



Osaamista
ja oivallusta
tulevaisuuden
tekemiseen

Aatu Tauriainen

Salaatinsiementen pastörointi

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Sähkö- ja automaatiotekniikka

Insinöörityö

28.4.2021

Tekijä Otsikko	Aatu Tauriainen Salaatinsiementen pastörinti
Sivumäärä Aika	29 sivua + 5 liitettä 28.4.2021
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Sähkö- ja automaatiotekniikka
Ammatillinen pääaine	Automaatiotekniikka
Ohjaajat	Lehtori Kristian Junno
<p>Insinööriyössä oli tavoitteena yhden tuotantovaiheen automatisointi paikalliselle Snafu Oy -yritykselle. Tavoitteena oli automatisoida salaatinsiementen pastörinti, esipesu ja jäähdytys. Automatisoinnin avulla pyritään tasalaatuiseen lopputulokseen ja keräämään tietoa tuotantoeristä laadunvalvontaa varten.</p> <p>Pastörintin suunnittelua varten käytiin vierilemassa asiakkaan tuotantotiloissa ja tutustussa pastörintiprosessiin sekä itusalaatin viljelyyn. Vierailun pohjalta tehtiin arvio tarvittavasta laitteistosta ja prosessikaavio automatisoidusta toteutuksesta. Näiden pohjalta lähdettiin tilaamaan tarvikkeita, joiden avulla logiikkaohjelma pystyttiin luomaan ja testaamaan sen toiminnallisuutta testausympäristössä.</p> <p>Esipesun ja jäähdytyksen yhdistämistä osaksi pastörintiprosessia suunniteltiin mekaanisen toteutuksen näkökulmasta. Luotiin suunnitelmia eri toteutusvaihtoehdoista, joiden avulla mietittiin parasta ratkaisua.</p> <p>Lopputuloksena on valmis logiikkaohjelma kytkentäkaavioineen ja käyttöohjeineen sekä suunnitelma asennuksesta. Lisäksi logiikkaohjelma kerää talteen työkiertoista tietoa, jonka avulla jatkokehitys on mahdollista.</p>	
Avainsanat	Automaatio, Automatisointi, elintarviketeollisuus

Author Title	Aatu Tauriainen Pasteurization of lettuce seeds
Number of Pages Date	29 pages + 5 appendices 28 April 2021
Degree	Bachelor of Engineering
Degree Programme	Electrical and Automation Engineering
Professional Major	Automation
Instructors	Kristian Junno, Lecturer
<p>The goal of the thesis was to automate one production phase for the local Snafu Oy company. The goal was to automate the pasteurization, washing and cooling of salad seeds. The main objective of automation is to achieve a uniform result and to collect information on production batches for quality control purposes.</p> <p>In order to plan pasteurization, a visit was made to the customer's production facilities to get acquainted with the pasteurization process, as well as the cultivation of sprout salad. Based on the visit, an assessment of the required hardware and a process diagram of the automated implementation were made. Based on these, the order of supplies that is required for the creation of the logic program and the test environment were made.</p> <p>The integration of prewash and cooling into the pasteurization process was designed from the perspective of mechanical implementation. Plans were created for different implementation options, which were used to determine the best solution.</p> <p>The result is a complete logic program, with wiring diagrams and operating instructions, as well as a plan for installation. In addition, the logic program collects information from the cycles, which makes further development possible with the help of the collected information.</p>	
Keywords	Automation, food-industry

Sisällys

Lyhenteet

1	Johdanto	1
2	Projektin taustat	2
2.1	Silmusalaatti	2
2.2	Digi-Salama-hanke	3
2.3	Pastörintiprosessi	3
2.3.1	Punnitus	4
2.3.2	Esipesu	4
2.3.3	Pastörinti	5
2.3.4	Jäähdytys ja jatkokäsittely	6
3	Projektisuunnitelma	7
3.1	Tavoite	7
3.2	Prosessin suunnittelu	8
3.3	Mekaaninen suunnittelu toteutukselle	10
3.3.1	Hammashihna	11
3.3.2	Sylinteriavusteinen lineaariliike	12
3.4	Aikataulus	13
4	Projektin vaiheet	14
4.1	Vaatimusmäärittely	14
4.2	Tarjouspyyntöjen lähetys ja tilausten tekeminen	15
4.3	Lämpötilan mittaus ja punnitus	16
4.4	Testiympäristön rakentaminen	16
5	Logiikkaohjelman toiminnallisuus	19
5.1	Näytön toiminnallisuudet ja painikkeiden toiminnot	20
5.2	Turvalogiikka	22
5.3	Dataloki	22

5.4	Mittaus	23
5.5	Automaattitila	24
5.6	Manuaalitila	25
6	Yhteenveto	25
6.1	Aikataulussa pysyminen	25
6.2	Lopputulos	26
6.3	Etä- ja itsenäisen työskentelyn haasteet	26
6.4	Jatkokehitysmahdollisuudet	27
	Raporttien luominen	27
	Punnitus	27
	Käyttöliittymän parantaminen	27
	Keittoveden lämpötilan säätö logiikkaohjaimen avulla	28
	Lähteet	29
	Laiteluettelo	1
	Global variables	1
	Local variables	1
	Logiikkaohjelman lähdekoodi	1
	Laitteiston käyttöohjeet ja muut asiakirjat	1
	Liitteet	
	Liite 1. Laiteluettelo	
	Liite 2. Global variables	
	Liite 3. Local variables	
	Liite 4. Logiikkaohjelman lähdekoodi	
	Liite 5. Laitteiston käyttöohjeet ja asiakirjat	

Lyhenteet

CSV	Comma-separated values, pilkuilla erotetut arvot
AI	Analog input, analoginen tulo
AO	Analog outoput, analoginen lähtö
DLG	Data Log Function, datalokitoiminto
PDF	Portable document format, siirrettävä tiedostomuoto
PC	Personal computer, henkilökohtainen tietokone
DI	Digital input, digitaalinen tulo
DO	Digital output, digitaalinen lähtö
LED	Light-emitting diode, hohtodiodi
LCD	Liquid-Crystal Display, nestekidenäyttö
RTD	Resistance thermometer, vastuslämpötila-anturi
PT-100	Vastuslämpötila-anturi
PLC	Programmable logic controller, ohjelmoitava logiikka
SD	Secure digital, yleisesti käytetty muistikorttityyppi

1 Johdanto

Opinnäytetyössä suunniteltiin Digi-Salama-hankkeen projektina yhden tuotantovaiheen automatisointia paikalliselle Snafu Oy:lle, joka viljelee luonnonmukaista versosalaattia. Digi-Salama-hanke työllistää opiskelijoita asiakasprojekteihin. Projekteissa lähdetään kehittämään jotain asiakkaan liiketoiminnan osa-aluetta. Tämän asiakkaan tapauksessa suunniteltiin yhtäaikaaisesti myös yhteistyörobotin hyödyntämistä rasioiden pakkaamisessa kuljetuslaatikoihin.

Tämän työn tarkoituksena on dokumentoida projektin eri vaiheet ja toimia perustana jatkosuunnitelmille automatisointiin. Tuotantovaihe, jota automatisointi koskee, pitää sisälleen salaatin siementen pesemisen, pastöroinnin ja huuhtelemisen. Peseminen tapahtuu ennen pastörointia ja huuhteleminen pastöroinnin jälkeen. Aluksi tutustuttiin asiakkaan tiloihin ja seuraamassa pastörintiprosessia tuotannon aikana. Vierailun pohjalta ryhdyttiin suunnittelemaan asiakkaan kanssa prosessin automatisointia ja keskustelemaan tarkemmin projektin tavoitteista.

Ensiksi tilattiin ohjelmitava logiikka logiikkaohjelman toteuttamista varten. Ohjelma sisältää tarvittavat toiminnallisuudet pastöroinnin toteuttamiseen koneellisesti. Ohjelman testausta varten alettiin rakentaa testausympäristöä Metropolia Ammattikorkeakoulun automaatiolaboratorioon. Testausympäristön avulla pystyttiin suunnittelemaan myös, miten sama laitteisto saadaan asennettua asiakkaan tuotantotiloihin ja millaisia asennusteknisiä ratkaisuja tullaan tarvitsemaan.

Suurin hyöty, jota automatisoinnilla lähdettiin tavoittelemaan, on varmuus siitä, että jokainen erä siemeniä on käsitelty samalla tavalla. Työkierroista kerätään tietoa, jota voidaan hyödyntää prosessin kehittämisessä ja laadunvalvonnassa. Vertaamalla pastöroinnin lopputulosta käytössä olleisiin prosessiarvoihin datalokista pystytään tekemään tarvittavia säätöjä keittoaikoihin tai lämpötiloihin halutun lopputuloksen saavuttamiseksi.

2 Projektin taustat

2.1 Silmusalaatti

Snafu Oy on vuonna perustettu vuonna 2012, ja yrityksen nykyinen toimipiste on Keran halleilla Espoossa. Silmusalaatti on Snafu Oy:n tuotemerkki [1] ja tarkoittaa yrityksen luonnonmukaisesti tuotettua versosalaattia. Versot ovat alkuvaiheessa olevia kasveja, jotka on kasvatettu siemenestä idättämällä niitä vedessä [2]. Tuotteen maku määräytyy sen mukaan mitä eri versoja se sisältää, ja niitä voi helposti syödä sellaisenaan, aterian osana tai esimerkiksi salaattipohjana. Silmusalaattia löytyy hyvin pääkaupunkiseudun S-ryhmän ja K-kauppojen hyllyiltä (kuva 1) [3].



Kuva 1. Silmusalaattia Martinlaakson S-marketin hedelmä- ja vihannesosastolla.

Tällä hetkellä Silmusalaatin valikoima koostuu neljästä tuotteesta:

- Alfalfa: sisältää puna-apilaa ja sinimailasen versoja, sopii hyvin esimerkiksi avokadon kanssa tarjoiltuna ja on lapsille helpoiten lähestyttävä tuote.
- Parsakaali: sisältää puna-apilan, sinimailasen ja parsakaalin versoja, sopii tarjoiltavaksi hyvin esimerkiksi kalan kanssa.

- Tulinen: sisältää retikan, puna-apilan ja sinimailasen versoja, sopii esimerkiksi itämaisiiin ruokiin ja tortilloihin.
- Rouskuva: sisältää herneen, vihreän- ja punaisen linssin ituja. Hyvä kasvisproteiinin lähde, sellaisenaan syötäväksi aterian osana esimerkiksi keittoon ripoteltaessa. [4]

2.2 Digi-Salama-hanke

Digi-Salama-hankkeen ideana on auttaa Uudenmaan pieniä ja keskikokoisia yrityksiä hyödyntämään uusia teknologioita kuten esimerkiksi tekoälyä, yhteistyörobotiikkaa ja virtuaalisia kaksosia. Opiskelijat työskentelevät kehityshankkeissa, joissa ideoidaan, miten yrityksen toimintaan voidaan implementoida edellä mainittuja teknologioita ja täten säävuttaa kilpailuetua sekä paikallisilla että maailmanlaajuisilla markkinoilla ja luoda uutta osaamista ja kykyä soveltaa niitä sekä toteuttaa valmiita toimintakonsepteja.

Uusien teknologioiden adaptoimiseen omaan toimintaan niiden aikaisessa vaiheessa liittyy aina riskejä yritykselle, ja sitä kynnystä pyritään madaltamaan tarjoamalla opiskelijoiden apua harppauksen ottamisessa. Hanke kestää kaksi vuotta (1.9.2019–31.8.2021), ja se tulee työllistämään arviolta yhteensä noin 200 muutosagenttia, asiantuntijaa ja opiskelijaa eri projekteihin eri aloille. Hankkeen toteuttajina toimivat Metropolia ammattikorkeakoulu ja Vantaan kaupunki yhdessä Suomen Automaatioseura Ry:n, Suomen Robotiikkayhdistys Ry:n, Tampereen yliopiston TRINITY-hankkeen ja VTT:n DIH2-hankkeen kanssa. [5]

2.3 Pastörintiprosessi

Kävimme projektiryhmän kanssa vierailmassa asiakkaan tuotantotiloissa ja tutustumassa itusalaatin viljelyyn ja salaatinsementen pastörintiin. Vierailu auttoi ymmärtämään pastörintin merkityksen hyvän ruokatuotteen, omavalvonnan ja turvallisuusvaatimusten kannalta.

2.3.1 Punnitus

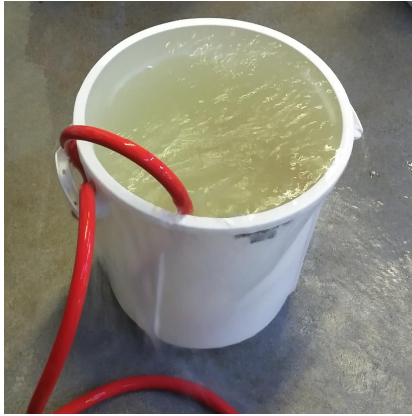
Prosessi alkaa avaamalla pakkaukset, joissa siemenet ovat, ja punnitsemalla ne kymmenen kilon kokoisiksi eriksi. Punnitsemalla varmistetaan, ettei käsiteltävän erän paino nouse liian suureksi. Liiallisen painon johdosta pastöroinnissa käytettävän keittoveden lämpötilaa laskee eivätkä kaikki siemenet saa tarvittavaa lämpökäsittelyä. Punnitut erät laitetaan ämpäreihin, jotka työnnetään kuvassa 2 näkyvällä kärryllä eteenpäin tuotantotilaan, jossa niiden jatkokäsittely suoritetaan.



Kuva 2. Siemen erät matkalla punnituksesta pesulle.

2.3.2 Esipesu

Ämpäreissä olevat siemen erät siirretään haavin muotoisiin siivilöihin, minkä jälkeen siivilät siirretään kuvassa 3 näkyvään vesiastiaan yksi kerrallaan ja niitä liotetaan kahden minuutin ajan, jotta kaikki lika ja pöly irtoavat niistä. Liotuksen jälkeen siivilä huuhdellaan juoksevalla vedellä ja annetaan veden valua kokonaan pois siemenistä ennen kuin ne siirretään keittoastiaan, jottei kiehuvan veden sekaan valuisi kylmää pesuvettä ja lämpötila laskisi.



Kuva 3. Pesuastia.

2.3.3 Pastörinti

Pesun jälkeen siemenet siirretään yksi siivilä kerrallaan keittopataan pastörintia varten, kuten on esitetty kuvassa 4. Keittopadan veden lämpötila on säädetty kyseistä siemenlajiketta vastaavaksi:

- herneet: +83–86 °C
- linssit: +80–83 °C
- alfalfa, apila ja retikka: +78–81 °C
- parsakaali: +74–76 °C.



Kuva 4. Operaattori suorittamassa pastörintia.

Operaattori säätää keittopadan lämpötilan haluttuun arvoon painamalla lämpötilan lisäys- tai vähennyspainiketta padan käyttöpaneelista. Tavoitelämpötila päivittyy padan lämpötilänäytölle. Lisäksi kuvassa 5 näkyvästä käyttöpaneelista löytyy käyttökytkin, hätä-seis-painike ja painikkeet padan kallistamiseen ja palauttamiseen pystyasentoon. [6]



Kuva 5. Keittopadan käyttöpaneeli.

Pastörinti kestää 20 sekuntia, ja sen tarkoituksena on keittää siemeniä siten että mahdollinen salmonella- ja kolibakteeri tuhoutuvat sen pinnasta [7] mutta siementen itävyys ei kärsi. Johtuen siementen rakenteiden eroavaisuuksista käytettävissä lämpötiloissa on paljon eroja, ja siksi on äärimmäisen tärkeää, että jokainen siemen on pastöroitu ohjeita noudattaen.

2.3.4 Jäähdytys ja jatkokäsittely

Pastöroinnin jälkeen siemenet siirretään jäähdytysastiaan, joka vastaa esipesussa käytettyä astiaa. Siemenet jäähdytetään kylmässä vedessä pikaisesti, jotta vältetään liiallisen lämpenemisen aiheuttamat vahingot. Kun siemenet ovat jäähtyneet tarpeeksi, ne siirretään kuvassa 6 näytettyyn rumpuun, jossa niiden idättäminen alkaa.



Kuva 6. Idätysrumpu.

3 Projektisuunnitelma

3.1 Tavoite

Projektin tavoitteena oli suunnitella, miten pastöroinnin, esipesun ja jäähdytyksen voisi automatisoida mahdollisimman yksinkertaisesti ja helposti toteutettavalla konseptilla. Tätä varten lähdettiin ensiksi luomaan logiikkaohjelmaa, joka hyödyntää asiakkaan valmiiksi omistamaa kalustoa ja suorittaa pastöroinnin eri siementyypeille sekä kerää tietoa työkiertoista, jolla varmistetaan ohjelman toimivan halutulla tavalla.

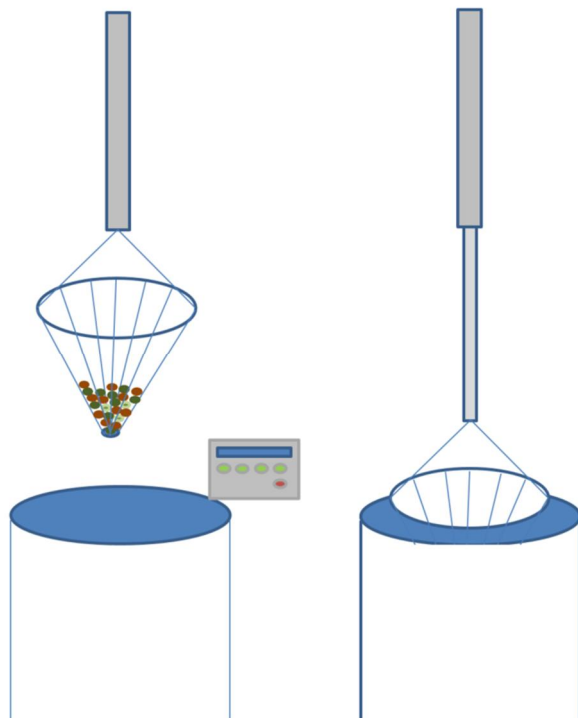
Riittävällä anturoinnilla ja ohjelman rakenteellisilla ominaisuuksilla kuten eri ohjelmilla eri siemenille pystytään takaamaan tasalaatuinen lopputulos laadunvalvonnan kannalta. Lisäksi saadaan varmuus siitä, että jokainen tuotantoerä on käsitelty samalla tavalla. Operaattorin suorittamassa manuaalisessa pastöroinnissa keittoveden lämpötila ja keittoaika vaihtelevat eikä ole kannattavaa tai luetettavaa dokumentoida jokaista erää, miten ne on käsitelty.

Keittopadan läheisyyteen asennetaan näyttöpäätte, josta voi lukea keittopadassa olevan veden lämpötilan ja valita painikkeella yhden neljästä suoritettavasta ohjelmasta tai liikutella sylintereitä manuaalisesti esimerkiksi huoltotilanteen sitä vaatiessa. Mikäli veden

lämpötila on riittävä valitulle ohjelmalle, pastörointi alkaa ja sylinteri laskee siivilän keittopataan 20 sekunniksi. Pastöroinnin aikana toinen sylinteri nykii siivilää edestakaisin, varmistaen, että vesi pääsee kosketuksiin kaikkien siementen kanssa. Kun keittoaika tulee päätökseen ja ohjelman suoritus lakkaa, nostetaan sylinteri takaisin aloituspisteeseen ja operaattori voi poistaa siivilän ja siemenet tarttujasta ja siirtää ne jäähdtykseen. Kun uusi erä siemeniä on vaihdettu tarttujaan, voi ohjelman aloittaa uudestaan ja voidaan siirtyä seuraavaan tuotantoerään.

3.2 Prosessin suunnittelu

Prosessin suunnittelun perustana toimi entuudestaan yrityksen omistama kalusto, johon kuului kaksi paineilmasyylinteriä (CP96SDB32-700C ja CD85N16-300-B-x2018), keittopata (Metos Culino) ja kaksi magneettiventtiiliä. Toinen sylintereistä, joka näkyy kuvassa 7, on varustettu 700 mm:n iskupituudella. Se roikkuu keittopadan yläpuolella ja kannattelee tarttujaa, johon siemenet sisältävä siivilä on asetettu. Siivilä on samanlainen kuin aiemmin käytetty, ja operaattori nostaa sen paikoilleen tarttujaan.

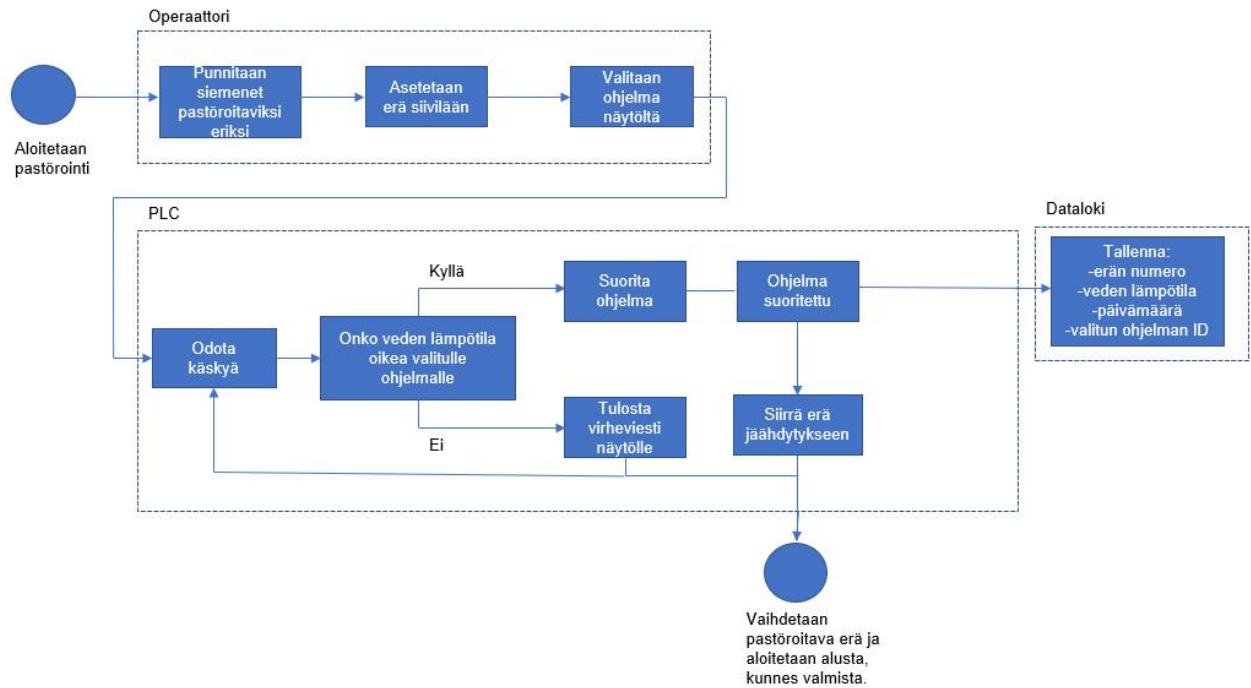


Kuva 7. Sylinterin laskeminen keittopataan.

Prosessin toimivuuden kannalta jokainen erä täytyy punnita joko logiikan tai operaattorin toimesta ja veden sekä siementen lämpötiloja mitata työkierron aikana. Kun siemenet asetetaan sylinterissä olevaan tarttujaan, vetoanturi punnitsee siemenet ja ilmoittaa operaattorille, jos niitä on liian vähän tai paljon, ja mikäli painon ylitys tai alitus tavoitteesta on sallituissa rajoissa, voi työkierto alkaa. Seuraava vaihe punnituksen jälkeen on keittoveden lämpötilan tarkastus, niin että se vastaa operaattorin valitsemaa siementyyppiä. Mikäli operaattori on valinnut pastöroivansa esimerkiksi herneitä, tulee erän olla sopivanpainoinen punnituksessa ja keittoveden olla sallitussa lämpötilassa (+83–86 °C herneen tapauksessa).

Mikäli kaikki edellä mainitut ehdot täyttyvät, niin valittu ohjelma alkaa, tai näyttöpäätteelle tulostetaan virheviesti, josta selviää, miksei ohjelmaa voitu aloittaa. Operaattori tekee tarvittavat korjaukset ja yrittää aloittaa ohjelman suorittamisen uudestaan. Pastöroinnin aikana logiikka tallentaa datalokiin aloitushetkellä mitatun veden lämpötilan, painon, valitun ohjelman, operaattorin käyttäjätunnuksen ja erän numeron. Kaikki tiedot on aikaleimattu, mikä helpottaa tiedon löytämistä mahdollisia virheitä jäljittäessä. Kaikki operaattorin ja PLC:n suorittamat toiminnot on kuvattu kuvan 8 prosessikaaviossa.

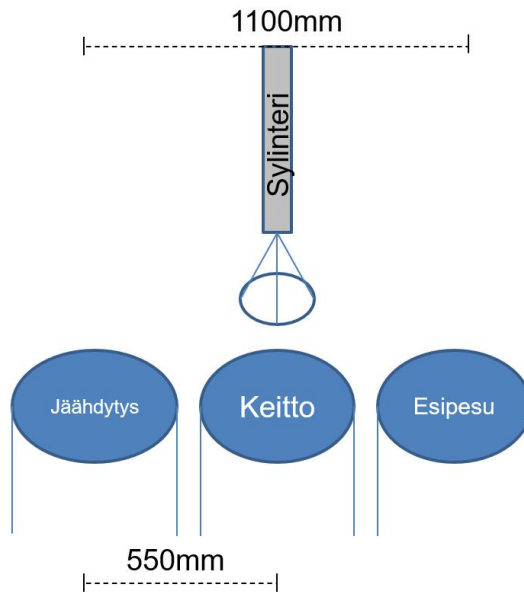
Pastöroinnin jälkeen siemenet siirretään jäähdytykseen, jonka aikana niiden lämpötilaa mitataan myös, jotta varmistetaan niiden jäähtyvän tarpeeksi ja vältetään mahdolliset liiallisen lämpenemisen aiheuttamat vauriot.



Kuva 8. Prosessikaavio.

3.3 Mekaaninen suunnittelu toteutukselle

Suurimmaksi haasteeksi jäähdytyksen ja esipesun automatisoinnin ja lisäämisen osaksi samaa työkiertoa pastöroinnin kanssa osoittautui kuvassa eri työvaiheiden välille tarvittavan lineaariliikkeen toteuttaminen, jota on havainnollistettu kuvassa 9. Esipesua tai jäähdytystä ei voi toteuttaa liian lähellä keittopataa, ettei kylmä vesi ajaudu sinne ja täten laske keittoveden lämpötilaa. Toiseksi lineaariliikkeen toteuttavat laitteet on haastava asentaa tuotantotilaan, jos pysyy alkuperäisessä tavoitteessa ja välttää asentamasta mitään lattialle. Esittelen seuraavassa konseptikuvia vaihtoehtoisista asennusmenetelmistä, joita suunniteltiin toteutusta varten. Kuvat ovat alustavia, ja niiden tarkoitus on vain hahmottaa konseptin ideaa.

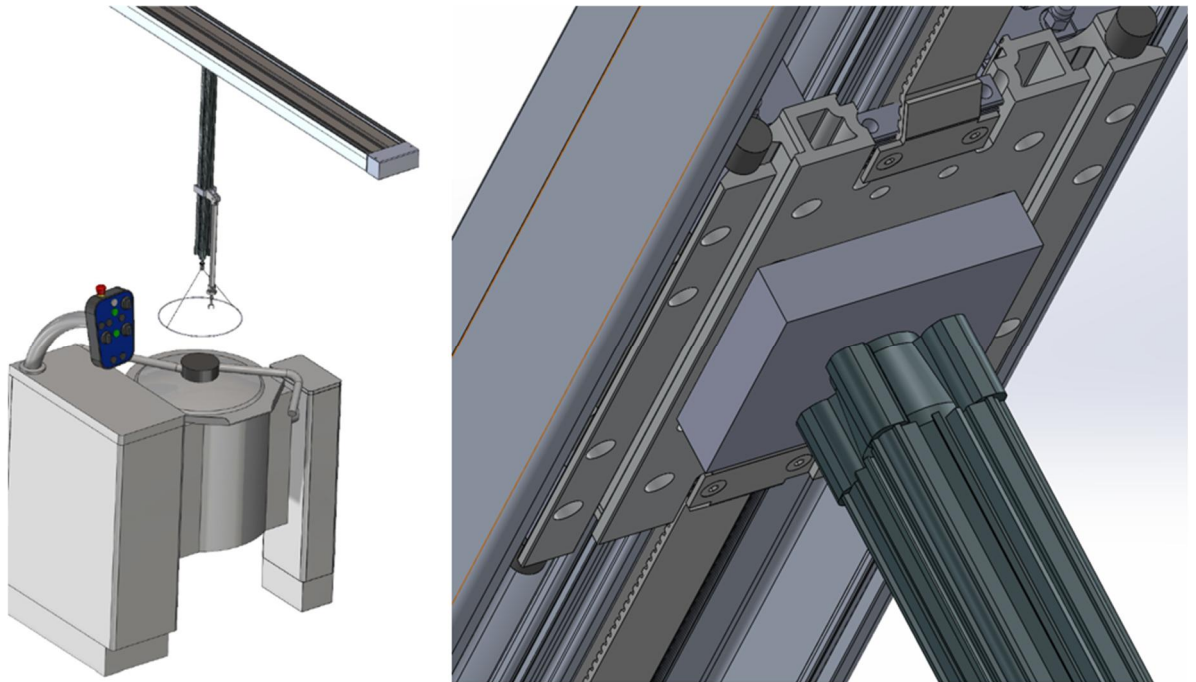


Kuva 9. Prosessin vaiheet ja liikkuminen niiden välillä.

3.3.1 Hammashihna

Sylinteri on kiinnitetty alumiiniprofiilista valmistettuun lineaariyksikköön, joka näkyy kuvassa 10. Lineaariyksikön voimanlähteenä toimii sähkömoottorin pyörittämä 25 mm:n levyinen hammashihna. Hammashihna siirtää siihen kiinnitettyä kelkkaa ja täten liikuttaa sylinteriä sen välityksellä. Lineaariliikkeen toteuttaminen tällä menetelmällä on yksinkertaista, mutta tilan korkeudesta johtuen se vaatii paljon tukirakenteita kattoon asennettuna.

Tällä menetelmällä voidaan hyödyntää lattialle sijoitettavia ämpäreitä tai muita astioita jäähdytyksen ja esipesun toteuttamisessa, sillä sylinteri liikkuu vaakatasossa ja sen etäisyys lattiaan nähden pysyy samana.



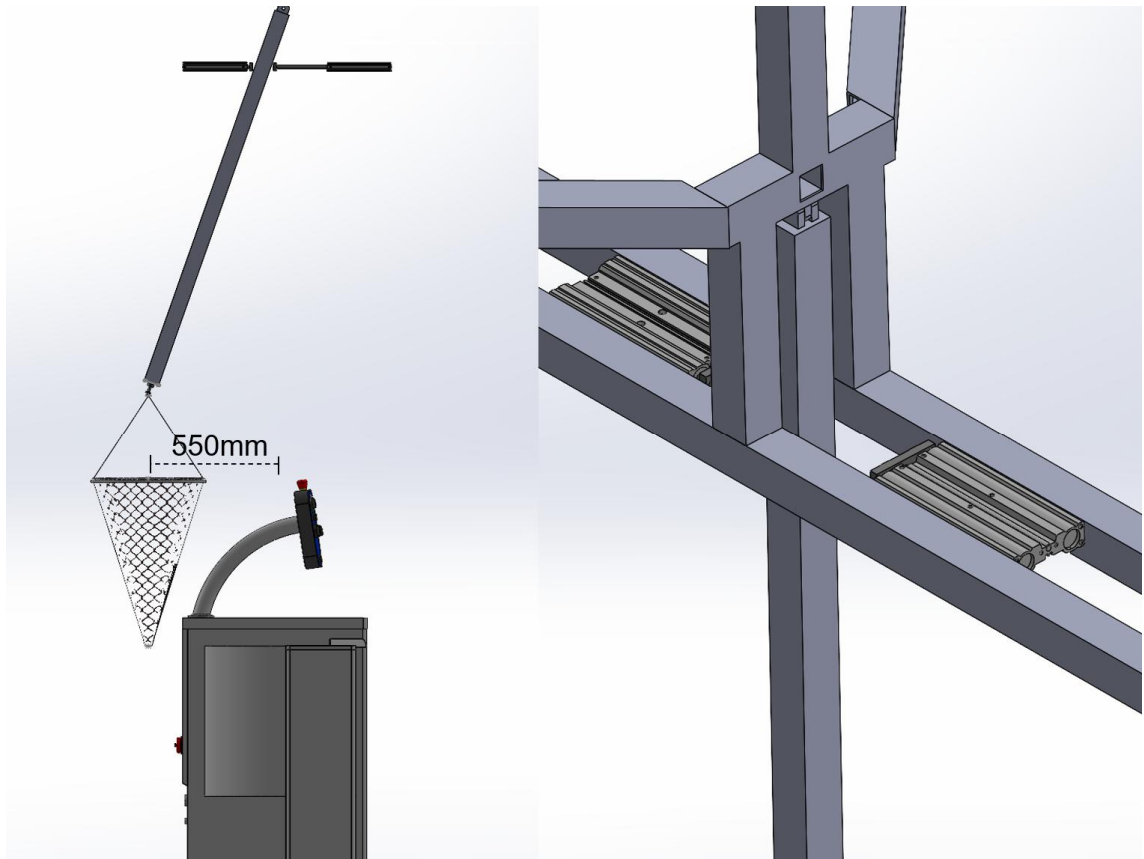
Kuva 10. Sylinteri kiinnitettynä hammashihna lineaarijohteeseen keittopadan yläpuolelle.

3.3.2 Sylinteriavusteinen lineaariliike

Kaksi pienempää sylinteriä työntää edestakaisin 1,5 metriä pitkää teräsputkea, jonka sisälle 700 mm:n iskupituudella varustettu sylinteri on sijoitettu. Teräsputki on kiinnitetty seinästä ja katosta tuettuun tukipisteeseen, jossa on nivel ja akseli, jotka mahdollistavat edestakaisin liikkeen. Kuvassa 11 kuvatun liikkeen avulla siivilään sijoitetut siemenet siirtyvät pois keittopadan päältä, ja ne on mahdollista pestä ja jäähdyttää laskemalla vettä niiden päälle yläpuolelle sijoitetusta suihkusta, siten ettei vesi valu keittopataan ja laske sen lämpötilaa.

Tällä menetelmällä voidaan mahdollisesti säästää kustannuksissa hammashihnaan verrattuna. Toiminnan varmistamiseksi täytyy tehdä lisäselvityksiä siitä, mikä iskupituus, asennuskorkeus ja voima vaaditaan edestakaisin liikkeen toteuttavilta sylintereiltä.

Edestakaisin-liikkeen aikana putken sisällä olevan sylinterin tulee olla sisäänvedettynä, jottei sen työntövarsi vaurioidu. Tästä johtuen jäähdytys sekä esipesu on pakko toteuttaa ylhäältäpäin laskemalla vettä siivilän päälle. Työntövarsta ei voida laskea alas sylinterin ollessa vinossa, mistä syystä alas asennettuun astiaan laskeutuminen ei ole mahdollista.



Kuva 11. Sylinteriavusteinen lineaariliike

3.4 Aikataulutus

Kun oli tutustuttu automatisoitavaan prosessiin ja arvioitu, mitä työvaiheita projektisuunnitelman toteutus tulee vaatimaan, pystyttiin näiden pohjalta luomaan kahden kuukauden suunnitelma. Taulukossa 1 on kuvattu kahden kuukauden aikataulu viikkokohtaisilla tavoitteilla.

Taulukko 1. Alkuperäinen aikataulu, tavoitteet kyseisen viikon loppuun mennessä

Viikko	Tavoite
40	Mekaniikan suunnittelu ja ideointi
41	Tarjouspyyntöjen tekeminen suunnitelmien pohjalta

42	Logiikkaan liittyvät tarjouspyynnöt
43	Tilausten tekeminen ja demon rakentaminen
44	Demon rakentaminen ja testaaminen
47	Asennus ja käyttöönotto
48	Hienosäätö paikan päällä, jatkon suunnittelu vko 48 eteenpäin

4 Projektin vaiheet

4.1 Vaatimusmäärittely

Projektille määrättyjen tavoitteiden pohjalta luotiin tiiviissä yhteistyössä asiakkaan kanssa vaatimusmäärittely projektille (taulukko 2). Vaatimusmäärittely on lista mitattavissa olevista määreistä, joiden avulla voidaan todeta lopputuloksen toimivan halutulla tavalla. Nämä vaatimukset on otettava huomioon aina mieltiessä laitehankintoja ja toteutustapoja projektiin liittyen.

Taulukko 2. Vaatimusmäärittely

Numero	Vaatimus	Toimenpide
1.0.0	Jokainen työkierto on oltava samanlainen.	Kerätään tietoa työkiertojen aikana mitatusta keittoveden lämpötilasta ennen ja jälkeen keittoa.
1.1.0	Puhdistettavien pintojen lukumäärä pidetään mahdollisimman pienenä.	Kaikki mikä on mahdollista, asennetaan irti lattiasta, jotta lattian puhtaanapito on mahdollisimman yksinkertaista.
1.2.0	Kaikkien työntekijöiden ymmärrettävissä olevat käyttöohjeet ja käyttöliittymä.	Kaikki dokumentit ja käyttöliittymä sekä englanniksi, että suomeksi.
1.3.0	Jokaisen työkierron on noudatettava sille asetettuja vaatimuksia.	Työkiertojen aikana seurattavat arvot kirjataan logiikan toimesta datalokiin, josta voidaan tarkistaa tarvittaessa, että kyseiset erät on pastoitu oikein.

1.4.0	Siemen ei saa vahingoittua pastöroinnin aikana.	Mitataan lämpötila ennen keittoa, ettei se ole liian korkea ja keitetään vain 20 sekuntia.
1.5.0	Ergonominen käyttöä.	Huolehditaan siitä, ettei mitään tarvitse nostaa yli vyötärön korkeudelle ja sijoitetaan näyttöpääte ja painikkeet, joista prosessia ohjataan miellyttävälle korkeudelle.
1.6.0	Vesi ei roisku keittopadasta.	Sylintereihin säätöventtiilit, joiden avulla nopeutta voidaan säätää.

4.2 Tarjouspyyntöjen lähetys ja tilausten tekeminen

Vaatimusmääritelmän ja projektisuunnitelman pohjalta lähdettiin tutustumaan eri laitevalmistajien ratkaisuihin ja miettimään, mikä soveltuu parhaiten asiakkaan tarpeisiin ja budjettiin. Ohjelmoitavassa logiikassa tarvittiin riittävien tulojen ja lähtöjen lisäksi datalokiominaisuus tiedon keräämistä varten ja erillinen näyttöpääte, jonka avulla käyttää järjestelmää. Useampien ohjelmointitapojen tuki oli tervetullut lisä ja lisenssivapaa ohjelmisto ohjelmointia varten välttämättömyys; lisäksi vaadittiin mahdollisuus lisäosille lämpötilanmittausta ja punnitsemista varten.

Kustannukset pyrittiin pitämään mahdollisimman alhaisina. Tavoitteena on täyttää kaikki vähimmäisvaatimukset pastöroinnin toteuttamiseksi. Valintoja tehdessä suositaan vaihtoehtoja, jotka tarjoavat myös laajennusmahdollisuuksia tulevaisuutta varten.

Tarjouspyyntöjä lähetettiin Siemensin, Beckhoffin ja Rockwell automationin logiikoiden jälleenmyyjille. Tarjouspyyntöihin lisättiin lyhyt kuvaus projektista, johon tuotteita haettiin. Tarjouksia saatiin Siemens Simatic S7:stä, Beckhoff CX5130-0155:stä sekä Rockwell Automationin Micro-850:sta. Kaikki vaihtoehdot olivat kilpailukykyisiä ja varteenotettavia ominaisuuksiltaan, mutta päädyimme kevyempään ratkaisuun kuin mitä saadut tarjoukset koskivat.

Edellä mainittujen valmistajien ominaisuuksiltaan riisutumpien logiikoiden valikoimasta päädyttiin valitsemaan Rockwell Automationin Micro820. Valinta perustui siihen, että Micro 820 mahdollisti yksinkertaisen LED-LCD-näytön sijoittamisen kytkentäkotelon ulkopuolelle, joten näyttöä sekä sen painikkeita voi käyttää prosessin ohjaamisessa tämän

ansiosta. Lisäksi Rockwell Automationin ohjelmointiympäristö oli ilmainen eikä vaatinut lisenssiä, ja kyseinen logiikka täytti muilta osin myös kaikki edellä mainitut vaatimukset. Logiikkaohjaimen lisäksi tilattiin myös 24 V:n virtalähteen, LED-LCD-näytön ja RTD-lisäosan lämpötilan mittausta varten. [8]

4.3 Lämpötilan mittaus ja punnitus

Lämpötilan mittausta ja punnitusta varten lähetin tarjouspyyntöjä Sarlinille ja Lapp Automationin alaisuudessa toimivalle Epic sensorsille niiden edustamiin tuotteisiin liittyen, minkä lisäksi kävin puhelinkeskusteluja näiden yritysten myyntiedustajien kanssa. Kaikki jälleenmyyjät ehdottivat tarkimman mittaustuloksen omaavaa A-luokan 4-johdinliitännällä toimivaa PT-100-lämpötilanmittausanturia, joka olisi yhteensopiva valitsemani logiikan RTD-muunnoskortin kanssa. Hintaeroja ei valmistajien välillä juuri ollut, joten päätin tilaamaan Sarlinilta, koska koin saavani siltä parasta asiakaspalvelua. [8; 9]

Punnitusta varten soveltuvaksi ratkaisuksi saatiin tarjous 25 kg:n Scaime ZFA -venymäanturista ja CPJ2S-muuntimesta [10;11], mutta niiden yhteishinta koettiin liian suureksi saatuun hyötyyn nähden prosessissa. Tästä syystä projektia päätettiin jatkaa eteenpäin ilman punnitsemista toistaiseksi. Logiikka valittiin kuitenkin niin, että siihen voi halutessaan myöhemmin hankkia tarvittavat laajennusosat, kun sopiva anturi punnitusta varten löytyy.

4.4 Testiympäristön rakentaminen

Tilattujen tarvikkeiden saavuttua rakennettiin Metropolian Myyrmäen-toimipisteen automaatiolaboratorioon testausympäristö logiikkaohjelman ohjelmointia ja testaamista varten. Testiympäristössä hyödynnettiin alumiiniprofiilia, jota oli entuudestaan laboratorion varastossa.

Profiilista rakennettiin kuvassa 12 näkyvä teline, joka nostaa sylinterin riittävään korkeuteen, jotta sitä pystyy liikuttamaan edestakaisin ilmassa. Tämä mahdollistaa ison sylinterin ja pienen sylinterin yhteistyön testaamisen. Pienempi sylinteri nykii siivilässä ole-

vasta narusta, mikä aiheuttaa liikettä siemenissä pastöroinnin aikana ja takaa että keitovesi on kosketuksissa kaikkien siementen kanssa. Tilattu logiikka, virtalähteet ja muut sähkökomponentit sijoitettiin kuvassa 13 näkyvään kytkentäkoteloon.

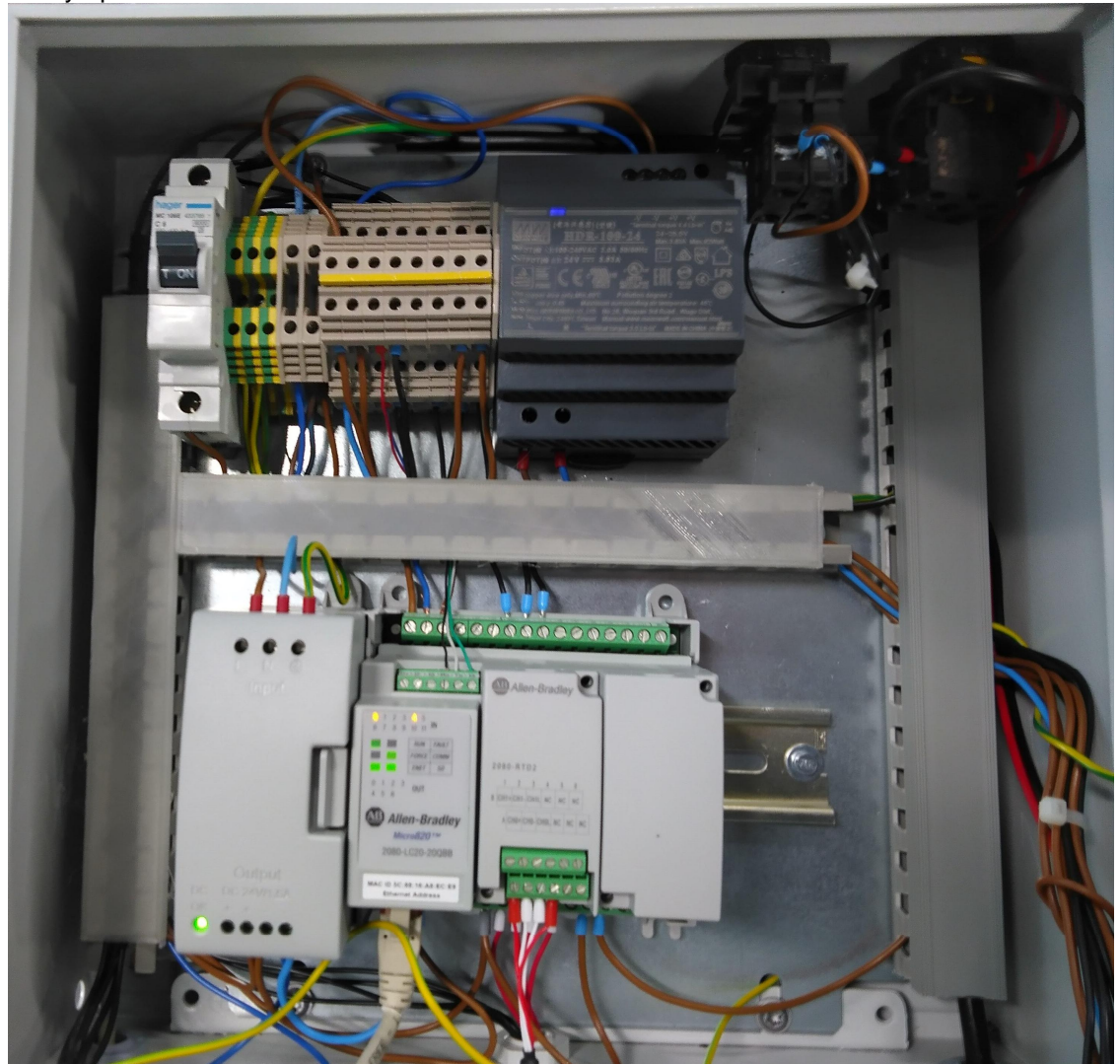
Lisäksi testiympäristössä pystyttiin testaamaan lämpötilan mittausta ja mittaustulosten muuntamista celsiusasteiksi sekä näytön ohjelmointia. Näyttö kommunikoi logiikan kanssa RS-232-sarjaväylän avulla, jonka välityksellä kuva näytöllä päivittyy ja näppäinten painallukset välittyvät logiikalle [13].

Kun halutut toiminnallisuudet saatiin toimimaan, purettiin testiympäristö laboratoriosta ja laitteet luovutettiin takaisin asiakkaalle. Laitteiden asennus asiakkaan tiloihin ja soveluksen testaus tuotannossa alkaa, kun tuotantotiloihin on tehty tarvittavat asennukset, jotka mahdollistavat sylintereiden kiinnitykset.



Kuva 12. Testiympäristö

automaatiolaboratoriossa.



Kuva 13. Kyt Kentäkotelo.

5 Logiikkaohjelman toiminnallisuus

Ohjelma on kehitetty Structured text -menetelmällä käyttäen Rockwell Automationin Connected Components Workbench V12 -sovellusta. Ohjelman toiminnallisuus perustuu ehtorakenteeseen, jonka avulla suoritettava ohjelmahaara valitaan operaattorin komentojen perusteella. Kaksi eri haaraa ovat manuaalijajo "Manual operation" ja automaattitila "Automatic operation".

Kuvassa 15 näkyvälle ruudulle tulostetaan toistuvasti tekstinä tietoa, josta ilmenee, mikä ohjelmahaara on käynnissä ja mitä aliohjelmia automaatiotilassa suoritetaan parhaillaan. Automaatiotilassa on aliohjelmat jokaiselle pastöroitavalle siementyypille, ja niitä voi suorittaa vain yhden kerrallaan. Useita eri aliohjelmiä ei voida suorittaa yhtä aikaa, ja niiden suorittaminen keskeytyy stop-painikkeella tai hätä-seis-painikkeella. Samanaikaisesti ruudulle tulostetaan tietoa riippuen ohjelmasta, esimerkiksi keittoveden lämpötila tai monesko työkierto on käynnissä.



Kuva 14. Käyttöliittymä automaattitilan aikana.

5.1 Näytön toiminnallisuudet ja painikkeiden toiminnot

Ohjelman suorituksen aikana kutsutaan toistuvasti KEY_READ_REM- ja LCD_REM-funktiot. KEY_READ_REM rekisteröi näytön näppäimien painallukset ja LCD_REM päivittää näytön tekstiä. [13]

KEY_READ_REM -funktio sisältää neljä parametria: Enable-tulon ja KEY_READ_REM, Sts ja KeyData -lähdet. Enable-parametrin ollessa tosi funktio tallentaa näyttöpäätteen näppäimien painallukset KeyData-parametrissa määriteltyyn muuttujaan. Käytin KeyData-parametrinä etumerkitöntä kokonaislukua, jonka tavut 0–7 saivat loogisen arvon vastaamaan jokaista näytön painiketta. Esimerkiksi ylänuolipainiketta painettaessa Key_datan ensimmäisen tavun tila on tosi ja vastaavasti alanuolipainiketta painettaessa toisen tavun tila on tosi. Näiden funktioiden kutsut on kuvattu esimerkkikoodissa 1. [13]

```
KEY_READ_REM_1(TRUE);
Key_Data := KEY_READ_REM_1.KeyData;
```

Esimerkkikoodi 1. Kutsutaan funktiot näytön alustusta varten.

Ohjelman suorituksen aikana näyttöpäätteen ylös ja alas-nuolinäppäimiä käytetään manuaali- ja automaattitilan välillä siirtymiseen. Toimintatilojen vaihto nuolinäppäimillä on kuvattu esimerkkikoodissa 2. Funktionäppäimiä 1–4 käytetään eri toimintojen suorittamiseen, kuten ohjelman valitsemiseen automaattitilassa tai sylintereiden liikuttamiseen manuaaltilassa. F5-painike nolaa laskurin arvon, joka laskee suoritettuja työkiertoja. F6-painike toimii yleisenä seis-painikkeena.

```

If(_IO_EM_DI_00 = TRUE) THEN
  if (Key_data.0 = true and Cycle_on = FALSE) THEN
    Screen_var:=0;
    Case_Number:=0;

    ELSIF (Key_data.1 = true and Cycle_on = FALSE) THEN

      Screen_var:=1;
      Case_Number:=5;

  END_IF;

  if (screen_var = 0 and Cycle_on = false) THEN

    Case_Number:=0;

  END_IF;
END_IF;

```

Esimerkkikoodi 2. Toimintatilojen vaihto.

LCD_REM-funktio sisältää 12 parametria: Enable, Font, Line 1, Line 2, Line 3, Line 4, Line 5, Line 6, Line 7 ja Line 8 tulot sekä LCD_REM- ja Sts-lähdöt. Line 1–Line 8 -parametreihin määritetään merkkijonomuuttujat, joiden sisältö tulostetaan parametria vastaavalle riville näytöllä. Enable on looginen parametri, joka määrittää ehdon, onko funktio käytössä, Font-parametriin määritellään kokonaislukuna fontin koko. Ohjelman suorituksen aikana L1, L2, L3, L4, L5, L6, L7 ja L8 -merkkijonomuuttujien sisältöä muutetaan eri suoritusvaiheissa, jolloin niiden sisältö päivittyy näytölle. Esimerkkikutsu funktiolle kuvattu esimerkkikoodissa 3. [13]

```
LCD_REM_1(TRUE, Font_var, L1, L2, L3, L4, L5, L6, L7, L8);
```

Esimerkkikoodi 3. LCD_REM_1 kutsu

5.2 Turvalogiikka

Ohjelman suorituksen aikana luetaan jatkuvasti turvalogiikan X1-lähtöä, joka kertoo turvareleen tilan (tosi tai epätosi). Häätä-seis-painiketta painettaessa turvareleen tila muuttuu epätodeksi ja sen kärjet avautuvat, jonka seurauksesta niihin kytketyt käyttöjännitteet katkeavat sekä paineilma poistuu järjestelmästä. [14; 15]

Samanaikaisesti näytölle päivitetään operaattorille ohjeet, miten toimia toiminnan palauttamiseksi sekä kaikkien lähtöjen tila muutetaan epätodeksi. Nämä turvalogiikan toiminnot on kuvattu esimerkkikoodissa 4.

```

If (_IO_EM_DI_00 = FALSE) THEN

    L1 := 'EMERGENCY STOP';
    L2 := '';
    L3 := 'Press the blue button';
    L4 := 'on the cabinet';
    L5 := 'to resume operation';
    L6 := '';
    L7 := '';
    _IO_EM_DO_00:=False;
    _IO_EM_DO_01:=False;
    _IO_EM_DO_02:=False;
    _IO_EM_DO_03:=False;
    Cycle_on:=false;

END_IF;

```

Esimerkkikoodi 4. Turvalogiikka.

Hätä-seis-tilanteesta palautus onnistuu hätä-seis-painike vapauttamalla ja palauttamalla rele oletusarvoon kuittaus painikkeella. Kuittauksen tarvetta kuvaa sininen led-valo, joka syttyy hätä-seis-tilanteen johdosta, valon ohjaus on kuvattu esimerkkikoodissa 5.

```
_IO_EM_DO_04:=NOT _IO_EM_DI_00;
```

Esimerkkikoodi 5. Turvareleen kuittauspainikkeen LED-valon ohjaus.

5.3 Dataloki

Tiedon keräämiseen käytetään DLG-funktiota, joka mahdollistaa suoritettavan ohjelman muuttujien arvon kirjaamisen SD-kortilla olevaan lokitiedostoon. Lokitiedosto on .CSV-

muotoa, ja sen sisältöä pääsee lukemaan siirtämällä SD-kortin tietokoneeseen tai ottamalla yhteys logiikkaan Remote acces toolin ja Ethernet-kaapelin välityksellä. Remote acces tool asentuu Connected components workbenchin asennuksen yhteydessä. [12]

DLG-funktio sisältää yhteensä viisi parametria: Enable, TSEnable, CfgId, Status ja ErrorID. Enablen tietotyyppi on looginen, ja sen ollessa tosi funktio suoritetaan. TSEnablen tietotyyppi on looginen, ja se määrittää, ovatko aikaleimat käytössä. CfgId on etumerkitön kokonaisluku, joka määrittää, mikä datasetin ID-numero on käytössä (1–10). Dataset on ohjelmakohtainen käyttäjän määrittelemä kokoelma valittuja muuttujia, joiden arvoja halutaan tallentaa datalokiin. Status on etumerkitön 8-tavuinen kokonaisluku, joka ilmoittaa funktion senhetkisen tilan, esimerkiksi: 0=Idle, ja 1=Busy. ErrorID on etumerkitön 32-tavuinen kokonaisluku, joka tiedon kirjauksen epäonnistuessa tallentaa arvokseen virhetilaa vastaavan numeron, esimerkiksi 1="microSD card is missing." [12]

Suunnitellussa logiikkaohjelmassa DLG-funktion instanssi DLG_1 suoritetaan aina, kun automaatti tila on päällä, jota ilmaisee "Screen_var=0" ja sylinteri on laskettu alas, minkä johdosta anturi _IO_EM_D0_00 tila muuttuu todeksi (esimerkkikoodi 6). Aikaleimat ovat käytössä, ja käytössä on dataset 1, joka sisältää suoritettavan ohjelman tunnuksen, keittoveden lämpötilan ja keittoajan.

```
DLG_1(_IO_EM_DO_00 and Screen_var=0, TRUE, 1);
```

Esimerkkikoodi 6. DLG_1-funktion kutsu.

5.4 Mittaus

Pastörintiin käytettävän veden lämpötila mitataan keittopataan asetettavalla PT-100 anturilla. Anturilta saatu mittaustulos tallentuu logiikassa _IO_P1_A1_00-muuttujaan, jonka arvo muunnetaan celsiusasteiksi laitevalmistajan ohjeiden mukaisen kaavan avulla [8]. Lämpötilan laskemista ohjelman aikana on kuvattu esimerkkikoodissa 7.

```
temperature:=( _IO_P1_AI_00-2700)/10;
```

Esimerkkikoodi 7. Mittausarvon muuntaminen celsiusasteiksi temperature-muuttujaan.

5.5 Automaattitila

Automaattitilassa ohjelma odottaa käskyä operaattorilta ja aloittaa operaattorin valitseman aliohjelman. Logiikka vertaa keittoveden lämpötilaa valitun ohjelman ala- ja ylärajaan, jotka ovat Program1:n tapauksessa muuttujat Program1_temp_low ja Program1_temp_high (esimerkkikoodi 8). Mikäli lämpötila on sallittujen rajojen sisällä, voidaan ohjelma suorittaa, tai muussa tapauksessa tulostetaan ruudulle virheviesti, joka ilmoittaa lämpötilan olevan joko liian korkea tai matala valitulle ohjelmalle. Liian korkean tai alhaisen lämpötilan korjaamiseksi operaattorin täytyy nostaa tai laskea keittopadan lämpötilaa padan käyttöpaneelista ja odottaa lämpötilan muuttumista ennen kuin yrittää aloittaa ohjelman uudestaan.

Ohjelmia voi suorittaa vain yhden kerrallaan. Kun yhtä ohjelmaa suoritetaan, muuttuu "Cycle_on" -muuttujan tila todeksi, mikä estää muiden ohjelmien suorittamisen. Ohjelman suorittamisen voi keskeyttää ainoastaan seis-painikkeella tai hätäpysäytyksellä.

```
If( Key_data.4 = true and temperature >= Program1_temp_low and temperature <=
Program1_temp_high )THEN

    Count:= Count+1;
    Program1_bit:= TRUE;
    Case_Number:=1;

    //Program 1 error conditions

ELSIF( Key_data.4 = true and temperature>Program1_temp_high) THEN

    Case_Number:=6;

ELSIF( Key_data.4 = true and temperature<Program1_temp_low) THEN

    Case_Number:=7;

END_IF;
```

Esimerkkikoodi 8. Automaattitilan Program1-ohjelman aloitusehdot. Mikäli ehdot täyttyvät painiketta painaessa, ohjelma aloitetaan (case 1) tai jokin virheviesti tulostetaan (case 6 ja case 7). Muille ohjelmille on vastaava aloitusehto rakenne.

5.6 Manuaalitila

Manuaalillassa logiikka muuttaa lähtöjen 0–3 tilan vastaamaan suoraan niille määrätyn painikkeen tilaa. Esimerkiksi painiketta F1 painaessa sylinteri 1 liikkuu ylös ja vastaavasti F2 painaessa alas (esimerkkikoodi 9). Manuaalitila on tarkoitettu lähinnä huoltotilanteita tai muita erikoistilanteita varten.

```
//MANUAL MODE
5:
    L1 := 'Manual operation';
    L2 := 'F1 : Cyl 1 up';
    L3 := 'F2 : Cyl 1 down';
    L4 := 'F3 : Cyl 2 up';
    L5 := 'F4 : Cyl 2 down';
    L6 := '';
    L7 := 'Temperature: ' + ANY_TO_STRING(temperature);
    L8 := '';
If (screen_var = 1 and _IO_EM_DI_00 = true) THEN
    _IO_EM_DO_00:=Key_data.4;
    _IO_EM_DO_02:=Key_data.6;
    _IO_EM_DO_01:=Key_data.5;
    _IO_EM_DO_03:=Key_data.7;
END_IF;
```

Esimerkkikoodi 9. Manuaalitalan valikko.

6 Yhteenveto

Alkuperäiseen tavoitteeseen ei päästy aikataulun puitteissa, mistä johtuen tavoitetta muutettiin. Tarkoituksena oli saada tuotettua hyötyä asiakkaalle ja ratkaista mahdollisimman monta haastetta alkuperäisestä tavoitteesta.

6.1 Aikataulussa pysyminen

Aikataulu projektille oli neljä kuukautta, ja tavoitteena koko pastörintiprosessin automatisointi esipesu ja jäähdytys mukaan luettuna. Hyvin pian kuitenkin jouduttiin toteamaan, ettei alkuperäiseen tavoitteeseen päästä siinä ajassa. Tavoiteaikataulu osoittautui liian

lyhyeksi, koska projektin työstäminen alkoi myöhemmin kuin oli suunniteltu johtuen tilattujen tarvikkeiden toimitusajasta. Lisäksi viivästyksiä aiheuttivat erityisesti mekaanisen suunnittelun haasteet ja valintaan liittyvä päätöksenteko. Tästä johtuen päätettiin keskittyä logiikkaohjelman toiminnallisuuteen ja sen käyttöönottoon tuotannossa supistetussa muodossa. Koko prosessin automatisointia varten luotiin alustava konseptisuunnitelma, jonka pohjalta voi edetä pidemmälle suunnittelussa.

6.2 Lopputulos

Pastörintia varten saatiin luotua valmis logiikkaohjelma, joka sisältää kaikki välttämättömät toiminnallisuudet, jotta sitä voidaan käyttää sellaisenaan tuotannossa asentamalla sylinteri kiinteästi keittopadan yläpuolelle. Kaikki sähkökomponentit on asennettu kytkentäkoteloon, josta tarvittavat ohjausjännitteet saadaan kytkettyä moninapakaapelin avulla asennuksen yhteydessä.

6.3 Etä- ja itsenäisen työskentelyn haasteet

Vallitsevien koronavirusepidemiaan liittyvien suositusten takia [16] työskentely tapahtui etänä ja itsenäisesti. Asiakkaan kanssa oltiin yhteydessä puhelimitse sekä videoneuvottelujen välityksellä. Tämä aiheutti ongelmia projektin etenemisessä ja osaltaan myös opetti hyviä toimintatapoja, miten saada työt tehtyä näissä olosuhteissa. Kommunikoinnin merkitys korostuu: on tärkeää keskustella projektin etenemisestä viikoittain ja varmistaa että sovitut asiat on saatu tehtyä. Lisäksi työajanseurantajärjestelmät ja avoin sisällön tuottaminen pilvipalveluihin helpottavat kaikkia osallisia seuraamaan projektin etenemistä.

6.4 Jatkokehitysmahdollisuudet

Raporttien luominen

Keskustelujen pohjalta todettiin tärkeimmäksi ominaisuudeksi mahdollisuus saada automaattisia raportteja työkiirroista logiikalta PDF-muodossa joko sähköpostiin tai tiedostoksi tallennettuna paikallisesti tai pilvipalveluun sekä pystyä vaikuttamaan siihen, mitä tietoa raportteihin sisällytetään. Nykyinen versio datalokista on alkeellinen tekstitiedosto, joka tallentaa aikaleimalla valittujen muuttujien arvot, jotka on siirrettävä omalle PC:lle verkkokaapelin avulla.

Punnitus

Laadunvalvonnan kannalta on oleellista, että logiikka pystyisi punnitsemaan siivilässä olevan siemenerän ja tiedon avulla säätämään keittoaikaa tai antamalla virheilmoituksen. Punnitsemisen toteuttamista varten täytyy löytää oikea anturityyppi, joka voidaan asentaa tarttujaan ja saada punnitustulos käännettyä logiikan ymmärrettävään muotoon.

Käyttöliittymän parantaminen

Käyttökokemuksen parantaminen onnistuu helpoiten investoimalla kosketusnäyttöön, joka on tarpeeksi suuri, niin että siihen saadaan operaattorin tarvitsemat painikkeet ohjelmoitua ja tilaa jää vielä tarvittavan tiedon välittämiseen. Lisäksi edistyneempi näyttöpääte mahdollistaa edistyneempiä ohjelmointityökaluja, joiden avulla voidaan luoda käyttäjätilejä ja saada käyttäjätiedot tallennettua datalokiin ja raportteihin. Käyttäjätilien lisäksi voidaan luoda pääkäyttäjä, jolla on pääsy muuttamaan prosessiarvoja kuten käytössä olevia aikaviiveitä graafisen käyttöliittymän avulla, jolloin prosessin säätäminen onnistuu ilman ohjelmointiosaamista.

Keittoveden lämpötilan säätö logiikkaohjaimen avulla

Toimintavarmuuden lisäämiseksi voisi tutkia mahdollisuuksia keittopadan manuaalisen lämpötilasäädön poistamiseksi käyttöpaneelin kautta ja antaa logiikan säätää padan lämpötilaa. Tämä poistaisi mahdollisen virhetilanteen, jossa pastörinti aloitetaan sallituissa rajoissa ja painetaan lämpötilaa ylös tai alas säätävää painiketta tahattomasti.

Tämän toimintamallin toteuttaminen vaatii ohjelmaan rakenteellisia muutoksia ja mahdollisesti säätöpiirin toteuttamista lämpötilan ylläpitämiseen. Lisäksi se helpottaisi pidemmälle automatisoidun prosessin käyttämistä, jossa koneellisesti syötetään uusi erä pastöroitavaksi keittopataan. Logiikan säätäessä lämpötilaa ei operaattorin tarvitse päästä tekemään muutoksia käyttöpaneeliin.

Lähteet

- 1 Silmusalaatti. Verkkoaineisto. Silmusalaatti. <<https://www.silmusalaatti.fi>>. Luettu 23.4.2021
- 2 Liite 10 Itujen versojen ja krassien tuotanto. 2020. Verkkoaineisto. Ruokavirasto. <<https://www.ruokavirasto.fi/globalassets/tietoa-meista/asiointi/oppaat-jalomakkeet/yritykset/elintarvikeala/elintarvikealan-oppaat/liite-10-itujen-versojen-ja-krassien-tuotanto.pdf>>. Luettu 26.4.2021.
- 3 Usein kysyttyä. Verkkoaineisto. Silmusalaatti. <<https://www.silmusalaatti.fi/usein-kysyttya/>>. Luettu 23.4.2021
- 4 Tuotteet. Verkkoaineisto. Silmusalaatti. <<https://www.silmusalaatti.fi/tuotteet/>>. Luettu 23.4.2021.
- 5 Uudellamaalla Digi-Salamoi. Verkkoaineisto. Metropolia Ammattikorkeakoulu. <<https://digisalama.metropolia.fi/hanke/>>. Luettu 23.4.2021.
- 6 CULINO KEITTOPATA. 2021. Verkkoaineisto. Metos. <<https://www.metos.com/pdf/prods/userguide/Fl/4215209.pdf>>. Luettu 26.4.2021.
- 7 Usein kysyttyä EHECistä. 2018. Verkkoaineisto. Ruokavirasto. <<https://www.ruokavirasto.fi/henkiloasiakkaat/tietoa-elintarvikkeista/elintarvikkeiden-turvallisen-kayton-ohjeet/ruokamyrkytykset/ruokamyrkytyksia-aiheuttavia-bakteereja/escherichia-coli/usein-kysyttya-ehecista/>>. Luettu 23.4.2021.
- 8 Micro800 Plug-in Modules. 2018. Verkkoaineisto. Rockwell Automation. <https://literature.rockwellautomation.com/idc/groups/literature/documents/um/2080-um004_-en-e.pdf>. Luettu 23.4.2021.
- 9 Miten toimii Pt100-anturi. 2020. Verkkoaineisto. EPIC® SENSORS. <<https://www.epicsensors.fi/faq/miten-pt100-anturi-toimii/>>. Luettu 23.4.2021.
- 10 Nickel plated steel tension load cell ZFA. Verkkoaineisto. Scaime. <<https://scaime.com/product/post/zfa>>. Luettu 24.4.2021.
- 11 Strain gage conditioner CPJ / CPJ2S. Verkkoaineisto. Scaime. <<https://scaime.com/product/post/cpj---cpj2s>>. Luettu 24.4.2021.
- 12 Micro820™ Programmable Logic Controller. 2013. Verkkoaineisto. Rockwell Automation. <https://literature.rockwellautomation.com/idc/groups/literature/documents/pp/2080-pp004_-en-p.pdf>. Luettu 24.4.2021.

- 13 Micro820 Programmable Controllers User manual. 2019. Verkkoaineisto. Rockwell Automation. <https://literature.rockwellautomation.com/idc/groups/literature/documents/um/2080-um005_-en-e.pdf>. Luettu 24.4.2021.
- 14 Safety Relay Unit. User's Manuals. Verkkoaineisto. Omron. <https://assets.omron.eu/downloads/manual/en/v2/4022078_g9se-201_g9se-221-t_instruction_sheet_en.pdf>. Luettu 24.4.2021
- 15 Modular F.R.L. Units. Verkkoaineisto. SMC. <https://static.smc.eu/pdf/AC-B-B_EU.pdf> Luettu 24.4.2021
- 16 Alueellinen koronavirusepidemiatilanne ja suositukset. 2021. Verkkoaineisto. HUS. <<https://www.hus.fi/ajankohtaista/koronavirus-covid-19/alueellinen-koronavirusepidemiatilanne-ja-suositukset>>. Luettu 24.4.2021.

Laiteluettelo

Tuotekoodi	Selite
Micro820-2080-LC20-20QBB	Logiikkaohjain
2080-PS120-240VAC	Logiikan virtalähde
Micro800 Remote LCD Display	LCD-näyttöpääte
2080-RTD2	Logiikan plugin-moduuli lämpötilan mittaamiseen
D-M9PWL	3-johdin PNP-anturi (2 kappaletta)
MCN106E	1-napainen johdonsuojakatkaisija, C 6
AC20-F02CE-XFI02	Paineilman huoltolaite ja turvaventtiili
CP96SDB32-700C	Sylinteri 1
CD85N16-300-B-X2018	Sylinteri 2
	SMC 5/2 -venttiili (2 kappaletta)
	Danfoss 24v -venttiili
	PT-100-lämpötilanmittausanturi
OM679415	Omron G9SE-201 -turvarele
R1507000	Riviliitinkotelo 300 x 300 x 200 mm

Global variables

<input type="checkbox"/>	_IO_EM_DO_00	Cylinder_1_Up	BOOL	▼
	_IO_EM_DO_01	Cylinder_1_Down	BOOL	▼
<input type="checkbox"/>	_IO_EM_DO_02	Cylinder_2_Up	BOOL	▼
	_IO_EM_DO_03	Cylinder_2_Down	BOOL	▼
<input type="checkbox"/>	_IO_EM_DO_04		BOOL	▼
	_IO_EM_DO_05		BOOL	▼
<input type="checkbox"/>	_IO_EM_DO_06		BOOL	▼
	_IO_EM_DI_00	Safety_Relay_status	BOOL	▼
<input type="checkbox"/>	_IO_EM_DI_01		BOOL	▼
	_IO_EM_DI_02		BOOL	▼
<input type="checkbox"/>	_IO_EM_DI_03		BOOL	▼
	_IO_EM_DI_04	Censor_top	BOOL	▼

Local variables

DLG_1	DLG
LCD_REM_1	LCD_REM
L1	STRING
Font_var	UDINT
L2	STRING
L3	STRING
L4	STRING
L5	STRING
L6	STRING
L7	STRING
L8	STRING
KEY_READ_REM_1	KEY_READ_REM
Key_data	UDINT
iterations	INT
temperature	UINT
Case_Number	INT
Cycle_on	BOOL
Screen_var	INT
Program1_bit	BOOL
Program2_bit	BOOL
Program3_bit	BOOL
Program4_bit	BOOL
Cycle_on_TOF	TOF
Cylinder_Down_RS	SR
Cycle_On_RS	RS
Program_timer	TON
Cylinder_up_RS	RS
Cylinder2_up_RS	RS
Cylinder2_timer2	TON
Cylinder2_down_RS2	RS
Start_Delay	TON
Cylinder2_down_RS1	RS
Reset_bit	BOOL
Cylinder2_timer1	TON
Cycle_On_RS2	RS
Cylinder2_timer3	TON
Cylinder2_up_RS2	RS
Cylinder2_timer4	TON
Start_Delay_RS	RS

Logiikkaohjelman lähdekoodi

```
//Initial module calls and other constants

KEY_READ_REM_1(TRUE);
Key_Data := KEY_READ_REM_1.KeyData
//Key presses are stored in Key_Data variable.

LCD_REM_1(TRUE, Font_var, L1, L2, L3, L4, L5, L6, L7, L8);
//LCD_REM updates the status text on remote LCD panel. Variable strings L1,
L2, L3 store the information that is printed on the display.

temperature:=(_IO_P1_AI_00-2700)/10;
//Temperature calculation

_IO_EM_DO_04:=NOT _IO_EM_DI_00;
//Led output that is activated when safety relay status is false

DLG_1(_IO_EM_DO_00 and Screen_var=0, TRUE, 1);
//Datalogging is active while cylinder is moving up and automatic mode is ac-
tivated module call, Config ID can be 1-10

//Safety relay status check

If(_IO_EM_DI_00 = FALSE) THEN

    L1 := 'EMERGENCY STOP';
    L2 := '';
    L3 := 'Press the blue button';
    L4 := 'on the cabinet';
    L5 := 'to resume operation';
    L6 := '';
    L7 := '';
    _IO_EM_DO_00:=False;
    _IO_EM_DO_01:=False;
    _IO_EM_DO_02:=False;
    _IO_EM_DO_03:=False;

    Cycle_on:=false;
END_IF;

//Cycle on variable controls

Cycle_on_TOF(_IO_EM_DO_01 OR _IO_EM_DO_00 OR _IO_EM_DO_03 OR _IO_EM_DO_02 or
Cycle_On_RS.Q1, T#500ms);
//Cycle_on is true for 1 second after any of the cylinders has stopped moving

Cycle_on:=Cycle_on_TOF.Q or NOT _IO_EM_DI_00
// stays true for program duration with off delay Q and emergency stop

//MENU SCREEN FUNCTIONS
```



```
If(_IO_EM_DI_00 = TRUE) THEN

//When safety relay status input is true this is executed.

    if(Key_data.0 = true and Cycle_on = FALSE) THEN

//IF structure for increasing or decreasing screen_var by up and down key
presses on LCD panel

        Screen_var:=0;
        Case_Number:=0;

        ELSIF(Key_data.1 = true and Cycle_on = FALSE) THEN

            Screen_var:=1;
            Case_Number:=5;

        END_IF;

    if(screen_var = 0 and Cycle_on = false) THEN

        Case_Number:=0;

    END_IF;

END_IF;

//AUTOMATIC MODE PROGRAM START CONDITIONS

    if(_IO_EM_DI_00 = TRUE and Screen_var =0 and Cycle_on = FALSE) THEN

//Program 1 start condition

If( Key_data.4 = true and temperature >= Program1_temp_low and temperature <=
Program1_temp_high )THEN

//Pea starting conditions

        Count:= Count+1;
        Program1_bit:= TRUE;
        Case_Number:=1;

//Program 1 error conditions

ELSIF( Key_data.4 = true and temperature>Program1_temp_high) THEN
//Temperature high error condition

        Case_Number:=6;

ELSIF( Key_data.4 = true and temperature<Program1_temp_low) THEN
//Temperature low error condition

        Case_Number:=7;

    END_IF;

//Program 2 start condition
```

```
IF(Key_data.5 = true and temperature >= Program2_temp_low and temperature <=
Program2_temp_high)THEN
//Lentil starting conditions

                Count:= Count+1;
                Program2_bit:=TRUE;
                Case_Number:=2;

//Program 2 error conditions

                ELSIF(Key_data.5 = true and temperature>Program2_temp_high)
THEN
//Temperature high error condition

                Case_Number:=8;

ELSIF(Key_data.5 = true and temperature<Program2_temp_low) THEN
//Temperature low error condition

                Case_Number:=9;

                END_IF;

//Program 3 start condition

IF(Key_data.6 = true and temperature >= Program3_temp_low and temperature <=
Program3_temp_high)THEN
//Alfala, alpila, retikka starting conditions

                Count:= Count+1;
                Program3_bit:=TRUE;
                Case_Number:=3;

//Program 3 error conditions

ELSIF(Key_data.6 = true and temperature>Program3_temp_high) THEN
//Temperature high error condition

                Case_Number:=10;

ELSIF(Key_data.6 = true and temperature<Program3_temp_low) THEN
//Temperature low error condition

                Case_Number:=11;

                END_IF;

//Program 4 start condition

IF(Key_data.7 = true and temperature >= Program4_temp_low and temperature <=
Program4_temp_high)THEN
//Broccoli starting conditions

                Count:= Count+1;
                Program4_bit:= TRUE;
                Case_Number:=4;

//Program 4 error conditions

ELSIF(Key_data.7 = true and temperature>Program4_temp_high) THEN
//Temperature high error condition
```

```
        Case_Number:=12;

ELSIF(Key_data.7 = true and temperature<Program4_temp_low) THEN
//Temperature low error condition

        Case_Number:=13;

        END_IF;

//Reset count start condition

        IF(Key_data.8 = true)THEN

                Count := 0;

        End_IF;

END_IF;

CASE Case_Number OF
//Menu Screen

    0:
        L1 := 'Automatic operation';
        L2 := 'F1 : Pea';
        L3 := 'F2 : Lentil';
        L4 := 'F3 : Alfala/alpila/retikka';
        L5 := 'F4 : Broccoli';
        L6 := 'F5: Reset count F6: STOP';
        L7 := 'Temperature: ' + ANY_TO_STRING(temperature);
        L8 := 'Count: ' + ANY_TO_STRING(Count);
        _IO_EM_DO_00:=False;
        _IO_EM_DO_01:=False;
        _IO_EM_DO_02:=False;
        _IO_EM_DO_03:=False;

//Program 1

    1:
        IF(Key_data.9 = true)THEN

                Case_Number:=0;

        END_IF;

        L1 := 'Executing program 1';
        L2 := '';
        L3 := '';
        L4 := '';
        L5 := '';
        L6 := 'F6: STOP';
        L7 := '';
        L8 := '';

//Cylinder 1 movements
Cylinder_Down_RS(Program1_bit or Program2_bit or Program3_bit or Program4_bit,
_IO_EM_DI_05 or Key_data.9);
        _IO_EM_DO_01:=Cylinder_Down_RS.Q1;
//Moving cylinder 1 down
```

d

```

//Starting condition turns back to false
    Program1_bit:=False;

//starting program timer when cylinder 1 is down and censored has activated
//Cycle_on_RS keeps cycle on during the wait
    Cycle_On_RS(_IO_EM_DI_05, Program_timer.Q or Key_data.9);
    Program_timer(Cycle_on and NOT Key_Data.9 AND _IO_EM_DI_05, T#20S);

//Moving cylinder1 up after program timer has finished.
    Cylinder_up_RS(Program_timer.Q, _IO_EM_DI_04 or Key_data.9);
    _IO_EM_DO_00:=Cylinder_up_RS.Q1;

//Cylinder2 movement

//2 On delay timers move cylinder 2 back and forth in a loop

Cylinder2_timer1(NOT _IO_EM_DO_02 and _IO_EM_DI_05, T#1000ms
//This delay determines how long cylinder 2 does up movement
_IO_EM_DO_03:=Cylinder2_timer1.Q and NOT TOF_1.Q and NOT Cylinder_up_RS.Q1;

Cylinder2_timer2(_IO_EM_DO_03 and _IO_EM_DI_05 or _IO_EM_DO_00 , T#1000ms);

    TOF_1(Cylinder2_timer2.Q, T#1000ms);

//This Off delay controls how long cylinder 2 stays down
    _IO_EM_DO_02:=TOF_1.Q;

//End of program 1

//Program 2

    2:
        IF(Key_data.9 = true)THEN
            Case_Number:=0;

            END_IF;

            L1 := 'Executing program 2';
            L2 := '';
            L3 := '';
            L4 := '';
            L5 := '';
            L6 := 'F6: STOP';
            L7 := '';
            L8 := '';

//Cylinder 1 movements
//Moving cylinder 1 down
Cylinder_Down_RS(Program1_bit or Program2_bit or Program3_bit or Program4_bit,
_IO_EM_DI_05 or Key_data.9);
    _IO_EM_DO_01:=Cylinder_Down_RS.Q1;

//Starting condition turns back to false
    Program2_bit:=False;

```

```

//starting program timer when cylinder 1 is down and censor has activated
//Cycle_on_RS keeps cycle on during the wait
    Cycle_On_RS(_IO_EM_DI_05, Program_timer.Q or Key_data.9);
    Program_timer(Cycle_on and NOT Key_Data.9 AND _IO_EM_DI_05, T#20S);

//Moving cylinder1 up after program timer has finished.
    Cylinder_up_RS(Program_timer.Q, _IO_EM_DI_04 or Key_data.9);
    _IO_EM_DO_00:=Cylinder_up_RS.Q1;

//Cylinder2 movement

//2 On delay timers move cylinder 2 back and forth in a loop
    Cylinder2_timer1(NOT _IO_EM_DO_02 and _IO_EM_DI_05, T#1000ms);

//This delay determines how long cylinder 2 does up movement
    _IO_EM_DO_03:=Cylinder2_timer1.Q and NOT TOF_1.Q and NOT Cylin-
der_up_RS.Q1;

    Cylinder2_timer2(_IO_EM_DO_03 and _IO_EM_DI_05 or _IO_EM_DO_00,
T#1000ms);
    TOF_1(Cylinder2_timer2.Q, T#1000ms);

//This Off delay controls how long cylinder 2 stays down
    _IO_EM_DO_02:=TOF_1.Q;

//End of program 2

//Program 3

    3:
        IF(Key_data.9 = true)THEN
            Case_Number:=0;

        END_IF;

        L1 := 'Executing program 3';
        L2 := '';
        L3 := '';
        L4 := '';
        L5 := '';
        L6 := 'F6: STOP';
        L7 := '';
        L8 := '';

//Moving cylinder 1 down
    Cylinder_Down_RS(Program1_bit or Program2_bit or Program3_bit or
Program4_bit, _IO_EM_DI_05 or Key_data.9);
    _IO_EM_DO_01:=Cylinder_Down_RS.Q1;

//Starting condition turns back to false
    Program3_bit:=False;

//starting program timer when cylinder 1 is down and censor has activated
//Cycle_on_RS keeps cycle on during the wait
    Cycle_On_RS(_IO_EM_DI_05, Program_timer.Q or Key_data.9);
    Program_timer(Cycle_on and NOT Key_Data.9 AND _IO_EM_DI_05, T#20S);

//Moving cylinder1 up after program timer has finished.

```

```
Cylinder_up_RS(Program_timer.Q, _IO_EM_DI_04 or Key_data.9);
_IO_EM_DO_00:=Cylinder_up_RS.Q1;

//Cylinder2 movement

//2 On delay timers move cylinder 2 back and forth in a loop
Cylinder2_timer1(NOT _IO_EM_DO_02 and _IO_EM_DI_05, T#1000ms);

//This delay determines how long cylinder 2 does up movement
_IO_EM_DO_03:=Cylinder2_timer1.Q and NOT TOF_1.Q and NOT Cylin-
der_up_RS.Q1;

Cylinder2_timer2(_IO_EM_DO_03 and _IO_EM_DI_05 or _IO_EM_DO_00, T#1000ms);

TOF_1(Cylinder2_timer2.Q, T#1000ms);

//This Off delay controls how long cylinder 2 stays down
_IO_EM_DO_02:=TOF_1.Q;

//End of program 3

//Program 4

4:
  IF(Key_data.9 = true)THEN

    Case_Number:=0;

    END_IF;

    L1 := 'Executing program 4';
    L2 := '';
    L3 := '';
    L4 := '';
    L5 := '';
    L6 := 'F6: STOP';
    L7 := '';
    L8 := '';

    //Moving cylinder 1 down
    Cylinder_Down_RS(Program1_bit or Program2_bit or Program3_bit or Program4_bit,
    _IO_EM_DI_05 or Key_data.9);
    _IO_EM_DO_01:=Cylinder_Down_RS.Q1;

    //Starting condition turns back to false
    Program4_bit:=False;

    //starting program timer when cylinder 1 is down and censor has activated
    //Cycle_on_RS keeps cycle on during the wait
    Cycle_On_RS(_IO_EM_DI_05, Program_timer.Q or Key_data.9);
    Program_timer(Cycle_on and NOT Key_Data.9 AND _IO_EM_DI_05, T#20S);

    //Moving cylinder1 up after program timer has finished.
    Cylinder_up_RS(Program_timer.Q, _IO_EM_DI_04 or Key_data.9);
    _IO_EM_DO_00:=Cylinder_up_RS.Q1;
```

```

//Cylinder2 movements

//2 On delay timers move cylinder 2 back and forth in a loop
  Cylinder2_timer1(NOT _IO_EM_DO_02 and _IO_EM_DI_05, T#1000ms);

//This delay determines how long cylinder 2 does up movement
  _IO_EM_DO_03:=Cylinder2_timer1.Q and NOT TOF_1.Q and NOT Cylin-
der_up_RS.Q1;

      Cylinder2_timer2(_IO_EM_DO_03 and _IO_EM_DI_05 or _IO_EM_DO_00,
T#1000ms);
      TOF_1(Cylinder2_timer2.Q, T#1000ms);

//This Off delay controls how long cylinder 2 stays down
  _IO_EM_DO_02:=TOF_1.Q;

//End of program 4

//MANUAL MODE

5:

  L1 := 'Manual operation';
  L2 := 'F1 : Cyl 1 up';
  L3 := 'F2 : Cyl 1 down';
  L4 := 'F3 : Cyl 2 up';
  L5 := 'F4 : Cyl 2 down';
  L6 := '';
  L7 := 'Temperature: ' + ANY_TO_STRING(temperature);
  L8 := '';

  If (screen_var = 1 and _IO_EM_DI_00 = true) THEN
    _IO_EM_DO_00:=Key_data.4;
    _IO_EM_DO_02:=Key_data.6;
    _IO_EM_DO_01:=Key_data.5;
    _IO_EM_DO_03:=Key_data.
  END_IF;

//Error cases
6:

  L1 := 'ERROR';
  L2 := 'Temperature too high';
  L3 := 'for peas.';
  L4 := '';
  L5 := '';
  L6 := '';
  L7 := 'Temperature: ' + ANY_TO_STRING(temperature);
  L8 := 'Count: ' + ANY_TO_STRING(Count);

7:

  L1 := 'ERROR';
  L2 := 'Temperature too low';
  L3 := 'for peas.';
  L4 := '';
  L5 := '';
  L6 := '';
  L7 := 'Temperature: ' + ANY_TO_STRING(temperature);
  L8 := 'Count: ' + ANY_TO_STRING(Count);

```

```
8:
    L1 := 'ERROR';
    L2 := 'Temperature too high';
    L3 := 'for lentil.';
    L4 := '';
    L5 := '';
    L6 := '';
    L7 := 'Temperature: ' + ANY_TO_STRING(temperature);
    L8 := 'Count: ' + ANY_TO_STRING(Count);
9:
    L1 := 'ERROR';
    L2 := 'Temperature too low';
    L3 := 'for lentil.';
    L4 := '';
    L5 := '';
    L6 := '';
    L7 := 'Temperature: ' + ANY_TO_STRING(temperature);
    L8 := 'Count: ' + ANY_TO_STRING(Count);
10:
    L1 := 'ERROR';
    L2 := 'Temperature too high';
    L3 := 'for Alfala/alpila.';
    L4 := '';
    L5 := '';
    L6 := '';
    L7 := 'Temperature: ' + ANY_TO_STRING(temperature);
    L8 := 'Count: ' + ANY_TO_STRING(Count);
11:
    L1 := 'ERROR';
    L2 := 'Temperature too low';
    L3 := 'for Alfala/alpila.';
    L4 := '';
    L5 := '';
    L6 := '';
    L7 := 'Temperature: ' + ANY_TO_STRING(temperature);
    L8 := 'Count: ' + ANY_TO_STRING(Count);
12:
    L1 := 'ERROR';
    L2 := 'Temperature too high';
    L3 := 'for Broccoli';
    L4 := '';
    L5 := '';
    L6 := '';
    L7 := 'Temperature: ' + ANY_TO_STRING(temperature);
    L8 := 'Count: ' + ANY_TO_STRING(Count);
13:
    L1 := 'ERROR';
    L2 := 'Temperature too low';
    L3 := 'for Broccoli.';
    L4 := '';
    L5 := '';
    L6 := '';
    L7 := 'Temperature: ' + ANY_TO_STRING(temperature);
    L8 := 'Count: ' + ANY_TO_STRING(Count);

END_CASE;
```


Laitteiston käyttöohjeet ja muut asiakirjat

[Micro800 Plug-in Modules](#)

[Micro800™ Programmable Controller External AC Power Supply](#)

[Micro800 Programmable Controllers General Instructions](#)

[Omron G9SE-201 Safety Relay Unit Users manual](#)

[Micro820 Product profile](#)

[Micro820 user manual](#)

[Micro800 remote LCD installation instructions](#)