SECTION A

PLC(PROGRAMMABLE LOGIC CONTROLLER)

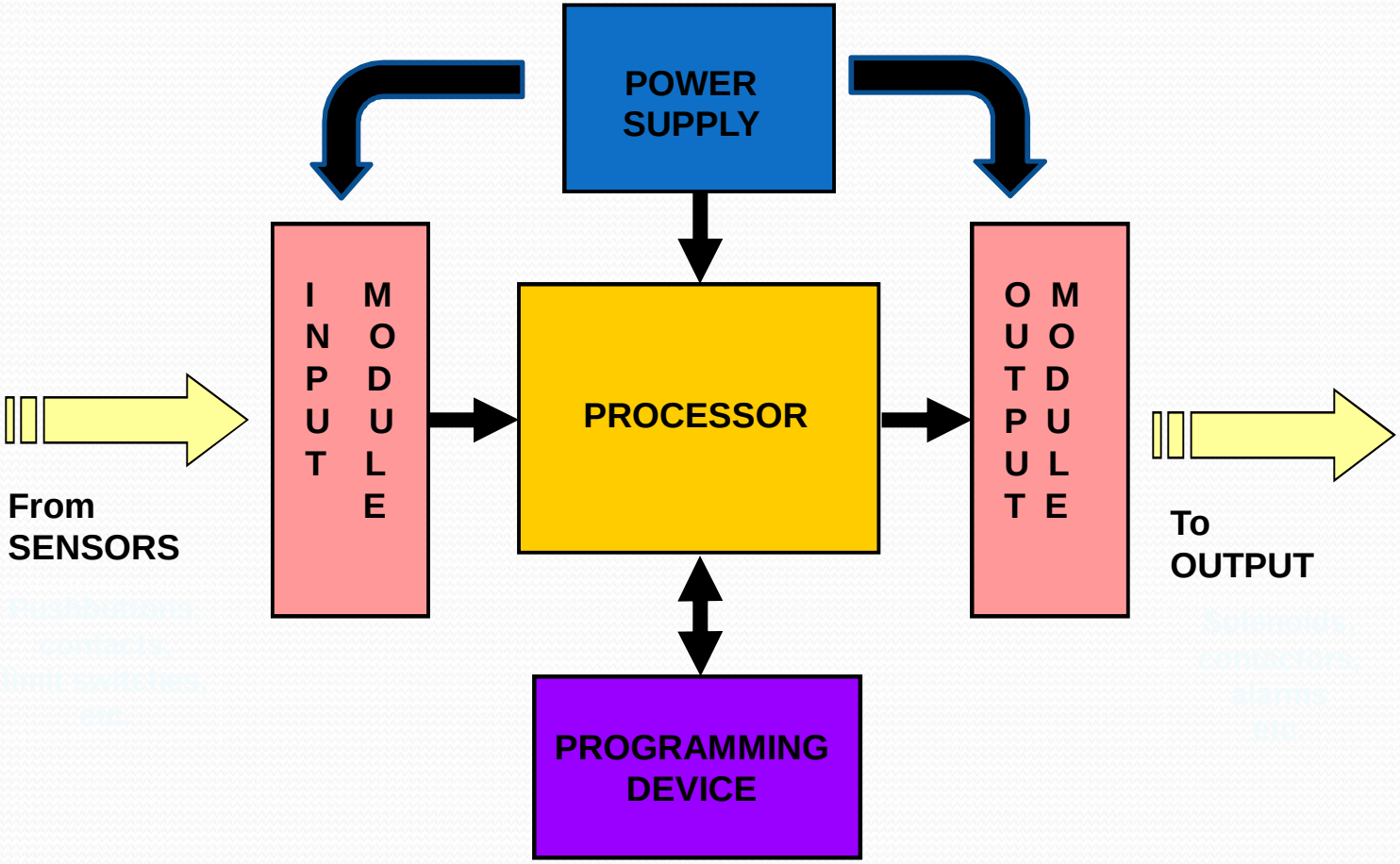
What is PLC?

PLC is a **digital computer** designed for **multiple inputs and output** arrangements, **extended temperature ranges**, **immunity to electrical noise**, and **resistance to vibration and impact**. A PLC is an example of a **real time system**.

History of PLC

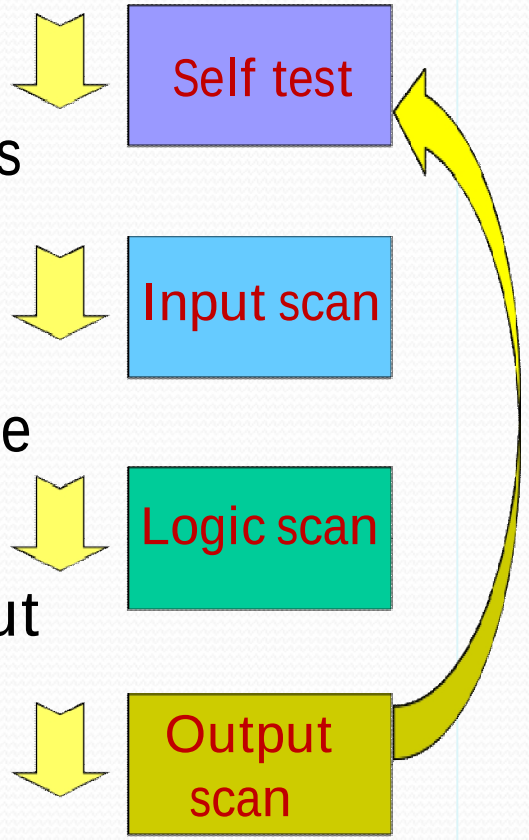
- PLC was introduced in late 1960's
- First commercial & successful Programmable Logic Controllers was designed and developed by **Modicon** as a relay replacer for General Motors.
- Earlier, it was a machine with thousands of electronic parts.
- Later ,in late 1970's,the microprocessor became reality & greatly enhanced the role of PLC permitting it to evolve form simply relay to the sophisticated system as it is today.

Major Components of a Common PLC



PLC operation sequence

1) **Self test**: Testing of its own hardware and software for faults.
2) **Input scan**: If there are no problems, PLC will copy all the inputs and copy their values into memory.
3) **Logic solve/scan**: Using inputs, the ladder logic program is solved once and outputs are updated.
4) **Output scan**: While solving logic the output values are updated only in memory when ladder scan is done, the outputs will be updated using temporary values in memory.



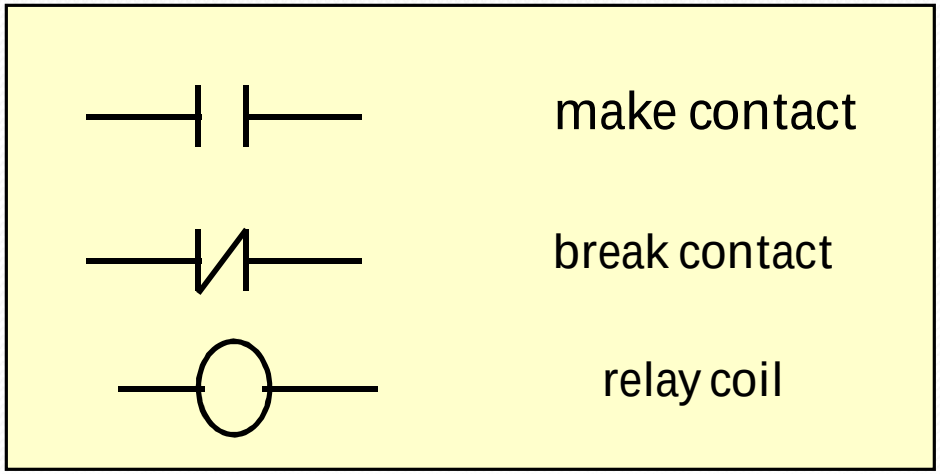
Programming Languages of PLC

Most common languages encountered in PLC programming are:

1) Ladder Logic
2) Functional Block Diagram
3) Sequential Function Chart
4) Boolean mnemonics

Ladder Logic

- The ladder logic is the oldest programming language for PLC.
- It is well suited to express Combinational logic.
- The main ladder logic symbols represent the elements :



OR GATE

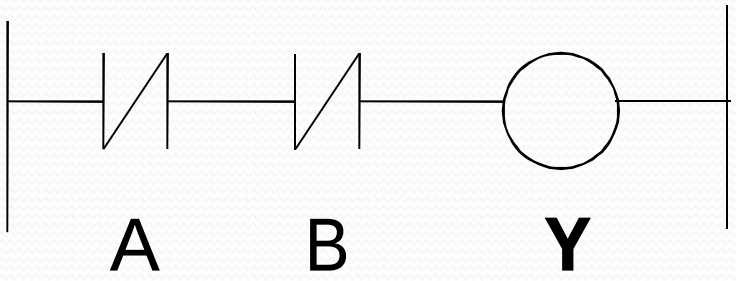
AND Gate

A	B	Logic(Y)
OFF	OFF	OFF
OFF	ON	OFF
ON	OFF	OFF
ON	ON	ON

A	B	Logic(Y)
OFF	OFF	OFF
OFF	ON	ON
ON	OFF	ON
ON	ON	ON

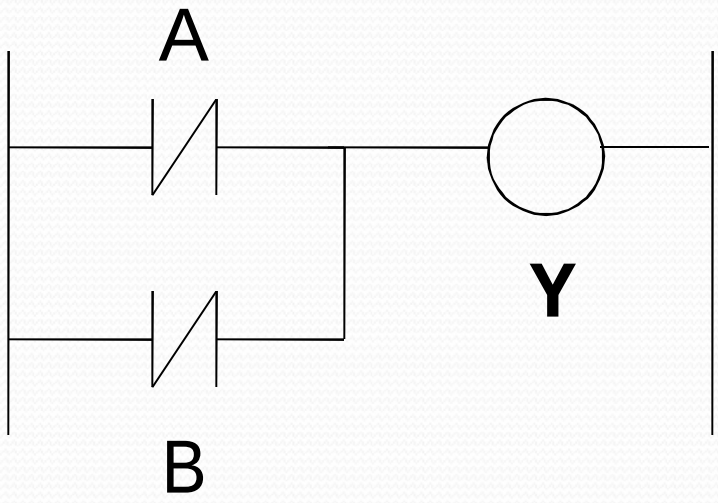
NOR Gate

A	B	Logic(Y)
OFF	OFF	ON
OFF	ON	OFF
ON	OFF	OFF
ON	ON	OFF

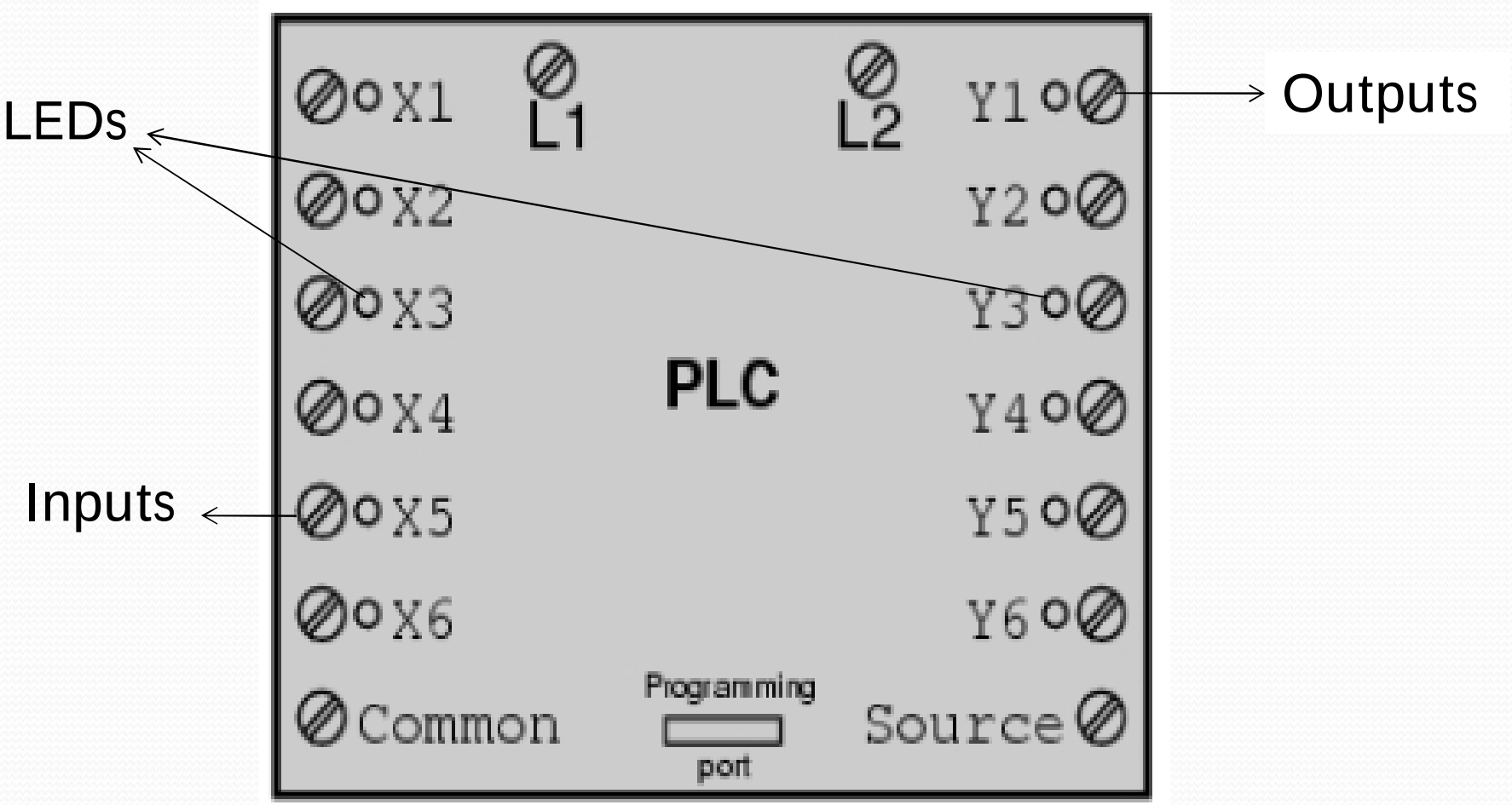


NAND Gate

A	B	Logic(Y)
OFF	OFF	ON
OFF	ON	ON
ON	OFF	ON
ON	ON	OFF

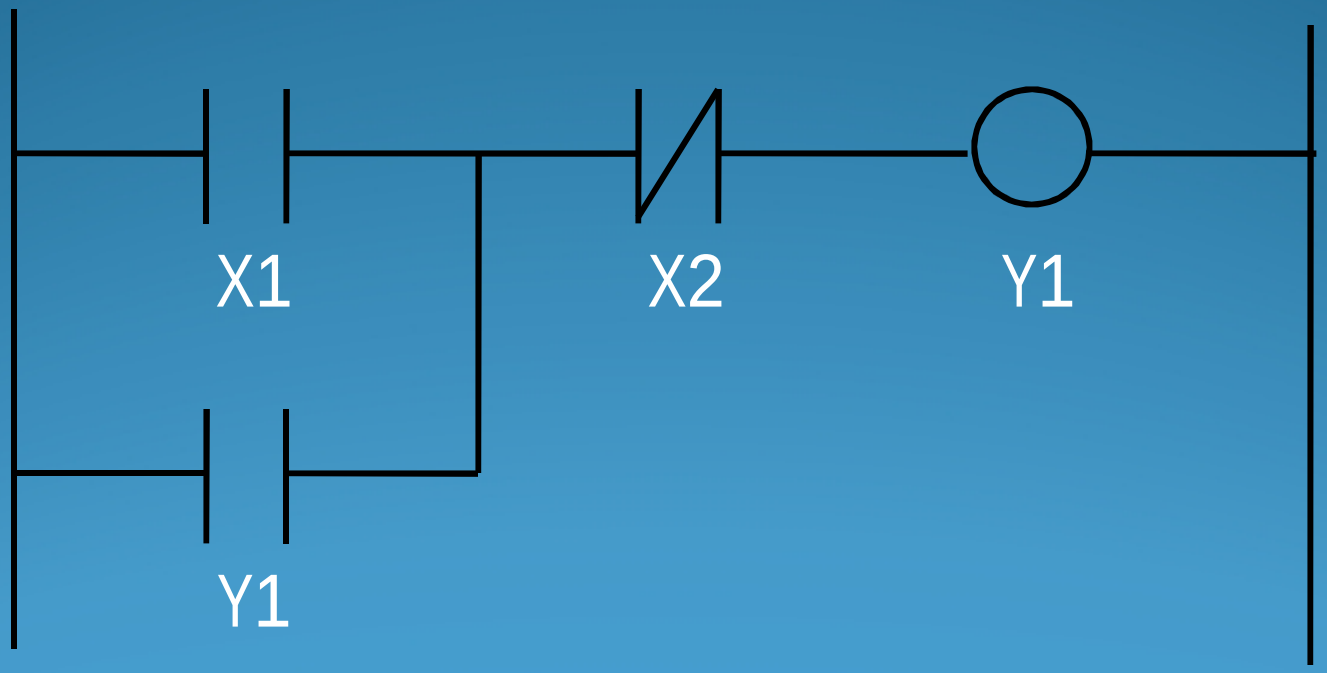


Block diagram of a PLC

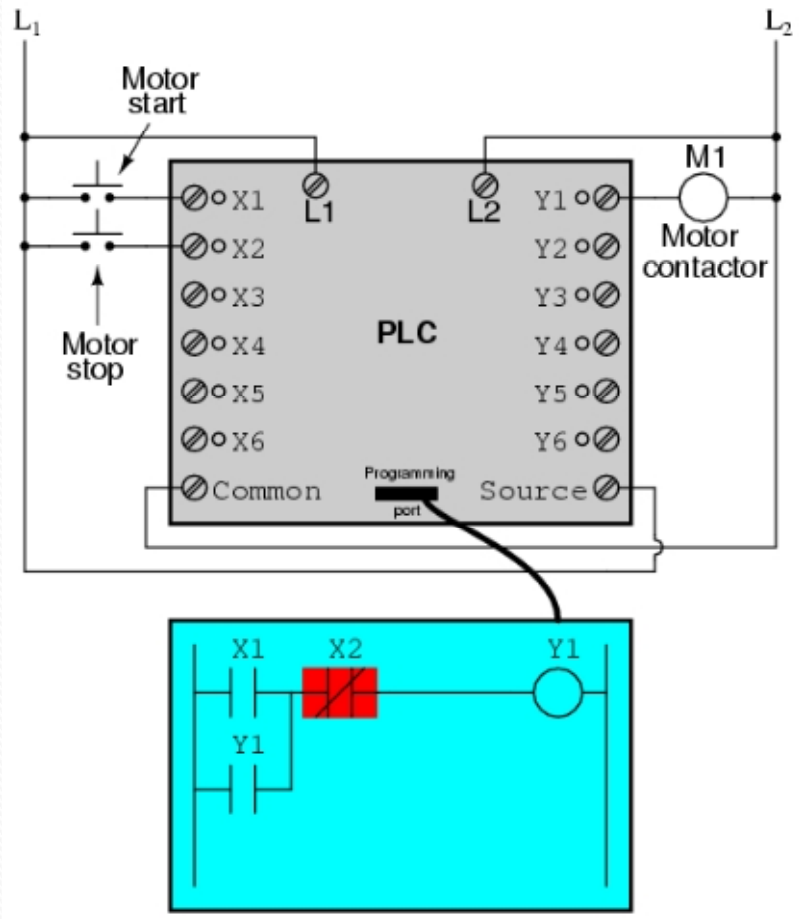


Programming Example:

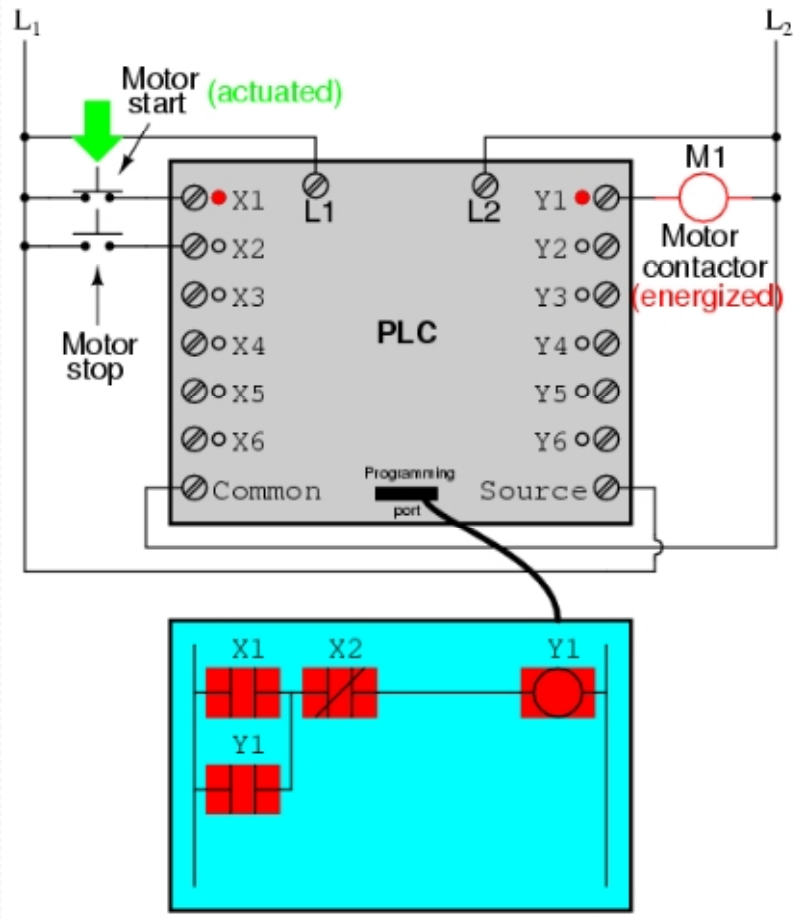
Ladder Logic Program for Start/Stop of Motor :



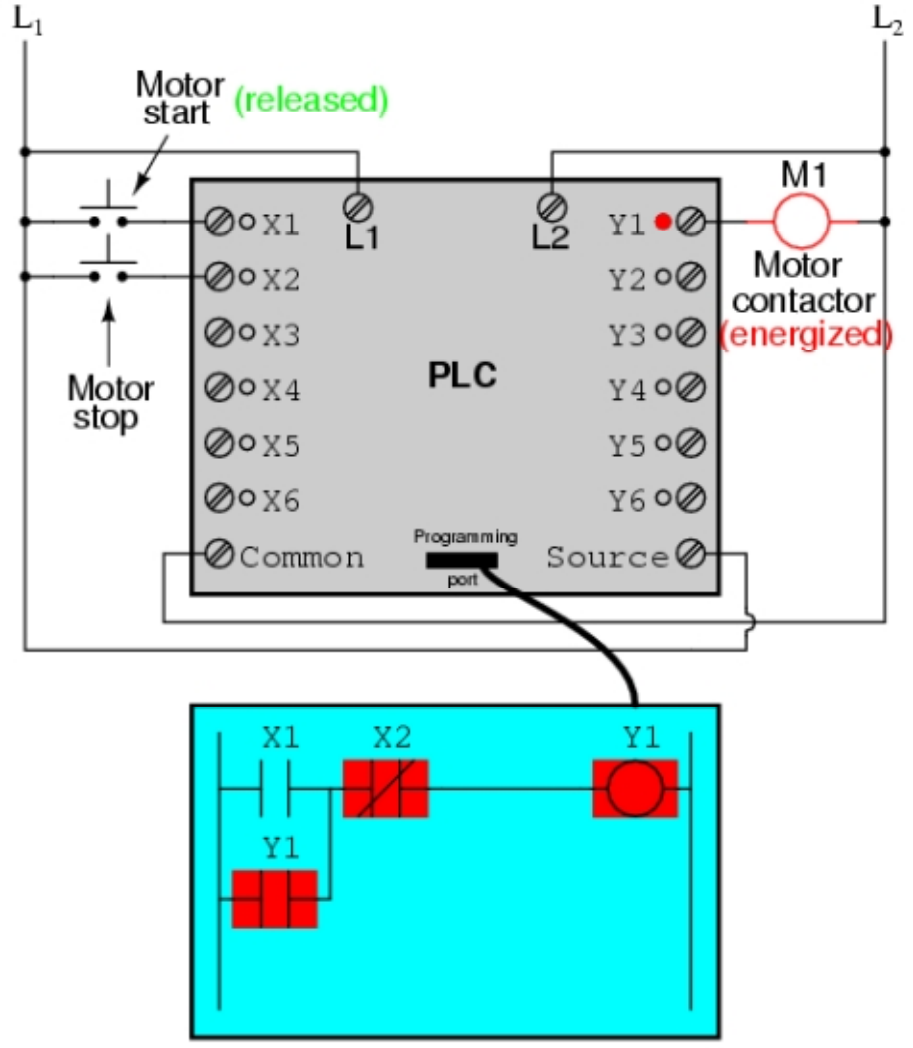
Programming PLC:



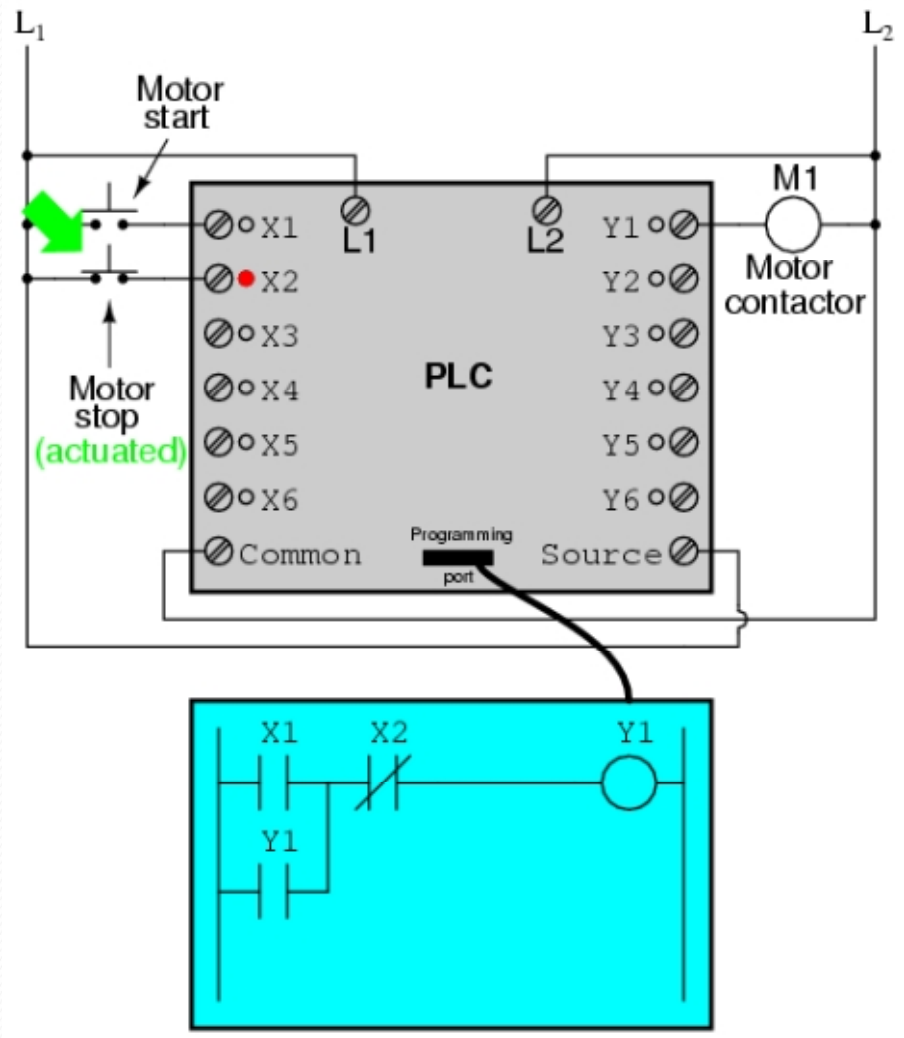
Starting of Motor:



Continuous Running of motor when Start Button is released:

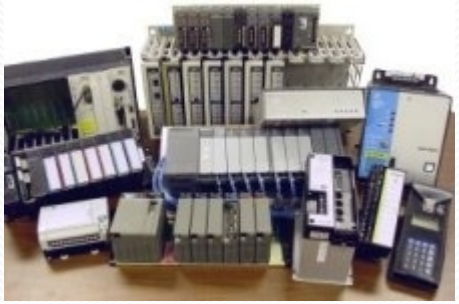


To Stop the Motor :



PLC's Are ...

- Similar to a Microcontroller:
  - Microprocessor Based
  - Onboard Memory for Storing Programs
  - Special Programming Language: Ladder Logic
  - Input/Output Ports



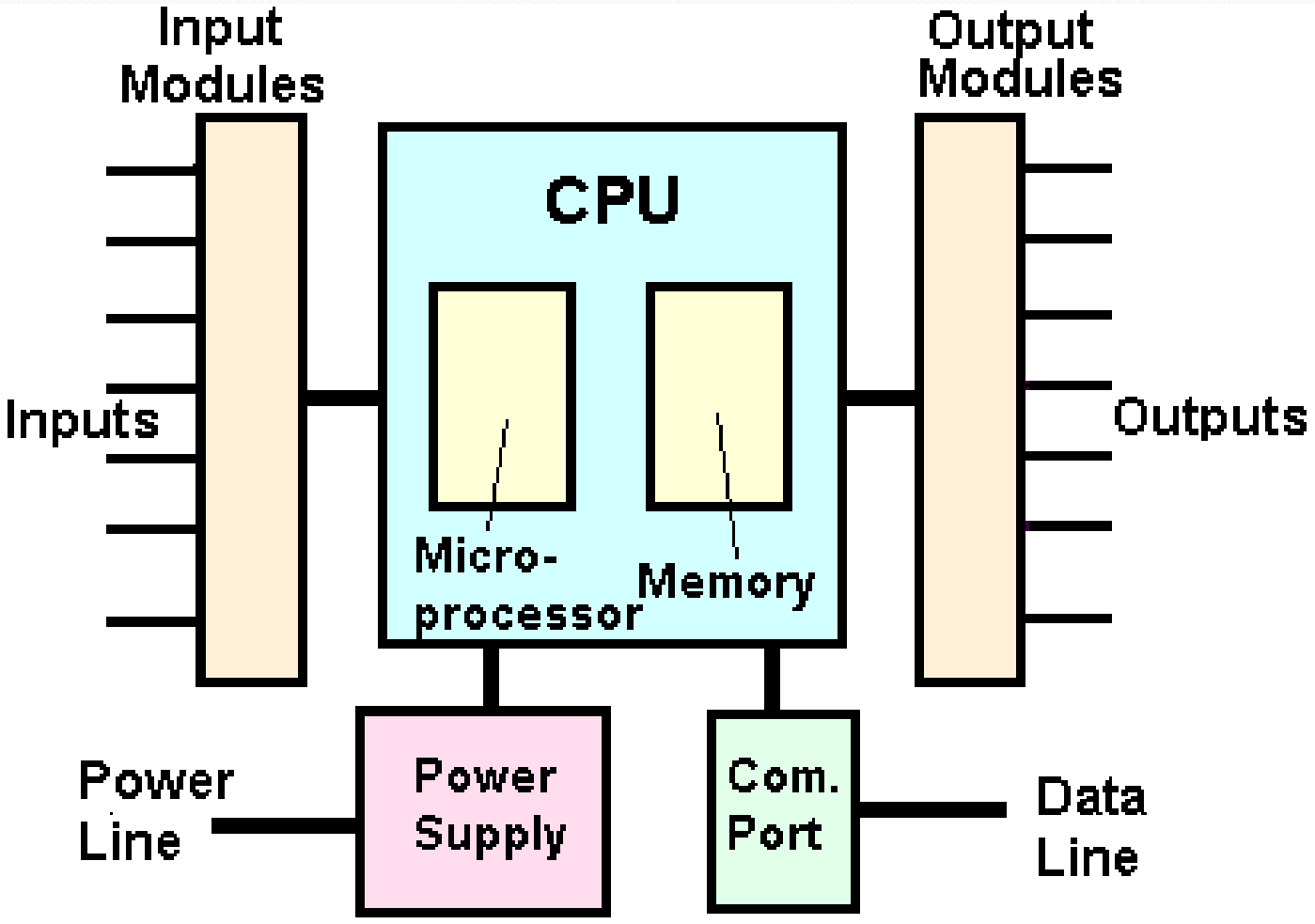
PLC's Are...

- Dissimilar to Microcontrollers:
  - Intended for Industrial Applications
  - I/O Designed to interface with Control Relays
  - Emphasis on Maximum Reliability

PLC's

- Widely Applied in Every Industry
- Were Developed to Simplify the Implementation of Control Automation Systems in Plants and Assembly Lines
- Designed to Minimize the Number of Control Relays in a Process and Maximize the Ways Relays can be Used
- First Applied to Automobile Industry in the Late 1960's
- Flexible, Reliable and Low Cost

PLC Components



PLC Components

I/O Modules

- Input Modules: Input Signals can be AC or DC, Analog or Digital
- Output Modules: Outputs are either AC or DC Analog Signals (Although it is possible to 'Construct' Digital Outputs)
- Modern PLC's have Expansion Ports to Increase the Number of Available Inputs and Outputs

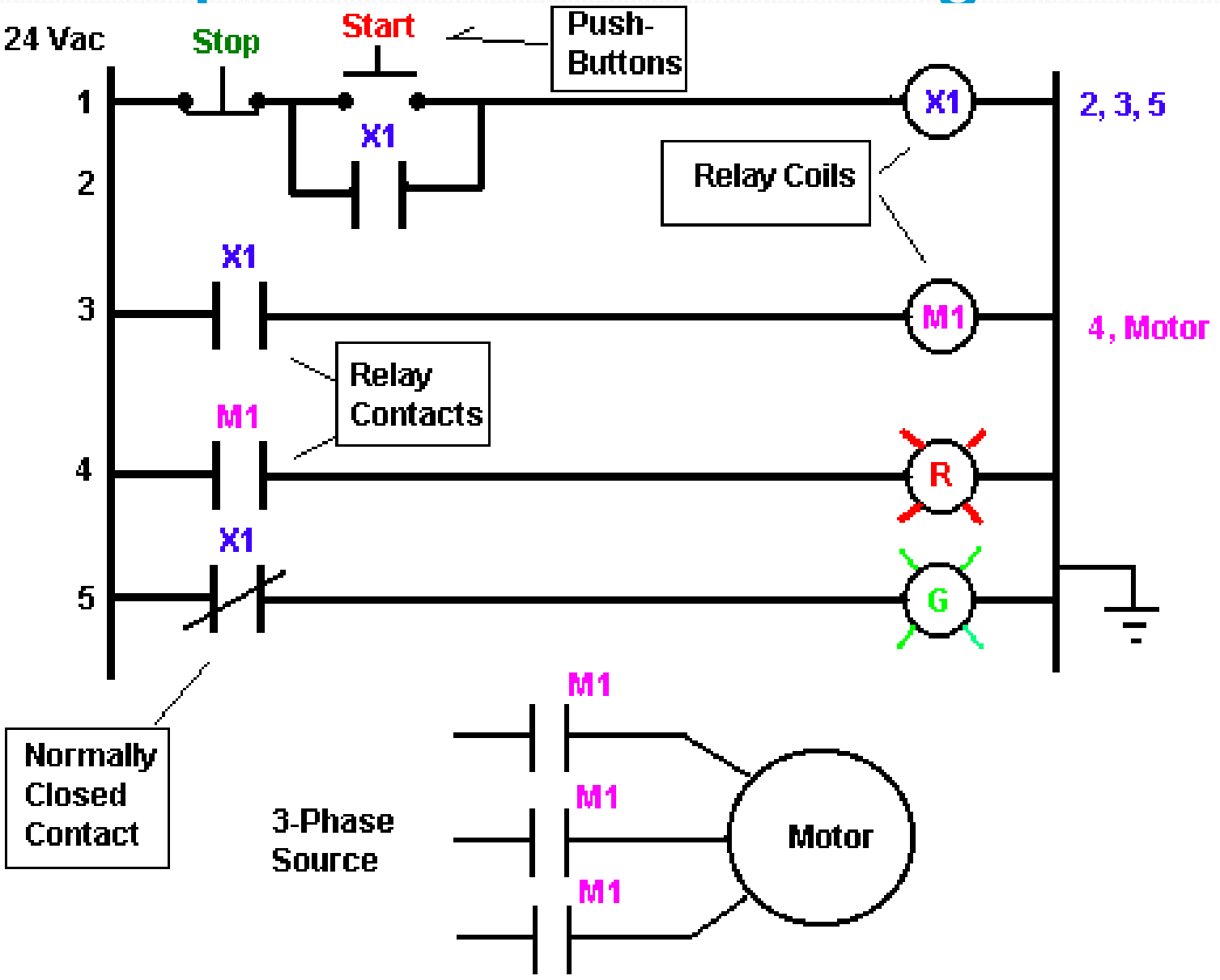
Examples of I/O Signals

- Inputs:
  - Pushbutton (Energizing or Grounding an Input)
  - Relay Contact Output
  - DC Voltage Level
  - Digital Logic Signal (+5V or 0 V, etc)
- Outputs:
  - 24 V ac
  - 120 V ac
  - 120 Vdc
  - etcetera

PLC's Use Ladder Logic

- Ladder Logic Diagrams Provide a Method to Symbolically Show How Relay Control Schemes are Implemented
- Relay Contacts and Coils, Inputs and Outputs lie on "Rungs" Between the Positive and Ground Rails

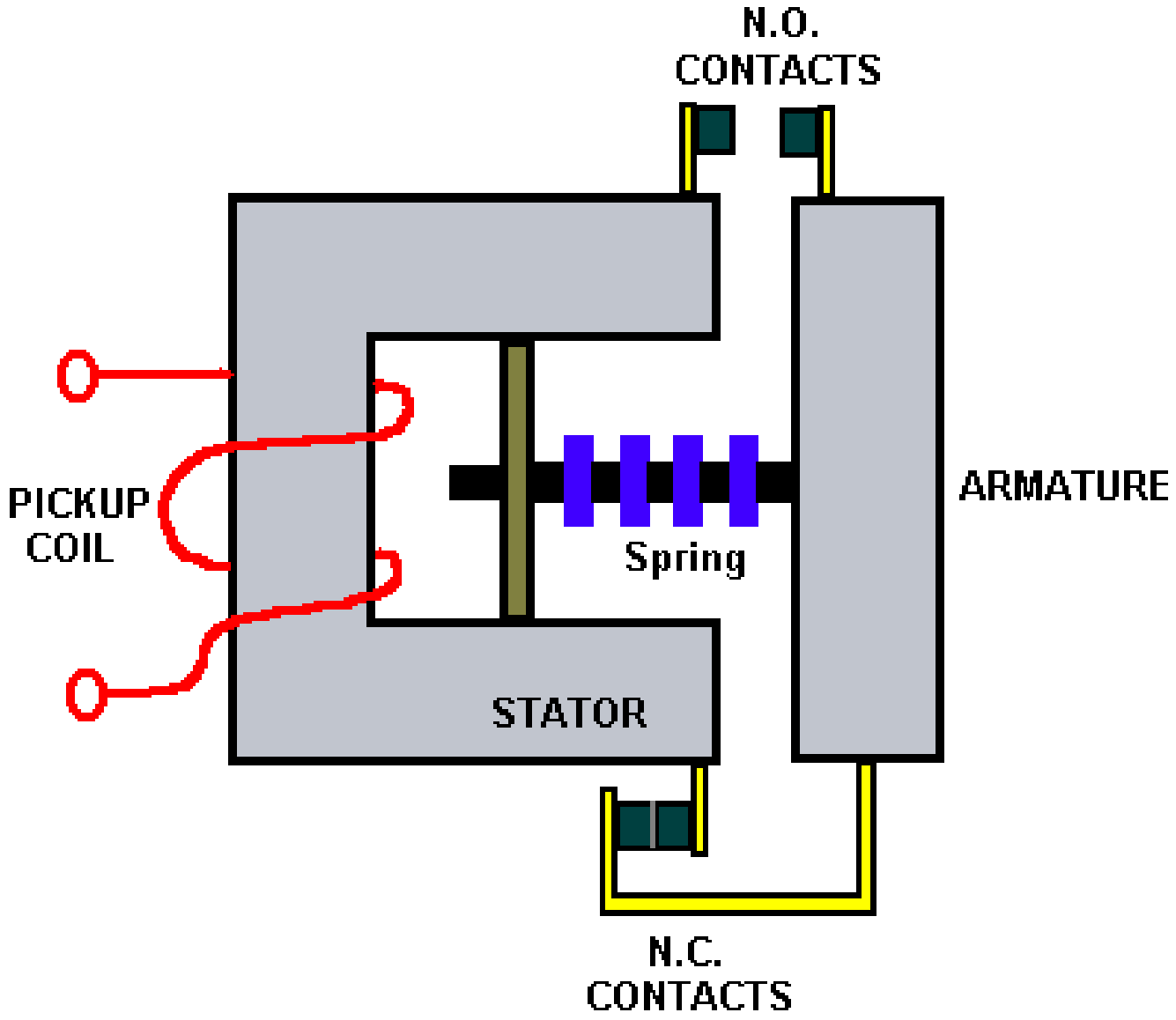
Example of Ladder Diagram



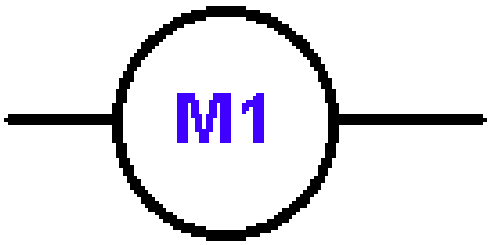
Relays

- In General, Relays Transform a Control Signal into a Control Action
- Relays Provide:
  - Isolation Between Input and Output
  - Leverage (Small Signal Can Control Large Action)
  - Automation (Minimize Human Interaction with a Control Process)

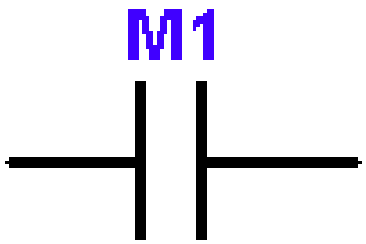
Relay Components



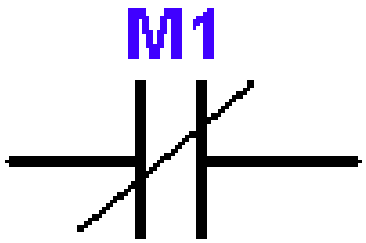
Basic Relay Symbols



Relay
Pickup
Coil



Normally Open
Contacts for
Relay Coil M1



Normally Closed
Contacts for
Relay Coil M1

Relay Applications

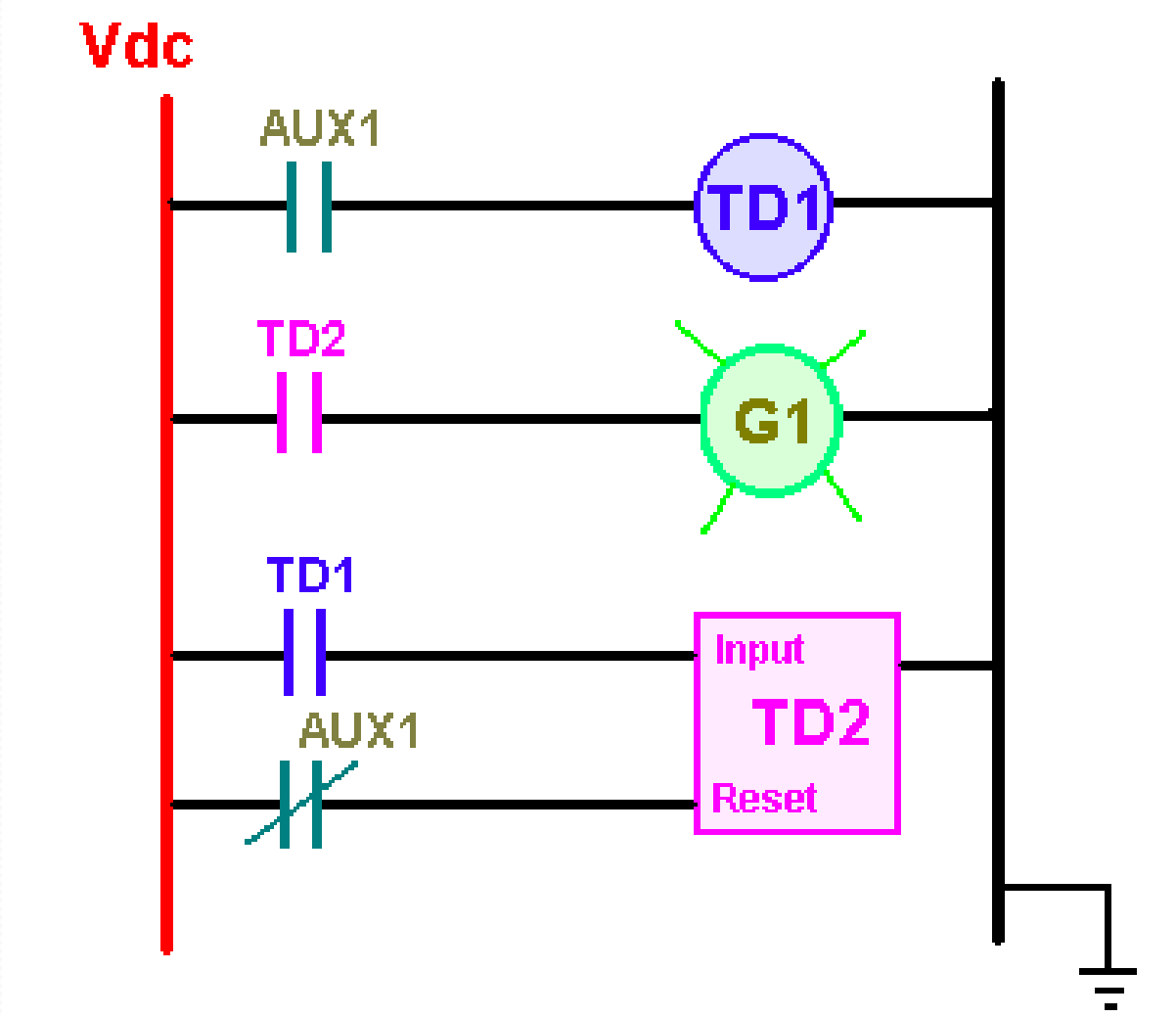
- Relays can be Designed to Perform Many Functions
  - Detect Out of Limit Conditions on Voltages and Currents
  - Start Motors
  - Prevent Motors from Over Heating
  - Control Assembly Lines
  - Adjust Lighting

Time Delay Relays

- When TD Relay Pick-Up Coil is Energized, a Delay is Initiated
- Normally Open Contacts Wait to Close until Delay is Completed
- Normally Closed Contacts Wait to Open until Delay is Completed
- Very Useful for Creating a Sequence of Control Events

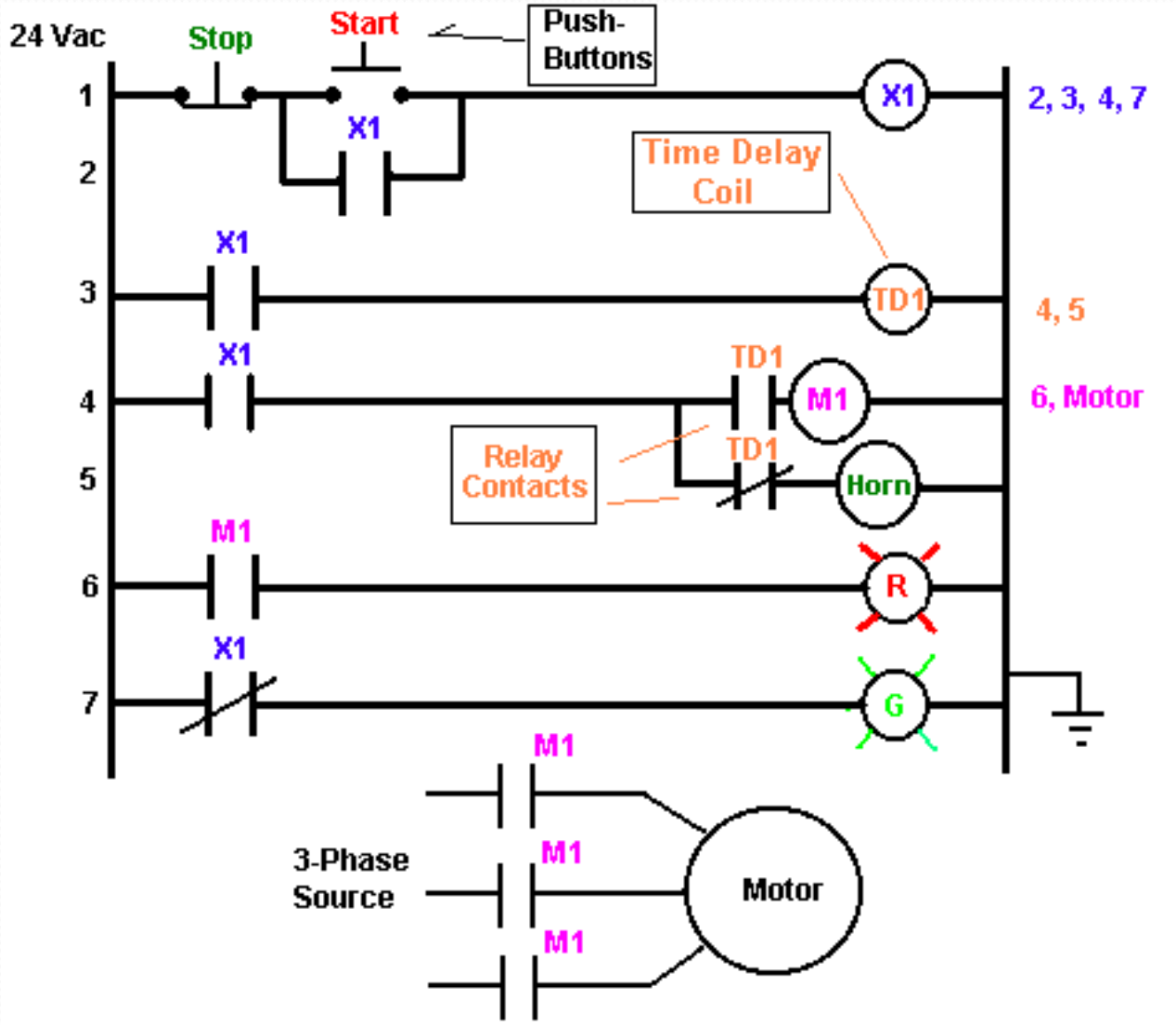
Time Delay Symbol

- Can be Constructed With or Without a Reset Input



Making Use of Delays

- Delay Motor Start While Alarm Sounds for Safety



Counters

- Counter Relays must “Count” a pre-determined number of events before changing contact status
- Can Count Up (UpCounter) or Count Down (DownCounter)
- e.g. An UpCounter is set to 8 and is programmed to detect every occurrence of a 5 Volt pulse. When it has detected 8 such occurrences, the NO Contacts close and the NC contacts open.
- Great for making Real-Time Clocks, etc

Special Functions

- Modern PLCs can perform many Math and Logic Functions without additional Ladder Logic Programming
  - Differentiation, Integration
  - **+, -, *, /**
  - Boolean Logic Functions (AND, NOT, OR)
  - Master Control Functions (Reset, etc)

Motor Protection

- Essential Part of Motor Control
- Protect against:
  - Under Voltage
  - Under Frequency (AC Machines Only)
  - Over Current
  - Over Heating
  - Over Speed
  - Over Load

LADDER LOGIC FUNCTIONS

- Ladder logic input contacts and output coils allow simple logical decisions. Functions extend basic ladder logic to allow other types of control. For example, the addition of timers and counters allowed event based control.
- Combinatorial Logic and Event functions have already been covered. This includes Data Handling and Numerical Logic.

Objectives:

- To understand basic functions that allow calculations and comparisons
- To understand array functions using memory files
- Functions for data handling, mathematics, conversions, array operations, statistics, comparison and Boolean operations.

SEQUENTIAL FUNCTION CHARTS

Objectives:

- Learn to recognize parallel control problems.
- Be able to develop SFCs for a process.
- Be able to convert SFCs to ladder logic.

Basic PLC Function Categories

- Combinatorial Logic
  - relay contacts and coils
- Events
  - timer instructions
  - counter instructions
- Data Handling
  - moves
  - mathematics

INSTRUCTION LIST PROGRAMMING

- Objectives:
  - To learn the fundamentals of IL programming.
  - To understand the relationship between ladder logic and IL programs

- Instruction list (IL) programming is defined as part of the IEC 61131 standard.
- It uses very simple instructions similar to the original mnemonic programming languages developed for PLCs.
- It is the most fundamental level of programming language - all other programming languages can be converted to IL programs.
- Most programmers do not use IL programming on a daily basis, unless they are using hand held programmers.

Advantages of PLCs:

- Reliability.
- Flexibility in programming and reprogramming.
- Cost effective for controlling complex systems.
- Small physical size, shorter project time.
- High speed of operation.
- Ability to communicate with computer systems in the plant.
- Ease of maintenance /troubleshooting.
- Reduced space.
- Energy saving.

Disadvantages of PLCs

- PLC devices are proprietary it means that part or software of one manufacturer can't be used in combination with parts of another manufacturer.
- Limited design and cost option
- Fixed Circuit Operations.
- PLCs manufacturers offer only closed architectures.

Applications:

- Wherever automation is desired the PLCs are best suited to meet the task.

- Few examples of industries where PLCs are used :
  1) Robots manufacturing and control
  2) Car park control
  3) Train control station system
  4) Food processing
  5) Materials handling
  6) Machine tools
  7) Conveyer system etc.

TROUBLE SHOOTING FEATURES

- Install sinking and sourcing inputs and outputs properly—one wrong wire and it won't work
- Implement safety circuits correctly in PLC applications to protect people and equipment
- Prevent noise, heat, and voltage variations from ruining your PLC system
- Implement a step-by-step static and dynamic start-up checkout to guarantee smooth PLC system operation
- Design preventive safety and maintenance into your total control system

TROUBLE SHOOTING FEATURES

- Learn no-nonsense troubleshooting procedures to reduce downtime
- Troubleshoot analog I/O and avoid undesirable count jumps
- Learn 6 preventive maintenance procedures to keep your PLC system running fault free
- Learn a step-by-step procedure for finding hidden ground loops
- Learn how to deal with leaky inputs
- Identify vibration problems and use them for preventive engineering control
- Control excessive line voltage and avoid intermittent shutdowns