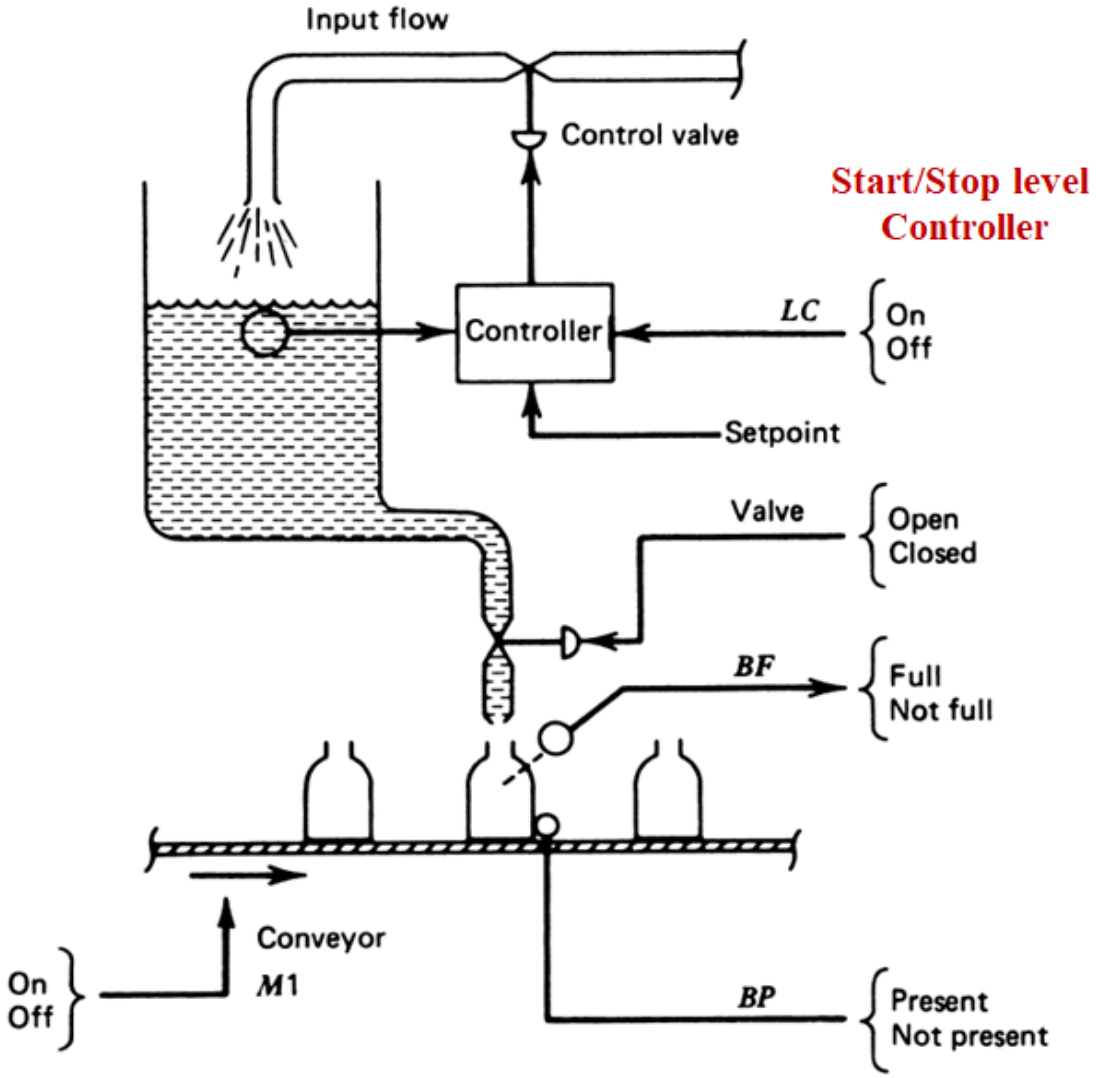
Discrete State Process Control using Programmable Logic Controllers

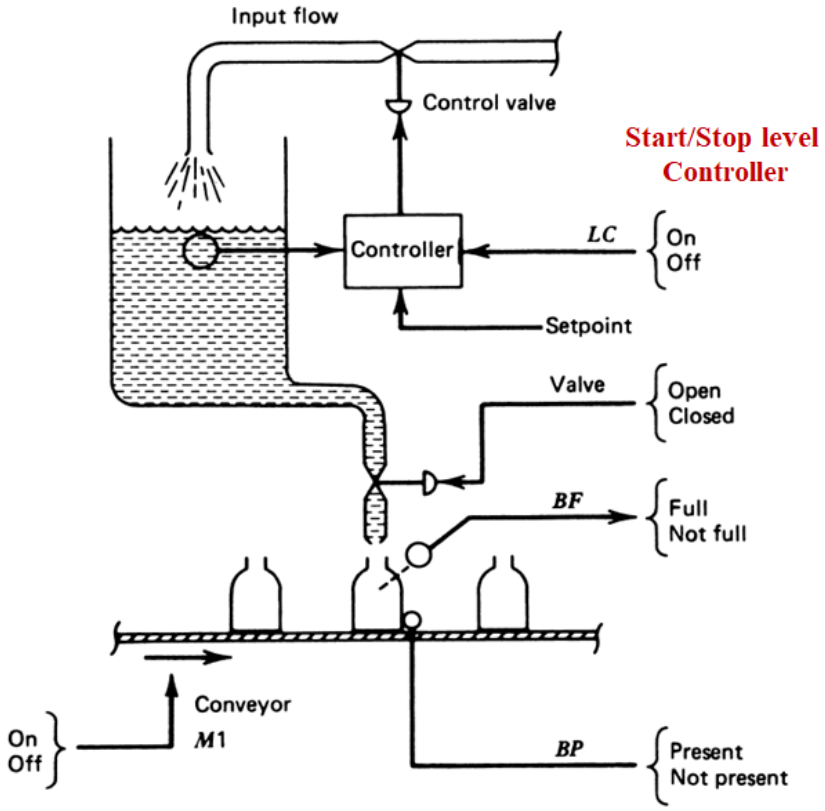
Ladder Diagram for Automatic Bottling system

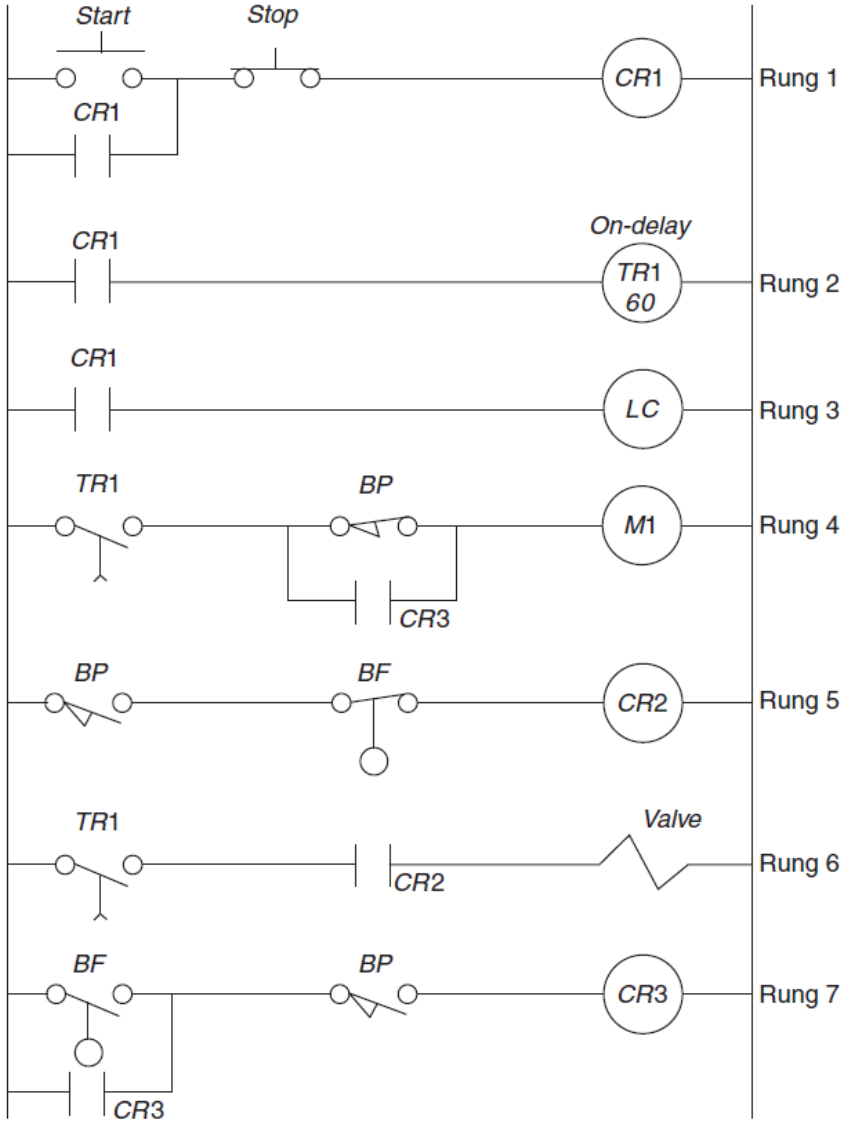


Consider the tank with the fluid and consider that periodically a bottle comes into position under the outlet valve and is filled. Also the level must be maintained at the setpoint *while the outlet valve is opened and the bottle filled.* This requirement may be necessary to ensure a constant pressure head during bottle filling. This process will require that a continuous-level control system be used to adjust the input flow rate during bottle-fill through the output valve. The continuous control system will be turned on or off just as would a valve or motor or other discrete device.

"Assume that when the level-control system is commanded off, input valve is closed and a 1-minute prefill is required for initialization"

Ladder Diagram for Automatic Bottling system





Valve should be opened when the bottle is not filled and is present

Valve should be closed when the bottle is filled and is present

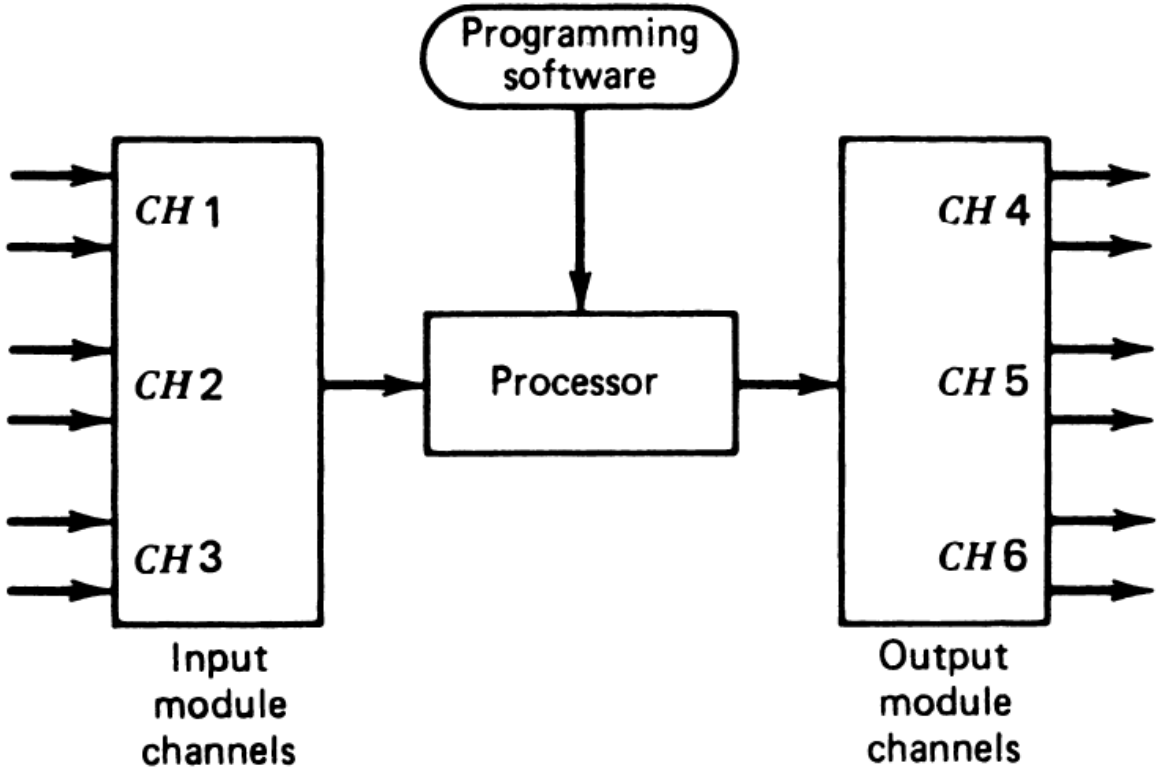
Programmable Logic Controllers

"The modern solution for the problem of how to provide discrete-state control is to use a computer-based device called a *programmable controller* (PC) or *programmable logic controller* (PLC)"

The move from relay logic controllers to computer-based controllers was because of the following reasons:

1. The input and output variables of discrete-state control systems are binary in nature, just as with a computer.
2. Many of the "control relays" of the ladder diagram can be replaced by software, which means less hardware failure.
3. It is easy to make changes in a programmed sequence of events when it is only a change in software.
4. Special functions, such as time-delay actions and counters, are easy to produce in software.
5. The semiconductor industry developed solid-state devices that can control high power ac/dc in response to low-level commands from a computer, including SCRs and TRIACs.

Programmable Logic Controllers

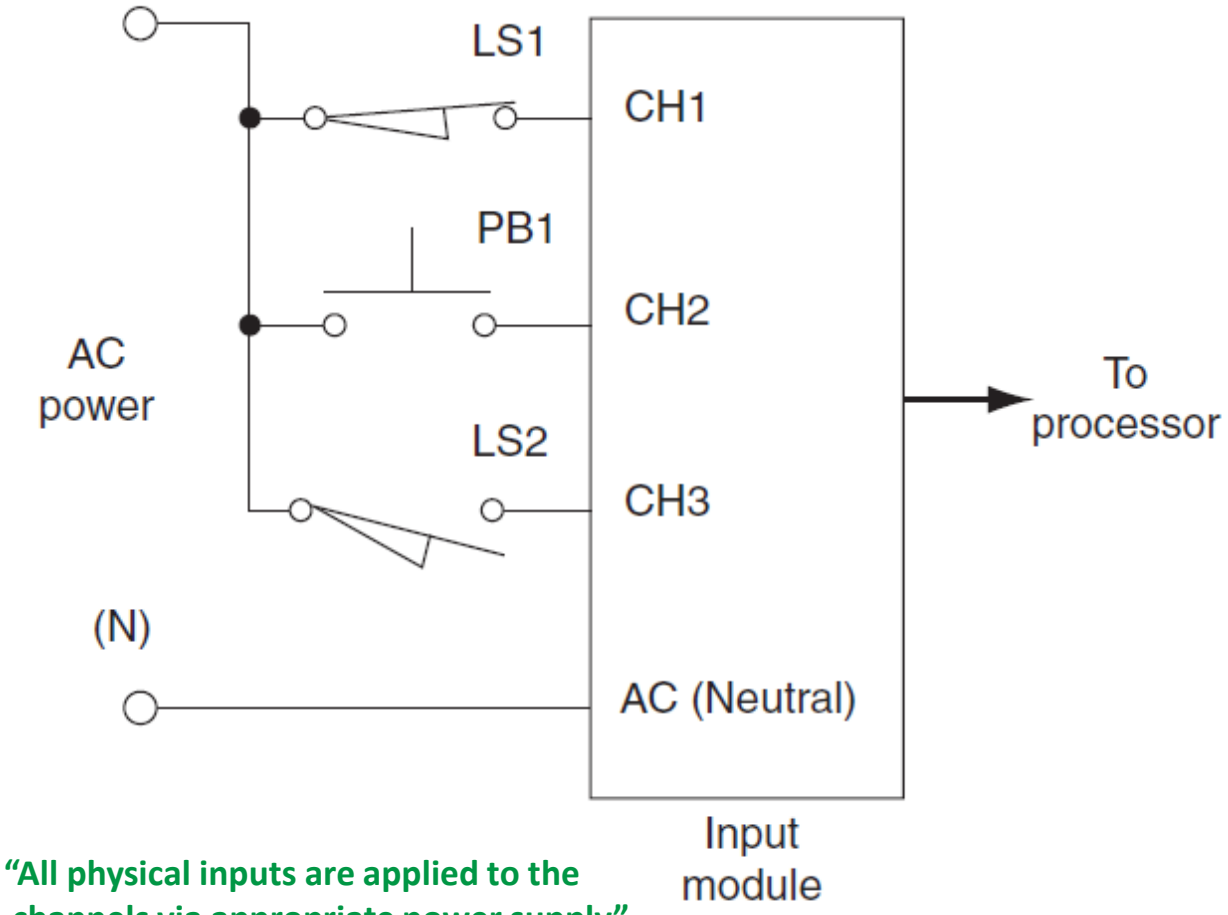


Basic structure of the programmable logic controller (PLC)

Processor:
The processor is a computer that executes a program to perform the operations specified in a ladder diagram or a set of Boolean equations.
The processor performs arithmetic and logic operations on input variable data and determines the proper state of the output variables.

Programmable Logic Controllers

Typical wiring to a PLC input module

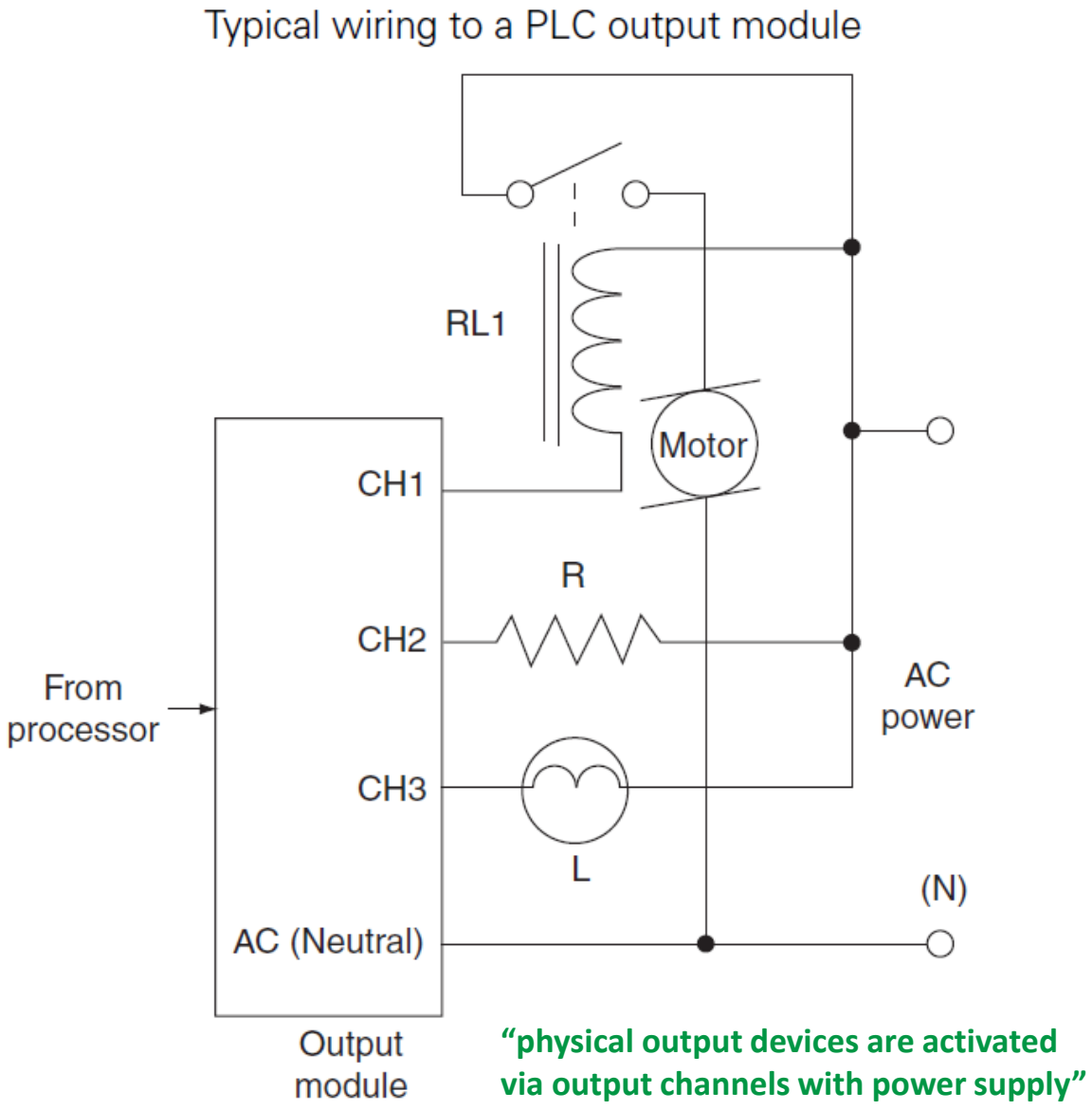


"All physical inputs are applied to the channels via appropriate power supply"

Input Modules:

The input modules examine the state of physical switches and other input devices and put their state into a form suitable for the processor. The PLC is able to accommodate a number of inputs, **called *channels.*** As shown in Figure, If a switch such as PB1 was closed, the ac voltage was impressed on the input module that would internally provide a conversion to a digital signal (**1** or **0**) as required by the computer-based PLC processing unit.

Programmable Logic Controllers

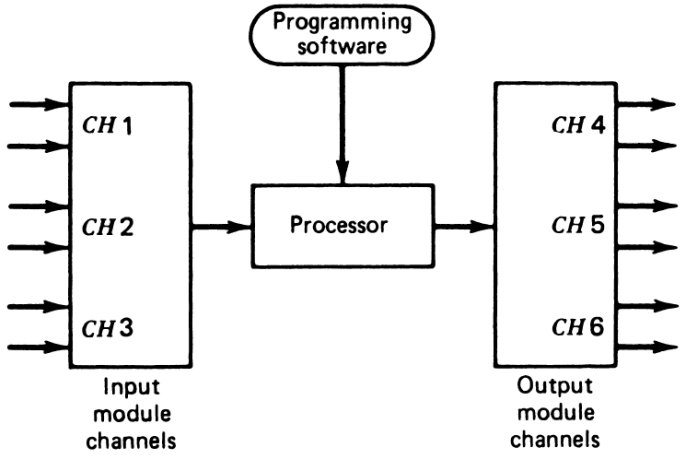


"physical output devices are activated via output channels with power supply"

Output Modules:

- The objective of the output module is to supply power to an external device such as a motor, light, solenoid, and so on, as required by the ladder diagram.
- The output of the PLC is a high (**1**) or low (**0**) signal of low power dc. Thus, the modern output module is designed to input the processors digital output and use this to activate relays, SCRs, TRIACs, BJTs, and other solid-state switches that can handle the high-power ac requirements.
- In this case the PLC output module can be wired directly to the output device and the ac power source, as shown in Figure. This shows how the channels of the output module are connected between the ac power and common.
- If the device, such as the motor in Figure, requires a current that exceeds the module output capability, then external relays or solid-state switches can be used.

Programmable Logic Controller Operation



Basic structure of the programmable logic controller (PLC)

The operation of the programmable controller can be considered in two modes namely:

1. I/O scan mode
2. Execution mode

I/O Scan Mode:

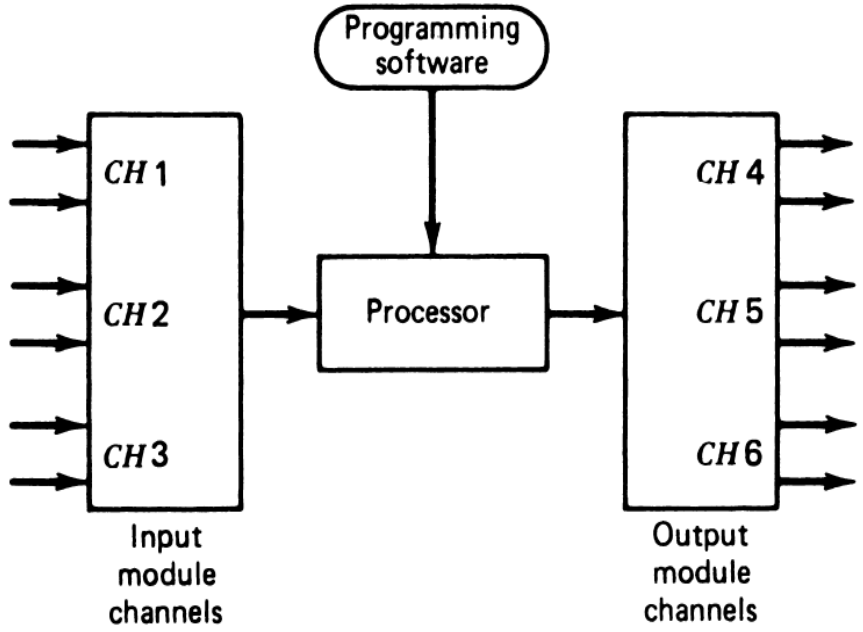
During the I/O scan mode, the processor updates all outputs and inputs the state of all inputs one channel at a time. The time required for this depends on the speed of the processor.

Execution Mode:

- During this mode, the processor evaluates each rung of the ladder diagram program that is being executed sequentially, starting from the first rung and proceeding to the last rung. As a rung is evaluated, the last known state of each switch and relay contact in the rung is considered, and if any TRUE path to the output device is detected, then that output is indicated to be energized, that is, set to ON.
- At the end of the ladder diagram, the I/O mode is entered again, and all output devices are provided with the ON or OFF state determined from execution of the ladder diagram program. All inputs are sampled, and the execution mode starts again.

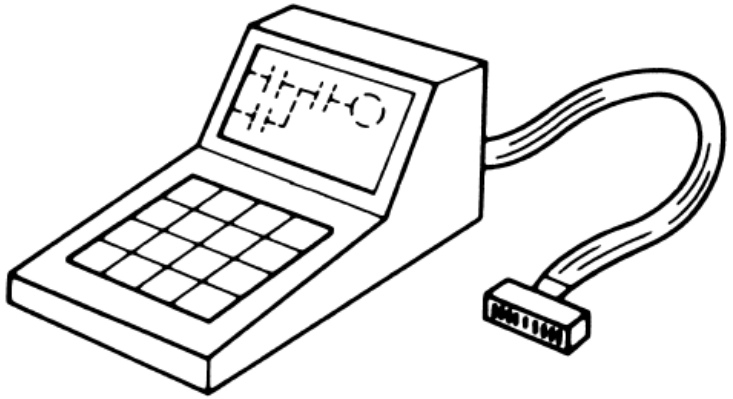
Programmable Logic Controller: Scan Time

- An important characteristic of the programmable controller is **how much time is required for one complete cycle of I/O scan and execution**. This depends on how many input and output channels are involved and on the length of the ladder diagram program. A typical maximum scan/execution time is 5 to 20 ms.
- The speed of the controller depends on the clock frequency of the processor. The higher the clock frequency, the greater the speed, and the faster the scan/execution time.
- The **length of time for one scan consists of three parts:** (1) input time, (2) execution time, and (3) output time. Most of the scan time comes from the execution phase.
- The scan time may have an impact on the ability of the PLC to detect events that occur on the inputs. For example, if some limit switch goes to the ON state for less than a scan time, it may be missed by the PLC.



Basic structure of the programmable logic controller (PLC)

Programmable Logic Controller: Programming Unit

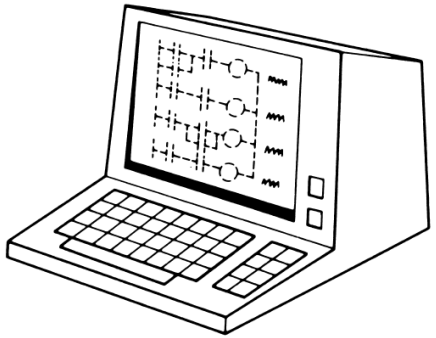


Hand-held PLC programming unit.

- The programming unit is an external electronic package that is connected to the programmable controller when programming occurs. The unit usually allows input of a program in ladder diagram symbols. The unit then transmits that program into the memory of the programmable controller.

- In this unit, the ladder diagram is displayed one rung at a time in a special liquid crystal display (LCD). The user can enter a program, perform diagnostic tests, run the program through the programmable controller, and perform editing of the installed program. The installed program is stored in a temporary memory that will be lost without ac power or battery backup. The program can be permanently “burned” into a ROM for final installation.

Programmable Logic Controller: Programming Unit



Desktop terminal PLC programming unit

- Once the program has been debugged, the programming unit can be disconnected, and the programmable controller can now operate the process according to the ladder diagram program. There is the danger of loss of the program because of power failure, but this can be prevented by placing the program into the permanent memory.

- The dedicated PLC programming unit can be replaced by software and I/O boards on common desktop or laptop personal computers. **Special software allows the pc to program the PLC and then download the program into the PLC unit.**

- Modern plants employ **computer network technology to link PLCs and other control hardware via network communication systems**. These systems may use custom installations of Ethernet-type local area networks (LANs) or dedicated control networks such as. In any case, it is often possible to download the PLC programming from a development computer to the **Foundation Fieldbus or Profibus** the PLC in which it will be used. The network can also be used to edit an existing PLC program and to transfer process data to a network computer to analyze plant operation.

Programmable a PLC

- The programmable controller has **no "real" relays or relay contacts**. The only real devices are those that are actually part of the process being controlled—that is, limit switches, motors, solenoids, and so on. We continue to use symbols for relays and relay contacts, even though they are software symbols.

- When the ladder diagram for some event sequence was developed previously, each switch device, output device, and relay was referred to by a label. For example, *CR*1 referred to control relay 1, and the contacts for that relay were referred to by the same label. Other designations included *LS*1 for a limit switch, *M*1 for a motor relay, and so on. The programmable controller uses a similar method of identifying devices, but it is referred to as the **device *address* or *channel.***

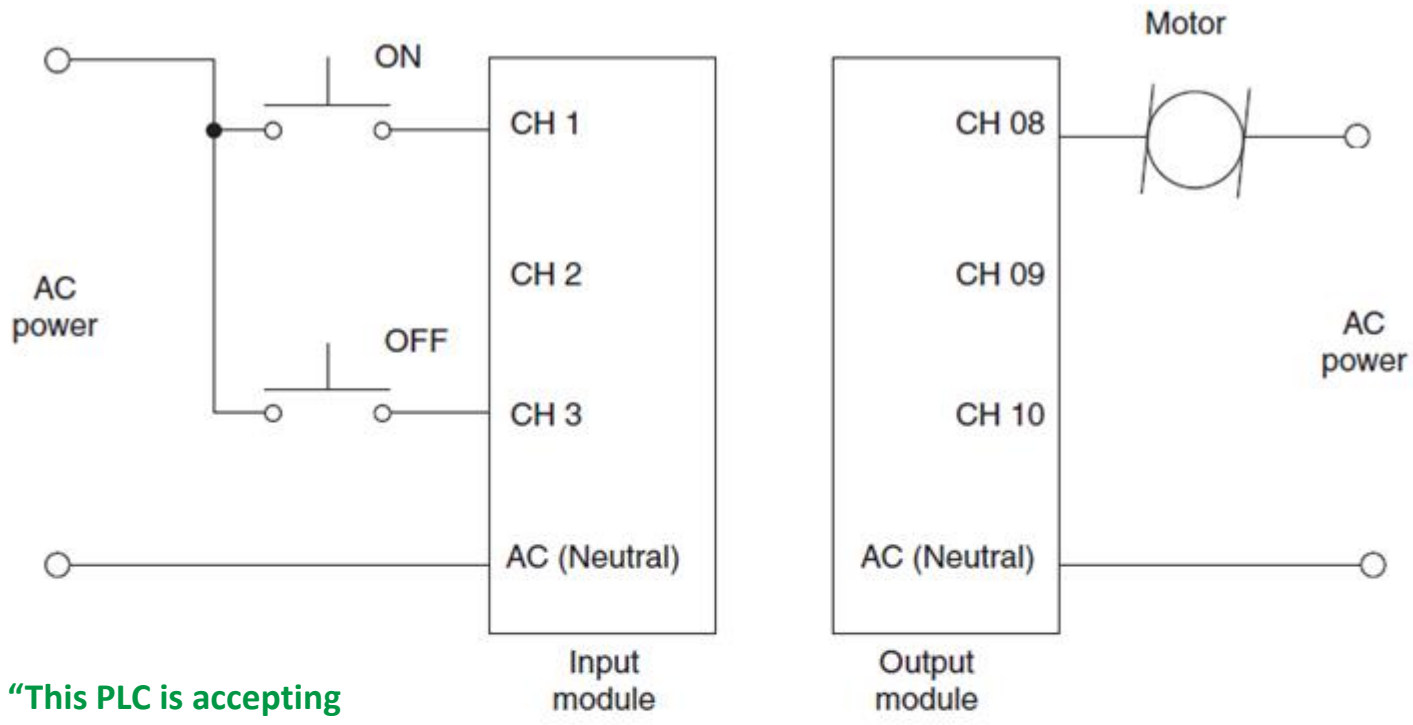
PLC addressing

Function	Address
Input channels	00 to 07
Output channels	08 to 15
Internal relays	16 to 31
Timers and counters	32 to 39

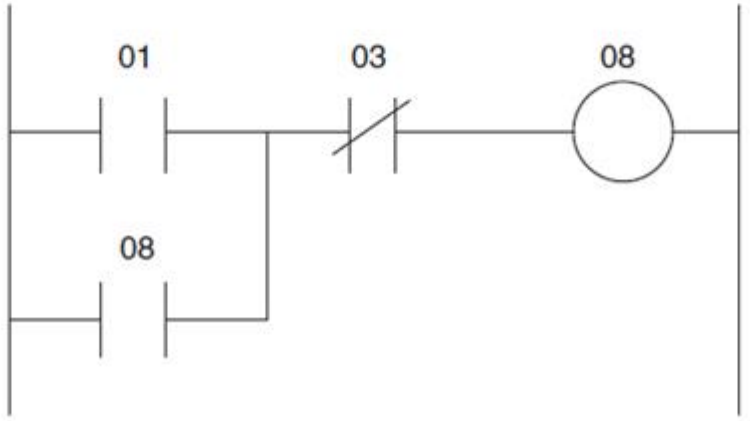
Programmable a PLC

PLC addressing

Function	Address
Input channels	00 to 07
Output channels	08 to 15
Internal relays	16 to 31
Timers and counters	32 to 39



"This PLC is accepting AC power input for I/O channels"



Physical connections to the PLC (a) and the programmed ladder diagram (b) for a motor start/stop system

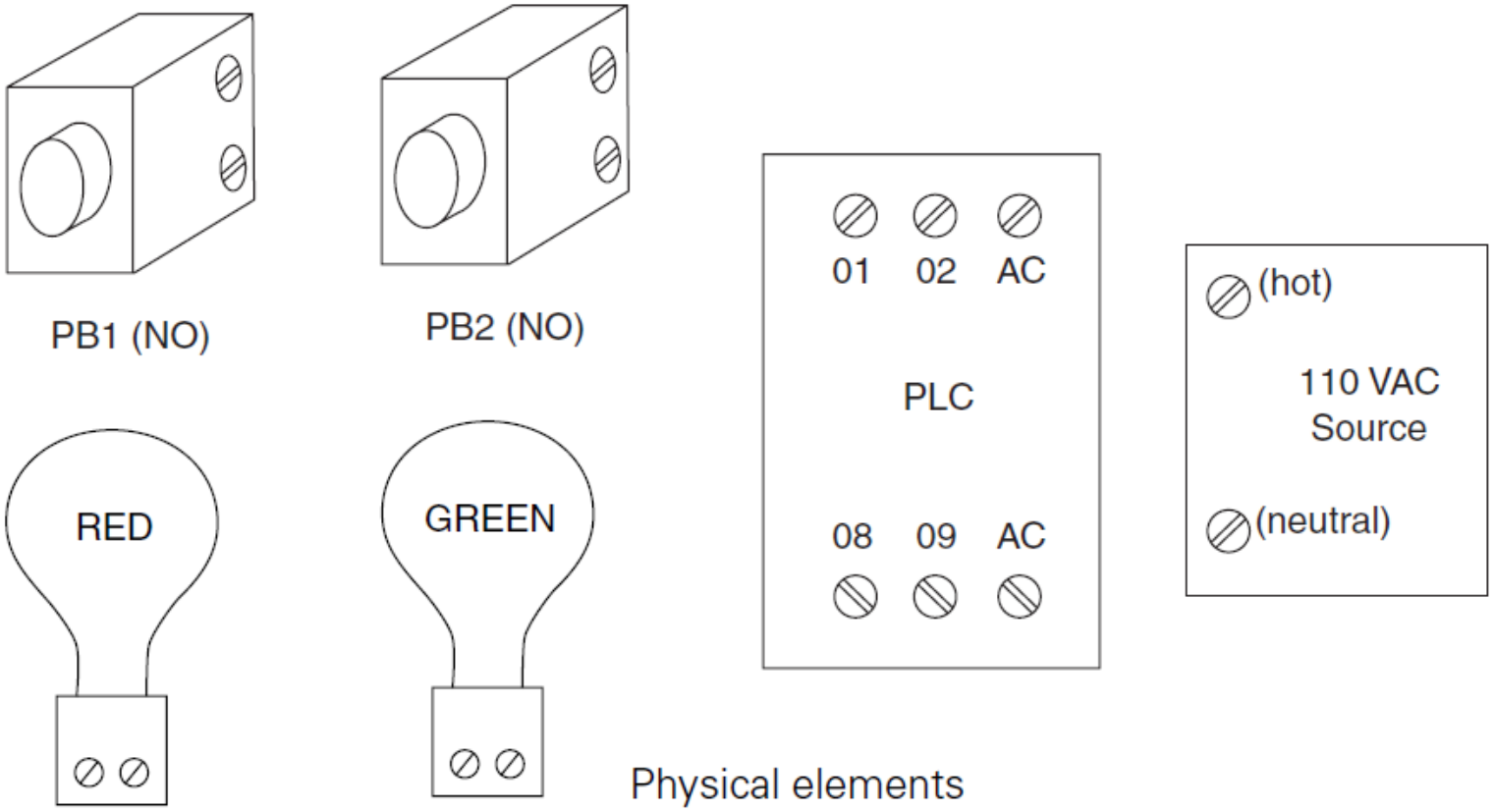
Programmable a PLC

Objective:

1. If *PB*1 alone is pushed, the red light turns on.
2. If *PB*2 alone is pushed, the green light turns on.
3. If both buttons are pushed at once, neither light turns on.

PLC addressing

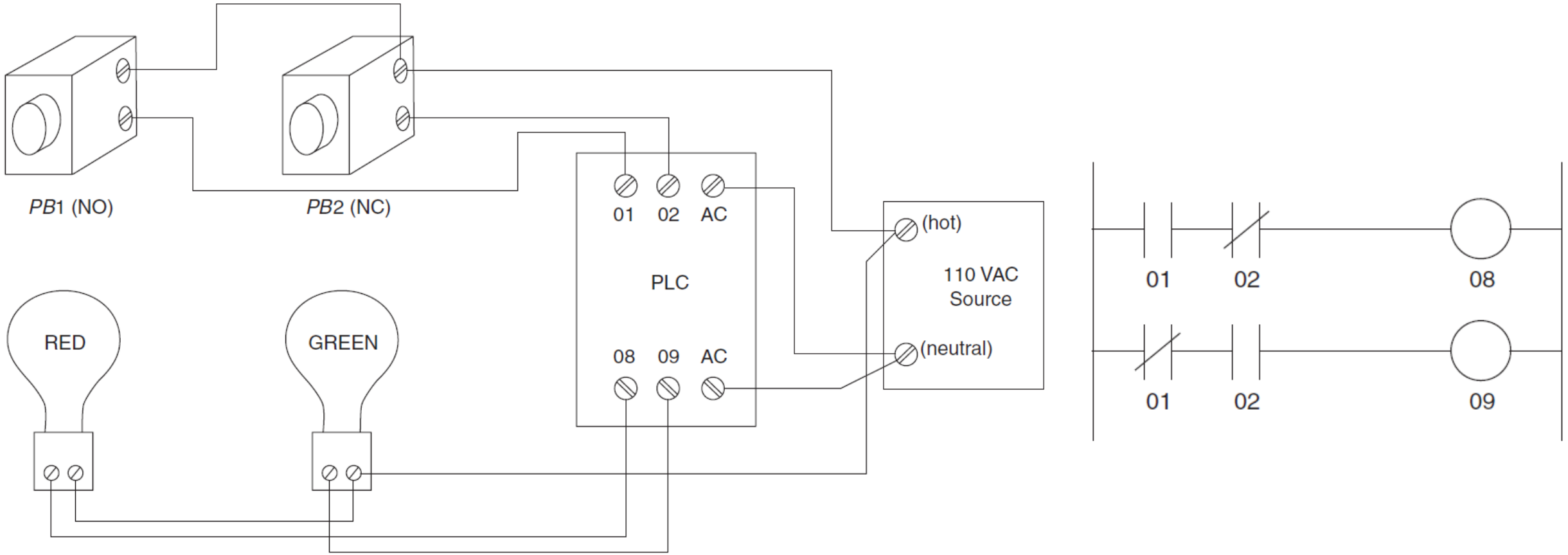
Function	Address
Input channels	00 to 07
Output channels	08 to 15
Internal relays	16 to 31
Timers and counters	32 to 39



Physical elements

Programmable a PLC

"This PLC is accepting AC power input for I/O channels"



PLC extends the neutral supply to the bulbs, when the output ports:
08 and 09 are activated

Programmable a PLC : Software functions with an Example

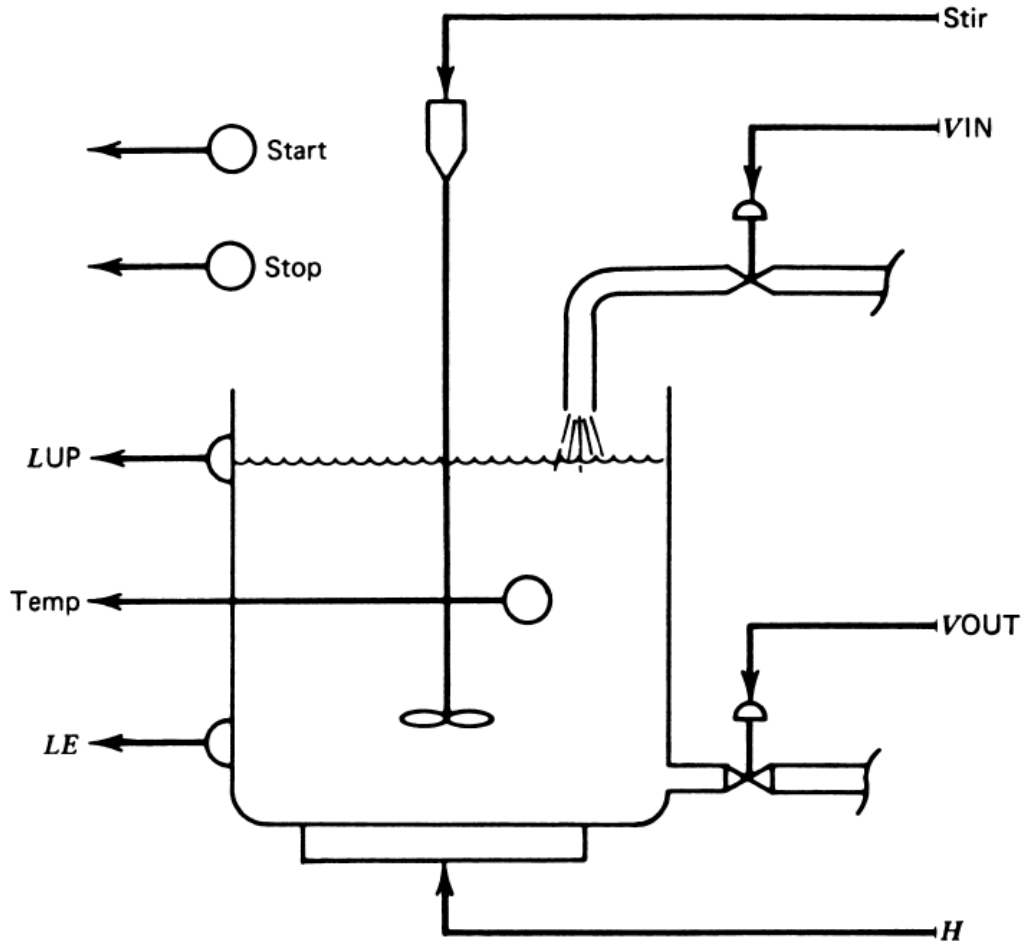
Description: Prepare the physical and programmed ladder diagram for the control problem shown in Figure below. The global objective is to heat a liquid to a specified temperature and keep it there with stirring for 30 min.

The hardware has the following characteristics:

1. START push button is NO, STOP is NC.
2. NO and NC are available for the limit switches.

The event sequence is

1. Fill the tank.
2. Heat and stir the liquid for 30 min.
3. Empty the tank.
4. Repeat from step 1.



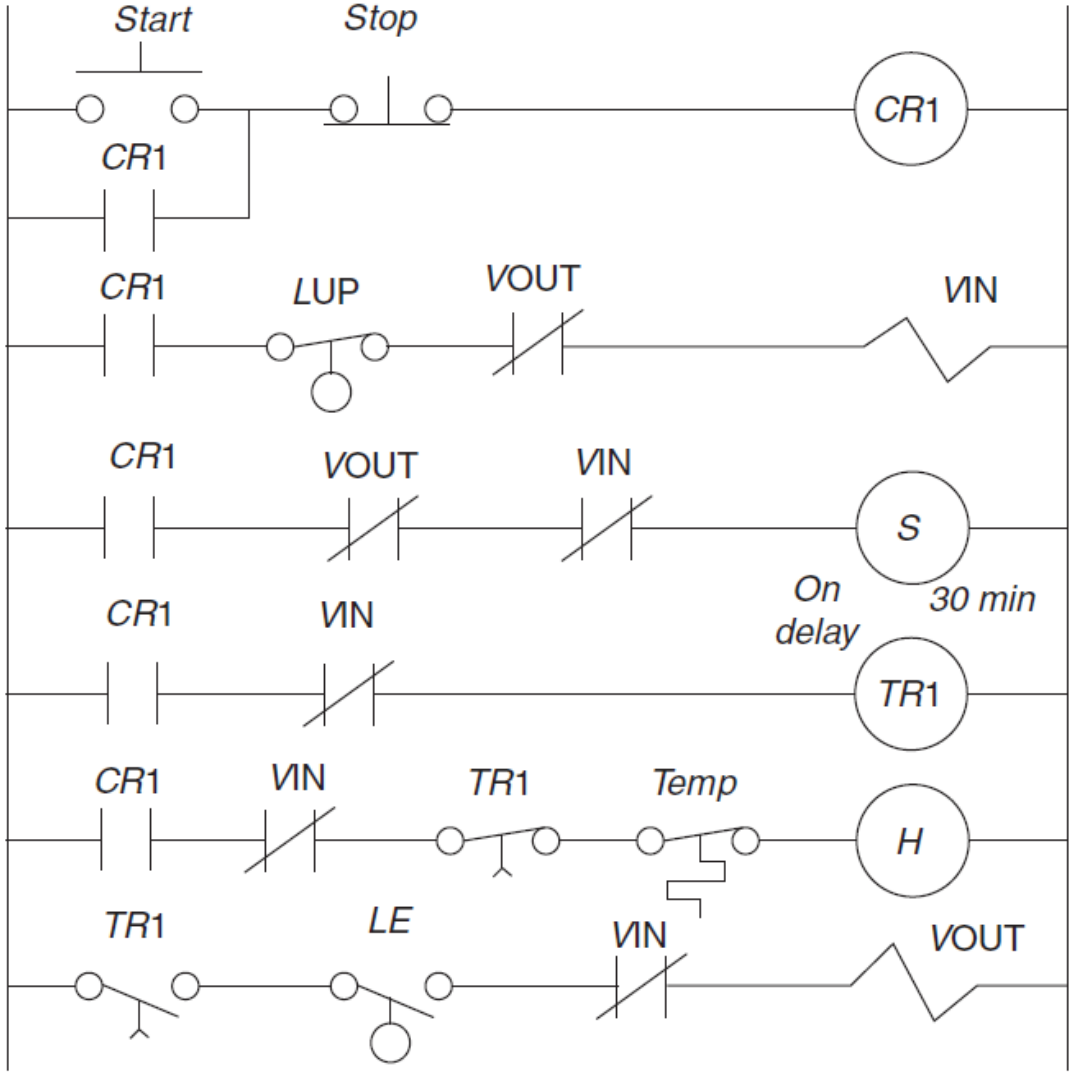
Programmable a PLC : Software functions with an Example

The hardware has the following characteristics:

1. START push button is NO, STOP is NC.
2. NO and NC are available for the limit switches.

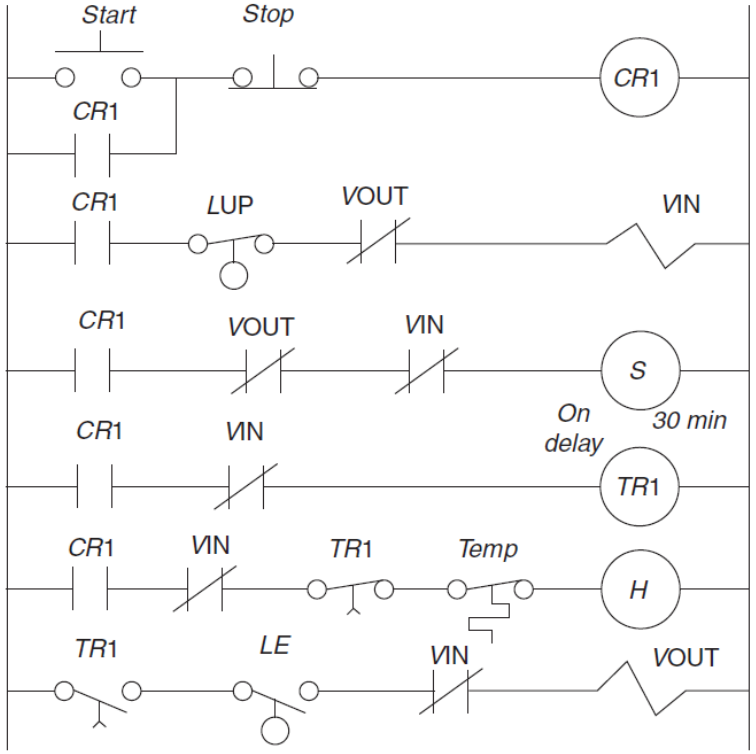
The event sequence is

1. Fill the tank.
2. Heat and stir the liquid for 30 min.
3. Empty the tank.
4. Repeat from step 1.



Physical Ladder Diagram

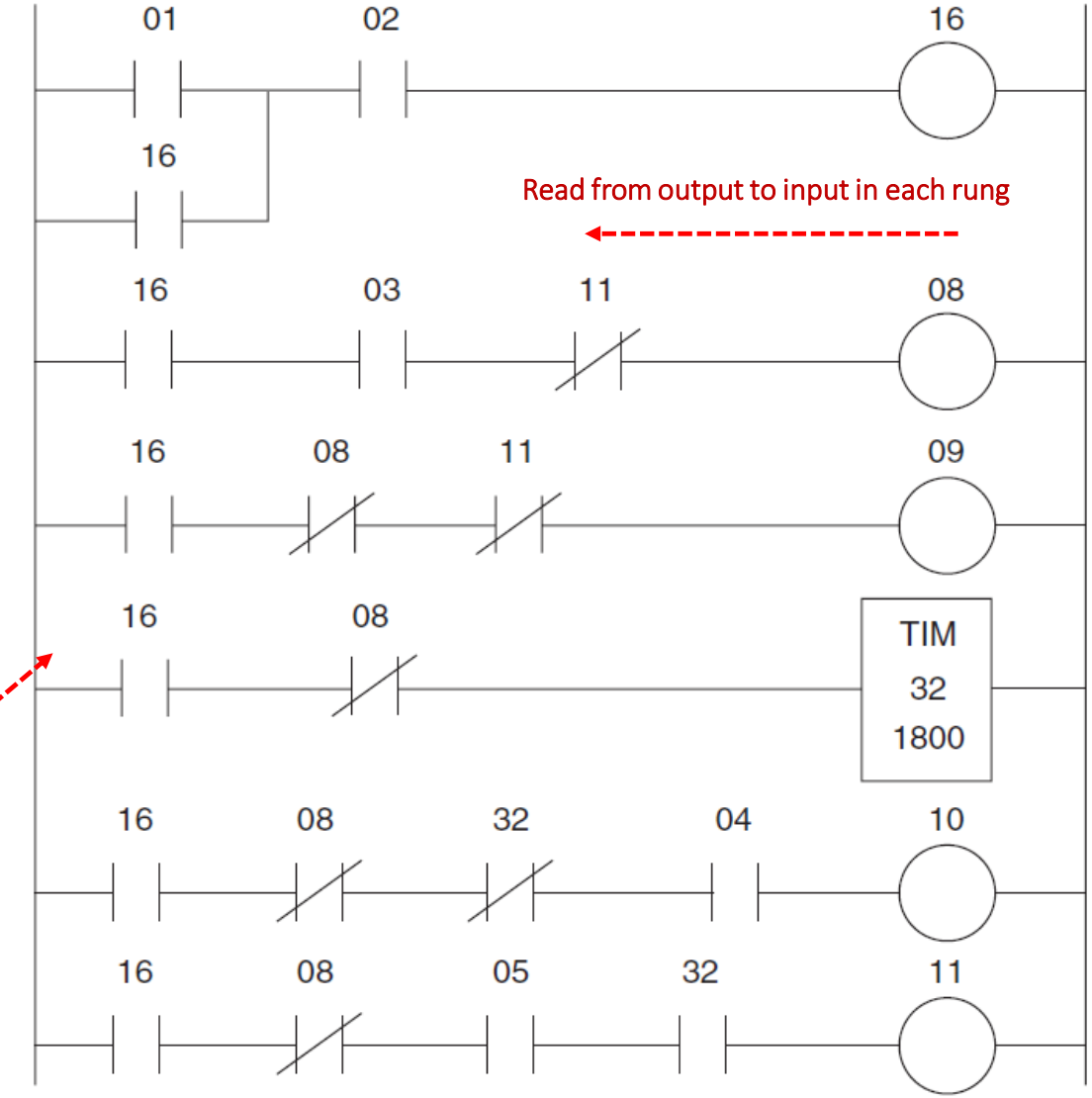
Programmable a PLC : Software functions with an Example



Physical Ladder Diagram

PLC addressing

Function	Address
Input channels	00 to 07
Output channels	08 to 15
Internal relays	16 to 31
Timers and counters	32 to 39



Programmed ladder Diagram

Inputs:
01 = NO Start switch
02 = NC Stop switch
03 = Full level switch (opens on rising level)
04 = Temperature switch (opens on rising temperature)
05 = Empty switch (closes on rising level)

Outputs:
08 = Input valve
09 = Stir motor
10 = Heater
11 = Output valve

Timer tick = 1 second

THANK YOU