

PLCs : A Simple Introduction for Absolute Beginners**1. What a PLC Actually Is**

A Programmable Logic Controller or PLC is simply a small industrial computer designed to run machines and processes. Where a normal office computer runs apps, a PLC runs **control logic**.

Its job is straightforward:

- **Look at what's happening** (inputs)
- **Decide what to do** (the program)
- **Make something happen** (outputs)

That's it. Every PLC on the planet, no matter the brand, follows this same pattern.

PLCs are used everywhere:

- factories
- water treatment plants
- oil & gas skids
- food production
- packaging lines
- HVAC systems
- conveyors
- pumps and motors

If something moves, measures, opens, closes, starts, or stops there's usually a PLC behind it.

2. Why PLCs Replaced Hard-Wired Control

Before PLCs existed, machines were controlled using:

- relays
- timers
- contactors
- miles of wiring

Changing how a machine behaved meant physically rewiring panels. It was slow, expensive, and prone to failure.

PLCs solved all of that.

Instead of rewiring, you just **change the program**. Instead of dozens of relays, you have **one CPU**. Instead of guessing what's wrong, you have **diagnostics**.

This is why PLCs became the standard across every industry.

3. The Core Building Blocks of Any PLC System

Every PLC system, regardless of brand, is built from the same basic parts.

Inputs

These are signals coming *into* the PLC from the real world.

Examples:

- pushbuttons
- switches
- sensors
- level switches
- pressure switches
- proximity sensors

Inputs tell the PLC what's happening.



Elite providers of Measurement and Control Solutions

Outputs

These are signals the PLC sends *out* to control equipment.

Examples:

- motors
- solenoid valves
- relays
- alarms
- lights

Outputs are how the PLC makes things happen.

The CPU

This is the “brain” of the PLC. It runs the program, processes inputs, updates outputs, and handles communication.

The Program

This is the logic you write:

- “If this happens, turn that on.”
- “If the level is high, stop the pump.”
- “If the temperature is too low, open the steam valve.”

The program is stored in the CPU and runs continuously.

4. Digital vs Analog Signals

PLCs deal with two types of signals.

Digital (On/Off)

These are simple true/false conditions.

Examples:

- a switch is pressed
- a valve is open
- a motor overload has tripped

Digital signals are either **1 (on)** or **0 (off)**.

Analog (Variable)

These are continuously changing values.

Examples:

- pressure
- temperature
- flow
- level
- speed

Analog signals let the PLC make decisions based on real measurements, not just on/off states.

5. How a PLC “Thinks” – The Scan Cycle

Every PLC runs in a loop called the **scan**.

The scan repeats thousands of times per second:

1. **Read inputs**
2. **Execute the program**
3. **Update outputs**
4. **Handle diagnostics and communication**

This loop never stops while the PLC is in RUN mode.

Understanding the scan is important because it explains why PLCs react so quickly and consistently.

6. A Simple Real-World Example

Imagine a pump controlled by two pushbuttons:

- **Start button** → tells the PLC to run the pump
- **Stop button** → tells the PLC to stop the pump

The PLC reads the buttons, runs the logic, and energises the output that drives the pump starter.

Even this simple example shows the full PLC cycle:

- Inputs: Start/Stop buttons
- Logic: “If start is pressed, turn pump on. If Stop is pressed, turn pump off.”
- Output: Pump motor starter

This is the foundation of all automation — just scaled up.

7. Why PLCs Matter

PLCs are used because they are:

- reliable
- fast
- easy to modify
- easy to troubleshoot
- flexible
- scalable

They allow machines to be controlled safely, consistently, and efficiently.

Whether you're working on a small machine or a large industrial plant, the principles are the same.

Want the Full Beginner's PLC Pack?

The full version expands this into a complete, practical guide for new technicians and engineers.

Inside the **Ultimate PLC Beginner Pack**, you'll get:

- deeper explanations of inputs, outputs, and CPU operation
- how digital and analog signals really work
- number systems (binary, hex, BCD) explained simply
- how PLC memory is structured
- how the scan cycle affects real-world behaviour
- common ladder logic instructions explained
- real examples of start/stop circuits, interlocks, and alarms
- troubleshooting basics
- typical PLC architectures used in industry
- common mistakes beginners make and how to avoid them



ECAS

Elite providers of Measurement and Control Solutions

If you're starting out in automation, this pack gives you the foundation you need to understand any PLC system.

Get the complete forty-nine-page version with real world examples, troubleshooting, logic, hardware, and more. [Click here to get instant access.](#)



ECAS

Elite providers of Measurement and Control Solutions

PLC Ultimate Beginner Pack

A Complete Introduction to Programmable Logic Controllers



Clear Explanations

Practical Examples

Real World Applications

Beginner Friendly Language

Created by

Easson Control and Automation Services Ltd (ECAS)