OULUN SEUDUN
AMMATTIKORKEAKOULU

Pekka Tervo

HAPPIKOMPRESSORIN OHJAUKSEN MODERNISOINTI

HAPPIKOMPRESSORIN OHJAUKSEN MODERNISOINTI

Pekka Tervo
Opinnäytetyö
Kevät 2012
Tekniikan yksikkö, automaatiotekniikan osasto
Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Automaatiotekniikka, projektointi

Tekijä: Pekka Tervo
Opinnäytetyön nimi: Happikompressorin ohjauksen modernisointi
Työn ohjaaja: Timo Heikkinen
Työn valmistumislukukausi ja -vuosi: Kevät 2012 Sivumäärä: *53* + *7*

Työssä perehdyttiin ABB:n 800xA-järjestelmään. Esimerkkiprojektina tehtiin Kemira Chemicals Oy Oulun tehtaalla sijaitsevan happikompressorin ohjauksen modernisointi ja kääntäminen Allen&Bradley RsLogix 5:stä ABB:n järjestelmään. Työ toteutettiin syksyn 2011 ja kevään 2012 aikana samaan aikaan Vetyperoksidi 2- ja Muurahaishappotehtaan automaatiojärjestelmän päivityksen kanssa. Toimeksiantaja työlle oli ABB Oy Service Prosessiautomaatio.

Työn tavoitteena oli integroida happikompressorin ohjaus tehtaan automaatiojärjestelmään samalla ylläpitäen vanhan ohjauksen toiminta uudessa 800xA-järjestelmässä. Työssä käytettiin hyväksi Etteplan Oyj:n toimittamaa toiminnanselostusta ja vanhaa Allen&Bradley RsLogix 5:llä toteutettua ohjausta sekä yleistä materiaalia kompressoreista ja paineensäädöstä.

Työssä saatiin selkeä kuva ABB:n 800xA-järjestelmästä ja sen työskentelyvirrasta ohjelmoijan näkökulmasta. Työn aikana kasvoi ymmärrys kompressoreissa tapahtuvista ilmiöistä ja opittiin kaasun paineistamisessa huomioon otettavia asioita. Työn laajuus rajattiin tehdashyväksyntätestiin, joka toteutettiin asiakkaan ja suunnittelutoimiston henkilökunnan kanssa. Tehdashyväksyntätestin jälkeen ohjelma otettiin käyttöön.

Asiasanat: Kompressori, automaatiojärjestelmä, teollisuusautomaatio, paineensäätö.

ALKULAUSE

Insinöörityön tilaajana oli ABB Oy Service Prosessiteollisuus, jossa työn valvojana toimi Kimmo Peltola. Oulun seudun ammattikorkeakoulusta työn ohjaajana toimi Timo Heikkinen. Haluan kiittää heitä molempia insinöörityön loppuun saattamisen avustamisesta.

Heidän lisäksi haluan kiittää uudistusprojektin pääsuunnittelijoita Aulis Mursua ja Markku Lahtelaa sekä Vesa Urpelaista kärsivällisestä auttamisesta työn kuluessa.

Oulussa 27.4.2012

Pekka Tervo

SISÄLTÖ

TIIVISTELMÄ 3
ALKULAUSE 4
SISÄLTÖ 5
1 JOHDANTO 7
2 KEMIRA CHEMICALS OY OULUN TEHTAIDEN JÄRJESTELMÄUUDISTUS 8
2.1 Vetyperoksidin valmistus 8
2.2 Automaatiojärjestelmän uudistus 8
3 HAPPIKOMPRESSORI 10
3.1 Happikompressorin toiminta 10
3.2 Happikompressorin ohjaus 11
3.3 Allen&Bradley RsLogix 5 12
4 ABB 800xA-JÄRJESTELMÄ 14
4.1 Topologia 15
4.2 Engineering workplace 17
4.3 Operaattoriympäristö 18
4.4 Kirjastot 19
4.5 Control Builder 20
4.5.1 Strukturoitu ohjelmointiperiaate 20
4.5.2 Ohjelmointikielet 22
4.5.3 Muuttujat 26
4.5.4 Datatyypit 29
4.5.5 Bulkdata manager 32
4.5.6 Laitteistokonfiguraatio 33
4.6 Graphics Builder 34
4.7 Laitteisto 34
5 KÄÄNNÖSTYÖ 36
5.1 Toiminnankuvaus 36
5.1.1 Paineensäätö ja pumppaussuoja 36
5.1.2 Säätötavan valinta 37
5.1.3 Öljyvoitelu 39
5.1.4 Tiivistekaasu 39
5.2 Toteutus 40
5.3 Pumppauskäyrän muodostaminen 40

5.4 Tiedonsiirto, historiankeruu ja suoritusvälin valinta 41
6 VALVOMOKUVAT 42
7 TEHDASHYVÄKSYNTÄTESTI 46
8 YHTEENVETO.....48
LÄHTEET.....50
LIITTEET 53
Liite 1 Paineensäätö .....
Liite 2 Pumppaussuoja.....

1 JOHDANTO

ABB Oy Service Prosessiautomaatio ja Kemira Oy Oulun tehtaat antoivat mahdollisuuden toteuttaa opinnäytetyön, joka käsitti Atlas&Copcon valmistaman happikompressorin ohjauksen siirtämisen Allen&Bradley RsLogix 5 -perustaiselta ohjaukselta ABB:n 800xA-järjestelmään.

Vanhempi pääasiallinen automaatijärjestelmä koostui ABB:n Master Piece-sarjasta, johon osa happikompressorin IO-tiedoista oli tuotu etävalvontaa varten. Happikompressoria oli ohjattu happikompressorin läheisyydessä sijaitsevasta paikallisohjauskotelosta. Myös erillinen näyttö sijaitsi happikompressorin läheisyydessä. Työssä oli tarkoitus tehdä kolmesta neljään operointinäyttöä 800xA-järjestelmään.

Uuden ohjauksen myötä happikompressoria tulisi voida ohjata etänä vetyperoksiditehtaan valvomosta, ja uusi ohjaus pyritään toteuttamaan siten, että se eroaisi mahdollisimman vähän muista prosessialueista. Käännöstyö tehdään pääasiassa Etteplan Oy:n toimittaman toiminnankuvauksen mukaan ja tarvittaessa tukeudutaan vanhaan Allen&Bradley RsLogix 5:n lähdekoodiin. Lopullinen hyväksyntä työlle saadaan tehdashyväksyntätesteissä, jotka toteutettiin Kemiran henkilöstön kanssa.

2 KEMIRA CHEMICALS OY OULUN TEHTAIDEN JÄRJESTELMÄUUDISTUS

Kemira Chemicals Oy Oulun tehtaiden vetyperoksidin valmistusprosessia ohjasi entuudestaan ABB:n Master Piece -automaatijärjestelmä. Järjestelmäuudistuksessa oli tarkoitus päivittää ohjaus ABB:n 800xA-järjestelmään. Toimitetut uudet prosessiasemat ovat ABB:n AC 800M -tuoteryhmään kuuluvia 800xA-järjestelmän oletusprosessiasemia.

2.1 Vetyperoksidin valmistus

Vetyperoksidi on kirkas neste, jonka viskositeetti on hieman vettä korkeampi. Vetyperoksidin kemiallinen kaava on H_2O_2, jossa on kaksi osaa vetyä ja kaksi osaa happea. Vetyperoksidi on tehokas valkaisuaine, ja sitä käytetään siinä tarkoituksessa muun muassa selluteollisuudessa. Muita käyttökohteita on muun muassa vedenpuhdistusprosessissa ja kosmetiikkateollisuudessa. Maailmanlaajuisesti vetyperoksidia valmistettiin 2,2 miljoonaa tonnia vuonna 2006 ja trendi on nouseva.

Vetyperoksidi ei ole palava aine alle 85 %:n pitoisuutena, mutta voimakkaana hapetusaineena se aiheuttaa palamisvaaran palavan materiaalin kanssa. Vetyperoksidi on luontoystävällinen aine, koska vetyperoksidin hajoaminen muodostaa vettä ja happea.

2.2 Automaatiojärjestelmän uudistus

Järjestelmäuudistusprojektissa automaatio-ohjelmat käännettiin ABB:n Advant-pohjaisesta koodista 800xA-pohjaiseen. Käännöstyö toteutettiin toimilohkokielellä, lukuunottamatta sekvenssejä, jotka toteutettiin struktuuritekstipohjaisena. Valvomokuvat päivitettiin Visual Basic -pohjaisista kuvista PG2-kuviin ABB:n tarjoamalla Graphics Builder -työkalulla. MB-sarjan prosessiasemat vaihdetaan AC 800M -perheeseen kuuluviin prosessiasemiin.

Happikompressorin ohjauksen päivitys erosi paljon muurahaishappotehtaan ja vetyperoksiditehtaan päivityksestä, sillä happikompressorin ohjaus oli toteutettu eri valmistajan tarjoamalla ratkaisulla ja eri kielellä kuin ABB:n tarjoama ratkaisu.

3 HAPPIKOMPRESSORI

Happikompressori on Atlas&Copcon toimittama. Kompressorin ohjauslogiikka on toteutettu Rockwell Automationin Allen&Bradley RsLogix 5 -sarjalla. Happikompressori on toiminut moitteetta vuosia, mutta sen automaatio on ollut lähes täysin erillään tehtaan pääasiallisesta automaatiojärjestelmä. Vain muutamia mittaustietoja on tuotu ABB:n järjestelmään.

Syynä happikompressorin ohjauksen päivitykseen oli sama kuin muunkin tehtaan päivityksessä eli modernisointi ja varaosien saatavuuden pieneneminen. Tarkoitus oli myös muuttaa ohjaustapaa. Päivityksen jälkeen happikompressoria tulisi voida ohjata etänä vetyperoksiditehtaan valvomosta sen sijaan, että sitä ohjattaisiin happikompressorin läheisyydessä olevasta paikallisohjauskotelosta.

3.1 Happikompressorin toiminta

Happikompressorin tehtävä on paineistaa puhdasta happikaasua. Happikaasua käytetään vetyperoksiditehtaan hapetus-osaprosessissa. Happikompressorin toiminnan kannalta on erityisesti otettava huomioon puhtaan hapen palamisvaara. Kompressori on kaksivaiheinen kineettinen radiaalikompressori. (24.)

Kummankin vaiheen jälkeen happikaasua jäähdytetään. Kun kaasua paineistetaan, muodostuu paljon lämpöenergiaa, joten voi tuntua erikoiselta jäähdyttää paineistettua kaasua, jolloin osa tehdystä työstä menee hukkaan. Kun paineistettua happea viedään hapetusprosessiin putkilinjastoa pitkin, se jäähtyy ja syntyy kosteutta. Välijäähdytyksellä pyritään siis poistamaan mahdollisimman suuri osa kosteudesta. Toissijainen tehtävä välijäähdytyksellä on alentaa happikaasun lämpötilaa, jotta sitä voitaisiin edelleen käsitellä. (21.)

Jäähdytys toteutetaan vaiheiden jälkeen olevilla painesäiliöillä. Painesäiliössä muodostuneet nestepisarat pääsevät valumaan pohjalle. Yleisesti kompressoreissa painesäiliö toimii myös pienenä paineentasaajana sekä -varaajana. Jos käsiteltävän aineen pyynti vaihtelee voimakkaasti, voidaan painesäiliöiden avulla tasata äkkinäisiä paineenvaihteluita, sillä niissä virtaus on rauhallisempaa. Käsiteltävän happikompressorin painesäiliöt ovat kuitenkin sen verran pienet, ettei niillä ole suurta merkistystä säädön kannalta. (21.)

Yksi prosessiosa-alue happikompressorissa oli öljyvoitelu. Voitelun keskeisemmät tehtävät ovat seuraavat:

- pienentää kitkaa, kulumista ja energiantarvetta
- kuljettaa kitkasta ja puristusprosessista aiheutuvaa lämpöä pois kompressorin liikkuvista osista
- parantaa hyötysuhdetta
- suojata ruosteelta.

Tiivistekaasun tehtävä on pitää öljy ja happikaasu erillään toisistaan palovaaran pienentämiseksi sekä estää epäpuhtauksien pääsy happikaasuun. Öljykierrätys synnyttää öljysumua, joka on vaarallista ympäristölle ja ihmisille. Tästä syystä öljysumunpoiston tulee käynnistyä aina, kun öljykierrätys on käynnissä. (21.)

3.2 Happikompressorin ohjaus

Ohjauksen tehtävä on ennen kaikkea pitää prosessi turvallisella pumppausalueella. Kun happikompressori toimii turvallisessa alueella, ohjauksen tulee ylläpitää loppupainetta astellussa asetusarvossa. Hapen pyyntimäärä asetellaan hapetusprosessissa virtaussäätöpiireihin, jotka ei kuulu happikompressorin käännöstyöhön. Happikompressorin

moottorinohjauksen relepohjaisuudesta huolimatta uudesta ohjauksesta pyrittiin tekemään yhteneväinen koko tehtaan kanssa.

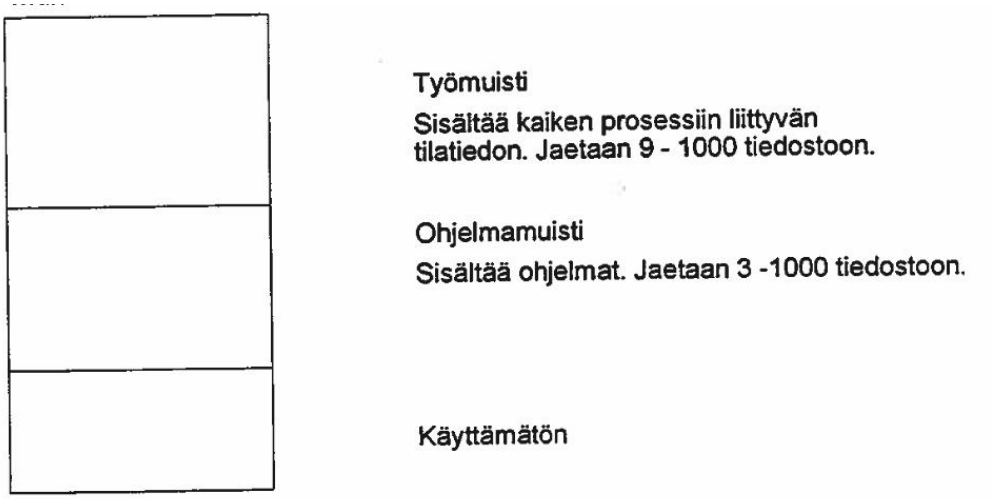
3.3 Allen&Bradley RsLogix 5

Vanha ohjaus oli toteutettu Allen&Bradleyn PLC-5 -perheeseen kuuluvalla ratkaisulla. Ohjelmointikielenä oli käytetty tikapuukieltä. Logiikka on jo verrattaen vanhaa ja sitä on käytössä enää harvoin. PLC-5 -perheessä keskusyksikkö, muisti ja liitäntäpiirit on kaikki rakennettu yhteen yksikköön. Keskusyksikkö käsittelee prosessin tilatiedot ohjelman mukaisesti ja ohjaa siten prosessia. Muisti jakaantuu pääasiassa kahteen alueeseen: työmuistiin ja ohjelmamuistiin. Nämä molemmat voidaan jakaa 1000 tiedostoon.

Työmuistissa on tulojen ja lähtöjen tilatiedot, ajastimien ja laskureiden asetus- ja oloarvot, apumuisti ja käsiteltävät lukuarvot, eli se sisältää kaiken ohjattavaan prosessiin liittyvän tilatiedon. Ohjelmamuistissa on käyttäjän tekemä ohjelma, joka voi jakaantua 1000 erilliseen osa-ohjelmaan. Liitäntäpiirit välittävät tiedot tulokorteilta tulorekisteriin ja lähtörekisterintilat lähtökorteille, kommunikoivat hajautetuille kehikoille ja toisille logiikoille, tietokoneille ja ohjelmointilaitteille PLC-väylän kautta.

Prosessorin työkierto jakaantuu kahteen osaan: I/O-kiertoon ja ohjelmakiertoon. I/O-kierrossa prosessori lukee tulokorteille tulevat tiedot ja kirjoittaa lähtökorteille menevät tiedot. Tämän jälkeen suoritetaan ohjelmakierto, joka suorittaa käskyt yksi kerrallaan ja tallettaa lähtötiedot välimuistiin seuraavaa I/O-kiertoa varten. Työkierto tulee ottaa huomioon ohjelmoitaessa.

PLC-5 -prosessorien muisti koostuu pienistä muistiyksiköistä, eli biteistä. Bitit yleensä ryhmitellään 8 tai 16 bitin sanoiksi. Tavuja PLC-5 perheessä ei käytetä. Koko muisti voidaan ajatella yhdeksi isoksi alueeksi, joka jaetaan käyttötarkoituksen mukaisesti eri osa-alueisiin seuraavan kuvan 1 tapaan.



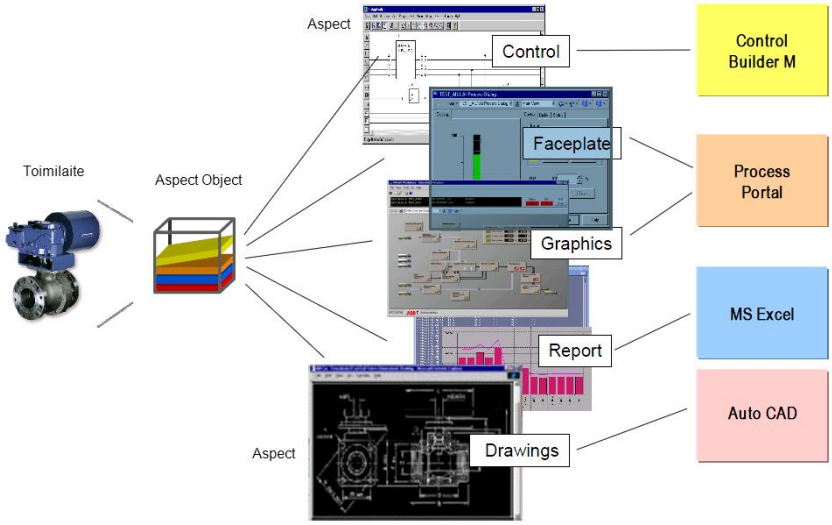
KUVA 1. Allen&Bradleyn RsLogix 5:n muistialue (22)

RsLogixin ohjelmoinnin periaate on piirikaaviomuotoinen tikapuukaavio. Perusmuodossa ohjelmarivi muodostuu sen vasemmassa päässä olevista ehdoista toiminnalle ja sen oikeassa päässä olevista toimintakäskyistä. RsLogix 5:ssä on niin sanottu käskykanta, jossa käskyt vastaavat toimilohkoja ABB:n 800xA-järjestelmän ohjelmointityökalussa. (22.)

4 ABB 800xA-JÄRJESTELMÄ

ABB:n 800xA-järjestelmä on yksi ABB:n tarjoamista automaatiojärjestelmistä. 800xA-järjestelmä käsittää hajautetun ohjausjärjestelmän lisäksi paljon muuta. Järjestelmä toimii pohjana, jonka päälle rakennetaan prosessin ohjaamisen ja ylläpidon vaatimat osat. Standardeihin perustuva järjestelmä mahdollistaa sen liittämisen useisiin muihin ympäristöihin, oli se sitten eri automaatiojärjestelmä tai ERP-tason ohjelmisto. 800xA-järjestelmä tukee yleisimpiä kenttäväyläprotokollia, joten sitä voidaan käyttää laajasti eri hankkeissa.

800xA-järjestelmä perustuu Aspect Object -arkkitehtuuriin, joka tekee järjestelmästä yhtenäisen ja informaatio-keskeisen. Aspect object voi kuvata esimerkiksi fyysistä toimilaitetta, tuotetta tai tilausta. Aspect object on kyseessä olevan asian pakkaus, joka sisältää tietoa siitä. Eri tietoja kutsutaan aspekteiksi. Aspekteja voi olla esimerkiksi tilausmääritelmä, resepti tai räjäytyskuva (kuva 2). Aspektit ovat pääasiassa ABB:n järjestelmän luomia, mutta aspekteihin voi myös liittää muiden sovelluksien tuottamaa informaatiota tai muihin ohjelmiin vietävää informaatiota, kuten Excel-taulukkoja, CAD-kuvia tai suoraa valvontakameran kuvaa. (1; 2.)

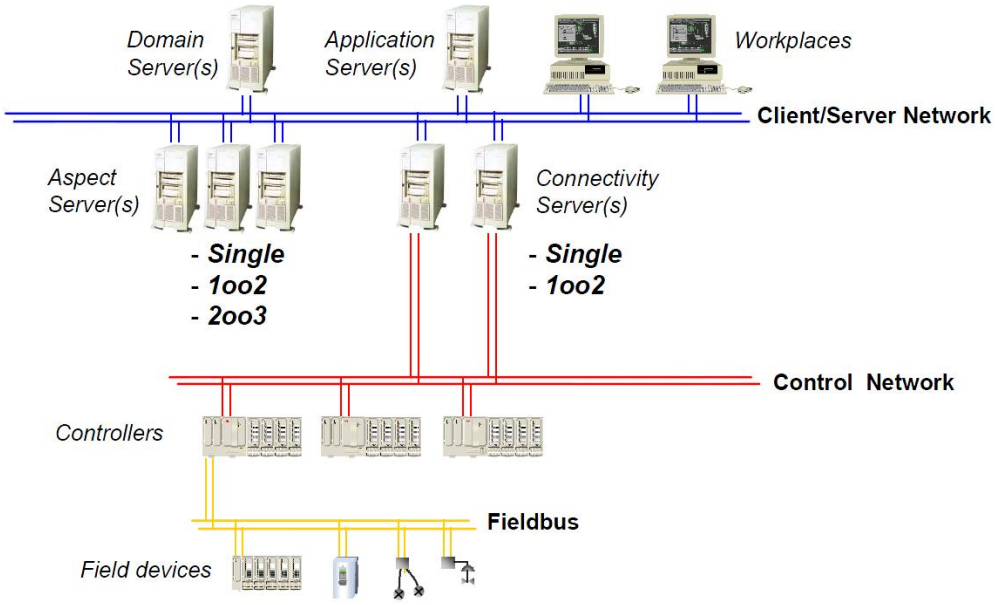


KUVA 2. Aspect Object-esimerkki (1)

4.1 Topologia

800xA-järjestelmän topologia koostuu tietokoneiden ja laitteiden välisestä kommunikaatiosta eri tiedonsiirtoprotokollilla. Seuraavalla sivulla on pelkistetty kuva topologiasta (kuva 3). Workplace on joko itsenäinen työasema isossa sovellutuksessa tai työaseman ja serverin yhdistelmä pienessä sovellutuksessa.

Erittäinen pieni sovellutus on mahdollista tehdä vain yhdellä PC:llä, jonka sisällä jokainen tarvittava serveri pyörii.



KUVA 3. 800xA-järjestelmän arkkitehtuuri (3)

Aspekti-serveri on 800xA-järjestelmän sydän, joka ylläpitää aspekti-tietokantaa ja hallinnoi objekteja. Aspekti-serverin täytyy olla aina tavoitettavissa kaikille verkon solmukohdille. Pienissä sovellutuksessa Aspekti-serverin voi yhdistää muihin servereihin.

Connectivity-serveri mahdollistaa tietoliikenteen prosessiasemille ja muihin datalähteisiin tietoverkossa. Connectivity-serveri hallinnoi muun muassa OPC-tiedonsiirtoa. Yhteen Conncetivity-serveriin voi maksimissaan kytkeä 24 prosessiasemaa.

Domain-serveriä ei välttämättä tarvita pienessä järjestelmässä, mutta silloin solmukohtia ja käyttäjiä hallinnoidaan Windowsin Workgroup-työkalulla. Kuvan 3 tapauksessa käyttäjäoikeuksia ja verkkoa hallinnoi itsenäinen Domain-serveri. Application-serveri voi pitää sisällään monentyyppisiä palveluita, kuten resurssien hallintaa ja historiankeruuta.

Työaseman ja servereiden toiminnan voi kuvata esimerkillä, jossa operointinäyttö avataan työasemalla. Kyseinen operointinäyttö on aspekti, joka haetaan Aspekti-serveriltä. Operointinäytöltä avataan venttiilin

Faceplate, joka on myös aspekti-serveriltä haettava venttiiliobjektin aspekti. Prosessiasemalta saadaan venttiilin dynaamisen auki/kiinni-osoituksen tiedot, jotka connectivity-serveri noutaa. (1; 3.)

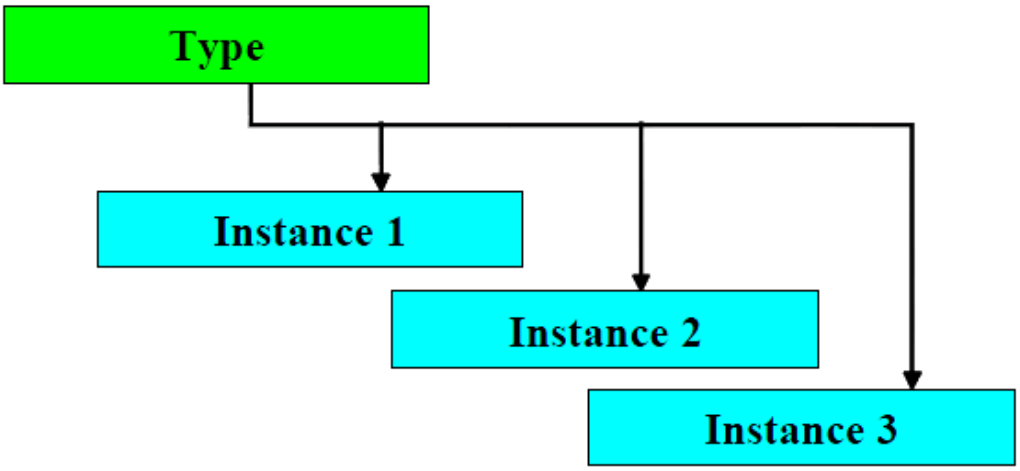
4.2 Engineering workplace

Engineerin Workplace on työkalu tai -ympäristö, jolla konfiguroidaan 800xA-järjestelmää. Sillä voidaan luoda ja hallinnoida aspect objecteja ja niiden aspekteja. Engineering Workplace koostuu useista puurakennestruktuureista, joista käytetyimpiä ovat Control-struktuuri ja Functional struktuuri.

Control-struktuuri rakentuu Control Builderilla tehtyjen ohjaussovelluksien mukaan. Se sisältää laitteistokonfiguraation sekä sovellus- ja ohjelmapuurakenteen. Käytännössä se on prosessinohjauksen rakenne. Functional struktuuri kuvaa tehtaan toiminnallisuutta. Tavan mukaista on rakentaa Functional struktuuri prosessiosa-alueiden mukaan.

Muita harvemmin käytettyjä mutta tarpeellisia struktuureja ovat Library struktuuri ja Object type struktuuri. Library struktuurissa sijaitsevat kaikki järjestelmässä olevat kirjastot ja mallipohjat, kuten hälytys- ja tapahtumalistakonfiguraatiot. Library struktuurin tehtävä on järjestää uudelleenkäytettäviä kokonaisuuksia.

Library struktuurissa olevia objektityyppejä pidetään Object type struktuurissa. Lähes kaikki aspect objectit ovat instansseja objektityypeistä (kuva 4).



KUVA 4. Tyypin ja instanssin suhde.

Tarkoitus on, että peruspiirejä ja sovelluksia voidaan uudelleenkäyttää. Esimerkiksi venttiilin ohjaukseen ei tarvitse aina luoda uutta ratkaisua,vaan käytetään jo olemassaolevaa ratkaisua. Yleisempi tapa on käyttää jo valmiiksi kirjastossa olevia moduuleja. Tarpeen vaatiessa voi tehdä omia moduuliratkaisuja. Tyyppipiirejä luotaessa on otettava huomioon, että kaikki muutokset, jotka tehdään moduuliin, vaikuttavat kaikkiin instansseihin. (4.)

4.3 Operaattoriympäristö

Operaattoriympäristö (Operator workplace) on osa Plant Explorer-työkalua, josta myös Engineering workplace avautuu. Operaattoriympäristön oikeudet rajataan yleensä sellaisiksi, ettei operaattori pääse muokkamaan järjestelmää. Ympäristön sydämenä on Operator workplace, jossa näytetään Graphics Builderilla tehdyt operointinäytöt sekä muuta informaatiota, kuten hälytys- ja tapahtumalistat. Operator Workplace luodaan workplace-struktuuriin, jonne myös hälytyslistat pyritään tekemään. Hälytyslistat voidaan toteuttaa monella tavalla. Hälytyksiä voidaan tarvittaessa jakaa useille eri hälytyslistoille, vaikkapa prosessin osa-alueiden mukaan.

4.4 Kirjastot

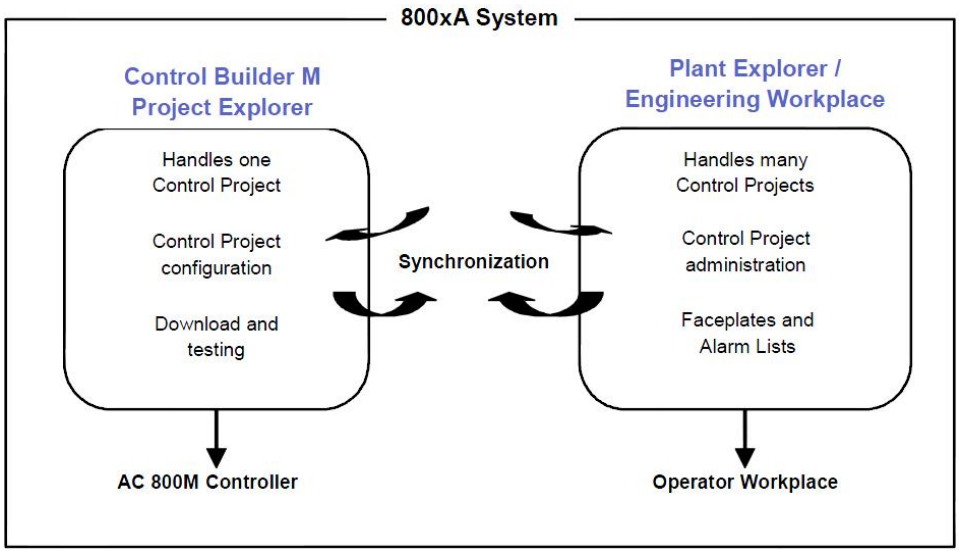
Kuten edellissä luvussa on mainittu, kirjastojen tehtävä on tarjota tyyppiratkaisuja prosessinohjaukseen. Kirjastot sisältävät kolmenlaisia tyyppimääritelmiä: datatyyppejä − kuten bool, real ja string − control moduletyyppejä ja toimilohkotyyppejä. 800xA-järjestelmä sisältää useita standardikirjastoja, mutta haluttaessa on mahdollista käyttää eri teollisuusaloille tarkoitettuja erikoiskirjastoja. Happikompressorin ohjaussovelluksessa käytettiin standardikirjastojen lisäksi Pulp&Paper-kirjastoa sekä itse tehtyä tyyppikirjastoa Kemira_TC, joka pohjautuu Pulp&Paper-kirjastoon. (5.)

Ohjelmissa käytettävien kirjastojen lisäksi on laitteistokirjastot, jotka sisältävät HWD-tiedostoja fyysisille laitteille. On myös mahdollista ja usein pakollista luoda omia laitteistokirjastoja. 800xA-järjestelmässä ei käytetä suoraan yleisimmin käytettyjä GSD-tiedostoja. GSD-tiedosto kertoo, kuinka kommunoida laitteen kanssa. GSD-tiedostot ovat niin sanottuja toimilaitteiden datalehtiä. GSD-tiedostoa tarvitaan, jotta toimilaitetta voidaan ohjata eri järjestelmissä profibusväylän kautta. 800xA-järjestelmässä GSD-tiedostot käännetään HWD-tiedostoiksi. Kääntäminen tehdään luomalla uusi laitekirjasto, johon lisätään laite GSD-tiedostolla. Tässä vaiheessa järjestelmä kääntää tiedoston HWD-tiedostoksi ja ohjelmoijan tehtäväksi jää jakaa I/O:t tarvittaviin paikkoihin. (19; 20.)

Pulp&Paper-kirjasto on suunniteltu sellu- ja paperiteollisuutta ajatellen, mutta sen tyyppiratkaisut sopivat siitä huolimatta usealla eri teollisuuden alalle, kuten tässä tapauksessa kemianteollisuudelle. Pulp&Paper tarjoaa muun muassa adaptiivisia säätimiä, automaattisesti virittyviä PID-säätimiä, taajuusmuuttajaohjauksia, sekvenssiratkaisuja ja venttiilinohjauksia. Kirjasto tarjoaa myös vastaaville toimilohkoille muun muassa prosessinäyttöelementit ja Faceplatet. (7; 8.)

4.5 Control Builder

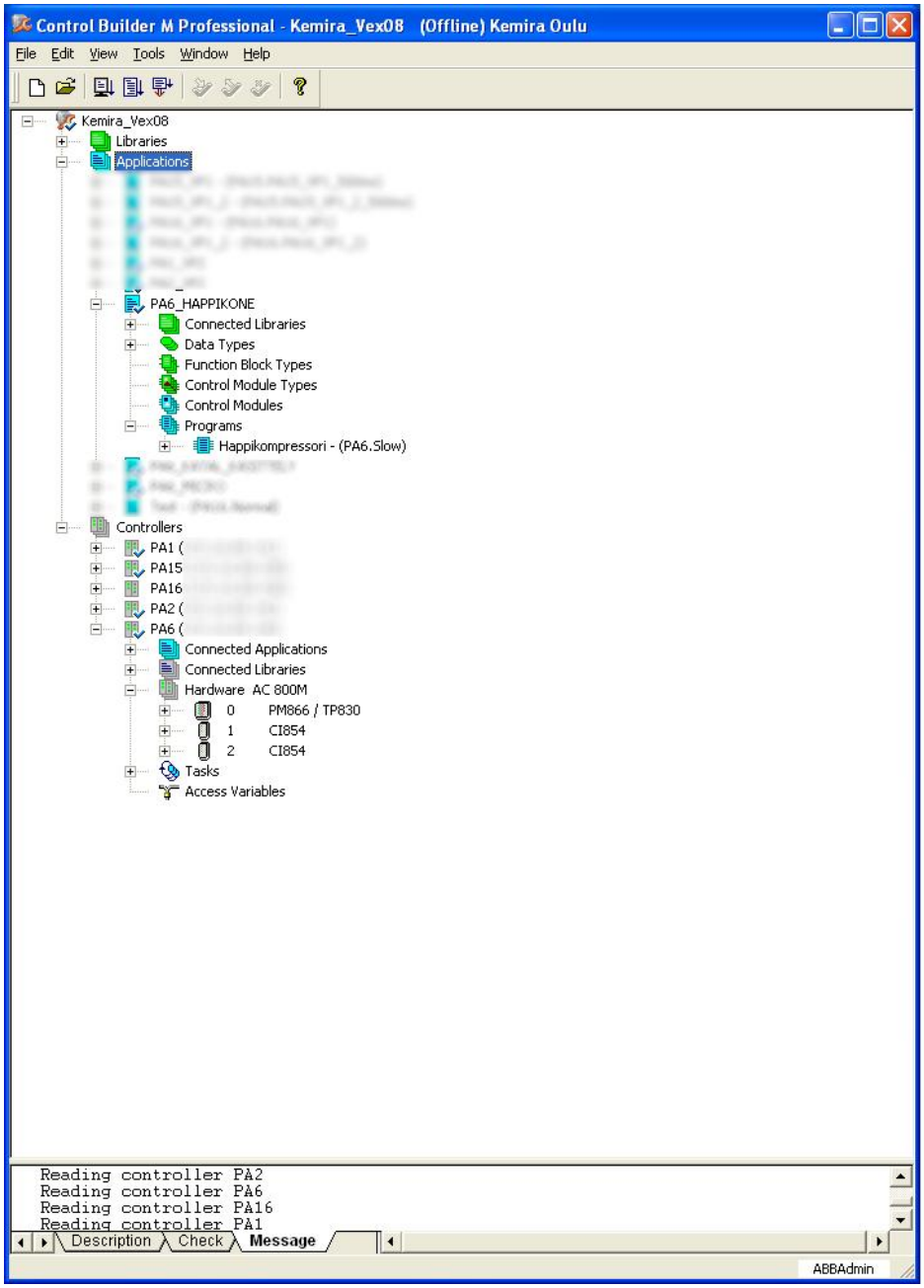
Control Builder M Professional on 800xA-järjestelmän ohjelmointityökalu. Tällä työkalulla tehdään ohjauskoodi, joka ladataan prosessiasemille. Control Builderilla voi käsitellä yhtä projektia kerrallaan, kun taas Engineerin workplacessa näkyvät kaikki projektit. Control Builderilla luodaan laitteiston ohjaamista koskeva sovellus kuten myös myös järjestelmän laitteisto-konfiguraatio koskien prosessiasemia sekä kommunikointi- ja IO-kortteja. Control Builderia ja Engineering workplacea tulee ajatella kahtena eri käyttöliittymänä prosessin ohjaamisen rakentamiseen ja ylläpitoon suunniteltuna työkaluna. (Kuva 5.) (9.)



KUVA 5. 800xA-systeemi (9)

4.5.1 Strukturoitu ohjelmointiperiaate

Control Builderissä ylin taso on Control Project eli projekti, jonka alta projekti lähtee puumaisesti alaspäin: projekti – aplikaatio – ohjelma – instanssi (kuva 6).



KUVA 6. Control Projectin puumainen rakenne.

Projekti voi sisältää 1−256 aplikaatiota. Aplikaatio eli sovellus on ylin puurakennemaisen projektin osa, jonka voi ladata prosessiasemalle. Prosessiasemalle voi ladata yhdestä kahdeksaan sovellusta. Sovelluksen voi rakentaa ohjelmilla tai Control moduleilla, jotka selitään myöhemmin, ja sekä että. Ohjelmat voivat sisältää eri code-blockeja, jotka voi käsittää

aliohjelmina. Control modulien etu on siinä, että ne vievät vähemmän muistia ja ovat yleensä nopeampia suorittaa kuin perinteiset ohjelmat.

Yksi sovellus voi sisältää 1−64 ohjelmaa. Tämän vuoksi eri prosessialueet voidaan kytketä eri nopeuksisiin intervalliaikoihin sekä ohjelmille voidaan antaa prioriteetit, niiden kriittisyyksien mukaan. Sovellukset jaetaan yleensä prosessikohtaisesti, kuten tässä työssä happikompressorille on oma sovelluksensa. Eri sovelluksien välinen tietoliikenne eroaa ohjelmien sisäisestä ja välisestä liikenteestä. (10.)

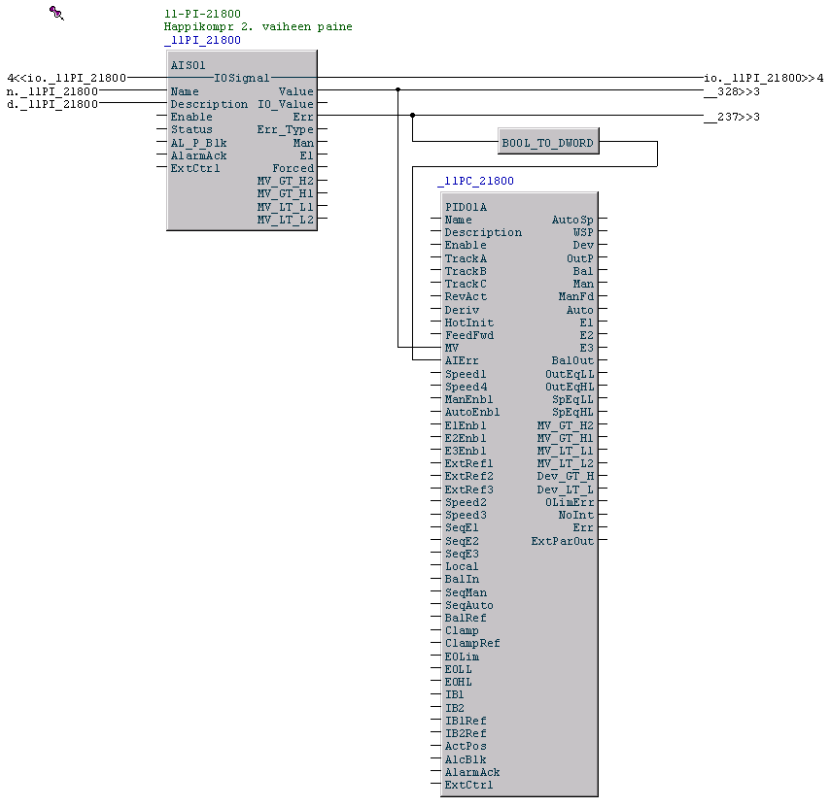
4.5.2 Ohjelmointikielet

800xA-järjestelmä tarjoaa useita eri ohjelmointikieliä, joihin kuuluvat toimilohko-ohjelmointi, struktuuritekstiohjelmointi, tikapuuohjelmointi, control module -ohjelmointi, instruction list -ohjelmointi ja sekvenssejä varten Sequential function chart -ohjelmointi.

Toimilohko-ohjelmointi

Functionblockit eli toimilohkot lisäävät toiminnallisuutta sovelluksessa. Esimerkiksi vakiokirjastoista BasicLib tarjoaa laskennallisuutta ja ProcessObjExtLib tarjoaa lohkoja toimilaitteiden ohjaamiseen. Sovelluksissa käytettävät toimilohkot ovat instansseja kirjastossa olevista objektityypeistä. Jotta tietyn kirjaston toimilohkoa voi käyttää sovelluksessa, täytyy kyseinen kirjasto kytkeä kyseessäolevaan sovellukseen. Halutessaan käyttäjä voi luoda omia toimilohkoja. (6.)

Happikompressorin käännöstyössä käytettiin toimilohko-ohjelmointia asiakkaan pyynnöstä. Toimilohko-ohjelmointi on verrattaen yksinkertaista. Control Builder näyttää toimilohkot laatikkoina, joissa vasemmalla puolella on tulot ja oikealla puolella lähdöt. (Kuva 7.)

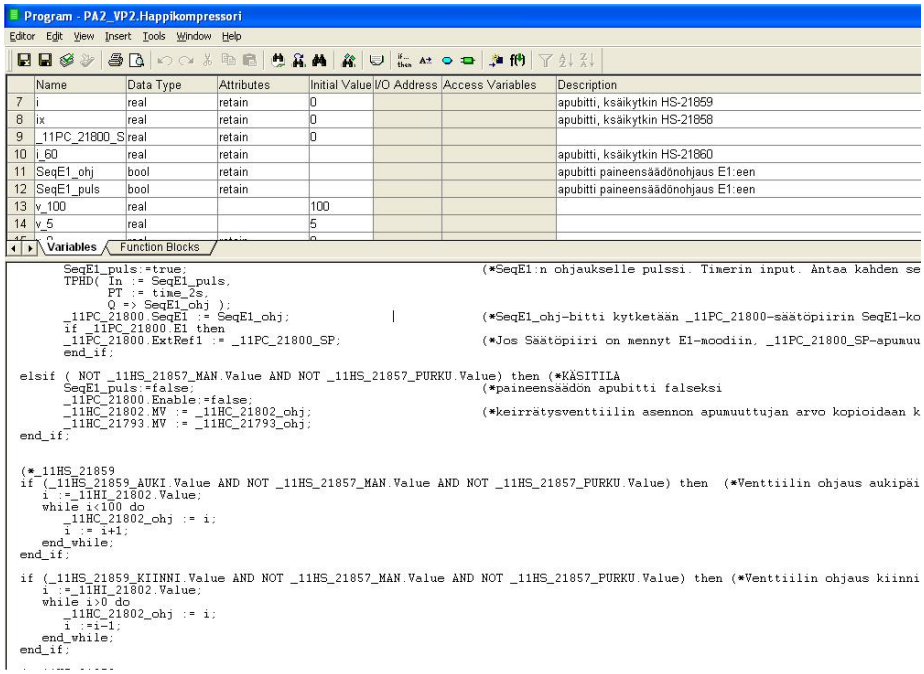


KUVA 7. Toimilohko-ohjelmointi

Toimilohkojen suoritusjärjestys code blockien sisällä on ylhäältä alas ja oikealta vasemmalle. Ohjelmoinnissa käytetään kirjaston tarjoamia toimilohkoja ja funktioita. Toimilohkoissa on sisääntulot ja ulostulot. Toimilohko-ohjelmoinnissa, kuten kaikissa muissakin, täytyy muistaa, että kytkettävät inputit ja outputit ovat samaa datatyyppiä. (14.)

Struktuuritekstiohjelmointi

Struktuuriteksti pohjatuu Pascal- ja C-ohjelmointikieliin. C-kielen kanssa samankaltaisuuksia on ydinkielessä, jonka toimintaa monipuolistetaan kirjastoista haettavilla toimilohkoilla, jotka saadaan toimimaan ja näkymään myös tekstimuotoisina. Struktuuriteksti on parhaimmillaan selkeää ja helppolukuista ja hyvien kommentointimahdollisuuksien vuoksi monimutkaisimpia kohtia voidaan selkeyttää lukijalle (kuva 8).

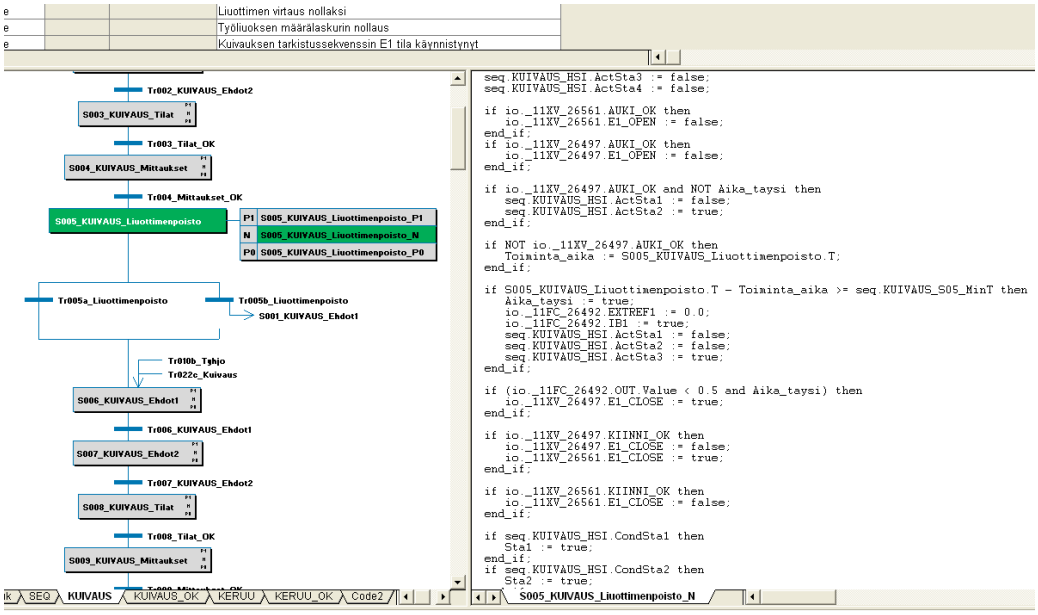


KUVA 8. Struktuuritekstimuotoinen ohjelmointi

Control Builderin ollessa On-line-tilassa eli yhteydessä prosessiasemaan, struktuuritekstimuotoinen code block voidaan myös näyttää toimilohko- tai tikapuumuodossa. (12; 13.)

Sekvenssiohjelmointi

Sequential Function Chart eli sekvennsiohjelmointikieli on helpoin ja järkevin tapa rakentaa sekvenssi. Se pohjautuu Grafcetiin – graafiseen sekvenssin kuvaukseen – joka kehitettiin Ranskassa ’70-luvulla. Control Builderin ohjelmointinäkymässä on kaksi ikkunaa: vuokaavio ja koodi-ikkuna. Vasemmalle rakennetaan prosessin vaatima vuokaavio. Grafcetin sääntöjen mukaan Control Builderissa askeleessa on kohdat P1, N ja P0. (Kuva 9.)



KUVA 9. Sekvenssiohjelmointi

P1 on nouseva reuna, kun askeleeseen tullaan. N on askeleen kohta, jota suoritetaan niin kauan kuin askeleessa viivytään. P0 on askeleen laskeva reuna. P1 ja P0 kohdissa olevat käskyt suoritetaan vain kerran, kun taas N kohdassa olevia käskyjä ja tarkistuksia tehdään niin kauan kunnes siirtoehdot ovat täyttyneet. Tr:ssä ovat askeleiden välissä olevat askelehdot. Askeleesta toiseen siirrytään, kun Tr:ssä olevat ehdot täyttyvät. Siirtoehdon täytyy olla boolean-muotoinen.

Koodin kirjoitus tapahtuu valitsemalla halutun askeleen kohta tai etenemisehto ja kirjoittamalla käskyt ja tarkistukset oikealla sijaitsevaan koodi-ikkunaan. Sekvenssinohjaukseen voi käyttää vain struktuuritekstipohjaista kieltä. (15; 16.)

Kontrollimoduuliohjelmointi

Kontrollimoduulit (Control Module) ovat toimilohkomaisia, mutta ne voivat pitää sisällään struktuuritekstillä tehtyä koodia, grafiikkaa ja muita toimilohkoja tai kontrollimoduuleja. Kontrollimoduuleilla ohjelman teko eroaa hieman toimilohkomaisesta ja ohjelmakeskeisestä työskentelystä. (6.)

Kontrollimoduuliohjelmointia ei voida sinällään pitää ohjelmointikielenä, koska kontrollimoduuli toimii käytännössä ohjelman tai koodin säilöjänä. Kontrollimoduuli on siis ohjelma, jonka ympärillä on koodia ja jonka sisälle voi kirjottaa koodia. Sen sisällä voi käyttää mitä tahansa viidestä eri ohjelmointikielestä. Kontrollimoduulin vahvuus on siinä, että se käyttää vähemmän muuttujia kuin perinteisemmät ohjelmat. Tämän lisäksi kontrollimoduuleissa suoritusjärjestykseen ei voi vaikuttaa, vaan se automaattisesti lajittelee koodin mahdollisimman vähän kuormittavaksi, kuitenkin ylläpitäen halutun toiminnallisuuden. Näistä syistä kontrollimoduulipohjainen koodi käyttää vähemmän muistia ja kuormittaa prosessiasemaa vähemmän kuin perinteinen ohjelmamuotoinen ohjelmointi. Kontrollimoduulipohjainen ohjelmointi pitää sisällään muitakin ominaisuuksia ja huomioon otettavia asioita, mutta siihen ei tässä perehdytä, koska käännöstyö toteutettiin toimilohko-ohjelmoinnilla. (17.)

4.5.3 Muuttujat

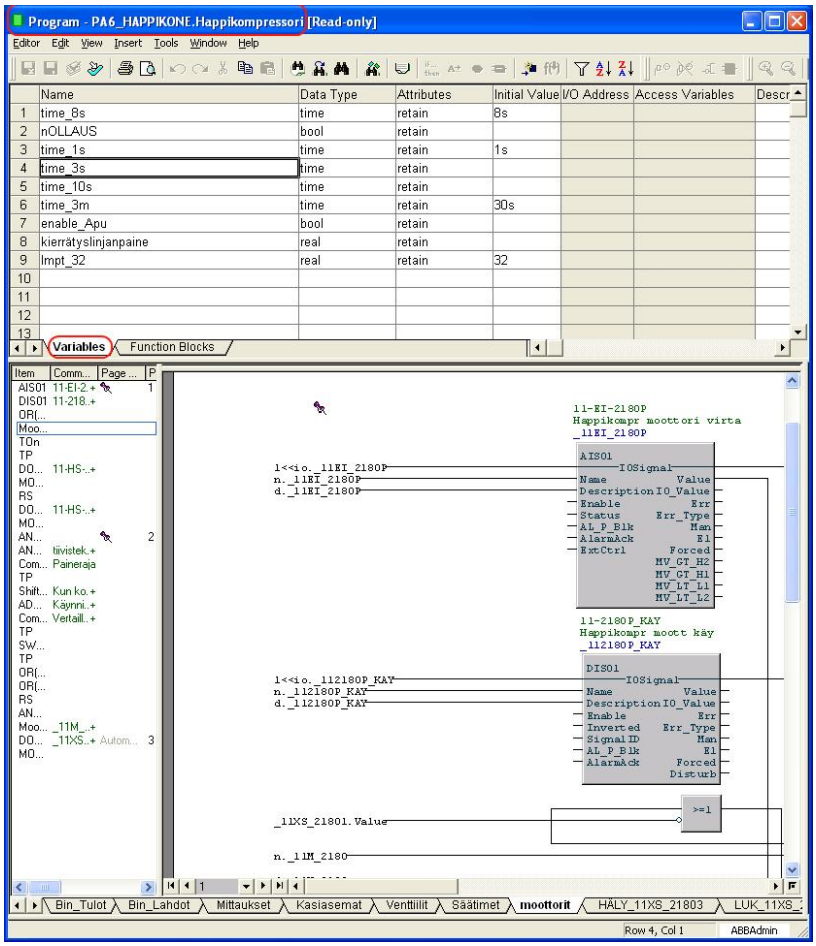
Ohjelman tekemiseen tarvitaan muuttujia. Muuttujiin tallennetaan tietoa, jota ohjelma niihin kirjoittaa. Muuttujat ovat aina jotakin datatyyppiä (bool, integer, string yms.). 800xA-järjestelmässä muuttujia on monia eri tyyppejä.

Muuttujien ominaisuuksiin kuuluvat myös attribuutit, joita ovat retain, coldreatain, constant ja hidden. Jos muuttujalla on attribuuttina retain, muuttujan arvo säilyy tallennettuna, kun prosessiasemalle suoritetaan warm restart, jolloin prosessiasemassa oleva sovellus käynnistetään uudelleen. Coldretain attribuutilla muuttujan arvo säilyy vaikka prosessiasemalle tehtäisiin cold restart eli prosesiasema tyhjennetään ja sovellus ladataan uudelleen. Constant attribuutti, tarkoittaa ettei ohjelma voi kirjoittaa muuttujalle arvoa, vaan sen arvo kopioidaan initial valuesta, eli alkuarvosta. Hidden attribuutti piilottaa muuttujan OPC-serveriltä. Tämä tarkoittaa, ettei 800xA-järjestelmä pysty lukemaan tai kirjoittamaan siihen. Jos muuttujaa ei

tarvita operaattorikäyttöliittymässä, voidaan muuttuja piilottaa, jottei se kuormittaisi verkkoa.

Ohjelmamuuttujat

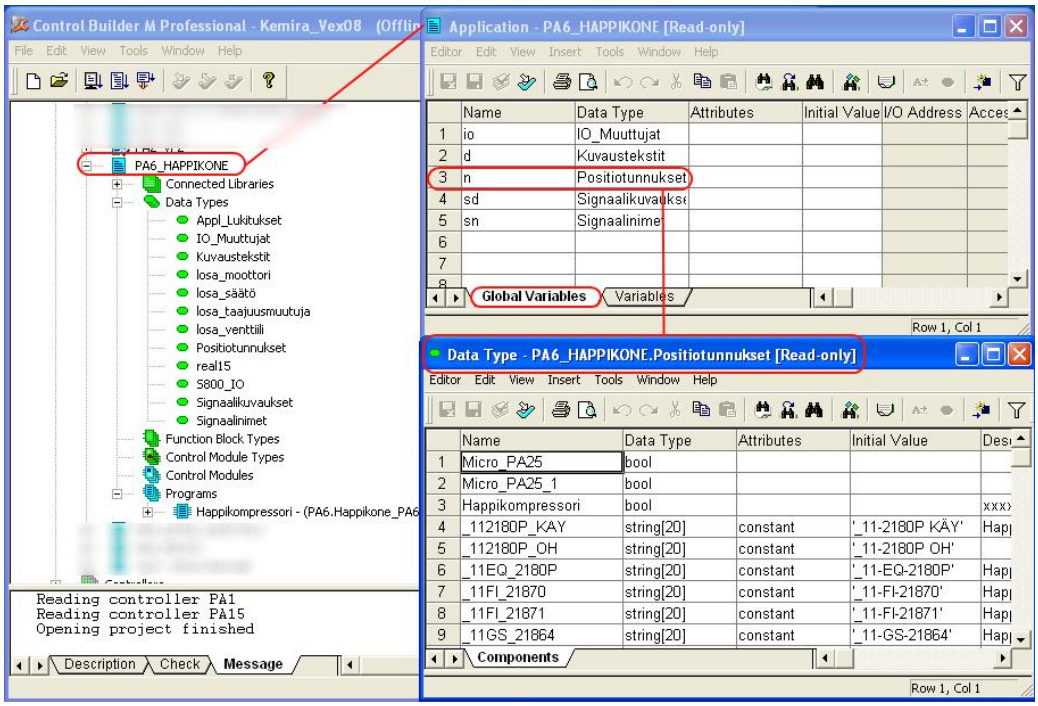
Ohjelmamuuttujia (program variable) käytetään nimensä mukaisesti ohjelman sisällä. Sitä ei voi käyttää eikä siitä voi lukea eri ohjelmassa. Ohjelman sisäiset muuttujat määritellään Control Builderin muuttujat-välilehteen (Kuva 10.)



KUVA 10. Ohjelmamuuttujat määritellään ohjelmaeditorissa variables-välilehdelle

Globaalit muuttujat

Globaalit muuttujat (global variables) ovat sovelluksen sisäisiä muuttujia. Sovelluksen sisäiset eri ohjelmat voivat kaikki lukea näitä muuttujia ja kirjoittaa niihin. Globaalit muuttujat määritellään sovelluksen tasolla. Tapana on määritellä globaaliksi muuttujiksi strukturoituja datatyyppejä, joihin käytettävät globaalit muuttujat kytketään. (Kuva 11.)



KUVA 11. Globaalit muuttujat määritellään sovellustasolla global variables – välilehteen

Harvinaisemmat muuttujat

Uusimmassa 800xA-järjestelmän versioissa 5.1 sovellusten väliseen kommunikointiin voidaan käyttää kommunkointimuuttjia (communication variable). Kommunikointimuuttujille annetaan nimi sekä suunta in tai out eli luetaanko muuttujaa jostain muualta vai lähetetäänkö sitä verkkoon. Lukevassa päässä muuttujan täytyy olla saman niminen kuin lähettävässä. Lähettävässä päässä IP-kenttään kirjoitetaan sen prosessiaseman IP-osoite,

jossa muuttuja sijaitsee. Lukevassa päässä IP-kenttään kirjoitetaan sen prosessiaseman IP-osoite, josta luetaan. Vaihtoehtoisesti IP-kenttään voidaan kirjoittaa ”auto”, jolloin Control Builder selvittää IP-osoitteet prosessiasemalle latauksen vaiheessa.

Harvemmin käytettyjä muuttujia ovat paikalliset muuttujat (local variable), joita käytetään toimilohkojen ja control modulien sisällä.

Muuttujiin viittaaminen

Ohjelma- ja kontrollimoduulieditoreissa muuttujiin viitataan nimellä. Jos tarvittava muuttuja on strukturoidun datatyypin sisällä, päästään strukturoidun datatyypin sisälle merkillä ”.”, kuten kuvassa 14 nähdään.

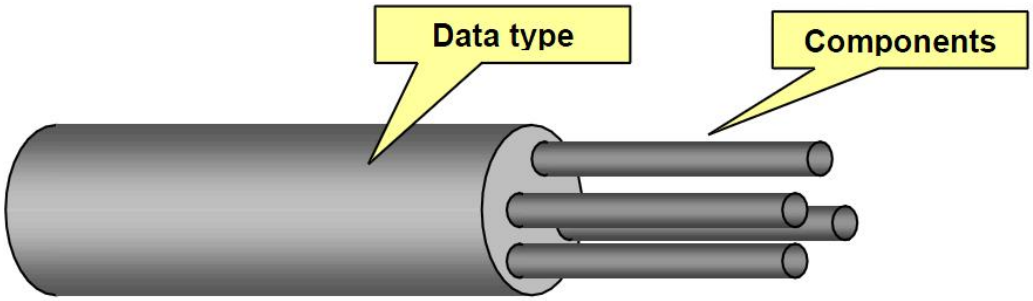
4.5.4 Datatyypit

800xA-järjestelmä käyttää useita eri perusdatatyyppejä ja strukturoituja datatyyppejä. Pohjan datatyypeille luovat simple datatyypit eli perusdatatyypit (taulukko 1).

TAULUKKO 1. 800xA-järjestelmän perusdatatyypit

Data type	Description	Bits	Default value	Remarks
bool	Boolean	1	False, 0	True and False (1/0 and On/Off are permitted alternatives)
Dint	Double integer	32	0	range is -2 147 483 647 to +2 147 483 647
Int	Integer	16	0	range is -32767 to +32767
Uint	Unsigned integer	16	0	range is 0 to 63335.
String	Character string *		‘’ (empty string)	see below
DWord	Bit string	32	0	Each of the 32 bits may take a value of True or False (1 or 0)
Word	Bit string	16	0	The equivalent of the 16 bit registers found in some PLCs. Each of the 16 bits may take a value of True or False (1 or 0)
Time	Duration		0s	Held in internal coded format. There are data type conversion functions to extract the actual hours, minutes and seconds from a time variable. Used for preset and elapsed variables connected to timers.
Date_And_Time	Date and time of day		1979-12-31-00:00:00	Stored in internal format, but may be converted to other data types using functions in the System library.
Real	Real number	32	0.0	Range is ± 1.7014 x 10±38

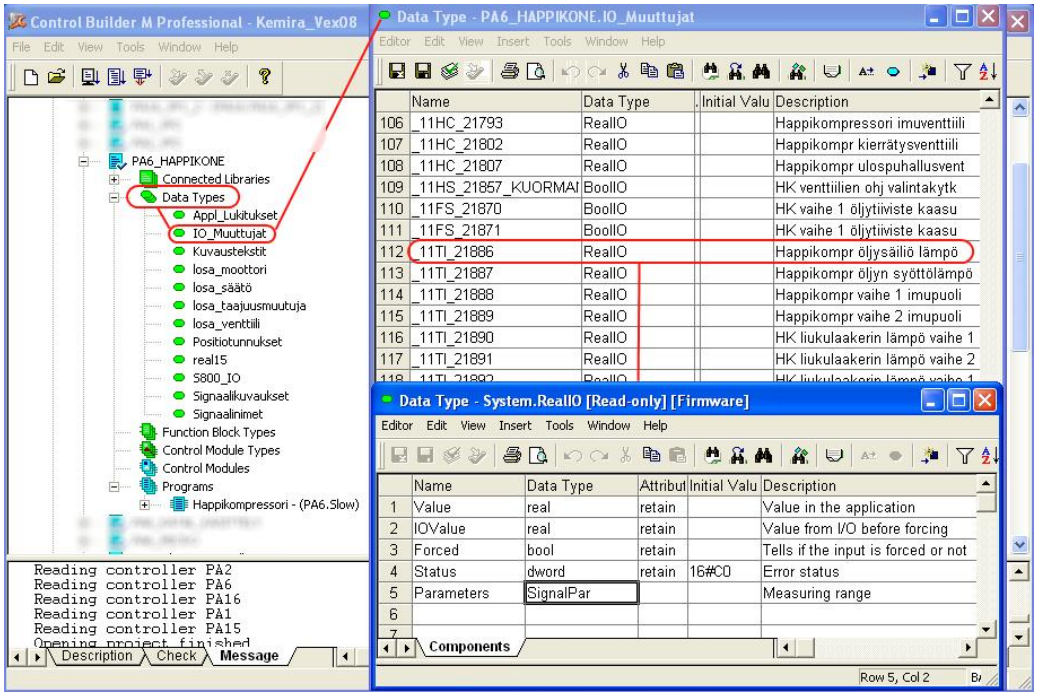
Sovelluksessa käytettävät muuttujat kytketään niitä vastaaville datatyypeille. Esimerkiksi analogiamuuttujalle käytettävä datatyyppi on RealIO, mutta reaalilukumuuttujan datatyyppi on real. Real on perusdatatyyppi ja RealIO on strukturoitu datatyyppi, joka sisältää yhden tai useamman muuttujan, jotka ovat jotain muuta datatyyppiä oli se sitten perusdatatyyppi tai toinen strukturoitu datatyyppi. (Kuva 12.)



Kuva 12. Strukturoitu datatyyppi (6)

Datatyyppi voi olla järjestelmässä valmiina oleva datatyyppi ja käyttäjä voi luoda täysin oman strukturoidun datatyypin.

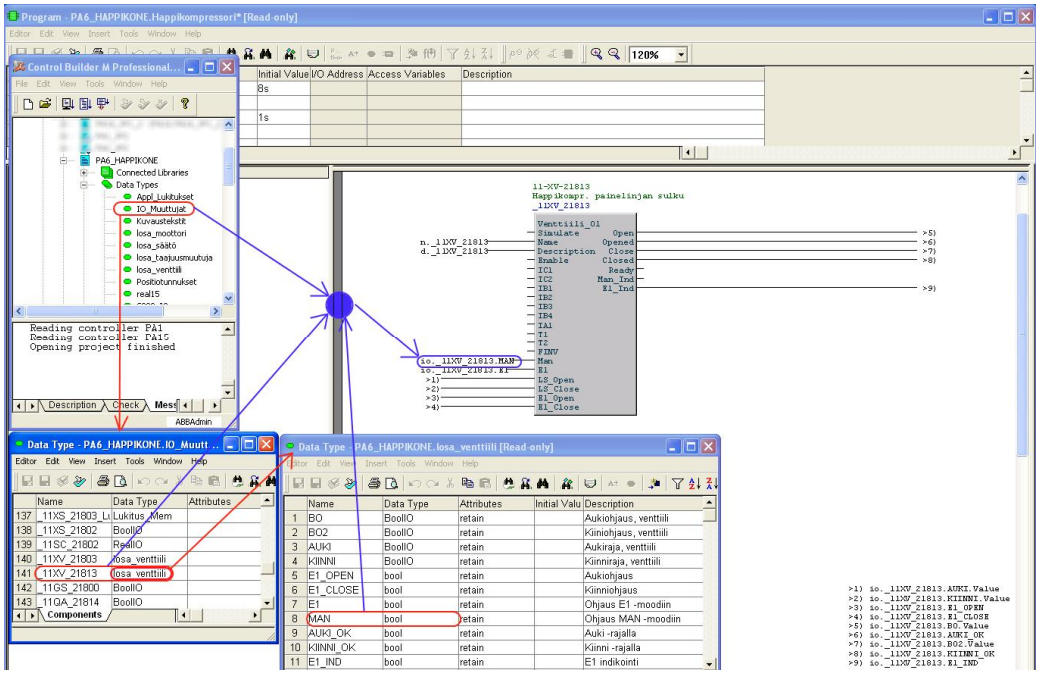
Strukturoitu datatyyppi pitää sisällään muuttujia, jotka on kytketty loppujen lopuksi perusdatatyyppeihin. Strukturoidut datatyypit pohjautuvat aina perusdatatyyppeihin. Esimerkiksi mainittu RealIO pitää sisällään muuttujat Value, IOValue, Forced, Status ja Parameters, joiden vastaavat datatyypit ovat real, real, bool, dword ja strukturoitu datatyyppi SignalPar. (Kuva 13.)



KUVA 13. RealIO:n datatyyppi

Value kertoo sovelluksessa käytettävän arvon, IOValue on IO-kanavaan tuleva arvo ennen pakottamista, forced kertoo onko Value pakotettu, Status kertoo mittauksen tilan ja Parameters kertoo muun muassa mittauksen ylä- ja alarajat. (6; 11.)

Käytännön esimerkkinä seuraavalla sivulla olevassa kuvassa 14 on käännöstyössä käytetty itse tehty strukturoitu datatyyppi on/off-venttiilille. Sovellukseen on luotu io-niminen globaali muuttuja, jonka datatyypiksi on annettu IO_Muuttujat. IO_Muuttujat-strukturoitu datatyyppi pitää sisällään kaikki sovelluksessa käytettävät IO-kommunikointiin käytettävien toimilohkojen muuttujat. IO_Muuttujat -datatyypin sisälle on luotu muiden muassa _11XV_21813-niminen muuttuja, jonka datatyypiksi on annettu itse tehty strukturoitu datatyyppi muotoa losa_venttiili. Tämä datatyyppi sisältää muuttujat, jotka kytketään Venttiili_01-toimilohkon yleisimmin käytettyihin tuloihin ja lähtöihin.



KUVA 14. Itse tehdyn strukturoidun datatyypin käyttö

Kuvassa 14 on esimerkin vuoksi MAN-käskyyn kytketty muuttuja. Polku tälle muuttujalle on ”io._11XV_21813.MAN”, jossa ”io” on globaali muuttuja, jonka sisällä on ”_11XV_21813”-niminen muuttuja, jonka sisällä on ”MAN”-muuttuja. Tämä yksittäinen MAN-muuttuja on vain ja ainoastaan _11XV_21813-muuttujan käytettävissä.

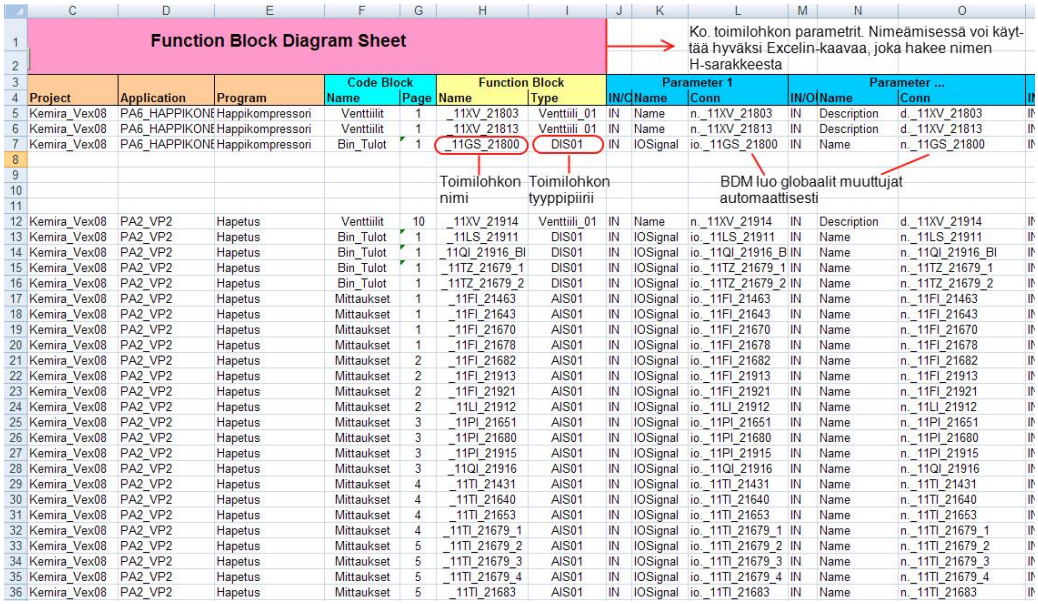
Strukturoidun datatyypin käytön edut ovat yhteinäisyydessä ja nopeudessa. Sen sijaan, että tehtäisiin jokaiselle tarvittavalle toimilohkon inputille ja outputille omat muuttujat erikseen, tehdään yksi strukturoitu datatyyppi toimilohkotyypille. Tällöin ei tarvitse luoda kuin yksi muuttuja tälle toimilohkolle ja kytkeä se sitä vastaavalle datatyypille. Tällä tavalla jokaisen saman tyyppisen toimilohkon muuttujien muodot ovat samat, ne löytyvät helpommin ja aikaa kuluu vähemmän.

4.5.5 Bulkdata manager

Bulkdata manager (BDM) on lisäosa Microsoft Excel -taulukkolaskentaohjelmaan. BDM on 800xA-järjestelmän työkalu, joka

tarjoaa nopean tavan luoda yksinkertaisia instansseja ohjaussovellukseen, mutta sitä voidaan käyttää myös lukuisiin muihin 800xA-järjestelmän osien konfigurointiin.

BDM:llä voidaan esimerkiksi määrittää tyyppipiirin nimi, tyyppipiirin tyyppi (AIS, DIS, PIDA01) sekä mihin paikkaan sovelluksessa se viedään. Kun kaikki halutut piirit on laitetty Excel-tiedostoon, voidaan instanssit ajaa ohjelmaan, jolloin BDM luo määritellyt piirit nimineen ja kytkee haluttuihin kohtiin muuttujat datatyyppeineen. (Kuva 15.)



	C	D	E	F	G	H	I	J	K	L	M	N	O
1–2	Function Block Diagram Sheet												
3				Code Block		Function Block		Parameter 1				Parameter ...	
4	Project	Application	Program	Name	Page	Name	Type	IN/O	Name	Conn	IN/O	Name	Conn
5	Kemira_Vex08	PA6_HAPPIKONE	Happikompressori	Venttiilit	1	_11XV_21803	Venttiili_01	IN	Name	n._11XV_21803	IN	Description	d._11XV_21803
6	Kemira_Vex08	PA6_HAPPIKONE	Happikompressori	Venttiilit	1	_11XV_21813	Venttiili_01	IN	Name	n._11XV_21813	IN	Description	d._11XV_21813
7	Kemira_Vex08	PA6_HAPPIKONE	Happikompressori	Bin_Tulot	1	_11GS_21800	DIS01	IN	IOSignal	io._11GS_21800	IN	Name	n._11GS_21800
8–11													
12	Kemira_Vex08	PA2_VP2	Hapetus	Venttiilit	10	_11XV_21914	Venttiili_01	IN	Name	n._11XV_21914	IN	Description	d._11XV_21914
13	Kemira_Vex08	PA2_VP2	Hapetus	Bin_Tulot	1	_11LS_21911	DIS01	IN	IOSignal	io._11LS_21911	IN	Name	n._11LS_21911
14	Kemira_Vex08	PA2_VP2	Hapetus	Bin_Tulot	1	_11QI_21916_BI	DIS01	IN	IOSignal	io._11QI_21916_B	IN	Name	n._11QI_21916_BI
15	Kemira_Vex08	PA2_VP2	Hapetus	Bin_Tulot	1	_11TZ_21679_1	DIS01	IN	IOSignal	io._11TZ_21679_1	IN	Name	n._11TZ_21679_1
16	Kemira_Vex08	PA2_VP2	Hapetus	Bin_Tulot	1	_11TZ_21679_2	DIS01	IN	IOSignal	io._11TZ_21679_2	IN	Name	n._11TZ_21679_2
17	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	1	_11FI_21463	AIS01	IN	IOSignal	io._11FI_21463	IN	Name	n._11FI_21463
18	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	1	_11FI_21643	AIS01	IN	IOSignal	io._11FI_21643	IN	Name	n._11FI_21643
19	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	1	_11FI_21670	AIS01	IN	IOSignal	io._11FI_21670	IN	Name	n._11FI_21670
20	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	1	_11FI_21678	AIS01	IN	IOSignal	io._11FI_21678	IN	Name	n._11FI_21678
21	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	2	_11FI_21682	AIS01	IN	IOSignal	io._11FI_21682	IN	Name	n._11FI_21682
22	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	2	_11FI_21913	AIS01	IN	IOSignal	io._11FI_21913	IN	Name	n._11FI_21913
23	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	2	_11FI_21921	AIS01	IN	IOSignal	io._11FI_21921	IN	Name	n._11FI_21921
24	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	2	_11LI_21912	AIS01	IN	IOSignal	io._11LI_21912	IN	Name	n._11LI_21912
25	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	3	_11PI_21651	AIS01	IN	IOSignal	io._11PI_21651	IN	Name	n._11PI_21651
26	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	3	_11PI_21680	AIS01	IN	IOSignal	io._11PI_21680	IN	Name	n._11PI_21680
27	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	3	_11PI_21915	AIS01	IN	IOSignal	io._11PI_21915	IN	Name	n._11PI_21915
28	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	3	_11QI_21916	AIS01	IN	IOSignal	io._11QI_21916	IN	Name	n._11QI_21916
29	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	4	_11TI_21431	AIS01	IN	IOSignal	io._11TI_21431	IN	Name	n._11TI_21431
30	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	4	_11TI_21640	AIS01	IN	IOSignal	io._11TI_21640	IN	Name	n._11TI_21640
31	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	4	_11TI_21653	AIS01	IN	IOSignal	io._11TI_21653	IN	Name	n._11TI_21653
32	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	4	_11TI_21679_1	AIS01	IN	IOSignal	io._11TI_21679_1	IN	Name	n._11TI_21679_1
33	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	5	_11TI_21679_2	AIS01	IN	IOSignal	io._11TI_21679_2	IN	Name	n._11TI_21679_2
34	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	5	_11TI_21679_3	AIS01	IN	IOSignal	io._11TI_21679_3	IN	Name	n._11TI_21679_3
35	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	5	_11TI_21679_4	AIS01	IN	IOSignal	io._11TI_21679_4	IN	Name	n._11TI_21679_4
36	Kemira_Vex08	PA2_VP2	Hapetus	Mittaukset	5	_11TI_21683	AIS01	IN	IOSignal	io._11TI_21683	IN	Name	n._11TI_21683

KUVA 15. Bulkdata managerin käyttöliittymä

4.5.6 Laitteistokonfiguraatio

Laitteistokonfigurointi (hardware-configuration) tapahtuu myös puumaisesti. Puurakenteen ylimmäisellä tasolla on kontrollerikansio (controller), jonka alle lisätään käytettävät prosessiasemat. Prosessiasemille annetaan niitä vastaavat verkon IP-osoitteet. Prosessiaseman alle laitteisto kasataan siten kuin se fyysisesti on rakennettu. Tarvittavat laitteet lisätään instansseina laitteisto-kirjastoista, joihin on ajettu kyseessä olevien laitteiden GSD-

tiedostot. Tarvittaessa on mahdollista luoda oma kirjasto niitä laitteita varten, joita 800xA-järjestelmässä ei ole valmiiina. (18.)

4.6 Graphics Builder

Graphics Builder on työkalu, jolla tehdään PG2-muotoisia openrointikuvia. Graphics Builderilla on historiansa puolesta kytköksiä Microsoft Visual Basic -kuvanmuokkaohjelmaan, jolla tehtiin aikaisemmat Visual Basic -muotoiset operointikuvat.

Graphics Builder on joustava ja tehokas kuvanmuokkaustyökalu. Kuviin on mahdollista lisätä dynaamisuutta omien kaavojen muodossa. Kirjaston tarjoamat graafiset elementit tarjoavat myös dynaamisuutta. Esimerkiksi kuljettimet, moottorit ja venttiilit eivät ole staattisia elementtejä vaan näyttävät tilansa ja osoittavat, kun ne ovat vaihtamassa tilaa.

Helpoin tapa piirtää operointikuva on luoda ensimmäisenä kuvassa näytettävä staattinen pohja eli putkistot, säiliöt ja muut tarvittavat elementit. Jos samanlaisia staattisia elementtejä käytetään usein, kannattaa niistä tehdä malli, joka tallennetaan Graphics Builderin solution-kirjastoon. Kun staattinen osa kuvasta on valmistunut aletaan lisätä dynaamisia elementtejä.

Klikkaamalla operaattorikuvissa olevia kirjastoelementtejä, kuten moottoria, avautuu siitä kyseessä olevan elementin faceplate, jonka kautta voidaan ohjata toimilaitetta. Faceplatesta käyttäjä pääsee myös tarkastelemaan objektin aspekteja, kuten vaikkapa säätöpiirin mittauksesta, asetusarvosta ja ohjauksesta muodostettua trendiä.

4.7 Laitteisto

800xA-järjestelmän oletuslaitteisto kuuluu AC 800M -perheeseen, mutta siihen on mahdollista liittää muitakin ABB:n laitteistoja sekä muiden valmistajien laitteita. AC 800M:n perusyksikkö koostuu prosessiasemasta, kommunikaatiovälikappaleesta, s800 I/O-moduulista ja teholähteestä.

Happikompressorin ohjelma ajettiin PM866-prosessiasemalle, jonka pohjalevynä toimii TP830-pohjalevy. Profibus-masterina toimii CI854-kortti ja profibuskorttina I/O-korteilla toimii CI850-kortti.

5 KÄÄNNÖSTYÖ

Vetyperoksidi- ja muurahaishappotehtaan ohjelmien käännöstyö tehtiin suoraan vanhan koodin pohjalta. Happikompressorissa päädyttiin toiseen ratkaisuun, koska vanha koodi oli eri valmistajan toimittama ja koodi oli kirjoitettu tikapuumuodossa. Vanha koodi oli lähestulkoon kommentoimatonta, joten sen selvittäminen olisi ollut työläämpää kuin toiminnankuvauksen mukaan tehty koodi ja olisi mahdollisesti viivästyttänyt koko uudistusprojektia.

5.1 Toiminnankuvaus

Etteplan Oyj:n Oulun toimipiste toimitti happikompressorin toiminnankuvauksen ja PI-kaaviot kolmesta happikompressorin osa-alueesta. Toiminnankuvaus oli kirjoitettu Atlas&Copcon materiaalien ja vanhan koodin pohjalta. Happikompressorin kolme eri osa-aluetta olivat tiivistekaasu, öljyvoitelu ja itse kompressoriosuus. Osa toiminnankuvauksesta löytyy liitteistä 1 ja 2.

5.1.1 Paineensäätö ja pumppaussuoja

Atlas&Copcon happikompressorin tehtävä on paineistaa happikaasu, jota käytetään vetyperoksidiprosessin hapetusosaprosessissa. Kompressori on tyypiltään kineettinen radiaalikompressori, jossa happikaasu virtaa vaiheiden akselien vastaisesti moottoreiden pyörivien siivistöjen vaikutuksesta. Yhtä tällaista perusosaa sanotaan vaiheeksi. Kyseessä olevassa happikompressorissa on kaksi tällaista vaihetta. Vaiheiden lukumäärä on yksi vaikuttava tekijä loppupaineeseen. Koska happikompressorin moottorin nopeutta ei voi säädellä, painetta säädetään säätöventtiilien avulla. Paineeseen vaikuttaa imu-, kierrätys- ja ulospuhallusventtiilin asento. Ideaalitilanteessa ulospuhallus- ja

kierrätysenttiili pysyvät kiinni. Paineen asetusarvo määräytyy sen mukaan, kuinka suuri vetyperoksiditehtaan happikaasun pyynti on.

5.1.2 Säätötavan valinta

Sopivan säätötavan valinta on optimointikysymys, johon pyritään etsimään vastausta kysymyksillä

- Sallittu käynnnistystiheys?
- Kompressorille sallittu lyhin käyntijakso?
- Energian kulutus eri säätötavoilla? (21.)

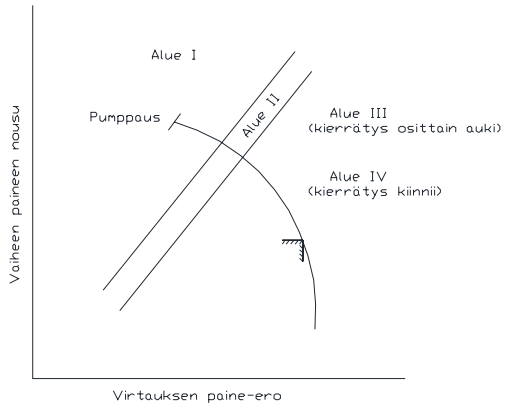
Happikompressorin moottorit olivat sähkökäyttöisä, mikä aiheuttaa rajoitteita muun muassa käynnistyksessä aiheutuvan lämmön vuoksi. Kompressoria ei mielellään tulla sammuttamaan ja käynnistämään tarpeen mukaan, vaan sen tulisi käydä jatkuvasti. Kompressorin sähkömoottori ei ole invertteriohjattu, joten sen kierrosnopeutta ei voida säätää. Painetta pyritään siten ohjaamaan portaattomalla kuristussäädöllä. Portaattomassa kuristussäädössä imuvirtausta kuristetaan, jolloin kompressorin tuotto alenee kulutusta vastaavaksi.

Haastavinta ohjauksessa oli pumppaussuojan toiminta ja sen sovittaminen paineensäätimen kanssa. Pumppaussuojan tehtävä on estää happikompressorin pumppaustila. Pumppaustila johtuu painepuolen paineenvaihtelusta. Jos paineenvaihtelun pulssien väli on 0,25−5 sekuntia, se tärisyttää laitteistoa ja voi vaurioittaa sitä. Ohjauksen tulee havaita pumppaus, ja kun se on havaittu laskea laitteen kuormitusta.

Kineettinen kompressori ei pysty toimimaan, jos tuottopuolen virtausta kuristetaan liikaa. Kompressorin siivistö sakkaa, ja virtauksen suunta kääntyy hetkellisesti päinvastaiseksi. Tällöin syntyy aikaisemmin mainittu virtauksen värähtely. Kompressorin mukaan liitetään usein

pumppauskäyrästö, mutta kompressorin ollessa iäkäs ovat useat sen dokumentit kadonneet. (21;23)

Pumppausta valvotaan kompressorin painesuhteen käyrältä. Kuvassa 16 esitetään kompressorin suhteellisen tuoton riippuvuutta suhteellisesta loppupaineesta. Alueella 1 on pumppausraja.



KUVA 16. Pumppaus riippuu virtauksen paine-eron suhteestä vaiheen paineen nousuun

Kompressorin toiminta-alueet on jaettu neljään alueeseen. Ideaalitilanne on, kun ollaan alueella neljä ja huonoin tilanne on kun ollaan alueella yksi, jolloin voi esiintyä pumppausta.

Alueella 1 kompressori toimii pumppauskäyrän vasemmalla puolella, jolloin pumppausvirhe on negatiivista. Se tarkoittaa sitä, että kaasun virtaus kääntyy hetkellisesti vastakkaiseen suuntaan. Tämä on lähin alue ennen varsinaista pumppaustilaa. Alueelle 1 joutuminen johtuu yleensä pyynnin laskusta. Tätä happikompressorin ohjaus ei kykene ennakoimaan, sillä hapetusosaprosessista ei ole takaisinkytkentää happikompressorille. Tällä

alueella imuventtiilin tulee avautua tarvittaessa paineensäätimen ohjaukseen perustuen ja kierrätysventtiilin tulee avautua pumppausuojan säätimen ohjauksen mukaan.

Alueella 2 kompressori toimii pumppauskäyrän oikealla puolella, mutta kuitenkin kuolleen alueen sisällä. Tällöin imuventtiili avautuu, jos kierrätysventtiili on täysin auki, paineensäätimen virheeseen perustuen. Kierrätysventtiiliä ohjaa pumppaussuojan säätöpiiri.

Alueella 3 kompressori toimii hyvin. Toiminta-alueella ollaan prosessin kannalta paremmalla puolella eli pumppausvirhe ei ole negatiivista eikä olla kuolleella alueella. Imuventtiili on vielä auki. Alueella 3 imuventtiiliä ohjaa pumppaussuojan säätöpiiri ja kierrätysventtiiliä ohjaa paineensäätöpiiri.

Ideaalinen tilanne on, että pumppauskäyrällä ollaan alueella 4, jolloin kierrätysventtiili on täysin kiinni ja painetta säädetään imuventtiilillä. Alueella 4 imuventtiiliä ohjaa paineensäätöpiiri.

5.1.3 Öljyvoitelu

Öljyvoitelun tehtävä on nimensä mukaan voidella kompressorin moottorin akselia. Kompressorissa on kaksi öljypumppua. Toinen öljypumppu on yhteydessä kompressorin päämoottorin akseliin eli se pyörii silloin, kun päämoottori pyörii. Tämän lisäksi on apuöljypumppu, jonka tehtävä on kierrättää öljyä päämoottorin ollessa sammuneena. Öljynpaineen laskiessa apuöljypumpun tulee myös käynnistyä, jotta öljynpaine pysyisi riittävän korkealla. Voiteluöljyn lämpötilaa ylläpidetään lämpövastuksella, joka ei ole lineaarisesti säädettävä, vaan ohjaus on päälle/pois -muotoa.

5.1.4 Tiivistekaasu

Tiivistekaasun tehtävä on pitää voiteluöljyä ja epäpuhtauksia poissa happikompressorin sisältä. Öljy on helposti syttyvää ainetta ja puhdas happikaasu on erittäin palamisherkkä. Jos nämä kaksi ainetta kohtaisivat,

olisi palamisvaara suuri. Tiivistekaasun paineen on oltava tarpeeksi korkealla, jotta happikompressori saa käynnistyä ja käydä.

5.2 Toteutus

Happikompressorin ohjauksen toteutus aloitettiin luomalla peruspiirit bulkdatamanagerilla. Digitaalitulot, analogiamittaukset, säätöpiirit ja venttiilit tehtiin omiin code-blockeihin. Järkevin ja helpoin tapa toteuttaa ohjaus oli luoda ensimmäisenä kaikki I/O-listassa olevat piirit, minkä jälkeen piirien ympärille luotiin tarvittava logiikka.

Happikompressori on prosessin kannalta kriittinen laite, ja sen käsittelemä happikaasu on palamisherkkää. Siksi laite sisältää useita lukituksia. Usein yksi lukitus voi aiheuttaa ketjureaktion, joka tekee alkuperäisen syyn selvittämisestä vaikeaa. Tästä syystä happikompressorin lukitusten seurantaan käytettiin itse tehtyä kirjastoelementtiä nimeltään Lukitus_Muisti. Lukitusmuisti koostuu useista SR-kiikuista ja pienestä pätkästä struktuuritekstiä, jotka on yksinkertaistettu yhdeksi toimilohkoksi, johon kytketään lukituksen aiheuttajat ja niiden kuvaustekstit. Lukitusmuisti-toimilohko näyttää graafisessa käyttöliittymässä kaikki voimassa olevat lukitukset sekä myös ensimmäisen lukituksen aiheuttajan.

5.3 Pumppauskäyrän muodostaminen

Suurinta vaivaa käännöksessä aiheutti se, että Allen&Bradleyn logiikkassa ei ollut käytetty yleisesti käytössä olevia säätöpiirejä, vaan ne oli toteuttu integraattoreilla, vertaajilla ja ramppitoiminnoilla. Tästä syystä erityisesti pumppaussuojan toiminnan selvittäminen oli vaikeaa. Pumppaussuojan toimintaa pyrittiin selvittämään vierailemalla Etteplanin toimistolla, jossa oli RsLogixin ohjelmointityökalu. Tarkoituksena oli selvittää millä mittauksilla pumppauskäyrä toteutettiin ja mitkä olivat sen raja-arvot.

Pumppauskäyrä muodostettiin 2. vaiheen virtausmittauksella ja paineella 2. vaiheen yli. Virtausmittaus muodostettiin paine-eromittauksesta.

Tulovirtausta 2. vaiheeseen kompensointiin lämpötilakertoimella. Painetta vaiheen yli kompensoitiin vakiokertoimella josta vähennettiin virhettä vakioarvolla. Tämän jälkeen lämpötilakompensoidusta tulovirtauksesta vähennettiin vaiheen yli olevan paineen kompensoitu arvo. Saadun käyrän raja-arvoista ja venttiilien asennoista saatiiin luotua neljä eri pumppausaluetta. Raja-arvojen lukemat tulevat todennäköisestä muuttumaan käyttöönotossa, sillä eri järjestelmästä tuodut arvot tuskin toimivat samalla tavalla. Pumppauskäyrän muodostamisessa olisi suuresti auttanut alkuperäisen pumppauskäyrän saaminen haltuun.

5.4 Tiedonsiirto, historiankeruu ja suoritusvälin valinta

Kuten aikaisemmin on mainittu, samoja muuttujia ei voi käyttää eri applikaatioissa. Happikompressori tarvitsi kuitenkin tilatiedon hapen syötön pikasulkuventtiililtä, joka sijaitsi eri applikaatiossa. Venttiilin tilatieto tuotiin happikompressorin applikaatioon kommunikointimuuttujan avulla.

Koska paineenvaihtelu on luonnostaan nopeaa, kytkettiin happikompressorin ohjelma taskiin, jonka intervalli-aika oli 250 millisekuntia ja prioriteetti 2. Tämän tulisi olla happikompressorin ohjaukselle riittävän nopea suoritusväli, jotta se voisi ohjata prosessia kyllin nopeasti.

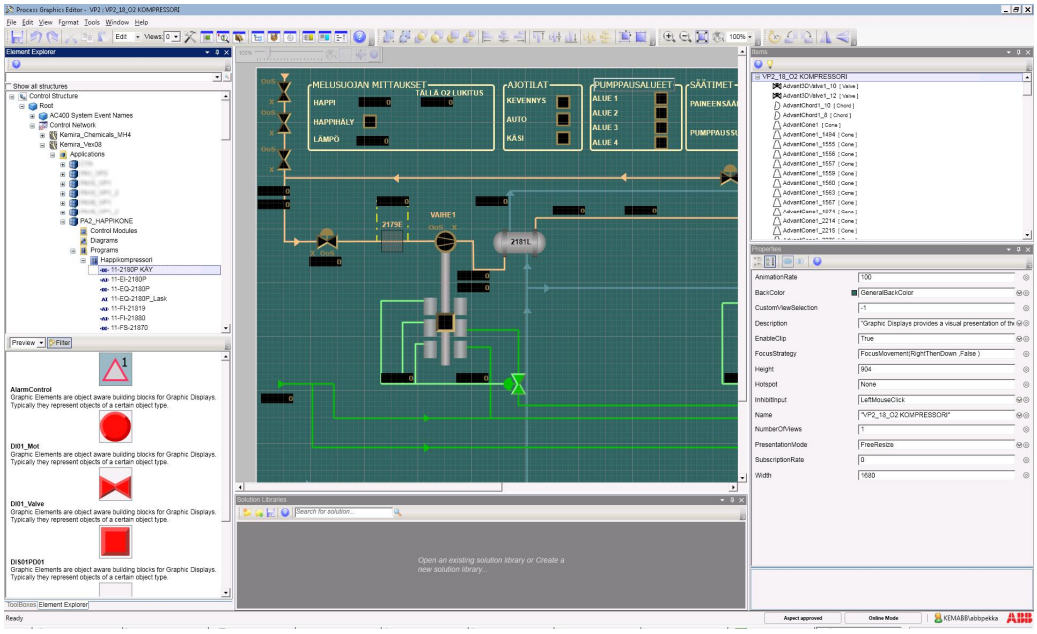
Historiankeruuta varten mittauksille, säätöpiireille ja moottoreille tehtiin loki-konfiguraatiot. Konfigraatiot tehtiin bulkdatamanagerilla, koska niiden luominen on pitkälti kopiointia ja toistoa. Järjestelmä kerää mittauksilta mittausarvoa, säätöpiireiltä mittausta, asetusarvoa ja ohjausta sekä moottoreilta käyntitietoa. Näitä tietoja kerätään lokitiedostoon alustavasti 10 sekunnin välein viikon ajaksi kerrallaan. Instanssien trendiaspektit osaavat hakea historitiedot niiden omista loki-tiedostoista.

Hälytysten hallintaa varten kaikkien piirien class-ominaisuudeksi asetettiin arvo 200, jolla mahdollistettiin happikompressorista tulevien hälytysten erottelu omaksi alueekseen.

6 VALVOMOKUVAT

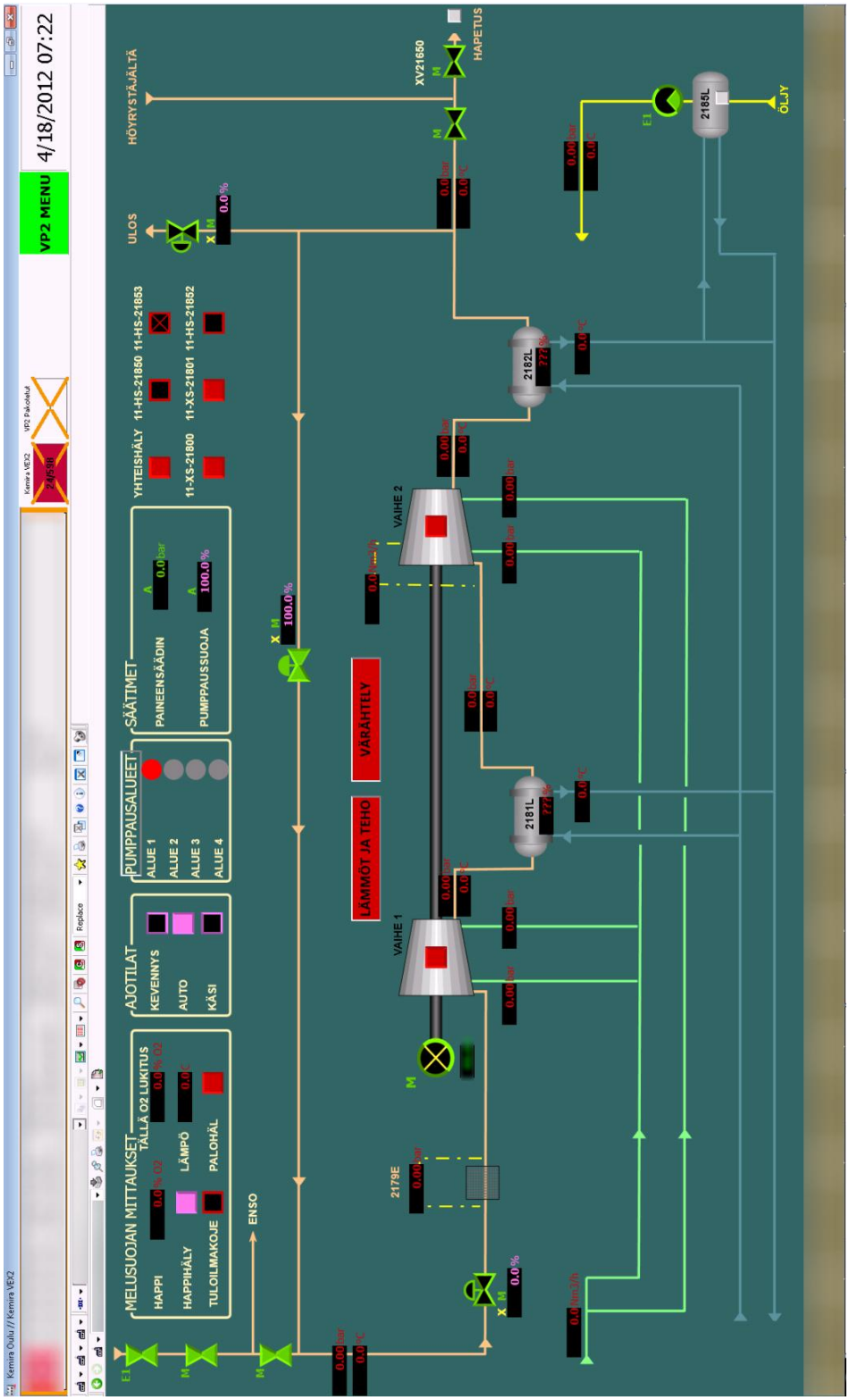
Kun piirit oli saatu valmiiksi, voitiin alkaa tehdä uusia operointikuvia. Operointikuvien pohjiksi valittiin vanhan järjestelmän Visual Basic -kuvat. Kaikki vanhassa järjestelmässä olevat kuvat käännettiin 800xA-järjestelmässä olevalla Migrate Tool -työkalulla. Migrate Tool -työkalu on käännöstyökalu, jolla vanhat Visual Basic -kuvat voidaan nopeasti kääntää uudemmiksi PG2-kuviksi. Migrate Tool käyttää hyväksi vanhojen ja uusien elementtien ”mäppäystä”, jossa vanhoille elementeille ja objekteille luodaan viittaukset vastaaviin PG2-elementteihin. Migrate Tool ei ole täysin vaivaton työkalu, vaan sen jälkeen on kuvia käytävä myös käsin läpi ja korjattava virheitä. Virheitä syntyy etenkin silloin, kun on käytetty itse tehtyjä kirjastoelementtejä.

Happikompressorista ei tarvinnut kääntää kuin pohjat, joten muokatuista elementeistä johtuvia ongelmia ei tullut. Muokattavia kuvia oli kaksi: happikompressori ja öljyvoitelu. Graphics Builder on yksinkertainen ja helppokäyttöinen kuvanmuokkausohjelma. Kun staattinen pohja on valmis, lisätään siihen element explorerista piirikohtaisia elementtejä. Kuvassa 17 vasemmalla puolella sijaitsee element explorer, jossa haetaan eri struktuureista tarvittavat objektit ja lisätään niiden elementtikuvat. Kuvanmuokkausnäkymän oikealla puolella näkyy properties-ikkuna, joka näyttää valitun kohteen ominaisuudet ja toiminnallisuudet.



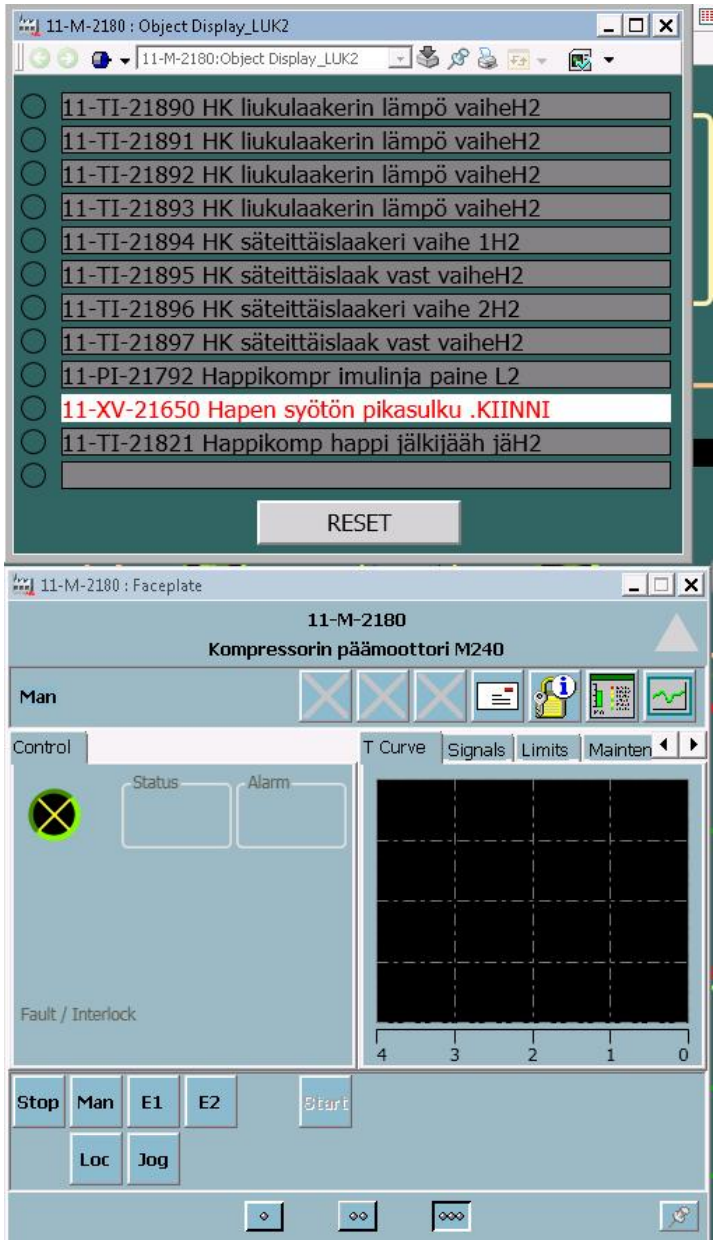
KUVA 17. Graphics Builderin muokkausnäkymä

Kuvat pyrittiin tekemään toimitetun PI-kaavion pohjalta, johon lisättiin tarvittaessa muita kohteita, kuten linkki pumppuassuojasta kertovaan html-sivuun. Kuvista pyrittiin tekemään mahdollisiman selkeitä ja yksioikoisia. Valmis happikompressorin kuva operator workplacesta katsottuna on seuraavalla sivulla (kuva 18).



KUVA 18. Happikompressorin operointinäyttö Operator Workplacen kautta

Toteutusvaiheessa mainitut lukitusmuistit liitettiin moottorilohkojen Faceplateen, jotta niiden käyttö olisi luontevampaa. Lukitusmuisti avataan faceplaten indicator-tasolla olevista ristillä merkityistä kuvakkeista. (Kuva 19.)



KUVA 19. Kompressorin päämoottorin lukitusmuistin graafinen ikkuna

7 TEHDASHYVÄKSYNTÄTESTI

Tehdashyväksyntätestissä ohjelmat käydään asiakkaan kanssa läpi ja tarkistetaan oikeat toiminnallisuudet. Testien jälkeen mahdolliset virheet korjataan ja muutokset toteutetaan lopullista käyttöönottoa varten. Tehdashyväksyntätestissä ohjelmien tulee olla siinä kunnossa, että lopullinen toiminnallisuuksien testaaminen voidaan suorittaa.

Tehdashyväksyntätestiä varten happikompressorille muodostettiin simulointiohjelma omaan code-blockiinsa. Simulointiohjelmaa tehtäessä tuli ottaa huomioon prosessin luonne ja selvittää, miten eri tapahtumat vaikuttavat toisiinsa. Simulointi vaatii myös ohjelman suoritusjärjestyksen ymmärtämisen.

Testiympäristössä ei ollut käytössä kortteja vaan pelkästään prosessorit. Tästä syystä simulointiohjelma sijoitettiin suoritettavaksi mittauksia ennen, jotta vikakoodit voitiin kirjoittaa hyväksi. Simuloinnissa pyrittiin simuloimaan paineen vaihtelua venttiilien asentoihin nähden. Ohjelmasta tuli niin sanottu ”looppi”, eli se kiertää ympyrää. Kuvassa 18 oikealla laidalla olevaan imupaineeseen asetellaan arvo, johon lisätään kierrätyslinjasta painetta kierrätysventtiilin asennosta riippuen. Kuvan oikean laidan päälinjan nurkasta siirretään painetta 1. vaiheeseen imuventtiilin asennon mukaan. Vaiheissa painetta nostetaan vakiokertoimilla. Loppupaineesta vähennetään paineen arvoa riippuen kierrätysventtiilin asennosta. Moottori- ja säätöpiirien moduuleissa on sisäänrakennettu simulointi, joka laitetaan päälle pistämällä simulate-bitti todeksi. Häiriöitä mallinnettiin satunnaisarvo-generaattorilla, jonka arvo lisätään päälinjan oikean laidan nurkkaan. Muita tarpeellisia mittauksia lisättiin simulointiohjelmaan, mutta niiden arvot olivat täysin käyttäjän aseteltavissa.

Tehdashyväksyntätesti kesti kolme päivää. Testissä käytiin läpi kaikki piirit ja niiden kytkennät oikeisiin I/O-kanaviin. Testien aikana päätettiin myös

mitä rajoituksia käyttöhenkilöstöllä on ohjaukseen ja mitkä asiat jätetään pelkästään ohjelman vastuulle.

Testeissä huolenaiheeksi nousi pumppauskäyrän muodostaminen. Vanhan ohjauksen mukaan tehty pumppauskäyrän muodostus koostui neljästä alueesta, joissa jokaisessa säätöventtiilejä ohjaava säätöpiiri vaihtui. Tämä voi rajatapauksissa aiheuttaa prosessiin huojuntaa. Uutta ohjausta varten tulisi kuitenkin ensin hankkia valmistajalta pumppauskäyrät. Muodostamalla pumppauskäyrä pelkästään virtausmittauksesta ja paineennoususta, säädöstä saataisiin pehmeämpi. Jälkimmäinen säätötapa saatetaan toteuttaa ennen käyttöönottoa, jos pumppauskäyrät saadaan hankittua tarpeeksi ajoissa.

Operointinäytöt käytiin läpi ja hyväksyttiin Kemiran vetyperoksiditehtaan operaattorihenkilöstön kanssa. Alkuperäinen Visual Basic -kuva, joka oli uuden PG2-kuvan pohjana, ei ollut koskaan miellyttänyt heitä. Operaattorihenkilöstön pyynnöstä operointinäytön pohjaa muutettiin toiveiden mukaisesti. Dyynamisista elementeistä pumppaussuojan alueindikaattorit muutettiin liikennevalomaisiksi.

8 YHTEENVETO

Insinöörityön tehtävänä oli onnistuneesti kääntää Kemira Chemicals Oy Oulun tehtailla sijaitsevan happikaasukompressorin ohjaus Allen&Bradley RsLogix 5 -järjestelmästä ABB:n 800xA-järjestelmään. Käännösprojekti käsitti ohjaussovelluksen lisäksi myös operointinäyttöjen piirtämisen. Työn ohella tehty kirjallinen osuus selvittää ABB:n 800xA version 5.1 ympäristöä ja 800xA-järjestelmän työvirtaa ja työskentelytapoja. Happikompressorin ohjauksen rakentaminen operointinäyttöineen on esimerkki siitä, kuinka 800xA-järjestelmässä työskennellään ohjelmoijan näkökulmasta.

Happikaasukompressorin toimintaan ja prosessoitavaan aineeseen tuli perehtyä, jotta turvallinen toiminta voitiin varmentaa. Happikaasun palamisherkkyyden vuoksi täytyi varmistua siitä, etteivät voiteluöljy ja happikaasu pääse fyysisesti koskettamaan toisiaan, vaan ne on aina erotettu typpikaasulla toisistaan. Toinen tärkeä asia oli happikompressorin laitteiston suojaaminen kaasun takaisinvirtaukselta.

Hapen pyynnin pieneneminen voi aiheuttaa takaisinvirtausta, joka voi vahingoittaa kompressorin siivistöä tai lievemmässä tapauksessa sammuttaa kompressorin. Jotta häiriötön käyttö voitaisiin mahdollistaa tulee ohjauksen ennakoida tuleva pumppaus ja pyrkiä välttämään se. Jos pumppaussuoja ei kykene toimimaan riittävän nopeasti ja ilmenee takaisinvirtausta, saa se kompressorin värähtelemään. Viimeistään värähtelyantureiden tulee sammuttaa kompressorin moottori ja asettaa imu- ja kierrätysventtiili kevennys-asentoon. Jos kompressori sammuu, ulospuhallusventtiili aukaistaan, mutta suljetaan uudelleen paineen laskettua tarpeeksi. Tämän tarkoitus on pitää kompressorin sisäinen paine ylipaineessa ympäristön paineeseen verrattuan, jottei epäpuhtauksia pääse prosessiin.

Pumppaussuojan toiminta saatetaan muuttaa ennen lopullista käyttöönottoa, jos tarvittavat tiedot saadaan hankittua kompressorin valmistajalta. Käyttöönotto tullaan suorittamaan toukokuussa. Ennen käyttöönottoa pyritään valmistelemaan uusia kaapelointeja ja toimilaitteta mahdollisimman pitkälle. Samaan aikaan käyttöönoton kanssa happikompressorin mekaaniset osat tullaan huoltamaan. Käyttöönoton alkaessa happikompressori tullaan puhdistamaan typpikaasulla. Säätöpiirien viritys ja ohjelman varmentaminen oikean laitteiston kanssa tullaan todennäköisesti suorittamaan typellä turvallisuuden takaamiseksi.

LÄHTEET

1. System 800xA with AC 800M Engineering. Kappale 2. Sisäinen dokumentti. ABB Oy.

2. System 800xA 5.1 System Planning. Liite A, sivut 307-310. Sisäinen dokumentti. ABB Oy.

3. System 800xA 5.1 System Planning. Network Setup and Management. Sivut 57-60. Sisäinen materiaali. ABB Oy.

4. System 800xA with AC 800M Engineering. Kappale 7. Sisäinen dokumentti. ABB Oy.

5. System 800xA with AC 800M Engineering. Kappale 7, sivut 1, 3 ja 18. Sisäinen dokumentti. ABB Oy.

6. System 800xA with AC 800M Engineering. Kappale 7, sivu 5. Sisäinen dokumentti. ABB Oy.

7. System 800xA with AC 800M Engineering. Kappale 7, sivu 20. Sisäinen dokumentti. ABB Oy.

8. Product Guide. Control^{IT} Pulp and Paper Control Library. Version 5.0-2 Sivu 4. Sisäinen dokumentti. ABB Oy.

9. System 800xA with AC 800M Engineering. Kappale 4, sivu 3. Sisäinen dokumentti. ABB Oy.

10. System 800xA with AC 800M Engineering. Kappale 4, sivut 15-16. Sisäinen dokumentti. ABB Oy.

11. System 800xA with AC 800M Engineering. Kappale 8, sivu 15. Sisäinen dokumentti. ABB Oy.

12. System 800xA with AC 800M Engineering. Kappale 10, sivu 3. Sisäinen dokumentti. ABB Oy.

13. Wikipedia 2011. Vapaa tietosanakirja. Saatavissa: http://fi.wikipedia.org/wiki/. Hakusanalla: C-kieli. Hakupäivä 20.12.2011

14. System 800xA with AC 800M Engineering. Kappale 9, sivu 3. Sisäinen dokumentti. ABB Oy.

15. System 800xA with AC 800M Engineering. Kappale 14, sivut 3-10. Sisäinen dokumentti. ABB Oy.

16. Grafcet. Evelyne BIEREL & Jean-Marc ROUSSEL. 1995. Saatavissa: http://webserv.lurpa.ens-cachan.fr/grafcet.html. Hakupäivä: 30.12.2011.

17. System 800xA with AC 800M Engineering. Kappale 12, sivut 3, 5, 8-12. Sisäinen dokumentti. ABB Oy.

18. System 800xA with AC 800M Engineering. Kappale 5, sivu 25. Sisäinen dokumentti. ABB Oy.

19. Comsoft. GSD-library. Saatavissa: http://www.comsoft.de/markets/industrial-communication-products/support/gsd-library/. Hakupäivä: 10.2.2012.

20. PI – Profibus Profinet International. GSD-Files. Saatavissa: http://www.profibus.com/index.php?id=127. Hakupäivä: 10.2.2012

21. Airila, Mauri – Hallikainen Keijo – Kääpä, Juha – & Laurila, Timo 1993. Kompressorikirja. Helsinki. Korpivaara Oy Hydor Ab.

22. Allen&Bradley. Allen&Bradley PLC-5 -perhe. Manuaali.

23. Airila, Mauri. 1998. An analytic study of the throttling control methods of liquid-cooled rotary screw air compressors, with a view to minimizing energy demand. Helsinki. Acta polytechnic Scandinavica.

24. Kivioja, Seppo. 2000. Konetekniikka. Espoo. Otatieto.

LIITTEET

Liite 1. Toiminnankuvaus. Paineensäätö

Liite 2. Toiminnankuvaus. Pumppaussuoja.

Etteplan Oyj

Ermo Lohi, Teppo Kivelä.

PAINESÄÄTÖ

11-PC-21800 Kompressorikierrätyksen paineensäätö

PI-KAAVIO 997263

Alla oleva selostus ja kuva suoraan Atlas Copco:n manuaalista kohdasta 5.2.1.2: Ohjausjärjestelmä pystyy käsittelemään useita ohjaussignaaleja, joilla kullakin on oma asetusarvo. Kaikkia signaaleja ei ole mahdollista pitää asetusarvoissa, joten järjestelmä ohjaa yhtä signaalia kerrallaan pitäen kaikki muut asetusarvon turvallisella puolella. Paineensäädön virhesignaalia käytetään suoraan, kaikki muut ovat mitoitettu niin, että ne voidaan korvata poistopaineen virheellä. Poistopaineen ohjaus on aina aktiivinen. Kun muut tilat ovat käytössä, poistopaineen ohjauksesta voi tulla ”backup” kun väärä asetusarvo syötetään tai prosessissa on häiriö.

Toimintatapa
Painesäätö pyrkii pitämään asetellun loppupaineen 11-PI-21800 asetusarvossa.

Kompressorin 2180P loppupainetta ja imupainetta säädetään avaamalla ja sulkemalla venttiiliä 11-HC-21802, jonka kautta kaasua ohjataan kompressorin painelinjasta takaisin imulinjaan. Jos imupaine on liian pieni (11-PI-21792), venttiili 11-HC-21802 avautuu ja päästää kaasua imulinjaan. Jos loppupaine (11-PI-21820) on liian suuri, venttiili 11-HC-21802 avautuu ja päästää kaasua imulinjaan. Kompressorin lukitus 11-XS-21801 asettaa säätäjän manuaalille.

”Kevennys”-tilassa paineensäätäjä 11-PC-21800 ohjaa venttiilin 11-HC-21802 auki 100 %.

Paineensäätäjän toiminta ja säätöparametrit tulee ABB:n selvittää alkuperäisestä Allen & Bradleyn koodista.

Lukitukset
-

Tiedot muihin piireihin
-11-HC-21793Happikompressori imuventtiili, säätö
-11-HC-21802Happikompressori kierrätysventtiili, säätö
-11-HC-21807Happikompressori, ulospuhallusventtiili, säätö
-11-SC-21802 Happikompressori, pumppaussuoja

11-HS-21857 Happikompressori, venttiilien ohjausmoodin valinta

PI-KAAVIO (ei kaaviota)

3 painiketta valvomon näytöllä: ”Kevennys” / ”Auto” / ”Käsi”

”Kevennys”-tilassa: Avaa kierrätysventtiilin 11-HC-21802 ja sulkee imuventtiilin 11-HC-21793. Imuventtiili jää 5 % auki. Kevennys menee automaattisesti päälle jos 11-XV-21650 menee kiinni.

”Auto”-tilassa: Asettaa säätimet automaatille, jolloin järjestelmä säätää venttiileitä painesäädön mukaan sekä siten, ettei kompressori joudu pumppaustilaan.

”Käsi”-tilassa: Imuventtiiliä 11-HC-21793 ja kierrätysventtiiliä 11-HC-21802 voidaan ohjata käsin niiden ohjauskytkimistä.

Lukitukset
-ei

Tiedot muihin piireihin
-11-HC-21793 Happikompressori imuventtiili, säätö
-11-HC-21802 Happikompressori kierrätysventtiili, säätö
-11-HC-21807 Happikompressori, ulospuhallusventtiili, säätö
-11-PC-21800 Kompressorikierrätyksen paineensäätö
-11-SC-21802 Happikompressori, pumppaussuoja

11-HI-21802 Happikompressori kierrätysventtiili, asento 0-100 % (11-HV-21802)

11-HC-21802 Happikompressori kierrätysventtiili, säätö (11-HC-21802)

PI-KAAVIO 997263

Venttiilin fyysisesti sijaitsee happikompressorin melusuojassa. 4mA = Auki / 20mA = Kiinni

Asentotiedosta AI-tieto järjestelmään 4…20mA.
Säätöviesti AO-tieto järjestelmästä 4…20mA

Toimintatapa
Venttiili säätää happilinjan kierrätyslinjan virtausta.

Venttiilin toiminta eri ohjausmoodeissa:
”Käsi”-tilassa: kierrätysventtiiliä 11-HC-21802 voidaan ohjata auki tai kiinni valvomon näytöltä säätimen omasta kuvakkeesta.

”Auto”-tilassa: Paineensäätäjä 11-PC-21800 ja pumppaussuoja 11-SC-21802 ohjaa venttiiliä.

”Kevennys”-tilassa: Paineensäätäjä 11-PC-21800 ohjaa venttiilin auki 100 %. Painelinjan venttiilin 11-XV-21650 sulkeutuessa rajatietoviestiä käytetään ajamaan kompressorin kierrätysventtiili arvoon 100 % auki. Tällöin kompressori kevenee ja

pysyy kevennettynä 3 minuuttia, jonka jälkeen painesäätö ajaa kompressorin asetusarvoonsa. Toimenpiteellä ennakoidaan pienentyvää tilavuusvirtaa ja alitetaan kaasun takaisinkierrätys ajoissa, jotta vältytään pumppaussuojan toimimiselta.

Lukitukset
-11-PC-21800 Kompressorikierrätyksen paineensäätö
-11-XV-21650 Hapen syötön pikasulku

Tiedot muihin piireihin
-11-XS-21801 Kompressorin käynnistysehto
-11-PC-21800 Kompressorikierrätyksen paineensäätö
-11-SC-21802 Happikompressori, pumppaussuoja

11-HI-21807 Happikompressori, ulospuhallusventtiili, asento 0-100 % (11-HV-21807)

11-HC-21807 Happikompressori, ulospuhallusventtiili, säätö (11-HC-21807)

PI-KAAVIO 997263

Venttiilin fyysisesti sijaitsee happikompressorin melusuojassa. 4mA = Auki / 20mA = Kiinni

Asentotiedosta AI-tieto järjestelmään 4…20mA.
Säätöviesti AO-tieto järjestelmästä 4…20mA

Toimintatapa
Venttiili säätää happilinjan ulospuhalluslinjan virtausta.

Venttiilin toiminta eri ohjausmoodeissa:
”Käsi”-tilassa: Ulospuhallusventtiiliä 11-HC-21807 voidaan ohjata auki tai kiinni valvomon näytöltä säätimen omasta kuvakkeesta.

”Auto”-tilassa: Paineensäätäjä 11-PC-21800 ja pumppaussuoja 11-SC-21802 ohjaa venttiiliä.

”Kevennys”-tilassa: Paineensäätäjä 11-PC-21800 ohjaa venttiilin auki 100 %.

Ulospuhallusventtiili avautuu aina kompressorin pysähtyessä ja sulkeutuu heti, kun paine kierrätyslinjassa laskee alle asetusarvon (0,32bar). Näin kompressorikiertoon jää lievä ylipaine, jolla estetään partikkeleiden, ilma jne. pääsy kompressorin ja sen kierron sisälle.

Lukitukset
-11-PC-21800 Kompressorikierrätyksen paineensäätö

Tiedot muihin piireihin

-11-PC-21800 Kompressorikierrätyksen paineensäätö
-11-SC-21802 Happikompressori, pumppaussuoja
-11-XS-21801 Kompressorin käynnistysehto

Etteplan Oyj
Ermo Lohi, Teppo Kivelä

PUMPPAUSSUOJA

Pumppaustila:

Pumppaus on luonteeltaan painepuolen paineen vaihtelua. Tyypillisesti pulssien väli on 0,25…5 sekuntia. Pumppaustila on tärinänomainen käyttötapahtuma ja saattaa aiheuttaa vaurioita roottorin siipiin. Jos valvontajärjestelmä huomaa pumppausta, koneen kuormitus laskee automaattisesti ja jää siihen tilaan kunnes kuittaus-painiketta painetaan.

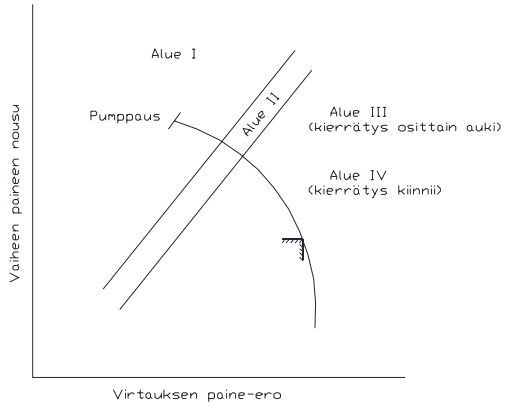
11-SC-21802 Happikompressori, pumppaussuoja

PI-KAAVIO 997263

Pumppaussuoja valvoo kompressorin painesuhdetta pumppauskäyrällä. Painesuhteen täytyy pysyä pumppauskäyrän turvallisella puolella.

Pumppaussuojan toiminta ja säätöparametrit tulee ABB:n selvittää alkuperäisestä Allen & Bradleyn koodista.

Alla oleva selostus ja kuva suoraan Atlas Copco:n manuaalista kohdasta 5.2.1.1: Pumppaustilaa valvotaan kompressorin toisen vaiheen painemittauksilla. Kun kompressori saavuttaa pumppauskäyrän kontrolliviivan, prosessin säätö vaihdetaan imuventtiililtä kierrätysventtiilille. Kun kierrätysventtiili on osittain auki, pumppaussuoja tarvittaessa säätää imuventtiiliä pitääkseen kompressorin poissa pumppausalueelta, mutta lähellä kontrolliviivaa. Kuvassa 1 on kaavio kompressorin alueista ja miten venttiilit toimivat kullakin alueella



Kuva 1

Alue I: Kompressori toimii pumppauskäyrän vasemmalla puolella (pumppausvirhe negatiivinen). Tämä on lähin alue ennen varsinaista pumppaustilaa.
-imuventtiili avautuu tarvittaessa säätimen virheeseen perustuen
-kierrätysventtiili avautuu pumppausvirheeseen perustuen

Alue II: Kompressori toimii pumppauskäyrän oikealle puolella, mutta kuolleen alueen sisällä
-imuventtiili avautuu jos kierrätysventtiili täysin auki, säätimen virheeseen perustuen
-kierrätysventtiili ohjaa säätimen virhettä

Alue III: Kompressori toimii hyvin, kierrätysventtiili osittain auki
-imuventtiili sulkeutuu pumppaussuojan säätöön perustuen
-kierrätysventtiili ohjaa säätimen virhettä

Alue IV: Kompressori toimii hyvin, kierrätysventtiili kiinni
-imuventtiili ohjaa säätimen virhettä
-kierrätysventtiili pysyy kiinni

Lukitukset
-

Tiedot muihin piireihin
-11-XS-21800 Kompressorin lukitusehto
-11-PC-21800 Kompressorikierrätyksen paineensäätö

11-FI-21880 Happikompressori, vaihe 2 määrä (PDT320)

PI-KAAVIO 997263

Toimintatapa

Painemittaus, mittaa O2-linjan virtausta vaiheessa 2, Painelähetin 4…20mA, x...xxx Yksikkö?
AI-tieto järjestelmään.

Käytetään pumppaussuoja pumppauskäyrän laskentaan.

TARKISTA MITTAUKSEN SKAALAUS JA YKSIKKÖ (vanha 0-250cm)?

Lukitukset
-ei

Tiedot muihin piireihin
-11-SC-21802 Happikompressori, pumppaussuoja