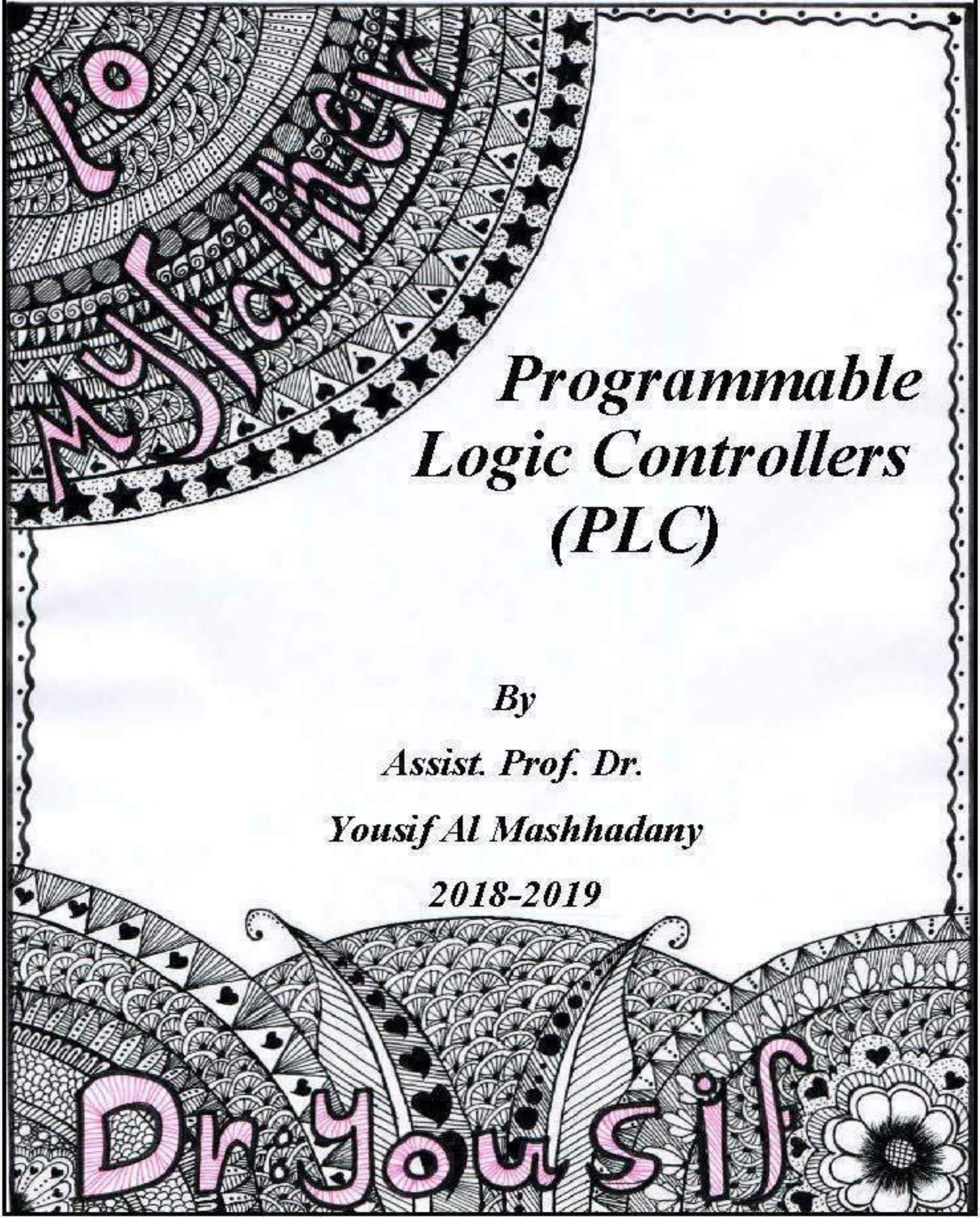


Programmable Logic Controllers (PLC)

By

Assist. Prof. Dr.

Yousif Al Mashhadany

2018-2019

No	*Subject*	*Week (3 H/W)*
1.	**Introduction to PLC** ➢ **History of PLC** ➢ **Programmable Logic Devices (PLDs)** **Introduction ; Microprocessor ; Microcomputer** ➢ **Hardware of PLC** ➢ **Internal architecture of PLC** **The CPU ; The buses ; Memory ; Input/output unit** **Sourcing and sinking** ➢ **PLC systems** **Programming PLCs**	**3**
2.	**Input-output devices** ➢ **Input devices** **Mechanical switches ; Proximity switches** **Photoelectric sensors and switches** **Encoders ; Temperature sensors** **Position/displacement sensors** **Strain gauges ; Pressure sensors ; Smart sensors** ➢ **Output devices** **Relay ; Directional control valves ; Motors** **Stepper motors** ➢ **Examples of applications** **Conveyor belt ; Traffic Light system** **Robot control system ; Liquid level monitoring**	**3**

3.	**I/O processing** ➢ **Input/output units** **Input units ; Output units** ➢ **Signal conditioning** ➢ **Remote connections** **Serial and parallel communications ; Serial standards ;** **Parallel standards ; Protocols ; ASCII codes** ➢ **Networks** **Distributed systems ; Network standards ;** **Examples of commercial systems** ➢ **Processing inputs** ➢ **I/O addresses**	**4**
4.	**Ladder and functional block programming** ➢ **Ladder diagrams** **PLC ladder programming ;** ➢ **Logic functions** **AND ; OR ; NOT ; NAND ; NOR ;** **Exclusive OR (XOR) ;** ➢ **Latching** ➢ **Multiple outputs** ➢ **Entering programs** **Ladder symbols** ➢ **Function blocks** **Logic gates ; Boolean algebra** ➢ **Program examples**	**4**

5.	**PLC Application Examples** ➢ **Temperature control** ➢ **Valve sequencing** ➢ **Conveyor belt control** ➢ **Control of a process**	**2**

References:

➢ *Theoretical:*

1. "Programmable Logic Controllers", Fourth Edition , By: W. Bolton, 2006.
2. "Programmable Logic Controllers, Basic Level TP01 Textbook", First Edition, 2002, By. R Bliesener, ..., A. Winter.

➢ *Practical*

1) LG Programmable Logic Controller, GLOFA GM7 Series. (Installation Manual).
2) Ladder Diagram, with GMWIN_V4.18_ENG(2014-12-18)_REL.
3) PLC Trainer For Electrical Eng. Dept./ UOA. "Design and Implement of a Programmable logic controller (PLC) for Classical Control Laboratory", By Yousif Al Mashhadany, Intelligent Control and Automation journal, (http://www.scirp.org/journal/ica), Vol. 3 No. 1, ISSN Online: 2153-0661, Feb. 2012

2018-2019

Section One:

Introduction to Programmable logic controllers

Introduction to PLC:

- **History of PLC**
- **Programmable Logic Devices (PLDs)**

  Introduction ; Microprocessor ; Microcomputer ; PLC
- **Hardware of PLC**
- **Internal architecture of PLC**

  The CPU ; The buses ; Memory ; Input/output unit

  Sourcing and sinking
- **PLC systems**

  Programming PLCs

The History of PLCs

In the 1960's Programmable Logic Controllers were first developed to replace relays and relay control systems. Relays, while very useful in some applications, also have some problems. The main problem is the fact that they are mechanical. This means that they wear down and have to be replaced every so often. Also, relays take up quite a bit of space. These, along with other considerations, led to the development of PLCs.

More improvements to PLCs occurred in the 70's. In 1973 the ability to communicate between PLCs was added. This also made it possible to have the controlling circuit quite a ways away from the machine it was controlling. However, at this time the lack of standardization in PLCs created other problems. This was improved in the 1980's. The size of PLCs was also reduced then, thus using space even more efficiently.

The 90's increased the collection of ways in which a PLC could be programmed (block diagrams, instruction list, C, etc.). They also saw PLCs being replaced by PC's in some cases. However, PLCs are still very much in use in all sorts of industries, and it's likely that they will remain there for quite some time.

Programmable Logic Devices (PLDs)

1. Introduction

Programmable Logic Devices (PLD) are programmable systems and are generally used in manufacturing automation to perform different control functions, according to the programs written in its memory, using low level languages of commands. There are following three types of PLDs are being employed in mechatronics systems.

a) Microprocessor

It is a digital integrated circuit which carries out necessary digital functions to process the information obtained from measurement system.

b) Microcomputer

It uses microprocessor as its central processing unit and contains all functions of a computer.

c) Programmable Logic Controller (PLC)

It is used to control the operations of electro-mechanical devices especially in tough and hazardous industrial environments.

A typical programmable machine has basic three components as shown in Figure 1:

1) Processor, which processes the information collected from measurement system and takes logical decisions based on the information. Then it sends this information to actuators or output devices.

2) Memory, it stores

   i. the input data collected from sensors

   ii. the programs to process the information and to take necessary decisions or actions. Program is a set of instructions written for the processor to perform a task. A group of programs is called software.

   iii. Input/output devices: these are used to communicate with the outside world/operator.

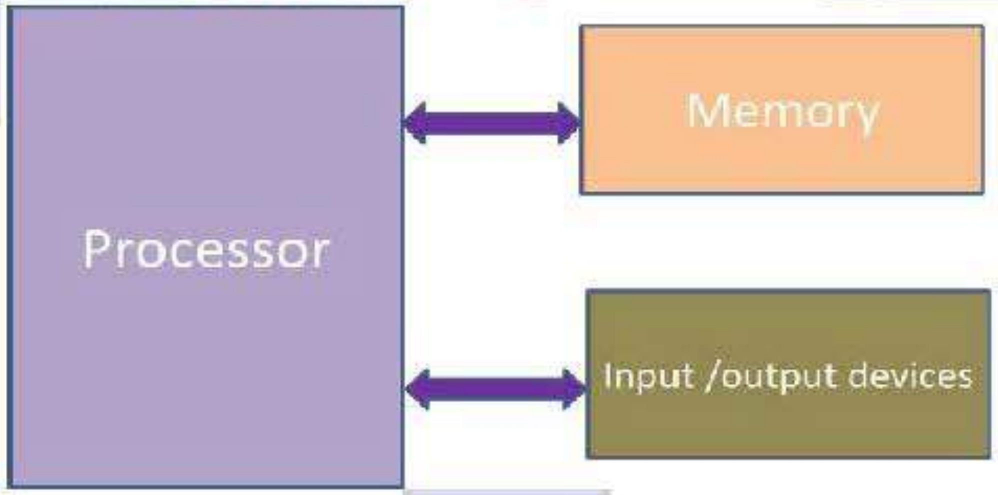


Figure 1. Components of a programmable logic

2. Microprocessor

It is a multi-purpose, programmable device that reads binary instructions from a storage device called memory, processes the data according to the instructions, and then provides results as output. In common practice it is also known as CPU (central processing unit). CPU can be referred as complete computational engine on a single chip. First Microcontroller, Intel 4004 was launched in 1971. It was able to process just 4 bits. It started a new era in electronics engineering. Microprocessor chip was one of the important inventions of the 20th century. Table .1 shows the history of micro-processors.

Applications of microprocessors are classified primarily in two categories:

- ❖ Reprogrammable Systems : Micro computers
- ❖ Embedded Systems : photocopying machine, Digital camera

Microprocessor works or operates in binary digits i.e. 0 and 1, bits. These bits are nothing but electrical voltages in the machine, generally 0 - low voltage level, and 1 - high voltage level. A group of bits form a 'word'. In general, the word length is about 8 bits. This is called as a 'byte'. A word with a length of 4 bits is called as a 'Nibble'

Microprocessor processes the 'commands in binary form' to accomplish a task. These are called as 'instructions'. Instructions are generally entered through input devices and can be stored in a storage device called memory.

Figure .2 and 3 show the configuration and basic blocks of a microprocessor. The functions of each element are as follows:

a) **ALU:** ALU stands for Arithmetical Logical Unit. As name indicates it has two parts:

1) Arithmetical unit which is responsible for mathematical operations like addition, subtraction, multiplication and division.
2) Logical unit which is dedicated to take logical decisions like greater than, less than, equal to, not equal to etc. (Basically AND/OR/NOT Operations).

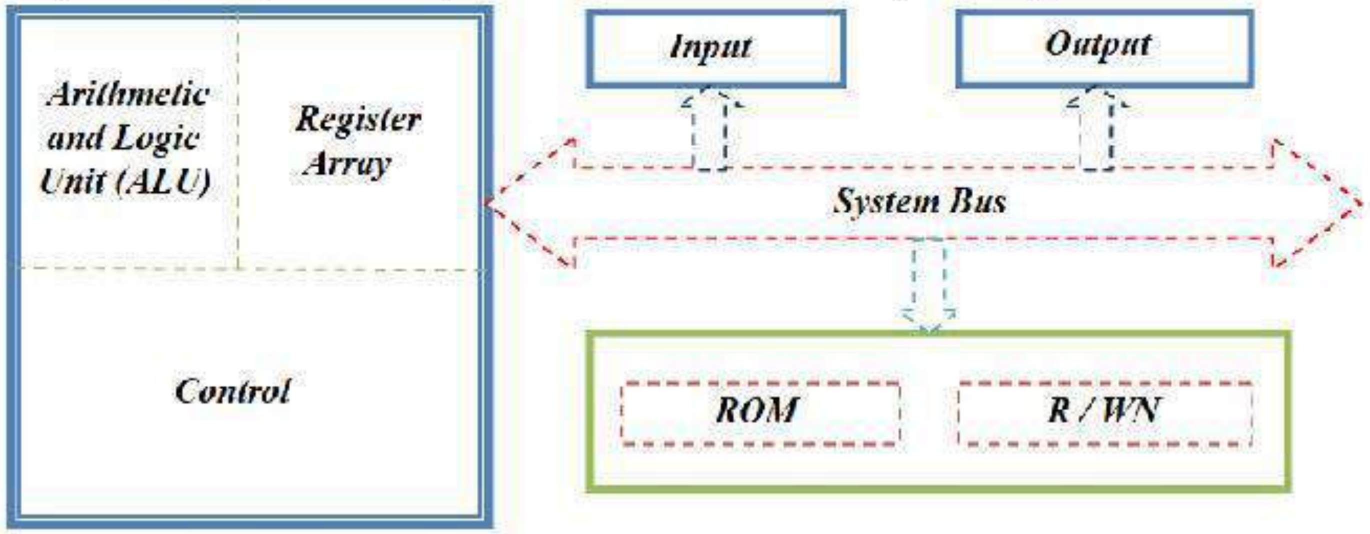


Figure.2 Schematic of configuration of a micro-processor

Micro-Processors Name	Date	No. of Transistors	Width of smallest wire on chip	Clock Speed	Data Width (In Bits)	Millions of Instructions per second (MIPS)
8080	1974	6000	6	2MHz	8	0.64
8088	1979	29000	3	5 MHz	16	0.33
80286	1982	134000	1.5	6MHz	16	1
80386	1985	275000	1.5	16	32	5
80486	1989	1200000	1	25	32	20
Pentium	1993	3100000	0.8	60	32	100
Pentium II	1997	7500000	0.35	233	32	300
Pentium III	1999	9500000	0.25	450	32	510
Pentium 4	2000	42000000	0.18	1.5 GHz	32	1700
Pentium 4P	2004	125000000	0.09	3.6 GHz	32	7000

Table .1. History of Micro-Processors

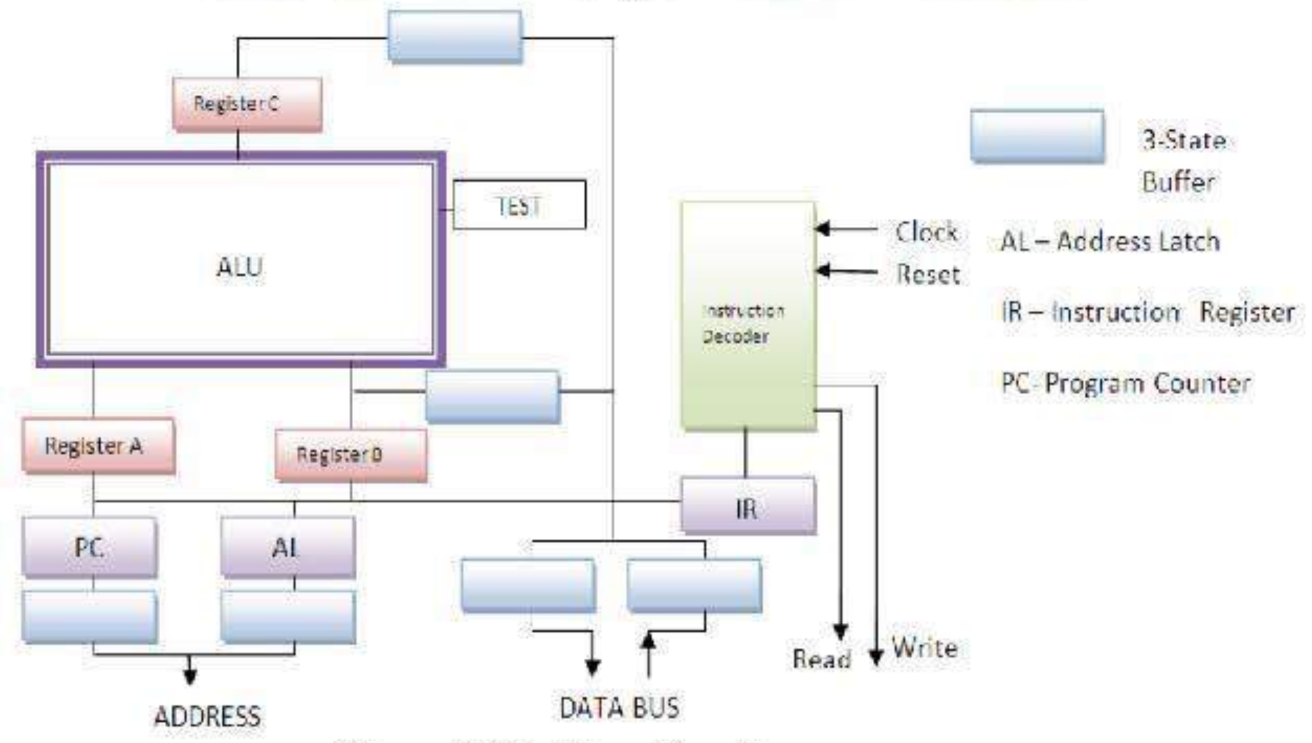


Figure.3 Working of a microprocessor

b) **Register Array:** Registers are small storage devices that are available to CPU or processors. They act as temporary storage for processing of intermediate data by mathematical or logical operations.

c) **Control:** This part of CPU is dedicated to coordinate data flow and signal flow through various types of buses i.e. Data Bus, Control Bus, and Address Bus etc. It directs data flow between CPU and storage and I/O devices.

d) **Memory:** There are two different types of memory segments being used by the CPU. First is the ROM which stands for Read Only Memory while other is R/W which stands for Read and Write Memory or Random Access Memory (RAM).

1) ***ROM:*** From this memory unit, CPU can only read the stored data. No writing operations can be done in this part of memory. Thus it is used to store the programs that need no alteration or changes like Monitor Program or Keyboard driver etc.

2) ***R/W:*** As name indicates it is opposite to ROM and used for both reading and writing operations. In general User's program and instruction are stored in this segment of memory unit.

e) **Input Devices:** Input devices are used to enter input data to microprocessor from Keyboard or from ADC which receives data from sensors/signal conditioning systems.

f) **Output Devices:** These devices display the results/conclusions coming out from ALUs either in soft copy (Monitor) or in Hard Copy (Printer).

❖ ***Functions of microprocessor***

Various functions of microprocessor are as follows:

- Microprocessor performs a variety of logical and mathematical operations using its ALU.
- It controls data flow in a system and hence can transfer data from one location to another based on the instructions given to it.
- A microprocessor can take necessary decisions and jump to a new set of instructions based on those decisions.

❖ *Elements of microprocessor*

A simple microprocessor consists of following basic elements (see Figure .3):

- Data Bus: Through data bus, the data flow between
    - a) various storage units
    - b) ALU and memory units
- Address Bus: It controls the flow of memory addresses between ALU and memory unit.
- RD (read) and WR (write) lines set or obtain the addressed locations in the memory.
- Clock line transfers the clock pulse sequence to the processor.
- Reset Line is used to restart execution and reset the processor to zero.
- Address Latch is a register which stores the addresses in the memory.
- Program Counter: It is a register which can increment its value by 1 and keeps the record of number of instructions executed. It can be set to zero when instructed.
- Test Register: It is a register which stores intermediate or in-process data of ALU operations. For example it is required to hold the 'carry' while ALU is performing 'addition' operation. It also stores the data which can be accessed by Instruction decoder to make any decision.
- 3-State Buffers: These are tri-state buffers. A tri-state buffer can go to a third state in addition to the states of 1 and 0.
- The instruction register and instruction decoder are responsible for controlling the operations of all other components of a microprocessor.
- There are following control lines present in a microprocessor, which are used to communicate instructions and data with the instruction decoder.
- Instruct the A register to latch the value currently on the data bus.
- Instruct the B register to latch the value currently on the data bus.
- Instruct the C register to latch the value currently output by the ALU.
- Instruct the program counter register to latch the value currently on the data bus.
- Instruct the address register to latch the value currently on the data bus.
- Instruct the instruction register to latch the value currently on the data bus.

- Instruct the program counter to increment.
- Instruct the program counter to reset to zero.
- Activate any of the six tri-state buffers (six separate lines).
- Instruct the ALU what operation to perform.
- Instruct the test register to latch the ALU's test bits.
- Activate the RD line.
- Activate the WR line

3. Microcomputer

Microcomputer is a microprocessor based system. It is a data processing system that employs a microprocessor as its central unit. Based on the input it takes decisions. These decisions are further used to control a system or to actuate an action or operation.

❖ *Microprocessor based programmable controller*

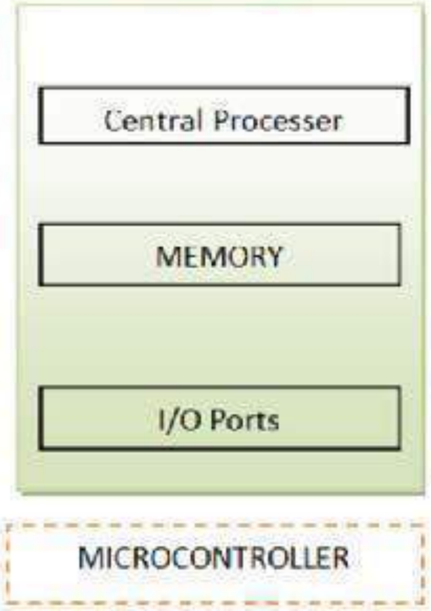


Figure.4 Schematic of microcontroller.

It is a microprocessor-based system. It implements the functions of a computer and a controller on a single chip. Generally microcontroller is programmed for one specific application and it is dedicated to a specific control function.

Microcontrollers find applications in automobiles, aircraft, medical electronics and home appliances. They are small in size and can be embedded in an electromechanical system without taking up much space. Thus we can have a system with its functions completely designed into a chip. However microcontrollers have very little user programmable memory. Various types of microcontroller chips available in market are: Motorola 68HC11, Zilog Z8 and Intel MCS51 and 96 series.

4. Programmable Logic Controllers

Any computer having input and output interfaces can be used to control external devices. However most of the computers are not industrially hardened. Input / Output devices of general-purpose microcomputers are not engineered to handle line-voltages and currents above transistor-transistor logic (TTL) levels. Also they are not designed to with-stand the temperature, humidity, and vibration on shop floors. These drawbacks of a general purpose computer have been rectified by developing a Programmable Logic Controller (PLC) with built-in isolation into their inputs and outputs.

"The programmable logic controller is defined as a digital electronic device that uses a programmable memory to store instructions and to implement functions such as logic, sequencing, timing, counting and arithmetic words to control machines and processes."

PLCs are generally used for incorporating automation in open loop systems where processes are to be performed in a sequential manner. PLCs are used for automation of assembly lines in industries. They are generally designed for multiple input multiple output (MIMO) systems. In PLCs, instructions are saved in nonvolatile memory. Some of the advantages of PLCs are:

- Cost effective
- Flexibility and ability to use similar system for other processes
- Programming interface is easier in comparison to other processers
- Resistant to impact and vibration
- Resistant towards electrical and mechanical noise
- Ability to work at high temperatures

Now let us study the structure and functioning of a PLC. Figure 5. Shows the basic elements of a PLC. It is basically a microproceeor based control system. Microprocessor communicates with the outside world with input/output devices via a circuitry. This circuitry protects the microprocessor and other elements of PLC from the high voltages and currents coming to the PLC.

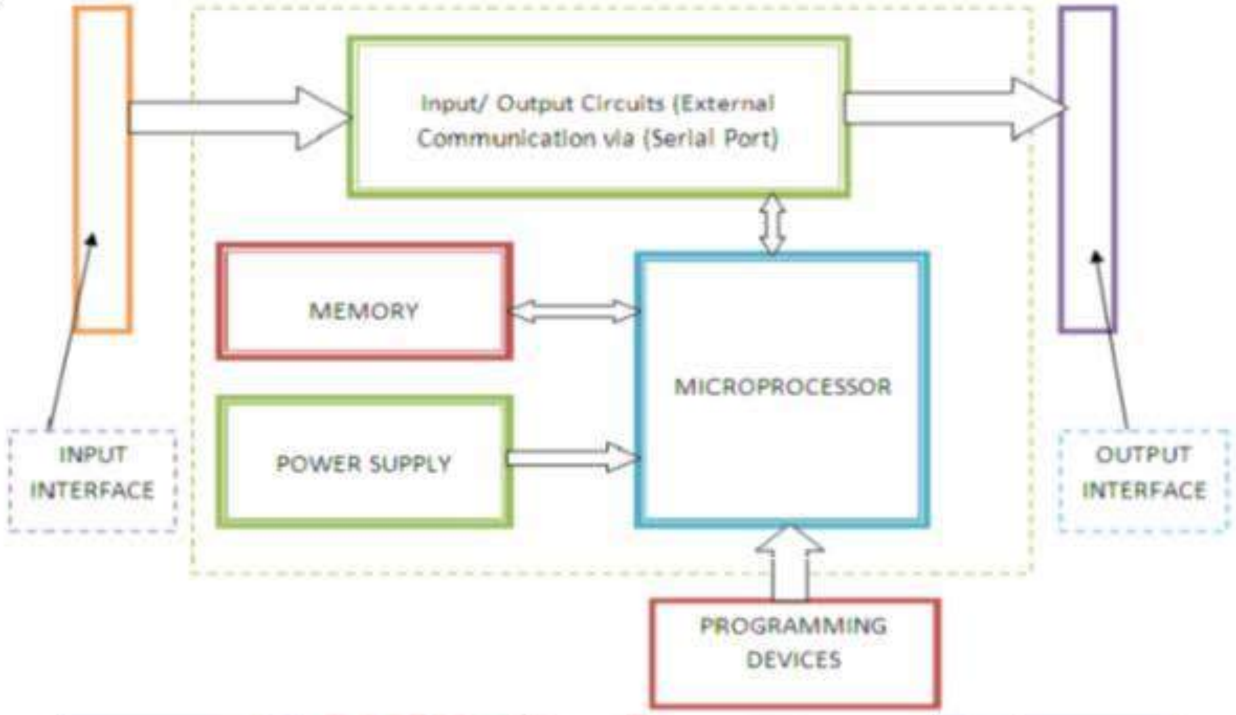


Figure.5 Block diagram of a PLC

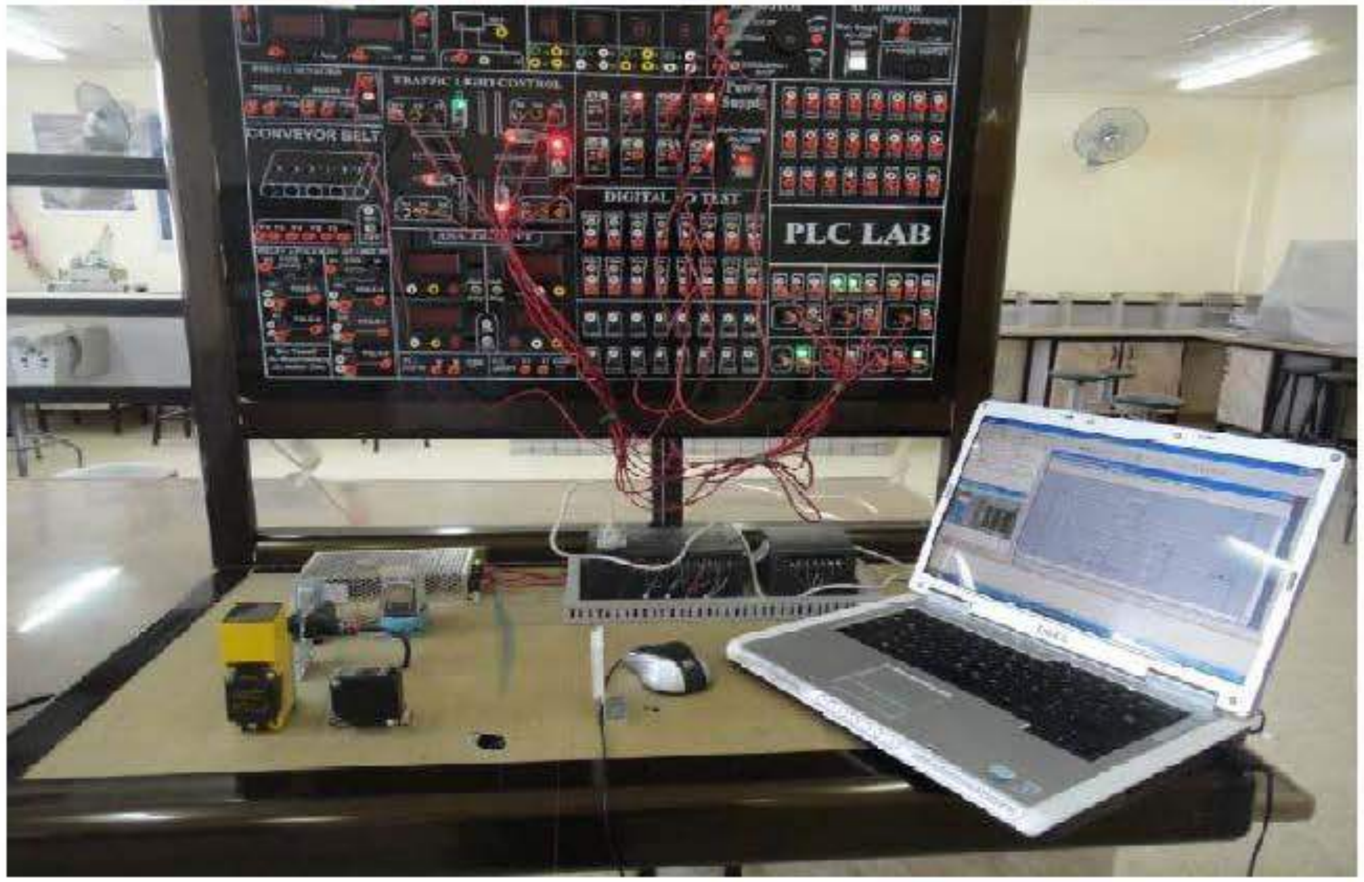


Figure 6. An Industrial PLC

Final Comparison between PLC, microcontroller and microprocessor can be summarized by the following; For PLC with microcontroller as follows:

1) A PLC will not fit in your wallet, a microcontroller will.
2) A PLC has nice LED on its cover, a microcontroller has no cover.
3) A PLC can sometime be snap unto a DIN rail, a microcontroller cannot.
4) A PLC has inbeded smoke which can be released with various techniques; a microcontroller will just change from glossy black to black.
5) A PLC is a 3 letter word, a microcontroller has more.
6) A PLC can make a nice traffic light system on its own, a microcontroller need other parts connected to it.
7) A PLC cannot be welded to a circuit board, a microcontroller can.
8) A PLC can communicate to the rest of the world, a microcontroller cannot.

While by compression between microprocessors and microcontroller we can get the following:

1) Microcontrollers have RAM, ROM, EEPROM embedded in it while we have to use external circuits in case of microprocessors.
2) As all the peripheral of microcontroller are on single chip it is compact while microprocessor is bulky.
3) Microcontrollers are far cheaper than microprocessors.
4) Microprocessors work much faster than microcontrollers.
5) Generally microcontrollers are low overall consumption of power less than microprocessors.
6) For larger applications are prefer microcontrollers not microprocessors.
7) Tasks performed by microcontrollers are less complex in microprocessors.
8) Microcontrollers are based on Harvard architecture while microprocessors are based on von Neumann model.

Hardware of PLC

Typically a PLC system has the basic functional components of processor unit, memory, power supply unit, input/output interface section, communications interface and the programming device. Figure .7 shows the basic arrangement.

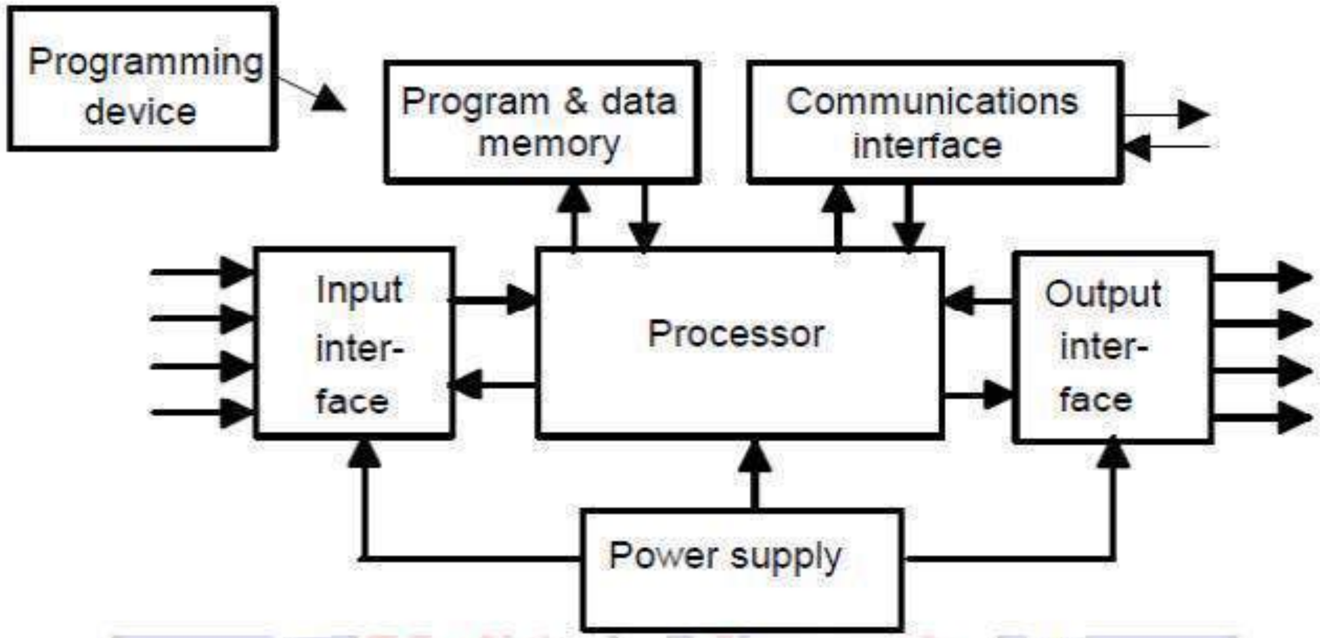


Figure .7. The PLC system

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.
2) The power supply unit is needed to convert the mains a.c. voltage to the low d.c. voltage (5V) necessary for the processor and the circuits in the input and output interface modules.
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 and data stored from the input for processing and for the output for outputting.
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, with the automatic drill, or other sensors such as photo-electric cells, temperature sensors, or flow sensors, etc. The outputs might be to motor starter coils, solenoid valves, etc. Input and output interfaces are discussed in section No. 2. Input and output devices can be classified as giving signals which are discrete, digital or analogue (Figure 8). Devices giving discrete or digital signals are ones where the signals are either off or on. 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.

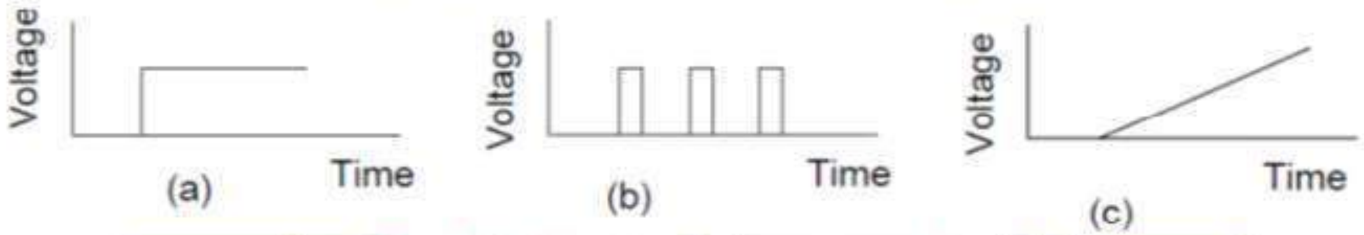


Figure 8. Signals: (a) discrete, (b) digital, (c) analogue

6) The communications interface is used to receive and transmit data on communication networks from or to other remote PLCs (Figure 9). It is concerned with such actions as device verification, data acquisition, synchronization between user applications and connection management.

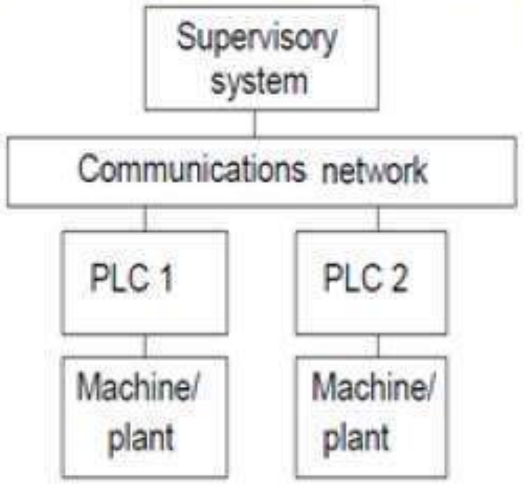


Figure .9. Basic communications model

Internal architecture of PLC

Figure 10 shows the basic internal architecture of a PLC. It consists of a central processing unit (CPU) containing the system microprocessor, memory, and input/output circuitry. The CPU controls and processes all the operations within the PLC. It is supplied with a clock with a frequency of typically between 1 and 8 MHz. This frequency determines the operating speed of the PLC and provides the timing and synchronization for all elements in the system. The information within the PLC is carried by means of digital signals.

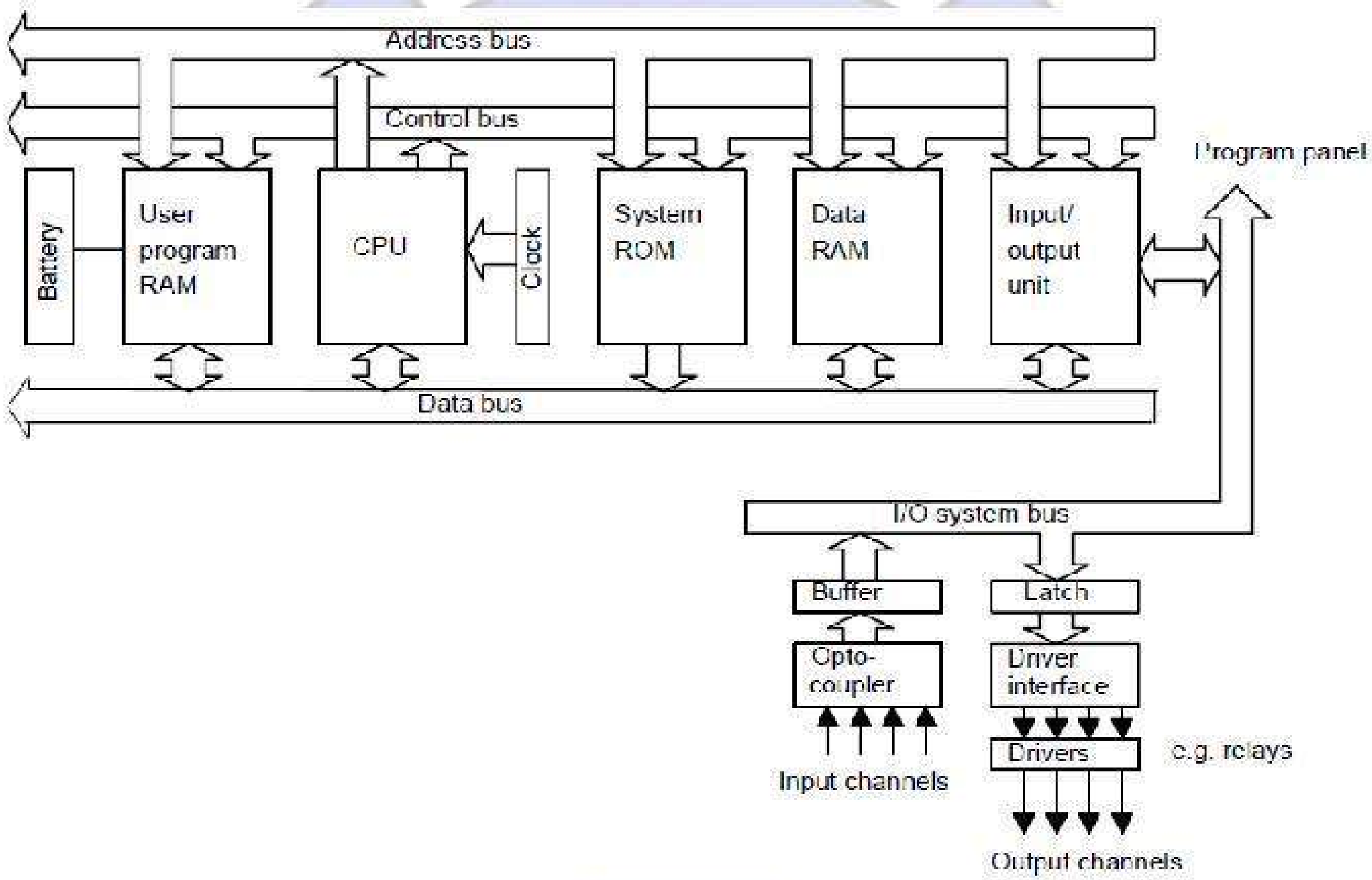


Figure .10. Architecture of a PLC

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.

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

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

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

1) The data bus carries the data used in the processing carried out by the CPU. A microprocessor termed as being 8-bit has an internal data bus which can handle 8-bit numbers. It can thus perform operations between 8-bit numbers and deliver results as 8-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 28 = 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 synchronizations.

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

- **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 storage capacity of a memory unit is determined by the number of binary words that it can store. Thus, if a memory size is 256 words then it can store 256 × 8 = 2048 bits if 8-bit words are used and 256 × 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 × 1 memory has 4 × 1 × 1024 bit locations. A 4K × 8 memory has 4 ×8 × 1024 bit locations. The term byte is used for a word of length 8 bits. Thus the 4K × 8 memory can store 4096 bytes. With a 16-bit address bus we can have 2^{16}

different addresses and so, with 8-bit words stored at each address, we can have $2^{16} \times 8$ storage locations and so use a memory of size $2^{16} \times 8/2^{10} = 64K \times 8$ which we might be as four $16K \times 8$ bit memory chips.

- **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. It is like a row of houses along a road, number 10 might be the 'house' to be used for an input from a particular sensor while number '45' might be the 'house' to be used for the output to a particular motor.

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 11 shows the principle of an optoisolator. When a digital pulse passes through the light-emitting diode, a pulse of infrared radiation is produced.

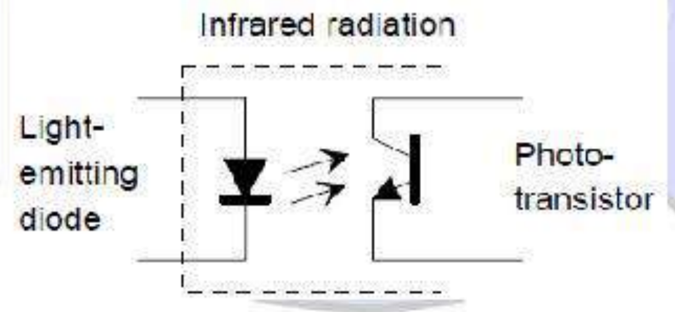


Figure .11. Optoisolator

This pulse is detected by the phototransistor and gives rise to a voltage in that circuit. The gap between the light-emitting diode and the phototransistor gives electrical isolation but the arrangement still allows for a digital pulse in one circuit to give rise to a digital pulse in another circuit.

The digital signal that is generally compatible with the microprocessor in the PLC is 5 V d.c. However, signal conditioning in the input channel, with isolation, enables a wide range of input signals to be supplied to it (see Chapter 3 for more details). 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 12). A small PLC is likely to have just one form of input, e.g. 24 V.

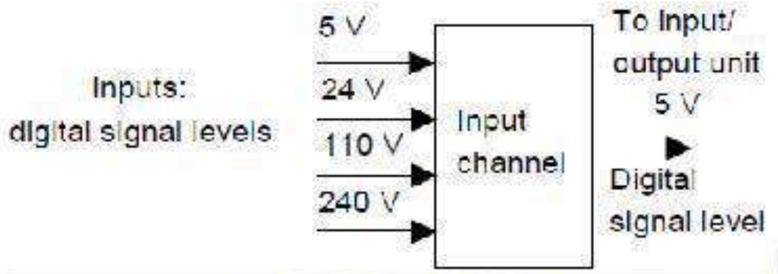


Figure .12. Input levels

The output from the input/output unit will be digital with a level of 5 V. However, after signal conditioning with relays, transistors or triacs, the output from the output channel might be a 24 V, 100 mA switching signal, a d.c. voltage of 110 V, 1 A or perhaps 240 V, 1 A a.c., or 240 V, 2 A a.c., from a triac output channel (Figure 13). 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.

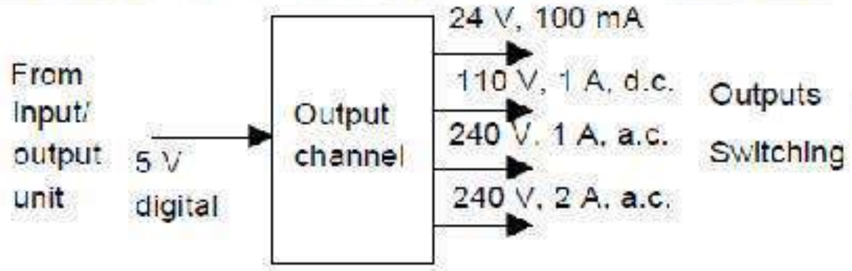


Figure .13. Output levels

Outputs are specified as being of relay type, transistor type or triac type (see Section 2 for more details):

1) With *the relay type,* the signal from the PLC output is used to operate a relay and 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 a.c. and d.c. switching. They can withstand high surge currents and voltage transients.

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 overcurrent and high reverse voltage. As a protection, either a fuse or built-in electronic protections are used. Optoisolators are used to provide isolation.

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

- **Sourcing and sinking:** The terms sourcing and sinking are used to describe the way in which d.c. devices are connected to a PLC. With sourcing, using the conventional current flow direction as from positive to negative, an input device receives current from the input module, i.e. the input module is the source of the current (Figure 14(a)). If the current flows from the output module to an output load then the output module is referred to as sourcing (Figure 14(b)). With sinking, using the conventional current flow direction as from positive to negative, an input device supplies current to the input module, i.e. the input module is the sink for the current (Figure 14(c)). If the current flows to the output module from an output load then the output module is referred to as sinking (Figure 14(d)).

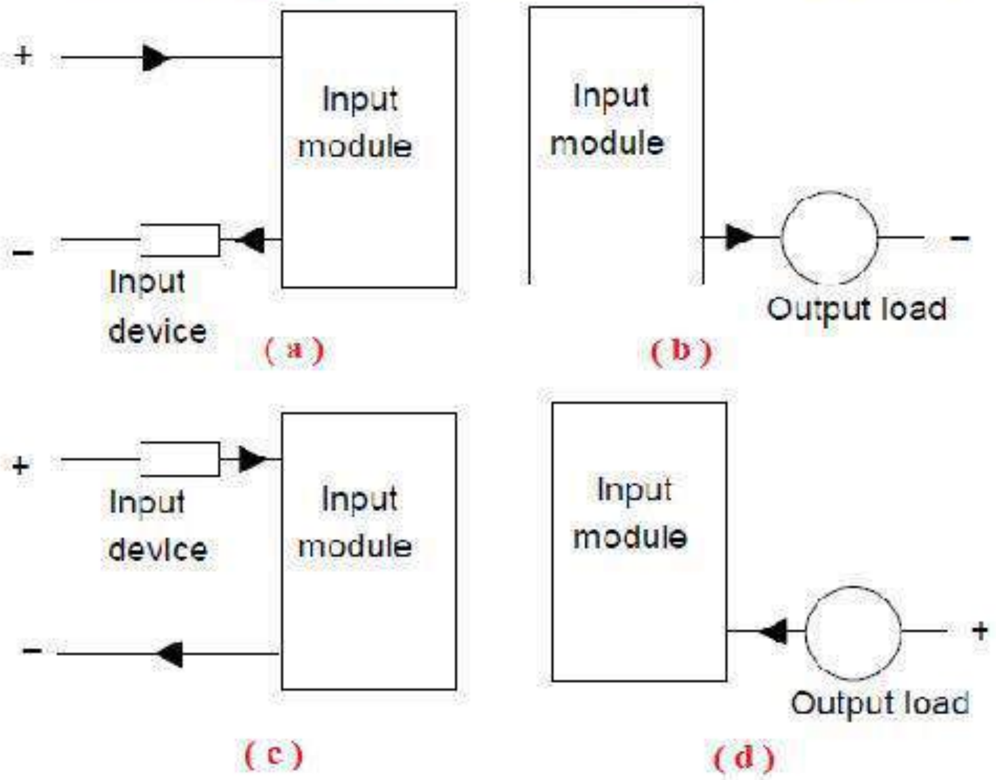


Figure .14. Sourcing and sinking

PLC systems

There are two common types of mechanical design for PLC systems; a single box, and the modular/rack types. The single box type (or, as sometimes termed, brick) 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. Typically such a PLC might have 6, 8, 12 or 24 inputs and 4, 8 or 16 outputs and a memory which can store some 300 to 1000 instructions. Figure 15 shows the GLOFA GM7 compact, i.e. brick, PLC and Table 2 gives details of models in that GLOFA range.

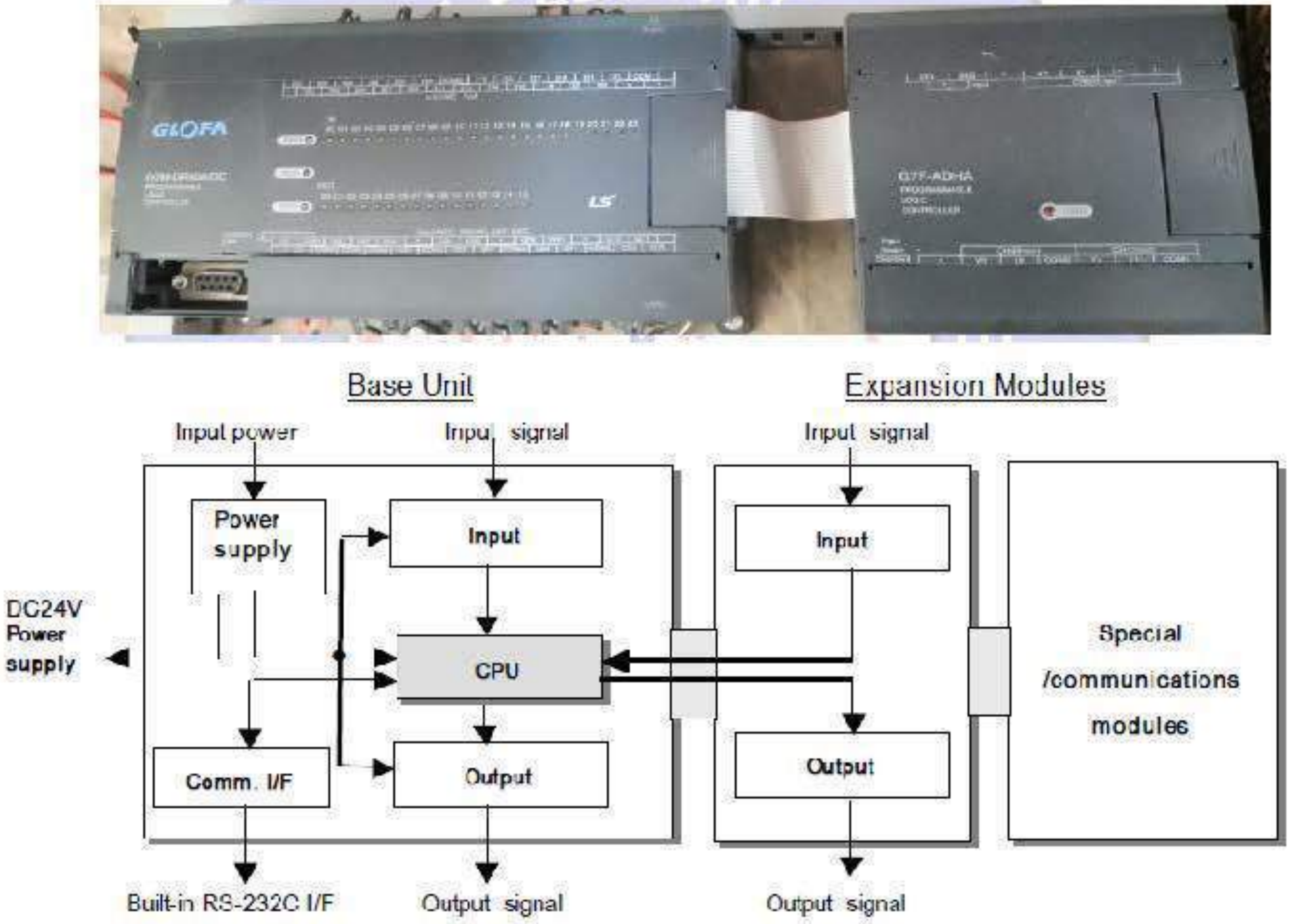
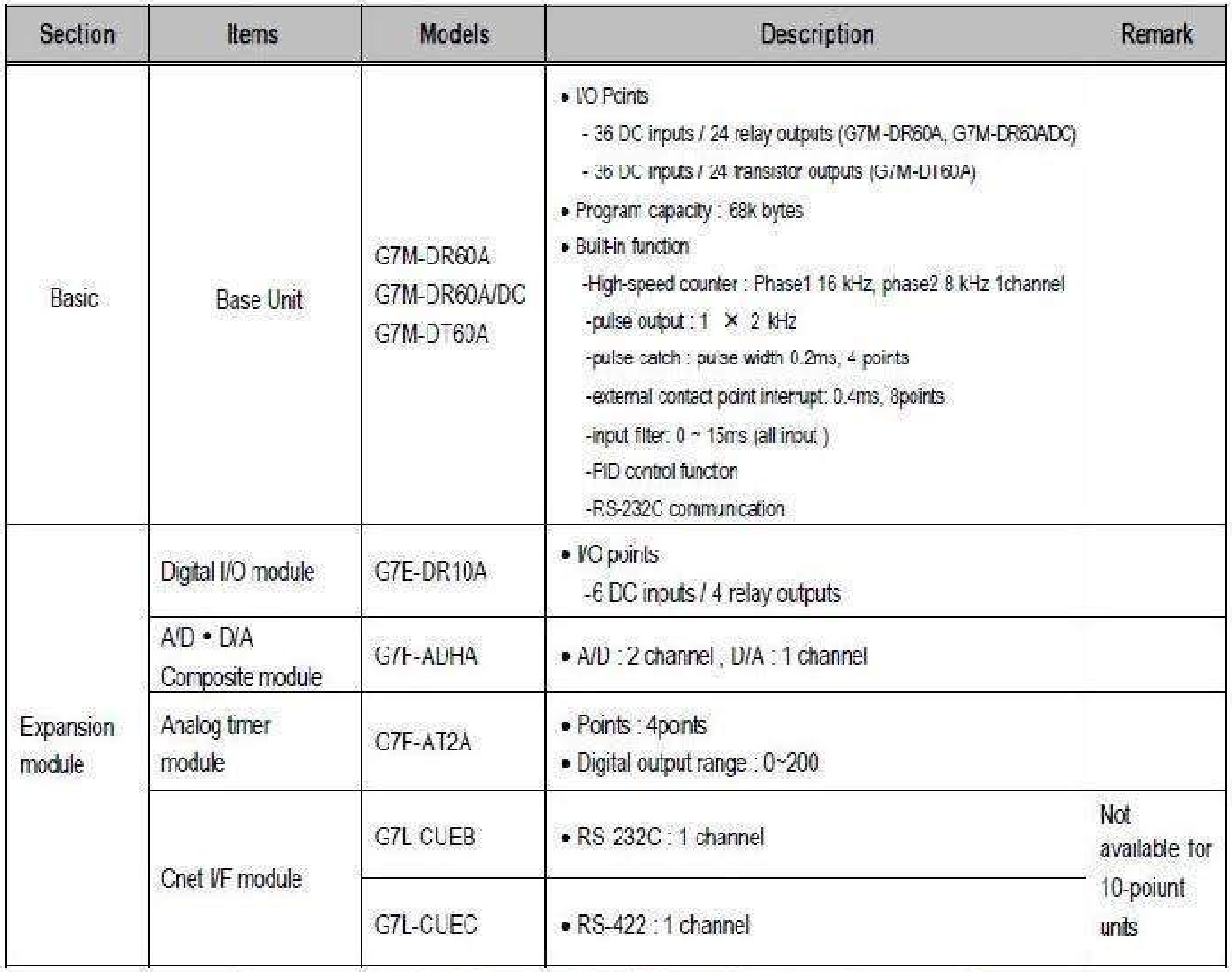


Figure. 15. The GLOFA GM7 compact and its flowchart



Section	Items	Models	Description	Remark
Basic	Base Unit	G7M-DR60A G7M-DR60A/DC G7M-DT60A	• I/O Points - 36 DC inputs / 24 relay outputs (G7M-DR60A, G7M-DR60ADC) - 36 DC inputs / 24 transistor outputs (G7M-DT60A) • Program capacity : 68k bytes • Built-in function -High-speed counter : Phase1 16 kHz, phase2 8 kHz 1channel -pulse output : 1 × 2 kHz -pulse catch : pulse width 0.2ms, 4 points -external contact point interrupt: 0.4ms, 8points -input filter: 0 ~ 15ms (all input) -PID control function -RS-232C communication	
Expansion module	Digital I/O module	G7E-DR10A	• I/O points -6 DC inputs / 4 relay outputs	
	A/D • D/A Composite module	G7F-ADHA	• A/D : 2 channel , D/A : 1 channel	
	Analog timer module	G7F-AT2A	• Points : 4ponts • Digital output range : 0~200	
	Cnet I/F module	G7L-CUEB	• RS-232C : 1 channel	Not available for 10-poiunt units
		G7L-CUEC	• RS-422 : 1 channel	

Table 2. Details of GLOFA models.

Systems with larger numbers of inputs and outputs are likely to be modular and designed to fit in racks. 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. The mix of modules required for a particular purpose is decided by the user and the appropriate ones then plugged into the rack. Thus it is comparatively easy to expand the number of input/output (I/O) connections by just adding more input/output modules or to expand the memory by adding more memory units. An example of such a modular system is provided by the Allen-Bradley

PLC-5 PLC of Rockwell automation (Figure. 16). PLC-5 processors are available in a range of I/O capacity and memory size, and can be configured for a variety of communication networks. They are single-slot modules that are placed in the left-most slot of a 1771 I/O chassis. Some 1771 I/O chassis are built for back-panel mounting and some are built for rack mounting and are available in sizes of 4, 8, 12, or 16 I/O module slots. The 1771 I/O modules are available in densities of 8, 16, or 32 I/O per module. A PLC-5 processor can communicate with I/O across a DeviceNet or Universal Remote I/O link. A large selection of 1771 input/output modules, both digital and analogue, are available for use in the local chassis, and an even larger selection available for use at locations remote from the processor. Digital I/O modules have digital I/O circuits that interface to on/off sensors such as pushbutton and limit switches; and on/off actuators such as motor starters, pilot lights, and annunciators.

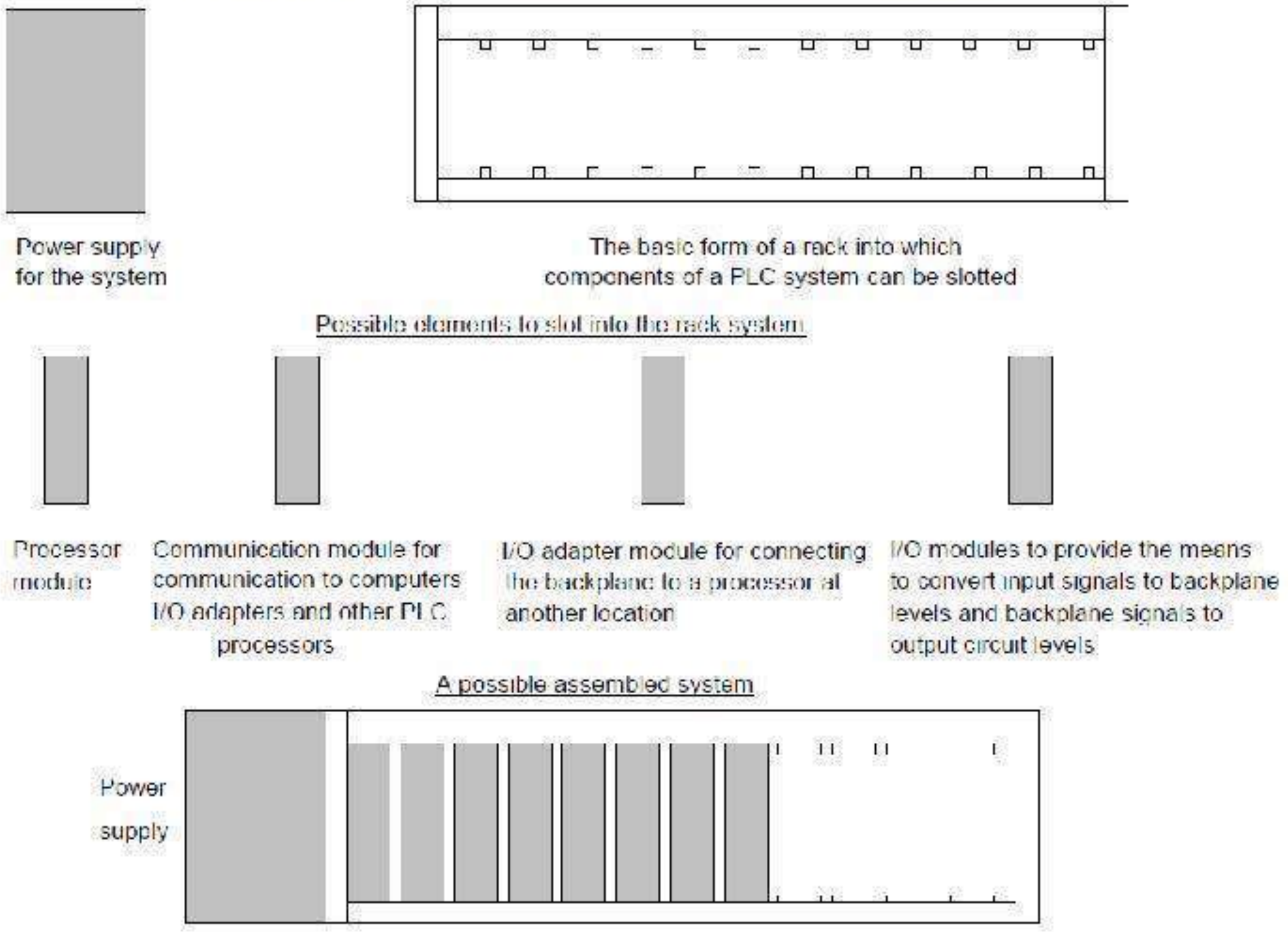


Figure. 16. A possible arrangement of a rack system

Analogue I/O modules perform the required A/D and D/A conversions using up to 16-bit resolution. Analogue I/O can be user-configured for the desired fault-response state in the event that I/O communication is disrupted. This feature provides a safe reaction/response in case of a fault, limits the extent of faults, and provides a predictable fault response. 1771 I/O modules include optical coupling and filter circuitry for signal noise reduction.

Digital I/O modules cover electrical ranges from 5…276V a.c. or d.c. and relay contact output modules are available for ranges from 0…276 V ac or 0…175 V dc. A range of analogue signal levels can be accomodated, including standard analogue inputs and outputs and direct thermocouple and RTD temperature inputs.

❖ Programming PLCs

Programming devices can be a hand-held device, a desktop console or a computer. Only when the program has been designed on the programming device and is ready is it transferred to the memory unit of the PLC.

a) *Hand-held programming* devices will normally contain enough memory to allow the unit to retain programs while being carried from one place to another.

b) *Desktop consoles* are likely to have a visual display unit with a full keyboard and screen display.

c) *Personal computers* are widely configured as program development work-stations. Some PLCs only require the computer to have appropriate software; others require 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 CD and copies easily made.

PLC manufacturers have programming software for their PLCs. For example, Mitsubishi have MELSOFT. Their GX Developer supports all MELSEC controllers from the compact PLCs of the MELSEC FX series to the modular PLCs including MELSEC System Q and uses a Windows based environment. It supports the programming methods (see Chapter 4) of instruction list (IL), ladder diagram (LD) and sequential function chart (SFC) languages. You can switch back and forth between IL and LD at will while you are working.

Section Two

Input-output devices

- **Input devices**

  Mechanical switches ; Proximity switches

  Photoelectric sensors and switches

  Encoders ; Temperature sensors

  Position/displacement sensors

  Strain gauges ; Pressure sensors ; Smart sensors

- **Output devices**

  Relay ; Directional control valves ; Motors

  Stepper motors

- **Examples of applications**

  Conveyor belt ; Traffic Light system

  Robot control system ; Liquid level monitoring

This section is a brief consideration of typical input and output devices used with PLCs. The input devices considered include digital and analogue devices such as mechanical switches for position detection, proximity switches, photoelectric switches, encoders, temperature and pressure switches, potentiometers, linear variable differential transformers, strain gauges, thermistors, thermotransistors and thermocouples. Output devices considered include relays, contactors, solenoid valves and motors.

Input devices

The term sensor is used for an input device that provides a usable output in response to a specified physical input. For example, a thermocouple is a sensor which converts a temperature difference into an electrical output. The term transducer is generally used for a device that converts a signal from one form to a different physical form. Thus sensors are often transducers, but also other devices can be transducers, e.g. a motor which converts an electrical input into rotation. 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 some of the more common terms used to define the performance of sensors.

1) Accuracy is the extent to which the value indicated by a measurement system or element might be wrong. For example, a temperature sensor might have an accuracy of ±0.1oC. The error of a measurement is the difference between the result of the measurement and the true value of the quantity being measured errors can arise in a number of ways, e.g. the term non-linearity error is used for the error that occurs as a result of assuming a linear relationship between the input and output over the working range, i.e. a graph of output plotted against input is assumed to give a straight line. Few systems or elements, however, have a truly linear relationship and thus errors occur as a result of the assumption of linearity (Figure 1(a)). The term hysteresis error (Figure 1(b)) is used for the difference in outputs given from the same value of quantity being measured according to whether that value has been reached by a continuously increasing change or a continuously decreasing change. Thus, you might obtain a different value from a thermometer used to measure the

same temperature of a liquid if it is reached by the liquid warming up to the measured temperature or it is reached by the liquid cooling down to the measured temperature.

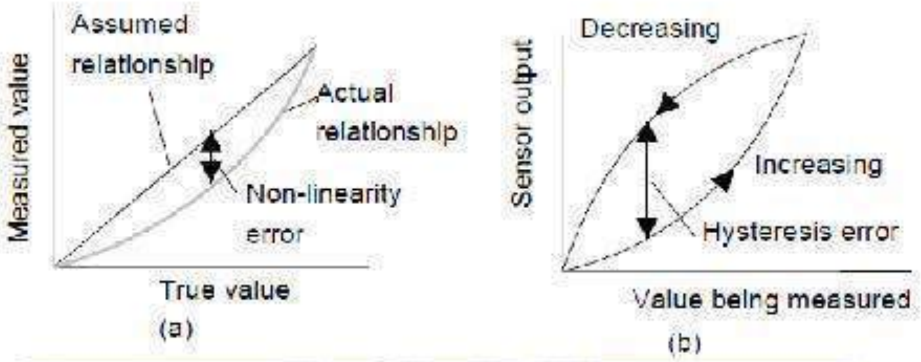


Figure.1. Some sources of error: (a) non-linearity, (b) hysteresis

2) The range of variable of system is the limits between which the input can vary. For example, a resistance temperature sensor might be quoted as having a range of −200 to +800oC.

3) When the input value to a sensor changes, it will take some time to reach and settle down to the steady-state value (Figure 2). The response time is the time which elapses after the input to a system or element is abruptly increased from zero to a constant value up to the point at which the system or element gives an output corresponding to some specified percentage, e.g. 95%, of the value of the input. The rise time is the time taken for the output to rise to some specified percentage of the steady-state output. Often the rise time refers to the time taken for the output to rise from 10% of the steady-state value to 90 or 95% of the steady-state value. The settling time is the time taken for the output to settle to within some percentage, e.g. 2%, of the steady-state value.

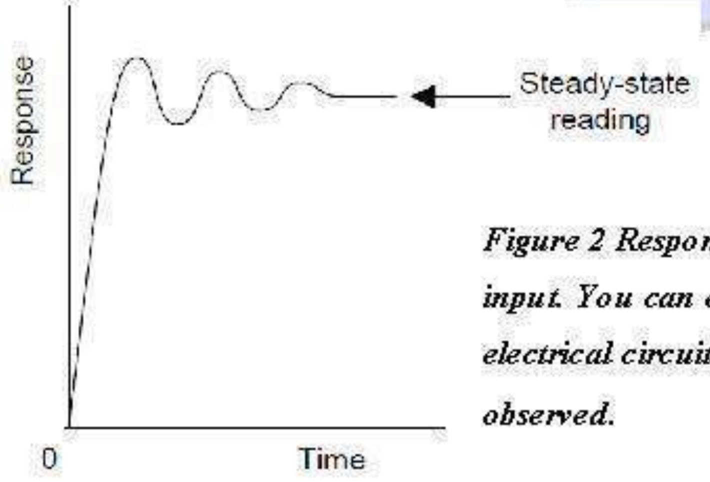


Figure 2 Response of a sensor or measurement system to a sudden input. You can easily see such a response when the current in an electrical circuit is suddenly switched on and an ammeter reading observed.

4) The sensitivity indicates how much the output of an instrument system or system element changes when the quantity being measured changes by a given amount, i.e. the ratio ouput/input. For example, a thermocouple might have a sensitivity of 20 µV/oC and so give an output of 20 µV for each 1°C change in temperature.

5) The stability of a system is its ability to give the same output when used to measure a constant input over a period of time. The term drift is often used to describe the change in output that occurs over time. The drift may be expressed as a percentage of the full range output. The term zero drift is used for the changes that occur in output when there is zero input.

6) The term repeatability is used for the ability of a measurement system to give the same value for repeated measurements of the same value of a variable. Common cause of lack of repeatability are random fluctuations in the environment, e.g. changes in temperature and humidity. The error arising from repeatability is usually expressed as a percentage of the full range output. For example, a pressure sensor might be quoted as having a repeatability of ±0.1% of full range. Thus with a range of 20 kPa this would be an error of ±20 Pa.

7) The reliability of a measurement system, or element in such a system, is defined as being the probability that it will operate to an agreed level of performance, for a specified period, subject to specified environmental conditions. The agreed level of performance might be that the measurement system gives a particular accuracy.

The following are examples of some of the commonly used PLC input devices and their sensors.

a) *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 workpiece on a machining table, the workpiece pressing against the switch and so closing it. The absence of the workpiece is indicated by the switch being open and its presence by it being closed. Thus, with the arrangement shown in Figure 3(a), the input signals to a single input channel of the PLC are thus the logic levels:

Workpiece not present 0 ; Workpiece present 1

The 1 level might correspond to a 24 V d.c. input, the 0 to a 0 V input.

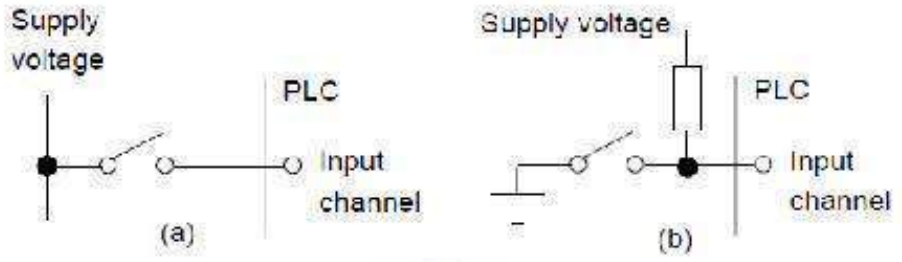


Figure .3 Switch sensors

With the arrangement shown in Figure 3(b), when the switch is open the supply voltage is applied to the PLC input, when the switch is closed the input voltage drops to a low value. The logic levels are thus:

Workpiece not present 1 ; Workpiece 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. An NO 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. 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 4 shows some examples. The cam (Figure 4(c)) can be rotated at a constant rate and so switch the switch on and off for particular time intervals.

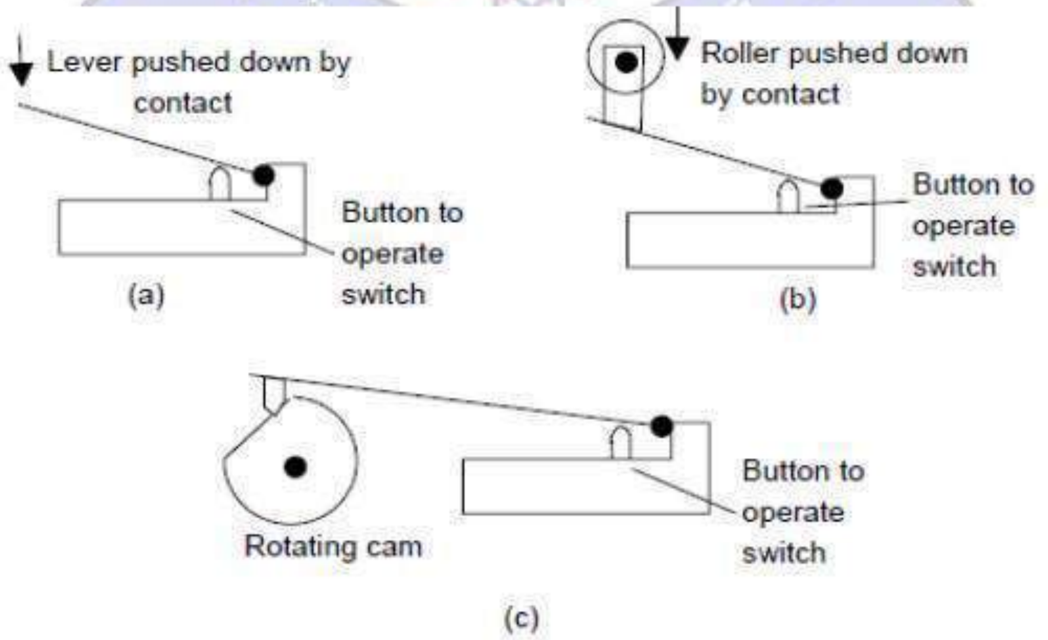


Figure .4 Limit switches actuated by: (a) lever, (b) roller, (c) cam

b) 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 5(a)). 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.

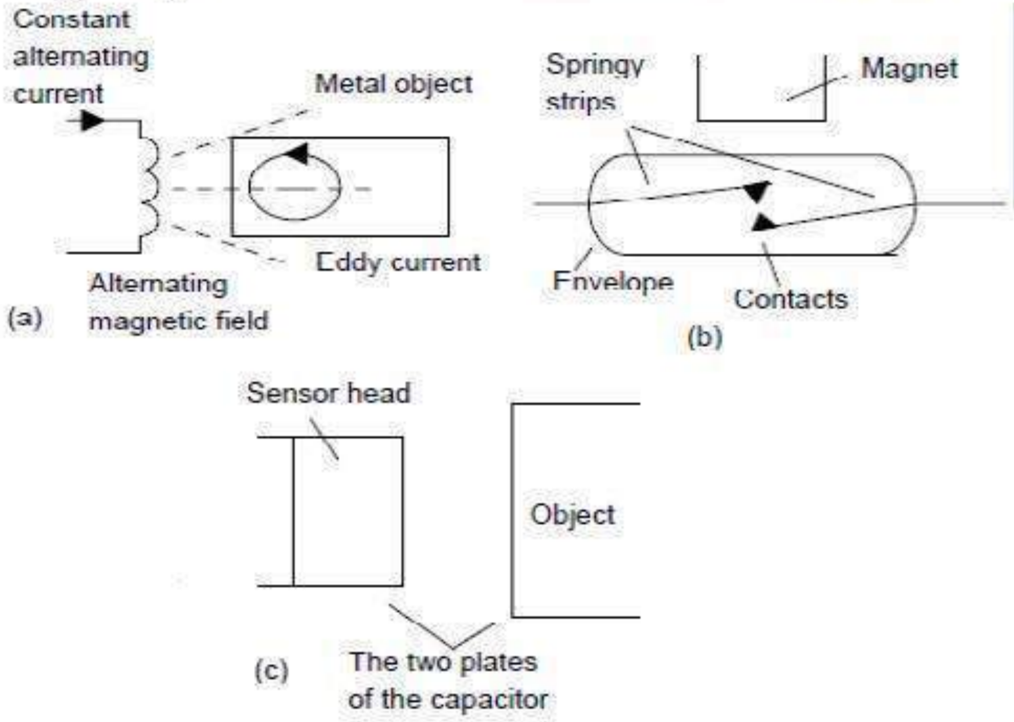


Figure 5. Proximity switches: (a) eddy current, (b) reed switch, (c) capacitive

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 5(b)). When a magnet or current-carrying coil is brought close to the switch, the strips become magnetised 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. 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 5(c)). 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. 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.

c) ***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 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 6(c) shows a U-shaped form where the object breaks the light beam. Another possibility is a photodiode. Depending on the circuit used, the output can be made to switch to either high or low when light strikes the diode. Yet another possibility is a photoconductive cell. The resistance of the photoconductive cell, often cadmium sulphide, depends on the intensity of the light falling on it.

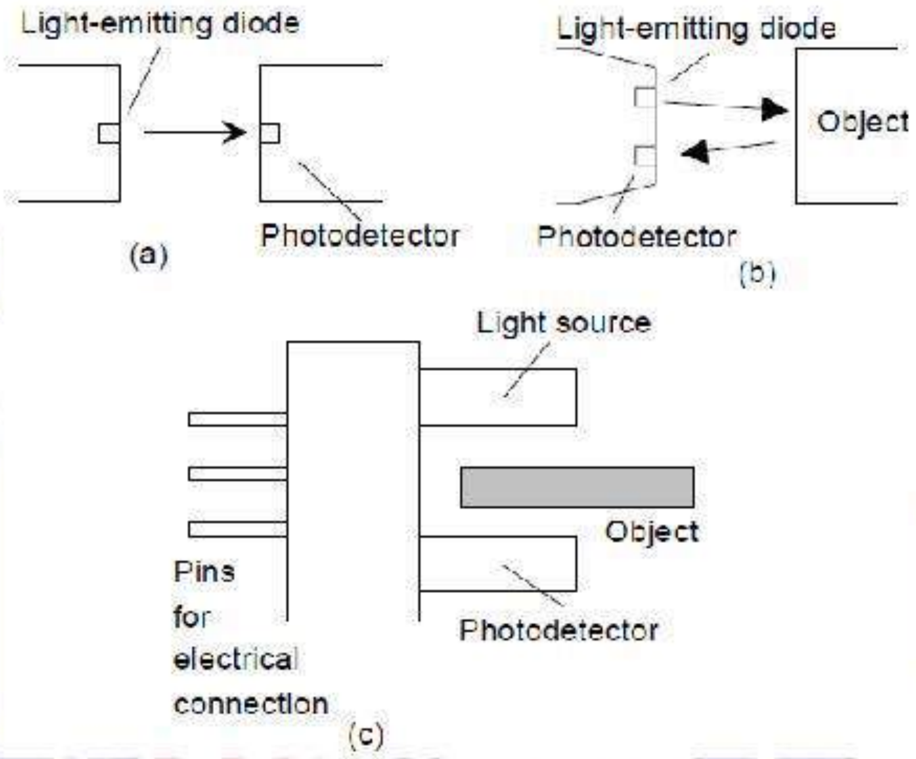


Figure 6. Photoelectric sensors

With the above sensors, light is converted to a current, voltage or resistance change. If the output is to be used as a measure of the intensity of the light, rather than just the presence or absence of some object in the light path, the signal will need amplification and then conversion from analogue to digital by an analogue-to-digital converter. An alternative to this is to use a light-to-frequency converter, the light then being converted to a sequence of pulses with the frequency of the pulses being a measure of the light intensity. Integrated circuit sensors are available, e.g. the Texas Instrument TSL220, incorporating the light sensor and the voltage to-frequency converter (Figure 7).

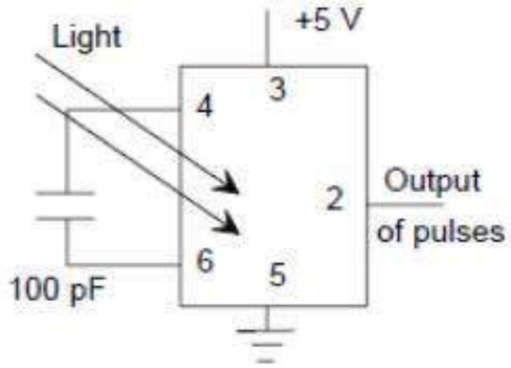


Figure .7 TSL220

d) 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 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 photodiode or phototransistor. 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 360o, a movement from one slot to the next is a rotation of 6o. By using offset slots it is possible to have over a thousand slots for one revolution and so much higher resolution.

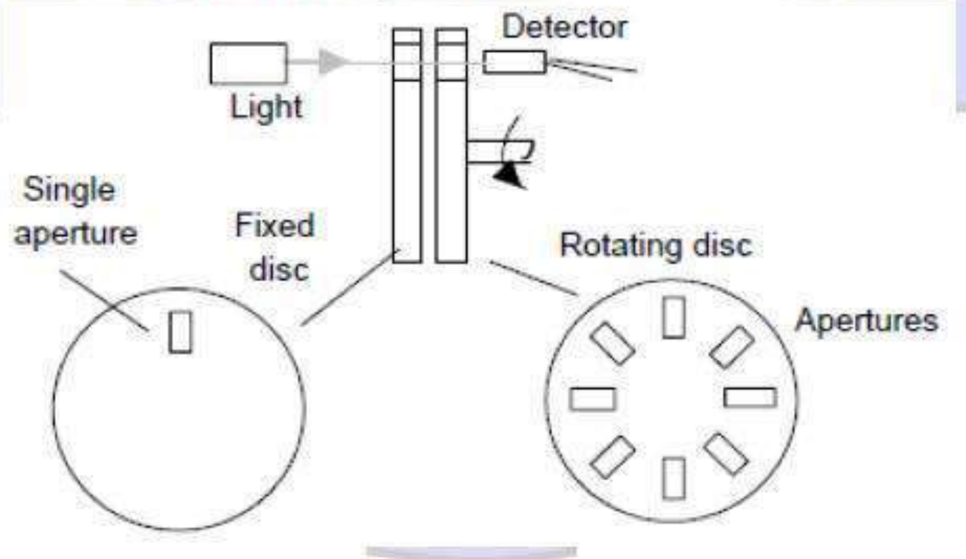


Figure 8. Basic form of an incremental encoder

The *absolute encoder* differs from the incremental encoder in having a pattern of slots which uniquely defines each angular position. With the form shown in Figure 9, the rotating disc has four concentric circles of slots and four sensors to detect the light pulses. The slots are arranged in such a way that the sequential output from the sensors is a number in the binary code, each such number corresponding to a particular angular position. With 4 tracks there will be 4 bits and so the number of positions that can be detected is $2^4 = 16$, i.e. a resolution of 360/16 = 22.5°. Typical encoders tend to have up to 10 or 12 tracks. The number of bits in the binary number will be equal to the number of tracks. Thus with 10 tracks there will be 10 bits and so the number of positions that can be detected is 2^{10}, i.e. 1024, a resolution of 360/1024 = 0.35 °.

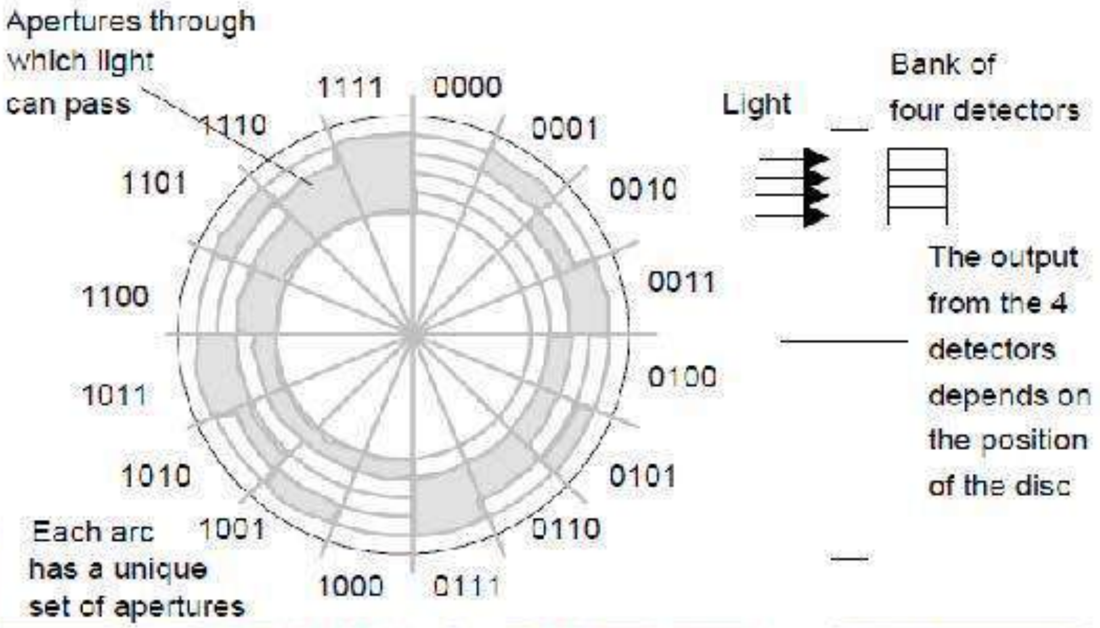


Figure 9. The rotating wheel of the absolute encoder. Note that though the normal form of binary code is shown in the figure, in practice a modified form of binary code called the Gray code is generally used. This code, unlike normal binary, has only one bit changing in moving from one number to the next. Thus we have the sequence 0000, 0001, 0011, 0010, 0011, 0111, 0101, 0100, 1100, 1101, 1111.

e) 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 10(a)). 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.

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 vary in a linear manner with temperature over a wide range of temperatures, though the actual change in resistance per degree is fairly small. Semiconductors, such as thermistors, show very large changes in resistance with temperature. The change, however, is non-linear. Such detectors can be used as one arm of a Wheatstone bridge and the output of the bridge taken as a measure of the temperature (Figure 10(b)). 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 10(c)). The output from either type of circuit is an analogue signal which is a measure of the temperature.

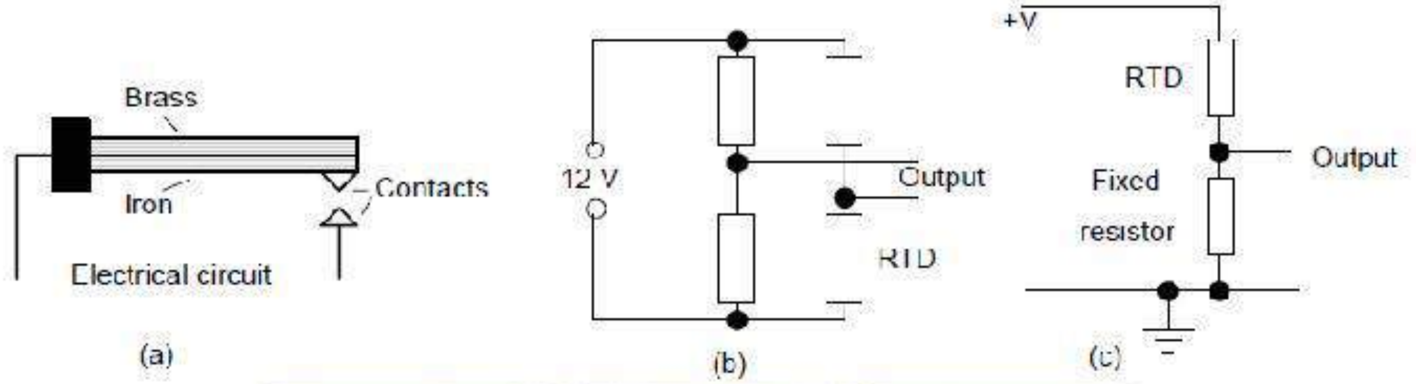


Figure 10. (a) Bimetallic strip (b) Wheatstone bridge, (c) potential divider circuits

Thermodiodes and *thermotransistors* are used as temperature sensors since the rate at which electrons and holes diffuse across semiconductor junctions is affected by the temperature. 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/°C when the supply voltage is +5 V (Figure 11(a)). A digital temperature switch can be produced with an analogue sensor by feeding the analogue output into a comparator amplifier which compares it with some set value, producing an output giving a logic 1 signal when the temperature voltage input is equal to or greater than the set point and otherwise an output which gives a logic 0 signal. Integrated circuits, e.g. LM3911N, are available, combining a thermotransistor temperature-sensitive element with an operational amplifier. When the connections to the chip are so made that the amplifier is connected as a comparator (Figure 11(b)), then the output will switch as the temperature traverses the set point and so directly give an on−off temperature controller.

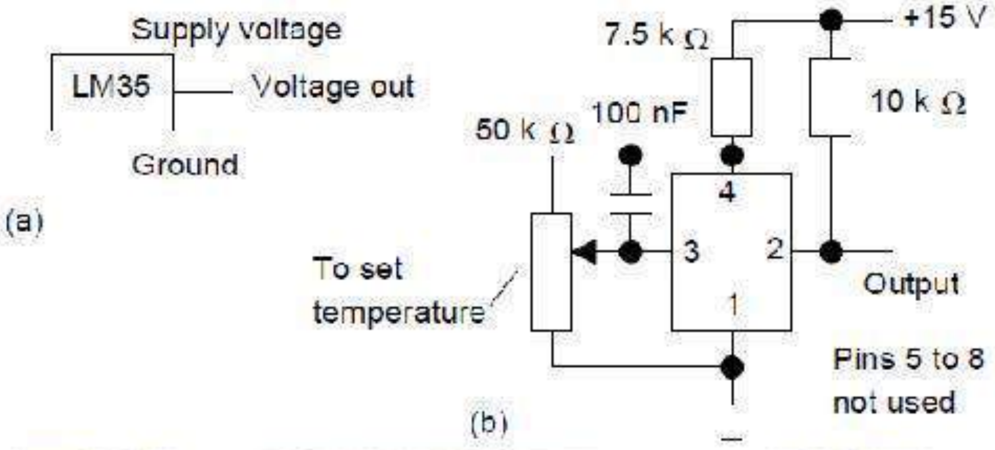


Figure 11. (a) LM35, (b) LM3911N circuit for on−off control

Another commonly used temperature sensor is the thermocouple. The *thermocouple* consists essentially of two dissimilar wires A and B forming a junction (Figure 12). 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.

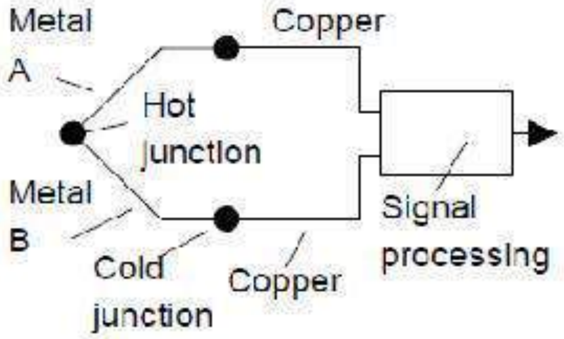
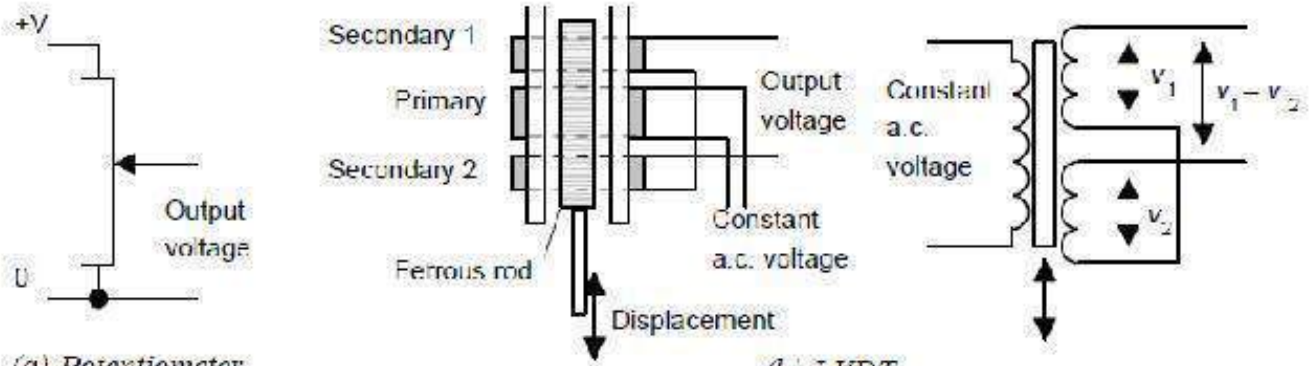


Figure 12. Thermocouple

f) Position/displacement sensors:

The term position sensor is used for a sensor that gives a measure of the distance between a reference point and the current location of the target, a displacement sensor being one that gives a measure of the distance between the present position of the target and the previously recorded position. Resistive linear and angular position sensors are widely used and relatively inexpensive. These are also called linear and rotary potentiometers. A d.c. voltage is provided across the full length of the track and the voltage signal between a contact which slides over the resistance track and one end of the track is related to the position of the sliding contact between the ends of the potentiometer resistance track (Figure 13 (a)). The potentiometer thus provides an analogue linear or angular position sensor. Another form of displacement sensor is the linear variable differential transformer (LVDT), this giving a voltage output related to the position of a ferrous rod. The LVDT consists of three symmetrically placed coils through which the ferrous rod moves (Figure 13 (b)).



(a) Potentiometer   *(b) LVDT*

Figure 13. (a) potentiometer; (b) linear variable differential transformer (LVDT).

When an alternating current is applied to the primary coil, alternating voltages, v1 and v2, are induced in the two secondary coils. When the ferrous rod core is centred 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, i.e. v1 – v2. 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. Capacitive displacement sensors are essentially just parallel plate capacitors. The capacitance will change if the plate separation changes, the area of overlap of the plates changes, or a slab of dielectric is moved into or out of the plates (Figure 14). All these methods can be used to give linear displacement sensors. The change in capacitance has to be converted into a suitable electrical signal by signal conditioning.

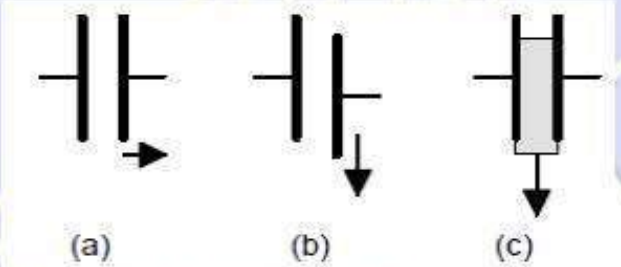


Figure 14. Capacitor sensors: (a) changing the plate separation, (b) changing the area of overlap, (c) moving the dielectric

g) Strain gauges:

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

$$\frac{\Delta R}{R} = G \times strain$$

where ΔR is the change in resistance for a wire of resistance R and G is a constant called the gauge factor. For metals the gauge factor is about 2 and for semiconductors about 100. Metal resistance strain gauges are in the form of a flat coil in order to get a reasonable length of metal in a small area. Often they are etched from metal foil (Figure 15(a)) and attached to a backing of thin plastic film so that they can be stuck on surfaces, like postage stamps on an envelope. The change in resistance of the strain gauge, when subject to strain, is usually converted into a voltage signal by the use of a Wheatstone bridge (Figure 15(b)). A problem that occurs is that the resistance of the strain gauge also changes with temperature and thus some means of temperature compensation has to be used so that the output of the bridge is only a function of the strain. This can be achieved by placing a dummy strain gauge in an opposite arm of the bridge, that gauge not being subject to any strain but only the temperature (Figure 16).

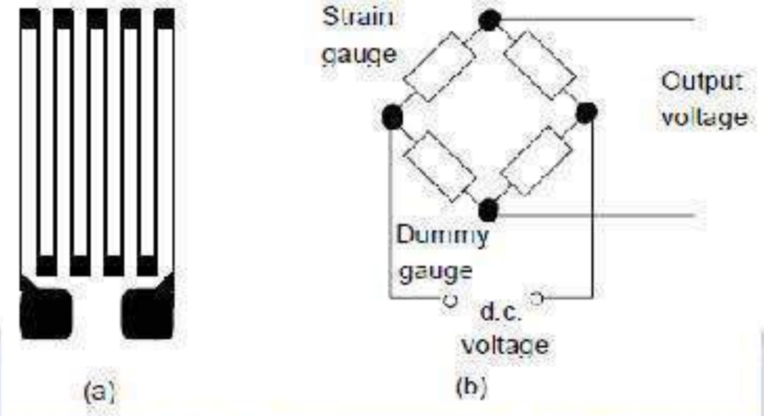


Figure 15. (a) Metal foil strain gauge, (b) Wheatstone bridge circuit with compensation for temperature changes

An alternative which is widely used is to use four active gauges as the arms of the bridge and arrange it so that one pair of opposite gauges are in tension and the other pair in compression. This not only gives temperature compensation but also gives a much larger output change when strain is applied. The following paragraph illustrates systems employing such a form of compensation. By attaching strain gauges to other devices, changes which result in strain of those devices can be transformed, by the strain gauges, to give voltage changes. They might, for example, be attached to a cantilever to which forces are applied at its free end (Figure 16(a)). The voltage change, resulting from the strain gauges and the Wheatstone bridge, then becomes a measure of the force. Another possibility is to attach strain gauges to a diaphragm which deforms as a result of pressure (Figure 16(b)). The output from the gauges, and associated Wheatstone bridge, then becomes a measure of the pressure.

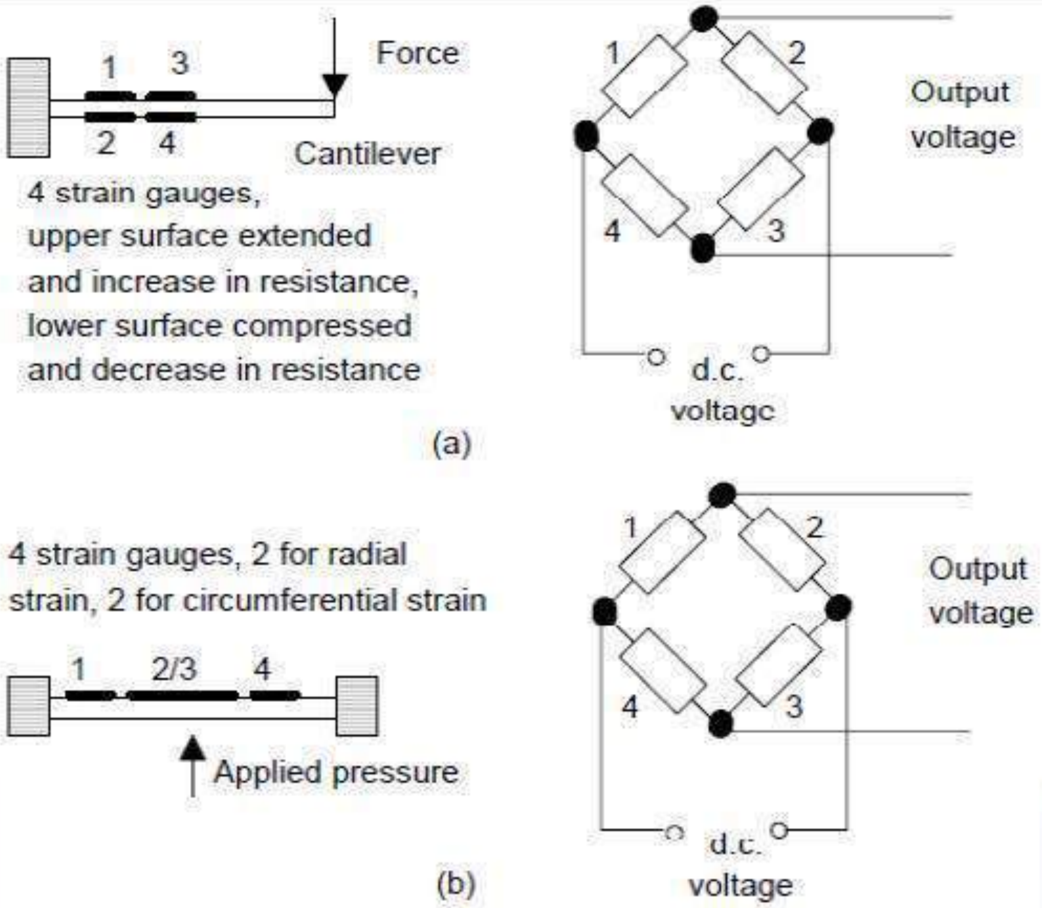


Figure 16. Strain gauges used for (a) force sensor, (b) pressure sensor

h) Pressure sensors:

Commonly used pressure sensors which give responses related to the pressure are diaphragm and bellows types. The diaphragm type consists of a thin disc of metal or plastic, secured round its edges. When there is a pressure difference between the two sides of the diaphragm, the centre of it deflects. The amount of deflection is related to the pressure difference. This deflection may be detected by strain gauges attached to the diaphragm (see Figure 16(b)), by a change in capacitance between it and a parallel fixed plate or by using the deflection to squeeze a piezoelectric crystal (Figure 17(a)). When a piezoelectric crystal is squeezed, there is a relative displacement of positive and negative charges within the crystal and the outer surfaces of the crystal become charged. Hence a potential difference appears across it. An example of such a sensor is the Motorola MPX100AP sensor (Figure 17(b)). This has a built-in vacuum on one side of the diaphragm and so the deflection of the diaphragm gives a measure of the absolute pressure applied to the other side of the diaphragm. The output is a voltage which is proportional to the applied pressure with a sensitivity of 0.6 mV/kPa. Other versions are available which have one side of the diaphragm open to the atmosphere and so can be used to measure gauge pressure; others allow pressures to be applied to both sides of the diaphragm and so can be used to measure differential pressures.

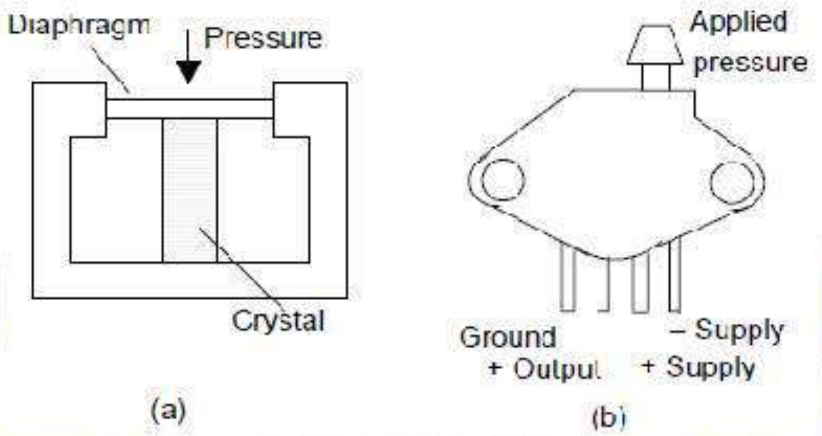


Figure 17. (a) Piezoelectric pressure sensor, (b) MPX100AP

Pressure switches are designed to switch on or off at a particular pressure. A typical form involves a diaphragm or bellows which moves under the action of the pressure and operates a mechanical switch. Figure 18 shows two possible forms. Diaphragms are less sensitive than bellows but can withstand greater pressures.

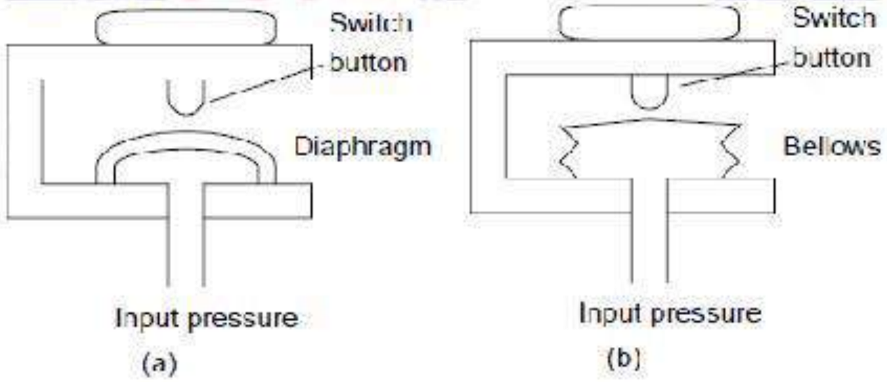


Figure 18. Examples of pressure switches

i) Smart sensors

The term smart sensor is used for a sensor which is integrated with the required buffering and conditioning circuitry in a single element. The circuitry with the element usually consists of data converters, a processor and firmware, and some form of non-volatile EEPROM memory (electrically erasable programmable read only memory, it is similar to EPROM – see Section No. 1). The term non-volatile is used because the memory has to retain certain parameters when the power supply is removed. Because the elements are processor-based devices, such sensors can be programmed for specific requirements. For example, it can be programmed to process the raw input data, correcting for such things as non-linearities, and then send the processed data to a base station. It can be programmed to send a warning signal when the measured parameter reaches some critical value. The IEEE 1451.4 standard interface for smart sensors and actuators is based on an electronic data sheet (TEDS) format which is aimed at allowing installed analogue transducers to be easily connected to digital measurement systems. The standard requires the non-volatile EEPROM embedded memory to hold and communicate data which will allow a plug-and-play capability. It thus would hold data for the identification and properties for the sensor and might also contain the calibration template, so facilitating digital interrogation.

Output devices

The output ports of a PLC are of the relay type or optoisolator with transistor or triac types depending on the devices connected to them which are to be switched on or off. 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.

a) Relay

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 relay, the term contactor being used when large currents are involved. 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 19). The result is that much larger currents can be switched on. Thus the relay might be used to switch on the current to a motor.

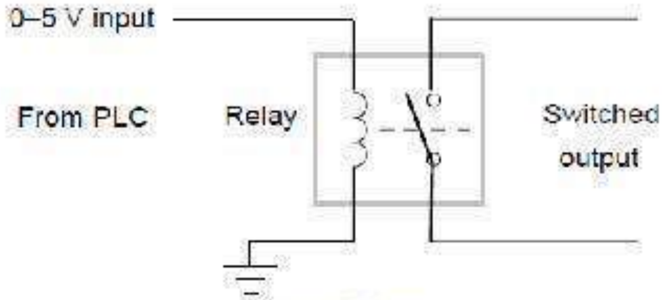
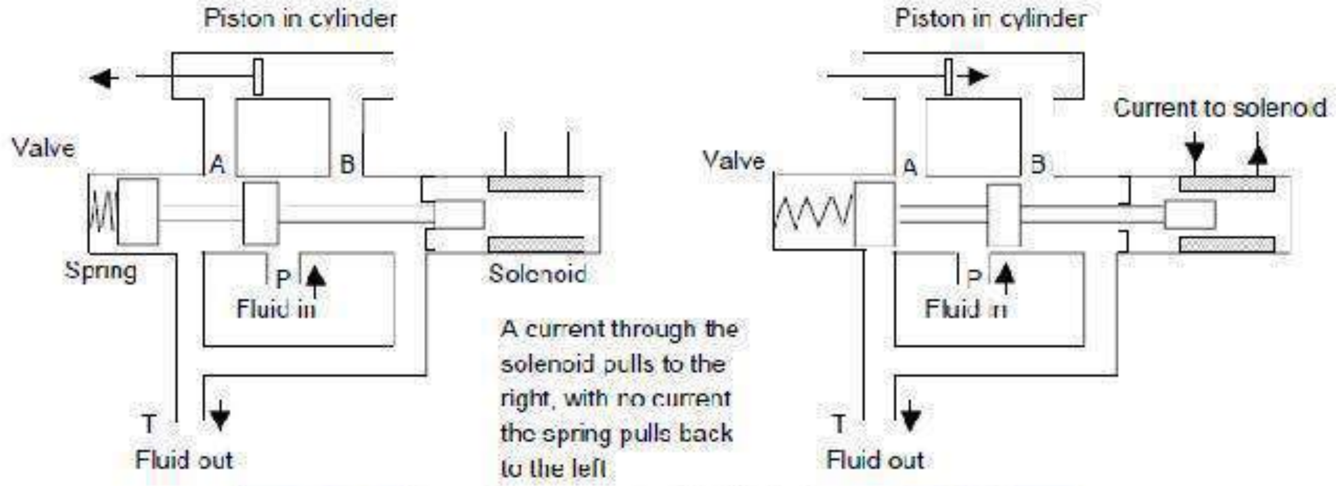


Figure 19. Relay used as an output device

b) *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 pressurised air or oil and so used to operate other devices such as a piston moving in a cylinder. Figure 20 shows one such form, a spool valve, used to control the movement of a piston in a cylinder. Pressurised 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 20(a)) the hydraulic fluid of pressurised 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 pressurised air to the left of the piston and exhausted from the right. The piston then moves to the right. The movement of the piston might be used to push a deflector to deflect items off a conveyor belt or implement some other form of displacement which requires power.

With the above valve there are the two control positions shown in Figure 20(a) and (b). Directional control valves are described by the number of ports they have and the number of control positions. The valve shown in Figure 20 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 20 consists of two squares (Figure 21(a)). 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 21(b) shows this for the valve shown in Figure 20. Figure 22 shows some more examples of direction valves and their switching positions.



A current through the solenoid pulls to the right, with no current the spring pulls back to the left

Figure 20 An example of a solenoid operated valve

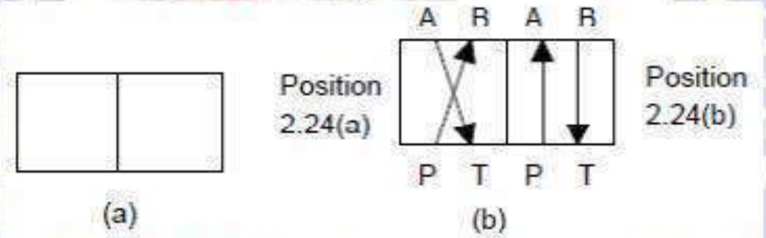
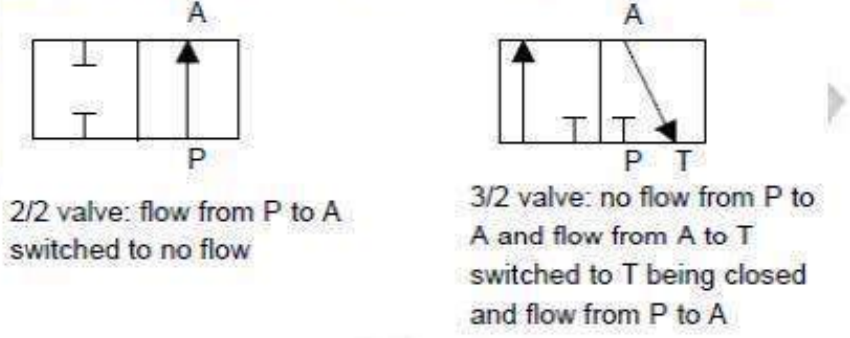


Figure 21 (a) The basic symbol for a two position valve, (b) the 4/2 valve



2/2 valve: flow from P to A switched to no flow

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

Figure 22. Direction valves

In diagrams, the actuation methods used with valves are added to the symbol; Figure 23 shows examples of such symbols. The valve shown in Figure 20 has a spring to give one position and a solenoid to give the other and so the symbol is as shown in Figure 23(d).

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 24(a)) is used for one which is powered by the pressurised 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 24(b)) is used when the cylinder is powered by fluid for its motion in both piston movement directions.

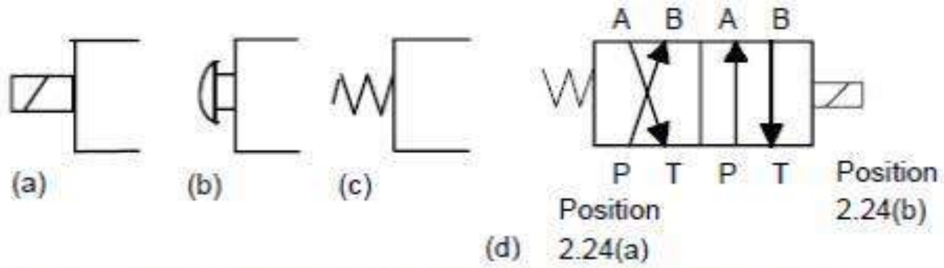


Figure 23. Actuation symbols: (a) solenoid, (b) push button, (c) spring operated, (d) a 4/2 valve

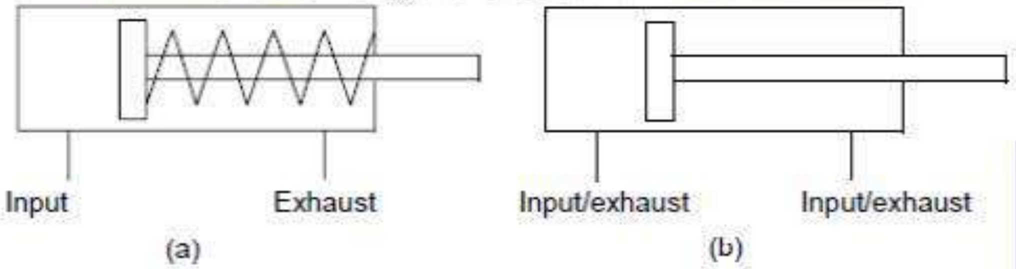


Figure 24. Cylinders: (a) single acting, (b) double acting

c) 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. When a current passes through the armature coil, forces act on the coil and result in rotation. Brushes and a commutator are used to reverse the current through the coil every half rotation and so keep the coil rotating. The speed of rotation can be changed by changing the size of the current to the armature coil. 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 25). The term pulse width modulation (PWM) is used since the width of the voltage pulses is used to control the average d.c. voltage applied to the armature. A PLC might thus control the speed of rotation of a motor by controlling the electronic circuit used to control the width of the voltage pulses.

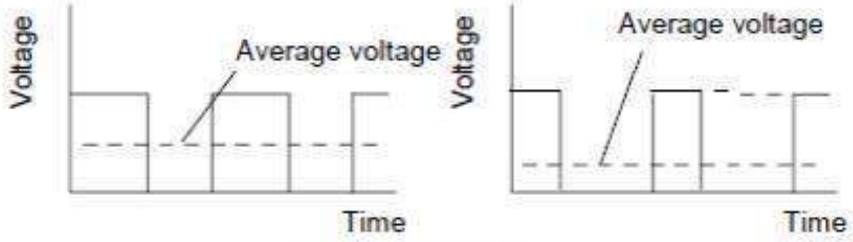


Figure 25. Pulse width modulation

Many industrial processes only require the PLC to switch a d.c. motor on or off. This might be done using a relay. Figure 26(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 to reverse the direction of the current applied to the armature coil. Figure 26(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.

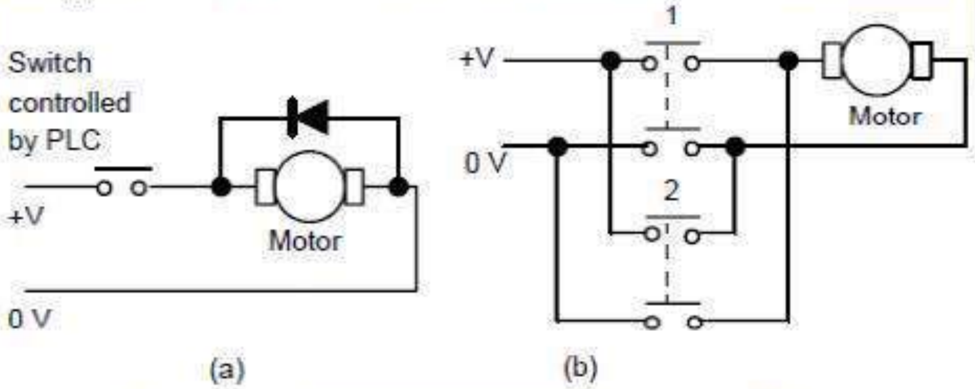


Figure 26. D.c. motor: (a) on–off control, (b) directional control

Another form of d.c. motor is the brushless d.c. motor. This uses a permanent magnet for the magnetic field but, instead of the armature coil rotating as a result of the magnetic field of the magnet, the permanent magnet rotates within the stationary coil. With the conventional d.c. motor, a commutator has to be used to reverse the current through the coil every half rotation in order to keep the coil rotating in the same direction. With the brushless permanent magnet motor, electronic circuitry is used to reverse the current. The motor can be started and stopped by controlling the current to the stationary coil. To reverse the motor, reversing the current is not so easy because of the electronic circuitry used for the commutator function. One method that is used is to incorporate sensors with the motor to detect the position of the north and south poles. These sensors can then cause the current to the coils to be switched at just the right moment to reverse the forces applied to the magnet. The speed of rotation can be controlled using pulse width modulation, i.e. controlling the average value of pulses of a constant d.c. voltage. 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 brushless permanent magnet motors, tend to be more widely used for control purposes.

d) Stepper motors:

The stepper or stepping motor is a motor that produces rotation through equal angles, the so-termed steps, for each digital pulse supplied to its input (Figure 27). Thus, if one input pulse produces a rotation of 1.8o then 20 such pulses would give a rotation of 36.0o. To obtain one complete revolution through 360o, 200 digital pulses would be required. The motor can thus be used for accurate angular positioning. If it is used to drive a continuous belt (Figure 28), it can be used to give accurate linear positioning. Such a motor is used with computer printers, robots, machine tools and a wide range of instruments where accurate positioning is required. There are two basic forms of stepper motor, the permanent magnet type with a permanent magnet rotor and the variable reluctance type with a soft steel rotor. Figure 29 shows the basic elements of the permanent magnet type with two pairs of stator poles.

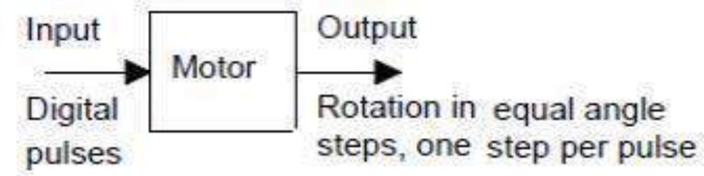


Figure 27. The stepping motor

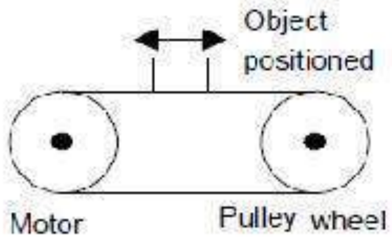
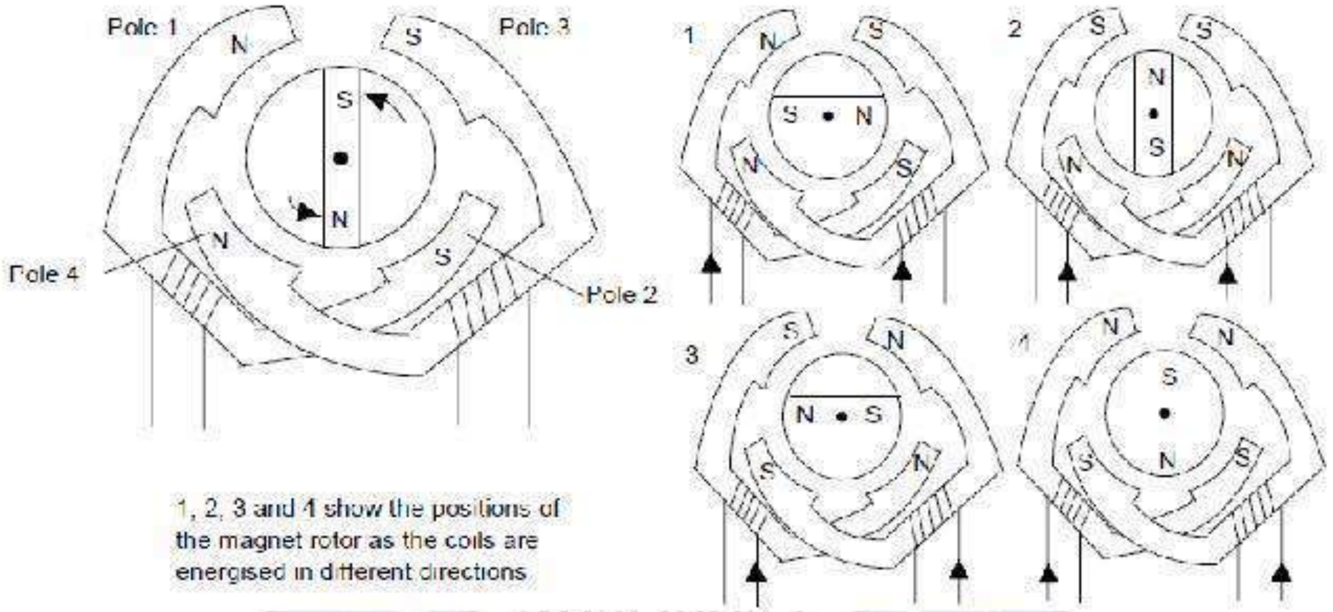


Figure 28. Linear positioning



1, 2, 3 and 4 show the positions of the magnet rotor as the coils are energised in different directions

Figure 29. The basic principles of the permanent magnet stepper motor (2-phase) with 90 o steps.

Each pole is activated by a current being passed through the appropriate field winding, the coils being such that opposite poles are produced on opposite coils. The current is supplied from a d.c. source to the windings through switches. With the currents switched through the coils such that the poles are as shown in Figure 29, the rotor will move to line up with the next pair of poles and stop there. This would be, for Figure 6.35, an angle of 45°. If the current is then switched so that the polarities are reversed, the rotor will move a step to line up with the next pair of poles, at angle 135° and stop there. The polarities associated with each step are:

Step	*Pole 1*	*Pole 2*	*Pole 3*	*Pole 4*
1	*North*	*South*	*South*	*North*
2	*South*	*North*	*South*	*North*
3	*South*	*North*	*North*	*South*
4	*North*	*South*	*North*	*South*
5	*5 Repeat of steps 1 to 4*			

There are thus, in this case, four possible rotor positions: 45°, 135°, 225° and 315°.

Figure 30 shows the basic principle of the variable reluctance type. The rotor is made of soft steel and has a number of teeth, the number being less than the number of poles on the stator. The stator has pairs of poles, each pair of poles being activated and made into an electromagnet by a current being passed through the coils wrapped round them. When one pair of poles is activated, a magnetic field is produced which attracts the nearest pair of rotor teeth so that the teeth and poles line up. This is termed the position of minimum reluctance. By then switching the current to the next pair of poles, the rotor can be made to rotate to line up with those poles. Thus by sequentially switching the current from one pair of poles to the next, the rotor can be made to rotate in steps.

There is another version of the stepper motor and that is a hybrid stepper. This combines features of both the permanent magnet and variable reluctance motors. They have a permanent magnet rotor encased in iron caps which are cut to have teeth. The rotor sets itself in the minimum reluctance position in response to a pair of stator coils being energised.

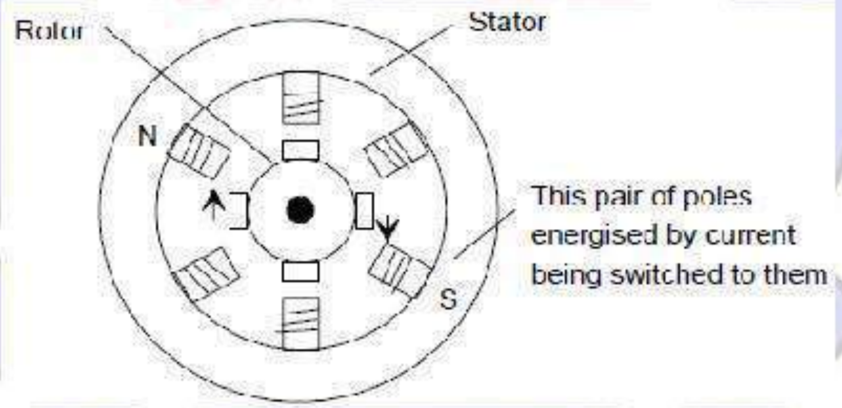


Figure 30. The principle of the variable reluctance stepper motor

There is another version of the stepper motor and that is a *hybrid stepper*. This combines features of both the permanent magnet and variable reluctance motors. They have a permanent magnet rotor encased in iron caps which are cut to have teeth. The rotor sets itself in the minimum reluctance position in response to a pair of stator coils being energised. To drive a stepper motor, so that it proceeds step-by-step to provide rotation, requires each pair of stator coils to be switched on and off in the required sequence when the input is a sequence of pulses (Figure 31). Driver circuits are available to give the correct sequencing and Figure 32 shows an example, the SAA 1027 for a four-phase unipolar stepper. Motors are termed unipolar if they are wired so that the current can only flow in one direction through any particular motor terminal, bipolar if the current can flow in either direction through any particular motor terminal. The stepper motor will rotate through one step each time the trigger input goes from low to high. The motor runs clockwise when the rotation input is low and anticlockwise when high. When the set pin is made low the output resets. In a control system, these input pulses might be supplied by a microprocessor.

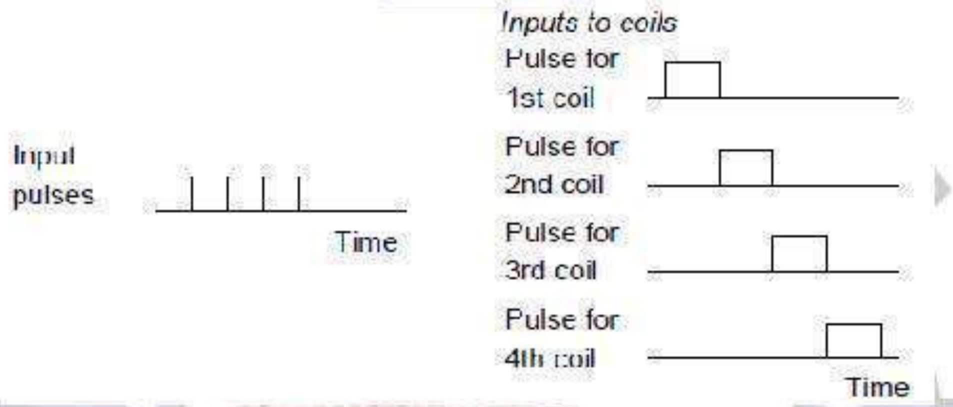


Figure 31. Input and outputs of the drive system

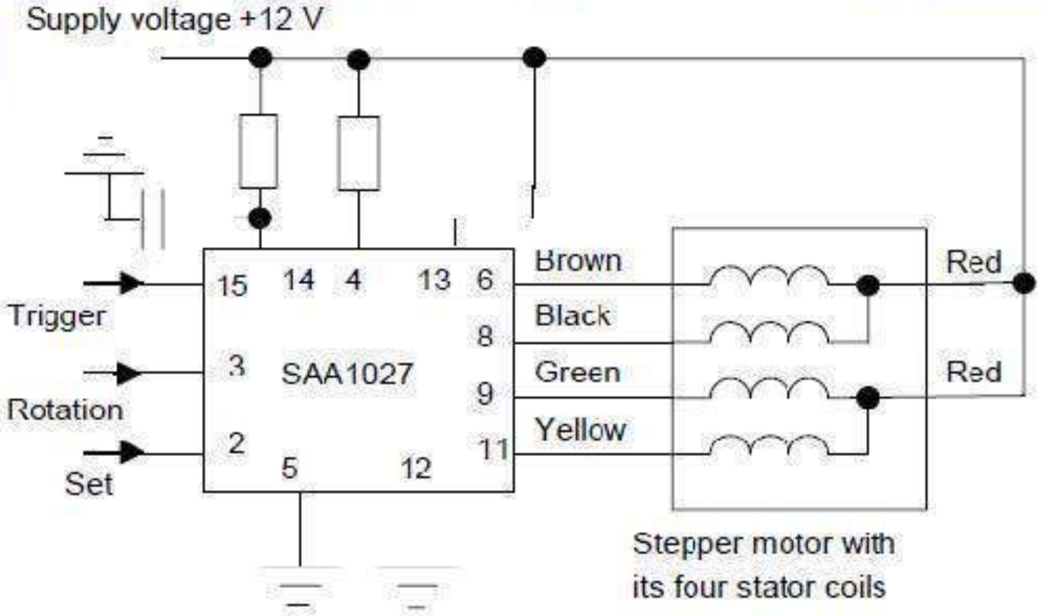


Figure 32. Driver circuit connections with the integrated circuit SAA1027

Examples of applications

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

1) A conveyor belt

Consider a conveyor belt that is to be used to transport goods from a loading machine to a packaging area (Figure 33). When an item is loaded onto the conveyor belt, a contact switch might be used to indicate that the item is on the belt and start the conveyor motor. The motor then has to keep running until the item reaches the far end of the conveyor and falls off into the packaging area. When it does this, a switch might be activated which has the effect of switching off the conveyor motor. The motor is then to remain off until the next item is loaded onto the belt. Thus the inputs to a PLC controlling the conveyor are from two switches and the output is to a motor.

Figure 33. Conveyor.

2) A lift

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

3) *A robot control system:*

Figure 34 shows how directional control valves can be used for a control system of a robot. When there is an input to solenoid A of valve 1, the piston moves to the right and causes the gripper to close. If solenoid B is energised, with A de-energised, the piston moves to the left and the gripper opens. When both solenoids are de-energised, no air passes to either side of the piston in the cylinder and the piston keeps its position without change. Likewise, inputs to the solenoids of valve 2 are used to extend or retract the arm. Inputs to the solenoids of valve 3 are used to move the arm up or down. Inputs to the solenoids of valve 4 are used to rotate the base in either a clockwise or anticlockwise direction.

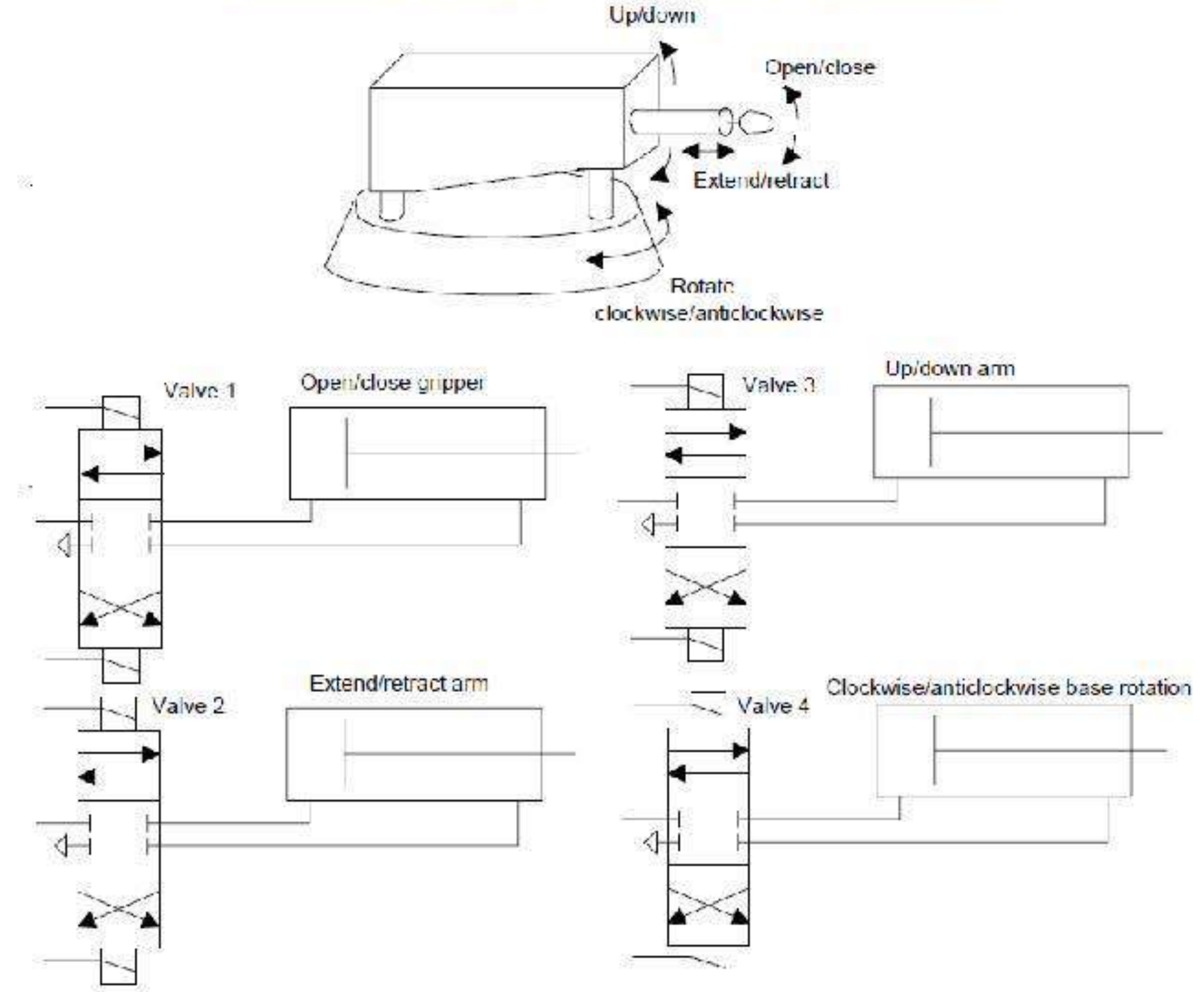


Figure 34. Robot controls.

4) *Liquid level monitoring:*

Figure 35 shows a method that could be used to give an on−off signal when the liquid in a container reaches a critical level. A magnetic float, a ring circling the sensor probe, falls as the liquid level falls and opens a reed switch when the critical level is reached. The reed switch is in series with a 39 ✡ resistor so that this is switched in parallel with a 1 kΩ resistor by the action of the reed switch. Opening the reed switch thus increases the resistance from about 37 Ω to 1 kΩ. Such a resistance change can be transformed by signal conditioning to give suitable on–off signals.

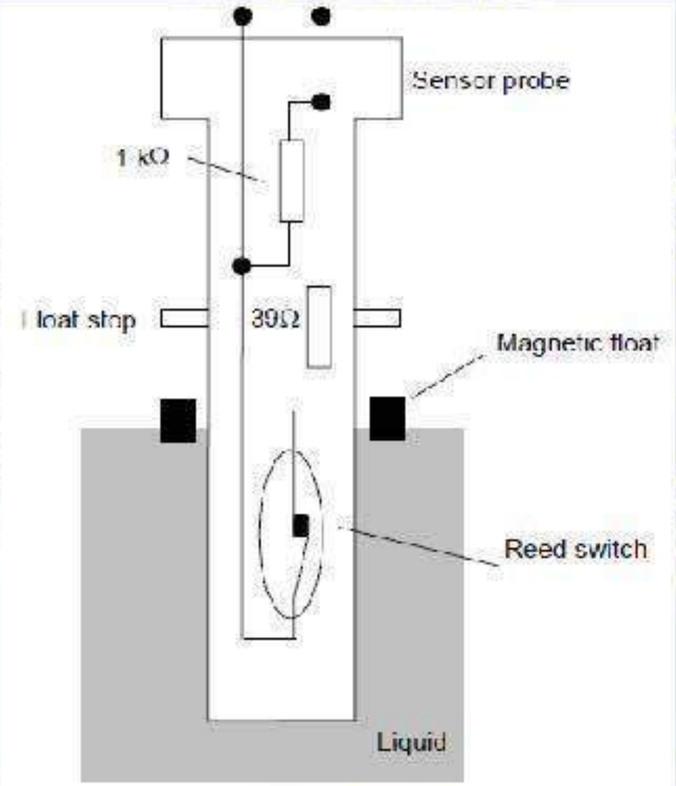


Figure 35. Liquid level monitoring.

Section Three

I/O processing

- **Input/output units**

  Input units ; Output units ;

- **Signal conditioning**
- **Remote connections**

  Serial and parallel communications ; Serial standards ;

  Parallel standards ; Protocols ; ASCII codes

- **Networks**

  Distributed systems ; Network standards ;

  Examples of commercial systems

- **Processing inputs**

  I/O addresses


2018 – 2019

Input / Output Processing

This section continues the discussion of inputs and outputs from section two and is a brief consideration of the processing of the signals from input and output devices. The input/output (I/O) unit provides the interface between the PLC controller and the outside world and must therefore provide the necessary signal conditioning to get the signal to the required level and also to isolate it from possible electrical hazards such as high voltages. This chapter includes the forms of typical input/output modules and, in an installation where sensors are some distance from the PLC processing, their communication links to the PLC.

Input/output units

Input signals from sensors and the outputs required for actuating devices can be:

1. Analogue, i.e. a signal whose size is related to the size of the quantity being sensed.
2. Discrete, i.e. essentially just an on–off signal.
3. Digital, i.e. a sequence of pulses.

The CPU, however, must have an input of digital signals of a particular size, normally 0 to 5 V. The output from the CPU is digital, normally 0 to 5 V. Thus there is generally a need to manipulate input and output signals so that they are in the required form. The input/output (I/O) units of PLCs are designed so that a range of input signals can be changed into 5 V digital signals and so that a range of outputs are available to drive external devices. It is this in-built facility to enable a range of inputs and outputs to be handled which makes PLCs so easy to use. The following is a brief indication of the basic circuits used for input and output units. In the case of rack instruments they are mounted on cards which can be plugged into the racks and so the input/output characteristics of the PLC can thus be changed by changing the cards. A single box form of PLC has input/output units incorporated by the manufacturer.

A. Input units

The terms sourcing and sinking refer to the manner in which d.c. devices are interfaced with the PLC (see Section 1). For a PLC input unit, with sourcing it is the source of the current supply for the input device connected to it (Figure 1(a)). With sinking, the input device provides the current to the input unit (Figure 1(b)).

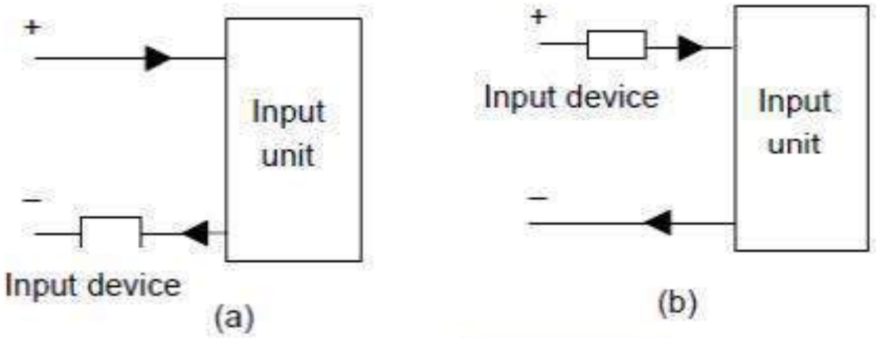


Figure 1. Input unit: (a) sourcing, (b) sinking

Figures 2 and 3 show the basic input unit circuits for discrete and digital d.c. and discrete a.c. inputs. Optoisolators (see Section 1,2) are used to provide protection. With the a.c. input unit, a rectifier bridge network is used to rectify the a.c. so that the resulting d.c. signal can provide the signal for use by the optoisolator to give the input signals to the CPU of the PLC. Individual status lights are provided for each input to indicate when the input device is providing a signal.

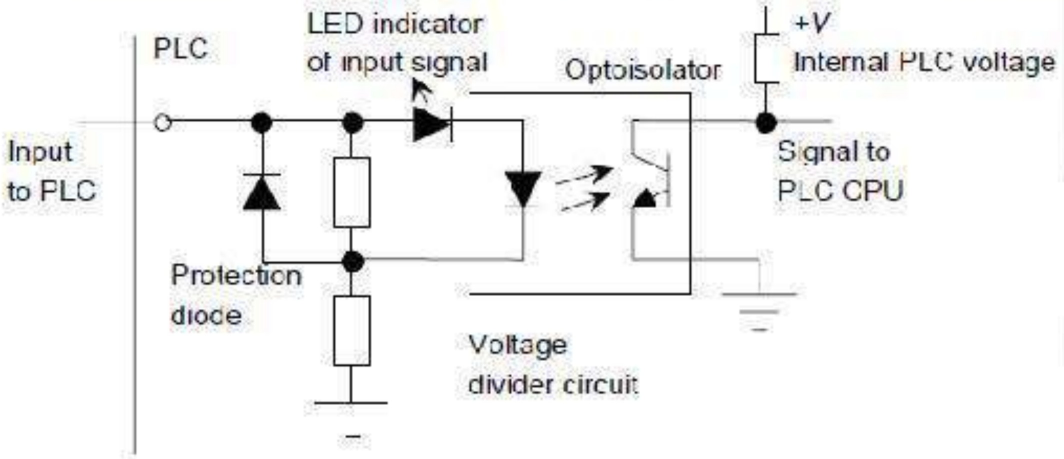


Figure 2. D.C. input unit.

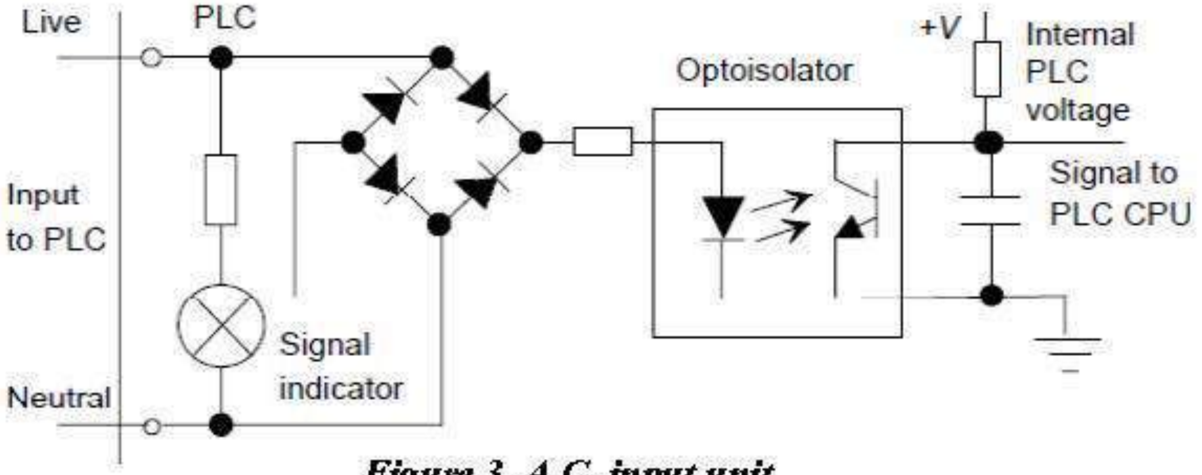


Figure 3. A.C. input unit.

Analogue signals can be inputted to a PLC if the input channel is able to convert the signal to a digital signal using an analogue-to-digital converter. With a rack mounted system this may be achieved by mounting a suitable analogue input card in the rack. So that one analogue card is not required for each analogue input, multiplexing is generally used (Figure 4). This involves more than one analogue input being connected to the card and then electronic switches used to select each input in turn. Cards are typically available giving 4, 8 or 16 analogue inputs.

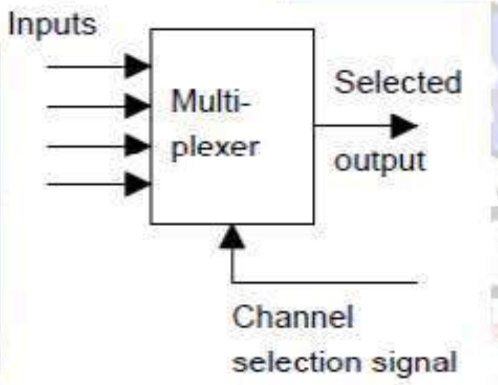


Figure 4. Multiplexer

Figure 5(a) illustrates the function of an analogue-to-digital converter (ADC). A single analogue input signal gives rise to on−off output signals along perhaps eight separate wires. The eight signals then constitute the so-termed digital word corresponding to the analogue input signal level. With such an 8-bit converter there are 28 = 256 different digital values possible; these are 0000 0000 to 1111 1111, i.e. 0 to 255. The digital output goes up in steps (Figure 5(b)) and the analogue voltages required to produce each digital output are termed *quantization levels*.

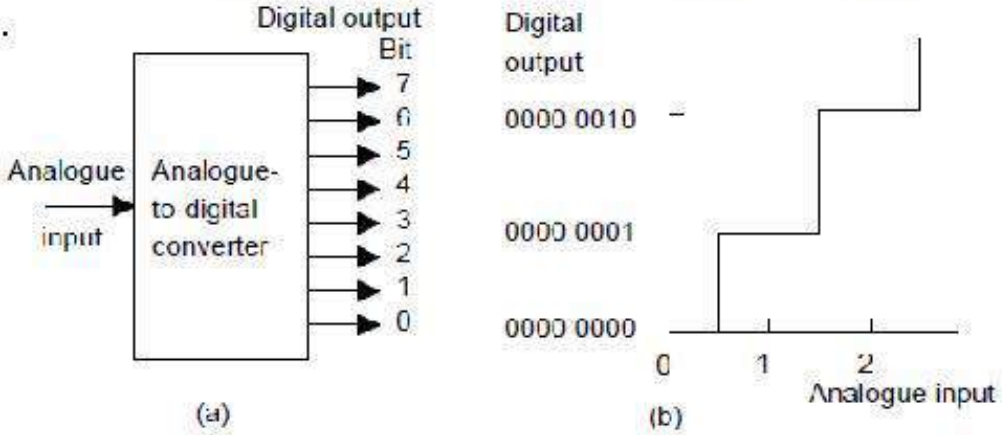


Figure 5. (a) Function of an analogue-to-digital converter, (b) an illustration of the relationship between the analogue input and the digital output.

The analogue voltage has to change by the difference in analogue voltage between successive levels if the binary output is to change. The term resolution is used for the smallest change in analogue voltage which will give rise to a change in one bit in the digital output. With an 8-bit ADC, if, say, the full-scale analogue input signal varies between 0 and 10 V then a step of one digital bit involves an analogue input change of 10/255 V or about 0.04 V. This means that a change of 0.03 V in the input will produce no change in the digital output. The number of bits in the output from an analogue-to-digital converter thus determines the resolution, and hence accuracy, possible. If a 10-bit ADC is used then 2^{10} = 1024 different digital values are possible and, for the full-scale an alogue input of 0 to 10 V, a step of one digital bit involves an analogue input change of 10/1023 V or about 0.01 V. If a 12-bit ADC is used then 2^{12} = 4096 different digital values are possible and, for the full-scale analogue input of 0 to 10 V, a step of one digital bit involves an analogue input change of 10/4095 V or about 2.4 mV. In general, the resolution of an n-bit ADC is $1/(2^n - 1)$.

The following illustrates the analogue-to-digital conversion for an 8-bit converter when the analogue input is in the range 0 to 10 V:

Analogue input (V)	Digital output (V)
0.00	0000 0000
0.04	0000 0001
0.08	0000 0010
0.12	0000 0011
0.16	0000 0100
0.20	0000 0101
0.24	0000 0110
0.28	0000 0111
0.32	0000 1000
etc	

To illustrate the above, consider a thermocouple used as a sensor with a PLC and giving an output of 0.5 mV per oC. What will be the accuracy with which the PLC will activate the output device if the thermocouple is connected to an analogue input with a range of 0 to 10 V d.c and using a 10-bit analogue-to-digital converter? With a 10-bit converter there is 2^{10} = 1024 bits covering the 0 to 10 V range. Thus a change of 1 bit corresponds to 10/1023 V or about 0.01 V, i.e. 10 mV. Hence the accuracy with which the PLC recognises the input from the thermocouple is ±5 mV or ±10°C.

B. Output units

With a PLC output unit, when it provides the current for the output device (Figure 6(a)) it is said to be sourcing and when the output device provides the current to the output unit it is said to be sinking (Figure 6(b)). Quite often, sinking input units are used for interfacing with electronic equipment and sourcing output units for interfacing with solenoids. Output units can be relay, transistor or triac. Figure 7 shows the basic form of a relay output unit, Figure 8 that of a transistor output unit and Figure 9 that of a triac output unit.

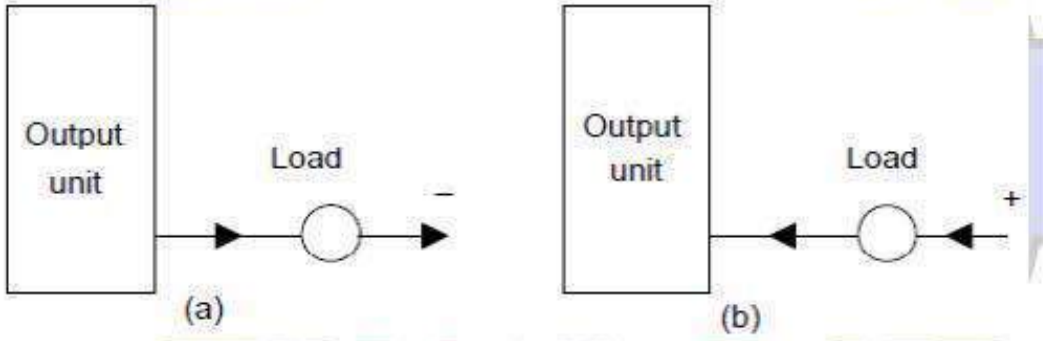


Figure 6. Output unit: (a) sourcing, (b) sinking

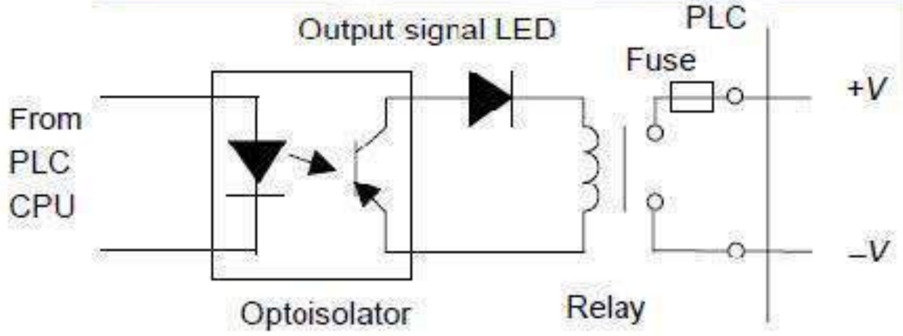


Figure 7. Relay output unit

Analogue outputs are frequently required and can be provided by digital-to-analogue converters (DACs) at the output channel. The input to the converter is a sequence of bits with each bit along a parallel line. Figure 10 shows the basic function of the converter.

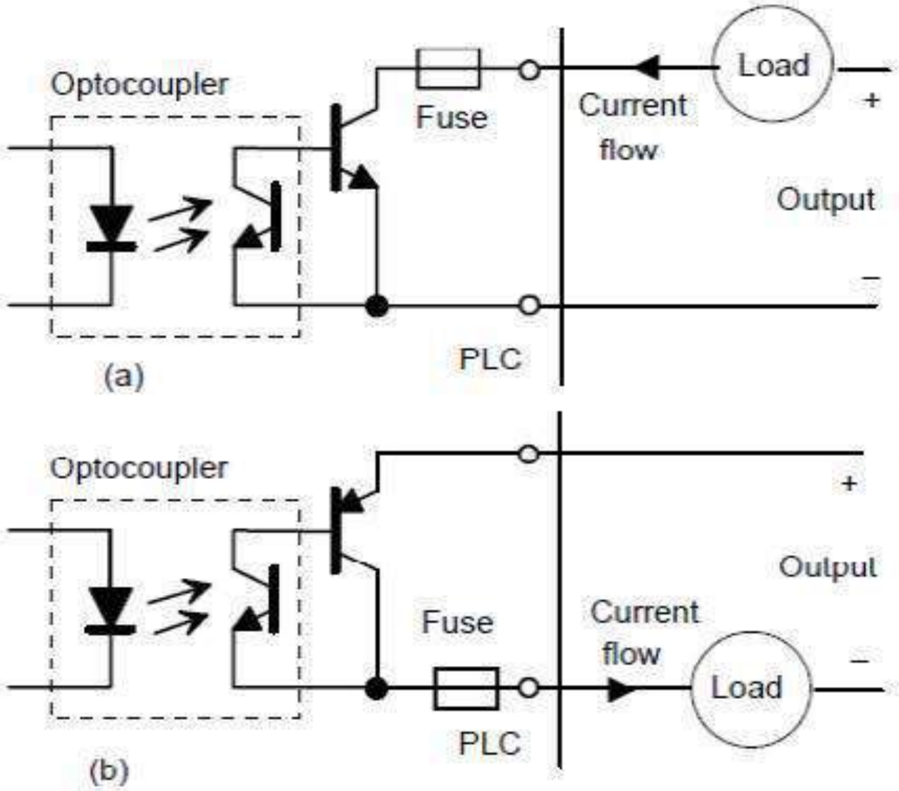


Figure 8. Basic forms of transistor output: (a) current sinking, (b) current sourcing

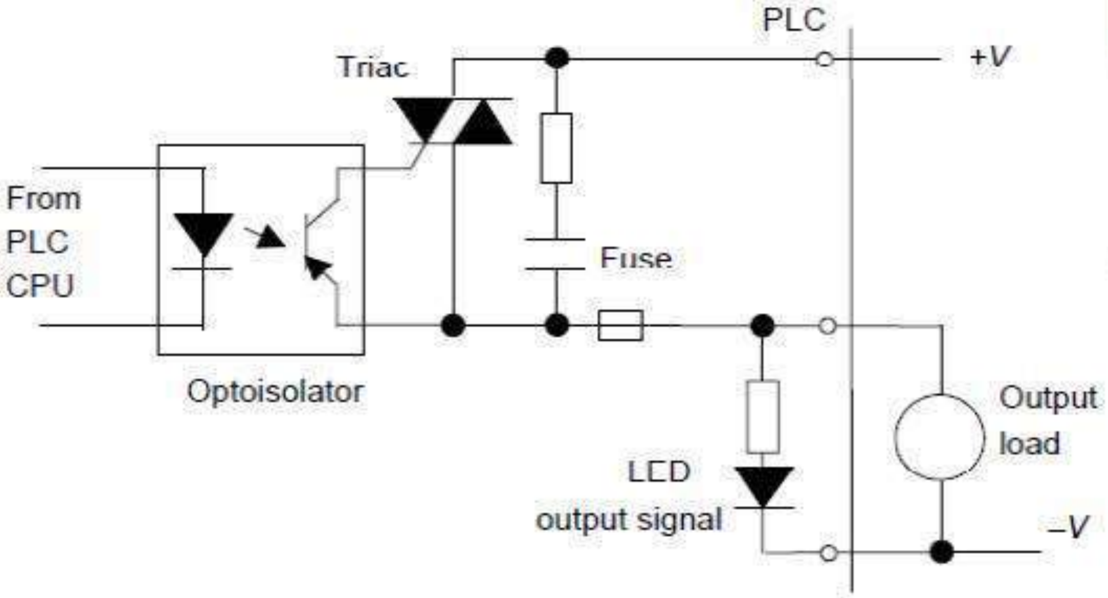


Figure 9. Triac output unit

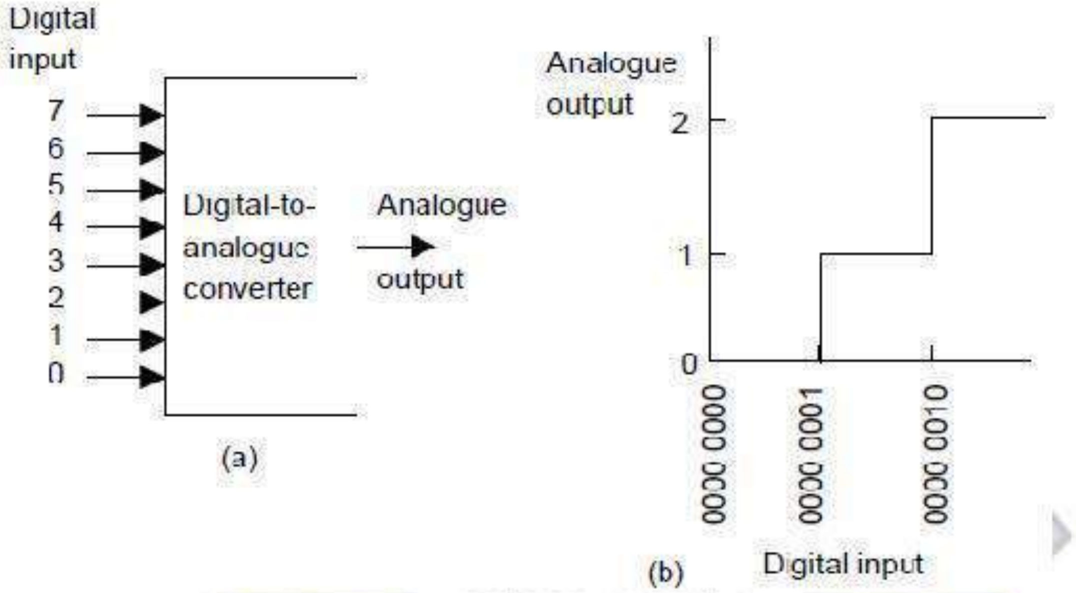


Figure 10. (a) DAC function, (b) digital-to-analogue conversion

A bit in the 0 line gives rise to a certain size output pulse. A bit in the 1 line gives rise to an output pulse of twice the size of the 0 line pulse. A bit in the 2 line gives rise to an output pulse of twice the size of the 1 line pulse. A bit in the 3 line gives rise to an output pulse of twice the size of the 2 line pulse, and so on. All the outputs add together to give the analogue version of the digital input. When the digital input changes, the analogue output changes in a stepped manner, the voltage changing by the voltage changes associated with each bit. For example, if we have an 8-bit converter then the output is made up of voltage values of 2^8 = 256 analogue steps. Suppose the output range is set to 10 V d.c. One bit then gives a change of 10/255 V or about 0.04 V. Thus we have:

Analogue input (V)	Digital output (V)
0000 0000	0.00
0000 0001	0.04
0000 0010	0.08+0.0=0.08
0000 0011	0.0.8+0.04=0.12
0000 0100	0.16
0000 0101	0.16+0.0+0.04=0.20
0000 0110	0.16+0.08=0.24
0000 0111	0.16+0.08+0.04=0.0.28
0000 1000	0.32
	etc

Analogue output modules are usually provided in a number of outputs, e.g. 4 to 20 mA, 0 to +5 V d.c., 0 to +10 V d.c., and the appropriate output is selected by switches on the module. Modules generally have outputs in two forms, one for which all the outputs from that module have a common voltage supply and one which drives outputs having their own individual voltage supplies. Figure 11 shows the basic principles of these two forms of output.

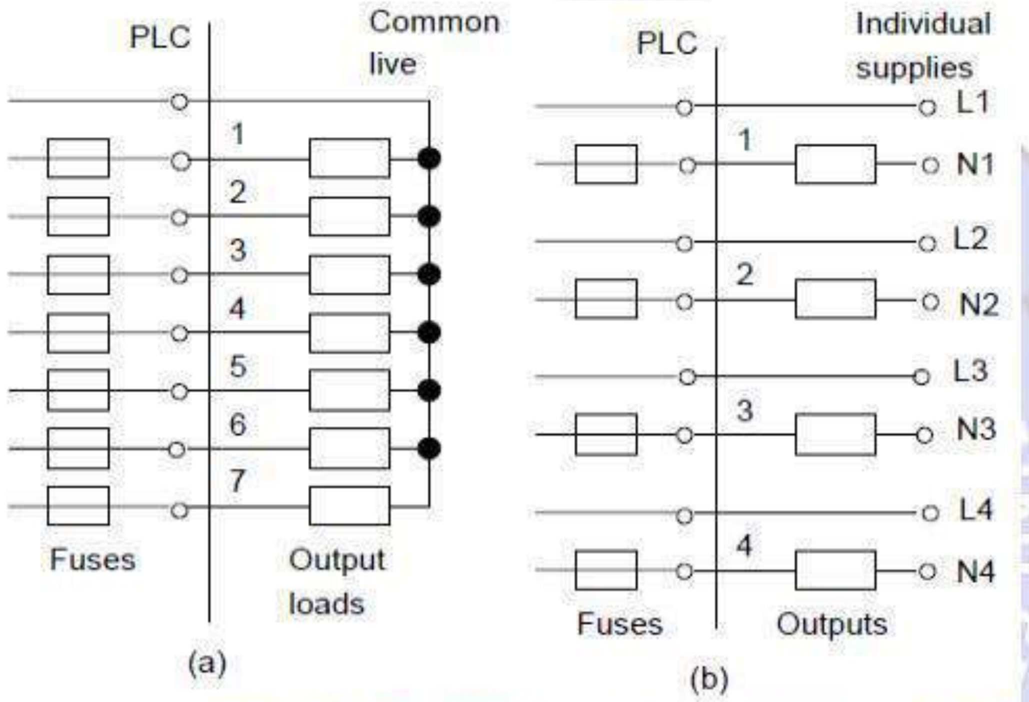


Figure 11. Forms of output: (a) common supply, (b) individual supplies.

Signal Conditioning

When connecting sensors which generate digital or discrete signals to an input unit, care has to be taken to ensure that voltage levels match. However, many sensors generate analogue signals. In order to avoid having a multiplicity of analogue input channels to cope with the wide diversity of analogue signals that can be generated by sensors, external signal conditioning is often used to bring analogue signals to a common range and so allow a standard form of analogue input channel to be used. A common standard that is used (Figure 12.a) is to convert analogue signals to a current in the range 4 to 20 mA and thus to a voltage by passing it through a 250 Ω resistance to give a 1 to 5 V input signal. Thus, for example, a sensor used to monitor liquid level in the height range 0 to 1 m would have the 0 level represented by 4 mA and the 1 m represented by 20 mA. The use of 4 mA to represent the low end of the analogue range serves the purpose of distinguishing between when the sensor is indicating zero and when the sensor is not working and giving zero response for that reason. When this happens the current would be 0 mA. The 4 mA also is often a suitable current to operate a sensor and so eliminate the need for a separate power supply.

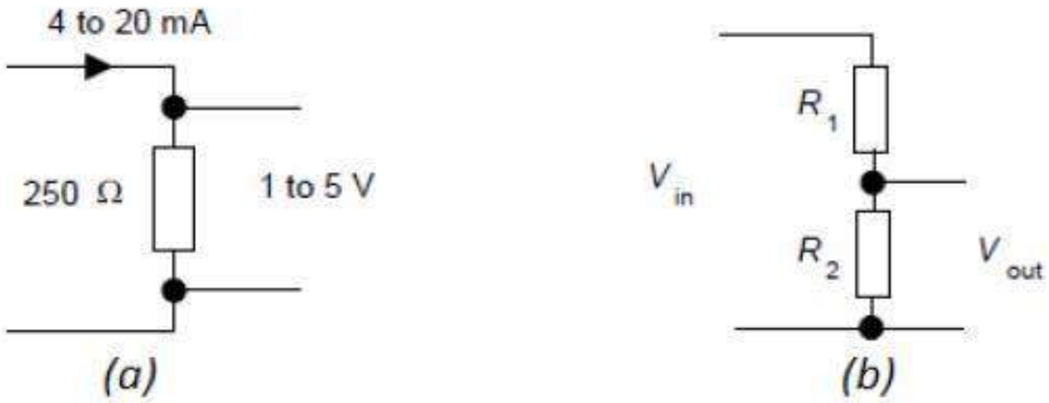


Figure 12. (a) Standard analogue signal. (b) Potential divider

A potential divider (Figure 12.b) can be used to reduce a voltage from a sensor to the required level; the output voltage level V_{out} is:

$$V_{out} = \frac{R_2}{R_1 + R_2} V_{in}$$

Amplifiers can be used to increase the voltage level; Figure 13 shows the basic form of the circuits that might be used with a 741 operational amplifier with Figure 13(a) being an inverting amplifier and Figure 13(b) a non-inverting amplifier. With the inverting amplifier the output V_{out} is:

$V_{\text{out}} = -\frac{R_2}{R_1} V_{\text{in}}$, and with the non-inverting amplifier: $V_{\text{out}} = \frac{R_1 + R_2}{R_1} V_{\text{in}}$

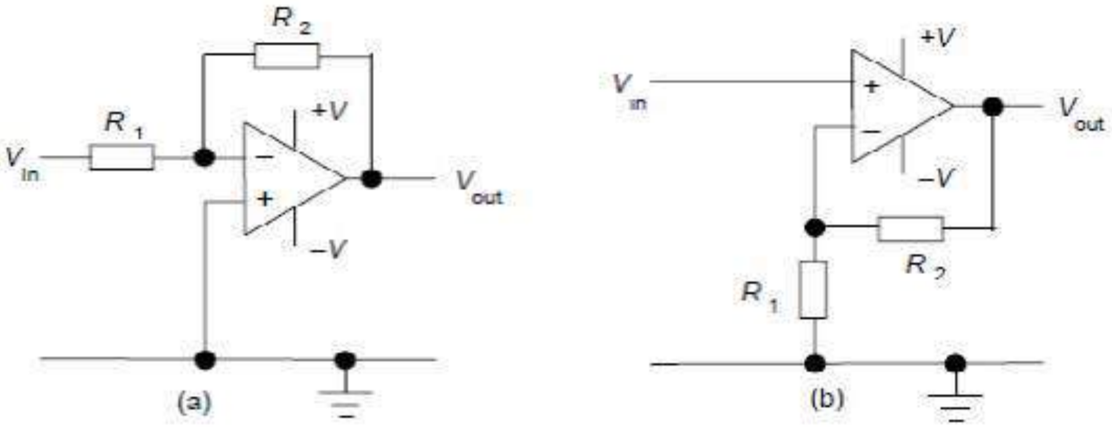
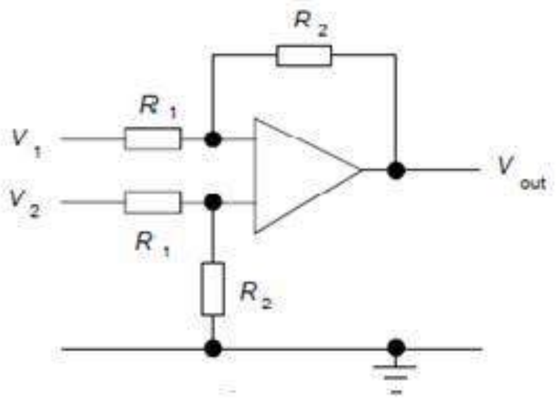


Figure 13. Operational amplifier circuits

Often a differential amplifier is needed to amplify the difference between two input voltages. Such is the case when a sensor, e.g. a strain gauge, is connected in a Wheatstone bridge and the output is the difference between two voltages or a thermocouple where the voltage difference between the hot and cold junctions is required. Figure 14 shows the basic form of an operational amplifier circuit for this purpose. The output voltage V_{out} is: $V_{\text{out}} = \frac{R_2}{R_1}(V_2 - V$

As an illustration of the use of signal conditioning, Figure 15 shows the arrangement that might be used for a strain gauge sensor. The sensor is connected in a Wheatstone bridge and the out-of-balance potential difference amplified by a differential amplifier before being fed via an analogue-to-digital converter unit which is part of the analogue input port of the PLC.



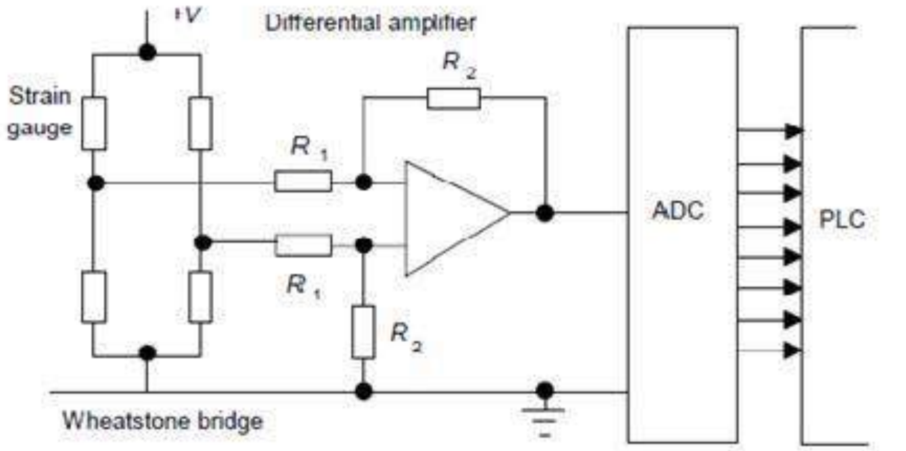


Figure 14. Differential amp.   *Figure 15. Signal conditioning with a strain gauge sensor.*

Remote Connections

When there are many inputs or outputs located considerable distances away from the PLC, while it would be possible to run cables from each such device to the PLC a more economic solution is to use input/output modules in the vicinity of the inputs and outputs and use just a single core cable to connect each, over the long distances, to the PLC instead of the multicore cable that would be needed without such distant I/O modules (Figure 16).

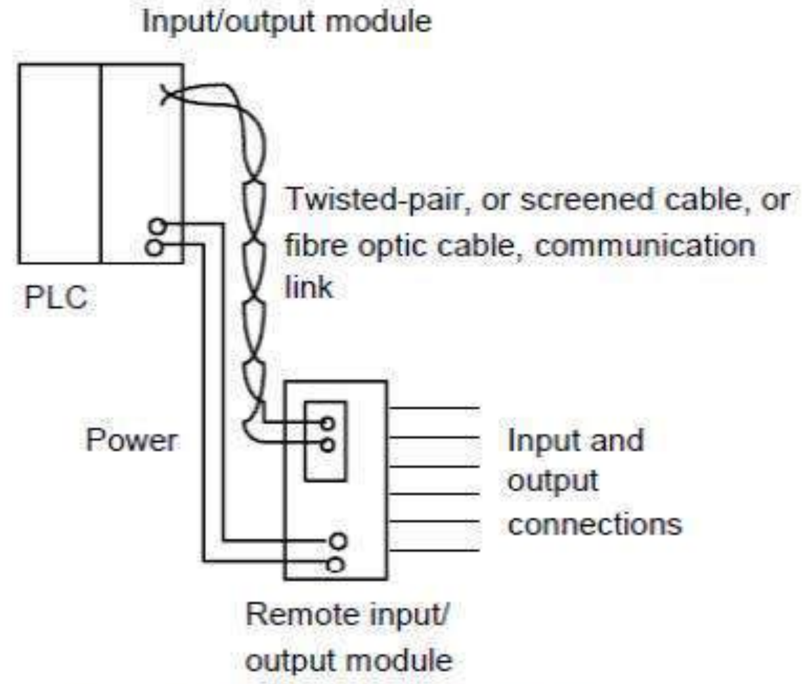


Figure 16. Use of remote input/output module

In some situations a number of PLCs may be linked together with a master PLC unit sending and receiving input/output data from the other units (Figure 17). The distant PLCs do not contain the control program since all the control processing is carried out by the master PLC. The cables used for communicating data between remote input/output modules and a central PLC, remote PLCs and the master PLC are typically twisted-pair cabling, often routed hrough grounded steel conduit in order to reduce electrical 'noise'. Coaxial cable enables higher data rates to be transmitted and does not require the shielding of steel conduit. Fibre-optic cabling has the advantage of resistance to noise, small size and flexibility and is now becoming more widely used.

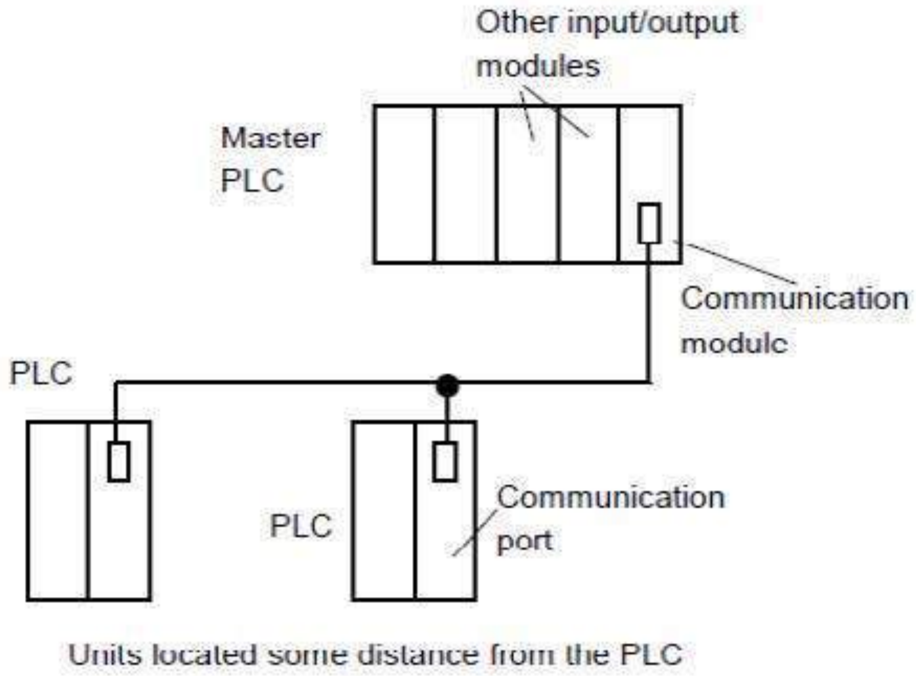


Figure 17. Use of remote input/output PLC systems

A. Serial and parallel communications

Serial communication is when data is transmitted one bit at a time (Figure 18(a)). Thus if an 8-bit word is to be transmitted, the eight bits are transmitted one at a time in sequence along a cable. This means that a data word has to be separated into its constituent bits for transmission and then reassembled into the word when received. Parallel communication is when all the constituent bits of a word are simultaneously transmitted along parallel cables (Figure 18(b)). This allows data to be transmitted over short distances at high speeds.

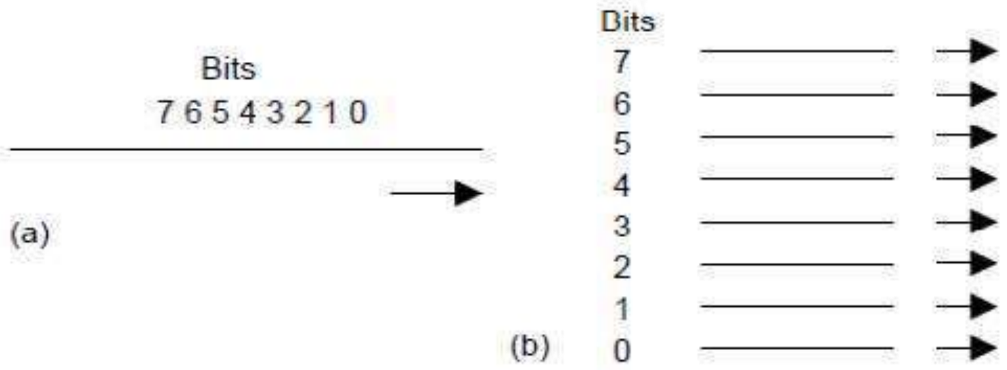


Figure 18. (a) Serial communication, (b) parallel communication

Serial communication is used for transmitting data over long distances. It is much cheaper to run, for serial communication, a single core cable over a long distance than the multicore cables that would be needed for parallel communication. With a PLC system, serial communication might be used for the connection between a computer, when used as a programming terminal, and a PLC. Parallel communication might be used when connecting laboratory instruments to the system. However, internally, PLCs work, for speed, with parallel communications. Thus, circuits called UARTS (universal asynchronous receivers–transmitters) have to be used at input/output ports to converts serial communications signals to parallel.

B. Serial standards

For successful serial communications to occur, it is necessary to specify:

1) The voltage levels to be used for signals, i.e. what signal represents a 0 and what represents a 1.
2) What the bit patterns being transmitted mean and how the message is built up. Bear in mind that a sequence of words are being sent along the same cable and it is necessary to be able to be able to determine when one word starts and finishes and the next word starts.
3) The speed at which the bit pattern is to be sent, i.e. the number of bits per second.
4) Synchronisation of the clocks at each end. This is necessary if, for example, a particular duration transmitted pulse it to be recognised by the receiver as just a single bit rather than two bits.
5) Protocols, or flow controls, to enable such information as 'able to receive data' or 'not ready to receive data' to be received. This is commonly done by using two extra signal wires (termed handshake wires), one to tell the receiver that the transmitter is ready to send the
6) Data and the other to tell the transmitter that the receiver is ready to receive data.
7) Error-checking to enable a bit pattern to be checked to determine if corruption of the data has occurred during transmission.

The most common standard serial communications interface used is the RS232. Connections are made via 25-pin D-type connectors (Figure 19) with usually, though not always, a male plug on cables and a female socket on the equipment. Not all the pins are used in every application.

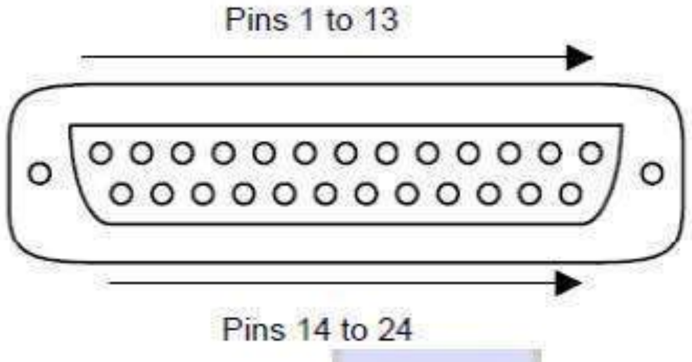


Figure 19. D connector

The minimum requirements are:

- ***Pin 1***: Ground connection to the frame of chassis
- ***Pin 2***: Serial transmitted data (output data pin)
- ***Pin 3***: Serial received data (input data pin)
- ***Pin 7***: Signal ground which acts as a common signal return path

A configuration that is widely used with interfaces involving computers is:

- ***Pin 1***: Ground connection to the frame of chassis
- ***Pin 2***: Serial transmitted data (output data pin)
- ***Pin 3***: Serial received data (input data pin)
- ***Pin 4***: Request to send
- ***Pin 5***: Clear to send
- ***Pin 6***: Data set ready
- ***Pin 7***: Signal ground which acts as a common signal return path
- ***Pin 20***: Data terminal ready

The signals sent through pins 4, 5, 6 and 20 are used to check that the receiving end is ready to receive a signal, the transmitting end is ready to send and the data is ready to be sent. With RS232, a 1 bit is represented by a voltage between −5 and −25 V, normally −12 V, and a 0 by a voltage between +5 and +25 V, normally +12 V.

The term baud rate is used to describe the transmission rate, it being approximately the number of bits transmitted or received per second. However, not all the bits transmitted can be used for data, some have to be used to indicate the start and stop of a serial piece of data, these often being termed flags, and as a check as to whether the data has been corrupted during transmission. Figure 20 shows the type of signal that might be sent with RS232. The parity bit is added to check whether corruption has occurred, with even parity a 1 being added to make the number of 1s an even number. To send seven bits of data, eleven bits may be required.

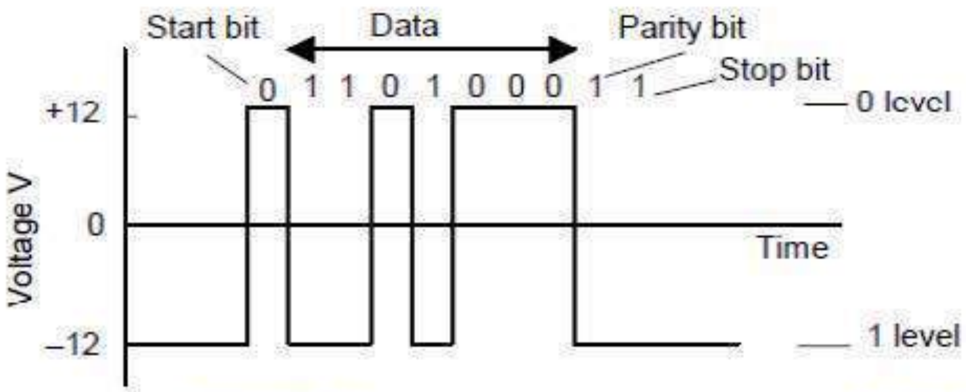


Figure 20. RS232 signal levels

C. Parallel standards

The standard interface most commonly used for parallel communications is IEEE-488. This was originally developed by Hewlett Packard to link their computers and instruments and was known as the Hewlett Packard Instrumentation Bus. It is now often termed the General Purpose Instrument Bus. This bus provides a means of making interconnections so that parallel data communications can take place between listeners, talkers and controllers. Listeners are devices that accept data from the bus, talkers place data, on request, on the bus and controllers manage the flow of data on the bus and provide processing facilities. There is a total of 24 lines, of which eight bi-directional lines are used to carry data and commands between the various devices connected to the bus, five lines are used for control and status signals, three are used for handshaking between devices and eight are ground return lines (Figure 21).

Commands from the controller are signalled by taking the Attention Line (ATN) low, otherwise it is high, and thus indicating that the data lines contain data. The commands can be directed to individual devices by placing addresses on the data lines. Each device on the bus has its own address. Device addresses are sent via the data lines as a parallel 7-bit word, the lowest 5-bits providing the device address and the other two bits control information. If both these bits are 0 then the commands are sent to all addresses, if bit 6 is 1 and bit 7 a 0 the addressed device is switched to be a listener, if bit 6 is 0 and bit 7 is 1 then the device is switched to be a talker. As illustrated above by the function of the ATN line, the management lines each have an individual task in the control of information. The handshake lines are used for controlling the transfer of data. The three lines ensure that the talker will only talk when it is being listened to by listeners. Table 1 lists the functions of all the lines and their pin numbers in a 25-way D-type connector.

Table 1 *IEEE-488 bus system*

Pin	Signal group	Abbreviation	Signal/function
1	Data	D101	Data line 1.
2	Data	D102	Data line 2.
3	Data	D103	Data line 3.
4	Data	D104	Data line 4.
5	Management	EOI	End Or Identify. This is used to either signify the end of a message sequence from a talker device or is used by the controller to ask a device to identify itself.
6	Handshake	DAV	Data valid. When the level is low on this line then the information on the data bus is valid and acceptable.
7	Handshake	NRFD	Not Ready For Data. This line is used by listener devices taking it high to indicate that they are ready to accept data.
8	Handshake	NDAC	Not Data Accepted. This line is used by listeners taking it high to indicate that data is being accepted.
9	Management	IFC	Interface Clear. This is used by the controller to reset all the devices of the system to the start state.
10	Management	SRQ	Service Request. This is used by devices to signal to the controller that they need attention.
11	Management	ATN	Attention. This is used by the controller to signal that it is placing a command on the data lines.
12		SHIELD	Shield.
13	Data	D105	Data line 5.
14	Data	D106	Data line 6.
15	Data	D107	Data line 7.
16	Data	D108	Data line 8.
17	Management	REN	Remote Enable. This enables a device on the bus to indicate that it is to be selected for remote control rather than by its own control panel.
18		GND	Ground/common.
19		GND	Ground/common.
20		GND	Ground/common.
21		GND	Ground/common.
22		GND	Ground/common.
23		GND	Ground/common.
24		GND	Ground/common.

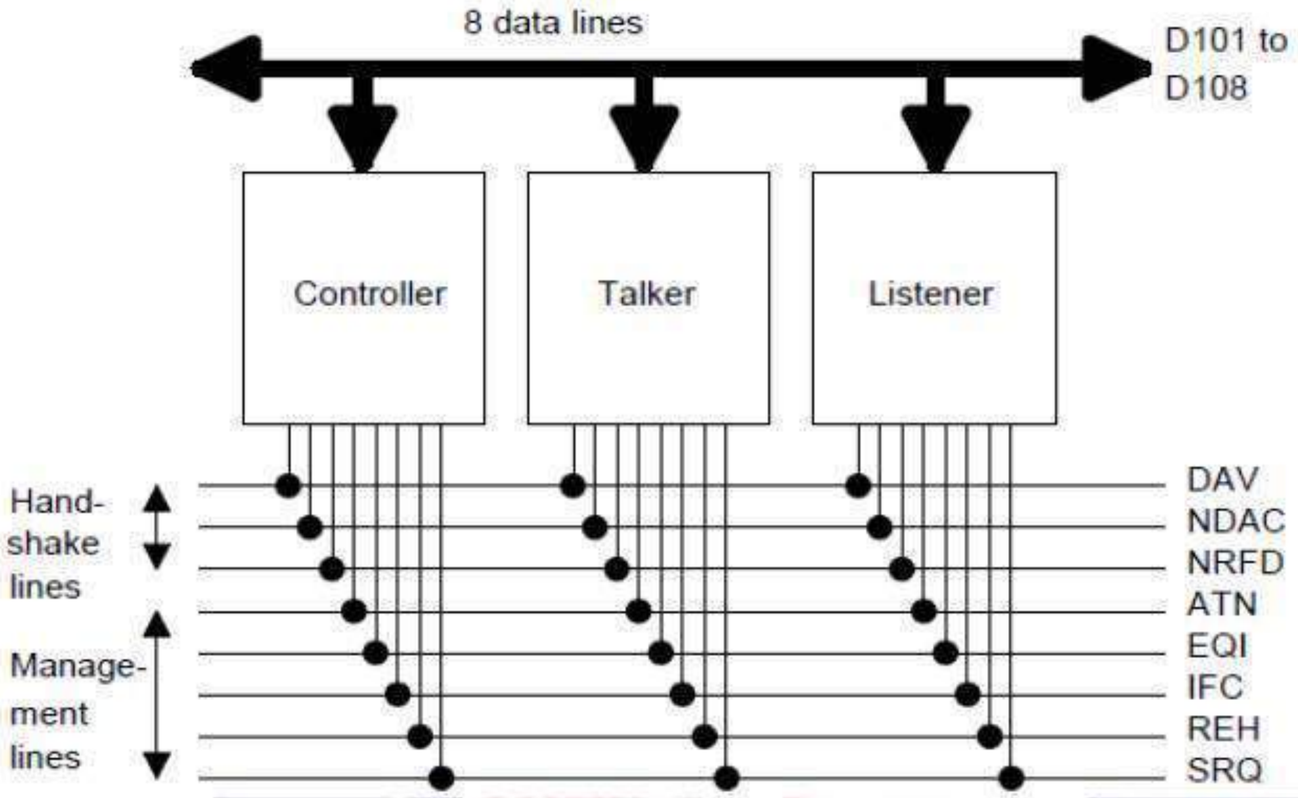


Figure 21. The IEEE-488 bus structure

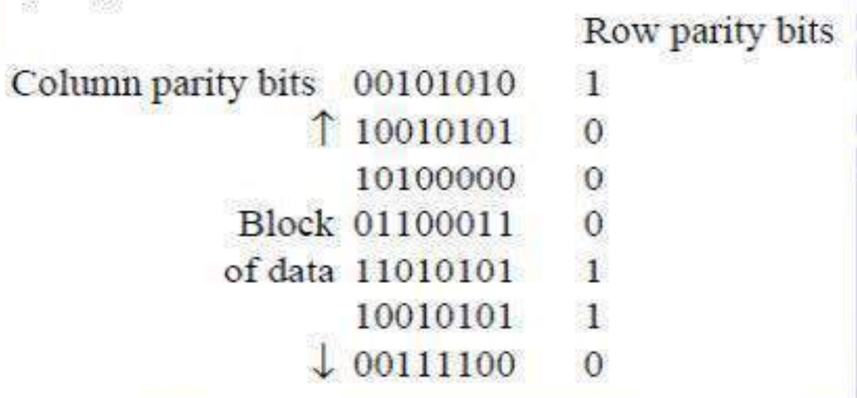
D. Protocols

It is necessary to exercise control of the flow of data between two devices so what constitutes the message, and how the communication is to be initiated and terminated, is defined. This is termed the protocol. Thus one device needs to indicate to the other to start or stop sending data. This can be done by using handshaking wires connecting transmitting and receiving devices so that a signal along one such wire can tell the receiver that the transmitter is ready to send (RTS) and along another wire that the transmitter is ready to receive, a clear to send signal (CTS). RTS and CTS lines are provided for in RS232 serial communication links.

An alternative is to use additional characters on the transmitting wires. With the ENQ/ACK protocol, data packets are sent to a receiver with a query character ENQ. When this character is received the end of the data packet has been reached. Once the receiver has processed that data, it can indicate it is ready for another block of data by sending back an acknowledge (ACK) signal. Another form, the XON/XOFF, has the receiving device sending a XOFF signal to the sending device when it wishes the data flow to cease. The transmitter then waits for an XON signal before resuming transmission. One form of checking for errors in the message that might occur as a result of transmission is the parity check. This is an extra bit added to a message to ensure that the number of bits in a piece of data is always odd or always even. For example, 0100100 is even since there is an even number of 1s and 0110100 is odd since there is an odd number of 1s. To make both these odd parity then the extra bit added at the end in the first case is 1 and in the second case 0, i.e. we have 01001001 and 01101000. Thus when a message is sent out with odd bit parity, if the receiver finds that the bits give an even sum, then the message has been corrupted during transmission and the receiver can request that the message be repeated.

The parity bit method can detect if there is an error resulting from a single 0 changing to a 1 or a 1 changing to a 0 but cannot detect two such errors occurring since there is then no change in parity. To check on such occurrences more elaborate checking methods have to be used. One method involves storing data words in an array of rows and columns.

Parity can then be checked for each row and each column. The following illustrates this for seven words using even parity.



		Row parity bits
Column parity bits	00101010	1
↑	10010101	0
	10100000	0
Block	01100011	0
of data	11010101	1
	10010101	1
↓	00111100	0

Another method, termed cyclic redundancy check codes, involves splitting the message into blocks. Each block is then treated as a binary number and is divided by a predetermined number. The remainder from this division is then sent as the error checking number on the conclusion of the message and enables a check on the accuracy of the message to be undertaken.

E. ASCII codes

The most widely used code for the transmission of characters is the ASCII code (American Standard Code for Information Interchange). This is a seven-bit code giving 128 different combinations of bits covering lower case and upper case alphanumeric characters, punctuation and 32 control codes. As an illustration, Table 2 shows the codes used for capital letters. Examples of control codes are SOH, for start of heading, i.e. the first character of a heading of an information message, as 000 0001; STX, for start of text, as 000 0010; ETX, for end of text, as 000 0011; EOT, for end of transmission, as 000 0011.

Table 2 *Examples of ASCII codes*

ASCII		ASCII		ASCII	
A	100 0001	N	100 1110	0	011 0000
B	100 0010	O	100 1111	1	011 0001
C	100 0011	P	101 0000	2	011 0010
D	100 0100	Q	101 0001	3	011 0011
E	100 0101	R	101 0010	4	011 0100
F	100 0110	S	101 0011	5	011 0101
G	100 0111	T	101 0100	6	011 0110
H	100 1000	U	101 0101	7	011 0111
I	100 1001	V	101 0110	8	011 1000
J	100 1010	W	101 0111	9	011 1001
K	100 1011	X	101 1000		
L	100 1100	Y	101 1001		
M	100 1101	Z	101 1010		

Networks

The increasing use of automation in industry has led to the need for communications and control on a plant-wide basis with programmable controllers, computers, robots, and CNC machines interconnected. The term local area network (LAN) is used to describe a communications network designed to link computers and their peripherals within the same building or site.

Networks can take three basic forms. With the star form (Figure 22(a)) the terminals are each directly linked to a central computer, termed the host, or master with the terminals being termed slaves. The host contains the memory, processing and switching equipment to enable the terminals to communicate. Access to the terminals is by the host asking each terminal in turn whether it wants to talk or listen. With the bus or single highway type of network (Figure 22(b)), each of the terminals is linked into a single cable and so each transmitter/receiver has a direct path to each other transmitter/receiver in the network.

Methods, i.e. protocols, have to be adopted to ensure that no more than one terminal talks at once, otherwise confusion can occur. A terminal has to be able to detect whether another terminal is talking before it starts to talk. With the ring network (Figure 22(c)), a continuous cable, in the form of a ring, links all the terminals. Again methods have to be employed to enable communications from different terminals without messages becoming mixed up. The single highway and the ring methods are often termed peer to peer in that each terminal has equal status. Such a system allows many stations to use the same network.

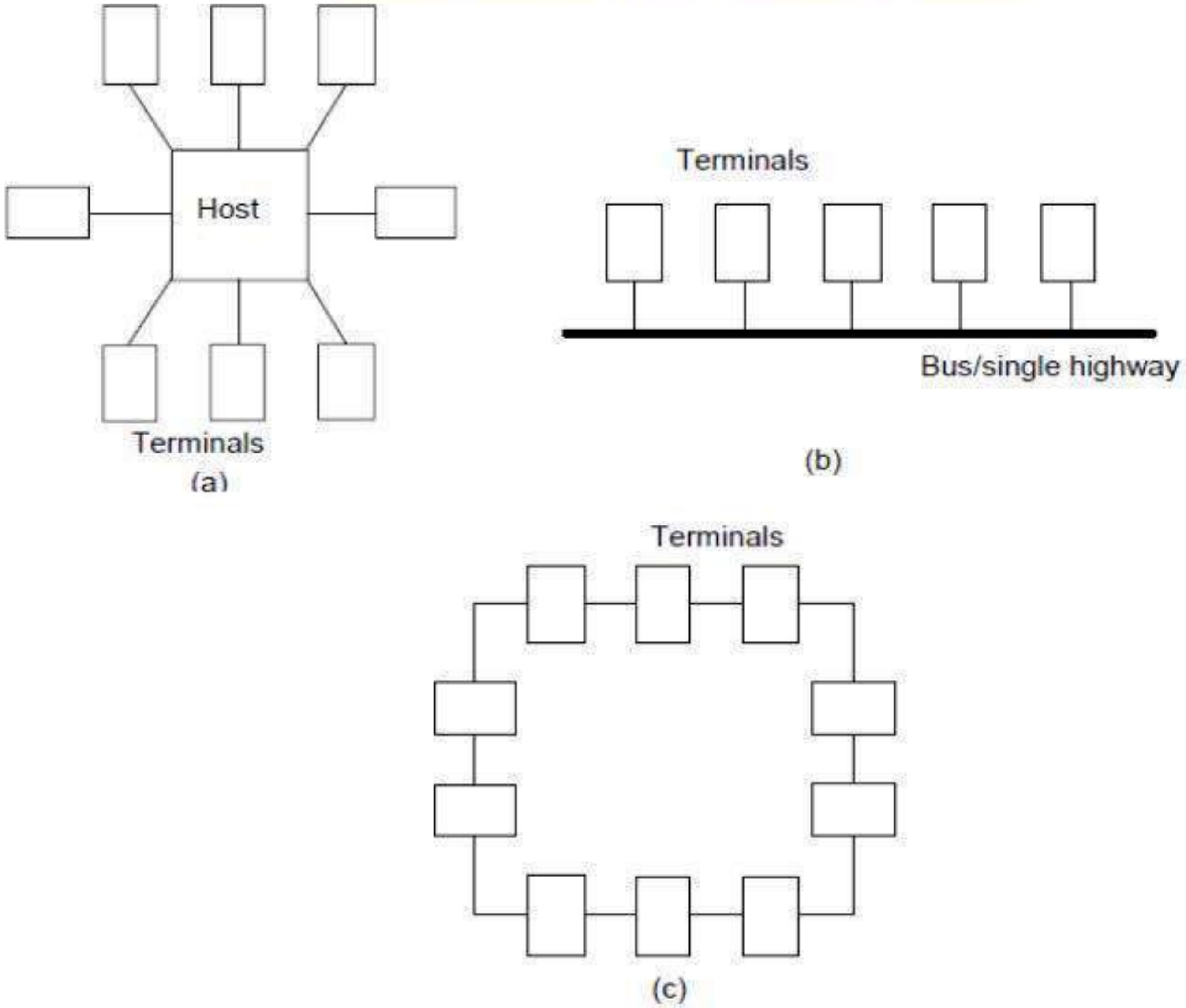


Figure 22. Networks: (a) star, (b) bus/single highway, (c) ring

Processing inputs

A PLC is continuously running through its program and updating it as a result of the input signals. Each such loop is termed a cycle. PLCs could be operated by each input being examined as it occurred in the program and its effect on the program determined and the output correspondingly changed. This mode of operation is termed continuous updating. Because, with continuous updating, there is time spent interrogating each input in turn, the time taken to examine several hundred input/output points can become comparatively long. To allow a more rapid execution of a program, a specific area of RAM is used as a buffer store between the control logic and the input/output unit. Each input/output has an address in this memory. At the start of each program cycle the CPU scans all the inputs and copies their status into the input/output addresses in RAM. As the program is executed the stored input data is read, as required, from RAM and the logic operations carried out. The resulting output signals are stored in the reserved input/output section of RAM. At the end of each program cycle all the outputs are transferred from RAM to the appropriate output channels. The outputs then retain their status until the next updating. This method of operation is termed mass I/O copying. The sequence can be summarised as (Figure 23):

1) Scan all the inputs and copy into RAM.
2) Fetch and decode and execute all program instructions in sequence, copying output instructions to RAM.
3) Update all outputs.
4) Repeat the sequence.

The time taken to complete a cycle of scanning inputs and updating outputs according to the program instructions, i.e. the cycle time, though relatively quick, is not instantaneous and means that the inputs are not watched all the time but samples of their states taken periodically. A typical cycle time is of the order of 10 to 50 ms. This means that the inputs and outputs are updated every 10 to 50 ms and thus there can be a delay of this order in the system reacting. It also means that if a very brief input cycle appears at the wrong moment in the cycle, it could be missed. In general, any input must be present for longer than the cycle time. Special modules are available for use in such circumstances. Consider a PLC with a cycle time of 40 ms. What is the maximum frequency of digital impulses that can be detected? The maximum frequency will be if one pulse occurs every 40 ms, i.e. a frequency of 1/0.04 = 25 Hz.

The cycle or scanning time for a PLC, i.e. its response speed, is determined by:

1) The CPU used.
2) The size of the program to be scanned.
3) The number of input/outputs to be read.
4) The system functions that are in use, the greater the number the slower the scanning time.

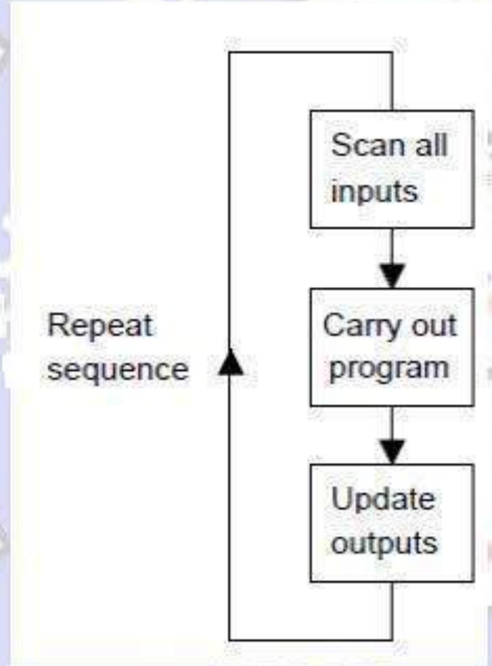


Figure 23. PLC operation

I/O addresses

The PLC has to be able to identify each particular input and output. It does this by allocating addresses to each input and output. With a small PLC this is likely to be just a number, prefixed by a letter to indicate whether it is an input or an output. Thus for the Mitsubishi PLC we might have inputs with addresses X400, X401, X402, etc., and outputs with addresses Y430, Y431, Y432, etc., the X indicating an input and the Y an output. Toshiba use a similar system. With larger PLCs having several racks of input and output channels, the racks are numbered. With the Allen-Bradley PLC-5, the rack containing the processor is given the number 0 and the addresses of the other racks are numbered 1, 2, 3, etc. according to how set-up switches are set. Each rack can have a number of modules and each one deals with a number of inputs and/or outputs. Thus addresses can be of the form shown in Figure 24 (a). For example, we might have an input with address I:012/03. This would indicate an input, rack 01, module 2 and terminal 03. With the Siemens SIMATIC S5, the inputs and outputs are arranged in groups of 8. Each 8 group is termed a byte and each input or output within an 8 is termed a bit. The inputs and outputs thus have their addresses in terms of the byte and bit numbers, effectively giving a module number followed by a terminal number, a full stop (.) separating the two numbers. Figure 24 (b) shows the system. Thus I0.1 is an input at bit 1 in byte 0, Q2.0 is an output at bit 0 in byte 2.

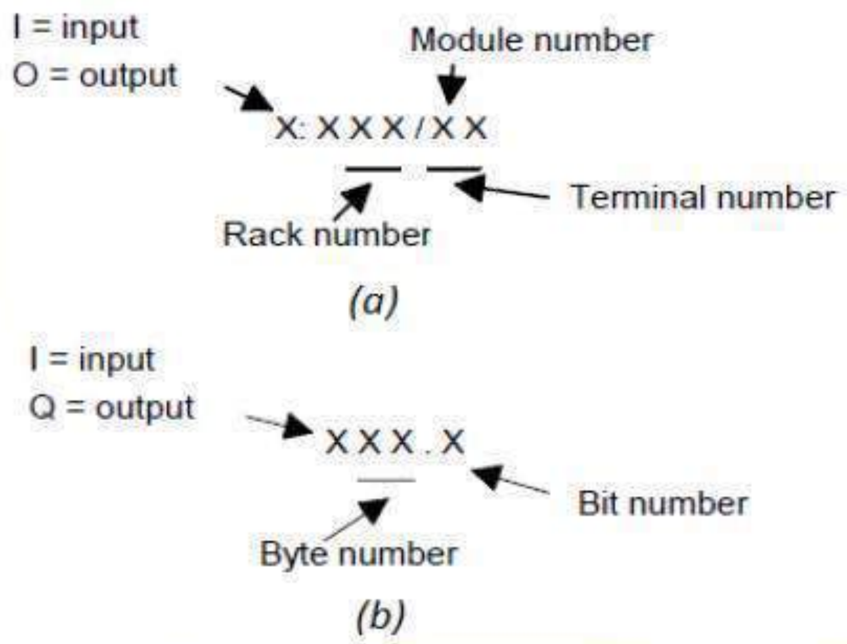


Figure 24 (a) Allen-Bradley PLC-5 addressing. (b) Siemens SIMATIC S5 addressing.

The GEM-80 PLC assigns inputs and output addresses in terms of the module number and terminal number within that module. The letter A is used to designate inputs and B outputs. Thus A3.02 is an input at terminal 02 in module 3, B5.12 is an output at terminal 12 in module 5. In addition to using addresses to identify inputs and outputs, PLCs also use their addressing systems to identify internal, software-created devices, such as relays, timers and counters.

Section Four

Ladder and functional block programming

- Introduction
- Ladder diagrams
  PLC ladder programming ;
- Logic functions
  AND ; OR ; NOT ; NAND ; NOR ;
  Exclusive OR (XOR) ;
- Latching
- Multiple outputs
- Entering programs
  Ladder symbols
- Function blocks
  Logic gates; Boolean algebra
- Program examples

Introduction

Programs for microprocessor-based systems have to be loaded into them in machine code, this being a sequence of binary code numbers to represent the program instructions. However, assembly language based on the use of mnemonics can be used, e.g. LD is used to indicate the operation required to load the data that follows the LD, and a computer program called an assembler is used to translate the mnemonics into machine code. Programming can be made even easier by the use of the so-called high level languages, e.g. C, BASIC, PASCAL, FORTRAN, COBOL. These use pre-packaged functions, represented by simple words or symbols descriptive of the function concerned. For example, with C language the symbol & is used for the logic AND operation. However, the use of these methods to write programs requires some skill in programming and PLCs are intended to be used by engineers without any great knowledge of programming. As a consequence, ladder programming was developed. This is a means of writing programs which can then be converted into machine code by some software for use by the PLC microprocessor.

This method of writing programs became adopted by most PLC manufacturers, however each tended to have developed their own versions and so an international standard has been adopted for ladder programming and indeed all the methods used for programming PLCs. The standard, published in 1993, is IEC 1131-3 (International Electrotechnical Commission). The IEC 1131-3 programming languages are ladder diagrams (LAD), instruction list (IL), sequential function charts (SFC), structured text (ST), and function block diagrams (FBD).

This section is an introduction to the programming of a PLC using ladder diagrams and functional block diagrams, with discussion of the other techniques in the next chapter. Here we are concerned with the basic techniques involved in developing ladder and function block programs to represent basic switching operations, involving the logic functions of AND, OR, Exclusive OR, NAND and NOR, and latching. Later chapters continue with further ladder programming involving other elements.

Ladder diagrams

As an introduction to ladder diagrams, consider the simple wiring diagram for an electrical circuit in Figure 1(a). The diagram shows the circuit for switching on or off an electric motor. We can redraw this diagram in a different way, using two vertical lines to represent the input power rails and stringing the rest of the circuit between them. Figure 1(b) shows the result. Both circuits have the switch in series with the motor and supplied with electrical power when the switch is closed. The circuit shown in Figure 1(b) is termed a ladder diagram.

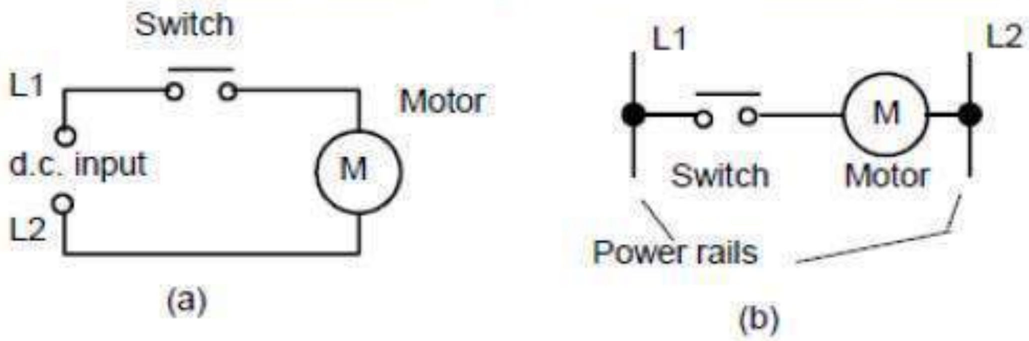


Figure 1. Ways of drawing the same electrical circuit

Figure 2 shows an example of a ladder diagram for a circuit that is used to start and stop a motor using push buttons. In the normal state, push button 1 is open and push button 2 closed. When button 1 is pressed, the motor circuit is completed and the motor starts. Also, the holding contacts wired in parallel with the motor close and remain closed as long as the motor is running. Thus when the push button 1 is released, the holding contacts maintain the circuit and hence the power to the motor. To stop the motor, button 2 is pressed. This disconnects the power to the motor and the holding contacts open. Thus when push button 2 is released, there is still no power to the motor. Thus we have a motor which is started by pressing button 1 and stopped by pressing button 2.

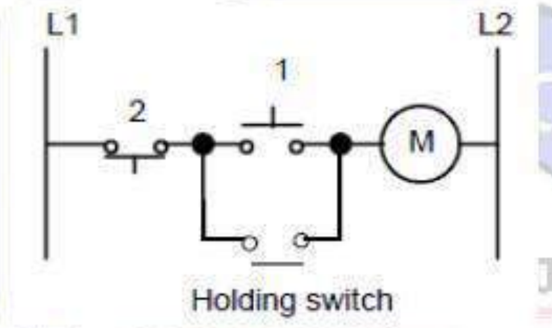


Figure 2. Stop-start switch

- **PLC ladder programming**

A very commonly used method of programming PLCs is based on the use of ladder diagrams. Writing a program is then equivalent to drawing a switching circuit. The ladder diagram consists of two vertical lines representing the power rails. Circuits are connected as horizontal lines, i.e. the rungs of the ladder, between these two verticals. In drawing a ladder diagram, certain conventions are adopted:

1) The vertical lines of the diagram represent the power rails between which circuits are connected. The power flow is taken to be from the left-hand vertical across a rung.
2) Each rung on the ladder defines one operation in the control process.
3) A ladder diagram is read from left to right and from top to bottom, Figure 3 showing the scanning motion employed by the PLC. The top rung is read from left to right. Then the second rung down is read from left to right and so on. When the PLC is in its run

mode, it goes through the entire ladder program to the end, the end rung of the program being clearly denoted, and then promptly resumes at the start (see Section 3). This procedure of going through all the rungs of the program is termed a cycle. The end rung might be indicated by a block with the word END or RET for return, since the program promptly returns to its beginning.

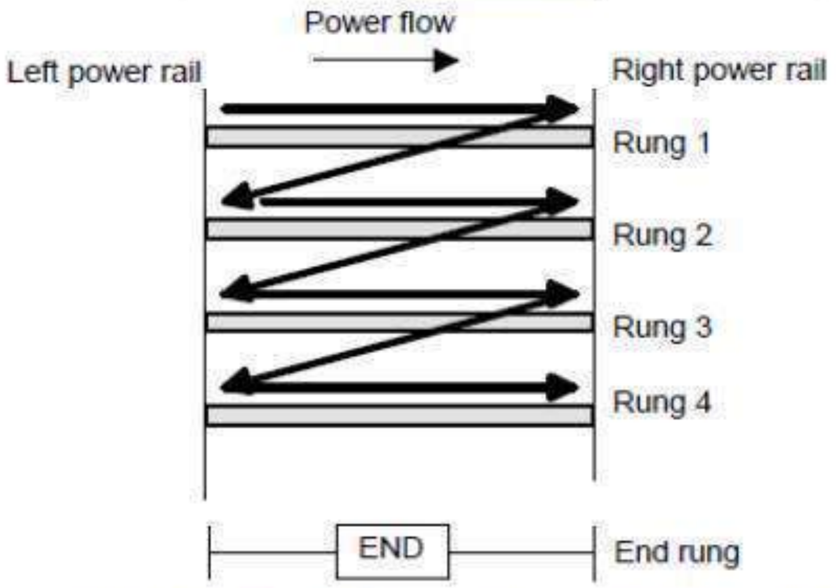


Figure 3. Scanning the ladder program

4) Each rung must start with an input or inputs and must end with at least one output. The term input is used for a control action, such as closing the contacts of a switch, used as an input to the PLC. The term output is used for a device connected to the output of a PLC, e.g. a motor.

5) Electrical devices are shown in their normal condition. Thus a switch which is normally open until some object closes it, is shown as open on the ladder diagram. A switch that is normally closed is shown closed.

6) A particular device can appear in more than one rung of a ladder. For example, we might have a relay which switches on one or more devices. The same letters and/or numbers

are used to label the device in each situation.

7) The inputs and outputs are all identified by their addresses, the notation used epending on the PLC manufacturer. This is the address of the input or output in the memory of the PLC.

In drawing ladder diagrams the names of the associated variable or addresses of each element are appended to its symbol. Thus Figure 4 shows how the ladder diagram of Figure 4(a) would appear using (a) Mitsubishi, (b) Siemens, (c) Allen-Bradley, (d) Telemecanique notations for the addresses. Thus Figure 4(a) indicates that this rung of the ladder program has an input from address X400 and an output to address Y430. When wiring up the inputs and outputs to the PLC, the relevant ones must be connected to the input and output terminals with these addresses.

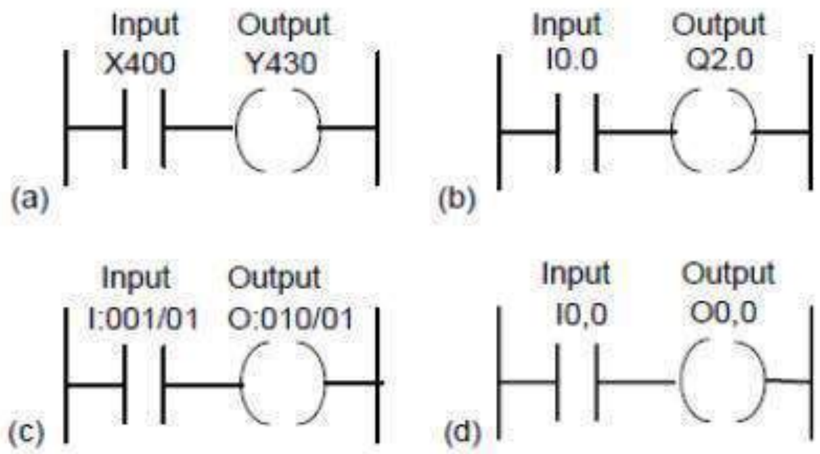


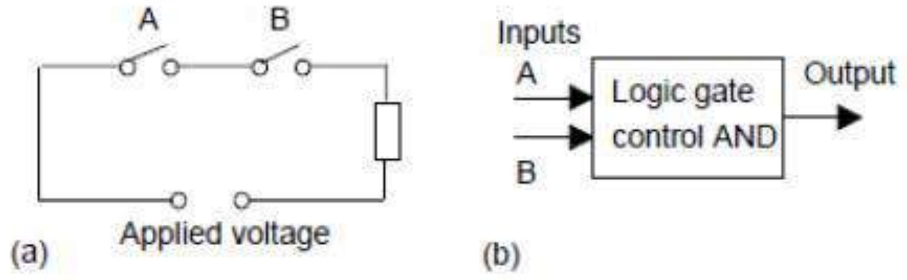
Figure 4 Notation: (a) Mitsubishi, (b) Siemens, (c) Allen-Bradley, (d) Telemecanique

Logic functions

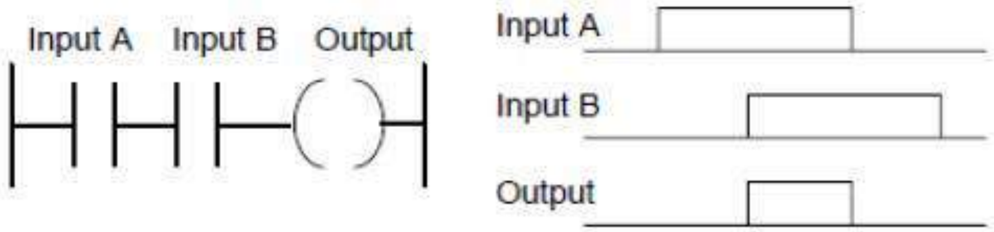
There are many control situations requiring actions to be initiated when a certain combination of conditions is realised. Thus, for an automatic drilling machine, there might be the condition that the drill motor is to be activated when the limit switches are activated that indicate the presence of the workpiece and the drill position as being at the surface of the workpiece. Such a situation involves the AND logic function, condition A and condition B having both to be realised for an output to occur. This section is a consideration of such logic functions.

- **AND gate**

Inputs		Output
A	B	
0	0	0
0	1	0
1	0	0
1	1	1



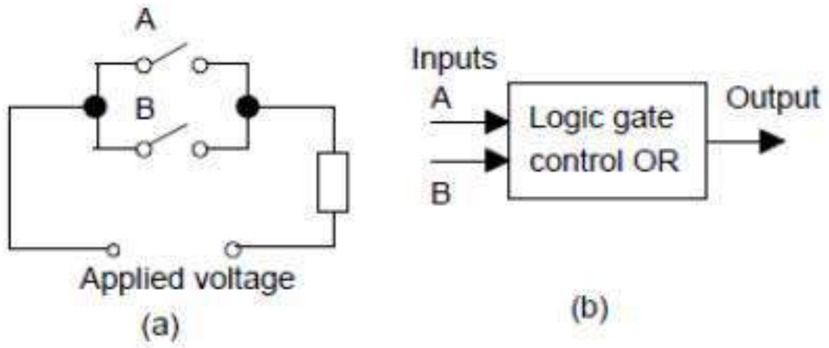
(a) AND circuit, *(b) AND logic gate*



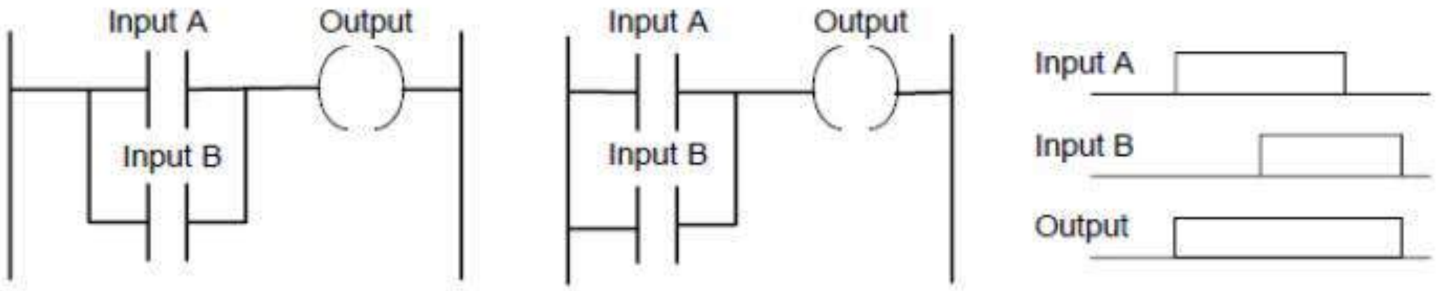
AND gate with a ladder diagram rung

➢ OR gate

Inputs		Output
A	B	
0	0	0
0	1	1
1	0	1
1	1	1



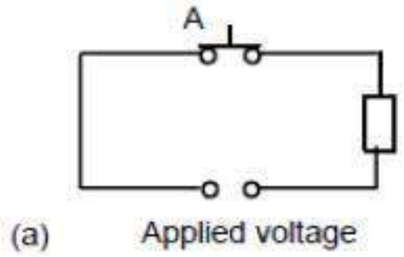
(a) OR electrical circuit, (b) OR logic gate

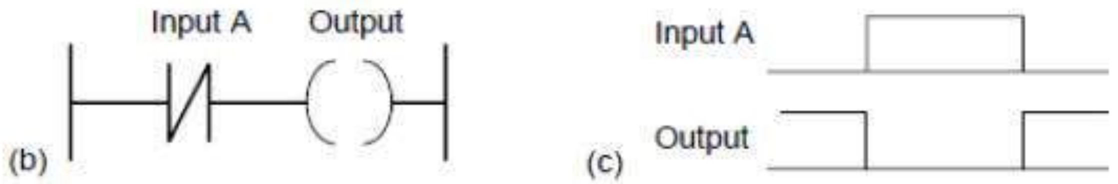


OR gate

➢ NOT gate

Input	Output
A	
0	1
1	0

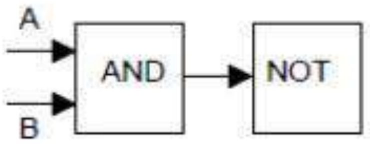




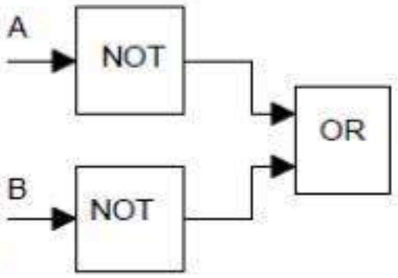
(a) NOT circuit, (b) NOT logic with a ladder rung, (c) high output when no input to A

➢ NAND gate

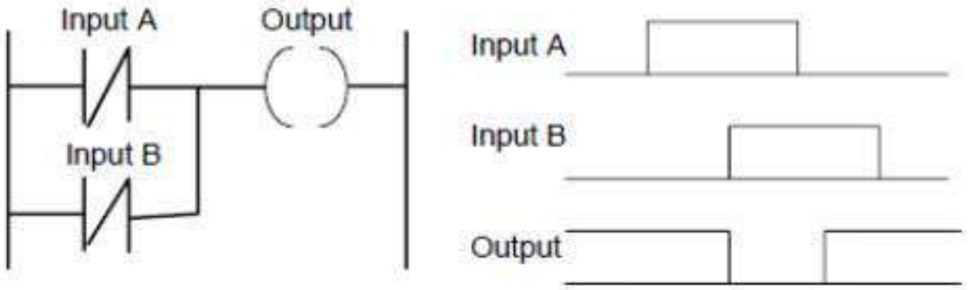
Inputs		Output
A	B	
0	0	1
0	1	1
1	0	1
1	1	0



(a)



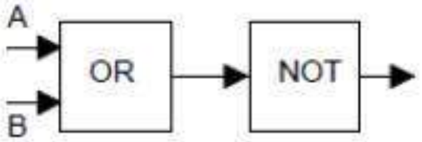
(b)



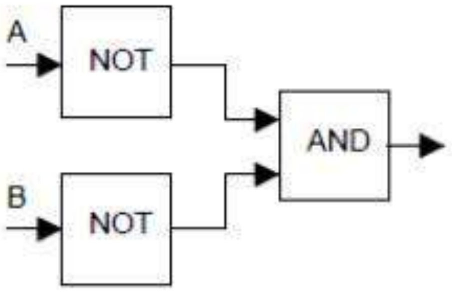
NAND gate

➢ NOR gate

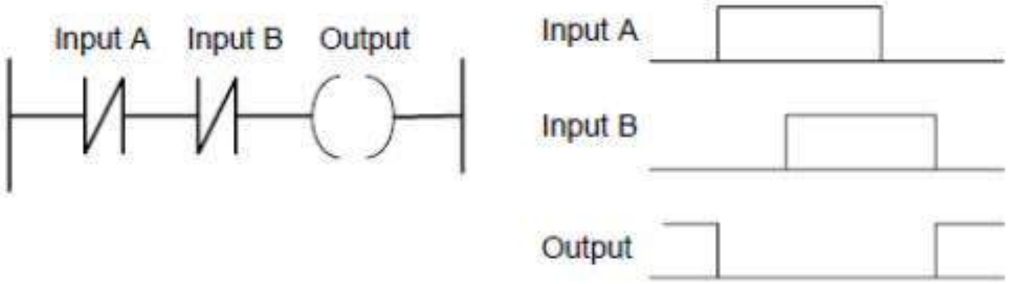
Inputs		Output
A	B	
0	0	1
0	1	0
1	0	0
1	1	0



(a)



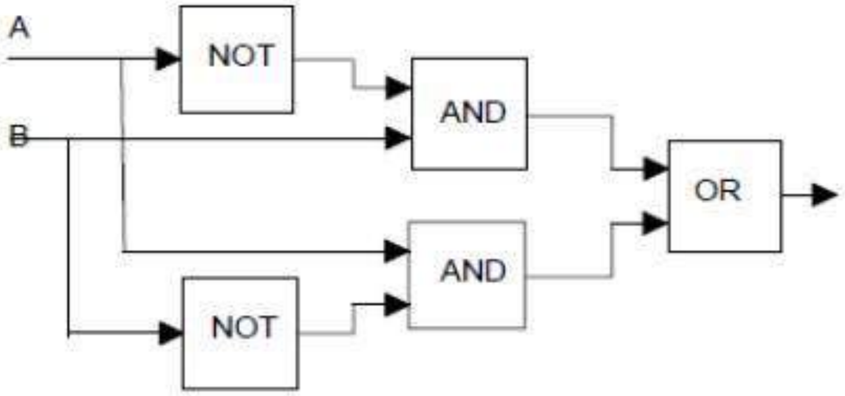
(b)

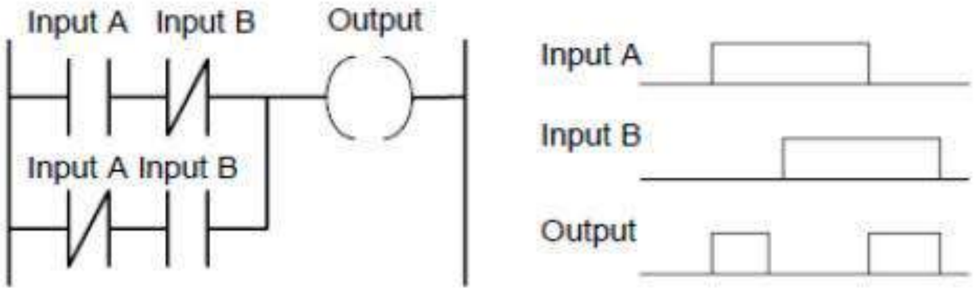


NOR gate

➢ Exclusive OR (XOR) gate

Inputs		Output
A	B	
0	0	0
0	1	1
1	0	1
1	1	0





XOR gate

Latching

There are often situations where it is necessary to hold an output energised, even when the input ceases. A simple example of such a situation is a motor which is started by pressing a push button switch. Though the switch contacts do not remain closed, the motor is required to continue running until a stop push button switch is pressed. The term latch circuit is used for the circuit used to carry out such an operation. It is a self-maintaining circuit in that, after being energised, it maintains that state until another input is received. An example of a latch circuit is shown in Figure 5.18. When the input A contacts close, there is an output. However, when there is an output, another set of contacts associated with the output closes. These contacts form an OR logic gate system with the input contacts. Thus, even if the input A opens, the circuit will still maintain the output energised. The only way to release the output is by operating the normally closed contact B.

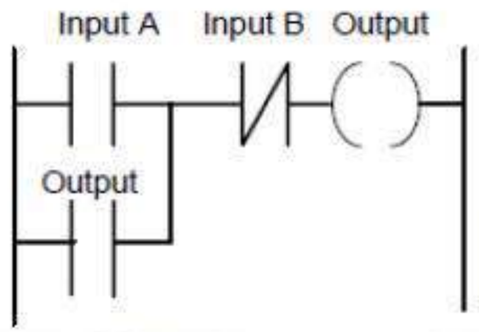


Figure 5. Latched circuit

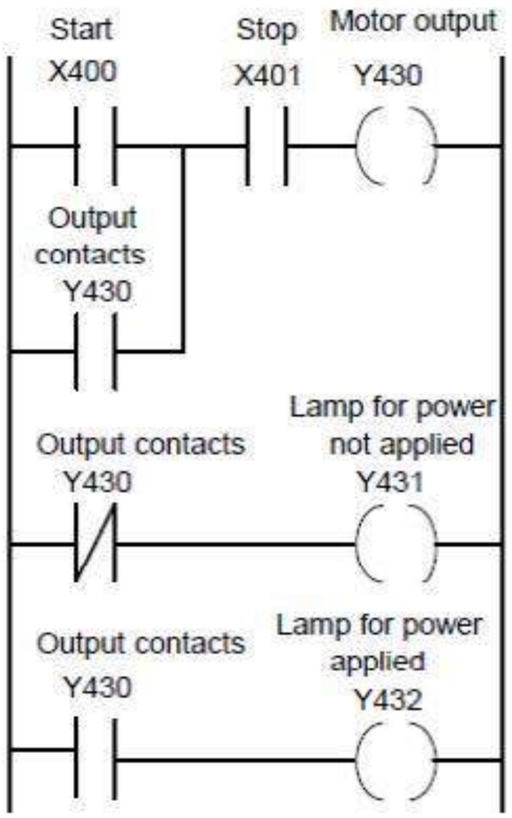


Figure 6. Motor on-off, with signal lamps, ladder diagram. Note that the stop contacts X401 are shown as being programmed as open. If the stop switch used is normally closed then X401 receives a start-up signal to close. This gives a safer operation than programming X401 as normally closed.

Multiple outputs

With ladder diagrams, there can be more than one output connected to a contact. Figure 7 shows a ladder program with two output coils. When the input contacts close both the coils give outputs.

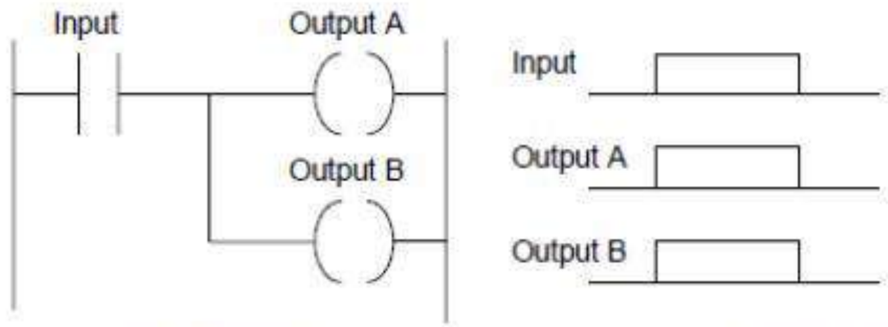


Figure 7. Ladder rung with two outputs

For the ladder rung shown in Figure 8, output A occurs when input A occurs. Output B only occurs when both input A and input B occur.

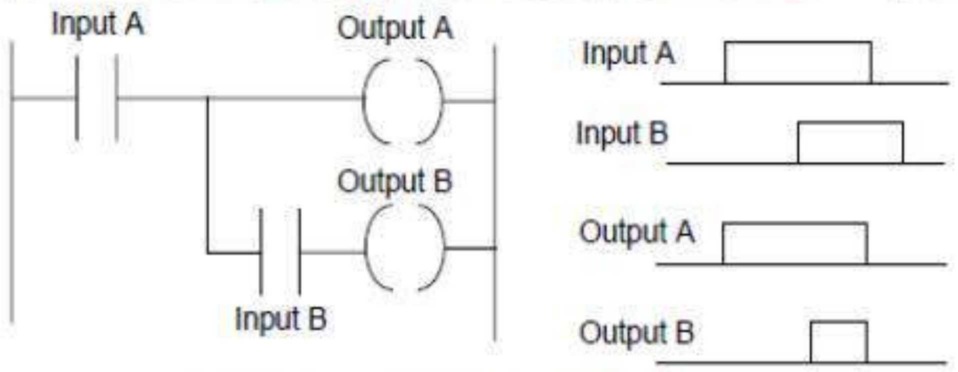


Figure 8. Ladder rung with two inputs and two outputs

Such an arrangement enables a sequence of outputs to be produced, the sequence being in the sequence with which contacts are closed. Figure 9 illustrates this with the same ladder program in Mitsubishi and Siemens notations. Outputs A, B and C are switched on as the contacts in the sequence given by the contacts A, B and C are being closed. Until input A is closed, none of the other outputs can be switched on. When input A is closed, output A is switched on. Then, when input B is closed, output B is switched on. Finally, when input C is closed, output C is switched on.

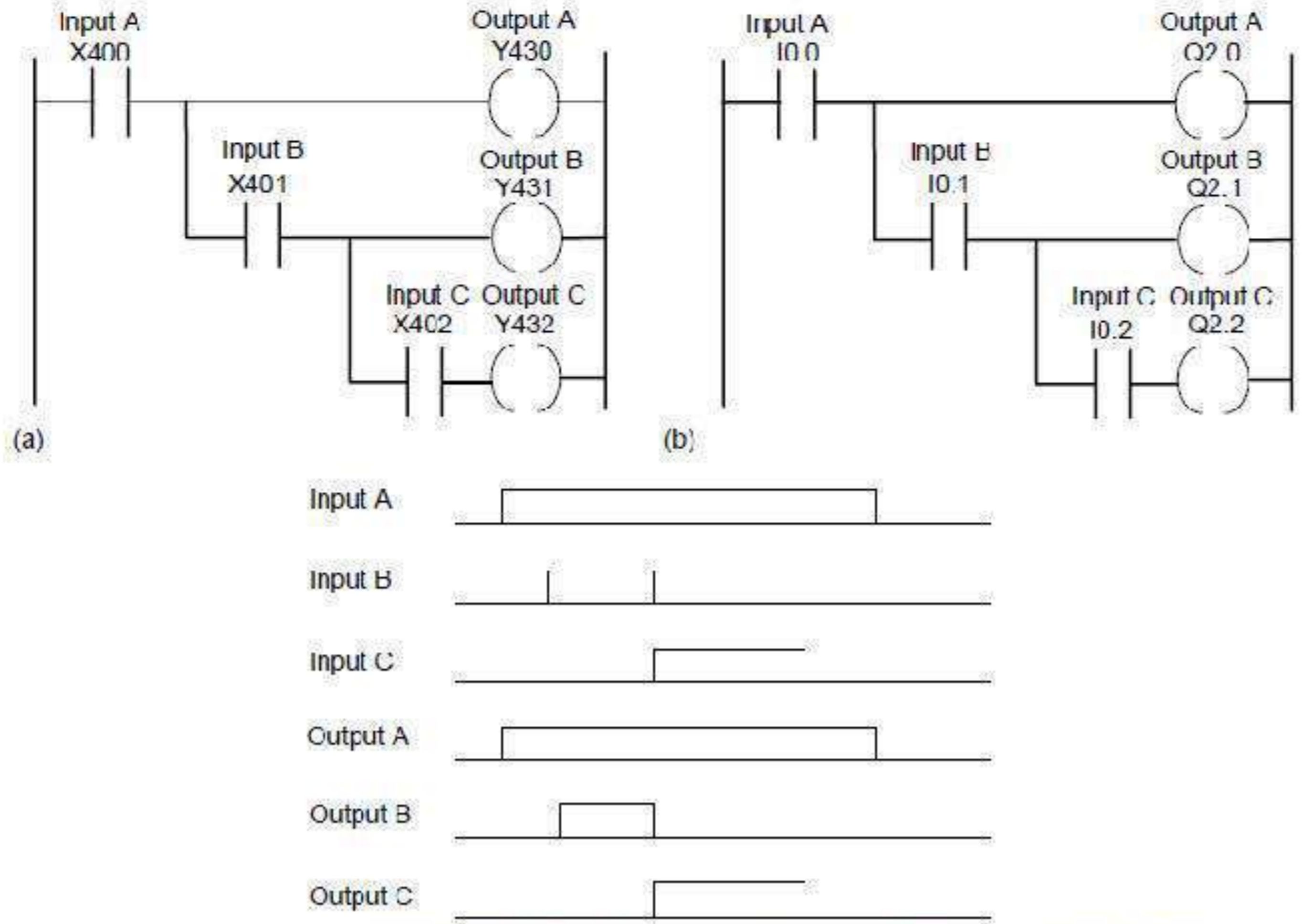


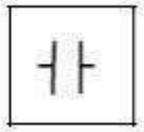
Figure 9. Sequenced outputs

Entering programs

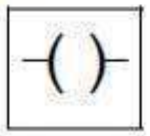
Each horizontal rung on the ladder represents an instruction in the program to be used by the PLC. The entire ladder gives the complete program. There are several methods that can be used for keying in the program into a programming terminal. Whatever method is used to enter the program into a programming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

➢ **Ladder symbols**

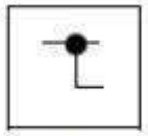
One method of entering the program into the programming terminal involves using a keypad having keys with symbols depicting the various elements of the ladder diagram and keying them in so that the ladder diagram appears on the screen of the programming terminal. For example, to enter a pair of contacts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



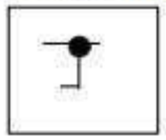
might be used, followed by its address being keyed in. To enter an output the key marked



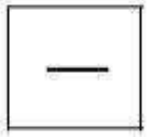
might be used, followed by its address. To indicate the start of a junction



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



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



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

Function blocks

The term function block diagram (FBD) is used for PLC programs described in terms of graphical blocks. It is described as being a graphical language for depicting signal and data flows through blocks, these being reusable software elements. A function block is a program instruction unit which, when executed, yields one or more output values. Thus a block is represented in the manner shown in Figure 10 with the function name written in the box.

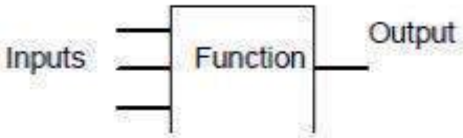
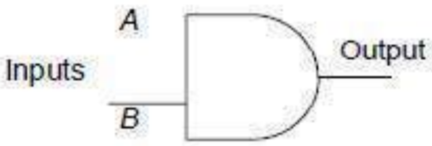
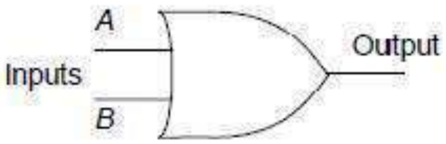
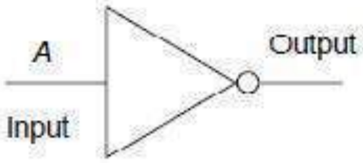


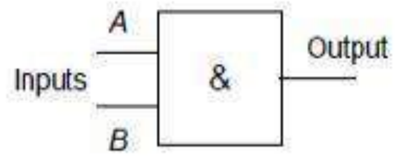
Figure 10. Function block

For all logic gates we have symbols as follow:

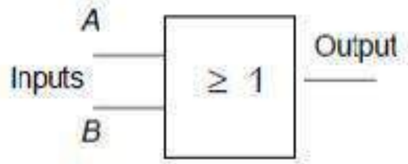




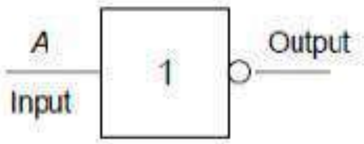




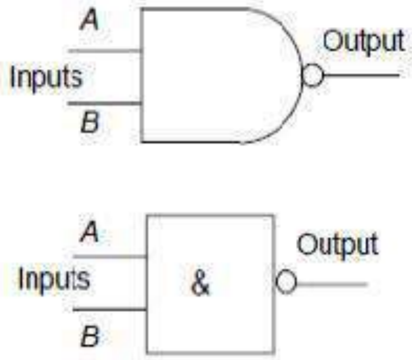
AND gate



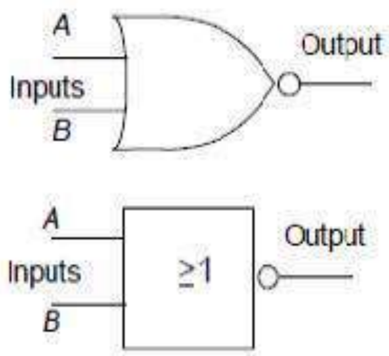
OR gate



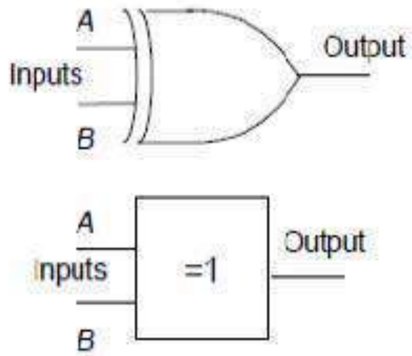
Not gate



NAND gate

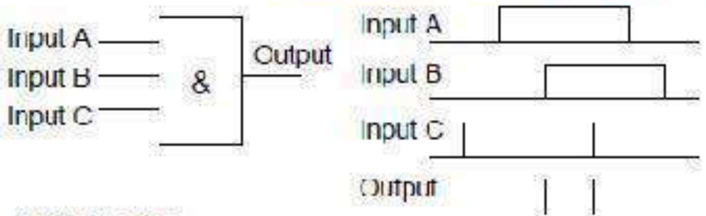


NOR Gate

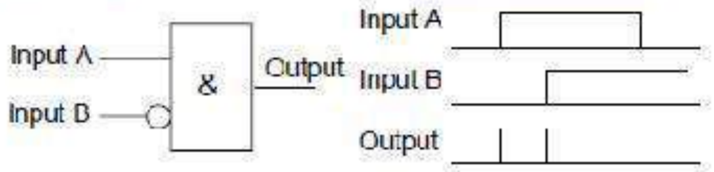


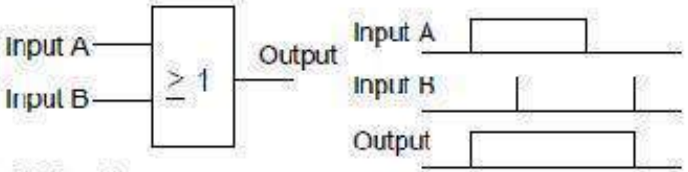
XOR Gate

Figure below shows the effect of such functional blocks in PLC programs:

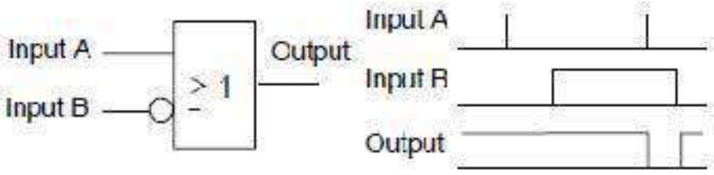


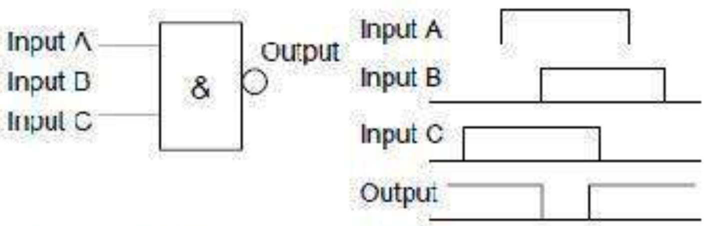
AND function



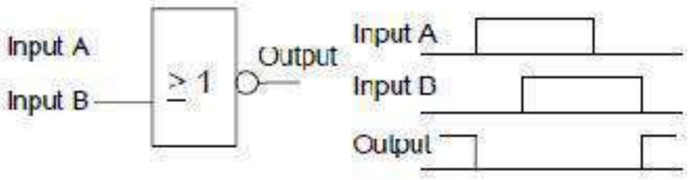


OR function

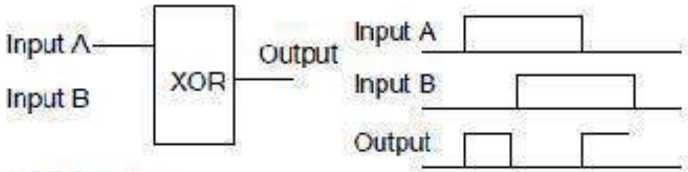




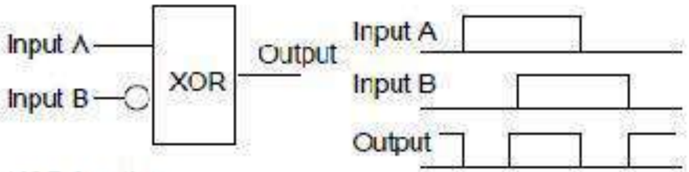
NAND function



NOR function



XOR function



XOR function

To illustrate the form of such a diagram and its relationship to the ladder diagram, Figure 11 shows an OR gate. When A or B inputs are 1 then there is an output.

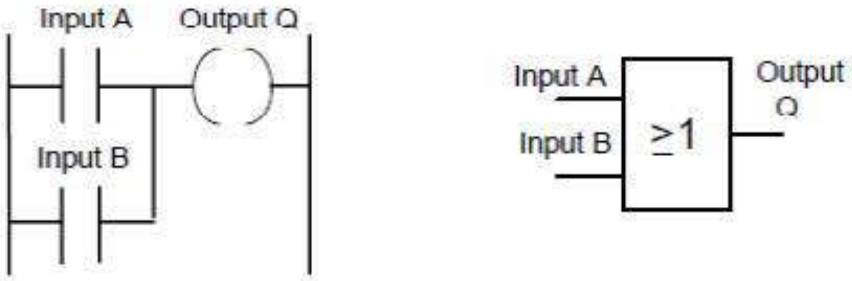


Figure 11. Ladder diagram and equivalent functional block diagram

Figure 12 shows a ladder diagram and its function block equivalent in Siemens notation. The = block is used to indicate an output from the system.

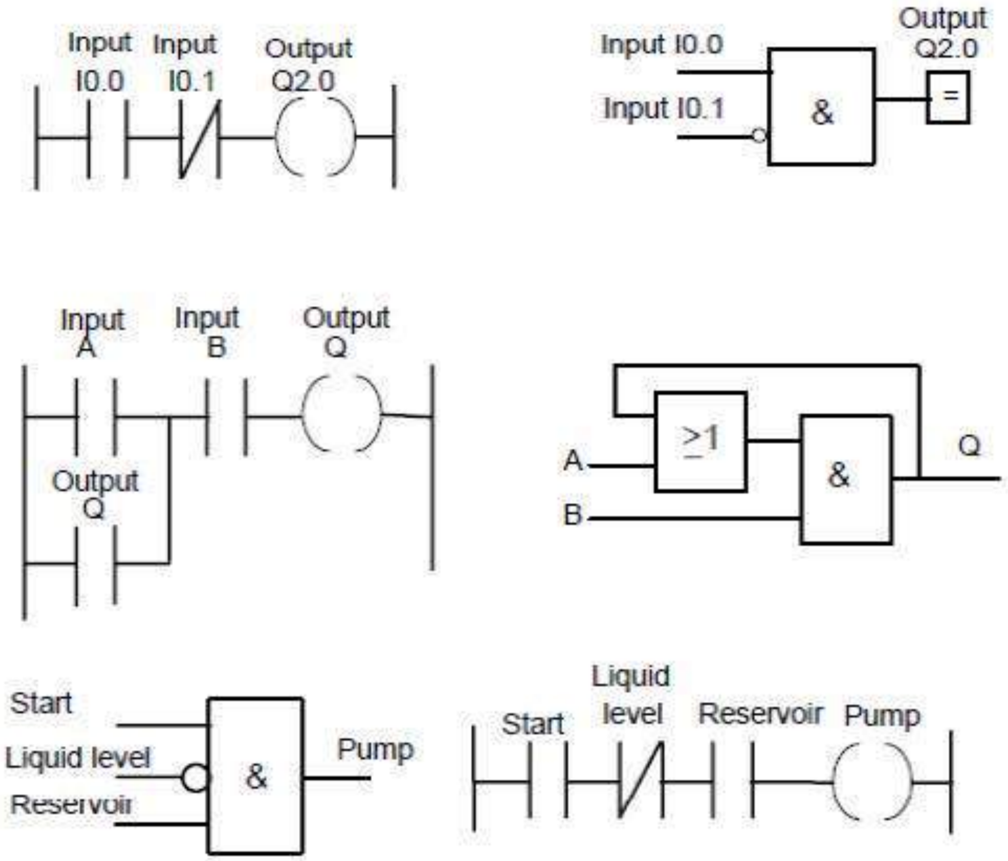


Figure 12. Ladder diagram and equivalent function block diagram

Boolean algebra

Ladder programs can be derived from Boolean expressions since we are concerned with a mathematical system of logic. In Boolean algebra there are just two digits, 0 and 1. When we have an AND operation for inputs A and B then we can write:

$$A.B = Q$$

Where Q is the output. Thus Q is equal to 1 only when A = 1 and B = 1. The OR operation for inputs A and B is written as:

$$A + B = Q$$

Thus Q is equal to 1 only when A = 1 or B = 1. The NOT operation for an input A is written as:

$$A = Q$$

Thus when A is not 1 there is an output. As an illustration of how we can relate Boolean expressions with ladder diagrams, consider the expression:

$$A + B.C = Q$$

This tells us that we have A or the term B and C giving the output Q. Figure 5.32 shows the ladder and functional block diagrams. Written in terms of Mitsubishi notation, the above expression might be:

$$X400 + X401.X402 = Y430$$

As a further illustration, consider the Boolean expression:

$$A + B = Q$$

Figure 13 shows the ladder and functional block diagrams.

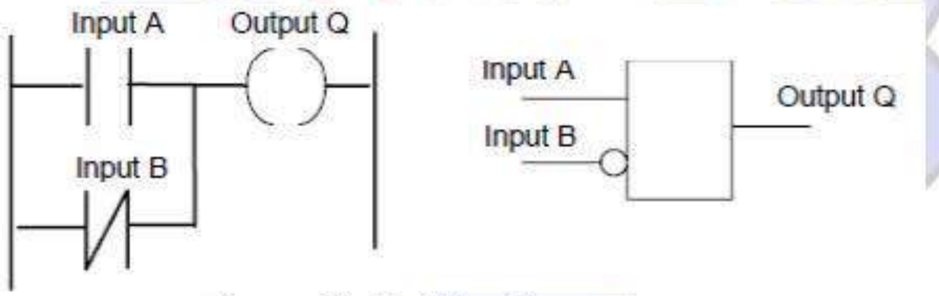


Figure 13. Ladder diagram

A.B From the OR gate we obtain an output of A.B + C. From the lower AND gate we obtain an output Q of:

$$(A.B + C).D.E.F = Q$$

The ladder diagram to represent this is shown in Figure 14.

Figure 14. Ladder diagram for (A.B + C).D.E.F = Q

Program examples

The following tasks are intended to illustrate the application of the programming techniques given in this chapter. A signal lamp is required to be switched on if a pump is running and the pressure is satisfactory, or if the lamp test switch is closed. For the inputs from the pump and the pressure sensors we have an AND logic situation since both are required if there is to be an output from the lamp. We, however, have an OR logic situation with the test switch in that it is required to give an output of lamp on regardless of whether there is a signal from the AND system. The function block diagram and the ladder diagram are thus of the form shown in Figure 15. Note that with the ladder diagram we tell the PLC when it has reached the end of the program by the use of the END or RET instruction. As another example, consider a valve which is to be operated to lift a load when a pump is running and either the lift switch is operated or a switch operated indicating that the load has not already been lifted and is at the bottom of its lift channel. We have an OR situation for the two switches and an AND situation involving the two switches and the pump. Figure 16 shows a possible program.

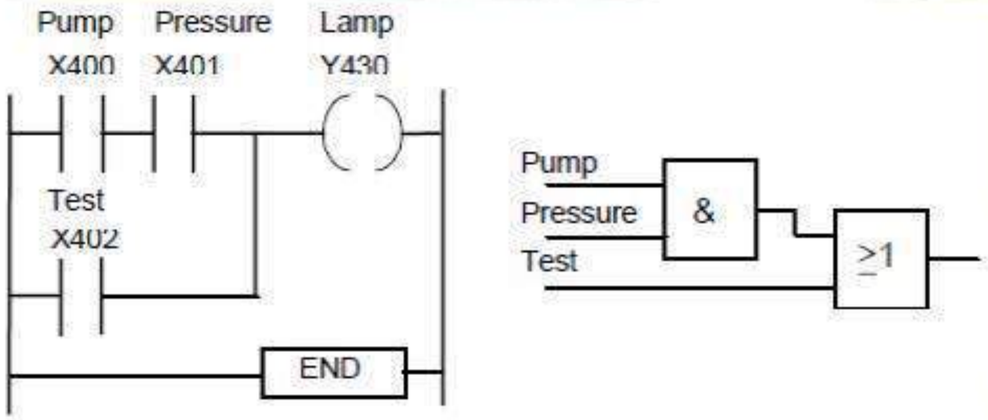


Figure 15. Signal lamp task

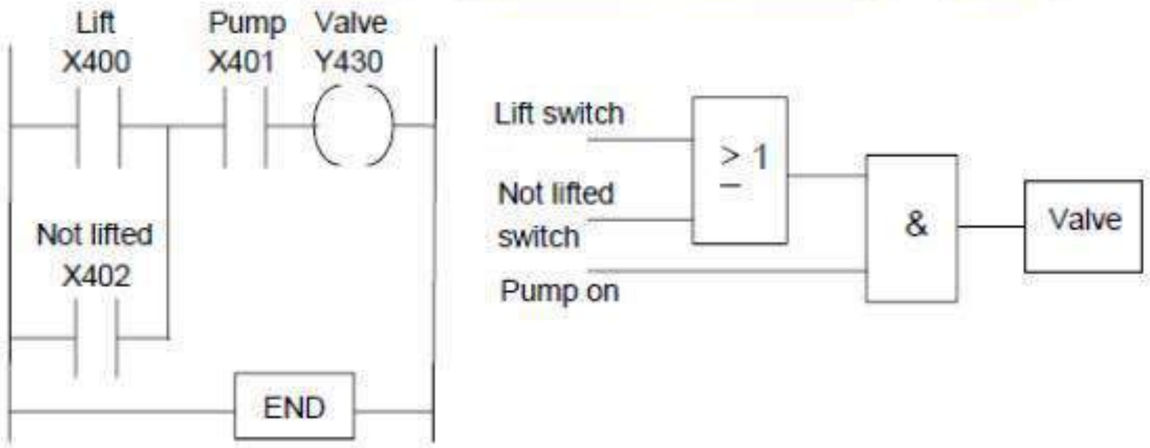


Figure 16. Valve operation program

As another example, consider a system where there has to be no output when any one of four sensors gives an output, otherwise there is to be an output. One way we could write a program for this is for each sensor to have contacts that are normally closed so there is an output. When there is an input to the sensor the contacts open and the output stops. We have an AND logic situation. Figure 17 shows the functional block diagram and the ladder diagram of a system that might be used.

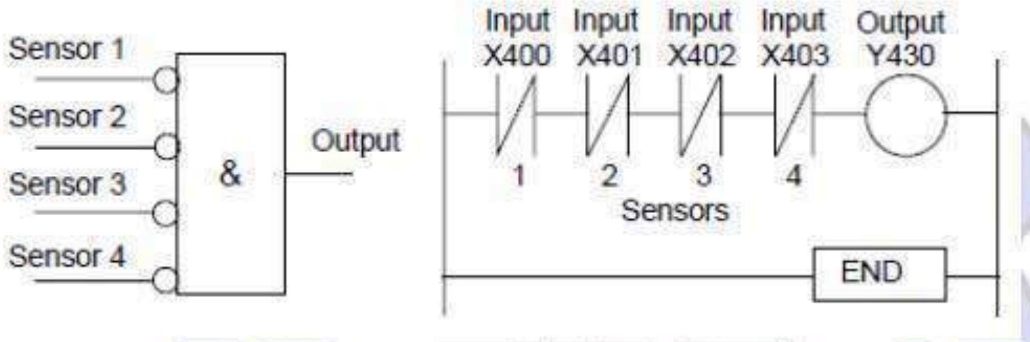
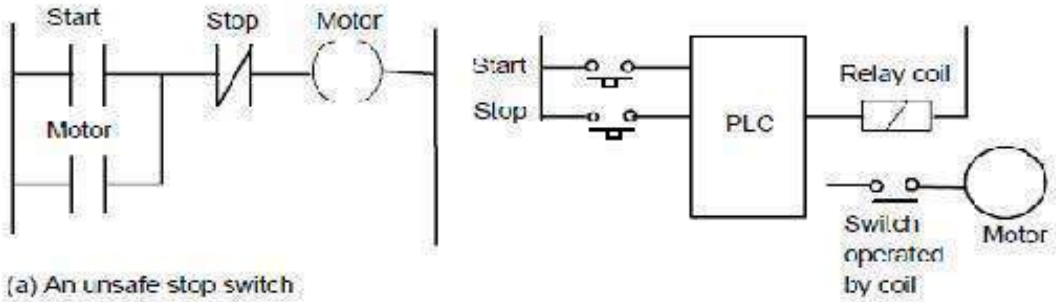


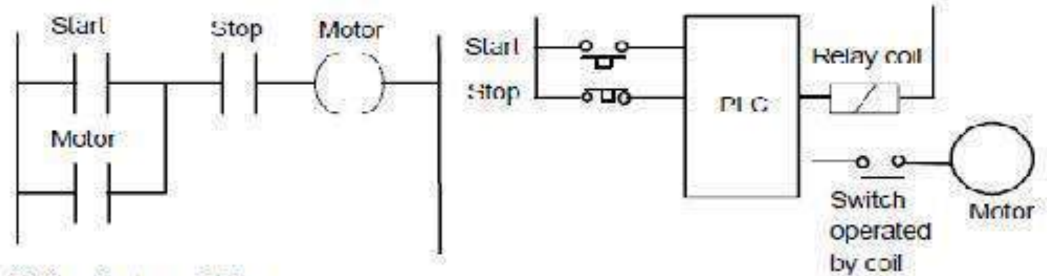
Figure 17. Output switched off by any one of four sensors being activated

- **Location of stop switches**

The location of stop switches with many applications has to be very carefully considered in order to ensure a safe system. A stop switch is not safe if it is normally closed and has to be opened to give the stop action. If the switch malfunctions and remains closed then the system cannot be stopped. Figure 18(a) illustrates this. A better arrangement is to program the stop switch in the ladder program as open in Figure 18(b) and use a stop switch that is normally closed and operating opens it. Thus there is an input signal to the system which closes the contacts in the program when it starts up.



(a) An unsafe stop switch



(b) A safe stop switch

Figure 18. Motor stop switch location

Figure 19 shows where we can safely locate an emergency stop switch. If it is in the input to the PLC (Figure 19(a)) then if the PLC malfunctions it may not be possible to stop the motor. However, if the emergency stop switch is in the output, operating it will stop the motor and also cause the start switch to become unlatched if the arrangement shown in Figure 19(b) is being used. The motor will thus not restart when the emergency stop button is released.

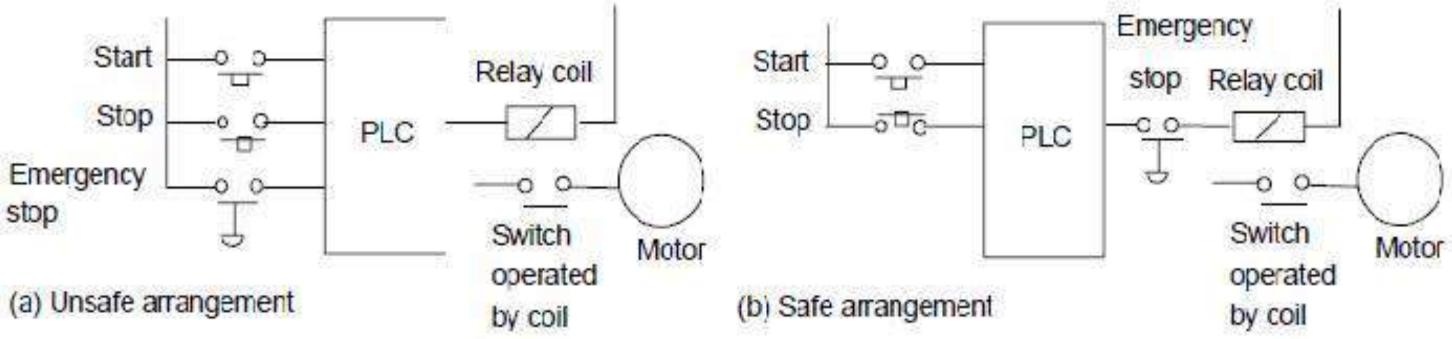


Figure 19. Location of emergency stop switch

- **Counter:**

Figure 20 shows a basic counting circuit. When there is a pulse input to In 1, the counter is reset. When there is an input to In 2, the counter starts counting. If the counter is set for, say, 10 pulses, then when 10 pulse inputs have been received at In 2, the counter's contacts will close and there will be an output from Out 1. If at any time during the counting there is an input to In 1, the counter will be reset and start all over again and count for 10 pulses.

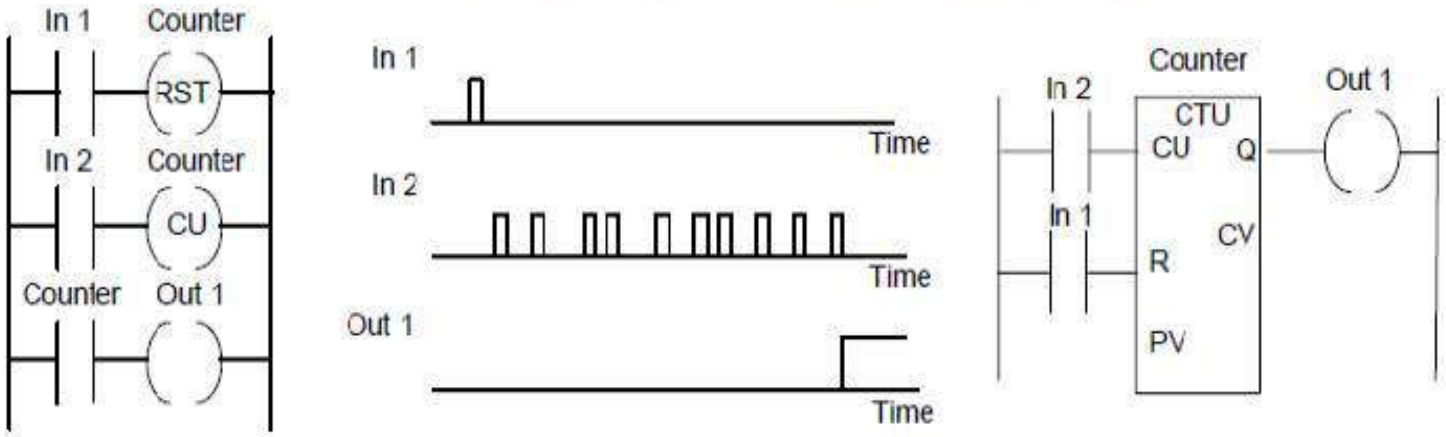


Figure 20. Basic counter program

Timers with counters

A typical timer can count up to 16 binary bits of data, this corresponding to 32 767 base time units. Thus, if we have a time base of 1 s then the maximum time that can be dealt with by a timer is just over 546 minutes or 9.1 hours. If the time base is to be 0.1 s then the maximum time is 54.6 minutes or just short of an hour. By combining a timer with a counter, longer times can be counted. Figure 21 illustrates this with an Allen-Bradley program. If the timer has a time base of 1 s and a preset value of 3600, then it can count for up to 1 hour. When input I:012/01 is activated, the timer starts to time in one second increments. When the time reaches the preset value of 1 hour, the DN bit is set to 1 and the counter increments by 1. The DN bit setting to 1 also reset the timer and the timer starts to time again. When it next reaches its preset time of 1 hour, the DN bit is again set to 1 and the counter increments by 1. With the counter set to a preset value of 24, the counter DN bit is set to 1 when the count reaches 24 and the output O:013/01 is turned on. We thus have a timer which is able to count the seconds for the duration of a day and would be able to switch on some device after 24 hours.

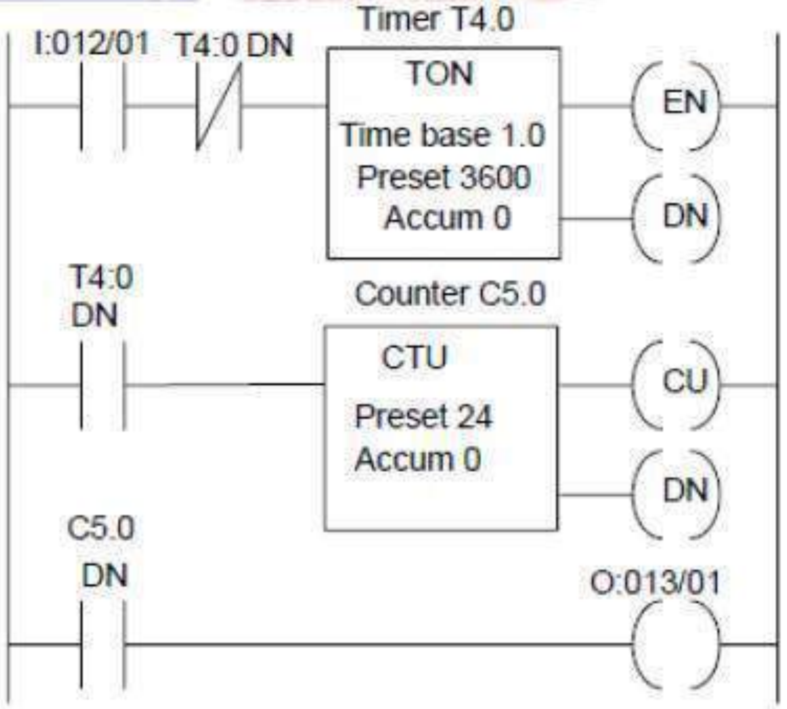


Figure 21. Using a counter to extend the range of a timer

Exp No (1) Basic Logic Gates Experiment

The procedure for this experiment as follows:

1- Switch OFF main supply 220V AC & switch OFF 24Vdc

2- Switch Off all tested digital input from (TI00…TI15)

3- Make the following Connection carefully:

	from	to
1	**T00**	**I00**
2	**T01**	**I01**
3	**T05**	**I05**
4	**T06**	**I06**
5	**T02**	**I02**
6	**T03**	**I03**

4- make the ladder diagram shown in figure with all parameters (*setting the initial condition of ENN1 variable to Logic 1*)

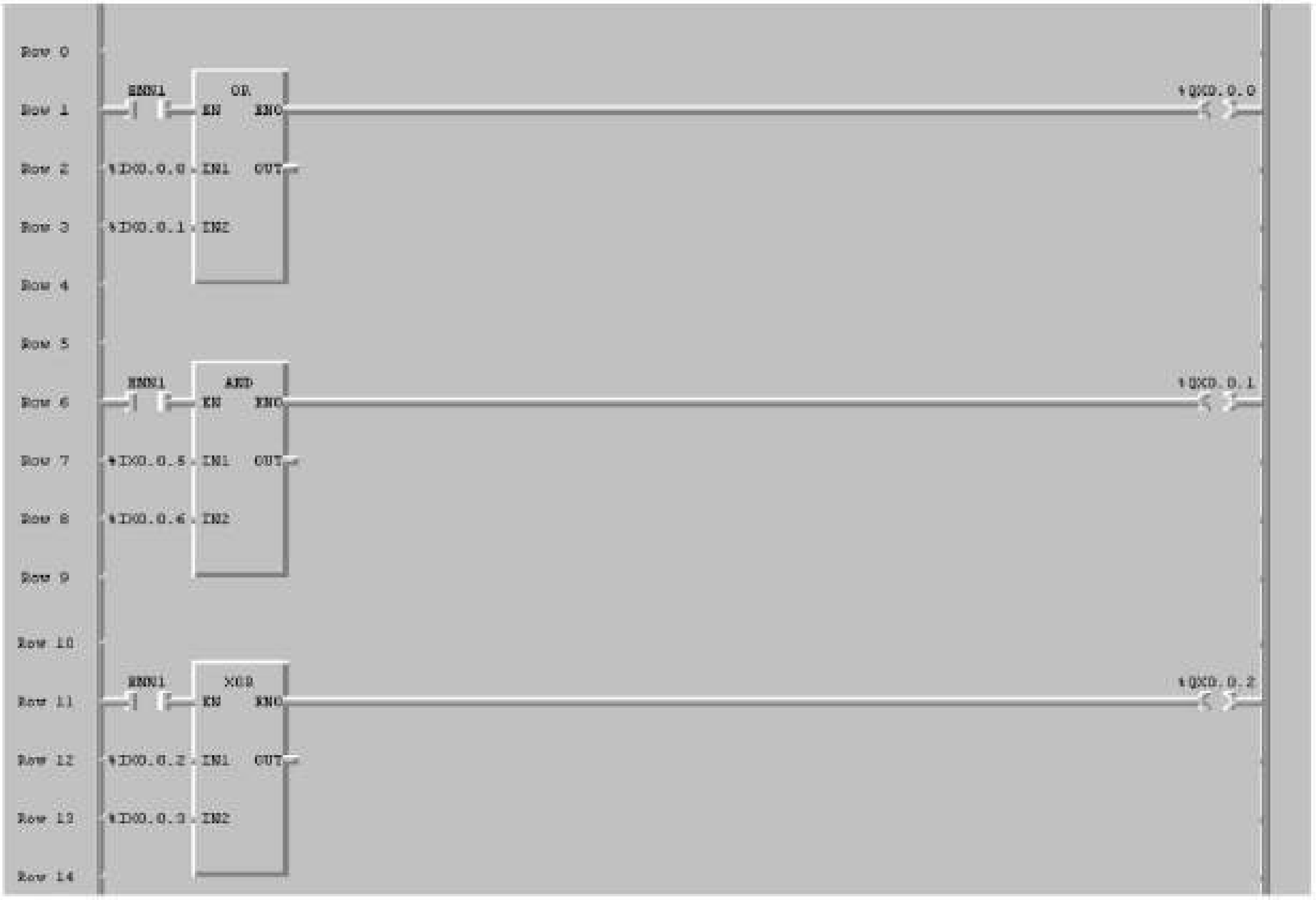
5- Switch ON main supply 220V AC

6- Download the ladder program to PLC

7- Three Digital output must appear on O00 , O01 & O02 according the following function:

	Output	function
1	**O00**	**I00 OR I01**
2	**O01**	**I05 AND I06**
3	**O02**	**I02 XOR I03**

8- Changes the status of (T00,T01,T02,T03,T05 &T06) switches and record the output (O00,O01 & O02)



Ladder Digram of Basic Logic Gates Experiment

Exp No (2) Clock Generation Experiment

The procedure for this experiment as follows:

1- Switch OFF main supply 220V AC & switch OFF 24Vdc

2- make the ladder diagram shown in figure with all parameters (*setting the initial condition of EN1 variable to Logic 1*)

3- Switch ON main supply 220V AC

4- Download the ladder program to PLC

5- One Digital output (O01) must blinking (5 sec Logic 1 & Logic 0 5sec)

Ladder Digram of Clock Generation Experiment

Exp No (3) Selection Continuous Up/Down Counter Experiment

The procedure for this experiment as follows:

1- Switch OFF main supply 220V AC & switch OFF 24Vdc

2- Switch Off all tested digital input from (TI00…TI15)

3- Make the following Connection carefully:

	from	to
1	**T01**	**I01**
2	**T02**	**I02**

4- make the ladder diagram shown in figure with all parameters (*setting the initial condition of EN1& TT variables to Logic 1*)

5- Switch ON main supply 220V AC

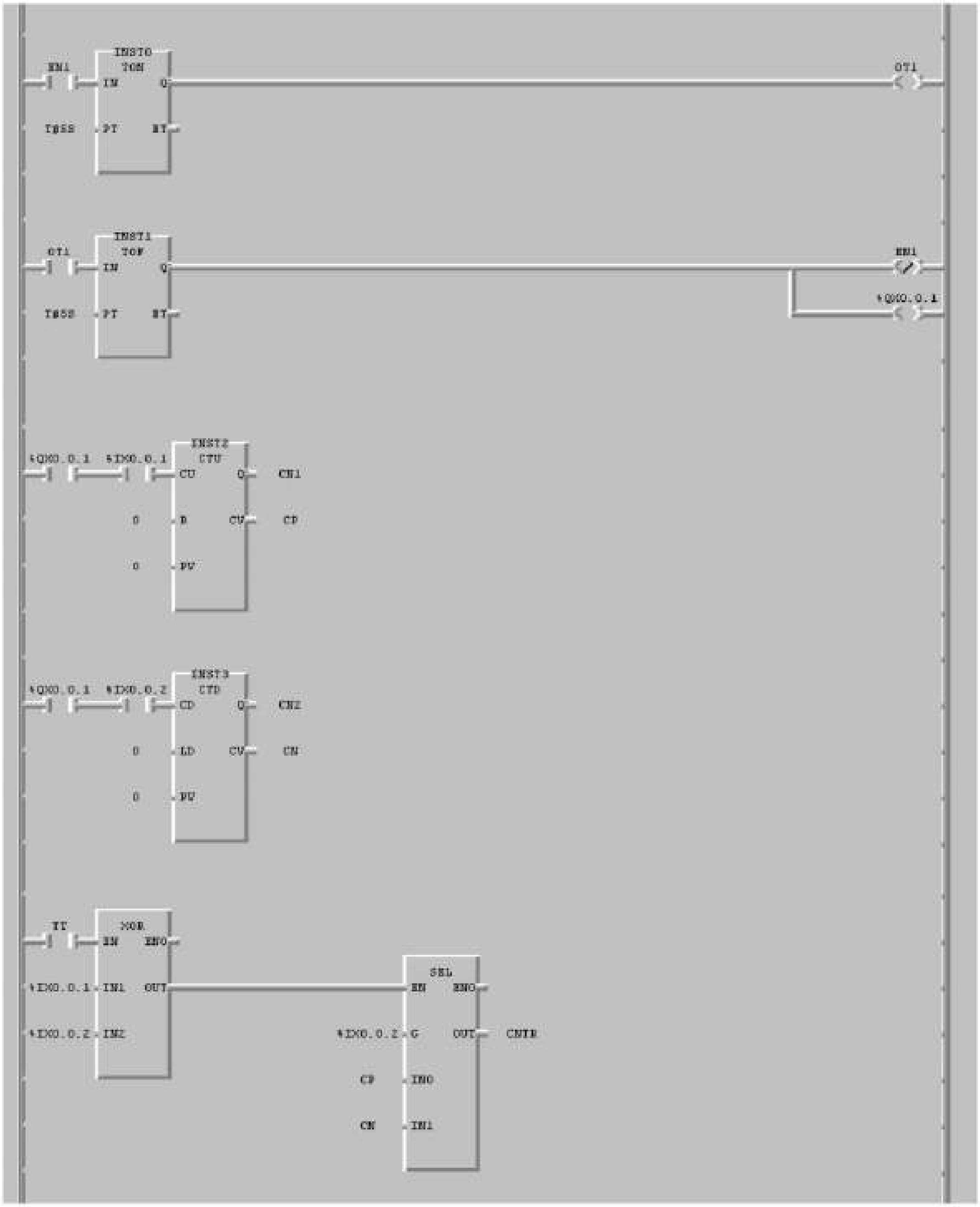
6- Download the ladder program to PLC

7- One Digital output (O01) must blinking (5 sec Logic 1 & Logic 0 5sec)

8- Switch ON (T01) counter up must start work

9- Switch ON (T02) counter down must start work

10- When two counter work SEL Module not work because XOR Out (Logic 0) which disable Selection Module, Now try to stop work one of counter using (T01 & T02) switches in order to enable SEL module and then record the results.



Ladder Digram of Selection Continuos Up/Down Counter Experiment

Exp No (4) Generation Second Clock using Triggered of uniform continuous clock Experiment

The procedure for this experiment as follows:

1- Switch OFF main supply 220V AC & switch OFF 24Vdc

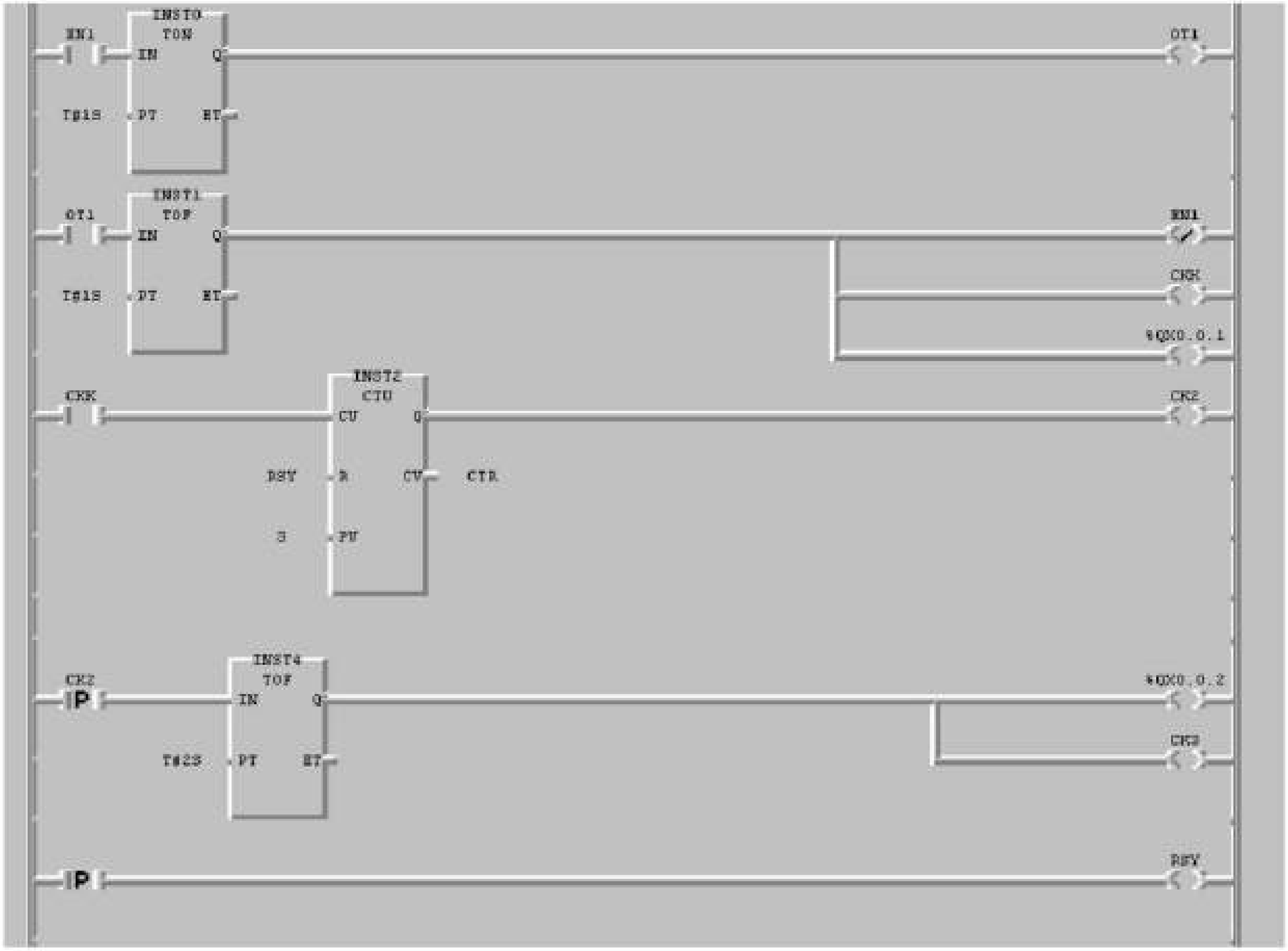
2- make the ladder diagram shown in figure with all parameters (*setting the initial condition of EN1& TT variables to Logic 1*)

3- Switch ON main supply 220V AC

4- Download the ladder program to PLC

5- Digital output (O01) must blinking (1 sec Logic 1 & Logic 0 1sec)

6- Digital Output (O02) must work for (2 sec) after 3 postive triger from first clock



Ladder Digram of Selection Continuos Up/Down Counter Experiment

Exp No (5): Conveyer Belt control Experiment

The procedure for this experiment as follows:

1- Switch OFF main supply 220V AC & switch OFF 24Vdc & switch OFF of conveyer belt.

2- Connect (24 V Dc) from Power supply to (COM) of (O01... O06).

3- Connect (O01... O06) to conveyer belt part according to the following diagram:

	from	to
1	**O01**	**P1**
2	**O02**	**P2**
3	**O03**	**P3**
4	**O04**	**P4**
5	**O05**	**P5**
6	**O06**	**P6**

4- Connect (REC) of Photo 1 to (I01) and (REC) of Photo 2 to (I03)

5- Connect (REC SIGNAL) of APPROXIMATE switch to (I02)

6- Switch ON main supply 220V AC

7- Download the ladder program of *conveyer belt control* to PLC

8- switch ON 24Vdc & switch ON conveyer belt & approximate switch & photo sensors

9- Activate photo sensor 1 for short time the process start work.

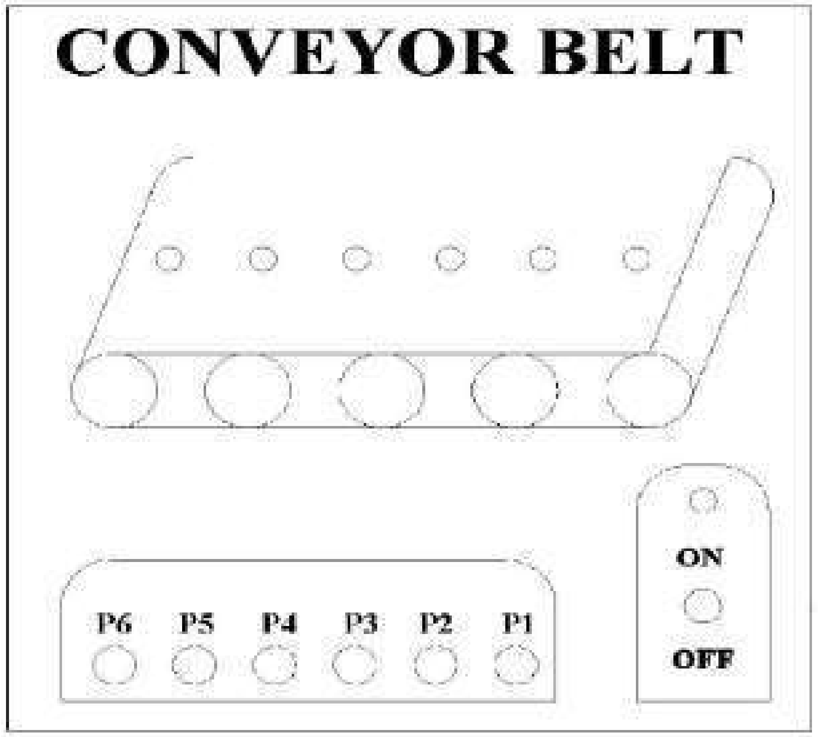
10- The conveyer belt LEDs must be activate according the following sequence :

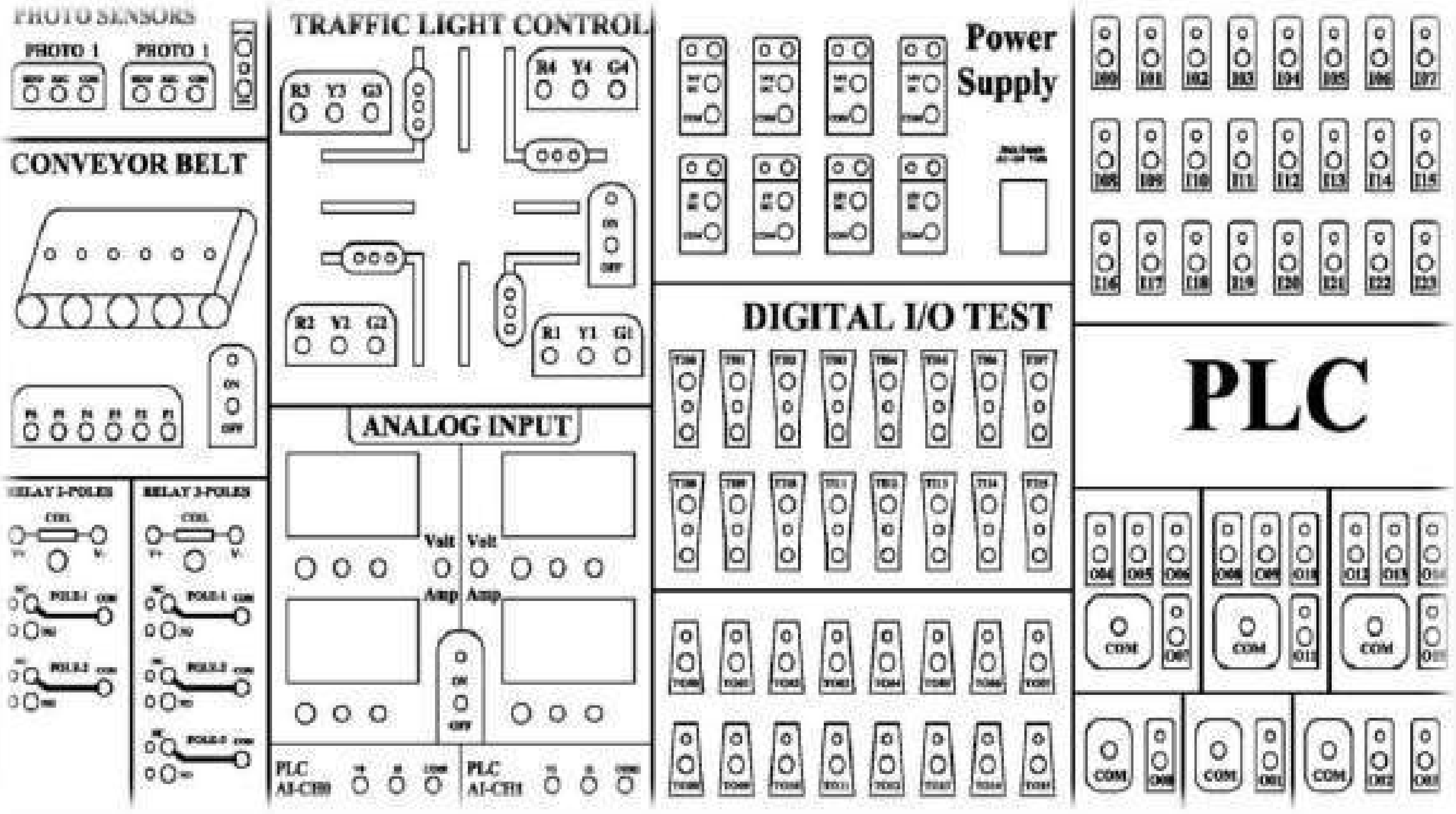
	Time	Blinking ON	Fix ON
State 1	0 to 2.5sec	P1	
State 2	2.6 to 5.0 sec	P1	P2
State 3	5.1 to 7.5 sec	P1	P2 & P3
State 4	7.6 to 10 sec	P1	P2 & P3
State 5	10.1 to 12.5 sec	P1	P2 & P3 & P4
State 6	12.6 to 15 sec	P1	P2 & P3 & P4 & P5
State 7	15.1 to 17.5 sec	P1 & P6	P2 & P3 & P4 & P5
State 8	17.6 to 20 sec		P2 & P3 & P4 & P5 & P6

11- Activate approximate switch for short time ,all LEDs(P1…P6) will be reset

12- Activate photo sensor 1 for short time the process play again.

13- Activate photo sensor 2 will freeze process until remove activate from sensor





Exp No (6): Traffic Light Control Experiment

The procedure for this experiment as follows:

1- Switch OFF main supply 220V AC & switch OFF 24Vdc

2- Connect (24 V Dc) from Power supply to (COM) of (O00... O11).

3- Connect (O00... O11) to traffic light part according to the following diagram:

	from	to
1	**O00**	**R1**
2	**O01**	**Y1**
3	**O02**	**G1**
4	**O03**	**R2**
5	**O04**	**Y2**
6	**O05**	**G2**
7	**O06**	**R3**
8	**O07**	**Y3**
9	**O08**	**G3**
10	**O09**	**R4**
11	**O10**	**Y4**
12	**O11**	**G4**

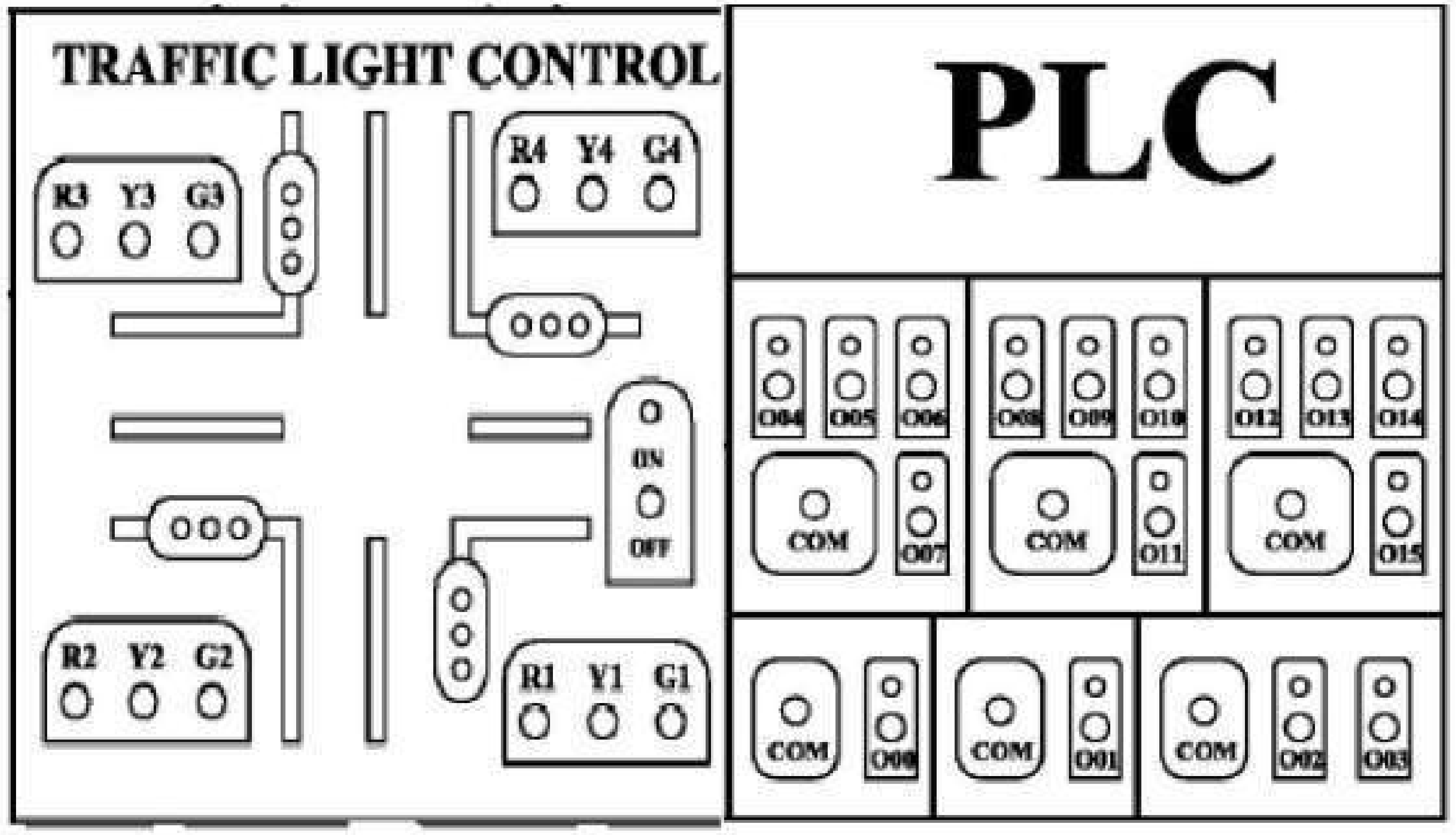
4- Switch ON main supply 220V AC

5- Download the ladder program of *traffic light control* to PLC

6- switch ON 24Vdc

7- The traffic light LEDs must be activate according the following sequence :

	Time	Traffic 1	Traffic 2	Traffic 3	Traffic 4
State 1	0 to 10sec	Green 1	Red 2	Red 3	Red 4
	11 to 15 sec	Yellow 1	Red 2	Red 3	Red 4
State 2	16 to 25 sec	Red 1	Green 2	Red 3	Red 4
	31 to 40 sec	Red 1	Yellow 2	Red 3	Red 4
State 3	41 to 45 sec	Red 1	Red 2	Green 3	Red 4
	46 to 50 sec	Red 1	Red 2	Yellow 3	Red 4
State 4	51 to 60 sec	Red 1	Red 2	Red 3	Green 4
	61 to 65 sec	Red 1	Red 2	Red 3	Yellow 4
Return to State 1	66 to 75 sec	Green 1	Red 2	Red 3	Red 4
and so on...	76 to 80 sec	Yellow 1	Red 2	Red 3	Red 4



All The above program setting according to the following front view of the PLC trainer, and all the definition according to table 1 in next page

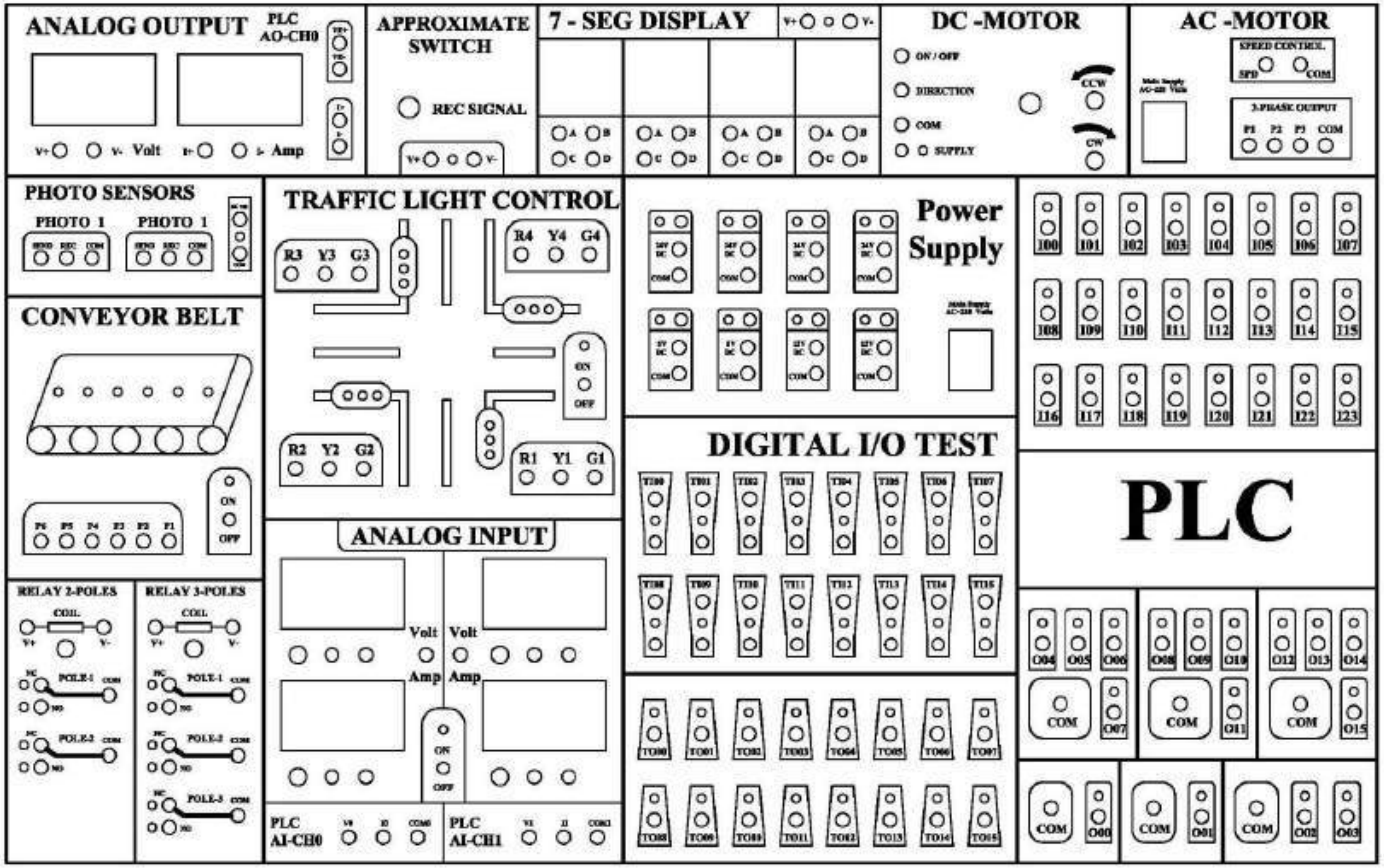


Table - 1- Description of symbols in PLC trainer

Label	Description	Label	Description
PLC Input & Output		**Approximate Switch**	
I00... I23	24 Digital Inputs	REC SIGNAL	Receive signal of Approximate sensor
O00...O15	16 Digital Output	**7-Seg Display**	
COM	Supply voltage of Digital Output	A B C D	4 Digital input (Hex code) of 1st , 2nd , 3rd & 4th (7-segmenat displays)
V0 com0	CH0 - Analog Voltage Input	**DC Motor**	
V1 com1	CH1 - Analog Voltage Input	ON / OFF	Supply voltage for DC Motor
I0 com0	CH0 – Analog Current Input	Direction	Change direction of DC Motor using logic (0 or 1)
I1 com1	CH1 - Analog Current Input	CW , CCW	LED Indication for Clockwise & counterclockwise
I+ I-	CH0 Analog Current Output	**Photo Sensors**	
V+ V-	CH0 Analog Voltage Output	SEND	Supply voltage for Sender Photo sensor 1 & 2
Digital I/O Test		REC	Feedback signal from receiver photo sensor 1 &2
TI00...TI15	16ch external digital input	COM	Common voltage
TO00...TO15	16ch external digital output	**Relay 2&3 Poles**	
Traffic Light Control		V+ V-	Supply Voltage for relay coil
R1 Y1 G1 R2 Y2 G2 R3 Y3 G3 R4 Y4 G4	Red , Yellow , Green (3 input of 1st , 2nd , 3rd & 4th traffic)	NC	Normally Closed of relay pole
		NO	Normally Open of relay pole
		COM	Pole Common (supply voltage for output)
Conveyor Belt		**Analog Output**	
P1 ….. P6	6 Digital Inputs for Sequence process	I+ I-	Tested point for reading current
		V+ V-	Tested point for reading voltage