

Lassi Lavanto

CRP-PROPULSIOLAITTEEN PIENOISMALLIN
AUTOMATISOINTI

Automaatiotekniikan koulutusohjelma
2015

CRP-PROPULSIOLAITTEEN PIENOISMALLIN AUTOMATISOINTI

Lavanto, Lassi
Satakunnan ammattikorkeakoulu
Automaatiotekniikan koulutusohjelma
Maaliskuu 2015
Ohjaaja: Suvela, Timo
Sivumäärä: 60
Liitteitä: 7

Asiasanat: logiikkaohjelmointi, CAN, ohjausjärjestelmät, pienoismallit

Tässä opinnäytetyössä automatisoitiin Steerprop Oy:n omistama CRP-propulsiolaitteen pienoismalli. Automatisoinnilla pyrittiin parantamaan pienoismallin kiinnostavuutta merenkulkualan tapahtumissa sekä antamaan messuvieraille laajempi kuva yrityksen tuottaman pääohjauksen rakenteesta.

Tavoitteena oli kehittää pienoismallille ohjausjärjestelmä, joka vastaa yrityksen tuottamien potkurilaitteiden pääohjausta. Varsinaisten potkurilaitteiden ohjausjärjestelmän komentosiltakomponenteista oli tarkoitus lisätä pienoismallille ohjaukskahva sekä digitaalinen kulmanäyttö. Näiden lisäksi järjestelmän toiminnallisuuden toteutusta varten pienoismallin uuteen järjestelmään oli lisättävä kulma-anturi, logiikka ja käyttöliittymäpaneeli. Työn sisältöön kuului myös kulmanäytön CAN-ominaisuuksien tutkiminen.

Opinnäytetyön suunnitteluosuus hoidettiin pääasiassa tietokonepohjaisia suunnitteluja ohjelmointiohjelmistoja käyttäen. Pienoismallin CAN-väyläpohjaisen tiedonvaihdon toteutus vaati suunnitteluvaiheessa paljon teoreettisen tiedon hankintaa CAN-protokollasta, CAN-komponenteista sekä Omronin CAN-kommunikointiyksiköstä.

Työn tutkimuksellisen osuuden tavoite oli lähettää kulmanäytön kaikki tarvitsema tieto logiikasta näytölle CAN-väylää käyttäen. Kulmanäytön CAN-ominaisuuksien tutkiminen sisälsi väylän monitorointia sekä väylän viestien analysointia, millä selvitettiin kulmanäytön tarvitsemien viestikehysten rakennetta.

Toteutusvaihe sisälsi järjestelmän kokoamisen suunnitteluaineiston pohjalta sekä järjestelmän testaamisen. Tulokseksi saatiin ohjausjärjestelmä, jonka komponentit ja toiminnallisuudet vastasivat varsinaisen potkurilaitteen pääohjausta. Pienoismallin lopullinen järjestelmä sisälsi myös markkinointia varten rakennettuja ominaisuuksia, joita ei varsinaisen potkurilaitteen ohjausjärjestelmä sisällä.

THE AUTOMATIZATION OF A MINIATURE CRP-PROPULSOR

Lavanto, Lassi

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in Automation

March 2015

Supervisor: Suvela, Timo

Number of pages: 60

Appendices: 7

Keywords: logic programming, CAN, control systems, miniatures

In this thesis, a miniature CRP-propulsor model owned by Steerprop Ltd was automatized. The reason for automatization was to enhance the attractiveness of the model and give a better view of the structure of the main control system produced by the company.

The aim of the thesis was to build a new control system to the miniature. The control system had to be similar to the main control system of the propulsors produced by the company so there were new components to be added in model's system. The required bridge instruments were a control lever and an angle display. Other necessary components were a programmable logic controller, a human-machine-interface and an angle sensor. The second aim of the thesis was to study the CAN-protocol features of the angle display.

The design part of the thesis was conducted using computer-based designing and programming software. The information exchange between devices was carried out with a CAN-bus solution which meant that the design part also required a lot of research on CAN-protocol and CAN-based components interfaces.

The research part of the thesis was to find out how all necessary information can be sent from the PLC to angle display via CAN-bus. The research included a lot of bus monitoring. It was also necessary to analyze a few messages on the CAN-bus to get knowledge of the structure of the message frames used by the angle display.

The implementation part of the thesis included assembly of the system. After the assembly all the functionalities of the system were tested. As an outcome, a control system with components and functionalities just like a real propulsor's control system was created. In this case there some extra functions were also built into the system, because the model is used in exhibitions as a marketing tool to the company.

TERMIT JA LYHENTEET

2D	Kaksiulotteinen esitystapa (2-Dimensional)
3D	Kolmeulotteinen esitystapa (3-Dimensional)
AMP	Viestien törmäysmenettely (Arbitration on Message Priority)
BCD	Binäärikoodattu esitystapa (Binary Coded Decimal)
CAN	Sarjaliikenneprotokolla (Controller Area Network)
CCW	Vastapäivään (Counterclockwise)
COB-ID	CANopen sanomanumero (Communication Object Identifier)
CPU	Keskusyksikkö (Central Processing Unit)
CRC	Tarkistusalgoritmi (Cyclic Redundancy Check)
CRP	Vastakkain pyörivät propellit (Contra-Rotating Propellers)
CSMA/CD	Siirtotien varausmenetelmä (Carrier Sense Multiple Access With Collision Detection)
CW	Myötäpäivään (Clockwise)
ECP	Konehuoneen ohjauspaneeli (Engine Control Room Panel)
Ethernet	Lähiverkko
I/O	Tulot ja lähdöt (Input/Output)
Indikaattori	Tapahtuman tai muutoksen näyttävä osoitin tai osoitinlaite
LD	Tikapuu/Relekaavio-ohjelmointi (Ladder Diagram)
mA-viesti	Milliampeeriviesti
NRZ	Johtokoodaus (Non-Return-to-Zero)
PDO	Sanomaprotokolla (Process Data Object)

PLC	Ohjelmoitava logiikka (Programmable Logic Controller)
PWM	Pulssinleveysmodulaatio (Pulse-width modulation)
Referenssi	Ohje- tai vertailuarvo
RPM	Kierrosnopeuden yksikkö (Revolutions Per Minute)
RS-232	Sarjaliikenneväylä (Recommended Standard 232)
SDO	Sanomaprotokolla (Service Data Object)
SFC	Vuokaavio-ohjelmointi (Sequential Function Chart)
ST	Rakenteinen teksti -ohjelmointi (Structured Text)
USB	Sarjaväyläarkkitehtuuri (Universal Serial Bus)
WCP	Siltaohjauspaneeli (Wheelhouse Control Panel)

SISÄLLYS

1	JOHDANTO.....	7
2	LÄHTÖTILANNE	8
3	JÄRJESTELMÄ.....	9
3.1	Järjestelmän yleiskuvaus.....	9
3.2	Komponentit	10
3.3	Ohjelmistot.....	13
4	SÄHKÖSUUNNITTELU	15
4.1	Moottorien ohjaus	15
4.2	Piirikaavio, osaluettelo ja sijoituskuvat	18
5	OHJELMISTOSUUNNITTELU.....	19
5.1	Ominaisuudet ja parametrit.....	19
5.2	Funktiot	20
5.2.1	APR-funktio	20
5.2.2	SFT-funktio	23
5.3	Logiikkaohjelma	25
5.3.1	Sykliset ohjelmat	26
5.3.2	Keskeytysohjelmat	34
6	CAN-VÄYLÄ	38
6.1	CAN-protokolla	38
6.2	CANopen	39
6.3	Viestikehys.....	40
7	KULMANÄYTÖN CAN-OMINAISUUKSIEN TUTKIMINEN.....	43
7.1	Väylärakenne	43
7.2	Viestien tutkinta.....	44
7.3	Lähetysten rakennus logiikkaohjelmaan.....	46
7.3.1	Lähetyspuskurien määrittäminen.....	47
7.3.2	Lähetettävän viestin dataisuus.....	48
7.3.3	Kirkkaudensäätö.....	51
7.3.4	Heartbeat	54
8	KÄYTTÖLIITTYMÄ	55
9	YHTEENVETO	56
	LÄHTEET.....	59
	LIITTEET	

1 JOHDANTO

Tämän työn tarkoituksena oli automatisoida Steerprop Oy:n omistaman CRP-potkurilaitteen pienoismallin ohjaus. Pienoismalli on hankittu yritykseen vuonna 2012 markkinointikäyttöä varten. Se kiertää vuosittain maailmalla merenkulkualan tapahtumissa (esim. Shipbuilding, Machinery & Marine Technology Trade Fair ja Europort Istanbul) edustamassa yrityksen tuotantoa.

Opinnäytetyön tavoitteena oli tuottaa mallille ohjausjärjestelmä, joka kuvaisi Steerpropin potkureiden ohjausjärjestelmää pääohjauksen osalta. Tarkoituksena oli sijoittaa mallin ohjaukseen yrityksen käyttämistä komponenteista ohjauskahva, digitaalinen kulmanäyttö sekä käyttöliittymäpaneeli. Työhön kuului järjestelmän suunnittelu, sähkösuunnittelu, ohjelmistosuunnittelu, osien tilaus sekä järjestelmän kokoonpano. Messutapahtumia varten mallille tuli tehdä asennus- ja käyttöohjeet sekä Steerpropin tuottamaa ohjausjärjestelmää esittelevä markkinointikansio. Pienoismallin automatisoinnilla pyrittiin parantamaan yrityksen messuosaston kiinnostavuutta merenkulkualan messuilla sekä havainnollistamaan vieraille yrityksen tuottamaa ohjausjärjestelmää.

Mallin automatisointiin sisällytettiin myös uuden digitaalisen kulmanäytön CAN-rajapinnan ja -ominaisuuksien tutkiminen ja testaus. Tutkimuksen tavoitteena oli toteuttaa kulmanäytön tiedonvaihto kokonaan CAN-pohjaisena. Tutkimuksen tuloksia on tarkoitus hyödyntää yrityksen ohjausjärjestelmässä tulevaisuudessa.

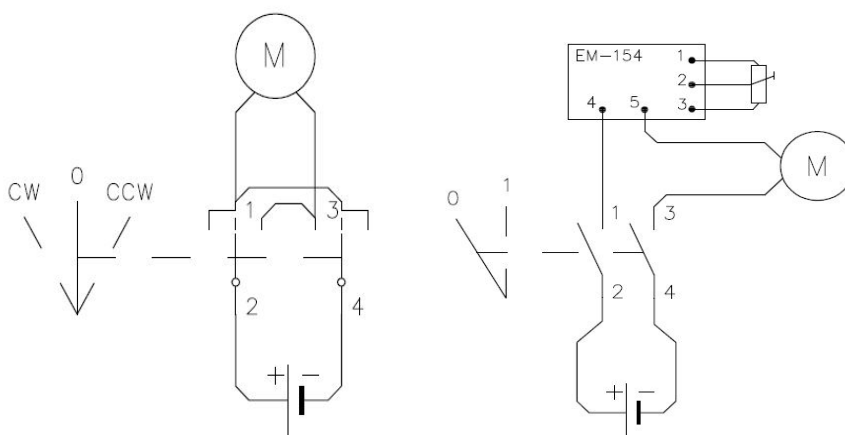
Työn tilaajana oli Steerprop Oy, joka suunnittelee ja valmistaa kääntyviä azimuth-propulsioilaitteita pääasiassa vaikeissa ja arktisissa olosuhteissa työskenteleviin laivoihin ja aluksiin. Yritys on perustettu vuonna 2000, ja sen liikevaihto vuonna 2013 oli 27 miljoonaa euroa. Steerprop työllistää tällä hetkellä n. 50 henkilöä suoraan ja alihankkijoidensa kautta noin 400. Yrityksen toimipisteet sijaitsevat Raumalla ja Jyväskylässä, ja sillä on myös tytäryhtiö Venäjällä.

2 LÄHTÖTILANNE

Pienoismalli (kuva 1) sisältää valmiiksi kaksi 24 V:n tasavirtamoottoria, joista toinen sijaitsee mallin pohjassa ja toinen mallin päällä. Pohjassa oleva moottori on liitetty hammashihnan välityksellä potkurilaitetta pyörittävään keskiakseliin ja päällä oleva moottori vaihteen välityksellä propelleja pyörittävään akseliin. Tähän asti potkurin käännön ohjaus on toiminut 3-asentoisella kytkimellä (CW-OFF-CCW). Propellien ohjaus on sisältänyt ON/OFF-kytkimen sekä potentiometrin ja Electromenin EM-154-himmentimen pyörimisnopeuden säätöön.



Kuva 1. CRP-laitteen pienoismalli



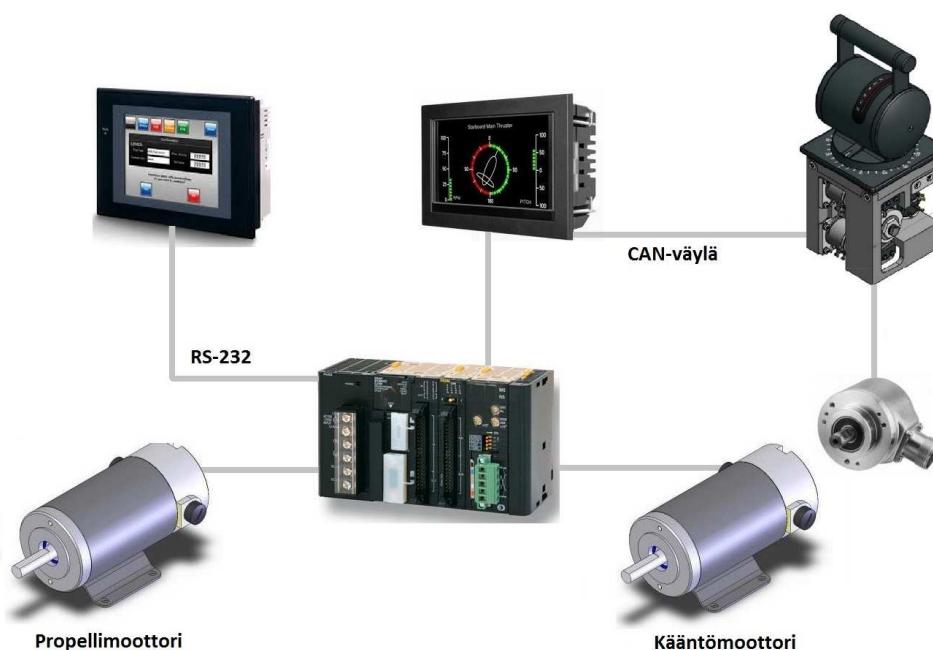
Kuva 2. Käännön- ja propellinohjauksen piirikaaviot

Uuden ohjausjärjestelmän rakentaminen ja komponenttien lisääminen mahdollistaa laitteelle paljon uusia haluttuja ominaisuuksia, joita aikaisemmalla toteutuksella (kuva 2) ei ole ollut mahdollista tai järkevää toteuttaa. Näitä ominaisuuksia ovat mm. paikoitus, follow up -ohjaus, kiihdytys- ja hidastusrampit, erilaiset indikoinnit, hälytykset sekä diagnostiikka.

3 JÄRJESTELMÄ

3.1 Järjestelmän yleiskuvaus

Uuden järjestelmän suunnittelu lähti liikkeelle järjestelmän kokonaisuuden ja liityntöjen (kuva 3) hahmottamisesta sekä komponentteihin ja tarvittaviin ohjelmistoihin tutustumisesta. Järjestelmään lisättäviä komponentteja olivat Omronin CJ1M-logiikka ja NS5-käyttöpäätte, Kwant Controlsin RSCU-ohjauskahva, DEIF:n XDi-kulmanäyttö sekä FSG:n absoluuttianturi.



Kuva 3. Järjestelmän laitteet ja liitynnät

Pienoismallin suuruuden ja muodon vuoksi oli tarpeen sijoittaa osa uusista komponenteista erilleen pienoismallista, jotta koko järjestelmän siirrot ja kuljetukset pysyisivät helppoina ja kevyinä. Tämän vuoksi ohjauslaitteille suunniteltiin oma ohjauslaatikko (liite 1). Yrityksen mekaniikkaosastolla SolidWorks-ohjelmistoa hyödyntäen piirrettiin 2D-kuvien mukainen 3D-kuva (kuva 4), joka lähetettiin laatikon valmistajalle.



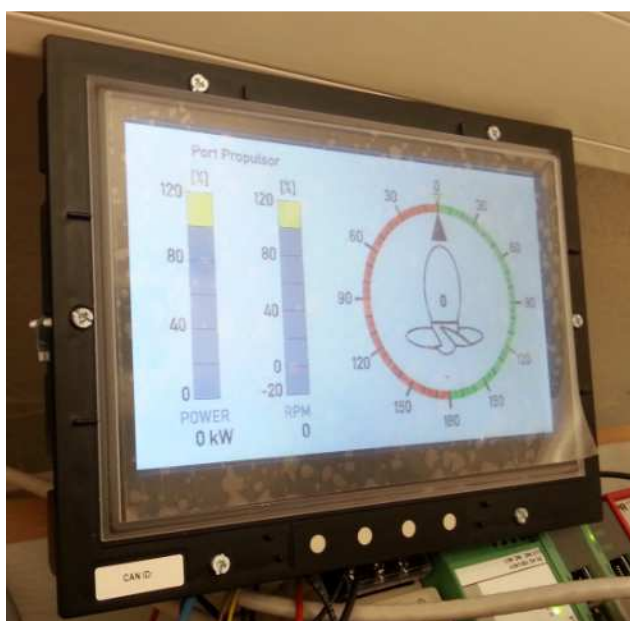
Kuva 4. Ohjauslaatikko

3.2 Komponentit

- Digitaalinen kulmanäyttö

XDi 192 Multi on tanskalaisen yrityksen DEIF A/S:n valmistama digitaalinen kulmanäyttö. Fyysisesti XDi-sarjan näyttöjä on kolmea eri kokoa (XDi 96, XDi 144 ja XDi 192), joista kutakin on valittavissa yhden (Single), kahden (Dual) tai useamman (Multi) indikaattorin sovelluksella. Näytön ohjelmisto on asiakaskohtaisesti muunneltavissa indikaattoreiden ulkonäön, sijoittelun ja skaalauksen osalta. Steerprop Oy:n käyttämä kulmanäyttö (kuva 5) sisältää 6 indikaattoria: kulman referenssipiste, todellinen kulma, prosentuaalinen rpm, todellinen rpm, prosentuaalinen teho sekä todellinen teho.

Näytön pohjassa on liittimet kahdelle erilliselle CAN-väylälle, joista molemmista on mahdollista tuoda ja lähettää dataa. CAN-väylän sijaan tiedot näytölle voidaan tuoda vaihtoehtoisesti myös analogiaviestein, sillä näytön takana on paikat kahdelle erilliselle analogiamoduulille. Näyttö on konfiguroitavissa CAN-väylän kautta käyttämällä CANopen-protokollan SDO-viestejä. (DEIF:n [www-sivut a.](#))



Kuva 5. DEIF:n XDi-kulmanäyttö

- Ohjauskahva

RSCU-H96 El.Shaft on hollantilaisen Kwant Controls B.V:n valmistama ohjauskahva (kuva 6), joka on tarkoitettu azimuth-propulsiolaitteen ohjaukseen. Kahva sisältää kaksi erilaista ohjainpiiriä. Molemmat piirit on kahdennettu, joten kahvassa on kaksi M12-liitäntää. Kahvan ohjainpiirit ovat kahvan asennon ohjainpiiri (nopeuspiiri) ja kahvan kulman ohjainpiiri (kulmapiiri). Molemmat piirit lähettävät ja vastaanottavat dataa CAN-protokollaa käyttäen. Molemmat piirit myös käyttävät omaa tunnistetta (Node-ID), jotta väylän muut laitteet viestin saadessaan kykene-



Kuva 6. Ohjauskahva

vät tunnistamaan, kumman piirin lähettämästä tiedosta on kyse. Tunnisteen avulla kahva myös vastaanottaessaan tunnistaa, kummalle piirille vastaanotettava data kuuluu. Molemmat piirit sisältävät ohjaimen lisäksi anturin ja moottorin. Kahva on lisäksi konfiguroitavissa (esim. piirien noodinumerot) CAN-väylän kautta käyttämällä CANopen-protokollan SDO-viestejä.

Piirien anturit mittaavat kahvan asentoa, jonka arvon kahva lähettää CAN-väylälle binäärikoodattuna CAN-viestikehyksen dataosuuden ensimmäisessä sanassa (kulmapiirin lähettämät arvot välillä $\pm 1800_{\text{dec}}$, nopeuspiirin lähettämät arvot välillä $0-1000_{\text{dec}}$). Viestikehyksen dataosuuden toinen sana on tilasana, jonka jokainen bitti tarkoittaa ennalta määritettyä tilatietoa (esim. bitti 9 = piirin lämpötilavaroitus). Piirien moottorit mahdollistavat kahvan asennon ulkoisen ohjauksen ja ns. tuntokohtien asettamisen kahvalle. Kahvalle lähetettävän viestikehyksen tulee koostua kahdesta sanasta. Ensimmäinen sana on kahvan asennon ulkoisen ohjauksen referenssiarvo, joka määrää mihin asentoon kahvan tulee ajaa itsensä (datan tulee olla kulmapiirille lähetettävissä viestissä välillä $\pm 1800_{\text{dec}}$ ja nopeuspiirille lähetettävässä viestissä $0-1000_{\text{dec}}$). Lähetettävän viestikehyksen dataosuuden toinen sana on tilasana, jonka jokainen bitti tarkoittaa ennalta määrättyä ohjausta (esim. bitti 1 = master/slave). (Kwant Controls B.V. 2013.)

- Kulma-anturi

MH 620 on kiertymäkulmaa mittaava magneettinen absoluuttianturi (kuva 7), jota valmistaa saksalainen yhtiö FSG. Steerprop Oy:n käyttämä anturiversio sisältää kaksi erillistä kanavaa, joissa molemmissa on sekä analoginen (mA-viesti) että digitaalinen (CAN-viesti) lähtö. CAN-lähdön kautta anturi lähettää datansa 12-bittisenä väylälle. Anturin kiertymäkulman arvot 0–360° vastaavat lähetyksessä lineaarisesti arvoja välillä 0–4095_{dec}. Vastaavasti analogialähdön signaali on 4–20 mA (0–360°). Molemmissa kanavissa on myös erillinen nol-laus-johto, jolla saadaan määritettyä anturin nollakulma. Johdon yhdistyessä jännitteen maatasoon määrittää anturi sen hetkisen kulmansa arvoon 180° (lähdöillä tällöin arvot 2047_{dec} ja 12mA). (FSG:n www-sivut.)



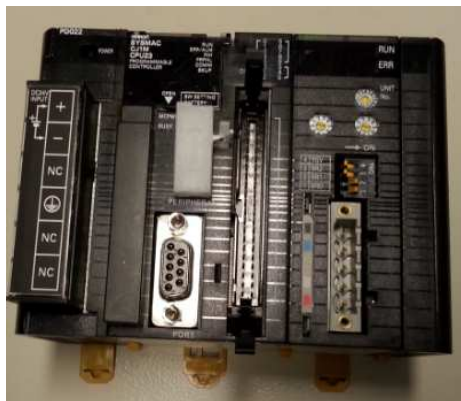
Kuva 7. Kulma-anturi kiinnitettyä mallin pohjaan

- Logiikka

CJ1M on Omronin valmistama modulaarinen logiikkasarja. Sarjaan kuuluu 6 erilaista keskusyksikköä, jotka eroavat toisistaan muistinsa ja sisäänrakennetun I/O:n puolesta. Tähän työhön valittiin CJ1M-sarjan keskusyksikkö CPU23 (kuva 8), jossa on sisäänrakennettuna 10 tuloa ja 6 lähtöä. Keskusyksiköstä löytyy myös kaksi sarjaliikenneporttia ja muistikorttipaikka. Logiikan kytkeminen CAN-väylään vaatii lisäksi CORT21-kommunikointiyksikön, joka on Omronin valmistama CAN-kommunikointia varten tarkoitettu lisämoduuli. (Omronin www-sivut a.)

- Käyttöpääte

Käyttöpäätteeksi valittiin Omronin 5,7 tuuman NS5-SQ11B-V2-kosketusvärinäyttö (kuva 9). Se on graafisesti ohjelmoitava komponentti, jolla voidaan valvoa ja ohjata järjestelmää. Näytössä on seuraavat liitännät: 1 x USB Slave, 1 x Ethernet ja 2 x RS-232. (Omronin [www-sivut b.](#))



Kuva 8. Logiikka



Kuva 9. Käyttöpääte

3.3 Ohjelmistot

- CX-Programmer

CX-Programmer on ohjelmointiohjelmisto Omronin CP-, CS- sekä CJ-sarjan logiikoille. Ohjelma kuuluu Omronin julkaisemaan CX-One nimiseen ohjelmistopakettiin. Ohjelma tukee kansainvälisen standardin IEC 61131-3 määrittämistä ohjelmointikielistä LD-, ST- sekä SFC-ohjelmointia. (Omronin [www-sivut c.](#))

- CX-Designer

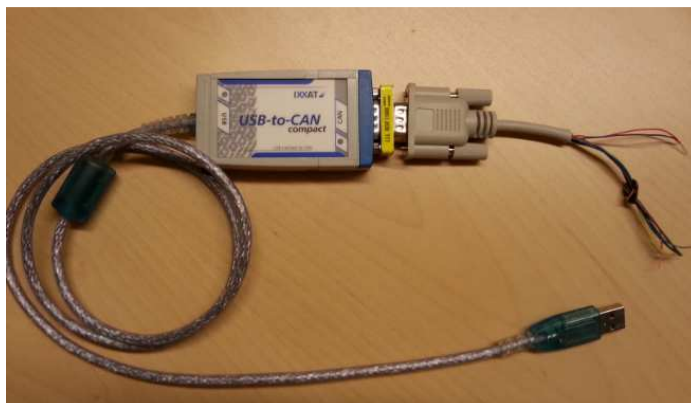
CX-Designer on käyttöpääteohjelmisto, jolla ohjelmoidaan graafisesti Omronin NS-käyttöpäatesarjan tuotteita. Ohjelma kuuluu Omronin julkaisemaan CX-One-ohjelmistopakettiin. (Omronin [www-sivut d.](#))

- CADS Planner

Kymdatán CADS Planner -tuotesarja on tarkoitettu tietokonepohjaiseen suunnitteluun. Sarja sisältää ohjelmistot sähkö-, LVI-, automaatio- ja talosuunnittelulle. Tässä työssä käytetään CADS Planner Electric -ohjelmistoa.

- MiniMon V3

MiniMon V3 on IXXAT Automation GmbH:n valmistama CAN-väylän monitorointiin tarkoitettu ohjelma, jolla voidaan virtuaalisesti monitoroida väylän viestejä sekä tilaa. Ohjelmalla voidaan myös lähettää käyttäjän itse määrittelemiä viestejä väylälle. Tästä syystä ohjelma on erittäin hyvä CAN-komponenttien konfigurointiin. Ohjelma tukee sekä 11- että 29-bittisiä tunnisteita (IXXAT:n [www-sivut](http://www.ixxat.com)). Fyysisesti väylään liitytään IXXAT:n USB/CAN-adapterilla (kuva 10).



Kuva 10. USB/CAN-adapteri

4 SÄHKÖSUUNNITTELU

4.1 Moottorien ohjaus

Sähkösuunnittelu painottui pitkälti moottorien ohjauksen suunnitteluun. Tärkeintä suunnittelussa oli mahdollistaa kiihdytys- ja hidastusrampit moottoreille. Tähän tarkoitukseen vertailtiin kahta erilaista toteutustapaa:

- Erillinen moottoriohjain

Electromenin EM-101 24V 3A 4-Quadrant DC -moottoriohjaimien (kuva 11) lisäys järjestelmään, jolloin moottorien ohjaus tapahtuu analogiasignaalein (0–10 V). Ohjain sisältää esiaseteltavat



Kuva 11. EM-101

rampit (0–5 s) kiihdytykseen ja jarrutukseen. Analogiohjauksen toteutus vaatii logiikkaan liitettävän analogia-I/O-moduulin (MAD42). (Electromenin [www-sivut.](http://www.sivut.fi))

- Logiikan PWM-lähtö

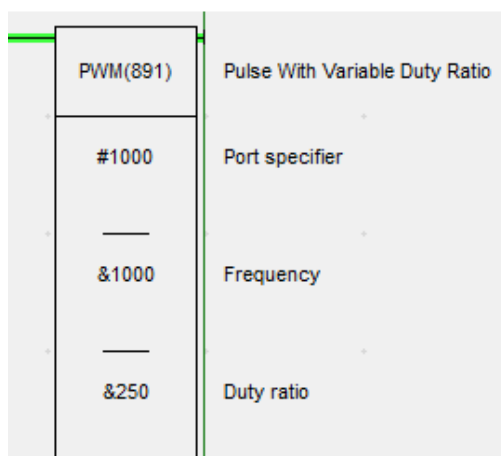
CJ1M-CPU23-logiikassa on sisäänrakennettuna kaksi PWM-lähtöä, joita hyödyntämällä saadaan järjestelmään haluttu nopeudensäätö. PWM eli pulssinleveysmodulaatio on tapa, jolla saadaan jännitettä säädettyä pulssisuhdetta muuttamalla. Tavallisen mekaanisen releen kytkentätaajuus ei riitä PWM-ohjaukseen, joten releeksi tulee valita puolijohderele. Moottorien ohjaukseen 100 Hz:n ohjaustaajuus on riittävä, joten esimerkiksi Phoenixin OPT-24DC/42DC/2-puolijohderele käy tähän tarkoitukseen (max. kytkentätaajuus 300 Hz).

Lopputuloksena päädyttiin käyttämään logiikan PWM-lähtöä, sillä tämä ominaisuus oli valmiiksi käytettävissä kyseisessä logiikkamallissa. Päätökseen vaikuttivat myös lisäkomponenttien kustannukset, sillä kahden puolijohdereleen hinta on huomatta-

vasti pienempi kuin analogiamoduulin ja kahden moottoriohjaimen hinta. Valittu ohjaustapa on lisäksi yksinkertaisempi kytkeä ja helpommin huollettavissa.

PWM-ohjaus otetaan käyttöön CX-Programmer-ohjelmassa omalla PWM(891)-funktioilla (kuva 12). PWM-funktion käyttö ohjelmassa kumoaa kyseisen lähdön normaalin lähtöosoitteen (OUT 4 = 2961.04, OUT 5 = 2961.05) tilan. PWM-funktiossa on kolme operandia (Omronin www-sivut e.):

1. "Port specifier" -operandi määrää funktion käyttämän PWM-portin (1000_{hex} = Output 4, 1001_{hex} = Output 5).
2. "Frequency" -operandi määrää käytettävän taajuuden (Hz) kymmenesosan tarkkuudella. Arvo 1000 tarkoittaa siis 100 Hz:n taajuutta. Lähtö pystyy toimimaan hyvin tarkasti alueella 0,1 Hz - 1 kHz. Tätä korkeammassa taajuuksissa alkaa esiintyä virhettä (max. lähtötaajuus 6,5535 kHz).
3. "Duty ratio" määrää pulssisuhteen kymmenesosan tarkkuudella (0–1000 = 0–100 %). Pulssisuhde määrää, kuinka suuren osan yhdestä pulssista lähdön tila on päällä. Arvolla 250 suhde on 25 % (kuva 13).



Kuva 12. PWM(891)-funktio

Kuvan 12 mukaisilla arvoilla ohjauksen jännitteen tulisi olla 12 voltia ja lähdön tulisi olla "1"-tilassa 2,5 ms ja "0"-tilassa 7,5 ms vuorotellen. Lähdön päälläoloajan saa laskettua kaavalla 1 ja ohjauksen jännitteen ilman kuormaa kaavalla 2.

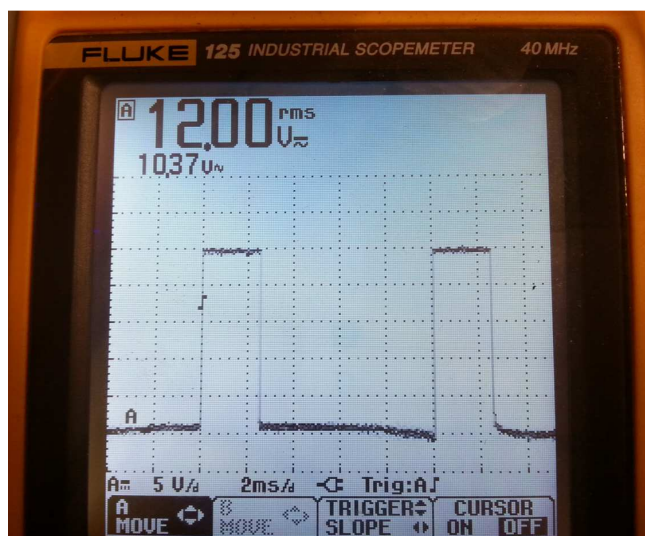
$$t = \frac{1}{f} * \frac{x}{100} = \frac{x}{100 * f} = \frac{25}{100 * 100} = 0,0025 = 2,5 \text{ ms} \quad (1)$$

jossa t on lähdön päälläoloaika (s), f on käytettävä taajuus (Hz) ja x on pulssisuhde (%).

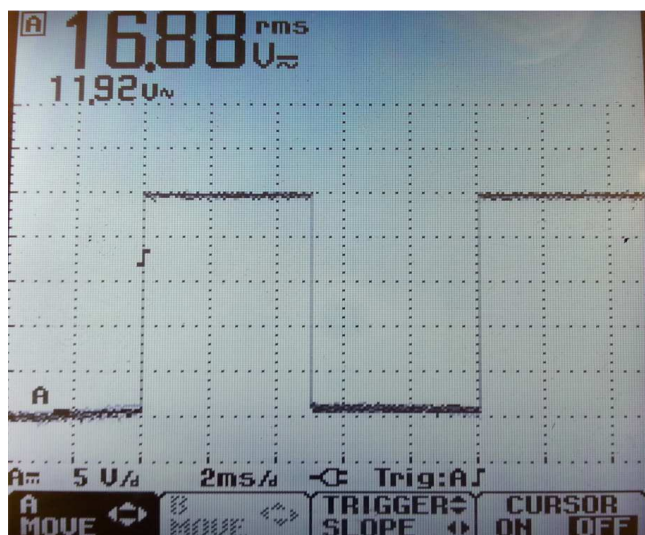
$$U = U_{max} * \sqrt{\frac{x}{100}} = 24 \text{ V} * \sqrt{\frac{25 \%}{100 \%}} = 12 \text{ V} \quad (2)$$

jossa U on ohjauksen jännite (V), U_{max} on lähdön maksimijännite (V) ja x on käytettävä pulssisuhde (%).

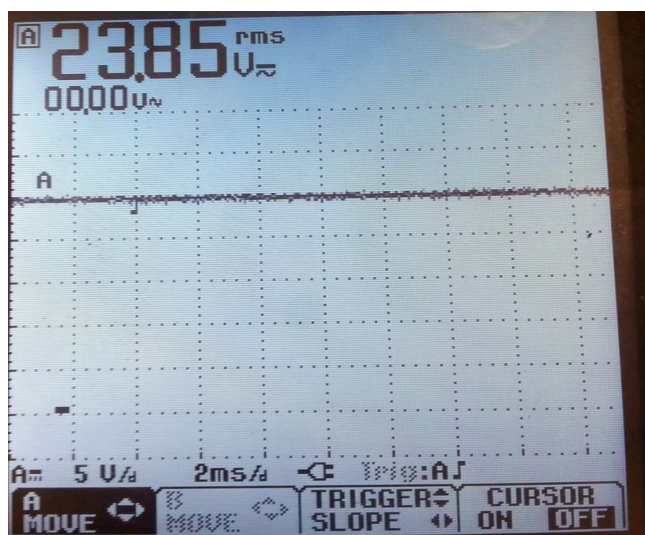
PWM-funktion toimivuutta testattiin mittaamalla logiikan lähtöä oskilloskoopilla ja syöttämällä logiikassa pulssisuhteen operandiin eri arvoja. Kuvista 13–15 nähdään jännitteen käyttäytyminen pulssisuhteen mukaan. Y-akselilla oskilloskoopissa on jännite, ja akseli on säädetty asetukseen 5 voltia/ruutu. X-akselilla on aika, jonka asetus on 2 ms/ruutu. Kuvissa ylävasemmalla on nähtävissä jännitteen lukema.



Kuva 13. Jännitteen kuvaaja 25 %:n pulssisuhteella (100 Hz)



Kuva 14. Jännitteen kuvaaja 50 %:n pulssisuhteella (100 Hz)



Kuva 15. Jännitteen kuvaaja 100 %:n pulssisuhteella (100 Hz)

4.2 Piirikaavio, osaluettelo ja sijoituskuvat

Järjestelmän piirikaavio (liite 2) ja sijoituskuvat (liite 4) piirrettiin Kymdatan CAD\$ Planner Electric -suunnitteluohjelmistolla. Osaluettelo löytyy liitteestä 3.

5 OHJELMISTOSUUNNITTELU

5.1 Ominaisuudet ja parametrit

Logiikkaohjelma tuli rakentaa niin, että se toteuttaa järjestelmältä vaaditut ominaisuudet ja parametrit. Ohjelman suunnitteluvaiheessa oli huomioitava, että suuri osa käyttöliittymään lisättävistä indikoinnoista ja toiminnoista oli rakennettava täysin logiikan varaan, sillä pienoismalliin sijoitettiin anturoinnin osalta vain kulma-anturi. Ilman anturointia toteutettavia visualisointeja olivat esimerkiksi propellin pyörintänopeus, teho, momentti sekä aluksen nopeus.

Järjestelmältä vaadittavat ominaisuudet ja parametrit:

- ajotavat: manuaaliajo, ECP-ajo ja automaattiajo (sekvensi)
- manuaali- ja automaattiajo follow-up-ohjauksella
- ECP-ajo ON/OFF-ohjauksella
- ohjauksehva seuraa potkurilaitetta ECP- ja automaattiajossa
- vikadiagnostiikka
- normaali kääntönopeus 1,5 rpm ($9^\circ/s$)
- ECP-ajon kääntönopeus 0,75 rpm
- kulmarajoituksen päällä ollessa kääntönopeus 0,5 rpm
- paikoitustarkkuus 1°
- kulmarajoitus ($\pm 20^\circ$) aluksen nopeuden ylittäessä 14 solmua (n. 26 km/h)
- käännön kiihdytysramppi 4 s
- käännön jarrutus lineaarisesti jäljellä olevan matkan (kulman) mukaan ($30^\circ - 0^\circ = 100-0\%$)
- käännön suunnanvaihto yli 265 asteen käännöissä
- kuvitteellisen aluksen nopeus 0–28 solmua
- propellien kuvitteellinen pyörintänopeus 0–150 rpm
- pääkoneen kuvitteellinen pyörintänopeus 0–750 rpm
- kuvitteellinen teho 0–7500 kW
- ohjauksehvan tuntokohdat ($0^\circ, \pm 90^\circ, 180^\circ$)

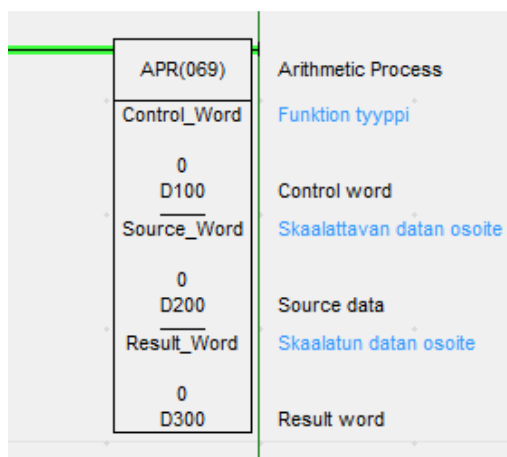
5.2 Funktiot

Logiikkaohjelma sisältää hyvin paljon arvojen skaalauksia sekä erilaisia sekvenssejä, joiden toteutukseen käytettiin CX-Programmerin valmiita APR(069)- ja SFT(010)-funktioita. Nämä funktiot ovat käytettävissä LD-kielen ohjelmoinnissa.

5.2.1 APR-funktio

APR(069)-funktio (Arithmetic Process) helpottaa ja nopeuttaa arvojen skaalauksia ja muuntoja LD-kielen ohjelmoinnissa erityisesti, kun skaalauksen kulmakerroin on vaihteleva. Käyttäjän määrittelemien lineaarimuunnoksien lisäksi sillä on mahdollista tehdä sini- ja kosinimuunnoksia. Funktio (kuva 16) sisältää kolme operandia (Omronin [www-sivut e.](http://www-omron.com)):

1. "Control word" määrittää skaalauksessa käytettävän funktion. Vaihtoehtoja ovat sinifunktio (0000_{hex}), kosinifunktio (0001_{hex}) tai lineaarinen skaalaus. Lineaarista skaalausta käytettäessä annetaan tälle operandille käyttäjän itse määrittelemä osoite logiikan muistista. Annettu osoite kertoo funktiolle, mistä osoitteesta alkaen funktion parametrit löytyvät.
2. "Source data" määrittää muistipaikan, jossa skaalattava arvo sijaitsee.
3. "Result data" määrittää muistipaikan, johon skaalattu data talletetaan.



Kuva 16. APR(069)-funktio

Kuvan 16 mukaiselle APR-lineaarifunktiolle täytyy lisäksi määrittää parametrit sen muistialueelle. Muistialueen ensimmäinen sana (kuvassa 16 D100) toimii funktion ohjesanana ja kertoo funktiolle lähtö- ja tulodatan pituudet ja tyyppin sekä parametrien määrän. Ohjesanan muodostuksessa tulee olla tarkka, sillä kaikki yhdistelmät eivät kelpaa (esim. etumerkillistä BCD-lukua ei voida muodostaa). Ohjesana muodostetaan seuraavasti (Omronin [www-sivut e.](http://www.sivut.e)):

Bit 15 = Input data format (0=binary, 1=BCD)

Bit 14 = Output data format (0=binary, 1=BCD)

Bit 13 = Source data form (0=directly source data, 1 = inverse $X_{max} - S$)

Bit 12 = 0

Bit 11 = Data type for source and result word (0=unsigned, 1=signed)

Bit 10 = Data length for source and result word (0=16-bit, 1=32-bit)

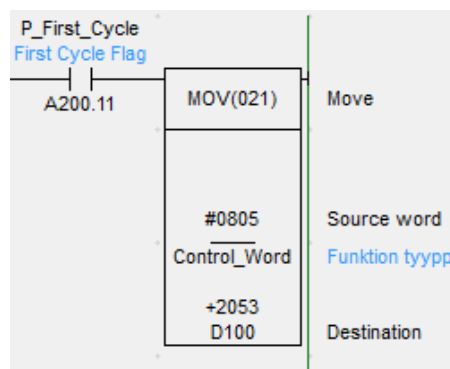
Bit 9 = Floating-point specification (0=integer data, 1=floating-point data)

Bit 8 = 0

Bit 0-7 = Number of coordinates minus one ($m - 1$, $1 \leq m \leq 256$).

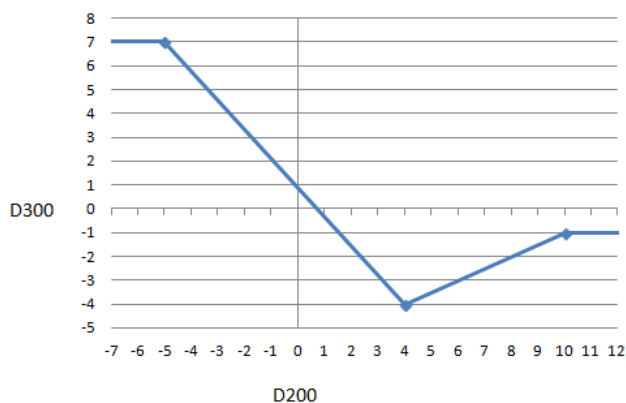
Ohjesanasta (= C) seuraavaan sanaan (C + 1) tulee siirtää koordinaatiston pisteen X_0 -arvo. Sitä seuraavaan sanaan (C + 2) siirretään koordinaatiston pisteen Y_0 -arvo. Tästä seuraavaan sanaan (C + 3) siirretään X_1 -arvo. Näin jatketaan, kunnes haluttu funktion kuvaaja on valmis.

Esimerkiksi kuvan 16 funktion lähtö- ja tulodata ovat tyypiltään 16-bittisiä etumerkillisiä integer-lukuja ja funktion halutaan sisältävän 6 koordinaattia (3 x:n arvoa, 3 y:n arvoa). Tällöin ohjesanaksi muodostuu 0805_{hex} ($0000\ 1000\ 0000\ 0101_{\text{bin}}$), joka siirretään sanaan D100 (kuva 17). Tämän jälkeen siirretään halutun kuvaajan toteuttavat parametrit ohjesanan jälkeisiin



Kuva 17. Funktion ohjesanan siirto

muistipaikkoihin: D101 (X_0) = -5, D102 (Y_0) = 7, D103 (X_1) = 4, D104 (Y_1) = -4, D105 (X_2) = 10, D106 (Y_2) = -1. Annetuilla arvoilla skaalauksesta muodostuu kuvan 18 mukainen kuvaaja.



Kuva 18. APR-funktion muodostama kuvaaja

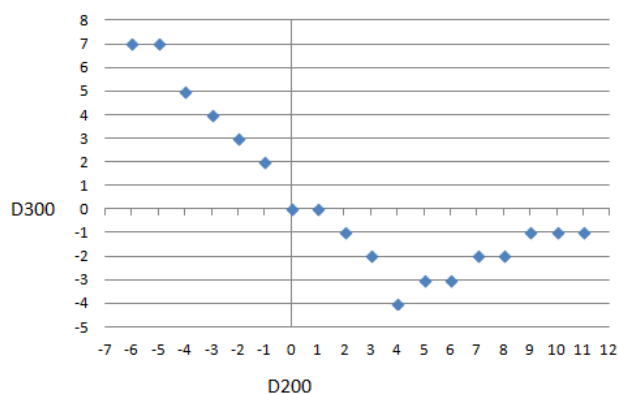
$$y = 7, x < -5$$

$$y = -\frac{11}{9}x + \frac{8}{9}, -5 \leq x \leq 4$$

$$y = \frac{1}{2}x - 6, 4 < x \leq 10$$

$$y = -1, x > 10$$

16-bittistä integer-lukua käytettäessä on huomattava, että APR-funktio ei käytä pyöristystä, vaan antaa laskun tuloksen kokonaisluvun ja hävittää jakojäännöksen. Vaikka APR-funktio käyttää annetuilla parametreilla ehtonaan yllä mainittuja funktioita, kuvaajan lähemmällä tarkastelulla huomataan, ettei se ole täysin lineaarinen (kuva 19). Kokonaisluvuilla todellinen kuvaaja on aina askelmainen.



Kuva 19. APR-funktion pistekaavio

5.2.2 SFT-funktio

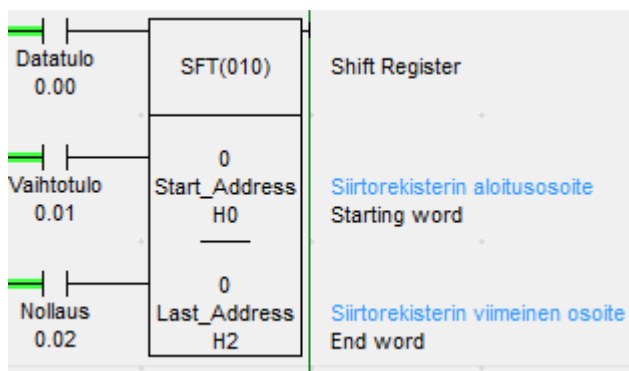
SFT(010)-funktiolla (Shift Register) saa luotua nopeasti siirtorekisterirungon, joten funktiota on hyvä käyttää sekvenssiohjelmoinnissa. Funktio sisältää kaksi operandia ja kolme tuloa (Omronin [www](http://www.omron.com)-sivut e):

Operandit:

1. "Starting word" määrittää siirtorekisterin ensimmäisen sanan. Muistialueena on hyvä käyttää paristovarmennettua H-apumuistialuetta, jolloin askel säilyy sähkökatkonkin yli.
2. "End word" määrittää, mihin sanaan rekisteri päättyy (oltava ensimmäisen operandin muistialueelta suurempi tai yhtäsuuri sanaosoite). Näin ollen se määrittää myös rekisterin rungon pituuden. Kuvassa 20 annetuilla operandeilla rekisteri sisältää kolme (H0, H1, H2) sanaa, jolloin askeleiden määrä on max. 48 (16 bit/sana).

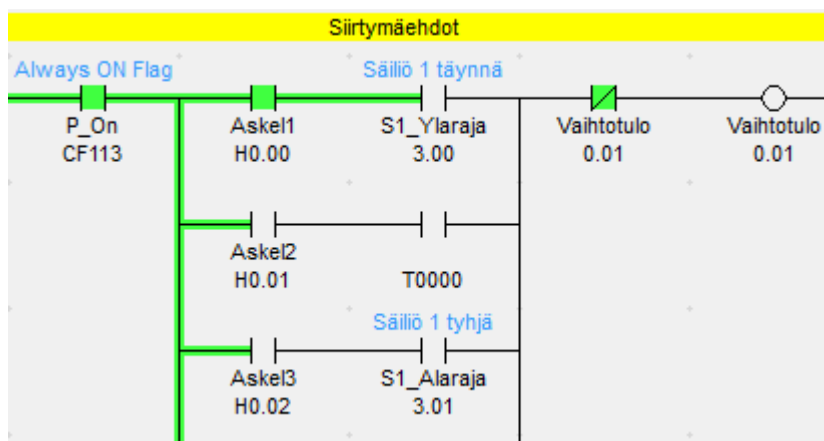
Tulot:

1. Ylin tulo siirtää tilansa jokaisella kellotuksella rekisterin ensimmäiseen bittiin (kuvassa 20 bittiin H0.00). Tällä voidaan esimerkiksi käynnistää askellusajo tai aloittaa toinen sekvenssi (samassa rekisterirungossa) edellisen ollessa vielä käynnissä. Sekvenssi voidaan käynnistää myös keskeltä esimerkiksi siirtämällä arvo 64 (0100 0000_{bin}) sanaan H1, jolloin sekvenssi käynnistyy askeleesta 23 (16 bit + 7 bit).
2. Keskimäinen tulo on kellotustulo eli jokaisella nousevalla reunalla se siirtää rekisterin jokaista bittiarvoa yhden eteenpäin.
3. Alin tulo on reset-tulo, joka aktivoituessaan nolaa koko rekisterin.

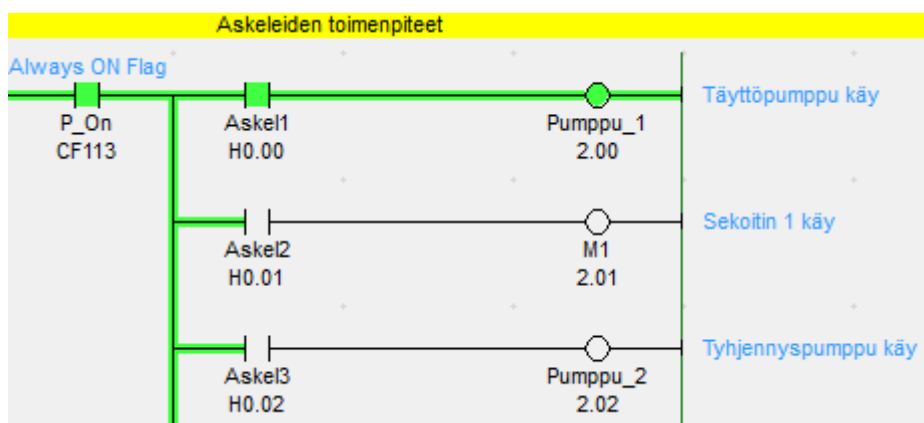


Kuva 20. SFT(010)-funktio

Siirtorekisterirungon jälkeen kootaan siirtymäehdot (kuva 21) ja askeleiden toimenpiteet (kuva 22).



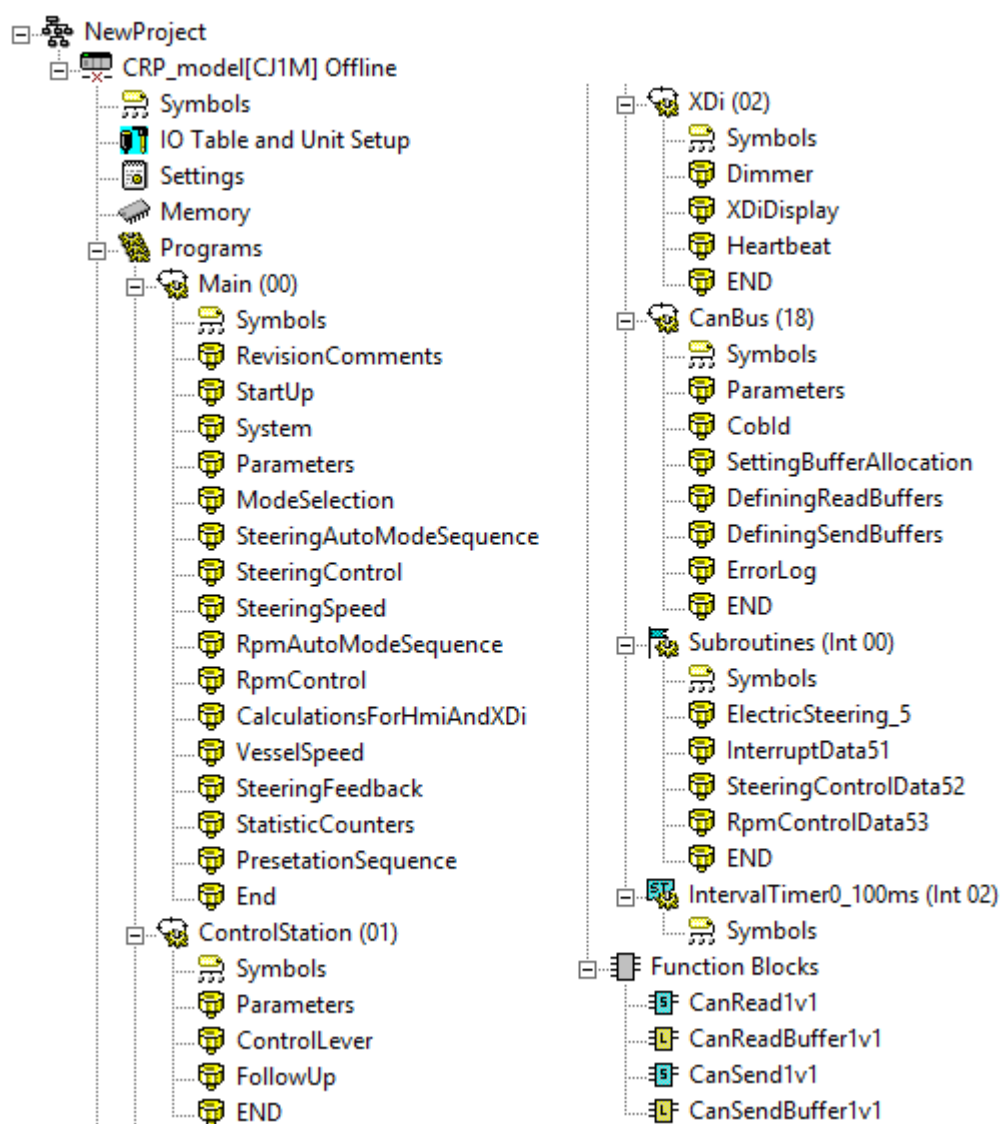
Kuva 21. Esimerkki siirtymäehtoien toteutuksesta



Kuva 22. Esimerkki sekvenssin toimenpiteistä

5.3 Logiikkaohjelma

Ohjelmasuunnittelu aloitettiin ottamalla yrityksen erään toimitusprojektin logiikkaohjelma pienoismallin logiikkaohjelman pohjaksi. Toimitusprojektit eroavat järjestelmältään hyvin paljon pienoismallin järjestelmästä, joten logiikkaohjelmaa jouduttiin muuttamaan paljon. Tarvittavien muutoksien suunnittelu aloitettiin tutustumalla paremmin CX-Programmerin ohjelmointiympäristöön sekä pohjaksi otetun logiikkaohjelman rakenteeseen ja toiminnallisuuksiin.



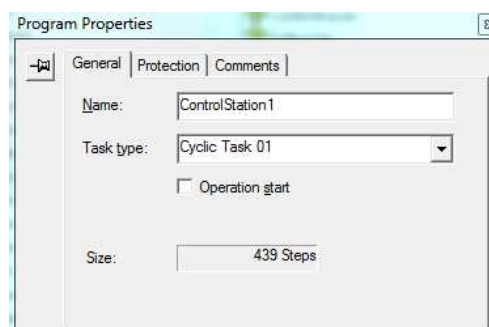
Kuva 23. Pienoismallin logiikkaohjelman lopullinen rakenne

CX-Programmerissa ohjelmarakenne (kuva 23) on kolmetasoinen. Projekti koostuu ohjelmista (taskeista), joista jokainen on joko syklinen ohjelma (Cyclic task) tai erik-

seen kutsuttava keskeytysohjelma (Interrupt task). Jokainen relekaaviopohjainen ohjelma on jaettu sektioihin (blokkeihin) ja jokainen sektio on jaettu virtapiireihin, joissa itse ohjelmakoodi sijaitsee. Pienoismallin ohjelman muokkaus aloitettiin ohjelman rakennemuutoksilla. Tavoitteena oli saada mahdollisimman selkeä rakenne pienoismallin ohjelmalle, jotta uuden ohjelmakoodin rakentaminen ja ryhmittely toiminnoittain olisi helppoa. Ensimmäisenä kaikki ylimääräiset pienoismallin toimintaan vaikuttamattomat ohjelmablokit poistettiin. Tämän jälkeen arvioitiin, mitä lisäyksiä logiikkaohjelma tarvitsee toteuttaakseen kaikki pienoismallilta vaaditut toiminnallisuudet. Nämä lisäykset ryhmiteltiin toiminnallisuuden mukaan ja jokaiselle ryhmälle luotiin pääohjelmaan (Main 00) oma blokki. Lopullinen logiikkaohjelma sisälsi neljä syklistä taskia, yhden normaalin keskeytystaskin ja yhden ajastetun keskeytystaskin.

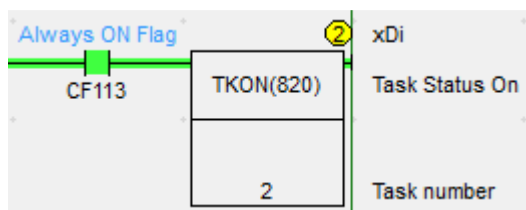
5.3.1 Sykliset ohjelmat

Logiikan pääohjelma (ohjelmanumero 00) on syklinen ohjelma, joka käynnistyy aina logiikan käynnistyessä. Logiikkaohjelmaan on kuitenkin mahdollista luoda lisää syklisiä ohjelmia. CJ1M-CPU2x-sarjan logiikoissa syklisiä ohjelmia voidaan lisätä 31 (ohjelmanumerot 01–31). Syklistä ohjelmaa luota-



Kuva 24. ”Operation start” -valinta

essa voidaan määrittää, että ohjelma on aina logiikan käynnistyessä automaattisesti mukana logiikan syklisessä kierrossa (kuva 24). Jos tätä määritystä ei tehdä, tulee jossain suoritettavassa ohjelmassa ennen kyseistä määrittämätöntä taskia kertoa logiikalle, otetaanko kyseinen ohjelma mukaan sykliseen kiertoon vai ei. Tämä tehdään käyttäen CX-Programmerin funktioita TKON(820) ja TKOF(821). Mikäli ohjelmankierto etenee sellaisen taskin kohtaan, jolle ei ohjelmassa ole vielä määritetty ”ON-” tai ”OFF-tilaa”, menee logiikka virhetilaan. Pienoismallin tapauksessa pääohjelman blokki ”StartUp” määrittää muut sykliset taskit mukaan logiikan ohjelmasuoritukseen (kuva 25). Ohjelman sykliset taskit ovat Main(00), ControlStation(01), XDi(02) sekä CanBus(18).



Kuva 25. XDi-taskin määrittäminen ohjelmakiertoon

- Main (00)

Main (00) on logiikkaohjelman pääohjelma. Se sisältää ohjelmablokit parametreille, kääntömoottorin ohjaukselle, propellimoottorin ohjaukselle, ajotavan valinnalle, takaisinkytkennälle sekä erilaisille sekvensseille ja laskuille.

1. Revision Comments -blokki sisältää selitykset ja päiväykset ohjelmaan tehdyille muutoksille. Blokki ei sisällä ohjelmakoodia.
2. Start Up -blokki määrittää ohjelmasuoritukseen mukaan otettavat sykliset taskit. Se suorittaa logiikan ensimmäisellä syklillä myös alirutiinit (ks. s. 36), jotka sisältävät funktioiden tarvitsemia parametrejä.
3. System-blokki sisältää itse tehtyjä pulssiajastimia ja laskureita, joiden toiminnallisuutta voidaan hyödyntää muualla ohjelmassa.
4. Parameters-blokki siirtää tarvittavien parametrien arvot ohjelman muuttujiin, joita käytetään itse ohjelmassa. Näin parametrejä muuttaessa riittää, että muutokset tehdään vain tähän blokkiin. Blokkissa määritettäviä parametrejä ovat esimerkiksi max. kääntönopeus, kiihdytysrampin pituus sekä vähimmäiskulma täydelle nopeusohjeelle.
5. Mode Selection -blokki määrää käytettävän ajotavan (manuaali-, ECP- vai automaattiajotila) ja toteuttaa ajotavanvaihdot käyttäjän pyynnöstä.
6. Steering Auto Mode Sequence -blokki on automaattiajotilan apuohjelma. Blokki suorittaa 32 askelta pitkää sekvenssiä, joka antaa erilaisia kulmareferenssejä potkurin asennolle.

7. Steering Control -blokin tehtävä on nopeusohjeen laskeminen ja kääntösuunnan määrittäminen. Blokki vertaa (manuaali- ja automaattitilassa) käytetyn ajotavan antamaa kulmareferenssiä potkurilaitteen todelliseen kulmaan. Kulmaeron mukaan blokki määrittää nopeusohjeen. Alle asteen erolla nopeusohje on 0 % ja yli 30 asteen erolla nopeusohje on 100 %. Muutoin nopeusohje 0–100 % on lineaarisesti verrannollinen kulmaeroon $0^\circ - 30^\circ$. ECP-ajolla nopeusohje on aina 50 %.
8. Steering Speed -blokki laskee kääntömoottoria ohjaavan pulssisuhteen mukaan arvion potkurin kääntönopeudesta.
9. Rpm Auto Mode Sequence -blokki on automaattiajotilan apuohjelma. Blokki suorittaa 16 askelta pitkää sekvenssiä, joka antaa erilaisia referenssejä propellien pyörimisnopeudelle.
10. Rpm Control -blokki vertaa propellien pyörintänopeuden ohjauksen suuruutta ja käytetyn ajotavan antamaa referenssiarvoa (= nopeusohjetta). Referenssin muuttuessa blokki muuttaa ohjauksen (0–100 %) uuden referenssin mukaiseksi (ohjauksen muutosnopeus on rajoitettu kiihdytys- ja jarrutusrampilla). Testaamalla todettiin moottorin tarvitsevan vähintään 2 voltin jännitteen propellien liikkeellelähtoon sekä sen, että jännitevälillä 22–24 V ei tapahtunut enää muutosta pyörintänopeuteen. Ohjauksen 1–100 % piti siis vastata moottorille menevää jännitettä 2–22 V. Moottorin toimiessa kuormana pulssisuhte ja jännite eivät enää noudattaneet kaavaa 2. Nyt 2 V jännitteen aikaansaamiseksi pulssisuhteen tuli olla 2 % ja 22 V jännitteen aikaansaamiseksi 60 %, joten blokki skaalaa ohjausarvot 1–100 % PWM-funktiolle arvoihin 20–600_{dec}.
11. Calculation For HMI And XDi -blokki sisältää erilaisia skaalauksia ja laskuja käyttöliittymän ja kulmanäytön indikoiteja varten.
12. Vessel Speed -blokki laskee kuvitteellisen aluksen nopeuden. Nopeudeksi valittiin 0–28 solmua. Blokki antaa APR-funktiolla potkurin kulman mukaan suhdearvon, joka kerrotaan propellien pyörintänopeuden ohjausarvolla.

13. Steering Feedback -blokki skaalaa CAN-väylältä tulleen kulmatiedon muun ohjelman ymmärtämään muotoon. Kulma-anturin lähettämä data $0-4095_{dec}$ (vastaa anturin kulma-arvoa $0^\circ - 360^\circ$) skaalautuu lineaarisesti välille $+180^\circ - (-180^\circ)$. Skaalauksen lisäksi blokki kuuntelee anturin sykettä ("heartbeat") väylällä ja antaa hälytyksen, jos anturi katoaa väylältä. Laitteen ja ohjauslaitikon suuntauksessa tulee anturin kulma-arvo nollata. Blokki hoitaa anturin nollauksen käyttäjän pyynnöstä.
 14. Statistic Counters -blokki sisältää erilaisia laskutoimituksia järjestelmän käytöstä. Esimerkiksi se laskee käyttötunnit ja minuutit propellin eri nopeusalueilla.
 15. Presentation Sequence -blokki sisältää 7 askelta pitkän sekvenssin. Käyttöliittymään rakennettu Steerprop-esitys käyttää sekvenssin askeleita esityksen sivunumeron määräämiseen.
- Control Station (01)

Control Station on syklinen taski, jonka blokit hoitavat ohjauskahvaan liittyvät toiminnot. Taski hakee kommunikointiyksikön ohjauskahvalle määritettyjen vastaanottopuskureiden tallentamat, CAN-väylältä saapuneet, datat logiikan muistista ja tallentaa ne logiikkaohjelmassa niille määritettyihin symboloituihin muuttujiin. Tämän jälkeen taski käy läpi kahvalta saapuneen tilasanan sekä hoitaa hälytykset, varoitukset ja toimenpiteet, jos tilasanasta löytyy hälytyksiä (esim. lämpötilavaroitus tai kahvan anturin vikahälytys). Molemmat kahvan piirit (kulma- ja nopeuspiiri) lähettävät ja vastaanottavat tietoa, joten molemmille piireille on määritettävä omat puskurinsa sekä vastaanottoon että lähetykseen.

Omronin CAN-kommunikointiyksikkö varaa trigger-bitin jokaiselle määritetylle lähetys- ja vastaanottopuskurille. Joka kerta, kun vastaanottopuskuriin saapuu viesti, on trigger-bitti "1-tilassa" yhden syklin ajan. Ohjauskahvan tapauksessa laitteen sykkeen kuuntelu on korvattu seuraamalla logiikkaohjelmassa kahvan vastaanottopuskureiden trigger-bittejä. Puskurin trigger-bitin ollessa liian pitkään

"0-tilassa" antaa logiikka hälytyksen kahvan katoamisesta väylältä. Seuraamalla laitteen datalle määritetyn vastaanottopuskurin trigger-bittiä, saadaan vähennettyä ylimääräisten vastaanottopuskureiden määrää, sillä "heartbeatin"-kuuntelu perustuu myös trigger-bittiin. Jokaisen laitteen sykkeen kuuntelulle määritetään oma vastaanottopuskuri, jonka trigger-bitin jokaisella nousevalla reunalla varmistutaan laitteen olemassaolosta väylällä.

Control Station -taski sisältää myös kahvan moottorien referenssin ja tilatietojen siirtämisen niille määritettyjen lähetyspuskureiden muistialueille. Tilasana muodostuu 16 bitistä. Tärkeimmät tilasanan bitit ovat sen ensimmäinen (= stepper motor enabled / disabled) ja toinen (= master / slave) bitti. Ensimmäinen bitti määrittää piirin moottorin toimintaan, jolloin esimerkiksi kahvan moottoreille SDO-viestein määritetyt tuntokohdat ovat aktiivisia. Tilasanan toinen bitti määrittää, seuraako kahvan piiri sille lähetettyä referenssiarvoa ("0-tila" = orjalaite).

- XDi (02)

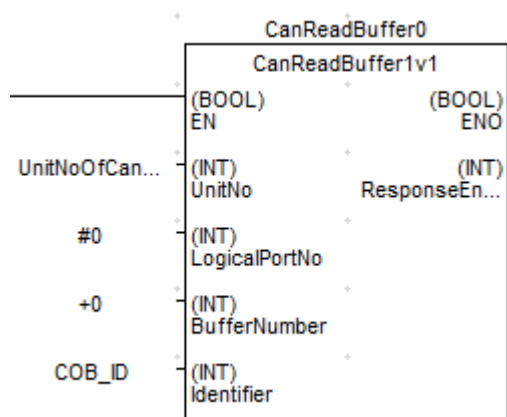
XDi on syklinen taski, joka rakennettiin logiikkaohjelmaan viimeisenä. Jokaista kulmanäytön indikaattoria varten määritettiin logiikkaan yksi lähetyspuskuri. XDi-taski siirtää jokaisen indikaattorin tarvitseman datan ja objektikirjaston osoitteen indikaattoria varten määritetyn lähetyspuskurin muistialueelle. Taski hoitaa myös kulmanäytön kirkkaudensäädön käyttäjän pyynnöstä. Kulmanäytön sykkeelle on määritetty oma vastaanottopuskuri, jonka trigger-bittiä taski kuuntelee varmistuakseen näytön olemassaolosta väylällä. Toisin kuin ohjauskahva (eri noodinnumero kulma- ja nopeuspiirillä), näyttö kuuntelee vain yhtä sanomanumeroa. Tästä syystä sille lähetetyn viestin yksilöivä koodi (objektikirjaston osoite) on oltava sisällytettynä CAN-kehysten dataosuuteen. Yksilöivä koodi määrittää, mille indikaattorille saapuvan kehysten tieto kuuluu. Ennen kuin XDi-taski saatiin rakennettua logiikkaohjelmaan, tutkittiin näytön CAN-ominaisuuksia sekä selvitettiin, millä tavalla eri indikaattorien tieto yksilöidään. Kulmanäyttöä ja sen ominaisuuksia on tutkittu tarkemmin luvussa 7.

- CanBus (18)

CanBus on syklinen taski, joka määrittää CAN-kommunikointiyksikölle (= väylämoduulille) sen käyttämät muistialueet sekä lähetys- ja vastaanottopuskurit.

1. Parameters-blokissa määritetään tunnistetietoja, joita väylämoduulin puskurit tarvitsevat väylän lukemista ja sinne lähetystä varten. Tällaisia tunnisteita ovat väylän komponenttien noodinumerot, PDO- ja SDO-tunnisteet sekä heartbeat-tunnisteet.
2. CobId-blokissa muodostetaan lähetystä ja vastaanottoa varten komponenttien COB-ID:t (= sanomanumerot), jotka määräytyvät komponentin noodinumeron ja käytettävän lähetysprotokollan tunnuksen mukaan (esim. PDO 1 + Node ID).
3. Setting Buffer Allocation -blokki kokoaa väylämoduulin tarvitsemat tiedot ja lähettää ne CX-Programmerin valmista CMND(490)-komentoa käyttäen väylämoduulille. Komennolla määritetään väylämoduulille sen lähetys- ja vastaanottopuskureiden maksimimäärä sekä niiden datan ja trigger-bittien muistialueet. Tässä työssä muistialueiden aloitusosoitteiksi määrättiin D10000 (lähetyspuskureiden data), D10200 (vastaanottopuskureiden data), CIO 4000 (lähetyspuskureiden trigger-bitit) ja CIO 4100 (vastaanottopuskureiden trigger-bitit). Jokainen puskurin varaa numeronsa perusteella 5 sanaa väylämoduulille määritetyltä datamuistialueelta. Näiden viiden sanan ensimmäinen sana kertoo datan pituuden ja loput neljä sanaa on varattu itse datalle (CAN-viestikehyksen dataosuus voi olla enintään 8 tavua). Jokainen puskurin varaa myös trigger-alueelta yhden bitin numeronsa perusteella. Esimerkiksi vastaanottopuskuri, jonka numeroksi määritetään 2, käyttää vastaanotetun datan tallennukseen nyt logiikan osoitteita D10210 - D10214. Puskurin käyttämä trigger-bitti on CIO 4100.02.
4. Väylämoduulin käyttämien osoitteiden määrittämisen jälkeen tulee moduulille määrittää lähetys- ja vastaanottopuskurit. Jokaisen puskurin määrittäminen hoidetaan lähettämällä luotavan puskurin parametrit väylämoduulille. Puskureiden

luonti on hoidettu sekvenssillä yksi puskuri/sykli. Puskureiden määrittämiseen löytyy toimitusprojektien logiikkaohjelmista valmiit toimilohkot CanReadBuffer1v1 (kuva 26) ja CanSendBuffer1v1 (kuva 27), jotka kokoavat puskurin määrittämisessä tarvittavat tiedot sekä lähettävät ne CMND(490)-komennolla väylämoduulille. Defining Read Buffers -blokissa määritetään valmista toimilohkoa käyttäen väylämoduulin vastaanottopuskurit. Toimilohko (kuva 26) tarvitsee puskurin määrittämistä varten moduulinumeron, porttinumeron, puskurinumeron sekä tunnisteeseen.

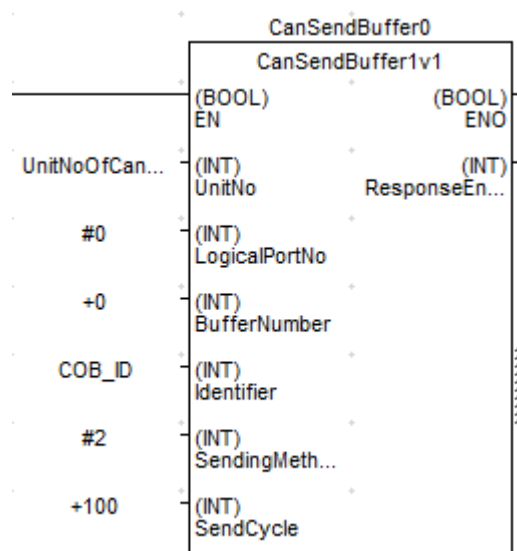


Kuva 26. Vastaanottopuskurin määrittävä toimilohko

Moduulinumero tarkoittaa kommunikointiyksikölle mekaanisesti asetettua sen yksilöivää numeroa, joka määrätään moduulissa olevan rullan avulla. Portti tarkoittaa viestin käyttämää logiikan sisäistä porttia. Puskurille määritettävä numero sen sijaan yksilöi kyseisen puskurin. Se määrää, mihin logiikan osoitteisiin puskuri tallettaa vastaanottamansa tiedot. Tunniste tarkoittaa kuunneltavan laitteen COB-ID:tä.

5. Defining Send Buffers -blokissa määritetään lähetyspuskurit valmista toimilohkoa hyödyntäen. Toimilohko tarvitsee puskurin määrittämistä varten moduulinumeron, porttinumeron, puskurinumeron, tunnisteeseen, lähetystavan sekä lähetysyökin ajan. Lähetyspuskuria määritettäessä tunnisteella tarkoitetaan vastaanottavan laitteen COB-ID:tä. Lähetystapa määrittää, mitä tulee tapahtua, jotta viesti lähetetään puskurista väylään. Kuvassa 28 on esitetty eri lähetystavat sekä niiden koodit. Lähetysyöki tarkoittaa aikaa, miten usein viesti lähetetään, jos lähetystapa on valittu syklisteksi. Trigger-lähetystavalla sen si-

jaan saadaan lähetettyä viesti väylään vain vaikuttamalla puskurin triggerbittiin. Toimilohko (kuva 27) kokoaa tiedot ja lähettää ne väyläyksikölle CMND(490)-komentoa käyttäen.



Kuva 27. Lähetyspuskurin määrittävä toimilohko

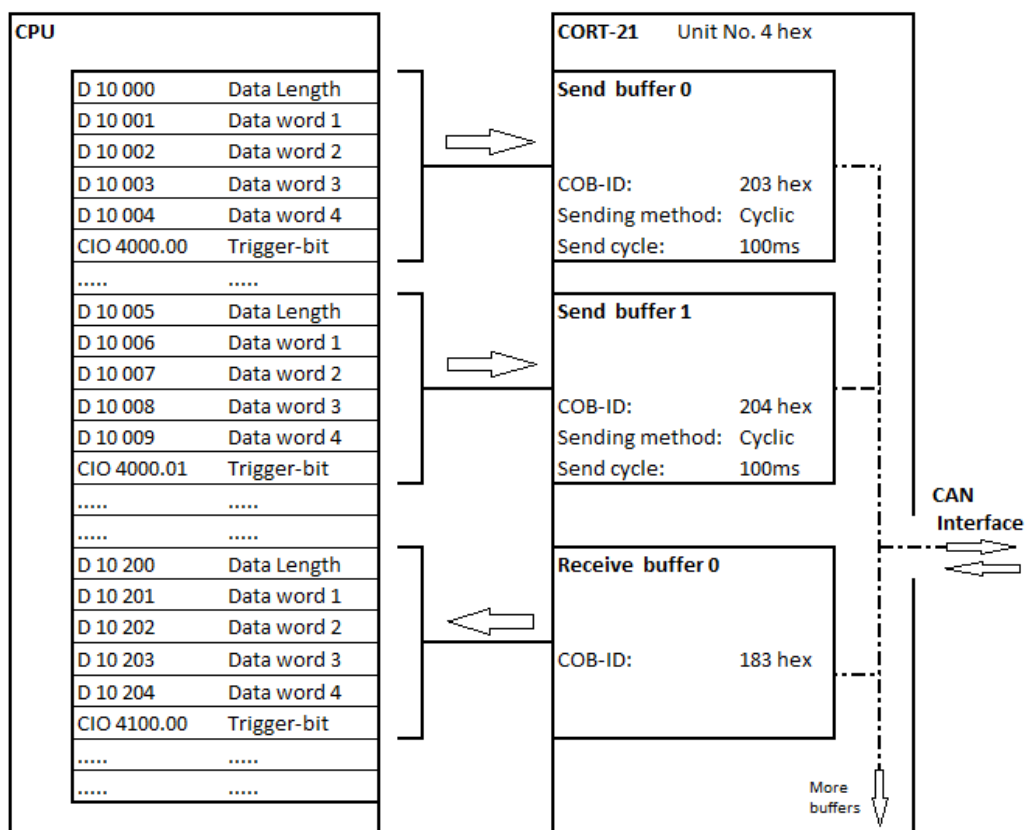
Method of sending:

#0001 _h	Message will be send on triggering the corresponding flag in the send trigger area.
#0002 _h	Message will be send cyclic. The cycle time is set in milliseconds.
#0004 _h	Message will be send on change of data.

Kuva 28. CAN-viestin lähetystavat logiikasta (Omron 2009, 57)

6. ErrorLog -blokki suorittaa vikakoodien haun väylämoduulista. Moduuli kykenee tallentamaan 15 edellistä virhetilannekoodia, jotka voidaan hakea moduulista CMND(490)-komentoa käyttäen.

CanBus-taski suorittaa ohjelmakoodiansa vain hetkellisesti logiikan käynnistyessä, sillä väylämoduulille on määritettävä sen käytettävissä oleva osoitteisto sekä sen lähetys- ja vastaanottopuskurit vain kerran. Pienoismallin ohjelmassa määritetään 4 vastaanottopuskuria ja 8 lähetysspuskuria, joten väylämoduulin konfigurointi kestää 13 sykliä (ensimmäisellä syklillä määritetään osoitteisto, minkä jälkeen määritetään yksi puskuri/sykli). Puskureiden konfiguroinnin jälkeen väylämoduuli päivittää ja hakee datansa suoraan logiikan muistista (kuva 29).



Kuva 29. Lähety- ja vastaanottopuskurien käyttämät muistialueet.

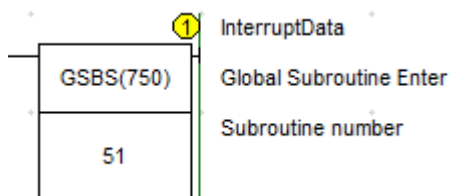
5.3.2 Keskeytysohjelmat

CJ1M-CPU2x-sarjan logiikoihin on mahdollista tehdä 256 (ohjelmanumerot Int 00 - Int 255) keskeytysohjelmaa, joista jokainen voidaan tarpeen tullen määrittää myös ylimääräiseksi sykliseksi ohjelmaksi. Keskeytystaskit, jotka määritetään sykliseksi, eivät sisällä kaikkia samoja ominaisuuksia kuin taskit, jotka ovat valmiiksi syklisiä. Keskeytystaskin voi myös määrittää ajastetuksi keskeytystaskiksi, I/O-keskeytystaskiksi, ulkoiseksi keskeytystaskiksi tai jännitekatkon keskeytystaskiksi. Nämä taskit suoritetaan aina heti niille määritettyjen ehtojen täytyessä. Ohjelma siis hyppää normaalista ohjelmakierrosta keskeytysohjelmaan, suorittaa ohjelman ja palaa takaisin normaaliin ohjelmakiertoon. Ajastettu keskeytystaski suoritetaan sille määritetyn aikavälin välein (ohjelmanumerot Int 02 ja Int 03). I/O-keskeytystaski suoritetaan, kun sille määritetty tietty tulobitti aktivoituu logiikkaan lisättyssä keskeytysmoduulissa (ohjelmanumerot Int 100 - Int 131). Jännitekatkon sattua logiikka pysyy hengissä n. 10 ms, jonka aikana logiikka suorittaa jännitekatkon keskeytystaskin (ohjelmanumero Int 01). Muut keskeytystaskit ovat normaaleita keskeytystaske-

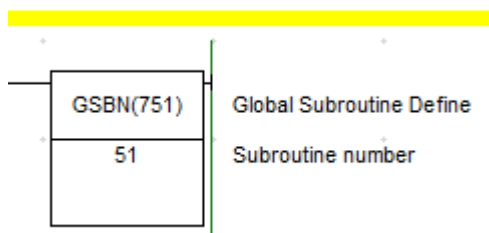
ja, jotka saadaan aktivoitua logiikkaan lisättävän erikois-I/O-yksikön tai väyläyksikön toimesta. Pienoismallin ohjelmassa on kaksi keskeytystaskia, joista toinen on ajastettu keskeytystaski ja toinen normaali keskeytystaski.

- Subroutines (Int 00)

Subroutines-taski on normaali keskeytystaski, jolle ei ole määritetty mitään keskeytystä aktivoivaa ulkoista komentoa. Taskia ei aktivoida milloinkaan, sillä sen alle on kerätty blokkeja, joita halutaan suorittaa eri aikaan ohjelmakierron aikana. Tällaiset blokit ovat alirutiineita. Ohjelmassa alirutiinin suorituskutsu hoidetaan GSBS(750)-komennon (kuva 30) avulla, jolloin ohjelma hyppää kutsussa määritettyyn alirutiiniin, suorittaa alirutiinin ja palaa takaisin normaaliin ohjelmakiertoon (jatkaa alirutiinin kutsukohdasta eteenpäin). Alirutiini-ohjelma alkaa aina alirutiinin yksilöivällä GSBN(751)-komennolla (kuva 31). Komennon jälkeen on itse ohjelmakoodi. Alirutiini lopetetaan aina GRET(752)-komentoon (kuva 32). Pienoismallin Subroutines-taski sisältää neljä alirutiinia.



Kuva 30. Alirutiinin no. 51 kutsu



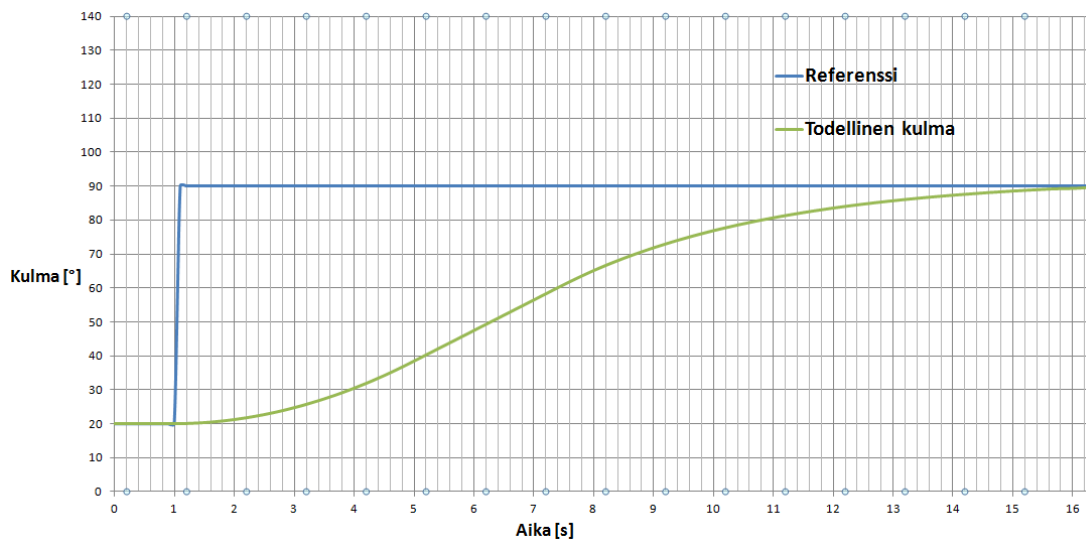
Kuva 31. Alirutiinin no. 51 aloituskäsky



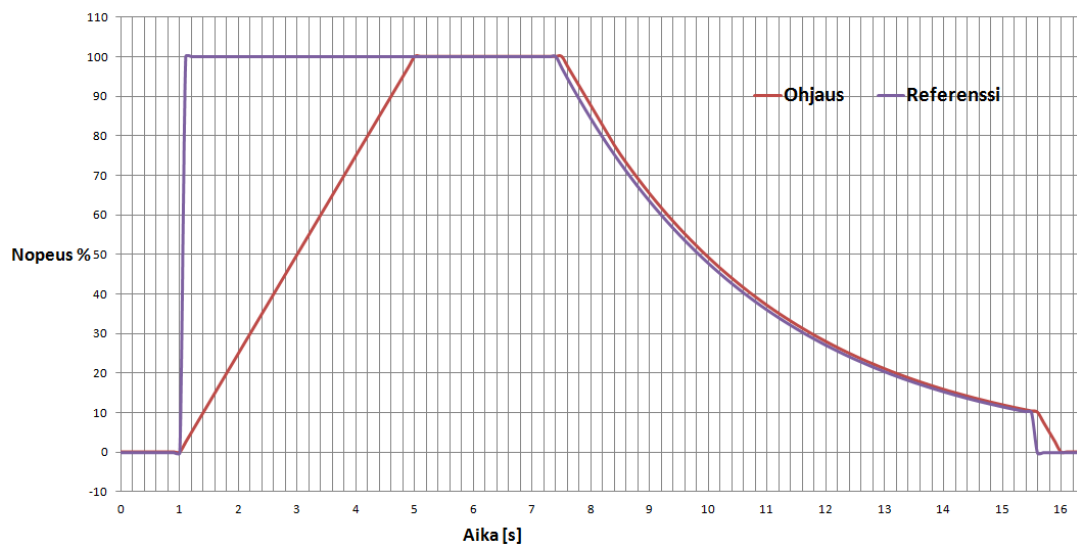
Kuva 32. Alirutiinin lopetuskäsky

1. Electric Steering -alirutiinia kutsutaan logiikan jokaisella syklillä. Blokki hoitaa kääntömoottorin ohjausarvon skaalauksen. Testaamalla todettiin kääntömoottorin tarvitsevan vähintään 2 V jännitteen liikkeelle lähtöön. Samalla tutkittiin, millä jännitteellä moottorin pyörintänopeus on 1,5 rpm. Tulokseksi saatiin 6,5 V. Moottorin toimiessa kuormana pulssisuhde ja jännite eivät noudata enää kaavaa 2. Nyt 2 V jännitteen aikaansaamiseksi pulssisuhteen tuli olla 0,7 % ja 6,5 V jännitteen aikaansaamiseksi 5,6 %. Tämän mukaan blokki skaalaa ohjausarvon 1–100 % PWM-funktiolle arvoihin $7-56_{dec}$.
 2. Interrupt Data -alirutiini suoritetaan vain logiikan ensimmäisellä syklillä. Blokki määrittää ajastetun keskeytystaskin ajan käyttäen MSKS(690)-komentoa. Komennon ensimmäiselle operandille annetaan määritettävän taskin tunnus ja toiselle operandille aika, kuinka usein taski suoritetaan.
 3. Steering Control Data suoritetaan vain logiikan ensimmäisellä syklillä. Blokki määrittää käännön ohjauksessa käytettäville APR-funktioille parametrit.
 4. Rpm Control Data suoritetaan vain logiikan ensimmäisellä syklillä. Blokki määrittää propellien ohjauksessa käytettäville APR-funktioille parametrit.
- Interval Timer (Int 02)

Interval Timer on ajastettu keskeytystaski, jonka suoritusväliksi on määritetty 100 ms. Taski vertaa käännön nopeusohjetta todelliseen kääntönopeuteen ja suorittaa tarvittavat kiihdytykset ja hidastukset. Taski hoitaa kokonaan käännön kiihdytysrampin lisäksi ohjausnopeutta jokaisella suorituskerrallaan. Jarrutukseen taski osallistuu vasta, kun matkan mukaan määräytyvä nopeusohje vähenee liian nopeasti (taski siis tarvittaessa hidastaa jarrutusramppia). Kääntömoottorin ohjaus tapahtuu siis kolmessa eri osassa. Steering Control -blokki määrittää nopeusohjeen ja kääntösuunnan, Interval Timer -taski hoitaa ohjausarvon rampituksen nopeusohjeen mukaiseksi ja Electric Steering -alirutiini skaalaa lopuksi ohjausarvon PWM-funktiolle. Potkurilaitteen kulman ohjearvon muuttuessa esimerkiksi 70 astetta blokit tuottavat kuvien 33 ja 34 mukaisen ohjauksen.



Kuva 33. Potkurilaitteen kulma ajan funktiona (70 asteen käännös)



Kuva 34. Kääntönopeus ajan funktiona (70 asteen käännös)

6 CAN-VÄYLÄ

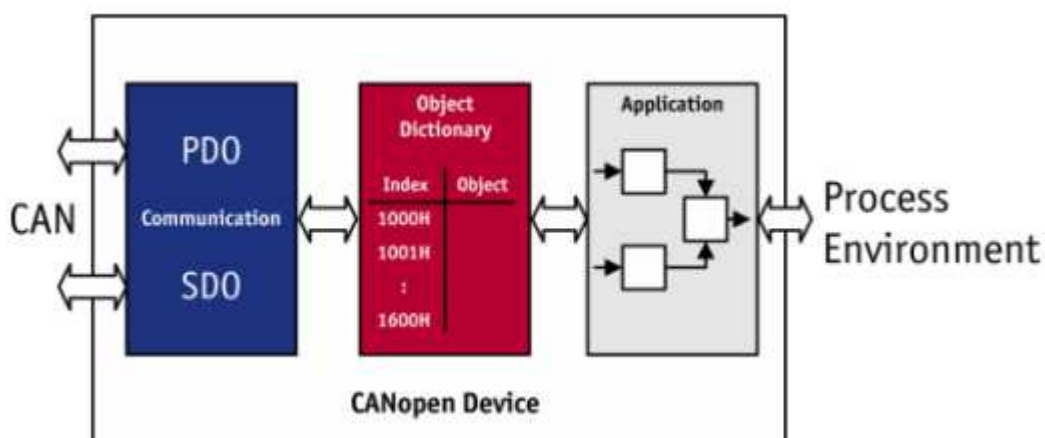
6.1 CAN-protokolla

CAN-väylä on alun perin autoteollisuuden suunniteltu sarjamuotoinen kenttäväylä, jolla on pyritty vähentämään johdotuksen määrää komponenttien lisääntyessä. Nykyään sitä käytetään myös laajalti automaatiassa (esim. koneet ja meriteollisuus) sen joustavuuden ja vikasietoisuuden vuoksi. CAN-protokollan standardi ISO-11898 määrittää vain fyysisen kerroksen ja siirtoyhteyskerroksen (Saha 2005). Tästä syystä sille on kehitetty lisämäärittelyjä, kuten DeviceNet ja CANopen.

CAN-väylän tiedonsiirto tapahtuu kahdella johtimella CAN_H ja CAN_L. Kun väylässä lähetetään looginen nolla, CAN_H-johtimen jännite nousee 3,5 volttiin ja CAN_L-johtimen jännite laskee 1,5 volttiin. Tästä syystä looginen nolla on CAN-väylässä dominoiva tila. Loogista ykköstä lähetettäessä molempien johtojen jännite on 2,5 V maahan nähden. Väylässä käytetään "NRZ-bit-stuffing"-johtokoodausta, jossa kahden peräkkäisen saman bitin välillä ei tapahdu tilanmuutosta. Väylän riittävä tahdistus hoidetaan viiden peräkkäisen saman bitin jälkeisellä vastakkaisella tahdistusbitillä. CAN-väylän komponentteja kutsutaan solmuiksi ja jokaisella solmulla on itsensä yksilöivä tunniste. Tunnisteen standardipituus on 11-bittistä, mutta CAN-protokollalle on määritetty myös 29-bittinen tunniste. Jokainen solmu kuuntelee väylää ja voi aloittaa lähetyksen vain, kun väylä on tyhjillään. Väylänvarausmekanismi CSMA/CD+AMP pitää huolen törmäyksen sattuessa, että suuremmalla prioriteetillä (= pienemmällä tunnisteella eli sanomanumerolla) sanomaa lähettävä komponentti saa jatkaa lähetystään. Tällöin pienemmän prioriteetin sanoman lähettäjä huomaa väylällä dominoivan nollatilan ja lopettaa lähetyksen. Väylän max. pituus vaihtelee 25 m:n (1 Mbit/s) ja monen kilometrin (10 kbit/s) välillä käytettävän siirtonopeuden mukaan. Yli kilometrin väylällä suositellaan käytettäväksi toistinta, mutta tällöin väylä menettää osan ominaisuuksistaan, sillä sanomankuittausmekanismi ei toimi toistimen takaa. Väylä päätetään molemmista päistä 120 Ω päätevastukseen heijastuksien estämiseksi. (Saha 2005.)

6.2 CANopen

CANopen on ylemmän tason protokolla, joka määrittää CAN-väylän sovelluskerroksen. Tärkein CANopenin määrittämä asia on ns. objektikirjasto, joka luo rajapinnan CAN-väylän ja laitteen sovelluksen välille (kuva 35). Objektikirjasto sisältää solmun parametrit sekä prosessidatan eli se toimii solmun tietovarastona. Jokaisella objektikirjaston tiedolla on oma osoite, joka koostuu indeksistä (16-bittinen) ja aliindeksistä (8-bittinen). Objektikirjasto on jaettu eri osakokonaisuuksiin (kuva 36). (Saha 2006.)



Kuva 35. CANopen laitteen rajapinnat (CANopen Solutions www-sivut a)

Index (hex)	Object
0000	not used
0001-025F	Data Types
0260-0FFF	Reserved for further use
1000-1FFF	Communication Profile Area
2000-5FFF	Manufacturer Specific Profile Area
6000-9FFF	Standardized Device Profile Area
A000-AFFF	Reserved for further use

Kuva 36. Objektikirjaston jako (CANopen Solutions www-sivut a)

CANopen-verkkoon liitetyille solmuille tulee antaa yksilöllinen noodinumero (Node-ID) väliltä 1-127 (1_{hex} - $7F_{\text{hex}}$). 11-bittisen tunnisteiden/sanomameron nimitys on COB-ID, joka muodostuu sanomaprotokollan tunnisteesta ja solmun noodinumerosta (kuva 37). Käytetyin protokolla on PDO, jota käytetään solmujen välillä vaihdetta-

van prosessidatan siirtoon. CANopen määrittelee neljä eritasoista PDO-protokollan tunnistetta lähetykseen ja vastaanottoon. Tästä syystä suurin valta viestinkehyksen prioriteettiin on käytettävällä PDO:lla ja tämän jälkeen vasta laitteen noodinumerolla. Solmujen välisen tiedonsiirron takaamiseksi jokaisen PDO:n prioriteetti on suurempi kuin SDO:n. SDO on protokolla, jolla kysellään ja muutetaan CANopen-laitteen parametreja ja asetuksia, joten SDO-protokollan viestin saadessaan laite lähettää aina vastauksen väylälle. Muita protokollia ovat NMT node control (verkon hallinta, esim. solmun käynnistys), Sync (PDO-sanomien tahdistus), TimeStamp (synkronoi solmujen kellot), Emergency (solmun virhetila), NMT node monitoring (solmun syke ja tilatieto) sekä LSS (käytetään solmun noodinumeron ja siirtonopeuden selvittämiseen ja muuttamiseen, jos laitteesta ei saa esim. dip-kytkimillä niitä muutettua).

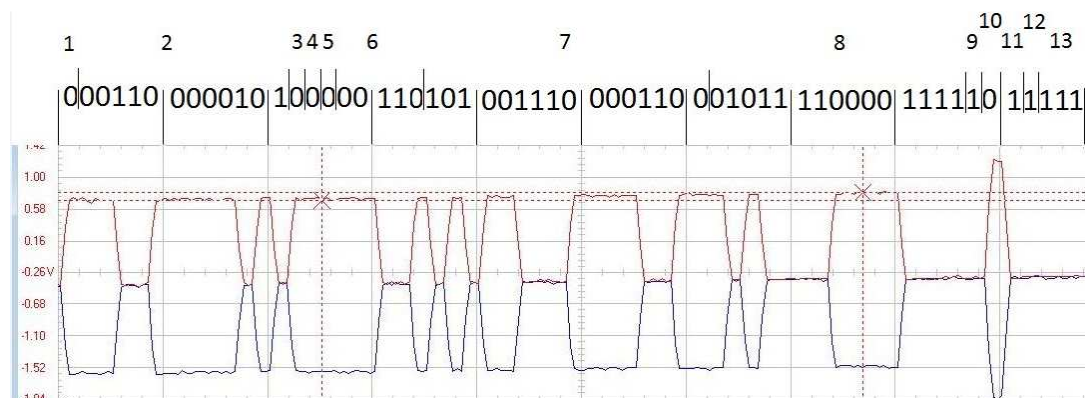
Communication object	COB-ID(s) hex	Slave nodes
NMT node control	000	Receive only
Sync	080	Receive only
Emergency	080 + NodeID	Transmit
TimeStamp	100	Receive only
PDO	180 + NodeID 200 + NodeID 280 + NodeID 300 + NodeID 380 + NodeID 400 + NodeID 480 + NodeID 500 + NodeID	1. Transmit PDO 1. Receive PDO 2. Transmit PDO 2. Receive PDO 3. Transmit PDO 3. Receive PDO 4. Transmit PDO 4. Receive PDO
SDO	580 + NodeID 600 + NodeID	Transmit Receive
NMT node monitoring (node guarding/heartbeat)	700 + NodeID	Transmit
LSS	7E4 7E5	Transmit Receive

Kuva 37. Sanomaprotokollat ja COB-ID:t (CANopen Solutions www-sivut b)

6.3 Viestikehys

Kehyksen koko CAN-väylässä voi vaihdella hyvinkin paljon riippuen lähetettävän datan määrästä (0–8 tavua) ja käytettävästä tunnisteesta (standardi 11-bittinen vai laajennettu 29-bittinen). Standardikehyksen koko on yleisesti 44bit + 8 * datatavut +

tahdistusbitit. Kuvassa 38 on oskilloskoopin kuva CAN-väylästä, jonne on MiniMon V3 -ohjelmalla lähetetty viesti, jonka COB-ID on 181_{hex} ($\text{PDO } 1 + \text{Node-ID} = 180_{\text{hex}} + 1_{\text{hex}}$) ja data $A70C_{\text{hex}}$. Siirtonopeudeksi on määritetty 125 kbit/s (n. $8 \mu\text{s/bit}$). Punainen viiva on CAN-H-johdon jännitteen kuvaaja ja sininen viiva CAN-L-johdon jännitteen kuvaaja.



Kuva 38. CAN-kehys oskilloskoopilla kuvattuna

1. SOF (Start-of-frame) on kehyksen aloitusbitti, jonka nousevalla reunalla tahdistetaan muut solmut.
2. Identifier on kehyksen 11-bittinen tunniste. Kuvassa 38 se sisältää 12-bittiä, sillä viiden saman bitin jälkeen on lähetetty yksi tahdistusbitti. Kun tahdistusbitti poistetaan jonosta, saadaan binääriluvuksi $001\ 1000\ 0001_{\text{bin}}$, joka vastaa lähetyksessä määriteltyä COB-ID:tä 181_{hex} .
3. RTR-bitti (Remote Transmission Request) määrittää, onko kyseessä varsinainen datakehys (arvo 0) vai kyselykehys (arvo 1). CAN-väylässä on mahdollista lähettää myös kyselykehyksiä, joilla saadaan jokin tietty solmu lähettämään datansa väylään.
4. IDE-bitti (Identifier Extension Bit) määrittää, onko tunniste 11-bittinen (arvo 0) vai 29-bittinen (arvo 1).
5. RES-bitti (Reserved) on aina dominantti.
6. DLC (Data Length Code) ilmaisee lähetettävän datan määrän tavuina (arvot välillä $0-8_{\text{dec}}$). Kuvan tapauksessa kenttä sisältää tahdistusbitin, mutta sen poistamalla saadaan binääriluku 0010_{bin} , joka vastaa lähetyksen kahta tavua.
7. DF (Data Field) sisältää lähetyksessä siirrettävän datan. Kuvan kehyksen binääriluku $1010\ 0111\ 0000\ 1100_{\text{bin}}$ vastaa lähetyksen dataa $A70C_{\text{hex}}$.

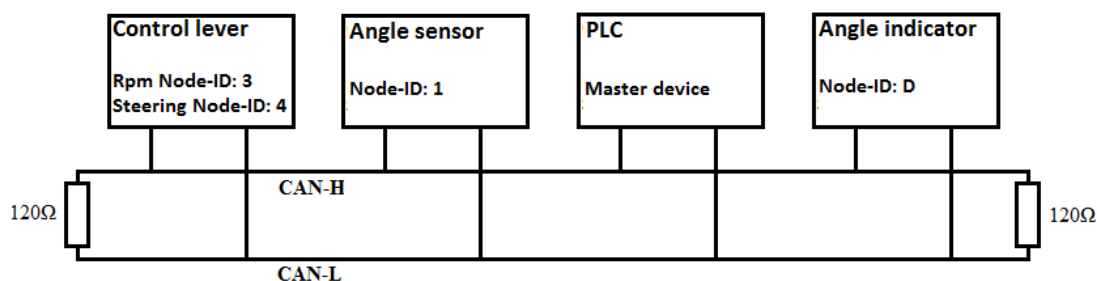
8. Sisältää CRC-tarkisteen kehykselle.
9. CRD-bitti (CRC Delimiter) on aina resessiivinen.
10. Tahdistusbitti.
11. ACK-bitti kertoo, onko jokin solmu vastaanottanut kehyksen. Lähettäjä lähettää tämän bitin resessiivisessä tilassa, ja viestin vastaanottava solmu kääntää sen dominantiksi. Bitti ei siis varmista, että jokainen kyseistä tietoa tarvitseva solmu olisi kehystä saanut.
12. ACD-bitti (ACK Delimiter) on aina resessiivinen.
13. EOF (End-of-frame) koostuu seitsemästä resessiivisestä bitistä ilman tahdistuksia. Näin muut solmut tietävät lähetyksen loppuneen. Väylä jää resessiiviseen tilaan, ellei jokin muu solmu aloita lähetystä heti edellisen kehyksen perään.
(Saha 2005.)

7 KULMANÄYTÖN CAN-OMINAISUUKSIEN TUTKIMINEN

Tutkimuksen tarkoituksena oli selvittää, kuinka XDi-kulmanäytön kaikki tarvitsema tieto viedään sille CAN-väylää pitkin. Tähän asti vain potkurilaitteen kulmatieto on tuotu suoraan CAN-väylää pitkin näyttöön, ja kaikkien muiden indikointien data on tuotu näyttöön analogisena. Analogiaviestit on kuitenkin ollut tarpeellista tuoda vain yhteen näyttöön, sillä näytön asetuksista saadaan näyttö muuntamaan saadut analogiasignaalit CAN-väylään muiden näyttöjen ymmärtämään muotoon. Väylään siirrettävät tiedot ovat potkurin kierrosnopeus, potkurin prosentuaalinen kierrosnopeus, pääkoneen teho, pääkoneen prosentuaalinen teho, kulman referenssipiste sekä näytön kirkkaudensäätö.

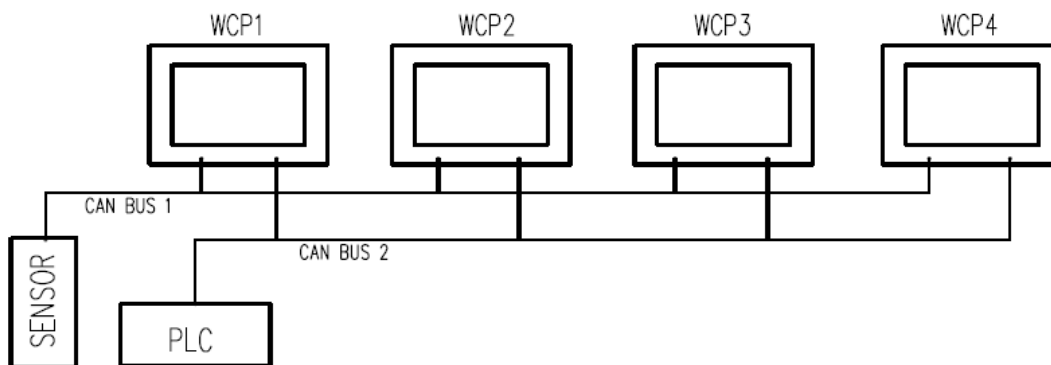
7.1 Väylärakenne

Pienoismallin tapauksessa samaan väylään on sijoitettu kulmanäytön ja kulma-anturin lisäksi myös ohjaukshahva ja logiikka (kuva 39).



Kuva 39. Pienoismallin CAN-väylän rakenne

Euroopan unionin MED-direktiivi (Marine Equipment Directive) määrää potkurin kulman indikoinnissa noudatettavaksi standardin ISO 20673 kohtaa 3.1 "*An indicator system consists of a transmitter and a receiver (indicator)*" (ISO 20673, 3.1). Kulma-anturin ja kulmanäytön väylään ei saa siis kytkeä mitään muuta laitetta. Tästä syystä toimitusprojekteissa kulmanäytölle on rakennettava kaksi erillistä väylää, joista toinen on vain kulmatiedolle ja toinen on muille indikoinneille (kuva 40). Kulmanäytön pohjassa on kaksi CAN-terminaalia, joten siihen voidaan liittää myös kaksi erillistä väylää.



Kuva 40. Toimitusprojektien väyläjako

7.2 Viestien tutkinta

Tutkiminen aloitettiin syöttämällä yhteen näyttöön analogiaviestit ja asettamalla kyseisen näytön ”XDi-net functions” -asetus päälle, jolloin se muuntaa saadut analogiasignaalit CAN-väylään. Tämän lisäksi asetettiin näytön ”Send XDi-net setup synch data on” -asetus päälle, jolloin käynnistyessään näyttö lähettää parametrit muille näytöille CAN-väylän kautta SDO-viestein. Näin saadaan muut väylän näytöt tietoisiksi, minkä näytön lähettämää dataa väylältä kuunnellaan (luo eräänlaisen master/slave-suhteen näyttöjen välille). CAN-väylä koostui tässä vaiheessa vain kulmanturista (Node-ID = 1_{hex}) ja näytöstä (Node-ID = B_{hex}). Näytöstä lähtevää CAN-väylää monitoroitiin MiniMon V3 -ohjelmalla (kuva 41).

Time / 10 mSec	Identifier	Format	Flags	Data
00:00:00.56		181 Std		E1 04
00:00:00.57		70B Std		05
00:00:00.58		181 Std		E1 04
00:00:00.60		181 Std		E1 04
00:00:00.60		20B Std		0B 01 30 07 76 F5 00 00
00:00:00.60		20B Std		0B 01 35 02 53 00 00 00
00:00:00.60		20B Std		0B 81 30 02 ED FB 00 00
00:00:00.60		20B Std		0B 91 30 02 DB FD 00 00
00:00:00.60		20B Std		0B 01 34 02 BC F7 00 00
00:00:00.61		20B Std		0B 11 34 02 D2 FE 00 00
00:00:00.62		181 Std		E1 04
00:00:00.64		181 Std		E1 04
00:00:00.66		181 Std		E1 04
00:00:00.68		181 Std		E2 04
00:00:00.70		181 Std		E1 04
00:00:00.72		181 Std		E1 04

Kuva 41. CAN-väylän monitorointi

Kuvasta 41 nähdään näytön (Identifier = 20B_{hex}) lähettävän väylään kuutta eri viestiä. Muut väylässä näkyvät viestit ovat kulma-anturin (Identifier = 181_{hex}) lähettämiä. 70B_{hex} on näytön lähettämä syke ("heartbeat"). Monitoroinnin jälkeen selvitettiin viestien sisällön tarkoitukset analysoimalla yhden viestin data. Analysointiin otettiin näytön lähettämän kolmannen viestin data 0B81 3002 EDFB 0000_{hex}. Ensimmäinen tärkeä huomio oli, että näytön käyttämä sanamuotoinen data kulkee väylällä vähemmän merkitsevä tavu ensin. Viestin dataosuuden todettiin rakentuvan seuraavasti:

- Ensimmäinen tavu 0B_{hex} määräytyy viestiä lähettävän näytön mukaan.
- Seuraavat kaksi tavua määräävät objektikirjaston osoitteen. Objektikirjaston osoitteen pituus on yksi sana, joten näytön lähettämät tavut on käännettävä. Tavujen käännön jälkeen saadaan osoitteeksi 3081_{hex}. Objektikirjaston osoite koostuu indeksistä ja instanssista (index + instance number). Instanssi on tarkoitettu esimerkiksi tilanteeseen, jossa samassa väylässä olevista näytöistä viisi indikoi moottorin 1 kierrosnopeutta ja toiset viisi moottorin 2 kierrosnopeutta. Usean näytön vuoksi on väylän kuormituksen vähentämiseksi hyvä käyttää näyttöjen broadcast-ID:tä, jolloin moottorin 1 ja moottorin 2 kierros-
lukutiedot on yksilöitävä jotenkin, jotta näytöt osaavat esittää oikean moottorin tiedot. Moottorien data yksilöidään tässä tapauksessa instanssinumerolla. Näytön parametreista saadaan määrättyä, mitä instanssia näyttö kyseisestä tiedosta kuuntelee. Instanssien maksimimäärä on 16 (0 – F_{hex}). Perusasetuksilla näyttö kuuntelee instanssia numero 1_{hex}. Tutkittavan datan indeksi on siis 3080_{hex} ja instanssi on 1_{hex}.
- Seuraava tavu on ali-indeksi, joka kertoo viestin datan tyyppin. Ali-indeksi 02_{hex} tarkoittaa etumerkillistä 16-bittistä integer-lukua. Ali-indeksin 07_{hex} datatyyppi on myös etumerkillinen 16-bittinen integer-luku, mutta kyseisen ali-indeksin käyttö on tarkoitettu vain kulman referenssipisteen arvolle.
- Kaksi seuraavaa tavua sisältävät varsinaisen datan. Datan pituus on siis yksi sana, joten tavut on taas käännettävä. Tavujen käännön jälkeen arvoksi saadaan FBED_{hex}.
- Viimeiset kaksi tavua ovat aina 0000_{hex}.

Analysoinnin jälkeen verrattiin monitoroituja viestejä kulmanäytön prosessidatan objektikirjastoa kuvaavaan taulukkoon (kuva 42), josta saatiin selitykset näytön lähettämälle kuudelle erilaiselle viestille (kuva 43).

XDi Index 0x2000 – 0x5FFF	Max # objects Instance	Object types	Note
0x2000 – 0x2FFF	4096	System and main control parameters	
0x3000 – 0x3FFF	4096	Variable XDi input / output data	
0x3000-0x31FF	512	Propulsion	
0x3000-0x300F	16	Azimuth & Rudder	"-" = bow turns to port
0x3010-0x301F	16	Rudder%	"-" = bow turns to port
0x3020-0x302F	16	Pitch angle	"-" = astern
0x3030-0x303F	16	Pitch %	"-" = astern
0x3040-0x307F	64	reserved	
0x3080-0x308F	16	Propeller RPM	"-" = astern
0x3090-0x309F	16	Propeller RPM%	"-" = astern
...			
...			
0x3400 - 340F	16	Electrical Power AC1	
0x3410 - 341F	16	Electrical Power AC1 %	

Kuva 42. XDi-näytön prosessidatan osoitteet objektikirjastossa (DEIF A/S 2014, 29)

Angle reference -->	0B 01 30 07 76 F5 00 00
Dimmer value -->	0B 01 35 02 53 00 00 00
Rpm value -->	0B 81 30 02 ED FB 00 00
Rpm% value -->	0B 91 30 02 DB FD 00 00
Power value -->	0B 01 34 02 BC F7 00 00
Power% value -->	0B 11 34 02 D2 FE 00 00

Kuva 43. Väylän viestien merkitykset

7.3 Lähetysten rakennus logiikkaohjelmaan

Seuraavaa testausta varten näytöstä irrotettiin analogiamoduulit, sillä nyt tarkoituksena oli lähettää kaikki näytön tarvitsema data CAN-väylää pitkin. Testauksessa havainnollistettiin toimitusprojektien mahdollista tilannetta, jossa jokaisen näytön toiseen CAN-terminaaliin on kytketty logiikasta tuleva CAN-väylä ja toiseen terminaaliin kulma-anturin väylä (kuva 40). Käytettäessä näytön molempia CAN-terminaaleja ja logiikan hoitaessa datan lähetyksen väylään tuli näytön asetukset määrittää uudelleen. Näytön molemmat terminaaliin tuli määrittää käyttöön sekä määrätä, mitä viestejä niistä kuunnellaan. Tämän lisäksi tuli määrittää terminaaleihin saapuvan tiedon

sanomaprotokollat, jotta näyttö osaa valita saapuvista kehyksistä oikeat. Asetuksien määrittämisestä tehtiin erilliset ohjeet (liite 5).

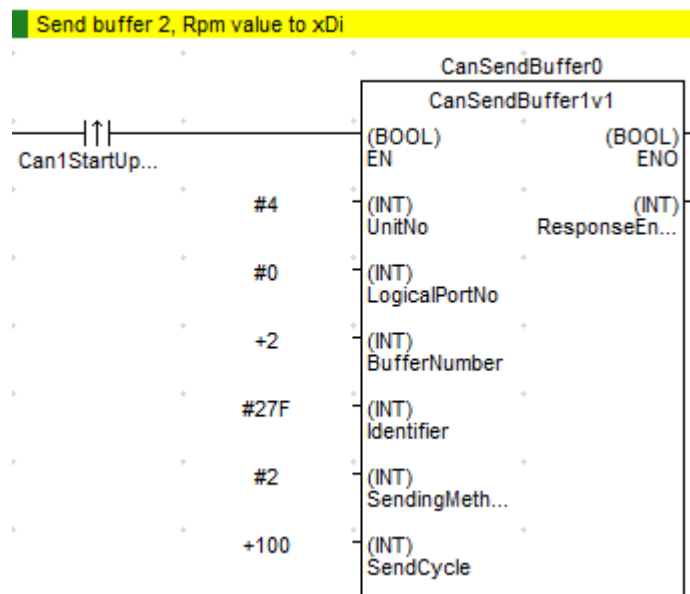
7.3.1 Lähetysoiskurien määrittäminen

Lähetysoiskurin määrittämisessä käytetään valmista toimilohkoa (kuva 44), jolle määritetään kommunikointiyksikön tarvitsemat tiedot. Lähetysoiskuria tehtäessä tulee toimilohkolle määrittää moduulinumero, portti, oiskurinumero, lähetysoista, lähetysoykli sekä tunniste. Tässä työssä CORT21-moduulinumero on 4_{hex} ja sisäisenä väylänä on käytetty porttia 0_{hex} . Oiskurinumeroiksi määritetään seuraava vapaana oleva oiskurinumero. Ohjaukskahvalle määritettyjen lähetysoiskureiden jälkeen seuraava vapaana oleva oiskurinumero on 2. Lähetysostavaksi valittiin syklinen lähetyso 100 ms:n välein.

Edellä mainittujen tietojen lisäksi tulee toimilohkolle määrittää tunniste eli mitä sanomanumeroa vastaanottava solmu kuuntelee. Näytön asetuksista määritettiin näytön ensimmäinen CAN-terminaali kuuntelemaan sanomaprotokollan TPDO1 (Transmit PDO 1) viestejä, sillä kulma-anturi lähettää viestinsä kyseisellä sanomaprotokollalla. Näytön toinen CAN-terminaali määritettiin kuuntelemaan sanomaprotokollan RPDO1 (Receive PDO 1) viestejä. Kuvasta 37 nähdään orjalaitteen sanoman vastaanotto-protokollan PDO 1 tunnisteiden olevan 200_{hex} .

Sanomanumeron (COB-ID) määrittämiseseen tarvitaan edellä mainitun sanomaprotokollan tunnisteiden lisäksi vastaanottavan solmun noodinumero. Toimitusprojekteissa on aina useita kulmanäyttöjä (esim. 4 kpl) samassa väylässä ja jokainen väylän näyttö tarvitsee kuvassa 43 esitetyt viestit. CAN-protokollan mukaan samassa väylässä olevilla solmuilla tulee jokaisella olla itsensä yksilöivä noodinumero, joten väylän näytöille ei voi määrittää samaa noodinumeroa. Jokainen väylän näyttö kuuntelee siis eri sanomanumeron viestiä. Tästä syystä tarvittavien lähetysoiskureiden määrä on 24 (6 viestiä $\cdot 4$ näyttöä). CORT21-kommunikointiyksikkö lisää logiikan sykliäikää $0,3$ ms + $0,0064$ ms \cdot oiskureiden määrä. Oiskurit lisäävät siis väylän kuormituksen lisäksi myös logiikan kiertoaikaa, joten oiskureiden määrä kannattaa pitää mahdollisimman vähäisenä. Näyttöjen osalta oiskureiden vähentäminen onnistuu käyttämällä

näyttöjen broadcast-ID:tä (= noodinumero, jota jokainen näyttö kuuntelee). Broadcast-ID näyttöille on $7F_{\text{hex}}$, joten COB-ID:ksi muodostuu $27F_{\text{hex}}$. Näin tehtäessä näyttöjen lukumäärä ei vaikuta enää puskureiden lukumäärään. Jokainen viesti tarvitsee silti oman puskurinsa, joten tarvittavien puskureiden lopullinen lukumäärä on 6.



Kuva 44. Esimerkki puskurin määrittämisestä

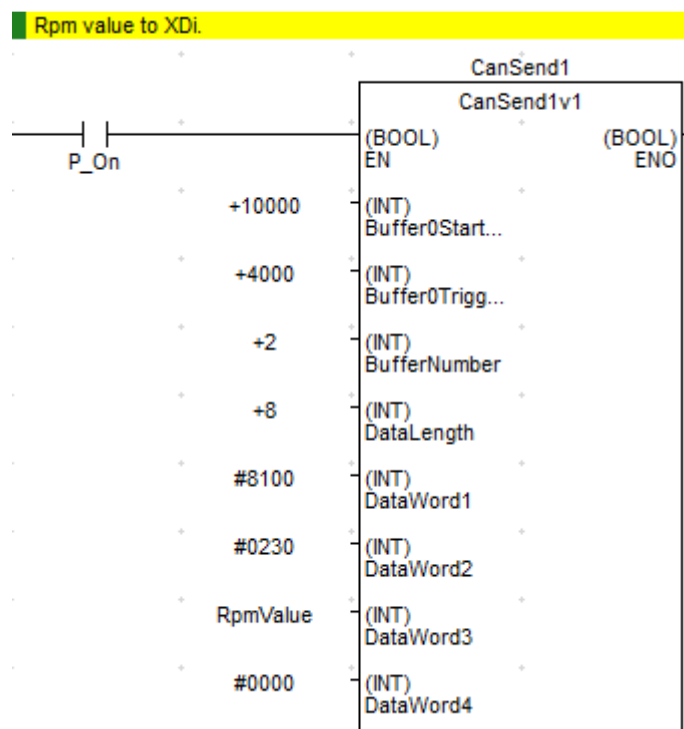
7.3.2 Lähetettävän viestin dataosuus

Puskureiden määrittämisen jälkeen testattiin, kuinka ja missä järjestyksessä data tulee logiikasta lähetettäväksi, jotta se on väylällä näyttöjen ymmärtämässä muodossa. Lähetystyökaluun käytetään valmiista toimilohkosta CanSend1v1 (kuva 45), joka siirtää sille määritetyt tulokset oikeaan lähetyspuskurin muistialueelle. Siirtoa varten toimilohkolle tulee määrittää lähetyspuskureiden datamuistialueen ensimmäinen osoite (D10000), lähetyspuskureiden trigger-bittien muistialueen ensimmäinen osoite (CIO 4000), puskurin numero (tällä toimilohko laskee, mihin muistialueiden sanoihin data tulee tallettaa), lähetettävän datan pituus tavuissa sekä itse data.

Väylältä monitoroitujen, näytön lähettämien viestien ensimmäinen tavu tarkoitti lähetettävän näytön noodinumeroa. Nyt logiikan lähettäessä viestit väylään ja käytettäessä broadcast-ID:tä voidaan kyseinen tavu määrittää arvoon 00_{hex} . Propellien pyörintänopeuden data tulee olla väylällä muodossa $0081\ 3002\ \text{xxxx}\ 0000_{\text{hex}}$, jossa xxxx

on propellien pyörintänopeus (tavut väärinpäin). Tässä muodossa näyttö osaa vastaanottaessaan sijoittaa viestin datan oikeaan paikkaan objektkirjastossaan. Oikealla paikalla tarkoitetaan tässä tapauksessa objektkirjaston paikkaa, josta näytön pyörintänopeuden indikaattori hakee tietonsa.

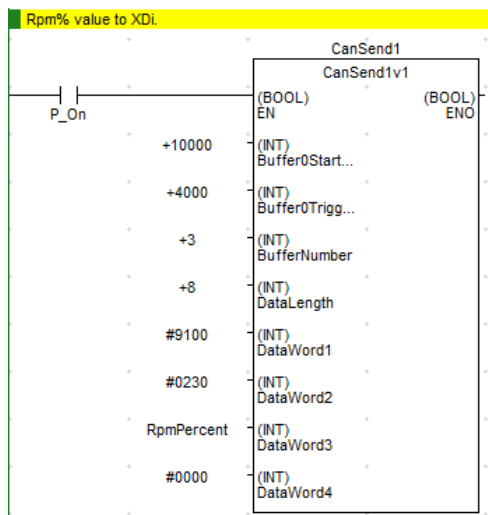
Lähetyspuskuri lähettää viestikehyksen dataosuuden väylään sana kerrallaan vähemmän merkitsevä tavu ensin. Tästä syystä väylälle halutun viestin sanat tulee lojiikkaohjelmaan määrittää tavut väärinpäin. Näin toimilohkolla (kuva 45) määritettävän viestin muoto on $8100\ 0230\ xxxx\ 0000_{hex}$. Propellien pyörintänopeustiedon $xxxx$ tulee kulkea väylällä väärinpäin, joten lähetykseen sen tavut on määritettävä oikeinpäin. Lähetystä varten tulee myös tietää, missä muodossa indikaattori käyttää tietoa. Propellien rpm-arvo lähetetään näytölle todellisena arvona, jonka resoluutio on 0,1 rpm. Datatyyppi on integer-luku, joten lähetettävät arvot voivat olla välillä $-32768_{dec} - 32767_{dec}$. Täten näytöllä kykenee esittämään propellien kierrosluvut välillä $-3276-3276$ rpm.



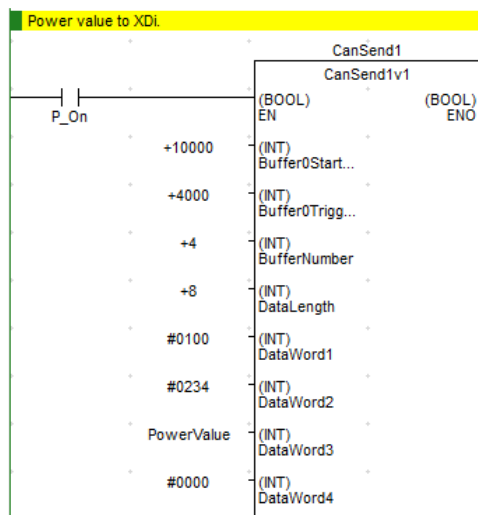
Kuva 45. Propellien pyörintänopeustiedon siirto puskurin muistialueelle

Vastaavasti selvitettiin muiden lähetettävien viestien muoto:

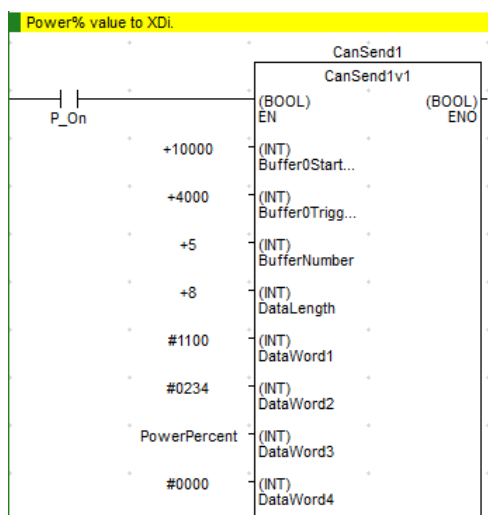
- Prosentuaalisen kierrosluvun indikaattorin (pylväs-indikaattori) käyttämän datan resoluutio on 0,1 prosenttia, joten lähetettävät arvot $-200-1200_{\text{dec}}$ vastaavat rpm-pylvään arvoja $-20-120$ %. Lähetyspuskurin muistialueelle siirrettävä data tavujen käynnön jälkeen on $9100\ 0230\ \text{xxxx}\ 0000_{\text{hex}}$ (kuva 46).
- Tehon numeraalisen arvon esitys näytössä käyttää resoluutiota 1 kW, joten arvot $-32768_{\text{dec}} - 32767_{\text{dec}}$ vastaavat näytöllä tehon arvoja $-32768-32767$ kW. Lähetyspuskurin muistialueelle siirrettävä data tavujen käynnön jälkeen on $0100\ 0234\ \text{xxxx}\ 0000_{\text{hex}}$ (kuva 47).
- Prosentuaalisen tehon indikaattorin (pylväs-indikaattori) käyttämän datan resoluutio on 0,1 prosenttia, joten lähetettävät arvot $0-1200_{\text{dec}}$ vastaavat teho-pylvään arvoja $0-120$ %. Lähetyspuskurin muistialueelle siirrettävä data tavujen käynnön jälkeen on $1100\ 0234\ \text{xxxx}\ 0000_{\text{hex}}$ (kuva 48).
- Käynnön referenssipisteen ilmaisevan nuolen käyttämä data on resoluutioltaan 0,1 astetta, joten sille lähetettävät arvot $0-1800_{\text{dec}}$ vastaavat kulmaa $0-180$ astetta. Lähetettävän datan etumerkillä ilmaistaan kulman suunta nollassa kulmaan nähden. Ympyrän vasen puoli ilmaistaan positiivisena ja oikea puoli negatiivisena. Näin ollen lähetettävän datan arvot ovat välillä $-1800-1800$ ($-180^\circ - 180^\circ$). Lähetyspuskurin muistialueelle siirrettävä data tavujen käynnön jälkeen on $0100\ 0730\ \text{xxxx}\ 0000_{\text{hex}}$ (kuva 49).
- Näytön kirkkaudensäätöön käytettävän viestin arvot $0-100$ vastaavat näytön kirkkauden asetusta $0-100$ %. Lähetyspuskurin muistialueelle siirrettävä data tavujen käynnön jälkeen on $0100\ 0235\ \text{xxxx}\ 0000_{\text{hex}}$.



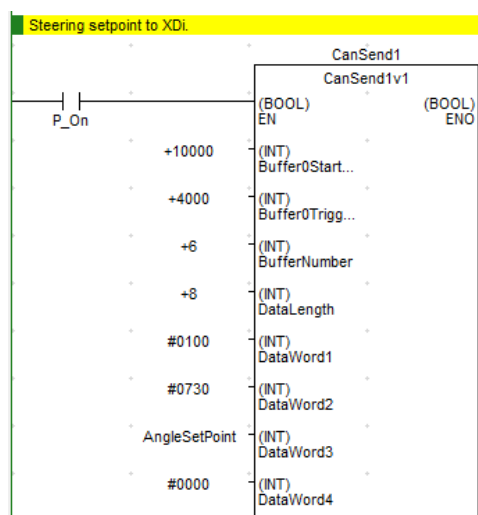
Kuva 46. Prosentuaalinen rpm



Kuva 47. Numeraalinen teho



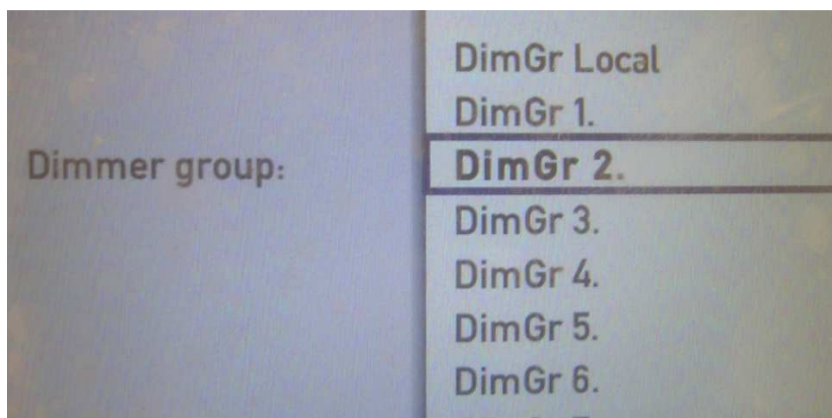
Kuva 48. Prosentuaalinen teho



Kuva 49. Kulman referenssi

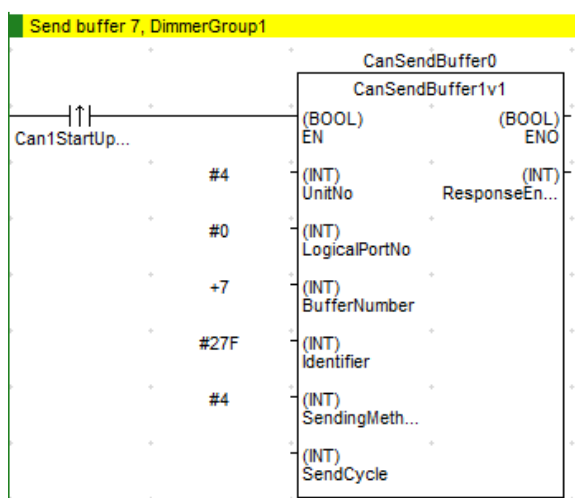
7.3.3 Kirkkaudensäätö

Väylässä olevien kulmanäyttöjen taustavalon kirkkautta voi joskus olla tarpeen muuttaa näyttökohtaisesti tai joukkokohtaisesti. Tästä syystä näytöille lähetettävät kirkkaudensäätöviestit tulee pystyä yksilöimään. Tämä onnistuu käyttämällä näyttöjen omia noodinumeroita tai näyttöjen objektikirjaston instansseja. Noodinumeroa käytettäessä tehdään jokaiselle näytölle oma kirkkaudensäädön lähetyspuskuri, sillä jokaisen näytön COB-ID on eri. Käytettäessä instanssia kirkkaudensäätöön tulee näytön kuuntelema instanssinumero määrätä näytön asetuksissa Dimmer Group -asetuksella (kuva 50). DimGr 1 -asetuksella näyttö kuuntelee instanssia 1, DimGr 2 -asetuksella instanssia 2 jne.



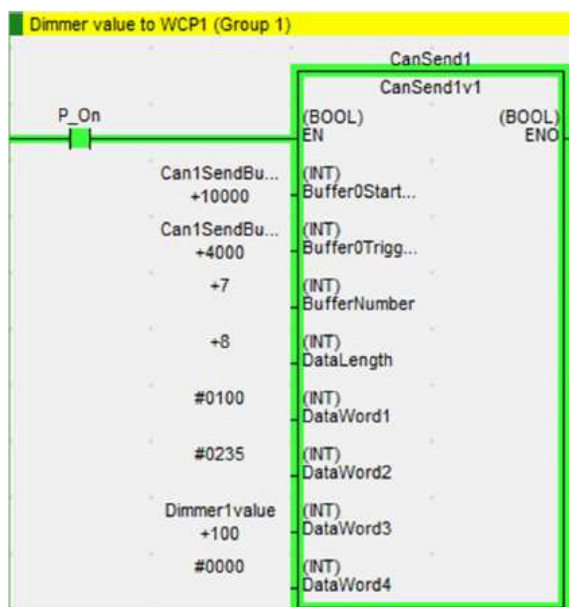
Kuva 50. Näytön kirkkaudensäädön instanssinumeron määrittäminen

Instanssinumeroa on hyvä käyttää, jos väylällä on näyttöjä, jotka tarvitsevat ryhmitäin eri kirkkausasetukset. Tällöin on suotavaa käyttää näyttöjen broadcast-ID:tä, jolloin ryhmien yksilöiminen tapahtuu instanssinumerolla. Näyttöjen instanssinumeroiden erotessa toisistaan tulee jokaiselle instanssiryhmälle määrittää kirkkaudensäätöön oma lähetyspuskuri. Kulmanäytön indikaattorien data-arvojen tulee päivittyä objekti kirjastossa tietyin väliajoin, joten indikaattorien datan lähetykseen käytettävien puskureiden lähetystapa on määritetty sykliseksi. Näyttö ilmoittaa virheestä, jos jonkin indikaattorin tarvitsemaa viestikehystä ei saavu tarpeeksi usein. Näytön kirkkaudensäädön prioriteetti on paljon pienempi, joten näyttö ei tarvitse syklisesti tietyin väliajoin kirkkaudensäädön viestikehystä, vaan näyttö käyttää edellisen kehysten arvoa, kunnes uusi kehys saapuu. Tästä syystä näyttöjen kirkkaudensäädön puskurien lähetystapa voidaan määrittää arvoon 4_{hex} (kuva 51), jolloin puskuri lähettää viestikehysten vain silloin, kun kirkkaudensäädön data-arvo muuttuu.

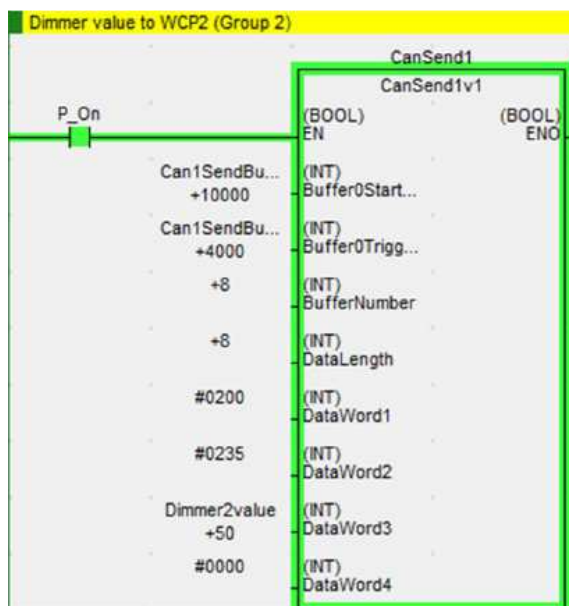


Kuva 51. Kirkkaudensäädön lähetyspuskurin luonti

Lähetämällä kirkkaudensäädön viestikehys väylään vain datan muuttuessa vähennetään CAN-väylän turhaa kuormitusta huomattavasti. Näyttöjen instanssien erotessa toisistaan puskurien muistialueille siirrettävistä sanoista ensimmäinen (DataWord1) määrittää viestin instanssin (kuvat 52 ja 53).



Kuva 52. Instanssi-ryhmä 1

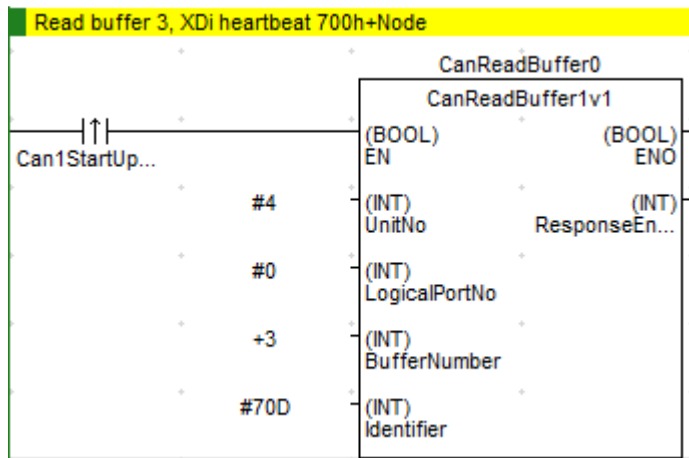


Kuva 53. Instanssi-ryhmä 2

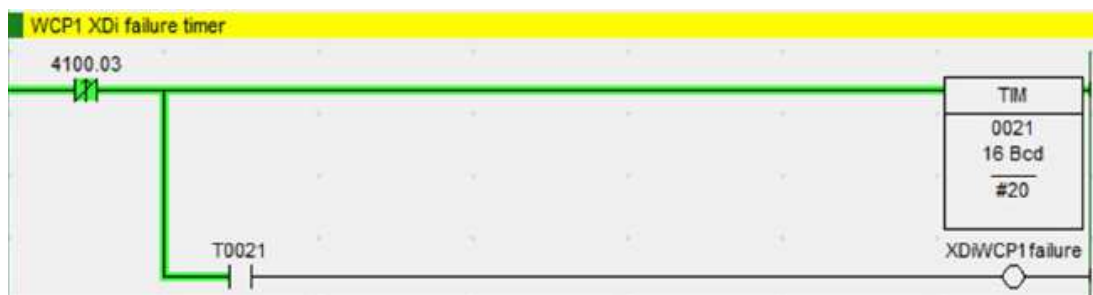
7.3.4 Heartbeat

Kaikkien väylän näyttöjen olemassaolon varmistamiseksi käytetään näyttöjen sykettä eli NMT Node Monitoring -protokollaa (kuva 37). Kyseisen protokollan tunniste on 700_{hex} , joten sanomanumerot ovat muotoa $700_{\text{hex}} + \text{Node-ID}$. Kulmanäyttöjen asetuksista asetetaan näytöt lähettämään sykettä CAN-väylään 2, jolloin logiikka voidaan määrittää kuuntelemaan näyttöjä. Näyttöjä ei tarvitse määrittää lähettämään sykettä väylään 1, sillä väylältä löytyy vain kulma-anturi, joka ei näytön lähettämää sykettä käsittele.

Jokaisen näytön lähettämän sykkeen tunnistekenttä sisältää näytön noodinumeron, joten sykkeiden yksilöiminen on helppoa. Sykkeen kuuntelua varten tulee jokaiselle väylän näytölle määrittää oma vastaanottopuskuri logiikkaan (kuva 54). Puskurien määritysten jälkeen tarvitsee vain seurata jokaisen määritetyn vastaanottopuskurin trigger-bittiä (kuva 55). Jos jokin trigger-bitti on "0-tilassa" liian pitkään, tiedetään kyseisen vastaanottopuskurin kulmanäytön "kadonneen" väylältä.



Kuva 54. Näytön (noodinnumero = D) sykettä varten määritettävä vastaanottopuskuri



Kuva 55. Sykkeen vastaanottopuskurin trigger-bitin seuraaminen

8 KÄYTTÖLIITTYMÄ

Käyttöliittymän pohjaksi otettiin yrityksen erään toimitusprojektin käyttöliittymän ohjelmapohja. Tarvittavien muutoksien tekeminen käyttöliittymäohjelmaan edellytti tarkempaa tutustumista sekä CX-Designer-ohjelmaan että valittuun ohjelmapohjaan. Lopullisen käyttöliittymän oli tarkoitus kuvata ja esitellä yrityksen tuottaman ohjausjärjestelmän käyttöliittymää mahdollisimman hyvin. Tästä syystä, toisin kuin logiikkaohjelmassa, suurin osa pienoismallin toimintaan vaikuttamattomista sivuista jätettiin käyttöliittymään.

Valmiin ohjelmapohjan sisältämien sivujen osalta muutokset tuli tehdä vain käyttöliittymän sivujen objekteille linkitettyihin osoitteisiin. Osoitteiden muokkauksen lisäksi käyttöliittymään piti rakentaa uusia pienoismallille tarpeellisia sivuja, kuten ajotavan hallintasivu ja komponenttien (ohjauskahva, takaisinkytkentäanturi ja kulmanäyttö) tilatietosivut. Automaattiajotilaa varten käyttöliittymään rakennettiin myös 7-sivuinen Steerprop-esitys, jonka saa pyörimään näytölle. Pienoismallin järjestelmän näytteille asettamisesta messutapahtumaan tehtiin erilliset ohjeet (liite 6). Asennusohjeiden lisäksi pienoismallin järjestelmän ja käyttöliittymän käytöstä tehtiin ohjeet (liite 7) messuvieraita varten.

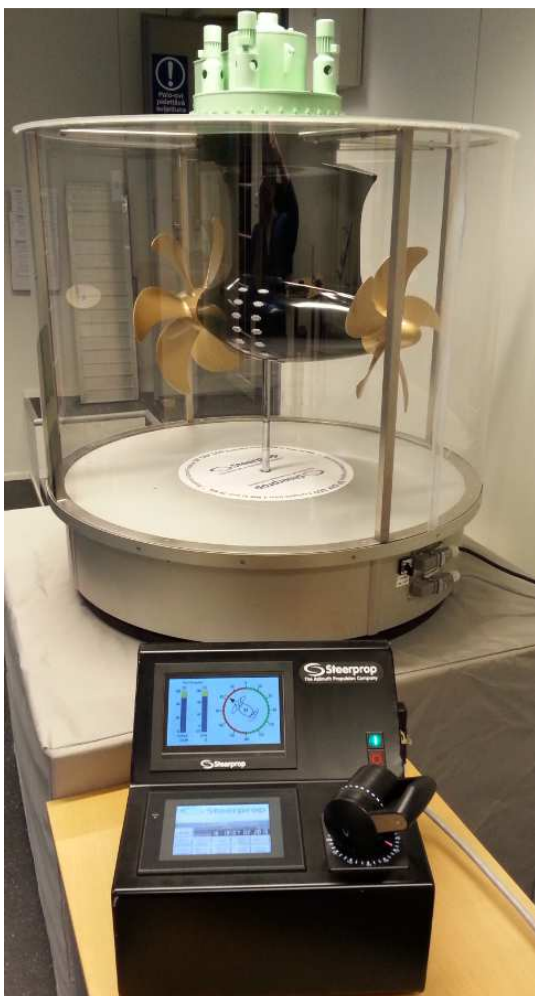
9 YHTEENVETO

Tämän opinnäytetyön tarkoituksena oli suunnitella ja toteuttaa CRP-potkurilaitteen pienoismallille uusi ohjausjärjestelmä, joka kuvaisi mahdollisimman hyvin Steerprop Oy:n tuottamaa ohjausjärjestelmää. Työn vaatimuksena oli sisällyttää pienoismallin järjestelmään Steerpropin käyttämät uudet ohjauskomponentit sekä käyttöliittymä-paneeli ja logiikka. Logiikan lisäyksen tarkoitus oli mahdollistaa pienoismallille oikean propulsiolaitteen toiminnallisuudet, kuten follow-up-ohjaus sekä kiihdytys- ja jarrutusrampit. Järjestelmän parametrien, kuten kääntönopeuden ja ramppien pituuksien, tuli vastata yrityksen tuottaman CRP 120 ECO -laitteen parametreja. Pienoismallia käytetään messutapahtumissa, joten ohjelmistoon vaadittiin edellä mainittujen toimintojen lisäksi automaattiajotila. Pienoismallin järjestelmän suunnitteluun sisällytettiin myös uuden kulmanäytön CAN-rajapinnan tutkiminen.

Työ oli mielenkiintoinen, sillä se sisälsi paljon erilaisia tehtäviä niin suunnittelun kuin käytännön työnkin osalta. Työn suunnitteluosuus koostui ohjauslaatikon mekaniikkasuunnittelusta, logiikan- ja käyttöliittymän ohjelmistosuunnittelusta sekä järjestelmän sähkösuunnittelusta. Näiden lisäksi tuli tilata sähkökomponentit, joita järjestelmä tarvitsee toiminnallisuuksiinsa. Suunnittelun jälkeen työ sisälsi järjestelmän kokoamisen ja johdotuksen sekä lopullisen järjestelmän täydellisen testaamisen.

Pienoismallin lopullinen järjestelmä (kuva 56) oli toimiva kokonaisuus, joka sisälsi vaaditut komponentit, parametrit sekä toiminnallisuudet. Vaadittujen ominaisuuksien lisäksi mallin ohjausjärjestelmä sisälsi paljon indikoiteja ja diagnostiikkaa. Messutilanteita varten käyttöliittymään rakennettiin toimitusprojektien käyttöliittymäsivujen lisäksi kattavat diagnostiikkasivut komponenttien toiminnasta, jotka hälytyksen sattuessa kertovat, mitä toiminnallisuuksia laite on menettänyt. Näin jonkin komponentin vikaantuessa messuvastaava tietää, voidaanko laitteen käyttöä jatkaa edes osalla toiminnoista. Pienoismallin järjestelmän lisäksi koottiin sen mukana kulkeva ohjausjärjestelmäkansio, joka sisältää asennus- ja käyttöohjeiden lisäksi paljon tietoa yrityksen ohjausjärjestelmän komponenteista sekä käyttöliittymästä. Opinnäytetyöhön sisällytetyn kulmanäytön tutkimisen osalta työ onnistui hyvin, sillä pienoismallin lopullisessa järjestelmässä kaikki kulmanäytön tarvitsemat tiedot viedään näytölle

CAN-väylää pitkin. Luokituslaitosten säännöstöjen vuoksi toimitusprojekteille oli suunniteltava erilainen väylärakenne kuin pienoismallille. Tästä syystä opinnäytetyössä tutkittiin ja toteutettiin tiedon vienti kulmanäytöille myös toimitusprojektien väylärakenteella. Toimitusprojektien näyttöjen asetuksista laadittiin erilliset ohjeet.



Kuva 56. Uusi järjestelmä toiminnassa

Tulevaisuudessa pienoismallin järjestelmää on tarkoitus parantaa automaattisella nopeussäädöllä. Potkurilaitteen käännohjauksen maksimijännitteellä 6,5 V saavutetaan kääntönopeus 1,5 rpm vasta, kun kääntömoottori on lämmennyt. Tästä syystä järjestelmän käynnistyshetkellä maksimikääntönopeus eroaa tavoitteellisesta. Tarkoitus on lisätä logiikkaohjelmaan ajastettu taski, joka laskee tietyin väliajoin kulmanuritetiedon muutoksesta potkurilaitteen kääntönopeuden (ohjauksen ollessa 100 %). Tuloksen mukaan taskin tulee muuttaa PWM-funktiolle menevän ohjausarvon skaalauksen maksimiarvoa, jolloin saadaan järjestelmään itseään muuttava nopeussäätö.

Työ kasvatti tekijänsä ammatillista osaamista paljon. Järjestelmää suunniteltaessa ja rakennettaessa tutustuttiin syvällisesti Omronin CX-Programmerin ja CX-Designerin ohjelmistoihin ja ominaisuuksiin. Myös CADS Planner -ohjelmiston käyttöä tuli harjoiteltua järjestelmän sähkösuunnittelun yhteydessä. Pienoismallin komponenttien tiedonvaihto perustui CAN-protokollaan, joten CAN-väylän väylärakenne ja tiedonvaihtoprotokollat sekä erilaiset CAN-pohjaiset komponentit tulivat tutuiksi. Opin näytetyötä tehdessä tuli opittua myös, kuinka Omronin logiikka saadaan liitettyä CAN-väylän master-laitteeksi siihen liitettävän CAN-väylämoduulin avulla ja kuinka kyseinen väylämoduuli konfiguroidaan. Tilaajayrityksen toimitusprojektien logiikka- ja käyttöliittymäohjelmaan tutustumisen ansiosta niiden lukeminen ja muokkaaminen tulevaisuudessa on helpompaa.

LÄHTEET

CANopen Solutions www-sivut a. 2015. Viitattu 26.02.2015.
http://canopen-solutions.com/canopen_fundamentals_en.html

CANopen Solutions www-sivut b. 2015. Viitattu 26.02.2015
http://www.canopensolutions.com/english/about_canopen/predefined.shtml

DEIF A/S. 2014. XDi-net/CANopen reference manual. Viitattu 28.02.2015. Steerprop Oy:n sisäinen verkkodokumentti.

DEIF:n www-sivut a. 2015. XDi data sheet 4921250067 UK. Viitattu 03.02.2015.
http://109.238.53.160//Publication/DownloadDocument.ashx?id=R1d47e12c7-bb75-44a2-a87e-d3a350f08865&FileName=_XDi_data_sheet_4921250067_UK

Electromenin www-sivut. 2015. EM-101 DC MOOTTORIOHJAIN 24V 3A 4-QUADRANT. Viitattu 10.02.2015.
http://www.electromen.com/files/5113/7811/4748/FI_em-101.pdf

FSG:n www-sivut. 2015. Measurement and Sensor Systems. Viitattu 03.02.2015.
http://www.fernsteuergeraete.de/images/pdfs-english/New_072013/FSG-AngularPositionTransducers-eng_072013.pdf

ISO 20673. Ships and marine technology. Electric rudder angle indicators. (ISO 20673:2007). 2007. International Organization for Standardization. Viitattu 11.03.2015. <http://www.iso.org/>

IXXAT:n www-sivut. 2015. miniMon Manual Vers. 1.3. Viitattu 03.02.2015.
<http://www.ixxat.com/download/4.02.0250.20013%20minimon%20v1.3.pdf>

Kwant Controls B.V. 2013. CANopen Interface Stepper Motor Control. Viitattu 03.02.2015. Steerprop Oy:n sisäinen verkkodokumentti.

Omron. 2009. User Defined CAN unit. Viitattu 03.03.2015.
<http://www.omronkft.hu/nostree/pdfs/plc/networks/w14e-en-01+cs1w-cort21+opermanual.pdf>

Omronin www-sivut a. 2015. CJ1M-CPU2[] Data Sheet. Viitattu 03.02.2015.
http://www.ia.omron.com/data_pdf/cat/cj1m-cpu2_ds_e_4_1_csm2075.pdf?id=1569

Omronin www-sivut b. 2015. Viitattu 03.02.2015.
http://industrial.omron.fi/fi/products/catalogue/automation_systems/hmi/scalable_hmi/ns5/default.html

Omronin www-sivut c. 2015. Viitattu 03.02.2015.
http://industrial.omron.fi/fi/products/catalogue/automation_systems/software/programming/cx-one/cx-programmer.html

Omronin www-sivut d. 2015. Viitattu 03.02.2015.
http://industrial.omron.fi/fi/products/catalogue/automation_systems/software/programming/cx-one/cx-designer.html

Omronin www-sivut e. 2015. Instructions reference manual. Viitattu 11.02.2015.
<http://downloads.omron.fi/IAB/Products/Automation%20Systems/PLCs/Modular%20PLC%20Series/CJ2M/W474/W474-E1-09%20INSTR%20REF%20MANUAL.pdf>

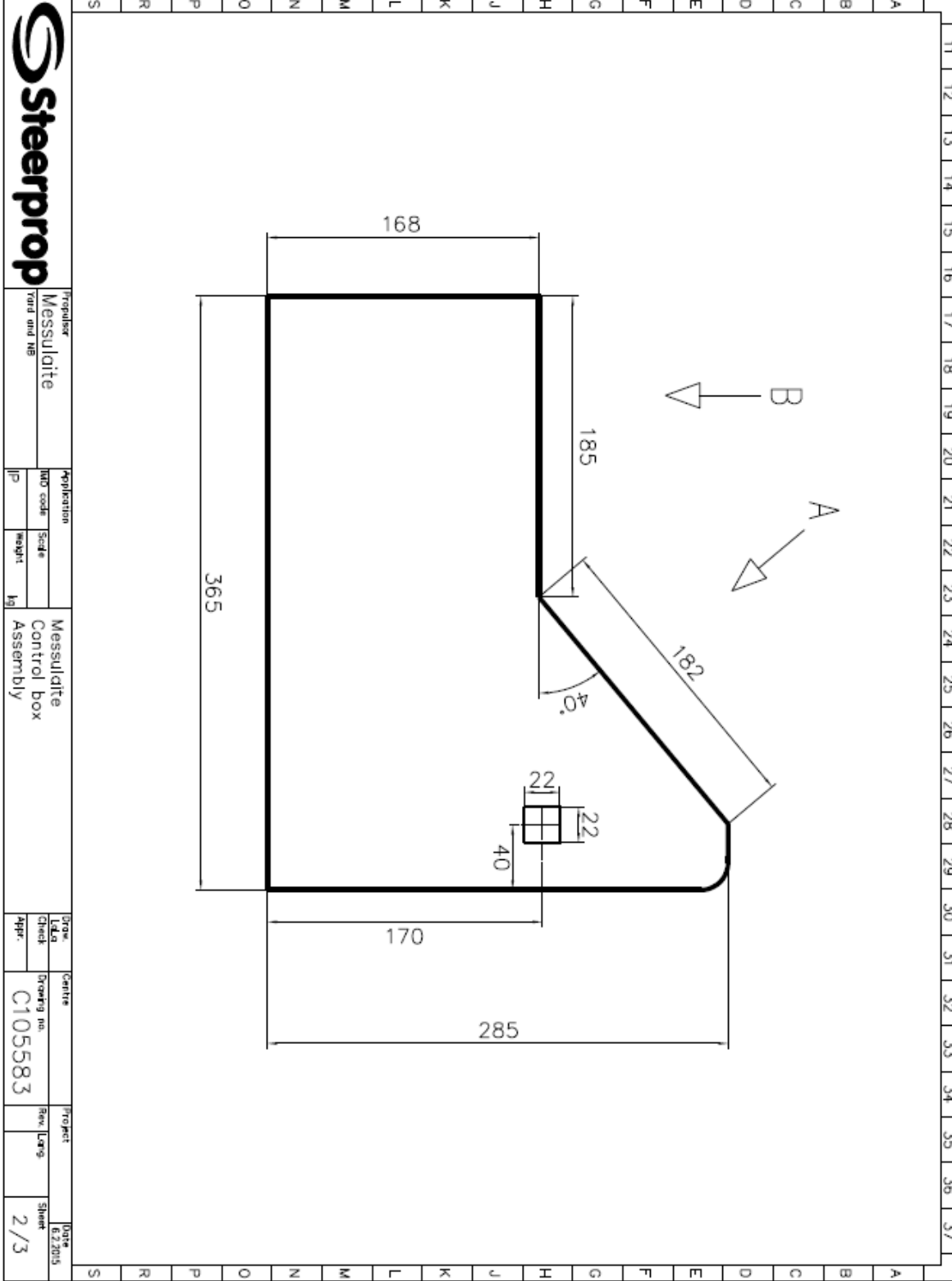
Saha, H. 2005. CAN-väylä. Viitattu 22.02.2015.
<http://www.canopen.fi/artikkelit/CAN.pdf>

Saha, H. 2006. CANopen perusteet. Viitattu 28.02.2015.
<http://www.canopen.fi/artikkelit/CANopen.pdf>

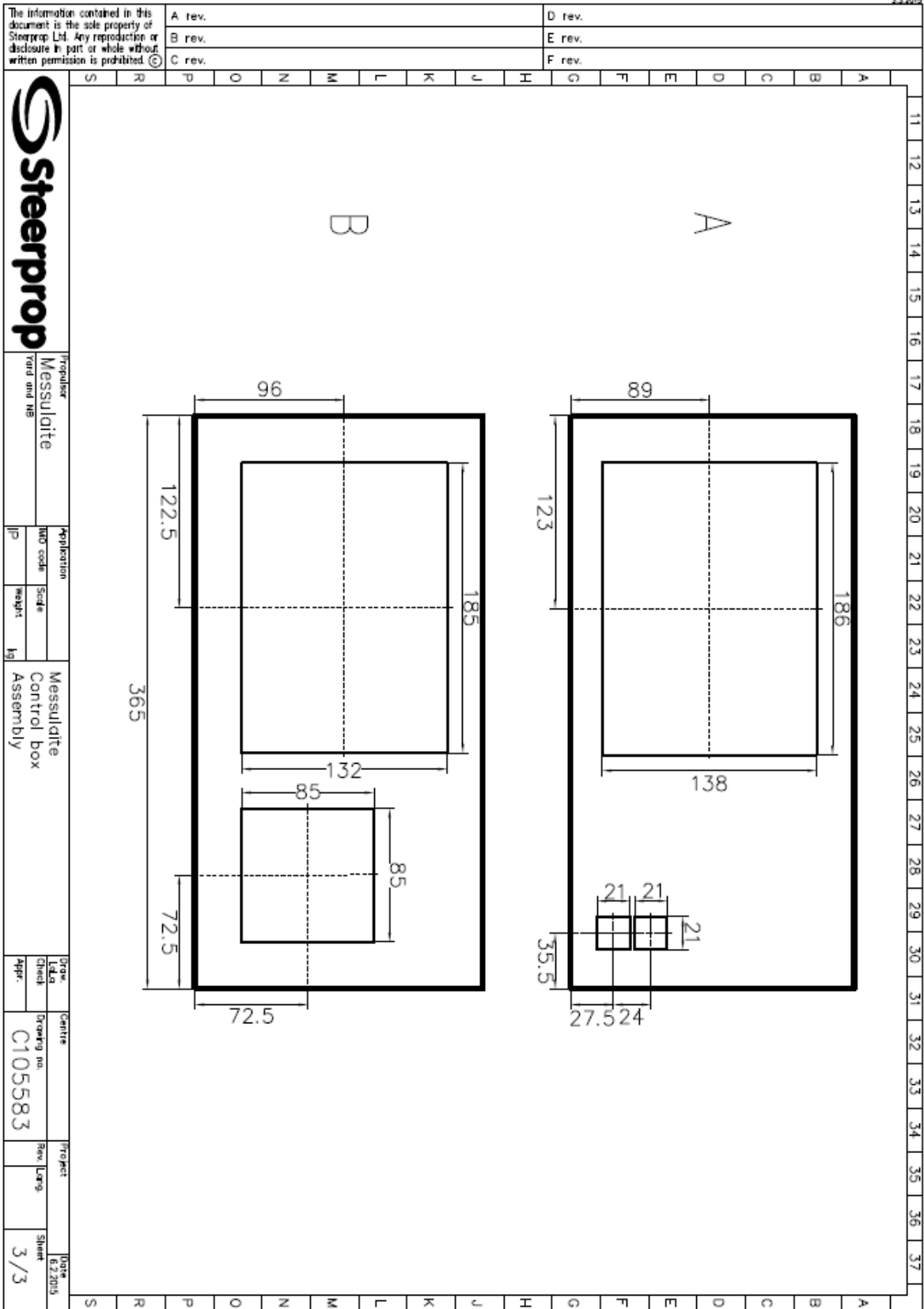
The information contained in this document is the sole property of Steerprop Ltd. Any reproduction or disclosure in part or whole without written permission is prohibited ©		A rev.										D rev.																									
		B rev.										E rev.																									
		C rev.										F rev.																									
		S R Q P O N M L K J I H G F E D C B A										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																									
		Proprietary Messulatte Ford and MB		Application IMO code IP		Scale weight kg		Messulatte Control box Basic design										Drawn Check Appr.		Centre Drawing no. C105583		Project Rev. Log		Date Sheet 6.2.2015 1/3													
		Drawn Check Appr.		Scale weight kg		Messulatte Control box Basic design										Drawn Check Appr.		Centre Drawing no. C105583		Project Rev. Log		Date Sheet 6.2.2015 1/3															

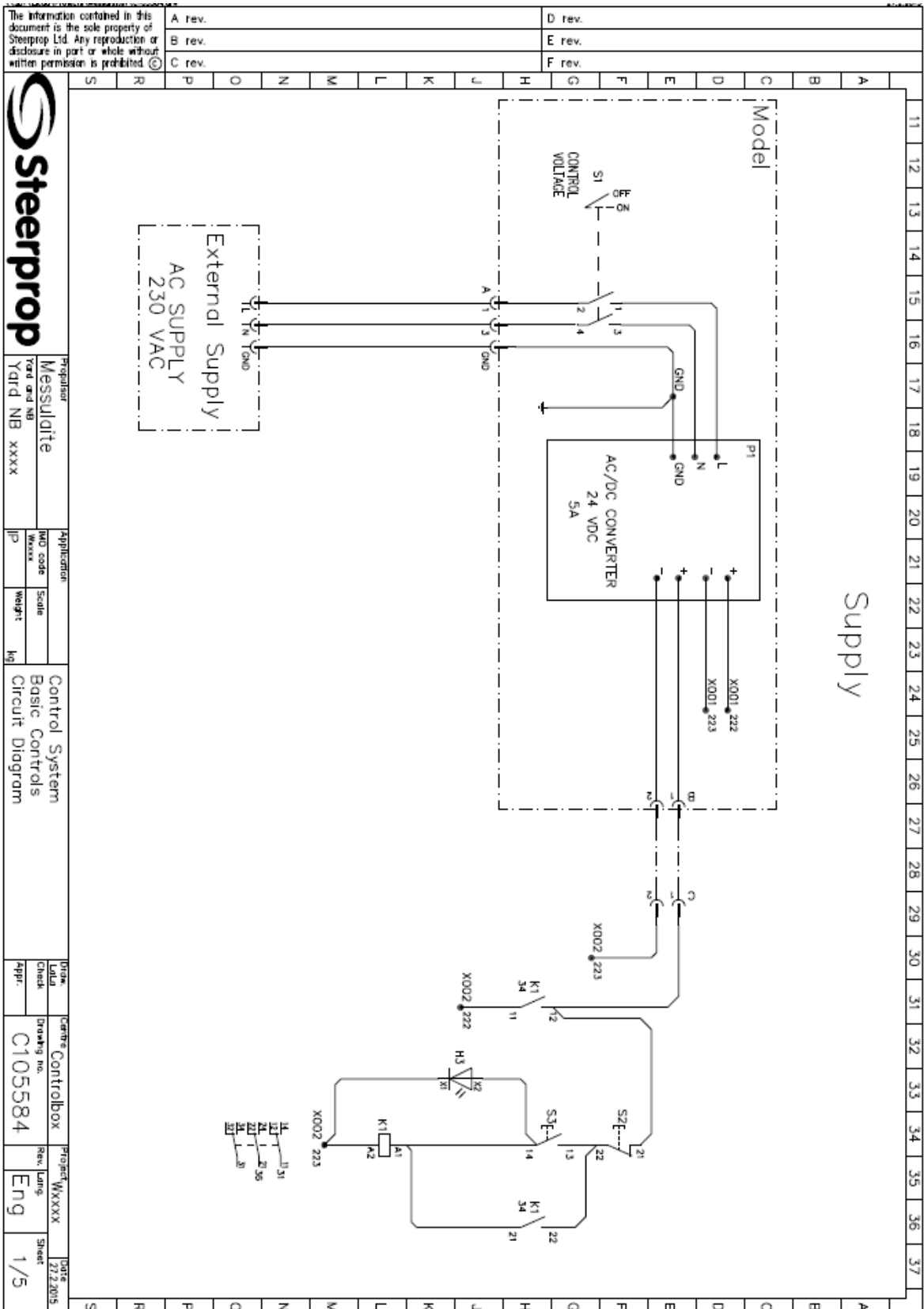
The information contained in this document is the sole property of Steerprop Ltd. Any reproduction or disclosure in part or whole without written permission is prohibited. ©

A rev.	D rev.
B rev.	E rev.
C rev.	F rev.



Proprietary		Application		Messulдите		Drawn		Centre		Project		Date	
Messulдите		Messulдите		Control box		Check		Drawing no.		Rev. / Lays		Sheet	
Form and NB		IP		Assembly		Appr.		C105583				6.2.2015	
		Scale										2 / 3	
		Height											
		Lg											





The information contained in this document is the sole property of Steerprop Ltd. Any reproduction or disclosure in part or whole without written permission is prohibited. ©

A rev.	D rev.
B rev.	E rev.
C rev.	F rev.

S R P O N M L K J I H G F E D C B A



Regulator
Messulidite
YdId NB xxxx

Application
IND code
Scale
IP Weight kg

Control System
Basic Controls
Circuit Diagram

Drawn
Check
Appr.

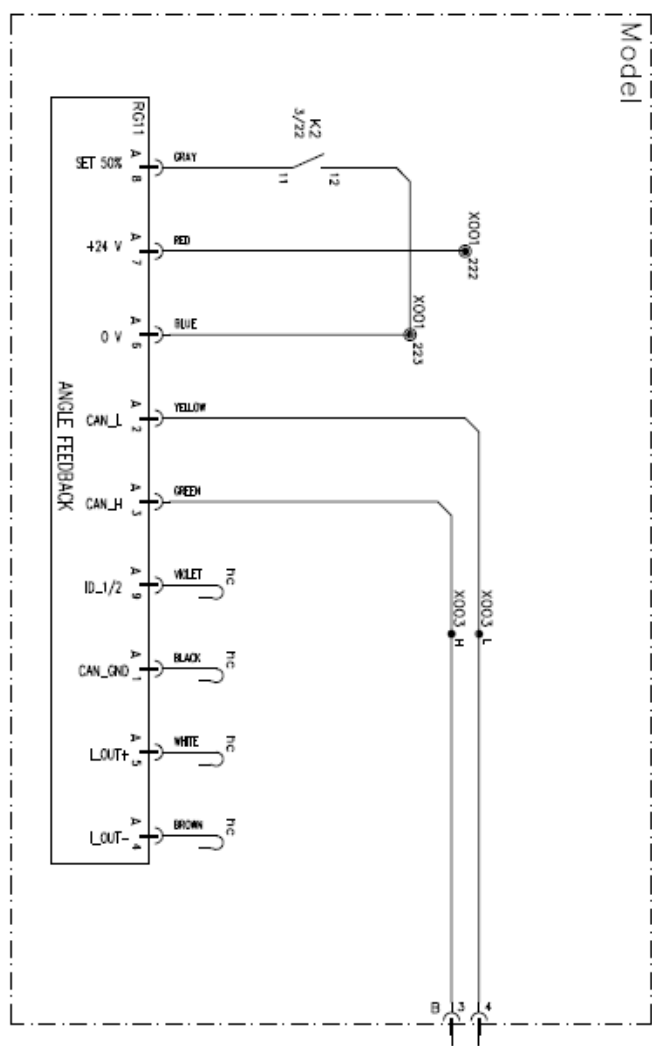
Centre Controlbox
Drawing no.
C105584

Project WXXXX
Rev. Lang
Eng

Date
27.2.2015
Sheet
2/5

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

Feedback



The information contained in this document is the sole property of Steerprop Ltd. Any reproduction or disclosure in part or whole without written permission is prohibited.

A rev.
B rev.
C rev.

D rev.
E rev.
F rev.



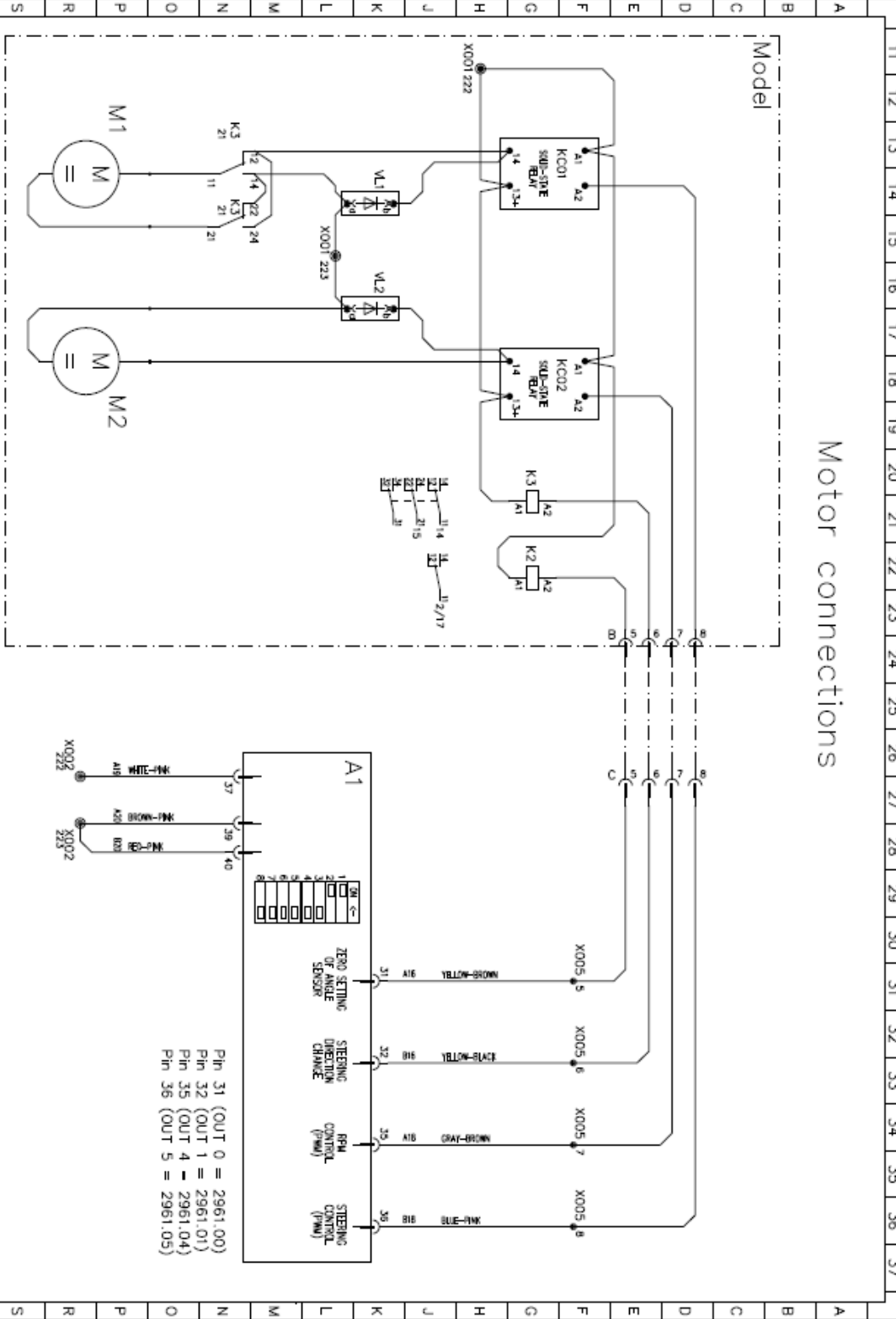
Prepulsator
Messulite
Yard NB xxxxx

Application
IMD code
Scale
Weight
kg

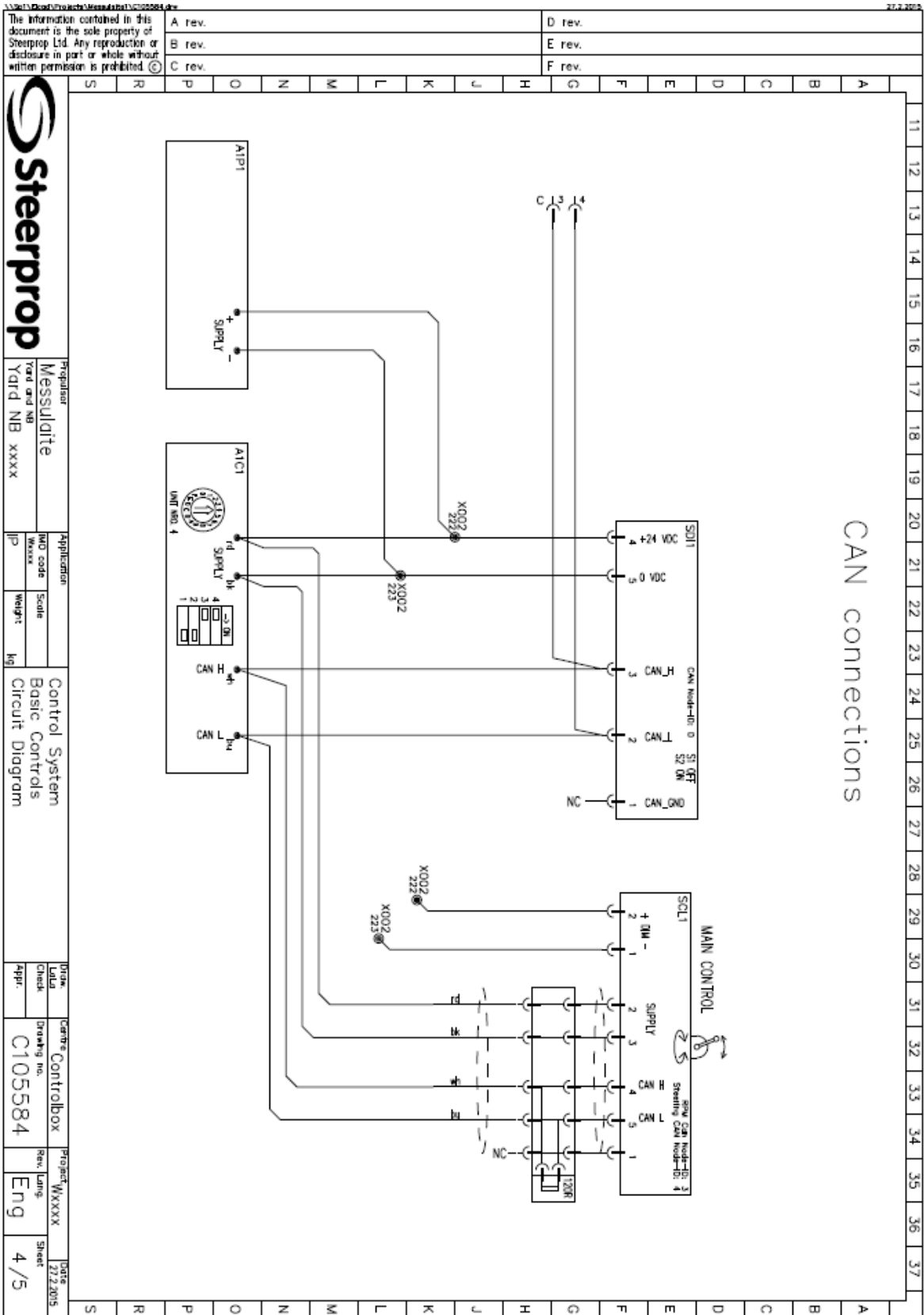
Control System
Basic Controls
Circuit Diagram

Order
Check
Appr.
Drawing no.
C105584

Project
Rev. Lang
Eng
Date
Sheet
3/5



Motor connections



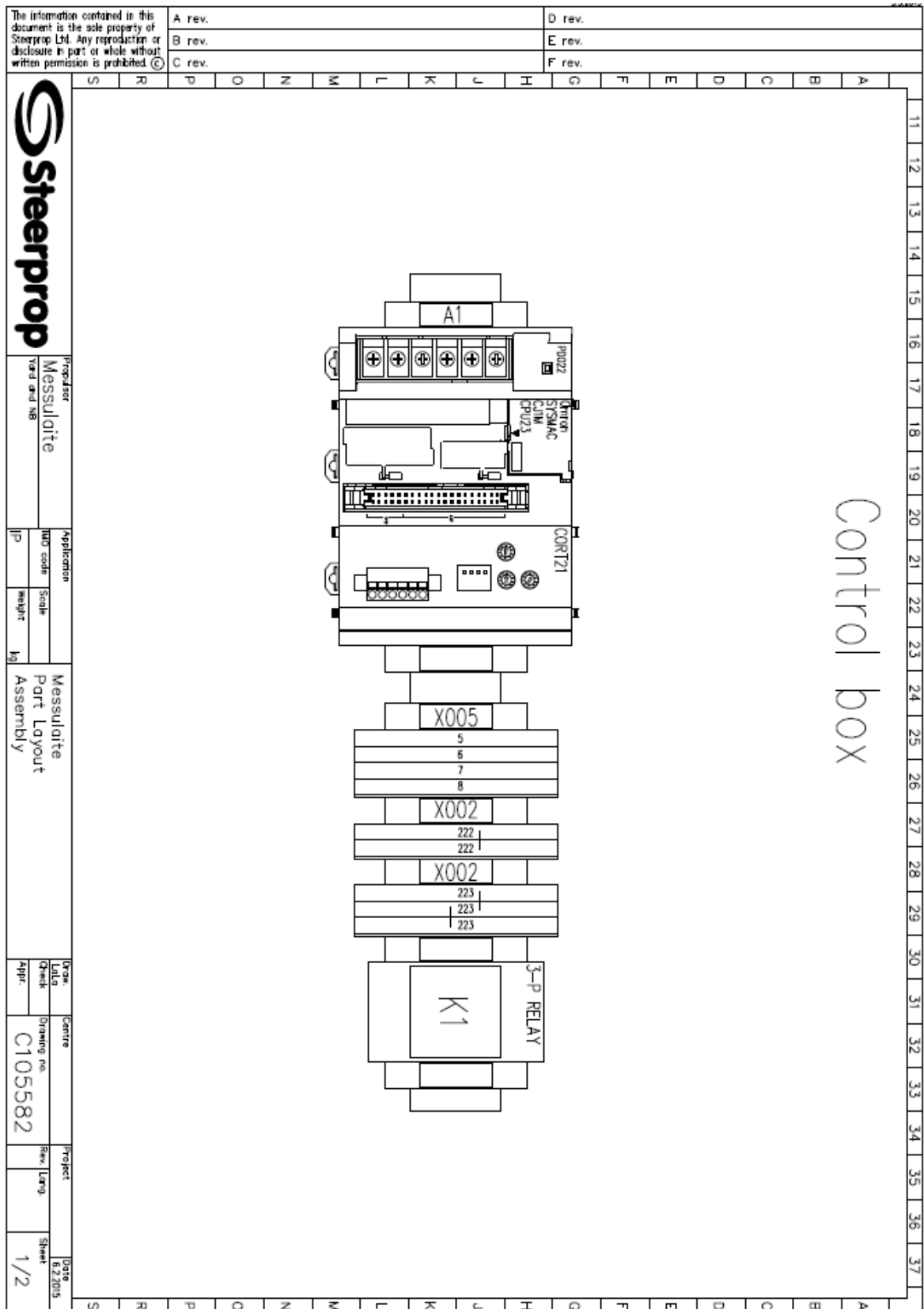
CAN connections

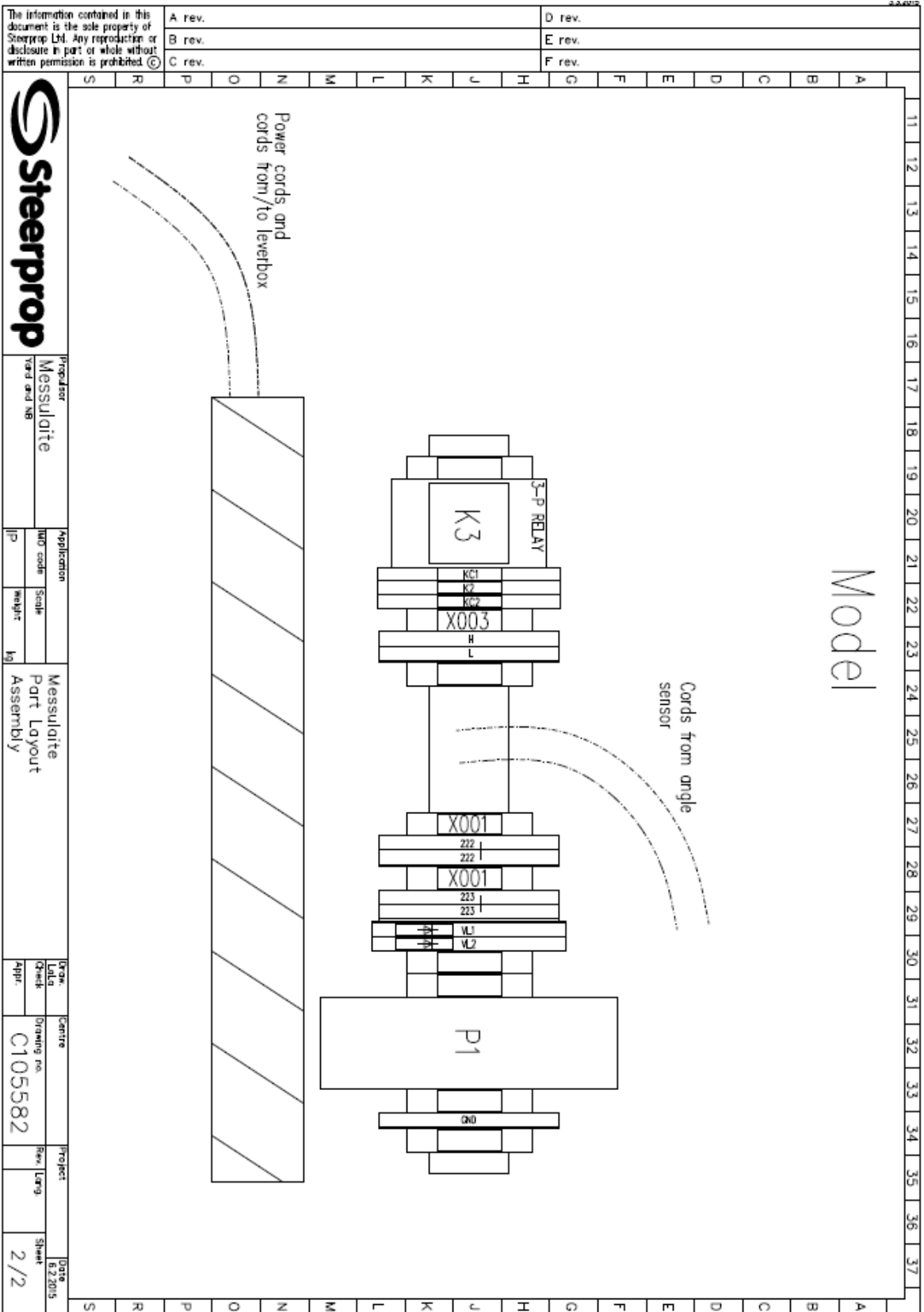
Device list

ID	Device	Description	Technical	Maker	Makers_code	Page
A	Housing	Housing 4 P (90 deg angle)	HAN 3A	Harting	09 20 003 0801	1
A	Connector	Plug connector (male) 4 P	HAN 4A	Harting	09 20 004 2611	1
A	Connector	Plug connector (female) 4 P	HAN 4A	Harting	09 20 004 2711	1
A	Hood	Housing for plug connector	HAN 3A	Harting	19 20 003 1443	1
A1	Connection cable		CJ1M I/O MILL 40	Omron	CBL-M40-200Y-FIN	3
A1	Connection cable	CBL-NT	RS232 port-RS232 Cable 2 m	Omron	CBL-NT	5
A1	CPU unit	10In/6out/100 ns/20 KSteps/640I/O/32 Kdm	Size 23 10DI 6DO	Omron	CJ1M-CPU23	3
A1C1	Bus module		User-defined CAN unit	Omron	CJ1W-CORT21	4
A1P1	Power supply	24VDC,2A(5 VDC)	24 VDC, 2 A	Omron	CJ1W-PD022	4
B	Connector	Plug connector (male) 8 P Quick lock (0.25 mm2 - 1.5 mm2)	HAN 8D-M-QL	Harting	09 36 008 2632	1,2,3
B	Hood	Housing for plug connector	HAN 3A	Harting	19 20 003 1443	1,2,3
B	Housing	Housing 4 P (90 deg angle)	HAN 3A	Harting	09 20 003 0801	1,2,3
B	Connector	Plug connector (female) 8 P Quick lock (0.25 mm2 - 1.5 mm2)	HAN 8D-F-QL	Harting	09 36 008 2732	1,2,3
C	Connector	Plug connector (male) 8 P Quick lock (0.25 mm2 - 1.5 mm2)	HAN 8D-M-QL	Harting	09 36 008 2632	1,3,4
C	Housing	Housing 4 P (90 deg angle)	HAN 3A	Harting	09 20 003 0801	1,3,4
C	Connector	Plug connector (female) 8 P Quick lock (0.25 mm2 - 1.5 mm2)	HAN 8D-F-QL	Harting	09 36 008 2732	1,3,4
C	Hood	Housing for plug connector	HAN 3A	Harting	19 20 003 1443	1,3,4
H3	LED lamp	28 VDC/ 14 mA, white	Led lamp T1 3/4 MG 61	EAO	10-2J13.1069	1
K1	Relay	3 contacts, 10 A, 30 VDC, 250 VAC	C3-A30DX24D, 24 VDC	Releco		1
K1	Relay base	11 pin	S3-B	Releco		1
K2	Relay base		PLC-BSP- 24DC/21	Phoenix Contact	29 67 21 9	3
K2	Relay module without relay base	Relay module without relay base	REL-MR-24DC/21	Phoenix Contact	29 61 10 5	3
K3	Relay base	11 pin	S3-B	Releco		3
K3	Relay	3 contacts, 10 A, 30 VDC, 250 VAC	C3-A30DX24D, 24 VDC	Releco		3
KC01	Optical Coupler without relay base	Optical Coupler without relay base	OPT-24DC/ 24DC/ 2	Phoenix Contact	29 66 59 5	3
KC01	Relay base		PLC-BSP- 24DC/21	Phoenix Contact	29 67 21 9	3
KC02	Optical Coupler without relay base	Optical Coupler without relay base	OPT-24DC/ 24DC/ 2	Phoenix Contact	29 66 59 5	3

Device list

ID	Device	Description	Technical	Maker	Makers_code	Page
KC02	Relay base		PLC-BSP-24DC/21	Phoenix Contact	29 67 21 9	3
P1	AC/DC converter	AC/DC power supply 5 A	QUINT-PS/1AC/24DC/5	Phoenix Contact	28 66 75 0	1
	Steerprop absolute encoder	2x4-20mA, 2xCANOpen	SAE-CAN2x2	FSG	2846Z01-257M004E/MH620-III/4-20/CAN	2
RG11	Connection cable	Plug M12 female with angle, 0.5 m cable	RG11 A	FSG	458131	2
S1	Switch	20 A	Door, 4 NO, OFF-ON	Kraus & Naimer	CA10SO-A324-620-FT2	1
S2	Front bezel	Flush mounting, 24x24 mm	Front bezel set 61	EAO	61-9930.0	1
S2	Lamp lens	Red, 24x24 mm	Plastic lens 61 Red	EAO	61-9671.2	1
S2	Contact element	1 NC + 1 NO	Switching element 04 NO+NC	EAO	704.901.5	1
S2	Push button actuator	for flush mounting	Illuminated push button actuator 61	EAO	61-1100.0	1
S3	Contact element	2 NO + lamp, max 250 VAC, 6 A	Switching element 61 for 2 pos.	EAO	61-8420.22	1
S3	Push button actuator	for flush mounting	Illuminated push button actuator 61	EAO	61-1100.0	1
S3	Front bezel	Flush mounting, 24x24 mm	Front bezel set 61	EAO	61-9930.0	1
S3	Lamp lens	Green, 24x24 mm	Plastic lens 61 Green	EAO	61-9671.5	1
SCL1	Steerprop Control Lever	FWD only 0-120°	RSCU-H96 El.shaft	Kwant	12008042	4
SCL1	Cable set	SAC-5P-FS SCO/920/3	RSCU	Phoenix Contact	1538131/920/3	4
SCL1	Terminal resistor	BUS TERMINATOR SAC-5P-M12MS CAN TR	RSCU	Phoenix Contact	1507816	4
SCL1	Connector set	T-CONNECTOR SAC-5P-M12T/2XMM12 VP	RSCU	Phoenix Contact	1541186	4
SD1	Frame	C105084	Frame for Xdi 192 M	Deif		4
SD1	Steerprop Angle & Rpm Indicator	C105084	XDi 192 M	Deif	XDi 192 M	4
SED1	Graphic display	5,7" (320x240), 256 colours, black case	Colour display	Omron	NS5-SQ11B-V2	5
VL1	Component plug	Diode 1N 4007, 1.3 kV, 1 A	ST-1N4007	Phoenix Contact	28 02 32 9	3
VL1	Component terminal	2x4 mm2	ZFK 4-TG	Phoenix Contact		3
VL2	Component terminal	2x4 mm2	ZFK 4-TG	Phoenix Contact		3
VL2	Component plug	Diode 1N 4007, 1.3 kV, 1 A	ST-1N4007	Phoenix Contact	28 02 32 9	3





The information contained in this document is the sole property of Steerprop Ltd. Any reproduction or disclosure in part or whole without written permission is prohibited. ©

A rev.	D rev.
B rev.	E rev.
C rev.	F rev.



Proposer	Messulaitte
IP	Scale
IP	Weight
kg	

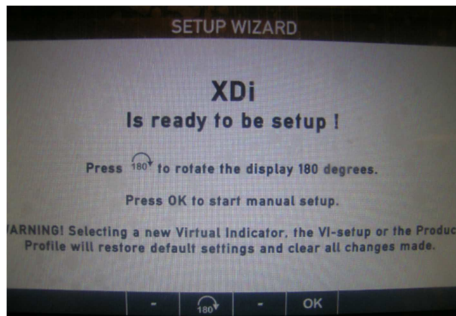
Application	Messulaitte Part Layout Assembly
IP	
kg	

Uran	Centre
Check	Drawing no.
Appr.	C105582
Rev.	Lang.
Sheet	Date
2/2	8/2/2015

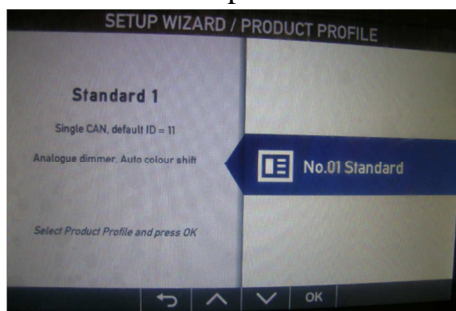
Deif XDi 192M -näytön asetukset:

Näytön ensimmäisen käynnistyksen yhteydessä tulee määrittää näytön asetukset:

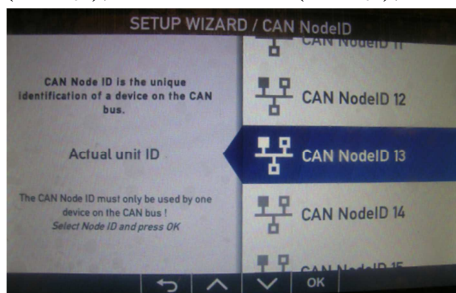
1. Valitaan näytön suuntaus (onko näyttö sijoitettu kohti potkurilaitetta vai poispäin potkurilaitteesta).



2. Määritetään tuoteprofiili.



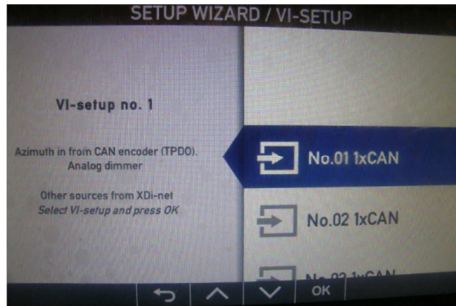
3. Valitaan näytön käyttämä noodinumero (Node-ID). Järjestys esimerkiksi WCP11 = 11 (= B_{hex}), WCP12 = 12 (= C_{hex}), WCP21 = 21 (= 15_{hex}) jne.



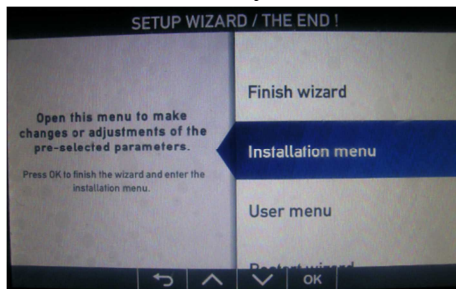
4. Määritetään laityyppi (määrittää näytössä olevan potkurilaitteen kuvan).



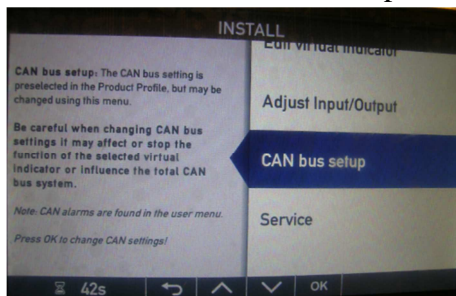
5. Määritetään saapuvan datan protokollat. Logiikan lähettäessä dataa näytölle CAN-väylää pitkin, valitaan vaihtoehdoista No.01 1xCAN. Tämä kertoo näytölle kulma-anturin tiedon saapuvan TPDO-protokollan tunnisteella (Transmit PDO 1 = 180_{hex} + Node-ID). Valitulla vaihtoehdolla muiden indikointien datatulojen protokolla asettuu XDi-net pohjaiseksi.



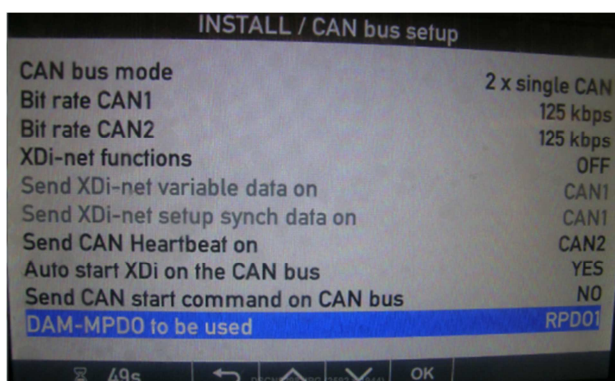
6. Tässä vaiheessa valitaan kohta ”Installation menu”, jossa tehdään muut tarvittavat muutokset CAN-väylän kommunikointiasetuksiin.



7. Valitaan kohta ”CAN bus setup”.



8. Määritetään asetukset seuraavanlaisesti:



CAN bus mode -> **2 x single CAN**, määrittää molemmat CAN-terminaalit käyttöön.

Bit rate -> molempien väylien tiedonsiirtonopeus on **125 kbit/s**.

XDi-net functions -> **OFF**, logiikka lähettää XDi-netin mukaista dataa väylälle näytön sijasta.

Send CAN Heartbeat on -> **CAN2**. Näin logiikalla pystytään kuuntelemaan väylää, ja huomataan, jos jokin kulmanäyttö katoaa väylältä. CAN1 väylälle ei tarvitse sykettä lähettää, sillä väylällä ei kulje kuin potkurin kulmatieto (ei sykkeen kuuntelijaa).

Auto start XDi on the CAN bus -> **YES**, käynnistyessään näyttö aloittaa CAN-väylän kuuntelun ilman erillistä NMT node control -protokollan käynnistyskäskyä.

Send CAN start command on CAN bus -> **NO**, erillistä käynnistyskäskyä ei tarvitse lähettää.

DAM-MPDO to be used -> **RPDO1**. Kertoo näytölle muiden indikointien datan saapuvan Receive PDO 1 -protokollan tunnisteella (RPDO1 = 200_{hex} + Node-ID). Logiikka on määritetty lähettämään viestit tunnisteella 200_{hex} + 7F_{hex} (näyttöjen broadcast-ID).

Asetuksien muuton jälkeen paina paluu näppäimestä, jolloin näyttö kysyy ”Synchronize to network?” -> Paina ”YES”.

9. Viimeisenä valitaan käytettävä kirkkaudensäätö. Kirkkaudensäädön tulokanavan asetuksiin pääsee käyttäjä asetuksista ("User menu" -> "Dimmer setup"). Dimmer source kohdassa määritetään onko kirkkaudensäätö hoidettu analogisena (analog) vai CAN-väylän (XDi-net) kautta.

Käytettäessä kirkkaudensäätöön näyttöryhmiä, valitaan lopuksi näytölle ryhmänumero ("User menu" -> "Dimmer group"). Ryhmänumero määrittää näytön kirkkaudensäädön käyttämän instanssinumeron.

1. Important Notice:

- These instructions are included for the both the exhibitor's and the model's safety.
- As the exhibitor is responsible for the model, failure to follow these instructions will have consequences, both financial (if the model is damaged) and otherwise.

2. General Instructions:

- The equipment must not be moved while the wiring is connected!
- Do not touch the wiring and components under the model under any circumstances!
- Do not remove the cover or any components of the control box!

3. Control box:



4. Installing the Exhibition Model:

- Set the model and the control box on sturdy platforms. Ensure that there is no danger of either falling.
- Attach the male connector of the grey intermediate cable to the lower connector (A) on the model. Attach the female connector on this cable to the connector (C) on the control box.
- Attach the female connector of the black power cable to the upper connector (B) on the model.
- The connectors will connect without force when they are in the right alignment. Ensure that the screws on the connectors are visible before connecting them. **Do not use excessive force to connect the cables!**
- Clip on the latches on the connectors to lock them into place.
- Check that the wiring is under not under any mechanical stress (e.g. compression, stretching, etc.)
- Connect the black power cable to the power socket (100 – 240V, 45 -60Hz).
- Turn the power switch on the side of model (D) to the 'I' position.
- Press the start button on the control box.
- Let all equipment fully power up before continuing.

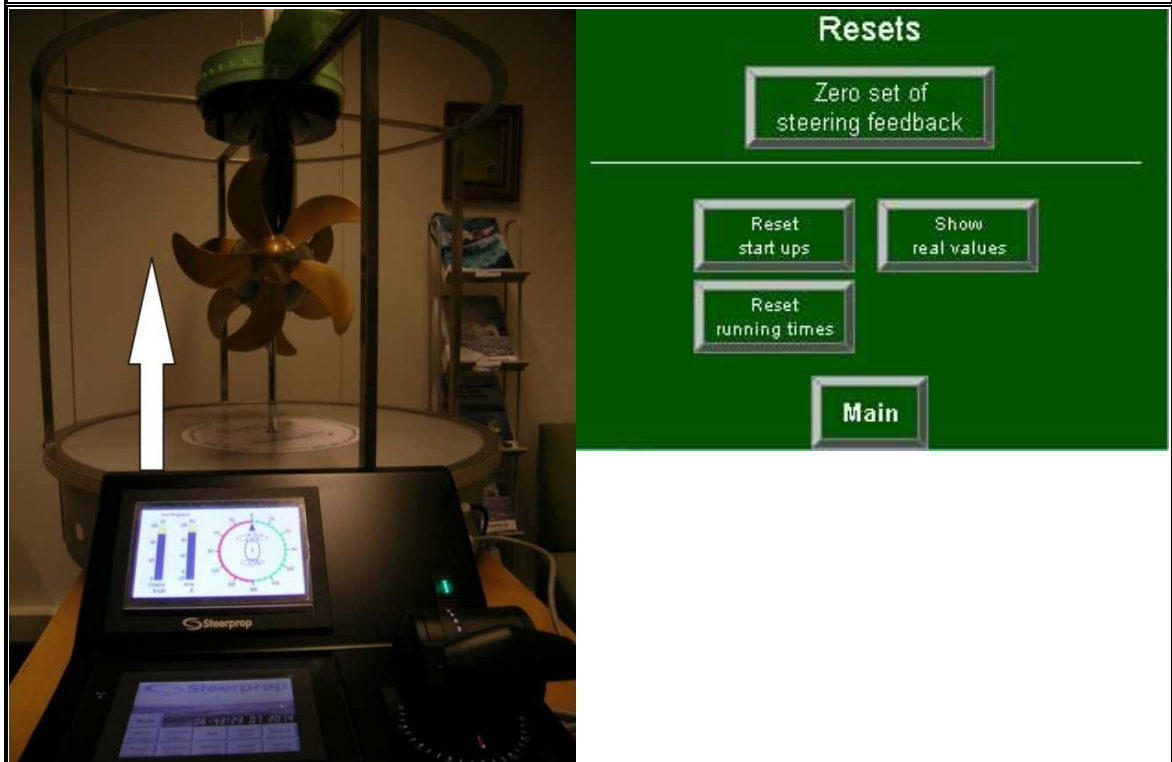


5. Alarms:

- Press the 'Alarm pages' virtual button on the service display and make sure that there are no active alarms. If there are any active alarms, follow the procedure lined out in the 'Troubleshooting' section of this manual.

6. Setting the direction:

- Select 'Manual control' on the 'Mode' pages of the Service display.
- Turn the model to face away from the control box with the control lever.
- The smaller 5-bladed rear propeller should face the control box.
- When the model has stopped turned, select 'Service settings' on the main page of the Service display.
- The password is 'sp'.
- Press the 'Zero set of steering feedback' on the Service settings page.
- The service display will ask you to confirm this with 'Are you sure?'
- Press 'Yes'.
- The Control lever and model should now be set with the same Zero direction.
- Now the model is ready for exhibition use.



7. After the exhibition is over:

- Select "Manual Control" on the Service display. Turn the RPM setting on the control level to 0% and the control lever steering angle to 0.
- When the propellers and the model have stopped turning, press the 'Off' button on the control box.
- Turn the power switch on the model to the 'O' position.
- Remove the black power cord from the power socket.
- Open the latches on the grey and black cords.
- DO NOT USE ANY FORCE TO OPEN THE LATCHES!
- Gently remove the cables from the connectors.
- Pack the control box and the model in their own transportation boxes. The cables should be put in the same box as the model.

8. Troubleshooting:

In every alarm or problem situation, turn the power off on the control box and the model, check ALL the cable connections and turn the power on again.
After turning the power on, check the service display for any active alarms.
If there are no active alarms, proceed to '**Setting the Direction**' section in this manual.
If there are active alarms, see the descriptions below to see what functionality the model has lost.

- "**Cs1 lever position signal failure**"

Manual Control: Only RPM control is available. Steering angle reference indication is unavailable.

Ecp Control: Functions as normal.

Exhibition mode: Functions as normal.

- "**Cs1 lever Rpm signal failure**"

Manual Control: Only steering control is available. RPM reference indication is unavailable.

Ecp Control: Functions as normal.

Exhibition mode: Functions as normal.

- **"Active feedback failure"**

Manual Control: Only RPM control is available.

Ecp Control: Functions as normal.

Exhibition mode: Only RPM steering is available.

The angle indicator and service display do not receive steering angle indication.

- **"Can bus communication failure" or "Can bus Off event"**

Manual Control: Unavailable.

Ecp Control: Functions as normal.

Exhibition mode: Only RPM steering available.

Angle indicator: No indication available.

Service display: The steering angle indication and the control level reference indication is unavailable.

- **"Cpu failure"**

Access the "System Status" on the Service display and select "PLC".

Check the "Plc error code".

If the code begins with 0, the model may be used, but some of the functionality has been lost.

If the code begins with 8, the model cannot and must not be used!

Immediately power off the control box and model!

- **"Angle indicator CAN signal failure"**

No indication available in angle indicator.

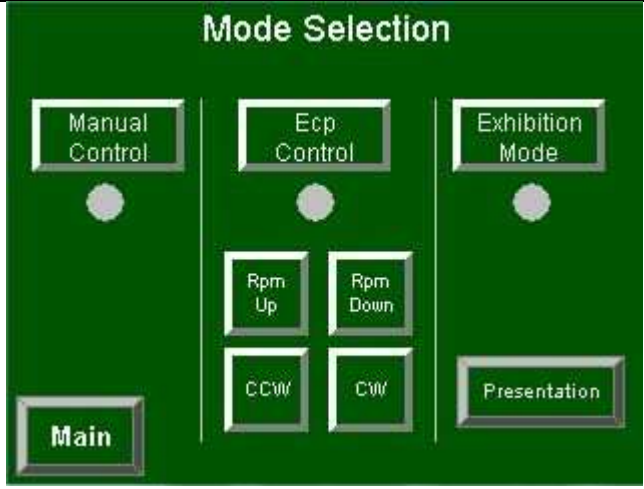
- **"Plc low battery"**

The model can be used normally in all control modes.

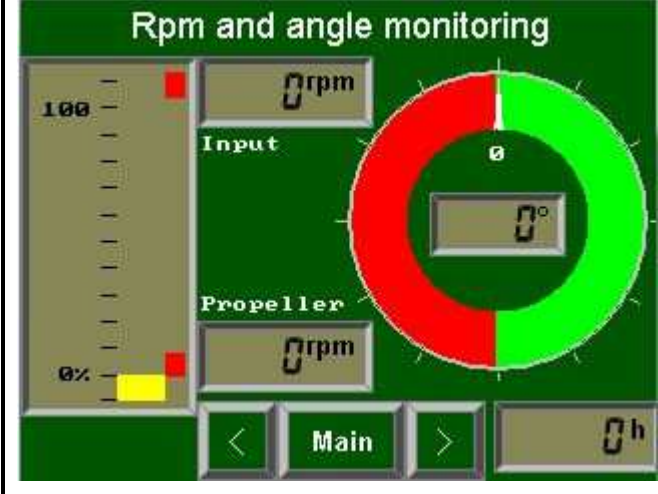
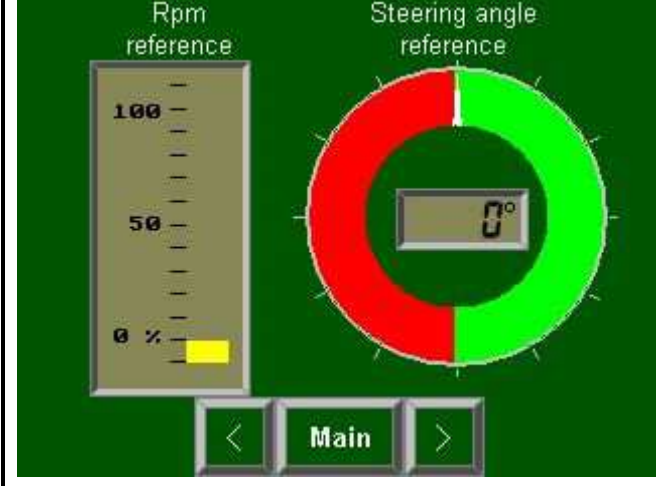

1. MAIN PAGE

	<p>MAIN PAGE</p> <p>Pages related to the model's operation:</p> <ul style="list-style-type: none"> -Mode -Meter displays -Trend graphs -System Status -System Setup -Alarm pages
	<p>Example pages:</p> <ul style="list-style-type: none"> -Start -Control station -Control inputs -Back Up -Service settings <p>(Service settings page is meant only for the exhibitor's use)</p>
	<p>NAVIGATION PUSH BUTTON</p> <ul style="list-style-type: none"> < Page back "Main" Main page > Page forward

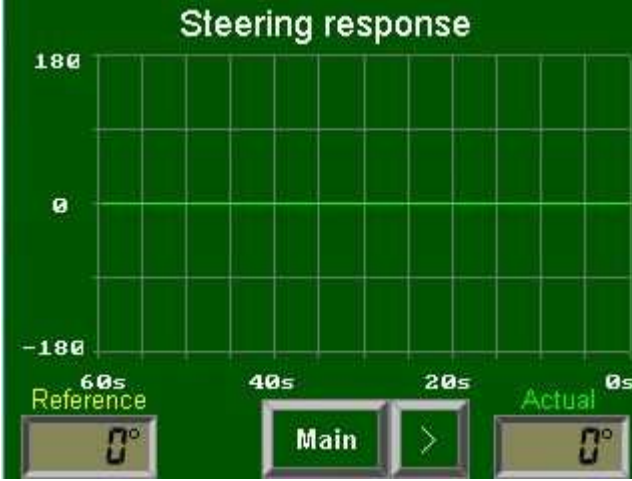
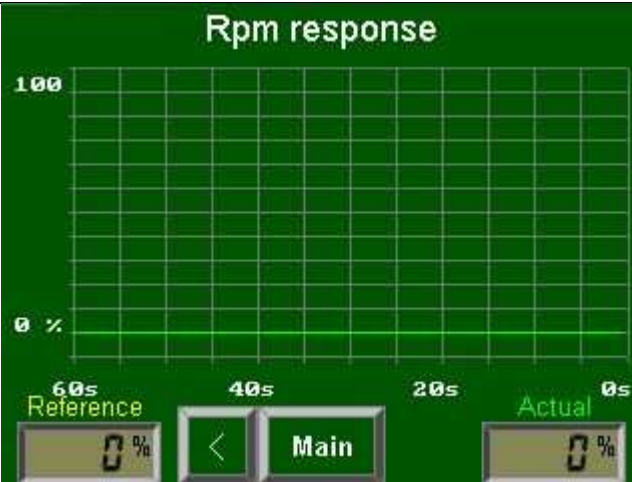
2. MODE PAGE

	<p>MODE PAGE is used to select the control mode.</p> <p>-Manual Control The model is controlled with the control lever.</p> <p>-Ecp Control (Engine Control Panel) The model is controlled with the "Rpm Up", "Rpm Down", "CW", "CCW" virtual buttons in the service play.</p> <p>-Exhibition Mode The model is controlled by the program.</p> <p>-Presentation will set the Steerprop presentation to run on the display. The 'Start' virtual button on the page will re-start the presentation.</p>
---	--

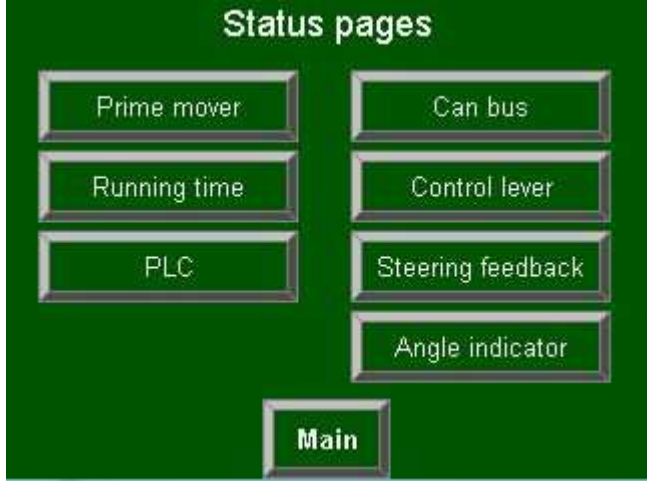
3. METER PAGES

	<p>RPM AND ANGLE MONITORING</p> <p>The meter page includes following information:</p> <ul style="list-style-type: none"> - Steering angle - Propeller rpm - Prime mover rpm - Running hours
	<p>REFERENCES</p> <ul style="list-style-type: none"> - Rpm reference indication - Steering angle reference indication
	<p>OTHER</p> <ul style="list-style-type: none"> -Rpm of Prime mover -Rpm of Propeller -Steering angle turning speed indication -Vessel speed indication <p>Notice! If the vessel's speed exceeds 14 knots, then a +/-20° steering angle limitation is activated. The steering speed will also be limited to 0,5 rpm. (There are no steering limitations in Exhibition mode.)</p>

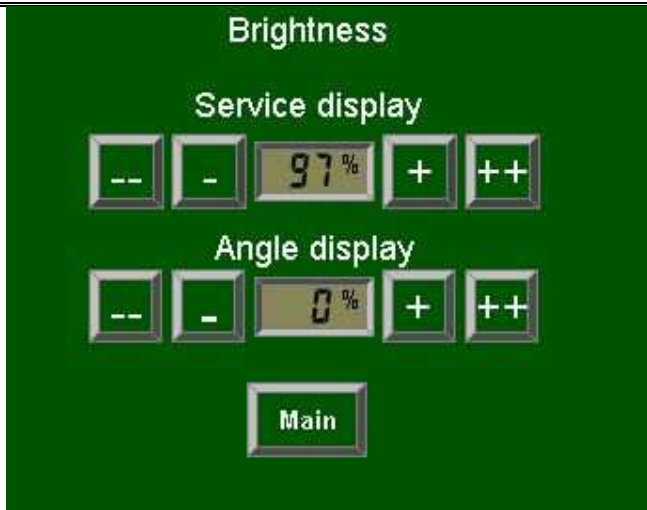
4. TREND GRAPHS

 <p>The 'Steering response' graph shows a coordinate system with a vertical axis ranging from -180 to 180 and a horizontal axis representing time from 0s to 60s. The graph area is currently empty. Below the graph, there are two digital displays: the left one is labeled 'Reference' and shows '0°', and the right one is labeled 'Actual' and also shows '0°'. A central control panel includes a 'Main' button and a right-pointing arrow.</p>	<p>STEERING RESPONSE</p> <p>Shows steering reference and actual steering position diagrams.</p>
 <p>The 'Rpm response' graph shows a coordinate system with a vertical axis ranging from 0 to 100 and a horizontal axis representing time from 0s to 60s. The graph area is currently empty. Below the graph, there are two digital displays: the left one is labeled 'Reference' and shows '0 %', and the right one is labeled 'Actual' and also shows '0 %'. A central control panel includes a 'Main' button and a left-pointing arrow.</p>	<p>RPM RESPONSE</p> <p>Shows rpm reference and actual rpm value diagrams.</p>

5. STATUS PAGES

	<p>MAIN STATUS PAGE</p> <p>Prime mover: Rpm reference, Rpm actual value, Prime mover load condition.</p> <p>Running time: Rpm range has divided into small ranges each having own running counter. All counters are summed to one total running counter.</p> <p>PLC: Status, start-up counter and error code indication of PLC.</p> <p>Can bus: Status of Can bus.</p> <p>Control lever: Status of control lever.</p> <p>Steering feedback: Status of feedback.</p> <p>Angle indicator: Status of the indicator.</p>
---	--

6. SETUP PAGE

	<p>SETUP PAGE</p> <ul style="list-style-type: none"> -Service display brightness setting -Angle indication brightness setting
---	--

7. ALARM PAGES

	<p>ACTIVE ALARM LIST Shows all active alarms.</p> <p>ALARM HISTORY Alarm history with time. Red block on left means that alarm has arose and green block means, alarm has disappeared.</p> <p>ALARM HISTORY Alarm history with number of occurred. Red block on left means that alarm has arose and green block means, alarm has disappeared.</p>
	<p>ALARM MESSAGE Whole alarm message can be seen by touching alarm row.</p>