

LOGIXPRO Based SCADA Simulations Model for Packaging System in Dry ICE Plant

Prashu Jain¹, K. Nithiyananthan^{2*}, Raghuraman³, Gowrishankar Kasilingam⁴

¹ Refrigeration and Oxygen Co. Ltd, Safat, Kuwait.

^{2, 3 & 4} Faculty of Engineering & Computer Technology, AIMST University, Malaysia

Article Info

Article history:

Received Dec 26, 2014

Revised Feb 25, 2015

Accepted Mar 8, 2015

Keyword:

Dry Ice plant
Ladder Diagram
LOGIXPRO
Relay Logic
SCADA

ABSTRACT

Supervisory Control and Data Acquisition (SCADA) systems control and monitor industrial and critical infrastructure functions, such as electricity, gas, water, waste, railway, and traffic. The main objective of this work is to develop SCADA simulation model for packaging system in dry ice plant. Dry ice is an important refrigerant for keeping foods cold and preventing bacterial growth during shipment. Dry ice used for cooling or freezing foods must be very clean and considered food grade to ensure that food it may touch will not be contaminated. Some recent developments for its use include using the pellets in blasting or cleaning and its increasing use in transporting medical specimens, including hearts, limbs, and tissues, for reattachment and transplantation. The manufacturing process of dry ice has not changed significantly in many decades and is a relatively simple process of pressurizing and cooling gaseous carbon dioxide. But because of its growing demand, packaging becomes vital. An attempt has been made to develop and automate LOGIXPRO based SCADA simulations for dry ice plant to improve packaging and extensively reduce operating labor costs.

Copyright © 2015 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

K. Nithiyananthan,
Faculty of Engineering & Computer Technology,
AIMST University,
Jalan Semeling – Bedong, 08100 Bedong, Kedah Darul Aman, Malaysia.
Email: nithiieee@yahoo.co.in nithi@aimst.edu.my

1. INTRODUCTION

Industrial processes involve certain chemical or mechanical steps to help in the manufacture of items for the industry. They form the basic components of a large scale industry. Therefore to help in producing these good more effectively and efficiently SCADA systems are used. The major attraction of SCADA is its ability to extensively reduce operating labor costs as the whole process is automated. The computer systems used to monitor and control major infrastructure are known by various names, among the most common (SCADA) systems. The system's name reflects its basic functions: it must provide data related to the operating state of the system and allow operators to remotely control the distributed system. By utilizing these services, system operators can effectively respond to changes in the process operating conditions or adapt to evolving production goals or changing corporate directives. The rising prevalence of SCADA systems in infrastructure is a result of the variety of benefits such systems can provide to the businesses that operate them. By shifting away from purpose-built hardware towards more flexible full-featured hardware running operation-specific software, development of control systems can be accomplished more quickly and at lower cost. Specialized interfaces can be designed for system operators that minimize the difficulty of management and allow rapid and effective reactions to changing process conditions. The use of SCADA systems allows high-level management of the industrial process by merging data from the many distributed portions of the process. This can help enhance the

robustness and reliability of the system. Finally, flaws in the design of the control system can be more easily addressed and operators may receive maintenance and support from vendors of SCADA software and hardware. Taken as a whole, these benefits provide a powerful incentive to migrate to SCADA solutions for control of complex distributed processes [4]. The differences in SCADA security and traditional IT approaches mean that even reliable and trusted solutions cannot be applied without significant testing. The potential damage of malfunctions and loss of availability of critical infrastructure further necessitate thorough testing. However, testing new solutions for SCADA systems is not easily accomplished. Live systems clearly cannot be used because of the potential damage unintended consequences could cause. Developing parallel but inactive systems for the purpose of testing is an approach that is often viable for testing network security, but would be prohibitively expensive for testing complex infrastructure installations. Instead, the complexity of SCADA systems calls for a thorough software simulation to help uncover the benefits and consequences of novel security solutions. Simulating SCADA solutions is a complex and difficult task, however. Because the development of a single-purpose simulation that captures the behavior of only a single system would be inefficient and costly, simulations should be composed of simple and reusable simulation components. Simulators dealing with the industrial process, the controller software, and the intervening network could be combined to form a simulation of the SCADA system as a whole. This requires coordinating a variety of simulation engines, each with a different set of internal data and with varying approaches to the progression of time and events.

2. SCADA SYSTEM OVERVIEW

A wide variety of industrial processes are managed via computerized control systems, and their diverse purposes mean that industrial control systems themselves are diverse in implementation. The term SCADA is most frequently used to describe systems whose assets are highly distributed geographically. The control of electrical grids and oil and gas pipelines, for instance, involves aggregating sensor measurements from hundreds of widely dispersed field devices so that operators can use a centralized control interface to manage the whole process in real time. Field devices are located physically close to the portion of the process that must be controlled, and monitor sensors and drive actuators connected to the process. They are connected to the SCADA control center via a wide area network which may use a variety of topology and protocols and be wired or wireless. Such systems must typically take into account the low bandwidth and relative lack of reliability of the networks in use, perhaps employing fault-tolerant hardware and algorithms. In addition, they must typically contend with legacy hardware and protocols since widely dispersed hardware devices are difficult and expensive to upgrade [1]. Much smaller scale operations, such as chemical manufacturing plants and pharmaceutical processing facilities, are also examples of SCADA systems. These geographically localized processes may reside entirely within a single plant floor and are sometimes differentiated from geographically dispersed SCADA systems with the term Distributed Control Systems (DCSs) [1]. These systems use field devices that are located physically close to the portion of the process under control and are connected to the master control center via the control network. The control of the whole process is modularized with the use of local controllers to provide fault tolerance and reduce the impact of a malfunction at a single field device. DCSs typically use a highly reliable and relatively high bandwidth LAN to connect field devices with the control center. In addition, physical security may be more effective since a geographically centralized system is less difficult and expensive to protect. Although the systems that employ SCADA are widely varied in topology, scale, and purpose, they are unified by a single type of architecture. The recognition of their fundamental similarities is important to the research of SCADA security, since it allows researchers to make use of general models of the class of the class of all SCADA systems. This general model is composed of four major parts: the process to be controlled, the field devices physically connected to it, the centralized control center, and the network that connects the controller and field devices. The relationship between these components is shown in Figure 1.

1.1. SCADA Architecture

The process is the physical phenomenon that operators seek to control. This portion of the system will be distinct in all SCADA systems. The process typically can be broken down into a number of smaller control problems. For instance, a plant producing a particular chemical in a reactor may need to control the temperature and pressure of the reaction as well as the volumes of the reactants. Each of these may be considered separate control problems, with local controllers engaged in the maintenance of each variable within established operating limits. However, these variables are interrelated; the pressure and volume of reactants of the reactor affect its temperature and vice versa. Local controllers performing localized tasks cannot effectively maintain the high level operation of the system, necessitating a centralized master control system to perform this task. Field devices interact with the process via sensors and actuators.

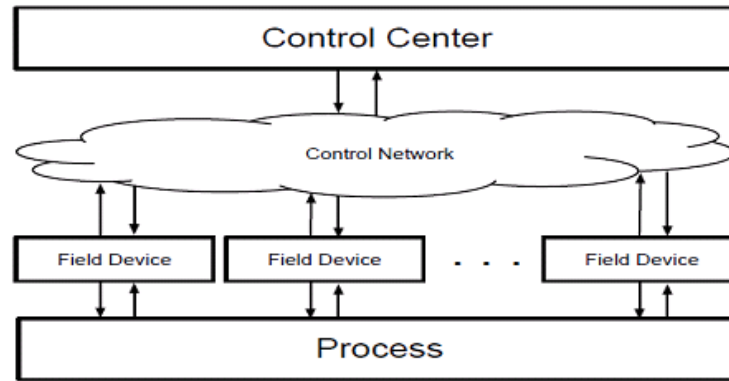


Figure 1. SCADA Architecture

They are sometimes termed Programmable Logic Controllers (PLCs), reflecting the fact that they act as controllers on a local level. Field devices deal with a localized control problem, but also send updates and receive commands from the master controller so that their local control loop can be operated in accordance with the overall process control strategy. For instance, a field device controlling the liquid level in a tank may receive liquid level readings from a sensor and be able to maintain the appropriate level by using an actuator that controls a runoff valve. Its local control problem would be to maintain the liquid level in the tank within some tolerance of a set value. Because this set point value is likely affected by other factors in the process, however, the field device would receive commands to set this value from the centralized control center. Because the state of the local control problem likely affects other the state of the process as a whole, the field device would send regular sensor updates or alarms to the control center. This forms a high level control loop that drives the lower level localized control loops. The local control loop's operation and relationship to the rest of the SCADA system is diagrammed in figure 2. Field devices may connect to a single sensor or actuator or may be connected to a large network of sensors and actuators and maintain a complex local control loop [2].

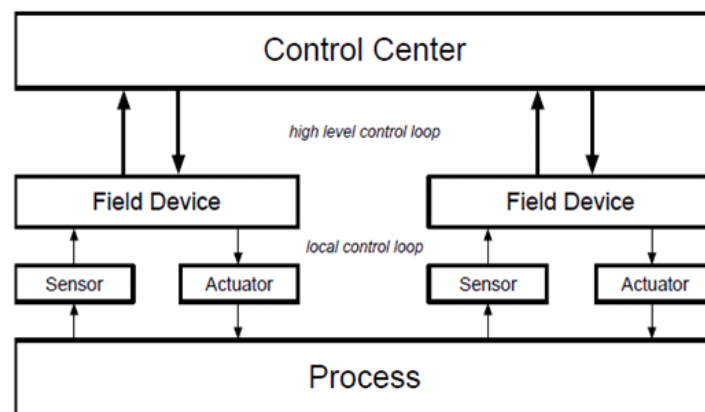


Figure 1. Field Device Control Loop

The control center acts as the master controller, maintaining the high level operation of the process. Many field devices are employed by SCADA systems to operate local control loops, each affecting a single control problem, but in atypical process these control problems are interrelated. For instance, the control system operating a canal may use a large number of field devices controlling the water levels in a system of locks. Because the control strategy of one lock directly affects the control strategy of its neighbors, a high level strategy must be employed to ensure correct operation. The control center sends control commands and receives sensor updates from the field devices to allow this high level control. Depending on the SCADA deployment, control centers may operate automatically or rely on the intervention of human operators. The control center provides the interface to the human operators of the system. This interface is called the Human Machine Interface (HMI) and allows the operators to see an aggregated view of the state of the process and provides the means to send control commands to field devices in order to maintain correct operation. A

control center may include several HMIs, each reflecting the requirements of its users. For instance, administrators and business managers require a different set of data and controls than system engineers. The control center is connected to field devices via the control network, and may also be connected to a corporate network or WAN to allow remote access to engineers and business administrators. The connection between the control center and field devices is provided by the control network. This may be a wired or wireless network and may operate with a variety of network protocols. Some control networks use TCP/IP while others use fieldbus protocols, which are simple protocols designed around the sensor update and control command communication patterns of SCADA networks [3]. Depending on the process, it may be important to provide real time guarantees or provide fault tolerant or redundant networks. Communication on control networks typically consists of control commands from the control center and sensor updates from the field devices. The communication can be asymmetric, with sensor messages being larger and more frequent than control messages. Some field devices communicate at fixed intervals while others use alarms to communicate only significant events. Prioritization of important control messages over bulk sensor readings is a typical communications requirement, as is some guarantee on the timeliness and stability of message delivery [3].

1.2. Programmable Logic Controller (PLC)

A programmable logic controller is a controller based on micro-processor which uses certain logic to accumulate instructions and carry out certain functions to control various machines and processes. Input devices like sensors and output devices like motors and valves are connected to the PLC. The programmer enters the various sets of instructions which are to be carried out by the PLC. These instructions are stored in the memory of the PLC. PLCs are designed to control and monitor tasks in an industry. Thus PLCs are generally designed to withstand harsh conditions like high temperature, vibrations and noise. Contain slots inside the controller which are required for various input and output devices and its effortlessly programmable. A wide range of PLCs are available in the market. Generally PLCs are in the box form containing six, eight, twelve or twenty inputs and four, eight or sixteen outputs. They are designed for on field use near the machine it is designed for. Systems that need a larger number of inputs and outputs are likely to be in modular form. Therefore the quantity of modules can easily be increased according to the need of the user. Figure 3 shows the Allen-Bradley PLC installed in a control panel.



Figure 3. Allen- Bradley Programmable Logic controller

2. SCADA BASED DRY ICE PLANT MODEL

The only raw material used in the manufacture of dry ice is carbon dioxide. This raw material is the byproduct of the refinement of gases emitted during the manufacture or refinement of other products. Most carbon dioxide used in the manufacture of dry ice in the United States is derived from refinement of gases given off during the refinement of petroleum and ammonia. The carbon dioxide emitted during these processes is sucked off and "scrubbed" to remove impurities for food grade carbon dioxide that will eventually become dry ice [7]. It is very much evident that dry ice making process is required to be automated and interfaced with SCADA control system. The figure 4 shows the generic SCADA data communications network suitable for any process control industry. Automation logic for proposed model has been carried out using LogixPro simulator [6].

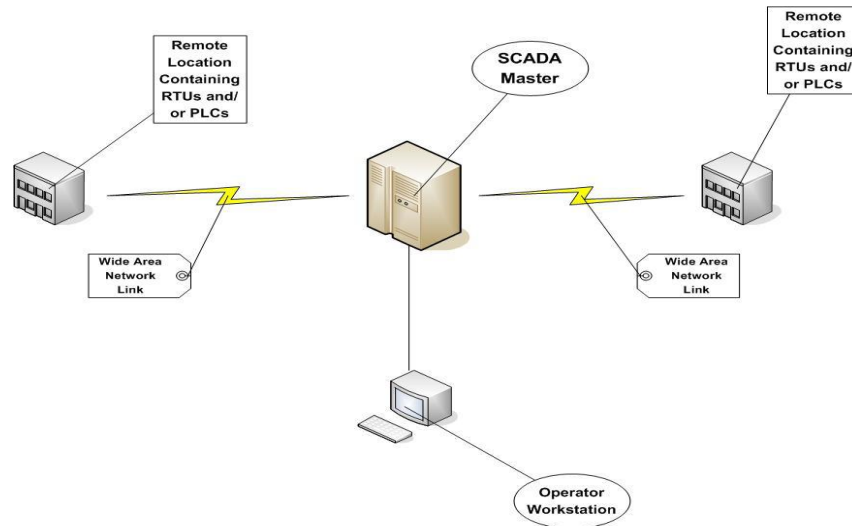


Figure 4. SCADA Data Communications Network

LogixPro simulator allows one to practice and develop programming skills. It contains the PLCs and all other various electrical components that are required to learn programming. Various programmes using LADDER LOGIC can be developed and the required simulations can be viewed. The Edit Panel provides an easy access to all the instructions. These instructions can easily be dragged and dropped into the program. Using ladder logic various programs can be created. These programs can be edited using the same edit panel. Once the program is ready to be viewed, we click on the Toggle Button of the Edit panel to bring the PLC Panel into view. On clicking the download button on the PLC Panel one can download the program into the PLC. After the program is downloaded, it can then be placed into RUN mode. This initiates the scanning of the program. Various Programmable Logic Controllers, which are connected to a variety of field sensing devices. A communications system is used for the transfer of data between the computers present in the central system and field data devices. These systems include telephone, satellite, cables, radio or a combination of any of these [6].

Steps to perform SCADA simulations for dry ice plant [7]:

- SCADA control engineers map out system that integrates HMI, various alarms, controller communications and multiple sensors for every new project. This gives the basic idea of how the SCADA simulation software look would look like.
- Once the mapping occurs and the basic details regarding the software are obtained, programming of the PLC's takes place.
- Testing a SCADA system usually takes place on the field of operation after the setup of the system is completed.
- In SCADA to change or modify the logic to meet new application necessities is comparatively easy.
- The training of SCADA operator is the responsibility of the person that develops the application.

3. DEVELOPMENT OF LADDER LOGIC MODEL FOR DRY ICE PLANT

The simple ladder logic has been implemented in LOGIXPro for automation of dry ice packaging plant. Ladder logic is widely used to program PLCs, where sequential control of a process or manufacturing operation is required. Ladder logic is useful for simple but critical control systems, or for reworking old hardwired relay circuits. As programmable logic controllers became more sophisticated it has also been used in very complex automation systems [8]. The ladder logic for rung 0 is shown in Figure 5. This rung gives the condition for machine start state. Here the binary start variable is energized when the start button is pushed. This start button is latched on with the binary start variable so that the start binary variable remains energized.



Figure 5. Ladder logic for rung 0

It is connected in series to the stop switch so that when the stop button is pushed the start binary variable gets de-energized. Hence when the start button is pushed the start variable is energized and when the stop switch is pushed the start variable is de-energized. The ladder logic for rung 1 is shown in Figure 6.

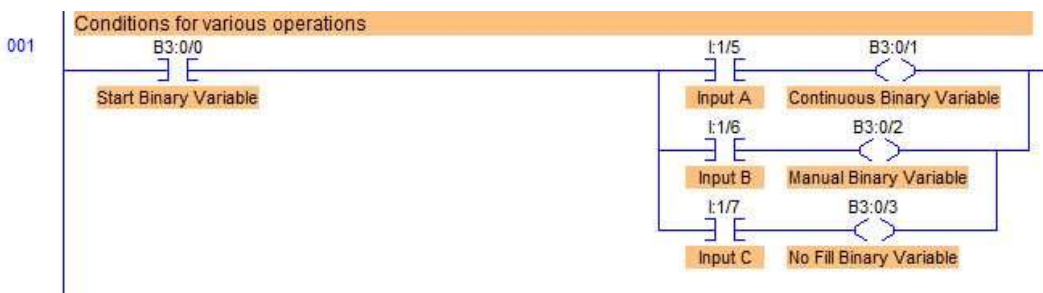


Figure 6. Ladder logic for rung 1

Rung 1 basically tells the remaining program which input mode is selected. The start variable has been connected in series to the 3 input switches, input A, input B and input C which in turn are connected to three binary variables namely continuous binary variable, manual binary variable and no fill binary variable. When an input is selected it energizes the connecting output binary variable. It means that if input A is selected then the continuous binary variable (B3:0/1) will get energized and output regarding condition A is simulated. Similarly that if input B is selected then the manual binary variable gets energized and the output regarding condition B is simulated. The ladder logic for rung 2 is shown in Figure 7.

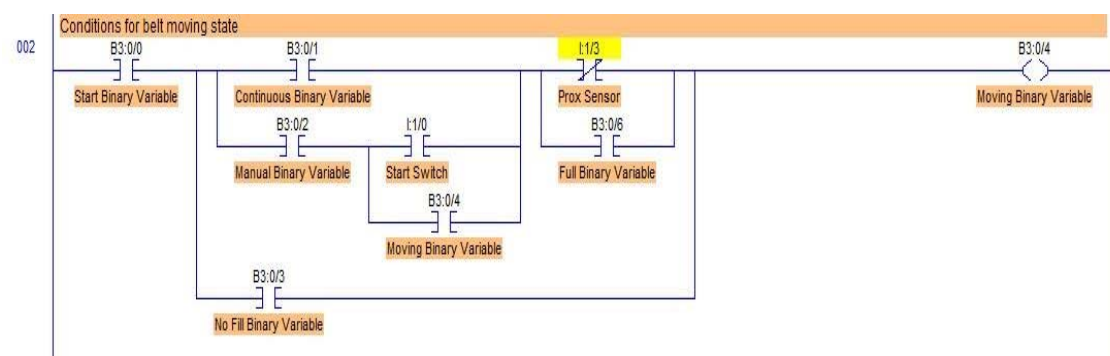


Figure 7. Ladder logic for rung 2

This run gives the conditions to move the conveyor belt. Again the start binary variable is connected in series with the three variables. For the first case when contact is energized and the prox sensor is not energized the belt keeps moving. As soon as the prox sensor gets energized, the contact is broken and the belt stops moving. For the second case when contact is energized and the start pushbutton is pressed then the conveyor belt moves. To keep the conveyor belt moving the start switch contact is latched with the moving binary variable. As soon as the prox sensor gets energized, the contact is broken and the belt stops moving. The full binary variable is connected in parallel to the prox sensor contact so that when

the box is full the belt starts moving again. In the third case when the no fill binary variable is energized the belt keeps moving. The Start binary variable is connected in series so that at any time when the stop pushbutton is pressed the belt stops moving. The ladder logic for rung 3 is shown in Figure 8. This rung gives us the condition regarding the motor.



Figure 8. Ladder logic for rung 3

Here the moving binary variable is connected to the output variable motor on. It means that whenever the moving binary variable is energized the motor is switched on otherwise it remains off. The ladder logic for rung 4 is shown in Figure 9. This rung tells us when the run light should be switched on.



Figure 9. Ladder logic for rung 4

Here the start binary variable is connected to the output variable of the run light. This means that whenever the start binary variable is energized the run light will be switched on. In other words whenever the machine is in operation the run light will be switched on. The ladder logic for rung 5 is shown in figure 10. This rung gives us the condition for filling state.

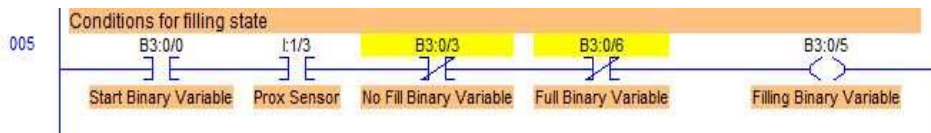


Figure 10. Ladder logic for rung 5

Here the filling binary variable is defined. This variable is connected with two open contacts, start binary variable and the proximity sensor. These sensors are connected to two normally closed contacts, no fill binary variable and full binary variable. Therefore when two open contacts are energized and the two closed contacts are in de-energized state the filling binary variable is energized. It means it will only be energized when the machine is running, the proximity sensor is on, and if the bucket is not full. The ladder logic for rung 5 is shown in figure 11. This rung gives us the condition for the fill light.



Figure 11. Ladder logic for rung 6

Here the filling binary variable is connected directly to the output variable of fill light. This means when the fill variable is energized the fill light is also enabled. In other words it means that when the box is filling the fill light will be on. The ladder logic for rung 7 is shown in figure12. This rung tells us when the solenoid valve should open. Here the filling binary variable is directly connected to the output variable solenoid valve.



Figure 12. ladder logic for rung 7

Hence when the filling variable is energized the output variable is also energized. In simple words it means that when the box to filling the solenoid valve is open.

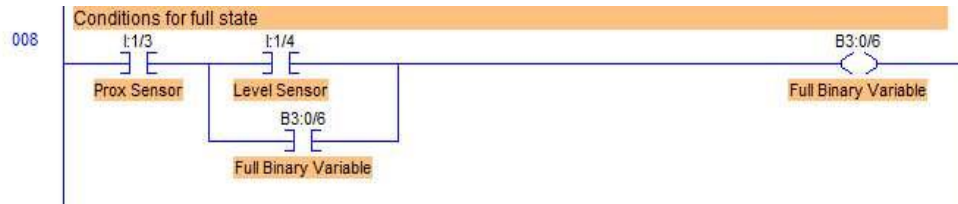


Figure 13.Ladder logic for rung 8

The ladder logic for rung 8 is shown in figure 13. This rung tells us when the box is full. Here the full binary variable is connected to the proximity sensor and the level sensor. The level sensor is then latched on with the full binary variable. This is done so that the full binary variable remains true once energized. In other words this means when the box is in place and is filled the box is full. The ladder logic for rung 9 is shown in figure 14. This is the rung gives the condition or for the full light.



Figure 14. Ladder logic for rung 9

Here the full variable and binary start variable are connected to the full light output variable. Hence when the full variable and the start variable are energized the full light will be switched on. In simple words it means that when the box is full and the machine is running the full light will be on. The ladder logic for rung 9 is shown in figure 15. This is the final rung and it tells us that the program has ended.



Figure 15. Ladder logic for rung 10

3.1. Complete ladder logic diagram for Dry Ice Packaging simulations

Combining all the rungs together shows the complete logic for dry ice packaging automation. Figure 16 shows the complete ladder logic diagram for dry ice plant which can be easily downloaded into PLC. When implemented in a programmable logic controller, the rules are typically executed sequentially by software, in a loop. By executing the loop fast enough, typically many times per second, the effect of simultaneous and immediate execution is obtained.

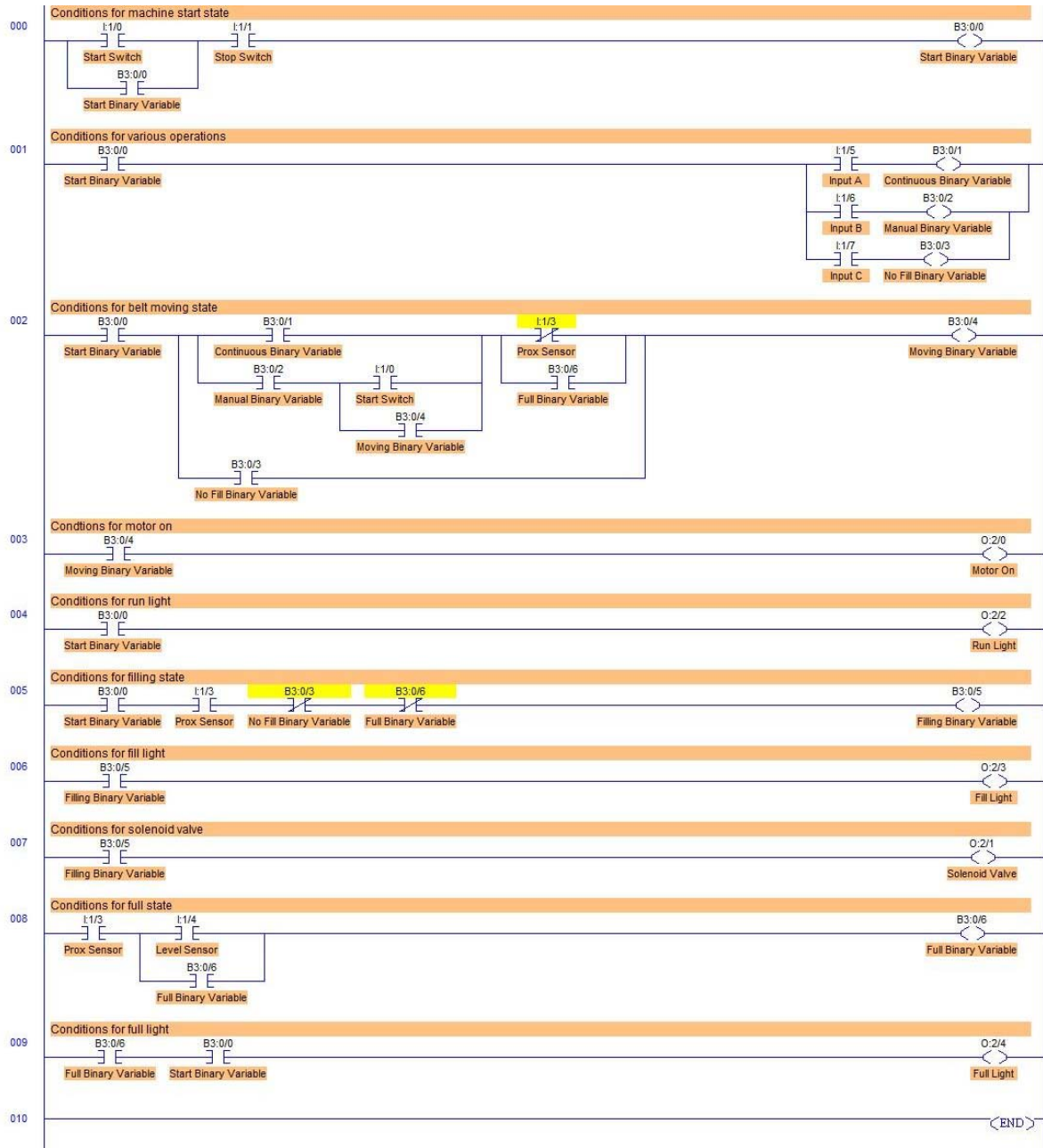


Figure 16. Complete Ladder logic Dry Ice packaging simulations

4. SIMULATIONS RESULTS

The ladder logic has been implemented and simulated in Windows XP based HP workstations connected in an Ethernet LAN. Figure 17 shows the screen of a dry ice plant simulation result for the proposed model [6]. The box is kept on a conveyor belt which moves with the help of a motor. When the box is in place the motor should stop and the solenoid valve should open and the box should start filling. This is done with the help of various sensors.

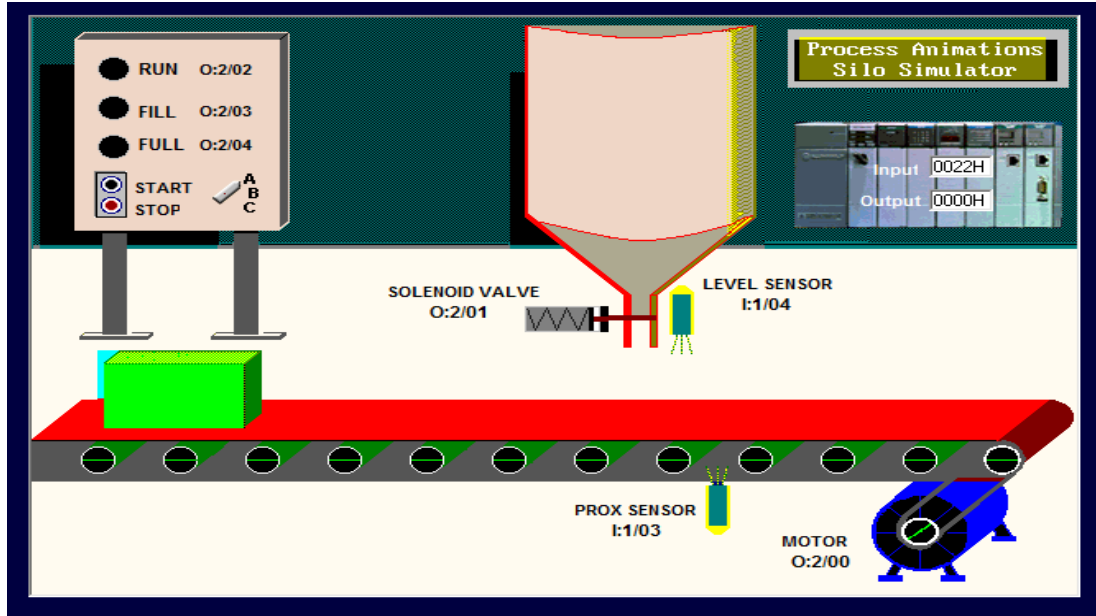


Figure 17. Dry ice plant Simulations

Also when the box is being filled the fill light should be on, when the program is running the run light should be on and when the box is filled the full light should be on. There are three switches that determine how the simulator should function. Using switch A as soon as the box is filled the conveyor belt should start moving. If switch B is selected then after the filling occurs, the box will only move once start button is pressed. Input c will only move the conveyor belt without filling the box as shown in Figure 18.

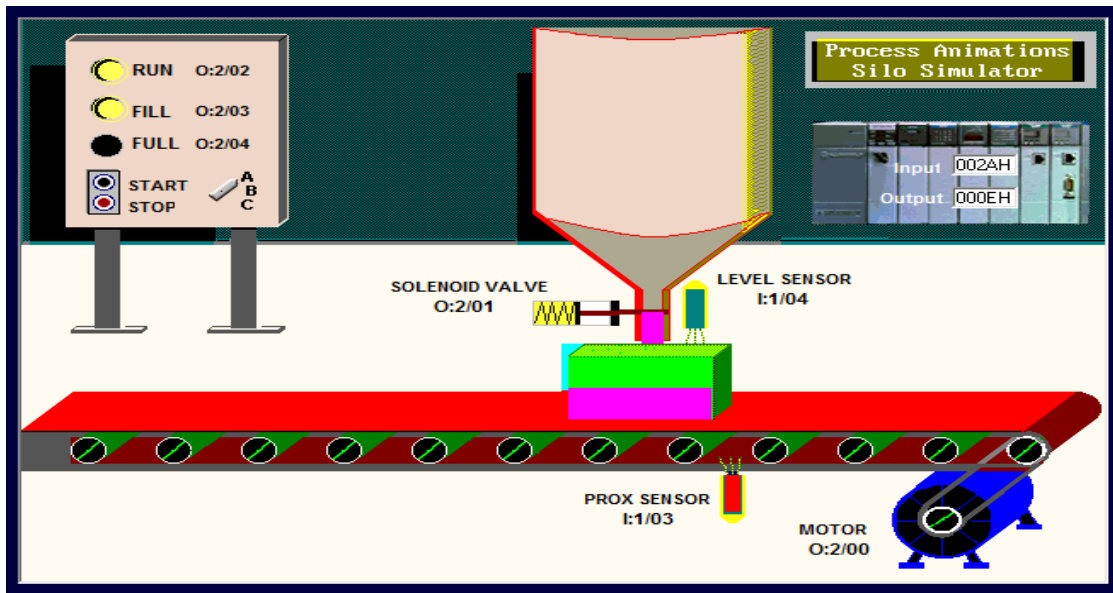


Figure 18. Packaging process simulation of dry ice

Figure 19 shows the box is full and signal from the level sensor will activate the motor. Using this approach, different kind of logic be tested, implemented and monitor continuous automated packaging process requirements.

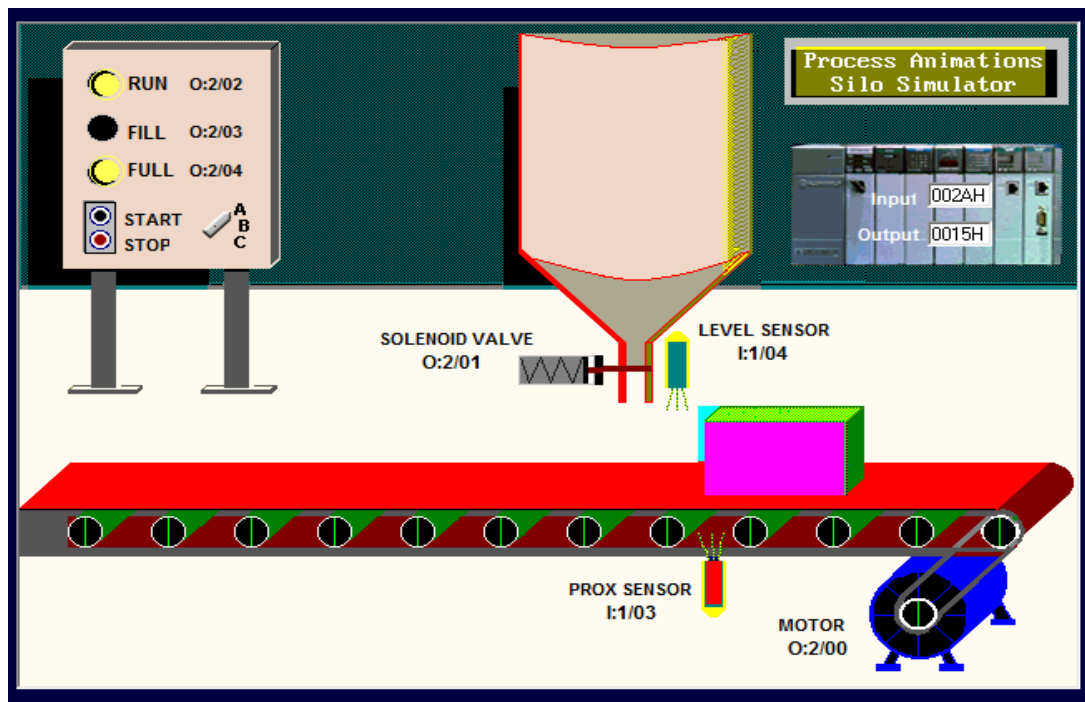


Figure 19. Final stage of packaging process.

5. CONCLUSION

SCADA simulations are an area of growing interest and one in which researchers face significant challenges in developing and testing solutions for various process. In response to these challenges the proposed research work allows the modern tool to be used for rapid SCADA simulation development and deployment for dry ice packaging. The plant functions or logics are implemented by simply downloading the ladder logic into available programmable logic controller. This logic can be modified and changed according to the need of the user. The scan time of every cycle can be adjusted as required. Improvements to the dry ice packaging platform focused on extending the proposed modeling environment to provide support for control and sensor messages sent across a control network. To support the expanded capabilities of the modeling environment, integration of the LogixPro simulator is implemented. . A practical implementation of this approach suggested in this paper was assessed based on sample simulations. Accordingly the proposed model can be implemented in real-time for large scale packaging of process control industries spread over geographically.

REFERENCES

- [1] Alvaro A. C´ardenas, Saurabh Amin, and Shankar Sastry. Research challenges for the security of control systems. In Proceedings of the 3rd conference on Hot topics in security, Berkeley, CA, USA, pp.: 6:1–6:6, 2008.
- [2] Andrew Davis, ‘Developing SCADA Simulations with C2 Wind Tunnel’, Thesis - Master of Science in Computer Science, Nashville, Tennessee.
- [3] Vinay M. Ijure, Sean A. Laughter, and Ronald D. Williams, “ Security issues in SCADA networks. Computers Security”, Vol. 25, No. 7, pp.:498 – 506, 2006.
- [4] A. Daneels and W. Salter, “What is SCADA? Int. Conf. on Accelerator and Large Experimental Physics”, International Conference on Accelerator and Large Experimental Physics Control Systems, 1999.
- [5] Frank D Peteruzella, “Programmable Logic Controllers”, McGraw-Hill, Third edition, 2005.
- [6] <http://www.thelearningpit.com/lp/logixpro.html>
- [7] <http://www.co2gasplants.com/dry-ice-plants.html>
- [8] <http://www.pacontrol.com/plc-ladder-logic.html>