

## Experiment 12

**Objective:** PLC Familiarization (Hardware & Programming)

### What is PLC?

A *programmable logic controller* (PLC) is a special form of microprocessor based controller that uses a *programmable memory* to store instructions and to implement functions such as logic, sequencing, timing, counting and arithmetic in order to control machines and processes.

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

1. Rugged and designed to withstand vibrations, temperature, humidity and noise.
2. Have interfacing for inputs and outputs already inside the controller.
3. Are easily programmed and have an easily understood programming language which is primarily concerned with logic and switching operations.

### PLC Hardware:

Typically a PLC system has two main types; the packaged (Fixed) type is a commonly used for small PLCs and is self-contained unit supplied as an integral compact package complete with all units. The modular (Rack) type can be used for all sizes of PLCs. It has the various functional units, *the mix of which decided by the user*, packaged in individual modules which can be plugged into sockets in a base rack (chassis). See figure1.



a. Fixed PLC



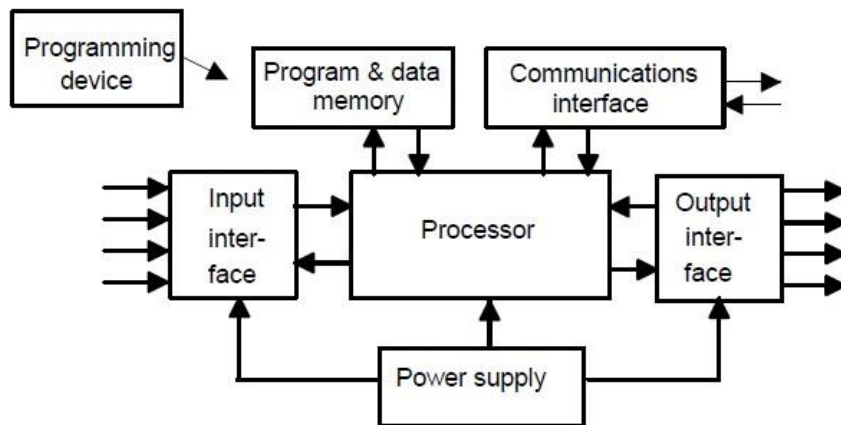
b. Compact PLC

Figure 1: PLC types.

The PLC system components are illustrated in figure2. And those are:

1. Processor unit (CPU)

2. Memory
3. Input/ Output section
4. Power supply unit
5. Programming device
6. Special devices( communications,.. etc)
7. Optional modules



*Figure2 : PLC system component*

The programming devices can be:

1. Hand-held device (HHT)
2. Desktop console
3. Computer

The IEC standard 1311 part 3 classifies programming methods into two categories:

1. Textual language, involves text and has two methods:
  - Instruction list
  - Structured text
2. Graphical language, involves graphical images and has two methods:
  - Ladder diagrams
  - Functional block diagrams

### **PLC Programming:**

The Simulate and programming is done in the PLC using *LOGO!Soft Comfort V6.0.17* software using the both graphical programming methods the Ladder diagram and the function blocks.

#### **1. The ladder diagram:**

The Ladder diagram is a way of programming PLC usually used by engineers who used to work with electrical circuit, because the similarity in the representation and symbols

The ladder logic uses graphic symbols similar to relay schematic circuit diagrams. Ladder diagram consists of two vertical lines representing the power rails. Circuits are

connected as horizontal lines between these two verticals. See figure3 which illustrates two ways of representing an electrical circuit. Figure3-b shows an electrical representation very close to the ladder representation.

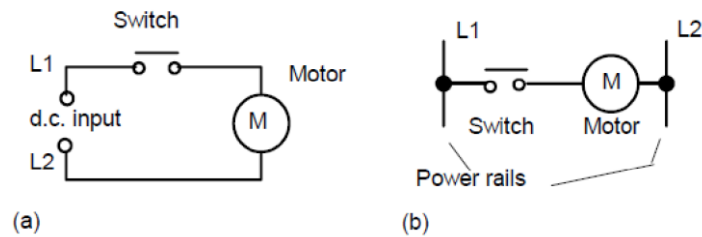


Figure3: Ways of representing an electrical circuit

The ladder diagram consists of two vertical lines representing the power rails, where circuits are connected as horizontal lines, i.e. the rungs of the ladder, between these two verticals contains the ladder coils, contactors and other components.

Ladder diagram features:

1. The power flows from left to right.
2. Output on right side cannot be connected directly with left side.
3. Contact cannot be placed on the right of output.
4. Each rung contains one output at least.
5. Each output can be used only once in the program.
6. A particular input a/o output can appear in more than one rung of a ladder.
7. The inputs a/o outputs are all identified by their addresses, the notation used to depend on the PLC manufacturer.

Ladder logic basic elements:

1. Contactor

Can be related to the hardware input ports of the PLC then to the input elements e.g. push buttons and sensors. They are two types:

- a. Normally Open (NO)

That mean when it activated it become close and allow power flow through it. See figure4.a

- b. Normally Close (NC)

That mean when it activated it become open and cut the power flow passing through it.see figure4.b

- c. Coils

Represent the output actuated elements and being connected the output hardware ports. See figure4. c.

- d. Boxes

Represent special functions in the program e.g. counters, timers... etc. see figure4.d

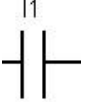
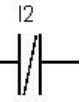

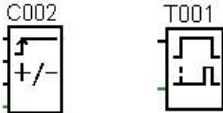
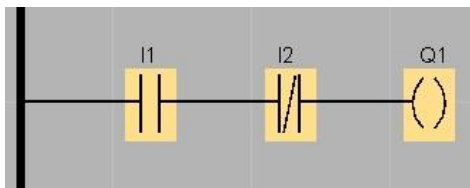
			
a. Normally open contactor	b. Normally Close contactor	c. coil	d. Ladder boxes counter and timer.

Figure 4. Ladder basic elements

Ladder programs operation:



*I1: normally open contactor (Input)*

*I2: normally close contactor (input)*

*Q1: coil (output)*

The power will start flowing from the right to until it reach the contactor with address (I1), that is NO when it being activated it become close and permit the power flow through it, and the power go through contactor address (I2), unless it being activated and turned to open, if it not the power will go to the output coil address (Q1) and by it being activated it will actuate the connected output.

For deep understanding, notes the timing diagrams in figure5.

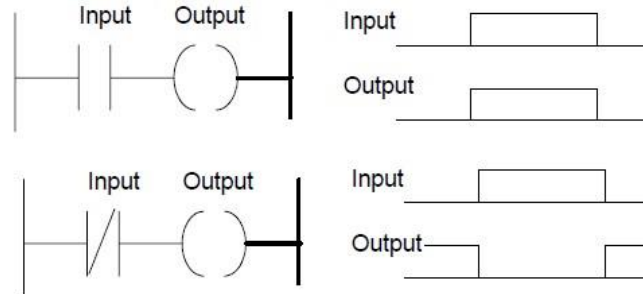


Figure5: ladder examples with timing diagram

**2. Function blocks diagram.**

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 representing 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 6 with the function name written in the box. It usually being preferred by engineer who familiar with the logic boxes of Boolean algebra.

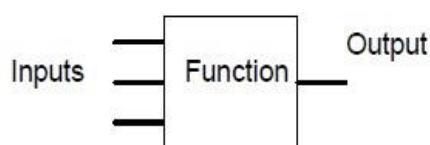


Figure6: Function Block.

Function blocks operation:

Figure 7 illustrates an AND operation being performed using the function block where two input used input A is NO and Input B is NC. The operation of this circuit was discussed previously in the ladder example.

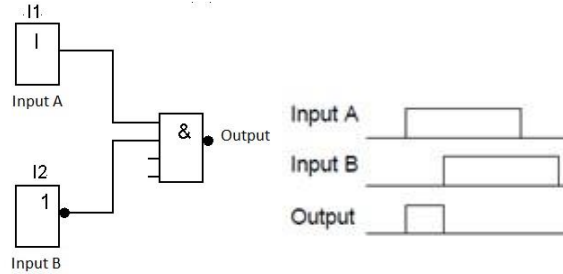


Figure7: AND operation using function diagram

Refer to the Function block manual to understand the function of each block.

Function block and ladder diagram examples:

