Chapter 1
PLC Basics

A **programmable logic controller** (**PLC**) or **programmable controller** is an industrial computer that has been ruggedized and adapted for the control of manufacturing processes, such as assembly lines, machines, robotic devices, or any activity that requires high reliability, ease of programming, and process fault diagnosis. A PLC is a microprocessor-based controller with multiple inputs and outputs. It uses a programmable memory to store instructions and carry out functions to control machines and processes.

Programmable Logic Controllers (PLCs) provide a range of benefits, which make them the preferred choice in modern industrial automation. Key advantages include:

- **Increased Reliability**: Once programmed and tested, a PLC can easily transfer its logic to other controllers, reducing the risk of wiring errors. The logic is internal, leading to reduced external hardwiring, and the reliability of solid-state components ensures long-term durability.
- **More Flexibility**: Programming a PLC is much simpler than hardwiring a control circuit. Users can modify input/output relationships by altering the software, making updates and changes much easier. Original equipment manufacturers (OEMs) can send new programs for system upgrades, and security measures can be implemented via key locks or passwords.
- **Lower Cost**: PLCs have replaced relay control logic in many applications due to the significant cost savings. For systems with more than a few control relays, PLCs are typically a cheaper solution, and relay control is becoming obsolete in all but power applications.
- **Communications Capability**: PLCs can communicate with other controllers and computer systems for functions like supervisory control, data gathering, process monitoring, and program uploads/downloads. This connectivity supports more integrated control systems.

- **Faster Response Time**: PLCs are designed for real-time, high-speed applications. They quickly respond to events in the field, such as triggering an operation based on sensor inputs in milliseconds, making them ideal for machines that process large volumes or require quick reactions.
- **Easier to Troubleshoot**: PLCs have built-in diagnostics and override features, making it simpler to detect and resolve software or hardware issues. Users can view control programs in real time to trace faults and correct them effectively.
- **Easier to Test Field Devices**: PLCs allow centralized testing of field devices. By wiring devices to a common PLC module, users can quickly check their operation without needing to visit each device's location, saving time in large systems.

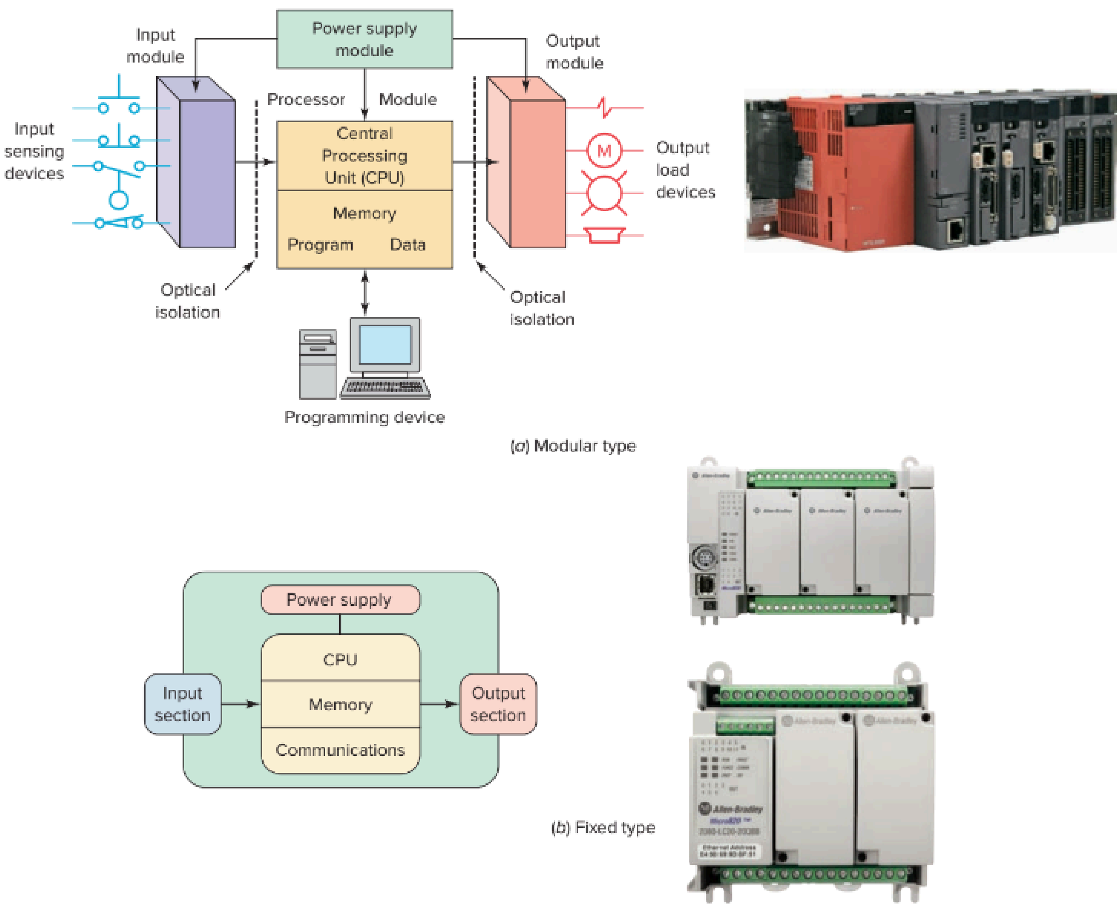


figure 1-1 Typical parts of a programmable logic controller.

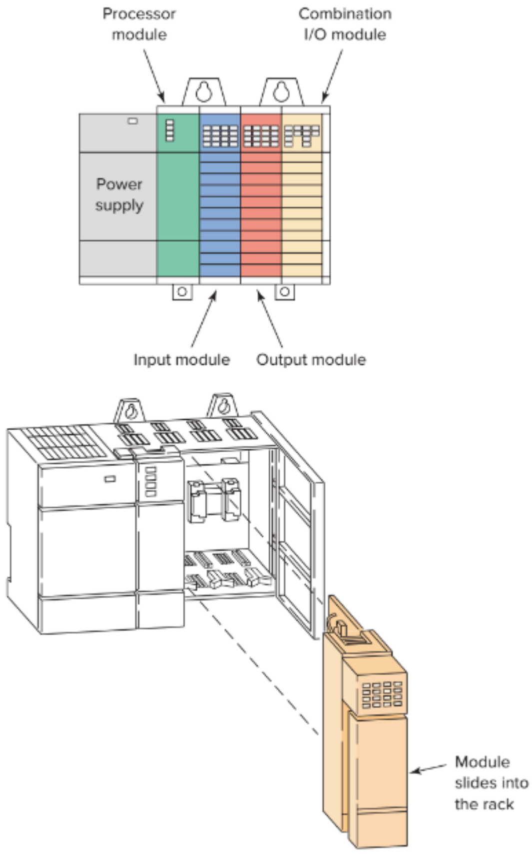


Figure 1-10 Modular I/O configuration.

Figure 1-12 Typical PLC processor modules.
Source: Image Courtesy of Rockwell Automation, Inc.

Figure 1-13 Typical PLC scan cycle.

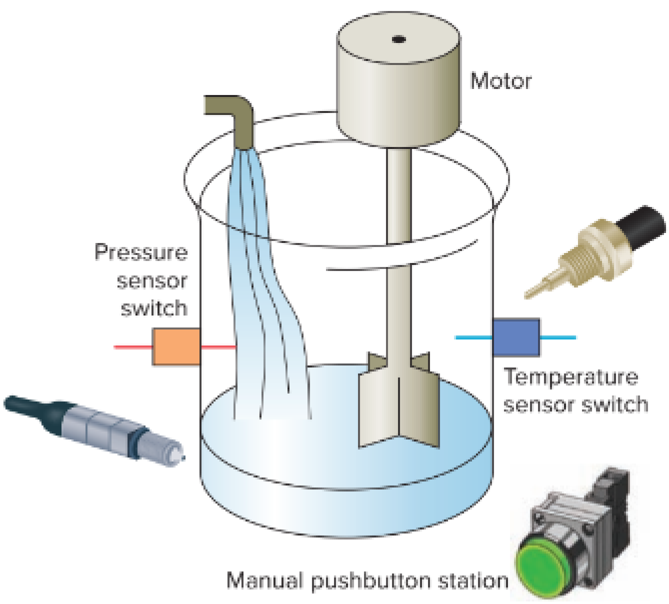
A typical **Programmable Logic Controller (PLC)** consists of several key components:

1. **Central Processing Unit (CPU)**: This is the "brain" of the PLC, responsible for executing logic, handling communication between modules, and storing program instructions and data.
2. **Input/Output (I/O) System**:
   - **Fixed I/O**: Found in smaller PLCs where the processor and I/O are packaged together, with a fixed number of input and output connections. This system is cost-effective but less flexible.
   - **Modular I/O**: Divided into compartments where separate modules (input/output) can be plugged into a rack, offering greater flexibility and options for expansion.
3. **Power Supply**: Provides DC power to the PLC modules. For larger systems, external power sources supply energy to field devices, while for smaller PLCs, the internal power supply may also power field devices.
4. **Programming Device**: A tool used to enter and modify the PLC's program. Commonly, personal computers with specialized software are used to program

the PLC. The programming is typically done in **Relay Ladder Logic (RLL)**, a graphical language based on relay control diagrams.

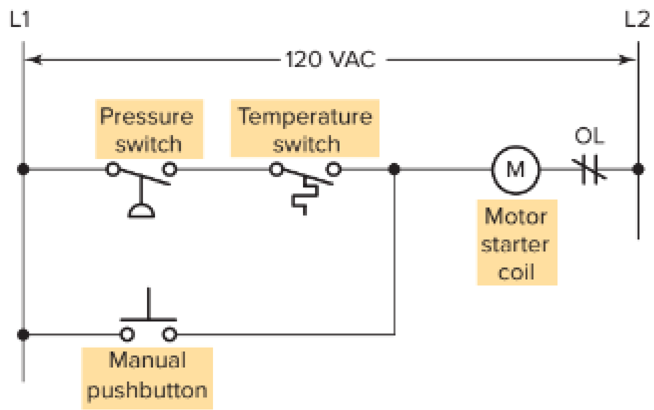
5. **Architecture**:
   - **Open Architecture**: Uses off-the-shelf components, allowing the PLC to connect with devices from different manufacturers.
   - **Closed Architecture**: Proprietary design, which limits compatibility with external systems but ensures optimized performance for specific applications.
6. **I/O System Connections**: The I/O system serves as an interface between the controller and field devices such as sensors (inputs) and actuators (outputs). These connections are typically hardwired, with optical isolators used to protect internal components.

The PLC operates in a **scan cycle**, where it reads inputs, executes the program, updates outputs, and performs diagnostic and communication tasks repeatedly. The use of programming devices (such as handheld devices or PCs) allows users to create, edit, and troubleshoot programs easily.



Automatic Control: The motor is automatically activated when both the **temperature** and **pressure** sensors detect values reaching their preset thresholds. These sensors are connected to the PLC, and when their respective contacts close, the PLC triggers the motor.

Manual Control: A separate **manual pushbutton** allows direct operation of the motor, bypassing the sensor conditions. This provides an additional layer of control for manual intervention when necessary.



In the example, the motor control problem can be solved using a **relay ladder diagram** or by employing a **Programmable Logic Controller (PLC)**. Here's how both methods work:

1. **Relay Method**:
   - The motor starter coil (M) is activated when both the **pressure** and **temperature switches** are closed, or when the **manual pushbutton** is pressed. This is done using physical relays and wiring.
2. **PLC Method**:
   - The same **input devices** (pressure switch, temperature switch, and manual pushbutton) are hardwired to the PLC's input module. The PLC uses its internal logic to replicate the control conditions:
     - It checks if both pressure and temperature switches are closed or if the manual button is pressed.
     - Based on this, it energizes the motor starter coil.
   - The PLC's addressing scheme determines where and how these inputs are connected, making it flexible for modification and easier to program or troubleshoot than traditional relay control.

- *A typical ladder logic program for this process is shown in figure 1-3.*

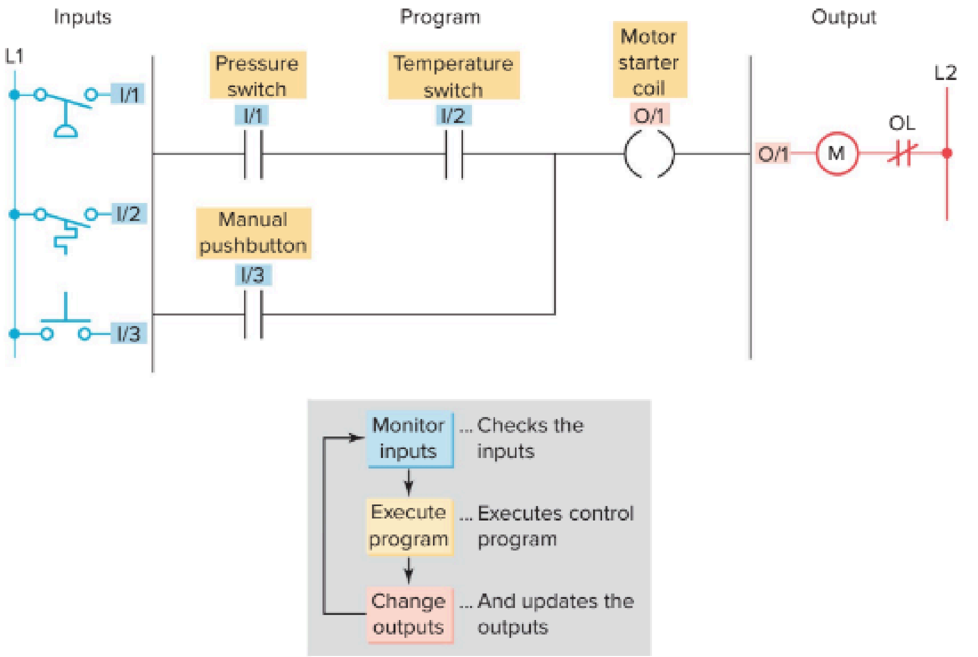


Figure 1-3

The **RUN operation** for a process control scheme follows this sequence of events:

1. **Input Examination**: The PLC checks the status of input devices (pressure switch, temperature switch, and pushbutton) and records their state in memory.
   - **Closed contact** = Logic 1
   - **Open contact** = Logic 0
2. **Ladder Diagram Evaluation**: The PLC evaluates the ladder logic diagram, assigning each internal contact an OPEN or CLOSED status based on the input's logic state (1 or 0).
3. **Logic Continuity**:
   - If there is logic continuity from left to right across a rung (i.e., all conditions are met), the output coil's memory location is set to logic 1, and the output module's interface contacts close.
   - If there is no continuity, the output coil's memory is set to logic 0, and the output module's contacts remain open.
4. **Scan Cycle**: The completion of one full cycle of this sequence is called a **scan**. The **scan time** measures how quickly the PLC can respond to changes.
5. **I/O Update**: While the output memory location is updated during the scan, the actual physical output (such as turning the motor on or off) is not updated until the end of the program scan during the **I/O scan**.

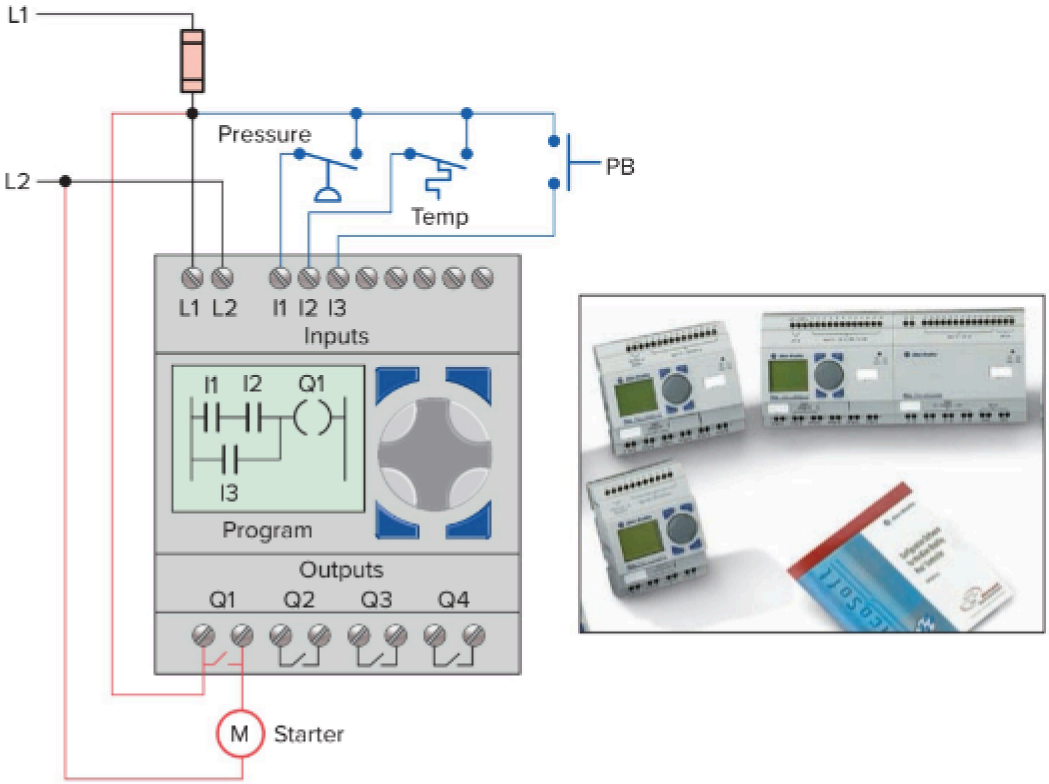


Figure 1-4

The Allen-Bradley **Pico controller** (with 8 inputs and 4 outputs) is used to control and monitor a process. The installation steps are as follows:

1. **Power Connection**: Fused power lines (correct voltage type and level) are connected to the controller's **L1** and **L2** terminals.
2. **Input Devices Wiring**:
    - **Pressure switch**, **temperature switch**, and **pushbutton** are hardwired between **L1** and the controller's input terminals **I1**, **I2**, and **I3**, respectively.
3. **Motor Starter Coil Wiring**: The motor starter coil is connected directly to **L2** and in series with the **Q1 relay output contacts** to **L1**.
4. **Program Entry**: The ladder logic program is entered using the **front keypad** and **LCD display** of the controller.
5. **Pico Programming Software**: Alternatively, programming software is available for the Pico controller, allowing users to create and test the program using a personal computer.

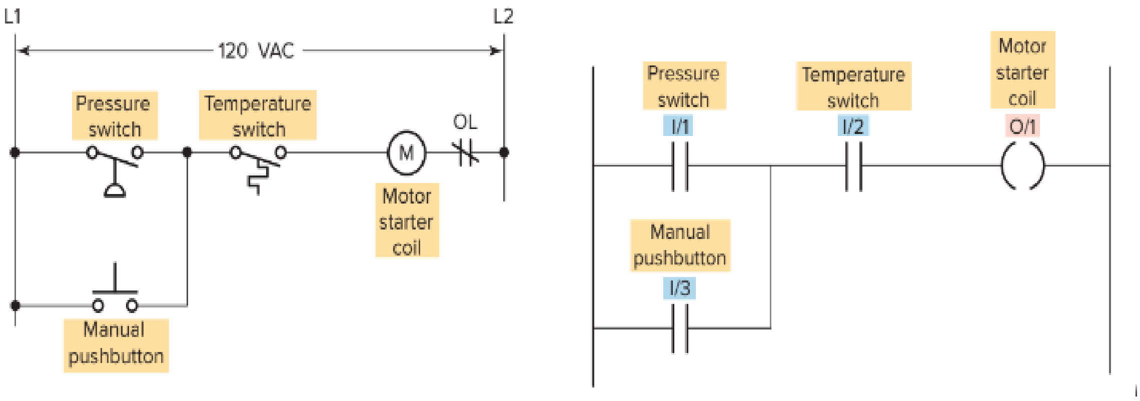


Figure 1-5 Relay ladder diagram for the modified process.

One of the key advantages of using a **PLC** is the ease of modifying its program. For example, if the process control circuit for a mixing operation needs to be changed so that the **manual pushbutton** can operate the system at any pressure but only after the **specified temperature** is reached, this modification is straightforward with a PLC.

- **In a relay-based system**, this change would require physical **rewiring** of the circuit.
- **In a PLC system**, no rewiring is needed. The **inputs and outputs remain the same**, and the only adjustment necessary is to update the **PLC ladder logic program** to reflect the new control requirements.

PLCs are categorized by their size, determined by the number of **Input/Output (I/O) points**:

- **Nano PLCs**: Less than 15 I/O points.
- **Micro PLCs**: 15 to 128 I/O points.
- **Medium PLCs**: 128 to 512 I/O points.
- **Large PLCs**: Over 512 I/O points.

When selecting a PLC, it's crucial to match it with the application's current needs but also consider future expansion.

PLC applications are classified into three main types:

1. **Single-ended (Stand-alone)**: One PLC controlling a single process without communicating with other PLCs or computers. Typically requires a smaller PLC.
2. **Multitask**: One PLC controls multiple processes, requiring sufficient I/O capacity and possible communication with other systems.

3. **Control Management**: A large PLC controls and communicates with several other PLCs and potentially a computer. This requires a large processor capable of managing multiple PLCs and networks.

Memory and Communication:

- **Memory Size** is critical for storing data, instructions, and control programs, typically measured in K units (1 K = 1024 bits/bytes/words).
- The required memory depends on factors like I/O points, control program size, data collection, supervisory functions, and future expansion.

In large processes where a PLC must communicate with a central system or other PLCs, provisions for a **data communication network** are required.

Table 1-1 Typical PLC Instructions

Instruction	Operation
XIC (Examine ON)	Examine a bit for an ON condition
XIO (Examine OFF)	Examine a bit for an OFF condition
OTE (Output Energize)	Turn ON a bit (nonretentive)
OTL (Output Latch)	Latch a bit (retentive)
OTU (Output Unlatch)	Unlatch a bit (retentive)
TOF (Timer Off-Delay)	Turn an output ON or OFF after its rung has been OFF for a preset time interval
TON (Timer On-Delay)	Turn an output ON or OFF after its rung has been ON for a preset time interval
CTD (Count Down)	Use a software counter to count down from a specified value
CTU (Count Up)	Use a software counter to count up to a specified value

PLC Chapter 1 Review Questions

1. **What is a PLC?**
   A Programmable Logic Controller (PLC) is a specialized computer used for automating industrial processes.
2. **Tasks PLCs perform besides relay switching**:
   - Timing
   - Counting
   - Data handling
   - Communication
3. **Six advantages of PLCs over relay-based systems**:
   - Increased flexibility
   - Easier to modify and troubleshoot
   - Lower cost for complex systems

- Higher reliability
- Communication capabilities
- Faster response time

4. **Open vs. Proprietary PLC architecture**:
    - **Open**: Compatible with devices from other manufacturers.
    - **Proprietary**: Requires components from the same manufacturer.
5. **Two ways I/O is incorporated into PLCs**:
    - Fixed I/O
    - Modular I/O
6. **Modular I/O connections to processor**:
   In a modular system, I/O modules plug into a rack, and the processor communicates with them via the backplane.
7. **Main functions of PLC components**:
    - **CPU**: Executes control programs.
    - **I/O Modules**: Interface with input and output devices.
    - **Programming Device**: Used to write and modify the control program.
    - **Power Supply**: Provides power to the PLC components.
8. **Two common PLC programming devices**:
    - Hand-held devices
    - Personal computers (PCs)
9. **Program and programming language**:
    - A **program** is a set of instructions the PLC follows.
    - A **programming language** defines the rules for writing these instructions.
10. **Standard programming language for PLCs**:
    Relay Ladder Logic (RLL)
11. **Relay ladder diagram (Figure 1-17)**:

- a. Pressure switch contacts close when pressure reaches a preset value.
- b. Temperature switch contacts close when the temperature reaches a preset value.
- c. The switches are connected in parallel.
- d. The motor starter coil is energized when either both switches close or the manual pushbutton is pressed.
- e. Voltage drops across open contacts:
    - (1) Pressure switch: Full voltage drop.
    - (2) Temperature switch: Full voltage drop.
    - (3) Manual pushbutton: Full voltage drop.

12. **PLC operates in real-time**:
    It processes input and output signals as they occur, responding immediately to field events.

13. **PLC ladder logic diagram (Figure 1-20)**:
- a. Symbols represent inputs, outputs, and logic elements.
- b. Numbers represent addresses of I/O points.
- c. I/2 represents the pressure switch.
- d. O/1 represents the motor starter.
- e. Two conditions for logic continuity: when both pressure and temperature switches are closed, or the pushbutton is pressed.
- f. The PLC scans inputs, evaluates the logic, updates outputs, and repeats the process.

14. **Change method in relay-based vs. PLC-based systems**:
    Relay systems require rewiring for changes, while PLC systems require only a program modification.
15. **Comparison between PLC and PC**:
- a. Hardware: PLCs are rugged and designed for industrial environments.
- b. Operating environment: PLCs withstand harsh conditions, while PCs are designed for general use.
- c. Programming: PLCs use ladder logic or other control-specific languages; PCs use general-purpose languages.
- d. Program execution: PLCs execute in real-time, while PCs process tasks sequentially.

16. **Two categories of software used with PLCs**:
- Programming software
- Monitoring software

17. **What is a PAC?**
    A Programmable Automation Controller (PAC) is a hybrid between a PLC and a PC, offering advanced processing and communication.
18. **Four criteria for categorizing PLCs**:
- Size (I/O points)
- Memory capacity
- Processing speed
- Communication capabilities

19. **PLC application types**:
- **Single-ended**: One PLC controls one process.
- **Multitask**: One PLC controls multiple processes.
- **Control management**: One PLC oversees multiple other PLCs.

20. **Memory capacity for a PLC with 16-bit words and 8 K words**:
    131,072 bits (16 x 8 x 1024)
21. **Five factors affecting PLC memory size**:
- Number of I/O points

- Size of the control program
- Data collection requirements
- Supervisory functions
- Future expansion

22. **PLC instruction set**:
    Refers to the set of commands or functions that the PLC is capable of executing, such as timers, counters, and math operations.