



PAINEKULJETTIMEN OHJAUS OHJELMOITAVALLA LOGIIKALLA

Joonas-Juho Kivinen

Opinnäytetyö
Huhtikuu 2013
Sähkötekniikka
Automaatiotekniikka

TAMPEREEN AMMATTIKORKEAKOULU
Tampere University of Applied Sciences

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Sähkötekniikka
Automaatiotekniikka

JOONAS-JUHO KIVINEN:
Painekuljettimen ohjaus ohjelmoitavalla logiikalla

Opinnäytetyö 33 sivua, joista liitteitä 8 sivua
Huhtikuu 2013

Opinnäytetyö toteutettiin Kopar konsernin Site Teollisuus Oy -yksikölle. Opinnäytetyön aiheena on laatia PLC-koodi ja HMI-käyttöliittymä painekuljettimelle. Samalla tehdään selvitys geneerisen ohjelmoinnin nykytilasta sekä sen hyödyntämismahdollisuuksista.

Koodi laaditaan Allen-Bradley Micro830 -logiikalle, koodin pohjana toimivat Siemensin S7-1200 ja S7-200 -logiikoiden koodit. Tavoitteena on laajentaa yrityksen PLC valikoimaa painekuljettimille ja laatia hyvin kommentoitu ja ammattimainen koodi. Uutena ominaisuutena joillekin yrityksen tarjoamille painekuljettimille tulee operointipaneeli.

Asiasanat: kuljetin, PLC, HMI, geneerinen, ohjelmointi,

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Electrical Engineering
Automation Technology

JOONAS-JUHO KIVINEN:
PLC Programming for Pressure Conveyor

Bachelor's thesis 33 pages, appendices 8 pages
April 2013

Thesis was done for Kopar group. Kopar group is an engineering and manufacturing corporation which produces bulk material handling solutions especially in non-ferrous smelter industries as well as in power and energy sectors. (Kopar[2], 2013)

Objective of the thesis is to program a PLC code and a HMI for pneumatic conveyor and to clarify the possibility for generic programming. Base for the new PLC code is formed by other PLC codes that are made for Siemens S7-1200 and S7-200 PLCs. The new code is made for Allen-Bradley Micro830 PLC. Purpose of the thesis is to broaden the PLC selection for pneumatic conveyors.

Key words: conveyor, PLC, HMI, generic, programming

SISÄLLYS

1	JOHDANTO.....	6
2	GENEERINEN OHJELMOINTI.....	7
2.1	Tarve	7
2.2	Standardit.....	7
2.2.1	SFS-EN 61131	8
2.2.2	IEC 61499 Function Blocks	10
2.3	Geneeriset ohjelmointiohjelmat.....	12
2.3.1	logi.cals	12
2.3.2	nxtControl	13
2.3.3	CoDeSys.....	13
3	TUHKAKULJETTIMEN OHJELMOINTI.....	14
3.1	Prosessi	14
3.2	Lähtötiedot.....	16
3.3	Ohjainlaitteisto.....	18
3.4	Connected Components Workbench.....	20
3.5	Vaatimukset	20
3.6	Koodi	21
3.7	Operointipaneeli.....	22
4	POHDINTAA.....	23
	LÄHTEET.....	25
	LIITTEET	26
	Liite 1. Esimerkki kuljettimen ohjelmoinnista tikapuukaaviota käyttäen	26
	Liite 2. Esimerkki kuljettimen ohjelmoinnista toimilohkokaaviota käyttäen	27
	Liite 3. Operointipaneelin näytöt.....	28

LYHENTEET JA TERMIT

TAMK	Tampereen ammattikorkeakoulu
op	opintopiste
PLC	programmable logic controller, ohjelmoitava logiikka
IPC	industrial PC, teollisuus PC
PAC	programmable automation controller, mikroprosessori- pohjainen kontrolleri
HMI	human machine interface, käyttöliittymä
geneerinen	yleinen, yleisluontoinen
modulaarinen	moduuleista koostuva
verkkotopologia	tapa jolla verkon laitteet on liitetty toisiinsa
POU	program organisation unit, ohjelman organisointiyksikkö
fluidisointi	kiinteän aineen kuohkeuttaminen ilman avulla

1 JOHDANTO

Kopar Group on konepajayritys, jolla on pitkäaikainen kokemus jauhemaisten materiaalien käsittelystä erityisesti vaativissa olosuhteissa. Kopar Groupiin kuuluu kolme toisiaan täydentävää yritystä; Kopar, Elmomet ja Site Teollisuus. Yrityksen vuotuinen liikevaihto on noin 20 miljoonaa euroa. Konserni työllistää yhteensä noin 130 henkilöä. (Kopar[2], 2013)

Kopar Group tuottaa mekaanisia ja pneumaattisia materiaalinkäsittelyjärjestelmiä, sulattojen prosessilaitteita, sekä raakaveden suodatuslaitteita. Koparin tuotteilla voidaan syöttää, kuljettaa, varastoida ja purkaa erilaisia materiaaleja aina hakkeesta kuparirikasteeseen ja tuhkasta ongelmajätteeseen. Järjestelmiä on käytössä mm. energia-, metallurgia-, sementti-, ja kemianteollisuudessa. (Kopar[2], 2013)

Käsittelen työssäni painekuljettimien ohjelmointia, kyseessä on paineilmalla toimiva järjestelmä, jossa aine etenee putkistoa pitkin. Painekuljettimella kuljetetaan prosesseissa jauhemaisia aineita suljetussa putkistossa. Suljettua kuljetinympäristöä hyödynnetään kun ei haluta kuljetettavan aineen olevan kosketuksissa ympäristön kanssa. Kuljettimia on tarjolla useita eri kokoja, kuljetettavan ainemäärän mukaan. (Kopar[2], 2013)

Yrityksen toimittamia painekuljettimia ohjataan tavanomaisesti Siemensin logiikoilla. Työssäni käänän koodin Allen-Bradleyn logiikalle, sekä toteutan käyttöliittymän saman valmistajan operointipaneelilla. Automaatiolaitteet, joihin tutustun tarkemmin työni aikana ovat Allen-Bradleyn valmistamia. Opettelen Allen-Bradleyn laitteistojen ohjelmointiin käytettävän Connected Components Workbench -ohjelman.

Lisäksi perehdyn geneeriseen ohjelmointiin. Eri laitevalmistajien toimittamat järjestelmät ovat erilaisia, mutta standardoinnin myötä ne kuitenkin lähentyvät toisiaan. Standardointi lisää eri automaatio-ohjelmistojen yhdenmukaisuutta, eli ne siirtyvät lähemmäksi geneerisyyttä. Selvitän työssäni standardoinnin nykytilaa ja kuinka pitkälle tekniikka on kehittynyt geneerisen ohjelmoinnin kannalta.

2 GENEERINEN OHJELMOINTI

2.1 Tarve

Parhaimmassa tapauksessa olisi vain yksi ohjelmisto, jolla voidaan ohjelmoida usean valmistajan toimittamia ohjelmoitavia logiikoita, geneerinen ohjelmointiohjelma. Tämän myötä myös laitevalmistajien ohjelmistokustannukset pienentyisivät. On olemassa voittoa tavoittelemattomia järjestöjä, jotka edistävät standardointia, kuten PLCopen, joka keskittyy IEC 61131 standardin kehittämiseen. (PLCopen, 2013)

Yrityksiltä vaaditaan tietotaitoa, jotta asiakkaan vaatimukset saadaan tyydytettyä. Automaation kannalta yksi näistä vaatimuksista voi olla esimerkiksi tietyn valmistajan PLC:n käyttäminen. On osattava eri automaatiovalmistajilta tulevat järjestelmät, jotta mahdollisimman monen asiakkaan vaatimuksiin kyetään vastaamaan. Ohjelmistojen merkitys ja siten hinta kasvavat niiden kehittyessä.

Useiden eri valmistajilta tulevien järjestelmien osaaminen ei ole standardoinnin myötä mahdotonta, mutta usean eri ohjelmiston ostaminen tai jonkin tietyn koodin teettäminen kolmannella yrityksellä lisää kustannuksia. Eikä yksin ohjelmiston ostaminen ja sen opetteleminen riitä, ohjelmistot kehittyvät jatkuvasti valmistajien lisätessä niihin ominaisuuksia ja päivityksetkin lisenssisopimuksesta riippuen voivat maksaa.

2.2 Standardit

Tilanteessa jossa kaikkia logiikoita voidaan ohjelmoida samalla ohjelmalla tulee standardoinnin olla sillä tasolla, että kaikki valmistajat noudattavat tuon hypoteettisen ohjelmointiohjelman tarjoamia ominaisuuksia. Aloitan perehtymisen geneeriseen ohjelmointiin standardoinnin kautta.

Standardoinnin myötä ohjelmistot ovat muuttuneet enemmän toisiaan muistuttaviksi, mikä on helpottanut mm. koulutusta. Kuitenkin jokaista eri automaatiolaitevalmistajan toimittamaa logiikka täytyy ohjelmoida valmistajan tarjoamalla ohjelmointiohjelmalla. Perehdyn seuraaviin standardeihin geneerisen ohjelmoinnin kannalta.

2.2.1 SFS-EN 61131

Kyseinen standardi polveutuu IEC (International Electro technical Commission) standardi 61131:sta. Standardin ensimmäinen versio julkaistiin vuonna 1993. Tämän jälkeen standardista on tullut laajasti hyväksytty laitevalmistajien keskuudessa. Standardi ei aluksi ollut kuin kokoelma eri valmistajien kehittämistä hyvistä ominaisuuksista. (RTAutomation, 2013)

Standardi määrittää PLC -järjestelmien minimivaatimukset niin fyysisen laitteiston kuin ohjelmistojen kannalta. Kyseinen standardi on enemmän ohjeistus kuin sääntö. Laitevalmistajien ei oleteta noudattavan suurta määrää standardissa määriteltyjä yksityiskoh-
tia, vaan osaa niistä. (John & Tiegelkamp 2010, 12)

Standardi on jaoteltu kahdeksaan osaan (John & Tiegelkamp 2010, 14):

1. Yleisinformaatio (General information)
2. Laitevaatimukset ja testit (Equipment requirements and tests)
3. Ohjelmointikielet (Programming languages)
4. Käyttöohjeistus (User guidelines)
5. Tiedonsiirtoliikennöinti (Communications)
6. Toiminnallinen turvallisuus (Functional safety) (ei SFS standardi)
7. Sumea säätö (Fuzzy control programming)
8. Ohjelmointikielten käyttö ja soveltaminen (Guidelines for the application and implementation of programming languages)

Tärkeimpänä PLC ohjelmoinnin kannalta voidaan pitää kohtaa kolme. Ohjelmointikielet määrittelevä SFS-EN 61131-3 muodostaa pohjan standardin muille osille, sekä suurimman hyödyn käyttäjälle helpottaessaan ohjelmoijien ja ylläpitohenkilöstön koulutusta. Standardiosa määrittää pienimmän toiminnallisuuden ohjelmistolle, standardiin kuuluu pakollisia, sekä paljon valinnaisia kohtia. (RTAutomation, 2013)

Ennen hyväksymistä laitevalmistajat sovelsivat omia ohjelmointikieliään, minkä seurauksena järjestelmien opettelu ja ylläpito vaati paljon koulutusta, aikaa ja näiden kautta rahaa. Uusien ja pienten yritysten oli lähes mahdotonta selvitä markkinoilla, koska ei haluttu kuluttaa resursseja uusien järjestelmien opetteluun. Standardi yhdisti useat eri laitevalmistajien käyttämät kielet viideksi standardikieleksi. (RTAutomation, 2013)

Standardissa määritellyt ohjelmointikielet ovat, toimilohkokaavio (FBD – function block diagram), tikapuukaavio (LD – ladder diagram), rakenteelliset tekstit (ST – structured text), käskylista (IL – instruction list) ja sekvenssikaavio (SFC – sequential function chart). Standardi vaatii tuen vähintään yhdelle edellä mainituista kielistä. (RTAutomation, 2013)

Varsinkin lähempänä konekieltä oleva rakenteellinen kieli ST mahdollistaa koodin leikkaamisen ohjelmointiohjelmalta toiselle, mikäli se ylipäättensä tukee kyseistä ohjelmointikieltä. Kuitenkin suurella todennäköisyydellä koodi ei toimi suoraan, mm. muistipaikkojen käyttö eroaa merkittävästi joidenkin valmistajien välillä. Graafisempien kielten siirtäminen em. tavalla on käytännössä mahdotonta.

Standardin täyttymiseen riittää pienimmän toiminnallisuuden toteutuminen (John & Tiegelkamp 2010, 12). Voidaan olettaa, että jokainen valmistaja kuitenkin määrittää ohjelmistoaan yli tuon minimitoiminnallisuuden, esimerkiksi riittää vain useamman standardin mukaisen ohjelmointikielen valinta tai omien toimilohkojen kehittäminen. Seuraa toisistaan eriäviä ohjelmistoja, jotka täyttävät standardia eri tavalla. Mistä puolestaan seuraa standardin mukaisia, mutta keskenään toimimattomia koodeja.

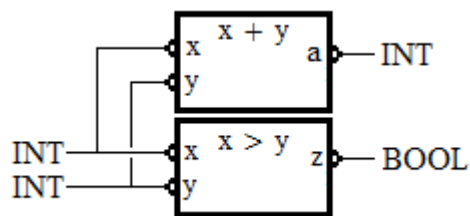
SFS-EN 61131-3 standardiin sisältyy 62-osainen taulukko, jonka valmistaja täyttää. Taulukosta käy ilmi mitkä osat kyseisestä standardista täyttyvät. Taulukkoja vertailemalla saadaan jonkinlainen käsitys valmistajien ohjelmointikielten eroavuuksista ja yhtäläisyyksistä. (John & Tiegelkamp 2010, 12)

SFS-EN 61131 luo pohjaa geneeriselle ohjelmoinnille, siihen pohjautuvia yleisiä ohjelmointi- ja dokumentointiohjelmaa on jo olemassa. Standardi ei kuitenkaan velvoita valmistajia tukemaan niitä. Varsinkin isot valmistajat kehittävät ohjelmistojaan standardia pidemmälle, jolloin ero ohjelmistojen välillä kasvaa ja standardi jää jälkeen. Myös loogiikkojen muistipaikkojen ja I/O:n käyttö eroavat merkittävästi toisistaan, standardikaan ei ota niihin kantaa.

2.2.2 IEC 61499 Function Blocks

Standardiin sisältyy geneerinen ohjelmointikieli, eikä kaikkien ohjauslaitteiden tarvitse olla samalta valmistajalta. Standardin keskeisimpänä ideana on automaatiojärjestelmien modulaarisuuden lisääminen niin kenttä- kuin ohjelmistotasolla. Kyseessä on IEC:n vuonna 2005 julkaisema standardi, jossa käsitellään uutta lähestymistapaa teollisuusautomaatioon. Instrumentointitilaan sijoitettujen isojen PLC:iden sijasta käytettäisiin jollain verkkotopologialla kentälle hajautettuja IPC sekä PAC pohjaisia ohjausjärjestelmiä. (Vyatkin[1], 2013)

Ohjelmointikieli perustuu tapahtumadiskreetteihin toimilohkoihin, joita ei tule sekoittaa SFS-EN 61131-3 standardin aikadiskreetteihin FBD:n toimilohkoihin (KUVIO 1). Tapahtumadiskreetti toimilohko suoritetaan vain silloin kun sen tapahtuma (event) sisään-tuloon tulee signaali, tapahtuman tuloja ja lähtöjä voi olla useampi kappale. Tapahtumat päivittävät niihin määritellyt data sisään- ja ulostulot, jotka voivat olla standardissa SFS-EN 61131 määriteltyjä datatyyppejä. Aikadiskreetti toimilohko puolestaan suoritetaan jokaisella ohjelmasyklillä. (Vyatkin[1], 2013)

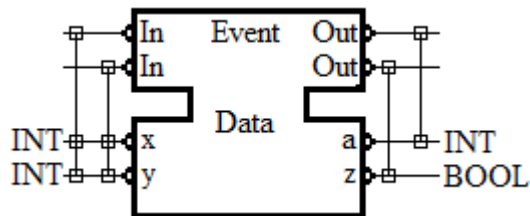


KUVIO 1: Standardin SFS-EN 61131-3 mukainen aikadiskreetti lohkokaavio.

Tapahtumapohjaisella suorituksella lisätään ohjelman tehokkuutta, vain tarpeelliset toiminnot suoritetaan kullakin ohjelmasyklillä. Hajautetussa järjestelmässä tämä on tärkeää sillä lohkojen suorittamatta jättäminen vähentää ylimääräistä liikennöintiä ohjainten välisessä verkossa. (Wenbin Dai & Vyatkin, 2013)

Standardi määrittelee kaksi toimilohkotyyppiä, perustoimilohko (basic) ja komposiittitoimilohko (composite). Toimilohkojen suorittamia funktioita ei standardissa määritellä, vaan ne ohjelmoidaan erikseen. Perustoimilohko sisältää tilakoneen, jonka eri tiloissa suoritetaan niihin ohjelmoidut toiminnot. Toimintojen luonnissa voi käyttää standardissa SFS EN 61131-3 määriteltyjä ohjelmointikieliä, PLC koodeja voi siis käyttää uudes-

taan, lisäksi standardi tukee korkeamman tason C- ja Java-kieliä. Kuvion 2 tapahtumadiskreetti toimilohko voi sisältää kuviossa 1 esitellyt toiminnot. Esim. alempaan tapahtumaan (event) tuleva pulssi suorittaa vertailun ja ylempi tapahtuma yhteenlaskun. Kuviossa näkyvät pystysuorat lohkon ulkopuoliset viivat kuvaavat mikä tapahtuma vaikuttaa mihinkäkin tulon/lähtöön. (Wenbin Dai W & Vyatkin, 2013)



KUVIO 2: Standardin IEC 61499 mukainen tapahtumadiskreetti toimilohko.

Komposiittitoimilohko koostuu perustoimilohkoista ja toisista komposiittitoimilohkoista. Niiden avulla muodostetaan ohjelmistohierarkia, ylin komposiittitoimilohko ja siinä olevat perustoimilohkot suoritetaan ensin, mikäli tarvetta, jonka jälkeen siirrytään syvemmälle ohjelmassa. Ohjelmiston suurin modulaarisuus saavutetaan kun ohjelmoidaan yksi toimilohko vastaamaan yhtä kentällä olevaa laitetta. Luodut toimilohkot tallennetaan kirjastoon ja niitä voi käyttää uudestaan toisessa projektissa. (Wenbin Dai & Vyatkin, 2013)

Standardi mahdollistaa kaikkien kentällä olevien laitteiden ohjelmoinnin yhdestä pisteestä samaan aikaan, toisin kuin perinteisissä järjestelmissä, joissa kullekin PLC:lle siirretään ohjelma erikseen. Prosessikokonaisuudesta saadaan modulaarisempi, sillä geneerisen ohjelmointikielen ansiosta kerran tehty ohjelma voidaan siirtää usean eri valmistajan laitteistolle. (Wenbin Dai & Vyatkin, 2013)

Standardi on vielä pitkälti kehitysasteella, mutta on jo olemassa valmistajia jotka tukevat sitä. Standardista on löytynyt epäkohtia, eräissä julkaisuissa on todettu, että sama ohjelma eri laitteistolla muodostaa erillaisia ohjauksia. Uuden standardin kehittäminen ja käyttöönotto tulee viemään vuosia. Mitä monimutkaisemmaksi automaatiojärjestelmät kehittyvät sitä tärkeämmäksi standardi IEC 61499 muodostuu. (Vyatkin[2], 2010)

2.3 Generiset ohjelmointiohjelmat

Tällä hetkellä on tarjolla geneerisiä ohjelmointiohjelmiä jotka noudattavat edellä käsitellyjä standardeja. Ne eivät kuitenkaan sovellu kaikkien PLC:den ohjelmointiin, vaan ovat enemmänkin IPC pohjaisia. Tutustun kolmen eri yrityksen tarjoamiin ohjelmistoihin pintapuolisesti yritysten omien internetsivujen kautta. Seurauksena ohjelmista antamani kuva voi olla harhaanjohtava.

2.3.1 logi.cals

Itävaltalaisen Kirchner Softin tekemä logi.cals ohjelmistopaketti tarjoaa työkaluja automaatiojärjestelmien suunnitteluun, ohjelmointiin ja ylläpitoon. Ohjelmistopaketti tarjoaa geneeristä ohjelmointia standardin IEC 61131-3 tarjoamalla kielillä IPC pohjaisilla laitteilla, sekä työkalut kokonaisten automaatiojärjestelmien suunnitteluun. Yrityksen tarjoamia ohjelmistoja ovat (logi.cals 2013):

logi.DOC	Lohkokaaviotyypinen mallintamistyökalu tehdas-suunnitteluun.
logi.CAD	IEC 61131-3 standardiin perustuva ohjelmoinnin-mallintamistyökalu.
logi.CED	'Syy ja seuraus'-taulukoiden luomisohjelma normaali- ja turvalogiikoiden määrittämiseen.
logi.DICT	Tietokantapohjainen sanakirja. (voidaan liittää kaikkiin ohjelmiin)
logi.VIS	Tarjoaa HMI-editorin ja .NET visualisoinnin
logi.RTS	Tekee IPC:stä PLC:n tavoin käyttäytyvän.
logi.PLC	5200 (MicroSys) ja CX (Bechhoff), yhtiön kumppaneiden tarjoamat ohjainlaitteet. Myös mikä tahansa muu IPC käy.
logi.CODE	Kokoelma komponentteja logi.CAD:in säätämiseen kohdejärjestelmään sopivaksi.
serverCOMP	OPC, WEB ja/tai COM/DCOM pohjainen ohjain-valvomo liittymä.

logi.SYS	Lisäosa jonka avulla logi.CAD muuttuu mallintamis-työkalusta ohjelmointitykaluksi.
logi.DIAG	Diagnisointityökalu järjestelmän ja ohjelmiston ennaltaehkäisevään huoltoon.
logi.GUARD	Monitorointityökalu

2.3.2 **nxtControl**

Ohjelmistoa pidetään ensimmäisenä kunnollisena IEC 61499 standardiin pohjautuvana ohjelmointiohjelmana. Ohjelmisto on suunniteltu hyödynnettäväksi talo-, kone- ja prosessiautomaatioissa. myös IEC 61131 standardia hyödynnetään IEC 61499 standardin lohkojen luonnissa. Ohjelma keskittyy IEC 61499 standardiin ja sillä ohjelmoidaan vain IPC ja PAC pohjaisia ohjausjärjestelmiä. (nxtControl, 2013)

2.3.3 **CoDeSys**

CoDeSys (Controlled Development System) on ohjelmointiin keskittyvä ohjelmistopaketti, jonka editointi ja simulointiohjelmat perustuvat kehittyneisiin ohjelmointikieliin, kuten Visual ja C++. Esittelemistäni ohjelmistoista CoDeSys on ainoa PLC ohjelmointiin soveltuva geneerinen ohjelmointiohjelma. Yhtiön sivuilta löytyy lista ohjelman tukemista ohjainlaitteistoista (CoDeSys[2], 2013) Listasta löytyy muun muassa kymmeniä eri PLC:itä, pääosin pieniltä valmistajilta. Listasta ei löydy muun muassa Allen-Bradleyn eikä Siemensin logiikoita. Ohjelmisto tukee myös useita eri verkkotopologioita, kuten Profibus, Profinet, Ethernet IP, EtherCAT, Sercos III, CANopen, ASi, DeviceNet ja J1939 IO-Link. (CoDeSys[1], 2013)

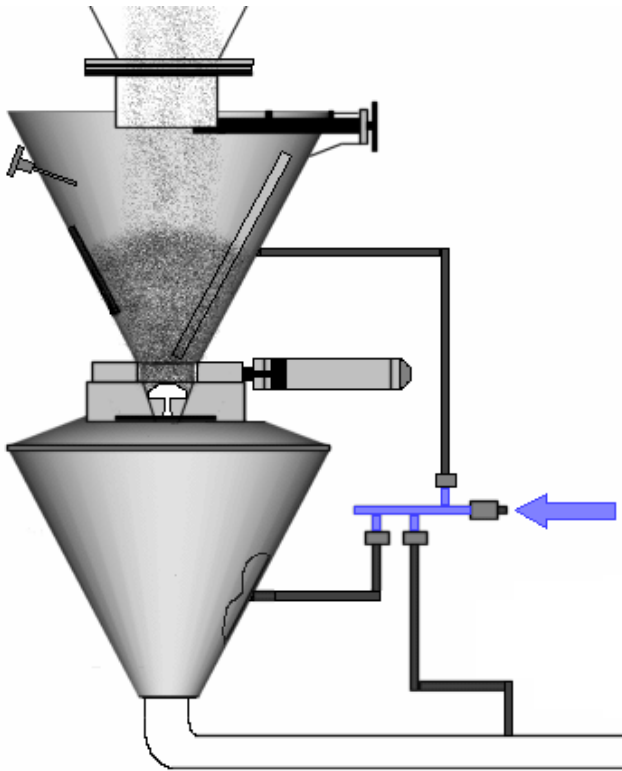
3 TUHKAKULJETTIMEN OHJELMOINTI

3.1 Prosessi

Puhun työssäni peruspainekuljettimista, joilla tarkoitan yrityksen painekuljetinlistasta löytyviä ELMO kuljettimia. Räätelöidyllä painekuljettimella puolestaan tarkoitan asiakkaan tarpeisiin yksilöllisesti tehtyä kuljetinta. Yrityksellä on kattava kokemus peruspainekuljettimista ja niitä tarjotaan asiakkaille valmiina kokonaisuuksina. Peruspainekuljettimen käyttöönotto vie vähän aikaa, eikä sitä aina edes tarvita. Työssäni laajennan peruspainekuljettimien PLC-valikoimaa Allen-Bradleyn logiikalla.

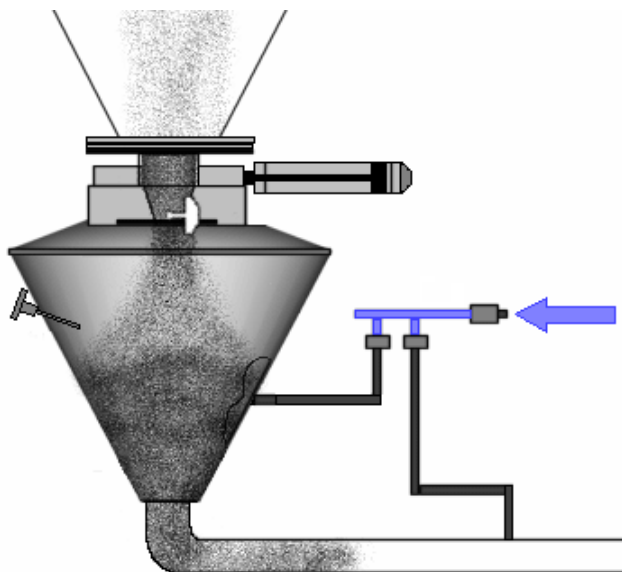
Peruspainekuljetin koostuu yleensä kahdesta päällekkäin asennetusta ja tilavuudeltaan yhtä isosta säiliöstä. Säiliöiden välissä on kalottiventtiili. Ylemmässä säiliössä on rajanturi, joka havaitsee säiliön täyttymisen. Täyttymisen jälkeen kuljetettava aine fluidisoidaan paineilman avulla. Kalottiventtiili avataan ja kuljetettava aine tippuu alempaan säiliöön.

Kuljetus tapahtuu kahden vuorottelevan painesyötön avulla alempaan säiliöön kiinnitettyä kuljetusputkea pitkin. Ensimmäinen syöttö on alemmassa säiliössä ja toinen putkessa, vuorottelun avulla kuljetettava aine pulssitetaan pienempiin osiin, jolloin tukoksen riski putkessa pienenee. Lopuksi varmistetaan putken tyhjeneminen puhaltamalla molemmilla syötöillä. Prosessin periaatekuva esitettynä kuviossa kolme (3).



KUVIO 3 Perustuhkakuljettimen periaatekuva, kuljetin täyttymässä. Paineensyötöt fluidisointia ja kuljetusta varten kuvattuna magneettiventtiilien kautta. (Kopar[1])

Kuljetin on mahdollista toteuttaa myös yhdellä säiliöllä. Tällöin pinta-anturi sijaitsee alemmassa säiliössä ja kalottiventtiiliä käytetään täyttöön. Tässä tapauksessa aineen fluidisaatio tapahtuu vain alemmassa säiliössä, samasta magneettiventtiilistä säiliön kuljetusilmansyötön kanssa. Kuljetus tapahtuu samalla periaatteella pulssittamalla kuin kahden säiliön versioissa. Yhden säiliön kuljetin kuvattuna kuviossa neljä (4).



KUVIO 4 Tuhkakuljetin ilman yläsäiliötä. (Kopar[1])

3.2 Lähtötiedot

Ohjelmoinnin lähtötietoina käytin ELMO 100 kuljettimen toimintaselostusta, sekä valmiita koodeja. Koodit ovat Siemensin logiikalle ja HMI:lle, molemmat oltiin tehty Siemensin Totally Integrated Automation V11 SP2 -ohjelmalla samaan tiedostoon. Koodi oli tehty räätälöidylle tuhkakuljettimelle, jonka toiminta eroaa peruskuljettimesta, tässä tapauksessa tuhkakuljettimia toimi kaksi rinnakkain (KUVA 1). Operointipaneeleja on yksi kutakin tuhkakuljetinta kohti. Koodi on FBD muodossa ja se on jaoteltu 42 piiriin.

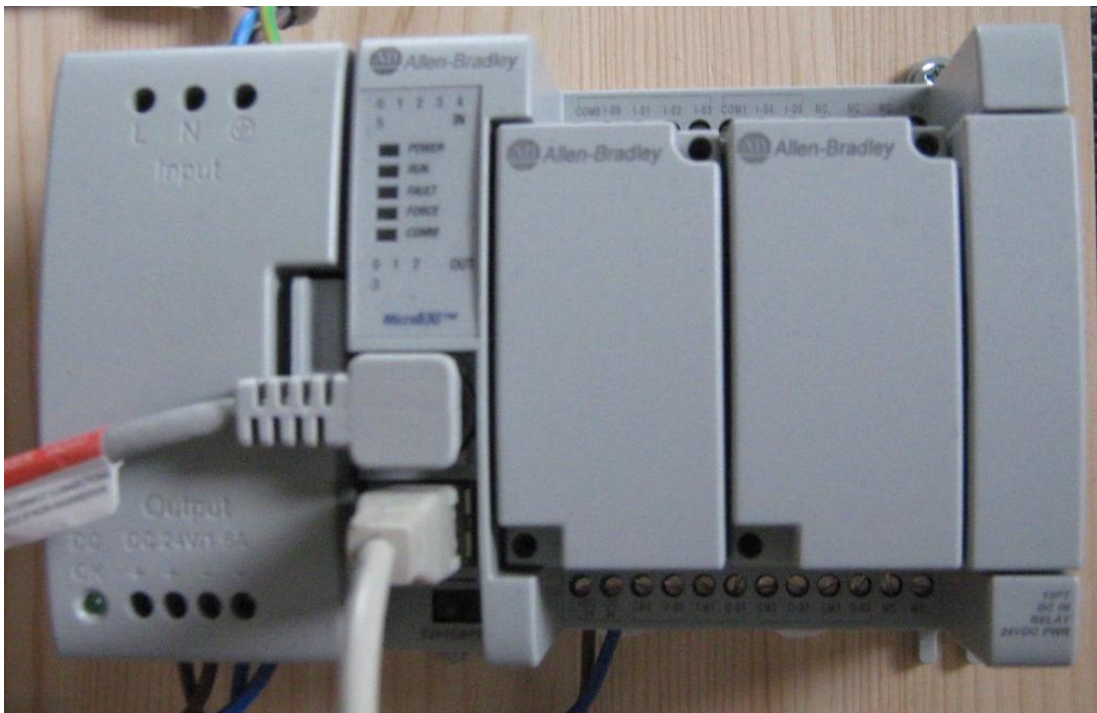


KUVA 1 Kaksi räätälöityä tuhkakuljetinta rinnakkain, esiasennettuna. (Kopar[1])

TAMKissa oli tuolloin TIA V11 SP1, eikä koodista auennut kuin HMI -osio. Päivitin yhden tietokoneen TIA:n SP2 tasolle, mikä ei sujunut täysin ongelmitta. Siemensin päivitysohjelma ei suoriutunut, koska se ei pystynyt lukemaan Microsoft Officen luomaa tietokoneen virtuaalilevyä. Päivityskohde ei sijainnut tuolla virtuaalilevyllä, mutta jos-

tain syystä päivitysohjelman on kyettävä lukemaan kakilta tietokoneen asemilta, ongelma ratkesi kun poistin Microsoft Officen.

Koodia purettaessa keräsin taulukkoon kaikki koodin sisältämät muistipaikat ja I/O:n datatyypeineen. Koodi sisältää yli 100 sisäistä muistipaikkaa, binäärilähtöjä ja -tuloja yhdeksän (9) kappaletta kumpaakin ja kaksi (2) analogia sisääntuloa. I/O:n lukumäärä aiheutti ongelman, työtäni varten toimitettu Allen-Bradleyn Micro830 2080-LC30-10QWB -logiikka (KUVA 2) ei sisältänyt yhtäkään analogiakanavaa, myös binäärikanavia oli liian vähän, 6 sisään ja 4 ulos.



KUVA 2 Micro830-LC30-10QWB -logiikka. Vasempaan reunaan kiinnitettyä virtalähde. Kuvassa näkyy paikat plug-in moduuleille.

Käyttöön toimitetussa logiikassa on kaksi laajennuspaikkaa, joista toiseen saa analogialaajennuksen, mutta tarpeeksi isoja binäärisiä laajennuskomponentteja ei ole tarjolla. Päädyttiin tilaamaan uusi logiikka, Allen-Bradleyn Micro830 2080-LC30-24QWB ja laajennusmoduuli analogiasignaalien lukemiseen.

3.3 Ohjainlaitteisto

Räätälöidyn tuhkakuljettimen ohjaus on tehty Siemens S7-1200 –logiikalle, jonka käyttöliittymänä Siemens KTP 600 -operointipaneeli. Peruspainekuljettimia ohjataan Siemens s7-200 -logiikoilla ilman operointipaneelia. Peruspainekuljettimen tilan kuvaamiseen käytetään kenttäkoteloon asennettuja valoja. Kuvassa 3 on esiteltynä pieni painekuljetin.



KUVA 3 Painekuljettimen kenttäkoteloon on kuljettimen tilaa kuvaavia valoja, sekä käynnistyskytkin. (Kopar[1])

Laitteisto, johon uusi ohjelma käännetään on Allen-Bradleyn Micro830 –logiikka, jonka käyttöliittymänä saman valmistajan PanelView C400 -operointipaneeli (KUVA 4). Operointipaneelissa on värillinen neljän tuuman kosketusnäyttö, jonka resoluutiona 480x272. Tarkoituksena on laatia räätälöidyn tuhkakuljettimen pohjalta perustuhkakuljettimelle operointipaneeli.



KUVA 4 Operointipaneeli, johon tuhkakuljettimen käyttöliittymä laaditaan.

Logiikka on yhteydessä tietokoneeseen USB-kaapelilla. HMI:n ohjelmointi voidaan suorittaa ethernet väylää pitkin, USB-kaapelilla tai USB-tikulla. USB-kaapelilla ohjelmointi vaatii ajureiden asentamisen, eikä tietokoneeni nykyinen käyttöjärjestelmä tukenut kyseisiä ajureita joten jäljelle jäi USB-tikku ja Ethernet. Valitsin edellisistä USB-tikun.

Logiikka ja HMI ovat toisiinsa yhteydessä COM-johtoa pitkin. Ensimmäisiä kertoja laitteistoa testatessani logiikan ja HMI:n välinen yhteys ei toiminut. Selvisi, että tämä johtui logiikan ja HMI:n yhteensopimattomista sisäisistä ohjelmistoversioista. Vika ratkesi kun päivitin logiikan revision tasosta 1.013 tasolle 2.011 ja HMI:n tasolta 1.040 tasolle 1.070.

Logiikan päivitys tapahtui USB-kaapelia pitkin Allen-Bradleyn omaa päivittämiseen tarkoitettua ControlFLASH ohjelmaa käyttäen. HMI:n päivitys tapahtui lataamalla päivitystiedosto tietokoneella USB-tikulle, kiinnittämällä USB-tikku HMI:hin ja suorittamalla päivitys HMI:llä. Päivitykset ja niihin tarvittavat ohjelmat ovat vapaasti ladattavissa Rockwell Automationin internetsivuilla.

3.4 Connected Components Workbench

Ohjelmisto on ladattavissa Rockwell Automationin internetsivuilta ilmaiseksi. Sillä ohjelmoidaan Micromaster sarjan pienlogiikoita sekä erilaisia Allen-Bradleyn operointipaneeleja. Ohjelmointiohjelma tukee standardin IEC 61131-3 määrittelemistä ohjelmointikielistä rakenteellista tekstiä (ST – structured text), toimilohkokaaviota (FBD – function block diagram) ja tikapuukaaviota (LD – ladder diagram).

Tutustuessani ohjelmointiohjelmaan sen viimeisin versio oli 2.00. Ohjelma oli helpokäyttöinen, eikä tutustumiseen mennyt paljon aikaa. Kyseisessä versiossa oli kuitenkin paljon vikoja, mm. tallennettua operointipaneelia ei pystynyt avaamaan uudestaan, vaan se täytyi aloittaa aina alusta. Pian kuitenkin julkaistiin kyseisen ohjelman versio 3.00 ja operointipaneelia pystyi ohjelmoimaan normaalisti. Ohjelmointiohjelma kaatui uudesta versioista huolimatta säännöllisesti ja työ tuli tallentaa usein.

3.5 Vaatimukset

Peruspainekuljettimia on tarkoitus käyttää samalla kenttäkotelolla. Ohjelmoinnin kannalta tämä tarkoittaa, että kuljetuskapasiteetiltaan erilaisia painekuljettimia ohjataan samalla logiikalla ja samalla ohjelmalla. Ohjelmaan sisältyy noin 20 eri prosessiparametriä, joiden perusteella painekuljetinta ohjataan. Prosessiparametrit eroavat kuljettimien välillä ja niistä käy ilmi muunmuassa kuljetuspaineen skaala ja -rajat, sekä tieto yläsäiliön käytöstä.

Prosessiparametreille luodaan laskennalliset lähtöarvot, mutta käyttöönotossa näitä arvoja voidaan joutua muuttamaan. Prosessiparametrien muuttamiseen vaaditaan ohjelmointityökalu, eli tietokone, jolla muokataan PLC:n koodia. Ohjelmointitietokoneen käyttö parametrien muuttamiseen kenttäoloissa on usein hidasta ja vaivalloista.

Ohjelmointityökalu voidaan korvata tässä tapauksessa operointipaneelilla, johon on ohjelmoitu valikko parametrien muuttamista varten. Yritys on toimittanut eräälle asiakkaalle Siemensin operointipaneelilla varustetun räätälöidyn painekuljettimen, kyseinen kuljetin eroaa jonkin verran yrityksen peruskuljettimista. Laadin työssäni peruskuljetti-

mille operointipaneelin, jonka pohjana käytän räätälöidylle kuljettimelle laadittua Siemensin paneelia.

3.6 Koodi

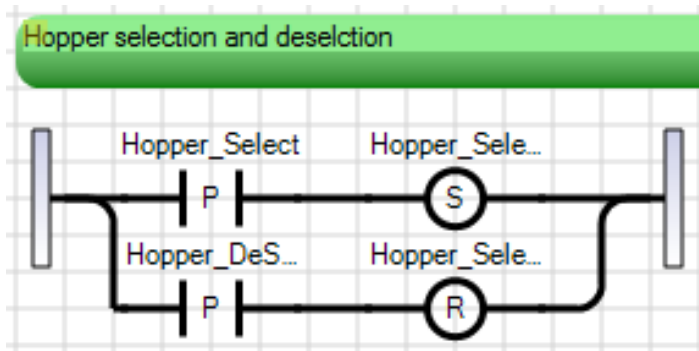
Laatimani koodi koostuu kymmenestä (10) pääohjelmasta, joissa on yhteensä noin 40 piiriä. Koodi toteutettiin sekvenssityyppisesti, ohjelman käynnistyessä siirrytään ensimmäiseen pääsekvenssiin ja ehtojen täytyttyä seuraavaan. Pääsekvenssit kattavat kymmenestä pääohjelmasta viisi kappaletta, loput ohjelmat suoriutuvat kokonaisuudessaan jokaisella ohjelmasyklillä. Jokaista pääsekvenssiä vastaa yksi boolelainen globaali muuttuja (global variable) ja ohjelma tehtiin niin, että ne voivat olla päällä vain ja ainoastaan yksi kerrallaan.

Pääsekvenssien alaisuudessa on alisekvenssejä, jotka ovat yhden ohjelman sisäisiä muuttujia (local variable). Alisekvenssit voisivat olla pääsekvenssejä, mutta ohjelmistohierarkian kannalta on parempi pitää koodin pääohjelmien lukumäärä sopivana. Pääsääntöisesti alisekvenssit ovat sellaisia ohjelmaosia, jotka suoriutuvat vain prosessin kannalta poikkeuksellisissa tilanteissa, esimerkiksi kuljetuspaineen nousu tietyn rajan yläpuolelle, viestii kuljetusputkiston tukkeutumisesta ja tällöin ohjelman tulee suorittaa tietyt toimintaselostuksessa määritellyt toiminnot.

Venttiilien ohjaus suoritettiin yhdellä pääohjelmalla, johon muut ohjelmat antoivat ohjauskäskyjä apumuuttujien avulla. Näin varmistetaan venttiilien ohjauksen estäminen tietyissä tilanteissa. Pyrin käyttämään venttiilien ohjauksessa mahdollisimman vähän asetus (set) käskyä, sillä kyseistä käskyä käyttämällä se tulee myös otta pois päältä (reset).

Ohjelma on laajuudeltaan saman kokoinen kuin mallina olleet koodit. Ohjelmoinnin kannalta kaikilla kuljettimen venttiileillä on vain kaksi asentoa, auki tai kiinni. Suurimman osan koodista toteutin tikapuukaaviota käyttäen, koska se on mielestäni selkein vaihtoehto booleilaisen ohjauksen toteuttamiseen. Kommentointi suoritettiin piireittäin. Liitteessä 1 on esiteltynä osa koodia kuljettimen paineistusvaiheesta. Kohtiin, joissa data oli analogista käytin enimmäkseen toimilohkokaaviota. Esimerkiksi kun, HMI ei tukenut ajastimiin käytettyä datatyyppeä aika (TIME), joten HMI:ltä tulevat reaalityöt

(REAL) oli muunnettava ajastimiin sopiviksi käytin FBD:tä, myös vertailu oli helpompi tehdä FBD:nä, josta esimerkki liitteessä 2. Ohjelman tarjoamalla toimilohkokaaviolla voi tehdä myös tikapuukaaviota, tästä esimerkki kuviossa 5. Toimilohkokaaviossa on erilliset toimilohkot vasemmalle ja oikealle kiskolle.



KUVIO 5 Tikapuukaavio toimilohkokaaviossa, nousevalla reunalla asetetaan (Hopper_Select) tai resetoidaan (Hopper_DeSelect) muuttuja (Hopper_Selected).

3.7 Operointipaneeli

HMI koostuu kuudesta eri näytöstä; Conveyor, Alarms, Calculated Faults, View Settings, Settings 1 ja Settings 2. Näyttöjen välillä liikutaan vasemmassa reunassa olevan valikon kautta. Ruudun oikeassa yläreunassa näkyy päivämäärä ja aika. Ohjelman valikoista löytyi valmiita kuvia prosesseissa käytettävistä yleisimmistä laitteista, kuten venttiileistä ja säiliöistä. Paine kuljettimen kuvaamiseen en kuitenkaan löytänyt sopivia kuvia, joten piirsin ne Microsoft Paint kuvankäsittelyohjelmalla.

Conveyor näyttö kuvaa prosessia ja siinä näkyvät muun muassa magneettiventtiilien ohjaustiedot ja täyttöventtiilin kiinni ja auki tiedot. Alarms ikkunassa näkyy hälytyslista, kuittauspainike ja hälytyslistan tyhjennyspainike. Calculated Faults ikkunassa näkyy kuinka monta kertaa täyttöventtiilin paineistus on ollut viallinen. Settings 1 ja Settings 2 -ikkunoissa asetetaan prosessiparametrit ja tämän vuoksi ne ovat salasanalla suojattuja. Settings View ikkunasta käyttäjä voi käydä katsomassa logiikkaan asetetut prosessiparametrit, tietämättä salasanaa. Liitteessä 3 esitetään näytöt ja niiden toiminta yksityiskohtaisesti.

4 POHDINTAA

Geneeriseen ohjelmointiin löytyy rajallinen mahdollisuus tietyillä ohjelmilla, kuten CoDeSysillä, mutta kattavaa geneeristä ohjelmointiohjelmaa PLC:den ohjelmointiin ei ole. Ideaalinen tilanne, jossa olisi vain yksi ohjelma PLC ohjelmointiin, on isojen laite- ja ohjelmistovalmistajien etujen vastaista. Suurilla valmistajilla on resursseja kehittää omia ohjelmistojaan ja niin ne tekevätkin. Siirtyminen yleiseen ohjelmointijärjestelmään voisi viedä yhtiötä kehityksessä askeleen taaksepäin. Geneerinen ohjelmointiohjelma lisäisi kilpailua automaatiolaittevalmistajien välillä, mikä on joillekin valmistajille hyödyksi ja toisille haitaksi.

Standardi IEC 61499 määrittää geneerisen ohjelmointikielen, mutta sen hyvin erilainen lähestymistapa teollisuusautomaatioon, sekä IPC pohjaisten ohjainten pakollinen käyttö vaikeuttavat sen käyttöönottoa. IPC:n ohjelmointi ja uuden ohjelmointikielen opettelu voi tuntua turhalta IEC 61131-3 standardin varjossa, myös kentälle hajautettu ohjausjärjestelmä muuttaisi merkittävästi automaatio suunnittelua, uuteen standardiin siirryttäessä ja ohjainlaitteistojen hajauttamisen seurauksena perinteistä dokumentointitapaa tulisi muuttaa.

IEC 61499 standardiin siirtymisen perimmäisin kysymys lienee seuraava: Miksi opetella jotain, mitä kukaan ei käytä? Vastaus on varmaankin, että: Kilpailu automaatiomarkkinoilla lisääntyisi. Asiakas voisi valita ohjaimen ominaisuuksien, eikä merkin mukaan. Kuinka paljon ominaisuuksia voi yhdessä ohjaimessa olla ja mitenkä ne olisivat parempia toisen valmistajan ohjaimiin verrattaessa? Yleisesti käytetty geneerinen ohjelmointiohjelma veisi myös laitteistot geneeriselle tasolle, sillä automaatiovalmistajien täytyisi tehdä laitteet lähtökohtaisesti yhteensopivaksi ohjelman kanssa.

Ohjelmointi tapahtui pääosin toimintaselostuksen pohjalta, mutta joitakin asioita katsoin käyttööni annetuista Siemensin koodeista. Tein ohjelman lähtökohtaisesti niin, että ymmärrän täysin mitä siinä tapahtuu. HMI:n toiminnot laadin pääpiirteittäin samanlaisiksi kuin räätälöidyn tuhkakuljettimen toiminnot.

Tekemääni Micro830:n ohjelmaa oli tarkoitus testata yhtiön tiloissa, mutta esteeksi tälle osoittautui uuden logiikan hidas toimitus. Ohjelma oli testattava toimitetulla logiikalla, siirsin koodin pieneen logiikkaan ja I/O:n kapasitettin yli jäävät ohjaukset osoitettiin

logiikan sisäisiin muistipaikkoihin. Testatessa karkeimmat virheet ohjelmasta ja HMI:stä tuli korjattua, mutta ohjelman toimivuus varmistuu vasta, kun se otetaan käyttöön.

LÄHTEET

CoDeSys[1]. Luettu 20.4.2013

<http://www.codesys.com/>

CoDeSys[2]. Lista tuetuista ohjaimista. Luettu 20.4.2013

<http://www.codesys.com/support-training/self-help/codesys-manual.html>

John & Tiegelkamp. IEC 61131-3 Programming Industrial Automation Systems. Second Edition. Springer 2010

Kopar[1]. Käyttöni annettuja kuvia.

Kopar[2]. Yritysesittely. Luettu 5.3.2013

<http://www.kopar.fi/>

logi.cals. logi.cals and the products. Luettu 24.3.2013

http://www.logicals.com/upload/powerdays/polen/02_RE_logi.cals_products_PowerDay_Polen_ENG.pdf

nxtControl. Luettu 24.3.2013

<http://www.nxtcontrol.com/en.html>

PLCopen. Luettu 18.3.2013

<http://www.plcopen.org/>

RTAutomation. Luettu 18.3.2013

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

Vyatkin[1]. IEC 61499 Function Blocks for Embedded and Distributed Control Systems Design. Luettu 23.3.2013

<http://homepages.engineering.auckland.ac.nz/~vyatkin/book/>

Vyatkin[2]. Status of IEC 61499 Maintenance and Further Development. Luettu 24.3.2013

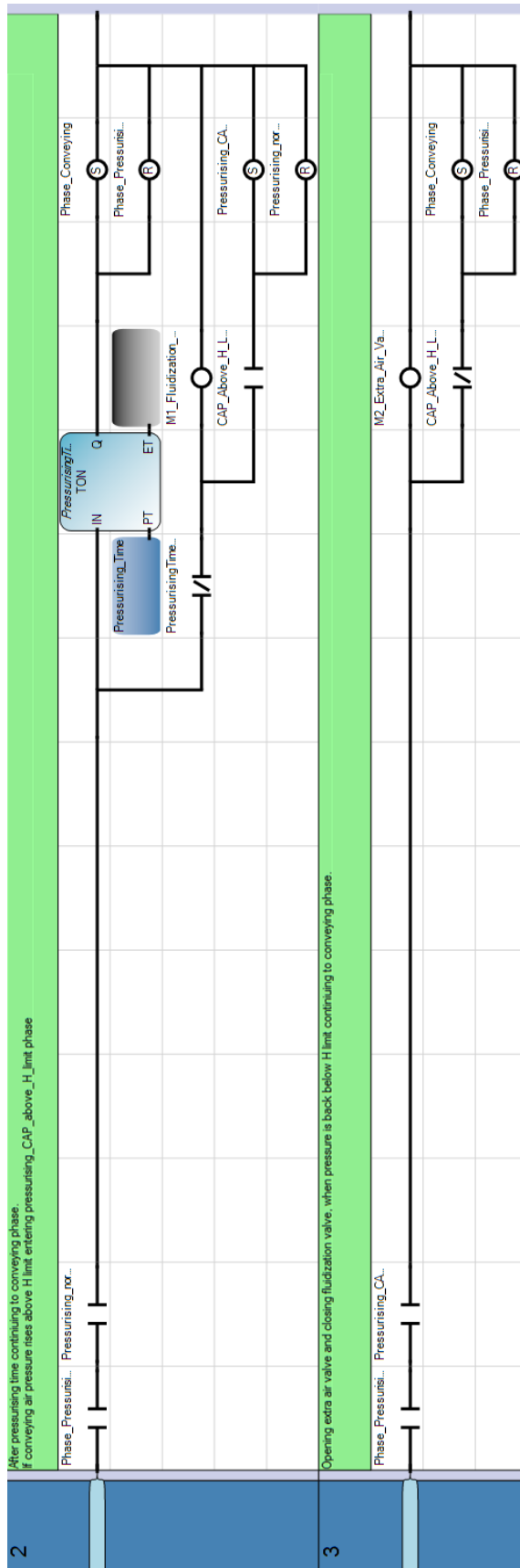
http://www.vyatkin.org/iec61499/IEC_Semantics_Maintenance.pdf

Wenbin Dai & Vyatkin. A Case Study on Migration from IEC 61131 PLC to IEC 61499 Function Block Control. Luettu 19.3.2013

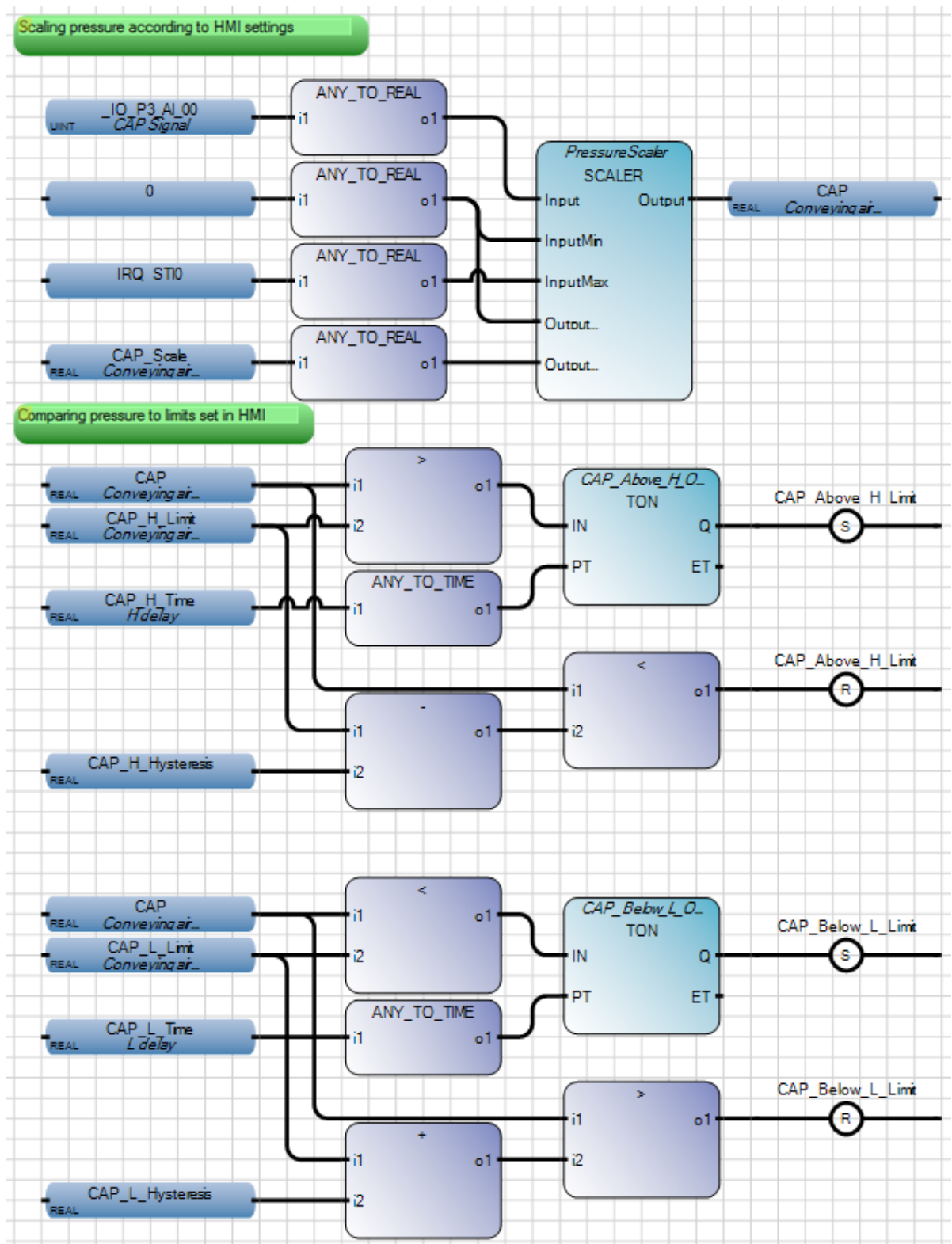
http://www.vyatkin.org/publ/indin_1%205_wd_vv.pdf

LIITTEET

Liite 1. Esimerkki kuljettimen ohjelmoinnista tikapuukaaviota käyttäen

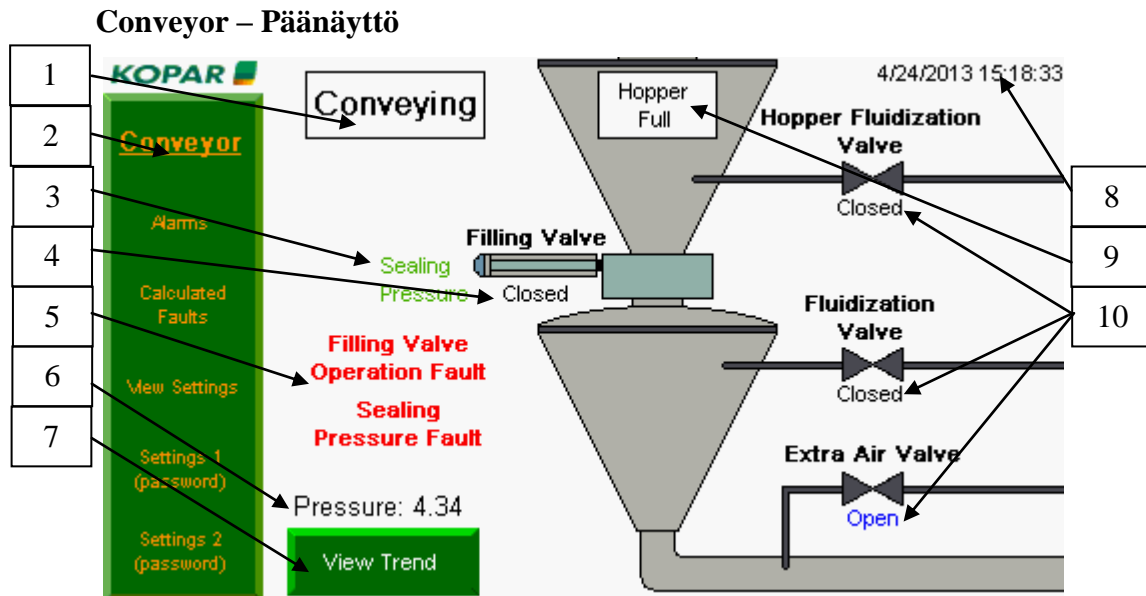


Liite 2. Esimerkki kuljettimen ohjelmoinnista toimilohkokaaviota käyttäen



Liite 3. Operointipaneelin näytöt

1(6)



1. Kuljettimen tila

Tilat ovat Filling, Purging, Conveying ja Stopped.

2. Näytön valinta

Auki oleva näyttö listassa korostettuna.

3. Täyttöventtiilin tiivistepaine

Tyhjä jos ei tiivistepainetta

4. Täyttöventtiilin kiinnirajatieto

5. Täyttöventtiilin virheet

Kumpikin virhe voi tulla erikseen

Tyhjä jos ei virhettä

6. Painetieto

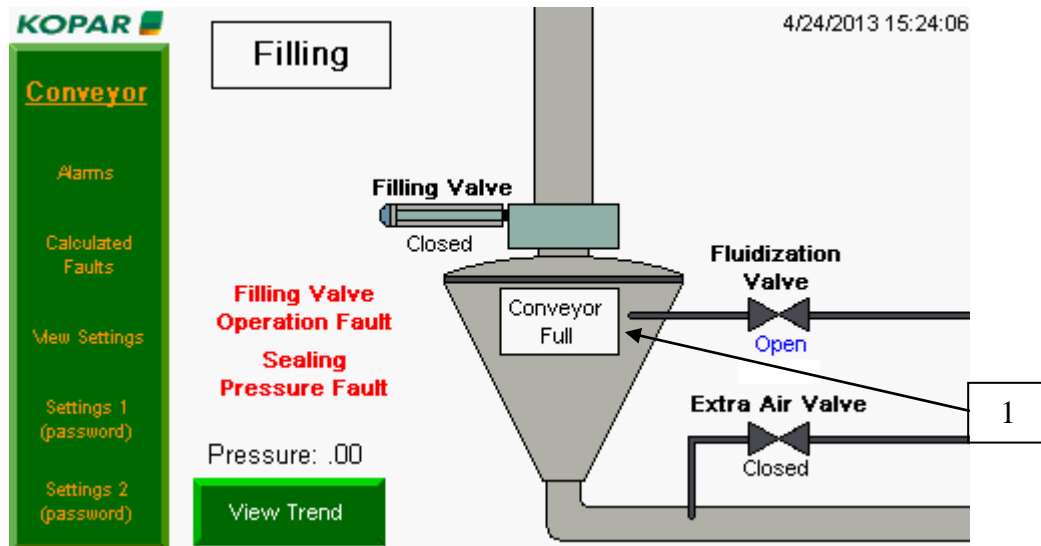
7. Avaa ja sulkee kuvaajan painesta

8. Päivämäärä ja kellonaika

9. Yläsäiliön rajapintatieto

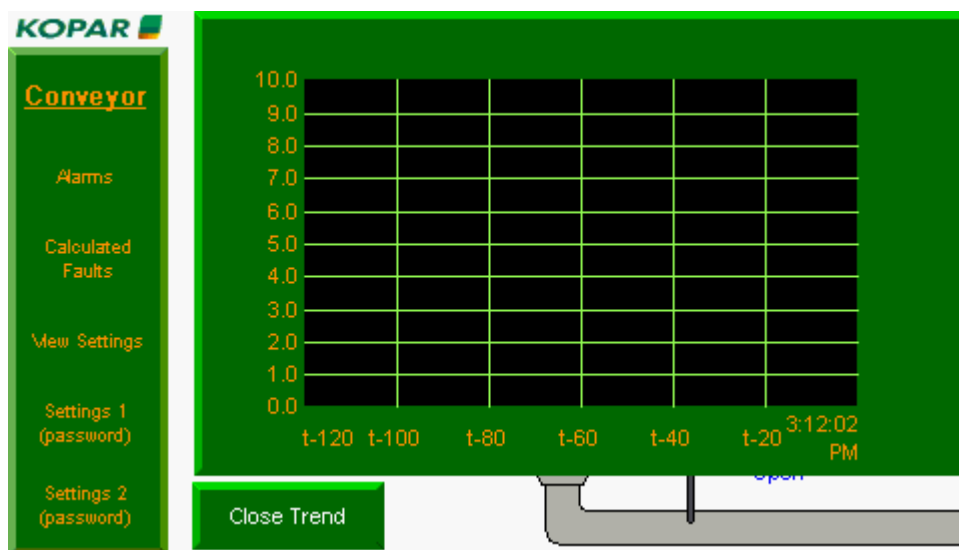
10. Magneettiventtiilien ohjaustiedot

Open auki ja Closed kiinni.



Näkymä kun yläsäiliötä ei ole

1. Säiliön rajapintatieto on siirtynyt alasäiliön kohdalle



Trendi näyttää paineen viimeisen kahden minuutin ajalta, skaalan ollessa nolasta (0) kymmeneen (10).

Alarms – Hälytykset

KOPAR 4/24/2013 15:29:34

Alarm Message	Ack?	Time	Date

Clear Ack All

1. Hälytystaulukko

Tiivistepaineen toimintahäiriö

Tiivistepaineen toimintahäiriöiden suurin sallittu määrä ylitetty

Täyttöventtiilin toimintahäiriö

Kuljetushäiriö

Pintahäiriö

2. Tyhjentää hälytyslistan

3. Hälytysten kuittaus

Calculated Faults – Virhelaskelmat

KOPAR 4/24/2013 15:31:39

Sealing faults: 0

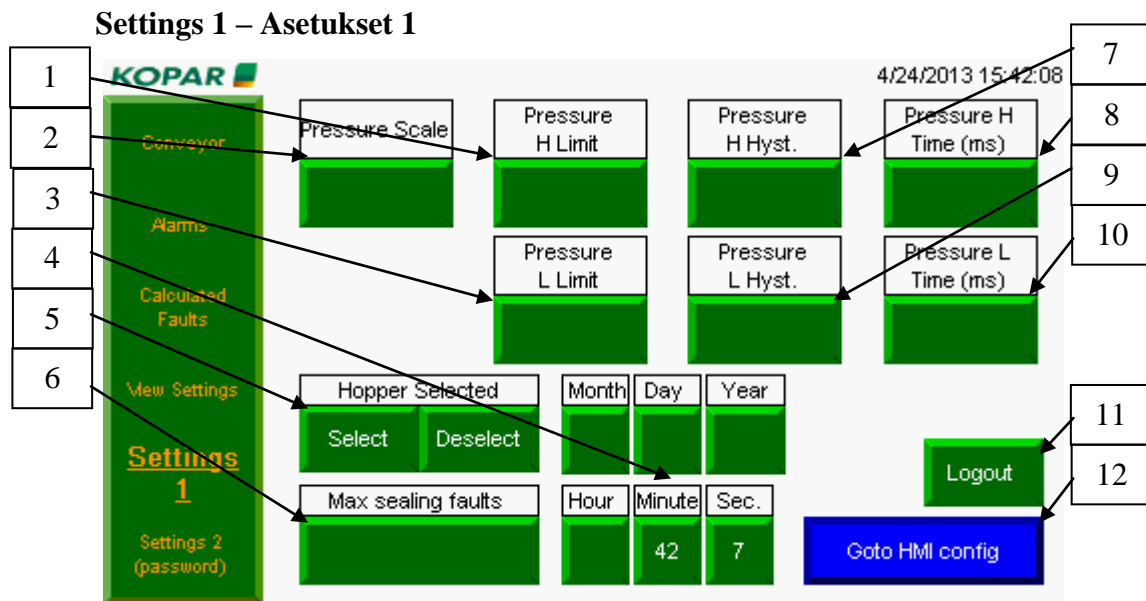
Pressure above H limit: 0

Reset calculations

1. Tiivistepaineen toimintahäiriöiden määrä

2. Paineen ylärajan ylitysten määrä

3. Laskureiden nollaus

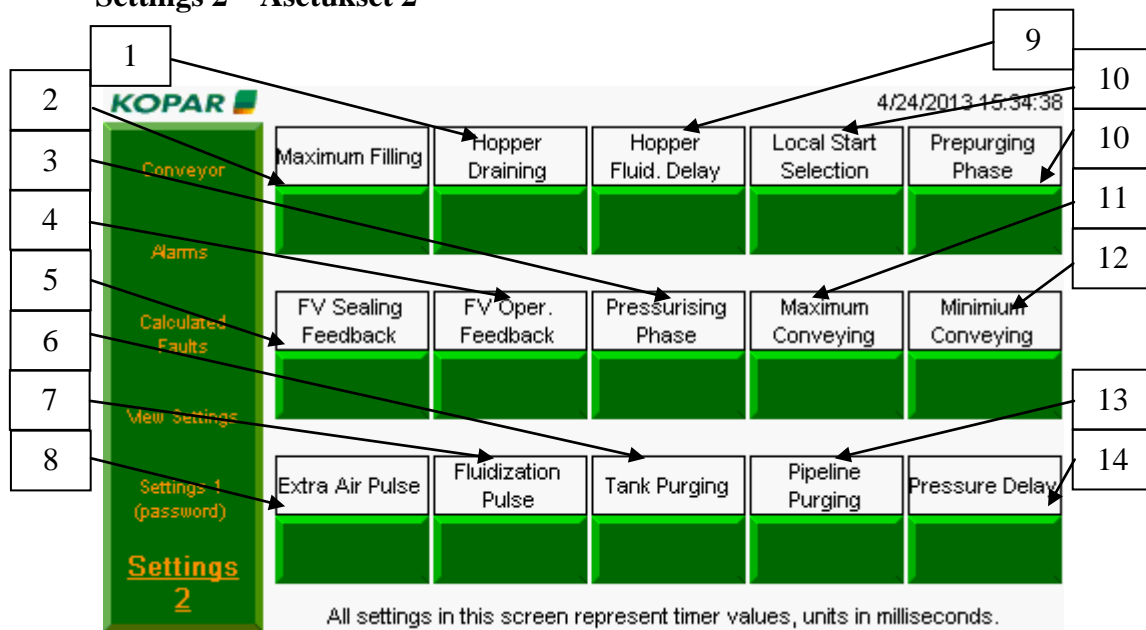


1. Paineen yläraja
2. Paineen skaala
3. Paineen alaraja
4. Ajan asettaminen
5. Yläsäiliön valinta
6. Suurin sallittu tiivistehäiriöiden määrä
7. Paineen hystereesi ylärajalla
8. Viive, jonka jälkeen paineen ylärajatieto asettuu, mikäli paine vielä on ylärajalla.
9. Paineen hystereesi alarajalla
10. Viive, jonka jälkeen painen alarajatieto asettuu, mikäli paine vielä on alarajalla.
11. Käyttäjän uloskirjaus

Tultaessa Settings 1 tai Settings 2 –valikkoihin HMI kysyy käyttäjätunnusta ja salasanaa. Tämän napin avulla käyttäjä uloskirjautuu ja ohjelma palaa Conveyor ikkunaan.

12. Käynti HMI:n asetuksiin

Settings 2 – Asetukset 2



1. Yläsäiliön tyhjennysaika
2. Suurin sallittu täyttöaika
 - Jos rajapinta-anturi ei reagoi täyttövaiheessa ennen kun aika on kulunut loppuun, niin ohjelma jatkaa seuraavaan vaiheeseen.
3. Paineistus aika
4. Täyttöventtiiliin takaisinkytkentä aika
5. Täyttöventtiilin tiivistepaineen takaisinkytkentä aika
6. Säiliön puhdistusaika
7. Fluidisaatioventtiili auki, lisäilmaventtiili kiinni
8. Lisäilmaventtiili auki, fluidisaatioventtiili kiinni
9. Viive yläsäiliön fluidisoinnissa täyttövaiheessa
10. Käynnistyskytkimen valinta-aika

Jos kytkintä pidetään käynnistysasennossa tätä aikaa kauemmin siirrytään täyttöön.

11. Suurin sallittu siirtoaika
12. Pienin sallittu siirtoaika
13. Putkiston loppuhuuhdelun aika
14. Paineistusviive kuljetussyklin lopussa

View Settings – Asetukset

KOPAR 4/24/2013 20:30:23

Conveyor	Pressure Scale: .00	Pressure H Limit: .00
Alarms	Pressure L Limit: .00	Pressure H Hyst.: .00
Calculated Faults	Pressure L Hyst.: .00	Pressure H Time: 0ms
View Settings	Pressure L Time: 0ms	Max sealing faults: 0
Settings 1 (password)	Hopper Selected	Hopper Fluid. Delay: 0ms
Settings 2 (password)	Hopper Draining: 0ms	Local Start Selection: 0ms
	Maximum Filling: 0ms	FV Sealing Feedback: 0ms
	FV Oper. Feedback: 0ms	Pressurising Phase: 0ms
	Prepurging Phase: 0ms	Minimum Conveying: 0ms
	Maximum Conveying: 0ms	Fluidization Pulse: 0ms
	Extra Air Pulse: 0ms	Pipeline Purging: 0ms
	Tank Purging: 0ms	Pressure Delay: 0ms

Ikkunassa näytetään kuljettimen asetukset, niin ettei niitä voi muuttaa.