# NEAR EAST UNIVERSITY

## Faculty of Engineering

## Department of Electrical and Electronic Engineering

# CAR PARK AREA CONTROL SYSTEM WITH PLC

**Graduation Project**
**EE – 400**

**Student:** Gülşah Tanrıkulu (20000724)

**Supervisor:** Özgür Cemal Özerdem

**Lefkoşa - 2005**

## ACKNOWLEDGEMENTS

*It is my pleasure to take this opportunity to express my greatest gratitude to many individuals who have given me a lot of supports during my four-year Undergraduate program in the **Near East University**. Without them, my Graduation Project would not have been successfully completed on time.*

*First of all, I would like to express my thanks to my supervisor **Mr. Özgür Cemal ÖZERDEM** for supervising my project. Under the guidance of him I successfully overcome many difficulties and I learned a lot about electronics. In each discussion, he used to explain the problems and answer my questions. He always helped me a lot and I felt remarkable progress during his supervisior. Also I would lie to thank to him for giving his time during the my study and my advisering.*

*I thank to **Assist. Prof. Dr. Kadri BÜRÜNCÜK** and **Mustafa GÜNDÜZ** for giving his time to me, I thank to **Prof. Dr Fakhreddin SADIKOGLU (Mamedov) and Laboratory Assistant Kamil DİMİLİLER** for helping us with our problems in our department and I thank to **Ali ÖZGEN** for giving the equipments that are used in the practical part of my project.*

*I also want to thank to all my friends and specially **Özcan ŞENER, Yasemin YILMAZ, Samet BİRİCİK, Mohammad Tariq MİR, Figen IŞIKER** and **Deniz IŞIKER** who supported and helped me all the time. I will never forget my wonderful times that I spent in Cyprus and Near East University with my good friends who helped my incorporeally during my studying in college.*

*Finally, special thanks for my family, especially my parents for being patientfull during my undergraduate degree study. I could never have completed my study without their encouragement and endless support.*

***Gülşah Tanrıkulu***

# ABSTRACT

As in olden days the industries and all other process control systems used in usual life was controlled and operated by using relays was a reliable and less costly instalation hardware. The problems comes when any system is desired to alter according to modern developments. To install a new system in industry by using relays was very tedious and expensive work therefore the need of exploring a reliable and portable system comes. After transistor and IC invention, it became easy to design a more advance control system that can alter operation by using hand held keyboard. Then invention of PLC came into being. PLC brought revoulationary changes in industrialisation. PLC brought automation in industrial application and now a days PLC is used in all kind of autmation purposes. It made industrial process control system very convience and easy to alter the operation.

The aim of my project is to explain the importance and demonstarte the use of PLC in usual life application.

# INTRODUCTION

Industry has begun to recognize the need for quality improvement and increase in productivity in the sixties and seventies. Flexibility also became a major concern ( ability to change a process quicly became very important in order to satisfy consumer needs).

Try to imagine automated industrial production line in the sixties and seventies. There was always a huge electrical board for system controls, and not infrequently it covered an entire wall. Within this board there was a great number of interconnected electromechanical relays to make the whole system work. By word 'connected' it was understood that electrician had to connect all relays manually using wires. An engineer would design logic for a system, and electricians would receive a schematic outline of logic that they had to implement with relays. These relay schemas often contained hundreds of relays. The plan that electrician was given was called 'ladder schematic'.Ladder displayed all switches, sensors, motors,valves, relays, etc. found in the system. Electrician's job was to connect them all together. One of the problems with this type of control was that it was based on mechanical relays. Mechanical instruments were usually the weakest connection in the system due to their moveable parts that could wear out. If one relay stopped working, electrician would be examine an entire system (system would be out until a cause of the problem was found and corrected).

The other problem with this type of control was in the system's break period when a system had to be turned off, so connections could be made on the electrical board. If a firm decided to change the order of operations (make even a small change), it would turn out to be a major expense and a loss of production time until a system was functional again.

It's not hard to imagine an engineer who makes a few small arrors during his project. It is also conceivable that electrician has made a few mistake in connecting the system. Finally, you can also imagine having a few bad companents. The only way to see if avarything is all right is to run the system. As system are usually not perfect with a first try, finding errors was an arounds process. You should also keep in mind that a product could not be made during these corrections and changes in connections. System had to be literally disabled before changes were to be performed. That meant that the entire production staff in that line of production was out of work until the system fixed up again.

Only when electrician was done finding errors and repeating, the system was ready for production. Expenditures for this kind of work were too great even for well-to-do companies.

The project consists of the three chapters, conclusion and reference.

Chapter-1 presents the use PLC`s used for industrial controlling, types of PLC`s, PLC layout, PLC operating system.

Chapter-2 presents PLC structure

Chapter-3 presents the informations about Siemens SIMATIC S7-200 micro controller and explanations of the devices that are used in the car park area applications.

# CHAPTER 1
# PLC AND THE TYPES OF PLC

## 1.1 What is a PLC ?

Programmable logic controllers, also called programmable controllers or PLC`s, are solid-state members of the computer family, using integrated circuits instead of electromechanical devices to implement control functions. They are capable of storing instructions, such as sequencing, timing, counting, arithmetic, data manipulation, and communication, to control industrial machines and processes. Figure 1-1 illustrates a conceptual diagram of a PLC application.

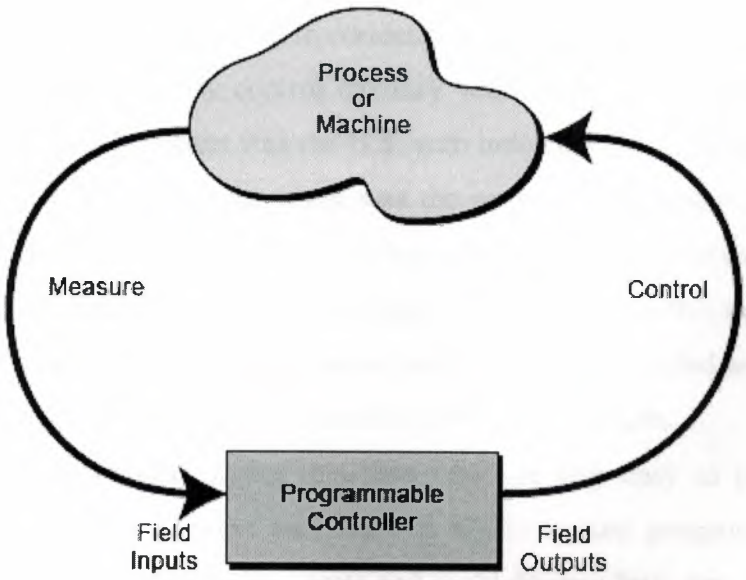



**Figure 1.1** PLC conceptual application diagram.

Programmable controllers have many definitions. However, PLC`s can be thought of in simple terms as industrial computers with specially designed architecture in both their central units (the PLC itself) and their interfacing circuitry to field devices (input/output connections to the real world).

## 1.2 PLC History

Early machines were controlled by mechanical means using cams, gears, levers and other basic mechanical devices. As the complexity grew, so did the need for a more sophisticated control system. This system contained wired relay and switch control elements. These elements were wired as required to provide the control logic necessary for the particular type of machine operation. This was acceptable for a machine that never needed to be changed or modified, but as manufacturing techniques improved and plant changeover to new products became more desirable and necessary, a more versatile means of controlling this equipment had to be developed. Hardwired relay and switch logic was cumbersome and time consuming to modify. Wiring had to be removed and replaced to provide for the new control scheme required. This modification was difficult and time consuming to design and install and any small "bug" in the design could be a major problem to correct since that also required rewiring of the system. A new means to modify control circuitry was needed. The development and testing ground for this new means was the U.S. auto industry. The time period was the late 1960's and early 1970's and the result was the programmable logic controller, or PLC. Automotive plants were confronted with a change in manufacturing techniques every time a model changed and, in some cases, for changes on the same model if improvements had to be made during the model year. The PLC provided an easy way to reprogram the wiring rather than actually rewiring the control system.

The PLC that was developed during this time was not very easy to program. The language was cumbersome to write and required highly trained programmers. These early devices were merely relay replacements and could do very little else. The PLC has at first gradually, and in recent years rapidly developed into a sophisticated and highly versatile control system component. Units today are capable of performing complex math functions including numerical integration and differentiation and operate at the fast microprocessor speeds now available. Older PLC`s were capable of only handling discrete inputs and outputs (that is, on-off type signals), while today's systems can accept and generate analog voltages and currents as well as a wide range of voltage levels and pulsed signals. PLC`s are also designed to be rugged. Unlike their personal computer cousin, they can typically withstand vibration, shock, elevated temperatures, and electrical noise to which manufacturing equipment is exposed.

develop their own versions of ladder logic language (the language used to program PLC`s). This complicates learning to program PLC's in general since one language cannot be learned that is applicable to all types. However, as with other computer languages, once the basics of PLC operation and programming in ladder logic are learned, adapting to the various manufacturers' devices is not a complicated process. Most system designers eventually settle on one particular manufacturer that produces a PLC that is personally comfortable to program and has the capabilities suited to his or her area of applications.

## 1-3 Advantages Of PLC`S

In general, PLC architecture is modular and flexible, allowing hardware and software elements to expand as the application requirements change. In the event that an application outgrows the limitations of the programmable controller, the unit can be easily replaced with a unit having greater memory and I/O capacity, and the old hardware can be reused for a smaller application.

A PLC system provides many benefits to control solutions, from reliability and repeatability to programmability. The benefits achieved with programmable controllers will grow with the individual using them—the more you learn about PLC`s, the more you will be able to solve other control problems.

Below is list some of the features of benefits obtained with a programmable controller.

Features and benefits of programmable contorller:

- Solid state
  Has high reliability
- Programable memory
  Very simply to change and flexible control
- Small size
  Has minimal space requirments
- Microprocessor based
  Communication capability is very high more over it has multifunctional capability
- Software timers/counters
  Easily changed preset
- Software control relays

Reduce hardware/wiring cost and space requirments

- Modular I/O interface

  Has facility of neat apperance of control panel, easily maintained and easily wired

- Modular architecture

  Has facility of instalation flexibility, easily instalable and expandability

- System variables stored in memory data

  Useful managment/maintenance and output can be in report form

PLC`s provide many other facilities and Opportunities such as remote I/O stations, varity of I/O interface,quick I/O disconnect, diagonastic indicator that are very beneficial in industrial control.

## 1.4 Types of PLC

There are two types of PLC`s which are Compact PLC`s & Modular PLC`s. The more detailed information will be given below.

### 1.4.1 Compact PLC`s

Compact PLC`s are manufactured such that all units forming the PLC are placed in a case. They are low price PLC with lower capacity. Small or medium size machine manufacturers usually prefer them. In some types compact enlargement module is present.

### 1.4.2 Modular PLC`s

Combining separate modules together in a board forms them. They can have different memory capacity, I / O numbers, power supply up to the necessary limits.

## 1.5 PLC Configurations

Programmable controllers (the shortened name used for programmable logic controllers) are much like personal computers in that the user can be overwhelmed by the vast array of options and configurations available. Also, like personal computers, the best teacher of which one to select is experience.

As one gains experience with the various options and configurations available, it becomes less confusing to be able to select the unit that will best perform in a particular application. Basic PLC`s are available on a single printed circuit board as shown in Figure 1.2.

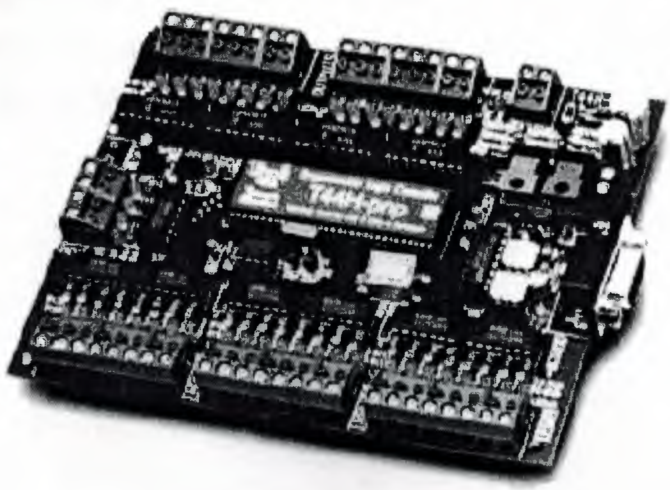


**Figure 1.2** Open Frame PLC
(Triangle research Inc., Pte. Ltd.)

They are sometimes called single board PLC`s or open frame PLC`s. These are totally self contained (with the exception of a power supply) and, when installed in a system, they are simply mounted inside a controls cabinet on threaded standoffs. Screw terminals on the printed circuit board allow for the connection of the input, output, and power supply wires. These units are generally not expandable, meaning that extra inputs, outputs, and memory cannot be added to the basic unit. However, some of the more sophisticated models can be linked by cable to expansion boards that can provide extra I/O. Therefore, with few exceptions, when using this type of PLC, the system designer must take care to specify a unit that has enough inputs, outputs, and programming capability to handle both the present need of the system and any future modifications that may be required. Single board PLC`s are very inexpensive (some less than $100), easy to program, small, and consume little power, but, generally speaking, they do not have a large number of inputs and outputs, and have a somewhat limited instruction set. They are best suited to small, relatively simple control applications.

PLC`s are also available housed in a single case (sometimes referred to as a shoe box) with all input and output, power and control connection points located on the single unit, as shown in Figure 1.3. These are generally chosen according to available program memory and required number and voltage of inputs and outputs to suit the application. These systems generally have an expansion port (an interconnection socket) which will allow the addition of specialized units such as high speed counters and analog input and output units or additional discrete inputs or outputs. These expansion units are either plugged directly into the main case or connected to it with ribbon cable or other suitable cable.

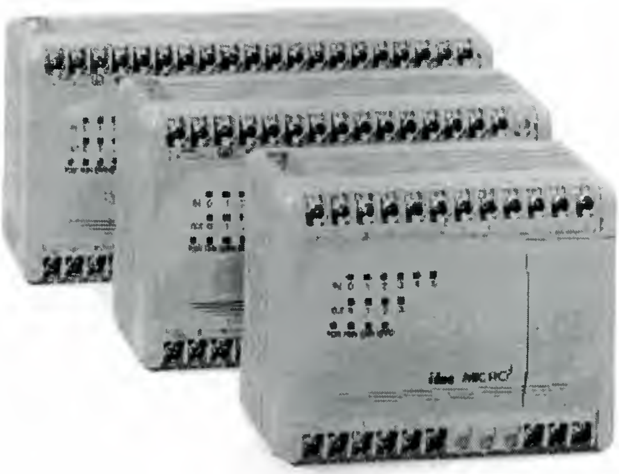


**Figure 1.3** Shoe box – Style PLC`s (IDEC Corp.)

More sophisticated units, with a wider array of options, are modularized. An example of a modularized PLC is shown in Figure 1.4.

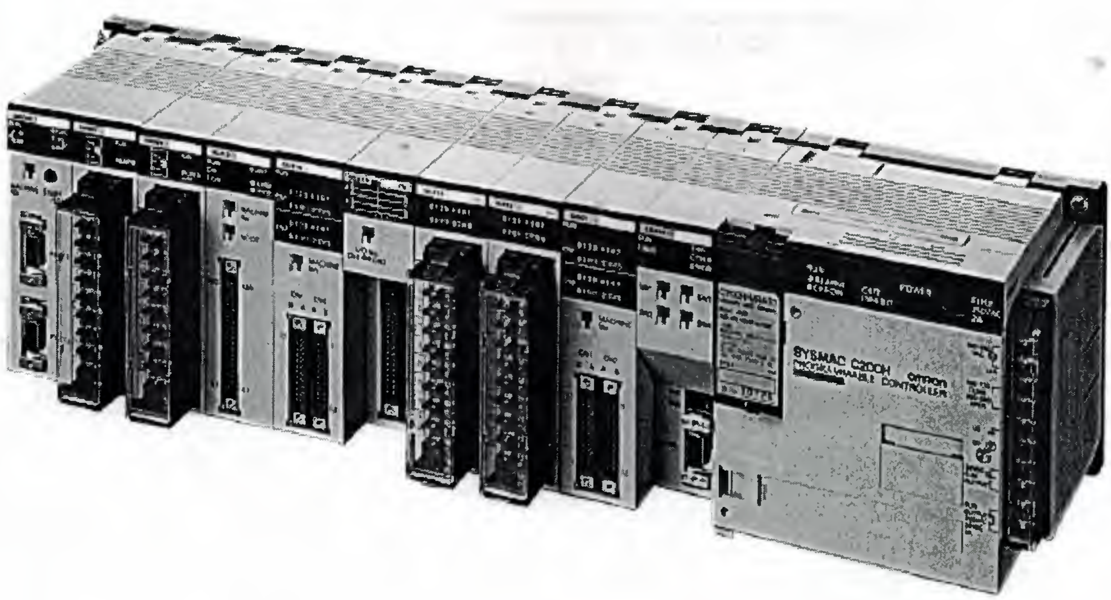



**Figure 1.4** Modularized PLC (Omron Electronics)

## 1.6 PLC Sizes and Scopes of Applications

Prior to evaluating the system requirements, the designer should understand the different ranges of programmable controller products and the typical features found within each range. This understanding will enable the designer to quickly identify the type of product that comes closest to matching the application's requirements. Figure 1.5, illustrates PLC product ranges divided into five major areas with overlapping boundaries. The basis for this product segmentation is the number of possible inputs and outputs the system can accommodate (I/O count), the amount of memory available for the application program, and the system's general hardware and software structure. As the I/O count increases, the complexity and cost of the system also increase. Similarly, as the system complexity increases, the memory capacity, variety of I/O modules, and capabilities of the instruction set increase as well.

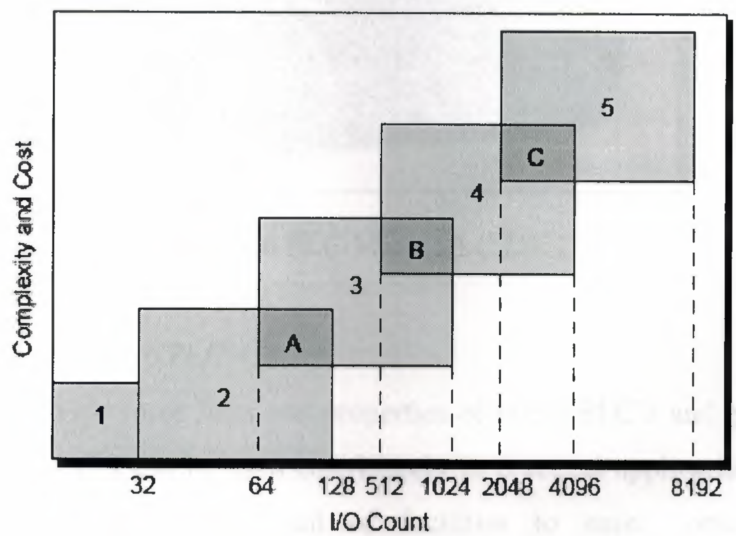



**Figure 1.5** PLC Product Ranges

The shaded areas in Figure 1.5, labeled A, B, and C, reflect the possibility of controllers with enhanced (not standard) features for a particular range. These enhancements place the product in a gray area that overlaps the next higher range. For example, because of its I/O count, a small PLC would fall into area 2, but it could have analog control functions that are standard in medium-sized controllers. Thus, this type of product would belong in area A.

Products that fall into these overlapping areas allow the user to select the product that best matches the application's requirements, without having to select the larger product, unless it is necessary. The following discussion presents information about the five PLC categories, as well as the overlapping categories.

### 1.6.1 Micro PLC`s

Micro PLC`s are used in applications that require the control of a few discrete I/O devices, such as small conveyor controls. Some micro PLC`s can perform limited analog I/O monitoring functions (e.g., monitoring a temperature set point or activating an output). Figure 1.6 shows a typical microcontroller, while Table 1.2 lists the standard features of micro PLC`s.


**Figure 1.6** PLC Micro PLC DL 105.

Standard Features of Micro PLC's:

This paragraph provide some facts and properties of micro PLC`s and give information about how micro PLC`s can be used convienecely in practical applications according to desire specifications moreover detail of facilities to ease practical operation furthurmore the way to operate the micro PLC`s.

- Up to 32 I/O
- 16-bit processor
- Relay replacer
- Memory up to 1K
- Digital I/O
- Built-in I/Os in a compact unit
- Master control relays
- Timers and counters
- Programmed with handheld programmer

## 1.6.2 Small PLC's

Small controllers are mostly used in applications that require ON/OFF control for logic sequencing and timing functions. These PLC`s, along with microcontrollers, are widely used for the individual control of small machines. Often, these products are single-board controllers. Table 1.3 lists the standard features of small PLC`s.

Area A. Area A includes controllers that are capable of having up to 64 or 128 I/O, along with products that have features normally found in medium-sized controllers.

The enhanced capabilities of these small controllers allow them to be used effectively in applications that need only a small number of I/O, yet require analog control, basic math, I/O bus network interfaces, LANs, remote I/O, and/or limited data-handling capabilities (see Figure 1.7 ). A typical application of an area A controller is a transfer line in which several small machines, under individual control, must be interlocked through a LAN.

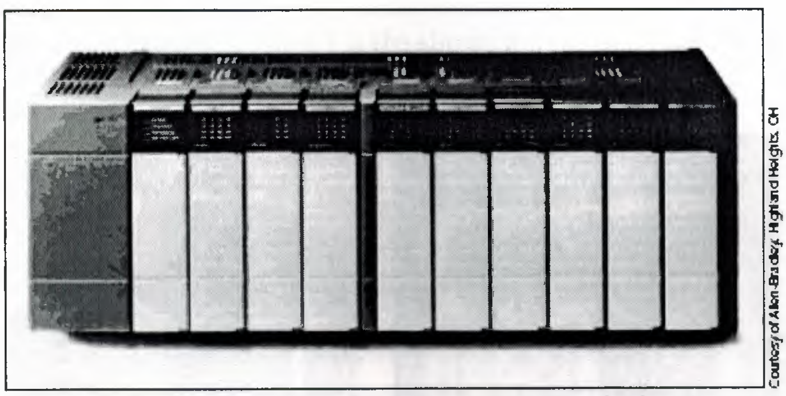



**Figure 1.7** Area A (SLC500) Controller Capable of Handling up to 72 Discrete and 4 Analog I/O.

Below, the features of small PLC's are given. The details, moreover provide information that what facilities small PLC`s can provide in practical applications.

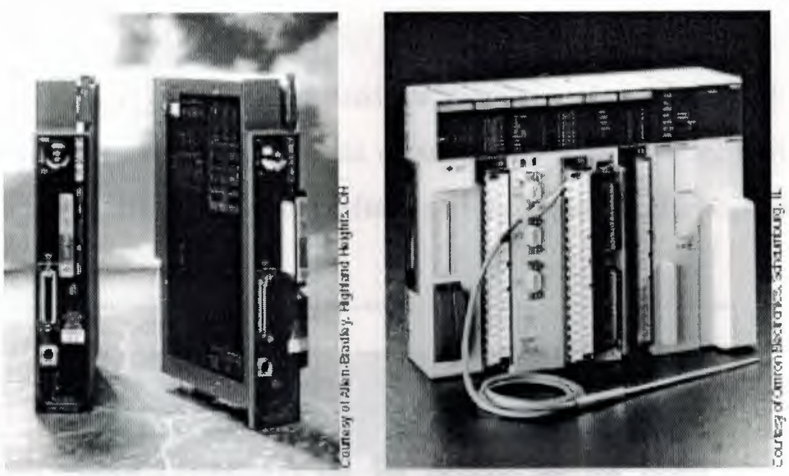
- Up to 128 I/O
- 16-bit processor
- Relay replacer
- Memory up to 2k

- Digital I/O only
- Ladder or Boolean language only
- Master control relays
- Timers/counters/shift registers
- Drum timers sequencers
- Programmed with handheld programmer

### 1.6.3 Medium PLC`s

Medium PLC`s (see Figure 1.8 (a) ) are used in applications that require more than 128 I/O, as well as analog control, data manipulation, and arithmetic capabilities. In general, the controllers in segment 3 have more flexible hardware and software features than the controllers previously mentioned. Table 1.4 lists these features.

**Area B.** Area B contains medium PLC`s that have more memory, tablehandling, PID, and subroutine capabilities than typical medium-sized PLC`s, as well as more arithmetic and data-handling instructions. Figure 1.8 (b) shows a PLC that falls into this category.



**(a)** **(b)**

**Figure 1.8 (a)** Medium-sized PLC 5/11 (left) and PLC 5/20 (right) processors with up to 5 12 I/O capacity. **(b)** Omron's area b CV500 PLC with a temperature control module (up to 1024 I/O).

Standard Features of Medium PLC's are given below.

- Up to 1024 I/O
- 16 or 32 bit processor
- Relay replacer and analog control
- Memory up to 4k words(expandable to 16k)
- Digital and Analog I/O
- Local and remote I/O
- Use of ladder and boolean language
- Master control relays
- Drum timers and sequencers
- Math Capabilities(Addition, subtraction, multiplication, division)
- Limited data handling(compare, data conversion, matrix fucctions)
- RS-232 communication port
- Local area networks
- Support I/O bus networks

### 1.6.4 Large PLC`s

Large controllers (see Figure 1.9) are used for more complicated control tasks, which require extensive data manipulation, data acquisition, and reporting. Further software enhancements allow these products to perform complex numerical computations.

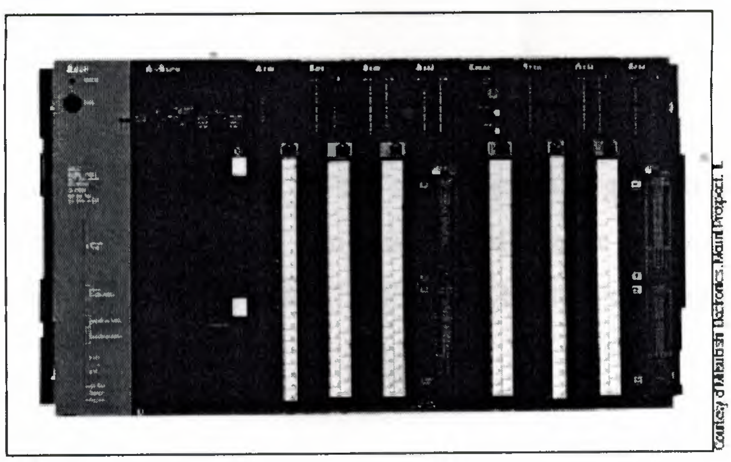


**Figure 1.9** Large Mitsubishi A3NCPU controller with 2048 I/O capacity.

Below the list of benefits and features of large PLC`s are given. Standard features of large PLC`s are;

- Up to 4096 I/O
- 16 or 32 bit processor
- Relay replacer and analog control
- Memory up to 12k words
- Expandable to 128k
- Digital and analog I/O
- Local and remote I/O
- Ladder or boolean language
- Master control relays
- Timers/counters/shift registers
- Drum timers and sequencers
- Subroutines and interrupts
- PID modules or software PID
- One or more RS-232 communication ports
- Host computer communication modules
- Math Capabilities( Addition, subtraction, mutliplication, division, squra root, double precsion)
- Extended data handling(Compare, data conversion, mover register/file, matrix functions, block transfer, binary table, ASCII table)
- Local area networks
- Support I/O bus netwroks

Area C. Area C includes the segment 4 PLC`s that have a large amount of application memory and I/O capacity. The PLC`s in this area also have greater math and data-handling capabilities than other large PLC`s. Figure 1.10 shows an example of this type of controller.

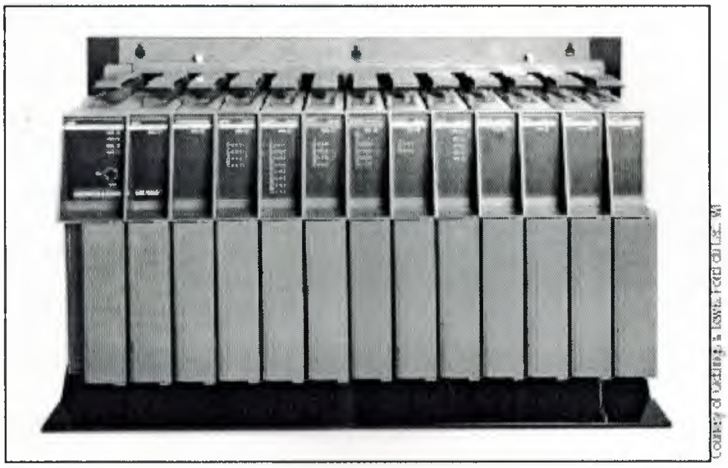


**Figure 1.10.** Giddings & Lewis's area C PIC900 with up to 3168 I/O and motion I/O, IEC programming, and floating-point math capabilities

## 1.6.5 Very Large PLC`s

Very large PLC`s (see Figure 1.11) are utilized in sophisticated control and data acquisition applications that require large memory and I/O capacities. Remote and special I/O interfaces are also standard requirements for this type of controller. Typical applications for very large PLC`s include steel mills and refineries. These PLC`s usually serve as supervisory controllers in large, distributed control applications. Table 1.6 lists standard features found in segment 5 PLC`s.

**Figure 1.11.** Very large PLC-3 from Allen-Bradley with 8190 I/O capability.

Standard Features of Very Large PLC's are given below.

- Up to 8192 I/O
- 32 bit processor or multiprocessors

- Relay replacer and analog control
- Memory up to 64k words(1 meg)
- Digital and analog I/O
- Remote analog I/O and special modules
- Local and remote I/O
- Ladder and Boolean laguage
- Fuction bolck/High level language
- Master control relays
- Timers/counters/shift registers
- Drum timers and sequencer
- Jump
- Subroutines and intrrupts
- Special fuction I/O modules
- Local area network
- PID modules or system software PID
- Two or more RS-232 communication ports
- Host computer communication modules
- Math capabilities(Addition, subtraction, multiplication, division, sqaure root, double precision, floating point, cosine functions)
- Powerful data handling(compare, data converison, move register/file, matrix functions, block transfer, binary table, ASCII table, LIFO, FIFO)
- Machine diagnostics
- Support I/O bus networks

CHAPTER 2

INTERNAL STRUCTURE

&

PROGRAMMING TECHNICS OF PLC

## 2.1 Principle of PLC Operation

A programmable controller, as illustrated in Figure 2.1, consists of two basic sections:

• the central processing unit

• the input/output interface system

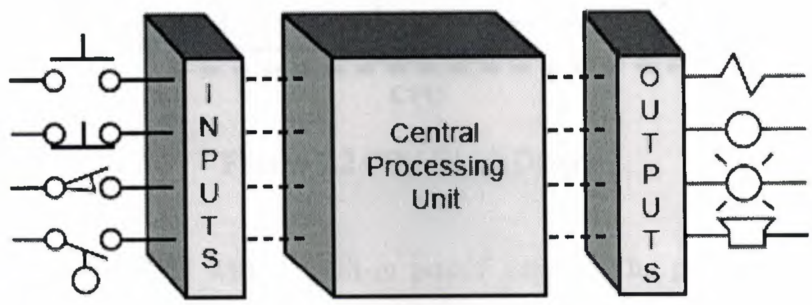



**Figure 2.1** Programmable Controller Block Diagram.

## 2.2 Central Processing Unit

The central processing unit, or CPU, is the most important element of a PLC. The CPU forms what can be considered to be the "brain" of the system. The three components of the CPU are:

• the processor

• the memory system

• the power supply

Figure 2.2 illustrates a simplified block diagram of a CPU. CPU architecture may differ from one manufacturer to another, but in general, most CPUs follow this typical three-component organization. Although this diagram shows the power supply inside the CPU block enclosure, the power supply may be a separate unit that is mounted next to the block enclosure containing the processor and memory.

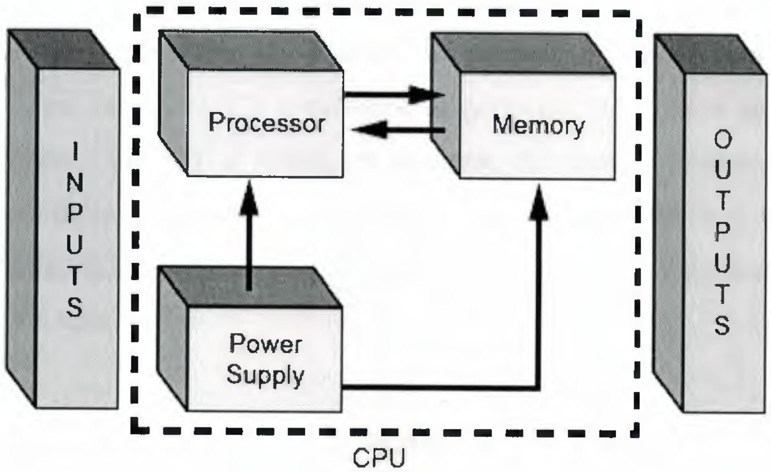



**Figure 2.2** CPU Block Diagram.

Figure 2.3 shows a CPU with a built-in power supply. The programming device, not regarded as part of the CPU per se, completes the total central architecture as the medium of communication between the programmer and the CPU.

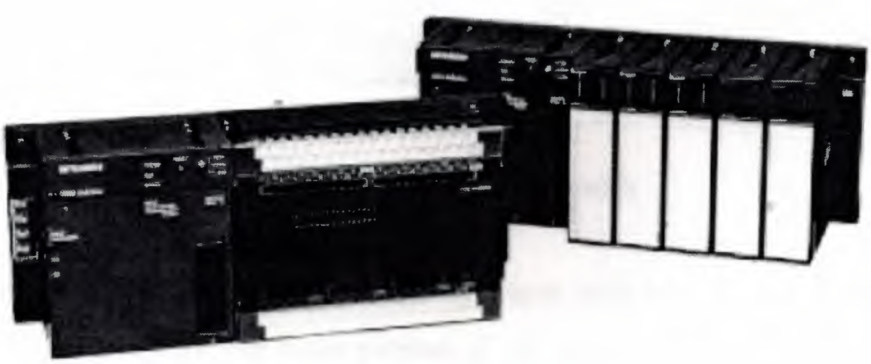


**Figure 2.3** Two PLC CPUs with built-in power supplies (left with fixed I/O blocks and right with configurable I/O).

The operation of a programmable controller is relatively simple. The input/output (I/O) system is physically connected to the field devices that areencountered in the machine or that are used in the control of a process. Thesefield devices may be discrete or analog input/output devices, such as limit switches, pressure transducers, push buttons, motor starters, solenoids, etc.The I/O interfaces provide the connection between the CPU and the information providers (inputs) and controllable devices (outputs). During its operation, the CPU completes three processes: (1) it reads, or accepts, the input data from the field devices via the input interfaces, (2) it executes, or performs, the control program stored in the memory system, and (3) it writes, or updates, the output devices via the output interfaces. This process of sequentially reading the inputs, executing the program in memory, and updating the outputs is known as scanning. Figure 2.4 illustrates a graphic representation of a scan.

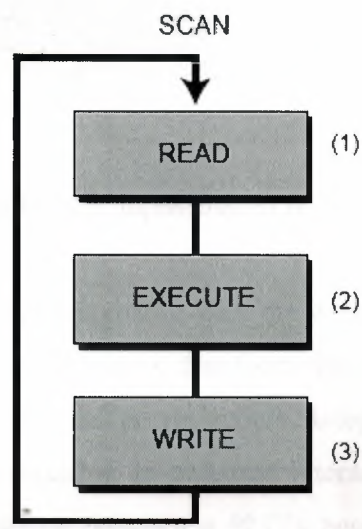



**Figure 2.4** Illustration of a scan.

The input/output system forms the interface by which field devices are connected to the controller (see Figure 2.5). The main purpose of the interface is to condition the various signals received from or sent to external field devices. Incoming signals from sensors (e.g., push buttons, limit switches, analog sensors, selector switches, and thumbwheel switches) are wired to terminals on the input interfaces. Devices that will be controlled, like motor starters, solenoid valves, pilot lights, and position valves, are connected to the terminals of the output interfaces. The system power supply provides all the voltages required for the proper operation of the various central processing unit sections.

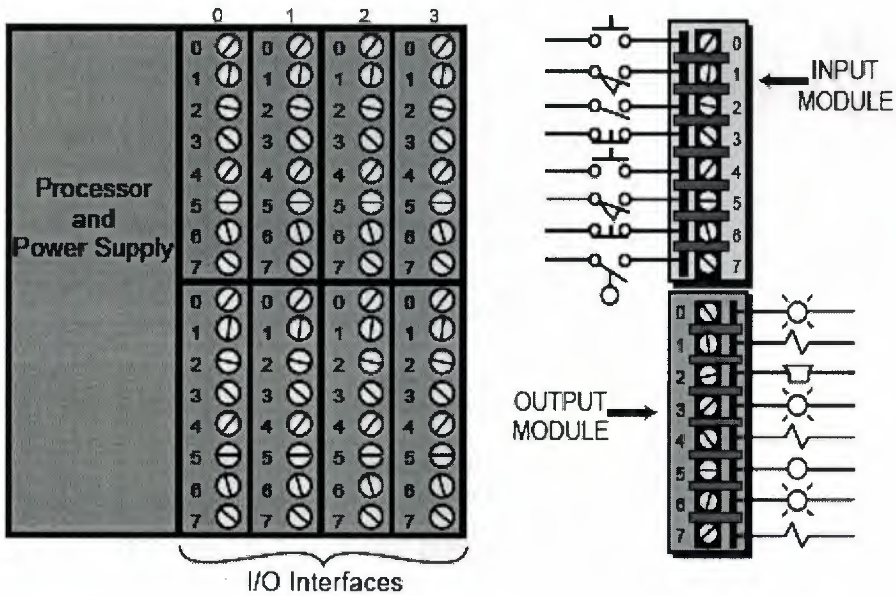



**Figure 2.5** Input/Output Interface.

The term CPU is often used interchangeably with the word processor; however, the CPU encompasses all of the necessary elements that form the intelligence of the system—the processor plus the memory system and power supply. Integral relationships exist between the components of the CPU, resulting in constant interaction among them. Figure 2.6 illustrates the functional interaction between a PLC's basic components. In general, the processor executes the control program stored in the memory system in the form of ladder diagrams, while the system power supply provides all of the necessary voltage levels to ensure proper operation of the processor and memory components.

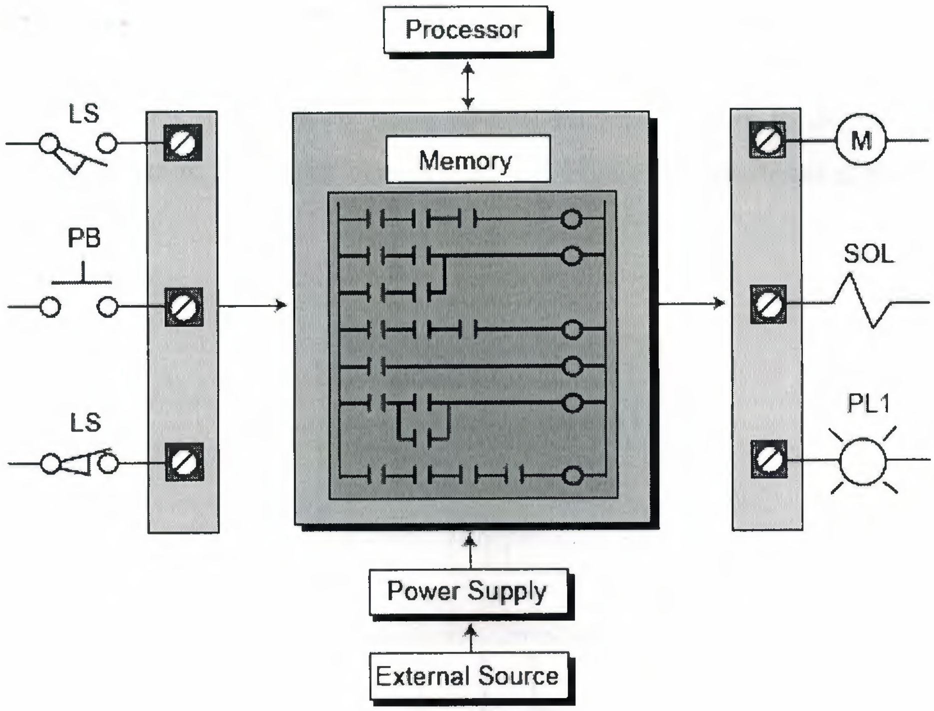



**Figure 2.6** Functional Interaction of a PLC System.

### 2.2.1 Processors

Very small microprocessors (or micros)—integrated circuits with tremendous computing and control capability—provide the intelligence of today's programmable controllers. They perform mathematical operations, data handling, and diagnostic routines that were not possible with relays or their predecessor, the hardwired logic processor. The principal function of the processor is to command and govern the activities of the entire system. It performs this function by interpreting and executing a collection of system programs known as the executive. The executive, a group of supervisory programs, is permanently stored in the processor and is considered a part of the controller itself. By executing the executive, the processor can perform all of its control, processing, communication, and other housekeeping functions. The executive performs the communication between the PLC system and the user via the programming device. It also supports other peripheral communication, such as monitoring field devices; reading diagnostic data from the power supply, I/O modules, and memory; and communicating with an operator interface.

The CPU of a PLC system may contain more than one processor (or micro) to execute the system's duties and/or communications, because extra processors increase the speed of these operations. This approach of using several microprocessors to divide control and communication tasks is known as multiprocessing. Figure 2.7 illustrates a multiprocessor configuration.

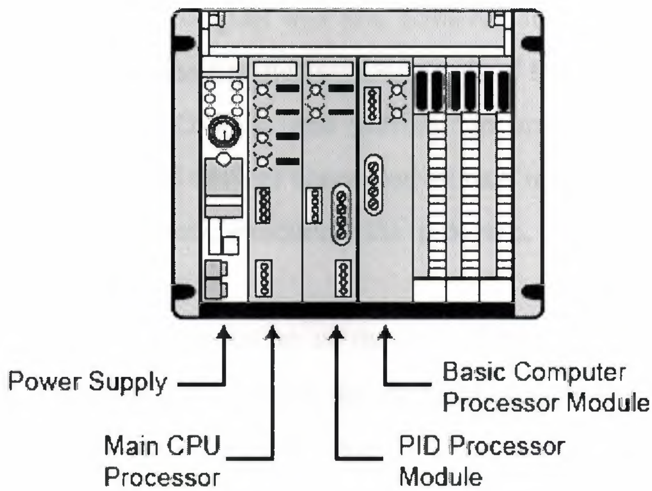



**Figure 2.7** A multiprocessor configuration.

Another multiprocessor arrangement takes the microprocessor intelligence away from the CPU, moving it to an intelligent module. This technique uses intelligent I/O interfaces, which contain a microprocessor, built-in memory, and a mini-executive that performs independent control tasks. Typical intelligent modules are proportional-integral-derivative (PID) control modules, which perform closed-loop control independent of the CPU, and some stepper and servo motor control interfaces.

The microprocessors used in PLCs are categorized according to their word size, or the number of bits that they use simultaneously to perform operations. Standard word lengths are 8, 16, and 32 bits. This word length affects the speed at which the processor performs most operations. For example, a 32- bit microprocessor can manipulate data faster than a 16-bit micro, since it manipulates twice as much data in one operation. Word length correlates with the capability and degree of sophistication of the controller (i.e., the larger the word length, the more sophisticated the controller).

### 2.2.2 Processor Scan

The basic function of a programmable controller is to read all of the field input devices and then execute the control program, which according to the logic programmed, will turn the field output devices ON or OFF. In reality, this last process of turning the output devices ON or OFF occurs in two steps. First, as the processor executes the internal programmed logic, it will turn each of its programmed internal output coils ON or OFF. The energizing or deenergizing of these internal outputs will not, however, turn the output devices ON or OFF. Next, when the processor has finished evaluating all of the control logic program that turns the internal coils ON or OFF, it will perform an update to the output interface modules, thereby turning the field devices connected to each interface terminal ON or OFF. This process of reading the inputs, executing the program, and updating the outputs is known as the scan.

Figure 2.8 shows a graphic representation of the scan. The scanning process is repeated over and over in the same fashion, making the operation sequential from top to bottom. Sometimes, for the sake of simplicity, PLC manufacturers call the solving of the control program the program scan and the reading of inputs and updating of outputs the I/O update scan. Nevertheless, the total system scan includes both. The internal processor signal, which indicates that the program scan has ended, is called the end-of-scan (EOS) signal.

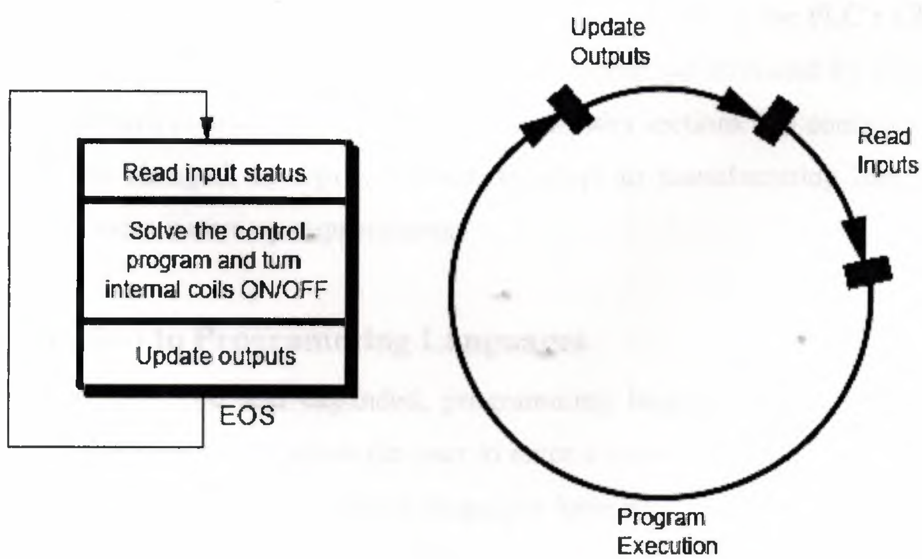



**Figure 2.8** PLC total scan representation.

The time it takes to implement a scan is called the scan time. The scan time is the total time the PLC takes to complete the program and I/O update scans. The program scan time generally depends on two factors: (1) the amount of memory taken by the control program and (2) the type of instructions used in the program (which affects the time needed to execute the instructions). The time required to make a single scan can vary from a few tenths of a millisecond to 50 milliseconds.

### 2.2.3 Power Supply

The system power supply plays a major role in the total system operation. In fact, it can be considered the "first-line manager" of system reliability and integrity. Its responsibility is not only to provide internal DC voltages to the system components (i.e., processor, memory, and input/output interfaces), but also to monitor and regulate the supplied voltages and warn the CPU if something is wrong. The power supply, then, has the function of supplying well-regulated power and protection for other system components.

### 2.2.4 Memory

The most important characteristic of a programmable controller is the user's ability to change the control program quickly and easily. The PLC's architecture makes this programmability feature possible. The memory system is the area in the PLC's CPU where all of the sequences of instructions, or programs, are stored and executed by the processor to provide the desired control of field devices. The memory sections that contain the control programs can be changed, or reprogrammed, to adapt to manufacturing line procedure changes or new system start-up requirements.

## 2.3 Introduction to Programming Languages

As PLCs have developed and expanded, programming languages have developed with them. Programming languages allow the user to enter a control program into a PLC using an established syntax. Today's advanced languages have new, more versatile instructions, which initiate control program actions. These new instructions provide more computing power for single operations performed by the instruction itself. For instance, PLCs can now transfer blocks of data from one memory location to another while, at the same time, performing a logic or arithmetic operation on another block. As a result of these new, expanded instructions, control programs can now handle data more easily. In addition to new programming instructions, the development of powerful I/O modules has also changed existing instructions. These changes include the ability to send data to and obtain data from modules by addressing the modules' locations. For example, PLCs can now read and write data to and from analog modules. All of these advances, in conjunction with projected industry needs, have created a demand for more powerful instructions that allow easier, more compact, function-oriented PLC programs.

## 2.4 Types of PLC Languages

The three types of programming languages used in PLCs are:

- ladder
- Boolean
- Grafcet

The ladder and Boolean languages essentially implement operations in the same way, but they differ in the way their instructions are represented and how they are entered into the PLC. The Grafcet language implements control instructions in a different manner, based on steps and actions in a graphicoriented program.

### 2.4.1 Ladder Language

The programmable controller was developed for ease of programming using existing relay ladder symbols and expressions to represent the program logic needed to control the machine or process. The resulting programming language, which used these original basic relay ladder symbols, was given the name ladder language. Figure 2.9 illustrates a relay ladder logic circuit and the PLC ladder language representation of the same circuit.

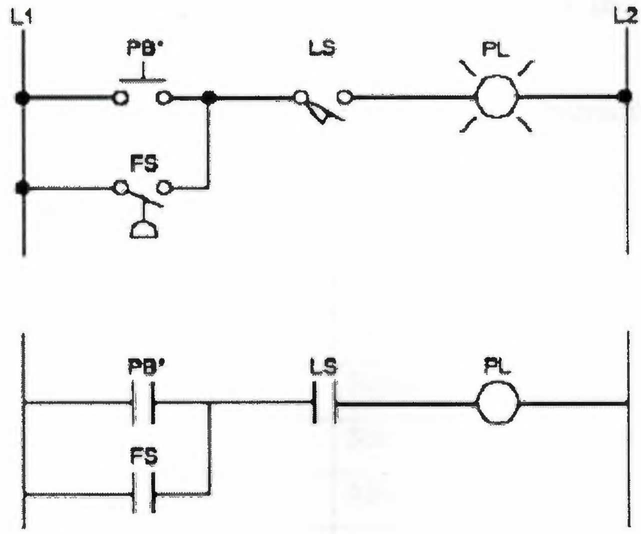



**Figure 2.9** Hardwired Logic Circuit and Its PLC Ladder Language Implementation.

The evolution of the original ladder language has turned ladder programming into a more powerful instruction set. New functions have been added to the basic relay, timing, and counting operations. The term function is used to describe instructions that, as the name implies, perform a function on data— that is, handle and transfer data within the programmable controller. These instructions are still based on the simple principles of basic relay logic, although they allow complex operations to be implemented and performed. New additions to the basic ladder logic also include function blocks, which use a set of instructions to operate on a block of data. The use of function blocks increases the power of the basic ladder language, forming what is known as enhanced ladder language.

The ladder languages available in PLCs can be divided into two groups:

• basic ladder language

• enhanced ladder language

Each of these groups consists of many PLC instructions that form the language. The classification of which instructions fall into which categories differs among manufacturers and users, since a definite classification does not exist. However, a de facto standard has been created throughout the years that sorts the instructions into either the basic or enhanced ladder language.

Table 2.1 shows a typical classification of basic and enhanced instructions.

**Table 2.1** PLC Instruction Set Classifications.

| Basic | Enhanced |
|---|---|
| Relay contact | Double-precision arithmetics |
| Relay output | Square root |
| Timer | Sort |
| Counter | Move register |
| Latch | Move register to table |
| Jump to-go to | First in-first out |
| Master control relay | Shift register |
| End | Rotate register |
| Addition | Diagnostic block |
| Subtraction | Block transfer (in/out) |
| Multiplication | Sequencer |
| Division | PID |
| Compare (=,<,>) | Network |
| Go to subroutine | Logic matrix |

Sometimes, basic ladder instructions are referred to as low-level language, while enhanced ladder functions are referred to as high-level language. The line that defines the grouping of PLC ladder instructions, however, is usually drawn between functional instruction categories. These instruction categories include:

• ladder relay
• timing
• counting
• program/flow control
• arithmetic
• data manipulation

• data transfer

• special function (sequencers)

• network communication

Although these categories are straightforward, the classification of them is subjective. For example, some people believe that basic ladder instructions include ladder relay, timing, counting, program/flow control, arithmetic, and some data manipulation. Others believe that only ladder relay, timing, and counting categories should be considered basic ladder instructions.

### 2.4.2 Boolean

Some PLC manufacturers use Boolean language, also called Boolean mnemonics, to program a controller. The Boolean language uses Boolean algebra syntax to enter and explain the control logic. That is, it uses the AND, OR, and NOT logic functions to implement the control circuits in the control program. Figure 2.10 shows a basic Boolean program.

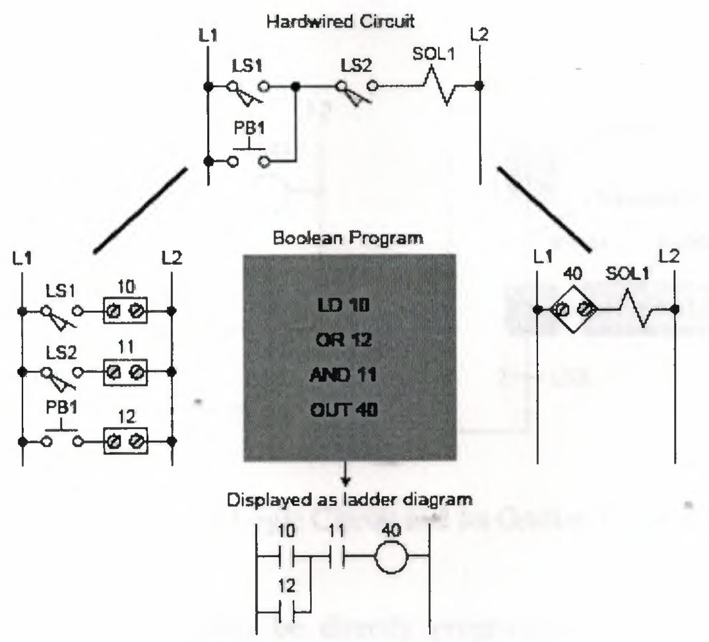



**Figure 2.10** Hardwired Logic Circuit and Its Boolean Representation.

The Boolean language is primarily just a way of entering the control program into a PLC, rather than an actual instruction-oriented language. When displayed on the programming monitor, the Boolean language is usually viewed as a ladder circuit instead of as the Boolean commands that define the instruction.

### 2.4.3 Grafcet

Grafcet (Graphe Fonctionnel de Commande Étape Transition) is a symbolic, graphic language, which originated in France, that represents the control program as steps or stages in the machine or process. In fact, the English translation of Grafcet means "step transition function charts."

Figure 2.11 illustrates a simple circuit represented in Grafcet. Note that Grafcet charts provide a flowchart-like representation of the events that take place in each stage of the control program. These charts use three components— steps, transitions, and actions—to represent events. The IEC 1131 standard's SFCs also use these components; however, the instructions inside the actions can be programmed using one or more possible languages, including ladder diagrams.

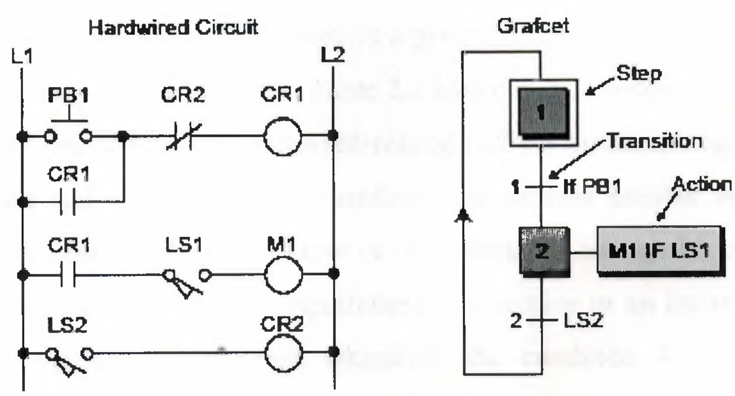



**Figure 2.11** Hardwired Logic Circuit and Its Grafcet Representation

Few programmable controllers may be directly programmed using Grafcet. However, several Grafcet software manufacturers provide off-line Grafcet programming using a personal computer. Once programmed in the PC, the Grafcet instructions can be transferred to a PLC via a translator or driver that translates the Grafcet program into a ladder diagram or Boolean language program. Using this method, a Grafcet software manufacturer can provide different PLCs that use the same "language." Figure 2.12 illustrates a typical translation that occurs when using Grafcet.

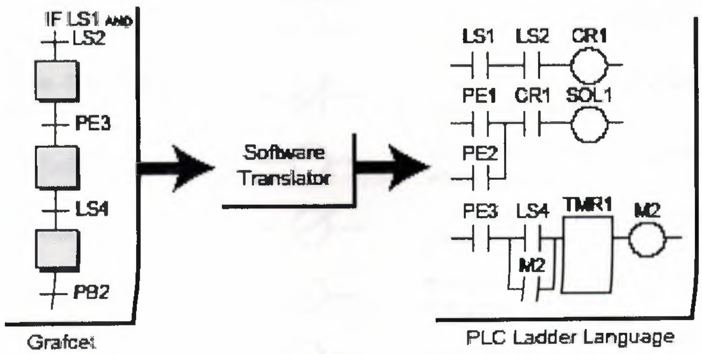



**Figure 2.12** Grafcet Translation

## 2.5 Ladder Relay Instructions

Ladder relay instructions are the most basic instructions in the ladder diagram instruction set. These instructions represent the ON/OFF status of connected inputs and outputs. Ladder relay instructions use two types of symbols: contacts and coils. Contacts represent the input conditions that must be evaluated in a given rung to determine the control of the output. Coils represent a rung's outputs. Table 2.2 lists common ladder relay instructions.

In a program, each contact and coil has a referenced address number, which identifies what is being evaluated and what is being controlled. The address number references the I/O table location of the connected input/output or the internal or storage bit output. A contact, regardless of whether it represents an input/output connection or an internal output, can be used throughout the control program whenever the condition it represents must be evaluated.

The format of the rung contacts in a PLC program depends on the desired control logic. Contacts may be placed in whatever series, parallel, or series/ parallel configuration is required to control a given output. When logic continuity exists in at least one left-to-right contact path, the rung condition is TRUE; that is, the rung controls the given output. The rung condition is FALSE if no path has continuity.

**Table 2.2** Ladder relay instructions.

| **Ladder Relay Instructions** *(Purpose: To provide hardwired relay capabilities in a PLC)* | | |
|---|---|---|
| **Instruction** | **Symbol** | **Function** |
| Examine-ON/Normally Open | | Tests for an ON condition in a reference address |
| Examine-OFF/Normally Closed | | Tests for an OFF condition in a reference address |
| Output Coil | | Turns real or internal outputs ON when logic is 1 |
| NOT Output Coil | | Turns real or internal outputs OFF when logic is 1 |
| Latch Output Coil | L | Keeps an output ON once it is energized |
| Unlatch Output Coil | U | Resets a latched output |
| One-Shot Output | OS | Energizes an output for one scan or less |
| Transitional Contact | ↑ | Closes for one scan when its trigger contact makes a positive transition |

The relay-type instructions covered in this section are the most basic programmable controller instructions. They provide the same capabilities as hardwired relay logic, but with greater flexibility. These instructions provide the ability to examine the ON/OFF status of specific bit addresses in memory and control the state of internal and external outputs.

## 2.6 Timers and Counters

PLC timers and counters are internal instructions that provide the same functions as hardware timers and counters. They activate or deactivate a device after a time interval has expired or a count has reached a preset value. Timer and counter instructions are generally considered internal outputs. Like relay-type instructions, timer and counter instructions are fundamental to the ladder diagram instruction set.

Timer instructions may have one or more time bases (TB) which they use to time an event. The time base is the resolution, or accuracy, of the timer. For instance, if a timer must count a 10 seconds event, the user must choose the number of times the time base must be counted to get to 10 seconds.

Therefore, if the timer has a time base of 1 second, then the timer must count ten times before it activates its output. This number of counts is referred to as ticks. The most common time bases are 0.01 sec, 0.1 sec, and 1 sec. Table 2.3 shows the number of ticks required for a 10 second count, based on different time bases.

**Table 2.3** Timebases

| Required Time | Number of Ticks | Time Base (secs) |
|---|---|---|
| 10 sec | 10 | 1.00 |
| 10 sec | 100 | 0.10 |
| 10 sec | 1000 | 0.01 |
| Note: Required time = (# of ticks)(Time base) | | |

Timers are used in applications to add a specific amount of delay to an output in the program. Applications of PLC timers are innumerable, since they have completely replaced hardware timers in automated control systems. As an example, timers may be used to introduce a 0.01 second delay in a control program. The program may require such a delay because the PLC turns ON its outputs very quickly as compared to the hardwired relay system it is replacing. This small delay will slow down the response of other components so that proper operation occurs.

Counter instructions are used to count events, such as parts passing on a conveyor, the number of times a solenoid is turned ON, etc. Counters, along with timers, must have two values, a preset value and an accumulated value. These values are stored in register or word locations in the data table. The preset value is the target number of ticks or counting numbers that must be achieved before the timer or counter turns its output ON. The accumulated value is the current number of ticks (timer) or counts (counter) that have elapsed during the timer or counter operation. The preset value is stored in a preset register, while the accumulated value is kept in an accumulated register. Both of these registers are defined during the programming of the instruction. Either the basic ladder format or the block instruction format can be used to implement timers and counters.

# CHAPTER 3
# PRACTICAL EXAMPLE BY USING SIEMENS PLC

## 3.1 Siemens PLCs

Siemens makes several PLC product lines in the SIMATIC® S7 family. They are: S7-200, S7-300, and S7-400 The S7-200 is referred to as a micro PLC because of its small size. The S7-200 has a brick design which means that the power supply and I/O are on-board. The S7-200 can be used on smaller, stand-alone applications such as elevators, car washes, or mixing machines. It can also be used on more complex industrial applications such as bottling and packaging machines. The S7-300 and S7-400 PLCs are used in more complex applications that support a greater number of I/O points. Both PLCs are modular and expandable. The power supply and I/O consist of seperate modules connected to the CPU. Choosing either the S7-300 or S7-400 depends on the complexity of the task and possible future expansion. Your Siemens sales representative can provide you with additional information on any of the Siemens PLCs.

## 3.2 S7-200 Micro PLCs

The S7-200 Micro PLC is the smallest member of the SIMATIC S7 family of programmable controllers. The central processing unit (CPU) is internal to the PLC. Inputs and outputs (I/O) are the system control points. Inputs monitor field devices, such as switches and sensors. Outputs control other devices, such as motors and pumps. The programming port is the connection to the programming device (See Figure 3.1).

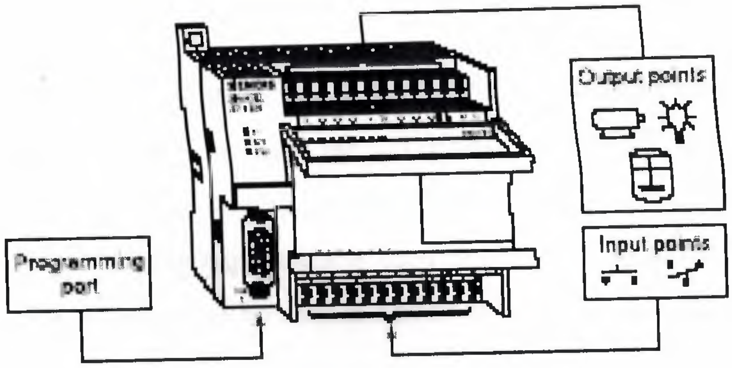



**Figure 3.1** Siemens S7-200 Micro PLC

There are three S7-200 CPU types: S7-221, S7-222, and S7-224. There are three power supply configurations for each CPU type (Table 3.1). The model description indicates the type of CPU, the power supply, the type of input, and the type of output (Figure 3.2).

**Table 3.1** S7-200 CPU Types

| Model Description | Power Supply | Input Types | Output Types |
|---|---|---|---|
| 221 DC/DC/DC<br>221 AC/DC/Relay | 20.4-28.8 VDC<br>85-264 VAC<br>47-63 Hz | 6 DC Inputs<br>6 DC Inputs | 4 DC Outputs<br>4 Relay Outputs |
| 222 DC/DC/DC<br>222 AC/DC/Relay | 20.4-28.8 VDC<br>85-264 VAC<br>47-63 Hz | 8 DC Inputs<br>8 DC Inputs | 6 DC Outputs<br>6 Relay Outputs |
| 224 DC/DC/DC<br>224 AC/DC/Relay | 20.4-28.8 VDC<br>85-264 VAC<br>47-63 Hz | 14 DC Inputs<br>14 DC Inputs | 10 DC Outputs<br>10 Relay Outputs |

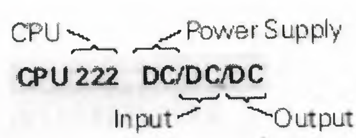



**Figure 3.2** S7-200 CPU Description

### 3.2.1 Mode Switch and Analog Adjustment

When the mode switch is in the RUN position the CPU is in the run model and executing the program. When the mode switch is in the STOP position the CPU is stopped. When the mode switch is in the TERM position the programming device can select the operating mode (Figure 3.3). The analog adjustment is used to increase or decrease values stored in special memory. These values can be used to update the value of a timer or counter, or can be used to set limits.

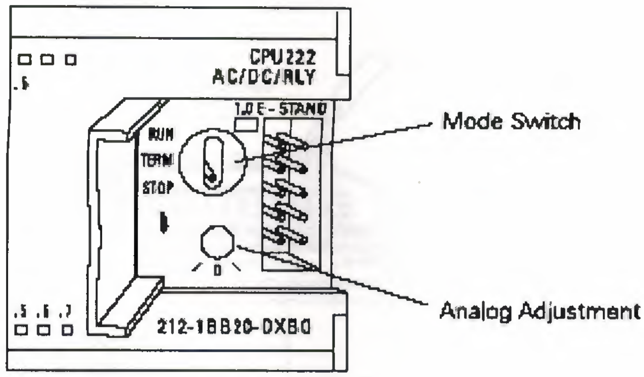



**Figure 3.3** Mode Switch and Analog Adjustment

### 3.2.2 Inputs

Input devices, such as switches, pushbuttons, and other sensor devices are connected to the terminal strip under the bottom cover of the PLC (See Figure 3.4).

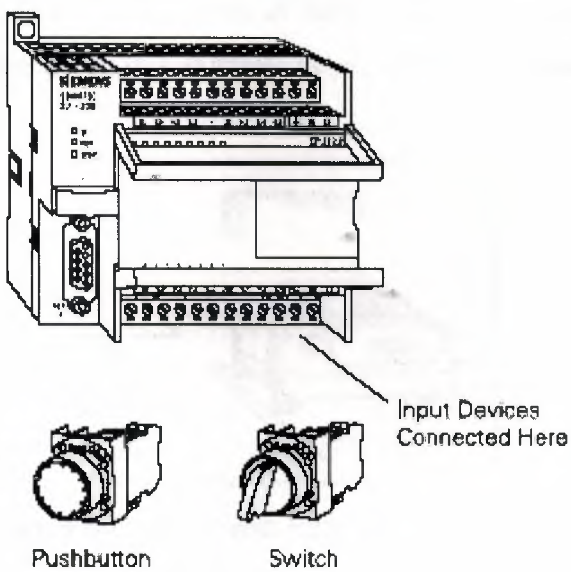



**Figure 3.4** Siemens S7-200 Inputs

### 3.2.3 Outputs

Output devices, such as relays, are connected to the terminal strip under the top cover of the PLC. When testing a program, it is not necessary to connect output devices. The LED status indicators signal if an output is active.

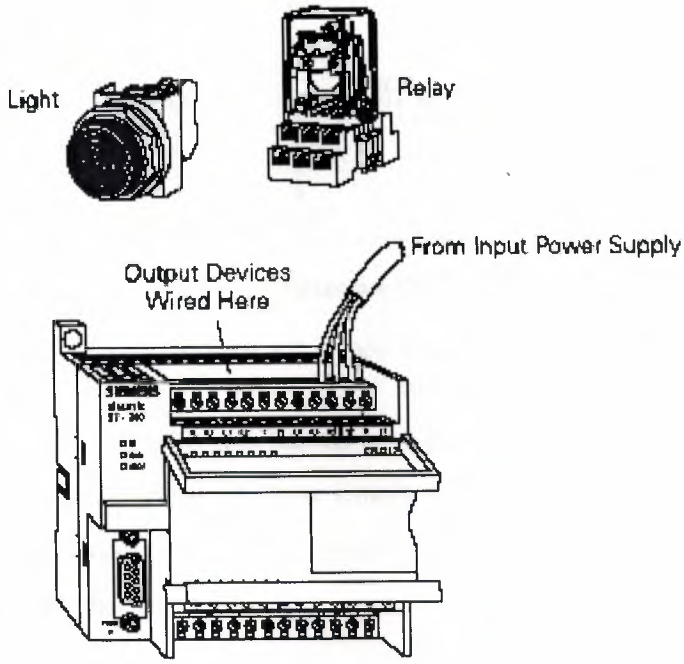



**Figure 3.5** Siemens S7-200 Outputs

## 3.3 Timers

Timers are devices that count increments of time. Traffic lights are one example where timers are used (Figure 3.6). In this example timers are used to control the length of time between signal changes.

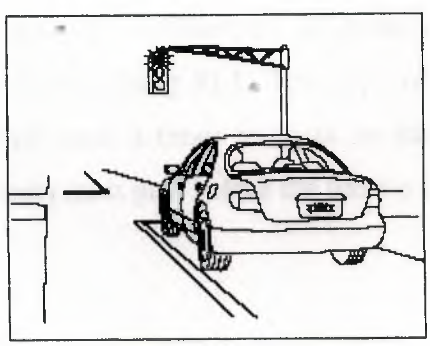


**Figure 3.6** Traffic Lights Example by Using Timers

Timers are represented by boxes in ladder logic. When a timer receives an enable, the timer starts to time. The timer compares its current time with the preset time. The output of the timer is a logic 0 as long as the current time is less than the preset time. When the current time is greater than the preset time the timer output is a logic 1. S7-200 uses three types of timers: On-Delay (TON), Retentive On-Delay (TONR), and Off-Delay (TOF) (Figure 3.7).

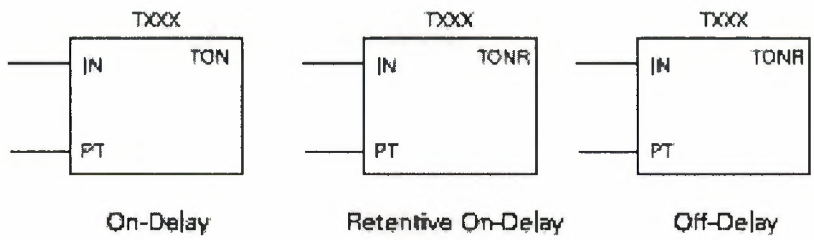



**Figure 3.7** Types of Timers

S7-200 timers are provided with resolutions of 1 millisecond, 10 milliseconds, and 100 milliseconds. The maximum value of these timers is 32.767 seconds, 327.67 seconds, and 3276.7 seconds, respectively. By adding program elements, logic can be programmed for much greater time intervals.

### 3.3.1 Hard-wired timing circuit

Timers used with PLCs can be compared to timing circuits used in hard-wired control line diagrams. In the Figure 3.8, a normally open (NO) switch (S1) is used with a timer (TR1). For this example the timer has been set for 5 seconds. When S1 is closed, TR1 begins timing. When 5 seconds have elapsed, TR1 will close its associated normally open TR1 contacts, illuminating pilot light PL1. When S1 is open, deenergizing TR1, the TR1 contacts open, immediately extinguishing PL1. This type of timer is referred to as ON delay. ON delay indicates that once a timer receives an enable signal, a predetermined amount of time (set by the timer) must pass before the timer.s contacts change state.

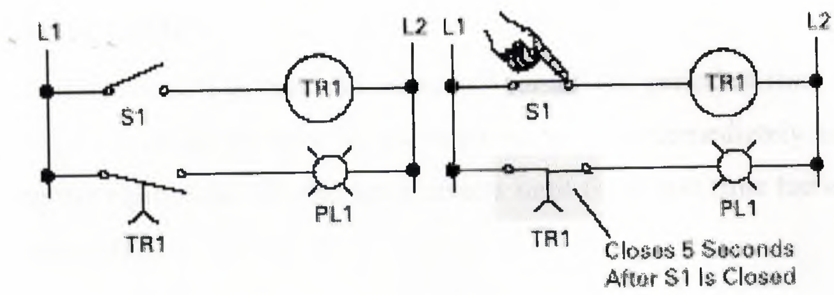



**Figure 3.8** Hard-Wired Timing Circuit

### 3.3.2 On-Delay (TON)

When the On-Delay timer (TON) receives an enable (logic 1) at its input (IN), a predetermined amount of time (preset time - PT) passes before the timer bit (T-bit) turns on. The T-bit is a logic function internal to the timer and is not shown on the symbol. The timer resets to the starting time when the enabling input goes to a logic 0 (Figure 3.9).

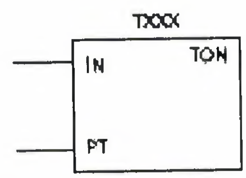



**Figure 3.9** On-Delay Timer

### 3.3.3 Retentive On-Delay (TONR)

The Retentive On-Delay timer (TONR) functions in a similar manner to the On-Delay timer (TON). There is one difference. The Retentive On-Delay timer times as long as the enabling input is on, but does not reset when the input goes off (Figure 3.10). The timer must be reset with a RESET (R) instruction.

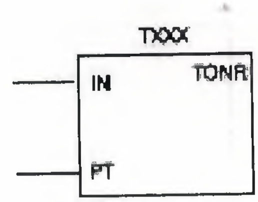



**Figure 3.10** Retentive On-Delay (TONR) Timer

### 3.3.4 Off-Delay (TOF)

The Off-Delay timer is used to delay an output off for a fixed period of time after the input turns off. When the enabling bit turns on the timer bit turns on immediately and the value is set to 0. When the input turns off, the timer counts until the preset time has elapsed before the timer bit turns off.

## 3.4 Counters

Counters used in PLCs serve the same function as mechanical counters. Counters compare an accumulated value to a preset value to control circuit functions. Control applications that commonly use counters include the following:

. Count to a preset value and cause an event to occur

. Cause an event to occur until the count reaches a preset value

A bottling machine, for example, may use a counter to count Bottles into groups of six for packaging (Figure 3.11).

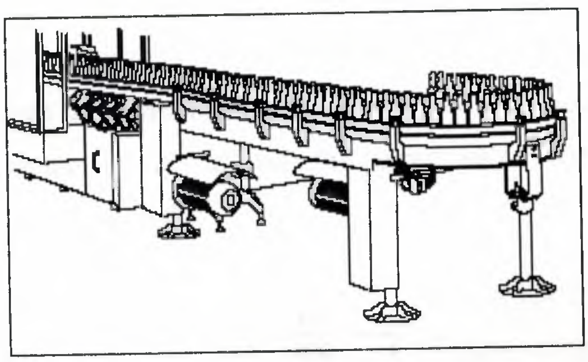


**Figure 3.11** A Bottling Machine Example by Using Counter

There are three types of counters used in Siemens PLCs (Figure 3.12).

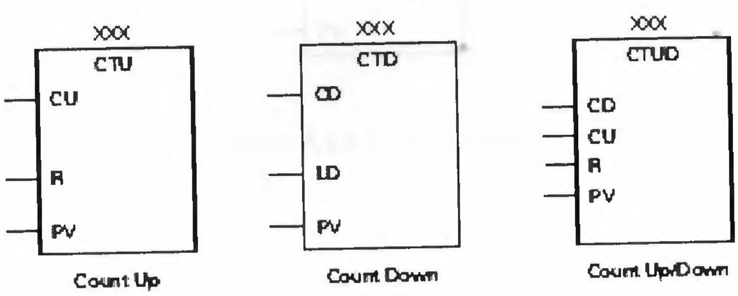



**Figure 3.12** Types of counters

### 3.4.1 Up Counter

The up counter counts up from a current value to a preset value (PV). Input CU is the count input. Each time CU transitions from a logic 0 to a logic 1 the counter increments by a count of 1. Input R is the reset. A preset count value is stored in PV input. If the current count is equal to or greater than the preset value stored in PV, the output bit (Q) turns on (not shown).

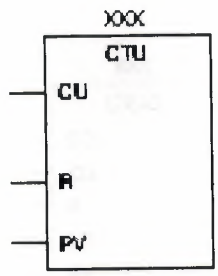



**Figure 3.13** Up Counter

### 3.4.2 Down Counter

The down counter counts down from the preset value (PV) each time CD transitions from a logic 0 to a logic 1. When the current value is equal to zero the counter output bit (Q) turns on (not shown). The counter resets and loads the current value with the preset value (PV) when the load input (LD) is enabled.

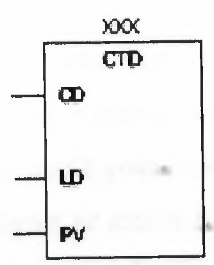



**Figure 3.14** Down Counter

### 3.4.3 Up/Down Counter

The up/down counter counts up or down from the preset value each time either CD or CU transitions from a logic 0 to a logic 1. When the current value is equal to the preset value, the output QU turns on. When the current value (CV) is equal to zero, the output QD turns on. The counter loads the current value (CV) with the preset value (PV) when the load input (LD) is enabled. Similarly, the counter resets and loads the current value (CV) with zero when the reset (R) is enabled. The counter stops counting when it reaches preset or zero.

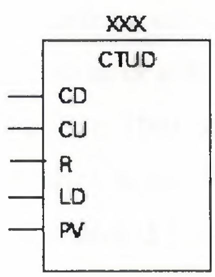



**Figure 3.15** Up-Down Counter

## 3.5 Equipments used in the Project

### 3.5.1 DC Motor

Many people are familar with DC motors. If you have ever had a electric toy train or car as a child, you may know how a DC motor works. If you were like me you probably took one apart and couldn't put it back together. Usually the brush springs get lost. The stator is the stationary outside part of a motor. The rotor is the inner part which rotates. In the motor animations, red represents a magnet or winding with a north polarization, while green represents a magnet or winding with a south polariztion. Opposite, red and green, polarities attract. The stator of a permanent magnet dc motor is composed of two or more permanent magnet pole peices. The rotor is composed of windings which are connected to a mechanical commutator. In this case the rotor has three pole pairs. The opposite polarities of the energized winding and the stator magnet attract and the rotor will rotate until it is aligned with the stator. Just as the rotor reaches alignment, the brushes move across the commutator contacts and energize the next winding. In the animation the commutator contacts are brown and the brushes are dark grey. A yellow spark shows when the brushes switch to the next winding. Notice that the comutator is staggered from the rotor poles. If the connections of a dc motor are reversed the motor will change directions. Though it will not always work as well in both directions. This is a permanent magnet dc motor. Two other types of dc motors are series wound and shunt wound dc motors. These motors also use a similar rotor with brushes and a commutator. However, the stator uses windings instead of permanent magnets. The basic principle is still the same. A series wound dc motor has the stator windings in series with the rotor. A shunt wound dc motor has the stator windings in parallel with the rotor winding. The series wound motor is more common. A series wound motor is also called a universal motor. It is universal in the sense that it will run equally well using either an ac or a dc voltage source. Reversing the polarity of both the stator and the rotor cancel out. Thus the motor will always rotate the same direction irregardless of the voltage polarity. A universal motor is in a sense an ac motor in that it will operate, from an ac power source. I prefer the term universal motor to avoid confusion with ac induction motors.

**Figure 3.16** DC motor

Brushless DC motors are refered to by many aliases: brushless permanent magnet, permanent magnet ac motors, permanent magnet synchronous motors ect. The confusion arises because a brushless dc motor does not directly operate off a dc voltage source. However, the basic principle of operation is similar to a dc motor. A brushless dc motor has a rotor with permanent magnets and a stator with windings. It is essentially a dc motor turned inside out. The brushes and commutator have been eliminated and the windings are connected to the control electronics. The control electronics replace the function of the commutator and energize the proper winding.

As shown in the animation the winding are energized in a pattern which rotates around the stator. The energized stator winding leads the rotor magnet, and switches just as the rotor aligns with the stator. There are no sparks, which is one advantage of the bldc motor. The brushes of a dc motor have several limitations; brush life, brush residue, maximum speed, and electrical noise. BLDC motors are potentially cleaner, faster, more efficient, less noisy and more reliable. However, BLDC motors require electronic control.

**Figure 3.17** Brushless DC Motors

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 tradespeople was greatly reduced.

### 3.5.2 Relays

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 Figure 3.18. 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 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).

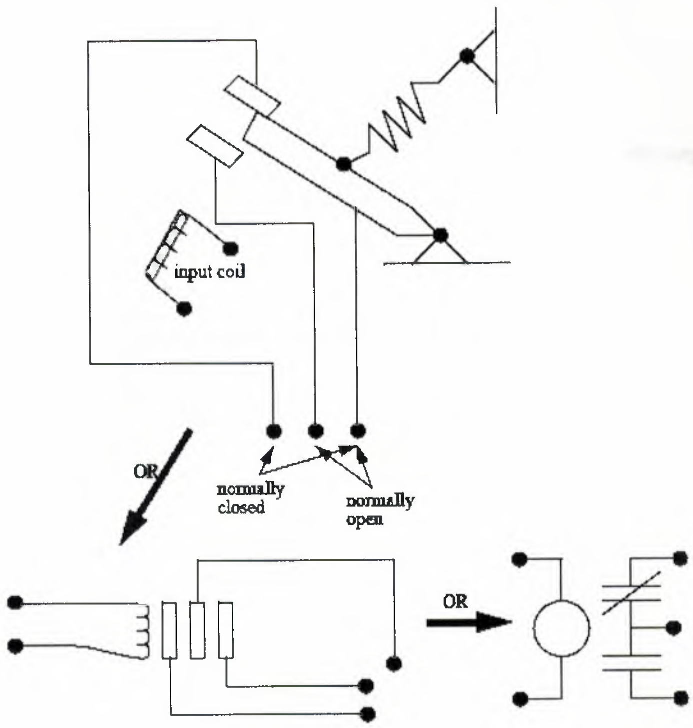



**Figure 3.18** Simple Relay Layouts and Schematics

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 3.19. 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 the input B. If current is flowing through the first two relays 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 off and B is on.

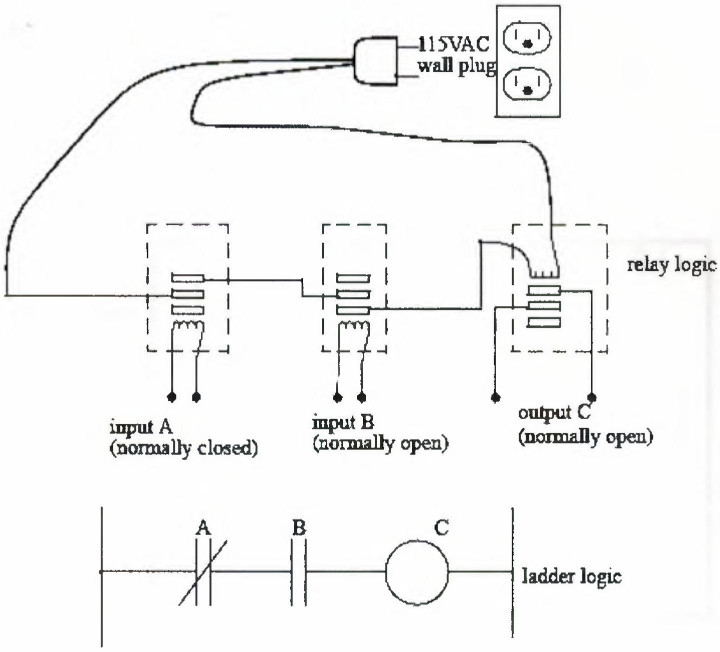



**Figure 3.19** A Simple Relay Controller

The example in Figure 3.19 does not show the entire control system, but only the logic. When we consider a PLC there are inputs, outputs, and the logic. Figure 3.20 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.

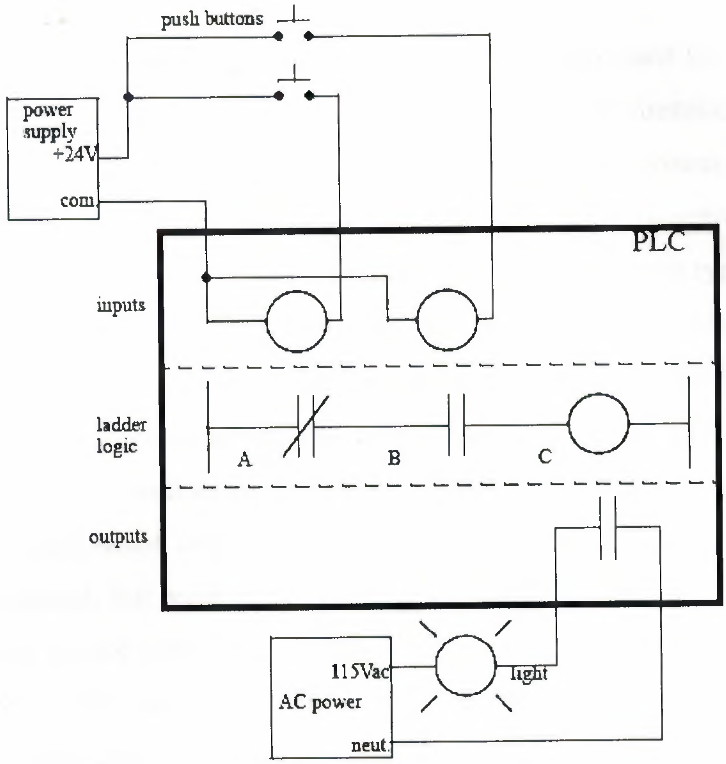



**Figure 3.20** A PLC Illustrated With Relays

Many relays also have multiple outputs (throws) and this allows an output relay to also be an input simultaneously. The circuit shown in Figure 3.21 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 labelled A or 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.

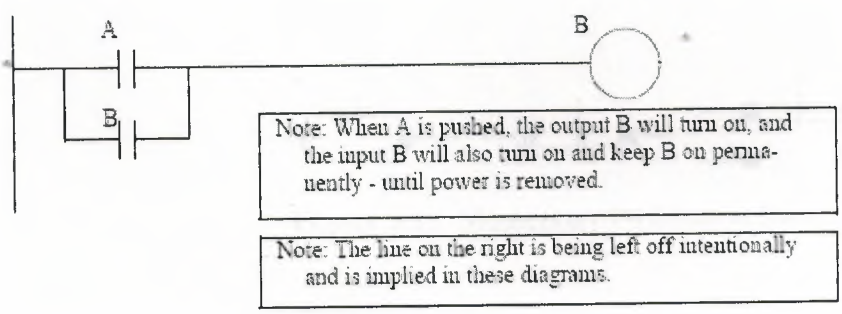



**Figure 3.21** A Seal-in Circuit

### 3.5.3 Pushbutton

There are two fundamental uses for switches. First, switches are used for operator input to send instructions to the control circuit. Second, switches may be installed on the moving parts of a machine to provide automatic feedback to the control system. There are many different types of switches, too many to cover in this text. However, with a basic understanding of switches, it is easy to understand most of the different types.

The most common switch is the pushbutton. It is also the one that needs the least description because it is widely used in automotive and electronic equipment applications.

There are two types of pushbutton, the momentary and maintained. The momentary pushbutton switch is activated when the button is pressed, and deactivated when the button is released. The deactivation is done using an internal spring. The maintained pushbutton activates when pressed, but remains activated when it is released. Then to deactivate it, it must be pressed a second time. For this reason, this type of switch is sometimes called a push-push switch. The on/off switches on most desktop computers and laborator oscilloscopes are maintained pushbuttons.

The contacts on switches can be of two types. These are normally open (N/O) and normally closed (N/C). Whenever a switch is in it's deactivated position, the N/O contacts will be open (non-conducting) and the N/C contacts will be closed (conducting). Figure 1-4 shows the schematic symbols for a normally open pushbutton (left) and a normally closed pushbutton (center). The symbol on the right of Figure 3.22 is a single pushbutton with both N/O and N/C contacts. There is no internal electrical connection between different contact pairs on the same switch. Most industrial switches can have extra contacts "piggy backed" on the switch, so as many contacts as needed of either type can be added by the designer.

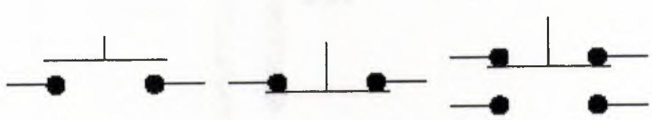


**Figure 3.22** Momentary Pushbutton Switches

Light emitting diodes must be chosen according to how they will be used, because there are various kinds. The diodes are available in several colors. The most common colors are red and green, but there are even blue ones. The device on the far right in the photograph combines a red LED and green LED in one package. The component lead in the middle is common to both LEDs. As for the remaining two leads, one side is for the green, the other for the red LED. When both are turned on simultaneously, it becomes orange. When an LED is new out of the package, the polarity of the device can be determined by looking at the leads. The longer lead is the Anode side, and the short one is the Cathode side. The polarity of an LED can also be determined using a resistance meter, or even a 1.5 V battery. When using a test meter to determine polarity, set the meter to a low resistance measurement range. Connect the probes of the meter to the LED. If the polarity is correct, the LED will glow. If the LED does not glow, switch the meter probes to the opposite leads on the LED. In either case, the side of the diode which is connected to the black meter probe when the LED glows, is the Anode side. Positive voltage flows out of the black probe when the meter is set to measure resistance It is possible to use an LED to obtain a fixed voltage. The voltage drop (forward voltage or VF) of an LED is comparatively stable at just about 2V.

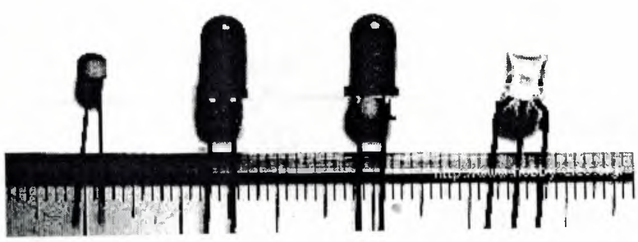


**Figure 3.25** Types of LED`s

This notation of having two small arrows pointing away from the device is common to the schematic symbols of all light-emitting semiconductor devices. Conversely, if a device is light-activated (meaning that incoming light stimulates it), then the symbol will have two small arrows pointing toward it

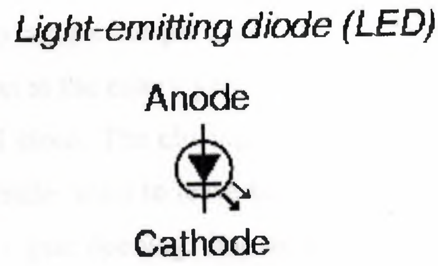



**Figure 3.26** LED symbol

. It is interesting to note, though, that LEDs are capable of acting as light-sensing devices: they will generate a small voltage when exposed to light, much like a solar cell on a small scale. This property can be gainfully applied in a variety of light-sensing circuits. Because LEDs are made of different chemical substances than normal rectifying diodes, their forward voltage drops will be different. Typically, LEDs have much larger forward voltage drops than rectifying diodes, anywhere from about 1.6 volts to over 3 volts, depending on the color. Typical operating current for a standard-sized LED is around 20 mA. When operating an LED from a DC voltage source greater than the LED's forward voltage, a series-connected "dropping" resistor must be included to prevent full source voltage from damaging the LED. Consider this example circuit:

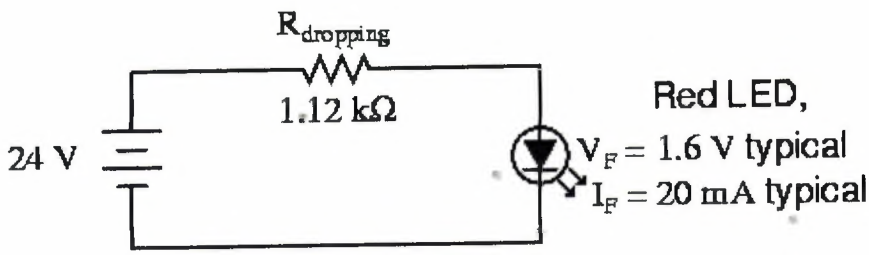



**Figure 3.27** Demonstration of LED in a circuit

### 3.6 Car Park Area Application

The purpose of project is to control car park area system. When any vehicle want to get in the car park, the push button at the entery gate is pressed gate will open. The opening time is 5 sec after that gate will close. The closing time of the entry gate is also 5 sec. In the same manner when any vehicle want to leave the car park, exit push button is pressed that open the exit gate, the exit gate opening time is 5 sec after that exit gate will close the closing time of exit gate is also 5 sec.

### 3.6.1 Ladder Diagram

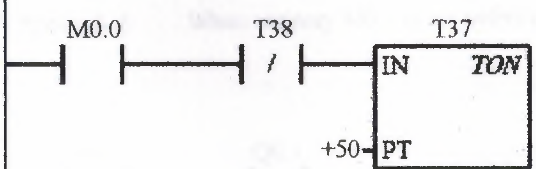
PROGRAM TITLE COMMENTS

Press F1 for help and example program

Network 1 When entery gate push button is pressed data is stored in memory

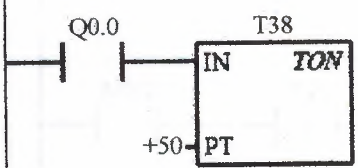
NETWORK COMMENTS


Network 2 When memory circuit is active and T38 is not active T37 start counting time.



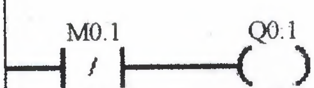

Network 3 When T37 is active ıt opens the entery gate.

Network 4 When Q0.0 is active T38 is started is counting.



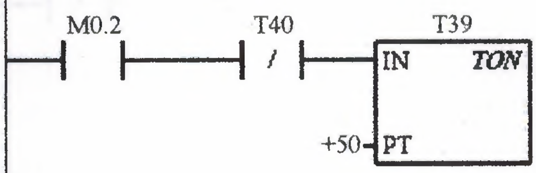

Network 5 When T38 is active it will active memory M0.1.


Network 6 When memory M0.1 is not active entery gate will close (Q0.1).



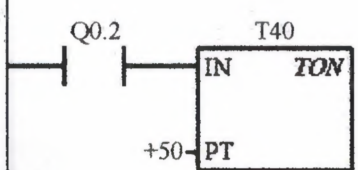

Network 7 When exit gate push button is pressed M0.2 will activate.

Network 8 When memory (M0.2) is active and T40 is not active T39 will start counting time.




Network 9 When T39 is active exit will open.

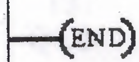

Network 10 When exit gate is open T40 will start counting time.




Network 11 When T40 is active it will activate memory (M0.3).

Network 12 When M0.3 is not active exit gate will close.


Network 13 The program is completed.



### 3.6.2 Statement List

**NETWORK 1** When entery gate push button is pressed data is stored in memory.

```
LD   I0.0
=    M0.0
```

**NETWORK 2** When memory circuit is active and T38 is not active T37 start counting time.

```
LD   M0.0
AN   T38
TON  T37, +50
```

**NETWORK 3** When T37 is active ıt opens the entery gate.

```
LD   T37
=    Q0.0
```

**NETWORK 4** When Q0.0 is active T38 is started is counting.

```
LD   Q0.0
TON  T38, +50
```

**NETWORK 5** When T38 is active it will active memory M0.1.

```
LD   T38
=    M0.1
```

**NETWORK 6** When memory M0.1 is not active entery gate will close (Q0.1).

```
LDN  M0.1
=    Q0.1
```

**NETWORK 7** When exit gate push button is pressed M0.2 will activate.

```
LD   I0.1
=    M0.2
```

**NETWORK 8** When memory (M0.2) is active and T40 is not active T39 will start counting time.

```
LD   M0.2
AN   T40
TON  T39, +50
```

**NETWORK 9** When T39 is active exit will open.

```
LD    T39
=     Q0.2
```

**NETWORK 10** When exit gate is open T40 will start counting time.

```
LD    Q0.2
TON   T40, +50
```

**NETWORK 11** When T40 is active it will activate memory (M0.3).

```
LD    T40
=     M0.3
```

**NETWORK 12** When M0.3 is not active exit gate will close.

```
LDN   M0.3
=     Q0.3
```

**NETWORK 13** The program is completed.

```
MEND
```

# CONCLUSION

In this project task and aim was to emphasis the idea of vitality of PLC in modern world, almost in all kinds of industrial and usual life applications PLC became an important component because PLC made the process and operation so easy and very convenience to control more over easily alterable according to desired operation. In olden days all of industrial processes were controlled manually and if any processes were desired to alter according to modern fashion and demand, to alter the system was very expensive and tedious work therefore PLC invention solved all these problems.

In this project, it is demonstrated how a car park that has specific limit of cars to be park inside the car park could be controlled by PLC. A ladder program that was written in order to achieve desired result from PLC was downloaded on PLC. PLC was controlling the car park system using counters and timers, timers were to control the opening and closing time of the entering and leaving gates and counters were to count cars entering and leaving the car park. Therefore PLC made this control system easy to be controlled if it is desired to change the limit of cars according to capacity available or entering and leaving gate time is required to change.

Even though the car park system was simple, it could be modified in sake of safety and easily understandable whether there is a room for car (or cars) in there or not, if there is any room available then what is location of that empty room, this could be done by using electronic indication board placed outside the entering of the car park moreover sensors circuitry could be use with each gate (entering and leaving) in order to achieve better efficiency.

The whole project was subjected to explain widely importance of PLC and facilities of PLC while using in any kind of practical application.

## REFERENCES

[1] **Özgür Cemal Özerdem**, *"Programmable Logic Controller and Programming"*, *Near East University Press, 2002.*

[2] **Beecher H.W.**, *"Introduction to PLC Programming and Implementation from Relay Logic to PLC Logic", 2nd ed., USA, 1997.*

[3] **Bryan L.A., Bryan E.A.**, *"Programmable Controller Theory and Implementation", 2nd ed., USA, 1997.*

[4] **Grob, Bernard**, *'Basic Electronics", McGraw Hill, Seventh Edition, 1993.*

[5] **Information about Siemens S-7 200 Micro PLC**: *"http://www.sea.siemens.com"*

[6] **Hackwarth J.R., Hackwarth F.D.**, *"PLC Programming Method and Applications", 3rd ed., 2000.*

[7] **Information about equipments:** *"http://www.plcs.net/chapters/relay7.htm"*