

Presentation on PLC



Guided By:
Prof. Ram Kanhe

By: Shrikant
Sunpreet
Rohit

Introduction to PLC

- A **programmable logic controller (PLC)** is a digital computer used for automation of electromechanical processes, such as control of machinery on factory assembly lines, control of amusement rides, or control of lighting fixtures.



History and Origin:

- Developed to replace relays in the late 1960s.
- PLC began in the 1970s, and has become the most common choice for manufacturing controls.
- The PLC was invented in response to the needs of the American automotive manufacturing industry (primarily General Motors).
- Costs dropped and became popular by 1980s.
- Now used in many industrial designs.

Programmable Controller Development

1968 : Programmable concept developed.

1969 : Hardware CPU controller, with logic instructions, 1K of memory and 128 I/O points.

1974 : Use of several (multi) processors within a PLC - timers and counters; arithmetic operations; 12 K of memory and 1024 I/O points.

1976: Remote input/output systems introduced.

1977 : Microprocessors - based PLC introduced.

Programmable Controller Development

- 1980 : Intelligent I/O modules developed.
Enhanced communications facilities.
Enhanced software features.
(E.g: Documentation)
Use of personal microcomputers as programming aids.
- 1983 : Low - cost small PLC's introduced
- 1985 : Networking of all levels of PLC, computer and machine using SCADA software.

Selecting a PLC:

Criteria:

- Number of logical inputs and O/P.
- Memory.
- Number of special I/O modules
- Scan Time.
- Communications.
- Software.

Types of PLCs

Fixed

Modular



Leading Brands Of PLC

AMERICAN

1. Allen Bradley
2. Gould Modicon
3. Texas Instruments
4. General Electric
5. Westinghouse
6. Cutter Hammer
7. Square D

EUROPEAN

1. Siemens
2. Klockner & Mouller
3. Festo
4. Telemechanique

- JAPANESE
1. Toshiba
 2. Omron
 3. Fanuc
 4. Mitsubishi



PLC

PLC

PLC

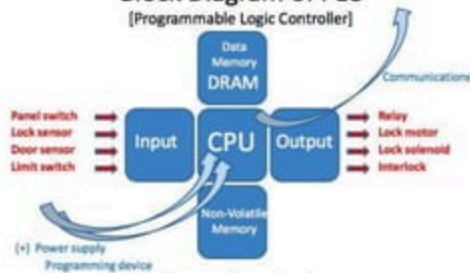
Basics Of PLC

Overview

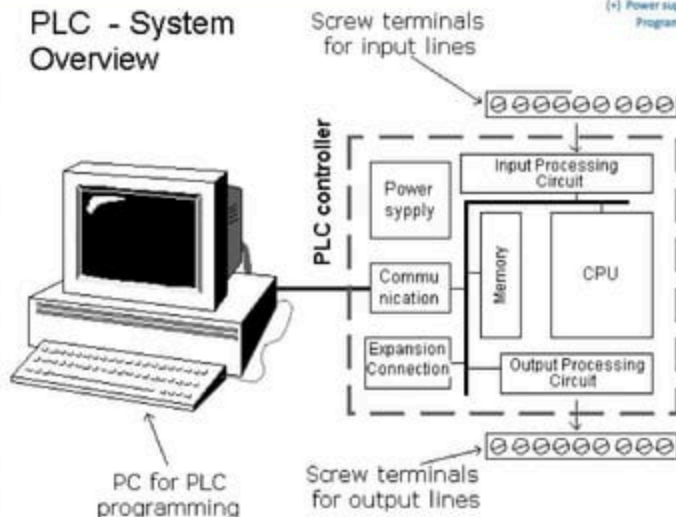


Block Diagram of PLC

[Programmable Logic Controller]



PLC - System Overview

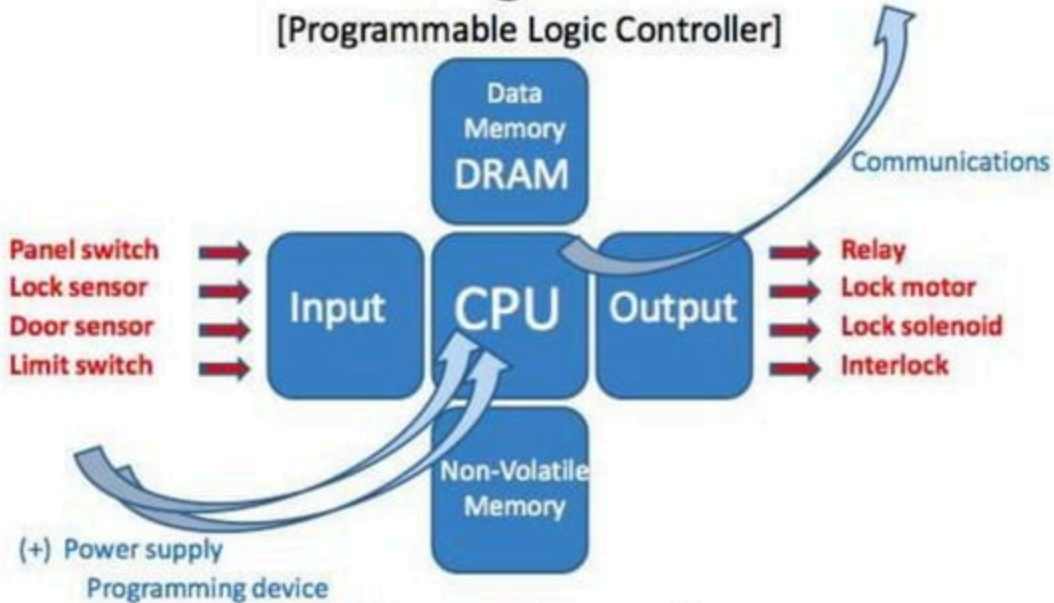


Door Control



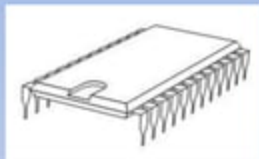
Block Diagram of PLC

[Programmable Logic Controller]

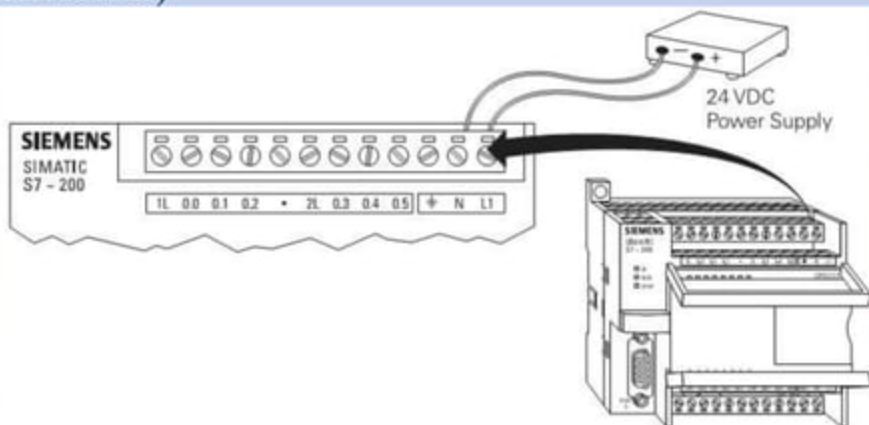


Door Control

- **CPU:** Its the unit containing the microprocessor.

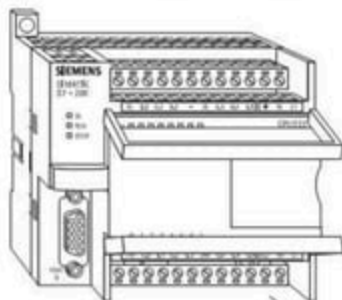


- **Power supply unit:** Its needed to convert the mains A.C. voltage to low D.C. Voltage(Normally Internal)



• Input-output sections:

are where the processor receives information from external devices and communicates information to external devices.



Pushbutton

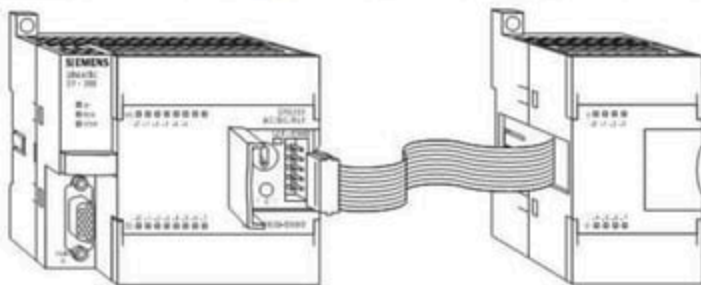


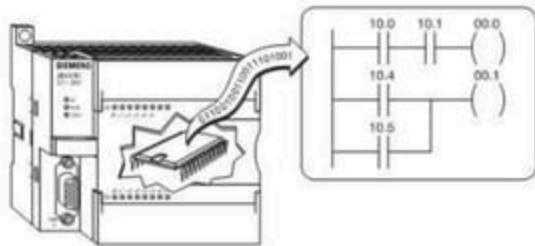
Switch

Input Devices
Connected Here

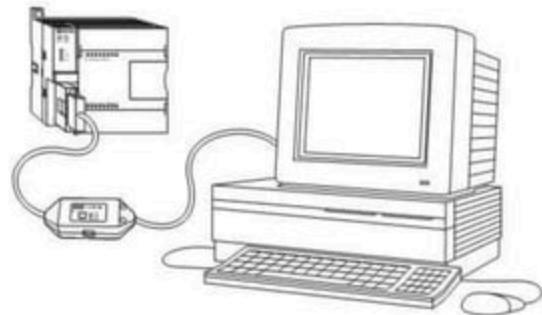
• Expansion Modules:

The S7-200 PLCs are expandable. Expansion modules contain additional inputs and outputs. These are connected to the base unit using a ribbon connector.





- **Memory unit:** is where the program is stored that is to be used for control actions.



- **Programming device:** is used to enter the required program into the memory of the processor.

PLC Operation:



The PLC program is executed as part of a repetitive process referred to as a scan. A PLC scan starts with the CPU reading the status of inputs. The application program is executed using the status of the inputs. Once the program is completed, the CPU performs internal diagnostics and communication tasks. The scan cycle ends by updating the outputs, then starts over. The cycle time depends on the size of the program, the number of I/Os, and the amount of communication required.

Fundamentals of Logic used in PLC

- PLC's employ different logic gates for control. Commonly used logic gates in PLC's:
- AND logic implementation to control devices:

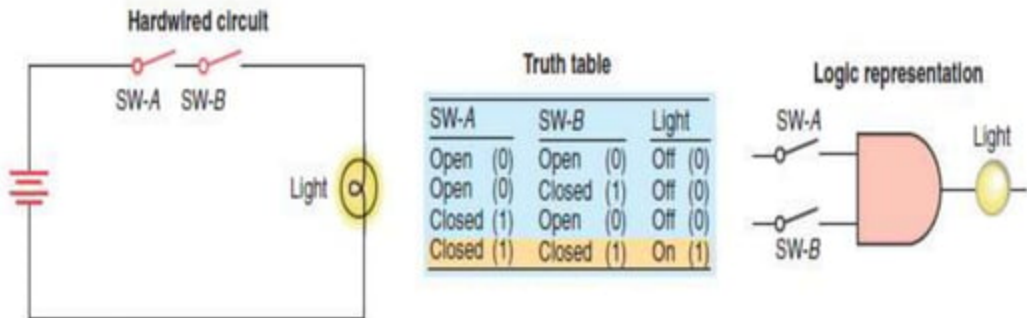


Figure 4-5 AND logic gate operates similarly to control devices connected in series.

➤ OR Gate Implementation:

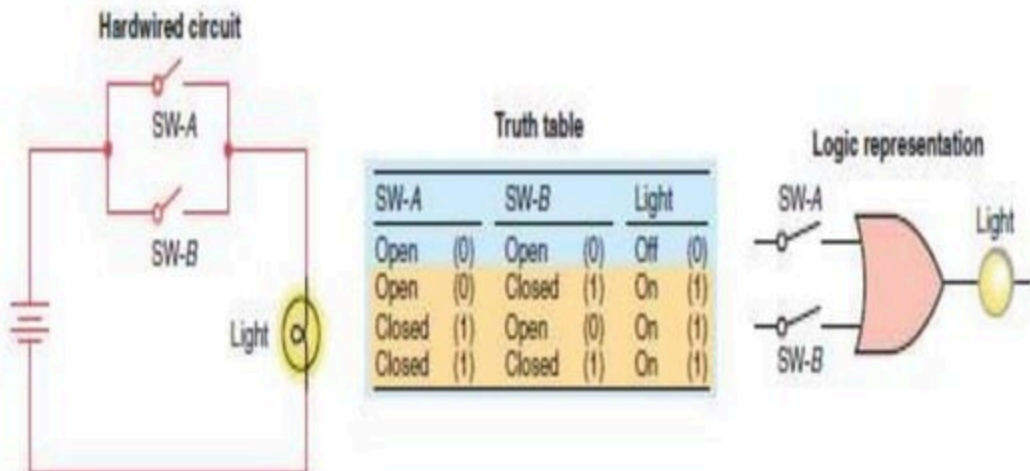


Figure 4-8 OR logic gate operates similarly to control devices connected in parallel.

➤ NOT Gate Implementation:

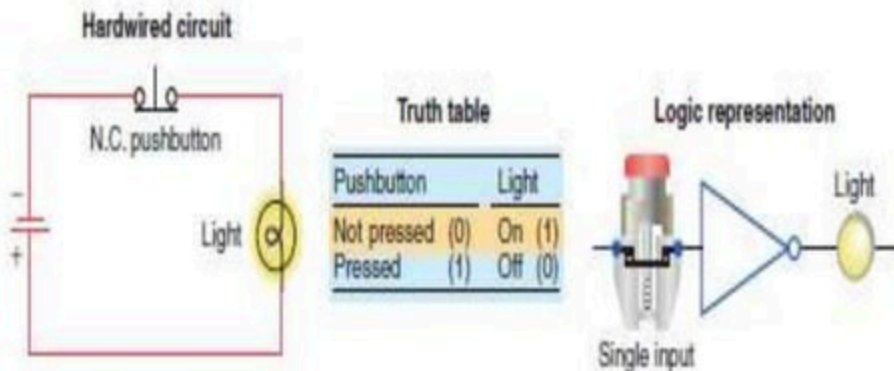


Figure 4-10 NOT function constructed using a normally closed pushbutton.

➤ X-OR Gate Implementation

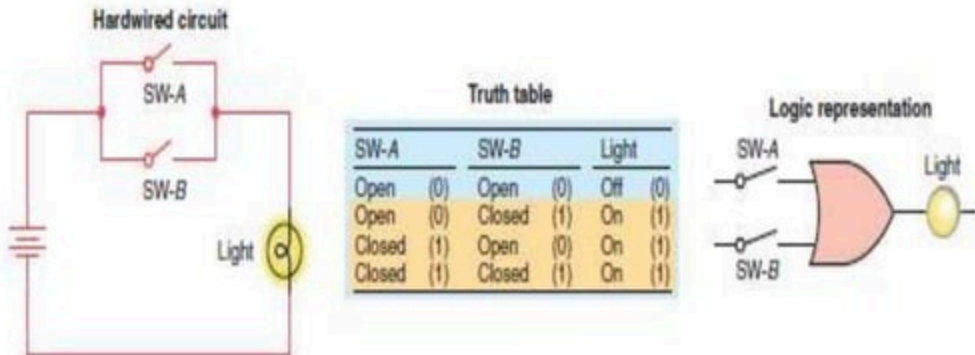


Figure 4-8 OR logic gate operates similarly to control devices connected in parallel.

Ladder Diagram and Programming:

Load: The load (LD) instruction is a normally open contact



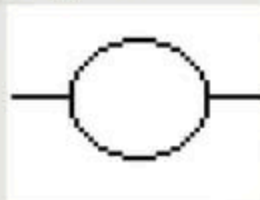
A Load (contact) symbol

Load Bar: The Load Bar instruction is a normally closed contact.



A Load Bar (normally closed contact) symbol

Out : The Out instruction is sometimes also called an Output Energize instruction. The output instruction is like a relay coil



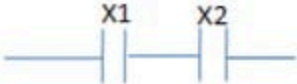
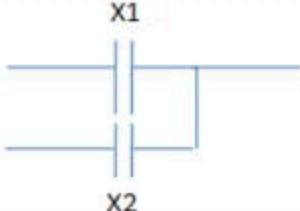
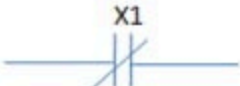
An OUT (coil) symbol

Out Bar: The out bar instruction is like a normally closed relay coil



An OUT Bar (normally closed coil) symbol

Logic elements

Ladder symbol	Hardware components
	X1 AND X2 ($X1.X2$)
	X1 OR X2 ($X1 + X2$)
	NOT X1

Boolean Instruction and Function**Graphic Symbol****Store (STR)–Load (LD)**

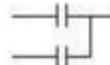
Begins a new rung or an additional branch in a rung with a normally open contact.

**Store Not (STR NOT)–Load Not (LD NOT)**

Begins a new rung or an additional branch in a rung with a normally closed contact.

**Or (OR)**

Logically ORs a normally open contact in parallel with another contact in a rung.

**Or Not (OR NOT)**

Logically ORs a normally closed contact in parallel with another contact in a rung.

**And (AND)**

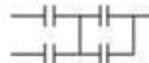
Logically ANDs a normally open contact in series with another contact in a rung.

**And Not (AND NOT)**

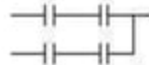
Logically ANDs a normally closed contact in series with another contact in a rung.

**And Store (AND STR)–And Load (AND LD)**

Logically ANDs two branches of a rung in series.

**Or Store (OR STR)–Or Load (OR LOAD)**

Logically ORs two branches of a rung in parallel.

**Out (OUT)**

Reflects the status of the rung (on/off) and outputs the discrete (ON/OFF) state to the specified image register point or memory location.

**Or Out (OR OUT)**

Reflects the status of the rung and outputs the discrete (ON/OFF) state to the image register. Multiple OR OUT instructions referencing the same discrete point can be used in the program.

**Output Not (OUT NOT)**

Reflects the status of the rung and turns the output OFF for an ON execution condition; turns the output ON for an OFF execution condition.



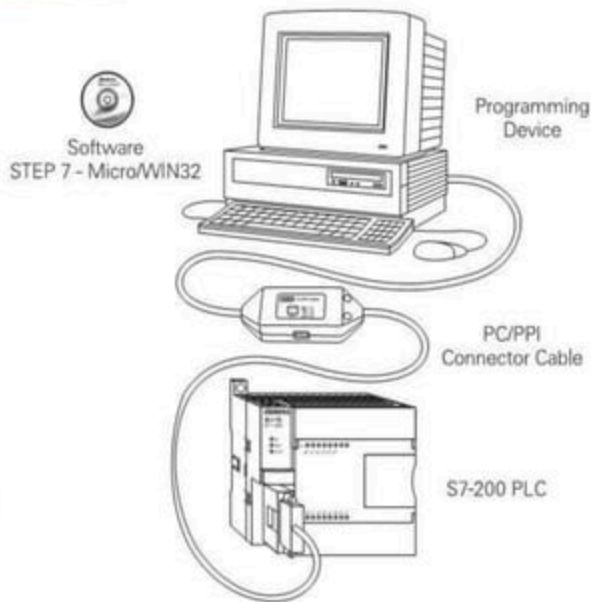
PLC

PLC

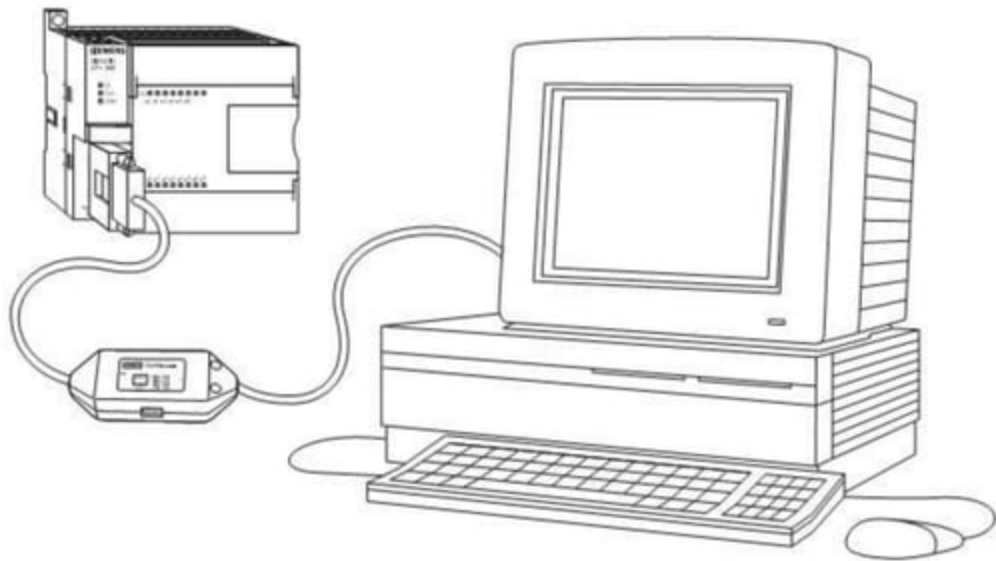
PLC

Programming a PLC :

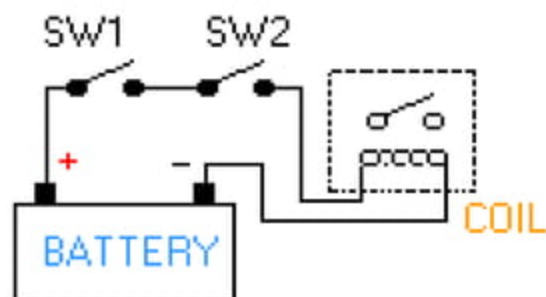
- In order to create or change a program, the following items are needed:
- PLC
- Programming Device
- Programming Software
- Connector Cable



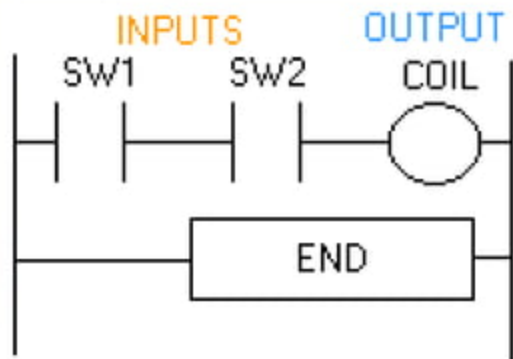
- You can use a personal computer as a programming device



Example of Ladder diagram:



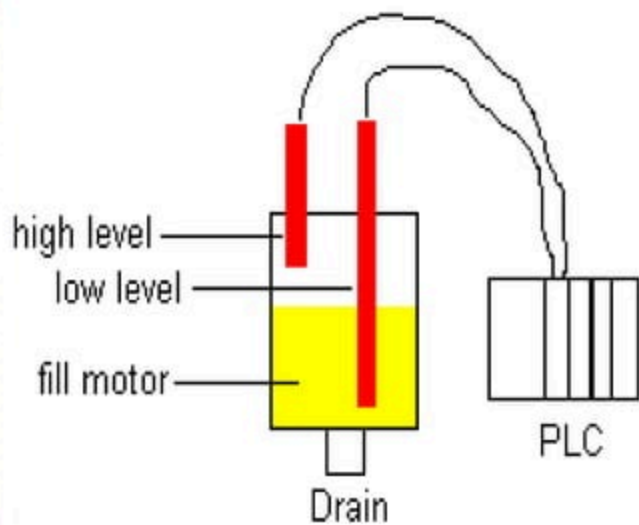
We can simulate this same circuit with a ladder diagram:



Example continued:

•We are controlling lubricating oil being dispensed from a tank. This is possible by using two sensors. We put one near the bottom and one near the top, as shown in the picture below

•Here, we want the fill motor to pump lubricating oil into the tank until the high level sensor turns on. At that point we want to turn off the motor until the level falls below the low level sensor. Then we should turn on the fill motor and repeat the process.



Dispensing oil from a tank

Examples continued:

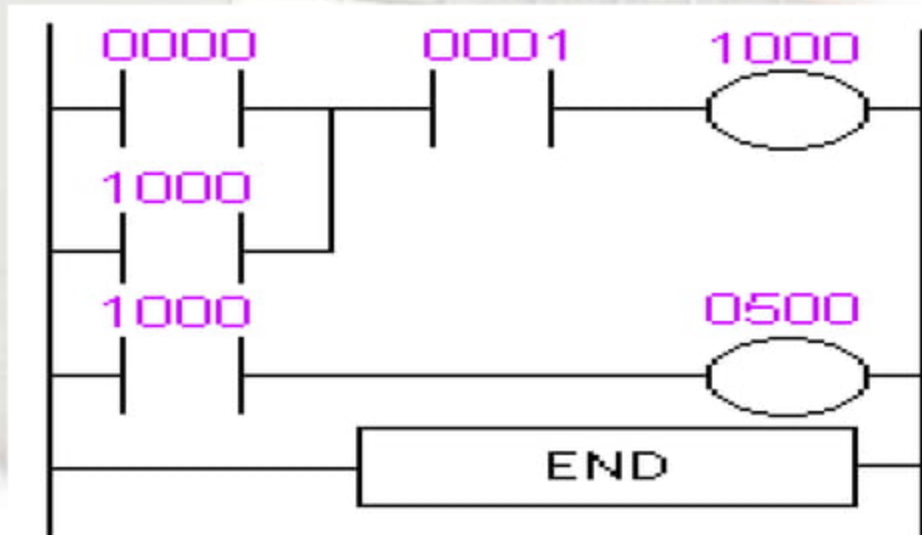
Inputs	Address
Low level sensor	0000
High level Sensor	0001

Output	Address
Motor	0500

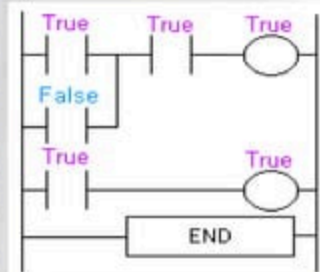
Internal Utility Relay
1000

Examples continued:

The Ladder Diagram

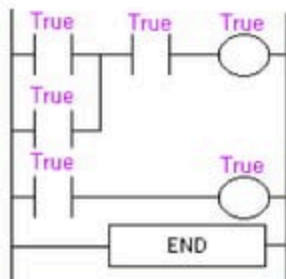


Examples continued:



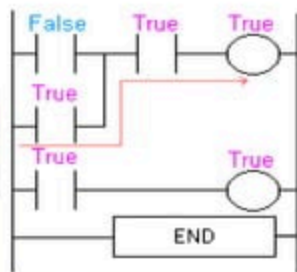
Scan 1

Initially the tank is empty. Therefore, input 0000 is TRUE and input 0001 is also TRUE



Scan 2

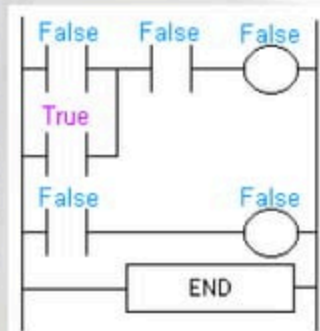
The internal relay is turned on as the water level rises.



Scan 3

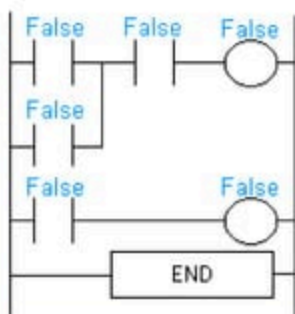
After scan 2 the oil level rises above the low level sensor and it becomes open. (i.e. FALSE)

Examples continued:



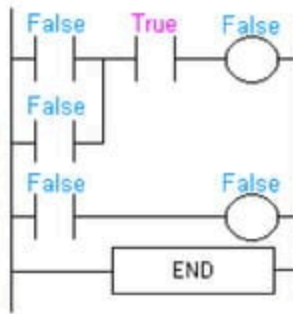
Scan 4

After scan 4 the oil level rises above the high level sensor at it also becomes open (i.e. false)



Scan 5

Since there is no more true logic path, output 500 is no longer energized (true) and therefore the motor turns off.



Scan 6

After scan 6 the oil level falls below the high level sensor and it will become true again.

Why PLC ?

- The controller is designed in modular form, so that sub-assemblies could be removed easily for replacement or repair.
- The control system needs the capability to pass data collection to a central system.
- The system becomes reusable.
- The method used to program the controller is simple, so that it can be easily understood by plant personnel.

PLC Compared to other systems:

A PLC can work in the harsh and extreme conditions of the industry where other systems such as a Micro-Controller cannot.

Programming is a lot easier as compared to other embedded systems which makes the job easy for the technicians.

PLC's allow end user to configure and control the application which is more useful in industries.

Advantages of PLCs:

- Less wiring.
- Wiring between devices and relay contacts are done in the PLC program.
- Easier and faster to make changes.
- Trouble shooting aids make programming easier and reduce downtime.
- Reliable components make these likely to operate for years before failure

Areas of Application

- Manufacturing/Machining
- Food/beverage
- Metals
- Power
- Mining
- Petrochemical/Chemical

Examples of PLC Programming Softwares:

1. Allen-Bradley – Rockwell Software
RSLogix 500
2. Modicon - Modsoft
3. Omron - Syswin
4. GE-Fanuc Series 6 – LogicMaster6
5. Square D- PowerLogic
6. Texas Instruments – Simatic
7. Telemecanique – Modicon TSX Micro