ²
Up to 10 m	1,23
10 to 20 m	1,38
20 – 50 m	1,57
Rainfall	
Yearly rainfall average	75 mm
In 24 h, maximum	90 mm
In 1 h, maximum	25 mm
Others	
Max. relative humidity	10 – 100 %
Atmospheric pressure	Sea level, 101.3 kPa
Specification of sand storms	10 days / year

Kuvio 3. Loppusijoituspaikan ilmasto-olosuhteet [4, s. 3, 4]

2 Laitteistosuunnittelu

2.1 Piirikaavio

Tässä työssä suunnittelua ei aloitettu niin sanotusti tyhjältä pöydältä, vaan se perustettiin jo aiemmissa projekteissa tehdyille piirikaavioille, joissa suurelta osin samaa laitteistoa oli sovellettu aiemmin. Työn opastaja Risto Huttunen kertoi minkä aiemman projektin piirikaaviot otettaisiin työn alle ja muokattaisiin vastaamaan tämän projektin laitteistoa. Tällainen suunnittelumenettely on hyvin tavanomaista suunnittelutoimistoissa. Näin ollen tehtäväksi muodostui Eplan kuvien päivittäminen poistamalla ylimääräiset komponenttisyömit kuvista, puuttuvien symbolien lisääminen, kytkentöjen tarkastaminen ja päivittäminen (liite 2, s. 20), osatietojen päivittäminen (liite 2, s. 1) sekä kaapin ja sen asennuslevyn osasijoittelun suunnittelu. Käytettävät komponentit saatiin tiedoksi Siemensin laitemyynnin laatimalla dokumentilla (liite 2). Nyrkkisääntönä piirikaavioissa esitettävien komponenttien tunnuksille Outotec pitää standardin mukaisia viitetunnuskirjaimia lisättynä sivunumerolla (varataan kaksi numeropaikkaa jolloin sivunumeron ollessa pienempi kuin 10, laitetaan numeron eteen nolla), sivun sarakenumerolla sekä tarvittaessa juoksevasta numeroinnista. Tällöin esimerkiksi piirikaavion (liite 3) sivulla 1, sarakkeessa 6 sijaitseva sähkökäyttöjen tehonsyötön häiriönpoistosuodin sai tunnukseksi –L 01 6. Tämä on perusteltua esimerkiksi käyttöönottoa silmälläpitäen, jolloin tarvittaessa nähdään suoraan komponentin viitetunnuksesta, millä sivulla ja sarakkeella sen kytkentä on esitetty piirikaavioissa (Kuvio 4).



Kuvio 4. Eplan, komponenttien viitetunnusten esittäminen

2.1.1 Jäähdytyskoneen mitoitus

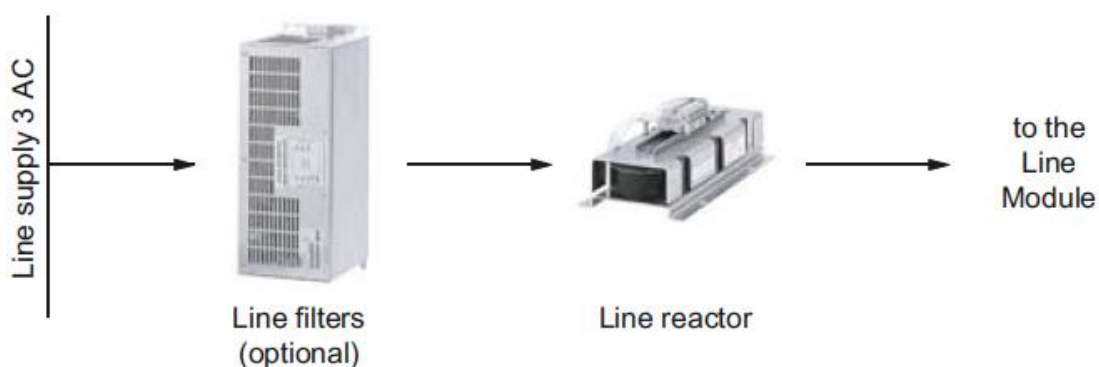
Järjestelmän ilmastointilaitteen mitoitus onnistui näppärästi Rittalin Therm mitoitusohjelman avulla. Ohjelmaan tarvitsi vain syöttää laitteiden yhteenlaskettu häviöteho (liite 2, s. 30), kaapin ulkolämpötila (Kuvio 3), sallittu kaapin sisälämpötila (liite 2, s. 3 – 7) ja kaapin sijoitus, joka tässä tapauksessa oli yksittäiskaappi seinää vasten. Ohjelman ehdottaman laitteen asennusohjeesta löytyivät tiedot syöttösulakkeen koosta ja johdinpaksuuksista.

2.1.2 Tehon syöttö

Ohjauskaappi saa pääkytkimensä kautta sekä 3- että 1-vaiheisen tehonsyötön. Tätä varten pääkytkin valittiin kuusinapaiseksi (liite 3, s. 1). Tehonsyötön ohjausjärjestelmään liittämisen helpottamiseksi varattiin kaapin asennuslevylle, kytkentäsuunnassa pääkytkimen eteen syöttökaapelin johdinpoikkipinnalle sopivat riviliittimet viitetunnuksella –X0.

2.1.3 Sähkökäyttöjen johdotus ja kaapelointi

Siemens sähkökäyttöyksikön tehonsyötön periaate esitetty kuviossa 5.



Kuvio 5. Siemens sähkökäyttöyksikön tehonsyötön periaate [5, s. 46]

Verkkosuotimet on suunniteltu vaimentamaan johtuvien häiriöiden päästöjä asianmukaisen EMC-lainsäädännön sisältämien spesifikaatioiden mukaisesti. EMC-lainsäädännöllä pyritään varmistamaan sähkölaitteiden sähkömagneettinen yhteensopivuus. Koneiden valmistajat ovat velvollisia huolehtimaan, että koneet ovat Euroopan Unionin (EU) EMC-direktiivin mukaisia. Noudattamalla komponenttitoimittajien asennusohjeita tämä asia tulee varmistetuksi. Siemens-asennuksessa nämä ohjeet ovat [5, s. 84]:

1. Kytkenäkaapelit pidettävä mahdollisimman lyhyinä. Kaapelit välillä syöttökuristin ja teholähde ja myös syöttökuristin ja häiriönpoistosuodatin on pidettävä mahdollisimman lyhyinä. Maksimi on 10 m.
2. On käytettävä suojattua kytkenäkaapelia, jonka suojasukka on kytkettävä kummastakin päästä. Suojasukan voi jättää pois vain jos:
 - kaapelien pituus ei ylitä 1 m
 - kaapelit ovat samalla tasolla asennuslevyn kanssa
 - kaapelit pysyvät fyysisesti erossa signaalikaapeleista.
3. Kaapelointeja ei viedä kuristimen läheltä. Jos tätä ei voida välttää, on minimi etäisyys 200 mm.

Nämä ohjeet liitettiin myös tehonsyötön piirikaavioon (liite 3, s. 1). Sähkökäyttöyksikön Drive-CliQ kommunikointikaapelointi on piirretty suoraan Siemensin ohjeiden mukaan (liite 2, s. 20), [6, p. 35, 36]. Sähkökäyttöjen johdotus ja kaapelointi on esitetty liitteessä 3.

2.1.4 Kenttäväylät

Kenttäväyläkaapelointeja ei sanan varsinaisessa merkityksessä järjestelmään tullut, vaan ne tulivat ohjauskaapin sisälle, ohjaavasta logiikasta käyttöpäätteelle ja servo-ohjainyksikköön. Ylempään ohjaukseen liittymistä varten kaappiin varattiin Siemensin Scalance X101 Ethernet mediamuunnin. Ylemmän ohjauksen liittymän testaus ja käyttöönotto eivät kuuluneet tämän työn piiriin. Piirikaavioissa kenttäväylien esittäminen tapahtuu yksinkertaisesti piirtämällä jatkuva viiva liitäntäpisteiden väliin (liite 3).

2.1.5 IO

Tuloja jakolaitteen ohjausjärjestelmässä on äärimmäisen vähän ja lähtöjä ei ollenkaan. Tuloihin liitettiin ainoastaan jokaisen servon niin sanottu 0-paikan induktiivinen anturi sekä järjestelmän hätä/seis-painike, joka kytkettiin turvatuloon. Turvatulojen konfiguraation kuvaus rajattiin tämän raportin ulkopuolelle.

2.2 Ohjauskaapin osasijoittelu

Kaapin ulkoilmaan sijoituksen vuoksi katsottiin parhaaksi tehdä sen sopivan tyyppin löytämisessä yhteistyötä Outotecin kaappitoimittajan Rittalin kanssa [7]. Asennuslevylle tulevien osien ulkomitoista saatiin käsitys tarvittavan asennuslevyn koosta. Yhdessä tämän ja ympäristöolosuhdetietojen kanssa kaappitoimittaja suositteli vartavasten ulkokäyttöön suunniteltua, Rittalin CT Toptec kaksoisrakennekaappia. Kaksoisrakenteen ansiosta voidaan olla varmoja, ettei kosteus kondensoidu sisälle kaappiin, vaan rakenteen välitilaan, josta se johtuu ulos hallitusti, aiheuttamatta ongelmia kaapin sisällä olevalle elektroniikalle.

Ensimmäiseksi sijoitettiin asennuslevylle servo-ohjainyksikkö kohtaan, jolle kaapin ilmastointilaitte syöttää viilennetyn ilman yksikön alapuolelle. Tämän alle sijoitettiin järjestelmän tehonsyötön komponentit ja liitännät. Asennuslevyn yläosaan sijoitettiin ohjauslogiikan ja muut liitännät (liite 3, s. 15).

Ohjauskaapin ulkoisessa olemuksessa suunniteltavaa olivat käyttöpäätteen ja sen suojaoven sijoitus sekä ilmastointilaitteen sijoitus. Kuvaan piirretään tavallisesti mukaan myös kaapin ulkomitat (liite 3, s. 16). Käyttöpäätteen sijoituskorkeutta määrättäessä on syytä huomioida ihmisen fysiikkaan perustuva ergonomia. Käyttöönottokokemusten perusteella tässä onnistuttiin hyvin.

3 Ohjelmointi

3.1 Ohjelmatyökierron spesifikaatio

Alkutilanteessaan ennen työkierron käynnistymistä akselit ovat nollapaikoissaan, mikä tarkoittaa purkajaluukulla keskiasentoa ja sektoriluukuilla ala-asentoa siten, että luukut ovat kosketuksissa keskenään sulkien mahdollisen purkaustien. Yksinkertaisuudessaan purkajan automaattinen työkierto on seuraavanlainen: kun käyttäjä painaa toimintonäppäintä automaattiajon käynnistämiseksi, avautuvat ensin sektoriluukut asentoon, joka on asetettu työparametrisivulla. Kun sektoriluukut ovat saavuttaneet kohdepaikkansa, on purkajaluukun vuoro aloittaa heiluntaliikkeensä. Purkajaluukku ajaa työparametrisivulla asetettuun kohdepaikkaansa ja odottaa materiaalin purkautumista ajan, joka myös on asetettu työparametrisivuilla. Odotusajan jälkeen purkajaluukku ajaa toiseen kohdepaikkaansa työparametrisivulla asetetulla nopeudella ja toistaa odotusajan myös tässä kohdepaikassa. Tätä heilutusliikesekvenssiä toistetaan kunnes käyttäjä pysäyttää sen käyttöpäätteeltä painamalla pysäytystoimintonäppäintä.

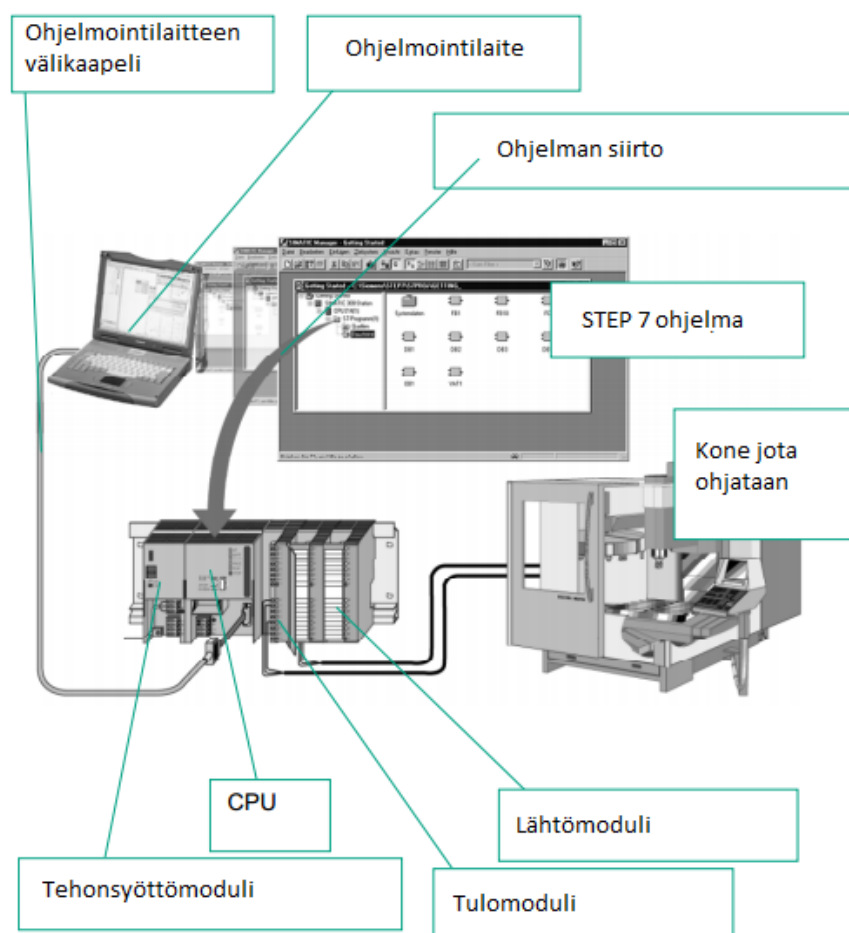
3.2 Siemens Simatic Step 7

STEP 7 on SIMATICin ohjelmointi- ja konfigurointiohjelma. Se on rakennettu automaation ohjelmoimiseksi sarjasta sovelluksia kuten

- parametrien konfigurointi ja laitteistoon osoittaminen
- käyttäjäohjelmien luominen ja testaus
- väylien ja liitäntöjen konfigurointi.

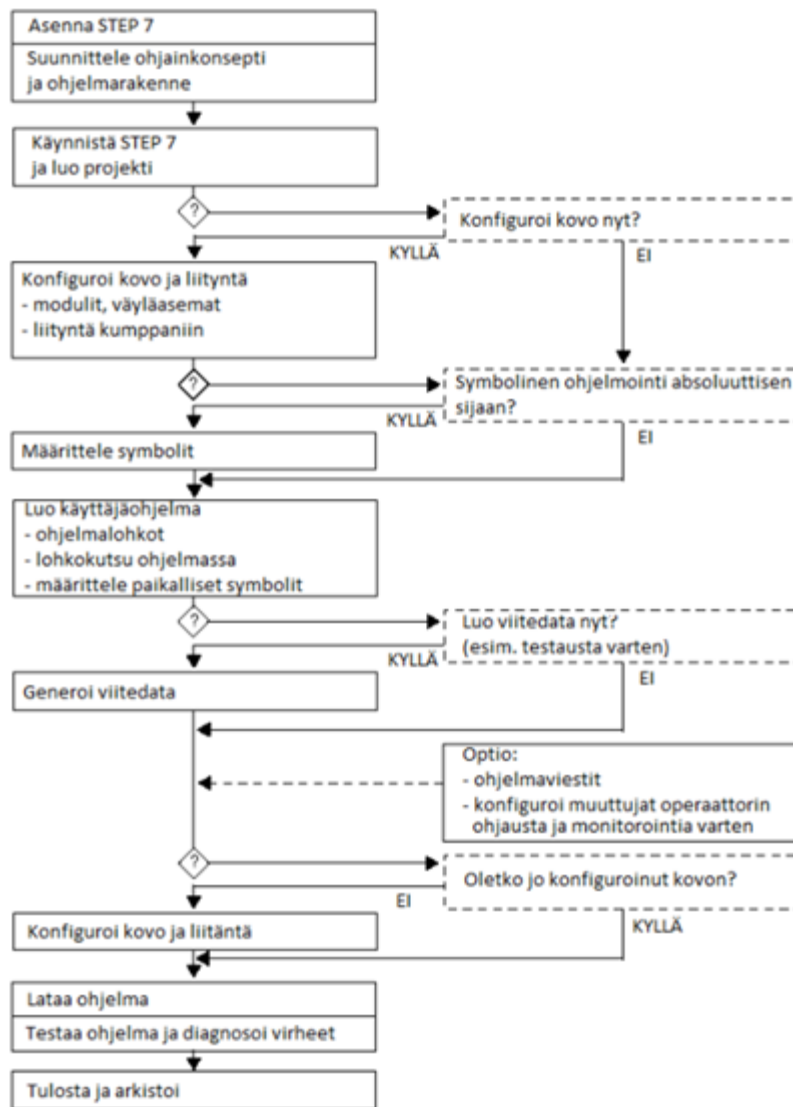
Näiden tehtävien suorittamiseksi tarjottu graafinen käyttöliittymä (GUI) tunnetaan nimellä SIMATIC Manager. Managerilla voi hallita projekteja ja kirjastoja, kutsua STEP 7 työkaluja, liittyä logiikkaan (PLC) ja editoida SIMATIC mikromuistikorttia.

Laitteistoympäristön periaate kaikkienensa on esitetty kuviossa 6.



Kuvio 6. Laitteistoympäristön periaate

Outotec haluaa käyttää ohjelmoinneissaan vakio-ohjelmapohjaa, joka jo sisältää muun muassa käyttöliittymän yleisilmeen, hälytysnäytöt, valmiita toimintolohkoja ja toiminto-ohjelmalohkopohjia ohjelmoinnin aloittamisen helpottamiseksi ja ohjelmoinnin yleisilmeen pitämiseksi halutunlaisena. Tämä työ perustettiin tällaiselle ohjelmapohjalle joten STEP 7 projektin luontia ei kuvata alusta alkaen, vaan raportin kuvaus keskittyy tähän pohjaan tehtyihin ohjelmallisäyksiin. Vakio-ohjelmapohja kopioitiin ja avatiin SIMATIC Manageriin. Kuvio 7 kuvaa automaatio-sovelluksen luomisen STEP 7:llä niin sanotusti tyhjältä pöydältä.



Kuvio 7. STEP 7 automaatio-sovelluksen luomisessa tarvittavat askeleet

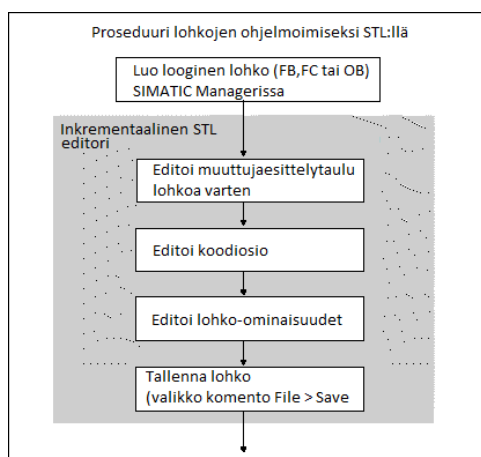
Kuviosta 7 nähdään, että sovelluksen luomiseksi on olemassa kaksi vaihtoehtoista proseduuria:

1. Laitteisto voidaan konfiguroida ensin ja sitten ohjelmoida lohkot.
2. Lohkot voidaan ohjelmoida ensin ilman laitteiston konfigurointia.

Jälkimmäistä vaihtoehtoa suositellaan esimerkiksi huolto- ja ylläpitotyölle, jolloin ohjelmoidut lohkot integroidaan olemassa olevaan projektiin. Tätä tapaa käytettiin myös tämän projektin ohjelmoinneissa.

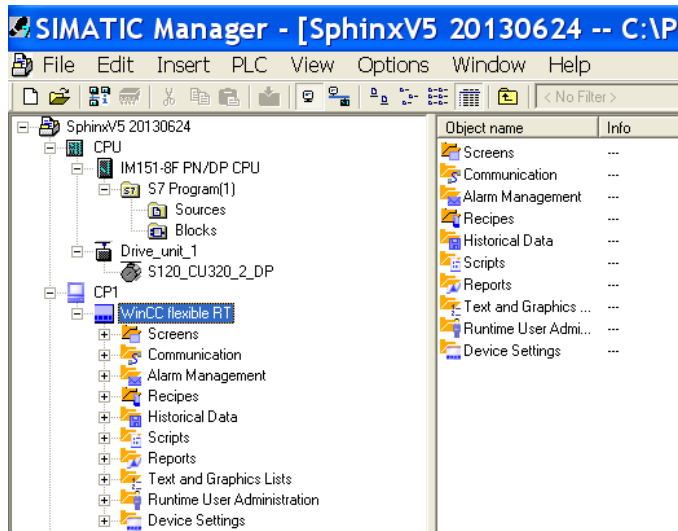
Loogiset lohkot (OB:t, FB:t, FC:t) sisältävät muuttujien esittely- ja ohjelmakoodiosion kuin myös niiden ominaisuudet. Ohjelmoitaessa on editoitava (Kuvio 8):

1. Muuttujien esittely. Tarkennetaan parametrit, parametrien järjestelmäattribuutit ja lohko-kohtaiset paikalliset muuttujat.
2. Ohjelmakoodiosiossa ohjelmoidaan logiikan prosessoima lohkokoodi. Ohjelmoinnissa voidaan käyttää tikapuulogiikkaa (LAD), toimintolohkodiagrammia (FBD) tai käskylistää (STL).
3. Lohko-ominaisuudet sisältävät lisätietoa, kuten aikaleiman tai systeemin lisäämän polun.



Kuvio 8. STEP 7 proseduuri lohkojen luomiseksi

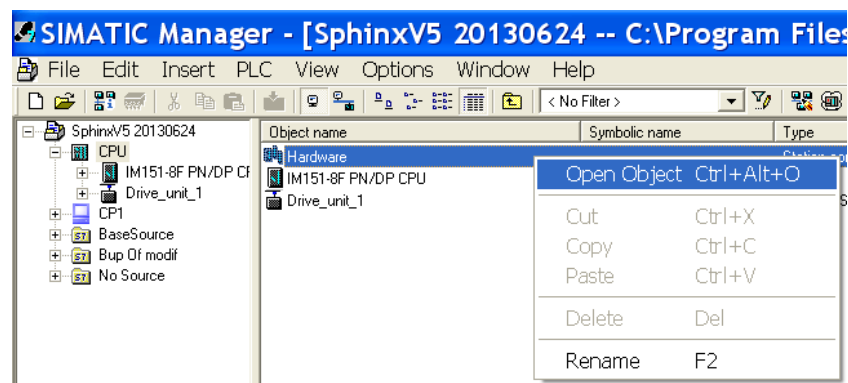
SIMATIC Managerissa näytetään STEP 7 projektien ja kirjastojen objektihierarkia samaan tapaan kuin Windows Explorerissa hakemistorakenne (Kuvio 9).



Kuvio 9. STEP 7 objektihierarkia

3.3 HW konfiguraatio

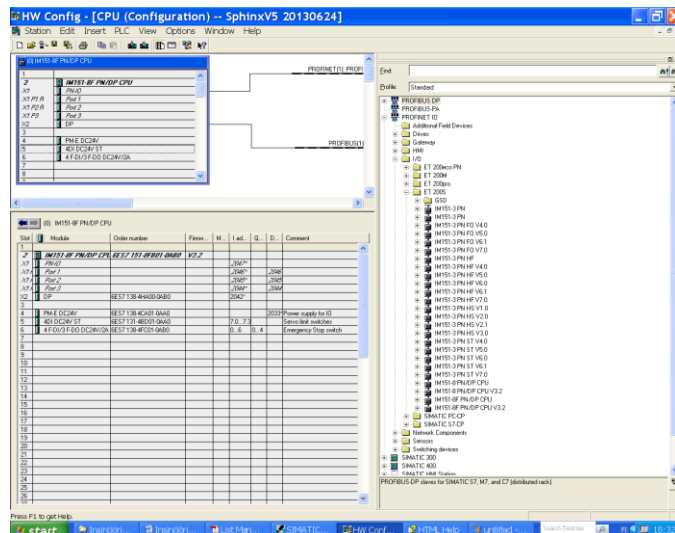
Kovomäärittelyn editori aukeaa Simatic Managerista *Open Object* toiminnolla (Kuvio 10).



Kuvio 10. Kovomäärittelyeditorin avaaminen Simatic Managerista

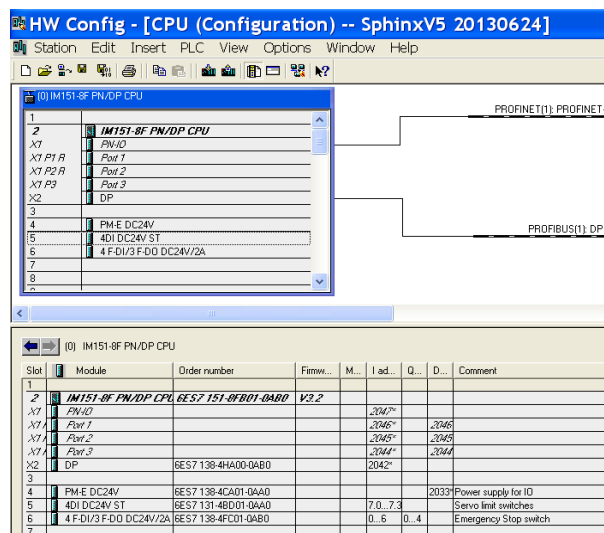
Kovon konfigurointiin liittyy kaksi HW-editorin ikkunaa (Kuvio 11):

1. Aseman ikkuna johon sijoitetaan kehikko aseman rakenteen määrittelyä varten.
2. Kovoluetteloikkuna josta voi valita tarvittavan kovon komponentit.



Kuvio 11. Step 7 HW-editorin ikkunat

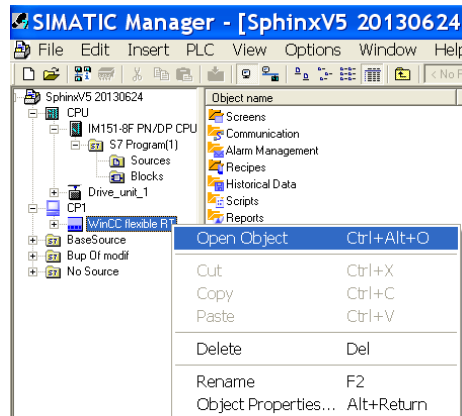
Termi konfiguraatio viittaa kehikoiden, modulien, hajautetun IO:n ja liitäntöjen alimodulien järjestelyyn aseman ikkunassa (Kuvio 12). Kehikot kuvataan konfigurointitaulukkona, joka sallii tietyn modulin määrän asentamisen kuten todellisessa kehikossa. Step 7 määrää automaattisesti osoitteet jokaiselle modulille. Kun PCL käynnistyy, sen CPU vertaa Step 7:llä luotua konfiguraatiota laitteen todelliseen konfiguraatioon (eli moduliin sijoitteluun), jolloin mahdolliset virheet huomataan välittömästi ja raportoidaan.



Kuvio 12. Step 7 PLC:n HW-konfiguraatio

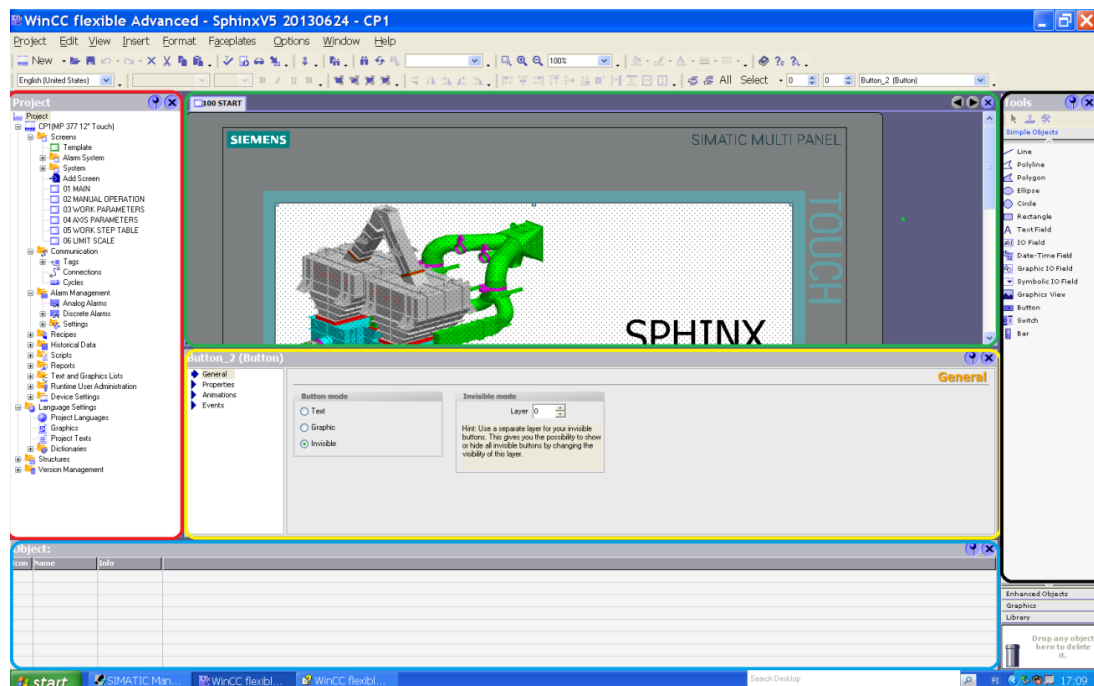
3.4 WinCC flexible työpöytä

Käyttöliittymän ohjelmoinnissa tarvittava WinCC flexible RT käynnistetään Managerista osoittamalla sitä hiirellä, avaamalla ponnahdusikkuna hiiren oikealla painikkeella napauttaen ja valitsemalla siitä toiminto *Open Object* (Kuvio 14).



Kuvio 14. WinCC flexiblen käynnistäminen SIMATIC Managerista

WinCC flexiblen työpöytä aukeaa (Kuvio 15).



Kuvio 15. WinCC flexible työpöytä

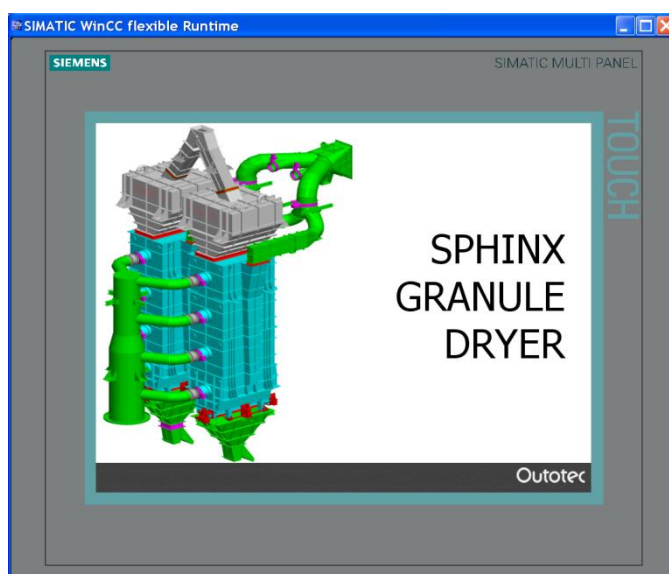
Kuviossa 15 punaisella rajattu alue visualisoi projektin rakenteen, vihreä rajaa työskentelyalueen, keltainen valittuna olevan objektin ominaisuudet, musta visualisoinnissa käytetyt työkalut ja sininen projektirakenteessa valittuna olevan kohteen sisällön. WinCC flexible tarjoaa siis jokaiselle konfigurointitehtävälle erityisen editorinsa.

3.5 Käyttöliittymä, HMI

Jakolaitteen ohjausta varten luotiin WinCC flexiblellä käyttöpäätteeseen sivut Main, Manual operation, Work parameters, Axis parameters. Käyttöön otossa konepajalla lisättiin sivut Work step table ja Limit scale. Seuraavassa on kuvaus kustakin sivusta.

3.5.1 HMI, avaussivu

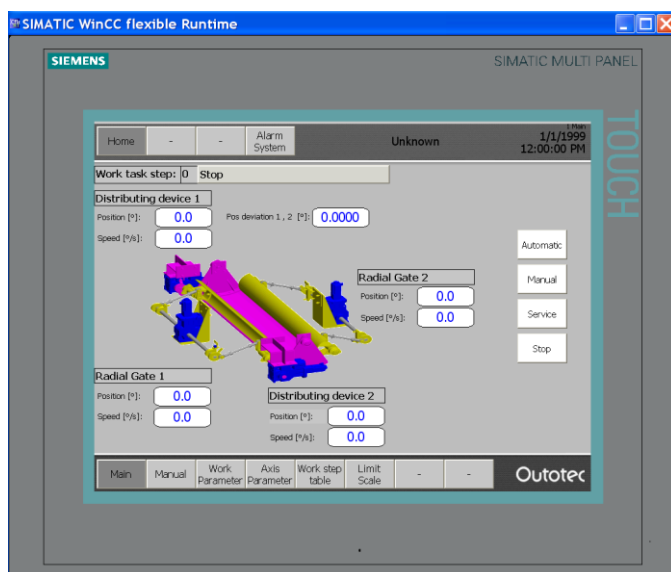
Vakio-ohjelmapohjassa oli jo valmiina käynnistyssivu. Siihen lisättiin mekaniikkasuunnittelijalta saatu kuivaustornin kuva sekä teksti kohdelaitteistosta (Kuvio 16). Tästä näytöstä operaattori pääsee eteenpäin yksinkertaisesti koskettamalla näytön pintaa, jolloin aukeaa niin sanottu pääsivu.



Kuvio 16. Käyttöliittymä, avaussivu

3.5.2 HMI, pääsivu

Pääsivu (Main) on näyttö, jolta käynnistetään automaattiajo, seurataan työkierron aikaisia paikka- ja nopeustietoja, avataan käsiajonäyttö, työparametrinäyttö, akseliparametrinäyttö, työaskeltaulu ja liikealueiden skaalausnäyttö (Kuvio 17). Muille sivuille käyttäjä siirtyy painamalla näytön alareunassa olevia toimintonäppäimiä, jotka ovat näkyvissä kaikilla sivuilla. Sivujen vaihtonäppäimet olivat käyttöpäätteen ohjelmapohjassa valmiina. Niiden näppäinteksti korjattiin vastaamaan kohdesivua sekä lisättiin toiminto, joka avasi halutun kohdesivun. Pääsivun vasempaan reunaan tehtiin toimintonäppäimet automaattiajon käynnistystä varten, käsi- ja huoltoajoja sekä pysäytystä varten. Pääsivulta ei voi syöttää mitään liikeensuoritukseen liittyvää parametrin arvoa, vaan kaikki numerokentät ovat vain luku-tyyppiä (read only). Jokaiselle servo-akselille on omat otsikko-, paikka- ja nopeuskentät. Lisäksi näytön yläosassa on kenttä, joka kertoo mitä osaa työkiertotaulusta ollaan parasta aikaa toteuttamassa.



Kuvio 17. Käyttöliittymä, pääsivu

Tekstikenttien luonti

Esimerkkinä päänäytön tekstikentät:

- Work task step
- Distributing device 1 ja 2
- Radial Gate 1 ja 2
- Position [°]
- Speed [°/s]
- Pos deviation 1, 2 [°].

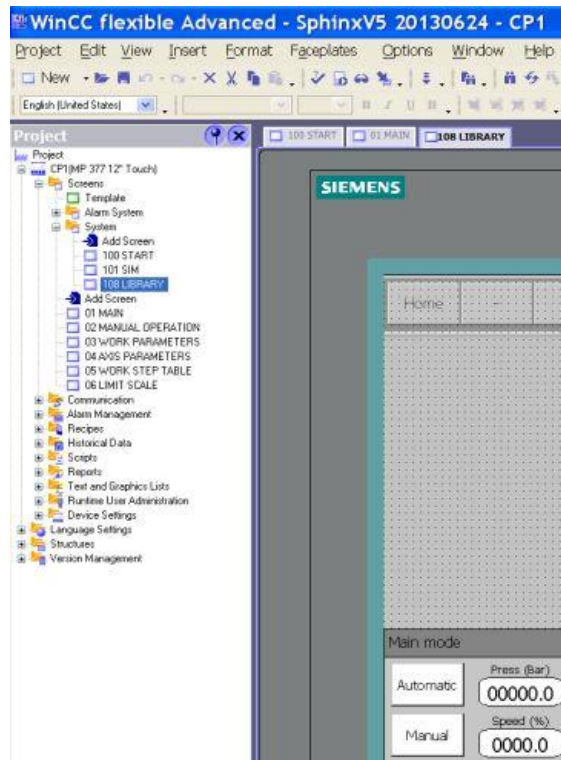
Työkalunäytön avaamista varten napautetaan työkalupalkista auki *View* alasvetovalikko, osoitetaan sieltä *Tools* -riviä ja napautetaan sitä, jolloin työkalunäyttö aukeaa työpöydälle (Kuvio 15). Tekstikentän voi luoda näytölle yksinkertaisesti napauttamalla *Tools* :ssa olevaan *TextField* toimintoon, siirtämällä kursorin näyttöalueella haluttuun paikkaan ja napauttamalla uudelleen hiiren painikkeella kentän kiinni näytölle. Tämän jälkeen kirjoitetaan teksti ohjelmointilaitteen näppäimistöltä. Mikäli tekstin ominaisuuksia halutaan muuttaa, tapahtuu se valitsemalla haluttu tekstiobjekti, avaamalla ponnahdusvalikko hiiren oikealla painikkeella ja valitsemalla avautuvasta valikosta objektin ominaisuudet (*Properties Ctrl+Shift+X*). Näytölle avautuu näyttö ominaisuuksista, jossa voi manipuloida paikkaa, geometriaa, tyyliä, väriä ja kirjaisintyyppiä.

Numerokentät

Esimerkkinä ovat päänäytön paikka- ja nopeuskentät. Käyttöliittymän ohjelmapohjaan on valmiiksi luotu numerokenttäobjekteja, jotka sisältävät ominaisuuksia esimerkiksi liittyen kyseessä olevan kentän muuttujan arvolla indikoitavaan raja-arvon ylitykseen. Nämä valmiit objektit löytyivät WinCC projektirakenteesta systeemin alta, sivulta 108 Library (Kuvio 18). Kyseessä oleva objekti kopioitiin, asetettiin se päänäytölle ja liitettiin siihen muuttuja.

Yksinkertaisemman numerokentän luominen näytölle tapahtuu osoittamalla ja napauttamalla työpöydän *Tools*-valikosta *IO Field*-toimintoa, siirtämällä kursori

näyttöalueella haluttuun paikkaan ja napauttamalla uudelleen hiiren painikkeella kenttä kiinni näytölle. Mikäli kentän ominaisuuksia halutaan muuttaa, tapahtuu se valitsemalla haluttu *IO Field*-objekti, avaamalla ponnahtusvalikko hiiren oikealla painikkeella ja valitsemalla avautuvasta valikosta objektin ominaisuudet (*Properties Ctrl+Shift+X*). Näytölle avautuu näyttö ominaisuuksista, jossa voi manipuloida paikkaa, geometriaa, tyyliä, väriä ja kirjaisintyyppiä.



Kuvio 18. HMI ohjelmapohjan objekti kirjasto

Objektiin liittyvän muuttujan lisääminen sen ominaisuuksiin on kappaleessa 3.6.

Käyttöpainikkeet

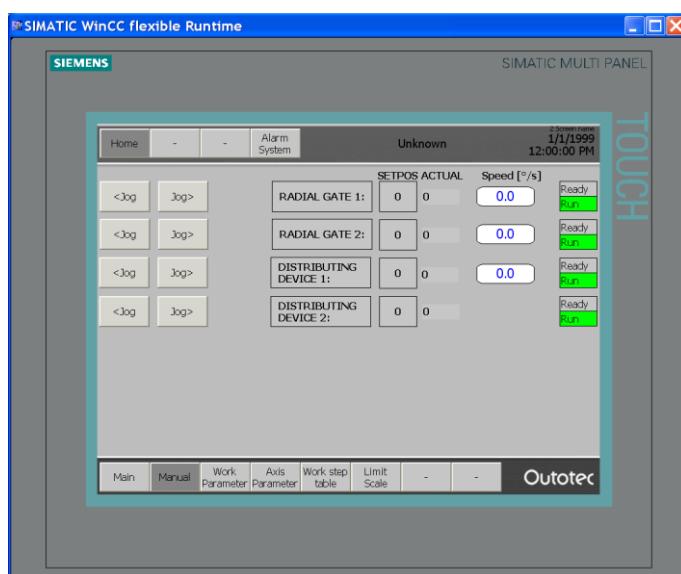
Samaan tapaan kuin numerokenttäobjektin kanssa, myös käyttöpainikkeita on luotu valmiiksi samaan kirjastoon. Käyttöpainikeobjekti kopioitiin kirjastosta, siirrettiin päänäytölle ja tarvittaessa muutettiin näppäinteksti.

Toimintonäppäimen voi luoda näytölle myös osoittamalla ja napauttamalla työpöydän *Tools*-valikosta *Button*-toimintoa, siirtämällä kursori näyttöalueella haluttuun paikkaan ja napauttamalla uudelleen hiiren painikkeella toimintonäppäin kiinni näytölle.

Toimintanäppäimeen liittyvän muuttujan lisääminen ominaisuuksiin on kappaleessa 3.6.

3.5.3 HMI, käsiajosivu

Käsiajosivu aukeaa toimintonäppäimellä Manual. Käsiajosivu on tarkoitettu yksittäisen servoakselin käsiajotoimintoja varten. Tässä näytössä akselia voidaan ajaa joko jogaten eli jatkuvasti suuntaansa, kunnes ajokäsky poistetaan tai asettamalla sille kohdepaikka. Jokaiselle akselille luotiin oma nimikenttä, joggauspainikkeet, kohdepaikka-asetuskenttä, paikan ja nopeuden oloarvokenttä sekä liikkeen tilatietokentät (Kuvio 19). Kentät luotiin kuten on kuvattu kappaleissa Tekstikenttien luonti, Numerokentät.

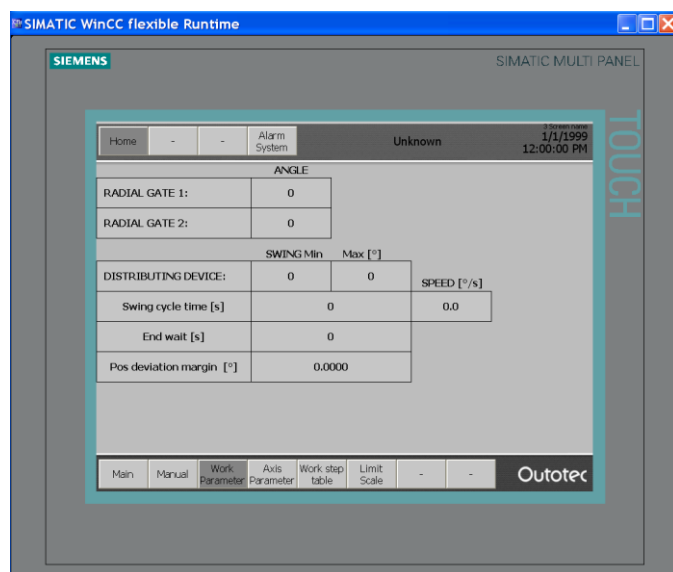


Kuvio 19. Käyttöliittymä, käsiajosivu

Käsiajosivulta, kuten muiltakin sivuilta, käyttäjä voi siirtyä mille tahansa käyttöliittymän sivuista painamalla näytön alareunassa näkyvää toimintonäppäintä.

3.5.4 HMI, työparametrisivu

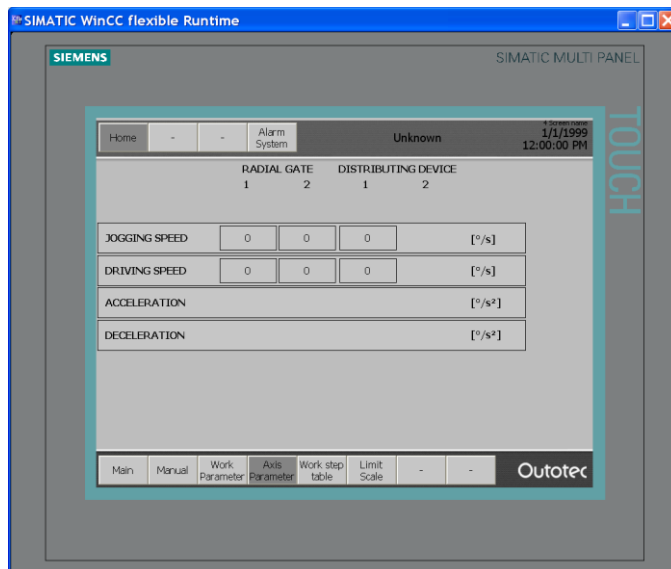
Työparametrisivulta (Kuvio 20) käyttäjä syöttää työkierron toteutuksessa tarvittavat parametrit. Näitä ovat sektoriluukkujen 1 ja 2 avauskulma, purkajan minimi ja maksimi heiluntakulmat, heilahduksen liikeaika, odotusaika kohdepaikassa ja sallittu paikkaeromarginaali. Lisäksi näytöltä voi lukea, minkälaisista nopeutta asetettu heilahduksen liikeaika tarkoittaa säädettävässä mittayksikössä.



Kuvio 20. Käyttöliittymä, työparametrit

3.5.5 HMI, akseliparametrit

Akseliparametrisivu on tarkoitettu käsiajosivun liikeparametrien asettamiseksi (Kuvio 21).



Kuvio 21. Käyttöliittymä, akseliparametrit

Käyttäjä syöttää haluamansa joggaus- ja kohdepaikka-ajon nopeudet kullekin akselille koskettamalla numerokenttää, jolloin näytölle avautuu numeronäppäimistö lukuarvon syöttämiseksi. Jakajan servo 2 noudattaa servo 1:n asetuksia, joten sille ei sallita syöttöä.

3.5.6 HMI, työaskeltaulu

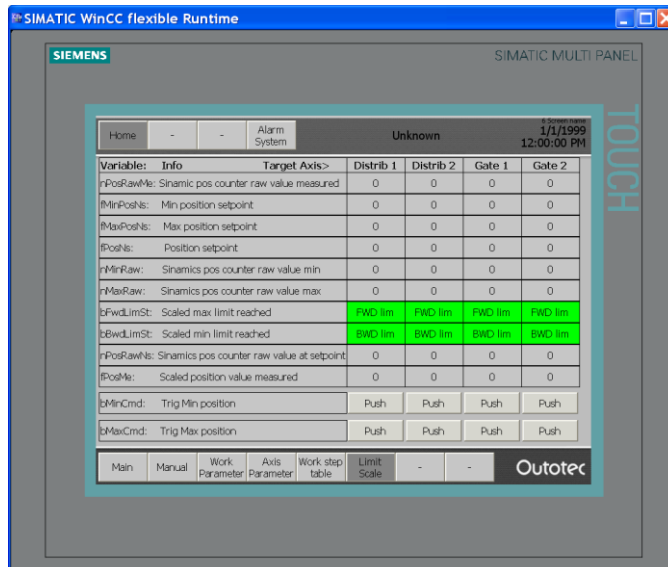
Työkierto katsottiin parhaaksi järjestää taulukkototeutuksena, jossa kulloisenkin vaiheen asetukset on selkeästi hahmotettavissa akselikohtaisesti (Kuvio 22). Tämän taulukon alkiot saavat arvonsa tärkeimmiltä osin työparametrinäytöstä (Kuvio 20). Taulukko onkin käyttöliittymässä pelkästään palvelemissa tulevaa järjestelmän käyttöönottajaa ja on luonteeltaan vain luku-tyyppiä (Read Only).

Step number	Target parameter	D12Vel	D12Pos	G1Vel	G1Pos	G2Vel	G2Pos
Step 00: Stop		0	0.0	0	0.0	0	0.0
Step 01: Start		0	0.0	0	0.0	0	0.0
Step 02: Open Gates		0	0.0	0	0.0	0	0.0
Step 03: Gates Ready		0	0.0	0	0.0	0	0.0
Step 04: Distributor Forward		0	0.0	0	0.0	0	0.0
Step 05: Distributor Forward turn point		0	0.0	0	0.0	0	0.0
Step 06: Distributor Backward		0	0.0	0	0.0	0	0.0
Step 07: Distributor Backward turn point		0	0.0	0	0.0	0	0.0
Step 08: Cycle Done		0	0.0	0	0.0	0	0.0
Step 09: Distributor to Gate 1 closing position		0	0.0	0	0.0	0	0.0
Step 10: Close Gate 1		0	0.0	0	0.0	0	0.0
Step 11: Distributor to Gate 2 closing position		0	0.0	0	0.000	0	0.0
Step 12: Close Gate 2		0	0.0	0	0.0	0	0.0
Step 13: Distributor to middle position		0	0.0	0	0.0	0	0.0
Step 14: Wait		0	0.0	0	0.0	0	0.0
Step 15: Done		0	0.0	0	0.0	0	0.0

Kuvio 22. Käyttöliittymä, työaskeltaulu

3.5.7 HMI, raja-arvoskaalaussivu

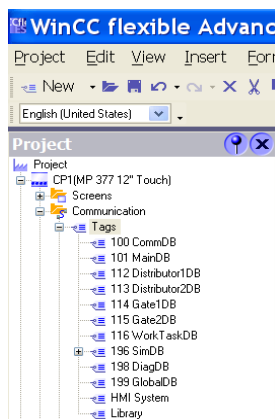
Raja-arvojen skaalaussivu on käyttöliittymässä servoakseleiden liikealueen skaalaamiseksi haluttuun koordinaatistoon. Tämän sivun avulla SINAMICSin servo-ohjaimien paikkalaskurien lukuarvot voidaan muokata paremmin käyttäjän ymmärtämään mittayksikköön. Tällaisiksi ajateltiin käyttöönotossa asteet, koska kyse on ympyräliikkeistä. Käyttäjä syöttää kunkin akselin *fMinPosNs* ja *fMaxPosNs* kenttiin numeroarvot minimi- ja maksimipaikoiksi eli liikealueeksi sekä lisäksi kertoo, mikä on akselin sen hetkinen asema syöttämällä sen kenttään *Position setpoint*. Konepajakäyttöönotossa ajettiin akselit ensin niin, että ne olivat irti mekanismista, STARTER käyttöönotto-ohjelmalla fyysiseen nollapaikkaansa, joka sitten syötettiin arvona nolla myös *Position setpointiksi*. Tämän jälkeen moottorit kiinnitettiin 0-paikkoihinsa asennettuun mekanismiin. Näiden jälkeen numerosyöttö pitää vielä trigata käyttöön painamalla toimintonäppäintä *Push*. Käyttöönottajän työn helpottamiseksi lisättiin tekstikenttiin mukaan myös muuttujan nimi (Kuvio 23).



Kuvio 23. Käyttöliittymä, liikealueiden skaalaukset

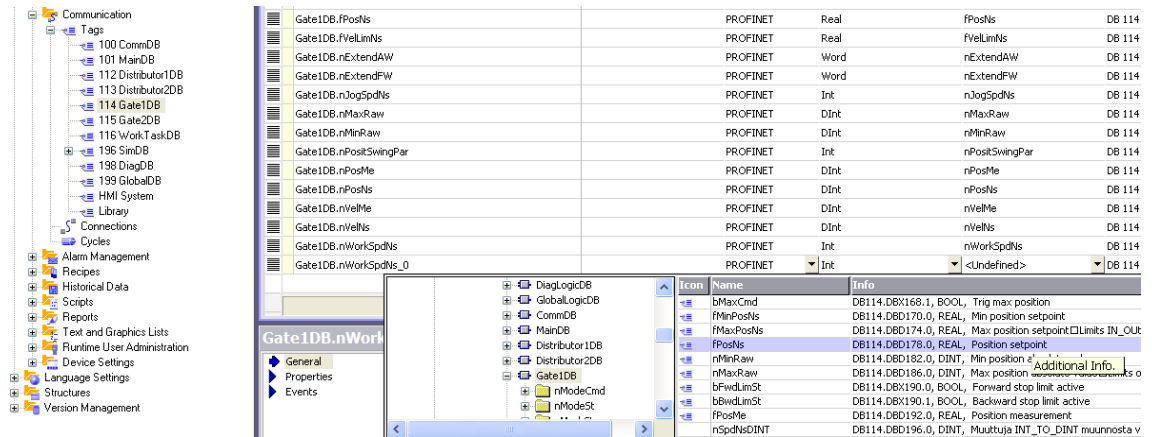
3.6 Muuttujien liittäminen käyttöliittymän objekteihin

Ulkoiset muuttujat mahdollistavat kommunikoinnin (tietojen vaihdon) käyttöliittymän (HMI) ja logiikan (PLC) välillä. Ne ovat PLC:ssä määritellyn muistipaikan kuvia joihin on luku- ja kirjoitusoikeus sekä HMI:stä, että PLC:stä. Editoinnin yksinkertaistamiseksi operandien absoluuttisilla osoitteilla on symboliset nimet Step 7:ssa ja ne on listattu symbolitaulussa (Kuvio 30). WinCC flexiblen tagit kytketään suoraan Step 7:n symboleihin. Tätä varten lisättiin jokaisen akselin ja työkierron DB:t WinCC flexiblen Tags listaan (Kuvio 24).



Kuvio 24. WinCC flexible, Tags

Kun käyttöpääätteessä halutaan liittää johonkin objektiin jonkin PLC:n ohjelman muuttuja, avataan Tag määrittely, valitaan kohteen DB:stä haluttu muuttuja ja liitetään se listaan hiirellä osoittaen ja tuplanapauttaen (Kuvio 25).



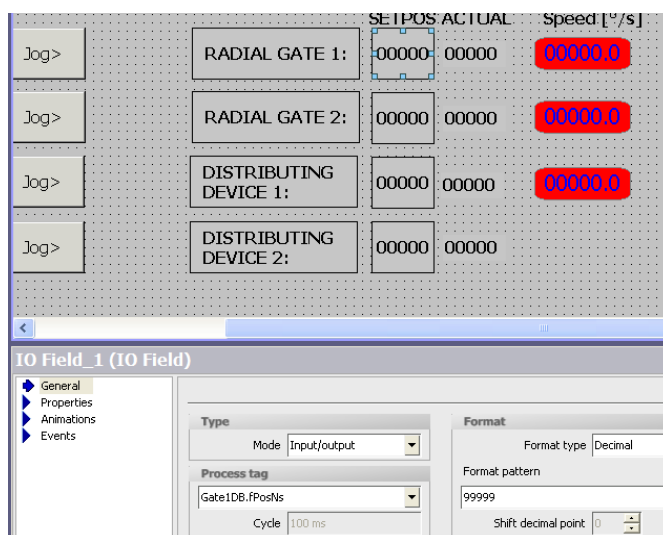
Kuvio 25. WinCC flexible, PLC muuttujan osoitus

Valittu PLC muuttuja liitetään listaan, josta nähdään sen Tag-nimi, liitäntätapa, tietotyyppi, symbolinimi DB:ssä ja osoite (Kuvio 26).

Name	Connection	Data type	Symbol	Address
Gate1DB.fPosNs	PROFINET	Real	fPosNs	DB 114 DBD 178

Kuvio 26. Step 7 DB muuttujan liittäminen HMI:n Tag-listaan

Tämän jälkeen Tag on vielä liitettävä HMI:n objektiin. Tätä varten osoitetaan haluttua objektia ja haetaan haluttu muuttuja sen *Process tag* kenttään. Samalla kerrotaan muuttujan tyyppi ja formaatti (Kuvio 27). Samoin menetellään kaikkien HMI:ltä manipuloitavien PLC muuttujien kohdalla.



Kuvio 27. WinCC flexible, muuttujan liittäminen objektille.

3.7 Step 7 ohjelmointi

Step 7 ohjelmointiohjelma sallii käyttäjäohjelman strukturoinnin yksittäisiin itsenäisiin ohjelmalohkoihin. S7 käyttäjäohjelmassa käytettävät lohkot ovat taulukossa 1 esitellyt.

Taulukko 1. Step 7 lohkot

Lohko	Kuvaus
OB = Organization Blocks l. rakennetoimintolohko	OB määrää käyttäjäohjelman suoritusjärjestyksen.
SFB = System Function Block, SFC = System functions. Järjestelmän toimintalohko ja järjestelmätoiminnot	SFB:t ja SFC:t on integroitu S7:n CPU:lle ja niiden avulla voi päästä käsiksi joihinkin tärkeisiin järjestelmän toimintoihin.
FB = Functions Blocks. Toimintalohkot	FB:t ovat ohjelmoitavia lohkoja muistiominaisuudella ts. muuttujien arvot säilyvät, vaikka lohkon suorituksesta poistutaan.
FC = FunCtions. Tehtävät.	FC:t sisältävät usein toistettavia ohjelmarutiineja. Muuttujien arvot eivät säily.
Instance DB = Instance Data Blocks.	Ilmentymä DB:t assosioivat lohkojen

Ilmentymä datalohko	kanssa kun FB/SFB:tä kutsutaan. Ne luodaan automaattisesti käännöksen aikana.
DB = D ata B lock. Datalohko	DB:t ovat käyttäjädatan talletusalueita

3.7.1 Step 7-ohjelmalohkot

OB:t edustavat liityntää käyttöjärjestelmän ja käyttäjäohjelman välillä. Käyttöjärjestelmän kutsumina ne ohjaavat syklistä ja keskeytyspohjaista ohjelman suoritusta, logiikan käyttäytymistä käynnistymisessä ja virheiden käsittelyä. OB:t päättävät yksittäisten ohjelmaosoiden suoritussekvenssin. Korkeamman prioriteetin OB voi keskeyttää alemman prioriteetin OB:n. OB1:ssä on pääohjelman suoritus. Sen prioriteetti on 1.

SFB:t ovat S7 CPU:hun esiohjelmoituja järjestelmätoimintalohkoja, joita voidaan kutsua käyttäjäohjelmassa. SFB:t ovat osa käyttöjärjestelmää, eikä niitä ladata ohjelman osana. Tarjolla on lohkot:

1. kommunikointiin konfiguroituihin liitännöihin
2. integroitua erityistoimintoja varten.

Myös **SFC**:t ovat S7 CPU:hun esiohjelmoituja järjestelmätoimintoja, joita voidaan kutsua käyttäjäohjelmassa. Tarjolla on lohkot:

1. kopioimiseen ja lohko-toimintoihin
2. ohjelman tarkistamiseen
3. kellon käsittelyyn ja suorituksen aikaisiin mittareihin
4. tietokokonaisuuksien (data set) siirtoon
5. tapahtumien siirtoon CPU:lta toisille monilaskentamoodissa
6. päiväajan ja aikaviivästettyjen keskeytysten hallintaan
7. synkronisten virheiden, keskeytysten ja asynkronisten virheiden hallintaan
8. staattisen ja dynaamisen järjestelmädatan tietoon
9. prosessikuvan päivitykseen ja bittikentän prosessointiin
10. modulien osoitukseen
11. hajautettuun IO:hon
12. globaalidata kommunikointiin
13. kommunikointiin ei-konfiguroitujen liitäntöjen kautta
14. lohkoihin liittyvien viestien generointiin.

FB:t ovat lohkoja, jotka ohjelmoidaan itse ja joilla on muistia. Niissä on ohjelma, joka suoritetaan aina, kun FB:tä kutsutaan eri loogisten lohkojen toimesta. Pysyvät muuttujat ja FB:hen siirretyt parametrit talletetaan ilmentymä DB:hen, josta talletettu tieto ei häviä, kun FB:n suoritus on valmistunut. Väliaikaiset muuttujat talletetaan paikalliseen datapinoon. Paikalliseen datapinoon talletettu tieto katoaa FB:n suorituksen jälkeen. Jokaiseen FB-kutsuun joka siirtää parametrejä, on liitetty ilmentymä DB. Kutsumalla useampaa kuin yhtä FB:n ilmentymää, voi ohjata useaa laitetta yhdellä FB:llä. Esimerkiksi moottori FB voi ohjata montaa erilaista moottoria käyttämällä eri ilmentymä DB asetuksia.

Ilmentymä **DB**:t liitetään jokaiseen parametrejä siirtävään toimintolohkoon. FB:n aktuaaliparametrit ja pysyvä data talletetaan ilmentymä DB:hen. Ilmentymä tarkoittaa toimintolohkon kutsua. Jos toimintolohkoa kutsutaan S7 ohjelmassa esimerkiksi viisi kertaa, on toimintolohkosta viisi ilmentymää. Ennen ilmentymä DB:n luomista vastaavan FB:n täytyy jo olla olemassa. DB:t eivät sisällä S7 komentoja. [8.]

3.7.2 Step 7 datatyypit

Kaikki ohjelmassa käytetty data täytyy identifioida tyypiltään. Tämä tarkoittaa muuttujan tyypin määrittämistä sen esittelyssä. Saatavilla on Step 7:n tarjoamat perustyytit, itseluotavat monimutkaiset tyypit ja parametrityypit, joilla määritellään FB:iin tai FC:iin siirrettävät parametrit. Tämä raportti koskettaa vain käyttäjän datatyyppiä (UDT) rajaussyistä ja siksi, että sellaista on käytetty kommunikaation luomiseksi ohjaavan logiikan ja servo-ohjainyksikön välille.

UDT:t ovat käyttäjän luomia tietotyyppiejä, joilla voidaan yhdistää perus- ja monimutkaiset tyypit yhdeksi (Kuvio 28).

```
TYPE UDT20
  STRUCT
    start: BOOL;
    setp: INT;
    value: WORD;
  END_STRUCT;
END_TYPE
```

Kuvio 28. UDT esimerkki

Servo-ohjainyksikön kovomäärittelyssä jokaiselle servoakselille on tiedonsiirtoa varten määritelty Siemensin tarjoama standardi PROFIdrive telegram 111, PDZ-12/12. Tämä telegrammi määrittelee, mitä parametrin arvoja ohjauksen ja servoakselin välillä siirretään hallitun paikoitusliikkeen tuottamiseksi ja monitoroimiseksi. PROFIdrive telegram 111:n rakenne on taulukoissa 2. ja 3.

Taulukko 2. Servo-ohjaimen lähettämä data Telegram 111:ssä

Rakennimenimi	Kuvaus	Tyyppi
SW1	Tärkein tilasana	BOOL
PosSW1	EPOS tilasana 1	BOOL
PosSW2	EPOS tilasana 2	BOOL
SW2	Tilasana 2	BOOL
MeldW	Sähkökäytön tila	BOOL
ActPos	Paikan oloarvo	DINT
ActSpd	Nopeuden oloarvo	DINT
FaultCode	Virhekoodi	WORD
AlarmCode	Hälytyskoodi	WORD
Res	Varalla. Käyttäjän määriteltävissä	WORD

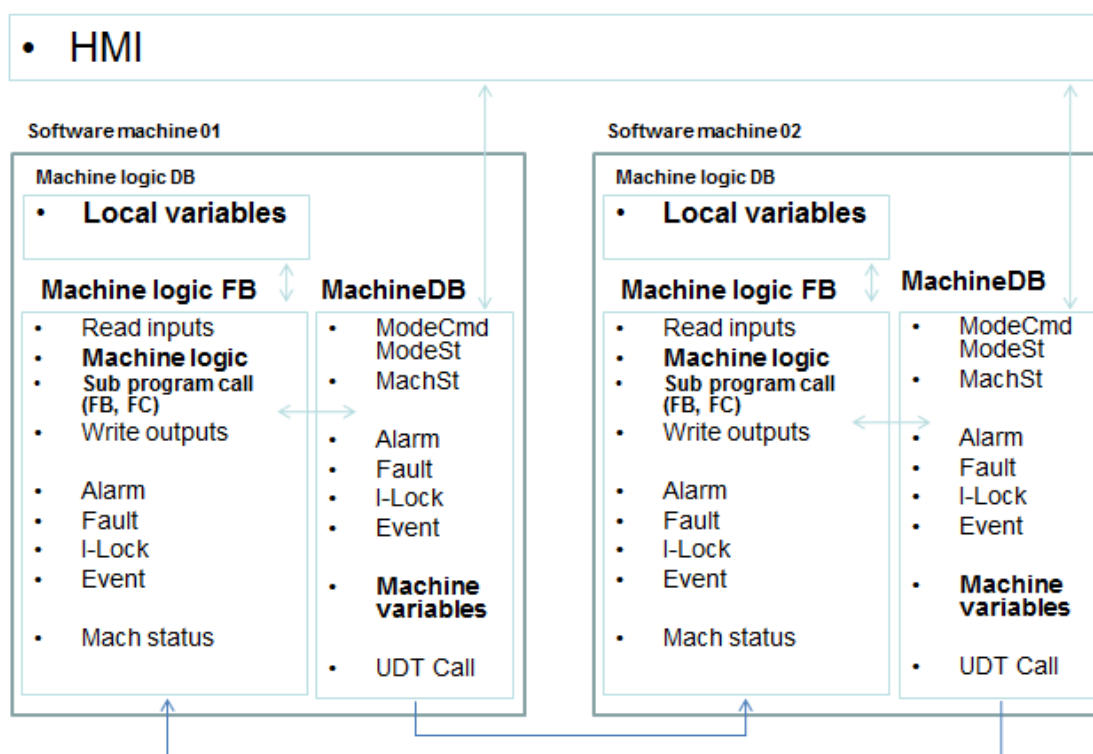
Taulukko 3. PLC:n lähettämä data Telegram 111:ssä

Rakennimenimi	Kuvaus	Tyyppi
CW1	Tärkein ohjaussana	BOOL
PosCW1	Paikoituksen ohjaussana 1	BOOL
PosCW2	Paikoituksen ohjaussana 2	BOOL
CW2	Ohjaussana 2	BOOL
OVERRIDE	EPOS Nopeuden ylikirjoitus	INT
MDIPos	EPOS Paikan asetusarvo	DINT
MDIVel	EPOS Nopeuden asetusarvo	DINT
MDIAcc	EPOS Kiihtyvyyden ylikirjoitus	INT
MDIDec	EPOS Hidastuvuuden ylikirjoitus	INT
Res	Varalla. Käyttäjän määriteltävissä.	WORD

Projektille ohjelmoitu UDT Telegram 111 (liite 17) luo tietotyypin, jolla tarpeelliset tila- ja komentosanat liikkeen toteuttamiseksi voidaan siirtää. Tätä tietotyyppiä hyväksikäyttäen on luotu paikoitustoimintolohko SinaPosFB (FB105) tulo- ja lähtömuuttujineen (liite 18). Jokainen servoakseli kutsuu tätä paikoitustoimintolohkoa oman LogicFB:n kautta (liite 5, 7, 9, 11).

3.7.3 Projektin ohjelmarakenne ja lohkot

Kuvio 29 havainnollistaa projektissa käytetyn ohjelmapihjan ohjelmarakenneperiaatetta. Jokaiselle servoakselille luodaan Step 7:llä datalohko (MachineDB), logiikkatoimintalohko (Machine logic FB) sekä logiikkatoimintalohkon datalohko (Machine logic DB).



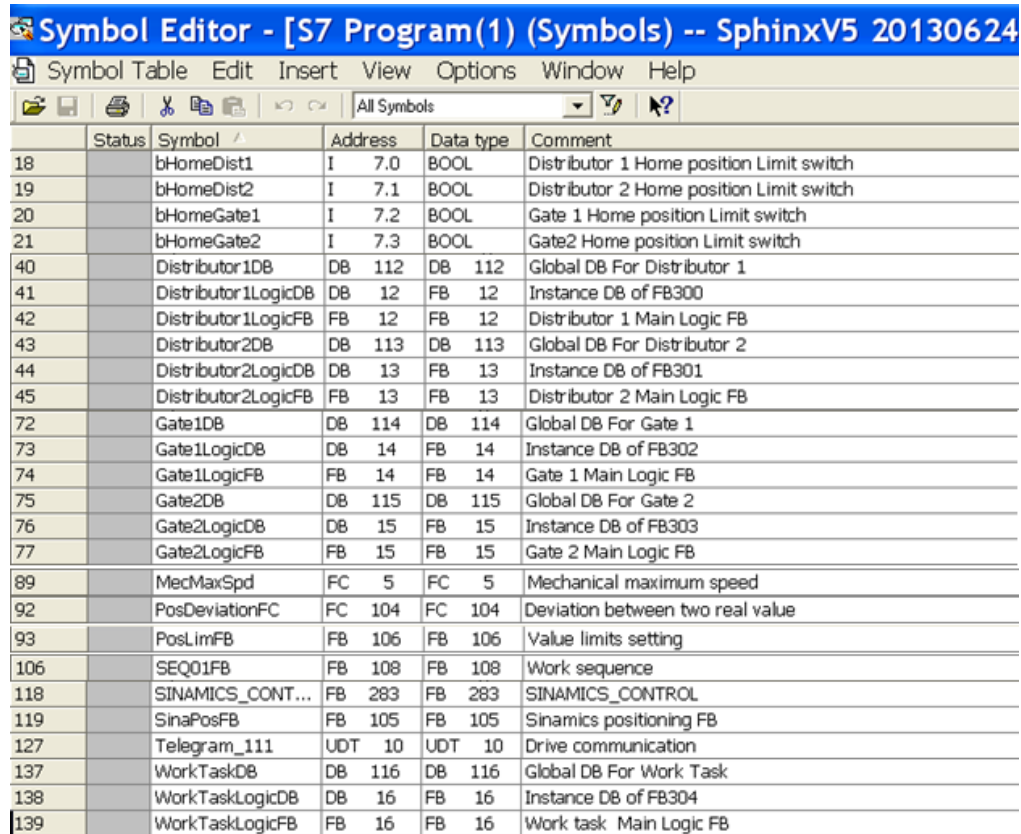
Kuvio 29. Ohjelmarakenteen periaatekaavio

Ohjelmapihjassa on valmiina ohjelmalohkoja, joista voi muokata käyttöön sopivan. Nämä ohjelmalohkot sisältävät jo valmiiksi muun muassa vikailmoitusten ja hälytysten hallinnan. Valmiita lohkoja tai niissä olleita toimintoja ei kuvata tässä raportissa. Tällainen DB Base pohja kopioitiin ja muokattiin eri akselien DB:ksi. Purkajaluukun servo 1:n DB nimettiin DB Distributor 1:ksi, servo 2:n DB Distributor 2:ksi, purkajan sektoriluukku 1:n DB nimettiin DB Gate 1:ksi ja 2:n DB Gate 2:ksi (liite 4, 6, 8, 10).

Step 7 ohjelmassa työskennellään I/O signaalien, bittimuistien, laskureiden, ajastimien, tietolohkojen ja toimintolohkojen osoitteiden kanssa. Nämä osoitteet ovat tavoitettavissa absoluuttisesti. Absoluuttiosoite koostuu osoitetunnisteesta ja

muistisijainnista (esimerkiksi Q 4.0, I 1.1). Liittämällä absoluuttiosoitteeseen symboli on ohjelma paljon helpompi lukea ja ongelmanratkaisu yksinkertaistuu huomattavasti. Tätä tarkoitusta varten Step 7:ssä on symbolitaulu.

Projektin symbolitauluun lisätyt symbolinimet, osoitteet ja datatyypit ovat kuviossa 30.

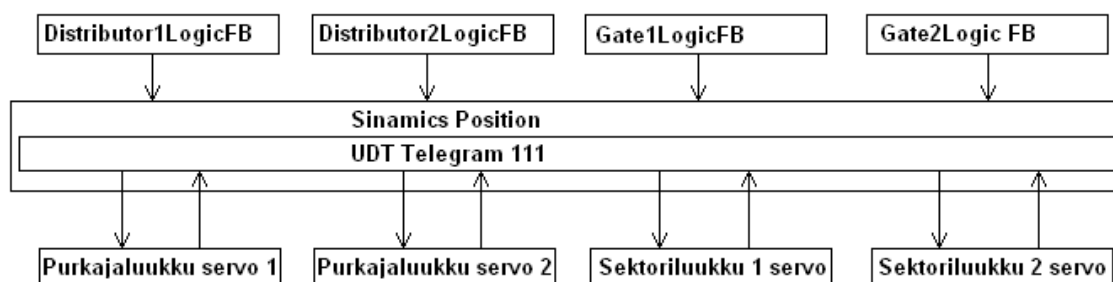


	Status	Symbol	Address	Data type	Comment
18		bHomeDist1	I 7.0	BOOL	Distributor 1 Home position Limit switch
19		bHomeDist2	I 7.1	BOOL	Distributor 2 Home position Limit switch
20		bHomeGate1	I 7.2	BOOL	Gate 1 Home position Limit switch
21		bHomeGate2	I 7.3	BOOL	Gate2 Home position Limit switch
40		Distributor1DB	DB 112	DB 112	Global DB For Distributor 1
41		Distributor1LogicDB	DB 12	FB 12	Instance DB of FB300
42		Distributor1LogicFB	FB 12	FB 12	Distributor 1 Main Logic FB
43		Distributor2DB	DB 113	DB 113	Global DB For Distributor 2
44		Distributor2LogicDB	DB 13	FB 13	Instance DB of FB301
45		Distributor2LogicFB	FB 13	FB 13	Distributor 2 Main Logic FB
72		Gate1DB	DB 114	DB 114	Global DB For Gate 1
73		Gate1LogicDB	DB 14	FB 14	Instance DB of FB302
74		Gate1LogicFB	FB 14	FB 14	Gate 1 Main Logic FB
75		Gate2DB	DB 115	DB 115	Global DB For Gate 2
76		Gate2LogicDB	DB 15	FB 15	Instance DB of FB303
77		Gate2LogicFB	FB 15	FB 15	Gate 2 Main Logic FB
89		MecMaxSpd	FC 5	FC 5	Mechanical maximum speed
92		PosDeviationFC	FC 104	FC 104	Deviation between two real value
93		PosLimFB	FB 106	FB 106	Value limits setting
106		SEQ01FB	FB 108	FB 108	Work sequence
118		SINAMICS_CONT...	FB 283	FB 283	SINAMICS_CONTROL
119		SinaPosFB	FB 105	FB 105	Sinamics positioning FB
127		Telegram_111	UDT 10	UDT 10	Drive communication
137		WorkTaskDB	DB 116	DB 116	Global DB For Work Task
138		WorkTaskLogicDB	DB 16	FB 16	Instance DB of FB304
139		WorkTaskLogicFB	FB 16	FB 16	Work task Main Logic FB

Kuvio 30. Step 7 symbolitaululisäykset

3.7.4 Paikoitustoimintolohko

Paikoitustoimintolohko (liite 18) Sinamics position (FB105 SinaPosFB) perustuu UDT Telegram 111:een (liite 17) ohjelmoituun PROFIdrive telegram 111:een ja se ohjelmoitiin projektille luomaan liityntä servo-ohjaimiin. Jokaisen servoakselin (purkajaluukun ja sektoriluukkujen servot) LogicFB (liite 5, 7, 9, 11) kutsuu tätä toimintolohkoa toteuttaessaan paikoitusliikkeitä (Kuvio 31).



Kuvio 31. Paikoitustoimilohkoperiaate

3.7.5 Servo-ohjaimien paikkalaskurien skaalaus liikealueelle

Servoakseleiden paikan skaalaamiseksi liikealueelle, käyttäjän paremmin ymmärtämään mittayksikköön ohjelmoitiin projektille toimintolohko Position limits FB (FB106 PosLimFB, liite 12). Skaalausta varten lohkolle syötetään akselin minimi ja maksimi liikealue, olopaikka sekä servo-ohjaimelta STARTER käyttöönotto-ohjelmalla luetut, ohjaimen paikkalaskuriin perustuvat minimi- ja maksiarvot sekä paikkalaskurin kertoma paikka (3.5.7 HMI, raja-arvoskaalaussivu). Jokaisen servoakselin LogicFB (liite 5, 7, 9, 11) kutsuu suorituksessaan tätä skaalaustoimintolohkoa.

3.7.6 Servoakseleiden loogiset toimintolohkot

Purkaja- ja sektoriluukkujen servojen toimintoja varten ohjelmoitiin toimintolohkot Distributor1LogicFB, Distributor2LogicFB, Gate1LogicFB ja Gate2LogicFB (liite 5, 7, 9, 11). Tätä varten kopioitiin ohjelmapohjasta BaseLogicFB:n josta muokattiin kullekin akselille omansa. BaseLogicFB sisältää valmiiksi toimintoja esimerkiksi vika-, virhe ja tilatietojen välittämiseksi HMI:lle. Lohkoihin tehtiin lisäykset: servo-ohjaimen paikkalaskurin skaalaus halutulle liikealueelle, servo-ohjaimen paikkasäädön päälle kytkentä (enable), paikoituksen päälle/pois kytkentä, käsiajo eteen / taakse, paikoitusajo, nopeuden yksikkömuunnos ja kotiraja.

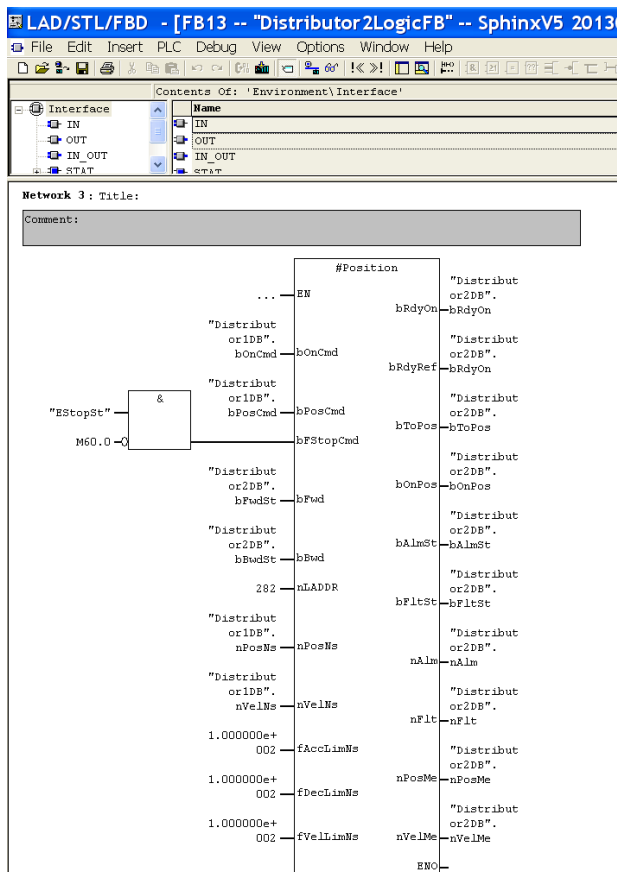
3.7.7 Purkajaluukun servojen yhteiskäyttö

Purkajaluukun paikan säätämiseksi kahdella servomootorilla oli mahdollista valita vaihtoehtoista:

1. Servot kytkettäisiin sähköisesti periaatteella master – slave jossa isäntä antaisi orjalleen momenttiohjeen. Tämä menettely sopii erityisesti mekanismeihin, jossa servoakseleiden välinen mekaaninen kytkentä on täydellisen jäykkä. Esimerkiksi jos servomootorit ovat kiinni samassa akselissa.
2. Servot toteuttaisivat itsenäisesti samaa kohdepaikka-asetusta. Kun liikkeensuorituksen parametreilla (nopeus, kiihtyvyys, hidastuvuus, kohdepaikka, liikkeensuorituksen aikaiset rasitukset kuten kitka ja kuorma) on tarkalleen samat arvot, pitäisi myös paikkasäädön toteutua saman aikaisesti. Tämän menettelyn vaarana on, että servomootorit alkavat toimia toisiaan vastaan mikäli niiden välinen kytkentä on liian jäykkä tai kuorma jakautuu epätasaisesti.

Konepajalla tehtyjen koeajojen perusteella tultiin siihen tulokseen, että vaihtoehto 2 sopii käytettäväksi akseleiden ja mekaniikan välisen joustavan kytkennän ja suuren välityksen vuoksi. Kummankin akselin liikkeen aikaista paikkavirhettä seuraamalla pääteltiin, etteivät akselit säädä paikkaansa toisiaan vastaan. Tämä asia jouduttanee kuitenkin varmistamaan vielä käyttöönotossa loppuasiakkaan luona, kun mekanisme kuormitetaan maksimikuormalla.

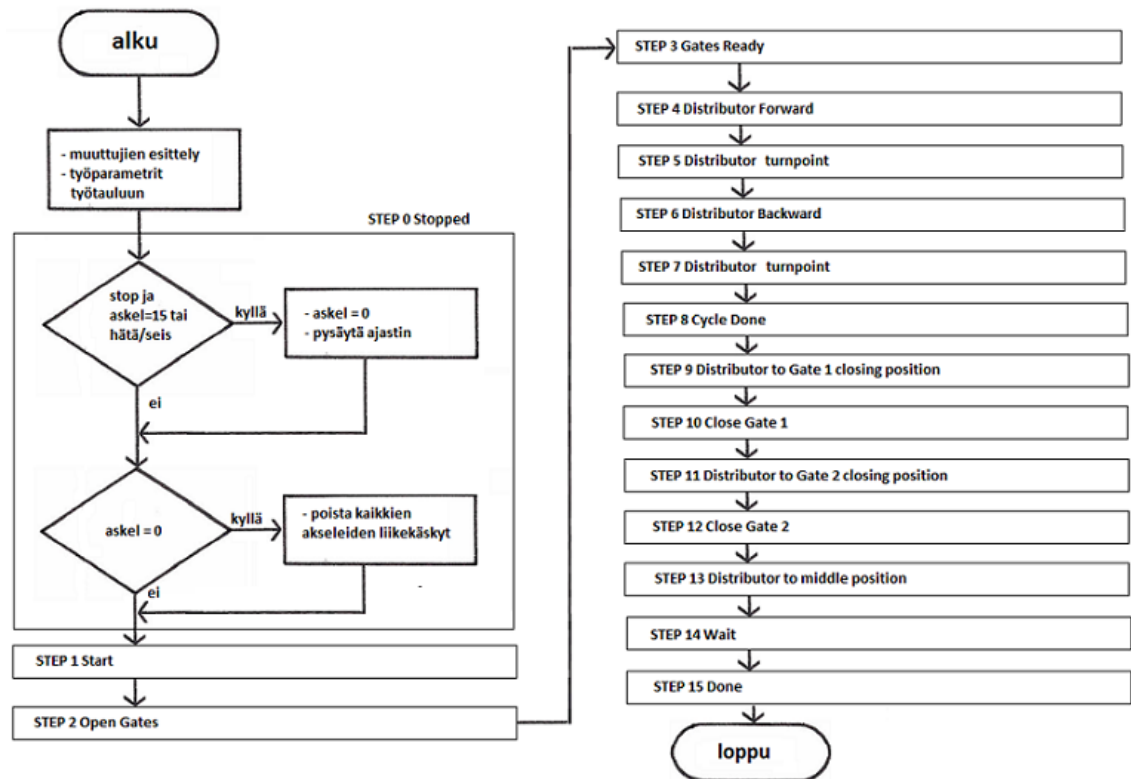
Käytännössä purkajaluukun servo 2:lle annetaan sen LogicFB:ssä servo 1:n kriittiset parametrit kuten *Distributor1DB.bOnCmd* eli ena, *Distributor1DB.bPosCmd* eli paikoitus päälle, *Distributor1DB.nPosNs* eli kohdepaikka ja *Distributor1DB.nVelNs* eli ajonopeus (Kuvio 32).



Kuvio 32. Purkajaluukun servo 1:n kriittiset liikeparametrit servo 2:lle

3.7.8 Työsekvenssi

Työsekvenssi (FB108 SEQ01FB, liite 13) on työkierron LogicFB:n (liite 16) kutsuma toimintolohko Outotecin spesifioiman työkierron toteuttamiseksi (3.1 Ohjelmatyökierron spesifikaatio). Vuokaavion lohkon STEP 0 logiikka toistuu muissa laatikoina kuvatuissa lohkoissa (Kuvio 33). Muuttuja 'askel' varmistaa, että sekvenssi tulee suoritettua loogisesti oikeassa järjestyksessä. Laatikoiden sisältö on kuvattu taulukko 4:ssä vuokaavion jälkeen.



Kuvio 33. Työsekvenssi

STEPit on kuvattu sanallisesti taulukossa 4. Taulukossa esiintyvä teksti-indeksi on HMI:n tekstitaulukon indeksi jolla päänäytössä juoksutetaan suorituksen alla olevan askeleen selväkielistä kuvausta (3.5.2 HMI, pääsivu). Soluissa olevat vaalean harmaat kenttäviitteet vastaavat vuokaavion nuolia.

Taulukko 4. Työsekvenssi

	1	2	3	4	5
a	STEP 0, Stopped	Ehto 1:	Stop ja askel = 15 tai hätä/seis ➡	Kyllä:	askel = 0, teksti-indeksi = 0, pysäytä ajastin (c2)
b		Ei:	STEP 0, Ehto 2 (c2)		
c		Ehto 2:	askel = 0	Kyllä:	poista liikekäskyt (e2)
d		Ei:	STEP 1 (e2)		
e	STEP 1, Start	Ehto 1:	Start ja askel = 0 ja globaali käy bitti	Kyllä:	askel = 1, teksti-indeksi = 1 pysäytä ajastin (g2)

f		Ei:	STEP 0, Ehto 2 (g2)		
g		Ehto 2:	askel = 1	Kyllä:	käynnistä ajastin (i2)
h		Ei:	STEP 2 (i2)		
i	STEP 2, Open Gates	Ehto 1:	askel = 1 ja ajastin $\geq 500\text{ms}$	Kyllä:	askel = 2, teksti-indeksi = 2, pysäytä ajastin (k2)
j		Ei:	STEP 2, Ehto 2 (k2)		
k		Ehto 2:	askel = 2	Kyllä:	käynnistä ajastin, asetta kohdepaikat, asetta ajonopeus, käynnistä liikkeet (m2)
l		Ei:	STEP 3 (m2)		
m	STEP 3, Gates Ready	Ehto 1:	ajastin $\geq 500\text{ms}$ ja paikat valmiit	Kyllä:	askel = 3, teksti-indeksi = 3, pysäytä ajastin (o2)
n		Ei:	STEP 3, Ehto 2 (o2)		
o		Ehto 2:	askel = 3	Kyllä:	käynnistä ajastin, poista liikekäskyt (p2)
p		Ei:	STEP 4 (q2)		
q	STEP 4, Distributor Forward	Ehto 1:	ajastin $\geq 500\text{ms}$ ja askel = 3 tai askel = 8	Kyllä:	askel = 4, teksti-indeksi = 4, pysäytä ajastin (s2)
r		Ei:	STEP 4, Ehto 2 (s2)		
s		Ehto 2:	askel = 4	Kyllä:	käynnistä ajastin, asetta kohdepaikat, asetta ajonopeus, käynnistä liikkeet (u2)
t		Ei:	STEP 5 (u2)		
u	STEP 5, Distributor Forward turn point	Ehto 1:	ajastin $\geq 500\text{ms}$ ja paikat valmiit ja askel = 4	Kyllä:	askel = 5, teksti-indeksi = 5, pysäytä ajastin (x2)
v		Ei:	STEP 5, Ehto 2 (x2)		
x		Ehto 2:	askel = 5	Kyllä:	käynnistä ajastin, asetta kohde- parametrit (z2)
y		Ei:	STEP 6 (z2)		
z	STEP 6, Distributor Backward	Ehto 1:	odotusaika kulunut ja askel = 5	Kyllä:	askel = 6, teksti-indeksi = 6, pysäytä ajastin (ab2)
aa		Ei:	STEP 6, Ehto 2		

			(ab2)		
ab		Ehto 2:	askel = 6	Kyllä:	käynnistä ajastin, aseta kohdepaikat, aseta ajonopeus, käynnistä liikkeet (ad2)
ac		Ei:	STEP 7 (ad2)		
ad	STEP 7, Distributor Backward turn point	Ehto 1:	ajastin \geq 500ms ja paikat valmiit ja askel = 6	Kyllä:	askel = 7, teksti-indeksi = 7, pysäytä ajastin (af2)
ae		Ei:	STEP 7, Ehto 2 (af2)		
af		Ehto 2:	askel = 7	Kyllä:	käynnistä ajastin, aseta kohdeparametrit (ah2)
ag		Ei:	STEP 8 (ah2)		
ah	STEP 8, Cycle Done	Ehto 1:	odotusaika kulunut ja askel = 7	Kyllä:	askel = 8, teksti-indeksi = 8, pysäytä ajastin (aj2)
ai		Ei:	STEP 8, Ehto 2 (aj2)		
aj		Ehto 2:	askel = 8	Kyllä:	käynnistä ajastin (al2)
ak		Ei	STEP 9 (al2)		
al	STEP 9, Distributor to Gate 1 closing position	Ehto 1:	Stop ja askel \geq 2 ja askel \leq 8	Kyllä:	askel = 9, teksti-indeksi = 9, pysäytä ajastin (an2)
am		Ei	STEP 9, Ehto 2 (an2)		
an		Ehto 2:	askel = 9	Kyllä:	käynnistä ajastin, aseta kohdepaikat, aseta ajonopeus, käynnistä liikkeet (ap2)
ao		Ei:	STEP 10 (ap2)		
ap	STEP 10, Close Gate 1	Ehto 1:	ajastin \geq 500ms ja paikat valmiina ja askel = 9	Kyllä:	askel = 10, teksti-indeksi = 10, käynnistä ajastin (ar2)
aq		Ei:	STEP 10, Ehto 2 (ar2)		
ar		Ehto 2:	askel = 10	Kyllä:	asetta kohdepaikka, aseta ajonopeus, käynnistä liike (at2)
as		Ei:	STEP 11 (at2)		
at	STEP 11, Distributor	Ehto 1:	ajastin \geq 500ms ja paikka valmis ja	Kyllä:	askel = 11, teksti-indeksi =

	to Gate 2 closing position		askel = 10		11, pysäytä ajastin (av2)
au		Ei:	STEP 11, Ehto 2 (av2)		
av		Ehto 2:	askel = 11	Kyllä:	käynnistä ajastin, aseta kohdepaikat, aseta ajonopeudet, käynnistä liikkeet (ay2)
ax		Ei:	STEP 12 (ay2)		
ay	STEP 12, Close Gate 2	Ehto 1:	ajastin \geq 500ms ja paikat valmiit ja askel = 11	Kyllä:	askel = 12, teksti-indeksi = 12, pysäytä ajastin (ba2)
az		Ei:	STEP 12, Ehto 2 (ba2)		
ba		Ehto 2:	askel = 12	Kyllä:	käynnistä ajastin, aseta kohdepaikka, aseta ajonopeus, käynnistä liike (bc2)
bb		Ei:	STEP 13 (bc2)		
bc	STEP 13, Distributor to middle position	Ehto 1:	ajastin \geq 500ms ja paikka valmis ja askel = 12	Kyllä:	askel = 13, teksti-indeksi = 13, pysäytä ajastin (be2)
bd		Ei:	STEP 13, Ehto 2 (be2)		
be		Ehto 2:	askel = 13	Kyllä:	käynnistä ajastin, aseta kohdepaikat, aseta ajonopeudet, käynnistä liikkeet (bg2)
bf		Ei:	STEP 14 (bg2)		
bg	STEP 14, Wait	Ehto 1:	ajastin \geq 500ms ja paikat valmiit ja askel = 13	Kyllä:	askel = 14, teksti-indeksi = 14, pysäytä ajastin (bi2)
bh		Ei:	STEP 14, Ehto 2 (bi2)		
bi		Ehto 2:	askel = 14	Kyllä:	käynnistä ajastin, pysäytä liikkeet (bk2)
bj		Ei:	STEP 15 (bk2)		
bk	STEP 15, Done	Ehto 1:	ajastin \geq 500ms ja askel = 14	Kyllä:	askel = 15, teksti-indeksi = 15, pysäytä ajastin (bm2)

bl		Ei:	STEP 15, Ehto 2 (bm2)		
bm		Ehto 2:	askel = 15	Kyllä:	käynnistä ajastin, asetta 'tehty'
bn		Ei:	Sekvenssin loppu		

Työsekvenssiä kutsutaan toimintalohkosta WorkTaskLogicFB (liite 16).

3.7.9 Työkierron toimintolohko

Työkierron toimintolohko (FB16 WorkTaskLogicFB, liite 16) kutsuu työsekvenssiä saadessaan HMI:ltä komennon käynnistyksestä. Myös purkajaluukun servojen välistä paikkaeroa valvotaan tässä ohjelmalohkossa kutsumalla toimintoa FC Position Deviation (FC104 PosDeviationFC, liite 14).

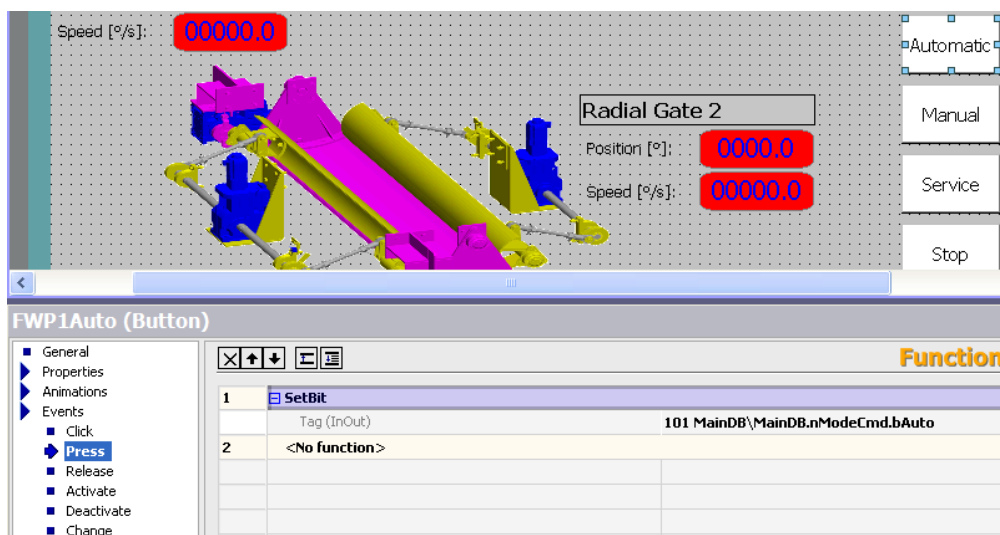
3.7.10 Lisäykset rakennetoimintolohkoon (OB)

Projektin ohjelmasuorituksen rakennetoimintolohko määrää ohjelmasuorituksen järjestyksen, joten siihen lisättiin tehtyjen ohjelmatoimintolohkojen kutsut (liite 19).

3.7.11 Signaalin kulku operaattorilta toimilaitteelle

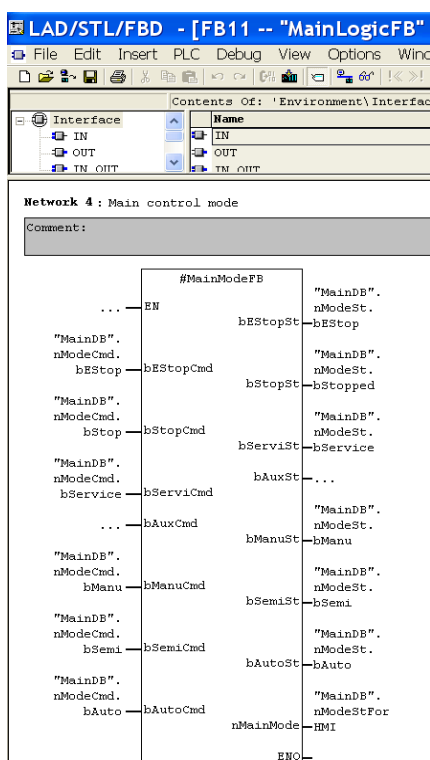
Selvyyden vuoksi kuvataan tässä kappaleessa vielä yhteenvetomaisesti miten käsky kulkee laitetta käyttävän operaattorin toimintonäppäimen painalluksesta kohdetoimilaitteelle, esimerkkinä automaattityökierron käynnistys ja sen toteuttama sektoriluukun 1 avaus. Osin tämän kuvauksen periaatetta visualisoitu kuviossa 31.

1. Käyttäjä painaa käyttöpääteen toimintonäppäintä *Automatic*
2. Toimintonäppäin asettaa MainDB:n bitin *MainDB.nModeCmd.bAuto* ykköseksi (Kuvio 34).



Kuvio 34. HMI, päänäyttö, Automatic toimintonäppäimen suorittama asetus

3. Käyttöpääte kommunikoi profinet liitännällään ohjauslogiikan CPU:lle (liite 3, s.12).
4. OB1:n kutsuma MainLogicFB (FB11) kutsuu MainModeFB:tä (FB111, Kuvio 35).



Kuvio 35. Step 7 MainLogicFB kutsuu MainModeFB:tä, MainDB.nModeCmd.bAuto = 1

5. MainModeFB asettaa bAutoCmd (= *MainDB.nModeCmd.bAuto*) komennolla muuttujansa nMode arvoksi 64, nollaa aiemmat komento- ja tilatiedot ja asettaa CASE muuttujan avulla bAutoSt muuttujan arvoksi 1 (Kuvio 36).

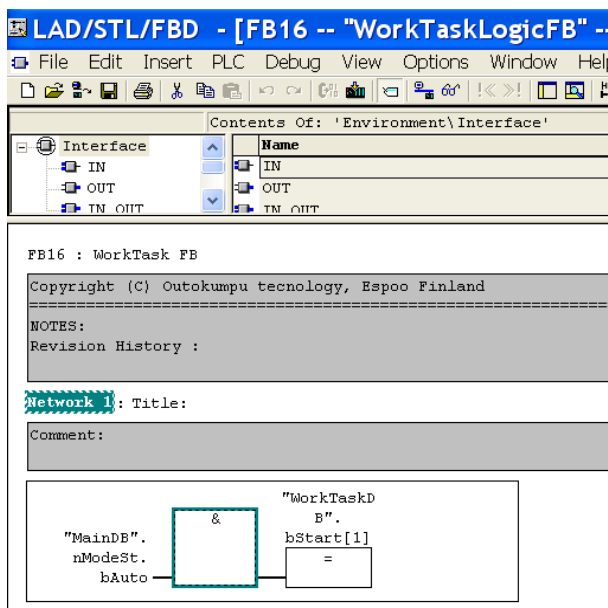
```

SCL - [FB Main Mode --
// Control mode selection
IF bEStopCmd THEN
  nMode := 1;
ELSIF bStopCmd THEN
  nMode := 2;
ELSIF bServicCmd THEN
  nMode := 4;
ELSIF bAuxCmd THEN
  nMode := 8;
ELSIF bManuCmd THEN
  nMode := 16;
ELSIF bSemiCmd THEN
  nMode := 32;
ELSIF bAutoCmd THEN
  nMode := 64;
END_IF;
// Reset of main mode commands
bEStopCmd := FALSE;
bStopCmd := FALSE;
bServicCmd := FALSE;
bAuxCmd := FALSE;
bManuCmd := FALSE;
bSemiCmd := FALSE;
bAutoCmd := FALSE;
// Reset of previous main mode
bEStopSt := FALSE;
bStopSt := FALSE;
bServicSt := FALSE;
bAuxSt := FALSE;
bManuSt := FALSE;
bSemiSt := FALSE;
bAutoSt := FALSE;
// Create new main mode
CASE DINT_TO_INT(nMode) OF
  1:
    bEStopSt := TRUE;
  2:
    bStopSt := TRUE;
  4:
    bServicSt := TRUE;
  8:
    bManuSt := TRUE;
  16:
    bManuSt := TRUE;
  32:
    bSemiSt := TRUE;
  64:
    bAutoSt := TRUE;
END_CASE;

```

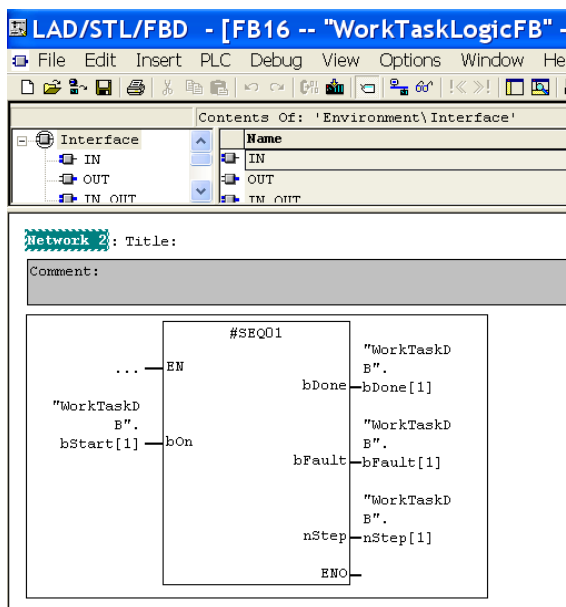
Kuvio 36. MainMode FB

6. MainLogicFB:ssä muuttuja *MainDB.nModeSt.bAuto* saa arvokseen *MainModeFB.bAutoSt*:n 1:sen (Kuvio 35).
7. *MainDB.nModeSt.bAuto* asettaa WorkTaskLogicFB:ssä muuttujan *WorkTaskDB.bStart[1]* ykköseksi (Kuvio 37).



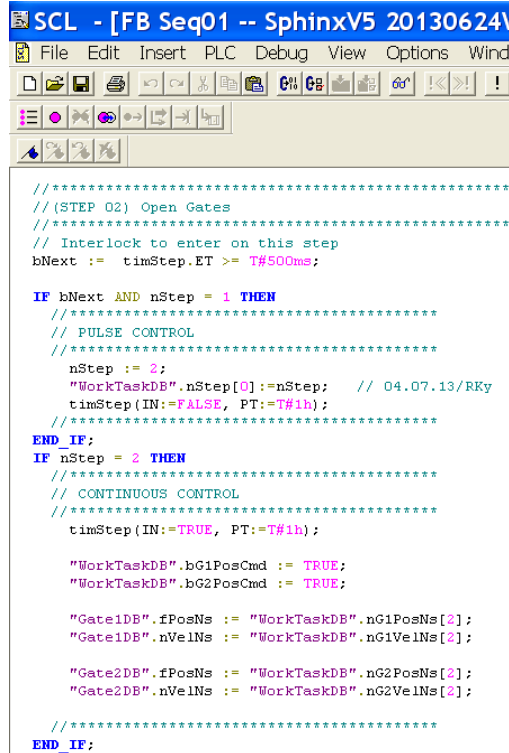
Kuvio 37. WorkTaskLogicFB

8. WorkTaskLogicFB kutsuu FB SEQ01:stä ja käynnistää työsekvenssin asettaen sen *bOn* muuttujalle *WorkTaskDB.bStart[1]* arvon 1 (Kuvio 38).



Kuvio 38. WorkTaskLogicFB kutsuu työsekvenssiä

9. Työsekvenssi FB SEQ01 etenee sektoriluukkujen avaamiseen ja antaa WorkLogicFB:lle komennon muuttujan *WorkTaskDB.bG1PosCmd* välityksellä käskyn ajaa sektoriluukkuja (Kuvio 39).



```

SCL - [FB Seq01 -- SphinxV5 20130624\
File Edit Insert PLC Debug View Options Wind
// *****
// (STEP 02) Open Gates
// *****
// Interlock to enter on this step
bNext := timStep.ET >= T#500ms;

IF bNext AND nStep = 1 THEN
// *****
// PULSE CONTROL
// *****
nStep := 2;
"WorkTaskDB".nStep[0] := nStep; // 04.07.13/RKy
timStep(IN:=FALSE, PT:=T#1h);
// *****
END_IF;
IF nStep = 2 THEN
// *****
// CONTINUOUS CONTROL
// *****
timStep(IN:=TRUE, PT:=T#1h);

"WorkTaskDB".bG1PosCmd := TRUE;
"WorkTaskDB".bG2PosCmd := TRUE;

"Gate1DB".fPosNs := "WorkTaskDB".nG1PosNs[2];
"Gate1DB".nVelNs := "WorkTaskDB".nG1VelNs[2];

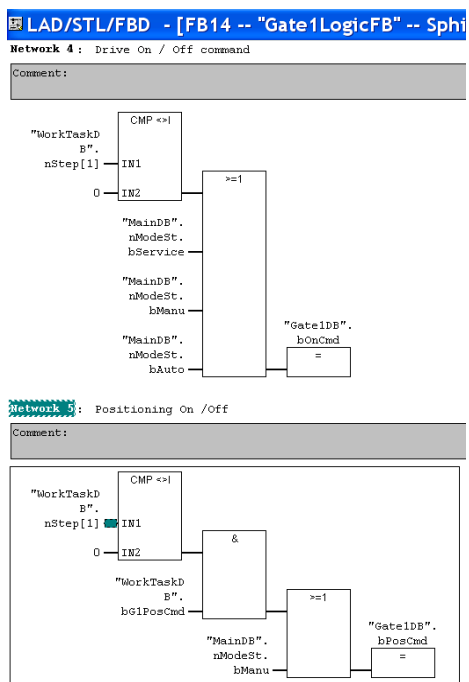
"Gate2DB".fPosNs := "WorkTaskDB".nG2PosNs[2];
"Gate2DB".nVelNs := "WorkTaskDB".nG2VelNs[2];

// *****
END_IF;

```

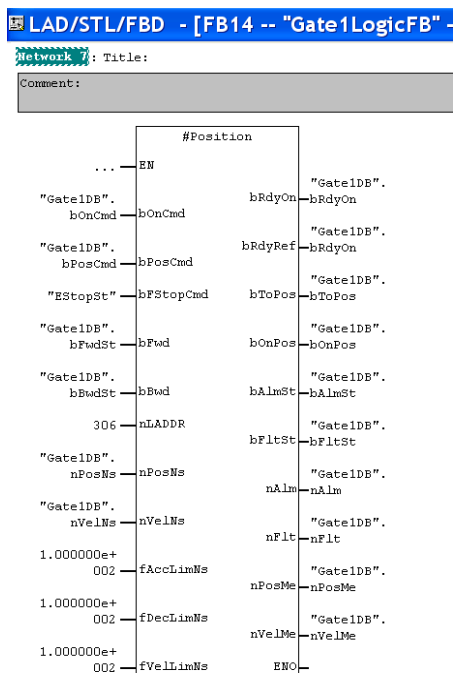
Kuvio 39. Työsekvenssi, sektoriluukkujen avaus

10. WorkTaskLogicFB:ssä *WorkTaskDB.nStep[1]*:lle annetaan FB SEQ01:n muuttujan *nStep* arvo 2 (Kuvio 38).
11. Gate1LogicFB:ssä *WorkTaskDB.nStep[1]* asettaa muuttujan *Gate1DB.bOnCmd* ykköseksi sektoriluukun 1 servo-ohjaimen enable signaalin kytkentää varten ja muuttujan *Gate1DB.bPosCmd* ykköseksi sektoriluukun 1 servo-ohjaimen paikkasäädön käynnistämiseksi (Kuvio 40).



Kuvio 40. GateLogicFB, enable- ja paikkasäätöbittien asetus

12. GateLogicFB kutsuu Sinamicsin paikoitustoimintolohkoa FB Sinamic Position (Kuvio 41).



Kuvio 41. Gate1LogicFB, Sinamics paikoitustoimintolohkon kutsu

13. Sinamics paikoitustoimintolohko asettaa UDT Telegram 111:n (Kuvio 42) muuttujat *Drv.Too.CW1.ON = bOnCmd* (asettaa servo-ohjaimessa parametrin p0840.00 [9, s. 358]) ja *Drv.Too.PosCW1.MDIStart = bPosCmd* (asettaa servo-ohjaimessa parametrin p2647 [9, s. 874]).

```

SCL - [FB SINAMICS POSITION -- SphinxV5 20130624\CPU\IM151-8F PN/DP CPU]
//*****
// READ TELEGRAM FROM SINAMICS
//*****

// Read telegram from drive
nRETVAL := DPRD_DAT( LADDR := INT_TO_WORD(nLADDR), RECORD := Drv.From );

//*****
// SINAMICS CW1
//*****

Drv.Too.CW1.Jog_1      := bBwd;           // Bit08 : p2589xxx EPOS jog 1 signal source
Drv.Too.CW1.Jog_2      := bFwd;           // Bit10 : p0854.00 PLC control request
Drv.Too.CW1.RefStart    := FALSE;         // Bit11 : p2595xxx EPOS referencing start
Drv.Too.CW1.res_12      := FALSE;         // Bit12 : xxxxxxxx Reserved
Drv.Too.CW1.BlockChange := FALSE;         // Bit13 : p2633xxx EPOS External block change
Drv.Too.CW1.res_14      := FALSE;         // Bit14 : xxxxxxxx Reserved
Drv.Too.CW1.res_15      := FALSE;         // Bit15 : xxxxxxxx Reserved
Drv.Too.CW1.ON          := bOnCmd;         // Bit00 : p0840.00 On / Off
Drv.Too.CW1.OFF2        := TRUE;          // Bit01 : p0844.00 Coast
Drv.Too.CW1.OFF3        := bFStopCmd;     // Bit02 : p0848.00 Quick stop
Drv.Too.CW1.Enable      := TRUE;          // Bit03 : p0852.00 Enable operation
Drv.Too.CW1.RejTrvTask   := TRUE;         // Bit04 : p2641xxx EPOS Reject traversing task
Drv.Too.CW1.IntMStop     := TRUE;         // Bit05 : p2640xxx EPOS Intermediate stop
Drv.Too.CW1.TrvStart     := FALSE;        // Bit06 : p2631xxx EPOS Activate traversing task
Drv.Too.CW1.AckFault     := bRetCmd;       // Bit07 : p2103.x0 Acknowledge faults

//*****
// SINAMICS POS CW1
//*****

Drv.Too.PosCW1.MDIPsTyp := TRUE;          // Bit08 : p2648xxx MDI Positioning type (1=Absolute, 0=Relative)
Drv.Too.PosCW1.MDIPos    := FALSE;        // Bit09 : p2651xxx MDI Positive direction
Drv.Too.PosCW1.MDINeg    := FALSE;        // Bit10 : p2652xxx MDI Negative direction
Drv.Too.PosCW1.res_11    := FALSE;        // Bit11 : xxxxxxxx Reserved
Drv.Too.PosCW1.MDITrTyp  := TRUE;         // Bit12 : p2649xxx MDI Transfer type (1=Continuous transfer)
Drv.Too.PosCW1.res_13    := FALSE;        // Bit13 : xxxxxxxx Reserved
Drv.Too.PosCW1.MDISetup  := FALSE;        // Bit14 : p2653xxx MDI setpoint acceptance edge
Drv.Too.PosCW1.MDIStart  := bPosCmd;      // Bit15 : p2647xxx MDI selection, In this mode make a flying changeover
Drv.Too.PosCW1.TrvBit_0  := FALSE;        // Bit00 : p2625xxx EPOS Traversing block selection Bit 0
Drv.Too.PosCW1.TrvBit_1  := FALSE;        // Bit01 : p2626xxx EPOS Traversing block selection Bit 1
Drv.Too.PosCW1.TrvBit_2  := FALSE;        // Bit02 : p2627xxx EPOS Traversing block selection Bit 2
Drv.Too.PosCW1.TrvBit_3  := FALSE;        // Bit03 : p2628xxx EPOS Traversing block selection Bit 3
Drv.Too.PosCW1.TrvBit_4  := FALSE;        // Bit04 : p2629xxx EPOS Traversing block selection Bit 4
Drv.Too.PosCW1.TrvBit_5  := FALSE;        // Bit05 : p2630xxx EPOS Traversing block selection Bit 5
Drv.Too.PosCW1.res_6     := FALSE;        // Bit06 : xxxxxxxx Reserved
Drv.Too.PosCW1.res_7     := FALSE;        // Bit07 : xxxxxxxx Reserved

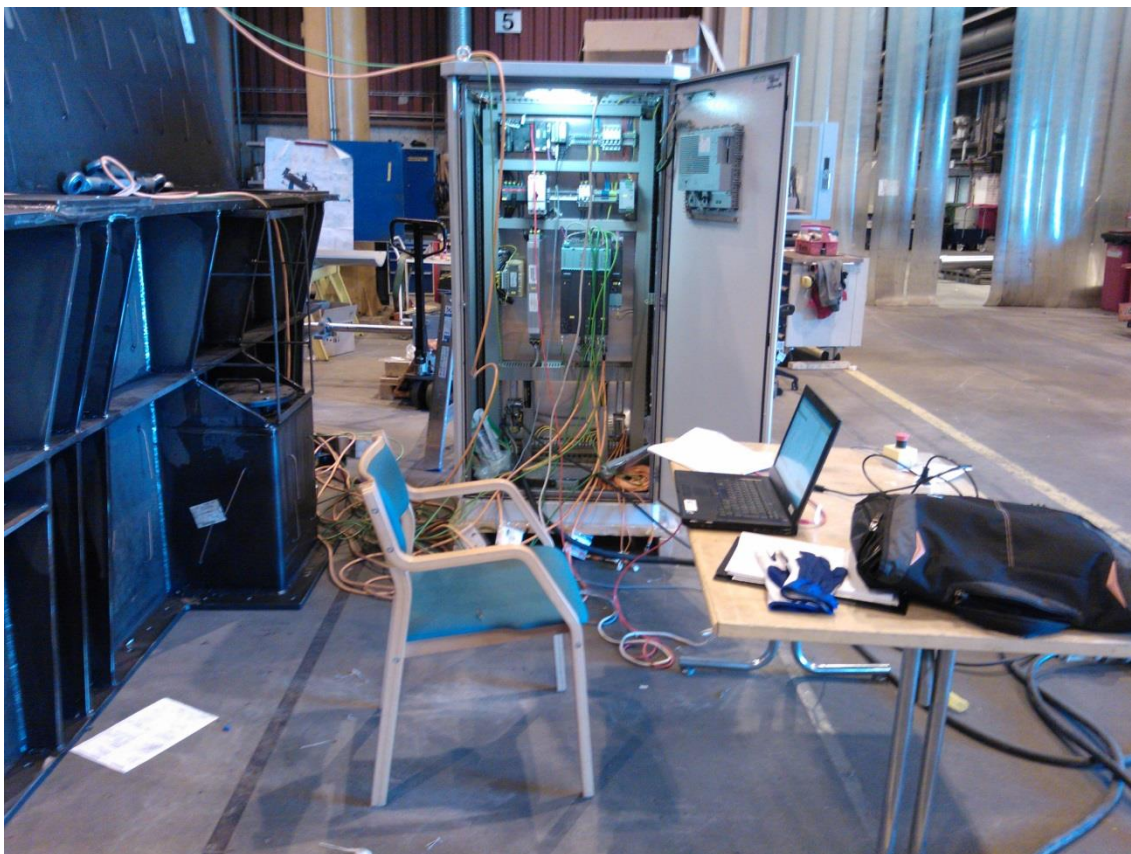
```

Kuvio 42. FB SINAMIC POSITION, UDT Telegram 111 asetukset

14. Liike käynnistyy ja sen toteutuminen raportoidaan samaa polkua takaisinpäin.

4 Käyttöönotto konepajalla

Kuvassa 43 on käyttöönottajän työpiste konepajalla, näkymä ohjauskaapista sekä vasemmassa reunassa osa jakopään runkoa.



Kuvio 43. Käyttöönotto konepajalla, työpistenäkymä 1

Testejä varten jakopää oli sijoitettu konepajan lattialle ylösalaisin. Kuvassa 44 on nähtävissä ohjauskaappi sivusuunnasta, ovi auki. Käyttöpäätteessä on auki pääsivu. Kuvassa 44 on takana olevassa jakopäässä nähtävissä edessä toisen sektoriluukun servomoottori kiinnitettynä vaihteeseen sekä sivuilla purkajaluukun vaihteet joissa takana kiinni servomoottorit.



Kuvio 44. Käyttöönotto konepajalla, työpistenäkymä 2

Käyttöönottoa varten tulee käytettävissä olla STARTER käyttöönottotyökaluohjelma ja sähkökäytön kytkennät tehtynä. Kaappivalmistajan ammattitaitoon kuuluu varmistaa komponenttien asennuksen ohjeiden mukaisuus. Käyttöönotto-ohje kuvaa näitä vaatimuksia kappaleessa Checklist (2) for commissioning chassis power [6, s. 19]. Kytkennät tarkistettiin laadittuja piirikaavioita vasten ja todettiin asianmukaisiksi.

STARTERissa on kaksi mahdollista tapaa luoda projekti: yhteydetön (offline) jossa kohdelaitteisto määritellään etukäteen ennen liittymistä siihen tai etsimällä yhteyteen kytkettynä (online) [6, s. 54-65]. Laitteiston etsiminen online oli perustellumpaa, kun samalla pystyttiin lukemaan muun ohella myös moottoritiedot. Tätä varten kytkettiin ohjelmointilaite (LPC) ja sähkökäyttöyksikkö toisiinsa sekä käynnistettiin avustaja *Project* alasvetovalikon *New with wizard...* toiminnolla. Prosessi on kuvattu jäljempänä kappaleessa: 4. 2 STARTER projektin luominen.

4.1 Siemens STARTER käyttöönotto työkaluohjelma

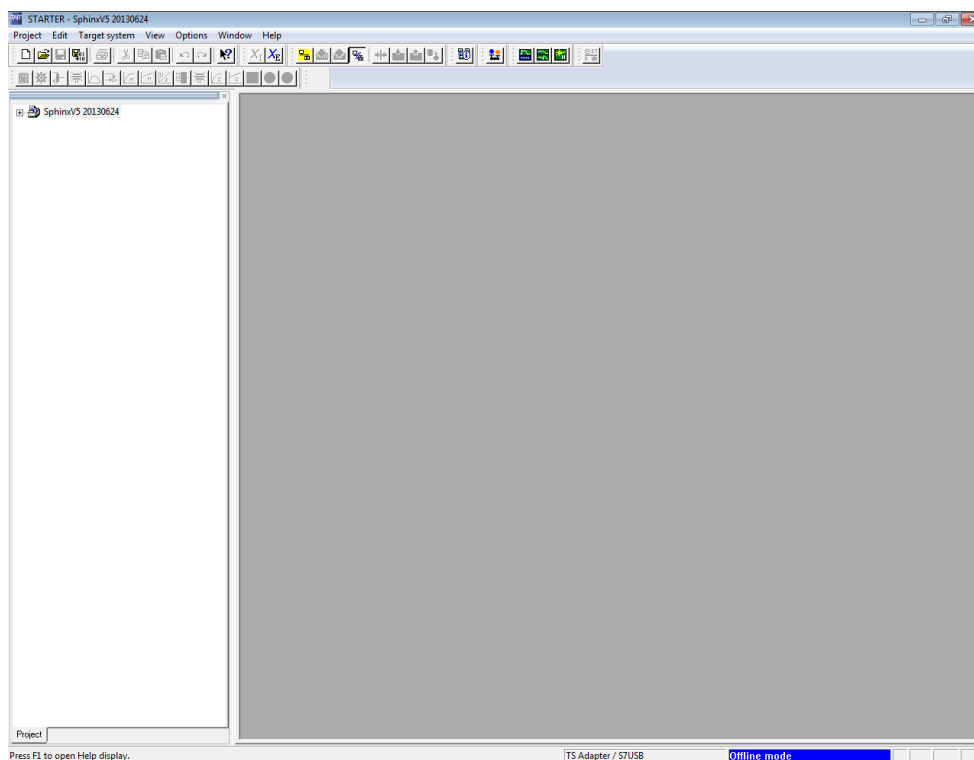
STARTER käyttöönotto työkaluohjelmaa voidaan käyttää MICROMASTER ja SINAMICS sarjojen sähkökäyttöjen ja järjestelmien konfigurointiin, käyttöönottoon, testaukseen, optimointiin ja diagnosointiin. Sähkökäytön konfiguraatio voidaan suorittaa ohjatusti niin sanotun avustajan (wizard) avustuksella. Kun ohjattu konfigurointi on suoritettu, voi käyttäjä muodostaa yhteyden sähkökäyttöön ja ladata asetukset siihen, sekä suorittaa erityisparametreja asianmukaisissa dialogeissa. Käyttäjä voi myös avata tilannekohtaisia opastuksia klikkaamalla *Help*-painiketta dialogi-ikkunoissa. PC:n toimintonäppäin F1 avaa aina sovelluksen koko opastuksen.

STARTER tukee seuraavia työkaluja projektin hoitamiseksi [6, s. 44 - 46]:

- Kopiointi työmuistista (RAM) pysyvämuistiin (ROM). Tällä varmistetaan, että data on saatavilla edelleen senkin jälkeen kun 24 V ohjausjännitesyöttö kytketään pois ja takaisin.
- Lataaminen kohdelaitteeseen. Data ladataan STARTERista ohjausyksikön työmuistiin.
- Lataaminen ohjelmointilaitteeseen (PG/PC tai LPC). Projekti ladataan ohjausyksiköstä STARTERIin.
- Tehdasasetusten palauttaminen. Kaikkien ohjausyksikön työmuistissa olevien parametrien asetus tehdasarvoihin.
- Käyttöönottoavustajan käyttäminen.
- Datavalikoiden luominen ja kopiointi.

Käyttäjiliityntää kutsutaan työpöydäksi (Kuvio 45). Työpöytä on jaettu kolmeen tärkeään toiminalliseen alueeseen:

1. Projektinavigaattoriin työpöydän vasemmalla laidalla
2. Työskentelyalueeseen työpöydän oikealla laidalla
3. Yksityiskohtaiseen näyttöön työpöydän alaosassa.

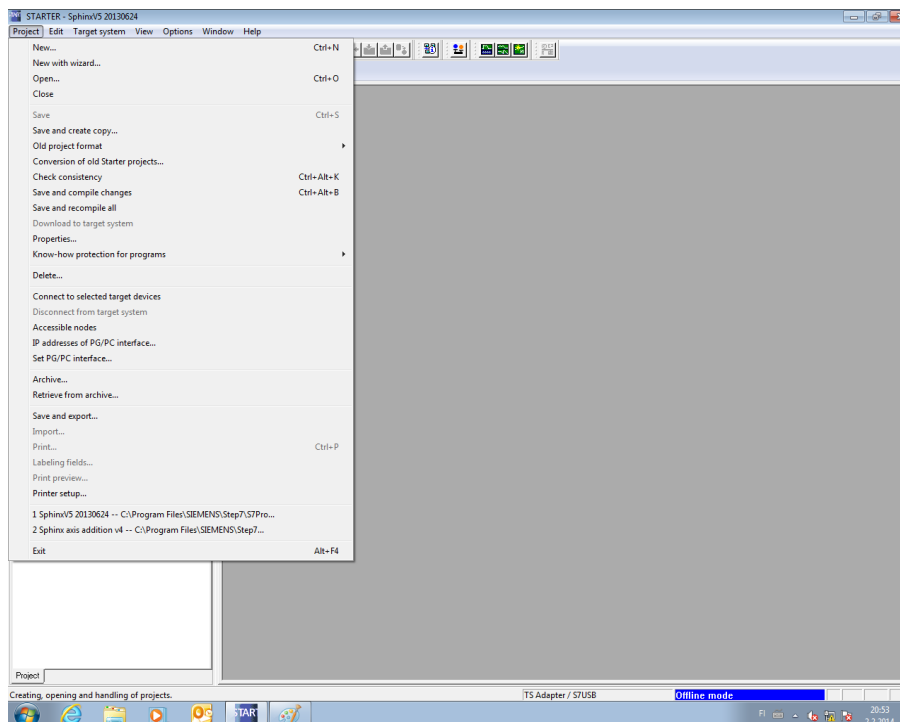


Kuvio 45. STARTER, työpöytä

STARTERin projektinavigaattorissa koko projekti on esitetty selkeällä, hierarkkisella puurakenteella. Samantyyppiset puuelementit on esitetty identtisillä ikoneilla ja jokaisella elementillä on luonteenomainen ikoninsa. Suoraan kytketyssä moodissa (online) ikonit ovat värikorostettuja osoittamaan, että kytkentä vastaavaan elementtiin kohdejärjestelmässä on luotu. Näkymä on samanlainen kuin Windows Explorerissa. Plus-merkki elementin edessä osoittaa, että se sisältää alielementtejä jotka voidaan avata napsauttamalla sitä.

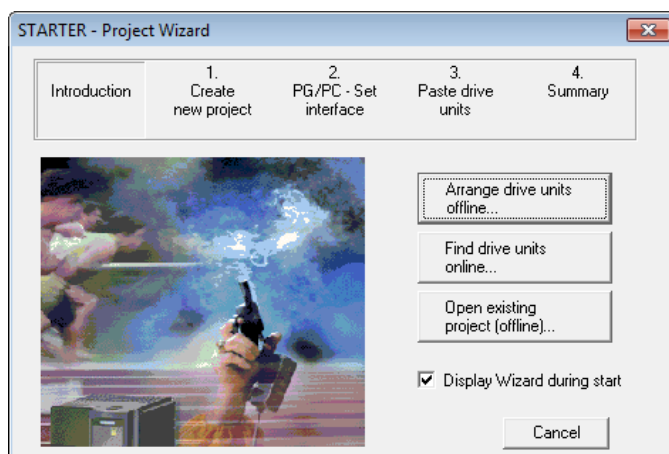
4.2 STARTER-projektin luominen

STARTER-projekti voidaan luoda lukemalla kohdelaitteiston kalustus online. Tätä varten kytkettiin PC:n ja kohdelaitteiston välille asianmukainen kaapeli ja käynnistettiin projektiavustaja eli wizard *Project* alavetovalikosta toiminnolla *New with wizard...* (Kuvio 46).



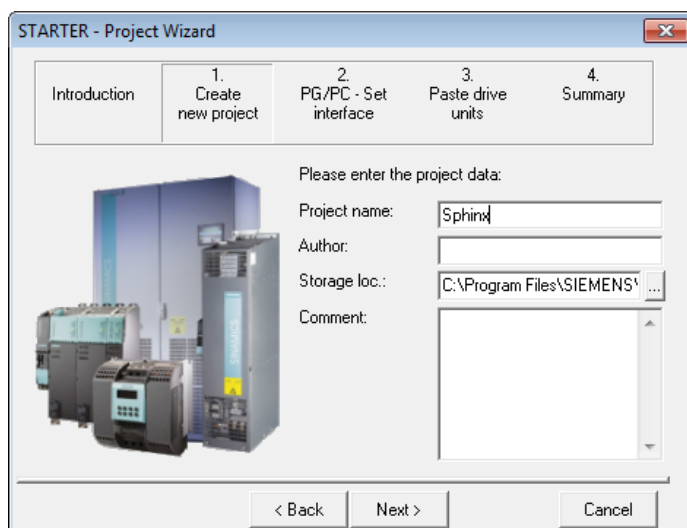
Kuvio 46. STARTER, *Project* alavetovalikko

Avustajan dialogin auettua toteutettiin kohdelaitteiston etsintä painamalla toimintonäppäintä *Find drive units online...* (Kuvio 47).



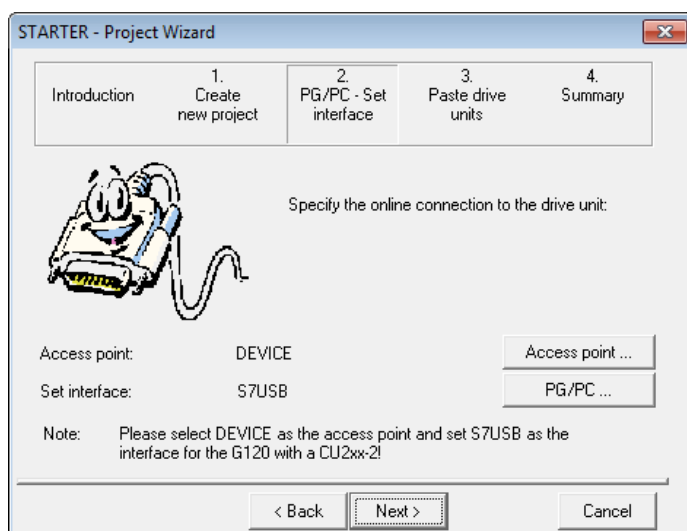
Kuvio 47. STARTER. Projektiavustaja, sähkökäyttöyksiköiden etsintä

Avustaja avasi *Create new project* dialogin johon syötettiin projektin nimi (Kuvio 48).



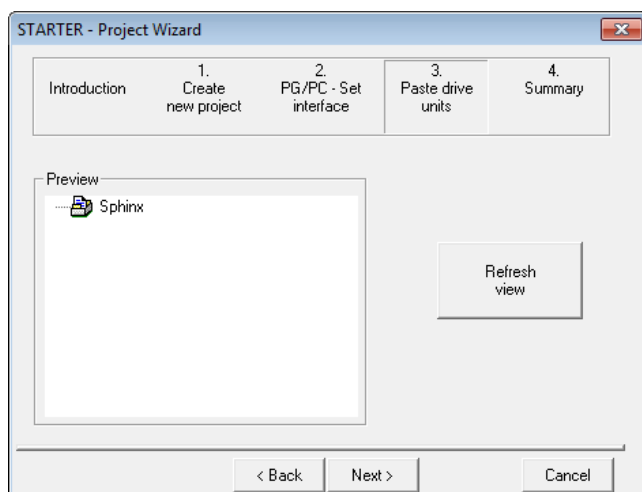
Kuvio 48. STARTER, Projektiavustaja, projektin nimi

Seuraavaksi avustaja avasi dialogin kommunikointiliittymän asetuksia varten (Kuvio 49).



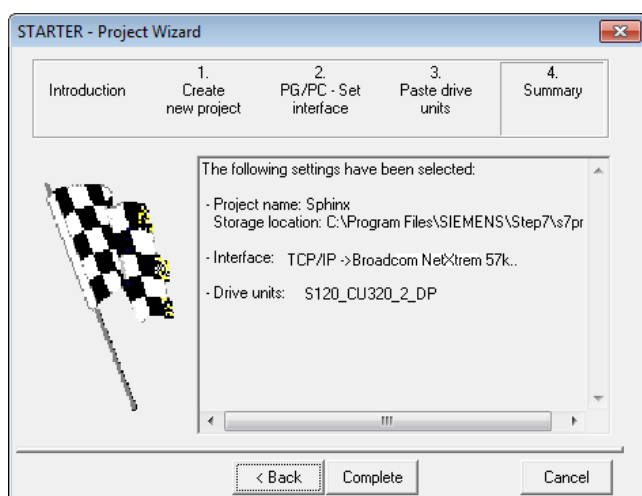
Kuvio 49. STARTER. Projektiavustaja, kommunikointiasetus

Avustajan prosessia jatkettiin painamalla toimintonäppäintä *Next*, jolloin seuraavassa vaiheessa projektiavustaja etsi kohdelaitteiston ja listasi löydettyt laitteet osioon *Preview* (Kuvio 50).



Kuvio 50. STARTER. Projektiavustaja, liitä sähkökäyttöyksiköt

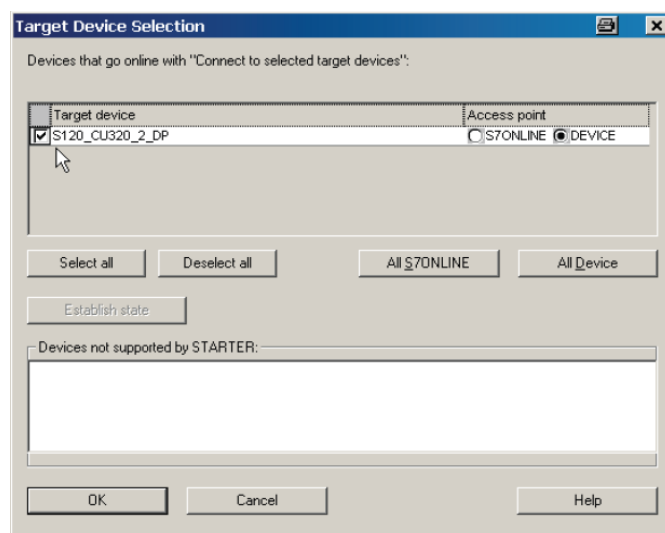
Viimeisessä dialogissa avustaja näytti yhteenvedon projektista (Kuvio 51) ja toimintonäppäimellä *Complete* avustaja suljettiin. [10, s. 26-28.]



Kuvio 51. STARTER. Projektiavustaja, yhteenvedo

4.2.1 Sähkökäyttöyksikön konfigurointi

Sähkökäyttöyksikön konfigurointia varten *Project* alasvetovalikosta valittiin toiminto *Connect to selected target devices* online moodiin kytkeytymiseksi. Kun kohdelaitteeseen kytkeydytään ensimmäisen kerran, avautuu dialogi *Target Device Selection* kohdelaitteen valitsemiseksi. Haluttu kohdelaitte valittiin, optio *DEVICE* aktivoitiin yhteyspisteeksi ja napautettiin *OK* näppäintä (Kuvio 52).



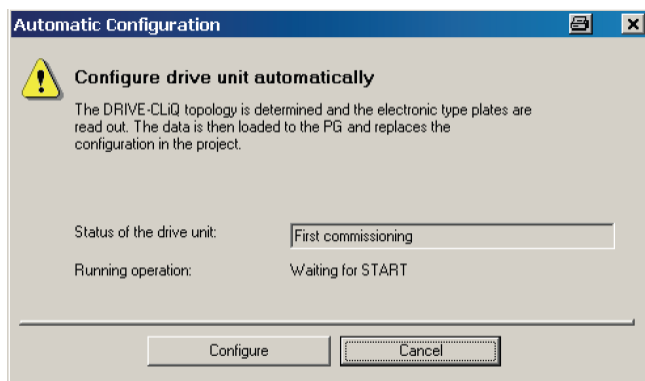
Kuvio 52. STARTER, kohdelaitte

Projektinavigaattoripuussa tuplanapautettiin *Automatic Configuration* optiota (Kuvio 53).



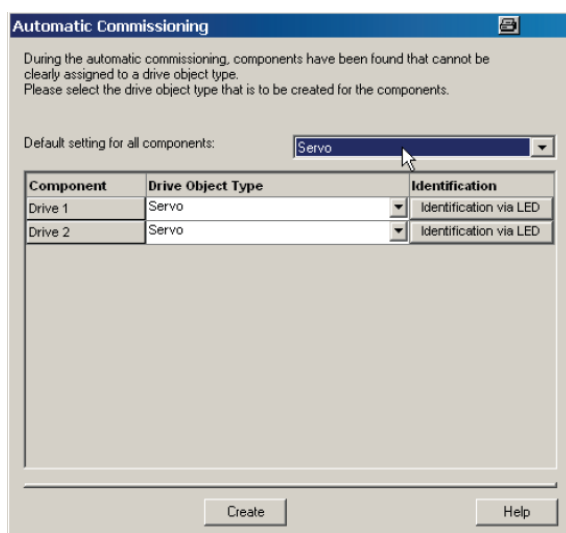
Kuvio 53. STARTER. Projektinavigaattoripuu, automaattinen konfigurointi

Configure toimintonäppäintä (Kuvio 54) napauttamalla käynnistettiin automaattinen konfigurointi, jolloin PC etsii DRIVE-CLiQ väylälle kytketyt objektit.



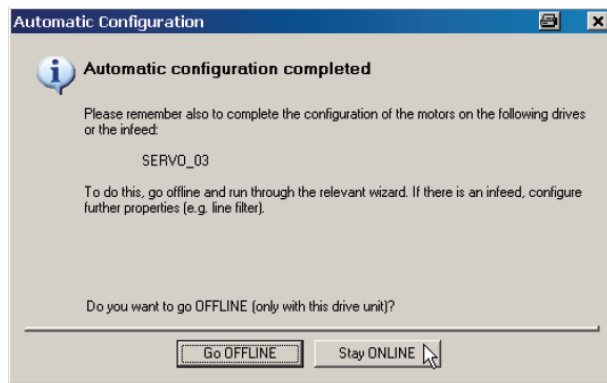
Kuvio 54. STARTER, sähkökäyttöyksikön automaattinen konfigurointi

Valitsemalla syöte *Servo* oletusasetukseksi kaikille komponenteille, asetettiin löydetyt sähkökäytöt servoiksi (Kuvio 55).



Kuvio 55. STARTER, löydettyjen sähkökäyttöjen tyypin asettaminen servoksi

Kun prosessi valmistui, avautui ilmoitus *Automatic configuration is complete* automaattisen konfiguroinnin valmistumisesta (Kuvio 56) [10, s. 29-31]. Ohjausyksikkö havaitsi servovahvistimet ja moottorit, siirsi laitetiedot ohjausyksikköön ja syötti automaattisesti oikeat tiedot parametreihin joita tarvittiin komponenttien käytössä.

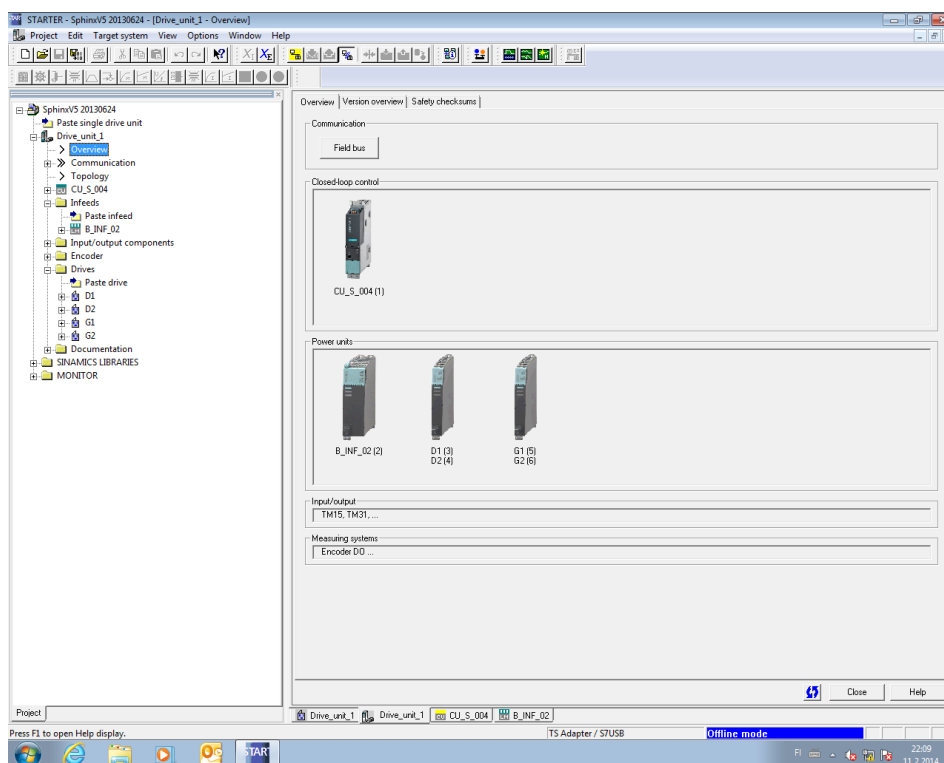


Kuvio 56. STARTER, automaattinen konfigurointi valmis

Seuraavassa on kuvattu joitakin keskeisimpiä granulikuivamen STARTER projektirakenteen näyttöjä.

Sähkökäyttöyksikkö, yleiskuvaus

Kuvassa 57 projektin sähkökäyttölaitteisto: suljetun silmukan servo-ohjain CU320-2DP (työkaluohjelmassa nimeytynyt CU_S004(1):ksi), tehoyksikkö välipiirin tasajännitettä varten B_INF_02(2), tuplamoottorimoduli purkajaluukun servomoottoreita D1 ja D2 varten sekä tuplamoottorimoduli sektoriluukkujen servomoottoreita G1 ja G2 varten.



Kuvio 57. STARTER, projektin sähkökäyttölaitteisto

Projektin rakenne projektinavigaattorissa:



Projekti. Koko projekti on talletettu tähän kansioon



Sähkökäyttöyksikkö. Sähkökäyttöyksikön konfigurointitieto on talletettu tänne. Projektin sähkökäyttöyksikkönä SINAMICS S120. Yhdessä yksikössä voi olla useita sähkökäyttöjä (esimerkiksi servovahvistimia, vektorikäyttöjä).



Kontrollimoduli (SINAMICS). Avoimen ja suljetun säätösilmukan toimintojen keskeinen komponentti. Kontrollimoduli kommunikoi järjestelmän muiden älykkäiden komponenttien kanssa.



Tehon sisään syöttö (SINAMICS). Generoi välipiirin DC jännitteen 3-vaihesyötöstä.



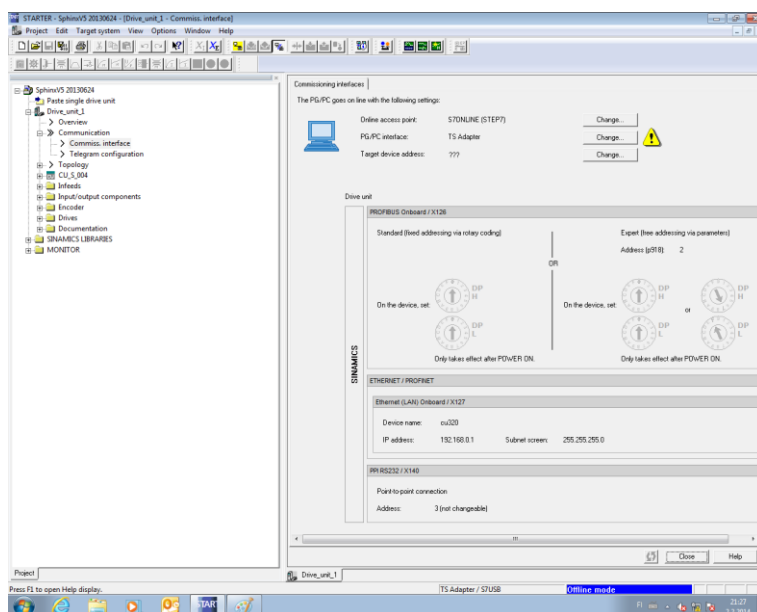
Sähkökäyttöteknologiaobjekti. Sähkökäytön konfiguraation ja parametrien määrittely tehdään täällä. Esimerkiksi servo tai suljetun silmukan vektorisäätö. Projektin servoakselit nimetty D1, D2 (purkajaluukun servo 1 ja 2), G1 ja G2 (sektoriluukkujen 1 ja 2 servot).



Liitä uusi objekti. Ikonin kaksoisnäpyttäminen liittää projektiin uuden objektin.

Sähkökäyttöyksikkö, käyttöönottooliittynän kommunikaatioasetus

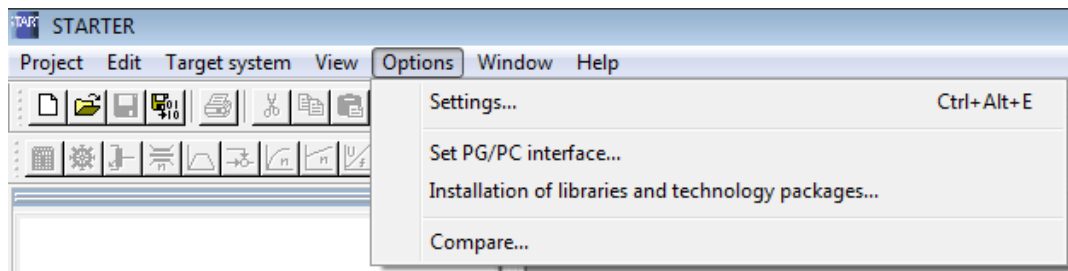
Käyttöönottooliittynän tiedonsiirtoasetukset tehdään STARTERin *Communication / Commiss. interface* näytössä käyttämällä *PG/PC interface.n Change...* toimintonäppäintä. Näytön alemmassa osassa on nähtävillä kohdelaitteessa tarjolla olevat kommunikointiliittynät (Kuvio 58).



Kuvio 58. STARTER, sähkökäyttöyksikkö käyttöönottooliittynän asetukset

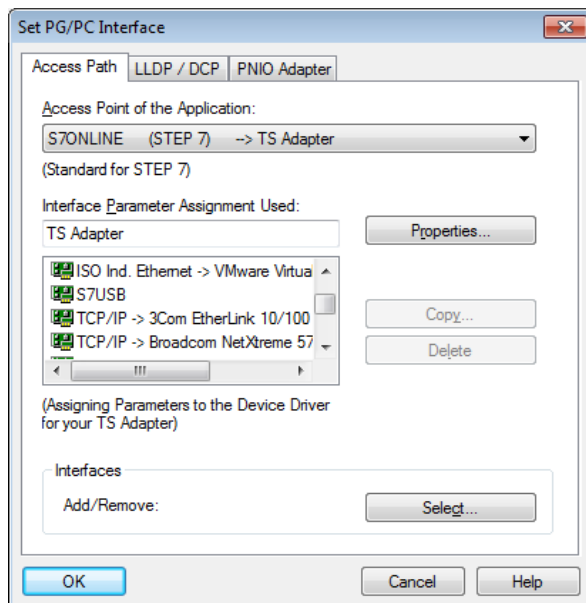
Vaihtoehtoisesti käyttöönottooliittynän tiedonsiirtoasetukset voidaan käynnistää *Options* alasvetovalikon *Set PG/PC interface...*-toiminnolla. Tätä toimintoa käytetään, kun

projekti halutaan luoda STARTERIin muodostamalla yhteys kohdelaitteistoon ja etsimällä sen komponentit (Kuvio 59).



Kuvio 59. STARTER, PC liitynnän asetus

Avautuvasta dialogista valitaan PC:n verkkoadapterin ohjain. Ellei listasta löydy PC:ssä olevaa ohjainta, on sellainen lisättävä listaan toimintonäppäimellä *Select...* (Kuvio 60).

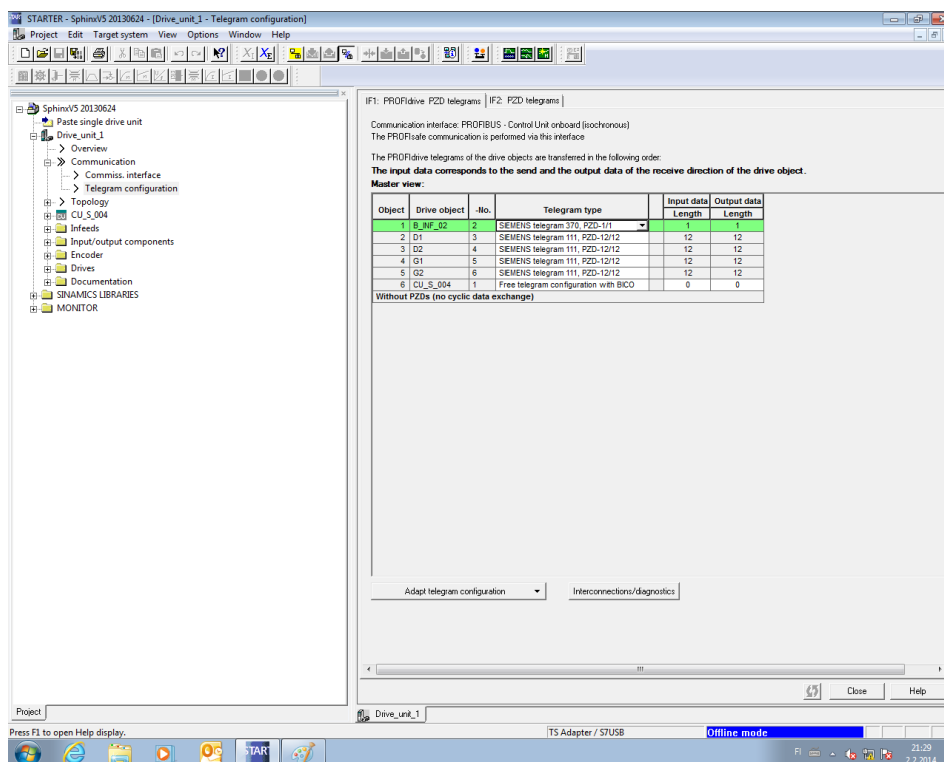


Kuvio 60. STARTER verkko-ohjain

Mikäli asetukset ovat väärät, ilmaantuu *Change...*-toimintonäppäimen perään keltainen varoituskolmio. Virheen mahdollinen syy saadaan esiin viemällä kursori kolmion päälle.

Sähkökäyttöyksikkö, kommunikointiliityntä

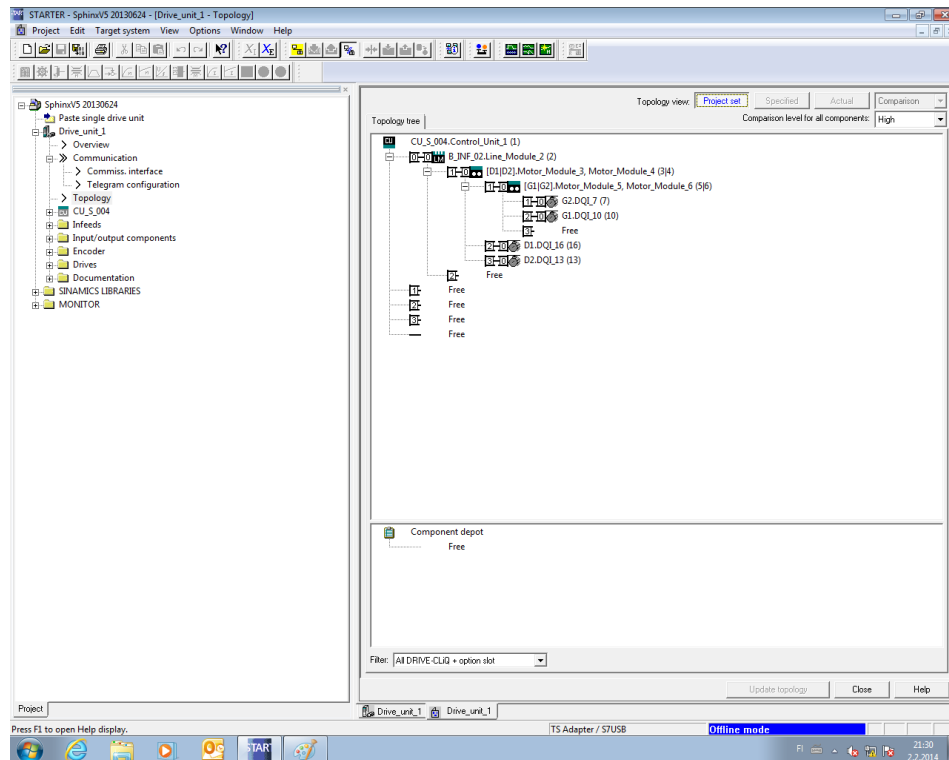
Ohjausyksikön PROFIBUS kommunikointiasetukset tehdään STARTERin *Communication / Telegram configuration* näytössä (Kuvio 61).



Kuvio 61. STARTER. Sähkökäyttöyksikkö, PROFIBUS kommunikointiliityntä

Sähkökäyttöyksikkö, kytkentätopologia

Kytkentätopologianäyttö (Kuvio 62) kuvaa SINAMICS sähkökäyttöjen DRIVE-CLiQ topologian puuna. Ohjausyksikkö CU320-2 DP on puun ensimmäinen elementti ja kaikki muut komponentit on liitetty siihen.



Kuvio 62. STARTER, sähkökäyttöyksikkö Drive-CLiQ topologia

Seuraavassa on lyhyt kuvaus projektin graafisista DRIVE-CLiQ ikoneista.



Ohjausyksikkö CU320-2DP.



Teholähde BASIC INFEED CONTROL.



Tuplamoottorimoduli. Kahden servomoottorin säätö.



DRIVE-CLiQ liitäntä. Numerot kuvaavat vastaavaa porttia.



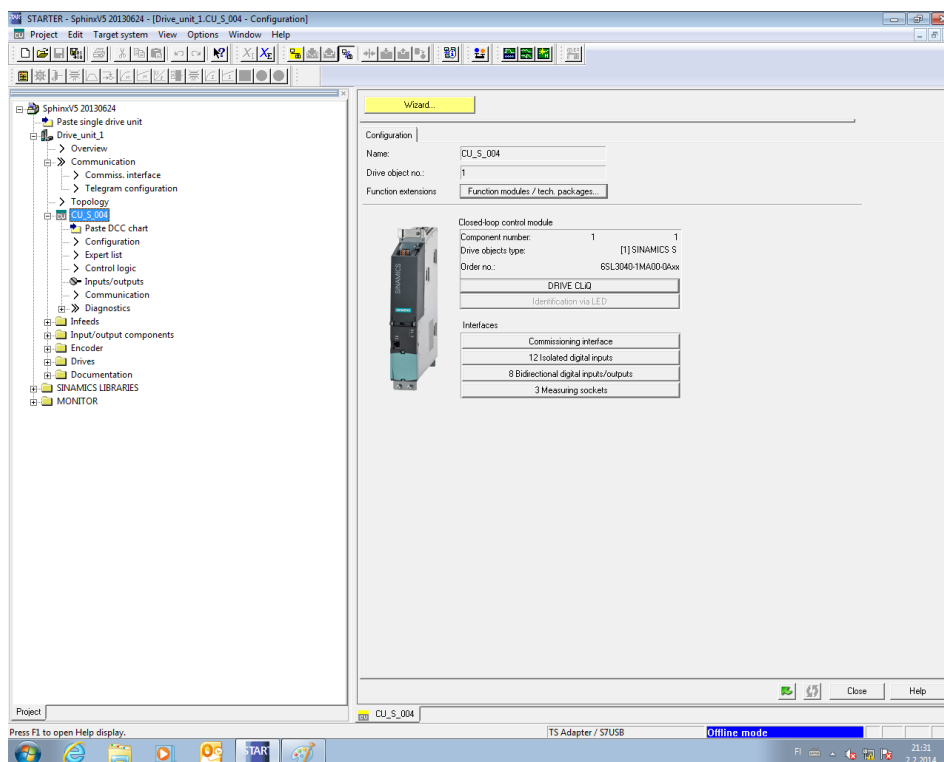
DRIVE-CLiQ moottorienkooderi.



Vapaa DRIVE-CLiQ liityntä.

Sähkökäyttöyksikkö, ohjausyksikkö

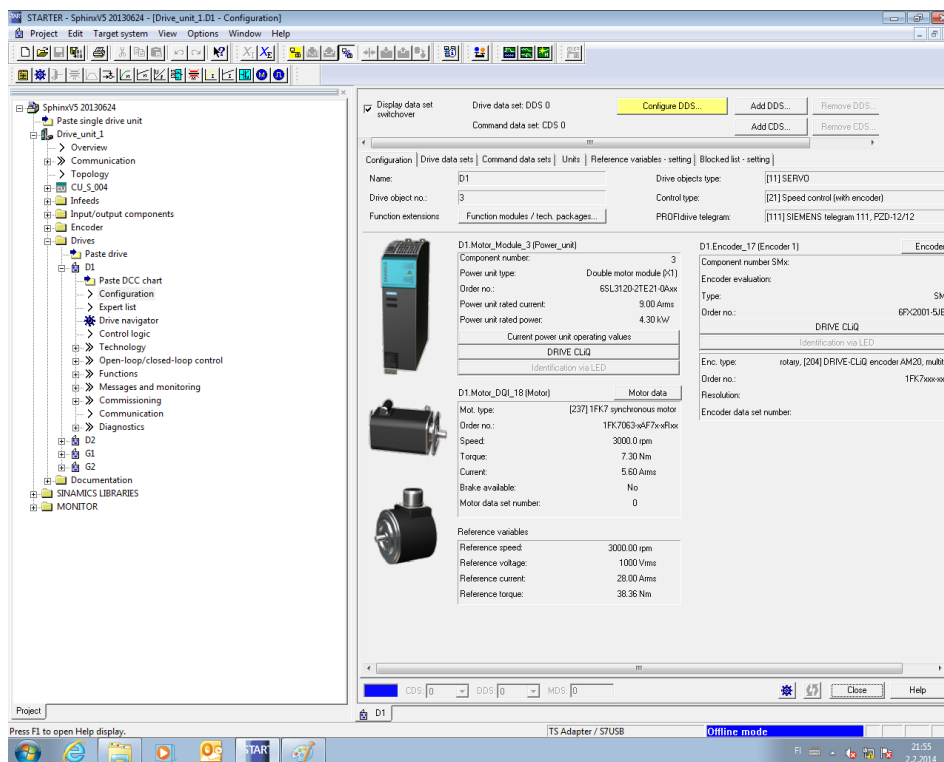
Ohjausyksikön konfiguraatio on suljetun silmukan ohjaus, joka tarkoittaa enkooderitakaisinkytkettyä servomoottoreiden ohjaustapaa (Kuvio 63).



Kuvio 63. STARTER, suljetun silmukan ohjausmoduli

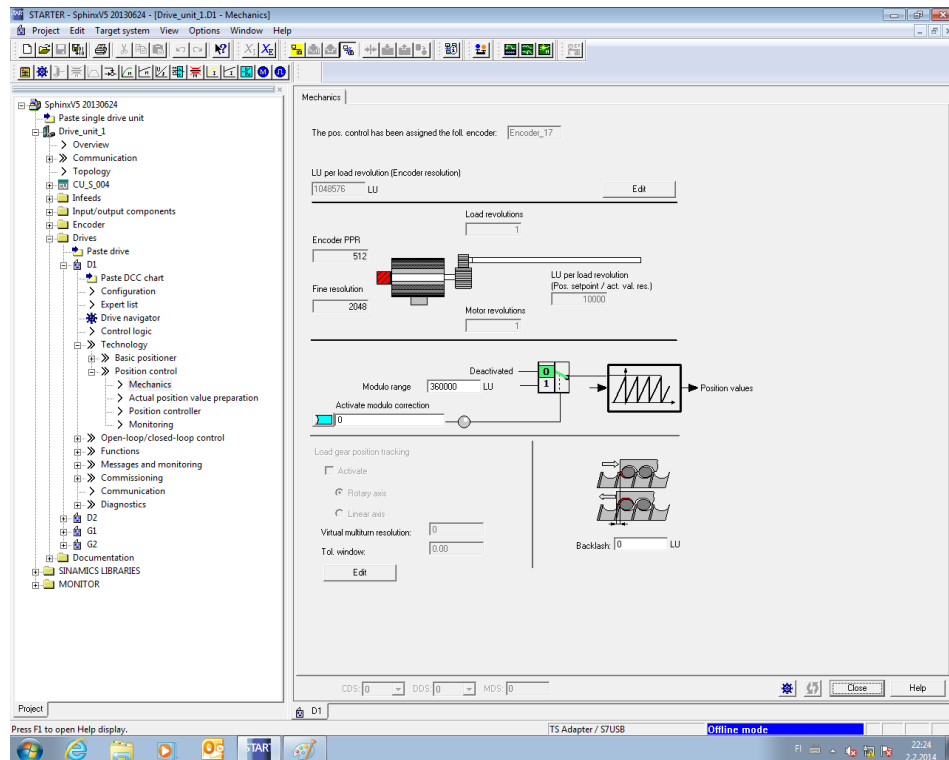
Sähkökäyttöyksikkö, servoakseliohjain

Servoakseliohjaimien asetukset ovat nähtävillä ja muutettavissa STARTERin *Drivers* alla olevassa akseliakohtaisessa konfiguraatiossa (*Drives / Akseli / Configuration*) (Kuvio 64). Kuviossa 64 on purkajaluukun servo 1:n konfiguraatio.



Kuvio 64. STARTER. Servovahvistin, konfiguraatio

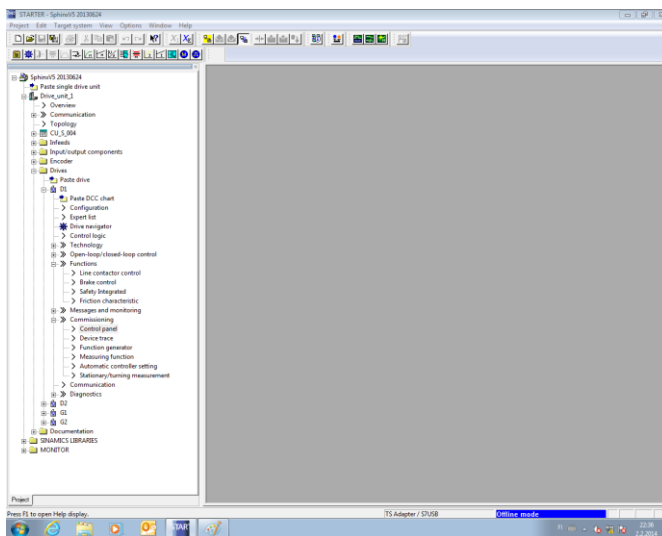
STARTERissa on mahdollista tehdä asetukset myös koskien mekanismia jota servoakseli liikuttaa. Tämä tapahtuu näytössä, joka löytyy kohdasta *Drives / haluttu akseli / Technology / Position control / Mechanics* (Kuvio 65). Syöttämällä asianmukaisesti kenttiin tiedot moottorin anturin resoluutiosta sekä mekanismin välityksistä, skaalautuu liikkeenohjaus haluttuun säädettävään mittayksikköön. Tätä ei kuitenkaan asiakkaan toiveesta tehty, vaan mahdolliset välityksen skaalaukset jätettiin sovellusohjelmalla tehtäväksi.



Kuvio 65. STARTER, mekaanisen välityksen asetus

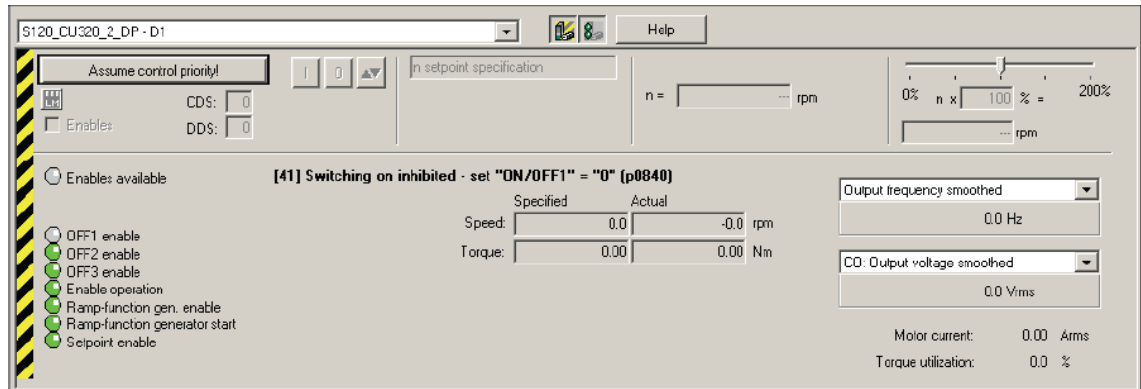
4.2.2 Sähkökäyttöjen käyttöönotto

Projektinavigaattoriin sähkökäyttöhakemiston alta löytyvästä *Commissioning* toiminnosta löytyvät työkalut servoakseleiden käyttöönottoon (Kuvio 66).



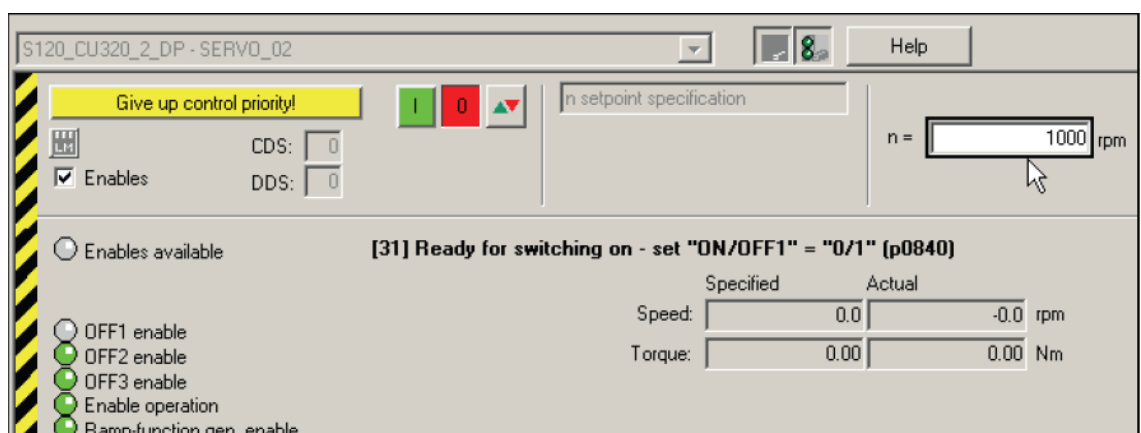
Kuvio 66. STARTER, käyttöönotto

Tärkein on *Control panel*. Sen avulla voi suorittaa käytön, monitoroinnin ja testauksen perustoiminnot. Toimintonäppäimellä *Assume control priority* käyttöönottaja ottaa ohjaketket omiin käsiinsä, perehdyttyään ensin toimintonäppäimen esiin tuomiin turvaohjeisiin ja hyväksyttyään ne (Kuvio 67).



Kuvio 67. STARTER. Käyttöönotto, ohjauspaneli

Asetuksella *Enables* aktivoituvat START (1, vihreä näppäin) ja STOP (0, punainen näppäin) toiminnot (Kuvio 68) [11]. START-näppäintä napauttamalla moottori kiihtyy syötettyyn pyörimisnopeuteen ja pysähtyy STOP-näppäimellä. Tässä kohden käyttöönottoa on syytä varovaisuuteen, jos servomoottorit ovat kiinni sellaisessa mekaniismissa, joka voi fyysisesti törmätä liikealueensa päällä. Jakopään mekanismit on kuitenkin suunniteltu niin, että liikkeet voivat pyöriä ympäri loputtomiin törmäämättä mihinkään.

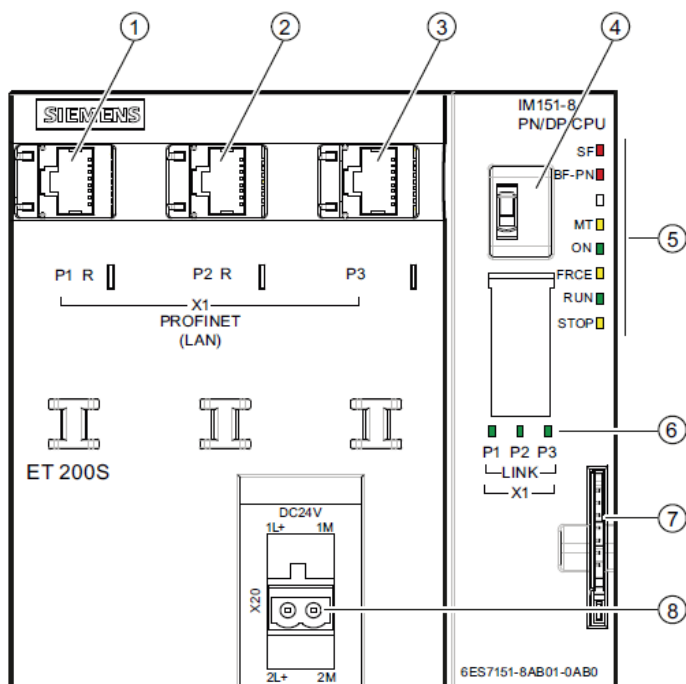


Kuvio 68. STARTER. Käyttöönotto, ohjauspaneli. Aktivoinnit

Kun *Control panel* in käytöstä halutaan luopua ja katkaista yhteys sähkökäyttöön, napautetaan toimintonäppäintä *Give up control priority!*.

4.3 Siemens Simatic ET200S logiikan käyttöönotto

Jotta ET200S:n IF 151-8 PN/DP CPU modulia voisi käyttää, täytyy SIMATIC mikromuistikortti olla asennettuna. Muistikortti asennetaan paikkaansa painamalla se oikein päin korttipaikkaansa johon se lukittuu (Kuvio 69). Kortissa oleva polarointikulmake varmistaa oikea-asentoisen asennuksen.



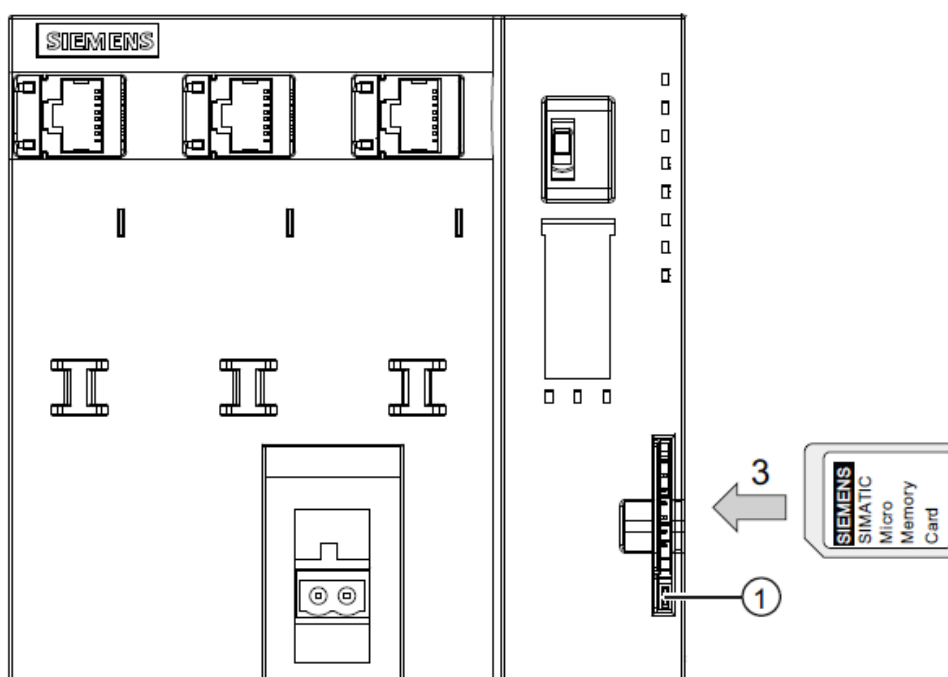
Kuvio 69. Simatic ET200S. IF 151-8 moduli

1. RJ45 pistoke. PROFINET liitynnän portti 1
2. RJ45 pistoke. PROFINET liitynnän portti 2
3. RJ45 pistoke. PROFINET liitynnän portti 3
4. Moodin valintakytkin
5. IM 151-8 PN/DP CPU liitäntämodulin tila- ja virhenäytöt
6. PROFINET liitännän tilanäytöt
7. SIMATIC Mikromuistikortin paikka
8. Syöttöjännitteen liitäntä.

Suosittelun käyttöönottoproseduuri on [12, s. 145]:

Asennetaan mikromuistikortti paikalleen, kytketään ohjelmointilaite (PG/PC tai LCP), kytketään jännitesyöttö päälle, jolloin ON LED syttyy. STOP LED vilkkuu 2 Hz:n taajuudella, kun IM 151-8 PN/DP CPU liitäntämoduli resetoituu muistikortin automaattisesti, loistaen tämän jälkeen kiinteästi. Ohjelman lataus suoritetaan SIMATIC Managerilla.

Muistikortin asentaminen (Kuvio 70) on käyttöönoton kannalta ensiarvoisen tärkeää. Jos sen jättää tekemättä, ei ET200S toimi lainkaan. Näin saattaa helposti käydä, koska muistikorttia ei toimiteta ET200S:n mukana, vaan se on tilattava erikseen.

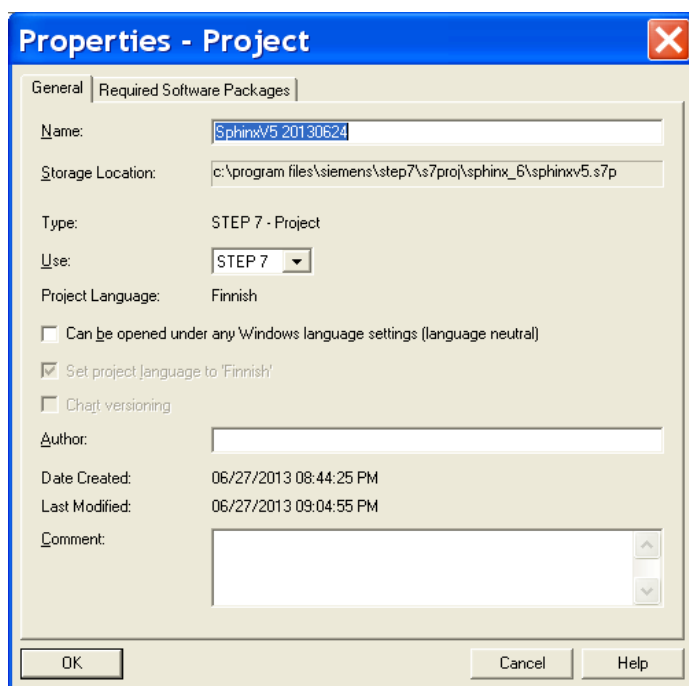


Kuvio 70. Muistikortin asentaminen

4.4 Sovellusohjelman käyttöönotto

Sovellusohjelman käyttöönottamiseksi se on ensin ladattava ohjelmankehityslaitteelta kohdelaitteeseen. Tätä ennen on kuitenkin varmistettava seuraavat:

1. Ohjelmointilaitteen ja logiikan välillä täytyy olla toimiva liityntä.
2. Lohkojen lataamiseksi logiikkaan on syöte *STEP 7* oltava objektin ominaisuuksien *Use* kohdassa (Kuvio 71).

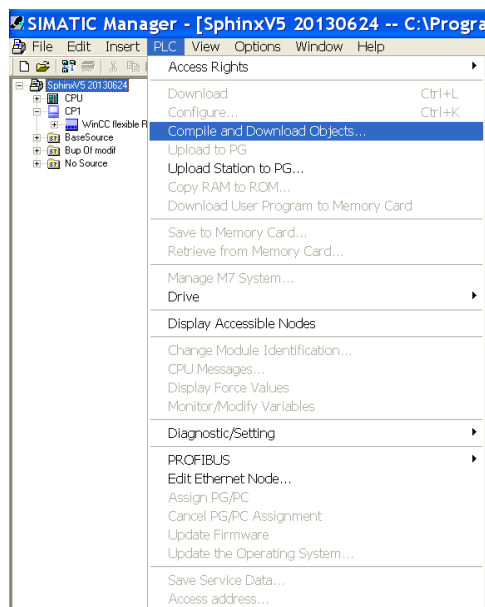


Kuvio 71. SIMATIC Manager, ladattavan objektin ominaisuudet

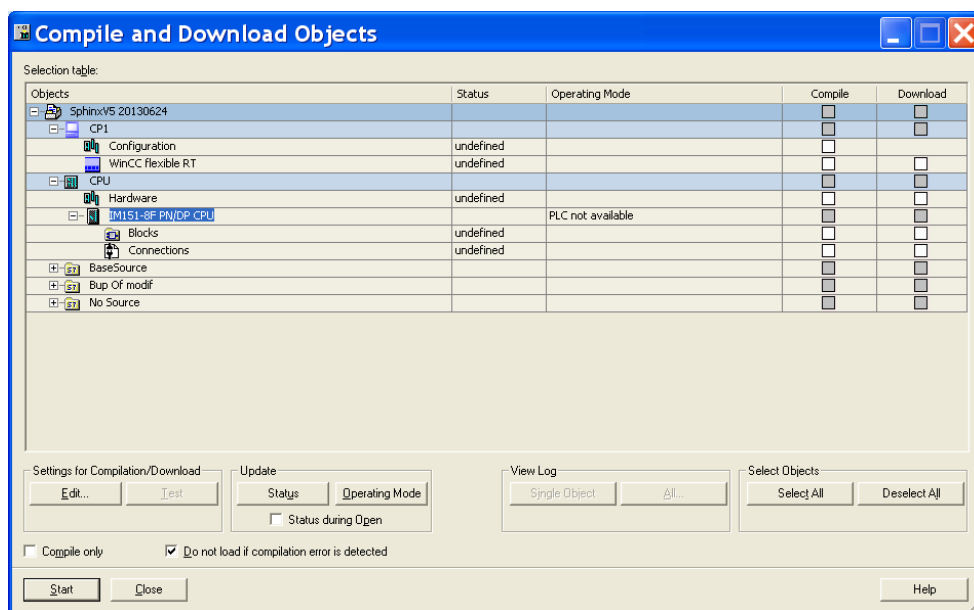
3. Kohdelogiikan CPU:n moodin valintakytkin on oltava asennossa, joka sallii latauksen (Kuvio 69, 4. Moodin valintakytkin). STOP tai RUN-P. STOP-asentoa suositellaan syystä, että RUN-P-asennossa ohjelma ladataan lohko kerrallaan. Mikäli ylikirjoitetaan vanhaa ohjelmaa voi syntyä konflikteja, jos lohkon parametrit ovat muuttuneet. STOP-moodissa suoritettu lataus ja niin sanottu warm restart prosessoivat ensin OB100 käynnistyslohkon. Jos käynnistyslohkon prosessointi onnistuu, vaihtaa CPU moodin RUN:iin ja sovellusohjelmien suoritus käynnistyy.

Lataus proseduuri on seuraavanlainen [13]:

1. Valitaan SIMATIC Managerissa objektit, jotka halutaan kääntää ja ladata.
2. Valitaan SIMATIC Managerin PLC alasvetovalikosta komento: *Compile and Download Objects...* (Kuvio 72), jolloin latausdialogi aukeaa (Kuvio 73).



Kuvio 72. SIMATIC Manager, projektin kääntäminen ja lataaminen



Kuvio 73. SIMATIC Manager, käännös- ja latausdialogi

3. Jos asetusta *Do not load if compilation error is detected* ei ole tehty, ladataan vain objektit joiden käännöksen aikana ei ole tapahtunut virhettä. Asetuksen ollessa päällä mitään ei ladata, jos jonkin objektin käännöksen aikana tapahtuu virhe.
4. *Compile* ja *Download* sarakkeissa valitaan objektit, jotka halutaan ladata.
5. Napautetaan *Start* toimintonäppäintä ja seurataan näytön ohjeita.

Sovellusohjelman käyttöönottamiseksi sen eri toiminnot testattiin yksitellen. Aluksi kokeiltiin akselikohtaisia käsiajoa (jog). Käyttöpäätteen päänäytöltä (Kuvio 17) valittiin toiminto Service ja avattiin käsiajosivu (Kuvio 19) painamalla toimintonäppäintä Manual pääsivun alareunassa. Kun kunkin akselin Jog toimintonäppäintä painettiin, tuli kyseessä olevan akselin liikkua niin kauan kuin näppäintä pidettiin painettuna tai saavutettiin liikealueen maksimi. Näppäimen päästämisen tuli pysäyttää liike. Mekanismin liikkumista seurattiin visuaalisesti kunnes oltiin varmoja siitä, että toiminto on halutunlainen. Samalla varmistettiin käyttöpäätteeltä, että paikan ja nopeuden oloarvot vastasivat asetettuja. Käsiajon jälkeen koekäytettiin vielä kohdepaikkaan ajo. Käyttöliittymän päänäytöltä valittiin toiminto Manual ja avattiin käsiajosivu. Käsiajosivun (Kuvio 19) kenttään *SETPOS* syötettiin kohdepaikan asetusarvo, jolloin mekanismi lähti liikkumaan sitä kohti. Ajon aikana varmistettiin käyttöpäätteeltä, että paikan ja nopeuden oloarvot vastasivat asetettuja ja että mekanismi pysähtyi hallitusti saavutettuaan kohdepaikan. Käsiajotestien lopuksi purkajan luukut ajettiin nollapaikkoihinsa.

Käsiajosten jälkeen testattiin automaattiajo. Tätä varten syötettiin ensin käyttöpäätteen Work Parameter sivulta (Kuvio 20) automaattiajon parametrit: sektoriluukkujen avauskulma, purkajan minimi ja maksimi liikekulmat, ajoaika ääriasennosta toiseen ja odotusaika kohdepaikassa. Sitten käynnistettiin automaattiajo painamalla toimintonäppäintä *Automatic*. Mekanismi toteutti työkiertoa ja käyttöönoton testit todettiin suoritetuiksi. Lopuksi vielä otettiin projektista kopiot ja toimitettiin ne työn tilaajalle ja muille asianosaisille.

5 Yhteenveto

Mielestäni saavutin asetetut tavoitteet hyvin, vaikkakin työn laajuudeksi asetettiin jotakuinkin kolmen lopputyön määrä. Valmiista dokumenttipohjista suunnittelutyöhön ryhtyminen uuden työkalun ominaisuuksien omaksumiseksi oli paljon nopeampaa kuin niin sanotusti tyhjältä pöydältä. Laitteistosuunnittelu valmistui suunnitellussa ajassa, kaappivalmistaja rakensi ohjauskaapit tehtyjen dokumenttien pohjalta ilman kohtalokkaita puutteita osaluettelossa ja toimitti ne konepajalle niin, että käyttöönotto voitiin aloittaa suunnitellusti.

Ohjelmoinnin aloittaminen käyttöpäätteestä antoi sopivan alkusysäyksen asiaan pääsemiseksi ja mielenkiinnon kohottamiseksi vaikeampiin ohjelmointeihin ryhtymiseksi. Syntyi prosessi, joka itse loi jatkuvasti vastauksia synnyttämiinsä kysymyksiin. Samalla testattiin kuinka idea valmiin ohjelmapohjan soveltamisesta projektin ohjelmointitarpeiden täyttämiseksi onnistuu käytännössä. Varsinkin tämä kokemus lienee hyödyllistä tietoa Outotecille, kun menettelyä lanseerataan laajempaan käyttöön ja mietitään miten dokumentoida ohjelmapohjaa riittävästi, jotta sen tapa lähestyä automaation ohjelmointeja olisi käyttäjälleen helpommin ja nopeammin omaksuttavissa. Nyt puuttuva tieto siirtyi pääasiassa kahdenvälisissä keskusteluissa, mitä järjestelmätoimintatavan kannalta ei voida pitää hyväksyttävänä. Kiitoksensa ansaitsevat työtoverit, jotka ystävällisesti pulmatilanteen ilmaantuessa tarjosivat avuksi asiantuntemustaan. Kokemukseni perusteella toteaisin kuitenkin, että valmiin ohjelmapohjan idea on kannatettava ja toimiva verrattuna tapaan, jossa jokainen ohjelmoija toteuttaisi oman näkemyksensä.

Käyttöönotto konepajalla käynnistyi kesäisessä 34 °C:n helteessä edeten kuitenkin suunnitellusti laitteiston käyttöönotosta itse sovelluksen testaukseen ja havaintoihin siitä, että laitteisto toteuttaa spesifikaation mukaisia toimintoja.

Oman haasteensa raportoinnille asetti työn laajuus ja siitä johtuva rajausten tarpeellisuus. Kuinka esittää asia sortumatta yksityiskohtien loputtomaan suohon? Vastavalmistunut insinööri sijoittuu työelämäänsä aluksi joihinkin näistä kolmesta aihepiiristä ja kokemuksen karttuessa laajentaneen työnsä kuvaa tässä esitetyn kaltaiseksi. Mielestäni tämä raportti kuvaa riittävällä tarkkuudella ja sujuvasti millaista tuollaisen työn sisältö voisi olla.

Hyötyihin luettakoon myös tämän raportin käyttökelpoisuus tulevassa käyttöönotossa kaiken dokumentaation kokoavana. Ohjelmoidun toiminnallisuuden ymmärtämisen kannalta hyödyllisimpänä pidän kappaletta: 3.7.11 Signaalin kulku operaattorilta toimilaitteelle. Oma hyötynsä lienee myös raportoinnin tuottamalla asian tuoreena mielessä pitämisellä, käyttöönottoa loppuasiakkaan luona odotellessa.

Suurimpana kantona kaskessa pidän Siemensin harjoitus- tai testauslaitteiston puutetta itse ohjelmointityön aikana. Nyt ohjelmoinnit jouduttiin mielestäni tekemään liian pitkälle pimennossa ilman varmuutta halutusta toiminnallisuudesta. Projektin hoidon ja aikataulun kannalta parasta olisi, että ohjelmointien toimivuus voitaisiin varmistaa sitä mukaa kun niitä tehdään ja jo ennen konepajatestejä. Miinus tästä lankeaa mielestäni Siemensille, jonka palvelurepertuaarista on hankalahkoa edes vuokrata sopivaa laitteistoa tällaiseen tarkoitukseen. Kilpailijoilta sellainen järjestyy kustannuksitta ja ilman saksalaista byrokratiaa. Oma lukunsa on myös Siemensin täydellistäkin täydellisempi dokumentaatio, joka pelkällä lukumäärällään aiheuttaa ikään kuin tiedon piiloutumista itseensä; puuta ei näe metsältä. Ihmiset joita kuuntelin Siemensin nettisivuilta dokumenttien etsimisen käyttökokemuksesta, implikoivat saaneensa vain epävarmuuden siitä -onko tämä juuri SE dokumentti jota etsin? Tämän opinnäytteen Siemens lähteiden yhteenlaskettu sivumäärä on 3948. Lisäksi on koko joukko muita Siemensin dokumentteja joihin ei tässä työssä ole viitattu, mutta jotka ovat olleet tarpeellisia asian omaksumiseksi.

Lähteet

- 1 Outotec. Yrityksen verkkosivut. <<http://www.outotec.com/fi/>>. Luettu 3.5.2014.
- 2 Outotec Turula. Konepajan verkkosivut. <<http://www.outotec.com/fi/Tuotteet-ja-palvelut/Services/Turulan-konepajan-laitevalmistus/>>. Luettu 3.5.2014.
- 3 Tampereen Keskustekniikka Oy. Yrityksen verkkosivut. <<http://www.keskustekniikka.fi/>>. Päivitetty 7.4.2014. Luettu 3.5.2014.
- 4 Site conditions Sphinx. Outotec. Doc 159/000-DET-M3-002. June 7, 2012
- 5 SIEMENS SINAMICS S120 Booksize Power Units manual, 01/2013
- 6 SIEMENS SINAMICS S120. Commissioning manual 10/2008
- 7 Rittal Oy. Yrityksen verkkosivut. <<http://www.rittal.com/fi-fi/content/fi/start/>>. Luettu 3.5.2014.
- 8 SIEMENS SIMATIC Manager, HTML-help, Blocks in the user program
- 9 SIEMENS SINAMICS S120/S150, List Manual 01/2012
- 10 SIEMENS SINAMICS S120. Getting Started 01/2013
- 11 SIEMENS SINAMICS STARTER, HTML-help, Control Field – Control Panel – Drive
- 12 SIEMENS SIMATIC ET 200S distributed I/O IM 151 PN/DP CPU interface module
- 13 SIEMENS SIMATIC Manager, HTML-help, Compiling and downloading Objects

Simatic Step 7 ohjelmalistaus, purkajaluukun servo 1:n DB

1 (5)

SphinxV5 20130624\CPU\IM151-8F PN/DP CPU\...\DB Distributor1

```

1  (* Copyright (C) Outokumpu technology, Espoo Finland
2  =====
3  NOTES :
4  Revision History :
5  =====*)
6
7  DATA_BLOCK "Distributor1DB"
8  TITLE = 'Distributor 1 DB'
9  VERSION : 1.1
10 AUTHOR : RKy
11
12 STRUCT
13
14 nModeCmd: // Main mode commands
15 STRUCT
16 bDummy08: BOOL; // Bit08:
17 bDummy09: BOOL; // Bit09:
18 bDummy10: BOOL; // Bit10:
19 bDummy11: BOOL; // Bit11:
20 bDummy12: BOOL; // Bit12:
21 bDummy13: BOOL; // Bit13:
22 bDummy14: BOOL; // Bit14:
23 bDummy15: BOOL; // Bit15:
24 bEStop:   BOOL; // Bit00: EMERGENCY STOP
25 bStop:    BOOL; // Bit01: STOP
26 bService: BOOL; // Bit02: SERVICE
27 bAux:     BOOL; // Bit03: AUX
28 bManu:    BOOL; // Bit04: MANUAL
29 bSemi:    BOOL; // Bit05: SEMI-AUTOMATIC
30 bAuto:    BOOL; // Bit06: AUTOMATIC
31 bDummy07: BOOL; // Bit07:
32 END_STRUCT;
33
34 nModeSt: // Main mode status
35 STRUCT
36 bMoveOK:  BOOL; // Bit08: Movement allowed
37 bDummy09: BOOL; // Bit09:
38 bDummy10: BOOL; // Bit10:
39 bDummy11: BOOL; // Bit11:
40 bDummy12: BOOL; // Bit12:
41 bDummy13: BOOL; // Bit13:
42 bDummy14: BOOL; // Bit14:
43 bDummy15: BOOL; // Bit15:
44 bEStop:   BOOL; // Bit00: EMERGENCY STOP
45 bStopped: BOOL; // Bit01: STOPPED
46 bService: BOOL; // Bit02: SERVICE
47 bAux:     BOOL; // Bit03: AUX
48 bManu:    BOOL; // Bit04: MANUAL
49 bSemi:    BOOL; // Bit05: SEMI-AUTOMATIC
50 bAuto:    BOOL; // Bit06: AUTOMATIC
51 bDummy07: BOOL; // Bit07:
52 END_STRUCT;
53
54 nModeStForHMI: INT; // Main mode for HMI
55
56 nMachSt: // Machine status
57 STRUCT

```

Simatic Step 7 ohjelmalistaus, purkajaluukun servo 1:n DB

2 (5)

```

58 bAlmAckCmd:  BOOL; // Alarm acknowledgement command
59 bFltAckCmd:  BOOL; // Fault acknowledgement command
60 bDummy10:    BOOL;
61 bDummy11:    BOOL;
62 bDummy12:    BOOL;
63 bDummy13:    BOOL;
64 bDummy14:    BOOL;
65 bDummy15:    BOOL;
66 bReady:      BOOL; // Ready
67 bRun:        BOOL; // Run
68 bAlarm:      BOOL; // Alarm
69 bAlarmAck:   BOOL; // Unacknowledged alarm
70 bFault:      BOOL; // Fault
71 bFaultAck:   BOOL; // Unacknowledged fault
72 bDisable:    BOOL; // Disable
73 bDummy05:    BOOL;
74 bDummy06:    BOOL;
75 bDummy07:    BOOL;
76 END_STRUCT;
77
78 nMachStForHMI: INT; // Machine status for HMI
79
80 nAW1: // Alarm word 1
81 STRUCT
82 bDummy08:    BOOL; // Bit08:
83 bDummy09:    BOOL; // Bit09:
84 bDummy10:    BOOL; // Bit10:
85 bDummy11:    BOOL; // Bit11:
86 bDummy12:    BOOL; // Bit12:
87 bDummy13:    BOOL; // Bit13:
88 bDummy14:    BOOL; // Bit14:
89 bDummy15:    BOOL; // Bit15:
90 bPosDevWith2TooHigh: BOOL; // Bit00: Pos deviation servo 2 too high
91 bDummy01:    BOOL; // Bit01:
92 bDummy02:    BOOL; // Bit02:
93 bDummy03:    BOOL; // Bit03:
94 bDummy04:    BOOL; // Bit04:
95 bDummy05:    BOOL; // Bit05:
96 bDummy06:    BOOL; // Bit06:
97 bDummy07:    BOOL; // Bit07:
98 END_STRUCT;
99
100 nAW1Ack: // Unacknowledged Alarm word 1
101 STRUCT
102 bDummy08:    BOOL; // Bit08:
103 bDummy09:    BOOL; // Bit09:
104 bDummy10:    BOOL; // Bit10:
105 bDummy11:    BOOL; // Bit11:
106 bDummy12:    BOOL; // Bit12:
107 bDummy13:    BOOL; // Bit03:
108 bDummy14:    BOOL; // Bit14:
109 bDummy15:    BOOL; // Bit15:
110 bDummy00:    BOOL; // Bit00:
111 bDummy01:    BOOL; // Bit01:
112 bDummy02:    BOOL; // Bit02:
113 bDummy03:    BOOL;
114 bDummy04:    BOOL; // Bit04:
115 bDummy05:    BOOL; // Bit05:
116 bDummy06:    BOOL; // Bit06:

```

Simatic Step 7 ohjelmalistaus, purkajaluukun servo 1:n DB

3 (5)

```

117 bDummy07: BOOL; // Bit07:
118 END_STRUCT;
119
120 nAWDummy: ARRAY[1..6] OF WORD; // Alarm word dummy
121 nExtendAW: ARRAY[1..4] OF WORD; // Extended Alarm for HMI
122
123 nFW1: // Fault word 1
124 STRUCT
125 bDummy08:      BOOL; // Bit08:
126 bDummy09:      BOOL; // Bit09:
127 bDummy10:      BOOL; // Bit10:
128 bDummy11:      BOOL; // Bit11:
129 bDummy12:      BOOL; // Bit12:
130 bDummy13:      BOOL; // Bit13:
131 bDummy14:      BOOL; // Bit14:
132 bDummy15:      BOOL; // Bit15:
133 bServodrive:    BOOL; // Bit00: Servodrive
134 bHomeNotOk:     BOOL; // Bit01: Home not Ok
135 bFollowingError: BOOL; // Bit02: Following error
136 bDummy03:      BOOL; // Bit03:
137 bDummy04:      BOOL; // Bit04:
138 bDummy05:      BOOL; // Bit05:
139 bDummy06:      BOOL; // Bit06:
140 bDummy07:      BOOL; // Bit07:
141 END_STRUCT;
142
143 nFW1Ack: // Unacknowledged Fault word 1
144 STRUCT
145 bDummy08: BOOL; // Bit08:
146 bDummy09: BOOL; // Bit09:
147 bDummy10: BOOL; // Bit10:
148 bDummy11: BOOL; // Bit11:
149 bDummy12: BOOL; // Bit12:
150 bDummy13: BOOL; // Bit13:
151 bDummy14: BOOL; // Bit14:
152 bDummy15: BOOL; // Bit15:
153 bDummy00: BOOL; // Bit00:
154 bDummy01: BOOL; // Bit01:
155 bDummy02: BOOL; // Bit02:
156 bDummy03: BOOL; // Bit03:
157 bDummy04: BOOL; // Bit04:
158 bDummy05: BOOL; // Bit05:
159 bDummy06: BOOL; // Bit06:
160 bDummy07: BOOL; // Bit07:
161 END_STRUCT;
162
163 nFWDummy: ARRAY[1..6] OF WORD; // Fault word dummy
164 nExtendFW: ARRAY[1..4] OF WORD; // Extended Fault for HMI
165
166 nEW1: // Event word 1
167 STRUCT
168 bDummy08: BOOL; // Bit08:
169 bDummy09: BOOL; // Bit09:
170 bDummy10: BOOL; // Bit10:
171 bDummy11: BOOL; // Bit11:
172 bDummy12: BOOL; // Bit12:
173 bDummy13: BOOL; // Bit13:
174 bDummy14: BOOL; // Bit14:
175 bDummy15: BOOL; // Bit15:

```

Simatic Step 7 ohjelmalistaus, purkajaluukun servo 1:n DB

4 (5)

```

176 bDummy00: BOOL; // Bit00:
177 bDummy01: BOOL; // Bit01:
178 bDummy02: BOOL; // Bit02:
179 bDummy03: BOOL; // Bit03:
180 bDummy04: BOOL; // Bit04:
181 bDummy05: BOOL; // Bit05:
182 bDummy06: BOOL; // Bit06:
183 bDummy07: BOOL; // Bit07:
184 END_STRUCT;
185
186 nEWDummy: ARRAY[1..6] OF WORD; // Event word dummy
187 nExtendEW: ARRAY[1..4] OF WORD; // Extended Event for HMI
188
189 nIW1: // I-Lock word 1
190 STRUCT
191 bDummy08: BOOL; // Bit08:
192 bDummy09: BOOL; // Bit09:
193 bDummy10: BOOL; // Bit10:
194 bDummy11: BOOL; // Bit11:
195 bDummy12: BOOL; // Bit12:
196 bDummy13: BOOL; // Bit13:
197 bDummy14: BOOL; // Bit14:
198 bDummy15: BOOL; // Bit15:
199 bDummy00: BOOL; // Bit00:
200 bDummy01: BOOL; // Bit01:
201 bDummy02: BOOL; // Bit02:
202 bDummy03: BOOL; // Bit03:
203 bDummy04: BOOL; // Bit04:
204 bDummy05: BOOL; // Bit05:
205 bDummy06: BOOL; // Bit06:
206 bDummy07: BOOL; // Bit07:
207 END_STRUCT;
208
209 nIWDummy: ARRAY[1..6] OF WORD; // I-Lock word dummy
210 nExtendIW: ARRAY[1..4] OF WORD; // Extended I-Lock for HMI
211
212 bHoming: BOOL; // Distributor 1 Homing start bit
213 nNegatSwingPar: INT; // Distributor swing position setpoint min
214 nPositSwingPar: INT; // Distributor swing position setpoint max
215 bDrvToSetPntCmd: BOOL; // Distributor 1 to setpoint command
216 bMechConnOpenCmd: BOOL; // Distributor mechanical connection open
218 nMoveFactorPar: REAL; // Adjustment factor
219 nWaitDelayPar: TIME; // Wait time at the end of movement [s]
221 nJogSpdNs: INT; // Distributor servo 1 jog forward speed
222 nHomeSpdNs: INT; // Distributor servo 1 homing speed.
223 nWorkSpdNs: INT; // Distributor servo 1 working speed.
224 nPosDevMarginPar: REAL; // Distributor pos deviation servo 1 and 2
228
229 //IN
230 bOnCmd: BOOL; // Drive On / Off command
231 bPosCmd: BOOL; // Positioning On /Off
232 bFStopCmd: BOOL; // Drive Fast stop (0 = FStop active)
233 bFwdCmd: BOOL; // Jog forward command
234 bFwdSt: BOOL; // Jog forward status
235 bBwdCmd: BOOL; // Jog backward command
236 bBwdSt: BOOL; // Jog backward status
237 nLADDR: INT; // Telegram first address from HW configuration
238 nPosNs: DINT; // Position setpoint
239 nVelNs: DINT; // Velocity setpoint

```


Simatic Step 7 ohjelmalistaus, purkajaluukun servo 1:n DB

5 (5)

```

240 fAccLimNs:    REAL; // Acceleration override (%)
241 fDecLimNs:    REAL; // Deceleration override (%)
242 fVelLimNs:    REAL; // Velocity override (%)
243
244 //OUT
245 bRdyOn:        BOOL; // Drive ready for switch on (Enable)
246 bRdyRef:       BOOL; // Drive ready for reference (Running)
247 bToPos:        BOOL; // Positionning running
248 bOnPos:        BOOL; // Target position reached
249 bAlmSt:        BOOL; // Drive alarm status
250 bFltSt:        BOOL; // Drive fault status
251 nAlm:          INT;  // Alarm code
252 nFlt:          INT;  // Fault code
253 nPosMe:        DINT; // Position measurement
254 nVelMe:        DINT; // Velocity measurement
255
256 //Limits IN
257 bMinCmd:        BOOL; // Trig min position
258 bMaxCmd:        BOOL; // Trig max position
259 fMinPosNs:      REAL; // Min position setpoint
260 fMaxPosNs:      REAL; // Max position setpoint
261
262 //Limits IN_Out
263 fPosNs:         REAL; // Position setpoint
264 nMinRaw:        DINT; // Min position absolute value
265 nMaxRaw:        DINT; // Max position absolute value
266
267 //Limits out
268 bFwdLimSt:      BOOL; // Forward stop limit active
269 bBwdLimSt:      BOOL; // Backward stop limit active
270 fPosMe:         REAL; // Position measurement
271 nSpdNsDINT:     DINT; // INT_TO_DINT muunnosta varten GatelLogicFB:ssä
272
273 END_STRUCT
274
275 BEGIN
276
277 fMinPosNs := -90.0; // Min position setpoint
278 fMaxPosNs := 90.0;  // Max position setpoint
279 nMinRaw   := L#150000; // Min position absolute value
280 nMaxRaw   := L#160000; // Max position absolute value
281 nVelNs    := L#600;    // Velocity setpoint
282
283 END_DATA_BLOCK

```

Simatic Step 7 ohjelmalistaus, purkajaluukun servo 1:n LogicFB 1 (9)

SIMATIC

SphinxV5 20130624\

CPU\IM151-8F PN/DP CPU\...\FB12 - <offline>

FB12 - <offline>

"Distributor1LogicFB" Distributor 1 Main Logic FB

Name: Family:

Author: MHK_RKy Version: 1.0

Block version: 2

Time stamp Code: 03/20/2014 02:50:13 PM

Interface: 07/03/2013 06:39:32 PM

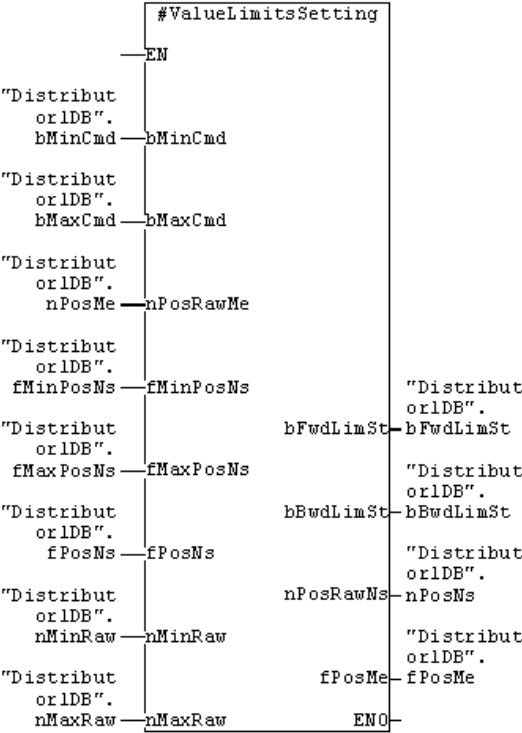
Lengths (block/logic/data): 02870 01868 00062

Name	Data Type	Address	Initial Value	Comment
IN		0.0		
OUT		0.0		
IN_OUT		0.0		
STAT		0.0		
Edge	Array [1..16] Of Bool	0.0		
bAlm	Array [1..16] Of Bool	2.0		
bAlmAck	Array [1..16] Of Bool	4.0		
bFlt	Array [1..16] Of Bool	6.0		
bFltAck	Array [1..16] Of Bool	8.0		
bILock	Array [1..16] Of Bool	10.0		
Alarm01	eventExtensionFB	12.0		
Fault01	eventExtensionFB	74.0		
Event01	eventExtensionFB	136.0		
ILock01	eventExtensionFB	198.0		
Limit01	LimitFB	260.0		
Position	SinaPosFB	408.0		
ValueLimitsSetting	PosLimFB	496.0		
TEMP		0.0		
MarginMax	DInt	0.0		
MarginMin	DInt	4.0		
Margin	Real	8.0		
Distributor1ActPosInReal	Real	12.0		
InRange	Bool	16.0		
Plug	Bool	16.1		
Accel	Word	18.0		
Decel	Word	20.0		
SpdOverride	Word	22.0		
Abs_AdjDone	Bool	24.0		
AbsHomeDone	Bool	24.1		

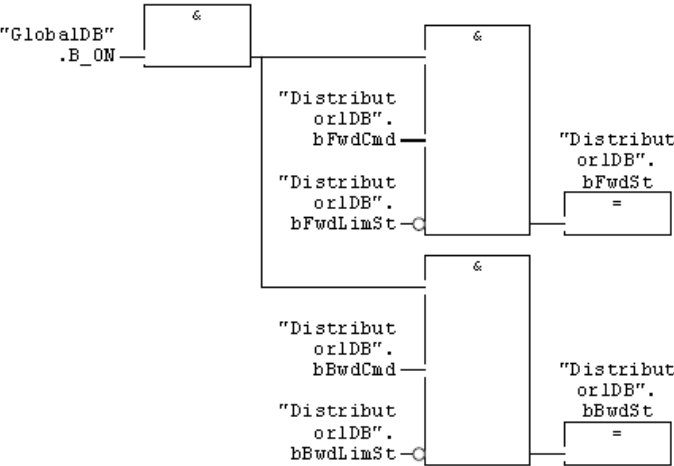
Simatic Step 7 ohjelmalistaus, purkajaluukun servo 1:n LogicFB 2 (9)

Block: FB12 Distributor1 FB
Copyright (C) Outokumpu technology, Espoo Finland =====
NOTES: Revision History :

Network: 1



Network: 2 Jog forward / backward



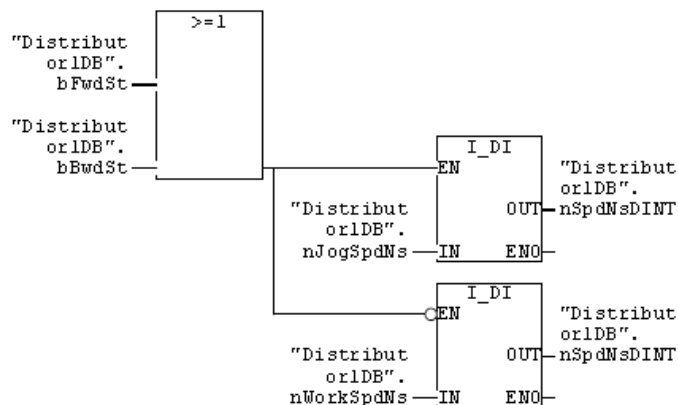
Simatic Step 7 ohjelmallistaus, purkajaluukun servo 1:n LogicFB 3 (9)

Network: 3 Speed HMI setting [deg/s] to convert into rpm

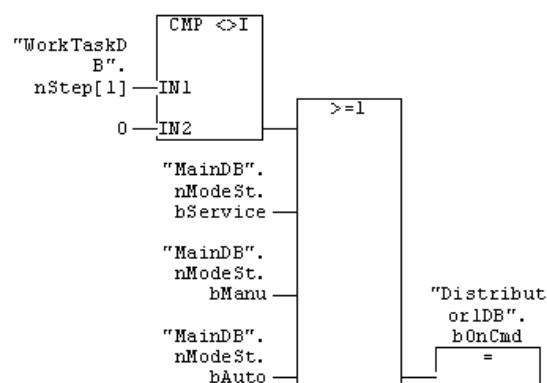
4.7.2013/RKy

Lisäsin tämän jotta voisin syöttää HMI:ltä oman nopeuden joggausta ja paikka-ajoa varten sekä käyttää työkierrossa jaksonajasta laskettua nopeutta. HMI:ltä syötteet yksikössä [deg/s] joten siirretään syötteet muuttujalle nSpdNsDINT, rpm:ksi muuttamista varten.

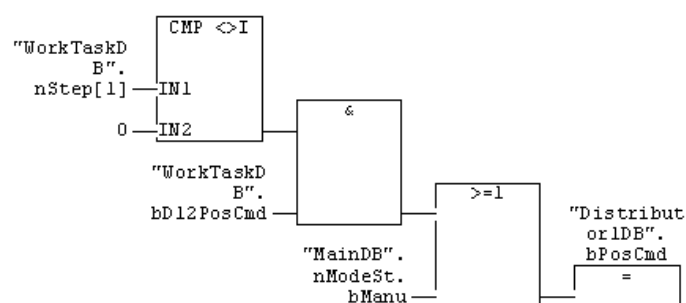
HUOM! nWorkSpdNs nopeusasetus paikoitusajoa varten. Työkierron aikainen nopeus lasketaan jaksonajasta FB Seq01:ssä.



Network: 4 Drive On / Off command

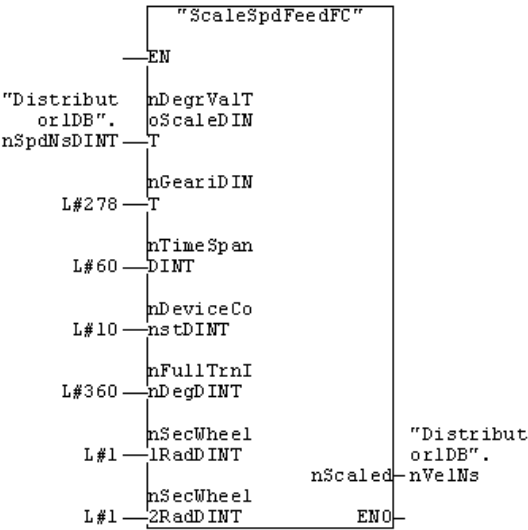


Network: 5 Positioning On /Off



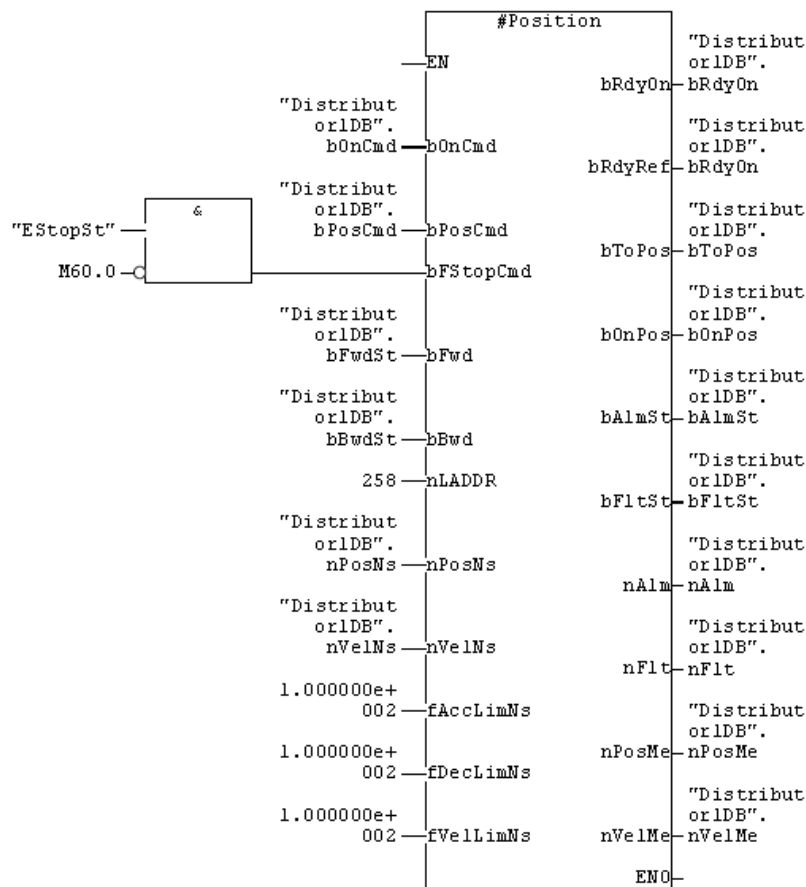
Simatic Step 7 ohjelmalistaus, purkajaluukun servo 1:n LogicFB 4 (9)

Network: 6	Scale deg/s into rpm
3.7.13/RKy Distributor 1: - Secondary wheel 1 radius 80 [mm]. NOTE: Here used value 1 for nSecWheel1RadDINT so far. - Secondary wheel 2 radius 540 [mm]. NOTE: Here used value 1 so far. NOTE: Here used value 1 for nSecWheel2RadDINT so far.	



Simatic Step 7 ohjelmalistaus, purkajaluukun servo 1:n LogicFB 5 (9)

Network: 7



Network: 8 ALARM

Network: 9 FAULT

Network: 10 I-LOCK

The diagram shows the pin connections for the #Alarm01 module. The pins are arranged in a vertical column on the left, with their corresponding signals and internal connections listed to the right.

Pin	Signal	Internal Connection
EN		
"Distribut orlDB". nMachSt.	bAlmAckCmd	bUnAckSt
bAlmAckCmd	bRstCmd	
T#2S	nOnTime	bEventSt
"Distribut orlDB". nAWl	nEW	nExtendAW
"Distribut orlDB". nAWlAck	nUnAckEW	EN0

The diagram shows the timing of the #Fault01 signal relative to several control and status signals. The signals are:

- #Fault01**: The primary signal being monitored, shown as a high pulse.
- EN**: Enable signal, which transitions from low to high at the start of the event.
- bRstCmd**: Reset command signal, which transitions from low to high during the event.
- nOnTime**: Not on time signal, which transitions from high to low during the event.
- nFW1**: Not full width 1 signal, which transitions from high to low during the event.
- nUnAckEW**: Not unacknowledged event width signal, which transitions from high to low during the event.
- bUnAckSt**: Bit unacknowledged status signal, which transitions from low to high during the event.
- bEventSt**: Bit event status signal, which transitions from low to high during the event.
- nExtendFW**: Not extend full width signal, which transitions from high to low during the event.
- EN0**: Enable 0 signal, which transitions from high to low during the event.

The diagram illustrates the sequence of events and the state of these signals during the #Fault01 event.

```

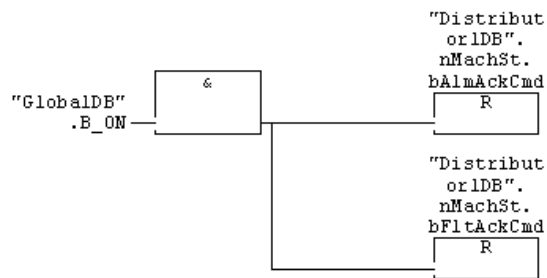
graph TD
    EN -- "#Event01" --> bUnAckSt
    bUnAckSt -- "GlobalDB".B_ON --> bEventSt
    bEventSt -- "T#2S" --> nOnTime
    nOnTime -- "'Distribut orlDB'.nEW1" --> nEW
    nEW -- "nUnAckEW" --> nEEW
    nEEW -- "nUnAckEW" --> ENO
  
```

```

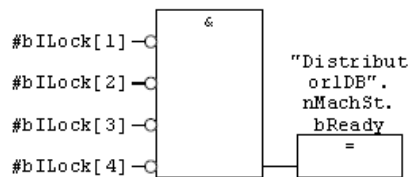
graph LR
    #ILock01 --- EN
    EN --- bUnAckSt
    bUnAckSt --- bEventSt
    bEventSt --- #bILock1
    #bILock1 --- "Distribut or IDE".nExtendIW[1]
    "Distribut or IDE".nExtendIW[1] --- nEW
    nEW --- nUnAckEW
    nUnAckEW --- ENO
    "GlobalDB".B_ON --- bRstCmd
    bRstCmd --- nOnTime
    nOnTime --- nIW1
    nIW1 --- nEW
    
```

Simatic Step 7 ohjelmallistaus, purkajaluukun servo 1:n LogicFB 7 (9)

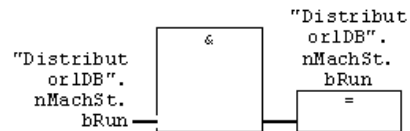
Network: 15 Reset of alarm/fault acknowledgement command



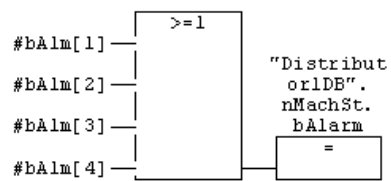
Network: 16 Status for HMI : READY TO START



Network: 17 Status for HMI : RUN

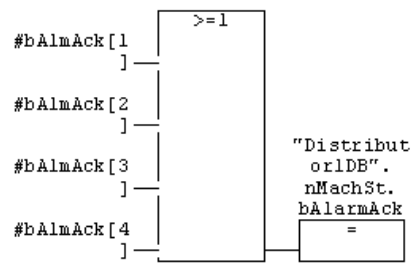


Network: 18 Alarm active

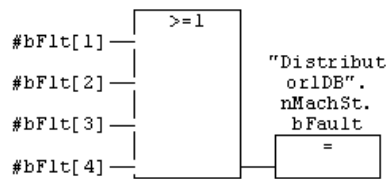


Simatic Step 7 ohjelmalistaus, purkajaluukun servo 1:n LogicFB 8 (9)

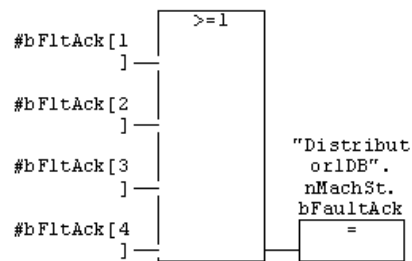
Network: 19 Unacknowledged alarm active



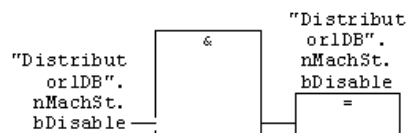
Network: 20 Fault active



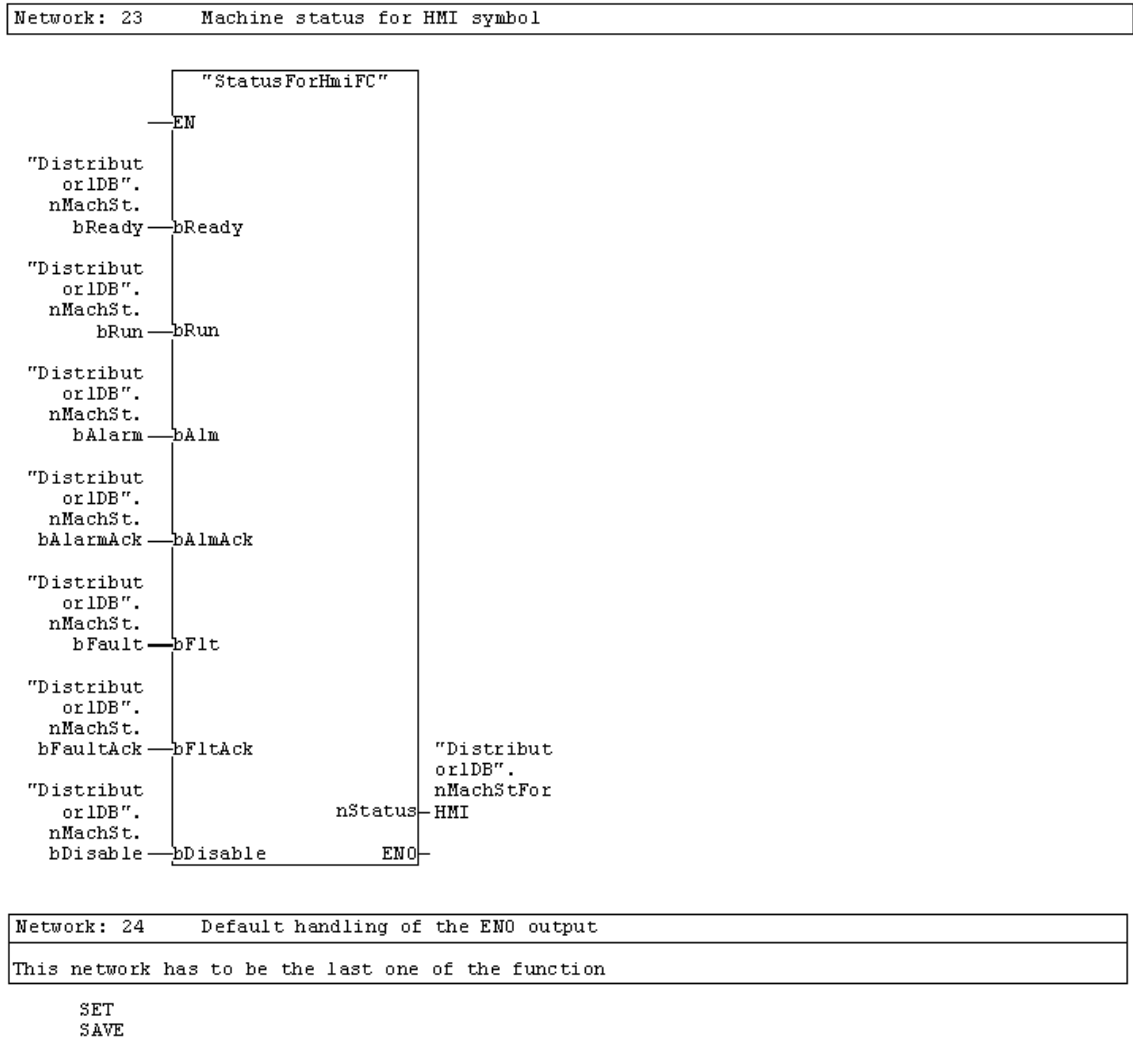
Network: 21 Unacknowledged fault active



Network: 22 Disable



Simatic Step 7 ohjelmalistaus, purkajaluukun servo 1:n LogicFB 9 (9)



Simatic Step 7 ohjelmalistaus, purkajaluukun servo 2:n DB

1 (5)

SphinxV5 20130624\CPU\IM151-8F PN/DP CPU\...\DB Distributor2

```

1  (* Copyright (C) Outokumpu technology, Espoo Finland
2  =====
3  NOTES :
4  Revision History :
5  =====*)
6
7  DATA_BLOCK "Distributor2DB"
8  TITLE = 'Distributor 2 DB'
9  VERSION : 1.1
10 AUTHOR : RKy
11
12 STRUCT
13
14 nModeCmd: // Main mode commands
15 STRUCT
16 bDummy08: BOOL; // Bit08:
17 bDummy09: BOOL; // Bit09:
18 bDummy10: BOOL; // Bit10:
19 bDummy11: BOOL; // Bit11:
20 bDummy12: BOOL; // Bit12:
21 bDummy13: BOOL; // Bit13:
22 bDummy14: BOOL; // Bit14:
23 bDummy15: BOOL; // Bit15:
24 bEStop:   BOOL; // Bit00: EMERGENCY STOP
25 bStop:    BOOL; // Bit01: STOP
26 bService: BOOL; // Bit02: SERVICE
27 bAux:     BOOL; // Bit03: AUX
28 bManu:    BOOL; // Bit04: MANUAL
29 bSemi:    BOOL; // Bit05: SEMI-AUTOMATIC
30 bAuto:    BOOL; // Bit06: AUTOMATIC
31 bDummy07: BOOL; // Bit07:
32 END_STRUCT;
33
34 nModeSt: // Main mode status
35 STRUCT
36 bMoveOK:  BOOL; // Bit08: Movement allowed
37 bDummy09: BOOL; // Bit09:
38 bDummy10: BOOL; // Bit10:
39 bDummy11: BOOL; // Bit11:
40 bDummy12: BOOL; // Bit12:
41 bDummy13: BOOL; // Bit13:
42 bDummy14: BOOL; // Bit14:
43 bDummy15: BOOL; // Bit15:
44 bEStop:   BOOL; // Bit00: EMERGENCY STOP
45 bStopped: BOOL; // Bit01: STOPPED
46 bService: BOOL; // Bit02: SERVICE
47 bAux:     BOOL; // Bit03: AUX
48 bManu:    BOOL; // Bit04: MANUAL
49 bSemi:    BOOL; // Bit05: SEMI-AUTOMATIC
50 bAuto:    BOOL; // Bit06: AUTOMATIC
51 bDummy07: BOOL; // Bit07:
52 END_STRUCT;
53
54 nModeStForHMI: INT; // Main mode for HMI
55
56 nMachSt: // Machine status
57 STRUCT

```

Simatic Step 7 ohjelmalistaus, purkajaluukun servo 2:n DB

2 (5)

```

58 bAlmAckCmd:  BOOL; // Alarm acknowledgement command
59 bFltAckCmd:  BOOL; // Fault acknowledgement command
60 bDummy10:    BOOL;
61 bDummy11:    BOOL;
62 bDummy12:    BOOL;
63 bDummy13:    BOOL;
64 bDummy14:    BOOL;
65 bDummy15:    BOOL;
66 bReady:     BOOL; // Ready
67 bRun:       BOOL; // Run
68 bAlarm:     BOOL; // Alarm
69 bAlarmAck:  BOOL; // Unacknowledged alarm
70 bFault:     BOOL; // Fault
71 bFaultAck:  BOOL; // Unacknowledged fault
72 bDisable:   BOOL; // Disable
73 bDummy05:   BOOL;
74 bDummy06:   BOOL;
75 bDummy07:   BOOL;
76 END_STRUCT;
77
78 nMachStForHMI: INT; // Machine status for HMI
79
80 nAW1: // Alarm word 1
81 STRUCT
82 bDummy08:    BOOL; // Bit08:
83 bDummy09:    BOOL; // Bit09:
84 bDummy10:    BOOL; // Bit10:
85 bDummy11:    BOOL; // Bit11:
86 bDummy12:    BOOL; // Bit12:
87 bDummy13:    BOOL; // Bit13:
88 bDummy14:    BOOL; // Bit14:
89 bDummy15:    BOOL; // Bit15:
90 bPosDevWith2TooHigh: BOOL; // Bit00: Pos deviation servo 2 too high
91 bDummy01:    BOOL; // Bit01:
92 bDummy02:    BOOL; // Bit02:
93 bDummy03:    BOOL; // Bit03:
94 bDummy04:    BOOL; // Bit04:
95 bDummy05:    BOOL; // Bit05:
96 bDummy06:    BOOL; // Bit06:
97 bDummy07:    BOOL; // Bit07:
98 END_STRUCT;
99
100 nAW1Ack: // Unacknowledged Alarm word 1
101 STRUCT
102 bDummy08:  BOOL; // Bit08:
103 bDummy09:  BOOL; // Bit09:
104 bDummy10:  BOOL; // Bit10:
105 bDummy11:  BOOL; // Bit11:
106 bDummy12:  BOOL; // Bit12:
107 bDummy13:  BOOL; // Bit13:
108 bDummy14:  BOOL; // Bit14:
109 bDummy15:  BOOL; // Bit15:
110 bDummy00:  BOOL; // Bit00:
111 bDummy01:  BOOL; // Bit01:
112 bDummy02:  BOOL; // Bit02:
113 bDummy03:  BOOL; // Bit13:
114 bDummy04:  BOOL; // Bit04:
115 bDummy05:  BOOL; // Bit05:
116 bDummy06:  BOOL; // Bit06:

```

Simatic Step 7 ohjelmalistaus, purkajaluukun servo 2:n DB

3 (5)

```

117 bDummy07: BOOL; // Bit07:
118 END_STRUCT;
119
120 nAWDummy: ARRAY[1..6] OF WORD; // Alarm word dummy
121 nExtendAW: ARRAY[1..4] OF WORD; // Extended Alarm for HMI
122
123 nFW1: // Fault word 1
124 STRUCT
125 bDummy08:      BOOL; // Bit08:
126 bDummy09:      BOOL; // Bit09:
127 bDummy10:      BOOL; // Bit10:
128 bDummy11:      BOOL; // Bit11:
129 bDummy12:      BOOL; // Bit12:
130 bDummy13:      BOOL; // Bit13:
131 bDummy14:      BOOL; // Bit14:
132 bDummy15:      BOOL; // Bit15:
133 bServodrive:    BOOL; // Bit00: Servodrive
134 bHomeNotOk:     BOOL; // Bit01: Home not Ok
135 bFollowingError: BOOL; // Bit02: Following error
136 bDummy03:      BOOL; // Bit03:
137 bDummy04:      BOOL; // Bit04:
138 bDummy05:      BOOL; // Bit05:
139 bDummy06:      BOOL; // Bit06:
140 bDummy07:      BOOL; // Bit07:
141 END_STRUCT;
142
143 nFW1Ack: // Unacknowledged Fault word 1
144 STRUCT
145 bDummy08: BOOL; // Bit08:
146 bDummy09: BOOL; // Bit09:
147 bDummy10: BOOL; // Bit10:
148 bDummy11: BOOL; // Bit11:
149 bDummy12: BOOL; // Bit12:
150 bDummy13: BOOL; // Bit13:
151 bDummy14: BOOL; // Bit14:
152 bDummy15: BOOL; // Bit15:
153 bDummy00: BOOL; // Bit00:
154 bDummy01: BOOL; // Bit01:
155 bDummy02: BOOL; // Bit02:
156 bDummy03: BOOL; // Bit03:
157 bDummy04: BOOL; // Bit04:
158 bDummy05: BOOL; // Bit05:
159 bDummy06: BOOL; // Bit06:
160 bDummy07: BOOL; // Bit07:
161 END_STRUCT;
162
163 nFWDummy: ARRAY[1..6] OF WORD; // Fault word dummy
164 nExtendFW: ARRAY[1..4] OF WORD; // Extended Fault for HMI
165
166 nEW1: // Event word 1
167 STRUCT
168 bDummy08: BOOL; // Bit08:
169 bDummy09: BOOL; // Bit09:
170 bDummy10: BOOL; // Bit10:
171 bDummy11: BOOL; // Bit11:
172 bDummy12: BOOL; // Bit12:
173 bDummy13: BOOL; // Bit13:
174 bDummy14: BOOL; // Bit14:
175 bDummy15: BOOL; // Bit15:

```

Simatic Step 7 ohjelmallistaus, purkajaluukun servo 2:n DB

4 (5)

```

176 bDummy00: BOOL; // Bit00:
177 bDummy01: BOOL; // Bit01:
178 bDummy02: BOOL; // Bit02:
179 bDummy03: BOOL; // Bit03:
180 bDummy04: BOOL; // Bit04:
181 bDummy05: BOOL; // Bit05:
182 bDummy06: BOOL; // Bit06:
183 bDummy07: BOOL; // Bit07:
184 END_STRUCT;
185
186 nEWDummy: ARRAY[1..6] OF WORD; // Event word dummy
187 nExtendEW: ARRAY[1..4] OF WORD; // Extended Event for HMI
188
189 nIW1: // I-Lock word 1
190 STRUCT
191 bDummy08: BOOL; // Bit08:
192 bDummy09: BOOL; // Bit09:
193 bDummy10: BOOL; // Bit10:
194 bDummy11: BOOL; // Bit11:
195 bDummy12: BOOL; // Bit12:
196 bDummy13: BOOL; // Bit13:
197 bDummy14: BOOL; // Bit14:
198 bDummy15: BOOL; // Bit15:
199 bDummy00: BOOL; // Bit00:
200 bDummy01: BOOL; // Bit01:
201 bDummy02: BOOL; // Bit02:
202 bDummy03: BOOL; // Bit03:
203 bDummy04: BOOL; // Bit04:
204 bDummy05: BOOL; // Bit05:
205 bDummy06: BOOL; // Bit06:
206 bDummy07: BOOL; // Bit07:
207 END_STRUCT;
208
209 nIWDummy: ARRAY[1..6] OF WORD; // I-Lock word dummy
210 nExtendIW: ARRAY[1..4] OF WORD; // Extended I-Lock for HMI
211
212 bHoming: BOOL; // Distributor 1 Homing start bit
213 nNegatSwingPar: INT; // Distributor swing position setpoint min
214 nPositSwingPar: INT; // Distributor swing position setpoint max
215 bDrvToSetPntCmd: BOOL; // Distributor 1 to setpoint command
216 bMechConnOpenCmd: BOOL; // Distributor mechanical connection open
217
218 nMoveFactorPar: REAL; // Adjustment factor
219 nWaitDelayPar: TIME; // Wait time at the end of movement [s]
220
221 nJogSpdNs: INT; // Distributor servo 1 jog forward speed
222 nHomeSpdNs: INT; // Distributor servo 1 homing speed.
223 nWorkSpdNs: INT; // Distributor servo 1 working speed.
224 nPosDevMarginPar: REAL; // Distributor pos deviation servo 1 and 2
225
226 //IN
227
228
229
230 bOnCmd: BOOL; // Drive On / Off command
231 bPosCmd: BOOL; // Positioning On /Off
232 bFStopCmd: BOOL; // Drive Fast stop (0 = FStop active)
233 bFwdCmd: BOOL; // Jog forward command
234 bFwdSt: BOOL; // Jog forward status
235 bBwdCmd: BOOL; // Jog backward command
236 bBwdSt: BOOL; // Jog backward status
237 nLADDR: INT; // Telegram first address from HW configuration

```

Simatic Step 7 ohjelmalistaus, purkajaluukun servo 2:n DB

5 (5)

```
238 nPosNs:      DINT; // Position setpoint
239 nVelNs:      DINT; // Velocity setpoint
240 fAccLimNs:   REAL; // Acceleration override (%)
241 fDecLimNs:   REAL; // Deceleration override (%)
242 fVelLimNs:   REAL; // Velocity override (%)
243
244 //OUT
245 bRdyOn:      BOOL; // Drive ready for switch on (Enable)
246 bRdyRef:     BOOL; // Drive ready for reference (Running)
247 bToPos:      BOOL; // Positionning running
248 bOnPos:      BOOL; // Target position reached
249 bAlmSt:      BOOL; // Drive alarm status
250 bFltSt:      BOOL; // Drive fault status
251 nAlm:        INT;  // Alarm code
252 nFlt:        INT;  // Fault code
253 nPosMe:      DINT; // Position measurement
254 nVelMe:      DINT; // Max position absolute value
// Velocity measurement
255
256 //Limits IN
257 bMinCmd:     BOOL; // Trig min position
258 bMaxCmd:     BOOL; // Trig max position
259 fMinPosNs:   REAL; // Min position setpoint
260 fMaxPosNs:   REAL; // Max position setpoint
261
262 //Limits IN_OUT
263 fPosNs:      REAL; // Position setpoint
264 nMinRaw:     DINT; // Min position absolute value
265 nMaxRaw:     DINT;
266
267 //Limits out
268 bFwdLimSt:   BOOL; // Forward stop limit active
269 bBwdLimSt:   BOOL; // Backward stop limit active
270 fPosMe:      REAL; // Position measurement
271 nSpdNsDINT:  DINT; // Muuttuja INT_TO_DINT muunnosta varten
272
273 END_STRUCT
274
275 BEGIN
276
277 END_DATA_BLOCK
```

Simatic Step 7 ohjelmalistaus, purkajaluukun servo 2:n LogicFB 1 (6)

SIMATIC

SphinxV5 20130624\

CPU\IM151-8F PN/DP CPU\...\FB13 - <offline>

FB13 - <offline>

"Distributor2LogicFB" Distributor 2 Main Logic FB

Name: **Family:****Author:** MHK_RKy **Version:** 1.0**Block version:** 2**Time stamp Code:** 03/20/2014 02:52:26 PM**Interface:** 06/27/2013 09:01:35 PM**Lengths (block/logic/data):** 02542 01554 00034

Name	Data Type	Address	Initial Value	Comment
IN		0.0		
OUT		0.0		
IN_OUT		0.0		
STAT		0.0		
Edge	Array [1..16] Of Bool	0.0		
bAlm	Array [1..16] Of Bool	2.0		
bAlmAck	Array [1..16] Of Bool	4.0		
bFlt	Array [1..16] Of Bool	6.0		
bFltAck	Array [1..16] Of Bool	8.0		
bILock	Array [1..16] Of Bool	10.0		
Alarm01	eventExtensionFB	12.0		
Fault01	eventExtensionFB	74.0		
Event01	eventExtensionFB	136.0		
ILock01	eventExtensionFB	198.0		
Limit01	LimitFB	260.0		
Position	FB 105	408.0		
ValueLimitsSetting	PosLimFB	496.0		
TEMP		0.0		
MarginMax	DInt	0.0		
MarginMin	DInt	4.0		
Margin	Real	8.0		
Distributor2ActPosInReal	Real	12.0		
InRange	Bool	16.0		
Plug	Bool	16.1		
Accel	Word	18.0		
Decel	Word	20.0		
SpdOverride	Word	22.0		
Abs_AdjDone	Bool	24.0		
AbsHomeDone	Bool	24.1		

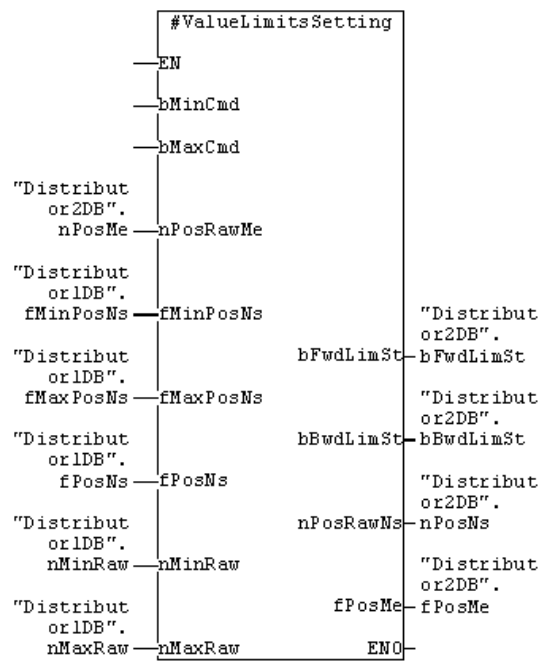
Simatic Step 7 ohjelmalistaus, purkajaluukun servo 2:n LogicFB 2 (6)

Block: FB13 Distributor2 FB

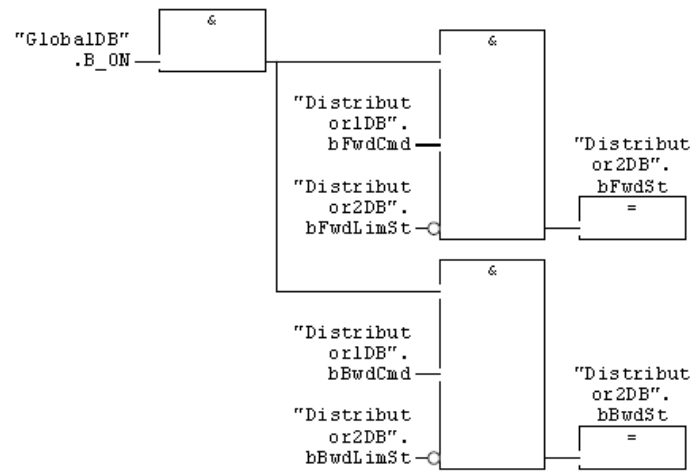
Copyright (C) Outokumpu technology, Espoo Finland
=====

NOTES:
Revision History :

Network: 1

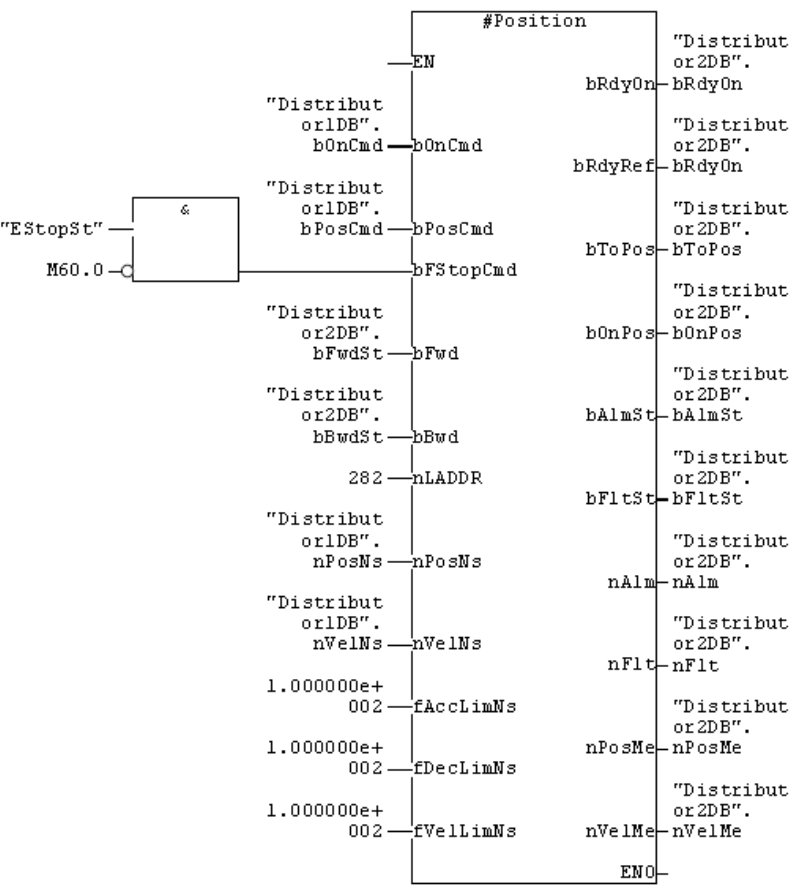


Network: 2 Jog forward / backward



Simatic Step 7 ohjelmalistaus, purkajaluukun servo 2:n LogicFB 3 (6)

Network: 3



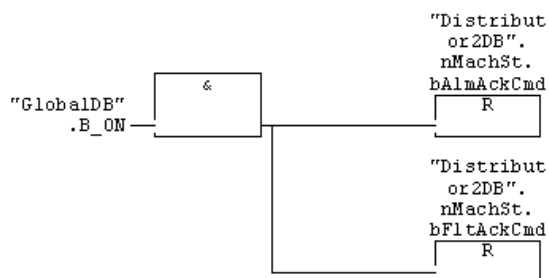
Network: 4 ALARM

Network: 5 FAULT

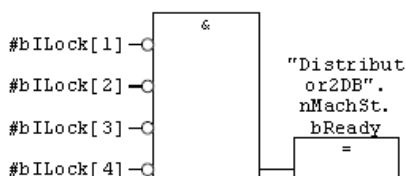
Network: 6 I-LOCK

Simatic Step 7 ohjelmallistaus, purkajaluukun servo 2:n LogicFB 5 (6)

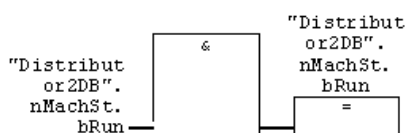
Network: 11 Reset of alarm/fault acknowledgement command



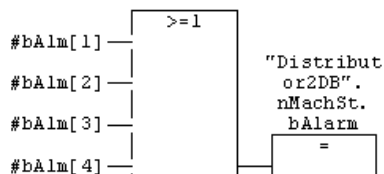
Network: 12 Status for HMI : READY TO START



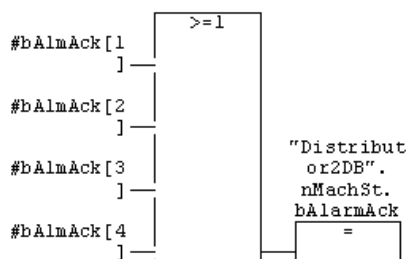
Network: 13 Status for HMI : RUN



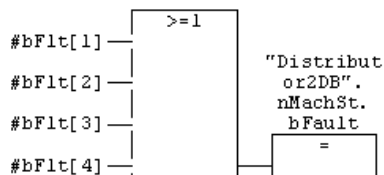
Network: 14 Alarm active



Network: 15 Unacknowledged alarm active

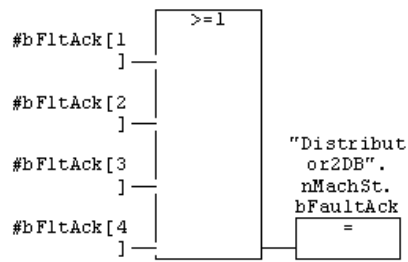


Network: 16 Fault active

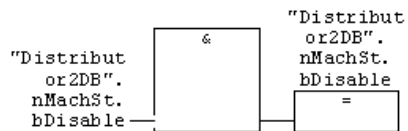


Simatic Step 7 ohjelmalistaus, purkajaluukun servo 2:n LogicFB 6 (6)

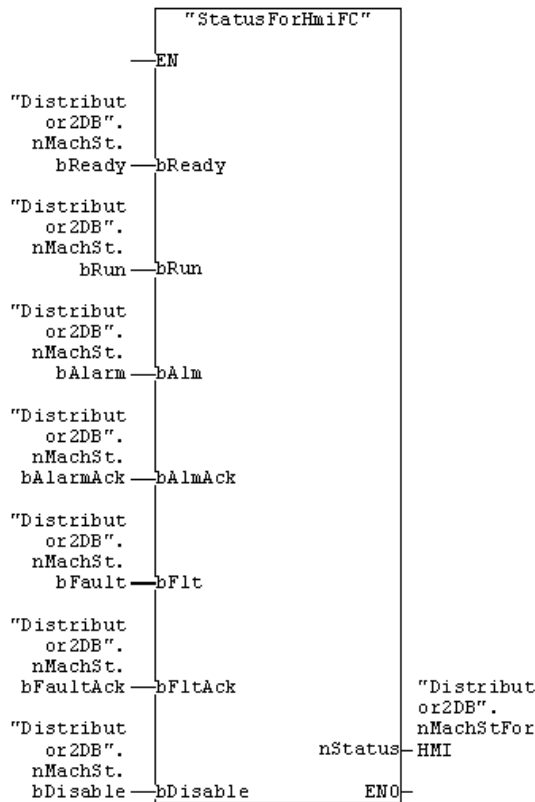
Network: 17 Unacknowledged fault active



Network: 18 Disable



Network: 19 Machine status for HMI symbol



Network: 20 Default handling of the ENO output

This network has to be the last one of the function

SET
SAVE

Simatic Step 7 ohjelmalistaus, sektoriluukku 1 servon DB

1 (5)

SphinxV5 20130624\CPU\IM151-8F PN/DP CPU\...\DB Gate1

```

1  (* Copyright (C) Outokumpu technology, Espoo Finland
2  =====
3  NOTES :
4  Revision History :
5  =====*)
6
7  DATA_BLOCK "Gate1DB"
8  TITLE = 'Gate 1 DB'
9  VERSION : 1.1
10 AUTHOR : RKy
11
12 STRUCT
13
14 nModeCmd: // Main mode commands
15 STRUCT
16 bDummy08: BOOL; // Bit08:
17 bDummy09: BOOL; // Bit09:
18 bDummy10: BOOL; // Bit10:
19 bDummy11: BOOL; // Bit11:
20 bDummy12: BOOL; // Bit12:
21 bDummy13: BOOL; // Bit13:
22 bDummy14: BOOL; // Bit14:
23 bDummy15: BOOL; // Bit15:
24 bEStop:   BOOL; // Bit00: EMERGENCY STOP
25 bStop:    BOOL; // Bit01: STOP
26 bService: BOOL; // Bit02: SERVICE
27 bAux:     BOOL; // Bit03: AUX
28 bManu:    BOOL; // Bit04: MANUAL
29 bSemi:    BOOL; // Bit05: SEMI-AUTOMATIC
30 bAuto:    BOOL; // Bit06: AUTOMATIC
31 bDummy07: BOOL; // Bit07:
32 END_STRUCT;
33
34 nModeSt: // Main mode status
35 STRUCT
36 bMoveOK:  BOOL; // Bit08: Movement allowed
37 bDummy09: BOOL; // Bit09:
38 bDummy10: BOOL; // Bit10:
39 bDummy11: BOOL; // Bit11:
40 bDummy12: BOOL; // Bit12:
41 bDummy13: BOOL; // Bit13:
42 bDummy14: BOOL; // Bit14:
43 bDummy15: BOOL; // Bit15:
44 bEStop:   BOOL; // Bit00: EMERGENCY STOP
45 bStopped: BOOL; // Bit01: STOPPED
46 bService: BOOL; // Bit02: SERVICE
47 bAux:     BOOL; // Bit03: AUX
48 bManu:    BOOL; // Bit04: MANUAL
49 bSemi:    BOOL; // Bit05: SEMI-AUTOMATIC
50 bAuto:    BOOL; // Bit06: AUTOMATIC
51 bDummy07: BOOL; // Bit07:
52 END_STRUCT;
53
54 nModeStForHMI: INT; // Main mode for HMI
55
56 nMachSt: // Machine status

```

Simatic Step 7 ohjelmalistaus, sektoriluukku 1 servon DB

2 (5)

```

57 STRUCT
58 bAlmAckCmd: BOOL; // Alarm acknowledgement command
59 bFltAckCmd: BOOL; // Fault acknowledgement command
60 bDummy10: BOOL;
61 bDummy11: BOOL;
62 bDummy12: BOOL;
63 bDummy13: BOOL;
64 bDummy14: BOOL;
65 bDummy15: BOOL;
66 bReady: BOOL; // Ready
67 bRun: BOOL; // Run
68 bAlarm: BOOL; // Alarm
69 bAlarmAck: BOOL; // Unacknowledged alarm
70 bFault: BOOL; // Fault
71 bFaultAck: BOOL; // Unacknowledged fault
72 bDisable: BOOL; // Disable
73 bDummy05: BOOL;
74 bDummy06: BOOL;
75 bDummy07: BOOL;
76 END_STRUCT;
77
78 nMachStForHMI: INT; // Machine status for HMI
79
80 nAW1: // Alarm word 1
81 STRUCT
82 bDummy08: BOOL; // Bit08:
83 bDummy09: BOOL; // Bit09:
84 bDummy10: BOOL; // Bit10:
85 bDummy11: BOOL; // Bit11:
86 bDummy12: BOOL; // Bit12:
87 bDummy13: BOOL; // Bit13:
88 bDummy14: BOOL; // Bit14:
89 bDummy15: BOOL; // Bit15:
90 bDummy00: BOOL; // Bit00:
91 bDummy01: BOOL; // Bit01:
92 bDummy02: BOOL; // Bit02:
93 bDummy03: BOOL; // Bit03:
94 bDummy04: BOOL; // Bit04:
95 bDummy05: BOOL; // Bit05:
96 bDummy06: BOOL; // Bit06:
97 bDummy07: BOOL; // Bit07:
98 END_STRUCT;
99
100 nAW1Ack: // Unacknowledged Alarm word 1
101 STRUCT
102 bDummy08: BOOL; // Bit08:
103 bDummy09: BOOL; // Bit09:
104 bDummy10: BOOL; // Bit10:
105 bDummy11: BOOL; // Bit11:
106 bDummy12: BOOL; // Bit12:
107 bDummy13: BOOL; // Bit03:
108 bDummy14: BOOL; // Bit14:
109 bDummy15: BOOL; // Bit15:
110 bDummy00: BOOL; // Bit00:
111 bDummy01: BOOL; // Bit01:
112 bDummy02: BOOL; // Bit02:
113 bDummy03: BOOL; // Bit03:
114 bDummy04: BOOL; // Bit04:
115 bDummy05: BOOL; // Bit05:

```

Simatic Step 7 ohjelmalistaus, sektoriluukku 1 servon DB

3 (5)

```

116 bDummy06: BOOL; // Bit06:
117 bDummy07: BOOL; // Bit07:
118 END_STRUCT;
119
120 nAWDummy: ARRAY[1..6] OF WORD; // Alarm word dummy
121 nExtendAW: ARRAY[1..4] OF WORD; // Extended Alarm for HMI
122
123 nFW1: // Fault word 1
124 STRUCT
125 bDummy08:      BOOL; // Bit08:
126 bDummy09:      BOOL; // Bit09:
127 bDummy10:      BOOL; // Bit10:
128 bDummy11:      BOOL; // Bit11:
129 bDummy12:      BOOL; // Bit12:
130 bDummy13:      BOOL; // Bit13:
131 bDummy14:      BOOL; // Bit14:
132 bDummy15:      BOOL; // Bit15:
133 bServodrive:    BOOL; // Bit00: Servodrive
134 bHomeNotOk:     BOOL; // Bit01: Home not Ok
135 bFollowingError: BOOL; // Bit02: Following error
136 bDummy03:      BOOL; // Bit03:
137 bDummy04:      BOOL; // Bit04:
138 bDummy05:      BOOL; // Bit05:
139 bDummy06:      BOOL; // Bit06:
140 bDummy07:      BOOL; // Bit07:
141 END_STRUCT;
142
143 nFW1Ack: // Unacknowledged Fault word 1
144 STRUCT
145 bDummy08: BOOL; // Bit08:
146 bDummy09: BOOL; // Bit09:
147 bDummy10: BOOL; // Bit10:
148 bDummy11: BOOL; // Bit11:
149 bDummy12: BOOL; // Bit12:
150 bDummy13: BOOL; // Bit13:
151 bDummy14: BOOL; // Bit14:
152 bDummy15: BOOL; // Bit15:
153 bDummy00: BOOL; // Bit00:
154 bDummy01: BOOL; // Bit01:
155 bDummy02: BOOL; // Bit02:
156 bDummy03: BOOL; // Bit03:
157 bDummy04: BOOL; // Bit04:
158 bDummy05: BOOL; // Bit05:
159 bDummy06: BOOL; // Bit06:
160 bDummy07: BOOL; // Bit07:
161 END_STRUCT;
162
163 nFWDummy: ARRAY[1..6] OF WORD; // Fault word dummy
164 nExtendFW: ARRAY[1..4] OF WORD; // Extended Fault for HMI
165
166 nEW1: // Event word 1
167 STRUCT
168 bDummy08: BOOL; // Bit08:
169 bDummy09: BOOL; // Bit09:
170 bDummy10: BOOL; // Bit10:
171 bDummy11: BOOL; // Bit11:
172 bDummy12: BOOL; // Bit12:
173 bDummy13: BOOL; // Bit13:

```


Simatic Step 7 ohjelmalistaus, sektoriluukku 1 servon DB

4 (5)

```

174 bDummy14: BOOL; // Bit14:
175 bDummy15: BOOL; // Bit15:
176 bDummy00: BOOL; // Bit00:
177 bDummy01: BOOL; // Bit01:
178 bDummy02: BOOL; // Bit02:
179 bDummy03: BOOL; // Bit03:
180 bDummy04: BOOL; // Bit04:
181 bDummy05: BOOL; // Bit05:
182 bDummy06: BOOL; // Bit06:
183 bDummy07: BOOL; // Bit07:
184 END_STRUCT;
185
186 nEWDummy: ARRAY[1..6] OF WORD; // Event word dummy
187 nExtendEW: ARRAY[1..4] OF WORD; // Extended Event for HMI
188
189 nIW1: // I-Lock word 1
190 STRUCT
191 bDummy08: BOOL; // Bit08:
192 bDummy09: BOOL; // Bit09:
193 bDummy10: BOOL; // Bit10:
194 bDummy11: BOOL; // Bit11:
195 bDummy12: BOOL; // Bit12:
196 bDummy13: BOOL; // Bit13:
197 bDummy14: BOOL; // Bit14:
198 bDummy15: BOOL; // Bit15:
199 bDummy00: BOOL; // Bit00:
200 bDummy01: BOOL; // Bit01:
201 bDummy02: BOOL; // Bit02:
202 bDummy03: BOOL; // Bit03:
203 bDummy04: BOOL; // Bit04:
204 bDummy05: BOOL; // Bit05:
205 bDummy06: BOOL; // Bit06:
206 bDummy07: BOOL; // Bit07:
207 END_STRUCT;
208
209 nIWDummy: ARRAY[1..6] OF WORD; // I-Lock word dummy
210 nExtendIW: ARRAY[1..4] OF WORD; // Extended I-Lock for HMI
211
212 bHoming: BOOL; // Gate 1 Homing start bit
213 nNegatSwingPar: INT; // Gate 1 position setpoint min
214 nPositSwingPar: INT; // Gate 1 position setpoint max
215 bDrvToSetPntCmd: BOOL; // Gate 1 to setpoint command
216 bMechConnOpenCmd: BOOL; //
217
218 nMoveFactorPar: REAL; // Adjustment factor
219 nWaitDelayPar: TIME; //
220
221 nJogSpdNs: INT; // Gate 1 servo jogging forward speed
222 nHomeSpdNs: INT; // Gate 1 servo homing speed.
223 nWorkSpdNs: INT; // Gate 1 working speed
224 nPosDevMarginPar: REAL; //
225
226 //IN
227 bOnCmd: BOOL; // Drive On / Off command
228 bPosCmd: BOOL; // Positioning On /Off
229 bFStopCmd: BOOL; // Drive Fast stop (0 = FStop active)
230 bFwdCmd: BOOL; // Jog forward command
231 bFwdSt: BOOL; // Jog forward status
232 bBwdCmd: BOOL; // Jog backward command

```

Simatic Step 7 ohjelmalistaus, sektoriluukku 1 servon DB

5 (5)

```

233 bBwdSt:      BOOL; // Jog backward status
234 nLADDR:      INT;  // Telegram first address from HW configuration
235 nPosNs:      DINT; // Position setpoint
236 nVelNs:      DINT; // Velocity setpoint
237 fAccLimNs:   REAL; // Acceleration override (%)
238 fDecLimNs:   REAL; // Deceleration override (%)
239 fVelLimNs:   REAL; // Velocity override (%)
240
241 //OUT
242 bRdyOn:      BOOL; // Drive ready for switch on (Enable)
243 bRdyRef:     BOOL; // Drive ready for reference (Running)
244 bToPos:      BOOL; // Positioning running
245 bOnPos:      BOOL; // Target position reached
246 bAlmSt:      BOOL; // Drive alarm status
247 bFltSt:      BOOL; // Drive fault status
248 nAlm:        INT;  // Alarm code
249 nFlt:        INT;  // Fault code
250 nPosMe:      DINT; // Position measurement
251 nVelMe:      DINT; // Velocity measurement
252
253 //Limits IN
254 bMinCmd:      BOOL; // Trig min position
255 bMaxCmd:      BOOL; // Trig max position
256 fMinPosNs:   REAL; // Min position setpoint
257 fMaxPosNs:   REAL; // Max position setpoint
258
259 //Limits IN_Out
260 fPosNs:       REAL; // Position setpoint
261 nMinRaw:      DINT; // Min position absolute value
262 nMaxRaw:      DINT; // Max position absolute value
263
264 //Limits out
265 bFwdLimSt:    BOOL; // Forward stop limit active
266 bBwdLimSt:    BOOL; // Backward stop limit active
267 fPosMe:       REAL; // Position measurement
268 nSpdNsDINT:   DINT; // Muuttuja INT_TO_DINT muunnosta varten
269
270 END_STRUCT
271
272 BEGIN
273
274 fMinPosNs := 0.0; // Min position setpoint
275 fMaxPosNs := 180.0; // Max position setpoint
276 nMinRaw := L#240000; // Min position absolute value
277 nMaxRaw := L#-400000; // Max position absolute value
278 nVelNs := L#600; // Velocity setpoint
279
280 END_DATA_BLOCK

```

Simatic Step 7 ohjelmallistaus, sektoriluukku 1 servon LogicFB 1 (7)

SIMATIC

SphinxV5 20130624\

CPU\IM151-8F PN/DP CPU\...\FB14 - <offline>

FB14 - <offline>

"GatelLogicFB" Gate 1 Main Logic FB

Name: **Family:****Author:** MHK_RKy **Version:** 1.0**Block version:** 2**Time stamp Code:** 03/20/2014 02:53:56 PM**Interface:** 07/03/2013 04:50:54 PM**Lengths (block/logic/data):** 02880 01878 00062

Name	Data Type	Address	Initial Value	Comment
IN		0.0		
OUT		0.0		
IN_OUT		0.0		
STAT		0.0		
Edge	Array [1..16] Of Bool	0.0		
bAlm	Array [1..16] Of Bool	2.0		
bAlmAck	Array [1..16] Of Bool	4.0		
bFlt	Array [1..16] Of Bool	6.0		
bFltAck	Array [1..16] Of Bool	8.0		
bILock	Array [1..16] Of Bool	10.0		
Alarm01	eventExtensionFB	12.0		
Fault01	eventExtensionFB	74.0		
Event01	eventExtensionFB	136.0		
ILock01	eventExtensionFB	198.0		
Limit01	LimitFB	260.0		
Position	SinaPosFB	408.0		
ValueLimitsSetting	PosLimFB	496.0		
TEMP		0.0		
MarginMax	DInt	0.0		
MarginMin	DInt	4.0		
Margin	Real	8.0		
GatelActPosInReal	Real	12.0		
InRange	Bool	16.0		
Plug	Bool	16.1		
Accel	Word	18.0		
Decel	Word	20.0		
SpdOverride	Word	22.0		
Abs_AdjDone	Bool	24.0		
AbsHomeDone	Bool	24.1		

Block: FB14 Gate1 FB

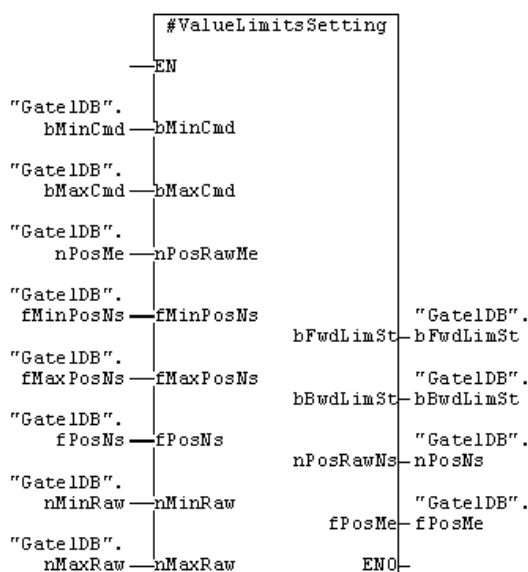
Copyright (C) Outokumpu technology, Espoo Finland

NOTES:

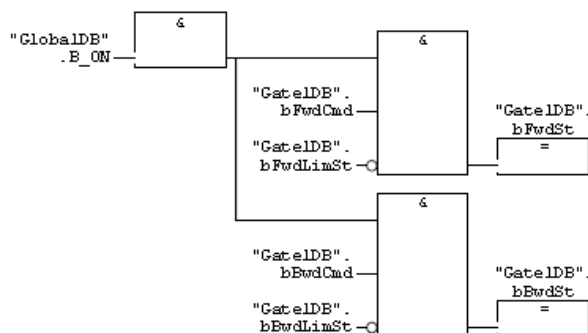
Revision History :

Simatic Step 7 ohjelmallistaus, sektoriluokku 1 servon LogicFB 2 (7)

Network: 1



Network: 2 Jog forward / backward

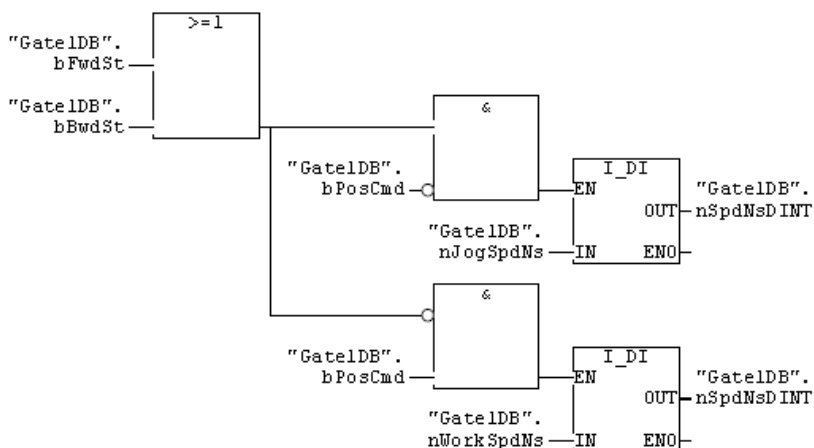


Network: 3 Speed HMI setting [deg/s] to convert into rpm

3.7.2013/RKy

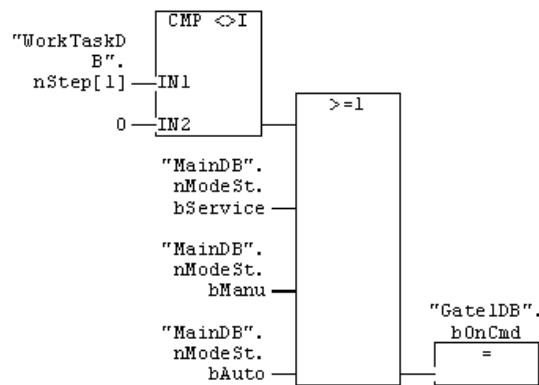
Lisäsin tämän jotta voisin syöttää HMI:ltä oman nopeuden joggausta ja paikka-ajoa varten.

HMI:ltä syötteen yksikössä [deg/s] joten siirretään syötteen muuttujalle `nSpdNsDINT`, rpm:ksi muuttamista varten.

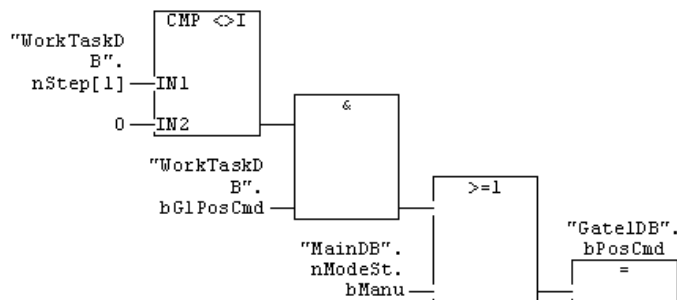


Simatic Step 7 ohjelmalistaus, sektoriluukku 1 servon LogicFB 3 (7)

Network: 4 Drive On / Off command



Network: 5 Positioning On /Off

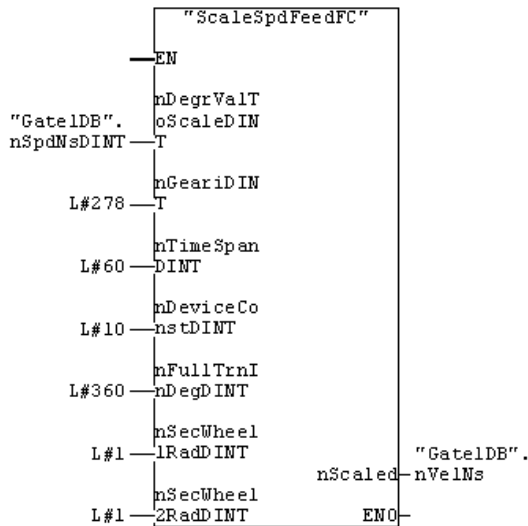


Network: 6 Scale deg/s into rpm

3.7.13/RKy

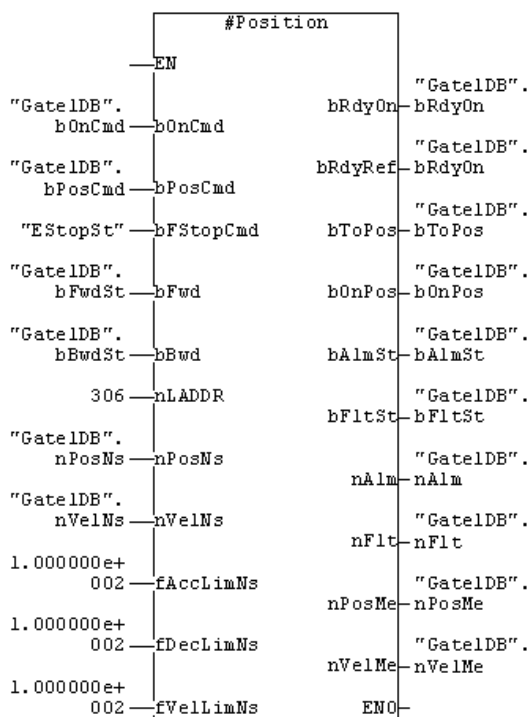
Gate 1:

- Secondary wheel 1 radius 80 [mm].
NOTE: Here used value 1 for nSecWheel1RadDINT so far.
- Secondary wheel 2 radius 160 [mm]. NOTE: Here used value 1 so far.
NOTE: Here used value 1 for nSecWheel2RadDINT so far.



Simatic Step 7 ohjelmallistaus, sektoriluukku 1 servon LogicFB 4 (7)

Network: 7

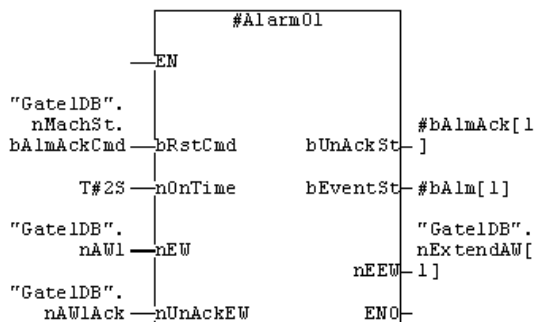


Network: 8 ALARM

Network: 9 FAULT

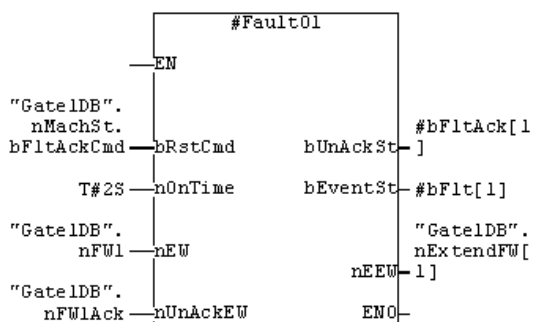
Network: 10 I-LOCK

Network: 11 Alarm extension module

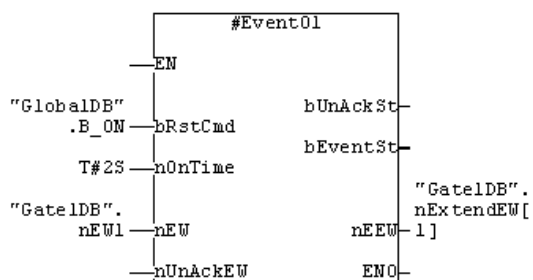


Simatic Step 7 ohjelmallistus, sektoriluukku 1 servon LogicFB 5 (7)

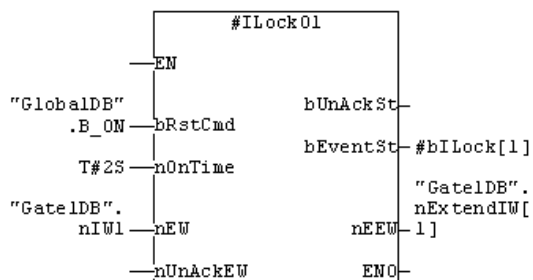
Network: 12 Fault extension module



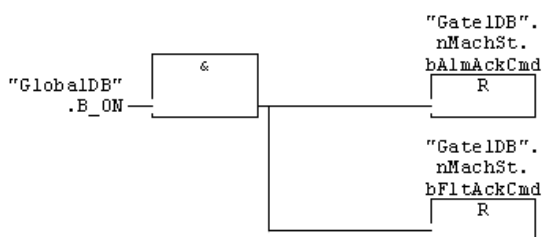
Network: 13 Event extension module



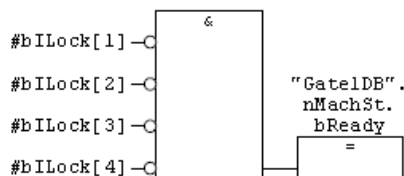
Network: 14 I-Lock extension module



Network: 15 Reset of alarm/fault acknowledgement command



Network: 16 Status for HMI : READY TO START



```
"GatelDB".nMachSt.bRun & "GatelDB".nMachSt.bRun =
```

```

#bAlm[1]
#bAlm[2]
#bAlm[3]
#bAlm[4]

```

```

#bAlmAck[1]
#bAlmAck[2]
#bAlmAck[3]
#bAlmAck[4]

```

>=1

"GatelDB".
nNachSt.
bAlarmAck
=

```

#bFlt[1] ---
#bFlt[2] ---
#bFlt[3] ---
#bFlt[4] ---

```

>=1

"Gate1DB".
nMachSt.
bFault
=

```

#bFltAck[1]
#bFltAck[2]
#bFltAck[3]
#bFltAck[4]

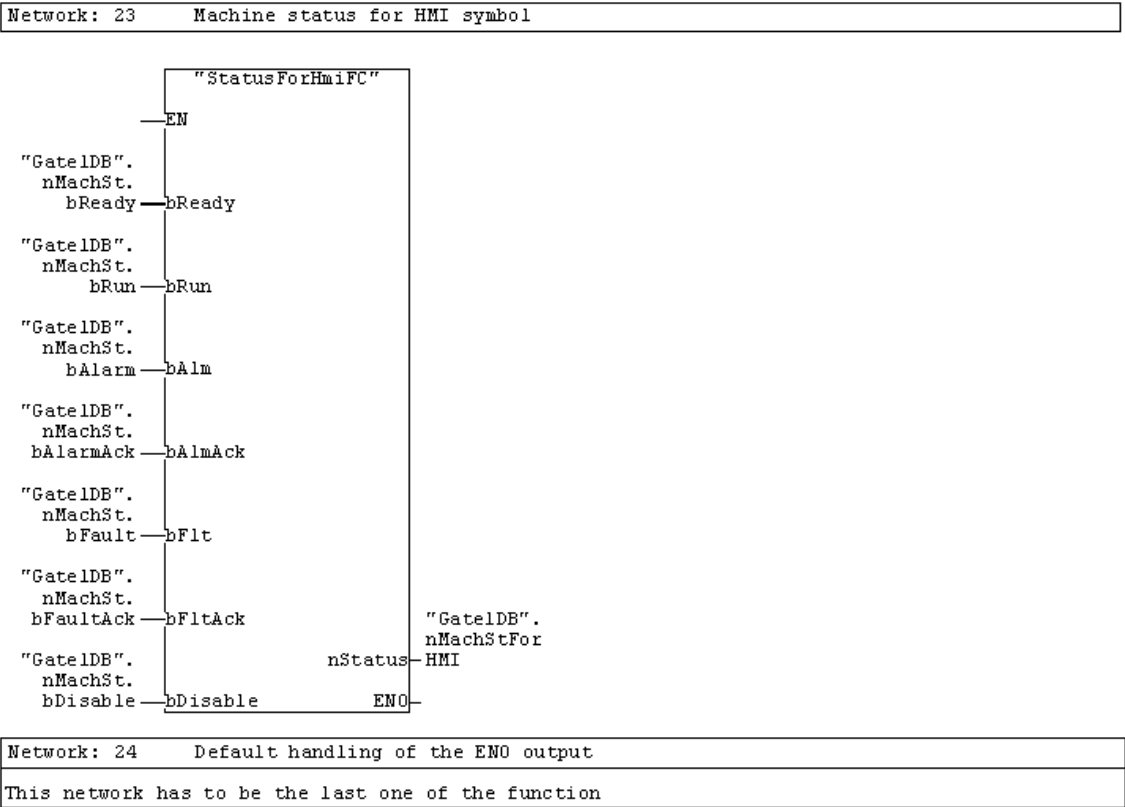
```

>=1

"GateIDB".
nNachSt.
bFaultAck
=

```
"GatelDB".nMachSt.bDisable & "GatelDB".nMachSt.bDisable =
```


Simatic Step 7 ohjelmallistus, sektoriluukku 1 servon LogicFB 7 (7)



SET
SAVE

Simatic Step 7 ohjelmalistaus, sektoriluukku 2 servon DB

1 (5)

SphinxV5 20130624\CPU\IM151-8F PN/DP CPU\...\DB Gate2

```

1  (* Copyright (C) Outokumpu technology, Espoo Finland
2  =====
3  NOTES :
4  Revision History :
5  =====*)
6
7  DATA_BLOCK "Gate2DB"
8  TITLE = 'Gate 2 DB'
9  VERSION : 1.1
10 AUTHOR : RKy
11
12 STRUCT
13
14 nModeCmd: // Main mode commands
15 STRUCT
16 bDummy08: BOOL; // Bit08:
17 bDummy09: BOOL; // Bit09:
18 bDummy10: BOOL; // Bit10:
19 bDummy11: BOOL; // Bit11:
20 bDummy12: BOOL; // Bit12:
21 bDummy13: BOOL; // Bit13:
22 bDummy14: BOOL; // Bit14:
23 bDummy15: BOOL; // Bit15:
24 bEStop:   BOOL; // Bit00: EMERGENCY STOP
25 bStop:    BOOL; // Bit01: STOP
26 bService: BOOL; // Bit02: SERVICE
27 bAux:     BOOL; // Bit03: AUX
28 bManu:    BOOL; // Bit04: MANUAL
29 bSemi:    BOOL; // Bit05: SEMI-AUTOMATIC
30 bAuto:    BOOL; // Bit06: AUTOMATIC
31 bDummy07: BOOL; // Bit07:
32 END_STRUCT;
33
34 nModeSt: // Main mode status
35 STRUCT
36 bMoveOK:  BOOL; // Bit08: Movement allowed
37 bDummy09: BOOL; // Bit09:
38 bDummy10: BOOL; // Bit10:
39 bDummy11: BOOL; // Bit11:
40 bDummy12: BOOL; // Bit12:
41 bDummy13: BOOL; // Bit13:
42 bDummy14: BOOL; // Bit14:
43 bDummy15: BOOL; // Bit15:
44 bEStop:   BOOL; // Bit00: EMERGENCY STOP
45 bStopped: BOOL; // Bit01: STOPPED
46 bService: BOOL; // Bit02: SERVICE
47 bAux:     BOOL; // Bit03: AUX
48 bManu:    BOOL; // Bit04: MANUAL
49 bSemi:    BOOL; // Bit05: SEMI-AUTOMATIC
50 bAuto:    BOOL; // Bit06: AUTOMATIC
51 bDummy07: BOOL; // Bit07:
52 END_STRUCT;
53
54 nModeStForHMI: INT; // Main mode for HMI
55
56 nMachSt: // Machine status
57 STRUCT

```

Simatic Step 7 ohjelmalistaus, sektoriluukku 2 servon DB

2 (5)

```
58 bAlmAckCmd:  BOOL; // Alarm acknowledgement command
59 bFltAckCmd:  BOOL; // Fault acknowledgement command
60 bDummy10:    BOOL;
61 bDummy11:    BOOL;
62 bDummy12:    BOOL;
63 bDummy13:    BOOL;
64 bDummy14:    BOOL;
65 bDummy15:    BOOL;
66 bReady:      BOOL; // Ready
67 bRun:        BOOL; // Run
68 bAlarm:      BOOL; // Alarm
69 bAlarmAck:   BOOL; // Unacknowledged alarm
70 bFault:      BOOL; // Fault
71 bFaultAck:   BOOL; // Unacknowledged fault
72 bDisable:    BOOL; // Disable
73 bDummy05:    BOOL;
74 bDummy06:    BOOL;
75 bDummy07:    BOOL;
76 END_STRUCT;
77
78 nMachStForHMI: INT; // Machine status for HMI
79
80 nAW1: // Alarm word 1
81 STRUCT
82 bDummy08: BOOL; // Bit08:
83 bDummy09: BOOL; // Bit09:
84 bDummy10: BOOL; // Bit10:
85 bDummy11: BOOL; // Bit11:
86 bDummy12: BOOL; // Bit12:
87 bDummy13: BOOL; // Bit13:
88 bDummy14: BOOL; // Bit14:
89 bDummy15: BOOL; // Bit15:
90 bDummy00: BOOL; // Bit00:
91 bDummy01: BOOL; // Bit01:
92 bDummy02: BOOL; // Bit02:
93 bDummy03: BOOL; // Bit03:
94 bDummy04: BOOL; // Bit04:
95 bDummy05: BOOL; // Bit05:
96 bDummy06: BOOL; // Bit06:
97 bDummy07: BOOL; // Bit07:
98 END_STRUCT;
99
100 nAW1Ack: // Unacknowledged Alarm word 1
101 STRUCT
102 bDummy08: BOOL; // Bit08:
103 bDummy09: BOOL; // Bit09:
104 bDummy10: BOOL; // Bit10:
105 bDummy11: BOOL; // Bit11:
106 bDummy12: BOOL; // Bit12:
107 bDummy13: BOOL; // Bit13:
108 bDummy14: BOOL; // Bit14:
109 bDummy15: BOOL; // Bit15:
110 bDummy00: BOOL; // Bit00:
111 bDummy01: BOOL; // Bit01:
112 bDummy02: BOOL; // Bit02:
113 bDummy03: BOOL; // Bit03:
114 bDummy04: BOOL; // Bit04:
115 bDummy05: BOOL; // Bit05:
116 bDummy06: BOOL; // Bit06:
```

Simatic Step 7 ohjelmallistus, sektoriluukku 2 servon DB

3 (5)

```

117 bDummy07: BOOL; // Bit07:
118 END_STRUCT;
119
120 nAWDummy: ARRAY[1..6] OF WORD; // Alarm word dummy
121 nExtendAW: ARRAY[1..4] OF WORD; // Extended Alarm for HMI
122
123 nFW1: // Fault word 1
124 STRUCT
125 bDummy08:      BOOL; // Bit08:
126 bDummy09:      BOOL; // Bit09:
127 bDummy10:      BOOL; // Bit10:
128 bDummy11:      BOOL; // Bit11:
129 bDummy12:      BOOL; // Bit12:
130 bDummy13:      BOOL; // Bit13:
131 bDummy14:      BOOL; // Bit14:
132 bDummy15:      BOOL; // Bit15:
133 bServodrive:    BOOL; // Bit00: Servodrive
134 bHomeNotOk:     BOOL; // Bit01: Home not Ok
135 bFollowingError: BOOL; // Bit02: Following error
136 bDummy03:      BOOL; // Bit03:
137 bDummy04:      BOOL; // Bit04:
138 bDummy05:      BOOL; // Bit05:
139 bDummy06:      BOOL; // Bit06:
140 bDummy07:      BOOL; // Bit07:
141 END_STRUCT;
142
143 nFW1Ack: // Unacknowledged Fault word 1
144 STRUCT
145 bDummy08: BOOL; // Bit08:
146 bDummy09: BOOL; // Bit09:
147 bDummy10: BOOL; // Bit10:
148 bDummy11: BOOL; // Bit11:
149 bDummy12: BOOL; // Bit12:
150 bDummy13: BOOL; // Bit13:
151 bDummy14: BOOL; // Bit14:
152 bDummy15: BOOL; // Bit15:
153 bDummy00: BOOL; // Bit00:
154 bDummy01: BOOL; // Bit01:
155 bDummy02: BOOL; // Bit02:
156 bDummy03: BOOL; // Bit03:
157 bDummy04: BOOL; // Bit04:
158 bDummy05: BOOL; // Bit05:
159 bDummy06: BOOL; // Bit06:
160 bDummy07: BOOL; // Bit07:
161 END_STRUCT;
162
163 nFWDummy: ARRAY[1..6] OF WORD; // Fault word dummy
164 nExtendFW: ARRAY[1..4] OF WORD; // Extended Fault for HMI
165
166 nEW1: // Event word 1
167 STRUCT
168 bDummy08: BOOL; // Bit08:
169 bDummy09: BOOL; // Bit09:
170 bDummy10: BOOL; // Bit10:
171 bDummy11: BOOL; // Bit11:
172 bDummy12: BOOL; // Bit12:
173 bDummy13: BOOL; // Bit13:
174 bDummy14: BOOL; // Bit14:
175 bDummy15: BOOL; // Bit15:

```

Simatic Step 7 ohjelmalistaus, sektoriluokku 2 servon DB

4 (5)

```

176 bDummy00: BOOL; // Bit00:
177 bDummy01: BOOL; // Bit01:
178 bDummy02: BOOL; // Bit02:
179 bDummy03: BOOL; // Bit03:
180 bDummy04: BOOL; // Bit04:
181 bDummy05: BOOL; // Bit05:
182 bDummy06: BOOL; // Bit06:
183 bDummy07: BOOL; // Bit07:
184 END_STRUCT;
185
186 nEWDummy: ARRAY[1..6] OF WORD; // Event word dummy
187 nExtendEW: ARRAY[1..4] OF WORD; // Extended Event for HMI
188
189 nIW1: // I-Lock word 1
190 STRUCT
191 bDummy08: BOOL; // Bit08:
192 bDummy09: BOOL; // Bit09:
193 bDummy10: BOOL; // Bit10:
194 bDummy11: BOOL; // Bit11:
195 bDummy12: BOOL; // Bit12:
196 bDummy13: BOOL; // Bit13:
197 bDummy14: BOOL; // Bit14:
198 bDummy15: BOOL; // Bit15:
199 bDummy00: BOOL; // Bit00:
200 bDummy01: BOOL; // Bit01:
201 bDummy02: BOOL; // Bit02:
202 bDummy03: BOOL; // Bit03:
203 bDummy04: BOOL; // Bit04:
204 bDummy05: BOOL; // Bit05:
205 bDummy06: BOOL; // Bit06:
206 bDummy07: BOOL; // Bit07:
207 END_STRUCT;
208
209 nIWDummy: ARRAY[1..6] OF WORD; // I-Lock word dummy
210 nExtendIW: ARRAY[1..4] OF WORD; // Extended I-Lock for HMI
211
212 bHoming: BOOL; // Gate 2 Homing start bit
213 nNegatSwingPar: INT; // Gate 2 position setpoint min
214 nPositSwingPar: INT; // Gate 2 position setpoint max
215 bDrvToSetPntCmd: BOOL; // Gate 2 to setpoint command
216 bMechConnOpenCmd: BOOL; //
217
218 nMoveFactorPar: REAL; // Adjustment factor
219 nWaitDelayPar: TIME; //
220
221 nJogSpdNs: INT; // Gate 2 jogging forward speed
222 nHomeSpdNs: INT; // Gate 2 homing speed.
223 nWorkSpdNs: INT; // Gate 2 working speed.
224 nPosDevMarginPar: REAL;
228
229 //IN
230 bOnCmd: BOOL; // Drive On / Off command
231 bPosCmd: BOOL; // Positioning On /Off
232 bFStopCmd: BOOL; // Drive Fast stop (0 = FStop active)
233 bFwdCmd: BOOL; // Jog forward command
234 bFwdSt: BOOL; // Jog forward status
235 bBwdCmd: BOOL; // Jog backward command
236 bBwdSt: BOOL; // Jog backward status
237 nLADDR: INT; // Telegram first address from HW configuration

```

Simatic Step 7 ohjelmalistaus, sektoriluukku 2 servon DB

5 (5)

```

238 nPosNs:      DINT; // Position setpoint
239 nVelNs:      DINT; // Velocity setpoint
240 fAccLimNs:   REAL; // Acceleration override (%)
241 fDecLimNs:   REAL; // Deceleration override (%)
242 fVelLimNs:   REAL; // Velocity override (%)
243
244 //OUT
245 bRdyOn:      BOOL; // Drive ready for switch on (Enable)
246 bRdyRef:     BOOL; // Drive ready for reference (Running)
247 bToPos:      BOOL; // Positioning running
248 bOnPos:      BOOL; // Target position reached
249 bAlmSt:      BOOL; // Drive alarm status
250 bFltSt:      BOOL; // Drive fault status
251 nAlm:        INT;  // Alarm code
252 nFlt:        INT;  // Fault code
253 nPosMe:      DINT; // Position measurement
254 nVelMe:      DINT; // Velocity measurement
255
256 //Limits IN
257 bMinCmd:     BOOL; // Trig min position
258 bMaxCmd:     BOOL; // Trig max position
259 fMinPosNs:   REAL; // Min position setpoint
260 fMaxPosNs:   REAL; // Max position setpoint
261
262 //Limits IN_Out
263 fPosNs:      REAL; // Position setpoint
264 nMinRaw:     DINT; // Min position absolute value
265 nMaxRaw:     DINT; // Max position absolute value
266
267 //Limits out
268 bFwdLimSt:   BOOL; // Forward stop limit active
269 bBwdLimSt:   BOOL; // Backward stop limit active
270 fPosMe:      REAL; // Position measurement
271 nSpdNsDINT:  DINT; // Muuttuja INT_TO_DINT muunnosta varten
272
273 END_STRUCT
274
275 BEGIN
276
277 fMinPosNs := 0.0; // Min position setpoint
278 fMaxPosNs := 180.0; // Max position setpoint
279 nMinRaw := L#780000; // Min position absolute value
280 nMaxRaw := L#-50000; // Max position absolute value
281 nVelNs := L#600; // Velocity setpoint
282
283 END_DATA_BLOCK

```

Simatic Step 7 ohjelmallistaus, sektoriluukku 2 servon LogicFB 1 (7)

SIMATIC

SphinxV5 20130624\

CPU\IM151-8F PN/DP CPU\...\FB15 - <offline>

FB15 - <offline>

"Gate2LogicFB" Gate 2 Main Logic FB

Name: **Family:****Author:** MHK_RKy **Version:** 1.0**Block version:** 2**Time stamp Code:** 03/20/2014 02:55:05 PM**Interface:** 07/03/2013 06:25:59 PM**Lengths (block/logic/data):** 02878 01878 00062

Name	Data Type	Address	Initial Value	Comment
IN		0.0		
OUT		0.0		
IN_OUT		0.0		
STAT		0.0		
Edge	Array [1..16] Of Bool	0.0		
bAlm	Array [1..16] Of Bool	2.0		
bAlmAck	Array [1..16] Of Bool	4.0		
bFlt	Array [1..16] Of Bool	6.0		
bFltAck	Array [1..16] Of Bool	8.0		
bILock	Array [1..16] Of Bool	10.0		
Alarm01	eventExtensionFB	12.0		
Fault01	eventExtensionFB	74.0		
Event01	eventExtensionFB	136.0		
ILock01	eventExtensionFB	198.0		
Limit01	LimitFB	260.0		
Position	SinaPosFB	408.0		
ValueLimitsSetting	PosLimFB	496.0		
TEMP		0.0		
MarginMax	DInt	0.0		
MarginMin	DInt	4.0		
Margin	Real	8.0		
Gate2ActPosInReal	Real	12.0		
InRange	Bool	16.0		
Plug	Bool	16.1		
Accel	Word	18.0		
Decel	Word	20.0		
SpdOverride	Word	22.0		
Abs_AdjDone	Bool	24.0		
AbsHomeDone	Bool	24.1		

Block: FB15 Gate2 FB

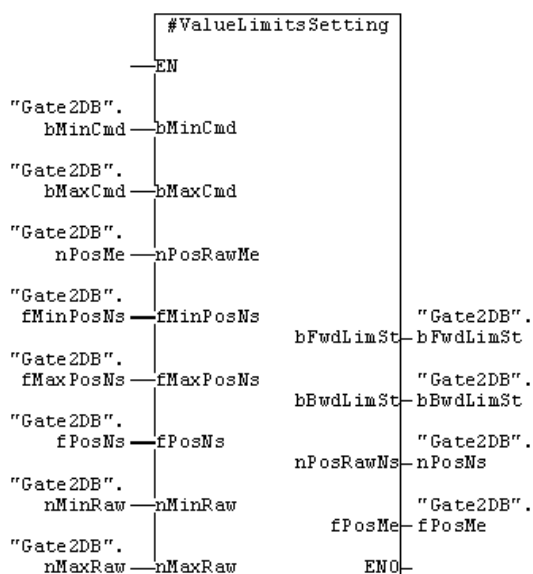
Copyright (C) Outokumpu technology, Espoo Finland

NOTES:

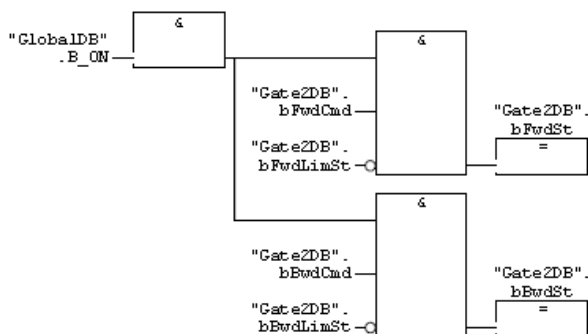
Revision History :

Simatic Step 7 ohjelmallistus, sektoriluokka 2 servon LogicFB 2 (7)

Network: 1



Network: 2 Jog forward / backward

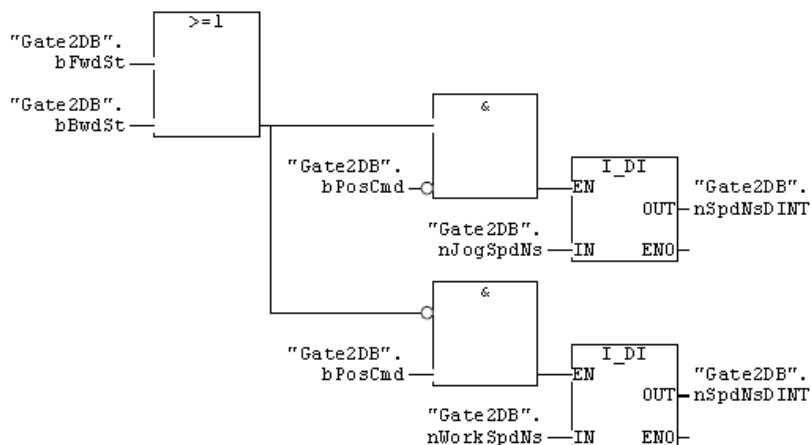


Network: 3 Speed HMI setting [deg/s] to convert into rpm

3.7.2013/RKy

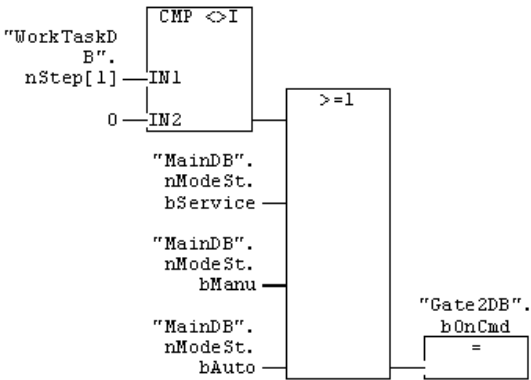
Lisäsin tämän jotta voisin syöttää HMI:ltä oman nopeuden joggausta ja paikka-ajoa varten.

HMI:ltä syötteet yksikössä [deg/s] joten siirretään syötteet muuttujalle nSpdNsDINT, rpm:ksi muuttamista varten.

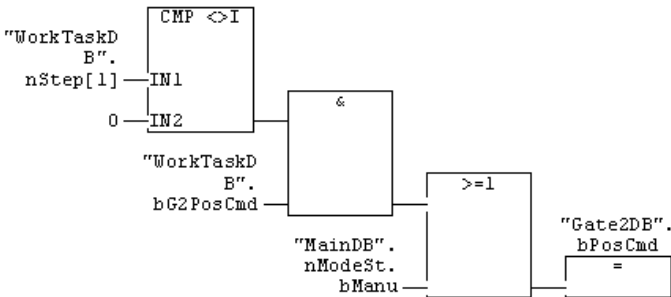


Simatic Step 7 ohjelmalistaus, sektoriluukku 2 servon LogicFB 3 (7)

Network: 4 Drive On / Off command

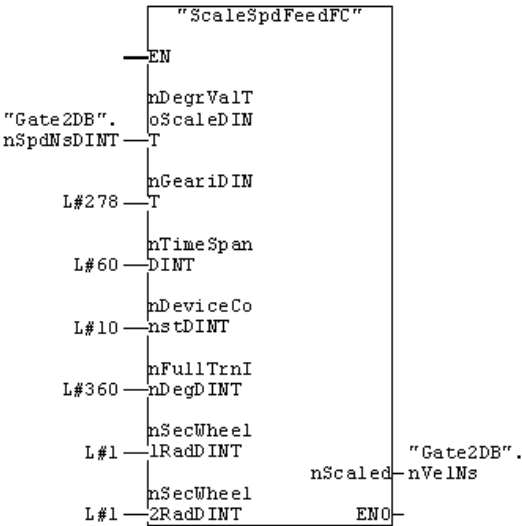


Network: 5 Positioning On /Off



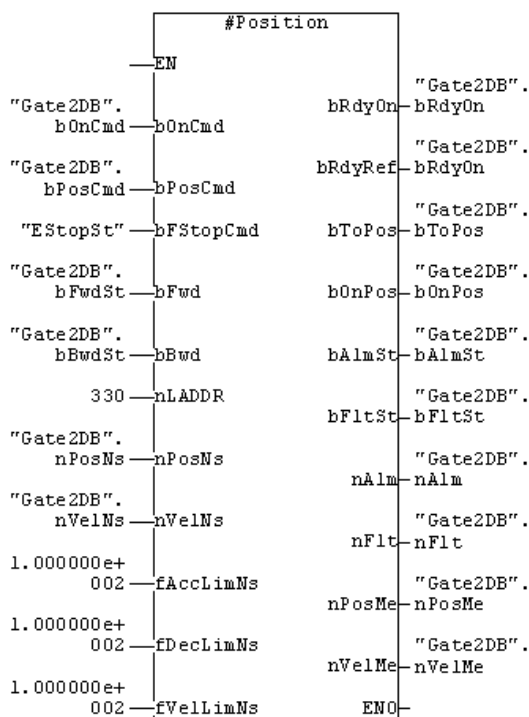
Network: 6 Scale deg/s into rpm

3.7.13/RKy
Gate 2:
- Secondary wheel 1 radius 80 [mm].
NOTE: Here used value 1 for nSecWheel1RadDINT so far.
- Secondary wheel 2 radius 160 [mm]. NOTE: Here used value 1 so far.
NOTE: Here used value 1 for nSecWheel2RadDINT so far.



Simatic Step 7 ohjelmallistus, sektoriluukku 2 servon LogicFB 4 (7)

Network: 7

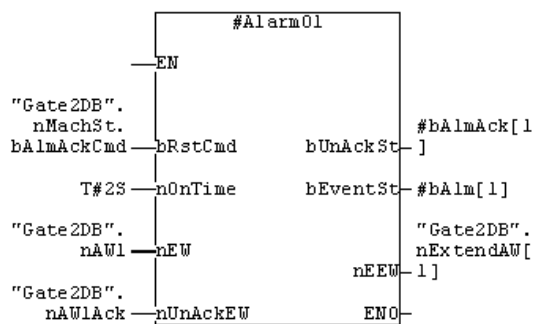


Network: 8 ALARM

Network: 9 FAULT

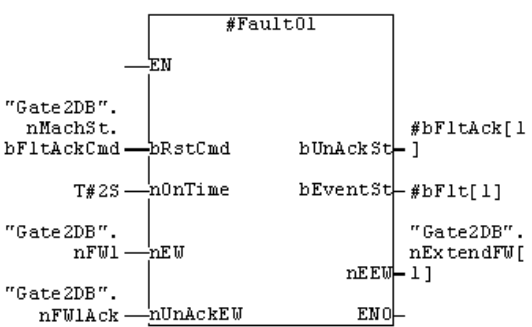
Network: 10 I-LOCK

Network: 11 Alarm extension module

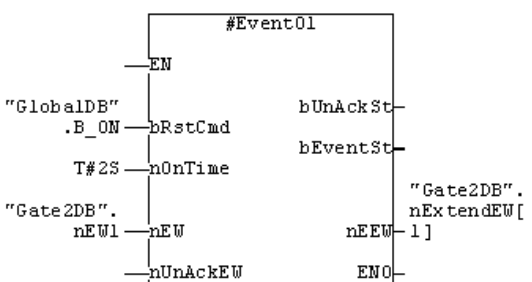


Simatic Step 7 ohjelmallistus, sektoriluokku 2 servon LogicFB 5 (7)

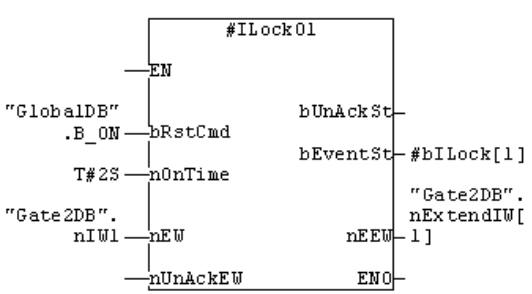
Network: 12 Fault extension module



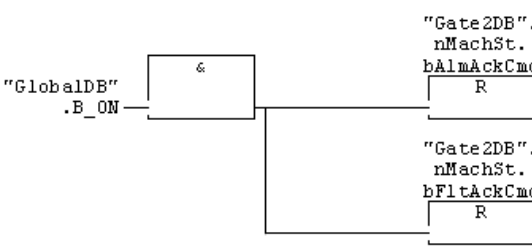
Network: 13 Event extension module



Network: 14 I-Lock extension module

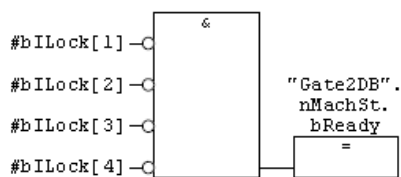


Network: 15 Reset of alarm/fault acknowledgement command

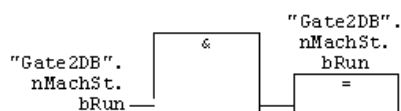


Simatic Step 7 ohjelmallistaus, sektoriluukku 2 servon LogicFB 6 (7)

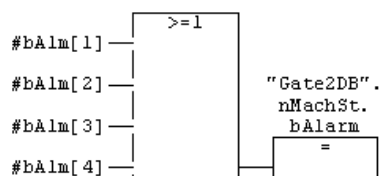
Network: 16 Status for HMI : READY TO START



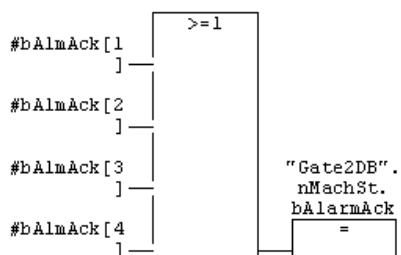
Network: 17 Status for HMI : RUN



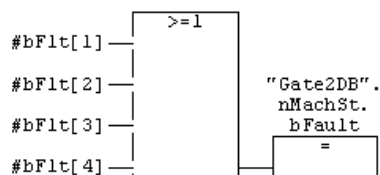
Network: 18 Alarm active



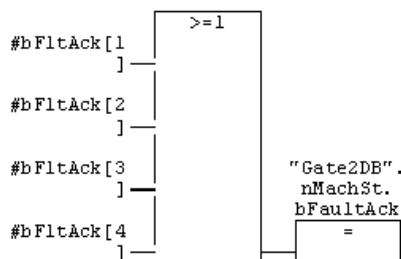
Network: 19 Unacknowledged alarm active



Network: 20 Fault active

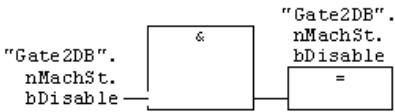


Network: 21 Unacknowledged fault active

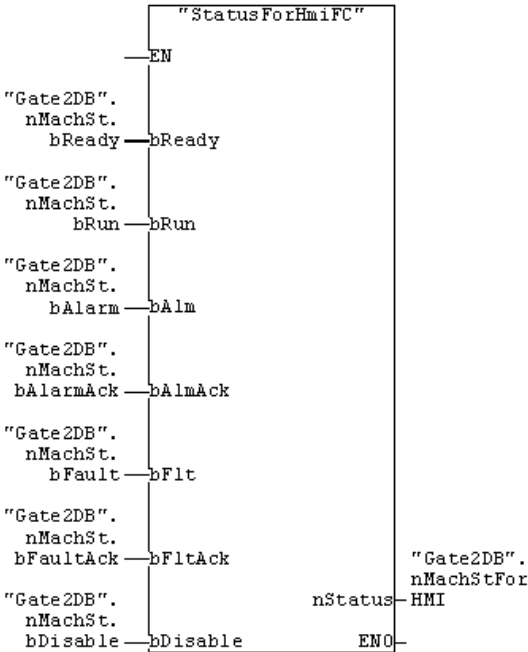


Simatic Step 7 ohjelmallistus, sektoriluukku 2 servon LogicFB 7 (7)

Network: 22 Disable



Network: 23 Machine status for HMI symbol



Network: 24 Default handling of the ENO output

This network has to be the last one of the function

SET
SAVE

Simatic Step 7 ohjelmalistaus, paikkalaskurin skaalaus FB

1 (2)

SphinxV5 20130624\CPU\IM151-8F PN/DP CPU\...\FB Position Limits

```

1  (* Copyright (C) Outokumpu technology, Espoo Finland
2  =====
3  NOTES : Position limits
4  Revision History :
5  =====*)
6
7  FUNCTION_BLOCK "PosLimFB"
8  TITLE = 'Position limits FB'
9  VERSION : '1.1'
10 AUTHOR : MHK
11
12 VAR_INPUT
13 bMinCmd:    BOOL; // Trig min position
14 bMaxCmd:    BOOL; // Trig max position
15 nPosRawMe:  DINT; // Position measurement
16 fMinPosNs:  REAL; // Min position setpoint
17 fMaxPosNs:  REAL; // Max position setpoint
18 END_VAR
19
20 VAR_IN_OUT
21 fPosNs:     REAL; // Position setpoint
22 nMinRaw:    DINT; // Min position absolute value
23 nMaxRaw:    DINT; // Max position absolute value
24 END_VAR
25
26 VAR_OUTPUT
27 bFwdLimSt:  BOOL; // Forward stop limit active
28 bBwdLimSt:  BOOL; // Backward stop limit active
29 nPosRawNs:  DINT; // Position setpoint
30 fPosMe:     REAL; // Position measurement
31 END_VAR
32
33 VAR
34
35 END_VAR
36
37 VAR_TEMP
38
39 END_VAR
40
41 BEGIN
42
43 //*****
44 // Min / Max absolute values
45 //*****
46
47 IF bMinCmd THEN
48   nMinRaw := nPosRawMe;
49 ELSIF bMaxCmd THEN
50   nMaxRaw := nPosRawMe;
51 END_IF;
52
53 //*****
54 // Measurement
55 //*****
56
57 IF nMaxRaw <> nMinRaw THEN

```

Simatic Step 7 ohjelmallistaus, paikkalaskurin skaalaus FB

2 (2)

```

58 fPosMe := ( fMaxPosNs - fMinPosNs ) / ( nMaxRaw - nMinRaw )
          * ( nPosRawMe - nMinRaw ) + fMinPosNs;
59 END_IF;
60
61 //*****
62 // Setpoint
63 //*****
64
65 IF fMaxPosNs <> fMinPosNs THEN
66 nPosRawNs := REAL_TO_DINT( ( nMaxRaw - nMinRaw ) / ( fMaxPosNs
          - fMinPosNs ) * ( fPosNs - fMinPosNs ) + nMinRaw );
67 END_IF;
68
69 //*****
70 // Setpoint limitation
71 //*****
72
73 IF fPosNs <= fMinPosNs THEN
74 fPosNs := fMinPosNs;
75 ELSIF fPosNs >= fMaxPosNs THEN
76 fPosNs := fMaxPosNs;
77 END_IF;
78
79 IF nMinRaw <= nMaxRaw THEN
80
81 IF nPosRawMe <= nMinRaw THEN
82 bFwdLimSt := FALSE;
83 bBwdLimSt := TRUE;
84 ELSIF nPosRawMe >= nMaxRaw THEN
85 bFwdLimSt := TRUE;
86 bBwdLimSt := FALSE;
87 ELSE
88 bFwdLimSt := FALSE;
89 bBwdLimSt := FALSE;
90 END_IF;
91
92 ELSE
93
94 IF nPosRawMe <= nMaxRaw THEN
95 bFwdLimSt := FALSE;
96 bBwdLimSt := TRUE;
97 ELSIF nPosRawMe >= nMinRaw THEN
98 bFwdLimSt := TRUE;
99 bBwdLimSt := FALSE;
100 ELSE
101 bFwdLimSt := FALSE;
102 bBwdLimSt := FALSE;
103 END_IF;
104
105 END_IF;
106 //*****
107
108 END_FUNCTION_BLOCK

```

Simatic Step 7 ohjelmallistaus, työsekvenssi FB Seq01

1 (11)

SphinxV5 20130624\CPU\IM151-8F PN/DP CPU\...\FB Seq01

```

1  (* Copyright (C) Outokumpu technology, Espoo Finland
2  =====
3  NOTES :
4  Revision History :
5  =====*)
6
7  FUNCTION_BLOCK "SEQ01FB"
8  TITLE = 'Sequence FB'
9  VERSION : '1.0'
10 AUTHOR : MHK_ RKy
11
12 VAR_INPUT
13
14 END_VAR
15
16 VAR_IN_OUT
17   bOn:          BOOL; // Sequence On / Off
18 END_VAR
19
20 VAR_OUTPUT
21   bDone:        BOOL; // Sequence done
22   bFault:       BOOL; // Sequence faulty
23   nStep:        INT;  // Current sequence step number
24 END_VAR
25
26 VAR
27   timStep:      TON; // Step timer
28 END_VAR
29
30 VAR_TEMP
31   bNext:        BOOL; // Next step
32   nMecMaxSpd:   REAL; // Distributors Mechanical maximun speed
33 END_VAR
34
35 BEGIN
36
37 //2.7.13/RKy
38 //Muunnos: Porttien nPositSwingPar REAL:ksi ja arvoksi nG1PosNs[2],
39 //nG2PosNs[2]:lle
40 "WorkTaskDB".nG1PosNs[2] := INT_TO_REAL("Gate1DB".nPositSwingPar);
41 "WorkTaskDB".nG2PosNs[2] := INT_TO_REAL("Gate2DB".nPositSwingPar);
42 // Distributorin nPosSwingPar REAL:ksi ja "WorkTaskDB".nD12PosNs[4]
43 // ja [5]:lle. Target ja turn point samaksi
44 // Distributorin nNegSwingPar REAL:ksi ja arvoiksi
45 // "WorkTaskDB".nD12PosNs[6] ja [7]:lle. Target ja turn point samaksi
46 "WorkTaskDB".nD12PosNs[4] :=INT_TO_REAL("WorkTaskDB".nPosSwingPar);
47 "WorkTaskDB".nD12PosNs[5] := "WorkTaskDB".nD12PosNs[4]; //FWD tp
48 "WorkTaskDB".nD12PosNs[6] :=INT_TO_REAL("WorkTaskDB".nNegSwingPar);
49 "WorkTaskDB".nD12PosNs[7] := "WorkTaskDB".nD12PosNs[6]; //BWD tp
50
51 //3.7.13/RKy
52 // Jaksonaikaa oltava enemmän kuin 0 jottei jaeta 0:lla
53 IF "WorkTaskDB".nCycleTime < 1 THEN
54   "WorkTaskDB".nCycleTime := 1;
55 END_IF;

```


Simatic Step 7 ohjelmallistaus, työsekvenssi FB Seq01

2 (11)

```

53 nMecMaxSpd := ((3000.0 / 60.0) / 278.0) * 360.0; // Jos skaalaukset
    tehdään distributorilevyn asennon mukaan, pitää tähän ottaa mukaan
    välittävien pyörien säteiden suhde 80[mm] / 540[mm]
54
55 "WorkTaskDB".fWrkSpd := ("WorkTaskDB".nD12PosNs[4]-
    "WorkTaskDB".nD12PosNs[6]) / INT_TO_REAL("WorkTaskDB".nCycleTime);
56
57 // Jos yritetään nopeampaa kuin pystytään niin otetaan maksimi
    käyttöön. Samalla asetetaan jaksonaika vastaamaan maksimia. Jakson
    ajalla tarkoitetaan ajoa ääriasennosta toiseen
58 IF "WorkTaskDB".fWrkSpd > nMecMaxSpd THEN
59     "WorkTaskDB".fWrkSpd := nMecMaxSpd;
60     "WorkTaskDB".nCycleTime :=
        REAL_TO_INT(("WorkTaskDB".nD12PosNs[4]-
            "WorkTaskDB".nD12PosNs[6])/"WorkTaskDB".fWrkSpd);
61 END_IF;
62 "Distributor1DB".nWorkSpdNs := REAL_TO_INT("WorkTaskDB".fWrkSpd);
63 "WorkTaskDB".nD12VelNs[4] := REAL_TO_DINT((278.0 / 6.0)
    * "WorkTaskDB".fWrkSpd * 10);
64 "WorkTaskDB".nD12VelNs[5] := "WorkTaskDB".nD12VelNs[4];
65 "WorkTaskDB".nD12VelNs[6] := "WorkTaskDB".nD12VelNs[4];
66 "WorkTaskDB".nD12VelNs[7] := "WorkTaskDB".nD12VelNs[4];
67 "WorkTaskDB".nD12VelNs[9] := "WorkTaskDB".nD12VelNs[4];
68 "WorkTaskDB".nD12VelNs[12] := "WorkTaskDB".nD12VelNs[4];
69 "WorkTaskDB".nD12VelNs[13] := "WorkTaskDB".nD12VelNs[4];
70
71 //*****
72 //(STEP 00) Stopped
73 //*****
74 IF ( NOT bOn AND nStep = 15 ) OR NOT "EStopSt" THEN
75     //*****
76     // PULSE CONTROL
77     //*****
78     nStep := 0;
79     "WorkTaskDB".nStep[0]:=nStep; // 04.07.13/RKy
80     timStep(IN:=FALSE, PT:=T#1h);
81     //*****
82 END_IF;
83 IF nStep = 0 THEN
84     //*****
85     // CONTINUOUS CONTROL
86     //*****
87     bDone := FALSE;
88
89     "WorkTaskDB".bD12PosCmd := FALSE;
90     "WorkTaskDB".bG1PosCmd := FALSE;
91     "WorkTaskDB".bG2PosCmd := FALSE;
92
93     //*****
94 END_IF;
95
96

```

Simatic Step 7 ohjelmallistaus, työsekvenssi FB Seq01

3 (11)

```

97 //*****
98 //(STEP 01) Start
99 //*****
100 // Interlock to enter on this step
101 bNext := "GlobalDB".B_ON;
102
103 IF bOn AND bNext AND nStep = 0 THEN
104 //*****
105 // PULSE CONTROL
106 //*****
107 nStep := 1;
108 "WorkTaskDB".nStep[0]:=nStep; // 04.07.13/RKy
109 timStep(IN:=FALSE, PT:=T#1h);
110 //*****
111 END_IF;
112 IF nStep = 1 THEN
113 //*****
114 // CONTINUOUS CONTROL
115 //*****
116 timStep(IN:=TRUE, PT:=T#1h);
117 //*****
118 END_IF;
119
120
121 //*****
122 //(STEP 02) Open Gates
123 //*****
124 // Interlock to enter on this step
125 bNext := timStep.ET >= T#500ms;
126
127 IF bNext AND nStep = 1 THEN
128 //*****
129 // PULSE CONTROL
130 //*****
131 nStep := 2;
132 "WorkTaskDB".nStep[0]:=nStep; // 04.07.13/RKy
133 timStep(IN:=FALSE, PT:=T#1h);
134 //*****
135 END_IF;
136 IF nStep = 2 THEN
137 //*****
138 // CONTINUOUS CONTROL
139 //*****
140 timStep(IN:=TRUE, PT:=T#1h);
141
142 "WorkTaskDB".bG1PosCmd := TRUE;
143 "WorkTaskDB".bG2PosCmd := TRUE;
144
145 "Gate1DB".fPosNs := "WorkTaskDB".nG1PosNs[2];
146 "Gate1DB".nVelNs := "WorkTaskDB".nG1VelNs[2];
147
148 "Gate2DB".fPosNs := "WorkTaskDB".nG2PosNs[2];
149 "Gate2DB".nVelNs := "WorkTaskDB".nG2VelNs[2];
150
151 //*****
152 END_IF;

```

Simatic Step 7 ohjelmallistaus, työsekvenssi FB Seq01

4 (11)

```

153
154
155 //*****
156 //(STEP 03) Gates Ready
157 //*****
158 // Interlock to enter on this step
159 bNext := timStep.ET >= T#500ms AND
160 ABS("Gate1DB".fPosNs - "Gate1DB".fPosMe) <= 0.05 AND
161 ABS("Gate2DB".fPosNs - "Gate2DB".fPosMe) <= 0.05;
162
163 IF bNext AND nStep = 2 THEN
164 //*****
165 // PULSE CONTROL
166 //*****
167 nStep := 3;
168 "WorkTaskDB".nStep[0] := nStep; // 04.07.13/RKy
169 timStep(IN:=FALSE, PT:=T#1h);
170 //*****
171 END_IF;
172 IF nStep = 3 THEN
173 //*****
174 // CONTINUOUS CONTROL
175 //*****
176 timStep(IN:=TRUE, PT:=T#1h);
177
178 "WorkTaskDB".bG1PosCmd := FALSE;
179 "WorkTaskDB".bG2PosCmd := FALSE;
180
181 //*****
182 END_IF;
183
184
185 //*****
186 //(STEP 04) Distributor Forward
187 //*****
188 // Interlock to enter on this step
189 IF nStep = 8 THEN
190 bNext := timStep.ET >= T#500ms;
191 ELSE
192 bNext := timStep.ET >= T#500ms;
193 END_IF;
194
195 IF bNext AND ( nStep = 3 OR nStep = 8 ) THEN
196 //*****
197 // PULSE CONTROL
198 //*****
199 nStep := 4;
200 "WorkTaskDB".nStep[0] := nStep; // 04.07.13/RKy
201 timStep(IN:=FALSE, PT:=T#1h);
202 //*****
203 END_IF;

```

Simatic Step 7 ohjelmallistaus, työsekvenssi FB Seq01

5 (11)

```

204 IF nStep = 4 THEN
205 //*****
206 // CONTINUOUS CONTROL
207 //*****
208 timStep(IN:=TRUE, PT:=T#1h);
209
210 "WorkTaskDB".bD12PosCmd := TRUE;
211 "Distributor1DB".fPosNs := "WorkTaskDB".nD12PosNs[4];
212 "Distributor1DB".nVelNs := "WorkTaskDB".nD12VelNs[4];
213 "Distributor2DB".fPosNs := "WorkTaskDB".nD12PosNs[4];
214 "Distributor2DB".nVelNs := "WorkTaskDB".nD12VelNs[4];
215
216 //*****
217 END_IF;
218
219
220 //*****
221 //(STEP 05) Distributor Forward turn point
222 //*****
223 // Interlock to enter on this step
224 bNext := timStep.ET >= T#500ms AND
225 ABS("Distributor1DB".fPosNs - "Distributor1DB".fPosMe) <= 0.05 AND
226 ABS("Distributor2DB".fPosNs - "Distributor2DB".fPosMe) <= 0.05;
227
228 IF bNext AND nStep = 4 THEN
229 //*****
230 // PULSE CONTROL
231 //*****
232 nStep := 5;
233 "WorkTaskDB".nStep[0]:=nStep; // 04.07.13/RKy
234 timStep(IN:=FALSE, PT:=T#1h);
235 //*****
236 END_IF;
237 IF nStep = 5 THEN
238 //*****
239 // CONTINUOUS CONTROL
240 //*****
241 timStep(IN:=TRUE, PT:=T#1h);
242 "WorkTaskDB".bD12PosCmd := TRUE;
243 "Distributor1DB".fPosNs := "WorkTaskDB".nD12PosNs[5];
244 "Distributor1DB".fVelNs := "WorkTaskDB".nD12VelNs[5];
245 "Distributor2DB".fPosNs := "WorkTaskDB".nD12PosNs[5];
246 "Distributor2DB".nVelNs := "WorkTaskDB".nD12VelNs[5];
247
248 //*****
249 END_IF;

```

Simatic Step 7 ohjelmallistaus, työsekvenssi FB Seq01

6 (11)

```

250
251
252 //*****
253 //(STEP 06) Distributor Backward
254 //*****
255 // Interlock to enter on this step
256 // 2.7.13/RKy Odotusaika "WorkTaskDB".nWait T#500ms:n tilalle
257 // HMI:stä syöttö sekunteina. Muunnos ms:ksi kertomalla 1000:lla
258
259 bNext := timStep.ET >=
    DINT_TO_TIME(INT_TO_DINT(1000 * "WorkTaskDB".nWait)); //T#500ms;
260
261 IF bNext AND nStep = 5 THEN
262     //*****
263     // PULSE CONTROL
264     //*****
265     nStep := 6;
266     "WorkTaskDB".nStep[0]:=nStep; // 04.07.13/RKy
267     timStep(IN:=FALSE, PT:=T#1h);
268     //*****
269 END_IF;
270 IF nStep = 6 THEN
271     //*****
272     // CONTINUOUS CONTROL
273     //*****
274     timStep(IN:=TRUE, PT:=T#1h);
275
276     "WorkTaskDB".bD12PosCmd := TRUE;
277     "Distributor1DB".fPosNs := "WorkTaskDB".nD12PosNs[6];
278     "Distributor1DB".nVelNs := "WorkTaskDB".nD12VelNs[6];
279     "Distributor2DB".fPosNs := "WorkTaskDB".nD12PosNs[6];
280     "Distributor2DB".nVelNs := "WorkTaskDB".nD12VelNs[6];
281
282 //*****
283 END_IF;
284
285
286 //*****
287 //(STEP 07) Distributor Backward turn point
288 //*****
289 // Interlock to enter on this step
290 bNext := timStep.ET >=
291 ABS("Distributor1DB".fPosNs - "Distributor1DB".fPosMe) <= 0.05 AND
292 ABS("Distributor2DB".fPosNs - "Distributor2DB".fPosMe) <= 0.05;
293
294 IF bNext AND nStep = 6 THEN
295     //*****
296     // PULSE CONTROL
297     //*****
298     nStep := 7;
299     "WorkTaskDB".nStep[0]:=nStep; // 04.07.13/RKy
300     timStep(IN:=FALSE, PT:=T#1h);
301     //*****
302 END_IF;

```

Simatic Step 7 ohjelmallistaus, työsekvenssi FB Seq01

7 (11)

```

303 IF nStep = 7 THEN
304 //*****
305 // CONTINUOUS CONTROL
306 //*****
307 timStep(IN:=TRUE, PT:=T#1h);
308
309 "WorkTaskDB".bD12PosCmd := TRUE;
310 "Distributor1DB".fPosNs := "WorkTaskDB".nD12PosNs[7];
311 "Distributor1DB".nVelNs := "WorkTaskDB".nD12VelNs[7];
312 "Distributor2DB".fPosNs := "WorkTaskDB".nD12PosNs[7];
313 "Distributor2DB".nVelNs := "WorkTaskDB".nD12VelNs[7];
314
315 //*****
316 END_IF;
317
318
319 //*****
320 //(STEP 08) Cycle Done
321 //*****
322 // Interlock to enter on this step
323 // 2.7.13/RKy Odotusaika "WorkTaskDB".nWait T#500ms:n tilalle
324 // HMI:stä syöttö sekunteina. Muunnos ms:ksi kertomalla 1000:lla
325
326 bNext := timStep.ET >= DINT_TO_TIME(INT_TO_DINT(1000 *
                                     "WorkTaskDB".nWait)); //T#500ms;
327
328 IF bNext AND nStep = 7 THEN
329 //*****
330 // PULSE CONTROL
331 //*****
332 nStep := 8;
333 "WorkTaskDB".nStep[0]:=nStep; // 04.07.13/RKy
334 timStep(IN:=FALSE, PT:=T#1h);
335 //*****
336 END_IF;
337 IF nStep = 8 THEN
338 //*****
339 // CONTINUOUS CONTROL
340 //*****
341 timStep(IN:=TRUE, PT:=T#1h);
342 //*****
343 END_IF;

```

Simatic Step 7 ohjelmalistaus, työsekvenssi FB Seq01

8 (11)

```

344
345
346 //*****
347 //(STEP 09) Distributor to Gate 1 closing position
348 //*****
349 // Interlock to enter on this step
350 bNext := NOT bOn;
351
352 IF bNext AND ( nStep >= 2 AND nStep <= 8 ) THEN
353 //*****
354 // PULSE CONTROL
355 //*****
356 nStep := 9;
357 "WorkTaskDB".nStep[0]:=nStep; // 04.07.13/RKy
358 timStep(IN:=FALSE, PT:=T#1h);
359 //*****
360 END_IF;
361 IF nStep = 9 THEN
362 //*****
363 // CONTINUOUS CONTROL
364 //*****
365 timStep(IN:=TRUE, PT:=T#1h);
366
367 "WorkTaskDB".bD12PosCmd := TRUE;
368 "Distributor1DB".fPosNs := "WorkTaskDB".nD12PosNs[9];
369 "Distributor1DB".nVelNs := "WorkTaskDB".nD12VelNs[9];
370 "Distributor2DB".fPosNs := "WorkTaskDB".nD12PosNs[9];
371 "Distributor2DB".nVelNs := "WorkTaskDB".nD12VelNs[9];
372
373 //*****
374 END_IF;
375
376
377 //*****
378 //(STEP 10) Close Gate 1
379 //*****
380 // Interlock to enter on this step
381 bNext := timStep.ET >= T#500ms AND
382 ABS("Distributor1DB".fPosNs - "Distributor1DB".fPosMe) <= 0.05 AND
383 ABS("Distributor2DB".fPosNs - "Distributor2DB".fPosMe) <= 0.05;
384
385 IF bNext AND nStep = 9 THEN
386 //*****
387 // PULSE CONTROL
388 //*****
389 nStep := 10;
390 "WorkTaskDB".nStep[0]:=nStep; // 04.07.13/RKy
391 timStep(IN:=FALSE, PT:=T#1h);
392 //*****
393 IF nStep = 10 THEN
394 //*****
395 // CONTINUOUS CONTROL
396 //*****
397 timStep(IN:=TRUE, PT:=T#1h);
398
399
400 "WorkTaskDB".bG1PosCmd := TRUE;
401 "Gate1DB".fPosNs := "WorkTaskDB".nG1PosNs[11];
402 "Gate1DB".nVelNs := "WorkTaskDB".nG1VelNs[11];
403 //*****
404 END_IF;

```

Simatic Step 7 ohjelmallistaus, työsekvenssi FB Seq01

9 (11)

```

406
407
408 //*****
409 //(STEP 11) Distributor to Gate 2 closing position
410 //*****
411 // Interlock to enter on this step
412 bNext := timStep.ET >= T#500ms AND
413       ABS("Gate1DB".fPosNs - "Gate1DB".fPosMe) <= 0.05;
414
415 IF bNext AND nStep = 10 THEN
416 //*****
417 // PULSE CONTROL
418 //*****
419 nStep := 11;
420 "WorkTaskDB".nStep[0] := nStep; // 04.07.13/RKy
421 timStep(IN:=FALSE, PT:=T#1h);
422 //*****
423 END_IF;
424 IF nStep = 11 THEN
425 //*****
426 // CONTINUOUS CONTROL
427 //*****
428 timStep(IN:=TRUE, PT:=T#1h);
429
430 "WorkTaskDB".bD12PosCmd := TRUE;
431 "Distributor1DB".fPosNs := "WorkTaskDB".nD12PosNs[12];
432 "Distributor1DB".nVelNs := "WorkTaskDB".nD12VelNs[12];
433 "Distributor2DB".fPosNs := "WorkTaskDB".nD12PosNs[12];
434 "Distributor2DB".nVelNs := "WorkTaskDB".nD12VelNs[12];
435
436 //*****
437 END_IF;
438

```


Simatic Step 7 ohjelmallistaus, työsekvenssi FB Seq01

10 (11)

```

439
440 //*****
441 //(STEP 12) Close Gate 2
442 //*****
443 // Interlock to enter on this step
444 bNext := timStep.ET >= T#500ms AND
445 ABS("Distributor1DB".fPosNs - "Distributor1DB".fPosMe) <= 0.05 AND
446 ABS("Distributor2DB".fPosNs - "Distributor2DB".fPosMe) <= 0.05;
447
448 IF bNext AND nStep = 11 THEN
449 //*****
450 // PULSE CONTROL
451 //*****
452 nStep := 12;
453 "WorkTaskDB".nStep[0]:=nStep; // 04.07.13/RKy
454 timStep(IN:=FALSE, PT:=T#1h);
455 //*****
456 END_IF;
457 IF nStep = 12 THEN
458 //*****
459 // CONTINUOUS CONTROL
460 //*****
461 timStep(IN:=TRUE, PT:=T#1h);
462
463 "WorkTaskDB".bG2PosCmd := TRUE;
464 "Gate2DB".fPosNs := "WorkTaskDB".nG2PosNs[12];
465 "Gate2DB".nVelNs := "WorkTaskDB".nG2VelNs[12];
466
467 //*****
468 END_IF;
469
470
471 //*****
472 //(STEP 13) Distributor to middle position
473 //*****
474 // Interlock to enter on this step
475 bNext := timStep.ET >= T#500ms AND
476 ABS("Gate2DB".fPosNs - "Gate2DB".fPosMe) <= 0.05;
477
478 IF bNext AND nStep = 12 THEN
479 //*****
480 // PULSE CONTROL
481 //*****
482 nStep := 13;
483 "WorkTaskDB".nStep[0]:=nStep; // 04.07.13/RKy
484 timStep(IN:=FALSE, PT:=T#1h);
485 //*****
486 END_IF;
487 IF nStep = 13 THEN
488 //*****
489 // CONTINUOUS CONTROL
490 //*****
491 timStep(IN:=TRUE, PT:=T#1h);
492 "WorkTaskDB".bD12PosCmd := TRUE;
493 "Distributor1DB".fPosNs := "WorkTaskDB".nD12PosNs[13];
494 "Distributor1DB".nVelNs := "WorkTaskDB".nD12VelNs[13];
495 "Distributor2DB".fPosNs := "WorkTaskDB".nD12PosNs[13];
496 "Distributor2DB".nVelNs := "WorkTaskDB".nD12VelNs[13];
497
498 //*****
499 END_IF;
500

```

Simatic Step 7 ohjelmallistaus, työsekvenssi FB Seq01

11 (11)

```

501
502
503 //*****
504 //(STEP 14) Wait
505 //*****
506 // Interlock to enter on this step
507 bNext := timStep.ET >= T#500ms AND
508 ABS("Distributor1DB".fPosNs - "Distributor1DB".fPosMe) <= 0.05 AND
509 ABS("Distributor2DB".fPosNs - "Distributor2DB".fPosMe) <= 0.05;
510
511 IF bNext AND nStep = 13 THEN
512 //*****
513 // PULSE CONTROL
514 //*****
515 nStep := 14;
516 "WorkTaskDB".nStep[0] := nStep; // 04.07.13/RKy
517 timStep(IN:=FALSE, PT:=T#1h);
518 //*****
519 END_IF;
520 IF nStep = 14 THEN
521 //*****
522 // CONTINUOUS CONTROL
523 //*****
524 timStep(IN:=TRUE, PT:=T#1h);
525
526 "WorkTaskDB".bD12PosCmd := FALSE;
527 "WorkTaskDB".bG1PosCmd := FALSE;
528 "WorkTaskDB".bG2PosCmd := FALSE;
529
530 //*****
531 END_IF;
532
533
534 //*****
535 //(STEP 15) Done
536 //*****
537 // Interlock to enter on this step
538 bNext := timStep.ET >= T#500ms;
539
540 IF bNext AND nStep = 14 THEN
541 //*****
542 // PULSE CONTROL
543 //*****
544 nStep := 15;
545 "WorkTaskDB".nStep[0] := nStep; // 04.07.13/RKy
546 timStep(IN:=FALSE, PT:=T#1h);
547 //*****
548 END_IF;
549 IF nStep = 15 THEN
550 //*****
551 // CONTINUOUS CONTROL
552 //*****
553 timStep(IN:=TRUE, PT:=T#1h);
554 bDone := TRUE;
555 //*****
556 END_IF;
557
558 END_FUNCTION_BLOCK

```

Simatic Step 7 ohjelmalistaus, purkajan servojen paikkaero

1 (1)

SphinxV5 20130624\CPU\IM151-8F PN/DP CPU\...\FC Position
Deviation

```

1  (* Copyright (C) Outokumpu technology, Espoo Finland
2  =====
3  NOTES : Position deviation between two encoder values
4  Revision History :
5  =====*)
6
7  FUNCTION "PosDeviationFC" : VOID
8  TITLE = 'Position Deviation FC'
9  VERSION : '1.0'
10 AUTHOR : RKy
11
12 VAR_INPUT
13 fPos1:      REAL; // First encoder to compare
14 fPos2:      REAL; // Second encoder to compare
15 fDevMargin: REAL; // Allowed deviation between first and two
16 END_VAR
17
18 VAR_IN_OUT
19
20 END_VAR
21
22 VAR_OUTPUT
23 bDevOutOfRange: BOOL; // 1 = Out range, 0 = In range
24 fPosDev:      REAL; // Deviation between first and two
25 END_VAR
26
27 VAR_TEMP
28
29 END_VAR
30
31 BEGIN
32
33 //*****
34 // Program code
35 //*****
36
37 fPosDev := ABS(fPos1 - fPos2);
38 bDevOutOfRange := fPosDev > fDevMargin;
39
40 //*****
41
42 END_FUNCTION

```

Simatic Step 7 ohjelmalistaus, työkierron DB

1 (7)

SphinxV5 20130624\CPU\IM151-8F PN/DP CPU\...\DB WorkTask

```

1  (* Copyright (C) Outokumpu tecnologia, Espoo Finland
2  =====
3  NOTES :
4  Revision History :
5  - 14062013/RKy: Variable PosDevOutMargin added
6  =====*)
7
8  DATA_BLOCK "WorkTaskDB"
9  TITLE = 'Work Task DB'
10 VERSION : 1.0
11 AUTHOR : RKy
12
13 STRUCT
14
15 nModeCmd: // Main mode commands
16 STRUCT
17 bDummy08: BOOL; // Bit08:
18 bDummy09: BOOL; // Bit09:
19 bDummy10: BOOL; // Bit10:
20 bDummy11: BOOL; // Bit11:
21 bDummy12: BOOL; // Bit12:
22 bDummy13: BOOL; // Bit13:
23 bDummy14: BOOL; // Bit14:
24 bDummy15: BOOL; // Bit15:
25 bEStop:   BOOL; // Bit00: EMERGENCY STOP
26 bStop:    BOOL; // Bit01: STOP
27 bService: BOOL; // Bit02: SERVICE
28 bAux:     BOOL; // Bit03: AUX
29 bManu:    BOOL; // Bit04: MANUAL
30 bSemi:    BOOL; // Bit05: SEMI-AUTOMATIC
31 bAuto:    BOOL; // Bit06: AUTOMATIC
32 bDummy07: BOOL; // Bit07:
33 END_STRUCT;
34
35 nModeSt: // Main mode status
36 STRUCT
37 bMoveOK:  BOOL; // Bit08: Movement allowed
38 bDummy09: BOOL; // Bit09:
39 bDummy10: BOOL; // Bit10:
40 bDummy11: BOOL; // Bit11:
41 bDummy12: BOOL; // Bit12:
42 bDummy13: BOOL; // Bit13:
43 bDummy14: BOOL; // Bit14:
44 bDummy15: BOOL; // Bit15:
45 bEStop:   BOOL; // Bit00: EMERGENCY STOP
46 bStopped: BOOL; // Bit01: STOPPED
47 bService: BOOL; // Bit02: SERVICE
48 bAux:     BOOL; // Bit03: AUX
49 bManu:    BOOL; // Bit04: MANUAL
50 bSemi:    BOOL; // Bit05: SEMI-AUTOMATIC
51 bAuto:    BOOL; // Bit06: AUTOMATIC
52 bDummy07: BOOL; // Bit07:
53 END_STRUCT;
54
55 nModeStForHMI: INT; // Main mode for HMI
56
57 nMachSt: // Machine status

```

Simatic Step 7 ohjelmalistaus, työkierron DB

2 (7)

```

58 STRUCT
59 bAlmAckCmd: BOOL; // Alarm acknowledgement command
60 bFltAckCmd: BOOL; // Fault acknowledgement command
61 bDummy10: BOOL;
62 bDummy11: BOOL;
63 bDummy12: BOOL;
64 bDummy13: BOOL;
65 bDummy14: BOOL;
66 bDummy15: BOOL;
67 bReady: BOOL; // Ready
68 bRun: BOOL; // Run
69 bAlarm: BOOL; // Alarm
70 bAlarmAck: BOOL; // Unacknowledged alarm
71 bFault: BOOL; // Fault
72 bFaultAck: BOOL; // Unacknowledged fault
73 bDisable: BOOL; // Disable
74 bDummy05: BOOL;
75 bDummy06: BOOL;
76 bDummy07: BOOL;
77 END_STRUCT;
78
79 nMachStForHMI: INT; // Machine status for HMI
80
81 nAW1: // Alarm word 1
82 STRUCT
83 bDummy08: BOOL; // Bit08:
84 bDummy09: BOOL; // Bit09:
85 bDummy10: BOOL; // Bit10:
86 bDummy11: BOOL; // Bit11:
87 bDummy12: BOOL; // Bit12:
88 bDummy13: BOOL; // Bit13:
89 bDummy14: BOOL; // Bit14:
90 bDummy15: BOOL; // Bit15:
91 bPosDevOutMargin: BOOL; // Bit00: Distributor servo's position
deviation out of margin
92 bDummy01: BOOL; // Bit01:
93 bDummy02: BOOL; // Bit02:
94 bDummy03: BOOL; // Bit03:
95 bDummy04: BOOL; // Bit04:
96 bDummy05: BOOL; // Bit05:
97 bDummy06: BOOL; // Bit06:
98 bDummy07: BOOL; // Bit07:
99 END_STRUCT;
100
101 nAW1Ack: // Unacknowledged Alarm word 1
102 STRUCT
103 bDummy08: BOOL; // Bit08:
104 bDummy09: BOOL; // Bit09:
105 bDummy10: BOOL; // Bit10:
106 bDummy11: BOOL; // Bit11:
107 bDummy12: BOOL; // Bit12:
108 bDummy13: BOOL; // Bit13:
109 bDummy14: BOOL; // Bit14:
110 bDummy15: BOOL; // Bit15:
111 bDummy00: BOOL; // Bit00:
112 bDummy01: BOOL; // Bit01:
113 bDummy02: BOOL; // Bit02:
114 bDummy03: BOOL; // Bit03:
115 bDummy04: BOOL; // Bit04:

```

Simatic Step 7 ohjelmalistaus, työkierron DB

3 (7)

```
116 bDummy05: BOOL; // Bit05:
117 bDummy06: BOOL; // Bit06:
118 bDummy07: BOOL; // Bit07:
119 END_STRUCT;
120
121 nAWDummy: ARRAY[1..6] OF WORD; // Alarm word dummy
122 nExtendAW: ARRAY[1..4] OF WORD; // Extended Alarm for HMI
123
124 nFW1: // Fault word 1
125 STRUCT
126 bDummy08: BOOL; // Bit08:
127 bDummy09: BOOL; // Bit09:
128 bDummy10: BOOL; // Bit10:
129 bDummy11: BOOL; // Bit11:
130 bDummy12: BOOL; // Bit12:
131 bDummy13: BOOL; // Bit13:
132 bDummy14: BOOL; // Bit14:
133 bDummy15: BOOL; // Bit15:
134 bDummy00: BOOL; // Bit00:
135 bDummy01: BOOL; // Bit01:
136 bDummy02: BOOL; // Bit02:
137 bDummy03: BOOL; // Bit03:
138 bDummy04: BOOL; // Bit04:
139 bDummy05: BOOL; // Bit05:
140 bDummy06: BOOL; // Bit06:
141 bDummy07: BOOL; // Bit07:
142 END_STRUCT;
143
144 nFW1Ack: // Unacknowledged Fault word 1
145 STRUCT
146 bDummy08: BOOL; // Bit08:
147 bDummy09: BOOL; // Bit09:
148 bDummy10: BOOL; // Bit10:
149 bDummy11: BOOL; // Bit11:
150 bDummy12: BOOL; // Bit12:
151 bDummy13: BOOL; // Bit13:
152 bDummy14: BOOL; // Bit14:
153 bDummy15: BOOL; // Bit15:
154 bDummy00: BOOL; // Bit00:
155 bDummy01: BOOL; // Bit01:
156 bDummy02: BOOL; // Bit02:
157 bDummy03: BOOL; // Bit03:
158 bDummy04: BOOL; // Bit04:
159 bDummy05: BOOL; // Bit05:
160 bDummy06: BOOL; // Bit06:
161 bDummy07: BOOL; // Bit07:
162 END_STRUCT;
163
164 nFWDummy: ARRAY[1..6] OF WORD; // Fault word dummy
165 nExtendFW: ARRAY[1..4] OF WORD; // Extended Fault for HMI
166
167 nEW1: // Event word 1
168 STRUCT
169 bDummy08: BOOL; // Bit08:
170 bDummy09: BOOL; // Bit09:
171 bDummy10: BOOL; // Bit10:
172 bDummy11: BOOL; // Bit11:
173 bDummy12: BOOL; // Bit12:
174 bDummy13: BOOL; // Bit13:
```

Simatic Step 7 ohjelmalistaus, työkierron DB

4 (7)

```

175 bDummy14: BOOL; // Bit14:
176 bDummy15: BOOL; // Bit15:
177 bDummy00: BOOL; // Bit00:
178 bDummy01: BOOL; // Bit01:
179 bDummy02: BOOL; // Bit02:
180 bDummy03: BOOL; // Bit03:
181 bDummy04: BOOL; // Bit04:
182 bDummy05: BOOL; // Bit05:
183 bDummy06: BOOL; // Bit06:
184 bDummy07: BOOL; // Bit07:
185 END_STRUCT;
186
187 nEWDummy: ARRAY[1..6] OF WORD; // Event word dummy
188 nExtendEW: ARRAY[1..4] OF WORD; // Extended Event for HMI
189
190 nIW1: // I-Lock word 1
191 STRUCT
192 bDummy08: BOOL; // Bit08:
193 bDummy09: BOOL; // Bit09:
194 bDummy10: BOOL; // Bit10:
195 bDummy11: BOOL; // Bit11:
196 bDummy12: BOOL; // Bit12:
197 bDummy13: BOOL; // Bit13:
198 bDummy14: BOOL; // Bit14:
199 bDummy15: BOOL; // Bit15:
200 bDummy00: BOOL; // Bit00:
201 bDummy01: BOOL; // Bit01:
202 bDummy02: BOOL; // Bit02:
203 bDummy03: BOOL; // Bit03:
204 bDummy04: BOOL; // Bit04:
205 bDummy05: BOOL; // Bit05:
206 bDummy06: BOOL; // Bit06:
207 bDummy07: BOOL; // Bit07:
208 END_STRUCT;
209
210 nIWDummy: ARRAY[1..6] OF WORD; // I-Lock word dummy
211 nExtendIW: ARRAY[1..4] OF WORD; // Extended I-Lock for HMI
212
213 nWait: INT; // Distributor time at the end of movement
214 fPosDeviation: REAL; // Allowed position deviation between D1 and 2
215 nCycleTime: INT;
216 nNegSwingPar: INT;
217 nPosSwingPar: INT;
218 fSpdFactor: REAL;
219 bMasterSlaveControl: BOOL;
220
221 bStart: ARRAY[0..11] OF BOOL; // Sequence start
222 bDone: ARRAY[0..11] OF BOOL; // Sequence done
223 bFault: ARRAY[0..11] OF BOOL; // Sequence fault
224 nStep: ARRAY[0..11] OF INT; // Sequence active step number
225
226 bD12PosCmd: BOOL;
227 nD12VelNs: ARRAY[0..20] OF DINT; // Wrk seq distributor vel setp
228 nD12PosNs: ARRAY[0..20] OF REAL; // Wrk seq distributor pos setp
229
230 bG1PosCmd: BOOL;
231 nG1VelNs: ARRAY[0..20] OF DINT; // Wrk seq gate 1 vel setpoint
232 nG1PosNs: ARRAY[0..20] OF REAL; // Wrk seq gate 1 pos setpoint

```

Simatic Step 7 ohjelmalistaus, työkierron DB

5 (7)

```

233
234 bG2PosCmd: BOOL;
235 nG2VelNs: ARRAY[0..20] OF DINT; // Wrk seq gate 2 vel setpoint
236 nG2PosNs: ARRAY[0..20] OF REAL; // Wrk seq gate 2 pos setpoint
237 fWrkSpd: REAL; // Work speed calculated from cycle time
238 fPosDevAct: REAL; // Pos deviation measured between D1 and 2
239 bPosDevOutRange: BOOL; // Bit for alarm message
240 END_STRUCT
241
242 BEGIN
243 // Step 0 Stop
244 nD12VelNs[0] := L#600; // Work sequence velocity setpoint
245 nD12PosNs[0] := 0.0; // Work sequence position setpoint
246 nG1VelNs[0] := L#600; // Work sequence velocity setpoint
247 nG1PosNs[0] := 0.0; // Work sequence position setpoint
248 nG2VelNs[0] := L#600; // Work sequence velocity setpoint
249 nG2PosNs[0] := 0.0; // Work sequence position setpoint
250
251 // Step 1 Start
252 nD12VelNs[1] := L#600; // Work sequence velocity setpoint
253 nD12PosNs[1] := 0.0; // Work sequence position setpoint
254 nG1VelNs[1] := L#600; // Work sequence velocity setpoint
255 nG1PosNs[1] := 0.0; // Work sequence position setpoint
256 nG2VelNs[1] := L#600; // Work sequence velocity setpoint
257 nG2PosNs[1] := 0.0; // Work sequence position setpoint
258
259 // Step 2 Open Gates
260 nD12VelNs[2] := L#600; // Work sequence velocity setpoint
261 nD12PosNs[2] := 0.0; // Work sequence position setpoint
262 nG1VelNs[2] := L#20000; // Work sequence velocity setpoint
263 nG1PosNs[2] := 120.0; // Work sequence position setpoint
264 nG2VelNs[2] := L#20000; // Work sequence velocity setpoint
265 nG2PosNs[2] := 120.0; // Work sequence position setpoint
266
267 // Step 3 Gates Ready
268 nD12VelNs[3] := L#600; // Work sequence velocity setpoint
269 nD12PosNs[3] := 0.0; // Work sequence position setpoint
270 nG1VelNs[3] := L#20000; // Work sequence velocity setpoint
271 nG1PosNs[3] := 0.0; // Work sequence position setpoint
272 nG2VelNs[3] := L#20000; // Work sequence velocity setpoint
273 nG2PosNs[3] := 0.0; // Work sequence position setpoint
274
275 // Step 4 Distributor Forward
276 nD12VelNs[4] := L#20000; // Work sequence velocity setpoint
277 nD12PosNs[4] := 55.0; // Work sequence position setpoint
278 nG1VelNs[4] := L#600; // Work sequence velocity setpoint
279 nG1PosNs[4] := 0.0; // Work sequence position setpoint
280 nG2VelNs[4] := L#600; // Work sequence velocity setpoint
281 nG2PosNs[4] := 0.0; // Work sequence position setpoint
282
283 // Step 5 Distributor Forward turn point
284 nD12VelNs[5] := L#20000; // Work sequence velocity setpoint
285 nD12PosNs[5] := 55.0; // Work sequence position setpoint
286 nG1VelNs[5] := L#600; // Work sequence velocity setpoint
287 nG1PosNs[5] := 0.0; // Work sequence position setpoint
288 nG2VelNs[5] := L#600; // Work sequence velocity setpoint
289 nG2PosNs[5] := 0.0; // Step 0 Stop

```


Simatic Step 7 ohjelmalistaus, työkierron DB

6 (7)

```

290
291 // Step 6 Distributor Backward
292 nD12VelNs[6] := L#20000; // Work sequence velocity setpoint
293 nD12PosNs[6] := -55.0;   // Work sequence position setpoint
294 nG1VelNs[6]  := L#600;   // Work sequence velocity setpoint
295 nG1PosNs[6]  := 0.0;     // Work sequence position setpoint
296 nG2VelNs[6]  := L#600;   // Work sequence velocity setpoint
297 nG2PosNs[6]  := 0.0;     // Work sequence position setpoint
298
299 // Step 7 Distributor Backward turn point
300 nD12VelNs[7] := L#20000; // Work sequence velocity setpoint
301 nD12PosNs[7] := -55.0;   // Work sequence position setpoint
302 nG1VelNs[7]  := L#600;   // Work sequence velocity setpoint
303 nG1PosNs[7]  := 0.0;     // Work sequence position setpoint
304 nG2VelNs[7]  := L#600;   // Work sequence velocity setpoint
305 nG2PosNs[7]  := 0.0;     // Work sequence position setpoint
306
307 // Step 8 Cycle Done
308 nD12VelNs[8] := L#600;   // Work sequence velocity setpoint
309 nD12PosNs[8] := 0.0;     // Work sequence position setpoint
310 nG1VelNs[8]  := L#600;   // Work sequence velocity setpoint
311 nG1PosNs[8]  := 0.0;     // Work sequence position setpoint
312 nG2VelNs[8]  := L#600;   // Work sequence velocity setpoint
313 nG2PosNs[8]  := 0.0; 20000; // Work sequence velocity setpoint
314
315 // Step 9 Distributor to Gate 1 closing position
316 nD12VelNs[9] := L#20000; // Work sequence velocity setpoint
317 nD12PosNs[9] := -30.0;   // Work sequence position setpoint
318 nG1VelNs[9]  := L#600;   // Work sequence velocity setpoint
319 nG1PosNs[9]  := 0.0;     // Work sequence position setpoint
320 nG2VelNs[9]  := L#600;   // Work sequence velocity setpoint
321 nG2PosNs[9]  := 0.0;     // Work sequence position setpoint
322
323 // Step 10 Close Gate 1
324 nD12VelNs[11] := L#20000; // Work sequence velocity setpoint
325 nD12PosNs[11] := 0.0;     // Work sequence position setpoint
326 nG1VelNs[11]  := L#20000; // Work sequence velocity setpoint
327 nG1PosNs[11]  := 0.0;     // Work sequence position setpoint
328 nG2VelNs[11]  := L#20000; // Work sequence velocity setpoint
329 nG2PosNs[11]  := 0.0;     // Work sequence position setpoint
330
331 // Step 11 Distributor to Gate 2 closing position
332 nD12VelNs[12] := L#20000; // Work sequence velocity setpoint
333 nD12PosNs[12] := 30.0;    // Work sequence position setpoint
334 nG1VelNs[12]  := L#20000; // Work sequence velocity setpoint
335 nG1PosNs[12]  := 0.0;     // Work sequence position setpoint
336 nG2VelNs[12]  := L#L#600; // Work sequence velocity setpoint
337 nG2PosNs[12]  := 0.0;     // Work sequence position setpoint
338
339 // Step 12 Close Gate 2
340 nD12VelNs[12] := L#20000; // Work sequence velocity setpoint
341 nD12PosNs[12] := 0.0;     // Work sequence position setpoint
342 nG1VelNs[12]  := L#20000; // Work sequence velocity setpoint
343 nG1PosNs[12]  := 0.0;     // Work sequence position setpoint
344 nG2VelNs[12]  := L#20000; // Work sequence velocity setpoint
345 nG2PosNs[12]  := 0.0;     // Work sequence position setpoint

```

Simatic Step 7 ohjelmalistaus, työkierron DB

7 (7)

```
346
347 // Step 13 Distributor to middle position
348 nD12VelNs[13] := L#2000; // Work sequence velocity setpoint
349 nD12PosNs[13] := 0.0; // Work sequence position setpoint
350 nG1VelNs[13] := L#600; // Work sequence velocity setpoint
351 nG1PosNs[13] := 0.0; // Work sequence position setpoint
352 nG2VelNs[13] := L#600; // Work sequence velocity setpoint
353 nG2PosNs[13] := 0.0; // Work sequence position setpoint
354
355 // Step 14
356 nD12VelNs[14] := L#600; // Work sequence velocity setpoint
357 nD12PosNs[14] := 0.0; // Work sequence position setpoint
358 nG1VelNs[14] := L#600; // Work sequence velocity setpoint
359 nG1PosNs[14] := 0.0; // Work sequence position setpoint
360 nG2VelNs[14] := L#600; // Work sequence velocity setpoint
361 nG2PosNs[14] := 0.0; // Work sequence position setpoint
362
363 // Step 15 Done
364 nD12VelNs[15] := L#600; // Work sequence velocity setpoint
365 nD12PosNs[15] := 0.0; // Work sequence position setpoint
366 nG1VelNs[15] := L#600; // Work sequence velocity setpoint
367 nG1PosNs[15] := 0.0; // Work sequence position setpoint
368 nG2VelNs[15] := L#600; // Work sequence velocity setpoint
369 nG2PosNs[15] := 0.0; // Work sequence position setpoint
370
371
372 END_DATA_BLOCK
```

Simatic Step 7 ohjelmalistaus, työkierron LogicFB

1 (6)

SIMATIC

SphinxV5 20130624\

CPU\IM151-8F PN/DP CPU\...\FB16 - <offline>

FB16 - <offline>

"WorkTaskLogicFB" Work task Main Logic FB
Name: **Family:**
Author: RKY **Version:** 1.0
 Block version: 2
Time stamp Code: 07/04/2013 02:43:43 PM
 Interface: 07/04/2013 02:43:43 PM
Lengths (block/logic/data): 01770 01112 00106

Name	Data Type	Address	Initial Value	Comment
IN		0.0		
OUT		0.0		
IN_OUT		0.0		
STAT		0.0		
Edge	Array [1..16] Of Bool	0.0		
bAlm	Array [1..16] Of Bool	2.0		
bAlmAck	Array [1..16] Of Bool	4.0		
bFlt	Array [1..16] Of Bool	6.0		
bFltAck	Array [1..16] Of Bool	8.0		
bILock	Array [1..16] Of Bool	10.0		
Alarm01	eventExtensionFB	12.0		
Fault01	eventExtensionFB	74.0		
Event01	eventExtensionFB	136.0		
ILock01	eventExtensionFB	198.0		
Limit01	LimitFB	260.0		
TON	TON	408.0		
StartWait	Bool	430.0	FALSE	
SEQ01	SEQ01FB	432.0		
PosDevInRange	InRangeFB	460.0		
TEMP		0.0		
DINTtoReal	Real	0.0		
GatesMoving	Bool	4.0		
DistributorsMoving	Bool	4.1		
Plug	Bool	4.2		
FaultAct	Bool	4.3		
Time_format	Time	6.0		
Wait1	Bool	10.0		
wait2	Bool	10.1		
PlugTunedDINT	DInt	12.0		
PlugTunedINT	Int	16.0		
GatesAtTarget	Bool	18.0		
Stop	Bool	18.1		
DistributorATO	Bool	18.2		
SwingLength	Int	20.0		

Simatic Step 7 ohjelmalistaus, työkierron LogicFB

2 (6)

Name	Data Type	Address	Initial Value	Comment
WorkSpdCalculated	Real	22.0		
SwingLengthDINT	DInt	26.0		
nCycleTimeDINT	DInt	30.0		
SwingLengthREAL	Real	34.0		
nCycleTimeREAL	Real	38.0		
DistribProgRpm	Real	42.0		
DistribProgRps	Real	46.0		
DistribGearSecSpdRpm	Real	50.0		
DistribGearSecSpdRps	Real	54.0		
WorkSpd	Real	58.0		
WorkSpdDINT	DInt	62.0		
Gate1AccDecDINT	DInt	66.0		
Gate2AccDecDINT	DInt	70.0		
Dist1AccDecDINT	DInt	74.0		
Dist2AccDecDINT	DInt	78.0		
WorkSpdINT	Int	82.0		

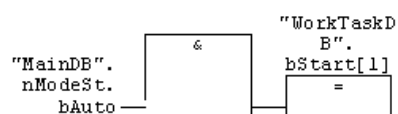
Block: FB16 WorkTask FB

Copyright (C) Outokumpu technology, Espoo Finland

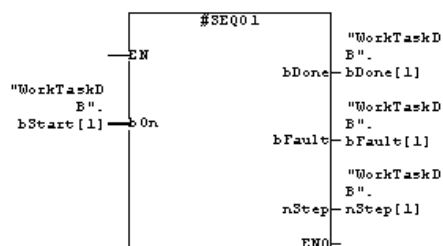
NOTES:

Revision History :

Network: 1



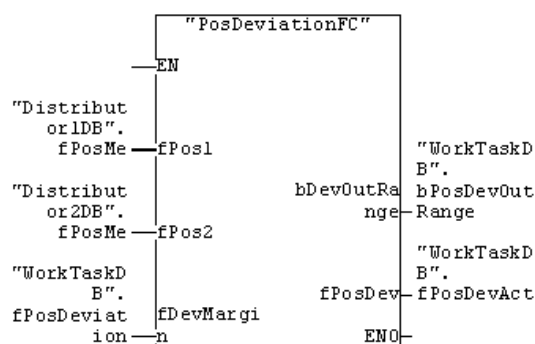
Network: 2



Simatic Step 7 ohjelmalistaus, työkierron LogicFB

3 (6)

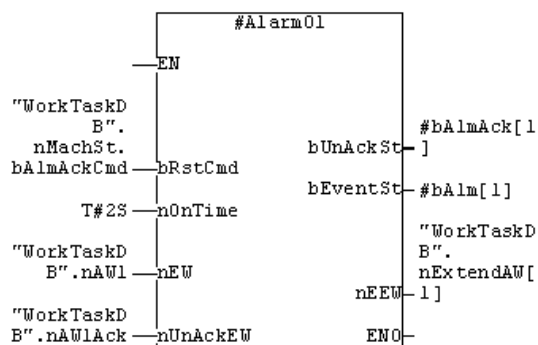
Network: 3 ALARM



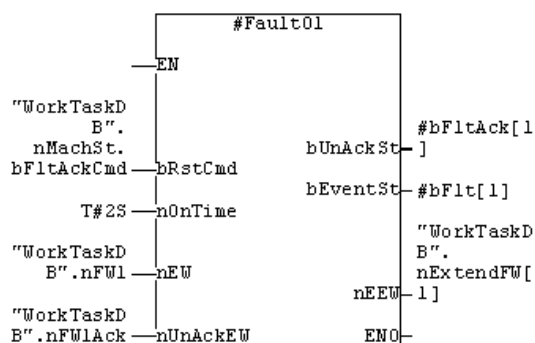
Network: 4 FAULT

Network: 5 I-LOCK

Network: 6 Alarm extension module

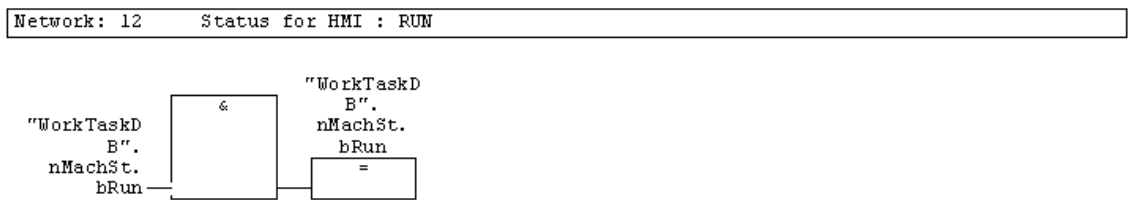
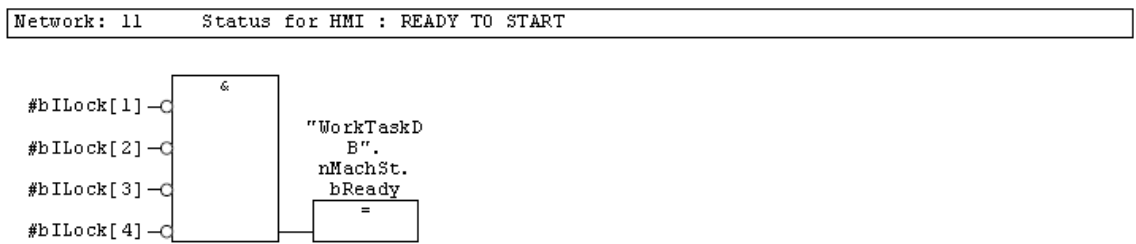
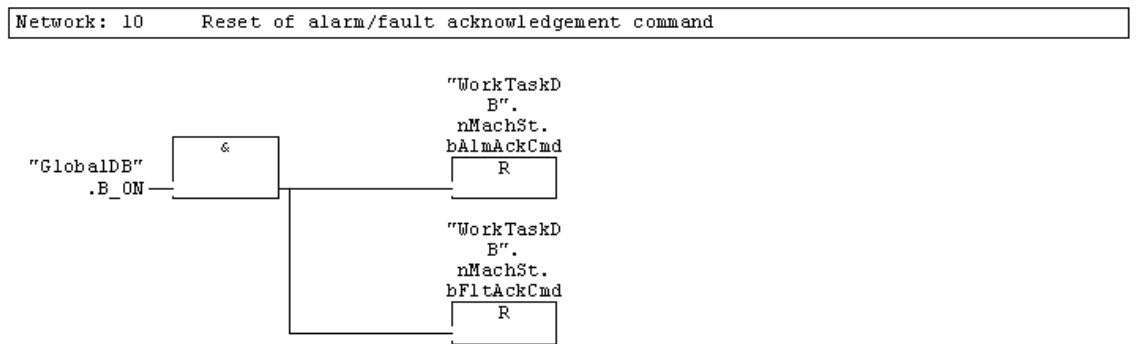
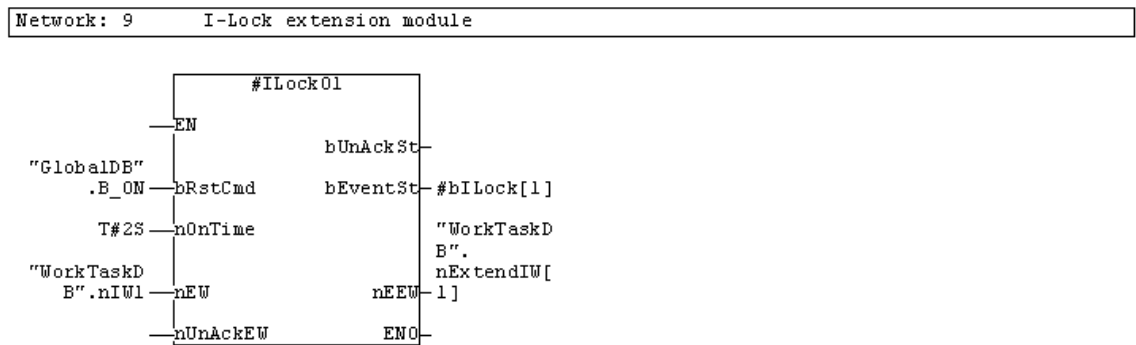
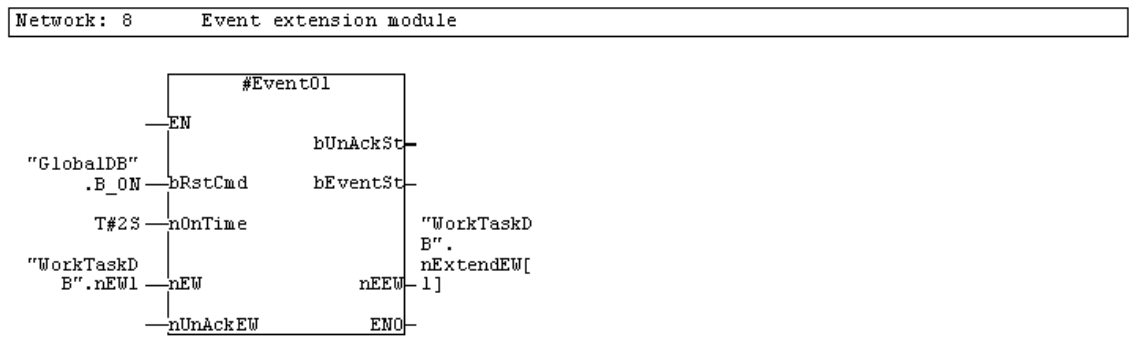


Network: 7 Fault extension module



Simatic Step 7 ohjelmalistaus, työkierron LogicFB

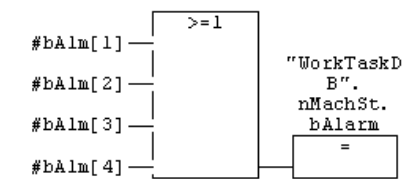
4 (6)



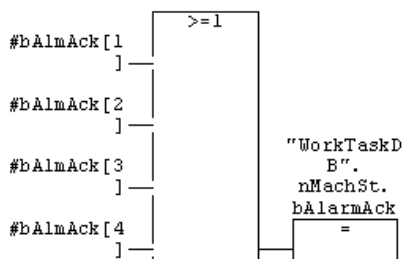
Simatic Step 7 ohjelmalistaus, työkierron LogicFB

5 (6)

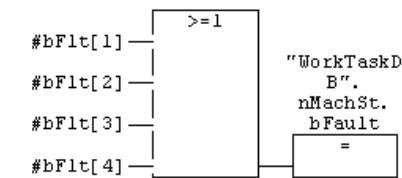
Network: 13 Alarm active



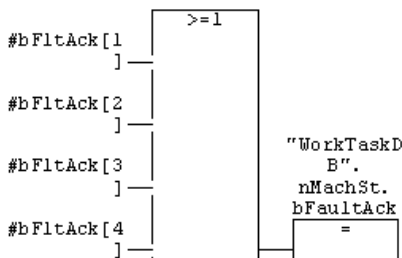
Network: 14 Unacknowledged alarm active



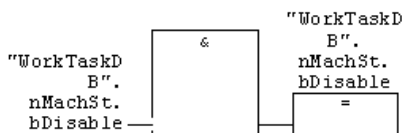
Network: 15 Fault active



Network: 16 Unacknowledged fault active



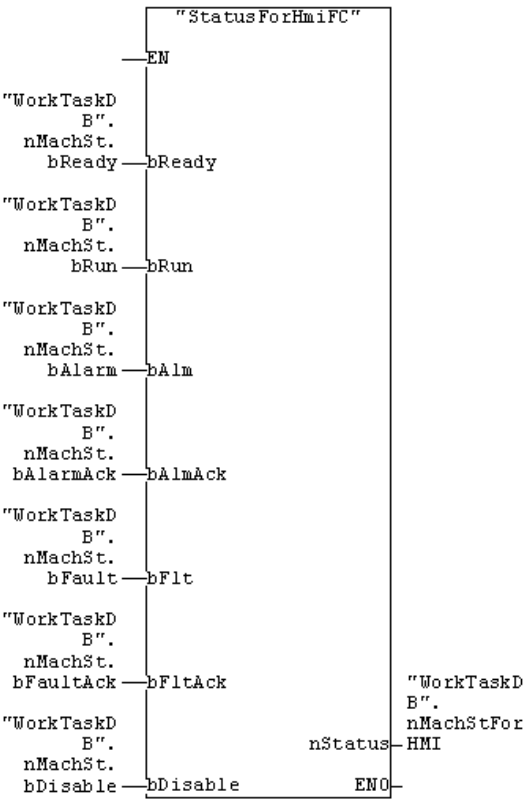
Network: 17 Disable



Simatic Step 7 ohjelmalistaus, työkierron LogicFB

6 (6)

Network: 18 Machine status for HMI symbol



Network: 19 Default handling of the ENO output

This network has to be the last one of the function

SET
SAVE

Simatic Step 7 ohjelmalistaus, UDT Telegram 111

1 (4)

SphinxV5 20130624\CPU\IM151-8F PN/DP CPU\...\UDT Telegram 111

```

1  (* Copyright (C) Outokumpu technology, Espoo Finland *)
2  (*=====*)
3  Purpose : SIEMENS Telegram 111 ( 24+24 BYTES )
4  NOTES :
5  Revision History :
6  =====*)
7

// Bit11 : r2094.01 Homing procedure active

8  TYPE "Telegram_111"
9  STRUCT
10
11  From:
12  STRUCT
13
14  SW1: // Status word 1
15  STRUCT
16  NoFlwErr:  BOOL; // Bit08 : r2684.08 Following error in tolerance
17  CtrlReq:   BOOL; // Bit09 : r0899.09 PLC Control request
18  TargPos:   BOOL; // Bit10 : r2684.10 Target position reached
19  RefPSet:   BOOL; // Bit11 : r2684.11 Reference point set
20  TrvTskAck: BOOL; // Bit12 : r2684.12 Ack, traversing block activate
21  Stndstill: BOOL; // Bit13 : r2199.00 n_act<speed threshold value 3
22  ACCEL:     BOOL; // Bit14 : r2684.04 Axis accelerating
23  DECEL:     BOOL; // Bit15 : r2684.05 Axis decelerating
24  RdyToOn:   BOOL; // Bit00 : r0899.00 Rdy for switch on
25  Ready:     BOOL; // Bit01 : r0899.01 Ready
26  Run:       BOOL; // Bit02 : r0899.02 Operation enabled
27  Fault:     BOOL; // Bit03 : r2193.03 Fault
28  NoOff2Act: BOOL; // Bit04 : r0899.04 No coasting active
29  NoOff3Act: BOOL; // Bit05 : r0899.05 No Quick Stop active
30  Inhibit:   BOOL; // Bit06 : r0899.06 Switching on inhibited active
31  Alarm:     BOOL; // Bit07 : r2139.07 Warning
32  END_STRUCT;
33
34  PosSW1: // Positioning status word 1
35  STRUCT
36  StpCamMinAct: BOOL; // Bit08 : r2684.13 STOP cam minus active
37  StpCamPlsAct: BOOL; // Bit09 : r2684.14 STOP cam plus active
38  JogAct:       BOOL; // Bit10 : r2094.00 Jog active
39  RefAct:       BOOL; // Bit11 : r2094.01 Homing procedure active
40  FlyRefAct:    BOOL; // Bit12 : r2684.01 Flying referencing active
41  TrvBlkAct:    BOOL; // Bit13 : r2094.02 Traversing block active
42  MDIStupAct:   BOOL; // Bit14 : r2094.04 MDI setup active
43  MDIPosAct:    BOOL; // Bit15 : r2094.03 MDI positioning active
44  ActTrvBit_0:  BOOL; // Bit00 : r2670.00 Active traversing Bit 0
45  ActTrvBit_1:  BOOL; // Bit01 : r2670.01 Active traversing Bit 1
46  ActTrvBit_2:  BOOL; // Bit02 : r2670.02 Active traversing Bit 2
47  ActTrvBit_3:  BOOL; // Bit03 : r2670.03 Active traversing Bit 3
48  ActTrvBit_4:  BOOL; // Bit04 : r2670.04 Active traversing Bit 4
49  ActTrvBit_5:  BOOL; // Bit05 : r2670.05 Active traversing Bit 5
50  res_6:        BOOL; // Bit06 : Reserved
51  res_7:        BOOL; // Bit07 : Reserved
52  END_STRUCT;
53
54  PosSW2: // Positioning status word 2

```

Simatic Step 7 ohjelmalistaus, UDT Telegram 111

2 (4)

```

55 STRUCT
56 PosSmCam1:    BOOL; // Bit8:r2683.08 Pos act value<=cam switc pos1
57 PosSmCam2:    BOOL; // Bit9:r2683.09 Pos act value<=cam switc pos2
58 TrvOut1:      BOOL; // Bit10:r2683.10 Direct out1 via travers block
59 TrvOut2:      BOOL; // Bit11:r2683.11 Direct out2 via travers block
60 FxStpRd:      BOOL; // Bit12:r2683.12 Fixed stop reached
61 FxStpClpTrqRd: BOOL; // Bit13:r2683.13 Fxed stop clamp torq reached
62 TrvFxStpAct:  BOOL; // Bit14:r2683.14 Travel to fixed stop active
63 CmdAct:       BOOL; // Bit15:r2684.15 Traversing command active
64 TrckModeAct:  BOOL; // Bit00:r2683.00 Tracking mode active
65 VeloLimAct:   BOOL; // Bit01:r2683.01 Velocity limiting active
66 SetPStat:     BOOL; // Bit02:r2683.02 Setpoint available
67 PrntMrkOut:   BOOL; // Bit03:r2684.03 Print mark out outer window
68 FWD:          BOOL; // Bit04:r2683.04 Axis moves forwards
69 BWD:          BOOL; // Bit05:r2683.05 Axis moves backwards
70 SftSwMinAct:  BOOL; // Bit06:r2683.06 Softlim switch minus reached
71 SftSwPosAct:  BOOL; // Bit07:r2683.07 Softlim switch plus reached
72 END_STRUCT;
73
74 SW2: // Status word 2
75 STRUCT
76 NotUsed:      BOOL; // Bit08 : r1406.08 Travel to fixed stop active
77 GlbTrggReq:   BOOL; // Bit09 : xxxxxxxx Global trigger request
78 PulsEn:       BOOL; // Bit10 : r0899.11 Pulses enabled
79 MotSwOverAct: BOOL; // Bit11 : r0835.00 Motor changeover active
80 SlvZykl_Bit0: BOOL; // Bit12 : xxxxxxxx Slave sign-of-life bit 0
81 SlvZykl_Bit1: BOOL; // Bit13 : xxxxxxxx Slave sign-of-life bit 1
82 SlvZykl_Bit2: BOOL; // Bit14 : xxxxxxxx Slave sign-of-life bit 2
83 SlvZykl_Bit3: BOOL; // Bit15 : Slave sign-of-life bit 3
84 ActDDS_Bit0:  BOOL; // Bit00 : r0051.00 Drive data set DDS eff, bit0
85 ActDDS_Bit1:  BOOL; // Bit01 : r0051.01 Drive data set DDS eff, bit1
86 ActDDS_Bit2:  BOOL; // Bit02 : r0051.02 Drive data set DDS eff, bit2
87 ActDDS_Bit3:  BOOL; // Bit03 : r0051.03 Drive data set DDS eff, bit3
88 ActDDS_Bit4:  BOOL; // Bit04 : r0051.04 Drive data set DDS eff, bit4
89 CmdActRelBrk: BOOL; // Bit05 : r2139.11 Alarm class bit 0
90 TrqContMode:  BOOL; // Bit06 : r2139.12 Alarm class bit 1
91 ParkAxisAct:  BOOL; // Bit07 : r0896.00 Parking axis active
92 END_STRUCT;
93
94 MeldW:
95 STRUCT
96 SpdOnTol: BOOL; // Bit08:r2199.04 SpdSetp-act val dev in tol
97 ESR:      BOOL; // Bit09 : r0887.12 Initiated ESR reaction
98 res_4:    BOOL; // Bit10 : xxxxxxxx Reserved
99 CE:       BOOL; // Bit11 : r0899.08 Controller enable
100 Rdy:      BOOL; // Bit12 : r0899.07 Drive ready
101 PE:       BOOL; // Bit13 : r0899.11 Pulse enable
102 res_6:    BOOL; // Bit14 : xxxxxxxx Reserved
103 res_7:    BOOL; // Bit15 : xxxxxxxx Reserved
104 NoRamp:   BOOL; // Bit00:r2199.05 1=Ramp-u/d compl, 0=gen act
105 TorqLim:  BOOL; // Bit01:r2199.11 Torq util[%]<torq thresh val2
106 SpdVal3:  BOOL; // Bit02:r2199.00 n_act<spd thresh val3 (p2161)
107 SpdVal2:  BOOL; // Bit03:r2197.01 n_act<=spd thresh val 2 (p2155)
108 VdcMinControl: BOOL; // Bit04:r0056.15 Vdc_min contr act Vdc<p1248
109 VarSigFunct: BOOL; // Bit05 : r3294xxx Variable signaling function
110 MotorOK:   BOOL; // Bit06 : r2135.14 No motor overtemperature alarm
111 PUnitOK:   BOOL; // Bit07 : r2135.15 No alarm, ther overl, pwr unit
112 END_STRUCT;
113

```

Simatic Step 7 ohjelmalistaus, UDT Telegram 111

3 (4)

```

114 ActPos:      DINT; // r2521.00 Actual position
115 ActSpd:      DINT; // r0063xxx Actual speed (Smoothed)
116 FaultCode:   WORD; // r2131xxx Actual fault code
117 AlarmCode:   WORD; // r2132xxx Actual alarm code
118 Res:         WORD; // xxxxxxxx Reserved
119
120 END_STRUCT;
121
122
123 Too:
124 STRUCT
125
126 CW1: // Control word 1
127 STRUCT
128 Jog_1: BOOL := FALSE; // Bit08 : p2589xxx EPOS jog 1 signal source
129 Jog_2: BOOL := FALSE; // Bit09 : p2590xxx EPOS jog 2 signal source
130 CtrlReq: BOOL := TRUE; // Bit10 : p0854.00 PLC control request
131 RefStart: BOOL := FALSE; // Bit11 : p2595xxx EPOS refer start
132 res_12: BOOL := FALSE; // Bit12 : xxxxxxxx Reserved
133 BlockChange: BOOL := FALSE; // Bit13 : p2633 EPOS Ext block change
134 res_14: BOOL := FALSE; // Bit14 : xxxxxxxx Reserved
135 res_15: BOOL := FALSE; // Bit15 : xxxxxxxx Reserved
136 ON: BOOL := FALSE; // Bit00 : p0840.00 On / Off
137 OFF2: BOOL := TRUE; // Bit01 : p0844.00 Coast
138 OFF3: BOOL := TRUE; // Bit02 : p0848.00 Quick stop
139 Enable: BOOL := TRUE; // Bit03 : p0852.00 Enable operation
140 RejTrvTask: BOOL := FALSE; // Bit04 : p2641 EPOS Rejec traver task
141 IntMStop: BOOL := FALSE; // Bit05 : p2640xx EPOS Intermediate stop
142 TrvStart: BOOL := FALSE; // Bit06 : p2631xx EPOS Act traver task
143 AckFault: BOOL := FALSE; // Bit07 : p2103.x0 Acknowledge faults
144 END_STRUCT;
145
146 PosCW1: // Positioning control word 1
147 STRUCT
148 MDIPsTyp: BOOL := FALSE; // Bit08:p2648 MDI Pos type 1=Abs, 0=Rel
149 MDIPos: BOOL := FALSE; // Bit09 : p2651xxx MDI Positive dir
150 MDINeg: BOOL := FALSE; // Bit10 : p2652xxx MDI Negative dir
151 res_11: BOOL := FALSE; // Bit11 : xxxxxxxx Reserved
152 MDITrTyp: BOOL := FALSE; // Bit12:p2649 MDI Trans typ 1=Cont trans
153 res_13: BOOL := FALSE; // Bit13 : xxxxxxxx Reserved
154 MDISetup: BOOL := FALSE; // Bit14 : p2653xxx MDI setp accept edge
155 MDIStart: BOOL := FALSE; // Bit15:p2647 MDI sel, make flyi changeo
156 TrvBit_0: BOOL := FALSE; // Bit00:p2625 EPOS Trav block sel Bit 0
157 TrvBit_1: BOOL := FALSE; // Bit01:p2626 EPOS Trav block sel Bit 1
158 TrvBit_2: BOOL := FALSE; // Bit02:p2627 EPOS Trav block sel Bit 2
159 TrvBit_3: BOOL := FALSE; // Bit03:p2628 EPOS Trav block sel Bit 3
160 TrvBit_4: BOOL := FALSE; // Bit04:p2629 EPOS Trav block sel Bit 4
161 TrvBit_5: BOOL := FALSE; // Bit05:p2630 EPOS Trav block sel Bit 5
162 res_6: BOOL := FALSE; // Bit06 : xxxxxxxx Reserved
163 res_7: BOOL := FALSE; // Bit07 : xxxxxxxx Reserved
164 END_STRUCT;
165
166 PosCW2: // Positioning control word 2
167 STRUCT
168 RefType: BOOL := FALSE; // Bit08:p2597 EPOS RefType sel,1=Fly 0=Se
169 RefStDi: BOOL := FALSE; // Bit09:p2604 EPOS Search ref,1=Neg, 0=Po
170 RefInpS: BOOL := FALSE; // Bit10:p2510 LR sel meas probe eval
171 RefEdge: BOOL := FALSE; // Bit11:p2511 LR meas probe eval edge
172 res_12: BOOL := FALSE; // Bit12 : xxxxxxxx Reserved
173 res_13: BOOL := FALSE; // Bit13 : xxxxxxxx Reserved

```

Simatic Step 7 ohjelmalistaus, UDT Telegram 111

4 (4)

```

174 SftLimAct: BOOL := FALSE; // Bit14:p2582 EPOS Softlimit activ
175 StpCamAct: BOOL := FALSE; // Bit15:p2568 EPOS STOP cam activ
176 TrckMode:  BOOL := FALSE; // Bit00:p2655.00 EPOS Sel tracking mode
177 SetRefPt:  BOOL := FALSE; // Bit01:p2596 EPOS Set reference point
178 ActRefCam: BOOL := FALSE; // Bit02:p2612 EPOS Set s-source ref cam
179 res_3:     BOOL := FALSE; // Bit03:xxxxxxx Reserved
180 res_4:     BOOL := FALSE; // Bit04:xxxxxxx Reserved
181 Jog_Inc:   BOOL := FALSE; // Bit05:p2591xxx EPOS Jogging incremental
182 res_6:     BOOL := FALSE; // Bit06:xxxxxxx Reserved
183 res_7:     BOOL := FALSE; // Bit07:xxxxxxx Reserved
184 END_STRUCT;
185
186 CW2: // Control word 2
187 STRUCT
188 res_0:     BOOL := FALSE; // Bit08 : xxxxxxxx Reserved
189 res_1:     BOOL := FALSE; // Bit09 : xxxxxxxx Reserved
190 res_2:     BOOL := FALSE; // Bit10 : xxxxxxxx Reserved
191 MotSwOver: BOOL := FALSE; // Bit11:p8280.00 Mot changeover fb sig
192 MsZyk_Bit0: BOOL := FALSE; // Bit12 : Master sign of life Bit0
193 MsZyk_Bit1: BOOL := FALSE; // Bit13 : Master sign of life Bit1
194 MsZyk_Bit2: BOOL := FALSE; // Bit14 : Master sign of life Bit2
195 MsZyk_Bit3: BOOL := FALSE; // Bit15 : Master sign of life Bit3
196 DDS_Bit0:  BOOL := FALSE; // Bit00:p0820.00 Drive data sel DDS bit0
197 DDS_Bit1:  BOOL := FALSE; // Bit01:p0821.00 Drive data sel DDS bit1
198 DDS_Bit2:  BOOL := FALSE; // Bit02:p0822.00 Drive data sel DDS bit2
199 DDS_Bit3:  BOOL := FALSE; // Bit03:p0823.00 Drive data sel DDS bit3
200 DDS_Bit4:  BOOL := FALSE; // Bit04:p0824.00 Drive data sel DDS bit4
201 res_3:     BOOL := FALSE; // Bit05:xxxxxxx Reserved
202 Res_4:     BOOL := FALSE; // Bit06:xxxxxxx Reserved
203 ActPrkAxis: BOOL := FALSE; // Bit07:p897xxx Parking axis sel
204 END_STRUCT;
205
206 OVERRIDE: INT := 16384; // p2646xxx EPOS Velocity override (100%)
207 MDIPos:   DINT := L#0;  // p2642xxx EPOS MDI position setpoint
208 MDIVel:   DINT := L#600; // p2643xxx EPOS MDI velocity setpoin
209 MDIAcc:   INT  := 16384; // p2644 EPOS MDI accel override (100%)
210 MDIDec:   INT  := 16384; // p2645xxx EPOS MDI decel override (100%)
211 Res:      WORD;          // xxxxxxxx Reserved
212 END_STRUCT;
213
214 END_STRUCT;
215 END_TYPE

```

Simatic Step 7 ohjelmalistaus, paikoitus FB Sinamics Position

1 (4)

SphinxV5 20130624\CPU\IM151-8F PN/DP CPU\...\FB SINAMICS
POSITION

```

1  (* Copyright (C) Outokumpu technology, Espoo Finland
2  =====
3  NOTES : SIEMENS SINAMICS POSITION CONTROL
4  Revision History :
5  =====*)
6
7  "SinaPosFB"
8  FUNCTION_BLOCK // Position measurement
9  TITLE = 'Sinamics position control FB'
10 VERSION : '1.0'
11 AUTHOR : MHK_RKy
12
13 VAR_INPUT
14 bOnCmd:      BOOL; // Drive On / Off command
15 bPosCmd:     BOOL; // Positioning On /Off
16 bFStopCmd:   BOOL; // Drive Fast stop (0 = FStop active)
17 bFwd:        BOOL; // Jog forward
18 bBwd:        BOOL; // Jog backward
19 nLADDR:      INT;  // Telegram first address from HW configuration
20 nPosNs:      DINT; // Position setpoint
21 nVelNs:      DINT; // Velocity setpoint
22 fAccLimNs:   REAL; // Acceleration override (%)
23 fDecLimNs:   REAL; // Deceleration override (%)
24 fVelLimNs:   REAL; // Velocity override (%)
25 END_VAR
26
27 VAR_IN_OUT
28
29 END_VAR
30
31 VAR_OUTPUT
32 bRdyOn:      BOOL; // Drive ready for switch on (Enable)
33 bRdyRef:     BOOL; // Drive ready for reference (Running)
34 bToPos:      BOOL; // Positioning running
35 bOnPos:      BOOL; // Target position reached
36 bAlmSt:      BOOL; // Drive alarm status
37 bFltSt:      BOOL; // Drive fault status
38 nAlm:        INT;  // Alarm code
39 nFlt:        INT;  // Fault code
40 nPosMe:      DINT; // Position measurement
41 nVelMe:      DINT; // Velocity measurement
42 END_VAR
43
44 VAR
45 bRstCmd:     BOOL; // Fault reset command
46 Drv: "Telegram_111"; // Siemens Telegram 111
47 END_VAR
48
49 VAR_TEMP
50 nRETVAL: INT; // Return value
51 END_VAR
52 BEGIN

```

Simatic Step 7 ohjelmalistaus, paikoitus FB Sinamics Position**2 (4)**

```

53
54 //*****
55 // READ TELEGRAM FROM SINAMICS
56 //*****
57
58 // Read telegram from drive
59 nRETVAl := DPRD_DAT( LADDR := INT_TO_WORD(nLADDR), RECORD :=
    Drv.From );
60
61 //*****
62 // SINAMICS CW1
63 //*****
64
65 Drv.Too.CW1.Jog_1 := bBwd; // Bit08 : p2589 EPOS jog 1 sign source
66 Drv.Too.CW1.Jog_2 := bFwd; // Bit10 : p0854.00 PLC control request
67 Drv.Too.CW1.RefStart := FALSE; // Bit11 : p2595 EPOS refer start
68 Drv.Too.CW1.res_12 := FALSE; // Bit12 : xxxxxxxx Reserved
69 Drv.Too.CW1.BlockChange := FALSE; // Bit13:p2633 EPOS Ext block cha
70 Drv.Too.CW1.res_14 := FALSE; // Bit14 : xxxxxxxx Reserved
71 Drv.Too.CW1.res_15 := FALSE; // Bit15 : xxxxxxxx Reserved
72 Drv.Too.CW1.ON := bOnCmd; // Bit00 : p0840.00 On / Off
73 Drv.Too.CW1.OFF2 := TRUE; // Bit01 : p0844.00 Coast
74 Drv.Too.CW1.OFF3 := bFStopCmd; // Bit02 : p0848.00 Quick stop
75 Drv.Too.CW1.Enable := TRUE; // Bit03 : p0852.00 Enable operation
76 Drv.Too.CW1.RejTrvTask := TRUE; // Bit04:p2641 EPOS Rej trav task
77 Drv.Too.CW1.IntMStop := TRUE; // Bit05:p2640 EPOS Intermediate stop
78 Drv.Too.CW1.TrvStart := FALSE; // Bit06:p2631 EPOS Activ trav task
79 Drv.Too.CW1.AckFault := bRstCmd; // Bit07 : p2103.x0 Ack faults
80
81 //*****
82 // SINAMICS POS CW1
83 //*****
84
85 Drv.Too.PosCW1.MDIPsTyp := TRUE; // Bit08:p2648 MDI PosType 1=Abs,0
86 Drv.Too.PosCW1.MDIPos := FALSE; // Bit09 : p2651 MDI Pos dir
87 Drv.Too.PosCW1.MDINeg := FALSE; // Bit10 : p2652 MDI Neg dir
88 Drv.Too.PosCW1.res_11 := FALSE; // Bit11 : xxxxxxxx Reserved
89 Drv.Too.PosCW1.MDITrTyp := TRUE; // Bit12:p2649 MDI TransType1=Cont
90 Drv.Too.PosCW1.res_13 := FALSE; // Bit13 : xxxxxxxx Reserved
91 Drv.Too.PosCW1.MDISetup := FALSE; // Bit14:p2653 MDI setp acc edge
92 Drv.Too.PosCW1.MDIStart := bPosCmd; // Bit15:p2647 MDI sel
93 Drv.Too.PosCW1.TrvBit_0 := FALSE; // Bit00:p2625 EPOS Trav sel Bit0
94 Drv.Too.PosCW1.TrvBit_1 := FALSE; // Bit01:p2626 EPOS Trav sel Bit1
95 Drv.Too.PosCW1.TrvBit_2 := FALSE; // Bit02:p2627 EPOS Trav sel Bit2
96 Drv.Too.PosCW1.TrvBit_3 := FALSE; // Bit03:p2628 EPOS Trav sel Bit3
97 Drv.Too.PosCW1.TrvBit_4 := FALSE; // Bit04:p2629 EPOS Trav sel Bit4
98 Drv.Too.PosCW1.TrvBit_5 := FALSE; // Bit05 : p2630 EPOS Trav block
99 Drv.Too.PosCW1.res_6 := FALSE; // Bit06 : xxxxxxxx Reserved
100 Drv.Too.PosCW1.res_7 := FALSE; // Bit07 : xxxxxxxx Reserved
101
102 //*****
103 // SINAMICS POS CW2
104 //*****
105
106 Drv.Too.PosCW2.RefType := FALSE; // Bit08:p2597 EPOS Ref type sel
107 Drv.Too.PosCW2.RefStDi := FALSE; // Bit09:p2604 EPOS Search ref
108 Drv.Too.PosCW2.RefInpS := FALSE; // Bit10:p2510 LR sel meas probe
109 Drv.Too.PosCW2.RefEdge := FALSE; // Bit11:p2511 LR probe eval edge
110 Drv.Too.PosCW2.res_12 := FALSE; // Bit12 : Reserved
111 Drv.Too.PosCW2.res_13 := FALSE; // Bit13 : xxxxxxxx Reserved

```

Simatic Step 7 ohjelmalistaus, paikoitus FB Sinamics Position

3 (4)

```

112 Drv.Too.PosCW2.SftLimAct := FALSE; // Bit14 : p2582 EPOS Soft lim
113 Drv.Too.PosCW2.StpCamAct := FALSE; // Bit15:p2568 EPOS STOP cam ac
114 Drv.Too.PosCW2.TrckMode := FALSE; // Bit00:p2655.00 EPOS Sel trac
115 Drv.Too.PosCW2.SetRefPt := FALSE; // Bit01:p2596 EPOS Set ref poin
116 Drv.Too.PosCW2.ActRefCam := FALSE; //Bit02:p2612 EPOSSetSigRef Cam
117 Drv.Too.PosCW2.res_3 := FALSE; // Bit03 : xxxxxxxx Reserved
118 Drv.Too.PosCW2.res_4 := FALSE; // Bit04 : xxxxxxxx Reserved
119 Drv.Too.PosCW2.Jog_Inc := FALSE; // Bit05 : p2591 EPOS Jog increm
120 Drv.Too.PosCW2.res_6 := FALSE; // Bit06 : xxxxxxxx Reserved
121 Drv.Too.PosCW2.res_7 := FALSE; // Bit07 : xxxxxxxx Reserved
122
123 //*****
124 // SINAMICS CW2
125 //*****
126
127 Drv.Too.CW2.res_0 := FALSE; // Bit08 : xxxxxxxx Reserved
128 Drv.Too.CW2.res_1 := FALSE; // Bit09 : xxxxxxxx Reserved
129 Drv.Too.CW2.res_2 := FALSE; // Bit10 : xxxxxxxx Reserved
130 Drv.Too.CW2.MotSwOver := FALSE; // Bit11:p8280.00 Mot changeo fb
131 Drv.Too.CW2.MsZyk_Bit0 := FALSE; // Bit12 : Master life Bit0
132 Drv.Too.CW2.MsZyk_Bit1 := FALSE; // Bit13 : Master life Bit1
133 Drv.Too.CW2.MsZyk_Bit2 := FALSE; // Bit14 : Master life Bit2
134 Drv.Too.CW2.MsZyk_Bit3 := FALSE; // Bit15 : Master life Bit3
135 Drv.Too.CW2.DDS_Bit0 := FALSE; // Bit00:p0820.00 Drive data set
136 Drv.Too.CW2.DDS_Bit1 := FALSE; // Bit01:p0821.00 Drive data set
137 Drv.Too.CW2.DDS_Bit2 := FALSE; // Bit02:p0822.00 Drive data set
138 Drv.Too.CW2.DDS_Bit3 := FALSE; // Bit03:p0823.00 Drive data set
139 Drv.Too.CW2.DDS_Bit4 := FALSE; // Bit04:p0824.00 Drive data set
140 Drv.Too.CW2.res_3 := FALSE; // Bit05 : xxxxxxxx Reserved
141 Drv.Too.CW2.Res_4 := FALSE; // Bit06 : xxxxxxxx Reserved
142 Drv.Too.CW2.ActPrkAxis := FALSE; // Bit07 : p897 Park axis select
143
144
145 //*****
146 // SINAMICS SETPOINTS TO DRIVE
147 //*****
148
149 // p2646xxx EPOS Velocity override (100%)
150 Drv.Too.OVERRIDE := REAL_TO_INT( fVelLimNs * 16384 / 100 );
151
152 // p2644xxx EPOS MDI acceleration override (100%)
153 Drv.Too.MDIAcc := REAL_TO_INT( fVelLimNs * 16384 / 100 );
154
155 // p2645xxx EPOS MDI deceleration override (100%)
156 Drv.Too.MDIDec := REAL_TO_INT( fVelLimNs * 16384 / 100 );
157
158 // p2642xxx EPOS MDI position setpoint
159 Drv.Too.MDIPos := nPosNs;
160
161 // p2643xxx EPOS MDI velocity setpoint
162 Drv.Too.MDIVel := nVelNs;
163
164 //*****
165 // WRITE TELEGRAM FROM SINAMICS
166 //*****
167
168 // Write telegram to drive
169 nRETVAL := DPWR_DAT( LADDR := INT_TO_WORD(nLADDR), RECORD :=
    Drv.Too );
170

```


Simatic Step 7 ohjelmalistaus, paikoitus FB Sinamics Position

4 (4)

```

171 //*****
172 // GENERAL STATUS
173 //*****
174
175 // Ready to On
176 bRdyOn := Drv.From.SW1.RdyToOn;
177
178 // Ready to Ref
179 bRdyRef := Drv.From.SW1.Run;
180
181 // Alarm
182 bAlmSt := Drv.From.SW1.Alarm;
183
184 // Fault
185 bFltSt := Drv.From.SW1.Fault;
186
187 // Fault reset
188 IF Drv.From.SW1.Fault THEN
189   bRstCmd := bClock2s;
190 ELSE
191   bRstCmd := FALSE;
192 END_IF;
193
194 // Target position reached
195 bOnPos := Drv.From.SW1.TargPos;
196
197 // Positionning running
198 bToPos := Drv.From.PosSW1.MDIPosAct;
199
200 // Alarm code
201 nAlm := WORD_TO_INT( Drv.From.AlarmCode );
202
203 // Fault code
204 nFlt := WORD_TO_INT( Drv.From.FaultCode );
205
206 // Position measurement
207 nPosMe:= Drv.From.ActPos;
208
209 // Velocity measurement
210 nVelMe := REAL_TO_DINT( (((Drv.From.ActSpd / 1073741824.0) *
    30000.0) / 60) / 278) * 360); //Skaalaus [deg/s]. 60[s], 278[i],
    360[deg]
211
212 //*****
213 END_FUNCTION_BLOCK

```


Simatic Step 7 ohjelmalistaus, rakennelohko OB1

1 (3)

SIMATIC

SphinxV5

20130624\CPU\IM151-8F PN/DP CPU\...\OB1 - <offline>

OB1 - <offline>

"z_OB001_CYCL_EXC" Cycle Execution

Name: Family:

Author: Version: 0.1

Block version: 2

Time stamp Code: 06/28/2013 03:07:41 PM

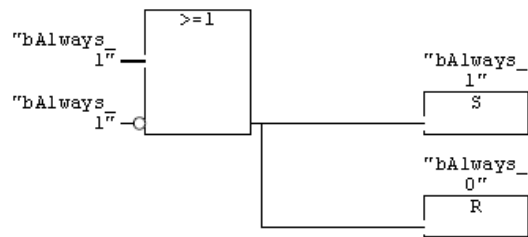
Interface: 02/15/1996 04:51:12 PM

Lengths (block/logic/data): 00552 00412 00030

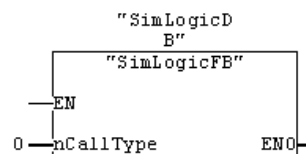
Name	Data Type	Address	Comment
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started
TEMPO	Int	20.0	
TEMP1	Int	22.0	

Block: OB1 "Main Program Sweep (Cycle)"

Network: 1 Always ON/OFF



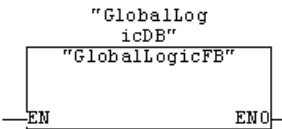
Network: 2 Simulator Logic FB / DB (OB1 First Call)



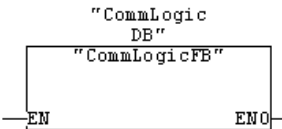
Simatic Step 7 ohjelmalistaus, rakennelohko OB1

2 (3)

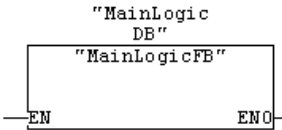
Network: 3 Global Logic FB



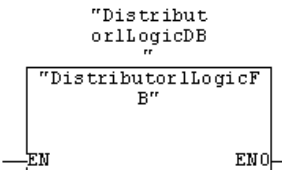
Network: 4 Communication logic



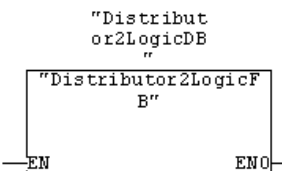
Network: 5 Main Logic



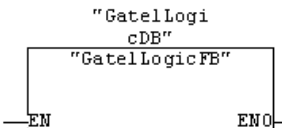
Network: 6 Distributor 1



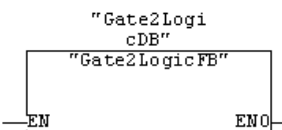
Network: 7 Distributor 2



Network: 8 Gate 1



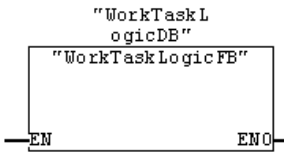
Network: 9 Gate 2



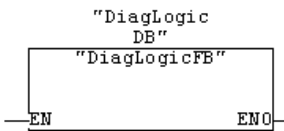
Simatic Step 7 ohjelmalistaus, rakennelohko OB1

3 (3)

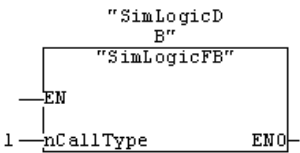
Network: 10 Work Task



Network: 11 Profibus / Profinet Diagnostic



Network: 12 Simulator Logic FB / DB (OB1 Last Call)



Network: 13 Reset First cycle on bit

