PLC and its Applications

Sadegh vosough[1] and Amir vosough[2]
Department of Electronics, Ahvaz Branch, Islamic Azad University, Ahvaz, Iran
Department of Mechanics, Mahshahr Branch, Islamic Azad University, Mahshahr, Iran
vosough.sadegh@yahoo.com, vosoogh_amir@yahoo.com

Abstract— **In this paper about the important concept of PLC and its application has been discussed. Early PLCs were designed to replace relay logic systems. These PLCs were programmed in "ladder logic", which strongly resembles a schematic diagram of relay logic. The main difference from other computers is that PLCs are armored for severe conditions (such as dust, moisture, heat, cold etc) and have the facility for extensive input/output (I/O) arrangements. PLC applications are typically highly customized systems so the cost of a packaged PLC is low compared to the cost of a specific custom-built controller design. On the other hand, in case of mass-produced goods, customized control systems are economic due to the lower cost of the components, which can be optimally chosen instead of a "generic" solution, and where the non-recurring engineering charges are spread over thousands or millions of units.**

Keywords– PLC-Feathers, Application of PLC and Ladder Logic

I. INTRODUCTION

Control engineering has evolved over time. In the past humans were the main methods for controlling a system. More recently electricity has been used for control and early electrical control based on relays. These relays allow power to be switched on and off without a mechanical switch. It is common to use relays to make simple logical control decisions. The development of low cost computer has brought the most recent revolution, the Programmable Logic Controller (PLC) [1]. The advent of the PLC began in the 1970s, and has become the most common choice for manufacturing controls. PLCs have been gaining popularity on the factory floor and will probably remain predominant for some time to come. Most of this is because of the advantages they offer. Cost effective for controlling complex systems.

- Flexible and can be reapplied to control other systems quickly and easily.
- Computational abilities allow more sophisticated control.
- Trouble shooting aids make programming easier and reduce downtime.

Reliable components make these likely to operate for years before failure. The PLC was invented in response to the needs of the American automotive manufacturing industry. Programmable logic controllers were initially adopted by the automotive industry where software revision replaced the re-wiring of hard-wired control panels when production models changed. Before the PLC, control, sequencing, and safety interlock logic for manufacturing automobiles was accomplished using hundreds or thousands of relays, cam timers, and drum sequencers and dedicated closed-loop controllers. The process for updating such facilities for the yearly model change-over was very time consuming and expensive, as electricians needed to individually rewire each and every relay. Digital computers, being general-purpose programmable devices, were soon applied to control of industrial processes. Early computers required specialist programmers, and stringent operating environmental control for temperature, cleanliness, and power quality. Using a general-purpose computer for process control required protecting the computer from the plant floor conditions. An industrial control computer would have several attributes: it would tolerate the shop-floor environment, it would support discrete (bit-form) input and output in an easily extensible manner, it would not require years of training to use, and it would permit its operation to be monitored. The response time of any computer system must be fast enough to be useful for control; the required speed varying according to the nature of the process [1]. In 1968 GM Hydramatic (the automatic transmission division of General Motors) issued a request for proposal for an electronic replacement for hard-wired relay systems. The winning proposal came from Bedford Associates of Bedford, Massachusetts. The first PLC, designated the 084 because it was Bedford Associates' eighty-fourth project, was the result [2]. Bedford Associates started a new company dedicated to developing, manufacturing, selling, and servicing this new product: Modicum, which stood for Modular Digital Controller. One of the people who worked on that project was Dick Morley, who is considered to be the "father" of the PLC [3]. The Modicon brand was sold in 1977 to Gould Electronics, and later acquired by German Company AEG and then by French Schneider Electric, the current owner. One of the very first 084 models built is now on display at Modicon's headquarters in North Andover, Massachusetts. It was presented to Modicon by GM, when the unit was retired after nearly twenty years of uninterrupted service. Modicon used the 84 moniker at the end of its product range until the 984 made its appearance. The automotive industry is still one of the largest users of PLCs. Further information about PLC can be found at [6], [10].

II. PLCs DEVELOPMENT

Early PLCs were designed to replace relay logic systems. These PLCs were programmed in "ladder logic", which strongly resembles a schematic diagram of relay logic. This program notation was chosen to reduce training demands for the existing technicians. Other early PLCs used a form of instruction list programming, based on a stack-based logic solver. Modern PLCs can be programmed in a variety of ways, from ladder logic to more traditional programming languages such as BASIC and C. Another method is State Logic, a very high-level programming language designed to program PLCs based on state transition diagrams. Many early PLCs did not have accompanying programming terminals that were capable of graphical representation of the logic, and so the logic was instead represented as a series of logic expressions in some version of Boolean format, similar to Boolean algebra.

As programming terminals evolved, it became more common for ladder logic to be used, for the aforementioned reasons and because it was a familiar format used for electromechanical control panels. Newer formats such as State Logic and Function Block (which is similar to the way logic is depicted when using digital integrated logic circuits) exist, but they are still not as popular as ladder logic. A primary reason for this is that PLCs solve the logic in a predictable and repeating sequence, and ladder logic allows the programmer (the person writing the logic) to see any issues with the timing of the logic sequence more easily than would be possible in other formats.

III. PROGRAMMING

Early PLCs, up to the mid-1980s, were programmed using proprietary programming panels or special-purpose programming terminals, which often had dedicated function keys representing the various logical elements of PLC programs [2]. Programs were stored on cassette tape cartridges. Facilities for printing and documentation were very minimal due to lack of memory capacity. The very oldest PLCs used non-volatile magnetic core memory.

IV. FUNCTIONALITY

The functionality of the PLC has evolved over the years to include sequential relay control, motion control, process control, distributed control systems and networking. The data handling, storage, processing power and communication capabilities of some modern PLCs are approximately equivalent to desktop computers. PLC-like programming combined with remote I/O hardware, allow a general-purpose desktop computer to overlap some PLCs in certain applications. Regarding the practicality of these desktop computer based logic controllers, it is important to note that they have not been generally accepted in heavy industry because the desktop computers run on less stable operating systems than do PLCs, and because the desktop computer hardware is typically not designed to the same levels of tolerance to temperature, humidity, vibration, and longevity as the processors used in PLCs.

In addition to the hardware limitations of desktop based logic, operating systems such as Windows do not lend themselves to deterministic logic execution, with the result that the logic may not always respond to changes in logic state or input status with the extreme consistency in timing as is expected from PLCs. Still, such desktop logic applications find use in less critical situations, such as laboratory automation and use in small facilities where the application is less demanding and critical, because they are generally much less expensive than PLCs. In more recent years, small products called PLRs (programmable logic relays), and also by similar names, have become more common and accepted. These are very much like PLCs, and are used in light industry where only a few points of I/O (i.e. a few signals coming in from the real world and a few going out) are involved, and low cost is desired. These small devices are typically made in a common physical size and shape by several manufacturers, and branded by the makers of larger PLCs to fill out their low end product range. Popular names include PICO Controller, NANO PLC, and other names implying very small controllers.

Most of these have between 8 and 12 digital inputs, 4 and 8 digital outputs, and up to 2 analog inputs. Size is usually about 4" wide, 3" high, and 3" deep. Most such devices include a tiny postage stamp sized LCD screen for viewing simplified ladder logic (only a very small portion of the program being visible at a given time) and status of I/O points, and typically these screens are accompanied by a 4-way rocker push-button plus four more separate push-buttons, similar to the key buttons on a VCR remote control, and used to navigate and edit the logic. Most have a small plug for connecting via RS-232 or RS-485 to a personal computer so that programmers can use simple Windows applications for programming instead of being forced to use the tiny LCD and push-button set for this purpose. Unlike regular PLCs that are usually modular and greatly expandable, the PLRs are usually not modular or expandable, but their price can be two orders of magnitude less than a PLC and they still offer robust design and deterministic execution of the logic. More recently, PLCs are programmed using application software on personal computers. The computer is connected to the PLC through Ethernet, RS-232, RS-485 or RS-422 cabling.

The programming software allows entry and editing of the ladder-style logic. Generally the software provides functions for debugging and troubleshooting the PLC software, for example, by highlighting portions of the logic to show current status during operation or via simulation. The software will upload and download the PLC program, for backup and restoration purposes. In some models of programmable controller, the program is transferred from a personal computer to the PLC though a programming board which writes the program into a removable chip such as an EEPROM or EPROM.

V. FEATURES

The main difference from other computers is that PLCs are armored for severe conditions (such as dust, moisture, heat, cold) and have the facility for extensive input/output (I/O) arrangements. These connect the PLC to sensors and actuators. PLCs read limit switches, analog process variables (such as temperature and pressure), and the positions of complex positioning systems. Some use machine vision. On the actuator side, PLCs operate electric motors, pneumatic or hydraulic cylinders, magnetic relays, solenoids, or analog outputs. The input/output arrangements may be built into a simple PLC, or the PLC may have external I/O modules attached to a computer network that plugs into the PLC.

VI. SCAN TIME

A PLC program is generally executed repeatedly as long as the controlled system is running. The status of physical input points is copied to an area of memory accessible to the processor, sometimes called the "I/O Image Table". The program is then run from its first instruction run down to the last rung. It takes some time for the processor of the PLC to evaluate all the ladders and update the I/O image table with the status of outputs. This scan time may be a few milliseconds for a small program or on a fast processor, but older PLCs running very large programs could take much longer (say, up to 100 ms) to execute the program. If the scan time was too long, the response of the PLC to process conditions would be too slow to be useful.

As PLCs became more advanced, methods were developed to change the sequence of ladder execution, and subroutines were implemented. This simplified programming and could also be used to save scan time for high-speed processes; parts of the program used, for example, only for setting up the machine could be segregated from those parts required to operate at higher speed. Special-purpose I/O modules[10], such as timer modules or counter modules, could be used where the scan time of the processor was too long to reliably pick up, for example, counting pulses from a shaft encoder. The relatively slow PLC could still interpret the counted values to control a machine, but the accumulation of pulses was done by a dedicated module that was unaffected by the speed of the program execution.

VII. SYSTEM SCALE

A small PLC will have a fixed number of connections built in for inputs and outputs. Typically, expansions are available if the base model has insufficient I/O. Modular PLCs have a chassis (also called a rack) into which are placed modules with different functions. The processor and selection of I/O modules is customized for the particular application. Several racks can be administered by a single processor, and may have thousands of inputs and outputs. A special high speed serial I/O link is used so that racks can be distributed away from the processor, reducing the wiring costs for large plants.

A. PLC Compared with other Control System

Allen-Bradley PLC installed in a control panel. PLCs are well-adapted to a range of automation tasks. These are typically industrial processes in manufacturing where the cost of developing and maintaining the automation system is high relative to the total cost of the automation, and where changes to the system would be expected during its operational life. PLCs contain input and output devices compatible with industrial pilot devices and controls; little electrical design is required, and the design problem centers on expressing the desired sequence of operations. PLC applications are typically highly customized systems so the cost of a packaged PLC is low compared to the cost of a specific custom-built controller design. On the other hand, in the case of mass-produced goods, customized control systems are economic due to the lower cost of the components, which can be optimally chosen instead of a "generic" solution, and where the non-recurring engineering charges are spread over thousands or millions of units. For high volume or very simple fixed automation tasks, different techniques are used. For example, a consumer dishwasher would be controlled by an electromechanical cam timer costing only a few dollars in production quantities. A microcontroller-based design would be appropriate where hundreds or thousands of units will be produced and so the development cost (design of power supplies, input/output hardware and necessary testing and certification) can be spread over many sales, and where the end-user would not need to alter the control. Automotive applications are an example; millions of units are built each year, and very few end-users alter the programming of these controllers. However, some specialty vehicles such as transit busses economically use PLCs instead of custom-designed controls, because the volumes are low and the development cost would be uneconomic [5].

Very complex process control, such as used in the chemical industry, may require algorithms and performance beyond the capability of even high-performance PLCs. Very high-speed or precision controls may also require customized solutions; for example, aircraft flight controls. Single-board computers using semi-customized or fully proprietary hardware may be chosen for very demanding control applications where the high development and maintenance cost can be supported. "Soft PLCs" running on desktop-type computers can interface with industrial I/O hardware while executing programs within a version of commercial operating systems adapted for process control needs [5].

Programmable controllers are widely used in motion control, positioning control and torque control. Some manufacturers produce motion control units to be integrated with PLC so that G-code (involving a CNC machine) can be used to instruct machine movements. PLCs may include logic for single-variable feedback analog control loop, a "proportional, integral, derivative" or "PID controller". A PID loop could be used to control the temperature of a manufacturing process, for example. Historically PLCs were usually configured with only a few analog control loops; where processes required hundreds or thousands of loops, a distributed control system (DCS) would instead be used. As PLCs have become more powerful, the boundary between DCS and PLC applications has become less distinct. PLCs have similar functionality as Remote Terminal Units. An RTU, however, usually does not support control algorithms or control loops. As hardware rapidly becomes more powerful and cheaper, RTUs, PLCs and DCSs are increasingly beginning to overlap in responsibilities, and many vendors sell RTUs with PLC-like features and vice versa.

The industry has standardized on the IEC 61131-3 functional block language for creating programs to run on RTUs and PLCs, although nearly all vendors also offer proprietary alternatives and associated development environments. In recent years "Safety" PLCs have started to become popular, either as standalone models (Pilz PNOZ Multi, Sick etc) or as functionality and safety-rated hardware added to existing controller architectures (Allen Bradley Guardlogix, Siemens F-series etc). These differ from conventional PLC types as being suitable for use in safety-critical applications for which PLCs have traditionally been supplemented with hard-wired safety relays. For example, a Safety PLC might be used to control access to a robot cell with trapped-key access, or perhaps to manage the shutdown response to an emergency stop on a conveyor production line. Such PLCs typically have a restricted regular instruction set augmented with safety-specific instructions designed to interface to emergency stops, light screens and so forth. The flexibility that such systems offer has resulted in rapid growth of demand for these controllers.

VIII. DIGITAL AND ANALOG SIGNALS

Digital or discrete signals behave as binary switches, yielding simply an On or Off signal (1 or 0, True or False, respectively). Push buttons, limit switches, and photoelectric sensors are examples of devices providing a discrete signal. Discrete signals are sent using either voltage or current, where a specific range is designated as *On* and another as *Off*. For example, a PLC might use 24 V DC I/O, with values above 22 V DC representing On, values below 2VDC representing *Off*, and intermediate values undefined. Initially, PLCs had only discrete I/O. Analog signals are like volume controls, with a range of values between zero and full-scale. These are typically interpreted as integer values (counts) by the PLC, with various ranges of accuracy depending on the device and the number of bits available to store the data. As PLCs typically use 16-bit signed binary processors, the integer values are limited between -32,768 and +32,767. Pressure, temperature, flow, and weight are often represented by analog signals. Analog signals can use voltage or current with a magnitude proportional to the value of the process signal. For example, an analog 0 - 10 V input or 4-20 mA would be converted into an integer value of 0 - 32767. Current inputs are less sensitive to electrical noise (i.e. from welders or electric motor starts) than voltage inputs.

IX. AREA OF APPLICATION OF PLC

Every system or machine has a controller. Depending on the type of technology used, controllers can be divided into pneumatic, hydraulic, electrical and electronic controllers. Frequently, a combination of different technologies is used. Furthermore, differentiation is made between hard-wired programmable (e.g. wiring of electro-mechanical or electronic components) and programmable logic controllers. The first is used primarily in cases, where any reprogramming by the user is out of the question and the job size warrants the development of a special controller. Typical applications for such controllers can be found in automatic washing machines, video cameras, and cars. However, if the job size does not warrant the development of a special controller or if the user is to have the facility of making simple or independent program changes, or of setting timers and counters, then the use of a universal controller, where the program is written to an electronic memory, is the preferred option. The PLC represents such a universal controller. It can be used for different applications and, via the program installed in its memory, provides the user with a simple means of changing, extending and optimizing control processes. The original task of a PLC involved the interconnection of input signals according to a specified program and, if "true", to switch the corresponding output. Boolean algebra forms the mathematical basis for this operation, which recognizes precisely two defined statuses of one variable: "0"and "1.

Accordingly, an output can only assume these two statuses. For instance, a connected motor could therefore be either switched on or off, i.e., controlled. This function has coined the name PLC: Programmable logic controller, i.e. the input/output behavior is similar to that of an electromagnetically or pneumatic switching valve controller; the program is stored in an electronic memory. However, the tasks of a PLC have rapidly multiplied: Timer and counter functions, memory setting and resetting, mathematical computing operations all represent functions, which can be executed by practically any of today's PLCs. The new PLC standard EN 61131 (IEC 61131) [11] previously valid PLC standards focusing mainly on PLC programming were generally geared to current state of the art technology in Europe at the end of the seventies.

This took into account non-networked PLC systems, which primarily execute logic operations on binary signals. Previously, no equivalent, standardized language elements existed for the PLC developments and system expansions made in the eighties, such as processing of analogue signals, interconnection of intelligent modules, networked PLC systems etc. Consequently, PLC systems by different manufacturers required entirely different programming. Since 1992, an international standard now exists for programmable logic controllers and associated peripheral devices (programming and diagnostic tools, testing equipment, man-to-machine interfaces etc.). In this context, a device configured by the user and consisting of the above components is known as a PLC system.

The new EN 61131 (IEC 61131) standard consists of five parts:

_ Part 1: General information
_ Part 2: Equipment requirements and tests
_ Part 3: Programming languages
_ Part 4: User guidelines (in preparation with IEC)
_ Part 5: Messaging service specification (in preparation with IEC)

X. LADDER LOGIC

Ladder logic is the main programming method used for PLCs. As mentioned before, ladder logic has been developed to mimic relay logic. The decision to use the relay logic diagrams was a strategic one. By selecting ladder logic as the main programming method, the amount of retraining needed for engineers and trades people was greatly reduced. Modern control systems still include relays, but these are rarely used for logic. A relay is a simple device that uses a magnetic field to control a switch, as pictured in Fig.1. When a voltage is applied to the input coil, the resulting current creates a magnetic field. The magnetic field pulls a metal switch (or reed) towards it and the contacts touch, closing the switch. The contact that closes when the coil is energized is called normally open. The normally closed contacts touch when the

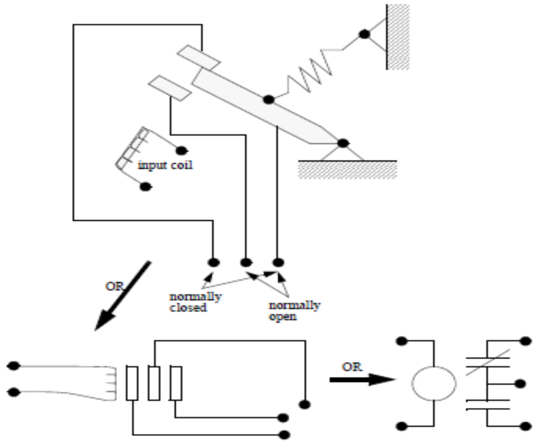

Fig. 1: Simple Relay Layouts and Schematics

input coil is not energized. Relays are normally drawn in schematic form using a circle to represent the input coil. The output contacts are shown with two parallel lines. Normally, open contacts are shown as two lines, and will be open (non-conducting) when the input is not energized. Normally closed contacts are shown with two lines with a diagonal line through them. When the input coil is not energized the normally closed contacts will be closed (conducting).

Relays are used to let one power source close a switch for another (often high current) power source, while keeping them isolated. An example of a relay in a simple control application is shown in Figure 3. In this system the first relay on the left is used as normally closed, and will allow current to flow until a voltage is applied to the input A. The second relay is normally open and will not allow current to flow until a voltage is applied to the input B. If current is flowing through the first two relays normally open normally closed input coil OR then current will flow through the coil in the third relay, and close the switch for output C. This circuit would normally be drawn in the ladder logic form. This can be read logically as C will be on if A is off and B is on.Figure2A Simple Relay Controllers The example in Fig. 2 does not show the entire control system, but only the logic.

The example in Figure 2 does not show the entire control system, but only the logic. When we consider a PLC there are inputs, outputs, and the logic. Figure 3 shows a more complete representation of the PLC. Here there are two inputs from push buttons. We can imagine the inputs as activating 24V DC relay coils in the PLC. This in turn drives an output relay that switches 115V AC that will turn on a light. Note, in actual PLCs inputs are never relays, but outputs are often relays. The ladder logic in the PLC is actually a computer program that the user can enter and change. Notice that both of the input push buttons are normally open, but the ladder logic inside the PLC has one normally open contact, and one normally closed contact. Do not think that the ladder logic in the PLC needs to match the inputs or outputs. Many beginners will get caught trying to make the ladder logic match the input types.

Many relays also have multiple outputs (throws) and this allows an output relay to also be an input simultaneously. The circuit shown in Figure 4 is an example of this; it is called a seal in circuit. In this circuit the current can flow through either branch of the circuit, through the contacts labeled A or

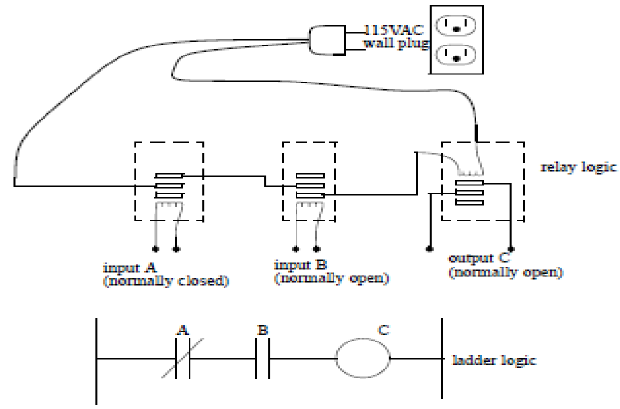

Fig. 2: A Simple Relay Controller

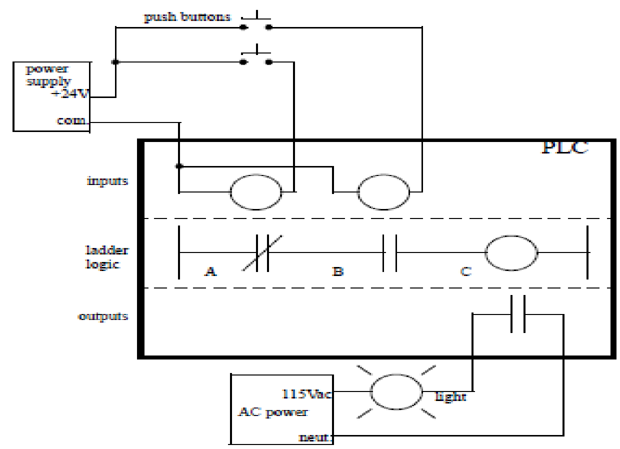

Fig. 3: A PLC Illustrated with Relays

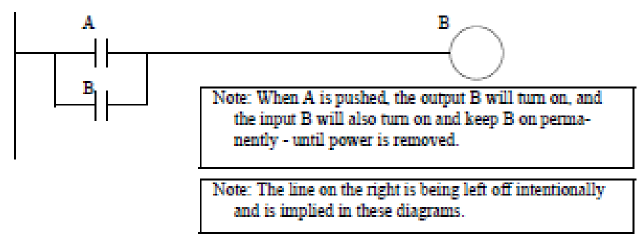

Fig. 4: A Seal-in Circuit

B. The input B will only be on when the output B is on. If B is off, and A is energized, then B will turn on. If B turns on then the input B will turn on and keep output B on even if input A goes off. After B is turned on the output B will not turn off.

XI. CONCULATION

In this paper about the concept of PLC and its application has discussed. The PLC was invented in response to the needs of the American automotive manufacturing industry. Programmable logic controllers were initially adopted by the automotive industry where software revision replaced the re-wiring of hard-wired control panels when production models changed.PLC applications are typically highly customized systems so the cost of a packaged PLC is low compared to the cost of a specific custom-built controller design. On the other hand, in the case of mass-produced goods, customized control systems are economic due to the lower cost of the components, which can be optimally chosen instead of a "generic" solution, and where the non-recurring engineering charges are spread over thousands or millions of units. Ladder logic is the main programming method used for PLCs. As mentioned before, ladder logic has been developed to mimic relay logic. The decision to use the relay logic diagrams was a strategic one. By selecting ladder logic as the main programming method, the amount of retraining needed for engineers and trades people was greatly reduced. Modern control systems still include relays, but these are rarely used for logic.

REFERENCES

[1] E. A. Parr, Industrial Control Handbook, Industrial Press Inc., 1999.

[2] M. A. Laughton, D. J. Warne, Electrical Engineer's Reference book, 16th Edition, Newnes, 2003.

[3] D. Morley, looks back on the 40th anniversary of the PLC Manufacturing Automation, 12 September, 2008.

[4] W. Bolton, Programmable Logic Controllers, Fifth Edition, Newnes, 2009.

[5] K. Gregory, Mc. Millan, M. Douglas. Considine (ed), Process/Industrial Instruments and Controls Handbook Fifth Edition, McGraw-Hill, 1999

[6] A. Garfield, Dragging Lazards into the 21st century. The Independent (London), 1999.

[7] E. N. Bakker, C. J. Kerkhof., and P. Sipkema, Signal transduction in spontaneous myogenic tone in isolated arterioles from rat skeletal muscle, Cardiovasc. Res. 41: 229-236, 1999.

[8] D. Kandray, *Programmable Automation Technologies*, Industrial Press, 2010.

[9] Prudential PLC– Contact Prudential Corporation Asia. Retrieved 18 April, 2011.

[10] Prudential PLC– Contact Jackson National Life Insurance Company. Retrieved 18 April, 2011.