PLC Programming For A Water Level Control System: Design and System Implementation

by

Haoqiang Ji
B.Sc., Beijing University of Civil Engineering and Architecture, 2013

A Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF ENGINEERING

in the Department of Mechanical Engineering

PLC Programming For A Water Level Control System: Design and System Implementation

by

Haoqiang Ji
B.Sc., Beijing University of Civil Engineering and Architecture, 2013

Supervisory Committee

Dr. Yang Shi,, Supervisor
(Department of Mechanical Engineering)

Dr. Ben Nadler, Committee Member
(Department of Mechanical Engineering)

Supervisory Committee

Dr. Yang Shi,, Supervisor
(Department of Mechanical Engineering)

Dr. Ben Nadler, Committee Member
(Department of Mechanical Engineering)

ABSTRACT

In industry, the water level control problem is a typical process control problem, and has been extensively studied in the literature. This report focuses on the design and implementation of a PLC-based water level control system. In this project, we have two primary objectives: the overall mechanical design of the system, and the PLC system design and implementation. In the mechanical design part, the finite element analysis is performed for the water tank to check the area that has high leaking risk. Additionally, a flow simulation in the water tank is conducted to analyze the effect of the transient pressure on the sensors. On the other hand, the water tank is modeled in Simulink, and simulation results have shown that the PID controller can regulate the water level to the desired position. Finally, the PLC ladder diagram is programmed, and the experimental results have verified the effectiveness of the design.

Contents

List of Tables

List of Figures

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincerest thanks to my supervisor, **Dr. Yang Shi**, a decent and professional scholar, for accepting me into his research group. During my MEng study, he offered me several outstanding suggestions. Also, he gave me many tips laid the foundation of this thesis. To provide the best support, he asked the MPC Consulting Ltd. for all the experimental devices. In the period of doing my project, he always encouraged me with great patience when I suffered from difficulties. Again, let me thank my supervisor, Dr. Yang Shi, for his encouragement, patience and mentorship.

Also, I would like to thank the committee member, **Dr. Ben Nadler** , for his great suggestions.

Furthermore, I would like to acknowledge all group members in ACIPL. I enjoyed the time we were together. Thank **Jicheng Chen** for revising my thesis, suggestions on my project, and many help in my life. Thank **Bingxian Mu** for his suggestions on improving writing and presentation skill. I appreciate suggestions from all group members.

Finally, I would like to thank my parents and girlfriend for their love and support. Also, thank MPC Consulting Ltd. for supporting all the experimental devices.

Chapter 1

Introduction

1.1 Overview of Industrial Control System

Over the course of the past years, an automatic process control has emerged from intuitive especially in industry. With a rapid increasing need of industrial process control, an efficient and effective method is needed. Technological revolutions hit the industrial world in 1960s, when the development of the first programmable logic controller (PLC) is explored. A basic problem in the design of a PLC based process control system is presented. Therefore, designing a PLC based level control system attracts great interests in learning and understanding PLC based industrial control.

The automatic control refers to the representation of human control functions to technical electric equipments. The objective of the automatic control is to apply mechanisms to the operation without continuous human interactions. The automatic control has been widely used in many different areas, such as building climate control, traffic light control, process industries, and so on. Especially, with a rapid technological development in computer society, and due to the expensive labor cost and the high requirement of productivity, automatic producing process control is getting more and more important in industry.

In most industrial processes, some certain variables are of importance, and are required to regulated around some desired values. For example, the temperature in a chemical process is important to maintain a steady output. The accuracy of the process control system is very crucial because it helps to improve the productivity of the process, and prevent many irreversible problems, such as damaging equipment, safety problems or environmental issues. Nowadays, the process control system is widely used in many different areas, such as waste water treatment, oil and gas purification, chemical, pulp and paper production, and food production. The block diagram of a typical industrial control system is shown in Fig 1.1. In a process control system, the controller always plays a significant role to transact the operating conditions of the system. In most industry implementation, the Programmable Logic Controller (PLC) is used due to its advantages, such as fast response, quick and simple trouble-shoot, low cost and high reliability. As a result, a PLC based process control system is widely used in industry.

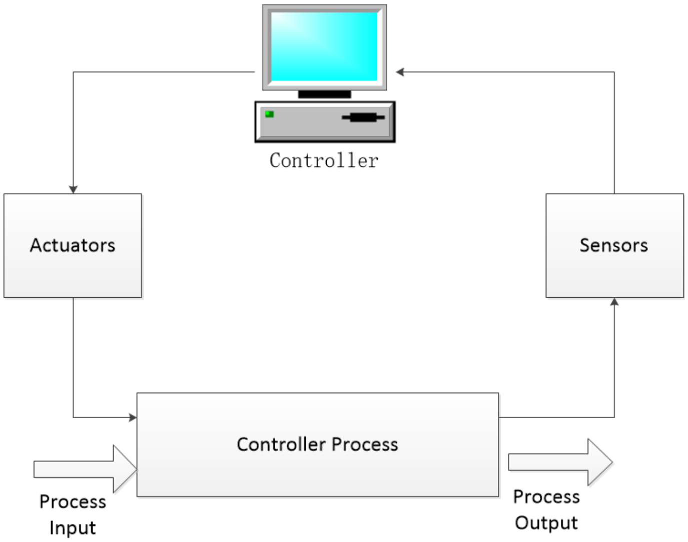


Figure 1.1: Industrial Control System

1.1.1 Overview of PLCs

Programmable Logic Controller (PLC) is a kind of computer that is specially designed for industrial control purpose [1]. It can achieve continual or discrete control goals in many different environments. Originally, PLCs are the substitutions of relays in an industrial control system in the 1960s [1]. During that period, a programmable logic controller is commonly abbreviated as PC. However, the abbreviation has been changed to PLC in order to avoid misunderstanding the programmable logic controller as a personal computer.

In the 1960s, the first PLC is designed by General Motors (GM) [6]. The first generation of PLC is shown in Fig 1.2. The GM company invites interested companies to design a device based on some specifications provided by the GM. At that time, the motivation of developing PLC is to find a chance to regenerate complex relay control systems. The device is required to be simply programmed, reliable and cheap to be maintained. In 1968, Gould Modicon Company develops the second PLC, Model 084 PLC. The primary operating principle of the first generation of PLC is based on the Boolean algebra, which means one variable only has two states(on and off). The first PLC can achieve the On-Off control only.

Based on the first generation of PLC production, more features and capabilities are developed in the second generation of PLC, and these have stimulated the applications of PLC in industry. As a result, the PLC can be used in a much more complicated process control system. Also, the PLC equips with more and more advanced functions, such as timer and counter function, memory setting and resetting, and mathematical computing operations. All these improvements give credits to the innovations and improvements in micro-controller and software programming techniques. Nowadays, the PLC consists of five common blocks as follows:

- Rack Assembly, which is used for mounting inputs and outputs terminals, power supply, and processor unit.
- Power Supply, which can provides direct current (DC) or alternating current (AC).
- Input/Output Section, which are used for connecting external terminals for devices to the PLC.
- Central Processing Unit (CPU), which controls all operations of the system.
- Programming Device, which is used for programming the PLC.

Another feature which is crucial for a PLC is that the PLC can deal with analog signal directly. In this way, the design of a process control system can be simplified. At the end of the seventies, analog inputs and outputs were expanded on a PLC together with the original digital inputs and outputs [1]. The representative product of the second generation of PLC is shown in Fig 1.3.

The real-time data can be monitored and managed efficiently by using the analog inputs and outputs in PLCs. Therefore, a PLC based control system can increase the system response speed, and readily detect faults. Furthermore, the cost of implementing a PLC is low, and the designed system is realiable. In general, the physical components, such as switches, lambs, pumps, valves, and so on, can be connected to the external terminal of the PLC. Then, all connected physical components can be controlled via PLC programming. In other words, the program components are the representations of mechanical components in PLC programming so that one mechanical component can be used for two or more control purposes. In industry, a system can consist of thousands switches or lamps, so reducing the number of physical components can decrease the total cost of the system. Also, every component has its service life. However, the components can be represented by the program components, so a PLC based control system has a high reliability. Finally, because the PLC matches most requirements of an industrial process control system, it has become commonly used in industry.

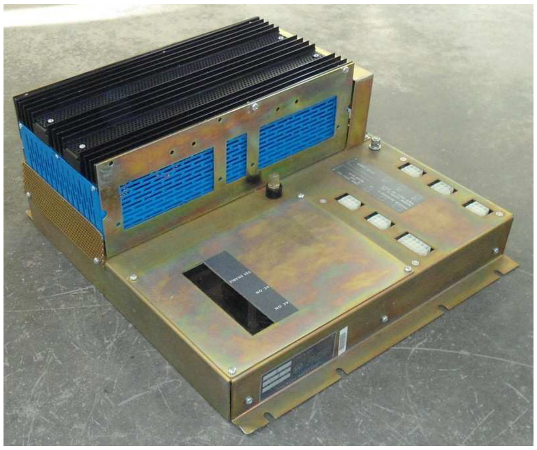


Figure 1.2: The first programmable logic controller of Allen Bradley, the Bulletin 1774 PLC. It was invented by Ernst Dummermuth in 1974 [1].

Figure 1.3: The Modicon 184, second generation of PLC designed by Modicon [1]

Currently, there are many PLC manufacturers in the market. Table 1.1 shows a list of PLC manufacturers in the global market. They produce many types of PLCs depending on size, cost, and functions to satisfy different requirements of different system specifications. A static of a survey of PLC manufacturers is shown in Fig 1.4. From Fig 1.4, we can see that the Schneider Electric leads the PLC market, with 45% North America PLC Market Share.

No	Manufactourer	No	Manufactourer
1	Siemens	9	Panasonic
2	ABB	10	Idec
3	Schneider	11	Keyence
4	Rockwell	12	Toshiba
5	Mitsubishi	13	Fuji
6	GE-Fanuc	14	Beckhoff
7	Omron	15	Bosch Reroth
8	Koyo		

Table 1.1: List of PLC

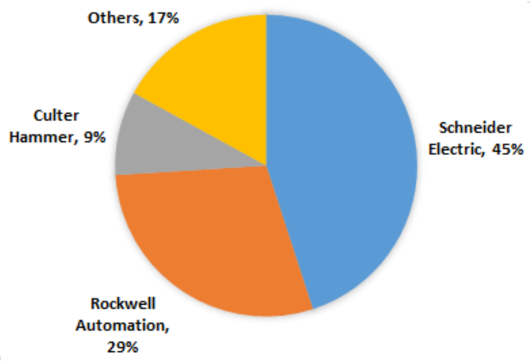


Figure 1.4: North America PLC Market Share [2]

1.1.2 PLC Program Devices

Three main programming devices can be found in the global market. They are Hand-Held programming device, simple display-keyboard programming device and laptop computer. Fig 1.5 shows a hands-free programming device. The programming language used in this device is called Boolean. The display screen can only show one line of the program. Simple display-keyboard programming device, which is shown in Fig 1.6, consists of a screen and a keyboard. The function of this type of programmer is similar to a Human Machine Interface (HMI). Also, a laptop computer can be used as a programming apparatus. However, it requires a specific programming software to program the PLC.

Figure 1.5: Typical Hand-Held Programming Device [1]

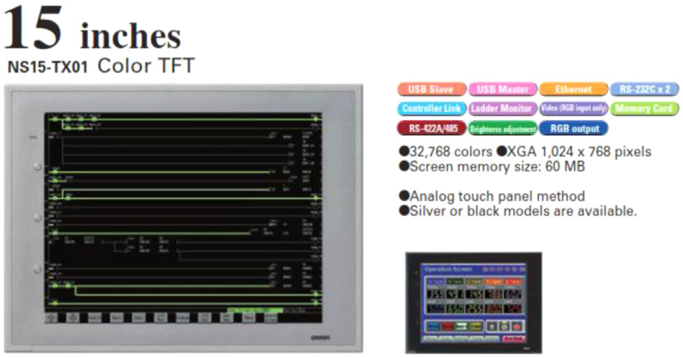


Figure 1.6: Omron Programming Device [1]

1.1.3 PLC Programming Language

The PLC accept many different programming languages, such as Ladder Diagram (LD), Sequential Function Charts (SFC), Function Block Diagram (FBD), Structure Text (ST), and Instruction List (IL) [7]. These languages are supported by standard IEC 61131-3 [7]. Some softwares are used for programming some specific editions of PLC. For example, Telepace Studio accepts LD and FBD, and Unity Pro accepts all the programming languages. Commonly, the ladder diagram and function block diagram are the two widely used programming methods in the PLC.

The ladder diagram resembles a ladder with two vertical rails and many rungs to program. In a ladder diagram, all the components must be connected to both ladder rails, since the two vertical rails represent the power supply. The current flows from the left of the ladder diagram to the right. An example of a ladder diagram is shown in Fig 1.7. From the Fig 1.7, it can be seen that the ladder diagram is similar to the traditional electrical circuit diagram. The symbols used in a ladder digram have their real physical functions.

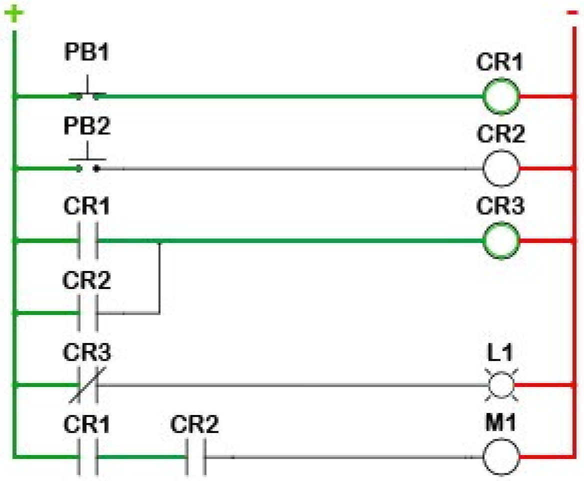


Figure 1.7: An Example of Ladder Diagram

Similar to the ladder diagram, a Function Block Diagram is also a graphical programming language. However, the blocks used in the function block diagram do not have physical functions. Instead, a function block describes a relationship or a function between the inputs and outputs. To program a function block diagram, some basic knowledge of logic functions is required, such as AND, OR, NAND, and NOR. The Fig 1.8 shows the AND in both ladder diagram (Left 1.8a) and Function block diagram (Right 1.8b).

There are three main components in a processor: the CPU, the arithmetic logic unit (ALU), and the memory. The ALU is to execute mathematical calculations and logic functions. Then, the results are stored in the memory. The memory is classified as ROM (Read-only-memory) and RAM (Random-access-memory). RAM is powered by a battery, so when the battery is removed or dead, the information stored in this memory gets lost. As a result, it is necessary to check the RAM battery voltage at the beginning of a PLC program.

(a) AND Function in Ladder Diagram (b) AND Function in Function Block Diagram

Figure 1.8: Comparison of LD and FBD

1.1.4 Communication Protocols

The processor is a single microprocessor which is able to control various kinds of industrial equipment by an acceptable PLC program. In PLCs, the processor chip is connected to the memory and input/output port. They exchanges data between each other by parallel address and control buses. It's desirable to consider the communication setup in the PLC.

In 1979, Modicon (now is Schneider Electric) develops a serial communication protocol, and names it as Modbus [8]. This communication protocol has been widely used in PLC industry because of four reasons: First of all, Modbus is intensively designed for PLC industrial applications: then, Modbus is an open source and a free copyright protocol; moreover, it can be easily set up and maintained; finally, Modbus can transfer datas without any limitations. To meet different requirements of different devices, the Modbus protocol has evolved with time [9]. The most commonly used communication protocols used with PLCs are Modbus RTU, Modbus ASCII, Modbus TCP, Modbus TCP/IP, and Modbus USB.

1.2 Literature Review

1.2.1 PLC Applications

In modern industry, PLC control systems have been extensively implemented in many applications, such as water and wastewater treatment control system, Sun-tracking system, wind energy system, photo-voltaic applications, heating ventilating and air-conditioning (HVAC) control, manufacturing and so on. One common feature of these applications is that they can be modelled as process control problems. For example in the water and wastewater treatment control system, the pumps and valves are controlled according to the real time data of the process. To be more specific, water level control problem is a typical process control problem. Therefore, in this section, some related literature is reviewed and discussed.

Yuriy et al. (2015) [10] designed a PLC based system to control liquid level by using Radar sensor remotely. This system measures the liquid level, volume, temperature, and pressure and control these measurements remotely. The system consists of the Radar sensor, temperature sensors, discrete level sensors and a programmable logic controller. The PLC is programmed by the FBD programming language. To transfer datas between all the components, the communication method that the authors choose is RS 485 bus with an ASCII based protocol. The authors state that communication protocol actually reduces or eliminates interference during the process of data transmission. In order to monitor the system, the authors also design a human machine interface (HMI). Finally, the experimental result corroborates the exactitude and reliability of their system.

Pooja et al. (2015) [11] carried a test on a PLC based single water tank control system using PID controller. In their system, an HMI which is programmed on NI-LabVIEW is connected to an Allen Bradley Micro830 PLC through the Modbus RTU communication protocol. According to Pooja et al., this system is designed for training purpose in order to have a complete understanding of PLC based process control system design. In their literature, some necessary modelings are introduced, such as the water tank modeling, transducer modeling, and the control valve modeling. Some parameters, such as the resistance of the control valve, and current to pressure (I/P) converter, are estimated depending on the experimental data by using the method of least squares. Furthermore, Pooja et al. applied PID algorithm into the PLC to achieve a better result. The PID parameters are calculated by using Ziegler Nicholas (Z-N) method. Finally, the authors conclude that the experimental result is matched with their prediction.

Mini and Shilpa (2016) [12] did a preliminary test on a liquid mixing and bottle filling system. This system is controlled by a PLC. In their literature, the system simulation based on Matlab/Simulink and PLC ladder diagram programming are included. There are three subsystems in the ladder diagram for mixing, filling, and level monitoring respectively. Mini and Shilpa described that three water tanks, several level sensors and a DC motor consist of the system. They explain that two different kinds of liquid are stored in two tanks separately. Two pumps controlled by the PLC pump preset amount of the two types of liquid into an overhead tank for mixing. Then, the DC motor drives a conveyor belt carrying bottles for filling. To control the amount of filling, a control valve controlled by the same PLC is applied to the overhead tank. In order to avoid overflow of the mixed liquid stored in the overhead tank, three level sensors are used to monitor the liquid level. The sensors are used for tracking low, high and emergency level respectively. Furthermore, Mini and Shilpa used a PID controller as the algorithm of the system to achieve a better controlled result. As a result of using PLC combining with PID controller to control the system, Mini and Shilpa summarized that PLC based control system could reduce human intervention and operating cost. At the same time, it can increase producing efficiency and driving safety.

Da'na et al. (2008) [13] developed a networked platform which can be used for remote monitoring and controlling PLCs. The reason of that, Da'na et al. mentioned, is that the computing ability of PCs is increasing and the available options of network protocols and standards are various. In their system, a Simatic S7 200/300 PLC is applied. The available communication methods of the PLC are LAN, WAN, or GSM. To achieve the final experimental goal, the key parts of their system are the Transmission Control Protocol / Internet Protocol (TCP/IP) setup and Global System for Mobile Communications (GSM).

In order to train engineering students on PID control processes, Dilsad and Mustrafa (2013) [14] designed a level control system using PID control algorithm on a programmable logic controller. In their literature, a description of PID parameters tuning is introduced. Also, based on the PID parameters, a ladder diagram design of level control system is explained. The feedback sensor used in this system is a level transmitter. Finally, as the experimental result shown in their paper, the PLC based PID level control system is stable.

Furthermore, Raza et al. (2011) [15] developed a PLC based PID control system for a heating tank control system. In their paper, Reza et al. applied an HMI into the system for advanced monitor and control purpose. Finally, the result shows that the PLC based PID control system works correctly in their design.

1.3 Motivation and Outline of the Project

As introduced in the previous sections, we can conclude that the PLC has been widely used in the process industry, and the applications will not stop growing. The motivation of the project is to design and implement the PLC in a water level control system. Since the water level control problem is a typical control problem in the process industry, it's meaningful to conduct such an experiment to simulate the common control problem. The organization of the project is illustrated in Fig 1.9.

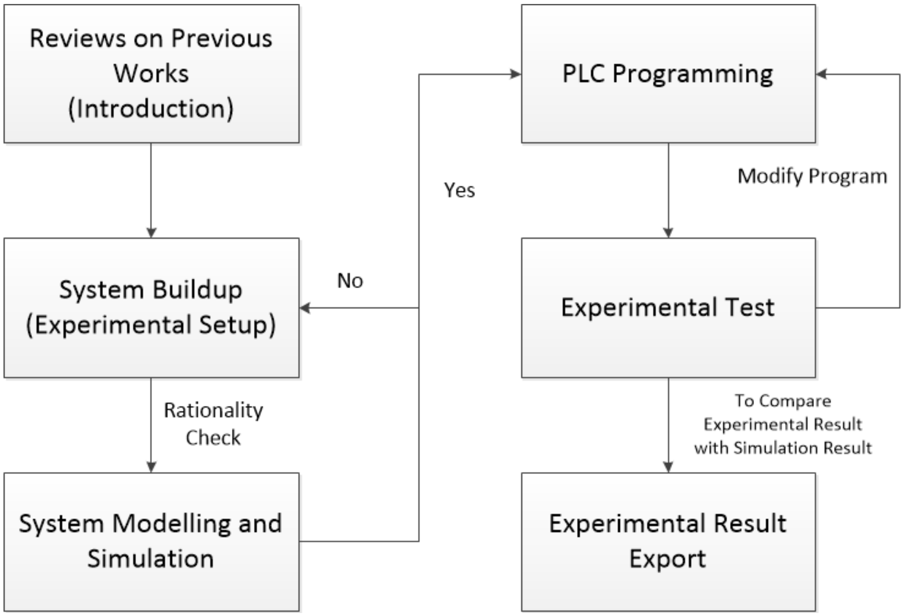


Figure 1.9: Process of Designing

Chapter 1 gives the background knowledge of the PLC, which includes the development of PLC, PLC program devices review and PLC program languages. Also, a literature review on PLC control system design is given. Chapter 2 describes the system setup, including dimensions of system frame and specifications of system components. Furthermore, it contains a finite element analysis and flow simulation analysis on the designed water tank. Chapter 3 is the experimental study to apply a PID algorithm to the PLC based water level control system. The main results of the project is introduced In Chapter 3. The simulation and experimental results are presented to demonstrated the effectiveness of the implementation. Finally, the report concludes with some conclusions and further work in Chapter 4.

Chapter 2

Experimental Setup

2.1 System Overview

A typical control system consists four components: the plant, actuator, sensor and controller. Sensors measure the state of plants, and transmit the data to controllers. Then, controllers will calculate the control input to the plant, and transfer the control input to actuators. In this way, the feedback mechanism is introduced to the system, and the desired control objective can be obtained. The objective of the water level control system is to control the water level of the tank, and track the desired reference water level. This is a typical control system in the process industry, and the monitors, feedback sensors, and actuators, are closely related to each other.

In this project, the focus is on the PLC system design and implementation. In this PLC system, the controller is the SCADAPack 350 PLC, which is manufactured by Schneider Electric. A control valve, Fisher 3661 Positioner, is characterized as the actuator in this PLC system. A pressure sensor, WIKA Pressure Transmitter, and a flowrate totalizator, SMT-101, are used to measure states of the system. Furthermore, a water tank and some pipes are designed for storing and draining water. The specifications of each component is detailed in Section 2.3.

The layout of the system is illustrated in Fig 2.1. The system frame is created based on a $36'' \times 19'' \times 34''$ $(L \times W \times H)$ cart. The frame is $36'' \times 19'' \times 41''$ which has enough space for the water tank $(4'' \times 31.5'', D \times H)$, pressure sensor, flowrate totalizator, and control valve. The SCADAPack 350 is placed in a sealed steel box located at the bottom layer of the cart. The flowrate totalizator follows the water tank with the pressure sensor, and the last component is the control valve. They are connected by plastic transparent hoses. All connections are straightly placed on a horizontal plane in order to make sure water is unhindered. The purpose of this layout is to reduce experimental error because gravity drives water in this system without any other external force or pressure.

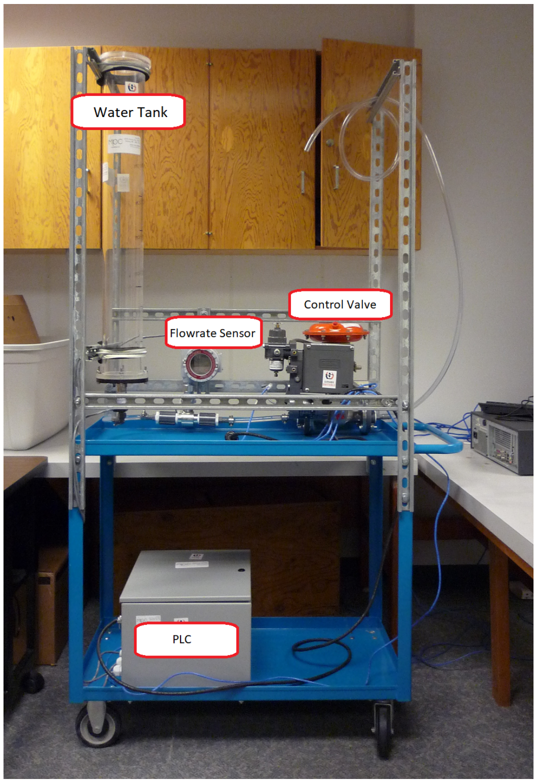


Figure 2.1: Water Level Control System Overview

2.2 Mechanical Design

The design of the water tank should be carefully checked to ensure safety for both users and components. A number of factors such as energy flow and distribution needs to be taken into consideration before making the appropriate decision. In this project, the energy carrier is water, so it is desirable to investigate the pressure distribution, the water tank displacement and flow behaviour when the system is operating. Furthermore, the energy distribution analysis is necessary for the selection of water tank seals or gaskets, water flow pipes, and control methods.

This section consists of finite element analysis (FEA) and flow simulation of the water tank based on SolidWorks. The FEA result provides pressure distribution which gives a range of the water tank gasket and material selection. Moreover, according to the FEA result, we can figure out the safe pressure limitation of the water tank. The Flow Simulation result provides a water flow circumstance. This result contributes a basis that can be used for physical optimization and adjusting parameters of a control algorithm.

2.2.1 Finite Element Analysis of the Water Tank

The purpose of FEA is to model products and systems in an actual environment. The result can be used to find hidden structure problems, and figure out solutions to the problems. This step is an important and fundamental milestone in a real industrial process design. In this project, the object of FEA is the water tank. A nonuniform force which stands for water is applied to the water tank. The bottom of the water tank is set as the fixed part. To get a pragmatic result, the mesh size should be as much fine as possible but this will result in a the longer simulating time. Therefore, the mesh size for this project is set automatically which is an average level. A process of setting up the FEA simulation is shown in Fig 2.2 and the results of FEA are shown in Fig 2.3. In FEA, the material assigned to the water tank is PVC 0.007 Plasticized which is similar to the water tank material. The specification of PVC 0.007 Plasticized is shown in Table 2.1.

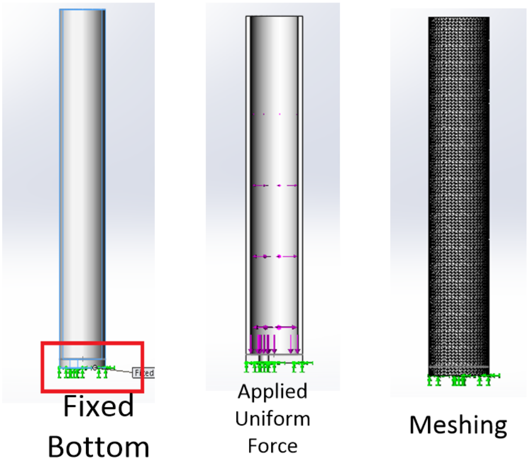
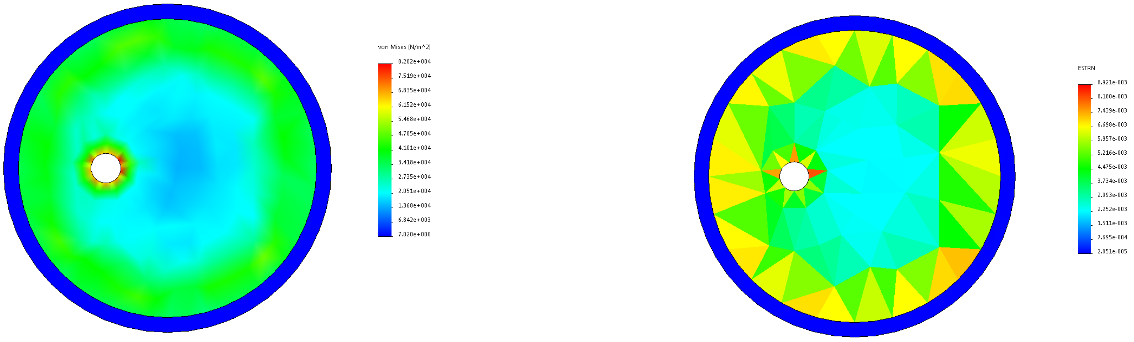


Figure 2.2: FEA Process on SolidWorks

Fig 2.3a shows the stress distribution of the water tank when the water level reaches the maximum value, 80 cm. The color is changing from blue to red which means stress is increasing. From the overview stress distribution at left of Fig 2.3a, it

Property	Value	Unit
Elastic Modulus	6	N/mm^2
Poisson's Ratio	0.47	N/A
Shear Modulus	2	N/mm^2
Mass Density	1290	kg/m^3
Tensile Strength	13	N/mm^2

Table 2.1: Specification of PVC 0.007 Plasticized



(a) Stress of Water Tank

(b) Strain of Water Tank

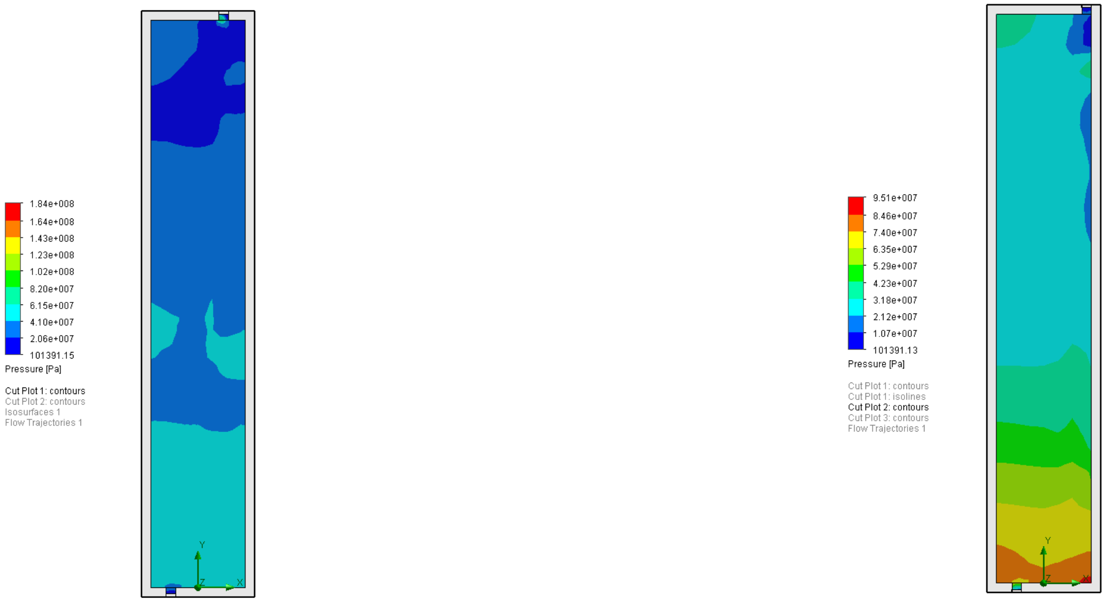
Figure 2.3: Results of Finite Element Analysis

shows clearly that the stress distributed on the water tank wall is in blue color which is safe for this project. However, the stress is increasing from the edge of the bottom to the draining hole and getting red at the edge of draining hole. After checking the stress at the edge of draining hole and comparing the maximum stress with the maximum stress limit of the material, this plastic material is eligible enough for this project.

Fig 2.3b shows the strain distribution of the water tank. In this figure, the strain distribution is similar to the stress. However, at the bottom of the water tank, the maximum strain occurs at both the edge of the water tank and the edge of draining hole. From this result obtained so far, it seems that the water tank has a high risk of leaking at the edge when pressure is getting larger. In this case, a rubber gasket is introduced to this water tank to avoid small leaking when water is filled into the water tank because of its contraction principle.

2.2.2 Flow Simulation

The FEA result provides a description of water tank material and seals selection. However, the positions of draining and filling holes are also important to the system because they directly influent the efficiency, accuracy, and safety of the water tank system. In this section, two flow simulation results are conducted. The results focus on the flow pressure distribution based on two different filling hole positions. In the first situation, the filling hole is placed near the center of the top of water tank. In another situation, the draining hole is placed near by the edge of the top of water tank. A comparison of front plane cut plot of the water tank flow pressure distribution is shown in Fig 2.4 and a comparison of flow trajectory is shown in Fig 2.5.

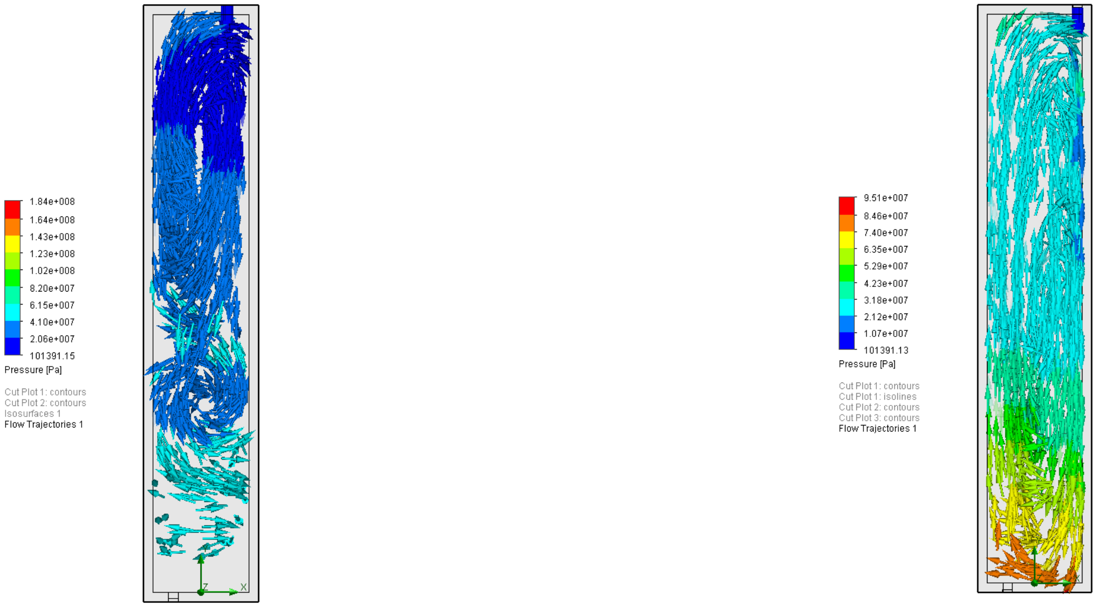


(a) Filling Hole Near Center

(b) Filling Hole Near Edge

Figure 2.4: Comparison of Flow Pressure Distribution

In the above figures, the color represents the flow pressure. When the flow pressure gets larger, the color gets brighter. From the Fig 2.4, we can see that with the same inflow, the flow pressure of Fig 2.4a is smaller than the Fig 2.4b. From the Fig 2.5, we can see that the flow of Fig 2.5a is slower than the Fig 2.5b. As we know that the pressure sensor is installed at the bottom of the water tank, so a smaller flow pressure and a stable flow at the bottom are important to increase the accuracy of the pressure sensor. As a result, the filling should be placed away from the edge of



(a) Filling Hole Near Center

(b) Filling Hole Near Edge

Figure 2.5: Comparison of Flow Trajectory

the water tank to reduce the pressure sensor errors.

2.3 PLC and Sensors

2.3.1 SCADAPack 350

SCADAPack 350 Overview

In this project, we select the SCADAPack 350, as shown in Fig 2.6, as the PLC micro controller. It is programmed in a software called Telepace Studio which is provided by Schneider Electric as well. The SCADAPack 350 is a low power Remote Terminal Unit (RTU) controller built up with a 5209 controller board [3]. It consists of an integrated power supply, analog and digital I/O, serial communications, 10/100 Mb/s Ethernet, 12Mb/s USB A and USB B ports and turbine flow meter counter inputs. Programming languages used for SCADAPack 350 are Relay Ladder Logic, IEC 61131-3 and the C language. Moreover, the expandable input and output (I/O) quantity of SCADAPack 350 can be adjusted by using up to a maximum of twenty 5000 I/O modules. A 5606 Integrated I/O module can operate with this 5209 controller board. Furthermore, the SCADAPack 350, which is a powerful economic controller, comprises several power saving modes, such as:

- Sleep Mode
- 24-power shutdown
- Ethenet port shutdown
- Communication port power control
- SCADAPack Vision power down
- USB disable and a reduced power mode

Figure 2.6: SCADAPack 350 Overview [3]

A board layout of SCADAPack 350 is shown in Fig 2.7. The SCADAPack 350 is powered on by a range of voltage from 11V DC to 30V DC. The input power is connected to positive and negative terminals respectively, which is located at connector P3 in Fig 2.7. Moreover, the details of the SCADAPack 350 are introduced in the sequel.

Digital and Analog I/O Port

There are eight analog inputs, with a range of 0-10V or 0-40mA, on the SCADAPack 350. According to the SCADAPACK 350 Hardware Manual [3], six of eight analog inputs are external inputs which allow other components to transfer data into the controller, and the other two are internal inputs which are used for monitoring RAM battery voltage and controller board temperature. Each external input can resolute the input into 15-bit. The first fifth external inputs, from number 0 to 4, have both

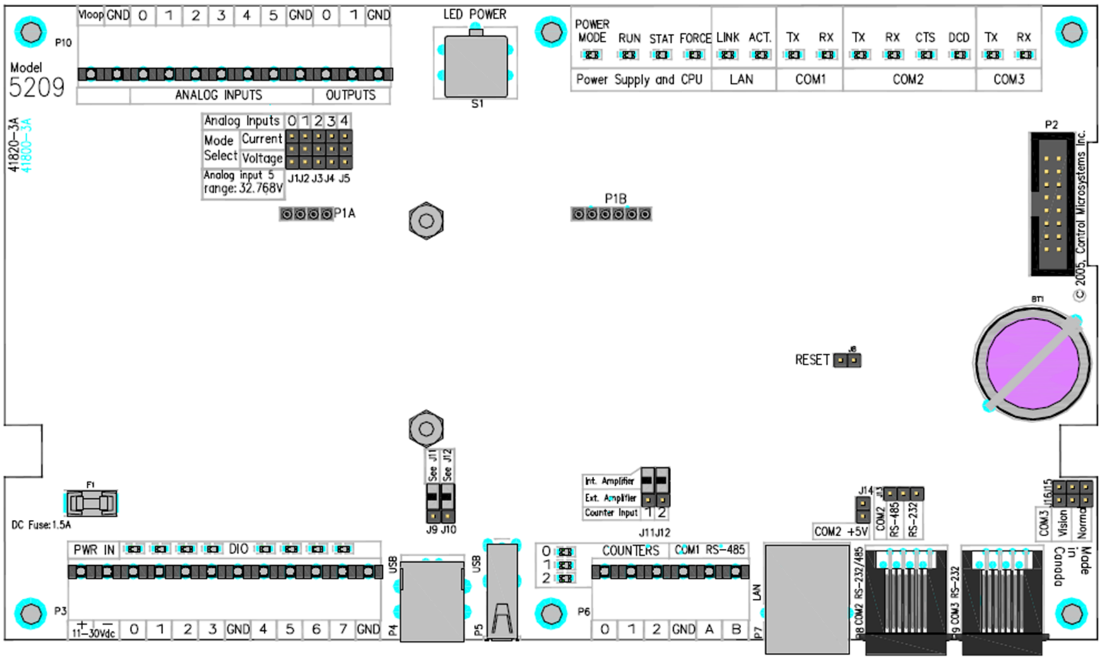


Figure 2.7: Board Layout of SCADAPack 350 [3]

voltage and current mode. The sixth input, or pin number 5, is used under voltage mode only, and its measurement range is from 0 to 32.768V [3]. The internal inputs can be designed based on a situation of a different project to measure input voltage, controller temperature, and battery voltage respectively.

The SCADAPack 350 can represent data in different ways under different modes. In current mode, 20mA is represented by a 14-bit data. As a result, the input resolution is 1.22 uA/count. To manage the range of input values, designers can select different registers managements. There are two kinds of registers managements: SCADAPack 350 10V/40mA I/O and SCADAPack 350 5V/20mA I/O [16]. The specification of these two kinds of registers is shown in Table 2.2.

There are two analog outputs on the SCADAPack 350. Similar to analog inputs, both analog outputs have current and voltage mode. However, the maximum current value is 20mA in current mode, and in voltage mode, an external load register should be connected to the current output channels in series in order to obtain a voltage value. The analog output data form is shown in Table 2.3.

Furthermore, the SCADAPack 350 is equipped with eight universal digital inputs and outputs. Each digital inputs and outputs channel has its corresponding LED indicator for service or troubleshooting. These eight channels can be defined as in-

Feature	**SCADAPack 350 5V/20mA I/O**		**SCADAPack 350 10V/40mA I/O**	
Measure Range	0-5V or 0-20mA		0-10V or 0-40mA	
Resolution	14-bit Value			
Data Form	**Channel 0-4**	**A/D Output Value**	**Channel 0-4**	**A/D Output Value**
	0mA	0	0mA	0
	1.22uA	1	1.22uA	1
	4mA	6552	4mA	3277
	10mA	16384	10mA	8192
	20mA	32767	20mA	16384
	39.999mA	NA	39.999mA	32767

Table 2.2: Specification of Registers and Data Form

Data	Current	Resistance	Voltage
0	0mA	250 Ohms	0V
		500 Ohms	0V
8	4.88uA	250 Ohms	1.22mV
		500 Ohms	2.44mV
6552	4mA	250 Ohms	1V
		500 Ohms	2V
16384	10mA	250 Ohms	2.5V
		500 Ohms	5V
24576	15mA	250 Ohms	3.75V
		500 Ohms	7.5V
32760	20mA	250 Ohms	5V
		500 Ohms	10V

Table 2.3: Analog Output Data Form

puts or outputs based on design requirements. The inputs are signed when they are attached with dry connects, such as switches and relay contacts. Similarly, the outputs are defined by addressing them in a program. Based on the programming principle of Telepace, the outputs are signed to the addresses from 1 to 12. The addresses of digital inputs in the SCADAPack 350 are from 10001 to 10013 which is corresponding to channel 0 to 12. Among these 13 channels, the first seven inputs or outputs are external channels located on terminal P3. They are controlled by giving value 0 or 1 to turn on or off. From channel 8 to 12, the channels are internal, which are used for turn on or off VLOOP output status, DC/DC converter status, VLOOP over-current status, digital output mismatch, and COM3 power for Human Machine

Interface (HMI) respectively.

Communication Port

Serial Communication Serial communication is going to send real time data over a communication channel or computer bus [17]. There are three serial communication ports build up on the SCADAPack 350. They authorize RS-232 and RS-485 communication. These serial communication ports are named COM1, COM2, and COM3. Referring to Fig 2.7 SCADAPack 350 Board Layout, the location of them is near by terminal P6. COM1 is a dedicated RS-485 port and COM3 is a dedicated RS-232 port. However, COM2 can be defined as either RS-232 or RS-485. The comparison of RS-232 and RS-485 is shown in Table 2.4

	RS-232	**RS-485**
Baud Rate	300, 600, 1200, 2400, 4800,9600,19200, 38400,57600,115200 Default: 9600	300, 600, 1200, 2400, 4800,9600,19200,38400, 57600, 115200 Default: 9600
Duplex	Full or Half Default: Half	Half Default:Half
Parity	Odd, None or Even Default: None	Odd, None or Even Default: None
Data Bits	7 or 8 Bits Default: 8 Bits	7 or 8 Bits Default: 8 Bits
Stop Bits	1 Bit	1 Bit
Receive Flow Control	ModbusRTU or None Default: ModbusRTU	None or Xon/Xoff Default: None
Transmit Flow Control	Ignore CTS or None Default: None	None or Xon/Xoff Default: None
Station	1 to 65534 Default: 1	1 to 65534 Default: 1
Protocol	None, Modbus RTU, Modbus ASCII, DF1 and DNP Default: Modbus RTU	None, Modbus RTU, Modbus ASCII, DF1 and DNP Default: Modbus RTU
Addressing Mode	Standard or Extended Default: Standard	Standard or Extended Default: Standard

Table 2.4: Comparison of RS-232 and RS-485

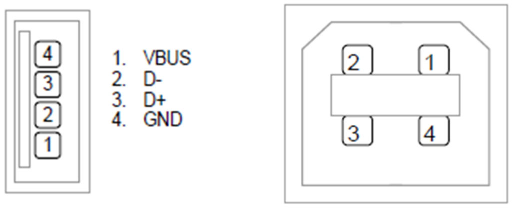
Ethernet Communication Ethernet communication is a kind of computer networking technologies widely used in local area network (LAN) [18]. One 10/100 Base-T Ethernet port is applied on the SCADAPack 350. The speed of data transmission through the Ethernet port is 10/100 Mb/s. The supported operating modes can be selected as half-duplex and full-duplex. To make connections to the LAN port, an RJ-45 modular connector is required. After connecting the SCADAPack 350 to a computer through LAN port, TCP/IP parameters, such as the address of controller, subnet mask, and gateway, should be set from the Telepace. The values of parameters are shown in Table 2.5:

Parameter	**Value**
IP Address	In the format 255.255.255.255 Default: 0.0.0.0
Subnet Mask	In the format 255.255.255.255 Default: 255.255.0.0
Gateway	In the format 255.255.255.255 Default: 0.0.0.0

Table 2.5: TCP/IC Parameter

- IP Address: the address of the controller
- Subnet Mask: the subnet on which the controller is located
- Gateway: the determination of how the controller communicates with devices outside its subnet

USB Port Universal Serial Bus (USB) is used for a connection, communication, and power supply between computers and electric devices. The SCADAPack 350 has two 2 USB ports. They provide both low speed (1.5Mb/s) and full speed(2.0 Mb/s) for communicating. One of these two USB ports is a host USB. The host USB port can also be used for extending the memory of the SCADAPack 350. An external flash USB can be connected to save operating data when the on-board memory is full. The other one, called peripheral port, can be used to connect the SCADAPack to a host computer. Different to Ethernet port, the protocol of USB port is Modbus USB in Telepace. The connections are shown in Fig 2.8.



Host USB

Peripheral USB

Figure 2.8: Diagrams of USB Ports [3]

LED Indicators

The last part of SCADAPack 350 board is the LED indicators section, which is located at the right top corner of Fig 2.7 SCADAPack 350 Board Layout. The purpose of LED indicators is to monitor the operating status of the SCADAPack 350. The LEDs are used for monitoring power supply and CPU, Ethernet communication, and serial communication. Each LED is labeled as its monitoring object. The description of LEDs is shown in Table 2.6:

	LED	**Function**
Power Supply and CPU	**Power Mode**	On when operating and the LEDs are enabled. Off when the LEDs are disabled. Off when powered off or in Sleep Mode.
	RUN	On when the ladder logic program is executing.
	STAT	Blinking when a controller condition exists.
	FORCE	On when I/O points are forced.
	USB STAT	Not currently used.
LAN	**LINK**	On when the LAN port has established a link
	ACT.	On to signal activity on the LAN port
COM1, COM2, and COM3	**RX**	On when receiving data on the corresponding serial port.
	TX	On when transmitting data on the corresponding serial port.
	CTS	On when the CTS input is asserted COM2.
	DCD	On when the DCD input is asserted COM2.
	Digital I/O	On when the corresponding I/O point is on. LEDs are dim in Sleep Mode when the corresponding I/O point is on.

Table 2.6: Description of LEDs

2.3.2 Pressure Sensor

The pressure sensor used in this project is a WIKA C-10 Pressure Transmitter. There are two kinds of C-10 sensors on the market. They are mostly the same except with different connectors. The one that is used in this project is a C-10 with MiniDIN connector. Fig 2.9 shows the WIKA C-10 pressure transmitter. The WIKA C-10 is a performance and economical sensor for many applications and used in a variety of areas, such as hydraulics, pneumatics, mechanical engineering, and general industrial applications [4]. Its performance and economy are reflected in its reliable performance.

Figure 2.9: WIKA C-10 Pressure Transmitter [4]

Material Measuring cell used in the C-10 is an all-welded stainless steel to increase media compatibility [4]. There are not internal soft sealing materials inside of the C-10 because an inner soft material may increase the risk of failure over time.

Input and Output Signals The input of this C-10 pressure sensor is the pressure on a piezoresistive measuring cell [4]. Thin film sensor technology is used on a higher pressure range to make the sensor reliable. Its pressure range is up to 300 psi. In this project, the C-10 is applied to test a water pressure, and the water pressure is related to the height of the water in the water tank. Then, the C-10 transmits the pressure or the height based on an equation to an analog output signal. The output is from 0 to 20 mA that can be integrated into the SCADAPack 350.

Accuracy To achieve an accuracy of $\leq$ 0.50 %, the C-10 has undergone lots of quality control testing and calibration [4]. To obtain perennial accuracy and stability, each C-10 can compensate temperature individually when operating in different temperature conditions.

Equation of Calculating Height Based on a model of the water tank, the pressure on the C-10 can be calculated as $\rho * g * h$. As knowing of ρ and g are constant parameters of density and gravity acceleration, the height h is corresponding to the pressure straightly. According to the output signal range, 4-20 mA, it can be known that when the pressure and height is 0, the output is 4 mA, and when the height or pressure is increasing, the output current is increasing linearly. Therefore, to know the actual height, it is necessary to find out a mathematic relationship between the height and the output value. A least square method is used to find the equation. The result is shown in Fig 2.10. The line of best fit for the experimental datasets is generated on Microsoft Excel based on the reading values of the pressure sensor. The visual demonstration is provided as Equation 2.1, where the height is x axle and the analog input value is y axle.

$$y = 101.27x + 6248.3 \tag{2.1}$$

Based on the Equation 2.1, the water height can be estimated in SCADAPack 350 data form. Therefore, a desired height can be easily set in the program.

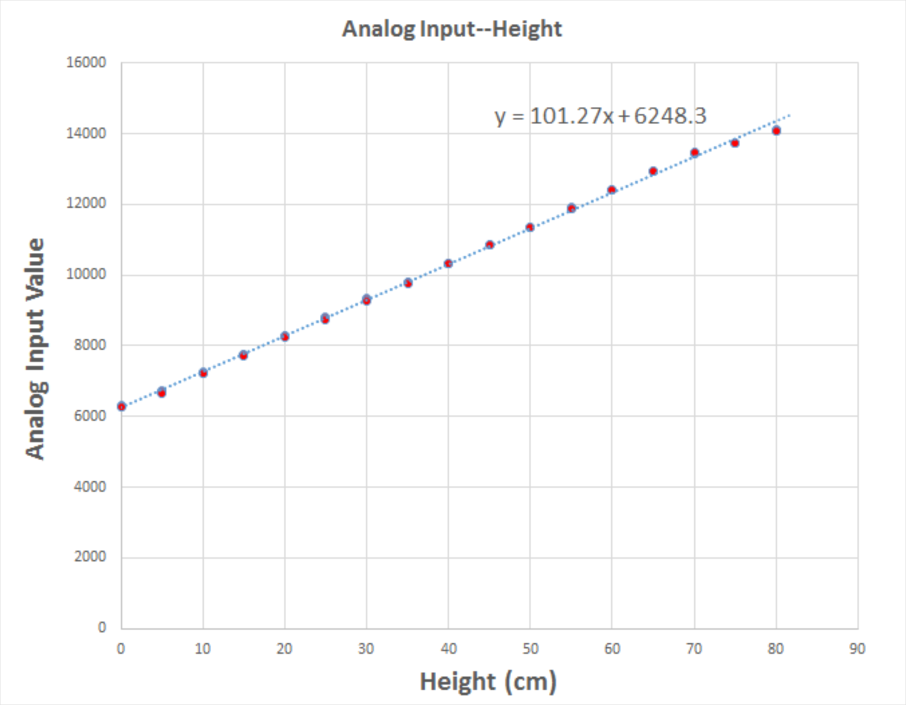


Figure 2.10: Relationship Between Analog Input and Height

2.3.3 Control Valve

A positioner is an actuator that is applied to open or close a valve. The positioner is operated by changing the air pressure to drive an actuator of the control valve. The air pressure is controlled by the output signal of the SCADAPack 350. In this project, the control valve used is a Fisher 3661 Positioner. An overview of the Fisher 3661 is shown in Fig 2.11.

Figure 2.11: Fisher 3661 Positioner [5]

This Fisher 3661 positioner is an electro-pneumatic single-acting positioner [5]. It can cooperate with many types of actuators on sliding-stem valves. Moreover, the Fisher 3661 is an accurate and fast-responding instrument which can resist vibrations of the environment. Also, the position of Fisher 3661 is proportional to a standard milliampere DC input from the SCADAPack 350 [5]. The data form is according to Table 2.3. The specification of the Fisher 3661 is shown in Table 2.7

There are two different types of actuator connections and two separate positioner connections. Each actuator connection can work with both positioner connections. As a result, there are four different mounting configurations in total. The one mounting configuration used in this project is air-to-retract actuator connection combining with direct positioner action. This mounting configuration is shown in Fig 2.12.

This mounting configuration can achieve a positive motion, which means when the input value increases, the output pressure of the actuator increases. As a result, the valve open or close with the input signal increasing or decreasing respectively. Fig 2.13 shows the operational schematic of Fisher 3661.

<table>
<tr><th colspan="2">Model</th><td>Fisher 3661</td></tr>
<tr><th rowspan="2">Signal</th><th>Input</th><td>4 - 20 mA DC</td></tr>
<tr><th>Output</th><td>Pneumatic Pressure</td></tr>
<tr><th colspan="3">Capacity</th></tr>
<tr><th colspan="2">Delivery</th><th>Exhaust</th></tr>
<tr><td colspan="2">1.4 Bar (20 Psig) Supply:
4.3 normal m3/hour(150 scfh)
2.4 Bar (35 Psig) Supply:
6.6 normal m3/hour(230 scfh)</td><td>1.4 Bar (20 Psig) Supply:
4.8 normal m3/hour(170 scfh)
2.4 Bar(35 Psig) Supply:
7.4 normal m3/hour(260 scfh)</td></tr>
<tr><th colspan="2">Configuration</th><td>Single-acting electro-pneumatic valve positioner</td></tr>
<tr><th rowspan="2">Positioner Adjustment</th><th>Span</th><td>19 to 50 mm</td></tr>
<tr><th>Gain</th><td>0.5 to 6% Proportional Band</td></tr>
<tr><th colspan="3">Pneumatic Pressure Supply and Consumption</th></tr>
<tr><th colspan="2">Air Consumption</th><th>Supply Pressure</th></tr>
<tr><td colspan="2">0.24 normal m3/hour (8.8 scfh) at
1.4 bar(20 psig)
0.33 normal m3/hour(12.3 scfh) at
2.4 bar (35 psig)</td><td>Recommended: 10% above actuator requirements
Maximum: 6.2 bar (90 psig) or
pressure rating ofactuator
Medium: Air</td></tr>
</table>

Table 2.7: Specification of Fisher 3661

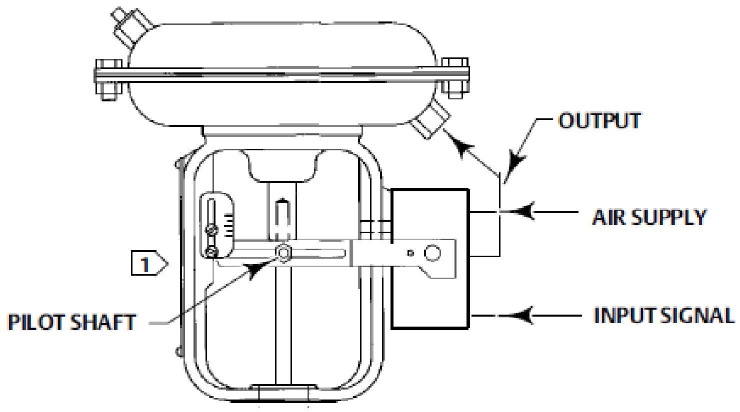


Figure 2.12: Fisher 3661 Mounting Configuration [5]

From Fig 2.13, the process of operation can be seen apparently. For a direct-acting mounting configuration control valve, which is the one used in this project, an increasing (or decreasing) analog input makes the beam move backward (or forward) through the input module. Then, the movement of the beam works on the relay to increase (or decrease) the pressure of actuator. The pressure changes of the actuator cause the feedback plate moving up (or down). As a result, the throttle of the valve

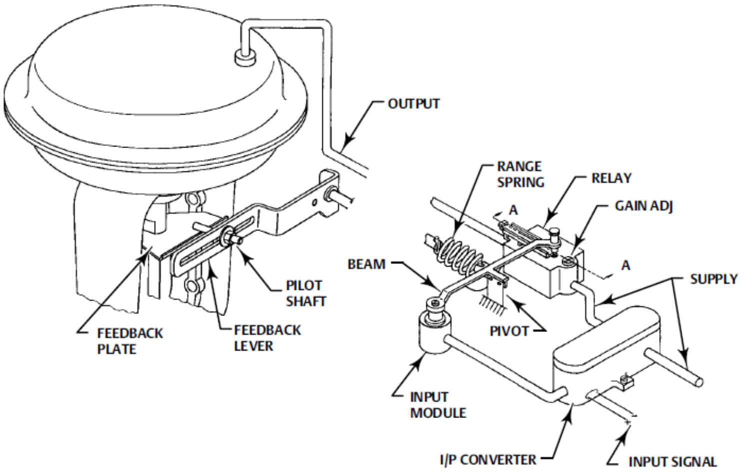


Figure 2.13: Fisher 3661 Operational Schematic [5]

is controlled by the output of the SCADAPack 350.

2.4 System Wiring Schematic

After knowing a power requirement of each device in the system, the next step is going to connect them to achieve control purpose. The overview of power requirement of each device is shown in Table 2.8.

Devices			
WIKA C-10		**Fisher 3661**	
Power Supply	10-30 V	**Power Supply**	NA
Output Signal	4-20 mA	**Input Signal**	4-20 mA
Devices			
SCADAPack 350			
Power Supply		24V	
5209 Controller Board		20mA x 5 = 100mA	
5305 Analog Output		20mA x 2 = 40mA	
Total		140mA	

Table 2.8: Overview of Power Requirement of All Devices

A crucial subject of a PLC is how to connect the PLC to the system being controlled. A complete PLC system design includes connections of all necessary devices, such as control valve, pressure sensor and flowrate sensor for example. Wiring a device to the SCADAPack 350 should include the consideration of appropriate power and current carrying capacity of wires. All considerations are necessary to ensure that all connections are connected properly and correctly to right terminals. Furthermore, before connecting devices to the SCADAPack 350, correct fuses should be added to provide proper protection to the system.

2.4.1 Wiring

In this project, the wiring technique, selection of fuses, and some safety notices are mainly introduced in this section. There are four main parts in the control box, which is shown in Fig 2.1. They are Power Supply 1 and 2, Terminal Block, and the SCADAPack 350. The power supply 1 is a DC adapter. It can transfer 120 VAC input to 20VDC, 600mA output. The power supply 2 is a GE Fanuc standard power supply programmable controller. The power supply 2 can transfer 120 VAC to 24 VDC, 800mA. The system wiring schematic is shown in Fig 2.14. Table 2.9 demonstrates the definition of acronyms of elements.

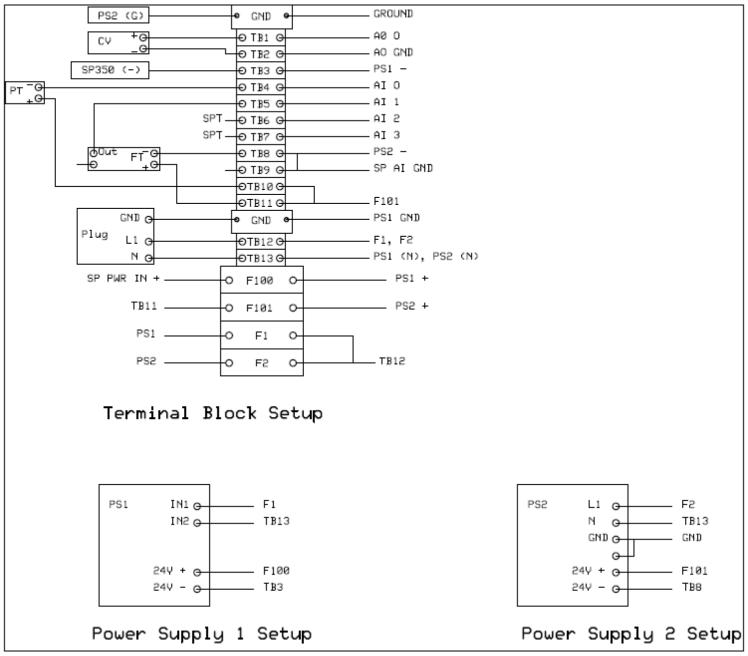


Figure 2.14: System Wiring Schematic

2.4.2 Fuses

From Fig 2.14, the external power supply, which provides a 120 VAC input to the system, is separated into three different wires: Live (L), Neutral (N), and Ground

Acronym	**Element**	**Acronym**	**Element**
CV	Control Valve	PT	Pressure Transmitter
F	Fuse	SP	SCADAPack 350
FT	Flow Transmitter	SPT	Spare Ports
GND	Ground	TB	Terminal Block
PS	Power Supply	AI / AO	Analog Input / Output

Table 2.9: Acronym Explanation

(GND). The PS1 and PS2 are connected to the three different wires correspondingly. To protect the system, fuse 1 and 2 (F1 and F2) are added to play protectors for the live wire in case of a fail external power supply. Then, a 20 VDC and maximum 600 mA current flows through F100 into the SCADAPack 350, and a 24 VDC and maximum 800 mA current flow through F101 into pressure transmitter and flowrate sensor. Both F100 and F101 project the devices in case of a fail PS1 or PS2. The specifications of fuses that are used in this project are shown in Table 2.10.

Name	AGC Series 2.5	**Name**	AGC Series 10
Fuse Current	2.5 A	**Fuse Current**	10 A
Voltage Rating	250 V	**Voltage Rating**	250 V

Table 2.10: Fuses Information

Chapter 3

PLC Programming and Experimental Results

In Chapter 2, we describe the specifications of each experimental components including SCADAPack 350 PLC, WIKA C-10 pressure sensor, SMRT 101 flowrate sensor, and Fisher 3660 positioner. Additionally, the discusses pressure distribution and flow simulation of the water have been analyzed. This Chapter focuses on the SCADAPack 350 PLC programming and the water tank system modelling. The simulation results of the water level control have been presented, and this chapter concludes with the experimental results.

To control the water level of the water tank accurately, it is important to monitor the system output, and compare the measurements with desired output. A PID control algorithm is used to eliminate the difference between the actual output and desired output. The PLC based water level control system is designed as a feedback control system. The pressure sensor monitors the water level by measuring the pressure, and sends the measurements to the SCADAPack 350. The difference between the actual height and the desired height is calculated, and a PID control algorithm is used for eliminating the difference. The control input is calculated by the PLC program, and the control valve takes its action to decrease the difference between the actual height and the desired height. When the difference is zero or nearly zero, the water level successfully tracks the desired reference signal.

3.1 Telepace Studio

In this project, we use Telepace Studio for the PLC programming. This software is used by electricians, engineers and programmers to program sequencing and process control [16]. The ladder diagram is adopted as the programming language in this project. The process of solving a ladder diagram can be briefly described as: I/O update, solve the ladder diagram, and update. In each solving ladder diagram process, all rungs in the ladder diagram can be solved. However, the inputs or outputs states are only changed at I/O update. Moreover, a ladder diagram is similar to an electrical diagram, but the direction of current flow is different from the electrical diagram. In ladder diagram, current can only flow from left to right and from up to bottom. After the logic of the ladder diagram has been defined, a data format is carried out to determine. According to the manual of Telepace Studio, this software can read the process value in analog and digital format, and the analog signal is represented by an integer [16]. Therefore, Telepace Studio satisfies the requirement of the SCADAPack 350.

3.1.1 Telepace Studio Setup

In Telepace Studio, a ladder diagram can be programmed and monitored in the Telepace Ladder Network Editor. To program a ladder diagram in Telepace Studio, there are several steps to follow. The flowchart in Fig 3.1 shows the steps of programming a ladder diagram in the Telepace Studio, and Table 3.1 shows the details of Telepace Studio setup.

3.1.2 PLC Programming

Desired Objective In this water level control system, the desired objective is to maintain the water level at the desired height by controlling the outflow speed of the water tank. In this project, the desired height is set at 40 cm. According to the (2.1), the height in the range of the pressure sensor can be measured, and the result of the (2.1) is the actual height represented by integers. As a result, the desired height is set as 10288 in the Telepace Studio. The actual height of the water level is calculated and represented in the same format in the Telepace Studio.

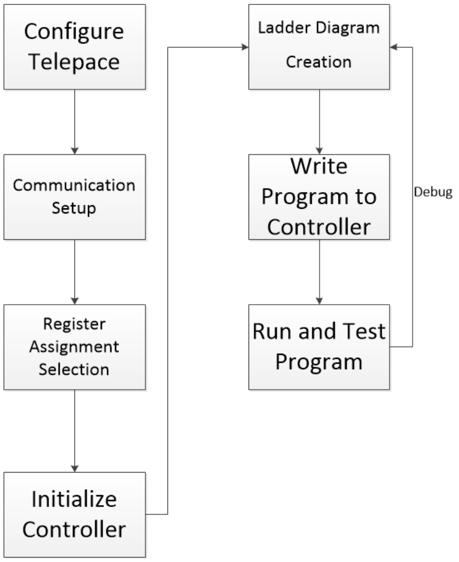


Figure 3.1: Steps of Telepace Setup

Features	**Details**		
Register Assignments	**Type of Controller**		SCADAPack 350 5V/10mA Controller
	Analog Inputs	30001	Pressure Sensor
		30002	Flowrate Transmitter
		30003 & 30004	Spare Port
	Analog Outputs	40001	Control Valve
		40002	Spare Port
Protocol Method	Modbus USB		
Serial Communication	**IP**	192.168.0.2	
	Subnet Mask	255.255.255.0	
	Gateway IP Address	192.168.0.2	
	Station Number	3	

Table 3.1: Details of Telepace Studio Setup

Experimental Process To design a ladder diagram for this project, the first step is to understand how the system works in the experimental. In order to achieve an accurate result, the first objective of the system is to get rid of the air pressure error. Then, the pressure sensor measures the water pressure and transmits the measured result to the PLC controller. Then, by using PID algorithm, the control input signal value is generated and transmitted to the control valve. Finally, the control valve adjust the outflow rate of the water tank to achieve the control objective. By repeating the above process until the actual height is nearly equal to the desired height, we can conclude that this PLC based water level control system achieved its goal. Fig 3.2 is a flowchart which shows the logic of the PLC program.

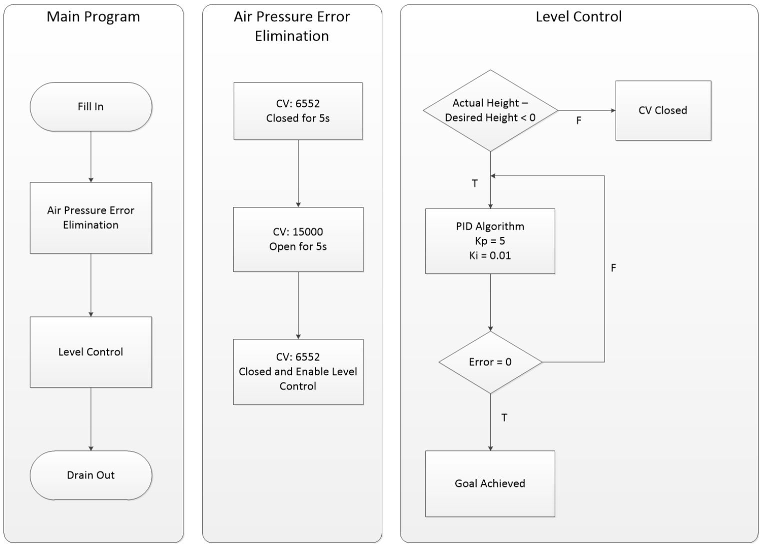


Figure 3.2: The Logic of The PLC Program

Ladder Diagram The ladder diagram consists of different types of function blocks and each function block has its particular register addresses. In the Telepace Studio, each address of a function block can be tagged as a different name in order to be easily monitored and modified in the host PC. The function blocks that are used in this project are SUB, STOF, PIDA, FTOS, ADD, CALL and SUBR, and DLGF. The definitions are shown in Table 3.2.

Function Blocks	**Definitions**
SUB	Subtract Signed Values
STOF	Signed Interger to Floating-point
PIDA	Analog Output PID
FTOS	Floating-Point to Signed Integer
ADD	Add Signed Values
CALL & SUBR	Execute Subroutine & Start of subroutine
DLGF	Data Log to File

Table 3.2: Function Block Definitions

Furthermore, because the Telepace Studio limits the size of ladder diagram up to 8 rungs, the program is written by 3 parts: main Program, air error elimination, and PID level control. The ladder diagram of the main program is shown in Fig 3.3. The first line of the main program is an execute subroutine which is the signal to call for air error elimination (CALL 100). The second line is a latching switch function which is used to turn off CALL 100 and turn on CALL 101, which is the network of the PID level control.

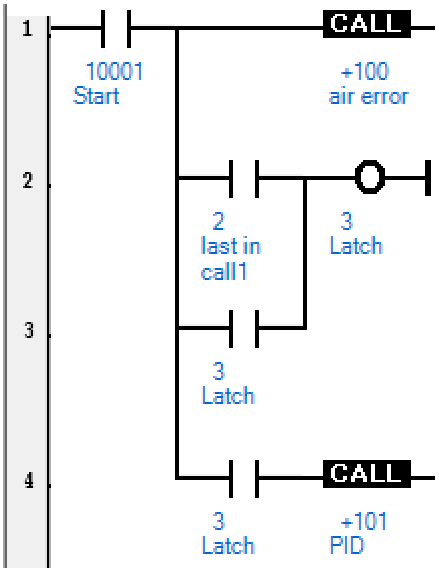


Figure 3.3: Main Program Ladder Diagram

In Fig 3.3, there is a boolean switch 10001 which is the start button of the system. When the start button is pushed, the air error elimination function is executed. Fig 3.4 shows the ladder diagram of air error elimination function. To get rid of the air pressure error, the control valve is turned off for 5 seconds, and turned on for 5 seconds repeatedly for a while. When the start switch is ON, the timer 1 is implemented, and a ‘close’ signal value is set on the control valve for 5 seconds. Then, the coil 1 is implemented. At the same time, the timer 2 is executed, and an ‘open’ signal value is set on the control valve for 5 seconds. Finally, after 5 seconds, a ‘close’ signal value is set on the control valve. At the same time, the coil 2 is executed. The coil 2 causes switch 2 in Fig 3.3 being turned on, and the latching coil 3 turns off the air pressure error elimination and turn on the PID level control network. Fig 3.5 shows the ladder diagram of PID level control.

In Fig 3.5, the desired height in the PLC program is set at 10288, which represents

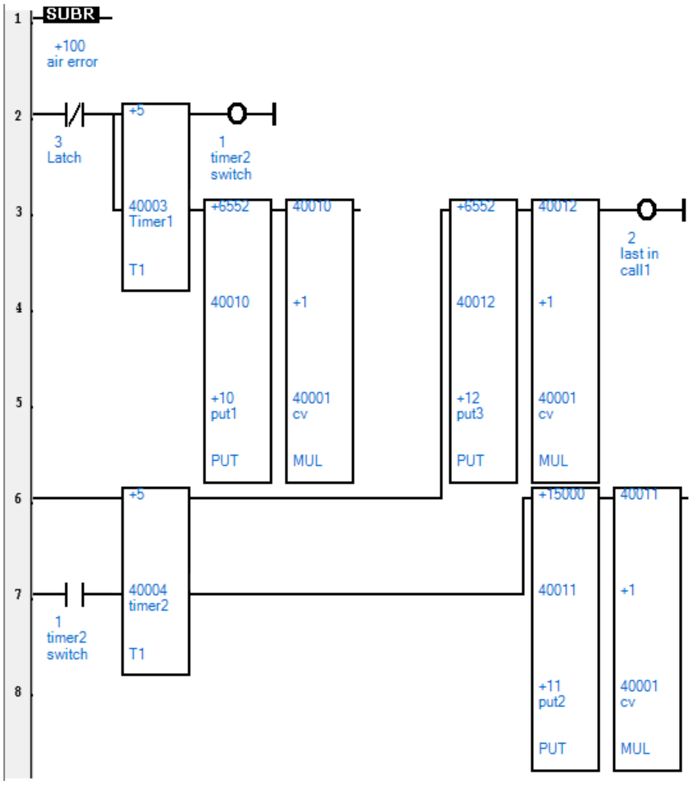


Figure 3.4: Air Pressure Error Elimination Ladder Diagram

40cm. When the PID level control network is activated, the pressure sensor transmits a value into register 30001. At the first function block, the value from 30001 is subtracted by the desired height, and the difference is stored at register 40013. Then, the difference is compared with 0 at CMP function block. If the error is less or equal to 0, which means the actual height is lower than the desired height, the control valve stays at the closed position. With the water level rises, the difference is getting smaller and smaller until the error greater than 0, which means the actual height is higher than the desired height. At this time, the error is transmitted to PIDA function block. At the PIDA function block, the error is calculated by PID algorithm, and a new signal value is generated and held at 40021 register. Before and after the PIDA function block, there are two number converters, STOF and FTOS. According to the Table 3.2, these two function blocks are used for converting integer to floating-point and floating-point to integer respectively. Finally, the signal value at register 40015 is added by 6552, which is the minimum operating current of the control valve, to operate the control valve. As a result of repeating the process above, the water level can be maintained at the desired height 40 cm with inlet flow and outlet flow. To

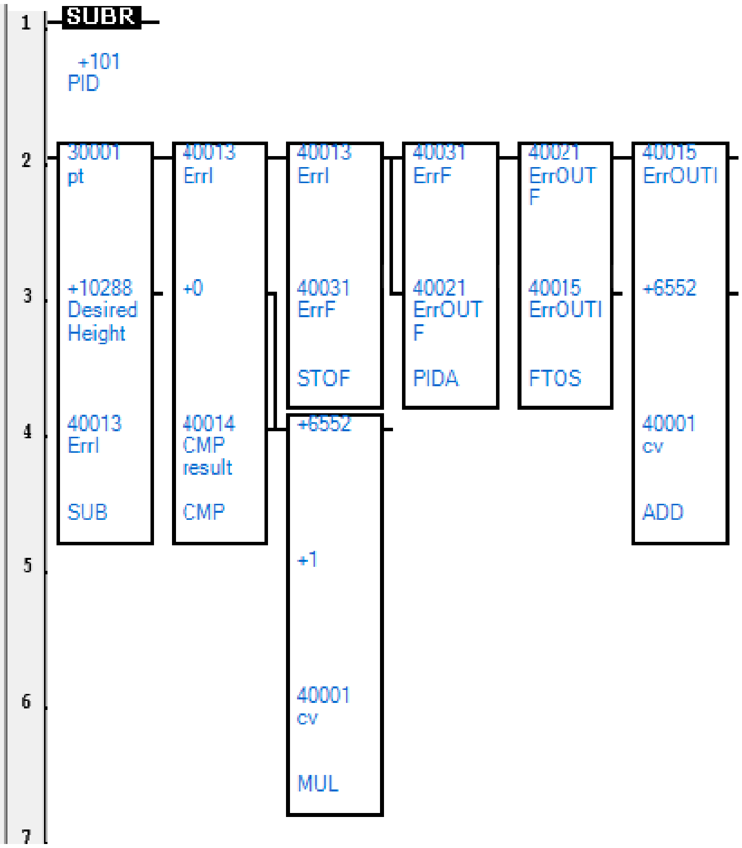


Figure 3.5: PID Level Control Ladder Diagram

drain all water out after the experimental test, there are two more lines under the main program. The ladder diagram is shown in Fig 3.6. The switch 10002 is going to fully turn on the control valve and the switch 10003 is going to close the control valve. This action is done manually.

3.2 System Modeling and Simulation

In this section, we will discuss the system modelling of the water tank system. A simulation is conducted to verify the model, and the simulation will help us to choose the PID control parameters in the experiment.

System Modeling The following symbols are defined:

- H: Actual Height
- H_0: Desired Height

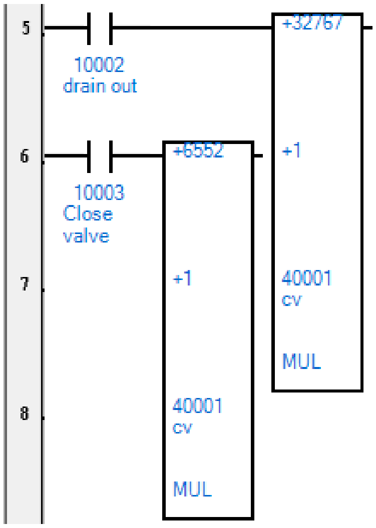


Figure 3.6: Drain Out Ladder Diagram

- A: Cross area of the water tank
- a: Cross area of the pipe
- q_{in}: Inflow of the water tank, which is assumed to be constant
- q_{out}: Outflow of the water tank
- g: Gravity Acceleration
- K_1: Valve flux proportional coefficient
- K_2: Flux proportional coefficient at desired height (H_0)

The physical meanings of the symbols of the water tank system are shown in Fig 3.7. It is noted that the A, a, q_{in}, and g are time invariant parameters, and q_{out} is related to the current value of the control valve.

Then, based on the law of conservation of mass, the actual height is equal to the total inflow mass subtracts the total outflow mass. As a result, (3.1) shows the relationship among these three parameters. The outflow can be calculated by (3.2).

$$A \times \frac{\partial H}{\partial t} = q_{in} - q_{out} \tag{3.1}$$

$$q_{out} = K_1 a \sqrt{2gH} \tag{3.2}$$

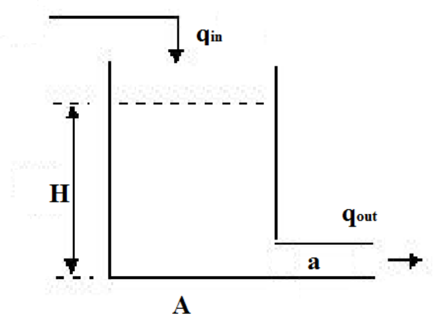


Figure 3.7: Water Tank System

From (3.2), it can be seen that the water level control system is a non-linear system. However, around the desired height, the outflow equation can be linearized as a linear model. (3.2) is rewritten approximately as (3.3):

$$\frac{q_{out}}{2gH} \approx \frac{K_1 \times a}{\sqrt{2gH_0}} = K_2 \tag{3.3}$$

$$q_{out} = K_2 \times 2gH \tag{3.4}$$

Then, by substituting (3.4) to (3.1), the model of the system is generated as (3.5):

$$A \times \frac{\partial H}{\partial t} = q_{in} - K_2 \times 2gH \tag{3.5}$$

where, $K_2 = \frac{K_1 \times a}{\sqrt{2gH_0}}$, and $K_1 = \frac{I_{act}}{I_{max}}$ because the valve coefficient is proportional to the valve position and equal to the actual current of the valve over the maximum current of the valve. By taking K_1 and K_2 in to (3.5), the model of the system can be represented by the actual current I_{act} and the actual height H. The equation is written as (3.6) The values of each parameters are shown in Table 3.3.

$$\frac{\partial H}{\partial t} = C_1 \times q_{in} - C_2 \times I_{act} \times H \tag{3.6}$$

where, the constant C_1 and C_2 can be calculated by $C_1 = \frac{1}{A}$, and $C_2 = \frac{2ga}{\sqrt{2gH_0} \times I_{max} \times A}$.

Simulation Result According to (3.5), the simulink model is built up as shown in Fig 3.8 and Fig 3.9. The simulink model consists of two parts: the main program and a subsystem which is the water tank model.

Parameters	**Value**	**Unit**
A	78.5	cm^2
a	1.25	cm^2
H_0	40	cm
q_{in}	40	cm^3/s
g	1000	cm/s^2

Table 3.3: Parameters

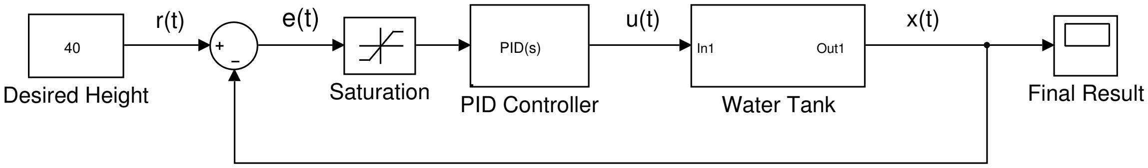


Figure 3.8: Main Simulink Block Diagram

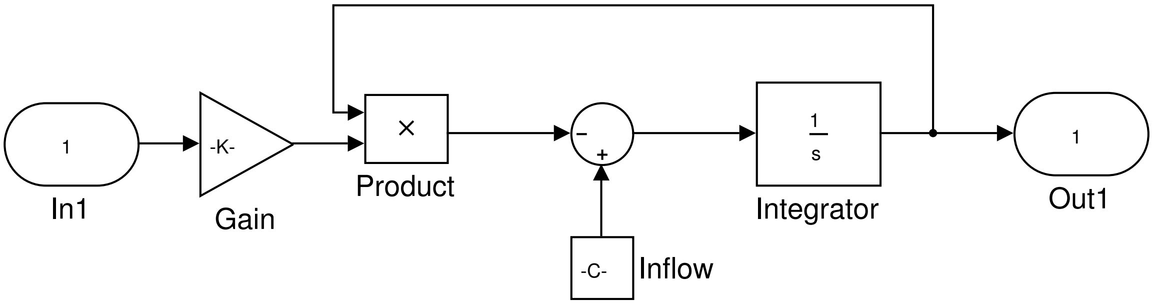


Figure 3.9: Water Tank System

The desired height is represented by a constant signal in the constant function block. The reference signal is marked by $r_{(t)}$. The error between the desired height and the actual height, which is represented by $e_{(t)}$, is generated after the sum function. The saturation function is set from the lower bound -20 to upper bound 0 because if the actual height is lower than the desired height, the system does not take any actions except being filled. When the actual height is near the desired height, the PID function block is enabled. The PID controller calculates a control signal $u_{(t)}$ based the $e_{(t)}$. Then, the control signal $u_{(t)}$ is applied to the subsystem which is tagged as Water Tank in Fig 3.8. In the subsystem, Fig 3.9, the gain block represents the parameter C_2. Since the inflow q_{in} and the cross area of the water tank A are constant, the result of $\frac{q_{in}}{A}$ is set as a number into a constant function block marking as inflow in Fig 3.9. Finally, the actual height is integrated by the integrator, and the output of the system is $x_{(t)}$. The simulation overview result is shown in Fig 3.10

and a magnified result is shown in Fig 3.11.

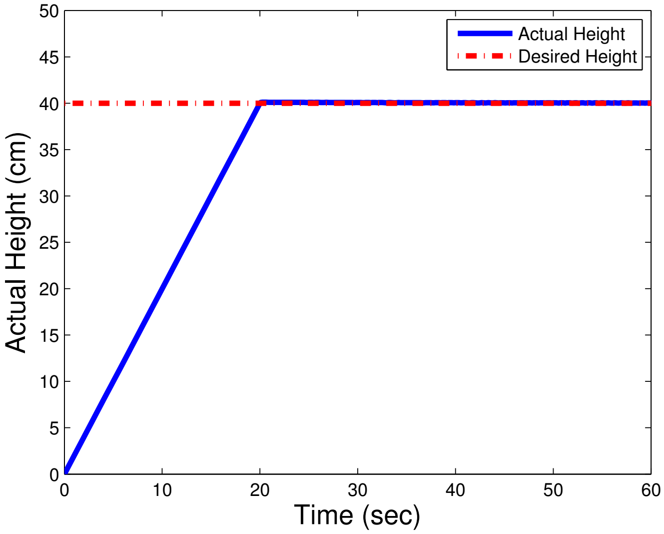


Figure 3.10: Overview Result

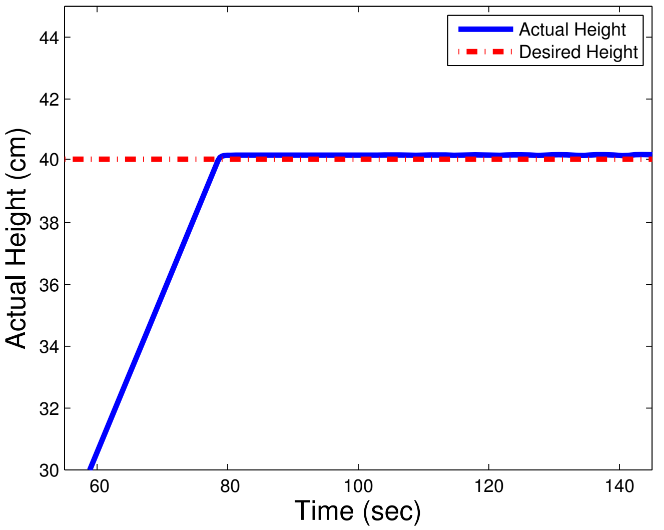


Figure 3.11: Magnified Result

From Fig 3.10, it can be seen that the actual height goes up with a constant slope because the system is under the filling process without piping out action. When the actual height is over 40 cm, the PID function is enabled. The PID adjustment is reflected in the period after 20 seconds. From the simulation result, we can see that the PID controller works very well, and the final goal is achieved. However, this simulation result does not cover air pressure error elimination.

3.3 Experimental Results

In this section, we will explore the experimental results of the water level control of the water tank. Then, by comparing the experimental results with the simulation results, we can verify the design of the overall system. Also, we compare the performance of the PID controller with that without PID controller.

3.3.1 Open Loop Control Scheme

In an open loop control system, the control valve only has two positions: fully close and fully open. When the actual height is higher than the desired height, the control valve is fully opened to drain out water. Similarly, the control valve is fully closed when the actual height is lower than the desired height. As a result, the water level raises up and down but does not stay at a constant level. The experimental data result of the open loop control scheme is shown in Fig 3.12.

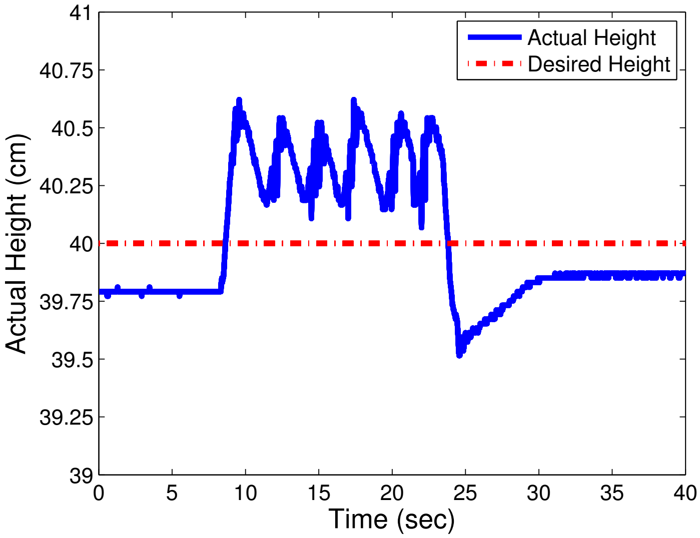


Figure 3.12: Experimental Result of Open Loop Control System

3.3.2 Closed-loop Control Scheme

Different from the open loop control, the position of the control valve can be adjusted upon the requirement of the system. For example, when the actual height decreases from a higher height to the actual height, the control valve is closed slowly instead of fully closed. The open-close degree of the control valve is determined by the difference between the actual height and the desired height. As a result, the actual height can be maintained nearly at the desired height. An overall experimental result is shown in Fig 3.13. Fig 3.14 shows a zoomed part of Fig 3.13 from 60s to 140s.

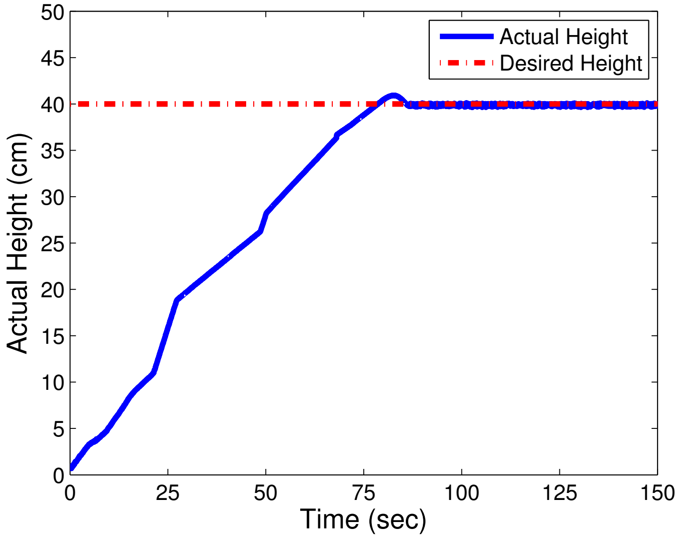


Figure 3.13: Experimental Result of Closed Loop Control System

3.3.3 Experimental Result Analysis

Comparing simulation results in Fig 3.10 with the experimental results in Fig 3.13, we can conclude that the overall experimental result is almost the same to the simulation result. The main difference occurs in the level raising up period. In the simulation result, the actual height is raising up in a constant rate; however, the raising rate in the experiment has a significant change. The reason is due to the sensitivity of the pressure sensor. During the period of the system operation, any vibrations happened to this system can cause a change to the value of the pressure sensor. As a result, the actual height can not raise up straightly in this system. Between Fig 3.11 and Fig 3.14, the difference is the transient time. In the simulation result, we can see that it

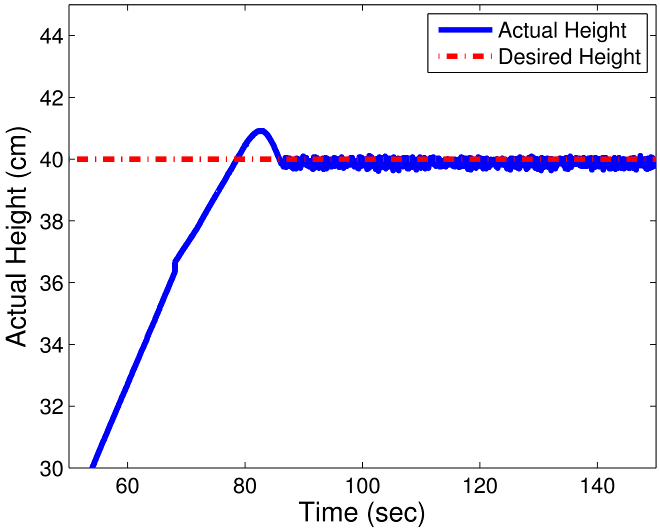


Figure 3.14: Zoomed Experimental Result of Closed Loop Control System

takes 40 seconds. However, in the experimental result, it only takes approximately 3.5 seconds in the transient period. This is because of the different models. As we mentioned above, around the desired height, the outflow can be modeled linearly in the simulation, and some intricate details are simplified; however, in the real system test, the outflow is a nonlinear model and more sophisticated than the simulation. Overall, the simulation result exhibits an approximate situation of the experimental result.

A comparison between Fig 3.12 and Fig 3.13 shows an importance of a PID controller. In an open loop control system Fig 3.12, it can be seen clearly that the actual height is maintained at a higher level than the desired height. Also, Fig 3.12 shows that the actual height oscillates around 43cm. Then, the actual height is near but still above the desired height because some remaining water in the inlet pipe goes into the water tank. All in all, the result of open loop control system is unacceptable. However, when a PID controller is added, the final result in Fig 3.13 is acceptable. Because there is a time delay between the SCADAPack 350 and the control valve, from Fig 3.13, we can still see errors between the actual height and the desired height, but the errors are invisible and acceptable.

Chapter 4

Conclusion and Future Works

4.1 Conclusion

This report investigates a designing process of PLC based level control system. In Chapter 1, a brief history and development of PLC are introduced. The initial idea of developing a PLC is to simply a process control by using PLC to replace electrical relays. With an increasing requirement of the industrial process, PLCs become a significantly used controller in most industries. Also, a water level control system is a typical process control system, so understanding how a water level control system works is necessary for us to know other design procedure in other industrial process control system. For this reason, a PLC based water level control system is designed in our laboratory.

The experimental setup is proposed in Chapter 2. In this system, an SCADAPack 350 PLC, Fisher 360 control valve, and WIKA C-10 pressure sensor are used. To mount all the equipment, a steel frame is installed on a cart. Furthermore, in order to select a water tank gasket and set the position of filling entrance correctly, a water tank finite element analysis and flow simulation are described in Chapter 2. In Chapter 2, the PLC schematic is discussed. With filling water into a water tank, the pressure sensor converts the water pressure to the analog signal. The PLC controls the valve based on the analog signal which comes from the pressure sensor.

After a complete experimental setup, Chapter 3 introduces the PLC ladder diagram programming, system modeling and simulation results, experimental results and system stability test results. The PLC ladder diagram programming is achieved on Telepace Studio which is developed for SCADAPack Series PLC. Before the experimental test, the system simulation is run on Matlab/Simulink. The system modeling is based on the law of conservation of mass. In order to get a better result, a PID control algorithm is introduced. In programming PLC, a PID control algorithm is used. A PID control block is provided in the Telepace Studio. By using PID control algorithm, we get better results on both experimental tests and the simulation. As results showing, the PLC based level control system is stable and accurate.

4.2 Future Works

In this project, the PLC based level control system is used to maintain the water level at a constant height. The simulation and experimental results show that the PLC contributes an accurate and stable performance. However, the weak point is that we only focus on the final result but ignore the system efficiency, such as responding time, air consumption, and power consumption and so on. As a result, a further study on improving system efficiency is required.

Next, due to financial limitation and equipment size limitations, this project only discusses on a simple water tank level control problem. However, in real industries, a control problem is more comlicated than what we did in our laboratory. Therefore, a couple water tank interacting system design and control will be studied in the future.

In addition, an HMI should be considered. In industry, an HMI is used in combination with a PLC based system. The HMI is playing a role in monitoring the system. Moreover, engineers can easily modify the system parameters setup using the HMI. However, the HMI programming is a challenging and valued work in designing a PLC based system so that the further work will focus on the HMI programming.

This project is a short distance control system. In order to apply PLCs to the real industrial control problems, how to design a long distance control and monitor system is necessary. From previous work which we reviewed in Chapter 1, an Internet based remote control technique can be used with a PLC system. To achieve that, the TCP/IP setup is worth to be proposed as a part of the future work.

Bibliography

[1] V. R. Segovia and A. Theorin, “History of Control History of PLC and DCS,” 2012.

[2] M. D. Schwartz, J. Mulder, J. Trent, and W. D. Atkins, “Control system devices: Architectures and supply channels overview,” *Technical Report SAND2010-5183, Sandia National Laboratories*, 2010.

[3] *SCADAPack 350 Installation, Operation and Maintenance Setup Manual*, 2nd ed., Schneider Elctric, 2011.

[4] *General Purpose Transmitter Type C-10*, WIKA Instrument Corporation, 1000 Wiegand Boulevard Lawrenceville, GA 30043, 2010.

[5] *Fisher 3660 and 3661 Positioner*, EMERSON Process Management, Marshalltown, Iowa 50158 USA, 2015.

[6] F. Didactic, “Programmable Logic Controllers Basic Level,” 2002.

[7] W. Bolton, *Programmable logic controllers.* Newnes, 2015.

[8] Wikipedia, “Modbus — wikipedia, the free encyclopedia,” 2017. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Modbus&oldid=775804169

[9] I. Modbus, “Modbus application protocol specification v1. 1a,” *North Grafton, Massachusetts (www. modbus. org/specs. php)*, 2004.

[10] Y. Kondratenko, O. Korobko, O. Kozlov, O. Gerasin, and A. Topalov, “PLC based system for remote liquids level control with radar sensor,” in *2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 1. IEEE, 2015, pp. 47–52.

[11] P. Panchal, A. Patel, and J. Barve, "PI control of level control system using PLC and LabVIEW based SCADA," in *2015 International Conference on Industrial Instrumentation and Control (ICIC).* IEEE, 2015, pp. 1196–1201.

[12] M. Sreejeth and S. Chouhan, "PLC based automated liquid mixing and bottle filling system," in *IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES).* IEEE, 2016, pp. 1–5.

[13] S. Da'na, A. Sagahyroon, A. Elrayes, A. Al-Ali, and R. Al-Aydi, "Development of a monitoring and control platform for PLC-based applications," *Computer Standards & Interfaces*, vol. 30, no. 3, pp. 157–166, 2008.

[14] D. Engin and M. Engin, "Auto-tuning of PID parameters with programmable logic controller," in *2013 IEEE International Conference on Mechatronics and Automation (ICMA).* IEEE, 2013, pp. 1469–1474.

[15] R. E. Samin, L. M. Jie, and M. A. Zawawi, "PID implementation of heating tank in mini automation plant using programmable logic controller (PLC)," in *2011 International Conference on Electrical, Control and Computer Engineering (INECCE).* IEEE, 2011, pp. 515–519.

[16] *Telepace Ladder Logic User and Reference Manual*, Control Microsystems, 48 Steacie Drive Kanata, Ontario, Canada, K2K 2A9, apr 2008.

[17] Wikipedia, "Serial communication — wikipedia, the free encyclopedia," 2017. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Serial_communication&oldid=765810491

[18] ——, "Ethernet — wikipedia, the free encyclopedia," 2017. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Ethernet&oldid=772682320