

Taneli Kaukoranta

Urakanlaskentasovellus

Opinnäytetyö

Kevät 2018

SeAMK Tekniikka

Automaatiotekniikka



SEINÄJOEN AMMATTIKORKEAKOULU
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikan yksikkö

Tutkinto-ohjelma: Automaatiotekniikka

Suuntautumisvaihtoehto: Sähköautomaatio

Tekijä: Taneli Kaukoranta

Työn nimi: Urakanlaskentasovellus

Ohjaaja: Niko Ristimäki

Vuosi: 2018

Sivumäärä: 34

Liitteiden lukumäärä: 1

Tämän opinnäytetyön tavoitteena oli suunnitella ja toteuttaa kosketuspaneelilla ja ohjelmoitavalla logiikalla helppokäyttöinen käyttöliittymä urakanlaskentaan, korvaamaan vanhaa mekaanista leimakelloa. Sovelluksen oli lähetettävä urakan raportit sähköpostin liitteenä työnjohtajille tai heidän sijaisille. Projektin toteutettiin, Siemensin TP700 comfort -kosketuspaneelilla ja S7-1211C -ohjelmoitavalla logiikalla.

Lopputuloksena oli helppokäyttöinen käyttöliittymä, joka vaatii käyttäjältään vain pienen perehdytyksen. Tässä raportissa syvennytään kuitenkin enemmän logiikka-ohjelmaan, joka on helpon käyttöliittymän alla melko monimutkainen.

Avainsanat: CSV, sähköposti, TP700, S7-1200, TIA-portal

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Faculty: School of Technology

Degree programme: Automation Engineering

Specialisation: Electric Automation

Author: Taneli Kaukoranta

Title of thesis: Contract Calculation Application

Supervisor: Niko Ristimäki

Year: 2018

Number of pages: 34

Number of appendices: 1

The goal of this thesis was to design and implement an easy to use interface for counting contracts, using a touch panel and a plc. The interface would replace an old mechanical punch clock. The application would have to send the contract report to the foremen and their substitutes through email. The project was carried out with the Siemens TP700 comfort –touch panel and S7-1211C –plc.

The outcome of this project was an easy to use interface that requires only little introduction. However, in this thesis more attention was paid to the logic program which is quite complex under the easy to use interface.

Keywords: CSV, E-mail, TP700, S7-1200, TIA-portal

SISÄLTÖ

Opinnäytetyön tiivistelmä.....	1
Thesis abstract.....	2
SISÄLTÖ.....	3
Kuvaluettelo	5
Käytetyt termit ja lyhenteet	6
1 JOHDANTO	8
1.1 Työn tausta	8
1.2 Työn tavoite	8
1.3 Atria Oyj.....	8
1.3.1 Atria Suomi Oy.....	9
2 TYÖNTUTKIMUS.....	10
2.1 Työ mittauskohteena.....	10
2.2 Ajanmäärittämistavat	11
2.2.1 Havainnointitutkimus.....	11
2.2.2 Ajankäyttötutkimus.....	12
2.2.3 Normaaliaikatutkimus.....	12
2.2.4 Liikeikatutkimus	12
2.3 Urakka a-sikateurastamossa	13
3 OHJELMOITAVAT LOGIIKAT.....	14
3.1 Simatic-logiikat.....	14
4 SUUNNITTELU JA TOTEUTUS.....	15
4.1 Projektin kuvaus.....	15
4.2 Laitteiden valinta	16
4.2.1 Kosketuspaneelin valinta	16
4.2.2 Logiikan valinta	18
4.3 Logiikkaohjelma	19
4.3.1 Lihaskaurakka.....	19
4.3.2 Emakkourakka	21
4.3.3 Kellot.....	22
4.3.4 Asetukset.....	22

4.3.5 Laskurit	22
4.3.6 Timerit.....	23
4.3.7 Create_string	23
4.3.8 Datablockien välinen tiedonsiirto	23
4.3.9 Email.....	23
4.4 CSV-tiedoston luominen paneelilla	24
4.4.1 CSV-tiedosto hälytyslistalla.....	24
4.4.2 CSV-tiedosto VBscriptillä	24
4.5 Sähköpostin lähettäminen logiikalla ja paneelilla	25
4.5.1 Sähköpostin lähettäminen paneelilla	26
4.5.2 Sähköpostin lähettäminen logiikalla	26
5 YHTEENVETO JA POHDINTA	33
LÄHTEET	34
LIITTEET	35
LIITE 1 Malliraportti	1

Kuvaluettelo

Kuva 1. Siemens MP270 -kosketuspaneeli (Siemens [Viitattu 13.3.2017].) 17

Kuva 2. Siemens KTP700 -näppäinkosketuspaneeli (Siemens 2017a). 17

Kuva 3. Siemens TP900 Comfort -kosketuspaneeli (Siemens 2017c). 18

Kuva 4. Ohjelmaosiot, jotka pitävät sisällään logiikkaohjelman 19

Kuva 5. Lihaskaurakan aloitusbitin set/reset 20

Kuva 6. Lihaskaurakka päällä -bitin Set/Reset 20

Kuva 7. DB1, käynnissä olevan urakan muistialue 21

Kuva 8. Tietojen tallentaminen CSV-tiedostoon 25

Kuva 9. Tmail_C-komento, joka lähettää sähköpostin 27

Kuva 10. Excel-taulukon purku-script 28

Kuva 11. Array of WString DB, joka sisältää CSV-tiedostosta puretun informaation 29

Kuva 12. Päivämäärän poiston SCL-koodi 30

Kuva 13. Liitteen nimen luontikoodi 30

Kuva 14. Concat-komento, joka yhdistää kaksi stringiä yhdeksi 30

Kuva 15. Strg_TO_Chars-komento, joka muuntaa stringin char-muotoon 31

Kuva 16. STL-koodi, joka etsii liitteestä huutomerkkin ASCII-koodia ja muuttaa sen rivinvaihdoksi 31

Kuva 17. Tmail_C_param datablock, johon määritellään sähköpostin asetukset 32

Käytetyt termit ja lyhenteet

Datablock	Muistialue, jonne tallennetaan aliohjelmien tuloksia.
Emakko	Naaraspuolinen sika, joka on synnyttänyt porsaita.
Firmware	Laitteen sisäinen ohjelmisto, joka ohjaa osin tai kokonaan laitteen toimintaa.
Function	Aliohjelma ilman staattista muistia.
Functionblock	Aliohjelma staattisella muistilla.
Induktiivinen anturi	Anturi, joka tunnistaa metalleja.
Kinneri	Metallinen, henkarin mallinen esine, jonka avulla sika kulkee, takajaloista roikkuen, rataa pitkin teurastamossa ja leikkaamossa.
Kosketuspaneeli	Kosketukseen reagoiva, ohjelmoitava näyttö.
KTP	Key touch panel.
Lihasika	Noin 6 kk:n ikäinen ja elopainoltaan noin 110 kiloinen sika.
MP	Multi panel.
Organisationblock	Organisointiohjelma, jonka sisälle listataan aliohjelmiä siinä järjestyksessä, jossa ne halutaan suorittaa.
Step7	Siemens-logiikoiden ohjelmointityökalu.
SMTP	Simple mail transfer protocol, sähköpostin välitysprotokolla.
TIA-portal	Totally integrated automation portal on ohjelmisto, joka yhdistää Siemensin Simatic-ohjelmistot, kuten Step7- ja WinCC-ohjelmistot, yhdeksi ohjelmakokonaisuudeksi.
TP	Touch panel.

VBscript

Visual basic script on yksinkertainen tekstiohjelmointikieli, jota voi käyttää Siemens-paneeleissa.

WinCC

Siemens-paneelien ohjelmointityökalu.

1 JOHDANTO

1.1 Työn tausta

Tällä hetkellä Atrian sikateurastamolla käytetään urakoiden laskentaan vanhanai-kaista leimakelloa. Leimakelloon täytyy syöttää pahvinen kortti, johon leimasin lei-maa kellonajan. Urakan päätteeksi työnjohtajat laskevat kellonajat yhteen ja vähen-tävät ruokatunnit. Heidän työnsä helpottamiseksi urakoiden muistiin kirjaaminen on päätetty digitalisoida.

1.2 Työn tavoite

Tavoitteena oli luoda mahdollisimman yksinkertainen ratkaisu, joka ei tarvitse paljon huomiota eikä opastusta. Työnjohtajien tarvitsisi käydä laitteella vain 4–6 kertaa päi-vässä: aamulla urakan alkaessa, molemmilla ruokatunneilla, illalla urakan päätty-essä ja emakoiden teurastusurakan yhteydessä.

1.3 Atria Oyj

Atria Oyj on suomalainen elintarvikealan yritys, joka on vuosien aikana kansainvä- listänyt toimintaansa ja jakautuu nyt neljään eri liiketoiminta-alueeseen. 1.1.2018 alkaen Atrian liiketoiminta-alueet ovat Atria Suomi, Atria Ruotsi, Atria Venäjä ja Atria Tanska & Viro. Atria on siis yksi johtavista ruoka-alan yrityksistä Pohjoismaissa, Baltian alueella ja Venäjällä. Vuonna 2017 konsernilla oli liikevaihtoa 1,43 miljardia euroa ja sen palveluksessa työskenteli noin 4449 henkilöä. (Atria [Viitattu 14.3.2018].)

1.3.1 Atria Suomi Oy

Atria Suomi Oy on Atria Oyj:n alla toimiva tytäryhtiö, joka vastaa Suomen liiketoiminnasta. Yrityksen tehtävänä on kehittää, valmistaa ja markkinoida tuoreita elintarvikkeita ja niihin liittyviä palveluja. Atria Suomi myy tuoreita naudasta, siasta ja siipikarjasta valmistettuja elintarvikkeita. Atria on Suomen johtava teurastamoteollisuuden ja muiden lihatuoteryhmien tuottaja. (Atria [Viitattu 13.3.2018].)

2 TYÖNTUTKIMUS

Työntutkimuksella pyritään parantamaan työntekijän hyvinvointia ja yrityksen kannattavuutta. Työntutkimuksen hyödyt työntekijälle ovat parantunut työturvallisuus, ergonomia ja ansioiden kasvu. Urakkapalkkauksen pohjana työntutkimus on kattavampi kuin perinteinen työarvon mittaus. Yrityksessä työntutkimusta käytetään tavoitteiden asettamiseen, tuotannon ja resurssien suunnitteluun. Työntutkimuksessa pyritään tarkastelemaan kriittisesti kaikkia työnteon tekijöitä, jotka vaikuttavat turvallisuuteen ja tehokkuuteen. Työntutkimuksessa ei enää keskitytä niin paljoa urakkapalkkaukseen ja työnarvon määritykseen, sillä teollisuudessa tehdyistä työtunneista vain 10 % on urakkatyötä. (EK-SAK tuottavuustyöryhmä 2011, 4-5.)

Menestyäkseen yrityksen ja sen henkilöstön täytyy sopeutua muutoksiin nopeasti. Kilpailijoita nopeampi reagointi muutoksiin antaa kilpailuedun. Yrityksen on siis oltava jatkuvassa kehityksessä. Tämän kehityksen tarkoitus on parantaa yrityksen tehokkuutta ja tuottavuutta, ja sitä kautta parantaa yrityksen kilpailukykyä, kannattavuutta ja työpaikkojen säilyvyyttä. Samalla työntekijän hyvinvointia viedään eteenpäin työympäristöä, työoloja ja työtapoja kehittämällä. (EK-SAK tuottavuustyöryhmä 2011, 5.)

2.1 Työ mittauskohteena

Työmittauksella selvitetään tietyn työvaiheen suorittamiseen tai tietyn tuotteen läpimenoon kuluva aika. Ajan täytyy olla suoritusten välillä sama, riippumatta suorittajasta tai tilapäisistä olosuhteidenmuutoksista. Työn suoritus aika kuitenkin vaihtelee henkilöiden välillä, jopa saman henkilön suorittama työaika voi vaihdella suorituskertojen välillä. Tästä syystä työmittauksien tulokset täytyy normalisoida. Normalisoitu aika saadaan mitattua, kun työtehtävän suorittaa keskinkertaisen ammattitaidon omaava harjaantunut työntekijä normaaleissa olosuhteissa, normaalilla työnopeudella ja tarkkaan määritetyllä työtavalla. (EK-SAK tuottavuustyöryhmä 2011, 24.)

2.2 Ajanmääritystavat

Jokaista ajanmittausta ennen on arvioitava vaadittava tarkkuustaso ja valittava käytettävä mittausten menetelmä. Koska ajanmäärityksessä on kyse tietyn työn tekoon kuuluvan ajan selvittämisestä, ei ole olemassa yhtä ajanmääritysmenetelmää, joka sopeisi kaikkiin mahdollisiin töihin. (EK-SAK tuottavuustyöryhmä 2011, 24.)

Työnmittausmenetelmiä ovat:

- Havainnointitutkimus
- Kelloaikatutkimus
- Normaaliaikatutkimus
- Ajankäyttötutkimus
- Liikeaikatutkimus
- Aikalaskelmat
- Standardiaikajärjestelmät (EK-SAK tuottavuustyöryhmä 2011, 24.)

Havainnointitutkimus ja ajankäyttötutkimus soveltuvat kumpikin ajankäytön jakautumisen selvittämiseen, kun taas normaaliaikatutkimus ja liikeaikatutkimus sopivat paremmin normalisoitujen aikojen mittaamiseen (EK-SAK tuottavuustyöryhmä 2011, 24).

2.2.1 Havainnointitutkimus

Havainnointitutkimus on monikäyttöinen, helppo ja nopea suorittaa. Sen aikana voidaan seurata useaa työpistettä ja työvaihetta. Havainnointitutkimuksella saadaan selkeä yleiskuva osaston ajankäytöstä ja eri koneiden tehokkuudesta. Havainnointitutkimuksessa tapahtumat erotellaan moneen osaan aikalajien perusteella, kuten häiriöaikaan, apuaikaan, tauko-aikaan ja tekemisaikaan. Tutkija kirjaa määräajoin tapahtumat ylös. (EK-SAK tuottavuustyöryhmä 2011, 24.)

2.2.2 Ajankäyttötutkimus

Ajankäyttötutkimus soveltuu pitkäkestoisten ja sellaisten töiden mittaamiseen, joiden työjärjestystä ei voida ennalta määritellä, tai se määritellään työn edetessä. Ajankäyttötutkimusta voidaan käyttää myös koneiden ja niiden käyttäjien toiminnan selvittämiseen. Ajankäyttötutkimuksen tapahtumat erotellaan kuten havainnointitutkimuksessa: häiriö-, apu-, tauko- ja tekemisaikaan. (EK-SAK tuottavuustyöryhmä 2011, 25.)

2.2.3 Normaaliaikatutkimus

Normaaliaikatutkimuksessa mitataan nimenmukaisesti työn normaaliaikaa. Tämä sopii suhteellisen lyhyiden ja toistuvien käsin tehtävien töiden ajan mittaukseen. Mittattava työ jaetaan työeriin, joiden ajat mitataan. Tarvittavan suureen mittaustarkkuuteen päästään lisäämällä aikahavaintojen määrää. (EK-SAK tuottavuustyöryhmä 2011, 24-25.)

2.2.4 Liikeikatutkimus

Liikeikatutkimusta käytetään työn erittäin yksityiskohtaiseen tarkkailuun. Tässä mittaustavassa ei käytetä lainkaan kelloa, koska työ on eritelty niin pieniin osiin, että niiden suorittamiseen kuluva aika on vakio, ja vakioajat määritellään aikastandardien avulla. Tämän mittaustavan käyttö vaatii hyvin koulutettua henkilöstöä, sitä käytetään lähinnä työmenetelmien kehittämiseen. (EK-SAK tuottavuustyöryhmä 2011, 25.)

2.3 Urakka a-sikateurastamossa

Atrian sikateurastamossa urakanlasku koostuu linja-ajasta ja sikamäärästä. Näistä kahdesta suureesta lasketaan yksi kertoimista, johon teurastajien palkanmaksu perustuu. (Palomäki 2018.)

3 OHJELMOITAVAT LOGIIKAT

Ohjelmoitavat logiikat ovat elektronisia laitteita, jotka on varustettu yhdellä tai usealla mikroprosessorilla. Logiikat ohjaavat prosessin laitteita, kuten sylintereitä ja moottoreita tulojen, lähtöjen ja väylien avulla logiikan muistissa olevia ohjelmia ja parametrejä noudattaen. Logiikat voidaan liittää erikokoisiin prosesseihin erilaisten tulo-, lähtö- ja väyläyksiköiden avulla. Signaalit voivat olla binäärisiä kaksitilatietoja esimerkiksi anturi, päällä ja pois, analogisia standardiviestejä, kuten vaa'an anturilta tuleva tieto, tai pulssijonoja, joita käytetään esimerkiksi paikoitusta tarkkailevissa absoluuttiantureissa. Nykyään sarjaliikenne on yleistä eri laitteiden ja hajautetun I/O- parissa, mutta myös verkkoliikenne laitteiden välillä on yleistymässä. (Omron 2009.)

3.1 Simatic-logiikat

Simatic on Siemensin tuotemerkki, jonka tuotteista löytyy ratkaisu kaikkiin automaation tarpeisiin, räjähdysvaarallisista tiloista aina laivateollisuuteen. Simatic-tuotteisiin kuuluvat pieni ja yksinkertainen Logo!, jolla voidaan ohjata muutamaa relettä ja i/o-lähtöä. Logoa ohjelmoidaan omalla Logo!-ohjelmistolla. Logosta seuraava askel on S7-1200-sarja, joka on kooltaan pieni, mutta laskentateholtaan logoa tehokkaampi. S7-1200-sarjaa ohjelmoidaan TIA-portal-ohjelmistolla. (Siemens [Viitattu 14.3.2017].)

4 SUUNNITTELU JA TOTEUTUS

Työn suunnittelu aloitettiin yhteistyössä tuotannon henkilökunnan edustajan kanssa. Edustaja ehdotti, että laitteesta tulisi täysin automaattinen, eikä sitä tarvitsisi huomioida millään tavalla päivän aikana. Lopulta kuitenkin päädyttiin siihen tulokseen, että täysin automaattisen laitteen luominen oli vähintäänkin hankalaa, ja toisekseen laitteella olisi hyvä käydä, jotta mahdolliset virhetilanteet saataisiin tietoon välittömästi. Edustajan kanssa käytiin läpi myös, mitä urakkaan kuuluu ja mitä siitä tulee jättää pois. Urakka koostuu siis linja-ajasta, josta otetaan pois ruokatauko silloin, kun ruokatauko on 35 minuutin mittainen. Jos työpäivä on ns. suora, tällöin ruokatauko on vain 20 minuutin pituinen, ja sitä ei poisteta linja-ajasta. Emakkourakka lasketaan erikseen omana urakkanaan. Vuoronvaihto tapahtuu joka päivä samalla minuutilla, ellei päivä ole suora.

4.1 Projektin kuvaus

Projektin ajatus on periaatteena erittäin yksinkertainen. Työntekijä valitsee aamulla päivän urakan asetukset ja painaa tämän jälkeen urakan aloituspainiketta. Laskenta alkaa, kun urakkapainiketta on painettu ja kinneri, josta sika roikkuu, heilauttaa radassa kiinni olevaa heiluria, johon on kiinnitetty induktiivinen anturi. Seuraavan keran laite vaatii huomiota ruokatunnilla. Kun viimeinen sika ennen ruokataukoa on mennyt anturin ohi, voidaan painaa Ruokatauko -painiketta. Tämä ei ole kiireellistä, koska laite tallentaa viimeisimmän sian ajan automaattisesti. Riittää, että painiketta painetaan tauon aikana, jotta laite osaa merkitä viimeisimmän sian ruokatunnin alkuajaksi ja seuraavan tulevan sian ruokatauon päätösajaksi. Vuoronvaihto ei aiheuta toimenpiteitä, kunhan vuoronvaihtoaika on määritetty/todettu aamulla oikeaksi. Seuraava toimenpide on illan ruokatunti. Emakkourakan alkaessa työntekijä painaa Emakkourakka -painiketta, kun viimeinen sika ennen emakkourakkaa on ohittanut anturin. Jos emakoiden jälkeen tulee vielä lihasikaa, painetaan Jatka lihasikaurakkaa -painiketta. Jos emakkojen jälkeen ei tule enää sikaa, painetaan Päivänlopetus -painiketta. Tämä tuo esiin ponnahdusikkunan, jossa luodaan USB-tikulle kuluneesta urakasta raportti. Tämän jälkeen aukeaa toinen ikkuna, jonka kautta valitaan ja lähetetään raportti sähköpostiin. Lopuksi laite palautuu alkusivulle.

4.2 Laitteiden valinta

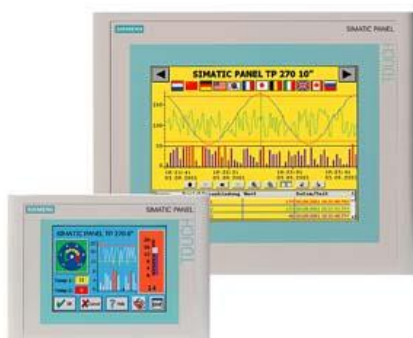
Laitevalmistajan valinta oli helppoa, koska Atrialla on käytössä laaja valikoima Siemensin logiikoita ja kosketuspaneeleita. Myös ohjelmistot ja lisenssit olivat helposti saatavilla. Valittavaksi jäi, luodaanko projekti teurastamalla jo käytössä oleville logiikoille vai asennetaanko projektille oma logiikka ja paneeli.

Sovellus olisi voitu tehdä myös PC:llä, mutta teurastamon haastavien olosuhteiden takia kustannukset olisivat nousseet liian korkeiksi. PC tarvitsee ison IP-luokitellun kaapin suojaamaan PC:n omaa koteloa, näyttöä, hiirtä ja näppäimistöä, kun taas logiikka ja kosketuspaneeli mahtuvat pieneen IP67-luokiteltuun koteloon. Koska laitetta ohjataan kosketuksella, hiiren ja näppäimistön tarvetta ei ole.

4.2.1 Kosketuspaneelin valinta

Paneelin valintaan vaikuttivat koko, hinta ja saatavuus.

Ensimmäisenä vaihtoehtona oli sikateurastamalla laajasti käytössä oleva Siemensin 10 tuumainen TP270-paneeli. Paneelin hyvänä puolena olivat valmiit ohjelmistot ja lisenssit. Isona miinuksena oli se, että TP270-paneelien valmistus ja jälleenmyynti on lopetettu 01.04.2008, niihin valmistettiin varaosia vuoteen 01.10.2017 asti. (Siemens [Viitattu 13.3.2017].) Tämä vaihtoehto hylättiin, koska varaosien loputtua laite jouduttaisiin korvaamaan joka tapauksessa uudella mallilla.



Kuva 1. Siemens MP270 -kosketuspaneeli (Siemens [Viitattu 13.3.2017].)

Seuraavana vaihtoehtona olivat KTP700 Basic- ja KTP900 Basic-näppäinkosketuspaneelyhdistelmät. KTP700- ja KTP900-mallien erona on kosketuspaneelin koko. (Siemens 2017a.) Nämä vaihtoehdot olivat hinnaltaan sopivia, mutta työssä ei tarvittu ulkoisia painikkeita, joten tämä vaihtoehto hylättiin.



Kuva 2. Siemens KTP700 -näppäinkosketuspaneeli (Siemens 2017a).

Viimeisenä vaihtoehtona olivat TP700 Comfort- ja TP900 Comfort-kosketuspaneelit. 700- ja 900-malleissa erona on kosketuspaneelin koko, virrankuutus ja kosketuspainikkeiden määrä, joilla voi ohjata suoraan S7-logiikan I/O-inputteja. (Siemens 2017c.)



Kuva 3. Siemens TP900 Comfort -kosketuspaneeli (Siemens 2017c).

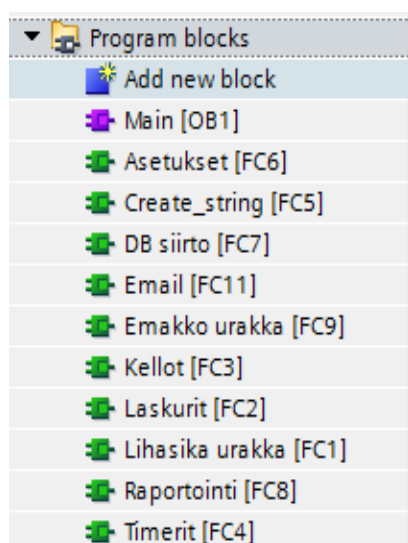
Pitkän pohdiskelun ja tutkimisen jälkeen paneeleista päädyttiin valitsemaan Siemensin 7 tuumainen TP700 Comfort -paneeli. Paneeliin päädyttiin, koska työn kannalta paneelin koko oli sopiva ja hintaero kokoa suurempaan TP900-paneeliin oli huomattavan iso. Paneeli oli myös helposti saatavilla, kyseisellä hetkellä tilauksella, mutta nykyään saatavissa suoraan Atrian omasta varastosta, joten mahdollisten laiterikkojen sattuessa korvaava laite on nopeasti saatavilla. Lisäksi kokeiltiin, pystyykö työtä toteuttamaan pelkällä paneelilla. Kävi kuitenkin ilmi, että paneeli ei pysty yksinkertaisin menetelmin laskemaan aikaa ja tallentamaan tuloksia omaan muistiinsa.

4.2.2 Logiikan valinta

Vanhojen jo käytössä olevien logiikoiden käyttö sivuutettiin heti alkuvaiheessa, koska työstä johtuva ohjelman latailu saattaisi keskeyttää tai häiritä laitoksen monimutkaista ohjelmaa. Valittavissa olivat seuraavat vaihtoehdot: S7-300-, S7-400- ja S7-1200-sarjan logiikat. Näistä päädyttiin 1200-sarjaan, tarkemmin 1211C-malliin, koska se oli laskentateholtaan, liitännöiltään ja hinnaltaan projektiin sopivin.

4.3 Logiikkaohjelma

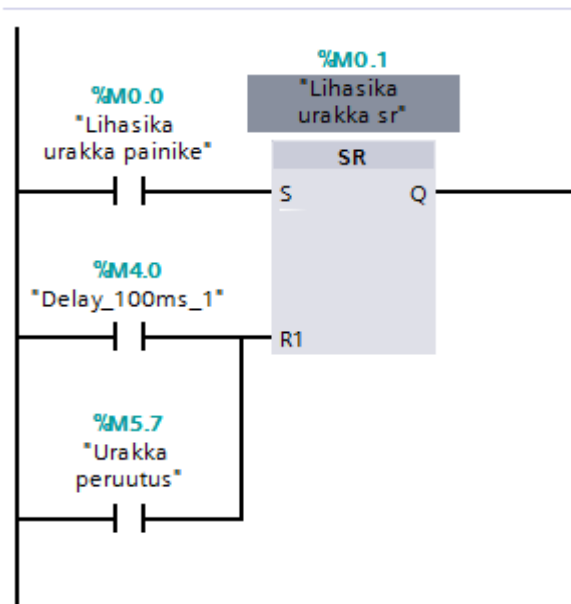
1211C-logiikka vaati ohjelmointiin vähintään TIA-portal V13 -ohjelmiston, joka on käyttäjäystävällinen ja helppo oppia erittäin laajan käyttöoppaansa ansiosta. Ohjelmointi aloitettiin luomalla projekti ja lisäämällä siihen 1211C-logiikka oikealla firmwarella. Ohjelma koostuu loppujen lopuksi kymmenestä eri osasta, jotta ohjelma pysisi siistinä ja helppolukuisena. Urakan kaikki tulokset tallennetaan DB1-muistiinpaikkaan.



Kuva 4. Ohjelmaosiot, jotka pitävät sisällään logiikkaohjelman

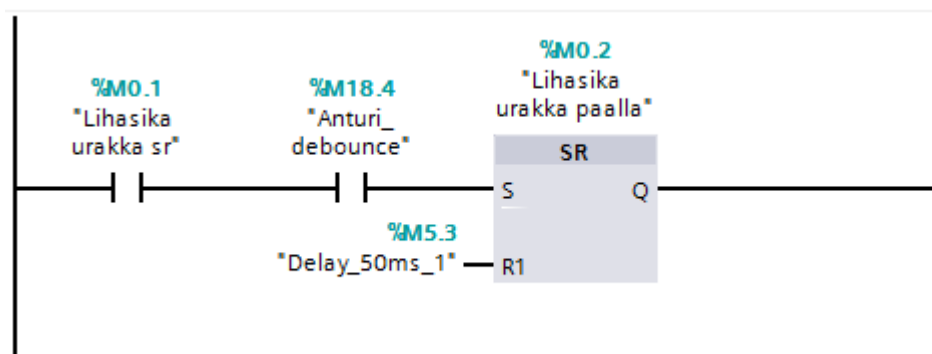
4.3.1 Lihasikaurakka

Varsinainen ohjelmointi aloitettiin Lihasikaurakan osiosta. Osio sisältää lihasikaurakan aloitusta odottavan bitin nimeltään lihasika urakka sr, joka aktivoituu, kun käyttäjä painaa paneelilta Urakan aloitus -painiketta.



Kuva 5. Lihaskaurakan aloitusbitin set/reset

Bitin tarkoitus on odottaa anturin heilautusta, joka merkitsee sitä, että ensimmäinen sika on saavuttanut anturin ja urakka voidaan aloittaa. Kun anturia on heilautettu, aloitusbitti nollaantuu ja varsinainen lihaskaurakka-bitti aktivoituu.



Kuva 6. Lihaskaurakka päällä -bitin Set/Reset

Kun tämä bitti on aktivoitunut ohjelman kello osio tallentaa ensimmäisen aktivointiajan DB1-paikan muistiin lihaskaurakan aloituskohtaan. Tämän jälkeen jokainen anturin heilautus siirtää heilautusajan muistiin DB1-paikkaan viimeisin sika kohtaan ja lisää yhden luvun lihaskaurakka laskuriin.

DB1			
Name	Data type	Start value	
Static			
Kellon aika	DTL	DTL#1970-01-014	
Viimeisin sika	DTL	DTL#1970-01-014	
Päivän lopetus	DTL	DTL#1970-01-014	
Viimeisin emakko	DTL	DTL#1970-01-014	
Lihäsika Urakan alku	DTL	DTL#1970-01-014	
Aloitus aamu ruuan jälkeen	DTL	DTL#1970-01-014	
Vuoronvaihto määrittys	DTL	DTL#1970-01-014	
Vuoronvaihto	DTL	DTL#1970-01-014	
Aloitus ilta ruuan jälkeen	DTL	DTL#1970-01-014	
Lihäsika Urakan lopetus	DTL	DTL#1970-01-014	
Emakko urakan aloitus	DTL	DTL#1970-01-014	
Emakko urakan lopetus	DTL	DTL#1970-01-014	
Aamu ruuan aloitus	DTL	DTL#1970-01-014	
Ilta ruuan aloitus	DTL	DTL#1970-01-014	
viimeisin sika ennen emakkoja	DTL	DTL#1970-01-014	
sika emakko urakan jälkeen	DTL	DTL#1970-01-014	
Alku - Loppu	Time	T#0ms	
Alku-Aamuruoka	Time	T#0ms	
Aamuruoka-Vuoronvaihto	Time	T#0ms	
Vuoronvaihto-Iltaruoka	Time	T#0ms	
Iltaruoka-Loppu	Time	T#0ms	

Kuva 7. DB1, käynnissä olevan urakan muistialue

Jokaisen heilautuksen aikaa ei tallenneta, vaan muistiin jää nimen mukaan vain viimeisin. Lisäksi tämä osio sisältää ruokataukobittit, molemmille aamu- ja iltaruokatauolle. Kun käyttäjä painaa paneelilla Ruokatauko-painiketta, aktivoituu aamu- tai iltabitti, riippuen onko vuoronvaihto aika ylitetty. Kun ruokataukobitti aktivoituu, siirtyy viimeisimmän sian aika automaattisesti muistiin ruoan aloituskohtaan. Ruokataukobitti nollaantuu, kun anturia seuraavan kerran heilautetaan, tällöin heilautusajankohta siirtyy muistiin ruokatauon lopetuskohtaan. Tällä tavalla ruokatauko voidaan leimata milloin vain ruokatauon aikana. Käyttäjän tarvitsee käydä laitteella vain kerran ruokatauon aikana.

4.3.2 Emakkourakka

Tämä osio sisältää emakkourakkabittin. Bitti aktivoituu, kun käyttäjä painaa paneelilta Emakkourakan aloitus -painiketta. Tällöin ohjelma tallentaa viimeisimmän sian muistiin, viimeisin sika ennen emakkourakkaa -kohtaan. Bitti nollaantuu, kun anturia heilautetaan, emakkourakkabitti aktivoituu ja heilautusajankohta siirtyy muistiin emakkourakan aloitus -kohtaan. Jokainen heilautus siirtää ajan muistiin viimeisin emakko -kohtaan ja lisää yhden luvun emakkolaskuriin. Emakkourakan voi lopettaa

painamalla päivänlopetus-painiketta, tai painamalla jatka urakkaa painiketta, joka jatkaa lihasikaurakkaa.

4.3.3 Kellot

Kellot-osiossa ovat kaikki aikaa laskevat ja tallentavat logiikkakäskyt. Tämä on suurin osio koko ohjelmassa. Kellonaika kirjoitetaan tämän osan ensimmäisessä virtapiirissä. Kirjoitus tapahtuu rd_loc_t-käskyllä, joka lukee logiikan sisäistä kelloa. Tämä kirjoitetaan jokaisella ohjelmakierrolla DB1-paikan kellonaikamuistipaikkaan. Kellonajat siirretään DB1-paikan kellonaikakohdasta move-käskyllä tarvittavaan muistipaikkaan, kun tietyt kriteerit täyttyvät. Esimerkiksi kun lihasikaurakkabitti aktivoituu, niin silloin kellonaika siirretään lihasikaurakan alkumuistipaikkaan. Täällä tapahtuu myös väliaikojen laskeminen t_diff-käskyllä, johon syötetään kaksi kellonaikaa, ja ulos tulee näiden kahden ajan väliaika.

4.3.4 Asetukset

Asetukset-osioista löytyvät move-käskyt, joilla siirretään vuoronvaihtotunti ja -minuutti, logiikkaohjelmaan. Työntekijä määrittää vuoronvaihtotunnin ja -minuutin paneelin asetukset välilehdeltä. Vuoro vaihtuu, kun asetettu tunti ja minuutti osuvat kohdalle, tällöin vuoronvaihtobitti aktivoituu ja iltavuoro alkaa.

4.3.5 Laskurit

Laskurit-osio pitää sisällään kaikki count_up-laskurit, jotka lisäävät yhden luvun sen hetken urakkaan aina, kun anturi heilahtaa. Täällä hoidetaan myös laskujen erotelu. Aamu- ja iltavuoron määrät lasketaan seuraavasti: aamun määrä tallennetaan vuoronvaihdossa DB1-paikan aamuvuoron sikamääräkohtaan, ja illalla tämä määrä erotetaan kokonaismäärästä, jolloin saadaan iltavuoron sikamäärä.

4.3.6 Timerit

Timerit-osiosta löytyvät kaikki veto- ja päästöhidasteiset ajastimet, joita tarvitaan ohjelman monissa eri osissa varmistamaan, että tietty toiminto ehtii tapahtua. Esimerkiksi anturin uudelleenaktivointia estävä päästöhidasteinen ajastin pitää huolen siitä, että ohjelma ei laske sikoja, jos heiluri heilahtaa voimakkaasti ja hyppää uudelleen saman sian aikana anturin tunnistusalueelle.

4.3.7 Create_string

Create_String-osio luo paneelin muistitikulta puretun CSV-tiedoston 116:sta solusta, muistiin tallennetuista merkkijonoista, yhden pitkän merkkijonon. Tämä tapahtuu concat-käskyllä, joka yhdistää kaksi merkkijonoa yhdeksi. Concat-käskyjä on siis 116, joilla muutetaan Wstring ryhmä yhdeksi Wstringiksi.

4.3.8 Datablockien välinen tiedonsiirto

Aina päivän päätteeksi tapahtuu tiedonsiirto DB muistialueiden välillä. Projektissa on seitsemän urakan tietoa sisältävää DB:tä. DB1-paikka pitää sisällään muistipaikat, joihin tallennetaan kaikki kellonajoista sikamääriin, urakan ollessa käynnissä. DB1-paikka pitää sisällään paljon tietoa, jota ei tarvitse arkistoida, vain oleelliset tiedot, kuten väliajat, aloitus- ja lopetusajat, siirretään. Loput kuusi paikkaa, DB10—DB15, ovat arkistoja, joihin data siirretään takaamaan virhetapauksia. DB15 on viimeinen arkisto, joka kirjoitetaan yli DB14-paikan tiedoilla. DB14 kirjoitetaan yli DB13 jne. Lopuksi DB1-paikka tyhjennetään kirjoittamalla kaikki tieto nolaksi.

4.3.9 Email

Ohjelman Email-osio pitää sisällään 1200-logiikan TMAIL_C-sähköpostikäskyn, jolla sähköpostin lähetys onnistuu helposti. Kyseinen käsky vaatii parametrinta varten instancedatablockin, johon syötetään smtp palvelimen IP-osoite, lähettäjän

sähköpostiosoite ja salasana, jos smtp palvelin vaatii lähettäjän todennuksen. Tämä osio pitää sisällään myös liitteen luontiin käytettävän askelohjelman.

4.4 CSV-tiedoston luominen paneelilla

CSV-tiedoston luominen paneelilla osoittautui mahdolliseksi kahdella tavalla. Oli mahdollista listata hälytykset taulukkoon tai kirjoittaa data VBscriptillä logiikan datablockista CSV-tiedostoon.

4.4.1 CSV-tiedosto hälytyslistalla

TIA-portal V13 -ohjelmistolla on mahdollista luoda Excel-taulukko hälytyslistaa hyväksi käyttäen. Tällä metodilla olisi saatu vain tapahtuma-ajat ylös. Tässä sovelluksessa on kuitenkin oleellista, että myös tapahtumien väliajat saadaan laskettua, joten tämä tapa ei ollut mahdollinen.

4.4.2 CSV-tiedosto VBscriptillä

Siemensin paneelit tukevat yksinkertaista VBscript-ohjelmointikieltä. VBscriptillä on mahdollista luoda tekstitiedostoja, mikä mahdollistaa CSV-tiedoston luomisen. CSV-tiedoston luominen tapahtuu valitsemalla haluttu tiedostopolku, ja luomalla sinne halutun niminen tiedosto. Tässä tapauksessa tiedoston nimen määrittää urakan päivämäärä ja polkuna toimii paneeliin kiinnitetty USB-muistitikku. Koodi myös tarkastaa, löytyykö polusta samannimistä tiedostoa, jos löytyy, se poistetaan. Tämän jälkeen alkaa varsinainen tekstin kirjoitus tiedostoon. Kuvassa 8 näkyvä ensimmäinen lineprint-komento aloittaa kirjoituksen riville yksi, toinen riville kaksi jne. Delimiter on puolipiste, jonka avulla Excel tietää, että teksti siirtyy seuraavaan soluun. Delimiteriin on myös lisätty \r merkitsemään puolipisteen paikkaa myöhemmässä vaiheessa. Newline on merkki, joka ilmoittaa rivinvaihdosta, tätä käytetään myös

sovelluksen myöhemmässä vaiheessa. SmartTag tuo tietoa suoraan logiikan DB-muistista.

```

8 Dim fo
9 Dim path
10 Dim mode
11 Dim delimiter
12 Dim gap
13 Dim Name
14 Dim FilePath
15 Dim objExcel
16 Dim DB
17 Const NEU=0
18 Dim newline
19 FilePath = "\\Storage Card USB\"
20 Name = "Urakan pvm " & SmartTags("DB10_Päivämäärä")
21 mode=2
22
23 path = FilePath & Name & ".csv"
24 delimiter=";"
25 gap=""
26 newline = "!"
27 On Error Resume Next
28
29 Set fo= CreateObject("FileCtl.File")
30
31 fo.open path,mode
32
33 If Err.Number<> 0 Then
34     ShowSystemAlarm "Error # " & CStr(Err.Number)& " " & Err.Description
35     Err.Clear
36     Exit Sub
37 End If
38
39 If fo.fileExists(path) Then
40     fo.deletefile path
41 End If
42
43 fo.LinePrint("Loki Luotu" & gap & delimiter & gap & Now & delimiter & newline)
44 fo.LinePrint(gap & delimiter & newline)
45 fo.LinePrint(gap & delimiter & "Kellon ajat" & delimiter & delimiter & "Ajat d
46 fo.LinePrint("Lihāsika urakan alku" & delimiter & SmartTags("DB10_Lihāsika Ura
47 fo.LinePrint("Aamu ruuan aloitus" & delimiter & SmartTags("DB10_Aamu ruuan alo
48 fo.LinePrint("Vuoronvaihto" & delimiter & SmartTags("DB10_Vuoronvaihto") & del
49 fo.LinePrint("Ilta ruuan aloitus" & delimiter & SmartTags("DB10_Ilta ruuan alo
50 fo.LinePrint("Lihāsika Urakan lopetus" & delimiter & SmartTags("DB10_Lihāsika
51 fo.LinePrint("Emakko urakan aloitus" & delimiter & SmartTags("DB10_Emakko urak
52 fo.LinePrint("Emakkourakan lopetus" & delimiter & SmartTags("DB10_Emakko uraka

```

Kuva 8. Tietojen tallentaminen CSV-tiedostoon

4.5 Sähköpostin lähettäminen logiikalla ja paneelilla

Sähköpostin lähettämisessä ei sinänsä ole mitään ihmeellistä, koska TIA-portal V13 sisältää kaikki oleelliset asetukset ja käskyt pelkän sähköpostin lähettämistä varten. Ongelmana on sähköpostin lähettäminen liitteen kanssa.

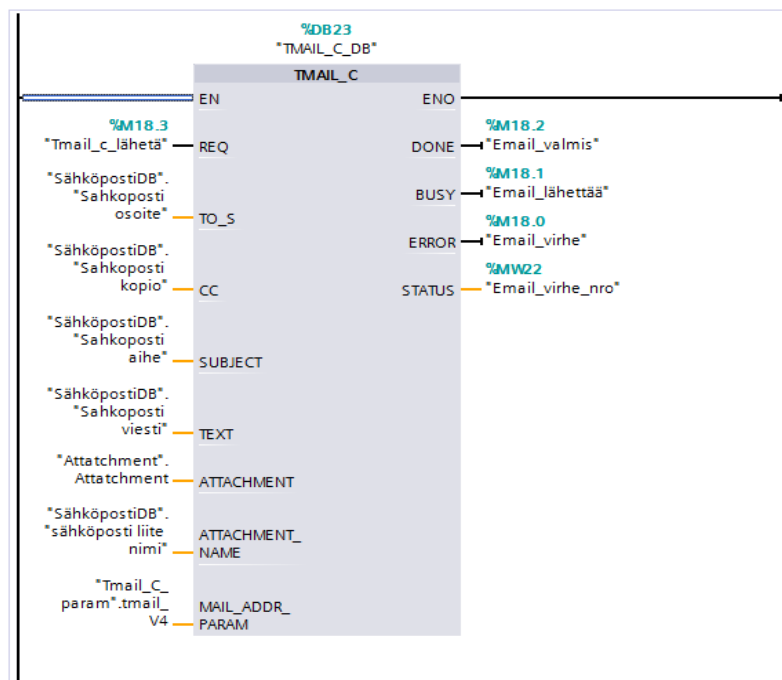
4.5.1 Sähköpostin lähettäminen paneelilla

TIA-portalissa sähköpostin lähettäminen on tehty erittäin yksinkertaiseksi. Ensin täytyy asettaa oikeat smtp-palvelinasetukset. Tämän jälkeen projektin sivulle lisätään painike, jonka Event-asetuksiin lisätään Click-välilehdelle Send email -käsky. Käskyn asetetaan lähettäjän ja vastaanottajan osoitteet, otsikko ja varsinainen teksti. Aina kun painiketta klikataan, lähettää se ennalta määritellyn viestin valittuihin osoitteisiin. Tällä tavalla ei voi lähettää liitteitä. Seuraava vaihtoehto oli käyttää scriptiä, joka käyttää koodia nimeltä CDO message. CDO message on kätevä koodinpätkä, jolla voidaan lähettää sähköpostiviestejä liitteiden kanssa. Koodi saatiin toimimaan moitteetta simulointiympäristössä, mutta kun projekti saatiin siirrettyä paneeliin ja paneeli vihdoinkin verkkoon, sähköposti ei lähtenyt. Syynä tähän oli CDO message -koodin käyttämä kirjastotiedosto, jota ei löytynyt paneelin Windows CE 6.0 -käyttöjärjestelmästä. Kirjaston lisääminen paneeliin osoittautui mahdottomaksi. CDO messagen-koodi todettiin hyödyttömäksi.

4.5.2 Sähköpostin lähettäminen logiikalla

Viimeinen vaihtoehto oli TIA-portal V13-ohjelmiston sisältämä TMAIL_C-komento. Tällä komennolla onnistui niin sähköpostin kuin liitteenkin lähettäminen. Liitteen lähettäminen oli kuitenkin hankalaa, koska liite täytyi olla byte-, word- tai doubleword-

muodossa. Tämä tarkoittaa sitä, että liitteen jokaisen kirjaimen tulee olla ASCII-koodia. Liite täytyy siis muuttaa muotoon Array of char.



Kuva 9. Tmail_C-komento, joka lähettää sähköpostin

Liitettä ei siis voi suoraan osoittaa muistitikulta, kuten CDO messagella, vaan se pitää ensiksi purkaa osiin ja koota uudelleen. Tässä apuun astuu jälleen paneelin VBScript. Verkosta löytyi koodi, jolla avataan Excel-taulukko ja tallennetaan jokai-

sen solun sisältämä informaatio logiikassa sijaitsevaan datablockiin. Aluksi solut tallennetaan koodin avulla, Array of Wstring -muodossa, jokainen solu erikseen omaan lokeroonsa.

```

16 mode=1
17 path = SmartTags("Sahkoposti liitepolku")
18 'path = "C:\storage card\test.csv"
19 On Error Resume Next
20
21 Set fo= CreateObject("filect1.file")
22
23 SmartTags("lahettaa") = 1
24 fo.open path, mode
25 SmartTags("lahettaa") = 2
26
27 If Err.Number<> 0 Then
28     SmartTags("ErrorNr") = "Error # " & Hex(Err.Number)
29     Err.Clear
30     Exit Sub
31 End If
32
33 SmartTags("lahettaa") = 3
34 Do While fo.EOF = False
35
36 SmartTags("lahettaa") = 4
37 MyZf = fo.LineInputString
38 field = Split (MyZf, "x")
39
40     For i=0 To 3
41         field(i)= Replace (field(i), " ", "")
42         HiField(j,i) = field(i)
43     Next
44
45 SmartTags("Sahkoposti liite[1]") = HiField(0,0)
46 SmartTags("Sahkoposti liite[2]") = HiField(0,1)
47 SmartTags("Sahkoposti liite[3]") = HiField(0,2)
48 SmartTags("Sahkoposti liite[4]") = HiField(0,3)
49 SmartTags("Sahkoposti liite[5]") = HiField(1,0)
50 SmartTags("Sahkoposti liite[6]") = HiField(1,1)
51 SmartTags("Sahkoposti liite[7]") = HiField(1,2)
52 SmartTags("Sahkoposti liite[8]") = HiField(1,3)
53 SmartTags("Sahkoposti liite[9]") = HiField(2,0)
54 SmartTags("Sahkoposti liite[10]") = HiField(2,1)
55 SmartTags("Sahkoposti liite[11]") = HiField(2,2)
56 SmartTags("Sahkoposti liite[12]") = HiField(2,3)
57 SmartTags("Sahkoposti liite[13]") = HiField(3,0)
58 SmartTags("Sahkoposti liite[14]") = HiField(3,1)
59 SmartTags("Sahkoposti liite[15]") = HiField(3,2)

```

Kuva 10. Excel-taulukon purku-script

Koodissa etsitään merkkiä x, joka kuvaa solunvaihtoa. Tästä merkistä koodi tietää olevansa solun lopussa ja siirtyy seuraavaan. Normaalisti merkin tilalla käytetään puolipistettä. Koska etsittävä merkki ei tallennu tässä prosessissa, täytyy puolipisteen tilalla käyttää jotain muuta merkkiä, sillä puolipistettä tarvitaan CSV-tiedoston solun vaihtoon. Tästä syystä tekstiin on lisätty merkki x, joka katoaa koodin toteutuksessa. Tässä vaiheessa tallennetusta informaatiosta katoaa myös rivinvaihdot. Tästä syystä raporttiin on lisätty !-merkki kuvaamaan rivinvaihtoa. Tämä kaikki ta-

pahtuu siinä vaiheessa, kun käyttäjä on valinnut paneelin muistitikulta CSV-tiedoston ja painanut Lähetä-painiketta. Ruudulle avautuu ikkuna, joka kertoo liitteen luonnin olevan käynnissä. Kun luonti on valmis, on CSV-tiedosto purettu Array of Wstringiin. Tämän jälkeen käyttäjä voi painaa ikkunassa olevaa Lähetä-painiketta, jolloin alkaa Array of Wstringin muuttaminen ASCII-koodiksi.

	Name	Data type	Start value	Monitor value
2	liiteWstringArray	Array[0..116] of WString[35]		
3	liiteWstringArray[0]	WString[35]	WSTRING#''	WSTRING#''
4	liiteWstringArray[1]	WString[35]	WSTRING#''	WSTRING#'LokiLuotu'
5	liiteWstringArray[2]	WString[35]	WSTRING#''	WSTRING#';31.08.201721:09:13'
6	liiteWstringArray[3]	WString[35]	WSTRING#''	WSTRING#';!
7	liiteWstringArray[4]	WString[35]	WSTRING#''	WSTRING#''
8	liiteWstringArray[5]	WString[35]	WSTRING#''	WSTRING#''
9	liiteWstringArray[6]	WString[35]	WSTRING#''	WSTRING#';!
10	liiteWstringArray[7]	WString[35]	WSTRING#''	WSTRING#''
11	liiteWstringArray[8]	WString[35]	WSTRING#''	WSTRING#''
12	liiteWstringArray[9]	WString[35]	WSTRING#''	WSTRING#''
13	liiteWstringArray[10]	WString[35]	WSTRING#''	WSTRING#';Kellonajat'
14	liiteWstringArray[11]	WString[35]	WSTRING#''	WSTRING#';!
15	liiteWstringArray[12]	WString[35]	WSTRING#''	WSTRING#';Ajatdesimaaleina!'
16	liiteWstringArray[13]	WString[35]	WSTRING#''	WSTRING#';Lihaisikaurakanalku'
17	liiteWstringArray[14]	WString[35]	WSTRING#''	WSTRING#';06:23:01'
18	liiteWstringArray[15]	WString[35]	WSTRING#''	WSTRING#';!
19	liiteWstringArray[16]	WString[35]	WSTRING#''	WSTRING#';=(B4-KOKONAISLUKU(B4))*24!'
20	liiteWstringArray[17]	WString[35]	WSTRING#''	WSTRING#';Aamuruuanaloytus'
21	liiteWstringArray[18]	WString[35]	WSTRING#''	WSTRING#';10:27:05'
22	liiteWstringArray[19]	WString[35]	WSTRING#''	WSTRING#';!
23	liiteWstringArray[20]	WString[35]	WSTRING#''	WSTRING#';=(B5-KOKONAISLUKU(B5))*24!'
24	liiteWstringArray[21]	WString[35]	WSTRING#''	WSTRING#';Vuoronvaihto'
25	liiteWstringArray[22]	WString[35]	WSTRING#''	WSTRING#';14:31:00'

Kuva 11. Array of WString DB, joka sisältää CSV-tiedostosta puretun informaation

Kun Lähetä-painiketta on painettu, ohjelma aloittaa liitteen muokkaamisen lopulliseen muotoon. Aluksi aktivoituu SCL-koodi, joka käy for-lauseella läpi array of Wstringin jokaisen osan ja etsii niistä päivämäärää 01.01. Kun kyseinen päivämäärä löytyy, koodi poistaa Wstringin alusta 10 merkkiä, jolloin aikalukuja sisältävistä osista poistuu vuosiluku ja päivämäärä.

```

1 FOR #loop := 0 TO 116 DO
2     // Statement section FOR
3     #return_find := FIND(IN1 := "WstringArray".liiteWStringArray[#loop], IN2 := #Date);
4     IF #return_find > 0 THEN
5         "WstringArray".liiteWStringArray[#loop] := DELETE(IN := "WstringArray".liiteWStringArray[#loop], L := 10, P := 2);
6
7     END_IF;
8
9 END_FOR;

```

Kuva 12. Päivämäärän poiston SCL-koodi

Kun poistokoodi on käynyt koko array of WStringin läpi, alkaa seuraava askel. Toisessa askeleessa luodaan liitteelle nimi. Tämä tapahtuu SCL-koodilla, joka ottaa liitteen luontipäivämäärän ja lisää siihen tekstin "teurastamon urakka".

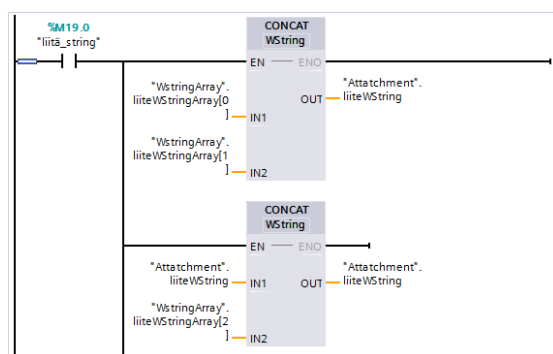
```

1 "Attachment".Liite_nimi[0] := "WstringArray".liiteWStringArray[2];
2 "Attachment".Liite_nimi[1] := MID(IN := "Attachment".Liite_nimi[0], L := 10, P := 2);
3 "Attachment".Liite_nimi[2] := CONCAT(IN1 := #liite4, IN2 := "Attachment".Liite_nimi[1]);
4 "Attachment".Liite_nimi[3] := CONCAT(IN1:= "Attachment".Liite_nimi[2], IN2:=#liite2);
5 "SähköpostiDB"."sähköposti liite nimi" := WSTRING_TO_STRING("Attachment".Liite_nimi[3]);

```

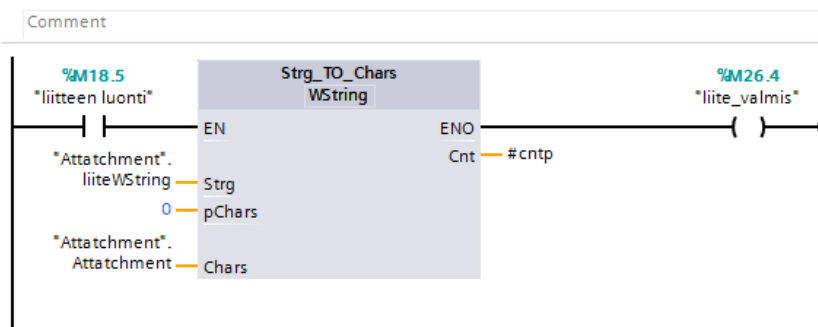
Kuva 13. Liitteen nimen luontikoodi

Kun liitteen nimi on luotu, Array of WStringistä luotiin concat-komennoilla yksi pitkä Wstring.



Kuva 14. Concat-komento, joka yhdistää kaksi stringiä yhdeksi

Seuraavaksi WString muutettiin character-muotoon Strg_TO_Chars-komennolla. Nyt liite on Array of Char -muodossa, kaikki 4060 merkkiä erikseen omissa muisti-paikoissaan.



Kuva 15. Strg_TO_Chars-komento, joka muuntaa stringin char-muotoon

Seuraaksi arraysta etsittään huutomerkki lyhyellä STL-koodilla, joka käy läpi koko Array of Charin merkki kerrallaan. Arraysta koodi etsii huutomerkkin heksalukua, joka on Word-muodossa "16#0021", vaihtaen sen rivinvaihdon heksalukuun "16#000A".

```

1 FOR #loop := 0 TO 4060 DO
2   IF "Attachment".Attachment[#loop] = 16#0021 THEN
3     UMOVE_BLK(IN := "Attachment".Linefeed[1],
4               COUNT := 1,
5               OUT => "Attachment".Attachment[#loop]);
6   END_IF;
7 END_FOR;
8

```

Kuva 16. STL-koodi, joka etsii liitteestä huutomerkkin ASCII-koodia ja muuttaa sen rivinvaihdoksi

Kun liitteestä on poistettu ylimääräiset päivämäärät, se on muutettu stringistä characteriksi ja siihen on lisätty rivinvaihdot, antaa askelohjelma tmail_c-komennolle käskyn sähköpostin lähettämiseksi. Tmail_c-komento lähettää sähköpostin ennalta määritettyihin sähköpostiosoitteisiin, Tmail_C_Param-instancedatablokissa määritetyn smtp-palvelimen kautta. Tmail_C_param on TIA-portalin automaattisesti luoma datablock, johon määritellään sähköpostiasetukset, kuten lähettäjän osoite,

Tmail_C_param (snapshot created: 5/9/2017 8:24:03 PM)								
	Name	Data type	Start value	Retain	Accessible f...	Visible in ...	Setpoint	Comment
1	Static							
2	▼ tmail_V4	TMail_V4			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
3	■ Interfaceld	HW_ANY	64		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		HW-identifier of IE-interface submodule
4	■ ID	CONN_OUC	16#0001		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		connection reference / identifier
5	■ ConnectionType	Byte	16#20		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		type of connction: 32= TMail_v4
6	■ ActiveEstablished	Bool	false		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		active/passive connection establishment
7	■ CertIndex	Byte	16#0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Index of SSA certificate on CIMCP
8	■ WatchDogTime	Time	T#1m		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		watchdog time to monitor SMTP server associ...
9	■ MailServerAddress	IP_V4			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IP address (IPv4) of mail server
10	■ ADDR	Array[1..4] of Byte			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
11	■ ADDR[1]	Byte	255		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
12	■ ADDR[2]	Byte	255		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
13	■ ADDR[3]	Byte	255		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
14	■ ADDR[4]	Byte	0		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		IPv4 address
15	■ UserName	String[254]	"		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		name of user
16	■ PassWord	String[254]	"		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		user password
17	■ From	EMAIL_ADDR			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		source mail address
18	■ LocalPartPlusAt...	String[64]	'email@'		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		local part of e-mail address plus "@" sign
19	■ FullQualifiedD...	String[254]	'osoite.com'		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		full qualified domain name part of e-mail add...

smtp-palvelimen IP-osoite, ja mahdollista todennusta varten salasana ja käyttäjätunnus. Tässä projektissa käytetään smtp-palvelinta, joka käyttää sähköpostiviestin todennukseen lähettäjän IP-osoitetta.

Kuva 17. Tmail_C_param datablock, johon määritellään sähköpostin asetukset

Kun sähköposti ja liite on lähetetty onnistuneesti, tyhjennetään liitettä sisällä pitävä array of char kirjoittamalla jokainen kirjain 16#0000, jolloin seuraava lähetettävä liite voidaan kirjoittaa tyhjälle alustalle. Sähköpostin liitteeseen tulostuu tämän työn liitteessä 1 olevan malliraportin kaltainen raportti.

5 YHTEENVETO JA POHDINTA

Projekti oli kokonaisuudessaan haastava. Työn haasteista suurin oli sähköpostin liitteen lähettäminen. Tämä osoittautui erittäin monimutkaiseksi ja vaikeaksi, koska ohjeita ei löytynyt yhdestä lähteestä, vaan ohjeita piti etsiä useista lähteistä, joista mikään ei vastannut suoraan siihen, mitä projektissa oltiin tekemässä.

Työn tekijä oppi työssä, kuinka etsitään ja luetaan koodia, ja kuinka sitä suodatetaan niin, että vain oleellinen jää jäljelle. Työtä vei eteenpäin, vaikeuksista huolimatta, tekijän päättymätön halu onnistua.

Projekti on ollut valmistuttuaan päivittäisessä käytössä Nurmon sikateurastamossa. Käyttöön otettaessa koodissa ilmeni muutamia virheitä, mutta ne korjattiin välittömästi. Laitteen käyttäjät ovat tyytyväisiä ja ovat antaneet ehdotuksia laitteen parantamiseksi, kuten esimerkiksi paneelilla näkyvät visuaaliset palautteet siitä, että sähköpostin lähettäminen on käynnissä ja sähköpostin lähetys on onnistunut. Työ tehtiin päivätyön ohessa.

LÄHTEET

- Atria. Ei päiväystä. Atria Suomi. [Verkkosivu]. Atria Oyj. [Viitattu 13.3.2018]. Saatavissa: <https://www.atria.fi/konserni/yritys/kansainvalisyys/atria-suomi/>
- Atria. Ei päiväystä. Atria Konserni. [Verkkosivu]. Atria Oyj. [Viitattu 14.3.2018]. Saatavissa: <https://www.atria.fi/konserni/yritys/kansainvalisyys/>
- EK-SAK tuottavuustyöryhmä. 2011. Työntutkimuksen käsitteitä, menettelytapoja ja käyttökohteita. [Verkkojulkaisu]. Teknoliogiateollisuus [Viitattu 11.5.2018.] Saatavissa: https://teknoliogiateollisuus.fi/sites/default/files/file_attachments/tyomarkkinat_kannustava_palkkaus_palkkaustapoja_tyontutkimuksen_menettelytavat.pdf
- Omron. 2009. Cx-One ja logiikkaohjelmointi 2009_2. Yleistä ohjelmoitavista logiikoista (1), 7-8. [Verkkomanuaali]. Omron Suomi. [Viitattu 15.3.2017]. Saatavissa: https://www.myomron.com/downloads/9.Local%20Material/Finnish/CX-One%20ja%20logiikkaohjelmointi%202009_2.pdf
- Palomäki, J. 2018. Työnjohtaja. A-sikateurastamo Oy. Haastattelu 11.5.2018
- Siemens. 2017a. Simatic HMI KTP700 Basic. [Verkkosivu]. Siemens AG. [Viitattu 23.1.2017]. Saatavissa: <https://support.industry.siemens.com/cs/pd/302298?pdtd=td&lc=en-US>
- Siemens. 2017b. Simatic HMI KTP900 Basic. [Verkkosivu]. Siemens AG. [Viitattu 23.1.2017]. Saatavissa: <https://support.industry.siemens.com/cs/pd/136172?pdtd=td&lc=en-US>
- Siemens. 2017c. Simatic HMI TP900 Comfort. [Verkkosivu]. Siemens AG. [Viitattu 24.1.2017]. Saatavissa: <https://support.industry.siemens.com/cs/pd/70559?pdtd=td&lc=en-WW>
- Siemens. 2017d. Simatic HMI TP700 Comfort. [Verkkosivu]. Siemens AG. [Viitattu 24.1.2017]. Saatavissa: <https://support.industry.siemens.com/cs/pd/127118?pdtd=td&lc=en-WW>
- Siemens. Ei päiväystä. Ohjelmoitavat logiikat. [Verkkosivu]. Siemens AG. [Viitattu 14.3.2017]. Saatavissa: http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/automaatiotekniikka/ohjelmoitavat_logiikat_simatic.php
- Siemens. Ei päiväystä. TP270 10 STN. [Verkkosivu]. Siemens AG. [Viitattu 13.3.2017]. Saatavissa: <https://support.industry.siemens.com/cs/pd/407584?pdtd=pi&dl=en&lc=en-PE>

LIITTEET

LIITE 1. Malliraportti

LIITE 1 Malliraportti

Malliraportti 27.4.2018
Loki Luotu 11:09

	Kellonajat	Ajat desimaaleina
Lihasikaurakan alku	7:00:00	7
Aamu ruuan aloitus	10:25:00	10,41667
Aamu ruuan lopetus	11:00:00	11
Vuoronvaihto	15:00:00	15
Ilta ruuan aloitus	18:00:00	18
Ilta ruuan lopetus	18:35:00	18,58333
Lihasikaurakan lopetus	21:00:00	21
Emakkourakan aloitus	21:15:00	21,25
Emakkourakan lopetus	21:45:00	21,75
	Väliajat ja kestot	
Alku-Aamuruoka	3:25:00	3,416667
Aamuruoka-Vuoronvaihto	4:00:00	4
Vuoronvaihto-Ilta ruoka	3:00:00	3
Ilta ruoka-Loppu	2:25:00	2,416667
Aamu ruokatauon kesto	0:35:00	0,583333
Ilta ruokatauon kesto	0:35:00	0,583333
Aamuvuoron kokonaiskesto	7:25:00	7,416667
Ilta vuoron kokonaiskesto	5:25:00	5,416667
Lihasikaurakan kesto	12:50:00	12,833333
Aamu emakkourakan kesto		0
Ilta emakkourakan kesto	0:30:00	0,5
	Sika määrät	
Aamun lihasikamäärä	xxxx	
Illan lihasikamäärä	xxxx	
Kokonaislihasikamäärä	xxxx	
Emakkomäärä	xxxx	