YILDIRIM BEYAZIT UNIVERSITY
GRADUATE SCHOOL OF NATURAL SCIENCES

PLC AND SCADA SYSTEM APPLICATION IN A CONSTRUCTION CHEMICALS MANUFACTURING PLANT

by

Rıdvan ÖZDEMİR

July, 2015

ANKARA

PLC AND SCADA SYSTEM APPLICATION IN A CONSTRUCTION CHEMICALS MANUFACTURING PLANT

A Thesis Submitted to the

Graduate School of Natural and Applied Sciences of Yıldırım Beyazıt University

In Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical and Electronics Engineering, Department of Electrical and Communications Engineering

by

Rıdvan ÖZDEMIR

July,2015

ANKARA

M. Sc. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled "PLC AND SCADA SYSTEM APPLICATION IN A CONSTRUCTION CHEMICALS MANUFACTURING PLANT" completed by RIDVAN ÖZDEMİR under supervision of ASSOC. PROF. DR. HÜSEYİN CANBOLAT and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Hüseyin CANBOLAT

Supervisor

Prof. Dr. Cengiz TAPLAMACIOĞLU

(Jury Member)

Prof. Dr. Fatih V. ÇELEBİ

(Jury Member)

Prof.Dr. Fatih V. ÇELEBİ

Director

Graduate School of Natural and Applied Sciences

To my loving family

ACKNOWLEDGEMENTS

I would like to my eternal thanks to my thesis advisor Assoc. Prof. Dr. Hüseyin Canbolat, I am fully indebted, for his understanding, patience and encouragement and for pushing me farther than I thought I could go.

My sincere thank must also go to the members of my thesis committee; Prof. Dr. Cengiz Taplamacıoğlu, Prof. Dr. Fatih V. Çelebi and Assoc. Prof. Dr. Hüseyin Canbolat, They generously gave their time, guidance, suggestions and feedbacks toward improving my work.

Special thanks must go to Ahmet Gümüşhan. Ahmet has been a great mentor and problem solver for the application of this work. I even do not think about if he is not around and helping me during the work.

I would like to express my gratitude to my parents-in-law Yalçın family, for their unfailing emotional support.

I deeply thank my parents, Pervin and Şükrü Özdemir for their unconditional trust, timely encouragement, and endless patience. It was their love that raised me up again when I got weary. The family of my brother, Furkan Özdemir and Arzu Özdemir and my unborn nephew, have also been generous with their love and encouragement.

Finally, I thank with love to my wife and life partner Ayşenur Özdemir and my four months baby Elif Mina Özdemir. Beyond being wife, she has been my best friend and a great companion, loved, entertained, supported and helped me survive this crucial period and not let me give up.

July, 2015 Rıdvan ÖZDEMİR

CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

PLC AND SCADA SYSTEM APPLICATION IN A CONSTRUCTION CHEMICALS MANUFACTURING PLANT

ABSTRACT

Industrial automation systems have been securing their position in the sector day by day. PLC-SCADA systems are frequently used in various fields of industry as they have many advantages like cost reducing solutions, standard product output, minimized error rate, availability of receiving real-time statistical and instantaneous data, operation efficiency and automatic control.

In this study a medium sized construction chemical plant is monitored with SCADA system and its automatic/manual control is provide by PLC command system through SDACA. SCADA program used is Reliance Design 4 and PLC program is ABB Control Builder Plus. Computer communication is through Ethernet. The number of total tag used in the system is 68.

This study can be divided into 3 sections. In the first section general information on PLC, its definition, history, structure and programming languages is provided. In the second section SCADA definition, its advantages and areas of usage and structure were our focus in general. In the third and final section general information on construction chemical plants, detailed information on the software and hardware used in the study and how the system works can be found step by step.

As a result with the automation system obtained, the amount of product produced in an hour and a daily slice of the facility, the amount of electricity consumed and number of workers in the facility has has been compared with the predicted values for the system having no automation.

Key Words: PLC, SCADA, Automation, Construction Chemicals Plant

YAPI KİMYASALLARI ÜRETİM FABRİKASINDA PLC VE SCADA SİSTEMİ UYGULAMASI

ÖZET

Endüstriyel otomasyon sistemleri, sektördeki yerini gün geçtikçe sağlamlaştırmaktadır. Maliyet düşüren çözümler, standart ürün çıktısı, en aza indirgenmiş hata oranı, gerçek zamanlı istatistiki ve anlık veri alabilme imkanı, operasyon verimliliği, otomatik kontrol gibi avantajları ile PLC-SCADA sistemleri ülkemizde de endüstrinin birçok alanında sıkça karşımıza çıkmaktadır.

Bu çalışmada orta büyüklükteki bir yapı kimyasalları fabrikası, kurulan SCADA sistemi ile izlenmiş ve PLC ve kumanda sistemi ile de yine SCADA üzerinden fabrikanın otomatik/manuel kontrolü sağlanmıştır. SCADA programı olarak Reliance Design 4, PLC programı olarak ABB Control Builder Plus kullanılmıştır. Bilgisayar haberleşmesi Ethernet ile yapılmıştır. Sistemde kullanılan toplam tag sayısı 68dir.

Bu çalışma genel olarak 4 bölümden oluşmaktadır. İlk bölümde PLC, tanımı, tarihçesi, yapısı ve PLC programlama dilleri hakkında genel bilgiler verilmiştir. İkinci bölümde genel olarak SCADA tanımı, avantajları, kullanım alanları, yapısı üzerinde durulmuştur. Üçüncü ve son bölümde ise yapı kimyasalları fabrikası hakkında genel bilgiler verilmiş, çalışmada kullanılan yazılım ve donanım ayrıntılı bir şekilde tanıtılmış ve sistemin nasıl kurulduğu adım adım anlatılmıştır.

Sonuç olarak elde edilen otomasyon sistemi ile tesiste bir saatlik ve bir günlük dilimlerde üretilen ürün miktarı, tüketilen elektrik miktarı, tesis içerisinde çalışan işçi dağılımları ortaya konulmuş ve manuel sistem için tahmin edilen değerlerle karşılaştırılmıştır.

Anahtar Sözcükler: PLC, SCADA, Otomasyon, Yapı Kimyasalları Tesisi

1. INTRODUCTION

The need for qualified, faultless, and much cheaper production which is required by the current economy has increased the pace of plants' transmission to the automation practices. PLC and SCADA systems are the most commonly used automation systems. All the sectors in industry benefit from automation. Therefore, the pace and quality of the operation and production increases while the number of industrial and occupational accidents decreases.

Pressure, level, flow rate, opacity, rigidity, motor on/off, malfunction, voltage and streamline flow etc. are some of the parameters that can be controlled by the automation processes.

The addition of SCADA to the automation systems provides both the ease of use and the opportunity of being able to monitor all the processing and operation courses on the screen.

A control system within the PLC and SCADA systems is a provision to complete the automation; however, it would not be enough by itself. Another significant point here is the correct selection and analysis of field elements. In other words, it is important that on what logical basis that a level measuring sensor will take data and it is also important to ensure the use of this sensor based on needs or the information on the range of the data transferred from a hydraulic pump is also significant.

2. PROGRAMMABLE LOGIC CONTROLLER (PLC)

2.1 Definition

Programmable Logic Controller (PLC) is a microcomputer system which processes the information received from the sensors in line with the program provided and transmits the results to working elements. It was developed in order to overcome negative aspects of relay control systems. PLC was developed in time and its areas of usage were broadened with various industrial control purposes such as sequence control, movement control (linear and rotary motion control), process control (temperature, pressure, humidity, velocity), data management (data collection, monitoring and reporting about the machine or process).

2.2 History

General Motors which is one of the car manufacturers of American motors in 1960s engaged in using computers for replacing relay sequencing in the control of the automated car plants. Based on this, General Motors generated a specification of an industrial computer.

As a response to the General Motors' specification, Bedford Associates and Allen Bradley manufactured separate systems each of which is nearly a counterpart of the commercial mini-computers of the time.

The central processor which was actually the computer was produced in such a way that it was able to survive in an industrial environment and the connection between this processor and the outside world was established by means of input and output cards that could be plugged. There were four different types of cards for these primitive gadgets, namely; DC digital input card, DC digital output card, AC digital input card and AC digital output card. The most significant thing about the card allocation was that it was totally optional for the users and, this optionality allowed flexibility. Figure 2.1 is an example for this type of system. [1]

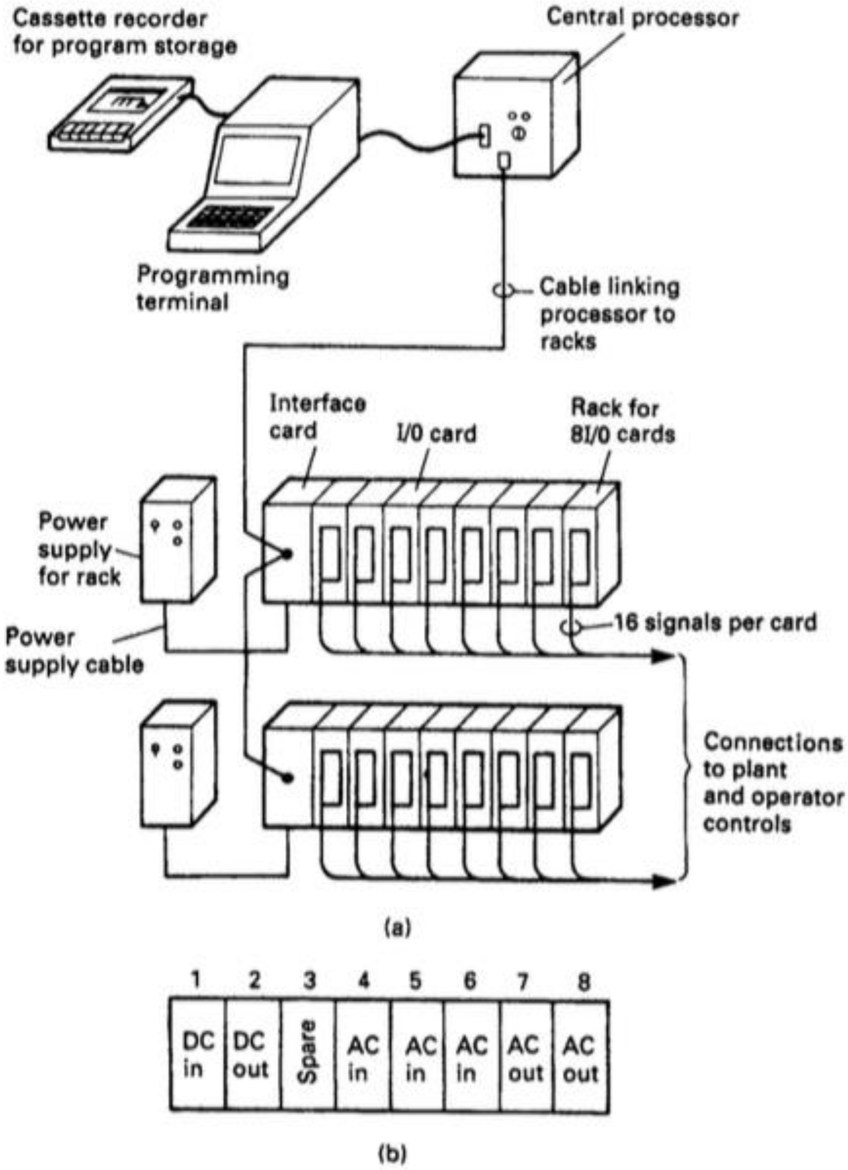


Figure 2.1 (a) Component parts of a PLC system **(b)** A typical rack of cards.

To create a new language of programming taking the relay schematic diagram as basis was quite an original idea. As is seen in Figure 2.2.a, pushbuttons can extend or retract a hydraulic cylinder effortlessly. Limit switches fix the stroke of the hydraulic cylinder and, hydraulic pump should be running in order to ensure the operation of the solenoids. Like the relay circuit which is required to control the cylinder, the computer program of Figure 2.2.b would be responsible to control this process. The reason for naming these programs afterwards as “ladder diagrams” was their resemblance to the rungs on a ladder. [1]

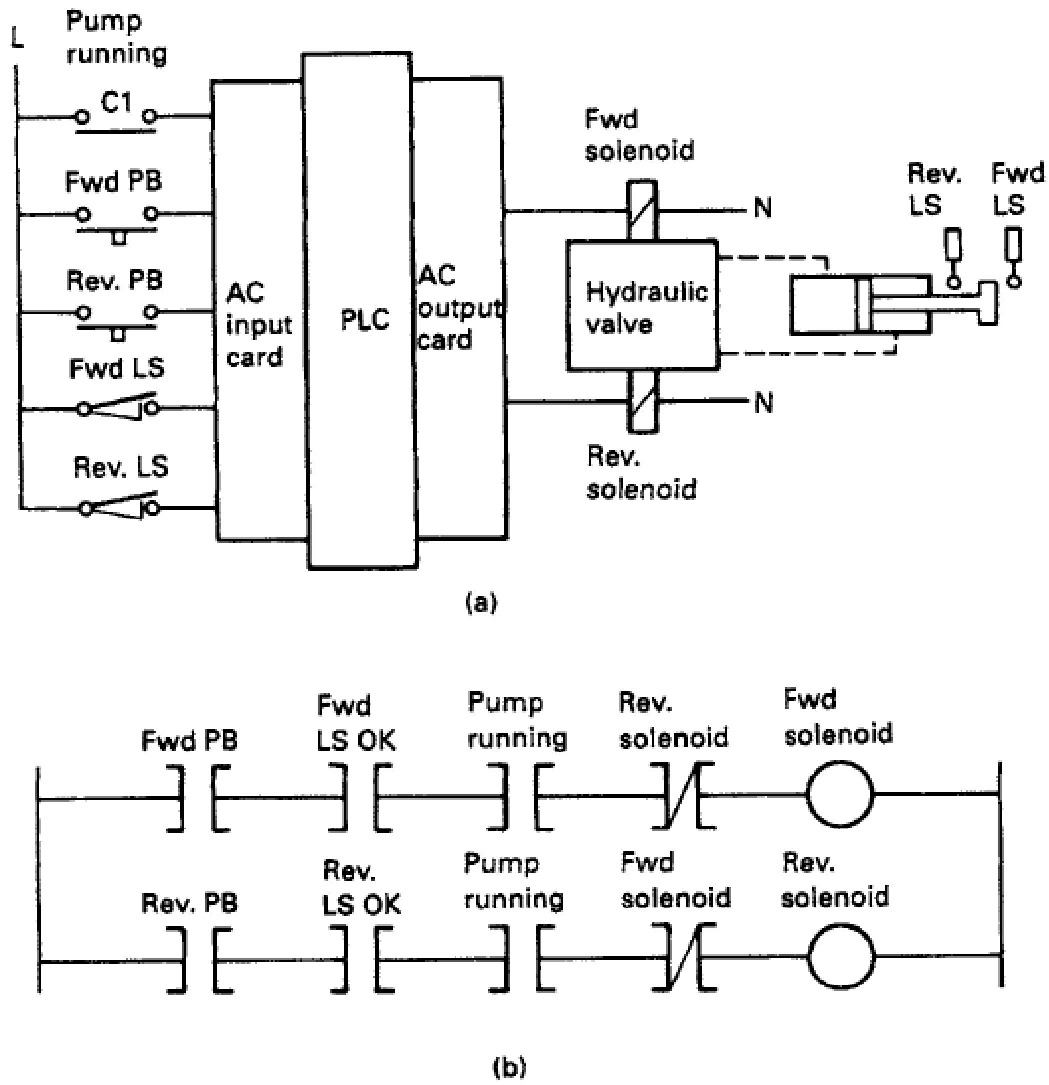


Figure 2.2 (a) A simple hydraulic cylinder controlled by a PLC **(b)** The 'ladder diagram' program used to control the cylinder.

In order to enter the program, a programming terminal with keys was used and these keys were demonstrating some basic relay symbols such as normally open/normally closed contacts, coils, timers, counters, parallel branches etc. The programmer of this primitive gadget is shown in Figure 2.3. The keys should have clear meanings on the program. Thanks to the batteries, the processor memory is protected against any possible loss of program or damage resulting from a failure of power. In addition, it was possible to store the programs on cassette tapes and this was enabling the use of the stored programs for distinctive products. [1]

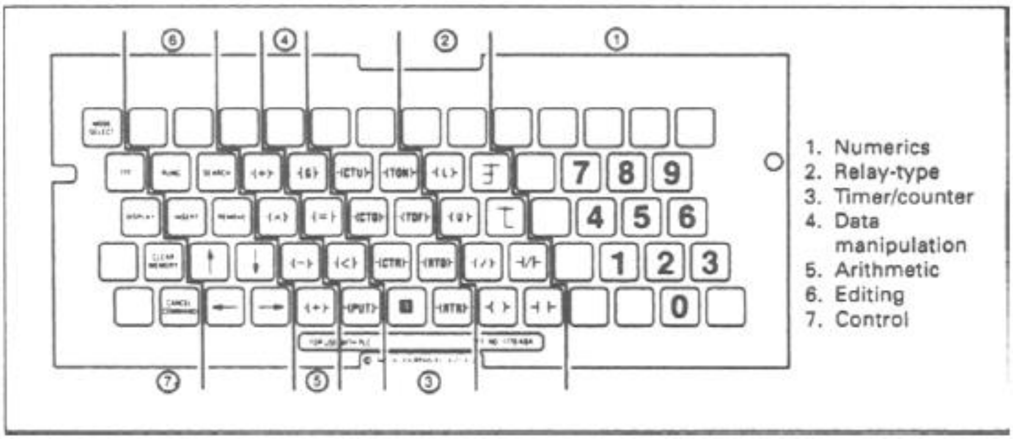


Figure 2.3 The programming terminal keypad for an early Allen Bradley PLC

Consequently, these programs were first called as “programmable controllers” and recently, the letters PC have been frequently used as an abbreviation. However, the name “programmable logic controller” is a registered trademark of Allen Bradley Company.

After it was invented in 1970s, PLC experienced a prompt boost and it took place in some other applications which have identical operation processes. In automotive assembly, the use of PLCs paved the way for their rapid acceptance for the applications such as machinery, packaging, etc. These devices had been used as controllers for a long time. As time went by, there was a search in the market for different applications which were able to ensure more functionality. In addition, the design of PLCs was not suitable for the processing of analogue signals and this was one of the reasons of this search in the market. However, the processing of analogue signals was only achieved with the invention of microprocessors in 1980s.

Through the development of microprocessors, the size reduced and the effectiveness of all PLCs increased. Essentially, the smaller units endured as sequence controllers and it was possible to handle an acceptable part of analogue control by the larger ones by means of other auxiliary facilities. Two different types of PLCs can be mentioned namely; small sized, stand-alone restricted operation PLCs and multi-functioning, much powerful PLCs. For instance; despite being quite cost-effective, a pump controller which is previously stated has limited functions. The smaller and low cost PLC would be perfect in case there is a need for local stand-alone control, which requires limited information or even no information. However, multifunctioning PLCs are the best solution for the need for remote control and for complicated control needs.

2.3 The Advantages of PLC Control

PLCs have many advantages in terms of physical size, cost, environment tolerance, communication capability, complex structure, ease of programming, capacity increase, speed of processing and monitoring functions. There are four phases of control namely; design, construction, installation, commissioning and maintenance as an additional phase to the 4 phases. Considering the four phases through which any kind of control system goes, PLC manifests its advantages in each of these four phases.

2.3.1 Design

Design is the first phase. In this phase, the primary step to be taken is to study on the required plant and to determine the control strategies. Prior to the construction, the design phase should be completed. Along with the use of a PLC system in this phase, there is only requirement of a rough estimate on the size of the machine and the information of I/O, the number of inputs and outputs. Taking into account the inevitable omissions, it is possible to set a healthy spare capacity with the input and output cards. Also, compared to their competences, PLCs are quite small and less space occupying devices, enabling smooth use of in any type of environment.

2.3.2 Construction

Construction is the next phase. The program is written in the course of this phase. A PLC system is composed of certain parts which are assembled to function together. Therefore, they are able to control a large number of machines simultaneously via the programs of each part which are stored in the memory. Also, due to being specifically designed for industrial environments, PLCs are resistant in these environments. PLCs are competent to operate smoothly under high and low temperature without being affected from vibrated and loud environments.

2.3.3 Installation

Installation is the following phase. Here, the cabling of operator controls is carried out and the control of limit switches, sensors are made. In order to prevent huge costs, a PLC system with previously established and checked desks can be used. Moreover, PLC systems are cost-effective both in terms of primary investment costs and the production surplus. Another point is that because less number of electrical equipment and cables are used, the operating cost decreases.

2.3.4 Commissioning

Commissioning is the fourth phase. Basically, the majority of the advantages are gained in this phase. It is difficult to make any changes on conventional systems. However, thanks to the spare memory capacity of PLCs, it is possible to make changes quite cheaply. While in conventional systems, the changes in commissioning generally get lost, all the changes made in PLC are stored in its memory. In order to increase the capacity of PLCs, additional I/O modules can be easily added. Also, it is possible to enlarge the memory of PLCs. Regarding the pace of the processing; PLC is able to manage a program which is composed of logical and arithmetic processes quite rapidly. In this regard, different CPUs can be selected depending on the rapid functioning times and performance. PLCs can communicate among each other, with the other computers and intelligent electronic devices. Also, ease of programming and chance of modification features of PLC are significant in this phase.

2.3.5 Maintenance

Maintenance stands as an additional phase to the four main phases. It is the phase starting with the operation of the plant and continues with production. Most of the time is spent on the faults in many plants. In this regard, a strong tool within PLC offers to facilitate the detection of the faults. The operation of a PLC program and a related system can be directly monitored on the SCADA screen. Moreover, the faults can be detected and a monitoring exercise is possible for the past processing/operations. [1]

2.4 PLCs as RTU

Many manufacturers mention that PLC’s advantage over Remote Terminal Unit (RTU) is that PLC can be used for general purposes and can easily be set up for multiple functions. Basic PLC installation may change but, compared to RTU, this is not significant in normal circumstances.

PLC is popular because of the following:

Economical solution

PLCs are more economical than cabled and relay RTU solutions.

Versatility and flexibility

PLCs can easily adapt their logic and hardware to changing control needs.

Easy to design and install

Due to the importance they attach to software, PLCs make the installation and design of SCADA systems easier.

More reliable

When installed correctly, PLCs are more reliable than classic relay solutions or RTUs manufactured for short term.

Improved control

Compared to RTUs, PLCs provide significantly improved control owing to their software capacity.

Physical advantage

PLC covers less space in comparison to alternative solutions.

Easier maintenance and diagnosis

Hardware/firmware/software problems in the system are identified easily and rapidly thanks to clear reports of the problems and software. In addition, problems related to processing and automation can also be identified.

2.5 Structure of PLCs

PLCs can be divided into two types based on their structure.

2.5.1 Compact Type PLC

These are small sized PLCs. They are manufactured considering a fixed (unconvertible) structure where power supply, input-output unit and central processing unit (CPU) have single module. Their structure is not feasible for further enhancement through additional modules or a limited number of modules may be added. Communication opportunities are limited on industrial network. They are generally used for executing small and local control operations. Such types of PLCs are also called as smart relays. They are more affordable than PLCs. [2] Mitsubishi Compact PLC – MELSEC FX3U and its product range is given as an example to compact type PLC in Figure 2.4 and Table 2.5

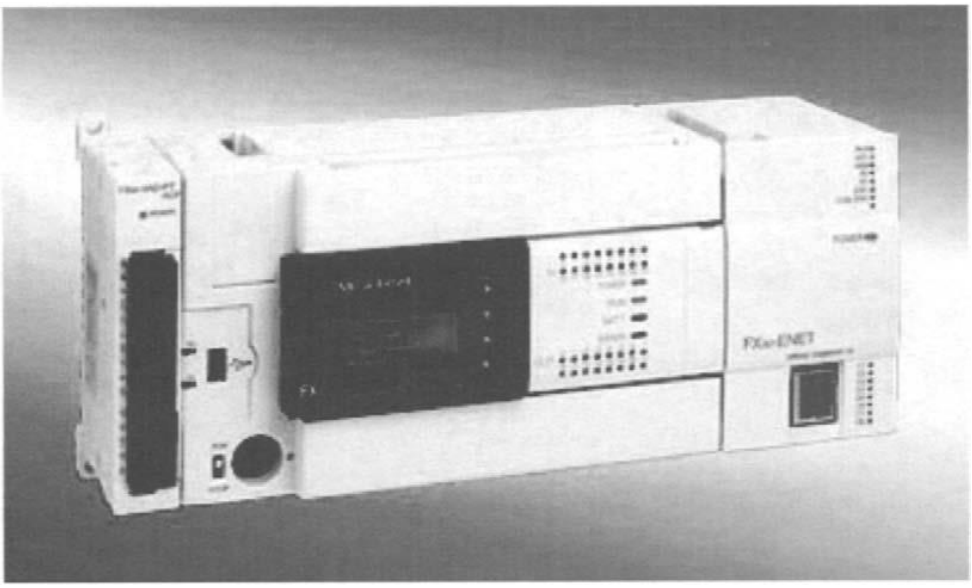


Figure 2.4 Mitsubishi compact PLC – MELSEC FX3U

Table 2.1 Mitsubishi compact PLC: MELSEC FX3U product range

Type	FX3U-16 MR	FX3U-32 MR	FX3U-48 MR	FX3U-64 MR	FX3U-80 MR
Power supply	100–240 V AC				
Inputs	8	16	24	32	40
Outputs	8	16	24	32	40
Digital outputs	Relay				
Program cycle period per logical instruction	0.065 µs				
User memory	64k steps (standard), FLROM cassettes (optional)				
Dimensions in mm (W × H × D)	130 × 90 × 86	150 × 140 × 86	182 × 90 × 86	220 × 90 × 86	285 × 90 × 86

2.5.2 Modular Type PLC

These PLCs are manufactured considering a structure which can be modified and expanded. PLCs with large capacities have their power supplies, input and output units and central processing units as separate modules. In case more than one PLC function at the same time, they are able to communicate through the network. Such types of PLCs where several module options are available can be expanded using the desired modules depending on various needs. [2] Figure 2.5 shows a modular type PLC.

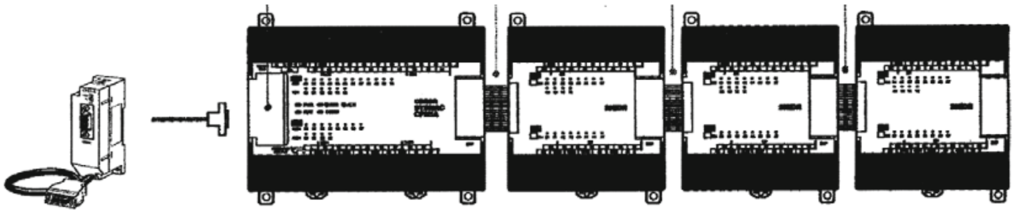


Figure 2.5 Basic configuration of the OMRON CPM1A PLC

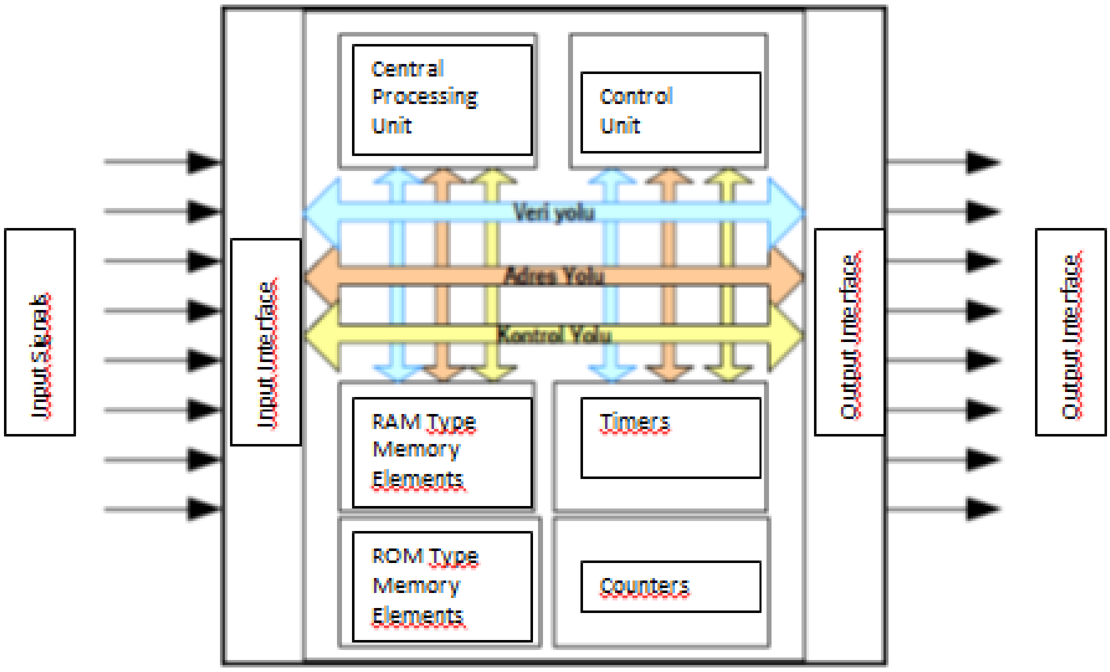


Figure 2.6 Internal structure of PLC

PLCs resemble microcomputers as far as their structure is concerned. As it is the case for all computers, fundamental components that PLCs have are shown in the block diagram above in Figure 2.6. These components are explained below:

2.5.3 Central Processing Unit

This unit ensures that PLC serves for the desired purpose by running the central processing unit user program which is part of PLC. The speed and architecture of CPU identifies the processing capacity of PLC. PLC is recognized as a whole system which is created when CPU and other modules are combined. The CPU in this sense does not refer to electronic chip which functions as central processing unit in PLC.

CPU is the main unit which is found in the backbone of PLC (Control Unit). Modular PLC systems make it possible to add various interfaces (digital module, analog module, Thermocouple Module, Industrial Communication Modules) to CPU.

2.5.4 Random Access Memory (RAM)

This is random access memory which can be written and read. It is a temporary memory unit. RAM memories cannot actually store information when power is cut off. In order to store some information on RAM in such cases, RAMs are fed with Super Capacitor and/or batteries. The data that is typed on this memory can be stored for a long time but when PLC is de-energized, the data is lost.

2.5.5 Electronically Programmable Read-Only Memory (EPROM) and Electronically Erasable Programmable Read-Only Memory (EEPROM)

The information that is written on Read-Only Memory (ROM) type memories is not erased from the memory even if the system is de-energized. ROM type memories (EPROM or EEPROM) have the system software on themselves called firmware which is necessary to operate PLC. Firmware is the software which is permanently stored in ROM type memories, ensures the first time operation of all computer systems and conducts fundamental functions within the system.

2.5.6 Power Supply

It is used to meet the power supply need of modules which are connected to PLC. It has various types, namely 2A, 5A and 10A.

2.5.7 Input / Output Modules

While the input module ensures that signals coming from elements such as sensors and button are transmitted to plc, output module is used to command elements such as motor, led, lambs, contactors or heating elements after they are processed by the program written on plc and outcomes are generated.

2.5.8 Docking Stations

In cases where the number of outputs and inputs is not enough to solve the command problem, the capacity of the device can be expanded by connecting several additional modules to the PLC system. In such a case, input and output units would be added to PLC. The number of input and outputs to be expanded varies depending on the brand and models of PLCs. The company which manufactures the docking station to be added to PLC should also be preferred for the docking station. These modules can be digital or analog.

2.6 PLC Programming Software Languages

2.6.1 Ladder Diagrams

Ladder diagrams are similar to relay logic diagrams used to represent relay control circuits. The main difference between these two types is the following properties of ladder programming which the relay logic diagrams do not have:

- All inputs are represented by contact symbols (-I I-)
- All outputs are represented by coil symbols (-( )-)
- Numeric operations are included in the graphical Ladder command set

Ladder Rungs

A program written in ladder language is composed of rungs which are sets of graphical instructions drawn between two vertical potential bars. The rungs are executed sequentially by the controller as it is shown in Figure 2.7.

The set of graphical instructions represent the following functions:

- Inputs/outputs of the controller (push buttons, sensors, relays, pilot lights…)
- Functions of the controller (timers, counters…)
- Math and logic operations (addition, division, AND, XOR…)
- Comparison operators and other numerical operations (A<B, A=B, shift, rotate…)
- Internal variables in the controller (bits, words…)

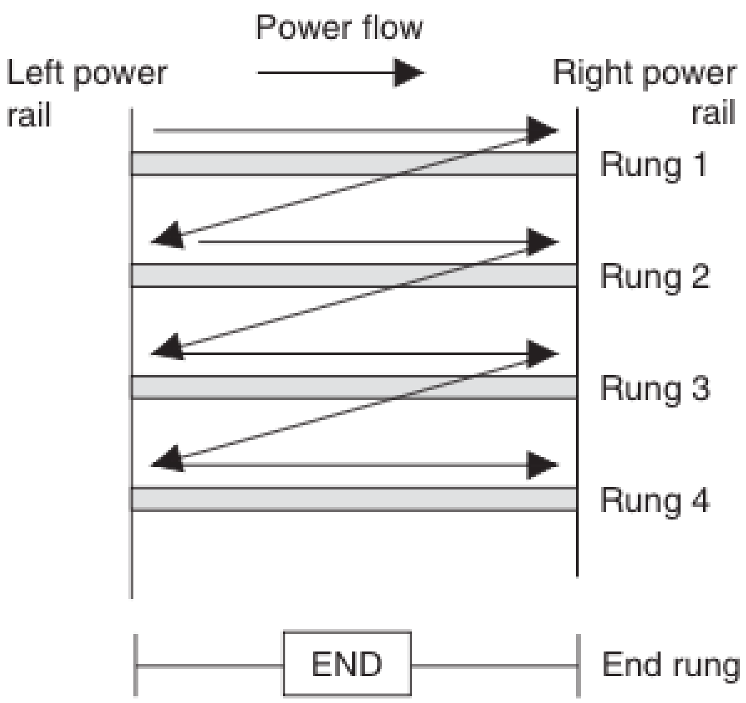


Figure 2.7 Scanning a ladder program

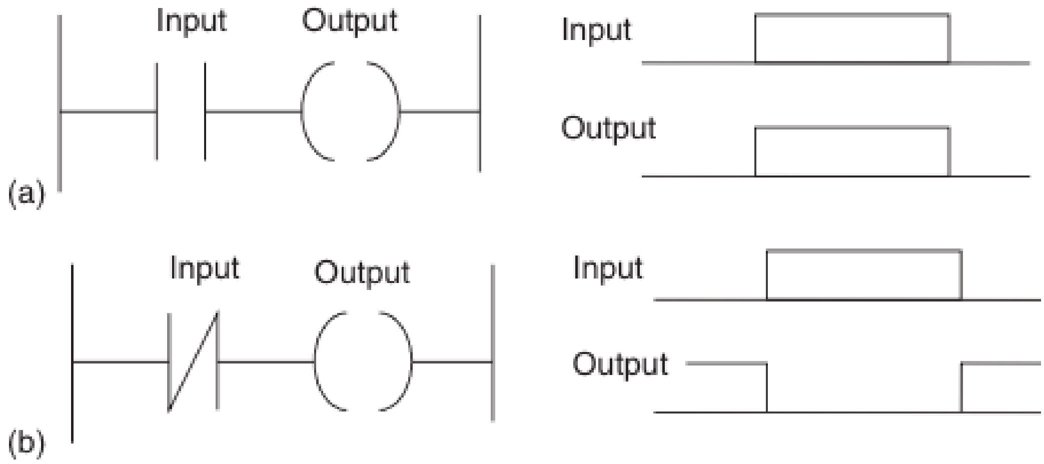


Figure 2.8 A ladder rung

These graphical instructions are arranged with vertical and horizontal connections leading eventually to one or several outputs and/or actions. A rung cannot support more than one group of linked instructions. [2] A ladder rung example is shown above in Figure 2.8.

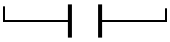
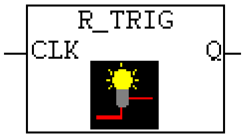
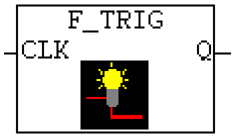
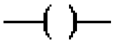
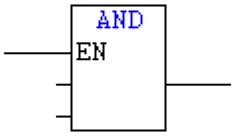
Ladder Diagram Blocks

Ladder diagrams are composed of blocks representing the program flow and functions. These blocks are mentioned below:

- Contacts
- Coils
- Program flow commands
- Function blocks
- Comparison blocks
- Operation blocks

Instructions on the ladder diagram are composed of graphic elements. Table 2.1gives the graphic demonstration of ladder elements which are often used:

Table 2.2 Ladder diagram graphics

Normally Open Contact	
Normally Closed Contact	
Normally Open Parallel Contact	
Normally Closed Parallel Contact	
Rising Edge	
Falling Edge	
Coil	
Set Coil	
Reset Coil	
Function Block	

2.6.2 Sequential Functiom Chart (SFC)

SFC is the abbreviated form of Sequential Function Chart. It also means State Transition Graph. SFC is a programming method and ladder functions which always change and evolve are demonstrated with this method.

When designing consecutive control or logic control circuits using SFC, programming becomes very easy without having to use conventional and complicated design methods. Since this system makes programming very easy, it is worthy of consideration as far as IEC standard is concerned. [2] An example program written in SFC shown in Figure 2.9.

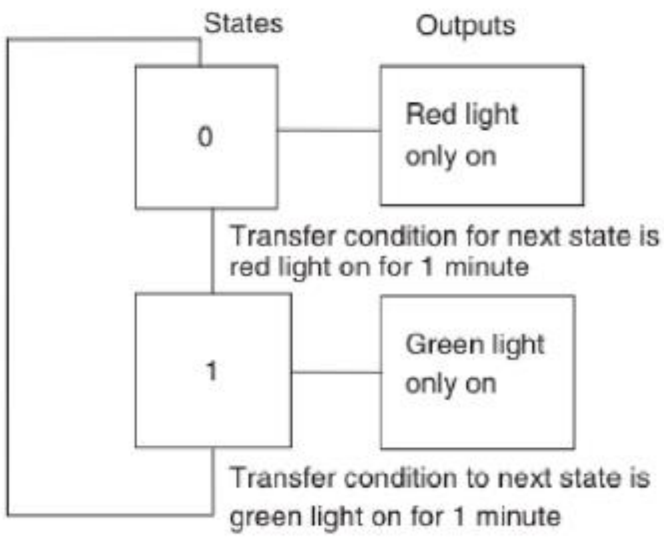


Figure 2.9 Traffic jam adjustment

2.6.3 Function Block Diagram (FBD)

FBD is a programming method which is based on the usage of logic ports and provides a schematic display method. Logic symbols used are shown as boxes. Input signals are located on the left side of the symbols while output signals are on the right side. This method is more easy to use for those who have received a digital electronic training. An exemplary program with FBD can be seen in Figure 2.10.

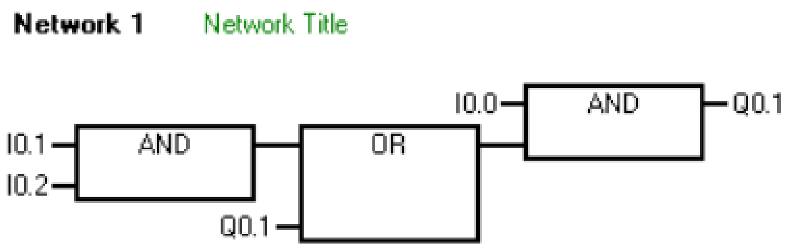


Figure 2.10 Program with function plan (FBD)

2.6.4 Statement List (STL)

STL uses commands which have the same function depending on the type and brand of PLC but which have some small software differences. A command is comprised of Mnemonic showing the operation carried out and operants showing the memory space on which operations are conducted. This method allows for a very broad programming possibility because it is the closest display method to machine code. STL technology targets individuals who are knowledgeable about computer technology.

Programs that have been written with STL, FBD and LADDER methods are convertible to each other provided that there have been no mistakes in program register and program compilation. An exemplary program with STL is seen in Figure 2.11.

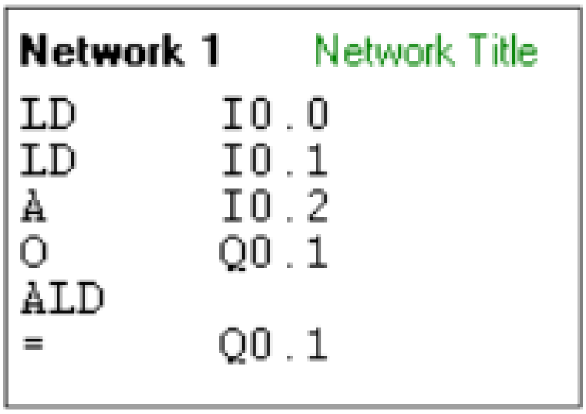


Figure 2.11 A program with statement list (STL)

2.7 Commonly Used Programming Structures

2.7.1 Relay

Relay is used as a switch in the power circuit in order to provide current value of the receivers having high current capacity. Relay has a number of contacts and is able to command more than one circuit simultaneously.

Some of the characteristics of relay are listed below:

- Relay is used for the power circuit switching because of its high current capacity.
- It is possible to switch more than one circuit at the same time as relay can have more than one contact.
- The input (bobbin) and output (contact) of relay are independent from each other, which enables us to use two types of power source circuit for input and output.
- When a circuit relay is selected, relay contacts should be chosen in a way that they can tolerate to 1.2-1.5 times of the necessary current value.
- When relay is in function vibration, noise and abrasion occur.

2.7.2 Logic Symbols

2.7.2.1 AND Circuit

When more than one button is connected serially, there is a need to press all the buttons simultaneously to light the lamp. If you just press one of the buttons, there will be no light. AND is the circuit which shows that when all inputs are ON, the output will be ON as well. [3]

2.7.2.2 OR Circuit

If the lamp is serially connected to buttons upon they are binded in parallel to each other, the lamp will have a light as long as you press one or both of the buttons. OR is the circuit which activates output circuit when you press one or both of the buttons simultaneously. [3]

2.7.2.3 NAND Circuit

NAND circuit is the negated AND. In this circuit, the lamp will have a light if you do not press both of the buttons simultaneously. When you press both of the buttons simultaneously, relay bobbin is energized and opens the contact which remains closed in normal conditions and as lamp circuit is normally transferred on a closed contact, the lamp has no more power and therefore will have no light. NAND is the circuit which shows that when all inputs are ON, the output is OFF. [3]

2.7.2.4 NOR Circuit

NOR is the negative OR, that is, the output lamp will have a light as long as you do not press both of the buttons which are binded in parallel to each other. When you press one or both of the buttons, the output lamp will not have a light. NOR is the citcuit which shows that when one or all of the inputs are ON, the output is OFF. [3]

2.7.3 Timer

Timer is a very practical element of the circuit. Timers are divided into two as ON delay timer and OFF-Delay timer. ON-Delay timers close their normally ON contact and open their normally OFF contact after a certain time when their bobbin is energized. OFF-Delay timers open their normally closed contact and close their normally open contact upon relay bobbin energy cut. [3] Figure 2.12 and 2.13 are examples of ON-Delay and OFF-Delay Timers respectively.

2.7.3.1 ON - Delay Timer

ON-Delay timer works as following:

(a) Voltage is applied to timer bobbin.
(b) The contacts of timer change positions at the end of the pre-set time.
(c) The voltage that is applied to timer bobbin is cut.
(d) Contacts return to their initial positions.

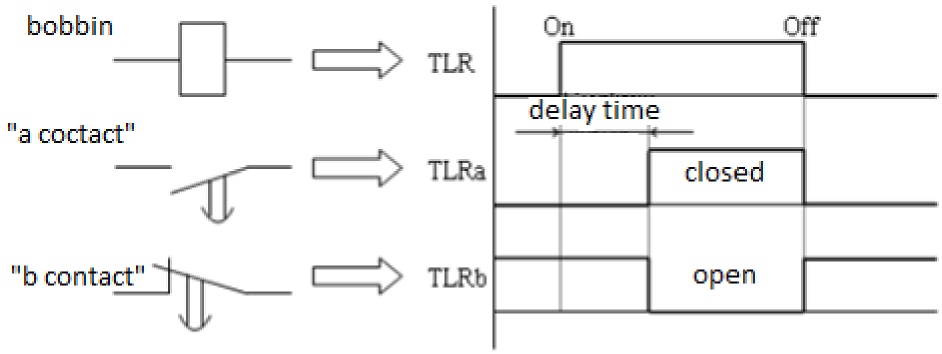


Figure 2.12 ON-Delay timer and its time graphic

2.7.3.2 OFF - Delay Timer

OFF-Delay timer works as following:

(a) Voltage is applied to timer bobbin.
(b) The contacts of timer change positions.
(c) The voltage that is applied to timer bobbin is cut.
(d) The contacts of timer return to their initial positions at the end of the pre-set time

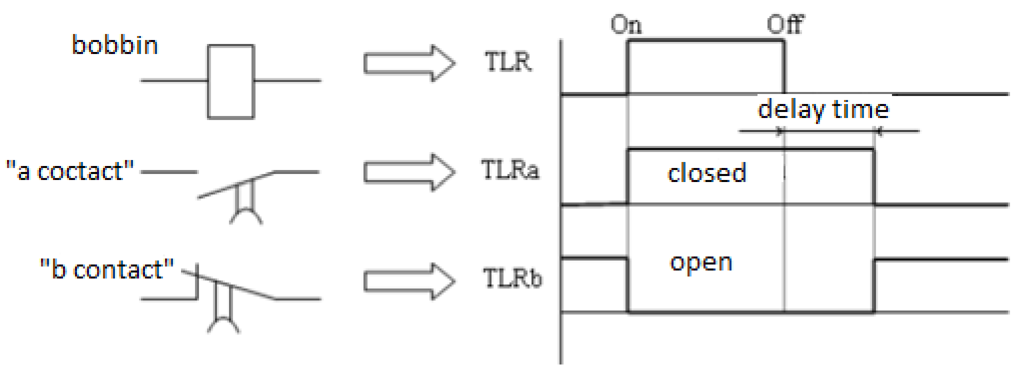


Figure 2.13 OFF-Delay timer and its time graphic

2.7.4 Counter

Counter is an element which enables us to count data applied to input. Counters are used to control and show numbers in the sequential diagram. Total counter serves to count and show the results on the screen. It does not have an output contact. [3] Figure 2.14 shows the time graphics of up counters and down counters.

Up counter

This counter increases the number by one in every input signal

Down counter

This counter decreases the number by one in every input signal.

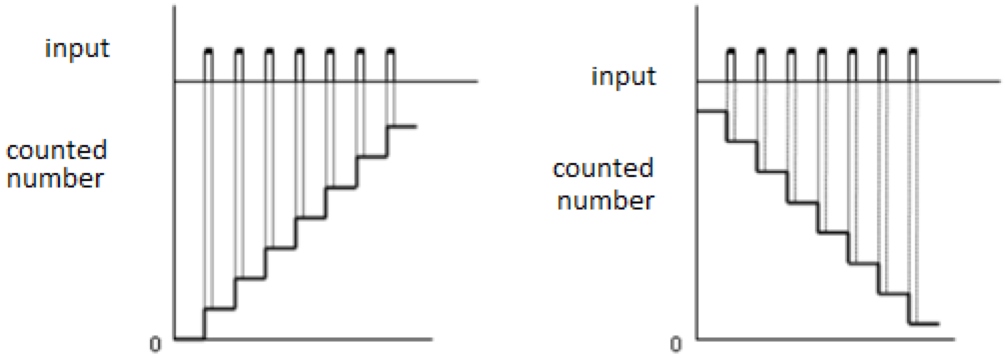


Figure 2.14 Up counter and down counter time graphics

Up/Down counter

This counter has a function which increases or decreases the total value depending on each signal. Up/Down counters' time graphic are shown below in Figure 2.15

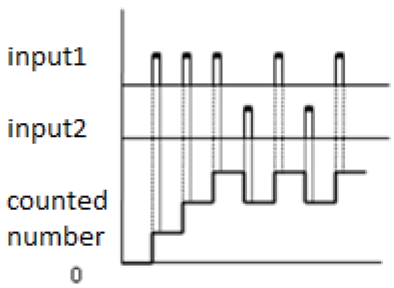


Figure 2.15 Up/Down counter time graphic

2.8 Devices that are Commonly Used in PLC Systems

2.8.1 Small Motors

This type of electric motors is used when low power is necessary. These generally work with DC voltage. These motors are connected to PLC through a relay or driver circuits (transistor, thyristor etc.) formed with semiconductors. If the engine is a low power engine (if PLC's output current is sufficient), it can be connected to PLC's output upon necessary measures are taken. [4] An example of a small motor is shown in Figure 2.16.

Figure 2.16 Small DC brushless electric motors

2.8.2 Solenoid Valves

Electrical energy is converted to linear motion through solenoid valves which are produced with the usage of magnetic energy of power circuit.

Solenoid valves in the system enable us to open and close fluid in a liquid or gas state on a remote control through electrical signal. Depending on the normal open (when there is no electrical signal) or normal close position of the valve, when the valve is in normal position with the impact of gravity, with the spring impact or with the pressure of the liquid itself, it takes a position opposite to normal (if it's open, it is closed, if closed, it is opened) thorough motion provided by the magnetic field formed by electrical signal. For three-way solenoid valves, a joint port is linked to one of the other two ports. Solenoid valves have broad and various fields of usage. [4] A general solenoid valve shown below in Figure 2.17.

Figure 2.17 Solenoid valve

2.8.3 Relays

Small power electromagnetic switches whose structure and appearance is available in Figure 2.18 and 2.19 respectively are named relay. Relays are comprised of three different parts as electromagnet, palette and contacts. Electromagnet is comprised of iron core wrapped with a bobbin. When voltage is applied to relay bobbin, relay is energized; pulls its palette and its contacts change position. When the voltage is cut, contacts return to their previous positions. [4]

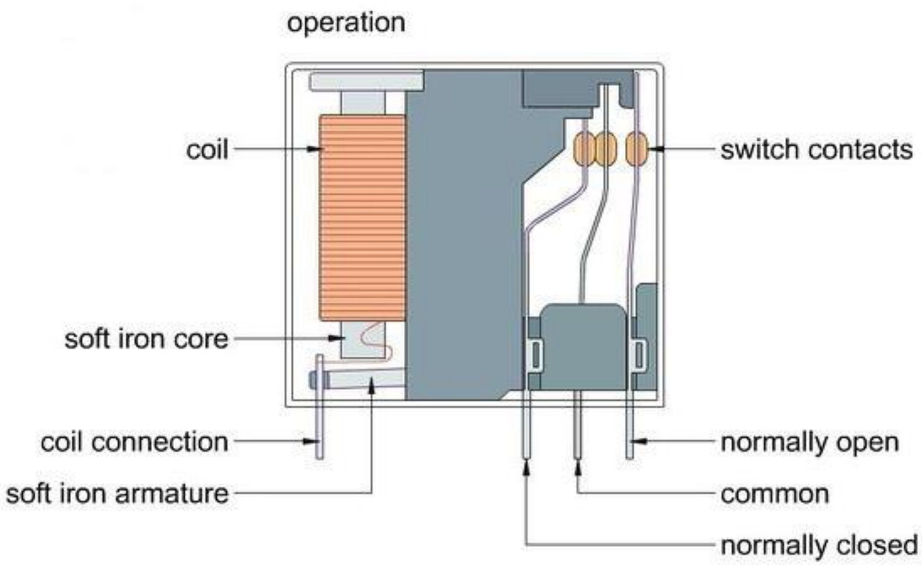


Figure 2.18 Structure of a relay

Figure 2.19 Appearance of a relay

2.8.4 Motors

Motors which are used for automation processes are selected based on the work they need to assume. They can work in direct current or alternating current. Alternating current motors are generally used. The reason why they are preferred is that their structure is simple, they are cheap and have a low maintenance cost. These motors are connected to PLC through relay or contactors. Motor driver circuits are used commonly. [4]

2.8.5 Output Control Lamps

These are pilot signal lamps (Figure 2.20) which show whether a circuit is working or not. They are frequently neon lamps working with 220 V. Furthermore, lamps that work with low power 24 V may also be used. They are manufactured as green, red and yellow. Green lamp means that the device is working while red lamp is used to show that the device has stopped working or is in alarm situation. [4]

Figure 2.20 Signal lamps

2.8.6 Switches

Switches are command elements that change contact position with physical motion. There are different types of switches, such as, pushbutton switch, toggle switch, touch switch, illuminated switch. Switches are generally manufactured in two types. [4]

2.8.7 Permanent Switches

Permanent switches preserve their latest condition until a new command is received. They are frequently used in the main input of command systems. Lamp switches and pacco switches are the examples for permanent switches and can be seen in Figure 2.21 and 2.22. [4]

Figure 2.21 Manually operated permanent lamb switches

Figure 2.22 Manually operated pacco switch

2.8.8 Buttons

There are two types of buttons:

2.8.8.1 Start Button

Contact operates normally in start buttons. When you press the button, open contacts are closed. When the impact on the button is removed, the closed contact opens immediately. This can also be called immediate contact buttons. Start button along with its symbol is seen in Figure 2.23. [4]

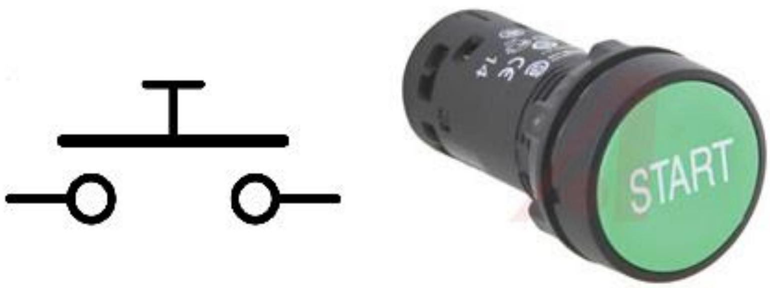


Figure 2.23 Start button and its symbol

2.8.8.2 Stop Button

Contact is normally closed in these buttons. When button is contacted, closed contact is opened; and stays open as long as the contact continues. When the force on the button is removed, contacts take their normal position. Stop button along with its symbol is seen in Figure 2.24. [4]

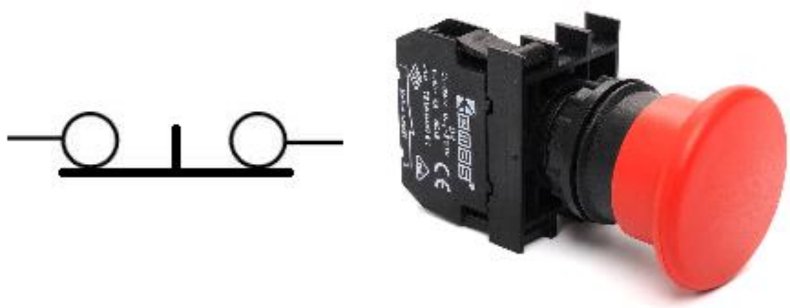


Figure 2.24 Stop button and its symbol

2.8.9 Contactor

Contactor is an electro mechanic device which switches off or on the electric circuit with the closure of another circuit and does this with energizing of a bobbin and has "normally open" and "normally closed" contact on. Generally, it is used for controlling electric motors from a distance. Contactors are connected to the circuit in series. After receiving output data of PLC, it logs out of the system or stays connected. The operation of the equipment checked is ensured in this way. This is especially used for operation of motors. Figure 2.25 shows a contactor.

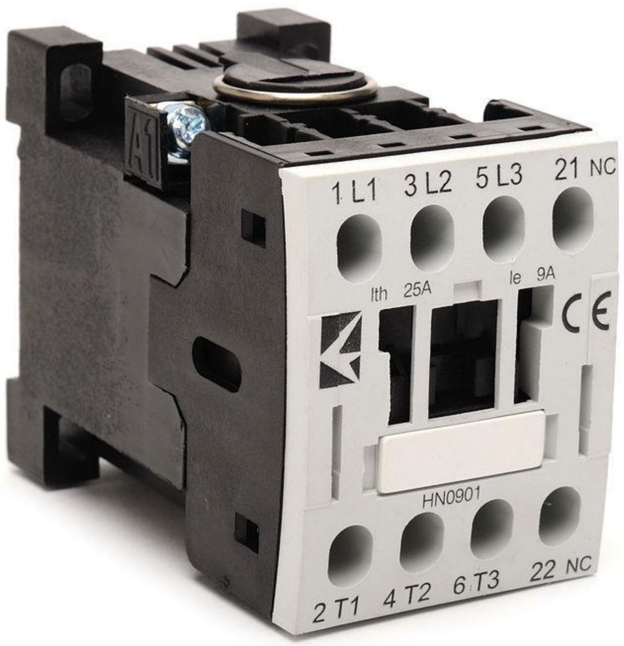


Figure 2.25 Contactor

3. SCADA

3.1 Definition

The term ‘SCADA’ is the abbreviation for “Supervisory Control And Data Acquisition”. SCADA systems have a master terminal unit, field unit, communication system and a SCADA software. This system is comprised of the following steps:

- Data collection,
- Transfer of the collected data to a central system,
- Carrying out the necessary analysis and calculations,
- Transfer of the information gathered to the screens used by operators.

SCADA system is used so as to monitor field equipments or facilities and controls are enabled through automatic commands or commands received from operators.

3.2 History

First SCADA (Supervisory Control and Data Acquisition) systems enabled data acquisition through display panels, light and strip chart recorders. These devices have been and are still being used for supervisory control and data acquisition in machines, factories and power generation facilities. The Figure 3.1 shows a sensor directed to a panel system.

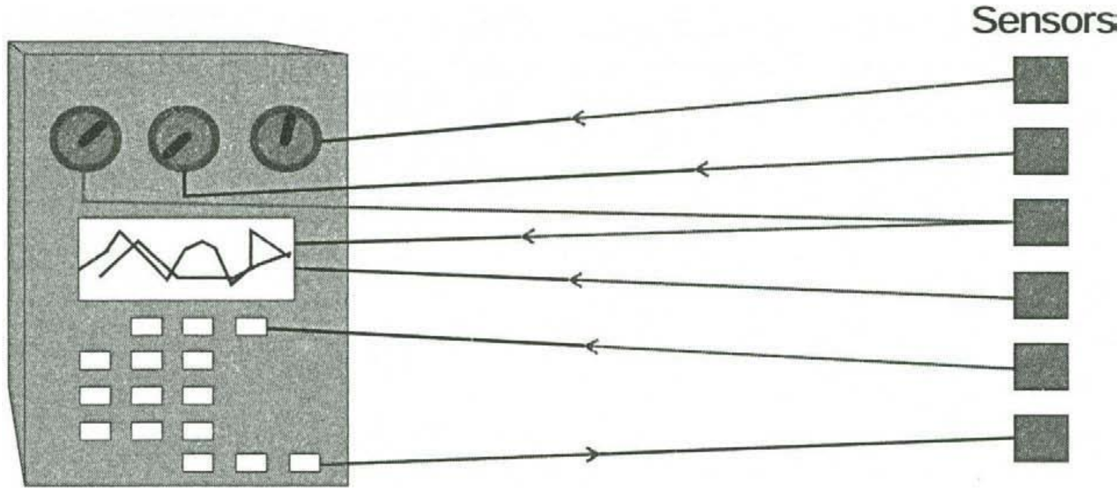


Figure 3.1 The sensor directed to a panel SCADA system

The sensor to panel type of SCADA system has the below mentioned advantages:

- It is simple and does not necessitate CPU, RAM, ROM and software programming.
- Sensor is directly connected to display, indicator and switch on the panel.
- Addition of devices such as switch and indicator is generally easy and cheap.

Disadvantages of the sensor to panel type of SCADA system:

- Cabling control becomes very hard after hundreds of sensors are installed.
- Data quality and type are minimum and not open to improvement.
- Installation of additional sensors becomes operationally difficult as the system gets bigger.
- It is pretty difficult to reshape the system.
- It is not possible to have a simulation that uses real data
- Data storage is on the minimum level and difficult to manage.
- It is not possible to monitor data and alarms remotely.
- Search and counter devices should be monitored for 24 hours. [5]

3.3 Advantages

SCADA system should provide advanced control and supervision for operators. SCADA system can have the following overall benefits in practice:

- Mimics which were defined by the user (Operation simulation) and monitoring of the operation through objects to be used on screen mimic (level, temperature, pressure, digital signals, situation of valves and motor, system situation etc.),
- Entry of production definitions and informing operator about the existing prescriptions through prescription screens,
- Entry and supervision of PID parameters,
- Keeping historical and real time trends of operational value,
- Instantaneous and periodical reporting (production, prescription, inventory.),
- Manual intervention by SCADA screens to automatic system,
- Display of alarm and events and registering them to the printer and/or database,
- Monitoring and control of the system on web if data servers in SCADA system are connected to the internet.

3.4. Services Provided By SCADA System

Services provided by SCADA system are as follows:

- Management and monitoring of the system from the center thanks to computerized central command,
- Reduction in the number of stable and mobile staff members, more efficient use of staff members,
- Savings from electricity, oil, water, heating costs etc.,
- Reduction of the product or service cost, cheaper purchase and profitability,
- Products or the services provided are in line with certain quality standards owing to continuous measurements, standardization of the products and services that consumers reach,

- Minimizing faults or failures that stem from negligence,
- Taking necessary measures for situations that may result in failures beforehand and thus protection of the hardware,
- Keeping regular and reliable records of statistical information on the operating system and enabling rapid information flow to the top management,
- Efficient and economic operation of the system
- Monitoring of the operation through mimics defined by the user and objects to be used on screen mimic (level, temperature, pressure, digital signals, situation of valves and motor, system situation etc.),
- Entry of production definitions and informing operator about the existing prescriptions through prescription screens,
- Input of the parameters necessary for the system (set point, minimum and maximum alarm values etc.) through parameter screens,
- Input and supervision of PID parameters,
- Keeping historical and real time records of operating values,
- Receiving instantaneous and periodical reports (production, prescription, inventory etc.),
- Manual intervention by SCADA screens to automatic system,
- Display of alarm and events and recording them to the printer and/or database and printing,
- Advanced quality control (for instance, statistical process control- PLC support). [6]

3.5 SCADA System's Field of Application

SCADA system has a number of field of application. It is used in most of the regional and local facilities spread to a vast geography. It can also serve as an infrastructure for other systems. Some of the fields are shown in below:

- Electric transmission and distribution lines
- Natural gas transmission and distribution lines
- Petroleum storage and distribution lines
- Water treatment, storage and distribution lines

- Illumination
- Heating, cooling and air conditioning,
- Fire warning and fire department systems
- Moving staircase and escalator
- Security door systems
- Camera security systems

3.6 SCADA Hardware

3.6.1 Master Terminal Unit

MTU (Master Terminal Unit) is the main station or computer which serves as the main controller. In modern SCADA systems, MTUs are computer based. This computer system can either involve a simple computer or a whole network of computers. In addition to these computer systems, complementary devices such as printers or back up units can also be within MTUs. MTUs are responsible for collecting field data from RTUs which are scattered in SCADA systems and analyzing them to take the necessary control measures.

3.6.2 Field Unit

Field units are hardware systems which actually carry out and control operations and which have processors in their structure. *RTU (Remote Terminal Unit), DCS (Distributed Control System)* and *PLC* (*Programmable Logic Controller*) can set an example for these systems. Field units enable data collection from the field and make field equipments carry out commands sent from the operator. (from *Master Terminal Unit*).

3.7 SCADA Software

SCADA software is an interface program which shows operations carried out by PLC (Programmable Logic Controller) and/or information coming from RTU (Remote Terminal Unit). This program is named as HMI (Human Machine

Interface). HMI is connected to a database which works continuously real-time. Database can instantaneously, hourly, daily, monthly and annually record information it receives from the server. The better the interface software is defined, designed, written, controlled and tested, the better the SCADA system works.

Duties of SCADA software are listed below:

- To supervise and monitor RTUs
- To inform operators instantaneously about the information, alarm and event alerts received from RTUs
- To implement commands of the operators immediately and inform operators about their results
- To form the image on screens of the projection system simultaneously with the computer screens
- To have print outs of alarm and event alerts along with statistical reports
- To control audio alert system
- To ban unauthorized access to SCADA system

3.8 Communication System

It is possible to use various cables (Ethernet, RS232, Profibus, RS-485 etc.) for communication system. Besides, communication through RF (Radio frequency) or satellite is also practicable. Cables are especially used in factories. However, cabling, starting from scratch for information transfer from remote units is very expensive and technically not feasible. In such cases, phone lines can be leased or communication through power transmission lines or even satellite or RF modems may be enabled.

Communication network is the most important part of SCADA system which determines the baud performance. The significance of this issue is clear when the size of the controlled systems are taken into consideration and as the whole communication which includes data transfer between interfaces that are connected to each other on various automation levels of the controlled systems is carried out on communication networks.

3.8.1 Ethernet

These are generally set up as 10 Mbps baseband coaxial cables. Carrier Sense Multiple Access and Collision Detection are media access control methods used by Ethernet. These are the most popular approaches with LANs and therefore will be elaborated in a more detailed way than alternative methods.

The idea of Ethernet began with the radio transmission experiences where a significant number of stations are in contact with each other in random times. Especially, before transmitting a message (on the general connection cables) to other node, a station (or node) listens for a bus (cable or radio) activity. If the station identifies that other nodes are not transmitting, it sends its own message. It is probable that another station is in transmission simultaneously. If there is collision for two nodes, each node suspends transmission (in different times because of random pauses) for a while before trying to restart transmission.

Integrated guideline and receiver - transmitter unit (defined as MAC unit) is comprised of the following parts:

- Receiver – transmitter unit is used for data transmission and receiving, determines collision, enables electrical isolation and guarantees bus working.
- Guideline that makes the physical connection of the coaxial cable.

Control card which is connected to the receiver/transmitter sheathed cable includes media access control unit which provides message segmentation and microprocessor for grid installation linked to protocols.

Here are some recommendations to decrease the number of collisions in Ethernet network:

- Keep all cables as short as possible.
- Keep excessive working sources and the distance between them as short as possible.
- To decrease backbone traffic, isolate these nodes from major network backbone
- Use buffered repeaters instead of repeater bits
- Control unnecessary packages that are directed to nonexistent nodes
- Do not forget that monitoring devices displaying network traffic increase the network traffic (collision traffic)
- Make sure that cable grounding is conducted from a single point above cable terminators [7]

3.8.2 RS-232

RS-232 is the name of serial communication protocols used by computers and other electronic devices. It is used to connect modems, printers, data collection modules, test device and control circuits. RS-232 is designed for exchange of information between two devices. The distance may change between 50 and 100 feet (15-30 m). The type of cables and bit rate play an important role at this point. [8]

In general, DB-25 is used as a connector. The pins which are often used are as follows:

Pin 1: Protective Ground

Pin 2: Transmitted serial output data

Pin 3: Transmitted serial input data

Pin 7: Signal Ground

Disadvantages:

- Speed up to 20 kbps at maximum of data transmission
- Easily affected by noise emitted on the signal line
- Distance between two devices is limited to change between 15-30 meters
- Point-to-point data flow
- It is affected by the noise very fast

3.8.3 RS-485

Similar to RS-232, RS-485 is also a serial communication protocol. It has various types of properties when compared to RS-232;

- Link length can be up to 4000 feet.
- Baud rate can be 10Mbps at maximum
- 32 drivers are allowed to function on the same line.
- Logic “0” changes between -1.5V and -6V whereas logic “1” changes between 1.5V and 6V. [8]

4. CONSTRUCTION CHEMICALS MANUFACTURING PLANT AND ITS AUTOMATION

4.1 Construction Chemicals and Their Manufacturing Plant

As is the case with many developed and developing countries, construction materials sector is the locomotive of the economy in Turkey. Turkish construction materials sector is growing at a significant rate year by year with the increase in the number of huge projects such as projects addressing at housing demands, renewable energy projects which cover infrastructure, urban transformation efforts, energy transmission lines that pass through Turkey, the project named “Channel Istanbul” and the planned construction of Istanbul’s 3^{rd} bridge.

It is possible to define construction materials as materials which are produced to be continuously used for all construction works including buildings, infrastructure and other construction processes. Therefore, construction materials sector includes areas of activity related to the manufacturing and launching of these products into the market. Areas of activity in this sector are listed below:

- Iron and steel
- Cement
- Plaster
- Marble
- Glass
- Paint
- Air-conditioning
- Wall paper
- Wood products
- Pipes
- Cables
- Construction chemicals

Construction chemicals sector, which is one of the above mentioned areas of activity, has two main product groups. The first group is comprised of filling-additive products which are used for cement, concrete and ready-mixed concrete production. These groups of products are used to overcome problems related to transfer, processing and preservation of concrete under various conditions and to improve strength and durability of the construction. Examples of the products within the first group are listed below:

- Plasticizers / fluidifiers
- Water impermeability additives
- Anti freezes
- Fibers for concrete
- Mould oils
- Curing agents
- Additional concrete products
- Mortar admixture
- Strengtheners
- Chrome reducers
- Polymer powders

The second group is comprised of filling and adhesive materials which are used for ceramic tile, tile, waterproofing and floor coatings. Examples of the products within the second group are listed below:

- Epoxy based coating system
- Polyurethane mastics
- Foams
- Silicon mastics
- Primers
- Joint filters
- Cement based admixtures
- Epoxy based admixtures

There are approximately 500 companies (as producers, sales and implementing companies) in the construction chemicals sector in Turkey. These companies employ around 13-15 thousand people. Market size of construction chemical sector in 2012 was 1.9 billion Turkish Liras. It is estimated that in parallel to the growth in the related sectors which create demand for construction chemicals, the market size was 2.3 billion Turkish Liras in 2013 and 2.6 in 2014. It is also estimated that the production value of the sector was respectively 1.7 billion Turkish Liras in 2012, 2.06 in 2013 and 2.33 in 2014. [9]

4.2 Automation of Construction Chemicals Manufacturing Plant

4.2.1 PLC

The PLC to be used in this study was selected after considering the requirements in the plant, analysing many parameters and interviewing with the users to learn their needs and desires. Design process needs to be handled at length. The better the system design is, the more improved the quality of the operation and the less the cost would be.

ABB brand was chosen to be used as PLC. “ABB Control Builder Plus” was chosen as physical design programme whereas “ABB CodeSYS” was decided to be used as PLC software programme. Programming language was written in the logic of “ST” and “CFC”.

4.2.1.1 Control Builder Plus And Creating A New Project

We use programmes provided by ABB in the system. One of these programmes called ABB Control Builder Plus enables us to carry out PLC physical design and write PLC programme using CodeSYS which is a part of this programme. Home page of the program is shown in Figure 4.1.

To create a new project, we first start Control Builder Plus programme.

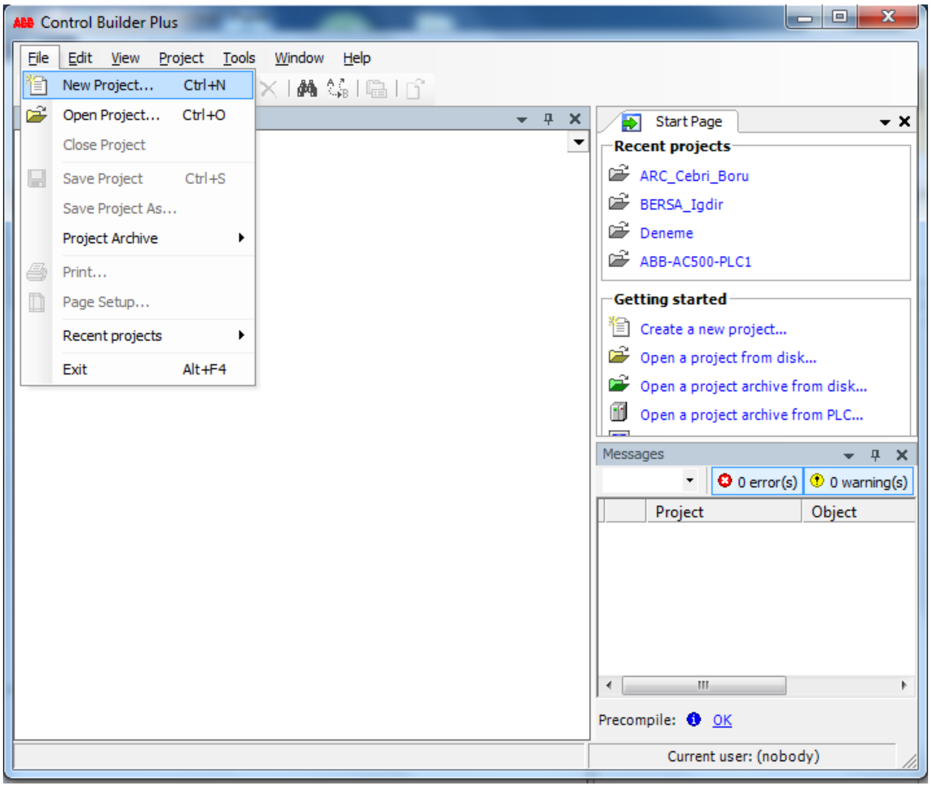


Figure 4.1 Home page of control builder plus programme

Choose *New Project* command under *File* as seen in the figure. Then select the PLC (if we know it in advance) to be used from the window that will pop up. If we do not want to choose from this window, select *Empty Project* and approve. The window that enables the selecting PLC types is shown in Figure 4.2.

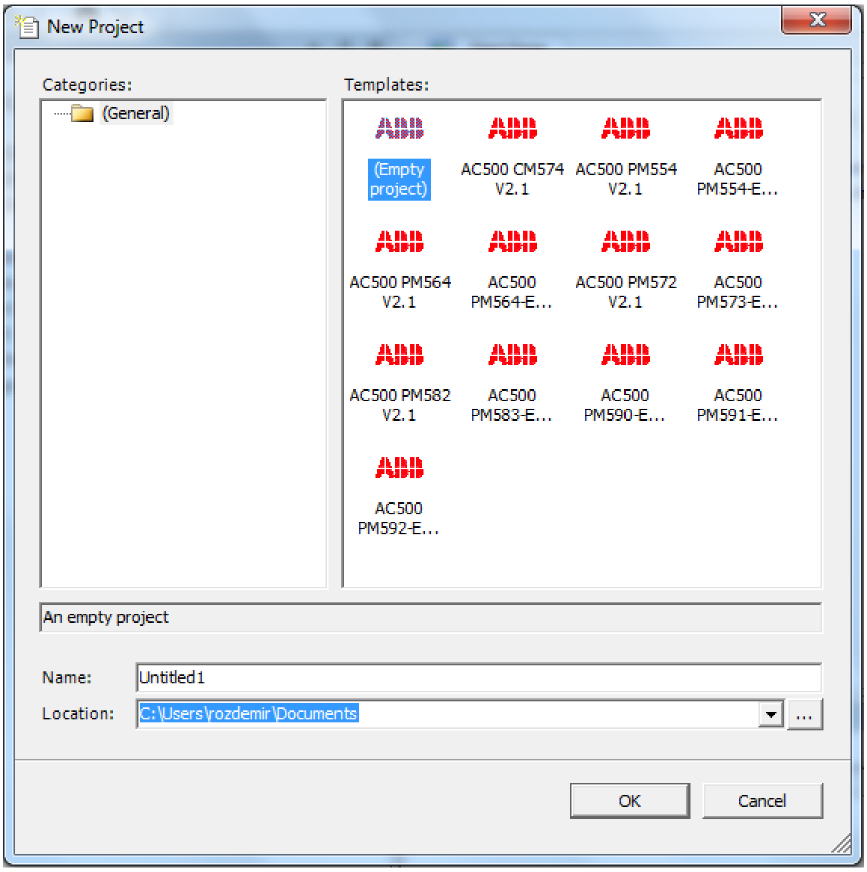


Figure 4.2 Selecting PLC type

4.2.1.2 PLC Selection

To define configurations through Control Builder Plus programme, select PLC. To complete this step, right click on *Untitled1* which is under a part called *Devices* and select *Add Device*. Then select the appropriate PLC from the popped up window in accordance with the defined needs. PLC selection window is shown in Figure 4.3.

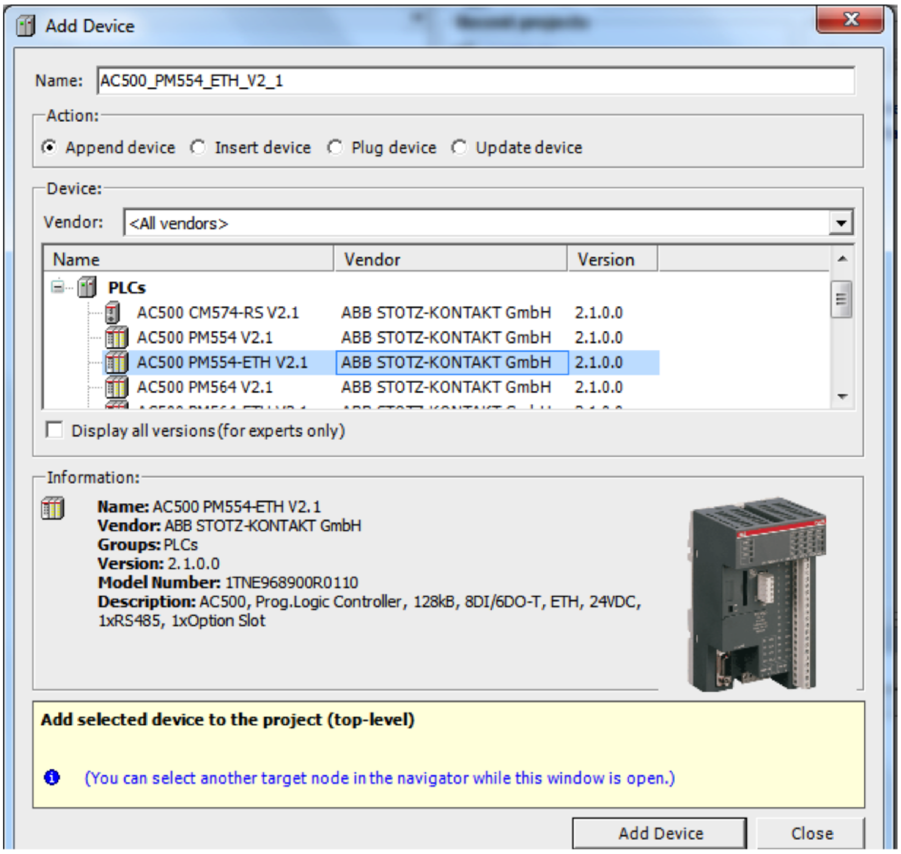


Figure 4.3 PLC selection window

The appropriate PLC is selected considering the number of motors, valves and sensors in the plant, the number and types of digital and analogue input and outputs to be used in the system, which depend on the selected types of motors, valves and sensors, and PLC voltage and storage requirements as well as telecommunication standards.

Following the identification of all these needs, the following PLC was selected and used for this study and is shown in Figure 4.4 and Figure 4.5;

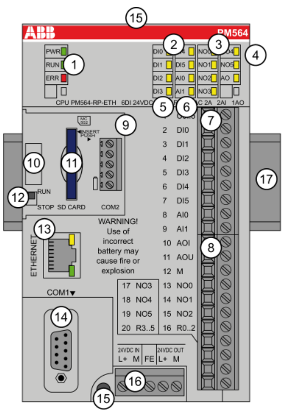
- ABB AC500 PM554-ETH V2.1

Technical specifications of the selected PLC are as follows:

- 128 kB memory
- 8 digital inputs

- 6 digital outputs
- Ethernet port
- 24 VDC feeding
- 1 RS-485 port
- 1 single slot option

Figure 4.4 ABB PLC



1	3 LEDs to display the status of the CPU	
2	PM554-xP	8 yellow LEDs to display the status of the digital input signals
	PM564-xP	6 yellow LEDs to display the status of the digital input signals 2 yellow LEDs to display the status of the analog input signals
3	PM554-xP	6 yellow LEDs to display the status of the digital output signals
	PM564-xP	6 yellow LEDs to display the status of the digital output signals 1 yellow LED to display the status of the analog output signal
4	I/O-Bus for connecting additional I/O modules	
5	Terminal number	
6	Allocation between terminal number and signal name	
7	Terminal block for input/output signals (9-pole)	
8	Terminal block for input/output signals (11-pole)	
9	5-pin removable connector for COM2 (optional)	
10	Handle bar for opening the cover for the expansion modules	
11	SD Memory Card slot (optional)	
12	RUN/STOP switch	
13	Ethernet interface (depending on model)	
14	9-pin SUB-D- jack (COM1) for RS-485 connection	
15	2 holes for wall-mounting with screws	
16	5-pin removable connector for power supply (24 V DC or 100-240 V AC - depending on model)	
17	DIN rail	

Figure 4.5 Specifications of an ABB PLC

4.2.1.3 Selection of Digital and Analogue Input and Output Modules

The number and variety of I/O modules to be selected vary depending on the PLC which is identified using the Control Builder Plus programme. To select the best I/O module, right click on the "I/O_Bus" which is found on the window on the left of the programme (Devices) and select "Add Device". Module or modules that are needed are selected from the window that pops up, shown in Figure 4.6. The modules that are selected at this stage are automatically added to the system by the programme.

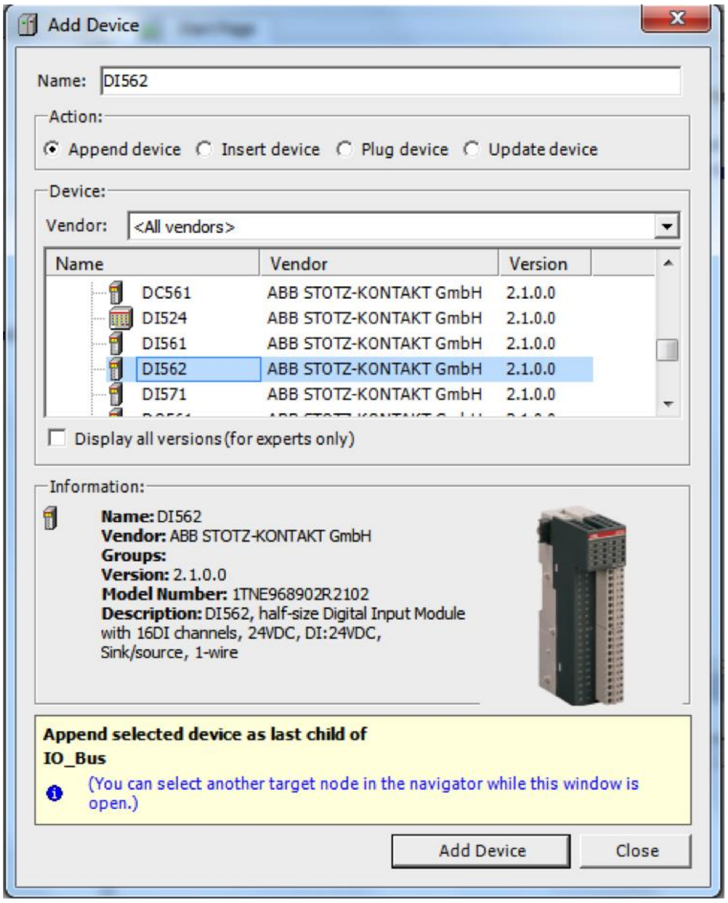


Figure 4.6 I/O modules selection window

Since 40 digital inputs, 25 digital outputs as well as 2 analog inputs and 1 analog output are needed by the system; additional modules of DI562 and DO561 have been used.

4.2.1.4 I/O Addressing

Upon each list, figures including information on the related module which covers I/Os are presented. Table 4.1 shows the onboard Inputs and Outputs and Figure 4.7 shows the PLC that this particular addressed I/Os.

Table 4.1 Onboard I/Os

Unit	Tag	Address	Type
Onboard I/Os	DI_Silo1_Helezonu_Ariza	%IX0.0	BOOL
	DI_Silo2_YildizBes_Helezonu_Ariza	%IX0.1	BOOL
	DI_Silo3_Helezonu_Ariza	%IX0.2	BOOL
	DI_Silo4_Helezonu_Ariza	%IX0.3	BOOL
	DI_Kucuk_Toz_Toplama1_Ariza	%IX1.1	BOOL
	DI_Aktarma_Bandi3_Ariza	%IX1.2	BOOL
	DI_Aktarma_Bandi1_Ariza	%IX1.3	BOOL
	DI_Kucuk_Toz_Toplama2_Ariza	%IX1.4	BOOL
	DI_Doner_Kafa_Ariza	%IX1.5	BOOL
	DI_Buyuk_Toz_Toplama1_Ariza	%IX1.6	BOOL
	DI_Buyuk_Toz_Toplama2_Ariza	%IX1.7	BOOL
	AI_Agrega_Kantari	%IW1	INT
	AI_Kimyasal_Kantari	%IW2	INT
	AO_Doner_Kafa	%QW0	INT
	DO_Yedek_Mot	%QX6.0	BOOL

Figure 4.7 Specifications of AC500 PM554-ETH

Table 4.2 shows the Inputs and their addresses and Figure 4.8 shows the module that this particular addressed Inputs.

Table 4.2 Inputs in DI562

Unit	Tag	Address	Type
DI562	DI_Mikser_Bos_Sensoru	%IX24.0	BOOL
	DI_Doner_Kafa_Dolu_Sensoru	%IX24.1	BOOL
	DI_Kimyasal_Kantar_Klepe_Acik	%IX24.2	BOOL
	DI_Kimyasal_Kantar_Klepe_Kapali	%IX24.3	BOOL
	DI_Agrega_Kantar_Klepe_Acik	%IX24.4	BOOL
	DI_Agrega_Kantar_Klepe_Kapali	%IX24.5	BOOL
	DI_Mikser_Klepe_Acik	%IX24.6	BOOL
	DI_Mikser_Klepe_Kapali	%IX24.7	BOOL
	DI_Silo1_Ust_Seviye_Sensoru	%IX25.0	BOOL
	DI_Silo1_Alt_Seviye_Sensoru	%IX25.1	BOOL
	DI_Silo2_Ust_Seviye_Sensoru	%IX25.2	BOOL
	DI_Silo2_Alt_Seviye_Sensoru	%IX25.3	BOOL
	DI_Silo3_Ust_Seviye_Sensoru	%IX25.4	BOOL
	DI_Silo3_Alt_Seviye_Sensoru	%IX25.5	BOOL
	DI_Silo4_Ust_Seviye_Sensoru	%IX25.6	BOOL
	DI_Silo4_Alt_Seviye_Sensoru	%IX25.7	BOOL

Figure 4.8 Specifications of DI562 module

Table 4.3 shows the Inputs and their addresses and Figure 4.9 shows the module that this particular addressed Inputs.

Table 4.3 Inputs in DI562

Unit	Tag	Address	Type
DI562	DI_Aktarma_Bandi2_Ariza	%IX22.0	BOOL
	DI_Filtre_Cikis_Helezonu_Ariza	%IX22.1	BOOL
	DI_Filtre_Cikis_YildizBesleme_Ariza	%IX22.2	BOOL
	DI_Kimyasal1_Helezon_Ariza	%IX22.3	BOOL
	DI_Kimyasal2_Helezon_Ariza	%IX22.4	BOOL
	DI_Filtre_Seviye_Sensoru	%IX22.5	BOOL
	DI_Mikser_Ariza	%IX22.6	BOOL
	DI_Bant_Start	%IX22.7	BOOL
	DI_Silo1_Klepe_Acik	%IX23.0	BOOL
	DI_Silo2_Klepe_Acik	%IX23.1	BOOL
	DI_Silo3_Klepe_Acik	%IX23.2	BOOL
	DI_Silo4_Klepe_Acik	%IX23.3	BOOL
	DI_Yedek_Mot_Ariza	%IX23.4	BOOL

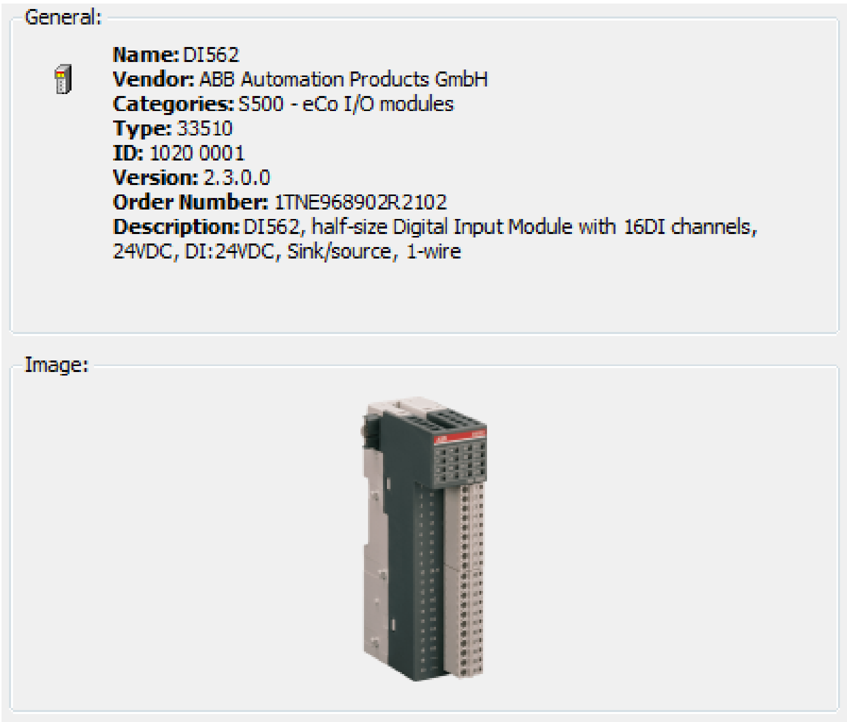


Figure 4.9 Specifications of DI562 module

Table 4.4 shows the Outputs and their addresses and Figure 4.10 shows the module that this particular addressed Outputs.

Table 4.4 Outputs in DO561

Unit	Tag	Address	Type
DO561	DO_Silo1_Helezonu_Run	%QX26.0	BOOL
	DO_Silo2_YildizBes_Helezonu_Run	%QX26.1	BOOL
	DO_Silo3_Helezonu_Run	%QX26.2	BOOL
	DO_Silo4_Helezonu_Run	%QX26.3	BOOL
	DO_Silo1_Klepe_Ac	%QX26.4	BOOL
	DO_Silo2_Klepe_Ac	%QX26.5	BOOL
	DO_Silo3_Klepe_Ac	%QX26.6	BOOL
	DO_Silo4_Klepe_Ac	%QX26.7	BOOL

Figure 4.10 Specifications of DO561 module

Table 4.5 shows the Outputs and their addresses and Figure 4.11 shows the module that this particular addressed Outputs.

Table 4.5 Outputs in DO561

Unit	Tag	Address	Type
DO561	DO_Filtre_Patlatma	%QX27.0	BOOL
	DO_Silo_Patlatma	%QX27.1	BOOL
	DO_Kucuk_TozToplama1_Helezonu_Run	%QX27.2	BOOL
	DO_Aktarma_Bandi3_Run	%QX27.3	BOOL
	DO_Aktarma_Bandi1_Run	%QX27.4	BOOL
	DO_Kucuk_TozToplama2_Helezonu_Run	%QX27.5	BOOL
	DO_Doner_Kafa_Run	%QX27.6	BOOL
	DO_Buyuk_TozToplama1_Helezonu_Run	%QX27.7	BOOL

Figure 4.11 Specifications of DO561 module

Table 4.6 shows the Outputs and their addresses and Figure 4.12 shows the module that this particular addressed Outputs.

Table 4.6 Outputs in DO561

Unit	Tag	Address	Type
DO561	DO_Buyuk_TozToplama2_Helezonu_Run	%QX28.0	BOOL
	DO_Aktarma_Bandi2_Run	%QX28.1	BOOL
	DO_Filtre_Cikis_Helezonu_Run	%QX28.2	BOOL
	DO_Kimyasal1_Helezonu_Run	%QX28.3	BOOL
	DO_Kimyasal2_Helezonu_Run	%QX28.4	BOOL
	DO_Kimyasal_Kantar_Klepe_Ac	%QX28.5	BOOL
	DO_Agrega_Kantar_Klepe_Ac	%QX28.6	BOOL
	DO_Mikser_Klepe_Ac	%QX28.7	BOOL

Figure 4.12 Specifications of DO561 module

When I/O addresses are assigned, you should pay attention not to overlap addresses. Therefore, addresses should be assigned according to a plan and basic information for addressing should be available. In short:

1 Double Word	=	2 Words
1 Word	=	2 Bytes
1 Byte	=	8 Bits

Furthermore the PLC used for this study has a memory of 128 Kbytes.

Based on the aforementioned information, it is significant to take selected type into account when addressing for tags.

To integrate I/Os into the programme, double click on the relevant module on the left window (Devices) to open a new window on the right. Select the menu called "I/O mapping" on this window to address the I/Os on the list.

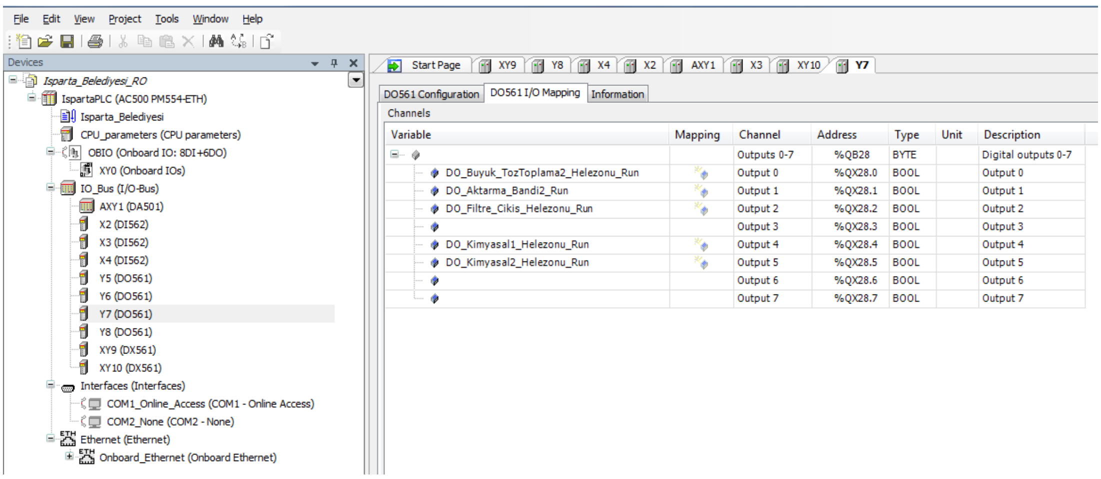


Figure 4.13 I/O addressing window

4.2.1.5 Communication between SCADA and PLC

To ensure communication you should first select Options command in *Project* tab in the CodeSYS program. Both of the preferences in the *Symbol Configuration* in the opened window are selected and by clicking on the *Configure symbol file* button, *export variables of object* and *export data entries* options are selected from the window opened. Only the file titled "global variables" among the files which are in the same window and which define the data to be presented for communication is selected. All the tags and global symbols that are defined in the PLC program are present in the file titled "Global variables". Global variables window can be seen in Figure 4.14 below.

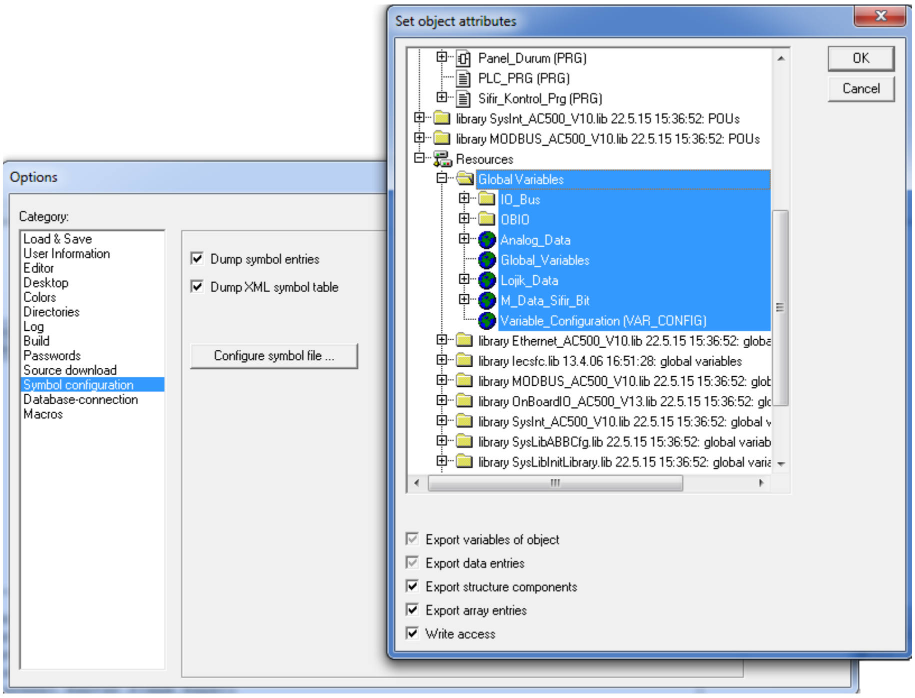


Figure 4.14 Global variables window

In the second step, server configurations of the computer are conducted. To this aim, CodeSYS program is used again. CodeSYS Configurator program is opened and necessary configurations are carried out as shown in the Figure 4.15. You should keep in mind that the name of your SCADA Project and the Project name should exactly be the same.

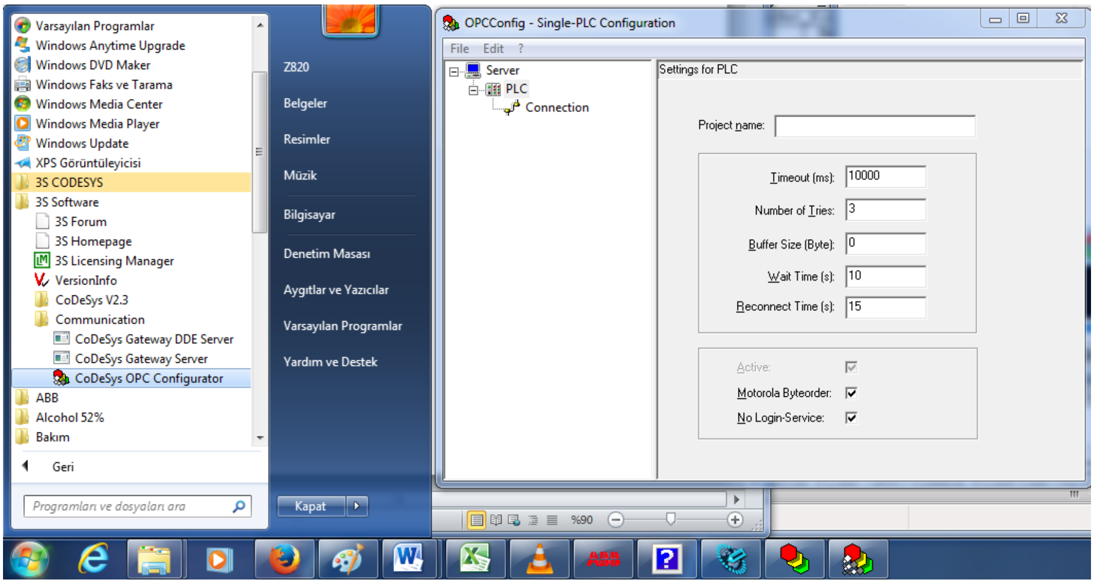


Figure 4.15 CodeSYS OPC Configuration program

As a last step, OPC server is set up. This is done through a module within SCADA program CodeSYS. *Devices* tab in the CodeSYS program is selected in order to install OPC server. Then, *New Device* is selected and *OPC* among the options in the window opened is selected. By clicking on the OPC symbol in the *miscellaneous* tab in OPC, *CodeSYS.OPC.02* option is defined.

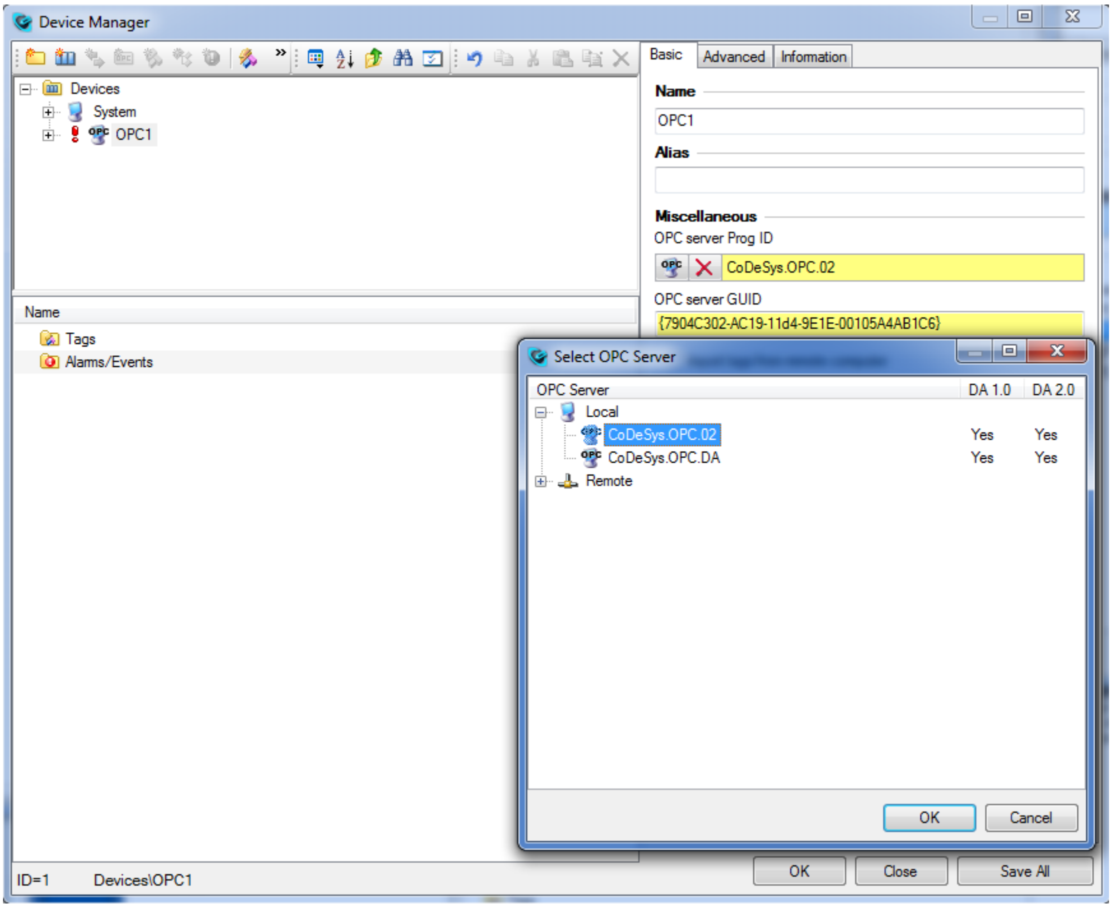


Figure 4.16 Selecting related server in the Device Manager window

After the above- mentioned processes are completed, you should click on the *Import from OPC Server* option which is under the title *Import of Tags* from *Device Manager* window in the Reliance 4 Design program in order to test whether communication is ensured or not. Selecting server in the Device manager window can be seen in Figure 4.16. Thus, you can see if the communication is enabled or not. If communication is enabled, it will be possible to see all the previously defined tags in the opened window. You need to select all the tags on the right side of the window and drag and leave them to the left. Thus, you will make sure that all the defined tags are usable on the SCADA window.

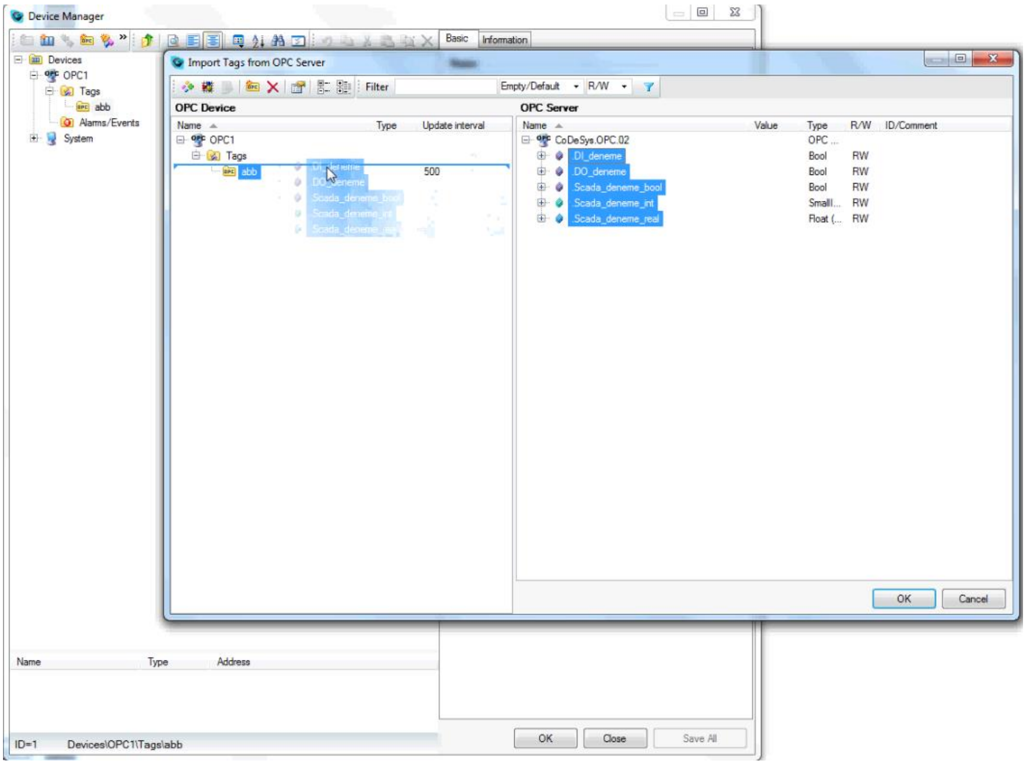


Figure 4.17 Importing tags

After communication is enabled, we can start customizing Reliance 4 Design program within the framework of our Project.

4.2.2 PLC Programming

As mentioned before, there are many different options available in PLC programming. Instead of using "ladder programming" which is the most popular PLC programming language, we used "ST" programming language in this study.

This option was chosen since ST programming language strongly resembles to C programming language and provides convenience for follow-up particularly in long and complicated situations.

When we double click on the menu called AC500 which is found just under the PLC selected using the left window (Devices) on the home page of Control Building Plus programme, "CodeSYS" program will start which is a PLC programming interface is shown in Figure 4.18.

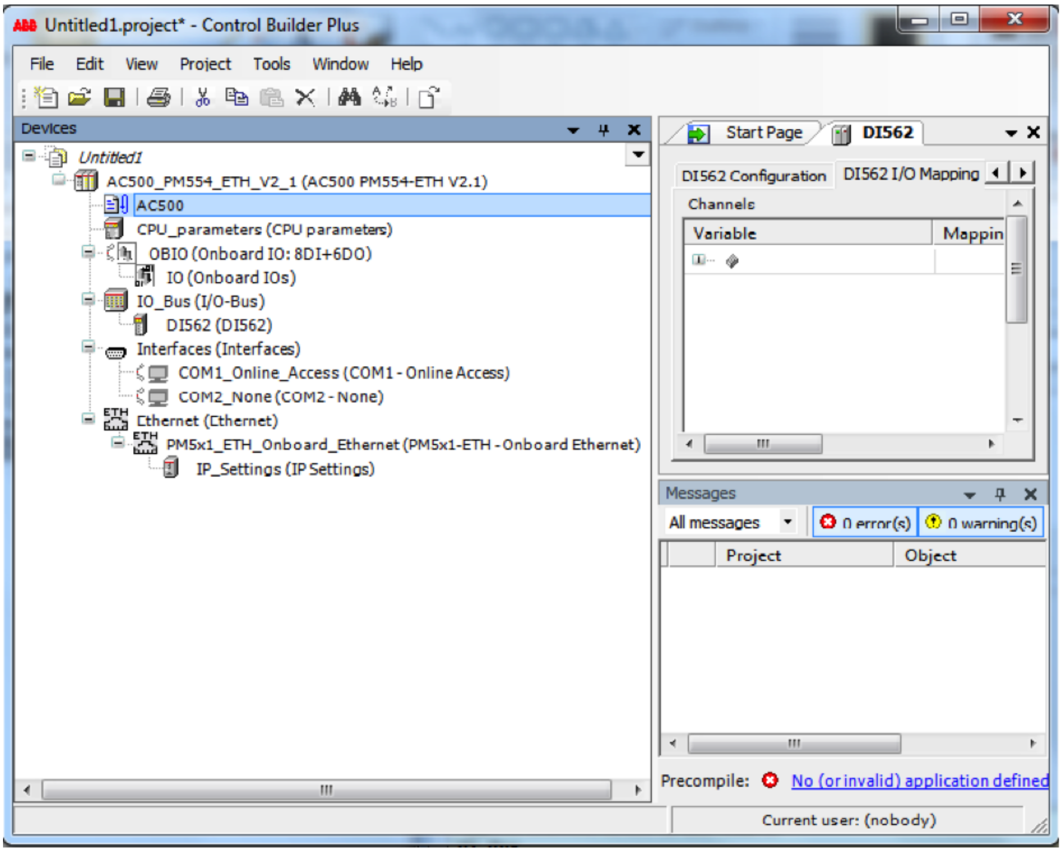


Figure 4.18 Starting the CodeSYS

Encoding can be easily conducted in this interface. Codes in the interface may be saved to PLC or tests may be carried out within the programme before saving the codes. The programming window in CodeSYS is shown in Figure 4.19.

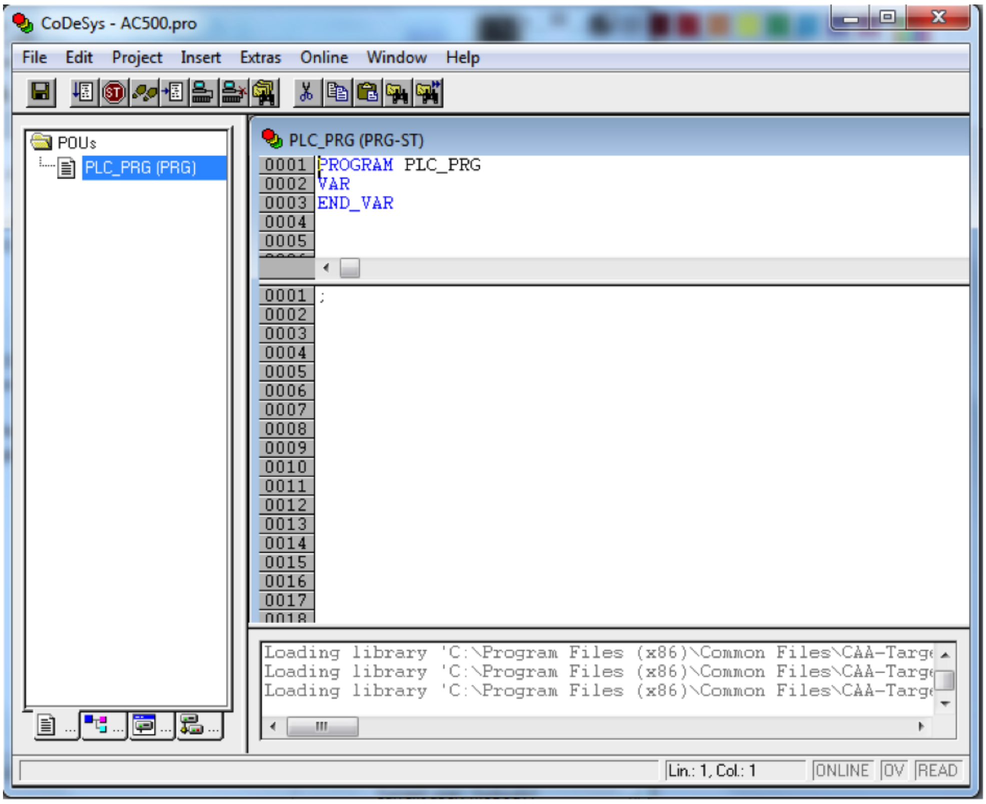


Figure 4.19 Programming window in CodeSYS

4.2.4 SCADA

The SCADA system equipment used in this study is the outcome of a meticulous study. SCADA screen types, communication with the field, software, system features and the equipment used were created considering the requirements of the users of the system, hardware in the plant and system requirements.

4.2.4.1 SCADA Software

“Reliance 4” is a SCADA software package designed for the monitoring and control of industrial processes in real time. “Reliance 4 Design” programme was used in this study as the software. The main page of Reliance is shown in Figure 4.20.

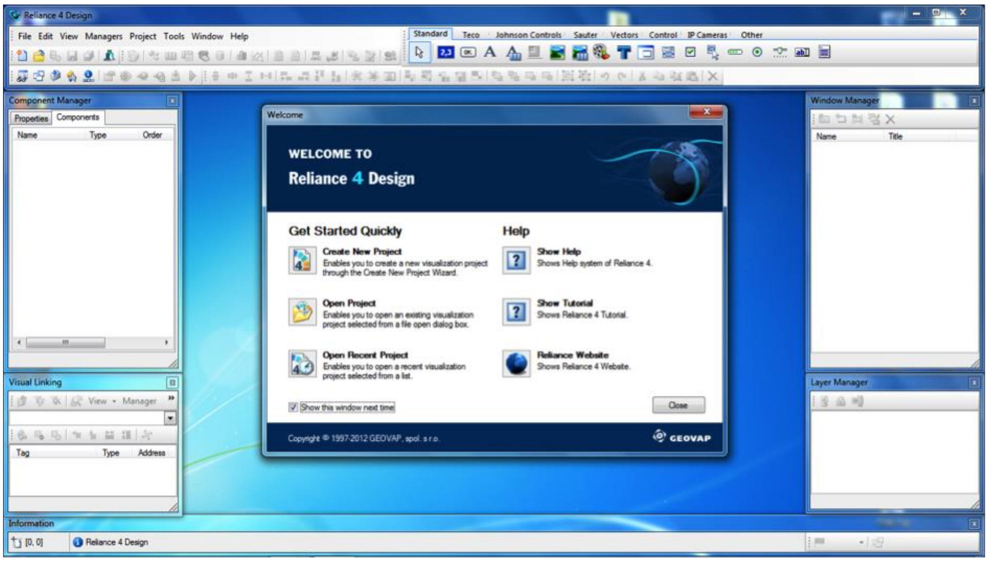


Figure 4.20 Reliance main page

4.2.4.1.1 Creating A New Project

To create a new project, choose *File* then *New Project* command. This will bring up the *Create New Project Wizard* window (Figure 4.21) where you can specify the project name, project encryption, location and the other options for the new project. [10]

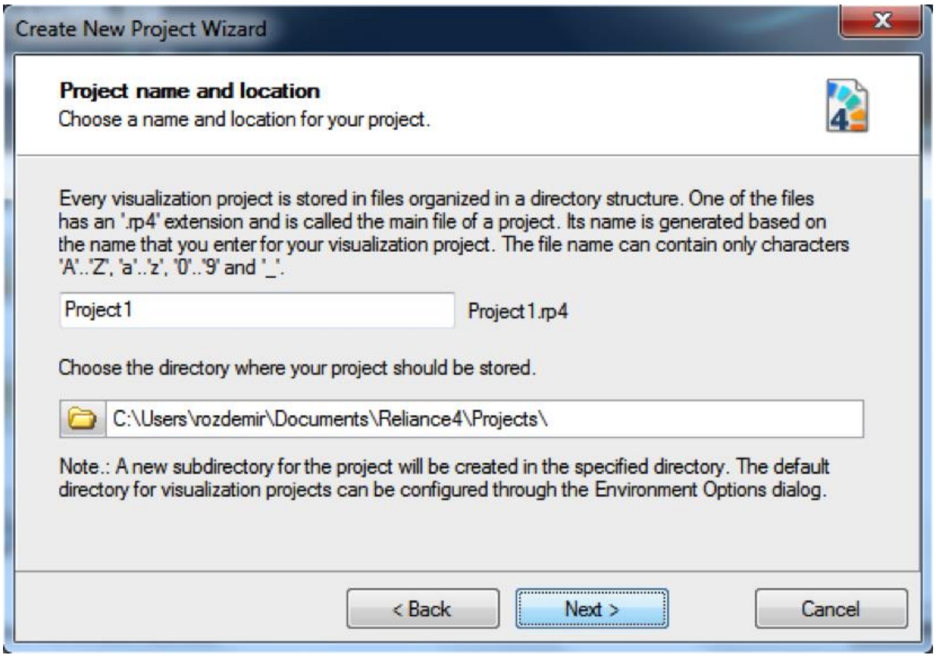


Figure 4.21 Create New Project Wizard window

4.2.4.1.2 Creating a New Window

To create a new window, choose *View* then the *Window Manager* toolbar then select the *New Window* command. You can determine the name of the window to be created, if it should be a normal window or dialogue box and its location on the screen using the window that is created. [10] The window is shown in Figure 4.22.

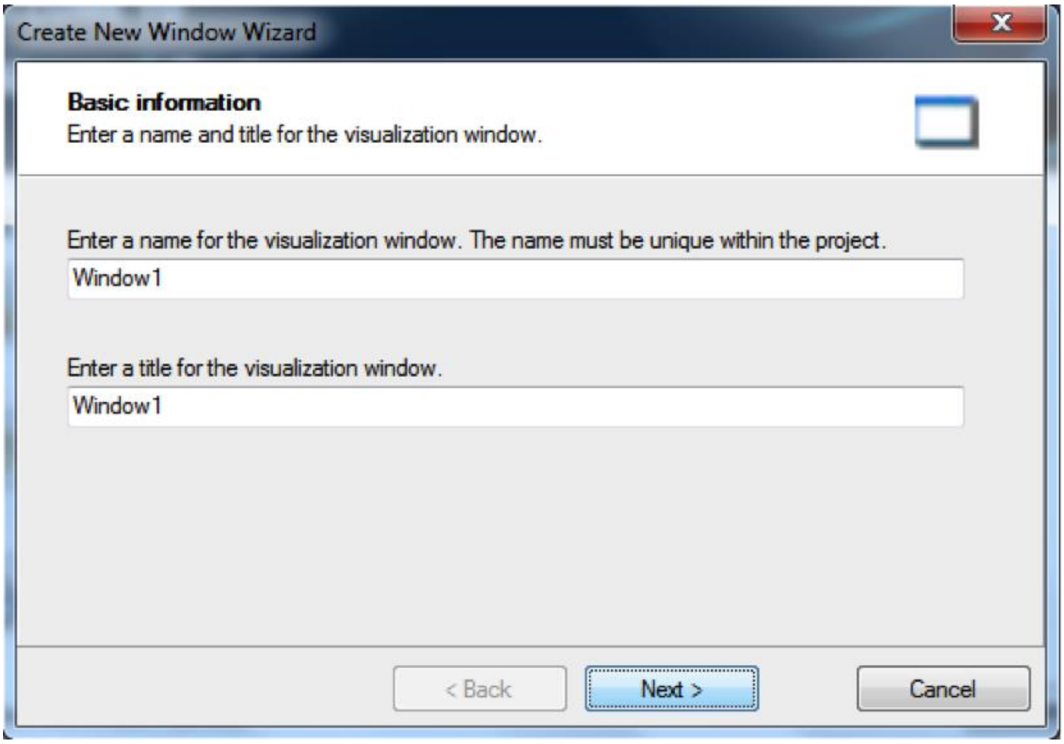


Figure 4.22 Create New Window Wizard window

4.2.4.1.3 Device Manager

When the *Device Manager* command that is located on the *Managers* menu is clicked on, user may define communication zones, alarms, events, tags and devices within the window that opens up. Figure 4.23 shows the Tags in Device Manager.

The command which symbolizes RTU, DCS, PLC or Virtual device is the *device* command. This command allows the user to define tags, alarms and events.

In summary, tags represent areas used in PLC memory in the SCADA program. SCADA program can govern memory areas of the program installed to PLC and input/output units thanks to these tags. There are two types of tags, namely the internal and external tags.

Internal Tags are memory chambers found in Reliance. They perform the same functions as PLC. These tags do not import any date from any address.

External Tag or Process Tags are memory chambers in the project for PLC or similar parts. These tags carry out functions of importing data from the field and transmitting the data to the field.

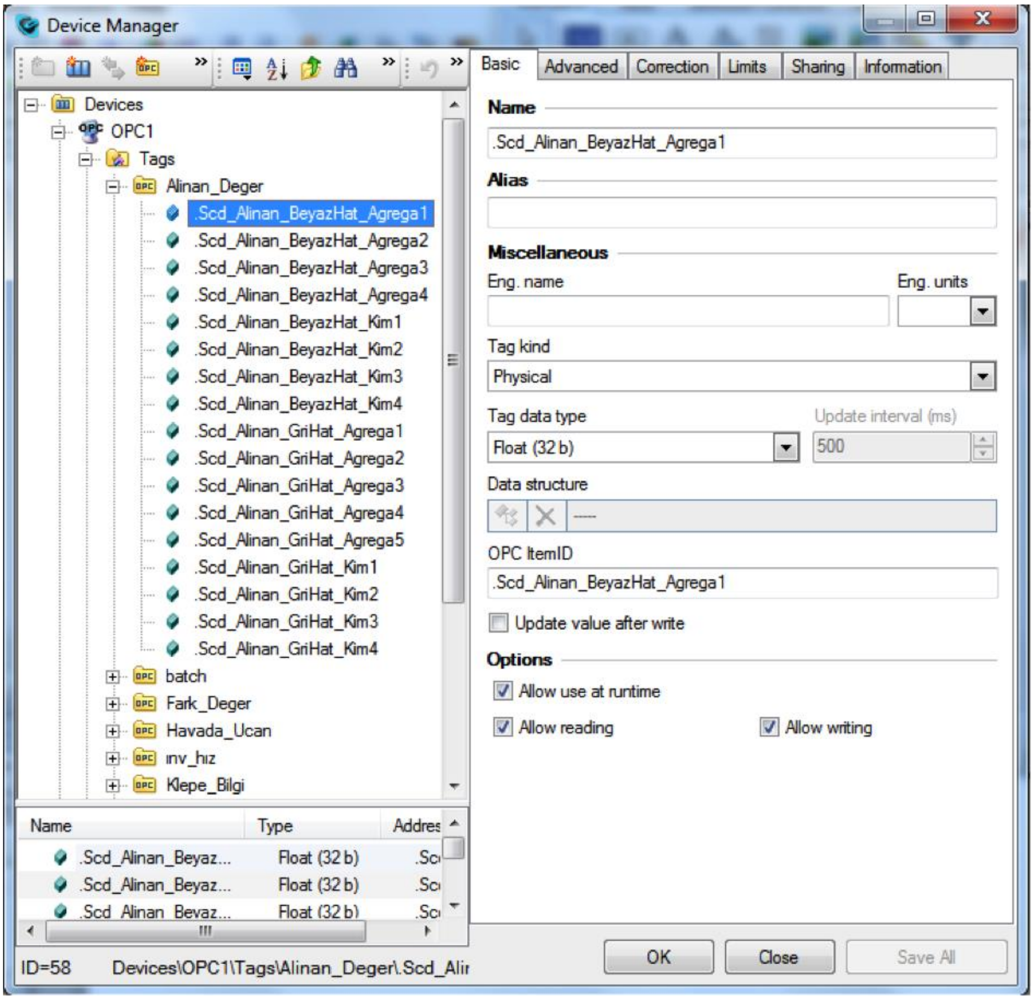


Figure 4.23 Tags in Device Manager

One of the important aspects to be considered at this stage is the fact that the type cannot be changed any more after selecting the desired Device in the New Device segment; because different types of devices have different device properties. Each device has parts called tags, alarms and events.

4.2.4.1.4 Managers Menu

It is possible to have access to many useful properties using *Managers* menu. These properties are among the ones most frequently used by software developers when customizing the SCADA program. Managers Menu is shown below in Figure 4.24.

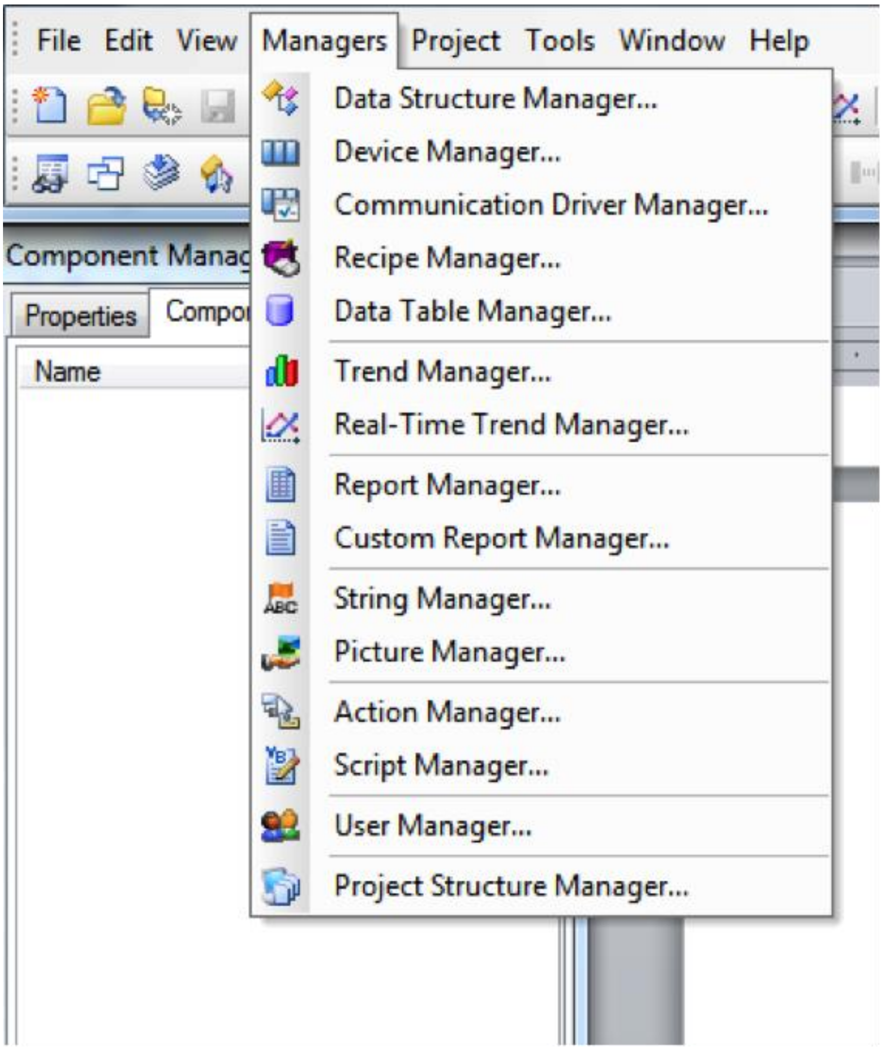


Figure 4.24 Managers Menu

4.2.4.1.5 The Data Structure Manager

This property allows the user to configure and define data structures and conduct import and export functions. The window is shown in Figure 4.25.

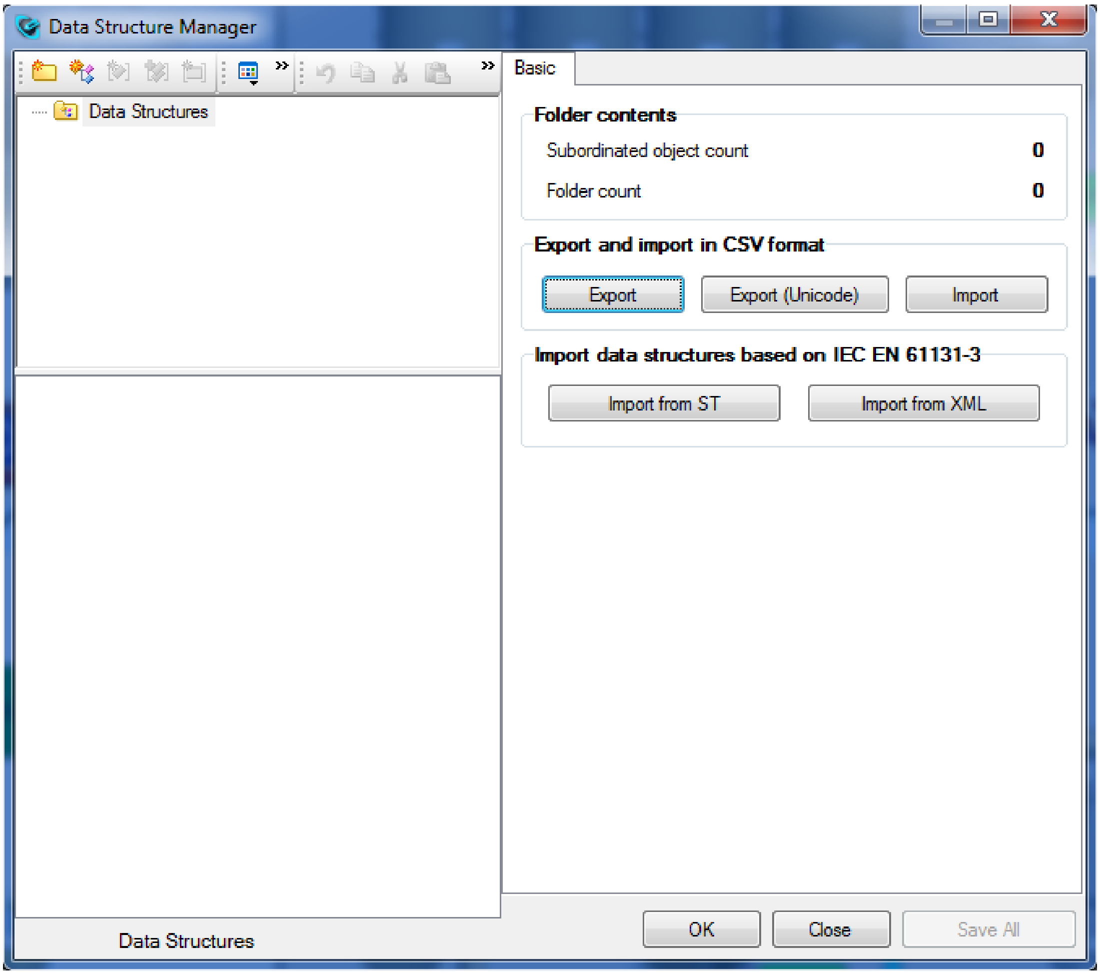


Figure 4.25 The Data Structure Manager window

4.2.4.1.6 The Communication Driver Manager

This property is used to adjust and configure the communication settings of drivers that are compatible with Reliance. The Communication Driver Manager window can be seen in Figure 4.26.

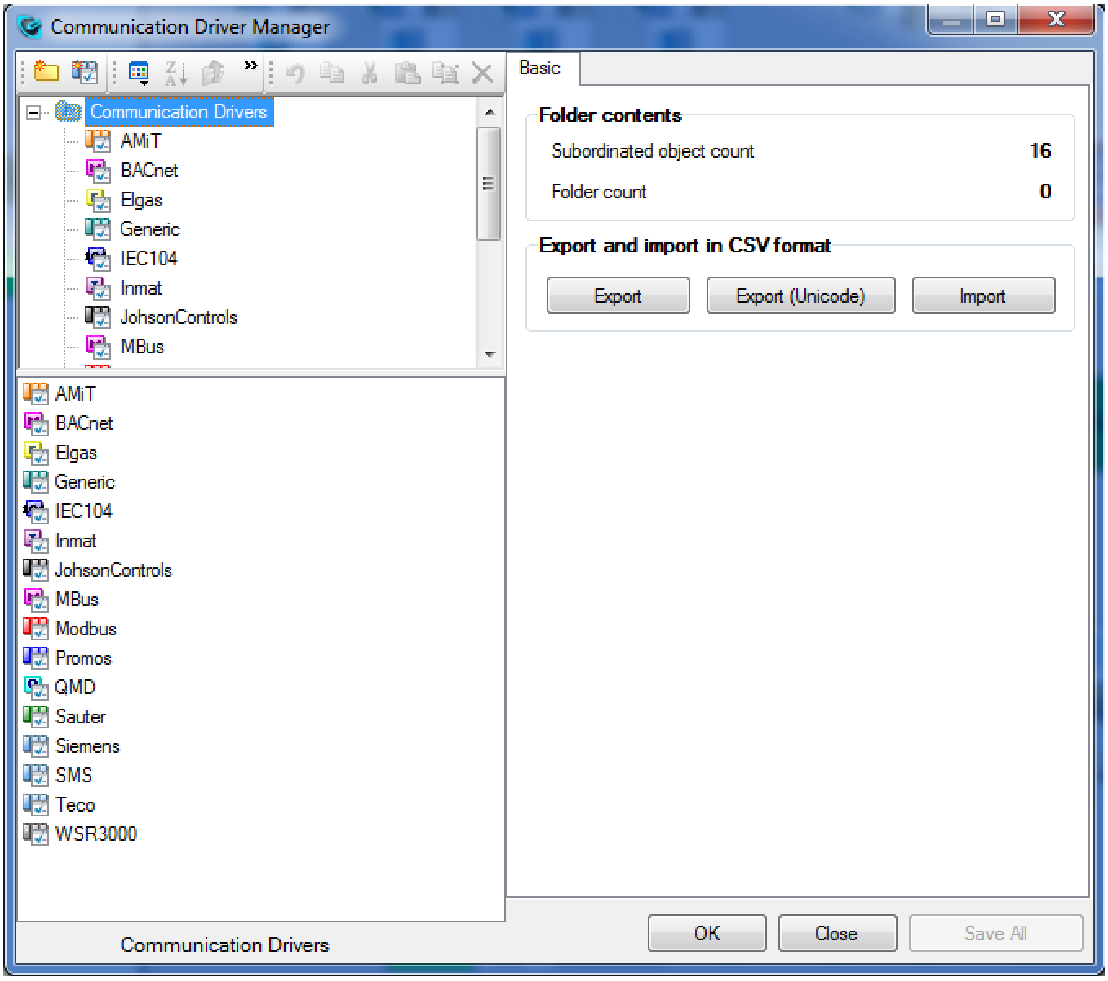


Figure 4.26 The Communication Driver Manager window

4.2.4.1.7 The Recipe Manager

This property allows the user to define and configure recipes in accordance with the desired conditions. When a recipe is created, tags that will be used by the recipe are selected and values of such tags are stored for a certain period of time. Recipes can be recalled by the program users when desired. Figure 4.27 shows the Recipe Manager Window and Figure 4.28 shows a sample recipe window.

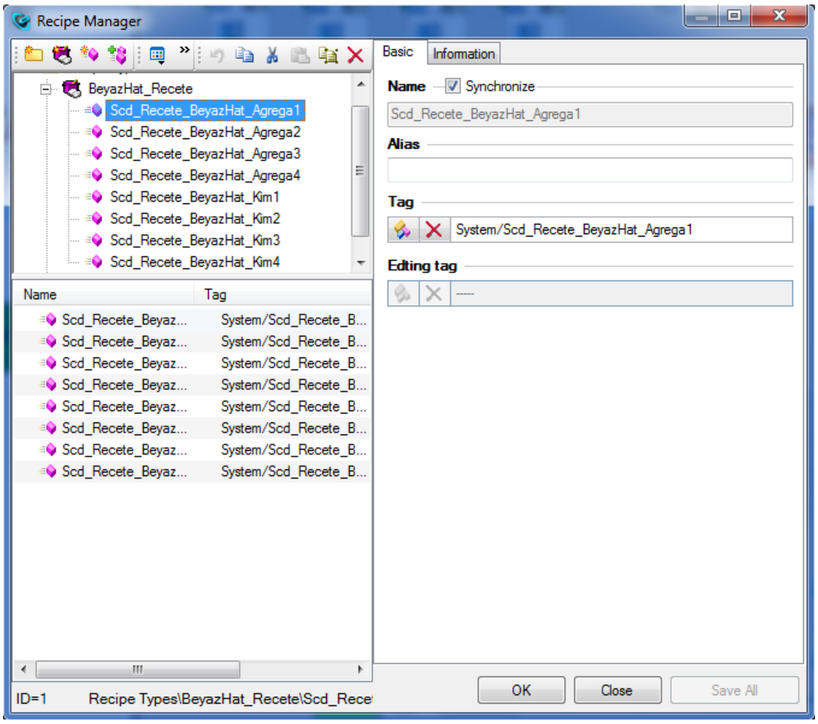


Figure 4.27 The Recipe Manager window

Figure 4.28 Sample recipe window

4.2.4.1.8 The Data Table Manager

Data table is created to enable access to historical data. Logged tags need to be identified to enable this access. The window is shown in Figure 4.29 below.

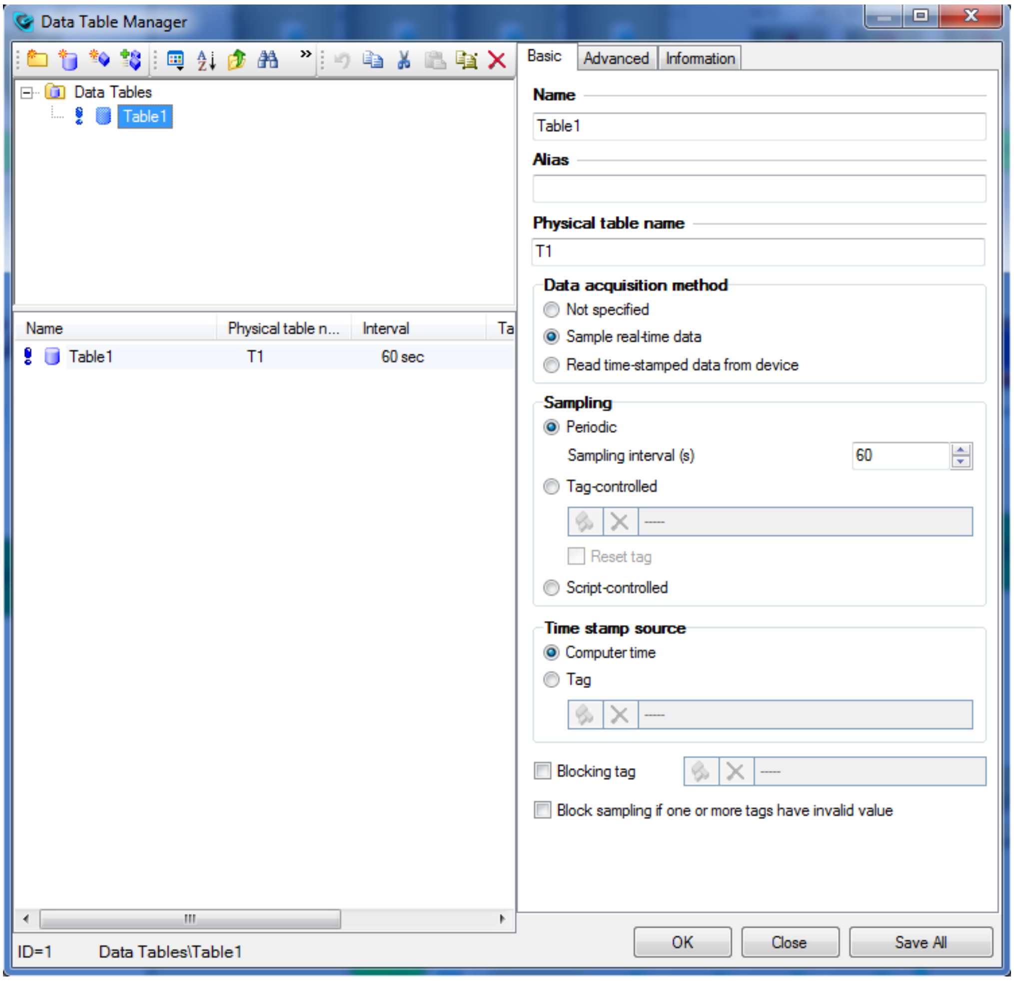


Figure 4.29 The Data Table Manager window

4.2.4.1.9 The Trend Manager

Trends are structures which constitute graphic presentation of the historical data. Trend Manager is used to define and configure trends as desired and can be seen in Figure 4.30. User may change many visuals under trends. Properties such as font, color, background color, title etc. in addition to the length and type of trend content are among the properties which can be selected. [10]

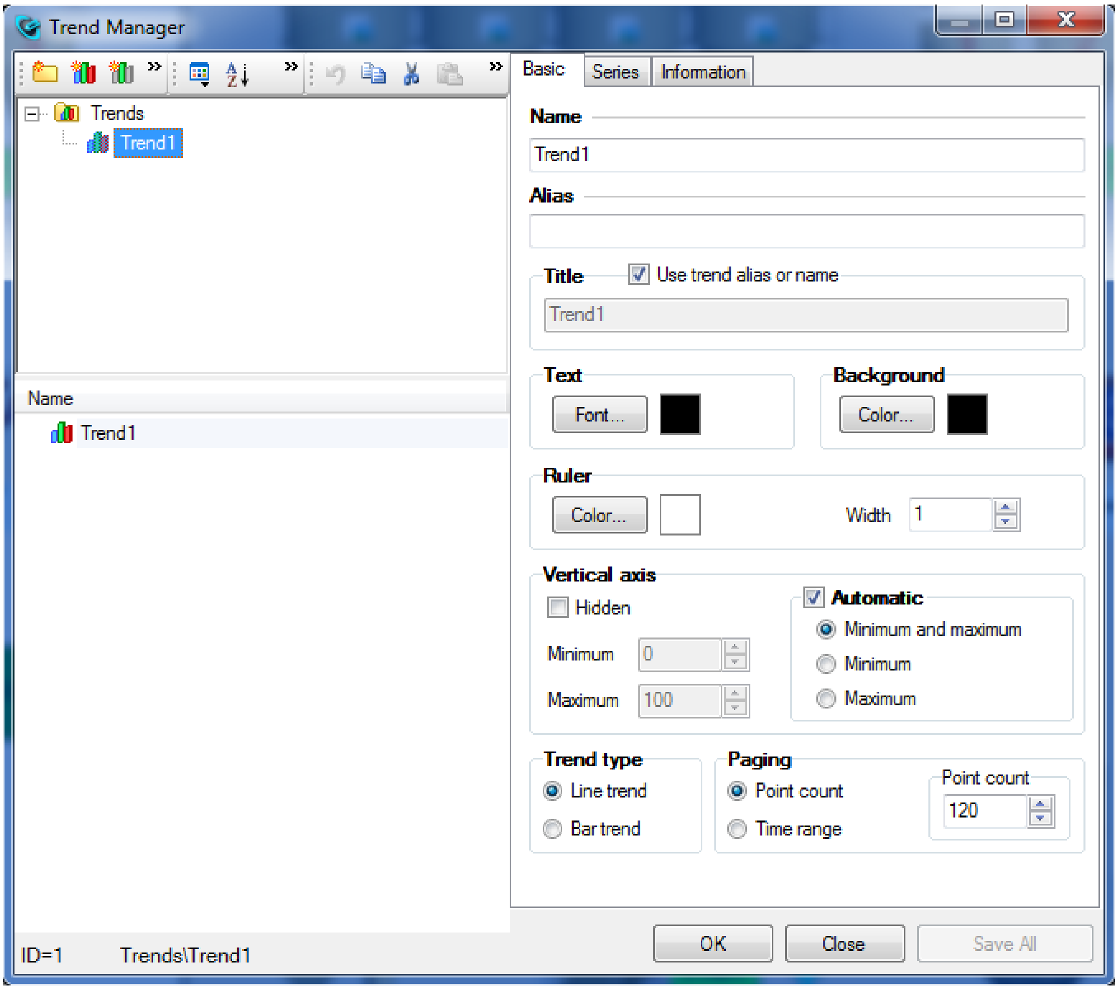


Figure 4.30 The Trend Manager

4.2.4.1.10 The Report Manager

This manager is a property which enables users the graphic and statistical presentation of data and is shown in Figure 4.31. It is possible to configure reports depending on what the users desire. Reports can be printed, saved in different format or solely displayed on the screen. The property called *Custom Report Manager* has same features as *Report Manager*, however it imports the report content in the format of *Fast Report, Text or Web Page*. [10]

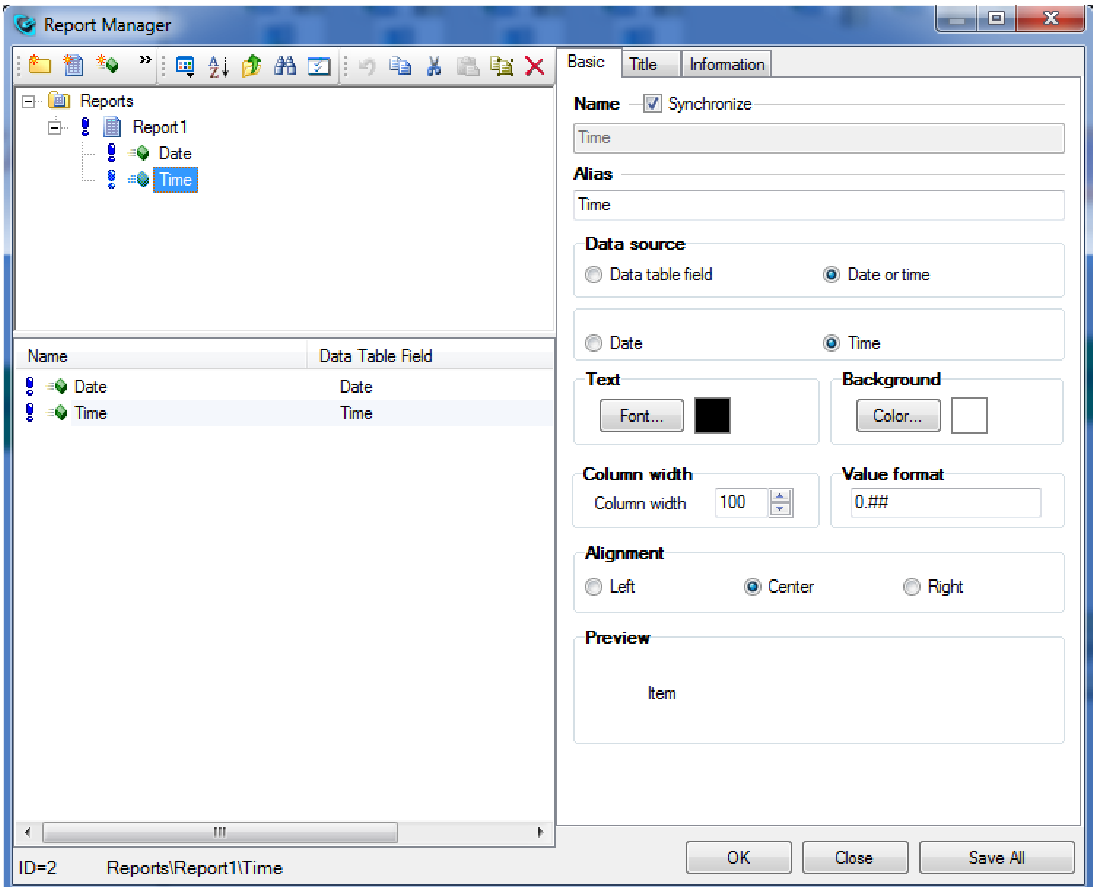


Figure 4.31 The Report Manager window

4.2.4.1.11 The String Manager

This manager allows you to be informed on the strings used in the project and manage them. The String manager window can be seen in Figure 4.32. If a string is modified via the String manager, all strings having the same name in the current project are modified and updated. In addition, the string manager has many options for languages to be used, which allows strings to be written in any desired language. [10]

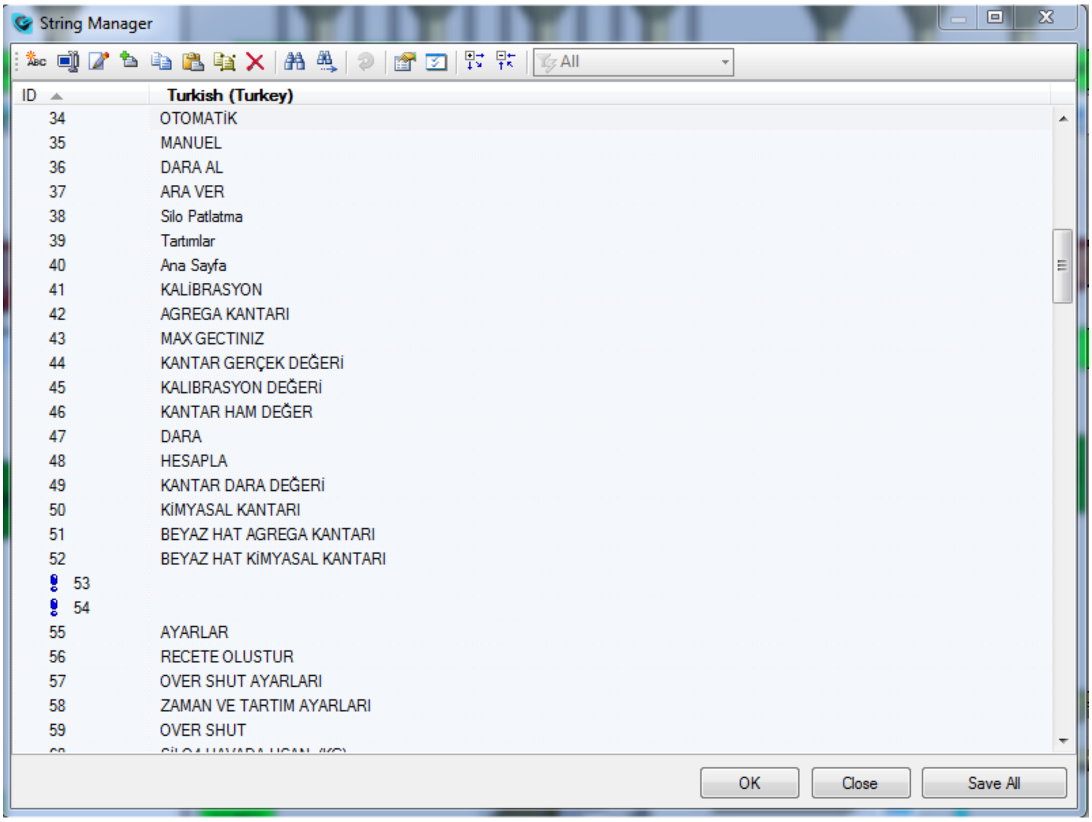


Figure 4.32 The String Manager window

4.2.4.1.12 The Picture Manager

It allows you to manage pictures used in a project. Pictures can be imported in different formats from any location on the computer. Once a picture has been imported, it can be displayed on any place in the visualization through the appropriate component (graphical object). The Figure 4.33 shows the Picture Manager window. Reliance 4 ships with a graphics library containing a lot of useful graphics to be used in visualization projects. [10]

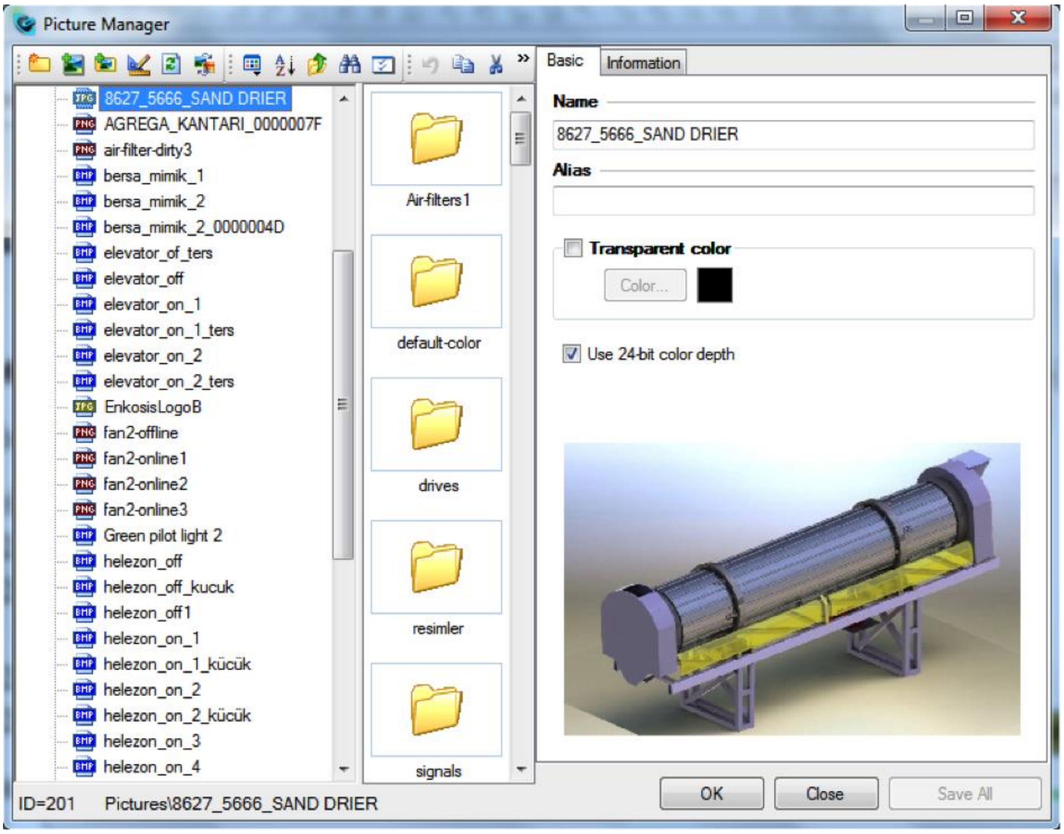


Figure 4.33 The Picture Manager Window

4.2.4.1.13 The Action Manager

Action Manager can carry out functions such as activating a window, showing a trend, terminating the program, displaying the recipes and displaying the reports. The Figure 4.34 shows The Action Manager window. [10]

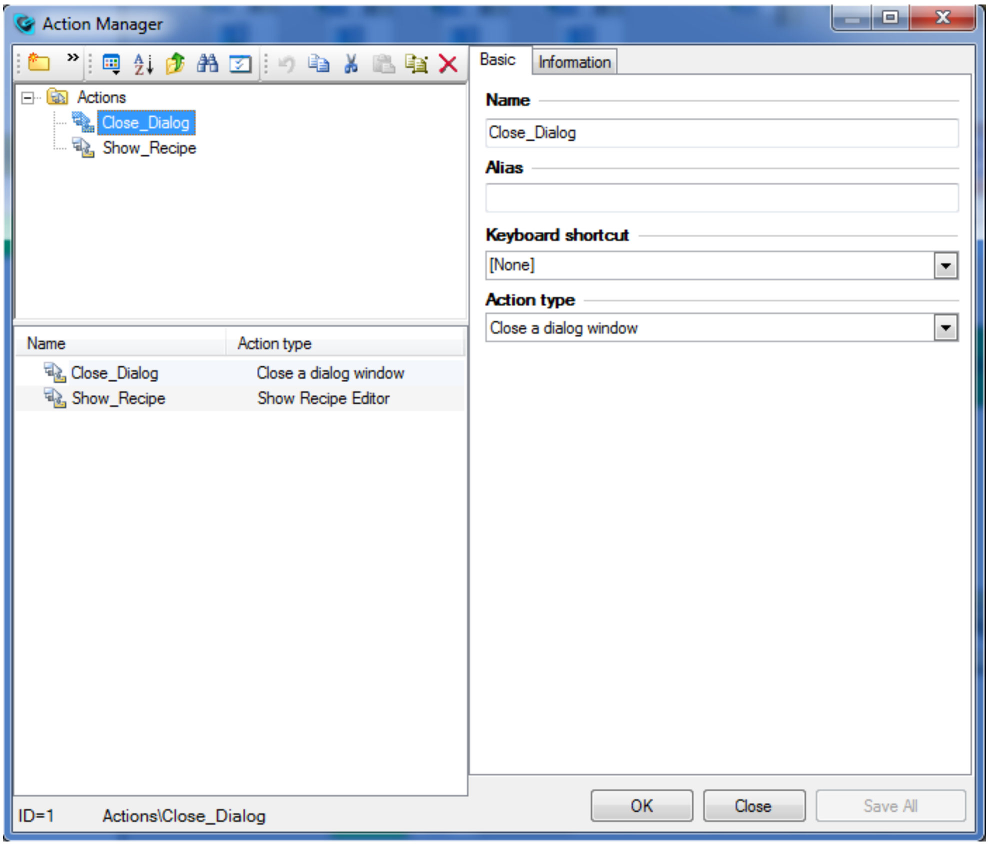


Figure 4.34 The Action Manager window

4.2.4.1.14 The Script Manager

This manager allows you to define and configure scripts. Figure 4.35 shows window.

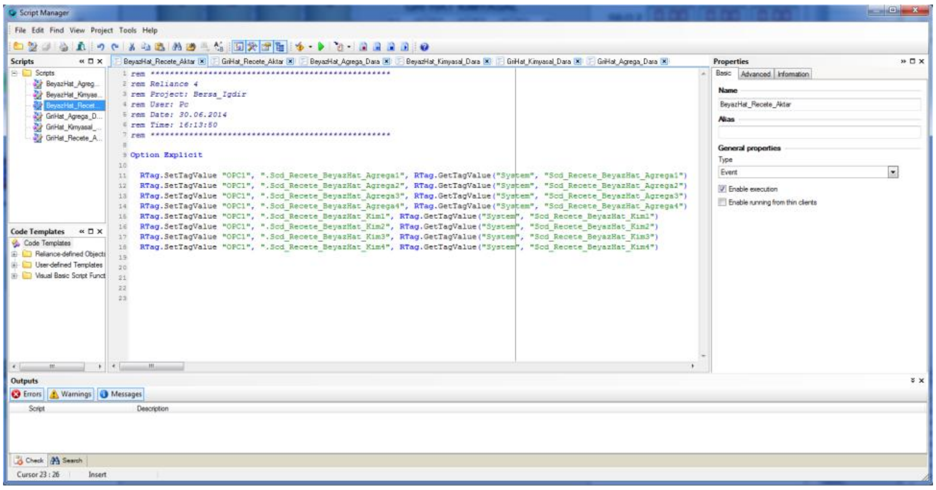


Figure 4.35 The Script Manager window

4.2.4.1.15 The User Manager

This manager is used to define and authorize users and can be seen in Figure 4.36. Options are created using this manager to define which user will be assigned to have access to which functions in the project, which user will be given full authority to use all functions or solely the authority to view. [10]

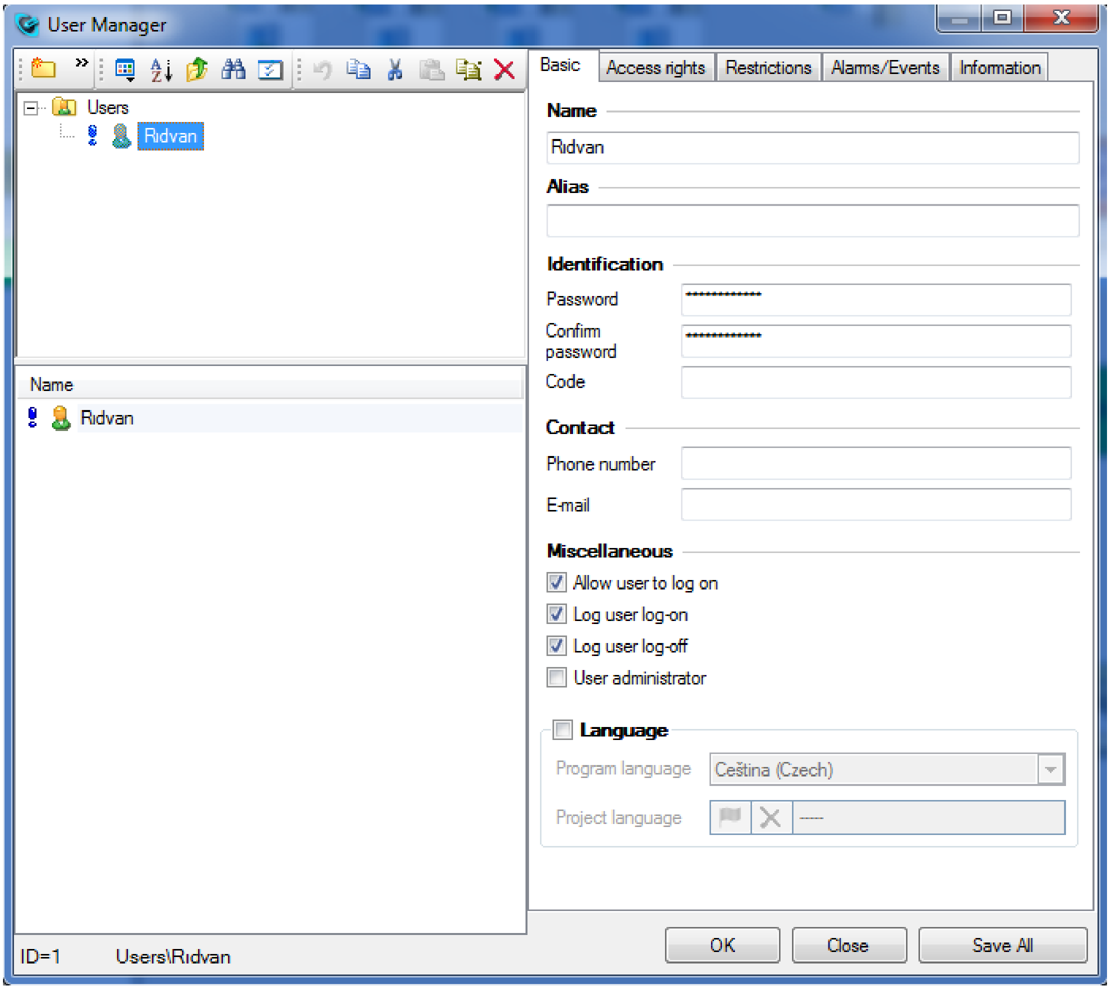


Figure 4.36 The User Manager window

4.2.4.1.16 Running A Project

To run the project, choose the *Project* menu and the choose *Start* command.

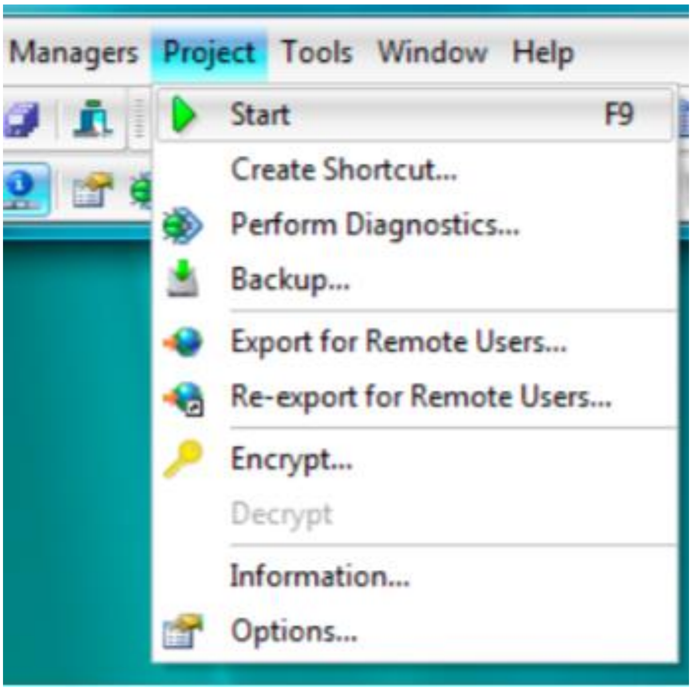


Figure 4.37 Project menu

The software launches automatically after clicking on Start button and the project starts. To exit the software, choose *File* menu and then choose *Exit* command under this menu. Project menu window is shown in Figure 4.37 above.

4.2.5 Building up Main Window and Related PLC Codes

Operators are able to monitor all the equipments and functioning in a factory on a sole screen and automatically or manually control all equipments and the system along with receiving real time or previous reports and statistics and form and enter prescriptions and do various adjustments for system functioning through the SCADA screen provided in Figure 4.38.

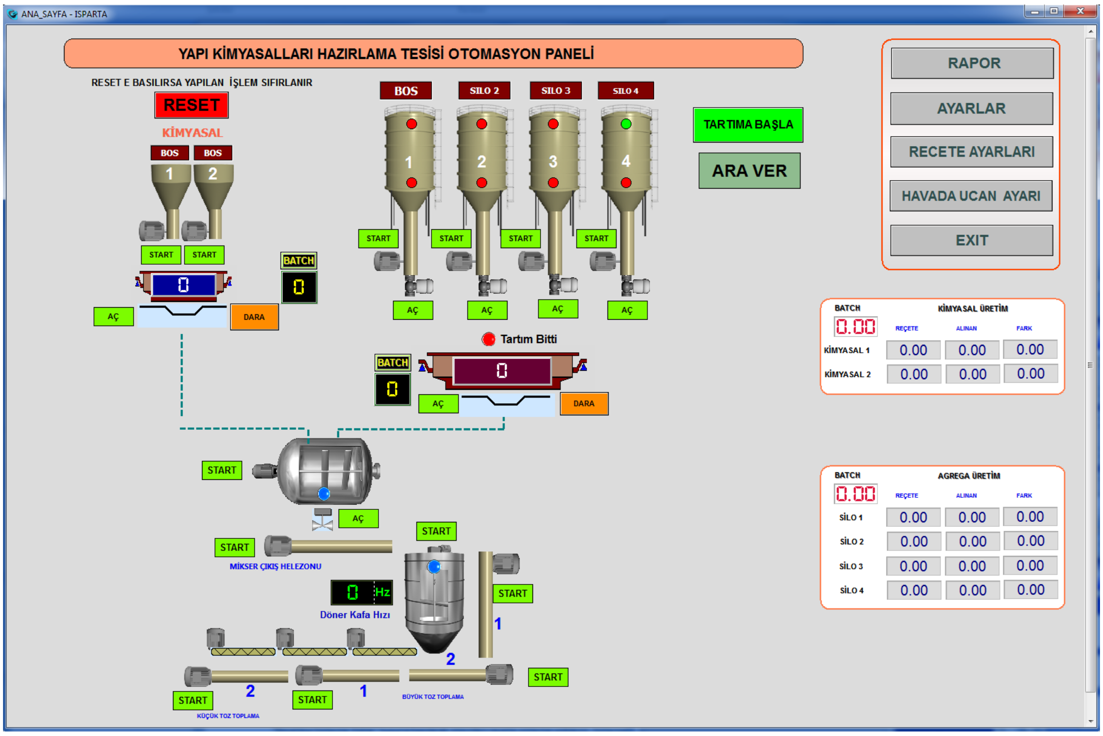


Figure 4.38 SCADA main screen

In short, the system functions as follows: there are two groups of silos in the system. In the 1st group there are 2 silos which have chemical materials inside. Other group has 4 silos and is named as aggregate. These silos have structural materials. When the operator selects one of the existing prescriptions or on his/her own will adjusts and presses “start weighing” button and if all the other conditions are provided (that is, silos have the materials and system has power etc.) weighing starts.

Weighing starts simultaneously for two groups. Weighing processes continue by weighing from only one silo each time until all the silos are weighed and prescriptions which show how much materials will be received from each silo are taken into account.

After weighing is finished, materials on conveyors are poured into the mixer one by one starting from the material in the aggregate silos. Upon pouring process is completed, mixer starts to rotate and carries on until the pre-defined mixing time is over.

If there are more than one “batch”, just after the mixer starts to run, weighing in silos restart to avoid loss of time.

When mixing process is completed, the valve under the mixer is opened and the mixed material is poured into the conveyor. Materials are then poured into the rotating head from this conveyor. However, there is a precondition for this process to continue, which is the information about the fullness. If the information about the fullness is received, materials in the conveyor are suspended and not poured into the rotating head. If the information about the fullness is not received for 15 seconds then conveyors are run and materials are poured into the rotating head. This process carries on until the new information about the fullness is received or all the material is poured into the rotating head.

In the meanwhile, second group of materials whose weighing process has been completed and which is hold in the conveyor is poured into the mixer when it is empty.

First group of materials which has been mixed in the rotating head for a while is then poured from the rotating head to be packed later. Packaging process is semi-automatic. Packaging is completed through conveyors and workers’ help.

4.2.5.1 Weighing Starts

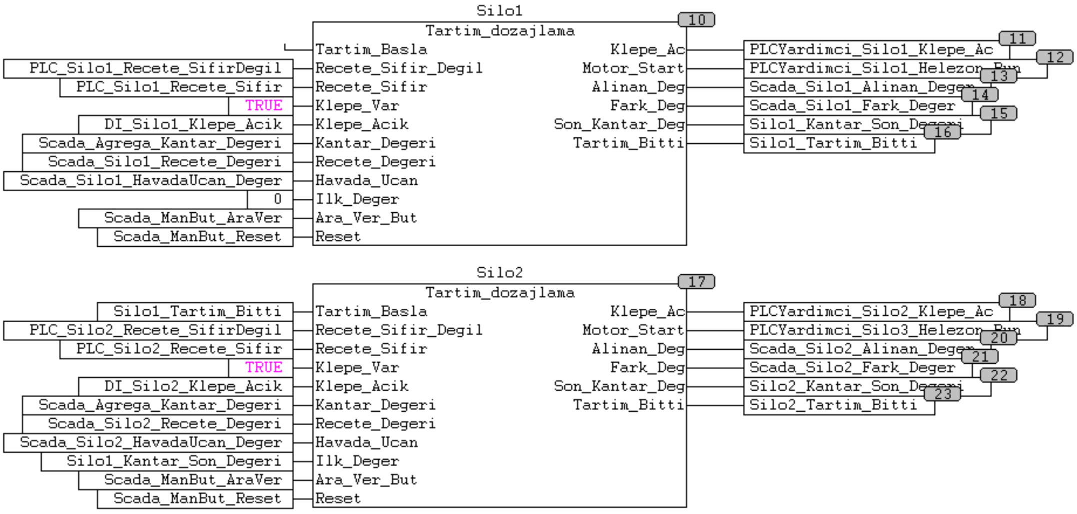


Figure 4.39 Function block of the weighing

After “start weighing” command is received, weighing process in silos starts respectively as mentioned before. In the Figure 4.39 above, you can see the PLC program which is written in order to control weighing process of the function blocks. As the bits seen at the left side of the block are sets, bits at the right side form sets in turn. As seen on the right side, first silo valve is opened to start weighing. Then, spiral shaped engine is run. Received value, the difference between and scale value information are identified and weighing stops when the desirable value is obtained.

Silos can work in turn through setting ‘Silo1_Tartım_Bitti’ (Silo1_Weighing_Finished) bit as one of the inlet bits of Silo2 block. Similarly ‘Silo2_Tartım_Bitti’ bit is a requisite to start Silo3 block.

This is how it works inside the blocks:

```
IF Tartim_Basladi AND Tartim_Step = 0 AND Recete_Sifir_Degil THEN
        Tartim_Step := 1 ;
END_IF


IF Tartim_Basladi AND Tartim_Step = 0 AND Recete_Sifir THEN
        Alinan_Deg:=0;
        Fark_Deg:=0;
        Son_Kantar_Deg:= Ilk_Deger;
        Tartim_Bitti:=TRUE;
        Tartim_Basladi:=FALSE;
END_IF
```

```
IF  Tartim_Step = 1  THEN
        Hedef_Deger := ( Recete_Degeri + Kantar_Degeri ) - Havada_Ucan ;
        Tartim_Step := 2 ;
END_IF

IF Tartim_Step = 2 AND Klepe_Var  THEN
        Klepe_Ac :=TRUE;
        Tartim_Step := 3 ;
END_IF

IF Tartim_Step = 3 AND NOT Klepe_Var  THEN
        Motor_Start :=TRUE;
        Tartim_Step := 4 ;
END_IF

IF Tartim_Step = 4 AND Hedef_Deger>= Kantar_Degeri AND NOT Motor_Start AND Klepe_Var AND NOT
Klepe_Ac  THEN
        Motor_Start :=TRUE;
        Klepe_Ac  :=TRUE;
        Tartim_Step := 4 ;
END_IF

IF Tartim_Step = 4 AND Kantar_Degeri>= Hedef_Deger  THEN
        Motor_Start :=FALSE;
        Klepe_Ac  :=FALSE;
        Tartim_Step := 5 ;
END_IF

IF Tartim_Step = 5 AND NOT Motor_Start AND NOT Klepe_Ac THEN
        Tartim_Step := 6 ;
END_IF

TON1(IN:= Tartim_Step = 6 , PT:= T#2s);

IF Tartim_Step = 6 AND  TON1.Q  THEN
        Tartim_Step := 7 ;
END_IF

IF Tartim_Step = 7   THEN
        Alinan_Deg := ( Kantar_Degeri - Ilk_Deger);
        Tartim_Step := 8 ;
END_IF

IF Tartim_Step = 8   THEN
        Son_Kantar_Deg := ( Alinan_Deg + Ilk_Deger);
        Tartim_Step := 9 ;
END_IF

IF Tartim_Step = 9   THEN
        Fark_Deg := ( Recete_Degeri - Alinan_Deg);
        Tartim_Step := 10 ;
END_IF


IF Tartim_Step = 10   THEN
        Tartim_Basladi:=FALSE;
        Tartim_Bitti := TRUE;
        Tartim_Step := 0 ;
END_IF

TON2(IN:= Tartim_Bitti, PT:= T#1s);

IF TON2.Q THEN
        Tartim_Bitti := FALSE;
END_IF
```

Step 4 is important in this code apart from security issues. Our target value, that is, the value desired in the prescription, is compared to the scale value. This loop is completed when scale value becomes equal to or a little bit more than the target value, which shows that the desired weighing has finished.

4.2.5.2 Pouring from the scale to the mixer

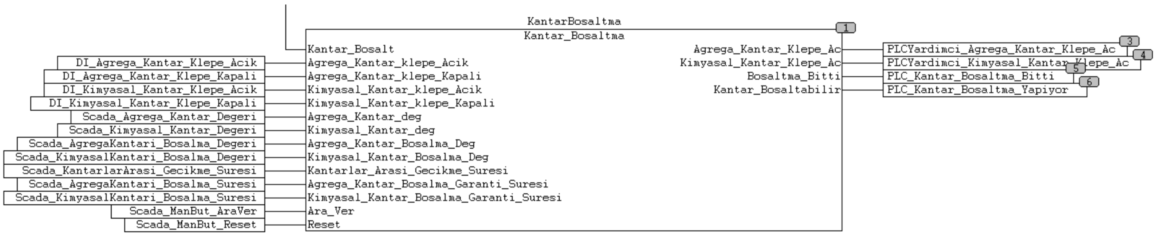


Figure 4.40 Function block of the pouring process to the mixer

One of the most important issues is the necessity to pour aggregate materials first during the process of pouring the materials to the mixer. As chemical materials are adhesive and fluid, we do not want them to be poured from the mixer first and stick to the surface of the mixer. Therefore aggregates are completed first. Moreover, when materials in the scale reach a certain level (to the basis weight determined by the operator or 100 grams etc.), scale valve is suspended for as long as desired and then closed. The reason why is that materials that can stay on the surface of the scale even if the valve is open should not cause the process to enter into an infinite loop. After the valves are closed the mixer starts the mixing process.

This is how it works inside the blocks:

```
IF Kantar_Step = 0 AND Kantar_Bosaltabilir THEN

        Agrega_Kantar_Klepe_Ac := TRUE;
        Kantar_Step := 1;
END_IF

IF Kantar_Step = 1 AND T_Kantarlar_Arasi_Gecikme.Q THEN

        Kimyasal_Kantar_Klepe_Ac := TRUE;
        Kantar_Step := 2;
END_IF

IF Kantar_Step = 2 AND Agrega_Kantar_klepe_Acik AND Agrega_Kantar_Bosalma_Deg>=Agrega_Kantar_deg
THEN

        Agrega_Kantar_Garanti_Sure_Baslat := TRUE;
END_IF
```

```
IF Kantar_Step = 2 AND DI_Kimyasal_Kantar_Klepe_Acik AND Kimyasal_Kantar_Bosalma_Deg>=
Kimyasal_Kantar_deg THEN

        Kimyasal_Kantar_Garanti_Sure_Baslat := TRUE;
END_IF

IF Kantar_Step = 2 AND T_Agrega_Kantar_Garanti_Sure.Q THEN

        Agrega_Kantar_Klepe_Ac := FALSE;
END_IF

IF Kantar_Step = 2 AND T_Kimyasal_Kantar_Garanti_Sure.Q THEN

        Kimyasal_Kantar_Klepe_Ac := FALSE;
END_IF

IF Kantar_Step = 2 AND Bitirme_Ton.Q THEN

        Agrega_Kantar_Garanti_Sure_Baslat := FALSE;
        Kimyasal_Kantar_Garanti_Sure_Baslat := FALSE;
        Bosaltma_Bitti := TRUE;
        Kantar_Step := 3;
END_IF

IF Kantar_Step = 3 AND T_Bitti.Q THEN

        Bosaltma_Bitti := FALSE;
        Kantar_Bosaltabilir := FALSE;
        Kantar_Step := 0;
END_IF

ELSE

        Agrega_Kantar_Garanti_Sure_Baslat := FALSE;
        Kimyasal_Kantar_Garanti_Sure_Baslat := FALSE;
        Agrega_Kantar_Klepe_Ac := FALSE;
        Kimyasal_Kantar_Klepe_Ac := FALSE;
        Kantar_Step := 0;
END_IF

IF Reset THEN

        Agrega_Kantar_Garanti_Sure_Baslat := FALSE;
        Kimyasal_Kantar_Garanti_Sure_Baslat := FALSE;
        Agrega_Kantar_Klepe_Ac := FALSE;
        Kimyasal_Kantar_Klepe_Ac := FALSE;
        Kantar_Step := 0;
        Bosaltma_Bitti := FALSE;
        Kantar_Bosaltabilir := FALSE;
END_IF
```

4.2.5.3 Blending Process in the Mixer

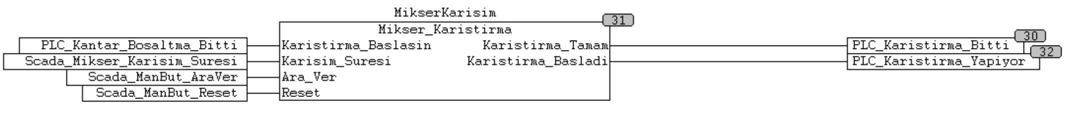


Figure 4.41 Function block of the blending process

After the scale is poured into the mixer, the blending process in the mixer continues as long as it is prescribed. This process is a pre-requisite of the process of pouring from the mixer. No pouring from the mixer can take place before the mixing is completed. Figure 4.41 shows the function block of the blending process above.

The code inside the mixing block of the mixer is as follows:

```
R_Karistirma(CLK:=Karistirma_Baslasin );

IF R_Karistirma.Q THEN

        Karistirma_Basladi := TRUE;
        Karistirma_Step := 0 ;
END_IF

IF Karistirma_Step = 0 AND Karistirma_Basladi THEN

        Karisim_Suresi_Baslasin := TRUE;
        Karistirma_Step := 1 ;
END_IF

T_Karisim_Suresi(IN:= (Karistirma_Step = 1 AND Karisim_Suresi_Baslasin AND NOT Ara_Ver ), PT:=
DWORD_TO_TIME(REAL_TO_DWORD(Karisim_Suresi *60000)));

IF  Karistirma_Step = 1 AND T_Karisim_Suresi.Q THEN

        Karistirma_Tamam := TRUE;
        Karistirma_Basladi := FALSE;
        Karisim_Suresi_Baslasin := FALSE;
        Karistirma_Step := 0;
END_IF

T_KarisimTamam(IN:= Karistirma_Tamam , PT:= T#1s);

IF T_KarisimTamam.Q THEN

        Karistirma_Tamam := FALSE;
END_IF

IF Reset THEN

        Karistirma_Tamam := FALSE;
        Karistirma_Basladi := FALSE;
        Karisim_Suresi_Baslasin := FALSE;
        Karistirma_Step := 0;
END_IF
```

4.2.5.4 Pouring Process from the Mixer

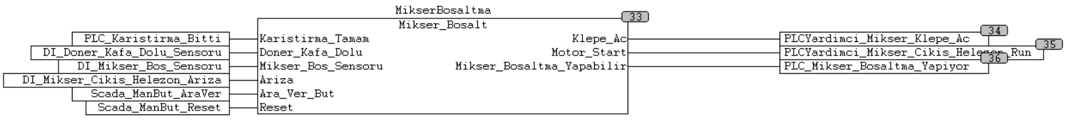


Figure 4.42 Function block of the pouring process from the mixer

Process of pouring from the mixer starts after the mixing is finished. The pre-requisites for this process include running the spiral shaped engine, not having pressed the button of “pause”, not having pressed the button of “reset” and seeing that rotating head full sensor is not activated. Otherwise, pouring process does not start. The function block of the pouring process from the mixer is shown in Figure 4.42 above.

The code inside the mixer pouring block is as follows:

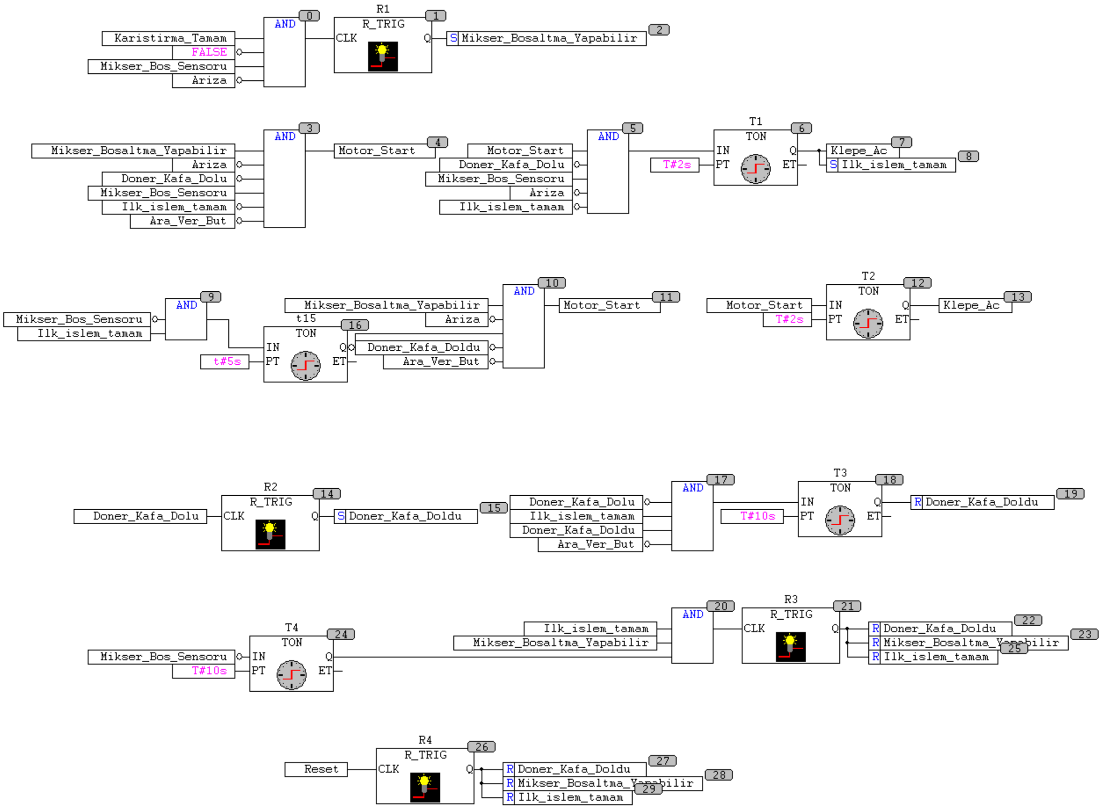


Figure 4.43 Code inside the mixer pouring function block

The above code in The Figure 4.43 shows the code inside the function block of the mixer pouring process. The most important part of this code is the information communicated by the rotating head full sensor. The “bit 1” information provided by this sensor stops the process. When the position is changed from 1 to 0, the system waits for 10 seconds before it restarts pouring process. If the full sensor still remains in the position of 0 in this time slot, the system resumes pouring process.

4.2.5.5 Rotating Head, Dust Absorption and Packaging Processes

If the engine of rotating head is not defective, the process takes place and remains in the position of “run”. The “run” position here means that the spiral shaped engine of the rotating head is run. After the spiral shaped engine is run, this process is followed by dust absorption process and running the conveyors which will carry out the packaging. Operator needs to issue a command for these processes. There are two dust collecting systems and 3 conveyor bands within the system.

The code which is needed by these processes is as follows:

```
R_DonerKafa(CLK:= Scada_ManBut_DonerKafa_Mot AND NOT DI_Doner_Kafa_Ariza );

F_DonerKafa(CLK:=  Scada_ManBut_DonerKafa_Mot AND NOT DI_Doner_Kafa_Ariza );

IF R_DonerKafa.Q THEN

        DO_Doner_Kafa_Run := TRUE;
END_IF

IF F_DonerKafa.Q THEN

        DO_Doner_Kafa_Run := FALSE;
        Scada_ManBut_DonerKafa_Mot  := FALSE;
END_IF

(*          *)

R_KucukTozToplama1(CLK:= Scada_ManBut_KucukTozToplama1_Mot AND NOT DI_Kucuk_Toz_Toplama1_Ariza AND
NOT DI_Kucuk_Toz_Toplama2_Ariza );

F_KucukTozToplama1(CLK:=   Scada_ManBut_KucukTozToplama1_Mot AND NOT DI_Kucuk_Toz_Toplama1_Ariza AND
NOT DI_Kucuk_Toz_Toplama2_Ariza );

IF R_KucukTozToplama1.Q THEN

        DO_Kucuk_TozToplama1_Helezonu_Run:= TRUE;
END_IF

IF F_KucukTozToplama1.Q THEN

        DO_Kucuk_TozToplama1_Helezonu_Run := FALSE;
        Scada_ManBut_KucukTozToplama1_Mot := FALSE;
END_IF


R_BuyukTozToplama1(CLK:= Scada_ManBut_BuyukTozToplama1_Mot AND NOT DI_Buyuk_Toz_Toplama1_Ariza AND
NOT DI_Buyuk_Toz_Toplama2_Ariza AND DO_Buyuk_TozToplama2_Helezonu_Run);

F_BuyukTozToplama1(CLK:= Scada_ManBut_BuyukTozToplama1_Mot AND NOT DI_Buyuk_Toz_Toplama1_Ariza AND
NOT DI_Buyuk_Toz_Toplama2_Ariza);

IF R_BuyukTozToplama1.Q THEN

        DO_Buyuk_TozToplama1_Helezonu_Run:= TRUE;
END_IF

IF F_BuyukTozToplama1.Q THEN

        DO_Buyuk_TozToplama1_Helezonu_Run := FALSE;
        Scada_ManBut_BuyukTozToplama1_Mot := FALSE;
END_IF

R_Bant_Start(CLK:=DI_Bant_Start AND NOT DI_Aktarma_Bandi1_Ariza AND NOT DI_Aktarma_Bandi2_Ariza AND
NOT DI_Aktarma_Bandi3_Ariza );

F_Bant_Stop(CLK:= DI_Bant_Start AND NOT DI_Aktarma_Bandi1_Ariza AND NOT DI_Aktarma_Bandi2_Ariza AND
NOT DI_Aktarma_Bandi3_Ariza );
```

```
IF R_Bant_Start.Q THEN

        DO_Aktarma_Bandi1_Run:= TRUE;
        DO_Aktarma_Bandi2_Run:= TRUE;
        DO_Aktarma_Bandi3_Run:= TRUE;
END_IF

IF F_Bant_Stop.Q THEN

        DO_Aktarma_Bandi1_Run:=FALSE;
        DO_Aktarma_Bandi2_Run:= FALSE;
        DO_Aktarma_Bandi3_Run:= FALSE;
END_IF
```

4.2.6 Building up Auxiliary Windows

These windows are created to help the operator do various adjustments and settings outside the home page. Settings include settings for recipes, inflight settings, time and weighing settings and calibration settings.

4.2.6.1 Settings

Figure 4.44 Settings window

This auxiliary window (Figure 4.44) asks the user in which field he/she wants to do the configurations. The user may switch to various configuration windows using this auxiliary window. These windows appear as follows in turn:

- Create a recipe
- Inflight settings
- Time and weighing settings
- Calibration.

4.2.6.2 Create Recipe

Figure 4.45 Recipe window

This auxiliary window (Figure 4.45) is the one used by the operators to create recipes. The value of materials to be received from silos is written in the relevant part of the windows in kilogram when creating the recipe. Then, the recipe is uploaded to the system by pressing the button of “Reçeteyi Sisteme Yükle”. The recipe uploaded to the system is implemented if “Tartıma Başla” button is pressed.

4.2.6.2.1 Recipe Zero Control Process

This process is related to checking if there is any operation entered through recipes. The control is conducted as shown below:

```
Agrega_SifirKontrolu1(

        Recete_Degeri1:= Scada_Silo1_Recete_Degeri ,
        Recete_Degeri2:= Scada_Silo2_Recete_Degeri ,
        Recete_Degeri3:= Scada_Silo3_Recete_Degeri ,
        Recete_Degeri4:= Scada_Silo4_Recete_Degeri ,
        Recete1_Sifir=> PLC_Silo1_Recete_Sifir ,
        Recete1_SifirDegil=> PLC_Silo1_Recete_SifirDegil ,
        Recete2_Sifir=> PLC_Silo2_Recete_Sifir ,
        Recete2_SifirDegil=> PLC_Silo2_Recete_SifirDegil ,
        Recete3_Sifir=> PLC_Silo3_Recete_Sifir ,
        Recete3_SifirDegil=> PLC_Silo3_Recete_SifirDegil ,
        Recete4_Sifir=> PLC_Silo4_Recete_Sifir ,
        Recete4_SifirDegil=> PLC_Silo4_Recete_SifirDegil );

Kimyasal_SifirKontrolu(

        Recete_Degeri1:= Scada_Kimyasal1_Recete_Degeri ,
        Recete_Degeri2:= Scada_Kimyasal2_Recete_Degeri ,
        Recete1_Sifir=> PLC_Kimyasal1_Recete_Sifir ,
        Recete1_SifirDegil=> PLC_Kimyasal1_Recete_SifirDegil ,
        Recete2_Sifir=> PLC_Kimyasal2_Recete_Sifir ,
        Recete2_SifirDegil=> PLC_Kimyasal2_Recete_SifirDegil ,
```

The following sub software was used for designation of zero and non-zero information in each recipe:

```
IF Recete_Degeri1 = 0 THEN
        Recete1_Sifir := TRUE ;
        Recete1_SifirDegil := FALSE ;
ELSE
        Recete1_Sifir := FALSE ;
        Recete1_SifirDegil := TRUE ;
END_IF

IF Recete_Degeri2 = 0 THEN
        Recete2_Sifir := TRUE ;
        Recete2_SifirDegil := FALSE ;
ELSE
        Recete2_Sifir := FALSE ;
        Recete2_SifirDegil := TRUE ;
END_IF

IF Recete_Degeri3 = 0 THEN
        Recete3_Sifir := TRUE ;
        Recete3_SifirDegil := FALSE ;
```

```
ELSE
		Recete3_Sifir := FALSE ;
		Recete3_SifirDegil := TRUE ;
END_IF

IF Recete_Degeri4 = 0 THEN
		Recete4_Sifir := TRUE ;
		Recete4_SifirDegil := FALSE ;
ELSE
		Recete4_Sifir := FALSE ;
		Recete4_SifirDegil := TRUE ;
END_IF
```

4.2.6.3 Inflight Settings

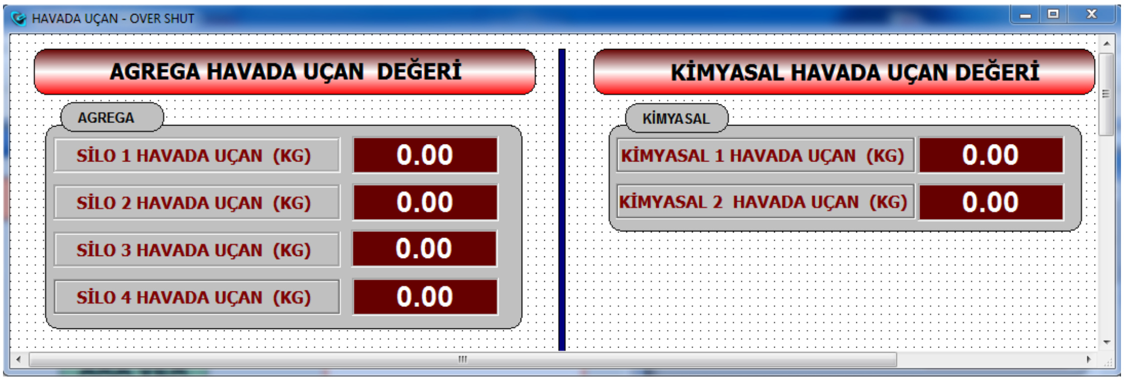


Figure 4.46 Inflight window

This auxiliary window (Figure 4.46) ensures that the inflight value of materials in each silo is determined in kilogram. Since some of the material is turned into dust during pouring from silos, a difference is created between received value and poured value. In order to avoid problems that may be created by this difference in the system, the value which is entered as “inflight” is added to poured value and that is how the received value is calculated. In short, “inflight” value is created to eliminate error margin.

4.2.6.4 Time and Weighing Settings

Figure 4.47 Time and weighing window

4.2.6.4.1 Pouring Values

The pouring values are indicated on the upper part of this window which is a tool developed for security purposes. In some cases, the material is not poured completely from the scales or it takes a lot of time to completely pour the material. This "pouring value" which is calculated in order to prevent delays in the system that may be caused by this little material which cannot be poured is essential and significant for speedy and stable functioning of the system. The pouring values are entered separately for each scale. This is caused by the fact that aggregate silo scales carry materials in quantities of many kilograms but the chemical scales carry materials in quantities of much fewer kilograms. For instance, if pouring value for the aggregate scale is selected as 20 kg, the scale supposes that all the materials in the system are fully poured when the material weight on the scales decreases to 20 kg during pouring and ends the pouring process after waiting for a time which is equal to "Kantar Boşalma Garantileme Süresi". The small amount of material which may remain on the surface of scales can be neglected.

4.2.6.4.2 Delay Time between Scales

Pouring the chemical materials to the mixer starts only after the aggregated material is poured. Otherwise, the inner surface of the mixer is covered with chemical materials since chemical materials are fluid and adhesive. This property of chemical materials prevents the materials from being mixed in a homogenous manner. Therefore, first the aggregated scale starts the pouring and then the chemical scales start pouring. The delay time between scales is used to be determined after how many seconds the chemical scale is going to start pouring following the end of pouring from the aggregated scale.

4.2.6.4.3 Scale Pouring Guarantee Time

This value is used to determine how much time it is going to take for the scale to stop pouring after the scale pouring value is reached.

4.2.6.4.4 Mixer Blending Time

As the name suggests, this time is used to determine how long the mixer is going to mix the materials.

4.2.6.5 Calibration

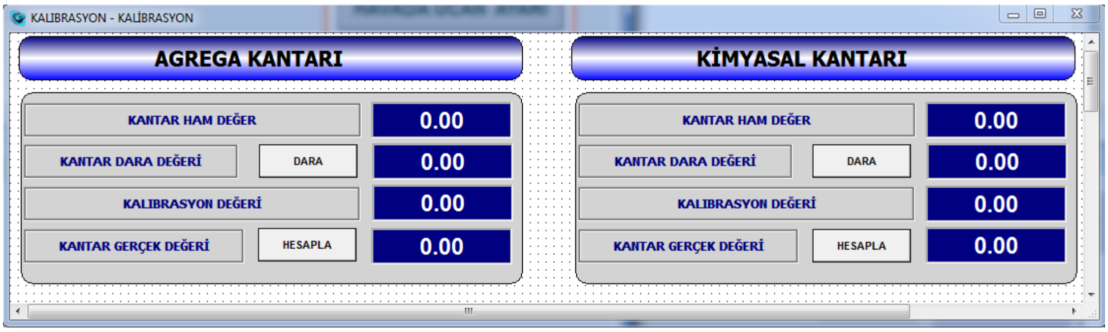


Figure 4.48 Calibration window

Weighing value is obtained through load cells on the scales; however it is needed to carry out another calculation here. The Figure 4.48 shows the calibration window. Analogue values communicated by load cells vary between 0 and 27648. These values may decrease or increase depending on the weight of the materials on the scale. This value is seen as “Kantar Ham Değer” on the window. When there is no material on the scale, in other words, when the scale is empty, the analogue value read on the window is “Kantar Dara Değeri”. It is necessary to find calibration value in order to calculate how much material is available on the scale, in other words, it is necessary to know how many kilograms cause how much increase in the analogue value. To find this out, a material of known weight in kilograms is placed on the scale. Then the value of increase added to the raw value is subtracted from the tare weight and the result is divided by the weight of the material placed on the scale which is in kilogram. The result is the “Kalibrasyon Değeri”. “Kantar Gerçek Değeri” is equal to the result obtained when the calibration value is multiplied by the raw value, in other words, to the actual weight of the material.

The code which is needed by calculating calibration value is as follows:

```
Kimyasal_Ham_Deger := MOVE(AI_Kimyasal_Kantari); (*Analog giristeki deger ham deger degiskenine tasiniyor*)

R_Kim_Dara_Al(CLK:= Scada_ManBut_Kimyasal_DaraAl ); (*Dara al butonu yukselen kenarinda dara aliniyor*)

IF R_Kim_Dara_Al.Q THEN
    Scada_KimyasalKantar_Dara_Degeri := MOVE(INT_TO_REAL (Kimyasal_Ham_Deger) );
END_IF

Scada_Kimyasal_Darasi_Alinan_Real_Deger :=MOVE(INT_TO_REAL(Kimyasal_Ham_Deger) - Scada_KimyasalKantar_Dara_Degeri ); (*Darasi alinan deger elde ediliyor*)

R_Kimyasal_Kalibre_Et(CLK:= Scada_ManBut_Kimyasal_KalibreEt ); (*Kalibre carpani bulunuyor*)

IF R_Kimyasal_Kalibre_Et.Q THEN
    Scada_Kimyasal_Kalibre_Carpani := ( Scada_KimyasalKantar_Kalibre_Degeri ) / ( Scada_Kimyasal_Darasi_Alinan_Real_Deger );
END_IF

FILTER_Kimyasal(IN:= ( Scada_Kimyasal_Kalibre_Carpani * Scada_Kimyasal_Darasi_Alinan_Real_Deger ), NOF:= 10 , POF:= T#25ms , FDV=> Scada_Kimyasal_Kantar_Degeri ); (*Gercek agirlik hesaplaniyor*)


(*                    Agrega Calculation                    *)

Agrega_Ham_Deger := MOVE(AI_Agrega_Kantari); (*Analog giristeki deger ham deger degiskenine tasiniyor*)

R_Agrega_Dara_Al(CLK:= Scada_ManBut_Agrega_DaraAl ); (*Dara al butonu yukselen kenarinda dara aliniyor*)

IF R_Agrega_Dara_Al.Q THEN
    Scada_AgregaKantar_Dara_Degeri := MOVE(INT_TO_REAL ( Agrega_Ham_Deger ) );
END_IF

Scada_AgregaDarasi_Alinan_Real_Deger :=MOVE(INT_TO_REAL(Agrega_Ham_Deger ) - Scada_AgregaKantar_Dara_Degeri ); (*Darasi alinan deger elde ediliyor*)

R_Agrega_Kalibre_Et(CLK:= Scada_ManBut_Agrega_KalibreEt ); (*Kalibre carpani bulunuyor*)

IF R_Agrega_Kalibre_Et.Q THEN
    Scada_Agrega_Kalibre_Carpani := ( Scada_AgregaKantar_Kalibre_Degeri ) / ( Scada_AgregaDarasi_Alinan_Real_Deger );
END_IF

FILTER_Agrega(IN:= ( Scada_Agrega_Kalibre_Carpani * Scada_AgregaDarasi_Alinan_Real_Deger ), NOF:= 10 , POF:= T#25ms , FDV=> Scada_Agrega_Kantar_Degeri ); (*Gercek agirlik hesaplaniyor*)
```

5. CONCLUSION

Construction chemicals production plants are the combination of electromechanical systems which are comprised of giant silos, conveyors, mixers, rotating heads, dust absorption units and packaging units.

This study handles the application of industrial SCADA systems on construction chemicals production plants. It is critically important both for the producers and the customers to ensure that the ratio of compounds within the construction chemical that is desired to be prepared is same for each product. Therefore, SCADA systems are indispensable for plants which produce more than 500 packages and hundreds of kilograms of product on a daily basis since such systems ensure reliability and efficiency, provide opportunities of instant monitoring for the whole plant and instant intervention for any part of the plant, enable running retrospective analyses in a sound manner and make it possible to conduct instantaneous stock control.

Mentioning about the facility, after the automation system provided, electrical consumption becomes 150 kWh and the daily consumption is 1200 kWh if the 8-hours work considered. The production values are 2.5 tons/h and 20 tons/day again considering 8-hours work. This means 800 packages (25 kg per package approximately) construction chemical product is produced daily.

Only 5 workers work in the facility after the system set up, one of them is in the MTU (Master Terminal Unit), one is responsible from quality control of the products and prepares the prescriptions, 2 of them are in the packaging area and one is using forklift to transport the packages to the storage yard.

If there is no automation system in the facility, production will held manually and 4 more workers will needed. These workers will have worked in the manufacturing yard, 2 of them in the aggregate materials silos and the other 2 of them in the chemical materials silos and their duties will have been weighing and mixing the materials.

It is given by the facility that 1 worker's cost as 1500 TL in a month and it becomes 6000 TL for 4 workers. Considering the automation system's total cost as 72 000 TL (60000 TL + VAT), it is easily estimated as investment amortized itself in one year even if only considering workers' cost.

The other consideration is in the rate of production. Although 5 workers work in the facility instead of 9 workers, there is up to %75 increase in the production rate. This huge difference is mainly due to the weighing time. In the automated system weighing time is about 8-9 minutes per prescription, this time becomes 26-27 minutes in manual work. We also know that the rest of the procedure takes 15-16 minutes. So if we compare the production speed of automated system and manual system, it is estimated that the automated system is %75 quicker. Means 800 packages product produced in a month in automated system instead of 450 packages.

The main qualifications of PLC supported SCADA systems are as follows:

- Time and labour saving
- Saving operating costs
- High functionality and high performance
- High efficiency
- Real time imaging and control
- Expandability
- Integration with other database applications

When preparing the content of this study, first general information is provided about components of the system to be applied for beneficiaries to get preliminary information. In this respect, first of all, general information on PLC, its definition covering all the aspects, history, structure, how it is programmed, its advantages and disadvantages is provided. Then equipment to be used in such systems is introduced. PLC programmes and differences among them are also part of the issues handled. The preliminary information provided in this study includes SCADA, its definition and history, its benefits, disadvantages, hardware and software infrastructure as well as communication protocols. In the light of this preliminary information provided, details regarding the construction chemicals industry where SCADA system is established, software programme selected for SCADA software, selected PLC hardware, selected communication protocol as well as SCADA and PLC software developed for the system are explained and examined step by step. [11]

6. REFERENCES

[1] Parr, E.A., *Programmable Controllers,* (3^{rd} Ed.), Elsevier, Great Britain, 2003

[2] Bolton, W., *Programmable logic controllers,* (5th Ed.), Elsevier, UK, 2009

[3] Milli Eğitim Bakanlığı, *PLC'ye Hazırlık* [online], Ankara, http://www.megep.meb.gov.tr/mte_program_modul/moduller_pdf/Plc%20ye%20Haz%C4%B1rl%C4%B1k.pdf [Ziyaret Tarihi: 05 March 2015]

[4] Milli Eğitim Bakanlığı, *Temel PLC sistemleri* [online], Ankara, http://www.megep.meb.gov.tr/mte_program_modul/moduller_pdf/Temel%20Plc%20Sistemleri.pdf [Ziyaret Tarihi: 20 February 2015]

[5] Bailey, D., & Wright, E., *Practical SCADA for industry*, Elsevier, Great Britain, 2003

[6] Milli Eğitim Bakanlığı, *SCADA sistemlerine giriş* [online], Ankara, http://www.megep.meb.gov.tr/mte_program_modul/moduller_pdf/Scada%20Sistemlerine%20Giri%C5%9F.pdf [Ziyaret Tarihi: 22 March 2015]

[7] Elektrik Mühendisleri Odası, *Kontrol Sistemleri – SCADA*, TMMOB Elektrik Mühendisleri Odası Yayınları, Ankara, 2012

[8] Clarke, G., & Reynders, D., *Practical Modern SCADA protocols,* Elsevier, Great Britain, 2004

[9] Kalkınma Bakanlığı, *Yapı Malzemeleri Sektör Raporu* [online], Ankara, http://www.dogumarmarabolgeplani.gov.tr/pdfs/YAPI%20MALZEMELER%C4%B0.pdf [Ziyaret Tarihi: 25 March 2015]

[10] Reliance, *Reliance Design* [online], http://www.reliance-scada.com/files-to-download/documentation/reliance3/Reliance3_Design_ENU.pdf [Ziyaret Tarihi: 11 March 2015]

[11] Özdemir, R., Canbolat, H., *A SCADA System in a Construction Chemicals Manufacturing Plant*, ELECO, 26-28 November 2015 Bursa, under review

7. CURRICULUM VITAE

PERSONAL INFORMATION

Surname : ÖZDEMİR
Name : Rıdvan
Date of Birth : 10/10/1985
Nationality : Turkish
Address : Ayyıldız mah. 1655 sok. Aras Park Sitesi 2/34 Etimesgut Ankara
Phone : +90 532 292 54 32
E-mail : rozdemir@thk.edu.tr; ozdemiridvan@gmail.com;
Marital Status: Married

EDUCATION

02/2013 – 07/2015 Master of Science, Electronics and Communication Engineering
YILDIRIM BEYAZID UNIVERSITY, ANKARA

09/2012 – 06/2014 Master of Science (M.Sc.), Electrical and Electronics Engineering
UNIVERSITY OF TURKISH AERONAUTICAL ASSOCIATION, ANKARA

02/2004 – 06/2008 Bachelor of Science (B.Sc.), Electrical and Electronics Engineering
BILKENT UNIVERSITY, ANKARA

09/2003 – 01/2004 Prep Class (24hour per week English)
BILKENT UNIVERSITY, ANKARA

WORK EXPERIENCE & INTERNSHIP

08/2011 – Present

Occupation or position held: Lecturer – Electrical and Electronics Engineering Department

Name and address of employer: University of Turkish Aeronautical Association (UTAA)

Main activities and responsibilities:

- ✓ Gave a course, physics 101.
- ✓ Gave a laboratory work in physics as laboratory assistant.
- ✓ Gave tutorial support in physics.
- ✓ Managed internships, may fests, student projects, student clubs as dean of students for all university students for two years.
- ✓ Participated R&D projects.
- ✓ Published papers in the field of robotics and automated control

Type of business or sector: Education

06/2012 – Present

Occupation or position held: Founder and Engineer

Name and address of employer: AYBE Automation Ankara Turkey

Main activities and responsibilities:

- ✓ Managing AYBE Limited Co.
- ✓ Managed project, Multi-CNC Project
- ✓ Manufactured more than 25 CNC router machines
- ✓ Manufactured 3,4,5 axis CNC milling machines

Type of business or sector: Automated Control, CNC Machines, Engineering and Science & Technology

08/2010 – 06/2012

Occupation or position held: Co-Founder and Engineer

Name and address of employer: ARI Group Ankara Turkey

Main activities and responsibilities: Engineer; Industrial PLC & SCADA Systems, Generator Synchronization

Type of business or sector: Electrical & Process Engineering

07/2007 – 08/2007 **Occupation or position held: Summer Internship**

Name and address of employer: ÖZTİRYAKİLER A.Ş İstanbul Turkey

Main activities and responsibilities: Internship at R&D Department

Type of business or sector: Industrial Kitchen Equipments

07/2006 – 08/2006 **Occupation or position held: Summer Internship**

Name and address of employer: Electromed A.Ş Ankara Turkey

Main activities and responsibilities: Internship R&D Department

Type of business or sector: Digital Metering Industry

PROJECTS

- ✓ As an engineer member of project committee for "Development of low - altitude UAV and the ground station for regional surveillance" Project financed by Ankara Development Agency, Ankara, 2013
- ✓ As a project manager and an engineer of project management team for "Multi-CNC" Project financed by Ministry of Science, Industry and Technology, Ankara, 2012
- ✓ As an engineer member of project committee for "2 450 kVA generators synchronization project for Osmaniye Duzici Public Hospital", Osmaniye, 2011
- ✓ As an engineer member of project committee for "2 1000 kVA generators synchronization project for Elazig Military Hospital", Elazig, 2011
- ✓ As an engineer member of project committee for "4 1000 kVA generators synchronization project for Konya Beyhekim Public Hospital", Konya, 2011
- ✓ As an engineer member of project committee for "2 750 kVA generators synchronization project for Beypazari Public Hospital", Ankara, 2011
- ✓ As an engineer member of project committee for "SCADA & PLC system integration of Sanliurfa Organized Industrial Zone waste water treatment plant", Sanliurfa, 2011
- ✓ As an engineer member of project committee for "Control panel integration of Miranda AŞ. sewage treatment plant", Ankara, 2011
- ✓ As an engineer member of project committee for "SCADA & PLC system integration of Kazan waste water treatment plant", Ankara, 2010

- ✓ As an engineer member of project committee for “2 1000 kVA generators synchronization project for Duzce University”, Duzce, 2010
- ✓ As an engineer member of project committee for “machine line automation sheetrock manufacturing plant for Tacer AŞ.”, Kırıkkale, 2010
- ✓ As an engineer member of project committee for “SCADA & PLC system integration of Elmadag waste water treatment plant”, Ankara, 2010
- ✓ As an engineer member of project committee for “Patient-Nurse Wireless Communication System” Project financed by BILKENT, Ankara, 2008
- ✓ As an engineer member of project committee for “Automated Packaging Machine” Project financed by BILKENT, Ankara, 2007

CONFERENCES & PUBLICATIONS

- ✓ Ozdemir R., Efe M. O., Kaya M. and Elfarra M., “People or objects tracking for regional surveillance purposes using rotary wing unmanned aerial vehicle”, UHUK, 2012
- ✓ Presentation in National Aerospace Conference about “People or objects tracking for regional surveillance purposes using rotary wing unmanned aerial vehicle” 2013
- ✓ Attended FAI (The Fédération Aéronautique Internationale) Rotorcraft Conference as the first and the only Turkish delegate, Lausanne, March, 2013
- ✓ Attended FAI (The Fédération Aéronautique Internationale) Amateur Built and Experimental Aircraft Conference as the first and the only Turkish delegate, Lausanne, April, 2013
- ✓ Attended FAI (The Fédération Aéronautique Internationale) Rotorcraft Conference as Turkish delegate, Lausanne, March, 2014
- ✓ Attended FAI (The Fédération Aéronautique Internationale) Amateur Built and Experimental Aircraft Conference as Turkish delegate, Lausanne, April, 2014
- ✓ Attended German Open Helicopter Championship as the first Turkish refree, Eisenach, July, 2014
- ✓ Efe M. O., Kaya M. and Elfarra M. and Ozdemir R. “Autonomous Trajectory Tracking and Imaging via a Quadrotor Type UAV: A System Integration Approach”, International mechatronics and machine vision in practice (m2vip), Ankara, 2013
- ✓ Presentation in International Mechatronics and Machine Vision in Practice Conference about “Autonomous Trajectory Tracking and Imaging via a Quadrotor Type UAV: A System Integration Approach”, 2013

KEY ACCOMPLISHMENTS & AWARDS

- **20015-Present;** Carrying out Program Management activities of 5 axis educational CNC milling machine project financed by The Scientific and Technological Research Council of Turkey (TUBITAK)
- **2014;** Grantee for manufacturing educational purpose CNC router from Ankara Development Agency (AKA)
- **2013-2015;** Dean of students in University of Turkish Aeronautical Association (UTAA)
- **2012;** Qualified for techno-entrepreneurship award from Ministry of Science, Industry and Technology
- **2008;** Elected as top young entrepreneur by Small and Medium Enterprises Development Organization (KOSGEB)
- **2003;** Placed 0.5 top percentile in University Entrance Exam out of approximately 1.8 million students
- **2000;** Placed 0.4 top percentile in High School Entrance Exam out of 800.000 students