The Simulation of the Control of an Industrial Mixer using PLC

Okoli F. I., Onubogu J. O., Okezie C. C., Okorogu V. N.

Abstract- This paper is as a result of the need for an effective, reliable and highly flexible method for the control of processes within an Industrial environment. The objective of this study is to simulate the control an industrial mixer and by doing this we can develop a program that can be used for the actual implementation of such a model. This mixer is controlled using a programmable logic controller and programmed using ladder logic language.

I. INTRODUCTION

Logix Pro 500 was developed by Bill Simpson in 2001, the Logix pro 500^{TM} is a windows-based application used for developing control programs for use within a programmable logic controller. [8] Logixpro 500^{TM} provides an integrated development environment (IDE) which enables us to write a ladder logic application program suitable for the operation of the mixer. It also provides a collection of models for programmable logic circuits from many different manufacturers such as Allen-Bradley and Siemens so that the application program can be tested with different models of programmable logic controllers with a view to choose the best suitable for an application ladder logic control application program is developed using the Logix pro 500^{TM}.It also provides a simulator to test the application which has been developed.

It is designed as an interactive, educational tool developed for the acquisition of programming skills commonly used in the PLC control of process equipment and systems. There are many in built functions in the Logix pro 500^{TM} software which makes programming the PLC much easier. It also has an SLC-500 emulator built into it that translates instructions to machine language. In addition to the basic input output relays, there are counters, timers and other special functions. In this paper work, we use timers in addition input relays to the make contact and the output relay to break contact. The software also has an integrated visualization with already created modules of different machines, plants and processes.

The PLC simulator: The simulation is used to recreate the operation of a real-life programmable logic circuit in software. The simulation is done using the software called Logix pro 500^{TM}. Logixpro 500^{TM} provides an integrated development environment (IDE) which enables us to write a ladder logic application program suitable for the operation of the mixer. It also has an emulator capable of emulating the instruction-set of a real programmable logic controller. It also provides a collection of models for programmable logic circuits from many different manufacturers such as Allen-Bradley and Siemens so that the application program can be tested with different models of programmable logic circuits with a view to choose the best suitable for an application. During the simulation, the action of the mixer can be viewed so that the program execution can show if it performs as desired and any abnormally can be detected and the changes made. Through the use of this simulator, a real-time control of the mixer can be virtually created and the properties of the mixer can be varied to suit application and purpose.

Justification of the Study

This paper is justified for the following reasons:-

- Programmable logic controllers serve as the main form of control for manufacturing today.
- The use of Programmable logic controllers for Industrial mixing will enable flexibility thereby allowing the manufacturer to meet a customer's specific demand much quicker.
- A programmable logic controller which has been programmed using ladder logic can be applied to highly customized systems so that the cost of a packaged PLC is low compared to the cost of a specific custom-built controller design.
- The programmable logic controller which has been programmed using ladder logic can employ different techniques to perform a task which could lead to increased efficiency where some techniques are not economically viable.
- A programmable logic controller which has been programmed using ladder logic can be used to monitor the output at each stage in the production process and to make adjustments in a timely fashion.

- ***Scope of the work***

This paper uses the model of an Industrial mixer which has already been developed in Virtual Machine^{TM}. The modelled components include pumps, thermostat, heater, level sensors, heater, the electric motor, the indicators and the liquids. A model of each of the component has been developed individually and then articulated. These components are to be controlled by the action of the programmable logic controllers which is to takes in field-inputs from these components and then give appropriate control signals as the field outputs. The software program which is used to perform the control is developed in ladder logic language using bottom -top approach (module by module) i.e. individual programs that will run the motor, the sensor, the heater will be developed individually and articulated.

Revised Version Manuscript Received on December 12, 2015.

Okoli F.I, Electronics and Computer Engineering, Nnamdi Azikiwe University, Awka (Anambra), Nigeria
Onubogu J.O, Electronics and Computer Engineering, Nnamdi Azikiwe University, Awka (Anambra), Nigeria
Okezie C.C, Electronics and Computer Engineering, Nnamdi Azikiwe University, Awka (Anambra), Nigeria
Okorogu V. N, Electronics and Computer Engineering, Nnamdi Azikiwe University, Awka (Anambra), Nigeria

The ladder logic program is developed using Logix pro 500™.The program is also simulated with Logix pro 500™ and the test results obtained are displayed on a tabular form.

- ***Block Diagram and Description***

The main steps that were undertaken to implement this paper are:

1. **Modelling the machine and its operation**: - Here we had to decide the equipment whose properties we wish to control and it's method of functioning. For this particular paper we wish to control the action of the Industrial mixer whose operation involves valves, sensors, electric motors, and the liquids. These properties are modelled using Virtual Lab software which provides Library of Parameterized modules which represent the actual behaviour of the components of the Industrial Mixer.
2. **Assignment of the input and outputs**:- Here all the input and output devices which are intended to be connected to the PLC are identified . The input devices are assigned input addresses while the output devices are also assigned output addresses.

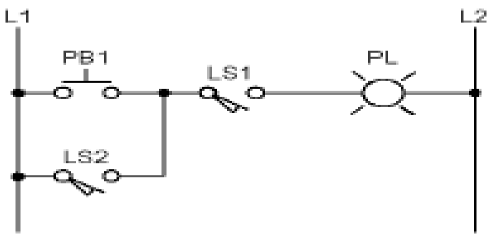


Fig2.9: A Simple Ladder Circuit

3. **Writing the Program**:-In this process a ladder diagram program which performs the control logic is written.
4. **Programming into Memory**:-Here first of all an I/O generation is done to prepare the system configuration of the programmable logic controller. Then the program is entered into memory either by a programming console or a computer -aided ladder software tool. It also involves debugging and checking for coding errors and the simulation of the whole operation in done in software.
5. **Running and Testing**: - This involves simulating the action of the operation of the Programmable logic controller within the model Industrial mixer in software and obtaining the test results in a tabular form.

Software Subsystem Design

The actual software is to be written in Ladder Programming Language. The mixer is designed to take in two liquids from the input pumps, stir and heat them to a mixture before collecting the mixture at the output pumps. All operation instructions come from the programmable logic controller to either switch ON/OFF any component of the mixer subsystem depending on its state or the state of another component. The mixer operates by taking in two different liquids from the input pumps one after the other and then performing a heating and stirring operation on this liquid within the vessel. The mixture collected from the output pump and the whole process restarts. The ladder diagram is the traditional way of representing electrical sequences of operations.[3] These diagrams represent the interconnection of field devices in such a way that the activation, or turning ON, of one device will turn ON another device according to a predetermined sequence of events. An example of a ladder diagram is shown

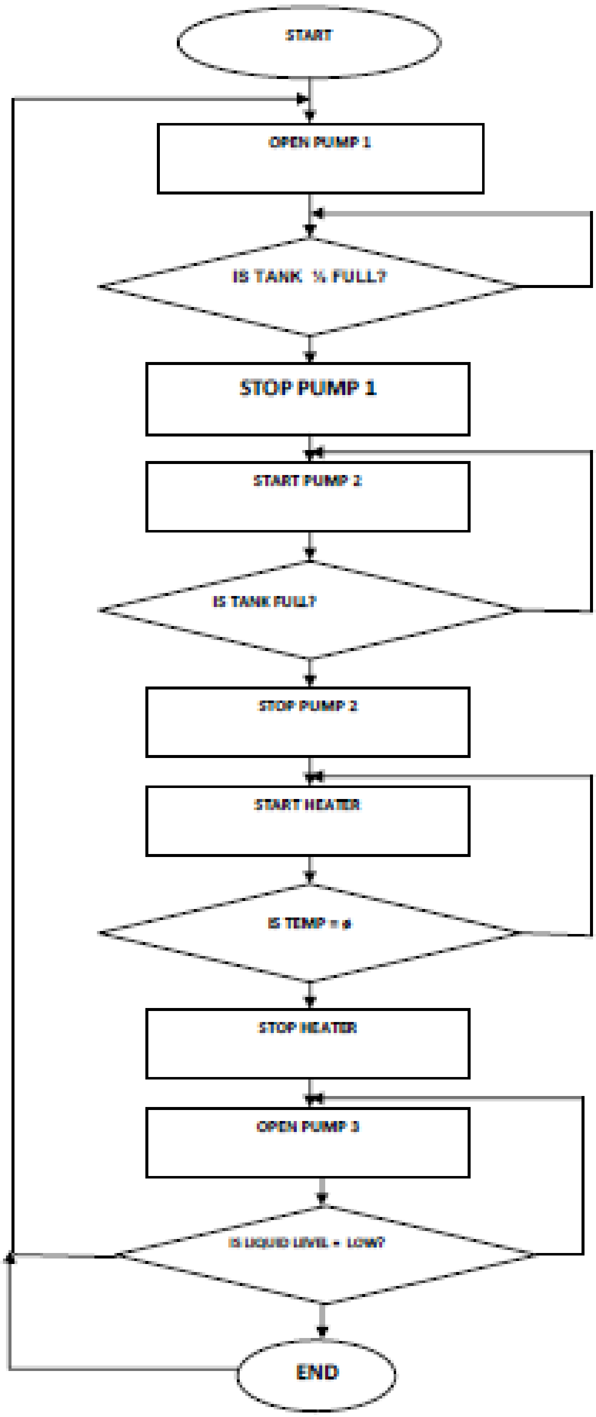
Ladder diagrams are easy to use and interpreted and are widely accepted in the industry. The programmable logic controller can also implement the ladder diagram so that control operations are performed more reliably. The control program used in the programmable logic controller is developed using ladder logic.

The CPU reads the status of inputs, energizes the corresponding circuit element according to the program, and controls a real output device via the output interfacesThe PLC system provides a design environment in the form of software tools running on a host computer terminal which allows ladder diagrams to be developed, verified, tested, and diagnosed. First, the high-level program is written in ladder diagrams. Then, the ladder diagram is converted into binary instruction codes so that they can be stored in random-access memory (**RAM**) or erasable programmable read-only memory (**EPROM**).[5]

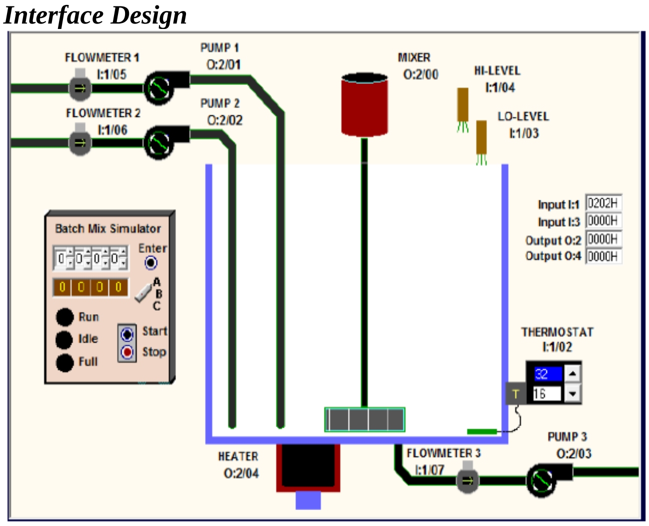
The actual software is to be written in Ladder Programming Language. The mixer is designed to take in two liquids from the input pumps, stir and heat them to a mixture before collecting the mixture at the output pumps. All operation instructions come from the programmable logic controller to either switch ON/OFF any component of the mixer subsystem depending on its state or the state of another component. The mixer operates by taking in two different liquids from the input pumps one after the other and then performing a heating and stirring operation on this liquid within the vessel. The mixture collected from the output pump and the whole process begins again. The program starts with the opening of PUMP1 to pour the first liquid into the vessel. A Timer is also set and with this Timer we can vary the quantity of the first liquid which we want to put into the vessel. This is made possible with a flow meter because the product of flow rate and time gives an exact amount of the liquid we wish to pour in first from PUMP1.

When the Timer completes the timing operation, it will deactivate PUMP1 and also open PUMP2 to pour in the second liquid until the containing vessel is FULL. The hi-level sensor usually indicates that the tank is FULL and at the same time stops PUMP2.It also starts the action of both the MIXER and the HEATER. The MIXER and HEATER must operate together at the same time in an AND operation i.e. starting one also starts the other and stopping one also stops the other. A temperature sensor is used to stop the HEATER and the MIXER. This sensor is active HIGH and usually starts PUMP3 in order to let out the mixture from the containing vessel. An empty containing vessel is usually indicated by the low-level sensor which also starts PUMP1 in order to repeat the mixing process all over again.

The system is also designed to empty all the liquid within the containing vessel whenever the STOP button is pushed. This is done with the help of a counter which toggles between states and counts pulses in order to keep PUMP3 open until the low-level sensor is activated. Also pushing the START button will RESET the entire system to begin operation from the beginning. The system is also programmed to indicate when it is IDLE which occurs when it is emptying the vessel after the STOP button has been pushed.



Interface Design



ACTUAL Simulation of the Mixer in Logix pro 500™

The interface design deals with how the input and output devices are assigned memory addresses according to input and output interfaces to which they are connected on the programmable logic controller. The interface design is determined by the particular programmable logic controller which is chosen for the control operation. For the control of the batch mixer we have chosen to use an 8-input discreet module with part number 1746I*8.

The table shows the different input and output device interface addresses and their description.

INPUT ADDRESS	DESCRIPTION	OUTPUTADDRESS	DESCRIPTI ON
I:1/00	START (NO,PB)	0:2/00	MIXER
I:1/01	STOP (NC,PB)	0:2/01	PUMP1
I:1/02	THERMOSTAT	0:2/02	PUMP2
I:1/03	LOWLEVELSENSOR	0:2/03	PUMP3
I:1/04	HI LEVELSENSOR	0:2/04	HEATER
T4:0/DN	TIMER	0:2/05	RUN
C5:0/DN	COUNTER	0:2/06	IDLE
		0:2/07	FULL

Software Programme

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

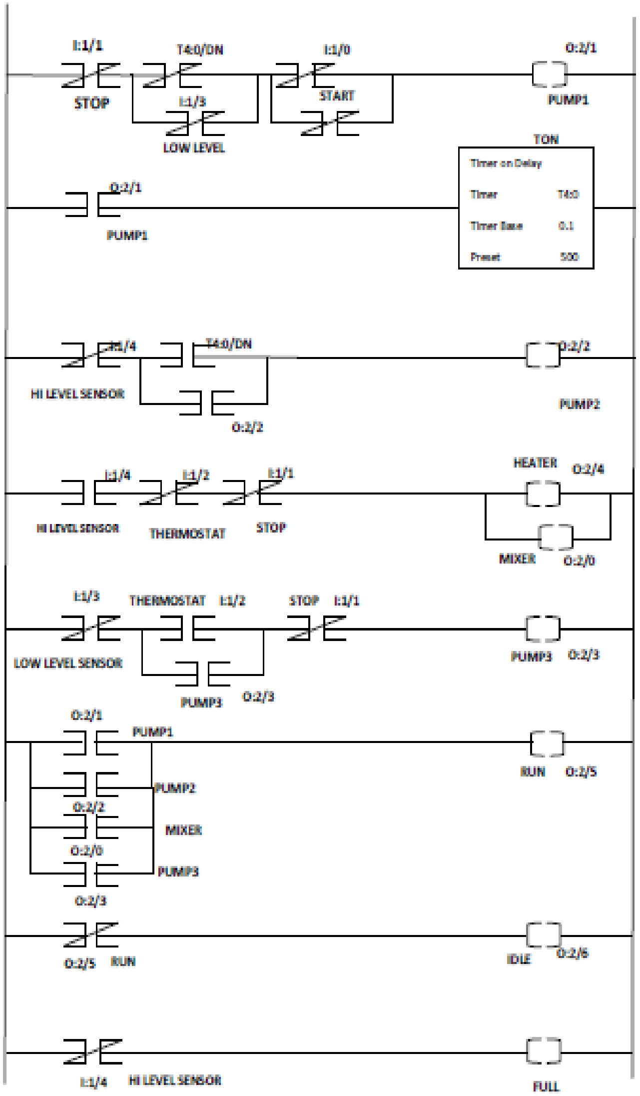


Fig3.2: Ladder Logic Programme of the System Operation

Software Operation

Figure3.2 shows the ladder logic application program for the control action. This code has 9 rungs which is executed within the CPU one after the other. It is seen from the ladder program that each device is assigned an address which is the memory address that holds the bit representing that particular device e.g. PUMP 1 is assigned the address O:2/01 which means that bit 1 of port 2 is an OUTPUT port which has been assigned to the device PUMP 1. This same mnemonic is applicable to all other memory addresses.

1. The first rung contains the instruction for opening PUMP1. It includes codes for the START & STOP instructions. It also includes code for the timer, [T4:0/DN], whose function will be explained in the second rung. There is an instruction here to use a counter to keep PUMP3 open when the mixer is suddenly stopped by pushing the STOP button on the control panel.
2. On the second rung, we see an instruction which uses the timer, [T4:0/DN], to stop the PUMP after a preset time. The time base is used to multiply the preset time to obtain the actual time. So for this particular application, the Timer is used to stop the PUMP1 after; (100 x 0.1)s i.e. 10s
3. The function of the instructions on the third rung is to START PUMP2 immediately the timer deactivates PUMP1. It is seen that it also includes the code for the Timer, [T4:0/DN]. The PUMP2 is stopped when the tank is full by a TANKFULL signal.
4. structions on the fourth rung are used to start the action of the electric motor that rotates within the MIXER. This action is initiated when TANKFULL causes PUMP2 to STOP. The HEATER also starts heating the vessel at the same time so that both actions are able to effect the mixing action.
5. The fifth rung contains the instruction that is used to stop the MIXER and at the same time open PUMP3. The rung a code LOWLEVEL that stops PUMP3 when all the liquid mixture has been discharged.
6. The sixth rung is essentially used to program the indicator RUN to show us when the machine is in operation i.e. when PUMP1, PUMP2, HEATER OR PUMP3 is functioning.
7. The seventh rung also contains another instruction that is used for the indicator. Here it is used TO Indicate the IDLE status of the system which occurs when the system is NOT running
8. The eight rung contains an instruction that is used to indicate FULL. This occurs when the full level sensor detects that the mixer is filled.

Testing

The testing of this system was carried out on three important parameters:-

- System Controls.
- System Indicators.
- Force Table testing.

System Controls Testing

This deals with the testing of START and STOP push buttons to determine if the mixer's normal operation meets the expected design requirements for system control. The START and STOP push buttons form the main controls of the system and the following results were obtained when they were tested.

Test Data	Expected Test Result	Actual Test Result
Push START Button	The system should start PUMP1, Timer should start operation, Timer should stop PUMP1 after 50s,Timer should start PUMP2,MIXER & HEATER should start when vessel is filled, MIXER & HEATER should stop when TEMP = 30°,PUMP3 should start,PUMP3 should stop when the vessel is empty, System should restart cycle by opening PUMP1	PUMP1 was started. Timer started operation.PUMP1 was stopped after 50s PUMP2 was started and also MIXER and HEATER started when the tank was filled. MIXER and HEATER stopped when the temperature = 30°.PUMP3 was startedPUMP3 was stopped when the vessel became empty System restarted operation
Push STOP Button	The system should stop its current operation. The system should open PUMP3 to remove any containing liquid. The system is expected to close PUMP3 when it is empty	All current operations were immediately Stopped.PUMP3 was opened and all remaining liquids were removed.PUMP3 was closed when the containing vessel became empty.

System Indicators Testing

This deals with the testing of RUN, IDLE and FULL indicators to determine if they function as expected and if their function meets the expected design requirements. The

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

indicators normally show a yellow light during an active state otherwise they are functioning. The table below shows the testing and the results:-

Test Data	Expected Test Result	Actual Test Result
RUN	RUN should be activated whenever PUMP1, PUMP2 PUMP3, MIXER or HEATER is running. Otherwise RUN should not be activated	RUN was activated by showing a yellow light when PUMP1, PUMP2, PUMP3, MIXER, HEATER was running. RUN was not activated at other times.
FULL	FULL should be activated when the containing vessel is FULL.	FULL was activated when the containing vessel is FULL.
IDLE	IDLE should be activated whenever the system is not running or when the system is stopped. It should also be activated when PUMP3 is discharging liquid from the vessel after STOP button has been pushed.	IDLE was activated when the STOP button was pushed IDLE was activated when PUMP3 was discharging liquid from the vessel after the STOP button has been pushed.

Table 4.2: System Indicators Test Results

The force table is a table that lists all the input and output ports of a programmable logic controller and their bit status within their memory locations. It is possible within the emulator to change the bit-state of a particular port thereby forcing a device that is connected to that particular port to operate or to stop operation. To carry this out on the mixer, we try to force a particular component of the mixer to operate irrespective of the current status of the program execution process. Three components were tested by this method; PUMP1, RUN and HEATER. The force table for each device was used to change the bit-state from 0 to 1 thereby forcing the device to operate. The results are shown in the table below:-

Test Data	Expected Test Result	Actual Test Result
Force PUMP1 By changing the bit-state to 1	PUMP1 should start operating irrespective of the state of the entire system.	PUMP1 started functioning even though the system was still mixing and heating
Force RUN by changing the bit to 1	RUN indicator light (yellow) should be activated irrespective of the state of the entire system	RUN indicator light (yellow) was activated even though the system was in IDLE mode
Force HEATER by changing the bit to 1	The HEATER should start heating irrespective of the state of the entire system. The MIXER should not start.	The HEATER started heating even though PUMP1 was still running without activating the MIXER

Table 4.3: Force Table Test Results

II. SUMMARY AND CONCLUSION

Summary of Achievement

This paper achieved a simulated control of an industrial batch mixer using a model in Logix 500. It also produced a Ladder logic program which can easily be used by any manufacturer who wishes to control a similar mixer by using a programmable logic controller.

Finally, the paper shows how quickly we can do the troubleshooting, testing and fault analysis of a component connected to a programmable logic controller by using its force table to change the bit-state of the port to which that component is connected to on the programmable logic controller.

5.5 Conclusions

In conclusion, this paper has shown how programmable logic controllers are a flexible and easily adaptable to typical industrial equipment such as the industrial mixer. These controllers are programmed by using a programming language called Ladder logic language. This language is easy to understand and is used by engineers and technicians thus can be changed to suit any particular need. The Industrial mixer is one of the several applications of the programmable logic controller to the control of an industrial process. This helps to eliminate huge costs

REFRENCES

1. Positite .M. "Omron Beginner's guide to PLC", www.mikrocontrol.co.yu, Serbia, 2003, pp 1-17
2. Barber Andy, Industrial Mixing, www.foodtechcompare.com, USA,
3. Bryan .L.A, "Programmable Logic Controllers Theory and implementation", second edition, Industrial Text Company, USA, 1997, pp 4-104
4. Wikipedia, "Industrial Mixer, www.wikipedia.com/industrial mixer, USA,
5. Dileep Krishnan, "Intelligent Traffic Light Control using PLC", Cochin University of Science and Technology, India, www.seminarpapers.com,
6. Gray Richard, "Ladder Logic Programming" Transtronics Inc, USA, www.xtronics.com
7. YASAR BIRBIR,SELCUK NOGAY, "PLC-based Monitoring and Control system for a 3-phase induction Motor", Marmara University, Technical Education Faculty, Turkey, www.csanyigroup.com
8. Thelearningpit, "About Logixpro 500", www.thelearningpit.com,USA

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication