Tommi Piiparinen

Development of a PLC-based Control & Monitoring Device

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Automation Engineering

Bachelor’s Thesis

14 March 2014

Metropolia

Authors Title Number of Pages Date	Tommi Piiparinen Development of a PLC-based Control & Monitoring Device 24 pages + 1 appendix 14 March 2014
Degree	Bachelor of Engineering
Degree Programme	Automation Engineering
Specialization option	Manufacturing Automation
Instructors	Marjatta Piironen, Principal Scientist Heikki Saarelainen, Lecturer

This Bachelor's thesis was conducted for a global chemistry company, Kemira Oyj and its purpose was to develop a prototype of a PLC based control & monitoring device and to study possible ways to create an easily adaptable communication interface between PLCs and PC software. The thesis also introduces the reader to different technics including motion control and PID controllers used in the prototype.

The thesis was conducted during 1.10.2013 – 30.3.2014 at the Kemira Research & Development Center in Espoo. All hardware for the prototype was selected before the start of the thesis. The prototype was constructed, programmed and tested at the R&D Center's workshop.

As the results of this thesis a fully functional prototype of a PLC based control & monitoring device was constructed, programmed and tested, a fairly easily adaptable communication library was programmed and tested and extensive documentation of the created programs was produced for the company's internal usage.

Keywords	PLC, PID, TIA Portal, OPC

Tekijä Otsikko Sivumäärä Aika	Tommi Piiparinen Logiikkapohjaisen prosessiautomaation säätö- ja mittalaitteen kehittäminen. 24 sivua + 1 liite 14.3.2014
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Automaatiotekniikka
Suuntautumisvaihtoehto	Kappaletavara-automaatio
Ohjaajat	Principal Scientist Marjatta Piironen Lehtori Heikki Saarelainen

Tämä insinöörityö tehtiin Kemira Oyj:lle, joka on maailmanlaajuinen kemianalan yritys. Insinöörityön tarkoituksena oli kehittää logiikkapohjainen prosessiautomaation säätö- ja mittalaite ja tutkia mahdollisia keinoja helposti muokattavan kommunikaatiorajapinnan luomiseksi logiikan ja tietokoneen väliseen kommunikaatioon. Insinöörityö myös esittelee prototyypissä käytettäviä tekniikoita kuten liikkeenohjausta ja PID-säätimiä.

Insinöörityö tehtiin 1.10.2013 – 30.3.2014 välisenä aikana Kemiran tutkimuskeskuksessa Espoossa. Kaikki prototyypin laitevalinnat oli tehty ennen insinöörityön aloittamista. Prototyyppi rakennettiin, ohjelmoitiin ja testattiin tutkimuskeskuksen verstaalla.

Insinöörityön tuloksena saatiin toimiva prototyyppi logiikkapohjaisesta mittalaitteesta sekä suhteellisen helposti muokattavissa oleva kommunikaatiorajapinta. Yrityksen sisäiseen käyttöön tuotettiin myös kattava dokumentaatio luodusta logiikkaohjelmasta sekä kommunikaatiorajapinnasta.

Avainsanat	PLC, PID, TIA Portal, OPC

Contents

Acronyms

AWP — *Automation Web Programming*. Special commands that allow reading and writing of Siemens PLC memory values from HTML pages, developed by Siemens.

DLL — *Dynamic link Library*. Executable file that allows programs to share cod and other resources.

HMI — *Human Machine Interface*. User interface in a manufacturing or process control system.

OLE — *Object Linking and Embedding*. Technology that allows linking and embedding to documents and objects, developed by Microsoft.

OPC — *OLE for Process Control*. Software interface standard.

PID — *Proportional-Integral-Derivative*.

PLC — *Programmable Logic Controller*. Computer device used in industrial control systems.

PN — *PROFINET*. Industrial Ethernet standard developed by Siemens.

PSU — *Power Supply Unit*.

SCADA — *Supervisory Control And Data Acquisition*. Computer system used to gather real-time data and analyze the collected data.

TCP — *Transport Control Protocol*. Protocol used to send data as messages over the internet.

TFT — *Thin Film Transistor*. A type of Liquid Crystal Display screen.

UPS — *Uninterruptible Power Supply*.

1 Introduction

1.1 Kemira Oyj

Kemira Oyj, the commissioner of this thesis, is a global chemistry company headquartered in Helsinki Finland. Kemira focuses on providing sustainable chemistry solutions for pulp & paper, oil & gas, mining and water treatment industries. Kemira employs almost 5000 people globally in 40 countries and had revenue of 2.2 billion euros in 2012. [1]

Kemira's Control and Monitoring Tools are built on Kemira's own and commercial technologies to extend Kemira's product application knowledge. These tools include modern online measurement and control systems with strong visual data analysis tools to help analyze large amounts of data they produce. [2]

Kemira has four Research & Development Centers, which are all part of the Kemira Center of Water Efficiency Excellence (SWEET). They are located in Espoo (Finland), Atlanta (USA), Shanghai (China) and São Paulo (Brazil). Kemira used 1.7 % of the group's revenue on research and development in 2012. This thesis was carried out for the Espoo Research & Development Center, focused on developing water treatment solutions for municipal treatment plants and water intensive industries. [3]

1.2 Purpose and Scope

Kemira uses online monitoring systems to control chemical processes. These monitoring systems are either commercially available sensors or specifically developed analyzers. Integrating new analyzers with common device environment is often challenging as different subcontractors may use various types of device manufacturers.

The purpose of this thesis is to develop a new version of a monitoring prototype and to study possible ways to create an easily adaptable communication interface between PLCs and PC software. Siemens PLC technology was selected for this prototype. The new version's PLC program, communication library between the PLC program and the analysis software and Siemens HMI visualization will be implemented in this thesis. The scope of the thesis and the can be seen in Figure 1.

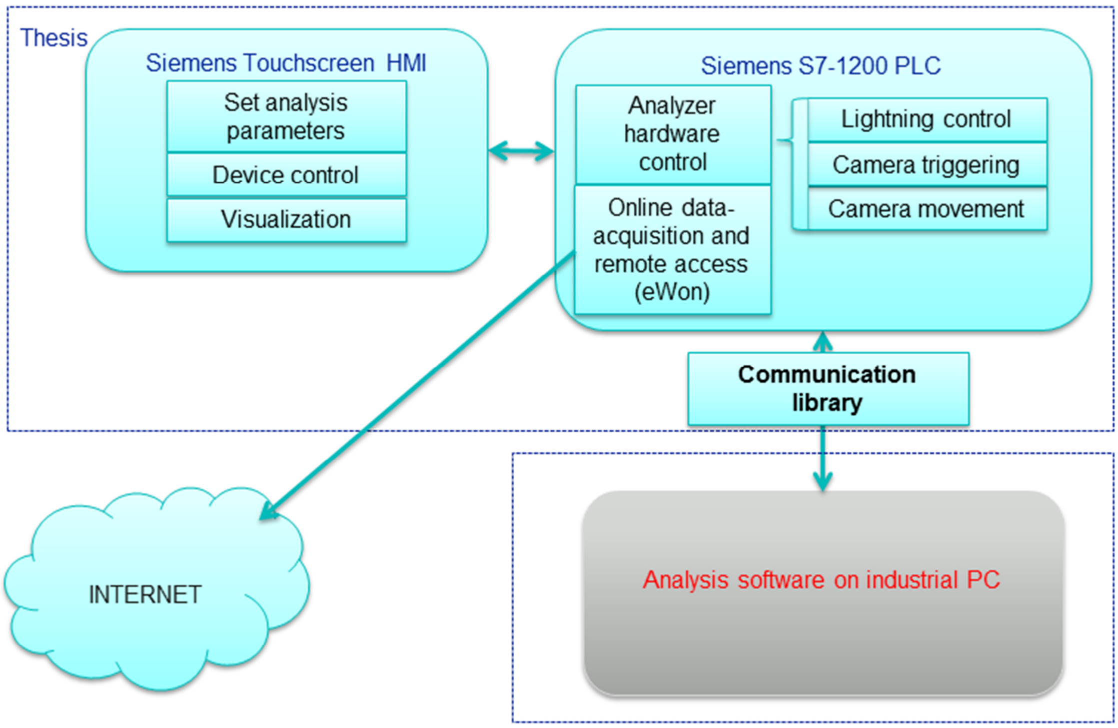

Figure 1. Scope of the Thesis

This thesis will focus on the different software implementations (PLC program, Communication library, HMI visualization) and techniques used to create the software. The development project was conducted during 1.10.2013 – 30.3.2014. All hardware for the prototype was selected before the start of the thesis.

1.3 Description of the Prototype

The prototype uses sensors and a camera to measure process values from two identical separate measuring lines. Images from the camera are analyzed by analysis software running on the industrial PC and the analysis results are transferred to the PLC's data block for data acquisition and visualization on the HMI panel. The camera is mounted on a linear guide powered by a stepper motor, which moves the camera between multiple imaging locations.

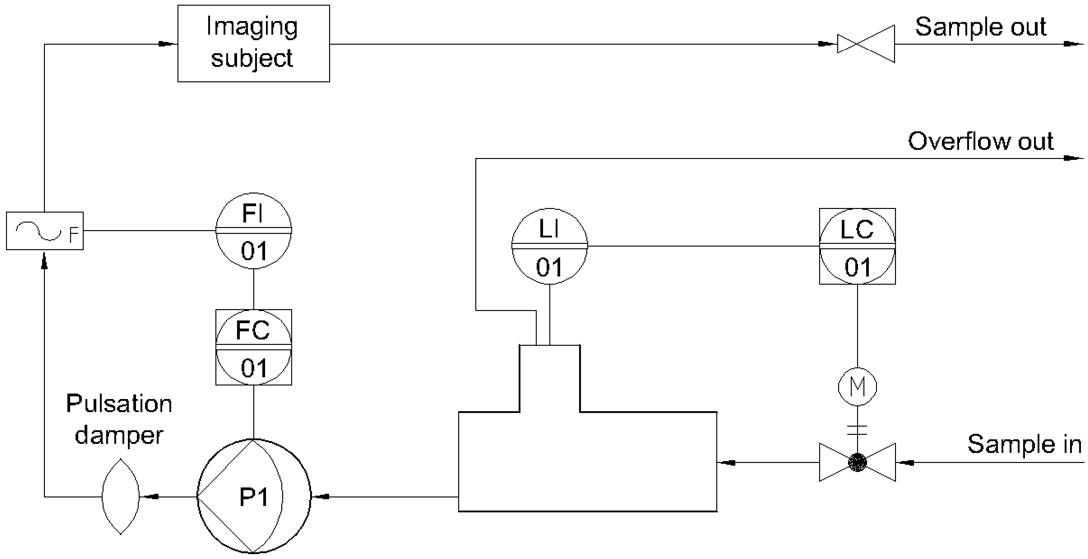

Figure 2. PI diagram for a single measuring line of the prototype

The PI diagram for a single line is described in Figure 2. The sample enters through an electrically actuated ball control valve to the sample tank. The ball control valve is controlled by a PID controller, which has feedback from the ultrasonic level sensor. The diaphragm pump, controlled by a PID controller taking feedback from the ultrasonic flow meter, then pumps the sample through the imaging subject and back pressure valve out of the enclosure.

The prototype is built into three stainless steel enclosures. A large enclosure at the bottom with two smaller enclosures that are mounted side by side on top of the larger enclosure. The PLC, motor driver, HMI panel and related equipment are mounted into the top left enclosure. The UPS system, power supplies and related equipment are mounted into the top right enclosure. Analysis hardware is mounted to the bottom enclosure and can be seen in Figure 3, also the HMI panel mounted to the top left enclosure is visible.

Figure 3. The finished prototype

Equipment used in the prototype

- Siemens SIMATIC HMI Panel
- Siemens S7-1200 series CPU and signal modules
- Siemens Industrial PC
- eWon industrial router

- Power supplies and UPS
- Camera
- Stepper motor driver
- Stepper motor
- Linear guide
- Two pressure difference sensors
- Two ultrasonic flow meters
- Two ultrasonic level sensors
- Two diaphragm pumps
  - With pulsation dampers and back pressure valves
- Two ball control valves with electronic actuators
- Two custom built sample tanks

2 Background

2.1 Siemens SIMATIC S7-1200 Series PLCs

The Siemens SIMATIC S7-1200 series PLCs are compact modular microprocessor based programmable logics intended for low to medium range device automation tasks, such as conveyor systems, packaging machines or wood handling machines. They contain ready to use PID-controllers and motion controlling technologies with commissioning control panels to simplify the startup and commissioning. [4] [5]

S7-1200 series PLCs are programmed, configured and engineered with Siemens Totally Integrated Automation Portal (TIA Portal). The TIA Portal integrates SIMATIC STEP 7, SIMATIC WinCC and SINAMICS StartDrive seamlessly into a single engineering framework. The SIMATIC STEP 7 integrated into the TIA Portal allows the user to program, configure, test and diagnose all SIMATIC controllers. Integrated SIMATIC WinCC makes it possible to program HMI visualizations from basic HMI panels all the way up to complete SCADA solutions. [6]

2.2 Siemens SIMATIC HMI Panels

The Siemens SIMATIC Human Machine Interface Panels are programmed in the TIA Portal with the integrated SIMATIC WinCC software. The SIMATIC HMI Panels have touch screens or control keys or both. Four different series allow the user to select the panel best suited for his/her requirements. The differences between series can be seen from Figure 4. The Comfort Panels are intended for demanding visualization purposes and the Basic Panels for cost effective entry level simple HMI applications. The Mobile Panels are highly protected against dust and splash water and intended for mobile use, the panels can show different user interfaces according to the point where they are connected. The Key Panels have large mechanical buttons that are easily labeled and integrated digital inputs and outputs to allow the connection of peripheral devices. [7]

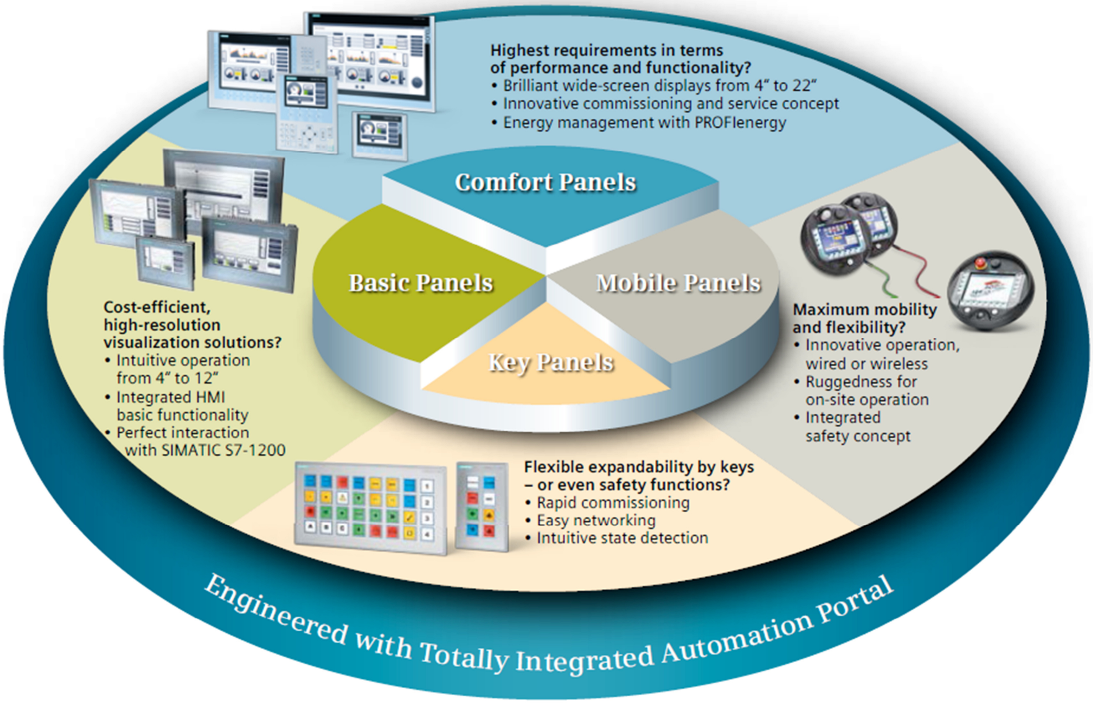

Figure 4. Siemens SIMATIC HMI Panels [7]

2.3 OPC and Data Access 3.0 Custom Interface

OPC Foundation provides freely accessible technical specifications that define standard sets of interfaces that allow data to be exchanged between different software. OPC is not manufacturer dependent and most of the PLC manufacturers have OPC compatibility. The current version 3.0 of the Data Access specification, henceforth OPCDA, defines an interface for client and server programs to access process data. The Custom interface is used in programming languages that support function pointers, in this case C++. [8]

2.3.1 Data Access Server

Data Access Servers provide clients access to different data sources. These data sources can be for example different manufacturer’s PLCs and their memory or different sensors connected directly or indirectly to the computer running the Data Access Server. [8]

2.3.2 Data Access Client

Data Access Clients use Data Access servers to access data exposed by the server. The client does not need to know how the server is connected to a data source. Clients can be virtually anything e.g. visualizations, simple excel sheets or parts of larger programs. The clients control themselves how they see the data provided by the server, by creating different OPC objects (Groups, Items) in the server they are connected to. [8]

2.4 Snap7 Ethernet Communication Suite

During the thesis alternative ways to communicate between the PLC and PC station were also examined. One of the found solutions is Davide Nardella’s Snap7 open source Ethernet Communication Suite. It is designed to overcome OPC’s limitations for large high speed data transfers. It is platform independent and supports 32 and 64 bit platforms. Snap7 is comparable to the OPCDA custom interface except that it does not require an OPCDA server to operate and is only compatible with Siemens PLCs. [9]

2.5 C++ Programming Language

C++ is a general purpose programming language with support for data abstraction, object-oriented programming and generic programming. It is derived from the programming language C by Bjarne Stroustrup and standardized by The International Standards Organization (ISO/IEC 14882). [10]

2.6 Dynamic Link Library

Dynamic Link Libraries are libraries that contain code and data and are intended to help using the same code again and use less memory. With DLLs a program can be split into multiple components and this makes it easier to update just one part of the program without the need to compile the whole program again. Largest advantages of DLL include modular architecture, easy deployment and installation and they also use fewer resources when used by more than one program. [11]

The communication library is compiled into a DLL which makes it possible to update the library without the need for the subcontractor to recompile the whole analysis program.

2.7 PID Controller

Proportional-Integral-Derivate controllers are used to control process values. They calculate an output value from the error between the current setpoint for the process value and the current measurement of the process value. The control value is a sum of three different terms, the proportional term P, the integral term I and the derivative term D. Controllers can also consists from different combinations of the three terms, for example P, PI or PD. In this case only the used terms are summed to produce the output value. The output value is used to control actuators such as pumps or valves to control process values. [12]

The proportional term P, also called gain, multiplies the error with a set constant, producing a proportional output value relative to the current error value. Its purpose is to remove the error between the setpoint and process value. If the controller consists only of the P term, it may leave a permanent constant error. The integral term I, is proportional to the magnitude and duration of the error. It is used to remove the permanent error remaining after the proportional term. The derivate term D is the derivate of the error value and its purpose is to speed up the controller's reaction to the error value. [12]

The controller needs to be tuned for different kinds of processes. The tuning is always performed for a specific operation point. The most basic tuning principle is called a step response test. It is conducted by creating a step change in the process while the controller is not used to control the process and measuring the change. The PID parameters are then calculated from the measured process reaction to the change. [12]

3 Communication library

3.1 Overview

The communication library is an OPCDA client created specifically to provide the analysis software with synchronous read and write access to the PLC's memory. It also utilizes OPCDA's OnDataChange callback in order to monitor the PLC tags that are used to signal the analysis software that it should begin or cancel operations. The communication library is made available for the analysis software as a Windows 64 bit DLL written in C++ and compiled with Microsoft Visual Studio Express 2012.

The analysis software requests the connection status from the communication library which then tries to establish connection to the PLC. The connection is then active until the analysis software is closed and provides access to various PLC memory variables for the analysis software via multiple functions.

3.2 Requirements for usage without modifications

3.2.1 PC Station

The communication library requires that the 64 bit Microsoft Windows computer in which it is used has Siemens SIMATIC NET OPC server installed and running locally.

In the TIA portal the PC Station should have OPC server configured to slot 1 and IE General to slot 2. IE General should be set to “Access via OPC server” in the OPC configuration.

3.2.2 PLC Station

PLC Station has to be named “S7-1200 station_1” and the PLC must be named “PLC_1”. It is also important that the connection between the OPC server and the PLC_1 is named “S7_Connection_1”. OPC tags should be configured to include all tags in the “ImageAnalysis” [DB7] datablock.

The PLC station must have a datablock [DB7] which can be accessed symbolically as “ImageAnalysis” and that it contains all the tags configured exactly as in Appendix 1. The datablock must not use the “Optimized block access” option found from the block properties.

4 PLC Program

4.1 Overview

The PLC program is used to control the hardware of the analyzer. It also has a data block used for online data-acquisition via a eWon router, that sends the data in the data block to a server. The controlled hardware consists of 2 control valves, 2 pumps and one linear guide driven by a stepper motor, camera and a LED ring light. An open user communication using the TCP protocol is also used to transmit eWon connection settings from the PLC to the eWon router.

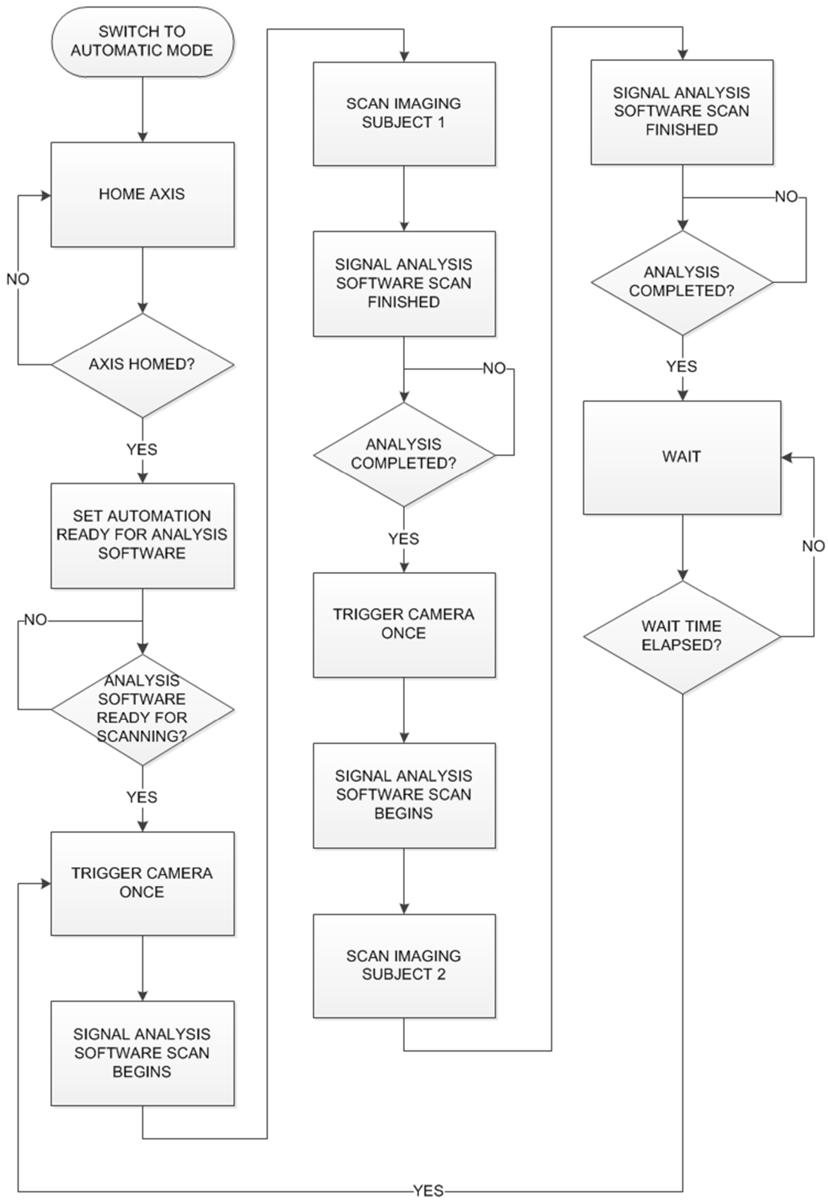

Figure 5. Simplified flowchart presentation of the automatic operation sequence

Figure 5 shows the simplified flowchart presentation for the automatic operation sequence. When the automatic operation is enabled the axis is first homed and then the analysis software is signaled that automation is ready, the automation then waits until the analysis software signals that it is ready. When the analysis software is ready the camera is triggered once in order to wake it up, as it was found out that during longer idle periods the camera won't trigger properly the first time it is triggered after a period of inactivity. Then the analysis software is signaled that a scan is beginning and then the imaging subject 1 is scanned and after the scan has been finished the analysis software is signaled that the scan has been completed.

The analysis software then analyses the images captured and writes the results of the analysis to the PLC. The imaging subject 2 is then scanned in a same manner as the imaging subject 1 starting with the “wakeup” triggering for the camera. After the analysis results for imaging subject 2 are received the PLC program waits for a user set time and then starts the scanning sequence from the beginning.

4.2 Transferring eWon Connection Settings

The original idea for transferring the eWon connection settings was to use the PLC’s web server and AWP commands to make a configuration file accessible for the eWon, which would then download the file and apply the connection settings. When the AWP commands are used the Siemens web server adds an empty line to the beginning of the file, which causes the file to be unreadable by the eWon’s basic programming language. When AWP commands are not used the empty line does not appear and the eWon’s basic programming language can read the file. The added empty line is marked in yellow in Figure 6.

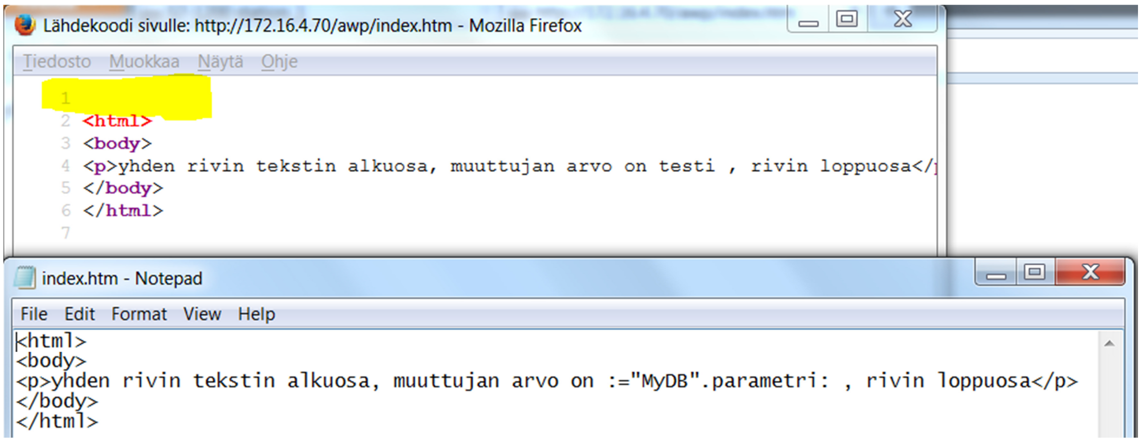

Figure 6. File generated by the Siemens web server showing the added empty line

The open user communication using TCP protocol is configured in the TIA Portal so that the PLC is the passive communication partner and the eWon is the active partner. The passive communication partner waits until the active partner opens the communication. The connection configuration is visible in Figure 7. Each time the eWon is rebooted it opens a TCP connection to the PLC and sends character “1” in order to request for the connection settings data to be sent from the PLC to the eWon.

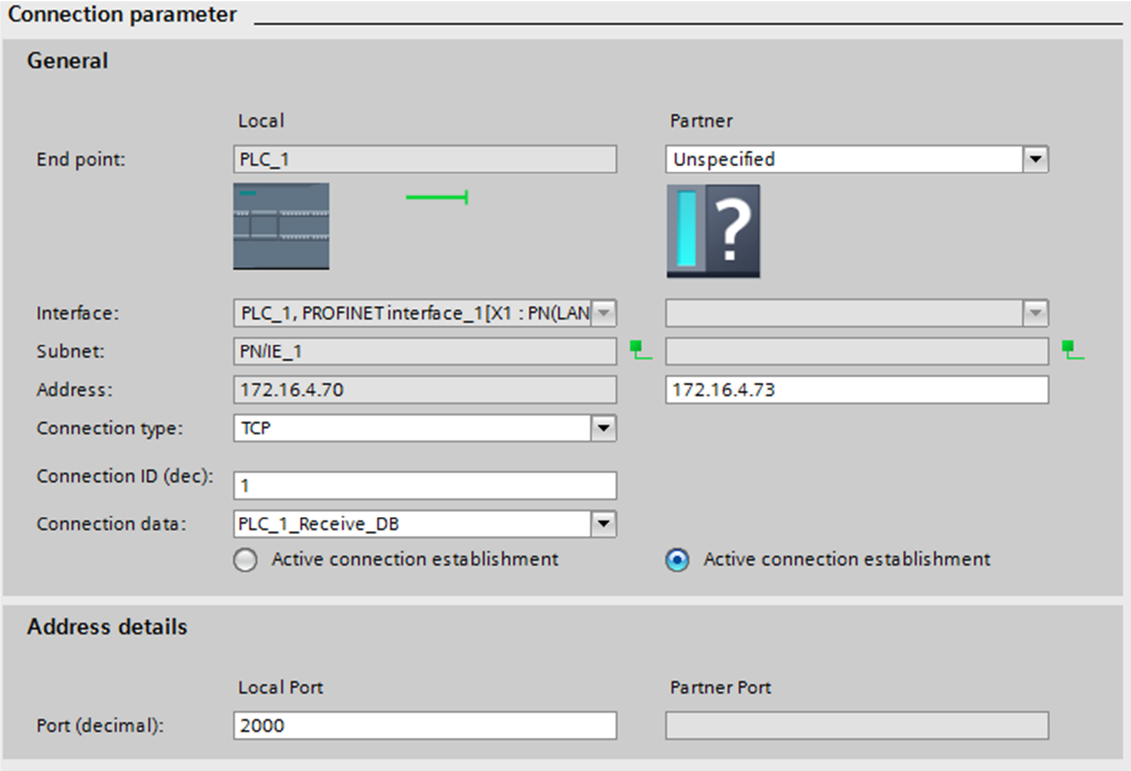

Figure 7. eWon TCP connection configuration in TIA Portal.

Two networks that are responsible for transmitting the eWon connection settings are visible in Figure 8. TRCV_C_ function block in Network 5 is always enabled and receives the data sent from eWon and saves it to the local static char variable #SendEwonReq. In network 6 the received character is compared if it is equal to the character ‘1’ and if it is equal the TSEND_C_ function block sends the eWon connection settings to the eWon. The connection settings are concatenated into a single formatted string from multiple variables and the string is then passed for TSEND_C_ function block.

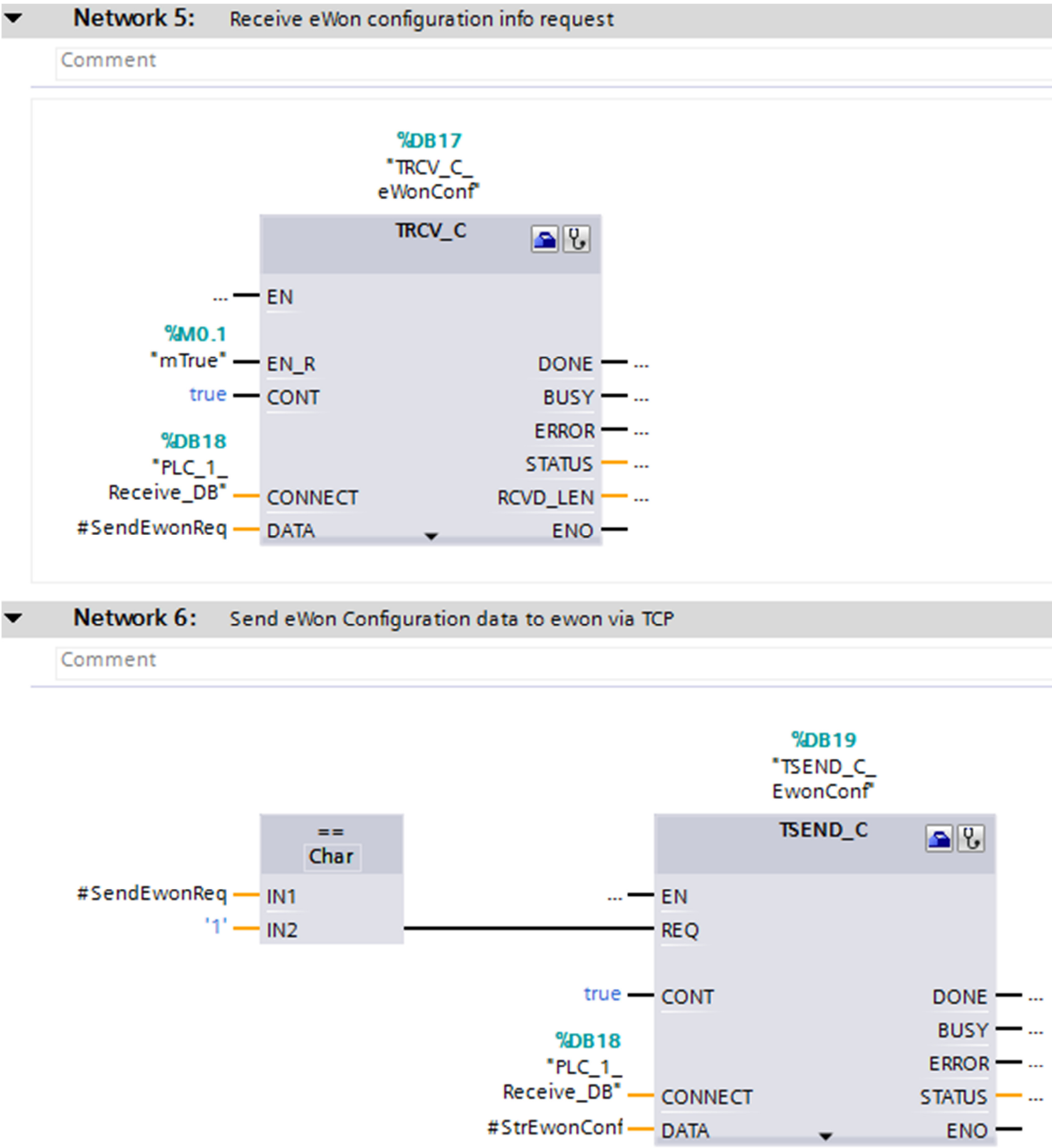

Figure 8. PLC program for transmitting the eWon connection settings

4.3 Camera Motion Control

The camera's motion is controlled with Siemens's built in motion control instructions and technology objects. Parameters for the physical drive are configured in the axis technology object. When the device is switched from manual to automatic mode the program homes the axis actively and then waits until the analysis software is ready. After the analysis software signals that it is ready for scanning the camera movement is controlled via two command tables, one for each imaging subject, multiple images are taken of each subject.

The camera is triggered by using the command step code parameter, which can be assigned individually for each command in the command table. The "MC_CommandTable" motion control instruction outputs the step code for the current command in execution. After each move between imaging locations a wait command with a step code of 2 is used to stop the camera at the imaging location and trigger the camera.

4.3.1 Technology Object – Axis

The Axis technology object is used to represent, control and configure the software and hardware used to control the axis. All the parameters for the drive interface, mechanics and position monitoring are configured in the Axis Technology object. Most important parameters are the hardware interface and the mechanics which are used to define how much pulses are required for one motor revolution and the distance moved per motor revolution. Figure 9 describes how the Axis technology object's software and hardware components interact. [13]

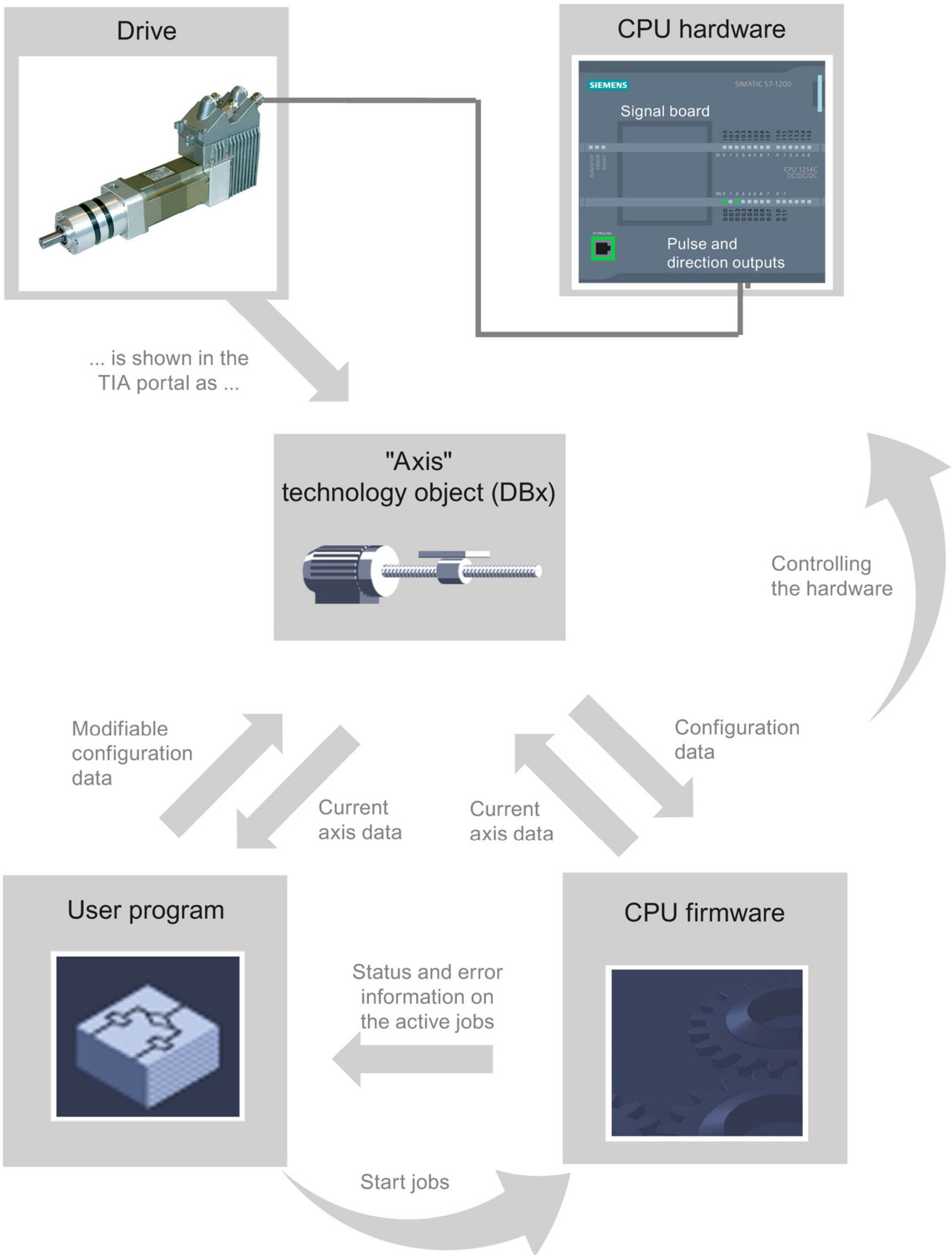

Figure 9. Hardware and software components of the Axis technology object [14]

4.3.2 Homing the Axis

Homing is matching the coordinates of the Axis Technology object to real physical location of the axis, this is extremely important in position-controlled axes. “MC_Home” motion control instruction is used to home the axis. In this application the axis is homed actively and the “MC_Home” motion control instruction moves the axis to left until it detects the homing switch, decelerates and moves back to the position of the homing switch.

4.3.3 Technology Object – Command Table

Command table technology object provides means to combine multiple motion control instructions into one movement sequence, a maximum of 32 instructions can be entered for a single command table. For each instruction a command type is selected and additional parameters such as a position, velocity or duration specified. The command table is executed with a “MC_CommandTable” motion control instruction. The use of command tables requires that the hardware has firmware version of V2.0 or greater. [13]

4.4 Level and Flow PID Controls

The tank levels for both lines and the flow rate of both lines are controlled with Siemens’s built in PID control instructions. As both the pumps and valves are controlled with an analog control signal the PID_Compact instruction is used.

4.4.1 PID Control Technology Objects

S7-1200 has two different PIDT1 control instructions, PID_Compact, that serves as a universal PID controller and PID_3Step, which is used for Boolean outputs. Both instructions have automatic tuning abilities and technology objects for configuring parameters. The instructions have to be called in a cyclical interrupt organization block and each instruction must have a unique technology object to operate properly. [14]

4.4.2 Automatic Tuning

Pretuning performs a jump change in the output value of the controller and calculates PID parameters from the maximum slope and dead time of the controlled system [15]. Fine tuning usually produces better PID parameters than pretuning, while fine tuning is being processed a constant oscillation of the process value is generated and the PID parameters are calculated from the amplitude and frequency of the generated oscillation. [16]

5 User Interface

The Siemens SIMATIC KTP 600 Basic Color PN touchscreen user interface is used to control the device, configure eWon connection settings, set analysis parameters and visualize the current status of the analyzer. The user interface was created with Siemens's WinCC included in the Totally Integrated Automation Portal.

While the touchscreen user interface was being developed it was found out that the text size would change when the interface was loaded from the engineering station to the HMI panel and partially making some texts unreadable. This behavior was fixed by setting the engineering stations text size from windows settings from "normal" 120% to "smaller" 100%, recompiling the user interface and loading it to the HMI panel. The problem was caused by the difference of the text size setting between the engineering station and the HMI panel which runs Windows CE and has the "smaller" setting as default.

6 Testing

6.1 Communication Library

The communication library was first created and tested in a mockup setup including only the PC, dummy analysis software and the PLC which only had the "ImageAnalysis" [DB7] data block. The dummy analysis software utilized all of the communication library's functions in the same way as the real analysis software uses them, so the communication library was considered completed when the dummy software worked flawlessly with the communication library. A few small additions and bug fixes were made to the communication library during the testing of the entire prototype.

6.2 PLC Program

The state machine was first tested by simulating different tags via the TIA Portal's force table. After all the transitions between different steps were found to work as planned the state machine was considered finished and further testing begun.

6.2.1 Motion Control

The PLC motion control was tested after the motor, driver and linear guide were mounted into the prototype. The motor was first tested by using the built in commissioning panel in the TIA Portal. After the command tables had been configured and tested so that their movement instructions covered all of the imaging subjects the motion control was considered completed.

6.2.2 Flow PID Controllers

The PID controllers used to control the flow were tested by using the TIA Portal's commissioning panel. First the pump was tested by setting a manual value of 50% to the controller which started the pump. The flow created by the pump was verified by the flow meter and also by measuring the amount of water pumped into a measuring cylinder during one minute.

Automatic pretuning for the controller was carried out from the commissioning panel and the controller switched to automatic mode. Automatic fine tuning was also carried out to further improve the accuracy of the controller. After both tunings were completed the parameters were uploaded to the TIA Portal project in order to avoid the need to commission each identical device separately.

The PID controllers were finally tested by setting a set point for the flow value from the HMI panel and again measuring the amount of water pumped into a measuring cylinder during one minute.

Occasionally the ultrasonic flow meters produced unreliable measurements, which caused the pumps to pump at full power until the measurements would produce reliable results again. When the flow meter problem was further investigated by running water through the flow meters directly from a water spigot without using the pumps the flow meters constantly produced reliable measurements.

While left unconfirmed at this time it seems that the unreliable measurements were caused by the pumps and the pulsating flow they produced. Even though the pulsation dampers greatly reduced the pulsating of the produced flow it seems that they did not reduce the pulsation enough.

6.2.3 Tank Level PID Controllers

Tank level PID controllers were first tested by setting a manual value of 0% which ended up opening the valve completely, as it was desired that a value of 0% would close the valve completely the actuators were dismounted from the valves and the valves were manually turned 90° in order to achieve the desired result. After remounting the actuators the valves were then controlled to 100%, which resulted in the valves opening completely and producing the desired reaction to the control signal.

While trying to run the automatic pretuning for the controllers it was found that the actuators are too slow. When the controller opens the valve completely to raise the water level in the tank to meet the setpoint the actuator is then too slow to close the valve when the water level closes in on the setpoint causing it to overshoot the setpoint and overflow the sample tank.

Setting the maximum output of the PID controller to 25% solved the problem as the controller can then only open the valve 25% at maximum, effectively making the valve react faster. As the valve opening value of 25% is still sufficient to let more water in the tank than the pump can pump out of it while the pump is running at full power the tank can still be filled.

6.3 Entire Prototype

While testing a scenario where the power supply to the device is lost it was found that if the PC was not shut down correctly it would not start again but instead enter an error recovery screen. This problem was remedied by setting the current boot status policy of the PC to ignore all failures and attempt a normal restart always by running the following command in a console with admin rights.

```
bcdedit /set {current} bootstatuspolicy ignoreallfailures
```

In overall the prototype worked reliably during the relatively short testing period at the workshop. The camera’s motion control proved to be accurate and reliable. The communication interface enabled the communication between the PLC and the analysis software and reliably transferred information between the software. The sample tank level PID controllers managed to keep the water levels in the tanks at the setpoint. Only the flow PID controllers were problematic as the ultrasonic flow measurements produced unreliable results from time to time. The user interface was tested by operating the prototype from the touch panel and some small changes were made to enhance the user experience.

7 Further Development Suggestions

7.1 Increase Reliability of the Flow Measurements

In order to increase the reliability of the flow meters either the pulsation of the flow must be reduced or the flow sensors changed to ones that are not affected by the pulsation of the flow. In order to eliminate the pulsation of the flow the pumps could be changed to pumps that do not produce pulsating flow. Also other possible ways to reduce the pulsation of the flow should also be investigated further.

7.2 Change the Communication Library to Use Snap7

By rewriting the communication library with Snap7 it is possible to reduce costs when compared to the Siemens SIMATIC NET OPC Server. Snap7 is open source and the Siemens SIMATIC NET OPC Server requires a bought license for each device. In my opinion Snap7 is also easier to program than OPCDA custom interface. This change was not carried out during this thesis due to the limited time available. On the other hand OPC has ready support for monitoring data changes whereas Snap7 does not have. Snap7 would also require the data change monitoring to be programmed also.

8 Summary

During the thesis a new fully functional PLC based monitoring device was constructed and programmed. The monitoring device was tested in workshop conditions for a limited time and found to work reliably disregarding the problems related to the flow measurements. Extensive documentation of the PLC program and communication library was also created for internal use by the company. The thesis also introduces the reader to different technics used in the prototype.

Approximately 50 % of the time used for this thesis was used to program the communication library, especially in the beginning a lot of time was spent on learning the OPC Data Access custom interface. As the communication library's source code is thoroughly commented it can be fairly easily adapted to be used for different analysis software or if the analysis software is updated. In my personal opinion, OPC Data Access custom interface is a lot more confusing than Snap7 and requires more programming knowledge and understanding. The usage of a commercial OPC client software development kit could possibly have speeded up the communication library programming.

The Siemens's readymade motion control instructions and PID controllers and their commissioning and configuration panels in TIA Portal made developing the PLC program quick and easy. The HMI visualization was modified from an existing visualization in order to speed up the development and to retain a uniform user experience with other monitoring devices of the company.

The objectives set for the thesis were achieved as the device worked reliably and the communication library is fairly easy to adapt for future use in different devices or with different software. Parts of the source code, such as connecting and disconnecting to the OPC server can be used directly for new appliances. Rewriting the communication library to use Snap7 would make the adaptation process simpler for people with little or no programming experience. The usage of Snap7 would restrict the PLCs compatible with the communication library to PLCs manufactured by Siemens.

References

1 Kemira Oyj, 2013. About us – Kemira. Available at: http://www.kemira.com/en/about-us/Pages/default.aspx [cited 26 October 2013].

2 Kemira Oyj, 2013. Control and monitoring technologies – Kemira. Available at: http://www.kemira.com/en/industries-applications/Pages/control-monitoring.aspx [cited 26 October 2013].

3 Kemira Oyj, 2013. Research Centers – Kemira. Available at: http://www.kemira.com/en/about-us/research-and-development/research-centers/Pages/default.aspx [cited 28 February, 2014].

4 Siemens AG, 2013. Siemens – S7-1200. Available at: http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/automaatiotekniikka/ohjelmoitavat_logiikat_simatic/s7_1200.htm [cited 27 October 2013].

5 Siemens AG, 2013. S7-1200 CPUs – PLC –Siemens. Available at: http://www.automation.siemens.com/mcms/programmable-logic-controller/en/simatic-s7-controller/s7-1200/cpu/Pages/Default.aspx [cited 27 October 2013].

6 Siemens AG, 2013. TIA Portal Brochure. Available at: http://www.siemens.fi/pool/products/industry/iadt_is/tuotteet/tia_portal/tia_portal_brochure.pdf [cited 19 February 2014].

7 Siemens AG, 2013. SIMATIC HMI Panels. Available at: http://www.automation.siemens.com/salesmaterial-as/brochure/en/brochure_panels_en.pdf [cited 7 March 2014].

8 Frank Iwanitz & Jürgen Lange, 2006. OPC Fundamentals, Implementation and Application. 3rd edition. Hüthig GmbH.

9 Davide Nardella. Snap7 Homepage, Overview. Available at: http://snap7.sourceforge.net/overview.html [cited 22 December 2013].

10 Bjarne Stroustrup. A Tour of C++. Available at: http://stroustrup.com/3rd_tour.pdf [cited 24 February 2014].

11 Microsoft, 2007. What is a DLL?. (4 December 2007) Available at: http://support.microsoft.com/kb/815065 [cited 23 February 2014].

12 Jari Olli, 2011. Control Engineering [Lecture Material] Helsinki University of Applied Sciences, unpublished.

13 Siemens AG, 2013. S7-1200 Motion Control V12 SP1. (10 September 2013) Available at: http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&objId=80384402 [cited 2 January 2014].

14 Siemens AG, 2013, S7-1200 Easy Book. Available at: http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=en&objid=34612486&subtype=133300&caller=vie [cited 2 January 2014].

15 Siemens AG. Pretuning. TIA Portal Information System. [cited 14 February 2014]

16 Siemens AG. Fine tuning. TIA Portal Information System. [cited 14 February 2014]

Configuration of the ImageAnalysis Datablock

APPENDIX REMOVED DUE TO COMPANY REQUEST

APPENDIX REMOVED DUE TO COMPANY REQUEST