

AU-E - Practical Programmable Logic Controllers (PLCs) for Automation and Process Control



Availability: In Stock

Price: \$139.94

Ex Tax: \$127.22

Short Description

This manual is designed to benefit you with practical up-to-date information on the application of PLCs for the automation and process control of plants and factories. It is suitable for people who have little or no exposure to PLCs, but expect to become involved in some or all aspects of PLC installation. It aims to give practical advice from experts in the field, to assist you to correctly plan, program and install a PLC with a shorter learning curve and more confidence.

Description

This manual is designed to benefit you with practical up-to-date information on the application of PLCs for the automation and process control of plants and factories. It is suitable for people who have little or no exposure to PLCs, but expect to become involved in some or all aspects of PLC installation. It aims to give practical advice from experts in the field, to assist you to correctly plan, program and install a PLC with a shorter learning curve and more confidence.

The inevitable question is which PLC is being used. This manual focuses on the generic PLC and uses the open programming IEC 61131-3 standard. The manual includes contributions from a number of experts and will become a valuable reference in your work. The information advances from the basics to

challenge even the most experienced engineer in the industry today.

Table of Contents

Download Chapter List

[Table of Contents](#)

First Chapter

Introduction to the PLC

1. Introduction to the PLC

In this chapter, we will learn the following:

- Introduction to the PLC
- Basic block diagram of the PLC
- Size of the PLC system
- Components of the PLC system
- PLC and process interaction
- Number system and codes

1.1 Introduction to the PLC

In the past, processes were controlled manually, which was a very tedious job. During the early years of control, man constructed hardware relay wiring to control the same process. However, relays could not meet all the needs of modern times, and a faster solution was required. Simply, when a change of control logic was required, the entire hardware wiring needed to be changed. This was time consuming as well as tiresome.

After a lot of toil, man finally designed the PLC. He overcame all the constraints and attained flexibility to carry out the necessary modifications.

“PLC” means “Programmable Logic Controller”. The word “Programmable” differentiates it from the conventional hard-wired relay logic. It can be easily programmed or changed as per the application’s requirement. The PLC also surpassed the hazard of changing the wiring.

The PLC as a unit consists of a processor to execute the control action on the field data provided by input and output modules.

In a programming device, the PLC control logic is first developed and then transferred to the PLC.

1.1.1 What can a PLC do?

- It can perform relay-switching tasks.
- It can conduct counting, calculation and comparison of analog process values.
- It offers flexibility to modify the control logic, whenever required, in the shortest time.
- It responds to the changes in process parameters within fraction of seconds.
- It improves the overall control system reliability.
- It is cost effective for controlling complex systems.
- Trouble-shooting becomes simpler and faster.
- An operator can easily interact with the process with the help of the HMI (Human-Machine Interface) computer.

1. Basic block diagram of the PLC

Figure 1.1 shows the basic block diagram of a common PLC system.

Figure 1.1

Block diagram of a PLC

As shown in the above figure, you will find the heart of the “PLC” in the center, i.e., the Processor or CPU (Central Processing Unit).

- The CPU regulates PLC program, data storage, and data exchange with I/O modules.

“I/O Modules” are the second important and basic components of any PLC system.

- Input and output modules are the media for data exchange between field devices and CPU. It tells CPU the exact status of field devices and also acts as a tool to control them.

The programming device is the third significant component.

- A programming device is a computer loaded with programming software, which allows a user to create, transfer and make changes in the PLC software.

Memory is the fourth notable component.

- Memory provides the storage media for PLC program as well as different data.

1. **Size of the PLC system**

PLCs are classified on basis of their sizes:

- A small system is one with less than 500 analog and digital I/Os.
- A medium system has I/Os ranging from 500 to 5,000.
- A system with over 5,000 I/Os is considered large.

1. **Components of the PLC system**

Figure 1.2 illustrates a sample PLC system and its components.

Figure 1.2

The main/chief components of a PLC

This is the actual system. What does it look like? Does it seem difficult to understand? Let us compare Fig. 1.2 component by component with Fig. 1.3, which is its simplified version.

Figure 1.3

Simplified component configuration of a PLC

1) CPU or processor

The main processor (Central Processing Unit or CPU) is a microprocessor-based system that executes the control program after reading the status of field inputs and then sends commands to field outputs. It is easy to perform arithmetic functions, manipulate data and calculate Boolean logic.

The PLC's memory contains the manufacturer's operating system and housekeeping functions. It also has program written by the user and data stored by the user related to the process or equipment being controlled.

In Fig. 1.3, the processor module is plugged into the first slot of the central or local rack, which is the normal practice.

The CPU card is connected with I/O modules through back-plane connections inside the rack or chassis. Hence, it is possible for the CPU to read the status of all input modules through the data bus. This is called local I/O chassis.

If the I/Os are located at remote places, the CPU accesses them through a remote I/O chassis. In such a case, the remote chassis will have a remote I/O communication module. It collects the data from I/O modules and sends it to the CPU through an I/O rack communication link. This is particularly useful when the process is divided up, and located in remote parts.

The CPU card has ports for communicating with the programming device as well as the operator's station.

2) I/O section

The I/O section consists of a rack and individual I/O modules, which are plugged into the rack and a DC power supply. A standard approach is to connect to the main processor rack with communication cables to a series of other I/O racks.

I/O modules act as "Real Data Interface" between field and PLC CPU. The PLC knows the real status of field devices, and controls the field devices by means of to the relevant I/O cards.

Various I/O modules are available. Each one of these will be discussed in more

detail in the following sections.

3) Programming device

A CPU card can be connected with a programming device through a communication link via a programming port on the CPU. Thus, it is possible to transfer programs from the programming device to the CPU, monitor the CPU's program online and make necessary changes in the CPU's program.

More details of a programming device will be discussed in the next chapter.

4) Operating station

An operating station is commonly used to provide an "Operating Window" to the process. It is usually a separate device (generally a PC), loaded with HMI (Human Machine Software).

This operating station can change any process set point, observe all process parameters, process alarms, etc.

1. PLC and process interaction

It is very interesting to see how the interaction between the PLC and the process takes place through I/O modules. This section will explain the role of different I/O modules as a part of PLC and process interaction.

The field devices or field data are broadly classified as follows:

- Digital or discrete or on/off type field devices
- Analog or continuous type field devices

For the sake of simplicity, the first examples of digital type field devices, PLC and process interaction can be studied.

1.5.1 Digital I/O and PLC

Fig. 1.4 shows how different types of digital or discrete field devices are connected to the PLC through digital I/O modules.

Figure 1.4

Digital I/O process interaction

Discrete or digital inputs are the most common types of field inputs. Selector switches, pushbuttons, limit switches, temperature switches, level switches, flow switches, etc., are common examples. All types of switches that permit digital contact are included in this class.

Depending on the field device contacts, they are normally connected to 110VAC, 230V AC and, 24VDC types of digital input cards installed in PLC rack. It is assumed to be 24VDC in Fig. 1.4.

Solenoid valves, relays and auxiliary contactors are discrete or digital output devices commonly used in the field. Depending on the field devices 110VAC, 230VAC and 24VDC types of digital output cards are installed in the PLC rack. It is assumed to be 24VDC in Fig. 1.4.

Depending on the state of field devices, i.e., Logic 1 or Logic 0 (True/False) the corresponding voltage signal will be received at a digital input module.

The CPU collects the status of 'Logic 0 or Logic 1 status' from I/P module for program execution.

At the end of program execution, the CPU will release the On/Off command to the digital output module in form of Logic 0 or Logic 1.

The digital output module passes this status on to field devices in form of voltage so that they will be turned on/off depending on the state of the output.

1.5.2 Analog I/O and PLC

Figure 1.5 shows how different types of analog or continuous types of field devices are connected with the PLC through analog I/O modules.

Figure 1.5

Analog I/O process interaction

Analog or continuous devices are generally used for getting feedback of the process parameter control. For example, temperature transducers (RTDs, thermocouples), level, pressure, flow, etc., transmitters.

Basically, all types of transducer devices giving continuous signal are included in this class.

Depending on field transducer devices, they are normally connected to 0-20 or 4-20 mA DC, 0-10VDC, RTDs, thermocouples, etc., type of analog input card, installed in the PLC rack.

Analog output actuators, commonly used in the field, include continuous actuators, I/P converter for valves, reference for drives, etc.

The corresponding signal received at the analog input module, depends on the value of field devices (i.e., either 4-20mA or some continuous value) and varies from a minimum to a maximum over the entire calibrated range.. This signal is converted by the input module, into a count using ADC (analog-to-digital converter).

The CPU collects the same count from an analog I/P module for program execution.

At the end of program execution, the CPU will once again release a count proportionate to the control value to the analog output module.

The analog output module passes the same count (either as a 4-20mA or some continuous signal) using DAC (digital-to-analog converter) to the field actuating devices. The 4-20mA signal actuator will either open/close the valve depending

on it's setup.

1. Number systems and codes

The commonly used number systems can be listed as:

- Decimal number system
- Binary number system
- Octal number system
- Hexadecimal (Hex) number system

One may also encounter BCD (binary coded decimal) in codes.

The following definitions / terms are commonly encountered:

- **Bit:** A single binary digit that can have either value 0 or 1.
- **Base:** This denotes the total number of digits used by the number system.
- **LSB (Least Significant Bit):** This is the bit that represents the smallest value.
- **MSB (Most Significant Bit):** This is the bit that represents the largest value.
- **Byte:** A group of 8 bits.
- **Nibble:** A group of 4 bits.
- **Word:** A group of 16 bits

1.6.1 Decimal number system

The decimal number system is the most common number system (and one which we should all have learnt in school!).

The decimal number system has a base of '10'. Base '10' means it uses ten unique numbers (0 to 9) for the entire number system. A specific weight value is allocated to each digit from the right to the left. Therefore, the value of a decimal number depends on the digit as well as its location.

For example,

If we take '2413' as a decimal number, its weight can be shown as,

$$\begin{array}{cccc} 2 & 4 & 1 & 3 \\ 3 \times 10^{\text{exp. } 0} & = & 3 \times 1 & = 3 \end{array}$$

$$1 \times 10^{\text{exp. } 1} = 1 \times 10 = 10$$

$$4 \times 10^{\text{exp. } 2} = 4 \times 100 = 400$$

$$2 \times 10^{\text{exp. } 3} = 2 \times 1000 = 2000$$

$$= 2413 \text{ (Decimal)}$$

1.6.2 Binary number system

The most important base for the microprocessor is base 2. Normally, it is also known as 'Binary', since binary means only two values. The binary system uses only two digits – zero (0) and one (1) – to represent all numbers.

Two states exist for any digital device – one as "On" state or Logic '1' and other as "Off" state or Logic "0".

As the base was '10' in the decimal system, here, the base is '2' in the binary system.

The weight of the digits is calculated in terms of base '2' like for example,

$$\begin{array}{cccccc} 1 & 0 & 0 & 1 & 1 \\ 1 \times 2^{\text{exp. } 0} & = & 1 \times 1 & = & 1 \end{array}$$

$$1 \times 2^{\text{exp. } .1} = 1 \times 2 = 2$$

$$0 \times 2^{\text{exp. } .2} = 0 \times 4 = 0$$

$$0 \times 2^{\text{exp. } .3} = 0 \times 8 = 0$$

$$1 \times 2^{\text{exp. } .4} = 1 \times 16 = 16$$

$$= 19 \text{ (Decimal)}$$

1.6.3 Octal number system

The base for this number system is '8'. That is why it is called 'Octal'.

The binary system uses eight digits – from zero (0) to seven (7) to represent all the numbers.

Octal Digit	0	1	2	3	4	5	6	7
Binary Equivalent.	000	001	010	011	100	101	110	111

Basically, it can be considered as a shorthand version of binary. Octal notation was used to provide an easy and more readable view of the binary data. But it is not frequently used, when compared to other number systems. For example, the octal number 24 is represented as,

$$2 \quad 4 \\ 4 \times 8^{\text{exp.}0} = 4 \times 1 = 4$$

$$2 \times 8^{\text{exp.}1} = 2 \times 8 = 16$$

$$= 20 \text{ (Decimal)}$$

1.6.4 Hexadecimal (Hex) number system

The base of this number system is '16'. It provides a shorter notation of the numbers, compared octal system. It is very popular and was introduced by IBM. The hexadecimal number system uses 16 digits. These are:

Numbers '0' to '9', and it uses letters A, B, C, D, E, and F to represent the decimal equivalents of 10, 11, 12, 13, 14 and 15.

Hex Digit	0	1	2	3	4	5	6	7
Binary Equivalent.	0000	0001	0010	0011	0100	0101	0110	0111

Hex Digit	8	9	A	B	C	D	E	F
Binary Equivalent.	1000	1001	1010	1011	1100	1101	1110	1111

As shown in above table, a hex digit is representing four binary digits (0000 to 1111).

For example, the hex number 2F is represented as,

$$0 \quad 0 \quad 2 \quad F$$

$$F \times 16^{\text{exp.}0} = F \times 1 = 15$$

$$2 \times 16^{\text{exp.}1} = 2 \times 16 = 32$$

$$= 47 \text{ (Decimal)}$$

1.6.5 Conversion of numbers

Table 1.1 gives an overview of all number systems we have seen so far. Note that all number systems start with digit '0'.

Table 1.1

Number formats

Decimal	Binary	Octal	Hex
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

1.6.6 BCD (Binary Coded Decimal)

Binary coded decimal uses binary numbers in coded format to represent the decimal numbers (unlike the normal way; discussed in the earlier section).

It is also called as 8421 BCD code. It basically employs four binary bits, with the weights 1, 2, 4 and 8 assigned to it. This is illustrated in the table given below:

Decimal Digit	0	1	2	3	4	5	6	7
BCD Equivalent.	0000	0001	0010	0011	0100	0101	0110	0111

Decimal Digit	8	9
BCD Equivalent.	1000	1001

For example, the decimal number 79 is represented in BCD format as,

0 1 1 1

1 0 0 1

'7'

'9'

= 79 (*Decimal*)

Each decimal number is replaced by its BCD equivalent, which is a four-digit binary number. The BCD number format is useful for the PLC while handling a large number of input and output data.

Nowadays, PLC's possess a certain number format conversion programming instructions. One can easily convert numbers from one format to another with the help of these instructions.

Should you wish to do some form of conversion, program the PLC with the necessary code...and it will be done.