

Pontus Pyhtinen

**VOC-POLTTOLAITOKSEN OHJAUSJÄRJESTELMÄN DATANKE-  
RUUN KEHITTÄMINEN**

# **VOC-POLTTOLAITOKSEN OHJAUSJÄRJESTELMÄN DATANKE- RUUN KEHITTÄMINEN**

Pontus Pyhtinen  
Opinnäytetyö  
Kevät 2018  
Sähkö- ja automaatiotekniikan tutkinto-  
ohjelma  
Oulun ammattikorkeakoulu

# TIIVISTELMÄ

Oulun ammattikorkeakoulu  
Sähkö- ja automaatiotekniikan koulutusohjelma, automaatiotekniikka

---

Tekijä(t): Pontus Pyhtinen  
Opinnäytetyön nimi: VOC-polttolaitoksen datankeruun kehittäminen  
Työn ohjaaja(t): Timo Heikkinen (Oamk), Aki Ronkainen (Ambit Oy)  
Työn valmistumislukukausi ja -vuosi: kevät 2018  
Sivumäärä: 33 + 2 liitettä

---

Tämän opinnäytetyön aiheena oli ohjelmoida tiedonkeruusovellus Siemensin S7-1200-logiikalle. Tavoitteena oli luoda dynaaminen tiedonkeruusovellus VOC-polttolaitoksen ohjausjärjestelmään. Työhön kuului myös ohjausjärjestelmän käyttöliittymään tehtävä tiedon graafinen esittäminen, mikä toteutettiin Siemensin TP1200 Comfort -paneelin avulla. Teoriaosuudessa käsitellään VOC-polttolaitosta, Siemensin TIA Portalia ja tiedonkeruuta.

Työn suorittaminen alkoi tutkimalla S7-1200-logiikoiden tiedonkeruun mahdollisuuksia. Logiikan ohjelman tekemiseen käytettiin Siemensin TIA Portalia, jolla tehtiin tiedonkeruusovellus logiikalle sekä tutkittiin tallennetun tiedon esittämistä graafisessa muodossa Siemensin TP1200 Comfort -paneelilla.

Opinnäytetyön tuloksena syntyi dynaaminen tiedonkeruusovellus, josta on mahdollista muuttaa erilaisia parametreja, kuten tiedon tallennusväliä ja säilytysaikaa. Graafia ei saatu näkymään tallennetuista mittaustiedoista, vaan jouduttiin tyytymään reaaliaikaisten mittausarvojen esittämiseen. Graafi olisi todennäköisesti saatu näkymään halutulla tavalla VBScriptillä ohjelmoimalla, mutta aikataulun takia se jouduttiin hylkäämään.

---

Asiasanat: ohjelmoitava logiikka, Siemens, TIA Portal, tiedonkeruu, VOC, VOC-polttolaitos

## ABSTRACT

Oulu University of Applied Sciences  
Electrical and automation engineering, Automation engineering

---

Author(s): Pontus Pyhtinen  
Title of thesis: Development of Data Logging for VOC Incinerator  
Supervisor(s): Timo Heikkinen, Aki Ronkainen  
Term and year when the thesis was submitted: spring 2018  
Pages: 33 + 2 appendices

---

The purpose of this thesis was to produce an automated data saving program using Siemens S7-1200 PLC. Part of the job was to present the saved data as a graph using Siemens TP1200 Comfort -panel. The customer was Genano Solutions, which ordered the data logging from Ambit Oy.

The automated data logging was supposed to be dynamic, meaning that several logging parameters could be changed according to situation. Work began with exploring data logging possibilities in Siemens TIA Portal -program using its included manuals and different Internet sources.

The result was a dynamic data logging application which can be modified according to required situation. The graphical presentation of saved data could not be made as had been intended mainly because of the restrictions of TIA Portal's trend maker tool. Had there been more time it may have been possible to do using VBScripts.

---

Keywords: programmable logic, Siemens, TIA Portal, data logging, VOC, VOC-incinerator

# SISÄLLYS

TIIVISTELMÄ	3
ABSTRACT	4
SISÄLLYS	5
SANASTO	6
1 JOHDANTO	7
1.1 Ambit Oy	7
1.2 Työn tavoitteet	8
2 VOC-POLTTOLAITOS	9
3 SIEMENS TIA PORTAL	10
4 TIEDONKERUU	11
5 TYÖN TOTEUTUS	12
5.1 Työn aloitus	12
5.2 Tiedonkeruusovelluksen vaatimukset	15
5.3 Logiikkasovelluksen suunnittelu	15
5.3.1 Datalokien automaattinen nimeäminen	20
5.3.2 Mittaustulosten graafinen esitys	21
5.3.3 Vanhojen datalokien poistaminen	23
5.3.4 Sovelluksen viimeistely	24
6 POHDINTA	30
LÄHTEET	32
Liite 1 Nykyisen ja vanhan lokin nimeämisosio (ei julkisessa opinnäytetyössä)	
Liite 2 Vertailulohko (ei julkisessa opinnäytetyössä)	

## SANASTO

CSV	Comma Separated Value, taulukkomuotoisen tiedon tiedostomuoto, jossa sarakkeet erotellaan pilkuilla.
CV	Current Value, tämänhetkinen arvo; Siemensin toimilohkoissa käytetty parametrin tai signaalin tunnus.
PLC	Programmable Logic Controller, ohjelmitava logiikka.
PV	Preset Value, asetusarvo; Siemensin toimilohkoissa käytetty parametrin tai signaalin tunnus.
TIA Portal	Totally Integrated Automation Portal, Siemensin kehittämä ohjelmointityökalu.
VOC	Volatile Organic Compounds, haihtuvat orgaaniset yhdisteet.

# 1 JOHDANTO

Opinnäytetyöksi sovittiin datankeruusovelluksen ohjelmointi Siemensin S7-1200-logiikalla ja TIA Portalilla sekä datan esitys graafisena Siemensin TP1200 Comfort -paneelilla. Datankeruusovelluksen tilaajana toimi kotimainen Genano Solutions ja toimittajana Ambit Oy. Genano Solutions on tuotantoprosesseista aiheutuvien ilman epäpuhtauksien puhdistuslaitteistoihin erikoistunut yritys (1).

Datankeruusovellus tulisi käyttöön VOC-polttolaitokseen Kuopioon. VOC on lyhenne sanoista Volatile Organic Compunds, eli haihtuvat orgaaniset yhdisteet. Yritys halusi tallentaa lämpötila- ja painemittauksia sähköiseen tietokantaan. Paneeliin tulevan käyttöliittymän tulisi kyetä näyttämään kuvaajia eri mittauksista ja siitä tulisi voida muuttaa kuvaajan aikaväliä ja datan tallennusväliä.

Tämän raportin liitteet ovat tarkoitettu ainoastaan yrityksen ja opinnäytetyön tekijän käyttöön, eivätkä ne ole näkyvillä opinnäytetyön julkisessa versiossa.

## 1.1 Ambit Oy

Tämä työ tehtiin Ambit Oy:lle. Ambit Oy on vuonna 1993 perustettu oululainen automaatioalan yritys, jossa työskentelee tällä hetkellä kolme henkilöä (2). Yrityksen toimintaan kuuluu energia-alan automaation konsultointi, suunnittelu, ohjelmisto ja järjestelmien toimittaminen. Ambit Oy kehittää ja valmistaa automaatiojärjestelmiä, jotka mahdollistavat kannattavan, tehokkaan ja ympäristöystävällisen tuotantoprosessien valvonnan. (3)

Vuonna 1984 perustettu oululainen sähkö- ja laitostekniikan yritys Slatek Oy osti Ambit Oy:n keväällä 2011, minkä jälkeen Ambitin tavoitteeksi tuli kasvaa merkittäväksi toimijaksi Suomen lisäksi muissa Pohjoismaissa. Slatekilla työskentelee liki 30 henkilöä ja se tekee yhteistyötä Ambitin kanssa. (3, 4.)

## 1.2 Työn tavoitteet

Ambit Oy sai helmikuussa 2018 tilauksen kotimaiselta Genano Solutions Oy:ltä. Tilauksen kohteena oli Genanon VOC-polttolaitoksen automaattinen tiedonkeruuratkaisu Siemensin ohjelmoitavilla logiikoilla. Datankeruun tulisi olla dynaaminen, eli mittauksien eri parametrejä, kuten mittausväliä ja datan säilytysaikaa, voitaisiin muuttaa. Lisäksi data olisi hyvä näyttää kuvaajana. Tällä hetkellä polttolaitoksella ei ole tiedonkeruuta.

Asiakkaalle tulee käyttöön Siemensin TP1200 Comfort -kosketusnäyttöpaneeli, johon tulisi suunnitella käyttöliittymä datankerulle. Dataa tallennettaisiin erillisiin tiedostoihin, joista käytetään nimitystä datalokit. Tallennussijaintina toimisi paneeliin liitetty muistitikku, koska PLC:ssä voidaan käyttää ainoastaan kalliita Siemensin muistikortteja. Käyttöliittymästä tulee voida käynnistää ja pysäyttää datankeruu, sekä nähdä reaaliaikaiset mittausarvot. Käyttöliittymästä tulisi myös saada näkymään kuvaaja tallennetuista mittausarvoista ja pystyä muuttamaan datankeräysväliä sekä aikaväliä, jolta dataa näytetään.



## 2 VOC-POLTTOLAITOS

VOC-yhdisteitä (Volatile Organic Compounds, haihtuvat orgaaniset yhdisteet) syntyy muun muassa erilaisia maaleja, lakkoja, metallien pinnoituksia ja tietyntyyppisiä muoveja valmistettaessa. Myös liikenteestä ja suurista kemian-, öljyn- ja energiantuotantolaitoksista tulee VOC-päästöjä. Niiden kemiallisia ominaisuuksia ovat alle 250 °C:n kiehumispiste ja 0,01 kPa:n paine 20 °C:ssa. (5) VOC-yhdisteet ovat ongelmallisia siten, että ne jäävät alempaan ilmakehään, mistä ne saaten mukana imeytyvät maaperään ja saastuttavat pohjavesiä ja satoa (6). Lisäksi ne pahentavat kasvihuoneilmiötä lisäämällä otsonin määrää alailmakehässä (5).

VOC-yhdisteiden vähentämiseen on monia eri tekniikoita, joista yksi on katalyyttinen poltto, jota Genano Solutionskin käyttää. Katalyyttisen polton etu termiseen polttoon on sen huomattavasti alempi polttolämpötila, joka on n. 300 °C, kun taas termisessä poltossa lukema on n. 850 °C. Katalyyttisen polton etuja ovat jäänospäästöjen olemattomuus, mikä johtuu alhaisesta lämpötilasta, sekä systeemin tuottama energia, jota voidaan hyödyntää polttolaitoksen käyttöenergiana. Käyttöenergian hyödyntämisen myötä VOC-polttolaitoksen käyttökustannukset ovat alhaiset. (7, 8.)

### 3 SIEMENS TIA PORTAL

TIA Portal (Totally Integrated Automation Portal) on saksalaisen Siemensin vuonna 2011 julkaisema ohjelmointityökalu Siemensin omille laitteistoille. Työkalu sisältää logiikoiden, käyttöliittymien ja turvaratkaisujen sekä taajuusmuuttajien ohjelmoinnin. TIA Portalissa on yhdistettynä Siemensin Step 7-, WinCC- ja SINAMICS Startdrive -ohjelmistot. Step 7 on tarkoitettu logiikoiden ohjelmoimiseen, WinCC käyttöliittymien tekoon ja Startdrive taajuusmuuttajien käyttöön. (9, 10) Tässä työssä käytettiin pääasiassa Step 7:ää, mutta hiukan myös WinCC:tä.

Step 7:ssä on mahdollista käyttää kolmea eri ohjelmointikieltä: LAD (Ladder Diagram), FBD (Function Block Diagram) ja SCL (Structured Control Language). Näistä kolmesta LAD ja FBD ovat graafisia ohjelmointikieliä ja SCL tekstipohjainen komentolista. Graafisten kielten etuina ovat helppo luettavuus. SCL soveltuu parhaiten monimutkaisten algoritmien ohjelmointiin, jotka voidaan kirjoittaa ehtolausekkeiden avulla.(11, s. 23; 12) Tämä työ toteutettiin LAD-kielellä, koska sitä on helppo lukea ja sovellus ei vaatinut monimutkaisia lausekkeitä.

## 4 TIEDONKERUU

Tiedonkeruu (data logging) perustuu tietokoneella tai erillisellä laitteella tehtävään mittaustiedon lukemiseen sensoreista sekä sen analysointiin ja tallennukseen. Tiedonkeruuta käytetään yleensä tieteellisissä kokeissa ja valvontajärjestelmissä, kuten esimerkiksi polttolaitoksen lämpötilojen ja paineiden seurannassa. Tiedonkeruun etuina ovat suurien datamäärien nopea kerääminen tarkasti. Tietokoneella toteutettu tiedonkeruu mahdollistaa myös ympärivuorokautisen tallentamisen ilman käyttäjää, minkä johdosta ihmiset voivat keskittyä muihin asioihin. Tiedot tallentuvat joko keräävän laitteen muistiin tai ne voidaan lähettää etälaitteeseen esimerkiksi internetin välityksellä. Tietojen lähetys mahdollistaa mittausten seurannan yhdestä paikasta, vaikka mittauksia tehtäisiinkin monessa eri sijainnissa. Tyypillisesti tieto tallennetaan tekstitiedostoon, josta voidaan nähdä mitattavan suureen lisäksi päivämäärä, kellonaika ja muita käyttäjälle tarpeellisia tietoja. Tekstitiedostoista voidaan luoda kuvaajia, jotka helpottavat havainnointia. (13, 14, 15.)

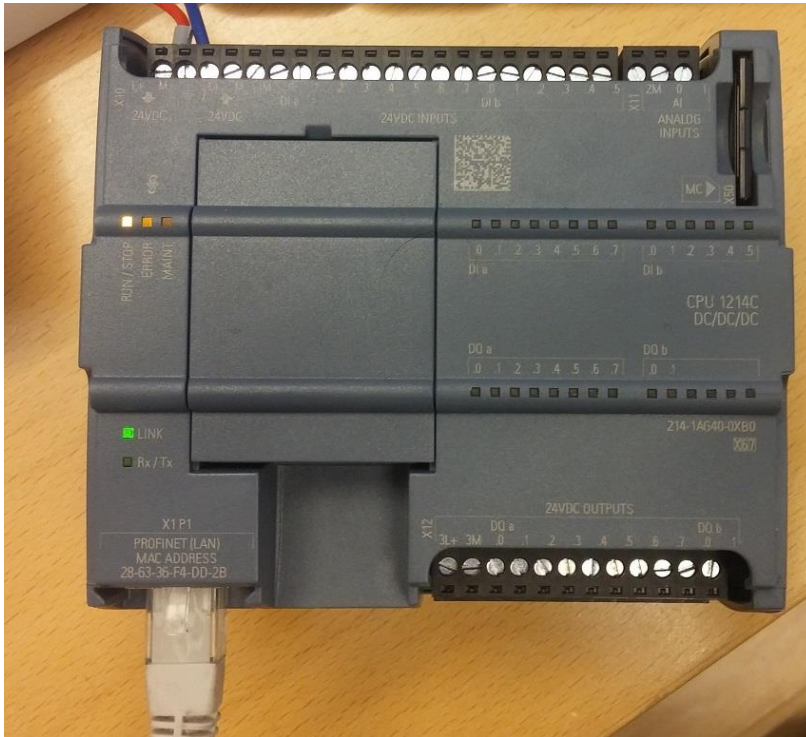
## 5 TYÖN TOTEUTUS

### 5.1 Työn aloitus

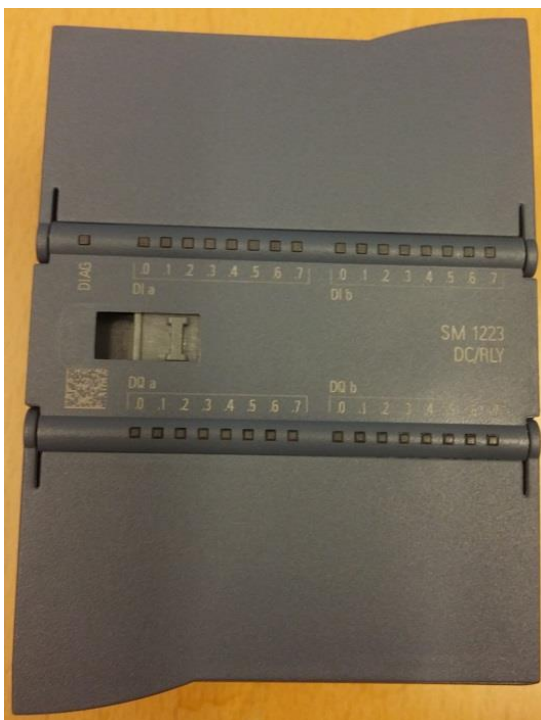
Työn toteutus alkoi TIA Portalin tarjoamien tiedonkeruuminaisuuksien kartoittamisesta. Netistä löytyi hyviä videotutoriaaleja datankeruun toteuttamiseen Siemensin S7-logiikoilla. Seuraavaksi alkoi selvitystyö, kuinka asiakkaan vaatimukset saataisiin parhaiten huomioon otettuna käytössä olevilla työkaluilla. Selvitettäväksi muodostui myös datalokien käyttämä tila, koska logiikalla tehdyt datalokit tallentuvat logiikan omalle muistikortille. Ongelmana oli muistikorttien korkea hinta, koska ainoastaan Siemensin omat muistikortit toimivat logiikoissa, joten täytyi ratkaista, miten saataisiin yhden vuoden ajalta mittaustietoja talteen mahdollisimman pienen muistikortin avulla. Tarkoituksena oli tallentaa datalokit logiikan muistikortille ja siirtää ne paneeliin liitetulle muistikortille, jonka jälkeen datalokit voitaisiin poistaa logiikasta. Myöhemmin selvisi, että datalokien tallennukseen sopivat myös tavalliset muistikortit.

Työhön kuului myös kosketusnäyttöpaneelin käyttöliittymän suunnittelu, joka näyttäisi kuvaajan datalokiin tallennetuista mittaustuloksista. Asiakas oli tilannut Siemensin TP1200 Comfort-paneelin, jonka kosketusnäytön resoluutio on 1280 x 800 pikseliä. Käyttöliittymän suunnittelusta TIA-ympäristössä ei minulla ollut aikaisempaa kokemusta, joten siitäkin aiheesta tuli paljon uutta tietoa.

Työssä käytettiin Siemensin S7-1200 -ohjelmoitavaa logiikkaa (kuva 1), digitaalisten lähtöjen lisäkorttia (kuva 2), analogisten liitäntöjen lisäkorttia (kuva 3), Siemens TP1200 Comfort -kosketusnäyttöpaneelia (kuva 4), Ambient Oy:n kannettava tietokonetta sekä Siemensin 256 Mt:n muistikorttia. Työn aikana käytettyjä ohjelmia ovat: Siemens TIA Portal V13, V14 ja V15, Siemens PLCSIM V13, Microsoft Word ja Microsoft Excel.



KUVA 1. Siemensin S7-1200-logiikka



KUVA 2. Digitaalisten liitântöjen laajennuskortti



*KUVA 3. Analogisten liitäntöjen lisäkortti*



*KUVA 4. Siemensin TP1200 Comfort -paneeli*

## 5.2 Tiedonkeruusovelluksen vaatimukset

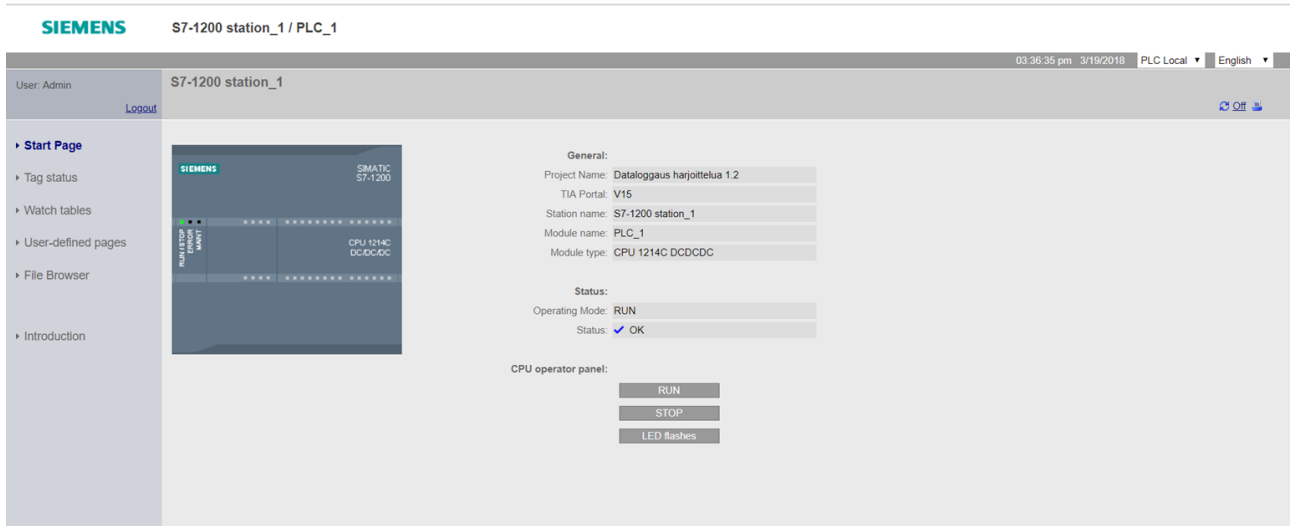
Asiakas halusi kerätä tietoa kuudesta lämpötilamittauksesta ja kolmesta painemittauksesta. Mittaukset tulisi kerätä talteen lämpötiloista yhden minuutin välein ja paineista puolen minuutin välein, minkä jälkeen ne arkistoitaisiin yhden tai useamman vuoden ajaksi. Lisäksi datasta olisi hyvä saada piirrettyä graafi, jonka aikaväliä voisi muuttaa kosketuspaneelistä, ja myös datankeruun aikaväliä pitäisi pystyä muuttamaan käyttäjän haluamaksi (dynaamisuus).

Asiakas tarvitsi siis luotettavan ja dynaamisen ratkaisun, jonka toteutus onnistuisi parhaiten ohjelmoitavilla logiikoilla eli PLC:llä (Programmable Logic Controller). Ohjelmoitavia logiikoita on monilta eri valmistajilta, mutta päädyimme Siemensin logiikoihin oman ja Ambitin kokemustaustan perusteella.

## 5.3 Logiikkasovelluksen suunnittelu

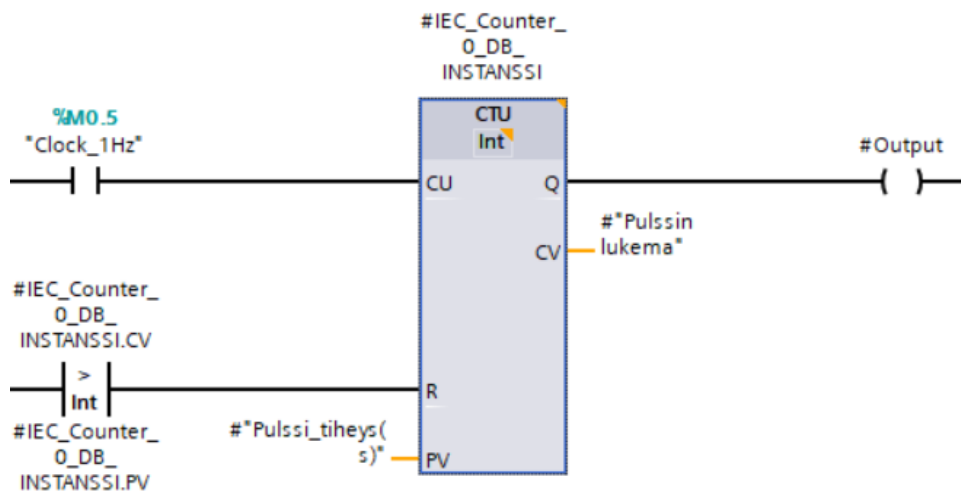
PLC:n ohjelmointi aloitettiin tekemällä datankeruusovellus netistä löytynyttä videotutoriaalia apuna käyttäen, josta selvisi datankeruun perusteet S7-1200-logiikoille (16). TIA Portalin toimilohkojen käyttöohjeita tutkimalla selvisivät lohkojen tulojen ja lähtöjen merkitykset. Pian huomattiin, ettei Siemensin PLCSIM-simulointiohjelmalla pysty simuloimaan datankeruuta, sillä datalokit tallentuvat PLC:n sisäiselle muistikortille. Tilatut logiikat eivät olleet vielä saapuneet, mutta saimme Slatekin puolelta lainaan samanlaisen logiikan. Mallisovelluksen onnistuneen kokeilun jälkeen sovellusta alettiin muokkaamaan sellaiseksi, että se tulisi täyttämään asiakkaan asettamat vaatimukset.

Tallennettuja tiedostoja pääsee tarkastelemaan syöttämällä logiikan IP-osoite selaimen osoitekenttään, jolloin aukeaa näkymä TIA Portalin Web server -aloitussivulle (kuva 5). Vaatimuksena on, että Web server on otettu logiikan asetuksissa käyttöön ja että serverille on luotu tarpeelliset oikeudet omaavat käyttäjätunnukset.



KUVA 5. Web serverin aloitussivu

Mittauksien aikavälien haluttiin olevan lämpötiloilla 60 s ja paineilla 30 s, mutta logiikasta löytyi valmiina ainoastaan 0,5 Hz:n kellopulssi ja sitä suurempia. Ambientin tekemässä sovelluskirjastossa oli valmiina sovelluksia eripituisille pulseille, joita käyttäjä voi muuttaa haluamansa pituisiksi. Pulssit oli tehty yhden hertsin kellopulssilla, vertailijalla ja Count Up (CTU) -laskurilohkolla (kuva 6).



KUVA 6. Pulssilohko

CTU-lohkon lähtö Q aktivoituu, kun laskurin tämänhetkinen arvo CV (Current Value) ylittää asetusarvon PV (Preset Value). CV:n arvoa kasvatetaan joka sekunti yhdellä ja lohkon toiminta aloitetaan alusta, kun CV on isompi kuin PV.

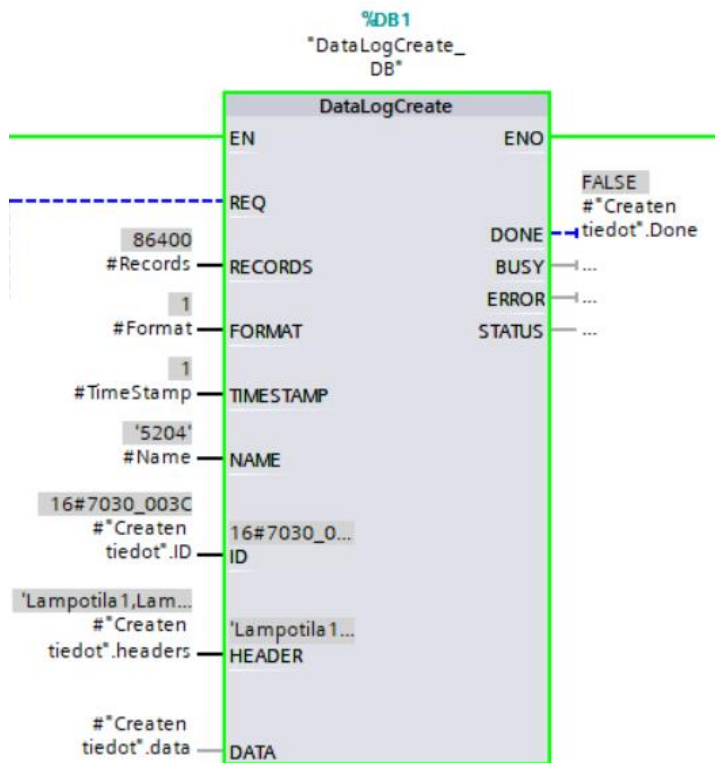


Asiakas ei ollut määritellyt aikaväliä, jonka yksi loki kattaisi. Mietinnän jälkeen päätettiin, että järkevin ratkaisu lokien jakamiseen olisi kuukauden välein. Kuukausittaisessa jaossa tiedostojen lukumäärä pysyisi kohtuullisena, jos tietoja säilytettäisiin yhden tai useamman vuoden ajan, ja niistä pystyisi näkemään kuukausittaiset keskiarvot. Tällöin uusi lokitiedosto luotaisiin, kun sen hetkisen kuukauden datat on kirjattu talteen.

Lokitiedosto luodaan DataLogCreate-lohkolla (kuva 7), joka suoritettaessa luo lokin käyttäjän määrittelemien parametrien mukaan. Käyttäjä voi määritellä lohkon tuloilla seuraavat asiat:

Records	Mittauksien maksimi lukumäärä yhtä lokia kohti.
Format	Lokin tiedostotyyppi, jonka vaihtoehtoina ovat "internal" ja "csv" eli Comma Separated Value (pilkulla erotetut arvot). Manuaalin mukaan Internal-vaihtoehto ei ole käytettävissä S7-1200 ja S7-1500 logiikoissa, joten käytettiin CSV:tä. Taulukkolaskentaohjelmistojen (Excel, OpenOffice Calc) kanssa yhteensopivana se sopii muutenkin paremmin tähän työhön.
Timestamp	Aikaleima eli kellonaika, jolloin mittaus on tehty tai muu talletettava arvo on muodostettu. Aikaleiman voi halutessaan jättää pois, mutta tässä yhteydessä se haluttiin näkyviin.
Name	Lokitiedoston nimi. Lokia ei voi luoda, jos sillä ei ole kelvollista nimeä. Nimen kriteerit on määriteltä TIA Portalissa DataLogCreate-lohkon manuaalissa.
ID	Lokin Identifier eli tunniste. Käyttäjän ei tarvitse määrittää tunnistetta, vaan lohko luo sen itse. Tuloon tulee liittää muuttuja, johon ID tallennetaan.
Header	Otsikko eli sarakkeen nimi. Mikäli halutaan useita sarakkeita, tulee niiden nimet erotella pilkulla.

Data Lokitiedostoon tallennettava tieto.



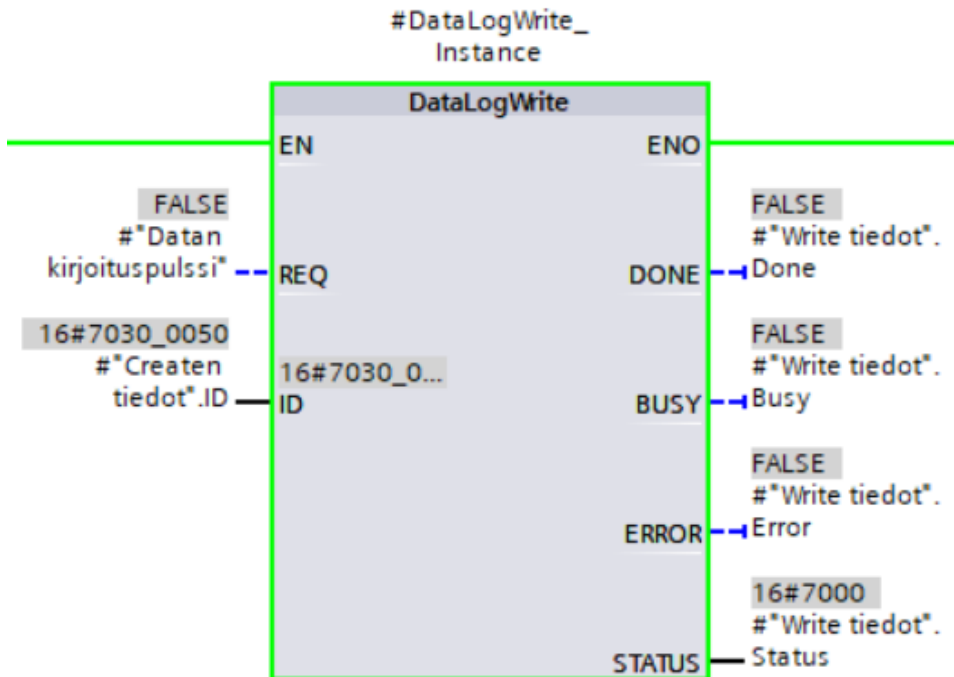
KUVA 7. DataLogCreate-lohko

DataLogCreate luo kirjoitusvalmiin tiedoston, jonka se jättää auki kirjoittamista varten ja jossa ei ole vielä muuta sisältöä kuin sarakkeiden nimet päivämäärälle, kellonajalle ja mittauksille (kuva 8).

	A	B	C	D	E	F	G	H	I	J	K	L
1	Record	Date	UTC Time	Lampotila1	Lampotila2	Lampotila3	Lampotila4	Lampotila5	Lampotila6	Paine1	Paine2	Paine3
2												
3												
4												

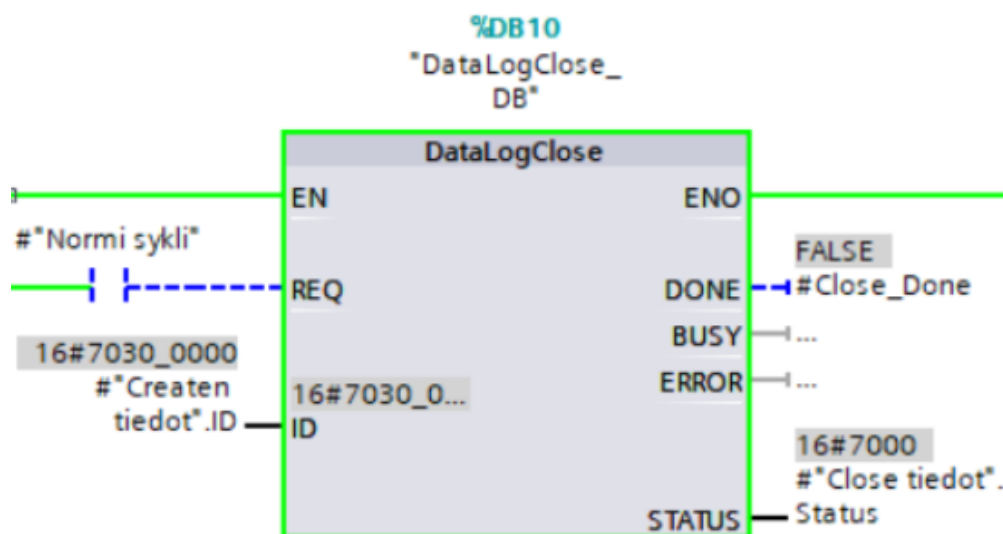
KUVA 8. DataLogCreaten luoma CSV-tiedosto avattuna OpenOffice Calc -ohjelmalla

Datalog-tiedostoon kirjoitetaan data DataLogWrite-lohkolla (kuva 9). DataLogWrite kirjoittaa ID-tulolla määritelyyn avoimeen datalokiin DataLogCreaten DATA-tuloon määritellyt mittaustiedot. Jos tiedosto tulee täyteen, kirjoituslohko jatkaa kirjoittamista viimeisen rivin jälkeen ensimmäisten rivien päälle, ellei kirjoitusta pysäytetä tai datalokia suljeta. DataLogCreaten ja -Writen avulla sovellus loi yhden tiedoston, johon arvoja tallennettiin.



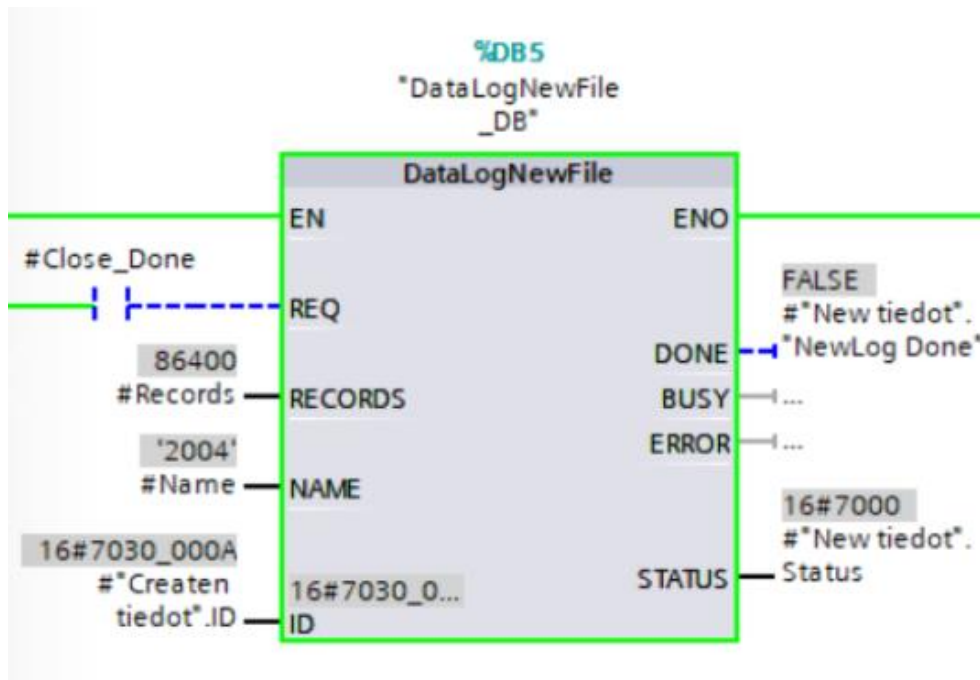
KUVA 9. DataLogWrite-lohko

Datalokit suljetaan DataLogClose-lohkolla (kuva 10), jotta tallennetun datan päälle ei kirjoitettaisi uutta dataa. DataLogClose sulkee suoritettaessa sen ID-tuloon määritellyn tiedoston. Datalokeja voi olla auki samanaikaisesti korkeintaan 10, joten kuukausittaisen lokijaon sekä pitkän säilytysajan takia ne suljettiin kirjoittamisen jälkeen.



KUVA 10. DataLogClose-lohko, jonka käynnistystuloon REQ on liitetty "Normi sykli" -muuttuja

DataLogNewFile (kuva 11) luo uuden tiedoston perustuen DataLogCreate-lohkon parametreihin ja suorittaa datan kirjoittamisen aikaisemmin luodulla DataLogWrite-lohkokolla. DataLogCreatella luodun lokin jälkeiset lokit voitiin luoda lisäämällä ohjelmaan lohkot DataLogClose ja DataLogNewFile. Jokaisella uudella lokilla tulee olla uniikki nimi, koska kahta samannimistä tiedostoa ei voi olla samaan aikaan Web serverillä.



KUVA 11. DataLogNewFile, jonka REQ-tuloon on liitetty "Close\_Done"-muuttuja

### 5.3.1 Datalokien automaattinen nimeäminen

Seuraava työvaihe oli datalokien automaattinen nimeäminen uniikeilla ja loogisilla nimillä. Tultiin työpaikkaohjaajan kanssa lopputulokseen, että paras tapa nimeä datalokit olisi kuukausi ja vuosi esimerkiksi 3/2018, koska lokit oltiin jaettu kuukausittain. Kauttaviivaa ei kuitenkaan voi käyttää datalokin nimessä, koska se ei ole sallittu merkki, joten lopullinen muoto olisi 32018. Vaihtamalla kuukausi ja vuosi toisinpäin tiedostot saa arkistoitua aikajärjestykseen nimen perusteella. Lokien nimeäminen muutettiin muotoon 201803.

Testausvaiheessa käytin datalokin nimeämiseen ympäri pyörähtävää laskuria, joka laski välillä 0–255 ohjelman ollessa käynnissä. Lisäksi määritin kirjattavan datan määräksi viisi riviä. Laskurin arvo muunnettiin String-tyypiseksi eli

merkkijonoksi ja tallennettiin muuttujaan. Automaattisen nimeämisen testaus vuosilla ja kuukausilla olisi ollut epäkäytännöllistä, koska uusi loki olisi luotu aina kuukauden vaihtuessa, ja sen takia usean lokin luomisen testaamisessa olisi mennyt liian paljon aikaa. Tässä vaiheessa sovellus kirjoitti yhden tiedoston täyteen, sulki sen ja sitten loi uuden lokin, johon kirjoittaminen jatkui.

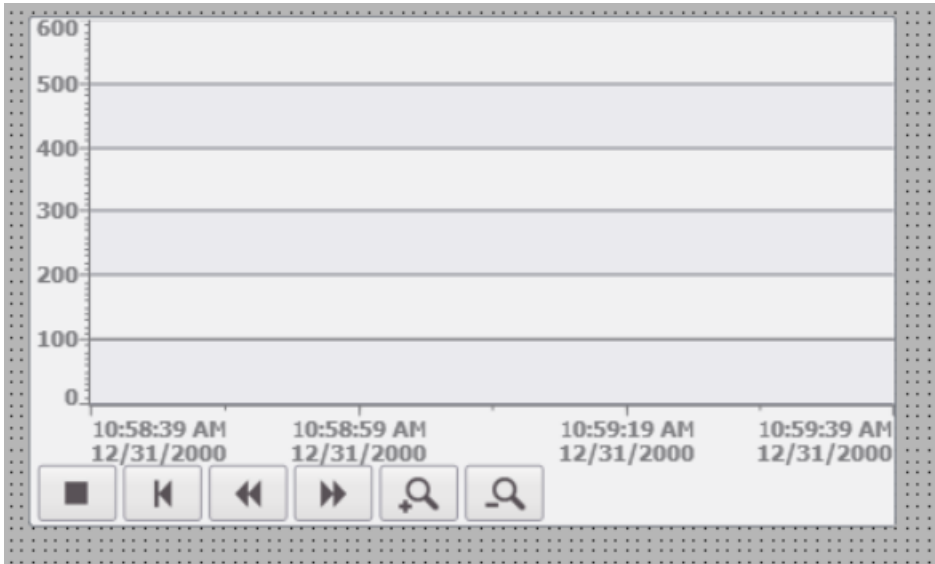
Testauksessa kävi ilmi, että välillä ohjelma loi yhdestä viiteen datalogia, minkä jälkeen mitään ei enää tapahtunut, vaikka ohjelman olisi pitänyt jatkaa pysäyttämiseensä asti. Huomattiin, että sovellus ei tee mitään, jos datalokin nimeämiseen käytetyn laskurin arvo oli pienempi kuin 100. Tämä johtui siitä, että laskurin arvoa muunnettaessa merkkijonoksi sille piti määrittää, kuinka monta merkkiä se sisältää. Olin määritellyt merkkijonon pituudeksi kolme merkkiä, mutta lukeman ollessa alle 100 oli yksi tai kaksi merkeistä tyhjä (NULL tai Space). Tämä tarkoitti sitä, että luku "11" muuntui muotoon " 11", mutta tyhjiä merkkejä ei saa olla datalokin nimessä. Datalokien lukumäärä vaihteli riippuen siitä, missä vaiheessa laskuri oli menossa, kun datankeruu käynnistettiin. Jos laskuri oli datankeruun käynnistyshetkellä lähellä ympäri pyörähtämistä, datalokeja ehdittiin luoda yksi ennen kuin lokin nimi muuttui kelvottomaksi, ja jos laskuri oli juuri vaihtunut 100:n datalokeja ehdittiin luoda viisi kappaletta. Jos lokin nimi vaihtui kelvottomasta kelvolliseksi sovelluksen suorittamisen aikana, se ei osannut jatkaa toimintaansa ja sovellus piti resetoida.

Sama ongelma tulisi vastaan myös kuukausilla, kun oltaisiin tammikuun ja syyskuun välillä. Yksi vaihtoehto ongelman korjaamiseen olisi kirjoittaa kuukausien nimet auki, koska tällöin ei merkkien määrää tarvinnut määrittää. Tähän Ambitilla olisi ollut olemassa valmis lohko. Toinen vaihtoehto olisi pakottaa kuukauden numeron eteen nolla, kun numero on alle 10. Päädyttiin jälkimmäiseen vaihtoehtoon, jossa kuukauden numeron eteen lisättiin nolla, koska se mahdollistaisi tiedostojen järjestämisen aikajärjestykseen nimen perusteella.

### **5.3.2 Mittaustulosten graafinen esitys**

Sovellus alkoi olla jo hyvällä mallilla, joten alettiin tutkimaan kosketusnäyttöpaneeliin tehtävää käyttöliittymää ja graafin näyttämistä. TIA Portalin käyttöliittymän suunnittelussa voitiin käyttää ohjelman valmista trendiviivan piirtäjää (kuva 12),

mutta ongelmaksi muodostui mittausarvojen lukeminen Web serveriltä. Web serveriin tallennettuja arvoja ei nimittäin pystynyt lukemaan suoraan trendin piirtotyökalulla.



*KUVA 12. Trendin piirtotyökalu*

Alettiin tutkimaan voisiko datalogeja siirtää Web serveriltä paneeliin. Yksi vaihtoehto olisi tehdä se VBScriptillä (Visual Basic Script). Netistä löytyi ohje (17, s. 27), joka sisälsi aiheeseen liittyvän esimerkin VBS:llä tehtävistä tiedostojen pitkäaikaisesta arkistoinnista. Skriptiä ei jostain syystä saatu toimimaan, joten alettiin selvittämään muita mahdollisuuksia.

Eräs Slatekin työntekijöistä ehdotti, voisiko FTP (File Transfer Protocol) -serveriä hyödyntää lokien siirrossa paneeliin, sillä paneelissa oli Windows-pohjainen käyttöjärjestelmä. Pian kuitenkin selvisi, ettei paneeli tue FTP-serveriä eikä myöskään jaettua verkkokansiota.

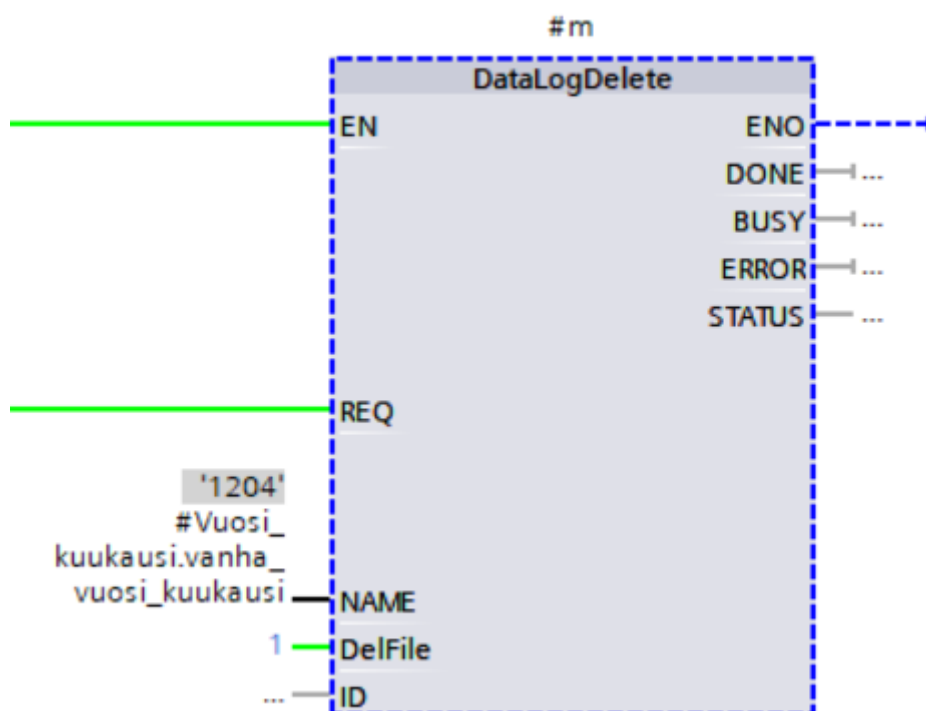
Selviteltiin lisää TIA:ssa olevaa trendin piirtotyökalua, sillä siinä vaikutti olevan tarvittavat ominaisuudet, kuten näytetyn trendin kelaaminen taaksepäin ja kaikkien tai yksittäisen trendin näyttäminen. Trendiin oli mahdollista valita piirrettävän graafin tuloksi joko käynnissä oleva mittaus tai sitten "datalog". Datalog vaihtoehto tarkoitti kuitenkin vain paneelin luomia logeja, joiden käyttö ei tässä tapauksessa ollut käytännöllistä, sillä logiikalla tehty dataloggaus on toiminnaltaan varmempi kuin paneelilla.

Graafin teko logiikalla tehdyistä datalokeista alkoi vaikuttaa todella haastavalta, mutta päätin vielä kokeilla voisiko luotuja datalokeja avata paneelissa olevalla Excel-ohjelmalla ja näyttää siten graafina näytöllä. Paneelin Excel ei kuitenkaan tukenut CSV-tiedostoja, vaan pelkästään xls- ja xlsx-tiedostoja eli tiedostot olisi pitänyt ensin muuntaa toiseen muotoon ja sitten siirtää paneeliin. Graafin piirto paneeliin hylättiin, koska sitä ei ollut asiakkaan vaatimuksissa ja sen toteuttaminen koitui liian haastavaksi näillä aikatauluilla. Palattiin takaisin logiikan puolelle viimeistelemään sovellusta.

### **5.3.3 Vanhojen datalokien poistaminen**

Täyteen tulleen datalokin sulkeminen ja uuden luominen perustui siihen, että datalokiin on kirjattu käyttäjän määrittelemä lukumäärä mittaustietoja. Lopullisessa ohjelmassa tämä tarkoitti 86 400 rivin kirjoitusta, jos dataa tallennetaan 30 päivän ajan 30 s:n välein. Kaikki kuukaudet eivät kuitenkaan sisällä 30 päivää, joten joissakin kuukausissa osa mittauksista menisi väärän kuukauden tiedostoon. Ambientilla oli valmis lohko, joka mahdollisti uuden lokin luomisen aina kuukauden vaihtuessa. Lohko mahdollisti myös muiden ajanhetkien käytön, kuten esimerkiksi tietyn kellonajan käytön, "liipaisimena". Nyt kun uusia logeja saatiin luotua automaattisesti, oli aika selvittää, kuinka vanhin loki saataisiin poistettua automaattisesti, jotta tila ei loppuisi kesken muistikortilta.

Päätettiin, että riittävä lokien säilytysaika olisi kaksi vuotta, koska käytössä olevalle muistikortille mahtuisi hieman yli kaksi vuotta dataa aikaisemmin määritellyllä 30 s:n tallennusvälillä. Datalokin poistoon oli TIA Portalissa valmis lohko DataLogDelete (kuva 13), joka poistaa name- tai id-tulolla määritellyn datalokin Web serveriltä.



KUVA 13. DataLogDelete-lohko

Mikäli ID-tulo oli määritelty, nimeä ei huomioitu lokin valinnassa, mutta jos haluttiin valita loki nimen perusteella, piti ID-tulo jättää tyhjäksi ja syöttää Name-tuloon poistettavan lokin nimi. Alkoi mietintä siitä, kuinka saataisiin syötettyä DataLogDeleten tuloon automaattisesti vanhimman datalokin nimi tai id. Ratkaisin ongelman vaihtamalla datalokien nimeämisen tapahtuvan järjestelmäajan mukaan, eli kun uusi loki luodaan, sen nimeksi tulee sen hetkinen aika. Tästä ajasta vähensin kaksi vuotta ja käytin sitä aikaa vanhimman datalokin nimeämiseen ja poistamiseen. Nimeämissovellus näkyy liitteessä 1. Nyt sovellus luo datalokeja kunnes se pysähtyy sekä poistaa vanhimman lokin.

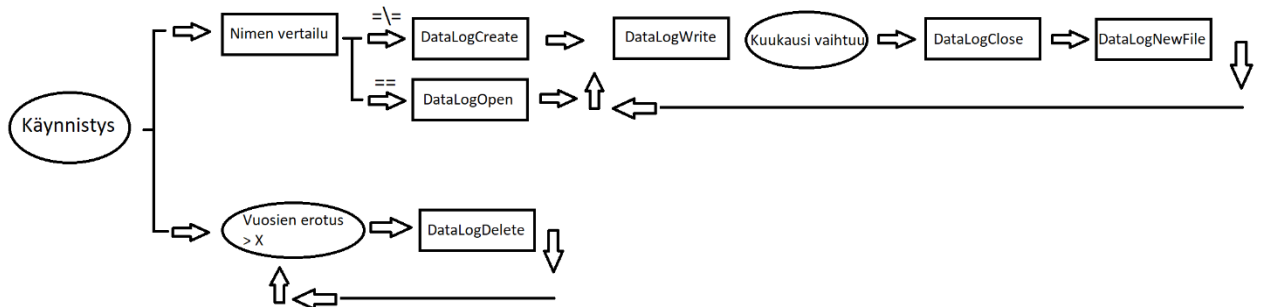
#### 5.3.4 Sovelluksen viimeistely

Sovellusta työstäessä pohdittiin myös, mitä tapahtuu sähkökatkoksen jälkeen jos logiikka käy pois päältä. Päätettiin, että logiikan tulisi jatkaa datankeruuta itsensä sähköjen palautuessa. Myös niin epätodennäköiseen asiaan kuin kuukauden ja siten myös lokin nimen vaihtumiseen sähkökatkon aikana oli varauduttava. Mielestämme helpoin ratkaisu olisi, että sovelluksessa olisi vertailija, joka vertaa viimeisimmän datalokin nimeä eli luontiaikaa nykyhetkeen, ja joko jatkaisi



kirjoittamista vanhaan lokiin tai sulkisi sen ja sitten loisi uuden lokin. Sovellus ottaisi ajan talteen joka kerta, kun DataLogWrite kirjoittaa arvon lokiin, ja vertaa sitä tämän hetkiseen aikaan. Mikäli aika olisi sama, sovellus avaisi lokin DataLogOpen-lohkolla ja alkaisi kirjoittaa siihen lisää dataa. Mikäli nimet eivät täsmäisi, tulisi viimeisin loki sulkea ja luoda uusi loki, johon datan kirjoittaminen tapahtuisi. Vertailulohko näkyy liitteessä 2.

Vertailulohkossa otetaan aluksi talteen tällä hetkellä kirjoitettavan datalokin nimi muuttuinaan "vanha\_aika". Vanhaa aikaa verrataan nykyiseen aikaan ja suoritetaan toimenpiteitä riippuen siitä, ovatko ajat samat. Jos ajat ovat samat, avataan vanha dataloki DataLogOpen-lohkolla ja annetaan DataLogWritelle käynnistyskäsky Busy Set -lähdöllä. Toisistaan poikkeavat ajat aktivoivat Normi sykli -lähdön, joka puolestaan käynnistää DataLogNewFilen ja sitä myötä ohjelman tavallisen suoritusjärjestyksen (kuva 14).



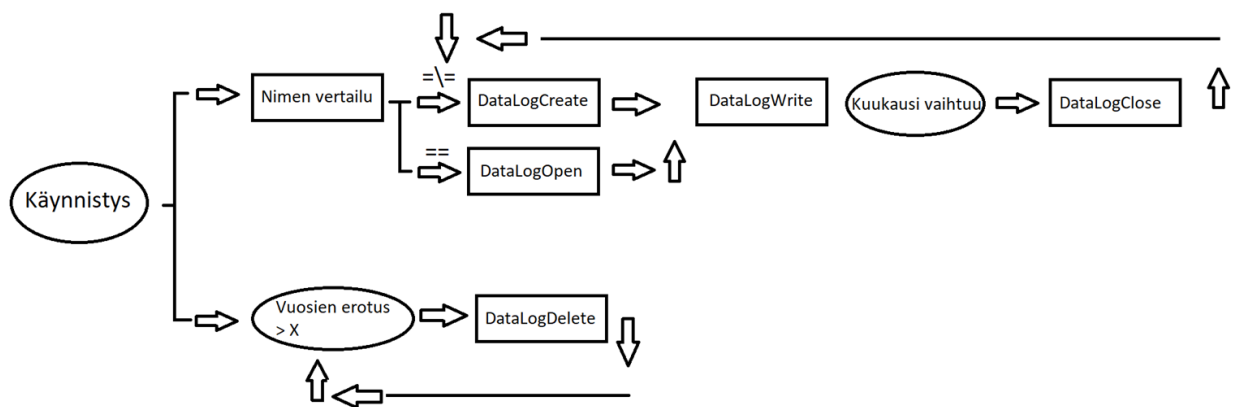
KUVA 14. Ohjelman lohkokaavio

Sähkökatkon sattuessa suoritusjärjestys on muuten sama, mutta mikäli nimet eroavat, suoritetaan DataLogClose ensin aikaisemmalle lokille ja sitten DataLogNewFile, -Write jne. Huomattiin, että jos sähkökatko tapahtui heti uuden lokin luonnin jälkeen ennen kirjoittamista, sovelluksen toiminta pysähtyi. Tämä johtui siitä, että Web serverillä oli pelkät sarakkeiden otsikot sisältävä tiedosto ja vanha\_aika ei ollut ehtinyt päivittyä. Kun vanha\_aika ei ollut päivittynyt, vertailulohko yritti luoda uudestaan jo olemassa olevaa lokia ja sovelluksen toiminta jumittui. Ongelma korjautui ottamalla vanha aika talteen aina, kun uusi loki luotiin sen sijaan, että tallennus tapahtuisi kirjoittamisen jälkeen.

Sovellusta tarkistaessa alettiin epäillä DataLogNewFilen tarpeellisuutta. Päätettiin testata, onnistuuko uuden tiedoston luonti DataLogCreaten avulla

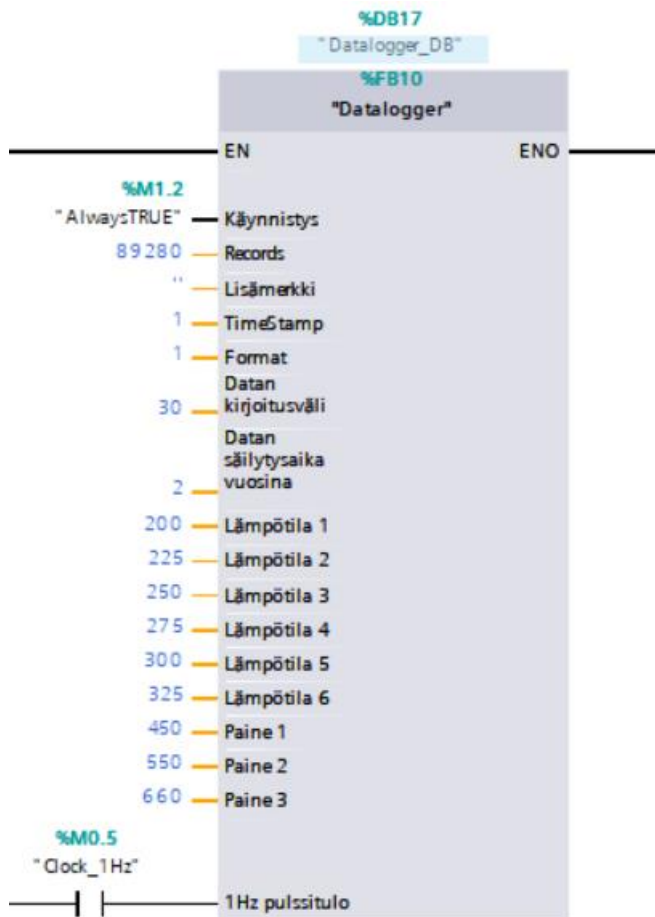
DataLogNewFileen sijaan. Testi suoritettiin poistamalla DataLogNewfile käytöstä väliaikaisesti ja yhdistämällä tarvittavat tulot ja lähdöt toisiinsa. Testin tuloksena huomattiin, että uusi loki voitiin luoda DataLogCreatella, joten DataLogNewFile poistettiin. DataLogNewFileen ideana on luoda uusi loki uniikilla nimellä perustuen DataLogCreatessa määriteltyihin parametreihin, mutta vertailijan ja automaattisen nimenvaihdon myötä sitä ei tarvittu.

Tuloksena sovellus toimi kuten aiemminkin, mutta siinä oli yksi lohko vähemmän (kuva 15). DataLogCreaten suoritukseen lisättiin yhden sekunnin viive, jotta aikaisempi loki ehditään varmasti sulkea, koska molemmat suoritettiin samalla liipaisulla.



KUVA 15. Parannelun ohjelman lohkokaavio

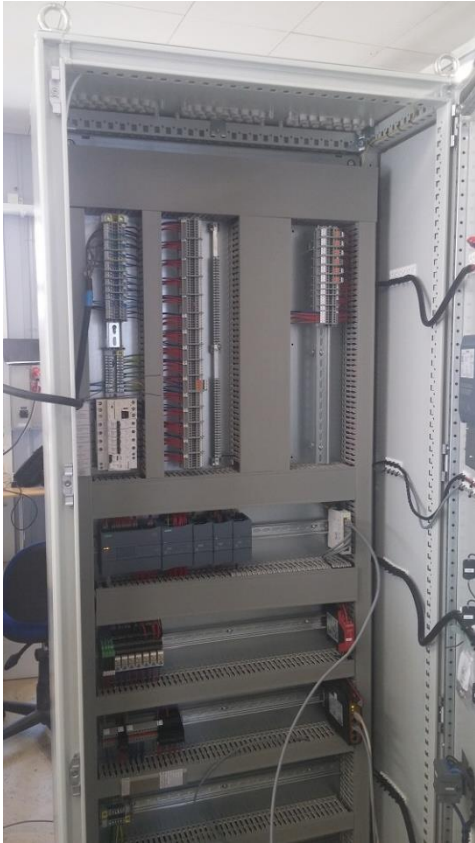
Sovellus jatkoi nyt lokien tekoa sähköjen katketessa, loi uuden lokin tarvittaessa ja poisti vanhimman lokin. Ohjelmaan muutettiin datalokien nimeäminen vuosikuukausi -muotoon, minkä jälkeen se liitettiin Ambit Oy:n TIA Portal -sovellukseen. TIA Portalissa sovelluksien yhdistäminen tapahtui yksinkertaisesti kopioi ja liitä -menetelmällä, jolloin ohjelma loi tämän raportin sovelluksesta yhden lohkon (kuva 16). Tätä lohkoa voidaan kopioida muihinkin järjestelmiin, joissa tiedonkeruuta tarvitaan.



KUVA 16. Dataloggaus-sovellus yhtenä lohkona

Sovellukseen oli määritelty tietyt muuttujat tulo-tyyppisiksi, jotka näkyvät lohkon vasemmassa reunassa. Näihin tuloihin käyttäjä syöttää haluamansa datalokien parametrit ja halutut lämpötila- ja painemittaukset. Lisämerkki-tulon avulla käyttäjä voi lisätä datalokien nimeen haluamansa merkin tai tekstin. Tämä mahdollistaa datalokien yksilöimisen jos Dataloggeri-lohkoa kopioidaan useaan eri kohteeseen.

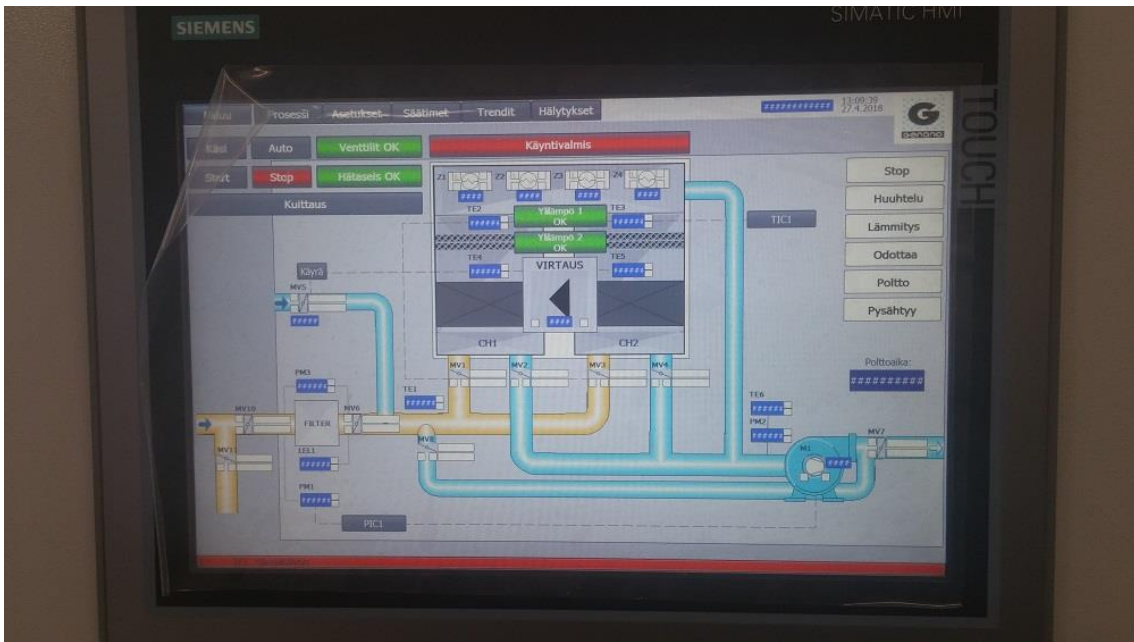
Lopuksi koko ohjelma ladattiin automaatiokaapissa (kuva 17) olevaan logiikkaan, minkä jälkeen kaappi lähetettiin Kuopion VOC-polttolaitokseen. Automaatiokaappiin oli Slatekin puolella asennettu tarvittavat kaapelit, logiikka ja lisäkortit (kuva 18) sekä paneeli (kuva 19).



*KUVA 17. Automaatiokaappi*



*KUVA 18. S7-1200-logiikka, digitaalisten liitântöjen laajennuskortti ja kolme analogisten liitântöjen laajennuskorttia*



KUVA 19. Siemensin TP1200 Comfort -paneeli, jossa näkyy VOC-polttolaitoksen käyttöliittymä

## 6 POHDINTA

Työn tarkoituksena oli ohjelmoida VOC-polttolaitokselle automatisoitu tiedonkeruusovellus, joka keräisi lämpötila- ja painemittausten tuloksia sähköiseen tietokantaan. Tietojen tallennusta ei ollut vielä olemassa ja sellainen tarvittiin, koska muun muassa ympäristöviranomaiselle on kyettävä osoittamaan polttolaitoksen toimivan oikein. Lisäksi tehtävänä oli selvittää, miten käytössä olevaan paneeliin saataisiin kuvaaja kerätyistä mittaustuloksista.

Henkilökohtaisiksi tavoitteiksi määriteltiin perehtyä TIA-ympäristön ja Siemensin ohjelmoitavien S7-logiikoiden tarjoamiin mahdollisuuksiin automatisoidussa datankeruussa sekä valmistaa tiedonkeruusovellus. Siemensin TIA-ympäristö ja S7-logiikat ovat maailmalla yleisessä käytössä erilaisissa automaattioratkaisuissa, joten sen osaamisesta on varmasti hyötyä tulevalle automaatioinsinöörille.

Työn aikana tuli mietittyä asiakkaan mahdollisesti tarvitsemia, mutta mainitsemattomia ominaisuuksia, koska tarkkoja vaatimuksia ei ollut, ja niiden toteutusta käytettävissä olevilla työkaluilla. Lisäksi pääsin perehtymään Siemensin käyttöliittymätyökaluun, josta minulla ei ollut aikaisempaa kokemusta.

Tarkkojen vaatimusten puuttumisen lisäksi työtä vaikeuttivat TIA Portalin version päivityksestä ilmenneet lisenssiongelmat ja yhteensopivuusongelmat Windows 10 -version kanssa. Työ aloitettiin Windows 7:llä, mutta käytössä ollut kannettava tietokone päivitettiin Windows 10:een. Käyttöjärjestelmän päivityksen myötä TIA Portal ei löytänyt tietokoneen verkkoasemia, vaikka ne näkyivät tietokoneen resurssienhallinnassa. Yksi näistä verkkoasemista oli varattu käyttööni ja olin talentanut kaikki opinnäytetyöhön liittyvät tiedostot sinne. Tästä eteenpäin käytin tiedostojen säilytykseen muistitikkua ja Oamkin opiskelijoiden käytettävissä olevaa OneDrive-pilvipalvelua. Lisenssiongelma oli, että ohjelmat näkyivät aktivoituina, mutta silti ne väittivät, että niitä ei ole aktivoitu. Ongelma ilmeni päivityksessä V13:sta V14:ään ja se jatkui vielä V15:ssä. Ongelman sai kierrettyä suorittamalla ohjelman järjestelmänvalvojana.

Lopputulos oli asiakkaan vaatimukset täyttävä automaattinen lämpötila- ja painemittausten tallennussovellus. Tallennetuista mittauksista ei saatu näkymään kuvaajaa sellaisena, kuin olisimme halunneet. Tähän olivat syynä lähinnä TIA Portalin rajoitukset sekä ajan ja kokemuksen puute VBScriptin käytöstä, koska sen avulla tiedon graafinen esitys olisi todennäköisesti onnistunut.

## LÄHTEET

1. Genano Solutions. Saatavissa: <http://genanosolutions.fi>. Hakupäivä 19.2.2018.
2. Ambit Oy. Finder. Saatavissa: <https://www.finder.fi/Automaatiota/Ambit+Oy/Oulu/yhteystiedot/124977>. Hakupäivä 19.2.2018.
3. Ambit. Ambit Oy. Saatavissa: [http://www.ambit.fi/sivu/fi/ambit\\_oy/](http://www.ambit.fi/sivu/fi/ambit_oy/). Hakupäivä 19.2.2018.
4. Slatek Oy. Finder. Saatavissa: <https://www.finder.fi/Vedenk%C3%A4sitteilylaitteita/Slatek+Oy/Oulu/yhteystiedot/194287>. Hakupäivä 19.2.2018.
5. VOC-päästöt. 2017. Wikipedia. Saatavissa: <https://fi.wikipedia.org/wiki/VOC-p%C3%A4st%C3%A4st%C3%B6t>. Hakupäivä 19.2.2018.
6. Ronkainen, Aki 2018. Ambit Oy. Keskustelu 19.2.2018.
7. VOC-päästöjen pienentäminen. Genano Solutions. Saatavissa: <http://genanosolutions.fi/voc-paastojen-pienentaminen/>. Hakupäivä 19.2.2018.
8. Regeneratiivisen katalyyttisen polton (RCO) edut VOC-puhdistuksessa. Formia Smart Flow. Saatavissa: <https://www.formiasmartvoc.com/fi/regeneratiivisen-katalyyttisen-poltton-rco-edut-voc-puhdistuksessa>. Hakupäivä 19.2.2018.
9. TIA Portal. Siemens. Saatavissa: <https://www.siemens.com/global/en/home/products/automation/industry-software/automation-software/tia-portal.html>. Hakupäivä 19.2.2018.
10. TIA Portal – teollisuusautomaation ohjelmistoalusta. Siemens. Saatavissa: [http://www.siemens.fi/fi/industry/teollisuuden\\_tuotteet\\_ja\\_ratkaisut/tuotesivut/tia\\_portal.php](http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/tia_portal.php). Hakupäivä 19.2.2018.



11. Mäenpää, Janne 2015. TIA Portal, S7-1200 ja WinCC –käyttöönotto. Opinnäytetyö. Helsinki: Metropolia Ammattikorkeakoulu. Saatavissa: <http://www.theseus.fi/bitstream/handle/10024/93547/ins.pdf?sequence=1&isAllowed=y> . Hakupäivä 14.5.2018.
12. IEC programming languages. Siemens. Saatavissa: <https://w3.siemens.com/topics/mea/en/tia-portal/controller-sw-tia-portal/simatic-step7-professional-v11/iec-programming-languages/pages/default.aspx> . Hakupäivä 14.5.2018.
13. Data Logger. 2018. Wikipedia. Saatavissa: [https://en.wikipedia.org/wiki/Data\\_logger](https://en.wikipedia.org/wiki/Data_logger) . Hakupäivä 19.2.2018.
14. Wheeler, Brian 2016. The Art of Data Logging. The Newstack. Saatavissa: <https://thenewstack.io/art-data-logging/> . Hakupäivä 19.2.2018.
15. Beal, Vangie. Data logging. Webopedia. Saatavissa: [https://www.webopedia.com/TERM/D/data\\_logging.html](https://www.webopedia.com/TERM/D/data_logging.html) . Hakupäivä 19.2.2018.
16. Al-Mukhtar, Ali 2017. S7-1200 Data logging Siemens PLC. Video. Saatavissa: [https://www.youtube.com/watch?v=E2tdtwrT\\_sE](https://www.youtube.com/watch?v=E2tdtwrT_sE) . Hakupäivä 15.2.2018.
17. Long-Term Data Archiving on SIMATIC HMI Operator Panels with WinCC (TIA Portal). 2015. Siemens. Saatavissa: [https://support.industry.siemens.com/cs/attachments/109477071/109477071\\_LongTermDataArchives\\_en\\_v10.pdf](https://support.industry.siemens.com/cs/attachments/109477071/109477071_LongTermDataArchives_en_v10.pdf) . Hakupäivä 26.3.2018.





