



SAVONIA

OPINNÄYTETYÖ - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

JÄTEVEDEN LAMELLI- PUHDISTIMEN AUTOMAATIO

TEKIJÄ: Iiro Nevalainen

Koulutusala Tekniikan ja liikenteen ala			
Koulutusohjelma Sähkötekniikan koulutusohjelma			
Työn tekijä Iiro Nevalainen			
Työn nimi Jäteveden lamellipuhdistimen automaatio			
Päiväys	14.12.2016	Sivumäärä/Liitteet	36/31
Ohjaaja(t) diplomi-insinööri Risto Rissanen, lehtori Jari Ijäs			
Toimeksiantaja/Yhteistyökumppani(t) Pure Water Finland Oy			
Tiivistelmä			
<p>Opinnäytetyön tavoitteena oli suunnitella ja rakentaa automaatiojärjestelmä jäteveden lamellipuhdistimen prosessin ohjaukseen. Pure Water Finland Oy tilasi Savonia-ammattikorkeakoululta jäteveden lamellipuhdistimen prototyypin suunnittelun ja rakentamisen. Prototyyppi toimisi uudenlaisen jätevedenpuhdistusmenetelmän testausalustana.</p> <p>Projektin alkuvaiheessa selvitettiin, millaisia laitteita prototyyppiin tulee ja millaiset ovat näiden laitteiden automaation tarpeet. Prototyypin laitteiston ohjaukseen tulisi rakentaa automaatiojärjestelmä jonka käyttö olisi yksinkertaista ja turvallista. Laitteistoa ohjaamaan valittiin Siemensin s300-sarjan ohjelmoitava logiikka (PLC), jonka avulla laitteiston automatisointi toteutettiin. PLC ohjelmoitiin käyttäen Siemensin Step7-ohjelmistoa. PLC:n jatkoksi liitettiin kaksi eri kenttäväylää, AS-i ja Profibus DP, joiden avulla kommunikointi kenttälaitteiden kanssa. Järjestelmän käyttöliittymänä toimi Siemensin KTP600-kosketusnäyttöpaneeli, jonka avulla käyttäjä antaa automaatiolaitteistolle komentoja ja valvoo järjestelmän toimintaa. Käyttöliittymä ohjelmoitiin käyttämällä Siemensin WinCC-ohjelmistoa ja liitettiin logiikkaan käyttäen Profibus-kenttäväylää. Automaatiojärjestelmän lisäksi rakennettiin myös hätä-seis-piiri laitteiston turvallisen käytön varmistamiseksi.</p> <p>Opinnäytetyön tuloksena rakennetun automaatiojärjestelmän avulla prosessin ohjaus on mahdollista kosketusnäytön avulla. Prototyyppi ei valmistunut opinnäytetyön aikana, vaan sen rakentamista ja kehitystä jatkettiin. Opinnäytetyön tuloksena rakennetun järjestelmän avulla prototyypin toimintaa voidaan testata ja kehittää. Rakennettuun automaatiojärjestelmään on mahdollista lisätä toiminnallisuutta jatkokehityksessä.</p>			
Avainsanat Jätevesi, lamellipuhdistin, automaatio, PLC, HMI, kenttäväylät, AS-i, Profibus			

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Electrical Engineering			
Author Iiro Nevalainen			
Title of Thesis Automation of Wastewater Lamella Purifier			
Date	14 January 2016	Pages/Appendices	36/31
Supervisor(s) Mr. Risto Rissanen, Master of Science (Technology) Mr. Jari Ijäs, Lecturer			
Client Organisation /Partners Pure Water Finland Oy			
<p>Abstract</p> <p>The aim of this thesis was to design and build an automation system for the wastewater lamella purifier. Pure Water Finland Ltd commissioned Savonia University of Applied Sciences to design and build a prototype of the wastewater lamella purifier. The prototype was to be a testing platform for a new type of waste water purification method.</p> <p>The project began by finding out what kind of machinery the prototype would have and what kind of automation the machinery would need. A simple and safe to operate automation system needed to be built to control the machinery of the prototype. The Siemens s300-series programmable logic controller (PLC) was selected for the automation of the machinery. The PLC was programmed using the Siemens Step7 software. AS-i and Profibus fieldbuses were used in the extension of the PLC to communicate with field devices. The Siemens KTP600 touch panel was used as a user interface (HMI) which allows the user to give commands and monitor the operation of the automation system. The HMI was programmed using the Siemens WinCC software and connected to the PLC through Profibus. An emergency stop circuit was built in addition to the automation system to ensure the safe operation of the system.</p> <p>As a result of this thesis, an automation system was built which allows process control over a touch panel. The prototype was not finished during the time of making this thesis. The development and construction of the prototype was continued. The automation system built as a result of this thesis made the testing and developing of the prototype possible. In the future, more functionalities can be added to the built automation system.</p>			
Keywords Wastewater, Lamella purifier, automation, PLC, HMI, Field Bus, AS-i, Profibus			

ESIPUHE

Opinnäytetyössä *Jäteveden lamellipuhdistimen automaatio* pääsin tutustumaan automaatiojärjestelmien suunnitteluun ja rakentamiseen. Työn tilasi Pure Water Finland Oy Savonia-ammattikorkeakoululta. Haluan kiittää Pure Water Finland Oy:tä ja tilaajan edustajaa Risto Kempaista mielenkiintoisesta opinnäytetyöaiheesta ja DI Risto Rissasta ja lehtori Jari Ijästä opinnäytetyön ohjaamisesta.

Lappeenrannassa 14.12.2016

Iiro Nevalainen

SISÄLTÖ

1	JOHDANTO	7
2	VEDENPUHDISTUSPROSESSIN KUVAUS	8
3	LAITTEIDEN OHJAUS.....	10
4	AUTOMAATIO	11
4.1	Ohjelmoitavat logiikat.....	11
4.2	Käyttöliittymä	12
4.2.1	AS-i.....	12
4.2.2	Profibus.....	14
4.3	Lamellien puhdistusjärjestelmä	16
4.4	Käyttöliittymä	17
4.5	Virransyöttö.....	17
4.6	Virran mittaus.....	18
4.7	Poistopuhallin	20
4.8	Taajuusmuuttajat.....	21
4.9	Hätä-seis-piiri	22
5	AUTOMAATION OHJELMOINTI	24
5.1	Ohjelmat	26
5.1.1	Moottorin start-stop-piiri	27
5.1.2	Lamellien puhdistusohjelma	29
5.2	Käyttöliittymän ohjelmointi	31
5.2.1	Kontaktorilähtöjen ohjaus	32
5.2.2	Venttiiliohjaus	34
6	YHTEENVETO.....	36
	LÄHTEET JA TUOTETUT AINEISTOT	37
	LIITTE 1: PLC-OHJELMISTO	38
	LIITE 2: AUTOMAATION PIIRIKAAVIOT.....	65

KÄSITTEET

PLC (Programmable logic controller) = Ohjelmoitava logiikkaohjain

Lamelli = Ohut levy tai liuska

Step7 = Siemensin logiikkaohjainten ohjelmointiympäristö PC:lle

WinCC = Siemensin kosketuskäyttöliittymien ohjelmointiympäristö PC:lle

S7-300 = Siemensin logiikkaohjaintuoteperhe

Input = Tulo

Output= Lähtö

I/O (Input/Output) = Tulo/lähtö

AS-i (AS-interface) = Kenttäväylä

Profibus DP = Kenttäväylä

HMI (Human-machine interface) = Käyttöliittymä

PSU (Power Supply) = Virtalähde

IC (Integrated circuit) = Integroitu virtapiiri

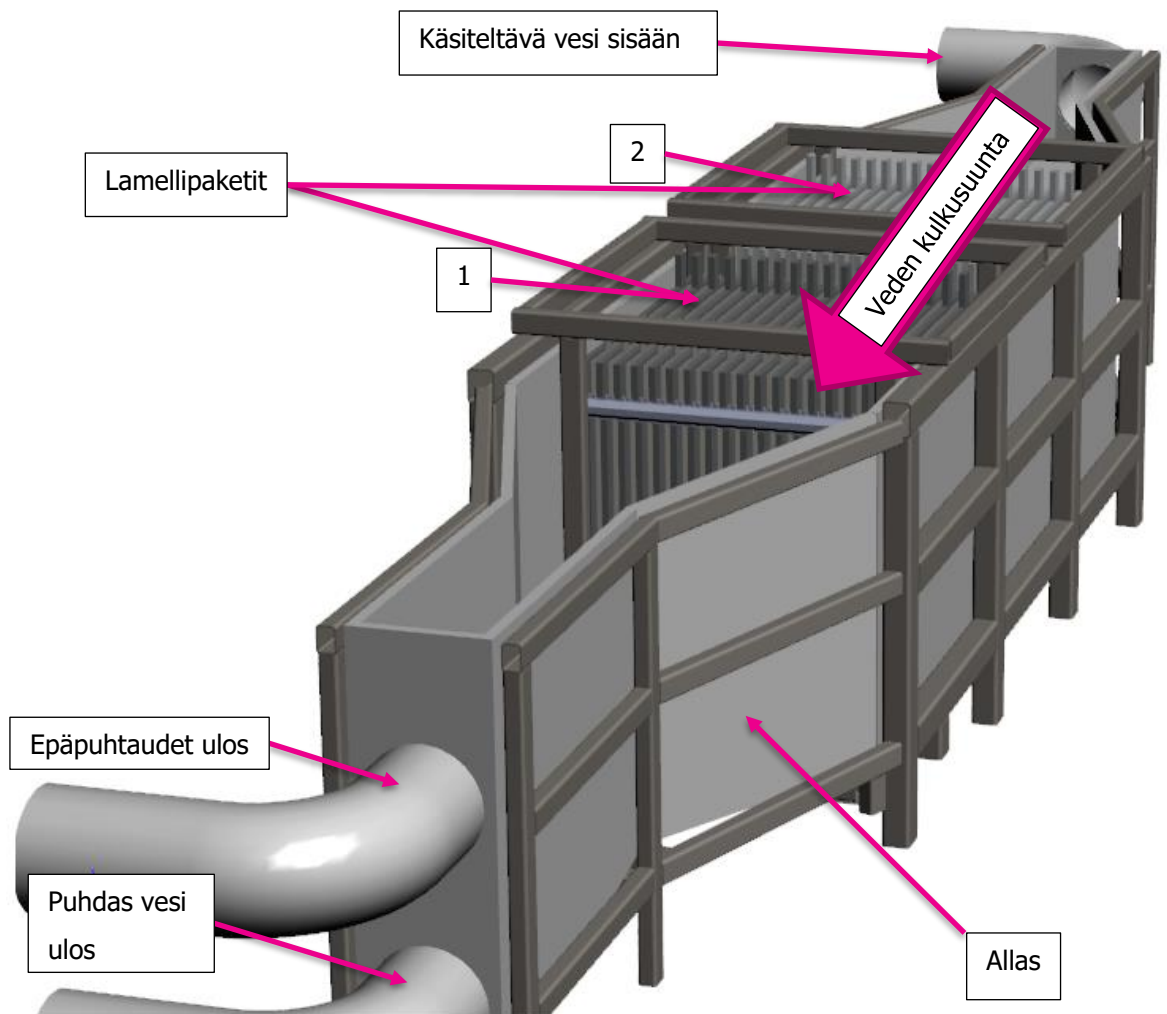
1 JOHDANTO

Opinnäytetyössä tavoitteena on suunnitella ja rakennetaa jäteveden lamellipuhdistimen automaatiojärjestelmä käyttäen Siemensin S7-300-sarjan ohjelmoitavia logiikoita sekä yhteensopivaa kosketuskäyttöliittymää. Automaatiojärjestelmän suunnittelu on osa projektia, jossa Pure Water Finland Oy tilasi Savonia-ammattikorkeakoululta jäteveden lamellipuhdistimen automaation suunnittelun ja rakentamisen. Jäteveden lamellipuhdistin hyödyntää täysin uudentyypistä vedenpuhdistusmenetelmää. Tavoitteena on suunnitella automaatiojärjestelmä, jonka avulla ohjataan puhdistusprosessissa tarvittavia laitteita. Prosessia pyritään automatisoimaan, jotta sen hallinta olisi helpompaa ja turvallista. Työn tilauksen tehneen Pure Water Finland Oy:n puolesta yhteyshenkilönä toimii Risto Kempainen.

2 VEDENPUHDISTUSPROSESSIN KUVAUS

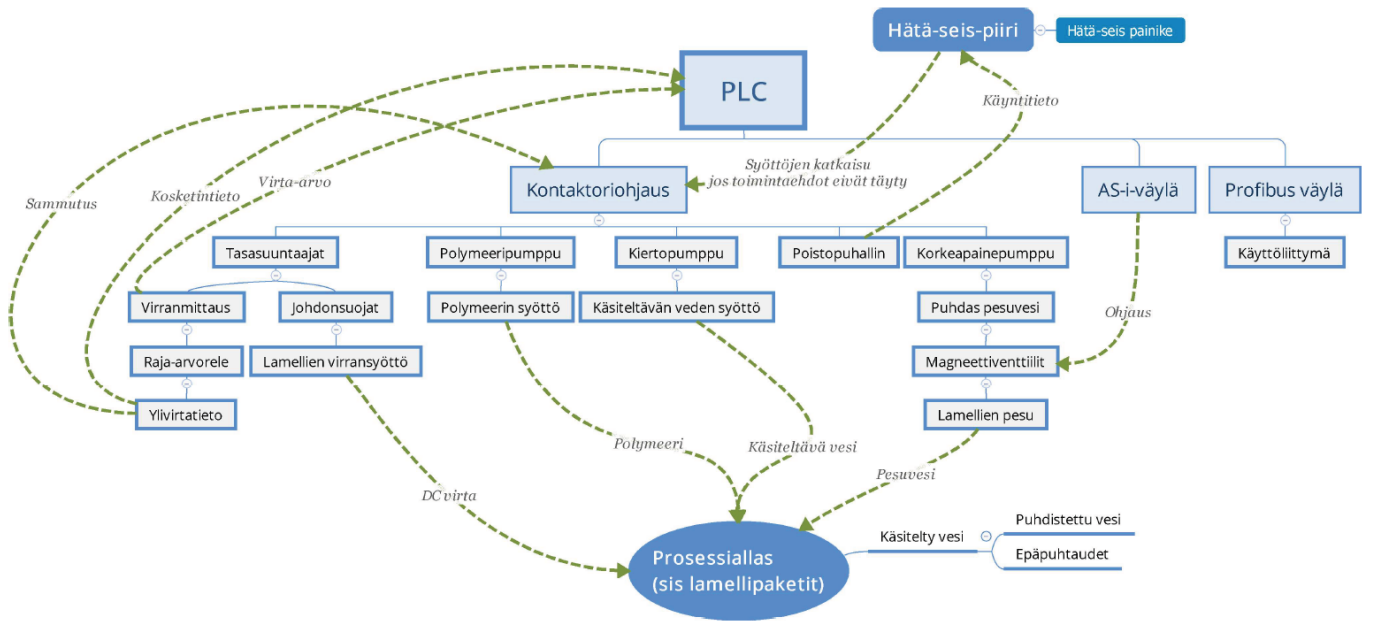
Tämä opinnäytetyö keskittyy vain vedenpuhdistuslaitoksen automaatioon, mutta kontekstiksi on hyvä ymmärtää vedenpuhdistusprosessin perusteet.

Yksinkertaistettuna prosessi toimii siten, että altaan päädyssä tuodaan käsiteltävä vesi altaaseen ja siihen sekoitetaan polymeeriä. Seos virtaa seuraavaksi lamellipakettien läpi, jolloin lamellien välillä kulkeva sähkövirta saa polymeerin tarttumaan veden epäpuhtauksiin. Polymeeriin sidotut epäpuhtaudet flokkautuvat pintaan ja altaan pohjalle jää vain puhdas vesi. Altaan toisessa päädyssä puhdas vesi ja epäpuhtaudet johdetaan omia kanaviaan pitkin ulos.



KUVA 1 Allaskokonaisuus (Kuva, Iiro Nevalainen)

Kuvassa 1 näkyviin lamellipaketteihin johdetaan tasavirtaa ja käsiteltävä vesi virtaa niiden lävitse. Lamellipaketti koostuu metallilevyistä, joista joka toinen on terästä ja joka toinen rautaa (paketti 1) tai alumiinia (paketti 2).

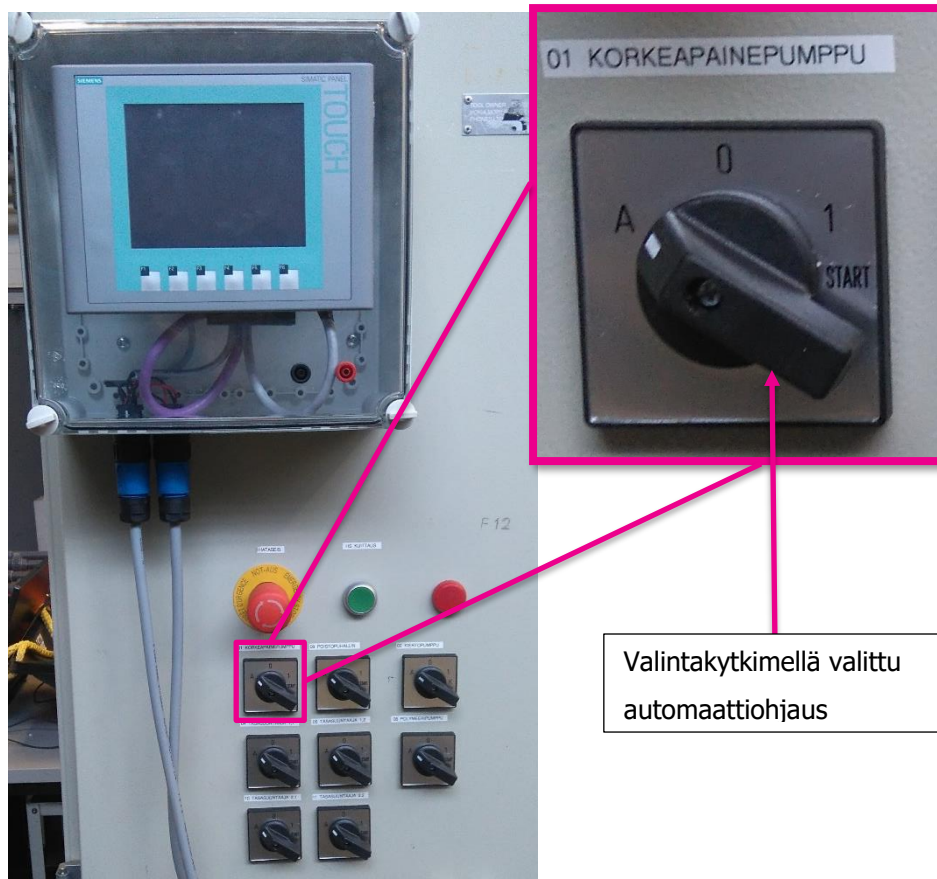


KUVA 2 Havainnekuva prosessista (Kuva, Iiro Nevalainen)

Kuva 2 havainnollistaa prosessin toimintaa, siihen liittyviä laitteita ja niiden tehtäviä.

3 LAITTEIDEN OHJAUS

Keskukseen suunniteltiin 11 kontaktoriohjattua lähtöä, joita voidaan ohjata käsin tai logiikan kautta. Syy ratkaisuun on käytön helppous. Käsiohjaus on hyödyllinen laitteiston nopeaan testaukseen ja diagnostiikkaan. Logiikan kautta ohjaaminen mahdollistaa prosessin automatisoinnin.



KUVA 3 Laitteiden ohjaus (Valokuva Iiro Nevalainen)

Haluttu laite voidaan ohjata käyntiin manuaaliohjauksella kääntämällä valintakytkin (kuva 3) asentoon start ja vapauttamalla kytkin, jolloin se palaa asentoon 1. Toiminto muistuttaa auton käynnistämistä avaimella. Manuaaliohjaus ohjaa suoraan kontaktorin pitopiiriä. Jos hätä-seis-piiri sammuttaa laitteen, tulee laite käynnistää uudelleen käyttämällä valintakytkintä uudelleen asennossa start. Tämä toiminta on suunniteltu vastaamaan konedirektiivin vaatimuksia. Hätä-seis-piiristä on kerrottu lisää luvussa 4.9 Hätä-seis-piiri. Valintakytkin asennossa A tarkoittaa automaattiohjausta. Automaattiohjauksessa PLC ohjaa laitetta ja käyttäjä syöttää komentoja kosketusnäytön avulla. PLC:n lähtö ohjaa kontaktoria releen välityksellä. Releen kela toimii PLC:lle sopivalla 24 VDC jännitteellä. Rele ohjaa edelleen kontaktoria, jonka kela toimii 220 VAC jännitteellä.

4 AUTOMAATIO

4.1 Ohjelmoitavat logiikat

Ohjelmoitava logiikka eli PLC on laite, joka suorittaa muistissaan ennalta määriteltyjä komentoja, kuten logiikkoja ja ajastimia. Logiikkaohjain koostuu prosessointiyksiköstä ja I/O-moduuleista.

Projektissa päädyttiin käyttämään Siemensin S7-300-logiikoita, koska logiikan komponentteja oli saatavilla Savonialta. Logiikka ohjelmoitiin Siemensin Step7 5.5-ohjelmistoa. Prosessoriksi logiikkaan valikoitui S7-314 PN/DP, koska se sisälsi valmiuden PID-säätimelle, jota tarvitaan prosessissa mm. virtauksen säätöön.



KUVA 4 I/O-moduulit (Valokuva Iiro Nevalainen.)

Työssä käytettiin seuraavia I/O-moduuleita:

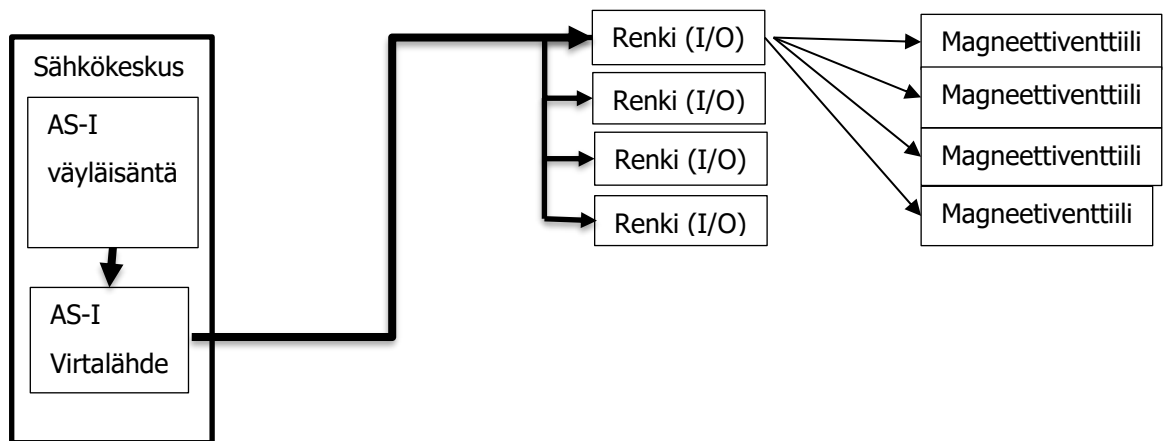
- CP 342-5, Profibus-liitäntämoduuli
 - CP 342-2, AS-i-kenttäväylän liitäntämoduuli
 - SM 321, digitaalinen tulomoduuli, 16 kanavaa
 - SM 322, digitaalinen lähtömoduuli, 16 kanavaa
 - SM 334 (2 kpl), analoginen I/O-moduuli, 4 tulokanavaa ja 2 lähtökanavaa, 8-bittinen resoluutio
- I/O moduulit kuvassa 4.

4.2 Kenttäväylät

Projektissa käytettiin kahta kenttäväylää. AS-i-väylää käytettiin kaapeloinnin vähentämiseksi keskussta kenttälaitteille eli magneettiventtiileille. Profibus DP-väylää käytettiin datan siirtämiseksi HMI:n ja PLC:n välillä. Profibus DP- ja AS-i-väylillä on omat käyttötarkoituksensa eikä niitä ole suunniteltu toimimaan samoissa tehtävissä. AS-i-väylä on suunniteltu toimimaan alemman tason kenttäväylänä kyvykkäämmän kenttäväylän, kuten Profibus DP:n tukena. AS-i-väylä soveltuu parhaiten kommunikointiin yksinkertaisten I/O-laitteiden kanssa. Profibus DP on ylemmän tason kenttäväylä, jonka avulla kommunikaatio ylemmän tason laitteiden, kuten logiikkaohjainten, välillä on mahdollista.

4.2.1 AS-i

Yksinkertainen AS-i-väylä koostuu AS-i-virtalähteestä (kuva 6), väyläisännästä ja rengistä (I/O-moduuli). Väyläisäntä toimii välikätenä ja kommunikoi I/O-moduulien ja ylemmän tason logiikan, kuten PLC:n kanssa. Väylässä on käytettävä juuri AS-i-väylään tarkoitettua virtalähdettä, joka syöttää virtaa väylään häiritsemättä väylässä kulkevaa signaalia. I/O-moduulit voivat sisältää neljä tuloa ja/tai neljä lähtöä. I/O-moduulit voivat itsessään olla kenttälaitteita, kuten etäisyysantureita, jotka liitetään suoraan väylään, tai relelähtöjä joihin voidaan liittää perinteisiä kenttälaitteita kuten magneettiventtiileitä. Projektissa käytettyyn vanhempaan AS-i-väylään oli mahdollista liittää 31 I/O-moduulia.



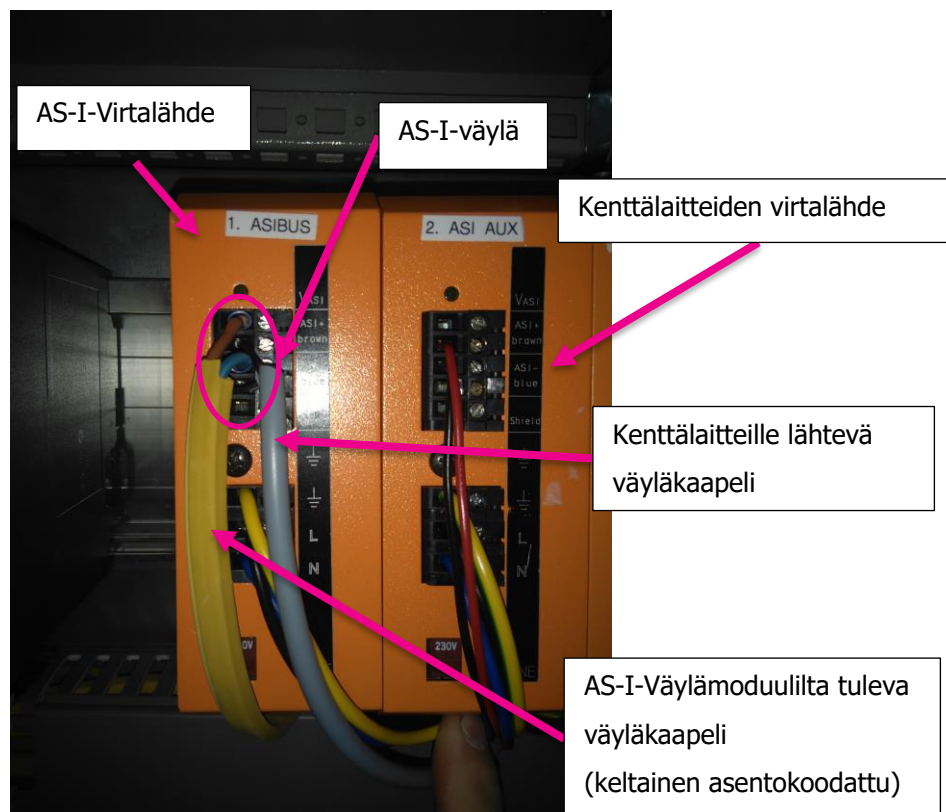
KUVA 5 AS-i-väylän rakenteen havainnollistaminen

Kuva 6 havainnollistaa AS-i-väylän puumaista rakennetta. Vaihtoehtoisesti väylä voidaan rakentaa myös tähdeksi. Väylä voidaan jakaa mistä tahansa kohdasta ja niin moneksi haaraksi, kuin on tarve.

AS-i-väylässä kulkee kahdenlaisia viestejä: väyläisännän lähettämiä viestejä ja väyläorjien lähettämiä vastauksia. Väylässä kulkevat viestit koostuvat väyläisännän lähettämästä pyynnöstä, väyläisännän tauosta, rengin vastauksesta ja rengin tauosta. Väyläisännän pyynnöt ovat pituudeltaan 14 bitin ajan pituisia ja renkien vastaukset 7 bitin ajan pituisia. Bitin aika vastaa 6 μ s:a. Tätä kutsutaan pollaavaksi isäntä-renki-kommunikaatioksi. Isännän tauko kestää vähintään 2 ja enintään 10 bitin aikaa. Jos isäntä ei saa vastausta 10 bitin ajassa, olettaa se ettei vastausta ole tulossa, ja se siirtyy seura-

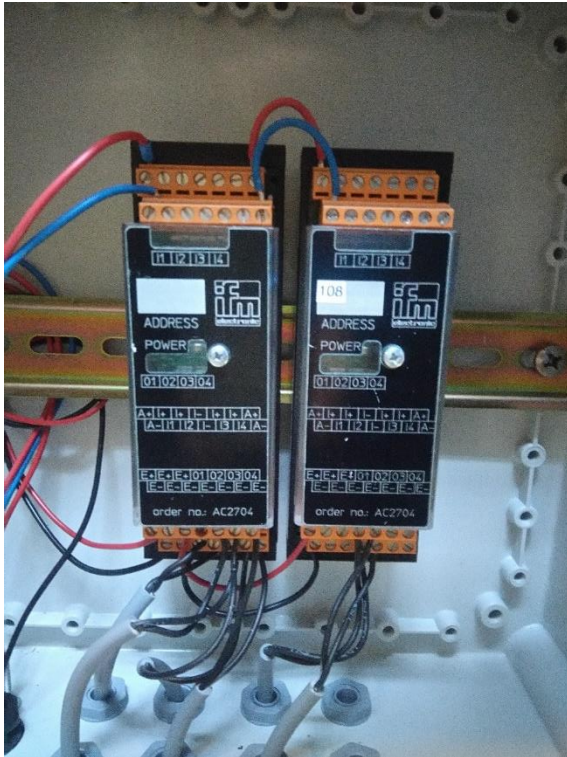
vaan pyyntöön. Väylässä liikkuva data on rajoitettu neljään bittiin per renki per sykli, joten. Suuremmat viestit joudutaan jakamaan useaan sykliin. (SMAR, 2016) AS-i-väylän rengin vasteaika on kohtalaisen lyhyt, parhaimmillaan 5 ms, (Siemens, 2016) joten se soveltuu nopeisiin automaation tarpeisiin. Väylä soveltuu parhaiten yksinkertaisten on-off-tyyppisten laitteiden kanssa kommunikointiin. AS-i-väylä soveltuu huonosti suurten datamäärien nopeaan siirtoon mm. siksi että väylässä liikkuvat viestit on rajoitettu neljään bittiin per laite. AS-i-väylä on avoin standardi, joten laitteita ja laitevalmistajia on paljon. Tästä syystä laitteet ovat kohtuuhintaisia ja niitä on hyvin saatavilla.

Venttiiliohjauksessa käytettiin AS-i-väylää, koska se helpotti kaapelointia ja vähensi logiikan output porttien tarvetta. AS-i-väylään kytkettiin I/O-moduuleja, joissa jokaisessa on neljä input ja output porttia. Moduulien ulostuloihin kytkettiin magneettiventtiilit, joita ohjaamalla korkeapaineinen vesiruisku puhdistaa lamellit. AS-i-väylämoduulien käyttöjännite ja data kulkevat samassa kaksijohtimessa kaapelissa. Väylän alku on S7-300-väylämoduulissa, josta se kulkee virtalähteen kautta I/O-moduuleille. Väyläkaapeli on tyypillisesti asentokoodattua keltaista parikaapelia, mutta käytännössä mikä tahansa poikkipinnaltaan vähintään 0,5 mm²:n kaapeli käy. Kuvassa 6 nähdään asentokoodattu keltainen väyläkaapeli, sekä tavallinen 0,5 mm²:n parikaapeli väyläkaapeleina.



KUVA 7 AS-i-virtalähteet (Valokuva Iiro Nevalainen)

Kenttälaitteet eli tässä tapauksessa magneettiventtiilit tarvitsevat oman käyttöjännitteensä, joka tuotetaan erillisellä virtalähteellä (kuva 6). Kaikki väylämoduulien ja kenttälaitteiden tarvitsema virta ja data voidaan siis viedä nelinapaisella kaapelilla sähkökeskukselta kenttälaitteille.



KUVA 8 AS-i-I/O-moduuli venttiiliohjaukseen (Valokuva Iiro Nevalainen)

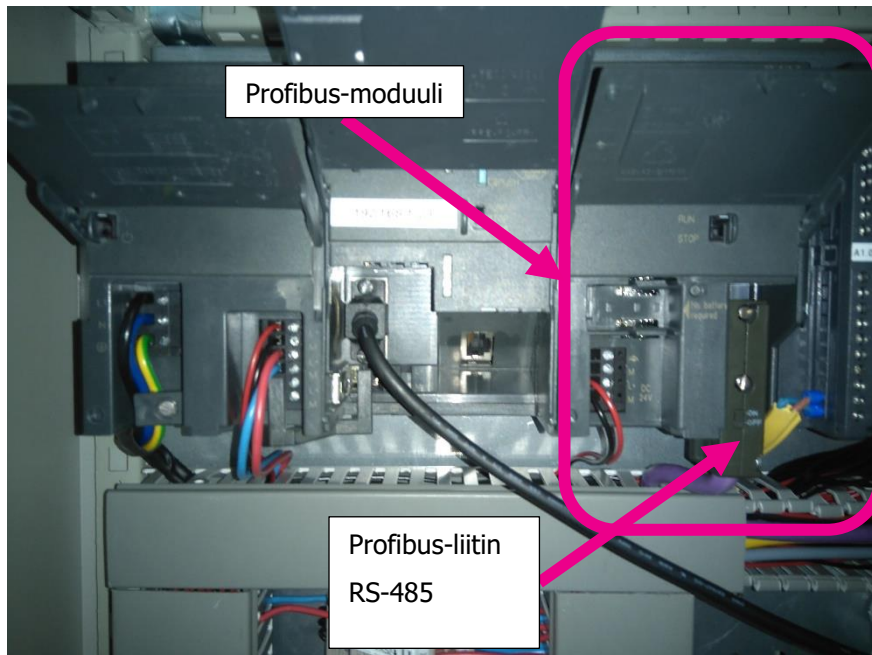
Kenttäkeskukseen tulee siis 30 V tasajännite ja AS-i-väylä. Väylä on kytketty moduulin yläosaan ja venttiileiden käyttöjännite 30 V moduulin alaosaan (kuva 7).

4.2.2 Profibus

Kenttäväylänä Profibus on AS-i-väylää huomattavasti monipuolisempi ja sen seurauksena myös kalliimpi. Projektissa käytetty Profibus DP (Decentralized Peripherals, hajautetut laitteet) on vain yksi Profibus-väylän tyypeistä. Myös Profibus DP:stä on useita versioita, joista käytettiin Profibus DPV0-versiota. Kommunikaatio tässä väylässä tapahtuu syklisesti siten, että väylän isäntä lähettää pyynnön väylälle ja väylässä oleva renki vastaa siihen. Koska kaikki väylässä olevat laitteet ”kuulevat” kaikki viestit, on väylän kaikille laitteille annettava osoite, jotta tiedetään mille laitteelle viesti on osoitettu. Kaikkiaan väylään voidaan kytkeä kerralla 125 laitetta. Jos väylässä on useita isäntä-laitteita, tapahtuu kommunikointi token-periaatteella. Väylässä ensimmäisenä oleva isäntä-laite aloittaa kommunikointisyklin ja ollessaan valmis luovuttaa tokenin seuraavalle isännälle. Viimeisen isäntä-laitteen ollessa valmis siirtyy token takaisin ensimmäiselle isännälle. Profibus DP:n siirtonopeus on 9,6 kbit/s – 12 Mbit/s. Kaikille väylän laitteille on valittava sama siirtonopeus. Väylän vasteaika riippuu kytkettyjen laitteiden määrästä ja käytetystä siirtonopeudesta, mutta se on yleensä korkeintaan

10 ms, jopa 1 ms. Väylän data liikkuu violetissa suojatussa parikaapelissa, joka on kytketty laitteisiin RS-485-liittimellä (kuva 9).

(National Instruments, 2010; PAControl.com, 2002)

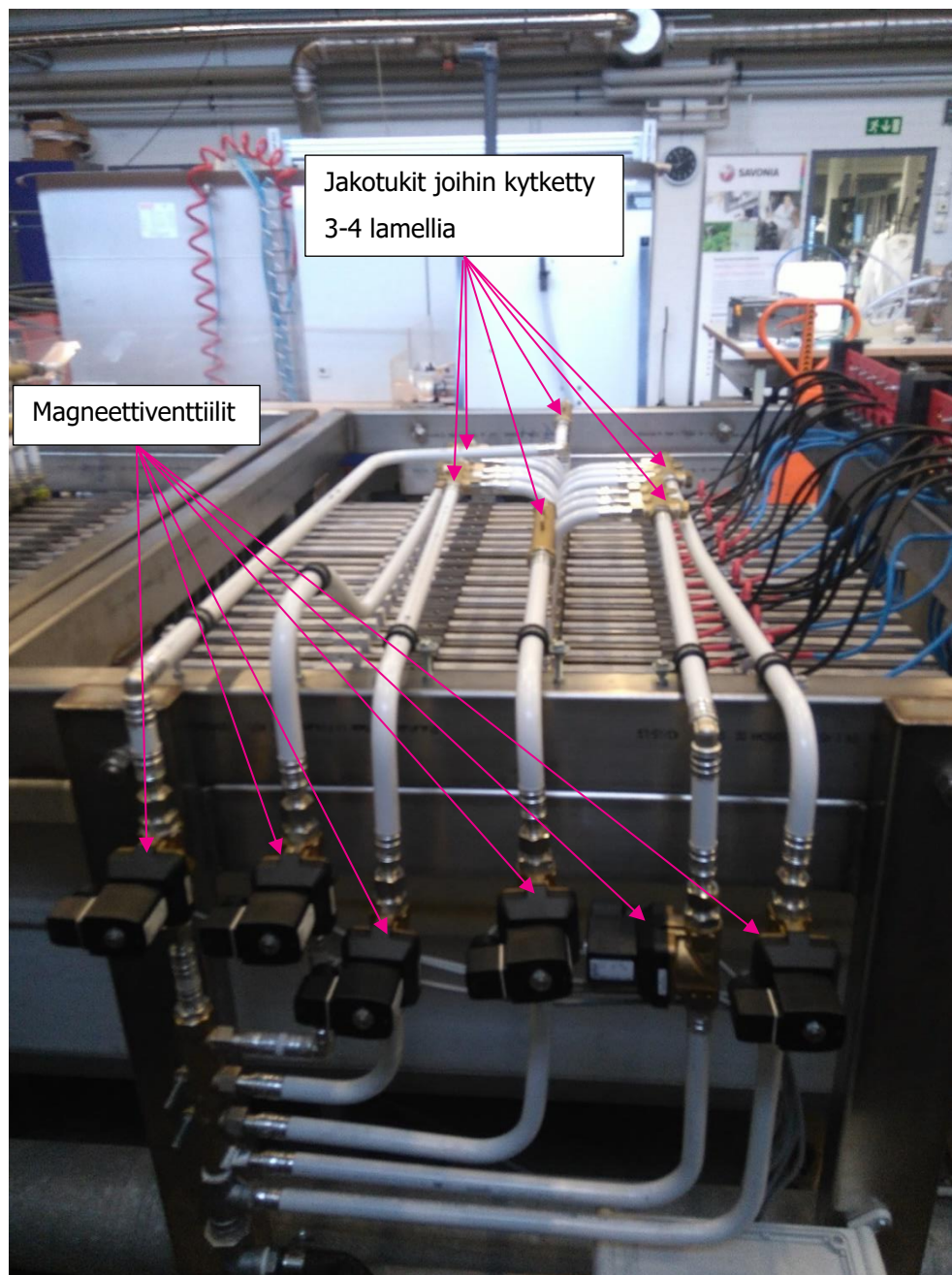


KUVA 9 Profibus (Valokuva Iiro Nevalainen)

Profibus DP soveltuu hyvin nopeaan tiedonsiirtoon kentälaitteiden kanssa. Nopean tiedonsiirron ansiosta se soveltuu mm. taajuusmuuttajien kanssa kommunikointiin, jolloin on tarve kommunikoida kentälaitteen kanssa lähes reaaliajassa. Saatavilla on useita taajuusmuuttajia jotka ovat yhteensopivia Profibus DP:n kanssa, joka helpottaa niiden kanssa kommunikointia ja vähentää kaapelointia. Jos taajuusmuuttajan kanssa kommunikoidaisiin ns. perinteisin keinoin, tapahtuisi kaikki kommunikaatio analogisia signaaleja käyttäen ja virhesignaalin mahdollisuus olisi huomattavasti suurempi. Lisäksi tarvittaisiin useita johtimia PLC:n ja taajuusmuuttajan välillä. Profibus DP:n kautta kommunikaatio tapahtuu digitaalisesti yhtä kaapelia käyttäen, jolloin kaapelointi vähenee ja virhesignaalin mahdollisuus pienenee huomattavasti. Väylää käyttämällä taajuusmuuttajalle voidaan antaa asetusarvoja ja saada takaisin toimintataajuus ja virheviestit jne. Profibus DP ei ole järkevä valinta kenttäväyläksi, jos on tarve yksinkertaisen on-off I/O:n hajauttamiseen. Yksinkertaisen I/O-laitteiston hajauttamiseen Profibus on liian järeä ja turhan kallis verrattuna alemman tason väyliin kuten AS-i:n.

4.3 Lamellien puhdistusjärjestelmä

Prosessin aikana lamellien pinnalle kertyy epäpuhtauksia, jotka täytyy puhdistaa mekaanisesti. Tästä syystä teräslamellit on rakennettu ontoiksi. Ontto lamelli koostuu kahdesta teräslevystä ja niiden väliin reunoille hitsatusta rimasta. Levyjen yläosassa on liittymäkohta vesiputkelle, josta lamellin sisään pumpataan korkealla paineella vettä (kuva 9). Vesi ruiskuaa poratuista rei'istä ulos ja puhdistaa lamellien pinnat. Vettä ei kuitenkaan pumpata kaikkiin lamelleihin yhtä aikaa, vaan puhdistusta ohjataan magneettiventtiilein (kuva 9), joita avataan yksi kerrallaan, jotta saavutetaan haluttu paine. Yhden magneettiventtiilin jälkeen on kytketty kolme tai neljä lamellia. Magneettiventtiileiden ohjaus on toteutettu PLC:llä.



KUVA 10 Lamellien puhdistusjärjestelmä (Valokuva, Iiro Nevalainen)

4.4 Käyttöliittymä

Käyttöliittymänä on Siemensin KTP600-kosketusnäyttöpaneeli (kuva 10), jonka ohjelmointi tapahtuu Siemensin WinCC-ohjelmistolla. Kosketusnäyttöpaneeliin on ohjelmoitu perustoiminnot eli laitteiden on/off-kytkennät sekä magneettiventtiilien pesusyklin käynnistys ja asetusarvojen vaihto. Kosketusnäyttöpaneeli tarjoaa laajat mahdollisuudet prosessin hallintaan ja antaa mahdollisuuden lisätä ominaisuuksia tulevaisuudessa.



KUVA 11 Siemens KTP600-HMI (Valokuva Iiro Nevalainen)

Kosketusnäyttö on liitetty ohjelmoitavaan logiikkaan Profibus-väylän avulla, joka on tarkoitettu laitteiden väliseen kommunikaatioon.

4.5 Virransyöttö

Prosessin virransyöttö tapahtuu kahdella hitsaustasasuuntaajalla (kuva 11). Tasasuuntaajia ohjataan vain on/off-periaatteella. Prosessissa on tarvetta säätää lamelleille syötettävää virtaa, mutta hitsaustasasuuntaajiin on tehtaalla rakennettu vain jännitteensäätömahdollisuus. Jännitteensäätö tasasuuntaajissa on toteutettu tavallisella potentiometrillä, jonka käyttöjännite on 30V AC. Vaihtovirralla operoivan potentiometrin yhdistäminen logiikkaan osoittautui erittäin hankalaksi. Jos jännitteen säätö automatisoidaan tulevaisuudessa, toteutetaan se todennäköisesti ulkoisella moottorilla, joka fyysisesti kääntää potentiometriä haluttuun asentoon.



Jänniteensäätöpottiometri

KUVA 12 Hitsaustasasuuntaaja (Valokuva Iiro Nevalainen)

Hitsaustasasuuntaaja on kykenevä hitsaamaan lamellit yhteen, mutta tämä pyritään välttämään. Hitsaamisen tai valokaaren estämiseksi virtaa tarkkaillaan huolellisesti virta-anturilla. Virta-anturi on kytketty logiikkaan mittausarvon näyttämistä varten ja raja-arvoreleeseen ylivirran katkaisemiseksi. Raja-arvoreleen lisäksi lamellit on suojattu pareittain johdonsuojakatkaisijoilla.

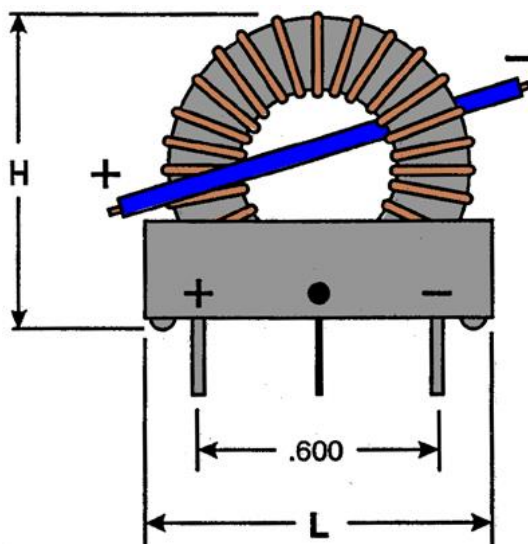
4.6 Virran mittaus

Kuvassa 12 oleva sininen komponentti on mitta-anturi, jonka läpi hitsaustasasuuntaajan syöttöjohto viedään. Mitta-anturi mittaa virtaa 0 – 1000 A ja lähettää siitä mittatiedon 0 – 10 V signaalina eteenpäin. Kuvassa vasemmalla vihreä komponentti on raja-arvorele, joka on asetettu vastaanottamaan 0 – 10 V mittasignaalia. Releeseen asetetaan haluttu raja-arvo 0 – 10 V jolloin sen koskettimet avautuvat ja tällöin tiedetään virran kasvaneen liian suureksi. Jos liian suuri virta havaitaan, releen avautuvat koskettimet katkaisevat tasasuuntaajaa syöttävien kontaktorien pitopiirin ja virransyöttö katkeaa. Tieto avautuvista koskettimista tulee myös logiikalle, joka pysäyttää koko prosessin.



KUVA 13 Virranmittauskeskus (Valokuva Iiro Nevalainen)

Virta-anturiksi valittiin Hall-ilmioon perustuva anturi (kuva 13). Tasavirran mittaamiseen on muitakin vaihtoehtoja, kuten esimerkiksi shunttivastus. Shunttivastus on halvempi vaihtoehto, mutta sen ominaisuudet ovat heikommät. Shunttivastuksessa syntyy ylimääräistä lämpöä ja tehohäviöitä ja sen mittataajuus on rajallisempi kuin Hall-anturin. (Emerald, 1998) Hall-ilmioon perustuvan mittarin toiminta perustuu johdinta ympäröivän magneettikentän keskittämiseen sisällä olevaan rautasydämeen. Rautasydän ei muodosta suljettua ympyrää johtimen ympärille, vaan siinä on ilmapäli. Ilmapäliin sijoitettu Hall-anturi tuottaa jännitelukeman suhteessa ilmapälissä kulkevaan magneettivuohon. (Fluke Inc., 2016)



KUVA 14 Hall IC-virta-anturin kokoonpano. (Emerald, 1998)

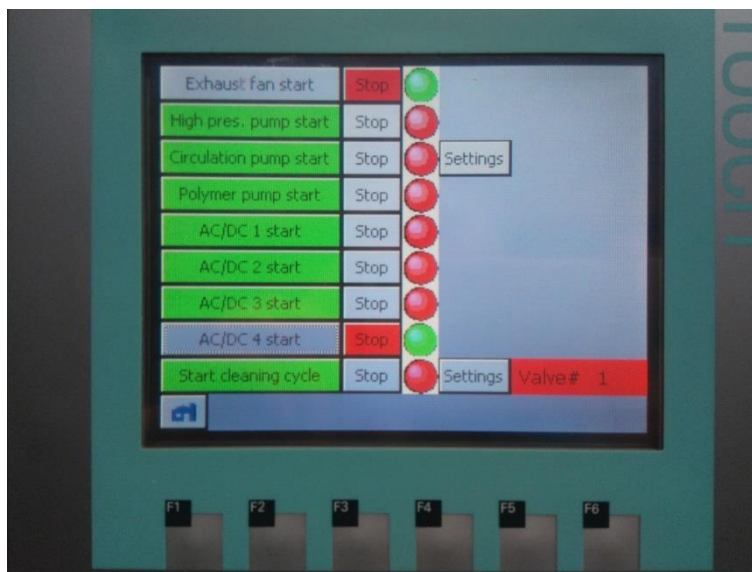
Hall-ilmioon perustuvassa virta-anturissa etuna on sen laaja mitta-alue taajuuden suhteen ja ensiöpiiristä galvaanisesti erottuminen.

4.7 Poistopuhallin

Prosessissa syntyy jonkin verran vetyä, kun sähköä johdetaan veden läpi. Vety on erittäin helposti syttyvä kaasu, ja se täytyy turvallisuussyistä poistaa tilasta mahdollisimman hyvin. Prosessialtaan ylle rakennettiin kaksikanavainen huuva, jonka avulla poistopuhallin imee prosessissa syntyvät kaasut ulos.

Poistopuhaltimen tulee olla käynnissä (kuva 14) aina, kun tasasuuntaajilla syötetään virtaa lamelleihin. Poistopuhaltimen sammumisesta tulee kosketintieto logiikalle, joka sammuttaa prosessin (kuva 15). Tämä tehdään siksi, että prosessissa on kipinöintivaara ja halutaan minimoida vedyn syttymisen mahdollisuus.

Kosketintieto menee myös keskuksen hätä-seis-piirille, joka sammuttaa jännitteensyötön tasasuuntaajille.



KUVA 15 Poistopuhallin ja tasasuuntaaja käynnissä (Valokuva Iiro Nevalainen)



KUVA 16 Poistopuhallin sammutettu, tasasuuntaajassa virhe (Valokuva Iiro Nevalainen)

4.8 Taajuusmuuttajat

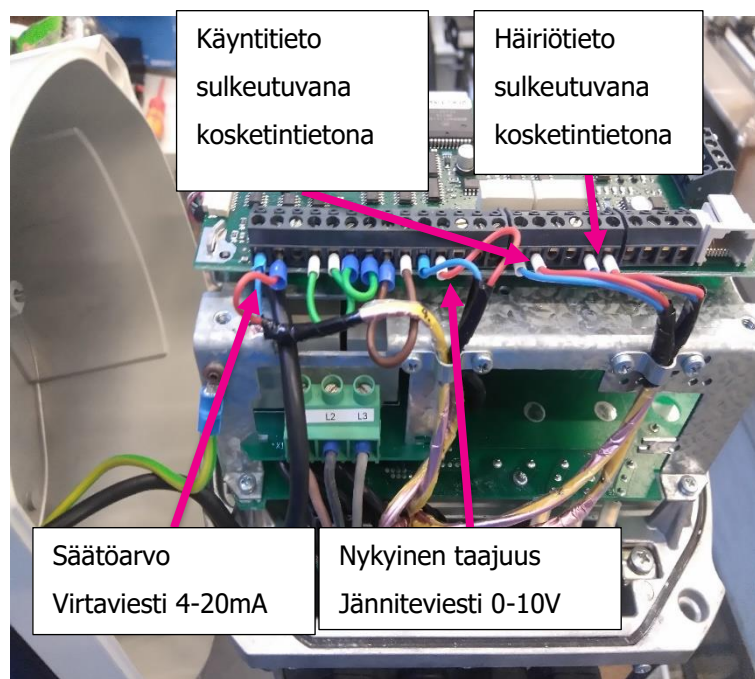
Prosessin hallinta vaatii useita taajuusmuuttajia esimerkiksi prosessikemikaalien syöttöön (kuva 16). Taajuusmuuttaja ohjelmoidaan säätämään moottorin kierrosnopeutta joko manuaalisäätöisesti tai jonkin mittaustuloksen mukaan.

Esimerkiksi vedenkierrätyspumppua säädetään halutun virtausnopeuden saavuttamiseksi. Virtausmittari antaa mittasignaalin PLC:n analogiseen tuloon, jonka avulla PLC laskee halutun taajuuden taajuusmuuttajalle. Säätöarvo viedään taajuusmuuttajalle jännite- tai virtaviestinä (kuva 17).

Teoriassa 0-60 Hz taajuusmuuttaja, jonka tulo on 0 – 10 V, tulkitsee 5 V jänniteviestin siten, että se säätää moottorin pyörimään 30 Hz taajuudella.



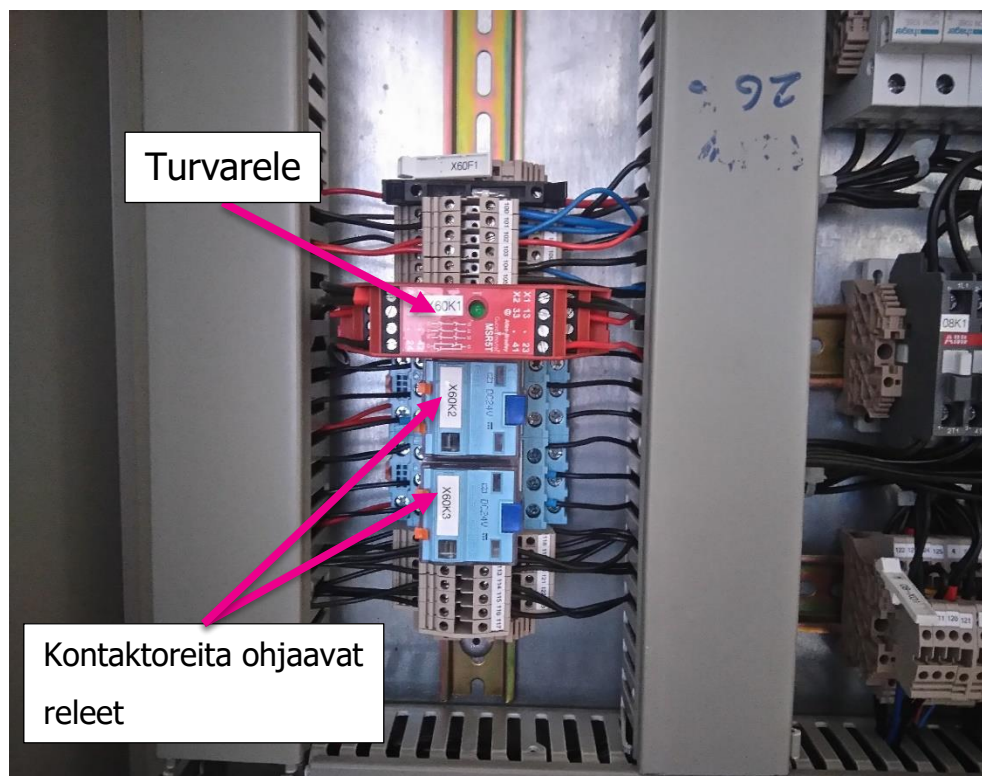
KUVA 17 Kiertopumpun taajuusmuuttaja (Valokuva Iiro Nevalainen)



KUVA 18 Taajuusmuuttajan I/O (Valokuva Iiro Nevalainen)

4.9 Hätä-seis-piiri

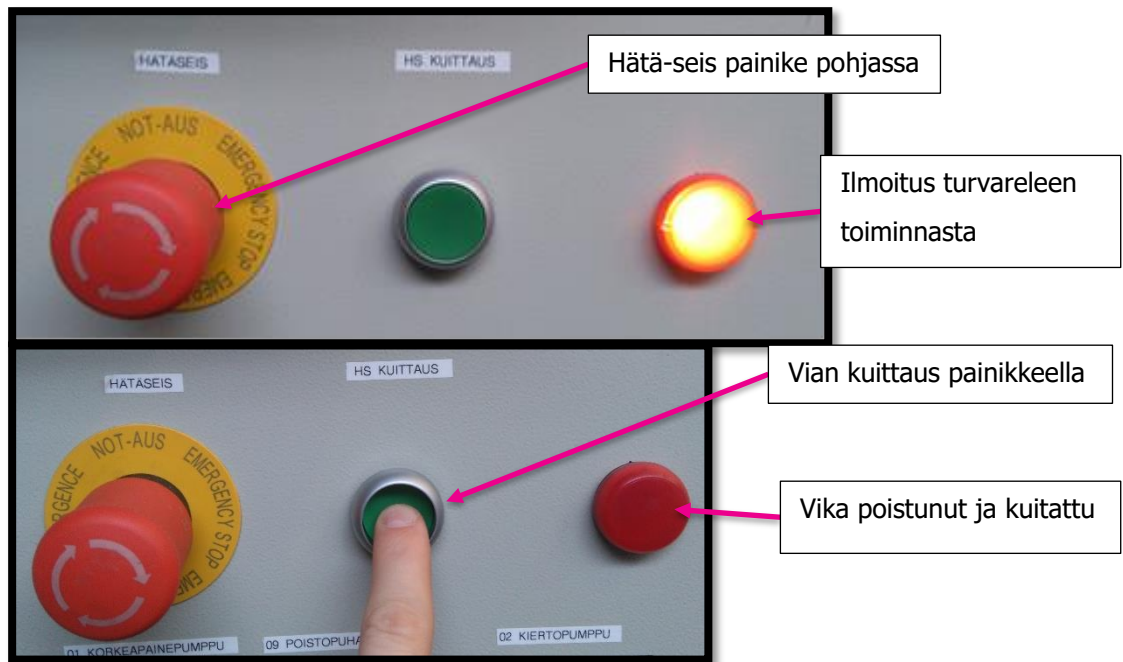
Turvarele on komponentti joka toteuttaa turvatoimenpiteitä virhetilanteessa. Turvarele on yksinkertainen ja tehokas keino täyttää turvastandardit. Turvareleen avulla laitteiston sammutus ja pysäytys voidaan toteuttaa turvallisesti. Turvarele toimii valvomalla laitteiston toimintaa kosketintietojen avulla. (Galco, 2016) Käytetty turvarele valvoo yhden silmukan jatkuvuutta. Silmukan avautuessa prosessi pysähtyy suunnitellusti. Silmukka kulkee hätä-seis painikkeen ja poistopuhallinta ohjaavan kontaktorin lävitse. Jos hätä-seis-painike on pohjassa tai poistopuhallin ei ole käynnissä, prosessi ei käynnisty. Jos prosessi on jo käynnissä ja silmukka avautuu, prosessi pysähtyy. Hätä-seis-piiri on suunniteltu pysäyttämään vain turvallisuuden kannalta tärkeät laitteet. Pysäytettävät laitteet ovat tasasuuntaajat ja kiertopumppu. Tasasuuntaajat pysäytetään, koska niiden käytössä on kipinäinti-vaara ja kiertopumppu pysäytetään, koska prosessi halutaan kokonaisuudessa pysäyttää virransyötön pysähtyessä. Hätä-seis-piiri koostuu turvareleen lisäksi kahdesta muusta releestä (kuva 18), joiden avulla kontaktorien päälle ohjautumista saadaan hallittua. Turvarele ohjaa muut releet vetämään, jos toimintaehdot täyttyvät. Hätä-seis-piiriin sisältyvien laitteiden kontaktoreiden pitopiirit kulkevat näiden releiden läpi. Näin ollen turvareleen reagoidessa vikaan hätä-seis-piiriin hallinnassa olevat laitteet eivät voi käynnistyä.



KUVA 19 Hätä-seis-piiri (Valokuva Iiro Nevalainen)

Hätä-seis-piiriin tärkeä ominaisuus on käyttäjän tekemä vian kuittaus (kuva 19). Jos vika havaitaan tai turvareleen ulkoinen virransyöttö katkeaa, rele reagoi vikaan. Kun vika poistetaan, esimerkiksi vapauttamalla hätä-seis-painike, prosessi ei saa automaattisesti jatkaa. Vika tulee aina kuitata erillisellä painikkeella. Näin prosessi ei käynnisty yllättäen, kun vika poistetaan. Vian kuittaaminenkaan ei

saa käynnistää konetta tai prosessia, vaan sen tarkoitus on mahdollistaa uudelleenkäynnistäminen.
(EUR-Lex, 2006)



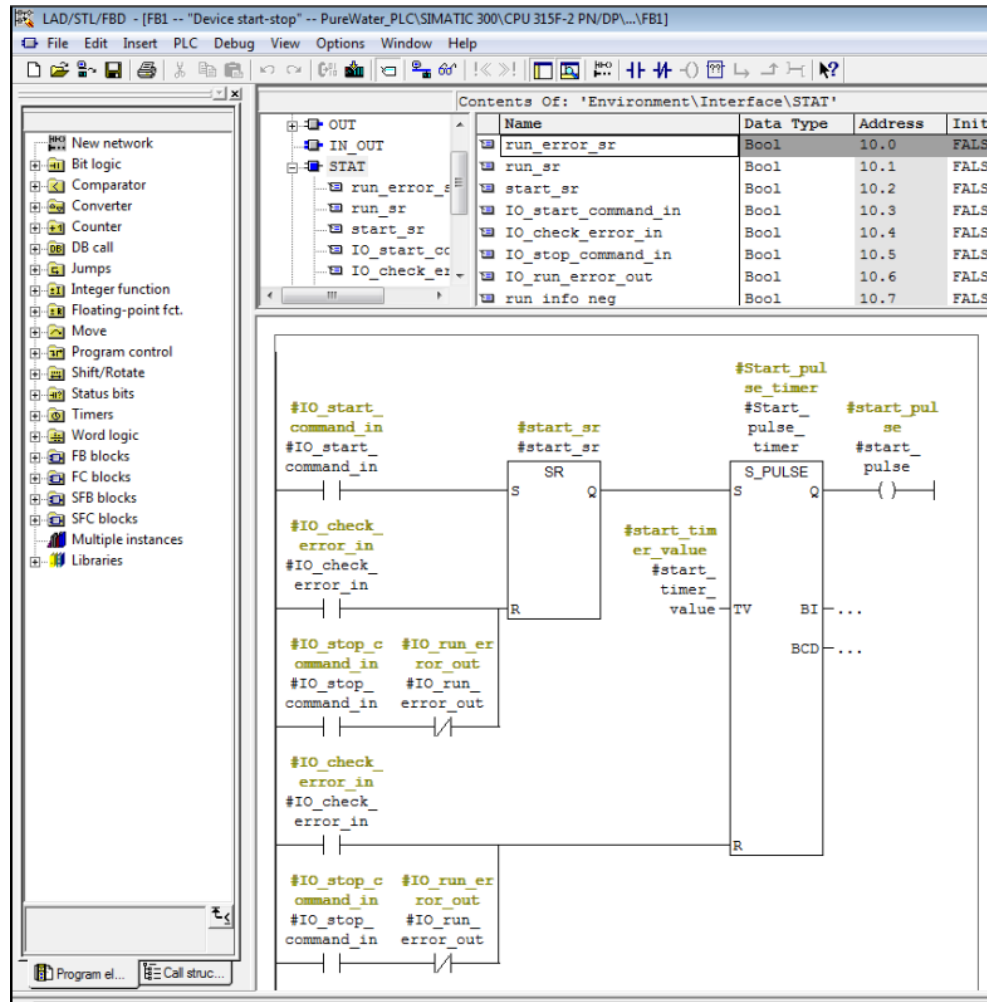
KUVA 20 Hätä-seis-piirin vian kuittaus (Valokuva Iiro Nevalainen)

5 AUTOMAATION OHJELMOINTI

Logiikan ohjelmointi toteutettiin Siemens Step7 ohjelman avulla.

Suurin osa ohjelmoinnista tehtiin ladder-kielessä, joka on visuaalinen, porrasmainen ohjelmointikieli.

Ladder-ohjelmoinnissa luodaan verkko, jonka sisään asetetaan haluttuja logisia portteja, kiikkuja ja ajastimia. Verkossa virtaa kuvitteellinen jännite vasemmalta oikealle.



KUVA 21 Step7 ladder-ohjelmointinäköymä (Valokuva Iiro Nevalainen)

Esimerkki ladder-ohjelmoinnista:

Esimerkkikuvassa (kuva 20) on vasemmalla tuloja, kuten #IO_start_command_in. Tämä on sisäinen muistibitti, jota ohjataan kosketusnäyttöpaneelistä. Kun bitti asetetaan asentoon 1, eli se muuttuu statuksesta 0 statukseen 1, asettuu SR-kiikku #start_sr statukseen 1 ja pysyy statuksessa 1, kunnes jokin tulo kiikun R-porttiin muuttuu statukseen 1.

Osa ohjelmoinnista oli kuitenkin helpompi toteuttaa STL-kielessä, joka on täysin tekstipohjainen ohjelmointikieli Step7:ssä. STL-kielessä ohjelmoitiin analogisten tulojen luku ja kirjoitus muistiin sekä analogisten lähtöjen luku muistista ja kirjoitus lähtöosoitteeseen (kuva 21).

Block: FC4

Network: 1

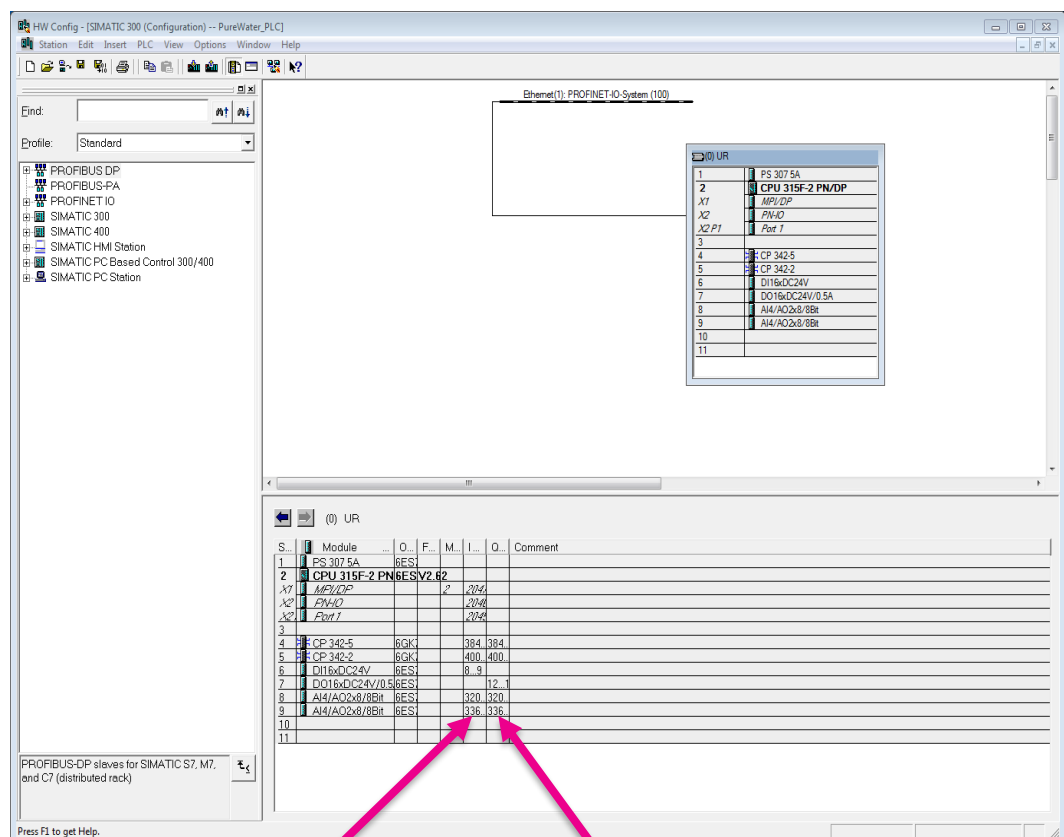
L	←	MB	100
T	←	PQB	400
L		MB	101
T		PQB	401
L		MB	102
T		PQB	402
L		MB	103
T		PQB	403

Luku muistitavusta 100

Kirjoitus lähtötavuun 400

KUVA 22 Esimerkki STL ohjelmasta (Valokuva Iiro Nevalainen)

Jokaiselle logiikan I/O-moduulille määräytyy automaattisesti omat osoitteet moduulin lähtöjä ja tuloja varten. Digitaaliset osoitteet ovat esim. I2.5 (input tavussa 2, bitissä 5). Analogiset osoitteet vievät koko tavun ja ovat tyyliltään esim. I100. Osoitteet voi tarkistaa HW Config alaohjelmasta step7:ssä (kuva 22).



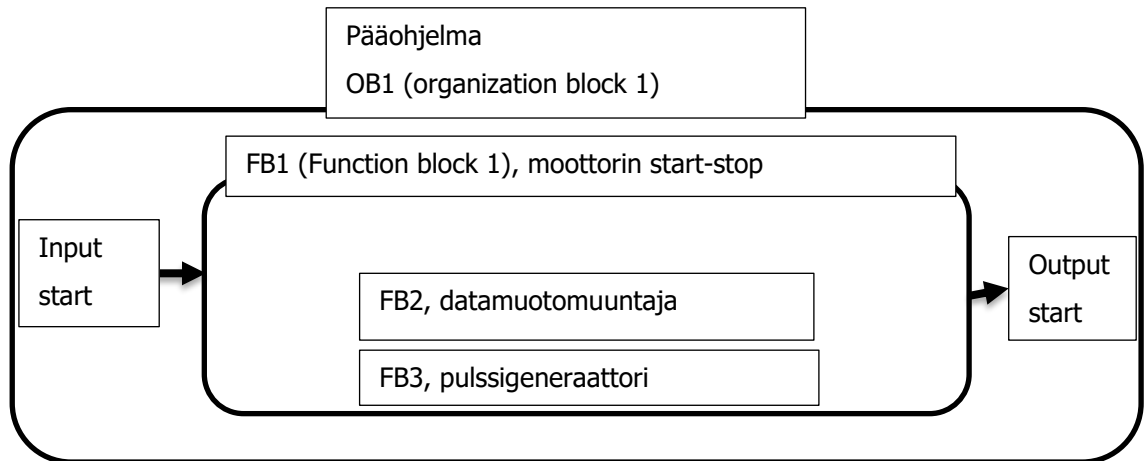
Tulo-osoitteet

Lähtöosoitteet

KUVA 23 Step7 HW Config (Valokuva Iiro Nevalainen)

5.1 Ohjelmat

Step7 ohjelmoinnissa voidaan ohjelmoida pienempiä kokonaisuuksia kerrallaan ja käyttää niitä muissa ohjelmissa myöhemmin (kuva 23). Tämä on hyvä siksi, että esimerkiksi tässä projektissa jokaisella koneella on sama start-stop-piiri, joka piti ohjelmoida vain kerran. Jos osakokonaisuuteen, eli blockkiin, tulee myöhemmin muutoksia, muutokset tulevat kerralla kaikkialle jossa osakokonaisuutta käytetään.

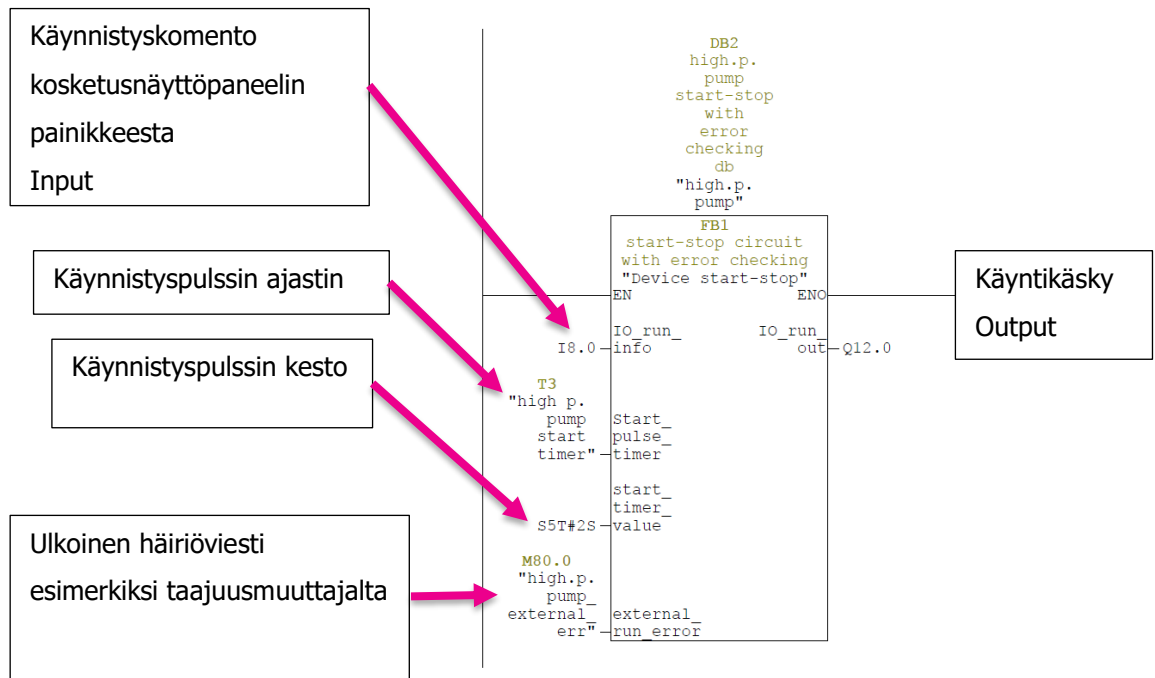


KUVA 24 Ohjelman rakenteen havainnollistaminen (Kuvio, Iiro Nevalainen)

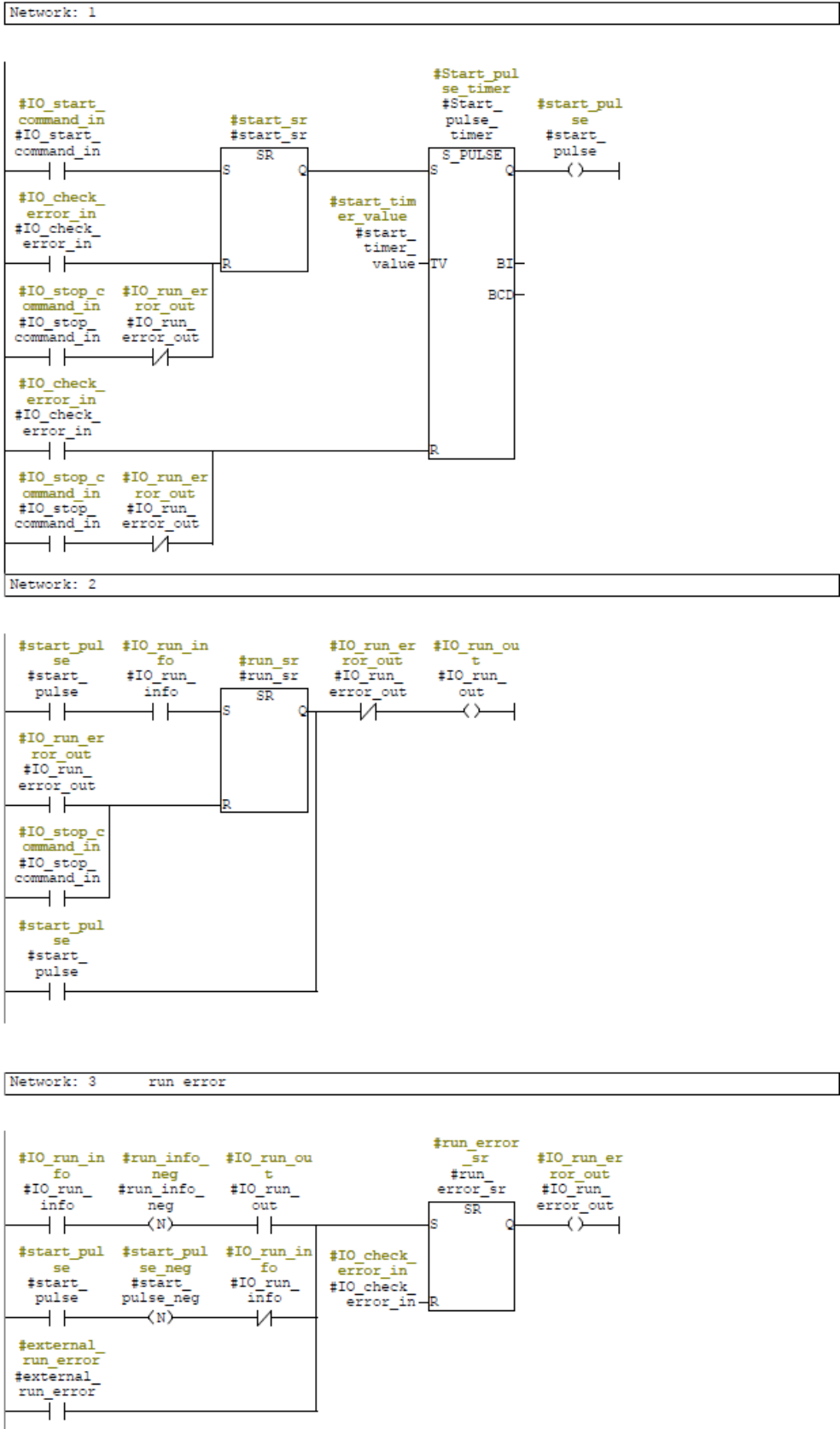
Ohjelmoinnissa tuli vastaan useita tilanteita, joissa valmiiksi ohjelmoitua blockkia voitiin käyttää kerta toisensa jälkeen. Tällainen ohjelmistorakenne säästää aikaa ja helpottaa muutosten tekoa.

5.1.1 Moottorin start-stop-piiri

Moottorin start-stop-piiriksi nimetty blockki (kuva 24 ja 25) ohjaa tässä prosessissa jokaista kontaktorilähtöistä laitetta. Piirin tavoitteena on ohjata laitteet käyntiin ja kiinni. Jos vika havaitaan, ilmoittaa ohjelma siitä error-viestillä ja sammuttaa laitteen eikä salli uudelleenkäynnistystä ennen kuin käyttäjä on kuitannut vian. Lisäksi piiri havaitsee, jos laite ei jostain syystä käynnisty komennosta huolimatta.



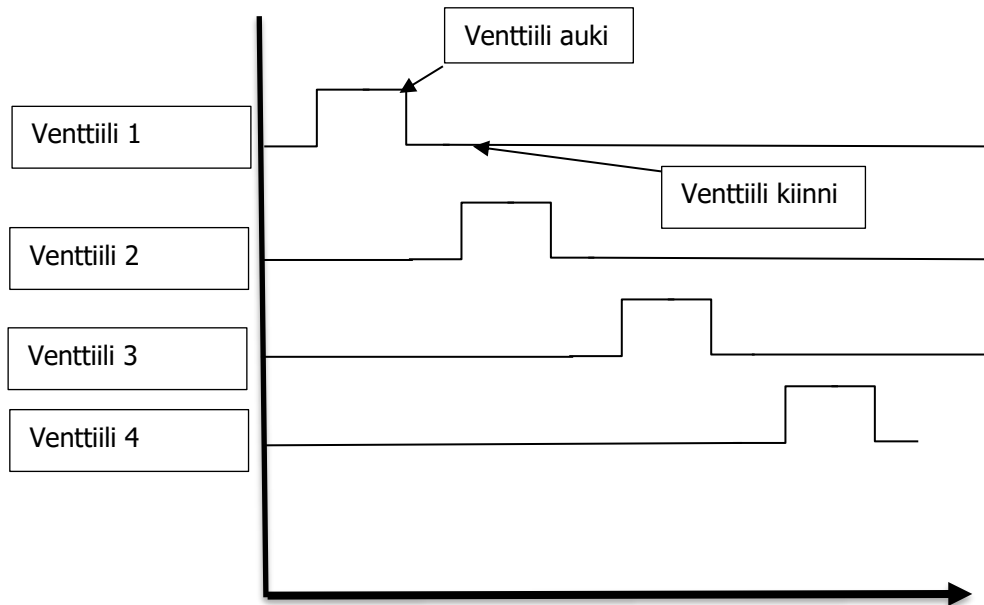
KUVA 25 Start-stop-piiri blockkina (Valokuva Iiro Nevalainen)



KUVA 26 Start stop piirin sisältö (Valokuva Iiro Nevalainen)

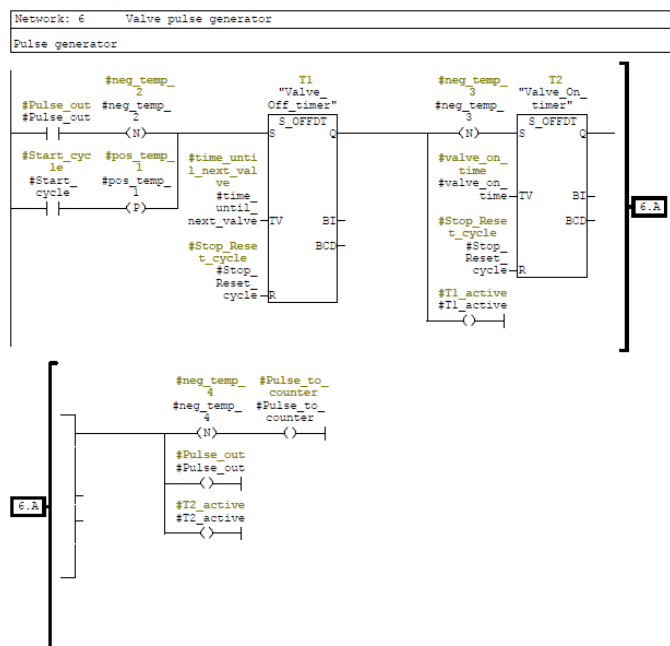
5.1.2 Lamellien puhdistusohjelma

Lamellien puhdistamista varten täytyi kirjoittaa ohjelma, joka avaa magneettiventtiileitä tietyin väliajoin. Ohjelmaa täytyi pystyä säätämään kesken puhdistusohjelman käyttäen kosketusnäyttöä. Ohjelmalla säädetään venttiilin aukioloaika ja se, kuinka pitkään venttiilit ovat kiinni ennen kuin seuraava aukeaa ja mitkä venttiilit ovat mukana syklissä (kuva 26).



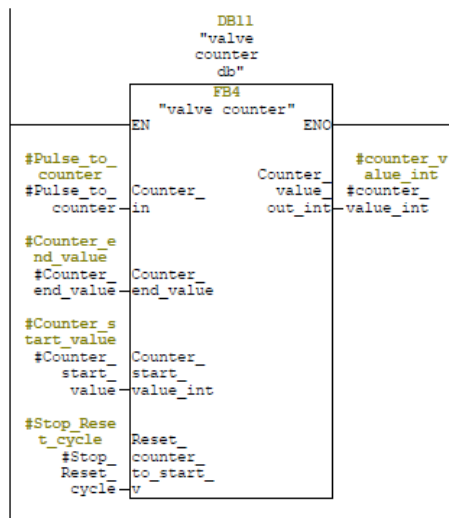
KUVA 27 Venttiilien avausyklin havainnollistaminen (Valokuva Iiro Nevalainen)

Venttiileiden avaus ja sulkua toteutettiin kahdella ajastimella (kuva 27). Ajastin ohjaa yhtä venttiiliä kerrallaan. Kun kummatkin ajastimet ovat laskeneet asetetusta arvosta nolleen, nousee laskurin arvo yhdellä ja siirtyy seuraavaan venttiiliin.

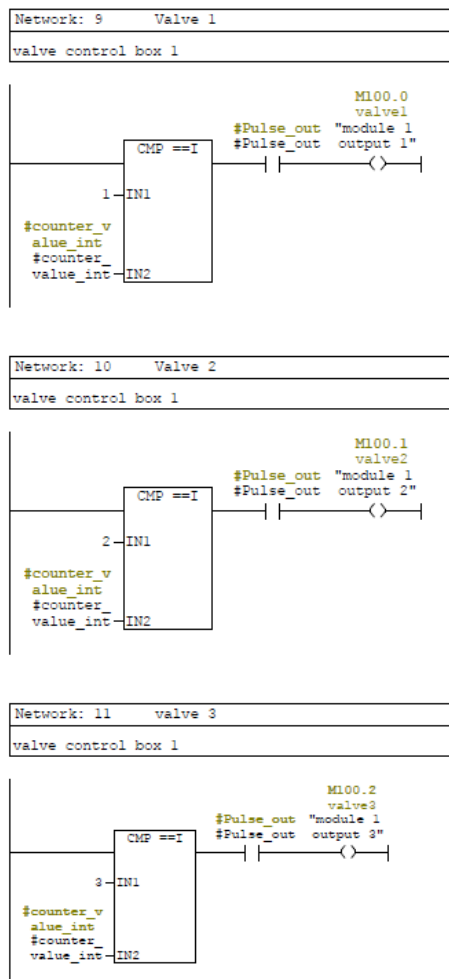


KUVA 28 Pulssigeneraattori (Valokuva Iiro Nevalainen)

Venttiili aukeaa kun "valve_on_timer" on aktiivinen ja laskurin arvo (kuva 28) vastaa venttiilille annettua numeroa (kuva 29).



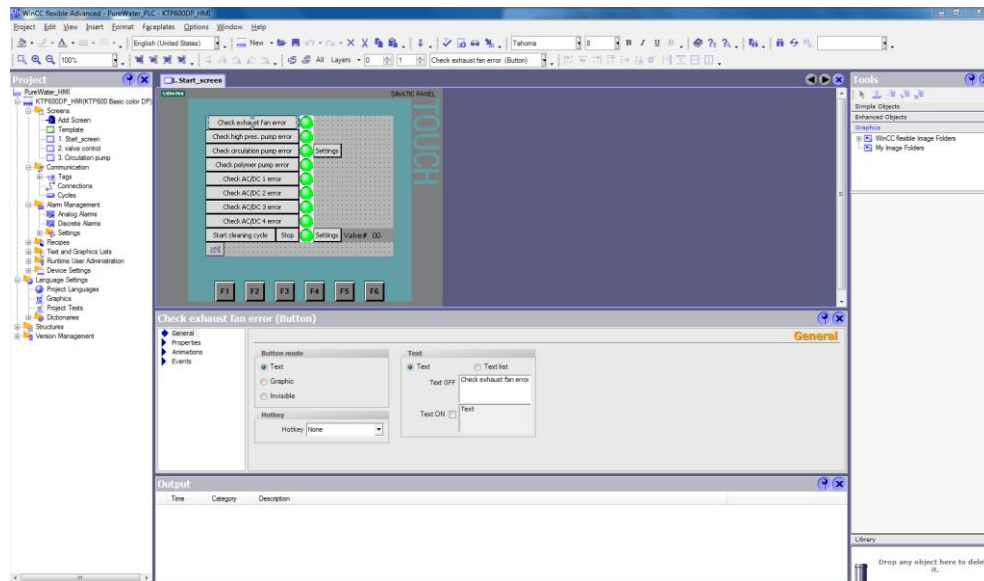
KUVA 29 Pulssilaskuri (Valokuva Iiro Nevalainen)



KUVA 30 Venttiilien aktivointipiirit (Valokuva Iiro Nevalainen)

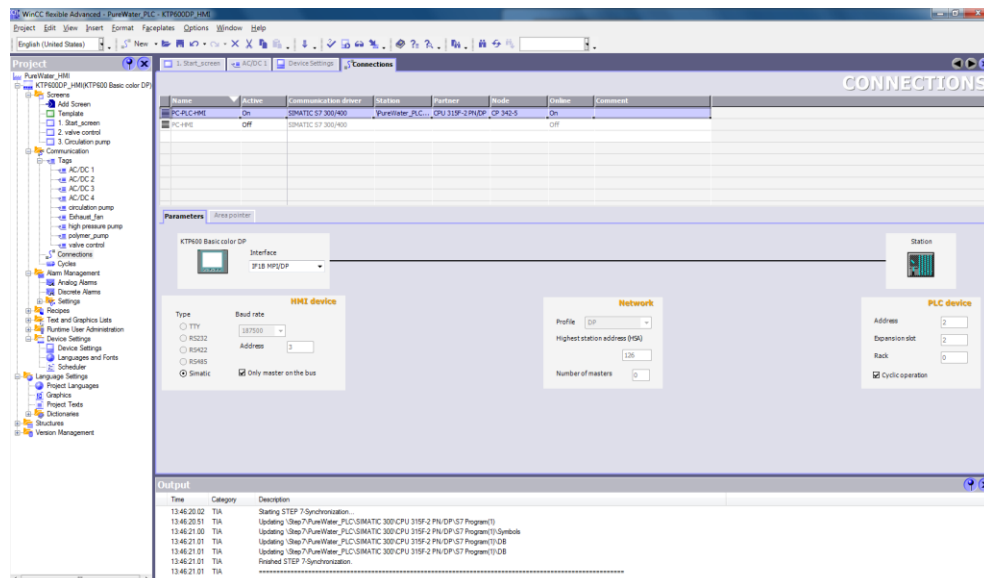
5.2 Käyttöliittymän ohjelmointi

Käyttöliittymän ohjelmointi kosketusnäyttöpaneelille tehtiin Siemens WinCC-ohjelmalla (kuva 30). Käyttöliittymän piti olla helppokäyttöinen, jotta käytön opetteluun ei kuluisi turhaa aikaa. Käyttöliittymään tulee tulevaisuudessa muutoksia, kun säädettäviä laitteita ja antureita tulee lisää.



KUVA 31 WinCC-ohjelmointiympäristö (Valokuva Iiro Nevalainen)

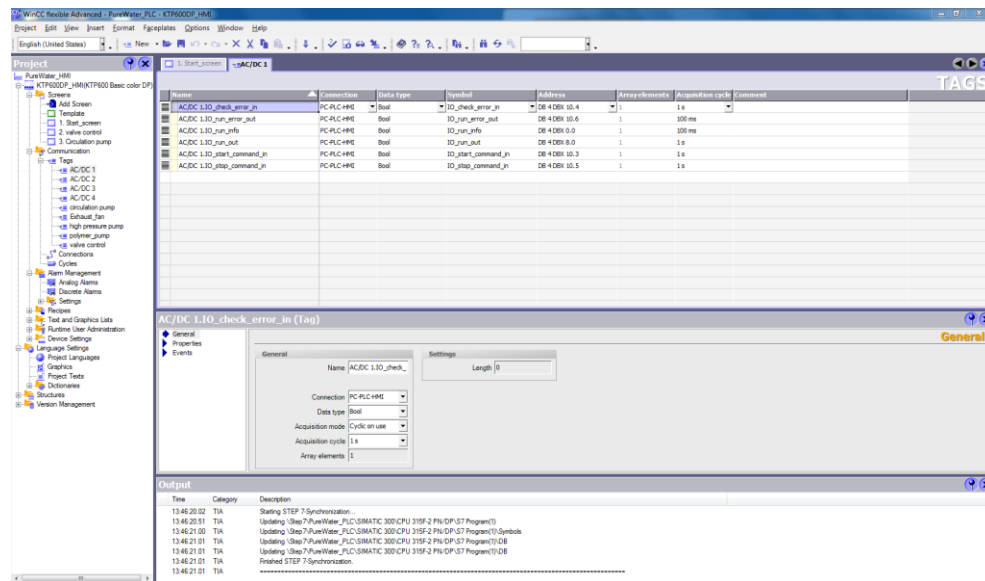
Paneelin liitetään logiikkaan Profibus-väylällä ja se konfiguroidaan ensin Step7 ohjelmassa HW Configuration alaohjelmalla. Paneeli pitää konfiguroida WinCC-ohjelmalla (kuva 31), jotta yhteys tietokoneelta toimii PLC:n kautta paneelille.



KUVA 32 WinCC yhteyden konfigurointi (Valokuva Iiro Nevalainen)

Yhteyden määrittämisessä asetetaan ensin väliasema eli s7-300 logiikkaohjain. Seuraavaksi määritetään käytettävä protokolla ja sen nopeus eli Profibus DP@187kpbs.

Ohjelmoitavat lähdöt ja tulot, kuten käynnistyspulsstin aktivointi, lisätään WinCC:ssä niin kutsuttuihin tageihin, jolloin ne ovat yhteydessä Step7:n kanssa.

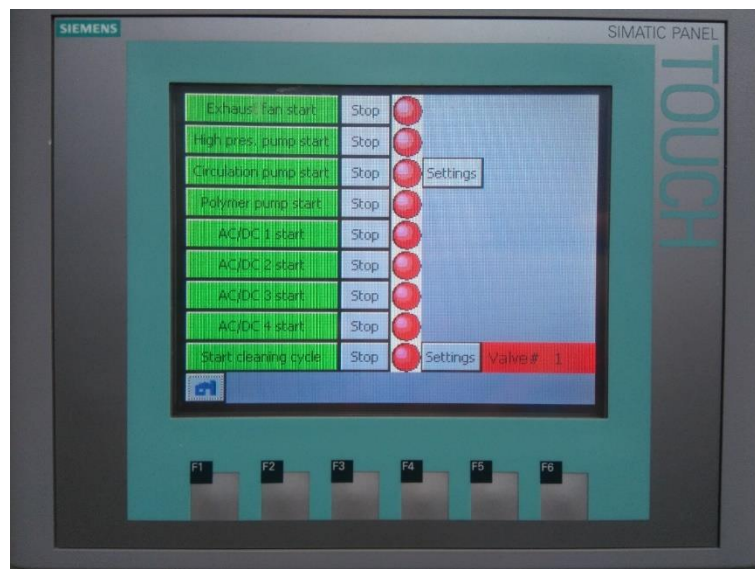


KUVA 33 WinCC tagien määrittäminen (Valokuva Iiro Nevalainen)

Tagin voi määrittää manuaalisesti tai pudotusvalikosta, jolloin valitaan mistä tietokannasta tagi haetaan (kuva 32). Pudotusvalikon kautta määritetyille tageille määräytyy automaattisesti nimi, käytettävä yhteys, tagin tyyppi ja osoite. Ohjelma päivittää tagien tilan PLC:n ja paneelin välillä vakioaika-tyksellä sekunnin välein, mutta sen voi muuttaa manuaalisesti esimerkiksi 100ms:ksi, jos on tarvetta esimerkiksi lukea ajastimen arvoja nopealla päivitysnopeudella. Esimerkkikuvassa (KUVA 33 WinCC tagi) luetaan kontaktoriohjauksessa tapahtuva error-viesti 100 ms välein, jotta viesti välittyy laitteen käyttäjälle viiveettä.

5.2.1 Kontaktorilähtöjen ohjaus

Kontaktorilähdöillä on käyttöliittymässä vain on-off-ohjaus sekä virheilmoituksen luku ja kuitaus. Tarvittaessa laitteelle ohjelmoidaan erikseen säätöruutu, kuten kiertopumpulle (circulation pump) on ohjelmoitu.



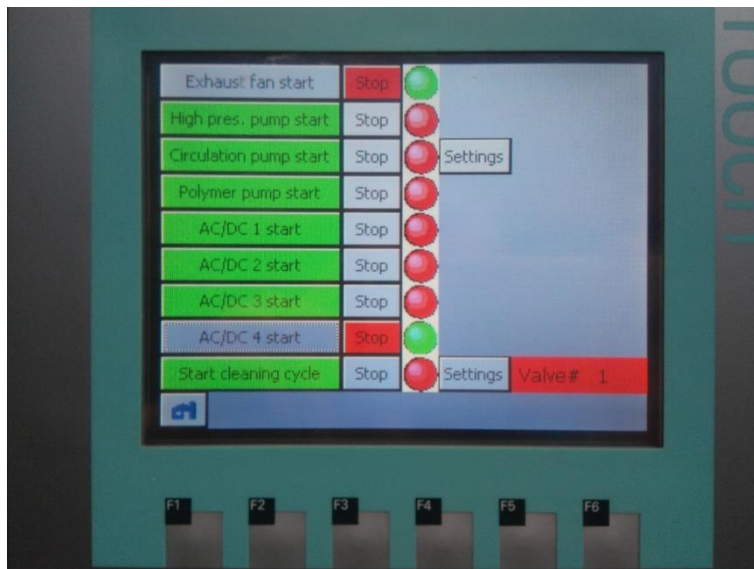
KUVA 34 Käyttöliittymän alituskuvaa (Valokuva Iiro Nevalainen)

Vihreän start-painikkeen painaminen paneelista käynnistää laitteen käynnistyspulssein (kuva 33). Käynnistyspulssi koettaa käynnistää laitetta esim. 2 s ajan. Jos laite käynnistyy pulssin aikana, jää laite käyntiin ilman virhettä. Jos laitteen käynnistys epäonnistuu, eli laitteelta ei vastaanoteta käyntitietoa tai laitteelta vastaanotetaan virheilmoitus, laite sammuu ja paneeli ilmoittaa virheestä.



KUVA 35 Virheilmoitus näytöllä (Valokuva Iiro Nevalainen)

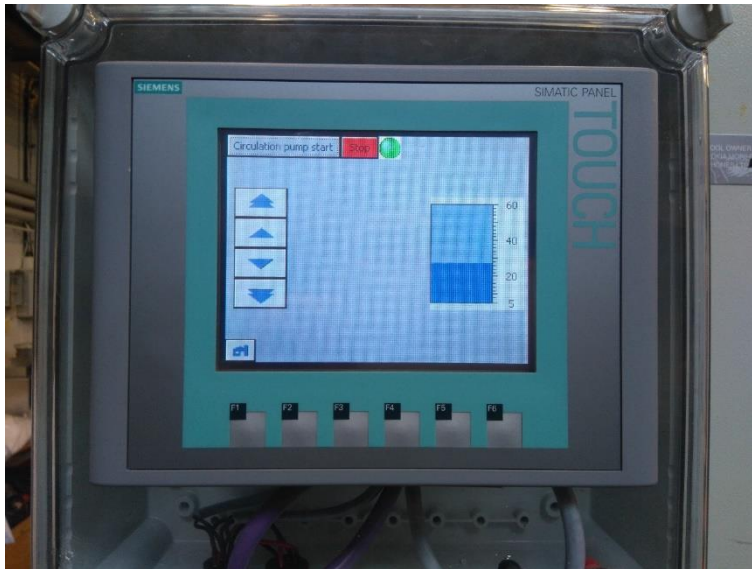
Jos laitteen käynnistystä halutaan yrittää uudelleen, tulee virheilmoitus kuitata painamalla punaista check error-painiketta (kuva 34).



KUVA 36 Laitteita käynnissä (Valokuva Iiro Nevalainen)

Onnistuneen käynnistykseen merkiksi punainen pilottivalo vaihtuu vihreäksi (kuva 35).

Tätä työtä varten ohjelmoitin malliksi kiertopumpun nopeudensäätöruutu, josta voi lukea kiertopumpun nopeutta ja muuttaa sitä.

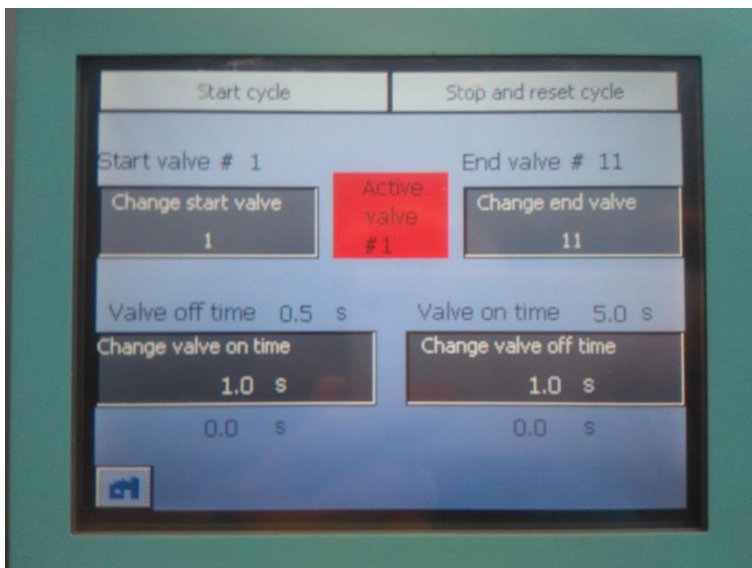


KUVA 37 Kiertopumpun säätöruutu (Valokuva Iiro Nevalainen)

Kuvassa (kuva 36) ylhäällä näkyvät käynnistä- ja sammuta-painikkeet ovat samat kuin aloitusruudulla laitteen nopeaa hallintaa varten. Kuvassa vasemmalla näkyvät painikkeet nostavat tai laskevat pumpun kierrosnopeutta. Kuvassa oikealla näkyvä palkki kuvaa pumpun nopeutta taajuutena 5 – 60 Hz.

5.2.2 Venttiiliohjaus

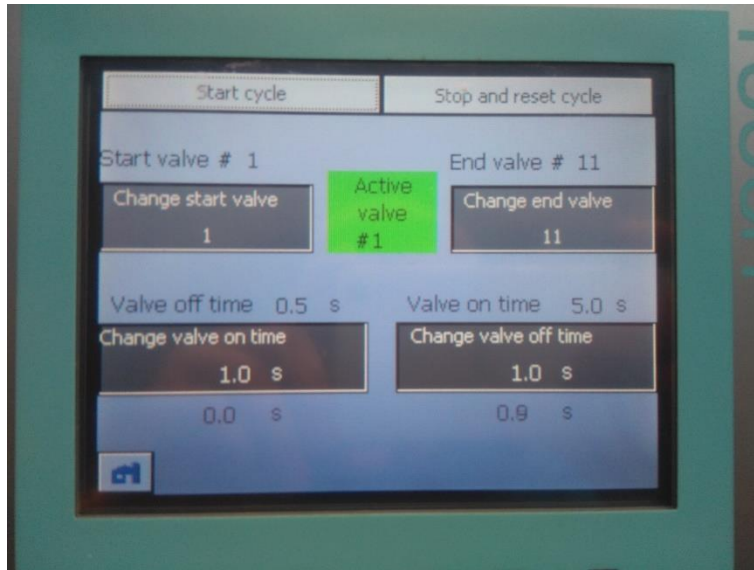
Lamelleiden pesuohjelman asetuksia täytyy pystyä säätämään kesken ohjelman, joten säätöä varten paneeliin ohjelmoitiin reaaliaikainen käyttöliittymä. Ohjelmalla säädetään venttiilien auki- ja kiinnioloaikoja sekä venttiilit, jotka halutaan sisällyttää ohjelmaan.



KUVA 38 Pesuohjelman säätöruutu (Valokuva Iiro Nevalainen)

Pesuohjelman säätöruudussa voidaan säätää, mikä venttiili aukeaa ensimmäisenä (Start valve #) ja minkä venttiilin jälkeen (End valve #) palataan ensimmäiseen. Venttiilin säätö tapahtuu "Change start valve" ja "Change end valve" painikkeilla (kuva 37).

Edellisen lisäksi on säädettävä kuinka pitkään venttiilit ovat suljettuina ja kuinka pitkään ohjattava venttiili on auki. Ruudussa näkyy tämänhetkinen asetusarvo "Valve off time" ja "Valve on time" tekstin jälkeen. Asetusta voidaan muuttaa painikkeilla "Change valve on time" ja "Change valve off time" painikkeilla. Edellä mainittujen painikkeiden alla näkyy ajastimissa jäljellä oleva aika (kuva 38).



KUVA 39 Pesuohjelma käynnissä (Valokuva Iiro Nevalainen)

Vihreällä pohjalla oleva "Active valve"-teksti kertoo ohjelman olevan käynnissä.

Kuvasta 38 voidaan tulkita, että venttiili 1 on auki vielä 0.9 sekunnin ajan.

6 YHTEENVETO

Opinnäytetyön tavoitteena oli rakentaa jäteveden lamellipuhdistimen prosessinohjaukseen toimiva automaatiojärjestelmä. Prosessin automaatio saatiin suunniteltua hyvin ottaen huomioon ajan ja laitteistojen, tai niiden puuttumisen, asettamat rajoitteet. Projektiin oli suunniteltu laitteita, joita ei toimitettu asennettavaksi opinnäytetyön aikana.

Kontaktorilähtöjen ohjauksessa oli huomioitava laitteiden standardin mukainen toiminta, jotta laitteiston käyttö on turvallista. Turvallisuuden kannalta tärkein komponentti on hätä-seis-piiri, joka pysäyttää prosessin vikatilanteessa eikä salli laitteiden yllättävää uudelleenkäynnistymistä. Myös automaatiohjauksessa turvallisuus oli otettava huomioon ohjelmoimalla logiikka valvomaan lähtöjen toimintaa. Logiikka ohjelmoitiin ilmoittamaan toiminnan virheestä ja pysäyttämään toiminta aivan kuten hätä-seis-piiri ja toiminnan jatkamiseksi vaaditaan käyttäjän kuittaus ennen uudelleenkäynnistystä. Laitteiston ja ohjelmiston toiminta tuli suunnitella mahdollisimman turvallisesti minimoimalla virheellisen ja yllättävän toiminnan mahdollisuus.

PLC:n ja HMI:n ohjelmointi vaati aluksi paljon ohjelmointityökalujen ja laitteiden toiminnan opiskelua. Ohjelmointi vaati paljon virheiden kautta oppimista, sillä aivan kaikkeen ei löytynyt sopivia op-paita. Ohjelmiston suunnittelussa tuli huomioida erityisesti se, että ohjelmisto toimii aina odotetulla tavalla. Jotta ohjelmisto toimii odotetusti, tulee ottaa huomioon kaikki mahdolliset variaatiot, joilla laitteiston toiminta ja käyttäjän syötteet vaikuttavat ohjelmiston toimintaan. Näin tuloksena on varmatoiminen ohjelmisto, jota on turvallista käyttää. Ohjelmistoa testattiin kaikilla mahdollisilla komentoyhdistelmillä niin kauan, että kaikki esille tulleet ohjelmistovirheet oli korjattu. Automaatiolaitteistossa tapahtuva laitteistovika on aina mahdollinen, ja siksi keskukseen rakennettiin laitteiden manuaalikäyttömahdollisuus sekä hätä-seis-piiri lisäämään turvallisuutta. Suurin ongelma ohjelmoinnissa oli saada HMI keskustelemaan logiikan kanssa. Syyksi paljastui HMI:n käyttöjärjestelmäversion ja ohjelmointityökalun välinen yhteensopimattomuus.

Laitteiden ohjaus saatiin kuitenkin toteutettua suurimmilta osin halutulla tavalla. Automaation toteutuksessa suurimmaksi puutteeksi jäi tasasuuntaajien virran säädön automatisoiminen, joka jäi toteuttamatta lähinnä tasasuuntaajien vanhuuden vuoksi.

Kenttäväylien käyttö projektissa oli välttämätöntä, koska käyttöliittymän ja logiikan väliseen tiedonsiirtoon ei ole muita vaihtoehtoja. AS-i-väylän käyttö ei ollut pakollista, mutta sen käyttö yksinkertaisti kaapelointia ja magneettiventtiilien ohjausta todella paljon. Kenttäväylien käyttö projektissa vaati kenttäväylien toimintaperiaatteiden ja teknisen toteutuksen opettelemista. Kenttäväylien käytöstä selvittiin ilman erityisiä ongelmia.

LÄHTEET JA TUOTETUT AINEISTOT

- Emerald, P. (6. 5. 1998). *Non-Intrusive Hall-Effect Current-Sensing Techniques Provide Safe, Reliable Detection and Protection for Power Electronics*. (Allegro MicroSystems, LLC) Haettu 19. 11. 2016 osoitteesta <http://www.allegromicro.com/en/Design-Center/Technical-Documents/Hall-Effect-Sensor-IC-Publications/Non-Intrusive-Hall-Effect-Current-Sensing-Techniques-for-Power-Electronics.aspx>
- EUR-Lex. (9. 6. 2006). *EUROOPAN PARLAMENTIN JA NEUVOSTON DIREKTIIVI 2006/42/EY*. Haettu 20. 11. 2016 osoitteesta koneista ja direktiivin 95/16/EY muuttamisesta (uudelleenlaadittu): <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32006L0042:FI:PDF>
- Fluke Inc. (2016). *Inside Hall Effect (ac, dc) clamp meters*. Haettu 19. 11. 2016 osoitteesta Fluke: <http://en-us.fluke.com/training/training-library/test-tools/clamp-meters/inside-hall-effect-clamp-meters.html>
- Galco. (2016). *Safety Relays - Where and How They Work*. (Galco Industrial Electronics) Haettu 20. 11. 2016 osoitteesta Galco Industrial Electronics: <https://www.galco.com/comp/prod/saftrela.htm>
- National Instruments. (22. 6. 2010). *PROFIBUS Overview*. Haettu 25. 10. 2016 osoitteesta National Instruments: <http://www.ni.com/white-paper/6958/en/>
- PAControl.com. (10. 2002). *PROFIBUS Technology and Application*. Haettu 25. 10. 2016 osoitteesta Process Automation Control: <http://www.pacontrol.com/download/profibus-overview.pdf>
- Siemens. (2016). *Siemens - AS-i liitännäiset kenttäkomponentit*. Haettu 16. 11. 2016 osoitteesta Siemens AG:n sivusto: http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/kone_ja_prosessiturvallisuus_seka_atex/koneturvallisuus/as_i_safety/as_i_liitannaiset_kenttakomponenti.htm
- SMAR. (2016). *SMAR - Tutorial on AS-i technology*. (SMAR) Haettu 19. 10. 2016 osoitteesta SMAR Industrial Automation: <http://www.smar.com/en/asi>

LIITTE 1: PLC-OHJELMISTO

SIMATIC

PureWater_PLC\

09/16/2016 09:10:22 AM

SIMATIC 300\CPU 315F-2 PN/DP\...\OB1 - <offline>

OB1 - <offline>

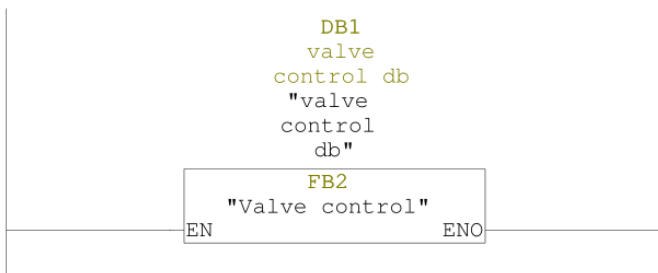
""

Name: **Family:**
Author: **Version:** 0.1
 Block version: 2
Time stamp Code: 09/06/2016 06:05:31 PM
 Interface: 07/01/2016 10:16:52 AM
Lengths (block/logic/data): 00908 00776 00036

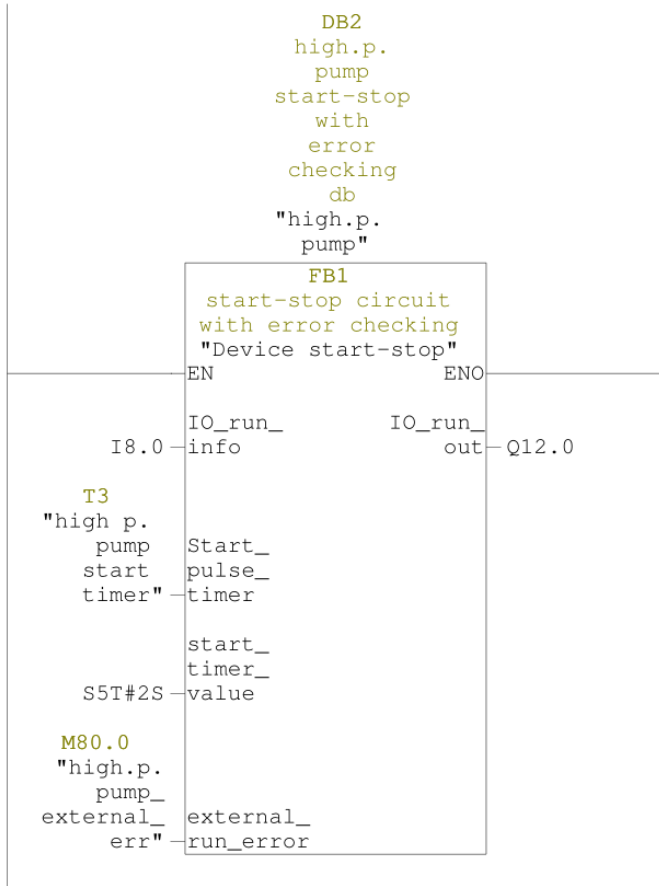
Name	Data Type	Address	Comment
TEMP		0.0	
OB1_EV_CLASS	Byte	0.0	Bits 0-3 = 1 (Coming event), Bits 4-7 = 1 (Event class 1)
OB1_SCAN_1	Byte	1.0	1 (Cold restart scan 1 of OB 1), 3 (Scan 2-n of OB 1)
OB1_PRIORITY	Byte	2.0	Priority of OB Execution
OB1_OB_NUMBR	Byte	3.0	1 (Organization block 1, OB1)
OB1_RESERVED_1	Byte	4.0	Reserved for system
OB1_RESERVED_2	Byte	5.0	Reserved for system
OB1_PREV_CYCLE	Int	6.0	Cycle time of previous OB1 scan (milliseconds)
OB1_MIN_CYCLE	Int	8.0	Minimum cycle time of OB1 (milliseconds)
OB1_MAX_CYCLE	Int	10.0	Maximum cycle time of OB1 (milliseconds)
OB1_DATE_TIME	Date_And_Time	12.0	Date and time OB1 started
Valvecontrol	Any	20.0	

Block: OB1 "Main Program Sweep (Cycle)"

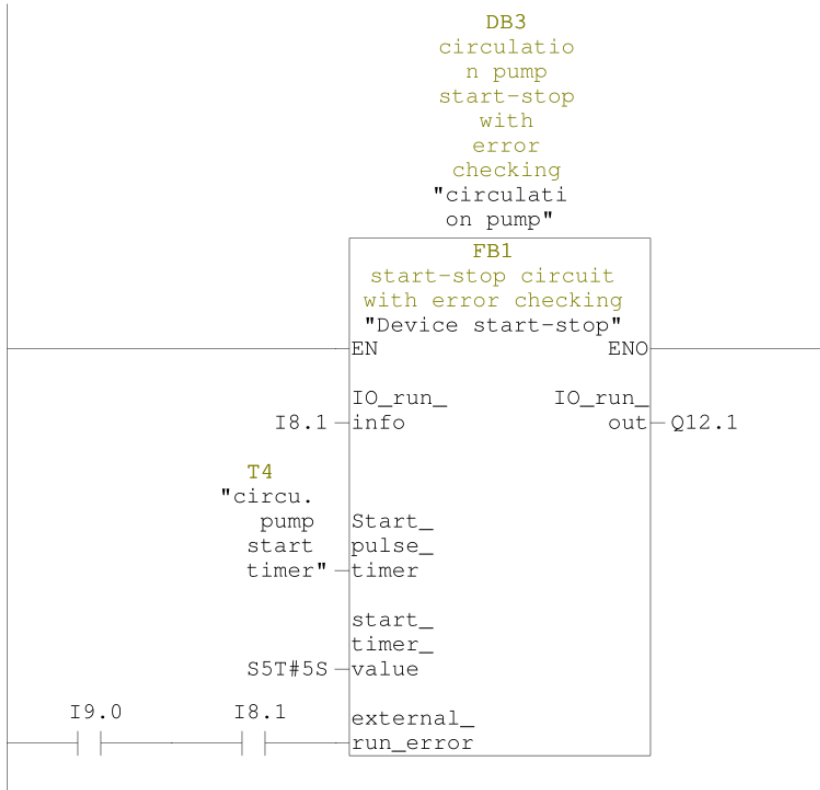
Network: 1



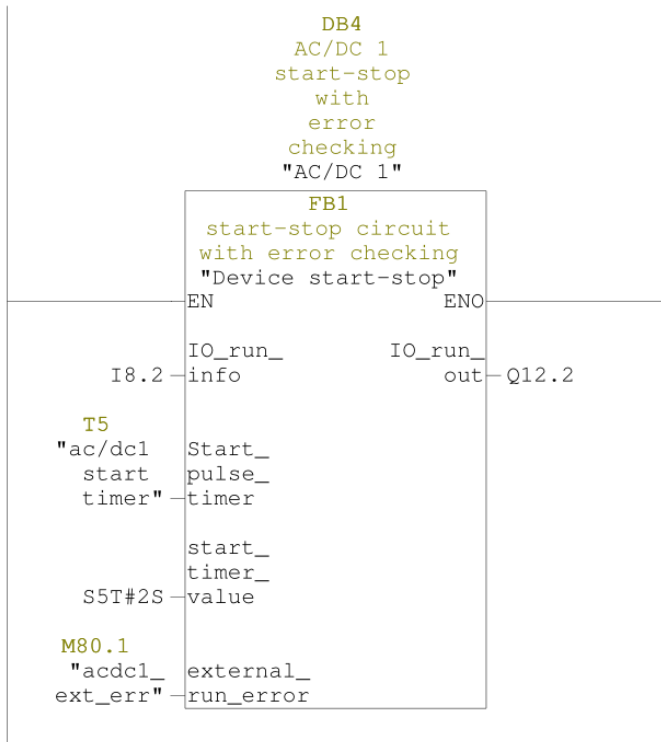
Network: 2



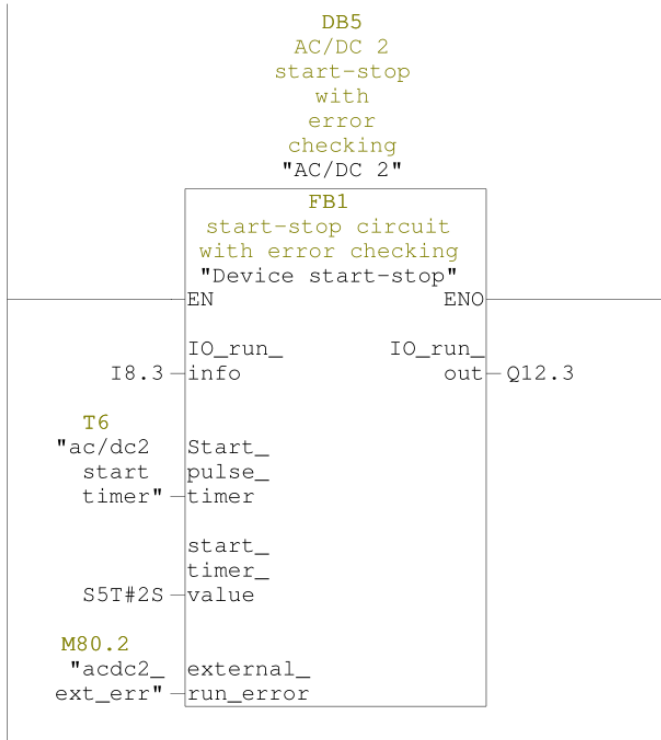
Network: 3



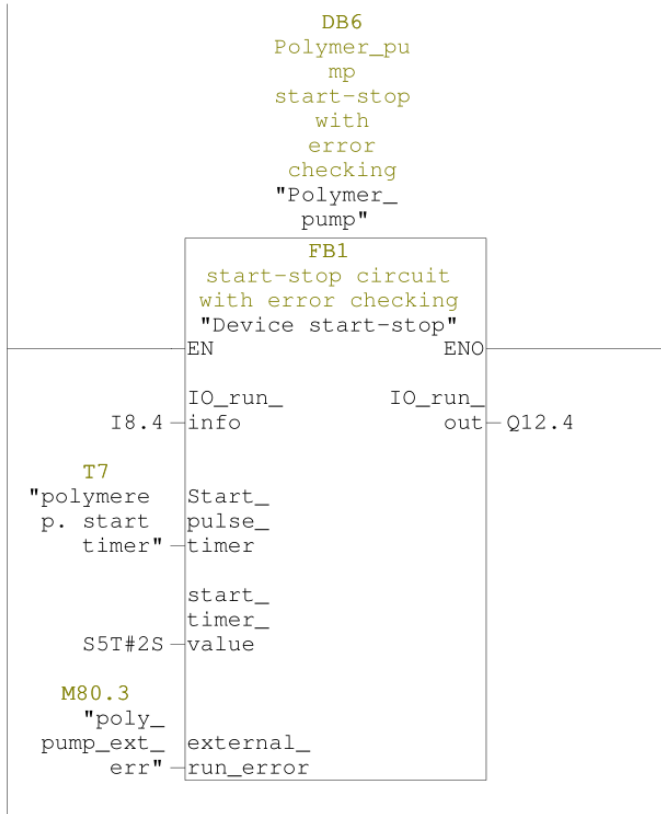
Network: 4



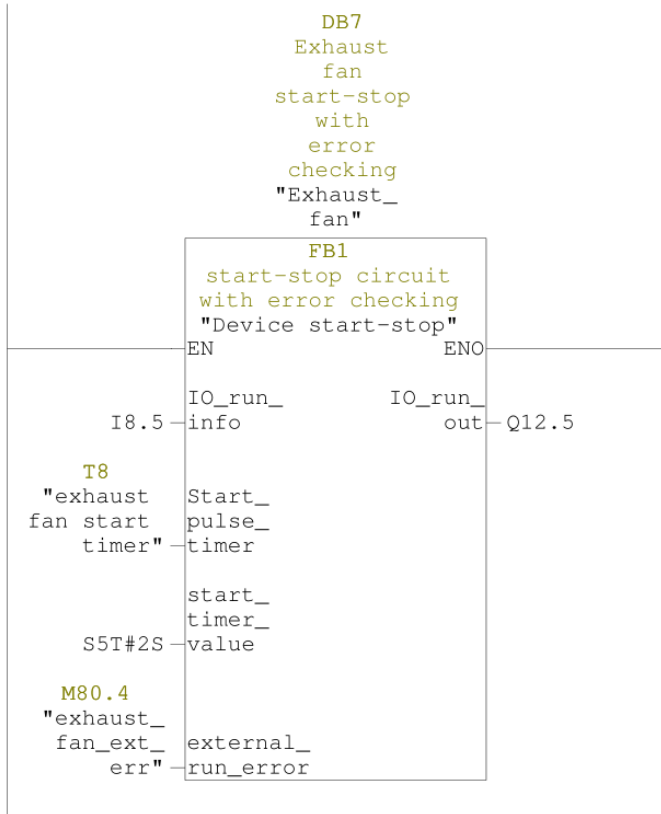
Network: 5



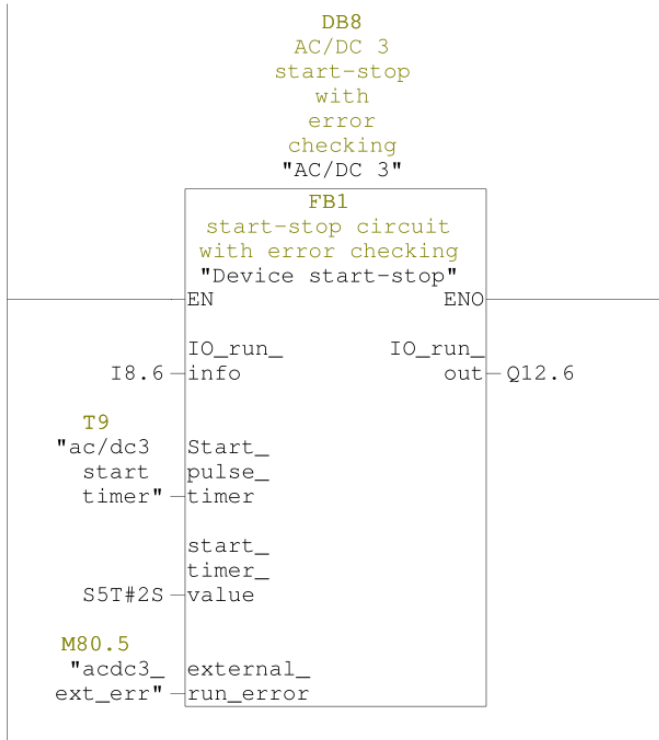
Network: 6



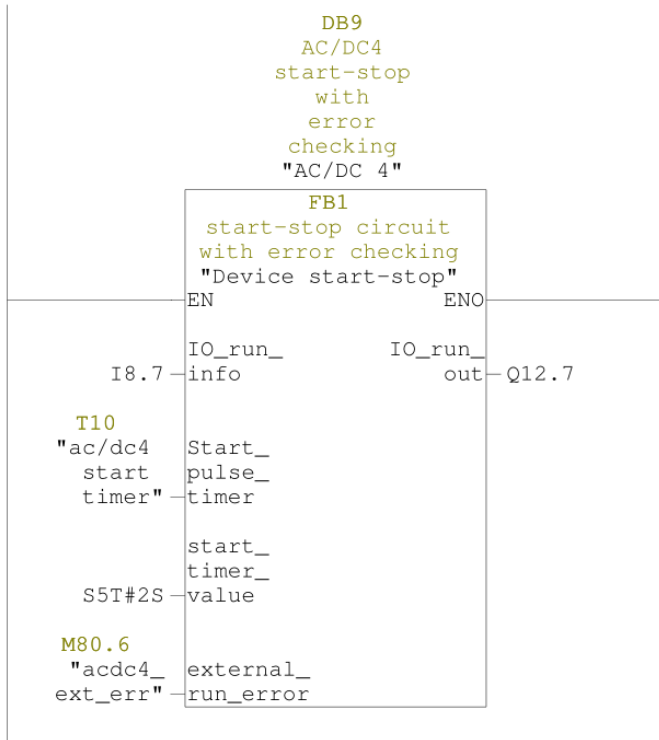
Network: 7



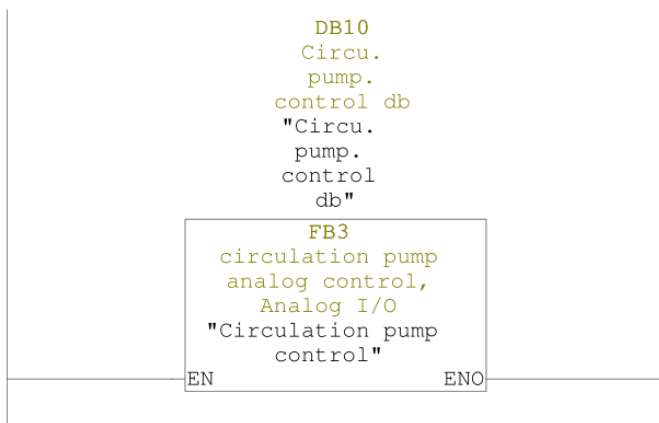
Network: 8



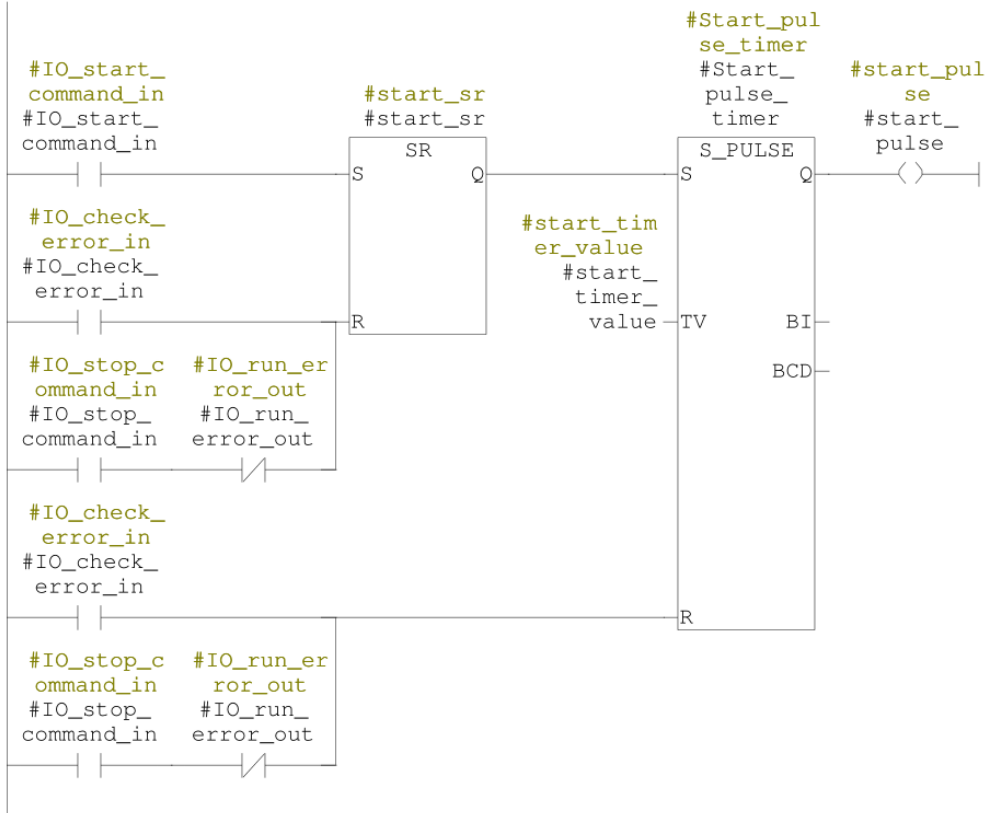
Network: 9



Network: 10



Network: 1



SIMATIC

PureWater_PLC\

09/16/2016 09:10:23 AM

SIMATIC 300\CPU 315F-2 PN/DP\...\FB1 - <offline>

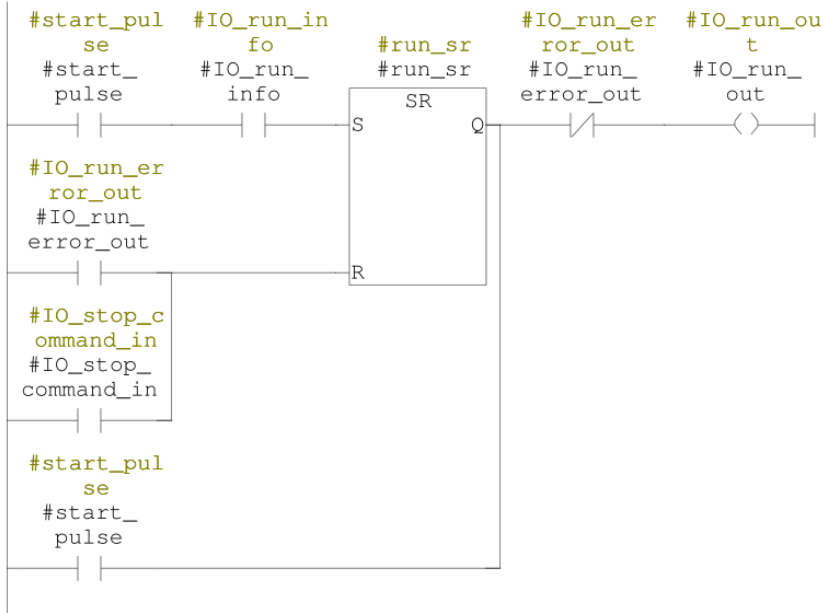
FB1 - <offline>

"Device start-stop" start-stop circuit with error checking
Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 09/05/2016 05:52:31 PM
Interface: 09/05/2016 05:52:31 PM
Lengths (block/logic/data): 00396 00268 00010

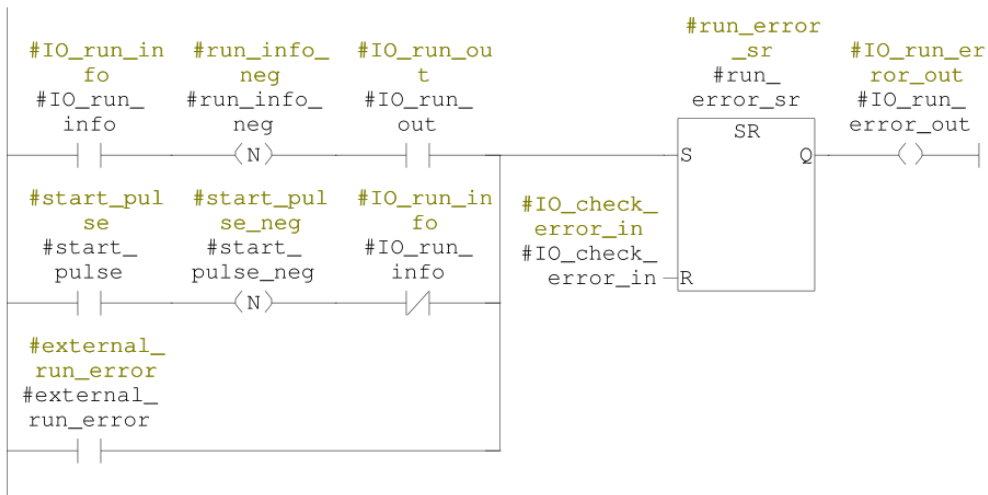
Name	Data Type	Address	Initial Value	Comment
IN		0.0		
IO_run_info	Bool	0.0	FALSE	
Start_pulse_timer	Timer	2.0		
start_timer_value	S5Time	4.0	S5T#0MS	
external_run_error	Bool	6.0	FALSE	
OUT		0.0		
IO_run_out	Bool	8.0	FALSE	
IN_OUT		0.0		
STAT		0.0		
run_error_sr	Bool	10.0	FALSE	
run_sr	Bool	10.1	FALSE	
start_sr	Bool	10.2	FALSE	
IO_start_command_in	Bool	10.3	FALSE	
IO_check_error_in	Bool	10.4	FALSE	
IO_stop_command_in	Bool	10.5	FALSE	
IO_run_error_out	Bool	10.6	FALSE	
run_info_neg	Bool	10.7	FALSE	
start_pulse_neg	Bool	11.0	FALSE	
start_pulse	Bool	11.1	FALSE	
TEMP		0.0		

Block: FB1

Network: 2



Network: 3 run error



SIMATIC

PureWater_PLC\

09/16/2016 09:10:25 AM

SIMATIC 300\CPU 315F-2 PN/DP\...\FB2 - <offline>

FB2 - <offline>

"Valve control"

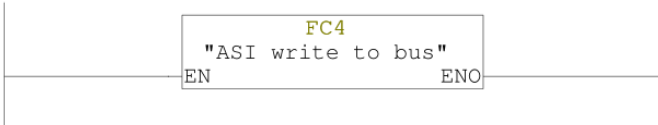
Name:
Author:
Time stamp Code:
Interface:
Lengths (block/logic/data):

Family:
Version: 0.1
Block version: 2
 09/05/2016 12:37:42 PM
 09/05/2016 12:37:42 PM
 00882 00662 00012

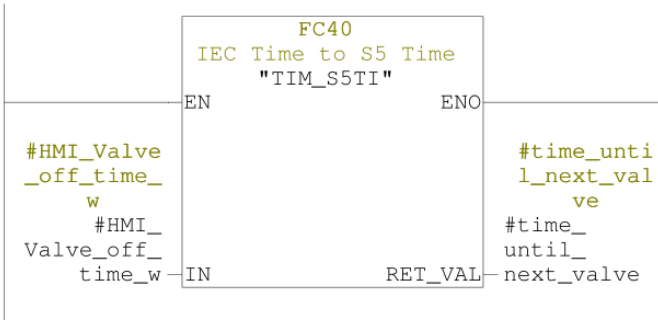
Name	Data Type	Address	Initial Value	Comment
IN		0.0		
OUT		0.0		
IN_OUT		0.0		
STAT		0.0		
valve_on_time	S5Time	0.0	S5T#1S	
time_until_next_valve	S5Time	2.0	S5T#1S	
counter_value_int	Int	4.0	0	
Counter_end_value	Int	6.0	11	
Counter_start_value	Int	8.0	1	
Stop_Reset_cycle	Bool	10.0	FALSE	
Start_cycle	Bool	10.1	FALSE	
HMI_Valve_off_time_w	Time	12.0	T#1S	
HMI_Valve_on_time_w	Time	16.0	T#1S	
HMI_Valve_off_time_r	Time	20.0	T#1S	
HMI_Valve_on_time_r	Time	24.0	T#1S	
neg_temp_2	Bool	28.0	FALSE	
neg_temp_1	Bool	28.1	FALSE	
pos_temp_1	Bool	28.2	FALSE	
Pulse_out	Bool	28.3	FALSE	
neg_temp_3	Bool	28.4	FALSE	
Pulse_to_counter	Bool	28.5	FALSE	
neg_temp_4	Bool	28.6	FALSE	
T1_active	Bool	28.7	FALSE	
T2_active	Bool	29.0	FALSE	
Valvepulse_active	Bool	29.1	FALSE	
Off_timer_BI	Word	30.0	W#16#0	
Off_timer_INT	Int	32.0	0	
TEMP		0.0		

Block: FB2

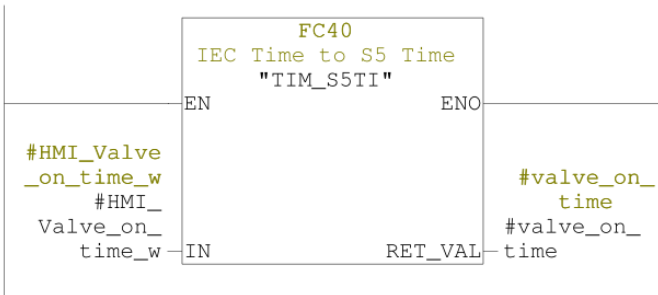
Network: 1
asi write bits



Network: 2 valve off time data type conversion for control
time conversion from hmi to plc
TIME format to S5TIME

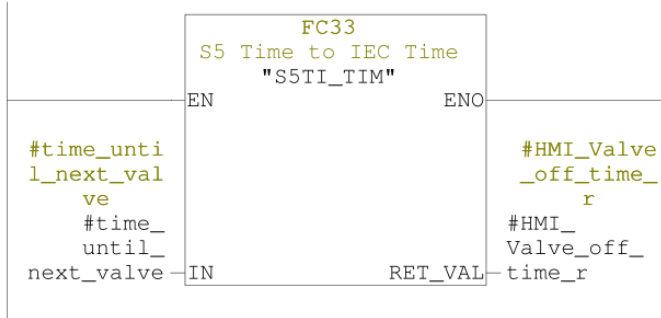


Network: 3 valve off time data type conversion for control
time conversion from hmi to plc
TIME format to S5TIME



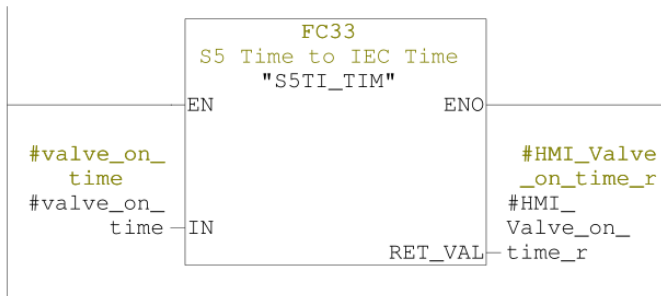
Network: 4 valve off time data type conversion for reading

Time conversion from plc to hmi
s5TIME to TIME conversion for current timer set value

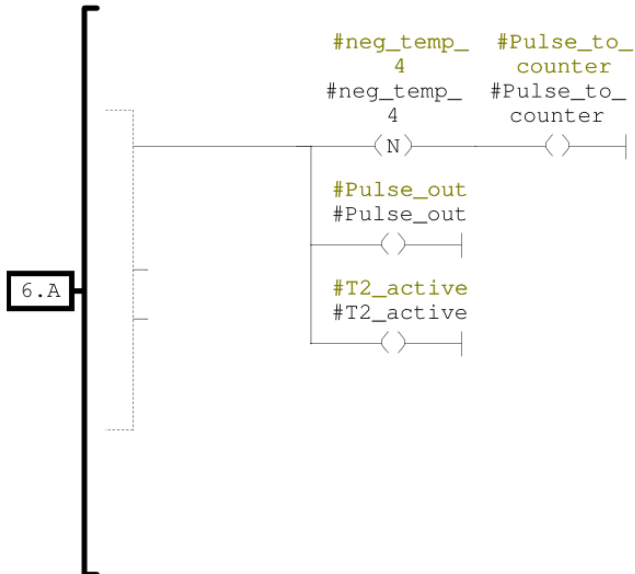
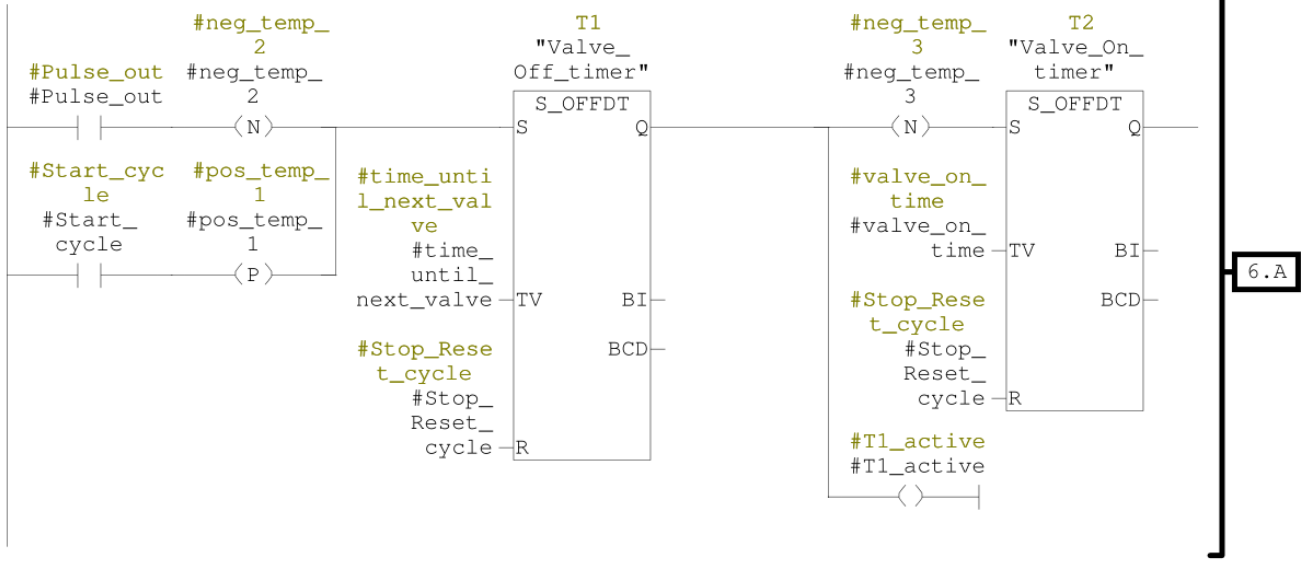


Network: 5 valve on time data type conversion for reading

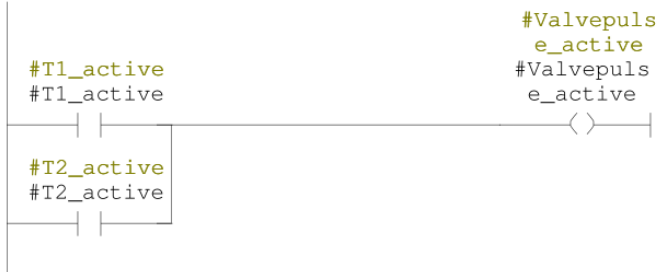
Time conversion from plc to hmi
s5TIME to TIME conversion for current timer set value



Network: 6 Valve pulse generator
 Pulse generator

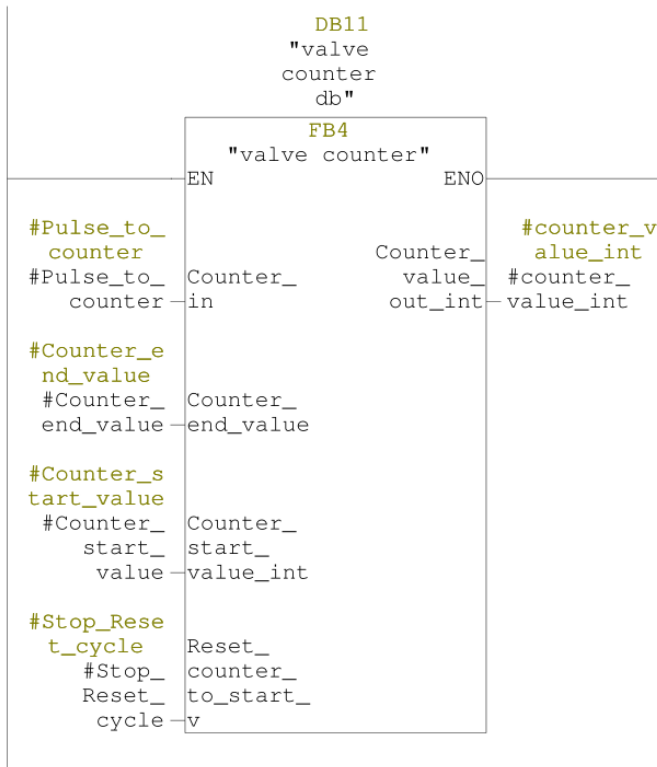


Network: 7



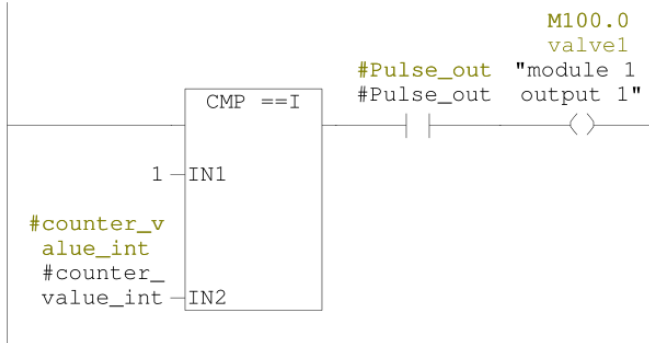
Network: 8 pulse counter

counter count each pulse to change output valve



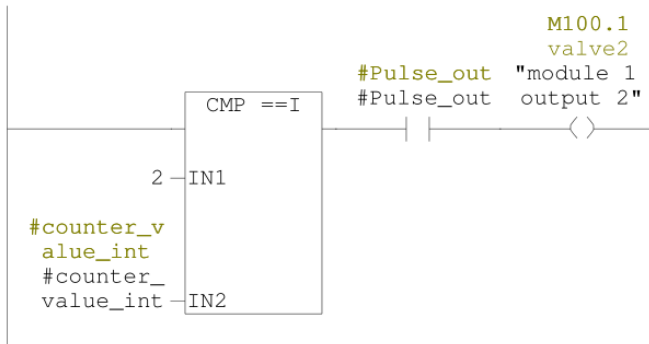
Network: 9 Valve 1

valve control box 1



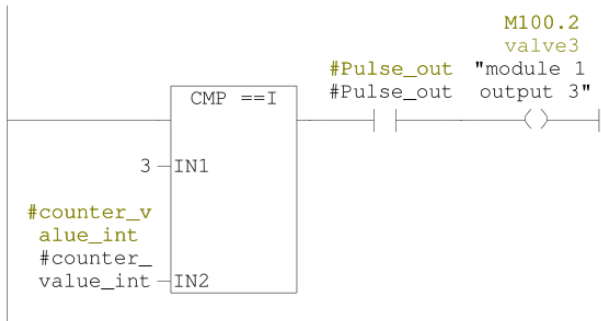
Network: 10 Valve 2

valve control box 1

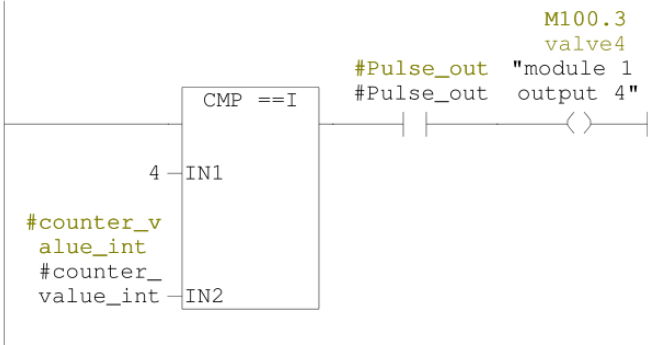


Network: 11 valve 3

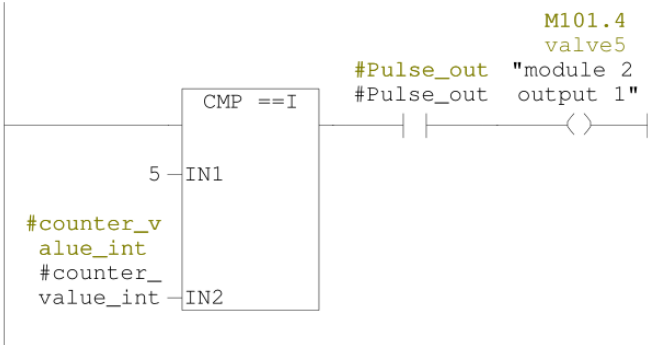
valve control box 1



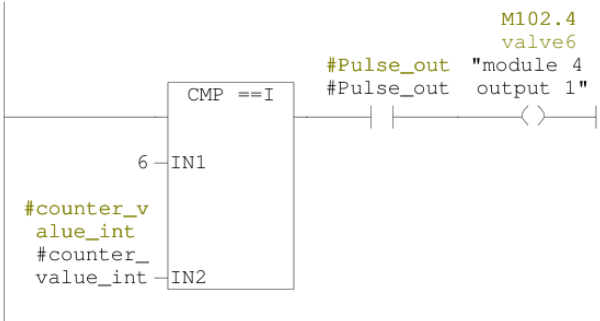
Network: 12 valve 4
 valve control box 1



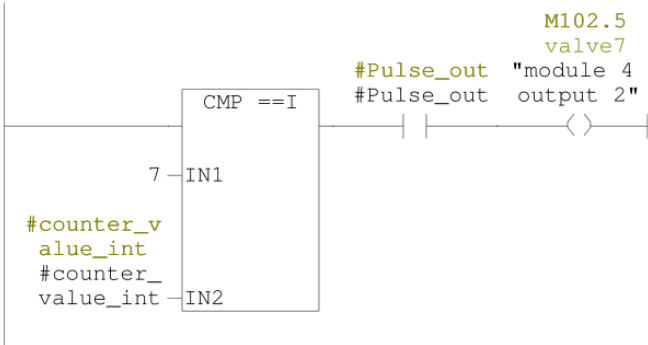
Network: 13 valve 5
 valve control box 1



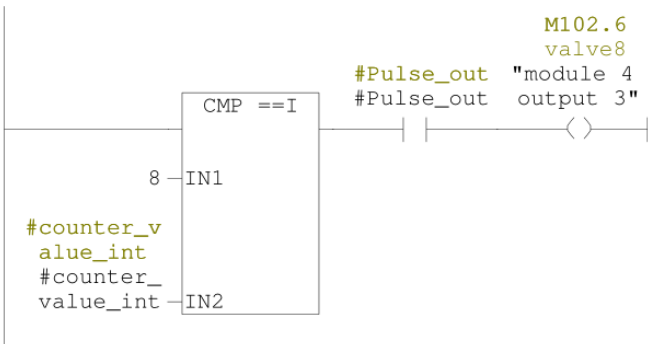
Network: 14 valve 6
 valve control box 2



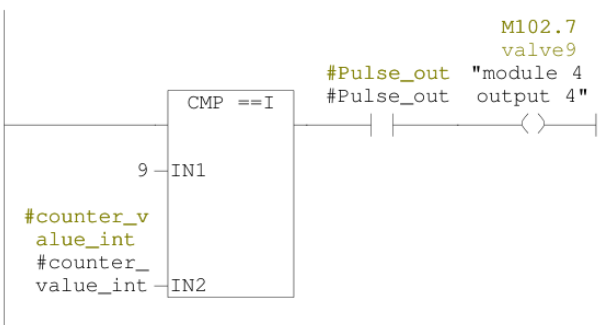
Network: 15 valve 7
 valve control box 2



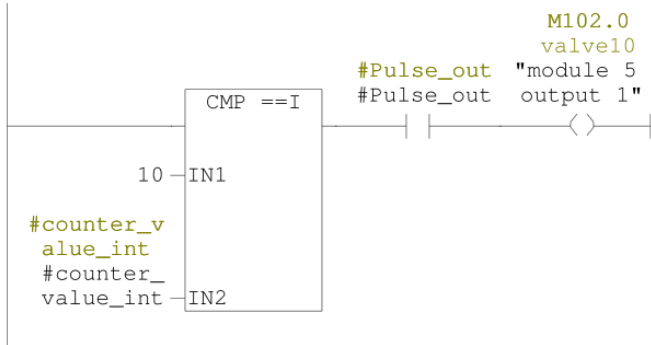
Network: 16 valve 8
 valve control box 2



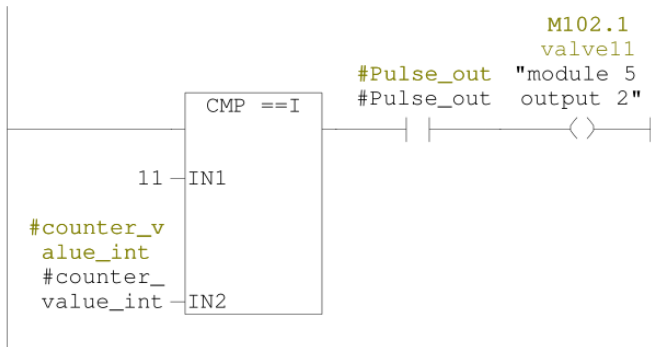
Network: 17 valve 9
 valve control box 2



Network: 18 valve 10
valve control box 2



Network: 19 valve 11
valve control box 2



FB3 - <offline>

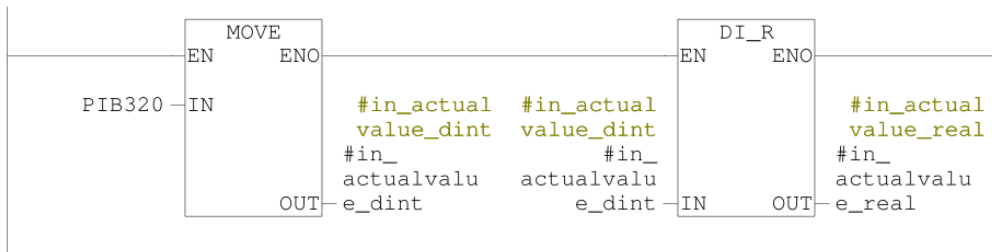
"Circulation pump control" circulation pump analog control, Analog I/O

Name: **Family:**
Author: **Version:** 0.1
Block version: 2
Time stamp Code: 09/06/2016 06:05:17 PM
Interface: 09/06/2016 06:05:17 PM
Lengths (block/logic/data): 00186 00074 00000

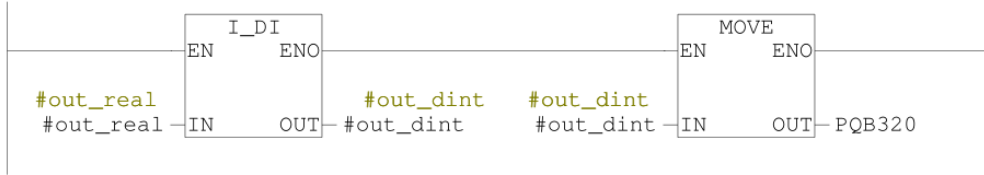
Name	Data Type	Address	Initial Value	Comment
IN		0.0		
OUT		0.0		
IN_OUT		0.0		
STAT		0.0		
in_actualvalue_dint	DInt	0.0	L#0	
in_actualvalue_bcd	DWord	4.0	DW#16#0	
in_actualvalue_real	Real	8.0	0.000000e+000	
out_dint	DInt	12.0	L#0	
out_real	Int	16.0	19	
TEMP		0.0		

Block: FB3

Network: 1



Network: 2



SIMATIC

PureWater_PLC\

09/16/2016 09:10:27 AM

SIMATIC 300\CPU 315F-2 PN/DP\...\FB4 - <offline>

FB4 - <offline>

"valve counter"

Name:
Author:
Time stamp Code:
Interface:
Lengths (block/logic/data):

Family:
Version: 0.1
Block version: 2
 08/10/2016 01:06:30 PM
 08/10/2016 12:43:00 PM
 00218 00102 00000

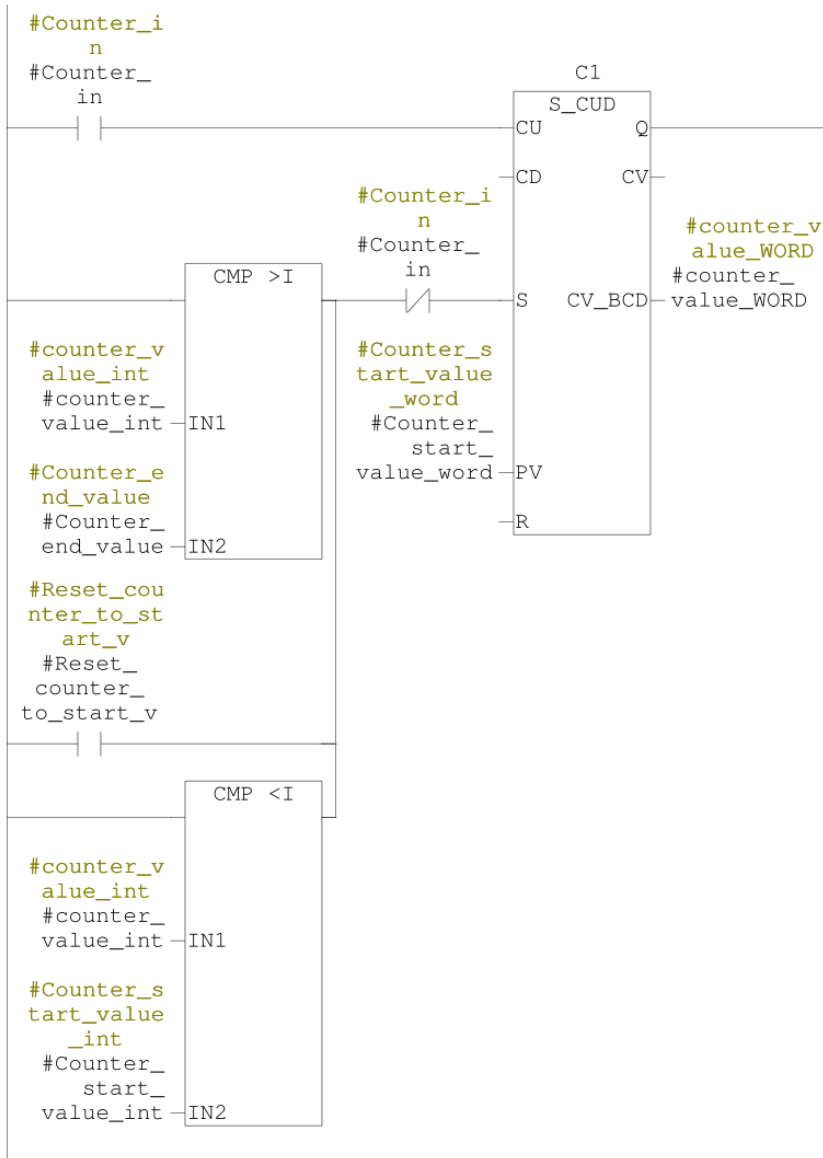
Name	Data Type	Address	Initial Value	Comment
IN		0.0		
Counter_in	Bool	0.0	FALSE	
Counter_end_value	Int	2.0	0	
Counter_start_value_int	Int	4.0	0	
Reset_counter_to_start_v	Bool	6.0	FALSE	
OUT		0.0		
Counter_value_out_int	Int	8.0	0	
IN_OUT		0.0		
STAT		0.0		
counter_value_WORD	Word	10.0	W#16#0	
counter_value_int	Int	12.0	0	
Counter_start_value_word	Word	14.0	W#16#0	
pos_temp1	Bool	16.0	FALSE	
TEMP		0.0		

Block: FB4

Network: 1

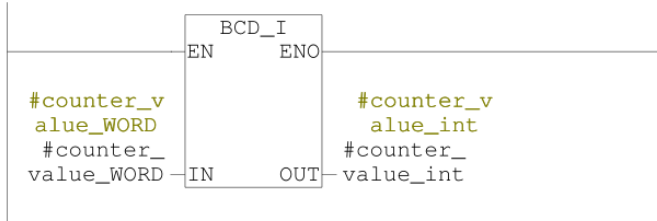
```

counter 1
set new start value: set_counter_start_value
actual counter value: counter_value_out_int
counter start value: counter_start_value_int
counter end value: Counter_end_value
    
```



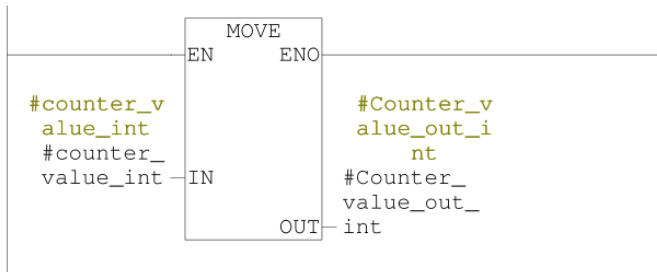
Network: 2

convert counter value word->int



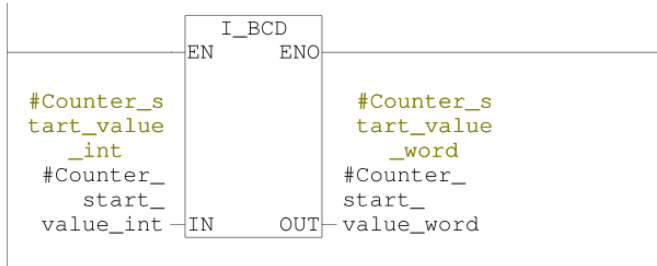
Network: 3

move actual counter value to output



Network: 4

Counter start value conversion int->word



SIMATIC

PureWater_PLC\

09/16/2016 09:10:29 AM

SIMATIC 300\CPU 315F-2 PN/DP\...\FC4 - <offline>

FC4 - <offline>

"ASI write to bus"

Name:
Author:
Time stamp Code:
Interface:
Lengths (block/logic/data):

Family:
Version: 0.1
Block version: 2
 07/05/2016 02:33:20 PM
 07/05/2016 01:48:23 PM
 00118 00026 00000

Name	Data Type	Address	Comment
IN		0.0	
OUT		0.0	
IN_OUT		0.0	
TEMP		0.0	
RETURN		0.0	
RET_VAL		0.0	

Block: FC4

Network: 1

```

L   MB   100
T   PQB  400

L   MB   101
T   PQB  401

L   MB   102
T   PQB  402

L   MB   103
T   PQB  403

```


LIITE 2: AUTOMAATION PIIRIKAAVIOT

