NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Electrical and Electronic Engineering

GATE CONTROLLING BY USING PLC

Graduation Project
EE- 400

Student: SEMIH AY (980375)

Supervisor: Mr. Özgür Cemal Özerdem

Lefkoşa -2003

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 am indebted to my supervisor, Mr. Özgür Cemal Özerdem, for his valuable guidance and encouragement throughout the entire Graduation Project. Whenever I have problem in my project, he shows his enthusiasm to solve problems. He has given his best support for me to conduct and enjoy my Graduation Project work.

I would like to thank Mr. Cemal Kavalcıoğlu, I could not have prepared this Graduation Project without the his generous help. He has shared their knowledge and experience, which are constructive to my Graduation Project. Valuable suggestion to assist my Graduation Project has been provided by him.

Last but not least, I would like to express my gratitude to Prof. Dr Fakhreddin Mamedov because he helped to me at each stage of my Undergraduate Education in Near East University.

Finally, I would like to thank my family, especially my parents for being patientful during my Undergraduate degree study. I could never have completed my study without their encouragement and endless support.

ABSTRACT

As PLC technology has advanced, so have programming languages and communications capabilities, along with many other important features. Today's PLCs offer faster scan times, space efficient high-density input/output systems, and special interfaces to allow non-traditional devices to be attached directly to the PLC. Not only can they communicate with other control systems, they can also perform reporting functions and diagnose their own failures, as well as the failure of a machine or process. Size is typically used to categorize today's PLC, and is often an indication of the features and types of applications it will accommodate.

In this project FATEK FALCON FB SERIES PLC have 16 inputs, 12 outputs, 24 volt DC input 220 Volt AC output PLC is used, and an automation of a gate and a door are realized by PLC programs.

Real life experimental operation is performed and objectives are achieved, physically operated.

TABLE OF CONTENTS

INTRODUCTION

A Programmable Logic Controller was defined by Capiel (1982) as: " A digitally operating electronic system designed for use in an industrial environment, which uses a programmable memory for the internal storage of instructions for implementing specific functions such as logic, sequencing, timing, counting and arithmetic to control through analog or digital input/output modules, various types of machines or processes." Which explains the device perfectly.

In the late 1960's PLC's were first introduced. The first PLC can be traced back to 1968 when Bedford Associates, a company in Bedford, MA, developed a device called a Modular Digital Controller for General Motors (GM). The MODICON, as it was known, was developed to help GM eliminate traditional relay-based machine control systems.

The aim of this project is the control of a gate and a door with FATEK FALCON FB SERIES PLC.

The project consists of the five chapters, and conclusion.

Chapter-1 presents PLC layout, Ladder logic, Types of PLC's, Comparison with other Control Systems, Advantages of PLC's, and chapter concluded with a brief information about Today's PLC's.

Chapter-2 presents designs, structure and operation, Memory, CPU, Programming consoles and Power suppliers of PLC.

Chapter-3 presents Logic instructions and graphic programming, Memory circuits, Facilities and Arithmetic instructions of PLC.

Chapter-4 presents the informations about Siemens SIMATIC S7-200 PLCs and application areas of this type of PLCs.

Chapter-5 presents a practical implementation of a gate and a door with PLC program.

1. INTRODUCTION

1.1 Basic of Programmable logic controllers

The need for low cost, versatile and easily commissioned controllers has resulted in the development of programmable-control systems standard units based on hardware CPU and memory for the control of machines or processes. Originally designed as a replacement for the hard-wired relay and timer logic to be found in traditional control panels, PLC's provides ease and flexibility of control based on programming and executing simple logic instructions. PLC's have internal functions such as timers, counters and shift registers, making sophisticated control possible using even the smallest PLC.

An appropriate definition of a programmable logic controller (PLC) is that it is a 'digital electronic device that use a programmable memory to store instructions and to implement specific functions such as logic, sequence, timing, counting and arithmetic to control machines and processes'.

Figure 1.1 below shows how the control action is achieved. Input devices and output devices from the machine or process to be controlled are connected to the PLC. A user enters a sequence of instructions in to PLC program memory. The controller then continuously monitors the state of the inputs and switches outputs according to the users program. Because of the stored program can be modified or changed completely, this result a flexible system, which can be used for control tasks that vary in nature and complexity.

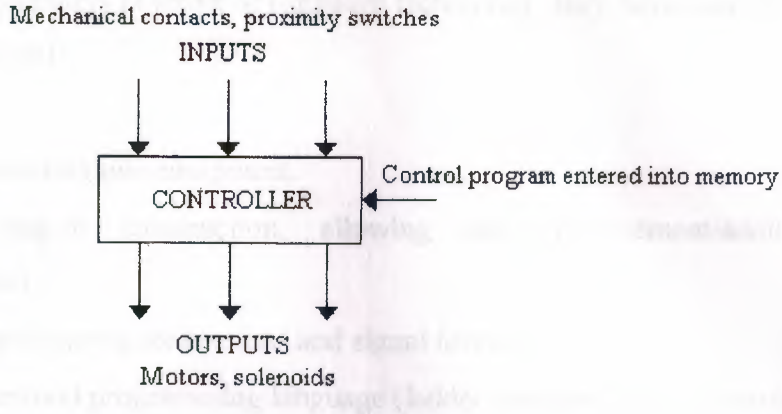


Figure 1.1 The control action of a PLC

A programmable control operates by examining the input signals from a process and carrying out logic instructions on these input signals, producing output signal to drive process equipment or machinery. Standard interfaces build into PLC's allow them to be directly connected to process actuators and transducers (pumps and valves) without the need for intermediate circuitry or relays.

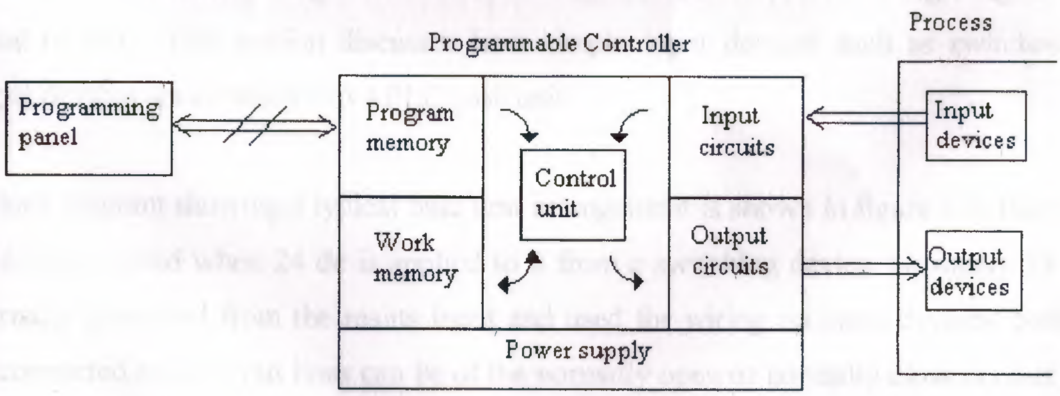


Figure 1.2 Programmable controller structure

Through using PLC's it became possible to modify a control system without having the disconnect or re-route a signal wire. It was necessary to change only the control program using a keypad or VDU terminal. Programmable controllers also require shorter installation and commissioning times than do hardwired systems. Although PLC's are similar to conventional computers in terms of hardware technology, they have specific features suited to industrial control:

- Rugged, noise immune equipment.
- Modular plug-in construction, allowing easy replacement/addition of units (input/output).
- Standard input/output connections and signal levels.
- Easily understood programming language (ladder diagram and function chart).
- Ease of programming and reprogramming in-plant.
- The interfacing for input and output devices is inside the controller.

These features make programmable controllers highly desirable in a wide variety of industrial-plant and process-control situations.

1.2 PLC Layout

As we know, a switch is logic element and so can be used to provide a logic signal input signal to PLC. This section discusses how simple input devices such as switches, and output devices are connected to a PLC base unit.

A block diagram showing a typical base unit arrangement is shown in figure 1.3. Each PLC input is energized when 24 dc is applied to it from a switching device. Normally 24 dc is internally generated from the mains input and used for wiring up input devices. Switches are connected to the input lines can be of the normally open or normally close contact type. When the run input energized, the outputs are switched according to the program and the condition of the inputs.

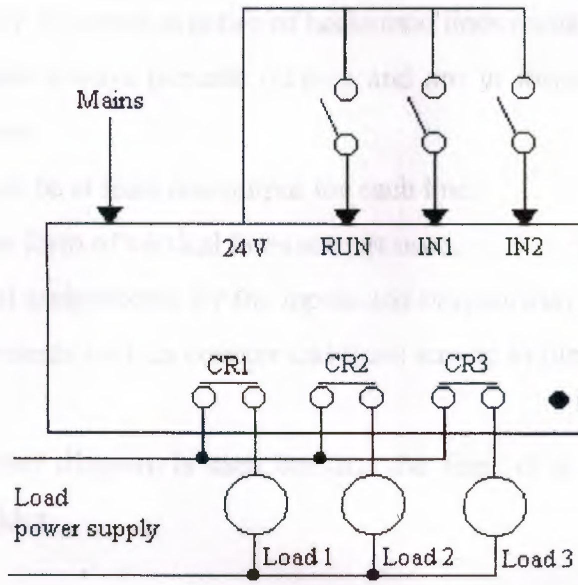


Figure 1.3 Base unit arrangement

The output loads can be switched from relay, transistor or triac contacts inside the PLC. Relays are widely used. Figure shows how loads such as widely used as solenoids, motors and heaters can be connected to relay contacts. This is fine provided that the maximum current rating for the relay contacts is not exceeded. For a heavy current load, a PLC output relay is used to drive a secondary switching device such as a solid-state relay or a contactor.

The input and output connection points on a PLC are allocated numbers so that they can be uniquely identified. Each manufacturer uses identification system, which depends on the number of input/output options.

1.3 Ladder Logic

With the majority of programmable logic controllers, writing a program is equivalent to drawing a switching circuit. The switching circuit is drawn in a ladder diagram format. This format requires that;

1. Circuits are arranged as series of horizontal lines containing inputs and outputs.
2. Inputs must always precede outputs and are in form of normally open and normally close forms.
3. There must be at least one output for each line.
4. Circuits in form of vertical lines are not used.
5. Numerical assignments for the inputs and outputs also shown in ladder diagram.
6. Other elements such as counter and timer can be implemented in ladder diagram.

The term ladder diagram is used because the lines of a completed diagram resemble the rungs of a ladder.

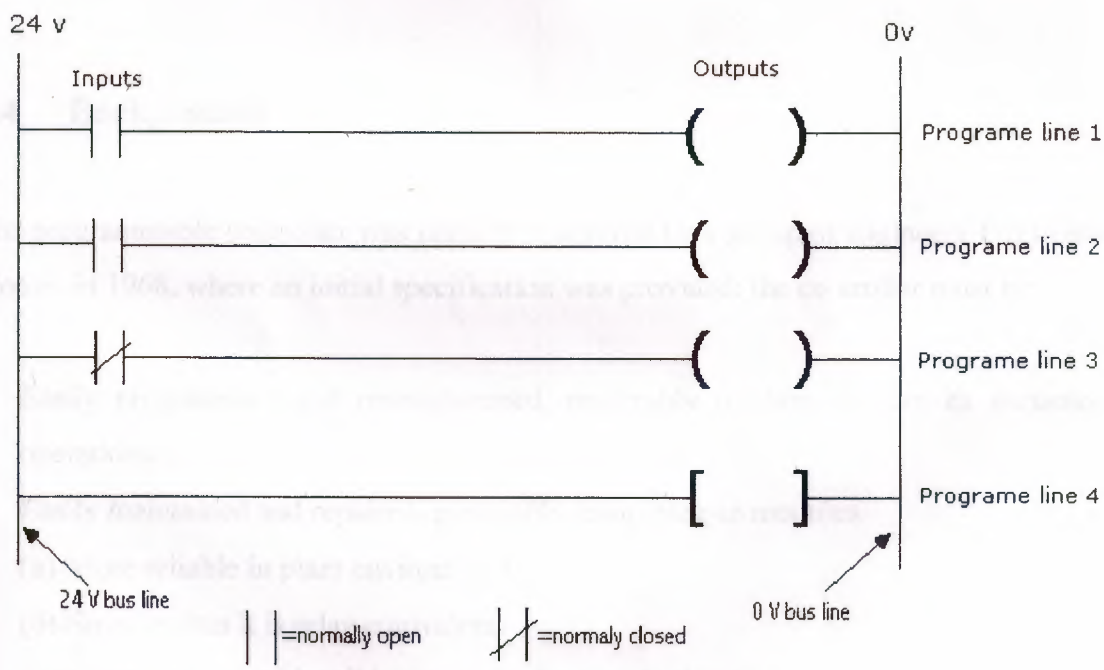


Figure 1.4 Ladder format

A ladder diagram can be transmitted to program bay using table 1.1.

Table 1.1 Ladder instructions

Instruction	Description
LOAD	Load logical state of start input
LOAD NOT	Load logical state of start input and inverter
AND	Logical AND operation
AND NOT	Logical AND NOT operation
OR	Logical OR operation
OR NOT	Logical OR NOT operation
OUT	Output

1.4 Background

The programmable controller was initially conceived by a group of engineers from General Motors in 1968, where an initial specification was provided: the controller must be:

- Easily programmed and reprogrammed, preferably in-plant to alter its sequence of operations.
- Easily maintained and repaired- preferably using plug-in modules.
- (a)-More reliable in plant environment.
  (b)-Smaller than it is relay equivalent.
- Cost competitive, with solid-state and relay panels than in use.

This provoked a keen interest from engineers of all disciplines in how to PLC could be used for industrial control. With this came demands for additional PLC capabilities and facilities, which were rapidly implemented as the technology became available. The instruction sets quickly moved from simple logic instructions to include counters, timers and shift registers, than onto more advanced mathematical functions on the machines.

Developments hardware were also occurring, with larger memory and greater numbers of input / output points featuring on new models.

Table 1.2 Chart of Programmable Controller Developments

Year	Nature of developments
1968	Programmable controller concept developed
1969	Hardware CPU controller, with logic instructions, 1K of memory and 128 I/O points
1974	Use several (multi) processor within a PLC timers and counters; arithmetic operations; 12K of memory and 10241 I/O points
1976	Remote input/output systems introduced
1977	Microprocessor-based PLC introduced
1980	Intelligent I/O modules developed Enhanced communication facilities Enhanced software features (e.g. documentation) Use of personal microcomputers as programming aids
1983	Low-cost small PLCs introduced
1983 on	Networking of all levels of PLC, computer, and machine under standard GM MAP specification. Distributed hierarchical control of industrial plants.

The increased rate of application of programmable controllers within industry has encouraged manufacturers to develop whole families of microprocessor-based systems having various levels of performance. The range of available PLC's now extends from small self-contained units with 20 digital I / O points and 500 program steps, up to modular systems with add-on function modules:

- Analogue I/O
- PID control (proportional, integral and derivative terms)
- Communications
- Graphics display
- Additional I/O
- Additional memory

This modular approach allows the expansion or upgrading of a control system with minimum cost and disturbance.

Programmable controllers are developing at a virtually the same pace as microcomputers, with particular emphasis on small controllers, positioning/numeric control and communication networks. The market for small controllers has grown rapidly since the early 1980's when a number of Japanese companies introduced very small, low cost units that were much cheaper than others available at that time. This brought programmable controllers within the budget of many potential users in the manufacturing and process industries, and this trend continues with PLC's offering ever-increasing performance at ever-decreasing cost.

1.5 Terminology PC-PLC

There are several different terms used to describe programmable controllers, most referring to the functional operation of the machine in question:

- PC programmable controller (UK Origin)
- PLC programmable logic controller (American Origin)
- PBS programmable binary system (Swedish Origin)

By their nature these terms tend to describe controllers that normally work in a binary environment. Since all but the smallest programmable controllers can now be equipped to process analogue inputs and outputs these labels are not representative of their capabilities.

For these reason the overall term programmable controller has been widely adopted to describe the family of freely programmable controllers. However, to avoid confusion with the personal computer PC, this text uses the abbreviation PLC for programmable (logic) controller.

1.6 Comparison with other Control Systems

Programmable controller emerge from the comparison as the best over all choice for control system, unless the ultimate in operating speed or resistance to electrical noise is required in which case hardwired digital logic and relays are chosen respectfully. For handling complex functions a conventional computer is still marginally superior to a large PLC equipped with relevant function cards, but only in terms of creating the functions, not using them here the PLC is more efficient through passing values to the special function module, which then handles the control function independently of the main processor, a multiprocessor system.

Programmable controllers have both hardware and software features that make them attractive as controller of a wide range of industrial equipment.

Table 1.3 provides a comparison between various control media. This only an approximate guide to their capabilities and further technical information can be obtained from the manufacturers data sheet on each specific systems.

Table 1.3 Comparison of Control System

Characteristic	Relay system	Digital logic	Computers	PLC system
Price per function	Fairly low	Low	High	Low
Physical size	Bulky	Very compact	Fairly compact	Very compact
Operating speed	Slow	Very fast	Fairly fast	Fast
Electrical noise	Excellent	Good	Quite good	Good
Installation	Time-consuming to design and install	Design time-consuming	Programming extremely time-consuming	Simple to program and install
Capable of complicated operations	No	Yes	Yes	Yes
Ease of changing function	Very difficult	Difficult	Quite Simple	Very simple
Ease of maintenance	Poor-large number of contacts	Poor if ICs soldered	Poor-several custom boards	Good-few standard cards

1.7 Types of PLC According to it's Build Mechanism

Types of PLC According to its' Build Mechanism.

1.7.1 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.

1.7.2 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.

Some examples: SIEMENS S5-115U, SIEMENS S7-200, MITSUBISHI PC40, TEXAS INSTRUMENT PLC's, KLOCKNER-MOELLER PS316, OMRON C200H.

1.8 Types of PLC According to it's Features

The increasing demand from industry for programmable controllers that can be applied to different forms and sizes of control tasks has resulted in most manufacturers producing a range of PLC's with various levels of performance and facilities.

Typical rough definitions of PLC size are given in terms of program memory size and the maximum number of input/output points the system can support. Table 1.4 gives an example of these categories.

Table 1.4 Categories of PLC

PLC size	Max I/O points	Use memory size
Small	40/40	1K
Medium	128/128	4K
Large	>128/>128	>4K

However, to evaluate properly any programmable controller we must consider many additional features such as its processor, cycle time language facilities, functions, and expansion capabilities.

A brief outline of the characteristics of small, medium of large programmable controller is given below, together with typical applications.

1.8.1 Small Sized PLCs

In general, small and 'mini' PLC's are designed as robust, compact units, which can be mounted on or beside the equipment to be controlled. They are mainly used the replaced hard-wired logic relays, timers, counters. That control individual items of plant or machinery, but can also be used to coordinate several machines working in conjunction with each other.

Small programmable controllers can normally have their total I/ O expanded by adding one or two I/ O modules, but if any further developments are required this will often mean replacement of the complete unit. This end of the market is very much concerned with non-specialist and users, therefore ease of programming and a 'familiar' circuit format are desirable. Competition between manufacturers is extremely fierce in this field, as they vie to obtain a maximum share in this partially developed sector of the market.

A single processor is normally used, and programming facilities are kept a fairly basic level, including conventional sequencing controls and simple standard functions: e.g. timers and counters. Programming of small PLC's is by way of logic instruction list (mnemonics) or relay ladder diagrams.

Program storage is given by EPROM or battery-backed RAM. There is now a trend towards EEPROM memory with on-board programming facilities on several controllers.

1.8.2 Medium Sized PLCs

In this range modular construction predominates with plug-in modules based around the Eurocard 19 inch rack format or another rack mounting system. This construction allows the simple upgrading or expansion of the system. This construction allows the simple upgrading or expansion of the system by fitting additional I/ O cards in to the cards into the rack, since most rack, systems have space for several extra function cards. Boards are usually 'ruggedized' to allow reliable operation over a range of environments.

In general this type of PLC is applied to logic control tasks that can not be met by small controllers due to insufficient I/O provision, or because the control task is likely to be extended in the future. This might require the replacement of a small PLC, where as a modular system can be expanded to a much greater extent, allowing for growth. A medium-sized PLC may therefore be financially more attractive in the long term.

Communications of a single and multi-bit processor are likely within the CPU. For programming, standard instructions or ladder and logic diagrams are available. Programming is normally carried out via a small keypad or a VDU terminal. If different sizes of PLC are purchased from a single manufacturer, it is likely that programs and programming panels will be compatible between the machines.

1.8.3 Large Sized PLCs

Where control of very large numbers of input and output points is necessary and complex control functions are required, a large programmable controller is the obvious choice. Large PLC's are designed for use in large plants or on large machines requiring continuous control. They are also employed as supervisory controllers to monitor and control several other PLC's or intelligent machines. e.g. CNC tools.

Modular construction in Eurocard format is standard, with a wide range of function cards available including analogue input output modules. There is a move towards 16-bit processor, and also multi-processor usage in order to efficiently handle a large range of differing control tasks.

For example;

- 16-bit processor as main processor for digital arithmetic and text handling.
- Single-bit processor as co-or parallel processor for fast counting, storage etc.
- Peripheral processor for handling additional tasks which are time-dependent or time-critical, such as:

  Closed-loop (PID) control
  Position controls
  Floating-point numerical calculations
  Diagnostic and monitoring
  Communications for decentralized
  Remote input/output racks.

This multi-processor solution optimizes the performance of the overall system as regards versatility and processing speed, allowing to PLC to handle very large programs of 100 K instructions or more. Memory cards can now provide several megabytes of CMOS RAM or EPROM storage.

1.9 Advantages

1.9.1 Accuracy

In relay control systems logical knowledge's carries in electro-mechanical contactors, they can lose data because of mechanical errors. But PLC's are microprocessor-based system so logical data are carried inside the processor, so that PLC's are more accurate than relay type of controllers.

1.9.2 Data Areas

Data memory contains variable memory, and register, and output image register, internal memory bits, and special memory bits. This memory is accessed by a byte bit convention. For example to access bit 3 of variable memory byte 25 you would the address V25.3.

Table 1.5 shows the identifiers and ranges for each of the data area memory types:

Table 1.5 Identifiers and ranges for data area memory types

Area Identifier	Data Area	CPU 212	CPU 214
I	Input	I0.0 to I7.7	I0.0 to I7.7
Q	Output	Q0.0 to Q7.7	Q0.0 to Q7.7
M	Internal memory	M0.0 to M15.7	M0.0 to M31.7
SM	Special Memory	SM0.0 to SM45.7	SM0.0 to SM85.7
V	Variable Memory	V0.0 to V1023.7	V0.0 to V4095.7

1.9.3 Logic Control of Industrial Automation

Everyday examples of these systems are machines like dishwashers, clothes washers and dryers, and elevators. In these systems, the output tend to be 220 V AC power signals to motors, solenoids, and indicator lights, and the inputs are DC or AC signals from user interface switches, motion limit switches, binary liquid level sensor, etc. Another major function in these types of controllers is timing.

1.9.4 Data Object

The S7-200 has six kinds of devices with associated data: timers, counters, analogue inputs, analogue outputs, accumulators and high-speed counters. Each device has associated data. For example, the S7-200 has counters devices. Counters have a data value that maintains the current count value. There is an also a bit value, which is set when the current value is greater than or equal to the present value. Since there are multiple devices are numbered from 0 to n. The corresponding data objects and object bits are also numbered.

Table 1.6 shows the identifiers and ranges for each of the data object memory types:

Table 1.6 Identifiers and ranges for data object memory types

Area Identifier	Data Area	CPU 212	CPU 214
T	Timers	T0 to T63	T0 to T127
C	Counters	C0 to C63	C0 to C127
AI	Analogue Input	AIW0 to AIW0	AIW0 to AIW30
AQ	Analogue Output	AQW0 to AQW30	AQW0 to AQW30
AC	Accumulator	AC0 to AC3	AC0 to AC3
HC	High-speed Counter	HC0	HC0 to HC2

1.9.5 Flexibility

When the control needs a change, relay type of controllers modification are hard, in PLC, this chance can be made by PLC programmer equipment.

1.9.6 Communication

PLC's are computer-based systems. That's why, they can transfer their data to another PC, or they can take external inputs from another PC. With this specification we can control the system with our PC. With relays controlled system it's not possible.

2. DESIGNS, STRUCTURE AND OPERATION

2.1 Basic Structure and operation of PLC

A block diagram of the internal structure of a PLC is shown in figure 2.1. The blocks consist of a central processing unit (CPU), a main memory and a buffer consisting of image memory and connection circuitry for digital input/output devices. A communication bus (i.e. a group of parallel wires used for transmitting digital signals) forms a common link to allow each element to share information.

The input image memory is used to hold the ON/OFF states of individual input ports. We use the binary system to represent the ON/OFF states because it is based on two digits (land 0) in image memory and ON state is stored as a binary 1 and on off state is stored as binary 0.

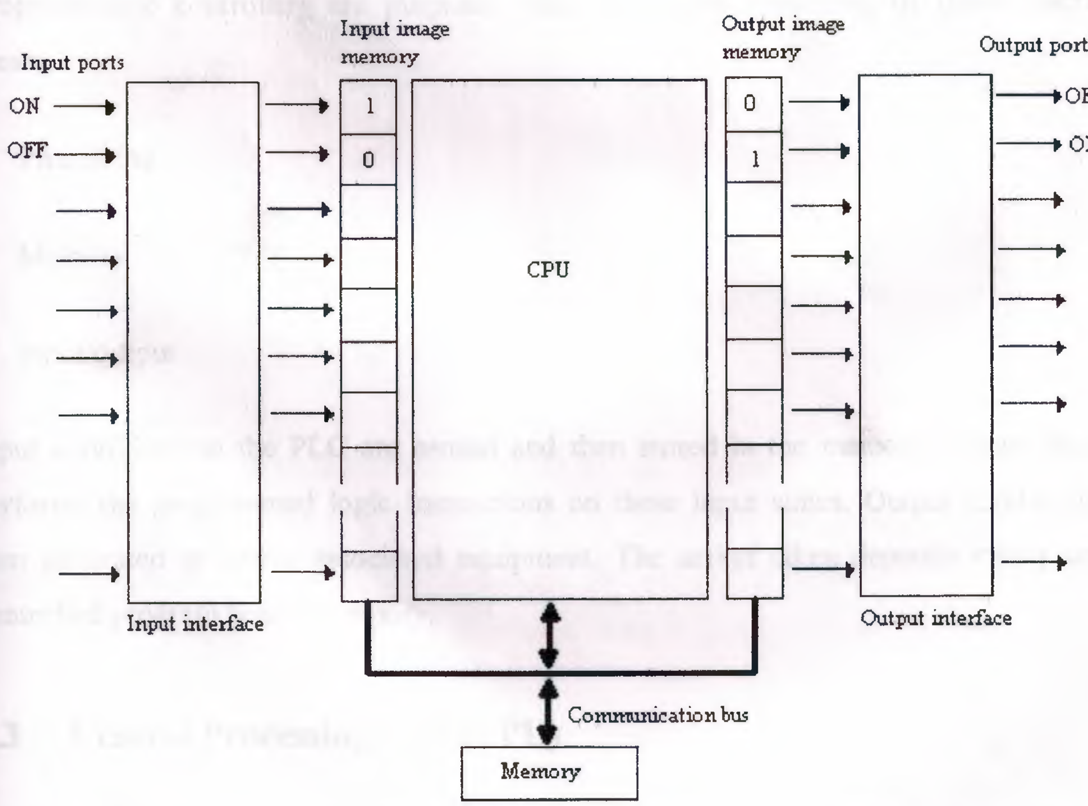


Figure 2.1 Internal structure of a PLC

The CPU processes the binary data stored in the input image memory and the corresponding data held in the output image memory according to the users program, which is stored in the main memory. The pit values held in the output image memory determine which output ports are energized. A binary 1 sets an output port ON and a binary 0 sets an output port OFF.

A special program called the operating system controls the action of the CPU and consequently the execution of the users program. The operating system is supplied by the PLC manufactures and is permanently held in memory. A PLC operating system is designed to scan image memory and the main memory, which stores the ladder diagram program.

2.2 PLC Hardware Design

Programmable controllers are purposed built computers consisting of three functional areas:

- Processing
- Memory
- input/output

Input conditions to the PLC are sensed and then stored in the memory, where the PLC performs the programmed logic instructions on these input states. Output conditions are then generated to derive associated equipment. The action taken depends totally on the controlled program held in memory.

2.3 Central Processing Unit (CPU)

The CPU controls and supervises all operations with in the PLC, carrying out programmed instructions stored in the memory. An internal communication highway or bus system carries information to and from the CPU, memory, and I/O units under control of CPU. The CPU is supplied with a clock frequency by an external quartz crystal or RC oscillator, typically between 1 and 8 megahertz depending on the microprocessor used and the area of application. The clock determines the operating speed of the PLC and provides timing / synchronization for all elements in the system.

It should be clear from this preamble that we use the memory to store various types of information. This information might be an image of input and output ports, the users program, the operating system or data. Different types of memory devices are used for different types of information.

2.4 Memory

Memory is characterized by its volatility. A memory is volatile if it loses its data when the power to it is switched off and non-volatile otherwise. Common types of memory include semiconductor memory and magnetic disk. The some types of semiconductor memory are:

1. RAM

Random access memory is a flexible type of read/write memory. All PLCs will have some amount of RAM, which is used to store ladder programs being developed by the user program data which needs to be modified and image data.

Ram is volatile. This means that RAM cannot be used to store data while the PLC is turned off unless the RAM is battery backed. A type of RAM called CMOS RAM (complementary metal-oxide semiconductor RAM) is suitable for use with batteries because it consumes very little power and operates over a very wide range of supply voltage.

2. ROM

A read only memory is programmed during its manufacture using a mask. It is a nonvolatile memory and provides permanent storage for the operating system.

3. EPROM

Erasable programmable read only memory is a type of ROM, which can be programmed by electrical pulses and erased by exposing a transparent quartz window found in the top of each device to ultraviolet light. EPROM is nonvolatile memory and provides permanent storage for ladder programs.

4. EEPROM

Electrically erasable programmable read only memory is similar to EPROM but is erased by using electrical pulses rather than ultraviolet light. It has the flexibility of battery backed CMOS RAM. However, writing data in to an EEPROM takes much longer than into a RAM.

In addition to program storage, a programmable controller may require memory for other function:

- Temporary buffer store for input/output channel status I/O RAM.
- Temporary storage for status of internal functions, e.g. timers, counters, marker relays, etc.

Since these consist of changing data (e.g. an input point changing state) they require RAM read/write memory, which may be battery backed in section.

2.4.1 Memory Storage Capacity

The storage capacity of a memory device is determined by the number of binary digits, i.e. the binary number 2^{10}. A 4K-byte memory is capable of storing 4*1024 words, each of 8 bits, and has a total storage capacity of 32768 bits.

Clearly, the storage capacity of the user memory will determine the maximum program size. As a guide, a 1K-byte memory will hold 1024 program instructions and data if these are stored as groups of 8 bits.

2.4.2 Memory Size

Smaller programmable controllers normally have a fixed memory size, due in part to the physical dimensions of the unit. This varies in capacity between 300 and 1000 instructions depending on the manufacturer. This capacity may not appear large enough to be very useful, but it has been estimated that 90% of all binary control tasks can be solved using less than 1000 instructions, so there is sufficient space to meet most user's needs.

Larger PLCs utilize memory modules of between 1K and 64K in size allowing the system to be expanded by fitting additional RAM or PROM memory cards to the PLC rack.

As integrated circuit memory costs continue to fall, the PLC manufacturers are providing larger program memories on all products.

2.4.3 Memory Map

Memory mapping is used to describe the situation in which input/output ports are controlled by writing data into the allocation of memory addresses of ROM, RAM and I/O is called a memory map. Figure 2.2 illustrates a memory map for a typical PLC. In this, image bits are stored in RAM above the user's program and data for flags, counters, and timers. Flags, counters, and timers are discussed below with most PLCs the memory map is already configured by the manufacturer. This means that the program capacity, the number of input/output ports and the number of internal flags, counters and timers are fixed.

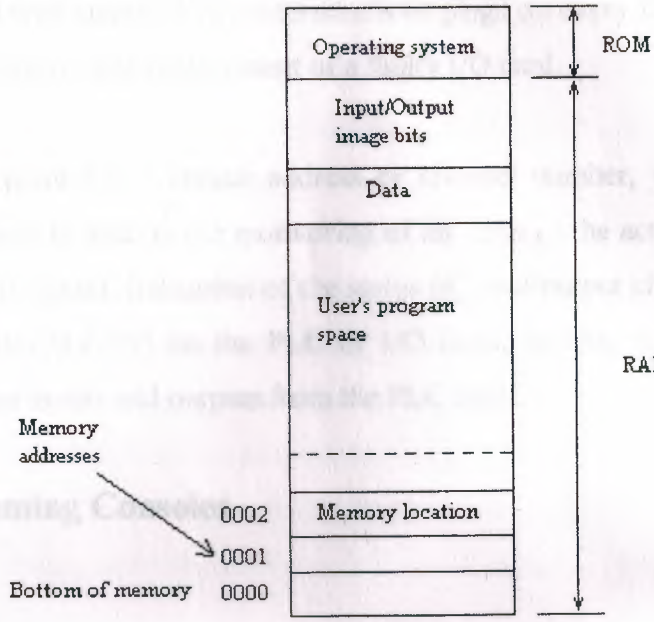


Figure 2.2 Memory map

2.5 Input / Output Units

Most PLCs operate internally at between 5 and 15V dc. (Common TTL and CMOS voltages), whilst process signals can be much greater, typically 24V dc to 240V ac at several amperes.

Input (Choice of): 5V (TTL level) switched I/P
24V switched I/P
110V switched I/P
240V switched I/P

Output (Choice of): 24V 100mA switched O/P
110V lamp
240V lAac (triac)
240V 2A ac (relay)

In all cases the input/output units are designed with the aim of simplifying the connections to process transducers and actuators to the programmable controller. For this purpose all

PLCs are equipped with standard screw terminals or plugs on every I/O point, allowing the rapid and simple removal and replacement of a faulty I/O card.

Ever input/output point has a unique address or channel number, which is used during program development to specify the monitoring of an input or the activating of a particular output with in the program. Indication of the status of input/output channels is provided by light emitting diodes (LEDS) on the PLC or I/O units, making it simple to check the operation of process inputs and outputs from the PLC itself.

2.6 Programming Consoles

Programs are entered into the PLC's memory using a program console (ladder). Program consoles vary from hand held system incorporating a small keyboard and liquid crystal displays (LCDs) to CRT (Cathode ray tube) terminals. Larger PLCs are often programmed using a visual display unit (VDU) with a full keyboard and screen display, connected to the controller via a serial link. VDUs provide improved programming facilities such as screen graphics and the inclusion of text comments that assist in the readability of a program.

2.7 Program Units

All but the simplest programming panels contain enough RAM to enable semi permanent storage of a program under development or modification. If the programming panel is a portable unit, its RAM is normally CMOS type with battery backup, allowing the unit to retain programs whilst being carried around a plant or factory floor. Only when a program is ready for use/testing it will be transferred to the PLC. Once the installed program has been fully tested and debugged, the programming panel is removed and is free to be used on other controllers.

The terminal may have a monitoring and forcing facility allowing real time observation of switches, gates and functions during program execution this can be valuable for troubleshooting, especially when the target process is remote or in accessible.

Recently most PC manufacturers have configured personal computers as program development workstations. The high-speed operation and screen graphics facilities of machines are ideal for graphics programming of ladder circuits. Also, the large memory available on modern 16-bit microcomputers is ideal for storage of several PLC programs complete a personal computers as a programmable controller workstation also provides the user with access to other useful software facilities for project management, such as databases, spread sheets, word processing and financial planning packages.

2.8 PLC Operating System

All PLC operating system execute a ladder program by scanning the logic states of the inputs and outputs stored in image memory. Most PLCs solve logic one rung at a time sequentially. The inputs of the first rung are scanned and the logic solved to determine the logic state of its output. This process is repeated for the second and third rung. When the 'END' rung is reached the scan cycle repeats itself so that each rung is scanned over and over again.

The program logic might involve simple AND, OR, NOT functions more advanced counting, timing, sequence and mathematical functions are available to the user the more sophisticated the operating system the more programming functions are provided.
The operating system is characterized by:

2.8.1 Scan Rate

The speed at which a PLC scans the memory is called the scan rate. The scan rate depends on how fast the CPU is clocked. It is expressed in terms of how many seconds it takes to scan a given amount of memory, usually 1K bytes.

The actual time to scan a program will depend on the scan rate, the length of the program and the types of functions used in a program. The faster the scan time the more often the inputs and outputs are checked.

2.8.2 Phasing Errors

The CPU, under the control of the operating system, scans the input image memory rather than the inputs themselves. The input image memory rather then the inputs themselves. The input memory is not changed while the CPU is scanning it. Thus it is possible for an input port to change its state from, say, off to on to off again before the input image memory is updated. A phasing error is said to have occurred when the CPU scan misses a change of state of an input port.

2.9 Programming PLC

The main requirement from any PLC programming language is that it may be easily understood and used in a control situation. This implies the need for a high level language to provide commands very close to the functions required by a control engineer, but without the complexity and learning time associated with most high level computer languages.

Ladder diagrams have been the most common method of describing relay logic circuits, so it was only natural to base PLC programming on them in order to create a familiar environment for the user and designer of small logic control systems.

2.9.1 Explanation of Ladder Diagrams

To show the relationship between a physical circuit and a ladder representation, consider the electric motor circuit as output in figure 2.3, The motor is connected to a power source via a switch in program line 1. The motor will turn on if the switch is made (closed). In the program line 3 the motor work if the input is activated (opened). The ladder diagram uses standard symbols to represent the circuit elements and functions found in a control system.

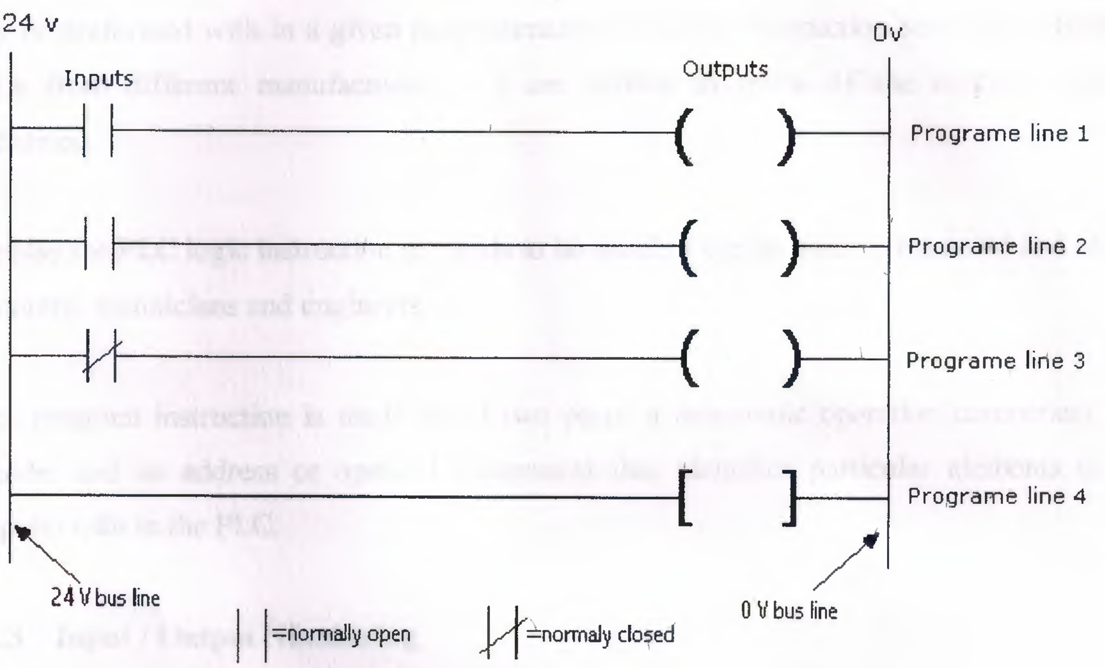


Figure 2.3 Motor circuit ladder diagram

The ladder diagram consists of two vertical lines representing the power rails plus circuit symbols that make up a rung of the ladder. Here the symbols represent three normally open switch contacts, one normally closed contact and one output device - the motor coil. Ladder symbols are used to construct any from of switched logic control system and the diagrams produced can be as complex as necessary for a particular application. An essential part of any ladder design is the documentation of the system and its operation, to allow any user to understand the ladder solution quickly.

2.9.2 Logic Instruction Set

The most common technique used for programming small PLCs is to draw a ladder diagram of the logic to be used, and then convert this into a programming panel attached to the programmable controller. These instructions are similar in appearance to assembly type codes, but refer to physical inputs, outputs and functions with in the PLC itself.

The instruction set consists of logic instructions (mnemonics) that represent the actions that may be preformed with in a given programmable controller. Instruction sets vary between PLCs from different manufacturers, but are similar in terms of the control actions performed.

Because the PLC logic instruction set tends to be small, it can be quickly mastered and used by control technicians and engineers.

Each program instruction is made up of two parts: a mnemonic operation component or opcode, and an address or operand component that identifies particular elements (e.g. outputs) with in the PLC.

2.9.3 Input / Output Numbering

These instructions are used to program logic control circuits that have been designed in ladder diagram from, by assigning all physical inputs and outputs with an operand (address) suitable to the PLC being used. The numbering systems used differ between manufacturers, but certain common terms exist. For example, the symbol X is used to represent inputs, and Y to label outputs.

A range of addresses will be allocated to particular elements. Thus for these programmable controllers the symbol X or Y is redundant, being used purely for the benefit of the user. However, for many PLCs both parts of the address are essential, since the I/O number ranges are identical.

2.10 Timing Considerations

Note that by virtue of the cyclic nature of the program I/O copy the status of inputs and outputs cannot be changed with in the same program cycle. If an input signal changes state after the copy routine, it will not be recognized until the next copy occurs.

The time to update all inputs and outputs depends on the total number to be copied, but is typically a few milliseconds in length. The total program execution time (or cycle time) depends on the length of the control program. Each instruction takes 1-10 μ. To execute depending on the particular programmable controller employed. So a 1K (1024) instruction program typically has a cycle time of 1-10 ms however, programmable controller programs are often much shorter than 1000 instructions, namely 500 steps or less.

2.11 Response Time

The response time of a PLC is the delay between an input being turned on and an output changing state. Delays are due to an output changing state. Delays are due to:

(a) The mechanical response of an output device such as a relay.

(b) The electrical response of an input circuit.

(c) The scan update of image memory.

Ladder circuits, which feed, back the logic states of output relays as inputs can cause a significant response time lag.

2.12 Power Supply

The CPU memory input/output are electronic components, which require power. A PLC incorporates a power supply for powering internal components and input ports.

Power supplies fall into two categories: linear and switch mode.

Linear Power Supply

A linear power supply uses a simple regulator circuit to convert the main supply to a constant dc voltage.

Switch Mode Power Supply

A switch mode power supply uses a high frequency switching regulator to produce a series of pulses. Averaging the pulses provides a smooth dc voltage. The main advantages of a switch -mode power supply are:

a) It is capable of providing a wide range of supply voltages (e.g. +/- 24V dc, +/- *15Vdc, +/-5Vdc,OV).*

b) Switch action makes it highly efficient so that the amount heat dissipated from the supply is small.

c) It is compact and lightweight.

Because of these advantages the switch mode power supply is often used in PLCs.

2.13 Remote Input / Output

When large number of input/output points are located a considerable distance away from the programmable controller, it is, uneconomic (and bulky) to run connecting cables to every point. A solution to this problem is to site a remote I/O unit near to the desired I/O points. This acts as a concentrator to monitor all inputs and transmit their status over a single serial communication link to the programmable controller. Once output signal have been produced by the PLC they are fed back along the communication cable to the remote I/O unit, which converts the serial data into the individual output, signals to derive the process.

2.14 Programming Large PLC

Virtually any function can be programmed, using the familiar ladder symbols via a graphic terminal or personal computer. Parameters are passed to relevant modules (such as PID) either by incorporating constants into the ladder, or via on screen menus for that module. There may in addition be computer-oriented languages (such as dialects of basic, etc.), which allow programming of function modules and sub routines.

There is process toward standardization of programming languages, with program becoming easier to overview improvement of text (comment) handling and improved documentation facilities. This is assisted by the application of personal computers as workstations.

2.15 Summary

The internal operation of any programmable controller is essentially similar to any other microprocessor-based system. Differences occur in the manner of input/output handling and the interface hardware providing. PLCs are specially designed to connect to most common industrial control systems, which are hardware specific, but they offer great flexibility through programming.

Today virtually every manufacturer of electronics control equipment markets a range of programmable controllers with facilities ranging from simple switched I/O through sophisticated continuous control. Developments in this area are continuing at a rate almost equal to that in the field of personal computing. Because of this, the power and operating speed of all programmable controllers is constantly improving, whilst equipment prices at worst remain steady, and frequently fall.

3. PROGRAMMING PLC SYSTEMS

3.1 Introduction

Logic instruction sets are used for programming PLC systems. The complete sets of basic logic instructions for two common programmable controllers are given below. Note the inclusion in these lists of additional instructions ORB and ANB to allow programming of more complex, multibranch circuits. Some typical instruction sets for Texas Instruments and Mitsubishi PLCs are given in table 3.1.

Table 3.1 Typical logic instruction sets.

Texas Instruments		Mitsubishi A series	
Mnemonic	Action	Mnemonic	Action
STR	Store (start a new rug of ladder diagram)	LD	Start rung with an open contact
OUT	Output	OUT	Output
AND	Series components	AND	Series components
OR	Parallel components	OR	Parallel components
NOT	Inverse action (used in conjunction with other instruction to invert their function)	..I	As for NOT, e.g.; ORI meaning NOR function
		ORB	Or together parallel Branches
		ANB	And together series circuit blocks

3.2 Logic Instructions and Graphic Programming

Logic instructions are the basic programming language for programmable controllers. Although logic instructions are relatively easy to learn and use, it can be extremely time-consuming to check and relate large coded program to the actual circuit function. In addition, logic instructions tend to vary between different types of PLC. If a factory or plant is equipped with a range of different controllers (a common situation), confusion can result over differences in the instruction sets.

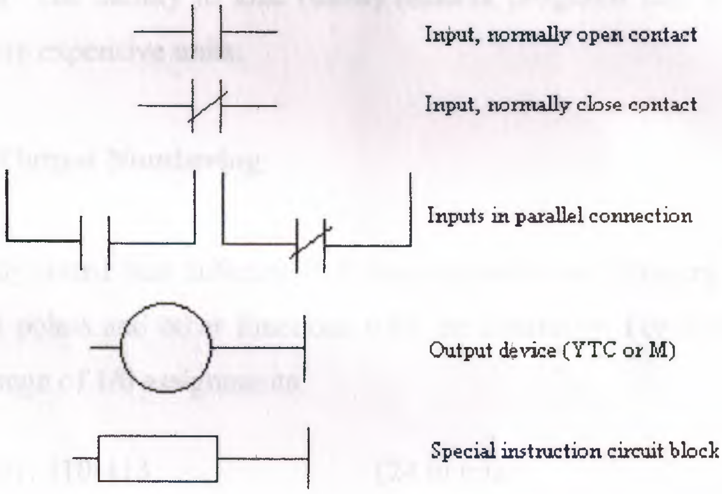


Figure 3.1 Graphic ladder symbols.

A preferable alternative is to use a graphic programmer, as available for several programmable controllers including the small Mitsubishi and Toshiba models from Japan. Graphic programming allows the user to enter his program as a symbolic ladder circuit layout, using standard logic symbols to represent input contacts, output coils, etc, as shown in the figure 3.1. This approach is user-friendlier than programming with mnemonic logic instructions, and can be considered as a higher-level form of language.

The programming panel translates or complies these graphic symbols into machine (logic) instructions that are stored in the PLC memory, relieving the user of this task.

Different types of graphic programmer are normally used for each family of programmable controllers, but they all support similar graphic circuit conventions. Smaller, hand-held panels are common for the small to medium-sized PLCs, although the same programming panel is often used as a 'field programmer' for these and larger PCLs in the same family. However, the majority of graphic programming for larger systems is carried out on terminal-sized units. Some of these units are also semi-portable, and may be operated alongside the PLC system under commissioning or test in-plant. In addition to screen displays, virtually all graphic-programming stations can drive printers for hard copy of programs and/or status information, plus program storage via battery-backed RAM or tape/floppy disk. The facility to load (blow) resident programs into EPROM ICs may be available on more expensive units.

3.2.1 Input / Output Numbering

It was previously stated that different PLC manufactures use different numbering systems for input/output points and other functions with the controller. For continuity, we will use the following range of 1/0 assignments:

Inputs: X400-407; 410-413 (24 in total)

Outputs: Y430-437; 500-507, 510-513 (16 in total)

3.2.2 Elementary Logic Circuit

The basic logic gates that may be formed using ladder logic were introduced under relay systems. These gates and others are now constructed using ladders symbols and logic instructions.

3.2.2.1 OR and AND Gates

These logic functions can be produced in ladder form very simply.

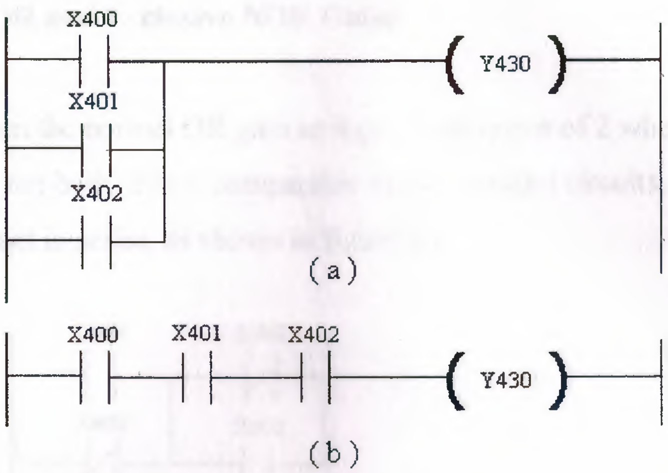


Figure 3.2 (a) OR gate; (b) AND gate.

3.2.2.2 NAND and NOR Gates

These logic functions can be produced in ladder form simply by replacing all contacts with their inverses; i.e. AND becomes ANI; OR becomes ORI, etc. this chamfers the function of the circuit. For example, the AND circuit with normally closed contacts becomes a NOR circuit.

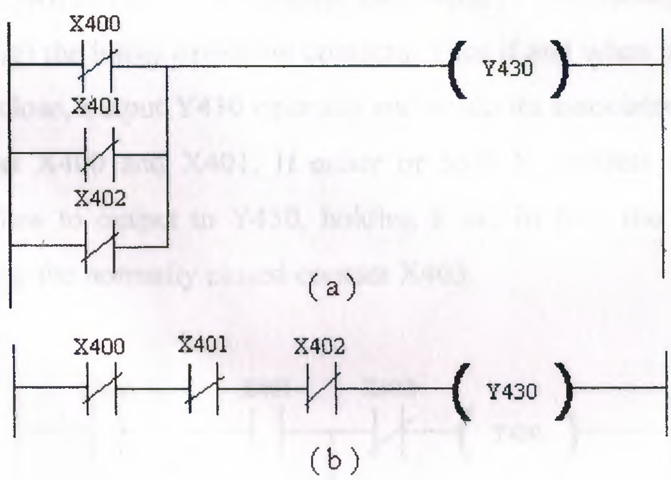


Figure 3.3 (a) NOR gate - if any contact is opened, Y430 will de-energize;

(b) NAND gate - All contacts must be opened to de-activate Y430.

3.2.2.3 Exclusive OR and Exclusive NOR Gates

This is different from the normal OR gate as it gives an output of 2 when either one input or the other is on, but not both, this is comparable to two parallel circuits, each with one make and one break contact in series, as shown in figure 3.4.

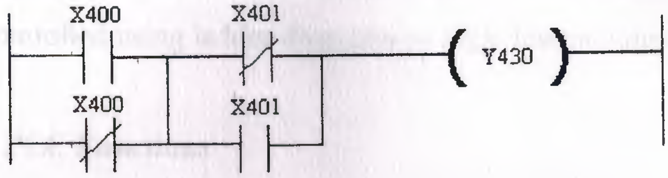


Figure 3.4 Exclusive - OR gate.

3.2.3 Memory Circuits

Memory elements are often required in logic control systems, being used to store brief command signals that must be transformed into continuous enabling signals. This type of operation is performed by a self- maintain circuit or latch.

A latch circuit is shown in Fig 3.5. It involves the output (Y430) having a bridge contact in parallel with (ORing) the initial operating contacts. Thus if and when the initiating contacts (X400 and X401) close, output Y430 operates and so do its associated contacts. Thus, the latch is now across X400 and X401. If either or both X contacts open, the latch path maintains power flow to output to Y430, holding it on. In fact, the only way to release Y430 is by operating the normally closed contact X403.

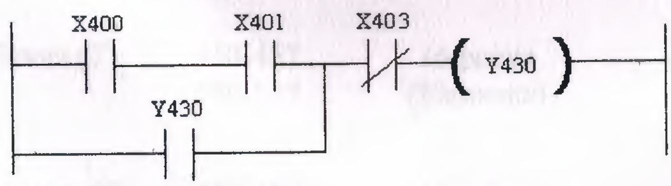


Figure 3.5 Memory latch circuits.

3.3 Facilities

In addition to the series and parallel connection of input and output (associated) contacts the majority of control tasks involve the use of time delays, even counting, storage of processed status data, etc. All of these requirements can be met using standard features found on most programmable controllers. These include timer, counters, markers and shift registers, easily controlled using ladder diagrams or logic instructions.

3.3.1 Standard PLC Functions

These internal functions are not physical input or output. They are simulated with in the controller. Each function can be programmed with related contacts (again simulated), which may be used to control different elements in the program (see figure 3.6).

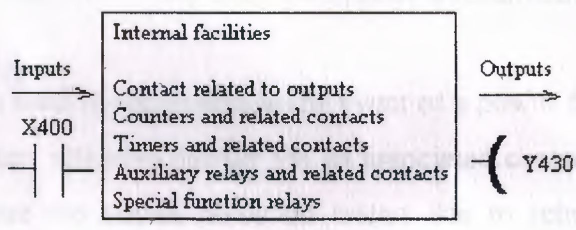


Figure 3.6 Standard PLC functions.

As with physical inputs and outputs, certain number ranges are allocated to each block of function. The number range will depend both on the size of PLC and the manufacturer. For example, for the Mitsubishi F40- series, the details are as follows (octal numbering has been used):

Timers (T)	450-457 550-557	16 points (Elements)
Counters (C)	460-467 560-567	16 points

The use of different number ranges assigned to each supportive function e.g. the timer circuit for this programmable controller are addressed from 450-457 and 550-557, a total of 16 timers. It is the specified number that identifies a function and its point to the PLC, not the prefix letter (T in this case). These prefixes are included only to aid the operator.

3.3.2 Retentive Battery-Backed Relays

If Power is cut off or interrupted while the programmable controller is operating the output relays and all standard marker relays will be turned off. Thus when power is restored, all contacts associated with output relays and markers will be off-possibly resulting in incorrect sequencing. When controlled tasks have to restart automatically after a power failure the use of battery-backed markers is required. In an above PLC there are 64retentive marker points, which can be programmed as for ordinary markers only storing pre-power failure information that is available once the system is restarted.

Retentive marker is used to retain data in the event of a power failure. Once input is closed to operate the marker, retentive marker via its associated contact. So even if input is open due to power failure the circuit holds on restart due to retentive marker retaining the 'operated' status and placing its associated contacts in the operated positions. Obviously input still controls the circuit, and if this input is likely to be energized (opened) by a power failure situation, then a further stage of protection may be used.

There are many other application of marker facilities, some of which work as elements in combination with PLC function.

3.3.3 Optional Function and Auxiliary Relays

Auxiliary Relays constitute an important facility in any programmable controller. This is basically due to their ability to control large number of associated contact and perform as intermediate switching elements in many different types of control circuit.

In addition many PLC manufacturer have provided additional, programmable functions associated with these auxiliary relays, to further extend their usefulness. A very common example is a " pulse" function that allows any designated marker to produce a fixed duration pulse at its contacts when operated rather that the normal dc level change. This pulse output is irrespective of the duration of relay operation thus providing a very useful tool for application such as program triggering, setting/resetting of timer and counter, etc.

3.3.4 Pulse Operating

The programming of this feature (and others like it) varies between controllers, but the general procedure is the same and very straight forward. A pulse- PLS instruction is programmed onto an auxiliary relay to out put a fixed-duration pulse (equal to one cycle time of the program) when operated. The relay may be used to output a pulse for either a positive or negative going input. Some circuit uses a PLS instruction on auxiliary relay to provide a use for counters and timers because they often require short duration to the restart of the counting or timing process.

3.3.5 Set-Reset

As with pulse-PLS, the ability to SET and RESET and auxiliary can often be produced by using appropriate instructions. These instructions are used to hold (latch) and reset the operation of the relay coils. The set (S) instruction causes the coil to self-hold. This remains until a reset (R) instruction is activated.

3.3.6 Timers

In a large proportion of control applications, there is a requirement for some aspects of timing control. Time is nearly always a part of a control system. PCs have software timer facilities that are very simple to program and use in a variety of situations. A PLC system must therefore include timers as a part of it's programming language. There are many types of timers.

The common method of programming a timer circuit is to specify the interval to be timed, and the conditions or events that are to start and/or stop the timer function. A 0 to 1 transition is delayed for a present time T, but a 1 to 0 transition is not delayed at all. An input signal shorter than T is ignored. This type of timer is called a delay. The off delay passes a 0 to 1 transition instantly but delays the 1 to 0 transition. A common use of the delay can be obtained from an on delay by using the inverse of the input signal and taking the inverse of the timer output signal. An edge-triggered pulse timer gives a fixed width pulse for every zero to one transmission at the timer input. The initiating event maybe produced by other internal or external signals to the controller. For example the timer T450 is totally controlled by a constant related to output Y430. Thus, T450 begins timing only when Y430 is operated. This is caused by input X400 and not X401. Once activated, the timer will 'time down' from its preset value in this case 3.5 seconds to zero, and then it's associated contacts will operate.

As with any other PLC contact, the timer contacts may be used to drive succeeding stages of ladder circuitry. Here the T450 contact is controlling output Y431. The enabling path to a timer may also form the 'reset' path, causing the timer to reset to the present value, whenever the path is opened. This is the case with most small PCs. The enabling path may contain very invalid logic, or only a single contact.

Techniques for programming the preset time value vary little between different programmable controls. Usually requiring the entry of a constant (k) command followed by the time interval in seconds and tenth (or hundredths) of a second. The timers on this Mitsubishi controller can time from 0.1 to 999.9s, and can be cascaded to provide longer intervals if required.

A timer of whatever types have some values that need to be set by the user. The first of these is the basic unit of time (that is what units the time is measured in). Common units are 10ms, 100ms, Is, 10s and 100s. The base unit does not affect the accuracy of the timer; normally the accuracy is similar to the programs scan. Next the timer duration (often called the preset) is defined. This is normally set in terms of the time base, a timer with a preset of

150 and a time base of 10ms will last 1.5s, for example. In small PLCs this preset is set by the programmer, in the larger PLCs the duration can be changed from within the program itself.

When a timer is used there are several signals that may be available:

- EN (for enable) is a mimic of the timer input.
- TT (for timer timing) is energized whilst the time is running.
- DN (for done) says the timer has finished.

In larger PLCs the elapsed time (often called the accumulated time) may be accessed by the program for use elsewhere (a program may be required to record how long a certain operation takes). PLC manufacturers differ on how a timer is programmed. Some treat the timer as a delay block with the preset being stored in a VALUE block. Siemens use a similar idea, but have different types of timer. Some however, uses the timer as a terminator for a rung, with a timer signals being available as contacts for use elsewhere.

The accumulated time in the timers discussed so far goes back to zero each time the input goes to a zero. This is known as a non-retentive timer. Most PLC timers are of this form. Occasionally it is useful to have a timer, which holds it's current value even though input signal has gone. When the input occurs again the timer continues from where it stopped. This, not surprisingly, is known as a retentive timer. A separate signal must be used to reset the timer to zero. If a retentive timer if not available on a particular PLC, the same function can be provided with a counter.

A typical timer can count up to 32767 base time units (corresponding to sixteen binary bits). Some older PLCs working in BCD can only count to 999. With a Is time base the maximum time will be just over 546min or about 9h. Where longer times are needed (or times with a resolution better than 1 s) timers and counters can be used together.

3.3.7 Counters

Counting is a fundamental part of many PLC programs. The PLC maybe required counting the number of times in a batch, or recording the number of times some event occurs. Whenever the number of process actions or events is of significance they must be detected or stored in some manner by the controller. Single or small numbers of events may be remembered by using latched relay circuits, but this is not suitable for larger event counts. Here programmable counter circuits are desirable, and are available on all PLCs. Not surprisingly, all PLCs include some form of counting element.

Although not only PLCs will have the facilities. There will be two numbers associated with the counter. The first is the count itself (often called the accumulated value), which will be incremented when a 0→1 transition is applied to the count down input. The accumulated value (count) can be reset to zero by applying a 1 to the reset input. Like the elapsed time in timer, the value of the count can be read and used by other parts of the program.

The second number is the preset, which can be considered as the target for the counter. If the count value reaches the preset value, a count complete or counts down signal is given. The preset can be changed by the program; a batching sequence.

Provided as and internal function can counter circuit or program in a similar manner to the timer circuit. But with the addition of a control path to signal event count to the counter block. Most PLC counter work as subtraction or down counter, as the current value is decremented from the program set value. PLC manufacturer handles counters, like timers in slightly different ways. The use of count up (CTU), count down (CTD and reset (RES) as rung terminator. With the count down signals available for use as a contact. Some treat a counter as an intermediate block in logic diagram or rung from which the required output signals can be used.

Like timers, most PLCs allow a counter to count up to 32767. Where larger counts are needed, counters can be cascaded with the complete (or done) signal from the first counter being used to step up the second counter and reset the first. Suppose counter 1 holds the being used to step up the second counter and reset the first. Suppose counter 1 holds the range 0-999, and counter 2 the thousands. If counter 2 holds 23516 and counter 1 holds 457, the total count is 23516457.

When the timer has not timed out, the DN signal is not present and the timer is running. When it reaches the preset, the DN signal occurs, resetting and restarting the timer. The resulting Is pulse is counted by successive counters to give accumulated seconds/minutes/hours/days/years. As each counter reaches it's preset it steps the next counter and resets it. This technique is widely used to log hours run for pumps, fans and similar devices for maintenance scheduling. In this case the 'event' in the second rung will be an auxiliary contact on the motor starter.

Long duration timers built from counters are normally retentive (i.e. they hold their value when the controlling event is not present). They can be made non-retentive by resetting the counters when the controlling event is not present, but this is rarely required.

3.4 Arithmetic Instructions

Numerical data implies the ability to do arithmetical operations, and all PLCs provide the ability to do at least four function mathematical operations (add, subtracts, multiply and divide).

3.4.1 BCD Numbering

All internal CPU operations are performed in binary numbers. Since it may be necessary to deal with decimal inputs and outputs in the outside world, conversion using binary coded decimal (BCD) numbering is provided on most PLCs. When data is already in binary format, such as analog values, it is placed directly in registers for use by other instructions.

Decimal input/output is often required for operator input via thumb wheel switches or similar devices, with a decimal of certain information back to the operator.

3.4.2 Magnitude Comparison

Magnitude comparison instructions are used to compare a digital value read from some input device or timer, etc., with a second value contained in a destination data register. Depending on the instruction more than, less than, or equal this will result in a further operation when the condition is met. For example, a temperature probe in a furnace returns an analog voltage representing the current internal temperature. This is converted into a digital value by an analog - to - digital converter module on the PC, where it is read from input points by a data transfer instruction and stored in data register DIO. The process requires that if the temperature is less than 200°C, then the process must half due to insufficient temperature. If the temperature is greater than 200°C and less than 250°C, then the process operates at normal rate (e.g. items are baked for 5 minutes each). If the temperature is between 250 and 280°C, then baking time is to be reduced to 3 minutes 25 seconds, and once temperature exceeds 280°C the process is to be suspended.

This is the type of area where magnitude comparison can provide the necessary control, in conjunction with other circuitry to drive the plant equipment.

Other common applications include the checking of counter and timer values for action part way through a counting sequence.

3.4.3 Addition and Subtraction Instructions

These instructions are used to alter the value of data held in data registers by a certain amount. This may be used simply to add/subtract an offset to an input value (in order to place it within range) before it is processed by other instructions. For example, when two different sensors are passing values to the controller and one sensor signal has to be compared against the other, but is a fundamentally smaller signal with a narrower output swing. It may be possible to add an offset to the smaller signal to bring it up near to the level of the larger one, thus allowing comparison to take place. The alternative would be to use signal-conditioning units to raise the sensor output before the PIC - an expensive option.

Other uses of + and - include the alteration of counter and timer presets by programmed increments when certain conditions occur.

3.4.4 Extended Arithmetic Function

All programmable controllers are being given more advanced features as the demand dictates, including more advanced arithmetic and data-handling facilities, along with a growing number of smaller controllers.

Extended arithmetic may include:

- Double precision add and subtract
- Multiplication and division
- Scientific functions

In many cases these arithmetic functions are carried out on signed values, where any value can be a positive or negative number. This can be very useful when dealing with analog values of between, say, -10 and +10V, thermocouple input signals of between -120° F and +60° F, or any other parameter that can produce an output which swings between positive and negative values.

The double precision functions normally use a pair of consecutive 16-bit registers to represent each value, allowing integer values of up to plus or minus $2147483647 2^{31}(-l)$ to be handle. Apart from the size of the data words, double-precision arithmetic operates in a similar fashion to conventional signed arithmetic.

Multiple operations normally use two single-precision (single register) signed values and multiple the contents. This results in a signed double- precision product that occupies two registers. Division operates on a double precision signed number (pair of registers) and divides it by a single-precision signed value, resulting in a signed single- precision quotient together with a remainder.

Multiplication and division facilities allow the construction of relatively sophisticated mathematical algorithms that can be used to (amongst other things) process signals and values to perform direct closed-loop control.

3.5 Summary

We have looked at a fairly broad rang of PLC instructions, functions and circuit applications. These are the fundamental building blocks for all logic and interlock control programs, and in some cases can constitute almost the complete solution. Even so, not all of the possible facilities that can be provided by programmable controllers have been introduced; several others exist for use mainly in sequential control and other specialist applications.

4. AN OVERVIEW OF SIEMENS S7-200 MICRO-CONTROLLER

4.1 Overview of an S7-200

STEP 7-Micro/WIN supports the S7-200 CPUs by giving you the features to set up and manage your application project. A project consists of the program you enter with STEP 7-Micro/WIN, along with the documentation you write for the program and the configuration you set up for the CPU.

You have the option of selecting either Ladder or Statement List as your programming language. With the S7-200 CPUs, you have a basic program structure that gives you flexibility in setting up any subroutines or interrupts that you program.

4.2 Introduction to the Simatic S7-200 Micro PLC

The Simatic S7-200 series is a line of micro-programmable logic controllers (Micro PLCs) that can control a variety of automation applications. Figure 4.1 shows an S7-200 Micro PLC. The compact design, expandability, low cost, and powerful instruction set of the S7-200 Micro PLC make a perfect solution for controlling small applications. In addition, the wide variety of CPU sizes and voltages provides you with the flexibility you need to solve your automation problems.

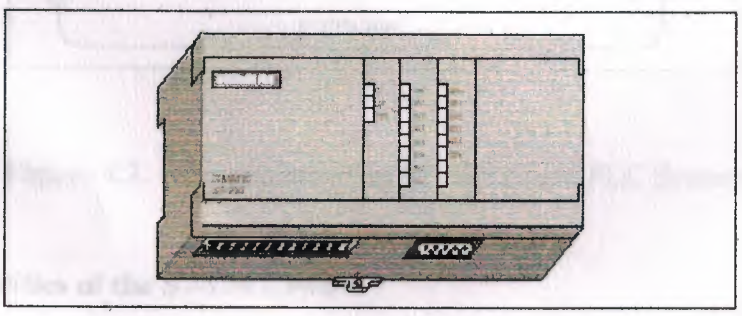


Figure 4.1. S7-200 Micro PLC

4.3 Comparing the Features of the S7-200 Micro PLCs

4.3.1 Equipment Requirements

Figure 4.2 shows the basic S7-200 Micro PLC system, which includes an S7-200 CPU module, a personal computer, STEP 7-Micro/WIN programming software, and a communications cable.

In order to use a personal computer (PC), you must have one of the following sets of equipment:

- A PC/PPI cable
- A communications processor (CP) card and multipoint interface (MPI) cable
- A multipoint interface (MPI) card. A communications cable is provided with the MPI card.

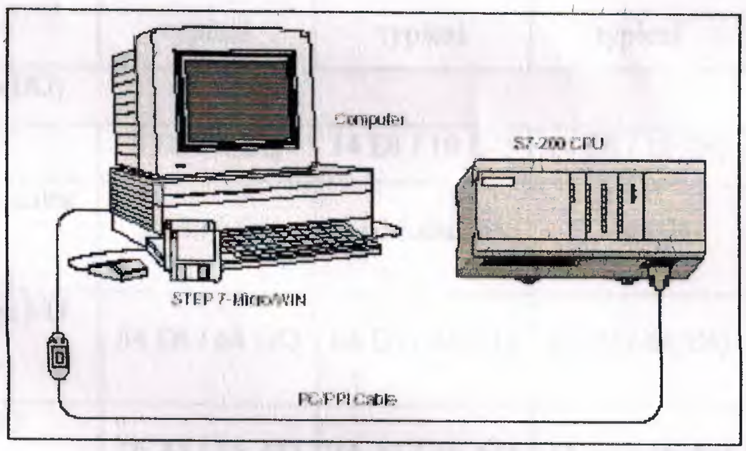


Figure 4.2. Components of an S7-200 Micro PLC System

4.3.2 Capabilities of the S7-200 CPUs

The S7-200 family indudes a wide variety of CPUs. This variety provides a range of features to aid in designing a cost-effective automation solution. Table 4.1 provides a summary of the major features of each S7-200 CPU.

Table 4.1. Summary of the S7-200 CPUs

Feature	CPU 212	CPU 214	CPU 215	CPU 216
Physical Size of unit	160mm x 80mm x 62mm	197mm x 80mm x 62mm	218mm x 80mm x 62mm	218mm x 80mm x 62 mm
Memory				
Program (EEPROM)	512 words	2 k words	4 k words	4 k words
User Data	512 words	2 k words	2.5 k words	2.5 k words
Internal Memory Bits	128	256	256	256
Memory Cartridge	None	Yes (EEPROM)	Yes (EEPROM)	Yes (EEPROM)
Optional Battery Cartridge	None	200 Days typical	200 Days typical	200 Days typical
Backup (super capacitor)	50 Hours typical	190 Hours typical	190 Hours typical	190 Hours typical
Inputs/Outpuls(I/O)				
Local I/O	8 DI / 6 DQ	14 DI / 10 DQ	14 DI / 10 DQ	24 DI / 16 DQ
Expansion Modules (max.)	2 Modules	7 Modules	7 Modules	7 Modules
Process-Image I/O Register	64 DI / 64 DQ	64 DI / 64 DQ	64 DI / 64 DQ	64 DI / 64 DQ
Analog I/O (expansion)	16 AI / 16 AQ	16 AI / 16 AQ	16 AI / 16 AQ	16 AI / 16 AQ
Selectable Input Filter	No	Yes	Yes	Yes

4.4 Major Components of the S7-200 Micro PLC

An S7-200 Micro PLC consists of an S7-200 CPU module alone or with a variety of optional expansion modules.

4.4.1 CPU Module

The S7-200 CPU module combines a central processing unit (CPU), power supply, and discrete I/O points into a compact, stand-alone device.

- The CPU executes the program and stores the data for controlling the automation task or process.
- The power supply provides electrical power for the base unit and for any expansion module that is connected.
- The inputs and outputs are the system control points: the inputs monitor the signals from the field devices (such as sensors and switches), and the outputs control pumps, motors, or other devices in your process.
- The communications port allows you to connect the CPU to a programming device or to other devices. Some S7-200 CPUs have two communications ports.
- Status Lights provide visual information about the CPU mode (RUN or STOP), the current state of the local I/O, and whether a system fault has been detected.

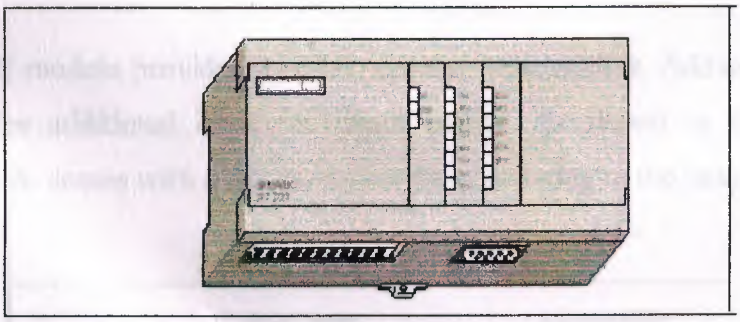


Figure 4.3 S7-212 CPU Module

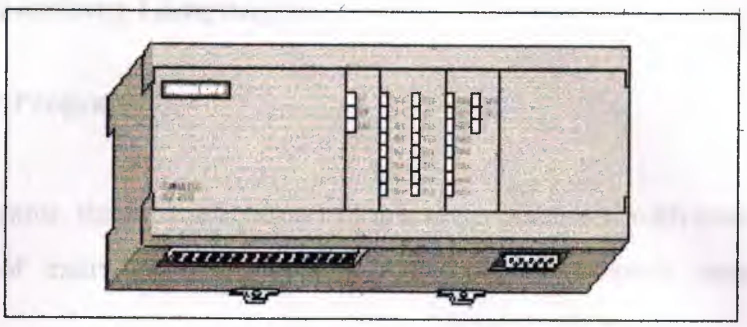


Figure 4.4 S7-214 CPU Module

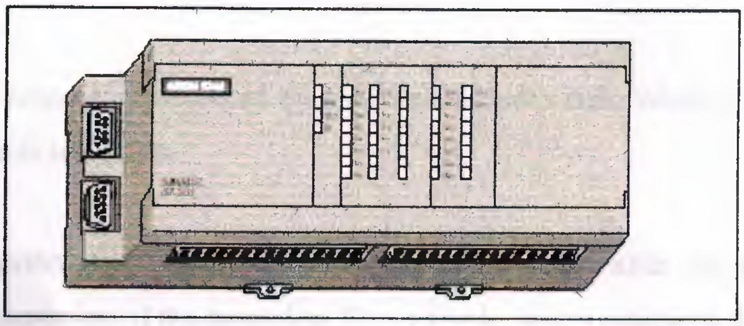


Figure 4.5 S7-215 and S7-216 CPU Module

4.4.2 Expansion Modules

The 7-200 CPU module provides a certain number of local I/O. Adding an expansion module provides additional input or output points. As shown in Figure 4.4, the expansion module comes with a bus connector for connecting to the base unit.

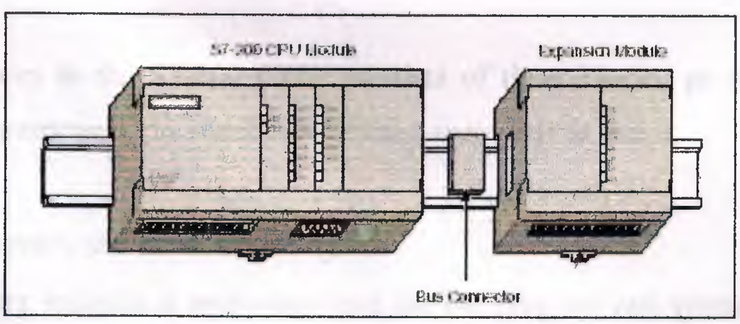


Figure 4.4 CPU Module with an Expansion Module

4.5 Programming Languages

4.5.1 Ladder Programs

In Ladder programs, the basic elements of logic are represented with contacts, coils, and boxes. A set of interconnected elements that make a complete circuit is called a network.

A hard-wired input is represented by a symbol called a contact. A normally open contact enables power flow when closed. A contact can also be normally closed. In this case, power flow occurs when the contact is opened.

A hard-wired output is represented by a symbol called a coil. When a coil has power flow, the output is turned on.

A box is a symbol for a complex operation performed within the CPU. The box simplifies programming of the operation. For example, boxes represent timers, counters, and math operations.

4.5.2 STL Programs

STL program elements are represented by a set of instructions for performing the desired functions. Instead of using the graphic display as shown by ladder programs, the STL program is shown in text format.

4.6 CPU Memory

The user memory in the S7-200 CPUs consists of three blocks: program, data, and configurable parameters. The blocks are defined according to usage:

- Program memory stores the user program.
- Data memory includes a temporary area for the program and storage of data. The temporary storage, calculations, and constants reside in data memory. Additionally,

data for timers, counters, high-speed counters, and analog inputs and outputs are stored in data memory.

- Configurable Parameter memory stores either the default or the modified parameters of the program setup. The configurable parameters include items such as protection level, password, station address, and retentive range information.

4.7 Simatic S7-200 Application Areas

The SIMATIC S7-200 series is a line of micro-programmable logic controllers (Micro PLCs) that can control a variety of automation applications. Compact design, low cost, and a powerful instruction set make the S7-200 controllers a perfect solution for controlling small applications. The wide variety of CPU sizes and voltages, and the windows-based programming tool, give you the flexibility you need to solve your automation problems.

4.7.1 The S7-200 is characterized by the following properties

- Easy entry
- Uncomplicated operation
- Peerless real-time characteristics
- Powerful communications capabilities

The S7-200 also covers areas where previously special electronics have been developed for cost reasons. Application areas include:

- Baling processes
- Plaster & Cement mixers
- Suction Plants
- Centralized lubricating systems/flange lubricating systems
- Woodworking machinery
- Gate controls
- Hydraulic lifts
- Conveyor systems

- Food & Drink Industry
- Laboratories
- Modem applications via dial-up, leased-line, or radio remote monitoring (SCADA)
- Electrical Installations

4.7.2 Mechanical features include

- Rugged, compact plastic housing using SIMATIC's prize-winning design
- Easily accessible wiring and operator control and display elements protected by front covers
- Installs on standard horizontal or vertical DIN rail or direct cabinet mounting with built-in mounting
- Terminal block as permanent wiring assembly (optional)

4.7.3 Design features

- International standards; Meets the requirements through compliance with VDE, UL, CSA and FM standards.
- The quality management system used during manufacturing has ISO 9001 certification; and Data back up; the user program and the most important parameter settings are stored in the internal EEPROM. A heavy-duty capacitor provides additional back up for all data over longer periods (typically up to 50 or 190 hours). An optional battery module ensures that the data remain stored for 200 days (typically) after power failure.

4.7.4 Benefits of the S7-200

The SIMATIC S7-200 Micro PLC is a full-featured programmable logic control system offering stand-alone CPUs, micro-modular expansion capability, and operator interface solutions. Almost any application that requires automation, from basic discrete or analog control, to intelligent networked solutions, can benefit by using the powerful S7-200 family of products.

The SIMATIC S7-200 offers real-time control with Boolean processing speeds of 0.37μs per instruction. This fast execution speed, combined with our 20Khz high-speed counters, interrupts, and 20KHz pulse outputs, provide quick responses in demanding real-time applications. The S7-200 has over 200 instructions, including math, PID, For/Next loops, subroutines, sequence control, and more.

All S7-200 CPUs offer at least one RS485 communication port with speeds up to 187.5Kbaud. This not only provides fast access for programming and maintenance, but also allows you to build master/slave networks with up to 31 stations.

Using our Freeport capability can also connect non-S7-200 devices, such as bar code readers, intelligent machines, etc.. With Freeport, you can easily adapt the S7-200 CPU to virtually any serial ASCII protocol.

5. PRACTICAL IMPLEMENTATION WITH PLC

5.1 Overview

The main aim of this chapter is to control a Garden Gate opening and closing operation and beside this control the Camera System and Doorbell. The inputs and outputs used in this PLC program are given in table 5.1.

Table 5.1 Inputs, Outputs and Timers that used in the Program

Inputs	X0	System Start button
	X1	Light Barrier sensor at the outside of garden
	X2	Light Barrier sensor at the inside of garden
	X3	Remote control receiver
	X4	Light Barrier at the door
	X5	Gate open sensor
	X6	Gate close sensor
	X7	Button to activate the Camera
	X8	Outer Doorbell button
	X9	Button to open the door/gate
Outputs	Y0	Controlling System Active Indicator Lamp
	Y1	Controlling operation of opening the gate
	Y2	Controlling the operation of closing the gate
	Y3	Controlling the operation of opening the door
	Y4	Controlling the Doorbell
	Y5	Controlling the Camera
Timers	T 200	Timing to close the gate after the car passed
	T 201	Timing to activate the doorbell automatically
	T 202	Timing to close the camera system

5.1.1 Explanation of Networks Used in Program

Network 1: This is used to activate or inactivate the system.

Network 2: This is used to drive the mechanism to open the gate. In order to open the gate when a vehicle interrupts the outer X1 (I0.1) or inner X2 (I0.2) light barrier sensor the door close sensor X6 (I0.6) should be active and from remote control sensor X3 (I0.3) or from the button X9 (I0.9). This network can be disactiveted only by gate open sensor X5 (I0.5).

Network 3: Door opening is controlled by this network. If there is no vehicle interrupts the light barrier sensors X1, X2 (I0.1, I0.2) and if remote control sensor X3 (I0.3) is activated or button X9 (I0.9) is pushed.

Network 4: This network is used to time up 10 seconds to close the gate when the gate is totally opened and the light barrier sensors X1, X2, X4 (I0.1, I0.2, I0.4) are no more interrupted by vehicle.

Network 5: Gate close mechanism is controlled by network 5. Network 5 is controlled only by timer T200 (T37).

Network 6: By using network 6 10 second is timed up to ring the doorbell after a vehicle interrupts the outer X1 (I0.1) on inner X2 (I0.2) light barrier sensor. In order to time up timer T201 (T38) the gate should be close. When camera system activated or the button X9 (I0.9) is pushed.

Network 7: This network controls the operation of doorbell. It is activated and inactivated by timer T201 (T38).

Network 8: Network 8 is used to determine the camera stay on active. Timer T202 (T39) is controlled by camera output Y5 (Q0.5).

Network 9: Cameras operation controlled by this network. This network can be activated by push button X7 (I0.7) and can be inactivated by the timer T202 (T39).

Network 10: In the last network we use END instruction to show to PLC that our program is ended here.

5.1.2 Statement List of the PLC Program

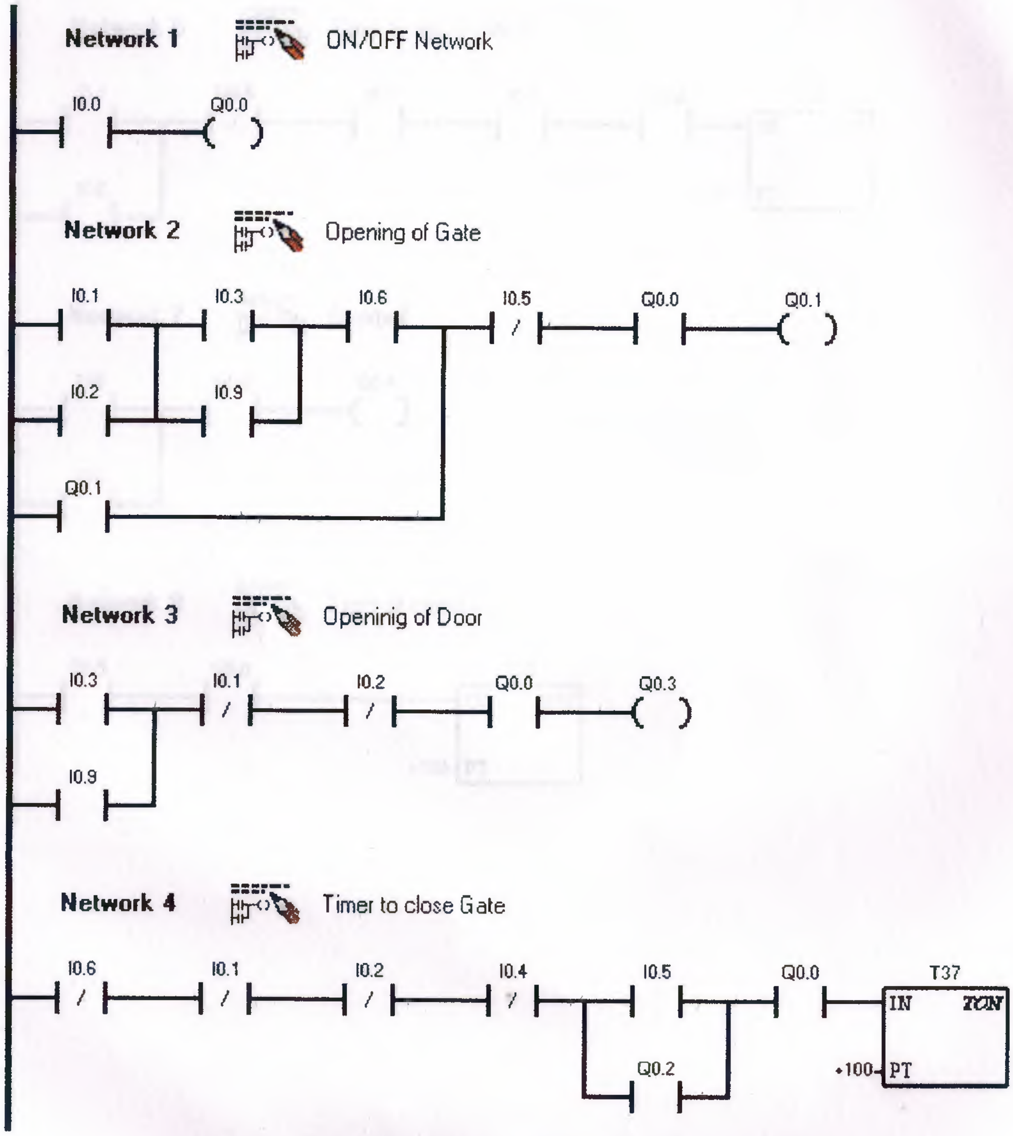
The statement list given below is written for Facon FB series FATEK PLC.

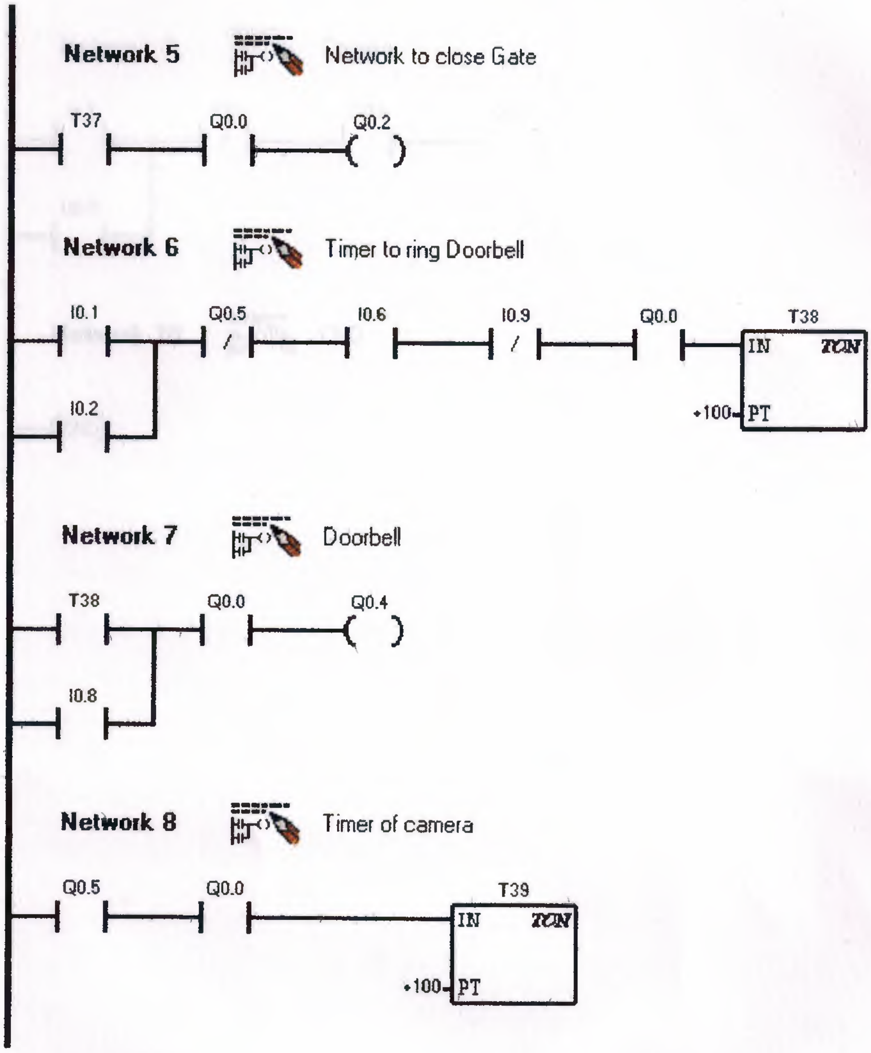
```
//Garden Gate Implementation

NETWORK 1 //ON/OFF Network
ORG        X   0
OUT        Y   0

NETWORK 2 //Opening of Gate
ORG        X   1
OR         X   2
LD         X   3
OR         X   9
ANDLD
AND        X   6
OR         Y   1
AND        X   5
AND        Y   0
OUT        Y   1

NETWORK 3 //Opening of Door
ORG        X   3
OR         X   9
AND NOT    X   1
AND NOT    X   2
AND        Y   0
OUT        Y   3

NETWORK 4 //Timer to close Gate
ORG NOT    X   6
AND NOT    X   1
AND NOT    X   2
AND NOT    X   4
LD         X   5
OR         Y   2
```

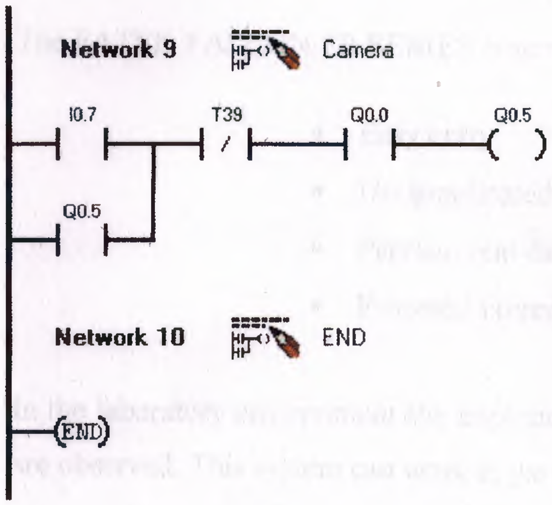
```
ANDLD
AND          Y      0
T200         PV:    10

NETWORK 5 //Network to close Gate
ORG          T200
AND          Y      0
OUT          Y      2

NETWORK 6 //Timer to ring Doorbell
ORG          X      1
OR           X      2
AND NOT      Y      5
AND          X      6
AND NOT      X      9
AND          Y      0
T201         PV:    10

NETWORK 7 //Doorbell
ORG          T201
OR           X      8
AND          Y      0
OUT          Y      4

NETWORK 8 //Timer of camera
ORG          Y      5
AND          Y      0
T202         PV:    10

NETWORK 9 //Camera
ORG          X      7
OR           Y      5
AND NOT      T202
AND          Y      0
OUT          Y      5

NETWORK 10          //END
END
```

5.1.3 Ladder Diagram of the PLC Program

The ladder diagram showed below drawn for siemens PLC.




Network 5
Network to close Gate
T37
Q0.0
Q0.2
Network 6
Timer to ring Doorbell
I0.1
Q0.5
I0.6
I0.9
Q0.0
T38
IN
TON
I0.2
+100
PT
Network 7
Doorbell
T38
Q0.0
Q0.4
I0.8
Network 8
Timer of camera
Q0.5
Q0.0
T39
IN
TON
+100
PT


Network 9
Camera
I0.7
T39
Q0.0
Q0.5
Q0.5
Network 10
END
END

CONCLUSION

The FATEK FALCON FB SERIES is summarized by the following properties:

- Easy entry
- Uncomplicated operation
- Peerless real-time characteristics
- Powerful communications capabilities

In the laboratory environment the implementation is carried out and satisfactory results are observed. This system can work in parallel with a home security system easily.

Alternative of this operation the same logic can be applied to a garage door or to apartment entrance door. Both of them were controlled and successful results were gotten.

Additionally irrigation of the garden by pre-defined time every day, controlling the illumination of garden can be added to the system.

In today's competitive world established control media, including relay, logic and computer systems, can and do provide effective control of industrial process and plant. However each of the above control media has limitations or disadvantages that may often be overcome through the use of Programmable Controller (PLC).

A modern Programmable Logic Controller is a simple control system, which is easy to use and versatile. Most automated factories employ programmable controllers in the control of production and assembly processes.

REFERENCES

[1] SIMATIC S7-200 Programmable Controller, System Manual

[2] Özgür Cemal Özerdem, Programmable Logic Controller and Programming, Near East University Press, Lefkoşa 2002.

[3] http://www.sea.siemens.com/controls/product/s7200/CNs7200.htm

[4] FATEK Falcon FB series Programmable Controller, User's Manual

[5] http://www.fatek.com/downloads/fatekeng.pdf