NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Electrical and Electronic Engineering

PROGRAMMABLE LOGİC CONTROLLER (PLC)

Graduation Project
EE- 400

Student: Hasan Ali Al-Hajouj

Instructor: Mr. Özgur Özerdem

Nicosia-2003

ACKNOWLEDGEMENTS

First of all, I would like to say how grateful I am to my supervisor, Dr .Özgur Ozerdem , friends, parents and brothers. I could not have prepared this Graduation Project without the generous help of Mr. Cemal Kavalcıoğlu.

I would like to thank my supervisor Dr. Özgür Özerdem Under his guidance, I successfully overcome many difficulties and learn a lot about PLCs . I asked him many questions in plc, he explained my questions patiently.

I would like to express my gratitude to Prof. Dr. Şenol Bektaş and Dr. Tayseer Al-Shanableh and prof.Dr Fakhreddin Mamedov for them because they helped to me at each stage of my Undergraduate Education in Near East University.

I also wish to thank my advisor Ass.Prof.Dr.kadr at my Undergraduate Education for his invaluable advices, for his help and for his patience also for his support.

Finally, I want to thank my family, especially my parents without their endless support, I could never have prepared this thesis without the encouragement and support of my parents.

TABLE OF CONTENTS

Chapter 1

1. Programmable Logic Controllers Sensors and Actuators

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 .

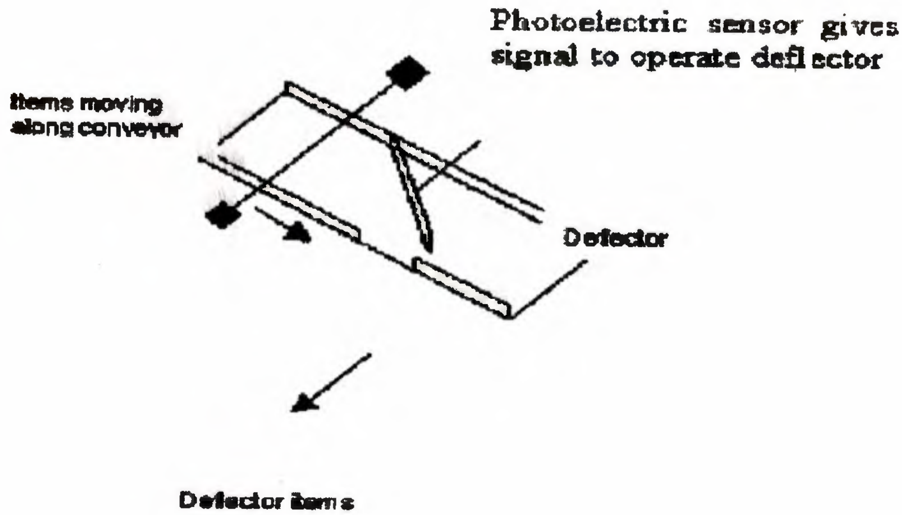


Figure 1.1 an example of a control task and some input sensors, (a) an automatic drilling machine, (b) a packing system .

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.

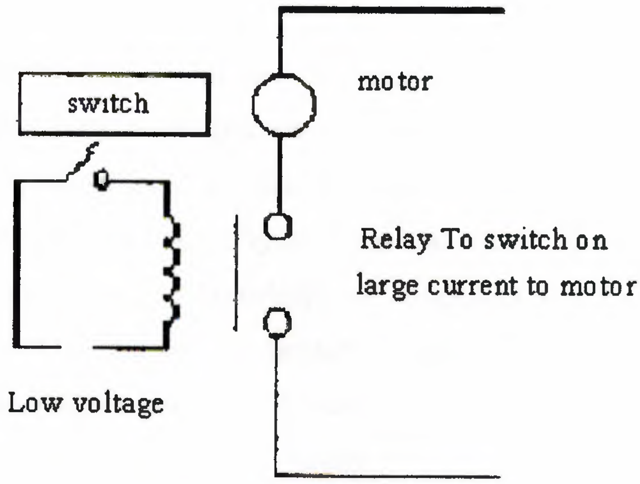


Figure 1.2 a control circuit

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.

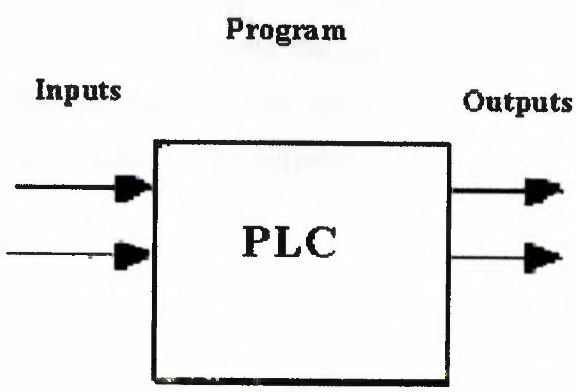


Figure 1.3 programmable logic controllers

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 as 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

1. 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.*

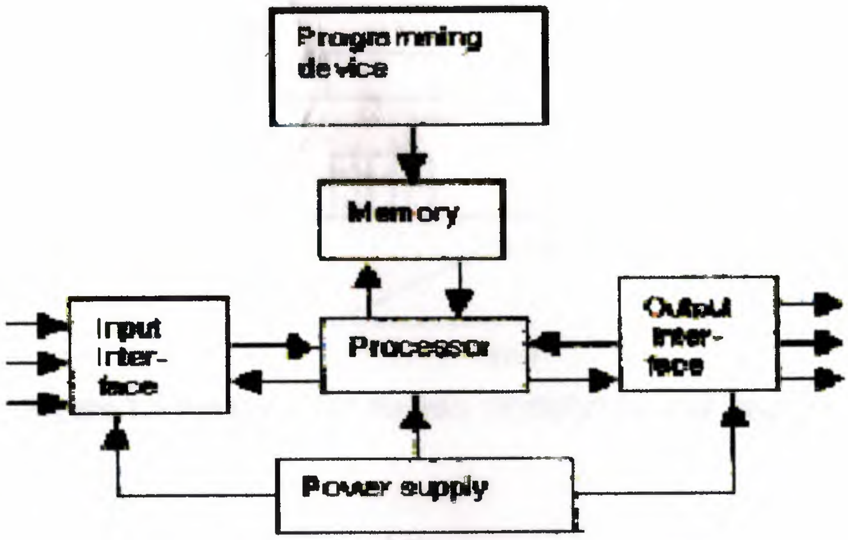


Figure 1.4 The plc system

2. The power supply unit is needed to convert the mains a.c. voltage to the low d.c. voltage (5 V) necessary for the processor and the circuits in the input and output interface modules.
3. 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.
4. The memory unit is where the program is stored that is to be used for the control actions to be exercised by the microprocessor.
5. 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.

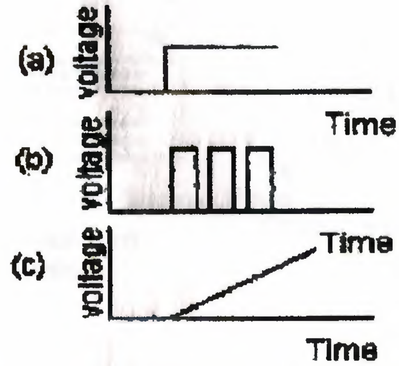


Figure 1.5 Signales: (a) discrete (b)digital (c)analogue

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.

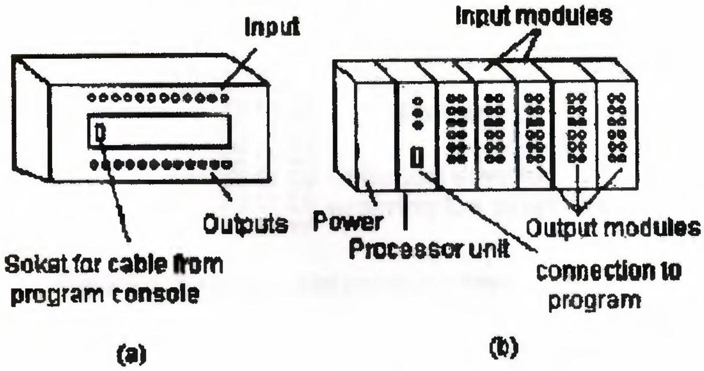


Figure 1.6 (a) single box, (b) modular/rail type

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.

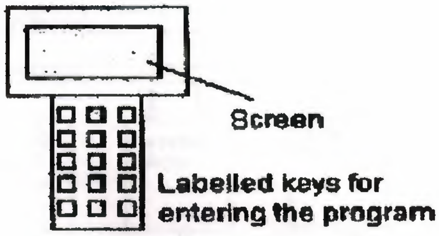


Figure 1.7 Hand-held programmer

1.2 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 I and 8 MIIz. 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.

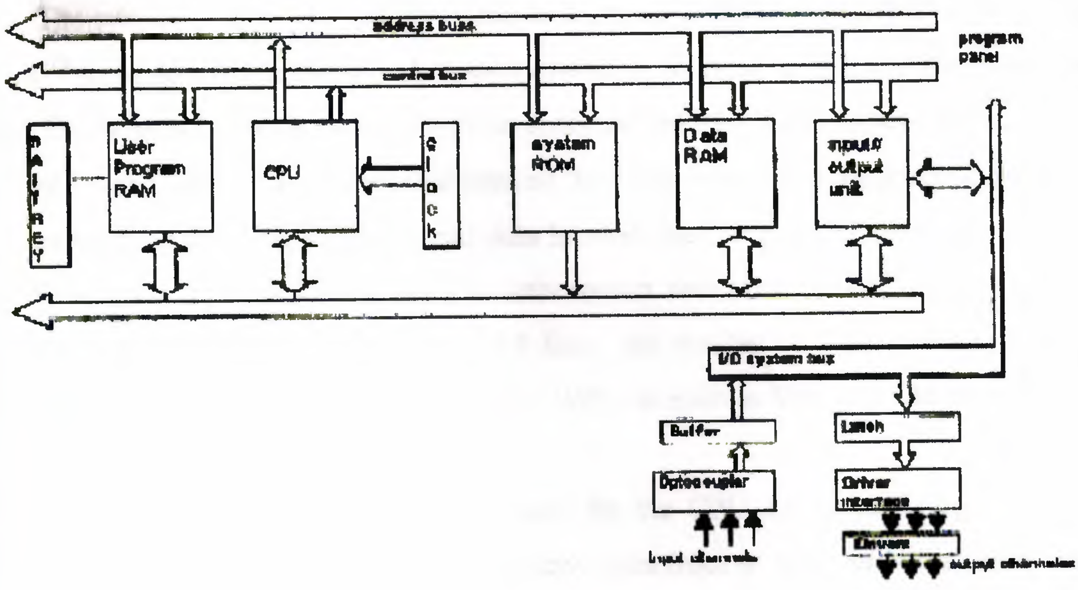


Figure 1.8 architecture of plc

1.3.1The 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 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 data stored at a particular location can be accessed by the CPU 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.

The programs and data in RAM can be changed by the user. All PLCs will have some amount of RAM to store programs that have been developed by the user and program data. However, to prevent the loss of programs when the power supply is switched off, a battery is used in the PLC to maintain the RAM contents for a period of time. After a program has been developed in RAM it may be loaded into an EPROM memory chip, often a bolt-on module to the PLC, and so made permanent. In addition there are temporary buffer stores for the input/output channels.

The number of binary words that it can store determines the storage capacity of a memory unit. Thus, if a memory size is 256 words then it can store 256 x 8 = 2048 bits if 8-bit words are used and 256 x 16= 4096 bits if 16-bit words are used. Memory sizes are often specified in terms of the number of storage locations available with 1K representing the number 2^10, i.e. 1024. Manufacturers supply memory chips with the storage locations grouped in groups of 1, 4 and 8 bits. A 4K x 1 memory has 4 X I x 1024 bit locations. A 4K x 8 memory has 4 x 8 x 1024 bit locations. The term byte is used for a word of length 8 bits. Thus the 4K x 8 memory can store 4096 bytes. With a 16-bit address bus we can have 216 different addresses and so, with 8-bit words stored at each address, we can have 216 x 8 storage locations and so use a memory of size 216 x 8/210 = 64K x 8 which we might have in the form of four 16K x 8 bit memory chips

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.

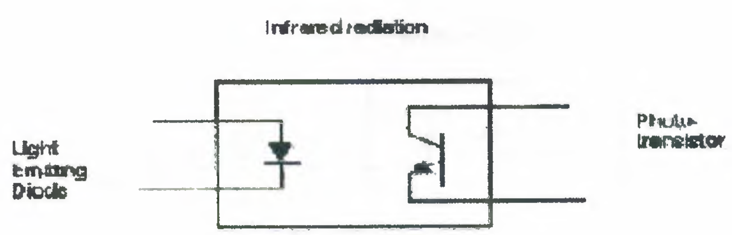


Figure 1.9 Optoisolator

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.

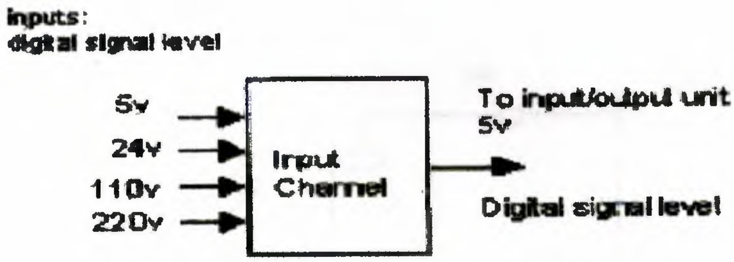


Figure 1.10 Input levels

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 photo transistor 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 trice type.

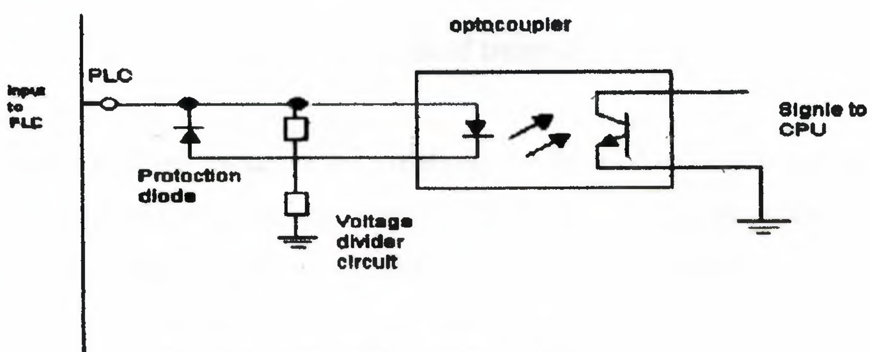


Figure 1.11 Basic d.c. input circuit

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.

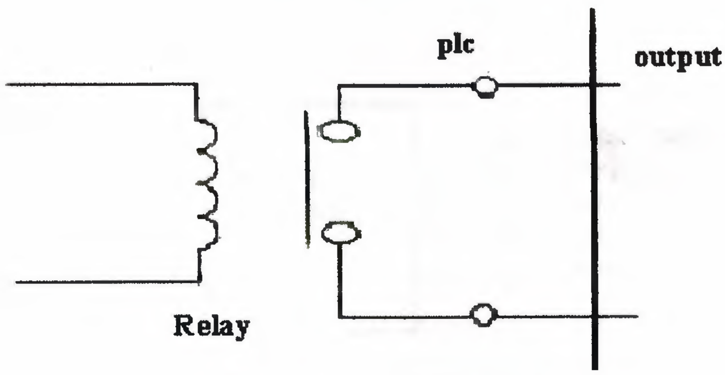


Figure1.12 relay output

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. Opt isolators are used to provide isolation. Figure 1.13 shows the basic form of such a transistor output channel.

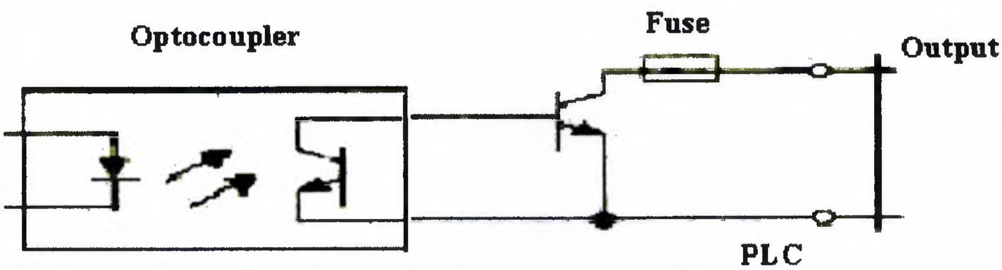


Figure 1.13 Basic form of transistor output

3- Triac outputs, with opt isolators 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 ac. 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.

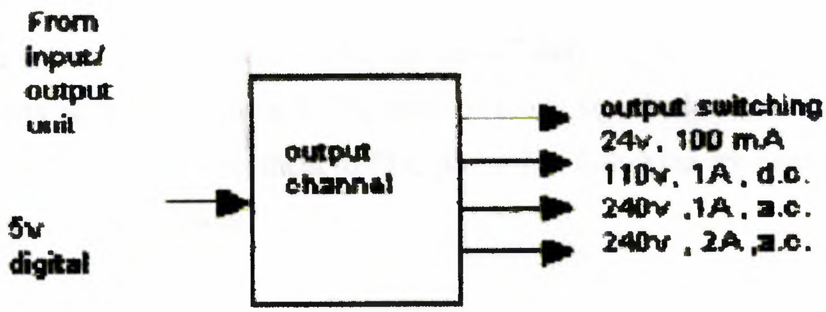


Figure 1.14 output levels

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 work piece 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.

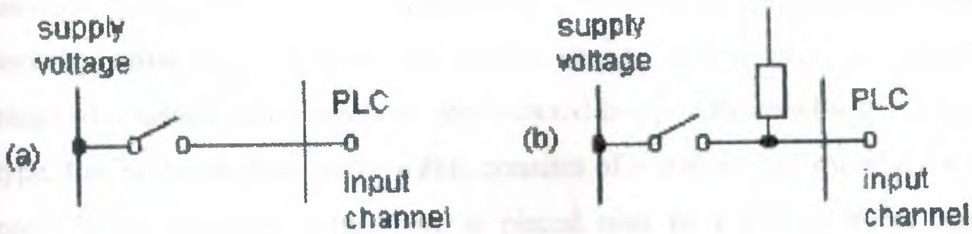


Figure 2.1 Switch sensor

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.

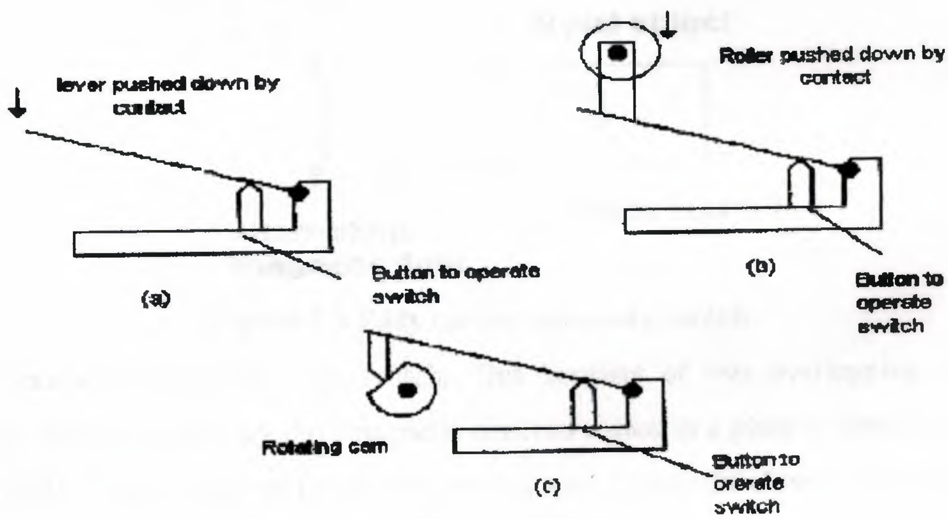


Figure 2.2 Limit switches actuated by. (a) Lever. (b) Roller. (c) Cam

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.

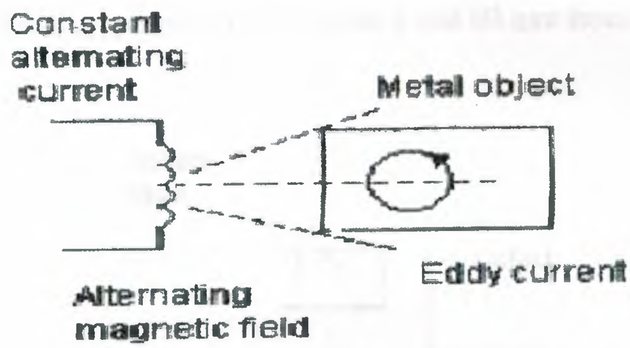


Figure 2.3 Eddy current proximity switch

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.

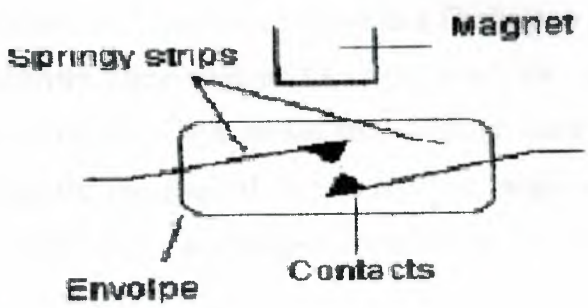


Figure 2.4 Read switch

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.

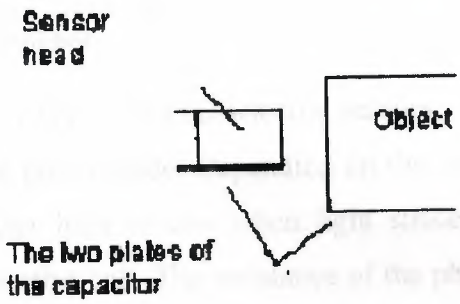


Figure 2.5 Capacitive proimity switch

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.

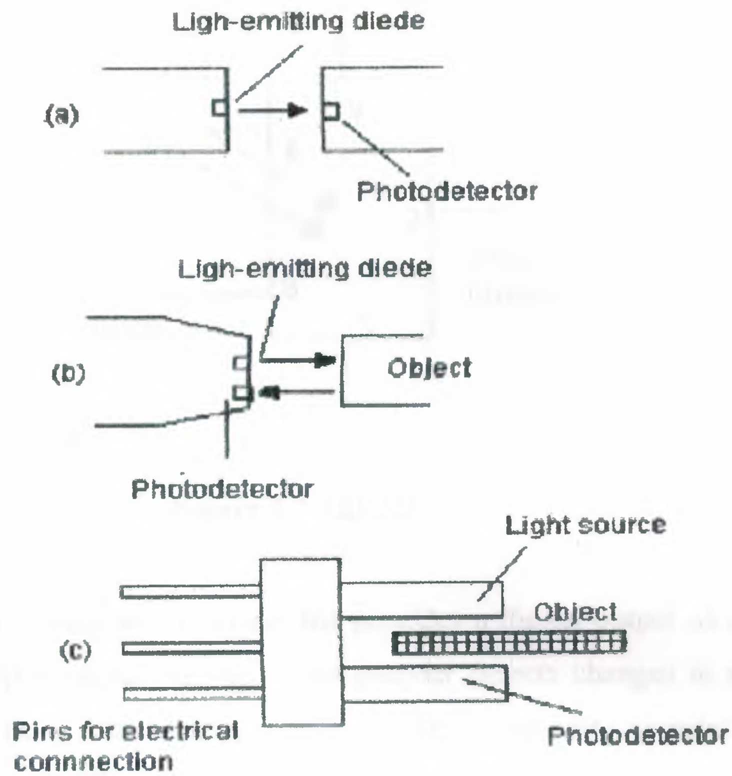


Figure 2.6 Photoelectric sensors

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 photo conductive cell. The resistance of the photo conductive 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 TSL22O, incorporating the light sensor and the voltage-to-frequency converter (Figure 2.7).

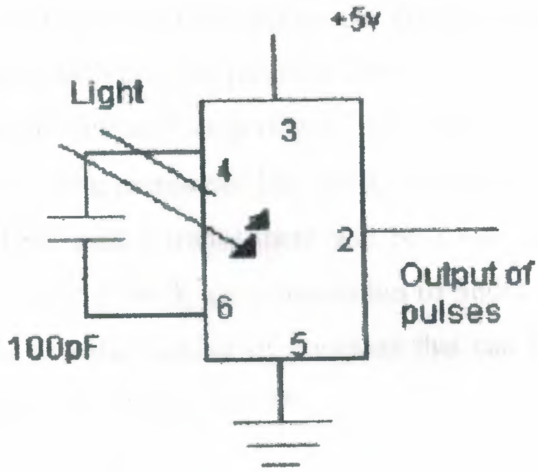


Figure 2.7 TSL220

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 3600, a movement from one slot to the next is a rotation of 60. By using offset slots it is possible to have over a thousand slots for one revolution and so much higher resolution.

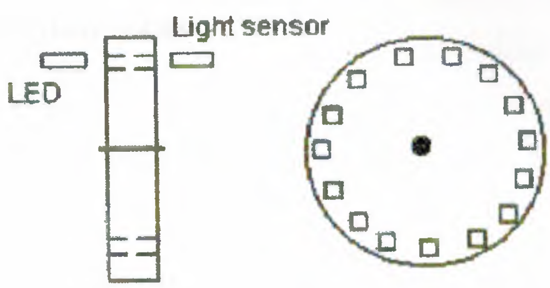


Figure 2.8 Basic form of an incremental encoder

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.

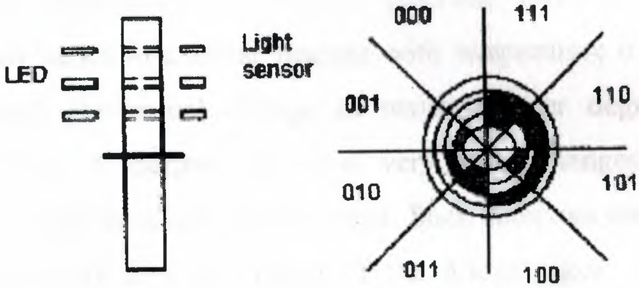


Figure 2.9 A 3_bit absolute encoder

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.

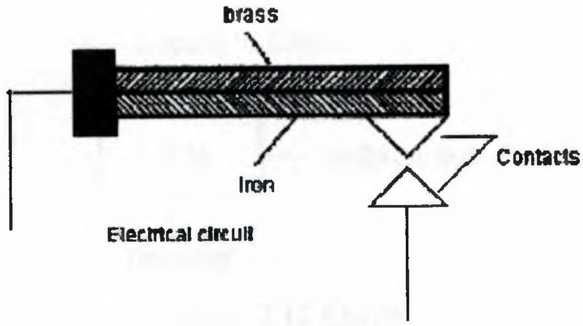


Figure 2.10 Bimetallic strip

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 an alloy, 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 Wheat stone 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.

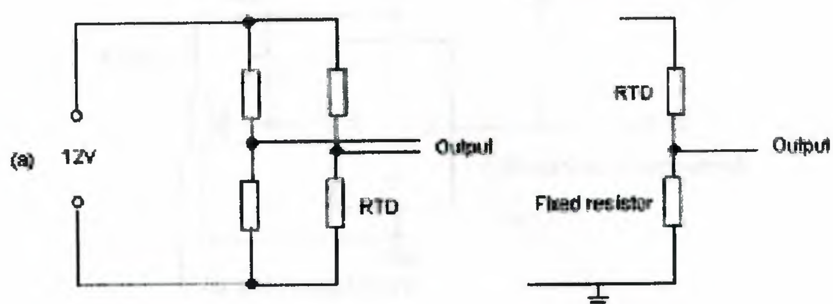


Figure 2.11 a) Wheatstone bridge, b) potential divider circuits

Thermo diodes and thermo transistors 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 mV/^{0}C when the supply voltage is +5 V (Figure 2.12).

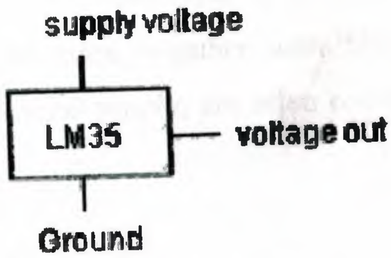


Figure 2.12 LM35

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 thermo transistor 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.

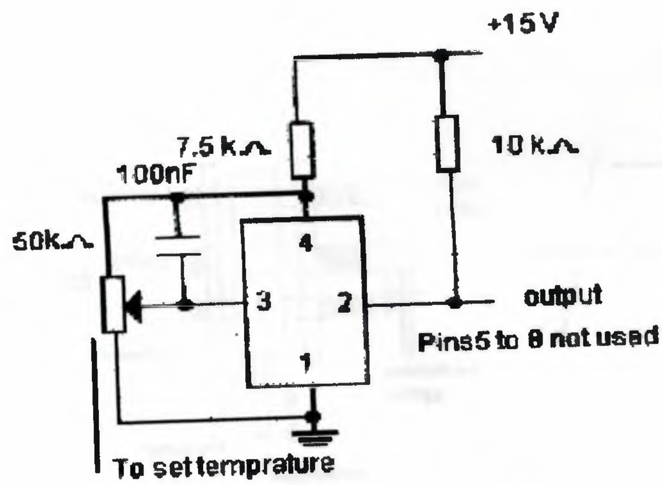


Figure 2.13 LM 11N circuit for on _of control

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.

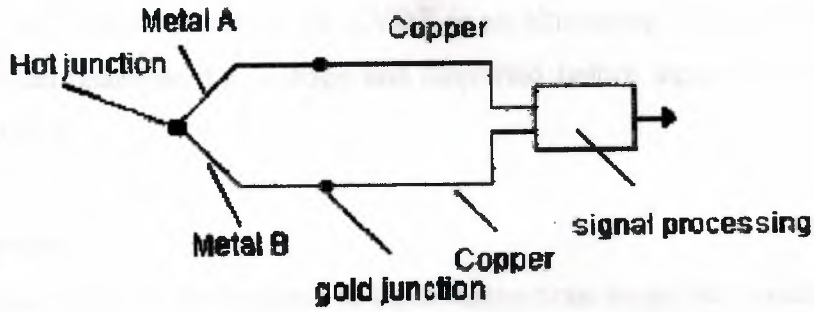


Figure 2.14 Thermocouple

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.

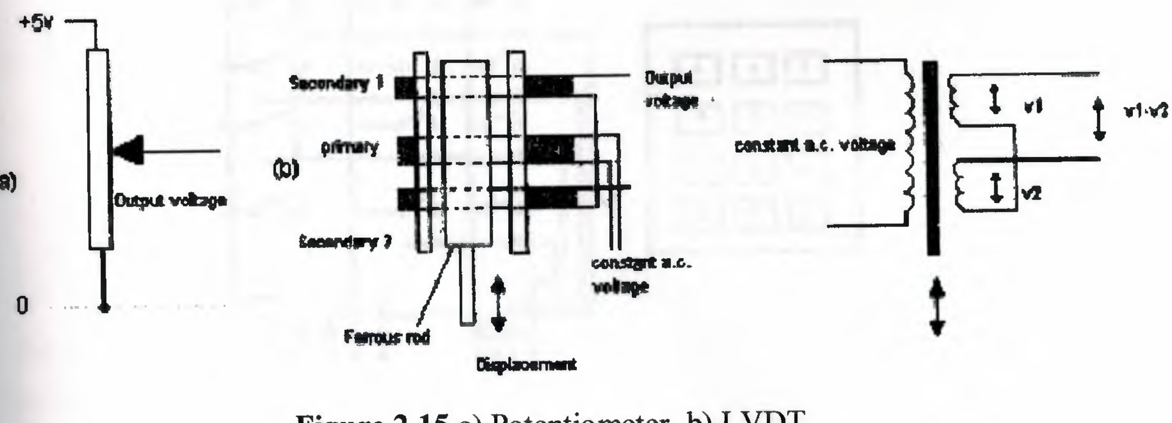


Figure 2.15 a) Potentiometer, b) LVDT

Another form is 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 .

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 cent red 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. 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 2.16) 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.

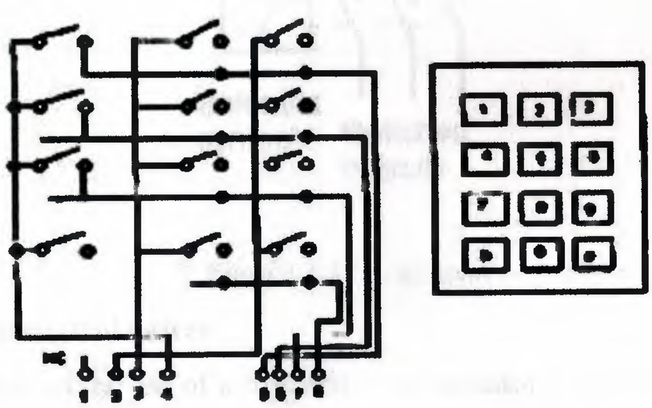


Figure 2.16 Way keypad

2.2. Output devices

The output ports of a PLC are of the relay type or opt isolator 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 2.17). 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.

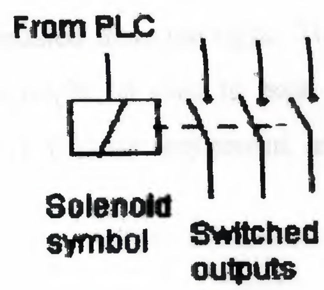


Figure 2.17 Contactor

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 2.18) shows one such form, a spool valve, used to control the movement of a piston in a cylinder.

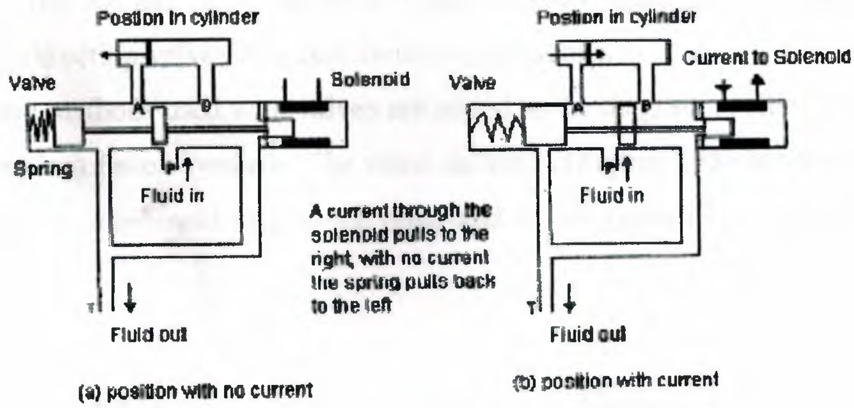


Figure 2.18 A example of solenoid operated vale

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 2.18(a)) the hydraulic fluid of 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 them 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.

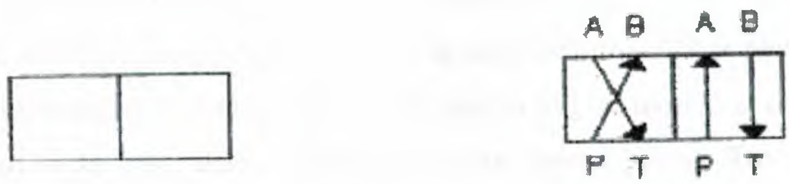


Figure 2.19 (a) Two position valve, (b) the 4/2 valve

With the above valve there are the two control positions shown in (Figure 2.18(a)) and (b). The number of ports they have and the number of control positions describes directional control valves. The valve shown in (Figure 2.18) 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 2.18) consists of two squares (Figure 2.19). Within each square the switching positions are then described by arrows to indicate a flow direction or a terminated line to indicate no flow path. (Figure

2.20) shows this for the valve shown in (Figure 2.18). (Figure 2.21) shows some more examples of direction valves and their switching positions.

The actuation methods used with valves are added to the diagram symbol; (Figure 2.22) shows examples of such symbols. The value shown in (Figure 2.18) has a spring to give one position and a solenoid to give the other and so the symbol is as shown in (Figure 2.23).

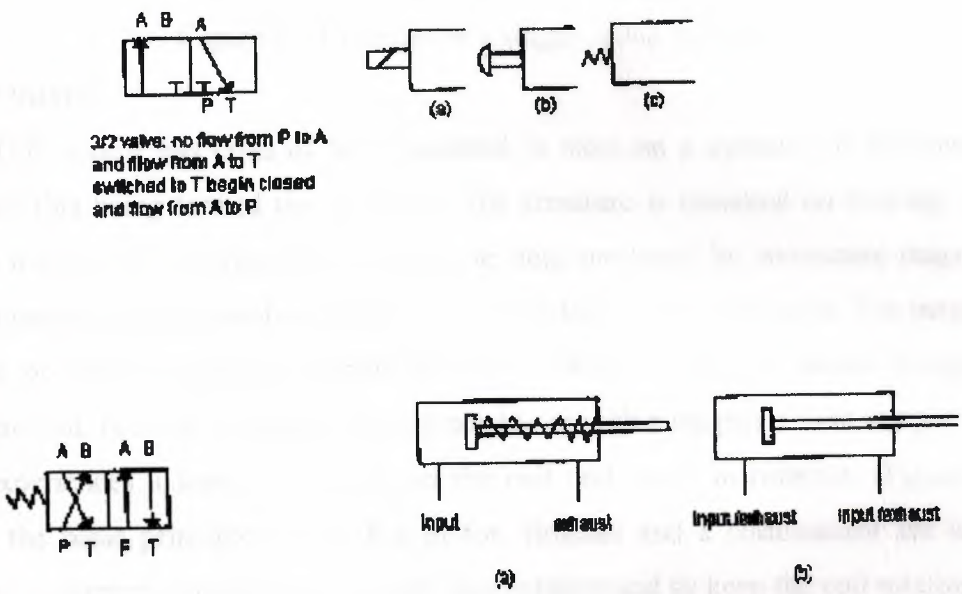


Figure 2.20 The 4/2 valve, Cylinders: (a) single action, (b) double action

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 2.20(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 2.20(b)) is used when the cylinder is powered by fluid for its motion in both piston movement directions. (Figure 2.21) shows how a valve can be used to control the direction of motion of a piston in a single-acting cylinder;

(Figure 2.22) shows how two valves can be used to control the action of a piston in a double acting cylinder.

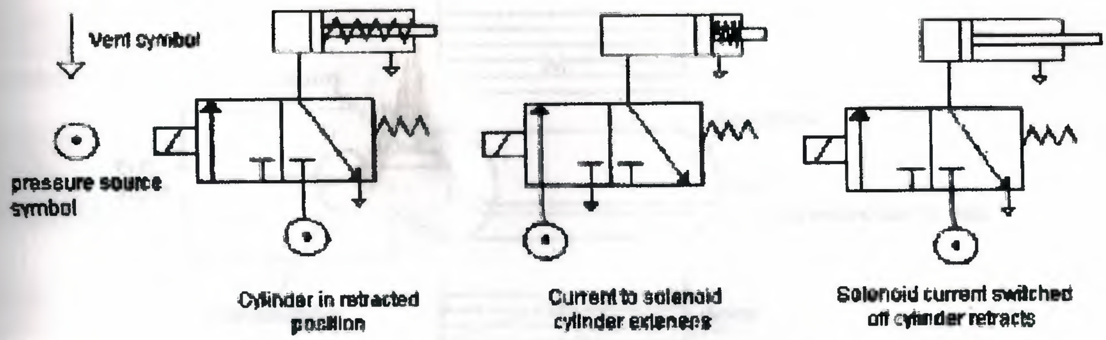


Figure 2.21 Control of a single acting cylinder

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 2.23) 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.

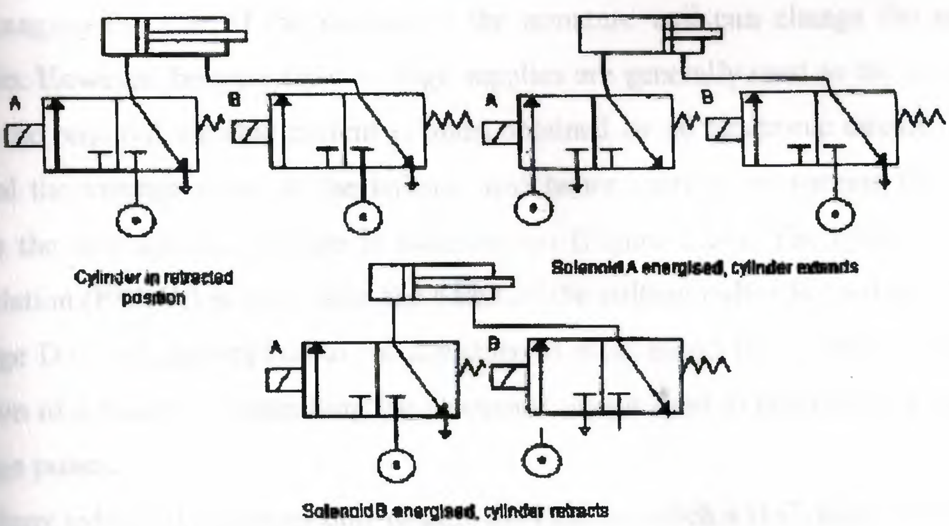


Figure 2.22 Control of double acting cylinder

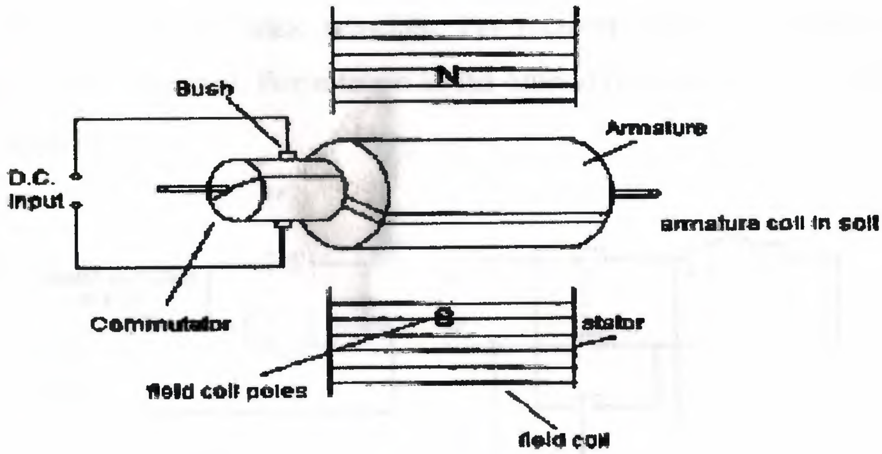


Figure 2.23 Basic element of a d.c motor

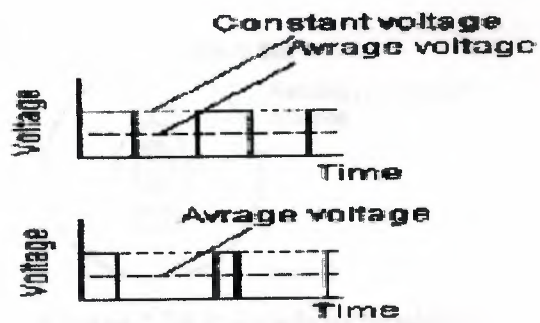


Figure 2.24 pulse width modulation

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 2.24). The term pulse width modulation (PW1Vf) 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 2.25 (a)) 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 2.25 (b) 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

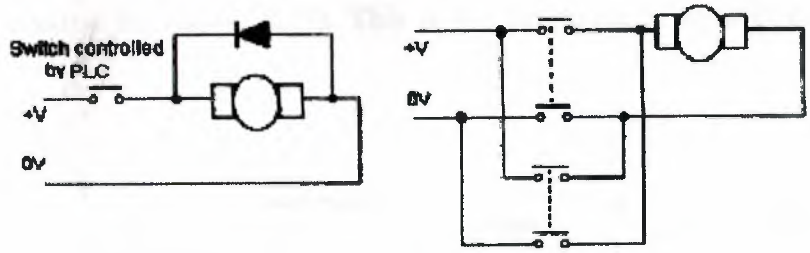


Figure 2.25 (a) On-of control, (b) Direction control for a d.c motor

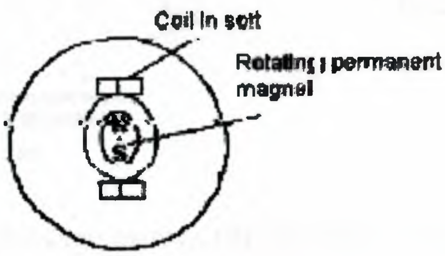


Figure 2.26 Principle of brushless

Another form of D.C. motor is the brush less 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 2.26) 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 brush less 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 2.24.

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 2.27). This is the single-phase squirrel-cage induction motor.

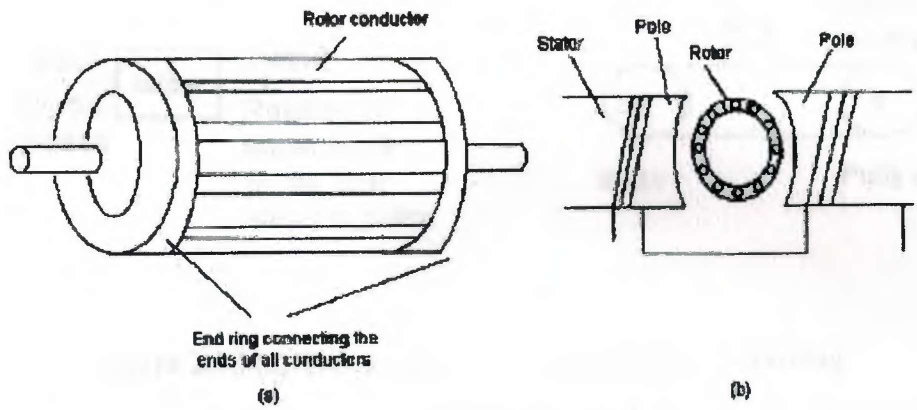


Figure 2.27 (a) Squirrel cage motors, (b) The motor with a single phase stator

The rotor is the squirrel cage of copper or aluminum bars fitting into slots in end rings to a 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 current flow through them. We thus have currents 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 alternating current applied to the stator. One way of varying the speed of the rotation is to use an electronic circuit to control the frequency of the current supplied to the stator. Though a.c. motors are cheaper, more rugged and more reliable than d.c. motors, the maintaining of constant speed and controlling that speed is generally more complex than with d.c. motors. As a consequence, D.C. motors, particularly brush less permanent magnet motors, tend to be more widely used for control purposes.

2.2.4. Stepper motors

The stepper motor or stepping motor is a motor that produces rotation through equal angles, that so-termed step, for each digital pulse supplied to its input (figure 2.28(a)).

Thus, if one input pulse produces a rotation of 1.8 degree then 20 such pulses would give rotation of 36 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's 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.

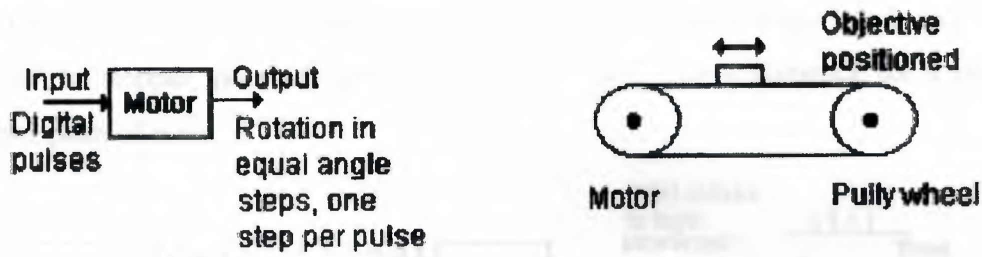


Figure 2.28 (a) The stepping motor, (b) Linear positioning

There are a number of forms of stepping motor. (Figure 2.29) 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 coil wrapped rounds 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 rotate 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.

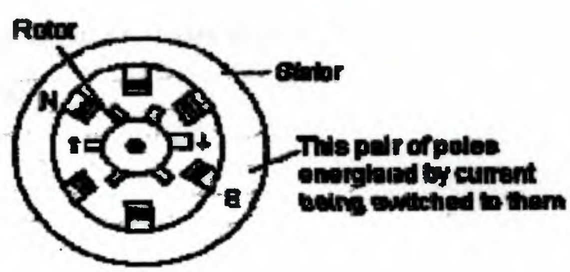


Figure 2.29 The principle of the variable reluctance stepper motor

upwards or 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 pressed at upper level to send the lift downwards or a push button is pressed at the lower level to request the life to move downwards but in both cases there is a condition that has to be met that a limit switch in indicates that the access gate to the life platform is closed. Thus the inputs switch and limit switches. The output from the control system in 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, and 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 singles 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 timer to keep the door open for the required time. The output from controller might be to solenoid operated pneumatic valves that used movement of a distance in cylinder to open and close the door. (Figure 2.33) 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 invented 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 invented side to a vent to the atmosphere. A similar arrangement is used from opening the door auto wards.

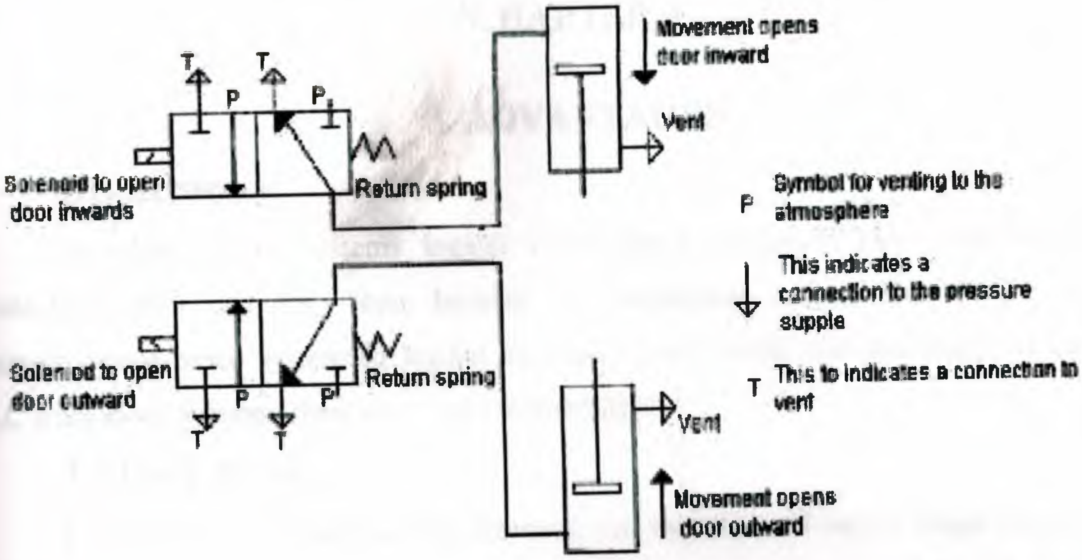


Figure 2.33 Pneumatic door opening system

CHAPTER 3

3. ADVANTAGES

3.1. Accuracy

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

3.2 Data Areas

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

The following table shows the identifiers and ranges for each of the data area memory types:

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

3.3 Logic Control of Industrial Automation

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

3.4 Data Object

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

The following table shows the identifiers and ranges for each of the data object memory types:

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

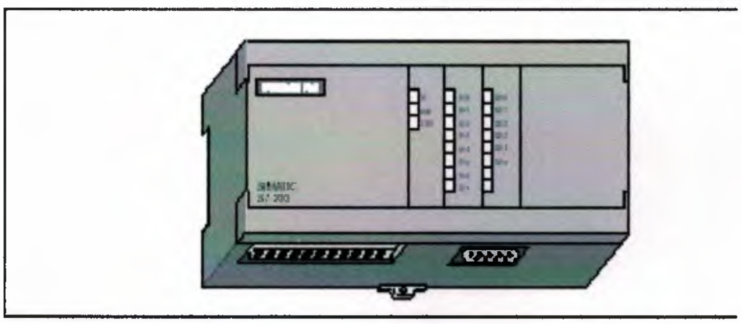


Figure 3.1 S7-212 CPU Module

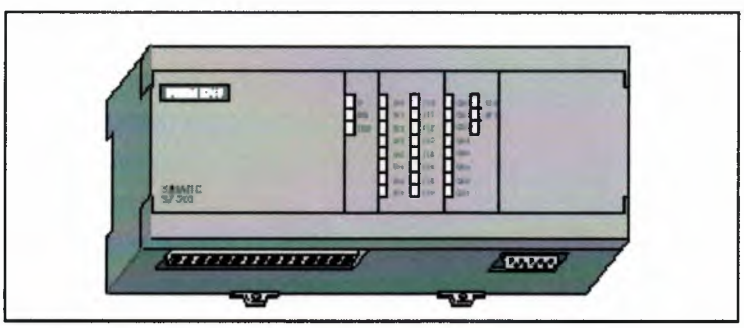


Figure 3.2 S7-214 CPU Module

3.5 Flexibility

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

3.6 Communication

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

CHAPTER 4

4. LADDER AND STL PROGRAM

4.1. Ladder Programs

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

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

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

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

4.2 STL Programs

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

4.3 CPU Memory

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

- Program memory stores the user program.

- Data memory includes a temporary area for the program and storage of data. The temporary storage, calculations, and constants reside in data memory. Additionally, data for timers, counters, high-speed counters, and analog inputs and outputs are stored in data memory.

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

4.4 Simatic S7-200 Application Areas

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

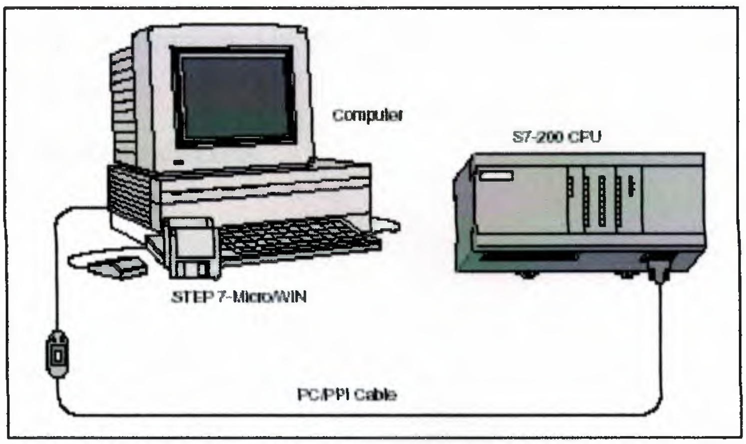


Figure 4.2 Components of an S7-200 Micro PLC System

4.4.1 The S7-200 is characterized by the following properties

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

The S7-200 achieves its full performance potential in distributed automation solutions thanks especially to the integrated ProFi Bus-DP connection. The application area of the SIMATIC S7-200 extends from replacing simple relays and contactors right up to more complex automation tasks.

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

- Baling processes
- Plaster & Cement mixers
- Suction Plants
- Centralized lubricating systems/flange lubricating systems
- Woodworking machinery
- Gate controls
- Hydraulic lifts
- Conveyor systems
- Food & Drink Industry
- Laboratories
- Modem applications via dial-up, leased-line, or radio remote monitoring (SCADA)
- Electrical Installations

4.4.2 Mechanical features include

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

4.4.3 Design features

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

4.4.4 Benefits of the S7-200

4.4.4.1Complete Automation Solution

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

4.4.4.2Value for OEMs

Wherever central controllers or expensive custom electronic control systems are used, the SIMATIC S7-200 offers a significantly more economical alternative. Our off-the-shelf, compact solution, is packed with features, and is accepted around the world as a Micro PLC standard.

4.4.4.3Real-time Speed & Versatility

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

4.4.4.4Integrated Communications

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

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

CHAPTER 5

5. PROGRAMMING OF PLC SYSTEMS

In the previous chapter we were introduced to logic instructions sets for programming PLC systems. The complete sets of basic logic instruction for two common programmable controllers are given below. Note the inclusion in these lists of additional instructions ORB and ANB to allow programming of more complex, multi branch circuits. The use of all these instructions and others is dealt with in this chapter. Some typical instruction sets for Texas instruments and Mitsubishi PLC's are given in table 5.1

Table 5.1 Typical logic instruction sets

Texas Instrument		Mitsubishi A series	
Mnemonic	Action	Mnemonic	Action
STR	Store	LD	Start rung with an open contact
OUT	Output	OUT	Output
AND	Series components	AND	Series elements
OR	Parallel components	OR	Parallel elements
NOT	Inverse action	...I	As for not
		ORB	Or together parallel branches
		ANB	And together series circuit blocks

5.1 Logic instruction sets and graphic programming

In the last chapter we introduced logic instructions as the basic programming language for programmable controllers. Although logic instructions are relatively easy to learn and use, it can be extremely time-consuming to check and relate a large coded program to actual circuit function.

In addition, logic instructions tend to vary between different types of PLC.

In a factory or plant is equipped with a range of different controllers (a common situation), confusion can result over differences in the instruction sets.

RELAY LOGIC SYMBOLS: (MISUBISHI PLC)

Input, normally open contacts

Input, normally closed contacts

Inputs in parallel connection

Output device

Special instruction circuit block

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

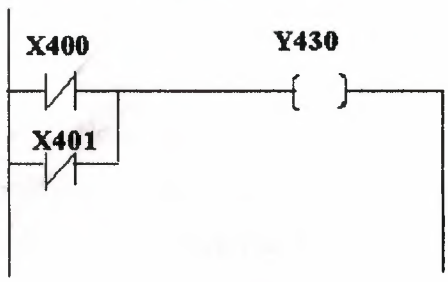
The programming panel translates or compiles these graphing symbols into machine instructions that are stored in the PLC memory, relieving the user of this task.

Different types of graphing programmer are normally used for each family of programmable controller, but they all support similar graphic circuit conventions. Smaller, hand-held panels are common for the small to medium-sized PLC's although the same programming panel often used as a 'field programmer' for these and larger

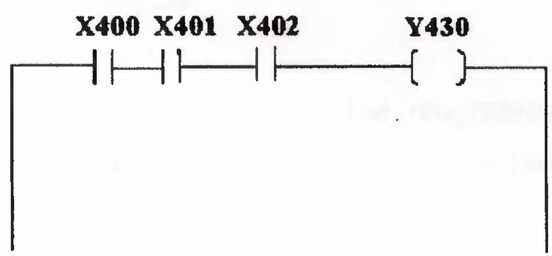
PLC'S in the sane family. However, the majority of graphic programming for larger systems is carried out on terminal sized units. Some of these units are also semi-portable, and may be operated alongside the PLC system under commissioning or test in plant. In addition to screen displays, virtually all graphic-programming stations can drive printers for hard copy of the programs and\or status information, plus program storage via battery-backed RAM or tape/ floppy disk. The facility to load resident programs into EPROM IC'S may be available on more expensive units.

5.1-1 Input/output numbering

It was previously stated that different PLC manufacturers use different numbering systems for input/ output points and other functions within the controller.



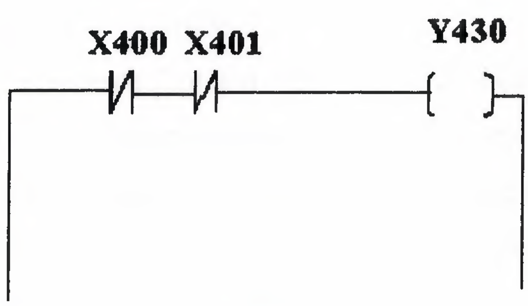
OR gate



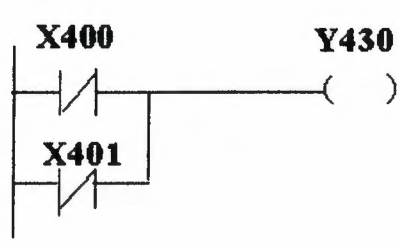
AND gate

5.1-2 Negation – NAND and NOR gates

These logic functions can be produced in ladder form simply by replacing all contacts with their inverse, AND becomes ANI; OR becomes ORI; etc. this changes the function of the circuit.



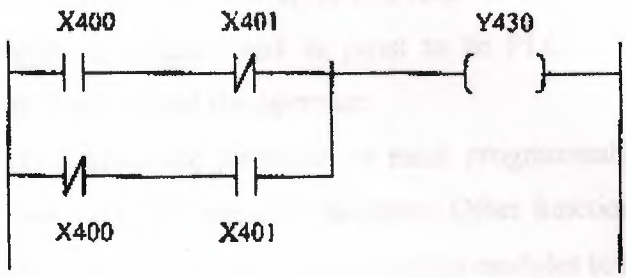
NOR gate



NAND gate

5.1-3 Exclusive – OR gate

This different form the normal OR gates as it gives an output of 1 when either one input or the other is on, but not both. This is comparable to two parallel circuits, each one with make and one break contact in series as shown in exclusive OR gate figure.



EXCLUSIVE – OR gate

Note the use of an ORB instruction in this example. The programmable controller reads the first two instructions, then finds another rung start instruction before an OUT instruction has been executed. The CPU therefore realizes that a parallel form of circuit exists and reads the subsequent instructions until an ORB instruction is found.

5.2 Facilities

5.2-1 Standard PLC functions

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

These internal functions are not physical input or output. They are simulated within the controller.

Each function can be programmed with related contacts, which may be used to control different elements in the program. As with physical inputs and outputs. Certain number ranges are allocated to each block of functions. The number range will depend both on the size of a PLC, and the manufacturer. For example, for Mitsubishi F-40 series, the details are as follows:

Timers T	450 - 457
	550 – 557

Counters C 460 – 467

560 – 567

The information illustrates the use of different number ranges assigned to each supported function. For example, the timer circuits for this programmable controller are addressed from 450 to 457 and 550 to 557, a total of 16 bit timers. It is the specified number that identifies a function and its point to he PLC, not the prefix letter. This prefixes are included only to aid the operator.

The functions listed are provided on most programmable controllers, although the exact format will vary between manufacturers. Other functions may also exist, either as standard or by the selection and fitting of function modules to the PLC rack.

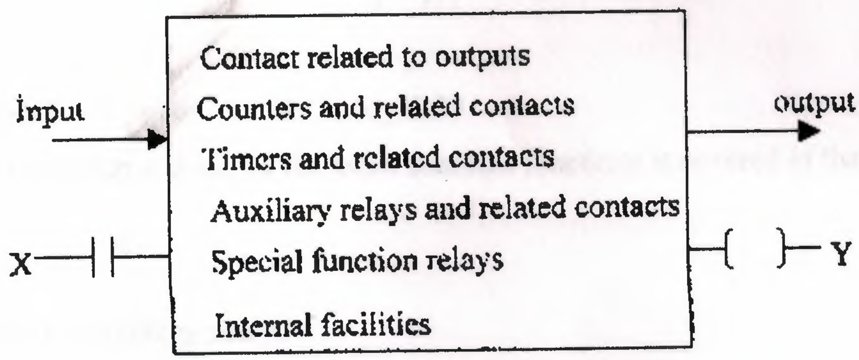


Figure 5.1 Standard PLC function

PC (F40) I/O ASSIGNMENTS

Inputs:	24 points	400 – 407
		410 – 413
		500 – 507
		510 – 513
Outputs:	16 points	430 – 437
		530 – 537
Timers:	16 points	450 – 457
		550 – 557

Counters:	16 points	460 – 467
		560 – 567
Auxiliary control		100 – 107
Relays:	128 points	170 – 177
		200 – 207
		270 – 277
Battery – backed:	64 points	300 – 307
Special function		370 – 377
Auxiliary relays:	5 points	70, 71,72,75,77

The operation and use of the listed standard functions is covered in the following sections.

5.2-2 Markers / auxiliary relays

Often termed control relays or flags, these provide general memory for the programmer, plus associated contacts. They also form the basis for shift-register construction. Normally a group of markers with battery back up is provided allowing process status information to be retained in the event of a power failure. These markers can be used to ensure safe startup/shut down of process plant by including them as necessary in the logic sequence.

Referring, the Mitsubishi F40 has:

128 auxiliary (marker) relays, 64 battery-backed markers.

5.2-3 Ghost contacts

In certain cases it will be necessary to derive an output from the combined logic of several ladder rungs, due to the number of contacts involved. The straightforward way of providing this is to common-up the respective circuit rungs and drives an internal relay or marker (M). This acts in the same manner as a 'physical' relay, in that it can have associated contacts except for the fact that it is simulated by software within the programmable controller, and has no external appearance whatsoever!

In common with other internal functions, auxiliary relays/markers can be programmed with as many associated contacts as desired. These contacts may be used anywhere in a ladder programs as elements in a logic circuit or as control contacts driving output relays or other functions.

5.2-4 Retentive battery – backed relays

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

In figure 5.3 retentive markers M300 is used to retain data in the event of a power failure. Once input X400 is closed to operate the M300 marker, M300 latches via it are associated contact.

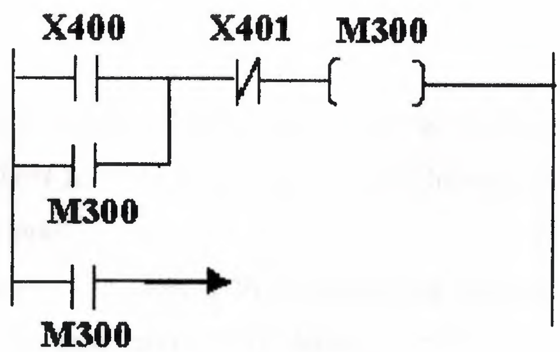


Figure 5.3 retentive markers

So even if X400 is opened due a power failure, the circuit is holds on restart due to M300 retaining the operated status and placing its associated contacts in the operated positions.

Obviously X401 still controls the circuit, and if this input is likely to be energized (opened) by a power-failure situation, than a further stage of protection may be used.

5.2-5 Optional functions on auxiliary relays

From the above text it is apparent that auxiliary relays constitute an important facility in any programmable controller. This is basically due to their ability to control a large numbers of associated contacts and perform as intermediate switching elements in many different types of control circuit.

In addition, many PLC manufacturers have provided additional, programmable functions associated with these auxiliary relays, to further extend their usefulness.

A very common example is a 'pulse' function that allows any designated marker to produce a fixed-duration pulse at its contacts when operated, rather than the normal d.c. level change.

This pulse output is irrespective of the duration of relay operation, thus providing a very useful tool for applications such as program triggering, setting/resetting of timers and counters etc.

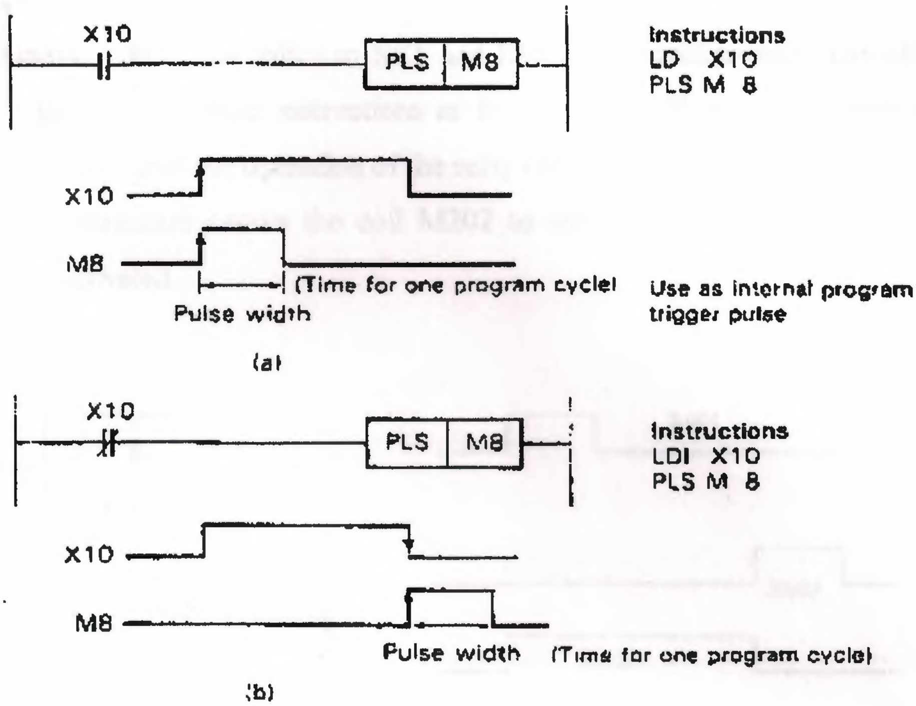
5.2-6 Pulse operation

The programming of these feature varies between controllers, but he general procedure is the same, and very straightforward.

A pulse-PLS instruction is programmed onto an auxiliary relay number. In (figure 5.4)

This configures the designated relay to output a fixed-duration pulse when operated. The examples show how the relay may be used to output a pulse for either a positive or negative going input.

The circuit in (figure 5.5) uses a PLS instruction on auxiliary relay 101 to provide a reset signal for a counter circuit C60. When input 0 is operated, a pulse is sent to relay 101, causing its contacts to pulse and reset counter C60. This is used here because counters and timers often require short duration resetting to allow the restart of the counting or timing.



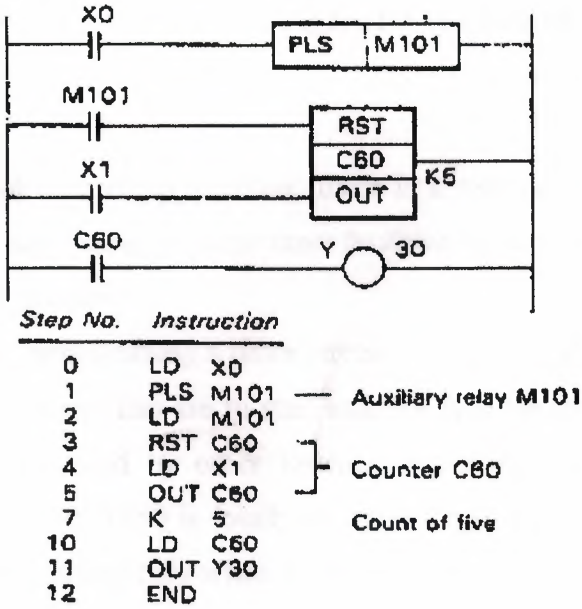


Figure 5.4 Pulse function on auxiliary relays (a) rise detection circuit

Figure 5.5 providing a pulse input to a counter circuit (b) drop detection circuit

5.2-7 Set and reset

As with pulse – PLS, the ability to SET and RESET an auxiliary relay can often be produced by using appropriate instructions as in figure 8.6. These instructions are used to hold (latch) and reset the operation of the relay coils.

The S-set instruction causes the coil M202 to self-hold. This remains until a reset instruction is activated.

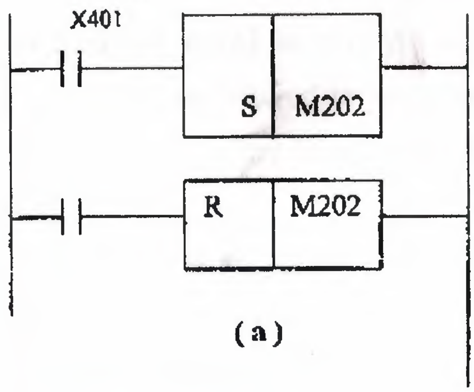


Figure 5.6 (a) set/reset

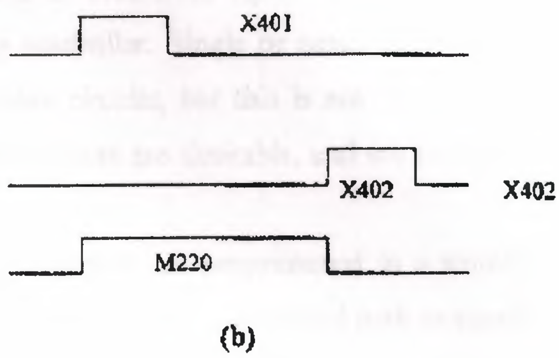


Figure 5.6 (b) time chart

5.2-8 Timers

In a large proportion of control applications, there is a requirement for some aspect of timing control. PC's have some software timer facilities that are very simple to program and use in a variety of situations.

The common method of programming a timer circuit is to specify the interval to be timed, and the conditions or events that are to start and / or stop the timer function. The initiating event may be produced by other internal or external signals to the controller. In this example the timer T450 is totally controlled by a contact related to output Y430. Thus, T450 begins timing only when Y430 is operated. This is caused by input X400 and not X401. Once activated, the timer will 'time-down' from its preset value in this case 3.5 seconds to zero, and then its associated contacts will operate.

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

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

5.2-9 Counters

Whenever the numbers of process actions or events are significance, they must be detected and stored in some manner by the controller. Single or small numbers of events may be remembered by using latched relay circuits, but this is not suitable for larger event counts. Here programmable counter circuits are desirable, and are available on all PLC's.

Provided as an internal function, counter circuits are programmed in a similar manner the timer circuits covered above, but with the addition of a control path to signal event counts to the counter block. Most PLC counters works as subtraction or 'down' counters, as the current value is decremented from the programmed set value.

5.2-10 Registers

From using a single internal or external relay as a memory device to store a single bit of information, other PLC facilities allow the storage of several bits of data at one time.

The device used to store the data is termed a register, and commonly holds 8 or 16 bits of information. Registers can be through of as individual bit-stores- in fact many programmable controllers form the data registers out of groups of auxiliary marker relays in the (figure 5.7)

Registers are very important for handling data that originates from sources than simple, single switches. Instead of binary data in one bit wide fort, information in parallel data form may be read into and out of appropriately sized registers. Thus, data from devices such as thumbwheel switches, analogue to digital converters, can be feed into appropriate PLC registers and used in later operations that will generate other bit or byte wide (8-bit) data to drive switched outputs or digital-to-analogue conversion units.

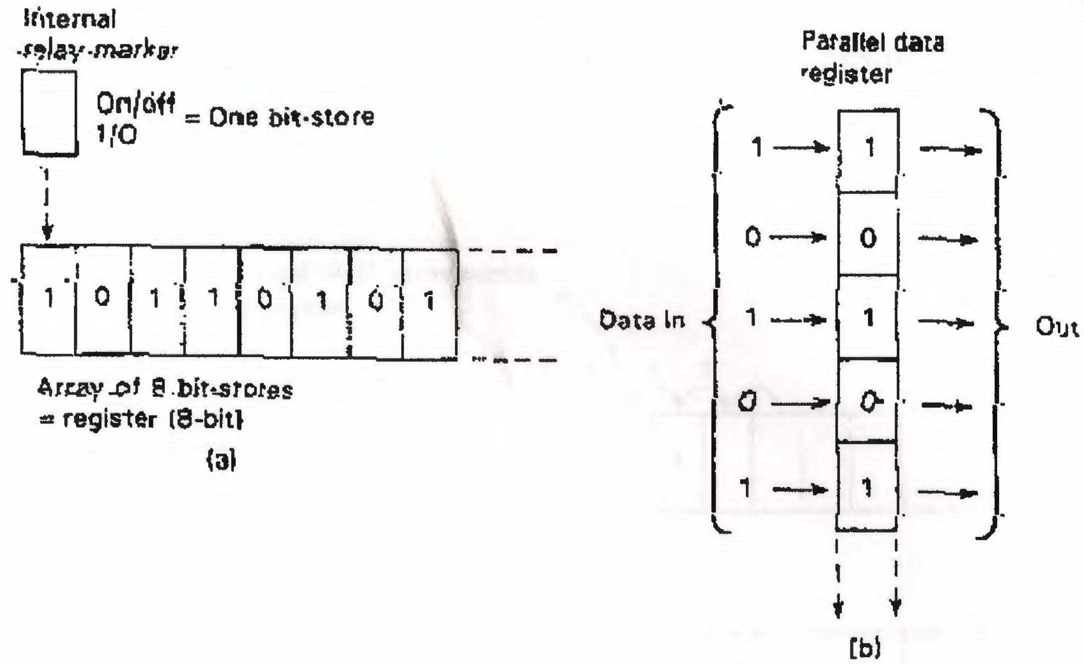


Figure 5.7 Register storage concept (a) array of bit stores; (b) parallel data register

5.2-11 Shift registers

A shift register provides a storage area for a sequence of individual data bits that are offered in series to its input line. The data are moved through the register under control of a shift or clock line as in the figure in the 8.8. The effect of a valid shift pulse is to move all stored digits one bit further in to a register, entering any new data in to the 'freed' initial bit positions. Since a shift register will only be a certain size. For example 8 or 16 bits, then any data in the last bit of the register will be shifted out and lost.

The usefulness of a shift register (SR) lies in the ability to control other circuits or devices via associated SR contacts that are affected by the stream through the register. That is, as with marker relays, when a marker is ON any associated contacts are operated.

In programmable controllers, shift registers are commonly formed groups of the auxiliary relays. This allocation is done automatically by the user programming a 'shift register function', which than reserves the chosen block of relays for that register and prohibits their use for any other function (including use as individual relays).

The example in (figure 5.8) shows a typical circuit for shift registers operation on a Mitsubishi PLC. Here the register is selected by programming in the shift instructions against the auxiliary relay number to be first in the register array M160. This instruction causes a block of relays M160-167 to be reserved for that shift register.

Note that only the first relay had to be specified, the remainder being implied by the instructions. This shows the controlling contacts on the input lines to the register – RESET, OUTPUT and SHIFT.

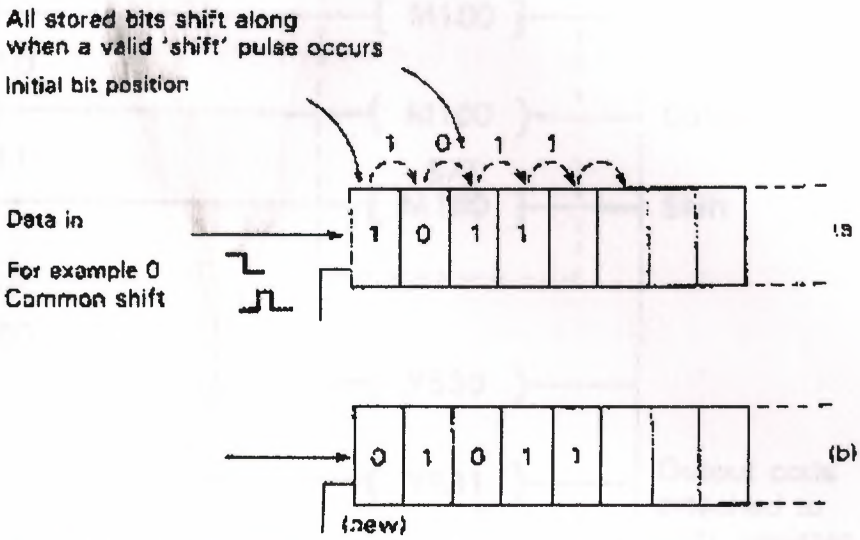
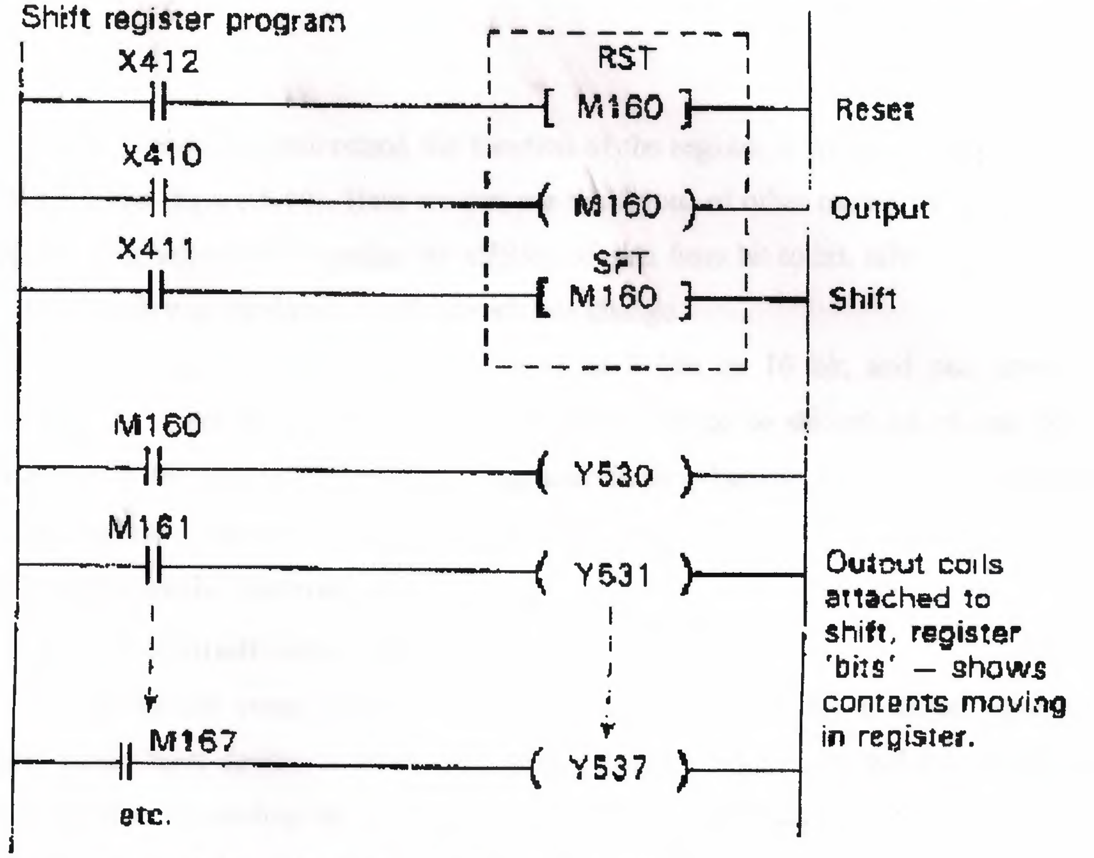


Figure 5.8 Shift register operation: (a) before (b) after 1 shift pulse

The auxiliary relays can be grouped in blocks of 8 to form 8- or 16-bit shift registers. This feature is programmed as shown below using M160–177 internal relays (only M160 is keyed in, the other bits being transparent).



The shift register contacts perform as follows:

RST – a pulse or closure resets SR contents to 0
OUT – logic level (0 or 1) offered to register on this rung.
SFT – pulse moves contents along one bit at a time (eventually contents are lost off the final bit memory).

Figure 5.9 Register contacts

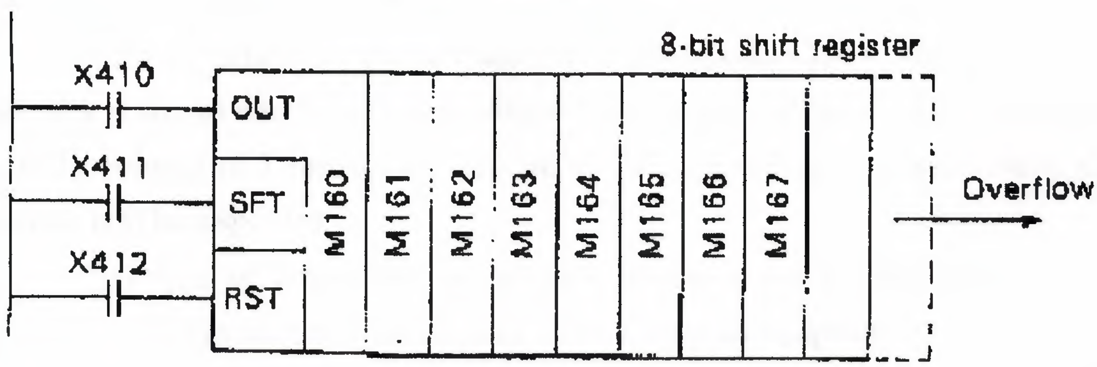


Figure 5.10 equivalent circuit

Note the M – contacts below the SR circuit that are used to drive output coils (M160 – 167 driving Y530 – 537).

It is easier to understand the function of the register if we look at an equivalent circuit in the (figure 5.10). Here we can see the layout of other marker relays following M160. This helps us to visualize the shifting of data from bit to bit, affecting other parts of the circuitry as the data (1 or 0) in each bits change.

Shift registers are commonly found as 8 bits or 16 bit, and can usually be cascaded to create larger shift arrays. This allows data to be shifted out of one register and in to a second register, instead being lost. Battery-backed markers can be selected as the register elements if it is necessary to retain register data through a power failure.

5.3 Arithmetic Instructions

5.3-1 Magnitude comparison

Magnitude comparison instructions arc used to compare a digital value read from some input device or timer, etc., with a second value contained in a destination data register. Depending on the instruction more than, less than, or equal, this will result in a further operation when the condition is met. For example, a temperature probe iii a furnace returns an analog voltage representing the current internal temperature. This is convened in to a digital value by an analog to digital converter module on the PC, where it is read from input points by a data transfer instruction and stored in data register DIO. The process requires that if the temperature is less than 200 C, then the process must halt due to insufficient temperature.

If the temperature is greater than 200 C and less than 250 C, then the process operates at normal rate. If the temperature is between 250 and 280 C, than baking time is to be reduced to 3 minutes 25 seconds, and once temperature exceeds 280 C the process is to be suspended.

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

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

5.3-2 Addition and subtraction instructions

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

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

CHAPTER 6

6- LADDER PROGRAM DEVELOPMENT

6.1. Software Design

When ladder programs are being developed to control simple actions or equipment, the amount of planning and actual design work for these short programs is minimal, mainly because there is no requirement to link with other actions or sections within the program. The ladder networks involved are small enough to be easily understood in terms of circuit representation and operation. In practice, of course, circuits are not limited to AND or OR gates, often involving mixed logic functions together with the many other programmable functions provided by modern programmable controllers.

When larger and more complex control operations have to be performed, it quickly becomes apparent that an informal and unstructured approach to software design will only result in programs that are difficult to understand, modify, troubleshoot and document. The originator of such software may posses an understanding of its operation, but this knowledge is unlikely to remain after even a short period of time away from that system.

In terms of design methodology, than, ladder programming is no different from conventional computer programming. Thus, considerable attention must be given to:

- Task definition / specification
- Software design techniques
- Documentation
- Program testing

6.1-1 System functions

Most industrial control systems may be considered as a set of functional areas or blocks, in order to aid the understanding of how the total system operates.

For example, each machine in a plant unit can be treated as a separate sub-process. Each machine process is then broken down in to blocks that may be described in terms of basic sequences and operations in the (figure 6.1) illustrate this approach.

A functional block could for example, consist of all actions required to control a certain machine in the process.

Block 1

Block 2

Block3

The division of programming tasks in to functional blocks is an important part design.

In logic programming, there are two different types of network that may be used to implement the function of a given block:

- Interlocks or combinational logic, where the output is purely dependent on the combination of the inputs at any instant in time.
- Sequential networks where the output is dependent not only on the actual inputs but on the sequence of the previous inputs and outputs.

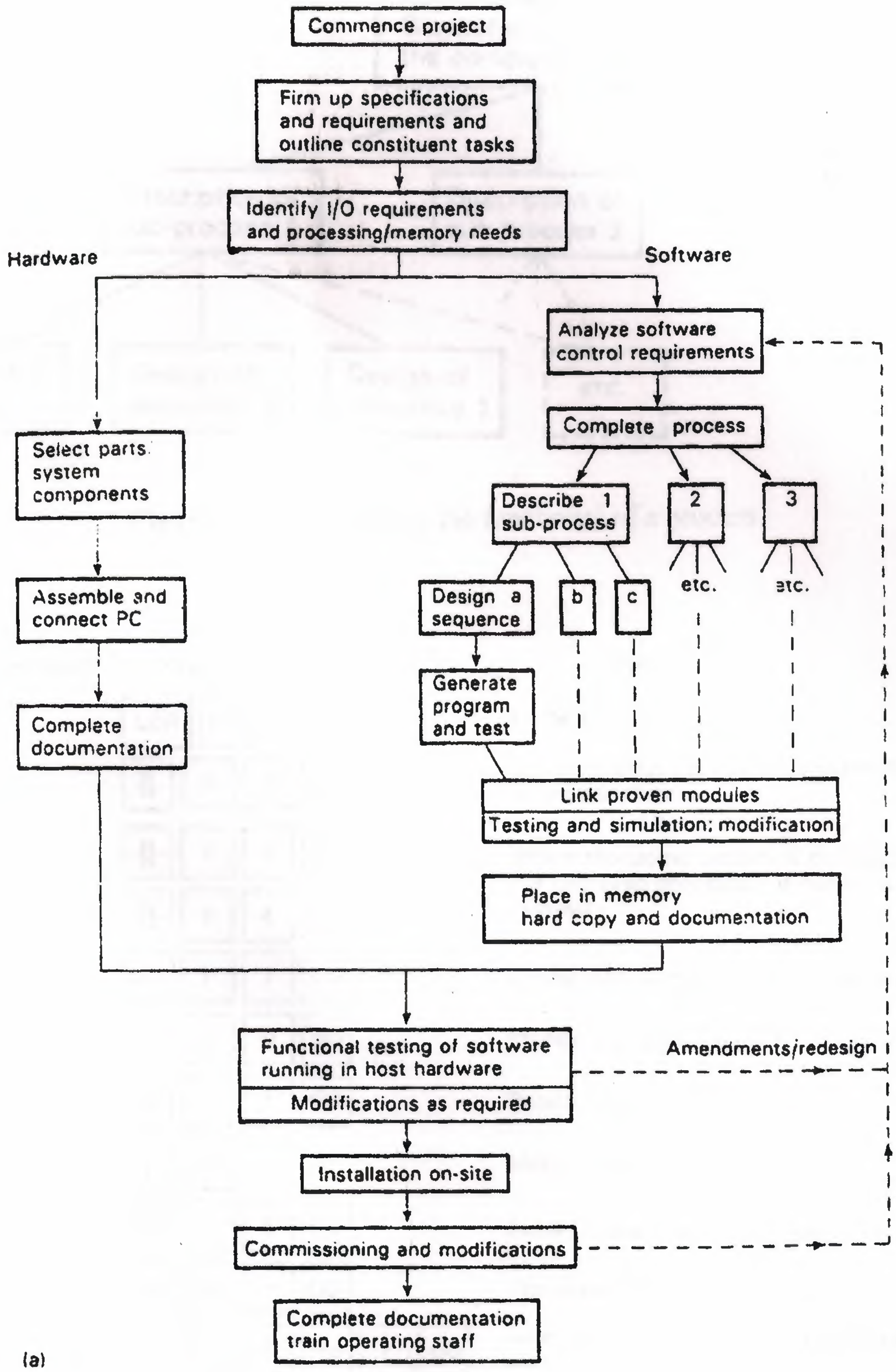
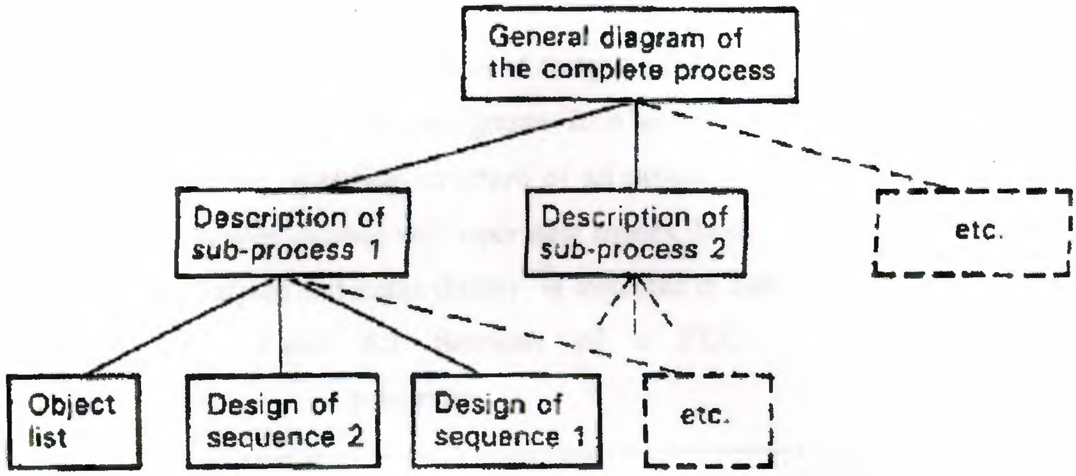


Figure 6.1 (a) PC system design procedure



(b)

Figure 6.1(b) describing the functional of a process

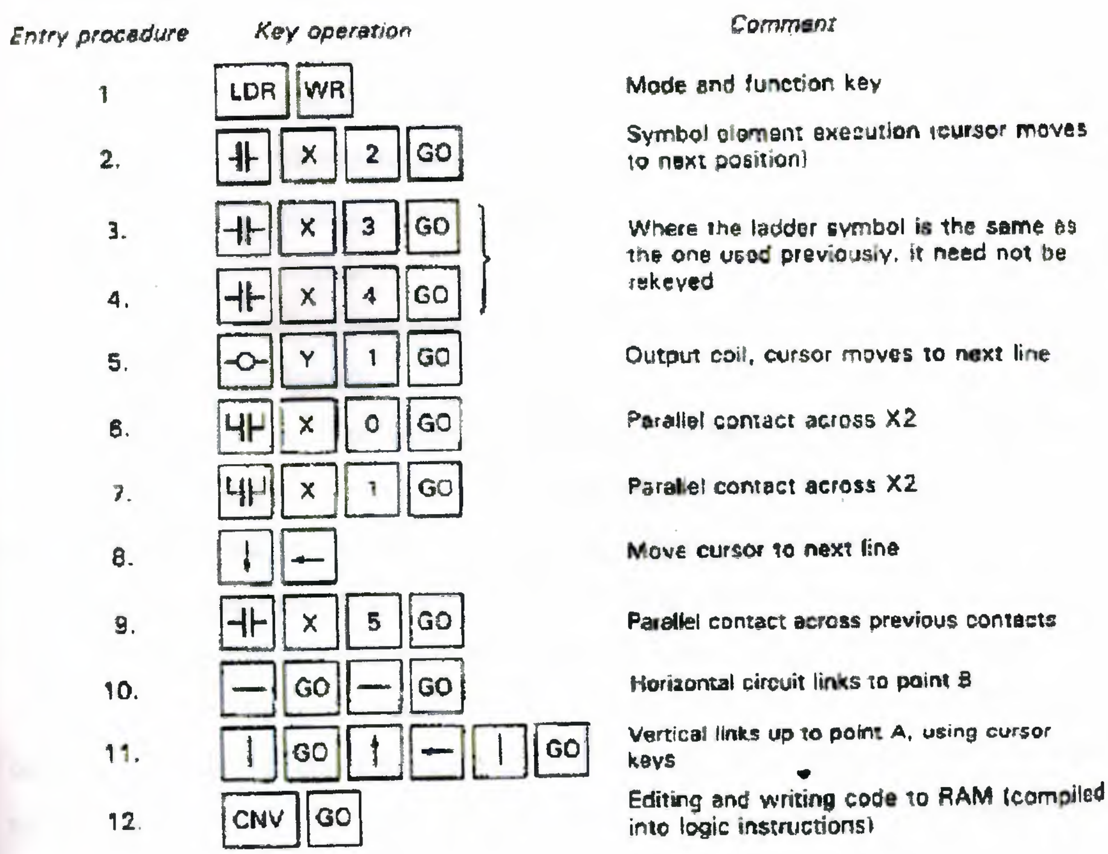


Figure 6.2 Graphing programming

6.2. Program Structure

At this stage in the investigation of design techniques, it is appropriate to discuss the layout and structure of PLC programs. It is sound practice to base any program layout on the general operating structure of all process-control Systems. This means having definite sections dealing with operating modes, basic functions, process chain or sequence, signal outputs and status display, as indicated in Table 6.1.

Table 6.1 Sections of a PLC program

Start
Operating modes and basic functions
Starting (basic) position
Enabling / reset conditions
Process operation / sequence logic
Signal outputs
Status / indicator output
Finish

a- Operating modes

Basic position: The controlled equipment is likely to have a basic or normal position, for example when all actuators are off and all limit switches are open. All these elements can be combined logically to signify and initialize a basic position, which may be programmed as a step in a sequential process.

Enabling / reset conditions: Most industrial processes have manual start mid stop controls that may be incorporated in to the PLC program structure at this point. These would be included as enabling and reset contacts, having overall control of the PLC in terms of run or stop. There may also be a manual switch to enable the system outputs, which would allow the program to run without driving the physical outputs, connected to the PLC a test function.

b- Process operation/sequence

This is the main topic of this chapter, involving the design and programming of combinational and sequential networks as necessary. The resultant outputs do not normally drive actuators directly, but instead are used to operate intermediate marker relays.

C- Signal output

Output signals to process actuators are formed by interlocking the resulting operation sequence outputs (markers) with any enabling conditions that exist in (a) above.

d- Status / indicator outputs

Process status is often displayed using indicator lambs or alarms, etc. Such elements are programmed in this section of the software.

By adopting this systematic approach to program structure, we can create reliable, easily understood software, which will allow rapid fault location and result in short process down times. The programs that are developed in this chapter deal mainly with the topic of process operation, but will be structured in this manner where possible.

6.3. Further Sequential Control Techniques

In many practical applications, a control system has to deal with a process sequence that requires the concurrent operation and control of more than one step Also, steps in a sequence may require a time delay or event count as entry criteria for a succeeding step. To describe the different types of parallel operation, we use the conventions

In figure, actions B OR C are taken, depending on the result of test A2. Either action will allow entry to action D. In the figure shows the format for a process where two actions A AND B are initialized once test A is true; also both tests B AND C must be true before progression to action D.

The equivalent function chart descriptions are illustrated. The number of parallel activities may be extended via the branching and converging rails. The chart in figure shows the tests that allow entry to steps B OR C, and also the individual tests or conditions that will allow resetting of the chosen step (test n and run). Notice the OR signs at each branch rail.

In figure the AND of steps is signified by the double connecting rails after test A and before test n. This means that all parallel steps (in this case B and C) are set once state A is active and test A is fulfilled.

6.4. Limitation of Ladder Programming

1- Ladder programs are ideal for combinational/interlock tasks and simple sequential tasks. However, the lack of comment facilities on most small programmers makes interpretation of any program extremely difficult.

2- when applied to complex sequential tasks, ladder programs become cumber some, difficult both to design and debug. This is mainly due to having to provide entry, hold and reset elements in every stage to ensure no sequence errors occur.

Several manufacturers are adopting a function block style of programming that removes most of this complexity. This employs basic programming symbols that are closely related to the function chart symbols used for program design purposes, as used earlier in this chapter.

6.4.1 Advanced graphic programming languages

The facility for programming using functional blocks is currently available on a few larger programmable controllers, such as those from Siemens and Telemechanique. This approach uses graphic blocks to represent sections of circuitry related to a particular task or part of a process. Each function block is user programmed to contain a section of ladder circuitry required to carry out that function. The sequential operation of the control system is obtained by progression from one block to next, where a step is entered only if it is entry conditions are fulfilled, in which case it becomes active and the previous step becomes inactive. Thus, there is no need to reset the previous step an important advantage over conventional ladder programming.

To examine or program the contents of each block the user would zoom in on the block in question. These windows in on the contents as shown, which are displayed on the programming panel. The necessary details are then entered in normal ladder format.

Provision for displaying simultaneous sequences is a further important feature of these programming methods, displaying the multi tasking ability of PLC's in an easy to follow manner.

6.4.2 Workstations

The traditional tool for programming PLC's is the small, hand held panel, which can provide only limited monitoring and editing facilities. Most manufacturers are now using personal computers as workstations on larger programmable controllers, in order to fully exploit these features, and those of graphic function blocks.

CHAPTER 7

7. CHOOSING INSTALLATION AND COMMISSIONING OF PLC SYSTEM

7.1. Feasibility Study

Under certain circumstances an initial feasibility study may be suggested or warranted prior to any decision on what solution will be adopted for a particular task. The feasibility study may be carried out either by in house experts or by external consultants. Often an independent specialist is preferred, having few or no ties to specific vendor equipment.

The scope of such a study can vary enormously, from simply stating the feasibility of the proposal, through to a comprehensive case analysis with complete equipment recommendations. Typically, though, a feasibility study of this nature encompasses several specific areas of investigation:

(a) **economic feasibility,** consisting of the evaluation of possible installation and development costs weighed against the ultimate income or benefits resulting from a developed system;

(b) **technical feasibility,** where the target process and equipment are studied in terms of function, performance and constraints that may relate to achieving an acceptable system;

(c) **Alternatives,** with an investigation and evaluation of alternative approaches to the development of the acceptable system.

Area (a), economic feasibility and worth, can only be addressed fully once the result of areas (b) and (c) are available, with estimated coatings, and direct indirect benefits being considered. Area (b) is detailed in the following sections, with background information for area (a) usually being compiled through liaison with company personnel. The achievement of a complete technical proposal requires us to know what the present and future company needs are in terms of plant automation and desired information systems.

Once the control function has been accurately defined, a suitable programmable control system has to be chosen from the wide range available. Following the identification of a suitable PLC, work can begin on aspects of electrical hardware design and software design.

PLC system is reached, there are several topics to be considered:

- Necessary input/output capacity; types of I/ O required;
- Size of memory required;
- speed and power required of the CPU and instruction set.

All this topics are to a large extent interdependent, with the memory size being directly tied to the amount of I/ O as well as program size. As the I/ O memory size rises, this takes longer to process and requires a more powerful, faster central processor if scan times arc remain acceptable.

7.2.3 I/0 requirements

The I/ O sections of a PLC system must be able to contain sufficient modules to connect all signal and control lines for the process. These modules must conform to the basic system specifications as regards voltage levels, loading, etc.,

- The number and type of I/ O points required per 'nodule;
- Isolation required between the controller and the target process;
- The need for high speed I/ O, or remote I/ O, or any other special facility;
- Future needs of the plant in terms of both expansions potential and installed spare I / O points,
- Power supply requirements of I/ O points are an on board PSU needed to drive any transducer or actuators?

In certain cases there may be a need for signal conditioning modules to be included in the system, with obvious space demands on the main or remote racks. When the system is to he installed over a wide area, the use of a remote or decentralized form of I/ O working can give significant economies in cabling the sensors and actuators to the PLC.

7.2.4 Memory and programming requirements

Depending on the type of programmable control let being considered, the system memory may be implemented on the same card as the CPU, or alternatively on dedicated cards. This ladder method is the more adaptable, allowing memory size to be increased as necessary- up to the system maximum, without a reciprocal change in CPU card.

As stated in the previous section, memory size is normally related to the amount of I/ O points required in the system. The other factor that affects the amount of memory required is of course the control program that is to be installed. The exact size of any program cannot be defined until of the software has been designed, encoded, installed and tested. However, it is possible to accurately estimate this size based on average program complexity A control program with complex, lengthy interlocking or sequencing routines obviously requires more memory than one for a simple process. Program size is also related to the number of I/ O points, since it must include instructions for reading from or writing to each point. Special functions are required for the control task may also require memory space in the unit PLC memory map to allow data transfer between cards. Finally additional space should be provided to allow for changes in the program, and for future expansion of the system.

There is often a choice of available memory type RAM or EPROM. The RAM form is the most common, allowing straightforward and rapid program alterations both before and after the system is installed. RAM contents are made semi-permanent by the provision of battery backing on their power supply. RAM must always be used for I/ O and data functions, as these involve dynamic data.

EPROM memory can be employed for program storage only, and requires the use of a special EPROM eraser / programmer to alter the stored code. The use of EPROMS is ideal where identical programmable controllers running the same control several machines.

However, until a program has been a hilly developed and tested, RAM storage should be used.

As mentioned in earlier chapters, microcomputers arc commonly used as program development stations. The large amounts of RAM and disk storage space provided in these machines allow the development and storage of many PLC programs, including related text and documentation. Programs can be transferred between the microcomputer and the target PLC for testing and alteration. EPROM programming can also often be carried out via the microcomputer.

7.2.5 Instruction set I CPU

Whatever else is left undefined; any system to be considered must provide an instruction set that is adequate for the task. Regardless of size, all PLC's can handle logic control, sequencing, etc. Where differences start to emerge are in the areas of data handling, special functions and communications. Larger programmable controllers tend to have more powerful instructions than smaller ones in these areas, but careful scrutiny of small medium machines can often reveal the capability to perform specific functions at surprisingly good levels of performance.

In modular programmable controllers there may be a choice of CPU card, offering different levels of performance in terms of speed and functionality. As the number of I/ O and function cards increases, the demands on the CPU also increase, since there ate greater numbers of signals to process each cycle. This may require the use of a faster CPU card if scan time is not to suffer.

Following the selection of the precise units that will make up the programmable controller for a particular application, the software and hardware design functions can be carried out independently.

7.3. Installation

The hardware installation consists of building up to necessary racks and cubicles, then installing and connecting the cabling.

The cabinet that contains the programmable controller and associated sub-racks must be adequate for the intended environment, as regards security safety band protection from the elements:

-security in the form of a. robust, lockable cabinet; safety, by providing automatic cut off facilities alarms if the cabinet door is opened; protection from humid or corrosive atmospheres by installation of airtight seals on the cubicle. Further electrostatic shielding by earthing the cubicle body.

For maintenance purposes, there must be easy access to the PLC racks for card inspection, changing etc. Main on/ off and status indicators can be built in to the cabinet doors, and glass or Perspex windows fined to allow visual checking of card status or relay/ contactor operation.

7.4. Testing and Commissioning

Once the installation work is completed, the next step is to consider the testing and commissioning of the PLC system.

Commissioning comprises two basic stages:

1- Checking the cable connections between the PLC and the plant to be controlled.

2- Installing the completed control software and testing its operation on the target process-

The system interconnections must be thoroughly checked out to ensure all input' output devices are wired to the correct I/ O points. In a conventional control system buzzing out the connections with suitable continuity test instruments would do this.

With a programmable, however, the programming panel may be used to monitor the status of inputs points directly this is long before the control software is installed which will only be done after all hardware testing is satisfactorily completed. Before any hardware testing is started, a thorough test of all mains voltages, earthing, etc. must be carried out.

With the programmer attached to the PLC, input points are monitored as the related transducer is operated, checking that the correct signal is received by the PLC. The same technique is used to test the various function cards installed in the system. For example, altering can check analog inputs the analog signal and observing a corresponding change in the data stored in the memory table.

In turn, the output devices can be forced by instructions from the programming panel. Checking their connection and operation. The commissioning team must ensure that any operation or disoperation of plant actuators will not result in damage to plant or personnel

Testing of some PLC functions at this stage is not always practical, such as for PID loops and certain communications channel. These require a significant amount of configuring by software before they can be operated, and are preferably tested once the control software has been installed.

Some programmable controllers contain in built diagnostic routines that can be used to check out the installed cards, giving error codes on a VDU or integral display screen. These diagnostic are run by commands from the programming panel, or from within a control program once the system is fully operational.

7.4.1 Software testing and simulation

The preceding sections have outlined the various stages in hardware design and implementation. Over the same period of time, the software to control the target process is developed, in parallel, for the chosen PLC system. These program modules should be tested and proved individually wherever possible, before being linked together to make up' the complete applications program. It is highly desirable that any faults or error be removed before the program is installed in the host controller.

The time required to rectify faults can be more than doubled once the software is running in the host PLC.

Virtually all programmable controllers, irrespective of size, contain elementary software checking facilities. Typically these can scan through an installed program to check for incorrect labels. Double output coils etc, Listings of all I/ O points used, counter/ timer settings and other information is also provided. The resulting information is available on the programmer screen or as a printout. However, this form of testing is only of limited value, since there is no facility to check the operation of the resident program.

In terms of time and cost economies, an ideal method for testing program modules is to reproduce the control cycle by simulation; since this activity can be carried out in the design workshop without having the actually connect up to the physical process. Simulation of tile process is done in a number of ways, depending on the size of process involved.

When the system is relatively small with only a handful of I/ O channels it is often possible to adequately simulate the process by using. Sets of switches connected up to the PLC as inputs, with outputs represented by connecting arrays of small lambs or relays in the figure 7.1 . This allows inputs to be offered to a test bed controller containing software under test, checking the action of the control program by noting the operation and sequence of the output lambs or relays. By operating the input switches in specific sequences, it is possible to test sequence routines within a program. Where fast response times are involved, the tester should use the programming panel to force larger time intervals into the timers concerned, allowing that part of the circuit to be tested by the manual switch method.

Most I/ O modules have LED indicators that show tile status of the channels. These can be used instead of additional test actuators where digital outputs arc concerned Analog inputs can be simulated in part by using potential dividers suitably connected to the input channel, and corresponding analog outputs connected either to variable devices such as small motors or to a moving coil meter configured to measure voltage or current. Standard sets of input switches and output actuators are normally available from PLC manufacturers.

When the system is larger with input/output channels and longer, more complex programs, the simple form of simulation described above becomes inadequate. Many larger PLC Systems are fitted an integral simulation unit that reads and writes information directly into the I/ O memory, removing the need to connect external switches, etc. The simulator is controlled from an associated terminal, which can force changes in input status and record all changes in output status as the program runs, for later scrutiny by the test team.

The program monitoring facility provided with most programming terminals should be used in virtually all these proceedings, since it allows the dynamic checking of all elements in the program including preset and remaining values as the program cycles.

It is important to realize that the display on the programmer does not up date as rapidly as the control program is executing, due to the delays in transmitting the data across to the terminal.

Contacts and other elements that are operated for only a few scans are unlikely to affect the display, but since a human observer could not detect this fast a change this is not a significant disadvantage. To display all changes, the PLC should be run in single step mode.

The monitor display shows a select portion of the ladder program, using standard symbols to depict contacts, output and present functions. All elements within the display are dynamically monitored.

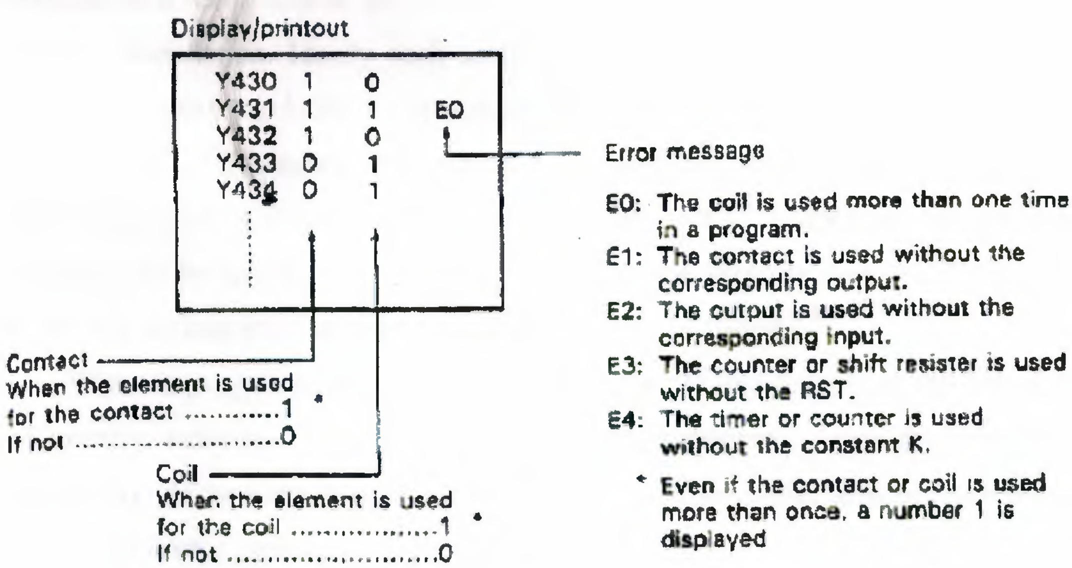


Figure 7.1 PLC printout of I/O static diagnostics information

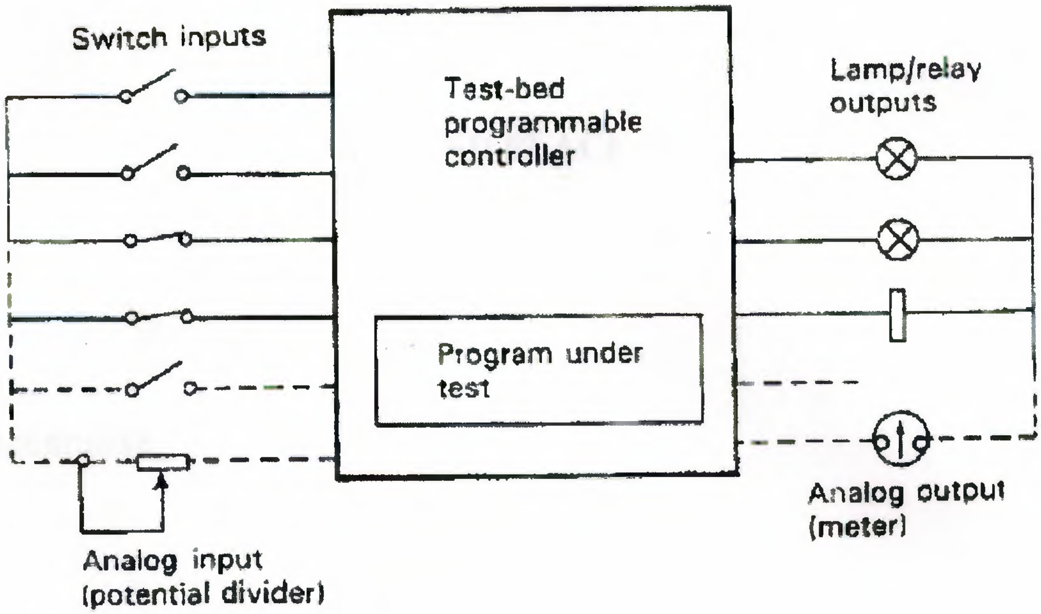


Figure7. 2 Process simulation using switches and lambs

7.4.2 Installing and running the user control program

Once the control software has been proved as far as possible by the above, methods on a test machine, the next step is to try out the program on the tested PLC hardware installation. Ideally each section of code should be downloaded and tested individually, allowing faults to be quickly localized if the plant misoperates during the program test. If this subdivided testing is not possible, another method is to include JUMP commands in the complete program to miss out all instructions except those in the section to be tested. As each section is proved, the program is amended to place the JUMP instructions so as to select the next section to be tested.

Where a programmable controller supports single step operation, this can be used the examine individual program steps for correct sequencing. Again the programming terminal should be utilized to monitor I / O status or any other area of interest during these tests. With continuous printouts if this is possible.

Chapter 8

PICK AND PLACE UNIT

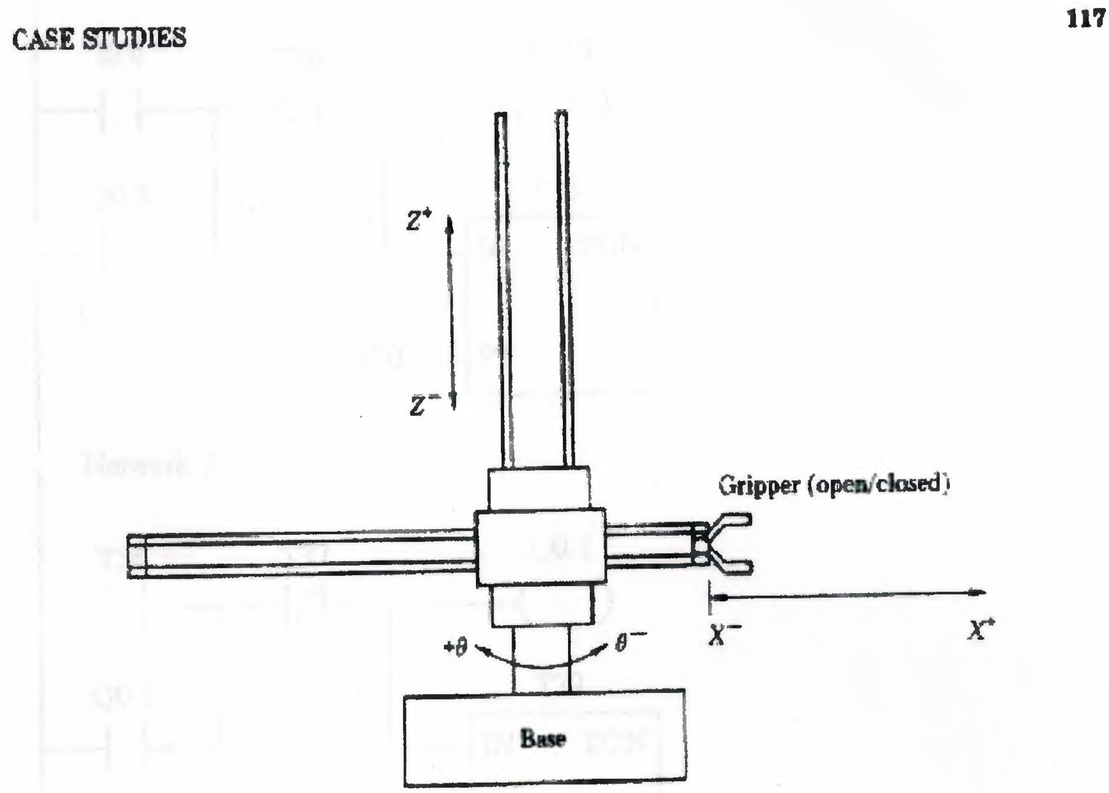


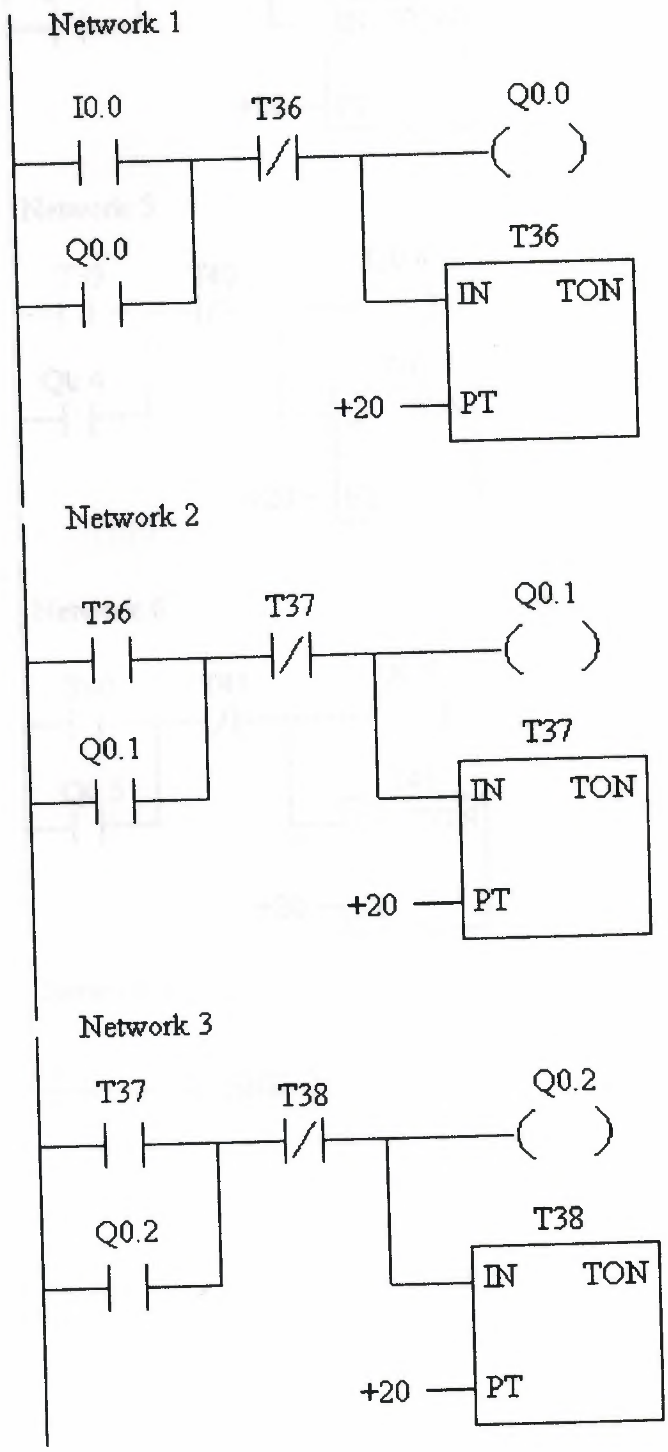
Figure 8.1 pick and place unit

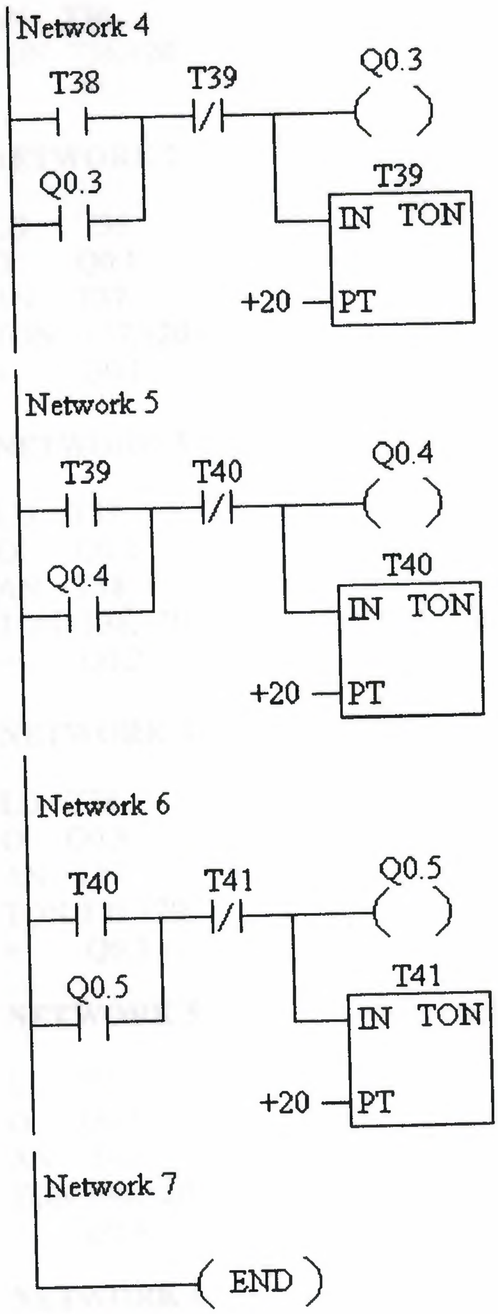
Being actuated by pneumatic pıstons.İn addıtıon, apneumatıcally operated grıpper maybe open or shut.as the movements are on/off actıons they are often descrıbed as bang-bang .

Initially the gripper is open and posıtıoned at X^- AND θ^- position.the plc is to (a) move the gripper to the X+ position, (b) close the gripper so that it takes hold of a components, (c) rotate the gripper through 180° to the θ+ position ,(d) release the component and (e) rotate the gripper back to the θ^- position so that the pick and place operation maybe repeated .

All movements are controlled by actuating appropriate solenoids.Interfacing involves connecting eight solenoids to eight output relay .

It is assumed that the relays are connected so that when energized they cause the following motions .




Network 4
T38
T39
Q0.3
Q0.3
T39
IN TON
+20
PT
Network 5
T39
T40
Q0.4
Q0.4
T40
IN TON
+20
PT
Network 6
T40
T41
Q0.5
Q0.5
T41
IN TON
+20
PT
Network 7
END

NETWORK 1

```
LD    I0.0
O     Q0.0
AN    T36
TON   T36,+20
=     Q0.0
```

NETWORK 2

```
LD    T36
O     Q0.1
AN    T37
TON   T37,+20
=     Q0.1
```

NETWORK 3

```
LD    T37
O     Q0.2
AN    T38
TON   T38,+20
=     Q0.2
```

NETWORK 4

```
LD    T38
O     Q0.3
AN    T39
TON   T39,+20
=     Q0.3
```

NETWORK 5

```
LD    T39
O     Q0.4
AN    T40
TON   T40,+20
=     Q0.4
```

NETWORK 6

```
LD    T40
O     Q0.5
AN    T41
TON   T41,+20
=     Q0.5
```

NETWORK 7

```
END
```

CONCLUSION

When developing this project, I see that PLC makes life easier.

Because there are too many advantages using PLC such as controlling several machines, easy to create and improve or check the program with single PC. In a small time interval we can adjust or change the operation. But there is one disadvantages in PLC , that's it minify the mans .