Amuzu Kossi

# AUTOMATED WASHING AND LEVEL CONTROLLING OF A TANK SYSTEM

Technology and Communication
2012

## Acknowledgement

I am heartily thankful to God almighty for seeing me through my entire progress of this project. My second thanks giving goes to my supervisor, Nieminen Juha, whose encouragement, guidance and support from the initial to the final level enabled me to develop an understanding of the subject.

Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of the project.

Amuzu Kossi

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Degree Program in Mechanical and Production Engineering

# ABSTRACT

| | |
|---|---|
| Author | Amuzu Kossi |
| Title | Automated Washing and Level Controlling of a Tank System |
| Year | 2012 |
| Language | English |
| Pages | 64 + 19 Appendices |
| Name of Supervisor | Nieminen Juha |

The main purpose of this project was to control the level of water in a production tank automatically by using a PID controller together with solenoid valves and water pumps that are controlled by programmable logic controller (PLC).
The control process level is divided into two phases namely the washing and the steady flow production phase.

A programming sequence was designed to control the process with the help of an easy to use graphic user interface known as Supervisory Control and Data Acquit ion (SCADA). The sequences were created with function block with several data blocks with STEP7 software application and linked to the graphic user interface. The sequence gives signals to the actuator, which in turns gives an output respond signal to the component to be controlled.

Proportional Integral Derivative (PID) controller was used to control the speed of the pump and also to measure the level of water in the tank using current and voltage signals. The current and voltage signals are scaled according the programming algorithmic calculations.
Level indications were designed with the programming device to monitor the level of water in tank at each stage of washing and steady flow phases to either high level or low level.

As a result, both tanks were filled to the desired level and controlled with the operator workstation on a centralized computer system. The PLC together with the programming device helps to control and monitor the process at each stage of the production phase.

Keywords: Sequences, PID controller, PLC, SCADA, SIMATIC, Operator workstation

# CONTENTS

## LIST OF FIGURES AND TABLES

# LIST OF APPENDICES

# 1. INTRODUCTION

This chapter describes the background, purpose and the scope of the thesis. It also states the main tasks and the structure of the work.

## 1.1 Background

The development of automation and control processes in industrial applications has increased significantly over the past decades. Involvement of human activities in industrial processes has caused so many problems, like safety, health and industrial damages both to humans and the developmental activities within the level of industrial production and service delivery.

The effect of automation and control systems in recent production and service delivery development has improved the safety and reliability in technology and most human services in the developing countries.

Some of the industries involved in the development of automation and control systems are the oil/gas industries, power generation companies, water and sewage treatment plants, chemical industries, pharmaceutical, food and beverage industries and some basic systems used by service providers in small companies and homes.

Automation processes is now aiming to progress in the so called complete automation which will remove all human machine interface will not be needed but just to enter parameters of the process to be controlled and the machine performs the rest of the designed activities ( Mikkor, 2004).

With this idea for the future development in technology, software applications like the programmable logical control programmable logical control (PLC) can help to achieve this aim for complete automation processes. All PLCs use logic as its

programmable language for sequential operations to complete both basic and complex task.

Using the PLC for controlling and monitoring processes in the industries such as the water treatment industries, has improved due to the highest degree of reliability of international safety standards and monitoring systems.

The main advantage of using PLC for controlling special devices in water treatment plant is the flexibility and change control via program without any other alteration (Jack, 2003).

Pumps, valves and level switches are the main devices found in the water treatment plants for controlling the operations of the treatment plants. Because water treatment plants use them in variety of operations, it is not possible to locate just one type of pump, valve and level switches in the treatment plant (Smith, 2004).

Controlling these devices in the treatment plants is very complex since most systems are far distance apart but can be controlled depending on the level of open-architecture, PLC-based control, monitoring systems, high performance Supervisory Control And Data Acquisition (SCADA) automation process and control systems used for the production (Synchrony,2001). Water treatment plants have two (2) main tanks for a complete production and severely other tanks for storage purposes. The first tank is designed and automated to control the operations of the water treatment, where the main process of treatment takes place before transported to the second tank for further treatment in case the first treatment has some contaminant in the water.

Monitoring and controlling pumps, valves and level switches which are part of the devices used for the process control in Function Block Diagram (FBD) requires some safety standards to ensure the safety of the process-based on the legal requirements including health and also safety policies (Peltonen, 2012).

Using PLC for water tank system can help to control all standards and safety requirements during the treatment process. Moreover, every process requires an operator interface that will allow human intervention in the process. Part of the design specification includes the design of the operator console, which enables the operator to start and stop the process with a pushbutton located on operator console (Siemens, 2010).

PLCs are continually improving in industrial processes especially water treatment plants to meet rising challenges in functionality, communication, size, software, implementation, and diagnostics. Monitoring and controlling devices in water treatment process will increase the four primary control disciplines, that is: drive, integrated motion, integrated process, and sequential control to suit each portion of the treatment processes (Gould, 2006).

## 1.2   Purpose and Aims

The objective for this thesis was to develop a simple process plant that can control and monitor the water level and production in a double tank using the PLC and operator workstation. The main objective of this thesis was to construct a programming sequences and actuators with Function Block Diagrams (FBD) and Data Blocks (DB) that can control the desired tank system.

The thesis will describe the main idea of process automation and controlling systems with PLC in water treatment production relating to the flow control and level applications. Designing sequences and actuators to control pumps, valve and level switches during the treatment process during the programming will be simulated with Function Blocks (FBs) together with the PID controller.

The main objective when designing programming sequence and the operator workstation for the nominal operation of water treatment production was to minimize the induced fatigue which can be caused by human while maximizing the flexibility, reliability, efficiency, standards and safety of the entire production processes.

Further it is very important that critical variables are kept within their limits to avoid misapplication of the devices being controlled in standard conditions. The design program will enable the flow rate and levels of the water to be controlled to suit each portion of the treatment relating to the pumps, valves and level switches.

## 1.3  Description of the thesis

For the project to be carried on, some preparation work, especially learning and practice of computer skills like the use of SIMATIC Manager STEP 7 and the In-Touch software application was done. Besides the study of computer skills, some other related topics were also considered to better create understanding of automation process and control systems.

This thesis work includes knowledge in the following fields:

- Automation and control process
- Proportional Integral–Derivative controller (PID controller)
- Process planning and operation planning in water treatment plants
- Design automation with Function block diagram(FBD)
- Functions of mechanical devices used in water treatment plants

The thesis work is composed of four main individual tasks: Literature review, Design and Schematic Diagram of the System, Programmable logic control and Remote Controlling of the Process.

## 1.4 Project scope

The following shows the scope of the thesis work:

- PLC (Programmable Logic Controller) as the main controlling device
- Designing of a program structure to control the process
- Controlling Level of water t water with PID controller
- Controlling water level between Tank A and Tank B so that they will be at the same level during production
- Designing an operator workstation (SCADA) control the process

## Project structure

For the thesis to flow smooth and accordingly, some approaches were taking into consideration. There were two steps that was taken into consideration to meet the project due date. These steps were:

1. Studies on hardware that were needed for the project to be completed, such as PLC, pump controllers and level switches
2. Studies on the available software that can be programmed on the PLC to the pump, valve and level switches

The Figure 1 below shows the structure or the work flow of the thesis that was used as a guideline to achieve the objectives of the work. The thesis started by doing a research on the topic and after doing literature review, equipment and proceedings were identified. Some of the equipment was: the level switches, pumps, transmitters, PLC and valves. At the same time, programming sequence was designed to control the tank system.
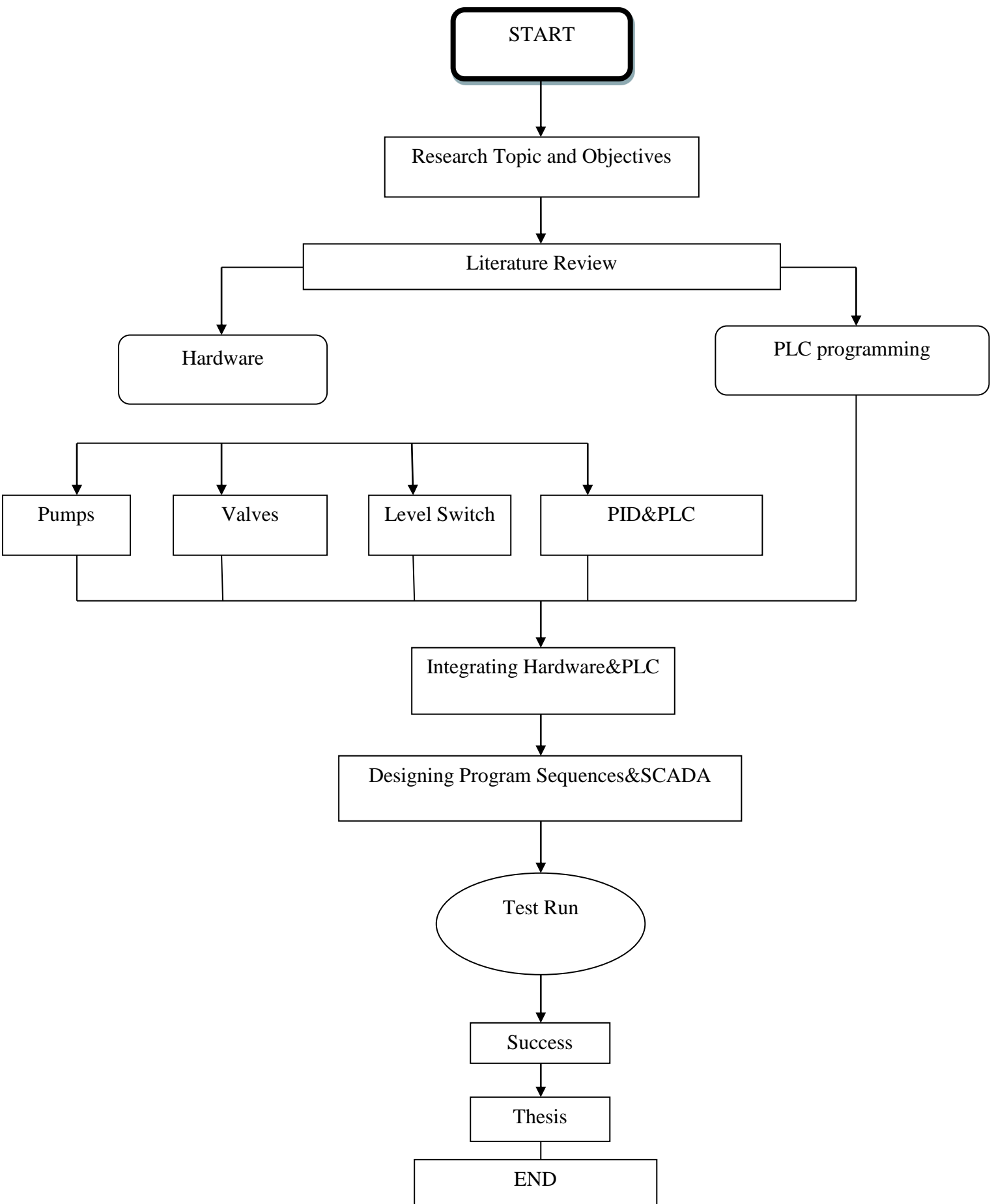
**Figure 1**. Structure of the project

## 2. LITERATURE REVIEW

This chapter provides a literature overview on automation and control process, PLC programming Process planning, and Design automation.

### 2.1 Automation and control process

Automatic control systems enable us to operate processes in excellent and accurate manner. Considering some process applications in the industries, as mentioned in the introduction need control systems to achieve industrial targets and objectives by continually measuring process variables such as temperature, pressure, level, flow and concentration, taking into actions such as opening valves, slowing down pumps and turning up heaters in order to maintain measured process variables at the operators set point values (Douglas, 2006).

**Safety**

The first motivation for automation and control systems is safety, which includes the safety of people, environment and the equipment used for the application processes.

The safety of people in the community and the personal in the production industries are the highest priority in any plant operation. Due to these reasons among others, the design of a process and associated control systems must always make the human safety the primary objective (Douglas, 2006).

**Reliability**

An automation and control system is based on the foundations of feedback theory and linear system analysis, and it integrates the concepts of network theory and communication theory. Therefore systems that are automated are safe and reliable to operate.

Reliability is achieved in automation and control systems as interconnection of components forming a system configuration that will provide a desired system

respond. The concepts of analysis of the system for reliability are the foundation provided by the linear system theory, which assumes a cause effect relationship for the components of a system.

The application of these theories has helped to improve automation and control systems. Component or process to be controlled is represented by block diagram, as shown in Figure 2. The input signal represents the cause relationship while the output signal represents the effect relationship of the process, which in turns represents the processing of the input signal to provide a desired output signal variable, often with power implication.
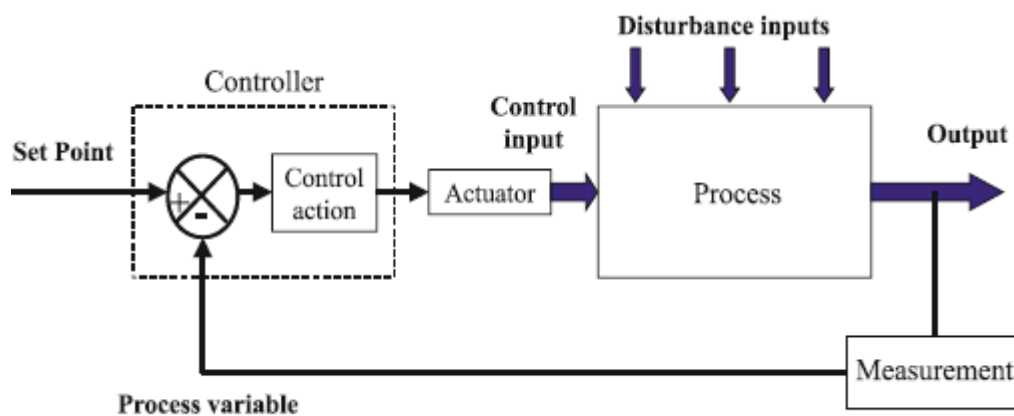


**Figure 2**. Process to be controlled (Douglas, 2006).

**Profit motive**

When the safety issues and reliability relating to production and safety are properly achieved, than the main control objectives can be focused on profit motive. Automatic control systems offer strong benefits in this regard.

Plant level control objectives motivated by profit include:

- Meeting final product specifications
- Minimizing waste production
- Minimizing environmental impact
- Minimizing overall production rate

Industries will achieve a high level of profit if they operate as close as possible to these minimum or maximum objectives (Douglas, 2006).

**Variability control reduction**

In general, a control system is a collection of electronic devices and equipment which are in place to ensure the stability, accuracy and smooth transition of a process or a manufacturing activity. The signals used to control this processes can exhibit large variability in a measured process variable when the process is controlled poorly.

Using a standard automation and control systems devices and application software can help to eliminate this poorly controlled variability of systems. The relationship between set point and process values are controlled very well not to cause violation.

To ensure that operating constraint limits are not exceeded, an operator specifies a set point (SP), that is, the point that the control system will maintain the process value (PV), which must be set far from the constraint to ensure it is never violated.

As shown in the Figure 2 below, a poorly controlled process can exhibit large variability in a measured process variable (e.g., temperature, pressure, level, flow, concentration) over time. The plot shown below is an example of a measured process variable (PV) which must not exceed a maximum value. And as often is the case, the closer we can run the process constraint, the greater our profit.
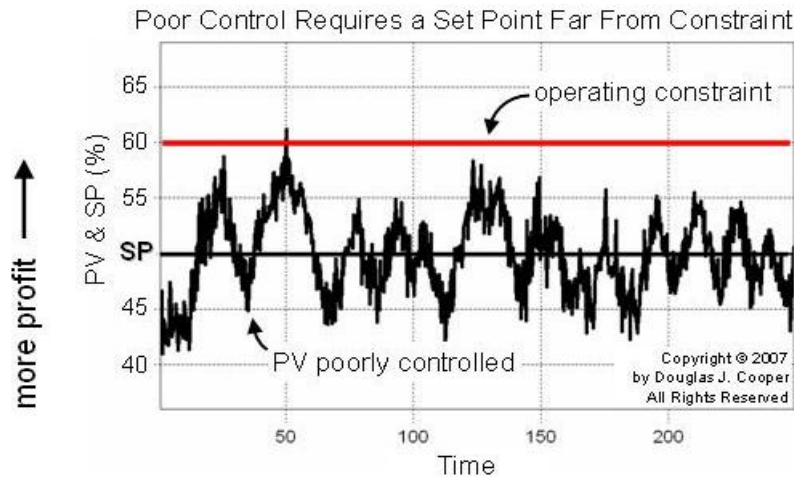
**Figure 3**. Control constraints with poor PV control (Douglas, 2006).

This process can be improved when the variability in the measured PV is significantly less as a result, the SP can be moved closer to the operating constraint.

With the plot shown in Figure 4 below, the SP is moved to 55%, the average PV is maintained closer to the specification limit while still remaining below the maximum allowed value. The result in this case will increase profitability of the operation (Douglas, 2006).

**Easier to troubleshoot**

PLCs have resident diagnostics and override functions that allow users to easily trace and correct software and hardware problems (Frank, 2005).

Many PLC types are available such as the Mitsubishi, Siemens, Nais, Omron and many more. In this thesis, I am going to use Siemens PLC which has all the inputs and outputs controllers. It is suitable to control the switching of valve and speed of motor pumps.
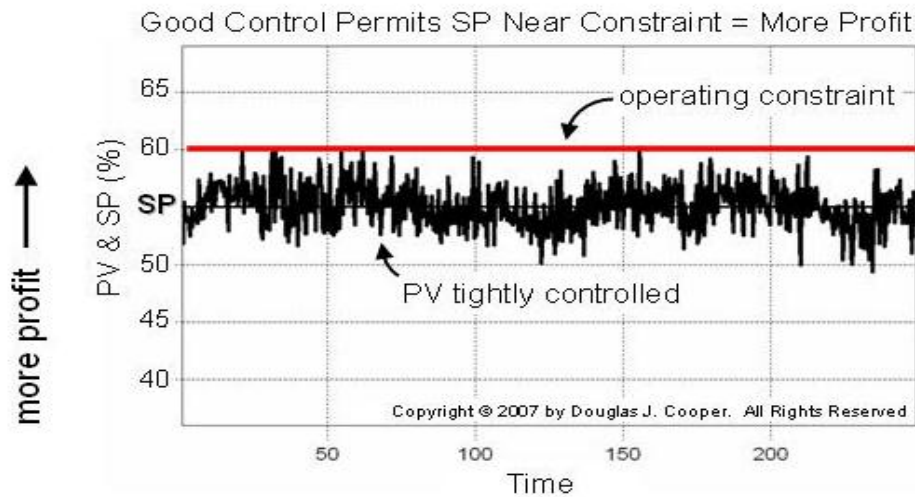
**Figure 4**.Good control permits SP near constraints.

## 2.2 Basic Control Concepts

The basic process control systems consist of a control loop as shown in Figure 2, above. The system is made of four main components which are listed below:

- Measurement of the state of the condition of a process
- The controller calculating an action based on the measured value against a pre-set or desired value (set point)
- An output signal resulting from the controller calculation which is used to manipulate the process action through some form of actuator
- The process that will be reacting to this signal from the controller calculation (manipulated value) and changing its state or condition.

Within the elements of a process control loop, there are two important signals used to control the process. They are called:

- Process Variable or PV
- Manipulated Variable or MV

In most industrial applications of process control systems, the PV is always measured with an instrument in the field and acts as an input to an automatic controller which takes action based on the value of it. Alternatively, the PV can be

an input to a data displaying so that the operator can use the reading to adjust the process through manual control and supervision (IDC Technologies, 2007).

The variable that is used to control or manipulate, in order to have control over the PV, is called the manipulated variable (MV). For instance, manipulating a valve to control flow can be used as an example to illustrate the difference between the manipulated and the process value. Here, the valve position is called the manipulated variable and the measured flow becomes the process variable (PV).

## 2.3  Principles of Control Systems

For a process to work effectively, the control input used should affect the output of the process. If the input condition changes, the following signals will have to respond:

- The output will rise or fall
- How response will be generated
- How long the output signal will change
- What will be the response curve or trajectory of the response

The answers to these questions can be generated by creating a mathematical model of the relationship between the chosen input and the output of the process in question. In process control design, a useful technique of block diagram modeling is used to assist in the representation of the process and its control system. The following section introduces the principles that should apply to most practical control loop situations (IDC Technologies, 2007).The process plant is represented by an input/output block as shown in the Figure 5below.

*Disturbances input*
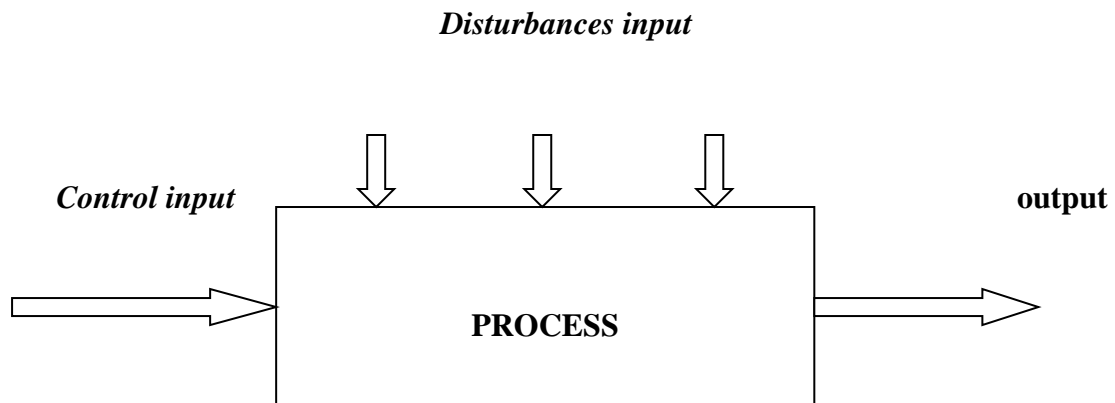


**Figure 5.**Simple process control system

Control inputs are also known as manipulated variables. The output is the process variable to be controlled.

The Figure 5 shows a controller signal that will operate on an input to the process, known as the manipulated variable. The process is drive with the input signal to control the output of the process to a particular value or set point. The output may be affected by other conditions in the process or by external actions such as changes in supply pressures or in the quality of materials being used in the process. These are all regarded as disturbance inputs and the control action will need to overcome their influences as well as possible.

The challenge in designing a process control is how to maintain controlled process variable at the target value or change it to meet production needs whilst compensating for the disturbances that may arise from inputs (IDC Technologies, 2007).

The Figure 6 shown below is an example of keeping the level of water in a tank in a constant height while others are drawing off from it. For this to be achieved, the input flow will be manipulated to keep the level steady. The value of a process model is that it provides a means of showing the way the output will respond to input actions. This is done by having a mathematical model based on the physical and chemical laws affecting the process (IDC Technologies, 2007).

The example below shows an open tank with a cross sectional area A is supplied with an inflow of water $Q_1$ that can be controlled or manipulated. The outflow from the tank passes through a valve with resistance R to the output flow $Q_2$. The level of water of the water or pressure head in the tank is denoted as H. The output pressure $Q_2$ will increase with represent to an increase in of the water level H in the tank and the level of the water will be steady when the output $Q_2$ and input $Q_1$ are equal (IDC Technologies, 2007).

The block diagram of the process in shown in Figure 6,
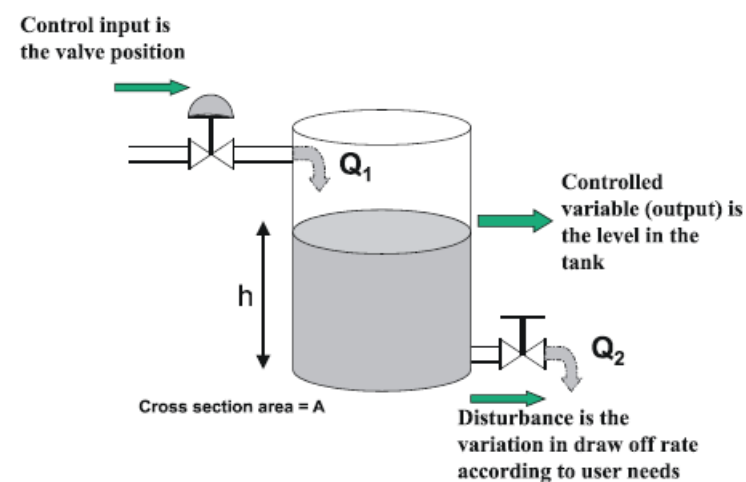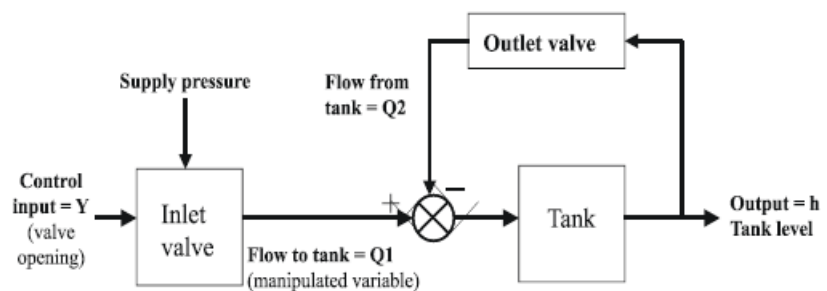


Figure 6.Elementary block diagram of tank process (IDC Technologies, 2007).

## 2.4  Control modes

For the process to control as stable, the five control modes should be taken into consideration. The five basic forms of modes used for process control are:

- On- Off
- Modulating
- Open Loop
- Feed forward
- Closed loop

**On-Off control:**

The oldest way for controlling process was to use switches which gives simple On and Off conditions. This is a discontinuous form of control action, and is also referred to as two-position control. For this on-off controller to be perfect the process is On when the measurement is below the set point (SP) and the manipulated variable (MV) is at its maximum value. Above the SP, the controller is at Off position and the manipulated value (MV) is at a minimum (IDC Technologies, 2007).

**Modulating control:**

If an output signal of a controller can move through a range of values, than it is modulation control. Modulate control takes place within a defined operating range only. The set limits of modulation must have upper and lower limits. Modulation control is a smoother form of control than step control; it can be used in both open and closed loop control systems (IDC Technologies, 2007).

**Open loop:**

Open loo type of control system does not self-correct, when the Process Variable (PV) drifts. This is because the control action (controller output signal OP) is not a function of the Process Variable (PV) or load changes. The system is always open and gives responses to the process depending on the control input variable.

**Feed forward control:**

A feed forward control is used for anticipating correction of manipulated variables that requires the delivering of the output variable. It is seen as a form of open loop control as the PV is not used directly in the control action (IDC Technologies, 2007).

**Closed loop or feedback control:**

The idea of closed loop control is to measure the PV to compare this with the SP, which is the desired, or target value and to determine a control action which results in a change of the Output (OP) value of an automatic controller.

In most cases, the ERROR (ERR) term is used to calculate the OP value.

ERR=PV-SP

If ERR=SP-PV has to be used, the controller has to be set for REVERSE control action (IDC Technologies, 2007).

## 2.5 Proportional Integral Derivative (PID)

The Proportional Integral Derivative (PID) controller is the most common controller in control systems. PID controllers were mostly used in the mid 1990`s for controlling loops in process controls.

The best features of a controller used for process control can only be achieved if the controller is well tuned. The discrete time PID controller tuning methods are presented to optimize the closed loop performance and improve robustness in the varying time delay systems (Lasse, 2008).

A controller can be defined as a device found in a closed-loop control that compares the measured value (actual value) with the desired value, and then calculates and outputs the manipulated variable. There are different controlled

systems used for different application in process control systems due to the varying responses. There are controlled systems which respond quickly, systems that respond very slowly and systems with storage property. Each of the controlled systems is selected depending on the application or process to be controlled (Bischoff, Hofmann, Terzi, 1997).

Each of these controlled systems mentioned above, changes the manipulated variable in different ways. For this reason there are various types of controllers each with its own control response (Bischoff et al, 1997).

Process control can be defined as a measuring process of variables, comparison of those variables with respect to its set point, and manipulating of the process in a way that will hold the variable at its set point when the set point changes or when a disturbance changes the process .

 Most processes contain many variables that need to be held at a set point and many variables that can be manipulated. Usually, each controlled variable may be affected by more than one manipulated variable and each manipulated variable may affect more than one controlled variable (John, 2006). However, most of the process control systems has its controlled variable and manipulated variables paired together so that one manipulated variable is used to control one variable with the application or process.

The controlled and the manipulated variables, each paired together with the control algorithm are referred to as a control loop. To achieve a better and effective control, there are a number of mathematical algorithms that compute together or individually to change an output variable based on the controlled variable (John, 2006).

## 2.6  The Control Loop

The process control loop contains the following elements:

- **The measurement of a process variable**: some of the devices like the sensor, more commonly known as the transmitter, measures some variable in the process such as the temperature, level of fluids pressure, or flow rate, and converts that measurement to a signal (typically 4 to 20 mA) for transmission to the controller or the control system (John, 2006).

- **The control algorithm**: The control system has a mathematical algorithm inside the system that executes the process at some time period (typically every second or faster) to calculate the output signal to be transmitted to the final control element (John, 2006).

- **A final control element**: Signals for the controller received from the some devices like the valve, air flow damper, motor speed controller, or other devices are used to manipulate the process, typically by changing the flow of some materials (John, 2006).

- **The process**: The process responds to the change in the manipulated variable with a resulting change in the measured variable. The dynamics of the process response are major factors in choosing the parameters used in the control algorithm (John, 2006).

### Standard PID Control

The standard PID control basically consists of two function blocks (FBs) which are made up of the algorithms for generating control and signal processing for continuous or step controllers. It is a control in which a standard function block incorporates the functionality of the controller (Siemens, 2003).

The controller and its properties of the functions in the measuring and adjusting channel are realized or simulated by means of the numeric algorithms of the function block. The data required for these cyclic calculations are saved in

control-loop specific data blocks. An FB is only required once to create several controllers (Siemens, 2003).

**Basic functions of standard PID control**

In many controlling tasks, there are many processes- influencing element of importance, but high requirements are also placed on the signal processing function. In addition to the actual controller with the PID algorithm functions for conditioning the set point and process variables as well as for revision the calculated manipulated variable are also integrated. The Figure 7 below shows the display and monitoring functions for a continuous controlling process (Siemens, 2003).
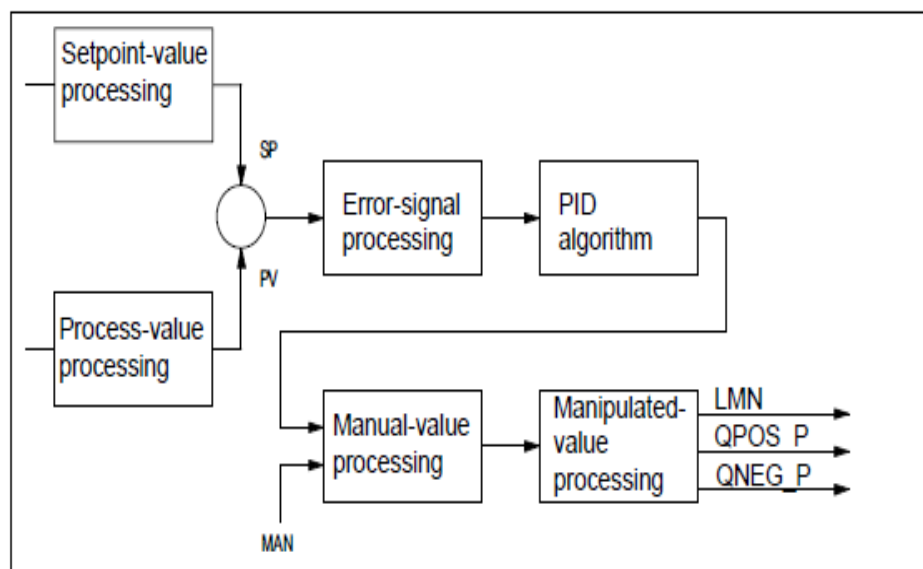


**Figure 7.** Function overview of the standard PID Controller (Siemens, 2003).

## 2.7 Role of the Control Algorithm

The basic purpose of a control system is to bring the process measurement to the set point whenever the set point is changed, and to hold the process measurement at the set point by manipulating the final control element. The process algorithm must be designed to quickly respond to changes in the set point (usually caused by

operator action) and to changes in the loads (disturbances). The design of the control algorithm must also prevent the loop from becoming unstable, that is, from oscillating (John, 2006).

**Auto/manual**

Most control systems allow the operator to make choices by placing an individual loops into either manual or automatic mode. Operators can adjust the output process in manual mode to bring the measured variable to the desired value. In automatic mode, the control loop manipulates the output to hold the process measurements at their set points (John, 2006).

**Action**

The most important parameter to configure within the PID algorithm is to depend which action is been processed. The action determines the relationship between the directions of change in the input and resulting change in the output. If the controller is direct acting, an increase in its input will result in an increase in its output. With reverse action an increase in its input will result in a decrease in its output (John, 2006).

## 2.8  PID Responses

All PID algorithm controllers are made up of three basic response, this are the proportional (or gain), integral (or reset), and derivative. Each of these response works on the algorithm depending on the action of the process.

**Proportional**

The most basic response is the proportional, or gain, response. This is the principle means of the controlling the process. The automatic controller needs to correct the controllers output, with an action proportional to the error. The correction starts from the output value at the beginning of the automatic control action.

In its pure form, the output of the controller is the error times the gain added to a constant known as the manual reset (John, 2006).

Output = E x G + k

Where:

- Output = the signal to the process
- E = error (difference between the measurement and the set point.
- G = Gain
- k = manual reset, the value of the output when the measurement equals the
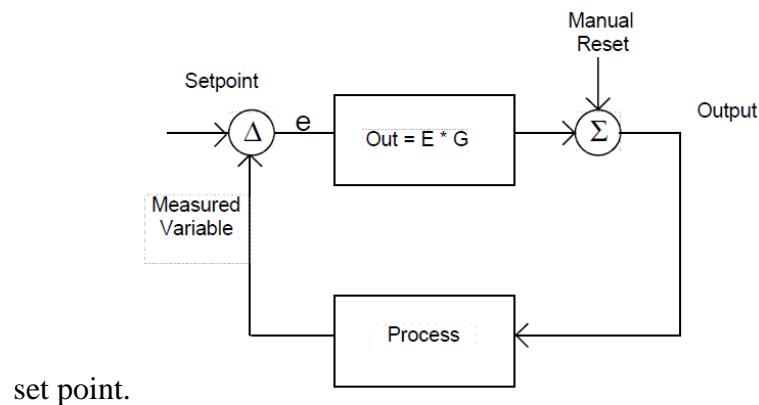


set point.

**Figure 8**. A control loop using a proportional only algorithm (John, 2006).

The output of the process will change when the process measurement, the set point, or the manual reset causes a change. If the process measurement, set, and manual reset are held constant, the output will be constant.

An example of proportional control can be a level with an adjustable pivot as shown in the figure below. The two variables that is the process measurement push on one end of the lever with the valve connected to the other end. The position of the pivot determines the gain and moving the pivot to the left will increase the gain because of the movement of the valve for a given change in the process measurement (John, 2006).
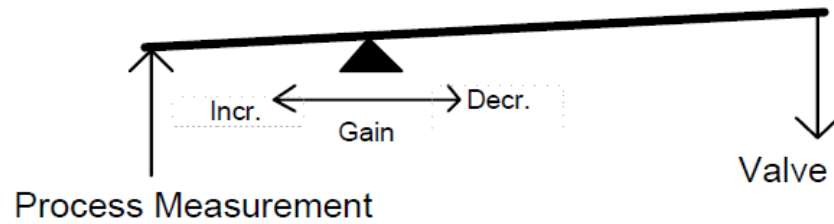
**Figure 9**. A lever used as a proportional only reverse acting controller.

**Proportional offset**

Proportional can only control an offset. Only the adjustment of the manual reset removes the offset. Figure 8 below shows example of a tank with liquid flowing in and flowing out under the control of the level controller. The flow can be considered independent and can be considered a load to the level control (John, 2006).

A pump is used to control the flow and it is proportional to the output of the controller.
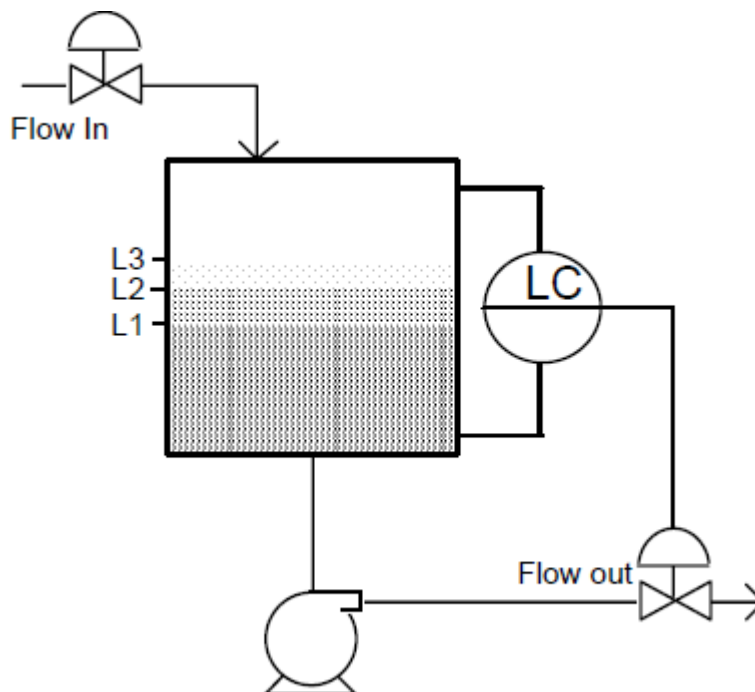


**Figure 10.** Proportional level controls (John, 2006).

Assuming that the level in the tank is at its set point of 50%, the output is 50% and both the flow in and the flow out are 500 gpm. Then we assume that the flow in is increased to 600gpm. It will be recognized that the level in the tank will rise more than the liquid going out. As the level increases, the valve will open and more flow will leave. If the gain is 2, each one percent increase in the liquid level will open the valve 2% and will increase the flow out by 20gpm. Therefore by the time the level reaches 55% (5% error) the output will be set at 60% and flow out will be 600gpm, the same as the flow in. The level will then be constant. The 5% error is known as the offset (John, 2006). However gain cannot be made infinite. In most loops there is a limit to the amount of gain that can be used. If this limit is exceeded the loop will oscillate.

**Integral mode (Reset)**

Integral action is normally used to assist proportional control in order to control towards no OFFSET in the output signal. This means that it controls towards no error (ERR=0). Both the proportional and the integral can be combined to control the process. This combination can be called as PI-control (IDC Technologies, 2007).

Formula for I-Control:

$$OP = \left(\frac{K}{T_{int}}\right) \int_{O}^{T} ERR \, dt$$

Formula for PI-Control:

$$OP = \left(\frac{K}{T_{int}}\right) \int_{O}^{T} ERR \, dt + (K * ERR + MAUAL)$$

$T_{int}$ is the Integral Time Constant.

**Derivative control (D)**

Derivative is the third and final element of PID control. It gives the responses to the rate of change of the process (or error). Derivatives are normally applied to the process only (John, 2006).

The only purpose of the derivative control is to add stability to a closed loop control system. The magnitude of the derivative control (D-Control) is proportional to the rate of change (or speed) of the process variable (PV). Using the D-control can be used to enhance the stability of a control loop at the expense of amplifying the rate of change of noise The D-Control on its own has no purpose; it is always used in combination with the P-Control or PI-Control. This results in a PD-Control or PID-Control. PID-Control is mostly used if D-Control is required (IDC Technologies, 2007).

Formula for D-Control:

$$OP = K \; {}^{*}Tder \; (dERR \; /dt \,)$$

Where

$T_{der}$ is the Derivative Time Constant.

## 2.9 Programmable Logic Controllers (PLC)

Programmable logic controller (PLC) is a specialized computer used to control machines and processes in most industrial operations. It uses a programmable memory to store instructions and execute specific functions that include on/off control, timing, counting, sequencing, arithmetic, and data handling.

PLCs were used to replace relay logic, which was used for most industrial applications in the past decades due to its ever increasing range of functions that are in found in many and more complex applications.

Most PLCs are based on the same principles of structure as those employed in computer architecture, it is capable not only of performing relay switching tasks

but also for performing other applications such as counting, calculating, comparing, and the processing of analog signals (Mohammad, 2008).

PLC`s are designed for multiple inputs and output arrangements, extended temperature ranges, immunity to electrical noise, and resistance to vibration and impacts. This reason makes the difference between general purpose computers to a PLC (Gary, 2006).

**Digital and analog signals**

Digital or discrete signals behave as binary switches, yielding simply an ON or OFF signal (1 or 0, True or False, respectively). Photoelectric sensors, pushbuttons and limit switches are examples of devices providing a discrete signal. Voltages and currents are used to set these discrete signals within a specific range where it will be designated as ON and another as OFF.

For example, a PLC might use 24V DC I/O, with values above 22V DC representing ON, values below 2VDC representing OFF, and intermediate values undefined. Initially, PLCs had on discrete I/O (Mohammad, 2008).

Analog signals are like volume controls that range from a value between zero and full scale. These values are typically interpreted as integer values (counts) by the PLC, with various ranges of accuracy depending on the device and the number of bits available to store the data. As PLCs are typically made up of 16bits signed binary processors, the integer values are limited between -32,768 and +32,767 (Mohammad, 2008).

Pressures, temperatures, flow and weight are often represented by analog signals. Analog signals can use voltage and current with a specific magnitude proportional to the value of the process signal. For example, an analog 4-40mA or 0-10V input would be converted into an integer value of 0-32767.

An analog signal output could send a 4 to 20 milliamp signal to a variable –speed drive to control the speed of a motor in proportion to analog signal received from the output module (Mohammad, 2008).

**Table 1**, below shows the valve position correlation to the module`s output voltage.

Table 1. The valve position correlation (Mohammad, 2008)

| ANALOG SIGNAL COMPARISON FOR SAMPLE ANALOG VALUE OUTPUT | | |
|---|---|---|
| Valve Position | Voltage Output Signal | Decimal Valve Output to Output Section |
| FULL OPEN | 10 | 32,767 |
| 50% | 5 | 16,384 |
| 40% | 4 | 13,107 |
| 30% | 3 | 9,830 |
| 20% | 2 | 6,553 |
| 10% | 1 | 3,276 |
| CLOSED | 0 | 0 |

**Basic components of PLC**

The PLC system is basically made up of five main components. These are the central processing unit (CPU), memory, the power supply unit, input and output module and programming device.

**Central processing unit (CPU)** is the unit that comprises of the microprocessor and interprets the input signals and carries out the control actions, according to the program stored in its memory, communicating the decisions as action signals to the outputs (Norhaslinda, 2008).

**Power supply unit** is needed to convert the main AC voltage to a low DC voltage (24V) necessary for the processor and the circuits in the input and output interface modules. Power supply modules may be connected to the bus or may have to be wired to the CPU module in modular PLC systems.

**Programming device** is used to enter the required program into the memory of the processor. The function block diagram or the ladder diagram of the program needs to be translated into mnemonic codes before being keyed in into the programming device (Norhaslinda, 2008).

**Memory unit** is mostly used to store the program functions that are to be used to control the actions by the microprocessor.

**Input and output modules** are modules where the processor receives information from the external devices and communicates information. Most of the input signals are from high level sensors, low level sensors, switches (start and stop button) and temperature controllers that communicates to the process before generating a signal to the output device. Some of the output devices that the controlled by these signals are pumps, heater, stirrer and solenoid valve (Norhaslinda, 2008).

**The rack or bus:** the CPU reads and writes inputs and outputs modules that are part of the modular PLC during every scan cycle. The CPU module is connected to each of the I/O modules via a set of parallel conductors called the bus. Some modules systems have the bus in a backplane circuit card in a rack and all the PLC modules are connected or plugged into slots of racks. In other modular systems, the I/O modules are plugged into the side of the CPU module or into the side of an I/O module that is already plugged into the CPU, so the bus conductors are connected through the I/O modules.

All bus conductors used in PLC systems are used for data that the CPU can send to or receive from the I/O modules, several bits at a time. The CPU must specify which of the I/O modules the CPU wants to read from or write to. The I/O modules addresses are assigned automatically according to how far the module is located away from the CPU module along the bus.

Some bus conductors are used for miscellaneous control signals passed between the CPU module and I/O modules and to provide power to run the circuitry in the modules. The bus does not provide power to operate the sensors or actuators attached to the I/O modules.

## 2.10 Components used for the tank system

The following components were used to control the process of the tank system. They are used together with the PLC and the programming sequence to control the parameters and the instrumentations used for the system design.

### Switch Valve

A solenoid valve is a device that regulates the flow of substances (gases, fluidized solids, slurries, or liquids) by opening, closing or partially obstructing various passage ways. When a process needs to be controlled automatically, solenoid valves are used. They are being used to an increase degree in the most varied types of plants and equipment. Selecting valves mostly depends on the application to be controlled in question. Valves are used in a variety of application including industrial, military, commercial, residential, and transportation (Mohammad, 2008).

Solenoid valves are controlled by electricity which energizes or de-energizes causing the valve to be in an opened or closed position. The actuator takes the form of an electromagnet. When energized, a magnetic field builds up which pulls a plunger or pivoted armature against the action of a spring. The plunger or pivot armature of the valve returns to its original position when de-energized.

**Level measurement (Sensors)**

Measurement can be defined as the estimation of the magnitude of some attribute of an object, such as length or weight, relative to a unit of measurement. Measurement usually involves using a measuring instrument, such as a ruler or scale, which is calibrated to compare the object to some standard, such as a meter or kilogram (Muhd, 2008).

Selection of measuring device depends on the type of process to be controlled and the materials going to be used for the process. There are many physical conditions and application variables that affect the selection of the optimal level monitoring solution for industrial and or commercial processes. Some of the physical states are temperature, pressure or vacuum, density, noise, vibration, mechanical shocks etc.

The level sensor processes the signal from it and, then processes the signal through the transmitter which converts the signal to readable format for the PID controller that will control communications and monitoring.

**Actuators**

An actuator is a device used in most industrial processes for controlling or converting electrical signal from PID controllers, such as the PLC, into a physical condition. There are always connected to the PLC output such as signals controlling motors. Starter used for starting motors can be one example of an actuator that is often connected to a PLC output. Depending on the status of the PLC output, the motor starter either provides power to the motor or prevents power form flowing to the motor.

**2.11 Descriptions of operator work stations (SCADA)**

The operator work station is an integral part in process automation and control systems. Operator work stations are most often computer terminals that are made up of networked SCADA central host computers. The central host computers are

used as a server for the SCADA application, and the entire operator terminal are connected to transfer and receive information to the central host computer depending on the command and the actions of the operator.

The Supervisory Control and Data Acquisition (SCADA) is a computer software application with the software components as an operator interface or Man Machine Interface/Human Machine Interface (MMI/HMI) package. The software is selected based on the nature of the SCADA application control systems. The purpose of SCADA is to monitor, control and alarm plants, typically water and wastewater treatment facilities or regional operating systems from a central location which includes the intake and effluent structure, pumping stations, chlorination stations, control valve stations and the others.

## 2.12 Components of SCADA

The SCADA system is made up of three main elements called the RTUs (Remote Telemetry Units), communications and an HMI (Human Machine Interface). The RTU most effectively collects information at site, whiles communications brings those collected information from the various sites or regional RTU sites to the central location, and occasionally returns the instructions to the RTU.

The HMI displays the information in a graphic form that makes it easily to understand, archive the data received, transits alarms and permits the operator control as required. The HMI is essentially a PC system running powerful graphics and alarm software programs.

The major advantage of using SCADA is the ability to significantly reduce operating labor cost, while at the same time improving the plant or regional system performance and reliability. The SCADA application helps in information gathering within a plant by preventing personnel to spend time wandering all over the site, and correspondingly the frequency of field site inspections required in a regional system can be minimized.

The SCADA system alarm call out can be avoided, since it indicates the nature and degree of a problem, while the ability to remotely control site equipment may permit an operator at home to postpone a site visit till working hours.

The based alarming used in SCADA is very reliable since it is in-house and tied directly to the process control.

**Phases used to create functional SCADA system**

There are five main basic phases used in creating the functional SCADA system for a process control. The activities and the parameters used must be considered before the selection of the software to be used to the program.

**Phase 1: Design of the system architecture**

The design of the system architecture includes all the important communication systems and the parameters to be controlled. All the site instrumentation that are used on site and needed to be monitored and control desired parameters also are involved during the first phase.

**Phase 2: Supply of the RTU**

This phase involves the supply of the RTU, communication and HMI equipment, which consist of the PC system and the necessary powerful graphic and alarm software programs.

**Phase 3: Programming**

This phase involves the whole programming of the communication equipment and the powerful HMI graphic and alarm software programs.

**Phase 4: Installation**

The installation of the communication equipment and the PC systems are completed at this phase. Linking the equipment and the PC system is the former task, typically much more involved at this installation phase.

**Phase5: Commissioning**

The commissioning of the system, during which communication and the HMI programming problems are identified and rectified, while the system is proven to the client, operator training and the system documentation is provided at the final phase of the SCADA design phase.

.

# 3. SYSTEM DESIGN

This chapter will outline the task involved in planning the automation project for the programmable controller (PLC) with the software and integrating hard ware devices.

## 3.1 Introduction

To achieve the automation and process control of the water level during the production, programming software called SIMATIC STEP 7 was used together with the devices going to be used for the application. The entire production consists of automation process of number of individual tasks. Identifying groups of related individual tasks within the process and then breaking these groups into smaller tasks is even the most complex process to be defined.

As each of the group is divided into smaller tasks, the tasks required for controlling each process becomes less complicated. Describing individual functional areas of the process was not the only task but also the various elements that control the area of process.

The STEP 7 programming software allows the process to be broken down into individual, self-contained program section that can be simplified, easier to modify, and debugging is simplified since testing can be done on separate sections of the program.

Using this software will help to create a user interface, which can be controlled easily on a control panel connecting the whole system. The programming languages Ladder Logic, Statement List, and Function Block Diagram are an integral part of the standard package. The language used for this programming is the FBD which is a graphic representation of STEP 7 programming language and uses logic boxes familiar from Boolean algebra to represent the logic.

The process was divided into two basic phases; the first phase involves washing the tanks before the water production is started. After the tanks are cleaned, the

entire production starts and the second phase will be controlling the water level at constant position in respective of the disturbances.

## 3.2 Design and Schematic Diagram of the System

The Figure 11 below shows the first phase of the design programming, water is pumped from a source to tank A to a desired level and drained to allow washing of the tanks for the production. After the first phase has been completed, the other pump2 starts immediately while the level in tank A is at low level. The tank B is also filled from the same source through tank A to tank B to a desired level, and then drained through valve 2.

The process allows both tanks to be cleaned before full production begins. For this process to be completed, signals, both inputs and outputs, are generated to control the entire application using programmable language.

The Figure 11 shown below shows the schematic diagram of the process and how the components work together to achieve the designed programming. It is made of two pumps, two process tanks, three valves and level indicators (high and low).

The process description is divided into two process sequences namely the steps for controlling the sequences and the actuators that control the components by taking the signals from the steps initiated to complete the sequence.

Data blocks, functions and function blocks are the logics used for the sequences and the required steps and the required steps.
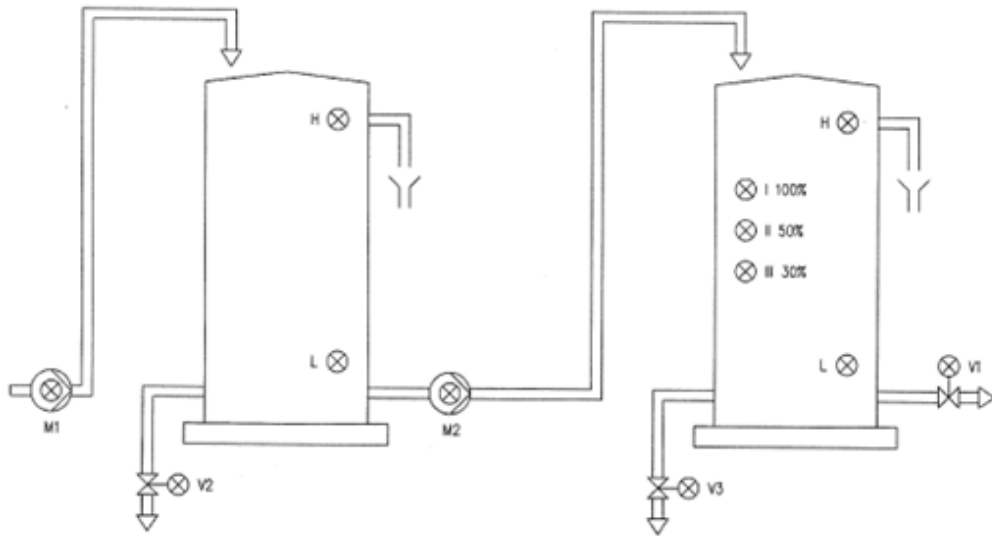
**Figure 11.** The schematic diagram of the process and the components

**Initial state**

The initial state of the process is defined as follows:

- The pump is at stop position (zero speed reference both pumps 1 &2)
- The valves are at open position (valve 1and 2)
- The tanks are empty (low level position tank 1&2)

**Functional sequences for Phase I&II**

The entire process control can be divided into in the following sections:

- Set the level indications to low level in both tanks1&2 (depending on the LIC.PV)
- Close valve V2 and keep valve V1 opened
- Start pump1
- Fill tank1 with water to desire washing level (depending on washing level reached with LIC1.PV)
- Stop pump1 (depending on the washing level reached of the water)
- Open valve 2 to drain the water (After desired level is reached in tank A)

- Close the valve2 again (after the water in tank1 is drained)
- Set the level in tank1 to minimum washing level (using the LIC1.PV)
- Start the pump1 again (depending on minimum washing speed)
- Fill the tank1 to minimum washing level
- Close valve 1 before starting pump2
- Start the pump2 to washing level reached (depending on washing speed for pump2 LIC2.PV)
- Stop both pumps (depending on the washing level reached in tank2)
- Open valve1 &2 to drain the water in both tanks
- Close valve2 after tank1 is empty
- Start pump1 to the steady flow speed (constant speed reference)
- Start pump2 to steady flow speed constant ( depending on the steady flow level reached in tank1)
- Control both pumps to steady flow speed constant reference and stable surface level of water

## 3.3  Detail description of the Functional sequences for Phase I&II

The programming sequence is divided into two main sections, the steps and actuators. The steps are made up of all the necessary actions that need to be controlled and made ready to move the signals to activate the actuators. The signal from each step is transferred to the actuators and when satisfied, gives the required responds to the input channel that will finally give the desired output signals.

The actuators comprise of the valves, PID controllers and some MOVE commands that take the signal responds from the steps than initiates the signals to the components and also compensates for the error that may generate during the process especially the PID controller.

**Steps descriptions**

This section describes what happens in each step and which parameters are controlled in responds to the signal initiated. Each step is ready at a time, under no condition should two or more steps activated at a time.

**Step0 – Initial step brings the process to initial state**

**Transition:** This is the initial step of the entire sequence. At this step, the startup is initiated and no other step is active and it is also reset the whole sequence.

**Controls:** The step is used to control both pumps (1&2) to zero speed reference and it also opens both valves. For these actuators to operate the initial step (**step0**) sets the LIC1.E3 and LIC2.E3 to activate position whiles the speed reference gives the zero speed command to the PID controller to stop both pumps.

**Step ready:** The step is ready when tank1&2 is empty (level of water at Low L1 & L2 indication) and the start sequence activated. (Reference to appendix 1)

**Step1- Close valve 2**

**Transition:** This step can be activated only when the previous step is ready. That is when the tanks are empty at low level indication and the start sequence activated.

**Controls:** the step1 controls the valve 2 to close position.

**Step ready:** The step is ready when valve2 is at the close position with the close indication ON and both tanks empty with the low level indication ON. (Reference to appendix 2)

**Step2-Start pump1 to fill tank1 to washing level**

**Transition:** Step2 can operate when step1 cond_out is ready. This can be activated when the previous step is also ready.

**Controls:** This step controls the speed of pump1 at a constant speed reference. For the pump to run the step activates the SET condition for LIC1.E3. This

operates the PID_CP_FB controller (LIC1) to run the pump1 to the washing speed level.

**Step ready:** The step is ready when the pump indication is at running mode and the washing level is reached in tank1. (Reference to appendix 3)

**Step3-Open valve2 to drain the tank1**

**Transition:** When step2 cond_out is ready, the next step is activated. The transition to this step is ready when the previous step is ready.

**Controls:** The step controls the valve2 in the actuator sequence. When the condition is satisfied, the cond_out gives a signal respond to active and this signal is transferred to the V2 to open valve2 whiles the pump1 is stopped to speed reference zero.

**Step ready:** The step is ready when valve2 is opened and the open indication is activated to ON position. (Reference to appendix 4)

**Step4-Close valve2 after the tank1 is empty**

**Transition:** When the step3 cond_out is active, the idle timer is activated to about 1minute and thirty seconds to enable the valve2 to close after the tank is drained.

**Control:** The step controls the closing of valve2 after draining the tank1 empty. When the ON-DELAY idle timer elapses, the step is activated to V2 actuator to close the valve2. The idle timer was used in the program sequence since no power is required to drain the tank. The water in the tank is drained with its own gravity so the ON-DELAY idle timer is used to activate this step. The transition to this step4 is ready when the previous step is ready.

**Step ready:** The step is ready when valve2 is closed and the closed indication is at ON position. The tank1 low level indication should also be at ON position. When closed indication is active and the low level indication is also active, than the step is ready to move to the next step. (Reference to appendix5)

**Step5-Start pump1 to fill tank1 to minimum washing level**

**Transition:** When step4 cond_out is active, the next step is activated. This step starts the pump1 again to run to its minimum washing level.

**Control:** The step controls the speed of pump1 at a constant speed reference (minimum washing speed). The step activates the actuator MOVE command to transfer the set speed reference to LIC1_EXT_REF3 control the minimum washing speed control. For this to operate the step activates the MOVE logic command that controls the minimum washing speed. The set minimum level speed reference is moved to the actuator input channel (LIC1.EXT_REF3) to control the speed of the pump1 to minimum washing level.

**Step ready:** The step5 is ready when the pump1 running indication is ON and the minimum level set point is reached. When ready, the cond_out transfers a signal state to the next step. (Reference to appendix 6)

**Step6-Start pump2 to fill tank2 to washing level**

**Transition:** When the previous step cond_out is active, than that step is ready. The step5 cond_out activates the next (step6) whiles its resets the previous step when ready.

**Control:** This step controls the speed of pump2 at a constant speed reference to washing level in tank2. To achieve this, the signal from the step6 obj_start activates the actuator MOVE which than controls the speed of the pump2 by transfer the set speed reference to LIC2_EXT.REF3. The move command is used to transfer the washing speed reference to the actuator (LIC2).

**Step ready:** The step is ready when the pump2 is running and the running indication ON. The washing level should also be at its set level to make the step ready. (Reference to appendix 7)

**Step7-Open both valves1&2 to drain both tanks**

**Transition:** When the previous step is ready, the step cond_out than activates the next step which is step7. The transition to the step is ready when the previous step is ready.

**Controls:** This step controls both valves1&2 to open position for the water to drain out of both tanks. The step also controls the speed of both pumps to zero speed reference by activating the MOVE command that controls the stopping of both pumps to zero speed reference.

**Step ready:** The step is ready when both valves1&2 are opened with the open indications ON and both tanks are empty with also the low level indications ON. (Reference to appendix 8)

**Step8-Close valves2 after both tanks are empty**

**Transition:** The transition to the step is ready when the previous step is ready. The previous step is ready when both valves are opened and the tanks are also empty.

**Control:** The step8 controls the valve actuator V2. When the cond_out for the step is activated, the activated signal is transferred to the valve actuatorV2 to close the valve.

**Step ready:** The step is ready when valve V2 is closed and the close indication ON. Both tanks should be empty with the low level indications also ON. (Reference to appendix 9)

**Step9-Start pump1 to steady flow level**

**Transition:** the transition to the step is ready when the previous step is ready, that is when V2 is closed and both tanks are empty with the low level indications ON.

**Control:** This step controls the steady flow speed reference of pump1 to constant speed reference. The set speed reference is moved from the MOVE command to

the LIC1_EXT.REF3 to run the pump. This is achieved when the signal of the previous step is activated.

**Step ready:** The step is ready when the pump1 run indication lamp is ON and the steady flow level in tank1 is reached. When the step is ready, it activates the next step after 3seconds. (Reference to appendix 10)

**Step10-Start pump2 to steady flow level**

**Transition:** The transition to the step is ready when the previous step is ready, that is when the steady flow level is reached and the pump1 running indication is ON.

**Control:** This step controls the steady flow speed reference of pump2 to constant speed reference. The set speed reference is moved from the MOVE command to the LIC2_EXT.REF3 to run the pump. This is achieved when the signal of the previous step is activated.

**Step ready:** The step is ready when the pump2 running indication lamp is ON and the steady flow level in tank2 is reached. Both pumps should running continues at a constant speed reference. (Reference to appendix 11)

**Actuator descriptions**

This section describes what happens in each actuator and which step controls each of the actuators. The actuators are controlled by the signals from the steps and the output responds are activated depending upon what action the step is recommended to initiate. The actuators output signals are moved the control the components such as the pumps, valves, indication lamps, transmitters etc.

**Actuator valve1 control (V1)**

The V1 actuator is the ON-OFF type of circuit control valve actuator. The valve is normally closed or opened depending upon the condition transferred into the input signal channel. The valve can be controlled either manually or auto mode control

depending on the choice of the operator. The mode can be selected by the operator, using the operator workstation SCADA user interface which has the graphical indication mode controllers.

The steps that control the actuator V1 to open position are as follows; step0, and step7. These steps are used in the auto mode control and when activated, it sets the valve to open position depending on which step is ready. The open indication comes to ON when the condition given is satisfied. The signal is activated, it than transfers that output signal to the actual valve to put the valve to open position with the open indication lamp ON. Another step that controls the valve to close position is step6. This step is also used in the auto mode control and when it is ready, it activates the valve1 to close position depending on which step is ready and active. The close indication is set to active when this step is initiated. The programming sequence is designed so that, when the valve open position is activated, than the close indication is set to close whiles if the close indication is activated, it sets the valve to open position. This is done to prevent both indications at ON position at the same time when two or more steps are activated.

The valve can be controlled manually opened or closed when the manual mode is set. The operator workstation has control push buttons for opening and closing the valves by the operator. When the manual mode is set, it's automatically puts the auto mode to OFF position and vice versa when the auto mode is set to ON position. All the actuators are controlled by the STEP.OBJ_START from the steps sequence and it puts the actuators to active mode or deactivate mode. (Reference to appendix 12)

**Actuator valve2 control (V2)**

The actuator valve2 controls the valve to normally open or close. The V2 actuator is also the ON-OFF type of circuit control valve actuator. The valve is normally closed or opened depending upon the condition transferred into the input signal channel. The valve can be controlled either manually or auto mode control depending on the choice of the operator. The steps that controls the actuator V2 to

open position are as follows; step0, step3 and step7. These steps are used in the auto mode control and when activated, its sets the valve to open position depending on which step is ready. The open indication comes to ON when the condition given is satisfied. When the signal is activated, it is then transferred the output signal to the actual valve with a DC voltage to control the valve to open position.

The actuator can be controlled to either manual or auto mode depending on the selection of the operator using the operator workstation user interface (SCADA).The steps that are used to control the same valves to closing mode are; step1, step4 and step8. The V2 actuator has its own indications to show which command is active at a time. Both indications are controlled by open command or the close command. (Reference to appendix 14)

**Actuator pump1-level control for tank1**

The level of the water in the tanks is controlled by the actuator pumps. Each of the tanks has its own actuator pump that controls the level of water using the signals from the analog input channel (LIC1.AI_CH). The measurement of the water level in the tanks are done by using a voltage signal between 2-10V, which is scaled to the measurement of the water surface between 0-100%. The actuator PUMP1 CONTROL (LIC1) is controlled by step0, step3 and step7. All the steps stated are used together with the MOVE command to stop the pump1 to zero speed reference. The MOVE command sends the reference set value (zero) which is REAL to LICI.EXT_REF3. The reference signal, which is zero, is transferred to stop the pump from running.

The other step that controls the actuator pump1 is the step2. This step runs the pump to its washing speed. The speed reference is set to the MOVE command which is transferred to the LIC1.EXT_REF3. Another step that is used to control the actuator is step5 and step6. These steps activate the actuator to control the pump1 to minimum washing speed. The washing speed is set to the MOVE

command that is transferred to the LIC1.EXT_REF3 to run the pump to the minimum washing speed.

Step9 is used to run the pump1 to steady flow speed. After the washing phase of tank1, the step9 is activated to run the pump1 to steady constant speed. The auto mode controller is used to achieve this steady flow of the pumps since both pumps used are not of the same specifications. During the washing phase, the LIC1.E3 mode is set by step0. The LIC1.E3 is activated to the set mode when the initial step is active. This sets the E3 mode to ON position by default when the process is started.

The LICI actuator has three main parameters that are used to control and measure the level of the water in the tanks. The parameters are the controlled variable (CV1), process variable (PV1) and the set point value (SP1). The process variable is used to measure the level of the water to its set point value. The controlled variable is the output value that can be controlled after the process variable has been compared to the set point value for compensation if possible. The PID controller has all the parameters the control these variables.

The actuator can be set to either manual control or auto control mode, depending on the choice of the operator. The graphic user indications are shown on the SCADA operator workstation. The pump speed is run with a current supply between 4-20mA which is scaled from zero (0) to hundred (100) percentage. (Reference to appendices 13, 16 and 17)

**Actuator pump2-level control for tank2**

The actuator PUMP2 CONTROL (LIC2) is also controlled by step0 and step7 to put the pump to the stop mode. All the steps stated are used together with the MOVE command to stop the pump2 from running mode. The MOVE command sends the reference set value (zero) which is REAL to LICI.EXT_REF3. The reference signal, which is zero, is transferred to stop the pump from running through the output channel of the MOVE command.

The other steps that control the actuator pump2 is the step6 which controls the washing speed set reference value to the LIC2 network block. This step runs the pump to its washing speed and runs until that step is reset. The speed reference is set to the MOVE command which is transferred to the LIC2.EXT_REF3.The pump runs at a constant washing speed reference when the step is activated.

Another step that is used to control the actuator is step10 which run the pump to a steady flow constant speed reference value. This step activates the actuator to control the pump2 to a steady speed control value. The steady speed is set to the MOVE command that is transferred to the LIC2.EXT_REF3 to run the pump to the constant speed that controls the level of the water at a steady state in the tank.

The LIC2.E3 is always set to ON position when the initial step is activated to start the whole process. By default the E3 mode is used to set the pump actuators to active mode when the start command is initiated.

The LIC2 actuator has three main parameters that are used to control and measure the level of the water in the tank2. The parameters are the controlled variable (CV2), process variable (PV2) and the set point value (SP2). The process variable is used to measure the level of the water to its set point value while the controlled variable is the output value that can be controlled after the process variable has been compared to the set point value. The PID controller has all the parameters to control these variables. The actuator can be set to either manual control or auto control mode depending on the choice of the operator. This is done on the operator workstation. (Reference to appendices 15, 18 and 19)

## 3.4  Tank level indications

The programming used for the level indications are logic gates and the PID controller. In this programming sequence, no level switch is used to indicate the level of water in the tank but the PID controller was used instant.

**Tank1 low level indication description**

The network shown below in Figure 12 represents tank1 low level indication lamp logic gate diagram. The comparator logic gate is used to set the lamp to ON indication when the conditions stated to it are satisfied. The less than or equal to logic comparator was used with the PUMP1 CONTROL (LIC1) process variable to control the low level indication lamp. When the processed variable or the measured value elapses less than or equal to the tank1 empty set value, the network becomes active causing the lamp indication Q124.0 to be active. The process variable is a real value that can be controlled depending on the command issued to the reference value. The Data.TKE1 is a value that is set and can be changed or modified to suit the level indication.
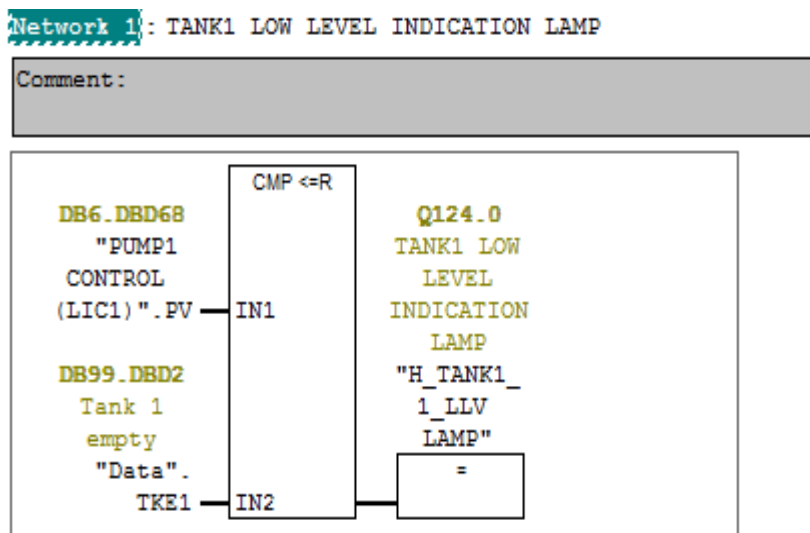


**Figure 12**. Low level indication lamp network block for tank1

**Tank2 low level indication description**

The network shown in Figure 13 is a logic gate comparator greater than or equal to that is used to control the low level indication in tank2. When the PUMP2 CONTROL (LIC2) PV elapses to the set point value Data.TKE2, activates the tank2 low level indication lamp Q124.3 to ON indication. An indication lamp is used to show the indication of the low level water in the tank.

**Tank1 high level indication description**

Level indications used for controlling level of water in water processing plants are important. The high level of water in the process tank is controlled by the network block used below in Figure 14. The logic gate comparator greater than or equal is used together with the process variable and the high level in tank1 (Data.HLT1) to set the tank1 high level indication lamp to ON position. when the process variable (PV) for tank1 elapses with the set point value in Data.HLT1, the lamp turns to active or ON showing the level of the water in tank to 100% full.
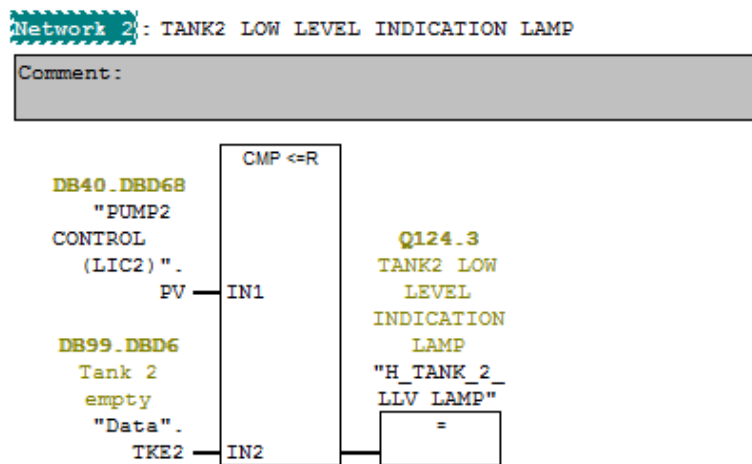


**Figure 13**.Low level indication lamp network block for tank2
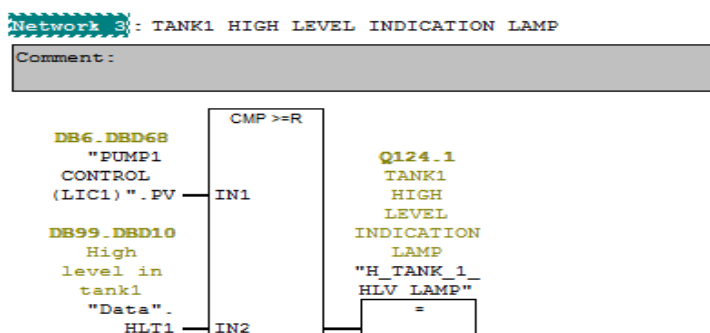


**Figure 14**. High level indication lamp network block for tank1

**Tank2 high level indication description**

The high level of water in the process tank is controlled by the network block used below in Figure 15. The logic gate comparator greater than or equal is used together with the process variable and the high level in tank2 (Data.HLT2) to set the tank2 high level indication lamp to ON position. when the process variable (PV) for tank2 elapses with the set point value in Data.HLT2, the lamp turns to active or ON showing the level of the water in tank to 100% full

```
Network 4 : TANK2 HIGH LEVEL INDICATION LAMP

Comment:


                        CMP >=R
        DB40.DBD68
          "PUMP2
        CONTROL                     Q124.6
         (LIC2)".                   TANK2
             PV ── IN1               HIGH
                                     LEVEL
        DB99.DBD14                 INDICATION
          High                       LAMP
        level in               "H_TANK_2_
         tank2                   HLV LAMP"
         "Data".                      =
          HLT2 ── IN2
```
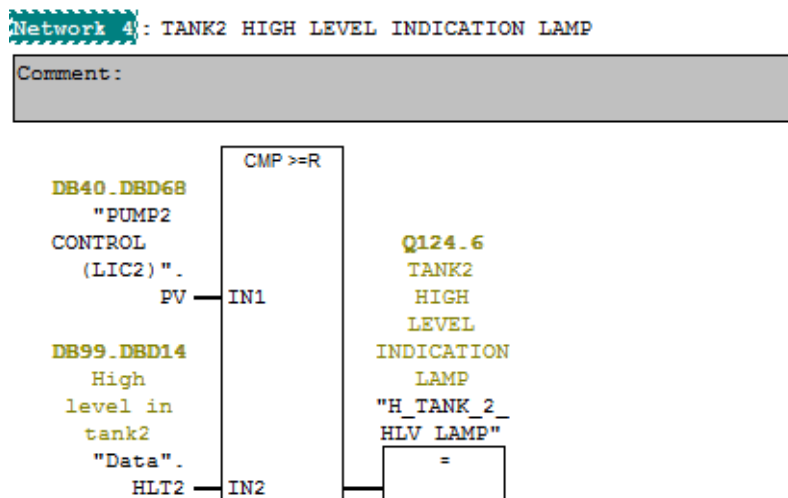
**Figure 15**.High level indication lamp network block for tank2

**Level of water at 30% in tank2**

The network shown below in Figure 16 describes the graphical representation of the level of water at 30% in tank2 during the production process. The equal to logic gate is used with process variable for LIC2 and the set point value in Data.LT30.
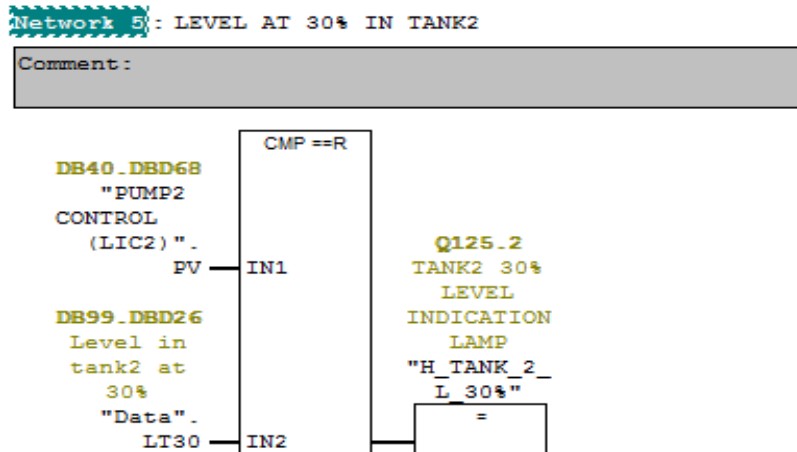
**Figure 16**. Level of water at 30% level indication lamp network block for tank2

**Level of water at 50% in tank2**

The network in Figure 17 shows the logic diagram used to measure the level of water in the process tank2 at half full (50%) level. The equal to logic gate is used together with process variable and the Data.LT50 set point value. The lamp comes to ON when the logic is satisfied.
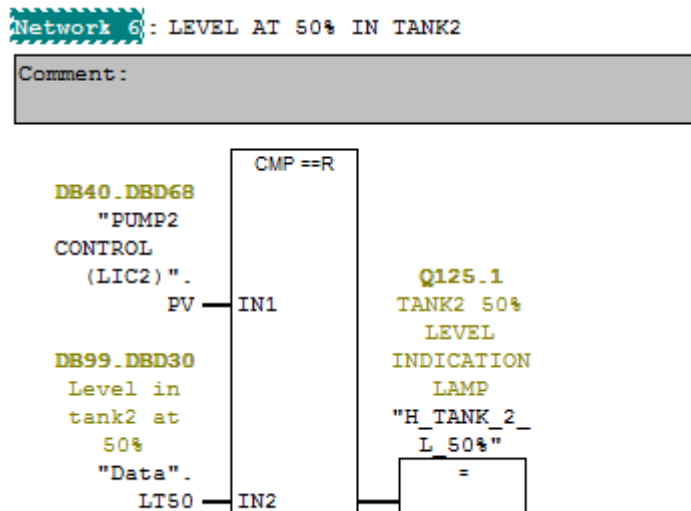


**Figure 17**. Level of water at 50% level indication lamp network block for tank2

```
Network 7: LEVEL AT MAXIMUM LEVEL 100% IN TANK2
Comment:

                    CMP ==R
    DB40.DBD68
       "PUMP2
    CONTROL
       (LIC2)".                Q125.0
          PV — IN1              TANK2
                                 100%
    DB99.DBD34                   LEVEL
     Level in                 INDICATION
     tank2 at                    LAMP
     maximum                  "H_TANK_2_
    value 100%                  L_100%"
      "Data".                      =
      MLT100 — IN2
```
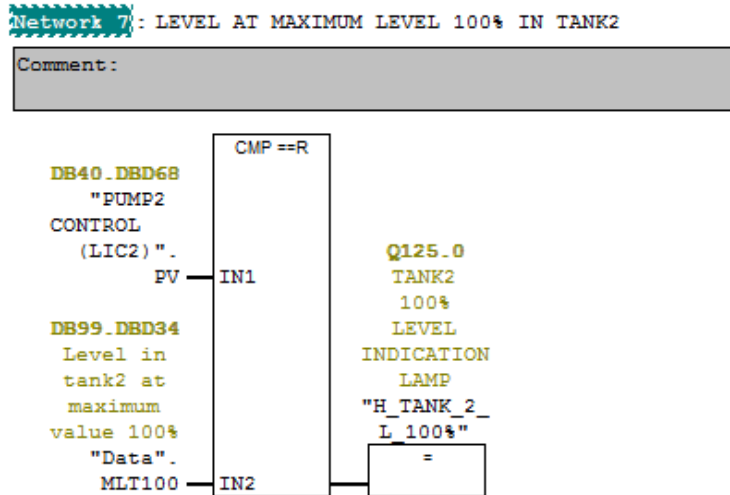
**Figure 18**.Level of water at 100% level indication lamp network block for tank2

**Level of water at 100% in tank2**

The Figure 18 shows the network block used to indicate the level of water in the tank at 100% full. The logic gate used is the equal to logic which sets the lamp to ON when the process variable elapses with the Data.LT100 set point value.

**Level of water at minimum in tank1**

The logic gate in Figure 19 is the graphical representation used to measure the minimum level reached in tank1 during the control process. The greater than logic gate is used to set the lamp indication ON when the process variable elapses with the set point value in Data.MWL. The minimum washing level is used in the logic to start the washing process in tank2 since the pump for tank2 cannot run without water in tank1.
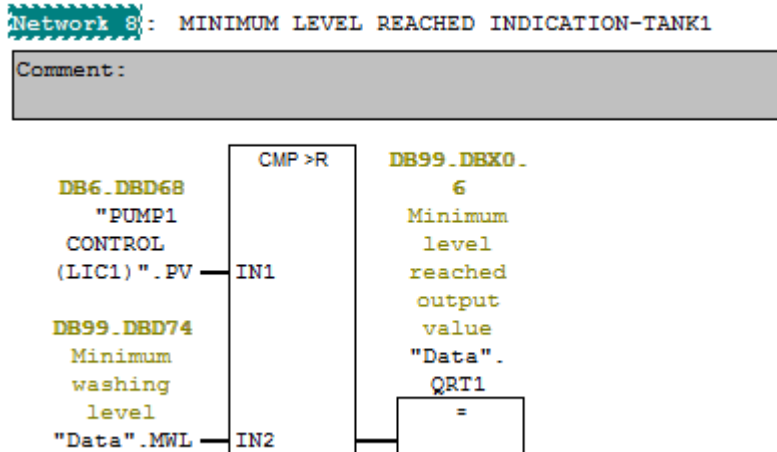
**Figure 19**. Level of water at minimum level indication lamp network block

**Level of water at steady flow indication in tank1**

The logic gate, greater than in figure 20 shown is used to control the steady flow level of the water in tank1. When the process variable elapses with the set point value in Data.SFL1 the indication lamp is put to ON position.
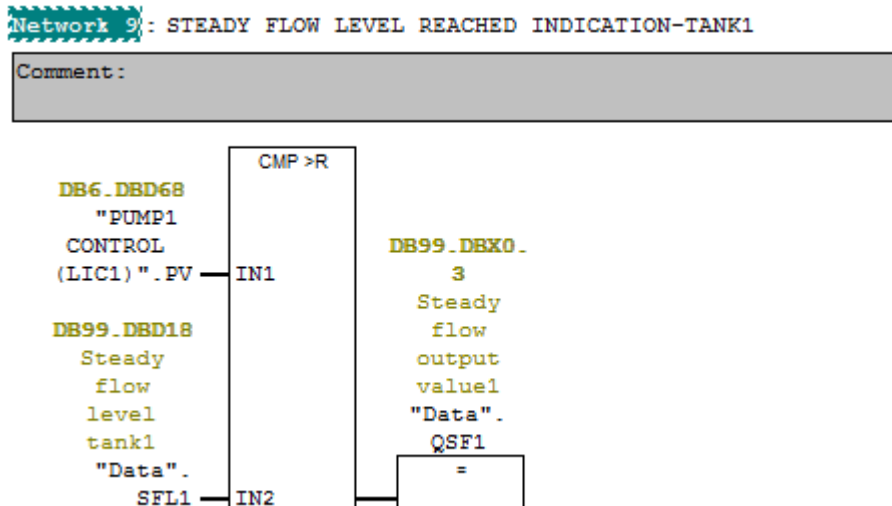


**Figure 20**. Level of water at steady flow level indication in tank1

**Level of water at steady flow indication in tank2**

The logic gate greater than in Figure 21 shown is used to control the steady flow level of the water in tank2. When the process variable elapses with the set point value in Data.SFL2 the indication lamp is put to ON position.
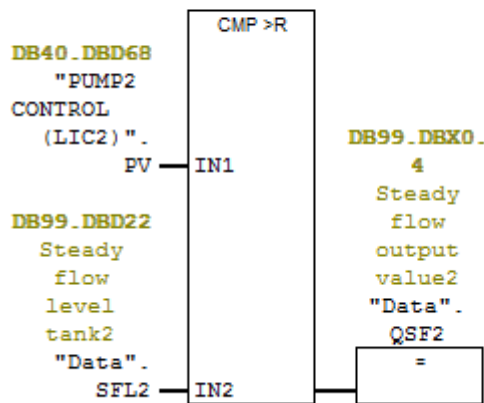


**Figure 21**. Level of water at steady flow level indication in tank2
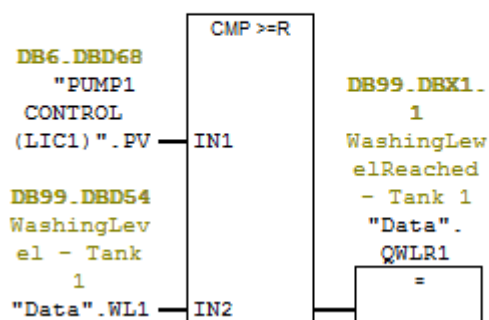


**Figure 22**. Washing level logic gate for tank1

**Washing level in tank1**

The logic gate shown in Figure 22 displays the logic that controls the washing level reached in tank1. The washing level logic gate plays an important role in the automation process for controlling the level of water with the PLC. When the washing level, which is one of the phases of controlling the water in the process tanks, the logic gate greater than and the equal to gates sets the indication output value to washing level reached mode that will put the pump1 to stop before the valve1 is opened for the water to be drained.

**Washing level in tank2**

The logic gate shown in Figure 23 displays the logic that controls the washing level reached in tank2. The logic gate greater than and equal to be used to control the level of water in tank2 when the conditions stated to it is achieved. The output data displays the mode to ON position when the washing level is reached.
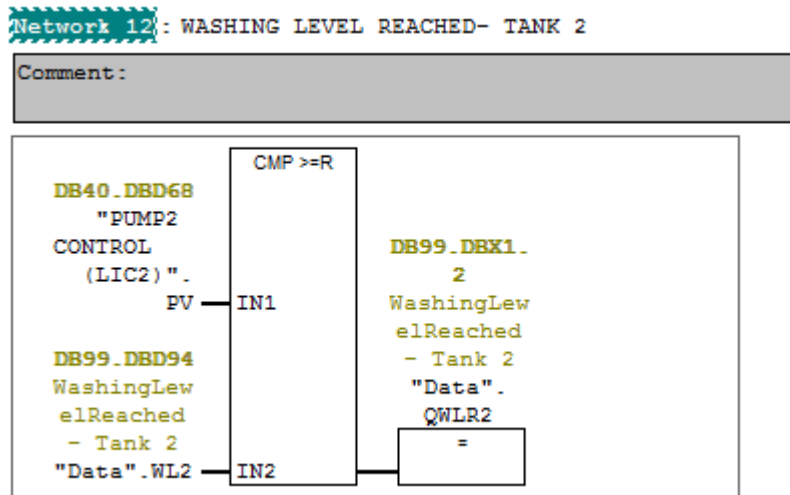


**Figure 23**. Washing level logic gate for tank2

**Open indication lamp for valve1**

The logic gate showed in Figure 24 displays the lamp indication for the open position for valve1. The logic AND gate is used to indicate the lamp to open position when the condition is satisfied. The AND is a logic operator that requires

that all input conditions to be logic 1 for the output to be logic. This implies that the lamp indication Q124.4 will come to ON when the input logic VALVE CONTROL (V1) OPNI is active.
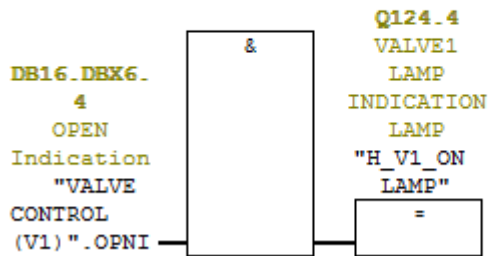


**Figure 24**. Lamp indication for opening valve1 logic block

**Open indication lamp for valve2**

The logic gate shown in Figure 25 displays the lamp indication for the open position for valve2. The logic AND gate is used to indicate the lamp to open position when the condition is satisfied. This implies that the lamp indication Q124.4 will come to ON when the input logic VALVE CONTROL (V2) OPNI is active.
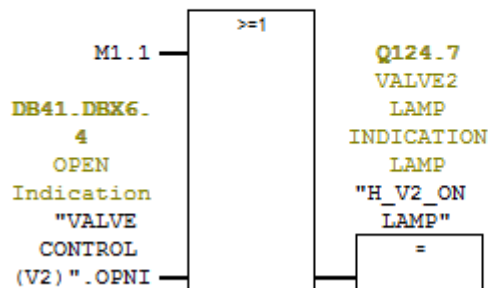


**Figure 25**. Lamp indication for opening valve2 logic block

**Running indication lamp for pump1**

The network below shows the logic operator that controls the running indication of the pump1 at run mode. The operator logic greater than is used to set the pump to running mode when the set point value in Data.SPVT1 elapses with the PUMP1 CONTROL(LIC1).LMN value.
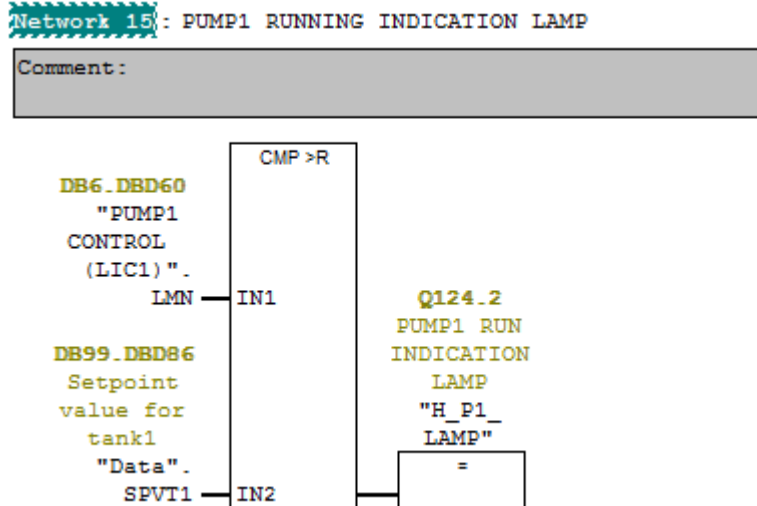


**Figure 26**. Lamp indication for run mode of pump1 logic block
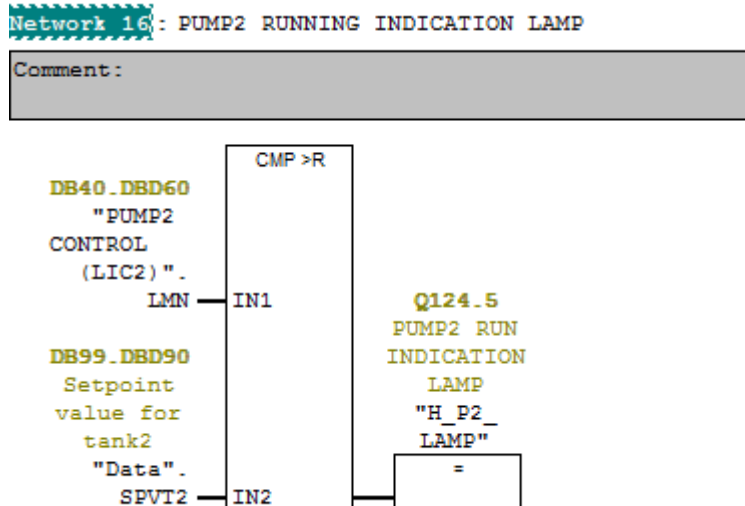


**Figure 27**. Lamp indication for run mode of pump2 logic block

**Running indication lamp for pump2**

The network above shown in Figure 27 is a logic operator that controls the running indication of the pump2 at run mode. The operator logic greater than is used to set the pump to running mode when the set point value in Data.SPVT2 elapses with the PUMP2 CONTROL(LIC2).LMN value.

### 3.5 Operator Workstation Description

The design of operator work station in process automation and control systems plays an important role in field of monitoring and controlling devices which are far away from the production site. The operator workstation is a graphical user interface that is designed to control the entire control automation process and also to help alarm faults and emergency during production.

During the designing of the operator workstation, there were five phases considered; these phases are going to be described in detail. The steps used to create the user interface are five main basic phases used in creating the functional SCADA system for a process control. The activities and the parameters used were considered before the selection of the software to be used to the program. The software used for designing the operator workstation is the In Touch design software development application. (See Appendix 20)

**Phase 1: Design of the system architecture**

The Figure 28 shown below is the graphical representation of the tank installation with the pumps and valves connections.

The design of the system architecture includes all the important communication systems and the parameters to be controlled. All the site instrumentation that are used on site and needed to be monitored and control desired parameters also are involved during the first phase.
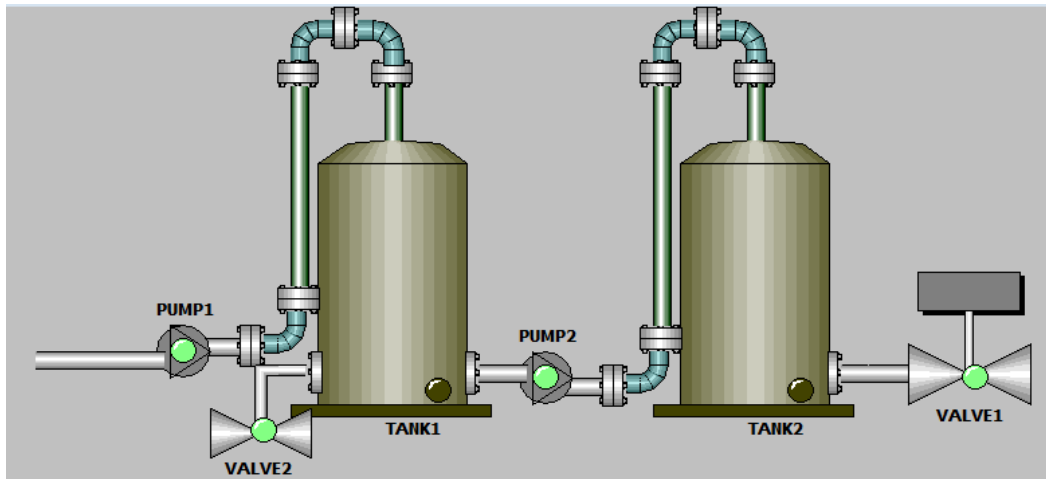
**Figure 28**. The tank installation

The parameters to be controlled on this user interface are the control valves and the pumps. Some of the monitoring indication lamps are displayed in the operator workstation, such as the low level indication lamps, pump running indication lamps and valves control indication lamps.
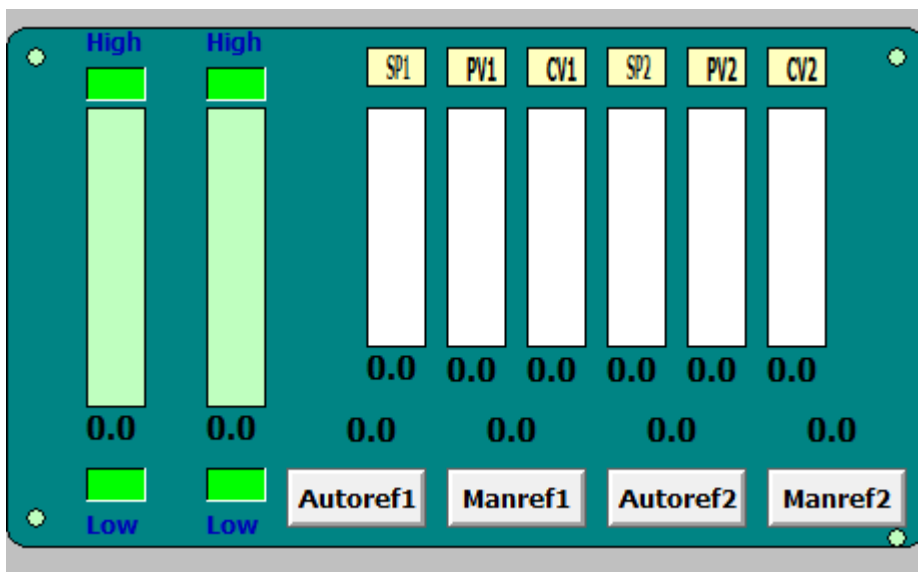


**Figure 29**.Graphic display of the user interface operator workstation

**Level control parameter user interface**

The graphic display above in Figure 29 shows the parameters to be shown and controlled during the use of the operator workstation. The tank level indication

shows the display of the level of water in the tank and variable control data display table is also shown, which is used to control the auto reference and manual reference variables.

**Pump control interface**

The graphic display below is the pump control operator workstation used to control the pumps manual at the manual mode. The user interface is used to control the pumps to run or stop depending on the command issued to by the operator. The pump run indications lamps are also displayed with the meter that displays the speed of the pump when it is running or stopped.

The push button below the meter is used to set the both pumps to auto mode, manual mode or E3 mode. The default parameter is the E3 mode when the process is activated by the initial step. (step0).
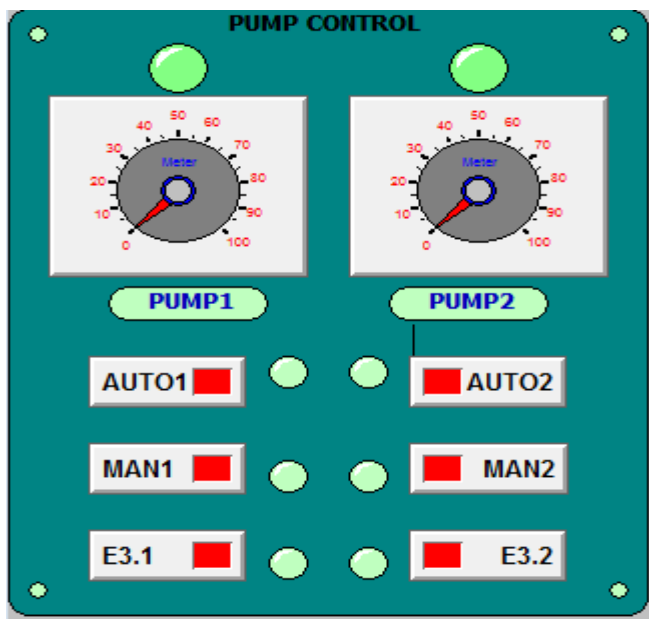


**Figure 30**. Graphic display of the pump control user interface

**Valve control and step indications**

The display below shows the design architecture of the valve control and the step indications. The start and stop push button is used to set the process and the stop

button is used to stop the sequence. The valves can be set to the auto or manual mode control with the indication lamp display showing which of the modes is active at a time.
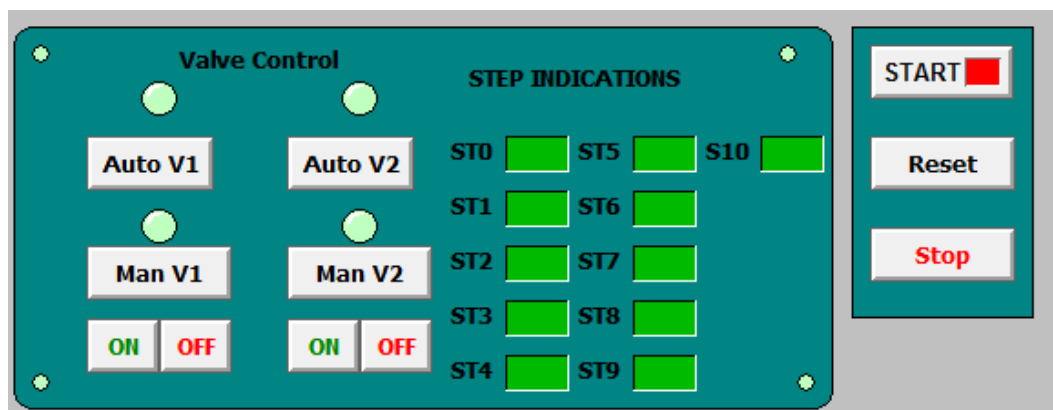


**Figure 31**.Graphic display of the valve control and the step indication lamps.

**Phase 2: Supply of the Remote Telemetry Unit**

This phase of the operator workstation involves the use of communication and human machine interface to link the PLC design program sequence. The type of software used to design the workstation is selected based on the parameters to be controlled and site instrumentations required. The In-Touch software application was used to design the operator workstation after considering the parameters to be controlled and monitored. The tag names used in designing the sequence with the STEP7 application software are stored in the operator workstation software with the identifications showing whether it is a discrete or real input or output value.

All the data address used for the PLC are linked to the remote telemetry unit and the communication units. The cables, moderns and other PC applications and units are connected during this phase to allow the connection between the PLC and the operator workstation to be efficient and safe to the operator and the working environment.

The system management console is the application used to link the operator workstation software to the PLC after all the designed programming are accurate and ready to operate.

**Phase 3: Programming**

The programming software application is designed during this phase of the operator workstation. The human machine interface is designed with In-Touch software application with powerful and safe graphics. Selections of the push buttons, indication lamps, alarms and some other parameters are programmed with the application software.

The tag names are linked to the push buttons and the instruments to be controlled on site during this phase of designing the operator workstation graphic user interface. During the programming phase, some important parameters were considered such as the alarming systems, the steps indications displaying which step is active at a time and the tank level indications displaying the level of water in the both tanks to low or high in order to stop the process when the tanks are low or over the high level set point value.

Selected user graphics are considered during the programming phase. Some graphics used are the valves, tanks, level indications lamps, pipes and other important instrumentation parameters that will be used on the production site.

**Phase 4: Installation of hardware component**

During the installation phase the hardware components are configured. The computers and the PLC units are connected together with the communication control units. The required communication parameters such as the modern, internet cables, computers are all connected together to run the operator workstation. Since the process tank has all the internal relays and power supply connections, much was not done during this project. The connections made were done between the PC and the PLC with internet cables and the modern. The

operator workstation was finally connected to the PLC and test ran after the check.

**Phase5: Commissioning**

This phase is the final stage of the operator workstation design. During this final phase of the program, the software together with the hardware will be ran to identify possible problems and fault that may arise during the production.

The system was run repeatedly to make sure that all part of the program is accurate and safe to run. All necessary corrections were made after the problems are indicated before on the hardware and the software application.

After the operator workstation was run repeatedly, some modification and correction were made with the recommendations from my supervisor. Some of the modifications made were e.g. the change of the valve control user interface and the pump control design human machine interface. It was necessary to make those changes due to safety and accuracy of the operator workstation program application.

The final documentation and acceptance was considered after the supervisor has accepted to work logic and application of the entire production.

# 4. CONCLUSION

This project was performed using control process system applications, supervision and monitoring of water treatment sample plant intended for the water treatment production industries.

It was intended to implement an expert system of process control and also be complemented with a set of program design and graphic user interface. The choice of a PLC as a control device, not only assures the guaranteed of the interconnectivity and compatibility of the entire process automation and various equipment through interfaces, but also its facilities the interoperability use of SCADA implementation but at the same time, it gives flexibility for future upgrades or modification of the design process.

The In-Touch SCADA, with which the Human Machine Interface (HMI) or operator workstation of the system was developed, has helped to reduce the processing times of the project due to its versatility and ease of programming. The user interface used for the design was simple and much easier to program and install the necessary parameters and units need to be controlled.

The reduction of the presence of human staff to a single operator is not detrimental to the operation of the process and installation, because the expert system proposed improves the quality of work of the operator and also reduces human errors which may cause problems to the production.

All this has an impact on increasing the security of the installation, and also prioritizes environmental protection and the main objective of this process control.

# REFERENCES

Ahti, M. Lembit, R. 2004. Industrial Engineering − Innovation as Competitive Edge for SME. Tallinn Technical University. Programmable Logic Controllers In, p55-57.

Abdullah, M.A.B.2008. Water level in tank using level sensor and PID controller. University Malaysia Pahang Faculty of Electrical & Electronics Engineering).Pahang. Accessed 10.3.2012.

Bischoff, H. & Hofmann, D. Terzi, E.V 1997. Process Control System. Denkendorf. Festo Didactic GmbH & Co.

Cooper, D J. 2006. Controlguru Practical Process Control. Connecticut, United States of America. Accessed 24.2.2012. http://www.controlguru.com/.

Dunning, G.2006.Introduction Programmable Logic Controllers. Accessed 24.3.2012.http://processcontrol.analog.com/en/programmable-logic-controllers/segment/pcia.html?gclid=CJqiqpXZgrACFcQtmAodaU-rsg

Gould, L.S. 2006 Programmable Logic Controllers Transformed. Automotive Design & Production. ABI/Inform Global, 56-58.

Hasim, N.B.2008.Water Level and Temperature Control Using Programmable Logic Controller (PLC). University Tehnikal Malaysia.Melaka.Accessed 14.3.2012

Ismail, M.H.B. 2008. Rain-water PLC based detector and valve switcher. University Malaysia Pahang. Pahang.1-13

Jack, H. 2012. Automating Manufacturing Systems with PLC-s. Accessed 16.3.2012.http://engineeronadisk.com/V2/book_PLC/engineeronadisk-78.html#pgfId-37303

Petruzella, F. D.2004.Activities Manual for Programmable Logic Controllers. p267. McGraw-Hill.

Safety Manager Peltonen J.2012. MIPRO Oy.MIPRO Oy web site.Finland .Accessed 7.3.2012.http://www.mipro.fi/eng.

Siemens.2010.Programming with STEP 7 Manuel A5E02789666-01.Accessed 7. 02 2012. http://www.siemens.com/simatic-tech-doku-potal.

Synchrony.2001. Trends in SCADA for Automated Water Systems. Accessed 3.4.2012. http://www.sunapsys.com/trends_SCADA.pdf.

Shaw, J.A.2006. Process Control Solutions.Rochester Duke Power Company, Taylor Instrument Company(ABB, Inc).2nd Edition. Accessed 1.4.2012. http://www.amplicon.com/info/i-o/index.cfm/rw.cm/google,ppc,process-controls.htm?gclid=CNei5fnZgrACFecumAodB0bEEg

Siemens A.G. 2003.Standard PID Control (manual).Nuemberg.Accessed 5.3.2012.http://cache.automation.siemens.com/dnl/DQ/DQ1MzA5AAAA_113708 4_HB/Stdpid_e.pdf

Technologies, IDC.2007. Industrial Autoamtion,IDC Engineering Pocket Guide

APPENDIX 1

**Network 17 : INITIAL STEP - Brings the proces to initial state**

```
TRANSITION TO THIS STEP:
- At start up - No other step is active
- Reset the sequence

CONTROLS:
-Pump 1 speed reference 0
-Pump 2 speed reference 0
-Valve (V2) open
-Valve (V1) open

STEP READY:
-Tank 1&2 empty - -Level of water at low L1 indication
-Start sequence                              .
```
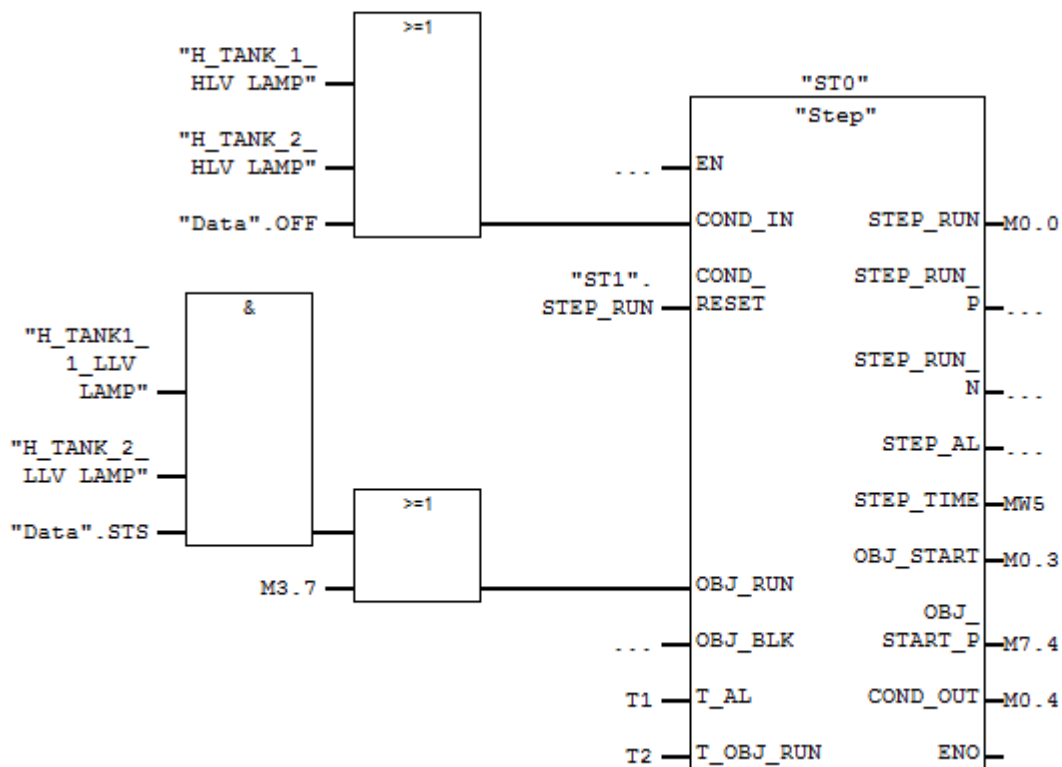
APPENDIX 2

Network 18 : START (CLOSE VALVE2)

TRANSITION TO THIS STEP:
- Prevoius step is ready

CONTROL:
-Closing the valve 2

STEP READY:
-Valve 2 closed, closed indication
-Tank1 low level indication
-Tank2 low level indication
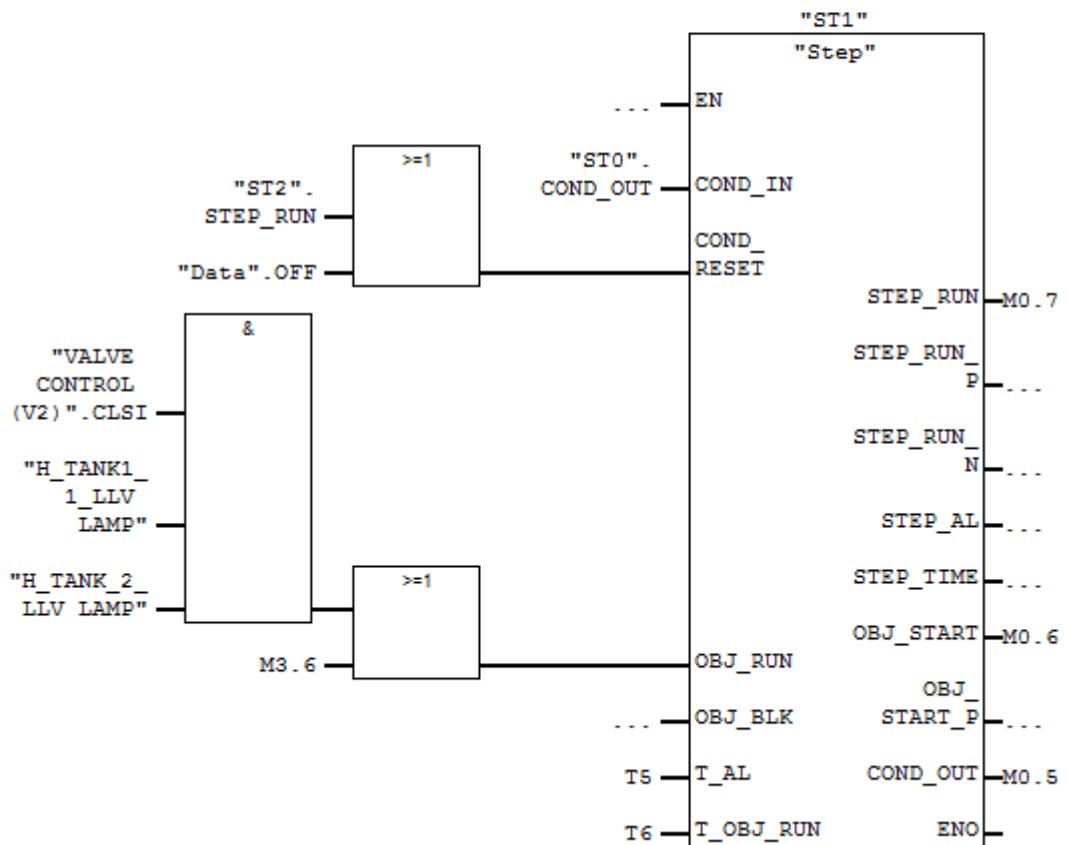
APPENDIX 3

Network 19 : Fill the tank to washing level -START PUMP1

```
TRANSITION TO THIS STEP:
- Prevoius step is ready

CONTROL:
-Speed of pump1 - constant speed


STEP READY:
-Pump 1 running
-Level is at washing level (50%) in tank 1
```
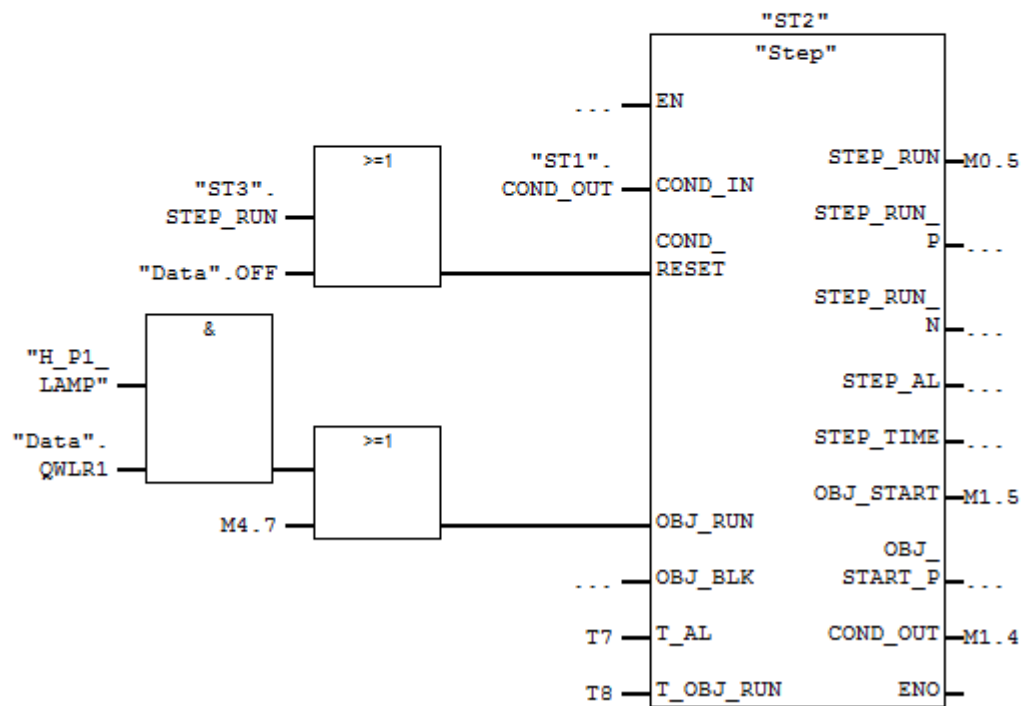
APPENDIX 4

Network 20 : Draining the tank1- open valve 2

```
TRANSITION TO THIS STEP:
- Prevoius step is ready

CONTROL:
-speed to zero reference p1
-opening valve 2

STEP READY:
-Valve 2 opened
-Open indication on
```
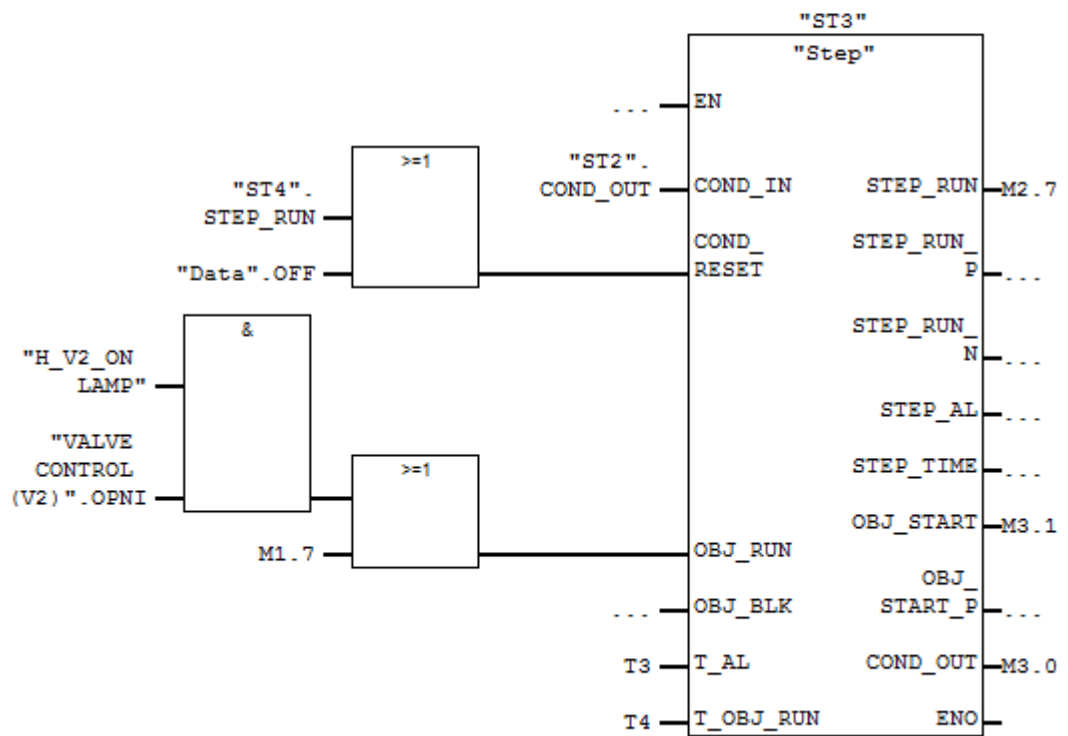
APPENDIX 5

**Network 21**: Fill tank1 to minimum level – Close valve 2

```
TRANSITION TO THIS STEP:
- Prevoius step is ready

CONTROL:
-Closing the valve 2

STEP READY:
-Valve 2 closed, closed indication
-Tank1 low level indication
```
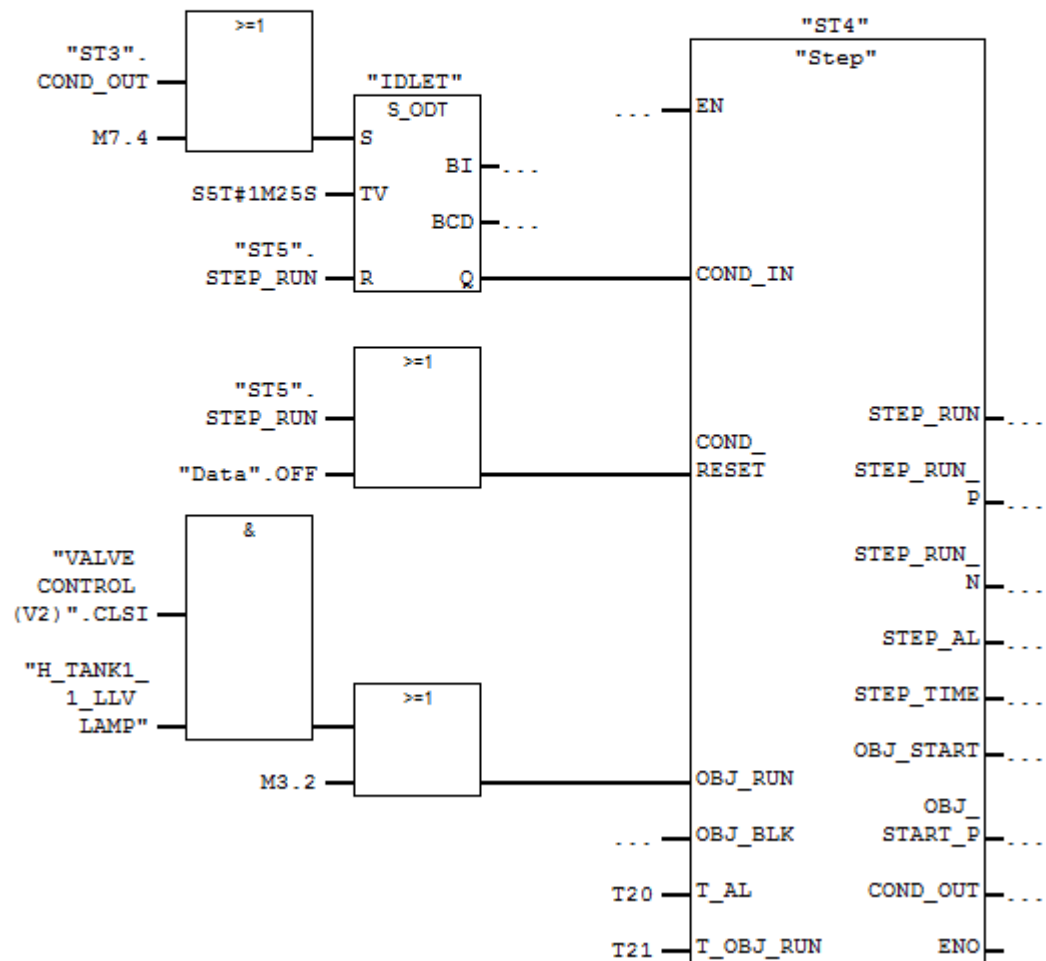
APPENDIX 6

Network 22 : Fill tank1 to minimum level - start pump1

```
TRANSITION TO THIS STEP:
- Prevoius step is ready

CONTROL:
-Speed of pump1
-controlling the level minimum

STEP READY:
-Pump 1 running
-level is at minimum in tank 1
```
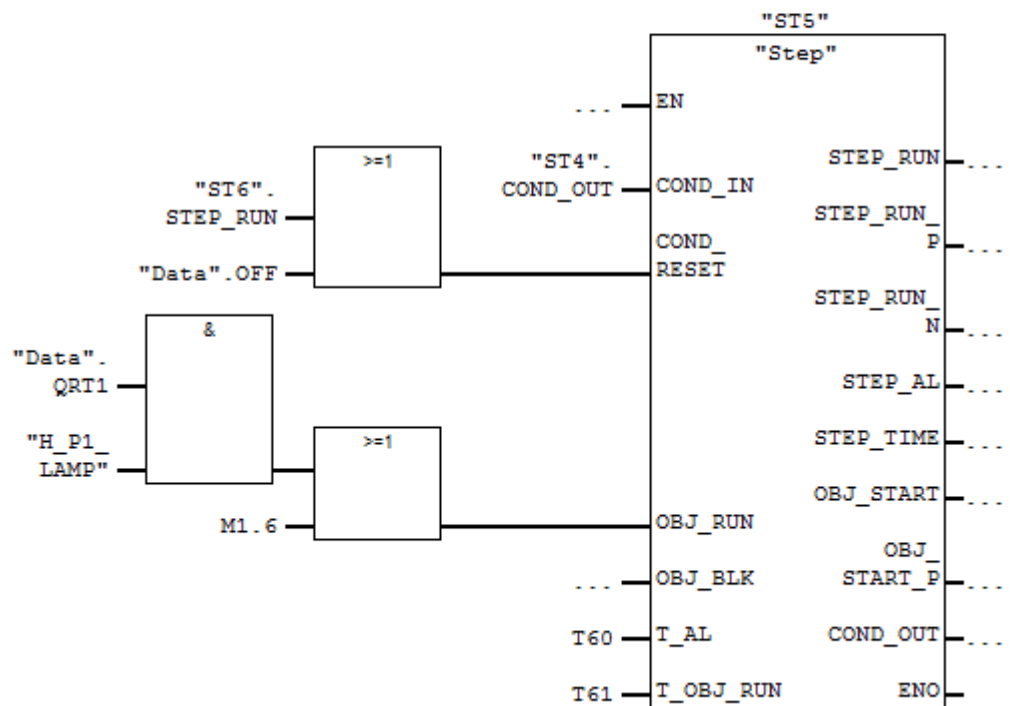
APPENDIX 7

Network 23 : Fill the tank2 to washing level -START PUMP2-Closing the valve 1

```
TRANSITION TO THIS STEP:
- Prevoius step is ready

CONTROL:
-Speed of pump2 - constant speed
-Closing the valve 1


STEP READY:
-Pump 2 running
-Level is at washing level (50%) in tank 1
```
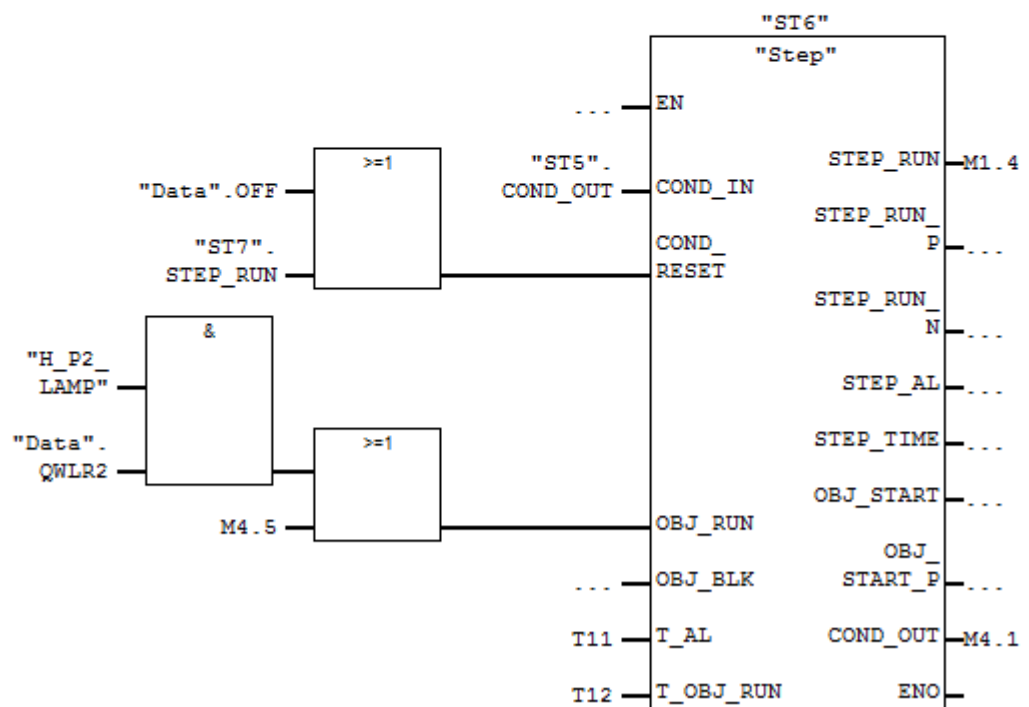
APPENDIX 8

**Network 24**: Draining water in both tanks-OPEN BOTH VALVES 1&2

```
TRANSITION TO THIS STEP:
- Prevoius step is ready
CONTROL:
-speed to zero reference at both pumps
-opening valve1&2
STEP READY:
-Valve1&2 is opened
-Level is empty, level of water at low L1 in both tanks indication
-Open indication on both valves
```
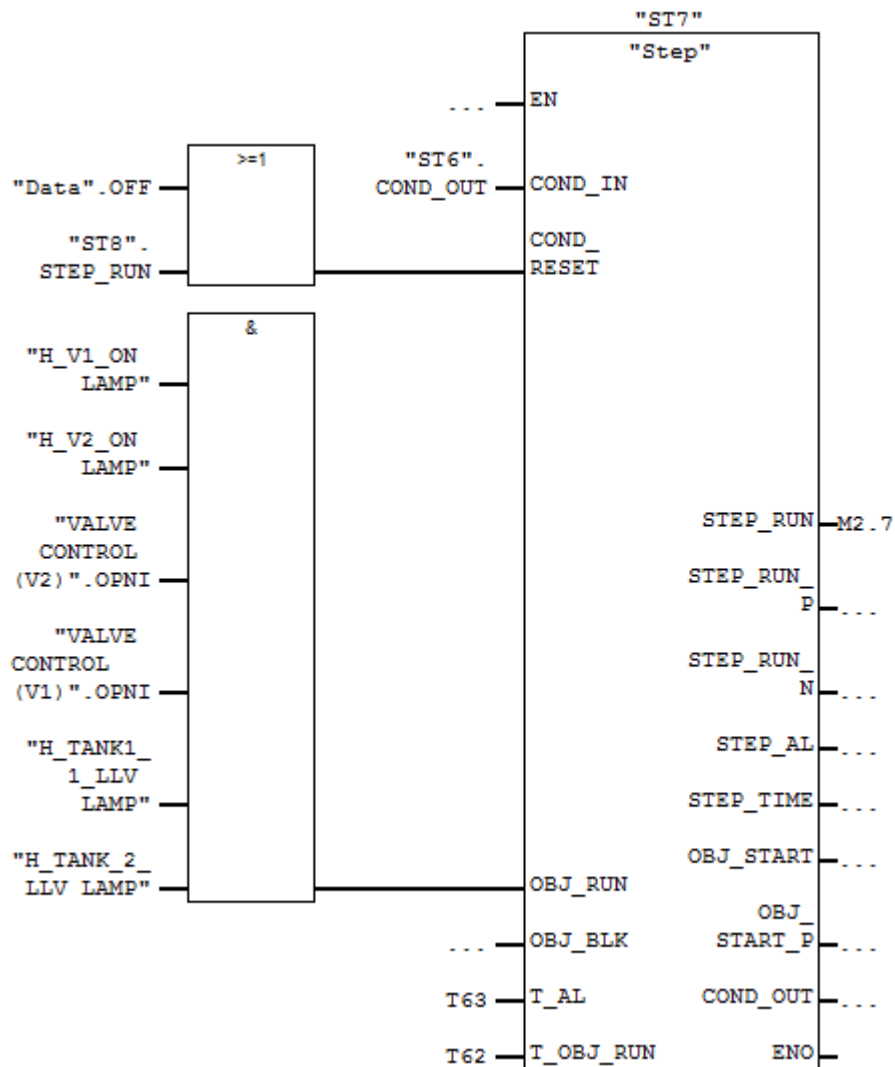
APPENDIX 9

Network 25 : START (CLOSE VALVE2)

```
TRANSITION TO THIS STEP:
- Prevoius step is ready

CONTROL:
-Closing the valve 2

STEP READY:
-Valve 2 closed, closed indication
-Tank1 low level indication
-Tank2 low level indication
```
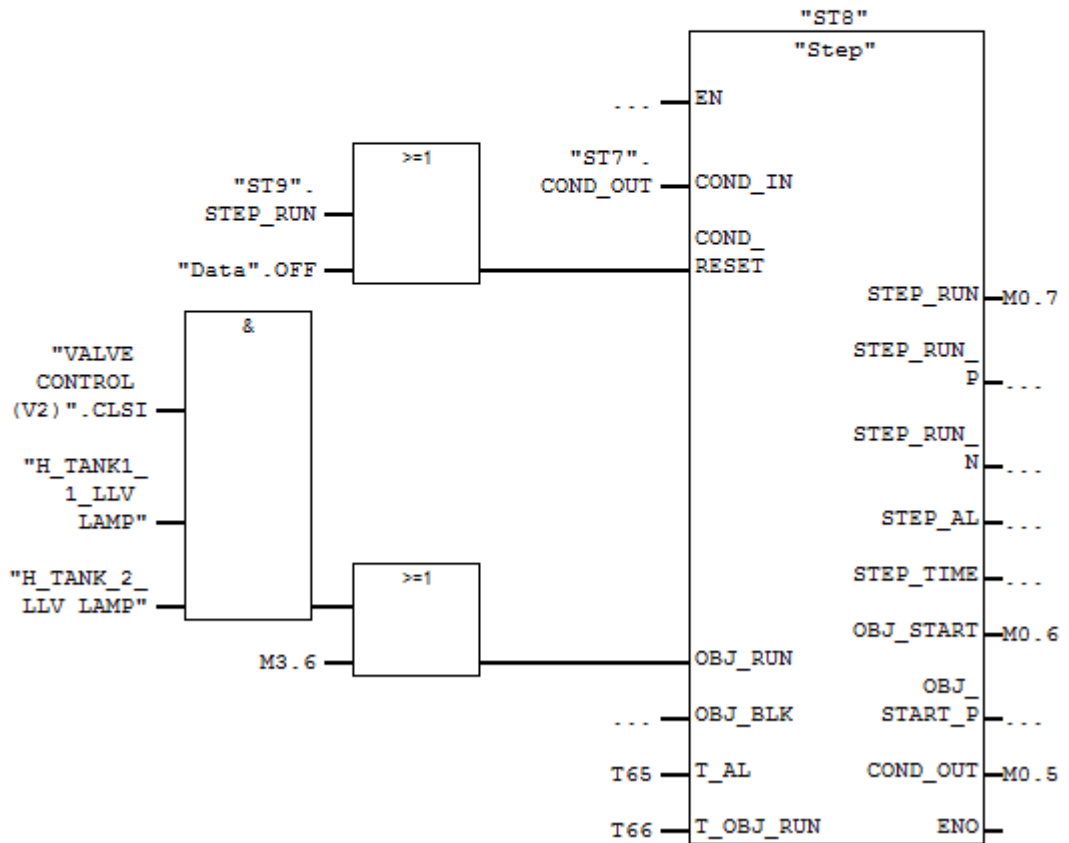
APPENDIX 10

Network 26 : Fill the tank1 to STEADY FLOW level -START PUMP1

```
TRANSITION TO THIS STEP:
- Prevoius step is ready

CONTROL:
-Speed of pump1 - constant speed


STEP READY:
-Pump 1 running
-Level is at steady flow level in tank 1
```
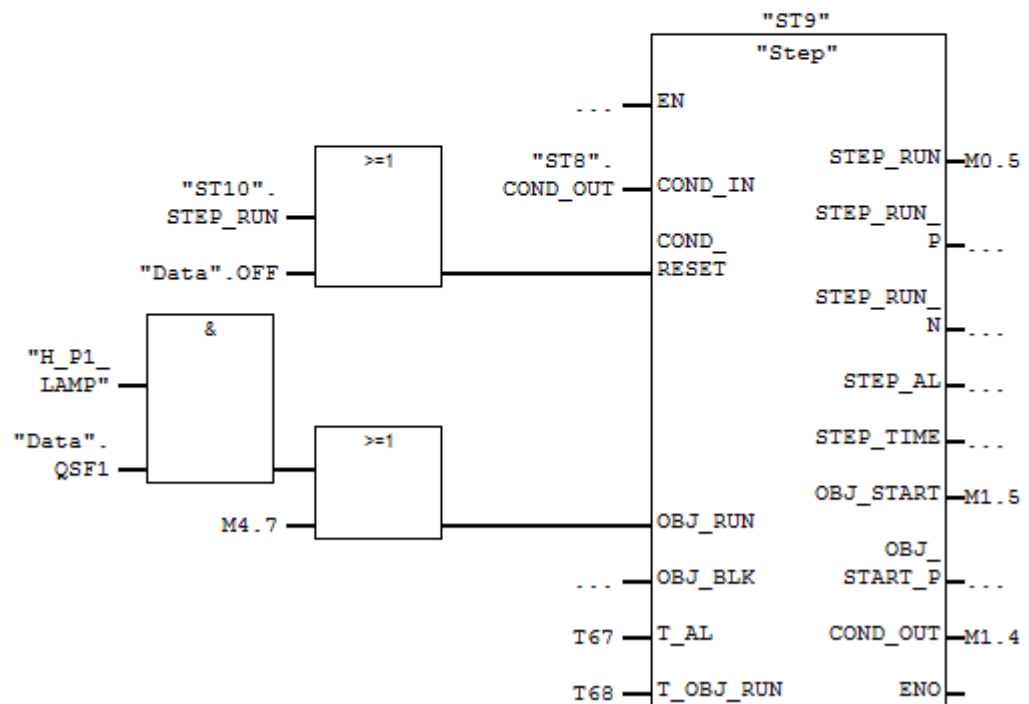
APPENDIX 11

Network 27 : Fill the tank2 to STEADY  level -START PUMP2

```
TRANSITION TO THIS STEP:
- Prevoius step is ready

CONTROL:
-Speed of pump2 - constant speed



STEP READY:
-Pump 2 running
-Level is at steady flow level  in tank 2
```
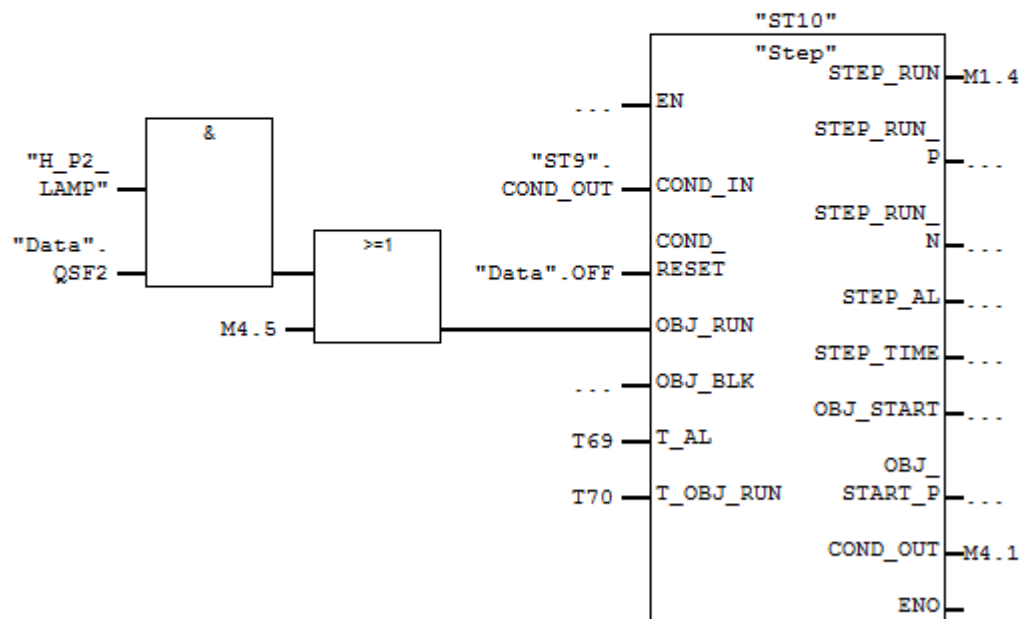
APPENDIX 12

Valve 1 control
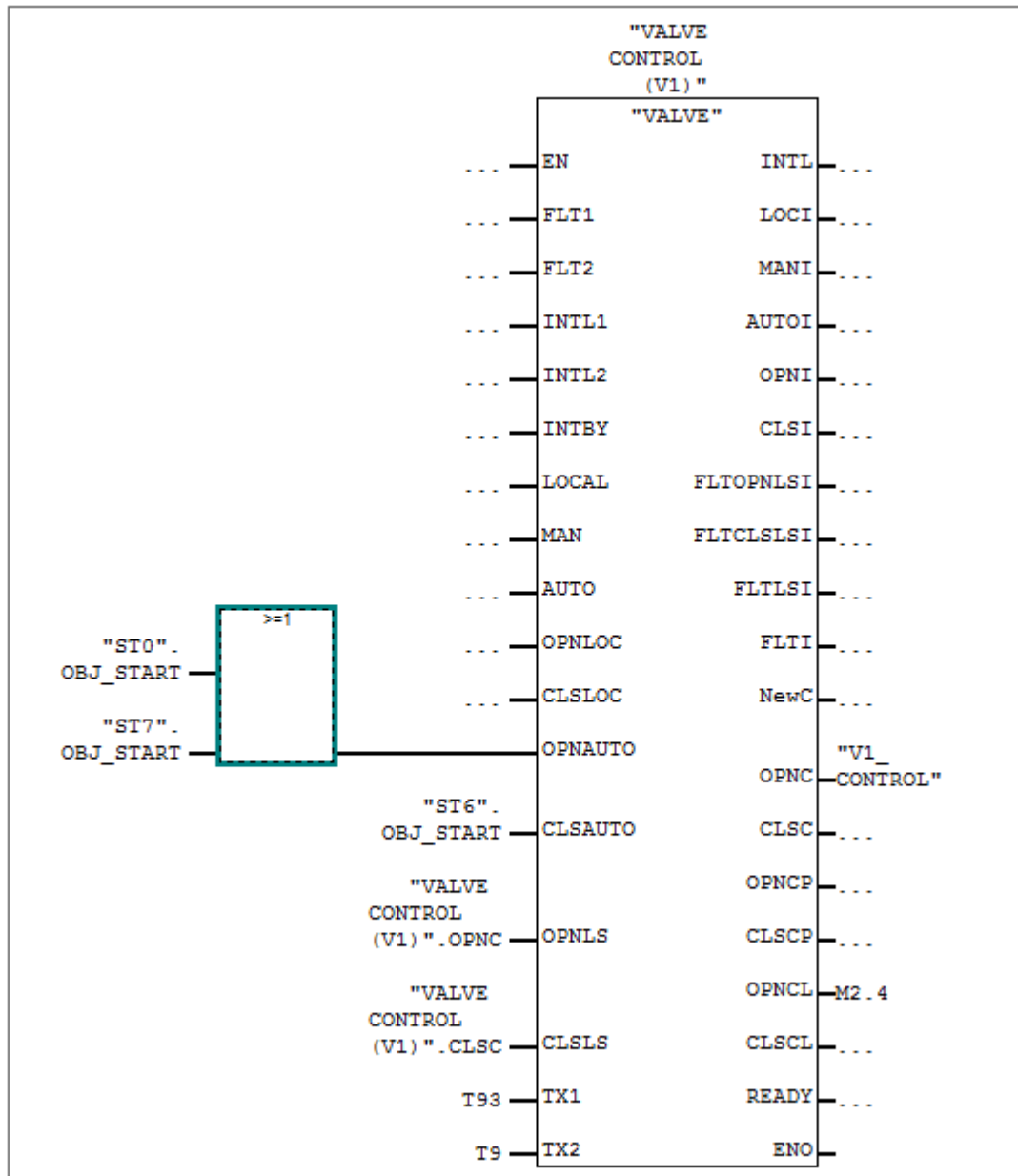
APPENDIX 13

Level control for tank 1 - Actuator is the pump 1

```
                        (LIC1)
                      "PID_CP_FB"
        ... ──│ EN
        ... ──│ EXT_REF1
        ... ──│ EXT_REF2
        ... ──│ AUTO_REF
        ... ──│ FF_REF
        ... ──│ BAL_REF
        ... ──│ MAN_REF
                              Spare1 │── ...
        ... ──│ SP_HLM
                              Spare2 │── ...
        ... ──│ SP_LLM
                              Spare3 │── ...
        ... ──│ LMN_HLM
                              Spare4 │── ...
        ... ──│ LMN_LLM
                                E1_I │── ...
    "TANK_1_
     LEVEL" ──│ AI_CH          E2_I │── ...

    "PUMP1                    AUTO_I │── ...
   CONTROL
    (LIC1)".                   BAL_I │── ...
   EXT_REF3 ──│ EXT_REF3
                                E3_I │── ...
        ... ──│ SP_RAMP
                               MAN_I │── ...
        ... ──│ SPROC_ON
                                QCAS │── ...
        ... ──│ E1
                                 LMN │── ...
        ... ──│ E2
                                  SP │── ...
        ... ──│ BAL
                                  PV │── ...
        ... ──│ AUTO
                                      "P1_SPEED
        ... ──│ E3          AO_CH │─ CONTROL"
        ... ──│ MAN           ENO │──
```

APPENDIX 14

```
                                    "VALVE
                                   CONTROL
                                    (V2)"
                                   "VALVE"

                        ...  ─  EN

                        ...  ─  FLT1

                        ...  ─  FLT2

                        ...  ─  INTL1
                                                    INTL  ─  ...
                        ...  ─  INTL2
                                                    LOCI  ─  ...
                        ...  ─  INTBY
                                                    MANI  ─  ...
                        ...  ─  LOCAL
                                                   AUTOI  ─  ...
                        ...  ─  MAN
                                                    OPNI  ─  ...
                        ...  ─  AUTO
                                                    CLSI  ─  ...
                        ...  ─  OPNLOC
                                                FLTOPNLSI ─  ...
                        ...  ─  CLSLOC
                                                FLTCLSLSI ─  ...

                                                   FLTLSI ─  ...

                                                    FLTI  ─  ...

                                                    NewC  ─  ...
                                                            "V2_
                                                    OPNC  ─ CONTROL"

                                    CLSAUTO         CLSC  ─  ...
```

```
 "ST0".
OBJ_START ─┐ >=1 │
 "ST3".     │    │
OBJ_START ─┤    │
 "ST7".     │    │
OBJ_START ─┘    │──── OPNAUTO

 "ST1".     ┌ >=1 │
OBJ_START ─┤    │
 "ST4".     │    │
OBJ_START ─┤    │
 "ST8".     │    │
OBJ_START ─┘    │──── CLSAUTO
```

```
 "VALVE
CONTROL
(V2)".OPNC ─ OPNLS        OPNCP ─ ...

                          CLSCP ─ ...
 "VALVE
CONTROL                   OPNCL ─ M2.4
(V2)".CLSC ─ CLSLS

                          CLSCL ─ ...
      T90  ─ TX1

                          READY ─ ...
      T91  ─ TX2

                            ENO ─
```

APPENDIX 15

```
                              "PID_CP_FB"
        ... ──┤EN
        ... ──┤EXT_REF1
        ... ──┤EXT_REF2
        ... ──┤AUTO_REF
        ... ──┤FF_REF
        ... ──┤BAL_REF
        ... ──┤MAN_REF
        ... ──┤SP_HLM            Spare1├── ...
        ... ──┤SP_LLM            Spare2├── ...
        ... ──┤LMN_HLM           Spare3├── ...
        ... ──┤LMN_LLM           Spare4├── ...
     "Tank_2_                      E1_I├── ...
      LEVEL" ──┤AI_CH             E2_I├── ...
      "PUMP2                     AUTO_I├── ...
     CONTROL                      BAL_I├── ...
      (LIC2)".                     E3_I├── ...
     EXT_REF3 ──┤EXT_REF3         MAN_I├── ...
        ... ──┤SP_RAMP             QCAS├── ...
        ... ──┤SPROC_ON             LMN├── ...
        ... ──┤E1                    SP├── ...
        ... ──┤E2                    PV├── ...
        ... ──┤BAL                       "P2_SPEED
        ... ──┤AUTO             AO_CH├──CONTROL"
        ... ──┤E3                      ENO├──
        ... ──┤MAN
```

APPENDIX 16

**Network 2**: Set the the LIC1.E3 to ON POSITION

Comment:

```
                                    "PUMP1
                          &         CONTROL
        "ST0".                      (LIC1)".E3
        OBJ_START_                      S
             P
```
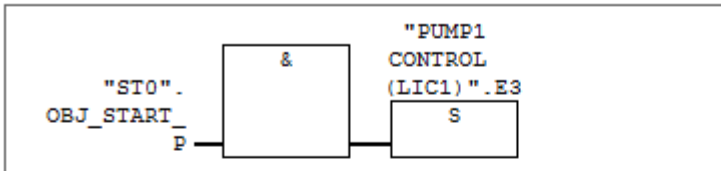
**Network 3** : Stop the pump1 - give zero speed command

Comment:

```
                        >=1
        "ST0".
        OBJ_START

        "ST3".
        OBJ_START

        "ST7".                MOVE      "PUMP1
        OBJ_START            EN         CONTROL
                                        (LIC1)".
                                    OUT EXT_REF3
        "Data".
        STOP                 IN
                                        ENO
```
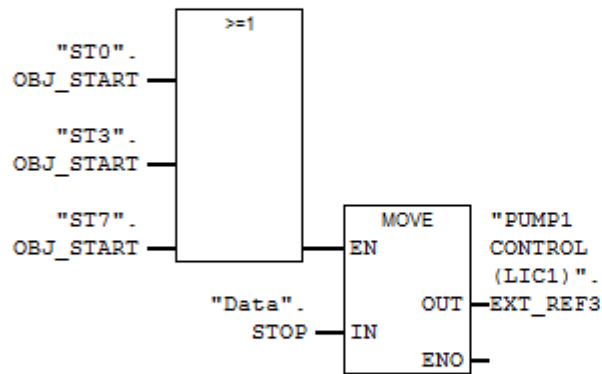
**Network 4** : Run the pump1 at washing speed

Comment:

```
                     MOVE      "PUMP1
        "ST2".                 CONTROL
        OBJ_START    EN        (LIC1)".
                           OUT EXT_REF3
        "Data".WS1   IN
                          ENO
```
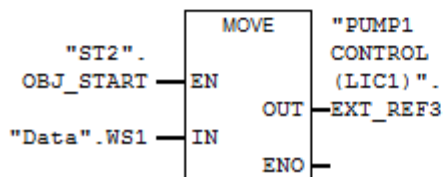
APPENDIX 17

**Network 5 : Run the pump1 to steady flow speed**

Comment:



```
                    MOVE          "PUMP1
      "ST9".                      CONTROL
   OBJ_START ──── EN              (LIC1)".
                         OUT ──── EXT_REF3
      "Data".
        SFS1 ──── IN     ENO ───
```
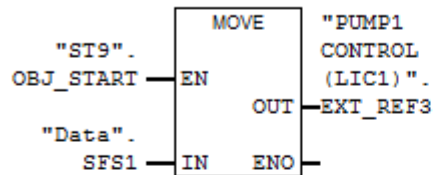
**Network 6 : Start pump1 at minimum washing speed**

Run the pump1 at minimum washing speed



```
                   >=1
      "ST5".
   OBJ_START ────
                              MOVE          "PUMP1
      "ST6".                                CONTROL
   OBJ_START ────           EN              (LIC1)".
                                   OUT ──── EXT_REF3
                 "Data".
                   MWS1 ──── IN
                            ENO ───
```
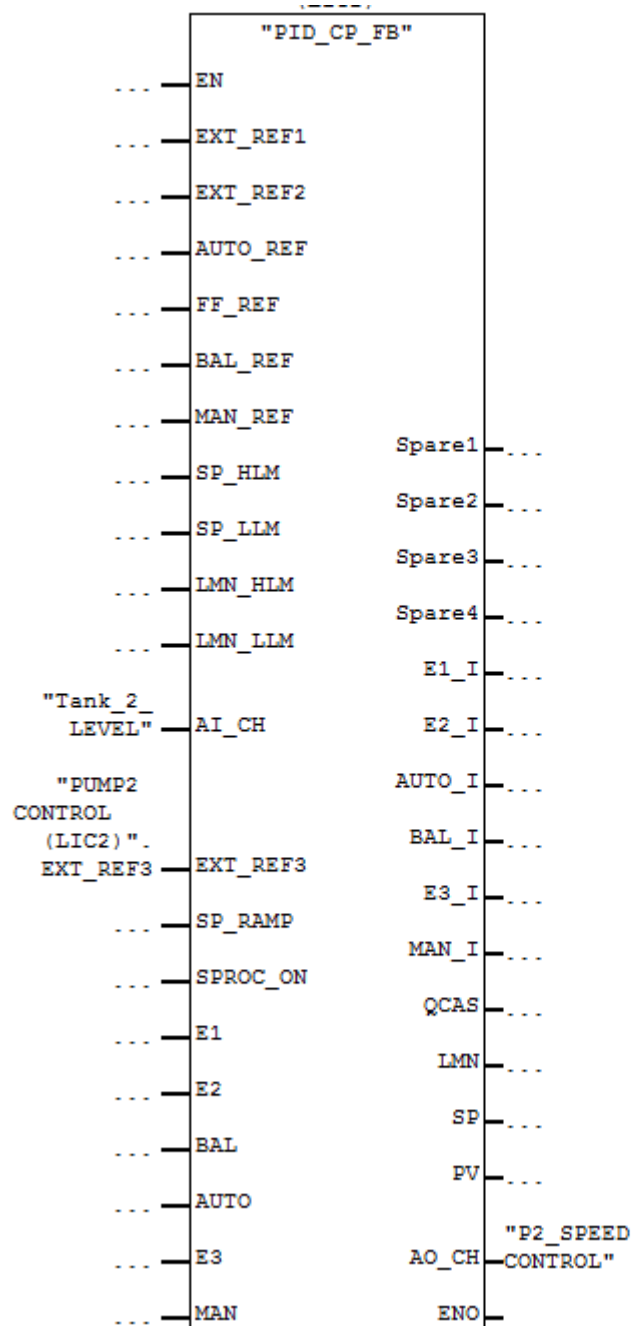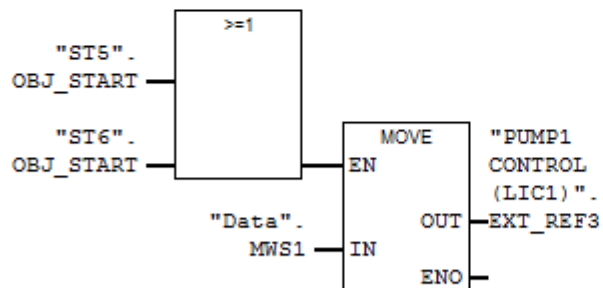
**Network 7 : Level control for tank 1 – Actuator is the pump 1**

APPENDIX 18

**Network 9 : Set the the LIC2.E3 to ON POSITION**

Comment:

```
                                      "PUMP2
                                     CONTROL
                        ┌───────┐    (LIC2)".
                        │   &   │       E3
        "ST0".          │       │    ┌───────┐
       OBJ_START_       │       │    │   S   │
              P ────────┤       ├────┤       │
                        └───────┘    └───────┘
```

**Network 10 : Stop the pump2 - give zero speed command**

Comment:

```
                   ┌───────┐
                   │  >=1  │
       "ST0".      │       │
      OBJ_START ───┤       │
                   │       │         ┌───────┐
                   │       │         │ MOVE  │   "PUMP2
       "ST7".      │       │         │       │  CONTROL
      OBJ_START ───┤       ├─────────┤ EN    │   (LIC2)".
                   └───────┘         │    OUT├── EXT_REF3
                        "Data". ─────┤ IN    │
                         STOP        │    ENO├──
                                     └───────┘
```
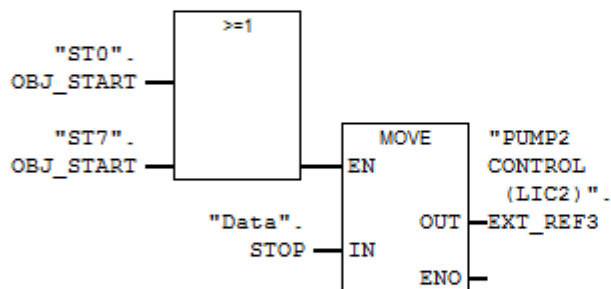
**Network 11 : Run the pump2 to washing speed**

Comment:

```
                        ┌───────┐
                        │ MOVE  │   "PUMP2
       "ST6".           │       │  CONTROL
      OBJ_START ────────┤ EN    │   (LIC2)".
                        │    OUT├── EXT_REF3
      "Data".WS2 ───────┤ IN    │
                        │    ENO├──
                        └───────┘
```

APPENDIX 19
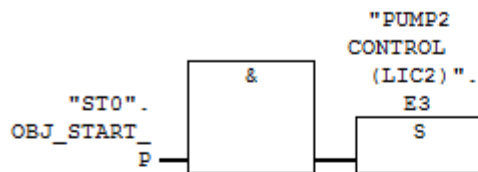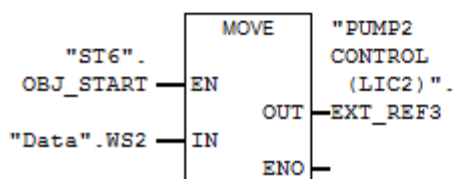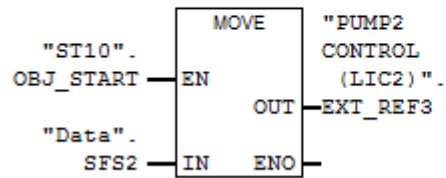
Network 12 : Run the pump2 to steady flow speed

Comment:

```
                        MOVE              "PUMP2
      "ST10".                            CONTROL
   OBJ_START —— EN                        (LIC2)".
                             OUT —— EXT_REF3
      "Data".
        SFS2 —— IN          ENO ——
```

APPENDIX 20