What is a PLC?

PLC stands for programmable logic controller. A PLC is a programmable computing device that is used to manage electromechanical processes, usually in the industrial niche. A PLC is sometimes referred to as an industrial PC, a term that describes a PLC's main function as a specialized industrial computing machine.

PLCs monitor the state of an input device such as signals from a light switch, and make decisions about the next state of an output device, for example switching a light on or off.

PLCs are also used to transfer information from devices at factories or from offsite locations to centralized applications, often running on PCs. PLCs are commonly used for device monitoring and reporting, to diagnose faults in hardware devices like industrial machines and tools, and to effect device events.

A **PROGRAMMABLE LOGIC CONTROLLER** (PLC) is an industrial computer control system that continuously monitors the state of input devices and makes decisions based upon a custom program to control the state of output devices.

Almost any production line, machine function, or process can be greatly enhanced using this type of control system. However, the biggest benefit in using a PLC is the ability to change and replicate the operation or process while collecting and communicating vital information.

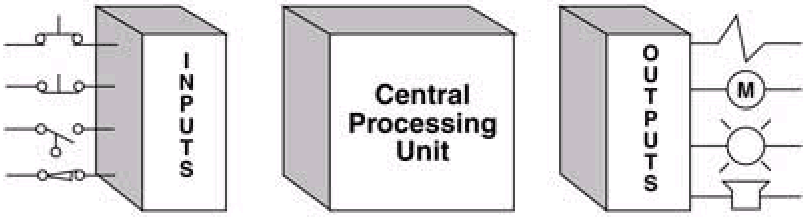
Another advantage of a PLC system is that it is modular. That is, you can mix and match the types of Input and Output devices to best suit your application.

History of PLCs

The first Programmable Logic Controllers were designed and developed by Modicon as a relay re-placer for GM and Landis.

- These controllers eliminated the need for rewiring and adding additional hardware for each new configuration of logic.
- The new system drastically increased the functionality of the controls while reducing the cabinet space that housed the logic.
- The first PLC, model 084, was invented by Dick Morley in 1969
- The first commercial successful PLC, the 184, was introduced in 1973 and was designed by Michael Greenberg.

What Is Inside A PLC?



The Central Processing Unit, the CPU, contains an internal program that tells the PLC how to perform the following functions:

- Execute the Control Instructions contained in the User's Programs. This program is stored in "nonvolatile" memory, meaning that the program will not be lost if power is removed
- Communicate with other devices, which can include I/O Devices, Programming Devices, Networks, and even other PLCs.
- Perform Housekeeping activities such as Communications, Internal Diagnostics, etc.

How Does A PLC Operate?

There are four basic steps in the operation of all PLCs; Input Scan, Program Scan, Output Scan, and Housekeeping. These steps continually take place in a repeating loop.

Four Steps In The PLC Operations

1.) Input Scan

- Detects the state of all input devices that are connected to the PLC

2.) Program Scan

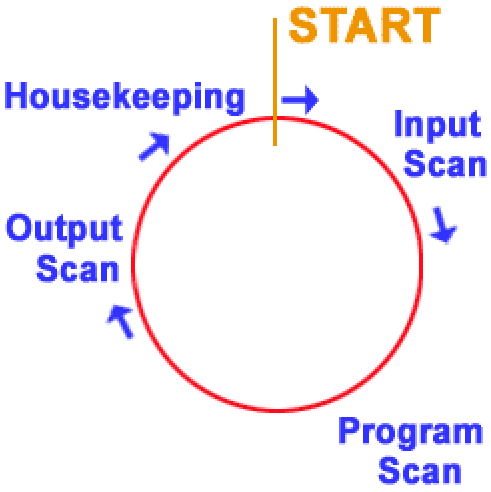
- Executes the user created program logic

3.) Output Scan

- Energizes or de-energize all output devices that are connected to the PLC.

4.) Housekeeping

- This step includes communications with programming terminals, internal diagnostics, etc...

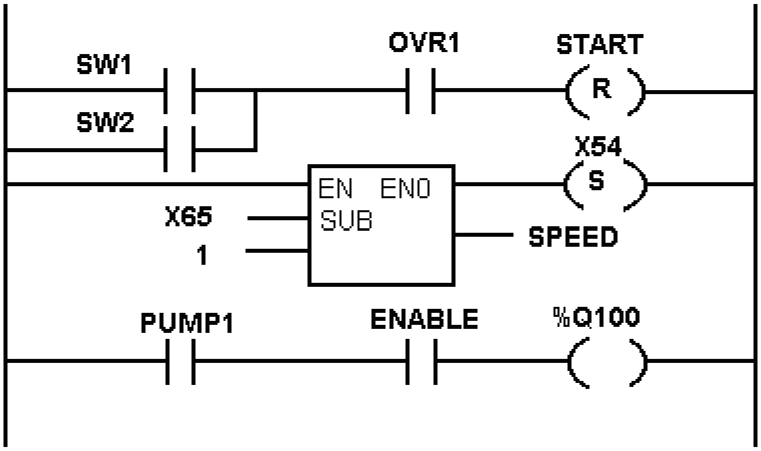


These steps are continually processed in a loop.

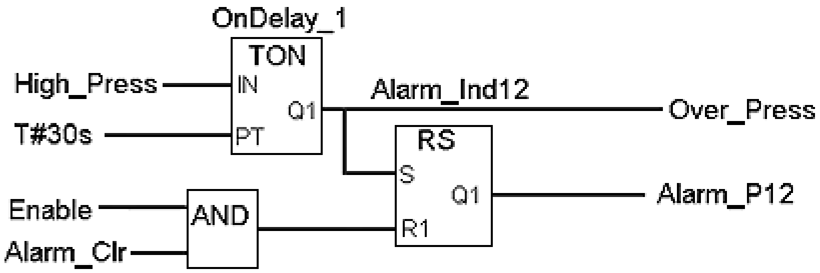
What Programming Language Is Used To Program A PLC?

While Ladder Logic is the most commonly used PLC programming language, it is not the only one. The following table lists of some of languages that are used to program a PLC.

Ladder Diagram (LD) Traditional ladder logic is graphical programming language. Initially programmed with simple contacts that simulated the opening and closing of relays, Ladder Logic programming has been expanded to include such functions as counters, timers, shift registers, and math operations.



Function Block Diagram (FBD) - A graphical language for depicting signal and data flows through re-usable function blocks. FBD is very useful for expressing the interconnection of control system algorithms and logic.



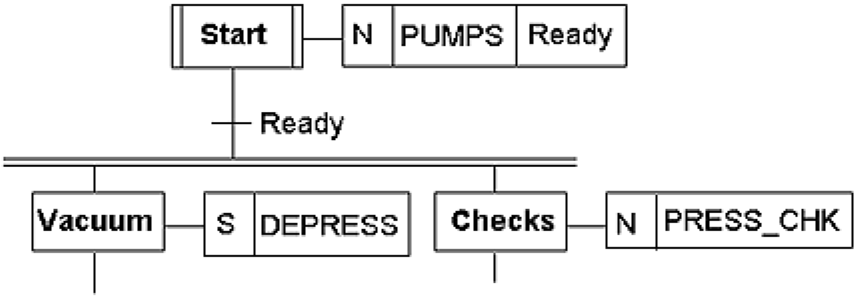
Structured Text (ST) – A high level text language that encourages structured programming. It has a language structure (syntax) that strongly resembles PASCAL and supports a wide range of standard functions and operators. For example;

```
If Speed1 > 100.0 then
   Flow_Rate: = 50.0 + Offset_A1;
Else
   Flow_Rate: = 100.0; Steam: = ON
End_If;
```

Instruction List (IL): A low level "assembler like" language that is based on similar instructions list languages found in a wide range of today's PLCs.

```
LD          R1
MPC         RESET
LD          PRESS_1
ST          MAX_PRESS
RESET:      LD   0
ST          A_X43
```

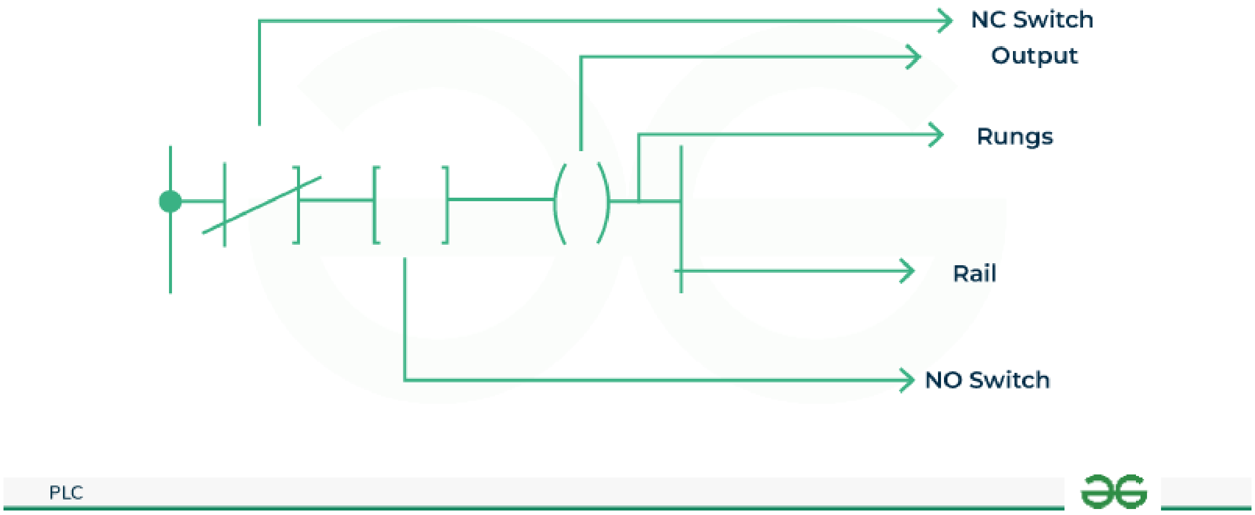
Sequential Function Chart (SFC) A method of programming complex control systems at a more highly structured level. A SFC program is an overview of the control system, in which the basic building blocks are entire program files. Each program file is created using one of the other types of programming languages. The SFC approach coordinates large, complicated programming tasks into smaller, more manageable tasks.



Programmable Logic Controller Programming

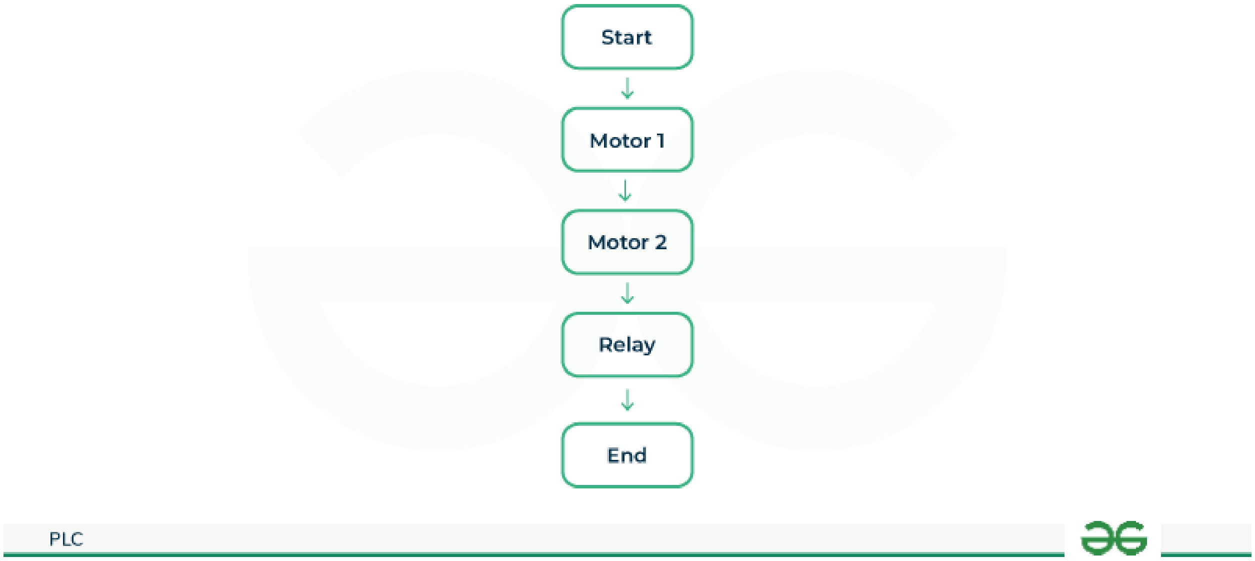
PLC is a programmable controller. We can create logic through programming to control a system. Their are several types of programming languages or strategies that can be followed. Some of them are :-

- **Ladder Logic :** Ladder logic is also known as Ladder diagram. It is basically a graphical representation of relay logic circuit, which consists two vertical parallel lines connected by horizontal lines. This lines are called rail and rungs. the vertical line works to supply power, and the vertical line is responsible for creating logic.



Ladder Logic

- **Function Block Diagram :** Function Block Diagram(FBD) in PLC programming is a graphical representation of control logic. It uses function blocks to depict operations, such as comparisons or timers, with inputs and outputs conducted by lines. Contacts and coils symbolize input conditions and output actions. Branches and junctions control logic flow. For instance, motor control system FBD might include a start or stop button, a motor and a control relay illustrating the interconnection of these components in a visual and intuitive format for programming complex control algorithms.



Function-Block-Diagram

Examples of PLC Programming

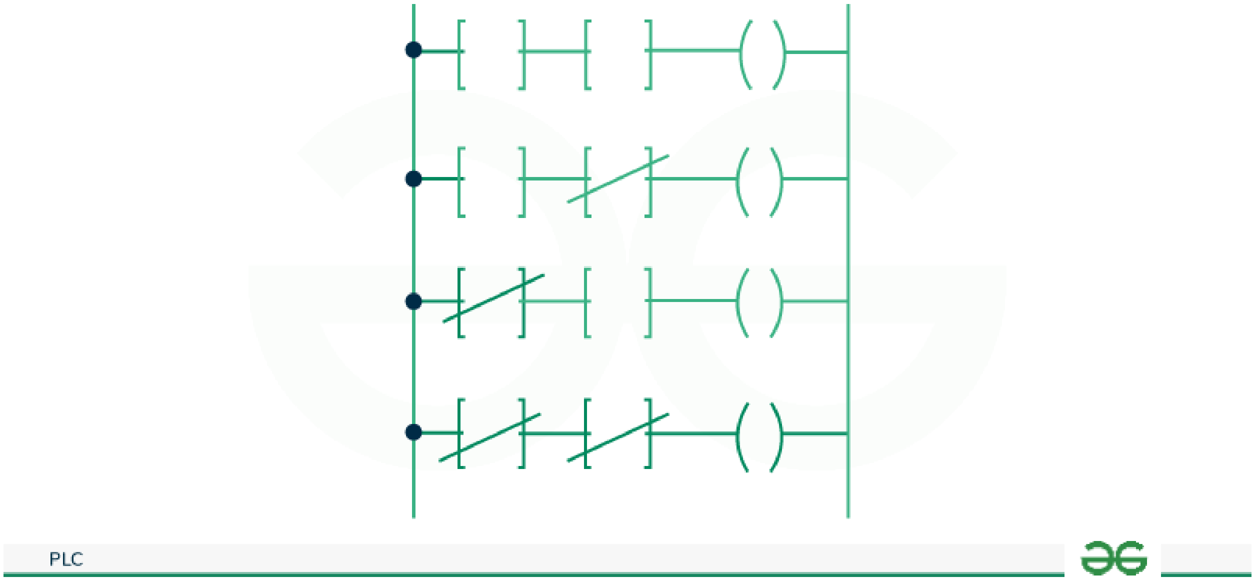
Lets program an AND logic for PLC using ladder logic :

Truth Table

A	B	Output(A.B)
0	0	0
0	1	0
1	0	0
1	1	1

Now from this truth table we will try to program it using ladder logic. In ladder logic there are two switches, NO(normally open) means it will not pass any signal and NC(normally closed) means it will pass signals. And there is a output terminal. Switches are connected with a vertical line, which is called rung, and those rungs are connected to the supply line, which is vertically, call rails. Rungs are also connected to output terminals to get output signals.

Here 0 means no current is passing, means we will use a NO switch for exact 0, and 1 means current or signal is passing, we will use a NC switch for that. and will se what is the output, and will check it with the truth table.



Ladder-logic-diagram-of-AND-gate

In the first three horizontal line, the output is 0, means no signal or current is coming, because at least one of the switch is NO. In the last rung the output is 1, means it passes current or signal. This is how we generally program a PLC.

What Do I Need To Consider When Choosing A PLC?

There are many PLC systems on the market today. Other than cost, you must consider the following when deciding which one will best suit the needs of your application.

- Will the system be powered by AC or DC voltage?
- Does the PLC have enough memory to run my user program?
- Does the system run fast enough to meet my application's requirements?
- What type of software is used to program the PLC?
- Will the PLC be able to manage the number of inputs and outputs that my application requires?
- If required by your application, can the PLC handle analog inputs and outputs, or maybe a combination of both analog and discrete inputs and outputs?

- How am I going to communicate with my PLC?
- Do I need network connectivity and can it be added to my PLC?
- Will the system be located in one place or spread out over a large area?

PLC Acronyms

The following table shows a list of commonly used Acronyms that you see when researching or using your PLC.

ASCII	American Standard Code for Information Interchange
BCD	Binary Coded Decimal
CSA	Canadian Standards Association
DIO	Distributed I/O
EIA	Electronic Industries Association
EMI	ElectroMagnetic Interference
HMI	Human Machine Interface
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
I/O	Input(s) and/or Output(s)
ISO	International Standards Organization
LL	Ladder Logic
LSB	Least Significant Bit
MMI	Man Machine Interface
MODICON	Modular Digital Controller
MSB	Most Significant Bit
PID	Proportional Integral Derivative (feedback control)
RF	Radio Frequency
RIO	Remote I/O
RTU	Remote Terminal Unit

SCADA	Supervisory Control And Data Acquisition
TCP/IP	Transmission Control Protocol / Internet Protocol

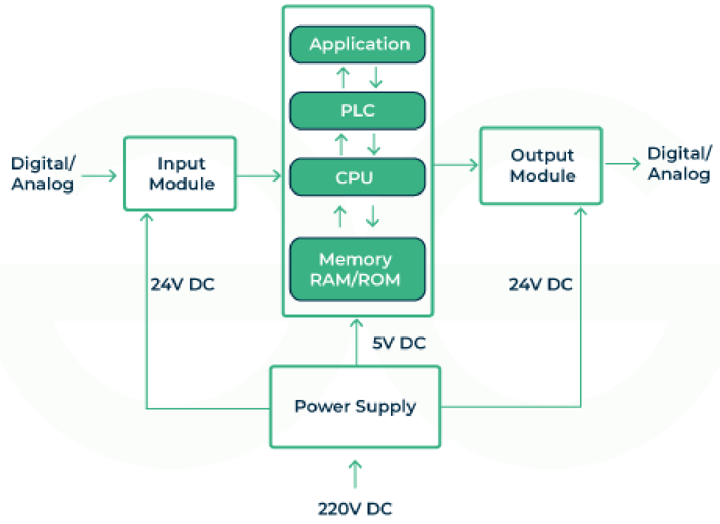
What is a relay logic system?

PLCs are the successors of relay logic systems, which are control systems that monitor and control low-level devices like switches, relays, timers, actuators, and motors. In relay logic systems, relays perform logic operations, known as relay logic, by activating and deactivating magnetic coils in electric circuits.

Relays can connect circuits that use different currents and voltages so, historically, they were ideal for controlling and coordinating different industrial automation devices.

The disadvantage of a relay logic system is that, like a switchboard, it is hard-wired and difficult to maintain due to the multiplicity of wires. The development of microcontrollers solved this problem by enabling relay logic processes to be coded programmatically and to be stored on a computer.

Basic Structure of PLC with Block Diagram



PLC

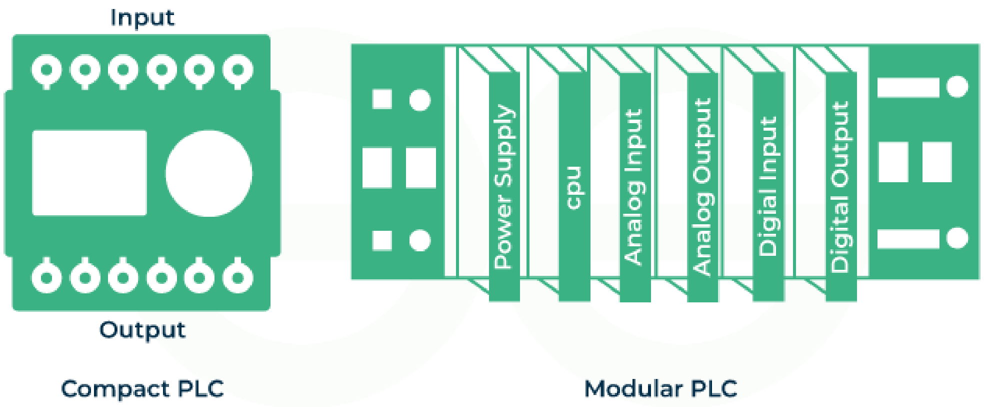
Types of PLC

PLCs can be classified into three main types based on their capabilities :

- **Compact PLC** : compact PLCs are small size and typically have a limited number of I/O points. They are designed for space is a critical factor, and the control requirements are relatively simple. Compact PLCs are often used in small-scale control applications where the number of devices to be controlled and monitored is limited. Examples include simple machines, standalone equipment, or processes with minimal complexity.

- **Modular PLC** : Modular PLCs are designed with s modular architecture, where the CPU, Power supply, and input-output modules are separate components. This modular designed provides flexibility in system configuration, making it easier to expand or modify the system based on changing requirements. Modular PLCs are suitable for applications that may grow or change over time. Industries with evolving control needs, such as manufacturing plants that may expand production lines.

- **Rack-Mount PLC** : This is a type of High end PLC which is specially designed for complex and demanding application. It has maximum number of I/O points also Rack-mount PLCs are modular and can easily expanded by adding additional modules to the rack.



PLC

Features of PLC

PLC comes with a wide range of features, which makes it versatile. The features are :-

- **Digital and analog I/O :** It can handle both digital and analog signals from the sensor and other input device and allows a wide range of input.
- **Scalability :** Many PLCs offer modulation in their setting by expanding the input/output number or CPUs to accomodate the changing requirements.
- **Communication interfaces :** PLCs support various communication devices and protocol like ethernet, modbus to devices like CPU monitor and control systems.
- **Programming Language :** It allows different programming languages like ladder logic, structured text, function block diagram etc. to the user.
- **Real-Time monitoring :** PLCs operate in real-time by continuously scanning and processing input signals to make rapid decisions and update output devices with minimal delay.
- **Mathematical Functions :** PLCs support mathematical functions and calculations. enabling users to perform computations within the control program.

Applications of PLC

- **Manufacturing Automation:** PLCs are extremely used in manufacturing industries for automating processes such as assembly lines, packaging, and material handling.
- **Process Control :** In industries like chemical, petrochemical, and pharmaceuticals, PLCs play a crucial role in controlling and monitoring complex process.
- **Water treatment and Distributions:** PLCs are employed in water treatment plants to control the purification process, monitor water quality, and manage the distribution of water in a network.
- **Food and Beverage Industry :** PLCs can be used in food processing plants for tasks like mixing, backing packaging, quality control etc.
- **HVAC System :** PLCs play vital role in heating, ventilation, ACs to managing temperature, humidity, air quality etc.

How does a PLC work?

PLC components

A typical PLC is connected to a power supply and consists of a central processing unit (CPU), a mounting rack, read-only memory (ROM), random access memory (RAM), input/output (I/O) modules, a power supply, and a programming device.

PLCs have a modular design; I/O and other specialized modules slide into a PLC rack. PLC modules are sometimes called cards.

Rack

A PLC rack can be compared to a car's chassis with which other components are connected. The connected components in a PLC rack are grouped into three sections: the CPU, multiple I/O modules, and the power source.

Power supply

The power supply converts alternating current (AC) to direct current (DC). DC is used by the CPU and I/O components.

CPU

As with a PC, the CPU is the brain of a PLC. The CPU has two operational modes: programming mode and run mode. In programming mode, the CPU downloads logic in the form of programming instructions created by a user on a PC. In run mode, the CPU executes the logic.

The CPU controls all PLC operations according to the programming instructions stored in memory. A control bus system transfers information to and from the CPU.

I/O modules

A PLC receives or senses data from input devices like proximity and photoelectric sensors, keyboards, level meters, timers, counters, console lights, electric motors, and temperature and pressure switches. The concept of sensing data refers to the nature of PLC input data which comes in the form of electronic signals.

Digital input cards handle discrete signals, for example on/off signals. Analog input cards convert voltage into numbers the CPU can understand.

PLC outputs include valves, starter motors, drives, actuators, solenoids, alarms, control relays, printers, and pumps.

Digital output cards turn devices on and off, for example a light. Analog output cards convert digital numbers to voltage, for example to drive machinery.

PLCs can make logical decisions and perform actions based on the input data they receive, for example processing input data and sending processed data to an output device. The processing of input data is executed by a programming device.

For instance, a temperature switch may monitor the temperature at a cooling plant and intermittently send this information via a PLC to a printer in a factory’s operation center.

Programming device

The programming device is usually a PC, console, or hand-held proprietary device. I/O modules direct input signals into a PLC's CPU where output signals are created. The format of output data is specified by an application program implemented on the programming device.

Memory and storage

ROM stores operating system data and drivers. RAM stores the status of, and details about, input and output data, and application programs.

Communication

I/O modules are responsible for transferring information between the PLC and communication networks.

To communicate with external devices, PLCs use Recommended Standard 232 (RS-232), which is a serial communication standard. RS-232 uses binary code to read and write data in the American Standard Code of Information Interchange (ASCII) format.

At the control level, PLCs communicate with field components at the physical level using a variety of communication protocols depending on the component. PLCs also use different communication protocols to communicate over networks and with wireless devices.

Modbus RTU is a serial communication protocol often used in industrial communication networks to transmit data over long distances. Serial communication protocols, however, lack the performance and speed of Ethernet protocols. Ethernet protocols used by PLCs include Ethernet TCP/IP, Modbus TCP/IP, and Profinet, which are used to connect to plant networks and to the internet. Proprietary protocols may be customized for proprietary devices. Universal Serial Bus (USB) protocols are used to connect to drivers and printers. For wireless devices, the Bluetooth protocol may be used.

PLC life cycle

A PLC performs four basic operational functions. First, it scans any connected input devices for their operational status. Second, it executes a program that determines what should happen with the input data. Third, it generates executed output data. Fourth, it uses an operational housekeeping function to perform internal diagnostics.

Human machine interface (HMI)

PLCs use a range of ports and communication protocols to connect to control applications like supervisory control and data acquisition (SCADA) systems.

Operators at industrial plants and line managers interact with a PLC in real time using an HMI. An HMI is the operator's dashboard, that is, the interface between a person and the PLC. An HMI allows an operator to coordinate, manage, and control industrial processes and devices.

Where is a PLC used?

PLCs are used for a wide variety of automated machine processes, for example controlling elevators or turning light switches on and off in smart buildings. PLCs are used to monitor security cameras and automated warning devices, traffic lights, and industrial processes like cutting glass and paper.

A PLC monitors run-time data, for example device uptime, operating temperature, and usage statistics. It also starts and stops processes and generates notifications if a machine malfunctions.

PLCs are used in product assembly, packaging, motion control, batch control, machine diagnostics and testing, and robotics processes.

How is a PLC programmed?

Ladder logic is commonly used to program PLCs but other languages are also used, for example function block diagram, structured text, sequential function chart, and instruction list.

Ladder logic uses graphical diagrams based on relay circuit hardware to express and specify the logical structure of processes in PLC software applications. Ladder logic code looks like an electrical schematic.

Advantages of PLCs

The advantages of PLCs are primarily the benefits attained by doing away with traditional hard-wired logic control systems. Compared to traditional logic control systems, PLCs are easier to build, install, maintain, and modify.

PLC components do not require complex wiring because the control logic is implemented by the software. Modifications and updates can easily be implemented by uploading a new ladder logic program. Modifications can be done remotely instead of manually as is the case with relay logic systems.

Ladder logic coding allows the fast development of complex logical expressions. Programming PLCs are relatively simple and because the design of a PLC system is modular, they can be installed in different physical setups and are easily scalable with multiple different input and output devices.

PLCs are specialized devices built to withstand harsh environmental conditions like cold and extreme heat, and dusty and humid conditions in factories.

A PLC is embedded with a real-time operation system that has certain housekeeping functions but does not require the supplementary utilities PCs use like antivirus software or registry cleaners.

Being microprocessor based, PLC execution times are faster than relay logic control systems, which have multiple mechanical parts.

Disadvantages of PLCs

Complex programming code is often better managed by traditional PCs than PLCs because they are task driven rather than event driven.

Although ruggedly reliable, systems that use PLCs usually need external monitors to display data in a user-friendly way.

When they break down, PLCs may require extensive troubleshooting by PLC specialists.

What is the difference between a PC and a PLC?

A PLC is often described as a "ruggedized digital computer" but there are some key differences.

While a PC is required to run a PLC, a PC does not need a PLC to function. This is because a PLC is run using a microcontroller while a PC runs a microprocessor.

A PLC uses a scan-based method to execute code while PCs use an event-based method to execute code.

PLC inputs are signals rather than the data fed into from PC drives. Often data from a PLC is collected by a PC and used to trigger work orders, reports, and notifications.

A PLC's operating system is designed to carry out control tasks and so, unlike with PCs, they do not generally use antivirus or registry cleaning utilities.

PLCs are programmed using proprietary vendor languages or ladder logic. PCs are programmed using high-level languages like Java or C++.

PLCs are not immune to cybersecurity attacks and malware but reported instances are fewer than PCs. The reason for this is that PLCs have limited functionality compared to PCs.

Small PLC systems are usually cheaper to implement than a similar use case for a PC but large PLC systems required for complex logic operations can be expensive. Being more suited to harsh environmental conditions, PLCs can be cheaper to physically maintain.

What is the difference between relay logic and ladder logic?

Control circuits in relay logic systems need to be hard-wired for every single control function. Relay logic relies on performing logic and control functions based on diagrams of physical electronic circuits.

Using ladder logic, logic and control functions are performed using a programmable microprocessor-based PLC. The PLC stores the ladder logic program in memory. Unlike hard-wired control systems, PLCs can be reprogrammed, saving time and money as expensive wires and cables do not need to be replaced. Only I/O devices in a PLC require hard wiring.

What is the difference between a microcontroller and a microprocessor?

The terms microcontroller and microprocessor are sometimes used interchangeably but there is a big difference between the two. A microprocessor chip only has a CPU. A microcontroller consists of a CPU, memory, and input/output (I/O) modules all embedded on one chip.

A microprocessor cannot operate independently of other components, like additional circuits and memory, I/O modules, and software.

A microprocessor is used when a lot of complex processing is required, for example for gaming and in laptops, and where a task is undefined. A microcontroller is used for repetitive, predefined tasks, for example in electronic equipment and industrial machines.

Microcontrollers are cheaper, smaller, and use less power than microprocessors.

FAQs on PLC

What is PLC ?

PLC or Programmable logic controller, is an industrial digital computer designed to control manufacturing process, machinery and other control system.

How PLCs are integrated with other automation systems ?

PLCs can be integrated with other systems, such as SCADA, through communication protocols like Modbus or OPC to provide a comprehensive and centralized control solution.

Can PLC programs be edited or modified ?

Yes. PLCs can be edited and modified easily, allowing for flexibility in adapting to changes in the manufacturing process.

What are the primary components of a PLC ?

A typical PLC system includes a CPU, memory, input and output modules, programming device and power supply.