

Hydraulisyylinterin testipenkin ohjausjärjestelmä

Antti Hietala

Opinnäytetyö
Toukokuu 2012

Automaatiotekniikan koulutusohjelma
Teknologiayksikkö





Tekijä(t) HIETALA, Antti	Julkaisun laji Opinnäytetyö	Päivämäärä 07.05.2012
	Sivumäärä 56	Julkaisun kieli Suomi
	Luottamuksellisuus () saakka	Verkojulkaisulupa myönnetty (X)
Työn nimi Hydraulisynterinin testipenkin ohjausjärjestelmä		
Koulutusohjelma Automaatiotekniikan koulutusohjelma		
Työn ohjaaja(t) STRÖM, Markku		
Toimeksiantaja(t) Control Express Finland Oy, Lindholm, Janne		
Tiivistelmä <p>Opinnäytetyön aiheena oli ohjausjärjestelmän toteutus hydraulisynterinin testaukseen. Sylintereitä testataan pitkäkestoisissa suorituksissa, joissa niitä kuormitetaan, sekä suoritetaan äkillisiä liikkeitä ja liikesuunnan muutoksia. Järjestelmä mittaa siinä esiintyviä paineita, lämpötilaa sekä tilavuusvirtausta ja tallentaa niitä myöhempiä käyttöä varten.</p> <p>Työ sisälsi sähkösuunnittelun sekä järjestelmässä käytettävän ohjelmoitavan logiikan ohjelmoinnin ja järjestelmän valvomon toteutuksen. Ohjelmoitavaksi logiikaksi valittiin ABB:n AC500-sarjaan loogiikka ja valvomotietokoneeksi Control Express Finland:n valmistama CEF 2x-teollisuustietokone. Tietojen tallennus tapahtuu valvomotietokoneella, jolle ohjelmoitava logiikka lähettää mittaustiedot UDP-protokollaa käyttämällä.</p> <p>Projekti aloitettiin käymällä läpi ohjausjärjestelmältä vaadittavat toiminnot sekä muut vaatimukset tilaajan kanssa. Seuraavaksi tehtiin sähkösuunnittelu ja ohjauskeskuksen layout-suunnittelu. Tämän jälkeen siirryttiin suunnittelemaan ja toteuttamaan ohjelmaa ohjelmoitavalle logiikalle. Viimeisessä vaiheessa tehtiin valvomo. Testikoneikon käyttöönotto viivästyi alkuperäisestä aikataulusta. Pelkän ohjausjärjestelmän toimintaa pystyttiin kuitenkin testaamaan laajasti, joten sen toiminta suunnitellulla tavalla pystyttiin todentamaan.</p> <p>Kun testikoneikon käyttöönotto myöhemmin on tehty, tullaan sitä todennäköisesti vielä laajentamaan lisäämällä testattavien sylintereiden lukumäärää sekä mittauksia. Tästä johtuen ohjausjärjestelmää tullaan myöhemmin kehittämään vastaamaan laajennusten asettamia vaatimuksia.</p>		
Avainsanat (asiasanat) Koneautomaatio, ohjausjärjestelmä, ohjelmoitava logiikka, PLC, valvomo		
Muut tiedot		



Author(s) HIETALA, Antti	Type of publication Bachelor's Thesis	Date 07.05.2012
	Pages 56	Language Finnish
	Confidential () Until	Permission for web publication (X)
Title Designing Control system for testing hydraulic cylinders		
Degree Programme Degree programme of Automation engineering		
Tutor(s) STRÖM, Markku		
Assigned by Control Express Finland Oy, Lindholm, Janne		
Abstract <p>The aim of the thesis was to design a control system for test machinery of hydraulic cylinders. The machinery will be used to test the cylinders for long periods of time, under load and sudden changes in the direction of movements. The control system will measure specific physical quantities, such as pressure, temperature and total flow in the system and record them for further use.</p> <p>The project included the electrical design of the system, programming of the PLC (Programmable Logic Controller) and creating interface for the system. The programmable logic controller from ABB's AC500-series was selected to be used in the system, with the CEF 2X industrial computer to host the SCADA (Supervisory Control And Data Acquisition). The programmable logic controller collects the data from the sensors installed in the system and sends it to the SCADA computer using the UDP protocol.</p> <p>At the beginning of the project a meeting was held, where the requirements for the control system were discussed. The next step was to do the electrical design for the system and the layout design of the control panel. This was followed by planning and programming of the PLC. After this was done, the SCADA was created. Due to delays in the schedule, the commissioning of the test machinery was postponed into the future. However, this made it possible to do extensive testing of the control system. The system met the requirements set at the beginning of the project.</p> <p>After the test machinery has been commissioned, there are future plans to develop it further to include more cylinders for testing and more measurements. This will lead to a demand of designing the control system further to meet the new requirements.</p>		
Keywords Test machinery, machine automation, control system, PLC, Programmable logic controller, SCADA		
Miscellaneous		

SISÄLTÖ

1	JOHDANTO	4
1.1	Toimeksiantaja.....	4
1.2	Projektin tavoitteet.....	4
1.3	Projektin vaatimukset.....	5
2	TEORIA.....	7
2.1	Ohjelmoitavat logiikat	7
2.2	Prosessisuureiden mittaukset	12
2.2.1	Painemittaus.....	12
2.2.2	Lämpötilamittaus	14
2.2.3	Virtausmittaus	15
2.2.4	Siirtymän mittaus	17
2.3	Pulssinleveysmodulaatio	19
2.4	User Datagram Protocol	20
3	OHJAUSJÄRJESTELMÄN TOTEUTUS	21
3.1	Projektin aloitus.....	21
3.2	Suunnittelu	22
3.3	Turvarele.....	23
3.4	Ohjelmointi	25
3.5	Proportionaaliventtiilin ohjaus.....	31
3.6	Käyttöliittymä	32
3.7	Mittaustietojen keruu ja tiedonsiirto	35
3.8	Tietojen käsittely	38
3.9	Antureiden skaalaus	38
4	YHTEENVETO	39

4.1 Tulokset	39
4.2 Pohdinta.....	40
LÄHTEET.....	41
LIITTEET	43
Liite 1. Hydraulikoneikon periaatekaavio	43
Liite 2. IO-lista	44
Liite 3. Ohjauskeskuksen Layout.....	45
Liite 4. Jännitteenjakelu.....	46
Liite 5. Piirikaaviot	49

KUVIOT

KUVIO 1. Ohjelmoitavan logiikan toimintaperiaate.....	8
KUVIO 2. Tulosignaalin havaitseminen logiikassa	9
KUVIO 3. Keskeytyksen lisääminen	10
KUVIO 4. Paineen mittaustavat.....	13
KUVIO 5. Virtauksen profiili.....	16
KUVIO 6. 5-bittinen koodianturi.....	18
KUVIO 7. Magneettinen kulma-anturi.....	19
KUVIO 9. Tavun suuruisen signaalin pulssinleveysmodulaatio.....	20
KUVIO 10. Turvareleen kytkentäkuva	24
KUVIO 11. Turvareleen esimerkkikytkentä.	24
KUVIO 12. Sekvenssin ohjaustiedot sisältävä taulukko	29
KUVIO 13. Työvaiheen määrittäminen.....	30
KUVIO 14. Pääikkuna	33
KUVIO 15. Sekvenssin luomisikkuna	34
KUVIO 16. Tiedontallennuksen toiminta.....	36
KUVIO 17. Tiedonsiirron toiminta	37

TAULUKOT

TAULUKKO 2. Digitaalitulon jännitealueet.....	8
TAULUKKO 3. Profibus DP:n tiedonsiirtonopeudet.....	11
TAULUKKO 4. Lähetettävät tiedot.....	35

1 JOHDANTO

1.1 Toimeksiantaja

Control Express Finland Oy (CEF) on Savonlinnassa toimiva, vuonna 1993 perustettu yritys. Yrityksen päätoimialana ovat teollisuustietokoneet, puolustusteollisuuden standardien mukaiset tietokoneet, tietoliikennelaitteet sekä ruggeroidut näytöt. Yrityksessä toimii lisäksi automaatio-osasto, joka suunnittelee ja toteuttaa teollisuuden automaatiosovelluksia, pääasiassa pesuloiden kuljetinjärjestelmiä, sekä paperiteollisuuden rullan pakkauskoneiden automatisointeja (Control Express Finland Oy, 2012).

Vuonna 2011 yrityksen liikevaihto oli noin 12 miljoonaa euroa. Yrityksessä työskentelee yhteensä noin 50 henkilöä, joista noin neljäsosa tuotekehityksen parissa.

1.2 Projektin tavoitteet

Projektissa toteutettiin hydraulisynterien testaukseen käytettävän hydraulikoneikon ohjausjärjestelmä. Ohjausjärjestelmä toimitetaan Waratah OM:lle, joka valmistaa itse koneikon. Valmista koneikkoa tullaan käyttämään hydraulisynterien testauksissa, joissa sylinteriä kuormitetaan pitkäkestoisesti, erilaisissa olosuhteissa, sekä suoritetaan äkillisiä liikkeitä sekä suunnanmuutoksia. Testien aikana koneikon antureiden mittaustietoja tallennetaan erilliseen tiedostoon, joita voidaan analysoida. Työ käsitti järjestelmän sähkösuunnittelun, sekä ohjausjärjestelmän toteutuksen. Alkuperäisen suunnitelmaan kuului lisäksi käyttöönotto, mutta aikataulumuutosten takia tämä jäi opinnäytetyön ulkopuolelle.

1.3 Projektin vaatimukset

Projektin alussa pidettiin palaveri asiakkaan kanssa testikoneikkoon liittyen. Asiakkaan kanssa keskusteltiin siitä, mitä toimintoja testipenkiltä halutaan, mitä sillä halutaan mitata sekä miten ohjausjärjestelmä toteutetaan. Laittevalinnat koneikkoon tekee Waratah OM. Ohjausjärjestelmälle asetettiin seuraavat vaatimukset:

Sylinteritestipenkin toiminta

Penkillä on tarkoitus testata hydraulisylintereitä. Testattavia sylintereitä on aluksi vain yksi, mutta tulevaisuudessa järjestelmä pystyttävä laajentamaan usealle. Alla oleva spesifikaatio on yhdelle sylinterille. Ohjausjärjestelmällä on pystyttävä ohjaamaan sylinterin liikettä sekä painetta, tunnistamaan asema sekä tallentamaan kytkettyjen antureiden signaalit. Järjestelmä tulee olla käyttäjän ohjelmoitavissa toiminto toiminnolta ja sen on pystyttävä toistamaan automaattisesti ohjelmoitua työsykliä sekä ilmoittamaan toimintahäiriöistä.

Ohjausjärjestelmällä toteutettavat toiminnot:

- Sylinterin ohjaus suuntaventtiilillä sisään/ulos
- Sylinterin paineen ohjaus proportionaaliventtiilillä
- Kuormitusventtiilin ohjaus auki/kiinni
- Kuormitusventtiilin vapautus
- Kuormitussylinterin paineakkujen poiskytkentä
- Ulkoisen epäpuhtauden ruiskutus
- Hydraulikoneikon sammutus/käynnistys
- Hälytys/vikailmoitus

Ohjausjärjestelmän tallentamat tiedot:

- Paineet
- Virtaus
- Lämpötilat

- Syklit ja matka, nopeus
- Anturisignaalien automaattinen ja manuaalinen tallennus
 - o Aseteltava tallennus taajuus (1- 1000ms)
 - o Automaattisen toiminta: tallentaa aseteltavalla välillä tiedot. Pitää esim. 10 min puskuria 1ms näytteenottovälillä muistissa. Aseteltavien hälytysrajojen ylittyessä tallentaa tapahtumahetkestä 10 min molempiin suuntiin.

Ohjausjärjestelmän lähdöt:

- 9 kpl ON/OFF 24VDC/2A
- 2 kpl Proportionaali 24VDC/2A
- Antureiden jännitesyötöt 12VDC/2A
- Antureiden jännitesyötöt 24VDC/2A

Ohjausjärjestelmän tulot:

- Hätäseis -toiminto
- Virtausmittaus 1kpl jännitesignaali 0,5-4,5V
- Painemittaus 6kpl jännitesignaali 0,5-4,5V
- Lämpötilamittaus 2 kpl (ilma + öljy). Jännitesignaali
- Sylinterin asematieto

Ohjausjärjestelmän käyttö

- Lähdöt vapaasti ohjelmoitavissa graafisen käyttöliittymän kautta
- Proportionaalivirroille aseteltava arvo
- Toiminnot ohjelmoitavissa 10 ms välein
- Kaikille antureille ohjelmoitavissa hälytysrajat tietyssä syklin vaiheessa, Esimerkiksi tilavuusvirta tarkastetaan tunnin välein tietyillä toiminnoilla
- Toimintojen ohjelmoitavissa sekvenssin on oltava helposti monistettavissa
- Ajastus toiminto, eli testipenkki käynnistyy ja sammuu automaattisesti tiettyinä aikoina

2 TEORIA

2.1 Ohjelmoitavat logiikat

Ohjelmoitavalla logiikalla (PLC, Programmable logic controller) tarkoitetaan yksittäistä, itsenäisesti toimivaa automaatiojärjestelmää, jonka vastuulla on tietty osaprosessi. Ohjelmoitava logiikka suorittaa tehtävänsä saamiensa toimintaohjeiden, eli siihen ladatun sovellusohjelman mukaan (Kippo & Tikka, 2008, 54). Ohjelmoitavat logiikat kehitettiin korvaamaan teollisuudessa aikaisemmin käytössä ollut reletekniikka. Releiden ongelmana oli komponenttien suuri tarve, sekä jo olemassa olevaan ohjausjärjestelmään tehtävien muutosten hankaluus, koska kaikki ohjaukset piti tehdä kiinteästi langoittamalla.

Toimiakseen ohjelmoitava logiikka tarvitsee informaatiota prosessin tilasta. Informaatio tuodaan logiikalle tulojen kautta, jotka voivat olla analogisia, tai digitaalisia. Analogisia tuloja ovat esimerkiksi lämpötila-anturit, jonka mittaama arvo voi saada mitä tahansa arvoja nollan ja anturin mitta-asteikon ylärajan väliltä. Signaali voi olla joko virta- tai jänniteviesti. Teollisuudessa käytetään useimmiten virtaviestiä, koska se on vähemmän altis sähköisille häiriöille. Virtaviesti on joko 0-20mA tai 4-20mA, joista jälkimmäinen on yleisempi. 4-20mA:n viestin avulla voidaan todentaa viestin oikeellisuus, sekä tämä mahdollistaa antureiden 2-johdin kytkennän. Menetelmää kutsutaan eläväksi nollassi. Jänniteviestinä voidaan käyttää 0-10V:n, 2-10V:n, 0-5V:n sekä 1-5V:n alueita. (Sivonen, 2001, 23).

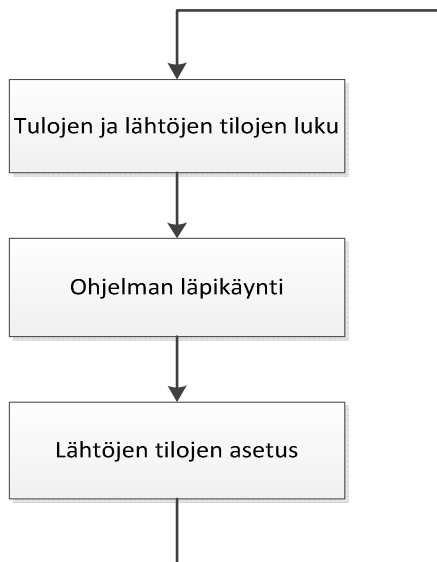
Digitaaliset signaalit ovat ON-OFF–tyyppisiä diskreettejä signaaleja. Erilaiset kytkimet, esimerkiksi lähestymiskytkimet käyttävät digitaalisia viestejä. Ohjelmoitavalle logiikalle anturilta tuleva viesti tulkitaan olevan joko päällä tai pois riippuen signaalin jännitteestä. Seuraavassa taulukossa on esitetty ABB:n DX522 I/O-moduulin jännitealueet, joilla digitaalitulo toimii (ABB, 2011).

TAULUKKO 1. Digitaalitulon jännitealueet

-3V..5V	Pois
5V..15V	Määrittelemätön
15V..30V	Päällä

Toimintaperiaate

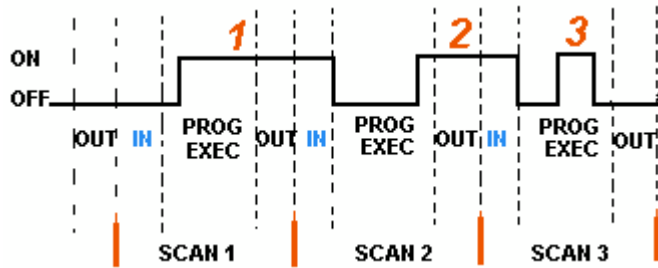
Ohjelmoitava logiikka selaa muistiin kirjoitetun ohjelmaa kiertävällä syklillä. Aluksi kaikkien logiikan tulojen ja lähtöjen tila luetaan. Tämän jälkeen tallennettu ohjelma käydään läpi ohjelmarivi kerrallaan. Kun koko ohjelma on käyty läpi, logiikka toteuttaa lähtöjen ohjauksen määritellyllä tavalla. Koska myös lähtöjen tila luettiin syklin alussa, ohjataan vain niiden lähtöjen tilaa, joissa on tapahtunut muutoksia.



KUVIO 1. Ohjelmoitavan logiikan toimintaperiaate

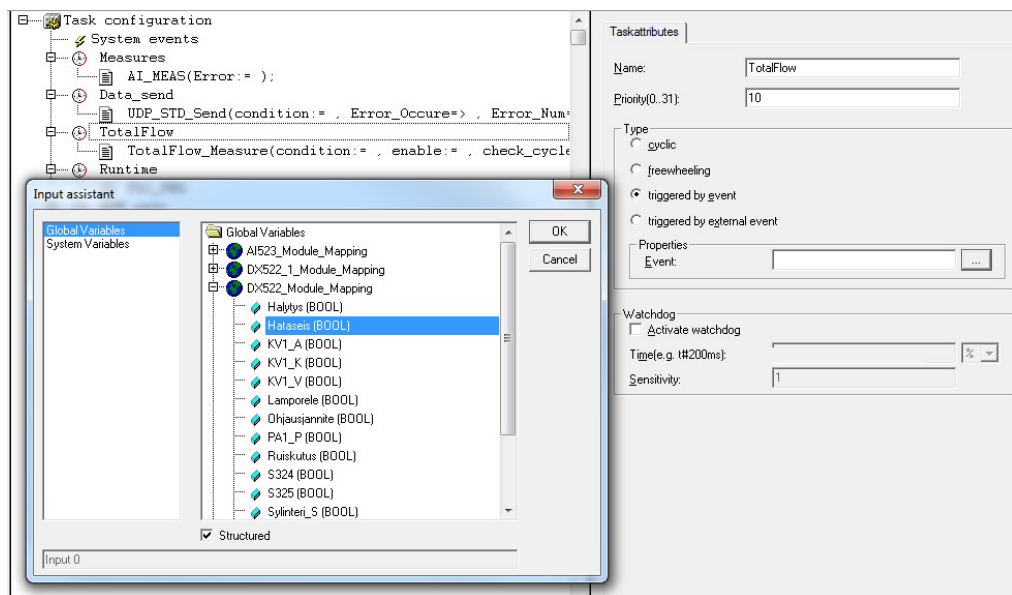
Koska ohjelmoitavan logiikan toimintaperiaatteen mukaan järjestelmän tulot luetaan vain ohjelmakierron alussa, pitää tämä ottaa huomioon ohjelmaa suunniteltaessa. Nykyisin käytössä olevat ohjelmoitavat logiikat suorittavat koko ohjelmakierron muutamissa millisekunneissa, mutta vanhemmissa järjestelmissä tämä saattaa kestää jopa 100 millisekuntia. Kuvio 2 esittää, miten yhden tulon kolme signaalia käsitellään ohjelmoitavassa logiikassa. Ensimmäinen signaali luetaan vasta toisessa ohjelmakierrossa, koska sen tila muuttuu, kun ensimmäisessä ohjelmakierrossa (SCAN 1) on jo

siirrytty tulojen ja lähtöjen tilan lukemisesta ohjelman suoritukseen. Vastaavasti toinen signaali luetaan vasta kolmannessa ohjelmakierrossa (SCAN 3). Kolmatta signaalia sen sijaan ei havaita ollenkaan, koska se ei ole päällä kun tulojen tilaa luetaan (Melore, 2011).



KUVIO 2. Tulosignaalin havaitseminen logiikassa. (<http://www.plcs.net/chapters/respdetails6.htm>)

Kuten aikaisemmin mainittiin, nykyisin toimitettavien ohjelmoitavien logiikoiden ohjelmakierrot ovat millisekuntien pituisia, jolloin todennäköisyys edellä mainittuun tilanteeseen on pienentynyt. Ohjelmaan voidaan myös lisätä keskeytystoimintoja, joissa jonkin tuloa seurataan jatkuvasti. Jos signaalin tila muuttuu, riippumatta siitä, missä kohdassa ohjelmankierto on, keskeytetään pääohjelman suoritus ja ohjelmoitava logiikka suorittaa keskeytykseen liitetyt toiminnot. Tämän jälkeen palataan takaisin pääohjelmaan ja normaalin ohjelmakiertoon. Seuraavassa kuviossa esitetään keskeytyksen lisääminen CoDeSys-ohjelmistossa.



KUVIO 3. Keskeytyksen lisääminen

Ohjelmointi

Ohjelmoitaville logiikoille julkaistiin 1990-luvulla kansainvälinen standardi, IEC61131, jonka kolmas osa (IEC61131-3) määrittelee paitsi ohjelmoitavissa logiikoissa käytettävät ohjelmointikieliset, myös käytettävät datatyypit. Standardin mukaisia ohjelmointikieliä on viisi, Relekaavio- tai tikapuuohjelmointi (LAD), Logiikkakaavio-ohjelmointi (FBD), Käskylista (IL), Strukturoitu teksti (ST) sekä Sekvenssiohjausohjelmointi (SFC) (Real Time Automation, 2009). Useimmissa tapauksissa eri valmistajien ohjelmoitavia logiikoita voidaan ohjelmoida vain joillakin näistä ohjelmointikielistä. Lisäksi eri valmistajilla voi olla itse kehitettyjä ohjelmointikieliä.

Ennen henkilökohtaisten tietokoneiden yleistymistä PLC:n ohjelmoiminen tapahtui erillisellä ohjelmointilaitteella, mutta nykyisin tämä tapahtuu ohjelmointiin tarkoitettuilla ohjelmistoilla. Vanhemmissa ohjelmoitavissa logiikoissa ohjelman siirtoon käytetään RS232-protokolla sarjaliikennettä, mutta nykyisin PLC:n ja tietokoneen välinen kommunikointi tapahtuu yleensä ethernetin avulla.

Kenttäväylät ja hajautus

1970-luvulla alkaneen kenttäväylien kehityksen tarkoituksena on ollut tuottaa ratkai-

suja, joilla automaatiojärjestelmien toiminnallisuus pystytään viemään lähemmäksi ohjattavaa prosessia. Kenttäväylää käyttämällä voidaan tulo- ja lähtömoduulit viedä mahdollisimman lähelle toimilaitteita jotka niihin on kytketty. Tätä järjestelyä kutsutaan hajautetuksi I/O:ksi.

Nykyisin käytössä olevat kenttäväylät pohjautuvat useisiin eri standardeihin. Kenttäväyläratkaisuihin on pyritty saamaan yhtenäinen standardi, mutta tässä ei ole onnistuttu. Ongelmana on automaation käyttö laajasti elämän eri osa-alueilla, eikä tiettyyn käyttöön suunnitellut kenttäväylät välttämättä sovellu toisenlaiseen käyttöön.

Yleisimpiä nykyään käytössä olevia kenttäväyläratkaisuja ovat

- PROFIBUS (Process Field Bus)
- PROFINET
- ASi (Actuator Sensor interface)
- CAN (Controller Area Network)
- Foundation Fieldbus
- HSE (High Speed Ethernet)

PROFIBUS- ja PROFINET-väylät ovat käytössä erityisesti Euroopassa, Foundation Fieldbus ja HSE taas Pohjois-Amerikassa. ASi on alemman tason väylä, jota käytetään nimensä mukaisesti yksittäisten toimilaitteiden tai antureiden kytkemiseen järjestelmään. CAN-väylää käytetään erityisesti ajoneuvoissa.

PROFIBUS-väylä koostuu kahdesta eri tarkoitukseen suunnitellusta ratkaisusta, DP ja PA. PROFIBUS PA-väylää käytetään mittatietojen lähettämiseen ja se vaatii aina PROFIBUS DP-väylän johon se liitetään DP/PA-kouplerin tai linkkerin avulla (Häkkinen, 2010). Profibus-väylän tiedonsiirtonopeus riippuu väylän pituudesta. Seuraava taulukko esittää siirtonopeuden riippuvuutta väyläsegmentin pituudesta.

TAULUKKO 2. Profibus DP:n tiedonsiirtonopeudet (ABB, 2011)

Siirtonopeus	Kaapelin pituus
9,6 - 187,5 kbit/s	1200 m
500 kbit/s	400 m
1,5 Mbit/s	200 m
3 - 12 Mbit/s	100 m

PS501 Control Builder Plus-ohjelmointiympäristö

ABB:n ohjelmoitavien logiikoiden ohjelmoimiseen käytetty ympäristö on nimeltään PS501 Control Builder Plus. Ohjelmisto sisältää Hardware-konfigurointiin tarkoitettua osion, sekä saksalaisen 3S-Software –nimisen yrityksen kehittämän CoDeSys (Controller Development System) V2.3:n.

CoDeSys on järjestelmäriippumaton IEC61131-3-standardin mukainen ohjelmointiympäristö. Tällä hetkellä se on käytössä yli 200 eri yrityksessä, ABB:n lisäksi mm. Bosch Rexroth AG:ssa ja Beckhoff Automation GmbH:ssa, ollen yksi yleisimmistä automaatiojärjestelmien ohjelmointiin käytetyistä työkaluista. Ohjelmointisovelluksen lisäksi CoDeSys -ympäristö sisältää käyttöliittymäsuunnitteluun tarkoitettua Visualization–työkalun, sekä oman tiedonsiirtoon tarkoitettua serverin ohjelmoitavien logiikoiden ja ohjelmointiympäristön väliseen kommunikointiin. Serverin kautta on mahdollista hakea tietoa ohjelmoitavalta logiikalta muiden standardien, kuten OPC-liittymän kautta (3s-software, 2012).

2.2 Prosessisuureiden mittaukset

2.2.1 Painemittaus

Paineen mittaus on yksi yleisimmistä prosesseissa mitattavista suureista. Paine tarkoittaa nesteen tai kaasun aiheuttamaa voimavaikutusta pinta-alayksikköä kohden.

Paineen suuruus lasketaan kaavalla

$$p = \frac{F}{A}, \text{ jossa}$$

p = paine,

F = voima ja

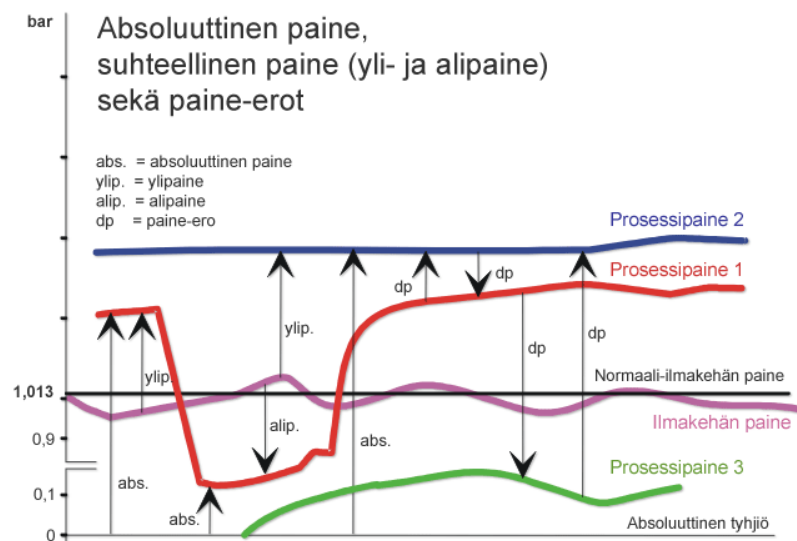
A = pinta-ala

Paineen yksikkönä käytetään SI-järjestelmän pascal, Pa.

$$1 \text{ Pa} = 1 \frac{\text{N}}{\text{m}^2}$$

Esiintyvät paineen arvot ovat usein 100 000 pascalin luokkaa, käytetään usein pascalin kerrannaisia kilopascaleita tai megapascaleita.

Sivosen (2001, 56) mukaan paine mitataan pääasiassa suhteellisena mittauksena, jossa ko. tilan painetta mitataan suhteessa ilmanpaineeseen. Toinen mahdollisuus on myös mitata painetta absoluuttisesti, jolloin vertailutilassa on tyhjiö. Kuviossa 4 havainnollistetaan eri tapoja, joilla painetta voidaan mitata.



KUVIO 4. Paineen mittaustavat
(KnowPulp–oppimisympäristö, 2011)

Paineanturilla saatuja arvoja voidaan ilmaista paikallisesti, jolloin anturi voi olla mekaanisesti toimiva, jolloin saatu paineen arvo muutetaan voimaksi tai liikkeeksi. Toinen tapa on muuttaa painetieto sähköiseksi viestiksi, jolloin anturi voidaan liittää prosessinohjauksessa käytettävään järjestelmään.

Primäärieliimenä käytetään kalvoa, johon paine aiheuttaa joko elastisen muodonmuutoksen, mekaanisen siirtymän tai jännitystilän. Kalvon toisella puolella on kalvon mekaaniseen muutokseen reagoiva tuntoelin. Joskus kalvon toisella puolella on hydraulinen neste, jonka välityksellä paine kohdistuu toiseen kalvoon, jonka muutokseen tuntoelin reagoi. Jos lähettimellä halutaan mitata paine-eroa, täytyy sähköinen tuntoelin sijoittaa kahden kalvon väliin. Absoluuttipainelähetimissä täytyy toiseen kalvoon vaikuttaa tyhjiö (KnowPulp–oppimisympäristö, paineen mittaaminen, 2010).

Tuntoelin, jolla kalvon muutosta seurataan, voi olla joko kapasitiivinen, induktiivinen, tai venymäliuska-anturi. Lisäksi käytössä on pietsosähköistä ja pietsoresistiivistä menetelmiä hyödyntäviä antureita.

2.2.2 Lämpötilamittaus

Yleisin prosessisuureiden mittauksista on lämpötilan mittaus. Lämpötila vaikuttaa suuressa määrin prosessiaineiden käyttäytymiseen ja täten prosessin toimintaan, joten lämpötilan mittaaminen on prosessin turvallisuuden toiminnan kannalta tärkeää.

Lämpötilan mittaamiseen on kehitetty useita eri tapoja, jotka pohjautuvat pääosin eri aineiden käyttäytymiseen lämpötilan muuttuessa. Aineiden lämpölaajeneminen, sekä ominaisvastuksen riippuvuus lämpötilasta pystytään havaitsemaan yksinkertaisilla menetelmillä. Esimerkkinä elohopeamittarit, joissa käytettävä elohopea reagoi voimakkaasti lämpötilan muutokseen mittaputkessa. Toinen lämpölaajenemista hyväksikäyttävä lämpötilanmittaus on bimetallilämpömittari, jossa kahdesta eri metallista valmistettu spiraali taipuu lämpötilan vaikutuksesta (Sivonen, 2000, 36).

Metallin resistanssi kasvaa lämpötilan kasvaessa. Metallista valmistettuja vastuselementtejä käytetään yleisesti teollisuuden lämpötilan mittauksissa. Yksi yleisimmistä antureissa käytettävistä metalleista on platina, joista valmistetut anturit tunnetaan yleisesti nimellä Pt100. Platinan ominaisvastus 0 °C lämpötilassa on sata ohmia, josta anturin nimi tulee. Platina on jalometalli, joten se säilyttää ominaisuutensa myös korkeammassa lämpötiloissa. Vastuselementtien etuna on lisäksi edullinen hinta. Resistanssin muutos ei ole suoraan lineaarinen, joten vastuksille tehdään taulukot, joista resistanssi tietyissä lämpötiloissa voidaan tarkistaa (Pihkala, 2004, 44).

Metallista valmistettujen elementtien sijaan voidaan käyttää puolijohteista valmistettuja termistoreita. Kuten metalleilla, myös puolijohteiden resistanssi riippuu vallitsevasta lämpötilasta. Puolijohteiden lämpötilakerroin on usein negatiivinen, päinvastoin kuin metalleilla. Termistoreja on kahta eri tyyppiä, NTC (Negative Temperature

Coefficient) ja PTC (Positive Temperature Coefficient). NTC-termistorin resistanssi pienenee ja PTC-termistorin kasvaa lämpötilan noustessa (Pihkala, 2004, 46).

Verrattuna metalleihin, termistorien lämpötilakertoimet ovat monta kertaa suurempia. Tämän takia niiden avulla pystytään mittaamaan pienempiä lämpötilanmuutoksia. Toisaalta niiden mittausalue on rajatumpi.

Hydraulikoneikossa käytetään Hydac:n valmistamaa ETS 7346-A-anturia, jonka lämpötilanmittaus perustuu anturin sisällä olevaan piistä valmistettuun puolijohdekomponenttiin. Anturille syötetään 12V:n jännitettä, jolloin lämpötilan kasvaessa anturin lähettämä virtaviesti kasvaa.

2.2.3 Virtausmittaus

Virtausmittauksessa mitataan putkistossa tai avonaisessa kanavassa jonkin poikkipinta-alan leikkauksen lävitse virtaavan aineen määrää. Virtauksen määrää voidaan mitata joko tilavuusvirtana, tai massavirtana. Tilavuusvirtauksessa mitataan nimensä mukaisesti sitä tilavuutta, joka kulkee läpileikkauksen läpi, massavirtauksessa taas kulkevan aineen massaa (Pihkala, 2004, 59).

Tilavuusvirran suuruus lasketaan kaavalla

$$\dot{V} = vA$$

jossa \dot{V} = tilavuusvirtaus,

v = virtausnopeus ja

A = poikkipinta-ala

Tilavuusvirran yksikkö on $\frac{m^3}{s}$.

Massavirtaus voidaan laskea tilavuusvirtauksesta, kun tiedetään virtaavan aineen tiheys. näin ollen massavirtaus lasketaan kaavalla

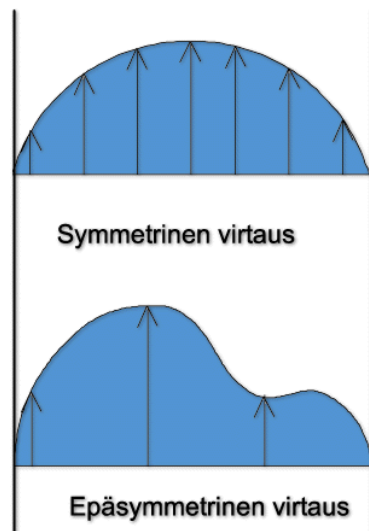
$$\dot{m} = \rho\dot{V}$$

jossa \dot{m} = massavirtaus,

ρ = tiheys ja

\dot{V} = tilavuusvirtaus

Virtausmittauksessa on huomioitava virtauksen profiili. Antureiden toiminnan kannalta kuvion 5 mukainen, symmetrinen virtaus on parempi. Käytännössä virtaus on kuitenkin usein epäsymmetristä. Virtauksen nopeuden mittaaminen tapahtuu yhdessä pisteessä tai keskiarvoista nopeutta putken halkaisijan suuntaisesti. Virtauksen profiilin ollessa symmetrinen putkessa virtaavan aineen keskimääräinen nopeus mitataan oikein. Epäsymmetrisessä virtauksen profiilissa ongelmaksi muodostuukin virtausnopeuden keskimäärän laskeminen (Pihkala, 2004, 59).



KUVIO 5. Virtauksen profiili

Virtausmittauksen mittauspiste tulisi valita putkistossa siten, että virtauksessa esiintyvät häiriöt saadaan minimoitua. Putkiston mutkat, venttiilit ja muut vastaavat aiheuttavat virtaukseen pyörteitä, jotka häiritsevät virtausta ja johtavat virheellisiin mittauksiin. Siksi virtausmittausta ennen ja jälkeen on tulisi olla suorat putkiosuudet, jotta virtaus tasaantuisi. Ennen mittausta olevan suoran putkiosuuden tulisi olla vähintään n. 5 kertaa putkiston halkaisija ja mittauksen jälkeen n. 2 kertaa putken halkaisija.

Virtauksen mittaamiseen on kehitetty suuri määrä eri tapoja. Suuri osa näistä perustuu paine-erojen mittaamiseen putkistossa. Kun virtausaukkoa pienennetään, kasvaa

virtaavaan nesteeseen nopeus ja vastaavasti paine pienenee. Paineen aleneminen on suoraan verrannollinen nopeuden muutokseen. Virtauksen ollessa turbulenttista paine-eron ja virtauksen riippuvuussuhde on neliöllinen (KnowPulp, 2012, Virtauksen mittaaminen). Muita käytössä olevia tapoja on mm. ultraäänimittaukset sekä induktiivisuuteen perustuvat mittaukset.

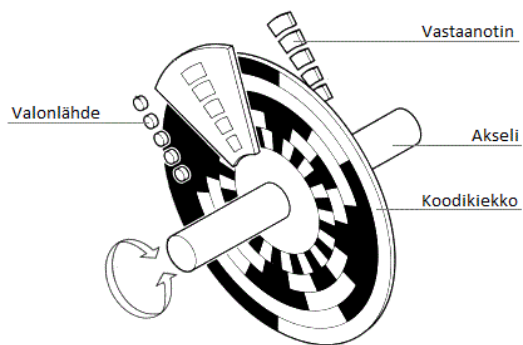
Tässä projektissa käytettävä mittaustapa on ns. turbiinanturi, jossa virtaava aine saa mittauspisteessä olevan turbiinin pyörimään, jonka pyörimisnopeus muutetaan analogiviestiksi (Hydac international, 2011). Kun mittauspisteessä olevan putken poikkipinta-ala tiedetään, saadaan tilavuusvirta selville. Tarkkuutensa vuoksi anturityyppi sopii hyvin suurien neste- tai kaasumäärien mittaukseen (KnowPulp, 2012, Virtauksen mittaaminen).

2.2.4 Siirtymän mittaus

Monissa automaation sovellusten toiminnoissa tarvitaan tarkkaa tietoa eri kappaleiden sijainnista. NC-koneiden ja robottien toiminta on riippuvainen siitä, että ne pysyvät asemoimaan itsensä mahdollisimman tarkasti. Sylinteri liikkuu tietyn akselin suuntaisesti, jolloin sen siirtymä on lineaarista. Linearianturit mittaava nimensä mukaisesti tätä siirtymää, mutta myös kiertyvää anturia voidaan käyttää hyväksi muuttamalla anturin kiertymä liikepituudeksi. Kiertyvää anturia käytetään pääasiassa kulmaliikkeen mittaamiseksi (Keinänen, Kärkkäinen, Metso & Putkonen, 2002, 186).

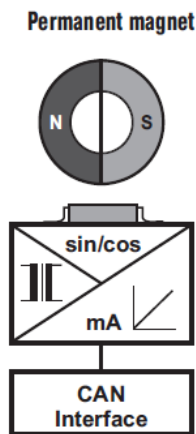
Siirtymän mittaukseen voidaan käyttää joko analogisia tai digitaalisia antureita. Kehityksen myötä nykyisin käytetään pääasiassa digitaalisia antureita, jotka ovat joko pulssi- tai koodiantureita. Analogisten antureiden käyttöä puoltavia seikkoja on melko yksinkertainen rakenne, halpa hinta, sekä niiden parempi vaikeiden olosuhteiden kesto. Anturit voivat olla joko absoluutti- tai inkrementaaliantureita. Absoluuttianturit tunnistavat oman kotiasemansa, eli nollakohdan. Inkrementaalianturit tarvitsevat asemansa määrittämisen apuvälineen, esimerkiksi laskurin. Nollakohta pitää määrittää erikseen ja jos jokin pulsseista jää laskematta, aiheutuu pysyvä virhe (Keinänen ym. 2002, 187). Virhe voidaan korjata ajamalla kappale kotiasemaan, joka tunnistetaan esimerkiksi lähestymiskytkimen avulla, jonka jälkeen laskuri nollataan.

Pulssi- ja koodianturit koostuvat optisesta lähettimestä, vastaanottimesta, sekä näiden välissä olevasta kiekosta. Pulssianturin kiekossa on määräväleihin aukkoja, joista vastaanotin havaitsee lähetetyn valon kiekon pyöriessä. Koodianturissa käytetään sitä vastoin koodikiekkoa. Lineaarista liikettä mitattaessa kiekon sijaan käytetään mittasauvaa (Keinänen ym. 2002, 189). Lukupäitä ja kiekon rivejä lisäämällä anturin tarkkuutta voidaan kasvattaa, esimerkiksi kuvion 6 mukainen, 5-bittinen koodianturi erottaa 32 eri kulma-asentoa, kun 8-bittinen anturi tunnistaa jo 256 kulma-asentoa. Anturi muodostaa vastaanottimien tilojen perusteella mittaviestin joko binääri- tai Gray-koodina.



KUVIO 6. 5-bittinen koodianturi (<http://www.scienceprog.com/using-gray-code-for-rotary-encoders/>)

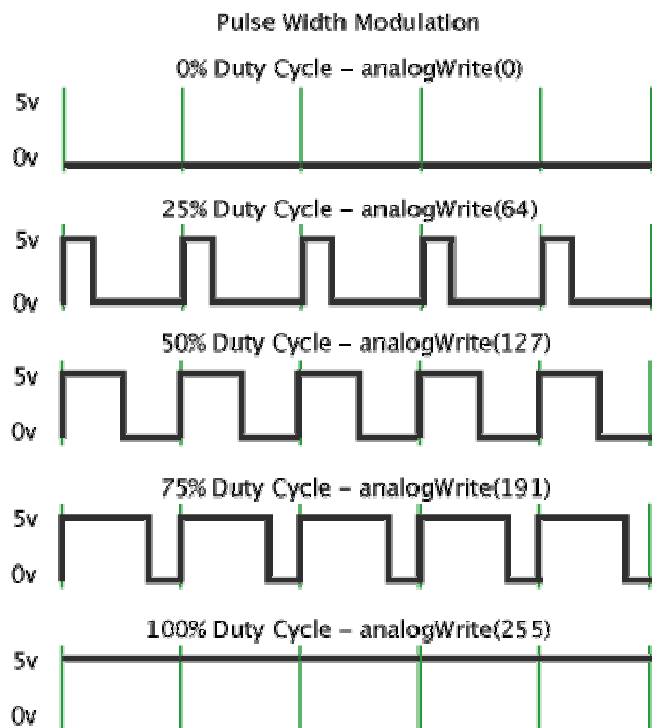
Projektissa käytetään sylinterin aseman mittaamiseen vaijerivetolaitetta, joka koostuu kelalla olevasta metallivaijerista, sekä digitaalisesta anturista, joka perustuu magneettikentän mittaamiseen. Vaijerin kelalle on asennettu kestmagneetti, jonka asennon perusteella anturi laskee siirtymän. Kun magneetti pyörii, pyörii myös sen muodostama magneettikenttä. Tämä muodostaa siniaallon, jonka perusteella voidaan laskea, paljonko magneetti-akseliyhdistelmä on pyörinyt. Yksi jakso tarkoittaa yhtä akselin pyörähdystä (FSG, 2012).



KUVIO 7. Magneettinen kulma-anturi (FSG, 2012).

2.3 Pulssinleveysmodulaatio

Pulssinleveysmodulaatiolla (Pulse Width Modulation, PWM), saavutetaan digitaalisen signaalin avulla analogiasignaalin toiminta. Tekniikka pohjautuu kanttiaallon luomiseen muuttamalla lähetettävän signaalin tilaa päälle ja pois päältä halutulla suhteella. Digitaalisen signaalin jännite on aina sama. Pulssinleveysmodulaatiossa tätä jännitettä kytketään päälle ja pois päältä, jolloin ohjattavalle toimilaitteelle tulevaa jännitteen keskiarvoa saadaan alennettua. Esimerkiksi lähetettäessä toimilaitteelle 24 voltia 50 % ajasta, toimilaitteelle tulevan jännitteen voidaan laskea olevan 12 voltia. Pulssinleveysmodulaatio sopii erityisesti moottoreiden ohjaukseen, koska liikeenergian takia moottorit eivät välittömästi pysähdy, vaikka niiden ohjaussignaalia kytketään pois päältä suurella taajuudella. Kuvio 9 esittää minkä arvon *AnalogWrite*-muuttuja saa eri arvoja sen perusteella, mikä signaalin tilojen suhde on (Hirtzel, T, 2012).



KUVIO 8. Tavun suuruisen signaalin pulssinleveysmodulaatio (<http://arduino.cc/it/Tutorial/PWM>)

Pulssinleveysmodulaatiota voidaan käyttää joko ohjaamalla esimerkiksi lampuja päälle ja pois päältä halutulla suhteella, tai muuttamalla signaali toimilaitteella pyörimisnopeudeksi tai venttiilin aukeamaksi.

2.4 User Datagram Protocol

UDP (User Datagram Protocol) on vastaavanlainen tiedonsiirrossa käytetty menetelmä, kuin TCP (Transmission Control Protocol). TCP pitää sisällään varmistuksia siirretyn datan perillemenosta, kun taas UDP:sta nämä mekanismit puuttuvat kokonaan, joten tarvittaessa luotettavaa tiedonsiirtoa, tulisi käyttää toista protokollaa. Varmistusten puuttuminen tiedonsiirrosta mahdollistaa kuitenkin suuremman siirtonopeuden (ABB, 2011).

Käytettäessä UDP:ta, lähetettävä paketti koostuu osoitetiedosta, sekä itse lähetettävästä datasta. Osoitetieto käsittää 8 tavua, ja se koostuu seuraavista tiedoista, joista

jokaiselle on varattu kaksi tavua:

- Lähdeosoitteen portti
- Kohdeosoitteen portti
- Datan koko
- Tarkistussumma

Jokainen lähetetty paketti käsitellään itsenäisesti, riippumatta muista lähetetyistä paketeista, joten paketit eivät välttämättä saavu lähetetyssä järjestyksessä. Tämän takia on huomioitava se, että vastaanotettu data pystytään laittamaan oikeaan järjestykseen (ABB, 2011).

3 OHJAUSJÄRJESTELMÄN TOTEUTUS

3.1 Projektin aloitus

Järjestelmässä käytettäväksi ohjelmoitavaksi logiikaksi valittiin ABB:n AC500-sarjan logiikka. Tähän päädyttiin, koska CodeSys -ympäristö tarjoaa valmiita ratkaisuja tiedon tallentamiseen ja mallintamiseen, esimerkiksi valmiita sovelluksia trendikäyrän piirtoon. Aluksi tiedontallennus oli tarkoitus tehdä logiikassa esimerkiksi muistikortille, mutta projektin edetessä tämän tavan tallennusnopeus todettiin riittämättömäksi. Käytettävä logiikka oli kuitenkin jo tilattu, sekä logiikan suorituskyky vastasi muuten järjestelmän tarpeita, joten ohjelmoitavaa logiikkaa ei nähty tarpeelliseksi vaihtaa. Ongelma tallennusnopeuden kanssa ratkaistiin siten, että ohjelmoitava logiikka lähettää mittaustiedot valvomotietokoneessa olevalle erikseen tehtävälle sovellukselle, joka suorittaa mittaustietojen tallennuksen.

Valvomotietokoneeksi valittiin Control Express Finland Oy:n valmistama CEF 2X-mallin teollisuustietokone. Järjestelmässä olevan proportionaaliventtiilin ohjaukseen valittiin aluksi Bernecker & Rainer-nimisen yrityksen PWM-ohjaukseen tarkoitettu

CM8323-moduuli. Tätä ei kuitenkaan ole mahdollista ohjata Profibus-väylän kautta, joten moduuliksi vaihdettiin saman yrityksen MM2436-moduuli.

Koska sylintereitä tullaan testaamaan mahdollisimman hankalissa olosuhteissa, pitää asema-anturille pystyä toimimaan näissä tilanteissa. Lisäksi testattavaa sylinteriä vaihdettaessa asema-anturin vaihtaminen uuteen sylinteriin piti olla vaivatonta. Anturin tyypiksi valittiin lopulta vaijerivetolaite, joka kytketään testattavana olevan sylinterin, sekä jarrusylinterin liitoskohtaan. Laitteeksi valittiin FSG Fernsteuergeräte:n toimittama SL3001–MR1023, koska sitä pystytään käyttämään alhaisissa lämpötiloissa.

3.2 Suunnittelu

Suunnittelussa pystyttiin aloittamaan toteutussuunnittelusta, asiakkaan tehtyä tätä ennen vaatimusmäärittelyn, sekä määritelmät hydraulikoneikon osalta. IO-lista tehtiin suoraan asiakkaan esityksen pohjalta. Hydraulikoneikosta ei ollut olemassa kaaviota, mutta lähtötietojen pohjalta oli mahdollista tehdä kaavio, josta koneikon toimintaperiaate, sekä ohjattavat toimilaitteet selviävät.

Asiakkaan valitsemista antureista osa vaatii käyttöjännitteeksi 12 volttia, toisten käyttäessä 24 volttia. Vaihtoehtona oli joko alentaa 24 voltin jännite 12 volttiin esimerkiksi jänniteregulaattorilla tai käyttää erillistä jännitelähdettä. Ratkaisuna päädyttiin käyttämään toista jännitelähdettä.

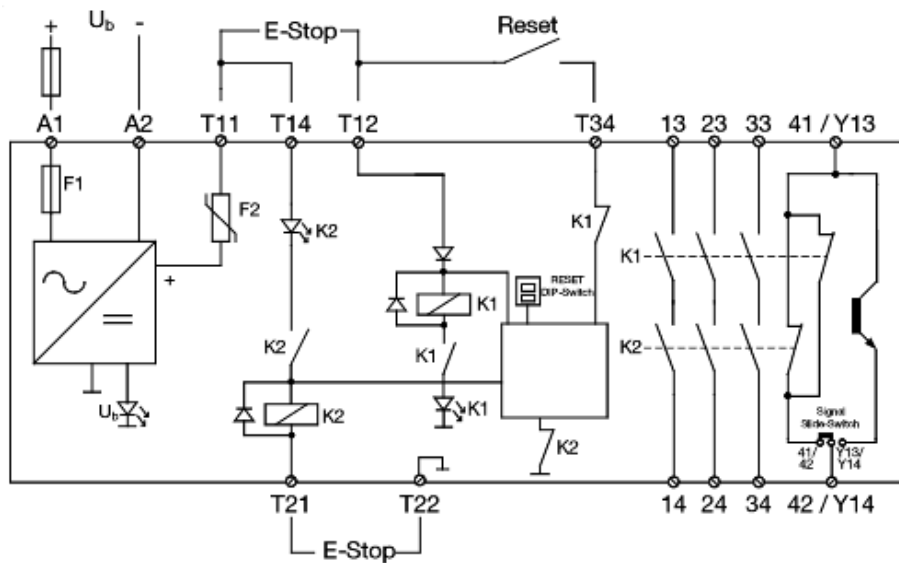
Ohjelmoitavan logiikan lähdöt ohjaavat releitä, jotka kytkevät toimilaitteiden käyttöjännitteen päälle tai pois. Ohjelmoitavan logiikan lähdöt pystyvät ohjaamaan maksimissaan 2 A:n resistiivistä kuormaa, tai 1,5 A:n induktiivista kuormaa. Järjestelmä ohjaa pääasiassa solenoidilla varustettuja venttiileitä, jotka voivat tietyissä tilanteissa aiheuttaa transientin, joka voi vahingoittaa I/O-moduulia. Transientilla tarkoitetaan äkillistä virran nousua, joka johtaa jännitteen ja resistanssin pienenemiseen. Käytettäessä releitä lähtöjen ohjauksessa logiikan ohjaussignaalin virta on huomattavasti pienempi, jolloin moduuleiden käyttöikä saadaan pidennettyä. Jos venttiilin soleno-

idi antaa transientin, vioittuu pelkästään ohjausrele, jonka vaihtaminen on huomattavasti helpompaa ja halvempaa kuin koko I/O-moduulin vaihtaminen. Erillisen releen sijasta on myös mahdollista kytkeä diodi lähdön yhteyteen. Ohjausjärjestelmässä käytettäviä venttiileitä saatetaan kuitenkin vaihtaa toisiin malleihin, jolloin digitaalisen moduulin kapasiteetti ei välttämättä enää riitä. Ulkoisia releitä käytettäessä PLC:n lähdön tarvitsema virta pysyy samana.

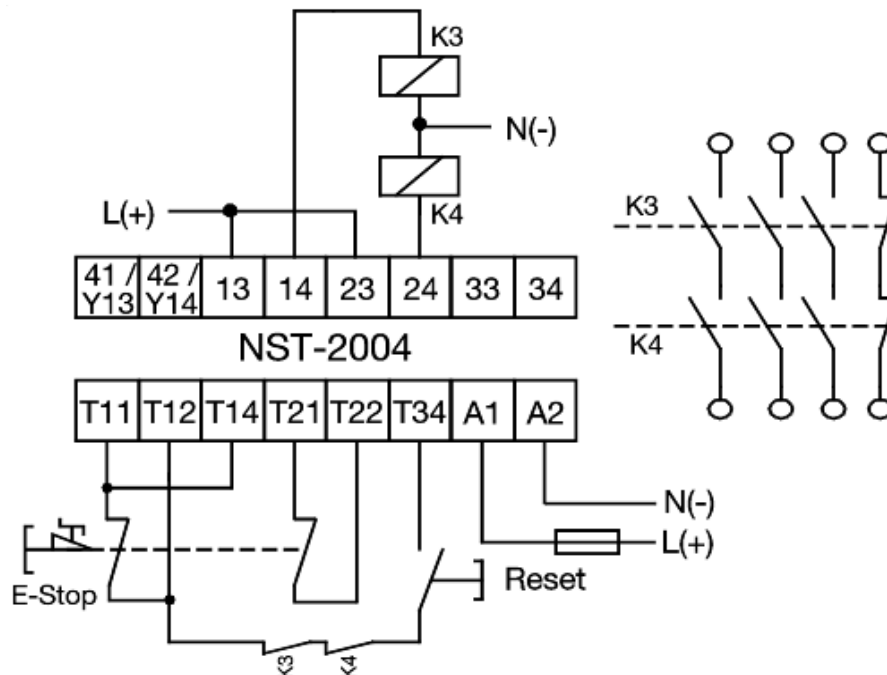
3.3 Turvarele

Ohjelmoitavat logiikat eivät saa olla vastuussa järjestelmän turvallisuudesta. Riittävän koneturvallisuuden takaamiseksi järjestelmässä tulee olla luotettava keino pysäyttää koneikko hätätilanteissa.

Järjestelmässä käytetään Duelcon valmistamaa NST 2004-mallin turvarelettä. Turvarele on kytketty sarjaan lähtöjen ohjausreleen K10 sekä apukontaktorien 1K1 ja 1K2 kanssa. Hätäseis-painiketta painettaessa releen lähtö aukeaa, jolloin kaikki järjestelmän lähtöjen ohjausreleet menettävät toimintajännitteensä. Hätäseis-painike on kytketty suoraan turvareleeseen, eikä se ole suorassa yhteydessä ohjelmoitavaan logiikkaan. Ohjelmoitavassa logiikassa on tulo, johon turvarele on kytketty, jota käytetään vain indikoimaan järjestelmässä turvareleen laukeamisesta.



KUVIO 9. Turvareleen kytkentäkuvaa (www.duelco.com/pdf/NST-2004.pdf).



KUVIO 10. Turvareleen esimerkkikytkentä (www.duelco.com/pdf/NST-2004.pdf).

Kuviossa 11 on esitetty turvareleen sisäiset kytkennät. Koska testipenkin turvareleesä ei ole riittävää määrää koskettimia (3 kpl), on releeseen kytketty lisäksi kaksi apukontactoria K3 ja K4 kuvion 10 mukaisesti, jotta turvareleen kautta ohjattavia koskettimia saadaan riittävästi.

Apureleiden normaalisti suljetut (NC) koskettimet kytketään sarjaan turvareleen kuitaukseen käytettävän kytkimen kanssa, jolloin apukontaktorien toiminta pystytään todentamaan (Duelco, 2011).

3.4 Ohjelmointi

Vaatimusmääritysten perusteella ohjelma jaettiin osakokonaisuuksiin toimintojen perusteella, sekä miten usein nämä toiminnot pitää suorittaa, jotta järjestelmä toteuttaa vaaditut toiminnot. Ohjelmoitavien logiikoiden prosessoreiden suorituskyky verrattuna tietokoneiden prosessoreiden suorituskykyihin on kuitenkin melko rajallinen, joten käytettävät resurssit pitää käyttää mahdollisimman hyvin. Määritysten perusteella tehty lopullinen ohjelma ei ole kovin suuri, kun otetaan huomioon järjestelmän tulot ja lähdöt. Erityisesti mittaustietojen keruu vaatii kuitenkin suhteessa ohjelman muihin osioihin paljon laskentatehoa prosessorilta.

Tästä johtuen ohjelman kokonaisuudet tärkeysjärjestyksessä ovat seuraavat:

1. Järjestelmän turvallisuus
2. Mittaustietojen keruu
3. Tietojen lähetys
4. Tilavuusvirran tarkastelu
5. Sylinterien ohjaus
6. Työvaiheen määrittäminen
7. Muut toiminnot

Osakokonaisuuksille on määritetty eri prioriteetit, jonka perusteella ohjelma suoritetaan. Priorisoinnin perusteella ohjelmoitava logiikka suorittaa ensin tärkeimmän ohjelman ja siirtyy sen jälkeen matalamman prioriteetin toimintoon. Jos prosessorin kuormitus on lähellä maksimia, suoritetaan korkeamman prioriteetin osiot ensin.

Testauksen toimintaperiaate

Kun operaattori aloittaa testauksen, pitää aluksi tehdä joko uusi sekvenssi, tai hakea aikaisemmin luotu sekvenssi ohjelmoitavan logiikan muistista. Testauksessa käytettävä sekvenssi tallennetaan automaattisesti erilliseen taulukkoon, josta ohjelma hakee käytettävä arvot testauksen edetessä. Sekvenssi on jaettu sataan vaiheeseen, joille jokaiselle voidaan määrittää suoritusaika, sekä sylinterien ohjauksessa käytettävien venttiileiden tilat. Sylinterien toiminta on toisistaan riippumattomia, ts. halutesa voidaan ohjata vain toista sylinteriä. Lisäksi voidaan ohjata kuorman vapauttavaa venttiiliä. Ohjelma havaitsee, miten monta eri vaihetta sekvenssiin on määritelty, joten operaattori pystyy syöttämään myös pienemmän määrän työvaiheita. Sekvenssin kooksi määritettiin sata vaihetta, joka mahdollistaa suuren määrän eri ohjaustapoja halutuilla arvoilla, muttei kuitenkaan vie liikaa muistia ohjelmoitavasta logiikasta.

Edellä mainittujen toimintojen lisäksi sekvenssiin määritellään toimintoja, jotka pysyvät muuttumattomina koko testauksen ajan. Hydraulikoneikkoon on asennettu paineakku, jolla voidaan vaikuttaa järjestelmässä esiintyviin paineiskuihin. Paineakku on kytketty järjestelmään ohjausventtiilin avulla, jonka toiminta riippuu edellä mainittujen sylinterien ohjausventtiileiden toiminnasta. Järjestelmään pystytään halutessa ruiskuttamaan epäpuhtauksia, jotta sylinterin toimintaa likaisessa ympäristössä voidaan tarkastella. Tämä toiminto suoritetaan erillään sylinterien ohjauksesta, käyttäjä syöttää päällä olo ajan sekä ajan minkä ruiskutus on pois päältä.

Operaattori syöttää ohjelmaan testin halutun kestoajan, sekä sekvenssin toistomäärän. Kun toinen näistä tulee täyteen, testi lopetetaan ja koneikko pysähtyy. Järjestelmä ilmoittaa merkkivalolla operaattorille testin päättyneen.

Ohjelman suoritusajat

Ohjausjärjestelmän vaatimuksissa oli mittaustietojen keruu vähintään 1 millisekunnin välein. Koko ohjelman suorittaminen millisekunnin sykleissä olisi kuitenkin kuormittanut ohjelmoitavaa logiikkaa liian paljon, joten ohjelma jaettiin aliohjelmiin, jotka suoritetaan eripituisissa sykleissä. Mittaustietojen keruu suoritetaan millisekunnin

syklissä, muut ohjelman osat 10 – 50 millisekunnin sykleissä. Tämän avulla prosessorin kuormitus pysyy 40 - 70 %:ssa testauksen aikana.

Sylinterien ohjaukset

Testattavan sylinterin ohjauksessa voidaan käyttää kolmea eri toimintatapaa. Halutessaan operaattori voi määrittää sylinterin ohjauksen tapahtuvan joko ajan perusteella, ääriasemien saavuttamisen perusteella tai näiden yhdistelmällä.

Ajan perusteella tapahtuvassa testauksessa sylinteriä ohjataan asetettujen aikojen mukaan ääriasemiin. Kun tietylle vaiheelle määritetty aika on kulunut loppuun, hakee ohjelma seuraavat arvot PLC:n muistista.

Operaattori voi halutessaan suorittaa joitakin työvaiheita asematiedon perusteella. Kun sylinteri saavuttaa toisen ääriaseman, siirrytään sekvenssissä välittömästi seuraavaan työvaiheeseen.

Edellä mainitut menetelmät voidaan myös yhdistää. Kun sylinteri on ohjattu haluttuun suuntaan ääriasentoon, ohjataan sylinteriä tähän suuntaan halutun ajan.

Tilavuusvirran tarkastus

Mahdollisten hydraulijärjestelmässä esiintyvien vuotojen takia ohjausjärjestelmä voidaan käskä seuraamaan halutuin väliajoin järjestelmän tilavuusvirtaa. Sylinterin mäntä ohjataan toiseen ääripäähän, jonka jälkeen lasketaan virtausmittauksen antamien arvojen keskiarvoa tietyn ajanjakson verran. Jos tarkastuksen aikana tilavuusvirran suuruus ylittää asetellun rajan, järjestelmä antaa hälytyksen.

Testauksessa käytössä oleva sekvenssi keskeytetään tarkastuksen ajaksi. Kun tarkastus on valmis, palataan siihen vaiheeseen, jossa oltiin tarkastuksen alkaessa. Koska sylinteriä ohjataan tarkastelussa toiseen ääripäähänsä, käytetään tarkastuksessa testattavan sylinterin ohjausventtiiliä. Muiden ohjausten, kuten kuormasyylinterin on oltava toimimattomana tarkastelun aikana.

Järjestelmän raja-arvot ja lukitukset

Järjestelmän on tarkoitus tallentaa mittaustietoja tilanteissa, joissa järjestelmän suu-reet lähestyvät kriittisiä arvoja. Itse järjestelmä ei kuitenkaan missään tilanteessa saa olla vaarallinen ympäristölleen. Järjestelmässä on erikseen lukitusarvot, joita operaattori ei pysty muuttamaan. Lukitusarvojen ylittyessä järjestelmä pysäyttää hydraulikoneikon sekä kytkee sylinterin testaukseen liittyvien lähtöjen jännitteen pois. Lukitus voidaan poistaa kuittaamalla se pois valvomosta, kun prosessin arvot ovat palanneet turvalliselle alueelle.

Lukitusten lisäksi järjestelmään on asetettu raja-arvot, joiden ylittyessä järjestelmä suorittaa mittaustietojen siirron dataloggerissa tallennuspuskurista .csv-tiedostoon. Operaattori pystyy muuttamaan näitä arvoja halutusti. Kun arvot ylittyvät, järjestelmä muuttaa taulukossa 1 olevaa *Tallennuspituus/hälytys*-sanaa. Datalogger suorittaa mittaustietojen siirron tämän sanan arvon muutosten perusteella.

Sekvenssin rakenne

Ennen ohjelmoinnin aloittamista oli määritettävä, miten sekvenssin rakenne tulisi toteuttaa. Teoriassa ohjelma muistuttaa sekvenssiohjelmointia, jossa ohjelma on jaettu työvaiheisiin, joita suoritetaan yksi kerrallaan, jonka jälkeen siirrytään seuraavaan. Testipenkillä suoritettavat testit saattavat kuitenkin olla hyvinkin erilaisia toisiinsa nähden, joten helpommaksi vaihtoehdoksi tuli syöttää ohjaustiedot taulukkoon, josta ohjelma hakee kullekin työvaiheelle määritellyt lähtöjen tilat.

Testauksessa käytettävät arvot sijoitetaan ohjelmoitavan logiikan muistissa olevaan taulukkoon. CoDeSys-ohjelmassa voidaan luoda omia datarakenteita, jonka avulla samaan taulukkoon voidaan tallentaa eri datatyyppejä.

Testauksen toiminnot voidaan jakaa kahteen ryhmään. Ensimmäiseen ryhmään kuuluu ne osat, joiden arvoja ei tarvitse muuttaa testauksen aikana, vaan niitä ohjataan alussa määriteltyjen arvojen mukaan. Näitä toimintoja ovat tilavuusvirran tarkastus, epäpuhtauden ruiskutus järjestelmään sekä paineakun venttiilin ohjaus. Toiseen ryhmään kuuluvat testattavan sylinterin ohjaus, kuormasynterinin ohjaus sekä kuorman vapautus.

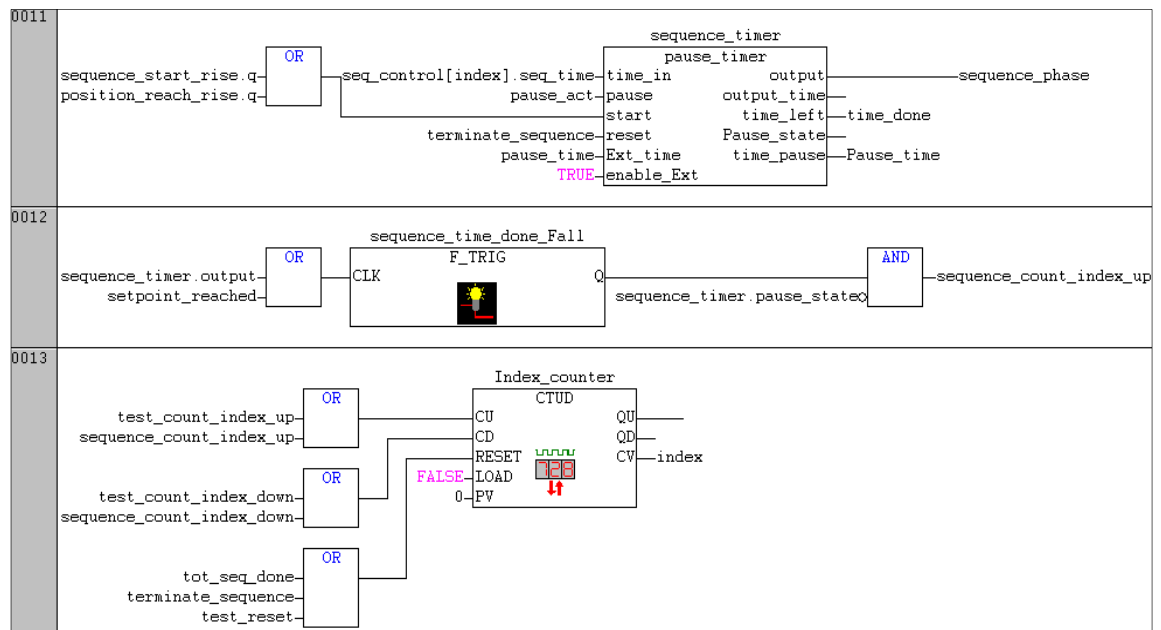
Ensimmäisen ryhmän toiminnot tallennetaan omaan taulukkoonsa ja toisen ryhmän toiminnot omaansa. Ensimmäisen ryhmän taulukon kooksi määritettiin yksi solu, koska näille toiminnoille ei tarvitse hakea uusia arvoja testauksen alettua.

Sylinterien ohjaus sekä kuorman vapautus taas toimivat ajastuksen perusteella, joten niille on tehtävä oma taulukko, johon määritetään sylinterien ohjaukset vaihe kerrallaan. Taulukon yksi solu tarkoittaa yhtä työvaihetta sekvenssissä. Kuviossa 12 on esitetty esimerkki yhden työvaiheen ohjauksille.

0058	Seq_Control
0059	Seq_Control[0]
0060	.seq_time = T#500ms
0061	.MV1_out = TRUE
0062	.MV1_in = FALSE
0063	.LoadValve_out = TRUE
0064	.LoadValve_in = FALSE
0065	.Release_load = FALSE
0066	.PWM1 = 50
0067	.time_mode = FALSE
0068	.position_mode = FALSE
0069	.Piston_setpoint = 0
0070	Seq_Control[1]
0071	Seq_Control[2]
0072	Seq_Control[3]
0073	Seq_Control[4]
0074	Seq_Control[5]
0075	Seq_Control[6]
0076	Seq_Control[7]
0077	Seq_Control[8]
0078	Seq_Control[9]
0079	Seq_Control[10]
0080	Seq_Control[11]
0081	Seq_Control[12]
0082	Seq_Control[13]
0083	Seq_Control[14]
0084	Seq_Control[15]
0085	Seq_Control[16]
0086	Seq_Control[17]

KUVIO 11. Sekvenssin ohjaustiedot sisältävä taulukko

Testattavan sylinterin (MV1_out, MV1_in) ja kuormasynterinin (LoadValve_out, LoadValve_in) ohjauksen lisäksi solu sisältää tiedon kuorman vapautukselle (Release_Load), proportionaaliventtiin ohjaustiedon (PWM1), valitun ohjausmuodon (time_mode/position_mode) sekä halutun asematiedon, jos sylinteriä ohjataan aseman mukaan (piston_setpoint). Sylinterien ohjauksessa käytettävät aliohjelmat haakevat arvot kullekin ajan hetkelle taulukon indeksin avulla. Indeksien arvo määritellään omassa aliohjelmassa, joka kasvattaa indeksiä ajastimen perusteella.



KUVIO 12. Työvaiheen määrittäminen

Kuviossa 13 esitetään toimintaperiaate, jolla työvaiheen numero määritellään. Rivi 11 sisältää ajastimen, jonka lähtö on ajastimen käydessä päällä. Käyntiajan ajastin saa taulukon osoitteesta `seq_control[index].seq_time`, jossa `index` osoittaa taulukon solu numeroa ja `seq_time` muuttujaa solussa.

Kun työvaiheen aika on kulunut loppuun, ajastimen lähtö menee pois päältä. Ajastimen lähdön signaalia seurataan `F_trig`-komennolla, joka havaitsee signaalin tilan muuttumisen aktiivisesta passiiviseksi. Toimintoa kutsutaan negatiivisen reunan havaitsemiseksi, jonka Q-lähtö on päällä yhden ohjelmakierron ajan, kun seurattavassa signaalissa havaitaan reuna. Vastaava toiminto on positiivisen reunan havaitseminen, joka seuraa signaalin muutosta päälle.

Kun ajastimen signaalissa havaitaan laskeva reuna, `Index_counter`-lohko kasvattaa indeksiä eli työvaiheen numeroa yhdellä. Aliohjelmat, jotka tarvitsevat työvaiheen numeroa ohjauksen tilojen määrittelyyn hakevat tällöin uudet arvot seuraavan ohjelmakierron alussa. Samalla ajastin alustetaan uudella ajalla.

CoDeSys-ympäristö tarjoaa ohjelmoijalle neljä erilaista ajastinta

- Timer On (TON), jossa ajastin käynnistyy, kun tulosignaali havaitaan nouseva reuna, ajastimen lähtö laitetaan päälle, kun aika on kulunut ja pysyy päällä tulosignaalin ollessa päällä.

- Timer Off (TOF), jossa ajastin käynnistyy, kun tulosignaalisissa havaitaan laskeva reuna. Ajastimen lähtö on päällä asetetun ajan verran.
- Timer Pulse (TP), jossa ajastin käynnistyy, kun tulosignaalisissa havaitaan nouseva reuna. Ajastimen lähtö on päällä asetetun ajan verran sekä
- Real Time Clock (RTC), joka toimii reaaliaikaisena kellona.

Tiettyissä tilanteissa ajastin pitää pystyä pysäyttämään, esimerkiksi kun tilavuusvirta tarkastetaan testauksen aikana. Kun testaus loppuu, tulee testauksen jatkuva aiemasta tilasta. Ajastimena käytetään Timer pulse-tyyppin ajastinta. Ajastimen lisäksi tarvitaan apumuuttuja, jonka tilaa muuttelemalla päätellään, halutaanko ajastin keskeyttää. Keskeytyksen alkaessa tallennetaan aika, joka ajastimella oli jäljellä tällä hetkellä. Keskeytyksen loppuessa ajastimen ajaksi asetetaan keskeytyshetken aika.

3.5 Proportionaaliventtiilin ohjaus

Järjestelmässä on proportionaaliventtiili, joka tarvitsee 2A:n virran toimiakseen. Venttiilillä ohjataan testattavan sylinterin nopeutta. Venttiili toimii pulssinleveysmodulaatiolla, mutta ABB:llä ei ole näitä vaatimuksia täyttäviä moduuleita, joten PWM-moduuliksi valittiin Bernecker & Rainer -nimisen (B&R) yrityksen MM2436-moduuli. Moduuli kytketään ohjelmoitavaan logiikkaan Profibus DP-väylän kautta.

Moduuli sisältää kaksi suurille taajuuksille tarkoitettua laskuria, joista toista tullaan myöhemmin käyttämään kierrosnopeuksien laskemiseen pulssianturin avulla. MM2436-moduulin lähtöjä voidaan käyttää joko virta- tai PWM-signaalien lähetyksessä. Konfiguroinnin yhteydessä valitaan, kumpaa tapaa halutaan käyttää. Moduulilla pystytään ohjaamaan yhteensä kahta eri toimilaitetta, joille on varattu yksi lähtö kummallekin. Kahden edellä mainitun lisäksi moduuli sisältää kaksi lähdön, joista ensimmäiseen syötetään ohjaussignaalin jaksonkesto. Viimeinen lähtö on varattu mahdollisten häiriöiden kuittaukseen (Bernecker & Rainer, 2011).

PWM-lähtöön voidaan syöttää arvoja väliltä -32768 – 32767. Tätä käytetään moottorien ohjauksessa, jos moottoria halutaan pystyä ohjaamaan sekä myötä- että vasta-

päivään. Venttiilien ohjauksessa käytetään kuitenkin vain arvoja väliltä 0 – 32767. Jotta käyttö olisi mahdollisimman selkeää, operaattori syöttää venttiilille arvon väliltä 0 – 100, joka skaalataan oikeaksi signaaliksi logiikassa.

3.6 Käyttöliittymä

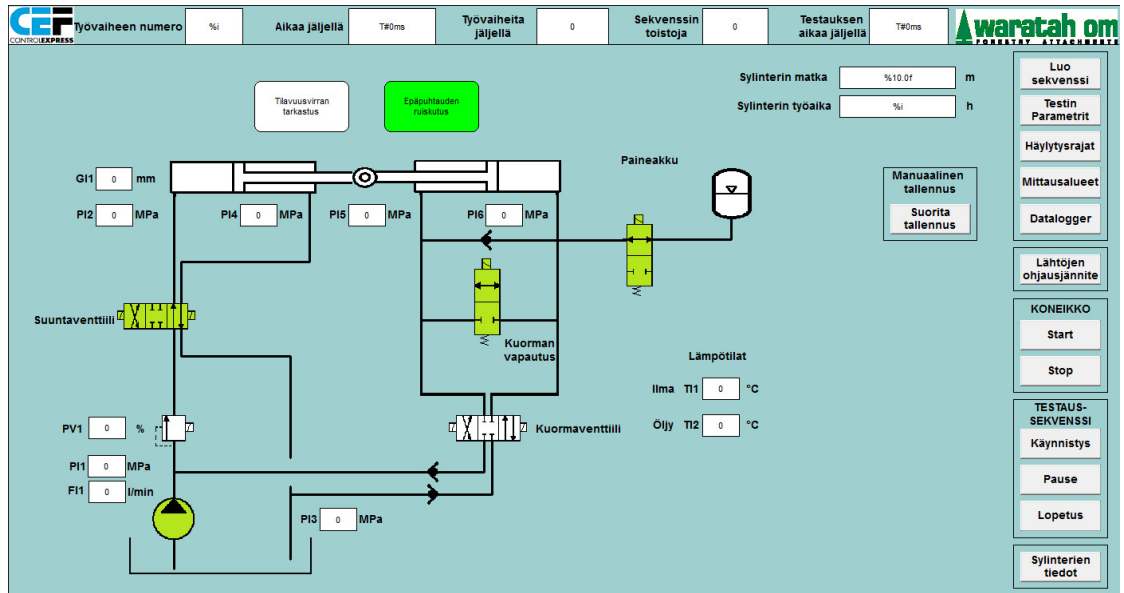
Käyttöliittymällä, tai valvomolla, tarkoitetaan sovellusta tai laitetta, jonka avulla käyttäjä pystyy antamaan komentoja ohjelmitavalle logiikalla ja saa tietoja prosessin tilasta (Kippo & Tikka, 2008, 46). Käyttöliittymän suunnittelu on yksi keskeisistä osista automaatiojärjestelmien toteutuksessa. Nykyisin käyttöliittymät sijoitetaan usein valvomotietokoneelle, jolloin operaattori syöttää arvoja näppäimistön ja hiiren avulla. Tämän lisäksi käyttöliittymä voidaan sijoittaa erilliseen operointipaneeliin.

Käyttöliittymä toteutettiin Visualization-työkalulla, joka sisältyy CoDeSys v2.3-ohjelmistopakettiin. Valvomo-ohjelmisto on asennettu valvomo-PC:lle, johon myös mittausdatan tallennus tapahtuu. Käyttöliittymän kautta syötetään arvot testauksessa käytettävään sekvenssiin ja sen kautta pystytään seuraamaan testauksen tilaa. Operaattorille tuli antaa mahdollisimman suuret vapaudet toteuttaa halutunlainen testaussekvenssi.

Toteutettaessa käyttöliittymää Visualization-työkalulla, tapahtuu valmiin käyttöliittymän kommunikointi ohjelmitavan logiikan kanssa samalla tavalla, kuin ohjelmaa luodessa. Tämän johdosta erillistä OPC-serveriä, tai vastaavaa kommunikointiin tarvittavaa sovellusta ei tarvita (3s-Software, 2012).

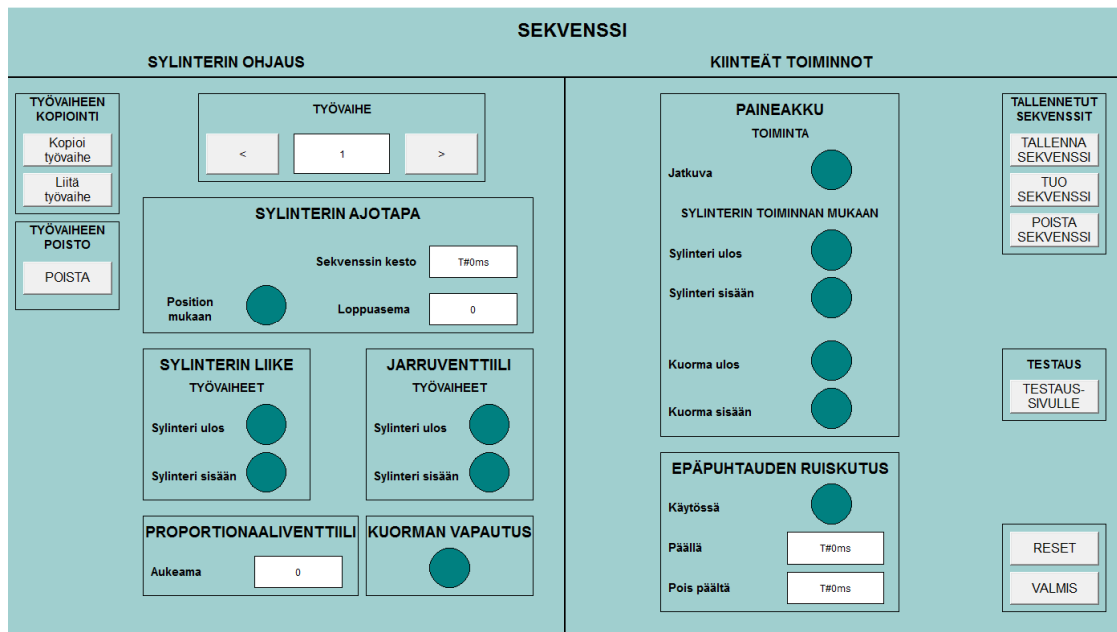
Käyttöliittymän suunnittelussa on huomioitava, että operaattorilla on jatkuvasti tarvittava tieto järjestelmän tilasta. Operaattorin on lisäksi pystyttävä ohjaamaan prosessia saamiensa tietojen perusteella mahdollisimman yksinkertaisesti ja vaivatta. Liika informaation esittäminen saattaa kuitenkin johtaa tietotulvaan, jolloin tärkeää tietoa saattaa jäädä huomiotta.

Projektin käyttöliittymä voidaan jakaa kahteen eri kokonaisuuteen. Pääikkunaan, jossa tapahtuu pääasiassa testauksen hallinta ja seuraaminen. Toisen osan muodostuu sekvenssin luontiin tarkoitetusta ikkunasta sekä siihen liittyvistä Pop up-ikkunoista.



KUVIO 13. Pääikkuna

Pääikkunan keskellä on hydraulikoneikkoa esittävä kaavio, josta operaattori näkee venttiilien tilat sekä antureiden mittaustiedot. Ikkunan oikeassa reunassa on testauksessa käytettävän sekvenssin rakentamiseen liittyvien ikkunoiden aukaisu, sekä koneikon ja testauksen hallintapainikkeet. Tarkoituksena on, että testausta aloitettaessa operaattori siirtyy järjestyksessä ylhäältä alas, jolloin kaikki toimenpiteet ennen testausta käydään läpi.



KUVIO 14. Sekvenssin luomisikkuna

Haluttaessa käytössä oleva sekvenssi voidaan tallentaa reseptiksi myöhempää käyttöä varten. Resepti tallennetaan ohjelmoitavan logiikan muistiin, josta se voidaan noutaa myöhemmin. Haettaessa reseptiä se tallennetaan sillä hetkellä työmuistissa olevan sekvenssin päälle.

Valvomon, sekä itse ohjelman suunnittelussa on huomioitava, ettei operaattori pysty toimillaan saattamaan prosessia tilaan jossa henkilö- tai materiaalivahinkoja voi syntyä. Kun operaattori käynnistää hydraulikoneikossa olevan pumpun, lukitaan suurin osa käyttöliittymän toiminnoista. Tässä vaiheessa operaattori pystyy vain aloittamaan, keskeyttämään ja lopettamaan testauksen, sekä pysäyttämään koneikon.

Vaatimusmääritysten mukaan järjestelmän on kyettävä toimimaan ilman operaattorin aktiivista valvontaa pitkiä aikoja. Käyttöliittymän kautta operaattorin tulee kuitenkin pystyä seuraamaan testiä tarvittaessa. Muutoksia testauksessa käytettäviin arvoihin ei kuitenkaan voi tehdä testauksen ollessa käynnissä, jotta mahdolliset virhepainallukset tai kirjoitusvirheet eivät muuta testausta. Ennen kuin testaus aloitetaan, operaattorin on mahdollista tarkastaa sekvenssi vaihe kerrallaan siirtymällä manuaalisesti edestakaisin sekvenssissä.

3.7 Mittaustietojen keruu ja tiedonsiirto

Vaativuusmäärityksen mukaisesti mittaustietojen tallennus tapahtuu vähintään millisekunnin välein. Analogiamittauksista tulleet arvot muutetaan logiikassa SI-järjestelmän suureiksi. Muunnetut arvot tallennetaan väliaikaisesti kaksiulotteiseen taulukkoon, josta ne lähetetään edelleen valvomo-PC:lle tallennettavaksi UDP:n (User Datagram Protocol) avulla. CoDeSys-ympäristöstä löytyy tähän tarkoitukseen valmis funktio. Yhteyden luomiseen UDP-protokollan avulla ohjelmitava logiikka tarvitsee kohdelaitteen IP-osoitteen, portin, jonka kautta liikennöinti tapahtuu, sekä tässä tapauksessa kommunikointiin käytettävän moduulin sijainnin hardware-konfiguraatiossa.

Tietojen lähetykselle haluttiin mahdollisimman pitkä aikaväli, jotta keskusyksikön kuormitus pysyy mahdollisimman kevyenä, mutta kaikki haluttu data ehditään lähettämään, ts. mittaustietoja sisältäviä paketteja ei häviäisi. Oman rajoituksensa asettaa ohjelmointiympäristön tarjoama funktio, joka asettaa lähetettävän paketin maksimikooksi 1464 sanaa (1 sana = 2 tavua). Lähetettävään pakettiin kuuluu tietty määrä seuraavan taulukon mukaisia tietoja.

TAULUKKO 3. Lähetettävät tiedot

	Määrä	koko (tavuina)	yht. (tavuja)
Painemittaus	6	2	12
Lämpötilamittaus	2	2	4
Virtausmittaus	1	2	2
Asematieto	1	2	2
Nopeus	1	2	2
Kokonaismatka	1	2	2
Kierrosnopeus	1	2	2
Taulukon solu/aikaleima	1	2	2
Tallennuspituus/hälytys	1	2	2
Tallennustaajuus	1	2	2
Yhteensä	16		32

Tiedonsiirtoa testaamalla sopivaksi tiedonlähetyksen sopivaksi kiertoajaksi määriteltiin 10 millisekuntia, jolloin lähetettävä paketti on riittävän pieni, mutta sekä ohjelmoitavalla logiikalla, sekä valvomo-PC:llä on riittävästi resursseja tiedon käsittelyyn. Lähetettävän paketin koko määräytyy siitä, montako kertaa mittaustietoja ehditään kirjoittamaan, kunnes tiedonlähetyksesi suoritetaan seuraavan kerran. Näin ollen kerralla lähetettävän paketin kooksi määriteltiin maksimissaan 320 tavua. Paketin kokoa voidaan vielä teoriassa kasvattaa, hidastamalla tiedonlähetyksen suoritusnopeutta, mutta tiedonsiirtoon haluttiin jättää ylimääräistä kapasiteettia, jotta mahdollisissa poikkeustilanteissa, kuten PLC:n ollessa täysin kuormitettuna ilmaantuvat piikit eivät kasvata lähetettävän paketin kokoa liian suureksi, joka johtaisi häiriöön lähetyksessä. Lisäksi mahdollisten laajennusten takia järjestelmään tulee jättää ylimääräistä kapasiteettia.

Operaattori pystyy testauksen alussa määrittelemään halutun taajuuden yhden millisekunnin – sekunnin väliseksi, jolla mittaustiedot tallennetaan järjestelmään. Käytännössä antureiden signaalit luetaan jatkuvasti millisekunnin välein, mutta tiedot tallennetaan järjestelmään asetetulla taajuudella.

```

IF (i MOD savefreq) = 0 THEN
    IF savefreq = 1000 THEN
        Meas_Send[a].Time_Stamp := 0;
    ELSE
        meas_send[a].time_stamp := i;
    END_IF

    meas_send[a].speed := GI1Speed.CylSpeed;
    Meas_Send[a].GI1 := GI1_Phys;
    Meas_Send[a].FI1 := FI1_Phys;
    Meas_Send[a].TI1 := TI1_Phys;
    Meas_Send[a].TI2 := TI2_Phys;
    Meas_Send[a].PI1 := PI1_Phys;
    Meas_Send[a].PI2 := PI2_Phys;
    Meas_Send[a].PI3 := PI3_Phys;
    Meas_Send[a].PI4 := PI4_Phys;
    Meas_Send[a].PI5 := PI5_Phys;
    Meas_Send[a].PI6 := PI6_Phys;
    Meas_send[a].Mileage := 0;
    Meas_send[a].RPM := REAL_TO_INT(100 * RPM_Calc.RPM_speed);
    Meas_send[a].Error := Limit_comparison.Error_status;
    Meas_send[a].Freq := saveFreq;

    IF a >= 1000 THEN
        a := 0;
    END_IF

    a := a + 1;
    UDP_write_Index := a;
END_IF;

IF i >= 1000 THEN
    i := 0;
END_IF

i := i + 1;

```

KUVIO 15. Tiedontallennuksen toiminta

Kuvio 16 esittää, miten tietojen tallennus taulukkoon suoritetaan halutulla taajuudella. Koska ko. ohjelma suoritetaan millisekunnin välein, kasvaa muuttuja *i* jokaisella suorituskerralla yhdellä. Kun tätä muuttujaa verrataan operaattorin asettamaan tallennustaajuuteen, saadaan MOD-komennolla määriteltyä ne tilanteet, jolloin näiden jakolaskun jakojäännös on nolla.

```

UDP_S_SEND.PORT := 8000;
IF Init = TRUE THEN
  Send_Index := UDP_write_index;
  totalLength := 0;
  Send_Counter := 0;
  Enable := FALSE;
  Init := FALSE;
  run := TRUE;
END_IF
IF UDP_S_SEND.EN = FALSE AND run = TRUE THEN
  IF UDP_write_Index <> Send_Index THEN
    IF UDP_write_Index > Send_Index THEN
      Length := UDP_write_Index - Send_Index;
    ELSE
      Length := 1001 - Send_Index;
    END_IF
    IF Length > 45 THEN
      Length := 45;
    END_IF
    Enable := TRUE;
  END_IF
END_IF
UDP_SEND(EN:=Enable,SLOT:= 0,INDEX:=0,IP_ADR:=16#0A0B0DF0,
  DATA:=ADR(Meas_send[Send_Index]),LEN:=(Length*SIZEOF(Meas_send[Send_Index])));
IF UDP_S_SEND.EN = TRUE AND UDP_S_SEND.DONE = TRUE THEN
  Send_Index := Send_Index + length;
  IF Send_Index = 1001 THEN
    Send_Index := 1;
  END_IF
  Enable := FALSE;
END_IF

```

KUVIO 16. Tiedonsiirron toiminta

Kuvio 17 esittää tiedonsiirron ohjelmoitavalta logiikalta valvomo-PC:lle. Liikennöinnin tärkein osa on UDP_SEND-funktio, joka suorittaa tiedon lähetyksen. Koska mittaus-tiedot kerätään taulukkoon, pitää lähetettävän solun osoitteen muuttua, jotta oikeat mittauksiedot lähetetään. Vertailukohtana käytetään taulukon solua, johon on viimeksi kirjoitettu mittauksietoja. Ohjelma vertaa sitä soluun, jossa oltiin edellisellä lähetyksellä ja määrittää paketiiksi kaikki mittauksietot edellisestä lähetyksestä viimeisimpään taulukon soluun.

3.8 Tietojen käsittely

Järjestelmän keräämän datan määrä nousee suureksi, (millisekunnin tallennustajuuudella mittaustietoja kertyy 32 kilotavua sekunnissa), joten hälytysrajojen sisällä olevia arvoja poistetaan dataloggerissa tietyn ajan kuluttua. Muussa tapauksessa mittaustiedoista luotava tiedosto olisi epäkäytännöllisen suuri. Koska mittaustiedot, jotka ovat alle asetettujen hälytysrajojen, ovat tarpeettomia, käytetään FIFO-puskuria, josta vanhoja mittaustietoja poistetaan sitä mukaan kun uusia mittaustietoja lähetetään.

Asiakkaan vaatimusmääritysten mukaisesti tiedostoon on tallennettava sekä raja-arvon ylitystä edeltävät, että välittömästi sen jälkeiset mittaustiedot. FIFO-puskurin avulla pystytään säilyttämään mittaustiedot ajalta ennen hälytystä. Hälytyksen tullessa mittaustiedot tallennetaan .csv-tiedostoon. Samaa tiedostoa käytetään tallentamaan kaikki saman päivän aikana tulleet hälytykset.

3.9 Antureiden skaalaus

Antureilta tulevat viestit tulee muuntaa ohjelmoitavassa logiikassa SI-järjestelmän mukaisiksi arvoiksi. Viesti on 16-bittinen integer-tyyppin lukuarvo. Muuntaminen tapahtuu kaavalla

$$A_{out} = \left(\frac{A_{max} - A_{min}}{S_{max} - S_{min}} \right) * S + \left[(A_{min} - S_{min}) * \left(\frac{A_{max} - A_{min}}{S_{max} - S_{min}} \right) \right],$$

jossa A_{out} = SI-järjestelmän suure

A_{max} = Anturin mittausalueen maksimiarvo

A_{min} = Anturin mittausalueen minimiarvo

S = Anturin mittaviesti

S_{max} = AI-moduulin signaalin minimiarvo = 0 (AC500 online help, 2011)

S_{min} = AI-moduulin signaalin maksimiarvo = 27648 (AC500 online help, 2011).

Tarvittaessa anturin mittausalueita pystytään muuttamaan, esimerkiksi tilanteissa joissa anturi vaihdetaan toiseen.

4 YHTEENVETO

4.1 Tulokset

Järjestelmän käyttöönotto viivästyi vuoden 2012 alusta toukokuulle 2012, eikä sitä ole vielä otettu käyttöön opinnäytetyöraportin kirjoittamisen aikaan. Järjestelmässä käytettävä ohjelmoitava logiikka sekä tulo- ja lähtömoduulit olivat kuitenkin käytettävissä koko projektin ajan, joten projektissa tehtyä ohjelmaa päästiin testaamaan erittäin paljon. Tämän lisäksi tiedontallennusta pystyttiin kokeilemaan ja säätämään sen parametreja hyvin laajasti.

Laajamittaisesta testauksesta huolimatta, on ohjelmassa todennäköisesti tiettyjä osioita, joiden toimintaa täytyy vielä säätää. Erityisesti sylinterin asematietoa hyväkseen käyttävät osiot saattavat vielä vaatia muutoksia tai tarkennuksia ennen kuin järjestelmä toimii täysin suunnitellulla tavalla. Nämä ongelmat ovat kuitenkin suhteellisen helppo löytää sekä rajata ongelma, joten käyttöönottilanteessa niiden ei pitäisi tuottaa ongelmia.

Kokonaisuutena järjestelmä täyttää kaikki projektin alussa sille asetetut vaatimukset. Tulevaisuudessa järjestelmää saatetaan laajentaa lisäämällä testattavien sylintereiden määrää, sekä muita mittauksia. Laajennukset vaativat erityisesti ohjelmoitavalta logiikalta suoritustehoa ja mittapisteiden lisäykset kuormittavat ohjelmoitavan logiikan ja valvomo-pc:n välistä tiedonsiirtoa. Vaikka ohjelmoitavan logiikan suorituskyyky on pyritty optimoimaan, on ohjelmassa todennäköisesti vielä toimintoja, joita tehostamalla laajennukset pystytään toteuttamaan ilman suurempia muutoksia itse ohjausjärjestelmään.

4.2 Pohdinta

Jälkikäteen nähtynä lähdin joillakin osa-alueilla hieman vääristä lähtökohdista, esimerkiksi valvomo syntyi osin ”lennosta”. Suunnitteleamalla etukäteen esimerkiksi luonnoksen valvomosta ja miettimällä mitä valvomosta tulee löytyä, olisi valvomo tullut valmiiksi pienemmällä vaivalla. Nyt siihen tehtiin isojakin muutoksia, jotka olisi ollut helppo välttää tekemällä enemmän pohjatyötä.

Kokonaisuutena projekti oli sopivan kokoinen opinnäytetyöksi. Lisäksi aikaa paneutua eri osa-alueisiin oli riittävästi. Työn aikana pystyinkin kehittämään ongelmanratkaisukykyjäni sekä loogista päättelyä, joista molemmat ovat tärkeitä suunnittelutyössä ja ohjelmoinnissa.

Teoriakursseilla opittuja asioita pystyi hyödyntämään melko paljon ja uuden ohjelmointiympäristön omaksuminen sujui melko kivuttomasti. Koska CoDeSys – ympäristö sisältää kaikki IEC61131-3 –standardin mukaiset ohjelmointikielet, tässä projektissa pystyin kokeilemaan eri ohjelmointikielten ominaisuuksia, sekä niiden vahvuuksia ja heikkouksia eri tilanteissa. Aluksi olin ajatellut toteuttavani ohjelman kokonaan FBD-ohjelmoinnilla, mutta projektin edetessä huomasin erityisesti ST-kielen hyödylliseksi monimutkaisemmissa funktioissa.

Projektissa pääsin tutustumaan ohjelmitaviin logiikoihin liittyviin ominaisuuksiin, joiden kanssa en aikaisemmin ollut työskennellyt. Ymmärrys ohjelmitavien logiikoiden mahdollisuuksista ja rajoituksista lisääntyi huomattavasti.

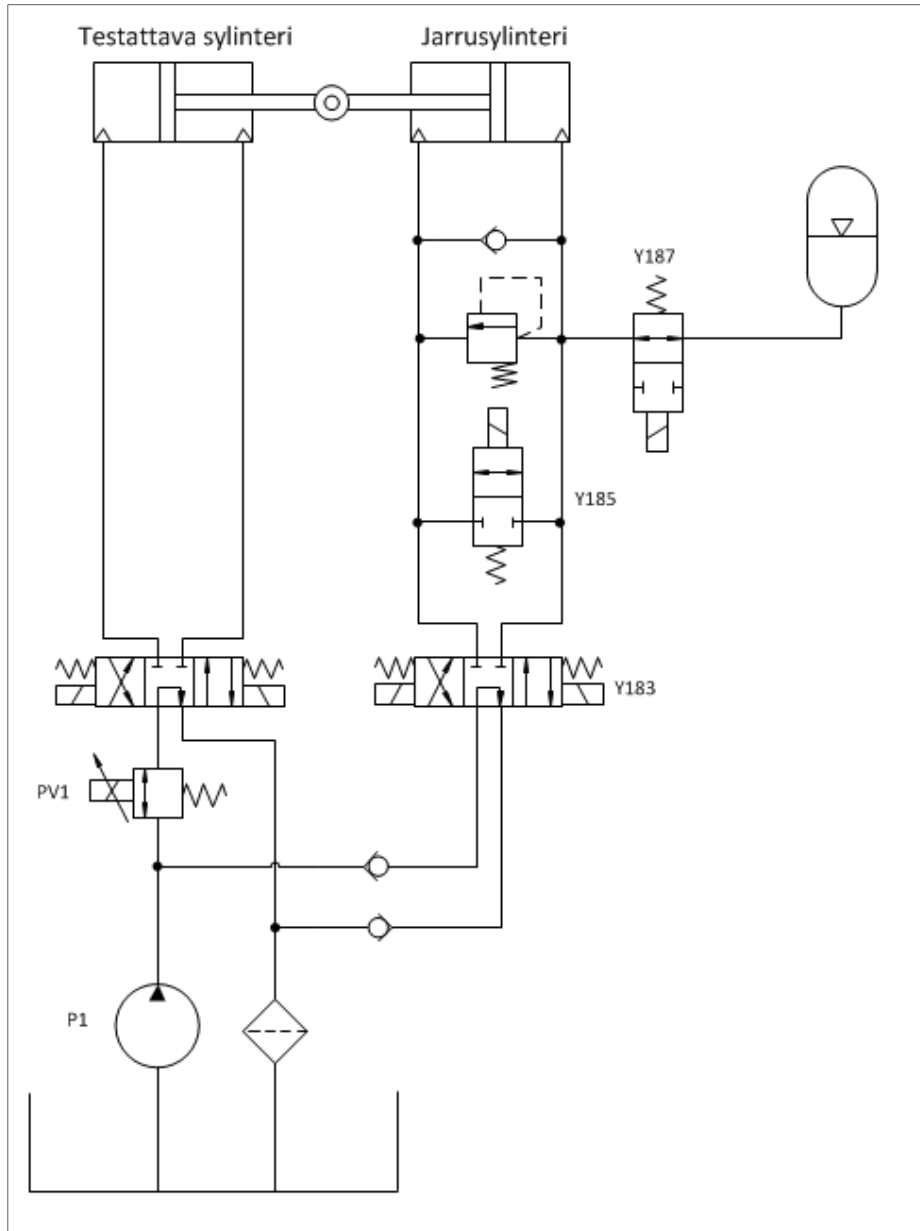
Real Time Automation, 2009. Viitattu 12.4.2012.

<http://www.rtaautomation.com/iec61131-3/>

Sivonen, M, 2001. Teollisuuden instrumentointi, AEL.

LIITTEET

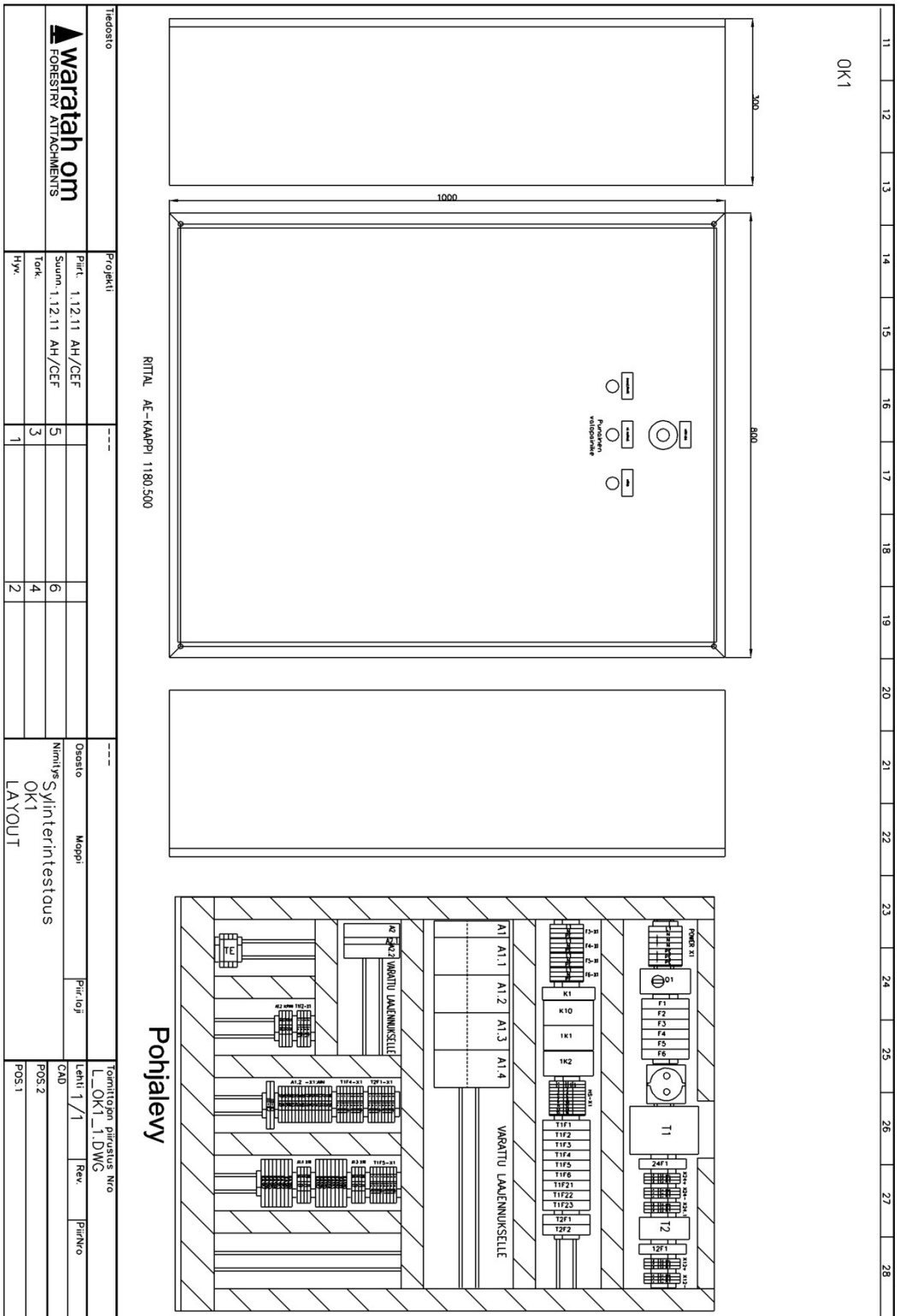
Liite 1. Hydraulikoneikon periaatekaavio



Liite 2. IO-lista

IO-LISTA				
Tekijä			Antti Hietala/CEF	
Päivämäärä			7.12.2011	
Moduuli	Osoite	Tyyppi	Kuvaus	Tunnus
			Tulot	
A1.2	%IW0	AI	Virtausmittaus	FI1
A1.2	%IW1	AI	Lämpötilamittaus, Ilma	TI1
A1.2	%IW2	AI	Lämpötilamittaus, Öljy	TI2
A1.2	%IW3	AI	Sylinterin asematieto	GI1
A1.2	%IW4	AI	Painemittaus 1	PI1
A1.2	%IW5	AI	Painemittaus 2	PI2
A1.2	%IW6	AI	Painemittaus 3	PI3
A1.2	%IW7	AI	Painemittaus 4	PI4
A1.2	%IW8	AI	Painemittaus 5	PI5
A1.2	%IW9	AI	Painemittaus 6	PI6
A1.3	%IX32.0	DI	Hätäseis Ok	Hataseis
A1.3	%IX32.1	DI	Ohjaujännite OK	Ohjaujännite
A1.3	%IX32.2	DI	Hydrauliikka käy	lampörelle
A1.3	%IX32.3	DI	Varalla	
A1.3	%IX32.4	DI	Sylinteri ääri raja sisällä	S 324
A1.3	%IX32.5	DI	Sylinteri ääri raja ulkona	S 325
A1.3	%IX32.6	DI	Varalla	
A1.3	%IX32.7	DI	Varalla	
A1.4	%IX48.0	DI	Varalla	
A1.4	%IX48.1	DI	Varalla	
A1.4	%IX48.2	DI	Varalla	
A1.4	%IX48.3	DI	Varalla	
A1.4	%IX48.4	DI	Varalla	
A1.4	%IX48.5	DI	Varalla	
A1.4	%IX48.6	DI	Varalla	
A1.4	%IX48.7	DI	Varalla	
			Lähdöt	
A1.3	%QX18.0	DO	Hälytys/vikailmoitus	OK1-H3
A1.3	%QX18.1	DO	Sylinterin ohjaus ulos	Y181
A1.3	%QX18.2	DO	Sylinterin ohjaus sisään	Y181
A1.3	%QX18.3	DO	Kuormitusventtiilin ohjaus auki	Y183
A1.3	%QX18.4	DO	Kuormitusventtiilin ohjaus kiinni	Y183
A1.3	%QX18.5	DO	Kuormitusventtiilin vapautus	Y185
A1.3	%QX18.6	DO	Paineakkujen pois kytkentä	Y186
A1.3	%QX18.7	DO	Epäpuhtauden ruiskutus	Y187
A1.4	%QX38.0	DO	Ohjaujännite	K10
A1.4	%QX38.1	DO	Koneikon start/stop	Kaynnistys
A1.4	%QX38.2	DO	Varalla	
A1.4	%QX38.3	DO	Varalla	
A1.4	%QX38.4	DO	Varalla	
A1.4	%QX38.5	DO	Varalla	
A1.4	%QX38.6	DO	Varalla	
A1.4	%QX38.7	DO	Varalla	
A2.2	%QW1.0	PWM	Proportionaaliventtiili 1	PV1
A2.2	%QW1.2	PWM	Proportionaaliventtiili 2	PV2

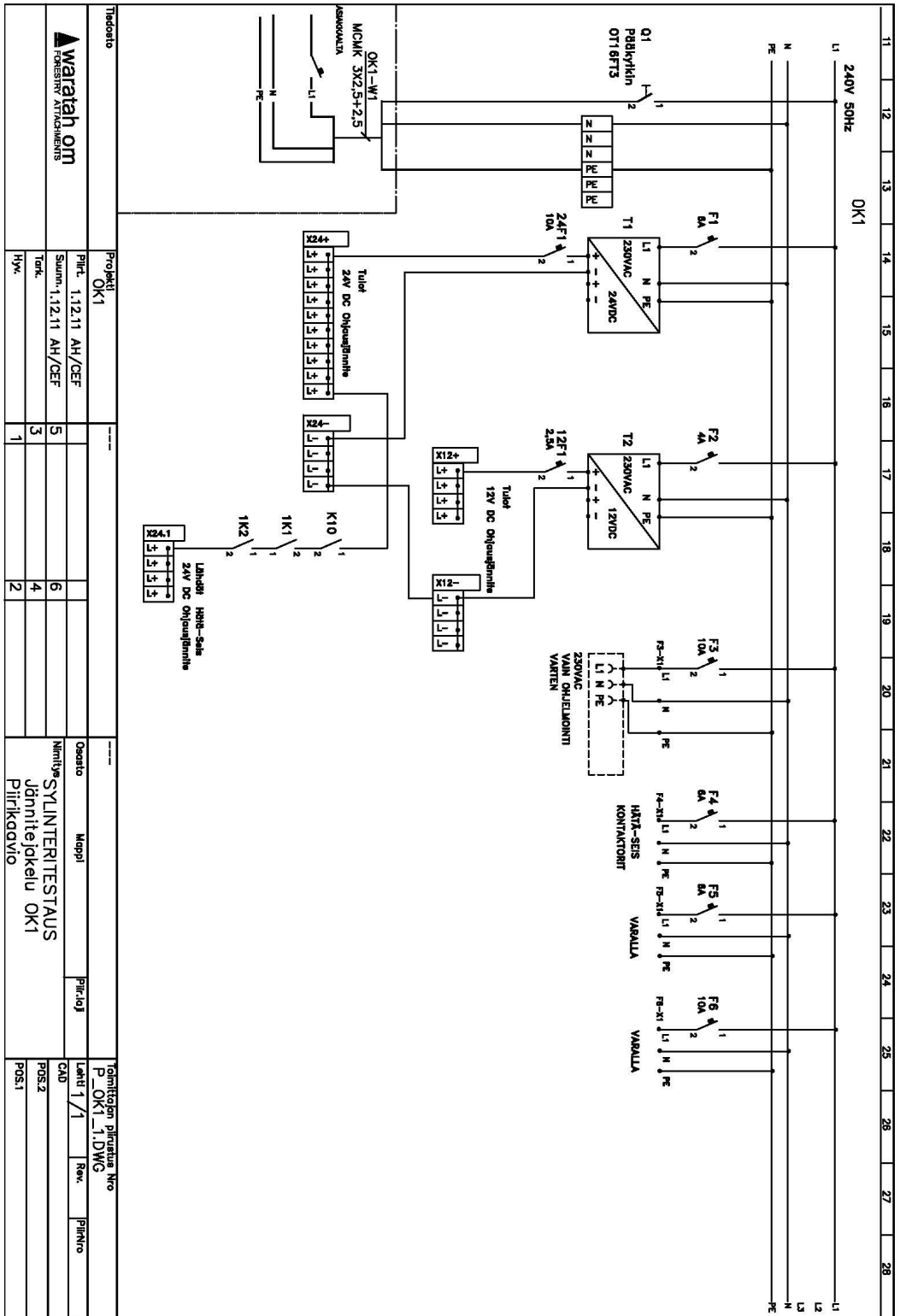
Liite 3. Ohjaukeskuksen Layout

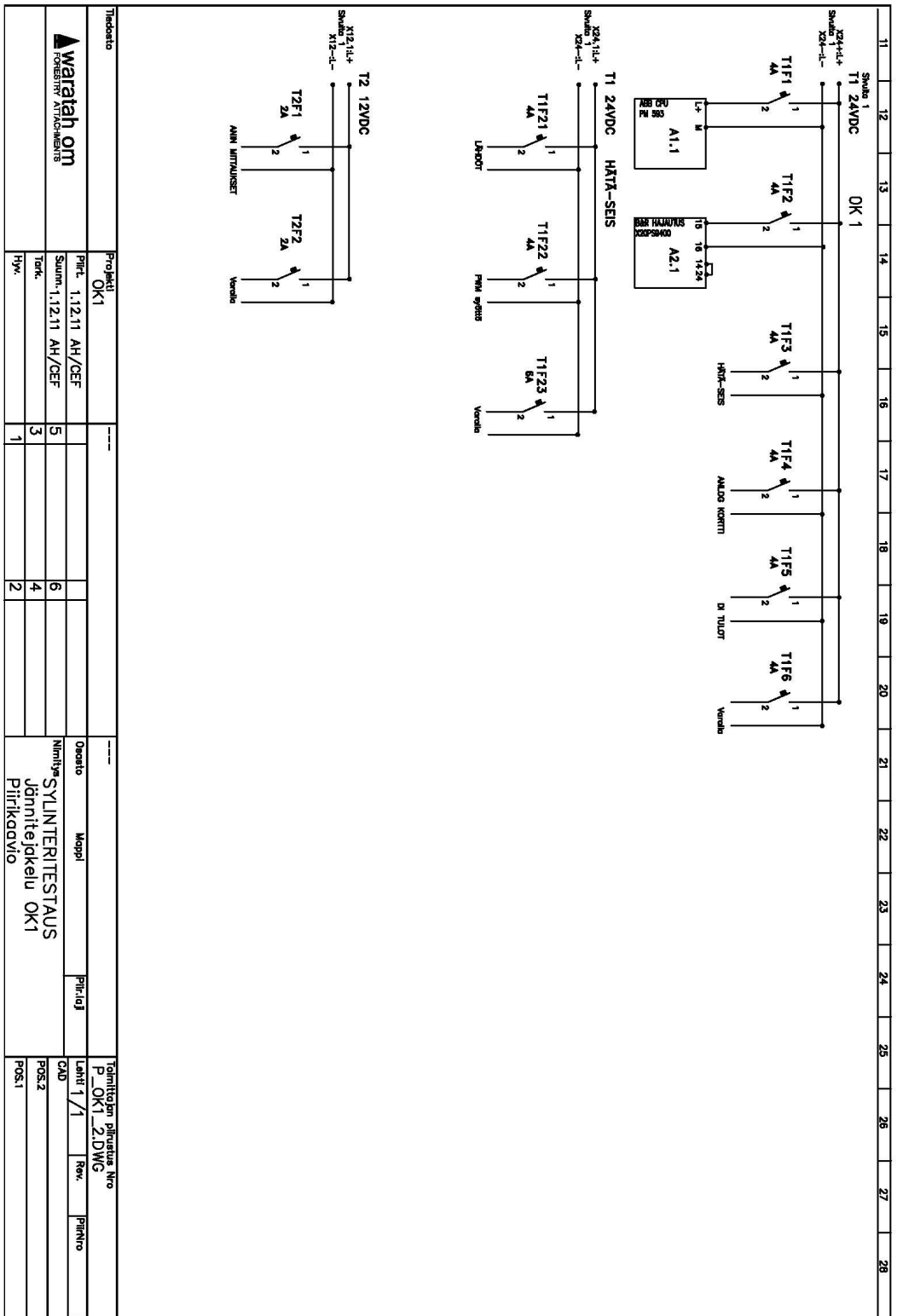


Tiedosto		Projekti		Osasto		Moppi		Päätyöj.		Tarkistus ja piirustus Nro	
waratah om FORESTRY ATTACHMENTS		Part. 1.12.11 AH/CEF		---		---		---		L_OK1_1.DWG	
		Saun. 1.12.11 AH/CEF		Nimike		Syllin terin testaus		CAD		Lähti 1/1	
		Tark. 3		OK1		OK1		POS.2		Rev	
		Hyt. 1		LAYOUT		LAYOUT		POS.1		Päivä	

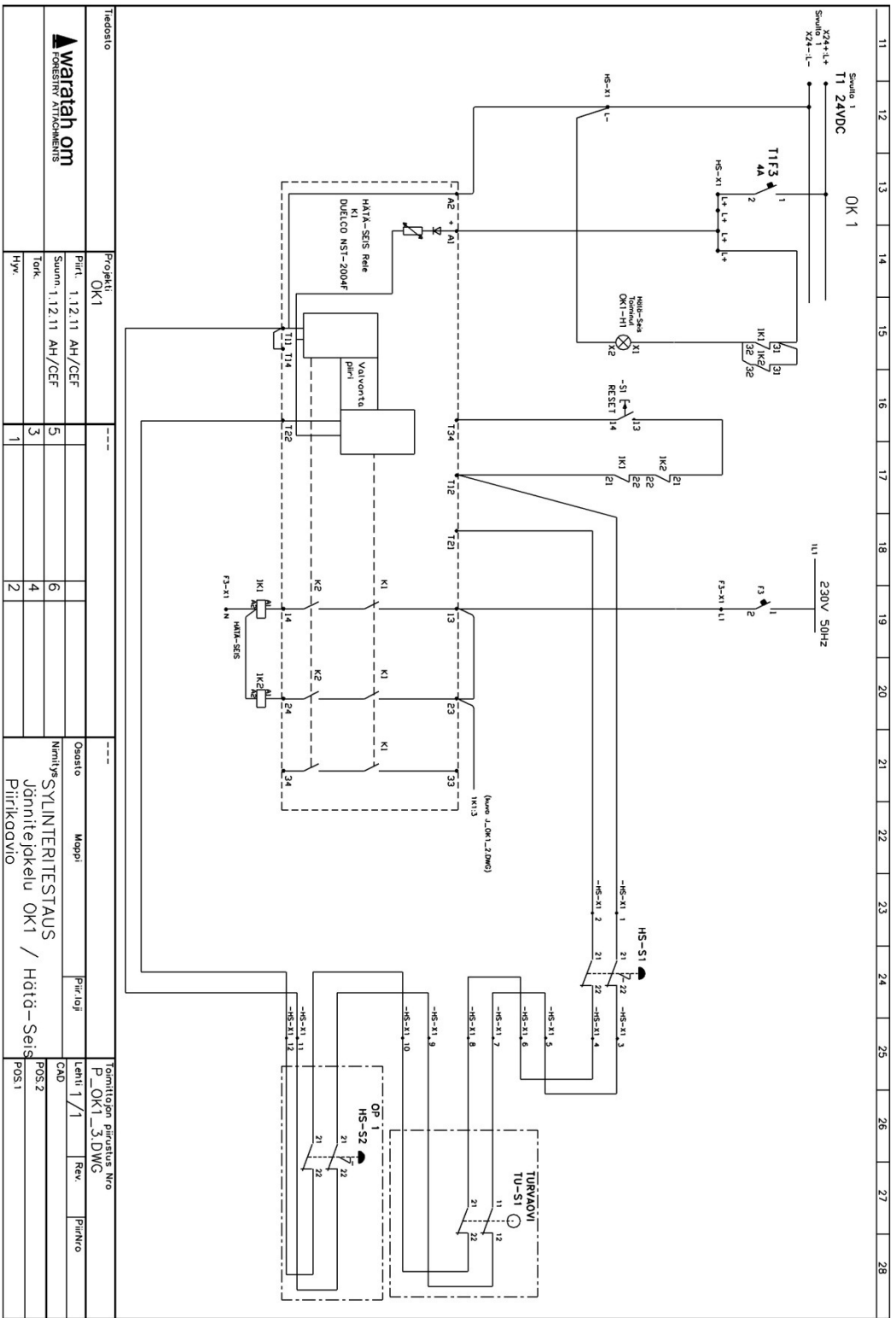
RITVA AE-KAAPPI 1180,500

Liite 4. Jännitteenjako

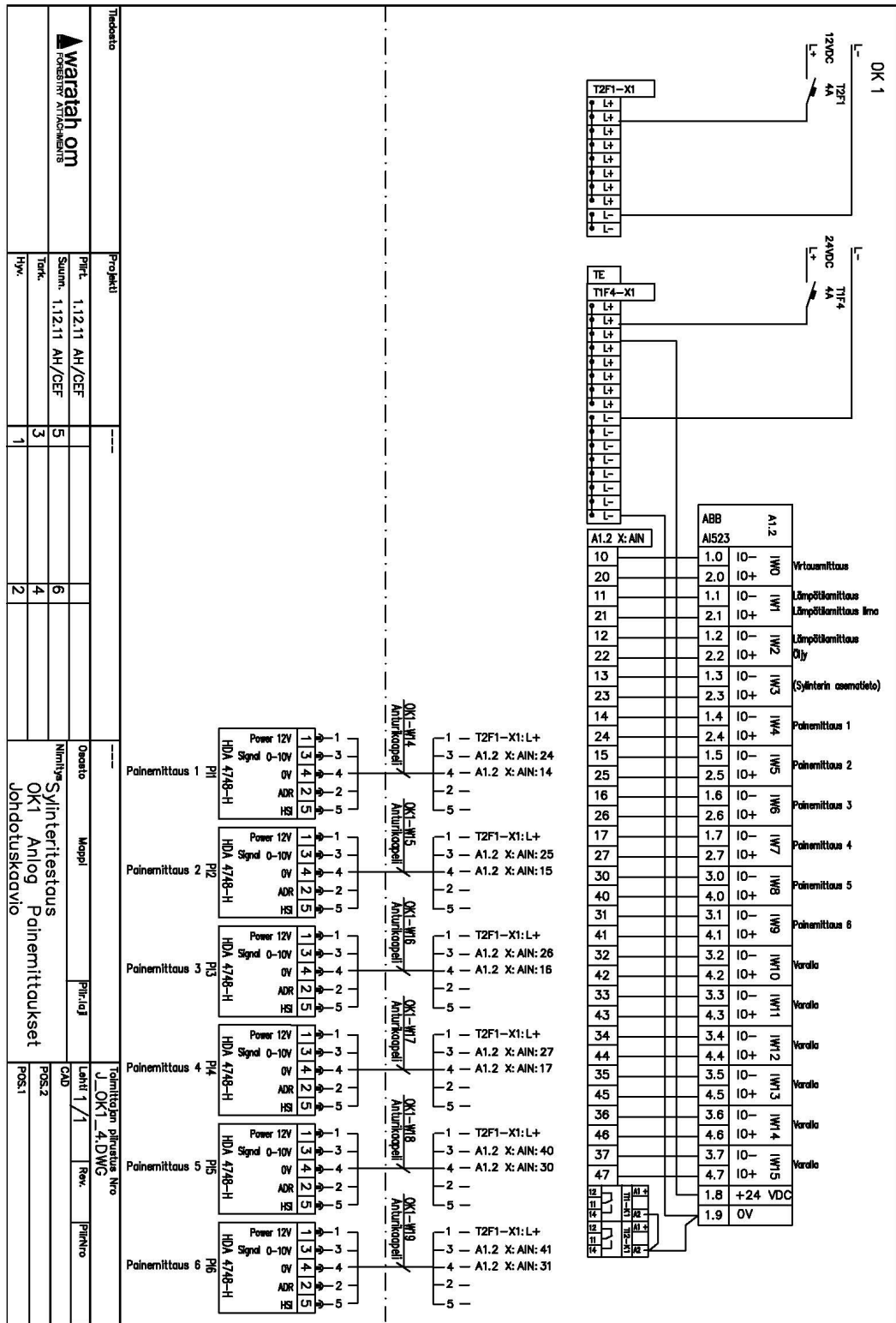




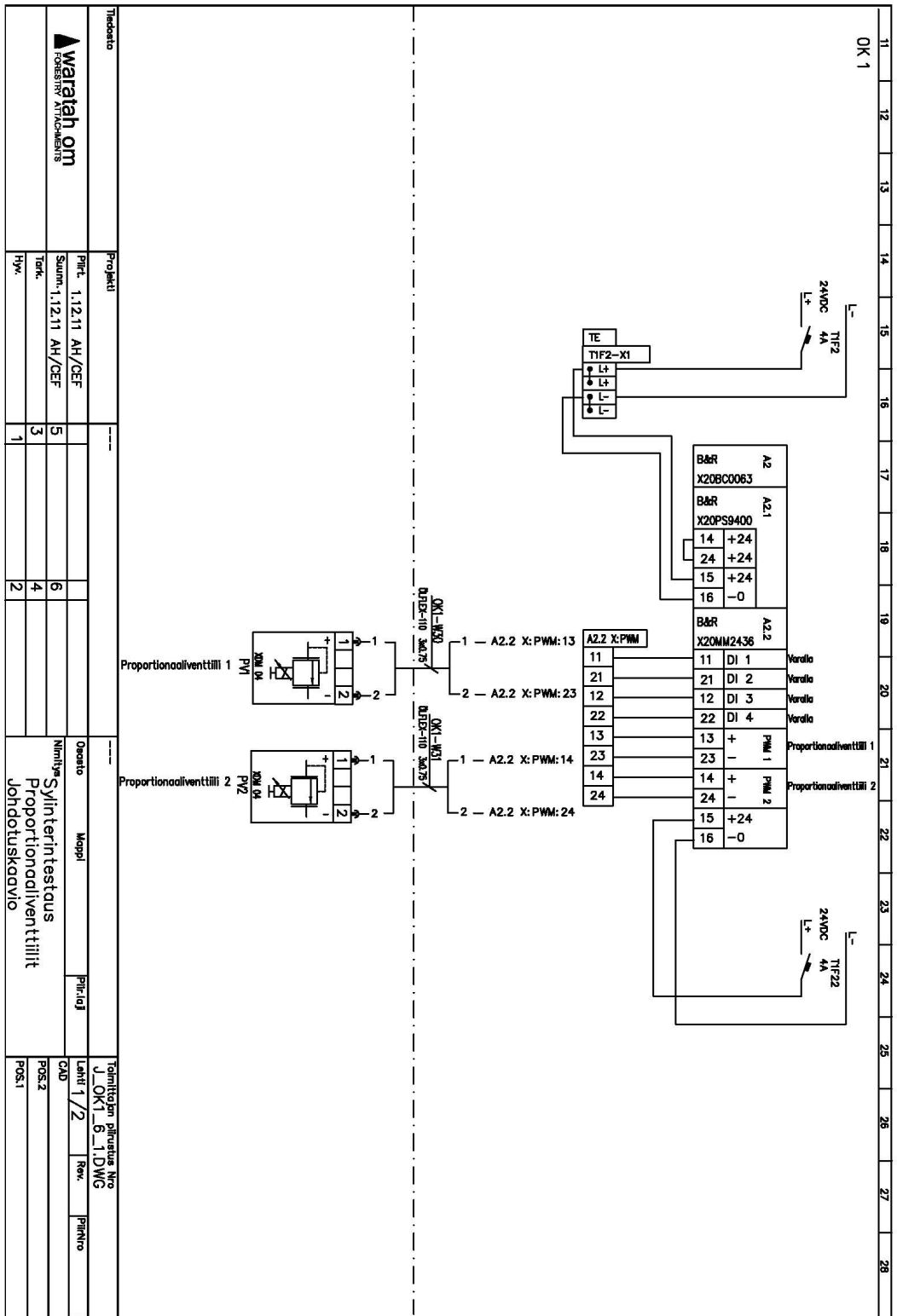
11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Sivusto 1 T1 24VDC OK 1																	
Sivusto 2 T2 12VDC																	
Projekt OK1																	
Pirtti 1,12,11 AH/CEF																	
Suunn. 1,12,11 AH/CEF																	
Tark.																	
Tyy.																	
Osa OK1																	
Moppi																	
Pirtti																	
Tehnteollisuuden osasto P-OK1_2.DWG																	
Lahdi 1 / 1																	
Rev.																	
Pirtti																	
Waratah om FORSEBY ATTACHMENTS																	
Nimety SYUNTERITESTAUS																	
Jännitejakeilu OK1																	
Pirttikäyttö																	
CAD																	
POS.2																	
POS.1																	



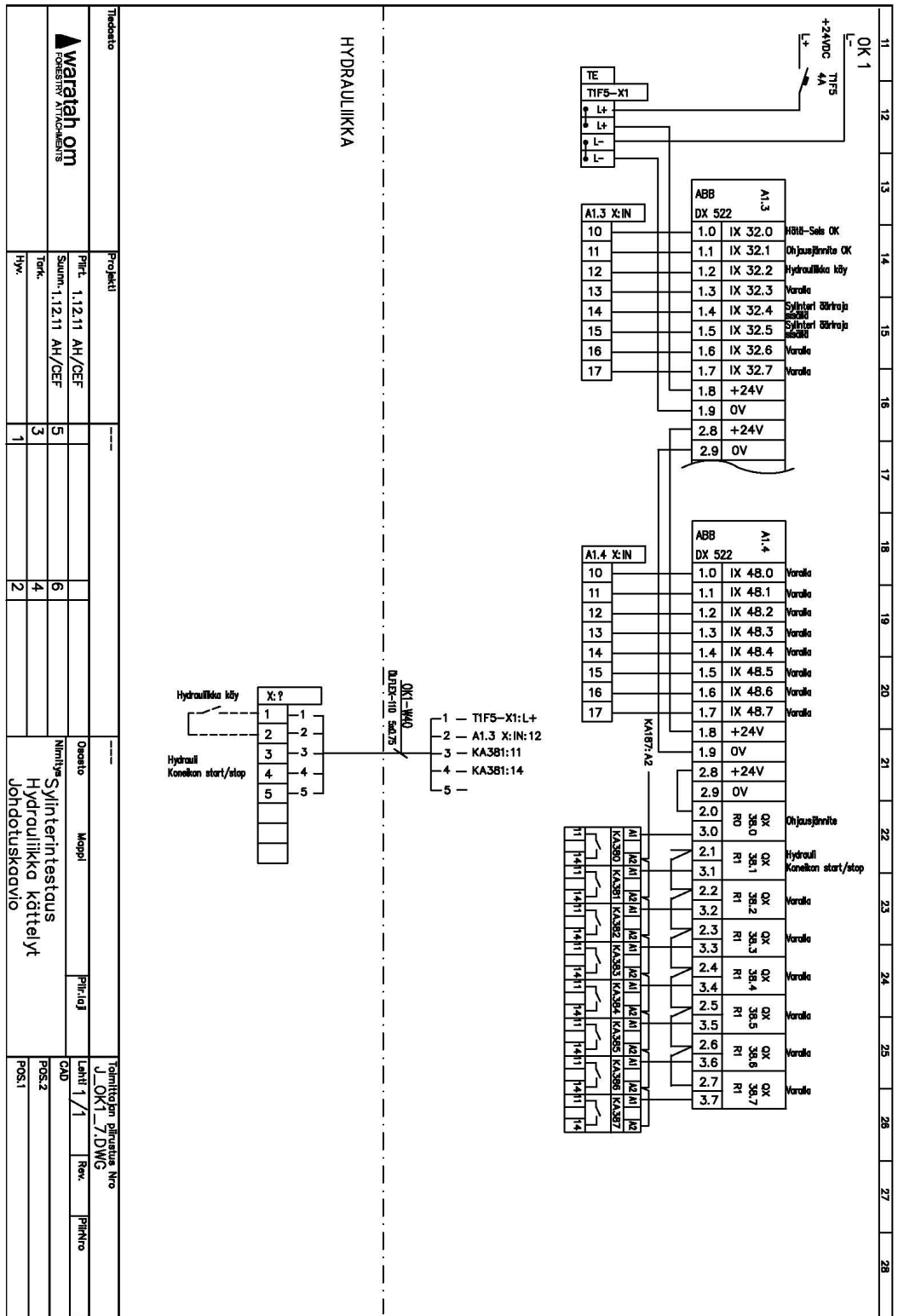
Tiedosto		Projekti		Osasto		Mappi		Pääloj		Tietojen perustus Nro	
waratah om		OK1		---		---		---		F_OK1_3.DWG	
FORSETR ATTACHMENTS		Puit. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1.12.11 AH/CEF		Suunn. 1.12.11 AH/CEF		5		6		---	
		Toik.		3		4		2		---	
		Hyv.		1		2		1		---	
		---		---		---		---		---	
		Suunn. 1									



Tiedosto		Projekti	
waratah om		1.12.11 AH/CEF	
FORSERY ATTACHMENTS		Suunn. 1.12.11 AH/CEF	
		Tekn. 3	
		Hyv. 1	
Osaeto		Kappi	
Nimike		Pii-Id	
Sylinteriasetus		Taittojen pöytänumero	
OK1 Anlog Painemittaukset		J_OK1_4.DWG	
Johdotuskaavio		Laitti 1/1	
		CAD	
		POS:2	
		POST:1	
		PiiNro	
		Rev.	



11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28																																																																																																												
OK 1																																																																																																																													
<table border="1"> <tr> <td>Projekti</td> <td colspan="17">Tiedosto</td> </tr> <tr> <td>Projekti</td> <td>1.12.11 AH/CEF</td> <td colspan="16"></td> </tr> <tr> <td>Suunn.</td> <td>1.12.11 AH/CEF</td> <td colspan="16"></td> </tr> <tr> <td>Tark.</td> <td></td> <td colspan="16"></td> </tr> <tr> <td>Hyv.</td> <td></td> <td colspan="16"></td> </tr> </table>																		Projekti	Tiedosto																	Projekti	1.12.11 AH/CEF																	Suunn.	1.12.11 AH/CEF																	Tark.																		Hyv.																																			
Projekti	Tiedosto																																																																																																																												
Projekti	1.12.11 AH/CEF																																																																																																																												
Suunn.	1.12.11 AH/CEF																																																																																																																												
Tark.																																																																																																																													
Hyv.																																																																																																																													
<table border="1"> <tr> <td>Osasto</td> <td>Moppi</td> <td colspan="16">Tehnteiden pöytäkirja Nro J_OK1_6_1.DWG</td> </tr> <tr> <td colspan="2">Nimike</td> <td colspan="16">Sylinterin testaus Proportionaaliventtiilit</td> </tr> <tr> <td colspan="2">Kuva</td> <td colspan="16">Johdotuskaavio</td> </tr> <tr> <td>Lahti</td> <td>1/2</td> <td>Rev.</td> <td colspan="15"></td> </tr> <tr> <td>POS.2</td> <td colspan="17"></td> </tr> <tr> <td>POS.1</td> <td colspan="17"></td> </tr> </table>																		Osasto	Moppi	Tehnteiden pöytäkirja Nro J_OK1_6_1.DWG																Nimike		Sylinterin testaus Proportionaaliventtiilit																Kuva		Johdotuskaavio																Lahti	1/2	Rev.																POS.2																		POS.1																	
Osasto	Moppi	Tehnteiden pöytäkirja Nro J_OK1_6_1.DWG																																																																																																																											
Nimike		Sylinterin testaus Proportionaaliventtiilit																																																																																																																											
Kuva		Johdotuskaavio																																																																																																																											
Lahti	1/2	Rev.																																																																																																																											
POS.2																																																																																																																													
POS.1																																																																																																																													



Tiedot		Projekti		Osa		Moppi		Taittojen pöytäkirja Nro	
▼ waratah om HOBESTRY ATTACHMENTS		Proj. 1.12.11 AH/CEF		Saun. 1.12.11 AH/CEF		Tark.		U_0K1_7.DWG	
		5		3		4		Rev.	
		1		2		POS.2		PiiNro	
		1		2		POS.1			