INTRODUCTION TO LADDER LOGIC FOR SEQUENTIAL CONTROL SCHEME

SANJUKTA BEHERA

Dept:Applied Electronics and Instrumentation Engg. C V RAMAN College Of Engineering, Bhubaneswar, Orissa, India

Abstract: Today, programmable logic controllers (PLCs) is the dominant technology deployed in control automation systems in modern factories. The main modeling method of the PLC is based on ladder logic diagrams (LLDs).However, as a system gets more complex, LLD implementations poses a stumbling block in the design of more complex and real-time PLCs.the main objective of this paper is to run a motor in different sequence mode and simultaneously run the light with a predefined delay in industrial control.

Keywords: Programmable logic controller (PLC), ladder logic diagram (LLD), sequential control

I. INTRODUCTION

A Programmable Logic Controller (PLC), introduced in the late 1960's, is a reconfigurable device which was invented to replace mechanical relays or switch circuits in an industrial control automation system. By processing the relays electronically, a PLC saves space, power consumption, increase design reuse and development time. The most widely applied method to model a PLC is to use the Ladder Logic Diagram (LLD). LLD mimics the electrical drawing of the mechanical relay controller; hence, it represents the configuration of mechanical relay states in the control automation system. Today, the dominant technique of implementing a PLC is based on a microprocessor or microcontroller. The microprocessor is programmed with a desired control sequence of the target machine (an industrial or manufacturing automation), and the outputs of the PLC provide the corresponding control signals to synchronize and control the operation of the machine. However, as the industrial control system advances rapidly and becomes more complex, system performance has become an issue in many manufacturing processes [3]. As the number of inputs/outputs and the length of control sequence in an LLD modeling the PLC become sufficiently large, the execution time increases exponentially, system performance degrades, and additional processing power is needed [4]. Current methods of implementing LLDs prove inadequate in the design of more complex and real-time PLCs required in today's advanced automation systems. Even with modern microprocessor technology, PLC *online* solving (i.e. the generation of the actual responses of the control model to real-time input) restricts the number of rungs that can be solved in a permitted scan cycle [1]. This scan time constraint requires the application of order in the rung solving of the LLD model (*sequential* solving). In complex models, unexpected behaviors and diagnosing difficulties arise more easily with this sequential solving requirement [1]. Relay logic diagrams are very important because they show the symbols for different input and output devices. The first step in creating a program for a PLC is creating the relay logic diagram. After the relay logic diagram is drawn, it is then converted to a PLC ladder logic diagram**,** often called a *ladder diagram***.** The ladder diagram is used to program the programmable logic controller (PLC). The program defines the operations to be performed by the PLC.

II. BASICS OF LADDER LOGIC

The main modeling language of the PLC is based on the so-called Ladder Logic Diagram (LLD). This graphical symbolic language is widely used in the design of PLC for industrial automation. LLD models the actual combination of relay contacts. A relay contact or a step in LLD is either (a) normally closed (NC) or (b) normally open (NO). They are controlled by logical inputs and state variables which are represented by the labels (e.g. alarm, stop), as shown in Figure 1. When an input triggers the step, the corresponding relay state changes to the opposite state, that is, the NC step is turned ON while the NO step is turned OFF. (a) Normally (b) Normally (c) Output Closed step Opened step coil (NC) (NO) Figure 1: Steps in LLD Figure 2 depicts an exemplar LLD of a simple safety circuit in a machinery control. As shown in the figure, steps can be configured in a single line or in a combination of lines. The left vertical bar represent a 24V power supply while the right vertical bar represent 0V or ground. Current flows from the left rail of the ladder, i.e. a combination of relay contacts or steps, to the right rail, i.e. the output coil (represented by the symbol in diagram (c) in Figure 1). The output coil models the logical state or output variables of the ladder.

A combination of steps and an output will construct a complete ladder logic network or also called a rung. For instance, the first rung in Figure 2 consists of a single line of steps, which defines an AND logic[4]. The second rung consists of three lines of steps, which defines an OR logic. The third rung is made up of two lines, modeling a combination of AND-OR logic.

All steps (relay contacts) response simultaneously to their inputs and update the output states. In a microprocessor based implementation, the rungs cannot all be solved at the same time[6]. There are two main techniques used to solve the rungs: horizontal or vertical scans. In horizontal scan, the steps of a rung are scan from left to right (or from right to left) and the rung is solved one at a time. In vertical scan, all the steps in a column are solved from the top-most rung to the bottom-most rung or from the bottom-most rung to the top-most rung. Once the column steps are solved, the procedure moves to the next column of steps. Once the process has reached the end of rung or column, the scan process is repeated again from the first rung or column, indefinitely. This process is referred to as the cyclic scan. In the example in Figure 2, horizontal scan is applied. In practical PLCs,[7] besides the combinations of bit logic, timers and counters, and sometimes also sequencers are also used in the LLD models. In very advanced PLCs, other complex features are also included, such as serial communication, Ethernet, data storage, ADC, and DAC.

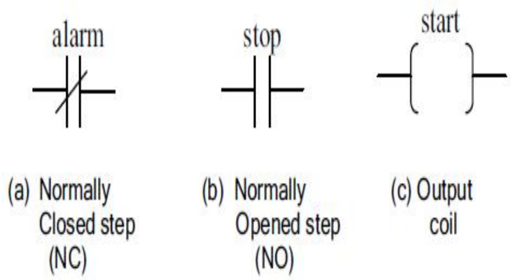

Figure 1: Steps in LLD

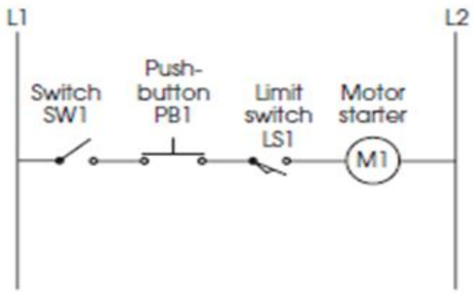

Figure 2 : A simple relay logic diagram

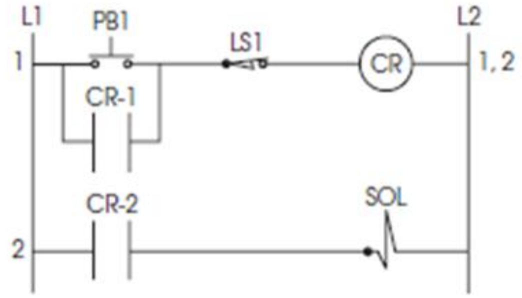

Figure 3: Use of a relay coil and contacts in a relay logic diagram.

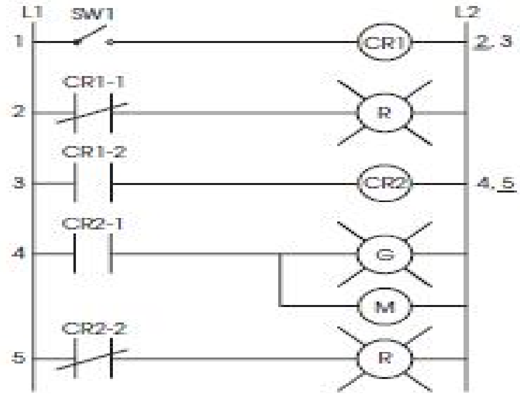

Figure 4: Relay coils and their associated contacts. The state of the contacts depends on the energized state of the coil.

III. CREATING RELAY LOGIC DIAGRAMS FOR SEQUENTIAL CONTROL

Two steps must be completed to draw a relay logic diagram of an industrial control system. In step one, clearly define the control problem.

This may involve drawing a schematic diagram of the system setup and discussing the problem with production and maintenance engineers and technicians. In step two, review and follow the nine relay logic diagram rules discussed in previous sections. This will allow you to draw the relay logic diagram. The following examples should be used as a guide to familiarize you with creating relay logic diagrams.

Step 1: Define the problem.

All the motor will operate in sequence when the input switch is made on. All the motor will on one after another with a 10msec delay.

Step 2: Draw the relay logic diagram

Figure 5 displays the relay logic diagram of the motor control system. The switch is a normally closed turns the motor M1. The normally open contact X1(NO) closes and energizes the delay for 10msec.The normally open contact X2 closes and turns the motor M2. The normally open contact X3 closes and energizes the delay for 10msec.The normally open contact X4 closes and turns the motor M3. This is the continuous run mode of operation for the motor. if we want to off the motor just to make the switch close. the same can be implement using NC with open switch.

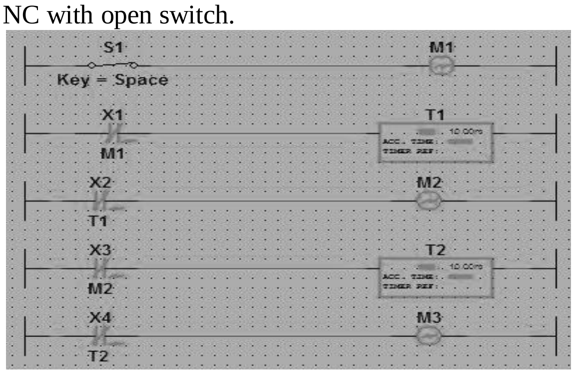

Figure 5: close the switch for running the motors continuously. One after another with a 10 msec delay

Step 1: Define the problem.
A motor can operate in continuous run mode or in temporary run mode also called *jog mode*. Therefore, in a motor control system, we need one stop pushbutton and two run pushbuttons. One pushbutton is for the continuous run mode of operation. The second one is for the temporary run mode of operation. The pushbutton for the continuous run mode of operation is labeled “start.” The pushbutton for the temporary run mode of operation is labeled “jog.” If we press the jog pushbutton, the motor will run as long as we are pressing the jog pushbutton, which is equivalent to closing a normally open switch. If we press the start pushbutton, the motor will start running continuously until we press the stop pushbutton. The red pilot light is on whenever the motor is operating.

Step 2: Draw the relay logic diagram.
Figure 6 displays the relay logic diagram of the motor control system. The stop pushbutton is a normally closed pushbutton. When the normally open start pushbutton is closed, the relay coil CR1 energizes. The normally open contact CR1 closes and seals or locks the relay coil. Therefore, the relay coil is energized as long as the normally closed stop pushbutton is closed. The normally open contact CR1 closes and turns the motor and red pilot light on. This is the continuous run mode of operation for the motor. With the motor off, pressing the normally open jog pushbutton turns on the motor and the red pilot light. Whenever the jog pushbutton
is released, the motor and the red pilot light turn off. Therefore, this represents the jog or temporary run mode of operation.

One motor and one pilot light are to be controlled in a control circuit. Use two start pushbuttons and two stop pushbuttons. Pressing any one of the start pushbuttons must turn on the motor. Pressing any one of the two stop pushbuttons must stop the motor operation. When the motor is running, the pilot light must be on.

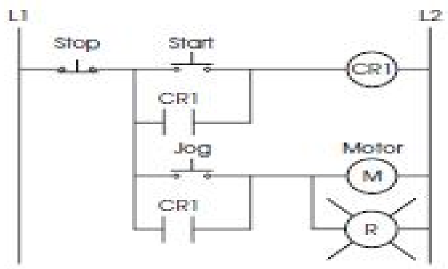

Figure 6: Press the start pushbutton for running the motor continuously. Press the jog pushbutton for running the motor temporarily

Follow the two steps required to draw the relay logic diagram of the following motor control problem.
Step 1: Define the problem. In a motor control system, one master stop pushbutton (Stop_PB) is available for stopping the operation at any time. If the temperature switch (TSW) is closed, pressing the red pushbutton (Red_PB) will turn on both motors one (M1) and two (M2). If TSW is open, pressing Red_PB will only turn on M2. If TSW is closed and the green pushbutton (Green_PB) is pressed once, M1 will run. Closing the temperature switch turns on the white pilot light.
Step 2: Draw the relay logic diagram.
Figure 7 displays the relay logic diagram of the circuit described in the first step. The normally closed stop pushbutton (Stop_PB) is the master stop pushbutton. If we press the normally open red pushbutton (Red_PB), motor two (M2) turns on. If the normally open temperature switch (TSW) is closed, then pressing the Red_PB will also turn on motor one (M1). Also, when TSW is closed, pressing the green pushbutton (Green_PB) will only turn on M1. When TSW is closed, the white pilot light (White_PLT) turns on.

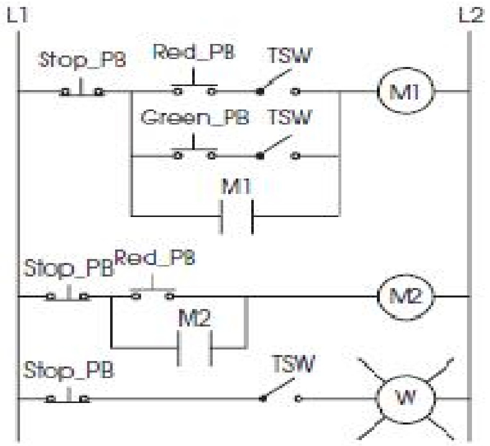

Figure 7:Close the temperature switch to turn on the white pilot light. Then, press the red pushbutton to turn on both motors. Press the green pushbutton to turn on only motor one (M1). Open the temperature switch and then press the red pushbutton to turn on only motor two (M2).

Follow the two steps required to draw the relay logic diagram of the following motor control problem.
Step 1: Define the problem. One motor and one pilot light are to be controlled in a control circuit. Use two start pushbuttons and two stop pushbuttons. Pressing any one of the start pushbuttons must turn on the motor. Pressing any one of the two stop pushbuttons must stop the motor operation. When the motor is running, the pilot light must be on.
Step 2: Draw the relay logic diagram.
Figure 8 displays the relay logic diagram of the circuit described in the first step. If we press either the green pushbutton (Green_PB) or the red pushbutton (Red_PB), the motor will operate. When the motor is running, the contact M1 closes and seals or locks the circuit. Pressing either stop pushbutton one (SPB1) or the stop pushbutton two (SPB2) breaks the current path and turns off the motor. When the motor is in operation, the red pilot light (Red_PLT) is on.

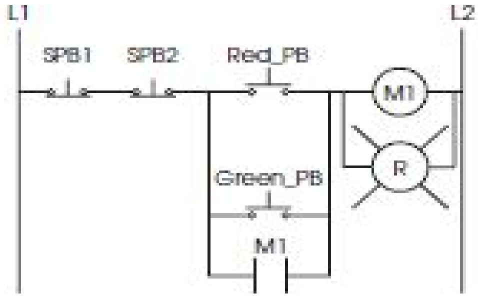

Figure 8: Press either the green pushbutton (PB4) or the red pushbutton (PB3) to turn on the motor. Press either one of the stop pushbuttons to turn off the motor.

CONCLUSION

Relay logic diagrams show the logical relationship between system components.

PLC is very versatile and effective tool in industrial automation. It cuts the production costs and increases the quality. Troubleshooting is easy in PLC. Wiring is complicated when electromechanical relays are used, where as wiring is easy when PLC's are used. Hence it is reliable to operate/ control a machine using PLC.

REFERENCES

[1] John T. Welch and Joan Carletta, "A Direct Mapping FPGA Architecture for Industrial Process Control Applications", Int. Conf. on Computer Design: VLSI in Computers & Processors, 17-20 Sept. 2000 Page(s):595 – 598.

[2] J.T. Welch, "An event chaining relay ladder logic solver", Computers in Industry, vol. 27, Jan. 1995, pp. 65-74.

[3] Wook Hyun Kwon, Jong-il Kim and Jaehyun Park,"Architecture of a Ladder Solving Processor LSP for PLC", IECON 1991.

[4] R. Filer and L. George Programmable Controllers and Designing Sequential Logic, 1992: Saunders College Pub.

[5] L.A. Bryan and E.A. Bryan Programmable Controllers: Theory and Implementation, 1988: Industrial Text. [3] C. D. Simpson Programmable Logic Controllers, 1994: Regents/Prentice-Hall. [6] W. G. Ian Programmable Controllers: Operation and Application, 1988: Prentice-Hall

[6] J.J. Harris, J.D. Broesch. R.M. Coon, "A Combined PLC and CPU Approach to Multiprocessor Control", Fusion Engineering, 1995. SOFE '95. 'Seeking a New Energy Era'., 16th IEEE/NPSS Symposium Volume 2, 30 Sept.-5 Oct. 1995 Page(s):874 – 877.

[7] Maria G. Ioannides, "Design and implementation of PLC-based monitoring control system for induction motor" IEEE Transactions on Energy Conversion, Vol. 19, No.3, pages 469-476, September 2004

★★★