Programmable Logic Controller

By

Dr. Mohammad Salah
Mechatronics Engineering Department
Hashemite University
Outlines

- Introduction
- What can a PLC do?
- Structure and Hardware
- Programming the PLC
- PLC Operation
- PLC and Networks
Introduction

- The **automatic control** of repetitious mechanical or physical processes is a common control problem.
- Traditionally, this kind of **process control** was done with **electromechanical devices** such as relays, timers, and sequencers.
- With this approach, the circuit must be rewired if the control logic changes, which was a particular problem to the **automotive industry**.
- In response to this problem, in the late 1960s, **General Motors** developed the specifications for a **programmable electronic controller** that could replace the hard-wired relay circuits.
Introduction

- Based on those specifications, Gould Modicon Company developed the first programmable logic controller (PLC).
- The word **Programmable** differentiates it from the conventional hard-wired relay logic.
- The **PLC** also surpassed the hazard of changing the wiring.
- The **PLC** is a small, **microprocessor-based process-control computer** that can be connected directly to such devices as switches, small motors, relays, and solenoids, and it is built to withstand the industrial environment.
Introduction

**PLCs** are used in both **SCADA** and **DCS** systems as the **control components** of an overall hierarchical system to provide **local management of processes** through feedback control.
Using a **PLC** requires setting up the hardware and software.

- The hardware installation consists of wiring the **PLC** to all switches and sensors of the system and to such output devices as relay coils, indicator lamps, or small motors.
Introduction

- The control program is usually developed on a PC, using software provided by the **PLC** manufacturer.
- This software allows the user to develop the control program on the monitor screen.
- Once the program is complete, it is automatically converted into instructions for the **PLC** processor.
- The completed program is then downloaded into the **PLC**.
- Once the program is in the **PLC**’s memory, the programming terminal can be disconnected, and the **PLC** will continue to function on its own.
What can a PLC do?

- It can perform relay-switching tasks
What can a PLC do?

- It trouble-shoots more simply and more quickly
- It offers flexibility to modify the control logic, whenever required, in the shortest time
What can a PLC do?

- It can conduct counting, calculation and comparison of analog process values.
- It responds to the changes in process parameters within fractions of seconds.
What can a PLC do?

- It is cost effective for controlling complex systems
- It improves the overall control system reliability
- It can be worked with the help of the HMI (Human-Machine Interface) computer
What can a PLC do?

- Power Supply
- Processor (CPU)
- Memories
- Input/output modules
- Programming Port
- PLC Bus
- Expansion Models
Structure and Hardware
PLCs are usually powered directly from 120 or 240Vac

The power supply converts the AC into DC voltages for the internal microprocessor components

It may also provide the user with a source of reduced voltage to drive switches, small relays, indicator lamps, and the like
The processor is a microprocessor-based CPU and is the part of the PLC that is capable of reading and executing the program instructions, one-by-one (such as the rungs of a ladder logic program).

A special program called the operating system controls the actions of the CPU and consequently the execution of the user’s program.

The operating system is supplied by the PLC manufacturer and is permanently held in memory.

A PLC operating system is designed to scan image memory and the main memory which stores the ladder diagram program.
Memories

- The **program memory** receives and holds the downloaded program instructions from the programming device.

- This memory is usually an EEPROM (electrically erasable programmable ROM) or a battery-backup RAM, both of which are capable of retaining data.

- **Data memory** is RAM memory used as a “scratch pad” by the processor to temporarily store internal and external program-generated data.

- For example, it would store the present status of all switches connected to the input terminals and the value of internal counters and timers.
Memories

Structure and Hardware

Operating system
Input/output image bits
Data
User’s program space

Memory addresses
Bottom of memory

Memory map.

0000
0001
0002

RAM
ROM
Input/Output Modules

- The I/O modules are interfaces to the outside world.
- These control ports may be built into the PLC unit or, more typically, are packaged as separate plug-in modules, where each module contains a set of ports.
- The most common type of I/O is called discrete I/O and deals with on-off devices.
- Analog I/O modules allow the PLC to handle analog signals.
Discrete Input Modules (DIM)

- **DIM** connect real-world switches to the **PLC** and are available for either AC or DC voltages (typically, 240 Vac, 120 Vac, 24 Vdc, and 5 Vdc).
- Circuitry within the module converts the switched voltage into a logic voltage for the processor.
Discrete Output Modules (DOM)

DOM provide on-off signals to drive lamps, relays, small motors, motor starters, and other devices.

Several types of output ports are available: Triac outputs control AC devices, transistor switches control DC devices, and relays control AC or DC devices (and provide isolation as well).
Analog Input Modules (AIM)

- An **analog input module** has one or more **ADCs** (analog-to-digital converters), allowing analog sensors, such as temperature, to be connected directly to the **PLC**.
- Depending on the module, the analog voltage or current is converted into an 8-, 12-, or 16-bit digital word.

![Diagram of an analog input module showing temperature sensor, ADC, and connection to processor.](a) Analog input module
Analog Output Modules (AOM)

- An **analog output module** contains one or more **DACs** (digital-to-analog converters), allowing the **PLC** to provide an analog output—for example, to drive a DC motor at various voltage levels.

![Diagram of an analog output module](image)

(b) Analog output module
Input/Output Modules

Specialized modules that perform particular functions are available for many PLCs. Examples include:

- **Thermocouple module** — Interfaces a thermocouple to the PLC.
- **Motion-control module** — Runs independently to control multi-axis motion in a device such as a robot
- **Communication module** — Connects the PLC to a network
- **High-speed counter module** — Counts the number of input pulses for a fixed period of time
- **PID module** — An independently running PID self-contained controller (PID control can also be implemented with software, as described later in this chapter)
Programming Port and PLC Bus

- The **programming port** receives the downloaded program from the programming device (usually a PC).
- The **PLC** does not have a front panel or a monitor; thus, to “see” what the PLC is doing (for debugging or troubleshooting), you must connect it to a PC.
- The **PLC bus** are the wires which contain the data bus, address bus, and control signals. The processor uses the bus to communicate with the modules.
Expansion Modules

- Most **PLCs** are expandable
- Expansion modules contain additional inputs and outputs
- These are connected to the base unit using a ribbon connector
PLC Programming

- A PLC program is not actually a wiring diagram but a way to describe the logical relationship between inputs and outputs.
- The PLC programming languages are:
  - Continuous Function Chart (CFC)
  - Structured Control Language (SCL)
  - Sequential Control and State Graph (Graph)
  - Statement List (STL) or Instruction List (IL)
  - Function Block Diagram (FBD)
  - Ladder Logic or Diagram (LAD)
- The most common language is the LAD, FBD, and STL but the most used language is LAD.
A **LAD** (special kind of wiring diagram) was developed to document electromechanical control circuits.

Ladder diagram programs are highly symbolic and are the result of years of evolution of industrial control circuit diagrams.

This type of diagram has two vertical wires (rails) on either side of the drawing to supply the power.

Each rung of the ladder diagram connects from one rail to the other and is a separate circuit, which typically consists of some combination of switches, relay contacts, relay coils, and motors.

It is common for the coil of a relay to be in one rung and the contacts to be in another.
Ladder Diagram

24 V

Inputs

 Outputs

0 V

Program line 1

Program line 2

Program line 3

Program line 4

24 V bus line

0 V bus line

\| = normally open

\|\| = normally closed

Ladder format
Ladder Diagram

- **Power Rails**
  - Hot rail
  - Neutral rail

- **SW 1**, **SW 2**, **SW 3**, **SW 4**
  - SW 1 and SW 2 are inputs.
  - SW 3 and SW 4 are in a relay circuit.

- **Pilot light** (R) (`R = red`)
  - Connected to the power rails.

- **RELAY A (Coil)**
  - Connected to SW 3 and SW 4.

- **Motor**
  - Connected to RELAY A (Contacts).

- **Door Solenoid**
  - Connected to SW B and SW C.

- **Relay Coil**
  - Connected to SW A and SW B.

- **Relay Contacts**
  - Connected to Door Solenoid.

- **Light**
  - Connected to Relay Contacts.
PLC Programming

Ladder Diagram

(a) Hardware

(b) Ladder diagram
Ladder Diagram - Timers

- The Timer instruction provides a time delay, performing the function of a time-delay relay (e.g., controlling the time for a mixing operation or the duration of a warning beep)
- The length of time delay is determined by specifying a preset value
- The timer is enabled when the rung conditions become TRUE
- Once enabled, it automatically counts up until it reaches the Preset value and then goes TRUE (and stays TRUE)
- There are two types of time delay (On and Off)
Ladder Diagram - Timers

(a) Symbols

- On-delay relay (NO)
- Off-delay relay (NO)

(b) Pneumatic time-delay relay

(c) Thermal time-delay relay

(d) Solid-state time-delay relay
Ladder Diagram - Timers

(a) Ladder diagram

(b) Furnace
Ladder Diagram - Timers

Switch
I : 1/3

Timer
T : 1
Time base 0.01
Preset 500
Accumulator 0

Light
O : 2/4

Time base  The value of 0.01 means that each count corresponds to 0.01 seconds.

Preset  The value of 500 means that the delay will last 500 counts, which in this case is 5 s (0.01s × 500 = 5 s).

Accumulator  Holds the value of the current count.

EN  A bit that is TRUE as long as the timer rung is TRUE.
Ladder Diagram - Timers

(a) Hardware

(b) Ladder diagram

Start sw 1:1/0
Full sw 1:1/1
Fill valve O:2/0
Full sw 1:1/1
Timer T:1
3 min
Mixer O:2/1

T:1/T

Empty sw 1:1/2
Drain valve O:2/3

T:1/DN

Start sw

Full sw

Fill valve

Fill valve

Mixer

Drain valve
A Counter instruction keeps track of the number of times some event occurs (e.g., the count could represent the number of parts to be loaded into a box).

Counters may be either count-up or count-down types. The Counter will increment (or decrement) every time the rung makes a FALSE-to-TRUE transition.

The count is retained until a RESET instruction (with the same address as the Counter) is enabled.

The Counter has a Preset value associated with it. When the count gets up to the Preset value, the output goes TRUE. This allows the program to initiate some action based on a certain count.
Ladder Diagram - Counters

- **Switch**: I : 1/2
- **Count-up**
  - **Counter**: C : 1
  - **Preset**: 50
  - **Accumulator**: 0
- **Reset sw**: I : 1/3
- **C : 1/DN**
- **Light**: O : 2/1

**Preset**: When the counter gets to the Preset value, it makes the DN bit go TRUE and keeps on counting.

**Accumulator**: Holds the value of the current count.

**EN**: Goes TRUE when the counter rung is TRUE.

**DN**: Stands for done—goes TRUE when the count meets or exceeds the PRESET value.

**RESET**: A separate instruction with the same counter address; when this bit goes TRUE, the Accumulator resets to zero (resets the counter).
Ladder Diagram - Sequencers

- The Sequencer instruction is used when a repeating sequence of outputs is required.
- Traditionally, electromechanical sequencers (Figure 12.10) were used in this type of application (where a drum rotates slowly, and cams on the drum activate switches).
- The Sequencer instruction allows the PLC to implement this common control strategy.
Ladder Diagram - Sequencers

Sequencer File Adr
The first address of the Sequence file, which stores the sets of outputs.

Destination Adr
The output address (or slot) to which the Sequence file words are transferred with each step.

Length
The length of the Sequence file, that is, the number of steps in the sequence.

EN
Goes TRUE when the Sequencer rung is TRUE.

DN
Stands for done—goes TRUE when it has operated on the last word in the Sequence file.

Control Adr
Address that stores the control bits (EN, DN) and words (length) of the sequencer.
Ladder Diagram - Sequencers

(a) Hardware

(b) Sequencer File

(c) Ladder diagram
The temperature in an electric oven is to be maintained by a 16-bit PLC at approximately 100°C, using two-point control (actual range: 98-102°C). An oven with an electric heating element driven by a contactor (high-current relay), an LM35 temperature sensor (produces 10 mV/°C), an operator on-off switch, and the PLC. The PLC has a processor and three I/O modules: a discrete input module (slot 1), a 16-bit analog input module (slot 2), and a discrete output module (slot 3). Draw the ladder diagram for this system.
Ladder Diagram - Comparators

1. Compare $A < B$?
   $(1:2 < 98^\circ)$

2. Compare $A > B$?
   $(1:2 > 102^\circ)$

3. On-off sw
   $1:1/0$
   $O : 3/1$

   Temperature $> 102^\circ$
   $O : 3/0$
   $O : 3/2$

   Temperature $< 98^\circ$
   $O : 3/0$
   $O : 3/2$

(b) Ladder diagram

Analog input voltage for $98^\circ = 98^\circ \times \frac{10 \text{ mV}}{1^\circ C} = 980 \text{ mV}$

Analog input voltage for $102^\circ = 102^\circ \times \frac{10 \text{ mV}}{1^\circ C} = 1020 \text{ mV}$
PLC Operation

- The PLC accomplishes its control mission by executing a loop (scan) over and over again.

- For each scan:
  - Inputs are read in
  - Outputs are calculated based on the inputs
  - Outputs are sent out to the real world

- The order of the rungs in the ladder diagram may be important because the rungs are executed sequentially from top to bottom.

- For all practical purposes, one PLC can run three different control applications at the same time. Each application would have its own assigned I/O terminal and its own set of rungs in the program.
PLC Operation

Executing a ladder program.
PLC Operation

One complete cycle of all these steps is a **scan**. The steps are as follows:

1. **Reading** the data from the input module and **stores** the values of all eight inputs, as a word
2. The PLC **evaluates** the logic indicated by the first rung
3. After completely dealing with the **first rung**, the PLC turns its attention to the **second rung**
4. As a final action of the scan, the **updated output data** in RAM are sent to the output module, causing all eight output terminals to be updated at once
PLC Operation

A Simple PLC Application and Ladder Diagram

(a) Hardware

(b) Ladder diagram
Physically, a network is a wire acting as an “electronic highway” that can pass messages between nodes (PCs and other electronic devices).

Each node on the network has a unique address, and each message called a data packet (includes the address of where it’s going and where it came from).

All data on the network is sent serially (one bit at a time) on one wire.

The most common type of network uses the bus topology, which means that all the nodes tap into a single cable.
PLC and Networks
PLC and Networks

There are three good reasons for using a network:

- A device network *simplifies wiring*. Clearly the network is a simpler system that uses less wire. This reduces the amount of wiring needed.

- With a network, the sensor *data arrives in better shape*. In the traditional system, a low-level analog voltage may have to travel many feet. The signal is subject to attenuation and noise and other losses.

- Network devices tend to be more *intelligent*. For example, a photo cell could send a message saying the light level has diminished, (indicating that the lens may be getting dirty or that someone has bumped it out of position).
PLC and Networks

(a) PLC connected to sensors and actuators using point-to-point wiring

(b) PLC connected to sensors and actuators using a device-level network
PLC and Networks

Three levels of networks

1. Information Level
   - Programming station
   - Bridge
   - Mainframe

2. Control Level
   - Motors
   - Motion controller
   - PLC
   - Bridge
   - PLC

3. Device Level
   - Valve
   - Limit switch
   - Photo sensor
   - Temp sensor
   - Motor drive
   - Motor