Programmable Logic Controller (PLC)

Prof. Dr. M. Zahurul Haq
http://teacher.buet.ac.bd/zahurul/

Department of Mechanical Engineering
Bangladesh University of Engineering & Technology

ME 6401: Advanced Mechatronics

Introduction

Programmable Logic Controller (PLC)

PLC is defined as follows by EN 61131-1:
A digitally operating electronic system, designed for use in an industrial environment, which uses a programmable memory for the internal storage of user-oriented instructions to implement specific functions such as logic, sequencing, timing, counting and arithmetic, to control, through digital or analogue inputs and outputs, various types of machines or processes.

Both the PC and its associated peripherals are designed so that they can be easily integrated into an industrial control system and easily used in all their intended functions.

Features

- Field programmable by the user.
- Contains preprogrammed functions e.g. logic, timing, counting etc.
- Reduces hard wiring and wiring cost.
- Packaged appropriately to use in an industrial environment.
- Monitoring, error checking and diagnostics capability.
- Analog, digital and voltage inputs.
- Industry standard programmable languages.
- Competitive in both cost and space requirements.
- Industry standard I/O interfaces; capable of communicating with other PLCs, computers and intelligent devices.
- Scans memory and I/O in a deterministic manner.

Images

- Compact PLC (Mitsubishi FXO, Festo IPC FEC Standard)
- Modular PLC (Siemens S7-300)
PLCs are designed to communicate directly with inputs from the process and control outputs.

- PLCs use industry standard programming languages that are easily learnt and understood. Programming is primarily concerned with logic, timing, counting and switching operations.
- PLCs are packaged appropriately for use in an industrial environment. Variations in levels of noise, vibrations, temperature and humidity will not adversely affect the operation.

PLC is essentially a microcomputer, tailored specifically for certain control tasks.

- **Hardware**: consists of the actual device technology, i.e. the PCBs, integrated modules, wires, battery, housing etc.
- **Firmware**: is the software part, known as **executive software**, that is permanently installed and supplied by the the PLC manufacturer. Programs are usually stored in ROM or EPROM.
- **Software**: is the user program. User programs are usually stored in the RAM.
PLC hardware does not differ significantly from computers. What makes the PLC special is the software. The executive software is the program that the PLC manufacturer provides internal to the PLC, which executes the user’s program.

- The executive software determines
  - what functions are available to the user’s program,
  - how the program is solved,
  - how the I/O is serviced,
  - what the PLC does during power up and down and fault conditions.

Multitasking

Some PLCs are capable of executing multiple tasks with a single processor. User program assigns I/O for each task separately. Multitasking may take several forms:

- **Time driven** - The user may be allowed to configure the processor to run each task on periodic time intervals. This feature allows the time-critical portion of the control system, such as the portion that controls high-speed motions or machine fault detection, to run faster than the noncritical portions, such as servicing indicator lights.

- **Event driven or Interrupt driven** - user defines a particular event, such as an input changing state or an output turning off, that causes each tasks to run.

Basic Hardware

PLC consists of five major sections:

1. **Power Supply**
2. **Memory**
3. **Central Processing Unit**
4. **I/O Interface**
5. **Programming Section**

- **Power Supply** - PLCs are generally powered from AC mains and power supply system converts AC voltages to required DC voltages.
- **Memory** - Program memory receives and holds program instructions. Data memory is used to temporarily hold data generated from processes or acquired through I/O devices.
- **Processor** is a micro-processor based CPU and is the part of PLC that is capable of reading and executing program instructions and data.
- **Program loader** is used to enter/change the user program into the memory and to monitor the program execution.
PLC Packaging

The manner in which a PLC & its I/O are packaged is critical in determining its suitability for an application.

- **Heat Removal** - appropriate means must be provided for generated heat removal to ensure low internal temperature. Commonly used methods include air venting, forced air circulation & heat sinking.
- **Mounting** - to be mounted inside NEMA rated enclosure.

Programming Devices

Programming a PLC involves three components:

1. **Handheld Programmers** - are small inexpensive devices. These typically have membrane keys for entering data and LCD displays to show one line of a ladder program.
2. **Dedicated Terminals** - are designed for one particular brand of PLC. These provide troubleshooting operation while the PLC is running.
3. **Micro-Computers / PCs** - are widely used to program and simulate the program. Tested programs are downloaded to the PLC using serial communications.

PLC Programming Languages

EN 61131-3 defines five PLC programming languages:

1. **Ladder Diagram (LD)**: graphic programming language derived from the circuit diagram of directly wired relay controls.
2. **Function Block Diagram (FBD)**: functions and function blocks are represented graphically and interconnected into networks.
3. **Instruction List (IL)**: textual assembler-type language consisting of an operator and an operand.
4. **Structured Text (ST)**: high level language based on Pascal.
5. **Sequential Function Chart (SFC)**: a language resource for the structuring of sequence-oriented control programs.
PLC Programming Languages...

Ladder Diagram (LD):

```
Part_TypeA Part_present Drill-ok Sleeve_in
```

```
Part_TypeB
```

Function Block Diagram (FBD):

```
Part_TypeA

Part_TypeB

Part_present Drill-ok

Sleeve_in
```

Instruction List (IL):

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LD</td>
<td>Part-TypeA</td>
</tr>
<tr>
<td>OR</td>
<td>Part-TypeB</td>
</tr>
<tr>
<td>AND</td>
<td>Part-present</td>
</tr>
<tr>
<td>AND</td>
<td>Drill-ok</td>
</tr>
<tr>
<td>ST</td>
<td>Sleeve-in</td>
</tr>
</tbody>
</table>

Structured Text (ST):

Sleeve-in : = (Part-TypeA OR Part-TypeB) AND Part-present AND Drill-ok;

IL Codes used differ to some extent from manufacturer to manufacturer, and IEC 1131-3 is the proposed standard to unify IL codes.

1 Allen-Bradley
2 Cutler-Hammer

The PLC has one input module and one output module. Two external switches (SW-0 & SW-1) are connected via terminal IN-0 and IN-1 of input module. Two terminals of the output module (OUT-0 & OUT-1) drive two indicator lamps (Lamp-0 & Lamp-1).

The ladder diagram has two rungs. The top rung will light Lamp-0 if both SW-0 and SW-1 are closed. The bottom rung will light Lamp-1 if either SW-0 or OUT-0 are closed.
If the self diagnostic check determine that the system is operating properly, PLC start scanning operation.

- Update the Input Image Table
- Scan Program Instructions
- Update Output Terminals

Three-step scanning process is continuous and is repeated many times each second. The time it takes to complete one scan depends on the size of the program and the microprocessor clock speed.

### Phases of PLC Software Generation

1. **Specification**
   - Verbal description of control task
   - Technology, positional sketch
   - Macrostructure of control program

2. **Design**
   - Function chart to IEC 60848
   - Logic chart with symbols of the EN 60617-12 (IEC 60617)
   - Function table
   - Definition of software modules
   - Part list and circuit diagram

3. **Realisation**
   - Programming in LD, FBD, IL, ST and SFC
   - Simulation of subprograms and overall program

4. **Commissioning**
   - Design of system
   - Testing of subprograms
   - Testing of overall program

### PLC Software Documentation

- One important and crucial component of a system is documentation, which is an essential requirement for the maintenance and expansion of the system.

- **Documentation should include:**
  - Problem definition
  - Positional sketch or technology pattern
  - Circuit diagrams
  - Terminal diagrams
  - Printout of control programs
  - Allocation list of inputs and outputs
  - Additional documentation
Considerations in Choosing PLC

- Number and Types of input & output points required
- Size and type of memory required
- Speed and power required of CPU and instruction set
- Manufacturer’s support and backup

Guidelines and Standards

<table>
<thead>
<tr>
<th>Standard</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN 50170</td>
<td>General purpose field communication system</td>
</tr>
<tr>
<td>EN 60204</td>
<td>Safety of machinery - Electrical equipment of machines</td>
</tr>
<tr>
<td>EN 60617</td>
<td>Graphical symbols for diagrams;</td>
</tr>
<tr>
<td></td>
<td>Part 12: Binary logic elements</td>
</tr>
<tr>
<td>EN 61131</td>
<td>Programmable controllers;</td>
</tr>
<tr>
<td></td>
<td>Part 1: general information</td>
</tr>
<tr>
<td></td>
<td>Part 2: equipment requirements and tests</td>
</tr>
<tr>
<td></td>
<td>Part 3: programming languages</td>
</tr>
<tr>
<td>EN 61131</td>
<td>Preparation of function charts for control systems</td>
</tr>
<tr>
<td>ISO 1219</td>
<td>Fluid power systems and components;</td>
</tr>
<tr>
<td></td>
<td>Part 1: graphic symbols</td>
</tr>
<tr>
<td></td>
<td>Part 2: circuit diagram</td>
</tr>
</tbody>
</table>

Detailed Hardware

Power Supply

- Provides voltage levels required for internal operations (typically +5 V dc or ± 12 V dc).
- Provides power for I/O modules.
- Provides constant voltages.
- Packaged properly to prevent overheat.
- Separate or built into the processing unit.
- It is one of the most critical components of a PLC -
  - It is typically non-redundant. Hence a failure of the power supply can cause the control system to fail.
  - It will usually contain high-voltage components. Hence, an isolation failure can create the potential for serious injury and fire.
Memory

The memory function of the CPU stores programs and data that the CPU needs to perform various operations. The memory is organized into several sections according to the functions they perform.

- **Executive Memory** - collection of system programs stored in ROM.
- **Scratch Pad** - is the work area used to temporarily store the binary information used by the processor. These are volatile memory as RAM-type chips are usually used. Battery backed-up CMOS RAM are also used which may last up to 10 years.
- **Processor File** - the memory block in which programmer stores and manipulates the software. The processor file is made up of *program files*, and *data files*.

Central Processing Unit (CPU)

- CPU executes a program stored in the executive memory which is set by the manufacturer.
- It organizes all control activity by receiving inputs, performing logical decisions according to the program, and control the outputs.
- CPU does not operate on the I/O directly. Rather, it works with the I/O image stored in the I/O image memory. The I/O interface is responsible for transferring the image outputs to the I/O system, reading the inputs from the I/O system, and writing them into I/O image memory.
- A ‘watchdog’ timer is provided to time the CPU to execute the user’s program. If this time exceeds a predetermined value, watchdog timer will indicate fault and execute subsequent predefined procedure.

Input/Output (I/O) Systems

- The important functions of an **Input Module**: Reliable signal detection
  - Voltage adjustment of control voltage to logic voltage
  - Protection of sensitive electronics from external voltages
  - Screening of signals.
- **Output Modules** conduct the signals of the CPU to final control elements, which are actuated according to the task. Main functions include:
  - Voltage adjustment of logic voltages to control voltage
  - Protection of sensitive electronics from spurious voltages from the controller
  - Power amplification sufficient for the actuation of major final control elements
  - Short-circuit and overload protection of output modules.
I/O Systems...

- I/O system acts as the eyes, ears and hands of PLCs.
- **Discrete I/O** - most common type. This uses bit to represent a signal that is separate and distinct, such as ON/OFF, open/close or energized/de-energized. The processor reads this as the presence or absence of power.
- **Data I/O** - complex system needs data, requires ADC/DAC.
- According to EN 61131-3, the designations for inputs, outputs and memory are:

<table>
<thead>
<tr>
<th>Bit Type</th>
<th>Designation</th>
<th>Memory Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Bit</td>
<td>I, Q, M</td>
<td>I, Q, M</td>
</tr>
<tr>
<td>8 Bit</td>
<td>IB, QB, MB</td>
<td>IB, QB, MB</td>
</tr>
<tr>
<td>16 Bit</td>
<td>IW, QW, MW</td>
<td>IW, QW, MW</td>
</tr>
</tbody>
</table>

Examples of I/O's

- **Discrete I/O Inputs** - pushbuttons, selector switches, joy sticks, relay contacts, pressure switches, level switches, starter contacts, temperature switches, flow switches, limit switches, photo-electric switches, and proximity switches.
- **Discrete I/O Outputs** - light, relays, solenoids, starters, alarms, valves, heating elements, and motors.
- **Data I/O Inputs** - potentiometers, temperature transducers, level transducers, pressure transducers, humidity transducers, encoders, bar code readers, wind speed transducers.
- **Data I/O Outputs** - analog meters, digital meters, stepper motors (signals), variable voltage outputs, and variable current outputs.

Advanced I/O Modules

- **Thermocouple Module**
- **Motion-control Module**
- **Communication Module**
- **High-speed Counter Module**
- **PID Module**

Input Module Circuits

- **AC Input module circuit**
- **DC Input module circuit**
### Output Module Circuits

- **Load** (120 V ac)
- **Load** (10-60 V dc)
- **Load** (AC or DC)

### Analog I/O Modules

#### (a) Analog Input Module
- Temperature sensor
  -10 to +10 V
  -To processor

#### (b) Analog Output Module
- From processor
  -10 to +10 V

### PLC Programming

### I/O Capacity

A factor that determines the size of a programmable controller is the controller’s I/O and capacity.

1. **Mini-Micro** – usually 32 or less I/O, but may have up to 64.
2. **Small** – usually 64 to 128 I/O, but may have up to 256.
3. **Medium** – usually 256 to 512 I/O, but may have up to 1024.
4. **Large** – usually 1024 to 2048 I/O, but may have many thousands more on very large units.

The I/Os may be directly connected to the PLC or may be in a remote location. I/Os in a remote location from the processor section can be hardwired back to the controller, multiplexed over a pair of wires, or sent by a fiber optic cable.

### Ladder Diagram (LD)

- The use of ladder programming involves writing a program in a manner to drawing a switching circuit. The ladder diagram consists of two vertical lines representing the power rails, and circuits are connected as horizontal lines.
- **Advantages of Ladder Language** -
  - It is readily understood and maintained.
  - It provides graphic display of program flow.
  - Programming is fast.
  - Generates more readable programs for sequence control.
PLC Programming

PLC Ladder Programming Conventions

1. The vertical lines of the diagram represent the power rails between which the circuits are constructed.
2. Each rung on the ladder defines one operation in the control process.
3. A LD is read from left to right and from top to bottom.
4. Each rung must start with an input or inputs and must end with at least one output.
5. Electrical devices are shown in their normal conditions, e.g. a normally closed switch is shown closed.
6. A device can appear in more than one rung of a ladder.
7. The inputs and outputs are all identified by their addresses, the notation used depending on the PLC manufacturers.

A Simple PLC Program

In a PLC ladder diagram, each symbol is accompanied by its RAM address because it tells the PLC where to find the present logical state of the symbol in memory. The address of the individual bit may be specified by three quantities: the file type, word number and the position of the bit within the word.

Logical Function: AND

<table>
<thead>
<tr>
<th>Flow sw</th>
<th>Overload sw</th>
<th>Valve 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>I:1:0</td>
<td>I:1:1</td>
<td>O:23</td>
</tr>
<tr>
<td>Valve 1</td>
<td>O:23</td>
<td>Warning light</td>
</tr>
<tr>
<td></td>
<td></td>
<td>O:24</td>
</tr>
</tbody>
</table>

Elements of Ladder Diagram

Contacts
- | | Normally open contact
- / | Normally closed contact
- | F Edge contact, positive edge
- | N Edge contact, negative edge

Coils
- | | Coil
- / | Negating coil
- | S Setting coil
- | R Resetting coil
- | F Edge coil, positive edge
- | N Edge coil, negative edge

Solenoid
### Logical Function: OR

<table>
<thead>
<tr>
<th>IN 1</th>
<th>IN 2</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

LD IN 1 OR IN 2 = OUT END

---

### Logical Function: NAND

<table>
<thead>
<tr>
<th>IN 1</th>
<th>IN 2</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

LD NOT IN 1 OR NOT IN 2 = OUT END

---

### Logical Function: NOR

<table>
<thead>
<tr>
<th>IN 1</th>
<th>IN 2</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

LD NOT IN 1 AND NOT IN 2 = OUT END

---

### Logical Function: XOR (Mitsubishi Example)

<table>
<thead>
<tr>
<th>IN 1</th>
<th>IN 2</th>
<th>OUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

LD X400 AND NOT X401
LD NOT X400 AND X401
ORB OUT Y430

ORB: OR branches/block together.
Latch Circuit

- It is a self-maintaining circuit in that, after being energized, it maintains that state until another input is received.
- Should power fail while the machine is on, the latch rung will be de-energized. However, when the power is restored, the machine will not automatically restarted. It must be manually restarted by pressing START switch.

Timers/Counters

- Timers/counters instructions results in internal outputs that provides the same functions as hardware timers/counters.
- These are used to activate or deactivate a device after an expired intervals/counts.
- Both of these require an accumulator resistor to store the elapsed count/time and a register to store the preset value.
- Timers can be linked together, the term is *cascade*, to give larger delay times than is possible with just one timer.

Timers

- Timers are output instructions that are internal to the PLC. These are capable of providing timed control of devices that they activate or deactivate.
- EN 61131-3 defines 3 types of timer function blocks:
  1. TP: Pulse Timing
  2. TON: On-delay timing
  3. TOF: Off-delay timing
- The length of the time delay is determined by specifying a Preset value. 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.
A batch process— which involves filling a container with a liquid, mixing the liquid, and draining the container — is automated with a PLC. The sequence of events is as follows:

1. A fill valve opens and lets the liquid into the container until it is full.
2. Liquid in the container is mixed for 3 minutes.
3. A drain valve opens and drains the tank.

Counters

Counters are used to detect and count piece members and events. Counter instruction is placed in a rung and will increment (or decrement) every time the rung makes a FALSE-to-TRUE transition. The count is retained until a RESET instruction is enabled. The counter has a preset value associated with it. When the count gets up to the preset value, the output goes TRUE.

EN 61131-3 differentiates 3 different counter modules:

- CTU: Incremental counter
- CTD: Decremental counter
- CTUD: Incremental/Decremental counter
In a certain motor starting circuit, when the control switches 1 and 2 are closed, motor 1 and the timer are energized. After the time delay, motor 2 starts. An overload on either motor will de-energize the corresponding motor (switch 3 and 4).

**Example: 2-point Controller**

The temperature in an electric oven is to be maintained at approximately 100° C, using two-point controller.