

A MECHANICAL VENTILATOR SYSTEM CONTROLLED BY PROGRAMMABLE LOGIC CONTROLLERS

Al Emig and Sedig S. Agili
Electrical Engineering Program
Penn State University at Harrisburg
777 West Harrisburg Pike
Middletown, PA 17057

Abstract

In this paper, an example of the use of programmable logic controllers (PLC), computer programming, and computer interfacing, for teaching purposes and student projects implementation is presented. This is carried out through the design and implementation of a mechanical ventilation system that can be used on respiratory patients and controlled by a PLC. The physical control and operation of the ventilator are accomplished by the PLC via analog input and output modules located within the PLC chassis. Operator adjustment, mode selection, and patient monitoring are accomplished using a personal computer running a visual basic application designed specifically for this project. The operator interface mimics the look of a traditional ventilator control chassis. A serial data connection provides the communication path between the PLC and the personal computer. Using a PLC and a computer interface (HMI) as the respirator control system has distinct advantages over traditional microprocessor control systems. These include: the operation and adjustment of the respirator can be accomplished at a location other than the location of the respirator, alarm conditions can also be determined and evaluated from a remote location, reprogramming of the system can be easily accomplished onsite should newer respiratory modes become available. Through the use of a PLC and a HMI many different applications, similar to the one presented here, can be designed and implemented. This senior capstone design project was designed and implemented by an electrical engineering technology student as part of graduation requirements.

Introduction

At the Bachelor of Science level, Penn State Harrisburg offers BS EE, and BS EET, degrees. The Bachelor of Science degree in Electrical Engineering provides an opportunity for students to pursue interests in electrical and electronic circuits, including digital circuits and VLSI and its fabrication, microprocessors and their applications, electromagnetics, communications, control systems, digital signal/image processing and computer vision. The BS EET program provides similar experience however; its strengths include an applied, hands-on approach and extensive laboratory experience. Through a senior capstone design project, both curricula emphasize written as well as verbal communication and a teamwork approach among students to attain a common goal.

Thus, students in automation and robotics, microcomputers, communications, and senior capstone design project courses have ample opportunities to design and implement hands-on practical applications. Currently, several students are involved in implementing practical application projects such as WI-FI security robot, auto lander, and computer numerical controller (CNC) universal milling machine.

The project presented in this paper provides an example, for use in other electrical engineering and electrical engineering technology programs, of a practical application of a PLC and computer interfacing. The concepts and skills used herein were obtained while in the study of electrical engineering technology. This paper combines various aspects of engineering principles including: PLC programming, computer

programming, communications (computer interfacing), and mechanical design. This capstone design project was implemented by an electrical engineering technology student as part of the graduation requirements.

Various styles and types of mechanical ventilators (respirators) are currently available and used in hospital facilities throughout the world. Each respirator is designed to provide ventilation and/or oxygen to hospitalized patients whose respiratory system either no longer functions or has a reduced capacity. The early machines did not “blow” air into a patient but operated by placing the patient into a chamber with the head sticking out and then drawing a negative pressure on the patient’s body. This method is called negative pressure ventilation and the machine was called an iron lung[1]. This method was used through the mid-twentieth century and the machine was most notably used for people who had polio. Today, however positive pressure ventilation methods are used to provide respiratory support.

These machines connect to a patient’s airway via either a mouthpiece or an inter-tracheal tube and this method is called intermittent positive pressure ventilation (IPPV). In this method, when the machine cycles, air is pushed in under positive pressure to a degree greater than the patients could deliver by themselves, and exhalation is passive, using the recoil nature of the chest to let air be exhaled[2]. There are two forms or control schemes used in IPPV, one is pressure control, where the lungs are inflated to a predetermined pressure and the other is volume control, where the lungs are inflated until a predetermined volume has been reached. These two methods have various modes in which the pressures or volumes are controlled. Some of these modes are: controlled mechanical ventilation (CMV), assist/control ventilation (A/C), intermittent mandatory ventilation (IMV), and assisted mechanical ventilation[3] (AMV). The selection of these modes are determined by the respiratory therapist or physician based upon the patient’s physical capabilities. These modes will be discussed in

greater detail in a later section.

The typical construction of a respirator and its control system consists primarily of mechanical adjustments (knobs on regulators) or electrical adjustments (potentiometers) located on the chassis of the respirator. The newer types of electronic respirators, as shown in Figure 1 use potentiometers and internal microprocessors. In this type of respirator, the data is entered through the keypad located below the rectangular display area shown in the upper left of Figure 1. Additionally, the ventilator is connected to a rubber bag that simulates the patient's lungs. One hose is part of the inspiratory circuit that delivers air to the patient and the other is part of the expiratory circuit that returns air back to the ventilator. This ventilator is an example of a conventional volume ventilator; in this case a Puritan-Bennett model[2] 7200. However, adjustments to these systems must be made locally at the respirator control panel.

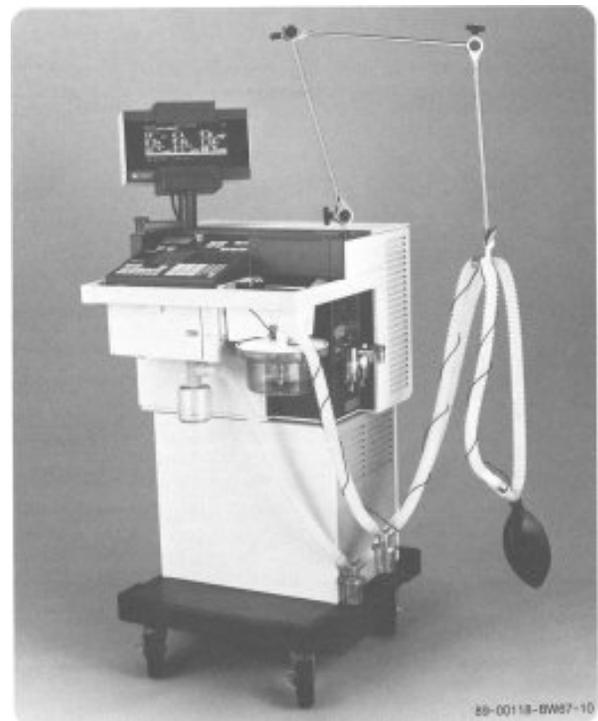


Figure 1: Modern ventilator showing operators panel (upper left portion of chassis).[2]

Using a PLC and a (HMI) as the respirator control system has distinct advantages over traditional microprocessor control systems. These include:

- The operation and adjustment of the respirator can be accomplished at a location other than the location of the respirator. The HMI can be located remotely by using wireless technology or by utilizing a hard connection via cabling.
- Alarm conditions can be determined and evaluated from a remote location.
- Reprogramming of the system can be readily accomplished onsite should newer respiratory modes become available.

In this paper, a respirator that is controlled by a PLC was constructed for the purpose of determining the operation of the HMI and PLC control system. The ventilator used was not a medical grade ventilator. This ventilator was constructed using industrial grade components. However, the HMI and PLC could qualify for medical grade operation. A computerized operator interface is used to adjust the respirator. This interface is a personal computer connected via serial cable to the PLC. Visual Basic was used to construct the interface. The interface provides complete functionality that would normally be provided only at the respirator control panel.

This paper is organized as follows. A detailed description of the PLC control system and the Visual Basic interface, as well as the respirator modes that are implemented in the project is described in the next section. The experimental testing procedures and their results are then described. Followed by a project summary and conclusion.

Detailed Description of the Ventilator System

As stated above, the control of the system is accomplished with a PLC. This PLC is an Allen-Bradley SLC 5/04. Programming of the PLC is accomplished using RSLogix. Figure 2

shows detailed block diagram of the mechanical and the electrical layout of the respirator used in this project. Note that other implementation can be employed. This PLC system consists of the following:

- Analog inputs: Two from main and airway pressure sensors and one from inhalation/exhalation flow transmitter.
- Analog outputs: One to control the inhalation/exhalation proportional control valve.
- Discrete outputs: Four outputs used to control the intermediary relays and subsequently the solenoid valves.

A. Analog Inputs to the PLC

Three analog inputs are required to monitor system pressures and airflow. Two Allen-Bradley 1746-NIO4V: Analog 2 Ch In/2 Ch Voltage Out cards are used to accomplish this task. The output from the three sensors is a 4-20 milliampere (mA) signal. The NIO4V card is configured by internal selector switches to enable this type of input. The electrical configuration of the two pressure sensors is a series loop consisting of a 24-volt direct current (dc) power supply, the pressure sensor, and the NIO4V card. The output from the flow transmitter is an isolated signal produced within the transmitter and does not require a separate 24-volt dc loop supply. However, a 15-volt dc supply is required to power the transmitter. The pressure sensors and the flow transmitter produce a linear output signal based upon the following criteria:

- 0 – 25 psig for the main pressure sensor.
- 0 – 6 psig for the airway pressure sensor.
- 2 – 200 liters per minute (L/min) for the flow transmitter.
- The analog outputs from these sensors are connected to three separate input channels on the NIO4V card. An analog to digital converter circuit within each input channel converts the analog input signal to a 0 – 16,384 bit digital representation based upon a 0 – 20 mA signal. Because the

analog input signal is 4 – 20 mA, a Scale with Parameter (SCP) instruction within the PLC is used for conversion for each of the three inputs.

The final output value from the SCP is in user defined engineering units thereby allowing the programmer, or other user, to more easily understand other portions of the program where this value is used. For example: it is easier to understand that 12.5 represents half-scale, or 12.5 psig from the pressure sensor, than to understand that a value of 8192 is half-scale.

B. Analog Output from the PLC

To control the proportional valve that modulates airflow to the patient, an output channel on the previously described NIO4V card is used. The output from this card is 0 – 10 volts dc. A value of zero volts fully closes the valve and a value of ten volts fully opens the valve. To provide control of the proportional valve, a proportional-integral-derivative (PID)

instruction within the PLC is used. The PID instruction provides a method to insure that the actual value of the process tracks the required value of the process. A brief description is as follows:

Within the PID instruction the programmer selects address locations used for the Process Variable (PV or actual value), Set Point (SP or required value), and Control Variable (CV or output). Tuning values for Controller Gain (P), Reset (I) in minutes per repeat, and Rate (D) in minutes, can be selected and adjusted based on the required response of the controlled element. Additionally, the PID function can be in either automatic or manual mode. In automatic mode the CV is varied based upon the tuning parameters selected and the error signal calculated by the PID instruction. In manual mode, the CV is set at a fixed value, selected by the programmer, of zero to 100 percent output and does not vary regardless of the calculated error signal.

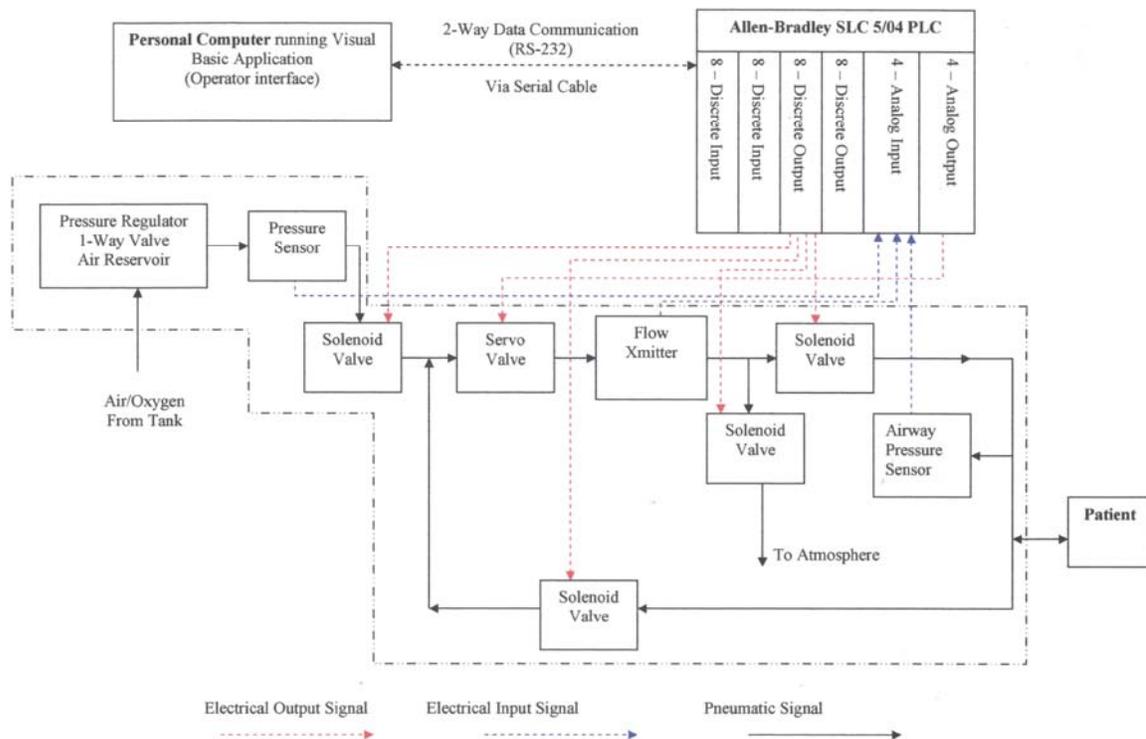


Figure 2: Detailed block diagram – electrical and mechanical.

The PID function will output a digital value of zero to 16383 representing an analog output on the NIO4V card of zero to five volts dc. Because the proportional valve requires zero to 10 volts dc for a valve opening of zero to 100 percent, an SCP instruction is used to modify the final output value from zero to 10 volts.

One issue that surrounds the use of the NIO4V card is that of maximum current output. This card can only supply a maximum current of 10 mA. The proportional control valve used in this project required approximately 600 mA of current for full valve opening at 10 volts dc. Therefore, a unity gain non-inverting buffer amplifier was constructed and placed between the output of the NIO4V card and the proportional control valve. This amplifier was constructed using a National Semiconductor LM386N-1 op-amp and a TIP3055 NPN transistor.

C. Discrete Outputs from the PLC

To control the actuation, opening and closing, of the four solenoids used on this project, an Allen-Bradley 1746-OB8: 8-output (trans-SCR) 10/50 Vdc is used. The outputs from this card are either on or off and do not allow for intermediary states. This card does not produce an output voltage from within the PLC rack. Therefore, an external 24-volt dc power supply is connected to the card to produce the necessary output voltage. Because the solenoids used in this project require 120-volts alternating current (ac), intermediary relays are placed between each output and its associated solenoid. Additionally, by placing these relays between the NIO4V card and the solenoids, inductive kickback caused when the solenoids energize and de-energize will be minimized.

To operate each solenoid a 120-volt ac power source is connected as a series loop through the normally open contacts on each relay and then to each of the solenoids. Therefore, when a 24-volt dc relay is energized by the output of the 1746-OB8 card, the solenoid connected to that particular relay will also be energized.

D. PLC Control System

The PLC that is used to control the mechanical ventilator is an Allen-Bradley 1747-L541: SLC 5/04 programmable logic controller and is programmed using Allen-Bradley RSLogix software. During the design phase of this project, it was determined that three user selectable ventilatory control modes would be implemented within this project. These three modes demonstrate the flexibility and functionality of this project. In the following, a description of each control mode and the methodology used in their programming is presented. A detailed description of each ventilator mode can be found in[3,4].

1) Volume-Controlled Ventilation (VCV): In this mode, all breaths are delivered by the ventilator. The ventilator delivers a preselected ventilatory rate, tidal volume, and inspiratory flow rate, independent of spontaneous effort on the part of the patient. When this mode is selected the following data values, obtained from the Visual Basic interface, are utilized:

- Preset Insp. Min. Vol. (L/min) – preset inspiratory minute volume. This is the volume required for each inspiratory breath and is the set point (SP) for the PID instruction.
- Breaths Per Minute – this is the total number of breaths allowed in one minute consisting of both the inspiratory phase and the expiratory phase. This number is used to divide 60 (seconds) into a value representing how long a breath should last. It is then placed into a timer that controls the total length of time for each breath.
- Insp. Time % - inspiratory time percent. This value is the percentage of the length of the inspiratory phase of each breath. For example: if this control is set at 50 percent and the breaths per minute is set at 10, the inspiratory phase will last three seconds and the expiratory phase will last three seconds. If however, this control is set at 25 percent and the breaths per minute stays the same, the inspiratory phase will last 1.5

seconds and the expiratory phase will last 4.5 seconds. Within the PLC the inverse of this number is calculated and multiplied by the breaths per minute calculation and is placed into a timer that controls the inhalation and exhalation solenoid valves.

- Pause Time % (Insp. Pause) – pause time percent on the inspiratory phase. This value is a percentage of how long, during the inspiratory phase, the inhaled air will remain in the patient’s lungs prior to allowing the expiratory phase to begin. For example: if breaths per minute is set at 10, Insp. Time % is set at 50, and Pause Time % is set at 10 the inspiratory phase will last for three seconds, there will be a pause for one second (when both the inhalation and the exhalation solenoids are closed), and the expiratory phase will last for three seconds. This number is multiplied by the breaths per minute value and placed into a timer that de-energizes the inhalation and the exhalation solenoid valves.
- Expired Minute Volume (L/min) – value not adjustable by user, writes to data display area on the Visual Basic interface. This value is the actual airflow through the airflow transmitter. This is the PV value used in the PID function and is controlled by the proportional control valve.
- Insp. Press. Level (Above PEEP) – inspiratory pressure level above the positive end expiratory pressure. This control is not used in VCV mode and is therefore “dimmed” on the Visual Basic interface.

Other controls and displays, as shown on the Visual Basic interface, are also available in this ventilator mode. However, these are general types and can be used for any of the three modes of ventilatory support. These controls and displays are discussed in a subsequent section of this report.

2) Synchronized Intermittent Mandatory Ventilation (SIMV): In this mode the machine breaths are delivered at a set rate and volume or pressure synchronized with the patient’s

inspiratory effort. This is achieved by the use of an “assist window”. This “window” opens at intervals determined by the set SIMV rate and remains open for a specified period. If a patient-generated breathing effort is detected while this window is open, a mandatory breath is delivered. If no patient effort is detected during this time, the ventilator delivers a mandatory breath. This mode of control functions as described previously in the VCV mode and utilizes the same six Visual Basic controls and display. In this mode however, the inhalation airflow is monitored and when it decreases to 25 percent of peak flow, the expiration phase begins. In addition, when this mode is selected the following control appears on the Visual Basic interface.

- SIMV Breaths per Minute – this control value is set lower than the main breaths per minute control. In this mode, the airway pressure sensor is monitored to determine if the patient is trying to inhale. This is evidenced by a negative pressure registering on the pressure sensor. The monitoring of the pressure sensor occurs during a time established by dividing 60 (seconds) by the main breath per minute control value and placing this value into a timer. If during this time a negative pressure is sensed by the airway pressure sensor, a mandatory breath is initiated. The total length of the inspiratory and the expiratory phase is calculated based on the SIMV breaths per minute control and placed into a timer. This timer value then actuates the inhalation and the exhalation solenoid valves. After this mandatory breath, the patient could breath spontaneously for the remainder of the SIMV cycle. However, if the system does not detect an effort (negative going pressure) during the monitoring phase, the system automatically initiates a breath at the end of this period.

3) Pressure-Controlled Ventilation (PCV): In this mode, the ventilator delivers all breaths in a pressure-limited and time-cycled manner, with

no possibility for patient triggering. When this mode is selected the following data values, obtained from the Visual Basic interface, are utilized.

- Preset Insp. Min. Vol. (L/min) – preset inspiratory minute volume. This control is not used in PCV mode and is therefore “dimmed” on the Visual Basic interface.
- Insp. Press. Level (Above PEEP) – inspiratory pressure level above the positive end expiratory pressure. This is the pressure required for each inspiratory breath and is the set point (SP) for the PID instruction. This value is added to the user selected PEEP level value and the control system tries to maintain this pressure level during the inspiratory and the expiratory breathing phases.
- Airway Pressure (cm H₂O) – airway pressure measured in centimeters of water column. This value not adjustable by user. It writes to a data display area on the Visual Basic interface. This value is the actual airway pressure sensed by the airway pressure sensor. This is the PV value used in the PID function and is controlled by the proportional control valve.

The other controls including: Breaths Per Minute, Insp. Time %, Pause Time % (Insp. Pause), and Expired Minute Volume (L/min) function is the same manner as previously described.

In addition to the aforementioned controls and displays, the following items are obtained and/or displayed on the Visual Basic interface.

- Working Pressure (PSI) – this is the main supply pressure to the mechanical ventilator and the value is not adjustable by user, writes to data display area on the Visual Basic interface.
- Control Type – this drop-down list box allows for the selection of the three ventilator control modes. Each of these modes sets a different bit within the PLC

and subsequently allows various portions of the program to function.

- Respirator ON/OFF button – this turns the outputs from the PLC on and off. Additionally, when the ventilator is turned on, it places the PID function in automatic and when the ventilator is turned off is places the PID function in manual with a CV value of zero.
- Lower and Upper Alarm Limit – the values selected for each of these controls determine the set point at which the Exp. Min. Vol. Low Alarm and Exp. Min. Vol. High Alarm will annunciate. These alarms are based upon the actual airflow from the airflow transmitter.
- Trig. Sensitivity (Below PEEP) – trigger sensitivity below the positive end expiratory pressure. This control determines the pressure at which a patient’s effort can start an inspiratory cycle. This pressure is calculated by subtracting the trigger sensitivity from the PEEP value. When this pressure level is obtained the ventilator will cycle.
- PEEP – positive end expiratory pressure. This control is used to maintain an internal pressure within the patient’s lungs. This is accomplished by examining the airway pressure sensor and when its value reaches that of the PEEP value, the exhalation solenoid valves close.

The following are values not adjustable by the user. These values are written to the data display area on the Visual Basic interface.

- Insp. Tidal Vol. mL – inspiratory tidal volume measured in milliliters. This quantity is calculated by monitoring the inhalation flow. The inhalation flow value is then divided by the inhalation time.
- Exp. Tidal Vol. mL – expiratory tidal volume measured in milliliters. This quantity is calculated by monitoring the exhalation flow. The exhalation flow value is then divided by the exhalation time.
- Exp. Min Vol. L/min – expired minute volume in liters per minute. This is the

actual airflow through the airflow transmitter. This is a secondary display.

- Peak Pressure (cm H₂O) – peak pressure measured in centimeters of water column. This quantity is the maximum airway pressure measured during each ventilatory cycle. This is obtained by examining the input pressure and if it is greater than the previous pressure, this value is stored. This value is cleared at the end of each expiratory phase.
- Pause Press. (cm H₂O) – pause pressure measured in centimeters of water column. This quantity is the measured air pressure that is maintained within the patient's airway when the Pause Time % control is set greater than zero.
- Average Airway Press. – average airway pressure measured in centimeters of water column. This quantity is a running average updated every tenth of a second.

The following items are the ventilator alarms programmed in the PLC. Each alarm sets a bit in word B3:50. This word is then transferred to the Visual Basic interface which then parses out each bit to its associated variable name. Reverse logic is used in this portion of the PLC program i.e. when a bit is set it is not in an alarm condition.

- Exp. Min. Vol. Low Alarm – expired minute volume low alarm. This alarm is adjusted by the Lower Alarm Limit control on the Visual Basic interface. When the airflow volume is less than this preset value for longer than two seconds the alarm bit is cleared. Clearing the bit causes the Visual Basic alarm annunciator to change from green to red.
- Exp. Min. Vol. High Alarm – expired minute volume high alarm. This alarm is adjusted by the Upper Alarm Limit control on the Visual Basic interface. When the airflow volume is greater than this preset value for longer than two seconds the alarm bit is cleared. Clearing the bit causes the Visual Basic alarm annunciator to change from green to red.

- Upper Airway Pressure Alarm – This alarm is adjusted by the Upper Pressure Limit control on the Visual Basic interface. This alarm functions similarly to the expired minute volume high alarm.
- Airway Disconnect Alarm – This alarm examines the airway pressure sensor and if the pressure is less than or equal to zero for longer than 15 seconds the alarm bit is cleared. Clearing the bit causes the Visual Basic alarm annunciator to change from green to red.

E. Visual Basic Interface and Program

Visual Basic 6.0 was used to create and program the user interface. This interface is used to make all adjustments to the ventilator. In addition, it provides the user with various informational displays and alarms. Because the function of each control, display, and alarm was described in Section II-D of this report a re-examination of every one of these is not required. This section of the report will focus on one particular item from each of the following sections of the Visual Basic interface: Overview and Construction – principles of Visual Basic programming and construction methodology, Communications – used to read and write information to the PLC, Slider Controls and Data Displays – sliders are used to adjust user selected parameters: data displays are read only and are used to display information, and Alarms – displays alarm condition by changing color.

1) Overview and Construction: Visual Basic is an event driven programming language that allows programmers to easily construct graphics that can be used within a computer program. Event driven means that when a user physically interacts with the running program, an action within the program takes place, i.e. pressing a button on the interface may cause a picture to appear.

For the project discussed in this paper the interface is modeled after the control panel of a Siemens Servo Ventilator Model 900C originally manufactured in 1984. Slight

modifications were made from the original control panel in the construction of the interface for this project. These include: replacing control knobs with sliders, changing analog gauges to digital gauges, adding a data display area to show relevant information continuously, and adding an alarm annunciator.

2) Communication: Communications between the PLC and the Visual Basic interface is accomplished via an RS-232 serial communications protocol. Both the PLC and the computer, acting as the operator interface, must have serial ports. All data transfer, both read and write, is accomplished within Visual Basic (however, the PLC communications parameters (baud rate, stop bits, parity bits, etc) must match those of the Visual Basic program). Serial communications are established by implementing an Active X control within Visual Basic during programming. The Active X control used in this project is ABCComSec.ocx from Consolitech Corp. This company was found on the Internet and offered a demonstration version of their PLC communication program. The implementation and programming of this control is virtually identical to standard Visual Basic controls, thereby the programmer does not need advanced skills to learn how to use this control.

3) Slider Controls and Data Displays: Slider controls are used in place of a rotating dial or knob. Minimum values occur when the slider is in its full leftmost position and maximum values occur when in its full rightmost position. Each slider is programmed for these minimum and maximum values.

The data displays are used for informational purposes only and cannot be accessed by the program operator.

4) Alarms: Alarms are used to indicate problems that occur with the patient's breathing. Alarms are shown at the bottom of the Visual Basic interface. They are represented by either a green or a red circle. Green indicates a non-alarm condition and red indicates an alarm condition.

Each alarm, with the exception of the Airway Disconnect alarm, can be set by the user. The alarm set points are adjusted by using the slider associated with that particular alarm. The Airway Disconnect alarm is activated when the airway pressure sensor detects a pressure value less than or equal to zero for longer than 15 seconds. As stated previously, all control including the alarms are done within the PLC.

Experimental Results

A performance evaluation of the mechanical ventilator project was conducted in different phases. The first phase was the construction and evaluation of the Visual Basic interface. The second phase was the programming of the PLC. The third phase involved an evaluation of the data exchange between the Visual Basic interface and the PLC. The fourth phase consisted of hardware interfacing between the analog sensors and the PLC. The final phase incorporated the complete mechanical ventilator hardware and an evaluation of the performance of the total system: PLC, Visual Basic interface, and the finalized mechanical portion of the project.

The first phase of the project consisted of insuring that the Visual Basic interface was a true representation of a mechanical ventilator's operator panel. It was also important to make the interface user friendly and not over-complicated. This resulted in the final interface shown in Figure 3. The methodology and function of the interface are described in the previous section of this report.

The second phase of the project involved a detailed analysis of the operation of the Siemens Servo Ventilator 900C on which this project was modeled. The text *Respiratory Therapy Equipment*⁵ was used to provide the necessary information and detailed descriptions for each ventilator mode and control utilized in this project. This text was used as a reference to ensure that the PLC functioned as intended. During this phase, finalization of the PLC program could not be accomplished because the

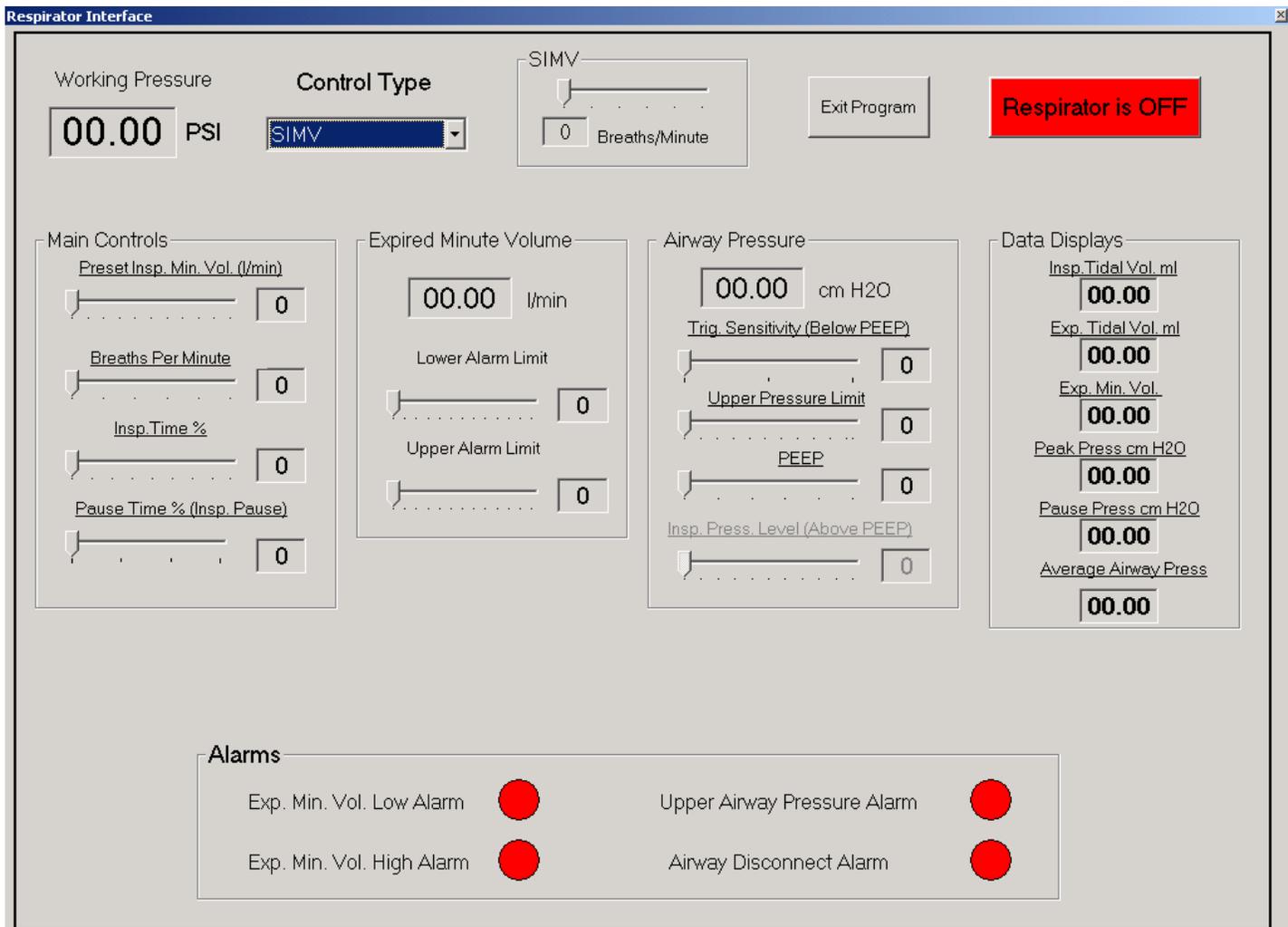


Figure 3: Final Visual Basic interface.

Visual Basic interface and the mechanical hardware were not connected to the PLC.

The third phase of the project required connecting the PC, which was running the Visual Basic interface, to the PLC. During this phase, modifications were made to the Visual Basic program. It was determined that the read operations were taking longer than expected, thereby causing the data displays on the Visual Basic interface to “blank out”. This was remedied by reading a “block”, i.e. F8:0-F8:25, of data at one time instead of reading individual data words, i.e. F8:0, F8:1, F8:2...F8:25 one at a time. Another modification made during this phase involved reading the current PLC data

when the Visual Basic interface was first started. Because the Visual Basic interface can be started or stopped at any time without affecting the PLC operation it became necessary to read all PLC data upon startup of the Visual Basic program. This was accomplished by programming a startup flag within the Visual Basic program.

This operates by initiating a read cycle as soon as the Visual Basic application starts. This initial read cycle updates all sliders, data displays, and the ON/ OFF indicator button. Once this operation is completed, the Visual Basic program functions as previously described. Upon completion of this phase of the

project the Visual Basic interface program was finalized.

The fourth phase of the project involved testing the analog input and analog output portion of the PLC program. To test the analog inputs a variable dc power supply was connected to each channel on the analog input card (I:5.0, I:5.1, and I:6.0). The voltage was varied from 0 to 10 volts and was monitored in the PLC while the PLC was in the run mode. It was determined that the Input Min./Input Max. parameter values and the Scaled Min./Scaled Max. parameter values were interchanged. This problem was corrected and the analog inputs were found to be functioning correctly. Next, the analog output, O:5.0, was tested. To test the analog output a digital voltmeter was connected to the PLC. Because the analog output is used to control the proportional valves, a PID instruction within the PLC is used. To vary the output voltage the PID instruction is placed in its manual mode and the CV is adjusted from 0 to 100 percent. However, the PID instruction is only capable of providing a maximum analog output value of five volts dc. Therefore, a SCP instruction must be used to increase the output voltage to ten volts. This was not discovered until this phase of testing. Once the SCP instruction was included in the PLC program, the analog output functioned as required.

The final phase of the project consisted of testing and evaluating the final PLC controlled mechanical ventilator system. After calibrating the two pressure sensors, connecting all mechanical and electrical hardware to the PLC, and connecting the air supply to the system the PID control algorithm was adjusted to provide optimal control of the airflow within the system. The final tuning parameters are: Controller Gain (P) equal to 250.00, Reset (I) equal to 0.03 minutes per repeat, and Rate (D) equal to zero (this parameter was not used). During testing of the system, additional logic was added to the PLC to “filter” or smooth out variations in some data display areas. Filtering was accomplished by moving the analog input data into a first-in-first-out (FIFO) PLC instruction and performing

a “running” average on the data. The filtering was done on the analog input values that correspond to the data displays. It should be noted that only the displays were filtered, the analog input values used for system control continued to use the non-filtered data.

Conclusion

The project presented within this paper functioned as expected. The Visual Basic interface and its program allowed the user to interact and evaluate data from the mechanical ventilator. Full ventilator functionality was provided through the interface. In addition, all adjustments to the ventilator are made from a remote location by using RS-232 communication protocol and a serial cable or can be made via wireless technology. The ventilator control system, provided within the PLC, functioned according to the descriptions obtained in the reference.[5] Using Visual Basic as an interface and a PLC as a control system is a viable alternative to existing ventilator systems.

The problems and difficulties that were encountered with the project were mostly mechanical in nature. The original proposal was made in the belief that an actual medical ventilator could be obtained. From this, modifications would be made to interface with the PLC and subsequently the Visual Basic program. However, because an actual ventilator could not be obtained at a reasonable cost, the ventilator used was designed and manufactured by electrical engineering technology students using industrial grade components. Note that, the HMI and PLC could qualify for medical grade operation. Despite this problem, the system that was constructed provided a reasonable representation of a mechanical ventilator. The authors’ feel that the PLC controlled mechanical ventilator application, given in this paper, can be used as an example for PLC and interface projects in other electrical engineering and electrical engineering technology schools.

References

1. Robert R. Kirby, Robert A. Smith, and David A. Desautels, Eds. Mechanical Ventilation. New York: Churchill Livingstone, 1985, pp. 3-6.
2. Martin Lawrence. (1999). *Pulmonary Physiology in Clinical Practice*. [Online]. Available: <http://mtnsinai.org/pulmonary/books/physiology>
3. Azriel Perel and M. Christine Stock. *Handbook of Mechanical Ventilatory Support*. Baltimore: Williams & Wilkins, 1991, pp. 8-19, 28, 93-95.
4. Dean R. Hess and Robert M. Kacmarek. *Essentials of Mechanical Ventilation*. New York: McGraw-Hill, 1996, pp. 17-23, 41-42.
5. Steven P. McPherson. *Respiratory Therapy Equipment 4th ed*. St. Louis: C.V. Mosby Company, 1990, pp. 321-334.

Biographical Information

Aloysius (Al) Emig, Jr., is a 2003 graduate of Penn State Harrisburg's Electronic Engineering Technology program. His experience includes almost 20 years in process control and instrumentation working with PLCs and HMIs. He has worked in various manufacturing areas including steel, paper converting, food and beverages, and pharmaceuticals. He is currently a Control Engineer at GES Technology, Inc. a systems integration company that specializes in manufacturing process and automation solutions.

Sedig S. Agili received his BS, MS, and Ph.D. in Electrical and Computer Engineering from Marquette University in 1986, 1989, and 1996, respectively. As a student, he was awarded fellowships from Marquette University and the U.S. Department of Education. Upon receiving his Ph.D., he joined the faculty at Marquette University where he taught several courses in electrical engineering and conducted research in the area of electro-optic devices, fiber optic communication and fiber optic sensors. In fall of 2001, he joined the electrical engineering and electrical engineering technology programs at Penn State University, Capital College. Currently he is an associate professor teaching and conducting research in electronic communications, fiber optic communications, fiber optic sensors and signal integrity of high-speed interconnects. He has authored numerous articles published in journals and conference proceedings, and made presentations at many conferences and seminars. He also worked for Astronaut Corporation of America in Milwaukee, Wisconsin where he was involved in designing optical projection and heads-up display systems. He is a senior member of the Institute of Electrical and Electronic Engineers, American Society for Engineering Education, and Sigma Xi.