INDUSTRIAL CONTROL AUTOMATION

• Programmable Logic Controllers
• Relay Circuits
• Ladder Logic

By: Patrick B. Roberts Sr.

Preface Employed at SYSCON - RKC as a Software Engineer, my responsibilities entail developing and maintaining applications that afford communications and user interfaces to the RKC product line of temperature control instrumentation.

One of the more advance instruments is the FAREX SR-Mini. These are modular systems from which customized systems can be constructed to satisfy a wide
range of applications. Individual modules, each with a specific set of functions, can be added to a base SR-Mini system.

The SR-Mini HG has the capability of Boolean logic, which is afforded to it by the DI-B module. The DI-B module has eight terminal inputs, thirty-two logic inputs, and eight logic outputs. Settings on the DI-B module are accessed via its eight channels. The DI-B modules are highly configurable; however, there currently does not exist a PC based, user-friendly interface to accomplish such a task.

In efforts to provide our customers with tools that will allow them to better utilize their RKC instrumentation, Syscon has decided to develop an interface for configuring the DI-B module. Furthermore, to provide an interface that the target audience is accustomed to, the interface will resemble ladder logic. I stress the word resemble since there are device constraints.

Being unfamiliar with ladder logic, research was required in order to familiarize myself with the project requirements. Having the Internet as a vast repository of information it was used as one of the principal resources. Having said that, all resources, both textual and electronic, will be properly cited in the bibliography. Furthermore, all Web sites from which concepts or images were gathered will be recognized with a URL to the site. The URLs for referenced material will be listed in the bibliography, and the URLs for images will be listed in the ALT tags of the image.

**INDUSTRIAL CONTROL AUTOMATION**

**Programmable Logic Controllers:**
Solid state devices that can control a process or machine and have the capability of being programmed or reprogrammed rapidly as the application may demand.
Relay Circuits:
An integral part of many control systems, the relay is an indirectly operated electrical switch. It is useful for implementing remote control of a system and for controlling high current devices with a low current control signal.

Ladder Logic:
A programming technique used to describe the logical interconnection of the electrical wiring of a certain control system. Ladder logic was developed because of its similarity to relay logic diagrams; thus, easing its acceptance into manufacturing.

Programmable Logic Controllers (PLCs) History:
In 1968 the earliest fully programmable logic controller (PLC) was developed by a consulting engineering firm named Bedford Associates, which has since changed its name to Modicon. The first PLC was designed as a dedicated computer control system that was built especially for the General Motors Hydromatic Division.

The first model was the 084 and went through a series of modifications in the early 1970s, which resulted in the 184 and 384 modules. During this period Modicon also produced several other models, the 284 and the 1084, which were subsequently followed by the 484. The 484 provided for the control of 256 inputs and 256 outputs from one processor.

Modicon was purchased by Gould Incorporated in 1977. In 1978, the Modbus data network was designed to enable other Modicon PLCs to transfer data to each other. In 1980, Modicon introduced the 84 Micro, a small, compact, low-cost, powerful PLC system. The system was comprised of 64 I/Os, counters, timers, sequencers, and mathematical functions.

Further developments resulted in the 584M (medium size), 584L (large size), 884, and 984 systems. The 884 and 984 systems were developed during the early 1980s. The design of these systems emphasized compatibility offering a wide range of modules. These modules included analog input, analog output,
analog multiplex, reed and solid state relay, TTL compatibility, and Proportional Integral Differential (PID) control.

**Processor:**
At the core of the PLC is the microprocessor. The chip, initially developed in 1970, was a 4-bit unit with a few diodes, resistors, capacitors, and transistors compressed into an integrated circuit (IC) chip. A few years later, the IC was followed by the medium scale integration (MSI). This allowed for the number of internal components to increase by about 100 per chip. The MSI was followed by the large scale integration (LSI). The LSI had anywhere from 5,000 to 10,000 transistors per chip. After LSI came very large scale integration (VLSI), which featured 600,000 devices per chip. Developments since then have increased the chips internal devices and power exponentially.

The microprocessor is a clock-driven time-sequential circuit that corresponds to the central processing unit (CPU) of a digital computer. The microprocessor in combination with random access memory (RAM) for data storage, programmable read-only memory (PROM) for storage of programming instructions, interfacing circuitry for communication to the outside world, and a power supply, form a microprocessor based system.

PLCs, like all digital computers ranging from super computers to the smallest microprocessor based systems, have three basic sections. They include:

- Central processing unit (CPU) or microprocessing unit (MPU)
- Memory: PROM, ROM, RAM and etc.
- Input/Output (I/O) section for communication with peripherals

In the figure below a typical processor system is shown in block diagram form.
Functionality:
The abstraction of a PLC is somewhat like a black box with a set amount of inputs from and outputs to the outside world. It has the capability to make decisions, store data, do timing cycles, perform simple arithmetic, convert codes, and so on. The principle difference between the black box abstraction and a hardwired logic system, is that specific code messages are stored in areas called program memory. These areas can be PROM or ROM and RAM chips.

As application demands change the benefits of a PLC, when contrasted to a hardwired logic system, become quite apparent. The hardwired logic system must be rewired to fit the new needs of the application. In contrast, the PLC simply needs to reprogrammed. The savings of time and expense can be significant. Furthermore, various recipes can be stored in memory and accessed when required, making the program extremely flexible.

PROM and ROM are used to store the coded messages or programs because they are nonvolatile. Thus, the programs stored in these chips remain intact when power is interrupted or turned off. RAM is used for storage and as a scratch area for temporary data while the program is running. This memory is volatile and data loss will occur when the power source is interrupted or turned off. In order to protect the data contained in RAM, PLCs are battery backed.

The system operates through interaction with the processor and program memory. When the system is initially powered up, the processor reads the first code word stored in memory and acts on this instruction. When completed, the nexted instruction is fetched and executed. This occurs until the task is completed. Hence, the name fetch-execute cycle. The processor communicates with the outside world with input and output modules.

When PLCs were first introduced they were met with reluctance. The use of these devices required the acquisition of new skills and methodologies. One of the stratgies to overcome this was the development of ladder logic. It was aimed at introducing a familiar technique for working with new technology.

Relay Circuits

A relay is a critical component of many control systems because they offer an indirectly operated electrical switch that can be used for remote control and to control high current devices with a low current control signal.
They operate on the same principles as a solenoid, except that the core is fixed in place. The relay has an input coil that uses a voltage/current to create a magnetic field. As the coil becomes magnetized it pulls a metal switch (or reed) towards it; thus, making an electrical contact. A contact that closes when the coil is energized is normally open. In contrast, a contact that the reed touches when the coil is not energized is normally closed.

In the figure to the right, the magnetic force pulls the armature toward the coil, and through the mechanical lever, forces the electrical contacts closed to close the controlled circuit. The springiness of the contacts cause them to open with the coil is de-energized. Relays can have normally open contacts (as the figure depicts), normally closed contacts, or various combinations of both.

Relays are used to let one power source close a switch for another. It is often the case that the other switch has a high voltage/current. The relay can accomplish this control while keeping the switches isolated.

Prior to the development of electronic programmable controllers, relay functions were performed by real hardware relays that were physically wired together. When a programmable controller is used, the only real hardware relays required are those for the actual outputs to the machine.

A solid state circuit used to store a sequence state or to combine multiple relay logic paths are referred to in design and programming as "control relays". In comparison to the hardwired relay method, control relays are solid-state memory locations. The digital code contents of these memory locations represent the state of a relay contact. Instead of using a meter to determine whether a real relay contact is set open or closed, the appropriate memory locations need only be examined.

Relay outputs are one of the most common outputs. Relays can be used with both AC and DC loads, as the figure to the left illustrates. A load is a term used to described whatever is connected to our outputs. It is called a load because the outputs are being loaded with something. If no load is connected to the outputs, and the output is connected directly to the power source, then damage to the outputs would most likely occur. Some common forms of loads are solenoid, lamp, motor, and etc. These loads come in a variety of electrical sizes.
When connecting a load to a PLC output, caution should be used so that the maximum current consumption of the load will not exceed that of the PLC output. An example of loads connected to a PLC is illustrated to the right. If the load draws too much current, beyond that specified for the PLC, damage could occur to the PLC output.

There are some loads that can be tricky, they are called "inductive loads". These loads have a tendency to deliver back current when turned on. This is similar to a voltage spike. In order to prevent any unexpected current flows diodes, varistors, and "snubber" circuits are used.

**Ladder Logic**

Ladder logic diagrams provide a standard representation of circuits that are convenient for design, analysis, and troubleshooting. They can be thought of as a means to describe outputs based on inputs. The vertical rails (the leftmost and rightmost vertical lines) indicate the power source, while the horizontal and vertical lines (the symbols placed on the rungs of the diagram) indicate the possible current flow. Buttons, contacts, limit switches, and coils each have a representative symbol that can be placed on the rungs of the ladder. Upon activation of appropriate contacts, the coil is energized and the appropriate contacts are in either a opened or closed state.

A simple implementation of ladder logic is illustrated in the two images below. Below and to the left is an electrical diagram of a switch and a motor, then below and to the right is the equivalent ladder logic diagram.
SYMBOLS & NOTATION:
Relay contacts or switches that are normally opened are depicted by two parallel vertical lines. When activated the switch or contact closes and power flows. Relay contacts or switches that are normally closed are depicted by two parallel vertical lines with a diagonal line running across them. When activated the switch or contact opens allowing power to flow. The figure to the right illustrates the appropriate symbols.

Outputs allow switches to close; thus, supplying or cutting off power to control devices. Ladder logic indicates what to do with an output, regardless of what is attached. Outputs can go to electrical outputs, or to memory.

Below are some output symbols and their respective meanings.

Considering the leftmost symbol above, when it is energized, the output turns on. In contrast, considering the middle symbol above, when it is energized, the output is turned off. The symbol above and to the right represents a one-shot relay. This is where an input will cause the output to go ON for one scan.
When L is energized, X (below, left) will be toggled ON and will stay on until U (below, middle) is energized. This is similar to a flip-flop and will stay set even when the PLC is turned off. Some PLCs allow immediate outputs and do not wait for a program scan to complete before setting an output. The rightmost symbol below represents this characteristic.

The notational identifiers for ladder logic diagrams are X, Y, and C, which is representative of inputs, outputs, and control relays, respectively.

**LOGIC OPERATIONS:**

- Switches and relay contacts can be wired into the various logic functions, such as OR, AND, and combinations of the two.

An example of the OR arrangement is illustrated in the figure to the left. In this scenario, each switch is located on separate rungs, each having a power path to the output. Therefore, if either switch, X1 or X2, is turned ON, then the appropriate contact closes and the output Y1 is energized.

The figure to the right illustrates the AND condition. In this case, the switches are lined up on the same rung; thus, there is only a single power path to the output. In order for the output Y1 to become energized, both the switches, X1 and X2, must be turned on.

As the diagram indicates, the switches are processed sequentially. For instance, if switch X1 is turned ON and switch X2 is turn OFF, power terminates at switch X2. Likewise, if switch X1 is turned OFF and switch X2 is turn ON, power terminates at switch X1 and switch X2 does not receive power.

The figure on the left illustrates a slightly more complicated combination of the logical operations and switches. This scenario combines AND and OR operations, as well as switches that are normally opened and normally closed.

To explain, when switch X4 is OFF, the motor is energized through a set of normally closed contacts (X4) on the switch. In the case that X4 is turned ON, its normally closed contacts open...
and the motor can only be energized by turning ON X1 or X3, which in turn closes the appropriate contact, X1 or X3.

If both switches X2 and X4 are turned ON, then their normally closed contacts will be open. In this state, only when switch X3 is turned ON and its contacts are closed will the motor become energized.

**TIMERS:**
In addition, timers can be introduced into a control system. The figure to the right illustrates how a timer would be included in a ladder logic diagram. Since the discussion of ladder logic is relative to PLCs, a brief description of PLC timers is appropriate.

Programmable controllers can effectively duplicate electromechanical sequencing drum timers, time-out timers, and counters. They are built into the controller and only need be programmed to be utilized. No additional hardware is required; thus, the electronic equivalent of the aforementioned hardware can often simplify and improve the performance of a control system. This can be accomplished without increasing cost.

**CONTROL RELAY SCHEME:**
Control relays not only provide decision-making through logic circuit connections, they also allow for the control of power at remote locations and/or allow for the control of a high-voltage or a high-current device with a low current switch.
A schematic representation of a control relay is in the figure to the left. As the figure shows, X1 is open, C1 is de-energized, and Y1 is connected to the 120V source through the normally closed contacts of C1.

In contrast, when X1 is closed, the coil C1 is energized and the movable contact moves to close the normally open contacts. This has the effect of removing the power from Y1 and applying power to Y2. The ladder logic equivalent is illustrated in the figure above and to the right. Note that all parts relative to the same relay are designated by the same number.

**PLC SCHEME:**
A variety of ladder logic schemes exist. For instance, when implementing ladder logic for PLCs the right hand rail is not shown, for the primary elements of programming are only relay contacts and coils. There are no symbolic distinctions made between limit switches, push buttons, and relay contacts. This particular diagramming scheme allows for up to ten elements in each horizontal rung, and up to seven rungs per network. Multiple networks may be utilized to complete a control circuit. Power flow within a network can only flow from left to right, or vertically; it may never flow towards the left.

The PLC implementation of ladder logic is depicted in the image below.
CONCLUSION:
Originally ladder logic was developed to mimic relay logic in efforts to ease its acceptance into the manufacturing arena. The initial use of programmable controllers met with some resistance due to the need to acquire new skills and approaches. With the use of ladder logic the paradigm shift was reduced considerably, therefore making programmable controllers (PLCs) more acceptable.

Original relay ladder logic diagrams showed how to hook-up inputs to run outputs. Since then more intricate circuits and control systems have been (and can be) constructed. Following is a brief listing of some of the implementations of ladder logic, minus the diagrams. The list is far from inclusive.

**Output Ladder Logic:**
The logic required to produce the outputs to operate a given machine.

**Drum Timer/Event Ladder Logic:**
A logic circuit describes a sequence conditions that must be met sequentially. For example the condition in step 1 must be satisfied before moving on to step 2, and so on.

**Error Detection Ladder Logic:**
As implied the error detection logic determines if the machine is malfunctioning. If so, all of the outputs are disabled to prevent equipment damage.

**Exclusive-OR Circuit:**
Used when it is necessary to prevent an output from energizing if two conditions, which can activate an output independently, occur simultaneously.

**Start/Stop Circuit:**
Used to start or stop a motor or a process, or to simply enable or disable some function.

**One-Shot Signal:**
One shot (transitional output) is a program generated pulse output that, when triggered, goes HIGH for the duration of one program scan and goes LOW. The one-shot is typically used as a clearing or resetting signal.

**Oscillator Circuit:**
A simple circuit used to generate a periodic output pulse of any duration.

Since ladder logic was developed to provide a familiar tool for those acquainted with hardwired, relay control systems, it has since become a main stay in industrial control automation.

**Bibliography**

**Texts:**

Industrial Control Automation

World Wide Web & Internet Sites:

AuroTek

Advantage Industrial Automation
http://www.advanceind.com

Allen-Bradley
http://www.ab.com

Allied Automation & Engineering
http://www.softwarecorp.com

Control Technology Corporation
http://www.control.com

Graff Automation
http://www.graff-automation.com

Honeywell, Industrial Automation Control
http://www.iac.honeywell.com

IDEAS, Inc.
http://www.ideas-automation.com

Quick Controls Ltd.
http://www.quick-controls.com

Schneider Automation Inc.
http://www.modicon.com

SoftPLC Corporation
http://www.softplc.com

Telenetics
http://www.telenetics.com

INDUSTRIAL CONTROL AUTOMATION

In the preface I mentioned the project that motivated the need for this report.
Therefore, feel that it is appropriate to show (briefly) how the application has
developed. Furthermore, since the report is centered around ladder logic, only a
few screen shots relative to the report will be illustrated. The project is nearing
the first alpha release.

The image above is one of the user interfaces implemented for configuring the
logic configurations of the DI-B module. It resembles Boolean logic and was
closely modeled from the literature contained in the manual for the device.

The image to the left is a software illustration of the DI-B Logic
module. The image is an active image. In otherwords, there
are two menu systems asscociated with the image, one is
accessed by pressing the left mouse button, and the other is
accessed by pressing the right mouse button. Pressing the left
mouse button while the mouse pointer is located over one of
the greenish gray rectangles will present the user with a choice
of the "Boolean Logic" interface (above) or the "Ladder Logic"
interface (below).

The rectangles are representative of the number of channels
(configurable logic sections) native to the device. A module
structure is created when an instance of a module is created.
This structure contains all module, channel, and cell data
relative to the given module.

Since the messaging system of the device is structured after
the Boolean logic, as depicted in the image above, the data
captured in the cells of the ladder diagram must be mapped to
Boolean logic accordingly.
As can be seen from the software illustration of the ladder diagram, it is not full blown ladder logic. This is due to device restrictions. Nonetheless, the ladder logic interface for configuring the DI-B module provides access to all configurable functions and settings.