

# LIUKURENKAAN TARKASTUSLAITTEEN AUTOMATISOINTI

Keijo Keränen

Opinnäytetyö  
Lokakuu 2013

Automaatiotekniikan koulutusohjelma  
Tekniikan ja liikenteen ala





Tekijä(t) Keränen, Keijo	Julkaisun laji Opinnäytetyö	Päivämäärä 28.10.2013
	Sivumäärä 172	Julkaisun kieli Suomi
		Verkkajulkaisulupa myönnetty ( X )
Työn nimi LIUKURENKAAN TARKASTUSLAITTEEN AUTOMATISOINTI		
Koulutusohjelma Automaatiotekniikka		
Työn ohjaaja(t) Selosmaa, Seppo, lehtori		
Toimeksiantaja(t) Stresstech Oy Pyyny, Jari, tuotantopäällikkö		
Tiivistelmä <p>Liukurenkaan tarkastuslaitteen automatisointi tehtiin opinnäytetyönä Stresstech Oy:lle. Suunniteltavan laitteiston tuli olla mahdollisimman yksinkertainen käyttää ja sen piti kyetä mittaamaan liukurenkaan tarvittavat pinnat. Opinnäytetyön myötä valmistunut laitteisto toimitettiin edelleen asiakkaalle, joka käyttää laitteistoa laadunvalvontaan osana omaa valmistusprosessiaan.</p> <p>Opinnäytetyö aloitettiin laitteiston sähkösuunnittelulla, jonka perusteella tehtiin tarvittavat laitteistohankinnat ja laadittiin dokumentaatio laitteiston kokoonpanoa varten. Automaatiosuunnittelun myötä laitteistolle saatiin toiminnallinen kuvaus, johon myös PLC-ohjelmointi perustuu. PLC-ohjelmoinnissa käytettiin CoDeSys-ohjelmointiympäristöä. Kommunikointiin PLC:n ja moottorinohjaimien välillä käytettiin CAN-väylää ja CANopen-tiedonsiirtoprotokollaa.</p> <p>Suunnittelun tuloksena valmistunut tarkastuslaitteisto vastasi sille asetettuihin vaatimuksiin. Laitteistolla pystyttiin suorittamaan vaaditut mittaukset ja se on riittävän yksinkertainen käyttää. Stresstech Oy:llä ei ollut aiempaa kokemusta laitteistossa käytetystä tekniikasta yhdistettynä Barkhausen-mittaukseen, joten laitteistoa käytettiin myös uuden tekniikan testaamiseen.</p> <p>Stresstech Oy:ssä suoritettujen testausten perusteella tarkastuslaitteen toimintaan oltiin tyytyväisiä. Varmistus suunnittelun onnistumisesta saadaan, kun asiakas on käyttänyt laitteistoa riittävästi sen loppusijoituspaikassa.</p>		
Avainsanat (asiasanat) Automaatiosuunnitelma, Codesys, liukurengas, PLC, PLC-ohjelmointi, servomoottori, sähkösuunnittelu		
Muut tiedot Liitteinä: toimintakuvaus 16 sivua, turvakytkenät 6 sivua, sähkökaapin layout 1 sivu, käyttäjäliityntä 1 sivu, PLC-ohjelma 52 sivua.		



Author(s) Keränen, Keijo	Type of publication Bachelor's Thesis	Date 28.10.2013
	Pages 172	Language Finnish
		Permission for web publication ( X )
Title AUTOMATING THE WASHER INSPECTION STAND		
Degree Programme Automation Technology		
Tutor(s) Selosmaa, Seppo, lecturer		
Assigned by Stresstech Oy Pyyny, Jari, production manager		
Abstract <p>The Bachelor's Thesis "Automating the washer inspection stand" was made for Stresstech Oy. The inspection stand was meant to be as easy to use as possible and it had to be able to measure both surfaces of the washer ring. The finished inspection stand was later shipped to the customer who uses it in the quality control as part of their own production line.</p> <p>The thesis was started by creating an electrical and automation design for the inspection stand. The stand was manufactured and the components used in the stand were purchased according to the electrical drawings. The automation design described how the inspection stand was supposed to work. The PLC programming was based on the automation design. The PLC program was made using the CoDeSys software platform. The CAN bus and CANopen communication protocol was used in communication between the PLC and the motor controllers for servomotors.</p> <p>The finished washer inspection stand met the requirements set. The stand was easy to use and it was able to perform the wanted measurements. In Stresstech Oy, there was no experience in technology used in the stand and in incorporating the new technology with Barkhausen Noise Analysis.</p> <p>After thorough testing at Stresstech Oy one can say that the washer inspection stand was a success. Was the design truly a success, it is the customer who decides after using the stand as a part of their production line.</p>		
Keywords Automation desing, Codesys, electrical desing, PLC, PLC programming, servomotor, washer ring		
Miscellaneous Attachments: functional description 16 pages, safety circuits 6 pages, layout of the electrical cabinet 1page, HMI 1 page, PLC program 52 pages.		

## Sisältö

1	<b>Johdanto</b> .....	6
2	<b>Stresstech Group</b> .....	7
3	<b>Laitteistolla suoritettavat mittaukset</b> .....	7
4	<b>Laitteiston rakenne</b> .....	9
4.1	Laitteiston layout .....	9
4.2	Ohjelmoitava logiikka .....	11
4.3	Servojärjestelmä .....	12
4.4	Turvarele.....	18
4.5	CAN-väylä ja CANopen.....	19
5	<b>Riskianalyysi</b> .....	20
5.1	Koneasetus.....	20
5.2	Riskien pienentäminen .....	21
5.3	Koneen riskien arviointi Stresstech Oy:ssä .....	24
6	<b>Liukurenkaan tarkastuslaitteen sähkösuunnittelu</b> .....	25
6.1	Suunnitteluohjelmisto.....	25
6.2	UL-hyväksyntä .....	26
6.3	Käyttöjännite, ohjausjännite sekä sähkön syöttö.....	28
6.4	Kaapelointi ja johdonsuojaus.....	32
6.5	Servomoottoreiden ja askelmoottorin turvakytkenät.....	38
6.6	Sähkökaapin layout.....	40
6.7	Käyttäjäraja-alue.....	43
7	<b>Liukurenkaan tarkastuslaitteen automaatio-suunnittelu</b> .....	45
7.1	Toiminnallinen kuvaus .....	45
7.2	Ohjelmointiympäristö.....	46
7.3	FHPP – Festo Handling and Positioning Profile -tiedonsiirtoprofiili.....	57
7.4	FCT – Festo Configuration Tool -ohjelmisto .....	61
7.5	Logiikkaohjelma .....	64
7.5.1	PLC_PRG: logiikkaohjelman pääohjelma .....	64
7.5.2	Järjestelmäfunktiot .....	65
7.5.3	Toimilohkot.....	69
7.6	Laitteiston testaus .....	72
7.6.1	Sähkökytkentöjen testaus .....	72
7.6.2	Askelmoottori.....	73

	2
7.6.3 Logiikkaohjelma.....	74
7.6.4 Häiriöt.....	75
<b>8 Pohdinta .....</b>	<b>77</b>

<b>Lähteet.....</b>	<b>84</b>
---------------------	-----------

<b>Liitteet .....</b>	<b>87</b>
-----------------------	-----------

Liite 1. Turvakytkenät .....	87
Liite 2. Sähkökaapin layout.....	93
Liite 3. Käyttäjiliityntä .....	94
Liite 4. Toimintakuvaus .....	95
Liite 5. PLC_PRG .....	111
Liite 6. HOME.....	116
Liite 7. LIGHTS .....	117
Liite 8. READYFOROPERATION .....	119
Liite 9. SIGNALLIGHTTEST .....	119
Liite 10. STOPFUNCTION .....	120
Liite 11. TRIGGER .....	120
Liite 12. JOG .....	121
Liite 13. HOMING .....	123
Liite 14. SEQUENCE FLAT .....	126
Liite 15. SEQUENCE CURVED .....	140
Liite 16. HOMESEQUENCE FLAT.....	159
Liite 17. HOMESEQUENCE CURVED .....	163

## **Kuviot**

Kuvio 1. Tarkasteltava liukurengas.....	8
Kuvio 2. RollScan 250 -analysaattori .....	9
Kuvio 3. Tarkastuslaitteiston layout.....	10
Kuvio 4. Festo CPX-CEC-M1 ohjelmitava logiikka .....	12
Kuvio 5. Servojärjestelmä .....	13
Kuvio 6. Servomootorit ja askelmoottori.....	16
Kuvio 7. Tarkastuslaitteistossa käytetyt servovahvistimet .....	17
Kuvio 8. Laitteistossa käytetyt turvareleet.....	18
Kuvio 9. Riskien pienentämisen prosessi.....	22

Kuvio 10. Riskin osatekijät .....	23
Kuvio 11. UL listing- ja Recognized component -merkinnät .....	26
Kuvio 12. 480Y / 277 V -sähköjärjestelmän rakenne .....	29
Kuvio 13. Syöttökaapelin kytkentään käytetty liitin .....	30
Kuvio 14. Laitteistossa käytetty muuntaja .....	31
Kuvio 15. Älykäs kuormituspiiri ja ylivirtasuoja .....	32
Kuvio 16. Liittimet PC-kaapin kyljessä .....	33
Kuvio 17. Johtimien ja kaapelien asennustavat .....	36
Kuvio 18. Vedonpoistokisko ja kaapelien läpivienti pohjalevyn läpi .....	41
Kuvio 19. Valmis sähkökaappi .....	42
Kuvio 20. Esimerkki IL-käskylistan käytöstä .....	47
Kuvio 21. Esimerkki ST-ohjelmointikielestä .....	47
Kuvio 22. Esimerkki FBD-ohjelmointikielestä .....	48
Kuvio 23. Esimerkki LD-ohjelmointikielestä .....	48
Kuvio 24. Esimerkki SFC-ohjelmointikielestä .....	49
Kuvio 25. Esimerkki CFC-ohjelmointikielestä .....	50
Kuvio 26. Muuttujan määrittely ja symbolilistan ylläpito .....	52
Kuvio 27. Muuttujaluettelo .....	53
Kuvio 28. Syntaksin värittely .....	54
Kuvio 29. Syöttöavustaja .....	54
Kuvio 30. Taskin määrittely .....	55
Kuvio 31. PLC Configuration .....	56
Kuvio 32. Väylän konfigurointi .....	57
Kuvio 33. FHPP:n periaate .....	58
Kuvio 34. Muistialueiden varaus .....	58
Kuvio 35. Moottorinohjainta ohjaava toimilohko .....	59
Kuvio 36. Komponenttien määrittely FCT:ssä .....	62
Kuvio 37. FCT:n oskilloskooppi .....	62
Kuvio 38. FCT:n vityysparametrit .....	63
Kuvio 39. Positiotaulukko .....	64
Kuvio 40. Toimilohkojen kutsu .....	65
Kuvio 41. Järjestelmäfunktion luominen .....	66
Kuvio 42. Järjestelmäfunktio INITin toiminnallisuus .....	67
Kuvio 43. Datalaskurit .....	68
Kuvio 44. Merkkivalopainikkeen S5 ohjaus .....	70

Kuvio 45. Ote HomeSequence_flat -toimilohkosta .....	71
Kuvio 46. Siirtoehto Trans3 .....	72
Kuvio 47. Askelmoottorin uusi sijoituspaikka.....	75
Kuvio 48. Euchnerin valmistama ovikytkin.....	76

## **Taulukot**

Taulukko 1. Yhdysvalloissa käytettävät sähköverkot .....	28
Taulukko 2. AWG-johtimien muuntotaulukko .....	34
Taulukko 3. Kaapelin tai johtimen kuormitettavuus .....	35
Taulukko 4. Asennustavoista aiheutuvat korjauskertoimet.....	36
Taulukko 5. Kaapelin ylikuormasuojana toimivan sulakkeen suurin nimellisvirta .....	37
Taulukko 6. Johtimien värit .....	38
Taulukko 7. Standardin sallimat painikkeiden värit.....	43
Taulukko 8. Standardin sallimat värit merkkivaloissa .....	44

## Opinnäytetyössä käytetyt lyhenteet

A	Ampeeri, sähkövirran perussuure
AC	Alternating Current, vaihtovirta
AWG	American Wire Gauge, amerikkalainen johdinkoon merkintätapa
CAN	Controller Area Network, automaatioväylä
CFC	Continuous Function Chart, graafinen ohjelmointikieli
CoDeSys	Controller Development System, ohjelmointiympäristö
DC	Direct Current, tasavirta
FB	Function Block, POU, toimilohko,
FBD	Function Block Diagram, graafinen ohjelmointikieli
FCT	Festo Configuration Tool, työkalu servomoottoreille
FHPP	Festo Handling and Positioning Profile, tiedonsiirtoprofiili
FUN	Function, POU, funktio
IL	Instruction List, käskylistä, tekstipohjainen ohjelmointikieli
IO	Input Output, ohjelmoitavan logiikan ulkoiset kytkennät, tulot ja lähdöt
LD	Ladder Diagram, relekaavio, graafinen ohjelmointikieli
NFPA	National Fire Protection Association, kansainvälinen standardointijärjestö
PLC	Programmable Logic Controller, ohjelmoitava logiikka
POU	Program Organization Unit, CoDeSys-projektien rakennuspalikka
Prog	Program, POU, pääohjelma CoDeSys-projekteissa
SFC	Sequential Function Chart, graafinen ohjelmointikieli
SS1	Safe Stop 1, servomoottorin turvatoiminto
ST	Structured Text, lausekeskeinen ohjelmointikieli
STO	Safe Torque Off, servomoottorin turvatoiminto
UL	Underwriters Laboratories, kansainvälinen testauslaitos
V	Voltage, jännitte



# 1 Johdanto

Stresstech Groupin asiakkaat tarvitsevat valmistamiensa tuotteiden laadunvalvontaan erilaisia mittalaitteita. Tähän tarpeeseen Stresstech Group vastaa joko toimittamalla asiakkaille standardituotteita tai, kuten opinnäytetyönä tehtävän liukurenkaan tarkastuslaitteiston tapauksessa, täysin räätälöityjä ratkaisuja, jotka vastaavat asiakkaiden tarpeisiin, vaatimuksiin ja toivomuksiin.

Yrityksellä ei ollut olemassa valmista ratkaisua asiakkaan esittämään tarpeeseen liukurenkaan pinnan laadun tarkistamiseksi, joten sellainen oli kehitettävä. Yrityksellä on pitkä historia Barkhausen-menetelmään perustuvasta mittauksesta, mutta sitä ei ole aikaisemmin yhdistetty liukurenkaan tarkastuslaitteistossa käytettävään tekniikkaan, kuten servomootoreihin. Liioin ei laitteistoon valituista, Feston toimittamista moottorinohjaimista, ohjaustavasta saati käytettävästä kenttäväylästä, CAN-väylästä, ollut yrityksessä aiempaa kokemusta. Valmistunut laitteisto oli prototyyppi, ensimmäinen laatuaan.

Toimeksiantaja Stresstech Oy teetti tarkastuslaitteiston sähkö- ja automaatio-suunnittelun opinnäytetyönä. Haastavan työn tarkoituksena oli asiakkaalle toimitettavan tarkastuslaitteiston valmistuksen lisäksi uuden tekniikan testaaminen käytännössä. Projektin myötä yrityksen henkilöstö sai arvokasta tietoa ja käyttökokemusta laitteistossa käytetyistä komponenteista sekä uudelta ohjelmointiympäristöstä. Tähän asti yrityksessä on käytetty pääasiassa Siemensin ja Mitsubishin valmistamia logiikoita, liukurenkaan tarkastuslaitteen valmistumisen myötä tuohon listaan voidaan lisätä laitevalmistaja Festo. Opinnäytetyönä valmistuneen tarkastuslaitteiston myötä Stresstech Group pystyy osoittamaan, että sen valmistamissa mittalaitteistoissa pystytään hyödyntämään useiden eri laitevalmistajien tuotteita, mistä on etua kirkkavilla markkinoilla.

## 2 Stresstech Group

Stresstech Group tuottaa mittauspalveluita ja -laitteita teollisuuden tarpeisiin ympäri maailmaa. Konsernin päätuotteita ovat:

- Barkhausen-menetelmän mittauslaitteistot
- röntgendiffraktio-mittalaitteet
- jäännösjännitysten testausvälineet, jotka perustuvat reiän poraamiseen.  
(About us n.d.)

Barkhausen-menetelmää käytetään useissa erilaisissa sovelluksissa etsittäessä materiaalivirheitä ja jäännösjännityksiä. Mitattavia tuotteita ovat esimerkiksi hammaspyörät ja nokka-akselit. (Mt.)

Röntgensädeanalysointilaitteilla mitataan jäännösjännitystä sekä jäännösausteeniitin määrää. PRISM-järjestelmällä mitataan jäännösjännitteitä, menetelmä perustuu reiän poraamiseen. (Mt.)

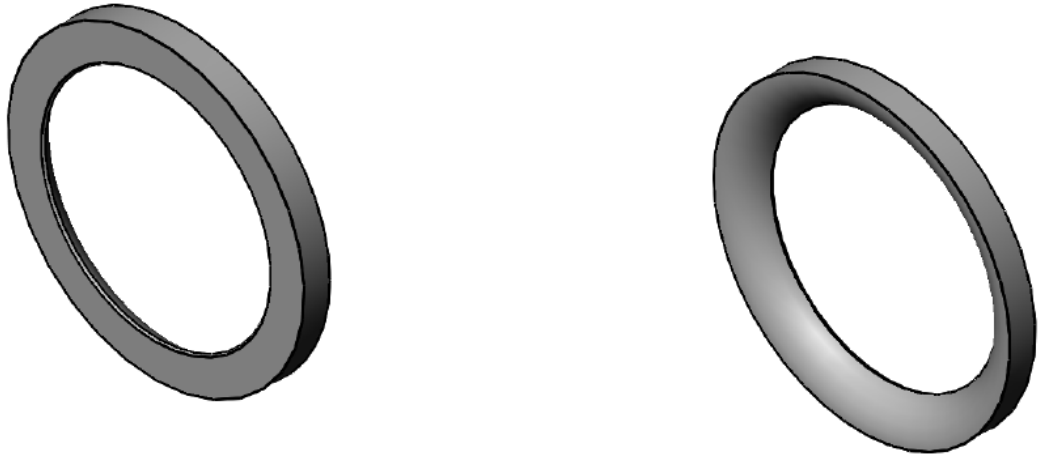
Stresstech Oy, American Stress Technologies Inc, Stresstech GmbH ja Stress-tech Bharat Pvt. Ltd. muodostavat Stresstech Groupin. Konsernin pääkonttori sijaitsee Vaajakoskella Jyväskylässä. (Mt.)

Vuonna 2012 yrityksen liikevaihto oli lähes yhdeksän miljoonaa euroa ja yrityksen tulos oli puoli miljoonaa euroa voitollinen. Yritys työllistää Vaajakoskella noin 60 henkilöä. (Tilinpäätöstilaisuus 2013.)

## 3 Laitteistolla suoritettavat mittaukset

Opinnäytetyönä valmistettava laitteisto toimitetaan asiakkaalle, joka käyttää laitteistoa laaduntarkkailuun. Tarkkailun kohteena on liukurengas, esitetty kuviossa 1, jonka molempia pintoja tullaan mittaamaan mm. hiontapalamisten

ja muiden pinnan epätäydellisyyksien havaitsemiseksi. Mittausmenetelmänä laitteistossa käytetään Barkhausen-menetelmää.



**Kuvio 1. Tarkasteltava liukurengas**

Barkhausen-menetelmä, jota kutsutaan myös magnetoelastiseksi tai mikro-magneettiseksi menetelmäksi, perustuu kohinan kaltaisen signaalin induktiiviseen mittaamiseen. Kohina syntyy, kun ferromagneettinen näyte altistetaan magneettikentälle. Ilmiö on saanut nimen keksijältään, professori Heinrich Barkhausenilta. Monet yleiset pinnankäsittelymenetelmät, kuten hionta ja induktiokarkaisu, aiheuttavat muutoksia käsitellyn kappaleen pinnan jännitykseen ja mikrorakenteeseen. Myös erilaiset dynaamiset prosessit, kuten väsyminen, aiheuttavat samankaltaisia muutoksia, joita voidaan tarkkailla Barkhausen-menetelmällä. (Barkhausen Noise Analysis n.d.) Tässä opinnäytetyössä ei käsitellä Barkhausen-menetelmää laajemmin.

Syntyneen kohinan havaitsemiseen ja analysointiin tarvitaan sensori ja mittalaitte. Mittalaitteita on sekä digitaalisia että analogisia. Opinnäytetyönä tehtä-

vässä laitteistossa mittalaitteena käytetään RollScan 250:tä, joka on esitetty kuviossa 2.

RollScan 250 on analoginen mittalaite, jota käytetään kappaleen pinnan laadunvalvontaan. Mittalaitteella havaitaan pinnan viat, jotka johtuvat mikrorakenteen muutoksista tai jännityksistä. (RollScan 250 n.d.) RollScan 250 -analyysointilaitteen tuottamaa mittausdataa kerätään ja analysoidaan ViewScan-ohjelmistolla (Software ViewScan n.d.).

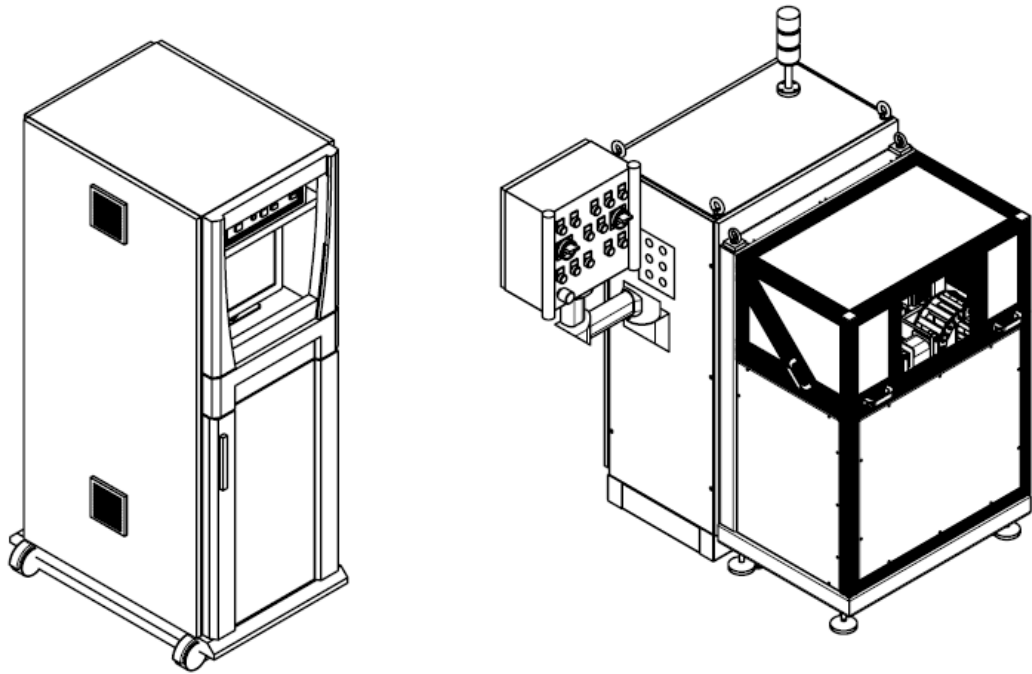


**Kuvio 2. RollScan 250 -analyysointilaitteen (RollScan 250 n.d.)**

## **4 Laitteiston rakenne**

### **4.1 Laitteiston layout**

Laitteistokokonaisuuteen kuuluvat laitteen runko, sähkökaappi, ohjauskotelo sekä erillinen PC-kaappi. Kuviossa 3 on esitetty laitteiston layout.



**Kuvio 3. Tarkastuslaitteiston layout**

PC-kaapissa on tietokoneen lisäksi mittalaite sekä mitattavan kappaleen magnetointiin käytettävä vahvistin. Sähkökaappi, runko sekä ohjauksetelo ovat fyysisesti kiinni toisissaan. PC-kaappiin tulevat kaapelit varustetaan liittimillä, jolloin niiden kytkentä nopeutuu asiakkaan tiloissa. Liittimet myös koodataan väärin kytkentöjen ehkäisemiseksi.

Mittaustapahtumassa vaaditaan sekä lineaariliikkeitä että pyörivää liikettä. Lineaariliikkeet toteutetaan servomootoreilla ja tarkoitukseen sopivilla liukujohteilla, liukurenkaan pyörytykseen käytetään askelmootoria. Liikkeiden toteuttamiseen tarvittava laitteisto valittiin Feston valikoimista.

Liikkeiden ohjaukseen käytetään ohjelmoitavaa logiikkaa, joka valittiin Feston valikoimista. Askelmootorin ja servomootoreiden moottoriohjaimet yhdistetään ohjelmoitavaan logiikkaan CAN-väylän avulla. Kommunikaatioon ohjelmoitavan logiikan ja ViewScan-ohjelmiston välillä käytettiin National Instrumentsin laitteistoa, jolla lähetetään logiikan lähdeistä saatava tieto tietokoneen Ethernet-liitäntään.

## 4.2 Ohjelmoitava logiikka

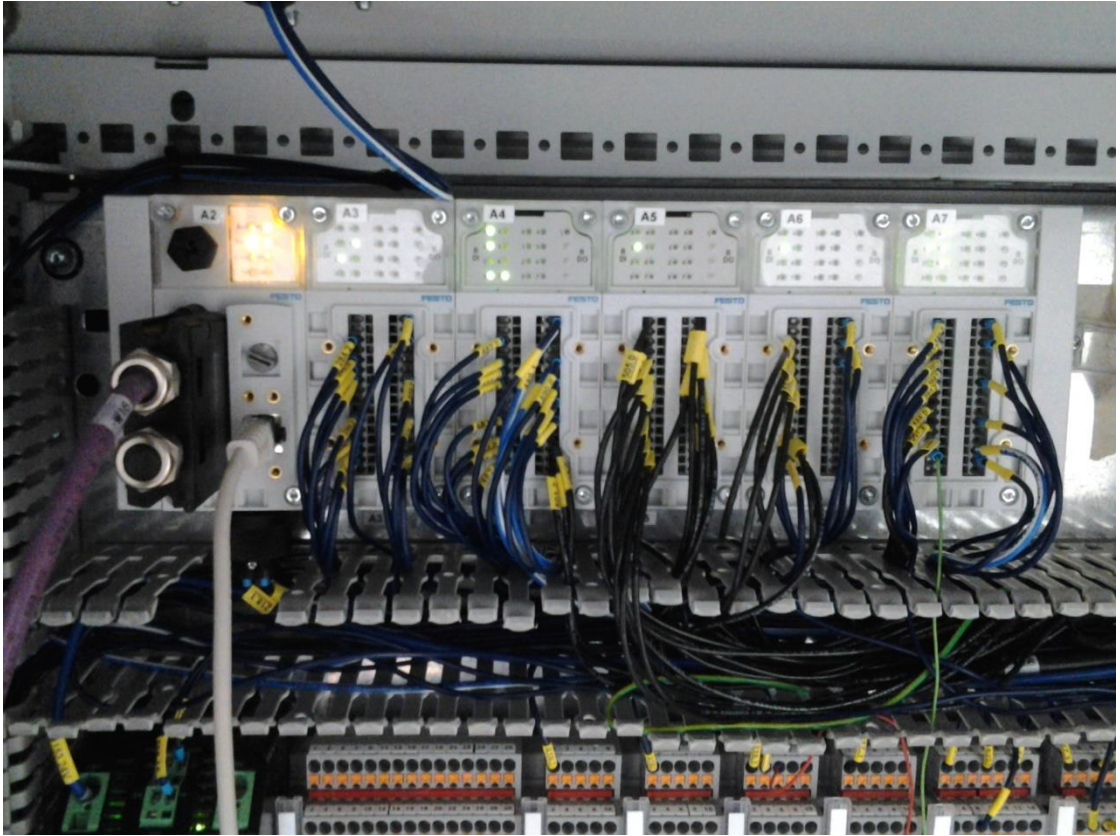
Ohjelmoitavat logiikat (PLC, Programmable Logic Controller) ovat ohjauslaitteita, jotka kehitettiin releohjauksen korvaajiksi. PLC reagoi ohjelman määrämällä tavalla anturilta saamaansa tietoon. Reagointinopeus riippuu logiikan ominaisuuksista, ohjelman määräämä reagointitapa on täysin ohjelmoijan käsissä. (Fonselius, Pekkola, Selosmaa, Ström & Välimaa 1996, 102.)

Ohjelmoitavat logiikat voidaan jakaa joko pieniin ja modulaarisiin tai pieniin, keskisuuriin ja suuriin logiikoihin. Ensiksi mainittu jakotapa perustuu PLC:n fyysisiin ominaisuuksiin. Pienet PLC:t on tarkoitettu yhden pienen koneen ohjaukseen. Modulaarinen PLC rakentuu moduuleista, joita ovat jännitelähdetyksikkö, prosessoriyksikkö ja erilaiset IO-yksiköt. Jaettaessa PLC:t pieniin, keskisuuriin tai suuriin PLC:hin käytetään jakoperusteena PLC:n kapasiteettia käsitellä IO:ta. (Mts. 105.)

Ohjelmoitavan logiikan tulot ja lähdöt, IO-yksiköt, ovat välttämättömiä prosessien valvonnassa ja ohjaamisessa. Ne voidaan jakaa kahteen perustyyppiin: loogisiin ja jatkuviin. Looginen ohjaus tarkoittaa sitä, että tulolla tai lähdöllä on vain kaksi tilaa: päällä tai pois päältä. Esimerkkinä voidaan käyttää hehku lampun ohjausta: jos lampun voi ainoastaan sytyttää tai sammuttaa, on ohjaus looginen. Jos lampua voidaan himmentää eri kirkkauksiin, on kyseessä jatkuva ohjaus. Logiikkaohjaukset toteutetaan pääasiassa loogisilla ohjauksilla. (Inputs and outputs 2010.)

Ohjelmoitavan logiikan lähtöön liitetyn toimilaitteen avulla PLC kykenee vaikuttamaan prosessin kulkuun ja tilaan. Tuloihin liitetyt anturit muuttavat fyysisen ilmiön tai tapahtuman sähköiseksi signaaliksi ja näin ohjelmoitava logiikka saa tietoa prosessin kulusta. (Mt.)

Opinnäytetyönä tehtävässä laitteistossa käytettävä PLC:n ohjausyksikkö on Feston CPX-CEC-M1, joka on esitetty kuviossa 4.



**Kuvio 4. Festo CPX-CEC-M1 ohjelmoitava logiikka**

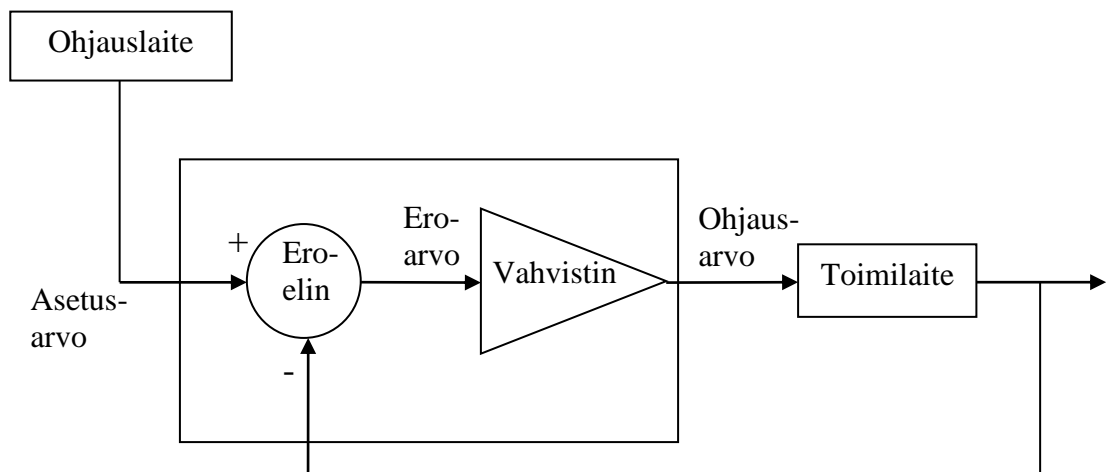
PLC käyttää CAN-väylää kommunikointiin muiden väylään liitettävien laitteiden, esimerkiksi moottoriohjaimien välillä. Ohjausyksikön lisäksi PLC on varustettu viidellä IO-yksiköllä, joissa kussakin IO on kahdeksan tuloa ja kahdeksan lähtöä. Feston PLC:lle ei ole erillistä virtalähdettä, vaan 24 V DC -käyttöjännite saadaan muuntajasta, jota käytetään laitteiston ohjausjännitteen tuottamiseen.

### 4.3 Servojärjestelmä

Servon tai servomekanismin toiminta perustuu negatiivisen takaisinkytkennän periaatteeseen. Negatiivisessa takaisinkytkennässä järjestelmän lähtösuu-

retta verrataan ohjaussuureeseen. Järjestelmä pyrkii pitämään suureiden erotuksen nollana. (Younkin 1996, 3–4.)

Servojärjestelmän rakenne, joka on esitetty kuviossa 5, koostuu ohjauslaitteesta, servovahvistimesta sekä toimilaitteesta. Ohjauslaite syöttää servovahvistimelle asetusarvon, johon verrataan takaisinkytkentänä saatavaa mittaustietoa. Tämän eroarvon perusteella servovahvistin ajaa toimilaitteen haluttuun asemaan tai nopeuteen. (Fonselius, Rinkinen & Vilenius 1998, 8.)



**Kuvio 5. Servojärjestelmä (Fonselius, Rinkinen & Vilenius 1998, 8, muokattu)**

Takaisinkytkennässä käytetään erilaisia ratkaisuja sen mukaan, mitä suuretta servon ohjaukseen käytetään. Aseman mittaus voidaan toteuttaa esimerkiksi pulssiantureilla tai potentiometreilla, nopeuden mittaamiseen voidaan käyttää esimerkiksi takogeneraattoria. (Mts. 8.)

### **Siirtymän ja kiertymiskulman mittaaminen**

Siirtymän ja kiertymän mittaamiseen voidaan käyttää analogisia tai digitaalisia antureita. Suoraviivaista liikettä mitataan lineaarisesti liikkuvilla antureilla, mutta myös pyörivää anturia voidaan käyttää, jos suoraviivainen liike



muutetaan ensin pyöriväksi liikkeeksi. Analogisen anturin toiminta perustuu anturin antamaan lähtöjännitteeseen ja sen suuruuteen. Digitaalisia antureita on sekä pulssi- että koodiantureita. Pulssianturin muodostamassa pulssijonossa olevien pulssien määrä on verrannollinen siirtymään, kun taas koodianturi muodostaa siirtymää vastaavan digitaalisanan. (Mts. 123.)

### **Pyörimisnopeuden mitta**

Pyörimisnopeus voidaan mitata analogisesti tai digitaalisesti. Analogisesti mitattaessa anturina käytetään takogeneraattoria, joka kiinnitetään pyörivän akselin päähän. Takogeneraattorin tuottama jännitetaso on suoraan verrannollinen pyörimisnopeuteen. Digitaalisessa pyörimisnopeuden mittaamisessa voidaan käyttää pulssianturia, jonka synnyttämän pulssijonon taajuudesta saadaan selvitettyä pyörimisnopeus. (Mts. 129–131.)

### **Servomoottori**

Servojärjestelmässä voidaan käyttää mitä tahansa sähkömoottoria. Moottorityyppien erilaisuuksista johtuen on tärkeää, että sovellukseen valittu moottori on oikean tyyppinen ja kokoinen. Yleensä servojärjestelmän sähkömoottoriksi valitaan kuitenkin servomoottori, jossa on sisäänrakennettuna pulssianturi ja takogeneraattori takaisinkytkentää varten. (Mts. 10, 139.)

Askelmoottori on servojärjestelmissä yleinen moottorityyppi. Se eroaa muunnaisista sähkömoottoreista siten, että se pyörii askeleittain, jotka toistuvat yhtä suurina jokaista ohjauspulssia kohti. Ohjauspulssin tuottamiseen tarvitaan erillinen moottorinohjain. Pakollisen ohjauselektroniikan lisäksi askelmoottorin huonoja puolia ovat suurin virran tarve sekä epätasainen pyörimisnopeus. Suuresta virran tarpeesta johtuen askelmoottorikäytöt ovat yleensä pieniä. Askelmoottorien hyviä ominaisuuksia ovat helppo ohjattavuus digitaalisilla komponenteilla sekä se, ettei nopeuden ja aseman takaisinkytkentää välttämättä tarvita. (Mts. 148–150.)

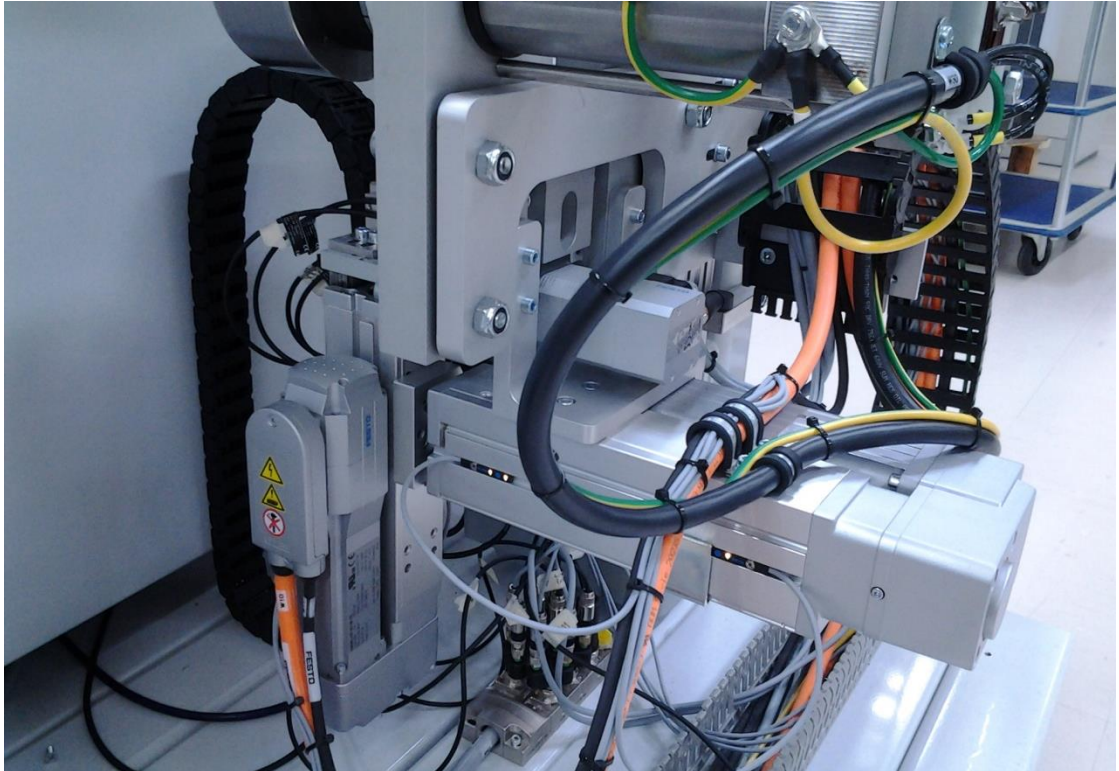
Opinnäytetyönä tehtävässä tarkastuslaitteistossa lineaariliikkeiden toteuttamiseen käytetään servomoottoreita ja kappaleen pyörittämiseen askelmoottoria. Moottorit ja tarvittavat moottorikaapelit ovat Feston valmistamia. Servomoottoreista suurikokoisin poikkeaa muista servomoottoreista siten, että se on varustettu jarrulla. Jarru tarvitaan, koska liikuteltava massa on suuri. Pienemmillä servomoottoreilla ja askelmoottorilla jarrua ei tarvita.

Moottoreissa on sisäänrakennettuna absoluuttianturi paikan ja nopeuden mittaamista varten. Moottorien käyttöjännite on 360 V DC nimellisvirran vaihdellessa 0,6–2,6 A:iin. (Servo motors EMMS-AS 2013, 4–5, 7, 11.)

Tarkastuslaitteistossa käytetyt moottorit ovat:

- EMMS-AS-40-M-LS-TS
- EMMS-AS-55-M-LS-TS
- EMMS-AS-70-M-LS-RSB.

Askelmoottori on tyypiltään EMMS-ST-42-S-SE-G2. Moottori on varustettu sisäänrakennetulla pulssianturilla, jolla saadaan moottorin asema ja nopeus selville. Moottorin käyttöjännite on 48 V DC ja nimellisvirta 1,8 A. (Stepper motors EMMS-ST 2013, 4–5.) Kuviossa 6 näkyvät servomoottorit EMMS-AS-40 ja EMMS-AS-55 sekä askelmoottori EMMS-ST-42.



**Kuvio 6. Servomootorit ja askelmoottori**

### **Servovahvistin**

Servovahvistimella ohjataan toimilaitetta ohjauslaitteelta saatavan ohjausarvon perusteella. Servovahvistimessa olevassa eroelimessä verrataan takaisin-kytkentänä saatavaa mittausarvoa ohjausarvoon. Tuloksena saatava eroarvo muutetaan vahvistimessa toimilaitteen ymmärtämään muotoon, esimerkiksi virraksi. Servovahvistimella syötetään servomoottorille sen tarvitsema virta. (Fonselius, Rinkinen & Vilenius 1998, 8–9, 131.)

Servovahvistinta valittaessa on varmistettava sen yhteensopivuus käytettävän servomoottorin kanssa, sillä erityyppiset servomoottorit tarvitsevat omanlaisensa servovahvistimen. Periaatteessa servovahvistintyytit eroavat toisistaan vain pääteasteen ja sitä ohjaavan elektroniikan perusteella. (Mts. 132–133.)

Opinnäytetyönä valmistuneessa tarkastuslaitteistossa käytettiin Feston valmistamia moottorinohjaimia sekä servomoottoreille että askelmoottorille. Ser-

vomootoreiden ohjaimet ovat tyypiltään CMMS-AS-C4-3A-G2. Askelmoottorin ohjain on tyypiltään CMMS-ST-C4-3A-G2. Yhteistä ohjaimilla on se, että niissä on integroituna CANopen-rajapinta, jota käytettiin ohjauksen toteutuksessa. Servomootoreiden sekä askelmoottorin paikoitus voidaan toteuttaa joko tallentamalla valmiiksi opetettuja asemia ohjaimien muistiin tai antamalla logiikan hoitaa paikan laskeminen ja näin antaa ajo-ohjeet moottorinohjaimille (Motor controllers, CMMS-AS, for servo motors 2012, 2–12). Molemmat ohjaimet tukevat mm. edellä mainittuja ajotapoja. Tarkastuslaitteistossa käytetyt servovahvistimet on esitetty kuviossa 7.



**Kuvio 7.** Tarkastuslaitteistossa käytetyt servovahvistimet

## 4.4 Turvarele

Koneiden, prosessien ja järjestelmien turvallinen käyttö varmistetaan turvareleillä. Esimerkiksi hätäpysäytys voidaan toteuttaa yksinkertaisesti turvareleellä ja hätäpysäytyspainikkeella. Riskikartoituksen vaatiessa korkeinta turvaluokkaa tulee mekaaniset komponentit kahdentaa. Mekaanisilla komponenteilla tarkoitetaan kontaktoreita ja hätäpysäytys-painikkeiden koskettimia. (Turvareleet ja hätäpysäytys 2013.)

Liukurenkaan tarkastuslaitteeseen tarvittiin kaksi turvarelettä. Toista turvarelettä käytettiin hätäpysäytyksen valvontaan ja toista turvaporttivahtina. Laitteistoon valittiin Pilzin PNOZ s5 -turvareleet, jotka on esitetty kuviossa 8. Valintaan vaikutti se, että releissä on kahdet hidastetut turvakoskettimet, joita tarvitaan servojen hallittuun pysäyttämiseen hätätapauksessa. Lisäksi Festo käyttää omissa käyttöohjeissaan esimerkkinä Pilzin tuotteita, joten hätäpysäytyskytkentöjen suunnittelua ei tarvinnut aloittaa tyhjästä. Turvareleiden lisäksi Pilzin valikoimista tilattiin ovikytkin laitteiston turvaoveen.



Kuvio 8. Laitteistossa käytetyt turvareleet

## 4.5 CAN-väylä ja CANopen

CAN-väylä (Controller Area Network) on alun perin suunniteltu autoteollisuuden tarpeisiin. Ajoneuvokäytön lisäksi CAN-väylää käytetään mm. robo-teissa ja ohjelmoitavissa logiikoissa. CAN ei ole periaatteeltaan kenttäväylä, mutta sitä käytetään usein kenttäväylän tapaan. (Alanen 2000, 1–2.) CAN on kansainvälisesti standardoitu ISO 11898-1:ksi, joka kattaa ISO:n OSI-mallin siirtoyhteyskerroksen (Controller Area Network 2013).

CAN-väylä tukee useaa yhtäaikaista isäntää. Väylän jokainen solmu (Node) voi lähettää väylälle dataa, jos väylä on toimettomana. Kaikki sanomat lähetetään yleisesti vastaanotettaviksi ja jokainen vastaanottava solmu päättää itse, ottaako se sanoman vastaan vai ei. Tämä takaa tietojen yhdenmukaisuuden, koska kaikki solmut käyttävät samaa informaatiota. Usean solmun yrittäessä lähettää viestin väylälle yhtä aikaa, pitää protokolla huolen siitä että, korkeimman prioriteetin omaava sanoma pääsee väylälle. (Mt.)

CANopen on yksi käytetyimmistä CAN-väylän sovelluskerroksen protokollista. CANopen määritettiin alun perin toimivaksi ainoastaan CAN-verkoissa, mutta se on yleisyytensä ja joustavuutensa ansiosta otettu osaksi mm. Ethernet/IP-verkon käyttöön perustuvia Ethernet Powerlink- ja EtherCAT-väyliä. (Saha 2006, 6.)

CANopenin määrittelemistä asioista oleellisin on jokaisessa solmussa sovelluksen ja väyläliikennöinnin toisistaan erottava objektikirjasto (object directory), jonka avulla voidaan tunnistaa solmu ja hallita kaikkea solmun toimintaa väylältä käsin. Sovelluskerroksen palveluiden sekä solmu- ja tiedonsiirtomallin lisäksi CANopen määrittelee myös muita tarpeellisia asioita, kuten liittimien tyyppejä pinnijärjestyksineen, solmun ajoitusparametrit liikennöintinopeuksittain sekä solmun tilaa kuvaavien merkkivalojen toiminnan. Näin taataan solmujen keskinäinen yhteensopivuus ja helpotetaan järjestelmäintegroitua. (Mts. 6.)

## 5 Riskianalyysi

### 5.1 Koneasetus

Valtioneuvoston asetus koneiden turvallisuudesta, ns. koneasetus, perustuu konedirektiiviin 2006/42/EY, jossa määritellään Euroopan talousalueella koneita koskevat tekniset vaatimukset ja vaatimuksenmukaisuuden osoittamismenettelyt. Myös Euroopan ulkopuolelta tuotavien koneiden on täytettävä koneasetuksen vaatimukset. Koneasetuksen velvoitteista huolehtii se osapuoli, jonka toimesta kone saatetaan markkinoille, käytännössä tällä tarkoitetaan koneen valmistajaa tai valmistajan valtuuttamaa maahantuojaa tai jälleenmyyjää. Asetus koskee uusia, Euroopan talousalueelle vietäviä, kotimarkkinoille sekä omaan käyttöön valmistettuja koneita, olivat ne sitten sarjavalmistaisia tai yksittäiskappaleita, pieniä käsikäyttöisiä tai suuria konelinjoja. (Koneturvallisuus 2008, 3–4.)

Koneasetuksen mukaan koneen valmistajan on:

- arvioitava riskit
- selvitettävä konetta koskevat turvallisuusvaatimukset
- suunniteltava ja rakennettava kone olennaisten turvallisuusvaatimusten mukaisesti
- laadittava käyttöohjeet ja tehtävä koneeseen tarvittavat merkinnät
- laadittava tekninen tiedosto
- tehtävä vaatimustenmukaisuusvakuutus
- kiinnitettävä koneeseen CE-merkintä. (Mts. 6.)

## **Kone**

Koneella tarkoitetaan toisiinsa liitettyjen osien tai komponenttien yhdistelmää, jossa ainakin yksi osa tai komponentti liikkuu ja jossa on tarvittavat toimilaitteet ohjaus- ja energiansyöttöpiireineen. Kone on kokoonpantu tiettyä toimintoa, kuten materiaalin työstöä varten. Koneella tarkoitetaan myös koneyhdistelmiä, jotka on järjestetty ja ohjattu toimimaan yhtenä kokonaisuutena. (SFS-EN ISO 12100 2010, 12.)

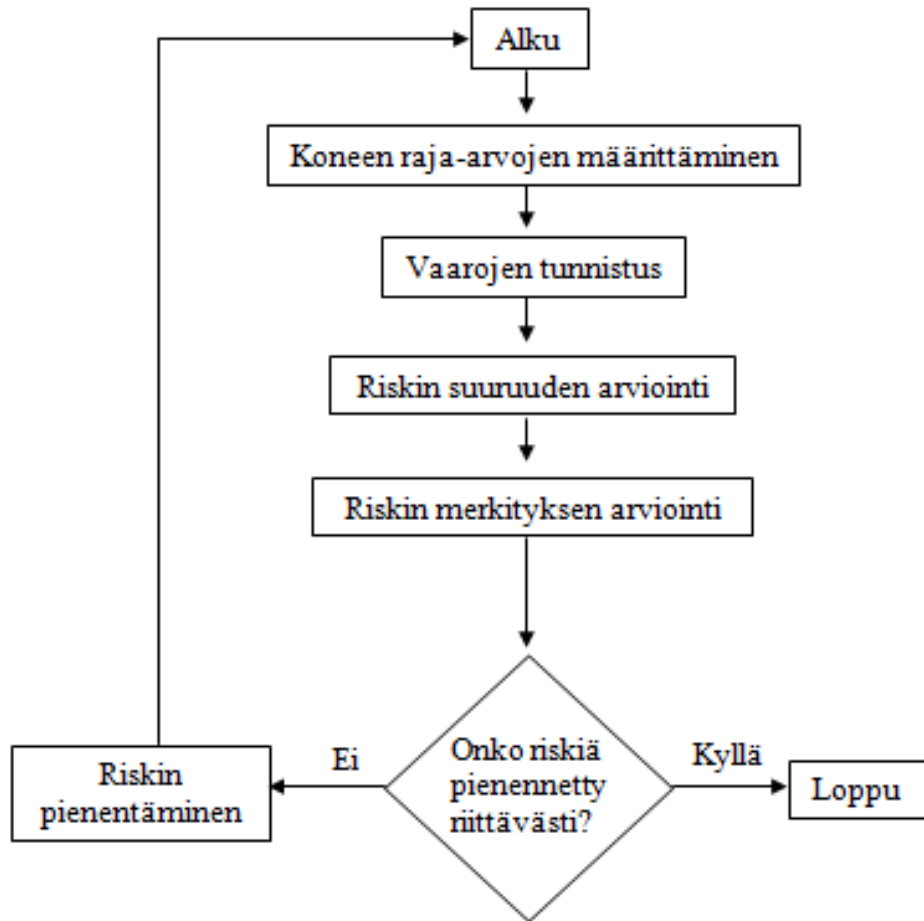
## **CE-merkintä**

CE-merkinnällä osoitetaan koneen täyttävän koneasetuksen olennaiset turvallisuusvaatimukset sekä muut konetta mahdollisesti koskevat ja CE-merkintää edellyttävät säännökset. Muita konetta koskevia direktiivejä ovat esimerkiksi pienjännitedirektiivi ja sähkölaitteiden magneettista yhteensopivuutta koskeva direktiivi. (Koneturvallisuus 2008, 16.)

## **5.2 Riskien pienentäminen**

Riskien pienentämisen prosessi SFS-EN ISO 12100:n mukaan etenee kuviossa 9 kuvatulla tavalla.





Kuvio 9. Riskien pienentämisen prosessi (SFS-EN ISO 12100 2010, 30, muokattu)

### Koneen raja-arvojen määrittäminen

Riskin arviointi aloitetaan koneen raja-arvojen määrittämisellä. Määritettävät raja-arvot ovat:

- käyttörajat, joihin sisältyvät koneen tarkoitettu käyttö ja kohtuudella ennakoitavissa oleva väärinkäyttö
- tilarajat, joihin sisältyvät mm. koneen tekemän liikkeen laajuus sekä ihmisen vuorovaikutus
- aikarajat, joihin sisältyvät koneen elinikä sekä suositeltavat huoltovälit

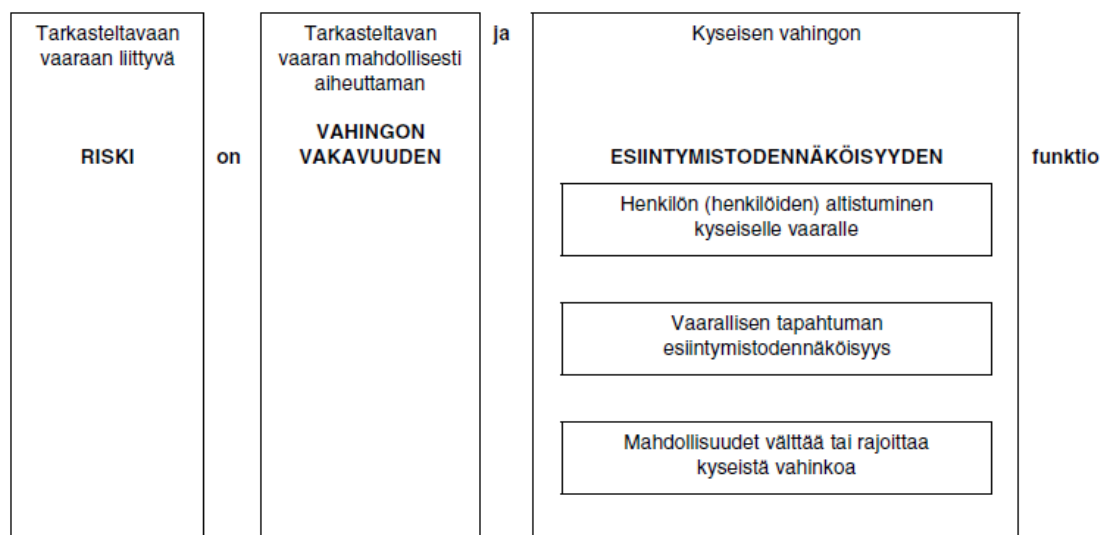
- muut raja-arvot, joita ovat esimerkiksi ympäristöön ja puhtaanapitoon liittyvät raja-arvot sekä käsiteltävän materiaalin ominaisuudet. (SFS-EN ISO 12100 2010, 36–38.)

### Vaaran tunnistaminen

Raja-arvojen tunnistamista seuraa vaaran tunnistaminen, jossa tunnistetaan koneen elinkaaren kaikkien vaiheiden aikaiset ja kohtuudella ennakoitavissa olevat vaaratilanteet ja vaaralliset tapahtumat. Vaarojen tunnistamiseksi on tunnistettava koneella suoritettavat toiminnot sekä tehtävät, joita konetta käyttävien henkilöiden on tarkoitus suorittaa koko koneen elinkaaren aikana. Näitä toimintoja ovat esimerkiksi koneen käynnistäminen sekä kunnossapidon työt. Kohtuudella ennakoitavia vaaratilanteita ovat esimerkiksi melu ja koneen rikkoutuminen. (Mts. 38–40.)

### Riskin suuruuden arviointi

Jokaiselle vaaratilanteelle suoritetaan riskin suuruuden arviointi määrittämällä riskin suuruuden osatekijät (Mts. 42), jotka on esitetty kuviossa 10.



Kuvio 10. Riskin osatekijät (SFS-EN ISO 12100 2010, 42)

Riskin suuruutta arvioitaessa on otettava huomioon kaikki käyttäjät, jotka voivat altistua vaaralle. Altistumisen suuruuden arvioimiseksi ovat koneen työmenetelmät ja toimintatavat selvitettävä ja analysoitava, myös tehtävät joiden vuoksi suojaustoimenpiteitä on otettava tilapäisesti pois käytöstä, on otettava huomioon. (Mts. 46–48.)

### **Riskin merkityksen arviointi**

Riskin merkityksen arvioinnissa päätetään, tarvitaanko riskin pienentämistä. Jos pienentäminen nähdään tarpeelliseksi, arviomenettely uusitaan suojaustoimenpiteiden jälkeen. Riittävä riskin pienentäminen saavutetaan kolmen askeleen menetelmällä. Askeleet ovat:

- vaaran poistaminen, tai riskin pienentäminen rakenteellisin toimenpitein
- riskiä on pienennetty riittävästi suojausteknisin toimenpitein
- käyttöä koskevissa tiedoissa on ilmoitus kaikista jäännösriskeistä, jos suojaustekniset toimenpiteet eivät ole käytännössä mahdollisia. (Mts. 50–52.)

### **Riskien vertailu**

Koneen tai koneen osiin liittyviä riskejä voidaan vertailla muiden samankaltaisten koneiden ja koneen osien riskeihin jos molemmat koneet ovat standardien mukaisia ja riskit vertailukelpoisia keskenään. Tämä ei kuitenkaan poista tarvetta noudattaa riskin arvioinnin prosessia. (Mts. 50.)

## **5.3 Koneen riskien arviointi Stresstech Oy:ssä**

Valmistettavien tuotteiden riskianalyysi laaditaan esisuunnittelun jälkeen projektin aloituspalaverin yhteydessä. Vakiotuotteista ei tehdä sarjanumerokoh- taista riskianalyysiä, vaan riittää että ensimmäisestä yksilöstä on tehty analyysi-

si ja seuraavien yksilöiden dokumentaatiossa viitataan ensimmäisen yksilön analyysiin. Asiakkaalle räätälöitävistä tuotteista tehdään jokaisesta oma riskianalyysinsä. (Koneturvallisuus ja CE-merkintä 2010, 4.)

Riskianalyysin tekoon osallistuvat kaikki tuotteen suunnitteluun ja testaus- ja todentamisvaiheisiin osallistuvat henkilöt. Näin vaaratekijöistä saadaan kattava näkemys sekä ratkaisuehdotukset niiden poistamiseksi saadaan aikaan hyödyntäen eri alojen asiantuntijoiden näkemyksiä. Riskien arvioinnista täytetään riskianalyysidokumentti. (Mts. 5.)

Riskianalyysiä tarkastellaan suunnittelun edetessä. Tuotteen lopputarkastuksessa tarkistetaan, että riskianalyysi vastaa lopullista tuotetta, riskit on huomioitu ja ovat riittävän pienellä tasolla. (Mts. 6.)

## **6 Liukurenkaan tarkastuslaitteen sähkösuunnittelu**

### **6.1 Suunnitteluohjelmisto**

Opinnäytetyönä tehtävän tarkastuslaitteen sähkösuunnittelu tehtiin EPLAN Electric P8 -sähkösuunnitteluohjelmistolla. Ohjelmisto on tietokantapohjainen ympäristö, jonka eri moduulit keskustelevat toistensa kanssa EPLAN Platform -alustatekniikan ansiosta. Esimerkkejä eri moduuleista ovat esimerkiksi jo mainittu Electric P8 sekä EPLAN Fluid, joka on hydrauliiikan ja pneumatiikan ohjaus- ja suunnitteluohjelmisto. Kaikki moduulit käyttävät samaa tietokantaa, joten sekaannuksia esimerkiksi vanhentuneiden kanssa ei tule. (Eplan Electric P8 2010.)

Piirikaavioiden lisäksi ohjelmistolla suunniteltiin asennuskotelon layout, koteloiden rei'itykset sekä generoitiin kokoonpanoon sekä asiakkaalle toimitettavat osa- ja kaapeliluettelot.

EPLAN on saksalainen ohjelmistotuottaja, Rittalin tytäryhtiö ja kuuluu Friedhelm Loh -konserniin. Yritys on toiminut kansainvälisesti yli 25 vuoden ajan ja sillä on yli 18 000 asiakasta ja yli 60 000 asennusta maailmanlaajuisesti. (Eplan Electric P8 2010.)

## 6.2 UL-hyväksyntä

Liukurenkaan tarkastuslaitteen sähkösuunnittelu aloitettiin heti riskianalyysin valmistuttua. Ensimmäisenä tehtiin laitevalinnat sekä selvitettiin käyttöjännite, jolla laitteisto tullaan sähköistämään. Laitevalintoja tehdessä yksi merkittävä seikka oli se, että laitetta tullaan käyttämään Pohjois-Amerikassa. Laitteistoon valittiin komponentteja, joilla on joko UL Listed- tai UL Recognized -merkintä. Myös kaapelien valinnassa otettiin huomioon se, että kaapelista on saatavilla Pohjois-Amerikan markkinoille hyväksytty versio.

Underwriters Laboratories on riippumaton kansainvälinen yritys, joka testaa ja myöntää sertifikaatteja turvallisiksi toteamilleen laitteille ja komponenteille (About UL 2013). Opinnäytetyön kannalta kaksi tärkeintä UL:n myöntämää merkintää on esitetty kuviossa 11.



**Kuvio 11. UL listing- ja Recognized component -merkinnät (Industrial control panels for the North American Market 2010, 35, muokattu)**

UL Listing -merkintä on UL-merkinnöistä yleisin. Merkki tarkoittaa sitä, että laite johon se on kiinnitetty, on läpäissyt UL:n asettamat turvallisuusvaati-

mukset. Recognized component -merkintää käytetään komponenteissa, joita käytetään suuremmissa laitekokonaisuuksissa. Yksittäisinä tällaiset komponentit voivat olla rakenteeltaan puutteellisia tai suorituskyvyltään rajoitettuja. (Industrial control panels for the North American Market 2010, 35.)

Yleisenä ohjeena UL-merkinnöistä voidaan todeta, että UL listing -merkinnällä varustettu laite on kokonaisuus, joka voidaan asentaa ilman erikoistyykaluja ja ohjeistusta. Recognized component -merkintää käytetään komponenteissa, jotka kootaan tuotantolaitoksessa toimivaksi kokonaisuudeksi. (Mts. 37–38.)

UL-hyväksyntä helpotti laitevalintaa siinä mielessä, että jos laitteella ei merkintää ollut, jätettiin se huomioimatta. Toisaalta laitevalinta oli merkinnän takia haastavaa, sillä Euroopan markkinoilla olevista tuotteista ei kyseistä merkintää välttämättä löydy. Lisäksi monilla laitevalmistajilla UL-hyväksytyillä laitteilla ja komponenteilla oli pitkähköt toimitusajat, joka tuotti ongelmia liukurenkaan tarkastuslaitteen kokoonpanossa. Hyväksytyjen laitteiden valikoima saattoi olla myös niin suppea, ettei sopivaa komponenttia löytynyt laitetoimittajan valikoimista. Näistä syistä johtuen komponentit valittiin suurilta valmistajilta, kuten ABB ja Rittal. Suurten valmistajien valikoimista löytyivät sopivat komponentit tarvittavilla hyväksynnöillä.

UL-hyväksytyjä laitteita ja komponentteja päädyttiin käyttämään varmuuden vuoksi. Asiakkaalta ei tullut vaatimuksia käytettävien komponenttien suhteen, vaan vaatimukset asetettiin itse. Näin meneteltiin sen takia, ettei laitetta tarvitsisi ryhtyä loppusijoituspaikassa korjailemaan ja vaihtelemaan komponentteja, jos asiakas haluaakin käyttää hyväksytyjä komponentteja. Myös mahdollisesti tarkastuslaitteistosta lähtevän tulipalon tai tarkastuslaitteiston aiheuttaman työtaturman takia päädyttiin käyttämään Pohjois-Amerikan markkinoille hyväksytyjä komponentteja.

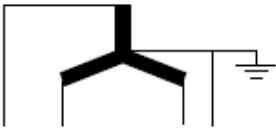

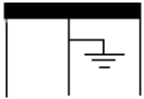
### 6.3 Käyttöjännite, ohjausjännite sekä sähkön syöttö

Yhdysvalloissa on käytössä useita erilaisia sähköverkkojärjestelmiä. Kolmi- vaiheiset, nelijohtimiset järjestelmät joiden pääjännitteet ovat 240 V ja 480 V ovat kuitenkin tärkeimmät käytössä olevat. Yksivaiheiset 120 V ja 240 V järjestelmät ovat lähinnä kotitalouskäytössä. (Industrial control panels for the North American market 2010, 28.)

Taulukossa 1 on esitetty Yhdysvalloissa käytössä olevien sähköverkkojen päätyypit.

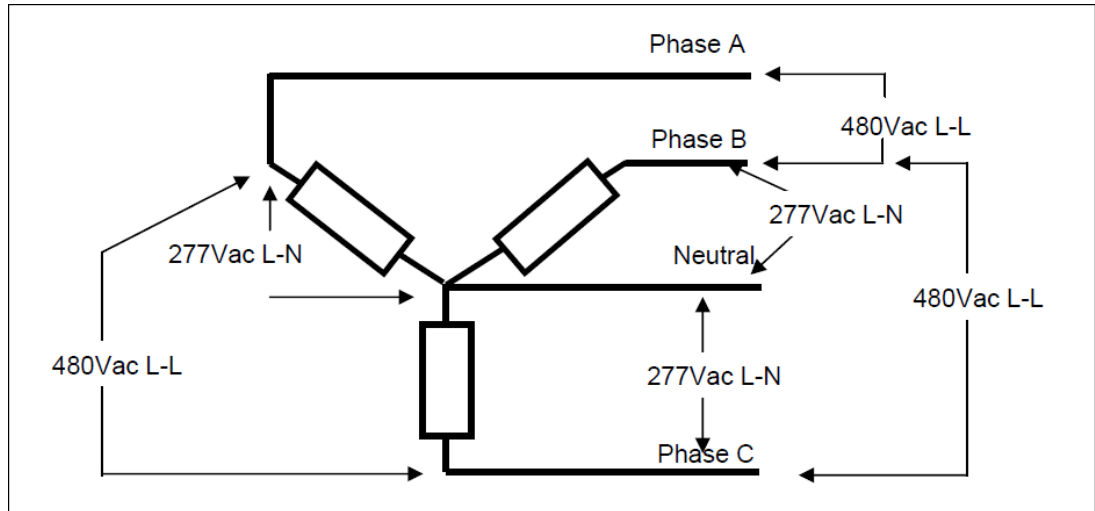
**Taulukko 1. Yhdysvalloissa käytettävät sähköverkot (Industrial control panels for the North American Market 2010, 28)**

#### Principal types of networks

Industrial and commercial buildings (Industry and Commercial)		Residential buildings (Residential)
		
3 phase, 4 wire	3 phase, 3 wire	1 phase, 3 wire
solidly grounded wye <b>Caution:</b> The PE must not carry any current. There is no PEN conductor => N = grounded conductor (white or gray); separate conductors must be used for PE and N.	corner grounded Delta	Single Phase, 120 / 240 V, grounded midpoint. <b>Example:</b> 120 V are used on the wall receptacle and 240 V for household cookers, air-conditioning units and laundry dryers.
208Y / 120 V	240 V	240 V to phase conductor
240Y / 131 V	480 V	120 V to ground
480Y / 277 V	600 V	
600Y / 347 V		

Y describes the "solidly grounded circuit". For example, the voltage between the phases is 480 V but that between the phase and grounding is just 277 V.

Asiakkaalta kysyttiin, minkälainen sähköverkko heillä on tuotantolaitoksessaan käytettävissä. Asiakkaan vastauksen mukaan heillä on saatavilla sekä 480Y / 277 V- että 208Y / 120 V -järjestelmät. Kuviossa 12 on esitetty 480Y / 277 V -järjestelmän rakenne.



**Kuvio 12. 480Y / 277 V -sähköjärjestelmän rakenne. (Common AC distribution configurations 2010)**

Laitehankinnat tehtiin sillä oletuksella, että tarkastuslaitteisto sähköistetään 208Y / 120 V -järjestelmällä. Projektin edetessä asiakas kuitenkin ilmoitti, että laitteiston sähköistykseen tullaan käyttämään 480Y / 277 V -järjestelmää. Järjestelmän 277 V:n vaihejännitteen takia laitteistoon lisättiin ulkoinen kolmi-vaihemuuntaja, jolla 480 V:n pääjännite muutetaan 200 V:n pääjännitteeksi. Näin vaihejännitteeksi saatiin tarvittava 115 V. Alhaisesta käyttöjännitteestä johtuen laitteiston virta kasvaa suureksi, tästä syystä syöttö toteutettiin kolmivaiheisena. Laitteiston arvioidun virrankulutuksen perusteella muuntaja-valmistajalta kysyttiin tarjous laitteistoon soveltuvasta muuntajasta. Muuntaja yhdistettiin sähkökaappiin suojatulla kaapelilla ja järeällä liittimellä. Pienempikin liitin olisi käytännössä riittänyt, mutta suurempi valittiin puhtaasti ulkonäöllisistä syistä. Valittu liitin on esitetty kuviossa 13.

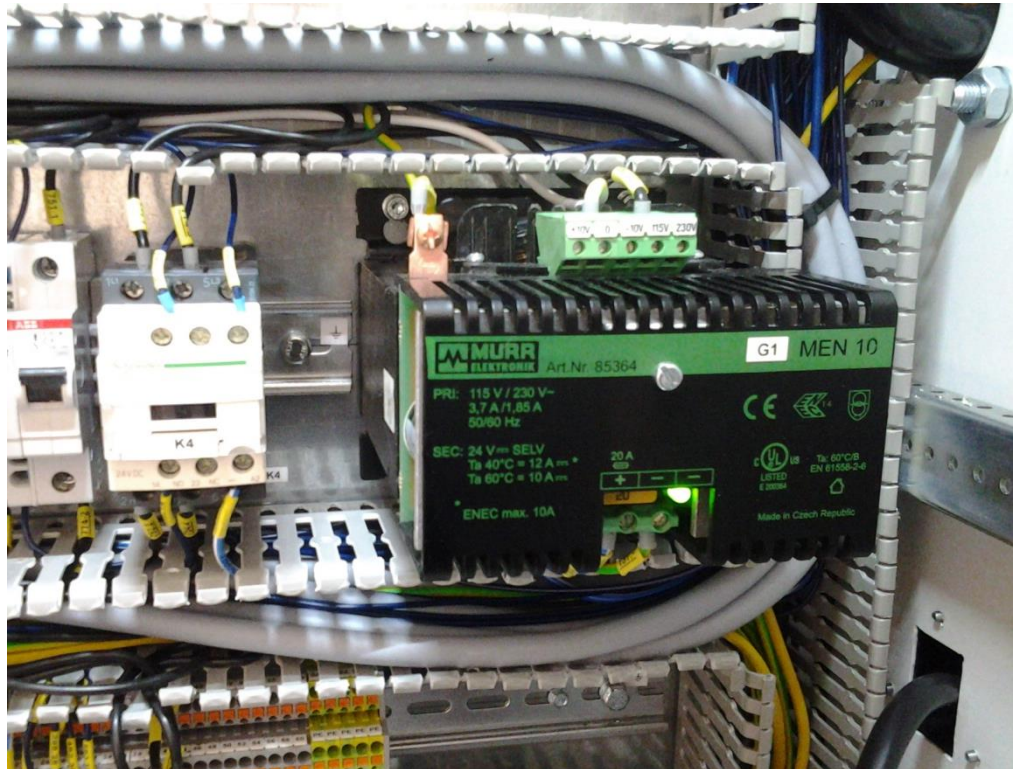




**Kuvio 13. Syöttökaapelin kytkentään käytetty liitin**

Vaiheita päädyttiin kuormittamaan siten, että vaiheella A sähköistetään sähkökaappi, vaiheella B huolehditaan magnetisoitvahvistimen virran tarpeesta sekä vaiheella C sähköistetään kaksi jäähdytintä, tietokone sekä mittalaite. Manuaaleista laskettuna vaiheen A kuormitukseksi saatiin kaiken kaikkiaan noin 18 A, vaiheita B ja C kuormitetaan molempia noin 16 A:n edestä. Todellisuudessa laitteisto ei kuormita yhtä vaihetta 18 A:lla, vaan kuormitus jää huomattavasti alhaisemmaksi. Sähkökaapissa olevia komponentteja olisi voitu syöttää kahdella vaiheella, näin olisi päästy symmetrisempään kuormitukseen. Tähän ei kuitenkaan ryhdytty, sillä käyttämällä yhtä vaihetta selkeästi yhteen tarkoitukseen saadaan järjestelmästä selkeä ja looginen. PC-kaapissa tarvittavat kaksi vaihetta sekä N-johdin päätettiin kuljettaa samassa kaapelissa. Suureksi kasvava virta oli huomioitava kaapelin valinnassa.

Tarkastuslaitteiston 24 V DC -ohjausjännite toteutettiin 10 A:n muuntajalla, joka on esitetty kuviossa 14.



**Kuvio 14. Laitteistossa käytetty muuntaja**

Ohjausjännitteen jakoon komponenteille käytettiin kuvion 15 älykästä kuormituspiiriä joka toimii myös ylivirtasuojana. Suurin yksittäinen 24 V DC -jännitteen kuluttaja on askelmoottorinohjain, joten yksi kuormituspiirin lähdöistä varattiin kokonaan askelmoottorinohjaimelle. Kuormituspiirin muilla lähdöillä tuotettiin servovahvistimien, logiikan sekä hätäpysäytys-piirien tarvitsemat ohjausjännitteet.



Kuvio 15. Älykäs kuormituspiiri ja ylivirtasuojaja

## 6.4 Kaapelointi ja johdonsuojaus

Tarkastuslaitteen sisäisessä johdotuksessa sekä kaapeloinnissa ohjeistuksena käytettiin SFS-EN 60204-1 -standardia, koska paremmin Pohjois-Amerikan markkinoilla käytettävää NFPA79 -standardia ei ollut saatavilla. NFPA79 on Pohjois-Amerikan teollisuuskoneiden sähköstandardi, joka koskee sähkölaitteita, jotka toimivat korkeintaan 600 V jännitteellä (Lapp Kabel muuttuneet UL-määräykset NFPA79-2007 ja -2012 2010). Käytännössä standardit SFS-EN 60204-1 ja NFPA79 ovat sisällöltään samanlaiset, eroavaisuuksia on lähinnä värikoodauksissa sekä johdinkoossa. Tästä johtuen 60204-1 -standardia voitiin käyttää ohjeistuksena. Tarvittavat täsmennykset värien ja johtimien kokojen osalta tarkistettiin Siemensin tekemästä UL-oppaasta Industrial control panels for the North American market.

PC-kaapin yhdistäminen muuhun laitteistoon tehtiin kaapeleilla. PC-kaapin sisälle kaapin kylkeen asennettiin kytkentärasia ja rasian kohdalle kaapin sei-

nään kolme liittintä. Näin PC-kaapille tuotavat kaapelit voidaan tarvittaessa irrottaa ja kytkeä nopeasti esimerkiksi asennusvaiheessa. Liittimet, jotka on esitetty kuviossa 16, koodattiin, joten virheellisen kytkennän mahdollisuus pieneni olemattomaksi. Kaapelit asennettiin vielä kaapelinsuojaputkiin mekaanisen kulumisen minimoimiseksi.



**Kuvio 16. Liittimet PC-kaapin kyljessä**

Laitteistossa käytetyt kaapelit ovat väyläkaapeleita lukuun ottamatta AWG-kaapelia (American Wire Gauge), jonka merkintä poikkeaa huomattavasti siitä mihin Euroopassa on totuttu. Taulukossa 2 on esitetty AWG-johtimien muunnontaulukko.

Taulukko 2. AWG-johtimien muuntotaulukko ([http://www.tnt-audio.com/clinica/awg\\_e.html](http://www.tnt-audio.com/clinica/awg_e.html))

Conversion table - American Wire Gauge - mm. - mm <sup>2</sup>						
AWG N°	Diam. mm.	Area mm <sup>2</sup>		AWG N°	Diam. mm.	Area mm <sup>2</sup>
1	7,350	42,400		16	1,290	1,3100
2	6,540	33,600		17	1,150	1,0400
3	5,830	26,700		18	1,024	0,8230
4	5,190	21,200		19	0,912	0,6530
5	4,620	16,800		20	0,812	0,5190
6	4,110	13,300		21	0,723	0,4120
7	3,670	10,600		22	0,644	0,3250
8	3,260	8,350		23	0,573	0,2590
9	2,910	6,620		24	0,511	0,2050
10	2,590	5,270		25	0,455	0,1630
11	2,300	4,150		26	0,405	0,1280
12	2,050	3,310		27	0,361	0,1020
13	1,830	2,630		28	0,321	0,0804
14	1,630	2,080		29	0,286	0,0646
15	1,450	1,650		30	0,255	0,0503

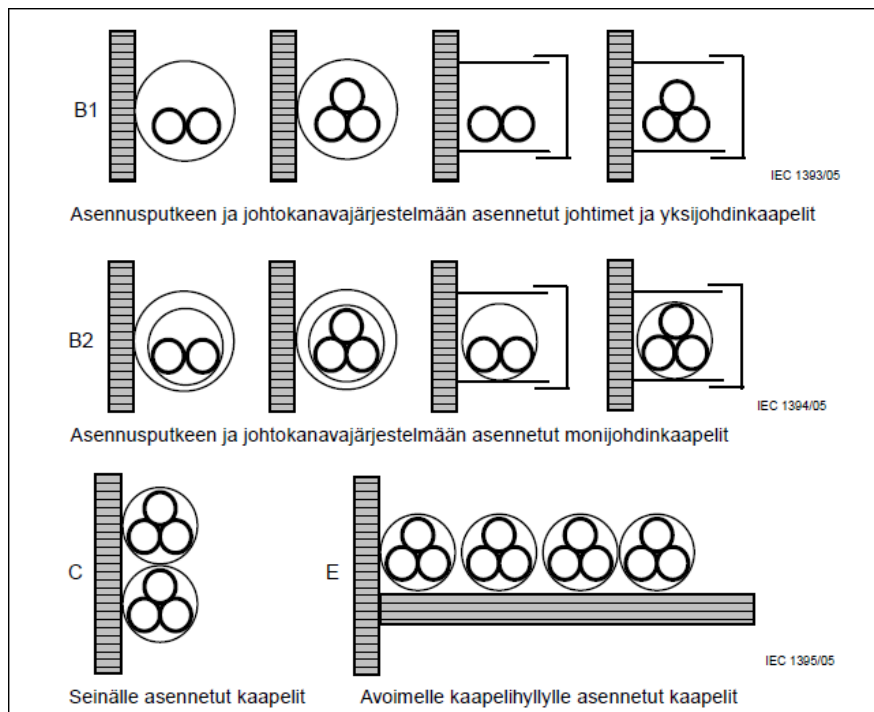
Tarkastuslaitteistoon tilattiin väyläkaapelin lisäksi kolmea erilaista kaapelia. Signaalikaapeliksi AWG 18 -johtimilla varustetut kaapelit todettiin riittäväksi, samoin toisen syöttökaapelin AWG 14 -johtimet olivat virrankestävyydeltään riittävät. Syöttökaapeliksi, jolla PC-kaapille tuotiin kaksi vaihetta ja N-johdin, valittiin kaapeli johdinkooltaan AWG 12, koska suuren virran takia AWG 14 -johdin ei ollut riittävän suuri.

Kaapeleiden valinnassa käytettiin apuna SFS-EN 60204-1 -standardia. Taulukossa 3 on esitetty kuormitettavuustaulukko, josta tarkistettiin kaapelin kuormitettavuus.

Taulukko 3. Kaapelin tai johtimen kuormitettavuus (SFS-EN 60204-1 2006, 124, muokattu)

	Asennusmenetelmä (ks. D.1.2)			
	B1	B2	C	E
Poikkipinta mm <sup>2</sup>	Kolmivaiheisen virtapiirin kuormitettavuus $I_z$ A			
0,75	8,6	8,5	9,8	10,4
1,0	10,3	10,1	11,7	12,4
1,5	13,5	13,1	15,2	16,1
2,5	18,3	17,4	21	22
4	24	23	28	30
6	31	30	36	37
10	44	40	50	52
16	59	54	66	70
25	77	70	84	88
35	96	86	104	110
50	117	103	125	133
70	149	130	160	171
95	180	156	194	207
120	208	179	225	240
Elektroniikka (parit)				
0,20	Ei sovelleta	4,3	4,4	4,4
0,5	Ei sovelleta	7,5	7,5	7,8
0,75	Ei sovelleta	9,0	9,5	10

Taulukon lisäksi oli käytettävä korjauskertoimia kuormitettavuuden selvittämiseksi, koska asennettavia kaapeleita/johdinpareja oli useita. Lämpötilakerrointa ei koettu järkeväksi muuttaa, vaikka laitteistoa ei todennäköisesti käytetä +40 °C tai korkeammassa lämpötilassa. Alhaisemmassa lämpötilassa kaapelia voitaisiin kuormittaa taulukoituja arvoja enemmän (SFS-EN 60204-1 2006, 182). Kuviossa 17 on esitetty standardin SFS-EN 60204-1 mukaiset kaapelien ja johtimien asennustavat.



Kuvio 17. Johtimien ja kaapelien asennustavat (SFS-EN 60204-1 2006, 184, muokattu)

Taulukossa 4 on esitetty asennustavoista aiheutuvat korjauskertoimet kaapelin tai johtimen kuormitettavuudelle.

Taulukko 4. Asennustavoista aiheutuvat korjauskertoimet (SF-EN 60204-1 2006, 184, muokattu)

Asennustavat (ks. kuva D.1) (ks. huom. 3)	Kuormitettujen virtapiirien/kaapelien lukumäärä			
	2	4	6	9
B1 (virtapiirit) ja B2 (kaapelit)	0,80	0,65	0,57	0,50
C yhteen kerrokseen asennetut kaapelit, kun kaapeleitten välissä ei ole ilmväliä	0,85	0,75	0,72	0,70
E yhdelle rei'itetylle kaapelihyllylle yhteen kerrokseen asennetut kaapelit, kun kaapeleitten välissä ei ole ilmväliä	0,88	0,77	0,73	0,72
E kuten edellä, mutta asennettuna 2...3 päällekkäiselle kaapelihyllylle, kun hyllyjen välinen etäisyys on 300 mm (ks. huom. 4)	0,86	0,76	0,71	0,66
Poikkipinnaltaan $\leq 0,5 \text{ mm}^2$ ohjausvirtapiirin johdinparit riippumatta asennustavasta	0,76	0,57	0,48	0,40

Kaapeleiden sopivuus arvioidulle virralle sekä valitulle johdonsuojakatkaisijalle tarkistettiin vielä taulukosta 5. Sulakkeen toimiessa kaapelin ainoana suojana, ovat sulakkeen koko ja kaapelin kuormitusvirta sidottuna toisiinsa, sillä

sulake suojaa kaapelia liialliselta lämpenemiseltä ylikuorman ilmetessä (Teknisiä tietoja ja taulukoita -käsikirja 2000, 8).

**Taulukko 5. Kaapelin ylikuormasuojana toimivan sulakkeen suurin nimellisvirta (Teknisiä tietoja ja taulukoita -käsikirja 2000, 8)**

Johdon sallittu kuormitusvirta vähintään / A	Suojaavan sulakkeen tai L-tyyppisen johdosuojakytkimen suurin sallittu nimellisvirta / A
14	10
20	16
25	20
32	25
41	35
58	50
73	63
93	80
116	100
146	125
185	160
232	200
292	250
366	315
464	400
583	500
733	630
930	800
1170	1000
1460	1250

Kaapeleiden ja yksittäisten johtimien suojaukseen käytettiin yksinapaisia johdonsuojakatkaisijoita, joihin päädyttiin niiden vähäisen tilantarpeen takia. Tarkastuslaitteistoa ei tulla kytkemään asiakkaan sähköverkkoon pistotulppala, joten tilannetta jossa vaiheen sijaan katkottaisiin N-johdinta, ei pääse syntymään. Tämäkin seikka puolsi yksinapaisten johdonsuojakatkaisijoiden valintaa. Yksinapaiset johdonsuojakatkaisijat ovat myös hankintahinnaltaan edullisempia, joten tarkastuslaitteiston hinta saatiin pidettyä alhaisempana.



Käytettävät johdonsuojakatkaisijat valittiin laitevalmistajien manuaalien perusteella. Laitteen manuaalissa useimmiten kerrotaan tarvittavan etukojeen laukaisukäyrä sekä nimellisvirta. Tarkastuslaitteistossa käytettävien pistorasioiden sähkönsyöttö varustettiin lisäksi vikavirtasuojalla, jossa ei ole ylivirtasuojasta. Tähän ratkaisuun päädyttiin siksi, että myös ylivirtasuojana toimivilta vikavirtasuojakytkimiltä puuttuu tarvittava UL-hyväksyntä. Laitteistossa käytettiin C- ja K-käyrän johdonsuojakatkaisijoita, nimellisvirroiltaan 6–16 A.

Sähkökaapissa ja PC-kaapin kytkentärasiaassa käytettyjen johtimien värit tarkistettiin Siemensin oppaasta Industrial control panels for the North American market. Käytetyt värit on esitetty taulukossa 6.

**Taulukko 6. Johtimien värit (Industrial control panels for the North American market 2010, 202)**

Color	Conductor	Remark
Green with or without one or more yellow stripes	Grounding conductor May only be energized in an emergency (not a PEN conductor)	Green/yellow is permitted in accordance with IEC 60204-1.
Black	All ungrounded control circuit conductors operating at the supply voltage	
Red	All ungrounded AC control circuit cables operating at a voltage less than the supply voltage	
Blue (dark blue)	Ungrounded DC control circuit cables	
Yellow or orange	Ungrounded control circuit cables which are energized when the main disconnect is in the "off" position.	In IEC 60204-1, orange is used for this purpose.
White or gray or three white stripes on conductors, but NOT on green, blue, orange or yellow	Grounded AC current-carrying control circuit conductor regardless of voltage	
White with blue stripes	Grounded DC current-carrying conductor	In IEC 60204-1, light blue is used for this purpose.
White with orange or yellow stripes	Grounded AC current-carrying conductor that remains energized when main disconnect switch is in the "off" position.	

## 6.5 Servomootoreiden ja askelmoottorin turvakytkennät

Koneen ohjausjärjestelmän on pysäytettävä kone ja varmistuttava siitä, että kone pysyy pysähdyksissä. Tämä koskee erityisesti pystyakseleita, joissa ei ole automaattista lukitusmekanismia tai vastapainoa. Servo-ohjaimet sekä askel-

moottorin ohjain tukevat sekä Safe Torque Off (STO)- että Safe Stop 1(SS1) - turvatoimintoja, jotka suojaavat odottamattomalta käynnistymiseltä standardin EN 61508 SIL2 sekä EN ISO 13849-1 PL d mukaan. (Assembly and installation Type CMMS-AS 2010, 71; Assembly and installation Type CMMS-ST 2010, 79.)

### **Safe Torque Off**

Safe Torque Off -turvatoiminnolla katkaistaan jännitesyöttö servovahvistimen pääteasteelta sekä tämän ohjaukselta. Seurauksena moottorin sähkösyöttö katkaistaan, eikä moottori enää kykene tuottamaan voimaa ja toteuttamaan vaarallisia liikkeitä. Jos STO-turvatoiminto aktivoidaan moottorin käydessä, moottori alkaa hidastua hallitsemattomasti. Jos moottori on varustettu jarrulla, jarru aktivoidaan. Jokainen STO-turvatoiminnon aktivoiminen rasittaa moottorin jarrua, joten tätä toimintoa tulisi käyttää ensisijaisesti jarruttomilla moottoreilla. (Assembly and installation Type CMMS-AS 2010, 73.)

### **Safe Stop 1**

Safe Stop 1 -turvatoiminto eroaa STO:sta siten, että servomoottori pysäytetään hallitusti, jonka jälkeen servovahvistimen pääteasteen jännitesyöttö katkaistaan. Näin moottori ei kykene tuottamaan voimaa ja vaarallisten liikkeiden toteuttaminen ehkäistään. (Mts. 79.)

Tarkastuslaitteiston servomoottoreissa käytetään SS1-turvatoimintoa. SS1-turvatoiminto valittiin käytettäväksi, koska servomoottorien liikuttamisessa liukujohteissa on kiinnitettynä suurehkoja massoja. Lisäksi suurin moottoreista on varustettu jarrulla, joka estää STO-turvatoiminnon käytön. Askelmoottorissa sen sijaan käytetään STO-turvatoimintoa. Turvatoimintojen kytkennät on esitetty liitteessä 1. Turvareleen kuittaustoiminto toimii siten, että kuittaussignaali kiertää jokaisessa moottorinohjaimessa olevan koskettimen kautta palaten takaisin turvareleelle.

## 6.6 Sähkökaapin layout

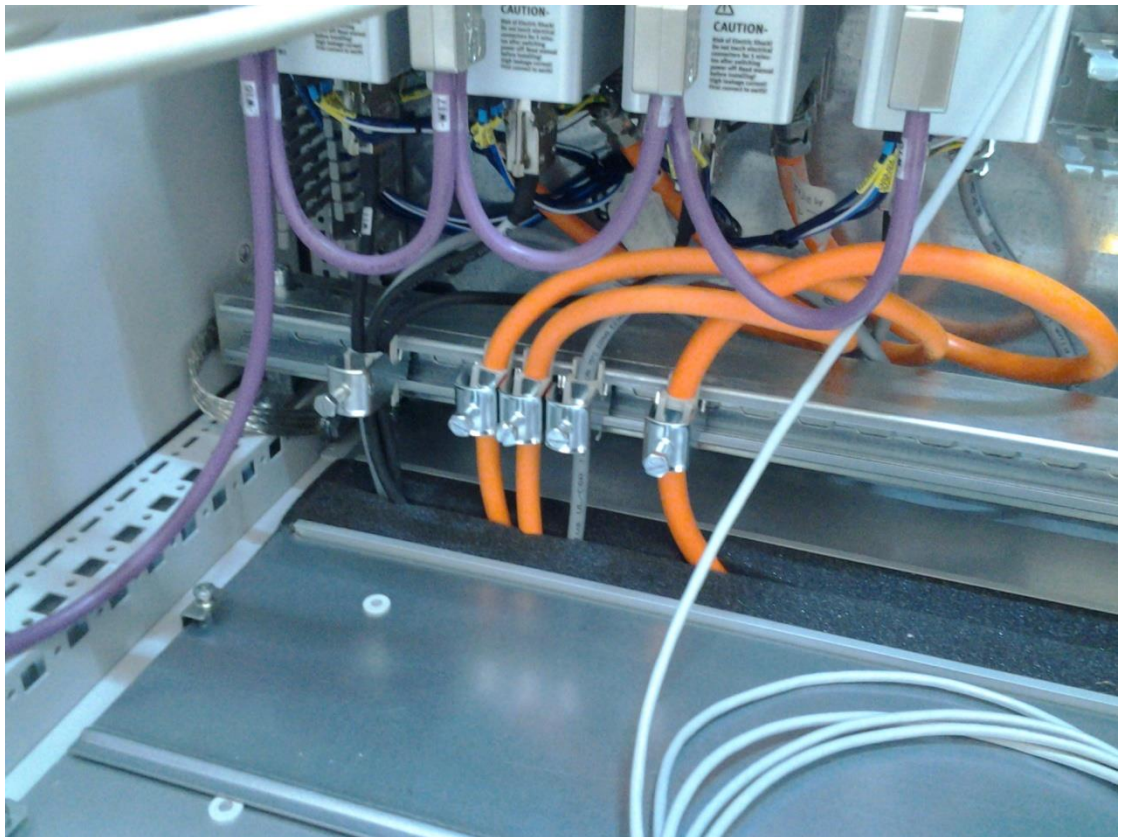
Tarkastuslaitteiston sähkökaapin layoutsuunnittelu aloitettiin, kun suurin osa laitteistossa käytettävistä komponenteissa oli selvillä. Alustavaa suunnittelua ja hahmottelua oli tehty arvioiden perusteella, jotta sähkökaapin kokoluokka selviäisi. Kaapista ei haluttu liian ahdasta. Sähkökaapin layout on esitetty liitteessä 2.

Ohjelmoitava logiikka IO-yksiköineen sijoitettiin sähkökaapissa ylimmäiseksi. Ratkaisuun päädyttiin tilankäytön tehostamiseksi, sillä logiikassa ainoastaan syöttöjännite syötetään laitteen alapuolelta, IO-yksiköiden kytkennät tehdään yksiköiden etupuolelle. Laitteen yläpinnasta ei lähde kytkentöjä, joten se voitiin asentaa lähelle sähkökaapin kattoa, ottaen huomioon riittävän ilmaraon jäähdytystä silmällä pitäen. DC-moottorin ohjainkortti sijoitettiin ohjelmoitavan logiikan viereen kaapin yläosaan.

Riviliitinryhmät jaoteltiin käyttötarkoituksen mukaan. Servo-ohjaimien IO-kaapelit kammattiin omiin riviliitinryhmiin, samoin meneteltiin turvareleen ovikytkimen kanssa. Sähkökaapista poistuvat kaapelit eli kaapelit, jotka menevät esimerkiksi PC-kaappiin, koottiin yhteen riviliitinryhmään kaapin alaosaan lähelle läpivientiaukkoja sekä vedonpoistokiskoa. Ohjausjännitteen jako tehtiin tehonjakajalla sekä sen lähtöihin liitetyillä riviliitinryhmillä. Näin kaikki laitteet eivät ole yhden ylivirtasuojan takana. Laitteistossa tarvittava 0 V DC jaettiin myös keskitetysti yhdestä riviliitinryhmästä. Jokaiseen riviliitinryhmään jätettiin tyhjiä riviliittimiä laajennusvaraksi lukuun ottamatta servo-ohjaimien IO-kaapeleiden riviliitinryhmiä.

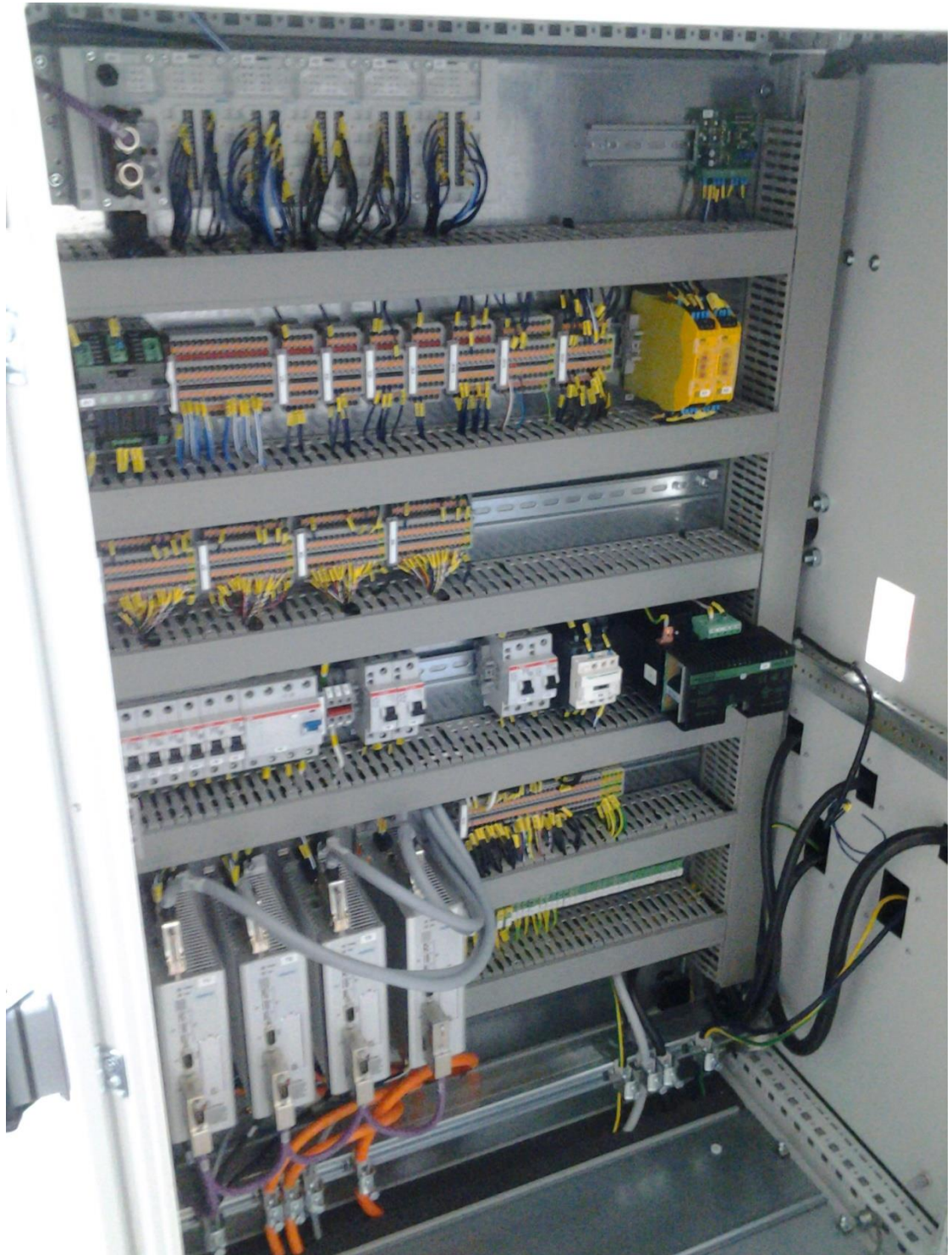
24 V:n ohjausjännitteen lisäksi sähkökaappiin tuodaan 115 V AC - käyttöjännite. Käyttöjännitteeseen kytkettävät laitteet, kuten johdonsuojakatkaisijat ja servo-ohjaimet, keskitettiin samalle puolelle kaappia. Tällä järjestyksellä päästiin tilanteeseen, jossa ohjaus- ja käyttöjännitteet kulkevat erillään toisistaan.

Servo-ohjaimet sijoitettiin kaapin alaosaan. Sijoituspaikka määräytyi ohjaimien suuren lämmöntuoton sekä ohjaimista lähtevien, jäykkien kaapeleiden takia. Sähkökaapista lähtevien kaapeleiden läpiviennit sijoitettiin sekä kaapin kylkeen että pohjalevyyn. Kylkeen sijoitettujen läpivientien kautta kaapelit viedään PC-kaappiin ja kaapin pohjalevyn läpi vedettävät kaapelit menevät koneen rungolla sijaitseville laitteille, kuten servomootoreille. Myös PC-kaapista koneen rungolle kulkevat kaapelit kulkevat sähkökaapissa olevan vedonpoistokiskon kautta. Kuviossa 18 on esitetty servokaapeleiden vedonpoisto sekä läpivienti pohjalevyn läpi.



**Kuvio 18. Vedonpoistokisko ja kaapelien läpivienti pohjalevyn läpi**

Kuviossa 19 on esitetty valmis sähkökaappi.



**Kuvio 19. Valmis sähkökaappi**

## 6.7 Käyttäjärajaus

Tarkastuslaitteen käyttöliittymästä haluttiin mahdollisimman yksinkertainen ja helposti omaksuttava. Kytкимиä ja painonappeja tuli käyttää mahdollisimman vähän ja laitteisto oli automatisoitava niin pitkälle, ettei se vaadi käyttäjältä mitään toimenpiteitä mittaussekvenssin aikana. Käyttäjä ainoastaan kiinnittää mitattavan kappaleen laitteistoon ja käynnistää sekvenssin, laitteisto hoitaa itse loput.

Standardia SFS-EN 60204-1 käytettiin apuna suunniteltaessa ohjauspaneelia. Taulukossa 7 on esitetty kyseisen standardin määräämät painikkeiden värit.

**Taulukko 7. Standardin sallimat painikkeiden värit (SFS-EN 60204-1 2006, 106)**

Väri	Merkitys	Selitys	Sovellutusesimerkkejä
PUNAINEN	Hätä	Vaikuttaminen vaarallisessa tilanteessa tai hätätilanteessa	Hätäpysäytys Hätätoiminnon käynnistys (ks. myös 10.2.1)
KELTAINEN	Normaalista poikkeava	Vaikuttaminen normaalista poikkeavissa tilanteissa	Normaalista poikkeavan tilanteen lopettaminen Keskeytyneen automaattisen työkierron käynnistäminen
SININEN	Pakollinen	Vaikuttaminen pakollista toimintaa edellyttävissä tilanteissa	Kuittaustoiminto
VIHREÄ	Normaali	Vaikuttaminen normaalitilanteen käynnistämiseksi	(Ks. 10.2.1)
VALKOINEN			KÄYNNISTYS/PÄÄLLÄ (suositeltava) SEIS/POIS
HARMAA	Ei määriteltyä erikoismerkitystä	Yleensä toimintojen käynnistäminen lukuun ottamatta hätäpysäytystä (ks. huomautus)	KÄYNNISTYS/PÄÄLLÄ SEIS/POIS
MUSTA			KÄYNNISTYS/PÄÄLLÄ SEIS/POIS (suositeltava)
HUOM. Kun painikkeen tunnistamiseksi käytetään täydentäviä koodauskeinoja (esim. pintakäsittely, muoto tai sijainti) saa samaa väriä VAKOINEN, HARMAA tai MUSTA käyttää eri toiminnolle (esim. VAKOINEN: KÄYNNISTYS/PÄÄLLE- ja SEIS/POIS-painikkeille).			

Tarkastuslaitteiston käyttäjärajauskuuluu painikkeiden lisäksi merkkivaloja sekä merkkivalopainikkeita. Merkkivaloja sijoitettiin ohjauspaneelin lisäksi valopylväeseen. Standardi SFS-EN 60204-1 määrää myös merkkivaloissa käytettävät värit. Värit on esitetty taulukossa 8.

**Taulukko 8. Standardin sallimat värit merkkivaloissa (SFS-EN 60204-1 2006, 108)**

Väri	Merkitys	Selitys	Käyttäjän toimenpide
PUNAINEN	Hätä	Vaarallinen tila	Vaarallisen tilanteen edellyttämä välitön toiminta (esim. koneen syötön pois kytkeminen, vaarallisen tilan huomaaminen, koneesta erillään pysyminen)
KELTAINEN	Normaalista poikkeava	Normaalista poikkeava tila Kriittisen tilan uhka	Valvonta tai toimintaan puuttuminen (esim. tarkoitetun toiminnon uudelleen asettelu)
SININEN	Pakollinen	Käyttäjän toimintaa vaativan tilan ilmaisu	Pakollinen toiminta
VIHREÄ	Tavanmukainen	Tavanmukainen tila	Vaihtoehtoinen
VALKOINEN	Neutraali	Muut tilat: voidaan käyttää, kun PUNAISEN, KELTAISEN, VIHREÄN tai SINISEN soveltuvuus on epäselvä	Valvonta

Painonappien ja merkkivalojen lisäksi tarkastuslaitteiston ohjauspaneelissa on kaksi valintakytkintä ja yksi avainkytkin. Valintakytkimiä käytetään sekä käsiäjön että mittaussekvenssin valintaan. Laitteiston käyttäjällä ei ole tarvetta käsiäjölle laitteiston toimiessa normaalisti. Käsiäjo onkin tarkoitettu vain virhe- ja opetustilanteita varten, tästä johtuen käsiäjon valinta on aktivoitava avainkytkimellä. Tarkastuslaitteiston ohjauspaneelin layout on esitetty liitteessä 3.

Tarkastuslaitteistossa on kaksi hätäpysäytys-painiketta. Toinen painike asennettiin ohjauspaneeliin, toinen sijaitsee pääkytkimen vieressä sähkökaapin kyljessä. Pääkytkintä voidaan käyttää syötön hätäpois-kytkentään. Taulukon 7 (s. 43) perusteella pääkytkimen ja hätäpysäytys-painikkeiden väreiksi määräytyivät punainen ja keltainen.

## 7 Liukurenkaan tarkastuslaitteen automaatio suunnittelu

### 7.1 Toiminnallinen kuvaus

Opinnäytetyönä tehtävän liukurenkaan tarkastuslaitteen automaatio suunnittelu käynnistyi yhtä aikaa sähkösuunnittelun kanssa. Automaatio suunnittelun ensimmäinen vaihe oli toiminnallisen kuvauksen laatiminen.

Toiminnallinen kuvaus määrittelee käyttäjien vaatimukset toteuttavan järjestelmän tai toiminnot, joiden on ajateltu vastaavan käyttäjän tarpeita. Toimintojen lisäksi kuvauksessa kuvataan toimintaympäristö sekä tiedot, joiden perusteella järjestelmä toimii. Toiminnallinen kuvaus on toimittajan laatima, käyttäjävaatimuksia tarkempi vastaus vaadittujen toimintojen toteuttamiseksi. Toiminnallinen kuvaus muodostaa pohjan järjestelmän suunnittelulle. (Laatu automaatiossa 2001, 157.)

Liukurenkaan tarkastuslaitteen toiminnallinen kuvaus laadittiin sellaiseksi, että siitä olisi mahdollisimman paljon hyötyä useassa laitteiston valmistuksen vaiheessa. Toiminnallisen kuvauksen avulla testauksesta vastaavat henkilöt voivat tutustua laitteiston toimintaan ja kykenevät antamaan vinkkejä, mikäli havaitsevat epäkohtia sekvenssissä. Logiikkaohjelmointi perustuu täysin toiminnalliseen kuvaukseen ja ne kehittyvätkin samanaikaisesti lopulliseen muotoonsa. Ohjelmointia tehtäessä toiminnallista kuvausta tulee tarkasteltua kriittisesti. Epäkohdat voidaan korjata ja näin toiminnallisen kuvauksen päivitys tapahtuu kuin itsestään. Laadittaessa laitteiston käyttöohjeita hyvin tehty toiminnallinen kuvaus on arvossaan. Toiminnallinen kuvaus on liitteenä 4.



## 7.2 Ohjelmointiympäristö

### IEC 61131-3

IEC 61131-3 on ensimmäinen todellinen pyrkimys yhtenäistää teollisuusautomaation ohjelmointikieliet. Se on maailmanlaajuinen ja täysin laitevalmistajista riippumaton. IEC 61131-3 kuuluu kahdeksanosaiseen IEC 61131 - standardiperheeseen ollen sen kolmas osa. Standardi voidaan jakaa kahteen osaan, joista toisessa määritellään ohjelmointikieliet (Programming languages) ja toisessa yleiset elementit (Common elements). (IEC 61131-3: a standard programming resource n.d.) IEC 61131-3 määrittelee ohjelmoitavien ohjauslaitteiden ohjelmointikielien syntaksin ja semantiikan (Introduction into IEC 61131-3 Programming Languages n.d). Syntaksi tarkoittaa kielen rakennetta, se määrittelee säännöt joiden perusteella kielen lailliset ilmaukset voidaan muodostaa. Semantiikka määrittelee näiden ilmausten merkityksen. (Ohjelmointikielten syntaksista ja semantiikasta n.d.)

### Standardin määrittelemät ohjelmointikieliet

IEC 61131-3 määrittelee viisi ohjelmointikieltä, kaksi tekstipohjaista ja kolme graafista. Tekstipohjaiset ohjelmointikieliet ovat IL (Instruction List) ja ST (Structured Text). Graafiset ohjelmointikieliet ovat LD (Ladder Diagram) ja FBD (Function Block Diagram) sekä SFC (Sequential Function Chart). (IEC 61131-3: a standard programming resource n.d.)

IL (Instruction List) on assembler-tyylinen ohjelmointikieli. Se perustuu rekisterien tehokkaaseen käyttöön. (Feston CoDeSys koulutusmateriaali 2011). Kuviossa 20 on esitetty esimerkki IL-käskylistan käytöstä.

```

LD      S2
OR      (K1
ANDN    S1
)
ST      K1

LD      S2
S       K1

LD      S1
R       K1

```

Kuvio 20. Esimerkki IL-käskylistan käytöstä

ST (Structured Text) on PASCAL-ohjelmointikielen kaltainen lausekielinen ohjelmointikieli. Käskyrakenteet, kuten IF, WHILE ja CASE, ovat ideaalisia ST:llä ohjelmoitavia rakenteita. (Mt.) Kuviossa 21 on esitetty esimerkki ST-ohjelmointikielestä.

```

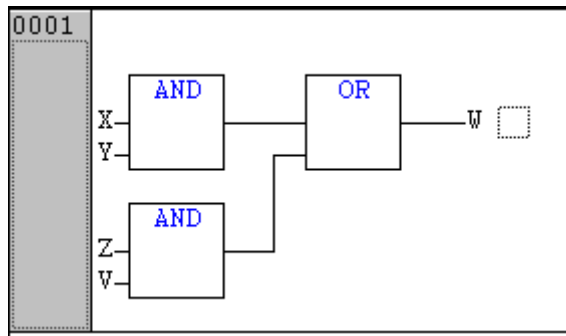
0001 PROGRAM RecordmodePositionControl
0002 VAR
0003     iStep      : INT := -1;
0004 END_VAR
0005
0001 CASE iStep OF
0002
0003 -1: IF pCanOpenMaster[0].nStatus = 5 AND pCanOpenNode[0].nStatus = 5 THEN
0004     iStep := 0;
0005     END_IF
0006
0007 0: Drive.Halt := TRUE;
0008     Drive.Stop := TRUE;
0009     Drive.EnableDrive := TRUE;
0010     IF Drive.Ready AND Drive.DriveEnabled THEN
0011         iStep := 10;
0012     END_IF
0013
0014 10: Drive.StartHoming := TRUE;
0015     IF Drive.DriveIsReferenced THEN
0016         Drive.StartHoming := FALSE;
0017         iStep := 20;
0018     END_IF
0019

```

Kuvio 21. Esimerkki ST-ohjelmointikielestä

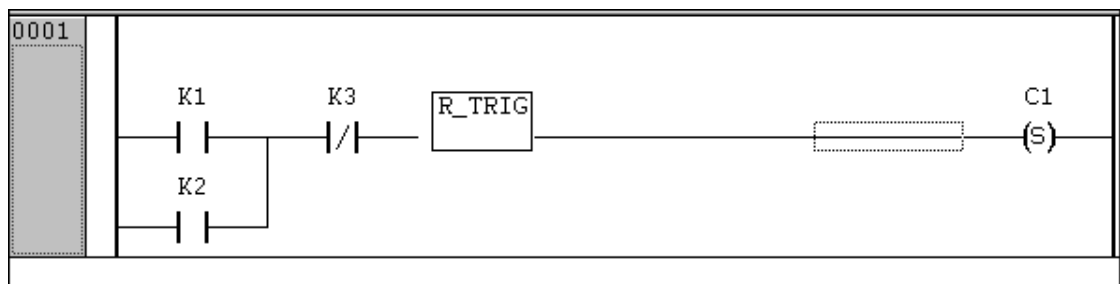
FBD (Function Block Diagram) on graafinen ohjelmointikieli Boolean- ja analogia-arvoille. FBD-editori CoDeSysissä tarjoaa nopean tavan ohjelmoida. Ohjelmaan lisätään automaattisesti halutun tyyppinen "laatikko" oikeilla tuloilla ja lähdöillä varustettuna. Laatikoiden välisiä yhteyksiä ei tarvitse piirtää erik-

seen, vaan ne syntyvät itsestään, kun ohjelmaan lisätään tuloja, lähtöjä tai operaatioita lähtöihin. (Mt.) Esimerkki FBD-kielestä on esitetty kuviossa 22.



Kuvio 22. Esimerkki FBD-ohjelmointikielestä

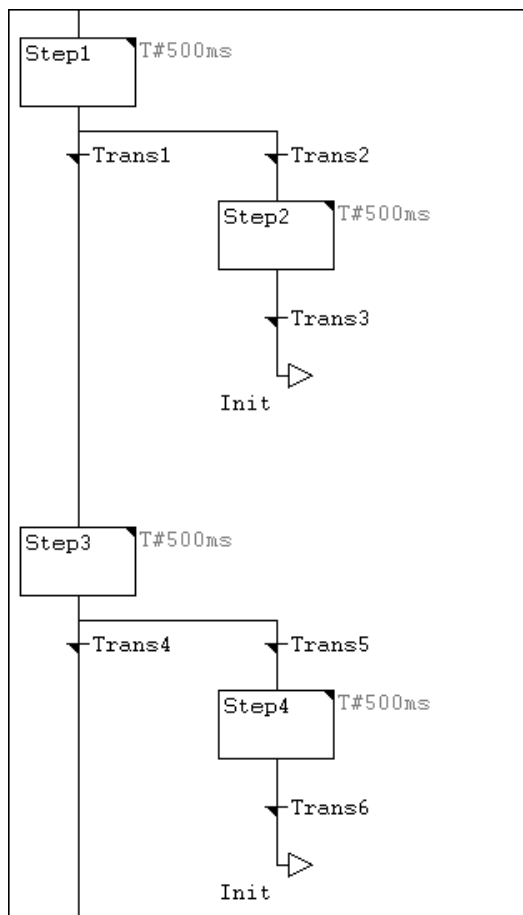
LD (Ladder) on kosketinkaavio, jolla relekontaktit esitetään samassa muodossa kuin sähkökaavioissa. (Mt.) Esimerkki LD:n käytöstä on esitetty kuviossa 23.



Kuvio 23. Esimerkki LD-ohjelmointikielestä

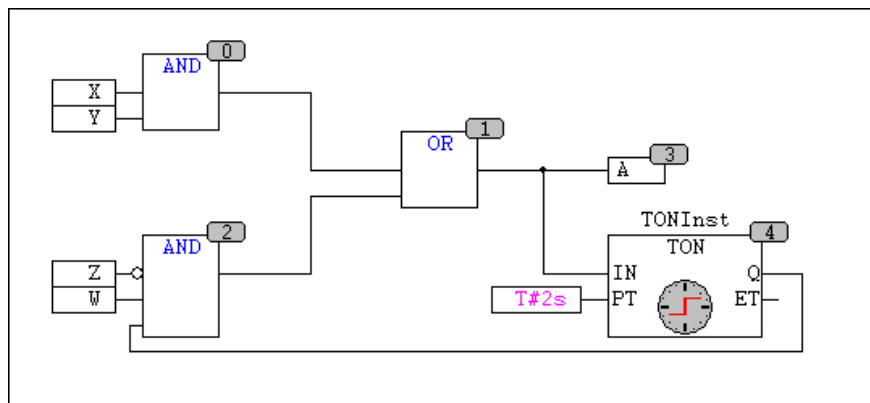
Sequential Function Chartilla (SFC) esitetään graafisesti ohjelma, joka muodostuu peräkkäisistä tapahtumista. SFC:n avulla monimutkainen ohjelma voidaan pilkkoa pienempiin, helpommin hallittaviin osiin, kuitenkin säilyttäen kokonaisuus eheänä. SFC:llä tehty ohjelma koostuu askeleista (Step), askeleet toisiinsa sitovista siirtoehdoista (Transition) sekä toiminnoista (Action).

Toiminnoilla luodaan SFC:n toiminnallisuus. Askeleessa määritelty toiminto suoritetaan, ja siirtoehdon täytyessä siirtoehto edeltävä askel deaktivoidaan ja seuraava askel aktivoidaan. Kukin SFC:n osa voidaan ohjelmoida millä tahansa standardin määrittelemällä ohjelmointikielellä, mukaan lukien SFC. SFC:tä voi olla ohjelmassa useita ja yhdessä SFC:ssä voi olla useampi rinnakkainen haara riippuen ohjattavan järjestelmän vaatimuksista. (IEC 61131-3: a standard programming resource n.d.). CoDeSys tarjoaa ohjelmoinnin ja debuggauksen helpottamiseksi kevyempää versiota SFC:stä. Tässä versiossa toiminnot ovat askeleen kanssa yhtä kauan aktiivisia. (Feston CoDeSys - koulutusmateriaali 2011). Kuviossa 24 on esitetty esimerkki SFC:n käytöstä.



Kuvio 24. Esimerkki SFC-ohjelmointikielestä

CFC (Continuous Function Chart), vapaa blokkikaavio, on muunnelma FBD-editorista. Toisin kuin FBD, CFC mahdollistaa suorat yhteydet lohkojen välillä, esimerkiksi takaisinkytkennän tai silmukan ohjelmoinnin. Tulojen, lähtöjen ja esimerkiksi operaatioiden väliset yhteydet eivät synny automaattisesti, vaan ohjelmoijan on piirrettävä ne itse. (Mt.) Esimerkki CFC: n käytöstä on esitetty kuviossa 25.



Kuvio 25. Esimerkki CFC-ohjelmointikielestä

### Standardin määrittelemät yleiset elementit

Standardin määrittämässä yleisissä elementeissä määritellään ohjelmointikielessä käytettävät datatyypit ja muuttujat. Esimerkkejä erilaisista datatyypeistä ovat mm. Integer-tyyppiset kokonaisluvut sekä totuusarvot (Boolean, tosi/epätosi). Ohjelmassa käytettävät muuttujat voivat olla joko paikallisia muuttujia (LOCAL VARIABLES) tai globaaleja muuttujia (GLOBAL VARIABLES). Jos muuttuja on määritelty paikalliseksi, voidaan samannimisiä muuttujia käyttää toisessa ohjelmassa (PROGRAM), toimilohkossa (FUNCTION BLOCK) tai funktiossa (FUNCTION), eikä konfliktia synny. Jos muuttujan halutaan olevan käytettävissä esimerkiksi useassa toimilohkossa, voidaan se määritellä globaaliksi muuttujaksi. Muuttujien määrittelyssä tulisi välttää lai-

teosoitteiden käyttöä. Tällä tavoin ohjelmakoodin uudelleenkäytettävyys on parempi, eikä koodi ole laitteistoriippuvainen. (Mt.)

### **Program Organization Unit, POU**

IEC 61131-3 -standardissa ohjelmat ja projektit rakentuvat Program Organization Unitien, POU, ympärille. Standardi määrittelee kolme erilaista POU:ta: Program, Function block ja Function. POU:t eroavat toisistaan tietyiltä ominaisuuksiltaan:

- Functionille (FUN) voidaan asettaa parametreja. FUN:lla ei ole muistia, joten jos sisääntuloparametrit (Input parameters) eivät muutu, on tulos aina sama. FUN:lla ei ole ulostuloja, vaan se palauttaa aina oman arvonsa. Esimerkiksi FUN, joka on määritelty Booleaniksi, palauttaa se aina joko totuusarvon TRUE tai FALSE.
- Function blockiin (FB) voidaan asettaa parametreja, sekä oleellisena erona FUN:in, sillä on muisti. Tästä on hyötyä käytettäessä esimerkiksi laskuria (Counter). FB muistaa tilansa, eikä laskuri aloita alusta kutsuttaessa Function Blockia uudestaan.
- Programia (Prog) voidaan kutsua "pääohjelmaksi". Prog on POU, joka käsittelee IO:ta, toiminnaltaan se on samankaltainen, kuin FB. (IEC 61131-3: Programming Industrial Automation Systems.)

### **CoDeSys**

CoDeSys (Controller Development System) on saksalaisen 3S-Smart Software Solutionsin kehittämä, kansainvälisen IEC 61131-3 -standardin mukainen ohjauslaitteiden, kuten ohjelmoitavien logiikoiden, ohjelmointiympäristö. Sadoilla laitevalmistajilla on valikoimissaan älykkäitä laitteita varustettuna CoDeSys-ohjelmointirajapinnalla. Tästä johtuen CoDeSys on käytössä maailmanlaajuisesti kaikenlaisissa automaatiota vaativissa sovelluksissa. (CODESYS 2013.)

Opinnäytetyönä tehtävän tarkastuslaitteiston logiikkaohjelma tehtiin Feston julkaisemalla CoDeSys provided by Festo -versiolla. Se perustuu 3S:n julkaisemaan CoDeSys 2.3:een. Eroja 3S:n julkaisemaan versioon on lähinnä visuaalisessa ilmeessä, Feston omissa help-tiedostoissa sekä kirjastoissa. Kirjastoissa on tarvittavat toimilohkot esimerkiksi servomoottorien ohjaamiseen.

### **Tärkeimpiä ominaisuuksia: automaattinen sekä manuaalinen symbolilistan täydennys**

IEC 61131-3:n mukaan jokainen ohjelmassa käytetty muuttuja on määriteltävä. Lisättäessä uusi muuttuja ohjelmaan kysyy automaattinen avustaja muuttujan tyyppin ja osoitteen sekä muita asetuksia, kuten alkuarvon. Annettujen tietojen perusteella ohjelmointiympäristö täydentää symbolilistoja sitä mukaa, kun ohjelmaa kirjoitetaan. Symbolilistat voidaan tehdä valmiiksi myös käsin, esimerkiksi tekstieditorilla. Valmis lista voidaan kopioida Copy/Paste -menetelmällä ohjelmointiympäristön määrittelyalueeseen, jolloin muuttujat ovat välittömästi käytettävissä. (Feston CoDeSys -koulutusmateriaali 2011.)

Muuttujan määrittely on esitetty kuviossa 26 ja muuttujaluettelo kuviossa 27.

**Kuvio 26. Muuttujan määrittely ja symbolilistan ylläpito**

```

CanOpen implicit Variables
0001 VAR_GLOBAL CONSTANT
0002     MAX_CTRLINDEX : INT := 0;
0003 END_VAR
0004 VAR_GLOBAL CONSTANT
0005     USE_CANOPEN_NODES : BOOL := TRUE;
0006     MAX_MASTERINDEX : INT := 0;
0007     MAX_NODEINDEX : INT := 0;
0008     MAX_SDOINDEX : INT := 15;
0009     MAX_PDOINDEX_RX : INT := 1;
0010     MAX_PDOINDEX_TX : INT := 1;
0011     MAX_MASTER_ODENTRY_IDX : INT := 0;
0012 END_VAR
0013 VAR_GLOBAL
0014     pCANopenMaster : ARRAY[0..MAX_MASTERINDEX] OF CanOpenMaster;
0015     pCanOpenNode : ARRAY[0..MAX_NODEINDEX] OF CanOpenNode;
0016     pCanOpenSDO : ARRAY[0..MAX_SDOINDEX] OF CanOpenSDO;
0017     pCanOpenPDO_Rx : ARRAY[0..MAX_PDOINDEX_RX] OF CanOpenPDO_Rx;
0018     pCanOpenPDO_Tx : ARRAY[0..MAX_PDOINDEX_TX] OF CanOpenPDO_Tx;
0019     ODMEntries: ARRAY[0..MAX_MASTER_ODENTRY_IDX] OF CanOpenODEntry;
0020 END_VAR
0021

```

Kuvio 27. Muuttujaluettelo

### Tärkeimpiä ominaisuuksia: syntaksin väritys ja syöttöavustaja

CoDeSysin ominaisuuksiin kuuluu automaattinen syntaksin väritys. Värilliset koodin osat tekevät koodista helppolukuisen. IEC 61131-3:n avainsanat, kuten “NOT” tai “VAR\_OUTPUT” esitetään sinisinä. Väärin kirjoitetut avainsanat esitetään punaisina, kommentit vihreinä ja aikamuuttujat sekä kiinteät totuusarvot magentana. Ohjelmointiympäristön yksi hyödyllisimpiä ominaisuuksia on syöttöavustin. Ohjelmakoodia kirjoitettaessa voidaan symbolilistoja tai järjestelmämuuttujia selata pikanäppäimen avulla. Kirjoittamalla muuttujan nimen alun ja tämän jälkeen näppäinoikotiellä CTRL + SPACE, listaa ohjelma saman alkuiset muuttujat, jolloin oikean muuttujan voi valita suoraan listalta. (Mt.)



Syntaksin väritys on esitetty kuviossa 28 ja syöttöavustaja kuviossa 29.

```

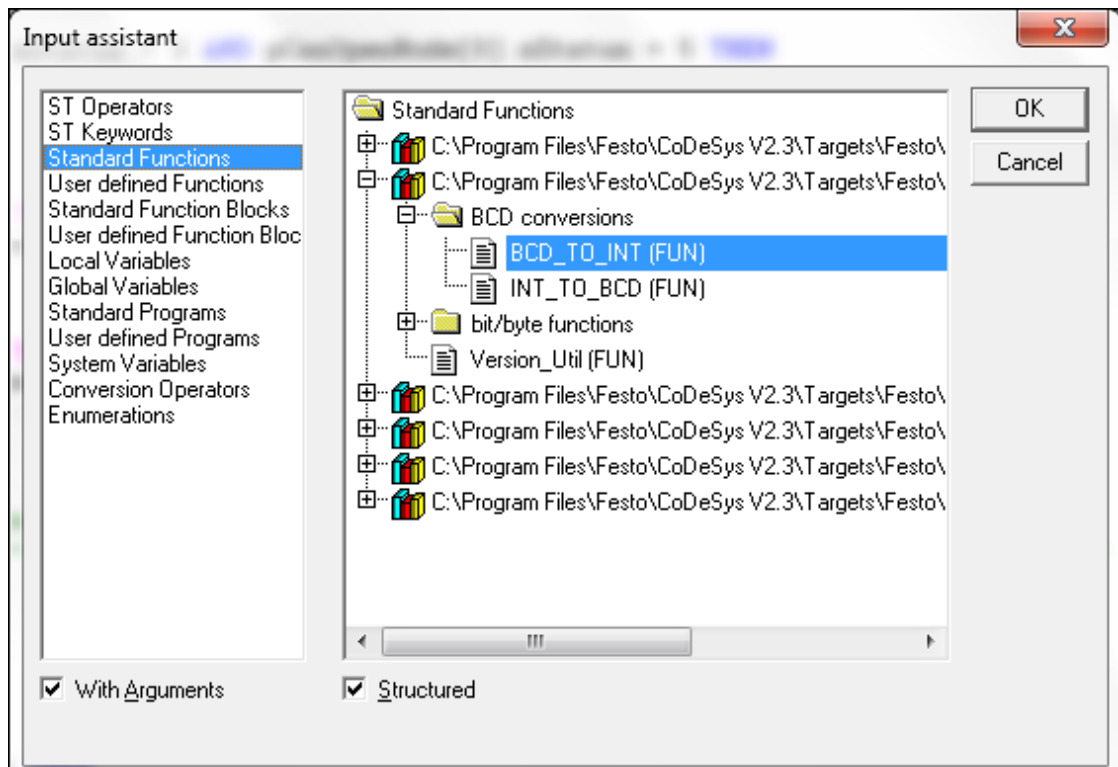
0: Drive.Halt := TRUE;
   Drive.Stop:= TRUE;
   Drive.EnableDrive := TRUE;
   IF Drive.Ready AND Drive.DriveEnabled THEN
     iStep := 10;
   END_IF

10: Drive.StartHoming := TRUE;
    IF Drive.DriveIsReferenced THEN
      Drive.StartHoming := FALSE;
      iStep := 20;
    END_IF

(* record mode - position control *)

```

Kuvio 28. Syntaksin väritys

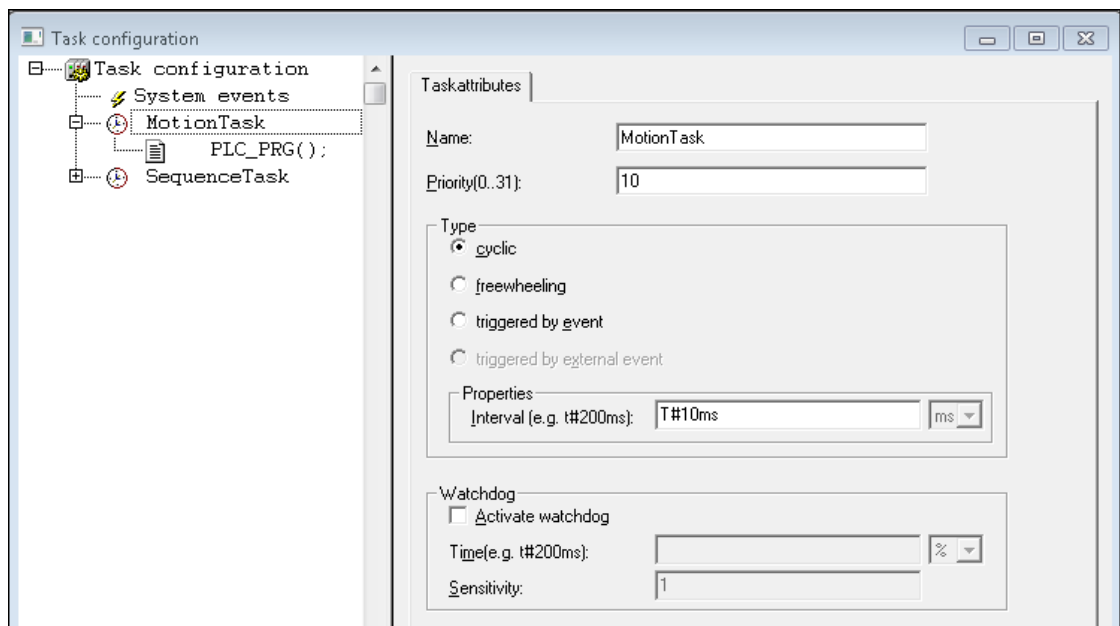


Kuvio 29. Syöttöavustaja

### Tärkeimpiä ominaisuuksia: Task ja PLC Hardware määrittely

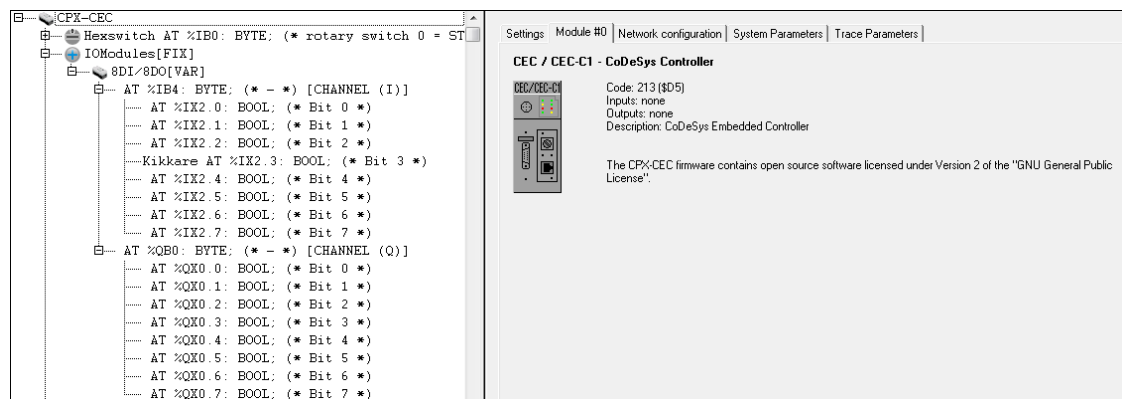
Taskit ohjaavat ohjelmien ja toimilohkojen suoritusta. CoDeSys tukee useiden IEC Taskien luontia. Käyttöjärjestelmästä riippuen Taskeja suoritetaan joko peräkkäisinä tai moniajona. Taskit määritellään ohjelman sisällä. Jokaisella Taskilla on prioriteettinumero (0...31), mitä suurempi numero, sen alhaisempi prioriteetti. Taskin vaihto kuluttaa prosessori-aikaa, joten niitä ei ole järkevää luoda liian montaa. Taskin käynnistys voi olla syklinen, jolloin se suoritetaan määrätyn aikavälein, tai jatkuvasti suoritettava (Freewheeling), jolloin sykli-aikaa ei ole määritelty, tai jonkun tapahtuman laukaisema (Triggered by event). Viimeinen vaihtoehto tarkoittaa sitä, että Taskin suoritus käynnistyy, kun ohjelmoijan määrittämä ehto, esimerkiksi muuttuja saa arvon TRUE, täyttyy. (Feston CoDeSys -koulutusmateriaali 2011.)

Kuviossa 30 on esitetty Taskin määrittely.



Kuvio 30. Taskin määrittely

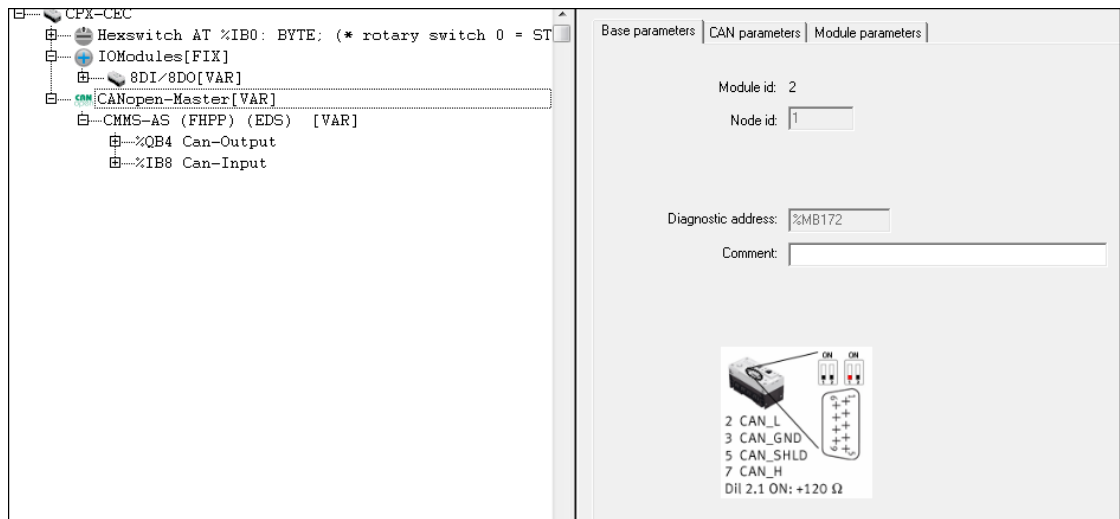
Järjestelmän fyysiset tulot ja lähdöt määritellään CoDeSysin sisäisellä työkalulla, PLC Configurationilla, joka on esitetty kuviossa 31. IO-yksiköt voidaan lisätä joko käsin laitekonfiguraatioon, tai laitteiston kokoonpano voidaan lukea suoraan logiikasta, jos logiikka on yhteydessä tietokoneeseen. Määrittelyn lopputuloksena on kuvion 31 vasemmassa laidassa esitetty IO-kuvaus, jossa IO-osoite on nähtävissä joko numeerisena osoitteena tai symbolisena nimenä, systeemimuuttujana (System variable). (Feston CoDeSys - koulutusmateriaali 2011.)



**Kuvio 31. PLC Configuration**

Jos käytettävä logiikka on CANopen tai Profibus DP master, voidaan kenttäväylä konfiguroida PLC Configurationilla käyttäen EDS- (CANopen) tai GSD- (Profibus DP) tiedostoja. (Mt.)

Väylän konfigurointi on esitetty kuviossa 32.



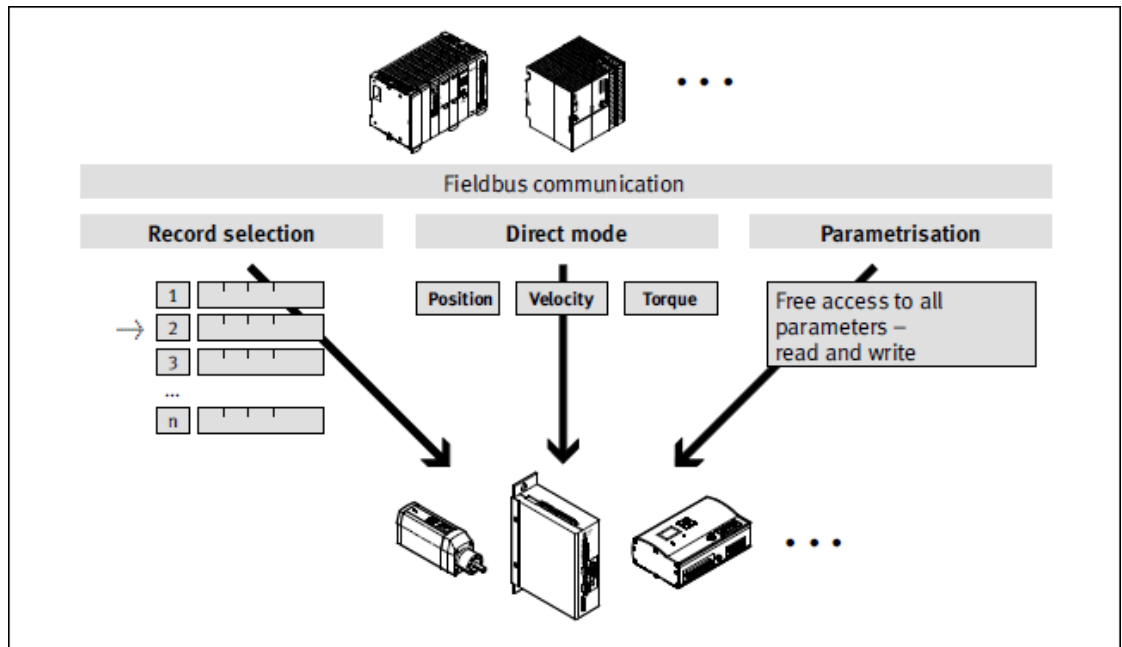
Kuvio 32. Väylän konfigurointi

### Tärkeimpiä ominaisuuksia: integroitu simulointi

CoDeSys mahdollistaa ohjelman debuggauksen ilman PLC hardwarea. Simulointitoiminnolla voidaan testata ohjelmistoa ennen sen lataamista logiikkaan. Fyysisten tulojen ja lähtöjen tilat voidaan simuloida PLC Configuration -työkalulla. (Feston CoDeSys -koulutusmateriaali 2011.)

### 7.3 FHPP – Festo Handling and Positioning Profile - tiedonsiirtoprofiili

FHPP (Festo Handling and Positioning Profile) on Feston kehittämä ja optimoitu tiedonsiirtoprofiili, joka mahdollistaa yhtenäisen, Feston valmistamien, erilaisten moottorinohjaimien ohjauksen ja ohjelmoinnin erilaisissa kenttäväyläympäristöissä. FHPP määrittelee ohjaimien IO-datarakenteen, parametroiden, toimintatilan sekä sekvenssiohjauksen. (Festo Handling and Positioning Profile 2010, 21.) Kuviossa 33 on esitetty FHPP:n periaate.



Kuvio 33. FHPP:n periaate (Festo Handling and Positioning Profile 2010, 21)

Kommunikaatio logiikan ja moottorinohjaimen välillä kenttäväylää pitkin perustuu kahdeksan tavun suuruisiin ohjaus- ja tilatietoihin, joita voidaan lukea ja kirjoittaa reaaliajassa. Ohjaimen tarvitsemat parametrit lähetetään erillistä parametrikanaavaa pitkin, joka on kahdeksan tavun suuruinen. (Festo Handling and Positioning Profile 2010, 22.)

Kuviossa 34 on esitetty ohjaus- ja tilatietoja varten määritellyt muistialueet, sekä parametrintikanavalle varattu muistialue.

```

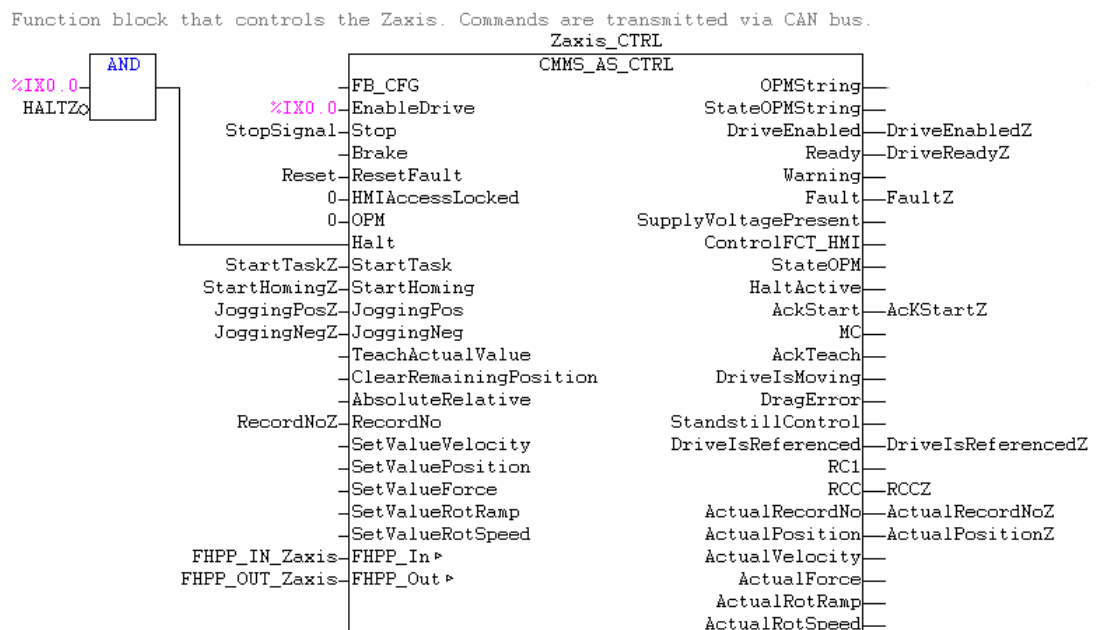
Yaxis_CTRL: CMMS_AS_CTRL;
Yaxis_INIT: CMMS_AS_PRM_INIT;
Yaxis_PRM: CMMS_AS_PRM_SINGLE;
FHPP_IN_Yaxis AT %IB40 : BYTE;
FHPP_OUT_Yaxis AT %QB36 : BYTE;
DATA_REF_Yaxis: FHPP_PRM_REF;
FPC_IN_Yaxis AT %IB48 : BYTE;
FPC_OUT_Yaxis AT %QB44 : BYTE;

Z2axis_CTRL: CMMS_AS_CTRL;
Z2axis_INIT: CMMS_AS_PRM_INIT;
Z2axis_PRM: CMMS_AS_PRM_SINGLE;
FHPP_IN_Z2axis AT %IB56 : BYTE;
FHPP_OUT_Z2axis AT %QB52 : BYTE;
DATA_REF_Z2axis: FHPP_PRM_REF;
FPC_IN_Z2axis AT %IB64 : BYTE;
FPC_OUT_Z2axis AT %QB60 : BYTE;

```

Kuvio 34. Muistialueiden varaus

FHPP:n ansiosta kommunikaatio moottorinohjaimien ja logiikan välillä on äärimmäisen yksinkertaista. Asentamalla sopiva lisäosapaketti ohjelmointiympäristöön, saadaan käyttöön valmiit toimilohkot, joilla voidaan lähettää tarvittavat komennot ja parametrit ohjaimille, jotka vuorostaan lähettävät tietoa omasta tilastaan takaisin logiikkaan. Käytännössä ohjelmoijan ei tarvitse tietää, minkä tavun ja mihin bittiin hänen on kirjoitettava, jotta laite käyttäytyy hänen toivomallaan tavalla. Kuviossa 35 on esitetty FB, jolla ohjataan moottorinohjainta.



Kuvio 35. Moottorinohjainta ohjaava toimilohko

FHPP määrittelee kaksi erilaista moottorinohjaimen toimintatilaa, joita ovat

- Record selection mode
- Direct mode.

Record selection modea käytettäessä moottorinohjaimen voidaan tallentaa 63 erilaista positiota taulukkomaisessa muodossa. Jokainen taulukon rivi pitää

sisällään kaikki paikoituksessa tarvittavat parametrit, kuten nopeuden ja kiihdytysrampit. Paikoitus record modessa tapahtuu siten, että ohjaimelle lähetetään väylää pitkin tieto, minkä taulukon rivin mukaan ohjain ajaa seuraavaan pisteeseen. (Festo Handling and Positioning Profile 2010, 23.)

Direct mode eroaa record selection modesta siinä, että paikoitus tapahtuu täysin logiikan käskemänä. Se on record selection modea joustavampi tapa paikoitukseen, sillä sitä on helpompi säätää, esimerkiksi työstettäessä erikokoisia kappaleita. Record selection modea käytettäessä koko paikoitustaulukko on parametroitava uudelleen, jos työstettävä kappale muuttuu. Direct modessa uudet arvot on helppoa syöttää uusina parametreina logiikkaohjelmaan. (Mts. 25.)

Opinnäytetyössä lineaariliikkeiden ohjaukseen käytettiin record selection modea, sillä laitteistolla mitataan aina samanlaisia kappaleita. Logiikkaohjelma yksinkertaistuu, kun kolmelle moottorinohjaimelle ei tarvitse lähettää useita parametreja, vaan ainoastaan taulukon rivinumero. Pyöritystä ohjattiin direct modella, sillä pyöritykselle ei tarvittu tarkkaa paikoitusta. Lisäksi direct modella ajettaessa vältytään edestakaiselta liikkeeltä. Record selection modella pyöritys olisi pitänyt käskeä aina sekvenssin päätteeksi takaisin ensimmäiseen taulukon arvoon. Tästä olisi seurannut turhaa moottorin ajamista tilanteessa, jossa mittauksen aikana moottori on pyörinyt 100 kierrosta ajettaessa taulukon arvoon 2.

Suurin puute FHPP:ssä on se, ettei sen avulla pystytä toteuttamaan synkronoituja liikkeitä eri ohjaimien välillä, vaan tähän tarkoitukseen on tukeuduttava CoDeSysin SoftMotion -tekniikkaan, joka mahdollistaa servojen synkronoidut liikkeet. Huono puoli SoftMotionin käytössä on se, että FHPP:ssä käytetyt toimilohkot eivät ole tällöin käytettävissä. Tästä syystä tarkastuslaitteiston liikkeet päädyttiin toteuttamaan FHPP:tä käyttäen.

## 7.4 FCT – Festo Configuration Tool -ohjelmisto

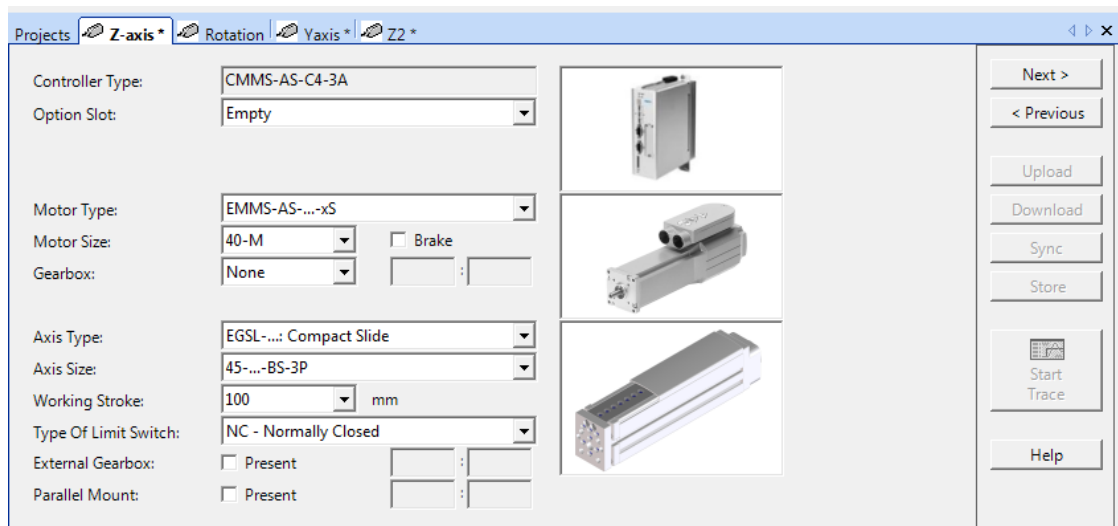
Festo Configuration Tool on ohjelmisto, joka on tarkoitettu Feston valmistamien komponenttien asetusten määrittämiseen sekä käyttöönottoon. FCT:n avulla luodaan yhteys tietokoneen ja säädettävän komponentin välille. (Festo Configuration Tool n.d., 4.)

FCT:n käyttö voidaan jakaa kolmeen eri osa-alueeseen:

1. asetusten määrittäminen
2. viritys
3. positioiden tallennus.

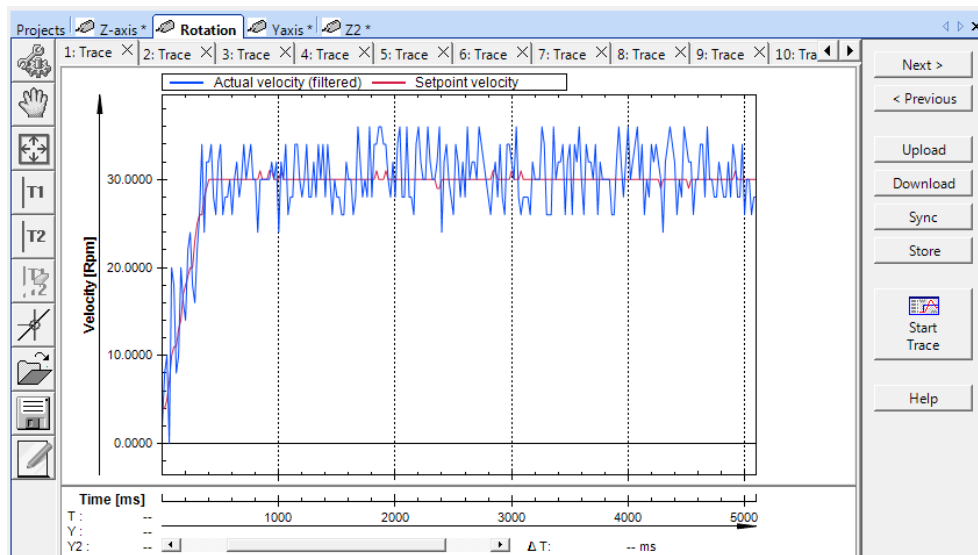
Asetusten määrittäminen vaikuttaa, koostuuko käytettävä kokoonpano puhtaasti Feston tuotteista, vai onko Feston moottoriin yhdistetty jonkun muun valmistajan valmistama komponentti. Jos käytettävät komponentit ovat Feston valikoimista, osaa ohjelma kertoa mm. liikkeiden maksiminopeudet sekä sopivat hidastus- ja kiihdytysrampit. Asetusten määrittämisessä valitaan myös käytettävä ohjaustapa, eli käytetäänkö moottorinohjaimen käskytykseen väylää tai IO-rajapintaa. Lisäksi moottorinohjaimen IO-rajapintaan lähetettävät signaalit voidaan määrittää asetuksista. Kuviossa 36 on esitetty tarkastuslaitteiston pienemmän pystyliikkeen laitteistomäärittäminen FCT:ssä.





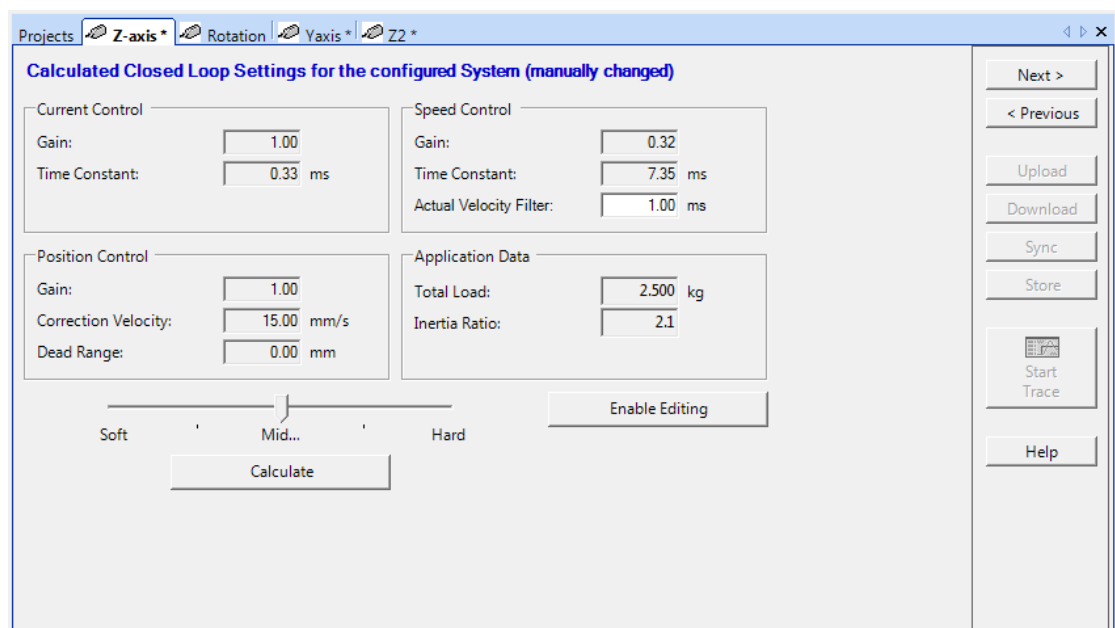
Kuvio 36. Komponenttien määrittäminen FCT:ssä

Käytettäessä Feston komponentteja ohjelmisto osaa virittää moottorin automaattisesti siihen kytketyn oheislaitteiston perusteella. Käyttäjän on vain valittava, onko liike pysty- vai vaakasuuntaan sekä liikuteltava massa. Yhdistelemällä eri valmistajien komponentteja ohjelma ei pysty virittämään moottoria, vaan käyttäjän on tehtävä se käsin. Tällöin käyttäjän apuna on, kuviossa 37, kaksikanavainen oskilloskoopi, jonka avulla moottori voidaan virittää käsin.



Kuvio 37. FCT:n oskilloskoopi

Käyttäjän säädettävänä on kolme eri säätöpiiriä. Kaksi säätöpiireistä on PI-säätimiä ja viimeinen on pelkkä P-säädin. Ensimmäinen säätöpiiri on nimeltään Current control, toinen on nimeltään Speed Control ja kolmas Position Control. Current control vastaa aseteltavaa virtarajaa, Speed control hoitaa nopeussäädön ja Position control vastaa paikoituksesta. Kuviossa 38 on esitetty FCT:n viritysparametrit.



**Kuvio 38. FCT:n viritysparametrit**

Positioiden tallennus tapahtuu taulukkomuodossa, joka on esitetty kuviossa 39. Servomoottori voidaan ajaa FCT:llä haluttuun positioon ja nappia painamalla tämä tallentuu taulukon halutulle riville. Jokaiselle riville voidaan määrittää esimerkiksi erilaiset kiihdytysrampit sekä loppunopeudet. Yksinkertainen liikesarja voidaan toteuttaa ilman ohjelmoitavaa logiikkaa käyttämällä pelkästään FCT:n taulukkoon tallennettuja pisteitä. Jokaiselle positiolle voidaan antaa ehto, mitä liikkeen päätyttyä tehdään. Laite voi odottaa hetken,

tiettyä tapahtumaa tai jatkaa välittömästi uudella liikkeellä.

FCT	No.	Mode	Position [mm]	Profile	Comman	Dest.	Input	Comment
	1	A	0.00	0	END			Home position.
	2	A	71.05	1	END			Starting point of measuring the flat side of the washe
	3	A	48.67	2	END			End point of measuring the flat side of the washer rin
	4	A	48.67	1	END			Moving closer to the washer ring. Same position as t
	5	A	72.09	2	END			Starting point of measurig the curved side of the ring
	6	A	48.67	3	END			End point of measuring the curved side of the washe
	7	A						
	8	A						
	9	A						
	10	A						
	11	A						
	12	A						
	13	A						
	14	A						
	15	A						
	16	A						
	17	A						
	18	A						
	19	A						

Kuvio 39. Positiotaulukko

## 7.5 Logiikkaohjelma

### 7.5.1 PLC\_PRG: logiikkaohjelman pääohjelma

Logiikkaohjelman runko muodostuu pakollisesta "PLC\_PRG"-nimisestä

POUsta, joka generoidaan automaattisesti uutta projektia luotaessa.

PLC\_PRG:n kautta muut ohjelmakoodissa olevat funktiot ja toimilohkot pysyvät vaikuttamaan logiikan tuloihin ja lähtöihin. Kuviossa 40 on esitetty toimilohkojen kutsu PLC\_PRG:ssä.

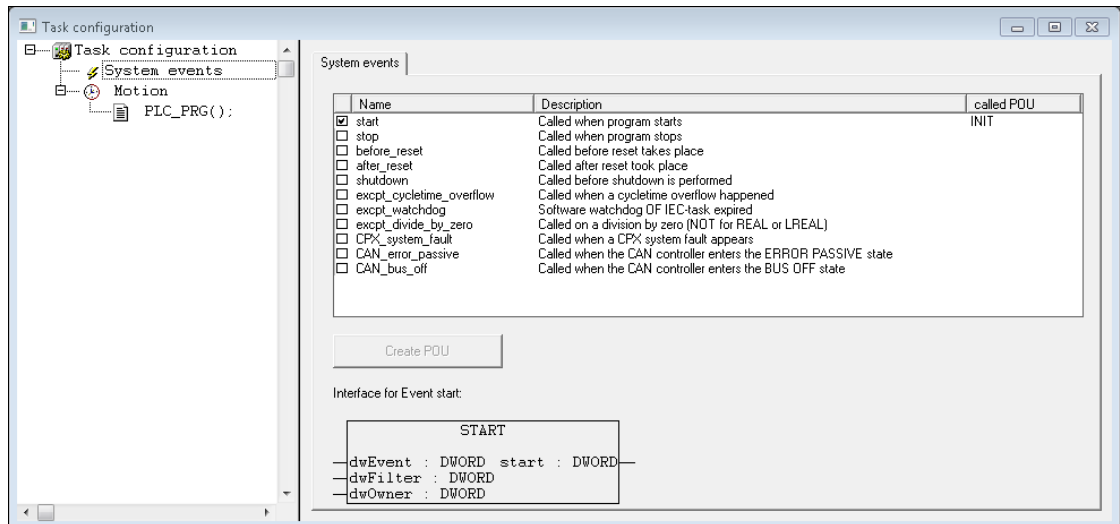
0019	Function block for homing the drives Homing Homing
0020	Function block controlling the signal lights Lights Lights
0021	Function block controlling the motions. Flat side of the washer ring Sequence_flat Sequence_flat
0022	Function block controlling the motions. Curved side of the washer ring Sequence_curved Sequence_curved

**Kuvio 40. Toimilohkojen kutsu**

Tarkastuslaitteiston toiminnallisuus on hajautettu eri toimilohkoihin, joita kutsutaan PLC\_PRG:ssä. Tällä menettelyllä ohjelma pysyy ymmärrettävänä ja siistinä. Moottorinohjaimien ohjaamiseen ja parametrintiin käytettävät toimilohkot sijoitettiin suoraan PLC\_PRG:hen. Nämä toimilohkot muodostavat ohjelmakoodin ytimen, joten niiden sijoittaminen pääohjelmaan koettiin mielekkääksi. PLC\_PRG on kokonaisuudessaan esitettyä liitteessä 5.

### 7.5.2 Järjestelmäfunktiot

Ohjelmassa on käytössä neljä erilaista järjestelmäfunktiota (System events), joita kutsutaan esimerkiksi logiikan käynnistyessä tai väylävirheen tapahtuessa. Funktion luodaan Task Configuration työkalulla, samalla millä määritellään Taskit. Työkaluun on listattu, kuvion 41 mukaan, tapahtumat jotka voivat käynnistää järjestelmäfunktion.



**Kuvio 41. Järjestelmäfunktion luominen**

Tarkastuslaitteistossa käytettävät järjestelmäfunctiot ovat:

- INIT
- CPX\_system\_fault
- CAN\_error\_passive
- CAN\_bus\_off.

## INIT

Ensimmäistä järjestelmäfunctiota, INIT, kutsutaan logiikan käynnistyessä. Funktion tarkoitus on resetoida ja käynnistää CAN-väylään kytketyt laitteet logiikan käynnistyessä. Tällä varmistetaan siitä, että tarvittavat laitteet ovat hereillä ja kunnossa, kun laitteistoa aiotaan käynnistyksen jälkeen käyttää. Väylän laitteiden herättelyn lisäksi funktiolla suoritetaan merkkivalojen testaus sytyttämällä tarkastuslaitteiston kaikki merkkivalot päälle logiikan käynnistyessä. Funktion toiminnallisuus on esitetty kuviossa 42.

```

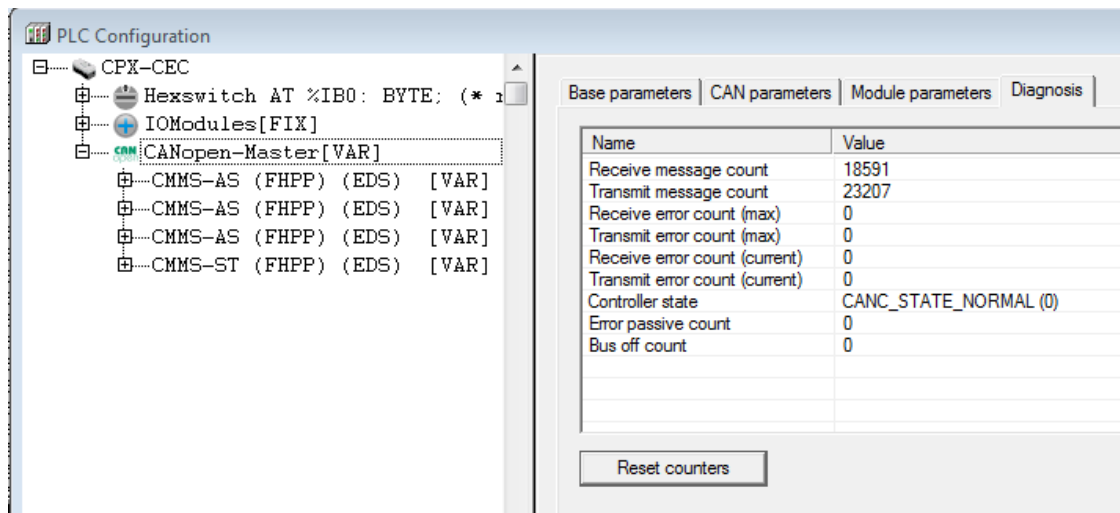
INIT (FUN-ST)
0001 FUNCTION INIT: DWORD
0002 VAR_INPUT
0003     dwEvent: DWORD;
0004     dwFilter: DWORD;
0005     dwOwner: DWORD;
0006 END_VAR
0007 (*This function resets and starts all the bus no
0008 variable TestSignalLights to true. TestSignalLi
0009
0010 VAR
0011 END_VAR
0012
0001 pCanOpenMaster[0].bUseResetAllNodes := TRUE;
0002 pCanOpenMaster[0].bUseStartAllNodes := TRUE;
0003 TestSignalLights:=TRUE;
0004

```

Kuvio 42. Järjestelmäfunktio INITin toiminnallisuus

### CANBUS\_passive ja CANBUS\_error

Järjestelmäfunktiot CANBUS\_passive ja CANBUS\_error liittyvät CAN-väylän häiriöihin. CANBUS\_passive -funktiota kutsutaan, kun väylässä on tapahtunut tiedonsiirtovirhe. Väylä ohjataan passiiviseen tilaan, kun virhemäärä ylittää arvon 128. Jos virhemäärä ylittää arvon 96, puhutaan tällöin voimakkaasti häiriintyneestä väylästä. CANBUS\_error -funktiota kutsutaan, kun virhetilanteiden määrä ylittää arvon 256. Tällöin väylä sammutetaan ja näin tiedonsiirto estyy. Datalaskurit, jotka on esitetty kuviossa 43, ovat nähtävillä ohjelmointiympäristössä, kun logiikka on päällä ja yhteys on muodostettu logiikan ja ohjelmointiympäristön välillä.



Kuvio 43. Datalaskurit

Kutsuttaessa funktiot CANBUS\_passive ja CANBUS\_error asettavat muuttujan BUSError tilaan TRUE. Tätä muuttujaa käytetään liikkeiden mahdollistaman STOP-signaalin muodostamisessa. Väylävirheen tapahtuessa liikkeiden toteuttaminen tulee mahdottomaksi. Tilasta ilmoitetaan operaattorille servovirheestä ilmoittavalla merkkivalolla.

### CPX\_system\_fault

CPX\_system\_fault -funktion toiminta on samanlainen, kuin väylävirheistä ilmoittavilla järjestelmäfunktioilla. Logiikka ei mene STOP-tilaan virheen satuessa, vaan ohjelman suoritusta jatketaan, jolloin seurauksena voi olla vaaratilanteita käyttäjälle sekä laitevaurioita. Jos CPX\_system\_fault -funktiota kutsutaan, asetetaan SystemError -muuttuja tilaan TRUE. Tätä muuttujaa käytetään samaan tapaan kuten aiemmin mainittua BUSError -muuttujaa. Kyseisiä muuttujia ei voida kuitata pois, vaan diagnostiikkatiedoista on tutkittava, mikä aiheutti virheen ilmenemisen. Kuittaaminen tapahtuu resetoimalla ja käynnistämällä logiikka uudelleen.

### 7.5.3 Toimilohkot

Logiikkaohjelma rakentuu useista eri toimilohkoista, joita kutsutaan pääohjelmassa. Laitteiston eri toimintoja varten on jokaiselle oma toimilohkonsa, esimerkiksi merkkivalojen ja käsiajon ohjaus. Toimilohkojen ohjelmointikielellä on ST lukuun ottamatta sekvenssejä, jotka on ohjelmoitu SFC:llä.

Toimilohkot voidaan jakaa liikkeitä toteuttaviin toimilohkoihin ja avustaviin toimilohkoihin. Avustavia toimilohkoja käytetään mm. STOP-signaalin muodostamiseen ja valojen merkkivalojen ohjaamiseen. Liikkeitä toteuttavista toimilohkoista voidaan esimerkkinä mainita referenssiajon sekvenssi.

#### Avustavat toimilohkot

Avustavilla toimilohkoilla muodostetaan liikkeiden toteuttamisessa tarvittavia muuttujia, valvotaan laitteiston tilaa sekä ohjataan laitteiston merkkivaloja. Avustavia toimilohkoja ovat seuraavat:

- Home, jolla tarkistetaan ovatko kaikki akselit kotiasemissa. Toimilohkon sisältö on liitteenä 6.
- Lights, jolla ohjataan laitteiston merkkivaloja. Toimilohkon sisältö on liitteenä 7.
- ReadyForOperation, jolla tarkistetaan, onko laitteisto valmiina aloittamaan mittaussekvenssin. Toimilohkon sisältö on liitteenä 8.
- SignalLightTest, jolla toteutetaan merkkivalojen testaus laitteiston käynnistyksen yhteydessä. Toimilohkon sisältö on liitteenä 9.
- StopFunction, jolla muodostetaan tarvittava STOP-signaali. Toimilohkon sisältö on liitteenä 10.
- Trigger, jolla käynnistetään RollScan 250 -mittalaitteen mittaus. Toimilohkon sisältö on liitteenä 11.



Kaikki avustavat toimilohkot ovat yksinkertaisia JA- sekä TAI-piireistä muodostuvia toimintoja. Kuviossa 44 on esitetty Lights-toimilohkossa oleva merkkivalopainikkeen S5 ohjaus.

```

FAULT:=(BUSError
OR SystemFault (*if BUSError or SystemFault is TRUE, S4 is blinking*)
OR SFCError)
OR (ServoFault()      (*If function ServoFault is true then signal light S4 is lit*)
AND Blinker)
OR SignalLights;

```

**Kuvio 44. Merkkivalopainikkeen S5 ohjaus**

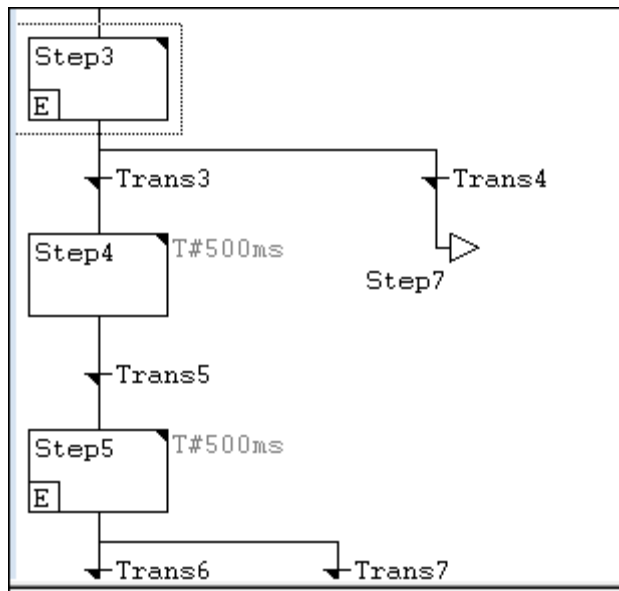
### Liikkeitä toteuttavat toimilohkot

Liikkeitä toteuttavilla toimilohkoilla ohjataan laitteiston liikkeiden toteuttamiseen käytettäviä muuttujia sekä logiikan tuloja ja lähtöjä. Liikkeitä toteuttavia toimilohkoja ovat:

- Jog, jolla ohjataan laitteiston käsiajoa. Toimilohkon sisältö on liitteenä 12.
- Homing, jolla ohjataan akseleiden referenssiajoa. Toimilohkon sisältö on liitteenä 13.
- Sequence\_flat, joka sisältää mittaussekvenssin liukurenkaan tasaisen pinnan mittausta varten. Toimilohkon sisältö on liitteenä 14.
- Sequence\_curved, joka sisältää mittaussekvenssin liukurenkaan kaarevan pinnan mittausta varten. Toimilohkon sisältö on liitteenä 15.
- HomeSequence\_flat, joka sisältää kotiajosekvenssin liukurenkaan tasaiselle pinnalle. Toimilohkon sisältö on liitteenä 16.
- HomeSequence\_curved, joka sisältää kotiajosekvenssin liukurenkaan kaarevalle pinnalle. Toimilohkon sisältö on liitteenä 17.

Toimilohko Jog eroaa muista liikkeitä toteuttavista toimilohkoista siinä, että se on ohjelmoitu käyttäen ohjelmointikielenä ST:tä, muut lohkot on ohjelmoitu SFC:llä. Sisällöltään Jog on avustavien toimilohkojen kaltainen koostuen yksinkertaisista JA- sekä TAI-piireistä.

Kuviossa 45 on esitetty ote HomeSequence\_flat -toimilohkosta.



**Kuvio 45. Ote HomeSequence\_flat -toimilohkosta**

Sekvensseissä käytettävien askeleiden ja siirtoehtojen ohjelmointikielen käyttäjä voi valita vapaasti kaikista IEC 61131-3 -standardin määrittelemistä kielistä, yksittäinen askel voi sisältää jopa pienen sekvenssin. Tässä ohjelmassa sekvenssien askeleet ja siirtoehdot ohjelmoitiin ST-ohjelmointikielellä.

Sekvenssien siirtoehdot ovat yksinkertaisia, pääasiassa JA-piireistä koostuvia ehtoja sekvenssin jatkumiselle. Kuviossa 46 on esitetty HomeSequence\_flat -sekvenssin siirtoehdon Trans3 sisältö.

Transition Trans3 (ST) - HomeSequence_flat (FB-SFC)	
0001	ActualPositionZ = 0
0002	AND Z_MOTIONCOMPL
0003	AND StopSignal
0004	AND DEMAGN
0005	

**Kuvio 46. Siirtoehto Trans3**

Kuviosta 46 nähdään, että kotiajosekvenssi HomeSequence\_flat etenee seuraavaan askeleeseen, jos Z-akselin positio on 0, akselin liike on pysähtynyt, STOP-signaali on päällä ja liukurengas on magnetoitu.

## 7.6 Laitteiston testaus

### 7.6.1 Sähkökytkentöjen testaus

Tarkastuslaitteiston testaamisen aikana laitteistossa ilmeni sekä sähköisiä että mekaanisia suunnitteluvirheitä ja puutteita. Sähköistyksen osalta suurin vastoinkäyminen koettiin käynnistettäessä servo-ohjaimia ensimmäistä kertaa. Kaksi ensimmäistä ohjainta käynnistyivät ja toimivat normaalisti. Käynnistettäessä viimeinen ohjain laukesivat laitteiston kaikki johdonsuojakatkaisijat. Kytkennät tarkastettiin, mutta esimerkiksi oikosulkua kytkennöistä ei löydetty. Ohjaimia yritettiin käynnistää uudestaan, jolloin kaksi ohjainta kolmesta todettiin rikkoontuneiksi. Sähkökuvien tarkistuksen ja laitevalmistajan konsultaation jälkeen ohjaimet rikkoneeksi viaksi paljastui suunnitteluvirhe laitteiston sähkökuvissa. Epähuomiossa moottorinohjaimen jarruvastusliitântään oli piirroksissa kytketty 24 V ja 0 V. Laiterikko oli väistämätön, sillä näissä liittimissä tasajännite nousee useisiin satoihin voltteihin. Virheellisen suunnitelma päättyi valmiiksi kytkennäksi siitä huolimatta, että toinen suunnittelija tarkisti kuvat, eikä kokenut sähköasentajakaan huomannut virheellistä kytkentää. Testauksen viivästymisen lisäksi moottorinohjaimien rikkoutuminen

aiheutti taloudellisia kuluja, sillä rikkoutuneiden ohjaimien tilalle oli ostettava uudet vastaavat.

Virheellisen jarruvastuskytkennän lisäksi moottoreiden jarrukytkentään käytettävät johtimet olivat ristissä. Käytännössä tällä virheellä oli merkitystä ainostaan suurimman akselin toimintaan, sillä moottorin jarru ei vapautunut, kun akselia koetettiin viritysvaiheessa ajaa.

Asennusvaiheessa sähköasentaja oli kytkenyt pystyliikkeistä vastaavien servomoottoreiden ja näiden moottorinohjaimien väliset kaapelit ristiin. Tämä aiheutti erikoista käytöstä viritysvaiheessa, kun suurempaa pystyliikettä ei saatu liikkumaan kovista yrityksistä huolimatta. Liikkeet eivät onnistuneet, sillä moottorinohjaimessa oli väärät parametrit väärälle akselille. Kun moottorikaapelit kytkettiin oikeisiin ohjaimiin ja moottoreihin, toimivat akselit moitteettomasti.

### **7.6.2 Askelmoottori**

Askelmoottorista muodostui varsinainen riesa laitteiston testauksen edetessä. Moottorin viritys osoittautui mahdottomaksi ilman alennusvaihteistoa. Pyörítettävä massa, magnetointirullat, olivat liian raskaat ja pyörimisnopeus liian alhainen askelmoottorin ominaisuuksiin nähden. Alhaisella kierrosnopeudella moottori pyöri nykimällä, kuten kuvion 38 kuvaaja (s. 62) kertoo. Lisättäessä moottorin kierrosnopeutta käytös hieman asettui, mutta nostettaessa kiihdytysramppia suuremmaksi, ei moottori pysynyt säädön mukana, jolloin moottorinohjain meni vikatilaan ilmoittaen liian suuresta erosta asetusarvon ja oloarvon välillä. Moottorista loppui yksinkertaisesti voima. Moottoriin tilattiin alennusvaihteisto, jolloin moottori pystyttiin virittämään ja liikkeet toteuttamaan.

Moottorin virittämiseen käytettävä FCT osoittautui todella huonoksi ohjelmaksi suunniteltuun käyttötarkoitukseen. Ohjelman help-tiedostot olivat ylimalkaisia ja puutteellisia, eikä niistä ollut juurikaan apua. Ohjelmasta myös puuttuu toiminto, joka suorittaisi askelvastekokeen ja virittäisi ohjaimen automaattisesti. Lisäksi käsin tehtävän virityksen apuna käytettävään oskiloskooppiin ei saa näkyviin riittävästi muuttujia, jotta viritys saataisiin tehtyä järkevästi. Käsin virittäminen etenikin puhtaasti yrityksen ja erehdyksen kautta.

### 7.6.3 Logiikkaohjelma

Logiikkaohjelmaan tehtiin muutoksia ja korjauksia testauksen edetessä, jotta laitteisto saatiin toimimaan halutulla tavalla. Suurilta muutoksilta vältyttiin perusteellisen toimintakuvauksen ansiosta, lähinnä ensimmäiseen ohjelmaversioon jouduttiin lisäämään sekvensseihin lisäaskeleita, joilla käsitellään sekvenssin keskeyttämistä.

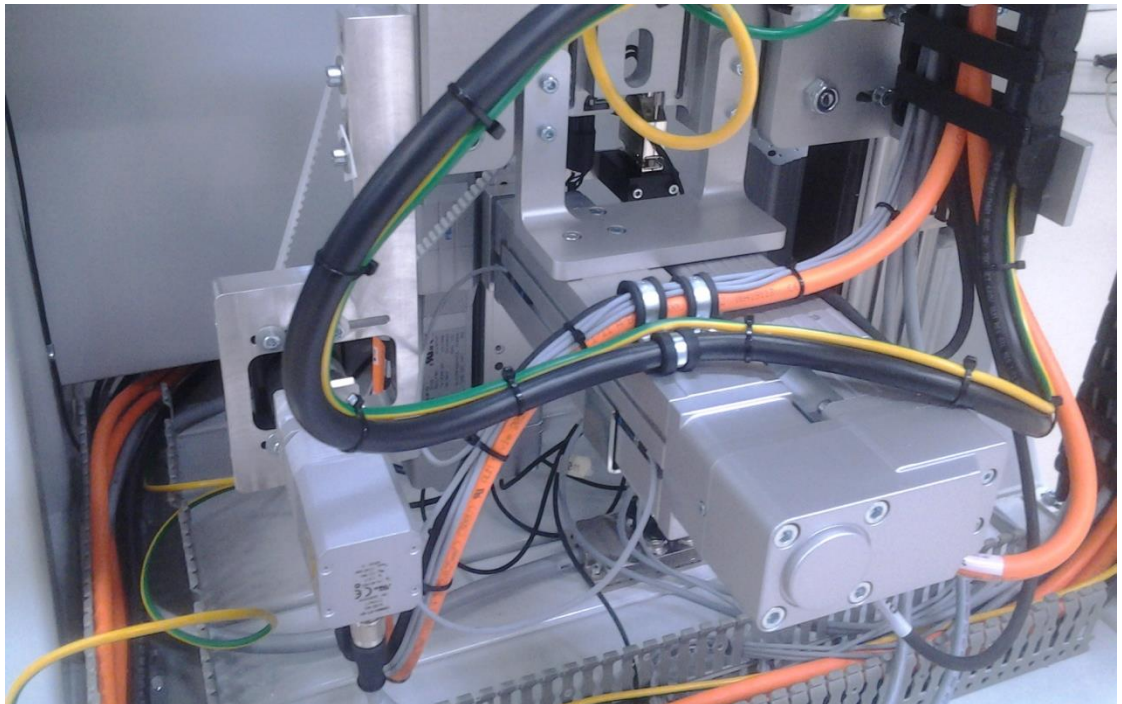
Anturin kääntöön käytettävän DC-moottorin ohjaus tehtiin uudelleen testausvaiheessa, sillä alkuperäinen moottorin ohjaus ei toiminut lainkaan. Esimerkiksi moottorin ohjauksessa käytettävä päästöhidastus lisättiin, koska anturin kääntymisen tunnistava rakenne pysäytti anturin ennen kuin se oli kääntynyt oikeaan asentoon. Tätä seikkaa ei pystytty ennakoimaan ohjelman ensimmäistä versiota ohjelmoitaessa.

Merkkivalojen ohjausta muutettiin kokeneemman suunnittelijan kanssa käytöjen keskusteluiden perusteella. Lopputuloksena suurimmasta osasta vilkkuvista valoista luovuttiin sekä hätäpysäytys-painikkeen ja Start-painikkeen valot vaihdettiin valkoisista valoista punaiseen ja vihreään. Myös häiriöstä ilmoittavien merkkivalojen toimintaa yksinkertaistettiin ja yhdistelmävaloista luovuttiin.

#### 7.6.4 Häiriöt

Liukurenkaan mittaukseen käytettävä anturi on herkkä sähköisille häiriöille, joten ennen varsinaisia mittauksia laitteistoa testattiin anturin kanssa mahdollisten häiriöiden selvittämiseksi. Ensimmäisillä testikerroilla häiriöitä esiintyi runsaasti.

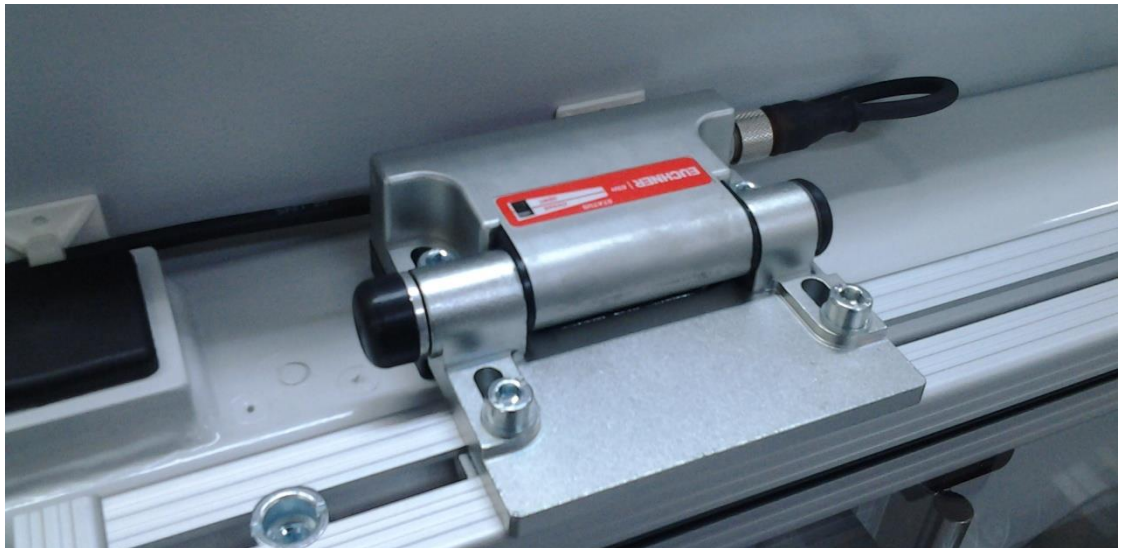
Suurimmaksi häiriön lähteeksi osoittautui askelmoottori. Tilannetta koetettiin korjata eristämällä anturi muusta laitteistosta, mutta tuloksetta. Myös lisämaadoituksia kokeiltiin, mutta turhaan. Häiriöt saatiin putoamaan kohtuulliselle tasolle ainoastaan siirtämällä askelmoottori mahdollisimman kauas anturista sekä koteloimalla ja eristämällä askelmoottori muusta laitteistosta. Askelmoottorin uusi sijoitus näkyy kuviossa 47.



**Kuvio 47. Askelmoottorin uusi sijoituspaikka**

Askelmoottorin lisäksi tarkastuslaitteistossa käytettävä Pilzin ovikytkin aiheutti suuria häiriöitä anturiin. Kuten askelmoottorin kanssa meneteltiin, myös ovikytkimen kotelointia testattiin, mutta tällä järjestelyllä häiriöitä ei pystytty pudottamaan riittävästi. Ovikytkimen paikkaa ei myöskään voitu siirtää kauemmas anturista, joten ainoa vaihtoehto oli vaihtaa ovikytkin toisen tyyppiseen kytkimeen. Tilalle valittiin kuviossa 48 oleva Euchnerin valmistama ovikytkin.

Ovikytkimen vaihdon myötä myös ovipiirin kytkennät oli suunniteltava uudelleen. Myös Pilzin turvareleiden havaittiin aiheuttavan häiriöitä anturiin. Turvareleet vaihdettiin kokeilun vuoksi Euchnerin valikoimista löytyviin malleihin, mutta koska havaittavaa eroa releiden välillä ei ollut, vaihdettiin Pilzin releet takaisin niiden parempien ominaisuuksien takia.



**Kuvio 48. Euchnerin valmistama ovikytkin**

Anturin häiriöherkkyydestä johtuen häiriöiden poistaminen perustui kokeilemiseen. Erilaisilla lisämaadoituksilla ja eristämisillä häiriöt onnistuttiin laskemaan hyväksyttävälle tasolle. Tältä osin laitteiston testaaminen olikin asiaan paremmin perehtyneiden henkilöiden vastuulla.

## 8 Pohdinta

Opinnäytetyön tarkoituksena oli asiakkaalle valmistettavan liukurenkaan tarkastuslaitteiston sähkö- ja automaatio suunnittelu. Laitteiston tuli olla mahdollisimman yksinkertainen käyttää ja liukurenkaan molemmat pinnat tuli voida mitata ilman, että käyttäjän tarvitsee tehdä muutoksia laitteistoon. Tarkastuslaitteella piti kyetä tekemään mittauksia myös ilman tietokonetta käyttäen tiedonkeruuseen ja käsittelyyn ainoastaan RollScan-mittalaitetta. Laitteiston käyttäjärajapinnan tuli olla mahdollisimman yksinkertainen ja riittävän informatiivinen käytettävyyden kannalta. Suunnitelmien perusteella tehtiin laiteistohankinnat sekä rakennettiin laitteisto. Varmistus laitteiston onnistumisesta selviää, kun laitteisto toimitetaan asiakkaalle ja tämä ottaa sen aktiiviseen käyttöön. Omat kokemukset laitteen toiminnasta testivaiheessa olivat kuitenkin varsin myönteiset, joten laitteiston suunnittelun ja toteutuksen voidaan sanoa onnistuneen.

Laitteiston saattaminen suunnitelmista valmiiksi tuotteeksi oli pitkä ja mielenkiintoinen prosessi, josta ei haasteita puuttunut. Suuri takaisku koettiin heti projekti alkumetreillä, kun projekti päätettiin jäädyttää noin kuudeksi kuukaudeksi. Sähkösuunnittelu oli tuolloin saatu juuri käyntiin ja ensimmäiset kytkentäkuvat olivat hyvää vauhtia valmistumassa. Kun projektin pariin palattiin uudelleen tammikuussa 2013, jouduttiin suunnittelutyö aloittamaan käytännössä alusta, sillä mielessä välkkyneet visiot tulevasta laitteesta olivat unohtuneet muistiinpanoista huolimatta, sillä muut työt olivat sekoittaneet ajatukset pahemman kerran. Tästä huolimatta sähkösuunnittelu valmistui melko nopeassa tahdissa kevään aikana. Laitteiston kokoonpano alkoi keväällä jatkuen läpi koko prosessin aina syksyyn saakka. Omat mausteensa kokoonpanoon toivat haasteet mekaniikkasuunnittelussa, sillä laitteiston alkuperäinen mekaniikkasuunnittelija oli siirtynyt muihin tehtäviin. Hänen suunnitelmiaan korjaamaan käskettiin toinen suunnittelija, joka onnistuikin tehtä-



vässään erittäin hyvin. Suunnittelijoiden vaihtumisista ja kesälomista huolimatta laitteisto valmistui kesän aikana.

Sähkösuunnittelun suurin virhe oli kiistämättä virheellinen servovahvistimen kytkentä, jonka seurauksena rikottiin kaksi servovahvistinta. Muilta osin sähkösuunnittelussa vältyttiin suuremmilta kömmähdyksiltä. Esimerkiksi komponenttivalinnat, ovikytkin pois lukien, olivat onnistuneita. Joukkoon mahtui myös ilahduttavia onnistumisia, kuten esimerkiksi onnistunut laitteiston sähköistys sekä servomoottoreiden ensi yrittämällä toimiva turvakytkentä. Jos sähköistyksestä voisi muuttaa jotain näin jälkikäteen, olisi tuo muutos ehdottomasti käyttöjännitteen nostaminen 115 V:sta 230 V:iin. Näin johtimissa kulkevat virrat saataisiin pienemmiksi ja voitaisiin käyttää pienempää johdinkoa, joka taas toisi marginaalisen säästön komponenttikuluissa. Laitteiston kaikki muut komponentit toimivat sekä 115 V- että 230 V -käyttöjännitteellä, mutta valitut jäähdyttimet pystyvät käyttämään vain 115 V -käyttöjännitettä. Jäähdyttimet taas valittiin sen takia, että suunnitteluvaiheessa oletettiin, että laitteiston syöttö voidaan toteuttaa 115 V -käyttöjännitteellä. Tällöin ei olisi myöskään tarvittu laitteistoon nyt tarvittavaa ulkoista muuntajaa.

Automaatiosuunnittelu aloitettiin yhtä aikaa sähkösuunnittelun kanssa ja ensimmäinen versio toimintakuvauksesta valmistuikin nopeaan tahtiin sähkösuunnitelmien rinnalla. Yrityksessä ei ole aiemmin juurikaan tehty kattavia toimintakuvauksia, joten mallidokumentaatiota ei ollut saatavilla. Tämä seikka kirvoitti hyvin värikästäkin keskustelua aiheesta. Tarkastuslaitteistoon tehtyä toimintakuvausta käytettiin esimerkkinä korostettaessa toimintakuvauksen tärkeyttä niin sähkö- kuin automaatiosuunnittelussakin, tavoitteena saada aikaan muutoksia yrityksen toimintatapoihin. Laadittua toimintakuvausta käytettiin pohjana toisen prototyypilaitteiston suunnittelussa, joten voidaan sanoa tavoitteen täyttyneen ainakin jossain määrin. Automaatiosuunnittelua hankaloitti myös se, ettei käytettävissä ollut tarkempaa tietoa ja käsitystä toimintaperiaatteeltaan samankaltaisen laitteiston toiminnasta, koska kyseisestä

laitteistosta ei ollut toimintakuvausta saatavilla. Toimintaan liittyviä asioita tuli esille tipoittain, jolloin jo kertaalleen tehtyä suunnitelmaa oli korjattava tai tehtävä lisäyksiä, niinpä ensimmäinen ja lopullinen versio toimintakuvauksista poikkeavat toisistaan melkoisesti.

Automaatiosuunnittelun kannalta selkeimmät kehityskohteet ovat käyttäjärajapinnassa sekä logiikan ja tietokoneen välisessä kommunikaatiossa. Käyttöliittymää suunniteltaessa väkevin ohjenuora oli käytön helppous. Tästä johtuen käyttöliittymästä karsittiin pois kaikki ylimääräinen ja jätettiin vain käytön kannalta oleelliset merkkivalot ja painikkeet. Tästä seurauksena oli se, ettei häiriötiloista pystytä tehokkaasti informoimaan käyttäjää, koska muutamalla valolla ei voida kertoa kovinkaan monesta häiriötilasta yksinkertaisesti ja tehokkaasti. Jos valoja olisi lisätty käyttöliittymään, olisi siitä tullut sekava ja sitä myötä monimutkainen käyttää. Kaikkein järkevin tapa käyttöliittymän toteuttamisesta olisi mielestäni ollut perinteisten painikkeiden ja vipujen hylkääminen ja hankkia tilalle nykyaikainen kosketusnäyttö. Tuolloin suunnittelijalla olisi ollut vapaammat kädet käyttöliittymän pyörittelemiseen ja muokkaamiseen. Näytölle on helppoa luoda lisää painikkeita tai poistaa ylimääräisiä, perinteiseen paneeliin tehtävät muutokset ovat peruuttamattomia.

Kommunikaation toteutus logiikan ja tietokoneella pyörivän ViewScan-ohjelmiston välillä on toinen kehityskohde. Nykyinen toteutus National Instrumentsin IO-moduulin avulla on oikaisu sieltä, missä aita on matalin. Tähän ratkaisuun päädyttiin ajan säästämiseksi, sillä yrityksessä on aiemmin käytetty vastaavaa IO-moduulia, joten toiminta sen kanssa oli tuttua jo entuudestaan. Ohjelmoitava logiikka olisi voitu yhdistää tietokoneeseen Ethernet-kaapelilla, sillä logiikassa oli valmius tällaiseen. Keskusteluiden päätteeksi tätä ratkaisua ei pidetty aikataulullisesti järkevänä, koska olemassa oleva kommunikaatitoteutus logiikan ja ViewScan-ohjelmiston välillä olisi vaatinut muokkausta, sillä se on kehitetty käytettäväksi Siemensin logiikoiden kanssa.

Aiheeseen perehtyneen suunnittelijan aikataulu ei kireytensä takia mahdollistanut Ethernet-verkon kautta tapahtuvan kommunikoinnin toteuttamista.

Logiikkaohjelman tekeminen osoittautui lopulta pelättyä suoraviivaisemmaksi tehtäväksi, kiitos tästä lankeaa kattavalle toimintakuvaukselle. Logiikkaohjelman valmistumista viivästytti laitteiston sähköistyksen viivästyminen, sekä ongelmat mekaniikkasuunnittelussa. Ohjelmointiympäristöön ja käytettävien moottorinohjainten ja logiikan toimintaan päästiin perehtymään työpisteeseen viritellyn testipenkin turvin. Ensimmäinen versio logiikkaohjelmasta valmistuikin hyvissä ajoin, ennen kuin laitteisto oli vielä ajokunnossa. Kuten arvata saattaa, ensimmäinen ohjelmaversio ei toiminut täysin toivotulla tavalla, vaan ohjelmaa päivitettiin laitteiston testauksen edetessä. Lopullisen ohjelmaan voidaan olla tyytyväisiä, mutta muihin toimintoihin jäi paranneltavaa.

Logiikkaohjelmia on yhtä monenlaisia kuin on tekijöitäkin, joten ohjelman oikeellisuudesta voidaan keskustella loputtomiin. Servomoottoreiden ohjauksen toteutus taulukkomenetelmällä (Record selection mode) puolustaa kuitenkin paikkaansa sillä, että laitteistolla tullaan mittamaan vain ja ainoastaan yhdenlaisia näytteitä. Siinä mielessä ohjelmoinnissa ei tarvinnut ottaa huomioon mahdollisesti muutettavia parametreja. Ohjelman toteutuksessa olisi voinut tehdä muutamankin asian toisin. Tärkeimpänä muutoksena ja jatkokehityskohteena näkisin SoftMotion-ohjaustavan käyttöönoton. Nykyisellä ohjauksella, joka nojaa täysin FHPP:hen, voidaan toki toteuttaa yhdenaikaisia liikkeitä, mutta nuo liikkeet ovat toisistaan riippumattomia. Laitteisto ei kykene akseleiden synkronoituun ajoon, eli akseleita ei voida ajaa esimerkiksi kotiasemaan siten, että kaikki saavuttavat tuon aseman samalla hetkellä.

FHPP ei suoraan taivu tuollaiseen ohjaukseen. Kaksi moottorinohjainta olisi voitu yhdistää toisiinsa erillisellä kaapelilla ja määrittää niistä toinen ns. masteriksi. Tämä keino tuntui varsin keinotekoiselta ja kankealta, käytettäviä laitteita ohjataan kuitenkin kenttäväylää pitkin ja yhden lisäkaapelin kytkeminen kahden ohjaimen välille tuntuu vähintäänkin kyseenalaiselta. Ja tuolloin voi-

taisiin toteuttaa synkronoitu ohjaus vain kahden akselin välillä. Ohjattavia akseleita on kuitenkin kolme. Parempi keino olisi ollut toteuttaa moottoreiden ohjaus CoDeSysin SoftMotionilla. Tähän ei kuitenkaan ryhdytty, sillä tällöin Feston tekemät moottorinohjaustoimilohkot eivät olisi olleet käytettävissä ja tällöin esimerkiksi akseleiden parametointi olisi ollut huomattavasti monimutkaisempaa toteuttaa. Laitteistossa ei ollut tarvetta synkronoiduille liikkeille, joten FHPP oli järkevämpi ohjaustapa yksinkertaisuutensa ansiosta. Amatillisesti SoftMotion olisi ollut järkevämpi, mutta tuolloin pitkäksi venähtänyt aikataulu olisi venynyt entisestään.

Virhetilanteiden käsittely jäi puutteelliseksi, vaillinaiseksi ja kankeaksi. Laitteisto toki keskeyttää liikkeit mahdollisen väylävirheen tai ohjelman suorituksessa tapahtuneen virheen johdosta. Ongelma on siinä, miten asiasta ilmoitetaan käyttäjälle, mikä virhe on ollut kyseessä. Tämä ongelma taas juontaa juurensa yksinkertaiseen käyttöliittymään ja kommunikaatiotapaan logiikan ja tietokoneen välillä. Käytännössä logiikan ja tietokoneen välinen kommunikaatio on yksisuuntainen tie, NI:n IO-moduuli on pelkkä input-kortti, jolla muutetaan logiikan IO:lta saatava tieto Ethernet-kaapelin kautta välitettäväksi tiedoksi, jota ViewScan-ohjelmisto osaa tulkita. Jos kommunikaatio olisi toteutettu Ethernet-kaapelilla, olisi käyttäjää voitu todennäköisesti informoida tietokoneen ruudulla, mikä vikatila kulloinkin on päällä. Myös ViewScanin tilatietoja olisi tuolloin voitu käyttää hyväksi laitteiston ohjauksessa. Toisaalta, laitteisto vastaa nyt tilaajan toiveisiin yksinkertaisesta laitteesta, joka kykenee tietokoneesta riippumattomaan mittaustapahtumaan.

Ohjelmakoodissa käytettävien toimilohkojen toteutus jakoi mielipiteitä. Valittu tapa, jossa muuttujista kerättävillä tiedoilla muodostetaan ehtoja sekä JA-että TAI-piireillä, muodosti nopeasti hyvin pitkiä ja vaikeaselkoisia piirejä. Toinen keino olisi ollut tehdä useita pieniä funktioita, joita kutsumalla avustavissa toimilohkoissa muodostettaisiin tarvittava signaali. Mielestäni lukuisien pienien funktioiden luominen olisi tehnyt ohjelmakoodista vaikean tul-

kittavan, vaikkakin avustavat toimilohkot olisivat muuttuneet yksinkertaisemmiksi. Tämä pyrittiin korjaamaan kattavalla kommentoinnilla ohjelmakoodissa. Ohjelmakoodin sekvenssit olisi myös voitu toteuttaa nykyisten toimilohkojen sijaan Programeina (PRG). Etuna tässä tavassa olisi ollut kunkin sekvenssin ensimmäisen siirtoehdon yksinkertaistuminen, käytettäessä Programia ohjelmakutsu voidaan liittää haluttuun tapahtumaan (Event). Toisaalta Programia käytettäessä koko ohjelman suoritus hidastuu, mikä ei tässä tapauksessa ole ongelma, sekä ylimääräisten virheilmoitusten räjähdysmäisen kasvu projektia tarkistettaessa. Ohjelmointiympäristö ilmoittaa jokaisen tarkistuksen yhteydessä, jos usealla POU:lla on kirjoitusoikeus logiikan ulostuloihin, vaikka IO:n käsittelyyn käytettäisiinkin vain yhtä Programia. Muuttujia tarkastellessa voidaan todeta, että lähes kaikkien muuttujien määrittely globaaleiksi, oli tarpeetonta. Tämä oli kuitenkin tietoinen valinta ohjelmaa kirjoitettaessa, sillä näin vältettiin päällekkäiset nimeämiset.

Oppimisen ja ammatillisen kehittymisen kannalta liukurenkkaan tarkastuslaitteen automatisointi oli aiheena täydellinen, projekti oli kattava läpileikkaus siitä, mitä sähkö- ja automaatio suunnittelija urallaan kohtaa, alkaen laitevalinnoista aina lopputestaukseen ja valmiin laitteen lähettämiseen asiakkaalle. Sanonta ”tekemällä oppii” pätee tässäkin tapauksessa, sillä minulla ei ollut minkäänlaista kokemusta käytännön sähkösuunnittelusta tai siinä käytettävistä työkaluista, myös logiikkaohjelmoinnissa käytetty CoDeSys-ohjelmointiympäristö oli minulle entuudestaan tuntematon. Sähkösuunnittelussa huomioonotettavien asioiden määrä oli hämmentävän suuri ja standardeihin olikin tarttuva ahkerasti, erityisesti standardi SFS-EN ISO 60204-1 tuli projektin myötä tutuksi. Projektin automaatio-osio opetti erityisesti ns. pohjatyön merkityksen. Hyvin tehdyn automaatio suunnittelun turvin on helppoa aloittaa varsinainen logiikkaohjelmointi, joka etenee varsin nopeaan tahtiin, koska ohjelman pääpiirteet on jo valmiiksi mietitty, eikä ohjelmointivaiheessa

enää tarvitse pohtia mihin suuntaan ohjelmaa viedään, vaan voidaan keskittyä suunnitelman seuraamiseen ja toteutukseen.

Opinnäytetyö oli kasvattava kokemus. Se opetti nöyryyttä ja vastuunkantoa, omista virheistä oli kannettava vastuu. Omien virheiden myöntäminen olikin yksi vaikeimpia asioita, joita projektin aikana tuli vastaan. Välillä oli nieltävä ylpeytensä ja kuunneltava viisaampien kritiikkiä, ohjeita ja kehotuksia. Omien voimavarojen tunnistamisen nostaisin yhdeksi tärkeimmistä asioista, joita opin itsestäni opinnäytetyön aikana. Välillä opinnäytetyö oli yksinkertaisesti siirrettävä syrjään, etteivät omat voimat loppu kesken urakan. Projekti opetti paljon myös ihmisten välisestä kommunikaatiosta ja sen tärkeydestä. Ilman työtovereiden kanssa käytyjä keskusteluja lopullinen tarkastuslaitteisto olisi varmasti aivan erinäköinen.

## Lähteet

About UL. 2013. Yritysesittely UL:n sivustolla. Viitattu 18.3.2013.

<http://www.ul.com>, Company, About UL.

About us. N.d. Yritysesittely Stresstech Groupin sivustolla. Viitattu 15.1.2013.

<http://www.stresstechgroup.com>, about us.

Alanen, J. 2000. CAN - ajoneuvojen ja koneiden sisäinen paikallisväylä. Jarmo Alasen laatima koulutusmateriaali. Viitattu 30.1.2013.

[http://www.oamk.fi/~eero/Opetus/Ohjausjarjestelmat/CAN/CAN-perusteet\\_AlasenMateriaalia.pdf](http://www.oamk.fi/~eero/Opetus/Ohjausjarjestelmat/CAN/CAN-perusteet_AlasenMateriaalia.pdf)

Assembly and installation Type CMMS-AS. 2010. Servo-ohjaimen käyttöohje Feston sivustolla. Viitattu 17.4.2013.

<http://www.festo.com/net/SupportPortal/Files/52938/564228g1.pdf>

Assembly and installation Type CMMS-ST. 2010. Askelmoottorinohjaimen käyttöohje Feston sivustolla. Viitattu 17.4.2013.

<http://www.festo.com/net/SupportPortal/Files/52033/573125g1.pdf>

Barkhausen Noise Analysis. N.d. Tuote-esittely Stresstech Groupin sivustolla. Viitattu 23.1.2013. <http://www.stresstechgroup.com>, Products, Barkhausen Noise grinding burn and heat treat defect testing equipment, Barkhausen Noise Analysis.

CODESYS. 2013. Artikkelit Wikipediassa. Viitattu 13.6.2013.

<http://en.wikipedia.org/wiki/CoDeSys>.

Common AC power distribution configurations. 2010. Ametekin ohjeistus sähköjärjestelmistä. Viitattu 6.3.2013. <http://www.programmablepower.com>, Support, FAQ's, AC Power Distribution.

Controller Area Network. 2013. CAN-väylän teknologia. Artikkelit CAN in Automation –sivustolla. Viitattu 30.1.2013. <http://www.can-cia.org>, System design, Technology, CAN.

Eplan Electric P8. 2010. Tuote-esite Eplanin sivustolla. Viitattu 9.5.2013.

<http://www.eplan.fi>, esitteet, EPLAN Electric P8.

Festo Handling and Positioning Profile. 2010. FHPP tiedonsiirtoprofiilin ohjeistus Feston sivustolla. Viitattu 22.7.2013.

[http://www.festo.com/net/fi\\_fi/SupportPortal/Downloads/149957/52018/555696g1.pdf](http://www.festo.com/net/fi_fi/SupportPortal/Downloads/149957/52018/555696g1.pdf)

Festo Configuration Tool. N.d. FCT -ohjelmiston help -tiedosto. Viitattu 22.7.2013.

Fonselius, J., Pekkola, K., Selosmaa, S., Ström, M. & Välimaa, T. 1996. Automaatiolaitteet. Helsinki: Edita.

Fonselius, J., Rinkinen, J. & Vilenius, M. 1998. Servotekniikka. Helsinki: Edita.

IEC 61131-3: a standard programming resource. N.d. Artikkelit PLCopen -sivustolla. Viitattu 13.6.2013.

[http://www.plcopen.org/pages/tc1\\_standards/downloads/intro\\_iec.pdf](http://www.plcopen.org/pages/tc1_standards/downloads/intro_iec.pdf).

Industrial control panels for the North American market. 2010. Määräyksiä ja ohjeita teollisuuden ohjauskaappien valmistukseen. Fuerth: Siemens AG.

Inputs and outputs. 2010. Perustietoa ohjelmoitavien logiikoiden I/O-yksiköistä. Viitattu 9.5.2013. <http://engineeronadisk.com>, eBook: Automating Manufacturing Systems with PLC, Inputs and outputs.

Introduction into IEC 61131-3 Programming Languages. N.d. Artikkelit PLCopen -sivustolla. Viitattu 13.6.2013. <http://www.plcopen.org>, TC1 - Standards, IEC 61131-3.

Koneturvallisuus. 2008. Työsuojeluhallinnon ohje koneiden valmistajalle. Viitattu 2.3.2013.

[http://tyosuojelujulkaisut.wshop.fi/documents/2008/12/tso\\_16-2009.pdf](http://tyosuojelujulkaisut.wshop.fi/documents/2008/12/tso_16-2009.pdf)

Koneturvallisuus ja CE-merkintä. 2010. Stresstech Oy:n menetelmäohje. Viitattu 4.3.2013. Julkaisu saatavissa Stresstech Oy:n tietojärjestelmässä.

Laatu automaatioissa - parhaat käytännöt. 2001. Helsinki: Suomen automaatioseura ry.

Lapp Kabel muuttuneet UL -määräykset NFPA79-2007 ja -2012. 2010. Muutoksia koneenrakennuksen kaapelivalintaan USA:n markkinoilla. Ohjeistus SKS:n sivustolla. Viitattu 19.3.2013. <http://www.sks.fi>, SKS automaatio, Kaapelit ja johtimet, Uutuudet ja uutiset, Lapp Kabel Muuttuneet UL -määräykset NFPA79-2007 ja -2012.

Motor controllers CMMS-AS, for servo motors. 2012. Servomoottorien teknisiä tietoja Feston sivustolla. Viitattu 12.3.2013.

[https://www.festo.com/cat/fi\\_fi/data/doc\\_engb/PDF/EN/CMMS-AS\\_EN.PDF](https://www.festo.com/cat/fi_fi/data/doc_engb/PDF/EN/CMMS-AS_EN.PDF)

Ohjelmointikielten syntaksista ja semantiikasta. N.d. Oulun yliopiston tietojenkäsittelytieteiden laitoksen luentomateriaali ohjelmointikielten periaatteista. Viitattu 13.6.2013.

[http://www.tol oulu.fi/kurssit/okp/Luennot/OKP\\_Syntaksi.html](http://www.tol oulu.fi/kurssit/okp/Luennot/OKP_Syntaksi.html).



RollScan 250. N.d. Tuote-esittely Stresstech Groupin sivustolla. Viitattu 9.2.2013. <http://www.stresstechgroup.com>, Products, Barkhausen Noise grinding burn and heat treat defect testing equipment, Analyzers, RollScan 250.

Saha, H. 2006. CANopen perusteet. Skannattu artikkeli CANopen.fi – sivustolla. FLUID Finland 1, 6-11. Viitattu 30.1.2013. <http://www.canopen.fi/artikkelit/CANopen.pdf>

Servo motors EMMS-AS. 2013. Servomoottoreiden teknisiä tietoja Feston sivustolla. Viitattu 26.2.2013. [https://www.festo.com/cat/fi\\_fi/data/doc\\_engb/PDF/EN/EMMS-AS\\_EN.PDF](https://www.festo.com/cat/fi_fi/data/doc_engb/PDF/EN/EMMS-AS_EN.PDF)

SFS-EN ISO 12100. 2010. Koneturvallisuus. Yleiset suunnitteluperiaatteet, riskin arviointi ja riskin pienentäminen. 3.p. Helsinki: Suomen standardisoimisliitto. Viitattu 13.3.2013. <http://www.jamk.fi/kirjasto>, Nelli-portaali, SFS Online.

SFS-EN ISO 60204-1. 2006. Koneturvallisuus. Koneiden sähkölaitteisto. Osa 1: Yleiset vaatimukset. 3.p. Helsinki: Suomen standardisoimisliitto. Viitattu 29.3.2013. <http://www.jamk.fi/kirjasto>, Nelli-portaali, SFS Online.

Software ViewScan. N.d. Tuote-esittely Stresstech Groupin sivustolla. Viitattu 6.5.2013. <http://www.stresstechgroup.com>, Products, Barkhausen Noise grinding burn and heat treat defect testing equipment, Software ViewScan.

Stepper motors EMMS-ST. 2013. Askelmoottoreiden teknisiä tietoja Feston sivustolla. Viitattu 26.2.2013. [https://www.festo.com/cat/fi\\_fi/data/doc\\_engb/PDF/EN/EMMS-ST\\_EN.PDF](https://www.festo.com/cat/fi_fi/data/doc_engb/PDF/EN/EMMS-ST_EN.PDF)

Teknisiä tietoja ja taulukoita -käsikirja. 2000. PDF-julkaisu. 9.p. 22.7.2013. Julkaisu saatavilla Stresstech Oy:n tietojärjestelmässä.

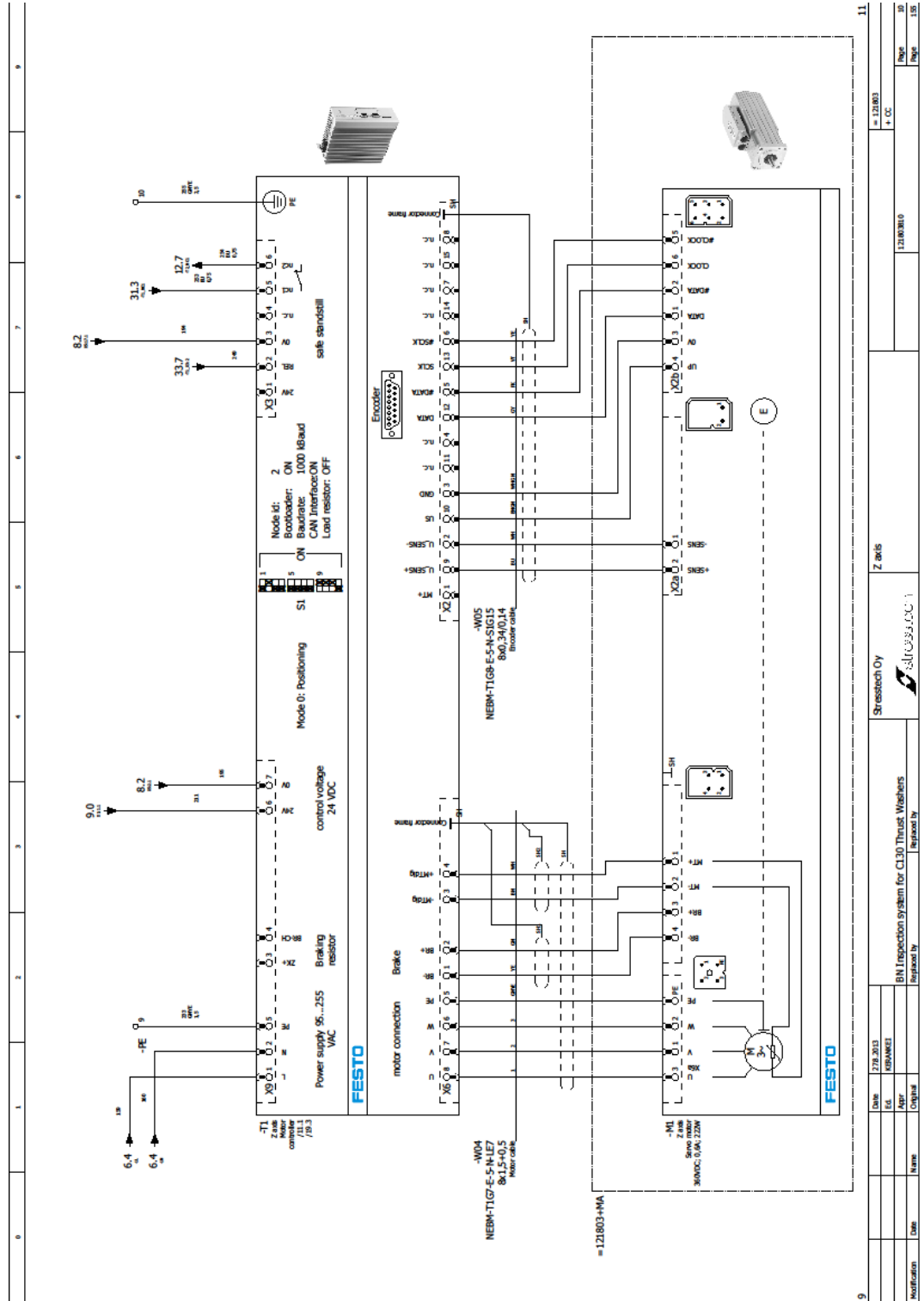
Tilinpäätöstilaisuus Stresstech Oy:n henkilöstölle Laajavuorella talvella 2013.

Turvareleet ja hätäpysäytys. 2013. Tuote-esittely Siemensin sivustolla. Viitattu 24.2.2013. <http://www.siemens.fi>, Industry, Teollisuus, Tuotteet ja järjestelmät, Kone- ja prosessiturvallisuus sekä ATEX, Koneturvallisuus, Turvareleet ja hätäpysäytys.

Younkin, G. 1996. Industrial servo control systems: fundamentals and applications. New York: MARCEL DEKKER.

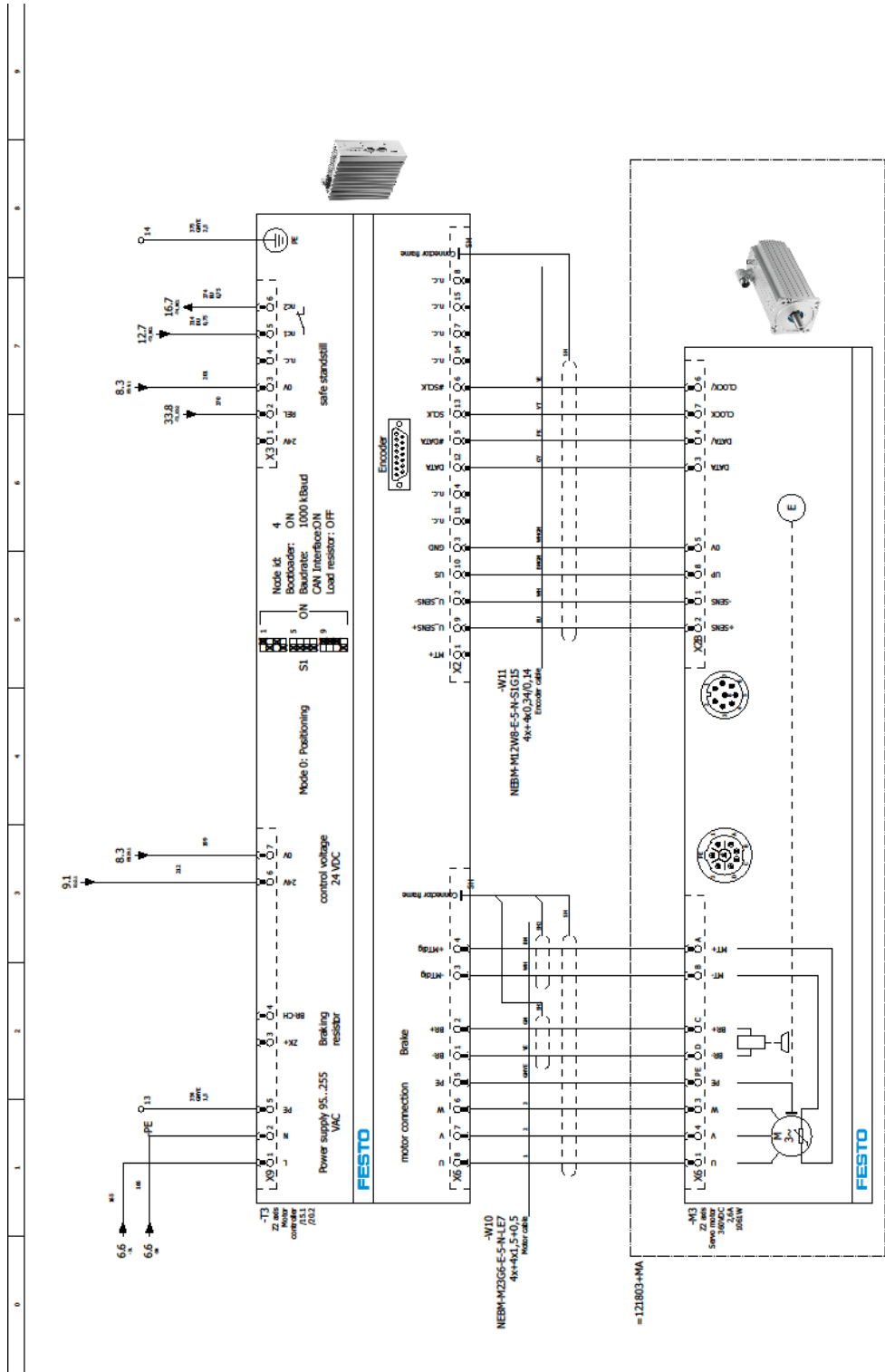
# Liitteet

## Liite 1. Turvakytkennät



9	10	11
12	13	14
15	16	17
18	19	20
21	22	23
24	25	26
27	28	29
30	31	32
33	34	35
36	37	38
39	40	41
42	43	44
45	46	47
48	49	50
51	52	53
54	55	56
57	58	59
60	61	62
63	64	65
66	67	68
69	70	71
72	73	74
75	76	77
78	79	80
81	82	83
84	85	86
87	88	89
90	91	92
93	94	95
96	97	98
99	100	101
102	103	104
105	106	107
108	109	110
111	112	113
114	115	116
117	118	119
120	121	122
123	124	125
126	127	128
129	130	131
132	133	134
135	136	137
138	139	140
141	142	143
144	145	146
147	148	149
150	151	152
153	154	155
156	157	158
159	160	161
162	163	164
165	166	167
168	169	170
171	172	173
174	175	176
177	178	179
180	181	182
183	184	185
186	187	188
189	190	191
192	193	194
195	196	197
198	199	200
201	202	203
204	205	206
207	208	209
210	211	212
213	214	215
216	217	218
219	220	221
222	223	224
225	226	227
228	229	230
231	232	233
234	235	236
237	238	239
240	241	242
243	244	245
246	247	248
249	250	251
252	253	254
255	256	257
258	259	260
261	262	263
264	265	266
267	268	269
270	271	272
273	274	275
276	277	278
279	280	281
282	283	284
285	286	287
288	289	290
291	292	293
294	295	296
297	298	299
300	301	302
303	304	305
306	307	308
309	310	311
312	313	314
315	316	317
318	319	320
321	322	323
324	325	326
327	328	329
330	331	332
333	334	335
336	337	338
339	340	341
342	343	344
345	346	347
348	349	350
351	352	353
354	355	356
357	358	359
360	361	362
363	364	365
366	367	368
369	370	371
372	373	374
375	376	377
378	379	380
381	382	383
384	385	386
387	388	389
390	391	392
393	394	395
396	397	398
399	400	401
402	403	404
405	406	407
408	409	410
411	412	413
414	415	416
417	418	419
420	421	422
423	424	425
426	427	428
429	430	431
432	433	434
435	436	437
438	439	440
441	442	443
444	445	446
447	448	449
450	451	452
453	454	455
456	457	458
459	460	461
462	463	464
465	466	467
468	469	470
471	472	473
474	475	476
477	478	479
480	481	482
483	484	485
486	487	488
489	490	491
492	493	494
495	496	497
498	499	500
501	502	503
504	505	506
507	508	509
510	511	512
513	514	515
516	517	518
519	520	521
522	523	524
525	526	527
528	529	530
531	532	533
534	535	536
537	538	539
540	541	542
543	544	545
546	547	548
549	550	551
552	553	554
555	556	557
558	559	560
561	562	563
564	565	566
567	568	569
570	571	572
573	574	575
576	577	578
579	580	581
582	583	584
585	586	587
588	589	590
591	592	593
594	595	596
597	598	599
600	601	602
603	604	605
606	607	608
609	610	611
612	613	614
615	616	617
618	619	620
621	622	623
624	625	626
627	628	629
630	631	632
633	634	635
636	637	638
639	640	641
642	643	644
645	646	647
648	649	650
651	652	653
654	655	656
657	658	659
660	661	662
663	664	665
666	667	668
669	670	671
672	673	674
675	676	677
678	679	680
681	682	683
684	685	686
687	688	689
690	691	692
693	694	695
696	697	698
699	700	701
702	703	704
705	706	707
708	709	710
711	712	713
714	715	716
717	718	719
720	721	722
723	724	725
726	727	728
729	730	731
732	733	734
735	736	737
738	739	740
741	742	743
744	745	746
747	748	749
750	751	752
753	754	755
756	757	758
759	760	761
762	763	764
765	766	767
768	769	770
771	772	773
774	775	776
777	778	779
780	781	782
783	784	785
786	787	788
789	790	791
792	793	794
795	796	797
798	799	800
801	802	803
804	805	806
807	808	809
810	811	812
813	814	815
816	817	818
819	820	821
822	823	824
825	826	827
828	829	830
831	832	833
834	835	836
837	838	839
840	841	842
843	844	845
846	847	848
849	850	851
852	853	854
855	856	857
858	859	860
861	862	863
864	865	866
867	868	869
870	871	872
873	874	875
876	877	878
879	880	881
882	883	884
885	886	887
888	889	890
891	892	893
894	895	896
897	898	899
900	901	902
903	904	905
906	907	908
909	910	911
912	913	914
915	916	917
918	919	920
921	922	923
924	925	926
927	928	929
930	931	932
933	934	935
936	937	938
939	940	941
942	943	944
945	946	947
948	949	950
951	952	953
954	955	956
957	958	959
960	961	962
963	964	965
966	967	968
969	970	971
972	973	974
975	976	977
978	979	980
981	982	983
984	985	986
987	988	989
990	991	992
993	994	995
996	997	998
999	1000	1001





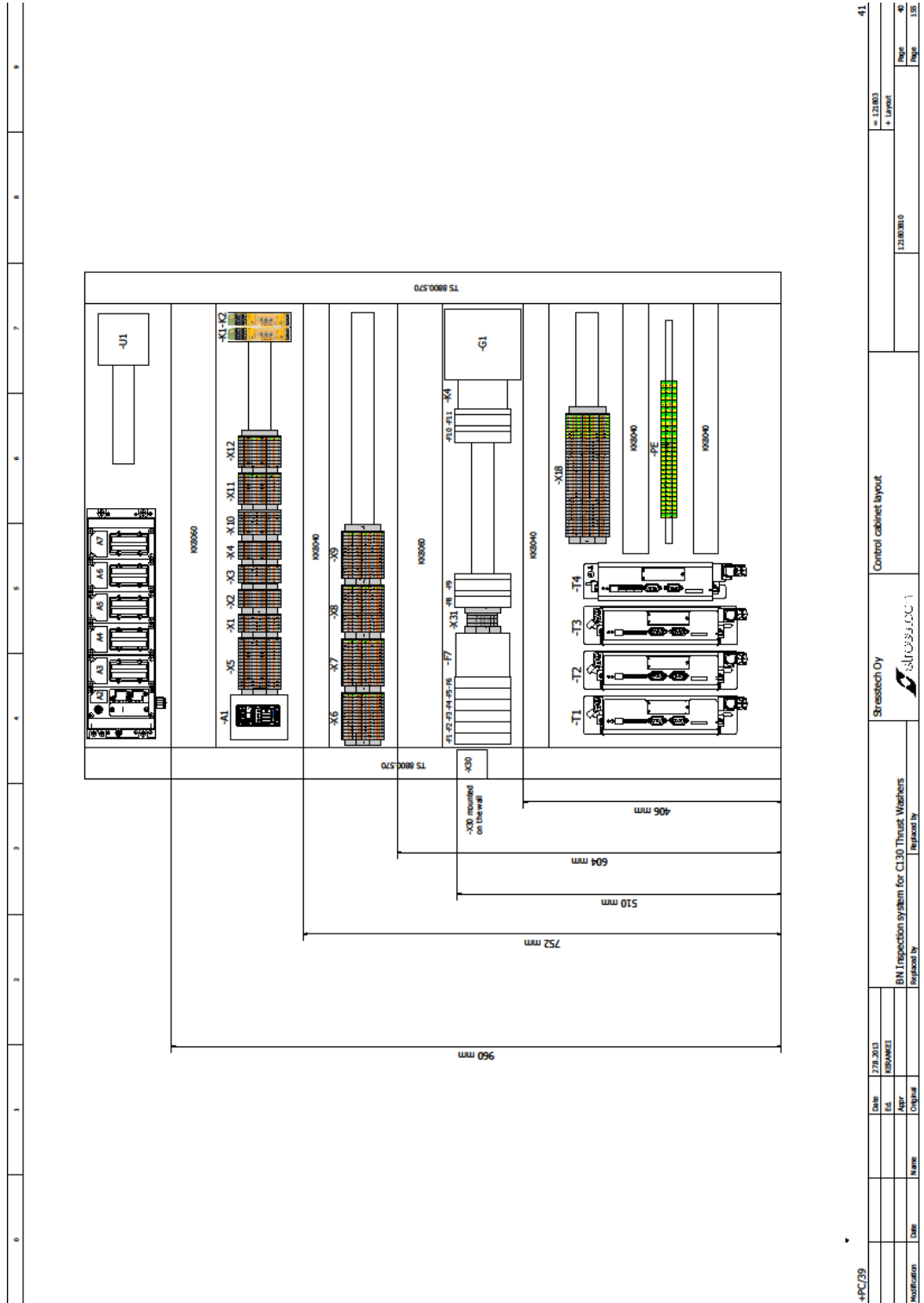
13	Date	27.01.2013	Strattech Oy	Z2 axis	121803	15
	Author	HEIKKINEN			+ CC	
	Name	Original	BN Inspection system for C130 Thrust Washers		121803	14
	Date		Revised by			15







# Liite 2. Sähkökaapin layout



+PC/29		Date: 27.8.2013		Sheet: 41	
Author: M. J. J. J.		Revision: 1		Date: 12.10.2010	
Title: BN Inspection system for C130 Thrust Washers		Project: Control cabinet layout		Page: 41	
Revised by:		Revised by:		Page: 15	
Name:		StressTech Oy		Page: 15	
Date:		S.T.039.J.C.C.1		Page: 15	





## Liite 4. Toimintakuvaus



Keijo Keränen

TOIMINTAKUVAUS

1 (16)

2013-07-19

### 1 BN Inspection system for Washer ring

Liukurenkaan sisä- ja ulkopinnan tarkastuslaite Barkhausen-menetelmällä.

#### 1.1 Toimintaperiaate

Tarkastuslaite suorittaa ennalta määritellyn sekvenssin, joka liikuttaa mittausanturia pitkin liukurenkaan pintaa. Toiminnan käynnistämiseen vaaditaan toimenpiteitä operaattorilta, joten laite ei ole täysin automaattinen.

#### 1.2 Laitteisto

Laitteisto liikkeiden toteuttamiseen:

- 3 servomootoria ohjaimineen
- 1 askelmoottori ja ohjain
- 1 DC-moottori ja ohjainkortti
- 3 lineaariakselia
- 6 induktiivista kytkintä lineaariakseleihin
- ohjelmoitava logiikka, 40I + 40O, CANopen
- ohjauspaneeli CP

Laitteisto kytketään 3-vaihesyöttöön, pääjännite on 480 V/60 Hz ja vaihejännite 277 V/60 Hz. Laitteistoon kuuluvalla muuntajalla käyttöjännitteeksi saadaan pääjännitteeksi 200 V ja vaihejännitteeksi 115 V. Komponenttien käyttöjännite 115 V. Servomootorit ovat DC-servoja, käyttöjännite 360 V DC, askelmoottorin käyttöjännite 48 V DC ja DC-moottorin käyttöjännite 24 V DC. Servo-ohjaimet ja askelmoottorin ohjain kytketään CAN-väylään, jonka kautta hoidetaan servojen ohjaus, protokollana CANopen-protokollaperhe. DC-moottorin ohjaus toteutetaan mootto-

rinohjainkortilla, jonka ohjaukseen käytetään logiikan IO:ta. Laitteiston 24 V DC ohjauksjännite tuotetaan 10 A:n tasavirtalähteellä ja tehonjakajalla, joka toimii myös ylivirtasuojana.

Lisäksi laitteistoon kuuluu Pc-kaappi ja sähkökaappi.

Pc-kaappi:

- tietokone
- RollScan 250
- magnetointivahvistin
- 2 pistorasiaryhmää, joista toinen on varattu magnetointivahvistimelle
- jäähdytin
- IO-moduuli kommunikaatioon PLC:n ja PC:n välille
- tulostin
- 2 kpl usb-sarjavälämuuntimia

Sähkökaappi:

- johdonsuojakatkaisijat ja vikavirtasuoja
- riviliitinryhmät
- 2 turvarelettä
- pääkytkin
- 2 hätäpysäytys-painiketta
- kaapelointi

### 1.3 Käyttäjiliityntä HMI

Laitteistoa ohjataan ohjauspaneelin kautta. Ohjauspaneelissa on seuraavat ohjauskytkimet ja merkkivalot:

- **S1**, punainen hätäpysäytys-painike, **EMG**
- **S2**, vihreä valopainike, **START/RUN**
- **S3**, musta painike, **STOP**
- **S4**, sininen valopainike, **FAULT**
- **S5**, valkoinen valopainike, **HOME**
- **S6**, valkoinen valopainike, **DOOR ACK**
- **S7**, musta painike, **JOG+**
- **S8**, musta painike, **JOG-**
- **S9**, valintakytkin, **TEMPLATE SELECTOR**
- **S10**, valintakytkin, **JOG SELECTOR**
- **S13**, punainen valopainike, **EMG ACK**
- **S14**, avainkytkin, **ACTIVATE JOG**
- **H8**, vihreä merkkivalo, **SERVO ENABLE**
- **H9**, vihreä merkkivalo, **JOG ACTIVE**
- **H11**, vihreä merkkivalo **ROLLSCAN MEASURING**
- **H12**, vihreä merkkivalo, **POWER**

Ohjauspaneelin lisäksi käyttäjälle annetaan valomajakalla tietoa laitteiston toiminnasta. Majakassa on kolme valoelementtiä:

- punainen
- vihreä
- valkoinen

Merkkivalot joko välkkyvät tai palavat staattisesti. Välkkyvällä valolla on eri merkitys kuin jatkuvasti palavalla. Seuraavassa on selitetty staattisesti palavien sekä vilkkuvien valojen merkitys.

#### **S2 START**

Merkkivalo syttyy, kun laitteisto suorittaa mittaussekvenssiä.

#### **S4 FAULT**

Valon vilkkuessa yhdessä moottorinohjaimista on toimintahäiriö. Valon palaessa on tapahtunut joko väylään viittaava virhe, tai logiikkaohjelman suorituksessa on tapahtunut virhe.

#### **S5 HOME**

Valon vilkkuessa servomoottorien referenssiajo on suorittamatta. Valon palaessa staattisesti voidaan kotiajo aktivoida tai kotiajosekvenssi on käynnissä.

#### **S6 DOOR ACK**

Valon palaessa turvaovi on auki tai ovea valvovaa turvarelettä ei ole kuitattu oven sulkemisen jälkeen.

#### **S13 EMG ACK**

Valo palaa, jos hätäpysäytys on aktivoitu tai hätäpysäytystä valvovaa turvarelettä ei ole kuitattu hätäpysäytyksen jälkeen.

**H8 SERVO ENABLE**

Merkkivalo palaa, kun servojen ohjauselektronikka on käynnissä eikä servoissa ole vikatiloja aktiivisena.

**H9 JOG ACTIVE**

Merkkivalo palaa, jos käsiajo on aktivoitu.

**H11 ROLLSCAN MEASURING**

Merkkivalo palaa, kun mittaus on käynnissä.

**H12 POWER**

Merkkivalo palaa, kun laitteisto on jännitteinen. Valo syttyy heti, kun virta kytketään laitteistoon.

**Valomajakan vihreä valo**

Valon palaessa mittaussekvenssi on käynnissä.

**Valomajakan valkoinen valo**

Merkkivalon palaessa laitteiston turvaovi on auki, tai turvaovea valvova turvarele on kuittaamatta oven sulkemisen jälkeen. Valon vilkkuesssa lukitusrengas on auki, tai lukitusrengasta valvova anturi on viallinen.

**Valomajakan punainen valo**

Merkkivalo palaa, kun laitteisto on hätäpysäytetty tai hätäpysäytystä valvova turvarele on kuittaamatta.

**1.4 Sekvenssin valinta**

Laitteistossa on määritelty kaksi sekvenssiä, joiden mukaan anturia liikutetaan. Kytkimellä S9 valitaan sopiva sekvenssi. Valintakytkimessä on neljä asentoa:

1. tasainen pinta
2. kaareva pinta
3. tasainen pinta referenssi

#### 4. kaareva pinta referenssi

Asentoja 3 ja 4 käytetään referenssimittauksissa. Normaaliolosuhteissa käytetään asentoja 1 ja 2. Kun sekvenssi on käynnistetty, ei valintakytkimen S9 asennolla ole enää merkitystä. Kytkimen asentoa ei kuitenkaan tule muuttaa kesken ajon.

### 1.5 Käsiajo

Laitteistoa voidaan ajaa myös käsiajolla. Avainkytkimellä S14 aktivoidaan käsiajo, valintakytkimellä S10 valitaan ohjattava servomoottori. Lisäksi pyörikykseen käytettävää askelmoottoria sekä anturin kääntämiseen käytettävää sähkömoottoria voidaan ajaa käsin.

Valintakytkimen asennot ovat:

0. Käsiajo pois päältä
1. Z-akselin käsiajo
2. Z2-akselin käsiajo
3. Anturin käännön käsiajo
4. Y-akselin käsiajo
5. Pyörikyksen käsiajo

Kun käsiajo on aktiivinen, ilmoitetaan tästä ohjauspaneelin merkivalolla H9. Käsiajossa akseleita liikutetaan painikkeilla S7 ja S8. S7 ajaa akselia ulos, S8 ajaa akselia sisään. Jos painikkeita S7 ja S8 painetaan yhtäaikaan, liikettä ei tapahdu. Käsiajo lopetetaan kääntämällä valintakytkin S10 ja avainkytkin S14 0-asentoon. Käsiajoa ei voida aktivoida kesken mittaussekvenssin.

### 1.6 Anturin suojaus törmäykseltä

Anturin törmäys mitattavaan kappaleeseen estetään induktiivisilla kytkimillä, jotka havaitsevat anturin varren kallistumisen. Kun anturin varsi kallistuu riittävästi, kytkin ei enää tunnista anturin varretta ja näin pysäyttää tarkastuslaitteiston kaikki liikkeet ja sytyttää valotomin punaisen valoelementin. Virhetilanteesta päästään

ajamalla käsiäjolla Y-akselia mitattavasta kappaleesta pois päin. Y-akselia ei voida ajaa kohti mitattavaa kappaletta, jos kytkin ei tunnista anturin vartta. Kun anturinvarti on jälleen kyllin suorassa, voidaan servot ajaa kotiasemiinsa valopainikkeella S5. Vian ilmettyä on FCT-työkalulla tarkistettava Y-akselin paikoitus ja tehtävä tarvittavat korjaukset servo-ohjaimen tallennettuun asemataulukkoon.

### 1.7 Servojen kotiajo

Kotiajo käynnistyy valopainikkeesta S5. Käynnistettäessä laite ensimmäisen kerran servot etsivät asematietonsa S5 painikkeesta painettuna. Tuona aikana servot etsivät toisen päätyrajan ja nollapulssin. Asematiedon etsintään käytettävä suunta määritellään FCT-työkalulla. Tämän jälkeen servot ajetaan FCT:llä määritellyn akselin nollapisteeseen. Tämä nollapiste toimii samalla projektin nollapisteenä sekä servon kotiasemana. Onnistuneen asematiedon hakemisen jälkeen servot ajetaan kotiasemaan, ohjaimen tallennetun taulukon arvoon nro 1. Servot eivät liiku tämän vaiheen aikana, sillä taulukon 1 asema on sama, kuin aiemmin mainitut projektiin ja akselin nollapisteet.

Kotiajolle on olemassa kaksi erillistä sekvenssiä. Käytettävä sekvenssi riippuu valintakytkimen S9 asennosta. Erilliset sekvenssit tarvitaan, ettei anturia ajeta päin mitattavaa liukurengasta. Jos mittaussekvenssi keskeytetään, hypätään PLC-ohjelmassa kotiajosekvenssiin, joka käynnistetään painikkeella S5. Kotiajosekvenssi ei käynnisty, jos valintakytkimen S9 asentoa on muutettu mittaussekvenssin aikana. Kotiajosekvenssi voidaan tällöin käynnistää palauttamalla S9 takaisin alkuperäiseen asemaan ja tämän jälkeen painamalla painiketta S5.

Kotiajo on suoritettava, jos käsiäjo on käytetty aktiivisena. Tämä siitä huolimatta, vaikka laitteita olisi siirretty lainkaan.

### 1.8 Mittaussekvenssin käynnistäminen

Mittaussekvenssi voidaan käynnistää kun:

- virta on kytkettynä



- mitattava kappale on paikallaan
- lukituskauluri on kiinni
- turvaovi on suljettuna
- servoissa ei ole häiriöitä
- servot ovat kotiasemissa
- servojen referenssiajo tehtynä
- hätäseis ei ole aktiivinen
- käsiajo ei ole aktiivinen

Tällöin merkkivalot H8 ja H11 palavat. Mittaussekvenssi käynnistetään valopainikkeella S2. Mittaussekvenssin käynnistäminen sytyttää valopainikkeen S2 sekä valomajakan vihreän valon.

#### 1.9 Mittaussekvenssin pysäyttäminen

Mittaussekvenssi voidaan pysäyttää painikkeella S3, joka pysäyttää servomootorit, askelmoottorin ja anturin kääntämiseen käytettävän DC-moottorin poistamalla näiltä käyntiä varten tarvittavan STOP-signaalin. Pysäyttämällä mittaussekvenssi myös RollScanin triggauksen katkeaa ja magnetointia ohjaava kontaktori K4 avautuu. Sekvenssiä ei voida jatkaa, mikäli toiminta on pysäytetty. Servot palautetaan kotiasemiinsa painikkeella S5.

Sekvenssin pysäyttäminen sytyttää ohjauspaneelin merkkivalon S5. Valopainike S2 ja valomajakan vihreä valo sammuvat.

#### 1.10 Mittaussekvenssin hätäpysäytys

Mittaussekvenssi voidaan hätäpysäyttää hätäpysäytys-painikkeilla S1 ja S15. Turvaovien avaaminen vaikuttaa hätäseis-painikkeen tavoin. Hätäpysäytys-piiriä ja turvaovipiiriä valvotaan turvareleillä K1 ja K2. Servo-ohjaimien ohjauselektro-



Keijo Keränen

TOIMINTAKUVAUS

9 (16)

2013-07-19

niikan ja servomoottorien virtalähteiden ohjauksen jännitteet kulkevat sarjassa turvareleiden kautta. Kun turvarele katkaisee piirin, ohjauselektronikan jännite katkeaa välittömästi. Tämän seurauksena ohjain hidastaa ja lopulta pysäyttää liikkeen. Liikkeen pysähtyttyä ohjaus kytketään pois päältä. Virtalähteiden ohjauksen jännitteet ovat tästä johtuen kytketty turvareleiden viiveellisiin koskettimiin, jotta pysähtyminen tapahtuu hallitusti servo-ohjaimen toteuttamana.

Hätäpysäytys-painikkeen painamisen seurauksena:

- liike pysähtyy
- valomajakkan punainen valo syttyy
- valopainike S13 syttyy
- valopainike S2 sammuu

Turvaopiiriin lauetessa:

- liike pysähtyy
- valomajakkan punainen valo syttyy
- valopainike S6 syttyy
- valopainike S2 sammuu

Hätäpysäytys-piiri kuitataan valopainikkeella S13. Turvaopiiri kuitataan valopainikkeella S6. Kuitaamisen jälkeen servot on ajettava takaisin kotiasemiinsa valopainikkeella S5. Hätäpysäytys voidaan tehdä myös pääkytkimellä Q1. Tällöin laitteistolta katkaistaan jännitesyöttö ja laitteisto jää jännitteettömäksi. Jännitteen katkeamisen jälkeen on suoritettava servojen referenssiajo. Sekvenssiä ei voida jatkaa hätäpysäytyksen jälkeen.

#### 1.11 Mittaussekvenssin kulku

Jos mittaussekvenssin käynnistysehdot täyttyvät, voi operaattori käynnistää mittaussekvenssin. Mittaussekvenssiä on kaksi, mitattaville pinnoille molemmille omansa. Sekvenssin kulku on molemmissa sekvensseissä samanlainen, erot löy-

tyvät anturin asennosta sekä asemista. Sekvenssi on automatisoitu, mutta operaattori voi halutessaan keskeyttää sekvenssin.

#### 1.11.1 Mittauspohjan valinta ViewScanissa

Mittaussekvenssin ensimmäinen vaihe on mittauspohjan valinta. Tieto valitusta mittauspohjasta välitetään logiikalta tietokoneelle National Instrumentsin IO-moduulin avulla. Mittauspohjan valintaan käytetään kolmea logiikan ulostuloa. Kahta ulostuloista käytetään sopivan mittauspohjan valintaan, yhdellä vahvistetaan valinta. Päälle laitettavat ulostulot riippuvat valintakytkimen S9 asennosta. Seuraavassa taulukossa on esitetty päälle kytkettävien ulostulojen ja valintakytkimen asennon suhde toisiinsa.

**Taulukko 1 Mittauspohjan valinta**

Valintakytkimen S9 asento	Ulostulon QX8.3 tila	Ulostulon QX8.2 tila	Valittu mittauspohja
1	0	0	Template 1
2	0	1	Template 2
3	1	0	Reference 1
4	1	1	Reference 2

Template 1 mittauspohjaa käytetään mitattaessa liukurenkaan tasaista pintaa. Template 2:ta käytetään kaarevan pinnan mittauksessa. Referenssimittauspohjia käytetään mitattaessa referenssinäytteitä, Reference 1 suoralle ja Reference 2 kaarevalle pinnalle.

Sekvenssin ensimmäisessä askeleessa halutut ulostulot asetetaan aktiivisiksi. Ensimmäinen askel ajastetaan pysymään päällä 500ms ajan. Tällä varmistetaan siitä, että NI:n IO-moduuli ehtii reagoimaan signaaliin. Tämän jälkeen sekvenssi siirtyy toiseen askeleeseen, jossa ulostulo QX8.0 asetetaan päälle. Toista askelta

pidetään päällä 500ms ajan, jonka jälkeen mittauspohjan valintaan tarvittavat ulostulot asetetaan pois päältä.

#### 1.11.2 Anturin kääntö

Anturin kääntö tapahtuu automaattisesti riippuen anturin ja valintakytkimen S9 asennosta. Anturin kotiasema on suojassa magnetointirullien alla. Anturin kääntö tehdään kääntöasemassa, sillä kotiasemassa anturi ei mahdu kääntymään. Anturi ajetaan kääntöasemaan Y-akselilla. Mitattaessa liukurengaan tasaista pintaa anturi ajetaan takaisin kotiasemaan käännön jälkeen. Kaarevaa pintaa mitattaessa anturi ajetaan käännön jälkeen ns. väliasemaan.

Anturin kääntöä voidaan kytkimien lisäksi ajastimella. Askeleen kestoksi on määritetty 2s. Jos kytkimet eivät aktivoidu määritellyn ajan sisällä, menee laitteisto virhetilaan.

#### 1.11.3 Magnetointirullien nosto

Magnetointirullilla sekä magnetoidaan liukurengas, että pyöritetään liukurengasta mittauksen aikana. Pyöritystä varten magnetointirullat nostetaan ylös kiinni liukurengaaseen. Nosto tapahtuu kahdessa osassa. Ensimmäisessä vaiheessa Z2-akseli ajetaan ylös n.1 mm:n päähän liukurengaasta. Kontaktori K4 ohjataan vetäneeksi sillä hetkellä, kun Z2-akselin liike pysähtyy. Tämän jälkeen, noston toisessa vaiheessa liukurengas nostetaan irti propellin varresta.

#### 1.11.4 Liukurengaan demagnetointi

Liukurengas demagnetoidaan RollScan 250:llä. Kun magnetointirullat on nostettu kiinni liukurengaaseen, käynnistyy pyöritys. Samalla hetkellä pyörityksen käynnistyessä ohjataan logiikan ulostulo QX8.4 päälle, joka ohjaa RollScania käynnistämään demagnetointirampin. Liukurengas pyörii demagnetoinnin aikana yhden täyden kierroksen. Pyörähdysnopeus on ~19 rpm (jolloin liukurengaan nopeus on 5 rpm eli 50mm/s).

#### 1.11.5 Anturin asema mittauksen alkaessa

Anturi ajetaan kiinni mitattavan pinnan sisäreunaan. Siirto tehdään Z- ja Y-akseleilla. Akseleita ei ajeta yhtä aikaa, vaan yksi liike kerrallaan. Kaarevaa pintaa mitattaessa anturin siirto tapahtuu neljässä osassa.

#### 1.11.6 Pyöryksen käynnistys

Liukurenkaan pyörytys toteutetaan magnetointirenkailla, joita pyöritetään hihna-käyttöisesti askelmootorilla. Kun askelmoottori on saavuttanut nopeuden 19 rpm (tällöin liukurengas pyörii nopeudella 5 rpm eli 50mm/s) tapahtuu RollScanin triggeraus ja mittaus alkaa. Tästä merkiksi ohjauspaneelin merkivalo H11 syttyy.

#### 1.11.7 Anturin liike pyöryksen aikana

Anturia kuljetetaan pitkin liukurenkaan pintaa. Liukurenkaan suoraa pintaa mitattaessa liike toteutetaan Z-akselilla, kaarevalla pinnalla käytetään Z ja Y-akseleita. Liukurenkaan pyöriessä anturia liu'utetaan mitattavan pinnan yläreunasta alareunaan. Yhdessä pyöryksen kanssa liike muodostaa spiraalin. Mittaus lopetetaan, kun anturi saavuttaa liukurenkaan mitattavan pinnan alareunan. Kun Z-akselin liike pysähtyy, pysäytetään pyörytys ja RollScanin triggeraus katkeaa. Samalla K4 kontaktori deaktivoituu.

#### 1.11.8 Anturin ajo kotiasemaan

Kun liukurenkaan pyörytys on pysähtynyt, ajetaan Z- ja Y-akselit kotiasemiinsa. Ajojärjestys riippuu siitä, kumpaa puolta liukurenkaasta on mitattu.

#### 1.11.9 Liukurenkaan demagnetointi

Kun anturi on ajettu kotiasemaansa, voidaan liukurengas demagneto ohjaamalla logiikan ulostulo QX8.4 pois päältä. Laskeva reuna laukaisee RollScanissa demagnetointirampin. Demagnetoinnin aikana liukurengasta ei pyöritetä.

#### 1.11.10 Magnetointirullien lasku

Mittaussekvenssi päättyy Z2-akselin saavuttaessa kotiaseman. Tällöin valopainike S2 sammuu ja uusi mittaussekvenssi voidaan aloittaa.

## 1.12 Kotiajosekvenssi

Mikäli mittaussekvenssi keskeytetään S3 painikkeella, siirtyy PLC-ohjelma mittaussekvenssistä kotiajosekvenssiin. Kotiajosekvenssissä on kaksi haaraa ja valittava haara määräytyy sen mukaan, kumpaa puolta liukurenkaasta on mitattu, kun sekvenssi on keskeytetty. Kotiajosekvenssin aktivoituessa valopainike S5, HOME, palaa jatkuvasti merkiksi siitä, että painamalla painiketta voidaan kotiajoaktivoida. Valopainike palaa koko kotiajosekvenssin ajan.

### 1.12.1 Demagnetointi

Jos mitattava liukurengas on magnetoitu, se demagnetoidaan kotiajosekvenssissä. Tämä päätellään logiikan ulostulon Q8.4 tilasta. Jos ulostulo on päällä, liukurengas demagnetoidaan ja jos ulostulo ei ole päällä, liukurengasta ei demagnetoida.

### 1.12.2 Akseleiden ajo kotiasemiinsa

Demagnetoinnin jälkeen akselit voidaan ajaa kotiasemiinsa. Ajo kotiasemiin tapahtuu tietystä järjestyksessä akseli kerrallaan. Tällä menettelyllä suojataan anturia rikkoutumiselta tilanteessa, jossa esim. anturi on paikoitettu mittamaan kuperan pinnan sisäreunaa, mutta kotiajon aktivoituessa Z2-akseli rikkoo anturin laskeutuessaan kotiasemaansa. Seuraavassa taulukossa on esitetty, missä järjestyksessä akselit ajetaan kotiasemiinsa.

**Taulukko 2 Akseleiden kotiutusjärjestys**

Akseli	Sekvenssi 1. Liukurenkaan tasainen pinta	Sekvenssi 2. Liukurenkaan kaareva pinta
Z-akseli	2.	1.
Y-akseli	1.	2.
Z2-akseli	3.	3.

Huomioitavaa sekvenssi 2:den kotiajossa on se, että ennen kuin Z-akseli voidaan ajaa kotiasemaansa, on Y-akseli ajettava ulos väliaseman (sama asema, johon akseli ajetaan anturin käännön päätteeksi). Näin varmistutaan siitä, ettei anturi vahingoitu laskettaessa Z-akseli kotiasemaansa. Kun kotiajosekvenssi on päättynyt, valopainike S5 sammuu ja uusi mittaussekvenssi voidaan aloittaa.

### 1.13 Lukitukset

Mittaussekvenssi ei käynnisty, jos:

- lukituskauluri on auki
- turvaovi on auki
- hätäpysäytys-piiri on aktiivinen
- käsiajo on aktiivinen
- servoissa häiriötä
- anturin käännön kytkimet ovat molemmat pois päältä
- anturin varren tilitin kytkimet pois päältä, tai vain toinen on päällä

### 1.14 Hälytykset

Laitteisto hälyttää, kun:

- hätäpysäytys-piiri on lauennut
- servo-ohjaimessa on häiriö
- ovipiiri on lauennut
- käsiajo aktiivinen
- lukituskauluri on auki
- kumpikaan anturin käännön sensoreista ei ole aktiivinen
- anturin tilitin on lauennut

- anturin kääntö epäonnistunut

Lukitusrenkaan tilaa ilmaistaan valomajakkan kirkkaalla valolla. Jos rengas on auki, tai sitä valvova sensori on epäkunnossa, vilkkuu valomajakkan kirkas valo.

#### 1.14.1 Servo Fault

Jos laitteiston merkkivalopainike **S4** syttyy, tarkoittaa se sitä, että yhdessä servo-ohjaimista tai askelmoottorin ohjaimessa on virhetilanne. Ennen virheen kuittamista tulee selvittää mikä virhe on kyseessä. Tämä voidaan tehdä kahdella tavalla:

- tulkitaan moottorinohjaimen näytöstä virhekoodi
- otetaan yhteys ohjaimen FCT-työkälulla, jolla voidaan selvittää, mistä virheestä on kysymys.

Moottorinohjaimen näytössä pyörivän mahdollisen virhekoodin selitykset löytyvät laitteiden manuaaleista.

Jos **S4** merkkivalo vilkkuu, tarkoittaa se sitä, että väylä on mennyt pois päältä virheiden suuren määrän takia.

#### 1.15 FCT – Festo Configuration Tool

FCT-ohjelmistolla hallinnoidaan askel- ja servomootoreiden moottorinohjaimia. Ohjelmaan on tallennettuna jokaisen laitteistossa käytettävän moottorin, moottorinohjaimen sekä liukujohteen tyypit ja ja tekniset ominaisuudet lukuunottamatta anturin kääntöön käytettävää moottoria ja tämän ohjainta.

Ohjelmistoa käytetään moottorien virittämiseen ja parametrintointiin. Mittaussekvenssissä käytettävät positiot opetetaan ohjaimille käyttäen FCT-ohjelmistoa. Tällöin moottoreita ajetaan käsiajolla FCT-ohjelmiston kautta käskytettynä. Halutut positiot tallennetaan taulukkoon, johon PLC-ohjelmassa viitataan. Taulukkoon voidaan lisäksi tallentaa jokaiselle askeleelle oma profiili, jossa eritellään mm.





Keijo Keränen

TOIMINTAKUVAUS

16 (16)

2013-07-19

nopeus, kiihtyvyys, hidastuvuus yms. Jokainen askel on kommentoitu, jotta kunkin askeleen toiminta selviää ohjelman käyttäjälle.

Jos sekvenssissä käytettäviä positioita on tarpeen jälkeinpäin korjata, voidaan korjaus tehdä FCT-ohjelmiston taulukkoon sen sijaan, että PLC:n ohjelmakoodiin kajotaan.

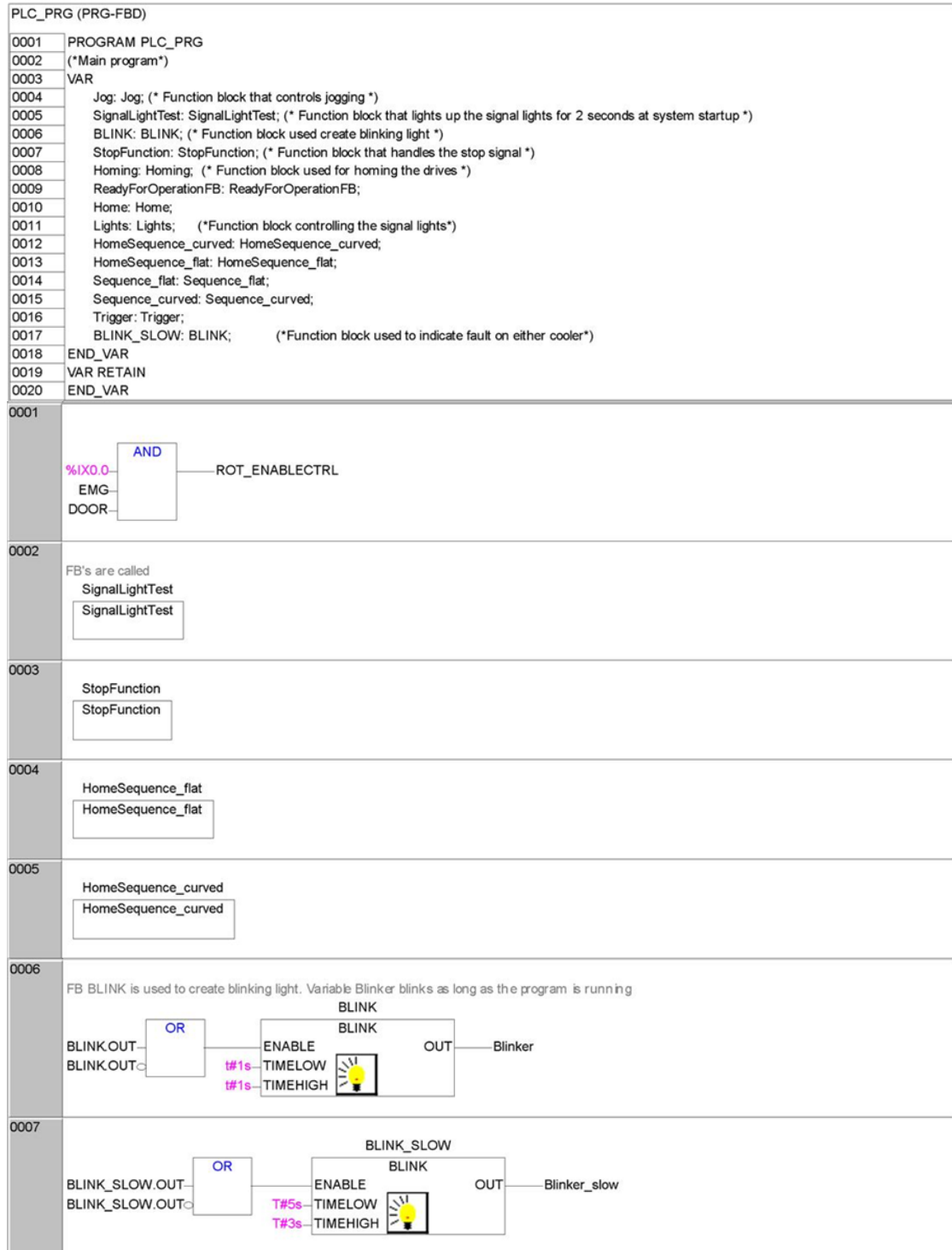
Yhteyden muodostamiseen tietokoneen ja moottorinohjaimen välille tarvitaan sarjakaapeli usb-sarjakaapelimuunnin. Sarjakaapeli kytketään moottorinohjaimen porttiin X5. Ohjelman tarkemmat ohjeet löytyvät ohjelman help-tiedostosta. Vain koulutettu henkilöstö saa tehdä muutoksia FCT-ohjelmistossa määriteltyihin asetuksiin.

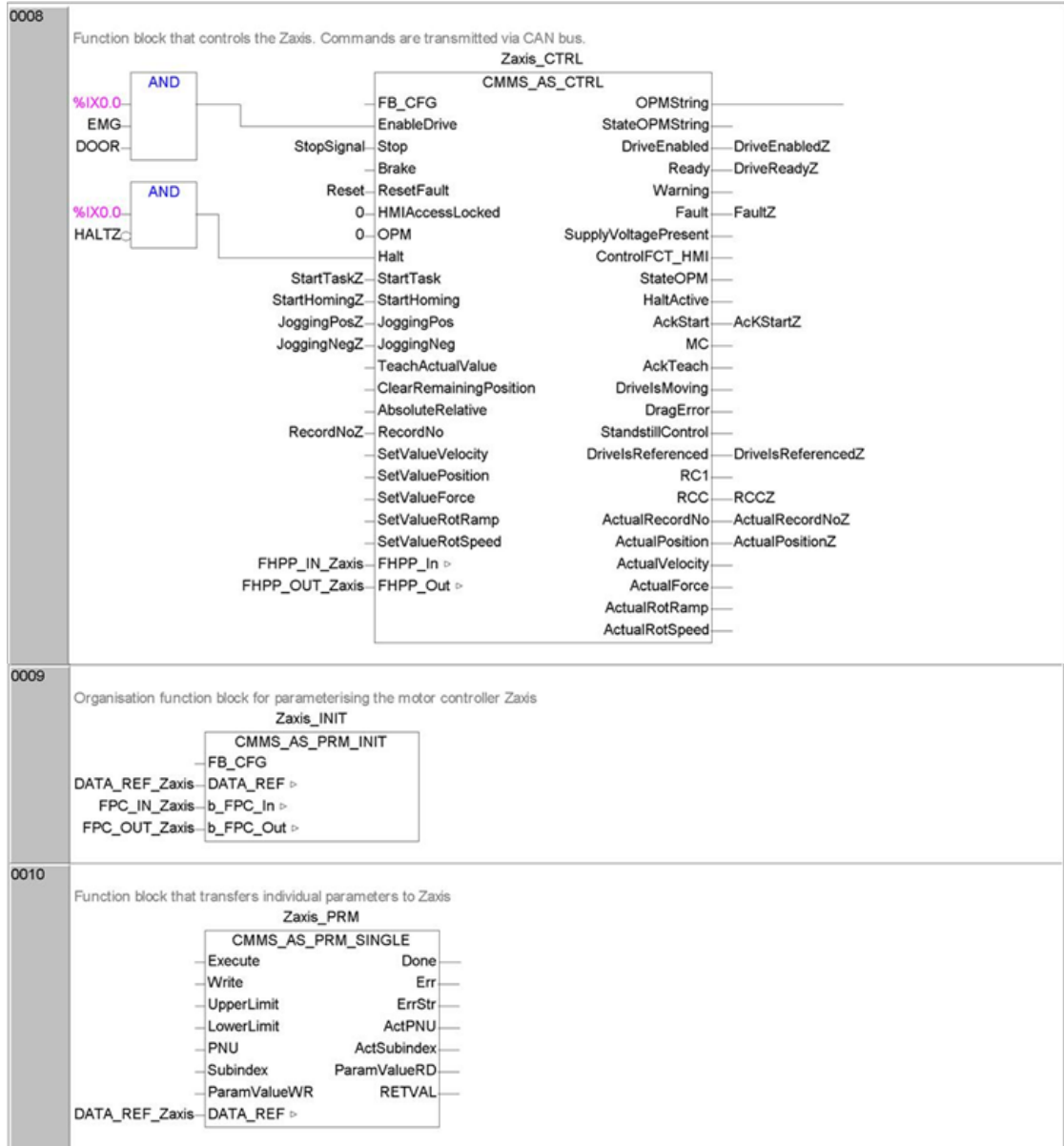
Jokaisessa moottorinohjaimessa, lukuunottamatta anturin kääntöön käytettävän moottorinohjainta, on SD-muistikortti, jolle on tallennettuna laitteen parametrit sekä mahdolliset ohjelmistopäivitykset. Moottorinohjaimia voidaan käyttää myös ilman muistikortteja, tällöin ohjaimet ilmoittavat virhetilanteesta käynnistyessään. Virhe kuittaantuu itsestään pois, joten laitteiston käytettävyyteen muistikortin puuttumisella ei ole merkitystä.

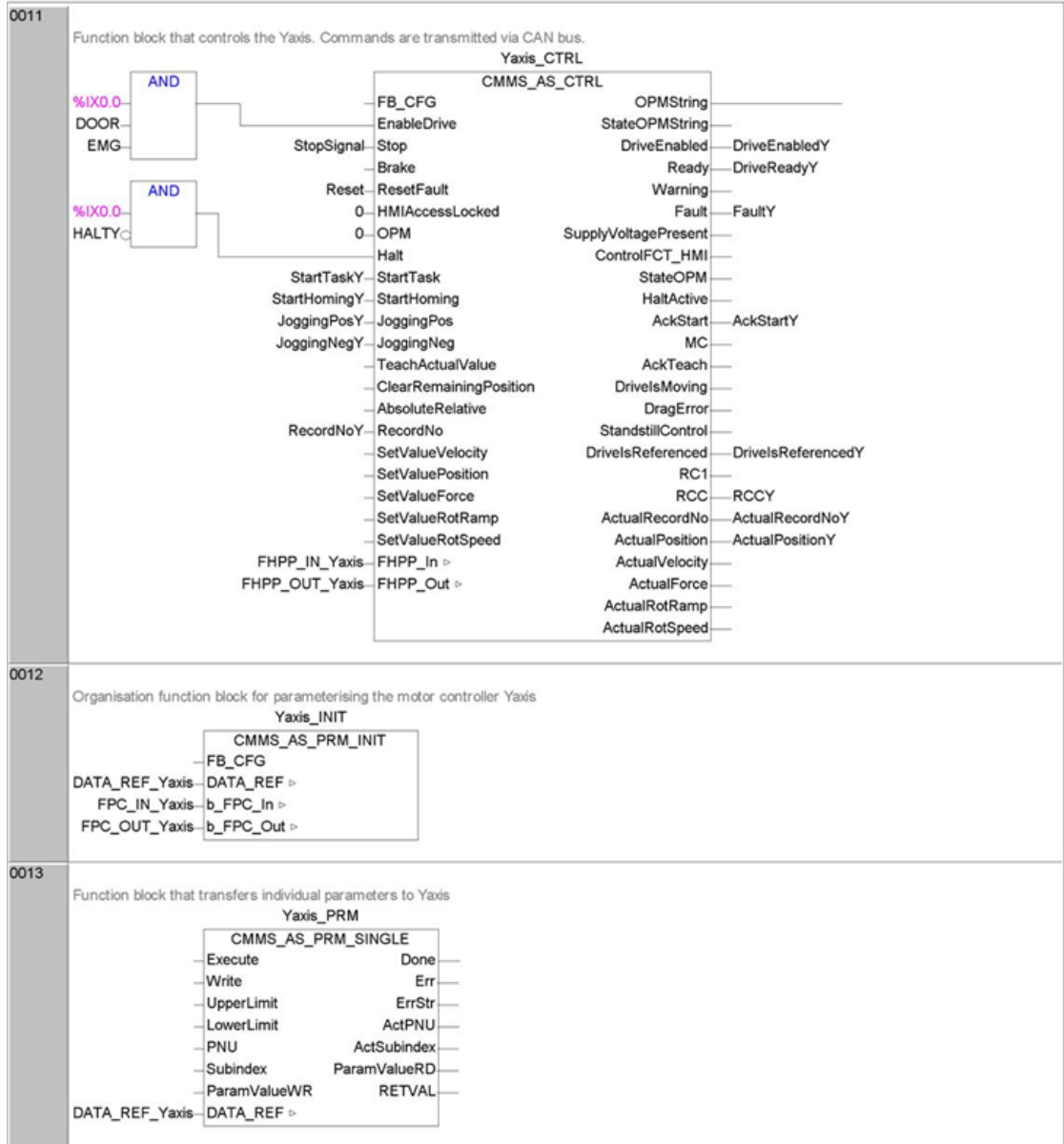
Käytettäessä moottorinohjainta ilman muistikorttia on DIP-kytkin 8 oltava OFF-asennossa. Kytkimen ollessa ON-asennossa ohjain etsii käynnistyessään muistikortilta uutta ohjelmistopäivitystä.

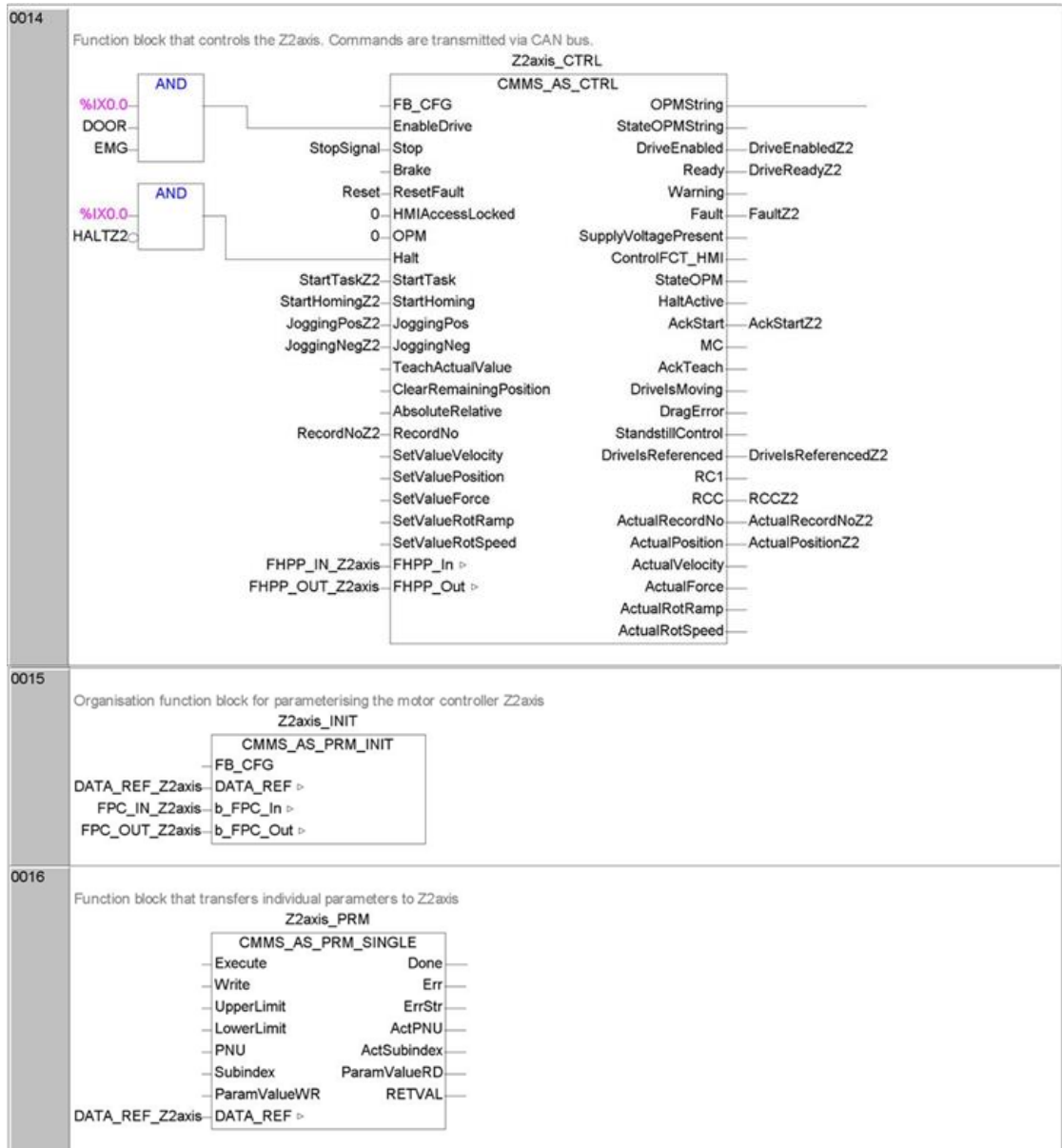
Muistikortille voidaan tallentaa moottorinohjaimessa olevat sen hetkiset asetukset ja parametrit FCT-työkalun avulla. FCT-työkalun asetuksista voidaan määritellä, tallennetaanko parametrit muistikortille vai ladataanko parametreja muistikortilta. Muistikortit toimivat varmuuskopiona parametreille, mikäli tiedot jostain syystä häviävät ohjaimien muistista.

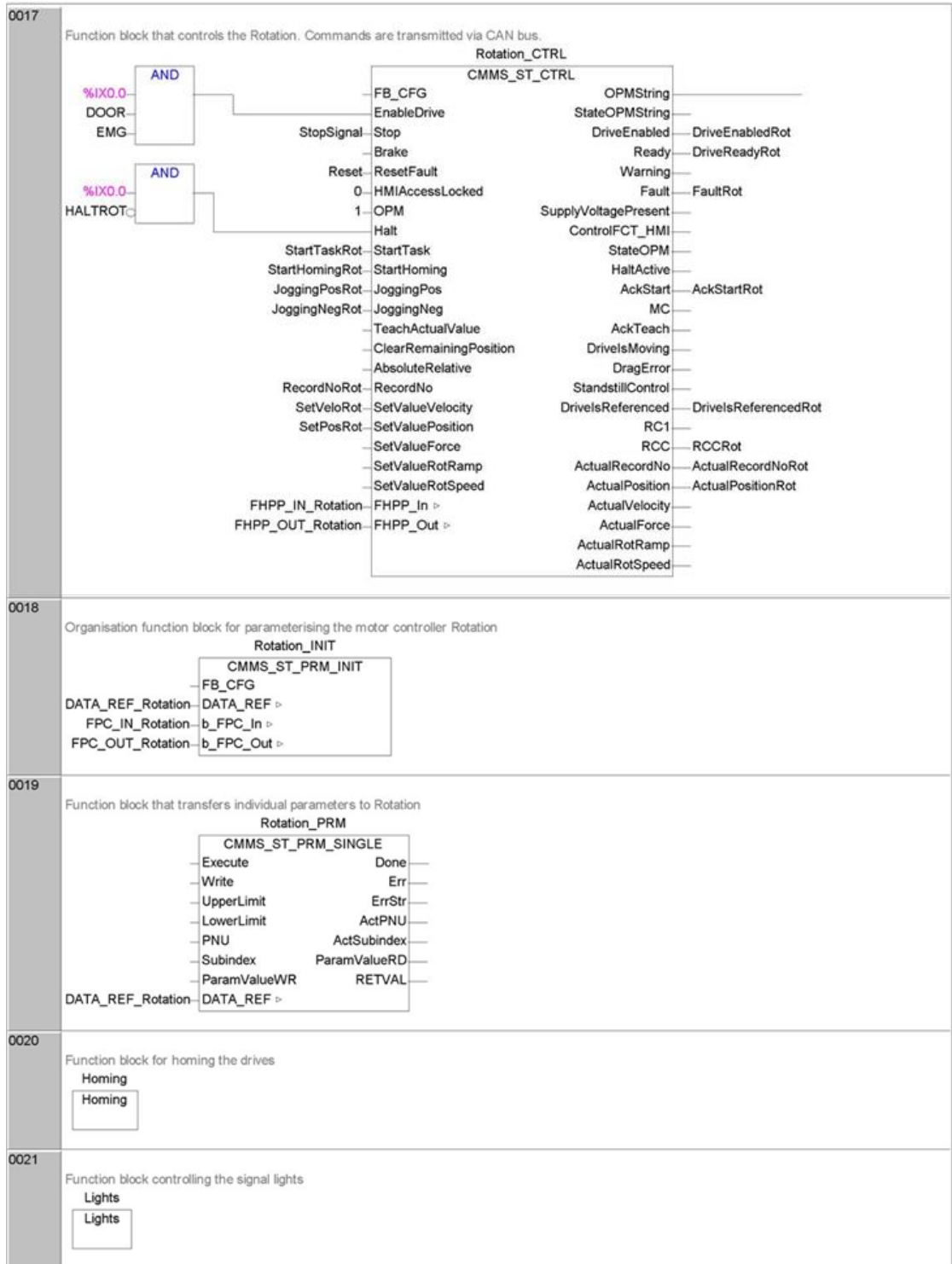
## Liite 5. PLC\_PRG











0022	Function block controlling the motions. Flat side of the washer ring Sequence_flat Sequence_flat
0023	Function block controlling the motions. Curved side of the washer ring Sequence_curved Sequence_curved
0024	FB for manually moving the drives Jog Jog
0025	FB that checks if the drives are at home position. Home Home
0026	FB that checks if drives are ready for operation ReadyForOperationFB ReadyForOperationFB
0027	Trigger Trigger

## Liite 6. HOME

Home (FB-ST)	
0001	FUNCTION_BLOCK Home
0002	VAR_INPUT
0003	END_VAR
0004	VAR_OUTPUT
0005	END_VAR
0006	VAR
0007	END_VAR
0001	HomeBit:= (DrivesReferencedRot
0002	AND DrivesReferencedY (*Homing is done and ok*)
0003	AND DrivesReferencedZ
0004	AND DrivesReferencedZ2)
0005	AND (ROT_MOTIONCOMPL (*Drives aren't moving*)
0006	AND Y_MOTIONCOMPL
0007	AND Z_MOTIONCOMPL
0008	AND Z2_MOTIONCOMPL)
0009	AND (ActualRecordNoY = 1 (*Drives are moved to record 1 on record table after homing*)
0010	AND ActualRecordNoZ = 1
0011	AND ActualRecordNoZ2 = 1)
0012	AND NOT JOG_activated
0013	AND (ActualPositionY = 0 (*Double check that drives are where they should after homing or homesequence*)
0014	AND ActualPositionZ = 0
0015	AND ActualPositionZ2 = 0);

## Liite 7. LIGHTS

```

Lights (FB-ST)
0001 FUNCTION_BLOCK Lights
0002 VAR_INPUT
0003 END_VAR
0004 VAR_OUTPUT
0005 END_VAR
0006 VAR
0007 END_VAR
-----
0001 (*Variable SignalLights is used to test if the signal lights work ok*)
0002 (* The lights are lit at startup and remain lit for 2 seconds*)
0003 (*IF BUSError or SystemFault is TRUE, then all the lights in control panel and light pillar are blinking*)
0004
0005 (*-----*)
0006
0007 JOG_ACT:=(Z_JOG
0008 OR Y_JOG
0009 OR Z2_JOG
0010 OR ROT_JOG      (*If jog mode selection switch is active then signal light H9 is lit*)
0011 OR DC_JOG)
0012 AND NOT Measuring)
0013 OR SignalLights; (*Lamptest*)
0014
0015 (*-----*)
0016
0017 FAULT:=(BUSError
0018 OR SystemFault (*If BUSError or SystemFault is TRUE, S4 is blinking*)
0019 OR SFCErrror)
0020
0021 OR (ServoFault()) (*If function ServoFault is true then signal light S4 is flashing*)
0022 AND Blinker
0023 AND E1_ERROR
0024 AND E2_ERROR
0025 AND NOT BUSError
0026 AND NOT SystemFault
0027 AND NOT SFCErrror)
0028
0029 OR SignalLights
0030
0031 OR ((NOT E1_ERROR (*If errormessage from coolers is missing, the FAULT-light flashes slowly*)
0032 OR NOT E2_ERROR) (*One can operate without the coolers running*))
0033 AND NOT ServoFault()
0034 AND NOT BUSError
0035 AND NOT SystemFault
0036 AND NOT SFCErrror
0037 AND Blinker_slow);
0038
0039 (*-----*)
0040
0041 HOME_LIGHT:=(
0042 (NOT DrivelsReferencedRot (*If the drives are not referenced the signal light S5 blinks.*/)
0043 OR NOT DrivelsReferencedY
0044 OR NOT DrivelsReferencedZ
0045 OR NOT DrivelsReferencedZ2)
0046 AND NOT SFCErrror
0047 AND NOT BUSError
0048 AND NOT SystemFault
0049 AND Blinker
0050 AND EMG      (*Safety door and emg must be ok*)
0051 AND DOOR)
0052 AND (NOT Y_JOG
0053 AND NOT Z2_JOG
0054 AND NOT Z_JOG      (*jog mode selector switch must off or jog mode activation key must 0*)
0055 AND NOT ROT_JOG
0056 AND NOT DC_JOG)
0057
0058 OR ((SEQUENCEINTERRUPTED (*If the sequence is interrupted the signal light S5 is lit.*/)
0059 OR SEQUENCE2INTERRUPTED)
0060 AND EMG
0061 AND DOOR      (*Safety door and emg must be ok*)
0062 AND NOT SFCErrror
0063 AND NOT BUSError
0064 AND NOT SystemFault
0065 AND (NOT Y_JOG
0066 AND NOT Z2_JOG
0067 AND NOT Z_JOG      (*jog mode selector switch must off or jog mode activation key must 0*)

```



```

0068 AND NOT ROT_JOG
0069 AND NOT DC_JOG))
0070
0071 OR (JOG_activated
0072 AND DrivesReferencedY
0073 AND DrivesReferencedZ      (*If jogmode is activated (and the jogmode selector switch is turned off)*)
0074 AND DrivesReferencedZ2    (*and the drives are referenced then the signal light S5 is lit*)
0075 AND DrivesReferencedRot
0076 AND NOT measuring
0077 AND EMG
0078 AND DOOR
0079 AND (NOT Y_JOG
0080 AND NOT Z2_JOG
0081 AND NOT Z_JOG      (*jog mode selector switch must off or jog mode activation key must 0*)
0082 AND NOT ROT_JOG
0083 AND NOT DC_JOG))
0084 AND (NOT SFCErr
0085 AND NOT BUSErr
0086 AND NOT SystemFault)
0087
0088 OR SignalLights;
0089
0090 (*-----*)
0091
0092 EMG_ACK:=NOT EMG      (*If emergency stop is active then signal light S13 is lit*)
0093 OR SignalLights;
0094
0095 (*-----*)
0096
0097 DOOR_ACK:=NOT DOOR   (*If safety door is open then signal light S6 is lit*)
0098 OR SignalLights;
0099
0100 (*-----*)
0101
0102 RSMEASURING:=RSTRIG; (*When RollScan is measuring the washer ring then signal light H11 is lit*)
0103
0104
0105 (*-----*)
0106
0107 ServosEnabled:=(DriveEnabledRot
0108 AND DriveEnabledY
0109 AND DriveEnabledZ
0110 AND DriveEnabledZ2) (*If the drives are enabled the signal light is lit*)
0111 AND (DriveReadyRot
0112 AND DriveReadyY      (*In addition drives need to be ready*)
0113 AND DriveReadyZ
0114 AND DriveReadyZ2)
0115 OR SignalLights;
0116
0117 (*-----*)
0118
0119 START_LIGHT:=Measuring; (*Start light S2 is lit when measuring.*)
0120
0121 (*-----*)
0122
0123 H1RD:=NOT EMG      (*If emg button is activated, the red light on signal pillar is lit*)
0124 OR SignalLights;
0125
0126 (*-----*)
0127
0128 H1GN:=Measuring      (*If the machine is measuring then green light in the signal pillar is lit*)
0129 OR SignalLights;
0130
0131 (*-----*)
0132
0133 H1CLR:=NOT DOOR
0134 OR SignalLights (*If the safety door is open or the safety relay monitoring the safety*)
0135 OR (NOT B11      (*door isn't acknowledged, the clear light in signal pillar is lit*)
0136 AND Blinker);

```

## Liite 8. READYFOROPERATION

```

ReadyForOperationFB (FB-ST)
0001 FUNCTION_BLOCK ReadyForOperationFB
0002 VAR_INPUT
0003 END_VAR
0004 VAR_OUTPUT
0005 END_VAR
0006 VAR
0007 END_VAR
-----
0001 ReadyForOperation:= (Z_READY      (*System is ready for operation when*)
0002 AND Y_READY
0003 AND Z2_READY    (*Drives are in ready state*)
0004 AND ROT_READY)
0005 AND StopSignal
0006 AND (NOT Z_JOG
0007 AND NOT Y_JOG    (*Jog mode isn't active*)
0008 AND NOT Z2_JOG
0009 AND NOT ROT_JOG
0010 AND NOT DC_JOG)
0011 AND HomeBit      (*Homebit is TRUE --> see HOME(FB)*)
0012 AND NOT Measuring (*Measuring sequence isn't active*)
0013 AND (NOT SEQUENCEINTERRUPTED
0014 AND NOT SEQUENCE2INTERRUPTED) (*Homesequence isn't active*)
0015 AND (NOT HOMESEQUENCE1
0016 AND NOT HOMESEQUENCE2);

```

## Liite 9. SIGNALLIGHTTEST

```

SignalLightTest (FB-ST)
0001 FUNCTION_BLOCK SignalLightTest
0002 VAR
0003     TOFInst: TOF;
0004 END_VAR
-----
0001 SignalLightTest.TOFInst(IN:=TestSignalLights , PT:=t#2000ms); (*turn off delay*)
0002 SignalLights:=SignalLightTest.TOFInst.Q;
0003
0004 IF SignalLights
0005 THEN TestSignalLights:=FALSE;  (*Variable TestSignalLights is set to FALSE 2 seconds after startup*)
0006 END_IF;                        (*TestSignalLights is set to TRUE on system start with INIT(FUN)*)
0007
0008

```

## Liite 10. STOPFUNCTION

StopFunction (FB-ST)	
0001	FUNCTION_BLOCK StopFunction
0002	VAR
0003	END_VAR
0004	(*Machine can perform movements when stop signal is TRUE*)
0001	StopSignal:=
0002	EMG (*StopSignal transmitted via CAN bus*)
0003	AND DOOR
0004	AND STOP_BUTTON
0005	AND B11 (*Locking ring must be shut*)
0006	AND (Z_ERROR (*No errors on controllers. When Z_ERROR = TRUE, then controller is ok*))
0007	AND Z2_ERROR
0008	AND Y_ERROR
0009	AND ROT_ERROR
0010	AND ((B9 (*If the BN sensor is tilted too much on either direction, the stopsignal is disabled*))
0011	AND B10)
0012	OR Y_JOG (*if the BN sensor is tilted too much it can be moved to opposite direction with jog mode*)
0013	(*AND NOT E1_ERROR (*Coolers are ok *) (*If you want the machine to operate without the coolers running,
0014	AND NOT E2_ERROR (*When coolers are running, the ERROR-variables must be TRUE*))
0015	AND FUSES_OK (*Fuses ok*)
0016	*)
0017	AND NOT BUSErr (*no Bus or System errors*)
0018	AND NOT SystemFault
0019	AND NOT SFCErr;
0020	
0021	STOP_ROT:=StopSignal;
0022	STOP_Y:=StopSignal; (*STOP_(name of the axis) is transmitted via controllers IO interface *)
0023	STOP_Z:=StopSignal;
0024	STOP_Z2:=StopSignal;

## Liite 11. TRIGGER

Trigger (FB-ST)	
0001	FUNCTION_BLOCK Trigger
0002	VAR_INPUT
0003	END_VAR
0004	VAR_OUTPUT
0005	END_VAR
0006	VAR
0007	TONInst_RSTRIG:TON;
0008	END_VAR
0001	K4:=(ROT_JOG AND MANUAL_TRIG) OR Activate_K4; (*These are for testing. RollScan can be triggered, when rotation is jogge
0002	Trigger.TONInst_RSTRIG(IN:=MANUAL_TRIG AND ROT_JOG, PT:=T#1000ms); (*K4 is activated first and after 1s trigger to Rol
0003	
0004	RSTRIG:=(Trigger.TONInst_RSTRIG.Q OR Activate_RSTRIG) AND K4_OK; (* This FB controls the trigger signal*)
0005	

## Liite 12. JOG

```

Jog (FB-ST)
0001 FUNCTION_BLOCK Jog
0002 VAR_INPUT
0003 END_VAR
0004 VAR_OUTPUT
0005 END_VAR
0006 VAR
0007
0008     RSInst1: RS;
0009     POSDIR: BOOL;
0010     RSInst2: RS;
0011     NEGDIR: BOOL;
0012     R_TRIG1: R_TRIG;
0013     R_TRIG2: R_TRIG;
0014     SRInst: SR;
0015     TONInst_neg: TON;
0016     TONInst_pos: TON;
0017     TONInst_RSTTRIG: TON;
0018
0019 END_VAR
0001 (*This function block is used to move the axes manually. The axes are moved one at the time*)
0002
0003 (*-----*)
0004
0005 JoggingPosZ:=Z_JOG      (*Manually moving the Zaxis*)
0006 AND JOG_POSITIVE
0007 AND NOT JOG_NEGATIVE  (*If S7 and S8 are activated simultaneously the drive won't move*)
0008 AND NOT Measuring    (*Jogging is not possible when measuring sequence active*)
0009 AND NOT Z2axis_CTRL.DrivesMoving
0010 AND NOT Yaxis_CTRL.DrivesMoving
0011 AND NOT Rotation_CTRL.DrivesMoving
0012 AND StopSignal;
0013
0014 (*-----*)
0015
0016 JoggingNegZ:=Z_JOG      (*Manually moving the Zaxis*)
0017 AND JOG_NEGATIVE
0018 AND NOT JOG_POSITIVE
0019 AND NOT Measuring
0020 AND NOT Z2axis_CTRL.DrivesMoving
0021 AND NOT Yaxis_CTRL.DrivesMoving
0022 AND NOT Rotation_CTRL.DrivesMoving
0023 AND StopSignal;
0024
0025 (*-----*)
0026
0027 JoggingPosZ2:=Z2_JOG   (*Manually moving the Z2axis*)
0028 AND JOG_POSITIVE
0029 AND NOT JOG_NEGATIVE
0030 AND NOT Measuring
0031 AND NOT Zaxis_CTRL.DrivesMoving
0032 AND NOT Yaxis_CTRL.DrivesMoving
0033 AND NOT Rotation_CTRL.DrivesMoving
0034 AND StopSignal;
0035
0036 (*-----*)
0037
0038 JoggingNegZ2:=Z2_JOG   (*Manually moving the Z2axis*)
0039 AND JOG_NEGATIVE
0040 AND NOT JOG_POSITIVE
0041 AND NOT Measuring
0042 AND NOT Zaxis_CTRL.DrivesMoving
0043 AND NOT Yaxis_CTRL.DrivesMoving
0044 AND NOT Rotation_CTRL.DrivesMoving
0045 AND StopSignal;
0046
0047 (*-----*)
0048
0049 JoggingPosY:=Y_JOG      (*Manually moving the Yaxis*)
0050 AND JOG_POSITIVE
0051 AND NOT JOG_NEGATIVE
0052 AND NOT Measuring
0053 AND B9
0054 AND NOT Zaxis_CTRL.DrivesMoving
0055 AND NOT Z2axis_CTRL.DrivesMoving
0056 AND NOT Rotation_CTRL.DrivesMoving
0057 AND StopSignal;
0058
0059 (*-----*)

```

```

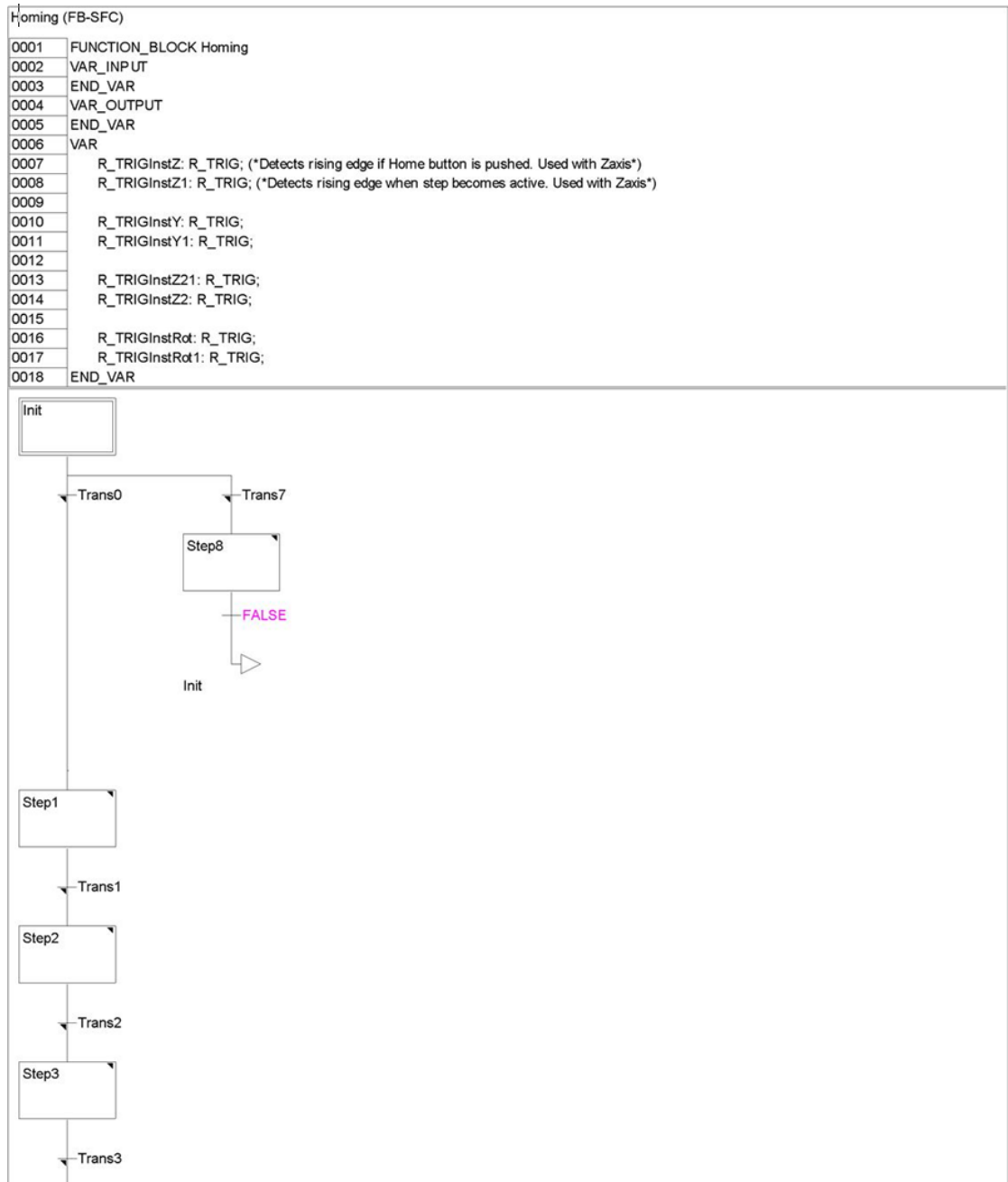
0060
0061 JoggingNegY:=Y_JOG          (*Manually moving the Yaxis*)
0062 AND JOG_NEGATIVE
0063 AND NOT JOG_POSITIVE
0064 AND NOT Measuring
0065 AND B10
0066 AND NOT Zaxis_CTRL.DrivesMoving
0067 AND NOT Z2axis_CTRL.DrivesMoving
0068 AND NOT Rotation_CTRL.DrivesMoving
0069 AND StopSignal;
0070
0072
0073 JoggingPosRot:=ROT_JOG      (*Manually moving the Rotation*)
0074 AND JOG_POSITIVE
0075 AND NOT JOG_NEGATIVE
0076 AND NOT Measuring
0077 AND NOT Zaxis_CTRL.DrivesMoving
0078 AND NOT Z2axis_CTRL.DrivesMoving
0079 AND NOT Yaxis_CTRL.DrivesMoving
0080 AND StopSignal;
0081
0082 (*-----*)
0083
0084 JoggingNegRot:=ROT_JOG      (*Manually moving the Rotation*)
0085 AND JOG_NEGATIVE
0086 AND NOT JOG_POSITIVE
0087 AND NOT Measuring
0088 AND NOT Zaxis_CTRL.DrivesMoving
0089 AND NOT Z2axis_CTRL.DrivesMoving
0090 AND NOT Yaxis_CTRL.DrivesMoving
0091 AND StopSignal;
0092
0093 (*-----*)
0094
0095 Jog_R_TRIG1(CLK:=JOG_POSITIVE); (*FB to detect rising edge when S7 is activated*)
0096 Jog_R_TRIG2(CLK:=JOG_NEGATIVE); (*FB to detect rising edge when S8 is activated*)
0097
0098 (*-----*)
0099
0100 Jog.TONInst_neg(In:=DC_END0, PT:=T#500ms); (*turn on delay 500ms*)
0101 Jog.TONInst_pos(In:=DC_END1, PT:=T#500ms);
0102
0103 Jog.RSInst1(SET:=Jog_R_TRIG1.Q(*FB used to rotate sensor clockwise*)
0104 AND NOT JOG_NEGATIVE
0105 AND StopSignal,
0106 RESET1:=NOT JOG_POSITIVE
0107 OR NOT StopSignal
0108 OR Jog.TONInst_pos.Q);
0109 POSDIR:=Jog.RSInst1.Q1;
0110
0111 (*-----*)
0112
0113 Jog.RSInst2(SET:=Jog_R_TRIG2.Q(*FB used to rotate sensor counterclockwise*)
0114 AND NOT JOG_POSITIVE
0115 AND StopSignal,
0116 RESET1:=NOT JOG_NEGATIVE
0117 OR NOT StopSignal
0118 OR Jog.TONInst_neg.Q);
0119 NEGDIR:=Jog.RSInst2.Q1;
0120
0121 (*-----*)
0122
0123 SPD2:=(DC_JOG          (*Controlling the output that's connected DC motor controller*)
0124 AND POSDIR           (*Rotates the sensor clockwise*)
0125 AND NOT Measuring (*Together with RWD rotates sensor counterclockwise*)
0126 AND StopSignal
0127 OR (DC_RWD
0128 OR DC_FWD))
0129 OR RWD;
0130
0131 (*-----*)
0132
0133 RWD:=(DC_JOG          (*Controlling the output that's connected DC motor controller*)
0134 AND NEGDIR           (*Together with SPD2 rotates the sensor counterclockwise*)
0135 AND NOT JOG_POSITIVE
0136 AND NOT Measuring
0137 AND StopSignal)
0138 OR DC_RWD;
0139
0140 (*-----*)

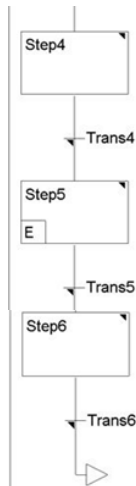
```

```

0141
0142 Jog.SRInst(SET1:=(DC_JOG (*FB that controls the Jog_activated variable*)
0143 OR ROT_JOG (*In order to make movements, the variable must be FALSE*)
0144 OR Y_JOG
0145 OR Z_JOG
0146 OR Z2_JOG),
0147 RESET:=HomeBit OR Reset_JogActivated);
0148 JOG_activated:=Jog.SRInst.Q1;
0149 (*Jog_activated variable is used to control Home light and in home sequences and in homing *)
0150
0151 (*-----*)
    
```

### Liite 13. HOMING





Init

Homing (FB-SFC).Transition Trans0 (ST)

```

0001 (TML1_SEL
0002 OR REF1_SEL
0003 OR TML2_SEL
0004 OR REF2_SEL)
0005 AND HOME_BUTTON
0006 AND (NOT DrivelsReferencedRot
0007 OR NOT DrivelsReferencedY
0008 OR NOT DrivelsReferencedZ
0009 OR NOT DrivelsReferencedZ2)
0010 AND (DriveEnabledRot
0011 AND DriveEnabledY
0012 AND DriveEnabledZ
0013 AND DriveEnabledZ2)
0014 AND StopSignal
0015 AND (DC_END0
0016 OR DC_END1)
0017 AND (NOT DC_JOG
0018 AND NOT ROT_JOG
0019 AND NOT Z_JOG
0020 AND NOT Z2_JOG
0021 AND NOT Y_JOG)

```

Homing (FB-SFC).Transition Trans7 (ST)

```

0001 (TML1_SEL
0002 OR REF1_SEL
0003 OR TML2_SEL
0004 OR REF2_SEL)
0005 AND HOME_BUTTON
0006 AND (NOT DrivelsReferencedRot
0007 OR NOT DrivelsReferencedY
0008 OR NOT DrivelsReferencedZ
0009 OR NOT DrivelsReferencedZ2)
0010 AND (DriveEnabledRot
0011 AND DriveEnabledY
0012 AND DriveEnabledZ
0013 AND DriveEnabledZ2)
0014 AND StopSignal
0015 AND (NOT DC_END0
0016 AND NOT DC_END1)
0017 AND (NOT DC_JOG
0018 AND NOT ROT_JOG
0019 AND NOT Z_JOG
0020 AND NOT Z2_JOG
0021 AND NOT Y_JOG)

```

Homing (FB-SFC).Action Step8 (ST)

```

0001 SFCErr:=TRUE;
0002

```

Homing (FB-SFC).Action Step1 (ST)

```

0001 Homing_R_TRIGInstZ(CLK:=HOME_BUTTON);
0002 Homing_R_TRIGInstZ1(CLK:=STEP1);
0003 StartHomingZ:=Homing_R_TRIGInstZ.Q OR Homing_R_TRIGInstZ1.Q;
0004 Reset_JogActivated:=TRUE;

```

Homing (FB-SFC).Transition Trans1 (ST)

```

0001 DrivelsReferencedZ
0002 AND ActualPositionZ = 0
0003 AND StopSignal

```

```

Homing (FB-SFC).Action Step2 (ST)
0001 Homing_R_TRIGInstY(CLK:=HOME_BUTTON);
0002 Homing_R_TRIGInstY1(CLK:=STEP2);
0003 StartHomingY:=Homing_R_TRIGInstY.Q OR Homing_R_TRIGInstY1.Q;

Homing (FB-SFC).Transition Trans2 (ST)
0001 DrivelsReferencedY
0002 AND ActualPositionY = 0
0003 AND StopSignal

Homing (FB-SFC).Action Step3 (ST)
0001 Homing_R_TRIGInstZ2(CLK:=HOME_BUTTON);
0002 Homing_R_TRIGInstZ21(CLK:=STEP3);
0003 StartHomingZ2:=Homing_R_TRIGInstZ2.Q OR Homing_R_TRIGInstZ21.Q;

Homing (FB-SFC).Transition Trans3 (ST)
0001 DrivelsReferencedZ2
0002 AND ActualPositionZ2 = 0
0003 AND StopSignal

Homing (FB-SFC).Action Step4 (ST)
0001 Homing_R_TRIGInstRot(CLK:=HOME_BUTTON);
0002 Homing_R_TRIGInstRot1(CLK:=STEP4);
0003 StartHomingRot:=Homing_R_TRIGInstRot.Q OR Homing_R_TRIGInstRot1.Q;

Homing (FB-SFC).Transition Trans4 (ST)
0001 DrivelsReferencedRot
0002 AND StopSignal

Homing (FB-SFC).Action Step5 - Entry (ST)
0001 RecordNoY:=1;
0002 RecordNoZ:=1;
0003 RecordNoZ2:=1;

Homing (FB-SFC).Action Step5 (ST)
0001 StartTaskY:=TRUE;
0002 StartTaskZ:=TRUE;
0003 StartTaskZ2:=TRUE;

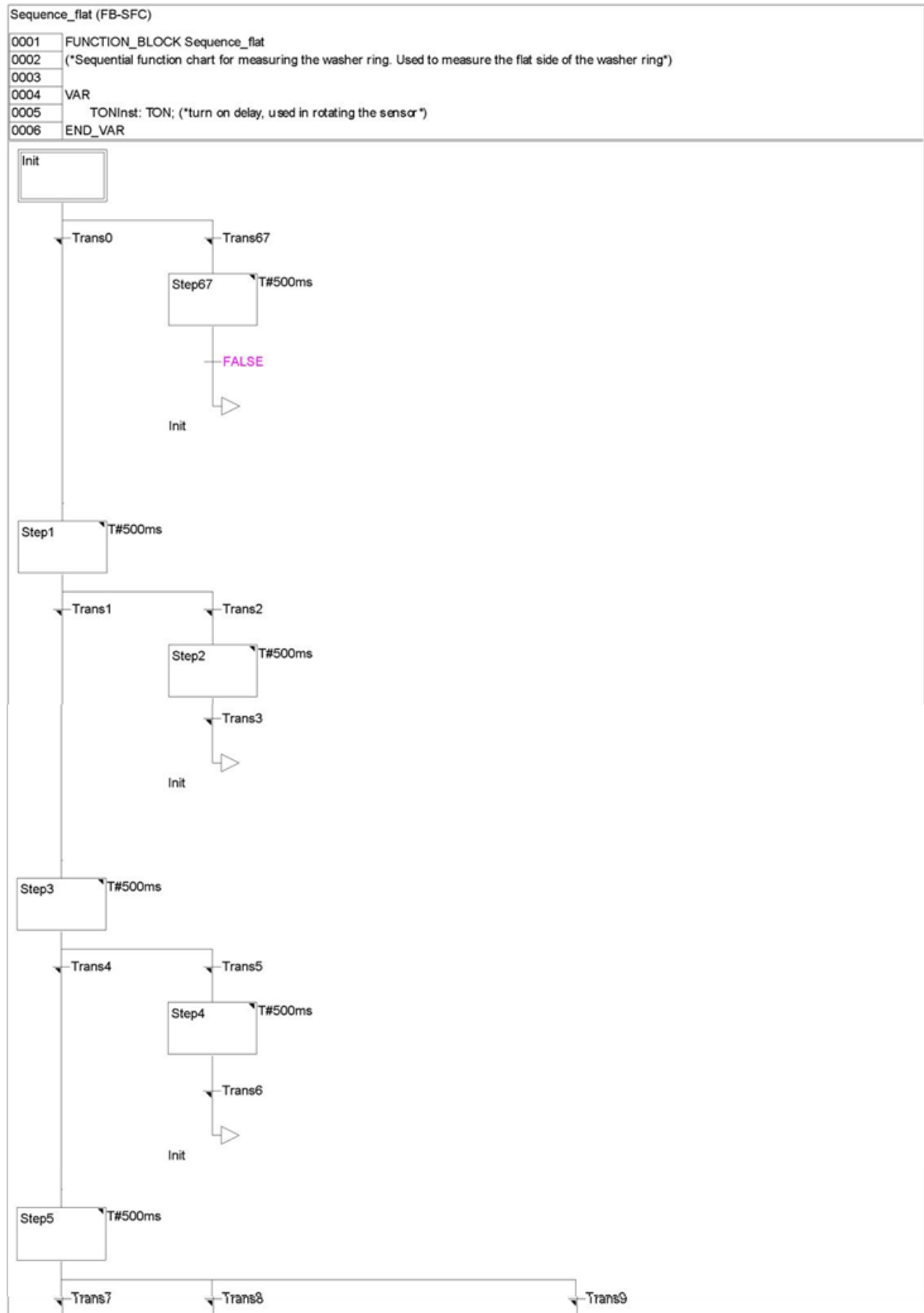
Homing (FB-SFC).Transition Trans5 (ST)
0001 ActualRecordNoY = 1
0002 AND ActualRecordNoZ = 1
0003 AND ActualRecordNoZ2 = 1

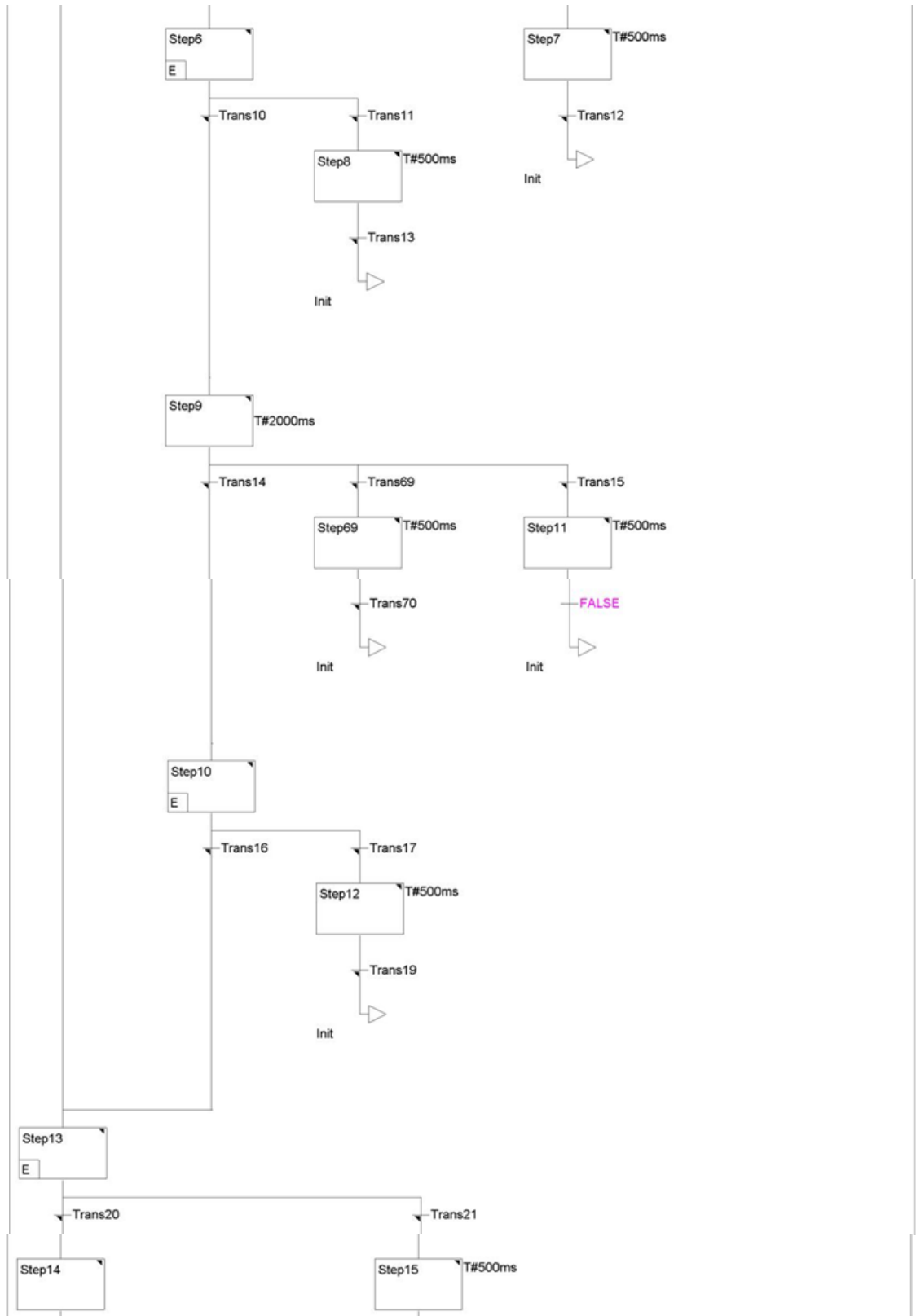
Homing (FB-SFC).Action Step6 (ST)
0001 StartTaskY:=FALSE;
0002 StartTaskZ:=FALSE;
0003 StartTaskZ2:=FALSE;
0004
0005 StartHomingRot:=FALSE;
0006 StartHomingY:=FALSE;
0007 StartHomingZ:=FALSE;
0008 StartHomingZ2:=FALSE;

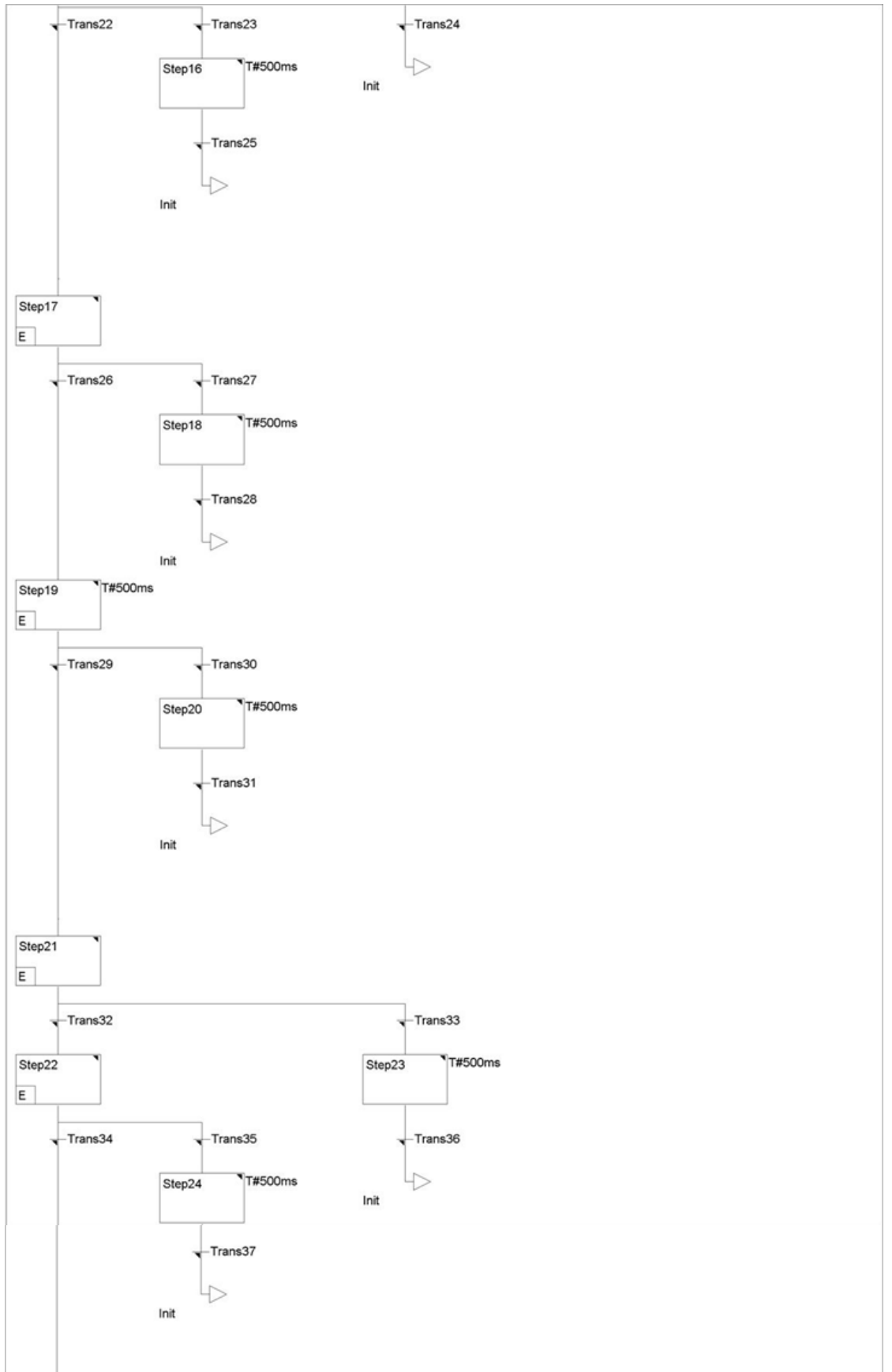
```

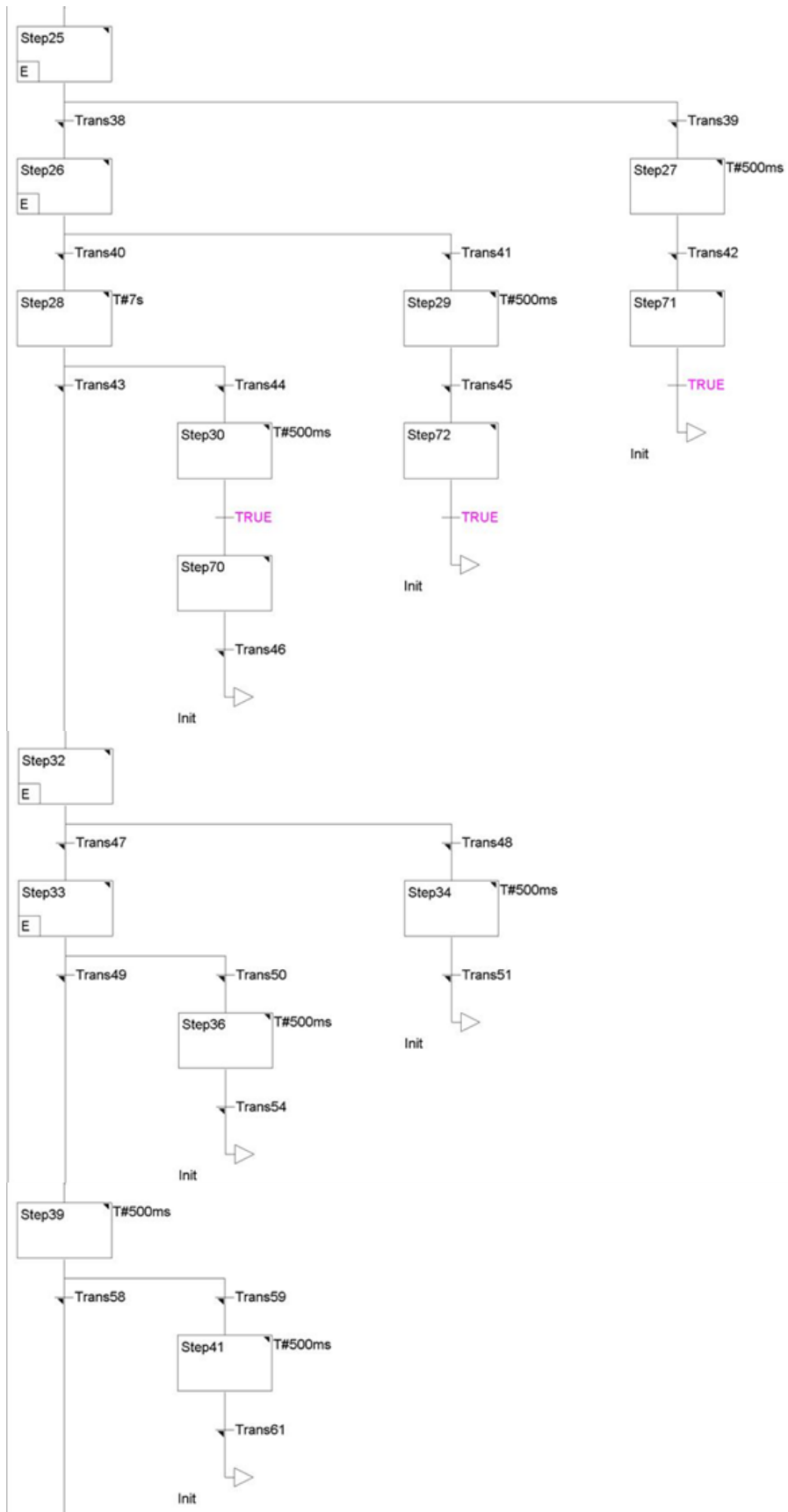


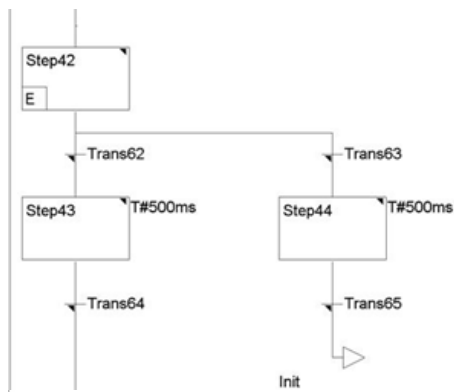
## Liite 14. SEQUENCE FLAT











Sequence\_flat (FB-SFC).Transition Trans0 (ST)

```

0001 (*Between the steps there are so-called transitions.
0002
0003 A transition condition must have the value TRUE OR FALSE. Thus it can consist OF either a boolean variable,
0004 a boolean address OR a boolean CONSTANT. It can also contain a series OF instructions having a boolean result,
0005 either in ST syntax (e.g. (i<= 100) AND b) OR in any language desired.
0006 But a transition may NOT contain programs, FUNCTION blocks OR assignments!*)
0007
0008 ReadyForOperation
0009 AND (TML1_SEL
0010 OR REF1_SEL)
0011 AND START_BUTTON
0012 AND (DC_END0
0013 OR DC_END1)
0014 (*Sequence starts if machine is ready for operation (drives enabled etc) and start button is activated*)
0015
0016

```

Sequence\_flat (FB-SFC).Transition Trans67 (ST)

```

0001 ReadyForOperation
0002 AND (TML1_SEL
0003 OR REF1_SEL)
0004 AND START_BUTTON
0005 AND (NOT DC_END0
0006 AND NOT DC_END1)
0007
0008 (*if neither one of the switches monitoring the sensor is active,*)
0009 (*the system is set TO "error state" in following step*)

```

Sequence\_flat (FB-SFC).Action Step67 (ST)

```

0001 SFCError:=TRUE;
0002 Measuring:=FALSE;
0003
0004
0005 (*If SFCError is set to TRUE, the system cannot operate. System has to be reset and then restarted*)
0006 (*This way the sensor is kept safe from collision*)
0007 (*If SFCError activates, there's something wrong with*)
0008 (*switches monitoring the sensor rotation or with the motor rotating the sensor*)
0009 (*Maintenance is required*)

```

Sequence\_flat (FB-SFC).Action Step1 (ST)

```

0001 (*With template selector switch the plc sends information to viewscan via National Instruments IO-module*)
0002 (*The used template is selected with 2 bits*)
0003
0004
0005 IF REF1_SEL THEN %QX8.3:=TRUE;
0006 ELSE %QX8.3:=FALSE; (*Proper template is selected with these outputs.*)
0007 END_IF (*Information is transmitted to ViewScan with the of NI IO-module*)
0008
0009 Measuring:=TRUE; (*variable is set to TRUE when system is performing the measuring sequence.*)
0010
0011
0012

```

Sequence\_flat (FB-SFC).Transition Trans1 (ST)

```

0001 StopSignal

```

Sequence\_flat (FB-SFC).Transition Trans2 (ST)

```

0001 NOT StopSignal
0002 (*If stopsignal is FALSE the sequence is interrupted*)
0003 (*Jump to Init returns to main program*)
0004 (*See stopfunction(FB) what is required to set the stop signal to TRUE*)

```

```

Sequence_flat (FB-SFC).Action Step2 (ST)
0001 SEQUENCEINTERRUPTED:=TRUE;      (*With these two variables the system knows which sequence is interrupted*)
0002 HOMESEQUENCE1:=TRUE;             (*and knows which home sequence is started*)
0003
0004 %QX8.3:=FALSE;                   (*PLCs output are set to false*)
0005
0006
0007 Measuring:=FALSE;                 (*Sequence is interrupted and therefore "Measuring" variable is set to FALSE*)
0008
0009
0010
0011

Sequence_flat (FB-SFC).Transition Trans3 (ST)
0001 HOMESEQUENCE1                   (*in order to continue, variables set to TRUE and FALSE are checked, if they are in the state the were set*)
0002 AND SEQUENCEINTERRUPTED
0003 AND NOT %QX8.3
0004 AND NOT Measuring
0005 AND StopSignal                   (*Stop signal must be present. For example EMG must be acknowledged if it's been activated*)

Sequence_flat (FB-SFC).Action Step3 (ST)
0001 (*Output Q8.0 is a bit used to confirm the template in Viewscan*)
0002 %QX8.0:=TRUE;

Sequence_flat (FB-SFC).Transition Trans4 (ST)
0001 %QX8.0
0002 AND StopSignal                   (*Stop signal is needed in order to continue*)

Sequence_flat (FB-SFC).Transition Trans4 (ST)
0001 %QX8.0
0002 AND StopSignal                   (*Stop signal is needed in order to continue*)

Sequence_flat (FB-SFC).Transition Trans5 (ST)
0001 NOT StopSignal                   (*Deactivated stop signal stops the sequence*)

Sequence_flat (FB-SFC).Action Step4 (ST)
0001 SEQUENCEINTERRUPTED:=TRUE;
0002 HOMESEQUENCE1:=TRUE;
0003
0004 %QX8.0:=FALSE;                   (*output set to FALSE*)
0005 %QX8.3:=FALSE;
0006 Measuring:=FALSE;

Sequence_flat (FB-SFC).Transition Trans6 (ST)
0001 SEQUENCEINTERRUPTED
0002 AND HOMESEQUENCE1
0003 AND NOT %QX8.0
0004 AND NOT %QX8.1
0005 AND NOT %QX8.2
0006 AND NOT %QX8.3
0007 AND NOT Measuring
0008 AND StopSignal

Sequence_flat (FB-SFC).Action Step5 (ST)
0001 %QX8.0:=FALSE;
0002 %QX8.1:=FALSE;
0003 %QX8.2:=FALSE;
0004 %QX8.3:=FALSE;
0005 (*Outputs are set to FALSE to avoid errors in ViewScan*)

Sequence_flat (FB-SFC).Transition Trans7 (ST)
0001 StopSignal
0002 AND NOT %QX8.0
0003 AND NOT %QX8.1
0004 AND NOT %QX8.2
0005 AND NOT %QX8.3
0006 AND DC_END0
0007 AND NOT DC_END1
0008
0009 (*if the sensor is in correct position, the sequence continues via this branch*)
0010

Sequence_flat (FB-SFC).Transition Trans8 (ST)
0001 StopSignal
0002 AND NOT %QX8.0
0003 AND NOT %QX8.1
0004 AND NOT %QX8.2
0005 AND NOT %QX8.3
0006 AND ((DC_END1
0007 AND NOT DC_END0)
0008 OR ( NOT DC_END0
0009 AND NOT DC_END1)))
0010
0011 (*Sensor will be turned in correct position, if this branch is chosen.*)
0012 (*The choice is made with switches monitoring the sensors position*)

```

## Sequence\_flat (FB-SFC).Action Step6 - Entry (ST)

```

0001 RecordNoY:=3; (*servos are operated in record selection mode.*)
0002          (* This is the number of the record which is executed next*)
0003
0004 (*Additional to a step action you can add an entry action and an exit action to a step.
0005 An entry action is executed only once, right after the step has become active.
0006 An exit action is executed only once before the step is deactivated.
0007 A step with entry action is indicated by an "E" in the lower left corner,
0008 the exit action by an "X" in the lower right corner.
0009 The entry and exit action can be implemented in any language.*)

```

## Sequence\_flat (FB-SFC).Action Step6 (ST)

```

0001 StartTaskY:=Step6; (*Rising edge in StartTask trigger the positioning task.*)
0002          (*Step6 is a local variable*)

```

## Sequence\_flat (FB-SFC).Transition Trans10 (ST)

```

0001 ActualRecordNoY = 3 (*Actual record on the controller corresponds to the one set on entry action*)
0002 AND Y_MOTIONCOMPL (*Motion is complete and drive isn't moving*)
0003 AND StopSignal (*Stop signal is present*)

```

## Sequence\_flat (FB-SFC).Transition Trans11 (ST)

```

0001 NOT StopSignal (*Sequence stopped bu user*)
0001 SEQUENCEINTERRUPTED:=TRUE;
0002 HOMESEQUENCE1:=TRUE;
0003 Measuring:=FALSE;
0004 StartTaskY:=FALSE; (*StartTask set to FALSE*)
0005

```

## Sequence\_flat (FB-SFC).Transition Trans13 (ST)

```

0001 SEQUENCEINTERRUPTED
0002 AND HOMESEQUENCE1
0003 AND NOT Measuring
0004 AND StopSignal
0005 AND NOT StartTaskY

```

## Sequence\_flat (FB-SFC).Transition Trans9 (ST)

```

0001 NOT StopSignal (*Sequence stopped by user*)

```

## Sequence\_flat (FB-SFC).Action Step7 (ST)

```

0001 SEQUENCEINTERRUPTED:=TRUE;
0002 HOMESEQUENCE1:=TRUE;
0003 Measuring:=FALSE;

```

## Sequence\_flat (FB-SFC).Transition Trans12 (ST)

```

0001 HOMESEQUENCE1
0002 AND SEQUENCEINTERRUPTED
0003 AND NOT Measuring
0004 AND StopSignal

```

## Sequence\_flat (FB-SFC).Action Step9 (ST)

```

0001 StartTaskY:=FALSE; (*just to make sure the starttask is false*)
0002 Sequence_flat.TONInst(In:=DC_END0, PT:=T#500ms); (*Turn on delay. DC motor moves for 500ms*)
0003          (* after the switch is activated*)
0004
0005 DC_RWD:=STEP9
0006 AND NOT Sequence_flat.TONInst.Q
0007 AND StopSignal;
0008
0009 (*If DC_END0 isn't activated in 2s, which is the maximum step time,
0010 the SFCErrror is set to TRUE and the sequence is stopped. Reset and restart are required*)

```

## Sequence\_flat (FB-SFC).Transition Trans14 (ST)

```

0001 StopSignal
0002 AND Sequence_flat.TONInst.Q (*When "turn on" timers output is TRUE, the sequence continues *)

```

## Sequence\_flat (FB-SFC).Transition Trans69 (ST)

```

0001 NOT StopSignal (*Sequence stopped by user*)

```

## Sequence\_flat (FB-SFC).Action Step69 (ST)

```

0001 SEQUENCEINTERRUPTED:=TRUE;
0002 HOMESEQUENCE1:=TRUE;
0003 Measuring:=FALSE;
0004 DC_RWD:=FALSE; (*Variable controlling the sensor rotation is set to FALSE*)

```

```

Sequence_flat (FB-SFC).Transition Trans70 (ST)
0001 SEQUENCEINTERRUPTED
0002 AND HOMESEQUENCE1
0003 AND NOT Measuring
0004 AND NOT DC_RWD
0005 AND StopSignal

Sequence_flat (FB-SFC).Action Step10 - Entry (ST)
0001 RecordNoY:=1; (*RecordNo 1 = home position *)

Sequence_flat (FB-SFC).Action Step10 (ST)
0001 StartTaskY:=STEP10;
0002 DC_RWD:=FALSE;
0003
0004 (*Yaxis is driven to home position after the sensor rotation*)

Sequence_flat (FB-SFC).Transition Trans16 (ST)
0001 ActualRecordNoY = 1 (*The record number on the drive corresponds to the one set in entry action*)
0002 AND Y_MOTIONCOMPL (*Axis has stopped*)
0003 AND StopSignal

Sequence_flat (FB-SFC).Transition Trans17 (ST)
0001 NOT StopSignal (*Sequence stopped by user*)

Sequence_flat (FB-SFC).Action Step12 (ST)
0001 SEQUENCEINTERRUPTED:=TRUE;
0002 HOMESEQUENCE1:=TRUE;
0003 Measuring:=FALSE;
0004 StartTaskY:=FALSE;

Sequence_flat (FB-SFC).Transition Trans19 (ST)
0001 SEQUENCEINTERRUPTED
0002 AND HOMESEQUENCE1
0003 AND NOT Measuring
0004 AND StopSignal
0005 AND NOT StartTaskY

Sequence_flat (FB-SFC).Transition Trans15 (ST)
0001 SFCErrror (*SFCErrror stops sequence*)

Sequence_flat (FB-SFC).Action Step11 (ST)
0001 Measuring:=FALSE;
0002 DC_RWD:=FALSE;
0003
0004
0005
0006

Sequence_flat (FB-SFC).Action Step13 - Entry (ST)
0001 RecordNoZ2:=2;
0002 (*Preselection of record number for record selection*)
0003 (*If you want to change the speed the axis moves, you must edit the position profile in the FCT*)

Sequence_flat (FB-SFC).Action Step13 (ST)
0001 StartTaskZ2:=STEP13; (*Positioning task starts*)
0002 StartTaskY:=FALSE;

Sequence_flat (FB-SFC).Transition Trans20 (ST)
0001 ActualRecordNoZ2 = 2
0002 AND Z2_MOTIONCOMPL
0003 AND StopSignal
0004 (*Z2axis has reached it's destination*)

Sequence_flat (FB-SFC).Action Step14 (ST)
0001 StartTaskZ2:=FALSE;

Sequence_flat (FB-SFC).Transition Trans22 (ST)
0001 NOT StartTaskZ2
0002 AND StopSignal

```



```

Sequence_flat (FB-SFC).Transition Trans23 (ST)
0001 NOT StopSignal

Sequence_flat (FB-SFC).Action Step16 (ST)
0001 SEQUENCEINTERRUPTED:=TRUE;
0002 HOMESEQUENCE1:=TRUE;
0003 Measuring:=FALSE;

Sequence_flat (FB-SFC).Transition Trans25 (ST)
0001 HOMESEQUENCE1
0002 AND SEQUENCEINTERRUPTED
0003 AND NOT Measuring
0004 AND StopSignal

Sequence_flat (FB-SFC).Action Step17 - Entry (ST)
0001 RecordNoZ2:=3;
0001 StartTaskZ2:=STEP17;
0002 Activate_K4:=TRUE; (*Contactor K4 activated*)

Sequence_flat (FB-SFC).Transition Trans26 (ST)
0001 ActualRecordNoZ2 = 3
0002 AND Z2_MOTIONCOMPL
0003 AND StopSignal
0004 AND K4_OK

Sequence_flat (FB-SFC).Transition Trans27 (ST)
0001 NOT StopSignal

Sequence_flat (FB-SFC).Action Step18 (ST)
0001 SEQUENCEINTERRUPTED:=TRUE;
0002 HOMESEQUENCE1:=TRUE;
0003 Measuring:=FALSE;
0004 Activate_K4:=FALSE; (*K4 deactivated*)
0005 StartTaskZ2:=FALSE;

Sequence_flat (FB-SFC).Transition Trans28 (ST)
0001 HOMESEQUENCE1
0002 AND SEQUENCEINTERRUPTED
0003 AND NOT Measuring
0004 AND NOT K4_OK
0005 AND StopSignal
0006 AND NOT StartTaskZ2

Sequence_flat (FB-SFC).Transition Trans21 (ST)
0001 NOT StopSignal (*Sequence stopped by user or error*)

Sequence_flat (FB-SFC).Action Step15 (ST)
0001 SEQUENCEINTERRUPTED:=TRUE;
0002 HOMESEQUENCE1:=TRUE;
0003 Measuring:=FALSE;
0004 StartTaskZ2:=FALSE;
0005
0006

Sequence_flat (FB-SFC).Transition Trans24 (ST)
0001 HOMESEQUENCE1
0002 AND SEQUENCEINTERRUPTED
0003 AND NOT Measuring
0004 AND StopSignal
0005 AND NOT StartTaskZ2

Sequence_flat (FB-SFC).Action Step19 - Entry (ST)
0001 SetPosRot:= ActualPositionRot + 4;
0002 SetVeloRot:=41; (*Rotation speed is 41% from the maximum speed --> 0.41 * 46 = ca.19 rpm*)
0003 (*Rotation is controlled in direct mode*)
0004 (*Speed in % from the maximum velocity and desired position are sent to the controller by PLC*)
0005 (*See FHPP manual for more information on operation modes*)

```

## Sequence\_flat (FB-SFC).Action Step19 (ST)

```

0001 StartTaskZ:=FALSE;
0002 StartTaskRot:=STEP19;
0003 (*Signal sent to RollScan and magnetizing ramp starts. DEMAGN is set to true when rotation starts*)
0004
0005 IF AckStartRot (*AckStartRot is acknowledgement sent by the controller. The positioning task is being processed*)
0006 AND K4_OK (*K4 for must activated and working properly*)
0007 THEN DEMAGN:=TRUE; (*Rising edge on variable "DEMAGN" triggers a magnetization ramp in rollscan.*)
0008 END_IF;
0009
0010
0011

```

## Sequence\_flat (FB-SFC).Transition Trans29 (ST)

```

0001 DEMAGN
0002 AND ROT_MOTIONCOMPL
0003 AND StopSignal
0004 (*Sequence will continue when motion is complete*)
0001 NOT StopSignal

```

## Sequence\_flat (FB-SFC).Action Step20 (ST)

```

0001 SEQUENCEINTERRUPTED:=TRUE;
0002 HOMESEQUENCE1:=TRUE;
0003 Measuring:=FALSE;
0004 Activate_K4:=FALSE; (*K4 deactivated*)
0005 StartTaskRot:=FALSE;
0006

```

## Sequence\_flat (FB-SFC).Transition Trans31 (ST)

```

0001 HOMESEQUENCE1
0002 AND SEQUENCEINTERRUPTED
0003 AND NOT Measuring
0004 AND StopSignal
0005 AND NOT K4_OK (*K4 is deactivated*)
0006 AND NOT StartTaskRot

```

## Sequence\_flat (FB-SFC).Action Step21 - Entry (ST)

```

0001 RecordNoZ:=2;
0002 (*If you want to change the speed the axis moves, you must edit the position profile in the FCT*)
0003

```

## Sequence\_flat (FB-SFC).Action Step21 (ST)

```

0001 StartTaskRot:=FALSE;
0002 StartTaskZ:=STEP21;

```

## Sequence\_flat (FB-SFC).Transition Trans32 (ST)

```

0001 ActualRecordNoZ = 2
0002 AND Z_MOTIONCOMPL
0003 AND StopSignal
0004

```

## Sequence\_flat (FB-SFC).Action Step22 - Entry (ST)

```

0001 RecordNoY:=2;

```

## Sequence\_flat (FB-SFC).Action Step22 (ST)

```

0001 StartTaskZ:=FALSE;
0002 StartTaskY:=STEP22;

```

## Sequence\_flat (FB-SFC).Transition Trans34 (ST)

```

0001 ActualRecordNoY = 2
0002 AND Y_MOTIONCOMPL
0003 AND StopSignal

```

## Sequence\_flat (FB-SFC).Transition Trans35 (ST)

```

0001 NOT StopSignal

```

## Sequence\_flat (FB-SFC).Action Step24 (ST)

```

0001 SEQUENCEINTERRUPTED:=TRUE;
0002 HOMESEQUENCE1:=TRUE;
0003 Measuring:=FALSE;
0004 Activate_K4:=FALSE;
0005 StartTaskY:=FALSE;

```

## Sequence\_flat (FB-SFC).Transition Trans37 (ST)

```

0001 HOMESEQUENCE1
0002 AND SEQUENCEINTERRUPTED
0003 AND NOT Measuring
0004 AND StopSignal
0005 AND NOT K4_OK
0006 AND NOT StartTaskY

```

```

Sequence_flat (FB-SFC).Transition Trans33 (ST)
0001 NOT StopSignal

Sequence_flat (FB-SFC).Action Step23 (ST)
0001 SEQUENCEINTERRUPTED:=TRUE;
0002 HOMESEQUENCE1:=TRUE;
0003 Measuring:=FALSE;
0004 Activate_K4:=FALSE;
0005 StartTaskZ:=FALSE;

Sequence_flat (FB-SFC).Transition Trans36 (ST)
0001 HOMESEQUENCE1
0002 AND SEQUENCEINTERRUPTED
0003 AND NOT Measuring
0004 AND StopSignal
0005 AND NOT K4_OK
0006 AND NOT StartTaskZ

Sequence_flat (FB-SFC).Action Step25 - Entry (ST)
0001 SetPosRot:=ActualPositionRot + 100; (*100 is added to the actual position
0002 because the purpose is to rotate the washer ring as long as it takes to perform measurement*)
0003 (*The rotation speed has already been set in step19*)

Sequence_flat (FB-SFC).Action Step25 (ST)
0001 StartTaskY:=FALSE;
0002 StartTaskRot:=STEP25;

Sequence_flat (FB-SFC).Transition Trans38 (ST)
0001 Rotation_CTRL.ActualVelocity = 41 (*When rotation has reached correct velocity the sequence moves on*)
0002 AND StopSignal

Sequence_flat (FB-SFC).Action Step26 - Entry (ST)
0001 RecordNoZ:=3;
0002 (*If you want to change the speed the axis moves, you must edit the position profile in the FCT*)

Sequence_flat (FB-SFC).Action Step26 (ST)
0001 StartTaskRot:=FALSE;
0002 Activate_RSTRIG:=TRUE; (*Trigger to RollScan is controlled with this variable. As long as the variable is true the rollscan measures*)
0003 StartTaskZ:=STEP26; (*If you want to change the speed the axis moves, you must edit the position profile in the FCT*)

Sequence_flat (FB-SFC).Transition Trans40 (ST)
0001 Z_MOTIONCOMPL
0002 AND ActualRecordNoZ = 3
0003 AND StopSignal

Sequence_flat (FB-SFC).Action Step28 (ST)
0001 StartTaskZ:=FALSE;
0002 Activate_RSTRIG:=FALSE; (*Trigger off*)
0003 HALTROT:=TRUE; (*See PLC_PRG: HALTROT is negated. If set to TRUE, the positioning stops*)
0004 (*The axis stops with defined braking ramp and the positioning task remains active.
0005 If StopSignal is removed, the drive stops with maximum braking ramp and positioning task is removed.*)
0006 %QX8.1:=TRUE; (*END bit to ViewScan*)
0007 DEMAGN:=FALSE;

Sequence_flat (FB-SFC).Transition Trans43 (ST)
0001 NOT Rotation_CTRL.DrivesMoving
0002 AND NOT DEMAGN
0003 AND StopSignal

Sequence_flat (FB-SFC).Transition Trans44 (ST)
0001 NOT StopSignal

Sequence_flat (FB-SFC).Action Step30 (ST)
0001 SEQUENCEINTERRUPTED:=TRUE;
0002 HOMESEQUENCE1:=TRUE;
0003 Measuring:=FALSE;
0004 Activate_K4:=FALSE;
0005 Activate_RSTRIG:=FALSE;
0006 Rotation_CTRL.ClearRemainingPosition:=TRUE;
0007 %QX8.1:=FALSE;

Sequence_flat (FB-SFC).Action Step70 (ST)
0001 Rotation_CTRL.ClearRemainingPosition:=FALSE;
0002 HALTROT:=FALSE;

```

Sequence\_flat (FB-SFC).Transition Trans46 (ST)

```

0001 SEQUENCEINTERRUPTED
0002 AND HOMESEQUENCE1
0003 AND NOT Measuring
0004 AND NOT K4_OK
0005 AND NOT Activate_RSTRIG
0006 AND StopSignal

```

Sequence\_flat (FB-SFC).Transition Trans41 (ST)

```

0001 NOT StopSignal

```

Sequence\_flat (FB-SFC).Action Step29 (ST)

```

0001 SEQUENCEINTERRUPTED:=TRUE;
0002 HOMESEQUENCE1:=TRUE;
0003 Measuring:=FALSE;
0004 Activate_K4:=FALSE; (*K4 deactivated*)
0005 Activate_RSTRIG:=FALSE; (*Trigger deactivated, measuring stopped*)
0006 StartTaskZ:=FALSE;
0007 %QX8.1:=TRUE; (*Output Q8.1 ends the measuring in ViewScan*)

```

Sequence\_flat (FB-SFC).Transition Trans45 (ST)

```

0001 SEQUENCEINTERRUPTED
0002 AND HOMESEQUENCE1
0003 AND NOT Measuring
0004 AND NOT K4_OK
0005 AND NOT Activate_RSTRIG
0006 AND NOT StartTaskZ

```

Sequence\_flat (FB-SFC).Action Step72 (ST)

```

0001 %QX8.1:=FALSE; (*Output is set to false to avoid errors in ViewScan*)

```

Sequence\_flat (FB-SFC).Transition Trans39 (ST)

```

0001 NOT StopSignal

```

Sequence\_flat (FB-SFC).Action Step27 (ST)

```

0001 SEQUENCEINTERRUPTED:=TRUE;
0002 HOMESEQUENCE1:=TRUE;
0003 Measuring:=FALSE;
0004 Activate_K4:=FALSE;
0005 StartTaskRot:=FALSE;
0006 %QX8.1:=TRUE; (*Output Q8.1 ends the measuring in ViewScan*)

```

Sequence\_flat (FB-SFC).Transition Trans42 (ST)

```

0001 HOMESEQUENCE1
0002 AND SEQUENCEINTERRUPTED
0003 AND NOT K4_OK
0004 AND NOT Measuring
0005 AND StopSignal
0006 AND NOT StartTaskRot

```

Sequence\_flat (FB-SFC).Action Step71 (ST)

```

0001 %QX8.1:=FALSE; (*Output set to false in ViewScan*)

```

Sequence\_flat (FB-SFC).Action Step32 - Entry (ST)

```

0001 RecordNoY:=1;

```

Sequence\_flat (FB-SFC).Action Step32 (ST)

```

0001 Activate_K4:=FALSE;
0002 StartTaskY:=STEP32;
0003 Rotation_CTRL.ClearRemainingPosition:=TRUE; (*Remaining distance of the positioning task is deleted with a rising edge*)
0004 %QX8.1:=FALSE;

```

Sequence\_flat (FB-SFC).Transition Trans47 (ST)

```

0001 Y_MOTIONCOMPL
0002 AND ActualRecordNoY = 1
0003 AND NOT K4_OK
0004 AND StopSignal

```

```

Sequence_flat (FB-SFC).Action Step33 - Entry (ST)
0001 RecordNoZ:=1;

Sequence_flat (FB-SFC).Action Step33 (ST)
0001 StartTaskY:=FALSE;
0002 StartTaskZ:=STEP33;
0003 HALTROT:=FALSE; (*Rotation is possible*)
0004 Rotation_CTRL.ClearRemainingPosition:=FALSE;

Sequence_flat (FB-SFC).Transition Trans49 (ST)
0001 Z_MOTIONCOMPL
0002 AND ActualRecordNoZ = 1
0003 AND StopSignal

Sequence_flat (FB-SFC).Transition Trans50 (ST)
0001 NOT StopSignal

Sequence_flat (FB-SFC).Action Step36 (ST)
0001 SEQUENCEINTERRUPTED:=TRUE;
0002 HOMESEQUENCE1:=TRUE;
0003 Measuring:=FALSE;
0004 StartTaskZ:=FALSE;

Sequence_flat (FB-SFC).Transition Trans54 (ST)
0001 SEQUENCEINTERRUPTED
0002 AND HOMESEQUENCE1
0003 AND NOT Measuring
0004 AND StopSignal
0005 AND NOT StartTaskZ

Sequence_flat (FB-SFC).Transition Trans48 (ST)
0001 NOT StopSignal

Sequence_flat (FB-SFC).Action Step34 (ST)
0001 SEQUENCEINTERRUPTED:=TRUE;
0002 HOMESEQUENCE1:=TRUE;
0003 Measuring:=FALSE;
0004 StartTaskY:=FALSE;
0005 HALTROT:=FALSE;
0006 Rotation_CTRL.ClearRemainingPosition:=FALSE;

Sequence_flat (FB-SFC).Transition Trans51 (ST)
0001 SEQUENCEINTERRUPTED
0002 AND HOMESEQUENCE1
0003 AND NOT Measuring
0004 AND NOT K4_OK
0005 AND StopSignal
0006 AND ROT_MOTIONCOMPL

Sequence_flat (FB-SFC).Action Step39 (ST)
0001 StartTaskZ:=FALSE;
0002 Activate_K4:=FALSE;

Sequence_flat (FB-SFC).Transition Trans58 (ST)
0001 StopSignal
0002 AND NOT K4_OK

Sequence_flat (FB-SFC).Transition Trans59 (ST)
0001 NOT StopSignal

Sequence_flat (FB-SFC).Action Step41 (ST)
0001 SEQUENCEINTERRUPTED:=TRUE;
0002 HOMESEQUENCE1:=TRUE;
0003 Measuring:=FALSE;

Sequence_flat (FB-SFC).Transition Trans61 (ST)
0001 HOMESEQUENCE1
0002 AND SEQUENCEINTERRUPTED
0003 AND NOT Measuring
0004 AND NOT K4_OK
0005 AND StopSignal

```

Sequence\_flat (FB-SFC).Action Step42 - Entry (ST)

0001 RecordNoZ2:=1;

Sequence\_flat (FB-SFC).Action Step42 (ST)

0001 StartTaskZ2:=STEP42;

0002

0003

Sequence\_flat (FB-SFC).Transition Trans62 (ST)

0001 Z2\_MOTIONCOMPL

0002 AND ActualRecordNoZ2 = 1

0003 AND StopSignal

Sequence\_flat (FB-SFC).Action Step43 (ST)

0001 StartTaskZ2:=FALSE;

0002 Measuring:=FALSE;

Sequence\_flat (FB-SFC).Transition Trans64 (ST)

0001 NOT Measuring

0002 AND StopSignal

Sequence\_flat (FB-SFC).Transition Trans63 (ST)

0001 NOT StopSignal

Sequence\_flat (FB-SFC).Action Step44 (ST)

0001 SEQUENCEINTERRUPTED:=TRUE;

0002 HOMESEQUENCE1:=TRUE;

0003 Measuring:=FALSE;

0004 StartTaskZ2:=FALSE;

Sequence\_flat (FB-SFC).Transition Trans65 (ST)

0001 SEQUENCEINTERRUPTED

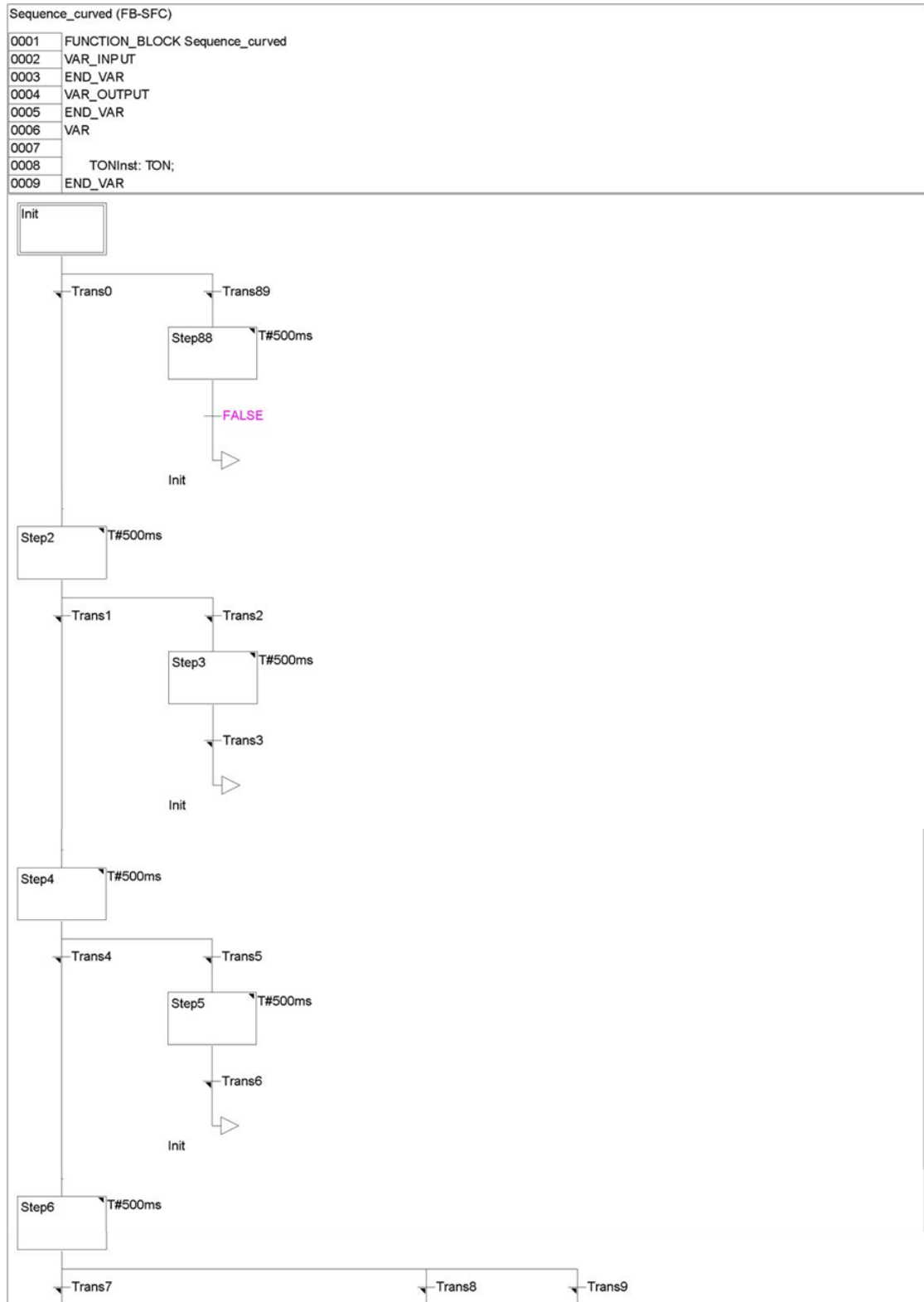
0002 AND HOMESEQUENCE1

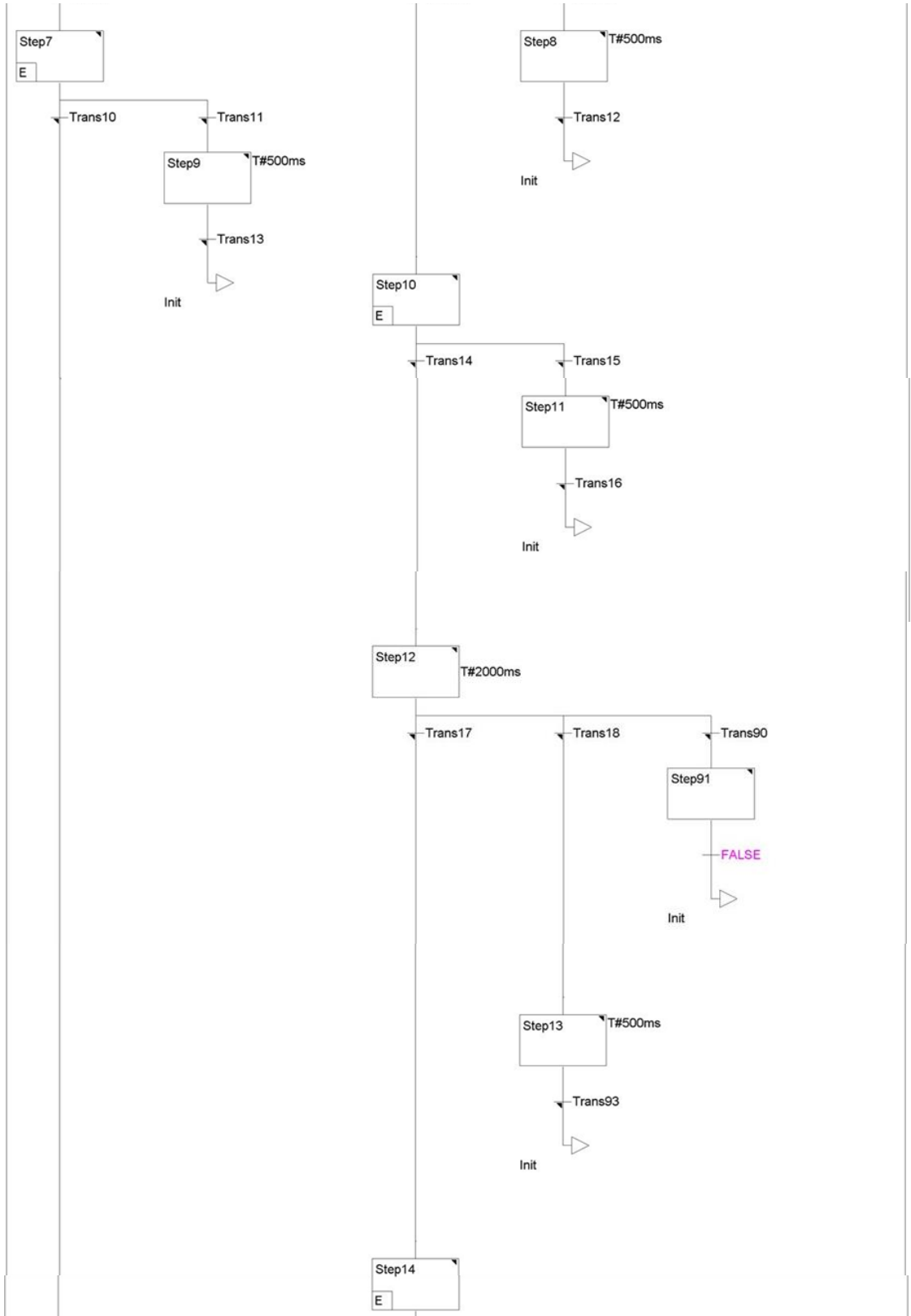
0003 AND NOT Measuring

0004 AND StopSignal

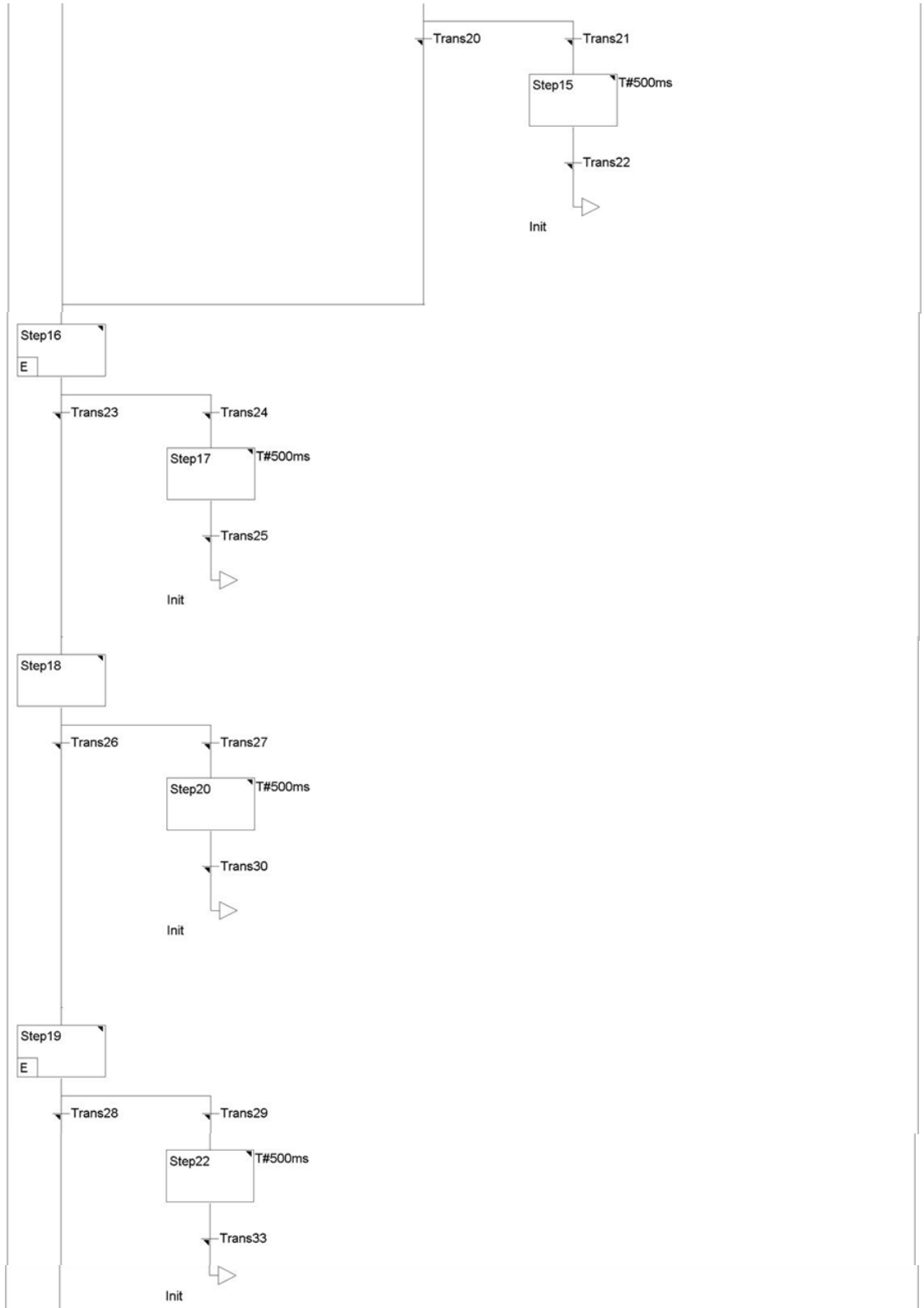
0005 AND NOT StartTaskZ2

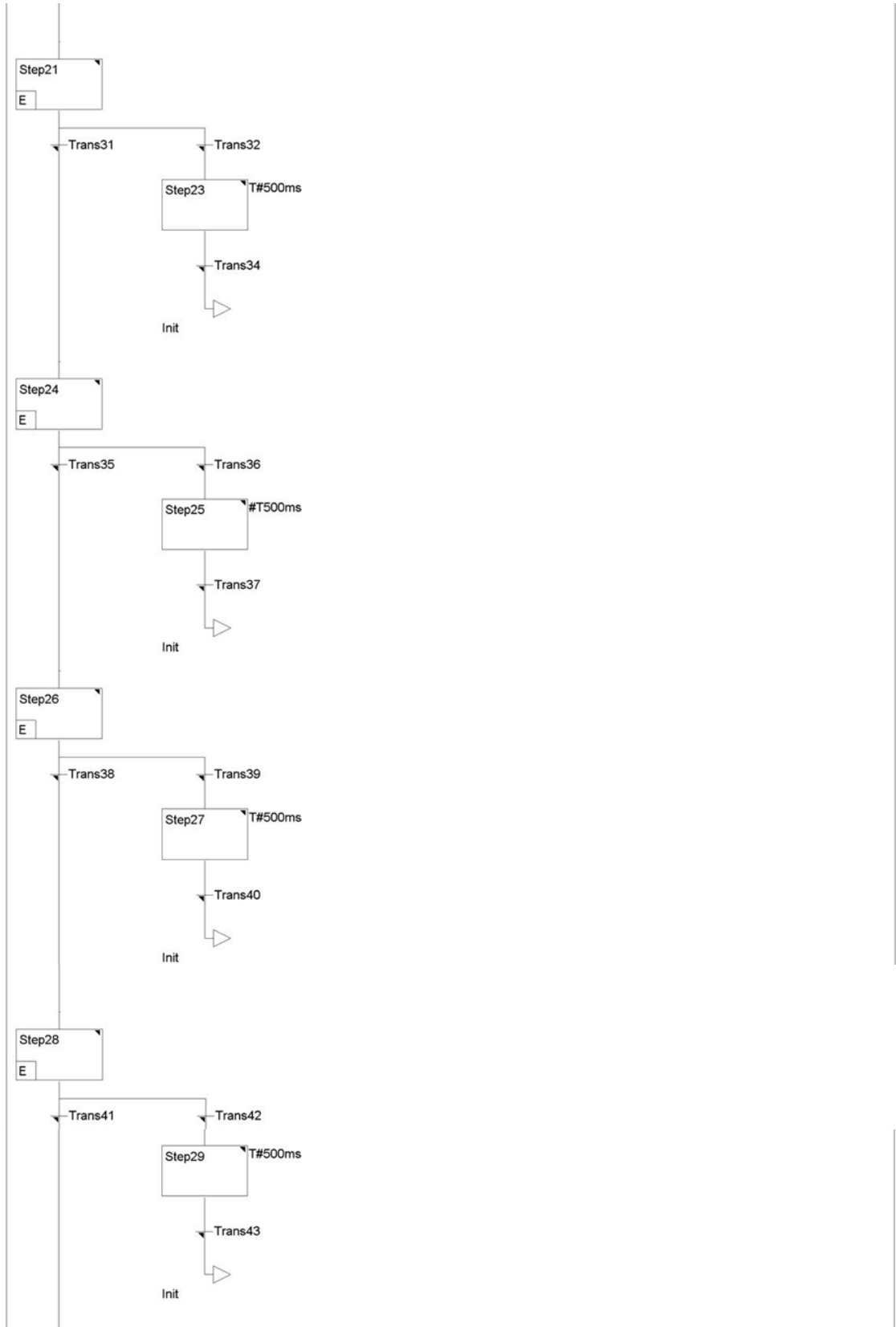
## Liite 15. SEQUENCE CURVED

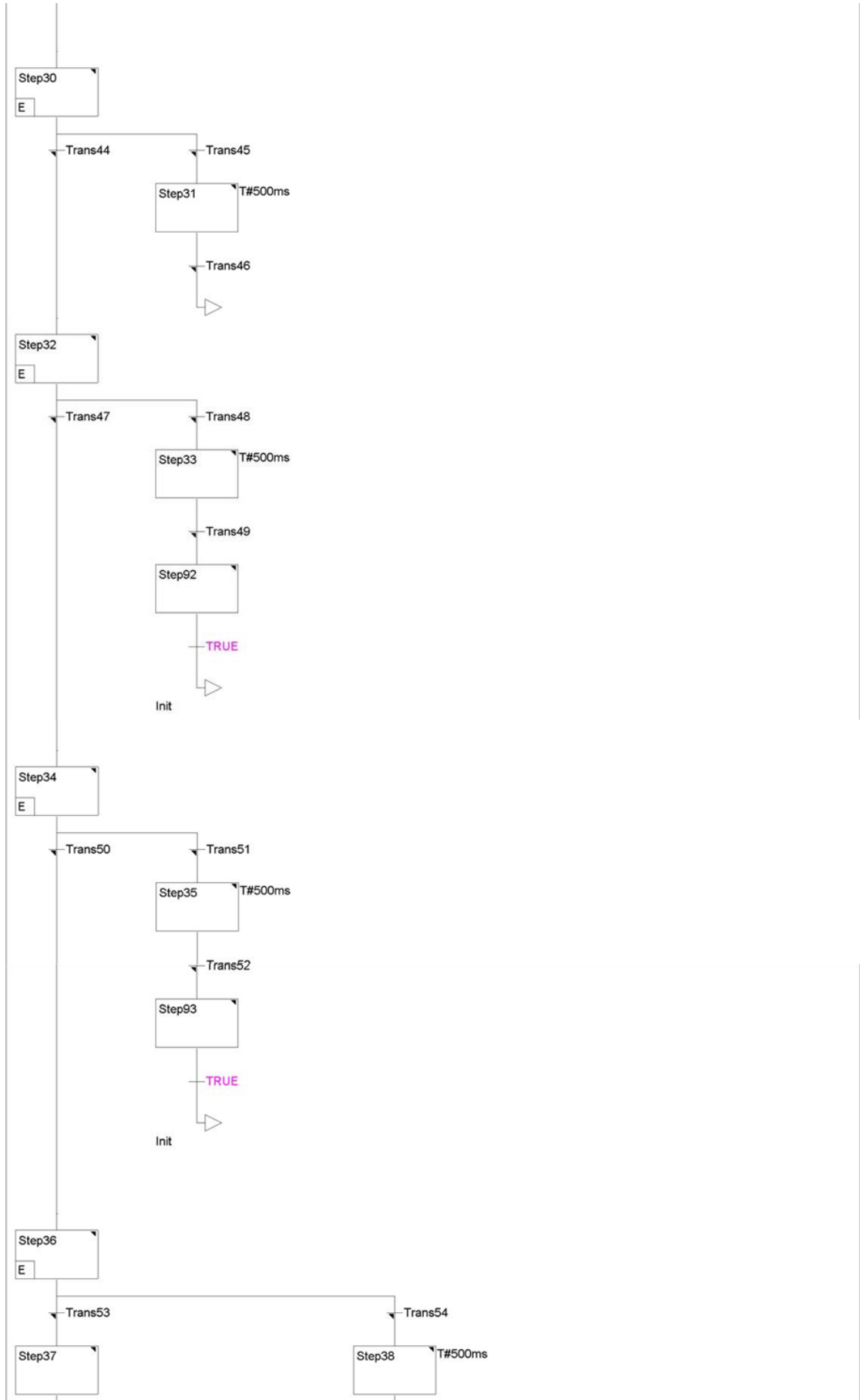


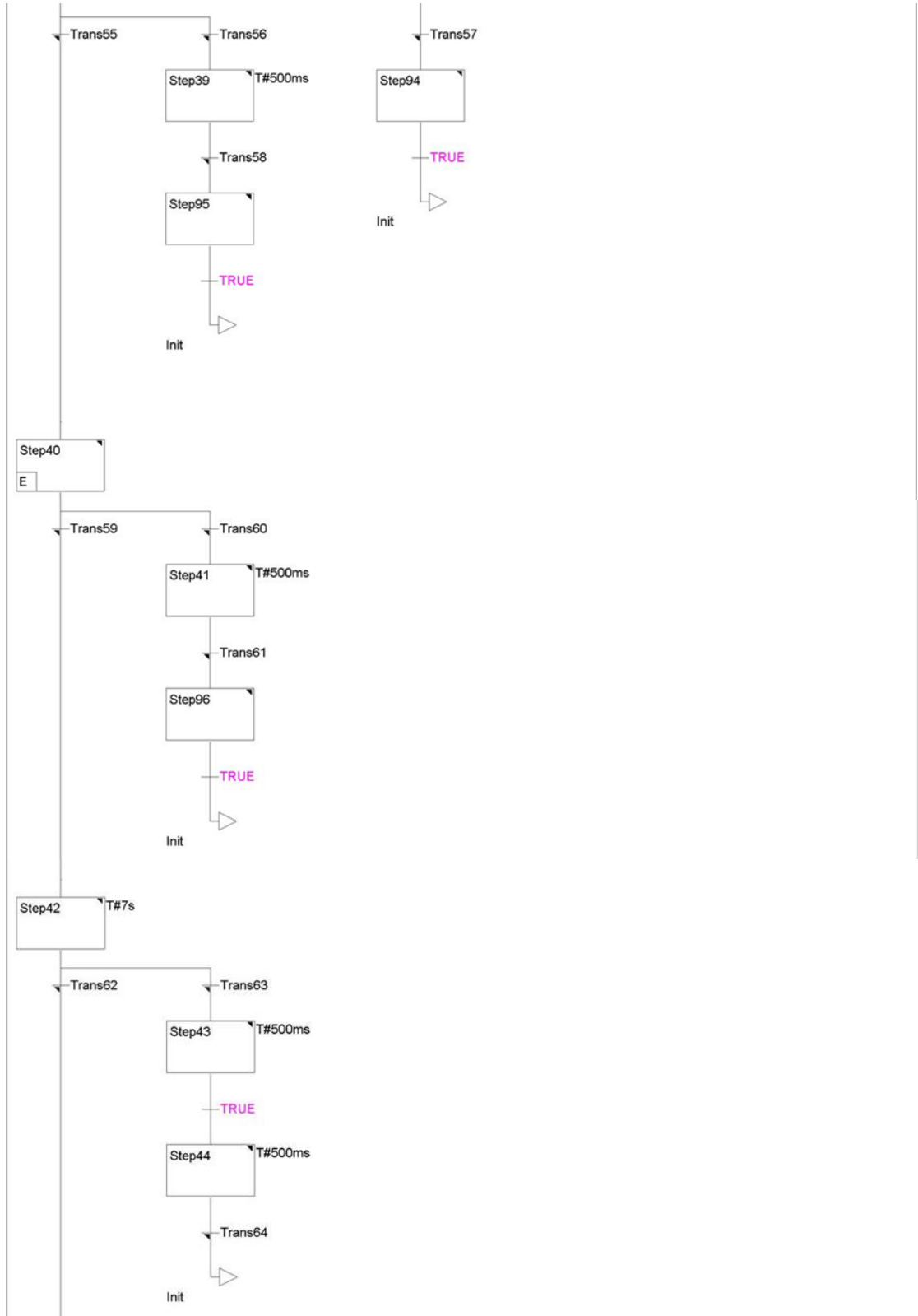


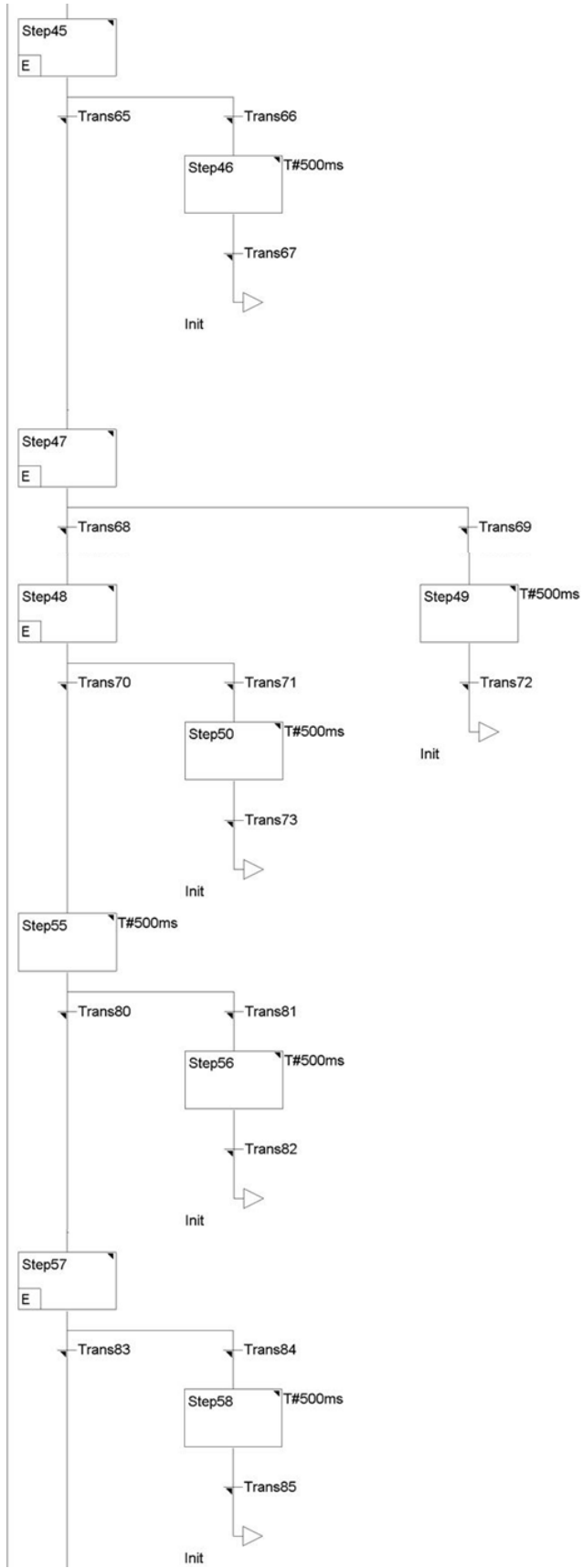


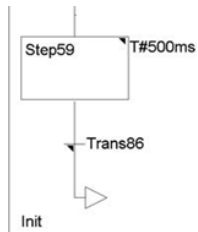












Sequence\_curved (FB-SFC), Transition Trans0 (ST)

```

0001 (*Between the steps there are so-called transitions.
0002
0003 A transition condition must have the value TRUE OR FALSE. Thus it can consist OF either a boolean variable,
0004 a boolean address OR a boolean CONSTANT. It can also contain a series OF instructions having a boolean result,
0005 either in ST syntax (e.g. (i<= 100) AND b) OR in any language desired.
0006 But a transition may NOT contain programs, FUNCTION blocks OR assignments!*)
0007
0008 ReadyForOperation
0009 AND (TML2_SEL
0010 OR REF2_SEL)
0011 AND START_BUTTON
0012 AND (DC_END0
0013 OR DC_END1)
0014
0015 (*Sequence starts if machine is ready for operation (drives enabled etc) and start button is activated*)

```

Sequence\_curved (FB-SFC), Transition Trans89 (ST)

```

0001 ReadyForOperation
0002 AND (TML2_SEL
0003 OR REF2_SEL)
0004 AND START_BUTTON
0005 AND (NOT DC_END0
0006 AND NOT DC_END1)
0007
0008 (*if neither one of the switches monitoring the sensor isn't active, the system is set to "error state" in following step*)

```

Sequence\_curved (FB-SFC), Action Step88 (ST)

```

0001 SFCErr:=TRUE;
0002 Measuring:=FALSE;

```

Sequence\_curved (FB-SFC), Action Step2 (ST)

```

0001 (*With template selector switch the plc sends information to viewscan via National Instruments IO-module*)
0002 (*The used template is selected with 2 bits*)
0003
0004 (*IF TML2_SEL THEN TML2:=TRUE;
0005 ELIF REF2_SEL THEN TML3:=TRUE;
0006 ELSE TML2:=FALSE;TML3:=FALSE;
0007 END_IF;*)
0008
0009 IF TML2_SEL THEN %QX8.2:=TRUE;
0010 ELIF REF2_SEL THEN %QX8.2:=TRUE;%QX8.3:=TRUE;
0011 ELSE %QX8.2:=FALSE;%QX8.3:=FALSE;
0012 END_IF;
0013
0014
0015 Measuring:=TRUE; (*variable is set to TRUE when system is performing the measuring sequence.*)
0016
0017

```

Sequence\_curved (FB-SFC), Transition Trans1 (ST)

```

0001 %QX8.2
0002 OR (%QX8.2 AND %QX8.3) (*if correct output is on and stopsignal is present, then the sequence is continued*)
0003 AND StopSignal

```

Sequence\_curved (FB-SFC), Transition Trans2 (ST)

```

0001 NOT StopSignal

```

Sequence\_curved (FB-SFC), Action Step3 (ST)

```

0001 SEQUENCE2INTERRUPTED:=TRUE; (*With these two variables the system knows which sequence is interrupted*)
0002 HOMESEQUENCE2:=TRUE; (*and knows which home sequence is started*)
0003
0004 %QX8.2:=FALSE; (*PLCs outputs are set to false*)
0005 %QX8.3:=FALSE;
0006
0007 Measuring:=FALSE; (*Sequence is interrupted and therefore "Measuring" variable is set to FALSE*)

```

```

Sequence_curved (FB-SFC).Transition Trans3 (ST)
0001 HOMESEQUENCE2 (*in order to continue, variables set to TRUE and FALSE are checked, if they are in the state the were set*)
0002 AND SEQUENCE2INTERRUPTED
0003 AND NOT %QX8.2
0004 AND NOT %QX8.3
0005 AND NOT Measuring
0006 AND StopSignal (*Stop signal must be present. For example EMG must be acknowledged if it's been activated*)

Sequence_curved (FB-SFC).Action Step4 (ST)
0001 (*Q8.0 is a bit used to confirm the template in Viewscan*)
0002 %QX8.0:=TRUE;

Sequence_curved (FB-SFC).Transition Trans4 (ST)
0001 %QX8.0 (*Selection bit is on*)
0002 AND StopSignal (*Stop signal is needed in order to continue*)

Sequence_curved (FB-SFC).Transition Trans5 (ST)
0001 NOT StopSignal (*Deactivated stop signal stops the sequence*)
Sequence_curved (FB-SFC).Action Step5 (ST)
0001 SEQUENCE2INTERRUPTED:=TRUE;
0002 HOMESEQUENCE2:=TRUE;
0003
0004 %QX8.0:=FALSE; (*outputs are set to FALSE*)
0005 %QX8.2:=FALSE;
0006 %QX8.3:=FALSE;
0007
0008 Measuring:=FALSE;
0009

Sequence_curved (FB-SFC).Transition Trans6 (ST)
0001 SEQUENCE2INTERRUPTED
0002 AND HOMESEQUENCE2
0003 AND NOT %QX8.2
0004 AND NOT %QX8.3
0005 AND NOT Measuring
0006 AND NOT %QX8.0
0007 AND StopSignal

Sequence_curved (FB-SFC).Action Step6 (ST)
0001 %QX8.0:=FALSE;
0002 %QX8.1:=FALSE;
0003 %QX8.2:=FALSE;
0004 %QX8.3:=FALSE;
0005 (*bits are set to FALSE to avoid errors in ViewScan*)
Sequence_curved (FB-SFC).Transition Trans7 (ST)
0001 StopSignal
0002 AND NOT %QX8.0
0003 AND NOT %QX8.1
0004 AND NOT %QX8.2
0005 AND NOT %QX8.3
0006 AND DC_END1
0007 AND NOT DC_END0
0008 (*if the sensor is in correct position, the sequence continues via this branch*)

Sequence_curved (FB-SFC).Action Step7 - Entry (ST)
0001 RecordNoY:=4; (*servos are operated in record selection mode. This is the number of the record which is executed next*)
0002
0003 (*Additional to a step action you can add an entry action and an exit action to a step.
0004 An entry action is executed only once, right after the step has become active. An exit action is executed only once before the step is deactivated.
0005 A step with entry action is indicated by an "E" in the lower left corner, the exit action by an "X" in the lower right corner.
0006 The entry and exit action can be implemented in any language.*)

Sequence_curved (FB-SFC).Action Step7 (ST)
0001 StartTaskY:=STEP7>(*Rising edge in StartTask trigger the positioning task.*)
0002 (*Step7 is a local variable*)

Sequence_curved (FB-SFC).Transition Trans10 (ST)
0001 ActualRecordNoY = 4 (*Actual record on the controller corresponds to the one set on entry action*)
0002 AND Y_MOTIONCOMPL (*Motion is complete and drive isn't moving*)
0003 AND StopSignal (*Stop signal is present*)
Sequence_curved (FB-SFC).Transition Trans11 (ST)
0001 NOT StopSignal (*Sequence stopped bu user*)

Sequence_curved (FB-SFC).Action Step9 (ST)
0001 SEQUENCE2INTERRUPTED:=TRUE;
0002 HOMESEQUENCE2:=TRUE;
0003 Measuring:=FALSE;
0004 StartTaskY:=FALSE; (*StartTask set to FALSE*)

```

```

Sequence_curved (FB-SFC).Transition Trans13 (ST)
0001 SEQUENCE2INTERRUPTED
0002 AND HOMESEQUENCE2
0003 AND NOT Measuring
0004 AND NOT StartTaskY
0005 AND StopSignal

Sequence_curved (FB-SFC).Transition Trans8 (ST)
0001 StopSignal
0002 AND NOT %QX8.0
0003 AND NOT %QX8.1
0004 AND NOT %QX8.2
0005 AND NOT %QX8.3
0006 AND ((DC_END0
0007 AND NOT DC_END1)
0008 OR (NOT DC_END0
0009 AND NOT DC_END1))
0010 (*Sensor will be turned in correct position, if this branch is chosen. The choice is made with switches monitoring the sensors position*)

Sequence_curved (FB-SFC).Transition Trans9 (ST)
0001 NOT StopSignal
0002 (*Sequence stopped by user*)

Sequence_curved (FB-SFC).Action Step8 (ST)
0001 SEQUENCE2INTERRUPTED:=TRUE;
0002 HOMESEQUENCE2:=TRUE;
0003 Measuring:=FALSE;

Sequence_curved (FB-SFC).Transition Trans12 (ST)
0001 HOMESEQUENCE2
0002 AND SEQUENCE2INTERRUPTED
0003 AND NOT Measuring
0004 AND StopSignal

Sequence_curved (FB-SFC).Action Step10 - Entry (ST)
0001 RecordNoY:=3;

Sequence_curved (FB-SFC).Action Step10 (ST)
0001 StartTaskY:=STEP10;

Sequence_curved (FB-SFC).Transition Trans14 (ST)
0001 ActualRecordNoY = 3
0002 AND Y_MOTIONCOMPL
0003 AND ActualPositionY >=90
0004 AND StopSignal

Sequence_curved (FB-SFC).Transition Trans15 (ST)
0001 NOT StopSignal

Sequence_curved (FB-SFC).Action Step11 (ST)
0001 SEQUENCE2INTERRUPTED:=TRUE;
0002 HOMESEQUENCE2:=TRUE;
0003 Measuring:=FALSE;
0004 StartTaskY:=FALSE;

Sequence_curved (FB-SFC).Transition Trans16 (ST)
0001 SEQUENCE2INTERRUPTED
0002 AND HOMESEQUENCE2
0003 AND NOT Measuring
0004 AND NOT StartTaskY
0005 AND StopSignal

Sequence_curved (FB-SFC).Action Step12 (ST)
0001 StartTaskY:=FALSE; (*just to make sure the starttask is false*)
0002
0003 Sequence_curved.TONInst(In:=DC_END1, PT:=T#500ms); (*Turn on delay. DC motor moves for 500ms after*)
0004 (*the switch is activated*)
0005
0006 DC_FWD:=STEP12
0007 AND NOT Sequence_curved.TONInst.Q
0008 AND StopSignal;
0009
0010 (*If DC_END1 isn't activated in 2s, which is the maximum step time,*)
0011 (*the SFCErrror is set to TRUE and the sequence is stopped.*)
0012 (*Reset and restart are required*)

Sequence_curved (FB-SFC).Transition Trans17 (ST)
0001 StopSignal
0002 AND Sequence_curved.TONInst.Q (*When "turn on" timers output is TRUE, the sequence continues *)

Sequence_curved (FB-SFC).Transition Trans18 (ST)
0001 NOT StopSignal (*Sequence stopped by user*)

```



```

Sequence_curved (FB-SFC).Transition Trans90 (ST)
0001 SFCErrror  (*SFCErrror stops sequence*)

Sequence_curved (FB-SFC).Action Step91 (ST)
0001 Measuring:=FALSE;
0002 DC_FWD:=FALSE;

Sequence_curved (FB-SFC).Action Step13 (ST)
0001 SEQUENCE2INTERRUPTED:=TRUE;
0002 HOMESEQUENCE2:=TRUE;
0003 Measuring:=FALSE;
0004 DC_FWD:=FALSE;  (*Variable controlling the sensor rotation is set to FALSE*)

Sequence_curved (FB-SFC).Transition Trans93 (ST)
0001 SEQUENCE2INTERRUPTED
0002 AND HOMESEQUENCE2
0003 AND StopSignal
0004 AND NOT Measuring
0005 AND NOT DC_FWD

Sequence_curved (FB-SFC).Action Step14 - Entry (ST)
0001 RecordNoY:=4;

Sequence_curved (FB-SFC).Action Step14 (ST)
0001 DC_FWD:=FALSE;
0002 StartTaskY:=STEP14;

Sequence_curved (FB-SFC).Transition Trans20 (ST)
0001 ActualRecordNoY = 4
0002 AND Y_MOTIONCOMPL
0003 AND StopSignal

Sequence_curved (FB-SFC).Transition Trans21 (ST)
0001 NOT StopSignal

Sequence_curved (FB-SFC).Action Step15 (ST)
0001 SEQUENCE2INTERRUPTED:=TRUE;
0002 HOMESEQUENCE2:=TRUE;
0003 Measuring:=FALSE;
0004 StartTaskY:=FALSE;

Sequence_curved (FB-SFC).Transition Trans22 (ST)
0001 SEQUENCE2INTERRUPTED
0002 AND HOMESEQUENCE2
0003 AND NOT Measuring
0004 AND NOT StartTaskY
0005 AND StopSignal

Sequence_curved (FB-SFC).Action Step16 - Entry (ST)
0001 RecordNoZ2:=2;
0002 (*Preselection of record number for record selection*)
0003 (*If you want to change the speed the axis moves, you must edit the position profile in the FCT*)

Sequence_curved (FB-SFC).Action Step16 (ST)
0001 StartTaskZ2:=STEP16;
0002 StartTaskY:=FALSE;

Sequence_curved (FB-SFC).Transition Trans23 (ST)
0001 ActualRecordNoZ2 = 2
0002 AND Z2_MOTIONCOMPL
0003 AND StopSignal

Sequence_curved (FB-SFC).Transition Trans24 (ST)
0001 NOT StopSignal

Sequence_curved (FB-SFC).Action Step17 (ST)
0001 SEQUENCE2INTERRUPTED:=TRUE;
0002 HOMESEQUENCE2:=TRUE;
0003 Measuring:=FALSE;
0004 StartTaskZ2:=FALSE;

Sequence_curved (FB-SFC).Transition Trans25 (ST)
0001 SEQUENCE2INTERRUPTED
0002 AND HOMESEQUENCE2
0003 AND NOT Measuring
0004 AND NOT StartTaskZ2
0005 AND StopSignal

```

```

Sequence_curved (FB-SFC).Action Step18 (ST)
0001 StartTaskZ2:=FALSE;

Sequence_curved (FB-SFC).Transition Trans26 (ST)
0001 NOT StartTaskZ2
0002 AND StopSignal

Sequence_curved (FB-SFC).Transition Trans27 (ST)
0001 NOT StopSignal

Sequence_curved (FB-SFC).Action Step20 (ST)
0001 SEQUENCE2INTERRUPTED:=TRUE;
0002 HOMESEQUENCE2:=TRUE;
0003 Measuring:=FALSE;

Sequence_curved (FB-SFC).Transition Trans30 (ST)
0001 HOMESEQUENCE2
0002 AND SEQUENCE2INTERRUPTED
0003 AND NOT Measuring
0004 AND StopSignal

Sequence_curved (FB-SFC).Action Step19 - Entry (ST)
0001 RecordNoZ2:=3;

Sequence_curved (FB-SFC).Action Step19 (ST)
0001 StartTaskZ2:=STEP19;
0002 Activate_K4:=TRUE; (*Contactor K4 activated*)

Sequence_curved (FB-SFC).Transition Trans28 (ST)
0001 ActualRecordNoZ2 = 3
0002 AND Z2_MOTIONCOMPL
0003 AND StopSignal
0004 AND K4_OK

Sequence_curved (FB-SFC).Transition Trans29 (ST)
0001 NOT StopSignal

Sequence_curved (FB-SFC).Action Step22 (ST)
0001 SEQUENCE2INTERRUPTED:=TRUE;
0002 HOMESEQUENCE2:=TRUE;
0003 Measuring:=FALSE;
0004 Activate_K4:=FALSE;
0005 StartTaskZ2:=FALSE;

Sequence_curved (FB-SFC).Transition Trans33 (ST)
0001 HOMESEQUENCE2
0002 AND SEQUENCE2INTERRUPTED
0003 AND NOT Measuring
0004 AND NOT K4_OK
0005 AND StopSignal
0006 AND NOT StartTaskZ2

Sequence_curved (FB-SFC).Action Step21 - Entry (ST)
0001 SetPosRot:= Rotation_CTRL.ActualPosition + 4;
0002 SetVeloRot:=41; (* Rotation speed is X% from the maximum rotation speed -> 0.41 * 46rpm = ca. 19rpm *)

Sequence_curved (FB-SFC).Action Step21 (ST)
0001 StartTaskZ2:=FALSE;
0002 StartTaskRot:=STEP21;
0003 (*Signal sent to RollScan and magnetizing ramp starts. DEMAGN is set to true when rotation starts*)
0004 IF AckStartRot
0005 AND K4_OK
0006 THEN DEMAGN:=TRUE; (*Rising edge triggers magnetizing ramp in rollscan*)
0007 END_IF;

0001 DEMAGN
0002 AND ROT_MOTIONCOMPL
0003 AND StopSignal
0004 (*Sequence will continue when motion is complete*)

Sequence_curved (FB-SFC).Transition Trans32 (ST)
0001 NOT StopSignal

```

Sequence\_curved (FB-SFC),Action Step23 (ST)

```

0001 SEQUENCE2INTERRUPTED:=TRUE;
0002 HOMESEQUENCE2:=TRUE;
0003 Measuring:=FALSE;
0004 Activate_K4:=FALSE;
0005 StartTaskRot:=FALSE;

```

Sequence\_curved (FB-SFC),Transition Trans34 (ST)

```

0001 HOMESEQUENCE2
0002 AND SEQUENCE2INTERRUPTED
0003 AND NOT Measuring
0004 AND StopSignal
0005 AND NOT K4_OK

```

Sequence\_curved (FB-SFC),Action Step24 - Entry (ST)

```
0001 RecordNoZ:=4;
```

Sequence\_curved (FB-SFC),Action Step24 (ST)

```

0001 StartTaskRot:=FALSE;
0002 StartTaskZ:=STEP24;

```

Sequence\_curved (FB-SFC),Transition Trans35 (ST)

```

0001 ActualRecordNoZ = 4
0002 AND Z_MOTIONCOMPL
0003 AND StopSignal

```

Sequence\_curved (FB-SFC),Transition Trans36 (ST)

```
0001 NOT StopSignal
```

Sequence\_curved (FB-SFC),Action Step25 (ST)

```

0001 SEQUENCE2INTERRUPTED:=TRUE;
0002 HOMESEQUENCE2:=TRUE;
0003 Measuring:=FALSE;
0004 Activate_K4:=FALSE;
0005 StartTaskZ:=FALSE;
0006
0007

```

Sequence\_curved (FB-SFC),Transition Trans37 (ST)

```

0001 HOMESEQUENCE2
0002 AND SEQUENCE2INTERRUPTED
0003 AND NOT Measuring
0004 AND StopSignal
0005 AND NOT K4_OK
0006 AND NOT StartTaskZ

```

Sequence\_curved (FB-SFC),Action Step26 - Entry (ST)

```
0001 RecordNoY:=8;
```

Sequence\_curved (FB-SFC),Action Step26 (ST)

```

0001 StartTaskZ:=FALSE;
0002 StartTaskY:=STEP26;

```

Sequence\_curved (FB-SFC),Transition Trans38 (ST)

```

0001 ActualRecordNoY = 8
0002 AND Y_MOTIONCOMPL
0003 AND StopSignal
0004 AND NOT StartTaskZ

```

Sequence\_curved (FB-SFC),Transition Trans39 (ST)

```
0001 NOT StopSignal
```

```

0001 SEQUENCE2INTERRUPTED:=TRUE;
0002 HOMESEQUENCE2:=TRUE;
0003 Measuring:=FALSE;
0004 Activate_K4:=FALSE;
0005 StartTaskY:=FALSE;

```

Sequence\_curved (FB-SFC),Transition Trans40 (ST)

```

0001 HOMESEQUENCE2
0002 AND SEQUENCE2INTERRUPTED
0003 AND NOT Measuring
0004 AND StopSignal
0005 AND NOT K4_OK
0006 AND NOT StartTaskY

```

Sequence\_curved (FB-SFC).Action Step28 - Entry (ST)

0001 RecordNoZ:=5;

Sequence\_curved (FB-SFC).Action Step28 (ST)

0001 StartTaskY:=FALSE;

0002 StartTaskZ:=STEP28;

Sequence\_curved (FB-SFC).Transition Trans41 (ST)

0001 ActualRecordNoZ = 5

0002 AND Z\_MOTIONCOMPL

0003 AND StopSignal

Sequence\_curved (FB-SFC).Transition Trans42 (ST)

0001 NOT StopSignal

Sequence\_curved (FB-SFC).Action Step29 (ST)

0001 SEQUENCE2INTERRUPTED:=TRUE;

0002 HOMESEQUENCE2:=TRUE;

0003 Measuring:=FALSE;

0004 Activate\_K4:=FALSE;

0005 StartTaskZ:=FALSE;

Sequence\_curved (FB-SFC).Transition Trans43 (ST)

0001 HOMESEQUENCE2

0002 AND SEQUENCE2INTERRUPTED

0003 AND NOT Measuring

0004 AND StopSignal

0005 AND NOT K4\_OK

0006 AND NOT StartTaskZ

Sequence\_curved (FB-SFC).Action Step30 - Entry (ST)

0001 RecordNoY:=5;

Sequence\_curved (FB-SFC).Action Step30 (ST)

0001 StartTaskZ:=FALSE;

0002 StartTaskY:=STEP30;

Sequence\_curved (FB-SFC).Transition Trans44 (ST)

0001 ActualRecordNoY = 5

0002 AND Y\_MOTIONCOMPL

0003 AND StopSignal

Sequence\_curved (FB-SFC).Transition Trans45 (ST)

0001 NOT StopSignal

Sequence\_curved (FB-SFC).Action Step31 (ST)

0001 SEQUENCE2INTERRUPTED:=TRUE;

0002 HOMESEQUENCE2:=TRUE;

0003 Measuring:=FALSE;

0004 Activate\_K4:=FALSE;

0005 StartTaskY:=FALSE;

Sequence\_curved (FB-SFC).Transition Trans46 (ST)

0001 HOMESEQUENCE2

0002 AND SEQUENCE2INTERRUPTED

0003 AND NOT Measuring

0004 AND StopSignal

0005 AND NOT K4\_OK

0006 AND NOT StartTaskY

Sequence\_curved (FB-SFC).Action Step32 - Entry (ST)

0001 SetPosRot:=ActualPositionRot + 100;

Sequence\_curved (FB-SFC).Action Step32 (ST)

0001 StartTaskY:=FALSE;

0002 StartTaskRot:=STEP32;

Sequence\_curved (FB-SFC).Transition Trans47 (ST)

0001 Rotation\_CTRL.ActualVelocity = 41

0002 AND StopSignal

Sequence\_curved (FB-SFC).Transition Trans48 (ST)

0001 NOT StopSignal

Sequence\_curved (FB-SFC).Action Step33 (ST)

0001 SEQUENCE2INTERRUPTED:=TRUE;

0002 HOMESEQUENCE2:=TRUE;

0003 Measuring:=FALSE;

0004 Activate\_K4:=FALSE;

0005 StartTaskRot:=FALSE;

0006 %QX8.1:=TRUE;

Sequence\_curved (FB-SFC).Transition Trans49 (ST)

```
0001 HOMESEQUENCE2
0002 AND SEQUENCE2INTERRUPTED
0003 AND NOT Measuring
0004 AND StopSignal
0005 AND NOT K4_OK
0006 AND NOT StartTaskRot
```

Sequence\_curved (FB-SFC).Action Step92 (ST)

```
0001 %QX8.1:=FALSE;
```

Sequence\_curved (FB-SFC).Action Step34 - Entry (ST)

```
0001 RecordNoZ:=6;
```

Sequence\_curved (FB-SFC).Action Step34 (ST)

```
0001 StartTaskRot:=FALSE;
0002 Activate_RSTRIG:=STEP34;
0003 StartTaskZ:=STEP34;
```

Sequence\_curved (FB-SFC).Transition Trans50 (ST)

```
0001 Zaxis_CTRL.DrivelsMoving AND
0002 StopSignal AND
0003 ActualPositionZ = 68
```

Sequence\_curved (FB-SFC).Transition Trans51 (ST)

```
0001 NOT StopSignal
```

Sequence\_curved (FB-SFC).Action Step35 (ST)

```
0001 SEQUENCE2INTERRUPTED:=TRUE;
0002 HOMESEQUENCE2:=TRUE;
0003 Measuring:=FALSE;
0004 Activate_K4:=FALSE;
0005 Activate_RSTRIG:=FALSE;
0006 StartTaskZ:=FALSE;
0007 %QX8.1:=TRUE;
```

Sequence\_curved (FB-SFC).Transition Trans52 (ST)

```
0001 SEQUENCE2INTERRUPTED
0002 AND HOMESEQUENCE2
0003 AND NOT Measuring
0004 AND NOT K4_OK
0005 AND NOT Activate_RSTRIG
0006 AND NOT StartTaskZ
```

Sequence\_curved (FB-SFC).Action Step93 (ST)

```
0001 %QX8.1:=FALSE;
```

Sequence\_curved (FB-SFC).Action Step36 - Entry (ST)

```
0001 RecordNoY:=6;
```

Sequence\_curved (FB-SFC).Action Step36 (ST)

```
0001 StartTaskZ:=FALSE;
0002 StartTaskY:=STEP36;
```

Sequence\_curved (FB-SFC).Transition Trans53 (ST)

```
0001 Zaxis_CTRL.DrivelsMoving
0002 AND ActualRecordNoY = 6
0003 AND Y_MOTIONCOMPL
0004 AND StopSignal
```

Sequence\_curved (FB-SFC).Action Step37 (ST)

```
0001 StartTaskY:=FALSE;
```

Sequence\_curved (FB-SFC).Transition Trans55 (ST)

```
0001 Zaxis_CTRL.DrivelsMoving
0002 AND ActualPositionZ = 62
0003 AND Y_MOTIONCOMPL
0004 AND StopSignal
```

Sequence\_curved (FB-SFC).Transition Trans56 (ST)

```
0001 NOT StopSignal
```

Sequence\_curved (FB-SFC).Action Step39 (ST)

```

0001 SEQUENCE2INTERRUPTED:=TRUE;
0002 HOMESEQUENCE2:=TRUE;
0003 Measuring:=FALSE;
0004 Activate_K4:=FALSE;
0005 Activate_RSTRIG:=FALSE;
0006 %QX8.1:=TRUE;
0007

```

Sequence\_curved (FB-SFC).Transition Trans58 (ST)

```

0001 SEQUENCE2INTERRUPTED
0002 AND HOMESEQUENCE2
0003 AND NOT Measuring
0004 AND NOT K4_OK
0005 AND NOT Activate_RSTRIG
0006 AND NOT StartTaskY
0007 AND StopSignal

```

Sequence\_curved (FB-SFC).Action Step95 (ST)

```

0001 %QX8.1:=FALSE;

```

Sequence\_curved (FB-SFC).Transition Trans54 (ST)

```

0001 NOT StopSignal

```

Sequence\_curved (FB-SFC).Action Step38 (ST)

```

0001 SEQUENCE2INTERRUPTED:=TRUE;
0002 HOMESEQUENCE2:=TRUE;
0003 Measuring:=FALSE;
0004 Activate_K4:=FALSE;
0005 Activate_RSTRIG:=FALSE;
0006 StartTaskY:=FALSE;
0007 %QX8.1:=TRUE;

```

Sequence\_curved (FB-SFC).Transition Trans57 (ST)

```

0001 SEQUENCE2INTERRUPTED
0002 AND HOMESEQUENCE2
0003 AND NOT Measuring
0004 AND NOT K4_OK
0005 AND NOT Activate_RSTRIG
0006 AND NOT StartTaskY
0007 AND StopSignal

```

Sequence\_curved (FB-SFC).Action Step94 (ST)

```

0001 %QX8.1:=FALSE;
0001 RecordNoY:=7;

```

Sequence\_curved (FB-SFC).Action Step40 (ST)

```

0001 StartTaskY:=STEP40;

```

Sequence\_curved (FB-SFC).Transition Trans59 (ST)

```

0001 ActualRecordNoY = 7
0002 AND Y_MOTIONCOMPL
0003 AND Z_MOTIONCOMPL
0004 AND StopSignal

```

Sequence\_curved (FB-SFC).Transition Trans60 (ST)

```

0001 NOT StopSignal

```

Sequence\_curved (FB-SFC).Action Step41 (ST)

```

0001 SEQUENCE2INTERRUPTED:=TRUE;
0002 HOMESEQUENCE2:=TRUE;
0003 Measuring:=FALSE;
0004 Activate_K4:=FALSE;
0005 Activate_RSTRIG:=FALSE;
0006 StartTaskY:=FALSE;
0007 %QX8.1:=TRUE;

```

```

Sequence_curved (FB-SFC).Transition Trans61 (ST)
0001 SEQUENCE2INTERRUPTED
0002 AND HOMESEQUENCE2
0003 AND NOT Measuring
0004 AND NOT K4_OK
0005 AND NOT Activate_RSTRIG
0006 AND NOT StartTaskY
0007 AND StopSignal

Sequence_curved (FB-SFC).Action Step96 (ST)
0001 %QX8.1:=FALSE;

Sequence_curved (FB-SFC).Action Step42 (ST)
0001 StartTaskY:=FALSE;
0002 Activate_RSTRIG:=FALSE;
0003 HALTROT:=TRUE;
0004 %QX8.1:=TRUE; (*END bit to ViewScan*)
0005 DEMAGN:=FALSE;

Sequence_curved (FB-SFC).Transition Trans62 (ST)
0001 NOT Rotation_CTRL.DrivesMoving
0002 AND NOT DEMAGN
0003 AND StopSignal

Sequence_curved (FB-SFC).Transition Trans63 (ST)
0001 NOT StopSignal

Sequence_curved (FB-SFC).Action Step43 (ST)
0001 SEQUENCE2INTERRUPTED:=TRUE;
0002 HOMESEQUENCE2:=TRUE;
0003 Measuring:=FALSE;
0004 Activate_K4:=FALSE;
0005 Activate_RSTRIG:=FALSE;
0006 Rotation_CTRL.ClearRemainingPosition:=TRUE;
0007 %QX8.1:=FALSE;

Sequence_curved (FB-SFC).Action Step44 (ST)
0001 Rotation_CTRL.ClearRemainingPosition:=FALSE;
0002 HALTROT:=FALSE;

Sequence_curved (FB-SFC).Transition Trans64 (ST)
0001 SEQUENCEINTERRUPTED
0002 AND HOMESEQUENCE1
0003 AND NOT Measuring
0004 AND NOT K4_OK
0005 AND NOT Activate_RSTRIG
0006 AND StopSignal

Sequence_curved (FB-SFC).Action Step45 - Entry (ST)
0001 RecordNoY:=4;

Sequence_curved (FB-SFC).Action Step45 (ST)
0001 (*Activate_K4:=FALSE;*)
0002 StartTaskY:=STEP45;
0003 Rotation_CTRL.ClearRemainingPosition:=TRUE;
0004 %QX8.1:=FALSE;

Sequence_curved (FB-SFC).Transition Trans65 (ST)
0001 ActualRecordNoY = 4
0002 AND Y_MOTIONCOMPL
0003 AND StopSignal

Sequence_curved (FB-SFC).Transition Trans66 (ST)
0001 NOT StopSignal

Sequence_curved (FB-SFC).Action Step46 (ST)
0001 SEQUENCE2INTERRUPTED:=TRUE;
0002 HOMESEQUENCE2:=TRUE;
0003 Measuring:=FALSE;
0004 StartTaskY:=FALSE;
0005 HALTROT:=FALSE;
0006 Rotation_CTRL.ClearRemainingPosition:=FALSE;

```

Sequence\_curved (FB-SFC).Transition Trans67 (ST)

0001 HOMESEQUENCE2  
 0002 AND SEQUENCE2INTERRUPTED  
 0003 AND NOT Measuring  
 0004 AND StopSignal  
 0005 AND NOT StartTaskY

Sequence\_curved (FB-SFC).Action Step47 - Entry (ST)

0001 RecordNoZ:=1;

Sequence\_curved (FB-SFC).Action Step47 (ST)

0001 StartTaskY:=FALSE;  
 0002 StartTaskZ:=STEP47;  
 0003 HALTROT:=FALSE;  
 0004 Rotation\_CTRL.ClearRemainingPosition:=FALSE;

Sequence\_curved (FB-SFC).Transition Trans68 (ST)

0001 Z\_MOTIONCOMPL  
 0002 AND ActualRecordNoZ = 1  
 0003 AND StopSignal

Sequence\_curved (FB-SFC).Action Step48 - Entry (ST)

0001 RecordNoY:=1;

Sequence\_curved (FB-SFC).Action Step48 (ST)

0001 StartTaskY:=STEP48;  
 0002 StartTaskZ:=FALSE;

Sequence\_curved (FB-SFC).Transition Trans70 (ST)

0001 Y\_MOTIONCOMPL  
 0002 AND ActualRecordNoY = 1  
 0003 AND StopSignal

Sequence\_curved (FB-SFC).Transition Trans71 (ST)

0001 NOT StopSignal

Sequence\_curved (FB-SFC).Action Step50 (ST)

0001 SEQUENCE2INTERRUPTED:=TRUE;  
 0002 HOMESEQUENCE2:=TRUE;  
 0003 Measuring:=FALSE;  
 0004 StartTaskY:=FALSE;

Sequence\_curved (FB-SFC).Transition Trans73 (ST)

0001 HOMESEQUENCE2  
 0002 AND SEQUENCE2INTERRUPTED  
 0003 AND NOT Measuring  
 0004 AND StopSignal  
 0005 AND NOT StartTaskY

Sequence\_curved (FB-SFC).Transition Trans69 (ST)

0001 NOT StopSignal

Sequence\_curved (FB-SFC).Action Step49 (ST)

0001 SEQUENCE2INTERRUPTED:=TRUE;  
 0002 HOMESEQUENCE2:=TRUE;  
 0003 Measuring:=FALSE;  
 0004 StartTaskZ:=FALSE;

Sequence\_curved (FB-SFC).Transition Trans72 (ST)

0001 HOMESEQUENCE2  
 0002 AND SEQUENCE2INTERRUPTED  
 0003 AND NOT Measuring  
 0004 AND StopSignal  
 0005 AND NOT StartTaskZ

Sequence\_curved (FB-SFC).Action Step55 (ST)

0001 StartTaskY:=FALSE;  
 0002 Activate\_K4:=FALSE;



Sequence\_curved (FB-SFC).Transition Trans80 (ST)

0001 NOT K4\_OK  
0002 AND StopSignal

Sequence\_curved (FB-SFC).Transition Trans81 (ST)

0001 NOT StopSignal

Sequence\_curved (FB-SFC).Action Step56 (ST)

0001 SEQUENCE2INTERRUPTED:=TRUE;  
0002 HOMESEQUENCE2:=TRUE;  
0003 Measuring:=FALSE;

Sequence\_curved (FB-SFC).Transition Trans82 (ST)

0001 HOMESEQUENCE2  
0002 AND SEQUENCE2INTERRUPTED  
0003 AND NOT Measuring  
0004 AND NOT K4\_OK  
0005 AND StopSignal

Sequence\_curved (FB-SFC).Action Step57 - Entry (ST)

0001 RecordNoZ2:=1;

Sequence\_curved (FB-SFC).Action Step57 (ST)

0001 StartTaskZ2:=STEP57;

Sequence\_curved (FB-SFC).Transition Trans83 (ST)

0001 Z2\_MOTIONCOMPL  
0002 AND ActualRecordNoZ2 = 1  
0003 AND StopSignal

Sequence\_curved (FB-SFC).Transition Trans84 (ST)

0001 NOT StopSignal

Sequence\_curved (FB-SFC).Action Step58 (ST)

0001 SEQUENCE2INTERRUPTED:=TRUE;  
0002 HOMESEQUENCE2:=TRUE;  
0003 Measuring:=FALSE;  
0004 StartTaskZ2:=FALSE;

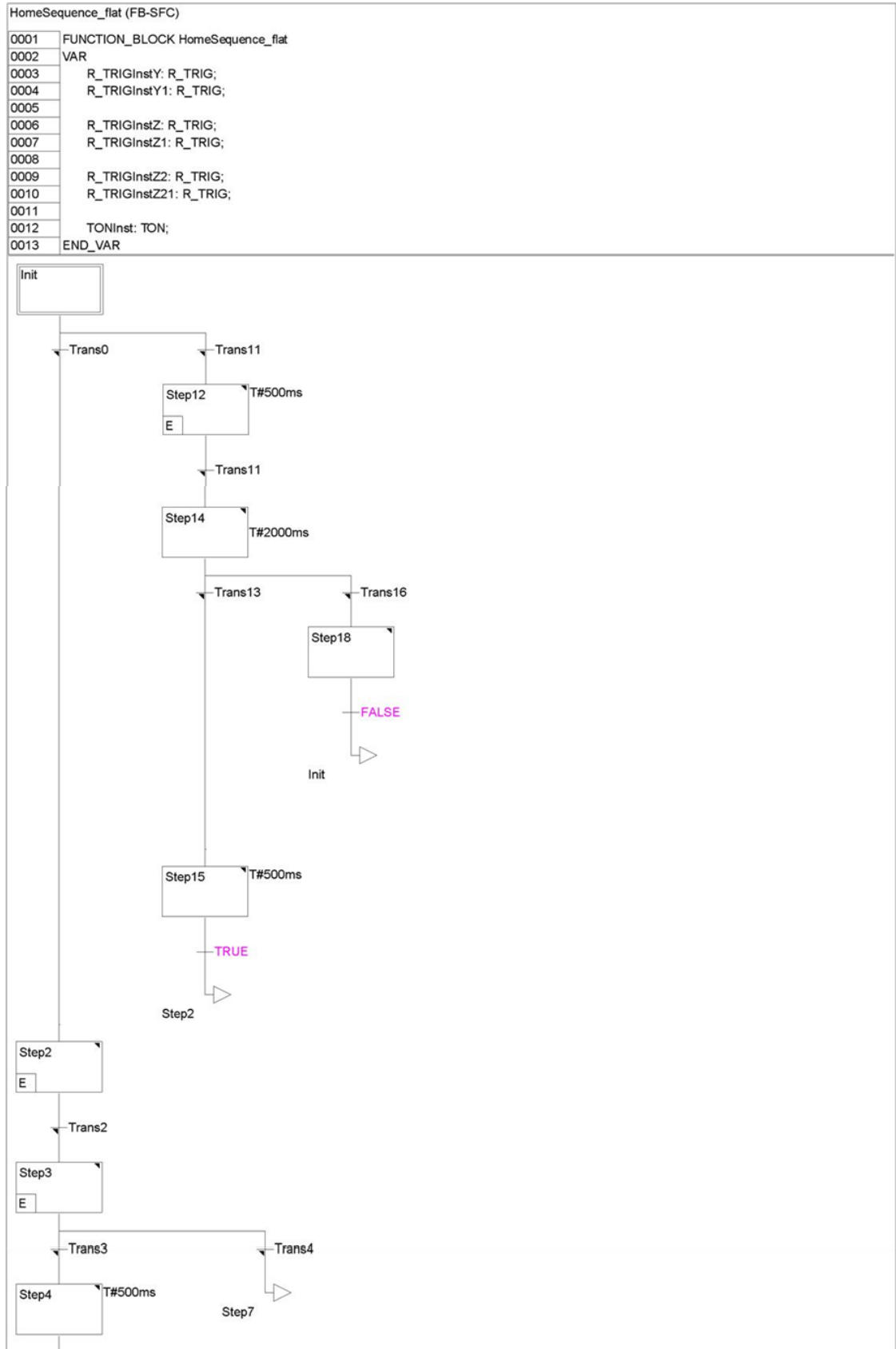
Sequence\_curved (FB-SFC).Transition Trans85 (ST)

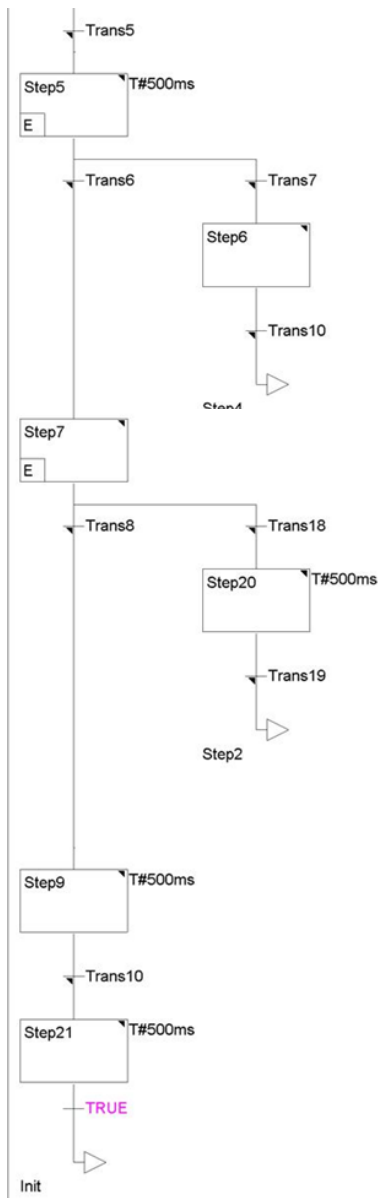
0001 SEQUENCE2INTERRUPTED  
0002 AND HOMESEQUENCE2  
0003 AND NOT Measuring  
0004 AND StopSignal  
0001 StartTaskZ2:=FALSE;  
0002 Measuring:=FALSE;

Sequence\_curved (FB-SFC).Transition Trans86 (ST)

0001 NOT Measuring  
0002 AND StopSignal

# Liite 16. HOMESEQUENCE FLAT





HomeSequence\_flat (FB-SFC).Transition Trans0 (ST)

```

0001 ((SEQUENCEINTERRUPTED
0002 AND HOMESEQUENCE1)
0003 OR JOG_activated)
0004 AND HOME_BUTTON
0005 AND StopSignal
0006 AND DrivelsReferencedRot
0007 AND DrivelsReferencedY
0008 AND DrivelsReferencedZ
0009 AND DrivelsReferencedZ2
0010 AND (TML1_SEL
0011 OR REF1_SEL)
0012 AND (DC_END0
0013 OR DC_END1)
0014 AND (TML1_SEL
0015 OR REF1_SEL)
    
```

HomeSequence\_flat (FB-SFC).Transition Trans11 (ST)

```

0001 ((SEQUENCEINTERRUPTED
0002 AND HOMESEQUENCE1)
0003 OR JOG_activated)
0004 AND HOME_BUTTON
0005 AND StopSignal
0006 AND DrivesReferencedRot
0007 AND DrivesReferencedY
0008 AND DrivesReferencedZ
0009 AND DrivesReferencedZ2
0010 AND (TML1_SEL
0011 OR REF1_SEL)
0012 AND (NOT DC_END0
0013 AND NOT DC_END1)
0014 AND (TML1_SEL
0015 OR REF1_SEL)

```

HomeSequence\_flat (FB-SFC).Action Step12 - Entry (ST)

```
0001 RecordNoY:=3;
```

HomeSequence\_flat (FB-SFC).Action Step12 (ST)

```

0001 HomeSequence_flat.R_TRIGInstY(CLK:=HOME_BUTTON);
0002 HomeSequence_flat.R_TRIGInstY1(CLK:=STEP12);
0003 StartTaskY:=HomeSequence_flat.R_TRIGInstY.Q OR HomeSequence_flat.R_TRIGInstY1.Q;

```

HomeSequence\_flat (FB-SFC).Transition Trans11 (ST)

```

0001 ActualPositionY = 90
0002 AND Y_MOTIONCOMPL
0003 AND StopSignal

```

HomeSequence\_flat (FB-SFC).Action Step14 (ST)

```

0001 StartTaskY:=FALSE;
0002 HomeSequence_flat.TONInst(In:=DC_END0, PT:=T#500ms);
0003
0004 DC_RWD:=STEP14
0005 AND NOT HomeSequence_flat.TONInst.Q
0006 AND StopSignal;
0007
0008 StartTaskY:=FALSE;

```

HomeSequence\_flat (FB-SFC).Transition Trans13 (ST)

```

0001 StopSignal
0002 AND HomeSequence_flat.TONInst.Q
0003 AND DC_END0

```

HomeSequence\_flat (FB-SFC).Transition Trans16 (ST)

```
0001 SFCErrror
```

HomeSequence\_flat (FB-SFC).Action Step18 (ST)

```

0001 DC_RWD:=FALSE;
0002 StopSignal:=FALSE;

```

HomeSequence\_flat (FB-SFC).Action Step15 (ST)

```

0001 DC_RWD:=FALSE;
0002 Reset_JogActivated:=FALSE;

```

HomeSequence\_flat (FB-SFC).Action Step2 - Entry (ST)

```
0001 RecordNoY:=1;
```

HomeSequence\_flat (FB-SFC).Action Step2 (ST)

```
0001 DC_RWD:=FALSE;
```

HomeSequence\_flat (FB-SFC).Transition Trans2 (ST)

```

0001 ActualPositionY = 0
0002 AND Y_MOTIONCOMPL
0003 AND StopSignal

```

```

HomeSequence_flat (FB-SFC).Action Step3 - Entry (ST)
0001 RecordNoZ:=1;

HomeSequence_flat (FB-SFC).Action Step3 (ST)
0001 HomeSequence_flat.R_TRIGInstZ(CLK:=HOME_BUTTON);
0002 HomeSequence_flat.R_TRIGInstZ1(CLK:=STEP3);
0003
0004 StartTaskZ:=HomeSequence_flat.R_TRIGInstZ.Q
0005 OR HomeSequence_flat.R_TRIGInstZ1.Q;
0006
0007 StartTaskY:=FALSE;

HomeSequence_flat (FB-SFC).Transition Trans3 (ST)
0001 ActualPositionZ = 0
0002 AND Z_MOTIONCOMPL
0003 AND StopSignal
0004 AND DEMAGN

HomeSequence_flat (FB-SFC).Action Step4 (ST)
0001 Activate_K4:=TRUE;
0002 StartTaskZ:=FALSE;

HomeSequence_flat (FB-SFC).Transition Trans5 (ST)
0001 K4_OK
0002 AND StopSignal

HomeSequence_flat (FB-SFC).Transition Trans4 (ST)
0001 ActualPositionZ = 0
0002 AND Z_MOTIONCOMPL
0003 AND StopSignal
0004 AND NOT DEMAGN

HomeSequence_flat (FB-SFC).Action Step5 - Entry (ST)
0001 SetPosRot:=ActualPositionRot + 4;
0002 SetVeloRot:=41;

HomeSequence_flat (FB-SFC).Action Step5 (ST)
0001 StartTaskRot:=Step5;
0002
0003 IF AckStartRot
0004 AND K4_OK
0005 THEN DEMAGN:=FALSE;
0006 END_IF;

HomeSequence_flat (FB-SFC).Transition Trans6 (ST)
0001 NOT DEMAGN
0002 AND ROT_MOTIONCOMPL
0003 AND NOT Rotation_CTRL.DrivesMoving
0004 AND StopSignal

HomeSequence_flat (FB-SFC).Transition Trans7 (ST)
0001 NOT StopSignal

HomeSequence_flat (FB-SFC).Action Step6 (ST)
0001 StartTaskRot:=FALSE;
0002 Activate_K4:=FALSE;

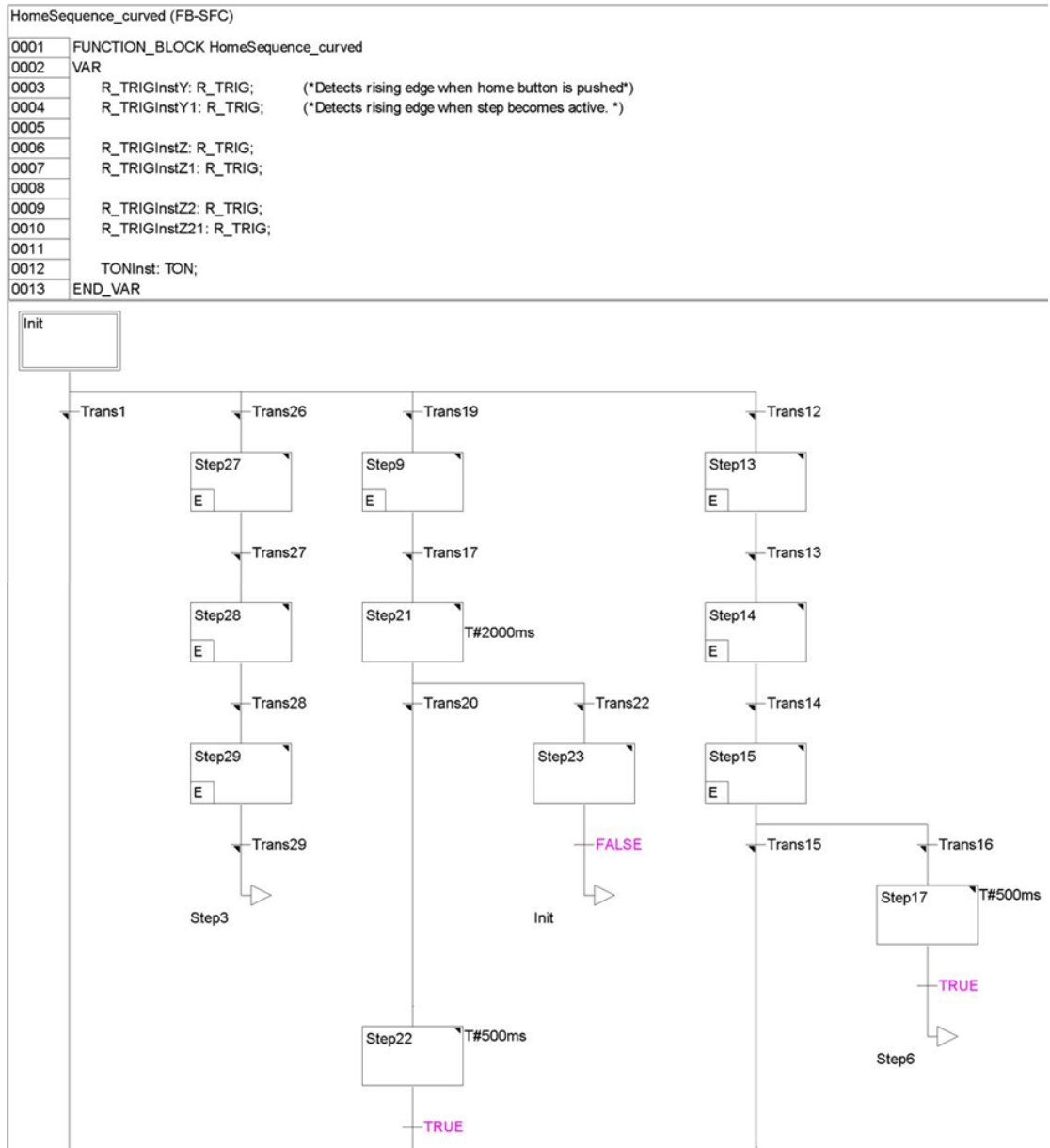
HomeSequence_flat (FB-SFC).Transition Trans10 (ST)
0001 HOME_BUTTON
0002 AND StopSignal

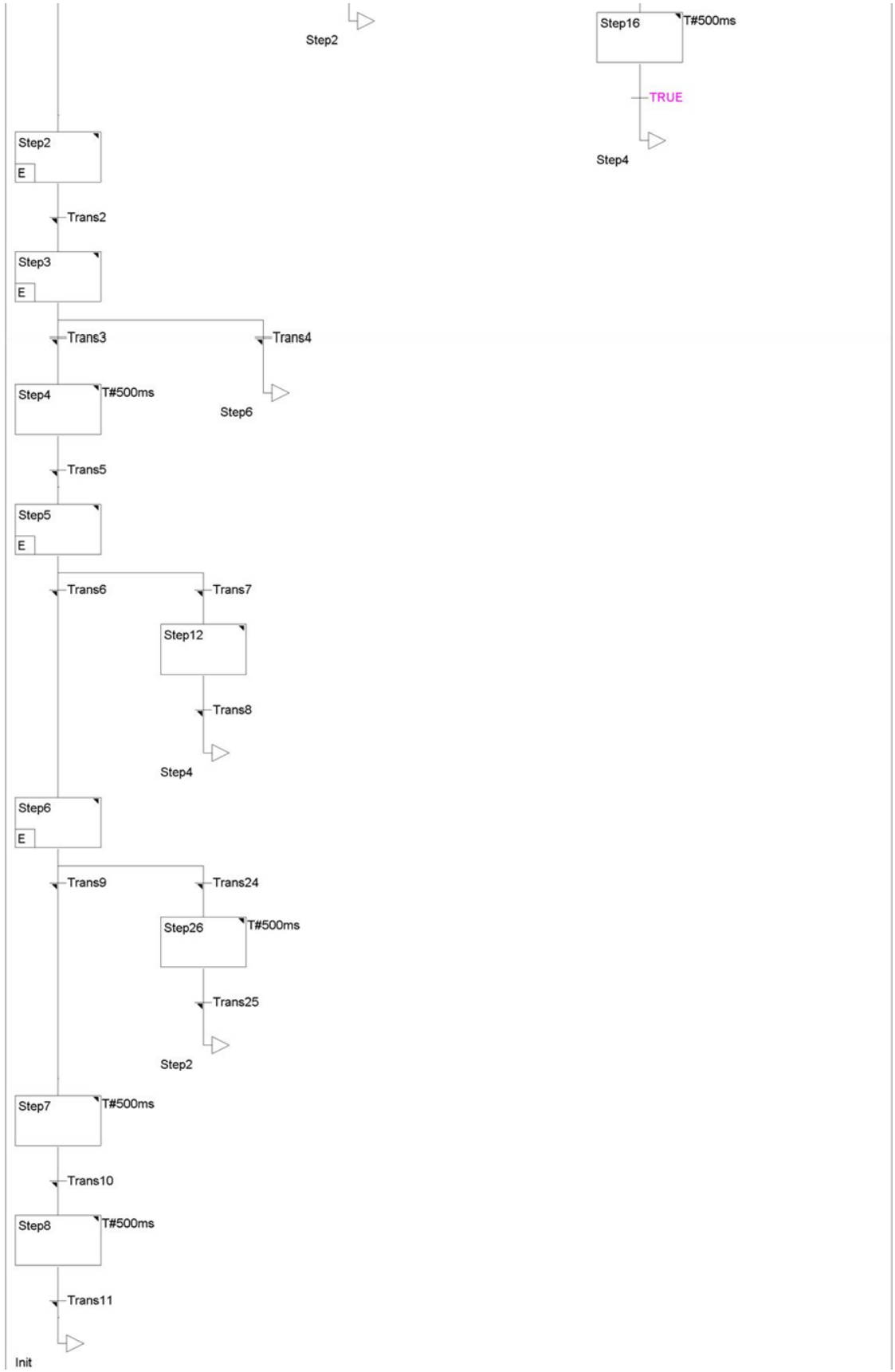
HomeSequence_flat (FB-SFC).Action Step7 - Entry (ST)
0001 RecordNoZ2:=1;

HomeSequence_flat (FB-SFC).Action Step7 (ST)
0001 HomeSequence_flat.R_TRIGInstZ2(CLK:=HOME_BUTTON);
0002 HomeSequence_flat.R_TRIGInstZ21(CLK:=STEP7);
0003
0004 StartTaskZ2:=HomeSequence_flat.R_TRIGInstZ2.Q
0005 OR HomeSequence_flat.R_TRIGInstZ21.Q;
0006
0007 Activate_K4:=FALSE;
0008 StartTaskRot:=FALSE;
0009 StartTaskZ:=FALSE;

```

# Liite 17. HOMESEQUENCE CURVED





HomeSequence\_curved (FB-SFC).Transition Trans1 (ST)

```

0001 ((SEQUENCE2INTERRUPTED
0002 AND HOMESEQUENCE2)
0003 OR JOG_activated)
0004 AND HOME_BUTTON
0005 AND StopSignal
0006 AND DrivelsReferencedRot AND DrivelsReferencedY
0007 AND DrivelsReferencedZ AND DrivelsReferencedZ2
0008 AND (TML2_SEL OR REF2_SEL)
0009 AND (ActualPositionY > 10
0010 AND ActualPositionY < 50)
0011 AND (DC_END0
0012 OR DC_END1)
0013 AND (TML2_SEL
0014 OR REF2_SEL)
0015

```

HomeSequence\_curved (FB-SFC).Action Step28 - Entry (ST)

```
0001 RecordNoZ:=1;
```

HomeSequence\_curved (FB-SFC).Action Step28 (ST)

```

0001 HomeSequence_curved.R_TRIGInstZ(CLK:=HOME_BUTTON);
0002 HomeSequence_curved.R_TRIGInstZ1(CLK:=STEP28);
0003
0004 StartTaskZ:=HomeSequence_curved.R_TRIGInstZ.Q
0005 OR HomeSequence_curved.R_TRIGInstZ1.Q;
0006
0007 StartTaskY:=FALSE;

```

HomeSequence\_curved (FB-SFC).Transition Trans28 (ST)

```

0001 ActualPositionZ = 0
0002 AND Z_MOTIONCOMPL
0003 AND StopSignal

```

HomeSequence\_curved (FB-SFC).Action Step29 - Entry (ST)

```
0001 RecordNoY:=1;
```

HomeSequence\_curved (FB-SFC).Action Step29 (ST)

```

0001 HomeSequence_curved.R_TRIGInstY(CLK:=HOME_BUTTON);
0002 HomeSequence_curved.R_TRIGInstY1(CLK:=STEP29);
0003
0004 StartTaskY:=HomeSequence_curved.R_TRIGInstY.Q
0005 OR HomeSequence_curved.R_TRIGInstY1.Q;
0006
0007 StartTaskZ:=FALSE;

```

HomeSequence\_curved (FB-SFC).Transition Trans26 (ST)

```

0001 ((SEQUENCE2INTERRUPTED
0002 AND HOMESEQUENCE2)
0003 OR JOG_activated)
0004 AND HOME_BUTTON
0005 AND StopSignal
0006 AND DrivelsReferencedRot AND DrivelsReferencedY
0007 AND DrivelsReferencedZ AND DrivelsReferencedZ2
0008 AND (TML2_SEL OR REF2_SEL)
0009 AND ActualPositionY < 10
0010 AND DC_END1
0011 AND (TML2_SEL
0012 OR REF2_SEL)

```

HomeSequence\_curved (FB-SFC).Action Step27 - Entry (ST)

```
0001 RecordNoY:=10;
```

HomeSequence\_curved (FB-SFC).Action Step27 (ST)

```

0001 HomeSequence_curved.R_TRIGInstY(CLK:=HOME_BUTTON);
0002 HomeSequence_curved.R_TRIGInstY1(CLK:=STEP27);
0003
0004 StartTaskY:=HomeSequence_curved.R_TRIGInstY.Q
0005 OR HomeSequence_curved.R_TRIGInstY1.Q;

```

HomeSequence\_curved (FB-SFC).Transition Trans27 (ST)

```

0001 ActualPositionY = 15
0002 AND Y_MOTIONCOMPL
0003 AND StopSignal

```

HomeSequence\_curved (FB-SFC).Transition Trans29 (ST)

```

0001 ActualPositionY = 0
0002 AND Y_MOTIONCOMPL

```



```

0003 AND StopSignal

HomeSequence_curved (FB-SFC).Transition Trans19 (ST)
0001 ((SEQUENCE2INTERRUPTED
0002 AND HOMESEQUENCE2)
0003 OR JOG_activated)
0004 AND HOME_BUTTON
0005 AND StopSignal
0006 AND DrivelsReferencedRot AND DrivelsReferencedY
0007 AND DrivelsReferencedZ AND DrivelsReferencedZ2
0008 AND (TML2_SEL OR REF2_SEL)
0009 AND (NOT DC_END0
0010 AND NOT DC_END1)
0011 AND (TML2_SEL
0012 OR REF2_SEL)

HomeSequence_curved (FB-SFC).Action Step9 - Entry (ST)
0001 RecordNoY:=3;

HomeSequence_curved (FB-SFC).Action Step9 (ST)
0001 HomeSequence_curved_R_TRIGInstY(CLK:=HOME_BUTTON);
0002 HomeSequence_curved_R_TRIGInstY1(CLK:=STEP9);
0003
0004 StartTaskY:=HomeSequence_curved_R_TRIGInstY.Q
0005 OR HomeSequence_curved_R_TRIGInstY1.Q;

HomeSequence_curved (FB-SFC).Transition Trans17 (ST)
0001 ActualPositionY = 90
0002 AND Y_MOTIONCOMPL
0003 AND StopSignal

HomeSequence_curved (FB-SFC).Action Step21 (ST)
0001 StartTaskY:=FALSE;
0002 HomeSequence_curved.TONInst(In:=DC_END1, PT:=T#500ms);
0003
0004 DC_FWD:=STEP21
0005 AND NOT HomeSequence_curved.TONInst.Q
0006 AND StopSignal;

HomeSequence_curved (FB-SFC).Transition Trans20 (ST)
0001 StopSignal
0002 AND HomeSequence_curved.TONInst.Q
0003 AND DC_END1

HomeSequence_curved (FB-SFC).Transition Trans22 (ST)
0001 SFCErrror

HomeSequence_curved (FB-SFC).Action Step23 (ST)
0001 DC_FWD:=FALSE;

HomeSequence_curved (FB-SFC).Action Step22 (ST)
0001 DC_FWD:=FALSE;

HomeSequence_curved (FB-SFC).Action Step2 - Entry (ST)
0001 RecordNoY:=1;

HomeSequence_curved (FB-SFC).Action Step2 (ST)
0001 HomeSequence_curved_R_TRIGInstY(CLK:=HOME_BUTTON);
0002 HomeSequence_curved_R_TRIGInstY1(CLK:=STEP2);
0003
0004 StartTaskY:=HomeSequence_curved_R_TRIGInstY.Q
0005 OR HomeSequence_curved_R_TRIGInstY1.Q;
0006
0007

HomeSequence_curved (FB-SFC).Transition Trans2 (ST)
0001 ActualPositionY = 0
0002 AND Y_MOTIONCOMPL
0003 AND StopSignal

HomeSequence_curved (FB-SFC).Action Step3 - Entry (ST)

```

```

0001 RecordNoZ:=1;

HomeSequence_curved (FB-SFC).Action Step3 (ST)
0001 HomeSequence_curved.R_TRIGInstZ(CLK:=HOME_BUTTON);
0002 HomeSequence_curved.R_TRIGInstZ1(CLK:=STEP3);
0003
0004 StartTaskZ:=HomeSequence_curved.R_TRIGInstZ.Q
0005 OR HomeSequence_curved.R_TRIGInstZ1.Q;
0006
0007 StartTaskY:=FALSE;

HomeSequence_curved (FB-SFC).Transition Trans3 (ST)
0001 ActualPositionZ = 0
0002 AND Z_MOTIONCOMPL
0003 AND StopSignal
0004 AND DEMAGN

HomeSequence_curved (FB-SFC).Action Step4 (ST)
0001 Activate_K4:=TRUE;
0002 StartTaskZ:=FALSE;

HomeSequence_curved (FB-SFC).Transition Trans5 (ST)
0001 K4_OK
0002 AND StopSignal

HomeSequence_curved (FB-SFC).Transition Trans4 (ST)
0001 ActualPositionZ = 0
0002 AND Z_MOTIONCOMPL
0003 AND StopSignal
0004 AND NOT DEMAGN

HomeSequence_curved (FB-SFC).Action Step5 - Entry (ST)
0001 SetPosRot:=ActualPositionRot + 4;
0002 SetVeloRot:=41;

HomeSequence_curved (FB-SFC).Action Step5 (ST)
0001 StartTaskRot:=STEP5;
0002
0003 IF AckStartRot
0004 AND K4_OK
0005 THEN DEMAGN:=FALSE;
0006 END_IF;

HomeSequence_curved (FB-SFC).Transition Trans6 (ST)
0001 NOT DEMAGN
0002 AND ROT_MOTIONCOMPL
0003 AND NOT Rotation_CTRL.DrivelsMoving
0004 AND StopSignal

HomeSequence_curved (FB-SFC).Transition Trans7 (ST)
0001 NOT StopSignal

HomeSequence_curved (FB-SFC).Action Step12 (ST)
0001 StartTaskRot:=FALSE;

HomeSequence_curved (FB-SFC).Transition Trans8 (ST)
0001 HOME_BUTTON
0002 AND StopSignal

HomeSequence_curved (FB-SFC).Action Step6 - Entry (ST)
0001 RecordNoZ2:=1;

HomeSequence_curved (FB-SFC).Action Step6 (ST)
0001 HomeSequence_curved.R_TRIGInstZ2(CLK:=HOME_BUTTON);
0002 HomeSequence_curved.R_TRIGInstZ21(CLK:=STEP6);
0003
0004 StartTaskZ2:=HomeSequence_curved.R_TRIGInstZ2.Q
0005 OR HomeSequence_curved.R_TRIGInstZ21.Q;
0006
0007 Activate_K4:=FALSE;
0008 StartTaskRot:=FALSE;

```

```

0009 StartTaskZ:=FALSE;

HomeSequence_curved (FB-SFC).Transition Trans9 (ST)
0001 StopSignal
0002 AND ActualPositionZ = 0
0003 AND Z2_MOTIONCOMPL
0004 AND ActualPositionY = 0
0005 AND ActualPositionZ = 0

HomeSequence_curved (FB-SFC).Transition Trans24 (ST)
0001 StopSignal
0002 AND (ActualPositionY > 0
0003 OR ActualPositionY < 0
0004 OR ActualPositionZ > 0
0005 OR ActualPositionZ < 0)

HomeSequence_curved (FB-SFC).Action Step26 (ST)
0001 StartTaskZ2:=FALSE;

HomeSequence_curved (FB-SFC).Transition Trans25 (ST)
0001 HOME_BUTTON (*press home button to continue*)
0002 AND StopSignal

HomeSequence_curved (FB-SFC).Action Step7 (ST)
0001 Reset_JogActivated:=TRUE;
0002 StartTaskZ2:=FALSE;

HomeSequence_curved (FB-SFC).Transition Trans10 (ST)
0001 Reset_JogActivated
0002 AND StopSignal

HomeSequence_curved (FB-SFC).Action Step8 (ST)
0001 Reset_JogActivated:=FALSE;
0002 SEQUENCE2INTERRUPTED:=FALSE;
0003 HOMESEQUENCE2:=FALSE;

HomeSequence_curved (FB-SFC).Transition Trans11 (ST)
0001 NOT HOMESEQUENCE2
0002 AND NOT SEQUENCE2INTERRUPTED
0003 AND NOT Reset_JogActivated
0004 AND StopSignal

HomeSequence_curved (FB-SFC).Transition Trans12 (ST)
0001 ((SEQUENCE2INTERRUPTED
0002 AND HOMESEQUENCE2)
0003 OR JOG_activated)
0004 AND HOME_BUTTON
0005 AND StopSignal
0006 AND DrivesReferencedRot AND DrivesReferencedY
0007 AND DrivesReferencedZ AND DrivesReferencedZ2
0008 AND (TML2_SEL OR REF2_SEL)
0009 AND ActualPositionY > 50
0010 AND (DC_END0
0011 OR DC_END1)
0012 AND (TML2_SEL
0013 OR REF2_SEL)
0014

HomeSequence_curved (FB-SFC).Action Step13 - Entry (ST)
0001 RecordNoY:=4;

HomeSequence_curved (FB-SFC).Action Step13 (ST)
0001 HomeSequence_curved.R_TRIGInstY(CLK:=HOME_BUTTON);
0002 HomeSequence_curved.R_TRIGInstY1(CLK:=STEP13);
0003 StartTaskY:=HomeSequence_curved.R_TRIGInstY.Q OR HomeSequence_curved.R_TRIGInstY1.Q;
0004
0005

HomeSequence_curved (FB-SFC).Transition Trans13 (ST)
0001 ActualPositionY = 77
0002 AND Y_MOTIONCOMPL

```

```

0003 AND StopSignal

HomeSequence_curved (FB-SFC).Action Step14 - Entry (ST)
0001 RecordNoZ:=1;
HomeSequence_curved (FB-SFC).Action Step14 (ST)
0001 HomeSequence_curved.R_TRIGInstZ(CLK:=HOME_BUTTON);
0002 HomeSequence_curved.R_TRIGInstZ1(CLK:=STEP14);
0003
0004 StartTaskZ:=HomeSequence_curved.R_TRIGInstZ.Q
0005 OR HomeSequence_curved.R_TRIGInstZ1.Q;
0006
0007 StartTaskY:=FALSE;

HomeSequence_curved (FB-SFC).Transition Trans14 (ST)
0001 ActualPositionZ = 0
0002 AND Z_MOTIONCOMPL
0003 AND StopSignal
0004

HomeSequence_curved (FB-SFC).Action Step15 - Entry (ST)
0001 RecordNoY:=1;
HomeSequence_curved (FB-SFC).Action Step15 (ST)
0001 HomeSequence_curved.R_TRIGInstY(CLK:=HOME_BUTTON);
0002 HomeSequence_curved.R_TRIGInstY1(CLK:=STEP15);
0003
0004 StartTaskY:=HomeSequence_curved.R_TRIGInstY.Q
0005 OR HomeSequence_curved.R_TRIGInstY1.Q;
0006
0007 StartTaskZ:=FALSE;

HomeSequence_curved (FB-SFC).Transition Trans15 (ST)
0001 ActualPositionY = 0
0002 AND Y_MOTIONCOMPL
0003 AND StopSignal
0004 AND DEMAGN

HomeSequence_curved (FB-SFC).Transition Trans16 (ST)
0001 ActualPositionY = 0
0002 AND Y_MOTIONCOMPL
0003 AND StopSignal
0004 AND NOT DEMAGN

HomeSequence_curved (FB-SFC).Action Step17 (ST)
0001 StartTaskY:=FALSE;

HomeSequence_curved (FB-SFC).Action Step16 (ST)
0001 StartTaskY:=FALSE;

```