



Near East University

Faculty of engineering

**Department of Electrical and Electronic
Engineering**

**Automation of a building by
(PLC)**

**Graduation project
EE- 400**

Student: Hassan Kamel Abu Tuaima (970670)

Supervisor: Mr. Özgür C. Özerdem

Lefkoşa-2001



NEU

48 57 57 38 67



CONTENTS

ACKNOWLEDGEMENT.....	i
ABSTRACT.....	ii
INTRODUCTION.....	iii
CHAPTER 1.....	1
Programmable logic controllers.....	1
1.1 Controller.....	2
1.2 Hardware.....	5
1.3 Internal architecture.....	8
CHAPTER 2.....	15
Inputs and Outputs devices.....	15
2.1 Input devices.....	15
2.2 Output devices.....	35
2.3 Examples of application.....	48
CHAPTER 3.....	51
PLCs – hardware design.....	51
3.1 Terminology-PC or PLC.....	51
3.2 Comparison with other control systems.....	51
3.3 PLCs-hardware design.....	53
3.4 Programming PLC.....	63
3.5 PLCs internal operation and signal processing.....	63
3.6 Types of PLC system.....	69
CHAPTER 4.....	76
Programming.....	76
4.1 Latching.....	76
4.2 Multiple outputs.....	77
4.3 Entering ladder programs.....	78
4.4 Instruction lists.....	80

4.5 ICE standards.....	89
4.6 Programming examples.....	90
CHAPTER 5.....	92
Timer and Counter.....	92
5.1 Timers.....	92
5.2 Programming timers.....	94
5.3 Off-delay timer.....	99
5.4 On-shot timers.....	101
5.5 Counters.....	103
5.6 Programming.....	104
5.7 Up and down counting.....	109
5.8 Sequencers.....	110
Automation of PLC program in a building.....	116
CONCLUSION.....	130
REFERENCES.....	131

ACKNOWLEDGEMENT

It's my pleasure, to thanks my supervisor **MR öZGÜR C. öZERDEM** for his cooperation, and making me to think and understand the PLC in a logical way.

Also I would like to represent my family with a great thanks and love for their financial support and for the continuous encouraged me to finish my education and to be an active person in the society especial thanks goes to my father **KAMEL ABOU TUAIMA**.

I would like to give my regards in this time to my fiancée that she was beside me in these years.

I am deeply thank goes to all my professors and instructors those who put me in the right way of thinking and working.

I would like to thank all my friends those who helped me to finish this project especially Mr. Ahmed Abu Hammad, and his brother Kamal, also I would like to thank Mr. Abedelfatah Siam.

Finally thanks all who are intrusting in PLC and read my project.

ABSTRACT

This project is generally about PLC. It has two parts.

Firstly, its has five chapters where, the first one gives a good explanation about programmable logic controlled (PLC), the second one has briefly explanation about working of the sensors, the third one has explained the PLC design and it's structure, the fourth one give explanation about programming logic and the last one give some explanation for timer and counter.

Secondly, the main program of automation a building by PLC, which gives some programs that control some application like illumination, pumping water from the basement to the roof, air conditioning, modern alarm system and heating the water from the roof.

INTRODUCTION

A programmable logic controller (PLC) is a device that was invented to replace the necessary sequential relay circuits for machine control. The PLC works by looking at its inputs and depending upon their state, turning on/off its outputs. The user enters a program, usually via software, that gives the desired results.

PLC's are used in many real world applications. If there is industry present, chances are good that there is a PLC present. If you are involved in a machining, packaging, material handling, automated assembly or countless other industries you are probably already using them. If you are not, you are wasting money and time. Almost any application that needs some type of electrical control has for a PLC.

For example, let's assume that when a switch turns on we want turn a solenoid on for 5 seconds and then turn it off regardless of how long the switch is on for. We can do this with a simple external timer. But what if the process included 10 switches and solenoids? We would need 10 external timers. What if the process also needed to count how many times the switches individually turned on? We need a lot of external counters.

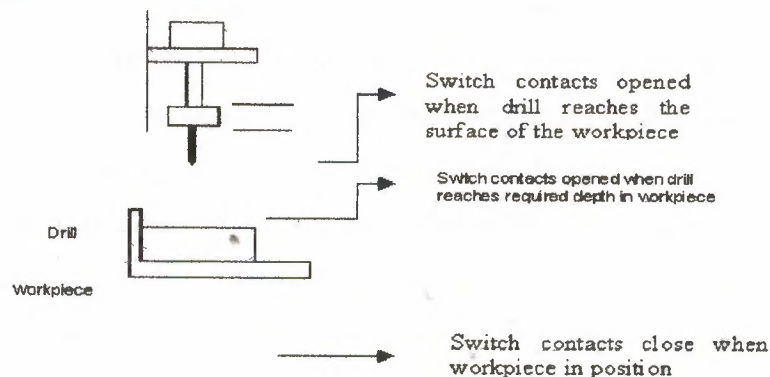
As you can see the bigger the process the more of a need we have for a PLC. We can simple program the PLC to count its inputs and turn the solenoids on for the specified time. This site gives enough information to be able to write programs far more complicated than the simply one above. We will take a look at what is considered to be the 'top 20' PLC instructions. It can safely estimated that with a firm understanding of these instructions one can solve more than 80% of the applications inexistence.

Chapter 1

PROGRAMMABLE LOGIC CONTROLLERS

This chapter is an introduction to the programmable logic controller, its general function, hardware forms and internal architecture. This overview is followed up by more detailed discussion in the following chapters.

What type of task might a control system have? It might be required to control a sequence of events or maintain some variable constant or follow some prescribed change. For example, the control system for an automatic drilling machine (Figure 1.1(a)) might be required to start lowering the drill when the work piece is in position, start drilling when the drill reaches the work piece, stop drilling when the drill has produced the required depth of hole, retract the drill and then switch off and wait for the next work piece to be put in position before repeating the operation. Another control system (Figure 1.1(b)) might be used to control the number of items moving along a conveyor belt and direct them into a packing case. The inputs to such control systems might be from switches being closed or opened, e.g. the presence of the work piece might be indicated by it moving against a switch and closing it, or other sensors such as those used for temperature or flow rates. The controller might be required to run a motor to move an object to some position, or to turn a valve, or perhaps a heater, on or off.



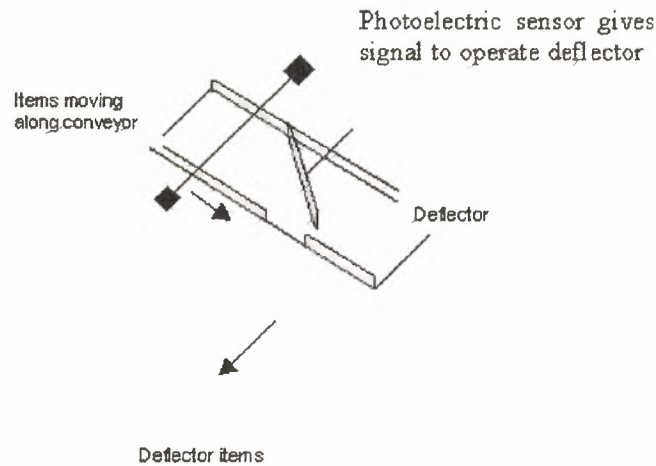


Figure 1.1 an example of a control task and some input sensors, (a) an automatic drilling machine, (b) a packing system

1.1 Controller

What form might a controller have? For the automatic drilling machine, we could wire up electrical circuits in which the closing or opening of switches would result in motors being switched on or valves being actuated. Thus we might have the closing of a switch activating a relay, which, in turn, switches on the current to a motor and causes the drill to rotate (Figure 1.2). Another switch might be used to activate a relay and switch on the current to a pneumatic or hydraulic valve which results in pressure being switched to drive a piston in a cylinder and so results in the work piece being pushed into the required position. Such electrical circuits would have to be specific to the automatic drilling machine. For controlling the number of items packed into a packing case we could likewise wire up electrical circuits involving sensors and motors. However, the controller circuits we devised for these two situations would be different. In the 'traditional' form of control system, the rules governing the control system and when actions are initiated are determined by the wiring. When the rules used for the control actions are changed, the wiring has to be changed.

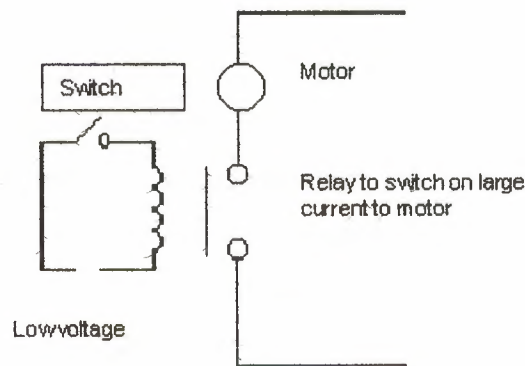


Figure 1.2 a control circuit

1.1.1 Microprocessor controlled system

Instead of hardwiring each control circuit for each control situation we can use the same basic system for all situations if we use a microprocessor-based system and write a program to instruct the microprocessor how to react to each input signal from, say, switches and give the required outputs to, say, motors and valves. Thus we might have a program of the form:

If switch A closes

Output to motor circuit

If switch B closes

Output to valve circuit

By changing the instructions in the program we can use the same microprocessor system to control a wide variety of situations. As an illustration, the modern domestic washing machine uses a microprocessor system. Inputs to it arise from the dials used to select the required wash cycle, a switch to determine that the machine door is closed, a temperature sensor to determine the temperature of the water and a switch to detect the level of the water. On the basis of these inputs the microprocessor is programmed to give outputs which switch on the drum motor and control its speed, open or close cold and hot water valves, switch on the drain pump, control the water heater and control the door lock so that the machine cannot be opened until the washing cycle is completed.

1.1.2 The Programmable Logic Controller

A programmable logic controller (PLC) is a special form of microprocessor-based controller that uses a programmable memory to store instructions and to implement functions such as logic, sequencing, timing, counting and arithmetic in order to control machines and processes (Figure 1.3) and are designed to be operated by engineers with perhaps a limited knowledge of computers and computing languages. They are not designed so that only computer programmers can set up or change the programs. Thus, the designers of the PLC have pre-programmed it so that the control program can be entered using a simple, rather intuitive, form of language. The term logic is used because programming is primarily concerned with implementing logic and switching operations, e.g. if A or B occurs switch on C, if A and B occurs switch on D. Input devices, e.g. sensors such as switches, and output devices in the system being controlled, e.g. motors, valves, etc., are connected to the PLC. The operator then enters a sequence of instructions, i.e. a program, into the memory of the PLC. The controller then monitors the inputs and outputs according to this program and carries out the control rules for which it has been programmed.

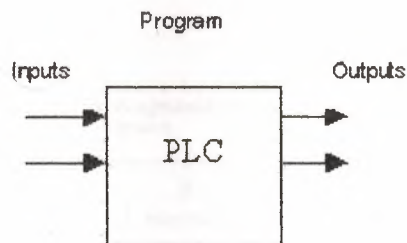


Figure 1.3 a programmable logic controller

PLCs have the great advantage that the same basic controller can be used with a wide range of control systems. To modify a control system and the rules that are to be used, all that is necessary is for an operator to key in a different set of instructions. There is no need to rewire. The result is a flexible, cost effective, system, which can be used with control systems, which vary quite widely in their nature and complexity.

PLCs are similar to computers but whereas computers are optimized for calculation and display tasks, PLCs are optimized for control tasks and the industrial environment. Thus PLCs are:

1 Rugged and designed to withstand vibrations, temperature, humidity and noise.

2 Have interfacing for inputs and outputs already inside the controller.

3 Are easily programmed and have an easily understood programming language which is primarily concerned with logic and switching operations.

The first PLC was developed in 1969. They are now widely used and extend from small self-contained units for use with perhaps 20 digital inputs/outputs to modular systems which can be used for large numbers of inputs/outputs, handle digital or analogue inputs/outputs, and also carry out proportional-integral-derivative control modes.

1.2 Hardware

Typically a PLC system has five basic components. These are the processor unit, memory, the power supply unit, input/output interface section and the programming device. Figure 1.4 shows the basic arrangement.

The processor unit or central processing unit (CPU) is the unit containing the microprocessor and this interprets the input signals and carries out the control actions, according to the program stored in its memory, communicating the decisions as action signals to the outputs.

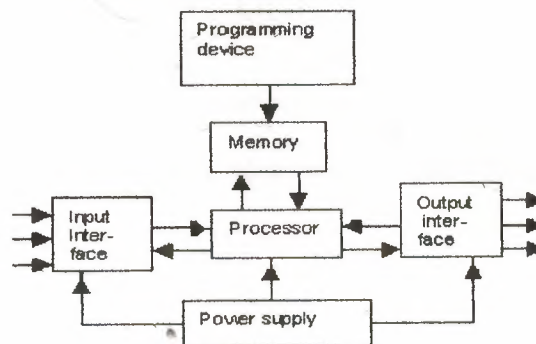


Figure 1.4 The PLC system

The power supply unit is needed to convert the mains a.c. voltage to the low d.c. voltage (5V) necessary for the processor and the circuits in the input and output interface modules.

The programming device is used to enter the required program into the memory of the processor. The program is developed in the device and then transferred to the memory unit of the PLC.

The memory unit is where the program is stored that is to be used for the control actions to be exercised by the microprocessor.

The input and output sections are where the processor receives information from external devices and communicates information to external devices. The inputs might thus be from switches, as illustrated in Figure 1.1(a) with the automatic drill, or other sensors such as photo- electric cells, as in the counter mechanism in Figure 1.1(b), temperature sensors, or flow sensors, etc. The outputs might be to motor starter coils, solenoid valves, etc. Input and output devices can be classified as giving signals which are discrete, digital or analogue (Figure 1.5). Devices giving discrete or digital signals are ones where the signals are. Thus a switch is a device giving a discrete signal, either no voltage or a voltage. *Digital* devices can be considered to be essentially discrete devices, which give a sequence of on_off signals. Analogue devices give signals whose size is proportional to the size of the variable being monitored. For example, a temperature sensor may give a voltage proportional to the temperature.

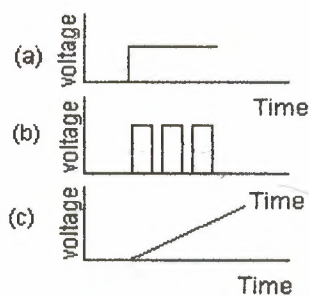


Figure 1.5 Signales: (a) discrete (b)digital (c)analogue

1.2.1 Mechanical design of PLC systems

There are two common types of mechanical design for PLC systems; a single box, and the modular and rack types. The single box type is commonly used for small programmable controllers and is supplied as an integral compact package complete with power supply, processor, memory, and input/output units (Figure 1.6(a)). Typically such a PLC might have 40 input/output points and a memory, which can store some 300 to 1000 instructions. The modular type consists of separate modules for power supply, processor, etc. which are often mounted on rails within a metal cabinet. The rack type can be used for all sizes of programmable controllers and has the various functional units packaged in individual modules, which can be plugged into sockets in a base rack (Figure 1.6(b)). The user and the appropriate ones then plugged into the rack decide the mix of modules required for a particular purpose. Thus it is comparatively easy to expand the number of input/output connections by just adding more input/output modules or to expand the memory by adding more memory units.

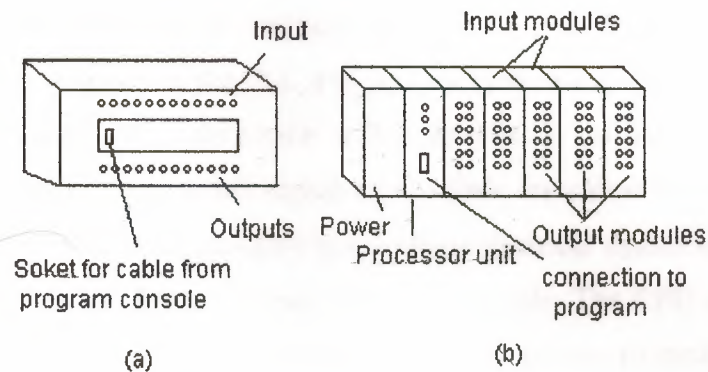


Figure 1.6 (a) single box, (b) modular/rail type

Programs are entered into a PLC's memory using a program device, which is usually not permanently connected to a particular PLC and can be moved from one controller to the next without disturbing operations. For the operation of the PLC it is not necessary for the programming device to be connected to the PLC since it transfers the program to the PLC memory. Programming devices can be a hand-held device, a desktop console or a computer. Hand-held systems incorporate a small keyboard and liquid crystal display, Figure 1.7 showing a typical form. Desktop devices are likely to have a visual display unit with a full keyboard and screen display. Personal computers are widely configured as program development workstations. Some PLCs only require the computer to have appropriate software, others special communication cards to interface with the PLC. A

major advantage of using a computer is that the program can be stored on the hard disk or a floppy disk and copies easily made. The disadvantage is that the programming often tends to be not so user-friendly. Hand-held programming consoles will normally contain enough memory to allow the unit to retain programs while being carried from one place to another. Only when the program has been designed on the programming device and is ready is it transferred to the memory unit of the PLC.

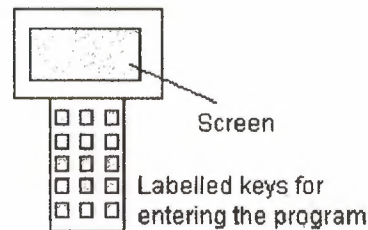


Figure 1.7 Hand-held programmer

1.3 Internal architecture

Figure 1.8 shows the basic internal architecture of a PLC. It consists of a central processing unit (CPU) containing the system microprocessor, memory, and input/output circuitry. The CPU controls and processes all the operations within the PLC. It is supplied with a clock with a frequency of typically between 1 and 8 MHz. This frequency determines the operating speed of the PLC and provides the timing and synchronization for all elements in the system. The information within the PLC is carried by means of digital signals. The internal paths along which digital signals flow are called buses. In the physical sense, a bus is just a number of conductors along which electrical signals can flow. It might be tracks on a printed circuit board or wires in a ribbon cable. The CPU uses the data bus for sending data between the constituent elements; the address bus to send the addresses of locations for accessing stored data and the control bus for signals relating to internal control actions. The system bus is used for communications between the input/output ports and the input/output unit.

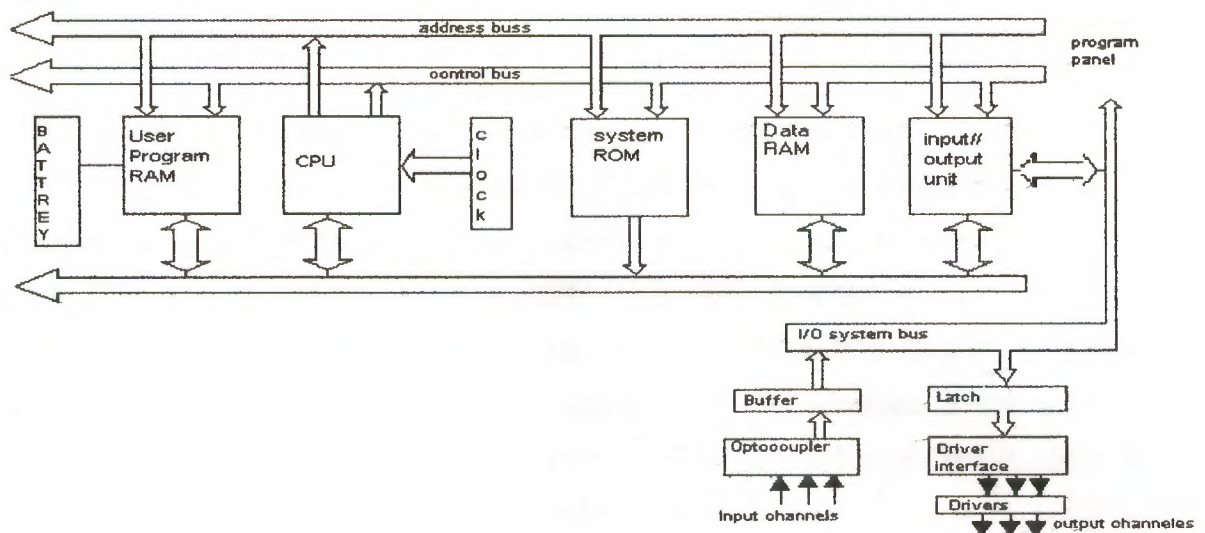


Figure 1.8 Architecture of a PLC

1.3.1 The CPU

The internal structure of the CPU depends on the microprocessor concerned. In general they have:

- 1 An arithmetic and logic unit (ALU) which is responsible for data manipulation and carrying out arithmetic operations of addition and subtraction and logic operations of AND, OR, NOT and EXCLUSIVE OR.
- 2 Memory, termed registers, located within the microprocessor and used to store information involved in program execution.
- 3 A control unit, which is used to control the timing of operations.

1.3.2 The buses

The buses are the paths used for communication within the PLC. The information is transmitted in binary form, i.e. as a group of bits with a bit being a binary digit of 1 or 0, i.e. on/off states. The term word is used for the group of bits constituting some information. Thus an 8-bit word might be the binary number 00100110. Each of the bits is communicated simultaneously along its own parallel wire. The system has four buses:

- 1 The data bus carries the data used in the processing carried out by the CPU. A microprocessor termed as being 8-bit has an internal data bus, which can handle 8-bit

numbers. It can thus perform operations between 8-bit numbers and deliver results as S-bit values.

2 The address bus is used to carry the addresses of memory locations. So that each word can be located in the memory, every memory location is given a unique address. Just like houses in a town are each given a distinct address so that they can be located. So each word location is given an address so that the CPU can access data stored at a particular location either to read data located there or put, i.e. write, data there. It is the address bus, which carries the information indicating which address is to be accessed. If the address bus consists of 8 lines, the number of 8-bit words, and hence number of distinct addresses, is $2^8 = 256$. With 16 address lines, 65 536 addresses are possible.

3 The control bus carries the signals used by the CPU for control, e.g. to inform memory devices whether they are to receive data from an input or output data and to carry timing signals used to synchronize actions.

4 The system bus is used for communications between the input/output ports and the input/output unit

1.3.3 Memory

There are several memory elements in a PLC system

1 System read-only-memory (ROM) to give permanent storage for the operating system and fixed data used by the CPU.

2 Random-access-memory (RAM) for the user's program.

3 Random-access-memory (RAM) for data. This is where information is stored on the status of input and output devices and the values of timers and counters and other internal devices. The data RAM is sometimes referred to as a *data table* or *register table*. Part of this memory, i.e. a block of addresses, will be set aside for input and output addresses and the states of those inputs and outputs. Part will be set aside for preset data and part for storing counter values, timer values, etc.

4 Possibly, as a bolt-on extra module, erasable and programmable read-only-memory (EPROM) for ROMs that can be programmed and then the program made permanent.

1.3.4 Input/output unit

The input/output unit provides the interface between the system and the outside world, allowing for connections to be made through input/output channels to input devices such as sensors and output devices such as motors and solenoids. It is also through the input/output unit that programs are entered from a program panel. Every input/output point has a unique address, which can be used by the CPU.

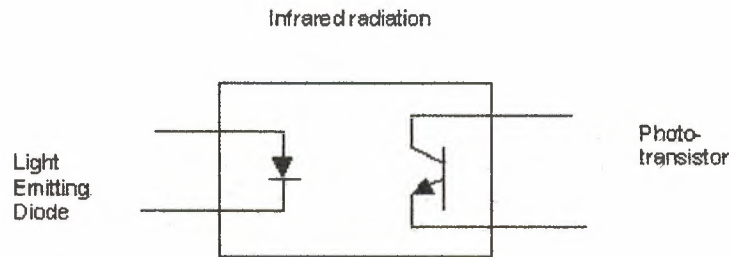


Figure 1.19 Optoisolator

The input/output channels provide isolation and signal conditioning functions so that sensors and actuators can often be directly connected to them without the need for other circuitry. Electrical isolation from the external world is usually by means of optoisolators (the term optocoupler is also often used). Figure 1.9 shows the principle of an optoisolator.

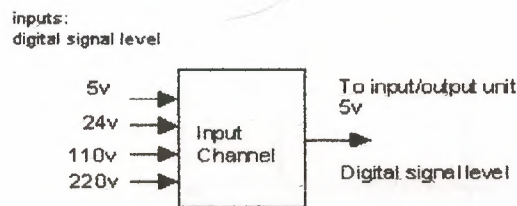


Figure 1.10 Input levels

When a digital pulse passes through the light-emitting diode, a pulse of infrared radiation is produced. This pulse is detected by the phototransistor and gives rise to a voltage in that circuit. The gap between the light-emitting diode and the phototransistor gives electrical isolation but the arrangement still allows for a digital pulse in one circuit to give rise to a digital pulse in another circuit. The digital signal that is generally compatible with the microprocessor in the PLC is 5 V d.c. However, signal conditioning in the input channel,

with isolation, enables a wide range of input signals to be supplied to it. A range of inputs might be available with a larger PLC, e.g. 5 V, 24 V, 110 V and 240 V digital/discrete, i.e. on-off, signals (Figure 1.10). A small PLC is likely to have just one form of input, e.g. 24 V. Figure 1.11 shows the basic form a d.c. input channel might take. Outputs are often specified as being of relay type, transistor type or triac type.

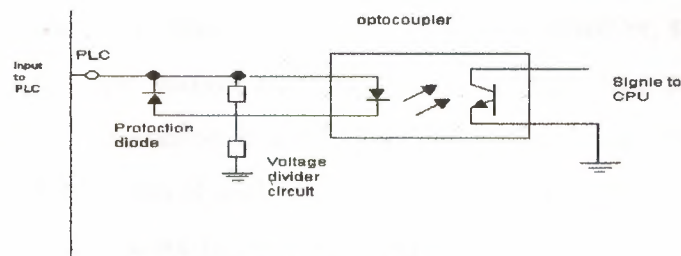


Figure 1.11 Basic d.c. input circuit

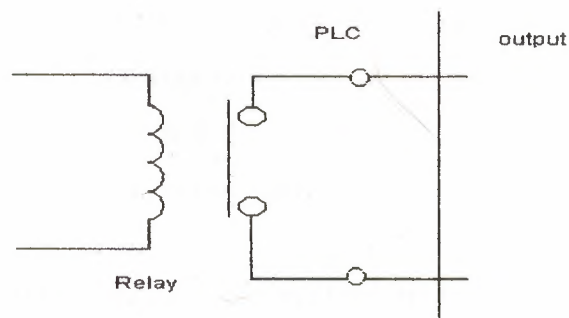


Figure 1.12 relay output

1 With the relay type, the signal from the PLC output is used to operate a relay and so is able to switch currents of the order of a few amperes in an external circuit. The relay not only allows small currents to switch much larger currents but also isolates the PLC from the external circuit. Relays are, however, relatively slow to operate. Relay outputs are suitable for ac. and d.c. switching. They can withstand high surge currents and voltage transients. Figure 1.12 shows the basic feature of a relay output.

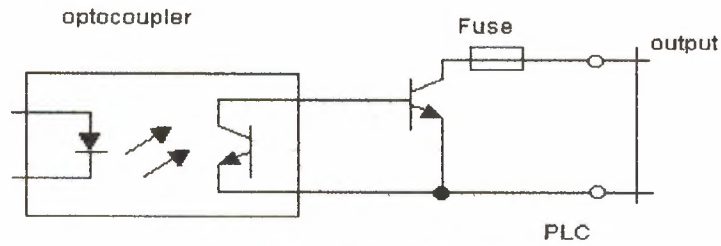


Figure 1.13 Basic form of transistor output

2 The transistor type of output uses a transistor to switch current through the external circuit. This gives a considerably faster switching action. It is, however, strictly for d.c. switching and is destroyed by over current and high reverse voltage. As a protection, either a fuse or built-in electronic protection is used. Optoisolators are used to provide isolation. Figure 1.13 shows the basic form of such a transistor output channel.

3 Triac outputs, with optoisolators for isolation, can be used to control external loads, which are connected to the ac. power supply. It is strictly for ac. operation and is very easily destroyed by over current. Fuses are virtually always included to protect such outputs.

The output from the input/output unit will be digital with a level of 5 V. However, after signal conditioning with relays, transistors or triacs, the output from the output channel might be a 24 V, 100 mA switching signal, a d.c. voltage of 110 V, 1 A or perhaps 240 V, 1 A a.c. or 240 V, 2 A a.c. from a triac output channel (Figure 1.14). With a small PLC, all the outputs might be of one type, e.g. 240 V a.c., 1 A. With modular PLCs, however, a range of outputs can be accommodated by selection of the modules to be used.

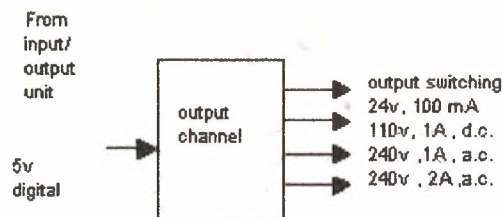


Figure 1.14 Output levels

The following illustrates the types of inputs and outputs available with a small PLC, one of the Mitsubishi F2 series:

Number of inputs 12

Number of outputs 8

Input specification:

Chapter 2

INPUTS AND OUTPUTS DEVICES

2.1 Input devices

Sensors which give digital/discrete, i.e. on-off, outputs can be easily connected to the input ports of PLCs. Sensors, which give analogue signals, have to be converted to digital signals before inputting them to PLC ports. The following are examples of some of the commonly used sensors.

2.1.1 Mechanical switches

A mechanical switch generates an on-off signal or signals as a result of some mechanical input causing the switch to open or close. Such a switch might be used to indicate the presence of a work piece on a machining table, the work piece pressing against the switch and so closing it. The switch being open and its presence indicate the absence of the workpiece by it being closed. Thus, with the arrangement shown in Figure 2.1(a), the input signals to a single input channel of the PLC are thus the logic levels:

Work piece not present 0

Work piece present 1

The 1 level might correspond to a 24 V d.c. Input, the 0 to a 0 V input. With the arrangement shown in Figure 2.1(b), when the switch is open the supply voltage is applied to the PLC input, when the switch is closed channel the input voltage drops to a low value. The logic levels are thus:

Supply Work piece not present 1

Work piece present 0

Switches are available with normally open (NO) or normally closed (NC) contacts or can be configured as either by choice of the relevant contacts. A

switch has its contacts open in the absence of a mechanical input and the mechanical input is used to close the switch. An NC switch has its contacts closed in the absence of a mechanical input and the mechanical input is used to open the switch.

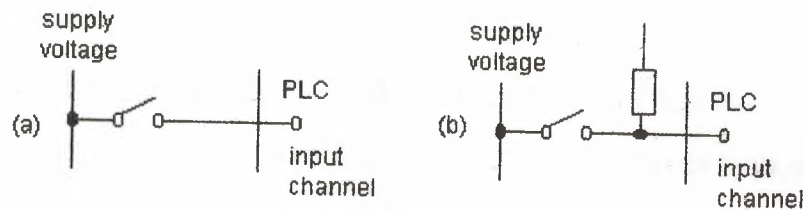


figure 2.1 Switch sensor

The term limit switch is used for a switch, which is used to detect the presence or passage of a moving part. It can be actuated by a cam, roller or lever. Figure 2.2 shows some examples. The cam (Figure 2.2cc)) can be rotated at a constant rate and so switch the switch on and off for particular time intervals.

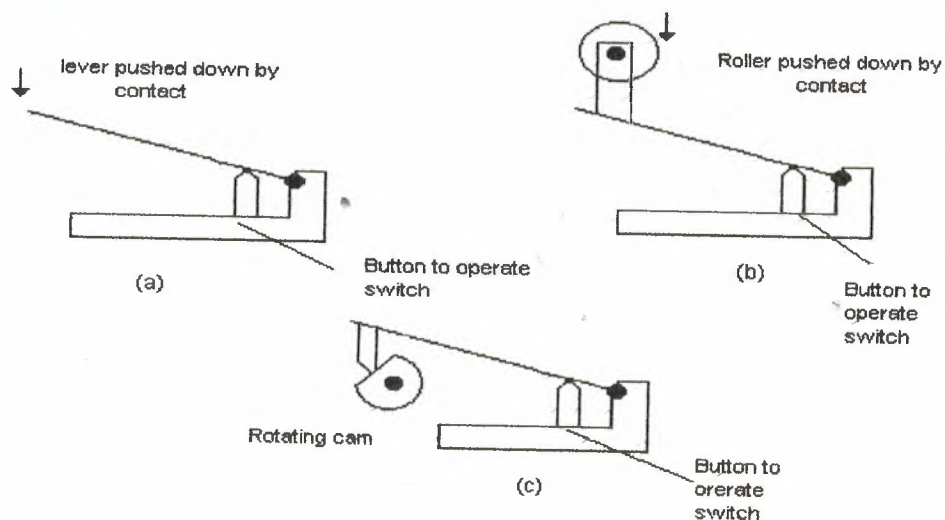


Figure 2.2 Limit switches actuated by: (a) lever, (b) roller, (c) cam

2.1.2 Proximity switches

Proximity switches are used to detect the presence of an item without making contact with it. There are a number of forms of such switches, some being only suitable for metallic objects.

The eddy current type of proximity switch has a coil, which is energized by a constant alternating current and produces a constant alternating magnetic field. When a metallic object is close to it, eddy currents are induced in it (Figure 2.3). The magnetic field due to these eddy currents induces an e.m.f. back in the coil with the result that the voltage amplitude needed to maintain the constant coil current changes. The voltage amplitude is thus a measure of the proximity of metallic objects. The voltage can be used to activate an electronic switch circuit, basically a transistor which has its output switched from low to high by the voltage change, and so give an on-off device. The range over which such objects can be detected is typically about 0.5 to 20 mm. Another type, the inductive proximity switch, consists of a coil wound round a ferrous metallic core. When one end of this core is placed near to a ferrous metal object there is effectively a change in the amount of metallic core associated with the coil and so a change in its inductance. This change in inductance can be monitored using a resonant circuit, the presence of the ferrous metal object thus changing the current in that circuit. The current can be used to activate an electronic switch circuit and so give an on-off device. The range over which such objects can be detected is typically about 2 to 15 mm.

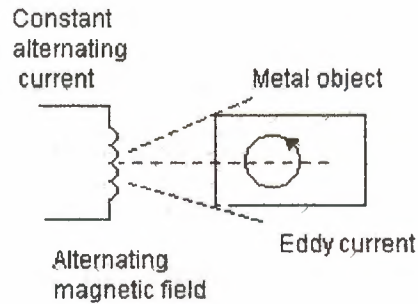


Figure 2.3 Eddy current proximity switch

Another type is the reed switch. This consists of two overlapping, but not touching, strips of a springy ferromagnetic material sealed in a glass or plastic envelope (Figure 2.4). When a magnet or current carrying coil is brought close to the switch, the strips become magnetized and attract each other. The contacts then close. The magnet closes the contacts when it is typically about 1 mm from the switch. Such a switch is widely used with burglar alarms to detect when a door is opened; the magnet being in the door and the reed switch in the frame of the door. When the door opens the switch opens.

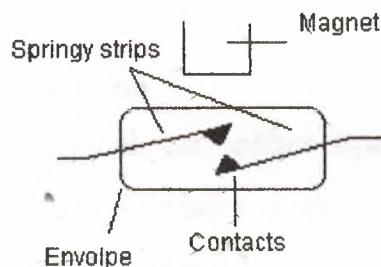


Figure 2.4 Reed switch

A proximity switch that can be used with metallic and non-metallic objects is the capacitive proximity switch. The capacitance of a pair of plates separated by some distance depends on the separation, the smaller the separation the

higher the capacitance. The sensor of the capacitive proximity switch is just one of the plates of the capacitor, the other plate being the metal object whose proximity is to be detected (Figure 2.5). Thus the proximity of the object is detected by a change in capacitance. The sensor can also be used to detect non-metallic objects since the capacitance of a capacitor depends on the dielectric between its plates. In this case the plates are the sensor and the earth and the non-metallic object is the dielectric. The change in capacitance can be used to activate an electronic switch circuit and so give an on-off device. Capacitive proximity switches can be used to detect objects when they are typically between 4 and 60 mm from the sensor head.

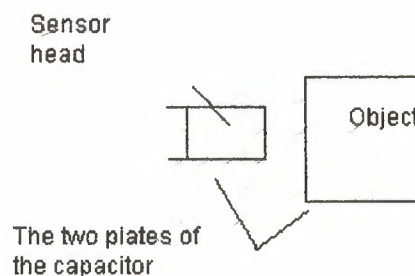


Figure 2.5 capacitive proximity switch

2.1.3 Photoelectric sensors and switches

Photoelectric switch devices can either operate as transmissive types where the object being detected breaks a beam of light, usually infrared radiation, and stops it reaching the detector (Figure 2.6(a)) or reflective types where the object being detected reflects a beam of light onto the detector (Figure 2.6(b)). In both types the radiation emitter is generally a light-emitting diode (LED). The radiation detector might be a phototransistor, often a pair of transistors, known as a Darlington pair. The Darlington pair increases the sensitivity. Depending on the circuit used, the output can be made to switch to

either high or low when light strikes the transistor. Such sensors are supplied as packages for sensing the presence of objects at close range, typically at less than about 5 mm. Figure 2.6(c) shows a U-shaped form where the object breaks the light beam.

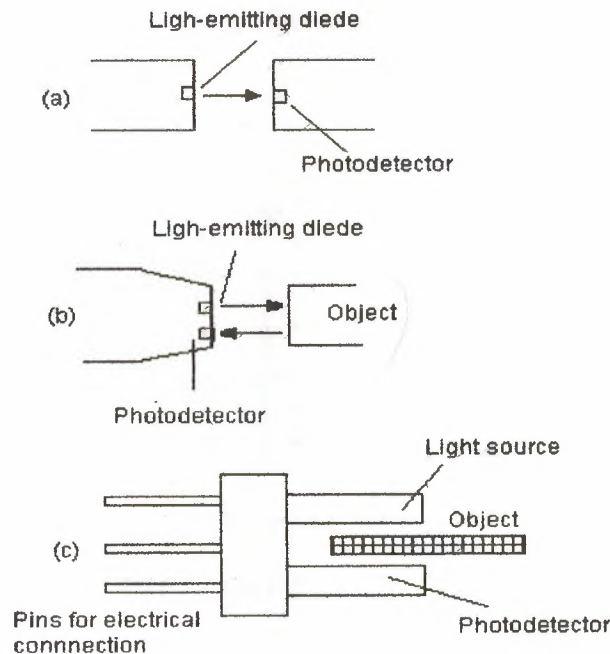


Figure 2.6 Photoelectric sensors

Another possibility is a photo diode. Depending on the circuit used, the output can be made to switch to either high or low when light strikes the diode. Yet another possibility is a photoconductive cell. The resistance of the photoconductive cell, often cadmium sulphide, depends on the intensity of the light falling on it.

With the above sensors, light is converted to a current, voltage or resistance change. If the output is to be used as a measure of the intensity of the light, rather than just the presence or absence of some object in the light path, the

signal will need amplification and then conversion from analogue to digital by an analogue-to-digital converter. An alternative to this is to use a light-to-frequency converter, the light then being converted to a sequence of pulses with the frequency of the pulses being a measure of the light intensity. Integrated circuit sensors are available, e.g. the Texas Instrument TSL220, incorporating the light sensor and the voltage-to-frequency converter (Figure 2.7).

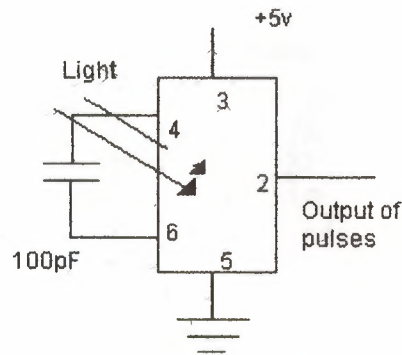


Figure 2.7 TSL220

2.1.4 Encoders

The term encoder is used for a device that provides a digital output as a result of angular or linear displacement. An increment encoder detects changes in angular or linear displacement from some datum position, while an absolute encoder gives the actual angular or linear position.

Figure 2.8 shows the basic form of an incremental encoder for the measurement of angular displacement. A beam of light, from perhaps a light-emitting diode (LED), passes through slots in a disc and is detected by a light sensor, e.g. a photo diode or photo transistor. When the disc rotates, the light

beam is alternately transmitted and stopped and so a pulsed output is produced from the light sensor. The number of pulses is proportional to the angle through which the disc has rotated, the resolution being proportional to the number of slots on a disc. With 60 slots then, since one revolution is a rotation of 360°, a movement from one slot to the next is a rotation of 6°. By using offset slots it is possible to have over a thousand slots for one revolution and so much higher resolution.

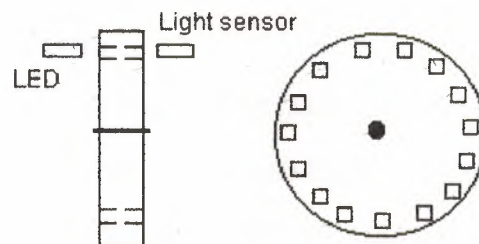


Figure 2.8 Basic form of an incremental encoder

The absolute encoder differs from the incremental encoder in having a pattern of slots, which uniquely defines each angular position. Figure 2.9 shows the form of such an encoder using three sets of slots and so giving a 3-bit output. Typical encoders tend to have up to 10 or 12 tracks. The number of bits in the resulting binary output is equal to the number of tracks. Thus with 3 tracks there will be 3 bits and so the number of positions that can be detected is $2^3 = 8$, i.e. a resolution of $360/8 = 45$ degree. With 10 tracks there will be 10 bits and the number of positions that can be detected is $2^{10} = 1024$ and the angular resolution is $360/1024 = 0.35$.

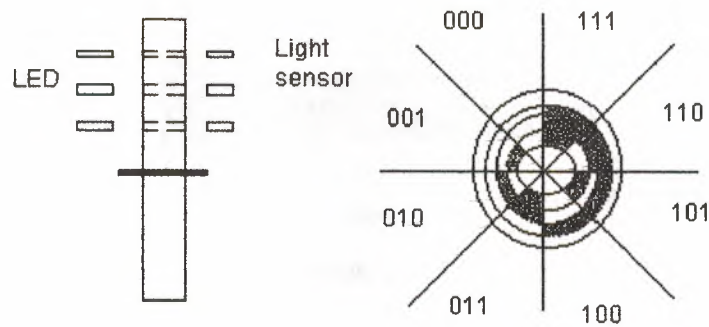


Figure 2.9 A 3-bit absolute encoder

2.1.5 Temperature sensors

A simple form of temperature sensor, which can be used to provide an on-off signal when a particular temperature is reached, is the bimetal element. This consists of two strips of different metals, e.g. brass and iron, bonded together (Figure 2.10). The two metals have different coefficients of expansion. Thus when the temperature of the bimetal strip increases the strip curves, in order that one of the metals can expand more than the other. The higher expansion metal is on the outside of the curve. As the strip cools, the bending effect is reversed. This movement of the strip can be used to make or break electrical contacts and hence, at some particular temperature, give an on-off current in an electrical circuit. The device is not very accurate but is commonly used in domestic central heating thermostats.

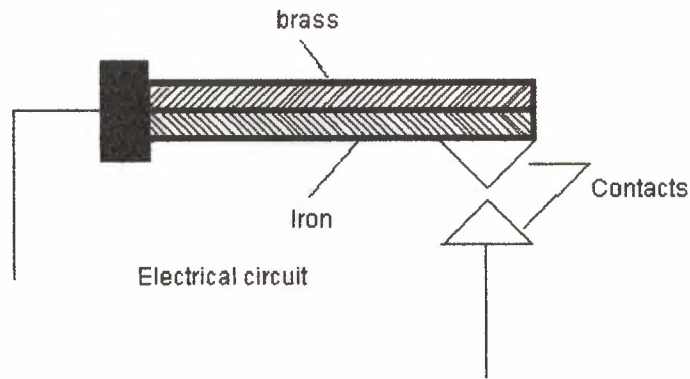


Figure 2.10 Bimetallic strip

Another form of temperature sensor is the resistive temperature detector (RTD). The electrical resistance of metals or semiconductors changes with temperature. In the case of a metal, the ones most commonly used are platinum, nickel or nickel alloys, the resistance of which varies in a linear manner with temperature over a wide range of temperatures, though the actual change in resistance per degree is fairly small. Semiconductors, such as thermistors, show very large changes in resistance with temperature. The change, however, is non-linear. Such detectors can be used as one arm of a Wheatstone bridge and the output of the bridge taken as a measure of the temperature (Figure 2.11(a)). Another possibility is to use a potential divider circuit with the change in resistance of the thermistor changing the voltage drop across a resistor (Figure 2.11(b)). The output from either type of circuit is an analogue signal which is a measure of the temperature.

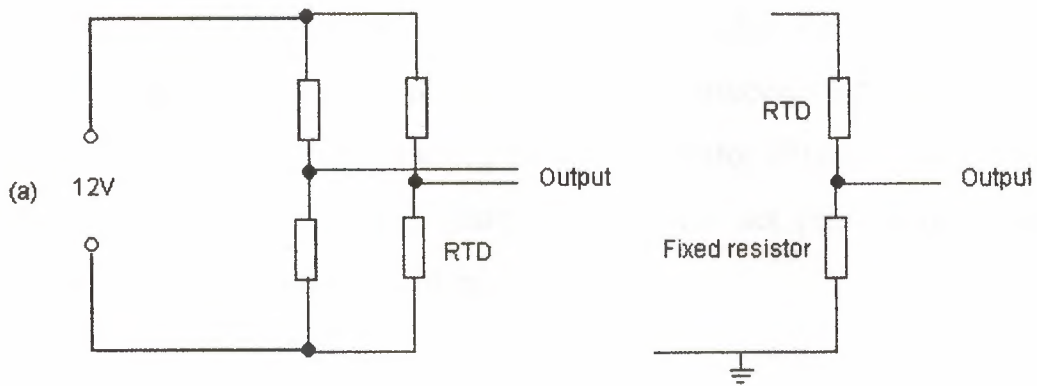


Figure 2.11 a) Wheatstone bridge, b) potential divider circuits

Thermodiodes and thermotransistors are used as temperature sensors since the temperature affects the rate at which electrons and holes diffuse across semiconductor junctions. Integrated circuits are available which combine such a temperature sensitive element with the relevant circuitry to give an output voltage related to temperature. A widely used integrated package is the LM35, which gives an output of $10 \text{ mV}/^\circ\text{C}$ when the supply voltage is $+5 \text{ V}$ (Figure 2.12).

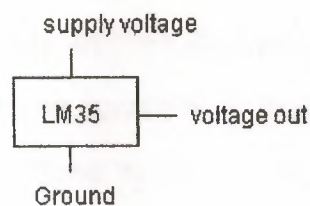


Figure 2.12 LM35

A digital temperature switch can be produced with an analogue sensor by feeding the analogue output into a comparator amplifier which compares it with some set value, producing an output giving a logic 1 signal when the temperature voltage input is equal to or greater than the set point and otherwise an output which gives a logic 0 signal. Integrated circuits, e.g.

LM3911N, are available combining a thermotransistor temperature sensitive element with an operational amplifier. When the connections to the chip are so made that the amplifier is connected as a comparator (Figure 2.13), then the output will switch as the temperature traverses the set point and so directly give an on-off temperature controller.

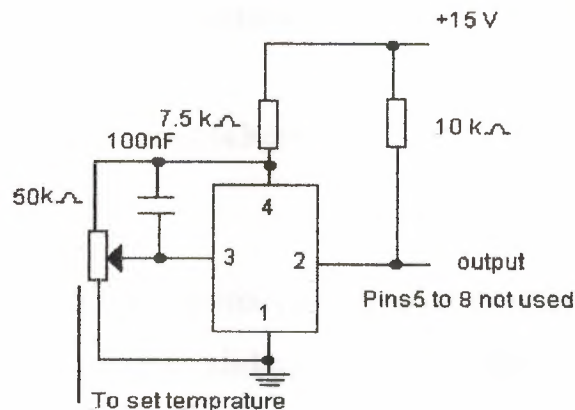


Figure 2.13 LM 11N circuit for on-off control

Another commonly used temperature sensor is the thermocouple. The thermocouple consists essentially of two dissimilar wires A and B forming a junction (Figure 2.14). When the junction is heated so that it is at a higher temperature than the other junctions in the circuit, which remain at a constant cold temperature, an e.m.f is produced which is related to the hot junction temperature. The voltage produced by a thermocouple is small and needs amplification before it can be fed to the analogue channel input of a PLC. There is also circuitry required to compensate for the temperature of the cold junction since its temperature affects the value of the e.m.f given by the hot junction. The amplification and compensation, together with filters to reduce

the effect of interference from the 50 Hz mains supply, are often combined in a signal-processing unit.

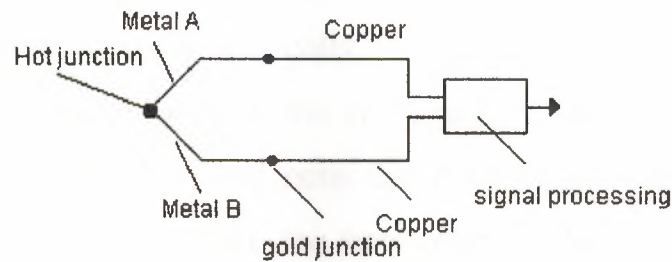


Figure 2.14 thermocouple

2.1.6 Displacement sensors

A linear or rotary potentiometer can be used to provide a voltage signal related to the position of the sliding contact between the ends of the potentiometer resistance track (Figure 2.15). The potentiometer thus provides an analogue linear or angular position sensor.

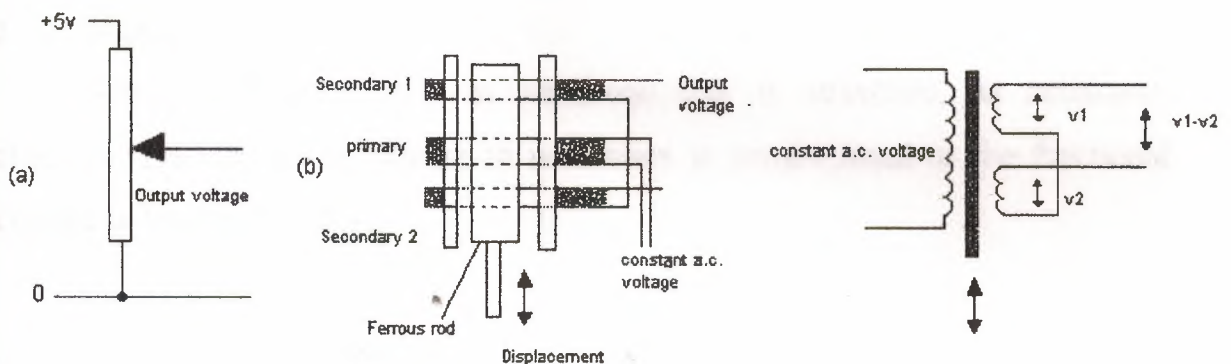


Figure 2.15 a) Potentiometer b) LVDT

Another form of displacement sensor is the linear variable differential transformer (LVDT), this giving a voltage output related to the position of a

ferrous rod. The LVDT consists of three symmetrically placed coils through which the ferrous rod moves (Figure 2.16).

When an alternating current is applied to the primary coil, alternating voltages are induced in the two secondary coils. When the ferrous rod core is centered between the two secondary coils, the voltages induced in them are equal. The outputs from the two secondary coils are connected so that their combined output is the difference between the two voltages. With the rod central, the two alternating voltages are equal and so there is no output voltage. When the rod is displaced from its central position there is more of the rod in one secondary coil than the other. As a result the size of the alternating voltage induced in one coil is greater than that in the other. The difference between the two secondary coil voltages, i.e. the output, thus depends on the position of the ferrous rod. The output from the LVDT is an alternating voltage. This is usually converted to an analogue d.c. voltage and amplified before inputting to the analogue channel of a PLC.

2.1.7 Strain gauges

When a wire or strip of semiconductor is stretched, its resistance changes. The fractional change in resistance is proportional to the fractional change in length, i.e. strain.

$$\Delta R/R = G * \text{Strain}$$

Where ΔR is the change in resistance for a wire of resistance R and G is a constant called the gauge factor. For metals the gauge factor is about 2 and for semiconductors about 100. Metal resistance strain gauges are in the form of a flat coil in order to get a reasonable length of metal in a small area.

Often they are etched from metal foil (Figure 2.16) and attached to a backing of thin plastic film so that they can be stuck on surfaces, like postage stamps on an envelope.



Figure 2.16 Metal foil strain gauges

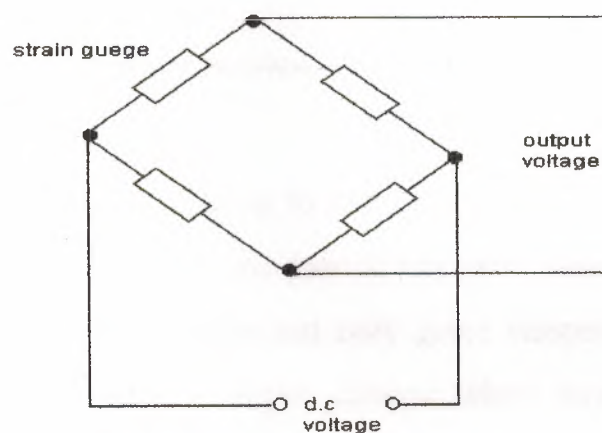


Figure 2.17 Wheatstone bridge circuit

The change in resistance of the strain gauge, when subject to strain, is usually converted into a voltage signal by the use of a Wheatstone bridge (Figure 2.17). A problem that occurs is that the resistance of the strain gauge also changes with temperature and thus some means of temperature compensation has to be used so that the output of the bridge is only a function of the strain.

This can be achieved by placing a dummy strain gauge in an opposite arm of the bridge, that gauge not being subject to any strain but only the temperature (Figure 2.18).

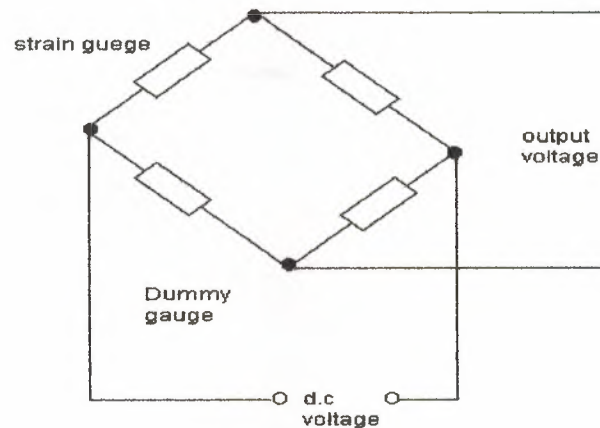


Figure 2.18 Temperature compensation

An alternative that is widely used is to use four active gauges as the arms of the bridge and arrange it so that one pair of opposite gauges is in tension and the other pair in compression. This not only gives temperature compensation but also gives a much larger output change when strain is applied. The following paragraph illustrates systems employing such a form of compensation. By attaching strain gauges to other devices, changes, which result in strain of those devices, can be transformed, by the strain gauges, to give voltage changes. They might, for example, be attached to a cantilever to which forces are applied at its free end (Figure 2.19(a)). The voltage change, resulting from the strain gauges and the Wheatstone bridge, then becomes a measure of the force. Another possibility is to attach strain gauges to a

diaphragm, which deforms as a result of pressure (Figure 2.19(b)). The output from the gauges, and associated Wheatstone bridge, then becomes a measure of the pressure.

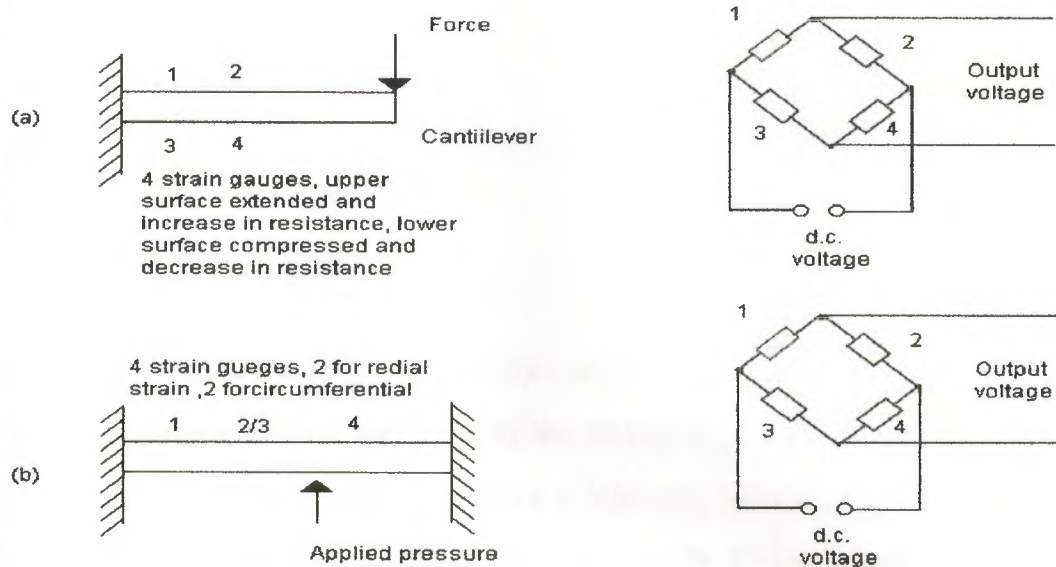


Figure 2.19 Strain gauges used for a) force sensor, b) pressure sensor

2.1.8 Pressure sensors

Commonly used pressure sensors, which give responses related to the pressure, are diaphragm and bellows types. The diaphragm type consists of a thin disk of metal or plastic, secured round its edges. When there is a pressure difference between the two sides of the diaphragm, the centre of it deflects. The amount of deflection is related to the pressure difference. This deflection may be detected by strain gauges attached to the diaphragm (see Figure 2.19(b)) or by using the deflection to squeeze a piezoelectric crystal (Figure 2.20). When a piezoelectric crystal is squeezed, there is a relative displacement of positive and negative charges within the crystal and the outer surfaces of the crystal become charged. Hence a potential difference appears

across it. An example of such a sensor is the Motorola MPX100AP sensor (Figure 2.21)

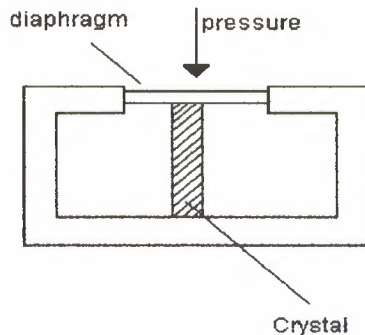


Figure 2.20 Piezoelectric pressure sensor

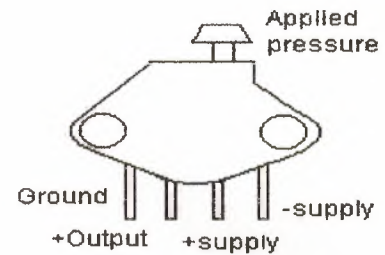


Figure 2.21 MPX100AP

This has a built-in vacuum on one side of the diaphragm and so the deflection of the diaphragm gives a measure of the absolute pressure applied to the other side of the diaphragm. The output is a voltage, which is proportional to the applied pressure with a sensitivity of 0.6 mV/kPa . Other versions are available which have one side of the diaphragm open to the atmosphere and so can be used to measure gauge pressure, others allow pressures to be applied to both sides of the diaphragm and so can be used to measure differential pressures.

Pressure switches are designed to switch on or off at a particular pressure. A typical form involves a diaphragm or bellows, which moves under the action of the pressure and operates a mechanical switch. Figure 2.22 shows two possible forms. Diaphragms are less sensitive than bellows but can withstand greater pressures.

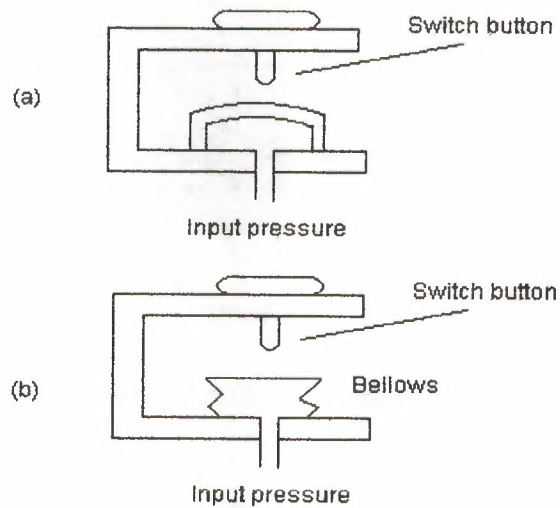


Figure 2.22 Example of pressure switches

2.1.9 Liquid level detector

Pressure sensors may be used to monitor the depth of a liquid in a tank. The pressure due to a height of liquid h above some level is $h\rho g$, where ρ is the density of the liquid and g the acceleration due to gravity. Thus a commonly used method of determining the level of liquid in a tank is to measure the pressure due to the liquid above some datum level (Figure 2.23). Often a sensor is just required to give a signal when the level in some container reaches a particular level. A float switch that is used for this purpose consists of a float containing a magnet, which moves in a housing with a reed switch. As the float rises or falls it turns the reed switch on or off, the reed switch being connected in a circuit, which then switches on or off a voltage.

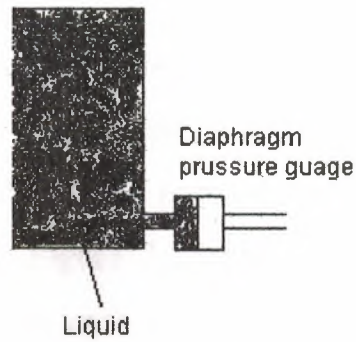


Figure 2.23 Liquid level sensor

2.1.10 Fluid flow measurement

A common form of fluid flow meter is that based on measuring the difference in pressure resulting when a fluid flows through a constriction. Figure 2.24 shows a commonly used form, the orifice flow meter. As a result of the fluid flowing through the orifice, the pressure at A is higher than that at B, the difference in pressure being a measure of the rate of flow. This pressure difference can be monitored by means of a diaphragm pressure gauge and thus becomes a measure of the rate of flow.

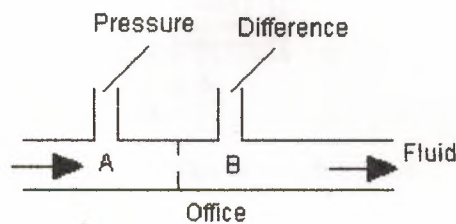


Figure 2.24 Orifice flow meter

2.1.11 Keypads

Many machines employ keypads to input instructions to set the conditions required for outputs such as temperatures or speeds. Such keypads commonly have buttons which, when pressed, operate conductive silicon rubber pads to make contacts. Figure 2.25 shows the form such a 12-way keypad can take. Rather than have each key wired up separately and so giving 12 inputs, the keys are connected in rows and columns and closing a single key can give a column output and a row output, which is unique to that key. This reduces the number of inputs required to the PLC.

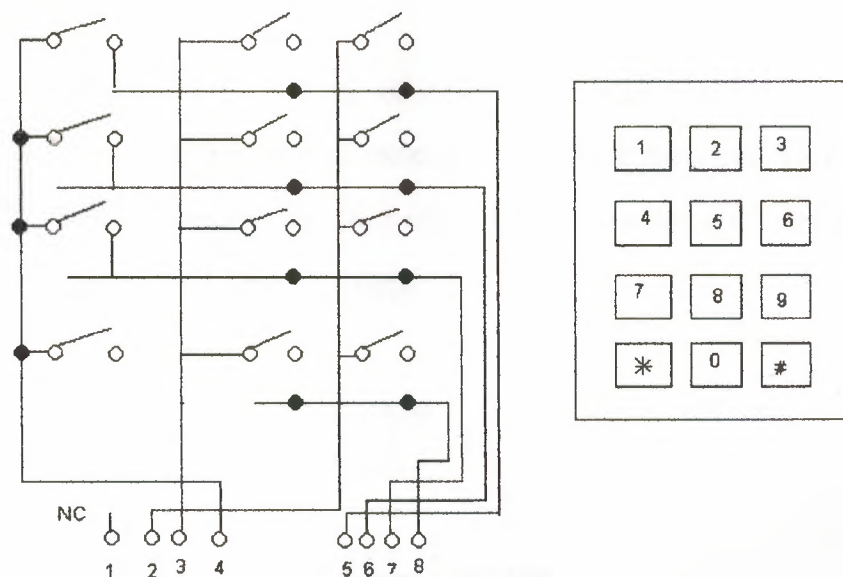


Figure 2.25 12-Way keypad

2.2 Output devices

The output ports of a PLC are of the relay type or optoisolator with transistor or triac types depending on the devices connected to them, which are to be switched on or off (see Section 1.3.4). Generally, the digital signal from an output channel of a PLC is used to control an actuator, which in turn controls some process. The term actuator is used for the device, which

transforms the electrical signal into some more powerful action, which then results in the control of the process. The following are some examples.

2.2.1 Contactor

Solenoids form the basis of a number of output control actuators. When a current passes through a solenoid a magnetic field is produced and this can then attract ferrous metal components in its vicinity. One example of such an actuator is the contactor. When the output from the PLC is switched on, the solenoid magnetic field is produced and pulls on the contacts and so closes a switch or switches (Figure 2.26). The result is that much larger currents can be switched on. Thus the contactor might be used to switch on the current to a motor.

Essentially a contactor is a form of relay, the difference being that the term relay is used for a device for switching small currents, less than about 10 A, whereas the term contactor is used for a heavy current switching device with currents up to many hundreds of amps.

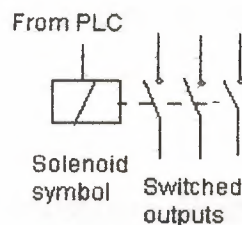


Figure 2.26 Contactor

2.2.2 Directional control valves

Another example of the use of a solenoid as an actuator is a solenoid operated valve. The valve may be used to control the directions of flow of pressurized air or oil and so used to operate other devices such as a piston

moving in a cylinder. Figure 2.27 shows one such form, a spool valve, used to control the movement of a piston in a cylinder.

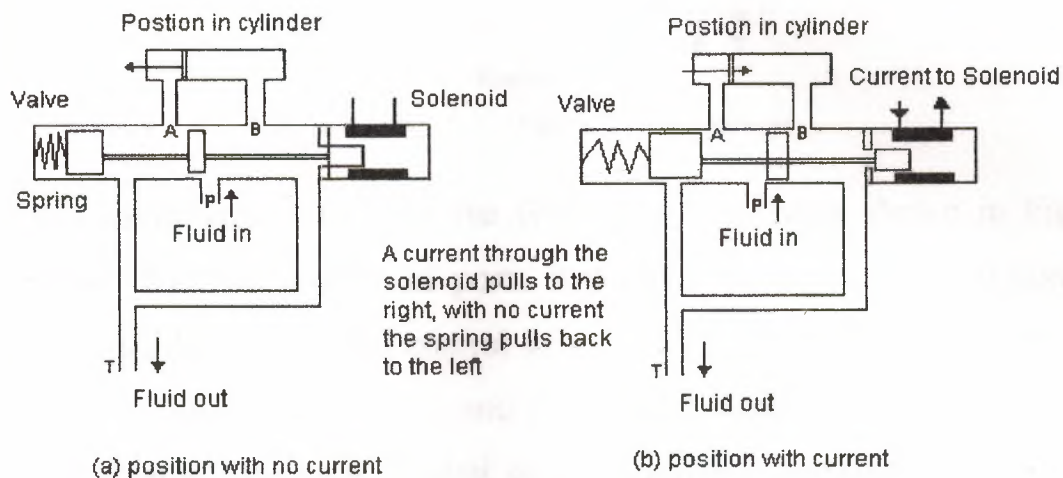


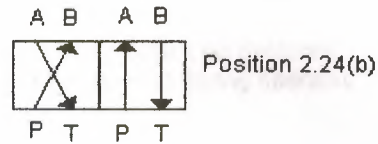
Figure 2.27 A example of solenoid operated valve

Pressurized air or hydraulic fluid is inputted from port P, this being connected to the pressure supply from a pump or compressor and port T is connected to allow hydraulic fluid to return to the supply tank or, in the case of a pneumatic system, to vent the air to the atmosphere. With no current through the solenoid (Figure 2.27(a)) the hydraulic fluid or pressurized air is fed to the right of the piston and exhausted from the left, the result then being the movement of the piston to the left. When a current is passed through the solenoid, the spool valve switches the hydraulic fluid or Pressurized air to the left of the piston and exhausted from the right. The piston then moves to the right. The movement of the piston might be used to push a deflector to deflect items off a

conveyor belt (see Figure 1.1(b)) or implement some other form of displacement, which requires power.



Figure 2.28 Two position valve

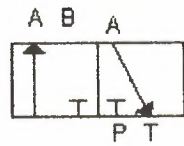


Position 2.27(a)

Figure 2.29 The 4/2 valve

With the above valve there are the two control positions shown in Figure 2.27(a) and (b). The number of ports they have and the number of control positions describes directional control valves. The valve shown in Figure 2.27 has four ports, i.e. A, B, P and T, and two control positions. It is thus referred to as a 4/2 valve. The basic symbol used on drawings for valves is a square, with one square being used to describe each of the control positions. Thus the symbol for the valve in Figure 2.27 consists of two squares (Figure 2.28). Within each square arrows to indicate a flow direction or a terminated line to indicate no flow path then describe the switching positions. Figure 2.29 shows this for the valve shown in Figure 2.27. Figure 2.30 shows some more examples of direction valves and their switching positions.

The actuation methods used with valves are added to the diagram symbol; Figure 2.31 shows examples of such symbols. The value shown in Figure 2.27 has a spring to give one position and a solenoid to give the other and so the symbol is as shown in Figure 2.32.



3/2 valve: no flow from P to A and flow from A to T switched to T begin closed and flow from A to P

Figure 2.30 Direction valves

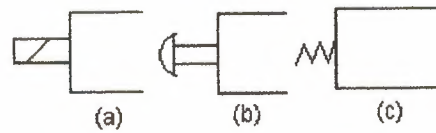


Figure 2.31 Actuation: (a) Solenoid (b) Push button, (c) Spring operated



Position 2.24(b)

Position 2.27(a)

Figure 2.32 The 4/2 valve

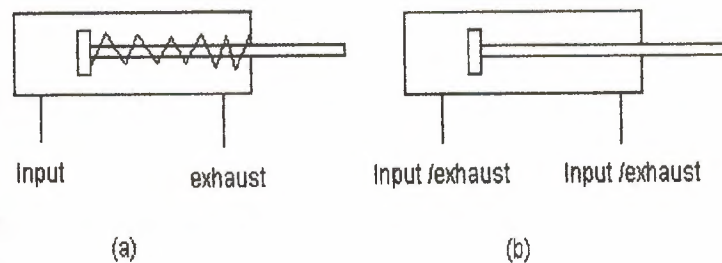


Figure 2.33 Cylinders: (a) single action, (b) double action

Direction valves can be used to control the direction of motion of pistons in cylinders, the displacement of the pistons being used to implement the required actions. The term single acting cylinder (Figure 2.33(a)) is used for one which is powered by the pressurized fluid being applied to one side of the piston to give motion in one direction, it being returned in the other direction by possibly an internal spring. The term double acting cylinder (Figure 2.33(b)) is used when the cylinder is powered by fluid for its motion in both piston movement directions. Figure 2.34 shows how a valve can be used to control the direction of motion of a piston in a single-acting cylinder; Figure

2.35 shows how two valves can be used to control the action of a piston in a double acting cylinder.

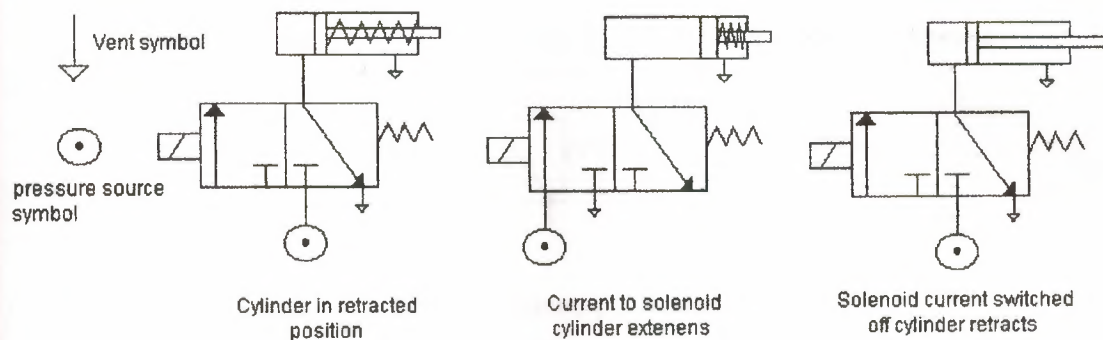


Figure 2.34 Control of a single-acting cylinder

2.2.3 Motors

A d.c. motor has coils of wire mounted in slots on a cylinder of ferromagnetic material, this being termed the armature. The armature is mounted on bearings and is free to rotate. It is mounted in the magnetic field produced by permanent magnets or current passing through coils of wire, these being termed the field coils. The permanent magnet or electromagnet is termed the stator. When a current passes through the armature coil, because a current carrying conductor with a magnetic field at right angles to it experiences a force, forces act on the coil and result in rotation. Figure 2.36 shows the basic principles of such a motor. Brushes and a commutator are used to reverse the current through the coil every half rotation and so keep the coil rotating.

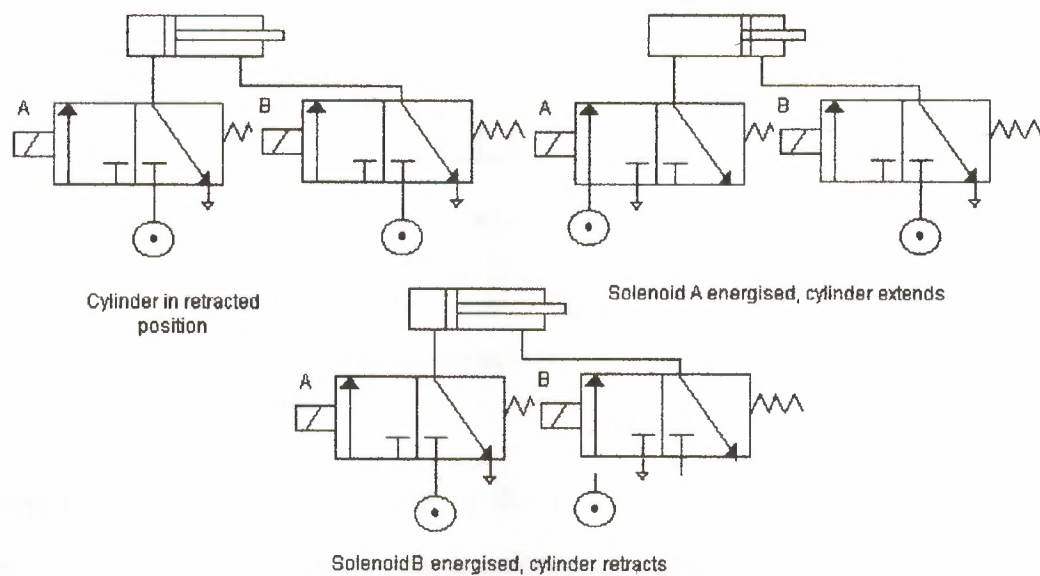


Figure 2.35 Control of double acting cylinder

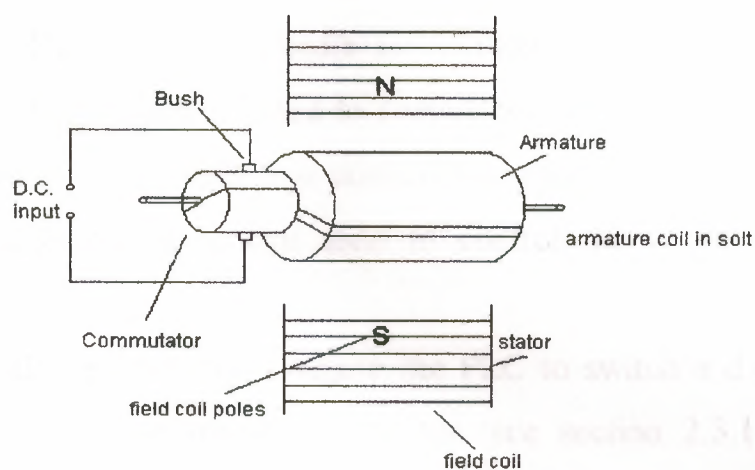


Figure 2.36 Basic element of a d.c. motor

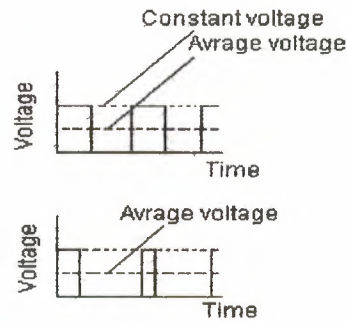


Figure 2.37 Pulse width modulation

Changing the size of the current to the armature coil can change the speed of rotation. However, because fixed voltage supplies are generally used as the input to the coils, the required variable current is often obtained by an electronic circuit. This can control the average value of the voltage, and hence current, by varying the time for which the constant d.c. voltage is switched on (Figure 2.37). The term pulse width modulation (PWM) is used since the width of the voltage pulses is used to control the average d.c. voltage applied to the armature. A PLC might thus control the speed of rotation of a motor by controlling the electronic circuit used to control the width of the voltage pulses.

Many industrial processes only require the PLC to switch a d.c. motor on or off. This might be done using a contactor (see section 2.3.1). Figure 2.38 shows the basic principle. The diode is included to dissipate the induced current resulting from the back e.m.f. Sometimes a PLC is required to reverse the direction of rotation of the motor. This can be done using relays or contactors to reverse the direction of the current applied to the armature coil. Figure 2.39 shows the basic principle. For rotation in one direction, switch 1 is closed and switch 2 opened. For rotation in the other direction, switch 1 is opened and switch 2 closed.

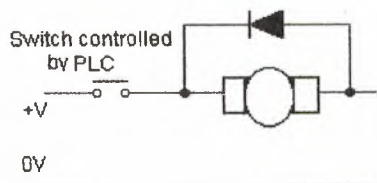


Figure 2.38 On-off control

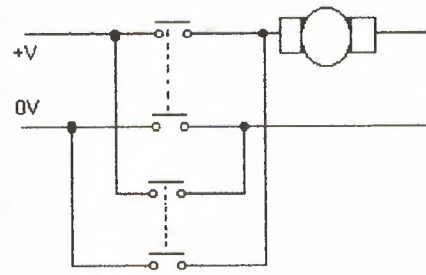


Figure 2.39 Direction control for a d.c. motor

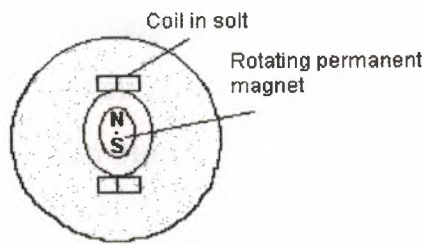


Figure 2.40 Principle of brushless d.c. motor

Another form of d.c. motor is the brushless d.c. motor. This uses a permanent magnet for the magnetic field but, instead of the armature coil rotating as a result of the magnetic field of the magnet, the permanent magnet rotates within the stationary coil. Figure 2.40 shows the basic principle, just one coil being shown. With the conventional d.c. motor, a commutator has to be used to reverse the current through the coil every half rotation in order to keep the coil rotating in the same direction. With the brushless permanent magnet motor, electronic circuitry is used to reverse the current.

The motor can be started and stopped by controlling the current to the stationary coil. To reverse the motor, reversing the current is not so easy because of the electronic circuitry used for the commutator function. One method that is used is to incorporate sensors with the motor to detect the position of the north and south poles. These sensors can then cause the current

to the coils to be switched at just the right moment to reverse the forces applied to the magnet. The speed of rotation can be controlled using pulse width modulation, i.e. controlling the average value of pulses of a constant d.c. voltage (see Figure 2.37).

Alternating current motors consist of two basic parts, a rotating cylinder called the rotor and a stationary part called the stator. The stator surrounds the rotor and has the coil windings that produce a rotating magnetic field in the space occupied by the rotor. It is this rotating magnetic field, which causes the rotor to rotate. One form of such a motor is illustrated in Figure 2.41. This is the single-phase squirrel-cage induction motor.

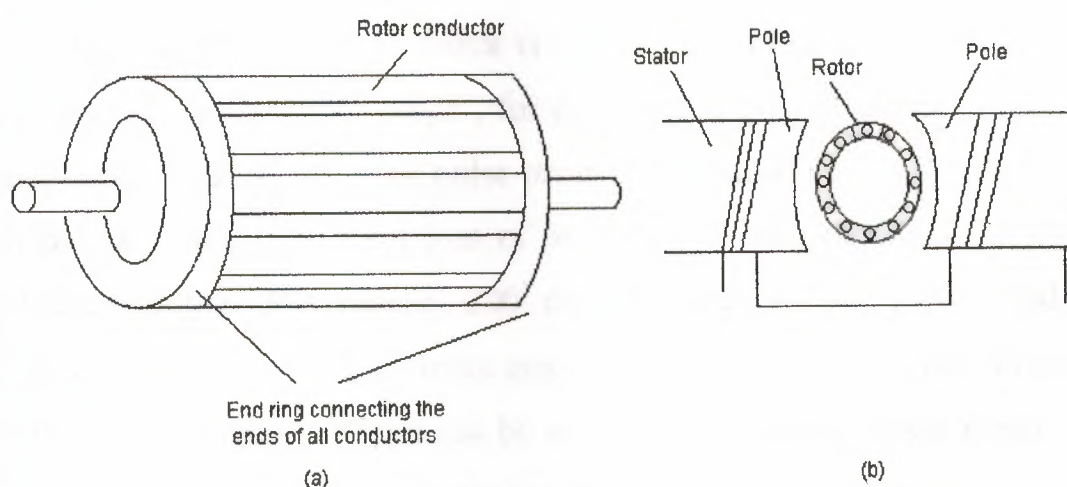


Figure 2.41 (a) Squirrel-cage rotors, (b) the rotor with a single-phase stator

The motor is the squirrel cage, consisting of copper or aluminium bars fitting into slots in end rings to form a set of parallel-connected conductors. There are no external electrical connections to the rotor. When an alternating current passes through the stator coil, an alternating magnetic field is produced and, as a consequence, an e.m.f. is induced in the rotor conductors and currents flow

through them. We thus have current flowing in conductors in a magnetic field and so forces act on them. Given an initial impetus, these forces continue the rotation. The rotor rotates at a speed determined by the frequency of the alternation current applied to the stator. One way of varying the speed of rotation is to use an electronic circuit to control the frequency of the current supplied to the stator.

Though a.c motor are cheaper, more rugged and more reliable then d.c. motors, the maintaining of speed and controlling that speed is generally more complex then with d.c. motors . As a consequence, d.c. motors, particularly burshless permanent magnet motors, tend to be more widely used for control purposes.

2.2.4 stepper motors

The stepper or stepping motor is a motor that produces rotation through equal angles, the so-termed steps , for each digital pulse supplied to its input (figure 2.42). Thus, if one input pulse produce a rotation of 1.8 degree then 20 such pulses would give a rotation of 36.0 degree . To obtain one complete revolution through 360 degree, 200 digital pulses would be required. the motor can thus be used for accurate angular positioning. If it is used to drive a continuous belt (figure 2.43), it can be used to give accurate linear positioning. Such a motor is used with computer printers, robots, machine tools and a wide range of instruments where accurate positioning is required.

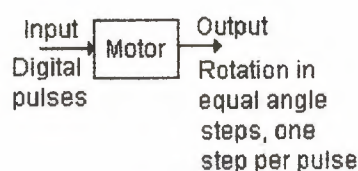


Figure 2.42 The stepping motor

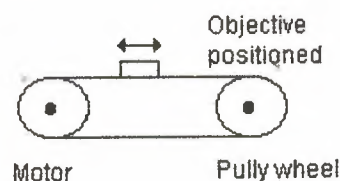


Figure 2.43 Linear positioning

There are a number of types of stepping motor. Figure 2.44 shows the basic principle of the variable reluctance type. The rotor is made of soft steel and has a number of teeth, the number being less than the number of poles on the stator. The stator has pairs of poles, each pair of poles being activated and made into an electromagnet by a current being passed through the coils wrapped round them. When one pair of poles is activated, a magnetic field is produced which attracts the nearest pair of rotor teeth so that the teeth and poles line up. This is termed the position of minimum reluctance. By then switching the current to the next pair of poles, the rotor can be made to rotate to line up with those poles. Thus by sequentially switching the current from one pair of poles to the next, the rotor can be made to rotate in steps.

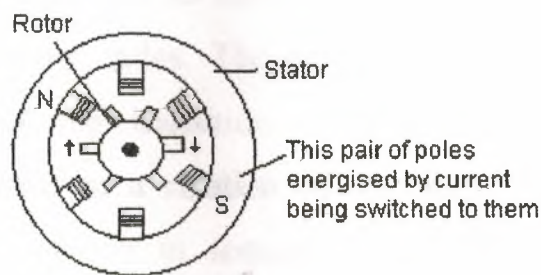


Figure 2.44 The principle of the variable reluctance stepper motor

To drive a stepper motor, each pair of stator coil has to be switched on and off in the required sequence. Thus the input to the motor of a sequence of poles has to provide outputs to the pairs of stator coils in correct sequence. The drive system used for this purpose consists essentially of two blocks, a logic sequence and a driver (figure 2.45(a)). A logic sequence takes the input of the poles and gives the required sequence of puts to control the driver so that it produces the required size outputs, in sequence, to activate the coils of the

stepper motor. Figure 2.45(b) starts illustrate a sequence for a stator having four pairs of coils.

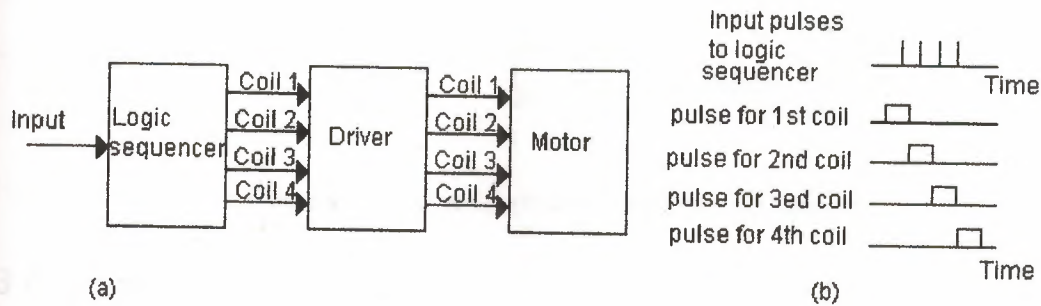


Figure 2.45 (a) Drive system for a four-phase stepper motor, (b) input and outputs of the drive system

Driver circuits can be obtained as integrated circuits. Figure 2.46 shows the integrated circuit SAA1027 and its connections for use with a stepper motor having four pairs of stator poles. The input to trigger the step rotation of stepper motor is a low to high transition for the voltage on the input to pin 15. Each such trigger results in a rotation of one step. The outputs from the integrated circuit are currents in sequence along the brown (pin6), black (pin8), green (pin9) and yellow (pin11) connections to the stator coils. Motor will run clockwise when pin3 is low, i.e. less than 4.5V, and anticlockwise when it is high, i.e. more than 7.5V. When pin2 is made low, the output resets to its initial position.

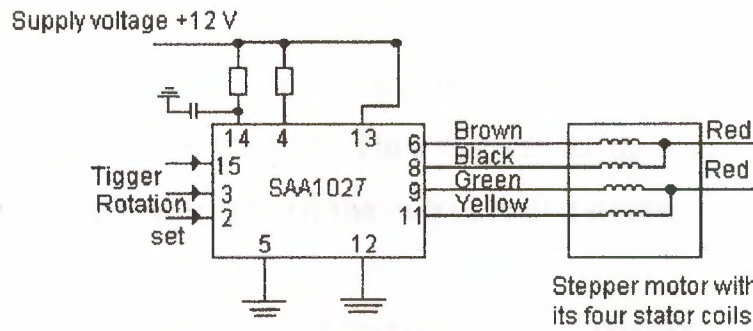


Figure 2.46 Driver circuit connections with the integrated circuit

2.3 Examples of applications

The following are some examples of control systems designed to illustrate the use of a range of a range of input and output devices.

2.3.1 A conveyor belt

Consider a conveyor belt that is to be used to transport goods from a loading machine to a packaging area (figure2.47). When an item is loaded onto the conveyor belt, a contact switch might be used to indicate that the item is on the belt and start the conveyor motor. The motor then has to keep running until

item reaches the far end of the conveyor and falls off into the packaging area.

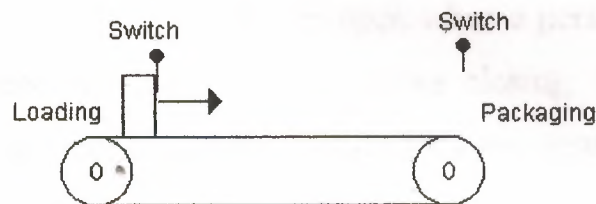


Figure 2.47 Conveyor

When it does this, a switch might be activated which has the effect of switching off the conveyor motor. The motor is then to remain off until the next item is loaded onto the belt. Thus the inputs to a PLC controlling the conveyor are from switches and the output is to a motor.

2.3.2 A lift

Consider a simple goods lift to move items from one level to another. It might be bricks from the ground level to the height where the bricklayers are working. The lift is to move upwards when a push button is pressed at the ground level to send the lift upwards or a push button is pressed at the upper level to request the lift to move upwards, but in both cases there is a condition that has to be met that a limit switch indicates that the access gate to the lift platform is closed. The lift is to move downwards when a push button is pressed at upper level to send the lift downwards or a push button is pressed at the lower level to request the lift to move downwards but in both cases there is a condition that has to be met that a limit switch indicates that the access gate to the lift platform is closed. Thus the inputs are switches and limit switches. The output from the control system is the signal to control the motor.

2.3.3 An automatic door

Consider an automatic door that is to open when a person approaches it, remain open for a specified time, say 5s, before closing. The input to the control system might be from a sensor to detect a person approaching from the outside and another sensor to detect a person approaching from the inside. These sensors might be heat sensitive semiconductor elements that give voltage signals when infrared radiation falls on them. There will also be inputs to the controller probably from limit switches to indicate when the door is fully open and a timer to keep the door open for the required time. The output from the controller might be to solenoid operated pneumatic valves that used movement

of a distance in cylinder to open and close the door. Figure 2.48 shows a simple valve system that might be used. When there is an output to the solenoid to open the door in words because a person has approached from outside the air pressure is applied, via Port, to the unvented side of the piston and cases it to move. When this solenoid is no longer energized, the spring returns the piston back by connecting the unvented side to a vent to the atmosphere. A similar arrangement is used from opening the door autowards.

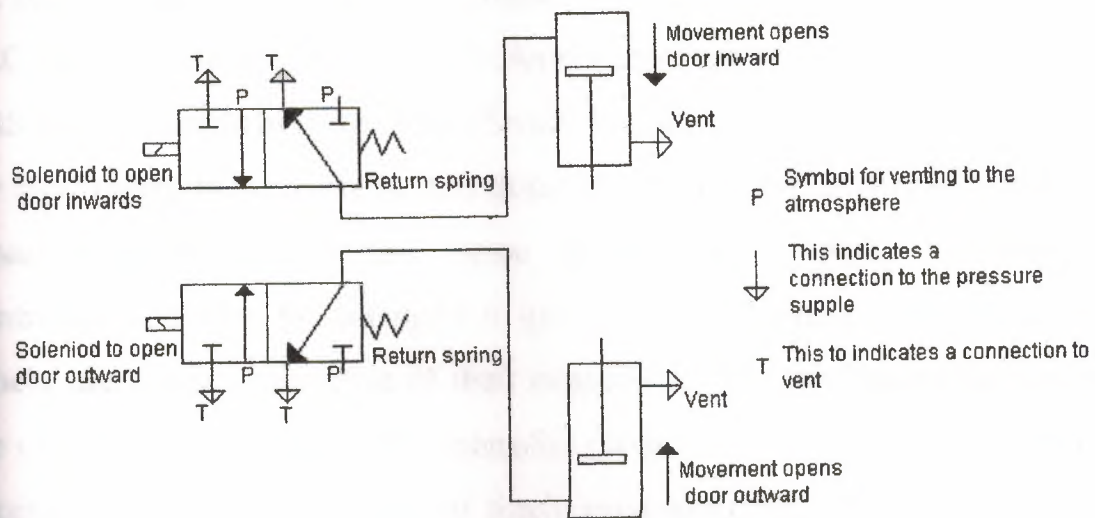


Figure 2.48 Pneumatic door opening system

Chapter 3

PLCs – HARDWARE DESIGN

3.1 TERMINOLOGY - PC OR 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 (on/off) environment. Since all but the smallest programmable controllers can now be equipped to process analog inputs and outputs these 'labels' are not representative of their capabilities. For this reason (and others) the overall term programmable controller (abbreviated to PC) 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.

3.2 COMPARISON WITH OTHER CONTROL SYSTEMS

Table 3.1 provides a comparison between various control media. This is only an approximate guide to their capabilities, and further technical information can be obtained from the manufacturers' data sheets on each specific system.

Programmable controllers emerge from the comparison as the best overall choice for a 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 respectively. 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 controllers of a wide range of industrial equipment. We shall now examine these features in some detail.

Characteristic	Relay systems	Digital logic	Computers	PLC systems
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 immunity	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 operation	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

Table 3.1 Comparison of control system

3.3 PLCs - HARDWARE DESIGN

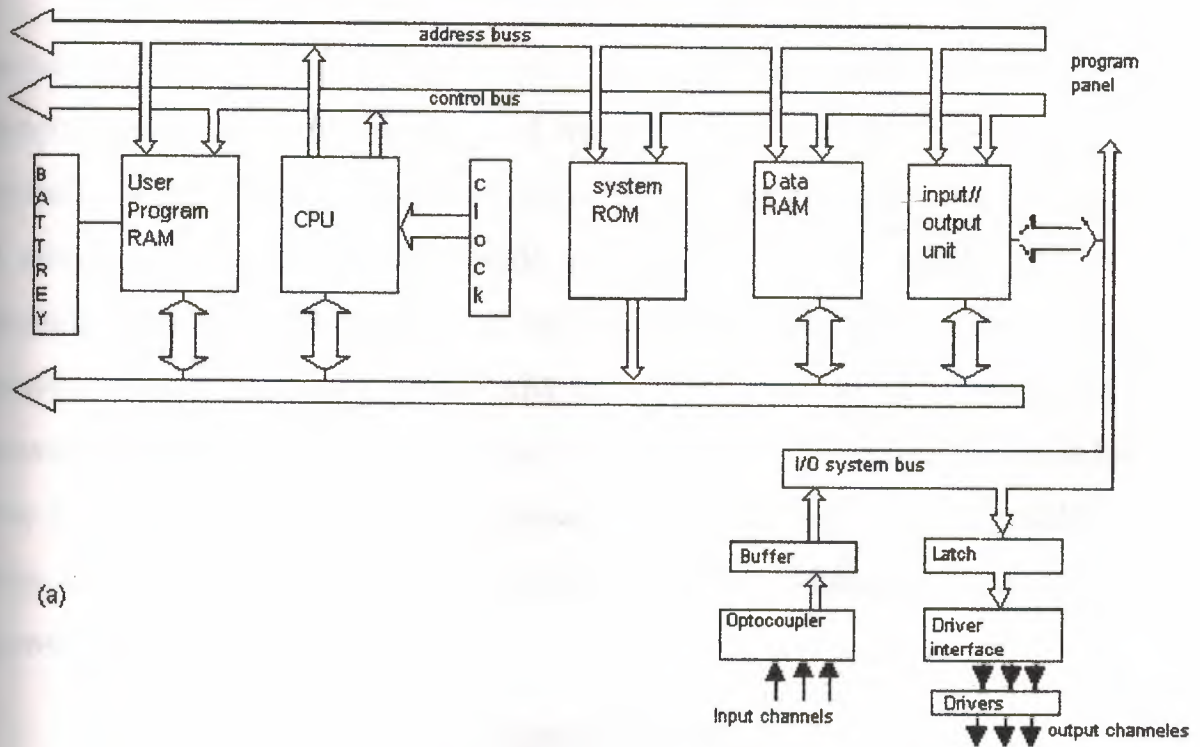
Programmable controllers are purpose-built computers consisting of three functional areas: processing, memory and input/output. Input conditions to the PLC are sensed and then stored in memory, where the PLC performs the programmed logic instructions on these input states. Output conditions are then generated to drive associated equipment. The action taken depends totally on the control program held in memory. In smaller PLCs individual printed circuit cards within a single compact unit perform these functions, whilst larger PLCs are constructed on a modular basis with function modules slotted into the backplane connectors of the mounting rack. This allows simple expansion of the system when necessary. In both these cases the individual circuit boards are easily removed and replaced, facilitating rapid repair of the system should faults develop.

In addition a programming unit is necessary to download control programs to the PLC memory.

3.3.1 Central processing unit (CPU)

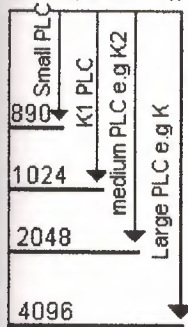
The CPU Controls and supervises all Operations within the PLC, carrying out programmed instructions stored in the memory. An internal communications highway, or bus system, carries information to and from the CPU, memory and I/O units, under control of the CPU. The CPU is supplied with a clock frequency 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 (See Fig. 3.1(a)). Virtually all modern programmable controllers are microprocessor-based, using a 'micro' as the system CPU (see Fig. 3.1(c)). Some larger PLCs

also employ additional microprocessors to control complex, time-consuming functions such as mathematical processing, three-term PID control, etc.



(1) User program memory (2) Data program memory

SI (read only)



Battery-backed
RAM of EPROM

(b)

M0	temporary memory (m) (markers or auxiliary relays)
T60	Timer and counter (T,C)
n	
	Other functions

Certain sections
battery-backed to
retain status
information

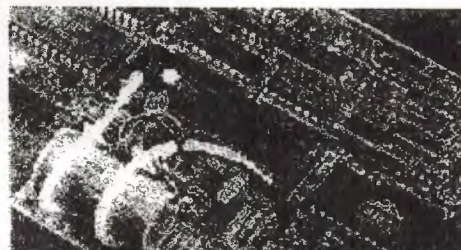


Figure 3.1 (a) Block diagram of PLC internal architecture; (b) memory allocation for a range of PLC (courtesy Mitsubishi Electric UK Ltd); (c) INTEL 8051 SCMP on a small PLC board.

3.3.2 Memory

For program storage all modern programmable controllers use semiconductor memory devices such as RAM read/write memory, or a programmable read-only memory of the EPROM or FEPRONI families. In virtually all cases RAM is used for initial program development and testing, as it allows changes to be easily made in the program. The current trend is to provide CMOS RAM because of its very low power consumption, and to provide battery back-up to this RAM in order to maintain the contents when power is removed from the PLC system. This battery has a lifespan of at least one year before replacement is necessary, or alternatively a rechargeable type may be supplied with the system, being recharged whenever the main PLC power supply is on.

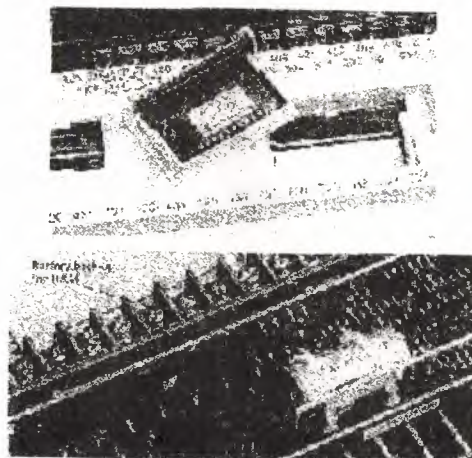


Figure 3.2 (a) EPROM facility on Mitsubishi PLC; (b) battery-backed unit on a small PLC (courtesy Nitsubishi Electric UK ltd).

This feature makes programs stored in RAM virtually permanent. Many users operate their PLC systems on this basis alone, since it permits future program alterations if and when necessary.

After a program is fully developed and tested it may be loaded (blown) into a PROM or EPROM memory chip, which are normally cheaper than RAM devices. PROM programming is usually carried out with a special-purpose programming unit, although many programmable controllers now have this facility built-in, allowing programs in the PLC RAM to be *downloaded* into a PROM IC placed in a socket provided on the PLC itself (Fig. 3.2.)

(b) In addition to program storage, a programmable controller may require memory for other functions:

1. Temporary buffer store for input/output channel status-I/O RAM (see input/output copying below).
2. 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 sections (see Fig. 3.1(b)).

3.3.3 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 users' needs.

	Small PLC	Medium PLC
Model	Mitsubishi F2 40	Allan-Bradley Mini PLC 2
CPU	Intel 8031	Zilog Z80A
Scan time	7m/s per 1K memory	20 m/s per 1K of memory
Memory	CMOS RAM 1K	CMOS RAM 1K
Max I/O	40 points	256 points
Analog I/O	No	128 points max plus PID
Language	Relay ladder/list	Relay ladder

Table 3.2 Hardware features of typical PLCs.

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.

3.3.4 Input/output units

Most PLCs operate internally at between 5 and 15 V d.c. (common TTL and CMOS voltages), whilst *process signals* can be much greater, typically 24 V d.c. to 240 V a.c. at several amperes!

The I/O units form the interface between the microelectronics of the programmable controller and the real world outside, and must therefore provide all necessary signal conditioning and isolation functions. This often allows a PLC lobe directly connected to process actuators and transducers (e.g. pumps and valves) without the need for intermediate circuitry or relays (Fig. 3.3).

To provide this signal conversion programmable controllers are available with a choice of input/output units to suit different requirements. For example:

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

Outputs (choice of): 24 V 100 mA switched 0/P

110V amp

240 V 1 A a.c. (triac)

240 V 2 A a.c. (relay)

It is standard practice for all I/O channels to be electrically isolated from the controlled process, using opto-isolator circuits (Fig. 3.4) on the I/O modules. An opto-isolator circuit consists of a light-emitting diode and a photo-transistor, forming an opto-coupled pair that allows small signals to pass through, but will clamp any high-voltage spikes or surges down to the same small level. This provides protection against switching transients and power-supply surges, normally up to 1500 V.

In small self-contained PLCs in which all I/O points are physically located on the one casing, all inputs will be of one type (e.g. 24 V) and the same for outputs (e.g. all 240 V triac). This is because manufacturers supply only standard function boards for economic reasons. Modular PLCs have greater Flexibility of I/O, however, since the user can select from several different types and combinations of input and output modules.

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

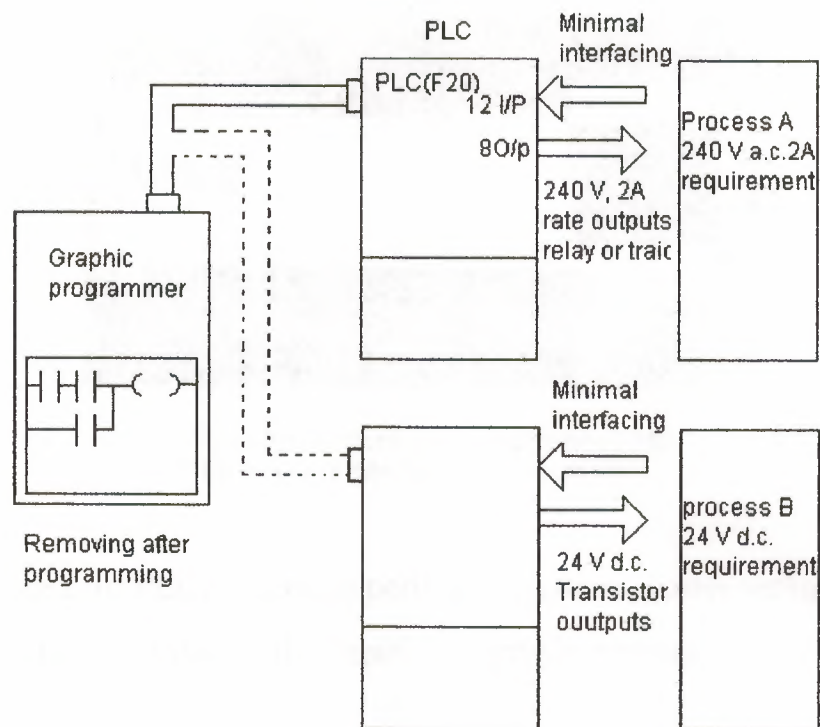


Figure 3.3 PLC input/output connected to plant equipment

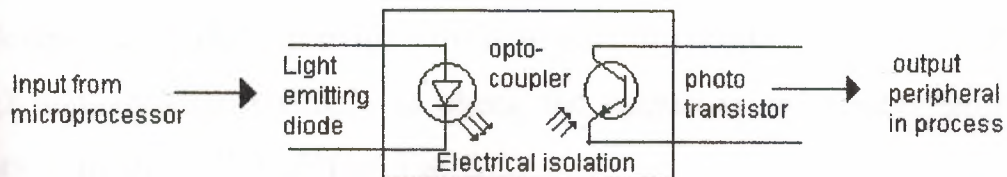


Figure 3.4 OPTo-isolator circuit.

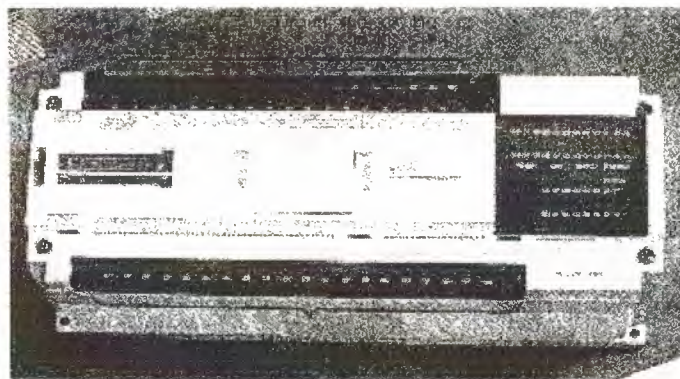


Figure 3.5 Input/output numbering for an F40 PLC (courtesy Mitsubishi Electric UK Ltd)

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.

Every input/output point has a unique address or channel number that is used during program development to specify the monitoring of an input or the activating of a particular output within the program. Indication of the status of input/output channels is provided by light-emitting diodes (LEDs) on the PLC or I/O unit, making it simple to check the operation of process inputs and outputs from the PLC itself (see Fig. 3.5).

3.3.5 Programming units

All but the most simple programming panels contain enough RAM to enable semipermanent 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 (Fig. 3.6). Only when a program is ready for use/testing will it 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 inaccessible.

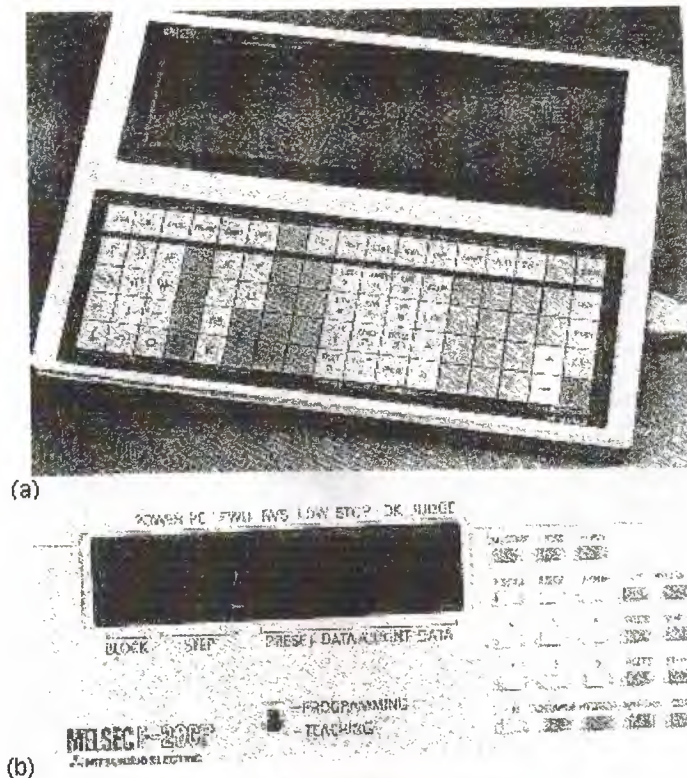
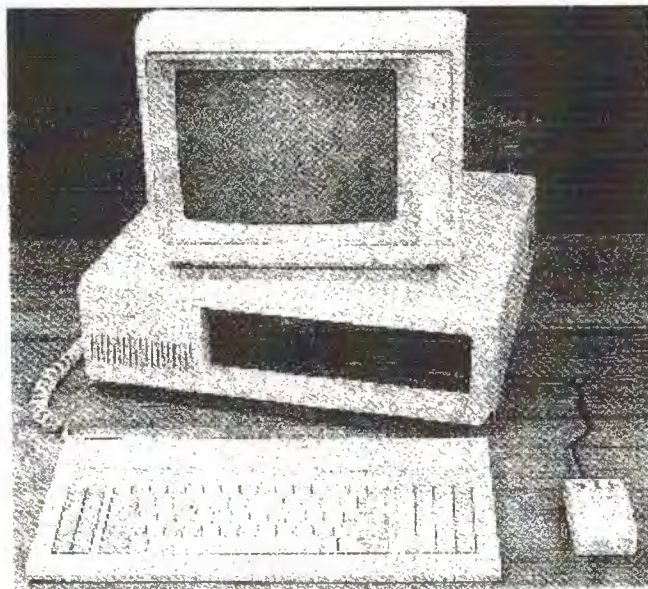


Figure 3.6 Programming panels
(courtesy Mitsubishi Electric UK Ltd)

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 (normally RS232). VDUs provide improved programming facilities such as screen graphics and the inclusion of text comments that assist in the readability of a program. Recently most PC manufacturers have configured personal computers as program development workstations. The high-speed operation and screen graphics facilities of machines like the IBM PC and COMPAQ (Fig. 3.7) are ideal for graphics programming of ladder circuits, etc. (Davie, 1985).

Also, the large memory available on modern 16-bit microcomputers is ideal for storage of several PLC programs complete with comments and documentation. Selecting a personal computer as a programmable controller workstation also provides the user with access to other useful software facilities for project management, such as databases, spreadsheets, word processing and financial planning packages.



(a)

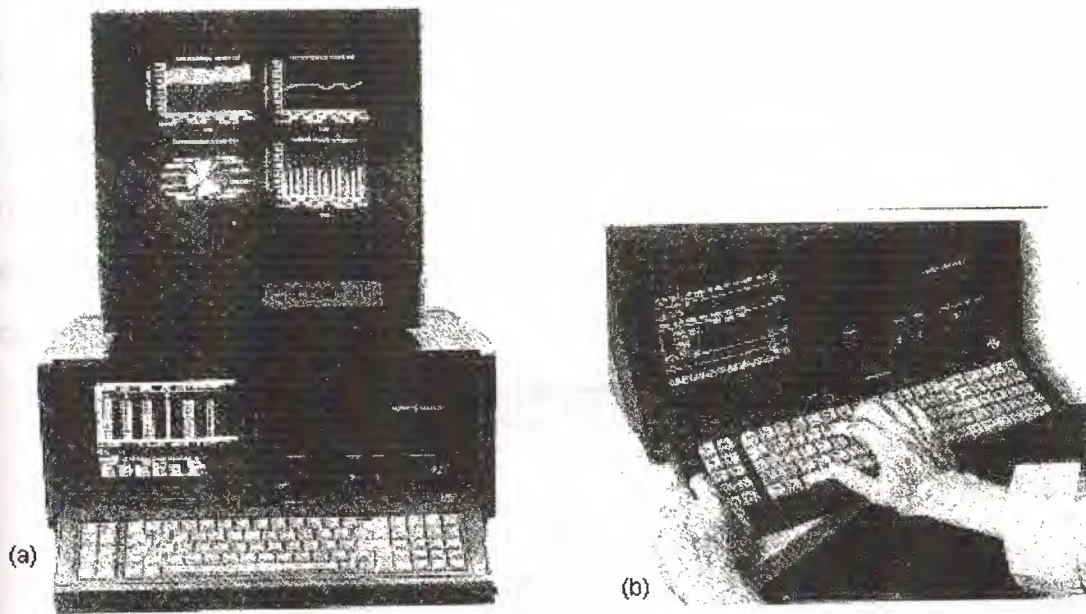


Figure 3.7 Personal computer workstation: (a) IBM PC-compatible; (b) enhanced IBM type; (c) portable version.

3.4 PROGRAMMING PLCs

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.

3.5 PLCs INTERNAL OPERATION AND SIGNAL PROCESSING

When a program is loaded into a PLC, each instruction is placed in an individual memory location (address).

The CPU contains a program counter register that points to the next instruction to be fetched from memory. When the CPU receives an instruction it is placed in the instruction register for decoding into the internal operations (microinstructions) required by that particular instruction. For example, this may result in further instructions being read from memory, or in a physical device being driven by the CPU.

Memory address	Opcode	Operand
0000	LD X400	
0001	AND X401	rung 1
0002	OUT Y430	
0003	LDI Y430	
0004	OUT Y431	rung 2
0005	etc	
.	.	.

Table 3.3 PLC instruction list of figure 3.8

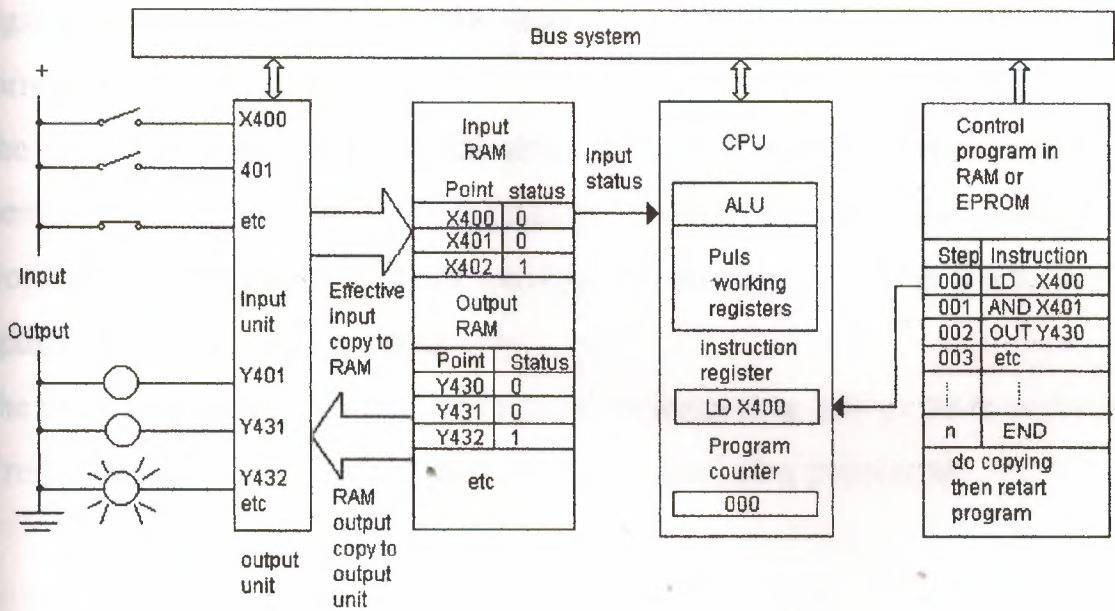


Figure 3.8 Signal processing in the CPU

When the programmable controller is initially set to run, the program counter will point to address 0000 - the location of the first instruction as shown in

Table 3.3 and figure 3.8. The CPU then fetches, decodes and executes this program instruction, here LD X400. Thus the CPU finds it has to examine the first element of a logic circuit a normally open contact (LD) associated with input channel (X) number 400. Input status is held in the I/O RAM, so the CPU scans the RAM location allocated to X400 and reads its status into a temporary memory location or internal register. The CPU is now expecting further instructions to complete the logic circuit. The program counter automatically increments to point to the next address, where the second instruction - AND X401 - is fetched and executed. This causes the CPU to scan the I/O RAM location of X401 and then perform a logical AND function with the values of X400 and X401, temporarily storing the result. The program counter steps on to address 0003 where the instruction OUT Y430 is fetched and decoded, instructing the CPU to pass on the result of the previous logical operation (X400 AND X401) to the location in I/O RAM corresponding to output Y430.

The sequence then continues until either the program counter reaches the last memory location, or an END instruction is encountered. In both cases the program counter is reset to 0000, causing the program to start at the beginning again - i.e.: It will cycle continuously.

The program counter can be programmed to jump over one or more addresses if required, so that the jumped instructions are not being processed

3.5.1 Input/output processing

There are two different methods used for I/O processing in programmable controllers: (a) continuous updating, and (b) mass input/output copying.

(a) Continuous updating

This involves the CPU in scanning input channels as they occur in program instructions, with a built-in delay to ensure that only valid input signals are read into the processor. (The delay typically 3 ms - prevents contact-bounce pulses and other noise from entering the PLC.) Output channels are driven (directly) when OUT instructions are executed following a logical operation. Outputs are latched in the I/O unit so they retain their status until the next updating (Fig. 3.9(a)).

(b) Mass I/O copying

In larger PLCs there may be several hundred input/output points. Since the CPU can deal with only one instruction at a time during program execution, the status of each input point must be examined individually to determine its effect on the program. Since we require a 3 ms delay on each input (see above), the total cycle time for a continuously sampled system becomes progressively longer as the number of inputs rises.

To allow rapid program execution, input and output updating may be carried out at one particular point in the program. Here a specific RAM area within the PLC is used as a buffer store between the control logic and the I/O unit. Each input and output has a cell in this I/O RAM. During I/O copying, the CPU scans all the inputs in the I/O unit and copies their status into the I/O RAM cells. This happens at the start/end of each program cycle (Fig. 3.9(b)).

As the program is executed, the stored input data is read one location at a time from the I/O RAM. Logic operations are performed on the input data, and the

resulting output signals are stored in the output section of the I/O RAM. Then, at the end of each program cycle the I/O copying routine transfers all output signals from the I/O RAM to the corresponding output channel, driving the output stages of the I/O unit. These output stages are latched, that is they retain their status until they are updated by the next I/O routine.

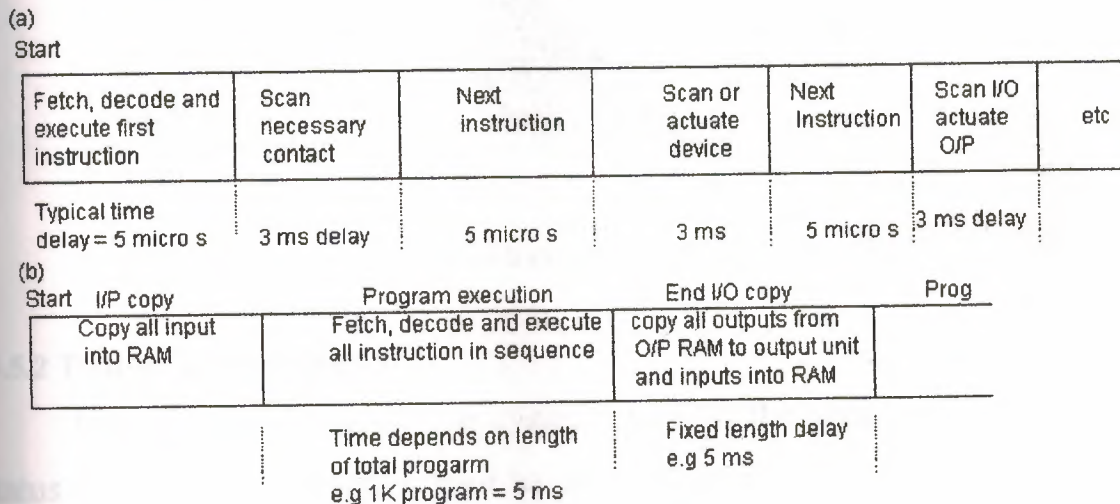


Figure 3.9 Inout scanning and reaction times (a) continuous updating
(b) mass I/O copying.

The CPU as a subroutine to the normal program carries out this task automatically. (A subroutine is a small program designed to perform a specific function that may be called by the main program. In this case the I/O subroutine is automatically accessed by the underlying CPU control program-the monitor.) I/O copying takes place between the end of one program cycle and the start of the next.

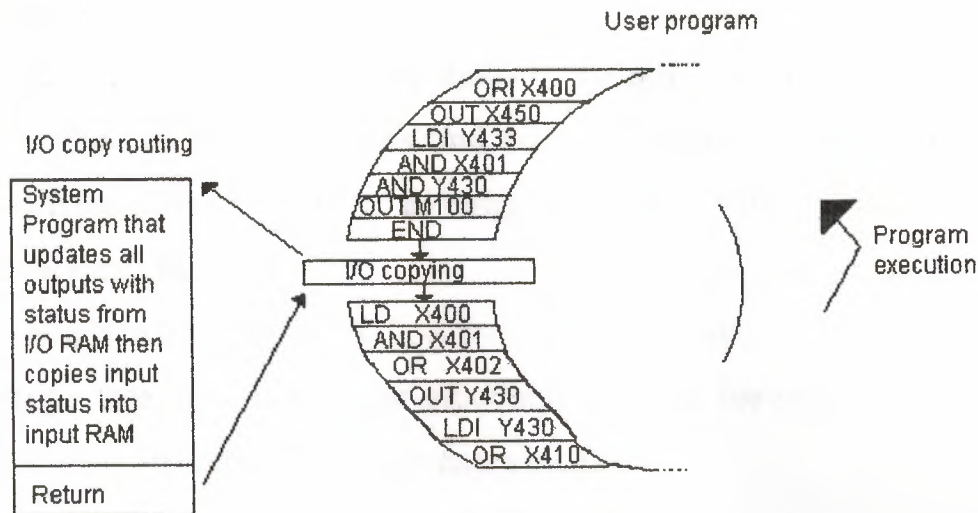


Figure 3.10 Input/output copying

3.5.2 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 within 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 micro second 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.

3.6 Types of PLC system

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 PLCs 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 3.4 gives an example of these categories.

PC size	Max I/O points	User memory size (no. of instruction)
Small	40/40	1 K
medium	128/128	4 K
Large	>128/>128	> 4 K

Table 3.4 Categories of PLC.

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

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

3.6.1 Small PLCs

In general, small and 'mini' PLCs (Fig. 3.11) are designed as robust, compact units that can be mounted on or beside the equipment to be controlled. They are mainly used to replace hard-wired logic relays, timers, counters, etc. 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.

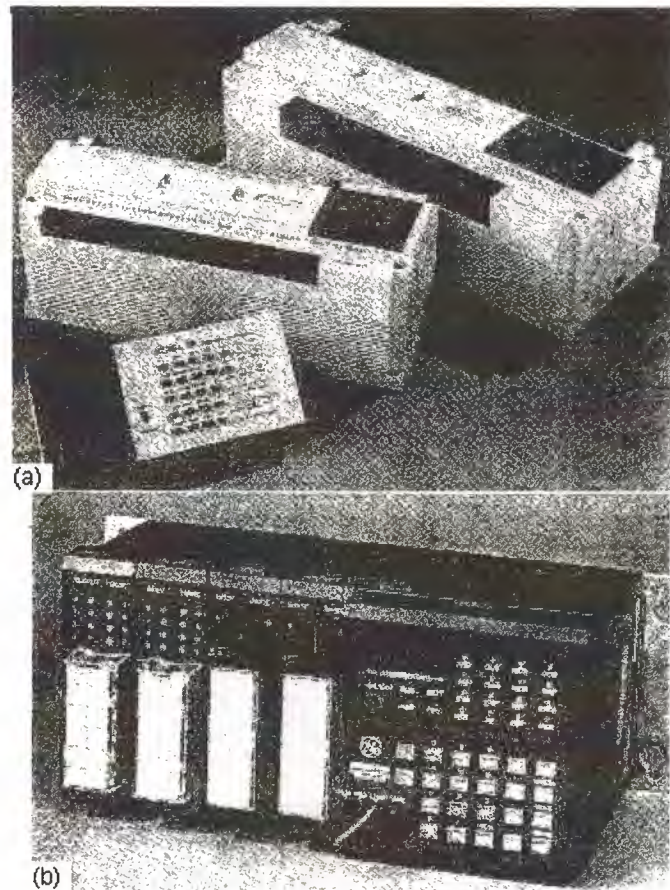


Figure 3.11 small PLC: (a) Mitsubishi F40 (courtesy Mitsubishi Electric UK Ltd); (b) GE series 1 (courtesy General Electric).

This end of the market is very much concerned with non-specialist end-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 at a fairly basic level, including conventional sequencing controls and simple standard functions: e.g. timers and counters. Programming of small PLCs is by

way of logic instruction lists (mnemonics) or relay ladder diagrams. (Standard symbols to IEC 617-7 are being introduced.)

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

Electrical:	240 V a.c. supply; 24 V d.c. on-board for input requirements; 12 input, 8 output points (easily expanded); LED indicators on all I/O points; ALL I/O opto-isolated; Choice of output: Relay (220 V 2 A rated); Triac (220 V d.c. 1 A); Transistor (24 V d.c. 1 A);
Programming:	320 -step memory (CMOS battery-backed RAM). Ladder logic or instruction set using hand-held or graphic LCD programmer, with editor, test and monitor facilities (detachable for use on other PCs; provides interface to printer and tape stor)
Facilities	8 counters, range 1-99 (can be cascaded); 8 timers, range 0.1-99 s (can be cascaded); 64 Markers/auxiliary relays; can be used individually or in Special function relays; Jump capability.

Table 3.5 Features of a typical small PLC- Mitsubishi F20

3.6.2 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 by fitting additional I/O 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 environment. See figure 3.12

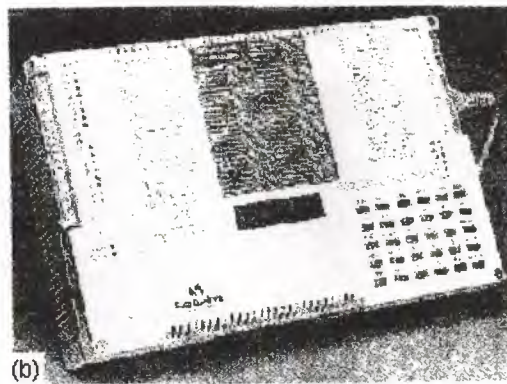
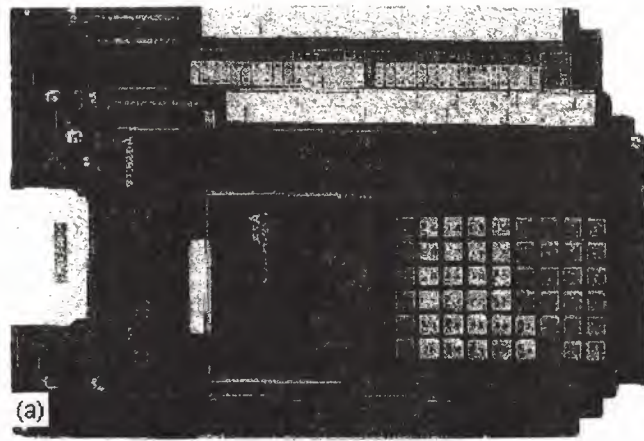


Figure 3.12 Medium-size PLCs: (a) Mitsubishi AO PLC; (b) SattControl 05-35 PC.

In general this type of PLC is applied to logic control tasks that cannot 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 required the replacement of a small PLC, whereas a modular system can be expanded to a much greater extent, allowing for growth. A medium-size PLC may therefore be financially more attractive in the long term. Communications facilities are likely to be provided, enabling the PLC to be including in a 'distribution control' system. Combinations of single and multi-bit processors are likely within the CPU. For programming, standard instructions or ladder and logic diagram are

available. Programming is normally carried out via a small keypad or a VDU terminal.

3.6.3 Large PLCs

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

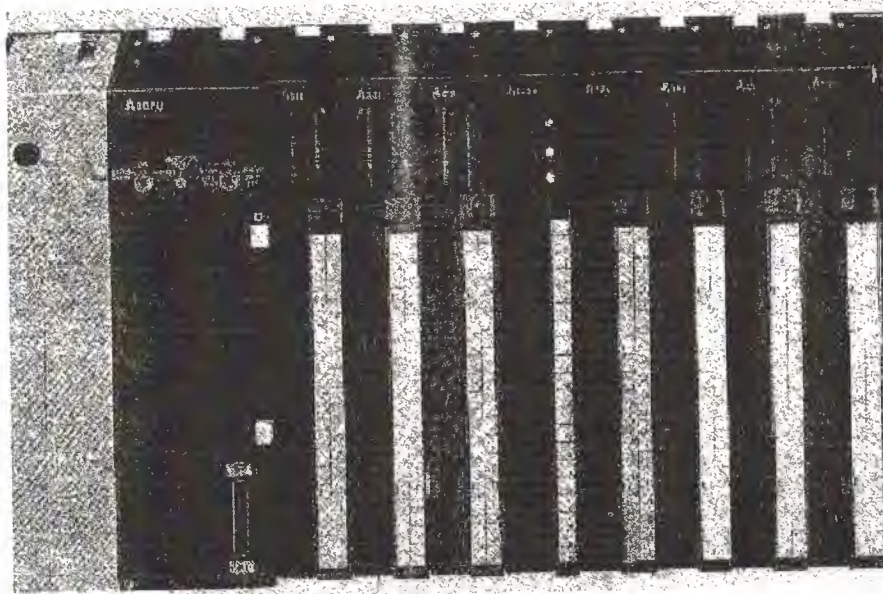


Figure 3.13 Large PLC

Modular construction in Eurocard format is standard, with a wide range of function cards available including analog input/output modules. There is a move towards 16-bit processors, and also multi-processor usage in order to efficiently handle a large range of differing control tasks see figure (3.13 and 3.14), for example:

- a) 16-bit processor as main processor for digital arithmetic and text handling.

b) Single-bit processor as co- or parallel processor for fast counting, storage, etc.

c) Peripheral processor for handling additional tasks that are time-dependent or time-critical, such as

Closed-loop (PID) control

Position controls

Floating-point numerical calculations

Diagnostics and monitoring

Communications for decentralized (distribute) I/O (figure 3.15)

Process mimics (screen graphics)

Remote input/output racks

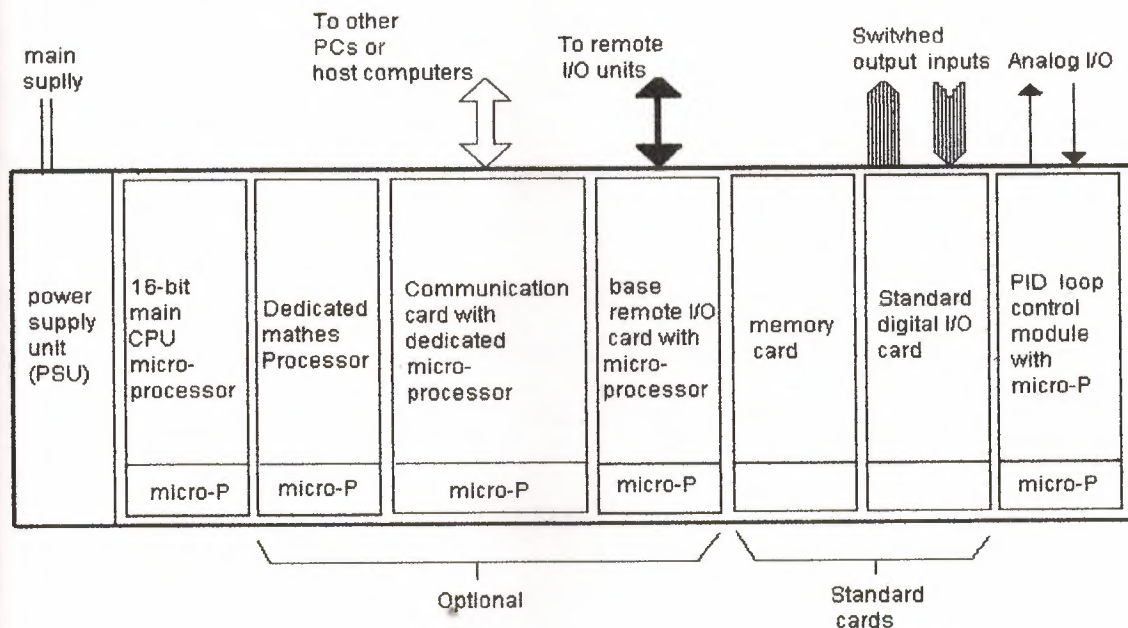


Figure 3.14 Multiprocessing structure in a large PLC

This multi-processor solution optimizes the performance of the overall system as regards versatility and processing speed, allowing the 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.

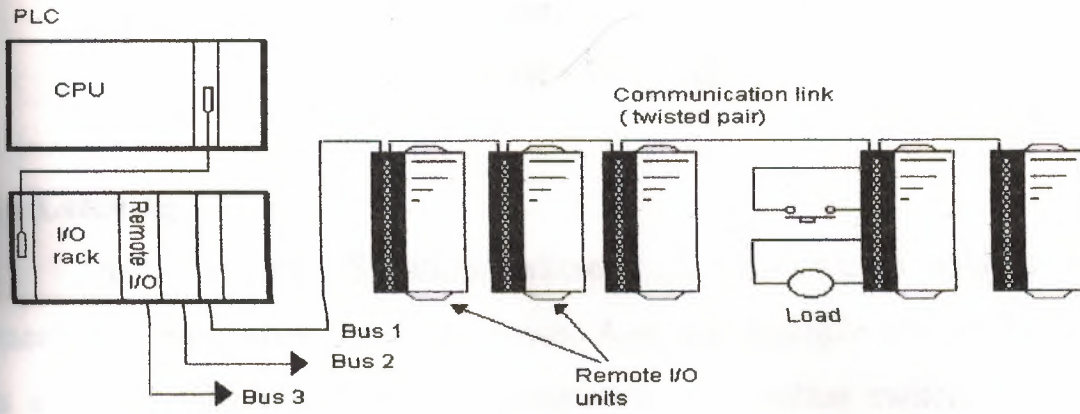


Figure 3.15 Remote input / output units.

Chapter 4

PROGRAMMING

4.1 Latching

There are often Situations where it is necessary to hold an output energized, even when the input ceases. A simple example of such a Situation is a motor, which is started by pressing a push button switch. Though the switch contacts do not remain closed, the motor is required to continue running until a stop push button switch is pressed. The term latch circuit is used for the circuit used to carry out such an operation. It is a self-maintaining circuit in that, after being energized, it maintains that State until another input is received.

An example of a latch circuit is shown in Figure 4.1, (b) showing the circuit in the Mitsubishi form of addresses. When the input A contacts close, there is an output. However, when there is an output, another set of contacts associated with the output closes. These contacts form an OR logic gate System with the input contacts. Thus, even if the input A opens, the circuit will still maintain the output energized. The only way to release the output is by operating the normally closed contact B.

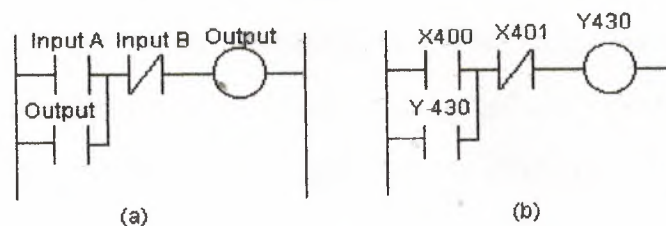


figure 4.1 Latched circuit

As an illustration of the application of a latching circuit, consider a motor controlled by stop and Start push button switches and for which one signal light must be illuminated when the power is applied to the motor and another when it is not applied. Figure 4.2 shows the ladder diagram in Mitsubishi notation.

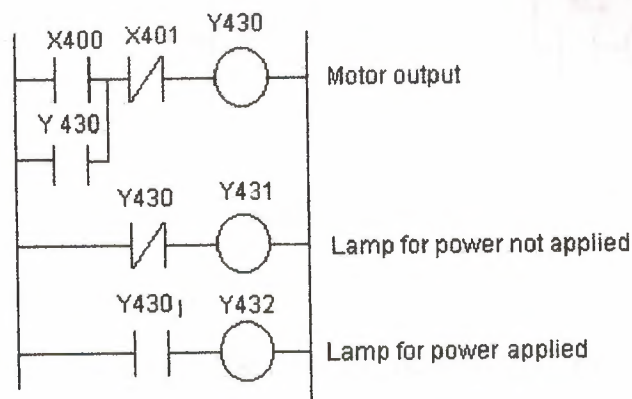


Figure 4.2 motor on-off, with signal lamps, ladder diagram

When X400 is momentarily closed, Y430 is energized and its contacts close. This results in latching and also the switching off of Y431 and the switching on of Y432. To switch the motor off, X401 is pressed and opens. Y430 contacts open in the top rung and third rung, but close in the second rung.

Thus Y431 comes on and Y432 off

4.2 Multiple outputs

With ladder diagrams, there can be more than one output connected to a contact. Figure 4.3 illustrates this with the same ladder program in Mitsubishi and Siemens notations. Outputs Y430, Y431 and Y432 are switched on as the

contacts in the sequence given by the contacts X400, X401 and X402 are being closed. Until X400 is closed, none of the other outputs can be switched on. When X400 is closed, Y430 is switched on. Then, when X401 is closed, Y431 is switched on. Finally, when X402 is closed, Y432 is switched on.

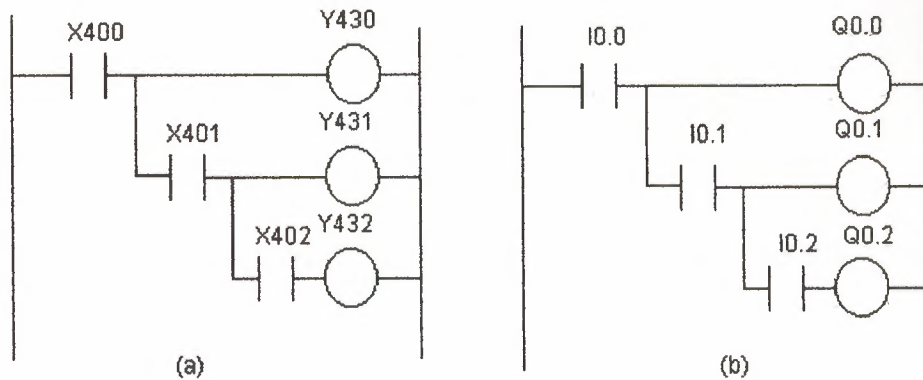


Figure 4.3 Multiple outputs

4.3 Entering ladder programs

Each horizontal rung on the ladder represents an instruction in the program to be used by the PLC. The entire ladder gives the complete program. There are several methods that can be used for keying in the program into a programming terminal. Whatever method is used to enter the program into a programming terminal or computer, the output to the memory of the PLC has to be in a form that can be handled by the microprocessor. This is termed machine language and is just binary code, e.g. 0010100001110001.

4.3.1 Ladder symbols

One method of entering the program into the programming terminal involves using a keypad having keys with symbols depicting the various elements of the ladder diagram and keying them in so that the ladder diagram appears on the screen of the programming terminal. For example, to enter a pair of contacts the key marked



might be used, followed by its address being keyed in. To enter an output the key marked



might be used, followed by its address To indicate the start of a junction



might be pressed; to indicate the end of a junction path



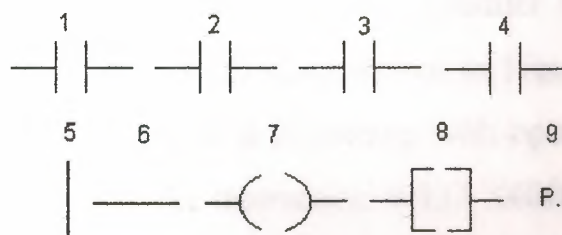
To indicate horizontal circuit links, the following key might be used:



The terminal then translates the program drawn on the screen into machine language.

Computers can be used to draw up a ladder program. These involve loading the computer with the relevant software, e.g. MEDOC for Mitsubishi PLCs, and then selecting items from menus on the screen. Thus, from the main menu that appears initially on the screen, Edit is selected. The menu then changes. Ladder can then be selected from this new menu and results in the screen showing a blank ladder diagram consisting of just two parallel rails. At the

bottom of the screen a series of ladder symbols appear



and are selected, after pressing F2 to enter the ladder diagrams working area and F7 to select the write mode, by entering the number which appears above the symbol. A window will appear in which the address of the item can be entered and then, on pressing enter, the symbol with its address will appear on the ladder. In this way the entire ladder program can be built up on screen. Text can be inserted, e.g. at the beginning of the program to describe its purpose, by pressing F2 to enter the working area and then F5. This opens a window into which text can be typed.

4.4 Instruction lists

Mother programming method, which can be considered to be the entering of a ladder program using text, is instruction lists. For this, mnemonic codes are used, each code corresponding to a ladder element. The codes used differ to some extent from manufacturer to manufacturer, though a standard IEC 1131-3 has been proposed. Table 4.1 shows some of the codes used by manufacturers, and the proposed standard, for instructions used in this chapter (see later chapters for codes for other functions).

Whenever a rung is started, it must use a start a rung code. This might be LD, or perhaps A or L or STR, to indicate the rung is starting with open contacts, or LDI, or perhaps LDN or LD NOT or AN or LN or STR NOT, to indicate it

is starting with closed contacts. All rungs must end with an output. This might be OUT or =

The following shows how individual rungs on a ladder are entered using the Mitsubishi mnemonics for the AND gate, shown in Figure 4.4. Step 0 is the Start of the rung with LD because it is starting with open contacts. Since the address of the input is X400, the instruction is LD X400. This is followed by another open contacts input and so step 1 involves the instruction AND with the address of the element, thus the instruction is AND X401. The rung terminates with an output and so the instruction OUT is used with the address of the output, i.e. OUT Y430. The single rung of a ladder would thus be entered as:

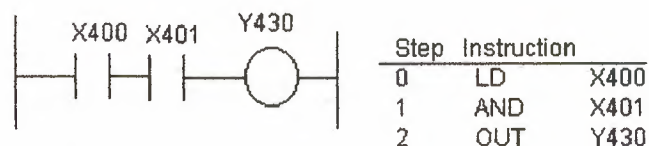


Figure 4.4 AND gate

IEC 131-3	mitsubishi	OMRON	Siemens	Telemec- anique	Sperher+ Schuh	
LD	LD	LD	A	L	STR	start rung with an open contact
LDN	LDI	LD NOT	AN	LN	STR NOT	Start a rung with a close contact
AND	AND	AND	A	A	AND	A series element with an open contact
ANDN	ANI	AND NOT	AN	AN	AND NOT	A series element with a closed contact
O	OR	OR	O	O	OR	A parallel element with an open contact
ORN	ORI	OR NOT	ON	ON	OR NOT	A parallel element with a closed contact
ST	OUT	OUT	=	=	OUT	An output

Table 4.1 Instruction code mnemonics

For the same rung with Siemens notation (Figure 4.5) we have:

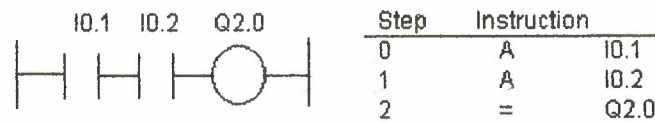


Figure 4.5 AND gate

For the same rung with the Telemecanique notation (Figure 4.6) we have:

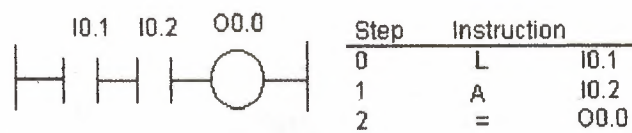


Figure 4.6 AND gate

Consider another example, an OR gate. Figure 4.7 shows the gate with Mitsubishi notation. The instructions for the rung start with an open contact are LD X400. The next item is the parallel OR set of contacts X401. Thus the next instruction is OR X40 1. The last step is the output, hence OUT Y430. The instruction list would thus be:

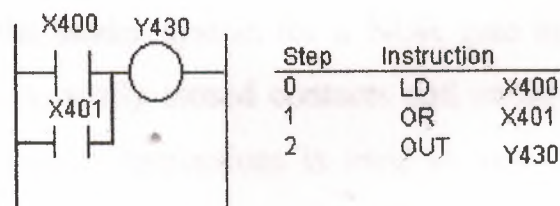


Figure 4.7 OR gate

Figure 4.8 shows the Siemens version of the OR gate. The following is the

Siemens instruction list:

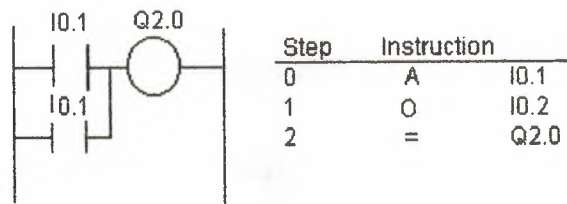


Figure 4.8 OR gate

Figure 4.9 shows the Sprecher+Schuh version of the OR gate. The following is thus the Sprecher+Schuh instruction list:

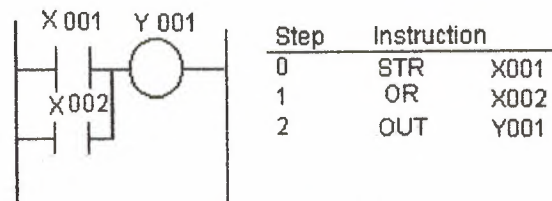


Figure 4.9 OR gate

Figure 4.10 shows the ladder system for a NOR gate in Mitsubishi notation. The rung starts with normally closed contacts and so the instruction is LDI. I when added to Mitsubishi instructions is used to indicate the inverse of the instruction. The next step is a series normally closed contact and so ANI, again the I being used to make an AND instruction the inverse. I is also the instruction for a NOT gate. The instructions for the NOR gate rung of the ladder would thus be entered as:

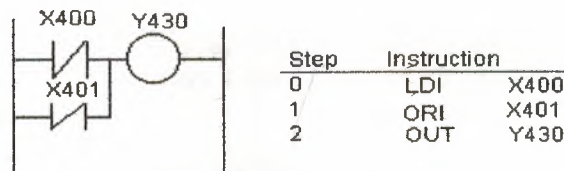


Figure 4.13 NAND gate

Figure 4.14 shows the NAND gate in Siemens notation. The instruction list is then:

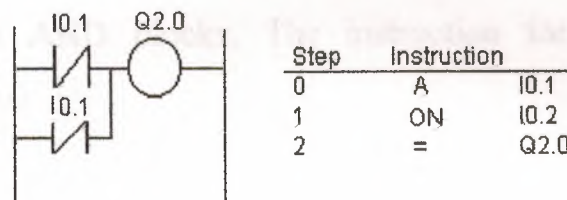


Figure 4.14 NAND gate

4.4.1 Branch codes

The Exclusive OR (XOR) gate shown in Figure 4.15 in Mitsubishi notation has two parallel arms with an AND situation in each arm. In such a situation Mitsubishi uses an ORB instruction to indicate 'OR together parallel branches'. The first instruction is for a normally open pair of contacts X400. The next instruction is for a series set of normally closed contacts X401; hence ANI X401. After reading the first two instructions, the third instruction starts a new line. It is recognized as a new line because it starts with LDI, all new lines starting with LD or LDI. But the first line has not been ended by an output. The PLC thus recognizes that a parallel line is involved for the second line and reads together the listed elements until the ORB instruction is reached. The mnemonic ORB (OR branches blocks together) indicates to the PLC that it should OR the results of steps 0 and 1 with that of the new branch

Step	Instruction	
0	A(Applies to the first set of brackets
1	A	IO.0 Steps 1 and 2 are in the first set of brackets
2	AN	IO.1
3)	
4	O(Applies to the second set of brackets
5	AN	IO.0 Step 5 and 6 are in the second set of brackets
6	A	IO.1
7)	
8	=	Q2.0

Figure 4.17(a) shows a circuit, in Mitsubishi notation, which can be considered as two AND blocks. The instruction for this is ANB. The instruction list is thus:

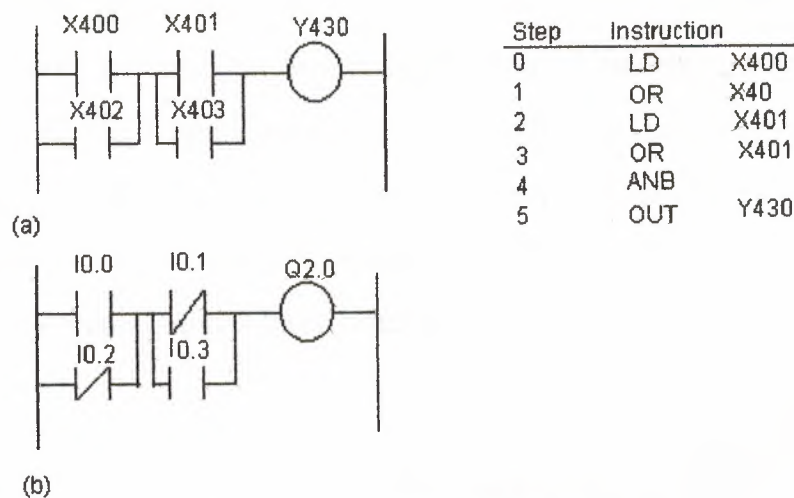


Figure 4.17 (a) Mitsubishi, (b) siemens

Figure 4.17(b) shows the same circuit in Siemens notation. Such a program is written as an instruction list using brackets. The instruction in step 0 applies to the result of steps 1 and 2. The instruction in step 4 applies to the result of steps 5 and 6. The program instruction list is:

with steps 2 and 3. The list concludes with the output OLT Y430. The instruction list would thus be entered as:

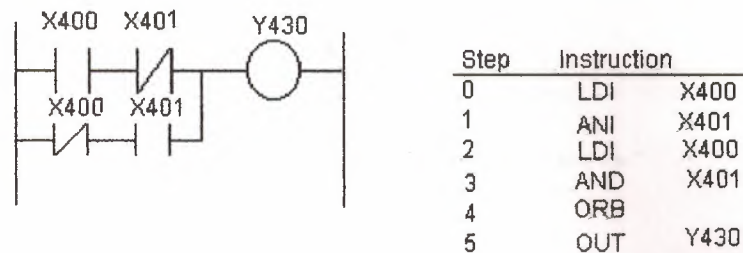


Figure 4.15 XOR gate

Figure 4.16 shows the Siemens version of Figure 4.40. Brackets are used to indicate that certain instructions are to be carried out as a block. They are used in the same way as brackets in any mathematical equation. For example, $(2 + 3)/4$ means that the 2 and 3 must be added before dividing by 4. Thus with the Siemens instruction list we have in step 0 the instruction A (. The brackets close in step 3. This means that the A in step 0 is applied only after the instructions in steps 1 and 2 have been applied.

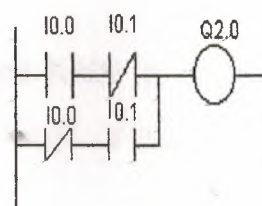


Figure 4.16 XOR gate

Step	Instruction	
0	A(
1	A	I0.0
2	O	I0.1
3)	
4	A(
5	A	I0.0
6	O	I0.1
7)	
8	=	
		Q2.0

4.4.2 More than one rung

Figure 4.18 shows a ladder, in Mitsubishi notation, with two rungs. In writing the instruction list we just write the instructions for each line in turn. The instruction LD or LDI indicates to the PLC that a new rung is starting. The instruction list is thus:

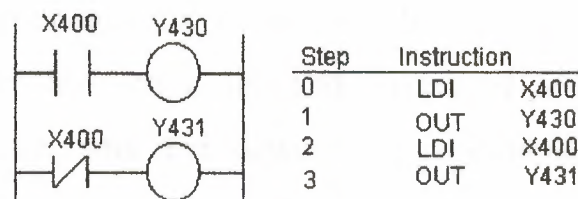


Figure 4.18 Toggel circuit

The system is one where when X400 is not activated, there is an output from Y431 but not Y430. When X400 is activated, there is then an output from Y430 but not Y431.

Figure 4.19 shows the same program in siemens notation. The = instruction indicates the end of a line. The A or AN instruction does not necessarily indicate the beginning of a rung since the same instruction is used for AND and AND NOT. The instruction list is then:

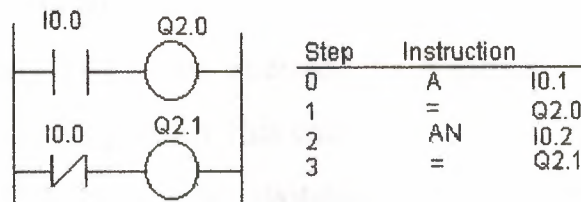


Figure 4.19 Toggle circuit

4.5 IEC standards

As will have been apparent from this chapter, different manufacturers have their own ideas of how PLCs should be programmed. The IEC has, however, produced a standard IEC 1131 part 3 (1993), also given as British Standard BS EN 61131/3, on programming PLCs. This classifies programming methods into two categories: textual language and graphical language. As the names imply, textual language involves text and graphical language graphical images like ladders and block diagrams. Textual language has two methods: instruction list and structured text. Graphical has two methods: ladder diagrams and function block diagrams. Standards are laid down for the writing of programs in these four methods with examples being given.

In this chapter the methods introduced are instruction list, ladder diagrams and function block diagrams. In this book the main emphasis is on instruction list and ladder programming.

4.6 Programming examples

The following tasks are intended to illustrate the application of the programming techniques given in this chapter.

A signal lamp is required to be switched on if a pump is running and the pressure is satisfactory, or if the lamp test switch is closed. For the inputs from the pump and the pressure sensors we have an AND logic situation since both are required if there is to be an output from the lamp. We, however, have an OR logic situation with the test switch in that it is required to give an output of lamp on regardless of whether there is a signal from the AND system. The function block diagram, the ladder diagram and the instruction list are thus of the form shown in Figure 4.20. Note that with the ladder diagram and the instruction list we have to tell the PLC when it has reached the end of the program by the use of the END instruction.

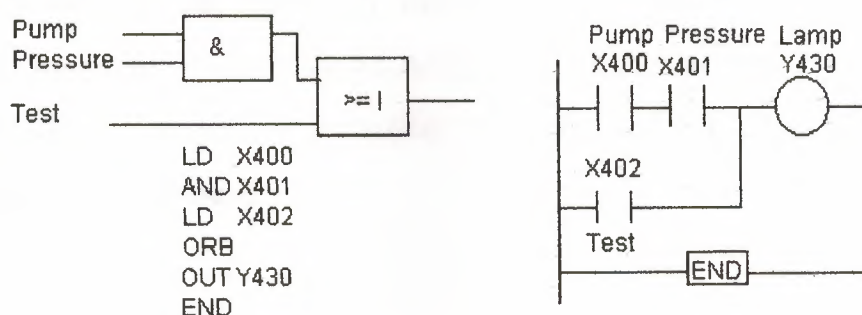


Figure 4.20 Signal lamp task

Consider another task. A motor is to be started when the start push button is pressed and remain on until the stop button is pressed. This requires the start button to be latched so that, after it has been pressed, the output remains on until the stop button is pressed; Figure 4.21 shows the function block diagram, the ladder diagram and the instruction list. Again note the use of the END

instruction to complete the program.

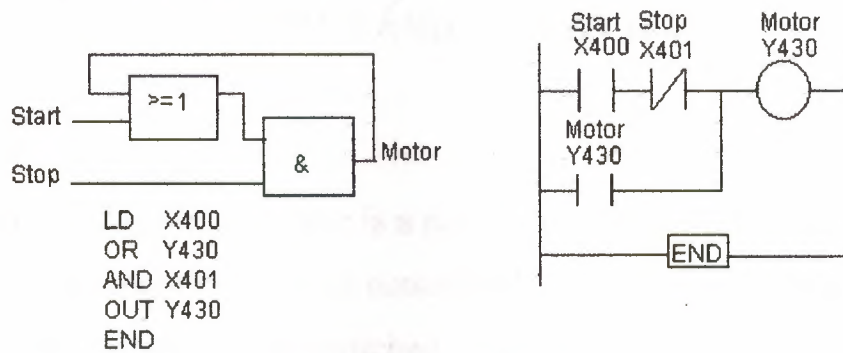


Figure 4.21 Motor task

As another example, consider a machine, which has four sensors to detect when safety features are not active. When there is an input to any one of these sensors, the machine must stop and an alarm sound. Each sensor we can consider giving an input, which has contacts that are normally closed so that the machine can run. When there is an input to the sensor the contacts open and the machine stops. With any one of the four sensors being required to switch the machine off we have an AND logic situation. Figure 4.22 shows the functional block diagram, the ladder diagram and the instruction list of a system that might be used.

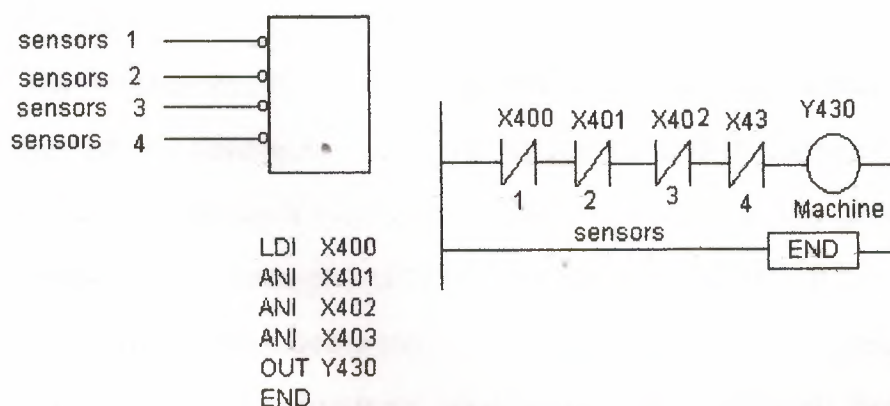


Figure 4.22 Machine safety controls task

Chapter 5

TIMER AND COUNTER

5.1 Timers

In many control tasks there is a need to control time. For example, a motor or a pump might need to be controlled to operate for a particular & interval of time, or perhaps be switched on after some time interval. PLCs thus have timers as built-in devices. Timers count fractions of seconds or seconds using the internal CPU clock (see Figure 1.8). This chapter shows how such timers can be programmed to carry out control tasks.

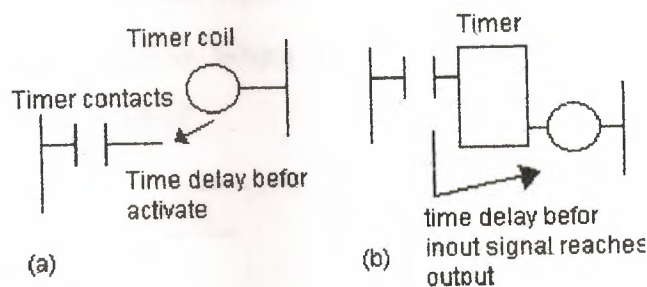


Figure 5.1 Treatment of timers

5.1.1 Types of timers

PLC manufacturers differ on how timers should be programmed and hence how they can be considered. A common approach is to consider timers to behave like relays with coils which when energized result in the closure or opening of contacts after some preset time. The timer is thus treated as an output for a rung with control being exercised over pairs of contacts elsewhere (Figure 5.1(a)). This is the predominant approach used in this book. Some treat a timer as a delay block which when inserted in a rung delays signals in that rung reaching the Output (Figure 5.1(b)).

There are a number of different forms of timers that can be found with PLCs. With small PLCs there is likely to be just one form, the on-delay timers. These are timers that come on after a particular time delay (Figure 5.2(a)). Off-delay timers are on for a fixed period of time before turning off (Figure 5.2(b)).

Another type of timer that occurs is the pulse timer. This timer switches on or off for a fixed period of time (Figure 5.2 (c)).

With function block diagram and the symbols used for timers by some manufacturers, TON is used to denote, on-delay, TOF off-delay. On delay is also represented by 0-T and off-delay by T-0. Pulse timer are denoted by TP. The timer duration for which a timer has been set is termed the preset and is set in multiples of the time base. Some timers' bases are typically 10 ms, 100 ms, 1 s, 10 s and 100 s. Thus a preset value of 5 with a time base of 100 ms is a time of 500 ms. For convenience, where timers are involved in this text, a timer based of 1 s has been used.

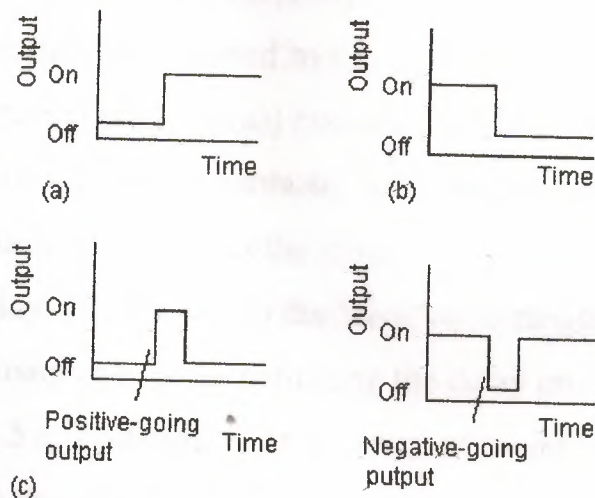


Figure 5.2 Timers: (a) on-delay, (b) off-delay, (c) pulse

5.2 Programming timers

All PLCs generally have delay-on timers, small PLCs possibly having only this type of timer. Figure 5.3 shows a ladder rung diagram involving a delay-on timer. The timer is like a relay with a coil, which is energized when the input In 1 occurs (rung). It then closes, after some preset time delay, its contacts on rung 2. Thus the output from Out 1 occurs some preset time after the input In 1 occurs.

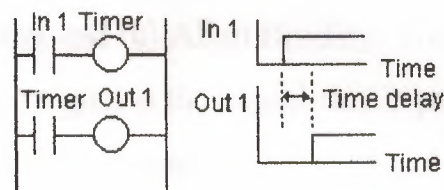


Figure 5.3 Ladder diagram involving a delay-on timer

Figure 5.4 shows how the above diagram, and program instruction, would appear when drawn in the form used by two different manufacturers of PLCs. The Mitsubishi figure (Figure 5.4(a)) considers the timer as an output, which gives a delayed time reaction to contacts. With the Siemens figure (Figure 5.4(b)), the manufacturer considers the timer to be a delay item in a rung, rather than as a relay. The symbol in the Siemens rectangle indicates a delay-on timer, the 0 coming after the T indicating the delay on. The time delay has been chosen to be 5 s. Techniques for the entry of preset time values vary. Often a constant it requires the entry of a constant K command followed by the interval by the interval in multiples of the time base used.

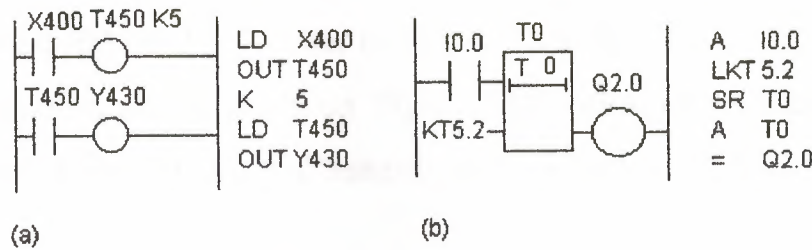


Figure 5.4 (a) Mitsubishi, (b) Siemens

Figure 5.5 shows ladder diagrams for three other manufacturers, (a) Telemecanique, (b) Toshiba and (c) Allen Bradley. With the Allen Bradley diagram the DN (for done) signal is the signal that is produced when the timer has finished its action, the EN (for enable) is the signal that is a replica of the timer input and used as instantaneous contacts.

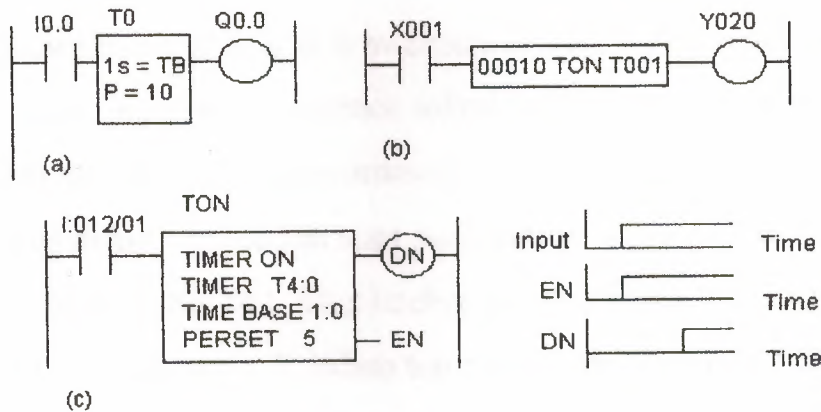


Figure 5.5 (a) Telemecanique, (b) Toshiba, (c) Allen Bradley

5.2.1 Sequencing

As an illustration of the use of a timer, consider the ladder diagram shown in Figure 5.6(a). When the input In I is on, the output Out 1 is switched on. The contacts associated with this output then start the timer. The contacts of the timer will close after the preset time delay, in this case 5.5 s. When this

happens, output Out 2 is switched on. Thus, following the input In 1, Out 1 is switched on and followed 5.5 s later by Out 2. This illustrates how timed sequence of outputs can be achieved. Figure 5.6(b) shows the same operation where the format used by the PLC manufacturer is for the timer to institute a signal delay.

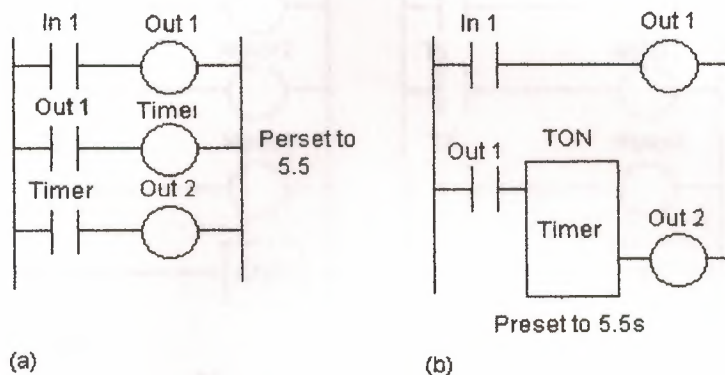


Figure 5.6 Sequenced outputs

Figure 5.7 shows two versions of how timers can be used to start three outputs, e.g. three motors, in sequence following a single start button being pressed. In (a) the timers are programmed as coils, whereas in (b) they are programmed as delays. When the start push button is pressed there is an output from internal relay IR1. This latches the start input. It also starts the timers, Ti and T2, and motor 1. When the preset time for timer 1 has elapsed then its contacts close and motor 2 starts. When the preset time for timer 2 has elapsed then its contacts close and motor 3 starts. Pressing the stop push button stops all the three motors. Since this is seen as a complete program, the end instruction has been used.

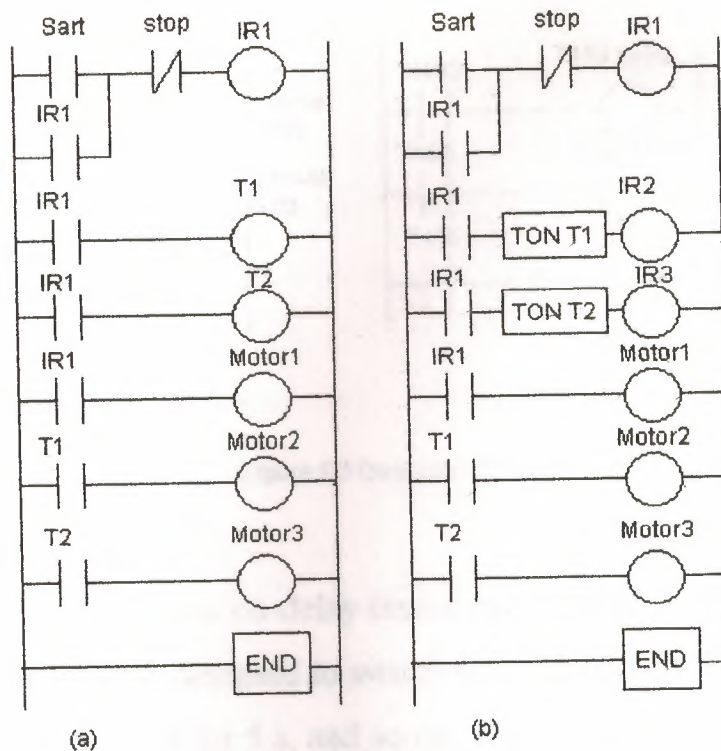


Figure 5.7 Motorsequence

5.2.2 Cascaded timers

Timers can be linked together; the term cascaded is used, to give longer delay times than are possible with just one timer. Figure 5.8(a) shows the ladder diagram for such an arrangement. Thus we might have timer 1 with a delay time of 999 5. This timer is started when there is an input to In 1. When the 999 5 time is up, the contacts for timer 1 close. This then starts timer 2. This has a delay of 100 5. When this time is up, the timer 2 contacts close and there is an output from Out 1. Thus the output occurs 1099 s after the input to In I. Figure 5. 8 (b) show the Mitsubishi version of this ladder diagram and the program instructions for that ladder.

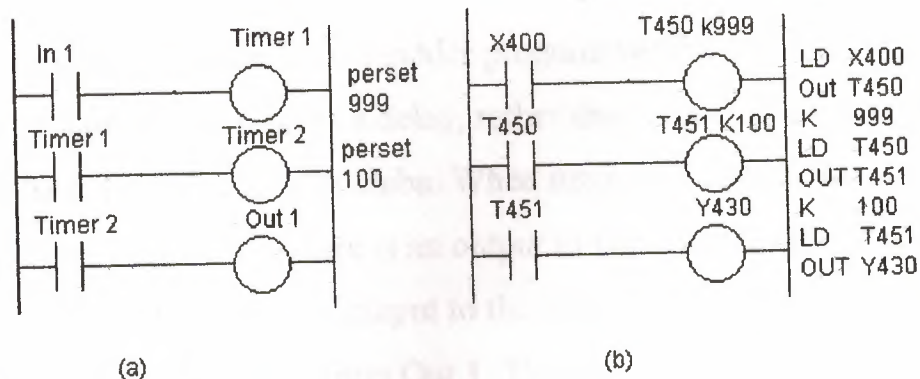


Figure 5.8 Cascaded Timers

5.2.3 On-off cycle timer

Figure 5.9 shows how on-delay timers can be used to produce an on-off cycle timer. The timer is designed to switch on an output for 5 s, then off for 5 s, then on for 5 s, then off for 5 s, and so on. When there is an input to In 1 and its contacts close, timer 1 stalls. Timer 1 is set for a delay of 5 s. After 5 s, it switches on timer 2 and the output Out 1. Timer 2 has a delay of 5 s. After 5 s, the contacts for timer 2, which are normally closed, open. This results in timer 1, in the first rung, being switched off. This then causes its contacts in the second rung to open and switch off timer 2. This result in the timer 2 contacts resuming their normally closed state and so the input to In 1 causes the cycle to start all over again.

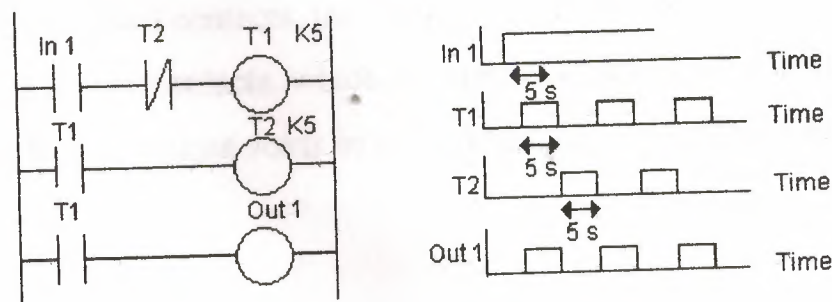


Figure 5.9 On-off cycle timer

Figure 5.10 shows how the above ladder program would appear in the format used with a timer considered as a delay, rather than as a coil. This might, for example, be with Siemens or Toshiba. When input In 1 closes, the timer T1 starts. After its preset time, there is an output to Out 1 and timer T2 starts. After its preset time there is an output to the internal relay IR1. This opens its contacts and stops the output from Out 1. This then switches off timer T2. The entire cycle can then repeat itself.

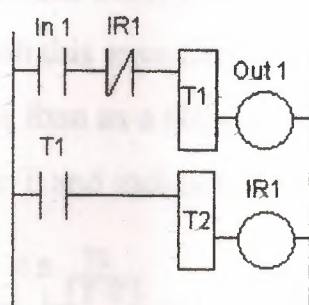


Figure 5.10 On-off cycle timer

5.3 Off-delay timer

Figure 5.11 shows how a delay-on timer can be used to produce a delay-off timer. With such an arrangement, when there is a momentary input to In 1, both the output Out 1 and the timer are switched on. Because the input is latched by the Out 1 contacts, the output remains on. After the preset timer time delay, the timer contacts, which are normally closed, open and switch off the output. Thus the output starts as on and remains on until the time delay has elapsed.

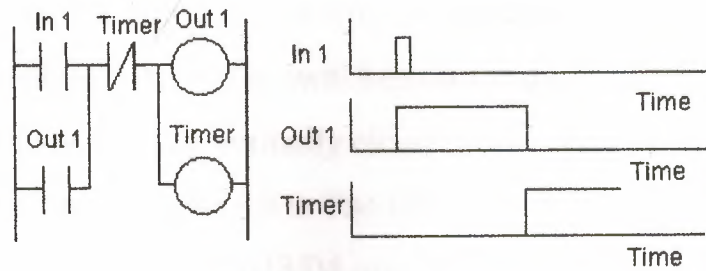


Figure 5.11 Delay-off timer

Some PLCs have, as well as on-delay timers, built-in off-delay timers and thus there is no need to use an on-delay timer to produce an off-delay timer. Figure 5.12 illustrates this for a Siemens PLC, giving the ladder diagram and the instruction list. Note that with this manufacturer, the timer is considered to be a delay item in a rung, rather than as a relay. In the rectangle symbol used for the timer, the 0 precedes the T and indicates that it is an on-delay timer.

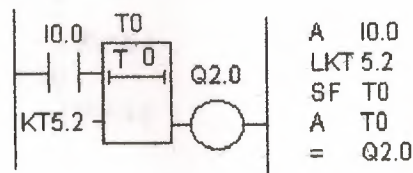


Figure 5.12 Off-delay timer

As an illustration of the use of an off-delay timer, consider the Allen Bradley program shown in Figure 5.13. TOF is used to indicate that it is an off-delay, rather than on-delay (TON) timer. The time base is set to 1:0 which is 1 s. The preset is 10 so the timer is preset to 10 s. In the first rung, the output of the timer is taken from the EN (for enable) contacts. This means that there is no time delay between an input to 1:012/01 and the EN output. As a result the EN contacts in rung 2 close immediately there is a 1:012/01 input. Thus there is an output from 0:013/01 immediately the input 1:012/01 occurs. The TT (for timer timing) contacts in rung 3 are energized just while the timer is running. Because the timer is an off-delay timer, the timer is turned on for 10 s before

turning off Thus the TT contacts will close when the set time of 10 s is running. Hence output 0:012/02 is switched on for this time of 10 s. The DN (for done) contacts which are normally closed, open after the 10 s and so output 0:013/03 comes on after 10 s. The DN contacts that are normally open close after 10 s and so output 0:013/04 goes off after 10 s.

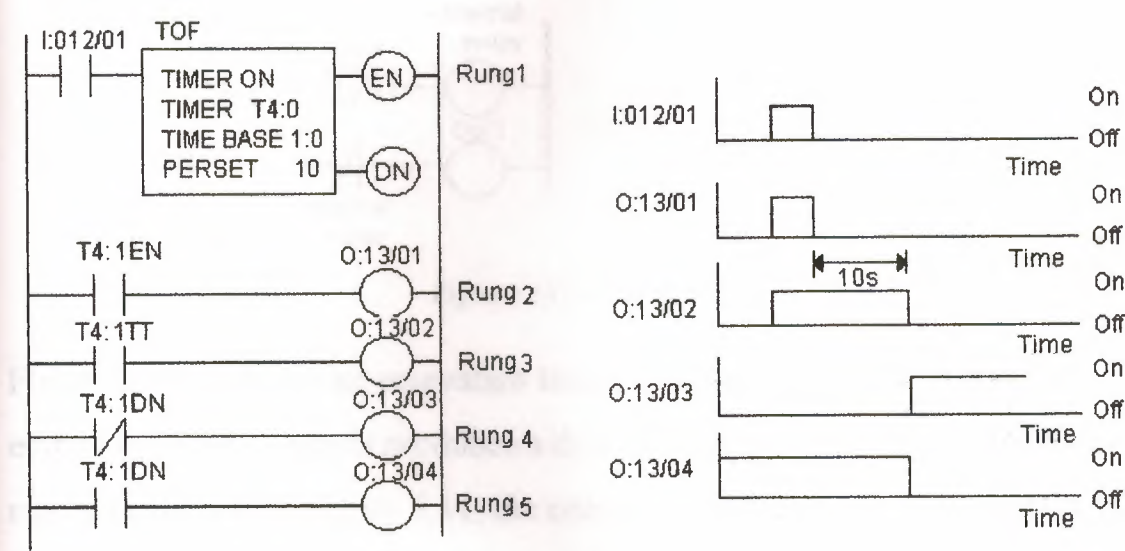


Figure 5.13 Application of an off-delay timer

5.4 On-shot timers

One-shot timers are used to produce a fixed duration output from some initiating input. Figure 5.14(a) shows a ladder diagram for a system that will give an output from Out I for a predetermined fixed length of time when there is an input to In 1, the timer being one involving a coil. There are two outputs for the input In 1. When there is an input to In 1, there is an output from Out 1 and the timer starts. When the predetermined time has elapsed, the timer contacts open. This switches off the output. Thus the output remains on for just the time specified by the timer.

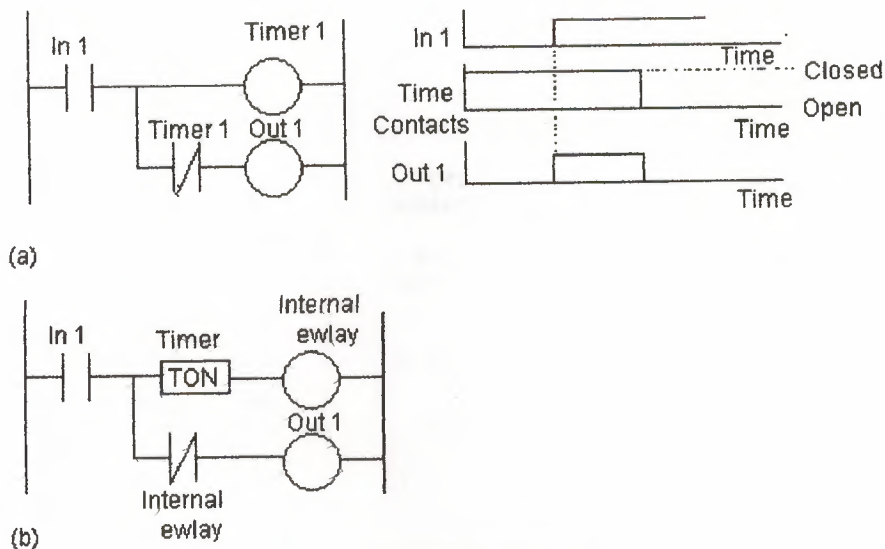


Figure 5.14 One-shot on-timer

Figure 5.14(b) shows an equivalent ladder diagram to Figure 5.13(a) but employing a timer which produces a delay in the time taken for a signal to reach the output. In Figure 5.14, the one-shot timer has an output switched on by an input for a predetermined time, then switching off. Figure 5.15 shows another one-shot timer that switches an output on for a predetermined time after the input ceases. This uses a timer and two internal relays. When there is an input to In 1, the internal relay IR 1 is energized. The timer does not start at this point because the normally closed In 1 contacts are open. The closing of the IR 1 contacts means that the internal relay IR 2 is energized. There is, however, no output from Out 1 at this stage because, for the bottom rung, we have In 1 contacts open. When the input to In 1 ceases, both the internal relays remain energized and the timer is started. After the set time, the timer contacts, which are normally closed, open and switch off IR 2. This in turn switches off IR 1. It also, in the bottom rung, switches off the output Out 1: Thus the output is off for the duration of the input, then being switched on for

a predetermined length of time.

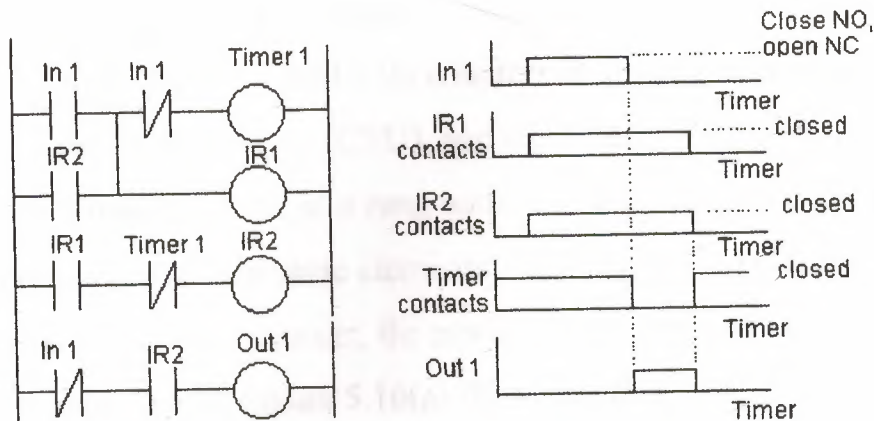


Figure 5.15 One-shot timer on, when output ceases

5.5 Counters

A counter allows a number of occurrences of input signals to be counted. This might be in a situation where items pass along a conveyor belt and a specified number have to be diverted into a box (see Figure 1.1(b)). It might be counting the number of revolutions of a shaft, or perhaps the number of people passing through a door. Counters for such applications are provided as built-in elements in PLCs. This chapter describes how such counters can be programmed.

4.5.1 Forms of counter

A counter is set to some preset number value and, when this value of input pulses has been received, it will operate its contacts. Thus normally open contacts would be closed, normally closed contacts opened. There are two types of counter, though PLCs may not include both types. These are down-counters and up-counters. *Down-counters* count down from the preset value to zero, i.e. events are subtracted from the set value. When the counter reaches the zero value, its contacts change state. Most PLCs offer

down counting. Up-counters count from zero up to the preset value, i.e. events are added until the number reaches the preset value. When the counter reaches the set value, its contacts change state.

Different PLC manufacturers deal with counters in slightly different ways.

Some count down (CTD), or up (CTU), and reset (RES) and treat the counter as though it is a relay coil and so a rung output. In this way, counters can be considered to consist of two basic elements: one relay coil to count input pulses and one to reset the counter, the associated contacts of the counter being used in other rungs. Figure 5.16(a) illustrates this. Mitsubishi is an example of this type of manufacturer. Others treat the counter as an intermediate block in a rung from which signals emanate when the count is attained. Figure 5.16(b) illustrates this. Siemens is an example of this type of manufacturer. CTD indicates the count down element, RST the reset element and CV the count value.

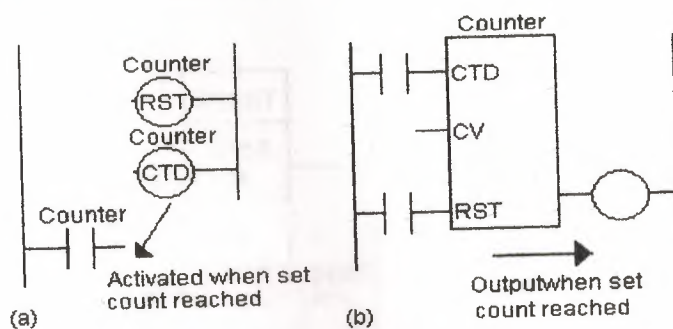


Figure 5.16 Forms of representation of counter

5.6 Programming

Figure 5.17 shows a basic counting circuit. When there is a pulse input to In 1, the counter is reset. When there is an input to In 2, the counter starts counting. If the counter is set for, say, 10 pulses, then when 10 pulse inputs have been received at In 2, the counter's contacts will close and there will be an output from Out 1. If at any time during the counting there is an input to In

1, the counter will be reset and start all over again and count for 10 pulses.

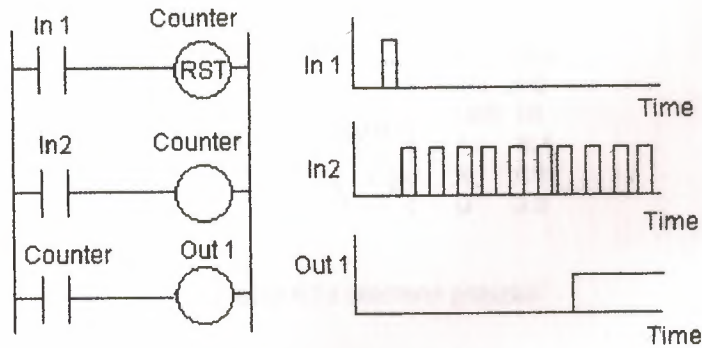


Figure 5.17 Basic counter program

Figure 5.18 shows how the above program, and its program instruction list, would appear with a Mitsubishi PLC. The reset and counting elements are combined in a single box spanning the two rungs. You can consider the rectangle to be enclosing the two counter 0 outputs in Figure 5.17. The count value is set by a K program instruction.

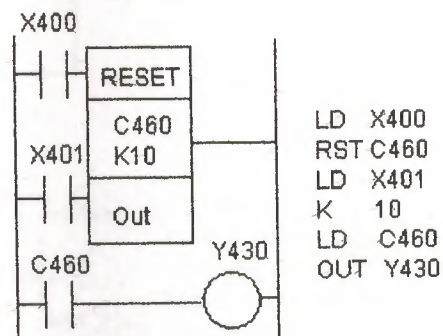


Figure 5.18 Mitsubishi program

Figure 5.19 shows the same program, and its program instruction list, with a Siemens PLC. With this ladder program, the counter is considered to be a delay element in the output line (as in Figure 5.16 (b)). The counter is reset by an input to 10.2 and counts the pulses into input 10.1. The CU indicates that it is a count-up counter; a CD would indicate a count-down counter. The counter

set value is indicated by the KC number.

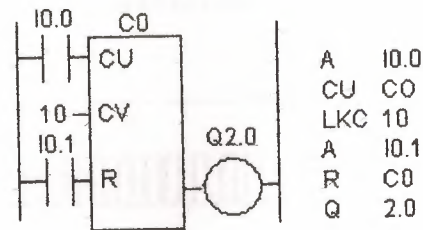


Figure 5.19 Siemens program

Figure 5.20 shows some more versions of the above ladder program for other manufacturers. Figure 5.20(a) is for Toshiba and (b) for Allen Bradley. Figure 5.21 shows how the outputs DN and CU are controlled by the count for the Allen Bradley configuration.

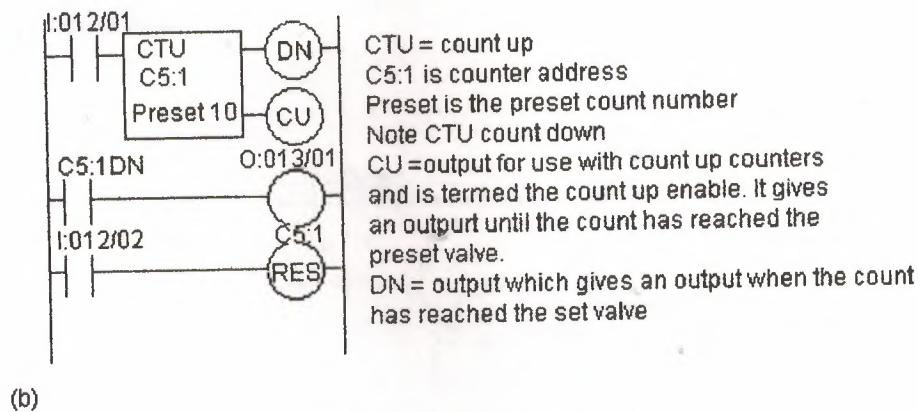
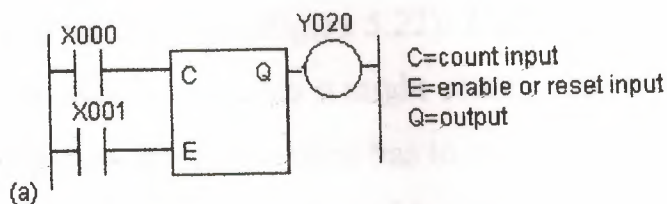


Figure 5.20 (a) Toshiba (b) allen Bradly programs

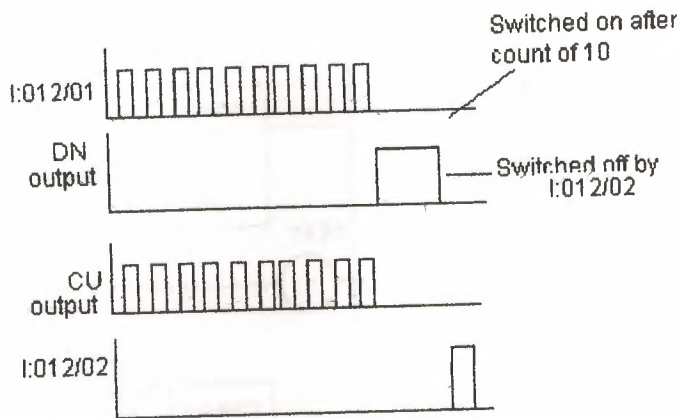


Figure 5.21 Count chart for Allen Bradley programs

5.6.1 Counter application

As an illustration of the use that can be made of a counter, consider the problem of the control of a machine, which is required to direct 6 tins along one path for packaging in a box and then 12 tins along another path for packaging in another box (Figure 5.22). A photocell sensor that gives an output every time a tin passes it might control a deflector plate. Thus the number of pulses from the sensor has to be counted and used to control the deflector. Figure 5.23 shows the ladder program that could be used. Mitsubishi notation has been used.

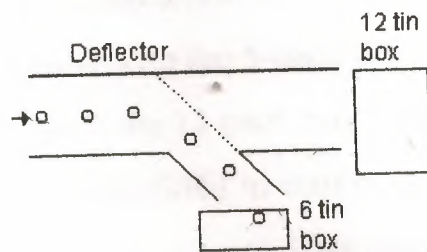


Figure 5.22 counting task

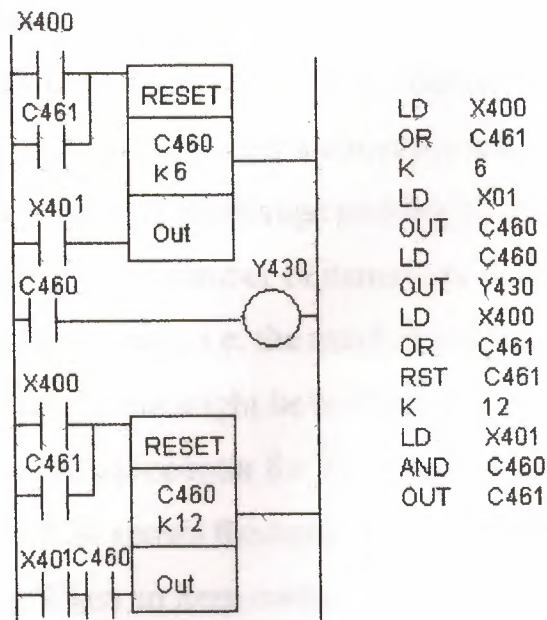


Figure 5.23 Ladder program for Figure 4.22 task

When there is a pulse input to X400, both the counters are reset. The input to X400 could be the push button switch used to start the conveyor moving. The input, which is counted, is X401. This might be an input from a photocell sensor that detects the presence of tins passing along the conveyor. C460 starts counting after X400 is momentarily closed. When C460 has counted six items, it closes its contacts and so gives an output at Y430. This might be a solenoid, which is used to activate a deflector to deflect items into one box or another. Thus the deflector might be in such a position that the first six tins passing along the conveyor are deflected into the 6-pack box, then the deflector plate is moved to allow tins to pass to the 12-pack box. When C460 stops counting it closes its contacts and so allows C461 to start counting. C461 counts for 12 pulses to X401 and then closes its contacts. This results in both counters being reset and the entire process can repeat itself.

5.7 Up and down counting

It is possible to program up- and down-counters together. Consider the task of counting products as they enter a conveyor line and as they leave it, or perhaps cars as they enter a multi-storage parking lot and as they leave it. An output is to be triggered if the number of items/cars entering is some number greater than the number leaving, i.e. the number in the parking lot has reached a 'saturation' value. The output might be to illuminate a 'No empty spaces' sign. Suppose we use the up counter for items entering and the count down for items leaving. Figure 5.24 shows the basic form a ladder program for such an application can take. When an item enters it gives a pulse on input In 1. This increases the count by one. Thus each item entering increases the accumulated count by 1. When an item leaves it gives an input to In 2. This reduces the number 1. Thus each item leaving reduces the accumulated count by 1. When the accumulated value reaches the preset value, the output Out 1 is switched on.

Figure 5.25 shows how the above system might appear for a Siemens PLC and the associated program instruction list. CU is the count up input and CD the Count down. R is the reset. The Set accumulator value is loaded via FO.0, this being an internal relay.

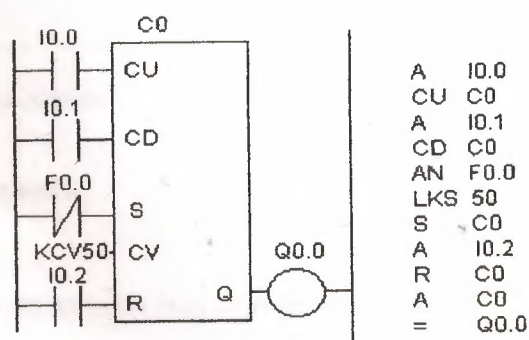
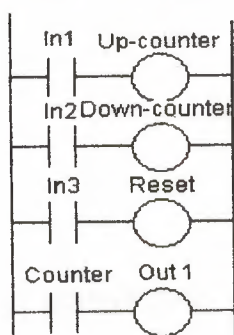


Figure 5.24 Using up-and down-counters

Figure 5.25 Up and down counting with a siemens PLC

5.8 Sequencers

The drum sequencer is a form of counter that is used for sequential control. It replaces the mechanical drum sequencer that was used to control machines that have a stepped sequence of repeatable operations. One form of the mechanical drum sequencer consisted of a drum from which a number of pegs protruded (Figure 5.26). When the cylinder rotated, contacts aligned with the pegs were closed when the peg impacted on them, and opened when the peg had passed. Thus for the arrangement shown in Figure 5.26, as the drum rotates, in the first step the peg for output 1 is activated, in step 2 the peg for the third output, in step 3 the peg for the second output, and so on. Pegs located at different distances along the drum could control different outputs. Another form consisted of a series of cams on the same shaft, the profile of the cam being used to switch contacts on and off

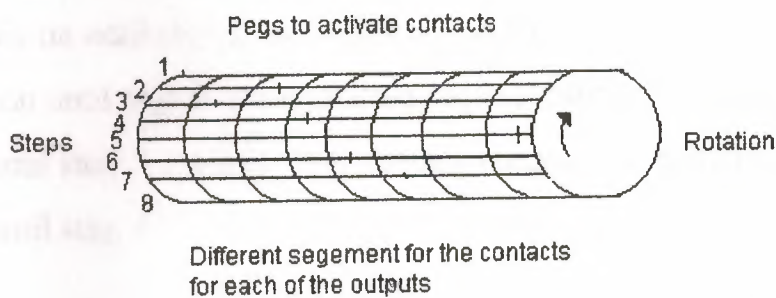


Figure 5.26 Drum sequencer

The PLC sequencer consists of a master counter that has a range of presets counts corresponding to the different steps and so, as it progresses through the count, when each preset count is reached can be used to control outputs. Each step in the count sequence relates to a certain output or group of outputs. The

outputs are internal relays, these in turn being used to control the external output devices.

Suppose we want output 1 to be switched on 5 s after the start and remain on until the time reaches 10 s, output 2 to be switched on at 10 s and remain on until 20 s, output 3 to be switched on at 15 s and remain on until 25 s, etc. We can represent the above requirements by a time sequence diagram, Figure 5.27 showing the required time sequence.

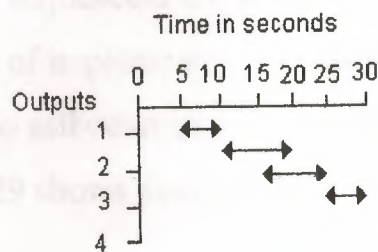


Figure 5.27 Timing diagram

We can transform the timing diagram into a drum sequence requirement.

Taking each drum sequence step to take 5 s gives the requirement diagram shown in Figure 5.28. Thus at step 1 we require output 1 to be switched on and to remain on until step 2. At step 2 we require output 2 to be switched on and remain on until step 4. At step 3 we require output 3 to be switched on and remain on until step 5. At step 5 we require output 4 to be switched on and remain on until step 6.

Set	Time(s)	Output			
		1	2	3	4
0	0	0	0	0	0
1	5	1	0	0	0
2	10	0	1	0	0
3	15	0	1	1	0
4	20	0	0	1	0
5	25	0	0	1	1
6	30	0	0	0	0

Figure 5.28 Sequencer requirements

The way in which drum sequencers are implemented depends on the manufacturer. One-way of implementing the above sequence is to use a timer, which is, programmed to self-reset every 5 s after being activated by an input being closed. Figure 5.29 shows the form such a sequencer implementation might take.

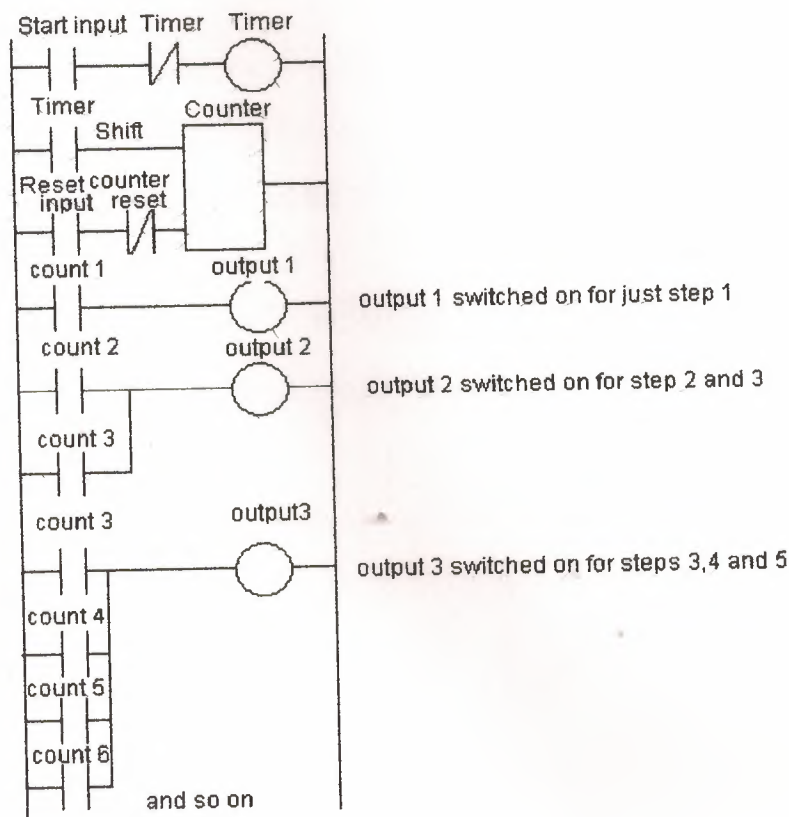
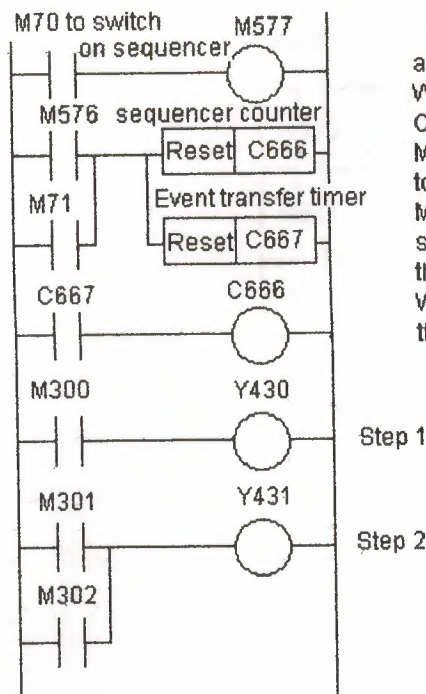


Figure 5.29 Use of a sequencer



and so on
 When M577 switched on,
 C666 and C667 used for sequencer
 M576 is used to reset the process back
 to step 1
 M71 provides a reset pulse and is a
 special relay which is activated when all
 the sequencer steps are completed
 When event time up, C666 switches to
 the next step

Figure 5.30 Sequencer with a Mitsubishi F2

With a PLC, such as a Toshiba, switching on the Step Sequence Initialize (STIZ) function block R500 (Figure 5.31) sets up the sequencer. This sets up the program for step 1 and R501. This relay then switches on output Y020. The next step is the switching on of R502. This switches on the output Y021 and also a delay-on timer so that R503 is not switched on until the timer has timed out. Then R503 switches on Y022 and also the next step in the sequence.

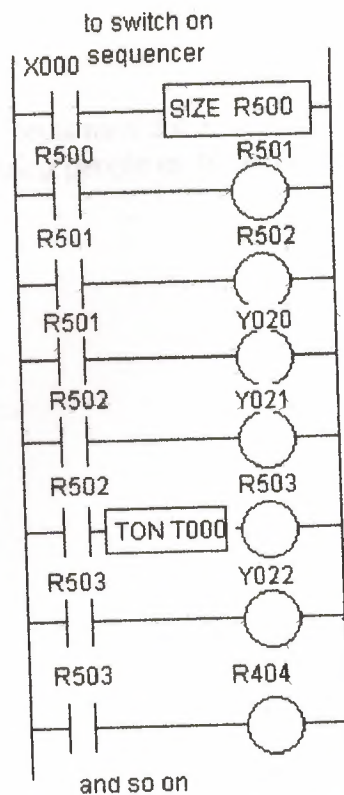


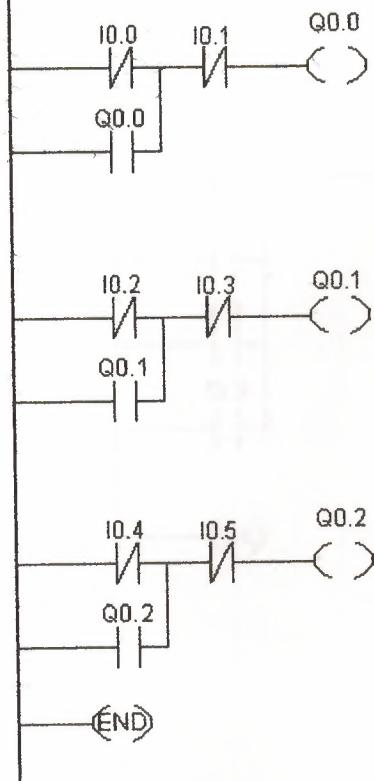
Figure 5.31 Sequencer with a Toshiba PLC

With the Allen Bradley form of PLC the sequencer is programmed by using a sequence of binary words in the form of the outputs required, e.g. those listed in Figure 5.28. Thus we would have the binary word sequence:

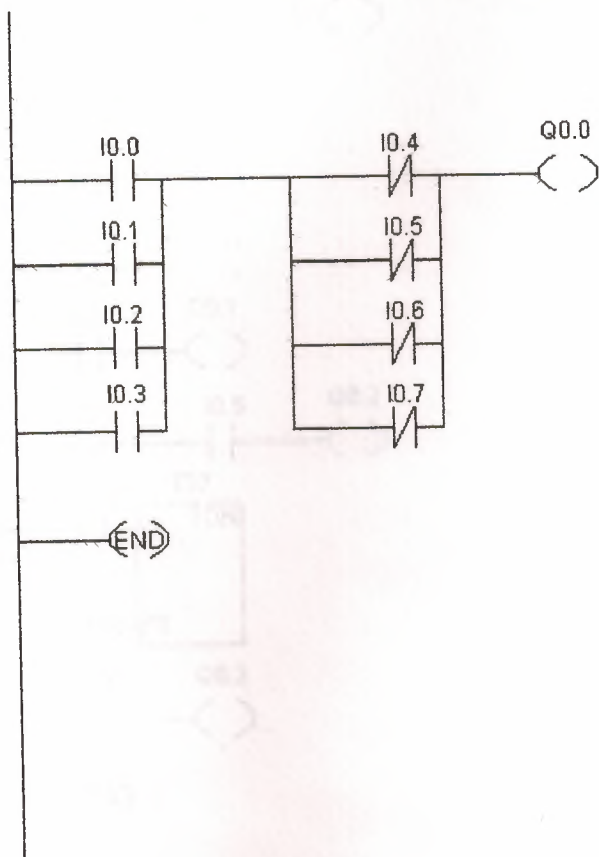
Input 4	Input 3	Input 2	Input 1
0	0	0	0
0	0	0	1
0	0	1	0
0	1	1	0
0	1	0	0
0	1	0	0
0	0	0	0

The binary words are inserted into the program using the programming device.

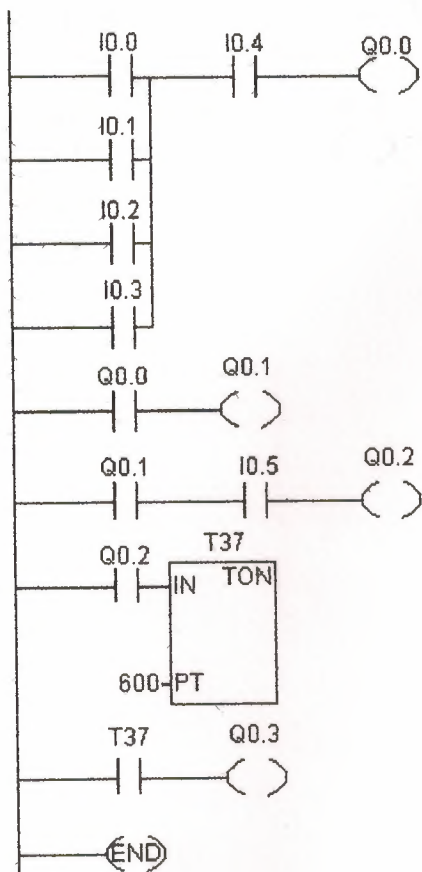
This program is illumination control for a room I0.0, I0.2, I0.4 are light's sensors and I0.1, I0.3 and I0.5 are for entering people in the three rooms.



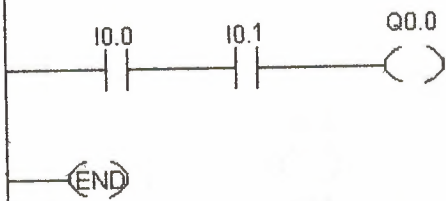
This program to pump the water from the basement to the roof tanks where I0.0 till I0.7 are water sensors that detect the level of the water in the eighth tanks four in the basement and four in the roof.



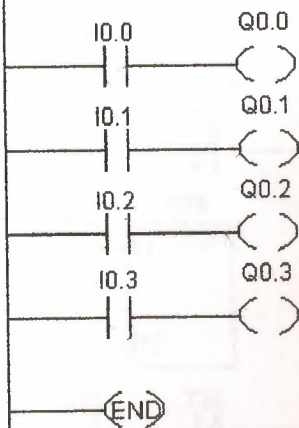
This program is alarm system that is going to make a phone message to the police I0.0, I0.1, I0.2 and I0.3 are switched are connected to the windows and I0.4 is switch to turn on/off the system.



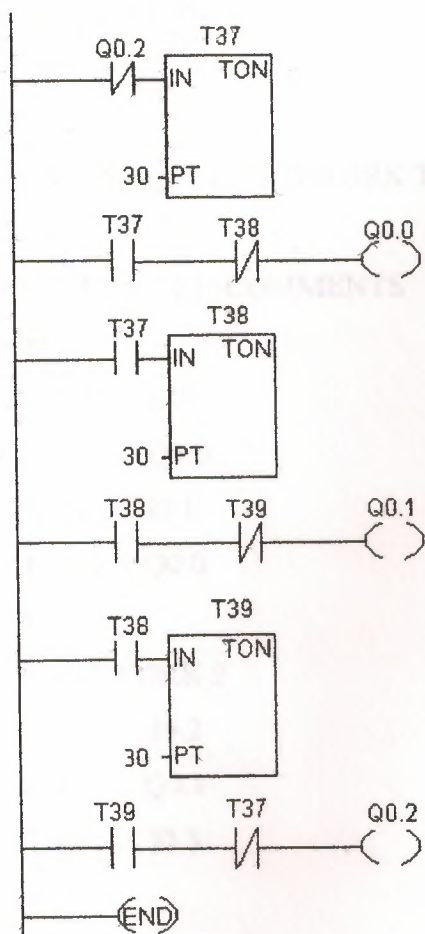
This program is heating the water flow from the roof I0.0 is a flow sensor and I 0.1 is a temperature sensor.



This program is an air conditioning program each output for each room I0.0, I0.1, I0.2 and I0.3 are temperature sensors



This program will switch 3 group of lamp each group connect in series for 3 seconds



```

1 //
2 //PROGRAM TITLE COMMENTS
3 //This program is illumination control for a room I0.0, I0.2, I0.4 are light's sensors and
  I0.1, I0.3 and I0.5 are for entering people in the three rooms.
4 //Press F1 for help and example program
5 //
6
7 NETWORK 1 //NETWORK TITLE (single line)
8 //
9 //NETWORK COMMENTS
10 //
11 LDN    I0.0
12 O      Q0.0
13 AN     I0.1
14 =      Q0.0
15
16 NETWORK 2
17 LDN    I0.2
18 O      Q0.1
19 AN     I0.3
20
21 NETWORK 3
22 LDN    I0.4
23 O      Q0.2
24 AN     I0.5
25 =      Q0.2
26
27 NETWORK 4
28 MEND

```

```
1 //
2 //PROGRAM TITLE COMMENTS
3 //This program to pump the water from the basement to the roof tanks where IO.0 till
IO.7 are water sensors that detect the level of the water in the eighth tanks four in the
basement and four in the roof.
4 //Press F1 for help and example program
5 //
6
7 NETWORK 1 //NETWORK TITLE (single line)
8 //
9 //NETWORK COMMENTS
10 //
11 LD    IO.0
12 O     IO.1
13 O     IO.2
14 O     IO.3
15 LDN   IO.4
16 ON    IO.5
17 ON    IO.6
18 ON    IO.7
19 ALD
20 =     Q0.0
21 //NETWORK2
22 MEND
```



```

1 //
2 //PROGRAM TITLE COMMENTS
3 //This program is alarm system that is going to make a phone message to the police I0.0,
  I0.1, I0.2 and I0.3 are switched are connected to the windows and I0.4 is switch to turn
  on/off the system.
4 //Press F1 for help and example program
5 //
6
7 NETWORK 1      //NETWORK TITLE (single line)
8 //
9 //NETWORK COMMENTS
10 //
11 LD    I0.0
12 O     I0.1
13 O     I0.2
14 O     I0.3
15 A     I0.4
16 =     Q0.0
17
18 NETWORK 2
19 LD    Q0.0
20 =     Q0.1
21
22 NETWORK 3
23 LD    Q0.1
24 A     I0.5
25 =     Q0.2
26
27 NETWORK 4
28 LD    Q0.2
29 TON   T37, +600

```

30

31 NETWORK 5 TITLE COMMENT

32 LD T37

33 = Q0.3

34

35 NETWORK5

36 MEND

```
1 //
2 //PROGRAM TITLE COMMENTS
3 //This program is heating the water flow from the roof I0.0 is a flow sensor and I 0.1 is a
temperature sensor.
4 //Press F1 for help and example program
5 //
6
7 NETWORK 1 //NETWORK TITLE (single line)
8 //
9 //NETWORK COMMENTS
10 //
11 LD I0.0
12 A I0.1
13 = Q0.2
14
15 NETWORK5
16 MEND
```



```

1 //
2 //PROGRAM TITLE COMMENTS
3 //This program is an air conditioning program each output for each room IO.0, IO.1,
IO.2and IO.1 are temperature sensors
4 //Press F1 for help and example program
5 //
6
7 NETWORK 1 //NETWORK TITLE (single line)
8 //
9 //NETWORK COMMENTS
10 //
11 LD IO.0
12 = Q0.0
13
14 NETWORK 2
15 LD IO.1
16 = Q0.1
17
18 NETWORK 3
19 LD IO.2
20 = Q0.2
21
22 NETWORK 4
23 LD IO.3
24 = Q0.3
25 NETWORK5
26 MEND

```

```
1 //
2 //PROGRAM TITLE COMMENTS
3 // This program will switch 3 group of lamp each group connect in series for 3 seconds
4 //Press F1 for help and example program
5 //
6 NETWORKS
7 NETWORK 1    //NETWORK TITLE (single line)
8 //
9 //NETWORK COMMENTS
10 //
11 LDN    Q0.2
12 TON    T37, +30
13
14 NETWORK 2
15 LD      T37
16 AN      T38
17 =       Q0.0
18
19 NETWORK 3
20 LD      T37
21 TON     T38,30
22 NETWORK 3
23 LD      T38
24 AN      T39
25 =       Q0.1
26
27 NETWORK 4
28 LD      T38
29 TON     T39, 30
30
```

31 NETWORK 5

32 LD T39

33 AN T37

34 = Q0.2

35

36 NETWORK5

37 MEND

CONCLUSION

When developing this project we see that PLC is making the operation in industrial places, and that's the reasons, why its gaining interest, Notice of most of the companies.

With the information observed from the lecturer and our researchers for this topic PLC, is a convenient tool wit a wide range of useful ways to be used. Such examples can be mentioned several machines cab be used at the same time, easy adjustments from the PLC program can be made within a few minutes by the keyboard, installed PLC programs can be controlled or checked before within the office laboratory, even the PLC program as for firm can be meet at the home. Its very protective and safe for the workers which they protected the danger, communication programs of PLC's within each other or within operates can happen with the PLC; the developed lantues have constructed the productivity, security establishment security fast productivity, quality and we can see that PLC is a very cheap program that can be fundamentally used .

In this project we have used automation of electric application by PLC. Where for each flat we should use one device of PLC.

REFERENCES

- [1] Hugh Jack, Programmable Logic Controllers, (June 1999).
- [2] ALAN J CRISPIN, Programmable logic controller and their engineering applications.
- [3] AIAN G. WARNOCK, Programmable controllers operation and application.