



SEINÄJOEN AMMATTIKORKEAKOULU
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Jaakko-Juhani Jääskö

CIP-pesujärjestelmän ohjauksen modernisointi

Opinnäytetyö

Kevät 2022

Automaatiotekniikan tutkinto-ohjelma



SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Tutkinto-ohjelma: Automaatiotekniikka

Suuntautumisvaihtoehto: Koneautomaatio

Tekijä: Jaakko-Juhani Jääskö

Työn nimi: CIP-pesujärjestelmän ohjauksen modernisointi

Ohjaaja: Matti Perälä

Vuosi: 2022

Sivumäärä: 33

Liitteiden lukumäärä: 0

Opinnäytetyö toteutettiin Atria-Lihavalmiste Oy:lle. Työn tavoitteena oli modernisoida osaston pesukeskuksen toiminnasta vastaava logiikka. Logiikan uusimisella parannettiin prosessin toimintavarmuutta ja korjattavuutta.

Opinnäytetyössä käydään läpi työssä käytettyä termistöä ja tutustutaan CIP-pesun peruseriaatteisiin. Teoriaosuudessa perehdytään myös ohjelmoitavien logiikoiden toimintaan ja niiden ohjelmoimiseen.

Työn lopputuloksena oli valmis mallipohja ohjelmalle, joka tullaan tulevaisuudessa asentamaan pesukeskukseen.

¹ Asiasanat: ohjelmoitavat logiikat, PLC, CIP, Atria

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Degree programme: Automation Engineering

Specialisation: Machine Automation

Author: Jaakko-Juhani Jääskö

Title of thesis: Modernising the Control of the CIP Washing System

Supervisor: Matti Perälä

Year: 2022

Number of pages: 33

Number of appendices: 0

The thesis was done for Atria-Lihavalmiste Oy. The aim of the work was to modernise the logic responsible for the operation of the washing centre. The replacement of the logic needed to be made to improve the reliability and repairability of the process.

In the thesis, a complete model base was designed for the software, which will be installed in the washing centre in the future. The thesis also studied the related terminology and the basic principles of CIP washing. Also the operation and programming of programmable logic were studied.

The thesis provided an overview of the techniques, applications and skills required when modernising a process, such as a washing centre that is being refurbished. The resulting software base and mapping of the washing process operation will make it easy to start the modernization in the future.

¹ Keywords: programmable logic, PLC, CIP, Atria

SISÄLTÖ

Opinnäytetyön tiivistelmä	1
Thesis abstract	2
SISÄLTÖ	3
Kuva-, kuvio- ja taulukkoluetelo	5
Käytetyt termit ja lyhenteet.....	7
1 JOHDANTO	8
1.1 Työn tausta	8
1.2 Työn tavoite.....	8
1.3 Työn rakenne	8
1.4 Työn rajaus	8
1.5 Yritysesittely	9
2 TEORIA.....	12
2.1 Yleistä PLC-logiikkaohjelmoinnista.....	12
2.2 Ohjelmointi	12
2.3 Ohjelmointikieliet.....	13
2.3.1 Kosketinkaavio.....	14
2.3.2 Toimintalohkokaavio	15
2.3.3 Käskylista.....	16
2.4 DCS.....	17
2.5 HMI.....	18
2.6 CIP	19
3 Logiikkajärjestelmät.....	21
3.1 Mitsubishi FX-32MR-ES	21
3.2 Siemens SIMATIC ET 200SP.....	21
3.3 Siemens SIMATIC MANAGER.....	22
3.4 CODESYS.....	23
4 Toteutus	24
4.1 Lähtötilanne.....	24

4.2	Tavoitteet	25
4.3	Logiikkaohjelman suunnittelu	25
5	Yhteenveto	31
LÄHTEET	32

Kuva-, kuvio- ja taulukkoluetelo

Kuva 1. Atrian omistuksessa olevat brändit ja tuotemerkit maakohtaisesti	11
Kuva 2. Siemens LOGO! Ohjelmoitava logiikka.....	12
Kuva 3. Siemens TP700 Comfort HMI-paneeli	19
Kuva 4. Mitsubishi FX-32MR-ES logiikka ja FX-48ER laajennusmoduuli	21
Kuva 5. Siemens SIMATIC ET 200SP lisäkortteineen	22
Kuva 6. Siemensin ohjelmoitava logiikka, virtalähde ja tulo- sekä lähtökortit.....	22
Kuva 7. Kuvaa 6 vastaavat komponentit SIMATIC Managerissa.....	23
Kuvio 1. Atria-konsernin henkilöstömäärän jakautuminen 2021	9
Kuvio 2. Atria-konsernin liikevaihdon jakautuminen alueittain vuosina 2019–2021	10
Kuvio 3. Sulkeutuva ja avautuva kosketin.....	14
Kuvio 4. AND-piiri tikapuukaaviona.....	15
Kuvio 5. OR-piiri tikapuukaaviona	15
Kuvio 6. AND-toimilohko	16
Kuvio 7. OR-toimilohko	16
Kuvio 8. AND-logiikkaoperaatio käskylistana ja relelogiikkapiirinä.....	17
Kuvio 9. Ohjelman alku.....	26
Kuvio 10. Siirtyminen huuhteluohjelmaan	27
Kuvio 11. Huuhteluohjelman kierto	28
Kuvio 12. Pesun loppu	28

Kuvio 13. Vaahdotussekvenssi29

Kuvio 14. Vaahdotussekvenssi **Virhe. Kirjanmerkkiä ei ole määritetty.**

Käytetyt termit ja lyhenteet

I/O	Input/Output, tarkoitetaan sisään- ja ulostuloja sisältävää moduulia
NC	NC tulee sanoista Normally Closed, jolla tarkoitetaan normaalisti suljetussa tilassa olevaa ohjattavaa asiaa, joka aukeaa sen aktivoituessa
NO	NO tulee sanoista Normally Open, jolla tarkoitetaan normaalisti avoimessa tilassa olevaa ohjattavaa asiaa, joka sulkeutuu sen aktivoituessa
CIP	Clean In Place eli CIP on teollisuudessa käytettävä prosessien pesumenetelmä
Muistibitti	Muistibitti pitää ohjelman sisällä sille annetun arvon muistissa
Integer	Kokonaisluku. Ohjelmoinnissa käytetään lyhennettä: INT
Function Block	Käytetään logiikkaohjelmoinnissa ohjelman optimoimiseen

1 JOHDANTO

1.1 Työn tausta

Työn taustana oli uusia kypsytyskaappien pesua ohjaava logiikka. Työ tehtiin Atria Oyj:n Nurmon tehtaalle Atria-Lihavalmiste Oy:lle. Osaston CIP-pesujärjestelmän logiikan ohjaus on toteutettu vanhanaikaisella Mitsubishin logiikalla. Päivittämällä pesua ohjaava logiikka, saadaan se liitettyä osastoa ohjaavaan päälogiikkaan. Näin esimerkiksi pesun kulkua saadaan tarkkailtua etänä ja ohjelmaan saadaan tehtyä muutoksia. Päivittämällä logiikka ennaltaehkäistään myös sen hajoaminen lähitulevaisuudessa ja parannetaan sen korjattavuutta.

1.2 Työn tavoite

Opinnäytetyön tavoitteena oli modernisoida Atrian lihavalmistetehtaalla sijaitsevien kypsytyskaappien pesua ohjaava logiikka. Uusi ohjaus tehdään Siemens ET 200sp logiikan hajautetun I/O:n päälle käyttäen Siemensin SIMATIC Manageria, joka on muutenkin käytössä yrityksessä.

1.3 Työn rakenne

Työn alussa käydään läpi yleisesti Atria Oyj:n taustaa ja liikevaihtoa sekä henkilöstömääriä eri toimipisteissä. Lisäksi perehdytään myös tarkemmin konsernin visioihin ja strategioihin. Seuraavassa luvussa paneudutaan ohjelmoitavien logiikoiden taustaan, niiden toimintaperiaatteisiin ja ohjelmointiin. Teoriaosuudessa kerrotaan myös CIP-pesun historiasta ja peruseriaateista. Kolmannessa luvussa tutustutaan työhön liittyviin logiikkajärjestelmiin ja avataan niiden toimintaa. Viimeisessä luvussa kerrotaan työn etenemisestä ja sen toteutuksesta.

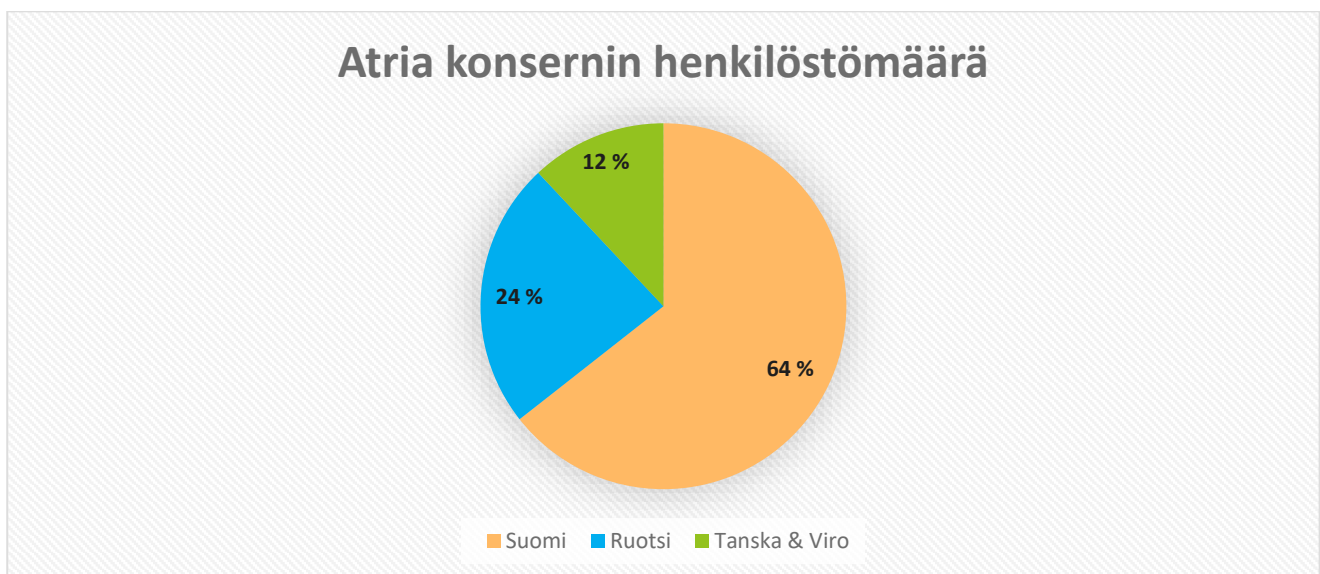
1.4 Työn rajaus

Pesujärjestelmän modernisointiin kuuluu myös sähkökuvien ja sähkökeskuksen uusiminen. Sähkökuvien piirtämisestä sekä sähkökeskuksen suunnittelusta vastasi toinen osapuoli,

jonka kanssa työtä alettiin suunnittelemaan. Tässä opinnäytetyössä keskitytään kuitenkin vain logiikan päivittämiseen. Työ pitää sisällään pesukeskuksen modernisoinnin suunnittelun, jonka pohjalta päivitys tullaan tulevaisuudessa tekemään.

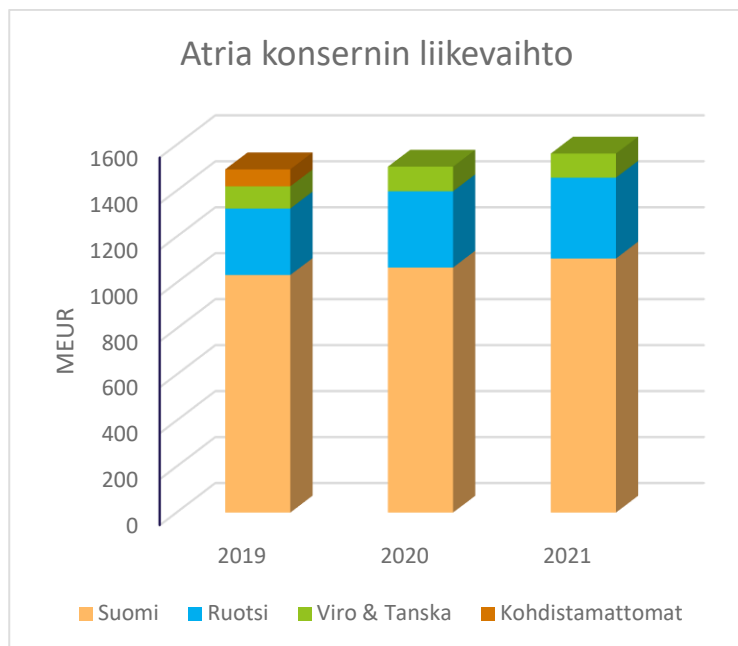
1.5 Yritysesittely

Työ tehtiin Atria Oyj:n Nurmon tehtaalle. Atria on vuonna 1903 perustettu pohjoiseurooppalainen ruokatalo, jonka konserni käsittää toimipisteitä Suomessa, Ruotsissa, Tanskassa ja Virossa. Atria Suomi on näistä merkittävin yksittäinen liiketoiminta-alue. Sen liiketoimintaan kuuluu tuoreiden lihatuotteiden ja muiden elintarvikkeiden valmistusta, kehitystä, myyntiä sekä markkinointia. (Atria 2022a.) Vuonna 2021 Atria-konsernin palveluksessa oli noin 3 700 työntekijää, 2390 Suomessa, 876 Ruotsissa ja 455 Virossa sekä Tanskassa (Atria Oyj 2022b).



Kuvio 1. Atria-konsernin henkilöstömäärän jakautuminen 2021 (Atria Oyj 2022b).

Atria Oyj:n osake on ollut listattuna Nasdaq Helsinki Oy:ssä vuodesta 1991 lähtien. Atrian liikevaihto vuonna 2021 oli noin 1,5 miljardia, josta Atria Suomen osuus oli noin 1 105,7 miljoonaa euroa. (Atria Oyj 2022a.)



Kuvio 2. Atria-konsernin liikevaihdon jakautuminen alueittain vuosina 2019–2021 (Atria Oyj 2022c).

Atrian visio on olla johtava pohjoiseurooppalainen ruokatalo, jonka toteutumista varten on määritelty seuraavia keskeisiä strategisia painopisteitä:

- Siipikarjan markkinajohtajuus Suomessa sekä sen vahvistaminen Ruotsissa
- Laajentaminen valmisruoassa hyödyntämällä asiakas- ja markkinointitietoa
- Markkinoita nopeampi kasvu Foodservice- ja pikaruokasektoreilla vahvistamalla asiakkuuksia, lisäämällä myyntipisteitä sekä kehittämällä uusia tuotteita
- Ruotsin liiketoiminnan kannattavuuden parantaminen ja kasvu
- Punaisen lihan kannattavuuden kasvattaminen parantamalla operatiivista tehokkuutta sekä tuotekategorioiden hallinnalla
- Toimitusketjun tehokkuuden parantaminen sen jokaisella osa-alueella. Myös vahvat arvostetut alueelliset ja kansainväliset tuotemerkit ovat olennainen osa strategiaa. Atrian hallussa olevista tuotemerkeistä ja niiden alabrändeistä lähes kaikilla on ykkös- tai kakkosasema markkinoillaan (Atria Oyj 2022d.)



Kuva 1. Atrian omistuksessa olevat brändit ja tuotemerkit maakohtaisesti (Atria, [viitattu 31.3.2022]).

Liha- ja siipikarjatuotteita valmistava Atria-Lihavalmiste Oy perustettiin vuonna 2004. Yrityksen liikevaihto oli vuonna 2020 yli 26 miljoonaa euroa ja tulos -569 000 euroa. Nettotuloprosentti oli -2,17 %. (Kauppalehti 2022.)

2 TEORIA

2.1 Yleistä PLC-logiikkaohjelmoinnista

Ohjelmoitavasta logiikasta käytetään lyhennettä PLC, joka tulee sanoista Programmable Logic Controller. Se on käytännössä mikroprosessorilla varustettu tietokone, joka on suunniteltu korvaamaan aikaisemmin tuotannossa käytettyjen ohjausjärjestelmien releet. Ohjelmoitava logiikka otettiin teollisuudessa ensin käyttöön autotehtaissa. Sen etuna aikaisemmin käytettyihin releohjauksiin verrattuna on helppo järjestelmien päivitys ilman uudelleenjohtotuksia ja vikadiagnostiikan nopeutuminen, joilla saadaan tuotannon seisokkiaikaa lyhennettyä. Ohjelmoitavia logiikoita on saatavilla laaja valikoima erimallisia ja -laatusia. Kuvassa 2 esimerkkinä Siemensin LOGO! -sarjaan kuuluva PLC. Kaikki sisältävät kuitenkin tulo- ja lähtöportteja, joihin kaikki laitteet kentällä ovat kytkettyinä. (Keinänen ym. 2007, 212.)



Kuva 2. Siemens LOGO! Ohjelmoitava logiikka (Siemens, [viitattu 8.3.2022]).

2.2 Ohjelmointi

Ennen ohjelmoitavia logiikoita ohjausjärjestelmät toteutettiin käyttämällä releohjauksia. Sähkösuunnittelijan tehtävänä oli laatia relekaavio piirustuslaudalle, jonka mukaan sähköasentajat tekivät asianmukaiset kytkennät ohjauskeskuksissa. Releohjauksilla oli mahdollista tehdä samat logiikkatoiminnot, jotka ovat käytössä tänäkin päivänä. Koska releillä toteutetut järjestelmät ovat hyvin monimutkaisia ja hankalia päivittää, niitä ei enää käytetä kuin hyvin yksinkertaisissa järjestelmissä ja koneiden turvatoiminnoissa. (Keinänen ym. 2007, 211.)

PLC-laitteet ovat vapaasti ohjelmitavia logiikoita. Tämä tarkoittaa, että ohjelmitaessa ei ole väliä missä järjestyksessä ohjelman toiminnalliset osat kirjoitetaan. Logiikka voidaan myös saada seuraamaan tiettyä käskyrakennetta askeltavasti. Koska ohjausjärjestelmät ovat usein monimutkaisia sisältäen useita liikkuvia kappeleita eri pääteasemilla, vapaasti ohjelmitava malli tekee ohjelmoinnista helpompaa. (Keinänen ym. 2007, 223.)

2.3 Ohjelmointikielet

Logiikkojen ohjelmoinnissa käytetään nykyään tietokonepohjaisia ohjelmia, joiden käyttöliittymällä ohjelmoiminen on sujuvaa ja monipuolisten ohjelmointiperiaatteiden käyttö mahdollista. Logiikkaohjelmointi perustuu ohjelmointikieliin, joiden peruselementit koostuvat esimerkiksi logiikkaporteista, ajastimista, laskureista ja apumuisteista. Näitä komentoja ei ole standardisoitu, mutta eri ohjelmointiohjelmien käskyjen toimintaperiaate on tarpeeksi samankaltainen, että kun hallitaan yhden valmistajan logiikkakieli, toisen oppiminen onnistuu sen käsikirjaan tutustumalla varsin helposti.

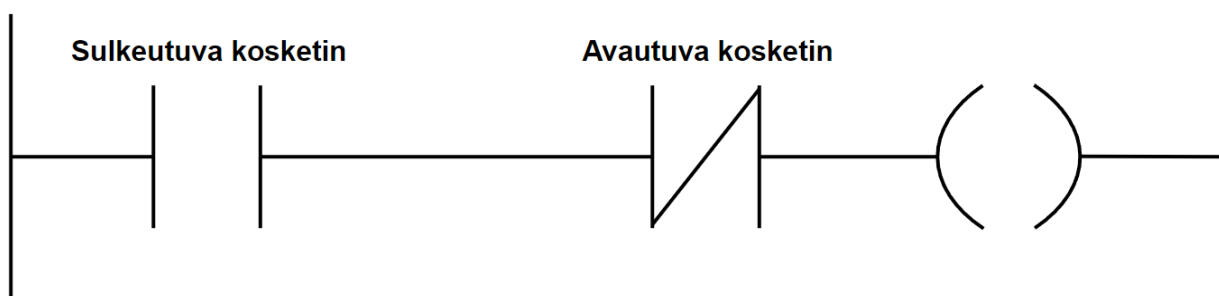
Kaikilla valmistajilla on omat ohjelmointikielet, joten ohjelmointiohjelmoinnista on tullut hyvin kirjavaa ja logiikan tuttuudesta sen yleisin valintaperuste. Tämän vuoksi on laadittu PLC ohjelmointistandardi IEC 61131-3, joka sisältää viisi eri ohjelmointikieltä. Nämä kielet on jaoteltu graafisiin editoreihin sekä tekstieditoreihin. Graafisiin editoreihin kuuluu Sequential Function Chart (SFC) eli toimilohkokaavio, Function Block Diagram (FBD) eli toimintalohko-ohjelmointi ja kosketinkaavioksi kutsuttu Ladder Diagram (LD). Tekstieditoreita ovat Structured Text (ST) eli rakenteellinen tekstieditori ja Instruction List (IL) eli käskylista. Näistä SFC on kuitenkin tarkoitettu enimmäkseen ylemmän tason ohjelmointiin, eikä se tarjoa kovin monipuolisia ominaisuuksia itse loogisten toimintojen ohjaamiseen.

Näistä ohjelmointikielistä ohjelmoija voi käyttää yhtä tai useampaa vaihtoehtoa projektissaan sen mukaan, mikä sopii parhaiten kulloisenkin toiminnan ohjelmoinnissa. Luetelluista ohjelmointiperiaatteista yleisimmin käytettyjä ovat "Statement List" eli käskylista (STL), kosketinkaavio (LD) ja toimintalohko-ohjelmointi (FBD). (Keinänen ym. 2007, 223–225.)

2.3.1 Kosketinkaavio

Kosketinkaaviosta käytetään myös nimitystä tikapuukaavio (Keinänen ym. 2007, 224). Ohjelmalla tuotetaan koodia järjestelemällä graafisia ohjelmointielementtejä, jotka koostuvat pääasiassa relekoskettimista, käämeistä ja laatikoista. Ohjelman rakentaminen alkaa vasemmalta sivulta virtakiskosta ja päätetään oikeaan reunaan nollakiskoon. (Berger 2003, 104.)

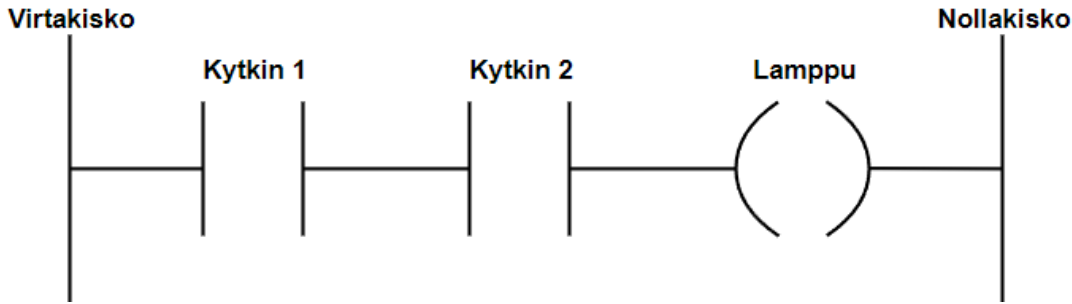
Relekoskettimia voidaan käyttää selvittämään tietyn osoitteen tila ja siten muodostamaan loogisia operaatioita. Koskettimia on kahdenlaisia, avautuvia ja sulkeutuvia. Avautuvien koskettimien läpi kulkee virta, kun siihen osoitetun muuttujan tila on aktiivinen. Sulkeutuva kosketin taas päästää virran läpi, kun siihen osoitettu muuttuja ei ole aktiivinen. (Berger, 2003, 104.)



Kuvio 3. Sulkeutuva ja avautuva kosketin

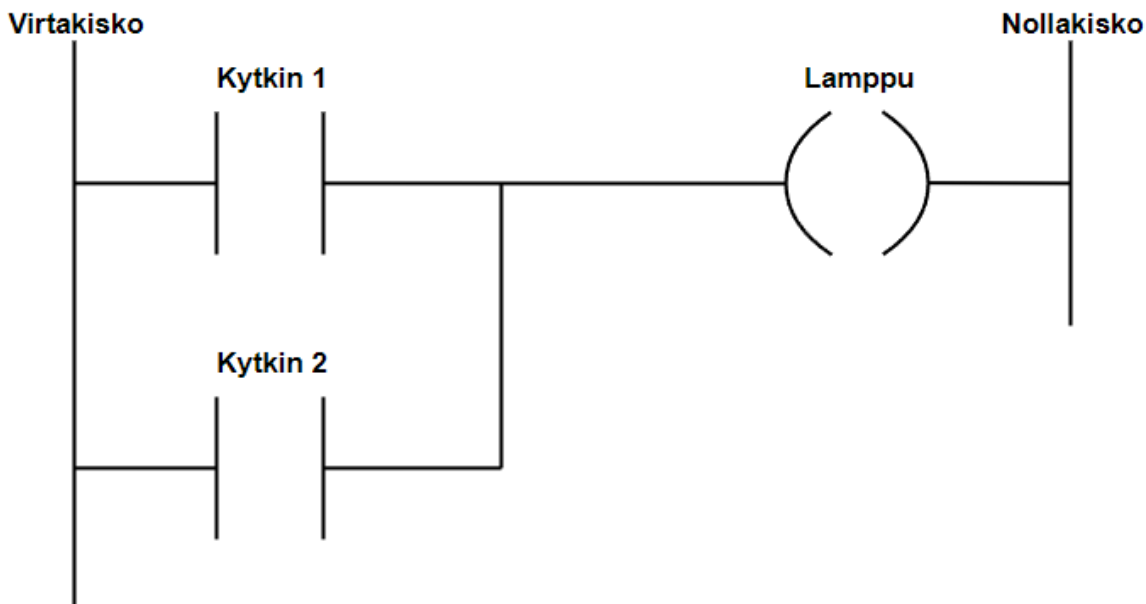
Käämeillä voidaan asettaa tai resetoida tietyn bittiosoitteen, kuten ulostulon, tila. Yksinkertaisilla käämeillä ulostulo pysyy päällä niin kauan, kun virta kulkee piirissä. Käämejä voidaan myös käyttää pitämään haluttua osoitetta päällä kauemmin "Set" komennolla. Käämejä käytetään myös ohjaamaan esimerkiksi ajastimia, laskureita ja kutsumaan eri lohkoja. (Berger, 2003, 105.)

Kuviossa 4 on malli tikapuukaaviolla muodostetusta AND-piiristä: kytkimien 1 ja 2 ollessa aktiivisena koskettimet sulkeutuvat ja virta pääsee kulkemaan lamppuun, jolloin se syttyy.



Kuvio 4. AND-piiri tikapuukaaviona

Kuviossa 5 on malli tikapuukaaviolla muodostetusta OR-piiristä: kytkimen 1 tai 2 ollessa aktiivisena virta pääsee kulkemaan koskettimien läpi sytyttäen lampun.

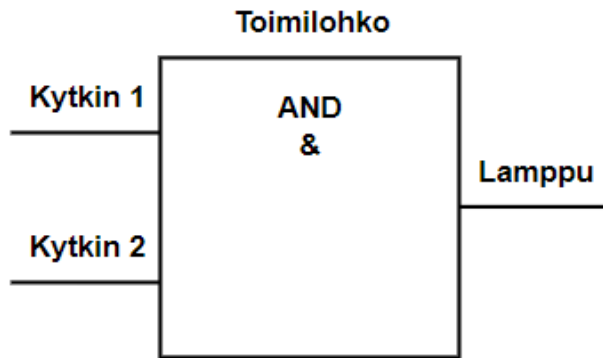


Kuvio 5. OR-piiri tikapuukaaviona

2.3.2 Toimintalohkokaavio

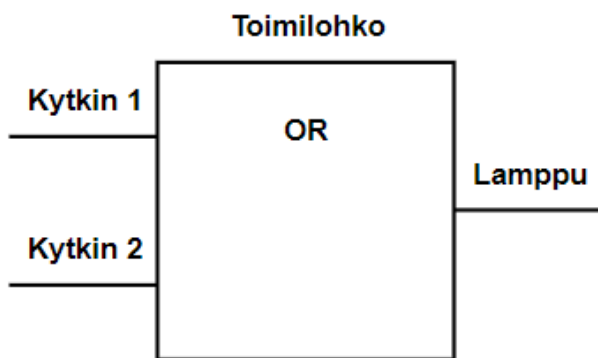
Toimilohkoja käyttävä toimintalohko-ohjelmointi muistuttaa näöltään mikropiireillä toteutettua ohjainkortin kaaviota. Toimintalohkot sisältävät samoja loogisia funktioita, joita esimerkiksi tikapuukaavion relekoskettimilla voidaan muodostaa. Yleisimpiä käytettyjä toimilohkoja ovat AND-, OR- ja NOT-portit. Käytössä on myös laaja valikoima esimerkiksi erilaisia laskureita ja ajastimia. (Keinänen ym. 2007, 224.)

Kuviossa 6 on malli toimintalohkokaaviolla muodostetusta AND-piiristä. Kytkimien 1 ja 2 ollessa aktiivisena ulostulona oleva lamppu syttyy.



Kuvio 6. AND-toimilohko

Kuviossa 7 on malli toimintalohkokaaviolla muodostetusta OR-piiristä. Kytkimien 1 tai 2 ollessa aktiivisena ulostulona oleva lamppu syttyy.



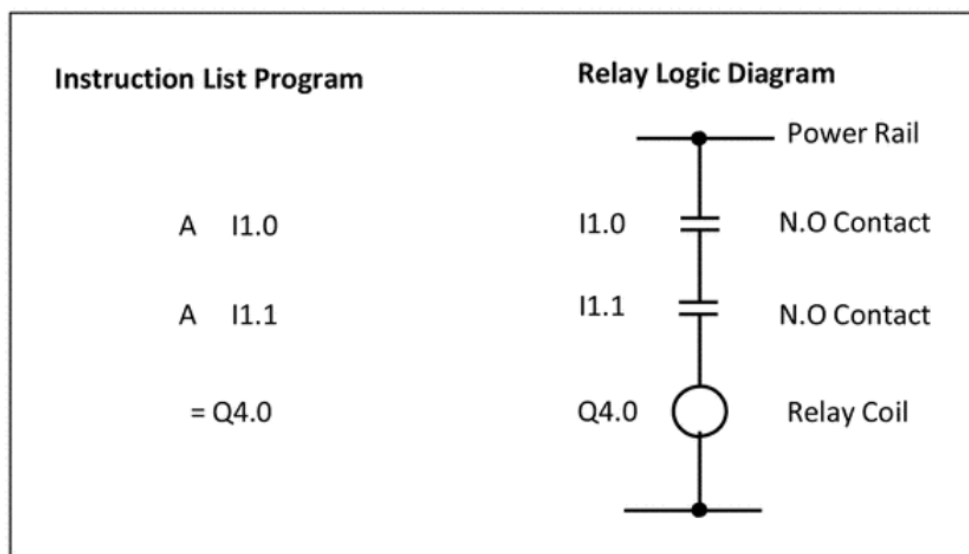
Kuvio 7. OR-toimilohko

2.3.3 Käskylista

Basic tai Pascal ohjelmointikieltä muistuttava käskylistaohjelmointi koostuu yksinkertaisista tekstimuotoisista komennoista. Se on niin sanottu rakenteellinen tekstieditori, jonka loogiset perustoiminnot IF-THEN-ELSE muodostavat ohjelman perusrakenteen. (Keinänen ym. 2007, 224.)

Siemensin S7 ohjelmistopakettissa on valittavissa yli 130 erilaista käskyä ja laaja valikoima osoitteita, ohjelmoitavasta logiikasta riippuen. Yksinkertaisimmat IL-ohjelmakielen käskyt ovat binäärimuotoisia. Nämä käskyt toteuttavat loogisia operaatioita yhden PLC:n muistibitin kanssa. Peruskäskyt palauttavat tiedon siitä, onko signaali aktiivinen (1) vai ei aktiivinen (0). Tällaisia käskyjä kutsutaan myös relelogiikkaohjeeksi (relay logic instruction), koska ne toteuttavat komentoja, jotka voivat korvata relelogiikkapiirin. (Hughes 2015, 366.)

Kuviossa 8 on esimerkki AND-logiikkaoperaatiosta IL-ohjelmointikielellä ja relelogiikkapiirinä. Lähtö Q4.0 aktivoituu, kun molemmat tulot, I1.0 ja I1.1, ovat aktiivisena.



Kuvio 8. AND-logiikkaoperaatio käskylistana ja relelogiikkapiirinä (Hughes 2015, 366)

2.4 DCS

Työssä käytettävä Siemensin SIMATIC ET 200SP hajautettu I/O liitetään osaston keskuslogiikkaan väylätekniikan avulla. Kyseinen moduuli on edullinen ja modulaarisuutensa vuoksi helposti korjattavissa tulo- tai lähtökorttien vaurioituessa. Sen yhteensopivuus päälogiikan kanssa tekee myös asentamisesta ja käyttöönotosta helppoa verrattuna eri sarjan malleihin.

DCS tulee sanoista Distributed Control System, joka tarkoittaa hajautettua ohjausjärjestelmää. Laajemmissa järjestelmissä, missä on useita eri ohjattavia prosesseja, käytetään usein hajautettua ohjausjärjestelmää. Tällaisessa järjestelmässä on useita itsenäisiä ohjauskeskuksia, hajautettuja I/O-moduuleja, jotka on asennettu ohjattavien prosessien lähelle. Monimutkaisia järjestelmiä on myös mahdollista ohjata käyttämällä ainoastaan ohjelmitavia logiikoita, mutta useissa tapauksissa hajautettu ohjausjärjestelmä on taloudellisesti kannattavampaa. Asentamalla itsenäinen ohjain lähelle kenttälaitteita, saadaan ohjattavat komponentit liitettävä suoraan hajautettuun ohjaimeen, jonka kautta prosessia ohjataan päälogiikalta. Lisäksi hajautuksen myötä tuotanto ei ole vain yhden ohjelmitavan logiikan varassa, vaan tuotanto voi jatkua prosessin jonkin vaiheen

vikaantumisesta huolimatta. Hajautettu I/O liitetään päälogiikkaan tyypillisesti yhden Ethernet-kaapelin avulla, mikä on huomattavasti yksinkertaisempaa kuin johdottaa kaikki kenttälaitteet erikseen. (RealPars, [viitattu 10.4.2022].)

2.5 HMI

Modernisoinnin yhteydessä pesukeskuksessa olevaan kuvan 3 mukaiseen Siemensin TP700 Comfort HMI-paneeliin lisätään mahdollisuus tarkastella pesun tilaa sen aikana ja säätää pesuohjelman muuttujia.

Automatisoitujen prosessien käyttäjien on hyvä saada informaatiota prosessin kulusta sen aikana. Ennen digitaalisia tietokoneita, nopeita yhteyksiä ja näyttöpaneeleita, prosessin komponenteilta oli suora yhteys käsin rakennettuihin kuvaajiin ohjaushuoneessa. Tällaista menetelmää käytetään nykyään harvemmin ja vain pienissä vakioprosesseissa.

Nykyään käytössä olevaa metodia kutsutaan HMI:ksi, joka tulee sanoista Human Machine Interface. Ohjelman avulla prosessin kulusta saadaan reaaliaikaista tietoa käyttäjälle. Yleensä näyttöpaneelilla on näkyvissä kokonaiskuva prosessin kulusta ja siihen liittyvien komponenttien tilasta. Ohjelmapohjaiselta graafiselta näytöltä voidaan myös esimerkiksi ohjata prosessia, päivittää muuttujia, seurata hälytyksiä, lähettää viestejä tai selata historiikkaa.

Useimmat HMI-ohjelmapaketit mahdollistavat käyttäjän räätälöidä ilmoituksia aktivoitumaan, joko digitaalisten I/O-bittien tai analogisten signaalien ylittäessä asetetut raja-arvot. Hälytyksiin täytyy reagoida nopeasti ja tehdä tarvittavat toimenpiteet tuotteen vaurioitumisen tai vaaratilanteen välttämiseksi. Ohjelma voidaankin suunnitella siten, että käyttäjän on reagoitava hälytykseen. Ilmoitukset raja-arvojen ylittymisestä myös auttavat minimoimaan prosessin seisonta-aikaa. Ohjelman tallentaessa prosessin tapahtumat, ongelmakohtiin voidaan puuttua myös historiaa selaamalla. Näin saadaan myös suoritettua laaduntarkkailua.

Tyypillisesti ilmoitukset ja prosessin tilan muutokset tallennetaan palvelimelle kronologisessa järjestyksessä. Palvelimet käyttävät tehokasta pakkausmenetelmää tilan

säästämiseksi. Suuret prosessit saattavat kuitenkin tuottaa raporteja huomattavia määriä, jonka vuoksi käyttäjän on asetettava enimmäistallennusaika tilan säästämiseksi.

HMI-ohjelmisto tarjoaa peruselementtejä, kuten nappeja, kytkimiä ja informaation visualisoinnin mahdollistavia kuvaajia, joilla voidaan rakentaa käyttöliittymä, joka vastaa kontrolloitavaa prosessia. Objekteja voidaan linkittää ohjausjärjestelmän sisäisiin muuttujiin animointia ja ohjausta varten. Ohjelmiston kirjastosta käyttäjä voi valita tai itse suunnitella erilaisia ohjausmoduuleita tai visualisointikomponentteja. Ohjelmoitaessa käyttöliittymää, voidaan sitä testata virtuaalisesti liittämättä sitä oikeaan laitteistoon. Simuloitaessa analogisia ja digitaalisia arvoja on niitä mahdollista muuttaa vastaamaan kentällä tapahtuvia muutoksia. (Hughes 2015, 367–370.)



Kuva 3. Siemens TP700 Comfort HMI-paneeli (Siemens, [viitattu 27.4.2022]).

2.6 CIP

CIP-pesujärjestelmä on saanut alkunsa meijerituotannon tarpeesta saada laitteistot puhdistettua usein, nopeasti ja tasaisella laadulla. Järjestelmää käytetäänkin nykyään laajemmissa prosesseissa, joissa samat pesut toistuvat useasti ja niiden tulee tapahtua nopeasti. Lyhenne CIP tulee englanninkielisistä sanoista Clean In Place, jolla tarkoitetaan paikan päällä tapahtuvaa laitteistojen puhdistamista ilman niiden purkamista. Koska

puhdistusprosessin aikana laitteistoa ei tarvitse purkaa, säästyy työvoimaa, mikä parantaa laitoksen tuottavuutta. Manuaaliseen pesuun verrattuna säästyy myös vettä ja pesuainetta, etenkin jos pesuneste tai huuhteluvesi kierrätetään uudelleenkäyttöä varten. Automatisoinnin ansioista saadaan myös vähennettyä vaarallisten kemikaalien kanssa työskentelyä.

CIP-pesujärjestelmä kehitettiin alun perin käytettäväksi nestemäisiä ruoka-aineita käsittelevien linjojen puhdistamiseen. Nykyään järjestelmä on kuitenkin laajasti käytössä myös muissa erilaisissa ja eri tuotteita käsittelevissä linjastoissa ja laitteissa. Sen käyttö on kuitenkin mahdollista vain, jos tuotantoprosessi on alun perin suunniteltu siihen. Pesujärjestelmää varten laitteiston on kestävä pesun aikana käytettäviä kemikaaleja, ja suunnittelussa on varmistuttava siitä, että kaikki prosessin lopputuotteen kanssa kosketuksissa olevat materiaalit tulevat puhtaaksi.

CIP-pesujärjestelmä koostuu vähintään yhdestä pumpusta, putkistosta ja suuttimista, joilla vaikuttavia aineita siirretään, sekä säiliöistä, jossa pesuaineita ja vettä säilötään. Nesteiden ja laitteiston kuumentamista varten tarvitaan myös lämmittimiä. Lisäksi järjestelmän venttiilien ohjausta varten käytetään paineilmaa. Pesun kiertoa ohjaa ohjelmoitava logiikka. Järjestelmä voi sisältää myös antureita ja mittareita, joiden avulla voidaan tarkkailla pesun kulkua ja lopputulosta. Näiden avulla ohjelma voi myös säätää eri pesukiertovaiheiden kestoa ja pesuaineen määriä.

Tyypillinen CIP-pesukierto sisältää useita vaiheita ja alkaa alkuhuuhTELulla, jolla irrotetaan kiinteä lika. Sen jälkeen kuumaa emäksistä pesuainetta kierrätetään järjestelmässä, jotta loput kiinteät aineet saadaan sulatettua ja liuotettua pois. Seuraavaksi pestävästä kohteesta huuhdellaan emästä sisältävä pesuaine pois vedellä. Tämä vaihe on tärkeä, koska jäljelle jäävä emäs neutralisoisi seuraavassa vaiheessa käytettävän happaman pesuaineen ja heikentäisi sen vaikutusta. Happokierron jälkeen vuorossa on desinfiointiaineen kierrätys, jonka jälkeen järjestelmä huuhdellaan huolellisesti vedellä. (Robinson 2000, 1806–1815.)

3 Logiikkajärjestelmät

3.1 Mitsubishi FX-32MR-ES

Nykyinen pesukeskuksen ohjaus on rakennettu kuvassa 4 näkyvän Mitsubishiin FX-32MR-ES ohjelmoitavan logiikan päälle. Siihen on liitetty samalta valmistajalta FX-48ER laajennusmoduuli lisäämään lähtö- ja tuloporttien määrää. Laajennusmoduuli saa virtansa PLC:n sisäänrakennetusta 24 VDC -virtalähteestä. FX-32MR-ES-logiikka sisältää yhteensä 32 kappaletta I/O-paikkoja ja laajennusmoduuli 48 kappaletta. Tuloportit ottavat vastaan 24 VDC -signaaleja ja lähtöportit toimivat releohjauksina. (Mitsubishi 2022.)



Kuva 4. Mitsubishi FX-32MR-ES logiikka ja FX-48ER laajennusmoduuli

3.2 Siemens SIMATIC ET 200SP

Pesukeskuksen toiminnasta vastaavaksi moduuliksi valikoitui Siemensin SIMATIC ET 200SP. Tämä hajautettu I/O keskustelee osaston päälogiikan (S7-400) kanssa ja vastaa pesukeskuksen toiminnasta.

SIMATIC ET 200SP on skaalautuva I/O-järjestelmä, johon voidaan liittää useita eri sisään- ja ulostulokortteja digitaalisten ja analogisten signaalien lukemista ja välittämistä varten. Valittavana on myös erilaisia kenttälaitteiden ohjaukskortteja, kuten esimerkiksi moottorin käynnistämiseen, pneumatiikkaan tai paikoitukseen tarkoitettuja kortteja.

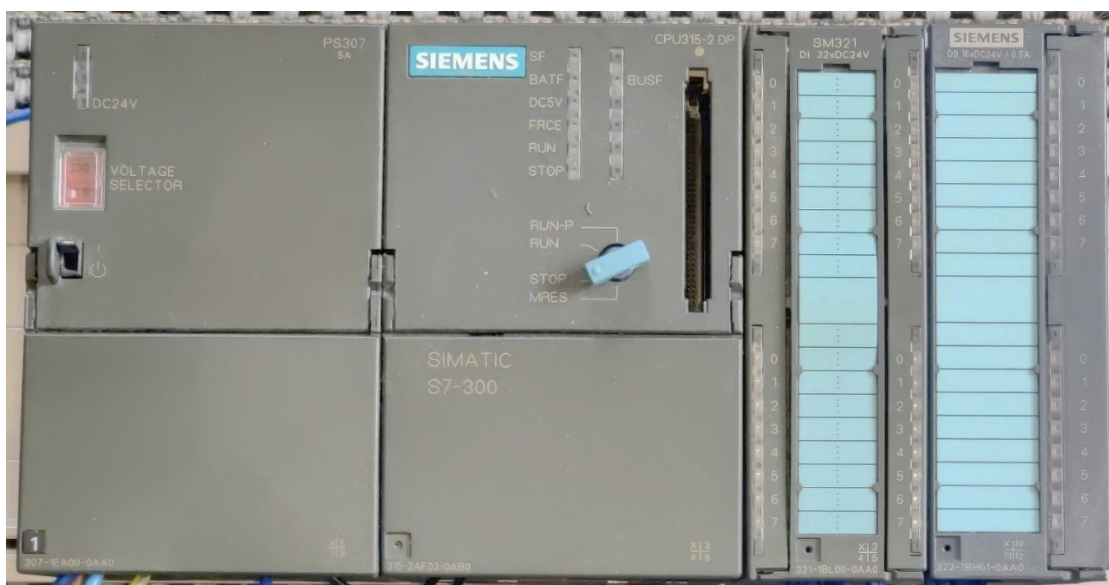
Hajautettuihin ohjauksiin soveltuva SIMATIC ET 200SP ei itse vastaa prosessin toiminnasta vaan se on yhteydessä ylemmän tason ohjelmoitavaan logiikkaan, jonka kautta se suorittaa käskyjä ohjelman mukaisesti. Yhteys PLC:hen muodostetaan käyttäen Ethernet-pohjaista tietoliikenneprotokollaa, kuten PROFINETiä tai PROFIBUSia. (Siemens, [viitattu 10.4.2022]).



Kuva 5. Siemens SIMATIC ET 200SP lisäkorteineen (Siemens, [viitattu 3.4.2022]).

3.3 Siemens SIMATIC MANAGER

Prosessin logiikka ohjelmoidaan käyttäen Siemensin Simatic Manageria. Se on olennainen työkalu STEP 7 automaatio-ohjelmassa, joka kuuluu Siemensin SIMATIC-tuoteperheeseen. SIMATIC Managerilla työskennellessään käyttäjällä on käytössään STEP 7 -automaatio-ohjelmistoon sisältyvät tuotteet. Ohjelman sisällä valitaan komponentit vastaamaan kentällä käytössä olevia tuotteita. (Berger 2003, 53.) Kuvassa 7 on esitetty kuvaa 6 vastaavat komponentin SIMATIC Manager-ohjelmistossa.



Kuva 6. Siemensin ohjelmoitava logiikka, virtalähde ja tulo- sekä lähtökortit

Slot	Module	Order number	Firmware	MPI address	I address	Q address
1						
2	CPU315-2 DP(1)	6ES7 315-2AH14-0AB0	V3.0	2		
X2	<i>DP</i>				<i>204,7"</i>	
3						
4	DI32xDC24V	6ES7 321-1BL00-0AA0			0...3	
5	DO16xDC24V/0.5A	6ES7 322-1BH01-0AA0				4...5

Kuva 7. Kuvaa 6 vastaavat komponentit SIMATIC Managerissa

3.4 CODESYS

Työn ohjelmointiosuus toteutettiin lopulta CODESYS-ohjelmointisovelluksella. CODESYS sisältää IEC 61131-3 standardin määrittämät ohjelmointikielet ja on toimintaperiaatteeltaan samankaltainen kuin lopullisessa työssä käytettävä SIMATIC Manager. Tällä ohjelmalla tehty pohja on siis helposti siirrettävissä lopulliseen työhön oikeiden tulo- ja lähtöporttien varmistuessa.

4 Toteutus

4.1 Lähtötilanne

Työ aloitettiin kenttäkartoituksella, jolla selvitettiin pesukeskuksen toimintaperiaate ja pesua ohjaavien komponenttien merkitys. Modernisoitavan pesukeskuksen tehtävänä on ohjata useamman eri kypsytyskaapin pesukiertoa vaiheittain. Yhden tai useamman kaapin pesu valitaan tarvittaessa kaappien vieressä olevalta käyttöliittymältä. Tämä ohjaa releen kautta tiedon pesukeskuksen logiikalle siitä, mitkä kaapit halutaan pesuun.

Pesu alkaa alkulämmityksellä, ja sen aikana pestävien kaappien päällä olevat kiertoilmapuhaltimet lämmittävät ja kostuttavat kaapit höyryn avulla pesua varten. Alkulämmityksen jälkeen kaappeja aletaan vaahdottamaan yksi kerrallaan numerojärjestyksessä ensimmäisestä alkaen. Esimerkiksi pestäessä kaapit kaksi ja kolme aloitetaan vaahdotus kaapista numero kaksi.

Vaahdotusta varten logiikka ohjaa pesukeskuksen pesuaine-, vesi- ja paineilmaventtiilin auki. Samaan aikaan ohjataan pesukeskuksessa oleva pumppu päälle sekä kypsytyskaapin päällä olevan vinoistukan magneettiventtiili (NC) auki, jotta vahto saadaan suihkutettua kaapin sisälle. Määrätyn ajan kuluttua venttiilit, pumppu, vinoistukka ja kiertoilma ohjataan pois päältä.

Vaahdotuksen jälkeen ohjelma siirtyy odottamaan, että lipeää sisältävä vahto ehtii vaikuttaa. Vaahdotus tehdään joka kaapille useamman kerran. Pesukierron aikana kaapit käyvät läpi myös huuhtelusekvenssejä ja lopuksi suoritetaan suihkuohjelma, jolla varmistetaan, ettei lipeää jää kaappiin.

Pesukierron aikana pesukeskuksessa olevan vesisäiliön pintaa tarkkaillaan pinta-anturilla. Säiliön saavuttaessa määrätyn alarajan magneettiventtiili ohjaa säiliöön lisää vettä, kunnes yläraja on saavutettu.

4.2 Tavoitteet

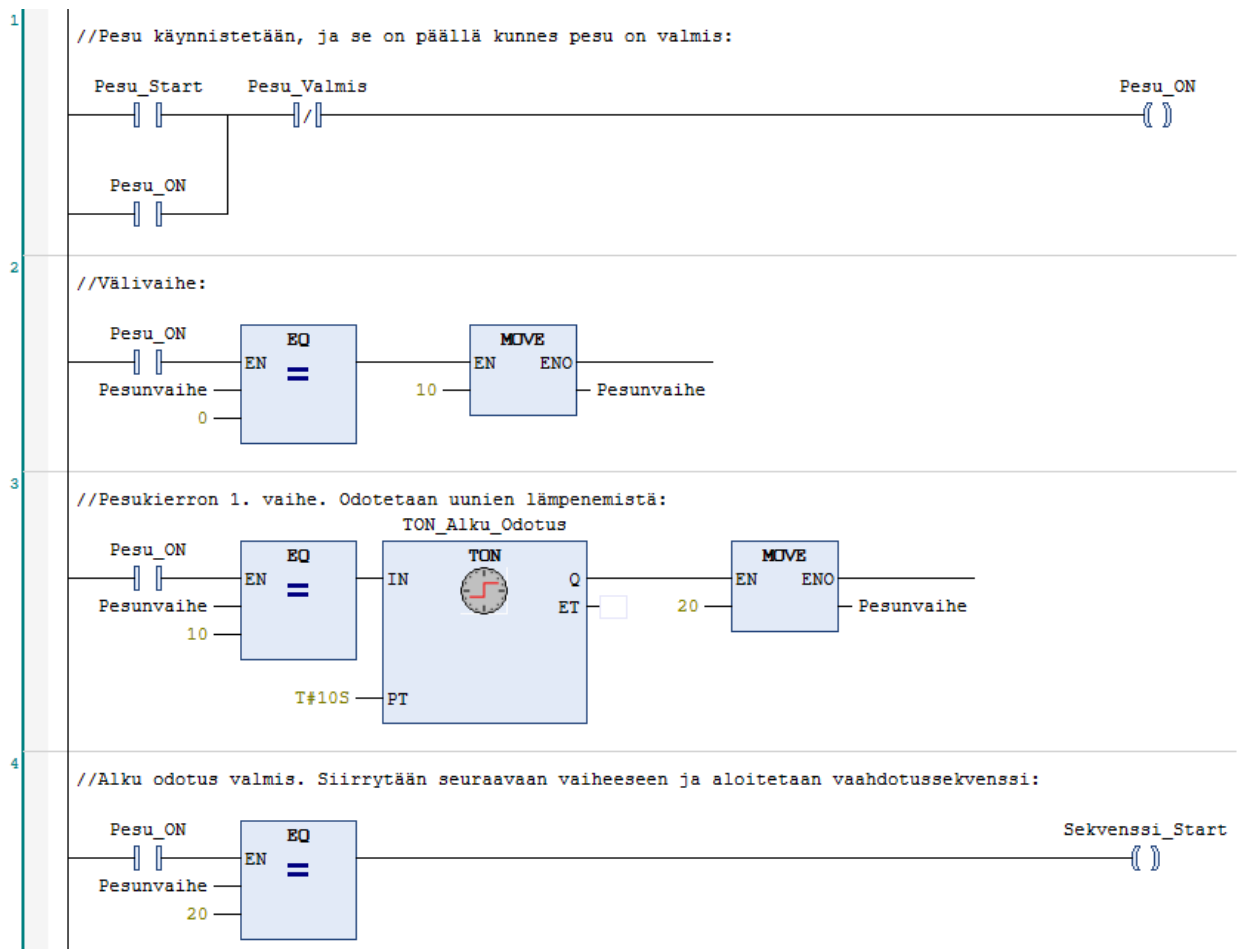
Opinnäytetyön alkuperäisenä tavoitteena oli suunnitella uusi logiikka pesukeskukseen SIMATIC Managerilla ja käyttöönottaa se sähkötoiden valmistuttua. Tarkoituksena oli suunnitella toimiva logiikkaohjelma, joka sisältää oikeat lähtö- ja tulomuuttujat oikeissa osoitteissa.

Covid-19-pandemian aiheuttaman komponenttipulan vuoksi Siemensin tuotteiden toimitusajat viivästyivät kuitenkin niin paljon, että työtä ei saatu toteutettua täysin loppuun. Työn aikana myös SIMATIC Managerin lisenssi vanhentui eikä sen uusiminen ollut järkevää. Ohjelmointi suoritettiin tästä syystä ilmaisella CODESYS-ohjelmistolla. Pesun kulkua ohjaava logiikka saatiin kuitenkin tehtyä valmiiksi samalla periaatteella ja ohjelmointikielellä, jolla se tullaan tekemään Siemensin ohjelmistolla komponenttien saapuessa.

Tulevaisuudessa pesukeskuksen ohjaus tullaan uusimaan täydellisesti, joten tässä työssä tavoitteena oli saada pesun kulku toimimaan kuten aikaisemminkin. Modernisoinnilla parannettiin prosessin huollettavuutta ja toimintavarmuutta sekä mahdollistettiin pesun kulun seuraaminen helposti.

4.3 Logiikkaohjelman suunnittelu

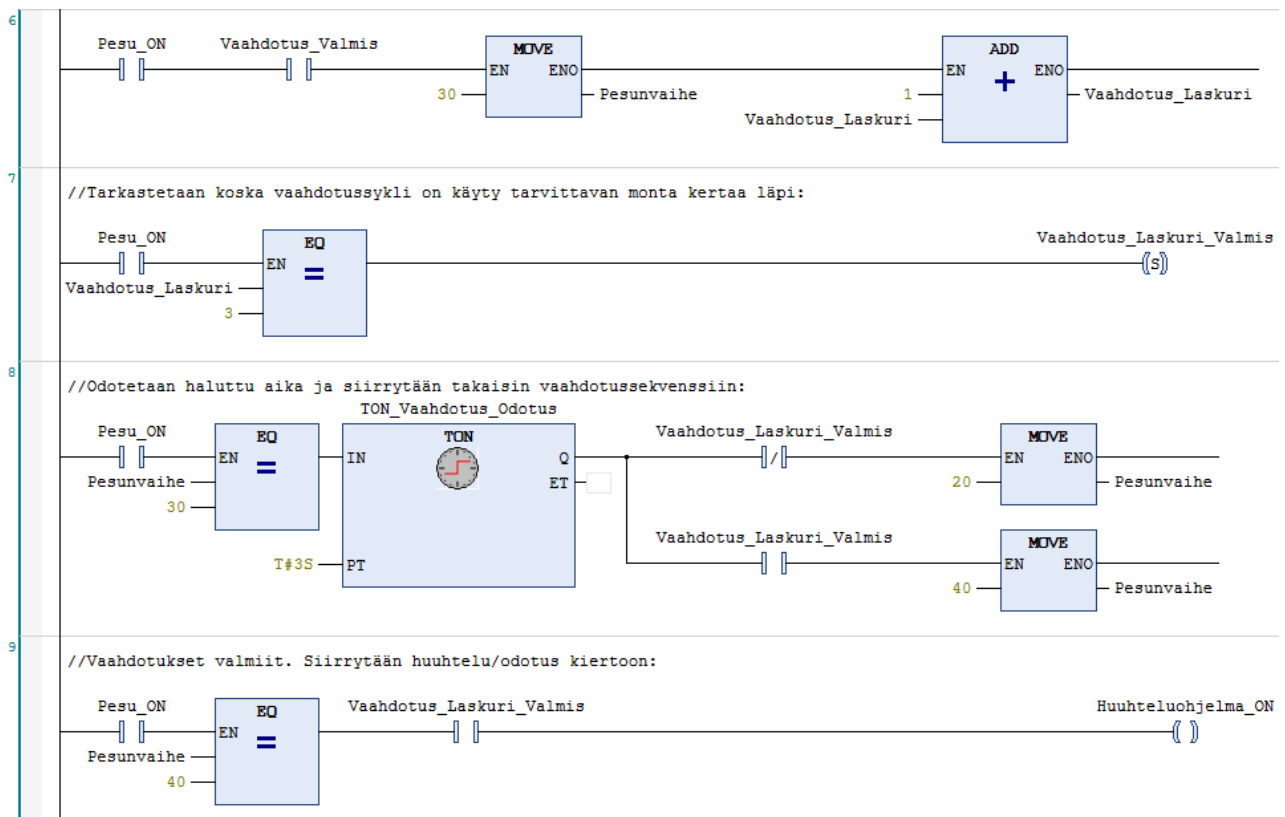
Pesun kulusta vastaava logiikkaohjelmointi toteutettiin CODESYS-suunnitteluohjelmistolla, LAD-kielellä. Ohjelma suunniteltiin käytettäväksi pohjana ja mallina tulevaisuudessa toteutettavaan pesukeskuksen modernisointiin. Ohjelmoinnin tavoitteena oli saada pesukierto toimimaan samalla tavalla kuin nykyisessä prosessissa. Näin ollen pesu aloitetaan tässäkin ohjelmassa alkuodotuksella, jonka aikana höyryn annetaan lämmittää kaapit pesua varten. Alkuodotus saatiin aikaiseksi kuvion 9 mukaisesti siirtymällä vaahdotussekvenssiin vasta ohjelman alussa olevan ajastimen "TON_Alku_Odotus" määräajan umpeuduttua. Ohjelmassa on käytetty paljon hyödyksi mahdollisuutta siirtyä vaiheesta toiseen muuttamalla määrätyn muuttujan arvoa eri pesukierron vaiheissa. Kuviossa 9 nähtävä apumuuttuja "Pesunvaihe" on integer-tyyppinen muuttuja, jonka arvoa muuttamalla "MOVE"-käskyllä siirytään ohjelmassa eri vaiheisiin.



Kuvio 9. Ohjelman alku.

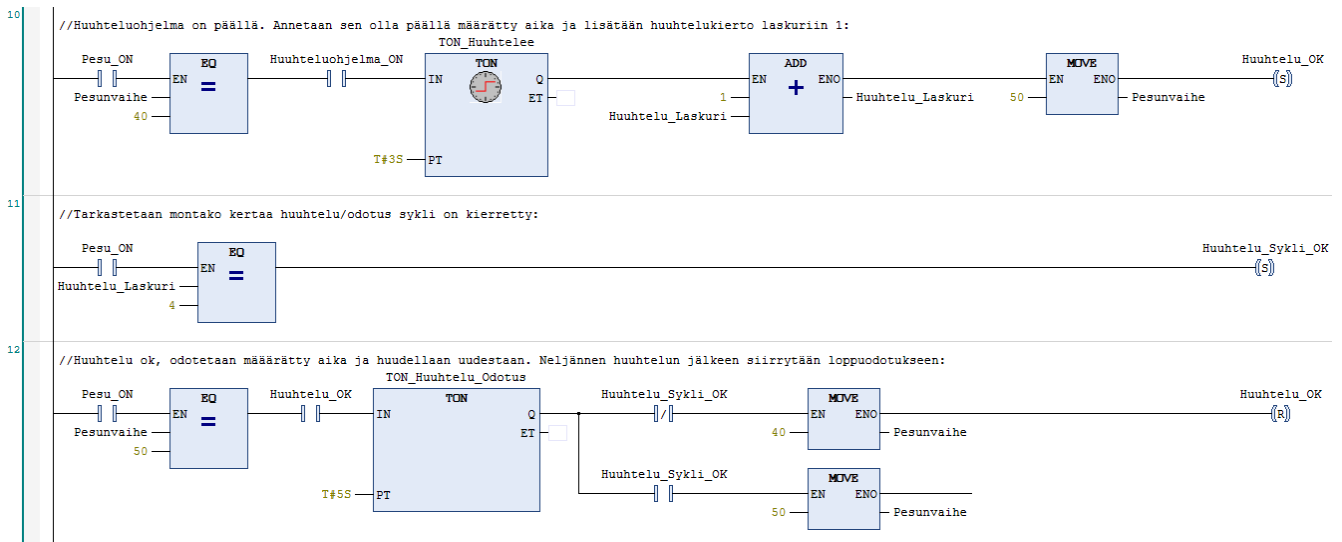
Vaahdotussekvenssi on toteutettu Function Blockilla (FB), jota pääohjelma ohjaa ja lukee. FB:n toiminnasta kerrotaan lisää myöhemmin. Alkuodotuksen jälkeen alkava sekvenssi käy yksitellen pesussa olevat kaapit lävitse ja pitää kyseistä kaappia vastaavan apumuuttujan päällä vaahdotuksen ajan. Tämän apumuuttujan avulla saadaan ohjattua tarvittavat ulostulot päälle vaahdotuksen ajaksi. Kun vaahdotussekvenssi on käynyt kaikki pesussa olevat kaapit läpi, se merkitsee sekvenssin valmiiksi, jolloin ohjelma saadaan siirtymään seuraavaan vaiheeseen kuvion 10 ensimmäisen osion mukaisesti. Samalla kasvatetaan integer-tyyppisen "Vaahdotus_Laskuri"-apumuuttujan arvoa yhdellä. Vertaamalla kyseistä muuttujaa haluttuun arvoon saadaan seurattua, koska vaahdotussykli on käyty riittävän monta kertaa lävitse. Vaahdotusten jälkeen lipeän annetaan vaikuttaa määrätty aika. "TON_Vaahdotus_Odotus"-ajastimella, määrätyn ajan umpeuduttua, siirrytään takaisin vaahdotussykliin. Vaahdotus-odotus kierto pysyy päällä, kunnes haluttu määrä vaahdotussyklejä on käyty läpi. Tällöin "Vaahdotus_Laskuri"-muuttujan arvo vastaa

määrättyä arvoa, jolloin "Vaahdotus_Laskuri_Valmis"-muuttuja asetetaan aktiiviseksi. Sen ollessa päällä "Pesunvaihe"-muuttujan avulla siirrytään huuhteluohjelmaan.



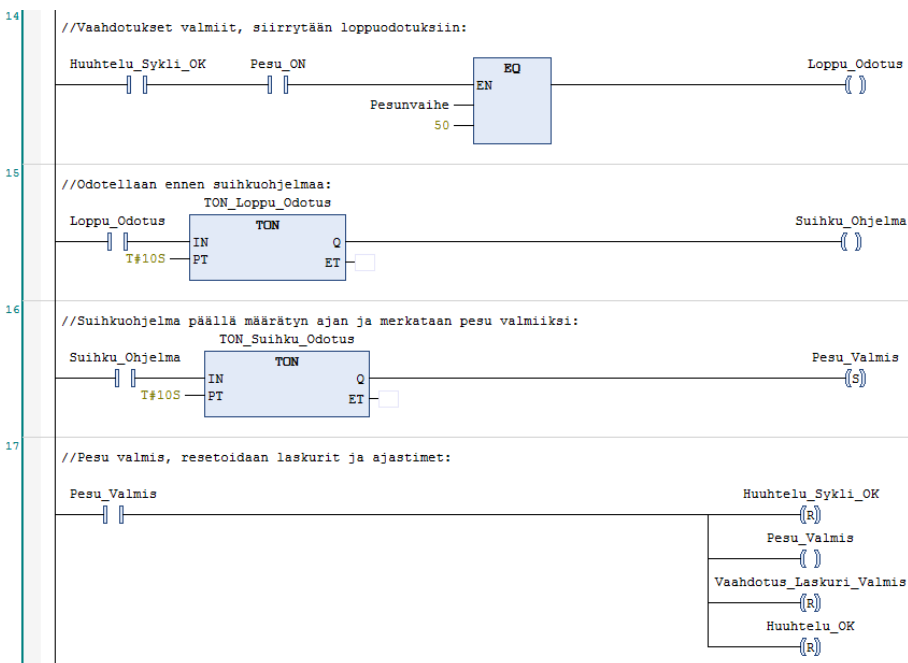
Kuvio 10. Siirtyminen huuhteluohjelmaan

Huuhteluohjelman ollessa päällä "Huuhtelu_ON"-apumuuttuja pysyy aktiivisena. Tällä muuttujalla saadaan ohjattua halutut ulostulot päälle huuhtelun aikana. Kuvion 11 mukaisesti huuhtelun annetaan olla päällä "TON_Huuhtele" -ajastimen määräämän ajan. Määrätyn ajan jälkeen kasvatetaan "Huuhtelu_Laskuri"-apumuuttujaa samalla tapaa kuin vaahdotussyklissä ja merkitään huuhtelusykli valmiiksi. Huuhtelun jälkeen odotetaan tietty aika, jotta vesi kaapeissa ehtii laskea ennen seuraavaa huuhtelua. Huuhtelu-odotuskiertoa käydään läpi, kunnes laskuri saavuttaa määrätyn arvon ja "Huuhtelu_Sykli_OK"-asetetaan aktiiviseksi. Tämän muuttujan ollessa aktiivinen siirrytään "Pesunvaihe"-muuttujan avulla seuraavaan vaiheeseen.



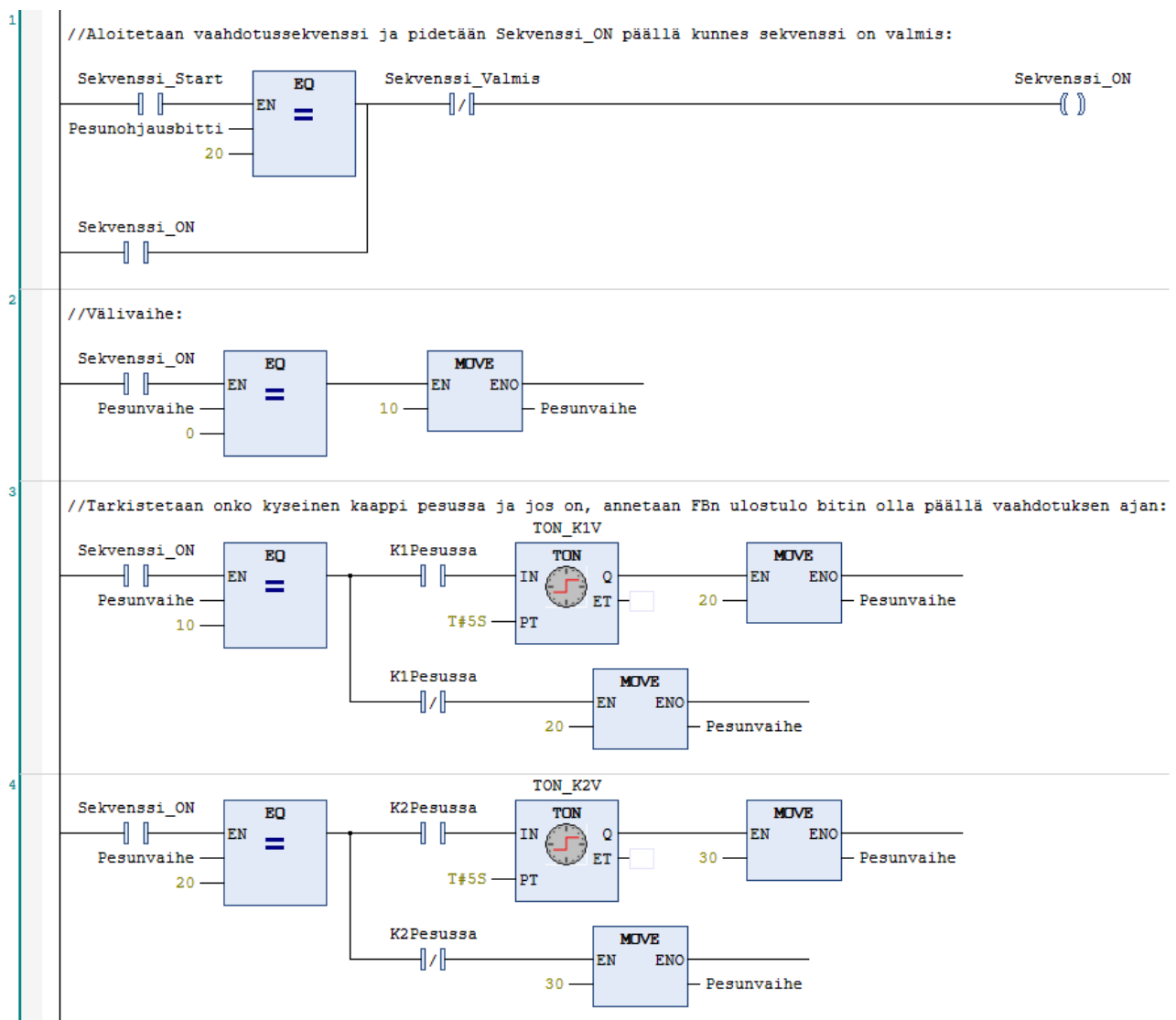
Kuvio 11. Huuhteluohjelman kierto

Pesun viimeisissä vaiheissa ohjelmapuolelta ei ohjata enää ulostuloja päälle, vaan pelkästään odotetaan haluttu aika. Odotusaika säädetään kuviossa 12 nähtävällä "TON_Loppu_Odotus"-ajastimella. Sen jälkeen annetaan toisaalta ohjatun suihkuohjelman olla päällä määrätty aika ja lopuksi merkitään pesu valmiiksi sekä resetoidaan ajastimet ja apumuuttujat.



Kuvio 12. Pesun loppu

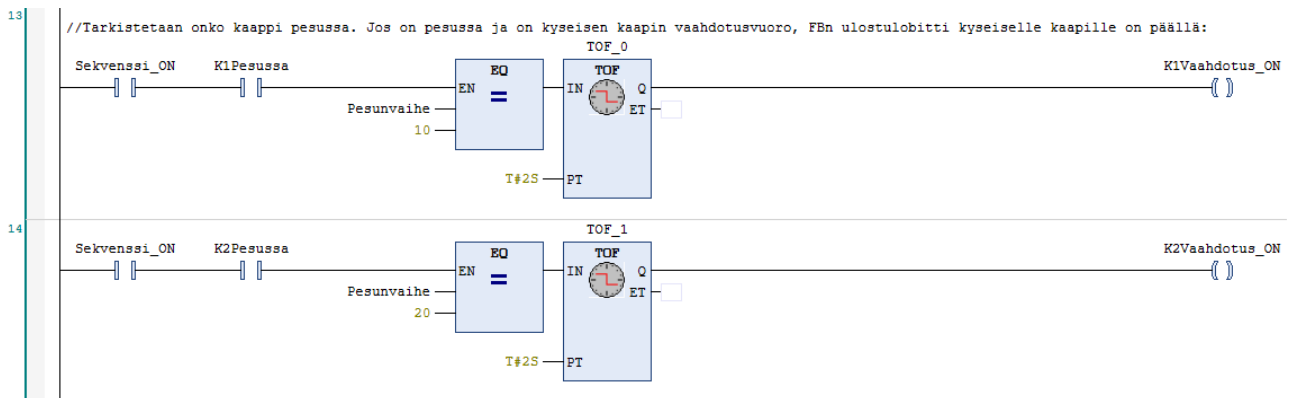
Vaahdotussekvenssistä vastaava Function Block selkeyttää ohjelman lukemista ja helpottaa mahdollisten muutoksien tekemistä. Sekvenssi alkaa pesuohjelman aktivoitessa "Sekvenssi_Start"-muuttujan ja antaessa luvun 20 "Pesunohjausbitti" muuttujalle. Tällöin "Sekvenssi_ON"-muuttuja asetetaan päälle. FB saa myös tiedon siitä, mitkä kaapit ovat pesussa, ja sen perusteella joko pitää tiettyä "Pesunvaihe" tilaa päällä tai siirtyä tarkastamaan numerojärjestyksessä seuraavan kaapin tilaa. Tämä tarkistus tehdään kuvion 13 tavoin jokaiselle kaapille.



Kuvio 13. Vaahdotussekvenssi

Vaahdotussekvenssi kertoo pääohjelmalle, kuinka kauan mitäkin kaappia vaahdotetaan. Se onnistuu pitämällä vaahdotuksessa olevan kaapin ulostulomuuttujan aktiivisena vaahdotuksen ajan kuviossa 14 näytetyllä tavalla. Ulostulomuuttujaa pidetään TOF eli Time

Off Delay ajastimella päällä muutaman sekunnin kauemmin määrättyä aikaa. Näin vältetään rasittamasta pumppua ja putkistoa turhaan. Vaahdotuksen aikana pesukeskuksen pumpun on hyvä antaa käydä ilman kovia painepiikkejä, mikä aiheutuisi, jos kaappien magneettiventtiilit avattaisiin ja suljettaisiin vuoron perään ilman pientä viivettä.



Kuvio 14. Vaahdotussekvenssi

5 Yhteenveto

Huolimatta siitä, että Siemensin toimitusaikojen viivästymisen vuoksi työtä ei saatu toteutettua loppuun asti, oli tämä työ mielenkiintoinen toteuttaa tähän pisteeseen asti. Osastolla tehdyt alkukartoitukset ja vanhojen sähkökuvien sekä ohjelmakierron tutkiminen olivat hyvin opettavaisia toimenpiteitä.

Jos työ olisi edennyt käyttöönottoon asti, asennuksen aikana olisi varmasti tullut vielä paljon uusia ratkottavia ongelmakohtia, mitä tässä vaiheessa ei osattu ottaa huomioon. Teoriapuolella asiat vaikuttavat monesti yksinkertaisilta ja vasta työtä toteutettaessa huomataan, etteivät asiat toimikkaan suunnitellusti. Ohjelmaa luotaessa pyrittiin kuitenkin pitämään mielessä käytännön asioita, kuten pumppuun kohdistuvia rasituksia pesun eri vaiheissa. Myös ohjelmassa käytetyt muuttujat on aseteltu ja nimetty siten, että tulevaisuudessa tehtävä HMI-ohjauspaneeli on helppo ohjelmoida.

Ohjelmointiosuutta tehtäessä tuli kerrattua opintojen aikana opittuja asioita logiikkaohjelmoinnista sekä opeteltua paljon uusia tekniikoita. Vaahdotusta ja huuhtelun kulkua ohjaavan sekvenssin rakentaminen oli työssä sekä haastavin että mielenkiintoisin osuus. Virheistä oppimalla, opittua tietoa hyödyntämällä ja uusien menetelmien opettelulla ohjelma saatiin toimimaan halutulla tavalla. Se toimii hyvänä pohjana tulevaisuudessa modernisoitavalle pesukeskukselle.

LÄHTEET

- Atria Oyj. 2022. Atrian brändit. [Verkkajulkaisu]. [viitattu 31.3.2022] Saatavana: <https://www.atria.fi/konserni/yritys/kansainvalisyys/>
- Atria Oyj. 2022a. Atria yritys. [Verkkajulkaisu]. [viitattu 31.3.2022] Saatavana: <https://www.atria.fi/konserni/yritys/>
- Atria Oyj. 2022b. Atria Ruotsi. [Verkkajulkaisu]. [viitattu 31.3.2022] Saatavana: <https://www.atria.fi/konserni/yritys/kansainvalisyys/atria-ruotsi/>
- Atria Oyj. 2022b. Atria Suomi. [Verkkajulkaisu]. [viitattu 31.3.2022] Saatavana: <https://www.atria.fi/konserni/yritys/kansainvalisyys/atria-suomi/>
- Atria Oyj. 2022b. Atria Tanska ja Viro. [Verkkajulkaisu]. [viitattu 31.3.2022] Saatavana: <https://www.atria.fi/konserni/yritys/kansainvalisyys/atria-tanska-viro/>
- Atria Oyj. 2022c. Atria taloustieto. [Verkkajulkaisu]. [viitattu 31.3.2022] Saatavana: <https://www.atria.fi/konserni/sijoittajat/taloustieto/>
- Atria Oyj. 2022d. Atrian strategiset painopisteet. [Verkkosivu]. [viitattu 31.3.2022] Saatavana: <https://www.atria.fi/konserni/yritys/strategia/strategiset-painopisteet/>
- Berger, H. 2003. Automating with SIMATIC: Integrated automation with SIMATIC S7-300/400: Controllers, software, programming, data communication, operator control and process monitoring. [viitattu 1.3.2022]. Saatavana: https://automatasjwborjasunexpo.files.wordpress.com/2014/01/hans_berger_-_automating_with_simatic.pdf
- Hughes, Thomas A. 2015. Measurement and Control Basics (5th Edition). [Verkkokirja]. International Society of Automation (ISA). [viitattu 15.3.2022]. Saatavana: Knovel-kirjakokoelmasta. Vaatii käyttöoikeuden.
- Kauppalehti. 2022. Atria-Lihavalmiste Oy. [Verkkajulkaisu]. Helsinki: Alma Media Oyj. [viitattu 31.3.2022]. Saatavana: <https://www.kauppalehti.fi/yritykset/yritys/18871226>
- Keinänen, T., Kärkkäinen, P., Lähetkangas, M. & Sumujärvi, M. 2007. Automaatiojärjestelmien logiikat ja ohjaustekniikat. [Verkkokirja]. Helsinki: Sanoma Pro. [viitattu 6.6.2021]. Saatavana: Ellibs-e-kirjakokoelmasta. Vaatii käyttöoikeuden.
- Mitsubishi. 6.1995. SPECIFICATIONS OF THE FX-32MR-ES. [Verkkodokumentti]. Mitsubishi. [viitattu 3.3.2022]. Saatavana: https://scemosystems.fi/downloads/pdf/mitsubishi_melsec_fx-32mr-es_programmable_logic_controller_technical_specification_datasheet.pdf

Mondi Anderson. 3.9.2018. What is the difference between PLC and DCS? | RealPars [Verkkosivu]. [viitattu 10.4.2022]. Saatavana: <https://realpars.com/difference-between-plc-and-dcs/>

Robinson, Richard K. 2000. Encyclopedia of Food Microbiology, Volumes 1-3. [Verkkajulkaisu]. Elsevier. [viitattu 23.3.2022]. Saatavana: Knovel-kirjakokoelmasta. Vaatii käyttöoikeuden.

Siemens. Ei päiväystä. LOGO! Basic Modules. [Verkkajulkaisu]. [viitattu 8.3.2022]. Saatavana: <https://new.siemens.com/global/en/products/automation/systems/industrial/plc/logo/logo-basic-modules.html#BasicModuleswithdisplay>

Siemens. Ei päiväystä. SIMATIC ET 200SP – the compact IO system for the control cabinet [Verkkosivu]. Siemens. [viitattu 3.4.2022]. Saatavana: <https://new.siemens.com/global/en/products/automation/systems/industrial/io-systems/et-200sp.html>

Siemens. Ei päiväystä. SIMATIC ET 200SP [Verkkajulkaisu]. [viitattu 10.4.2022]. Saatavana: <https://mall.industry.siemens.com/mall/en/WW/Catalog/Products/10144488?tree=CatalogTree>

Siemens. Ei päiväystä. TP700 Comfort. [Verkkosivu]. Siemens. [viitattu 10.4.2022]. Saatavana: https://www.automation.siemens.com/bilddb/search.aspx?objkey=P_ST80_XX_02173