8-2012

Water treatment and distribution simulation for a SCADA security testbed.

Kyle Thomas Moss
University of Louisville

WATER TREATMENT AND DISTRIBUTION SIMULATION FOR A SCADA SECURITY TESTBED

By

Kyle Thomas Moss

B. S., Western Kentucky University, 2008

A Thesis

Submitted to the Faculty of the

University of Louisville

J. B. Speed School of Engineering

as Partial Fulfillment of the Requirements

for the Professional Degree

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

University of Louisville

August 2012

WATER TREATMENT AND DISTRIBUTION SIMULATION FOR A SCADA SECURITY TESTBED

Submitted by:_____

Kyle Thomas Moss

A Thesis Approved On

7/24/2012

July 24th, 2012

by the Following Reading and Examination Committee:

_____

James H. Graham, Co-director

_____

Jeffrey L. Hieb, Co-director

_____

John Naber

ACKNOWLEDGEMENTS

I would like to thank Dr. Graham and Dr. Hieb for the opportunity to work on this project and for the constant support and advice they have provided over the course of the research. I would also like to thank Mr. Jacob Schriever for the assistance he has provided in the lab and the faculty of the Speed School of Engineering at the University of Louisville for their guidance and support during my studies in the program. I want say thanks to my parents, Roger and Vickie Moss, not only for their support during my time in college, but for their constant love and guidance throughout my life. Finally to all my family and friends, I want to say thanks for your support.

ABSTRACT

WATER TREATMENT AND DISTRIBUTION SIMULATION FOR A SCADA SECURITY TESTBED

Kyle Moss

July 24th, 2012

Supervisory Control and Data Acquisition (SCADA) systems are used in almost all industrial processes including use in the nation's critical infrastructure. The electric, water, and gas industries are merely a few that rely heavily on the use of SCADA systems in order to provide reliable service to the public. Any disruption in these systems would lead to major issues in day to day life and could produce a hazardous environment until the services are restored. SCADA equipment was first implemented decades ago, and in some cases the equipment deployed at that time is still in use today. As network technology emerged and advanced over the last several years, SCADA systems were adapted in order to provide network access and control from remote locations. This led to vulnerabilities in limiting access to the system and provided a means for hackers, hactavist, and nation-states to gain control of critical infrastructure SCADA systems in order to cause both physical and economical damage.

New technologies and research areas have emerged in an effort to thwart these possible intrusions and attacks. However, there is a need to have adequate means of

testing new security devices since it would be impractical to test on a functioning SCADA system. This leads to the development of simulations and testbeds that can provide a low-cost, easily configurable means of testing new cyber security devices.

A water treatment and distribution simulation was developed in order to provide this means of testing. The simulation encompasses two components. The first is a software simulation that provides virtualized components typically found in water systems such as pumps, valves, and water tanks. The second is a hardware component that provides an interface from the software to actual SCADA equipment such as remote terminal units and human machine interfaces. The simulation was tested with a prototype cyber security device to ensure functionality. Attacks were carried out on the SCADA system with and without the security device in place. The simulation allowed for both a virtualized and physical response to the attacks. The simulation provided a robust, cost-effective testbed for verifying the functionality of the security device.

TABLE of CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

Modern day processes are automatically controlled by computer systems. A combination of embedded systems, sensors, and software are used to increase efficiency and decrease cost. Almost all industrial processes, from electricity production and distribution, water treatment and distribution, transportation, oil and gas pipelines, and even the financial industry use Supervisory Control and Data Acquisition (SCADA) systems. SCADA systems allow for accurate monitoring of the overall process in near-real time for many cases, and allow operators to control and monitor systems at remote sites.

As new technologies have emerged, the use of remote monitoring and random access to the SCADA system by multiple operators and departments within a utility has increased the vulnerability of the system to cyber attacks. SCADA technology is expensive to deploy so components in the system are rarely upgraded. Many legacy systems have controllers and communication systems that are approaching 30 years old; designed in a time that cyber attacks were not an issue, which leaves them extremely vulnerable. More modern systems that utilize IP based communication techniques implement firewalls in order to provide a separation between the business local area network (LAN) and SCADA LAN. However many times the firewalls are configured inappropriately, with ports opened in order to allow remote access into the SCADA

network. Attackers are then able to exploit the holes in the firewall and gain access to the SCADA side through the business network. It is also important to note that security risks come from both outside threats, such as nation states, hacker groups, and independent hackers, as well as insider threats such as disgruntled employees.

In 2010, one of the most successful and highly noted attacks on a SCADA system was discovered. STUXNET used a rootkit to infect Siemens programmable logic controllers (PLCs) in Iranian nuclear facilities. The worm traversed Windows based operating systems until it found specific controllers configured with variable frequency drives and eventually caused severe damage to uranium enrichment centrifuges. The worm exploited three zero-day vulnerabilities [1].

For these reasons, there has been an increased push for research and development in the area of SCADA security. The University of Louisville is currently working on a field device security preprocessor using a microkernel running on a Gumstix® embedded processor [2]. The research efforts are focused on adding security to legacy systems used in the water sector. Other university labs, government labs, and private sector organizations are also performing research in this area.

A challenge for those testing devices that are under development arises because it would be impractical, and possibly extremely hazardous, to test the units on a functioning SCADA system. In the water sector, for example, errors in the hardware could lead to lose of control in the SCADA system or inability to collect data from remote sensors. Even more dangerous, the device could lead to damage of pumps, values, and water lines

in the system leaving the area without consumable water for an extended period. For this reason it is necessary to have a test bed for developing security devices.

In this thesis, the design and development of a water treatment and distribution simulation will be discussed. The simulation incorporates both software and hardware to mimic a water system and interfaces to a remote terminal unit (RTU) similar to those used in an actual process. By using a software-based approach to simulate a water system, there is not a need to purchase expensive hardware to test the prototype in development. A software simulation also allows for easy reconfiguration of the system.

This research is directly for use with the hardened remote terminal unit security pre-processor currently being developed at the University of Louisville. The security device is considered a "bump in the wire" approach, being connected to the RTU in line with the MTU/HMI. The simulation provides both a virtual and physical response to simulated attacks that are carried out on the SCADA network. Figure 1.1 shows a block diagram of how the simulation software and hardware fit into the system. The individual components will be discussed in more detail in later chapters.

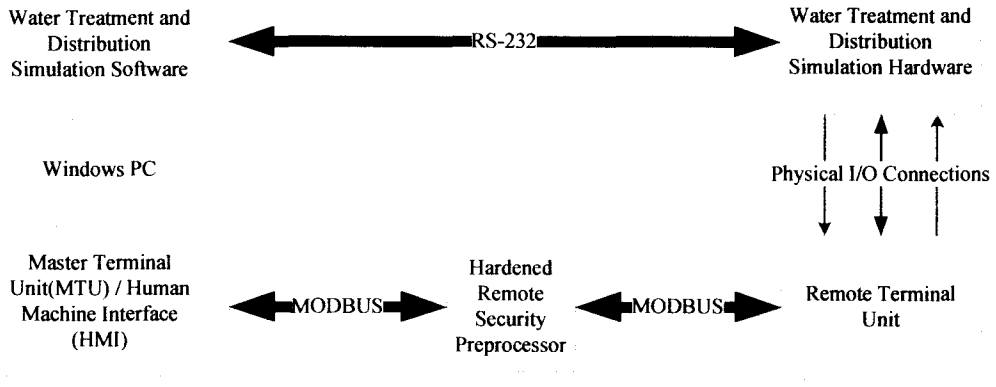


Figure 1.1: Illustration of Simulation Integration with SCADA Network

The remaining chapters in the thesis are organized as follows. Chapter 2 presents an overview of relevant literature in this research area. Chapter 3 presents the model development for the water sector simulations. Chapters 4 and 5 discuss the implementation of the water sector model in both a software and hardware simulation. Testing and results of the simulation are presented in chapter 6. Finally, conclusions and future work are discussed in chapter 7 of the thesis.

II. LITERATURE REVIEW

This chapter provides an overview of the published literature which is relevant to this thesis. It is divided into three parts. Part one is an overview of the water treatment and distribution system; the processes involved, the topology of typical systems, and standard operations. Part two is an overview of SCADA systems used in the water sector including the hardware used, communication systems implemented, operator interfaces, and the cyber threat to these systems. Finally, an overview of existing water simulations will be presented.

A. Water Treatment and Distribution Systems

Understanding the operation of a water treatment and distribution system is necessary in order to develop a model simulating their functions. Although no two systems are identical, there are many similar functions and components in the systems regardless of size or geographical location. Water systems involve two main processes, treatment and distribution. To understand the operations more thoroughly, these two processes will be discussed separately.

1. Water Treatment Process

Water treatment is the process of taking groundwater or surface water from a stream, lake, or river and removing harmful bacteria, viruses, dirt and other contaminates in order to provide safe drinking water to the public. The process varies depending on location and the source of the water, but usually involves five steps: coagulation, sedimentation, filtration, disinfection, and storage as shown in Figure 2.1.

Coagulation is the process of adding chemicals such as Alum to the water to help dirt and other particles stick together. As the particles combine together they become heavier and sink to the bottom of the tank in the sedimentation phase. With the dirt removed, the water is then passed through a filtration process. Most typically, sand, gravel, and charcoal are used to filter the water; removing smaller particles that were not removed during sedimentation.

Chlorine is then added in the disinfection phase in order to kill bacteria and other microorganisms in the water. Next, the water is tested to ensure it meets the Environmental Protection Agency's (EPA) standards and then is stored for distribution into the water system [3].

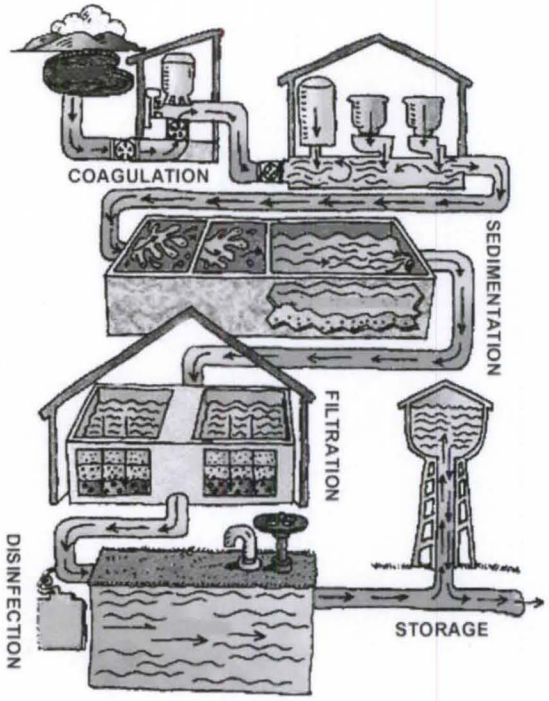


Figure 2.1: Water Treatment Process (EPA, 2012)

2. Water Distribution Process

After treatment, the water must be distributed to homes, businesses, and industries for use. Similar to the treatment process, the distribution system can vary greatly depending on its size and location, but there are many commonalities with all distribution systems. A distribution system is comprised of water pipes (lines), storage tanks, pumps, and valves. The distribution system provides uninterrupted, pressurized drinking water to the community it serves. Large capacity pumps transport water from the treatment storage tanks to large elevated water towers or ground tanks throughout the coverage area using water mains. Once pumped into elevated tanks, gravitational force exerted on the water pushes the water out into the distribution pipes at a pressure directly related to the difference in elevation of the water in the tank to the pipe system. The pressurized water

travels through the pipes of varying diameter branching off through smaller lines to water meters and eventually into the customer's building. In certain cases, additional pump stations may be connected to the distribution lines in order to maintain adequate pressure in the system. [4].

Each elevated storage tank includes at least two pumps; both are equipped with a valve that prevents backflow through the pump. An alternating switch attached to both pumps helps distribute the load on each pump equally over time. The pump which is scheduled to turn on during the next request is known as the lead pump and the secondary is known as the lag pump. During operation, the system will call for the lead pump to turn on when water levels in the tank drop below a set value, known as the lead minimum level. The pump will fill the tank until it reaches a second set value known as the tank maximum level. These two limits define the normal operating levels of the tank. In the event that the lead pump fails to turn on, or the demand for water exceeds the capacity of the lead pump to fill the tank, the water level in the tank will continue to drop. A lag pump minimum level is predefined in the system. When the tank reaches that point, the lag pump will turn on in order to maintain sufficient water in the tank. Once the tank reaches its maximum level, the pump(s) will shut off.

The water in the storage tanks must flow out into the distribution system to customers. This is done through various types and sizes of pipes buried underground. There are two topologies to water distribution pipe lines: branch architecture and loop architecture. Branch architecture is used in rural distribution where homes and businesses are widely separated. Large diameter water mains are distributed from the elevated tanks and smaller diameter pipes branch off into subdivisions or industrial parks.

These lines branch into smaller lines, which run to water meters at individual homes and businesses. This is similar to branches on a tree, their diameter and lengths become shorter at each branch point.

Loop architecture is used in more densely populated areas utilizing a large diameter water main that circles the area and loops back into itself. Small lines branch out of the loop into homes and businesses. Depending on the size of the area, multiple loops can be used. Figure 2.2 shows a water system with loop architecture for distribution.

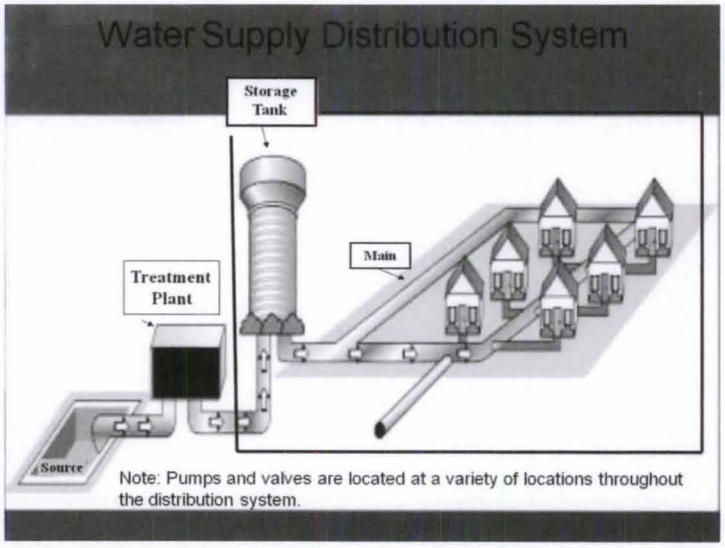


Figure 2.2: Water Distribution System (EPA, 2012)

B. SCADA Systems in Water Sector

1. Hardware

Throughout the treatment and distribution system, SCADA systems are implemented in order to: control pumps and values, measure and control the addition of chemicals to the treatment process, and to monitor tank levels and pressure levels.

Remote terminal units (RTU's) are used to collect sensor data such as tank levels and operation status of pumps, as well as to control relays, which turn pumps on and off. Intelligent RTUs are able to do control automatically based on predetermined metrics of the system but still provide telemetry and in-turn operators maintain control. The RTUs are programmed to run the system with little input from operators.

In most cases, pumps in a distribution system operate at a constant flow rate (either on or off), but some systems implement variable frequency drives in order to regulate the flow of water into storage tanks. Control of the pump's velocity would be controlled by the RTU as well. The RTU at each remote site transmits data and the system status to a master control unit (MTU) usually located at the control center for the water system. In many situations this is collocated with the treatment facility or treatment control center.

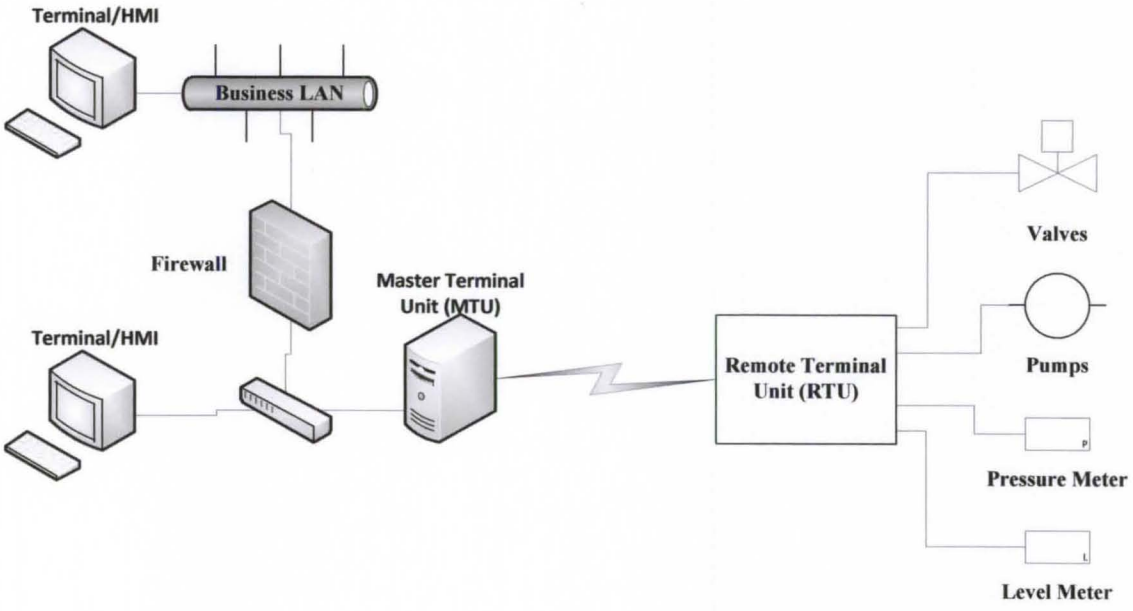


Figure 2.3: Typical SCADA System Diagram

2. Communication

Various communication mediums can be used within a SCADA system. For rural or large area distribution, radio frequency (RF) based technologies are the most practical form of communication. These systems use a modem connected to the RTU communication interface to convert from a digital to analog signal. The output is then connected to a radio and large antenna to transmit the data to another location. For metropolitan or smaller service areas, fiber optic, plain old telephone service (POTS), or IP based communication (Ethernet or Wi-Fi) can be used. Most water systems are unique and these methods of communication are only a few examples of what may be found.

In addition to various communication medium, there are also many SCADA protocols. MODBUS is a popular protocol and is one of only a few open protocols. MODBUS is an application layer messaging protocol which allows it to operate on any lower layer architecture and with devices connected on different types of networks or buses [5]. It operates on a client/server model with the MTU functioning as the server and the RTU as the client. As previously mentioned, RTUs have the functionality to control analog and discrete outputs (i.e. turning pumps on/off) and read sensor data on both analog and digital inputs (i.e. pressure levels, tank levels). Communication between the MTU and RTU use a specific set of function codes and other data in a MODBUS packet. This is outlined in detail in the MODBUS RFC.

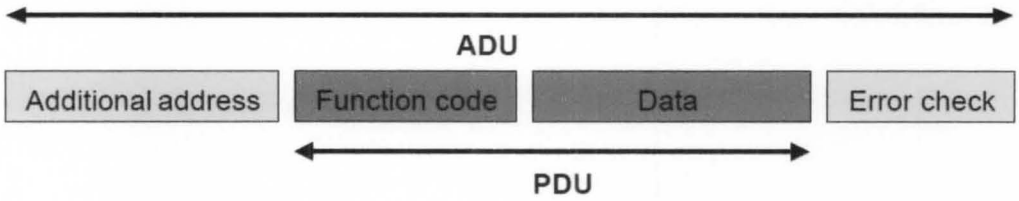


Figure 2.4: MODBUS Frame Architecture [5]

Primary tables	Object type	Type of	Comments
Discretes Input	Single bit	Read-Only	This type of data can be provided by an I/O system.
Coils	Single bit	Read-Write	This type of data can be alterable by an application program.
Input Registers	16-bit word	Read-Only	This type of data can be provided by an I/O system
Holding Registers	16-bit word	Read-Write	This type of data can be alterable by an application program.

Figure 2.5: MODBUS Data Types [5]

In its original form, MODBUS utilized RS-232 or RS-485 communication standards in the transport layer. Legacy systems in the water treatment and distribution system will probably implement this form. A TCP/IP version of MODBUS is also available. Figure 2.6 shows the MODBUS TCP/IP stack.

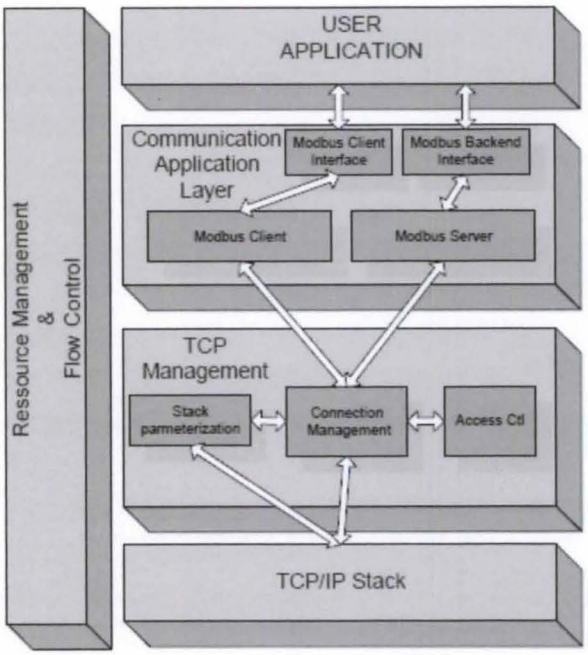


Figure 2.6: MODBUS TCP/IP Stack [5]

3. Operator Interface

Once the data is received by the MTU, there is a need to display this information to operators in the control center. This is done by use of human machine interfaces (HMI). The HMI provides the operator with the ability to manually turn pumps on and off and indicates any errors in the system such as low pressure levels. This may indicate a water main break, or failure of pumps and other devices in the system. In most cases, data recording devices are attached to the SCADA system to provide a record of operation. Figure 2.7 shows what a typical water treatment HMI would look like.

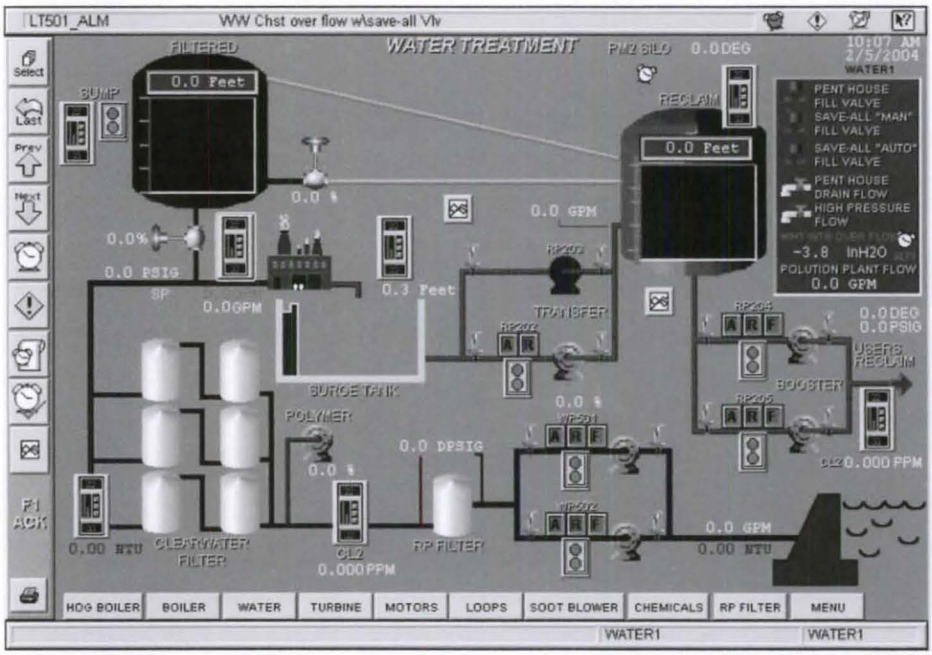


Figure 2.7: Water Treatment HMI [6]

4. Cyber Vulnerabilities in SCADA Systems

In 2005, a joint meeting of several Congressional Subcommittees was held to focus on the threat of a cyber attack on SCADA systems. At that meeting, experts, including government officials, national lab scientist, and other experts in the field, testified about the serious threat that was posed to critical infrastructure. Dr. Sam Varnado, Director of Information Operations Center at Sandia National Lab, outlined some of the research they were conducting using a Red Team approach. At that time several vulnerabilities in SCADA systems were uncovered [7].

In most cases, cyber threats are classified into one of the following categories: insider intentional threats, internal unintentional threats, external nontargeted threats, or malicious actors [8]. Insider intentional threats are those that occur from individuals working inside the utility, or those who have an expansive knowledge of the SCADA

system. An example of this type of attack occurred in Queensland, Australia in 2000 in which 800,000 liters of raw sewage was intentionally discharged from a wastewater treatment plant into local parks and rivers. Vitek Boden, a previous employee of the company that installed the SCADA system at the treatment facility, used his knowledge and access of the system to generate the spill [9].

As SCADA networks have slowly become integrated into corporate networks with access to the Internet and company internetworks, the threat of internal unintentional threats has increased. These threats occur due to inappropriate design of the networks, such as the lack of firewalls between the SCADA and business LAN, and improper IT procedures such as using default passwords and allowing broad access to the network to individuals that don't require it. These threats pose a real danger to SCADA security.

The third type of threat is generated from computer viruses and worms that are not targeted directly at industrial control systems (ICS). Improper application of security updates and patches by IT personnel can cause failure in critical system equipment such as safety systems and HMI computers. These viruses can also open pathways for intrusion to outside threats.

Finally, malicious actors are those who are targeting the ICS directly. This may include hackers, hactivist, and nation-states. Many times these are well financed groups with long backgrounds in malicious cyber activity. These attacks are difficult to prevent due to their complexity and sophistication [10] [11] [12] [13].

Cyber vulnerabilities exist in many areas of the SCADA system. Attacks could be directed on the HMI to provide the operator with incorrect information or take control

of the system. An attacker could also take control of the RTU directly, forcing pumps on or off which could lead to water tanks overflowing or completely emptying. Both could cause major disruption to the system. An attacker could gain access to the network to launch an attack through interception of wireless data via RF radios, or through a physical connection at a remote site.

C. Simulation of Water Treatment and Distribution

There are several technical areas in water resource management that require modeling the water treatment and distribution process. Many of these focus on system hydraulics and water quality. Since individual water systems are unique, meeting the specific requirements of its user base, it is important to have an accurate method of predicting the behavior of the system as changes are made, as well as identifying any issues that occur as a result of a simulated cyber attack.

Several simulations have already been developed to address some of these issues such as EPANET and KYPipe [14] [15]. EPANET was developed by the Environmental Protection Agency as a way to perform extended period simulations of hydraulic and water quality behavior within a pressurized pipe network. This software tracks the flow of water in pipes, the pressure at nodes, the height of water in each tank, and the concentration of chemicals throughout the network. The software can also simulate water age and source tracing. KYPipe is similar to EPANET in modeling the hydraulic properties of a pipe network. It offers a graphical user interface (GUI) which allows users to develop pipe system models easily. It can provide hydrant flow calculations and water quality analysis in addition to many other features.

Figure 2.8: KYPipe Simulation GUI

These software packages focus on the flow of water throughout the distribution system in order to model water quality and hydraulic flow. These models and simulations help researchers who are investigating methods to prevent and analyze how chemicals can transverse the pipe network in the event that toxins were intentionally introduced into the water system in order to cause harm to customers.

EPANET and KYPipe are excellent modeling packages for this type of research, but do not address the specific needs required for testing prototypes directed at preventing cyber attacks on the equipment itself. They only provide a numerical analysis and output and do not actually address the system operation such as control of pumps and sensor data throughout the network. They also do not provide any interface to actual equipment controlling the system hardware, the HMI, or connect to the communication network in any way.

The National SCADA Test-Bed program was established by the Department of Energy (DOE) at Idaho and Sandia National Laboratories. This test-bed was designed to

address issues with cyber security in the nation's electricity, oil, and gas industries. There has also been work in the development of test-beds that incorporate both a virtualized component that works in unison with a hardware component. These simulations allow for most of the intelligence and processing to occur within the software while interfaces to physical SCADA components is accomplished with additional hardware [16] [17] [18].

In order to develop a simulation that can be used to test the security preprocessor developed at the University of Louisville as well as other prototypes, a simulation is needed that incorporates components of EPANET and KYPipe such as flow throughout the pipe network and tank levels, but must also include control and simulated operation of pumps and valves in the system, and can interface with actual SCADA components. Also, these I/O parameters and analysis for the distribution side of the system must be carried into the treatment side in order to form a better simulation. The combination of all of these components will provide a robust model software package that can be easily configured and reconfigured for testing purposes.

III. MODEL DEVELOPMENT

Due to the complexity of water systems, model development began by clearly identify the necessary functions required for testing the target security device. Trying to model all aspects of the treatment and distribution process would be extremely difficult and beyond the scope of this project. To reduce the complexity, several assumptions are made which are outlined for each subset of the water system. Although the systems are explained separately, the final model includes all described.

A. Water Treatment

1. Calculations and Parameters

As outlined in Section 2.1, the water treatment process is normally comprised of five steps: coagulation, sedimentation, filtration, disinfection, and water storage. In addition to this, there must be a mechanism to bring water from the fresh water source (i.e. lake, river) to the treatment facility. This is accomplished through the use of pumps and water lines. Most of the processes in the treatment phase are time oriented, and their importance to the overall security of the system is less. There are a limited number of processes which are needed to be included in the model in order to provide adequate operations for testing. These are:

1) Provide a uninterrupted supply of fresh water to the treatment facility as needed

2) Provide chlorine to the filtered water for disinfection

3) Monitor the chlorination level of the water

4) Flow chlorinated drinkable water into the distribution system

Figure 3.1 shows the system components for the treatment system. This model assumes that water entering the treatment facility has passed through the coagulation, sedimentation, and filtration stations without issue. A reservoir of filtered clear water is stored for disinfection and then held until it is needed in the distribution system. In addition, the fresh water source and the chlorine source is assumed to be infinite; that is the water source will not empty as water is pumped into the treatment process and chlorine will always be available for disinfection. All calculations are performed every simulated one minute, which is defaulted to every one second in true time. This simulation interval can be changed.

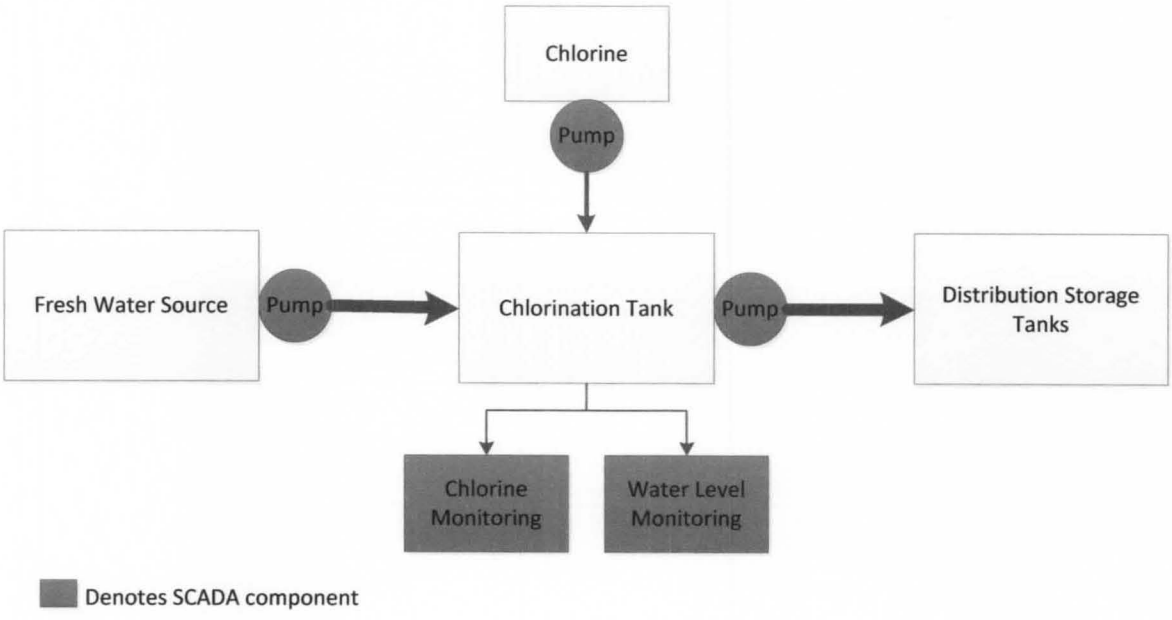


Figure 3.1: Water Treatment Model Flow Chart

Variables are assigned for the flow rate of the pumps in and out of the chlorination tank and the capacity of the tank. The volume of water (gallons) in the tank is calculated based on its current volume, the flow of water in (flow rate of pump), and the flow of water into the distribution system.

$$V_{water} = V_{current} + (V_{flow\ in} - V_{flow\ out})t$$

Tanks in treatment facilities are usually in-ground with rectangular prism geometry. Thus, the height of water in the reservoir can be calculated by

$$h_{water} = \frac{V_{water}}{A'_{tank}},$$

where V_{water} is the volume of water in the tank in cubic feet (1 ft^3 = 7.4805 gallons) and A'_{tank} is the cross sectional area of the tank *(l x w)*.

Various forms of chlorine are added to filtered water in order to kill micro-organisms. The levels of chlorine are strictly regulated by the U.S. Environmental Protection Agency under the Safe Drinking Water Act of 1974 [19]. Table 1 shows the maximum amount of chlorine allowable in drinking water.

Table 1: Standards on Chlorine Levels [19]

Disinfectant	MRDLG	MRDL
Chloramine	4 milligrams per liter (mg/L) or 4 parts per million (ppm)	4.0 mg/L or 4 ppm as an annual average
Chlorine	4 mg/L or 4 ppm	4.0 mg/L or 4 ppm as an annual average
Chlorine Dioxide	0.8 mg/L or 800 parts per billion (ppb)	0.8 mg/L or 800 ppb

From this rate the ideal concentration of chlorine between 3.5 and 4 ppm can be set. The chlorination rate of the system is variable to maintain an acceptable chlorine level as the flow rate of water in and out of the tank changes. Assume that the chlorine is uniformly distributed throughout the water tank. Thus, the concentration of chlorine in the water leaving the tank is equal to the concentration of the entire tank so that the concentration of chlorine does not change based on the outflow. Also assume the concentration of the filtered water entering the tank to be zero. The new chlorine concentration of the tank is then calculated to be

$$Cl_{concentration} = \frac{M_{Cl\,old}}{V_{water\,old}} + \frac{M_{Cl\,new}}{V_{old+new}},$$

where $M_{Cl\,old}$ is the current amount of Cl (mg) in the water, $V_{water\,old}$ is the amount of water in the tank minus any that was pumped into the distribution system, $M_{Cl\,new}$ is the amount of Cl(mg) added during the time frame (chlorination rate), and $V_{old+new}$ is the volume of water in the tank minus any outflow into the distribution system and adding any inflow (flow rate of pump). The chlorination can be configured to automatically change based on concentration levels or set to be manually updated by the operator.

2. Required Model Inputs and Outputs

There is a limited amount of equipment required for the operation of the treatment component of the simulation. A reservoir pump is required to provide water into the treatment facility for treatment. A valve is also required at the pump site in order to prevent backflow. A second pump is required to pump chlorine into the filtered water for disinfection. A level sensor is also required in order to measure the amount of water in the treatment tank.

Corresponding parameters or variables are required for each of these components. Each pump has a flow rate which should be configurable by the user. The maximum tank level and ideal chlorine levels are also configurable by the user. Each calculated value can be set to a default prior to running the simulation. Thus, the current treatment tank level can be set to any value less than or equal to the maximum tank level parameter. Each of the control (pumps, values) and sensor (tank levels) parameters has a corresponding physical connection to the RTU, whether it is a digital or analog signal. These connections are more thoroughly described in Chapter 4.

B. Water Distribution

1. Calculations and Parameters

Distribution systems vary widely based on the number of customers the water system serves, as well as the geographical layout of the area. The number of tanks, pumps, water lines, and even the topology of the water lines is independent to a specific system. The distribution model needs to be robust enough to allow for complete testing of the hardened security device while not being redundant in its features. Several assumptions are made in order to accomplish this task. First, a branch architecture system was chosen. This allows for pressure calculations at various points to be performed easily. From the treatment model, it can be assumed that a separate storage tank for treated water is not needed before being sent to distribution storage tanks. For simulation purposes the pump and water lines leaving the chlorinated water tank are the same as those that pump into the distribution system. Modeling a separate storage system

will not improve the simulation. It was determined that two elevated water storage tanks and their associated pumps and valves would be adequate for the distribution model.

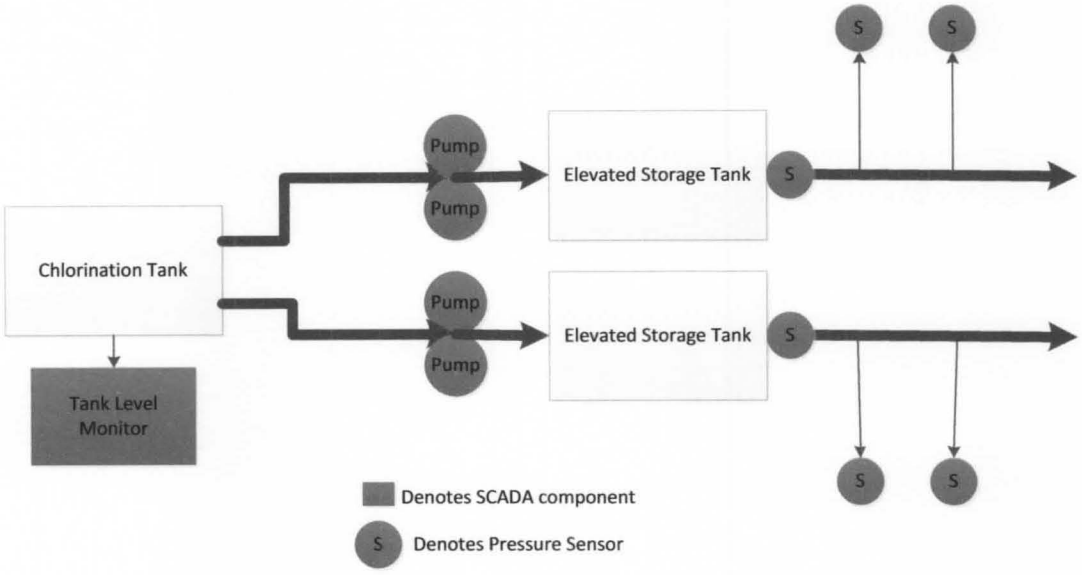


Figure 3.2: Distribution Model Flow Chart

Variables are assigned to flow rates of each pump in the system and for the capacity of each storage tank. Each tank is assumed to be a cylinder of varying height. Thus to calculate the capacity (volume) of the tank the equation is:

$$V_{cylinder} = \pi r^2 h,$$

Where r is the radius of the tank and h is the varying height based on the capacity variable assigned.

Water systems do not measure the amount of water in the storage tank in gallons, but instead in feet of water in the tank. The height of the water in the tank can be measured by determining the static head pressure of the water using a sensor located in the bottom of the standpipe of the tank. The pressure at any given point is dependent only on the height of the column of water, and independent of the total volume of water

(in this paper height of water is defined as the difference in elevation between the top of the water column and the pressure sensor). For this reason set the radius of the tank to a constant value and only change the height of the tank in order to increase capacity for the model. Since the specific gravity of water is known, a linear relation can be made between the measured pressure (psi) in the stand pipe and the height of the water in the tank.

$$h_{water} = 2.31\ ft/psi$$

Flow in and out of the tank is measured in gallons/min. Thus, calculate the volume of water in the tank based on its current volume in gallons, the flow of water into the tank based on the flow rate of the pumps, and the flow of water out of the tank based on demand.

$$V_{water} = V_{water\ current} + (V_{flow\ rate\ in} - V_{flow\ rate\ out})t$$

The cross-sectional area of the tank is set so the pressure can be inversely calculated by

$$P = \left(\frac{V_{water}}{A'_{tank}}\right) C,$$

Where P is the pressure read by the sensor in psi, V_{water} is the volume of water in the tank in cubic feet (1 ft^3 = 7.4805 gallons), A'_{tank} is the cross sectional area of the tank in square feet, and C is a constant conversion factor between pressure and height of water (1 psi / 2.31 ft).

2. Required Model Inputs and Outputs

Based on the assumptions discussed previously, the equipment required for the distribution component includes four pumps and four valves, two storage tanks and their associated level meters. As with the treatment component, all of the parameters for pump flow rates, maximum tank levels, and the default or starting values for current tanks levels can be defined by the user prior to running the simulation. Also, each of these parameters has a physical connection to the RTU used in testing as outlined in Chapter 4.

C. Water Demand

Several factors are considered to determine water usage in the system, 1) the number of customers served, 2) the time of day the usage is occurring, 3) the number of high use peripheral devices in the system (i.e. fire hydrants). Variable are assigned for the number of customers for each branch of the distribution system, each being independent of the other. For time of day data, a 24 hour demand curve, shown in Figure 3.3, was taken from previous work completed on a similar simulation [20]. Usage is extracted from this curve and used to calculate current demand along with the number of customers in each branch.

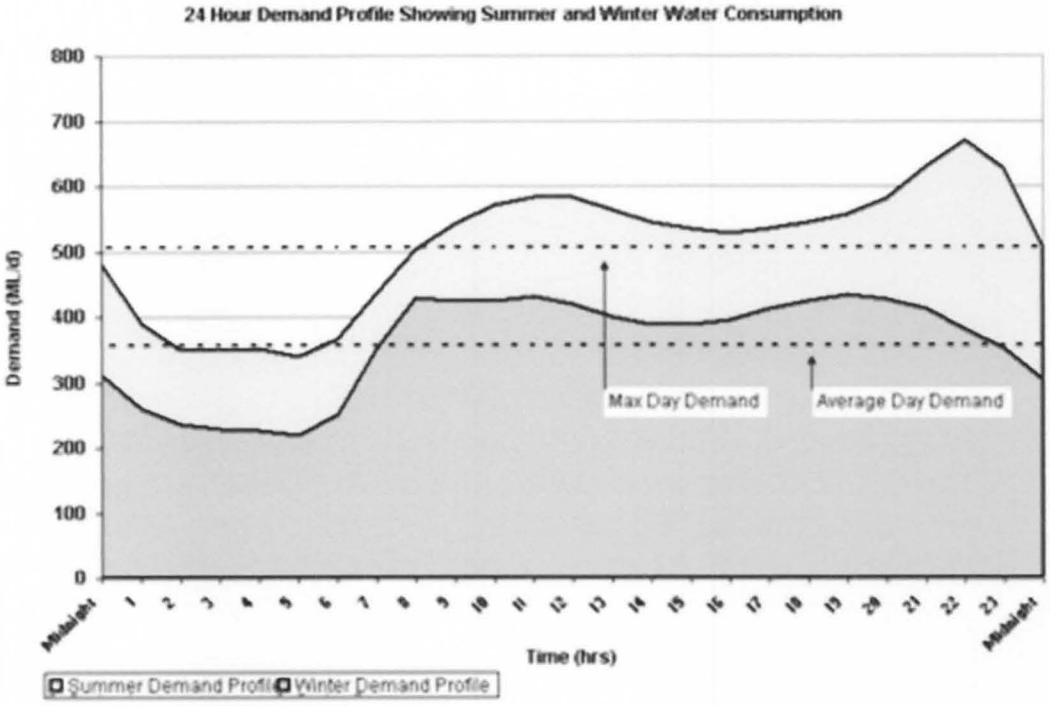


Figure 3.3: Time of Day Demand Curve

For this basic 24 hour model, demand in the system is calculated to be

,

where D is the demand for each branch, N is the number of customers, and is the coefficient of usage based on the time of day. This graph, shown in Figure 3.3, also distinguishes winter from summer months. However, seasonal information is not currently used in the model.

High use devices such as fire hydrants place additional usage on the system. These devices are random instantaneous surges on the demand of the distribution system and are not time varying. To model these, a simple Boolean function is implemented telling the system whether they are on or off. This generates peaks in the demand.

Pressure levels throughout the system are also an important consideration for a simulation. For the customer, acceptable water pressure ranges between 40-70 psi. The pressure throughout the system is determined based on the pressure at the water tower, and the losses due to friction through pipes, valves, elbows, and meters. To simplify the model we can assume straight pipes and no loses at branch points. This can be done because line breaks and tank levels are the most important pressure aspects to consider from a security aspect. Also, pipes are assumed to be at zero elevation with respect to the pressure sensor at each tank. Calculations are made assuming uniform pipe types and pipe diameters, thus friction loses are only dependent on distance from the tank.

IV. SIMULATION SOFTWARE

Previous work on a water treatment and distribution system had been completed at the University of Louisville. This work was developed for earlier versions of a security device. The model was similar, including control of pumps and valves, chlorination levels, and tank levels throughout the system. However, the hardware used to interface to the RTU used in testing had limited IO, thus limiting the functionality of the software simulation. The simulation also used two different programming languages to create the full simulation [20]. In order to improve on this work, the simulation was written in a single language, Microsoft Visual C#. This was chosen due to its ease of integration with external hardware and familiarity with the programming software. New hardware was also developed which described in Chapter 5. These changes allowed for a more robust simulation to be developed.

The models outlined in Chapter 3 were implemented in C# code using Microsoft Visual Studio (MVS) 2010. The designer feature of MVS allowed for a user-friendly graphical user interface (GUI) to be developed that displayed all important calculations and statuses of the simulation. The user interface also allows for simulation variables to be entered as discussed in the following section. These values can be modified as the simulation runs.

1. Simulation Parameters and Controls

The first component of the simulation contains the basic parameters required to run the simulation. These include:

1. Hour of Day
2. Minute of Day
3. Customers
4. Belknap Outflow
5. Medical Outflow

The hour and minute of day is defaulted to 0 but can be changed to any acceptable value before the simulation is started. As the simulation runs, the minute of day will increment once per second as defined by a timer running in the simulation. After 60 seconds in real time (60 minutes in the simulation), the hour of day will increment by one. This timing sequence can be changed by changing the timer settings in the code to allow a simulated minute and hour to be configured to any real world timing interval. The timing structure of the simulation is important if the model needs to replicate real world events in which the timing mechanism is crucial. If the simulation needs to run in real-time, the timer in the simulation should be set to true time or a 1:1 ratio. This would correspond to a timer tick every 60 seconds. For simulations were the chlorine concentration and rate of addition is crucial this timing scheme should be used.

The number of customers is also defaulted to 0, but should be changed prior to running the simulation. The number of customers and time of day are used to calculate the outflow for the two distribution subsystems, Medical and Belknap. This naming

convention is used for the two downtown campuses of the University of Louisville. The outflows are calculated using the formula outlined in Chapter 3.

The *Parameters* section also contains the simulation controls. These include:

1. Connect
2. Disconnect
3. Run Model
4. Stop
5. Reset

The front panel for the *parameters* section is shown in Figure 4.1. The connect button opens the serial port defined in part by the user and within the code of the simulation software. The disconnect button closes this port. The COM port number can be entered by the user into the COM Port textbox available in the parameters section. The default value is "1"; indicating "COM1" will be addressed when establishing a connection. All of the other parameters such baud rate, parity, stop bits, and flow control are defined within the code of the software and cannot be easily changed by the user. This is to prevent modifications of these parameters which will prevent a correct connection with the hardware. In order to run the simulation, the communication port must be open. The simulation uses the System.IO.Ports namespace in Microsoft Visual Studio C#, which contains classes needed to establish a serial communications connection to the external hardware.

ASCII characters and sequences of characters are passed between the software and hardware. These characters are used to initialize hardware settings, and exchange the status of pumps, tank levels, chlorination levels, and any other important information between the two components of the simulation. The COM value textbox displays the last

ASCII sequence received from the hardware. This can be used for troubleshooting issues with the simulation or for simply monitoring the exchange of data between the software and hardware. The simulation is configured to only run once a connection is established, i.e. the simulation control buttons (RUN, STOP, and RESET) are not enabled unless a connection is made.

With a connection established, the run button becomes enabled. When pressed, an initialize character is sent from the software to the hardware, which resets any analog or discrete value to their defaults. The hardware then responds with the status of all of the discrete inputs from the RTU which are updated in the software. After initialization, the model timer starts which causes the time of day values to increment and all other calculations to occur. Calculations are updated every simulated minute. The Stop button stops the model timer and all calculations. Reset sets all analog values in the simulation to their defaults and sends a new initialization character to the hardware.

Figure 4.1: Parameters and Simulation Controls

2. Water Treatment Component

The water treatment component is comprised of two simulated pumps: one for pumping water into the treatment tank (RES Pump) and one for pumping chlorine into the treatment tank. The reservoir pump includes a valve that opens and closes as the pump turns on and off. All of these components have a Boolean logic which is transmitted from the hardware to the simulation via the communication port. When the pumps are active, the pump symbol and valve change to a green color; when there are not active they are represented by a red symbol.

Using the formulas outlined in Chapter 3, the tank level and chlorine level in the tank is calculated when the simulation is running. The max tank level, current tank level, and pump flow rates can be adjusted by the user before the simulation runs. The default levels are shown in Figure 4.2. Chlorine level and tank level are analog outputs from the simulation software to the hardware which can be read by the RTU and HMI if desired.

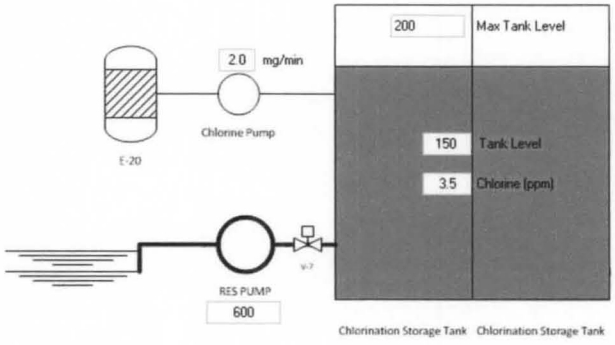


Figure 4.2: Water Treatment Simulation Component

3. Water Distribution Component

The water distribution component is comprised of four pumps and four valves, flow parameters for each of the pumps, and two tanks and their associated tank levels.

The tank levels are calculated using the formulas outlined in Chapter 3. The pump statuses are a Boolean logic controlled by the simulation hardware connected to the RTU. When the pumps are active, the pump symbol and valve symbol change to green; when there are not active they are represented by a red symbol.

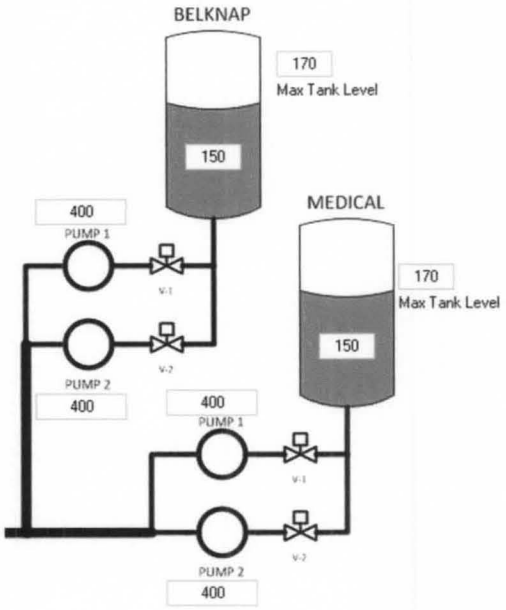


Figure 4.3: Water Distribution Simulation Component

4. Distribution Adjustments and Pressure Values

An important aspect to the simulation is to be able to adjust the normal operating conditions of the water system. This section allows for instantaneous spikes in outflow from each distribution subsystem as well as monitoring of pressure in the lines and a given distance from the elevated tank. As described in the previous chapter, assumptions are made that the entire pipe network is distributed on an even elevation and with a given pipe diameter and type. Thus, the pressure drop can be calculated as described using a simple linear reduction based on distance. These parameters cannot be changed via the

GUI by the user, but can only be changed within the code of the simulation. Each subsystem allows for two virtual pressure nodes to be placed at any distance from the tank. Fire hydrants provide a method of creating an instantaneous surge in outflow and the flow rate of the hydrant can be modified by the user. The pressure level at each node is an analog output from the simulation software to the hardware and can be read by the RTU and HMI if desired.

This section also contains four control buttons that can create simulated breaks in the pipe network at each pressure node. If the break button is used, the pressure at that node drops to 0. The corresponding analog output will also read 0. This can be used by the HMI to signal to the operator that there is an issue in the system.

Figure 4.4: Distribution Adjustments and Pressure Values

The completed simulation GUI is shown in Figure 4.5

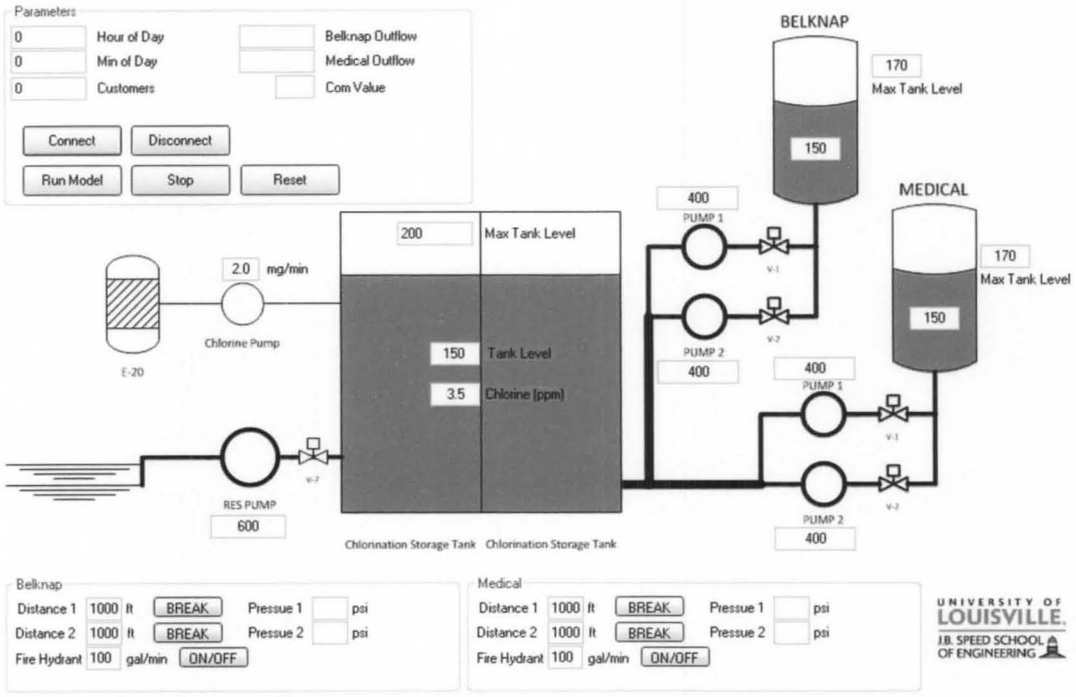


Figure 4.5: Graphical Interface of Treatment and Distribution Simulation

2. Human Machine Interface (HMI)

In order for the RTU to function appropriately and the simulation to be used, an HMI is needed to provide a method of controlling the RTUs outputs and reading the inputs. As described in Chapter 2, the common SCADA protocol for communicating with most RTUs is the MODBUS protocol. This is the case for the VersaTRAK used in this research, which allowed for both MODBUS RTU and MODBUS ASCII, two standard forms of the protocol. An HMI is needed which can communicate via these standards. Many off the shelf products were available, including several freeware options.

After testing the functionality and adaptability of some of these options, it was determined that a customized HMI could be developed using National Instruments

Laboratory Virtual Instrumentation Engineering Workbench or LabVIEW®. LabVIEW is a graphical, 'G', programming language that is used to expedite the development of programs for data acquisition, instrumentation control, and industrial process control. A MODUBUS virtual instrument (VI) is available as an add-on which reduced the amount of time to develop an HMI for use with the simulation and RTU. The HMI uses a VISA resource subVI to establish a connection to the RTU using the appropriate parameters. The HMI references this resource in the front panel as shown in Figure 4.6.

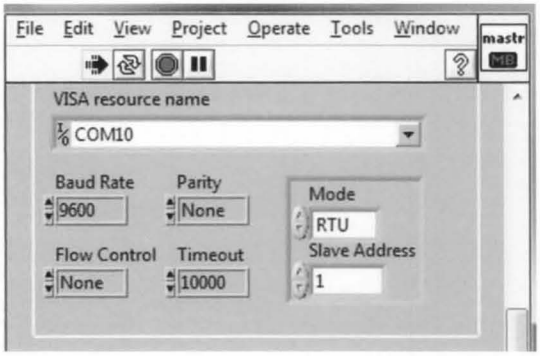


Figure 4.6: VISA Resource Parameters

Modifications to the existing interface have been made to increase the number of digital output controls as well as to read multiple input registers from the RTU which store the analog data from the simulation hardware. Also, the number of digital inputs has been expanded in order to match those of the RTU. Figure 4.7 shows the Boolean control buttons on the front panel of the HMI for the four coils used on the RTU. Figure 4.8 shows the twelve digital input indicators associated with the digital inputs of the RTU.

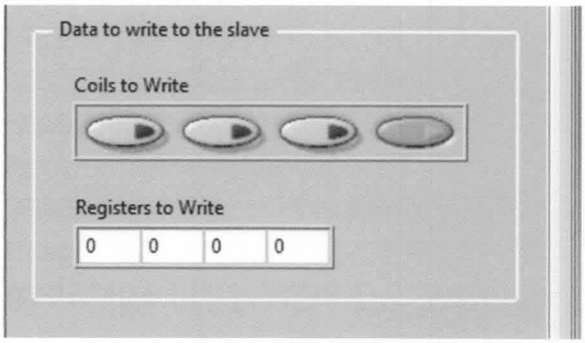


Figure 4.7: Digital and Analog Output Control

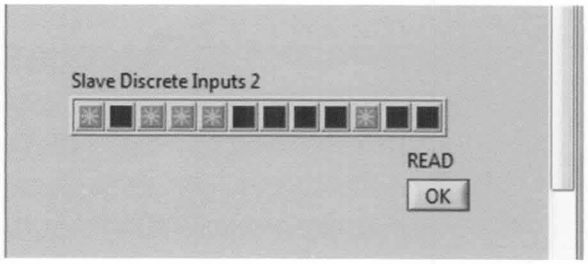


Figure 4.8: Digital Inputs from RTU

V. SIMULATION HARDWARE

This chapter describes specialized hardware which was designed, developed, and tested to support a more robust simulator for SCADA security evaluations. The hardware works alongside the software simulation component but was designed so that it can interface with other software developed for similar tasks easily.

1. RTU and I/O

The second component to the simulation is the hardware, which is used to interface to the RTU used for testing of the hardened security device. For this research project, the VersaTRAK® mIPm RTU/Controller is used. It is equipped with 12 discrete inputs, four discrete outputs, eight analog inputs, and two analog outputs. The RTU is connected to a computer equipped with a HMI, which is used to turn on digital outputs and display the status of both digital and analog inputs. The security of the communication between the HMI and the RTU is the area of interest for security testing.

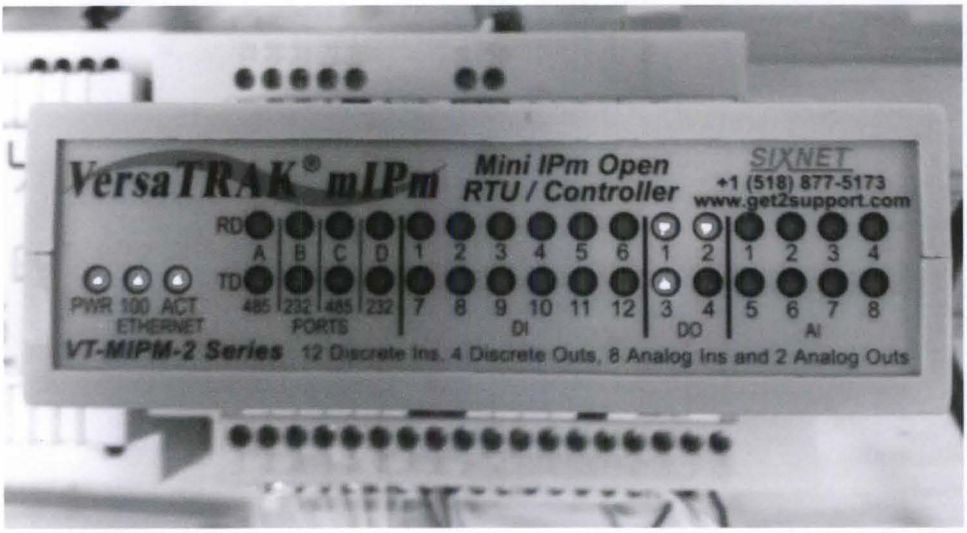


Figure 5.1: RTU used for Testing

In a water treatment and distribution system, the RTU will be the controller connected to the relays, which turn pumps on and off, close valves, and to sensors that monitor tank levels and other system conditions. The purpose of the simulation is to replace the pumps, valves, and sensors, with virtualized components, which are accomplished in the software portion of the simulation. In order to interface with the RTU; however, additional hardware is needed.

The types of inputs and outputs, both discrete and analog, need to be assessed before any hardware design begins. Reviewing the simulation and models outlines, it is determined that the I/O shown is Table 5.1 is required. Input/output listed in the table is from the point-of-view of the simulation hardware that interfaces with the RTU. It is worth noting that the specific RTU used in this research is not equipped with enough digital outputs (coils) or analog inputs to accommodate all of the variables used in the simulation. Instead of reducing the functionality of the simulation, the physical connections between the simulation hardware and the RTU will simply be left

unconnected. In the event a new RTU is acquired, the connections can be made and the simulation can be fully controlled.

Table 5.1: Simulation Hardware I/O

Description	Input / Output	Analog or Discrete
Pump 1A Belknap	Input	Discrete
Pump 2A Belknap	Input	Discrete
Pump 1B Medical	Input	Discrete
Pump 2B Medical	Input	Discrete
Tank Belknap Level	Output	Analog
Tank Medical Level	Output	Analog
Treatment Tank Level	Output	Analog
Chlorine Concentration	Output	Analog
Res Pump	Input	Discrete
Chlorine Pump	Input	Discrete
Pressure Node1 Belknap	Output	Analog
Pressure Node 2 Belknap	Output	Analog
Pressure Node 1Medical	Output	Analog
Pressure Node 2Medical	Output	Analog

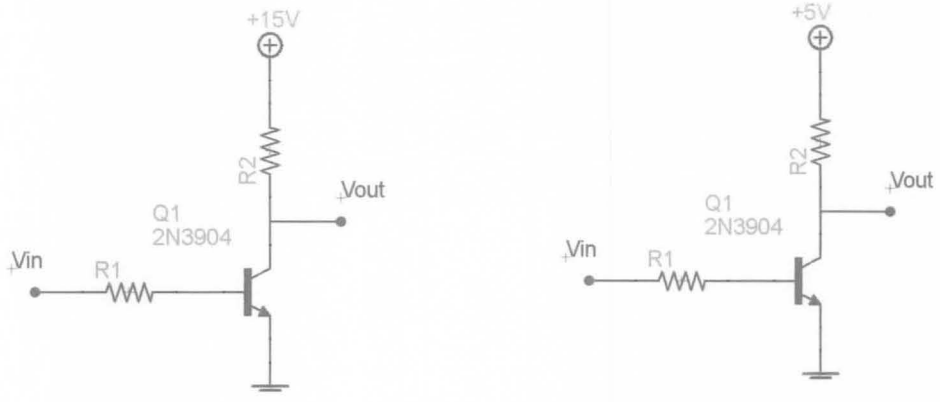
2. Hardware Circuitry and Controller

With the required I/O determined, and a method for controlling the RTU developed, the hardware required to interface from the simulation software to the RTU had to be developed. This hardware needed to be able to accomplish the following tasks:

1. Communicate over a RS-232 serial connection to the software
2. Store analog data from the simulation (i.e. tank levels, line pressure, chlorination levels)
3. Output analog signal to the RTU based on stored analog data from software
4. Read digital output signals from the RTU and store as Boolean variables
5. Send Boolean values from (4) to software via communication port
6. Output digital signal to RTU (for future use)

In order to accomplish these tasks, an embedded controller was required. A Microchip® PIC18F8722 microcontroller was used in order to process discrete and analog values from the RTU and communicate them to the software simulation via a RS-232 serial connection. The microcontroller was also used to process serial data received from the software simulation and output discrete and analog values to the RTU. The microcontroller serial port interfaced to a MAX232 TTL level-shifter in order to match appropriate signal voltage levels between the microcontroller and computer. The serial connection was full-duplex enabling the microcontroller to send and receive data simultaneously.

Additional electronics and circuits were required in order to interface from the microcontroller to the RTU. The input voltage range on the discrete I/O for the microcontroller was 0-5 V and the range on the RTU was 0-15 V. In order to prevent over voltage on the microcontroller and ensure a minimum threshold voltage for the RTU, a transistor switching circuit was used.



a) b)

Figure 5.2: Transistor Switch Circuits

Figure 5.2 shows the two transistor circuits used in order to convert the digital output voltages between the RTU and microprocessor. Figure 5.2a was used for signals originating at the microprocessor and figure 5.2b was used for signals originating at the RTU. In both cases, V_{in} was connected to the output pin of the originating device. When V_{in} was low (or 0V) no current flowed from the collector , which was connected to R2 in the figure, and the emitter, which was connected to ground. Thus, there is no voltage drop across R2 and the output voltage V_{out} is equal to the supply voltage, 15V or 5V depending on the case. When V_{in} is high, the base current becomes non-zero so the collector current becomes non-zero as well. For a 2N3904 npn transistor, the voltage drop across the base and collector is 0.7V. Thus the voltage drop across R2 is equal to the supply voltage minus 0.7V. The output V_{out} is then equal to 0.7V which is equivalent to a logical 0 or off. The transistor switch operates on an inverting topology [21].

The PIC18F8722 did not include any analog outputs, similar to most microcontrollers. In order to interface the simulation to the RTU, a method of generating analog outputs for the tank levels, chlorine levels, and line pressure had to be developed. It was determined that there were two methods to accomplish this goal.

The first option utilized the microcontroller's pulse-width modulation (PWM) modules. PWM is a common method used for digital-to-analog conversion (DAC). PWM is the generation of a series of pulses at a fixed amplitude, period, and frequency. The duty cycle of the signal is defined by the width of each pulse which can be varied. Figure 5.3 shows a PWM waveform. By adding a low-pass filter to a PWM signal, an analog output signal can be generated. The output voltage is proportional to the average time the PWM signal is spent in the "HIGH" state. For example, a 50% duty cycle would

correlate to a 2.5V analog output if the supply voltage was 5V as in the case with the microcontroller used in the hardware. A passive filter is adequate to produce an acceptable output if the frequency of the PWM signal is high enough. If not, an active filter design can be used [22].

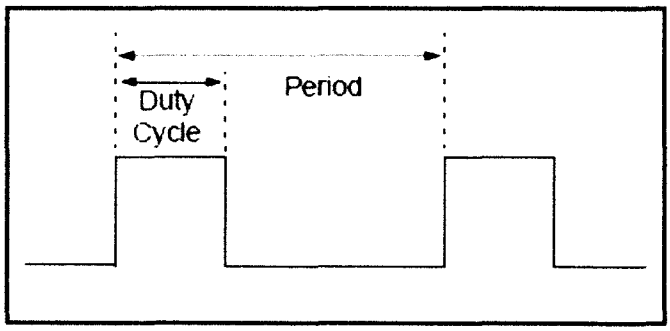


Figure 5.3: Pulse Width Modulation Waveform

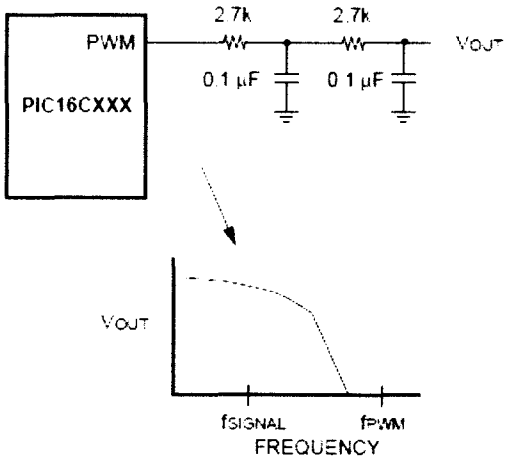


Figure 5.4: DAC using PWM and Passive Filter

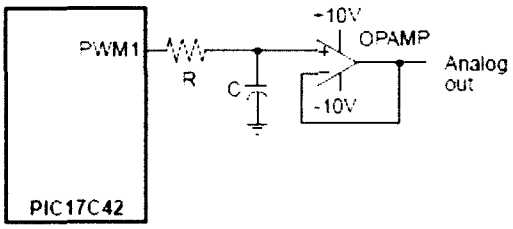


Figure 5.5: DAC using PWM and Active Filter

A second method to creating an external DAC is to use a resistor network which is driven by digital outputs [22]. This method is often referred to as an R2R ladder. In a typical application, eight discrete outputs are wired to a ladder configuration of resistors, with the resistors wired in series with the output pins being 2X the value as the rail resistors. This provides an equivalent 8-bit DAC. Figure 5.6 shows a single port R2R configuration with a microcontroller similar to the one used in this project. Figure 5.7 shows the output response of the R2R ladder.

The PIC18F8722 has nine I/O ports, each 8 bit, or equivalently, 72 possible digital I/O. Since a single port can support the necessary digital inputs required for the simulation, the remaining eight can be used for analog outputs, exactly the number of analog inputs available on the RTU. For this reason, and the limited number of PWM modules on the PIC, the R2R ladder options were used for this project.

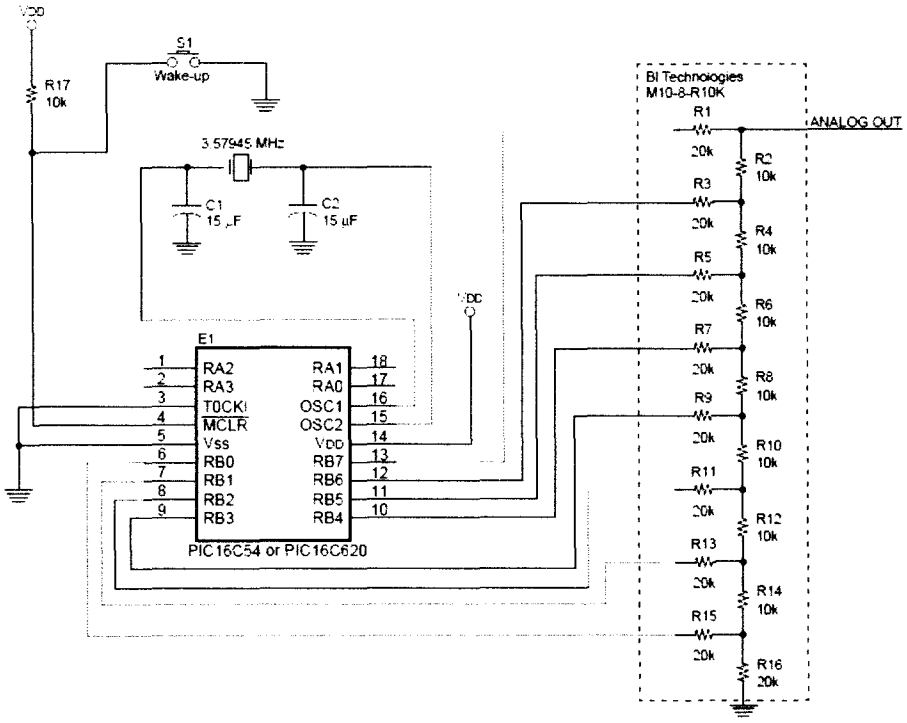


Figure 5.6: Single Port R2R Ladder

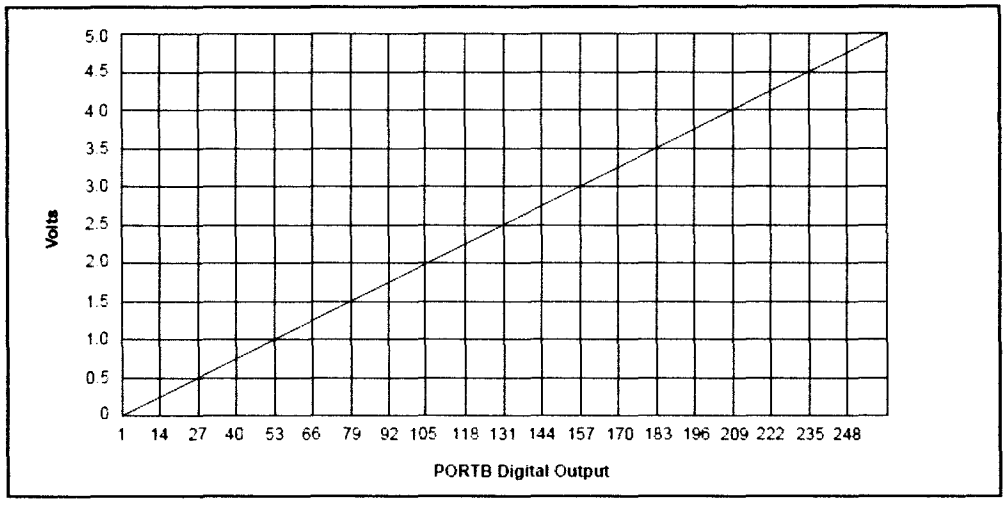


Figure 5.7: Output Response of Single Port R2R Ladder

Although the simulation does not currently include reading analog outputs from the RTUs two A0 ports, consideration was made in adding this option. The

microcontroller does have 16, 10-bit analog-to-digital converters (ADC) that can operate with a single-ended or differential voltage reference. However, as with most industrial controls equipment, the RTU used a 4-20mA current loop for analog outputs. In order to convert from current to voltage an operational amplifier circuit is required.

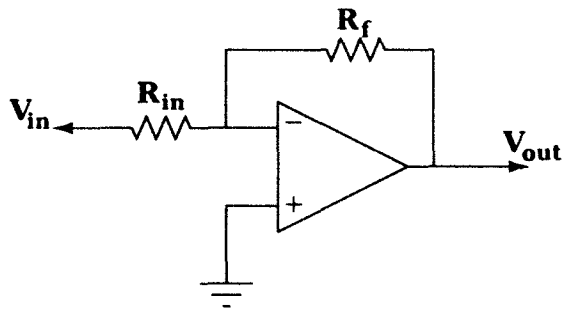


Figure 5.8: Current to Voltage Converter

Figure 5.8 shows the current to voltage converter circuit required. V_{in} is connected to the RTU. As the current from the current loop increases, V_{out} increases. This allows the microcontroller to have a voltage input based on a current output from the RTU. R_f must be a precision resistor for this circuit to function properly but the exact resistor values can vary due to the high impedance of the RTU I/O. [21]. Figure 5.9 shows the overall conversions and connections to interface the RTU and microcontroller. Figure 5.10 shows a simplified schematic of the microcontroller and additional electronics.

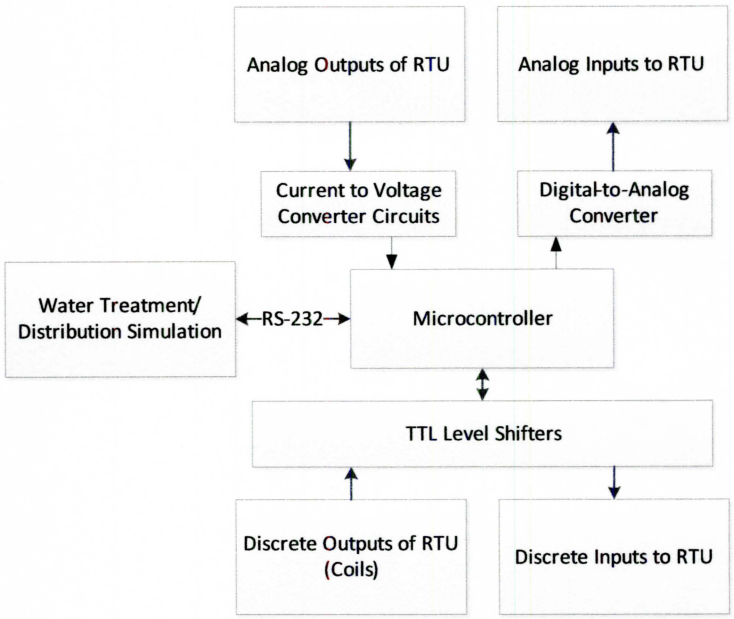


Figure 5.9: Block Diagram of I/O Interfaces with Microcontroller

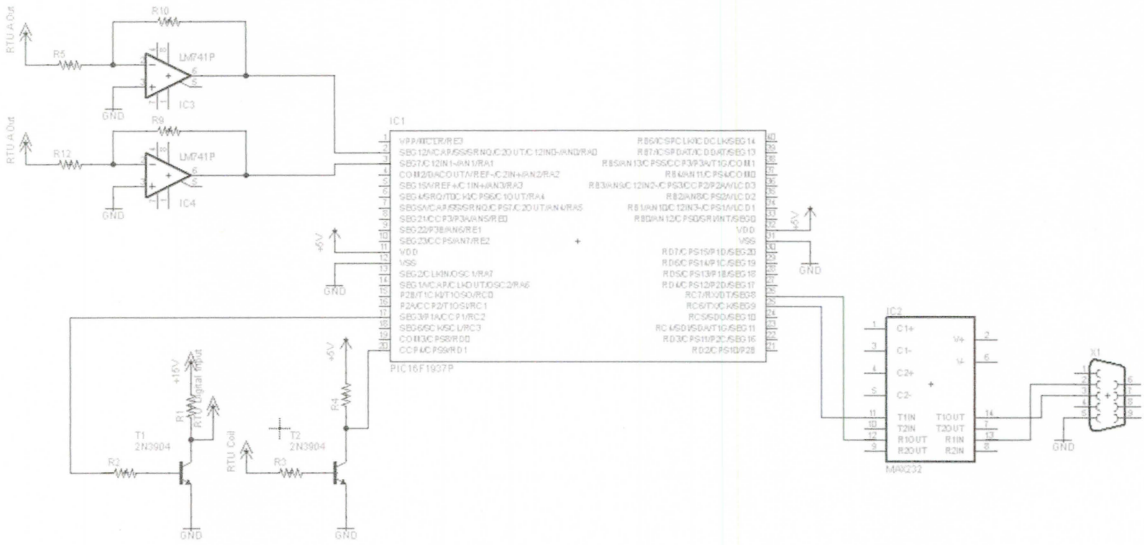


Figure 5.10: Simplified Schematic of I/O and Communication Circuits with Microcontroller

An additional software component was needed in order to program the microcontroller to process data received from the simulation software and on I/O pins connected to the transistor switches. A C compiler from CCS Inc® was used to program and load the code onto the microcontroller. This code was simplistic, toggling output

pins when specific ASCII characters were received on the serial port, and sending ASCII characters to the simulation depending on the state of its digital input ports. A PIC18F8722 development kit from CCS was used for compatibility with the programming software and ease of integration.

VI. TESTING

This chapter presents results from testing the simulator for proper operations, and also some representative SCADA security testing indicative of have the simulator will be used.

A. Software Simulation Testing

There are several items that were tested in order to check the functionality of the water treatment and distribution simulation. Testing included the simulation software, simulation hardware, the VersaTRAK RTU, and the LabVIEW HMI. The simulation software was run on a Windows 7 PC located in the Information Security Research Laboratory at the University of Louisville. The software was initially run in Microsoft Visual Studio so that any issues with the program could be debugged. After completion, an executable file was generated. Testing of the simulation was divided into the following test criteria:

I. Communication setup

II. Model calculations without external inputs (i.e. time of day, outflow)

III. Software indicator functionality from hardware outputs (i.e. pump status)

IV. Model calculations with external inputs (i.e. tank levels with pumps running)

V. Hardware signal generation from software variables (i.e. analog outputs to RTU)

1. Serial Port Connection Testing

The first aspect to testing the simulation software is to make sure that a connection can be made on the COM port specified in the parameters section. In cases where the PC has an available COM port on board, typically COM1, this can be accomplished by simply connecting a serial cable from the COM port located on the back of the machine to the DB9 connector available on the simulation hardware breadboard and verifying that the COM port number entered in the simulation matches the available port of the machine. In situations where the PC does not have a standard serial port, a USB to serial converter cable can be purchased. Once the physical connection has been made and power is provided to the hardware, the connect button can be pressed and the simulation control buttons will become enabled.

Figure 6.1: Control Buttons before Connection Established

When the connect button is pressed, the software will send a sequence of ASCII characters to the simulation hardware. The hardware will then respond with a sequence of ASCII characters which will be displayed in the COM value textbox of the software. This will indicate that a successful connection has been established. If the connection fails, an error message will appear on the screen.

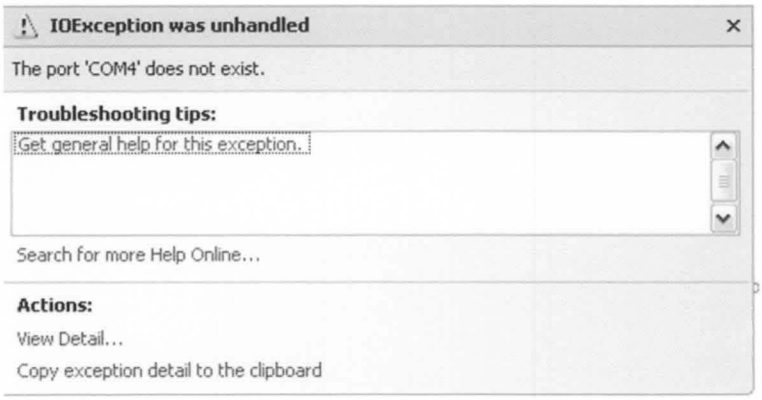


Figure 6.2: Example Error Message during Connection Process

2. Model Calculations without External Inputs

With a successful connection established, the remaining elements of the simulation can be tested. Pressing the "RUN" button will start the simulation and enable the "STOP" and "RESET" control buttons. As soon as the simulation begins, the "Min of Day" text box should begin to increment each second. To test the simulation model calculations without inputs from the hardware, the inputs to the transistor switching circuits were disconnected from the RTU, generating an "OFF" status for all of the pumps. This test examined the functionality of the model timer, and calculations of the outflow based on time of day and number of customers, the treatment tank level, and chlorine level in the tank.

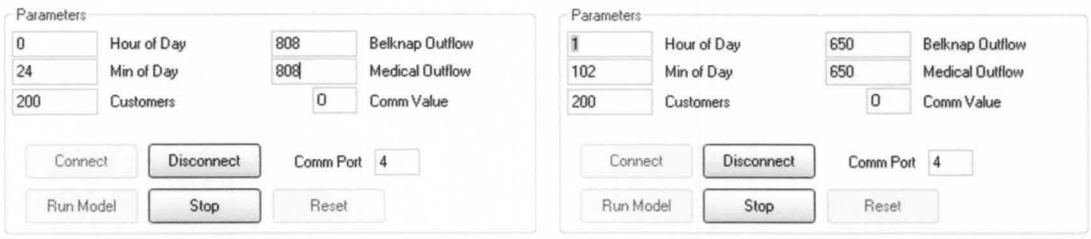


Figure 6.3: Outflow Calculations

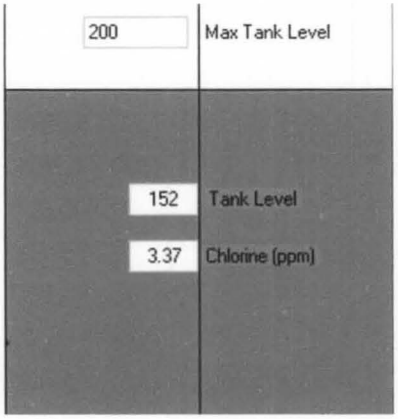


Figure 6.4: Chlorine and Tank Level Calculation

3. Software Indicators

The status of the pumps in the simulation is dependent on the inputs to the simulation hardware from the RTU. When a discrete output of the RTU is turned on using the HMI, the transistor switching circuit will pull an input pin on the simulation microcontroller low. The microcontroller will then send a character indicating the status of that pump which will change the color of the pump and value in the software. If the pump is running, the tank level will change in accordance with the formulas outlined in Chapter 3. As shown in Figure 5.5, turning on discrete outputs 1-3 turn on one pump of the Belknap Tank and both pumps of the Medical tank. The forth output is connected to the reservoir pump. As noted previous, the specific RTU used in testing is not equipped with enough discrete outputs to take full advantage of the simulation. Testing the other digital input based components of the hardware and software was completed by pulling the transistor switching circuits "HIGH" and "LOW" directly.

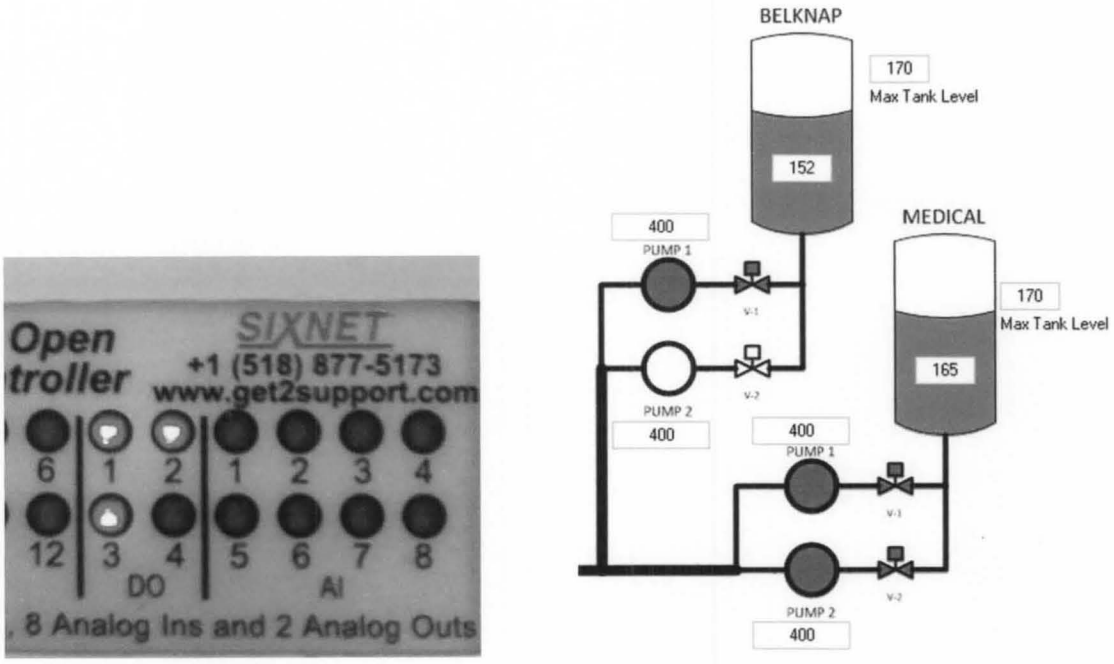


Figure 6.5: RTU Outputs Controlling Simulation Pumps

4. Model Calculations with External Input

As seen in Figure 5.3, when both pumps on tank Belknap are running, the rate at which the tank fills is increased. It is also important to note the flow rates of each pump which can be adjusted by the user must be set in proper proportions. If the flow rate of the reservoir pump is set to that of each of the elevated tank pump flow rates then the treatment tank will drop dramatically. This will also have a direct impact on the chlorine level in the tank.

It is important to note that all calculations in the simulation are made with each tick of the model timer. Pump statuses are updated based on ASCII characters received from the simulation hardware. When a character is received, an interrupt occurs which

reads in the character or characters on the serial port buffer and then analyzes the character to perform the appropriate operation. This execution occurs much faster than the 1 second interval between ticks. For that reason, there is no noticeable delay in the update of the pump status and the calculation of variables in the program.

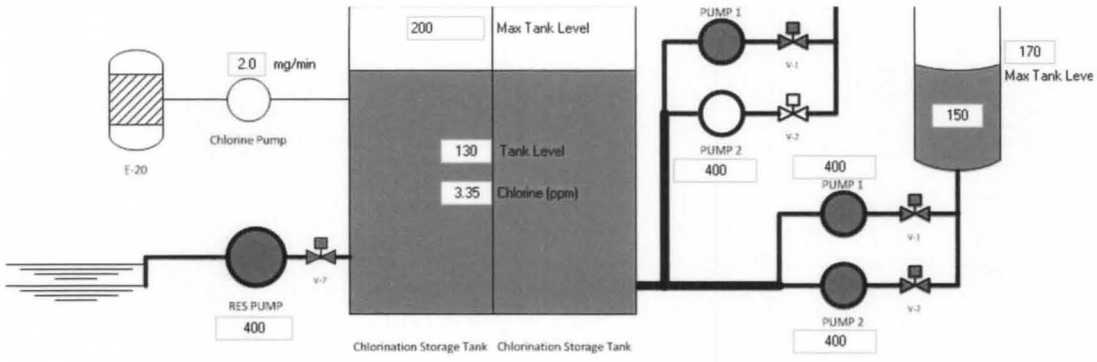


Figure 6.6: RTU Control of All Pumps

5. Hardware Output

On each tick of the model timer, the new value of all variables (analog values) in the program is sent over the communication port to the simulation hardware after the completion of their calculation. The microcontroller parses these values and updates the output accordingly. Since the RTU and HMI have no understanding of what the signal voltages received are in relation to (i.e. height, pressure), the values passed to the microcontroller from the software are a ratio of the current value to the maximum value. This fraction is then used to calculate the output of the associated PORT in order to generate the proportional analog signal to the RTU. The R2R ladder output was

measured to ensure that the proper voltage was being supplied to the RTU as seen in Figure 6.7.

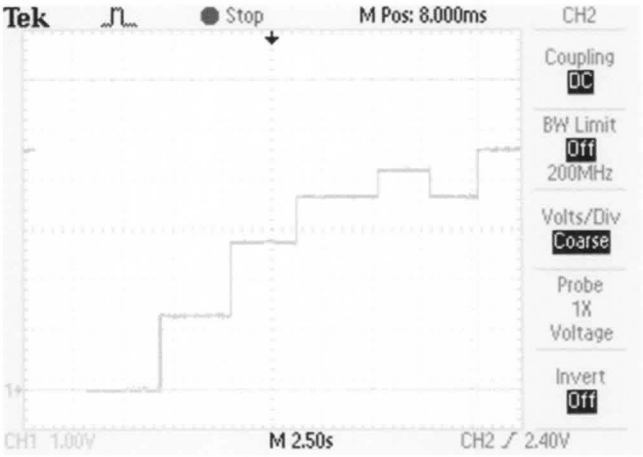


Figure 6.7: Scope Trace of R2R Ladder Output

B. SCADA Security Testing

Testing of the simulation with the hardened remote security preprocessor was completed to demonstrate the full functionality of the simulation. The simulation allows for both a virtual and physical realization of the operation of the SCADA system (i.e. status of pumps, tank levels, chlorine levels) as cyber attacks are carried out on the SCADA system. A compromised SCADA network was used in which a rogue computer gained access to the network and was able to issue MODBUS commands. For testing, the following scenarios were examined:

1. Turn on pump without security device (Write coils attack)
2. Falsify discrete input data without security device (Read request attack)
3. Turn on pump with security device (Write coils attack)

4. Falsify discrete input data with security device (Read request attack)

In scenario 1, a write coil command was issued on a compromised network from an unauthorized computer. In an ideal case, the RTU would only authorize and execute this command when issued from an authorized HMI. Without a security device in place, the command was successfully executed and a simulated pump was activated, filling one of the storage tanks in the simulation as seen in Figure 5.7. The HMI on the SCADA network indicated no pumps running as shown in Figure 5.8.

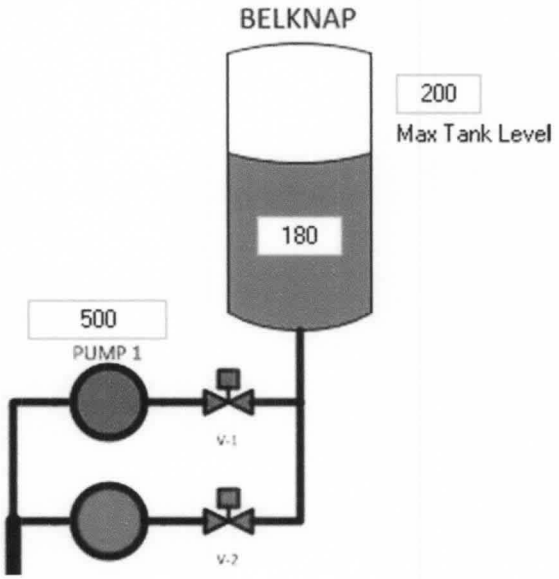


Figure 6.8: Unauthorized Pump Turn On (Write Coils Attack)

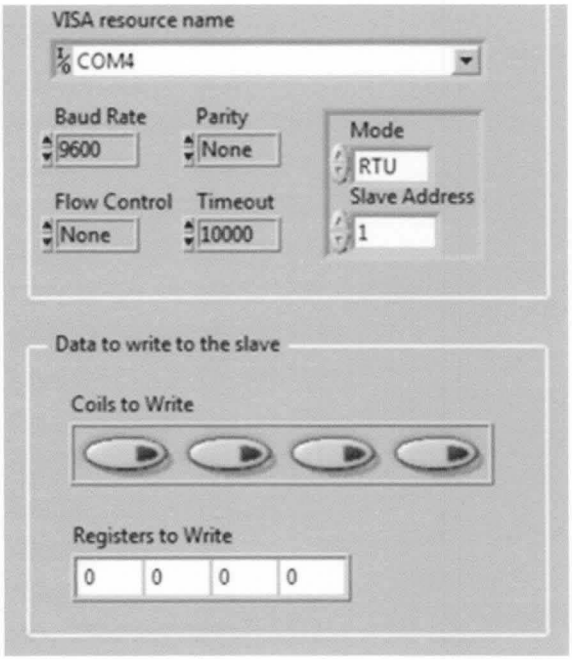


Figure 6.9: HMI Showing No Coils Written During Attack

A read request attack was also tested. In this scenario, a read discrete inputs request was issued from the MTU/HMI. The unauthorized computer intercepted this command and issued a false response indicating a true value for one of the inputs. For testing purposes, the simulation hardware was programmed to turn on a digital output of the simulation when a digital input from the RTU went high. When a pump was turned on using the HMI, a coil would go high on the RTU which would be detected by the simulation. The simulation would then output a high value back into the RTU which could be read by the HMI. Essentially a feedback loop was created to verify the simulation hardware understood to turn on a pump. Figure 5.9 shows the HMI which indicated that no coils have been written, yet shows a high status on a digital input. Figure 5.10 shows the simulation, also indicating no coils (pumps) have been turned on.

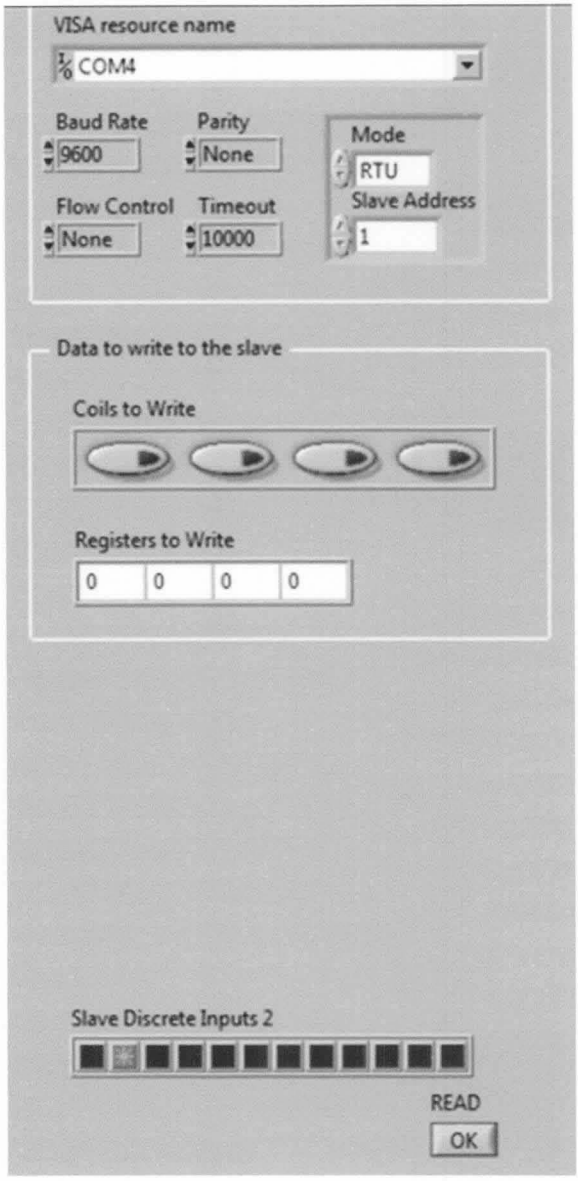


Figure 6.10: HMI Showing No Coils Written but a Pump Running (Discrete Input High)

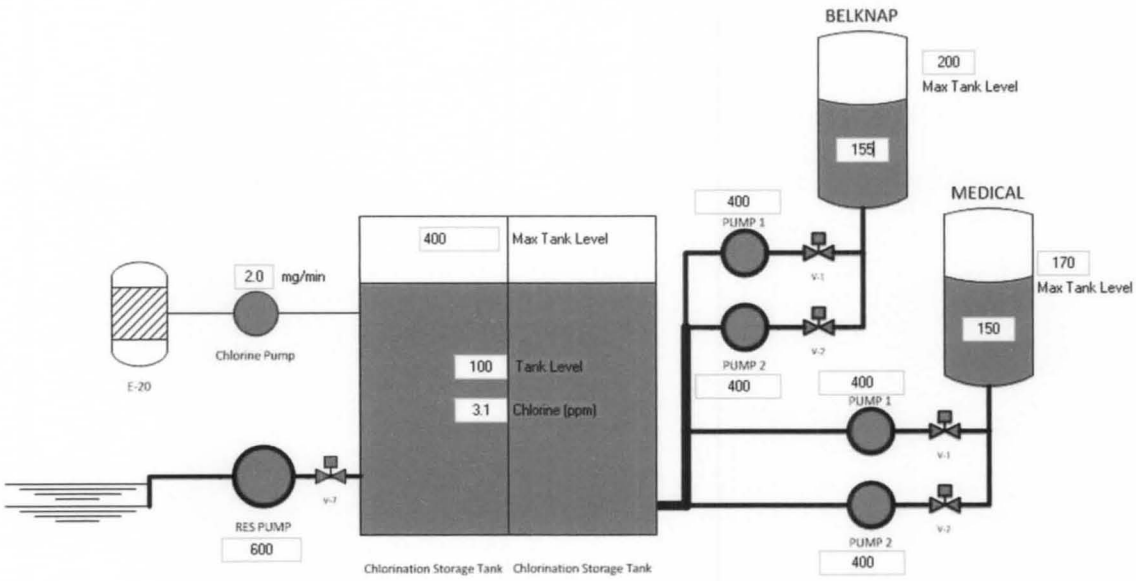


Figure 6.11: Simulation Showing No Pumps Running

Both of these scenarios were repeated with the security device in place (refer to Figure 1.1). In these cases, the security device challenged the write and read commands. Since the commands were not authorized, the MODBUS messages were not passed to the RTU thus no execution occurred in any part of the system including the simulation.

VII. CONCLUSIONS AND FUTURE WORK

Traditional IT security measures are insufficient for securing SCADA systems. The development of new SCADA security devices and approaches requires sector specific simulations for assessing and evaluation new SCADA security technology. The University of Louisville is currently developing a hardened remote terminal security pre-processor. In order to test this device, a water treatment and distribution simulation was designed due to the low cost of implementation and the versatility that a virtualized test bed offers.

Background research was conducted in order to more completely understand the operations of both the treatment process and distribution process that occur in the water system. Assumptions were made in order to focus the development of the simulation to best fit the testing criteria of the security device. Models for both the treatment and distribution subsystems were developed. These models were implemented in a user-friendly GUI based software simulation. Calculations and system statuses were transmitted over a serial connection to hardware designed to work in conjunction with the software component. This hardware interfaced multiple discrete and analog I/O to a RTU used for testing the security device. Testing of the complete system assured that the simulation performed according to specifications and provided adequate testing vectors for the security system.

There are several additions to the simulation that are planned that will increase the functionality of the system. First, the RTUs analog outputs can be used to control variables within the simulation software. The circuit design for interfacing the RTU and microcontroller in this manner has already been completed and outlined in previous chapters. The chlorination rate of the simulation is currently controlled by the user of the software. A slight modification to the code would allow for the HMI and RTU to control this variable for example.

The VersaTRAK RTU used in testing was not running any logic; its inputs and outputs were read and controlled by the HMI. Additional testing should be performed with a RTU running some type of operational logic. Work has been started on programming an Allen Bradley PLC using the associated RSLogix software. A simple ladder logic program can be developed that will allow the RTU to automatically turn on and off pumps as the tank levels change much like a typical system would do as outlined in chapter two. This would provide a fully functional testing system for cyber security devices that are being developed.

The water treatment and distribution simulation outlined will be used as a testbed for new devices aimed at securing SCADA systems. It offers a large amount of I/O that can be connected to a wide variety of RTUs that are used in the field today. Using a simple serial interface between the software and hardware, the system can be easily transported and set up in laboratories or industrial settings for additional testing. This testing system will offer a unique method of testing new cyber security devices and offer insight into any issues or problems that should be resolved before the device is deployed in a real world setting.

REFERENCES

[1] T M Chen, "Stuxnet, the real start of cyber warfare," *Network, IEEE*, vol. 24, no. 6, pp. 2-3, March 2010. [Online]. http://en.wikipedia.org/wiki/Stuxnet

[2] J Graham and J Hieb, "Designing Security-Hardened Microkernels for Field Devices," *IFIP International Federation for Information Processing*, vol. 290, no. Critical Infrastructure Protection II, pp. 129-140, 2008.

[3] US EPA. (2012, March) United States Environmental Protection Agency. [Online]. http://water.epa.gov/learn/kids/drinkingwater/watertreatmentplant_index.cfm

[4] US EPA. (2012, March) United States Environmental Protection Agency. [Online]. http://water.epa.gov/lawsregs/rulesregs/sdwa/tcr/distributionsystems.cfm

[5] MODBUS-IDA, "MODBUS Messaging on TCP/IP Implementation Guide," 2006.

[6] AutoMatrixInc. (2012) AutoMatrixInc. [Online]. http://automatrixinc.com/projects.html

[7] One Hundred Nineth Congress House of Representatives, "SCADA Systems and the Terrorist Threat: Protecting the Nation's Critical Control System," in *Joint Hearing before the Suncommittee on Economic Security, Infrastructure Protection, and Cyber Security with the Subcommittee on Emergency Prepardness, Science, and Technology, of the Committee on Homeland Security*, Washington D.C., 2005.

[8] Joseph Weiss, "Electronic Threats to Industrial Control Systems," in *Protecting Industrial Control Systems from Electronic Threats*. New York, NY, United States: Momentum Press, 2010, ch. 6, pp. 43-51.

[9] Marshall Abrams and Joe Weiss, "Malicious Control System Cyber Security Attack Case Study-Maroochy Water Services, Australia," Case Study 2008.

[10] V M Igure, S A Laughter, and R D Williams, "Security Issues in SCADA Networks," *Computers and Security*, vol. 25, no. 7, pp. 498-506, 2006.

[11] A. Miller, "Trends in Process Control Systems Security," *IEEE Security and Privacy Magazine*, vol. 3, no. 5, pp. 57-60, September 2005.

[12] D. Geer, "Security of Critical Control Systems Sparks Concern," *Computer*, vol. 3, no. 5, pp. 20-23, January 2006.

[13] T. Brown, "Security in SCADA Systems: How to Handle the Growing Menace to Process Automation," *Computing and Control Engineering Journal*, vol. 16, no. 3, pp. 42-47, 2005.

[14] Lewis A. Rossman, "EPANET 2 Users Manual," Cincinnati, OH, 2000.

[15] KYPipe, "KYPipe 2012 Users Manual," Lexington, KY, 2010.

[16] C. M. Davis et al., "SCADA Cyber Security Testbed Development," in *North American Power Symposium*, 2006, pp. 483-488.

[17] Carlos Queiroz, Abdun Mahmood, Jiankun Hu, Zahir Tari, and Xinghuo Yu, "Building a SCADA Security Testbed," in *Third International Conference on Network and System SEcurity*, 2009, pp. 357-364.

[18] Justin Adams, Jeffrey Hieb, James Graham, and Rammohan Ragade, "A Water System Simulation for Testing Security Enhancements to SCADA Field Sevices," in *International Conference on Computer Applications in Industry and Engineering*, 2011, pp. 305-310.

[19] US EPA. (2012, March) US Environmental Protection Agency. [Online]. http://water.epa.gov/drink/contaminants/basicinformation/disinfectants.cfm

[20] Justin Adams, "A WATER DISTRIBUTION AND TREATMENT SIMULATION FOR TESTING CYBER SECURITY ENHANCEMENTS FOR WATER SECTOR SCADA SYSTEMS," Computer Engineering and Computer Science, Thesis 2011.

[21] Paul Horowitz and Winfield Hill,.: Cambridge University Press, 1989.

[22] Rob Stein and John Day, "DA Conversion Using PWM and R-2R Ladders to Generate Sine and DTMF Waveforms," 2002.

APPENDIX I – Glossary

ADC – Analog to Digital Converter

ADU – Application Data Unit

ASCII – American Standard Code for Information Interchange

COM – Communication

DAC – Digital to Analog Converter

DAQ – Data AcQuisition

EPA –Environmental Protection Agency

GUI – Graphical User Interface

HMI – Human Machine Interface

I/O – Inputs/Outputs

IP – Internet Protocol

LabVIEW – Laboratory Virtual Instrumentation Engineering Workbench

LAN – Local Area Network

MTU – Master Terminal Unit

MVS – Microsoft Visual Studio

PC – Personal Computer

PDU – Protocol Data Unit

PWM – Pulse Width Modulation

PLC – Programmable Logic Controller

POTS – Plain Old Telephone Service

PPB – Parts Per Billion

PPM – Parts Per Million

PSI – Pounds per Square Inch

R2R – Resistor to Resistor

RES – Reservoir

RF – Radio Frequency

RFC – Request for Comments

RTU – Remote Terminal Unit

SCADA – Supervisory Control and Data Acquisition

TCP/IP – Transmission control Protocol/Internet Protocol

TTL – Transistor-Transistor Logic

USB – Universal Serial Bus

VI – Virtual Instrument

VISA – Virtual Instrument Software Architecture

APPENDIX II – Mathematical Equations for Simulation Variables

$$V_{water} = V_{current} + (V_{flow\ in} - V_{flow\ out})t$$

V_{water} is the volume of water in the tank in cubic feet (1 ft^3 = 7.4805 gallons)

$V_{current}$ is the current volume of water in tank

$V_{flow\ in}$ is the flow of water into the tank

$V_{flow\ out}$ is the flow of water out of the tank

$$h_{water} = \frac{V_{water}}{A'_{tank}}$$

h_{water} is the height of the water in the tank

A'_{tank} is the cross sectional area of the tank (l x w)

$$Cl_{concentration} = \frac{M_{Cl\ old}}{V_{water\ old}} + \frac{M_{Cl\ new}}{V_{old+new}}$$

$M_{Cl\ old}$ is the current amount of Cl (mg) in the water

$V_{water\ old}$ is the amount of water in the tank minus any that was pumped into the distribution system

$M_{Cl\ new}$ is the amount of Cl (mg) added during the time frame (chlorination rate)

$V_{old+new}$ is the volume of water in the tank minus any outflow into the distribution system and adding any inflow (flow rate of pump)

$$V_{cylinder} = \pi r^2 h,$$

r is the radius of the tank

h is the varying height based on the capacity variable assigned

$$h_{water} = 2.31\ ft/psi$$

$$P = \left(\frac{V_{water}}{A'_{tank}}\right) C,$$

P is the pressure read by the sensor in psi

V_{water} is the volume of water in the tank in cubic feet (1 ft^3 = 7.4805 gallons)

A'_{tank} is the cross sectional area of the tank in square feet

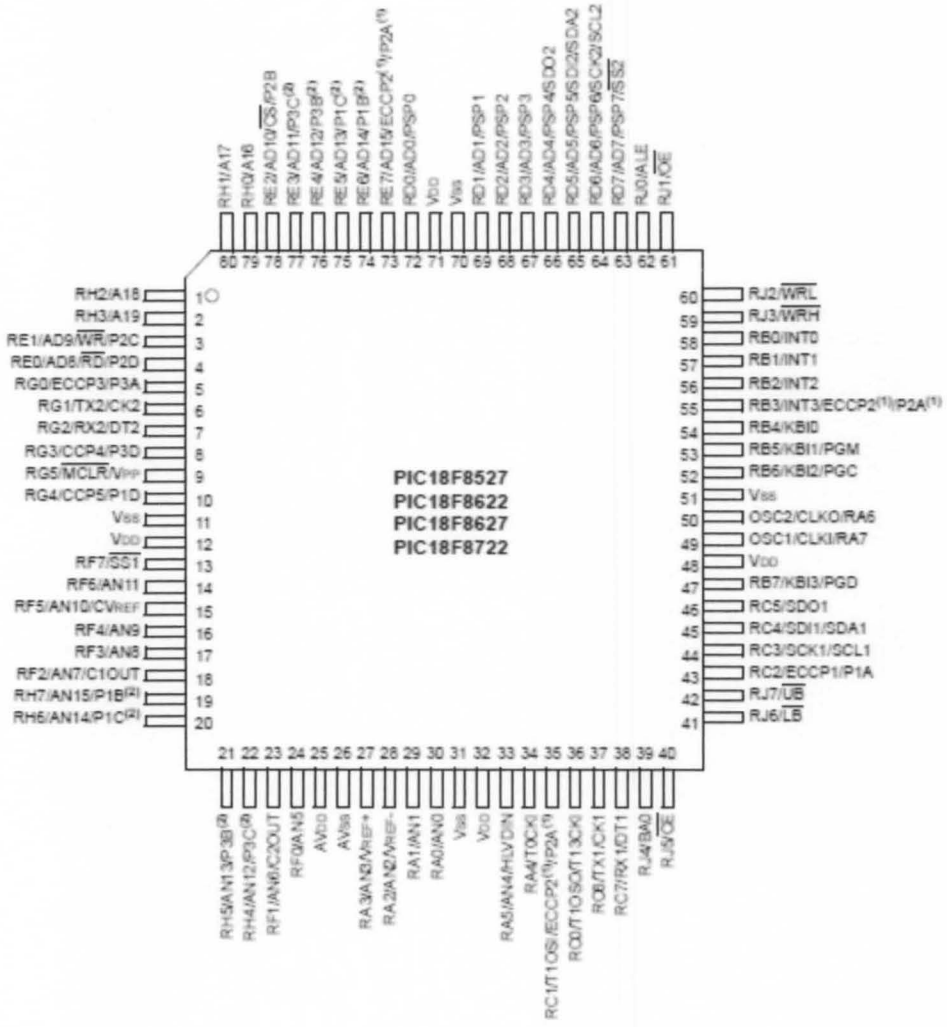
C is a constant conversion factor between pressure and height of water (1 psi / 2.31 ft).

$$D = N * U_{TOD},$$

D is the demand for each branch

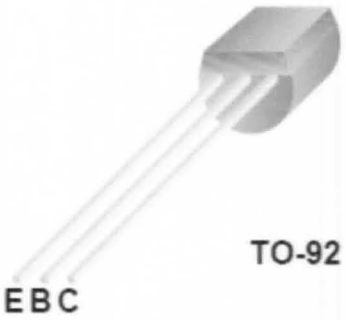
N is the number of customers

$U_{TOD}$is the coefficient of usage based on the time of day



PIC18F8722 Microcontroller Pin Definitions

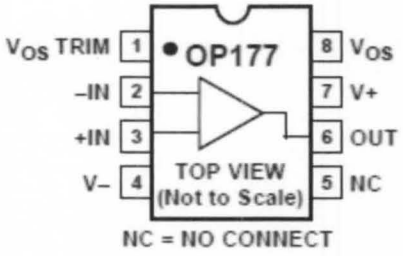
2N3904



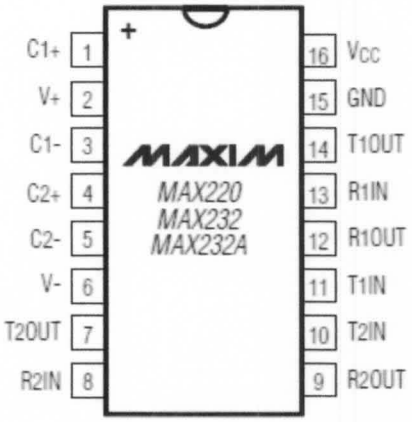
Absolute Maximum Ratings* T_a = 25°C unless otherwise noted

Symbol	Parameter	Value	Units
V_{CEO}	Collector-Emitter Voltage	40	V
V_{CBO}	Collector-Base Voltage	60	V
V_{EBO}	Emitter-Base Voltage	6.0	V
I_C	Collector Current - Continuous	200	mA
T_J, T_{stg}	Operating and Storage Junction Temperature Range	-55 to +150	°C

2N3904 NPN Transistor

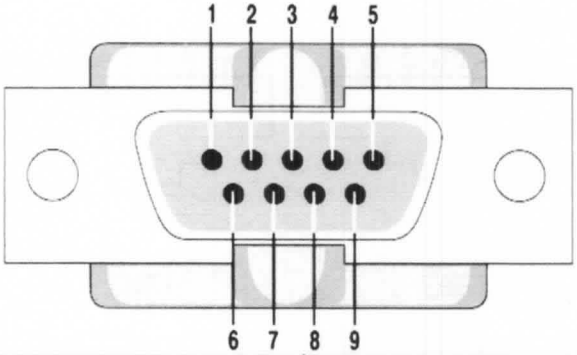


OP177 Ultra Precision Op-Amp Pin Definitions



CAPACITANCE (μF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	0.047	0.33	0.33	0.33	0.33
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

MAX232 Driver/Receiver



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

DB9 Connecter Pinout

CURRICULUM VITA

Kyle Moss

140 W. Washington St #512
Louisville KY, 40202

Home Phone: (270)799-2723
Work Phone: (270)781-3859
E-mail: kyle.moss@wku.edu
ktmoss01@louisville.edu

Education

M.S. Electrical Engineering, 2012
University of Louisville, Louisville, KY

M.S. Homeland Security Sciences, 2012
Western Kentucky University, Bowling Green, KY

B.S. Electrical Engineering, 2008
Western Kentucky University, Bowling Green, KY

Experience

Graduate Research Assistant, 2011-2012
J.B. Speed School of Engineering
University of Louisville

Electrical Engineer, 2009-2012
Applied Physics Institute
Western Kentucky University

Research Associate, 2008-2009
Applied Physics Institute
Western Kentucky University

Research Assistant, 2005-2008
Applied Physics Institute
Western Kentucky University

Honors

John Russell Award in Electrical Engineering, 2008
Department of Engineering
Ogden College of Science and Engineering
Western Kentucky University

Ogden Research Scholar, 2004-2008
Ogden College of Science and Engineering
Western Kentucky University

Publications

Craig Dickson, Stuart Foster, **Kyle Moss**, Anoop Paidipally, Jonathan Quiton, William Ray, and Phillip Womble, *Stochastic Modeling for Automatic Response Technology with Applications to Climate and Energy,* at the 8th Kentucky Entrepreneurship Conference, June 2012

Jeffrey L. Hieb, James H. Graham, Nathan Armentrout, and **Kyle Moss**, *Security Pre-Processor for Industrial Control Systems,* at the 8th Kentucky Entrepreneurship Conference, June 2012

Sowell, Dewayne Emmanual, Phillip Womble, Alexander Barzilov, and **Kyle Moss,** *Cost Effective Robotic Solutions for Military and Law Enforcement* at Western Kentucky University Student Research Conference, Bowling Green, KY, February 2010

Hamlet, Sean and Lodmell, Matthew, Stacy Wilson and **Kyle Moss**, *Redesign of Electrical System for Remotely Controlled ATV Platform* at Western Kentucky University Student Research Conference, Bowling Green, KY, February 2010

Berry, James Alex; Morrison, Travis; and Simpson, Mike "*Design of a Remotely-Controlled Mobile Platform for Field Neutron Interrogation*" (Dr. Stacy Wilson and Kyle Moss) at Western Kentucky University Student Research Conference, Bowling Green, KY, February 2010

Simpson, Michael, **Kyle Moss**, Joshua J. Pierce, Jon Paschal, and Phillip C. Womble, *Utilizing Wireless Technology for Enhancing the Electrocardiogram* at Western Kentucky University Student Research Conference, Bowling Green, KY, February 2009

Lodmell, Matthew and **Kyle Moss,** *ZigBee Radios for Homeland Security Applications* at Western Kentucky University Student Research Conference, Bowling Green, KY, February 2009

Davenport, Christopher**, Kyle Moss,** Ron Hopper, and Phillip Womble, *Building Access Security Using RFID Technology* at Western Kentucky University Student Research Conference, Bowling Green, KY, February 2009

Kyle Moss, Phillip C. Womble, Jon Paschal, Ryan Moore Jr., Randall Haupt, *Design of Electroencephalogram and Electrocardiogram Technology Including Wireless*

Integration for Use in Polysomnography at Western Kentucky University Student Research Conference, Bowling Green, KY, April 2008

Joshua J. Pierce, **Kyle Moss**, Jon Paschal, Phillip Womble, *Utilizing Wireless Technology for Enhancing the Electrocardiogram* at Argonne Undergraduate Symposium, Argonne, IL, November 2007

Christopher H. Davenport, Phillip C. Womble, Alexander Barzilov, Jon Paschal, James Lodmell, Robert Hernandez, **Kyle T. Moss**, *IED Neutralization Using Low Cost Unmanned Ground Vehicles* at Argonne Undergraduate Symposium, Argonne, IL, November 2007

Kyle Moss, Phillip Womble, Alexander Barzilov, Jon Paschal, Jeremy Board, *Wireless Orthogonal Sensor Networks for Homeland Security* at 2007 IEEE Conference on Technologies for Homeland Security, Woburn, MA, May 2007

Barzilov, P. Womble, I. Novikov, J. Paschal, Jeremy Board, and **Kyle Moss**, *Network of Wireless Gamma Ray Sensors for Radiological Detection and Identification* at the SPIE Defense and Security Symposium, Orlando, FL, April 2007

Alexander Barzilov, Jeremy Board, **Kyle Moss,** Ivan Novikov, Phillip Womble, *Algorithm for Identification and Location of Gamma-Ray Sources using Distributed Sensor Networks* at 2007 IEEE Conference on Technologies for Homeland Security, Woburn, MA, May 2007

Kyle Moss, Phillip Womble, Alexander Barzilov, and Jon Paschal, *Wireless Gamma Ray Spectrometer with Automatic Isotope Identification* at Argonne Undergraduate Symposium, Argonne, IL, November 2006

Kyle Moss, Phillip Womble, Alexander Barzilov, and Jon Paschal, *A Gamma Ray Spectrometer Based on Mobile Phone Technology* at Conference Experience for Undergraduates, Division of Nuclear Physics Conference, Nashville, TN, October 2006

Kyle Moss, Lindsay Hopper, Tim Morgan, Phillip Womble and Alexander Barzilov, *Rail Tank Car Leakage Detection System (RLEAKS)* at Sigma Xi Research Conference, Bowling Green, KY, April 2006

Tim Morgan, Jeremy Board, **Kyle Moss,** Eric Houchens, Ian Rice, Doug Harper, Phillip Womble, Alexander Barzilov, and Jon Paschal, *Rigaku D-Max B XRD: A Materials Characterization Tool for IBMAL* at Sigma Xi Research Conference, Bowling Green KY, April 2006

Alexander Barzilov, Phillip C. Womble, Jon Paschal, Lindsay Hopper, Ryan Moore, Timothy Morgan, Britton Wallace, Jeremy Board, **Kyle Moss**, and Joseph Howard,

Inexpensive Robot Platforms for Detection and Neutralization Applications at 7th International Symposium on Technology and the Mine Problem, Monterey, CA, 2006