Automation

Dr. Mohammed Abu mallouh

E-mail: mmallouh@hu.edu.jo

Office: E3045

Textbook: Petruzella, Frank D. (2005). Programmable Logic Controllers. McGraw Hill Companies Inc.

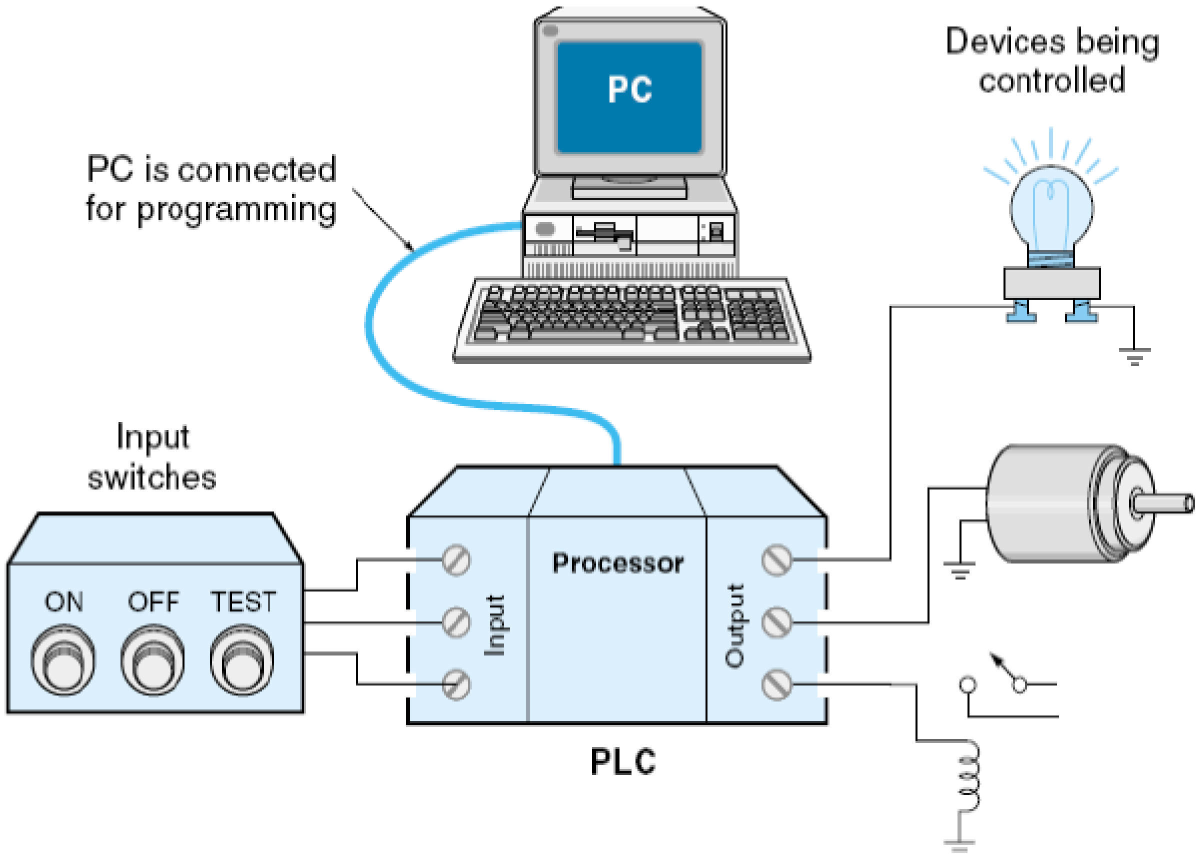
references:

1-notes from Dr.* Jeff Jackson *the university of Alabama

2-notes from Dr. Radu Muresan University of Guelph

Programmable Logic Controllers (PLCs)

- A programmable logic controller (PLC) is a specialized computer used to control machines and processes.



Example PLCs

Allen-Bradley PLC5

Allen-Bradley SLC500

Allen-Bradley Micrologix

Allen-Bradley Picocontroller

PLC Size Classification

utomation

Criteria

- number of inputs and outputs (I/O count)
- cost
- physical size

Allen-Bradley SLC-500 Family
- handles up to 960 I/O points

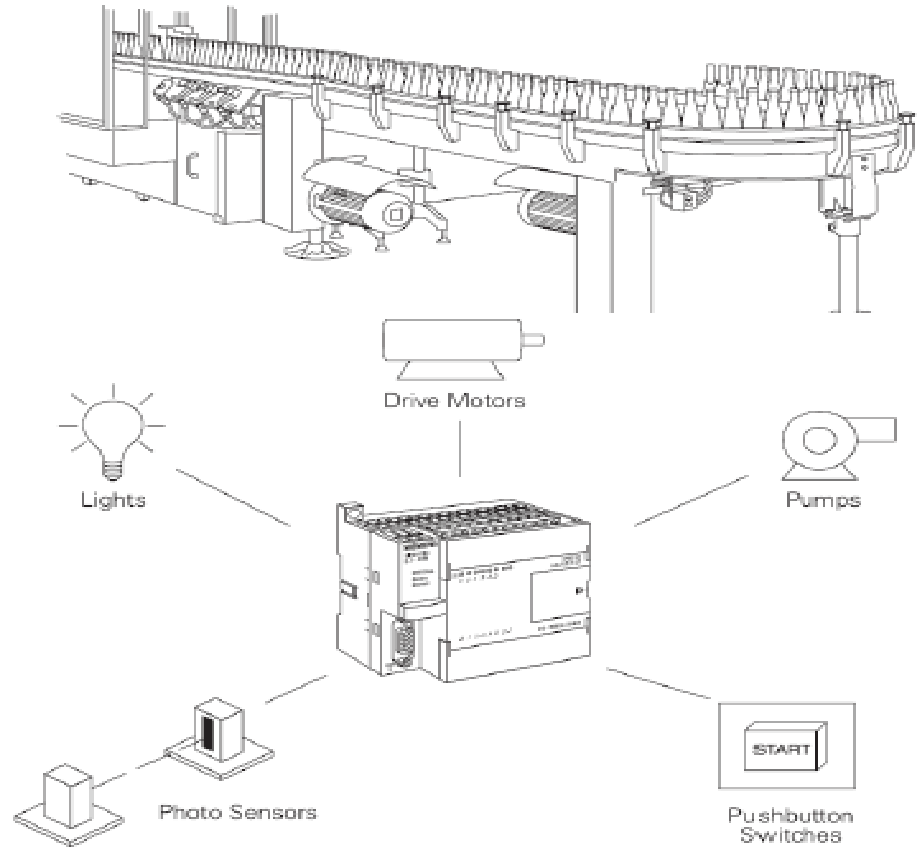
Micro PLC
- handles up to 32 I/O points

Nano PLC
- smallest sized PLC
- handles up to 16 I/O points

Programmable Logic Controllers (PLCs)

•It uses a programmable memory to store instructions and execute specific functions that include On/Off control, timing, counting, sequencing arithmetic and data handling

• Its purpose is to monitor crucial process parameters and adjust process operations accordingly

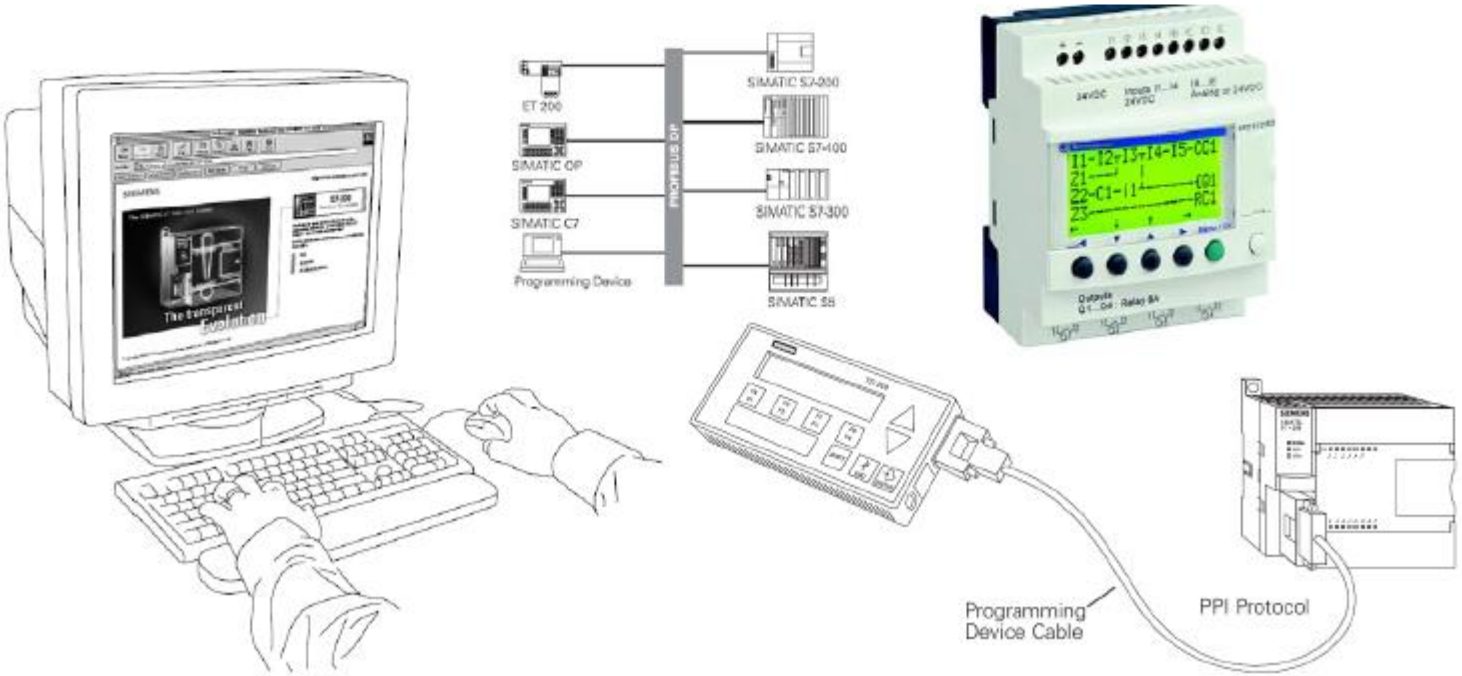


Programmable Logic Controllers (PLCs)

- Used extensively because the PLC
  - Is easy to set up and program
  - Behaves predictably
  - Ruggedized

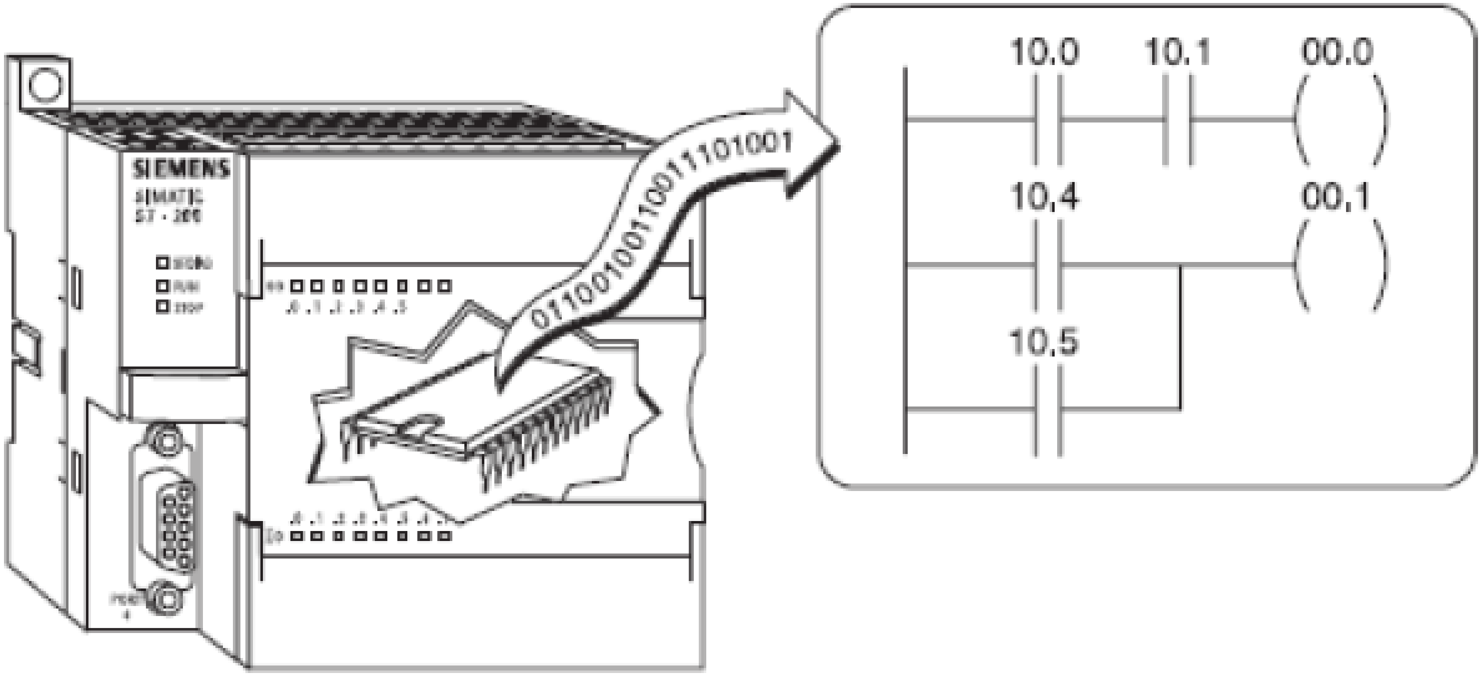
• It can be programmed, controlled, and operated by a person unskilled in operating (programming) computers.

• Essentially, a PLC's operator draws the lines and devices of ladder diagrams with a keyboard/mouse onto a display screen.

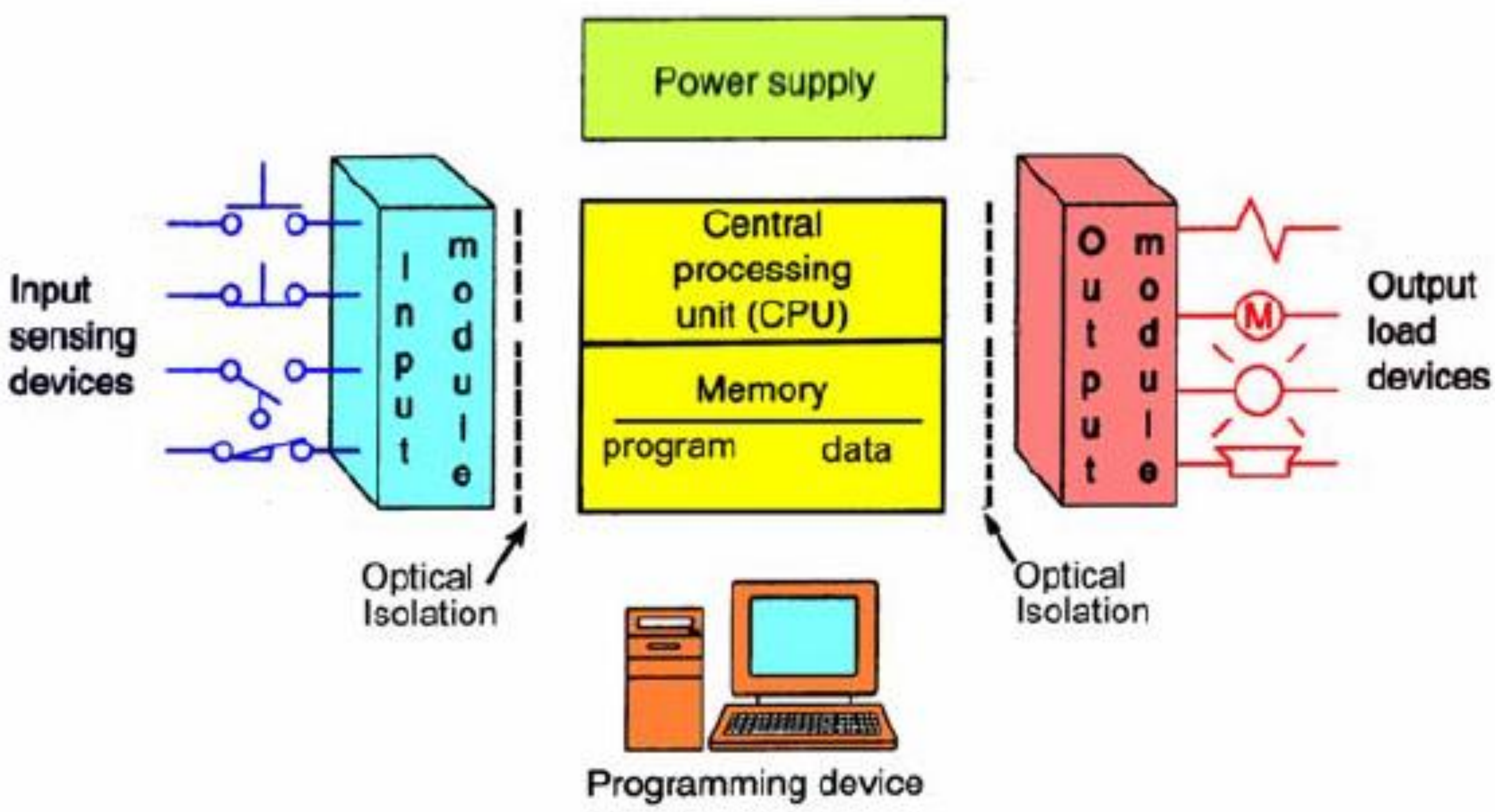


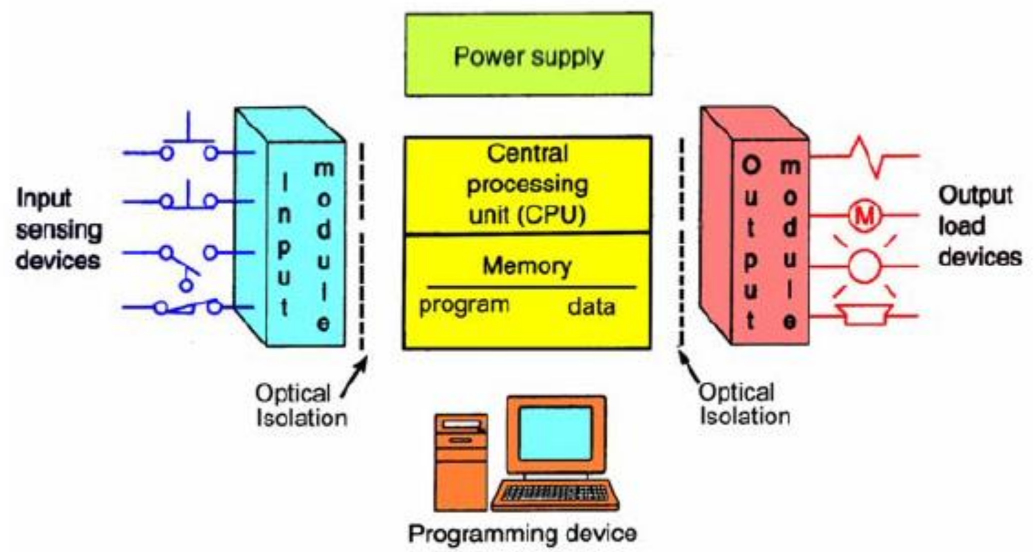
Programmable Logic Controllers (PLCs)

- The resulting ladder diagram is converted into computer machine language and run as a user program



PLC parts


Power supply
Central processing unit (CPU)
Memory
program
data
Input sensing devices
Input module
Output module
Output load devices
M
Optical Isolation
Optical Isolation
Programming device



Components in a PLC system

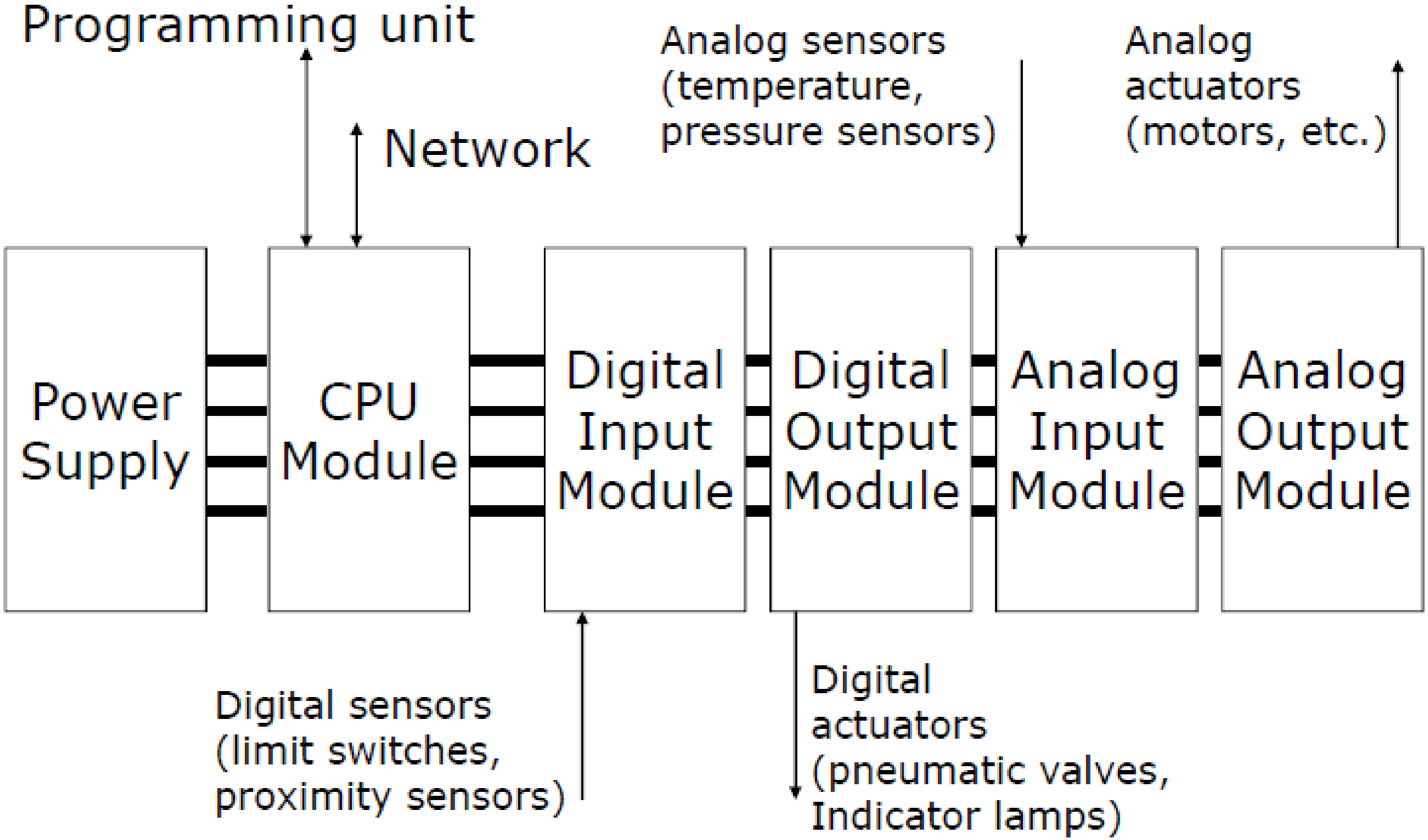
• CPU module, containing the processor and memory

• Input and output modules, to allow the PLC to read sensors and control actuators, a wide variety of types are available

• Power supply for the PLC, and often sensors and low power actuators connected to I/O modules

•A programming unit is necessary to create, edit and download a user program to the PLC

PLC in a automated system



PLC in Automated system

- The PLC takes the place of much of the external wiring required for control of a process.

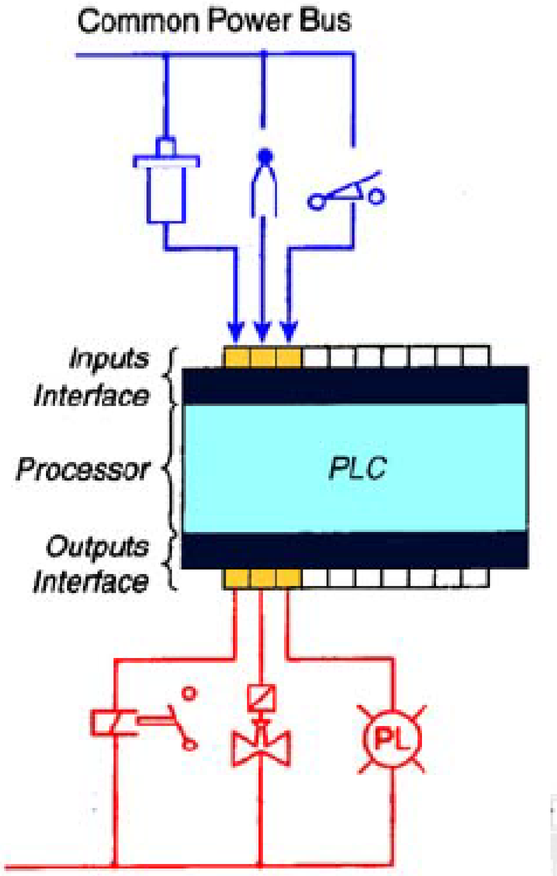
- The PLC will operate any system that has output devices that go on and off (known as discrete, or digital, outputs).

- It can also operate any system with variable (analog) outputs.

- The PLC can be operated on the input side by on-off devices (discrete, or digital) or by variable (analog) input devices.

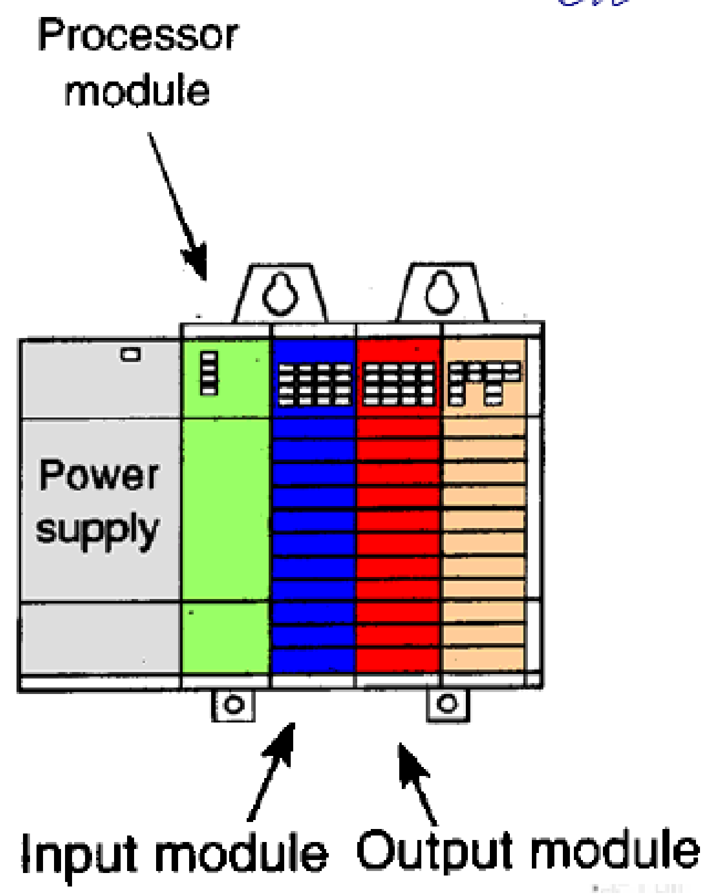
I/O Configurations

- Fixed I/O
  - is typical of small PLCs
  - comes in one package with no separate removable units
  - the processor and I/O are packaged together
  - lower in cost but lacks flexibility

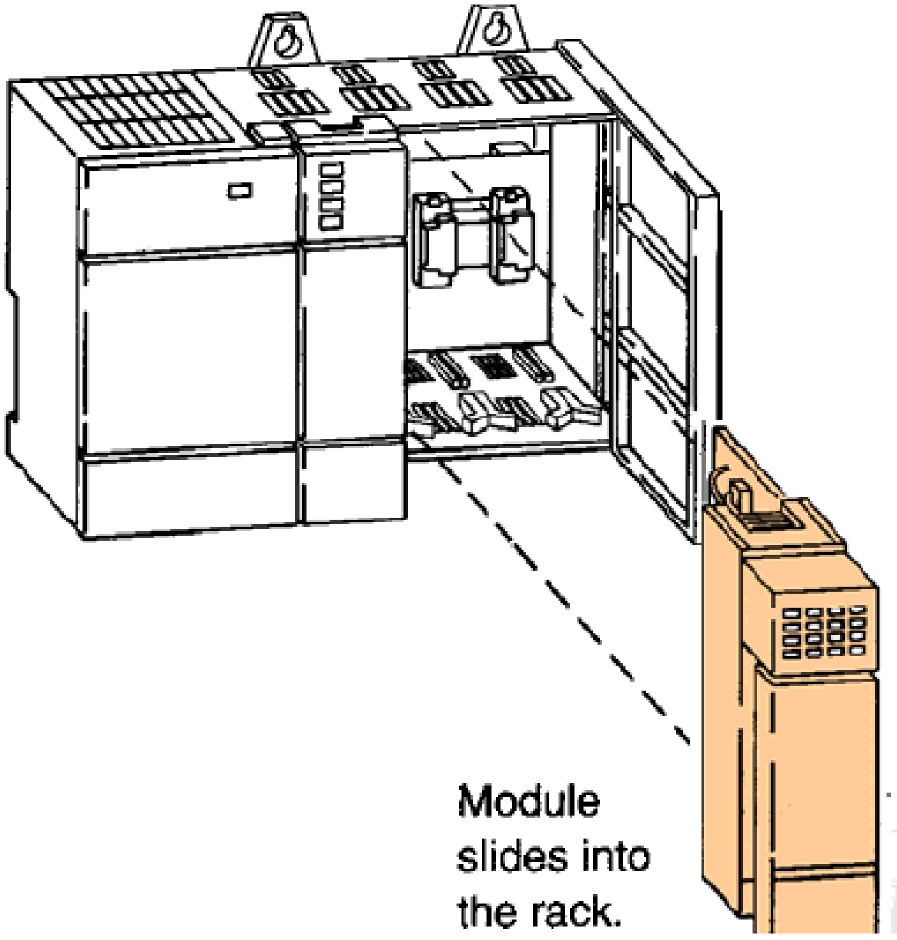


I/O Configurations

- Modular I/O
  - Is divided by compartments into which separate modules can be plugged.
  - This feature greatly increases your options and the unit's flexibility. You can choose from all the modules available and mix them in any way you desire.

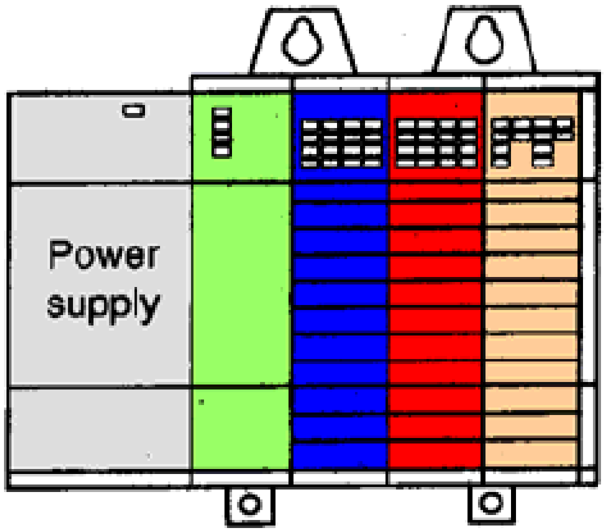


- Modular I/O
  - When a module slides into the rack, it makes an electrical connection with a series of contacts - called the *backplane*. The backplane is located at the rear of the rack.



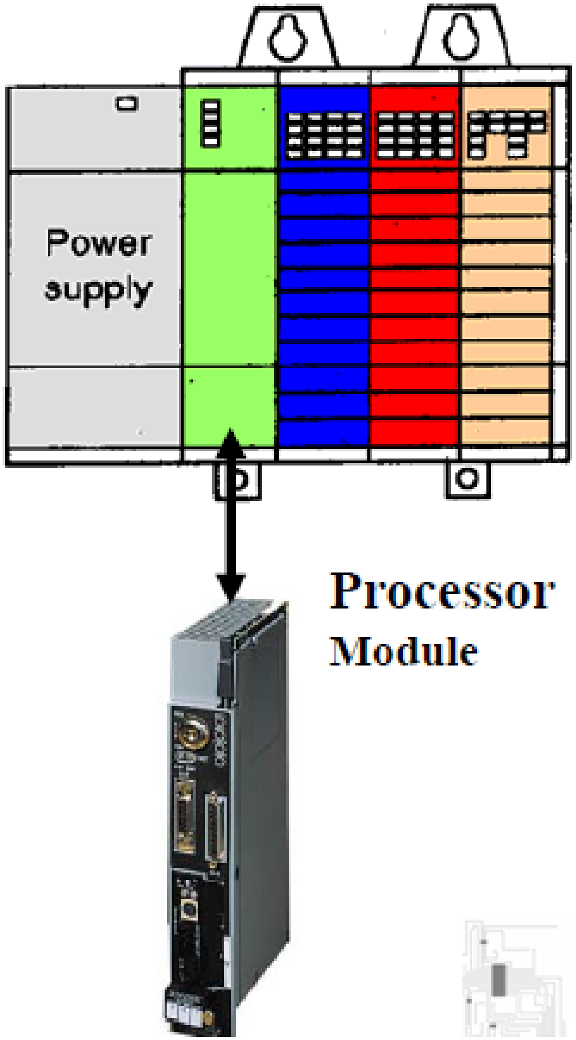
Power Supply

- Supplies DC power to other modules that plug into the rack.
- In large PLC systems, this power supply does not normally supply power to the field devices.
- In small and micro PLC systems, the power supply is also used to power field devices.



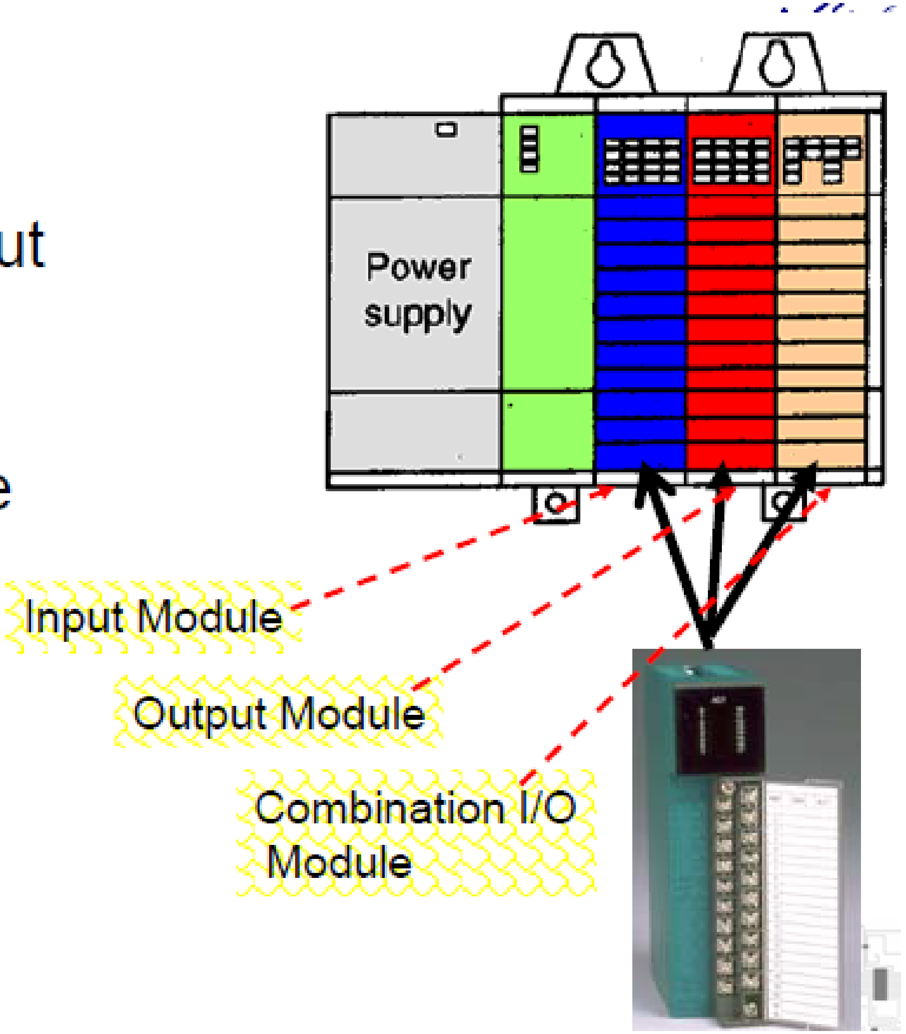
Processor (CPU)

- Consists of a microprocessor for implementing the logic, and controlling the communications among the modules.
- Designed so the desired circuit can be entered in relay ladder logic form.
- The processor accepts input data from various sensing devices, executes the stored user program, and sends appropriate output commands to control devices.



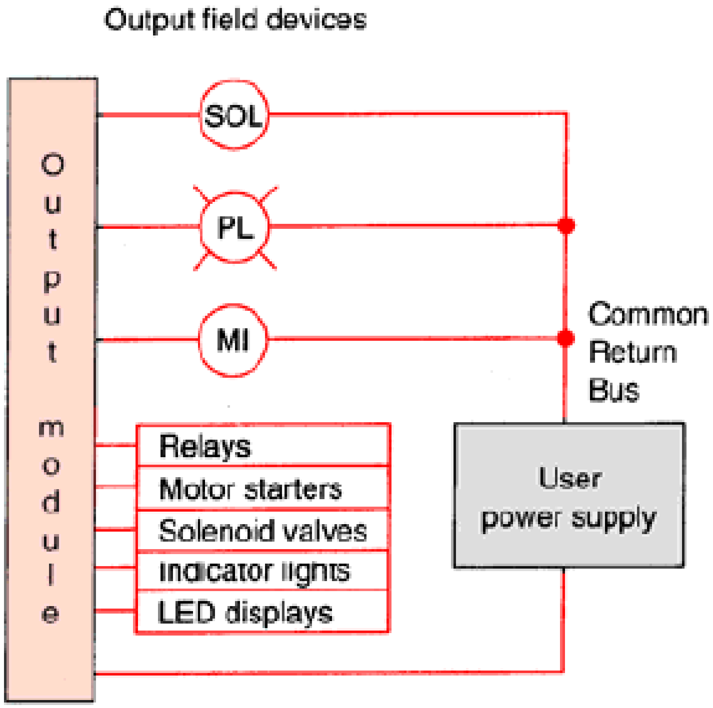
I/O Section

- Consists of input modules and output modules
- The I/O system forms the interface by which field devices are connected to the controller



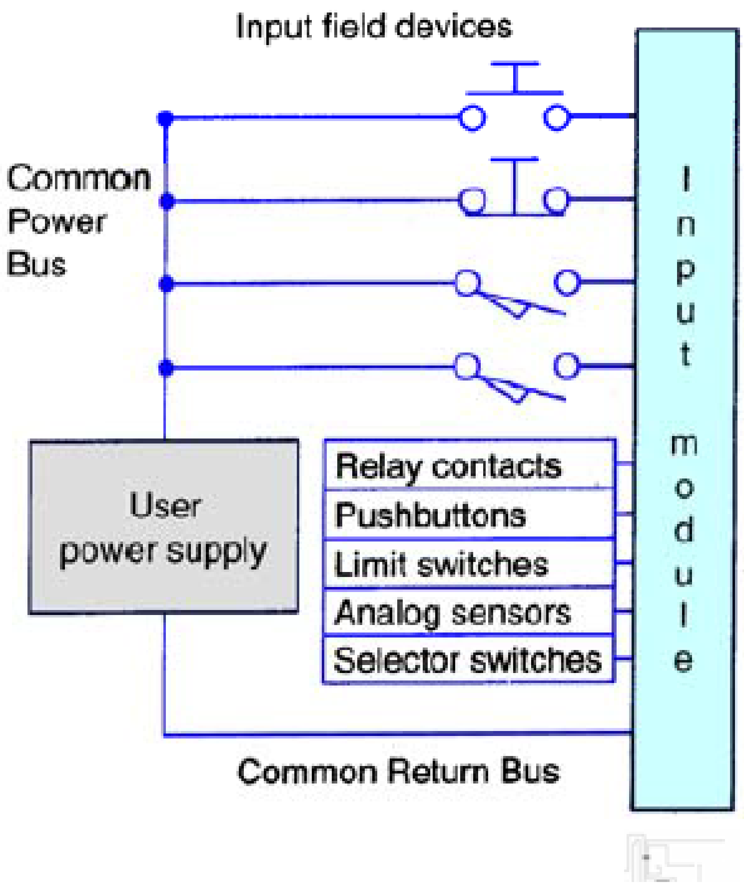
I/O Section

- Output modules
  - Forms the interface by which output field devices are connected to the controller.
  - PLCs employ an optical isolator which uses light to electrically isolate the internal components from the input and output terminals.

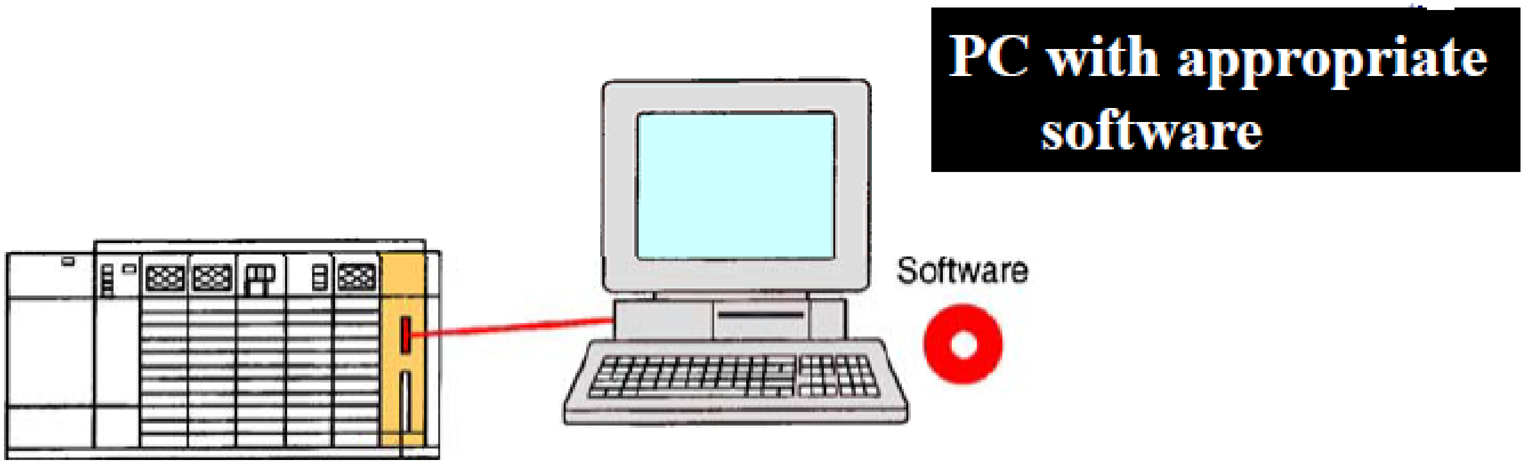


I/O Section

- Input module
  - Forms the interface by which input field devices are connected to the controller.
  - The terms “field” and “real world” are used to distinguish actual external devices that exist and must be physically wired into the system.



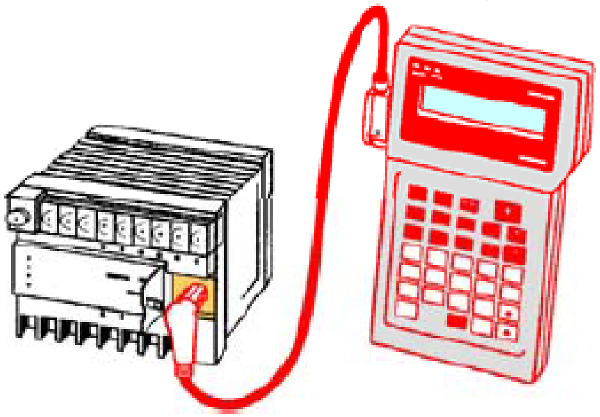
Programming Device



- A personal computer (PC) is the most commonly used programming device.
- The software allows users to create, edit, document, store and troubleshoot programs.
- If the programming unit is not in use, it may be unplugged and removed. Removing the programming unit will not affect the operation of the user program.

Programming Device

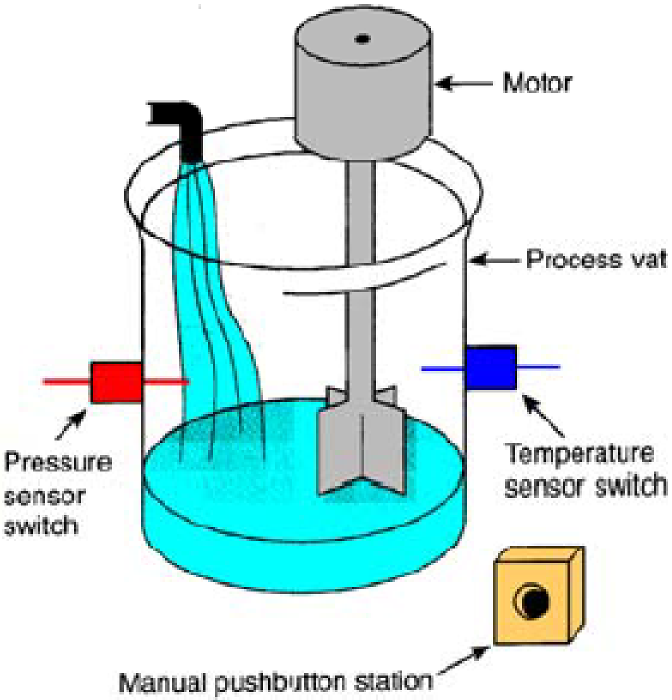
Hand-held unit with display



- Hand-held programming devices are sometimes used to program small PLCs.
- They are compact, inexpensive, and easy to use, but are not able to display as much logic on screen as a computer monitor.
- Hand-held units are often used on the factory floor for troubleshooting, modifying programs, and transferring programs to multiple machines.

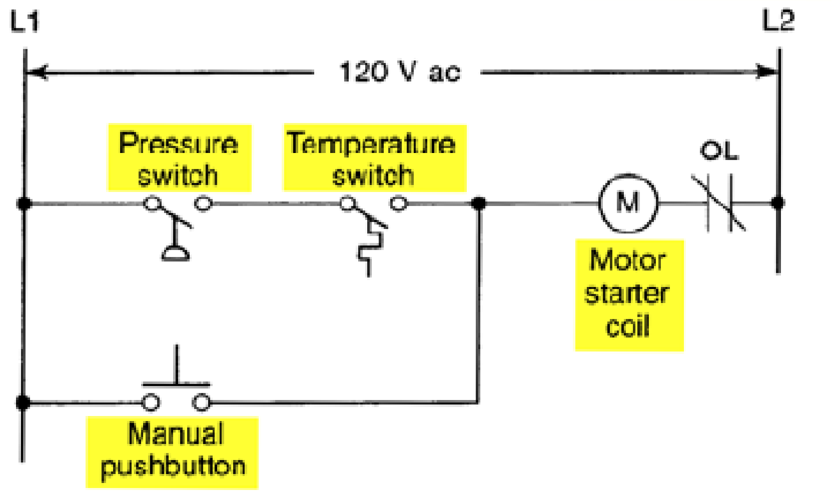
Principles of PLC Operations; Example: PLC Mixer Process Control Problem

- Mixer motor to automatically stir the liquid in the vat when the temperature and pressure reach preset values.
- Alternate manual pushbutton control of the motor to be provided.
- The temperature and pressure sensor switches close their respective contacts when conditions reach their preset values.



Process Control Relay Ladder Diagram

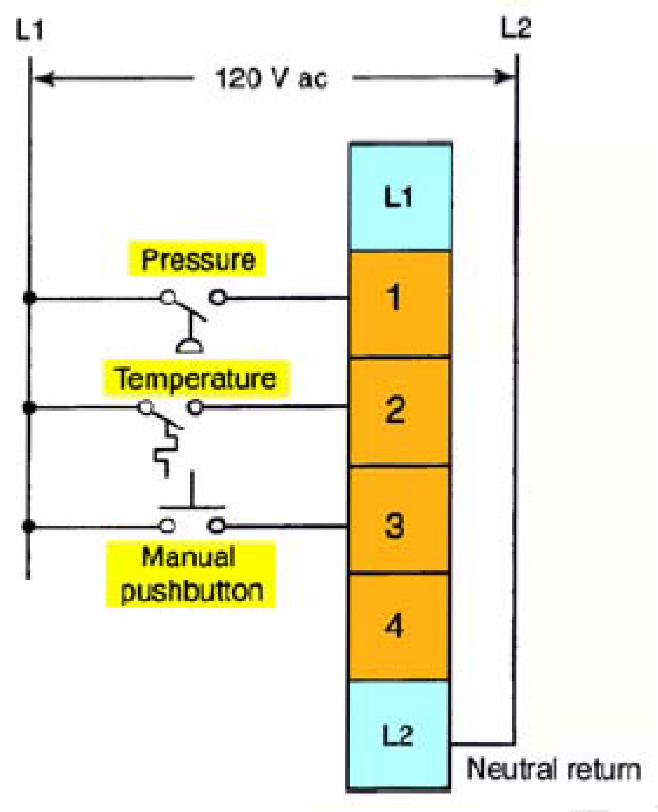
Radu Muresan



- Motor starter coil is energized when both the pressure and temperature switches are closed or when the manual pushbutton is pressed.

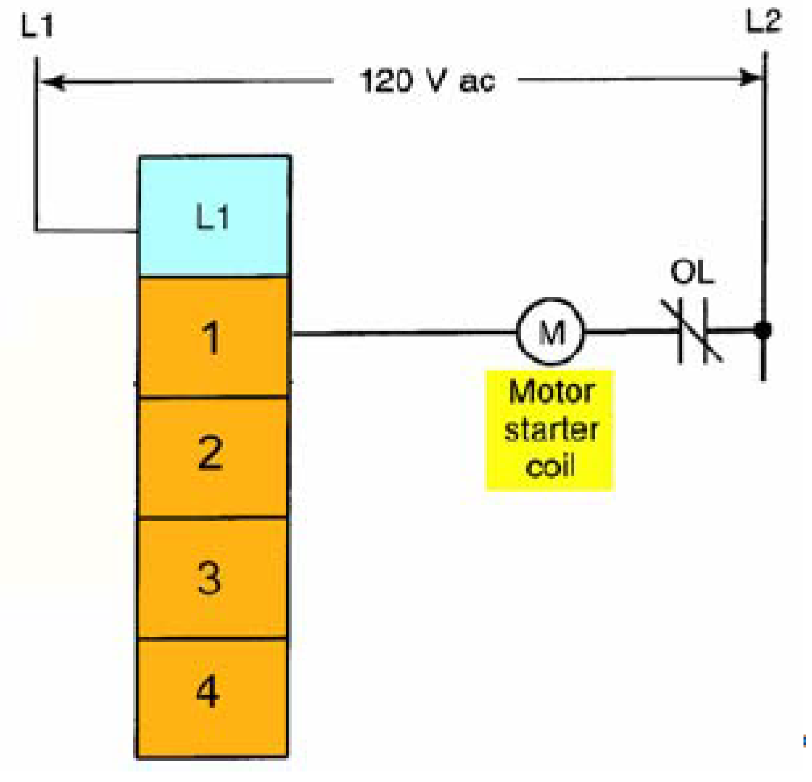
PLC Input Module Connections

- The same input field devices are used.
- These devices are wired to the input module according to the manufacturer's labeling scheme.

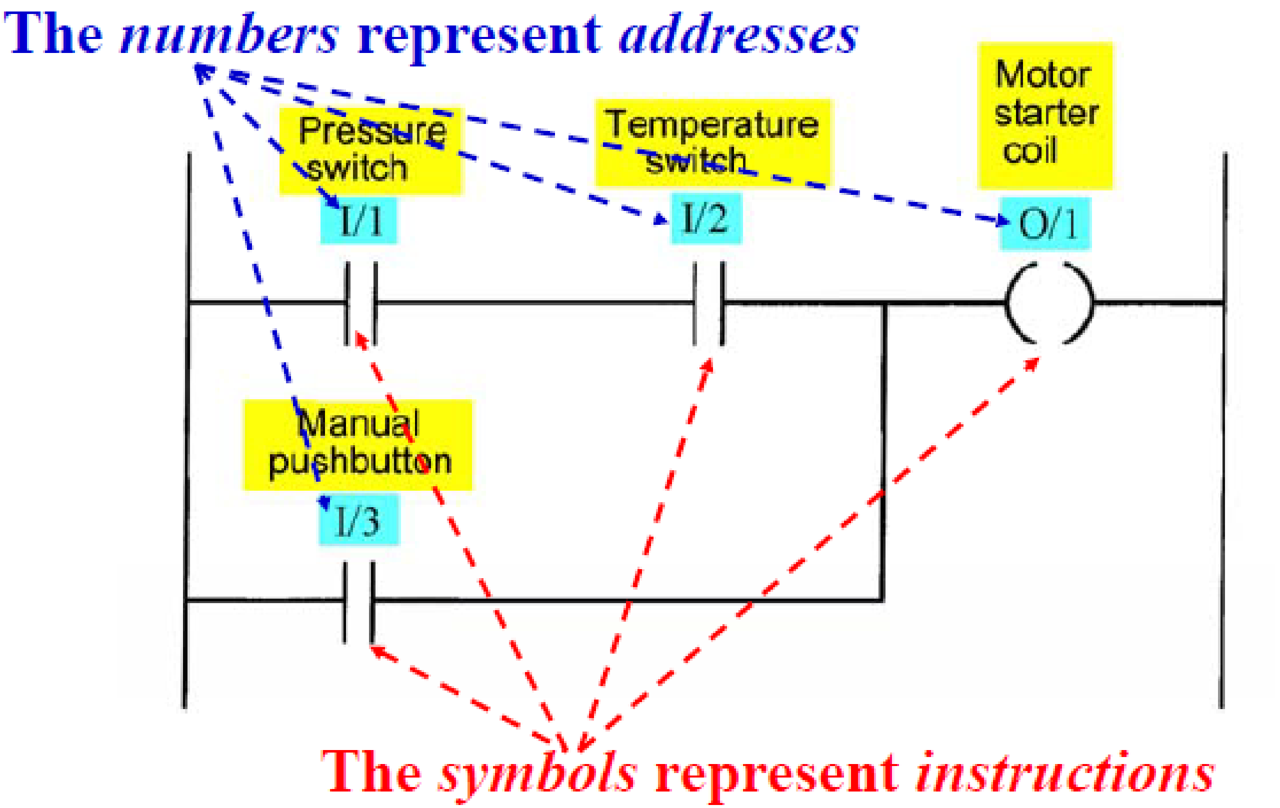


PLC Output Module Connections

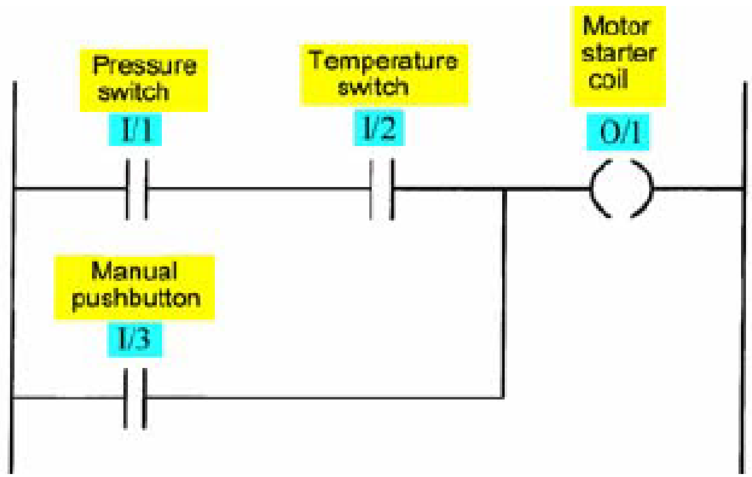
Same output field device is used and wired to the output module.



PLC Ladder Logic Program



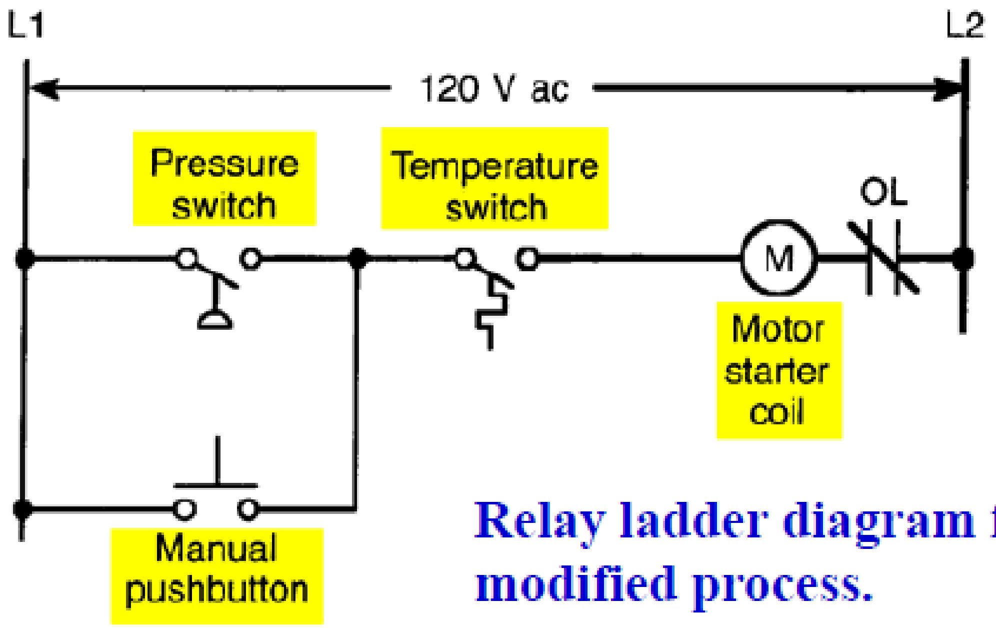
PLC Operating Cycle



Each -| |- can be though of as a set of normally open contacts

The -( )- can be considered to represent a coil that, when energized, will close a set of contacts.

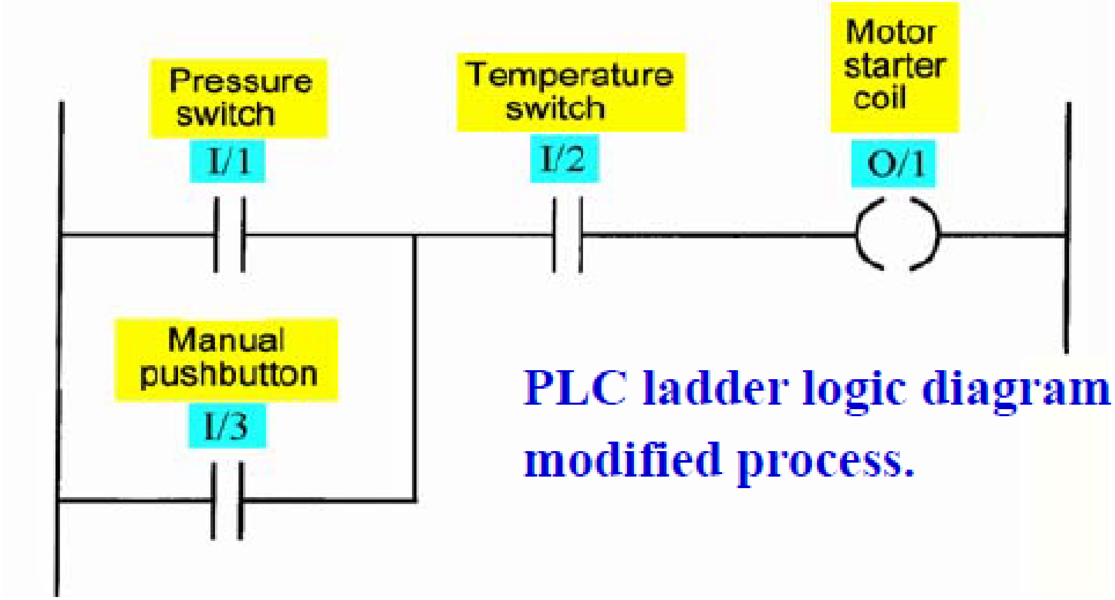
Modifying a PLC Program



Relay ladder diagram for modified process.

- The change requires that the manual pushbutton control should be permitted to operate at any pressure but not unless the specified temperature setting has been reached.
- If a relay system were used, it would require somere wiring of the system, as shown, to achieve the desired change.

Modifying a PLC Program



PLC ladder logic diagram for modified process.

- If a PLC is used, no rewiring is necessary!
  - The inputs and outputs are still the same.
- All that is required is to change the PLC program

History Of The PLC

- The first PLC systems evolved from conventional computers in the late 1960s and early 1970s.

- These first PLCs were installed primarily in automotive plants.

- Traditionally, the auto plants had to be shut down for up to a month at model changeover time.

- The early PLCs were used along with other new automation techniques to shorten the changeover time.

PLC Advantages

- **Flexibility**

– In the past, each different electronically controlled production machine required its own controller; 15 machines might require 15 different controllers.

– Now it is possible to use just one model of a PLC to run any one of the 15 machines.

– Furthermore, you would probably need fewer than 15 controllers, because one PLC can easily run many machines.

– Each of the 15 machines under PLC control would have its own distinct program.

PLC Advantages

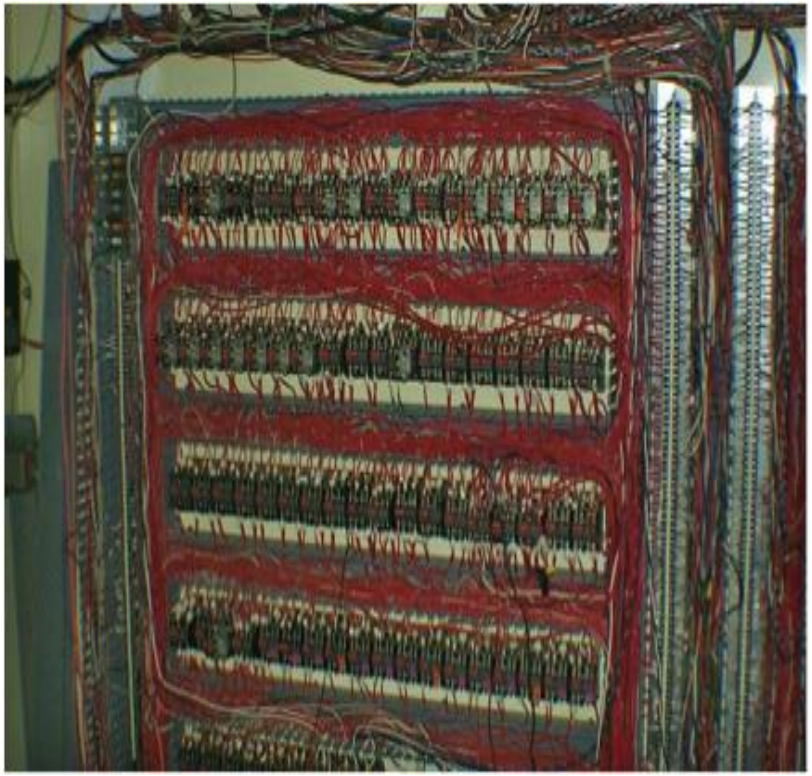
Implementing Changes and Correcting Errors

– With a wired relay-type panel, any program alterations require time for rewiring of panels and devices.

– When a PLC program circuit or sequence design change is made, the PLC program can be changed from a keyboard sequence in a matter of minutes.

– No rewiring is required for a PLC-controlled system.

– Also, if a programming error has to be corrected in a PLC control ladder diagram, a change can be typed in quickly.



Wired relay type panel

PLC type panel

Wired relay type panel

PLC type panel

PLC Advantages

• **Large Quantities of Contacts**

– The PLC has a large number of contacts for each coil available in its programming.

– Suppose that a panel-wired relay has four contacts and all are in use when a design change requiring three more contacts is made.

- Time would have to be taken to procure and install a new relay or relay contact block.
- Using a PLC, however, only three more contacts would be typed in. Contacts are now a “software” component

Communications Capability.

A PLC can communicate with other controllers or computer equipment to perform such functions as supervisory control, data gathering, monitoring devices and process parameters, and download and upload of programs.

PLC Advantages

- **Lower Cost**

– Increased technology makes it possible to condense more functions into smaller and less expensive packages.

– Now you can purchase a PLC with numerous relays, timers, and counters, a sequencer, and other functions for a few hundred dollars.

- **Pilot Running**

– A PLC programmed circuit can be evaluated in the lab. The program can be typed in, tested, observed, and modified if needed, saving valuable factory time.

PLC Advantages

• **Ladder or Boolean Programming Method**

– The PLC programming can be accomplished in the ladder mode by an electrician or technician. Alternatively, a PLC programmer who works in digital or Boolean control systems can also easily perform PLC programming.

• **Reliability and Maintainability**

– Solid-state devices are more reliable, in general, than Mechanical systems or relays and timers. Consequently, the control system maintenance costs are low and downtime is minimal.

PLC Advantages

- **Documentation**

– An immediate printout of the true PLC circuit is available in minutes, if required.

– There is no need to look for the blueprint of the circuit in remote files.

– The PLC prints out the actual circuit in operation at a given moment.

– Often, the file prints for relay panels are not properly kept up to date. A PLC printout is the circuit at the present time; no wire tracing is needed for verification.

PLC Disadvantages

- **Fixed Program Applications**

– Some applications are single-function applications. It does not pay to use a PLC that includes multiple programming capabilities if they are not needed.

– Their operational sequence is seldom or never changed, so the reprogramming available with the PLC would not be necessary.

references:

1-notes from Dr.* Jeff Jackson *,the university of Alabama

2-notes from Dr. Radu Muresan ,University of Guelph

3-notes from Dr. Mohammad Salah ,Hashemite University

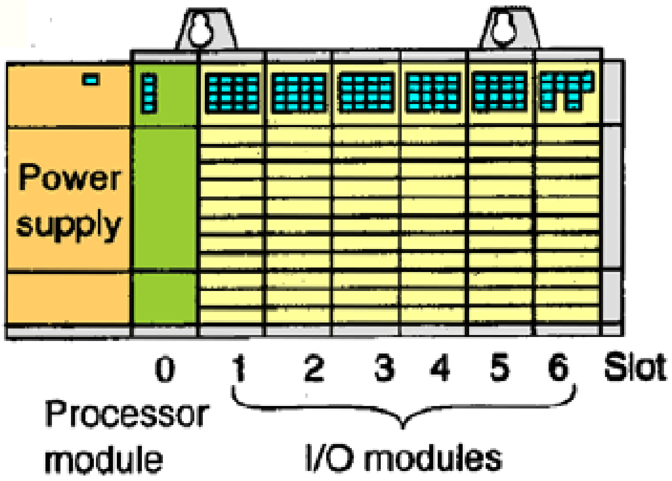
PLC Hardware components

•

I/O SECTION

Input and output (I/O) modules enable the PLC to both sense and control a process.

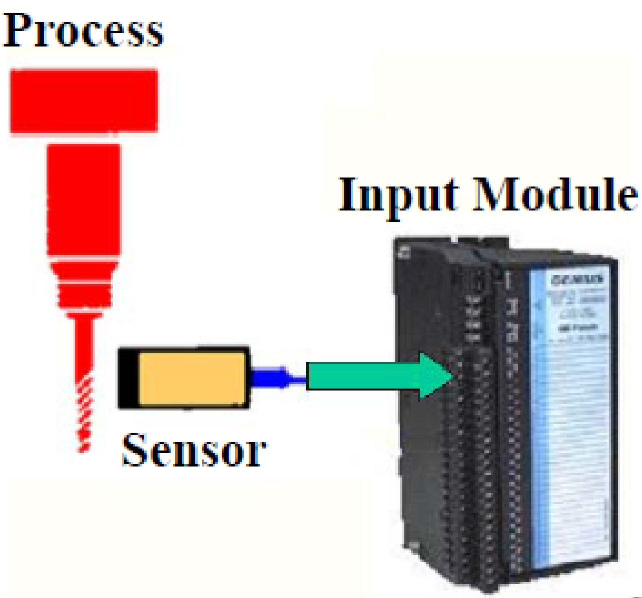
The I/O section consists of an I/O rack and individual I/O modules.



PLC Hardware components

Input Interface

Input interface modules **accept signals from the machine or process devices and convert them into signals that can be used by the controller.**



The input interface allows status information regarding processes to be communicated to the CPU.

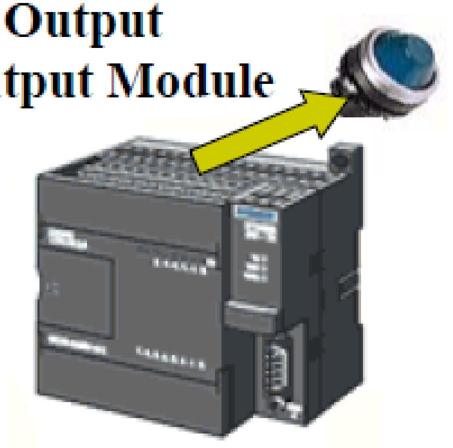
PLC Hardware components

Output Interface

Output interface modules **convert controller signals into external signals that can be used to control the machine or process.**

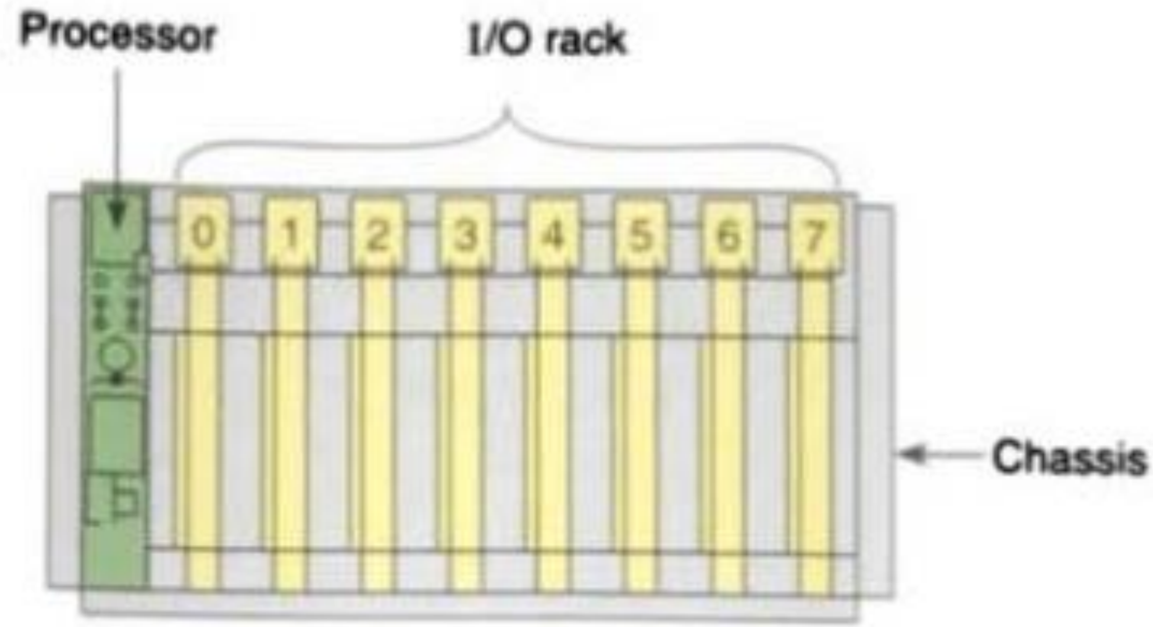
Pilot Light

External Output From Output Module



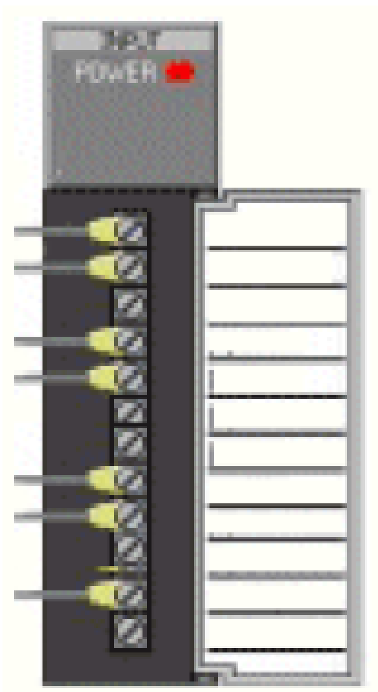
The output interface allows the CPU to communicate operating signals to the process devices under its control.

PLC Hardware components

- 

A *chassis* is a physical hardware assembly that houses devices such as I/O modules, processor modules, and power supplies. Chassis come in different sizes according to the number of slots they contain. In general, they can have 4, 8, 12, or 16 slots.

I/O Module Addressing



The **location** of a module within a rack and the **terminal number** of the module to which an input or output device is connected will determine the device's address.

Each input and output device must have a **specified address**.

This address is used by the processor to identify where the device is located in order to monitor or control it.

I/O Module Addressing

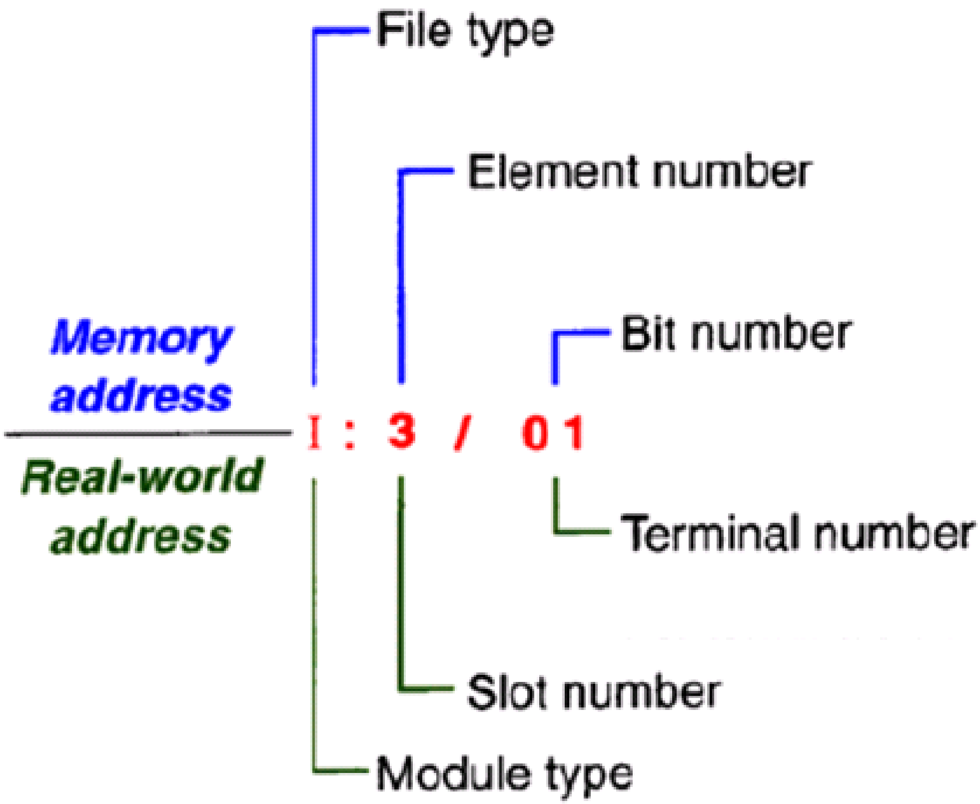
In general, the basic addressing elements include:

Type – the type determines if an input or output is being addressed.

Slot – the slot number is the physical location of the I/O module.

Bit used to identify the actual terminal connection in a particular I/O module.

SLC 500 Addressing

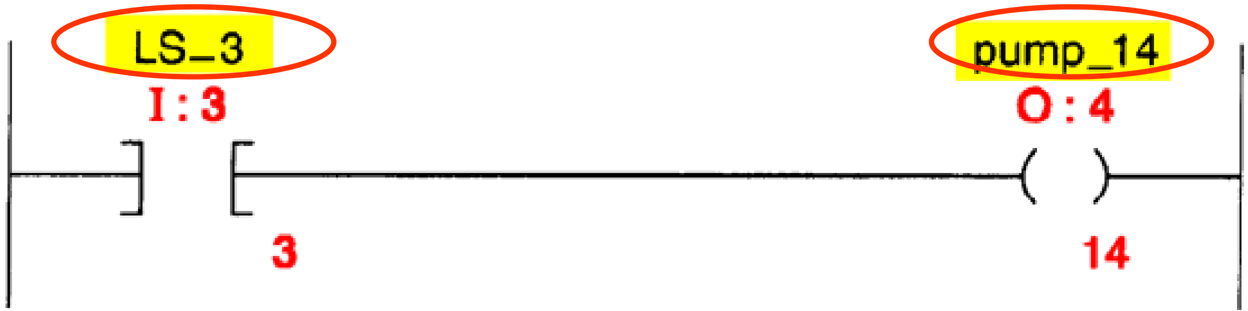


Examples:

O :4.0/15–Output module in slot 4, terminal 15

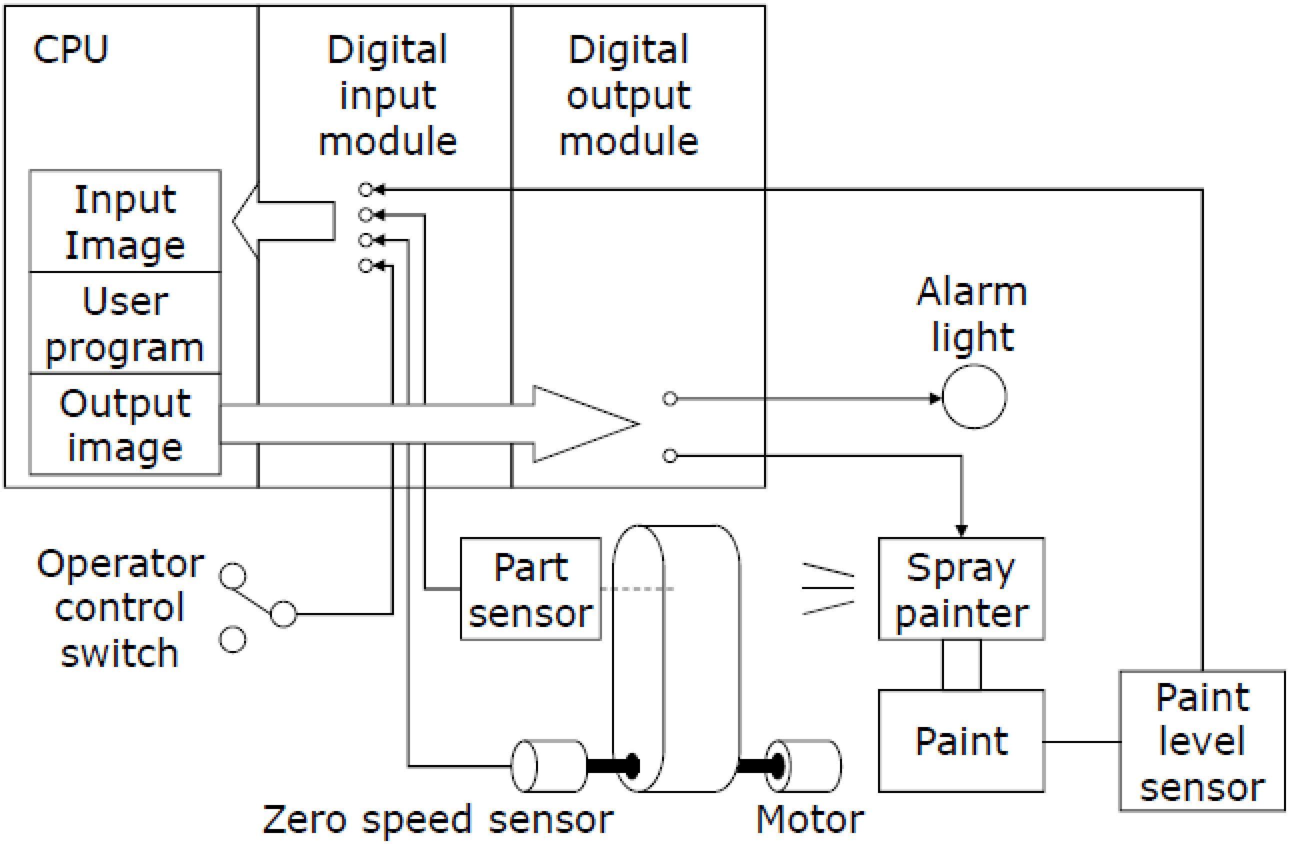
I :3.0/8–Input module in slot 3, terminal 8

Symbolic Addressing

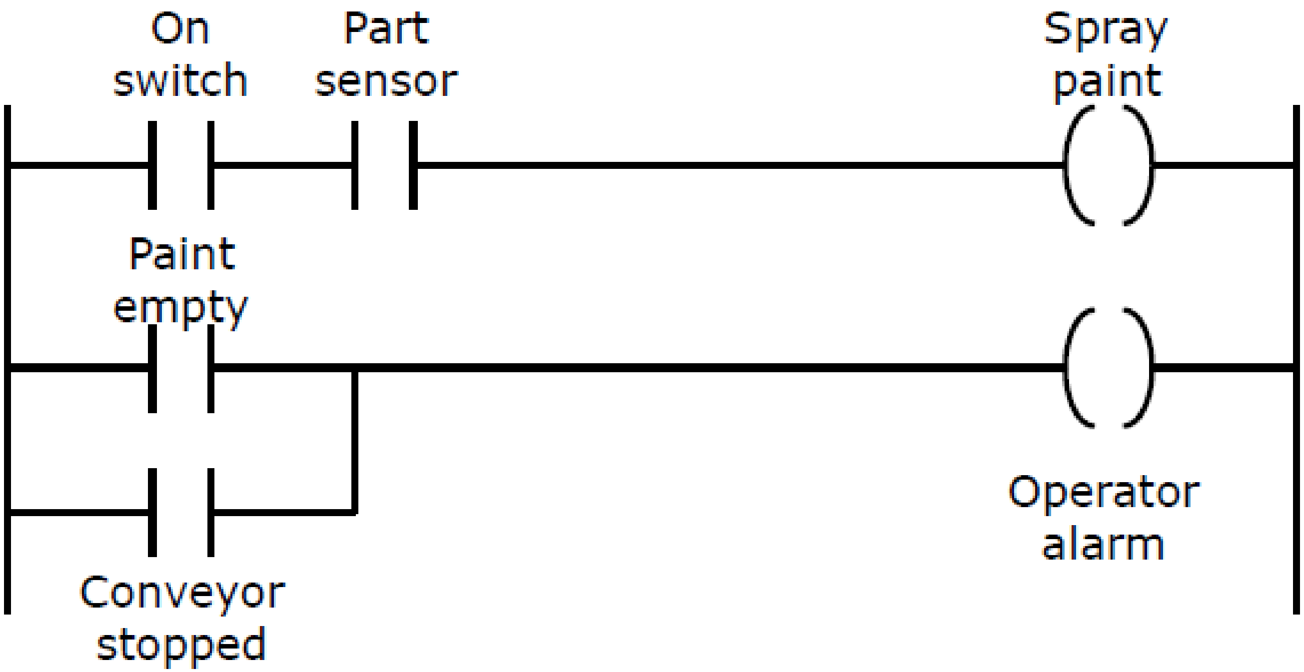


Symbolic addresses are real names or codes the programmer can substitute for a logical address because they relate physically to the application. They are a physical name convention for a location in the data table. In this example, the symbolic addresses are LS_3 and pump_14, while the actual addresses are I:3/3 and O:4/14, respectively.

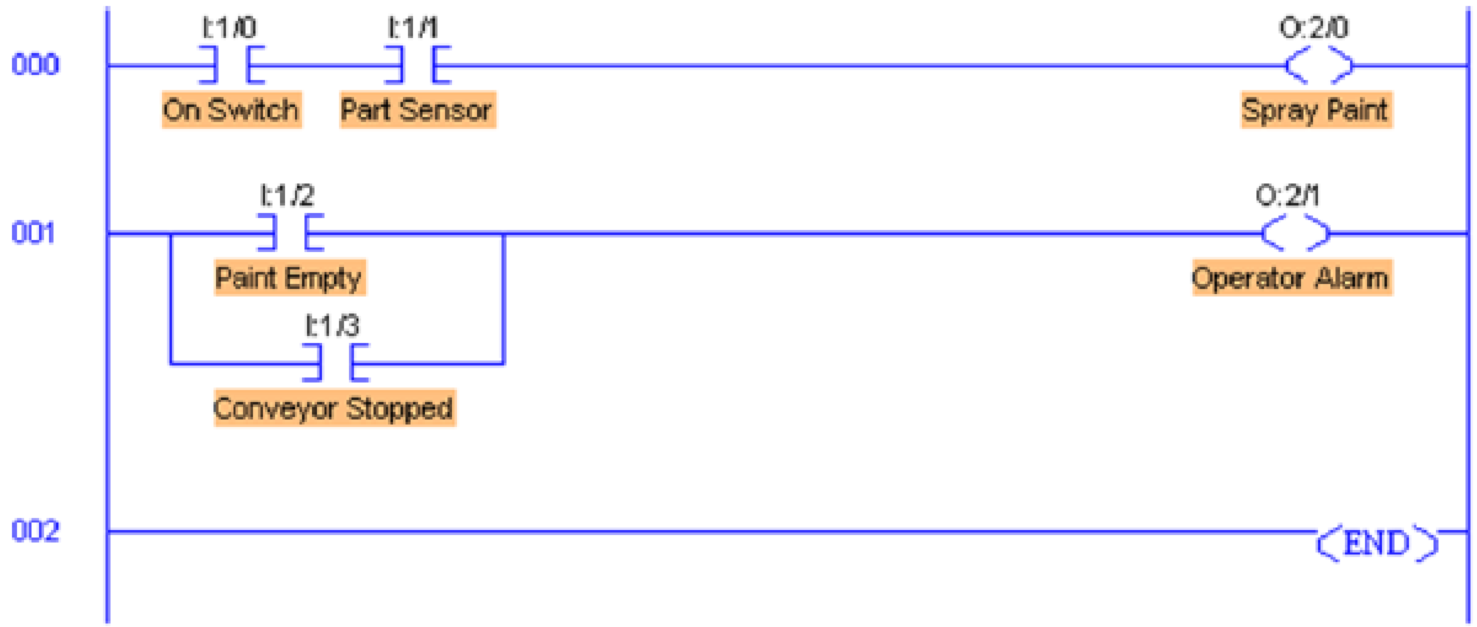
Two-rung ladder logic program



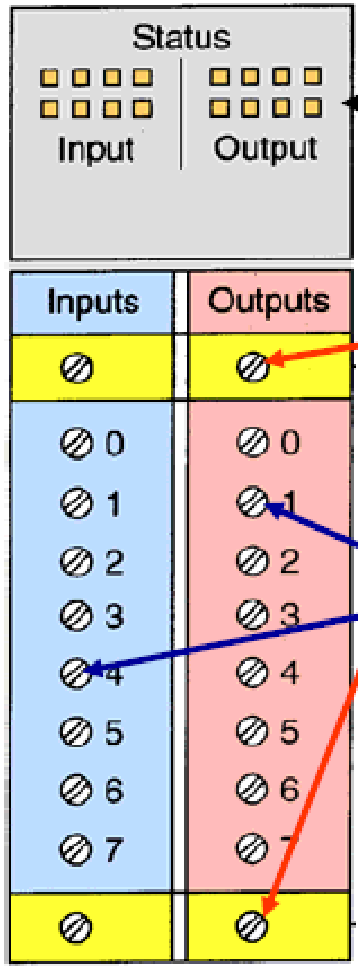
Two-rung ladder logic program



Example Program (In LogixPro Simulator)



Typical Combination I/O Module

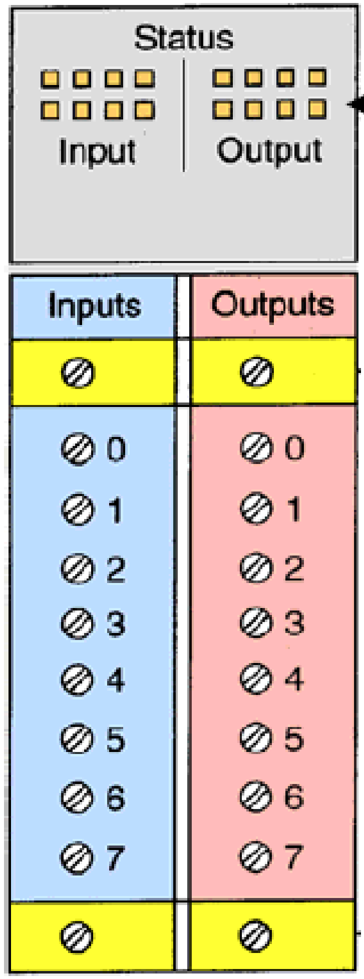


Status lights for each of the inputs and outputs

Connections to the power supply used to power the inputs and outputs.

Terminals for each input and output connection

Typical Combination I/O Module



Status lights for each of the inputs and outputs

Most modules have plug-on wiring terminal strips. The terminal strip is plugged into the actual module. If there is a problem with a module, the entire strip is removed, a new module is inserted, and the terminal strip is plugged into the new module.

Discrete I/O Modules

Discrete type I/O interface modules connects field devices of the ON/OFF nature.

Discrete input modules are used with field control devices such as selector switches, pushbuttons, and limit switches.

Discrete I/O Modules

Discrete type I/O interface modules connects field devices of the ON/OFF nature.

Discrete output modules are used with field load devices such as lights, small motors, solenoids, and motor starters that require simple ON/OFF switching.

Discrete I/O Modules

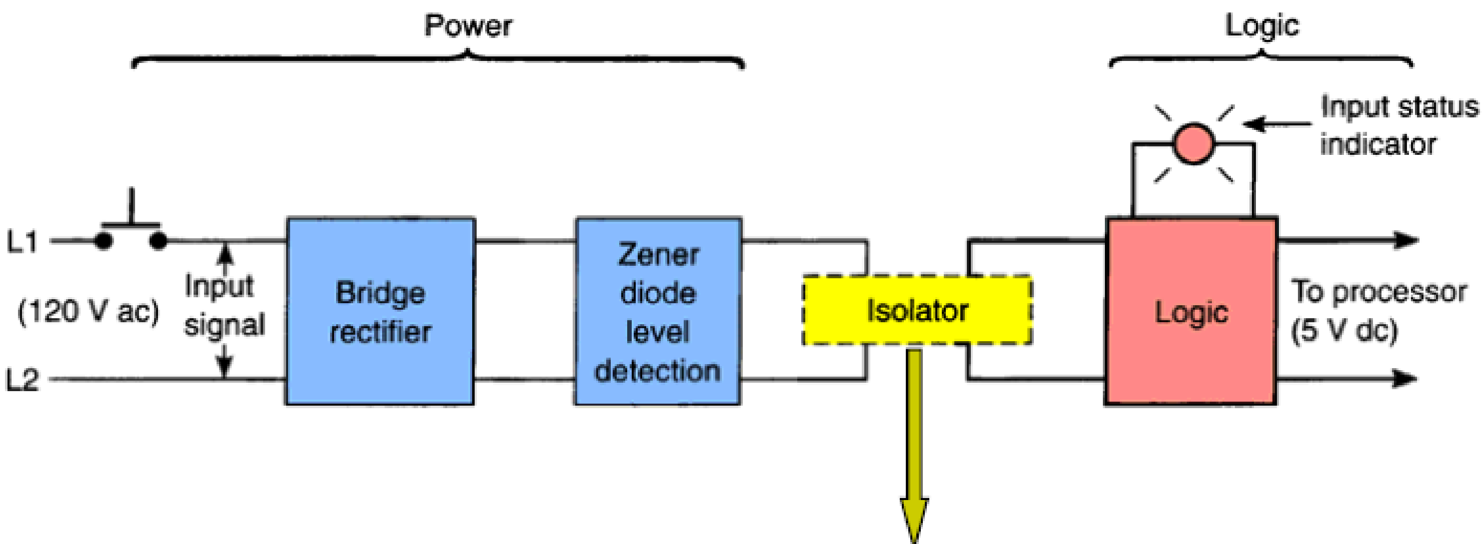
Each discrete I/O module is powered by some field-supplied voltage source of a specified value. Common voltage ratings are:

Input Interfaces	Output Interfaces
12 V ac/dc	12-48 V ac
24 V ac/dc	120 V ac
48 V ac/dc	230 V ac
120 V ac/dc	120 V ac
230 V ac/dc	230 V dc

AC Discrete Input Module

Radu Muresan

The input circuit is composed of two basic sections: the power section and the logic section.

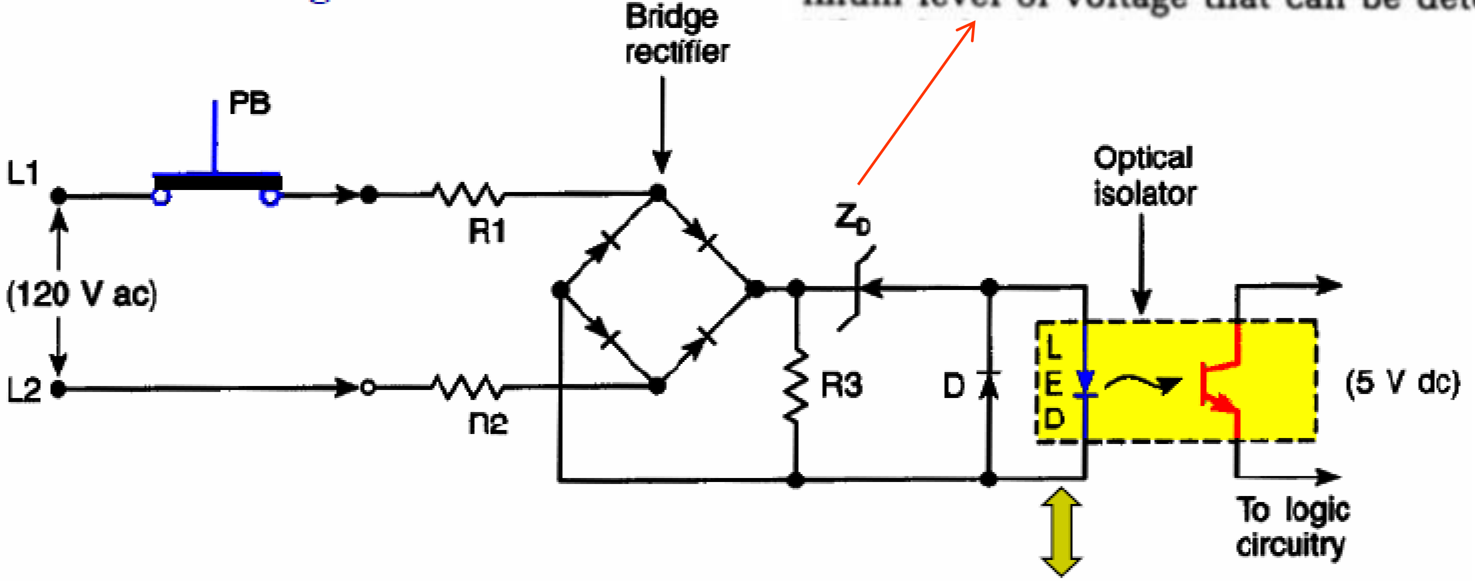


The power and logic sections are normally coupled together with a circuit, which electrically separates the two.

AC Discrete Input Module Operation

When the pushbutton is closed, 120 VAC is applied to the bridge rectifier through R1 and R2.

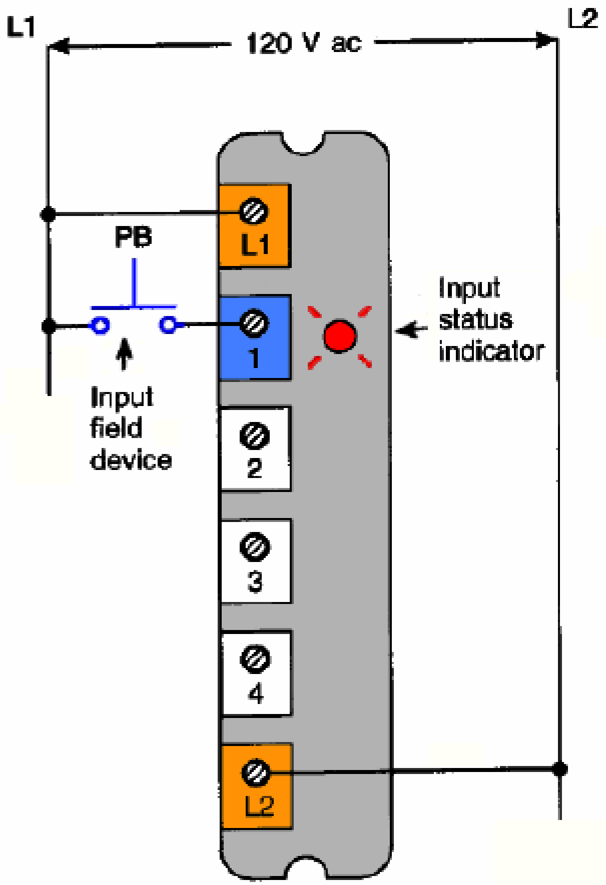
zener diode (Z_D) voltage rating sets the minimum level of voltage that can be detected.



This produces low level DC voltage across the LED.
The resulting light switches the phototransistor into conduction and the closed status of the pushbutton is communicated to the processor.

Input Module Tasks

Radu Muresan



Sense when a signal is received from a sensor

Convert the input signal to the correct voltage level for the particular PLC

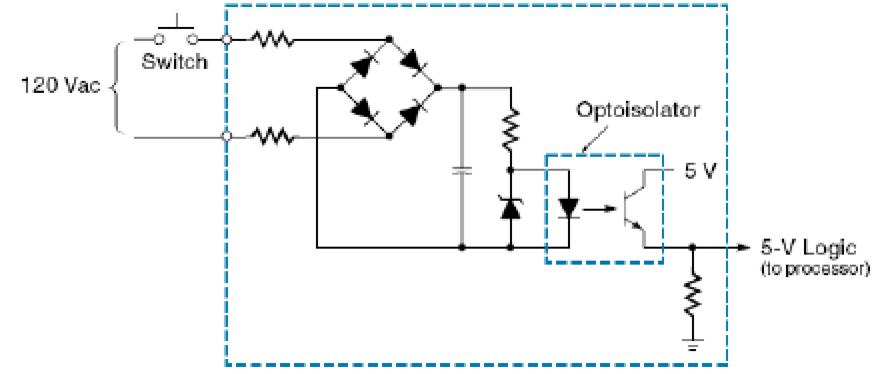
Isolate the PLC from fluctuations in the input signal's voltage or current

Send a signal to the processor indicating which sensor originated the signal

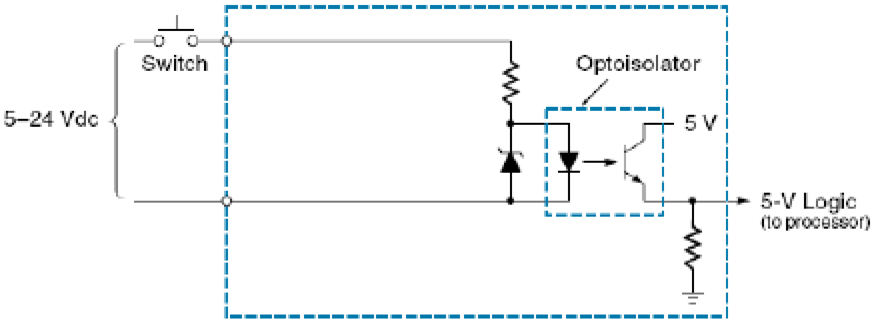
ENGG3490: Mechatronics ; W07. Addapted From Programmable Logic Controllers by F. D. Petruzella, McGraw-Hill

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



(a) AC input module circuit

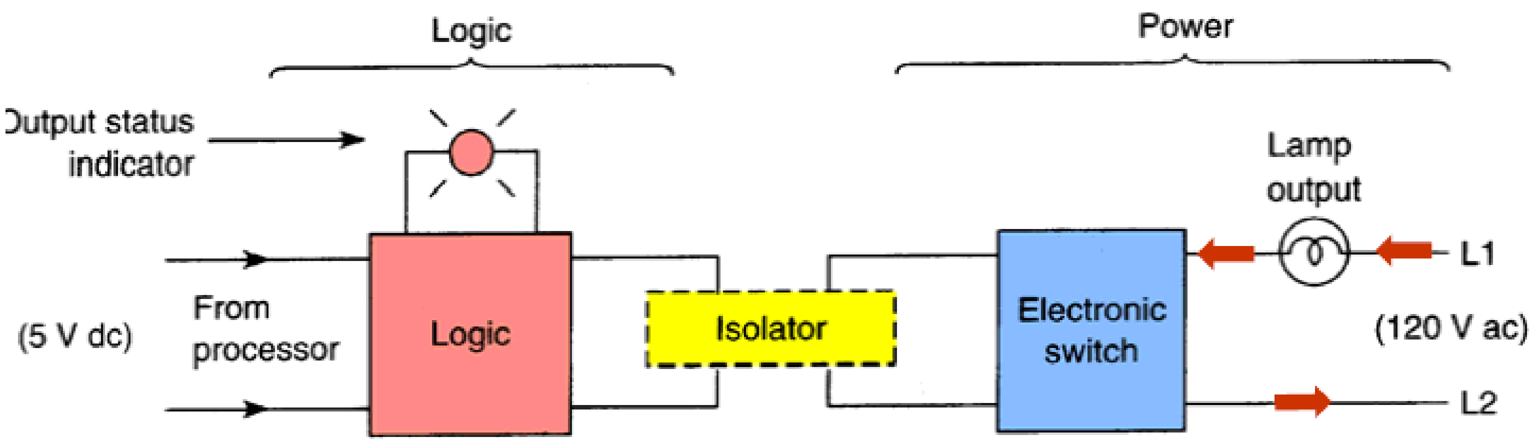


(b) DC input module circuit

AC Discrete Output Module

Radu Muresan

The output circuit is composed of two basic sections: the power section and the logic section, coupled by an isolation circuit.

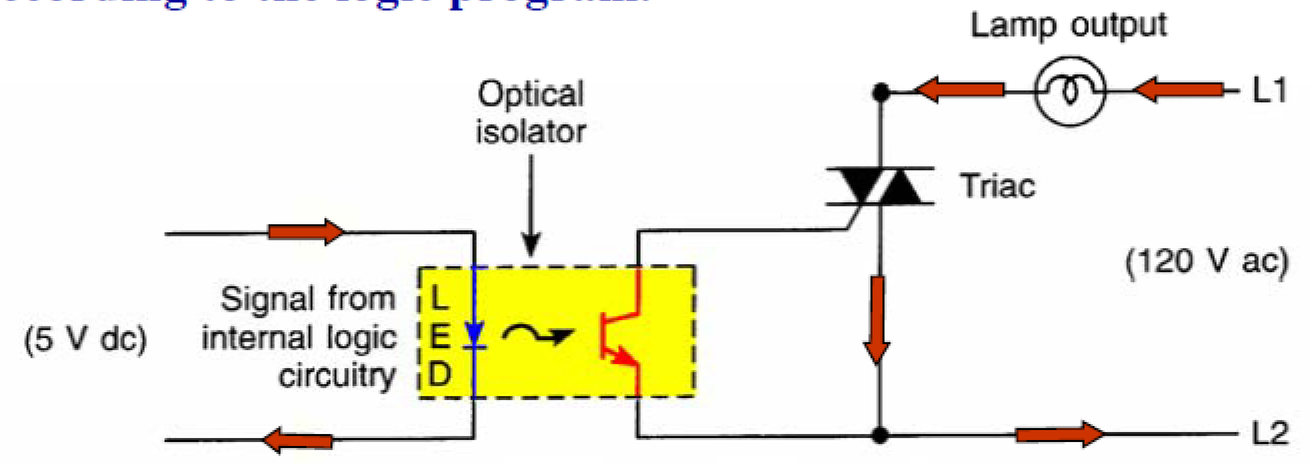


The output interface can be though of as a simple electronic switch to which power is applied to control the output device.

AC Discrete Output Module Operation

Radu Muresan

The processor sets the output status according to the logic program.

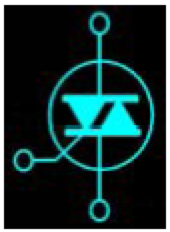


When the processor calls for an output, a voltage is applied across the LED of the isolator, which switches the phototransistor into conduction.

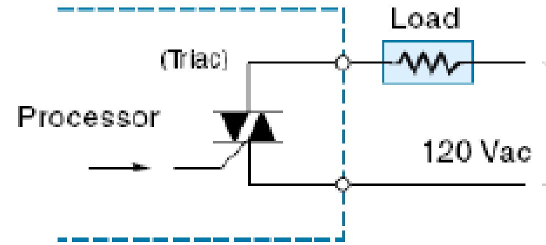
This in turn switches the Triac into conduction which, in turn, turns on the lamp.

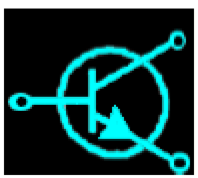
Types Of Discrete Output Modules

Are used to turn two-state devices either ON or OFF

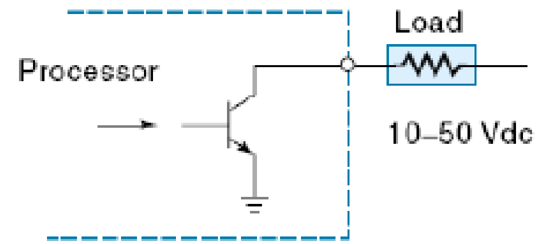


Triac outputs can be used only for control of AC devices.

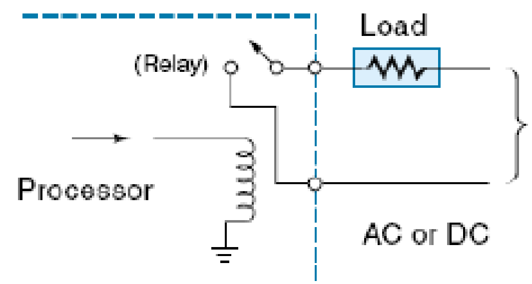




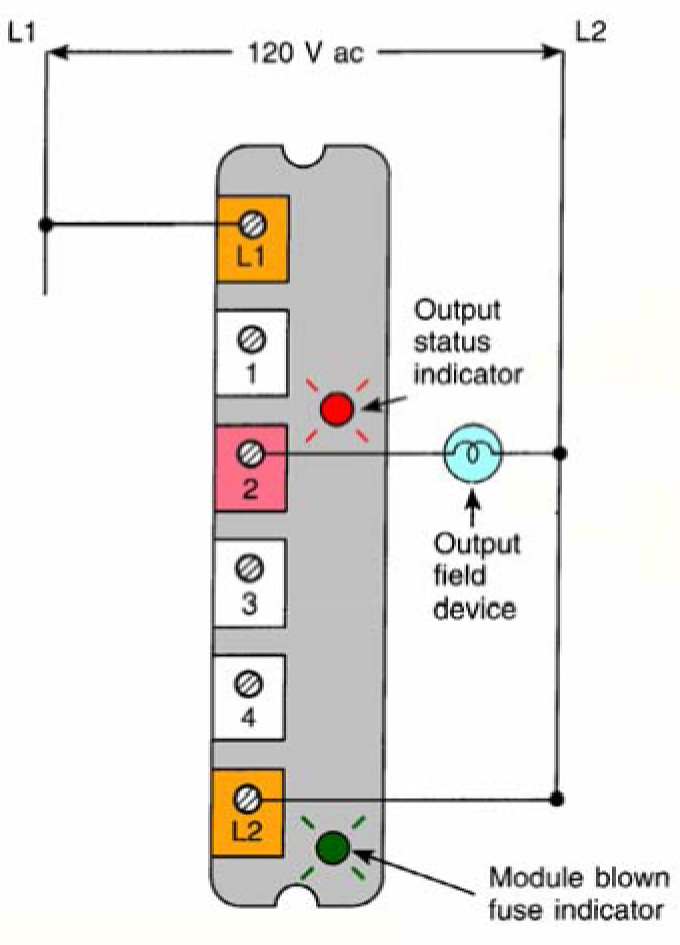
Transistor outputs can be used only for control of DC devices.



Relay outputs can be used with AC or DC devices. However they have a much slower switching time compared to solid-state outputs.



AC Discrete Output Module



Provided with LEDs that indicate the status of each output.

Fuses are generally required for each circuit. Some modules also provide visual indicators for fuse status.

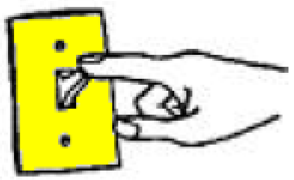
Individual AC outputs are usually limited to 1 or 2 amps. The maximum current load for any one module is also specified.

ımable Logic Controllers by F. D. Petruzella, McGraw-Hill 32

Discrete Versus Analog Devices

Radu Muresan

Discrete devices are inputs and outputs that have only two states: on and off.



ON/OFF
toggle switch

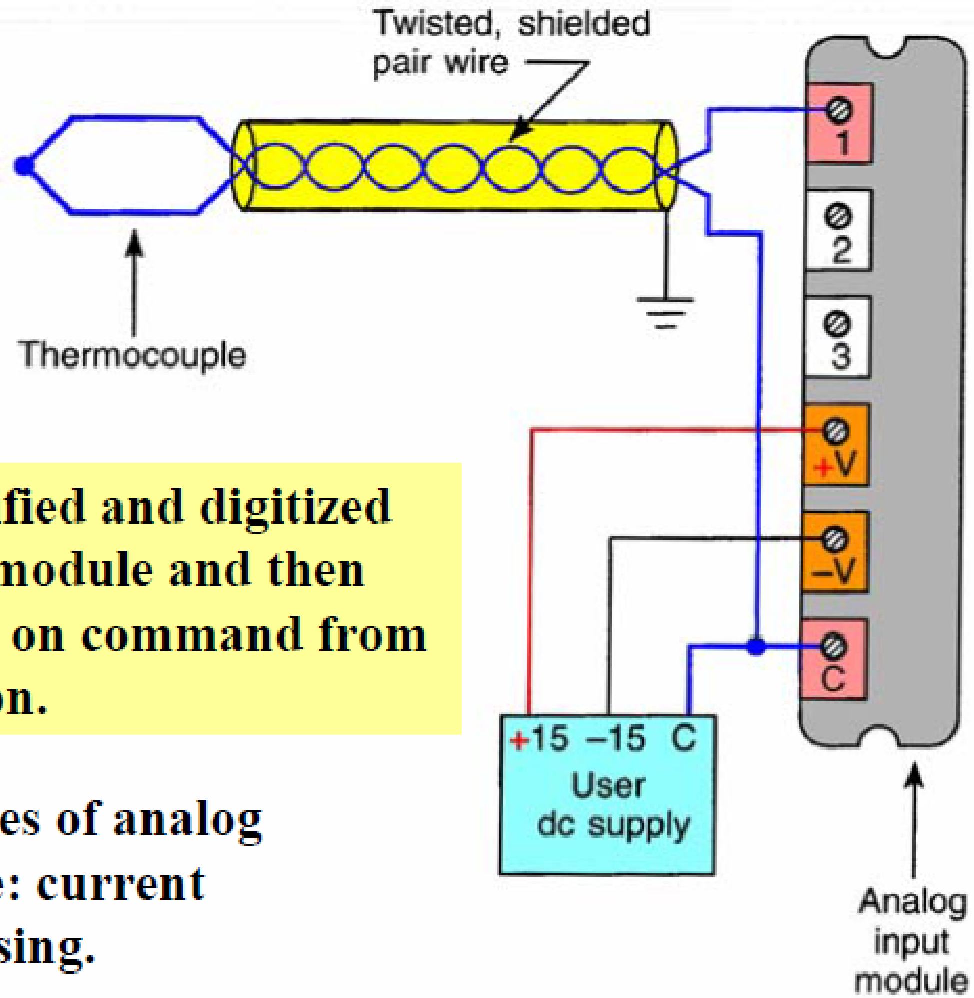
Analog devices are inputs and outputs that can have an infinite number or states.

Analog
control valve

Analog Input Interface Module

Radu Muresan

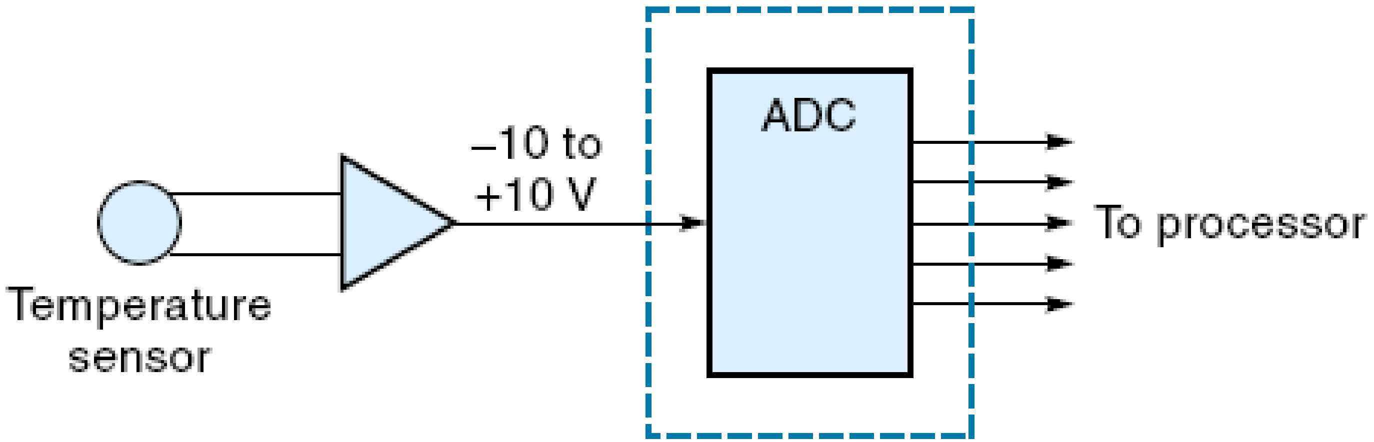
A varying low DC voltage proportional to the temperature being monitored is produced by the thermocouple.



This voltage is amplified and digitized by the analog input module and then sent to the processor on command from a program instruction.

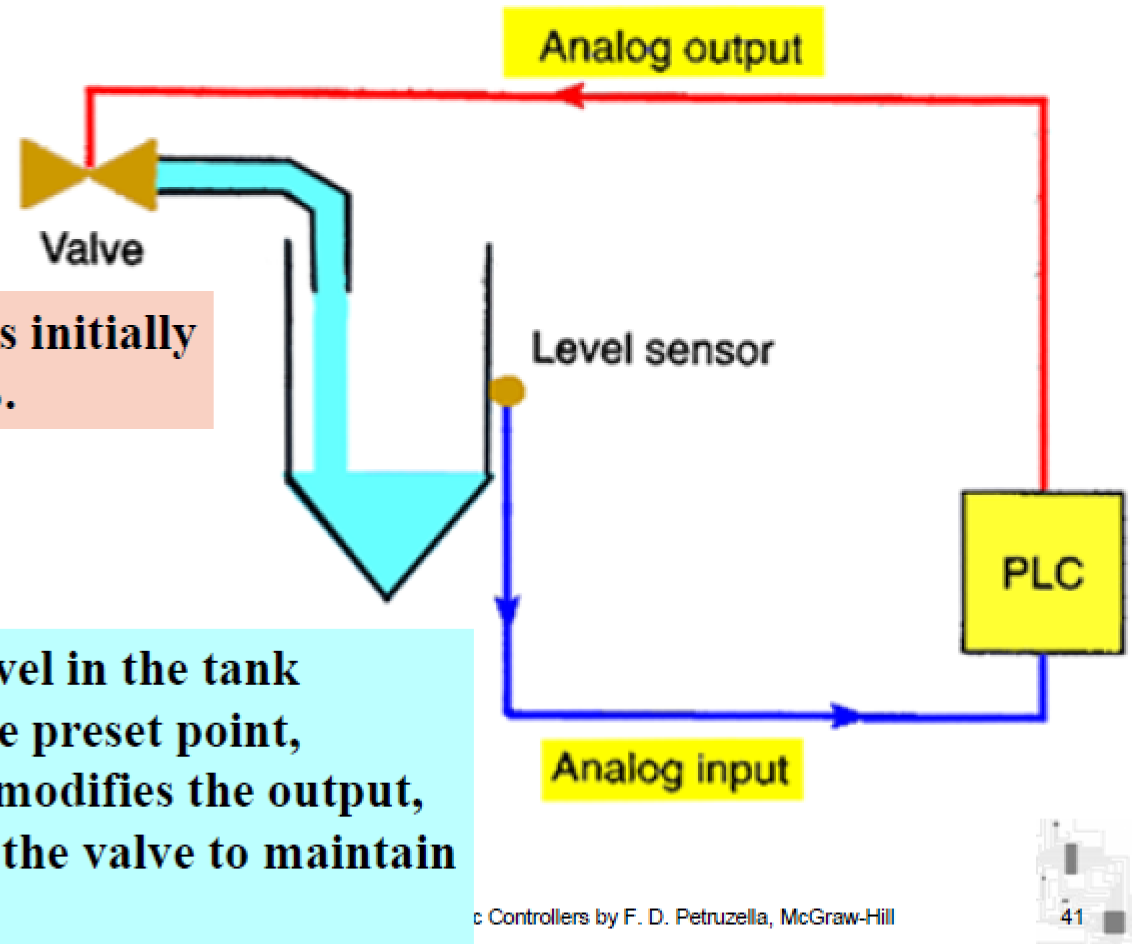
There are two basic types of analog input modules available: current sensing and voltage sensing.

Analog input interface modules contain the circuitry necessary to accept analog voltage or current signals from analog field devices. These inputs are converted from an analog to a digital value by an *analog-to-digital (A/D)* converter circuit. The conversion value, which is proportional to the analog signal, is



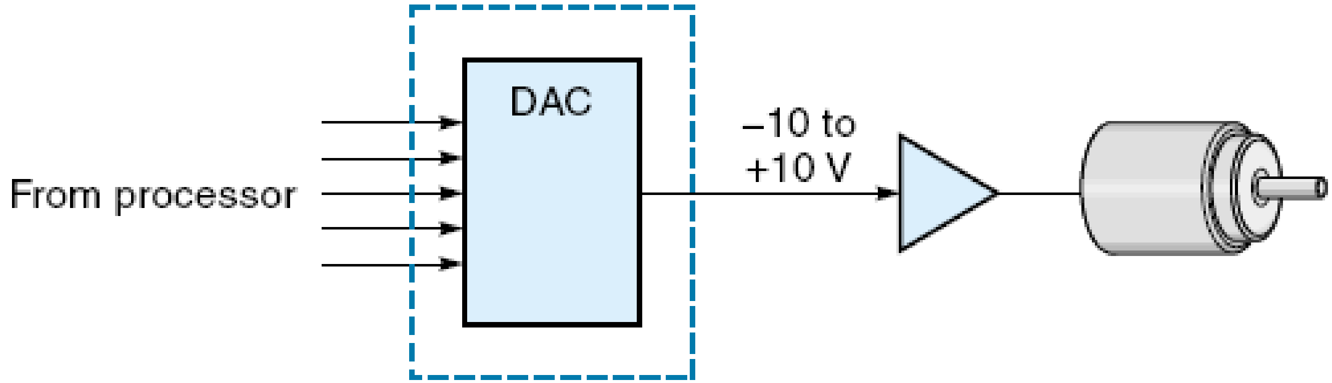
(a) Analog input module

Analog I/O Control System



The valve is initially open 100%.

As the fluid level in the tank approaches the preset point, the processor modifies the output, which adjusts the valve to maintain a set point.



(b) Analog output module

The *analog output interface module* receives from the processor digital data, which are converted into a proportional voltage or current to control an analog field device. The digital data is passed through a *digital-to-analog (D/A)* converter circuit to produce the necessary analog form.

Special I/O Modules

Radu Muresan

High-Speed Counter Module

Used to provide an interface for applications requiring counter speeds that surpass the capacity of the PLC ladder program.

They have the electronics needed to operate independently of the processor.

A typical count rate is 0 to 75 kHz, which means the module would be able to count 75,000 pulses per second.

Special I/O Modules

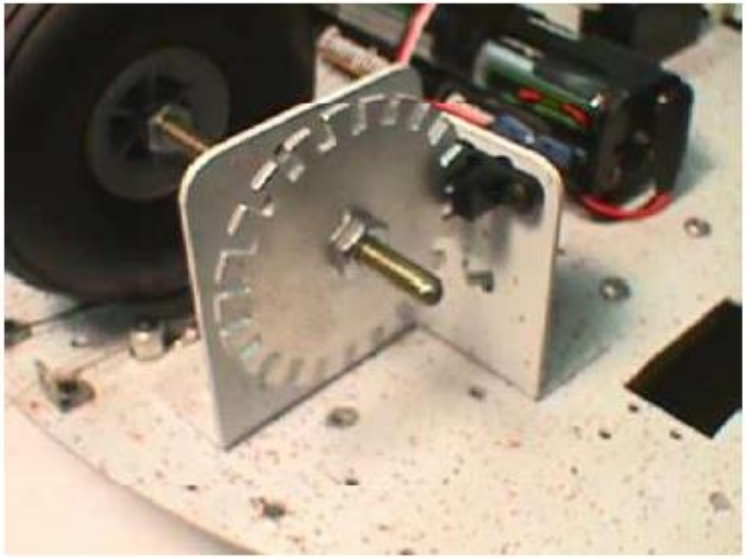
Radu Muresan

Thumb-Wheel Module

Allows the use of thumb-wheel switches for feeding information to the PLC to be used in the control program.

Special I/O Modules

Encoder-Counter Module



This module allows the user to read the signal from the encoder on a real-time basis and stores this information so it can be read later by the processor.

Special I/O Modules

Radu Muresan

Stepper-Motor Module

This module provides pulse trains to a stepper-motor translator, which enables control of a stepper motor.

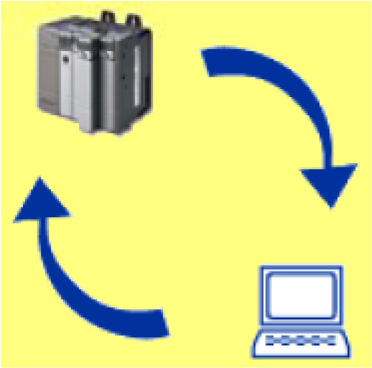
Intelligent I/O Modules

Radu Muresan

Have their own microprocessor on board that can function in parallel with the PLC.

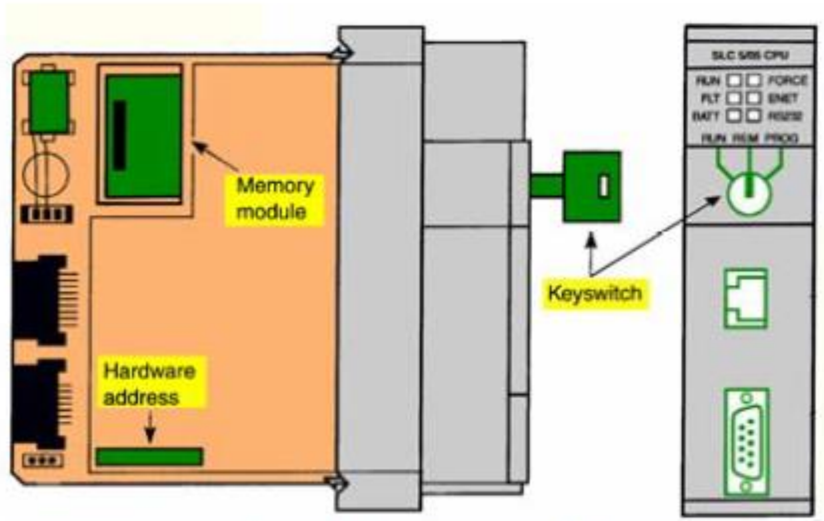
PID module is used in process control applications that incorporate PID algorithms. The PID module allows process control to take place outside the CPU.

Communications Module



- **Communication Module**
  As different systems are integrated, data must be shared throughout all the systems. PLCs must be able to communicate with computers, computer numerical control (CNC) machines, robots, and other PLCs. This module allows the user to connect the PLC to high-speed local networks that may be different from the network communication provided with the PLC.

Processor Unit



The processor executes the operating system, manages memory, monitors inputs, evaluates the user logic, and turns on the appropriate outputs.

Status indicators provide system diagnostic information.
Keyswitch allows you select different modes of operation.

Typical Processor Modes Of Operation

RUN Position

- Places the processor in the Run mode
- Executes the ladder program and energizes output devices
- Prevents you from performing online program editing in this position
- Prevents you from using a program/operator interface device to change the processor mode

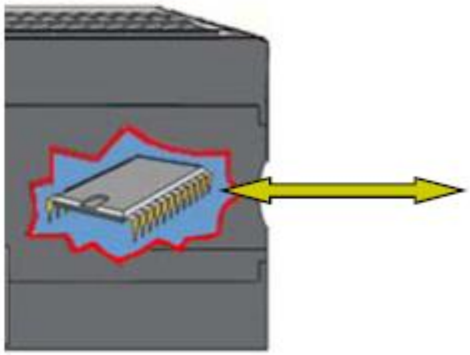
Typical Processor Modes Of Operation

Radu Muresan

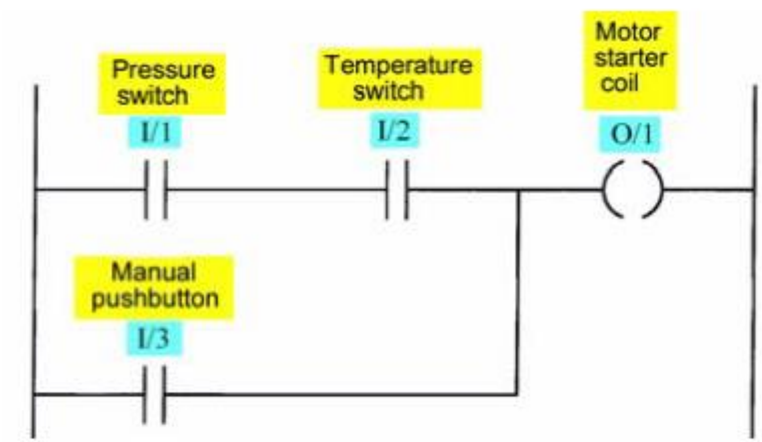
PROG Position

- Places the processor in the program mode
- Prevents the processor from scanning or executing the ladder program, and controller outputs are de-energized
- Allows you to perform program entry and editing
- Prevents you from using a program/operator interface device to change the processor mode

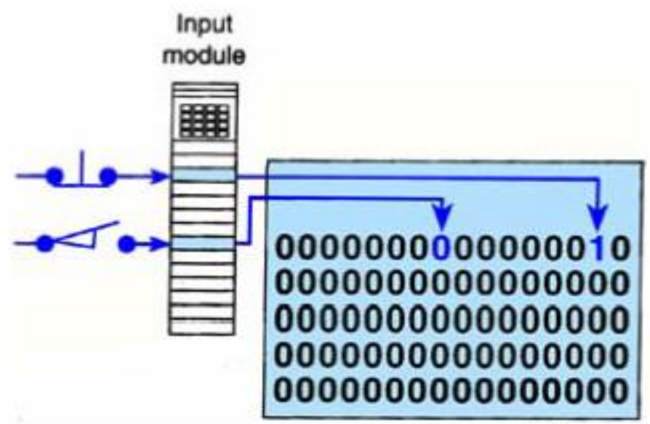
Memory Design



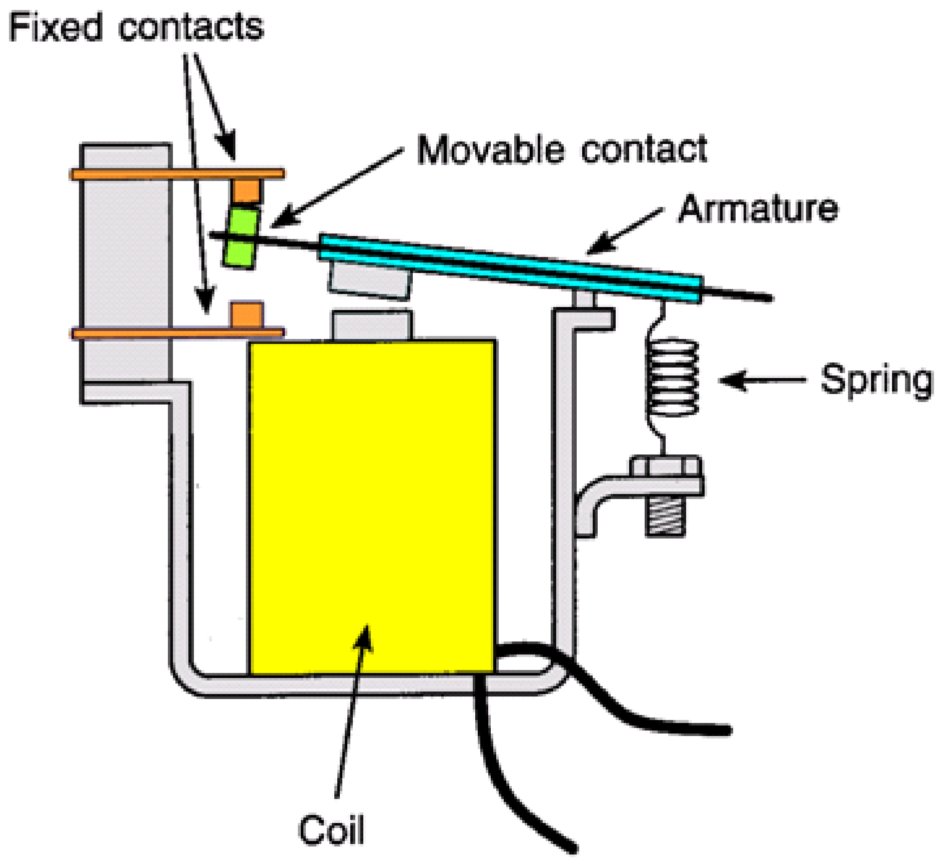
Memory is a physical space inside the CPU where the *program files* and *data files* are stored and manipulated.



Program File



Data File


Fixed contacts
Movable contact
Armature
Spring
Coil

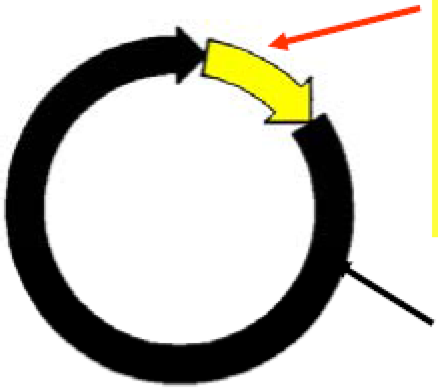
references:

1-notes from Dr.* Jeff Jackson *,the university of Alabama

2-notes from Dr. Radu Muresan ,University of Guelph

5.2 Program scan:

During each operating cycle, the processor reads all inputs, takes these values, and energizes or de-energizes the outputs according to the user program. This process is known as a *scan*.



I/O scan – records status data of input devices. Energizes output devices that have their associated status bits set to ON (1)

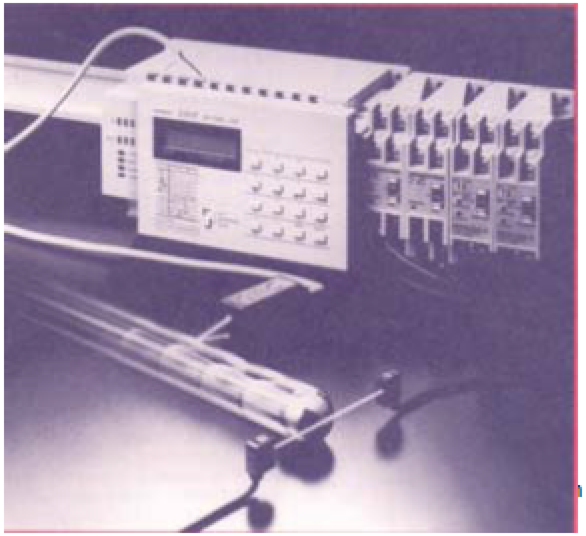
Program scan – instructions are executed sequentially

Because the inputs can change at any time, the PLC must carry on this process continuously.

Scan Process

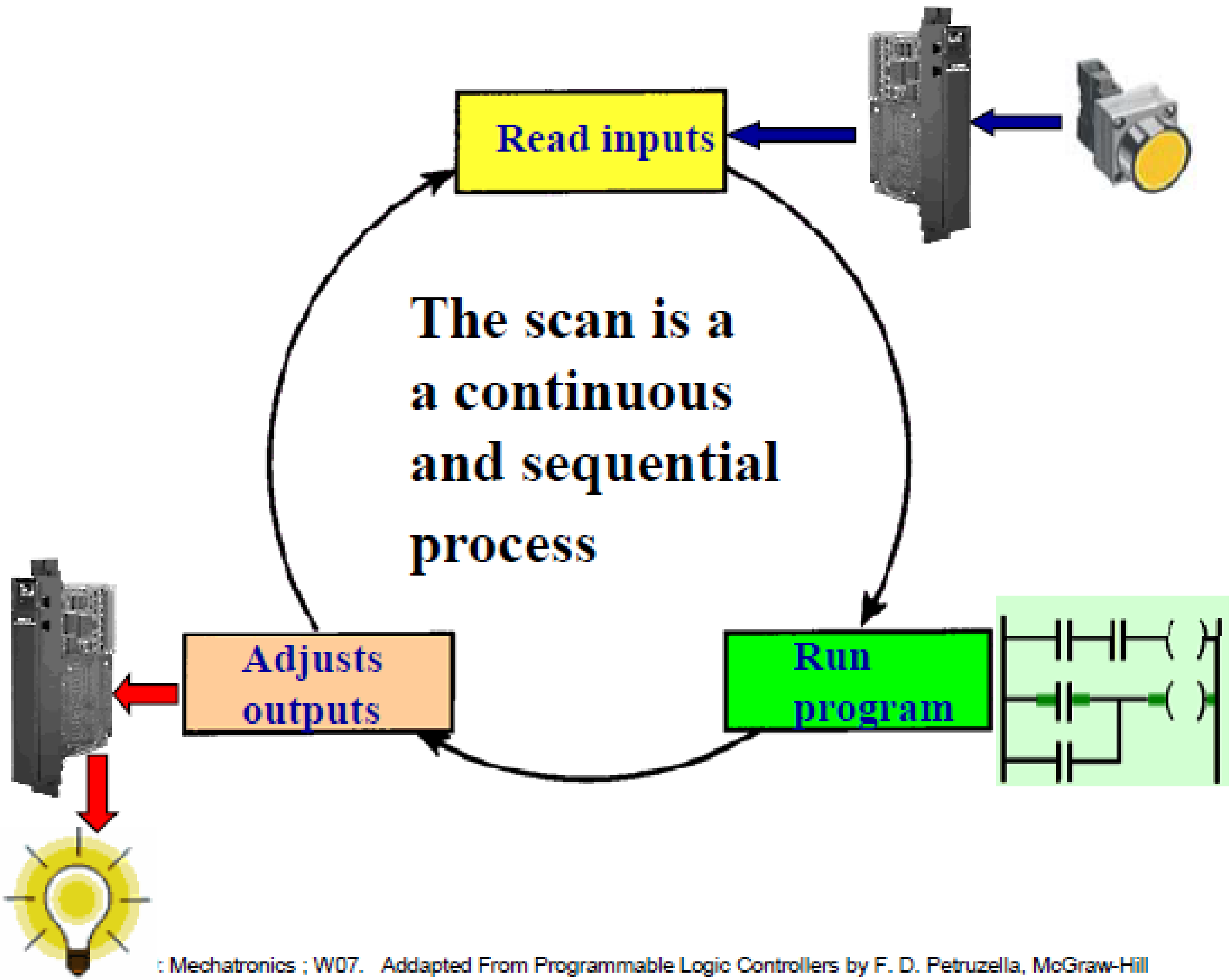
Radu Muresan

The scan *time* indicates how fast the controller can react to changes in inputs. Scan times vary with computer model and program content, and length. If a controller has to react to an input signal that changes states twice during the scan time, it is is possible that the PLC will never be able to detect this change.



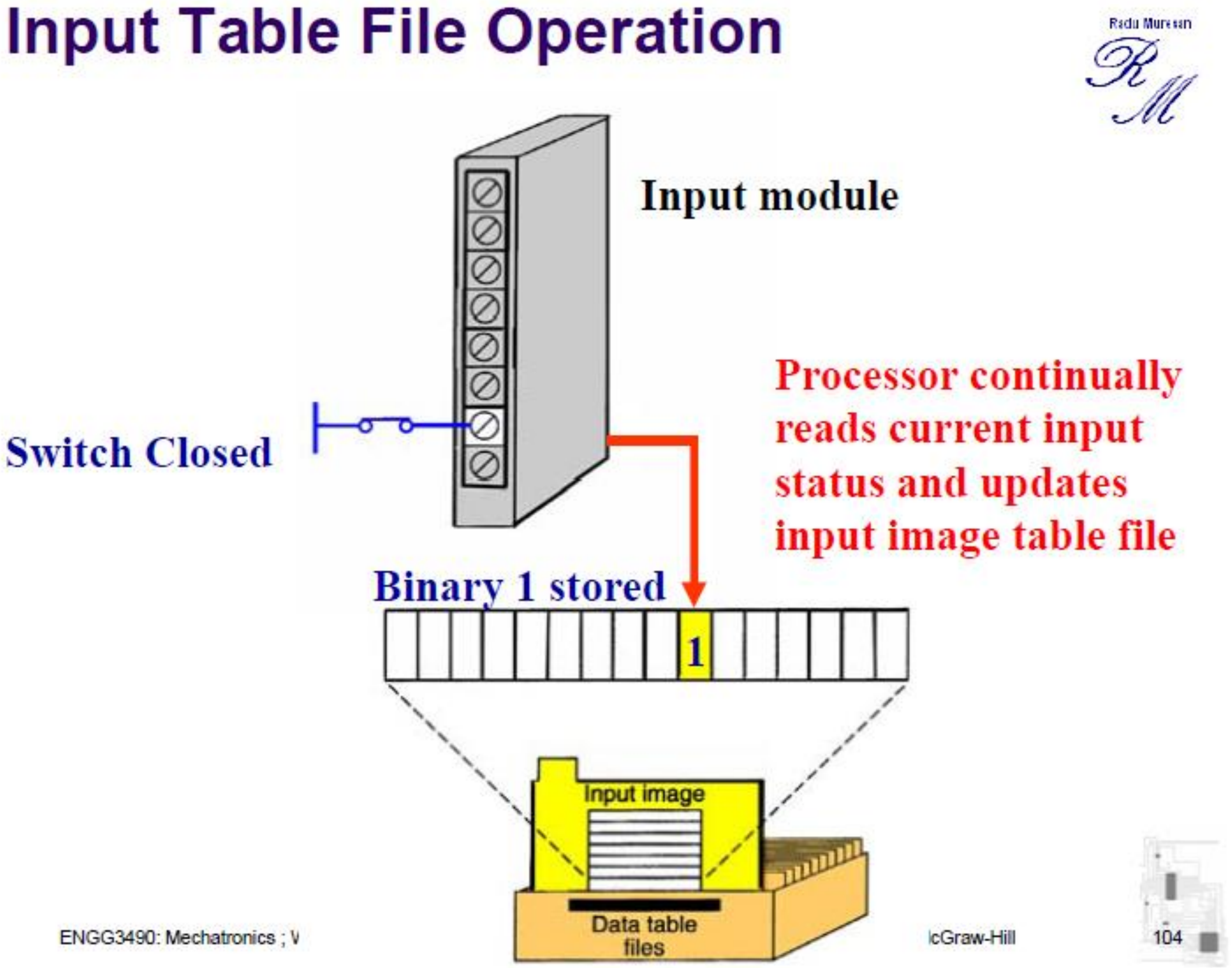
Scan time may be a concern in high speed operations

Scan Process

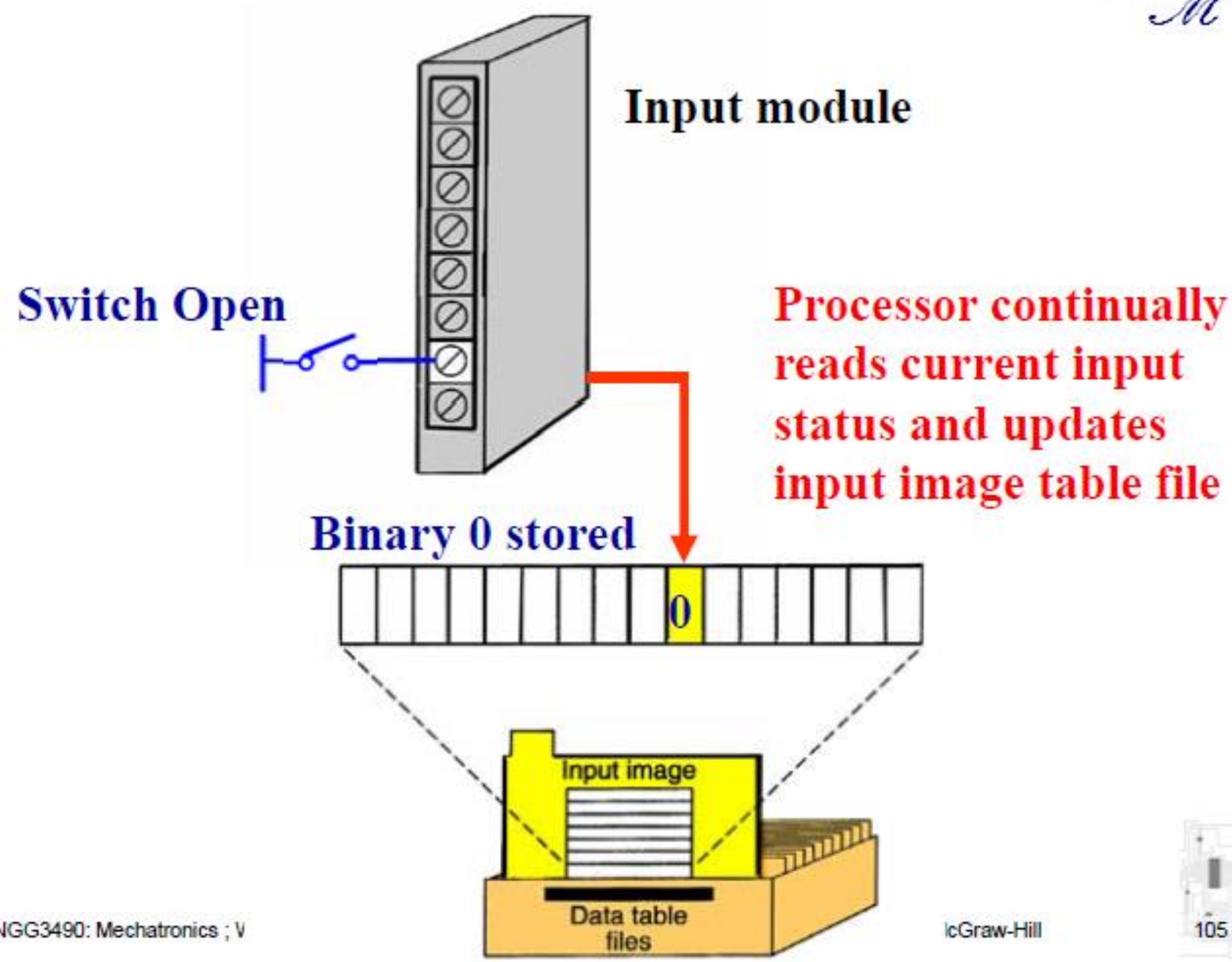


: Mechatronics ; W07. Addapted From Programmable Logic Controllers by F. D. Petruzella, McGraw-Hill

Input Table File Operation



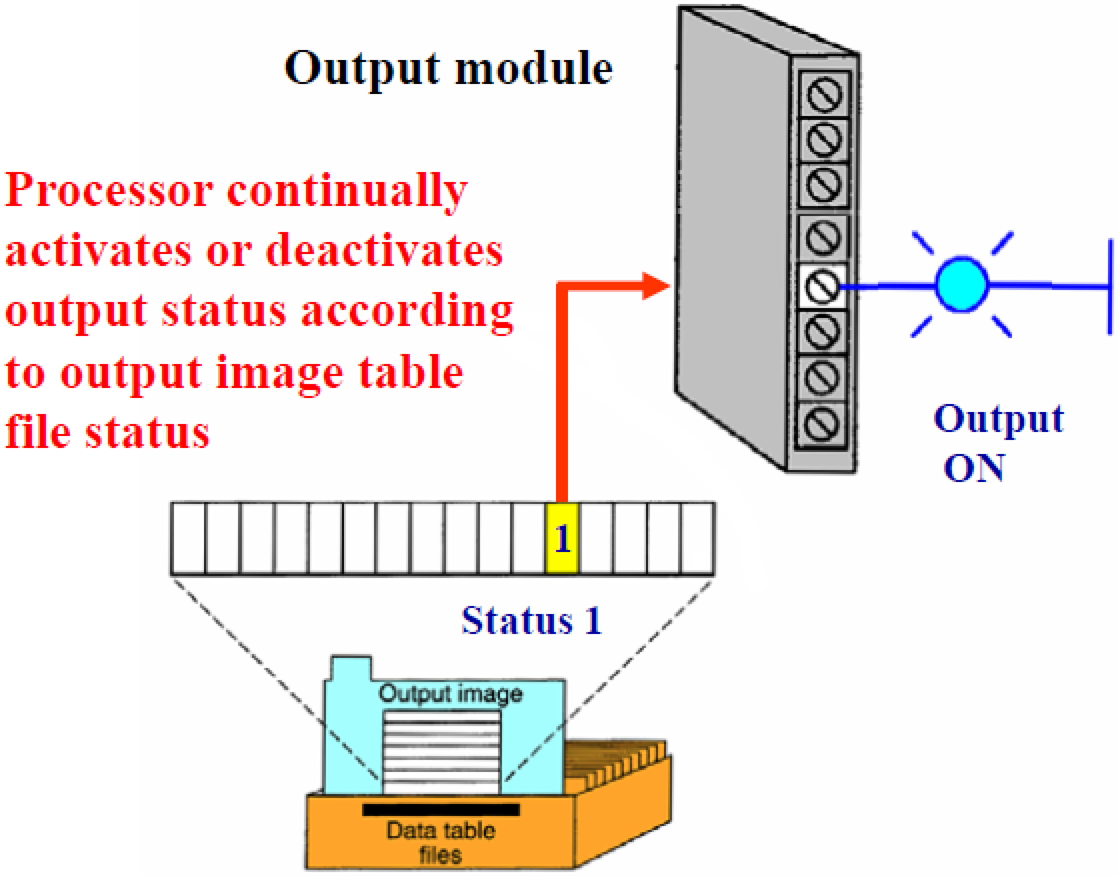
Input Table File Operation



ENGG3490: Mechatronics ; V ... lcGraw-Hill 105

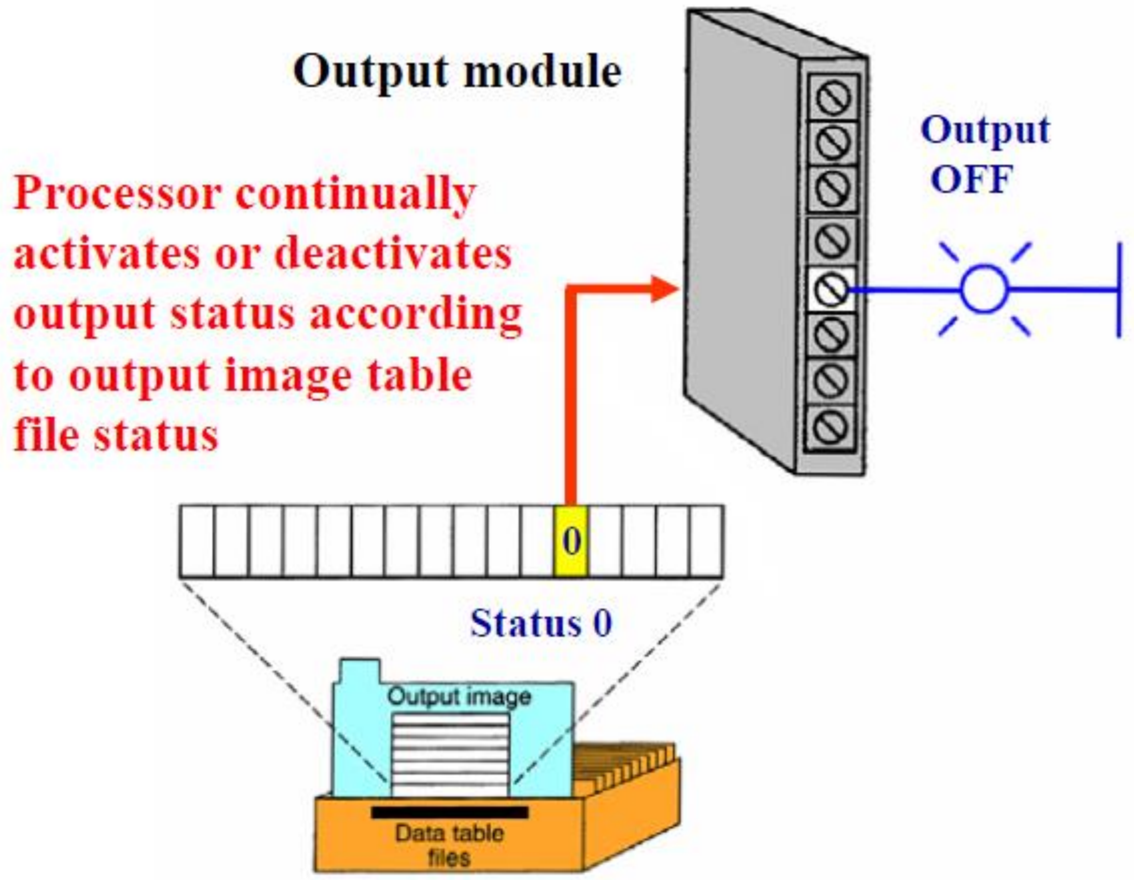
Output Table File Operation

Radu Muresan

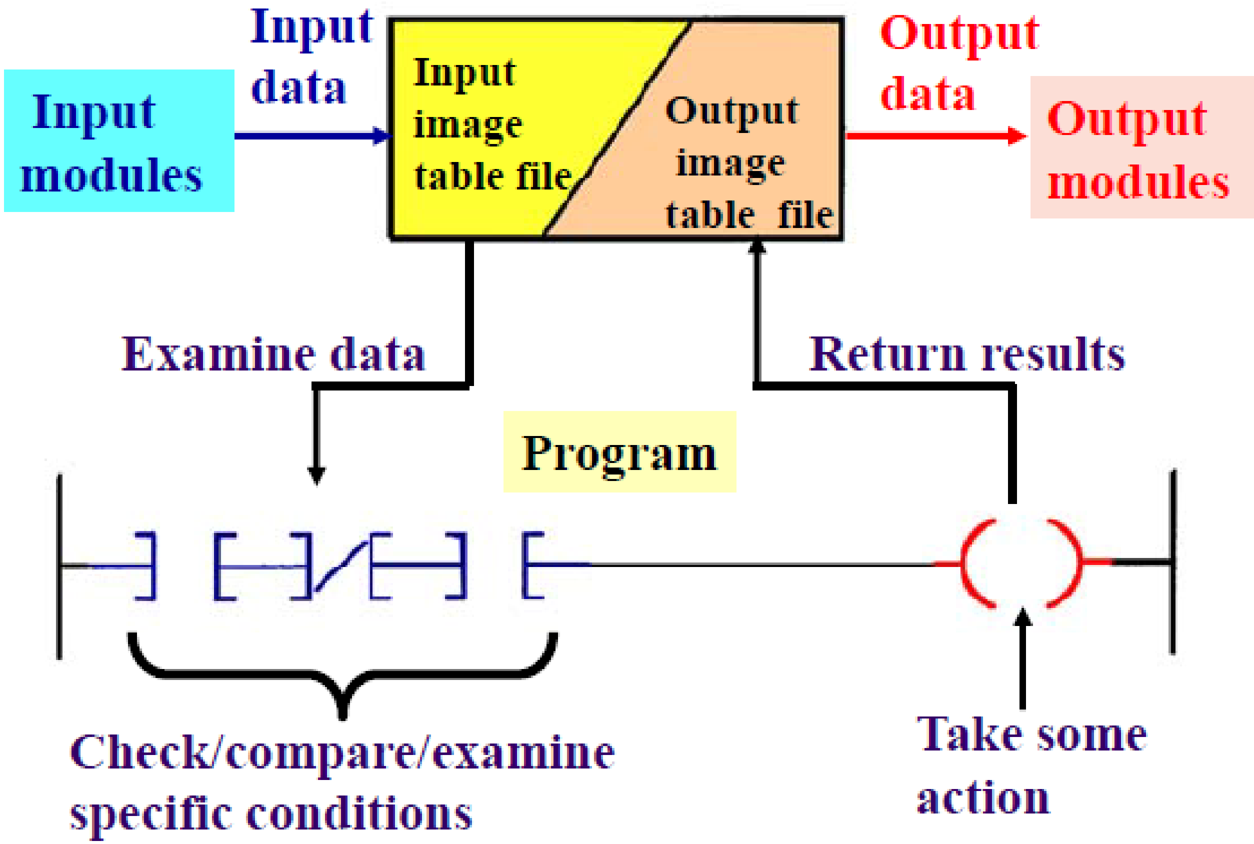


Output Table File Operation

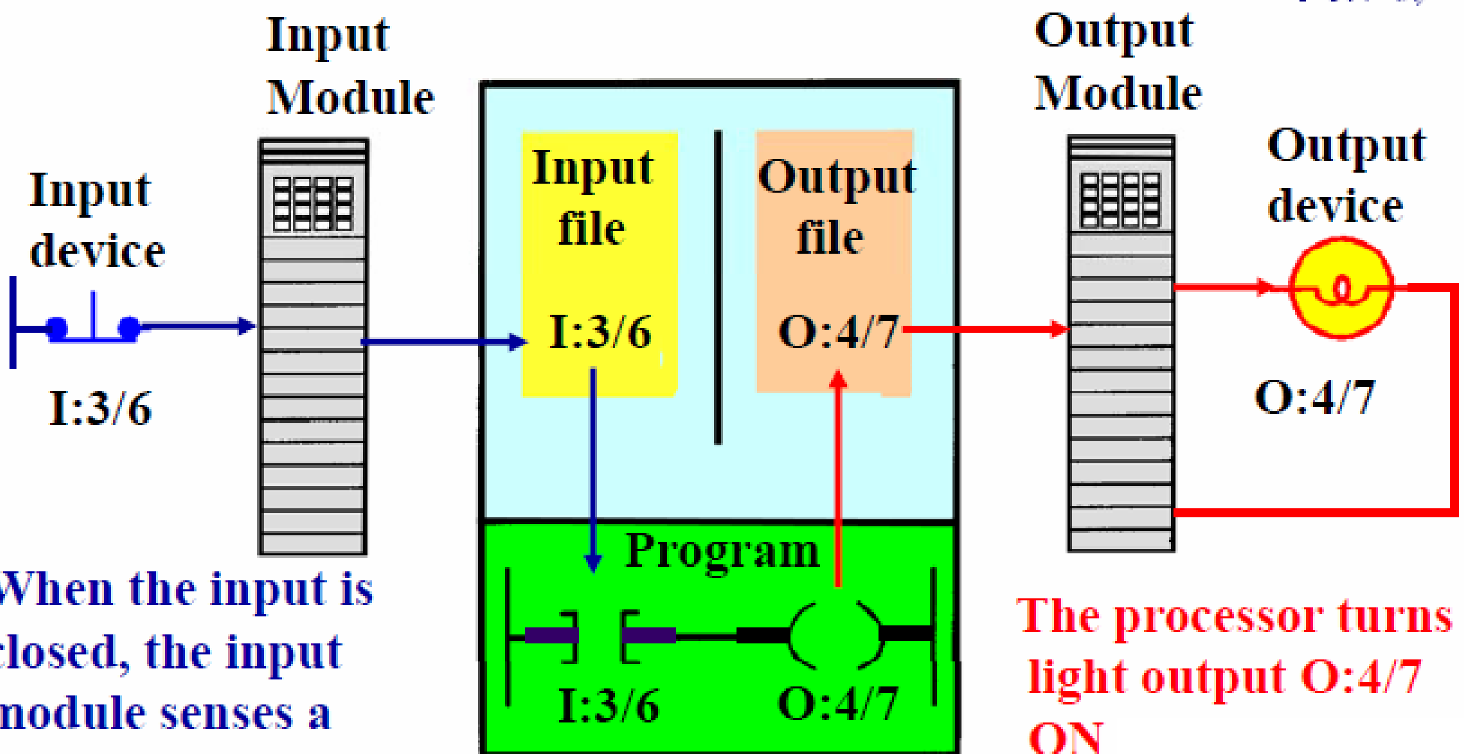
Radu Muresan



Data Flow Overview



Scan Process

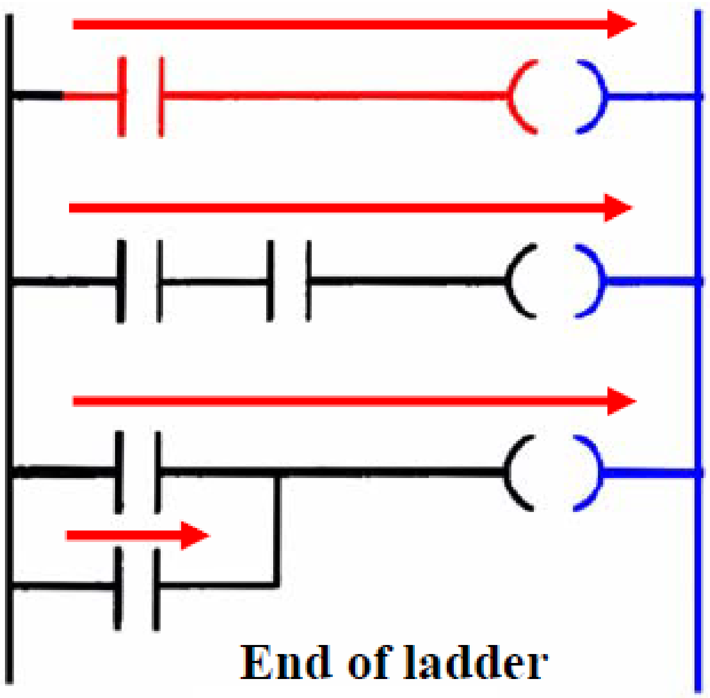


When the input is closed, the input module senses a voltage and an ON condition (1) is entered into the input table bit I:3/6

During the program scan the processor sets instructions I:3/6 and O:4/7 to ON (1).

The processor turns light output O:4/7 ON

Scan Patterns



Horizontal Scanning Order

The processor examines input and output instructions from the first command, top left in the program, horizontally, rung by rung.

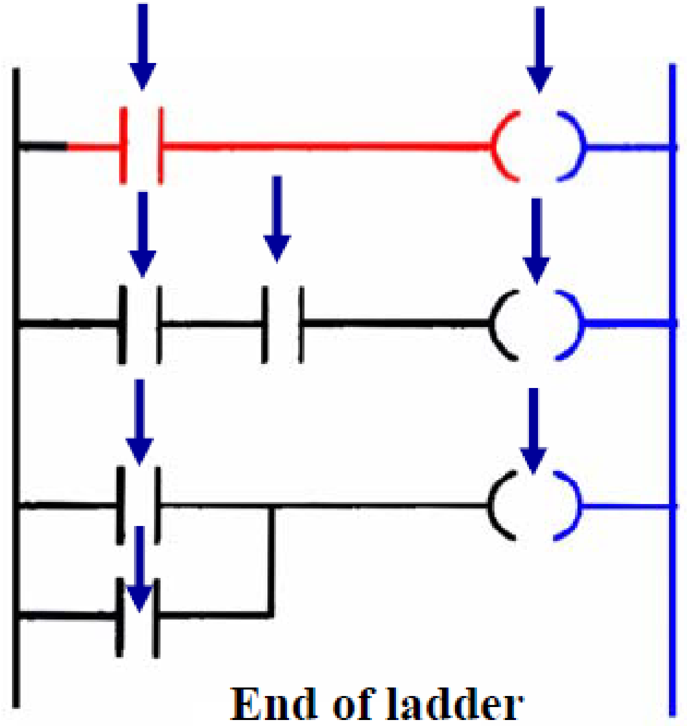
In addition to the program itself, the scan time is also dependent on the clock frequency of the processor!

Notice: We will use this scan pattern through out this course

Scan Patterns

Vertical Scanning Order

The processor examines input and output instructions from the first command, vertically, column by column and page by page. Pages are executed in sequence.

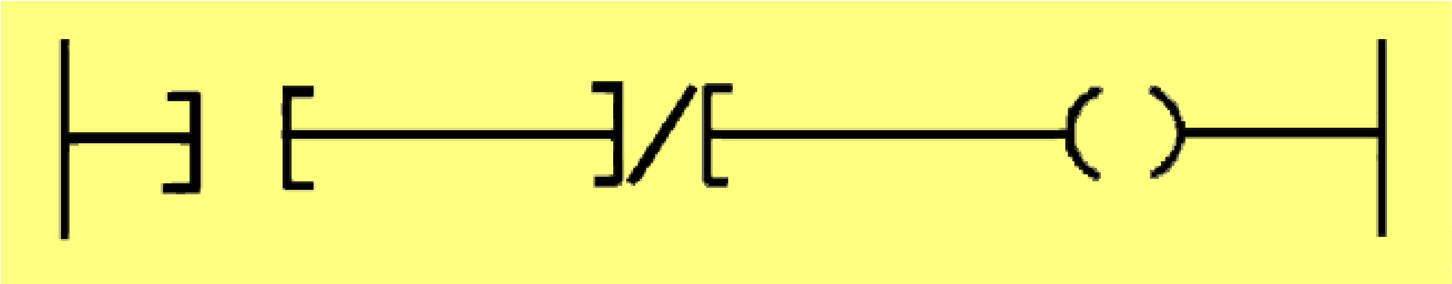


Misunderstanding the way the PLC scans can cause programming bugs!

Relay Type Instructions

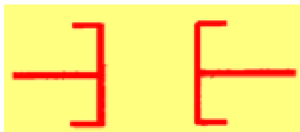
The ladder diagram language is basically a *symbolic* set of instructions used to create the controller program.

These ladder instructions symbols are arranged to obtain the desired control logic.



Examine If Closed (XIC) Instruction

Symbol

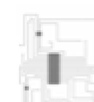


Analogous to the normally open relay contact. For this instruction we ask the processor to EXAMINE IF (the contact is) CLOSED (XIC)

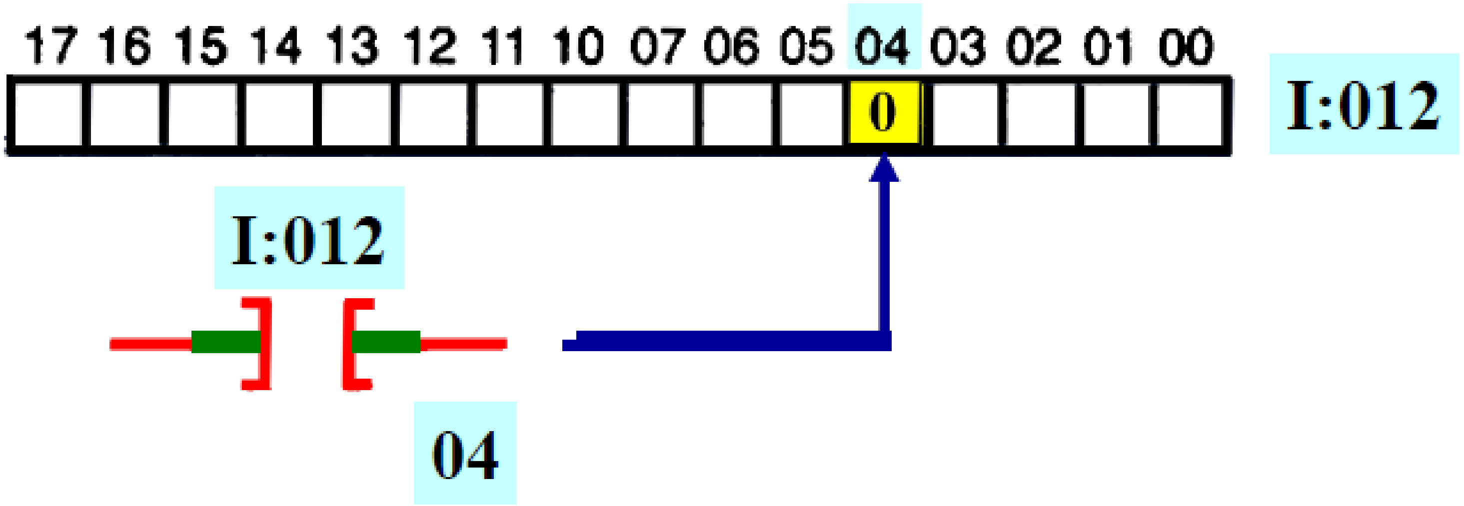
Typically represents any input. Can be a switch or pushbutton, a contact from a connected output, or a contact from an internal output.

Has a bit-level address which is examined for an ON condition.

The status bit will be either 1 (ON) or 0 (OFF).



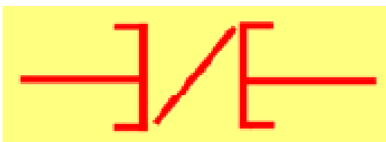
Examine If Closed (XIC) Instruction



If the status bit is 0 (OFF), then the instruction is FALSE.

Examine If Open (XIO) Instruction

Symbol



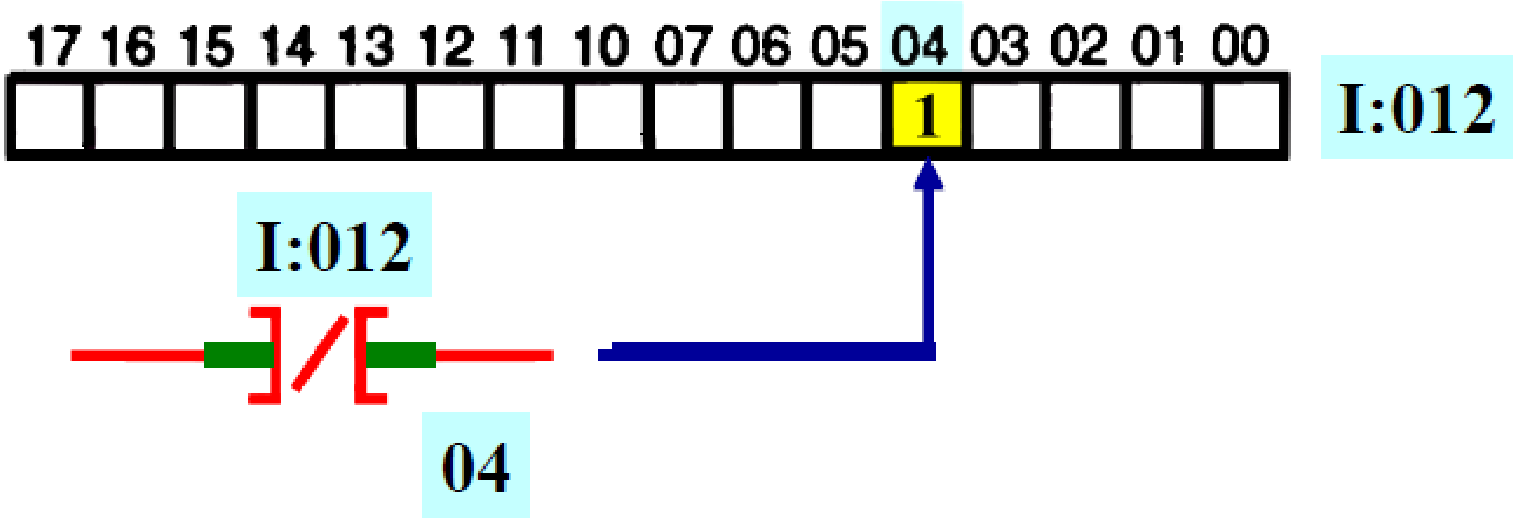
Analogous to the normally closed relay contact. For this instruction we ask the processor to EXAMINE IF (the contact is) OPEN (XIO).

Typically represents any input. Can be a switch or pushbutton, a contact from a connected output, or a contact from an internal output.

Has a bit-level address which is examined for an OFF condition.

The status bit will be either 1 (ON) or 0 (OFF).

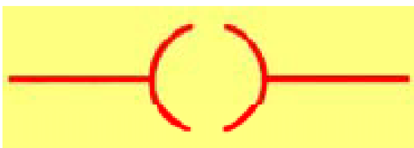
Examine If Open (XIO) Instruction



If the status bit is 1 (ON), then the instruction is FALSE.

Output Energize (OTE) Instruction

Symbol

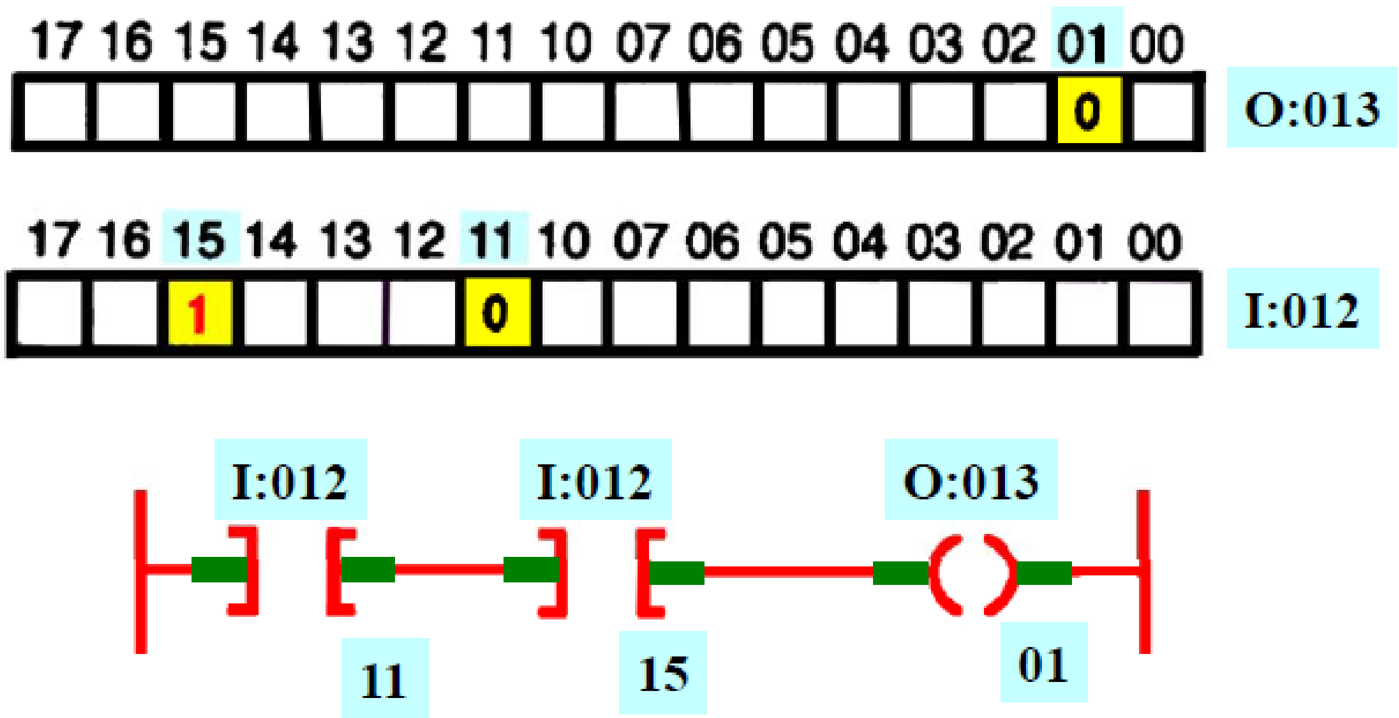


Analogous to the relay coil. The processor makes this instruction true (analogous to energizing a coil) when there is path of true XIC and XIO instructions in the rung.

Typically represents any output that is controlled by some combination of input logic. Can be a connected device or an internal output (internal relay).

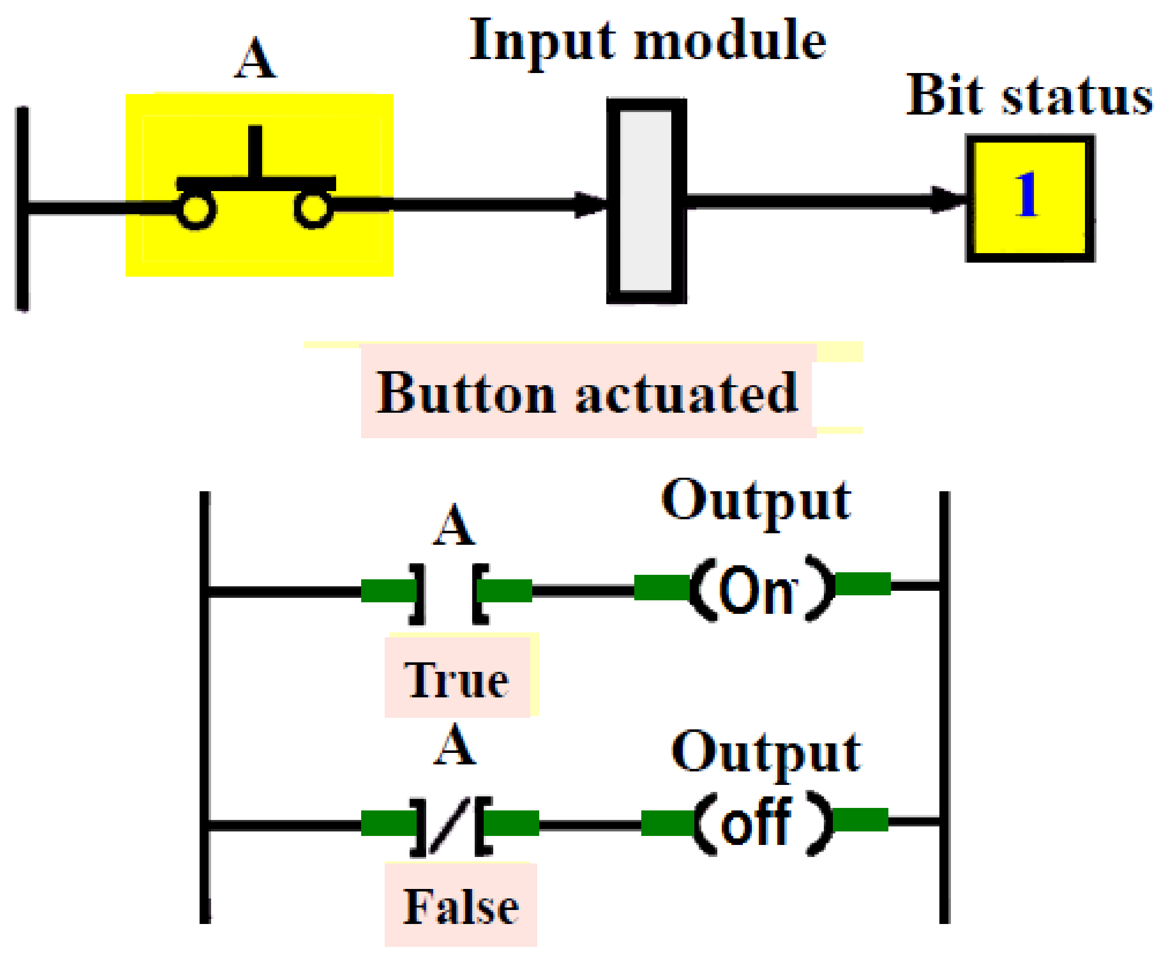
If any left-to-right path of input conditions is TRUE, the output is energized (turned ON).

Output Energize (OTE) Instruction

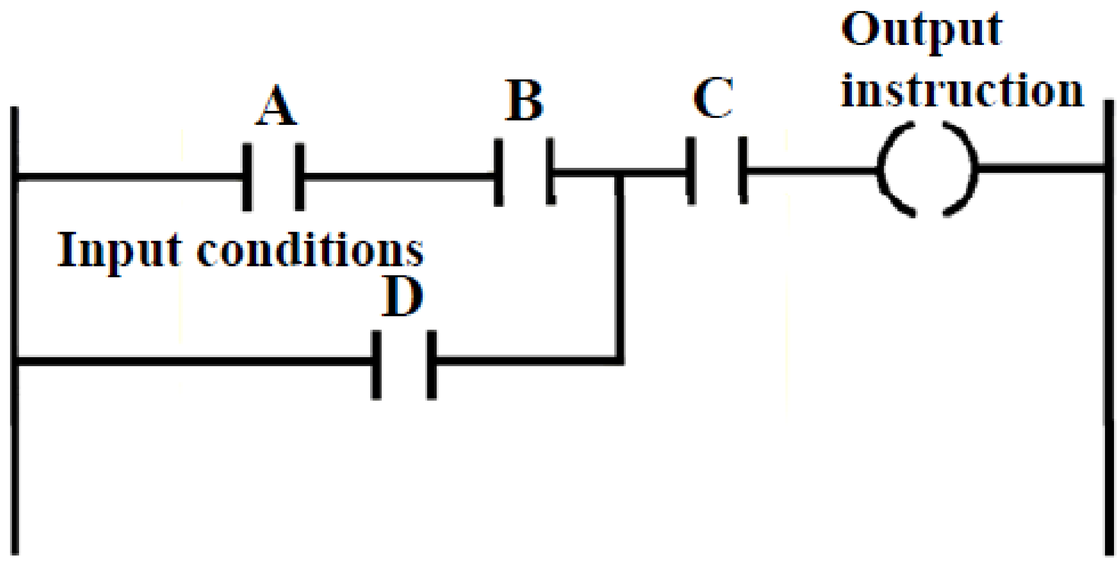


OUTPUT ENERGIZE instruction - FALSE

Status Bit Example



Ladder Rung

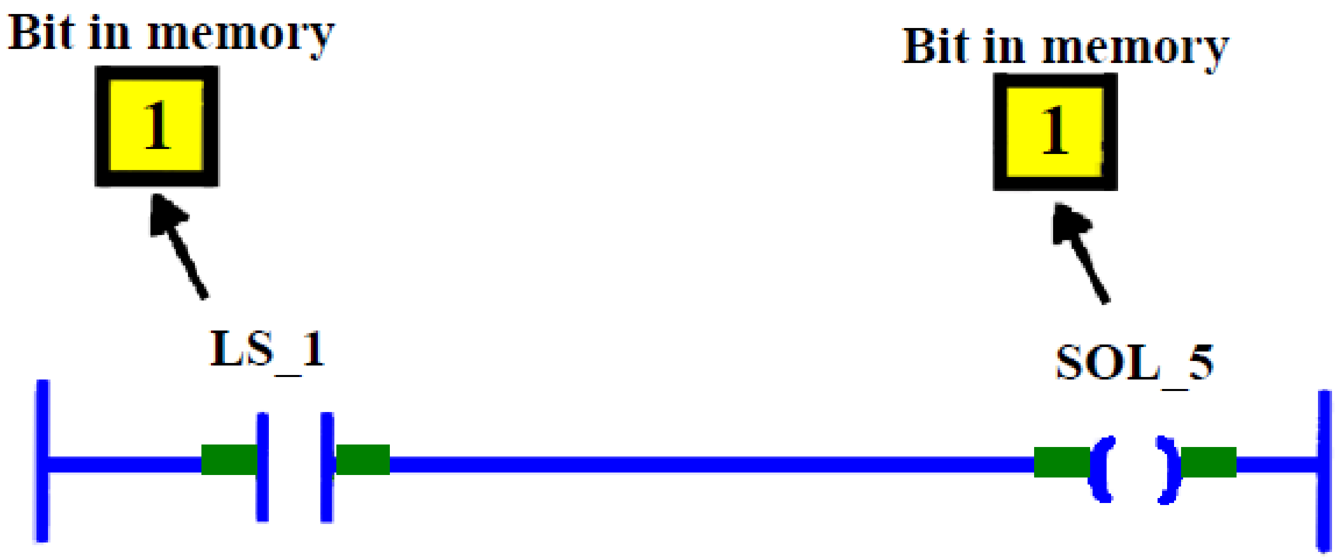


- A rung consists of a set of input conditions, represented by contact instructions, and an output instruction at the end of the rung, represented by the coil symbol.

Cont …

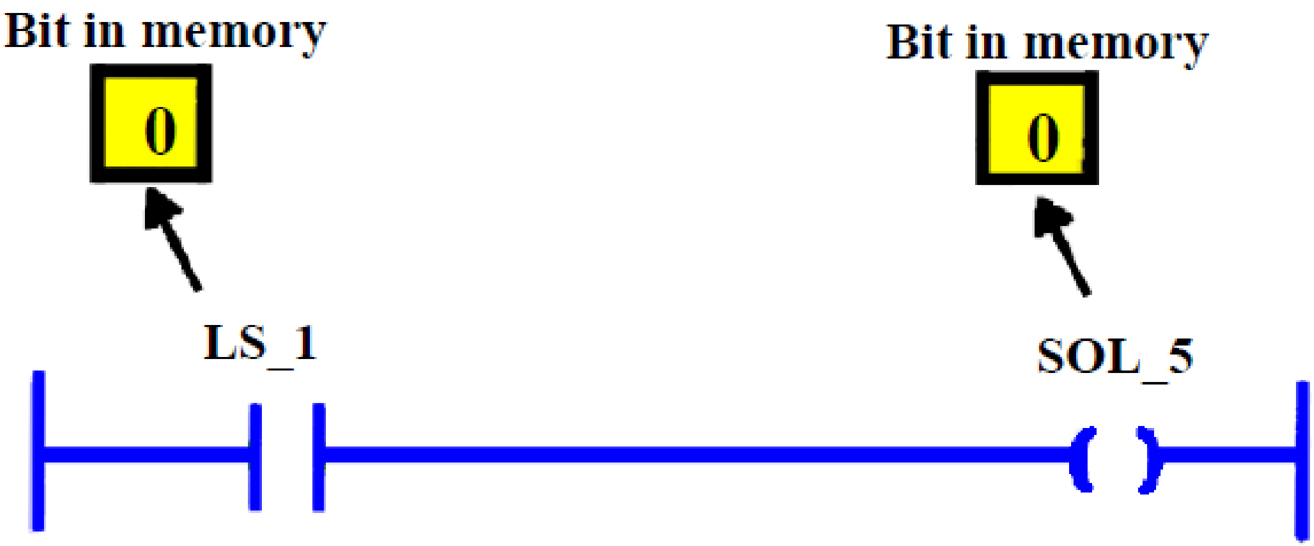
- Each contact or coil symbol is referenced with an address number that identifies what is being evaluated and what is being controlled.
- The same contact instruction can be used throughout the program whenever the condition needs to be evaluated.
- For an output to be activated or energized, at least one left-to-right path of contacts must be closed.
- A complete closed path is referred to as having logic continuity.
  - when logic continuity exists in at least one path, the rung condition is to be TRUE

Rung Continuity



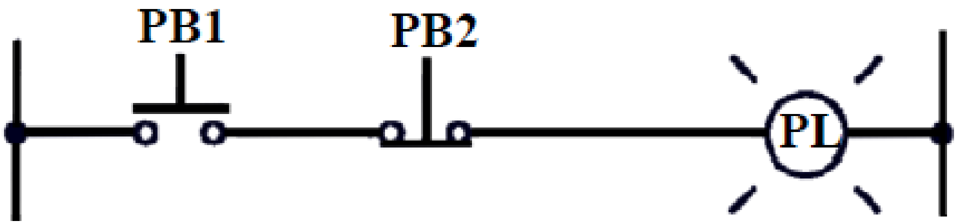
The Examine If Closed instruction is TRUE making the rung TRUE

Rung Continuity

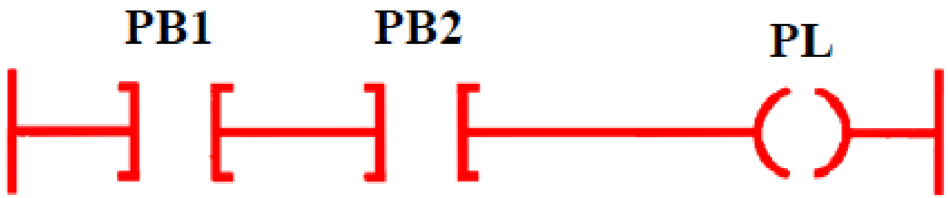


The Examine If Closed instruction is FALSE making the rung False

Programming The XIC Instruction



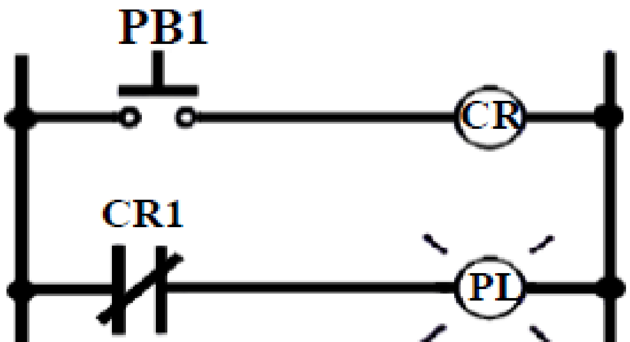
Hardwired Circuit



User program providing the same results

Note that both pushbuttons are represented by the XIC symbol. This is because the normal state of an input (NO or NC) does not matter! What does matter is that if contacts need to close to energize the output, then the XIC instruction is used. Since both PB1 and PB2 must close to energize the PL, the XIC instruction is used for both.

Programming the XIC Instruction



Hardwired Circuit

User program providing the same results

When the pushbutton is *open* in the hardwired circuit, relay coil CR is de-energized and contacts CR1 close to switch the PL on. When the pushbutton is *closed*, relay coil CR is energized and contacts CR1 open to switch the PL off. The pushbutton is represented in the user program by an XIO instruction. This is because the rung must be true when the external pushbutton is *open*, and false when the pushbutton is *closed*.

Operation of XIC and XIO Instructions

Summary of status conditions

If the data table bit is:	The status of the instruction is:		
	XIC Examine If Closed	XIO Examine If Open	OTE Output Energize
Logic 0	False	True	False
Logic 1	True	False	True

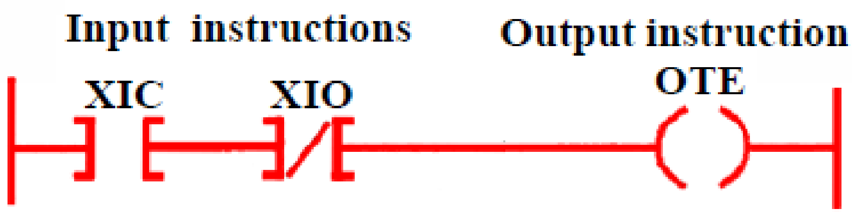
Operation of the XIC and XIO Instructions

Radu Muresan

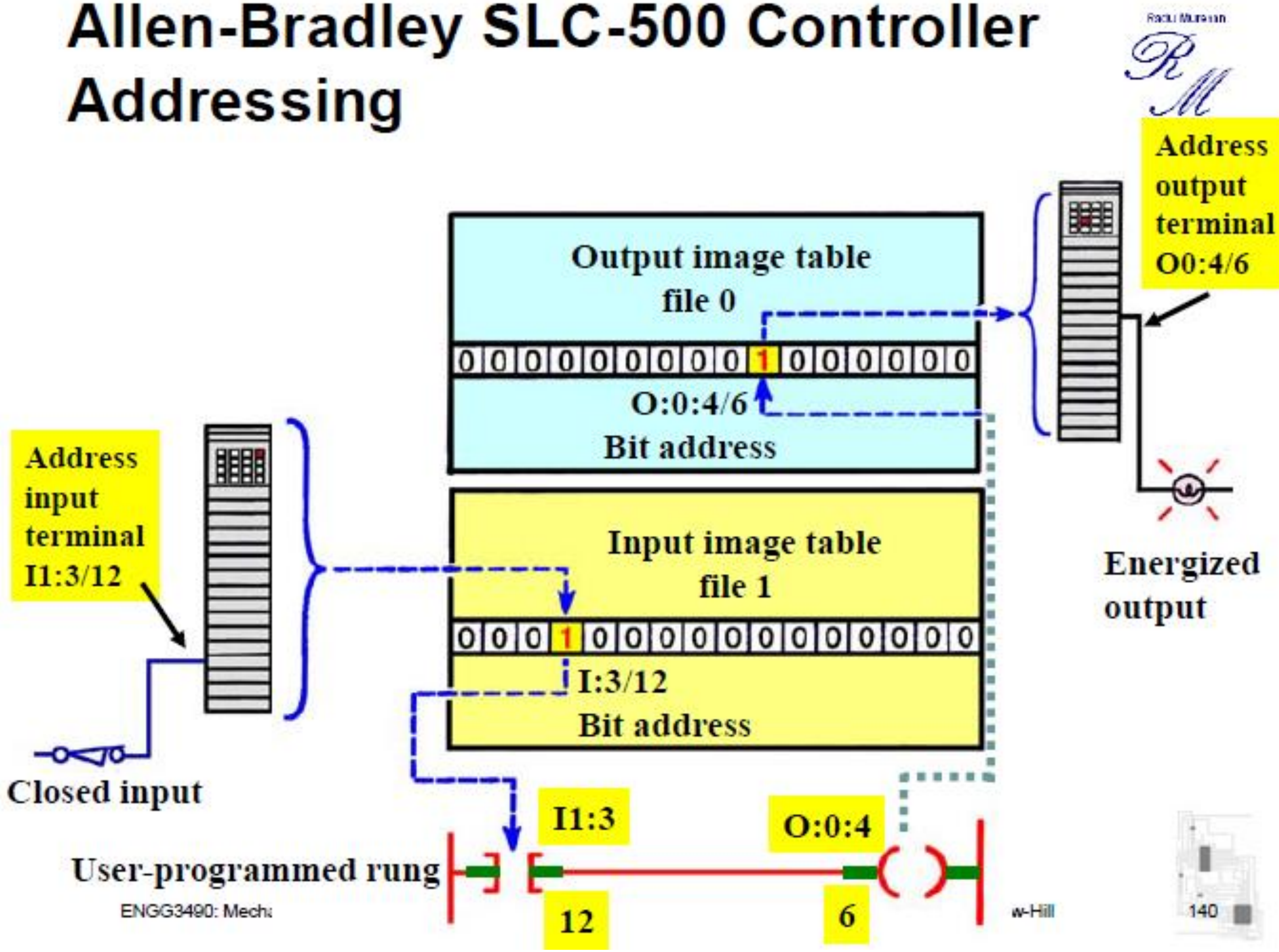
State of the output as determined by the changing state of the inputs in the rung

Time	Inputs		Output
	XIC	XIO	OTE
t_1 (initial)	False	True	False
t_2	True	True	Goes true
t_3	True	False	Goes false
t_4	False	False	Remains false

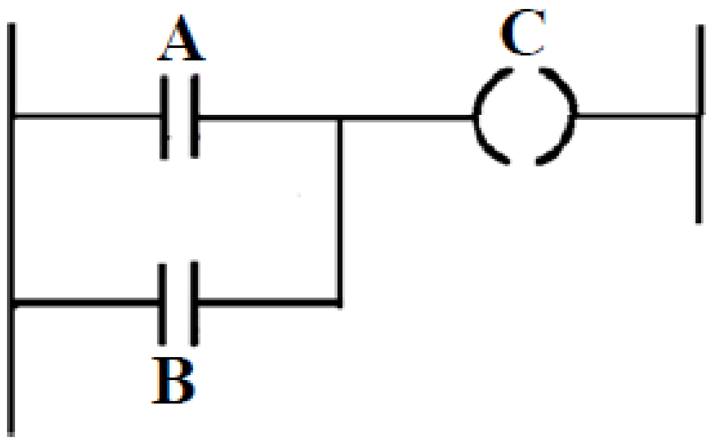
Bit status		
XIC	XIO	OTE
0	0	0
1	0	1
1	1	0
0	1	0



Allen-Bradley SLC-500 Controller Addressing

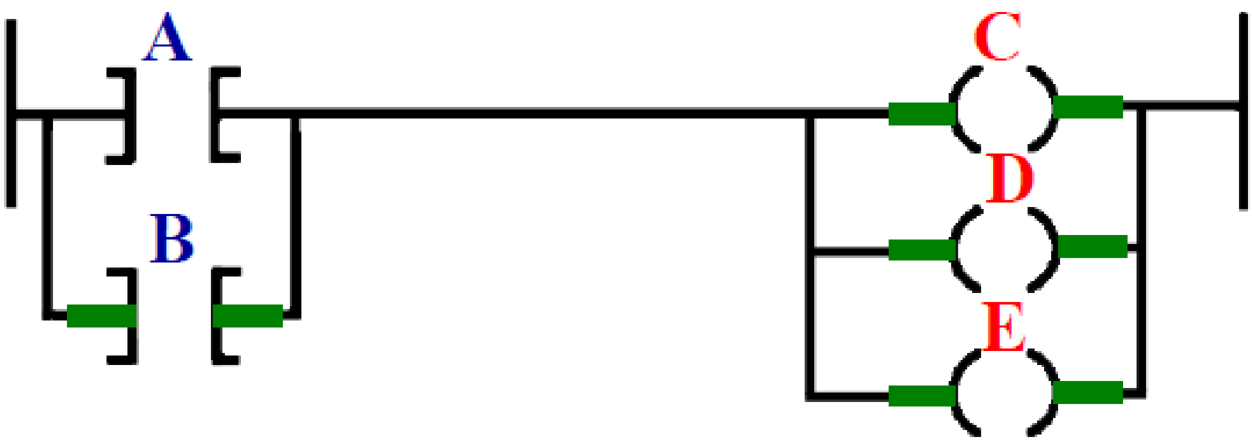


Parallel Input Branch Instructions



Branch instructions are used to create parallel paths of input condition instructions. If at least one of these parallel branches forms a true logic path, the logic is enabled.

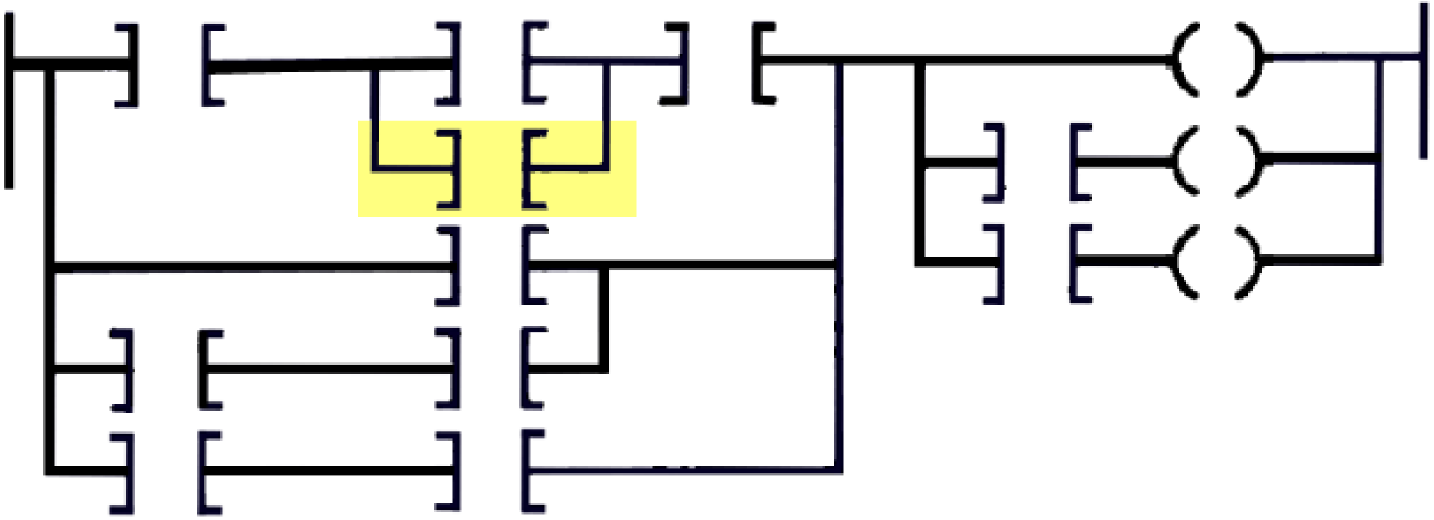
Parallel Output Branching



On most PLC models, branches can be established at both the input and output portion of the rung.

With output branching, you can program parallel outputs on a rung to allow a true logic path to control multiple outputs.

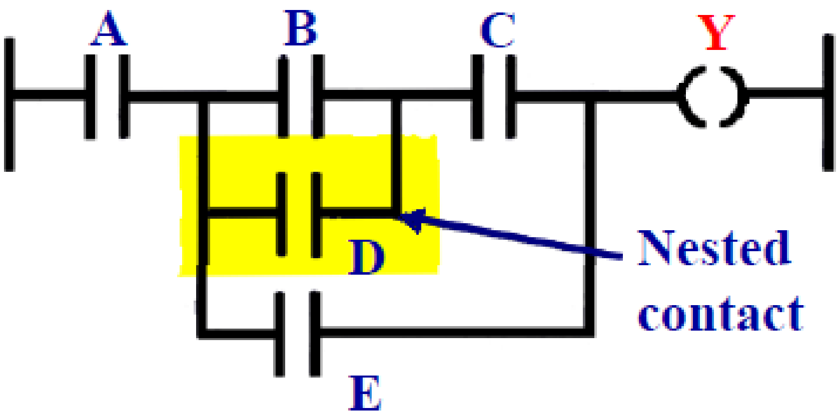
Nested Input and Output Branches



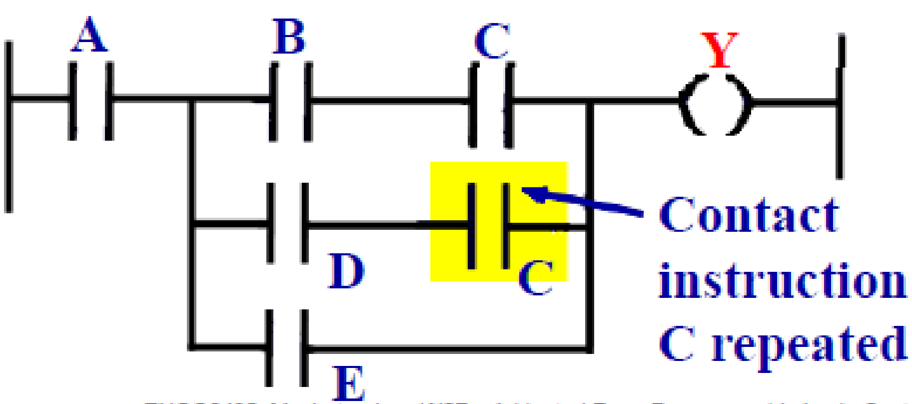
Input and output branches can be *nested* to avoid redundant instructions and to speed up the processor scan time.

A nested branch starts or ends within another branch.

Nested Contact Program

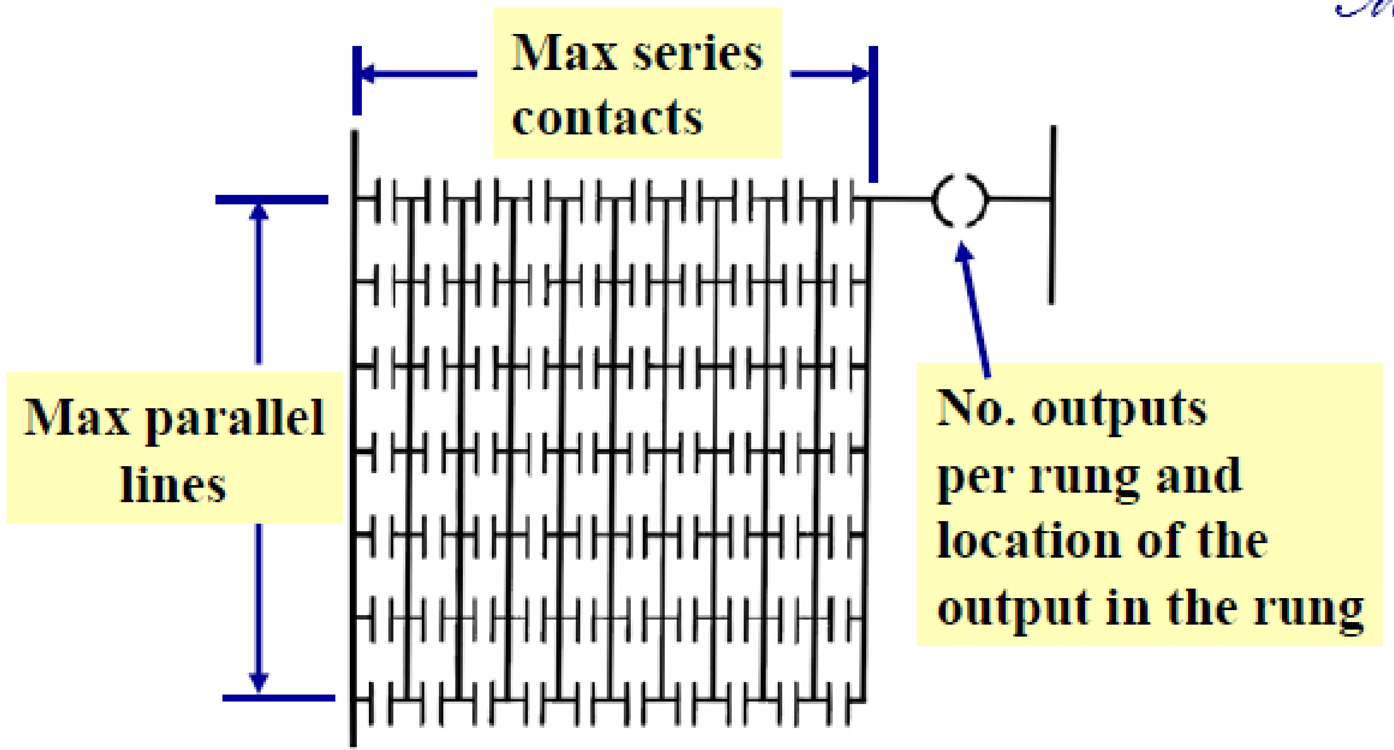


On some PLC models, the programming of a nested branch circuit cannot be done directly.



Reprogrammed to obtain the required logic.

PLC Matrix Limitation Diagram



There may be limitations to the number of series contacts instructions, number of parallel lines, and the number of outputs and their location on the rung.

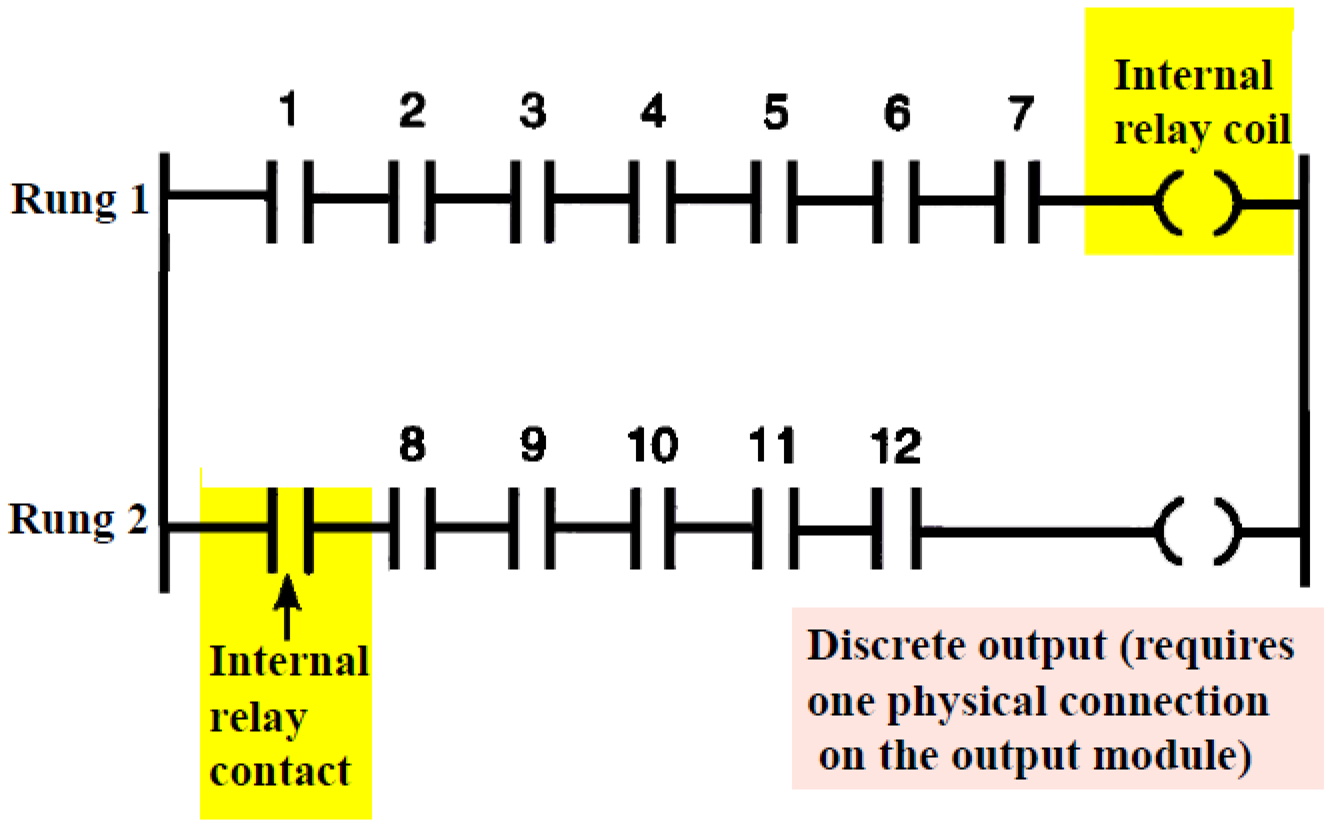
Internal Control Relay

The internal output operates just as any other output that is controlled by programmed logic; however, the output is used strictly for internal purposes.

The internal output does not directly control an output device.

The advantage of using internal outputs is that there are many situations where an output instruction is required in a program, but no physical connection to a field device is needed. Their use in this type of instance can minimize output card requirements.

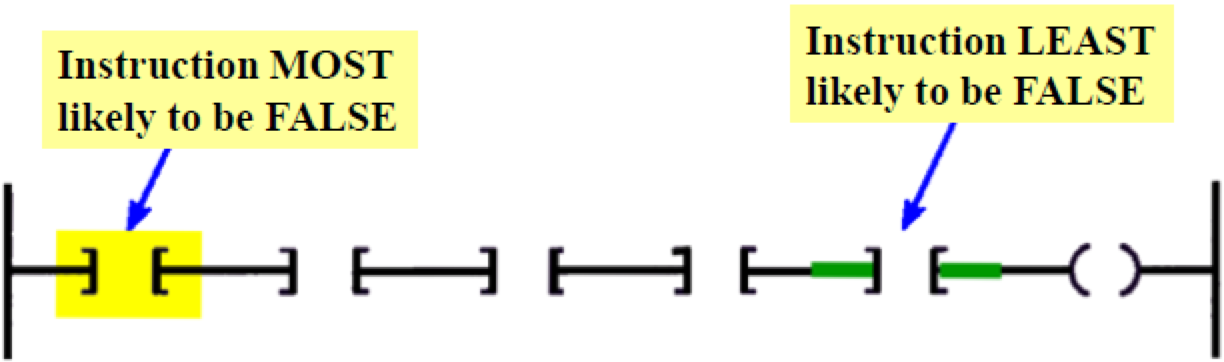
Extending the Number of Series Contacts Using an Internal Control Relay



Arranging Instructions for Optimum Performance

There is more than one way to correctly implement the ladder logic. In some cases one arrangement may be more efficient in terms of the amount of memory used and the time required to scan the program.

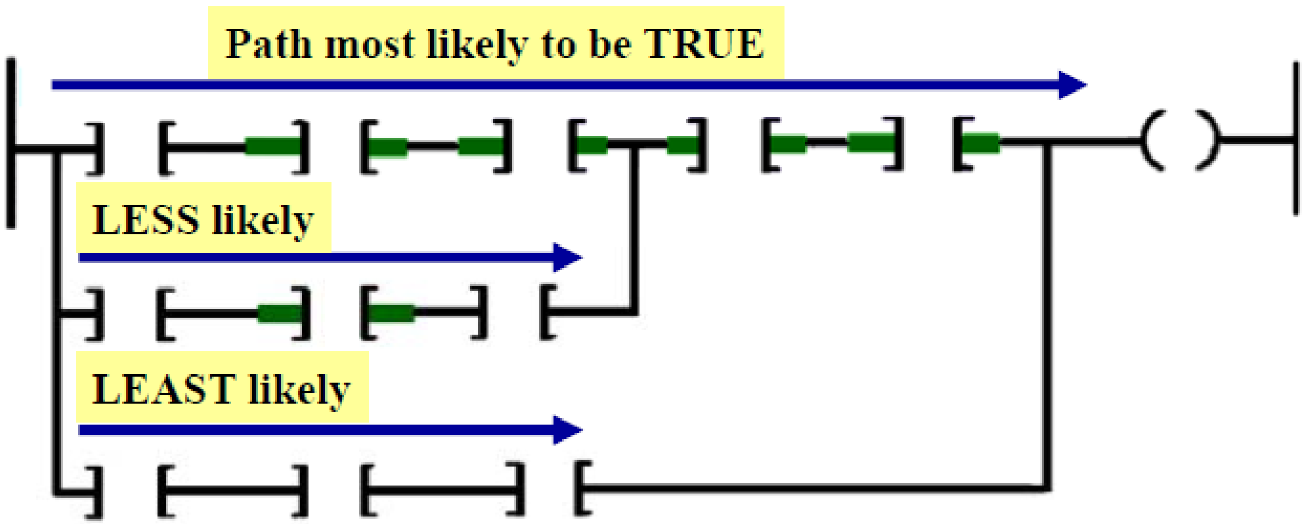
Sequence series instructions from the most likely to be FALSE (far left) to least likely to be FALSE (far right)



Once a processor sees a FALSE input instruction in series, it executes the remaining instructions FALSE, even if they are TRUE

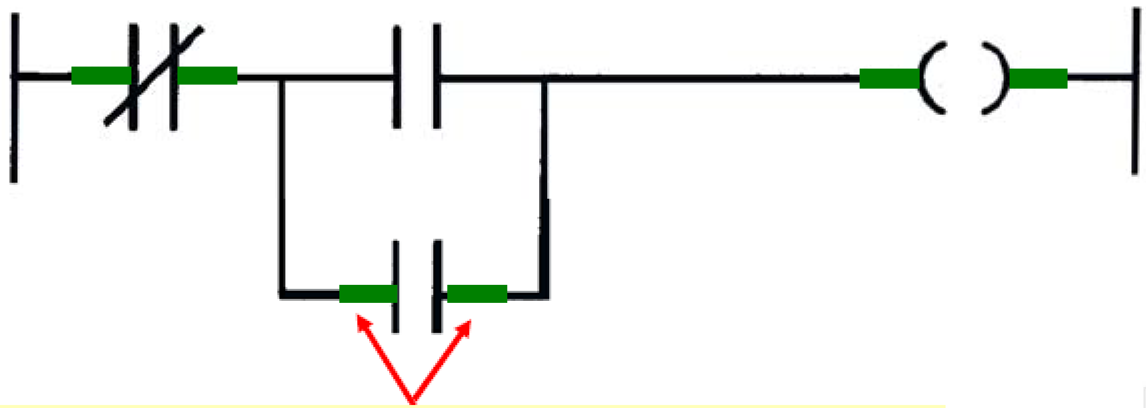
Arranging Instructions for Optimum Performance

If your rung contains parallel branches, place the path that is most often TRUE on the top. The processor will not look at the others unless the top path is FALSE.



Monitoring a Ladder Logic Program

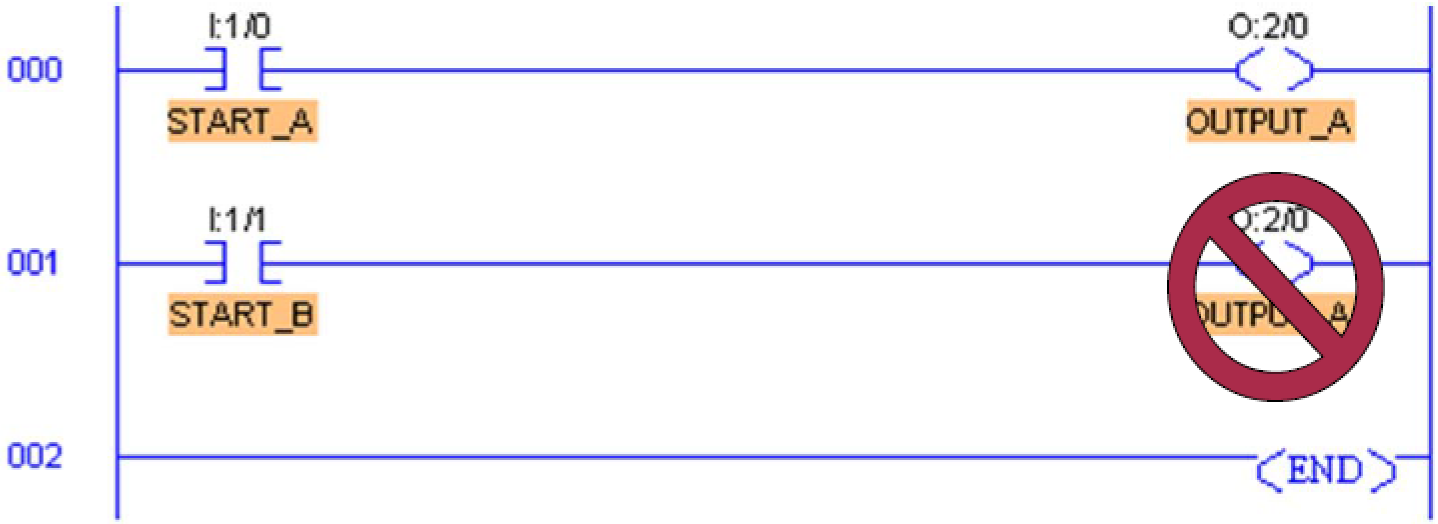
Operation of the logic is apparent from the highlighting of rungs of the various instructions on screen, which identifies the logic state in real time and has logic continuity.



Highlighted rungs indicate the instruction is true

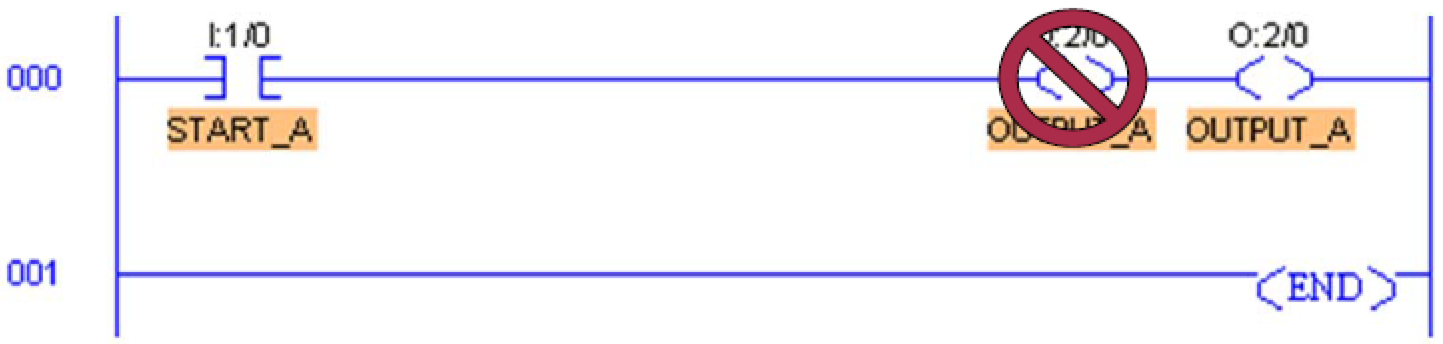
Properly formatted outputs

- An output energize instruction (OTE) referencing a specific output bit should appear only once in a ladder logic program



Properly formatted outputs

- Only one output energize instruction (OTE) should appear in a rung of ladder logic



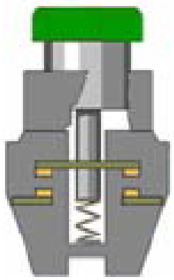
Properly formatted outputs

- If more than one output is to be controlled by a certain rung of ladder logic, the output energize (OTE) instructions can be placed in parallel

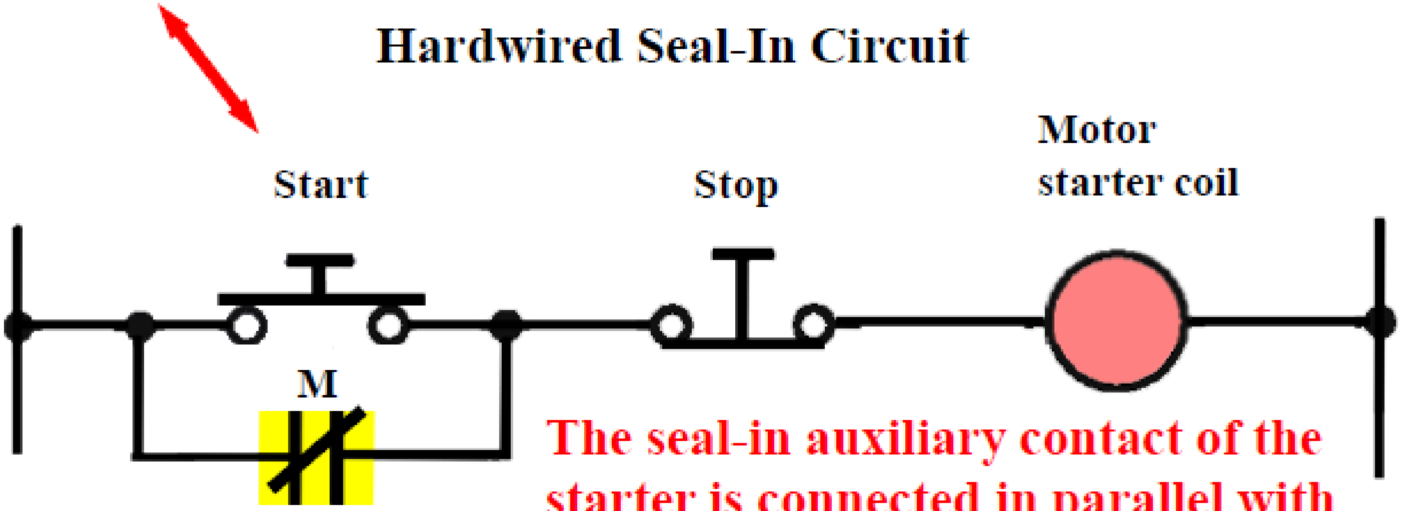
Start-stop-seal circuits

- For PLC systems without latch and unlatch instructions, a circuit is needed that will allow a process to start, continue to run after a start button is released, and stop under control of another button
  - A circuit that implements this functionality is commonly referred to as a **start-stop-seal** circuit
- A feedback path (i.e. a contact) that references the output is normally used to **seal** around the start contact

Seal-In Circuits

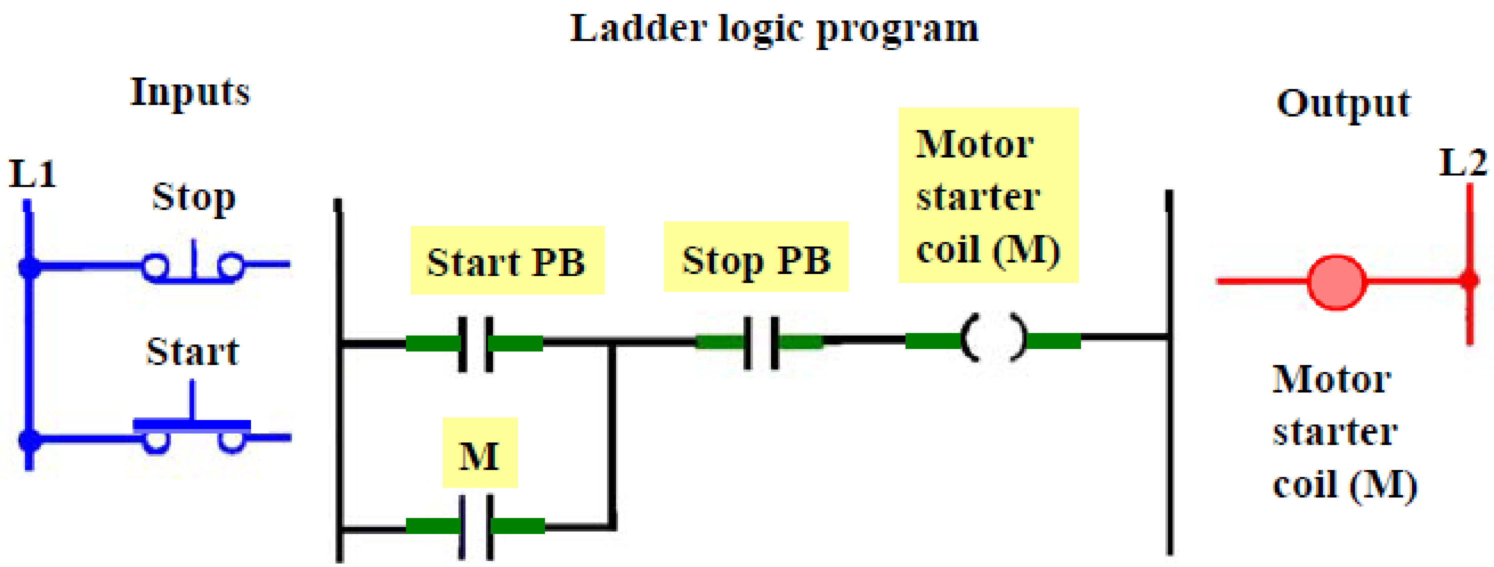


A seal-in circuit is a method of maintaining current flow after a momentary switch has been pressed and released.



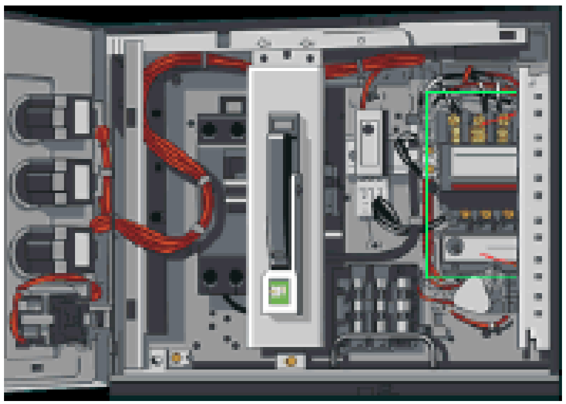
The seal-in auxiliary contact of the starter is connected in parallel with the Start button to keep the starter coil energized when the Start button is released.

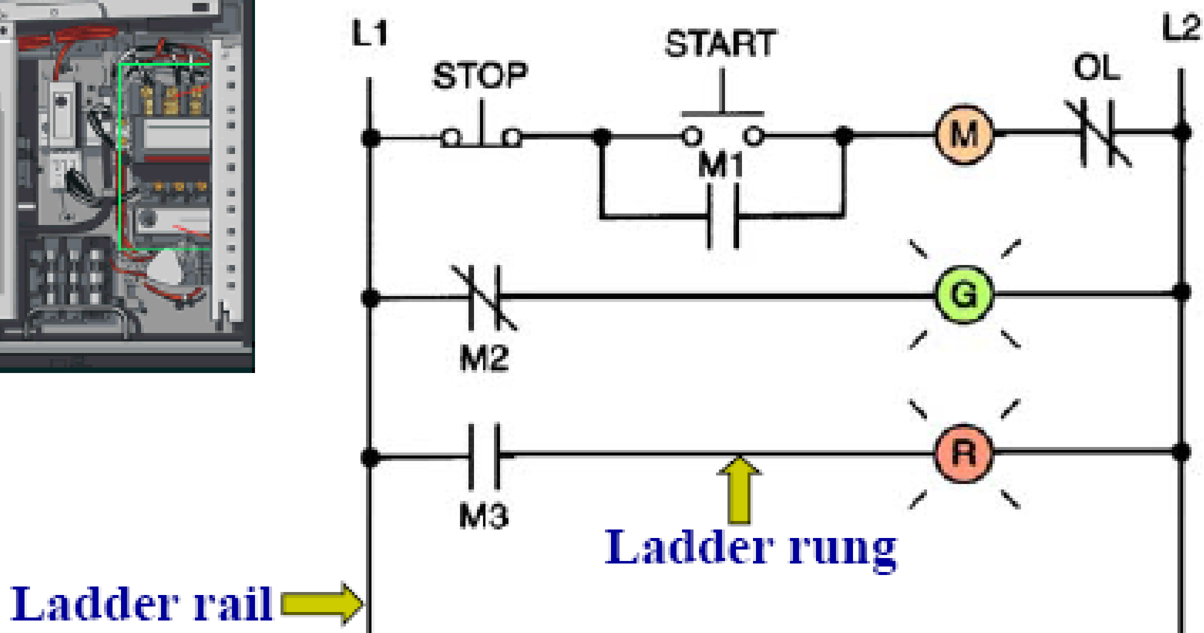
Programmed Seal-In Circuits



Both the Start and Stop buttons are examined for a closed condition because both buttons must be closed to cause the motor starter to operate.

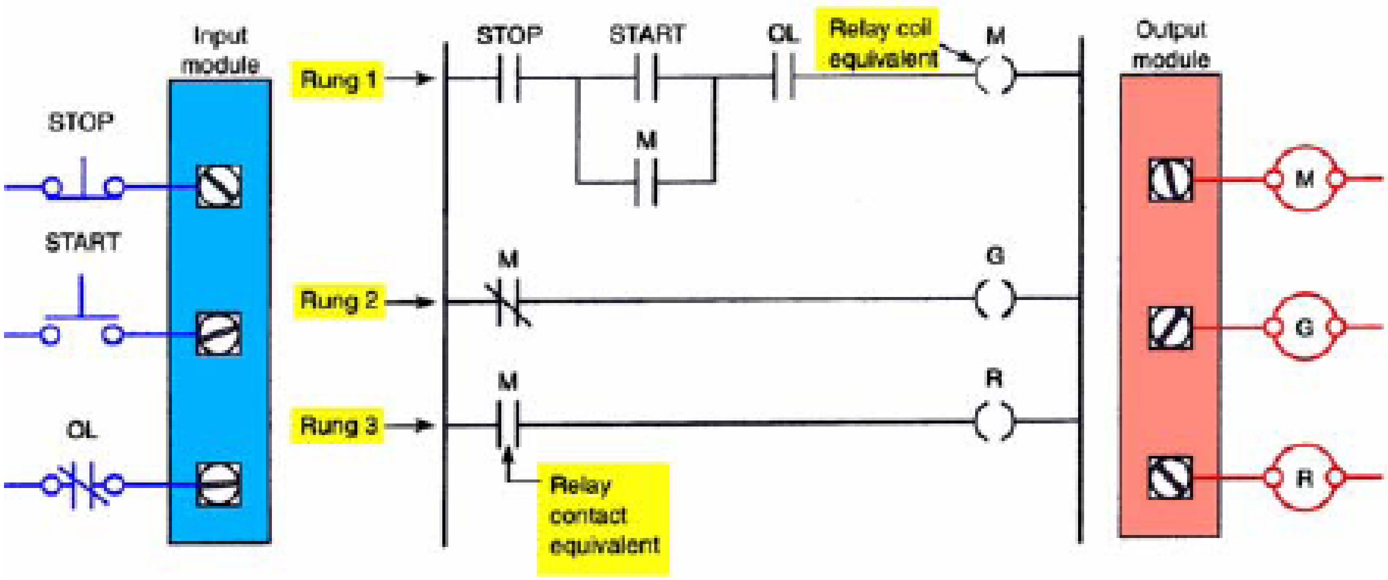
Hardwired Stop/Start Motor Control Circuit



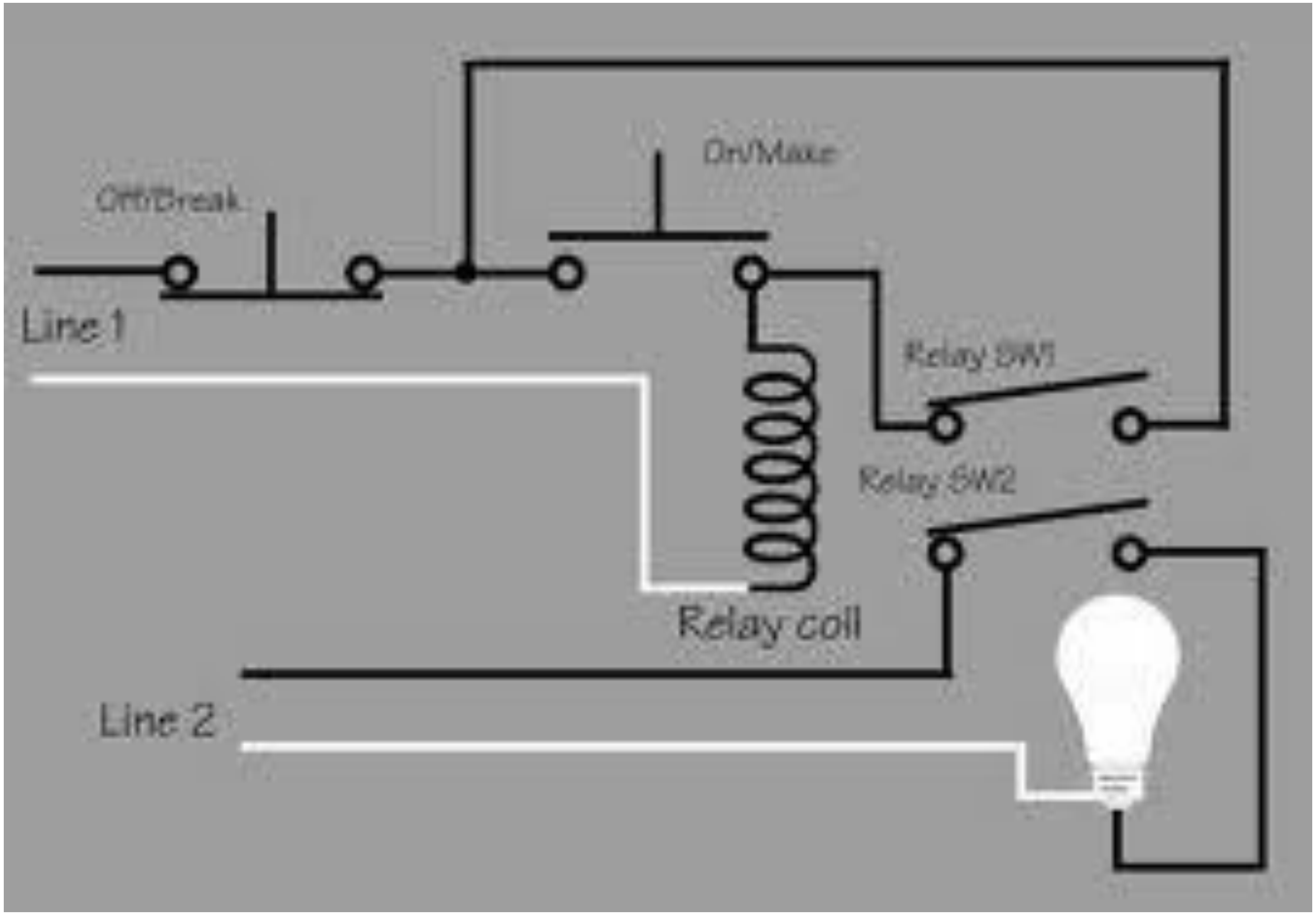


Control scheme is drawn between two vertical supply lines.

Programmed Stop/Start Motor Control Circuit



- A rung is the contact symbolism required to control an output. Each rung is a combination of input conditions connected from left to right with the symbol that represents the output at the far right.
- The instructions used are the relay equivalent of normally open (NO) and normally closed (NC) contacts and coils


Off/Break
On/Make
Line 1
Relay SW1
Relay SW2
Relay coil
Line 2

references:

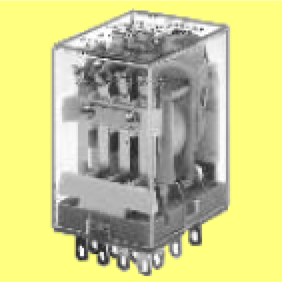
1-notes from Dr. **Jeff Jackson** *,the university of Alabama*

2-notes from Dr. Radu Muresan ,University of Guelph

Electromagnetic Control Relay

Radu Muresan

The PLC's original purpose was the replacement of electromagnetic relays with a solid-state switching system that could be programmed.



The programmable controller was designed to replace physically small control relays that make logic decisions but are not designed to handle heavy current or high voltage.

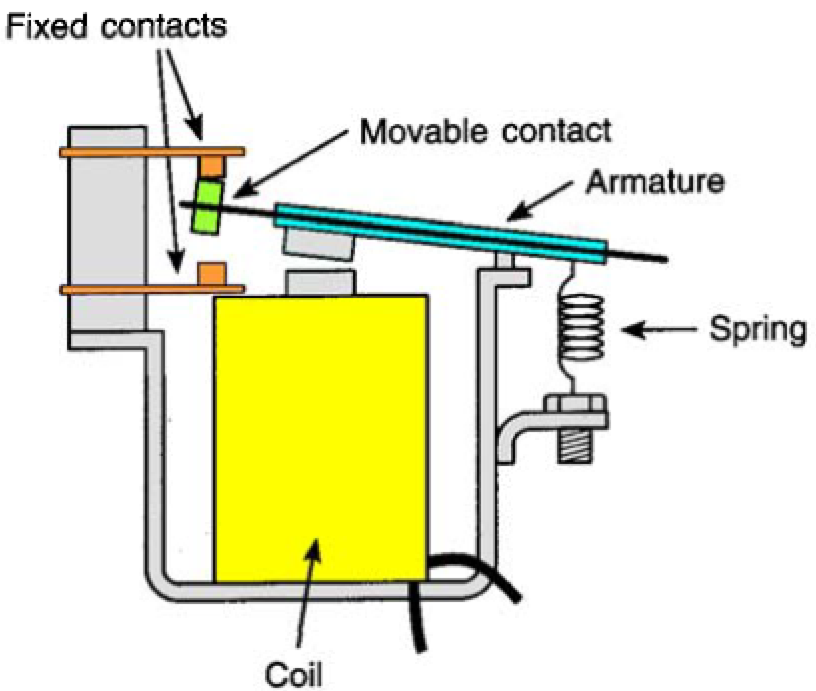
Electromagnetic relays, such as the lighting contactor shown, are still used as auxiliary devices to switch I/O field devices.

Addapted From Programmable Logic Controllers By F. D. Petruzella, McGraw-Hill

Electromagnetic Relay Operation

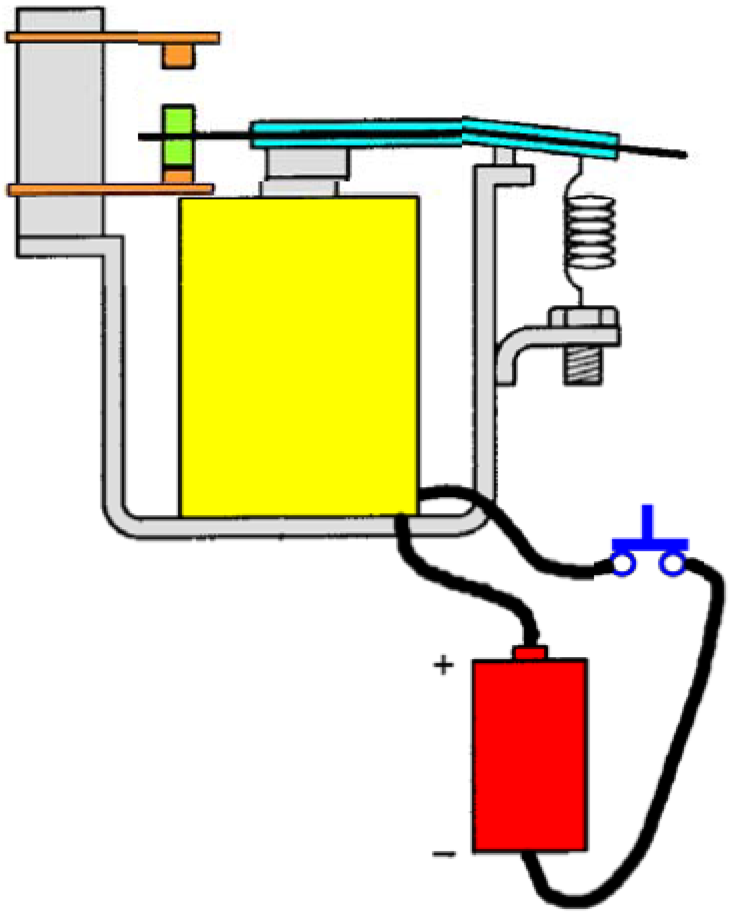
Radu Muresan

An electromagnetic relay is a magnetic switch. It uses electromagnetism to switch contacts.



A relay will usually have only one coil but may have any number of different contacts.

Electromagnetic Relay Operation

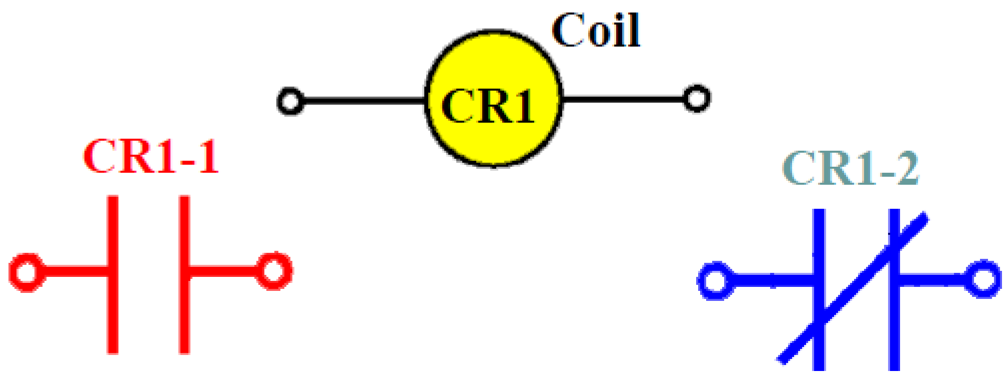


With no current flow through the coil (coil de-energized), the armature is held away from the core by spring tension.

When the coil is energized, the electromagnetic field moves the armature causing the contact points of the relay to open or close.

Radu Muresan

Relay Symbol



Normally open (NO) contact

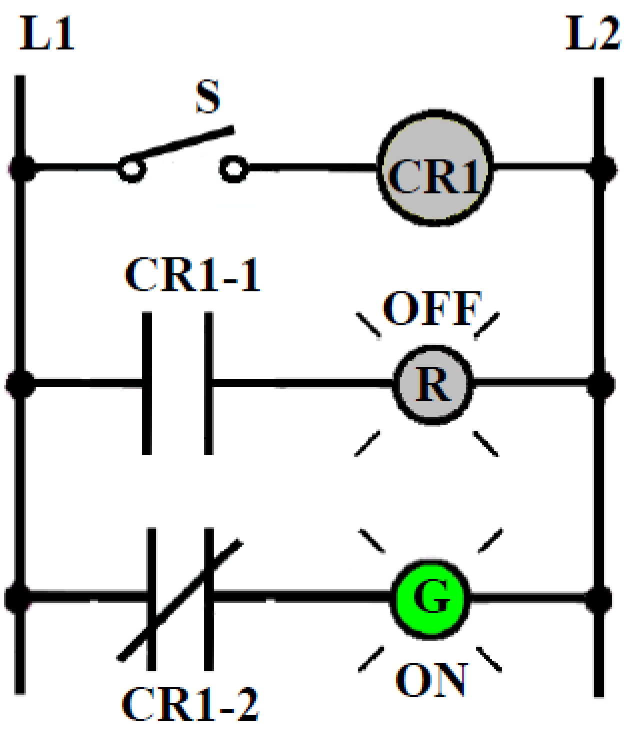
Normally closed (NC) contact

Contacts are open when no current flows through the coil but close as soon as the coil is energized.

Contacts are closed when no current flows through the coil but open as soon as the coil is energized.

Relay Circuit Operation

Radu Muresan

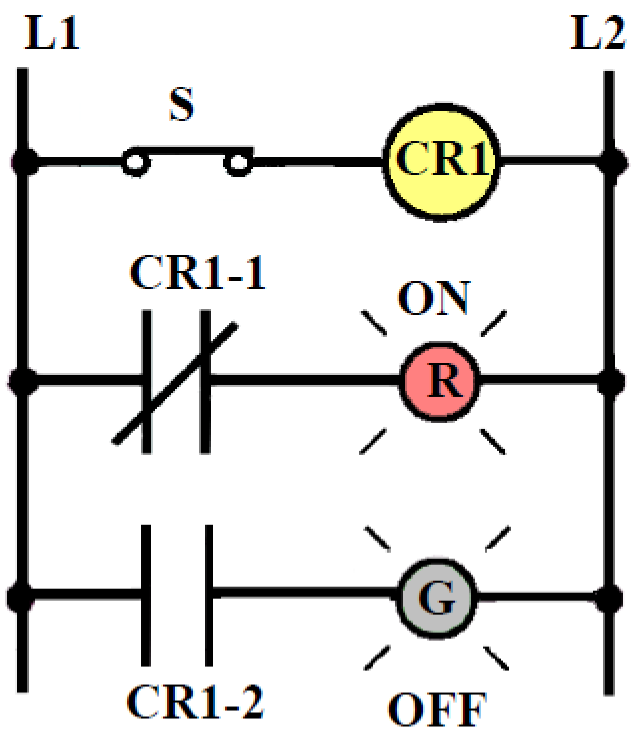


With switch S open:

- coil CR1 is de-energized
- contacts CR1-1 are open
- light R is off
- contacts CR1-2 are closed
- light G is on

Relay Circuit Operation

Radu Muresan



With switch S closed:

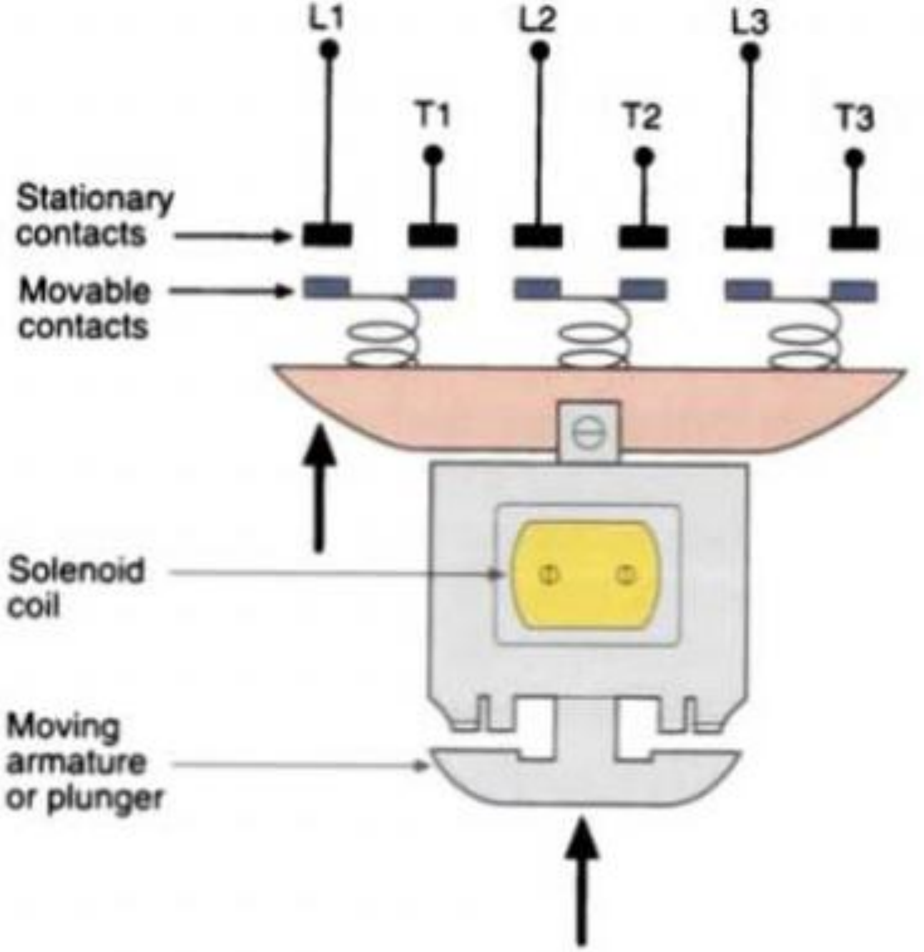
- coil CR1 is energized
- contacts CR1-1 are closed
- light R is on
- contacts CR1-2 are open
- light G is off

Magnetic Contactor

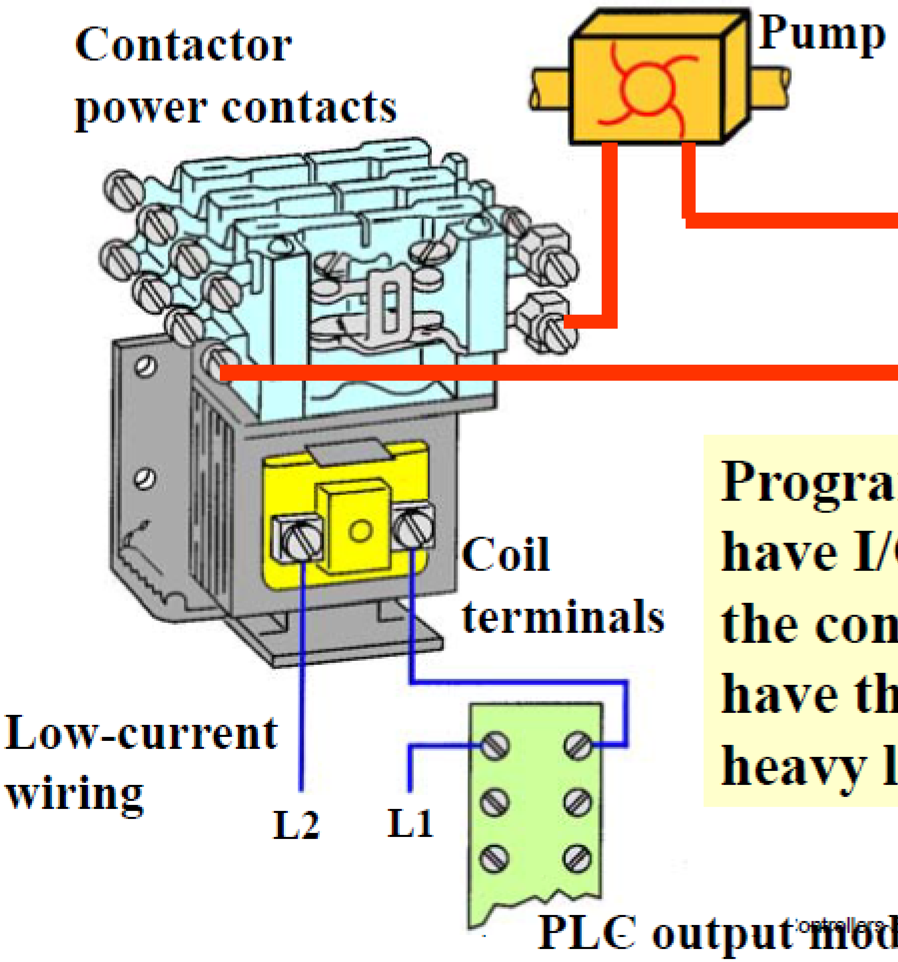
Radu Muresan

A contactor is a special type of relay designed to handle heavy power loads that are beyond the capability of control relays.

Contactors are designed to operate such loads as lights, heaters, transformers, capacitors, and electric motors for which overload protection is provided separately or not required.


L1
L2
L3
T1
T2
T3
Stationary contacts
Movable contacts
Solenoid coil
Moving armature or plunger

PLC Used in Conjunction with a Contactor



Programmable controllers have I/O capable of operating the contactor but they do not have the capacity to operate heavy loads directly.

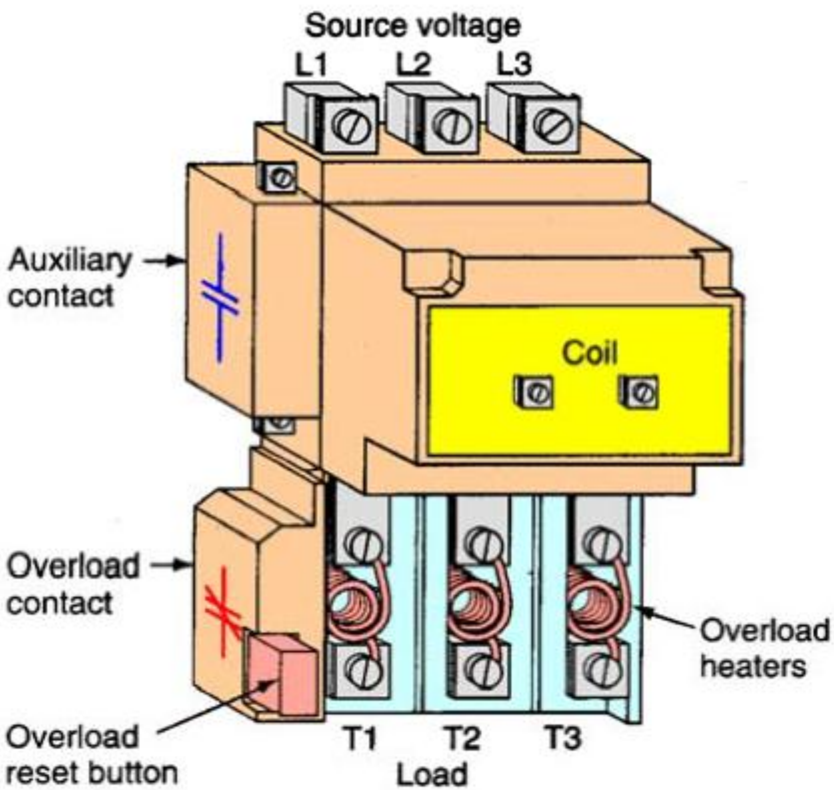
Magnetic Motor Starter

Radu Muresan

A magnetic motor starter is a contactor with an *overload relay* attached physically and electrically. They are electromagnetically operated switches that provide a safe method for starting large motor loads.

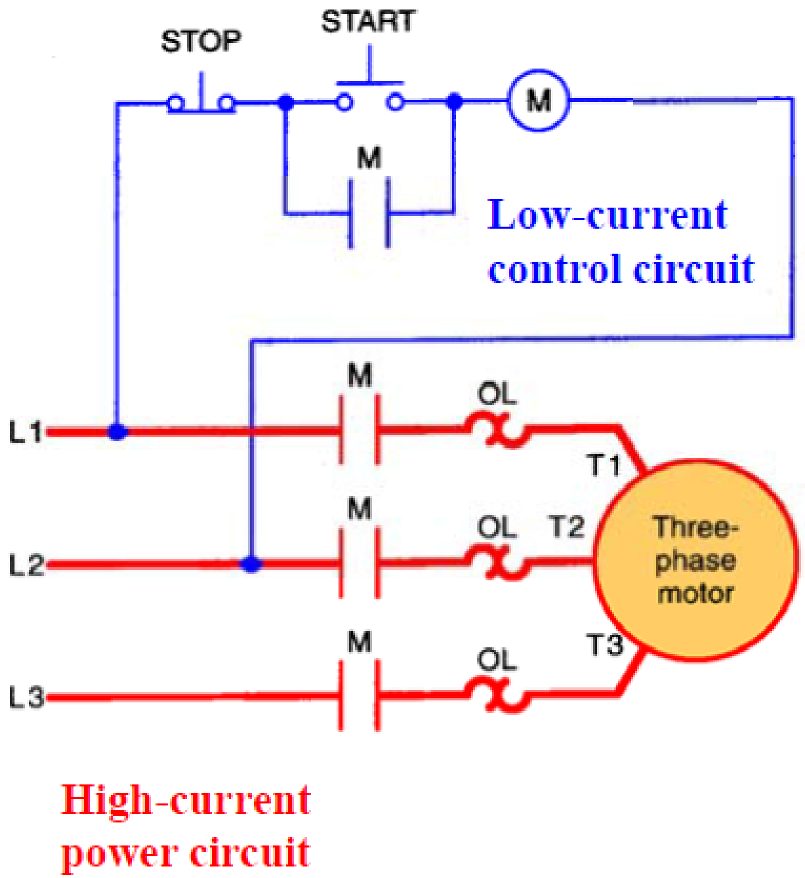
The overload relay will open the supply voltage to the starter if it detects an overload on a motor. Motor overload relay contacts are normally hardwired in series with the magnetic starter coil.

Magnetic Motor Starter



Overload heaters are connected in series with the contactor. If the motor becomes overloaded they cause a mechanical latch to trip. Tripping this latch opens a set of contacts that are wired in series with the voltage supply and motor.

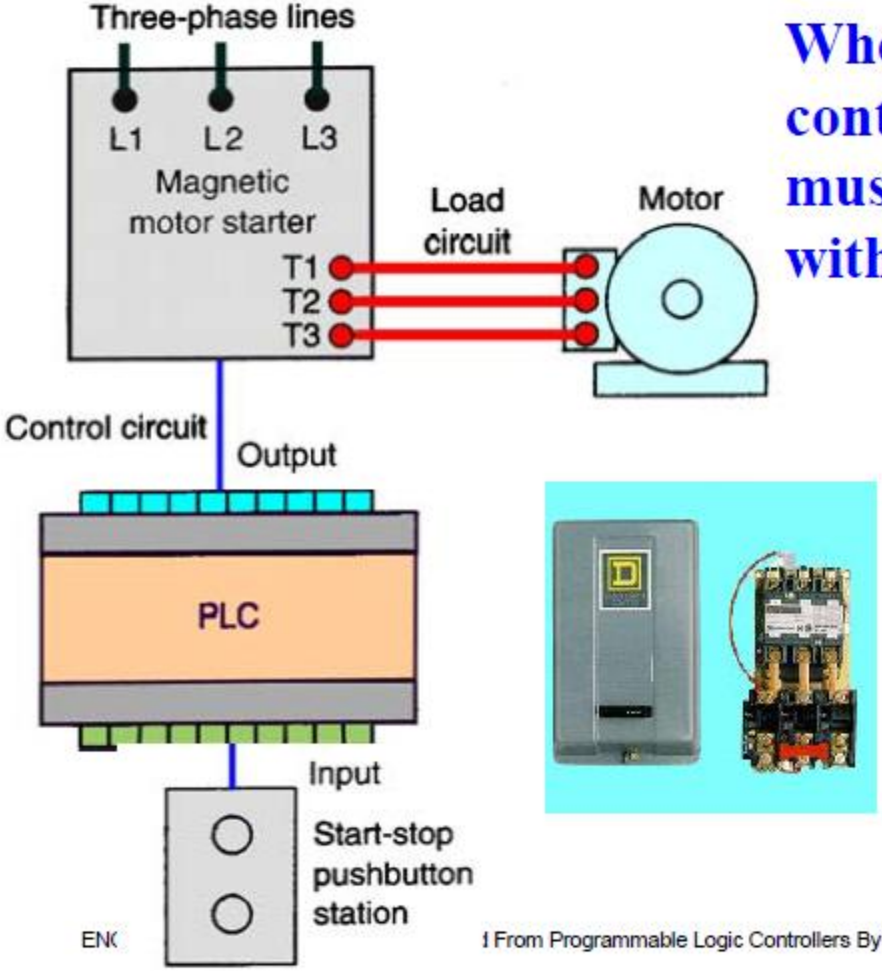
Across the Line AC Starter Operation



- When the start button is pressed, coil M energizes to close all M contacts.
- The OL contact opens automatically when an overload condition is sensed, to de-energize the M coil and stop the motor.

PLC Control of A Large Motor Load

Radu Muresan



When a PLC needs to control a large motor, it must work in conjunction with a starter.

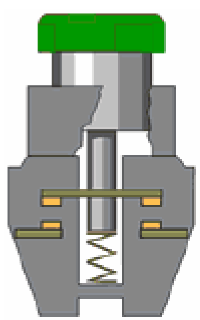
Motor starters are available in various standard National Electric Manufacturers (NEMA) sizes and ratings.

ENC ... d From Programmable Logic Controllers By F. D. Petruzella, McGraw-Hill

Manually Operated Switches

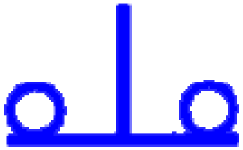
Manually operated switches are controlled by hand. Pushbutton switches are the most common form of manual control found in industry.

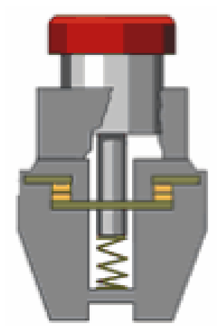
Normally Open (NO) pushbutton makes a circuit when it is pressed and returns to its open position when the button is released.



Manually Operated Switches

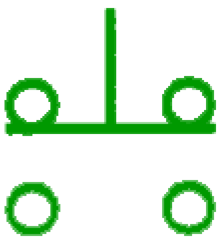
Radu Muresan



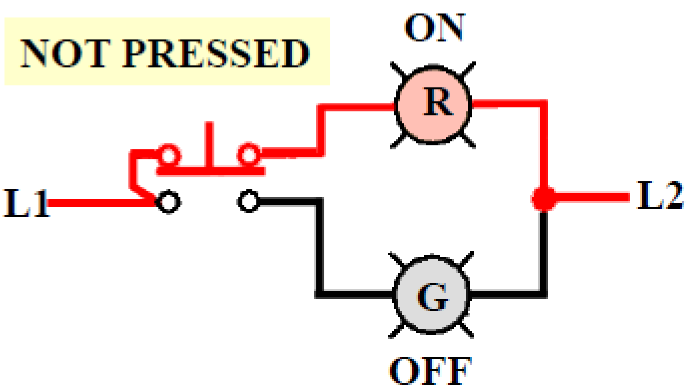


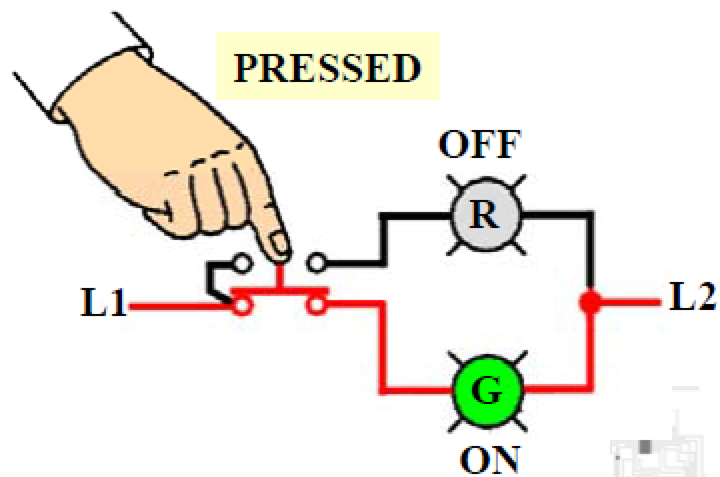
Normally Closed (NC) pushbutton opens the circuit when it is pressed and returns to the closed position when the button is released. The abbreviations NO and NC represent the state of the switch when it is *not* actuated.

Manually Operated Switches

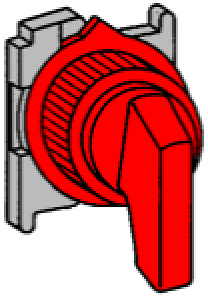


The break-make pushbutton is used for interlocking controls. In this switch the top section is NC, while the bottom section is NO. When the button is pressed, the bottom contacts are closed as the top contacts open.

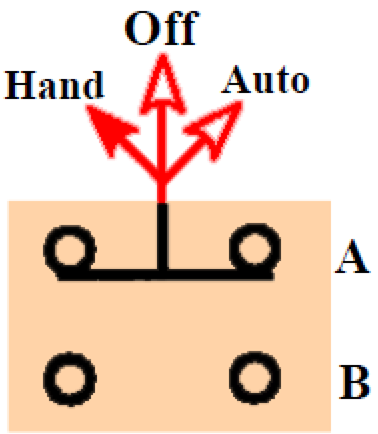




Selector Switch



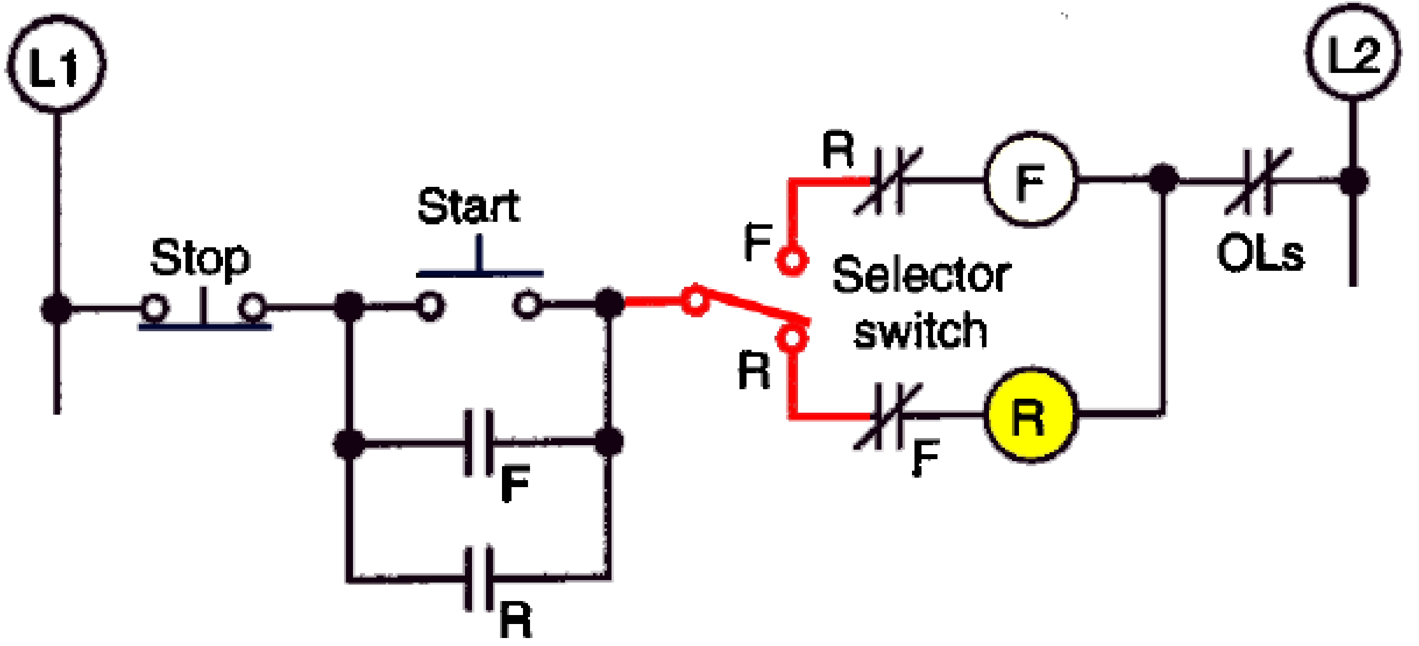
Selector switch positions are made by turning the operator knob – not pushing it.



	Contacts	
Position	A	B
Hand	X	
Off		
Auto		X

Selector switch positions may have two or more selector positions with either maintained contact position or spring return to give momentary contact operation.

Selector Switch Motor Reversing



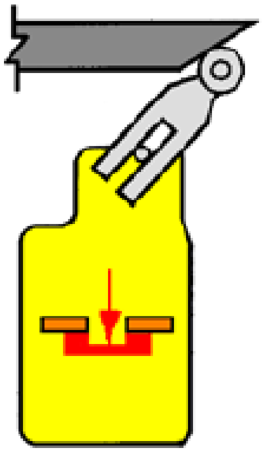
Selector switch used in conjunction with a reversing motor starter to select forward or reverse operation of the motor.

Mechanically Operated Switches

A mechanically operated switch is controlled automatically by factors such as pressure, position, and temperature.

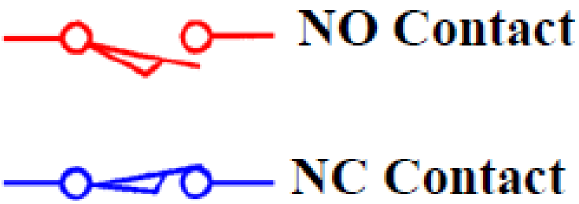
The limit switch is a type of mechanically operated switch designed to operate only when a predetermined limit is reached, and is usually actuated by contact with an object such as a cam.

Limit Switch Operation



Limit switches take the place of a human operator.

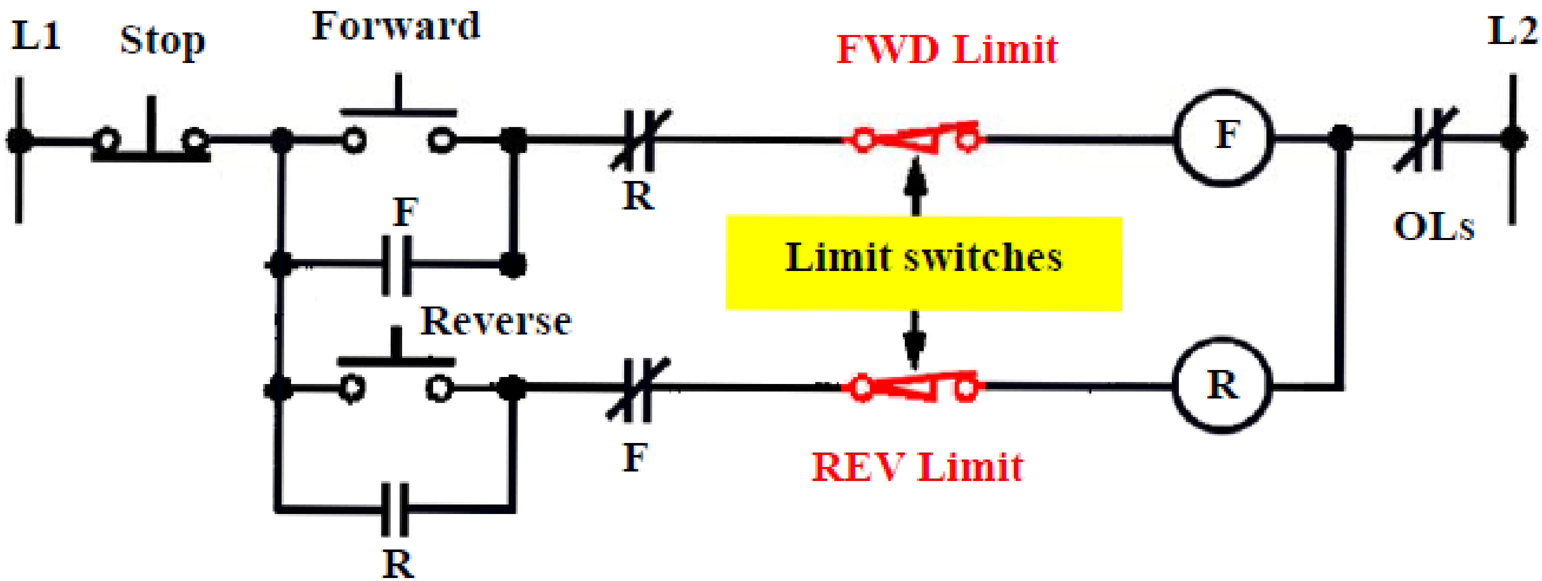
Symbols



They are often used in the control of machine processes to govern the starting, stopping, or reversal of motors.

Typical Limit Switch Circuit

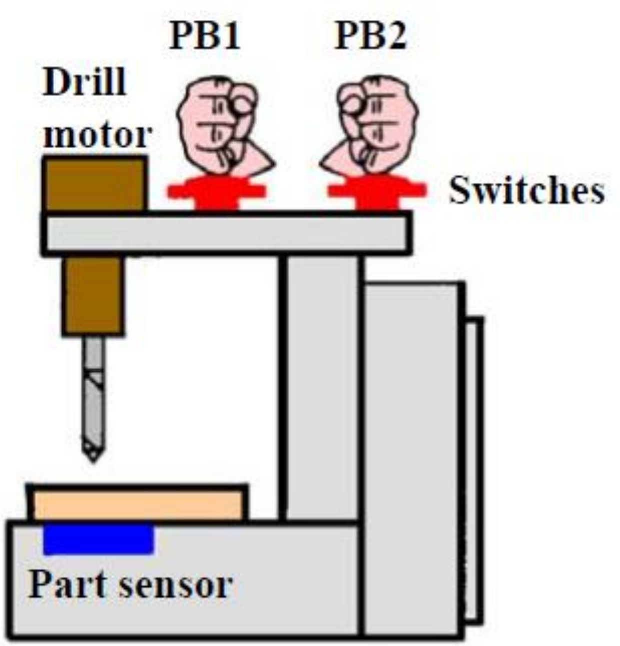
Radu Muresan



Control circuit for starting and stopping a motor in forward and reverse with limit switches providing over travel protection.

Writing a Program from a Narrative Description

Description :
A simple drilling operation requires the drill press to turn on only if there is a part present and the operator has one hand on each of the start Push buttons **This precaution will ensure that the operator's hands are not in the way of the drill.**



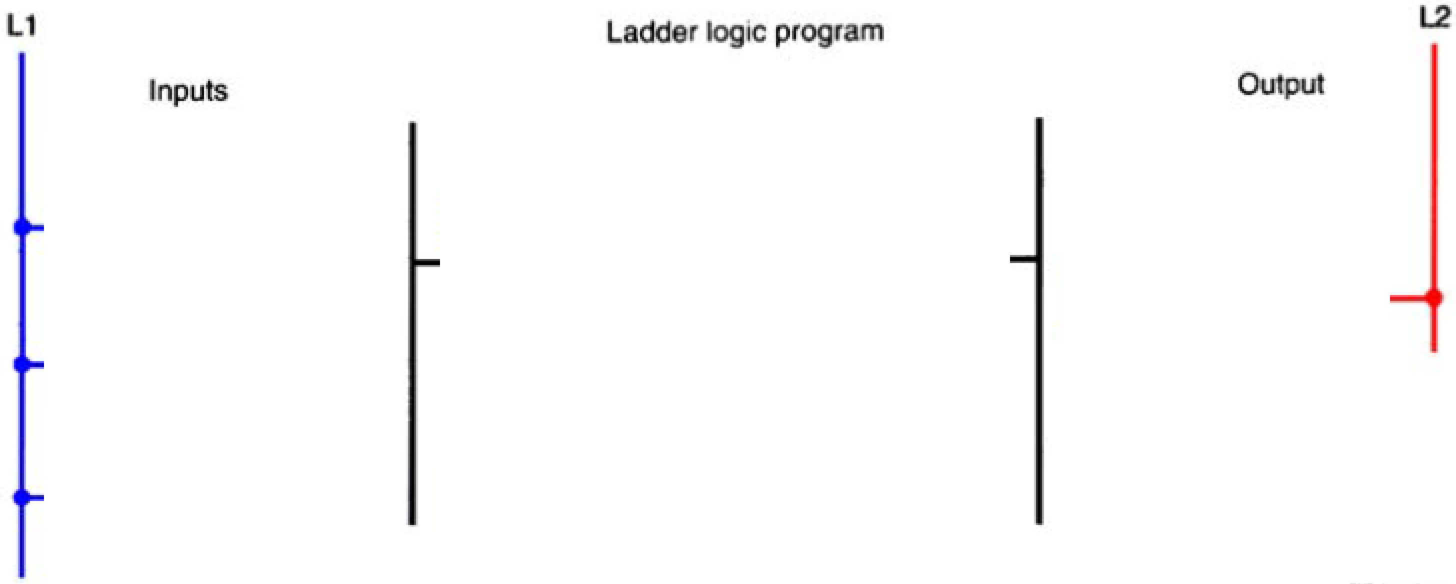
Solution to be added later

.

Writing a Program from a Narrative Description

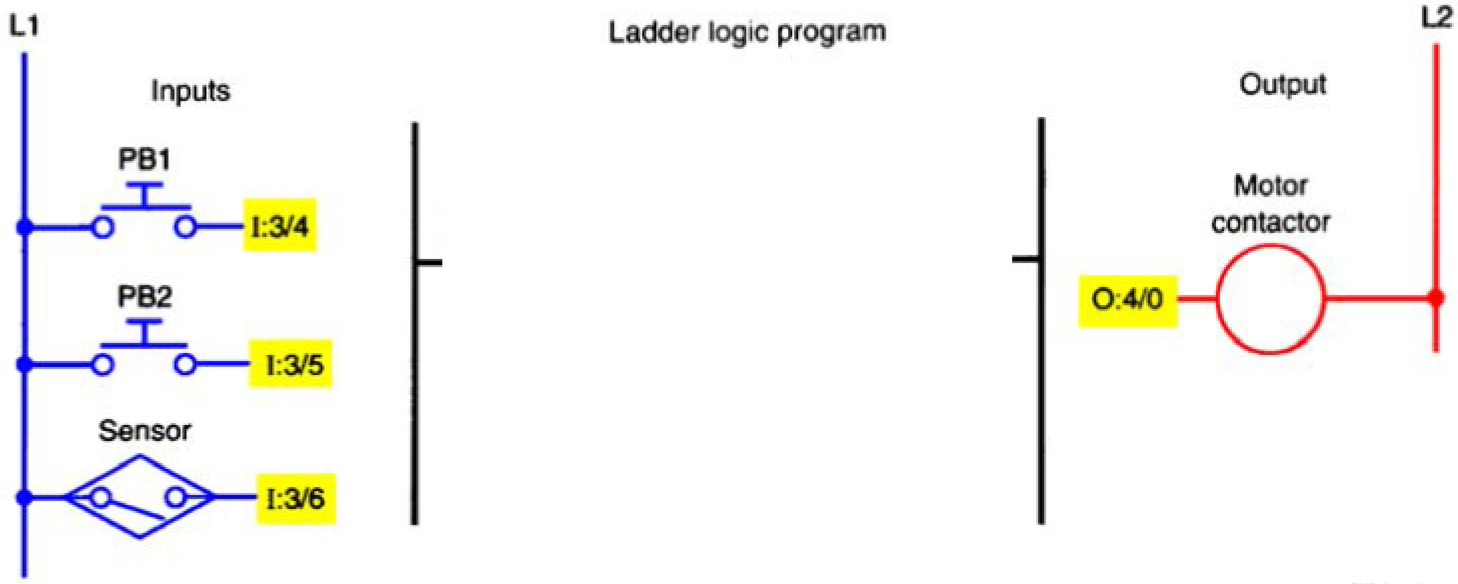
Description :

A simple drilling operation requires the drill press to turn on only if there is a part present and the operator has one hand on each of the start switches. This precaution will ensure that the operator's hands are not in the way of the drill.



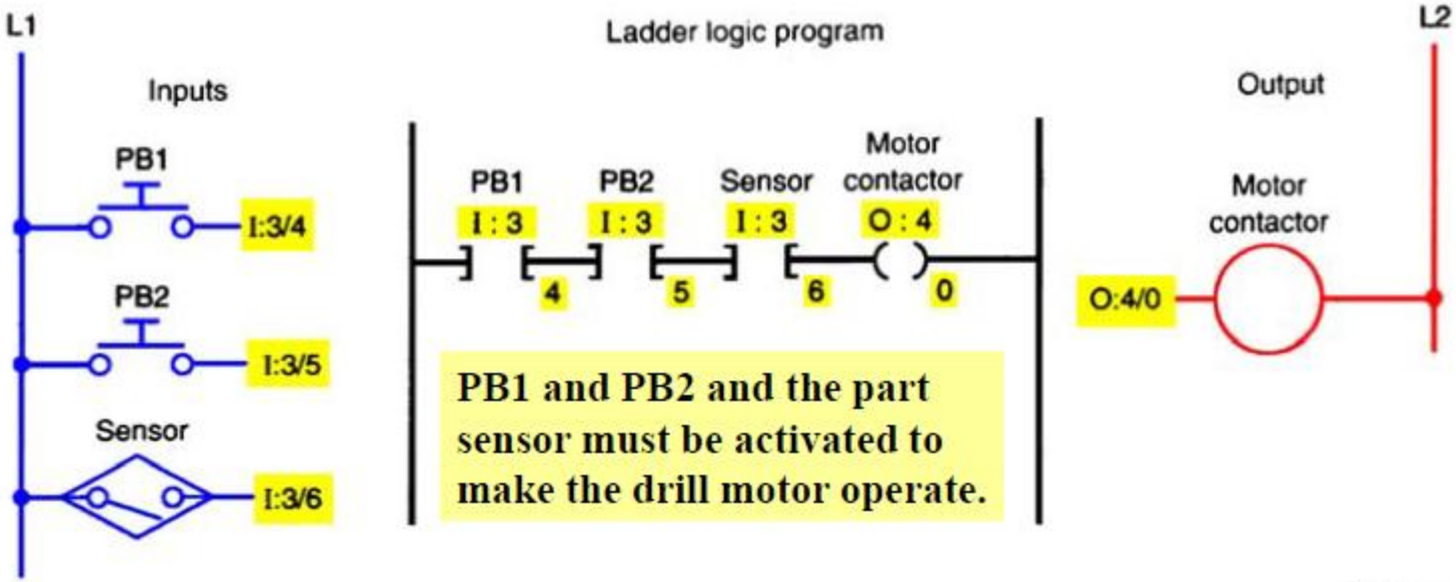
Writing a Program from a Narrative Description

Description :
A simple drilling operation requires the drill press to turn on only if there is a part present and the operator has one hand on each of the start switches. This precaution will ensure that the operator's hands are not in the way of the drill.



Writing a Program from a Narrative Description

Description :
A simple drilling operation requires the drill press to turn on only if there is a part present and the operator has one hand on each of the start switches. This precaution will ensure that the operator's hands are not in the way of the drill.



Example

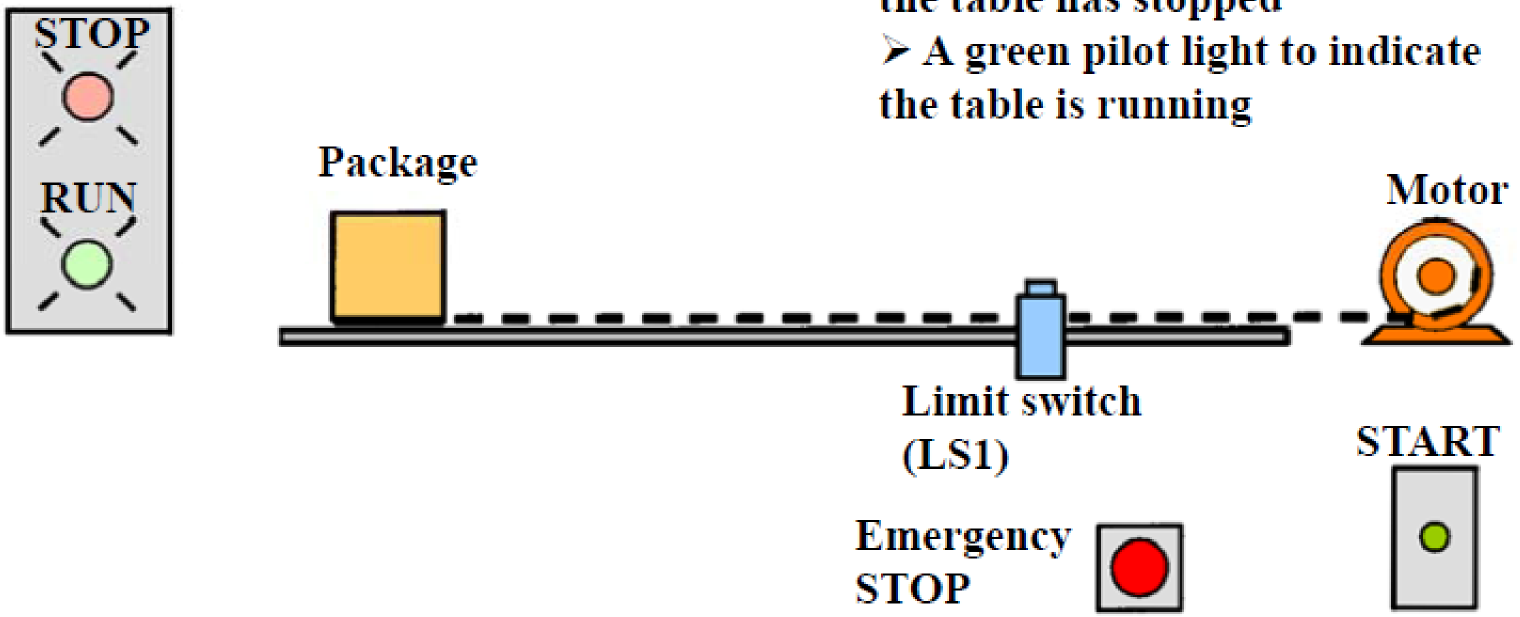
Hardwired Sequential Process

The sequential task is as follows:

1. Start button is pressed.
2. Table motor is started.
3. Package moves to the limit switch and stops.

Auxiliary Features:

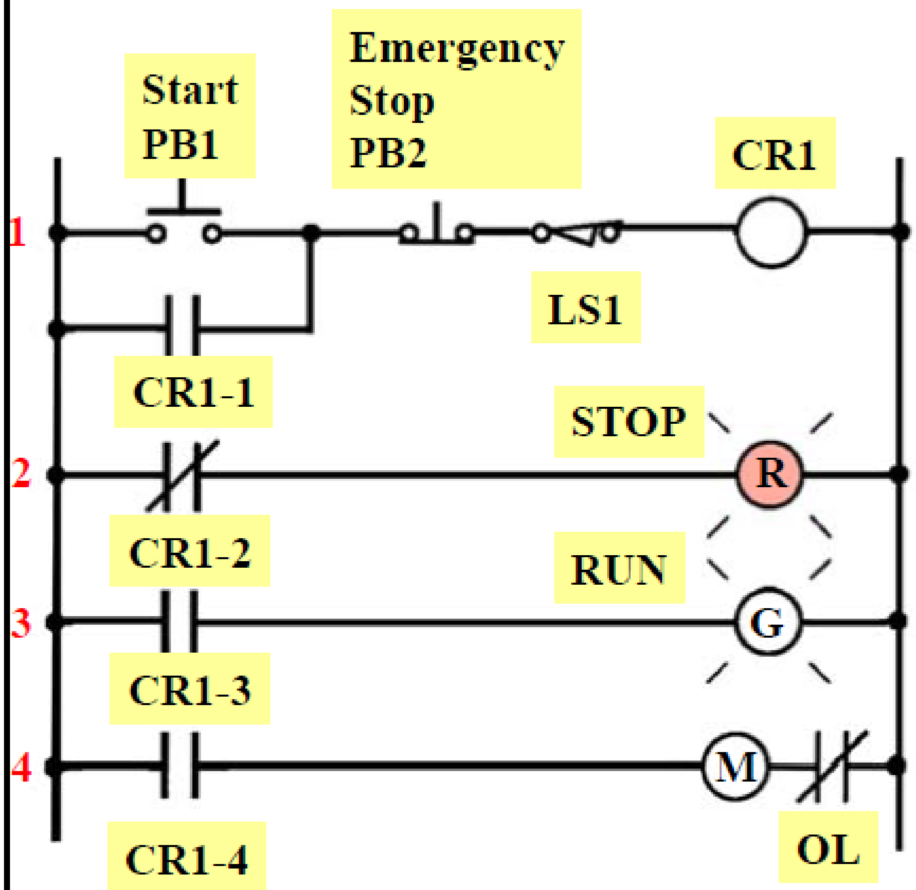
- An emergency stop button that will stop the table, for any reason, before the package reaches the limit switch position
- A red pilot light to indicate the table has stopped
- A green pilot light to indicate the table is running



Solution to be added later

.

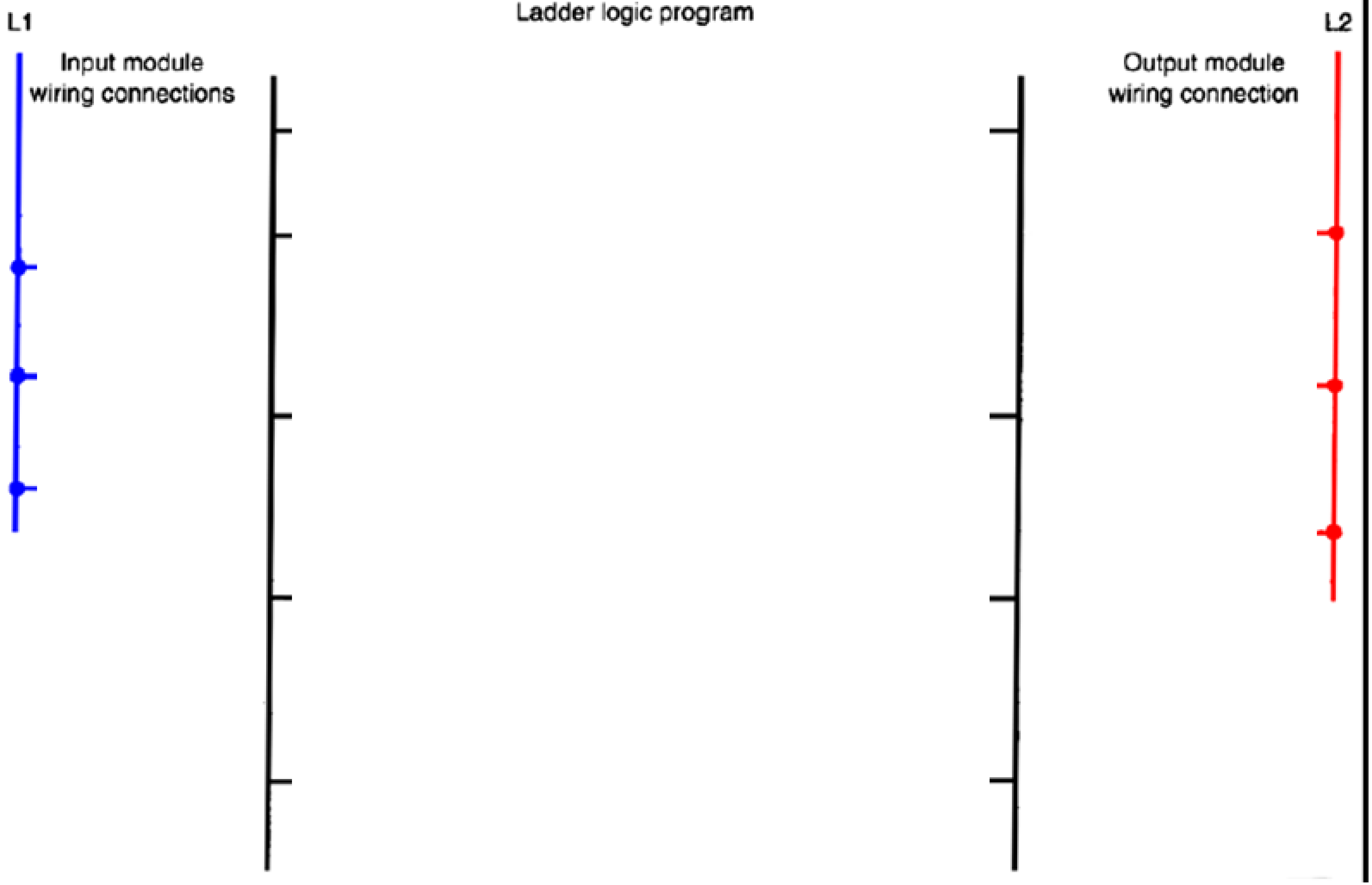
Hardwired Sequential Process



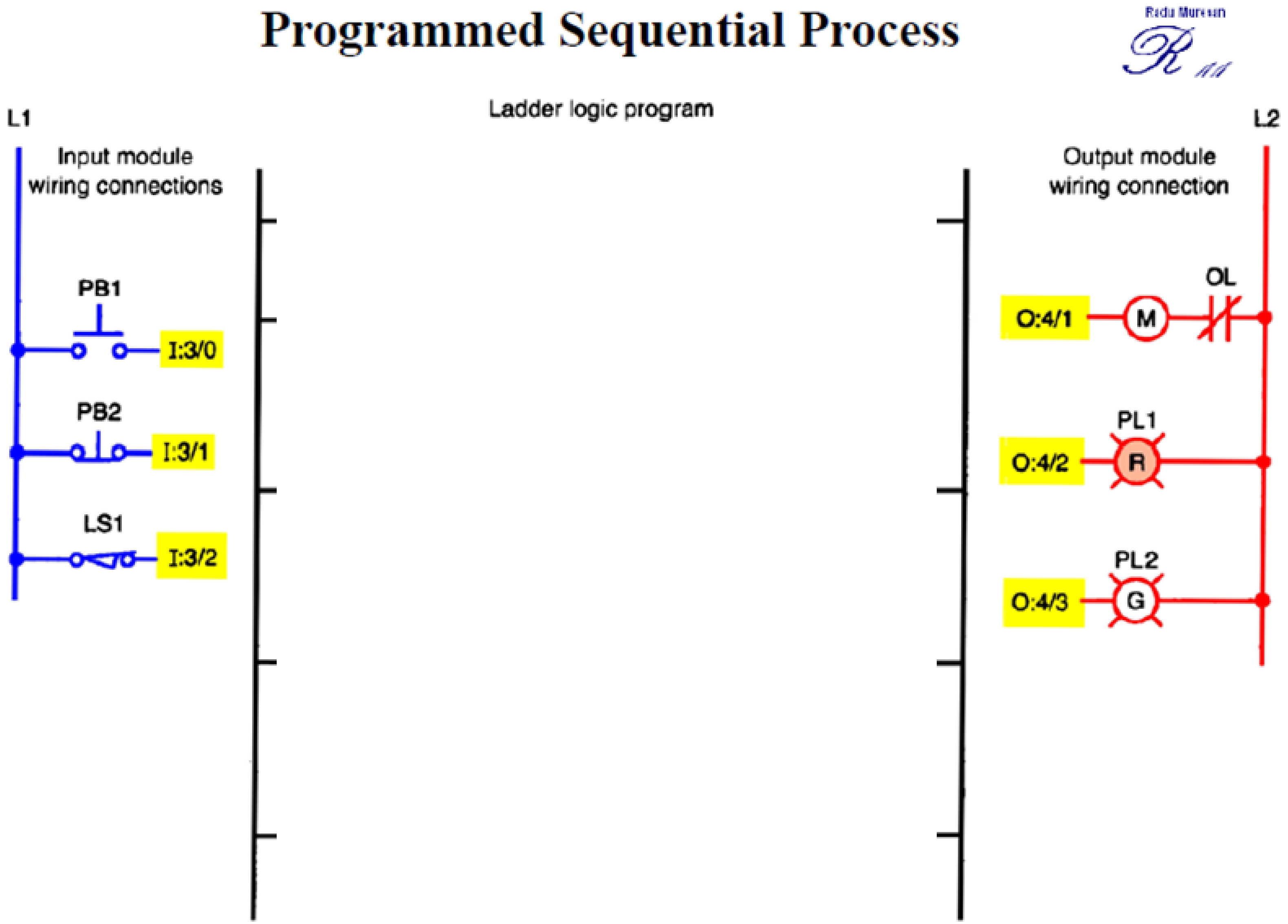
Summary of the control task:

- Start button is actuated.
- CR1-1 closes to seal in CR1
- CR1-2 opens, switching the red stop pilot light off
- CR1-3 closes, switching the green run pilot light on
- CR1-4 closes to energize the motor starter and motor
- The package moves to the limit switch to actuate it and de-energize coil CR1
- CR1-1 opens to open the seal-in contact
- CR1-2 closes, switching the red pilot light on
- CR1-3 opens, switching the green pilot light off
- CR1-4 opens to de-energize the starter coil, stop the motor, and end the sequence

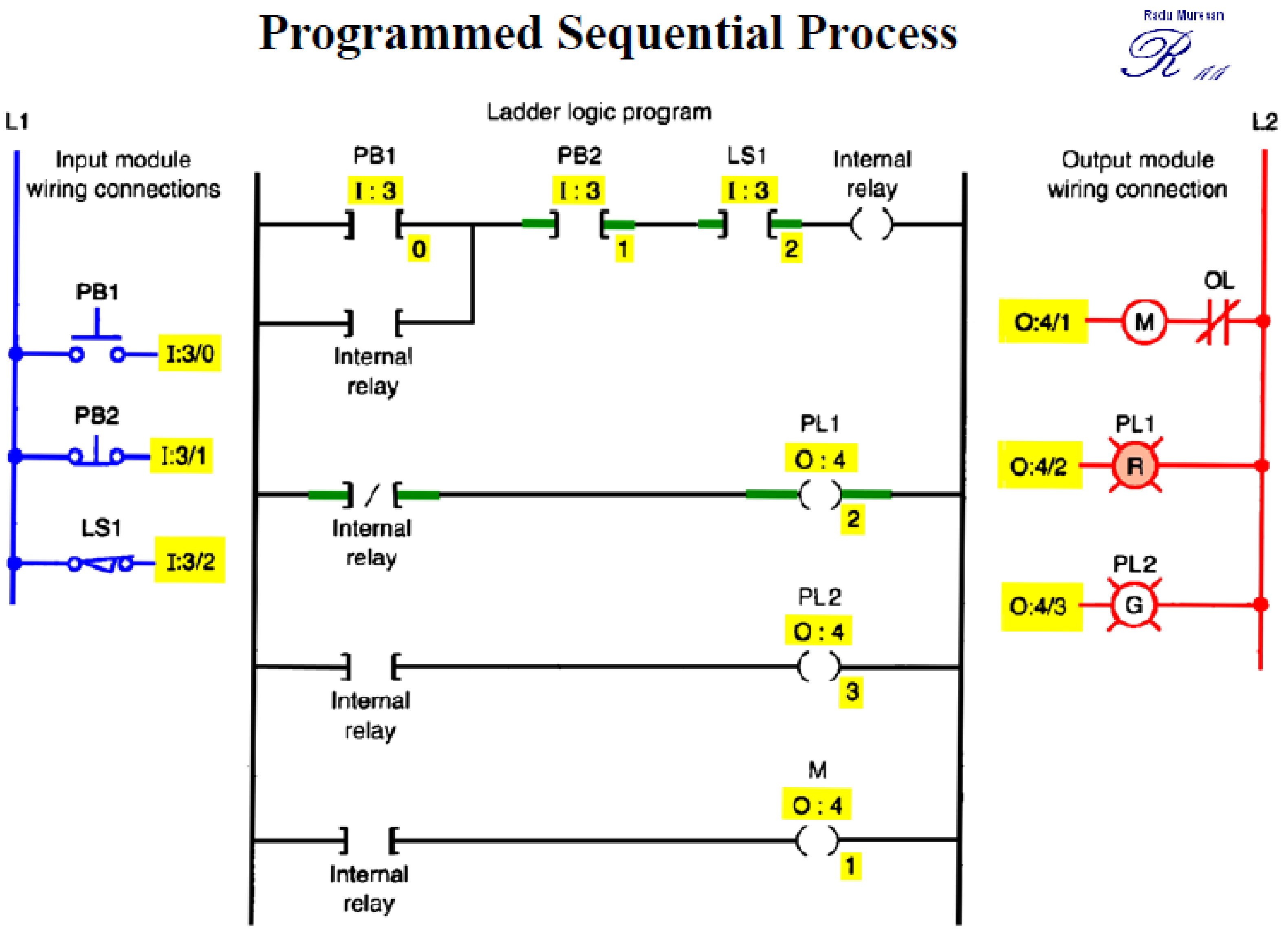
Programmed Sequential Process



Programmed Sequential Process



Programmed Sequential Process



Temperature Switch

The temperature switch or thermostat is used to sense temperature changes and is actuated by some specific environmental temperature change.

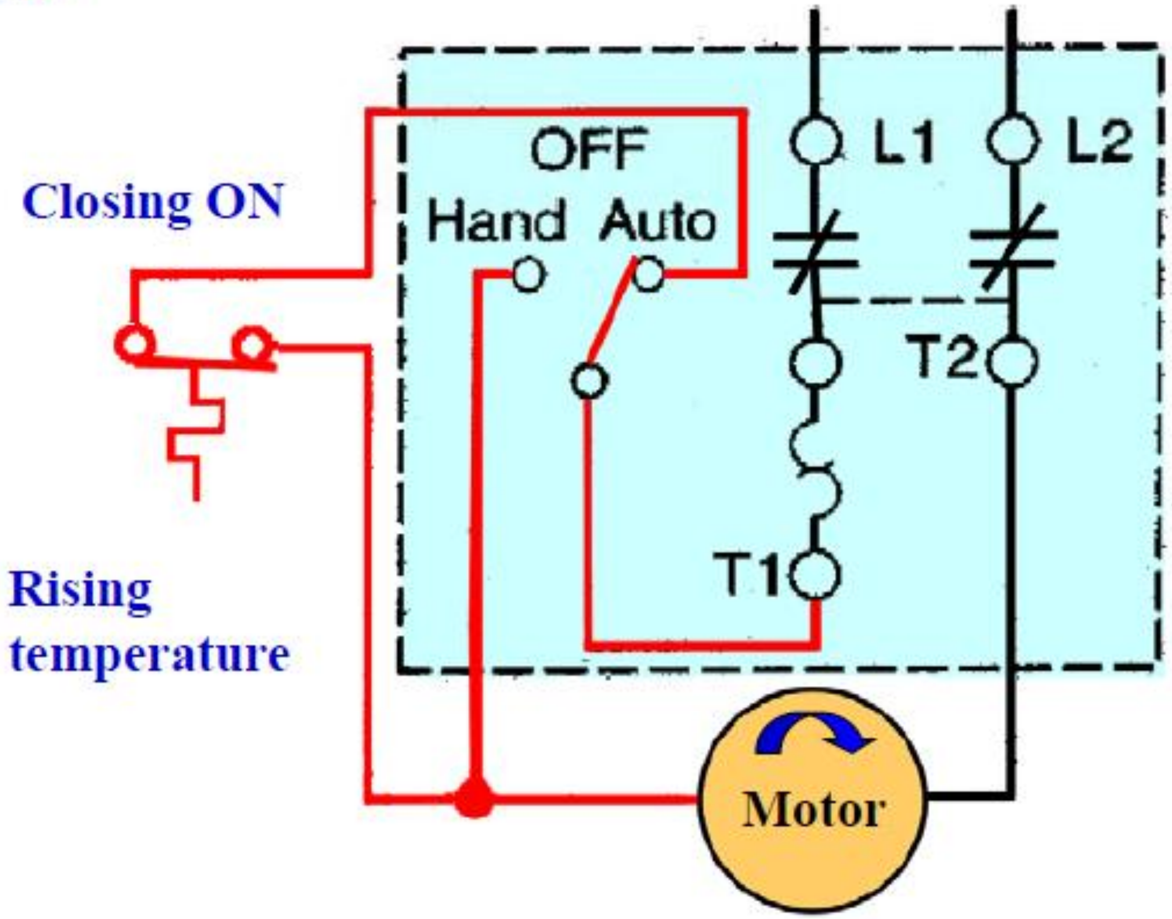
Responds to changes in temperature by opening or closing an electric circuit.

Symbols

NO Contact

NC Contact

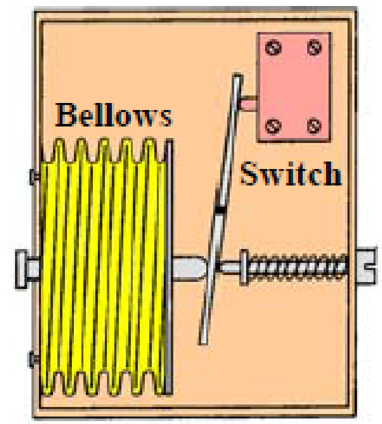
Temperature Switch Control of a Motor



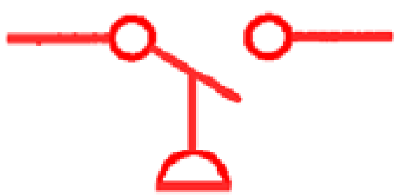
Pressure Switch

Pressure switches are used to control the pressure of liquids and gases and are activated when a specific pressure is reached.

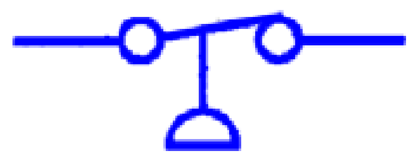
Opens or closes an electric circuit in response to a change in pressure.



Symbols

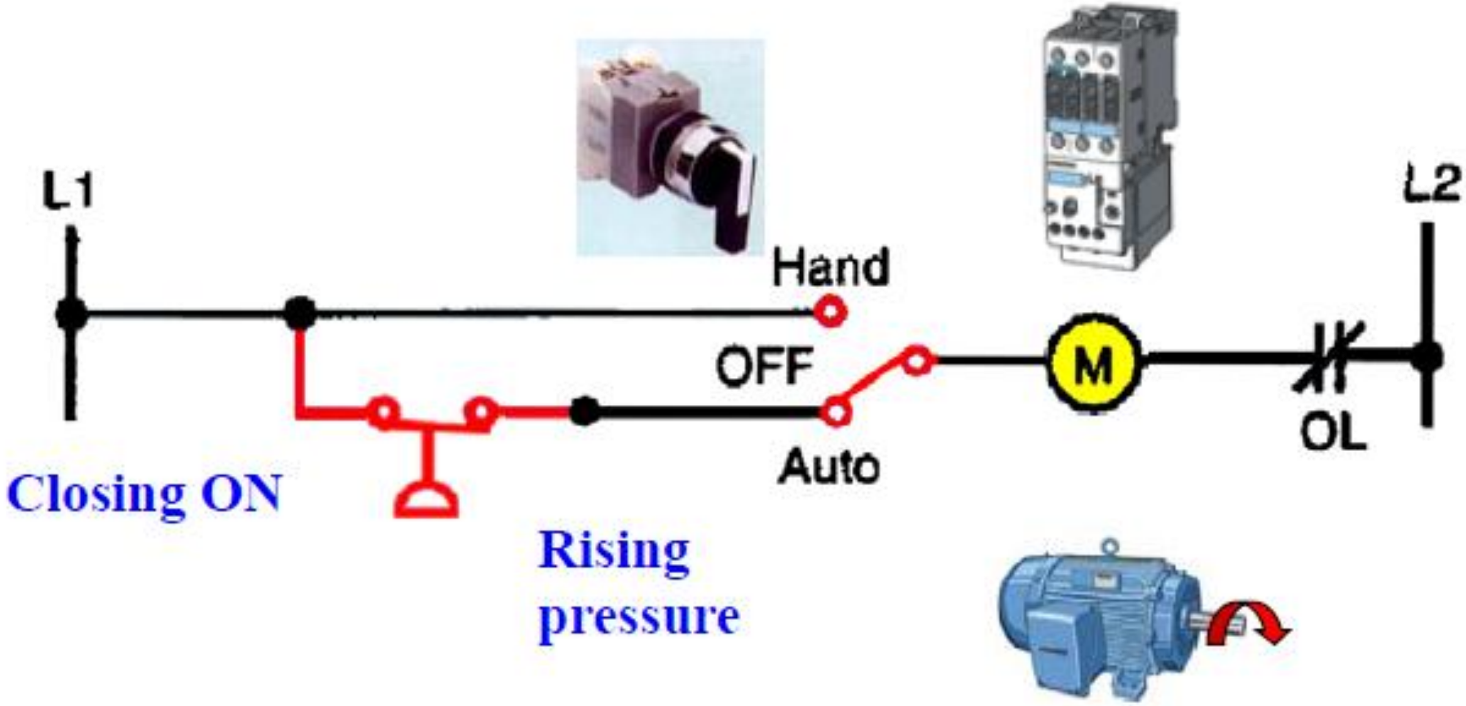


NO Contact



NC Contact

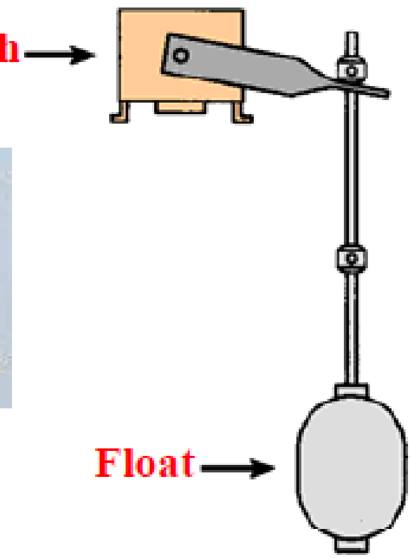
Starter Operated By a Pressure Switch



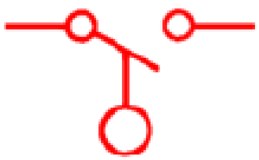
Level Switch

Level or float switches are used to sense the height of a liquid.

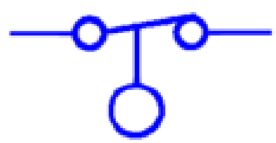
Opens or closes an electric circuit in response to a change in liquid level.



Symbols

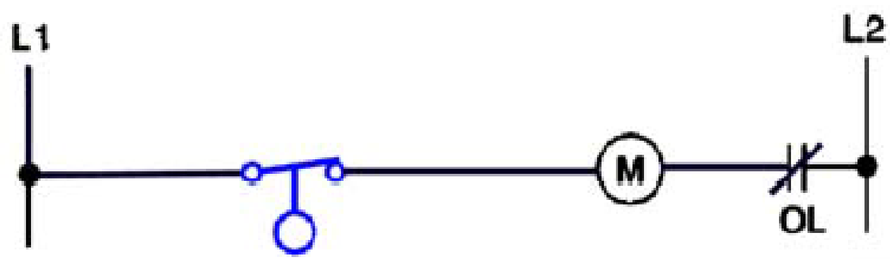


NO Contact



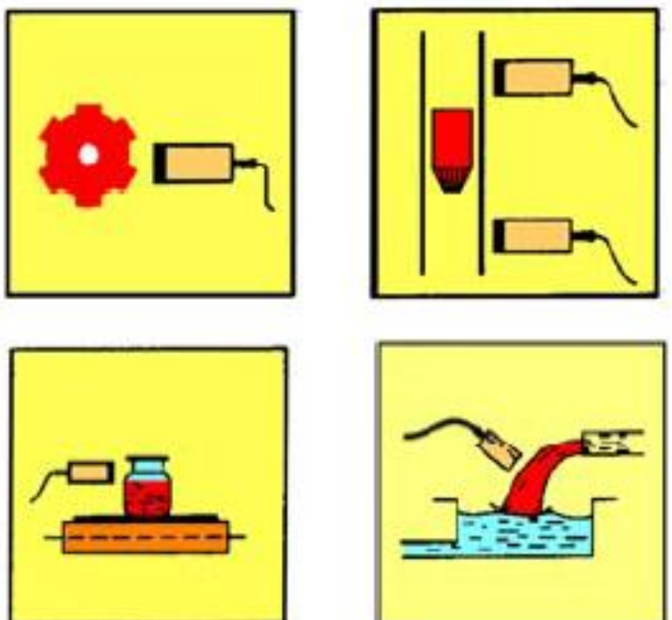
NC Contact

Two-wire level switch control of starter.



Proximity Sensors

Proximity sensors or switches detect the presence of an object without making physical contact with it.



Proximity Sensors Applications

Radu Muresan

The object being detected is too small, lightweight, or soft to operate a mechanical switch.

Rapid response and high switching rates are required.

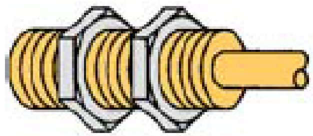
An object has to be sensed through nonmetallic barriers such as glass, plastic, and paper cartons.

Hostile environments conditions exist.

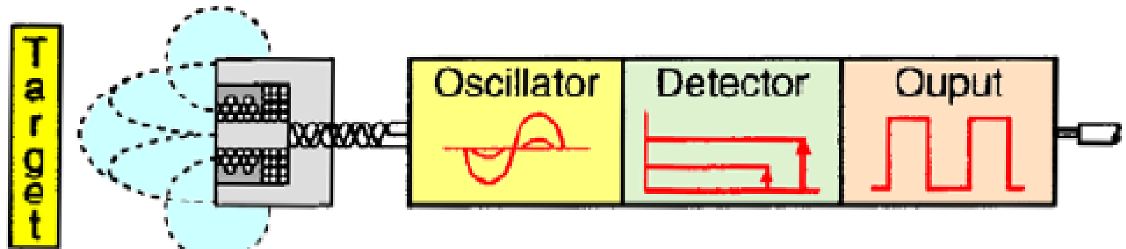
Long life and reliable service are required.

A fast electronic control system requires a bounce-free input signal.

Inductive Proximity Sensor Operation

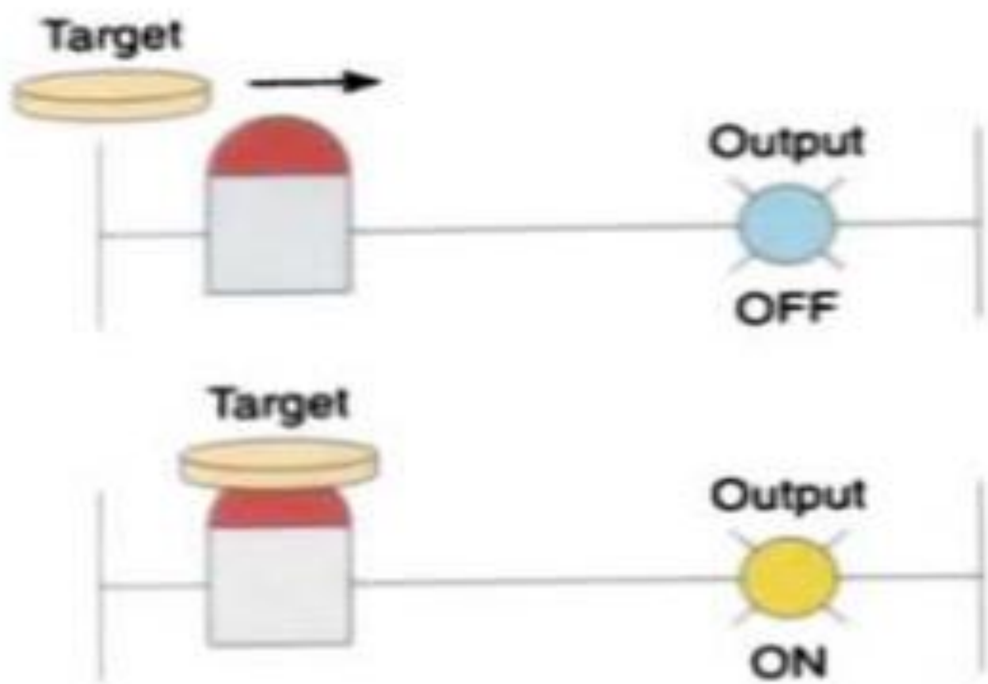


Barrel type



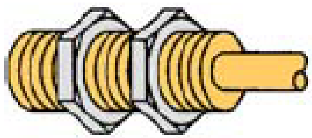
Block diagram

Metal target

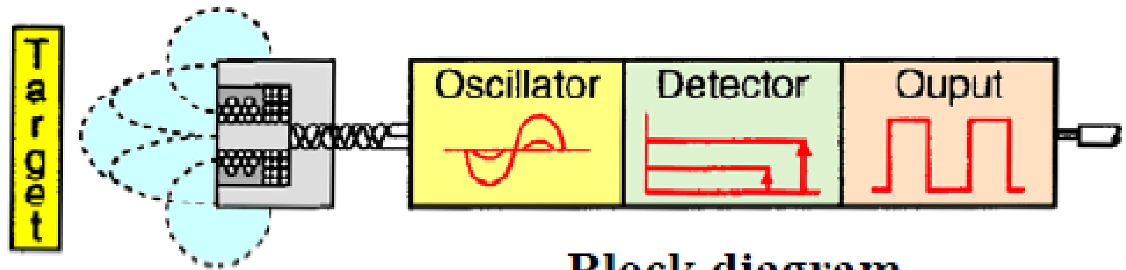


As the target moves into the sensing area, the sensor switches the output ON

Inductive Proximity Sensor Operation



Barrel type



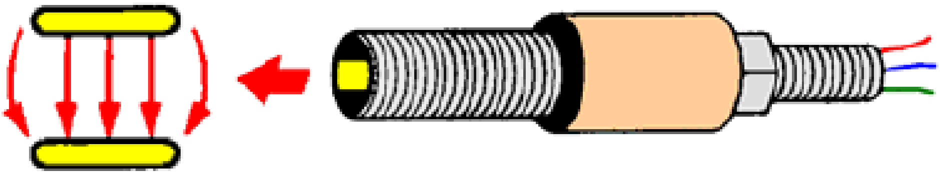
Block diagram

Metal target

In principle, an *inductive sensor* consists of a coil, oscillator, detector circuit, and solid-state output (Fig. 6-20 on page 141). When energy is supplied, the oscillator operates to generate a high-frequency field. At this moment, there must not be any conductive material in the high-frequency field. When a metal object enters the high-frequency field, eddy currents are induced in the surface of the target. These currents result in a loss of energy in the oscillator circuit, which in turn causes a smaller amplitude of oscillation. The detector circuit recognizes a specific change in amplitude and generates a signal that will turn the solid-state output on or off. When the metal object leaves the sensing area, the oscillator regenerates, allowing the sensor to return to its normal state.

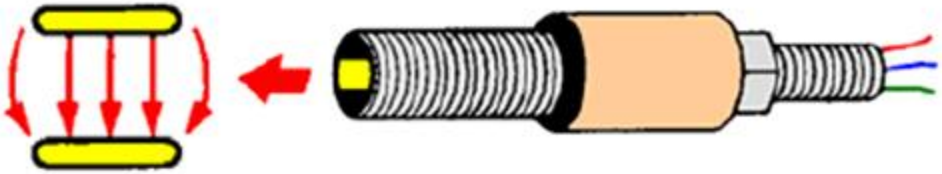
Capacitive Proximity Sensor

A capacitive proximity sensor can be actuated by both conductive and nonconductive material such as wood, plastics, liquids, sugar flour and wheat.



Operation is similar to that of inductive proximity sensor. Instead of a coil, the active face of the sensor is formed by two metallic electrodes – rather like an "opened capacitor".

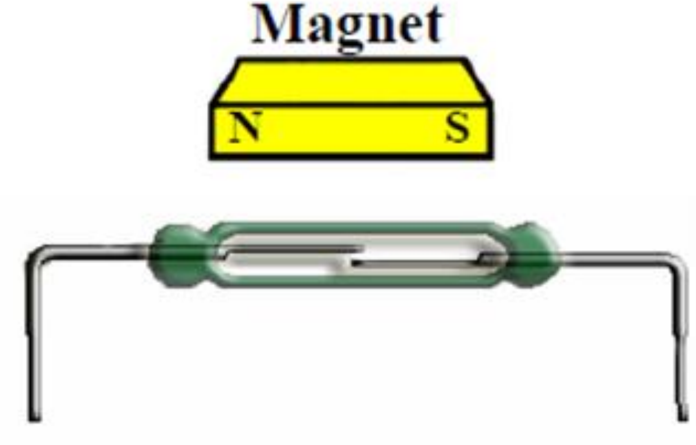
Capacitive Proximity Sensor



A *capacitive proximity sensor* is a sensing device actuated by conductive and nonconductive materials. The operation of *capacitive sensors* is also based on the principle of an oscillator. Instead of a coil, however, the active face of a capacitive sensor is formed by two metallic electrodes—rather like an "opened" capacitor. The electrodes (Fig. 6-24*a* on page 142) are placed in the feedback loop of a high-frequency oscillator that is inactive with "*no* target present." As the target approaches the face of the sensor, it enters the electrostatic field formed by the electrodes. This approach causes an increase in the coupling capacitance, and the circuit begins to oscillate. The amplitude of these oscillations is measured by an evaluating circuit that generates a signal to turn the solid-state output on or off.

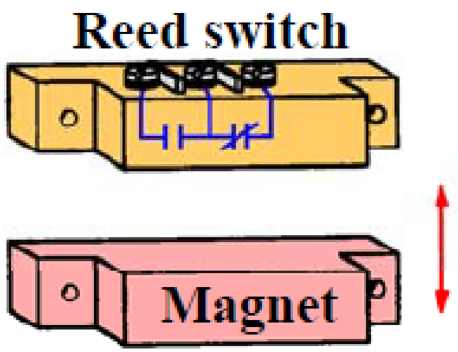
Magnetic Switch (Reed Switch)

A magnetic switch (also called a reed switch) is composed of flat contact tabs that are hermetically sealed (air-tight).

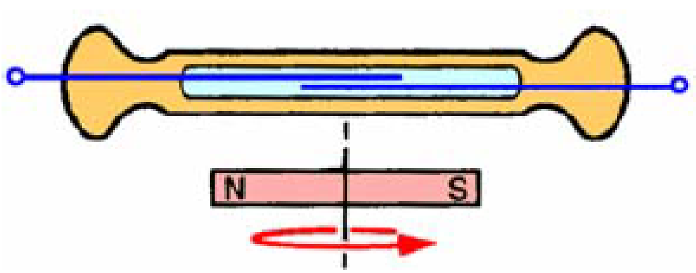


The switch is actuated by a magnet.

Reed Switch Activation



Proximity motion – movement of the switch or magnet will activate the switch

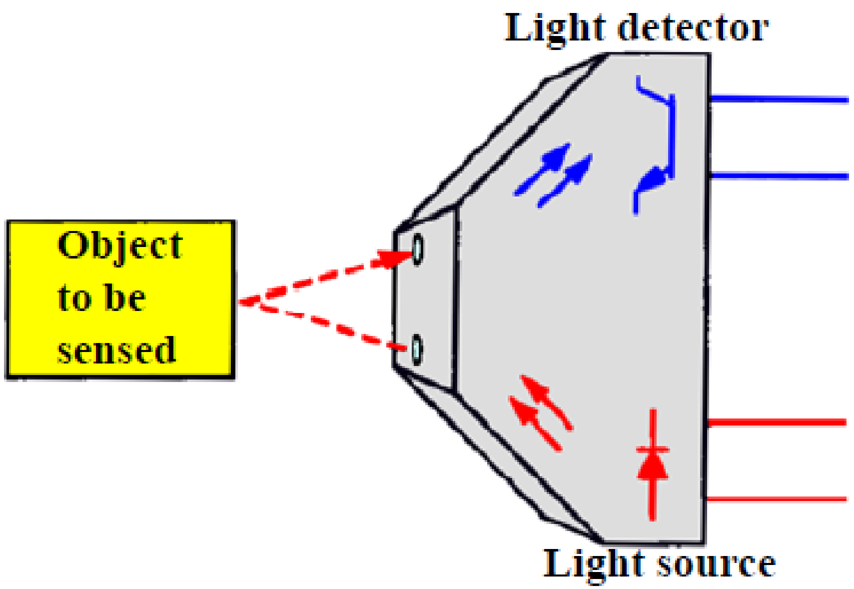


Rotary motion – switch is actuated twice for every complete revolution

Shielding – the shield short circuits the magnetic field; switch is activated by removal of the shield

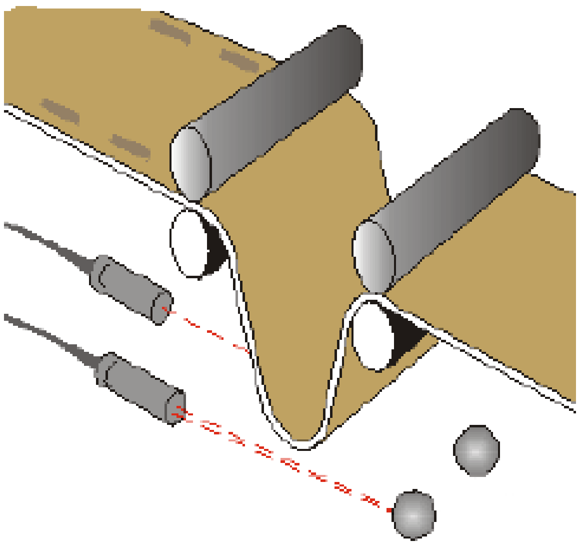
Photoelectric Sensor Operation

Most industrial photoelectric sensors use a light-emitting diode (LED) for the light source and a phototransistor to sense the presence or absence of light.



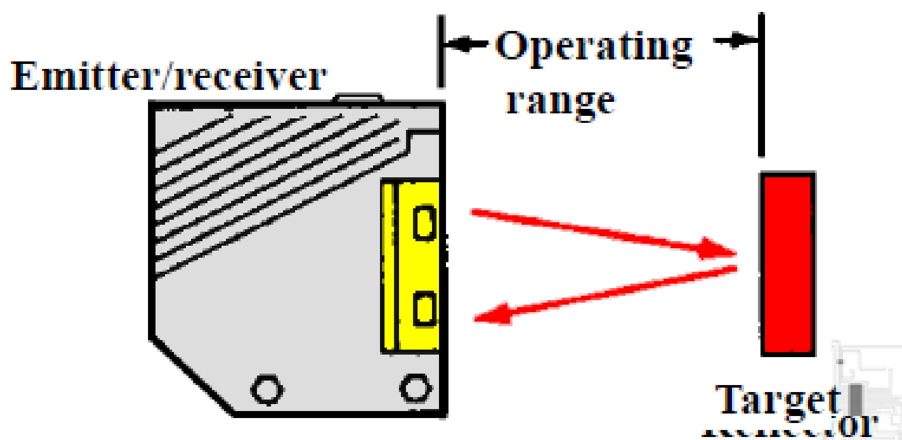
Light from the LED falls on the input of the phototransistor and the amount of conduction through the transistor changes. Analog outputs provide an output proportional to the quantity of light seen by the photodetector.

Reflective Photoelectric Sensor

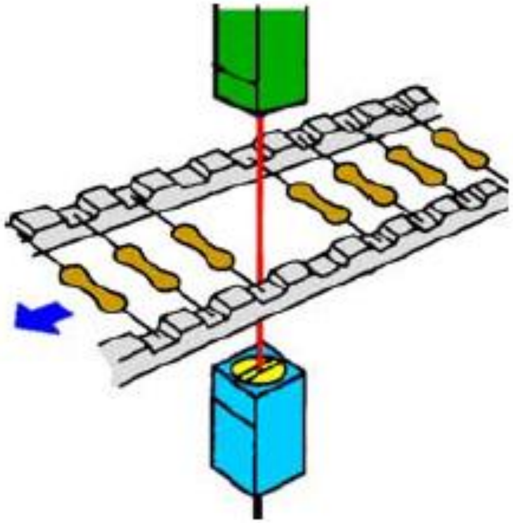


Emits a light beam (visible, infrared, or laser) from its light emitting element and detects the light being reflected.

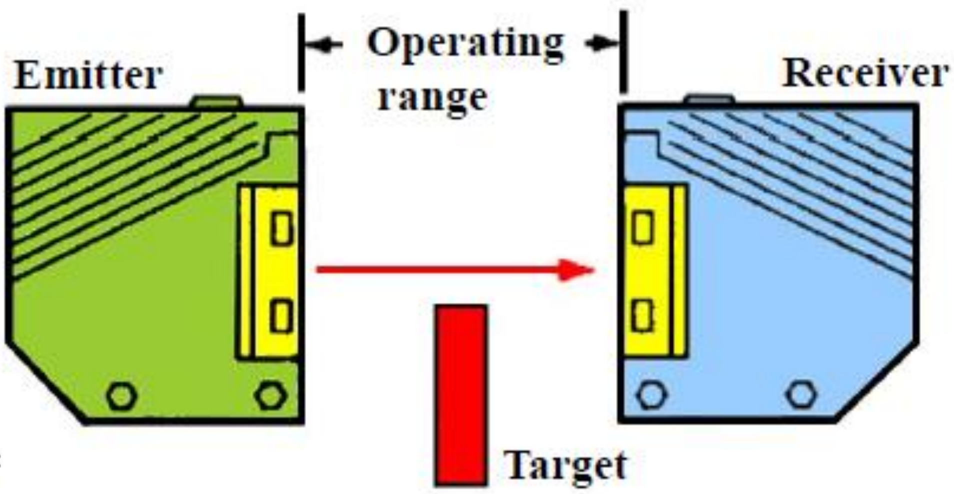
Diffused-reflective type



Through-Beam Type Photoelectric Sensor



A through-beam photoelectric sensor is used to measure the change in light quantity caused by the target's crossing the optical axis.



Output Control Devices

A variety of output control devices can be operated by the controller output module to control traditional processes. These include:


Pilot light


Control relay


Solenoid


Alarm


Solenoid valve


Heater


Motor starter

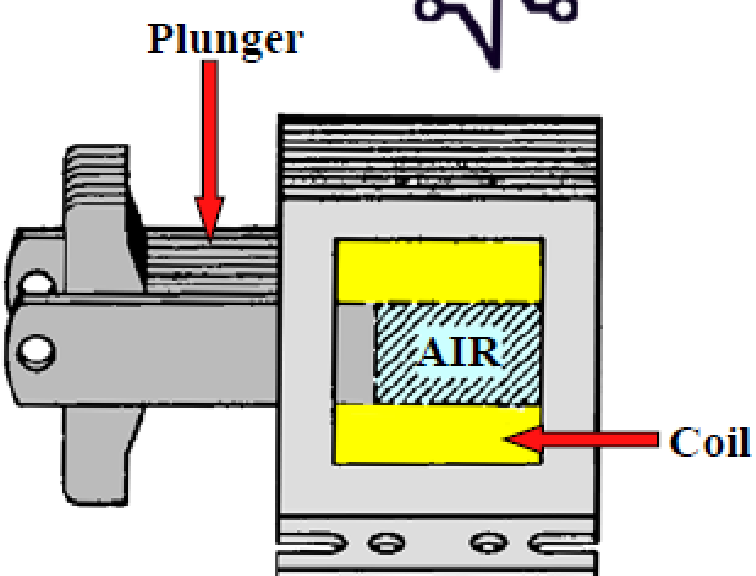

Small motor

Actuator

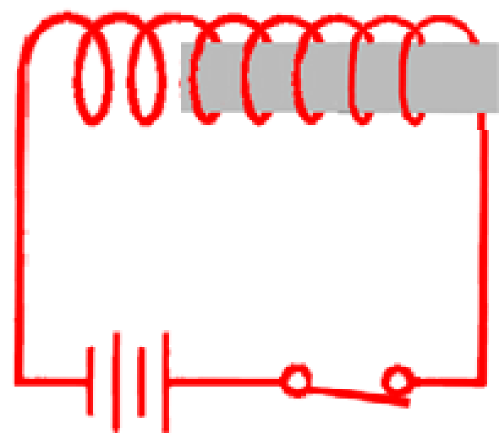
An actuator is any device that converts an electrical signal into mechanical movement. The principle types of actuators are relays, solenoids, and motors.

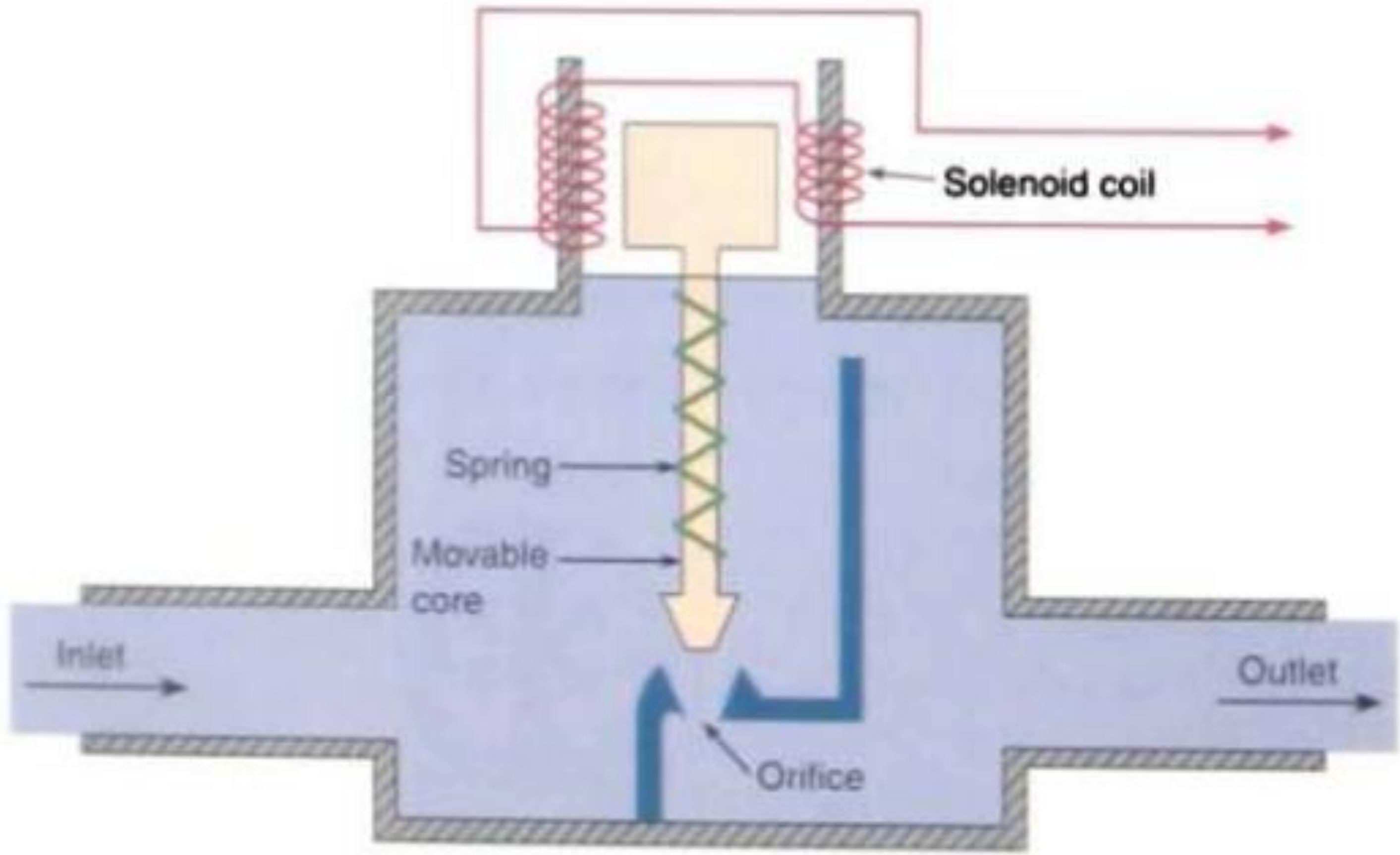
Solenoid

Symbol



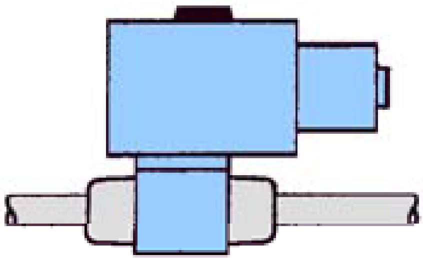
The solenoid converts electric current into linear motion.





(*a*) Operation

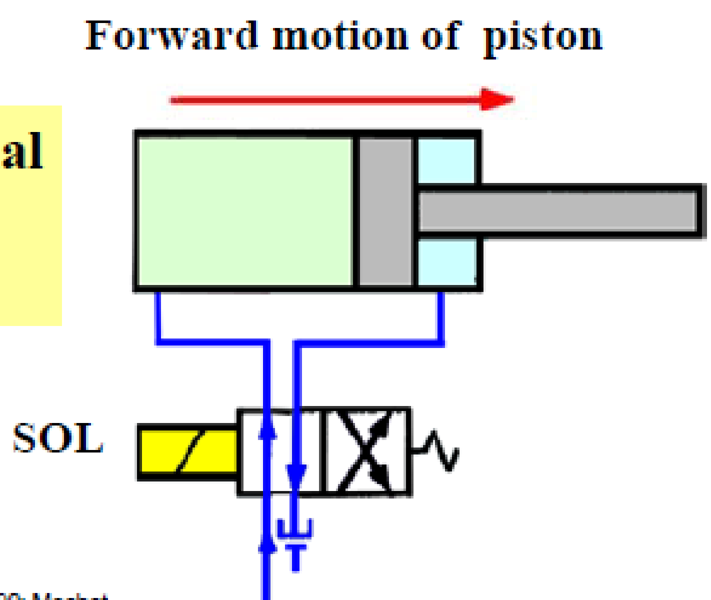
Solenoid Valve



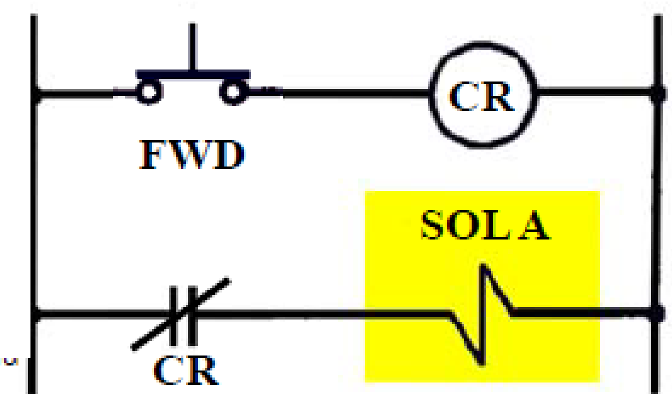
A solenoid valve is a combination of:

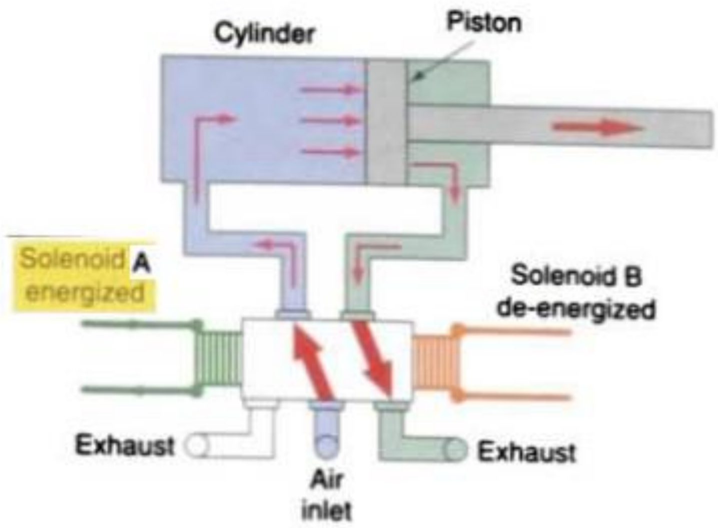
- a solenoid with its core or plunger
- a valve body containing an orifice in which a disc or plug is positioned to restrict or allow flow

Directional solenoid valve

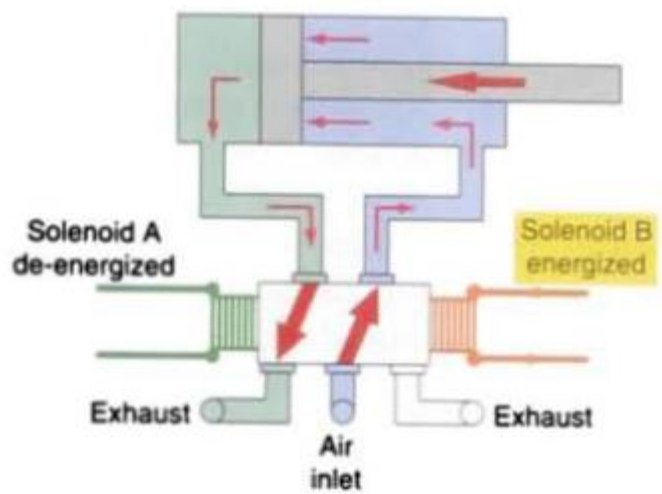


When SOL A is energized, the valve spool is shifted to redirect the fluid and move the cylinder forward





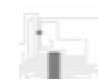
(*b*) Pneumatic cylinder extends when solenoid A is energized and solenoid B is de-energized.



(*c*) Energizing solenoid B reverses the pressure and exhaust valve positions as shown; cylinder retracts.

Example: Motorized Overhead Garage Door

- A motorized overhead garage door is to be operated automatically to preset open and closed positions.
- Devices used: see next slides
- Solution: sequence of operations
  - when the up button is pushed, the up motor contactor energizes and the door travels upwards until the up limit switch is actuated;
  - When the down button is pushed, the down motor contactor energizes and the door travels down until the down limit switch is actuated;
  - when the stop button is pushed, the motor stops. The motor must be stopped before it can change direction



Radu Muresan

Example: Motorized Overhead Garage Door

A motorized overhead garage door is to be operated automatically to preset open and closed positions. The field devices include one of each of the following:

- Reversing *motor contactor* for the up and down directions
- Normally *closed down limit switch* to sense when the door is fully closed
- Normally closed *up limit switch* to sense when the door is fully opened
- Normally open *door up button* for the up direction
- Normally open *door down button* for the down direction
- Normally closed *door stop button* for stopping the door
- Red *door ajar light* to signal when the door is partially open
- Green *door open light* to signal when the door is fully open
- Yellow *door closed light* to signal when the door is fully closed

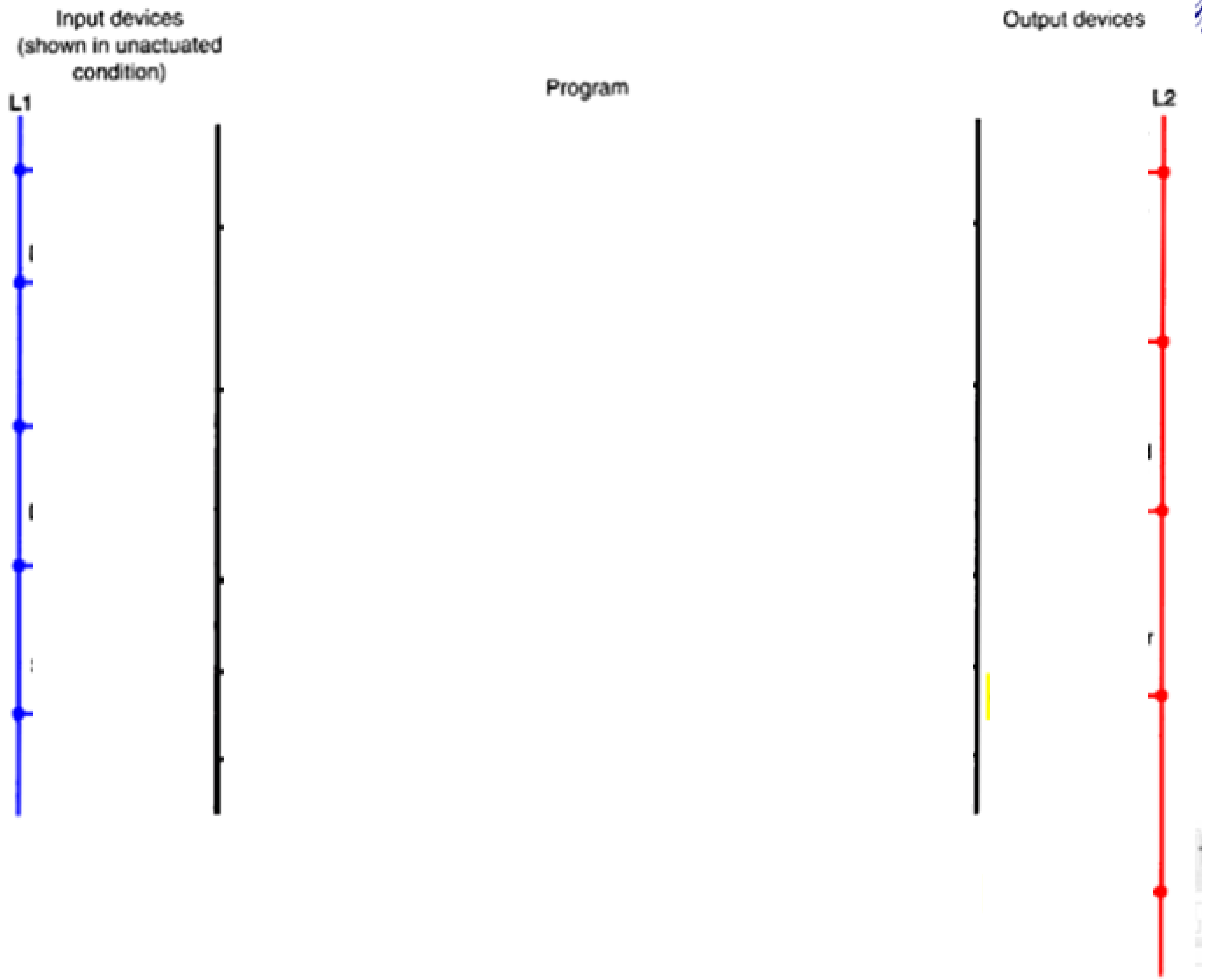
Solution:

- The sequence of operation is as follows:
  - When the up button is pushed, the up motor contactor energizes and the door travels upward until the up limit switch is actuated.
  - When the down button is pushed, the down motor contactor energizes and the door travels down until the down limit switch is actuated.
  - When the stop button is pushed, the motor stops. The motor must be stopped before it can change direction.
- Figure 6-59 on page 162 shows the ladder logic required for the process.

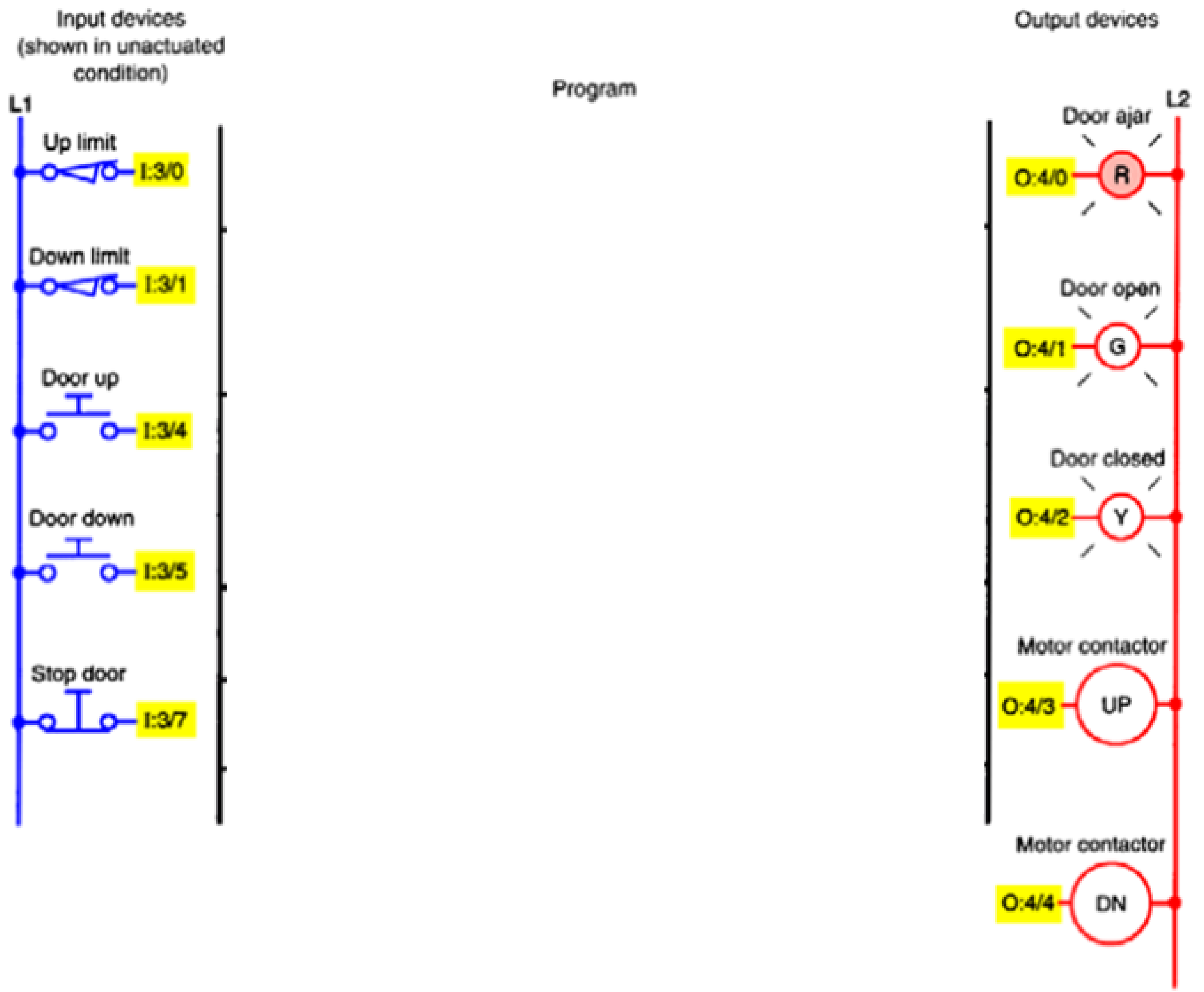
Solution to be added later

.

Motorized Door Program

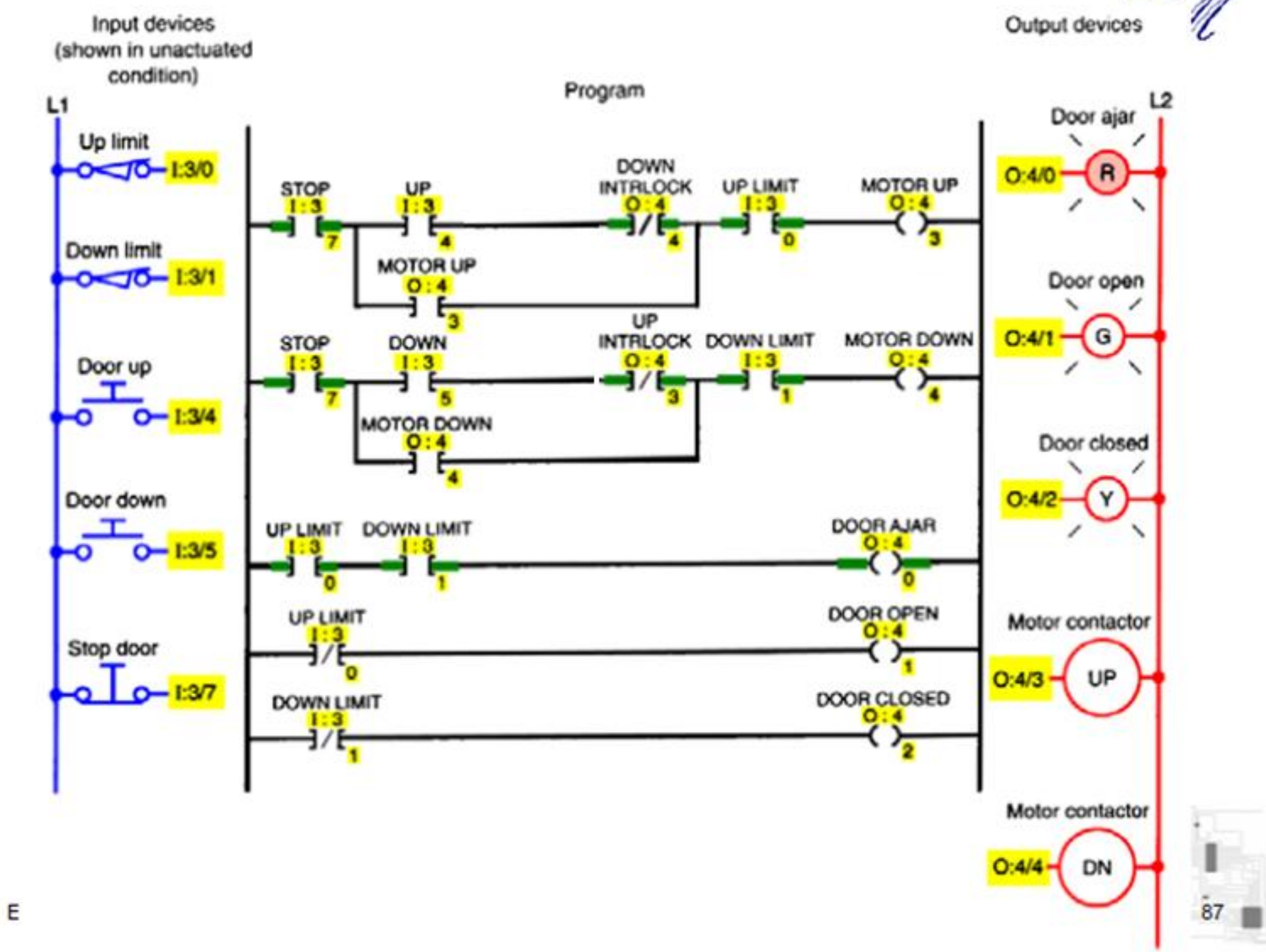


Motorized Door Program



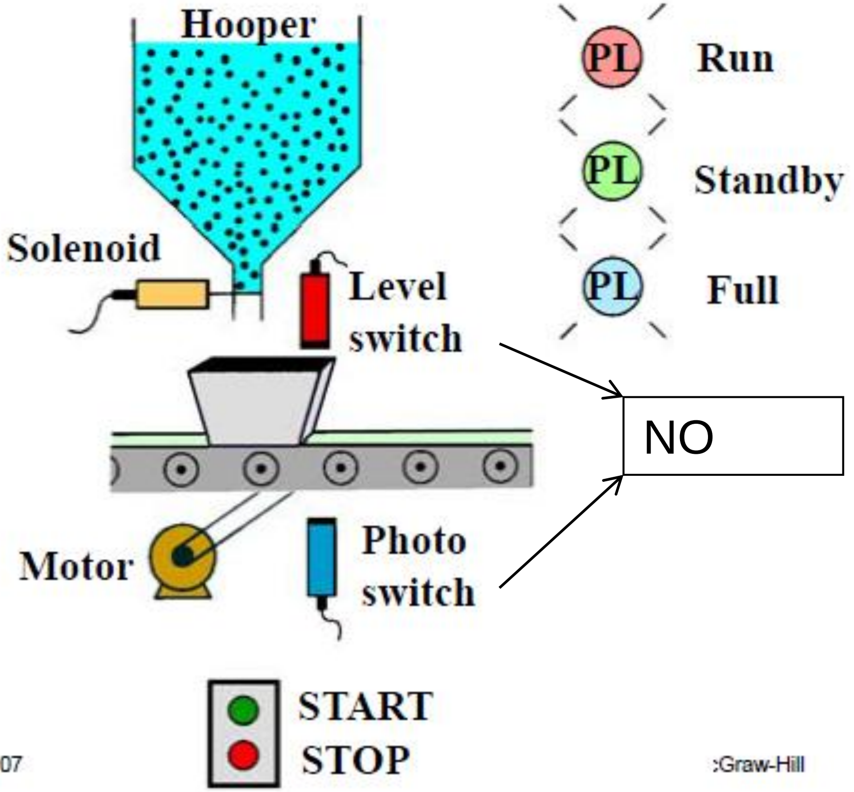
E

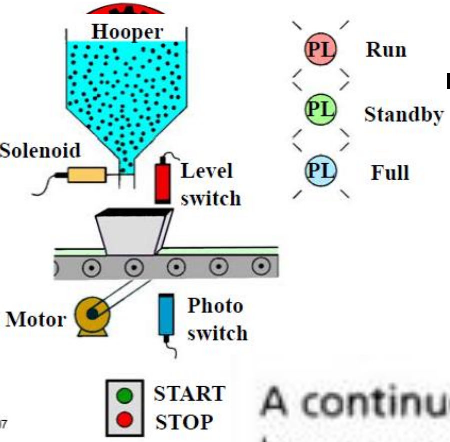
Motorized Door Program



Continuous Filling Operation Program

Description :
A continuous filling operation requires boxes moving on a conveyor to be automatically positioned and filled.





A continuous filling operation requires boxes moving on a conveyor to be automatically positioned and filled.

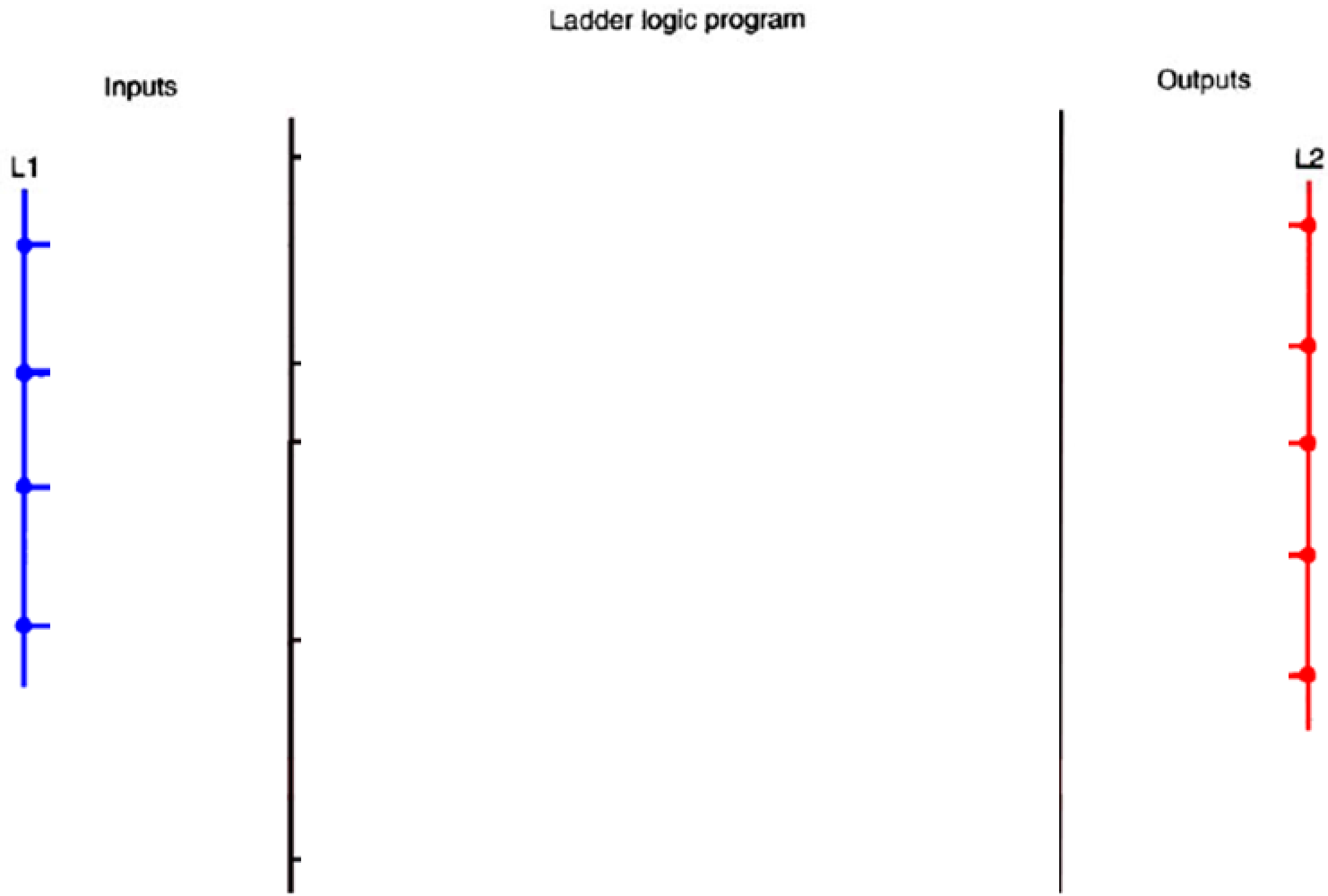
Solution:

- Figure 6-60*a* is an illustration of the process.
- The sequence of operation is as follows:
  - Start the conveyor when the START button is momentarily pressed.
  - Stop the conveyor when the STOP button is momentarily pressed.
  - Energize the RUN status light when the process is operating.
  - Energize the STANDBY status light when the process is stopped.
  - With the box in position and the conveyor stopped, open the solenoid valve and allow the box to fill. Filling should stop when the LEVEL sensor goes true.
  - Energize the FULL light when the box is full. The FULL light should remain energized until the box is moved clear of the photosensor.
- Figure 6-60*b* shows the ladder logic required for the process.

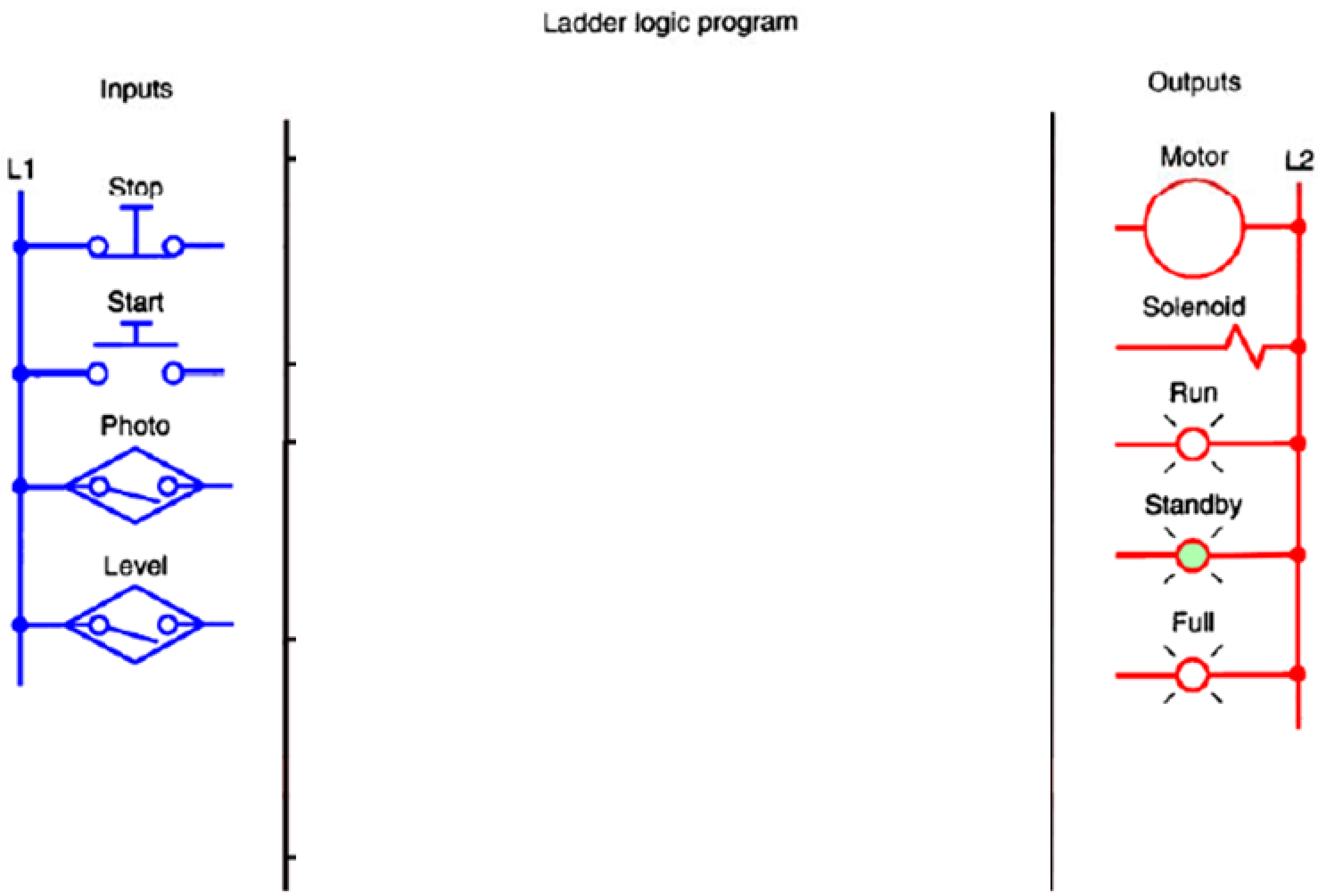
Solution to be added later

.

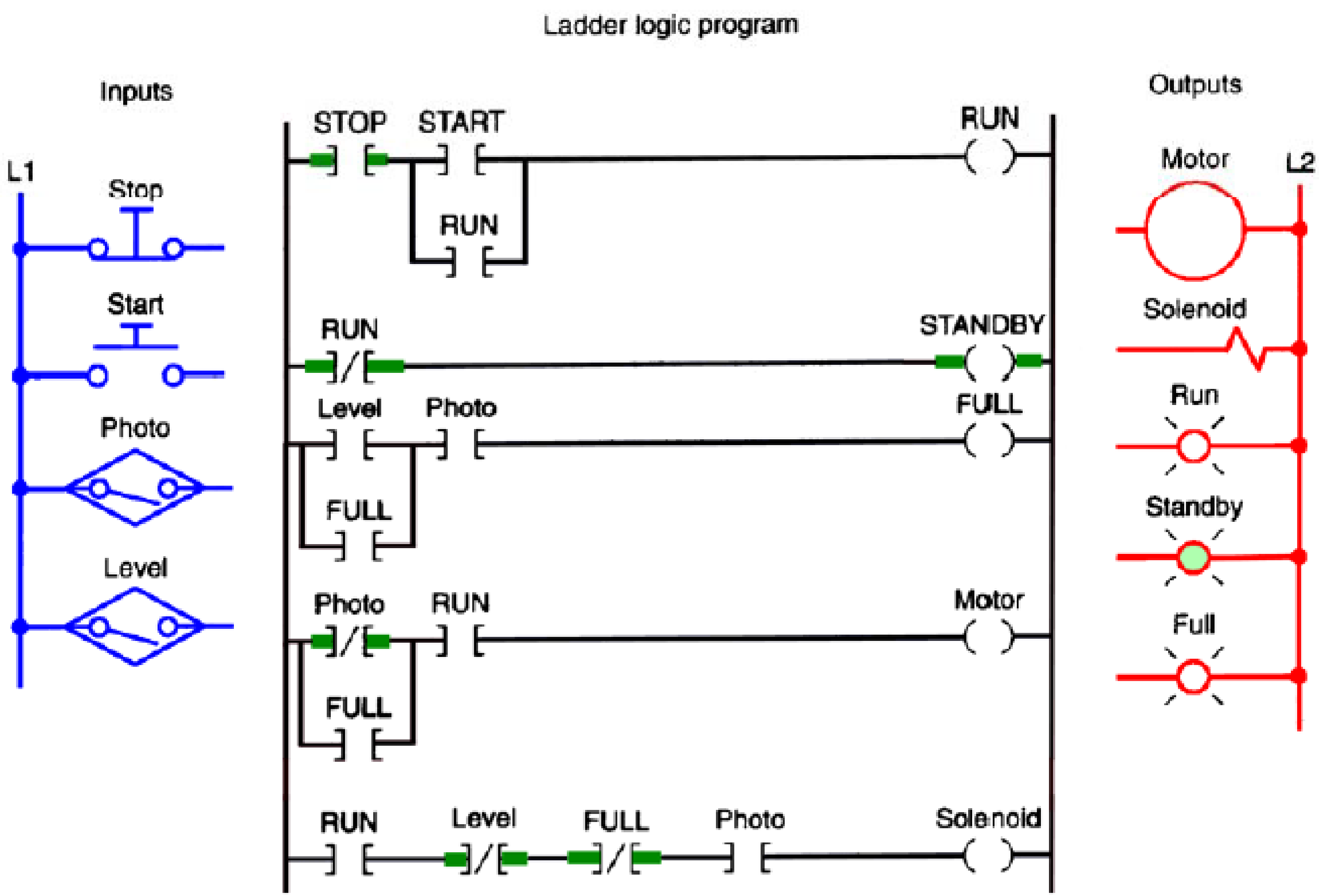
Continuous Filling Operation Program



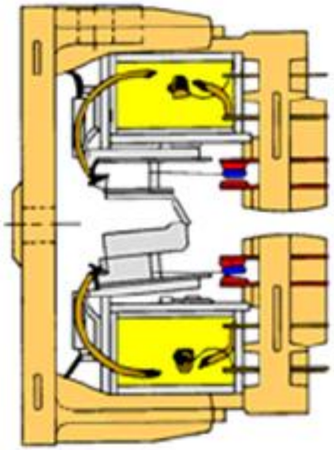
Continuous Filling Operation Program



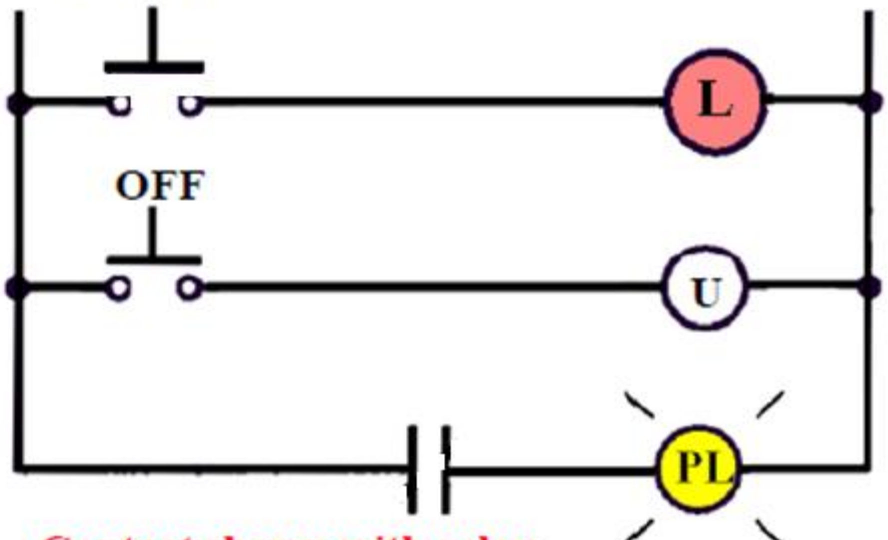
Continuous Filling Operation Program



Latching Relay



Latching relays are used where it is necessary for contacts to stay open and/or closed, even though the coil is energized momentarily.



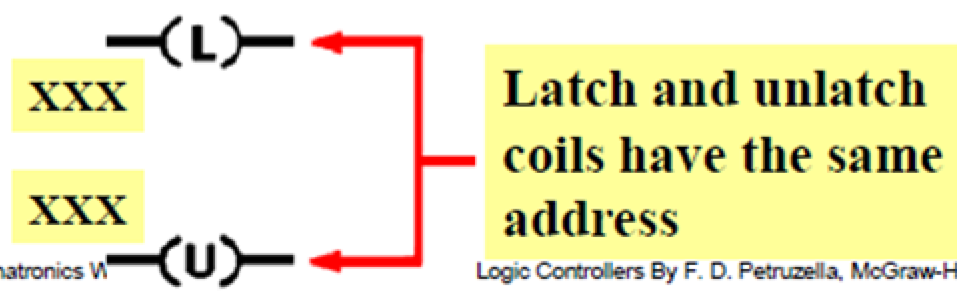
Contact shown with relay

Electromagnetic latching relay circuit

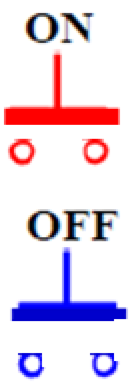
When the ON button is momentarily actuated, the latch coil is energized to set the relay to its latched position. The relay does *not* have to be continuously energized to hold the contact closed.

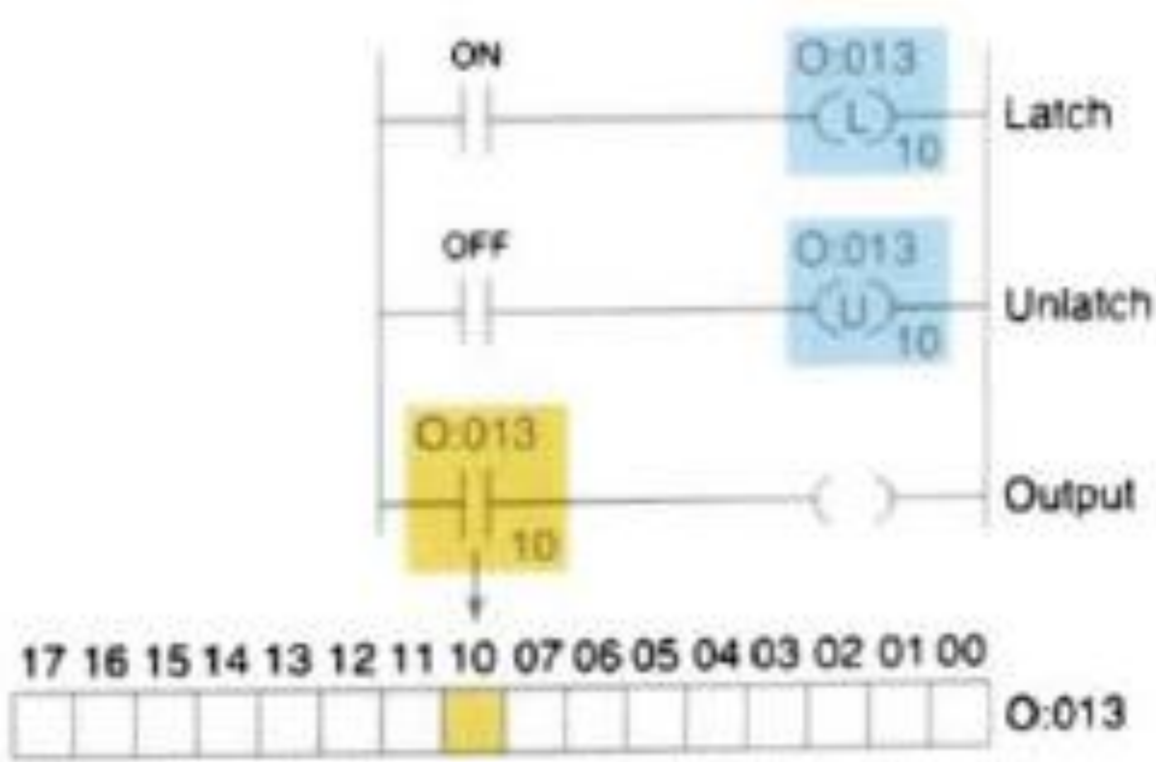
Programmed Latching Relay Instruction

Command	Name	Symbol	Description
OTL	Output Latch	-(L)-	OTL sets the bit to "1" when the rung becomes true, and retains its state when the rung loses continuity
OTU	Output Unlatch	-(U)-	OUT resets the bit to "0" when the rung becomes true and retains it



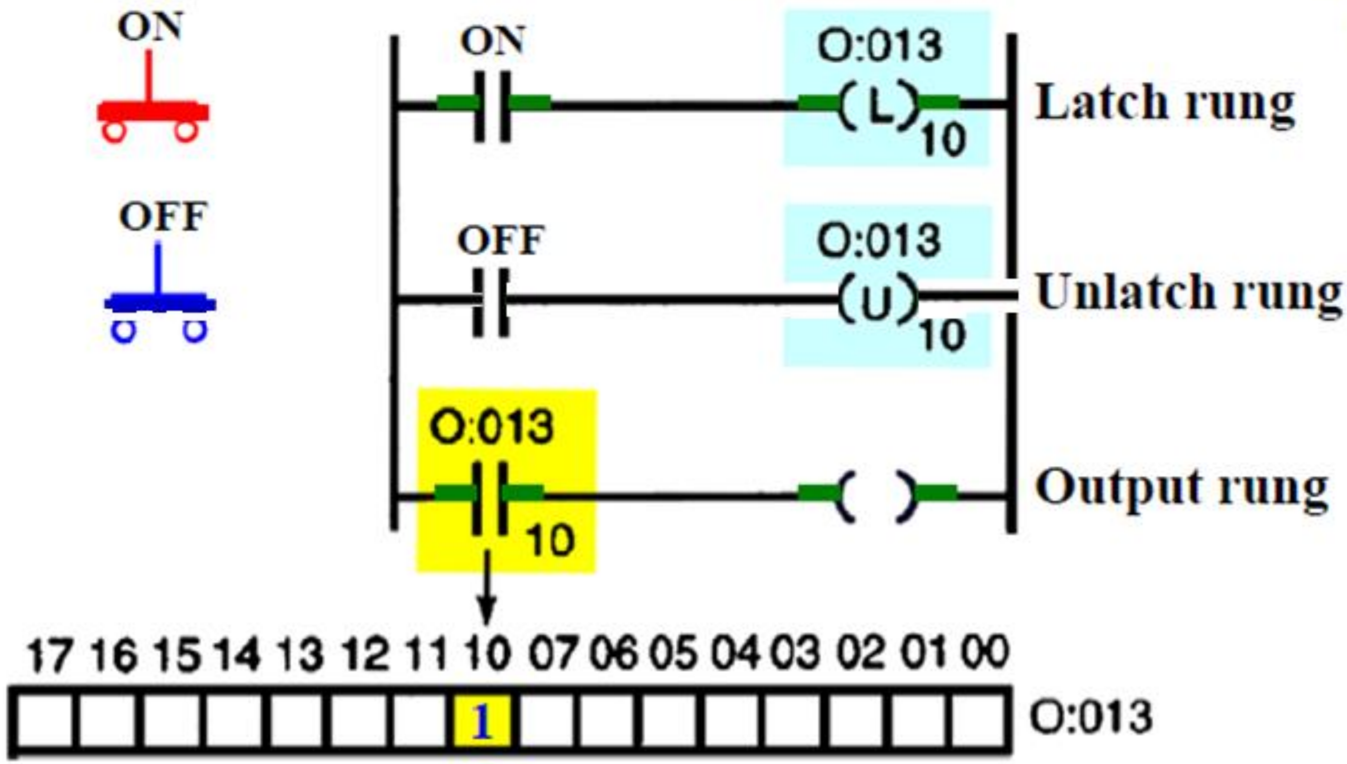
Latching Relay Program





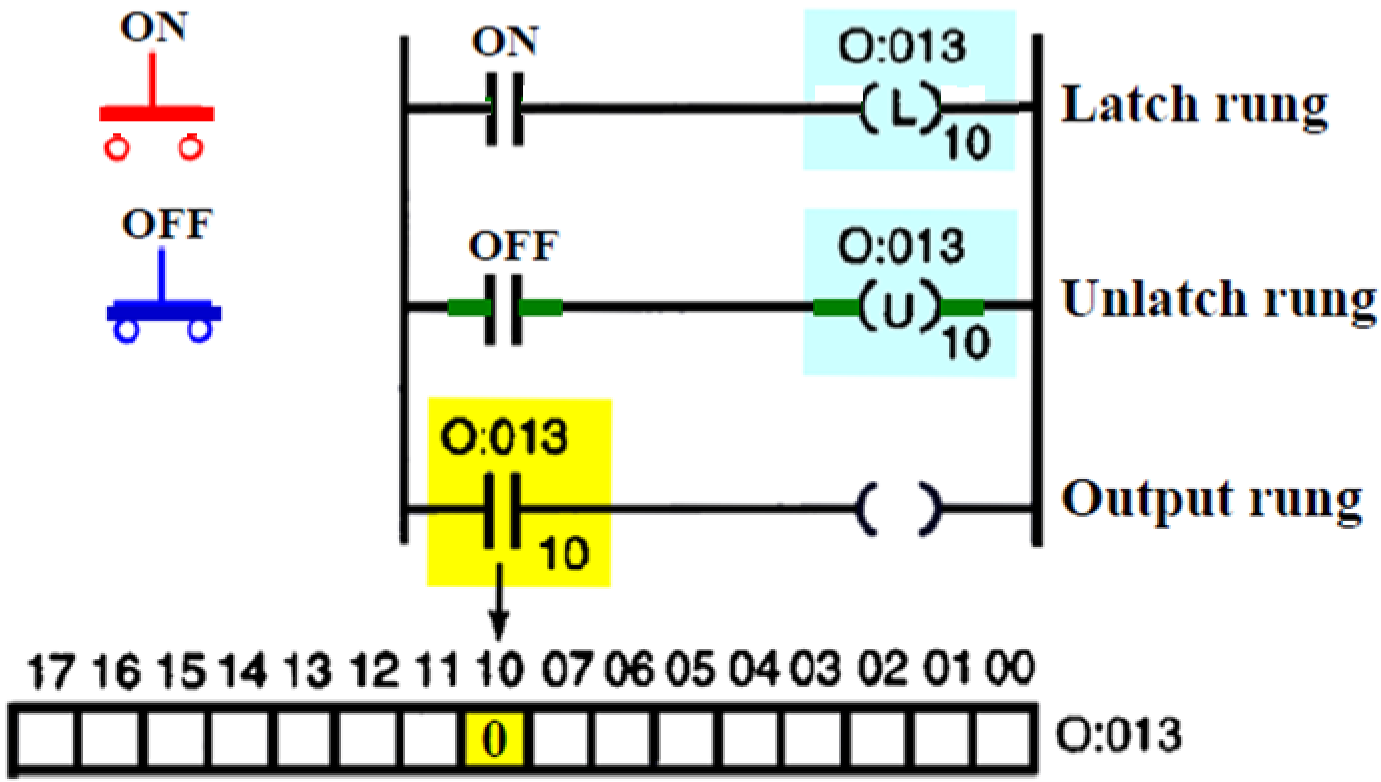
(*b*) Control logic

Latching Relay Program



When the ON button is momentarily actuated, the latch rung becomes true and the latch status bit (10) is set to 1, and so the output is switched on. This status bit will remain on (1) when logic continuity of the latch rung is lost.

Latching Relay Program



When the unlatch rung becomes true (OFF button actuated), the status bit (10) is reset back to 0 and so the output is switched off.

3. Study the ladder logic program in Figure 6-61, and answer the questions that follow:

a. Under what condition will the latch rung 1 be TRUE?

b. Under what conditions will the unlatch rung 2 be TRUE?

c. Under what condition will rung 3 be TRUE?

d. When PL1 is on, the relay is in what state (LATCHED or UNLATCHED)?

e. When PL2 is on, the relay is in what state (LATCHED or UNLATCHED)?

f. Assume the relay is in its LATCHED state and all three inputs are FALSE. What input change(s) must occur for the relay to switch into its UNLATCHED state?

g. If the examine if closed instructions at addresses I/1, I/2, and I/3 are all TRUE, what state will the relay remain in (LATCHED or UNLATCHED)?

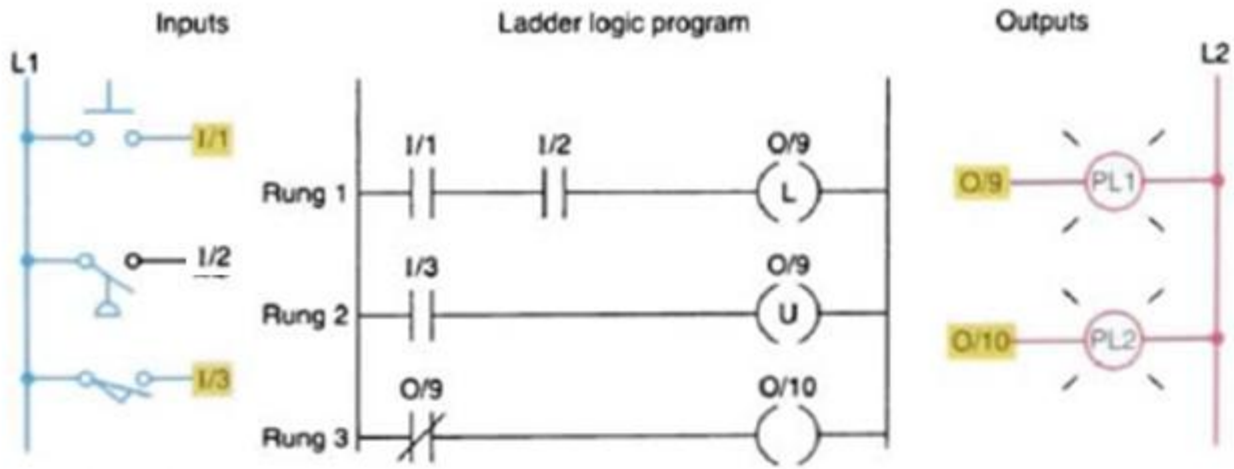
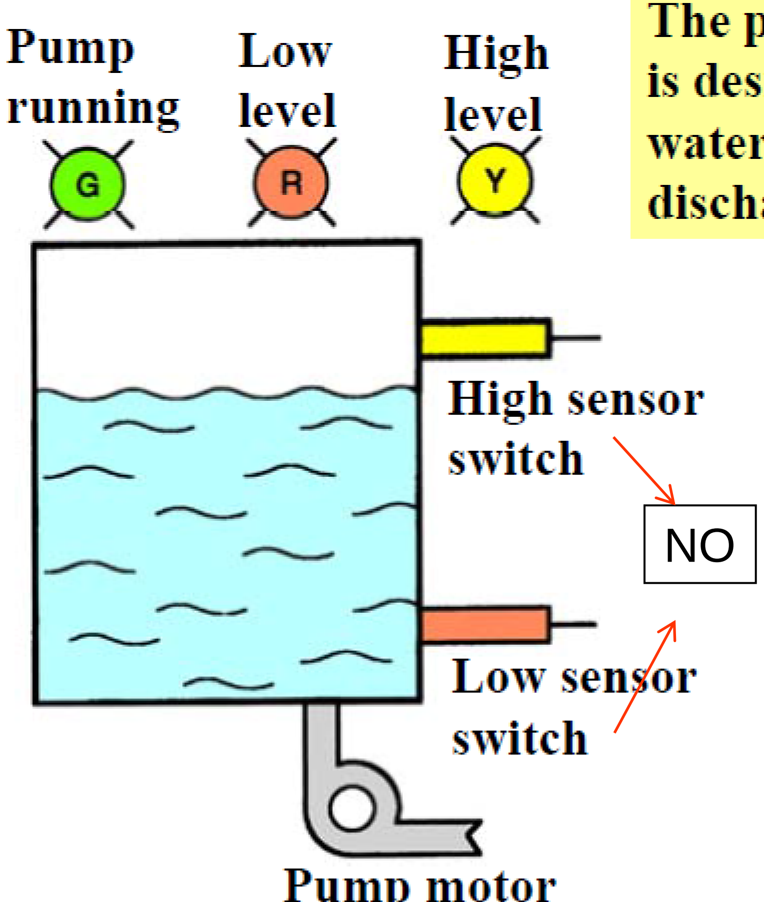


FIGURE 6-61

PLC Water Level Program



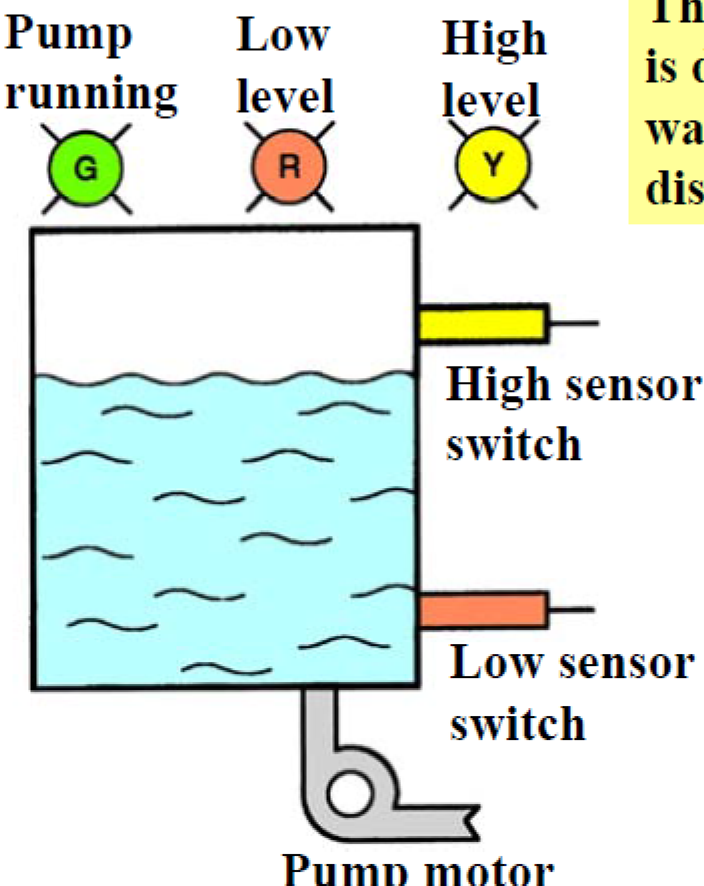
The program for this process is designed to control the level of water in a storage tank by turning a discharge pump on and off.

Modes of Operation

Off Position: the pump will stop if it is running and will not start if it is stopped.

Manual Mode: the pump will start if the water in the tank is at any level except low.

PLC Water Level Program



The program for this process is designed to control the level of water in a storage tank by turning a discharge pump on and off.

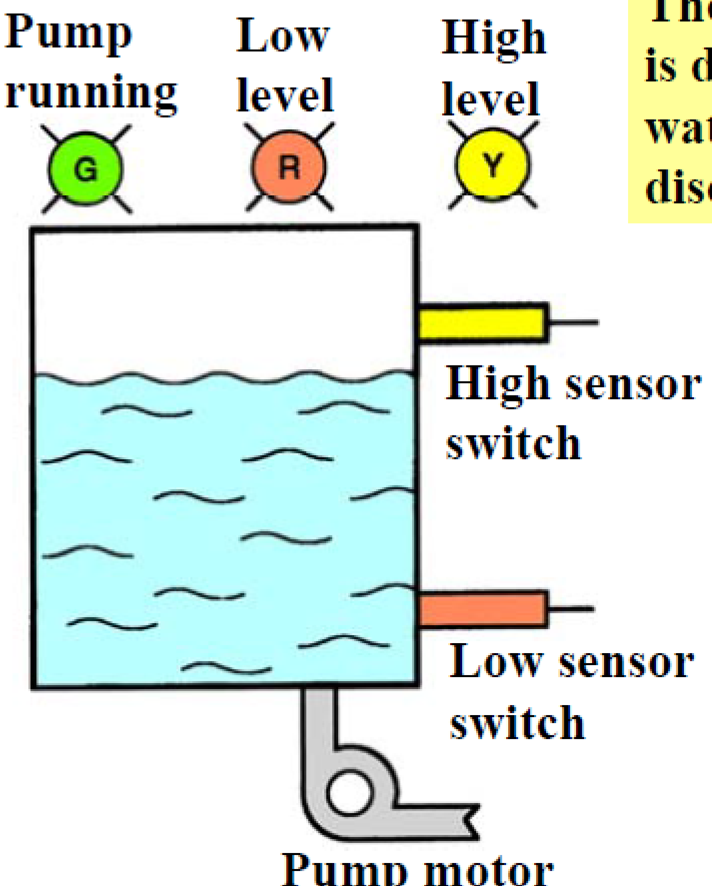
Modes of Operation

Automatic Mode:

- if the level of water in the tank reaches a high point, the water pump will start so that water can be removed from the tank, thus lowering the level
- when the water level reaches a low point, the pump will stop

PLC Water Level Program

Radu Muresan

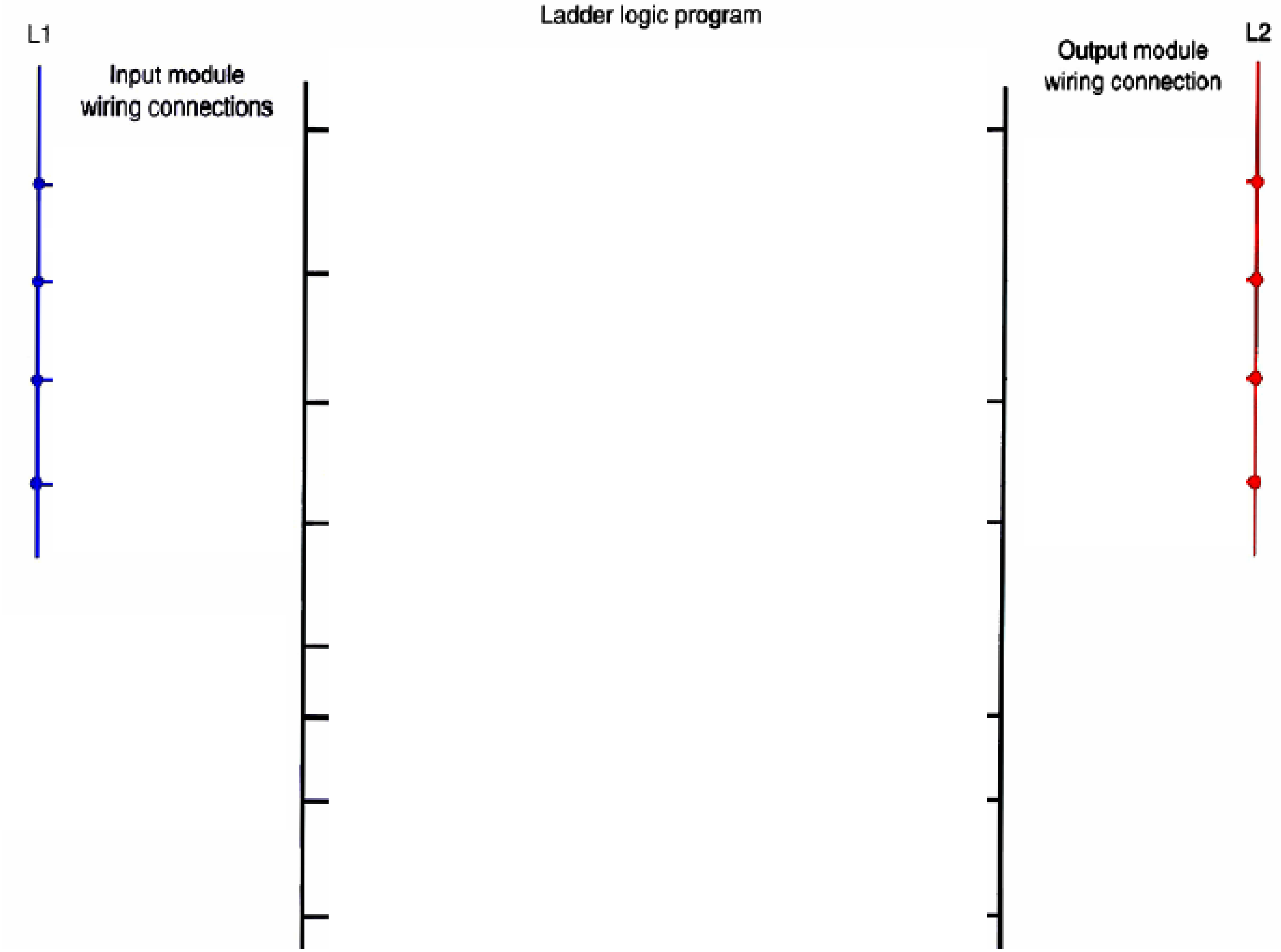


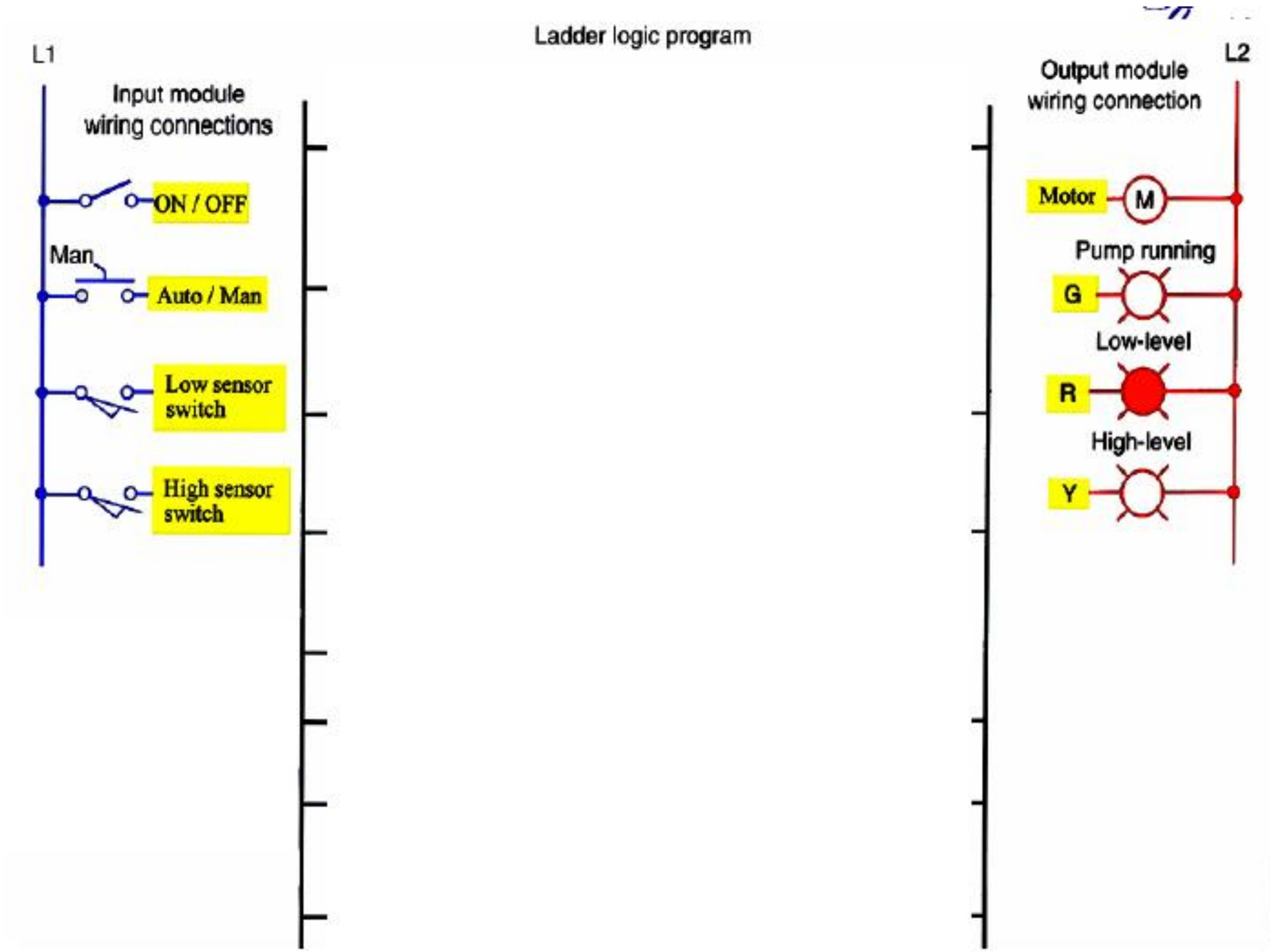
The program for this process is designed to control the level of water in a storage tank by turning a discharge pump on and off.

Modes of Operation

Status Indicating Lights:

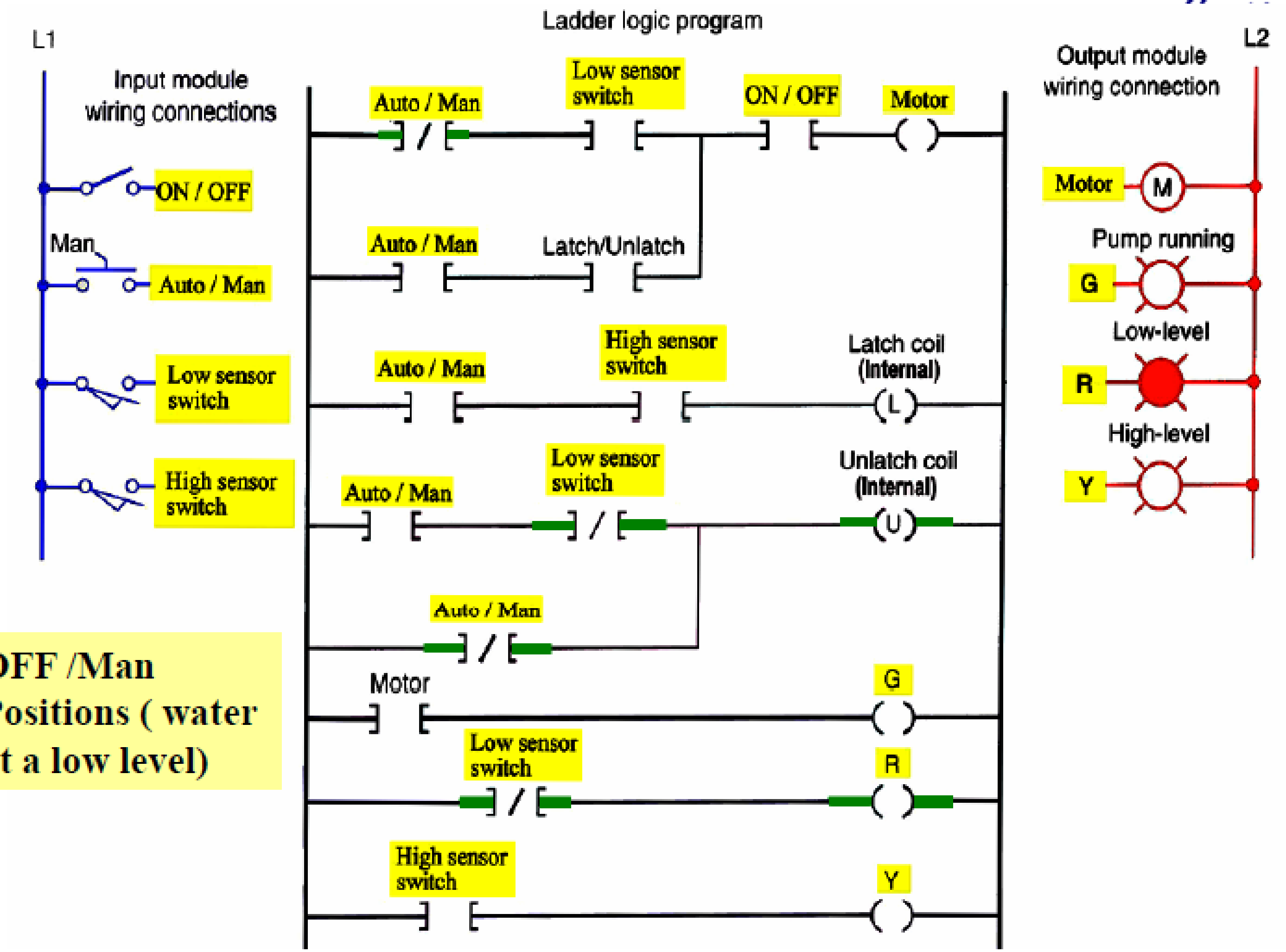
- water pump running light (green)
- low water level status light (red)
- high water level status light (yellow)


Ladder logic program
L1
Input module
wiring connections
L2
Output module
wiring connection


Ladder logic program
L1
L2
Input module
wiring connections
Output module
wiring connection
ON / OFF
Man
Auto / Man
Low sensor
switch
High sensor
switch
Motor
M
Pump running
G
Low-level
R
High-level
Y

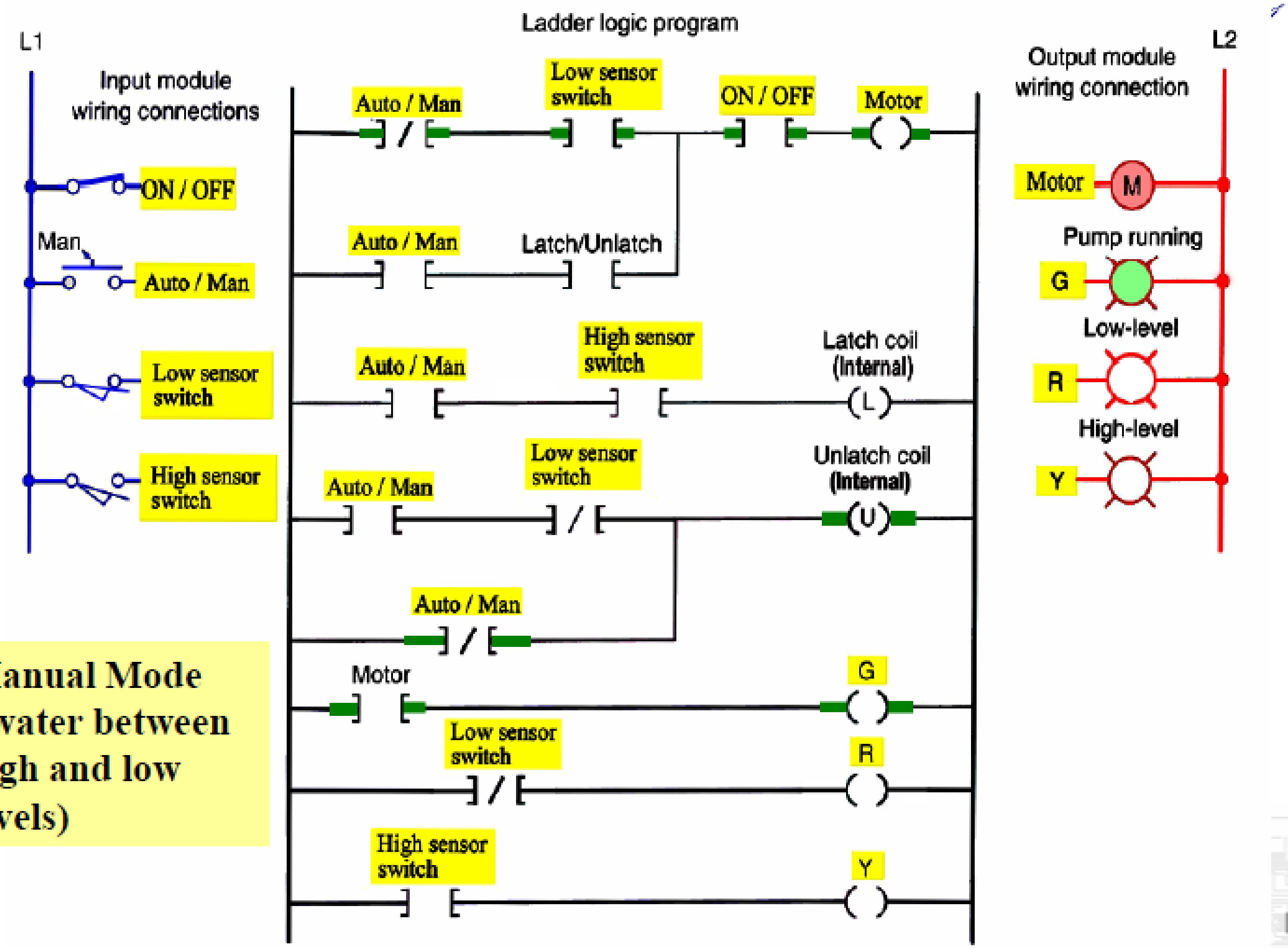
PLC Water Level Program

Radu Muresan



OFF /Man Positions (water at a low level)

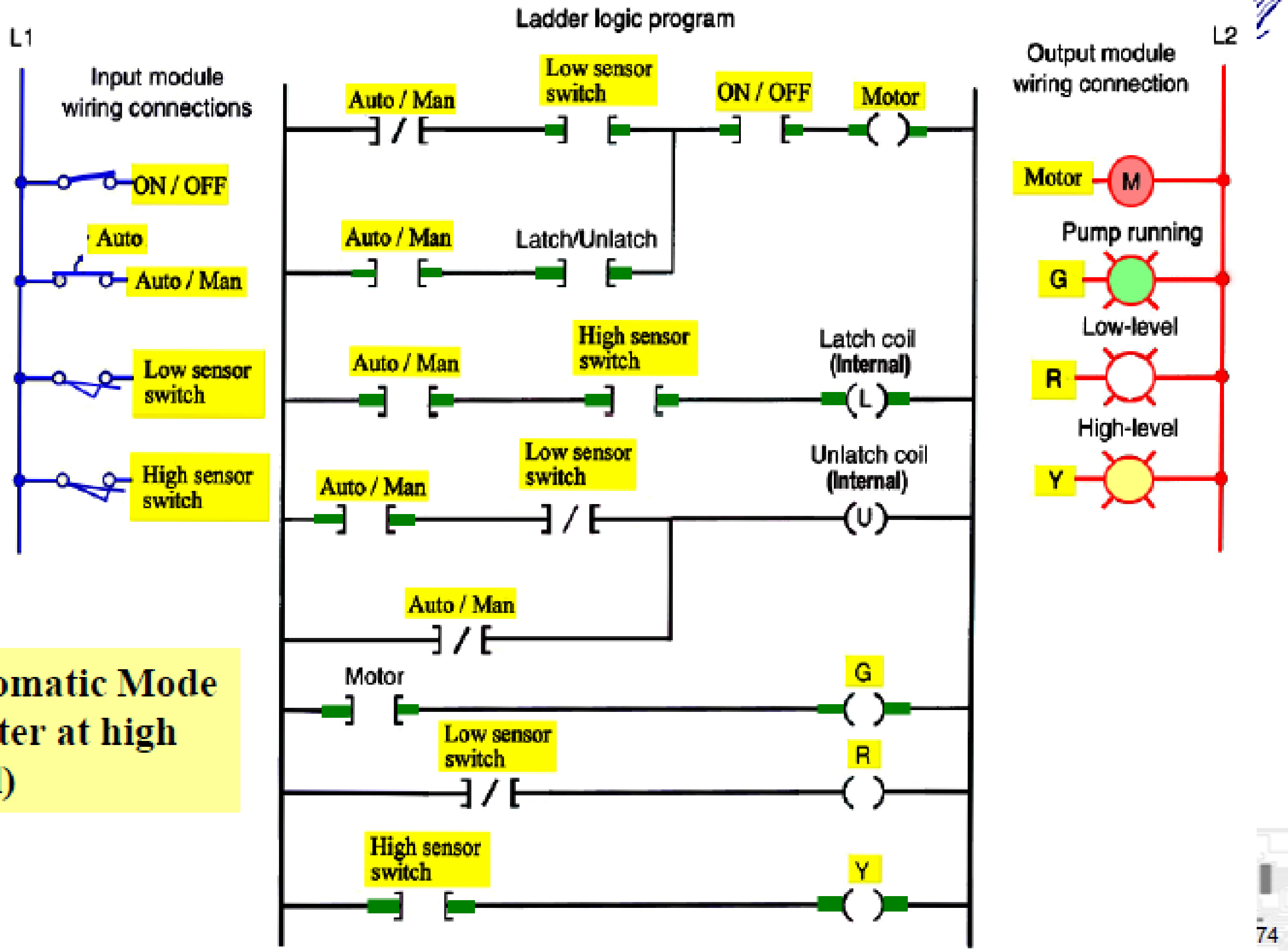
PLC Water Level Program



Manual Mode (water between high and low levels)

PLC Water Level Program

Radu Muresan



Automatic Mode (water at high level)

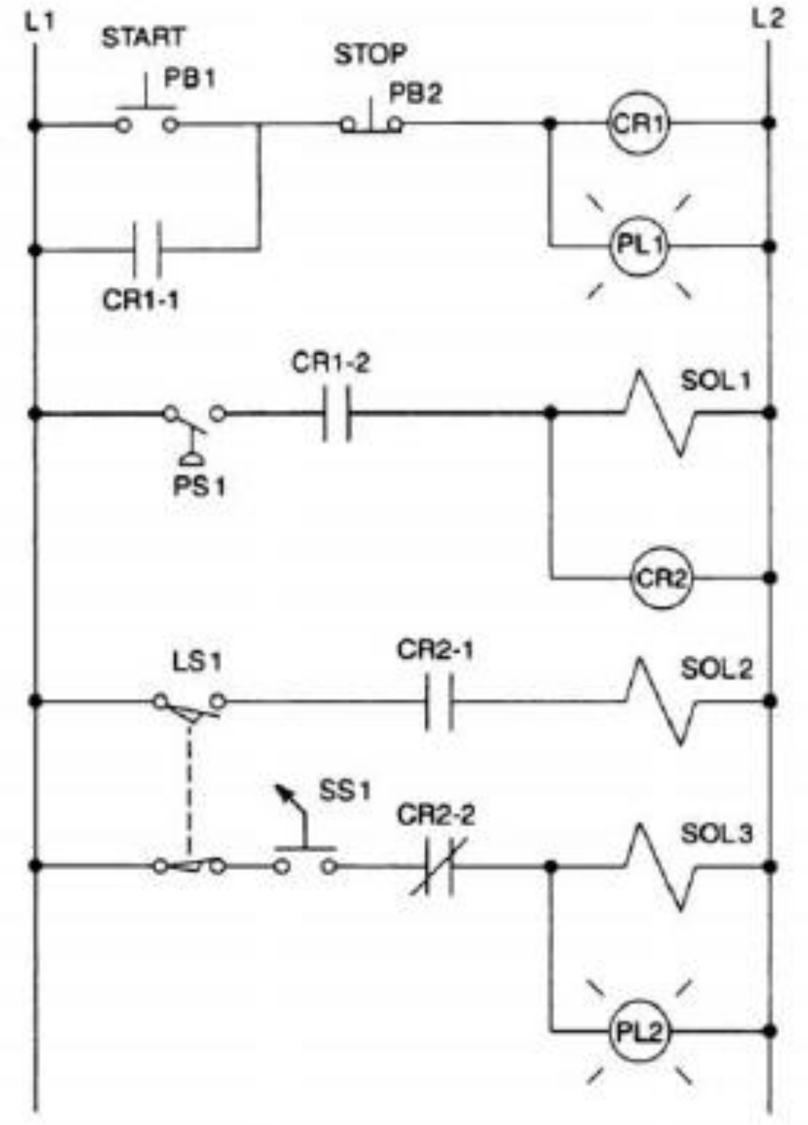
74

1. Design and draw the schematic for a conventional hardwired relay circuit that will perform each of the following circuit functions when an NC pushbutton is pressed:

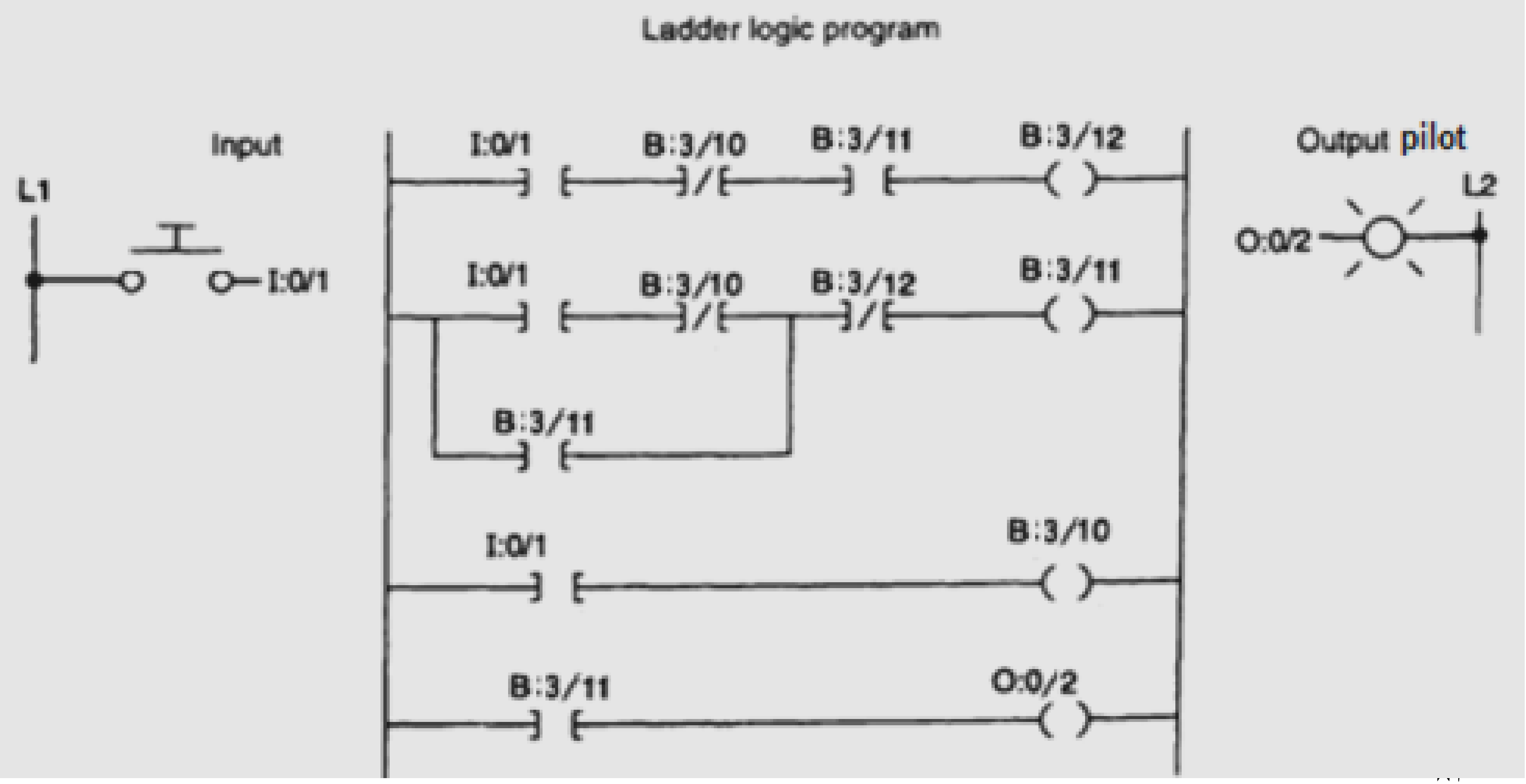
- Switch a pilot light on
- De-energize a solenoid
- Start a motor running

.

5. Design a PLC program and prepare a typical I/O connection diagram and ladder logic program that will correctly execute the hardwired control circuit in Figure 6-63.

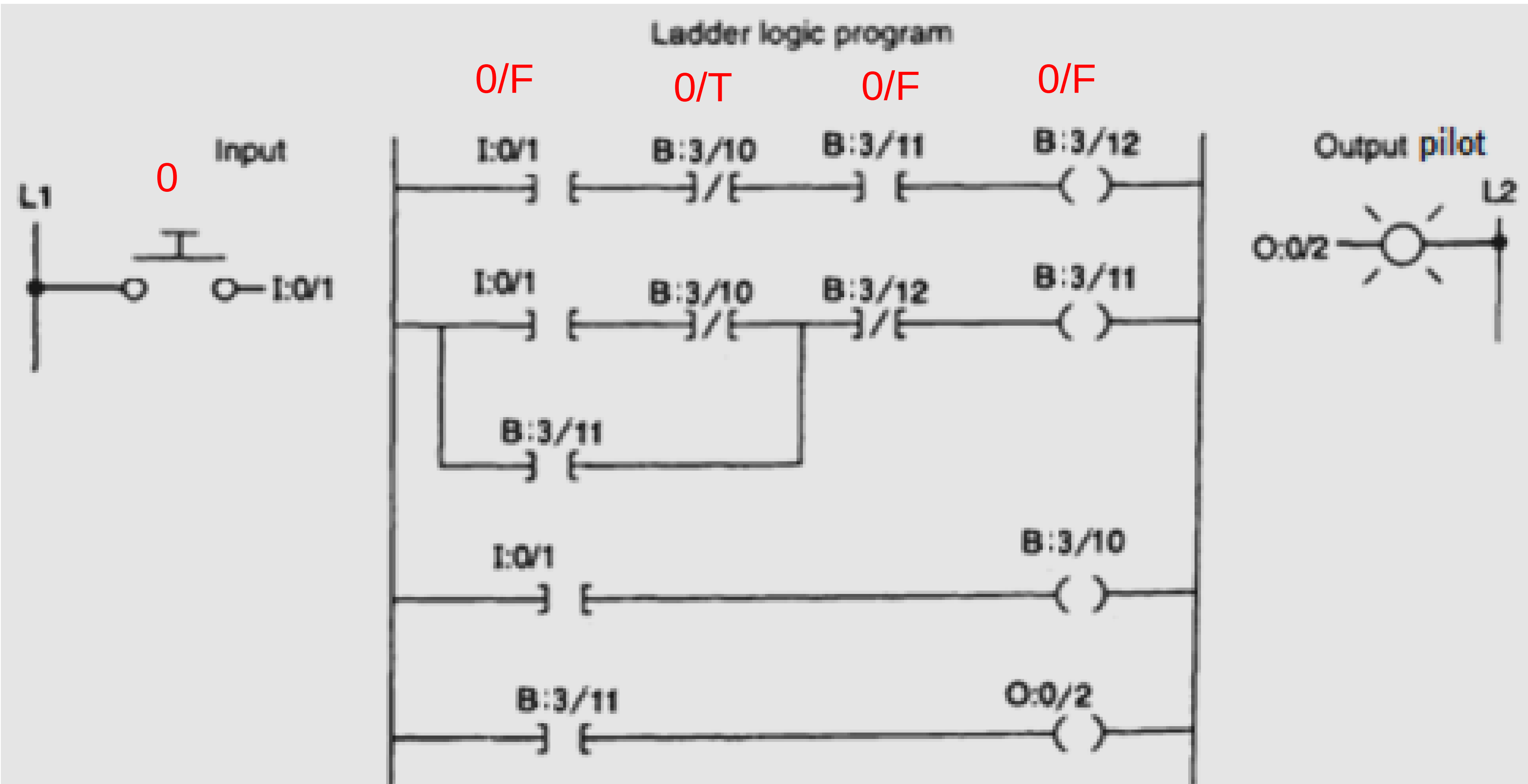


1) What will happened to the output pilot when the push button in the below ladder logic diagram is momentarily press for the **first** time?
2) What will happened to the output pilot when the push button in the below ladder logic diagram is momentarily press for the **second** time?



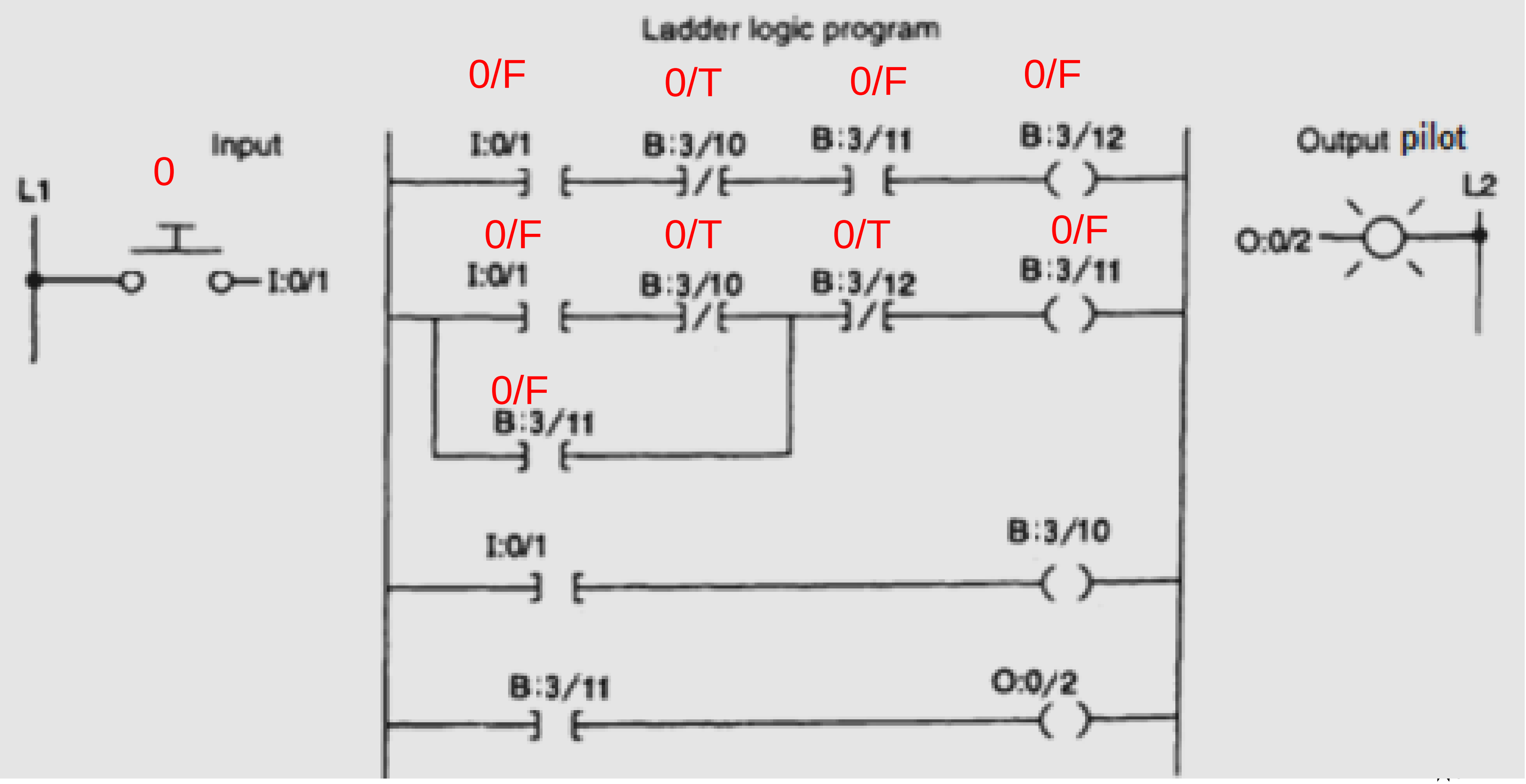
1) What will happened to the output pilot when the push button in the below ladder logic diagram is momentarily press for the **first** time?

Initial scan



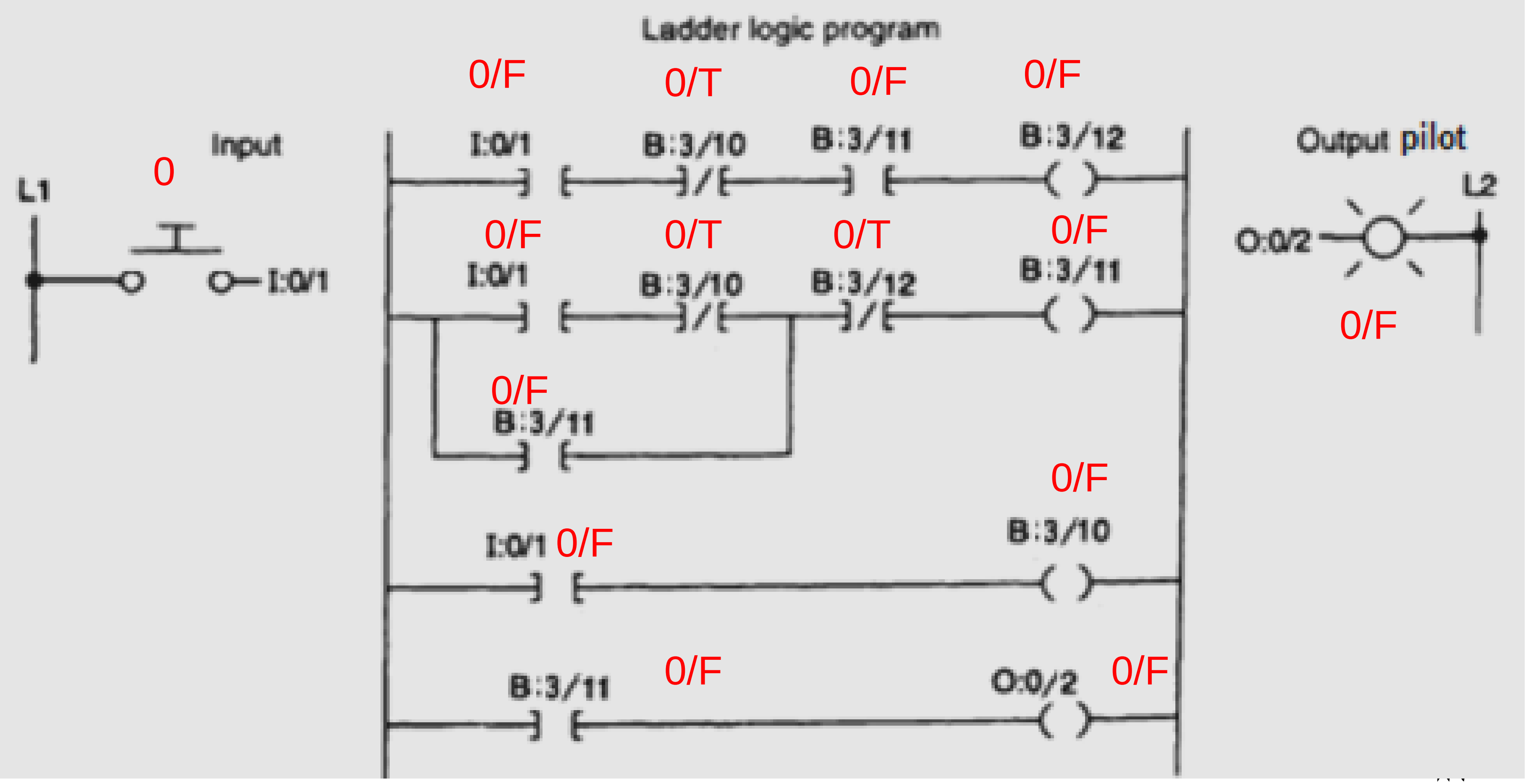
1) What will happened to the output pilot when the push button in the below ladder logic diagram is momentarily press for the **first** time?

Initial scan



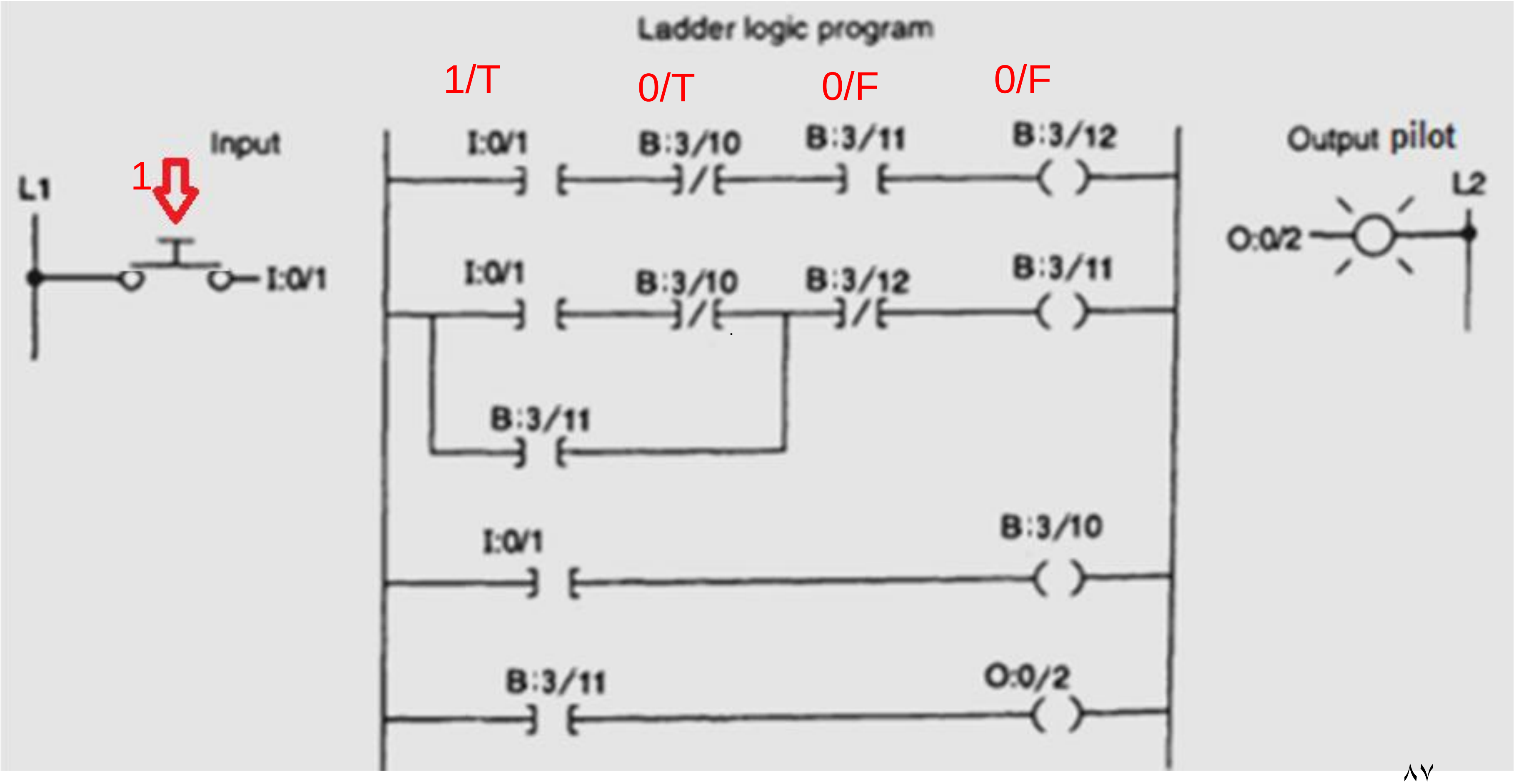
1) What will happened to the output pilot when the push button in the below ladder logic diagram is momentarily press for the **first** time?

Initial scan

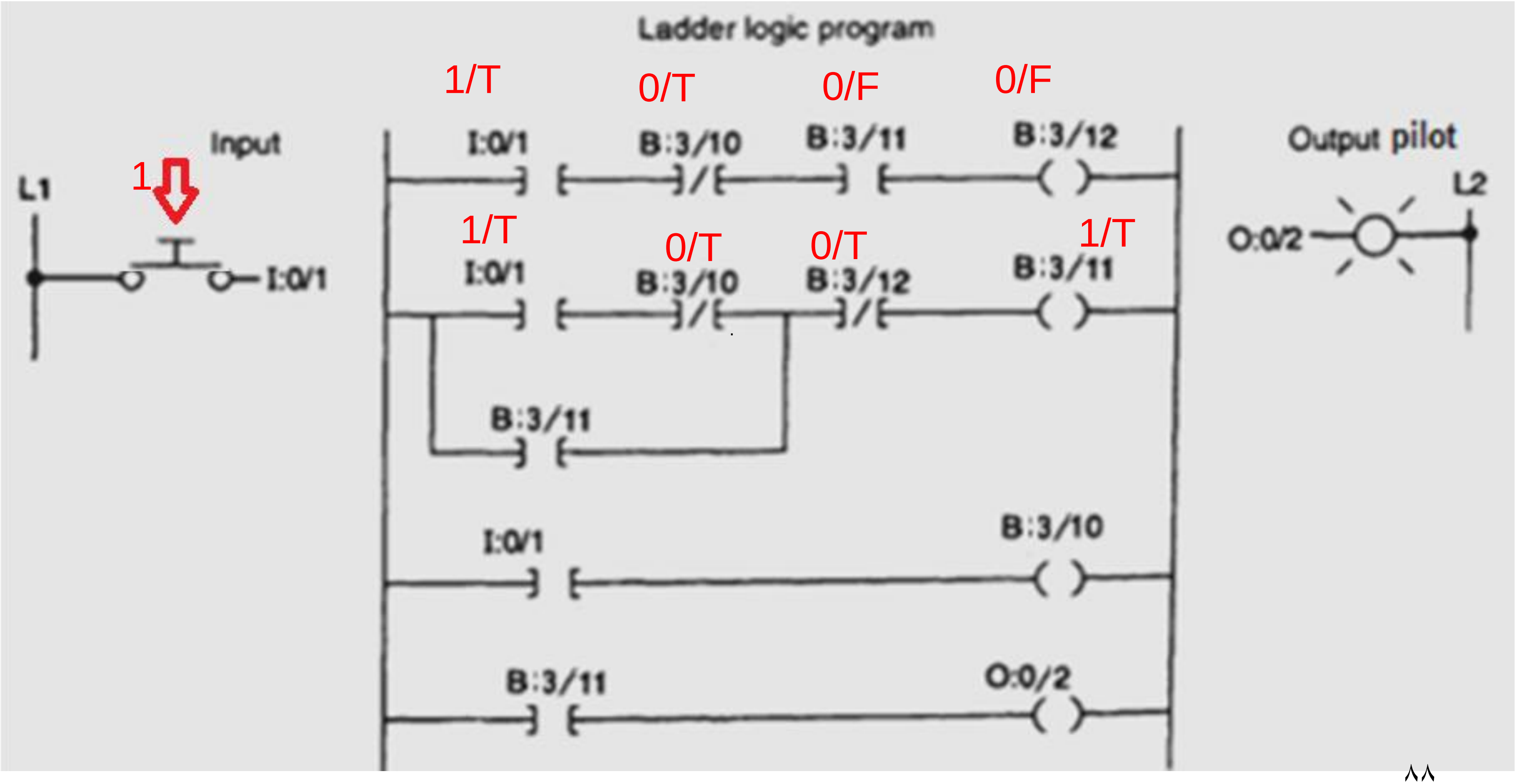


1) What will happened to the output pilot when the push button in the below ladder logic diagram is momentarily press for the **first** time?

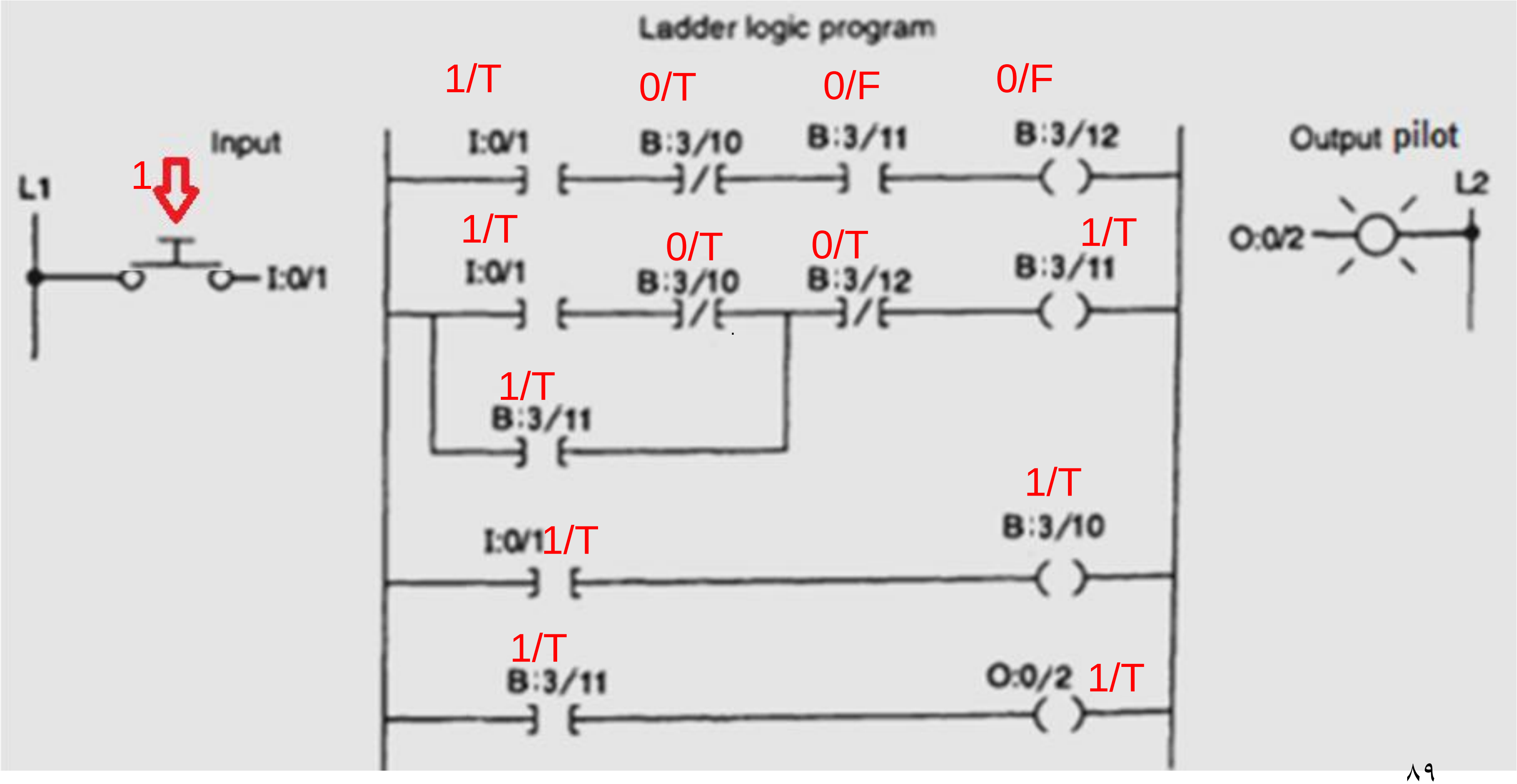
press push button



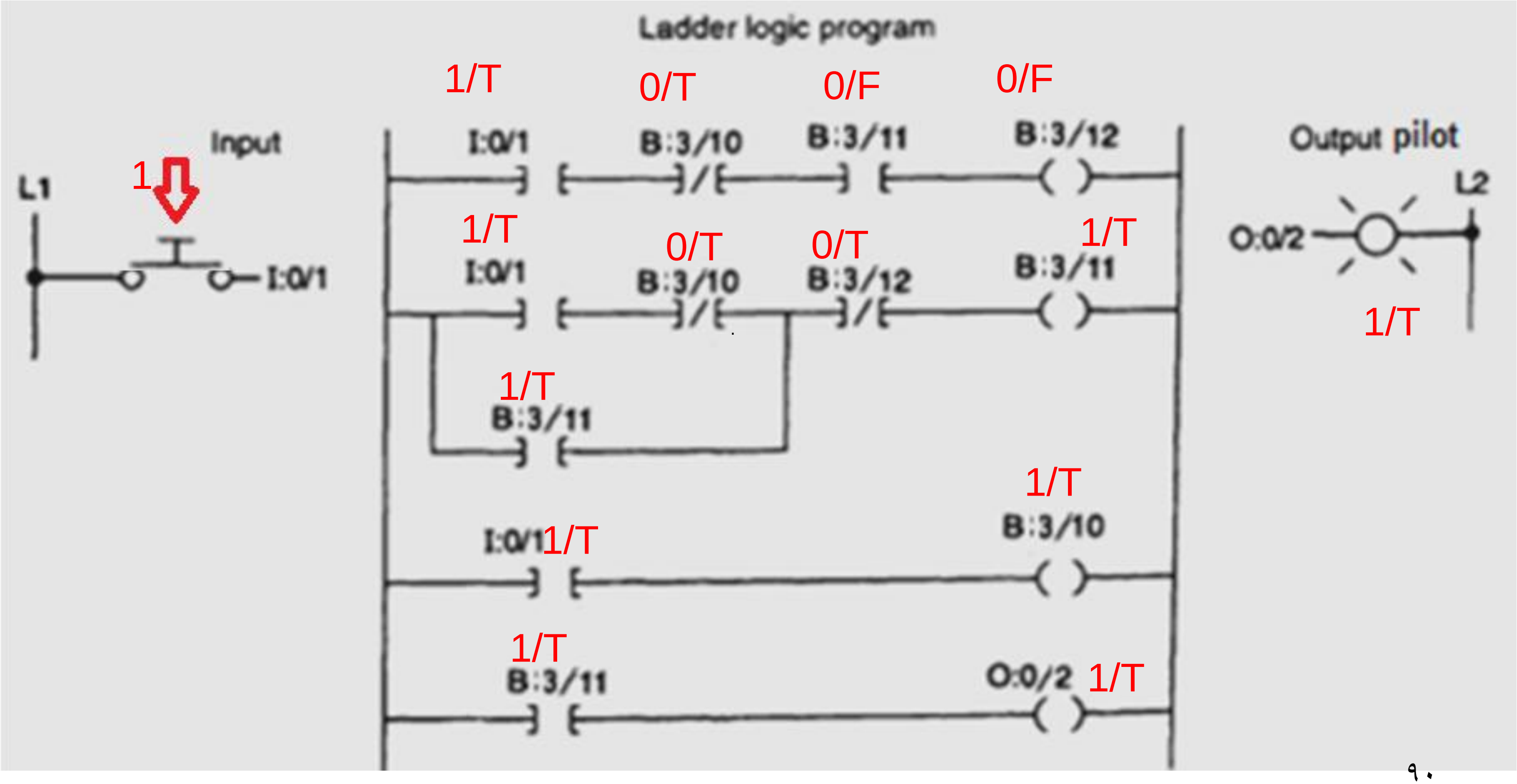
1) What will happened to the output pilot when the push button in the below ladder logic diagram is momentarily press for the **first** time?



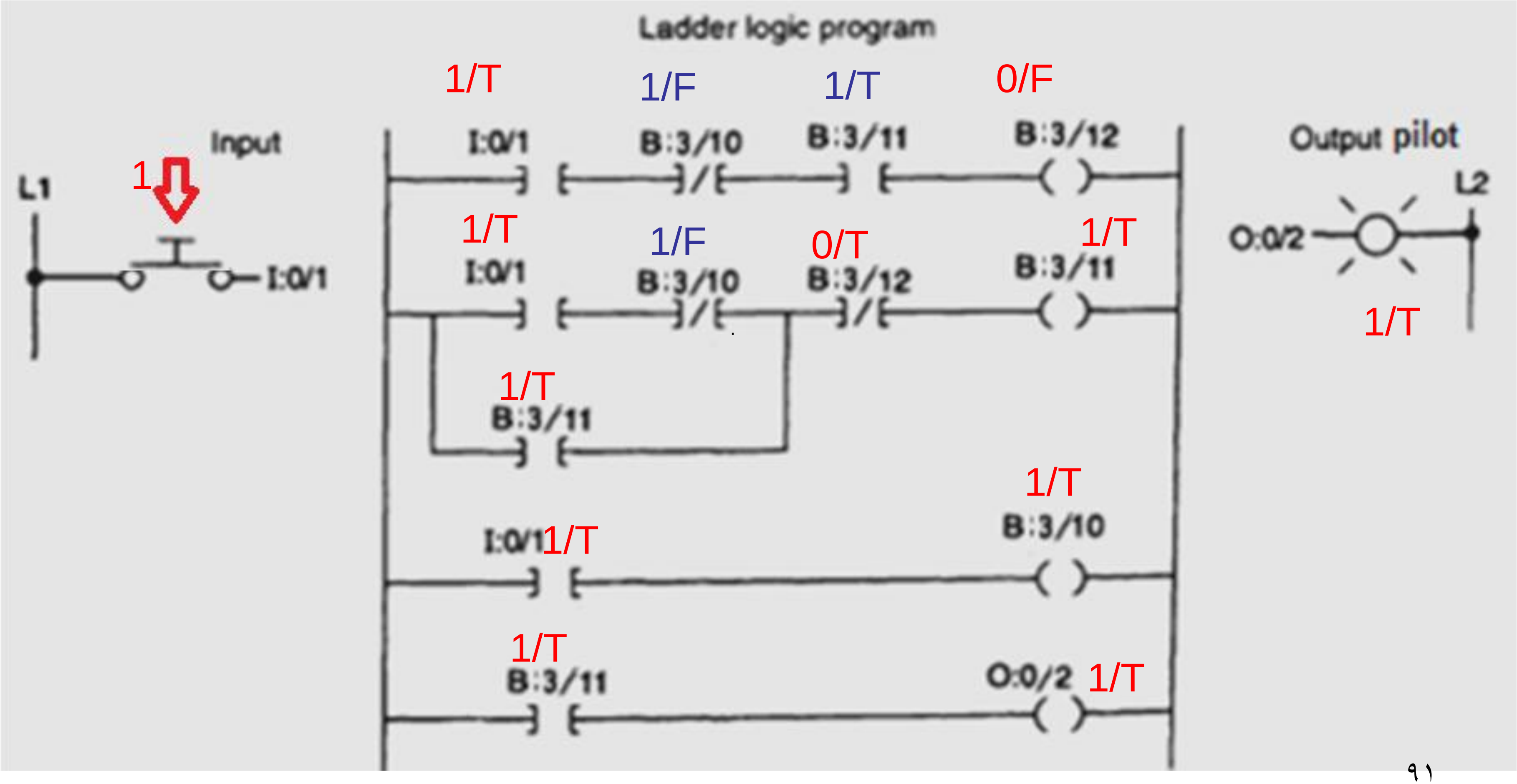
1) What will happened to the output pilot when the push button in the below ladder logic diagram is momentarily press for the **first** time?



1) What will happened to the output pilot when the push button in the below ladder logic diagram is momentarily press for the **first** time?

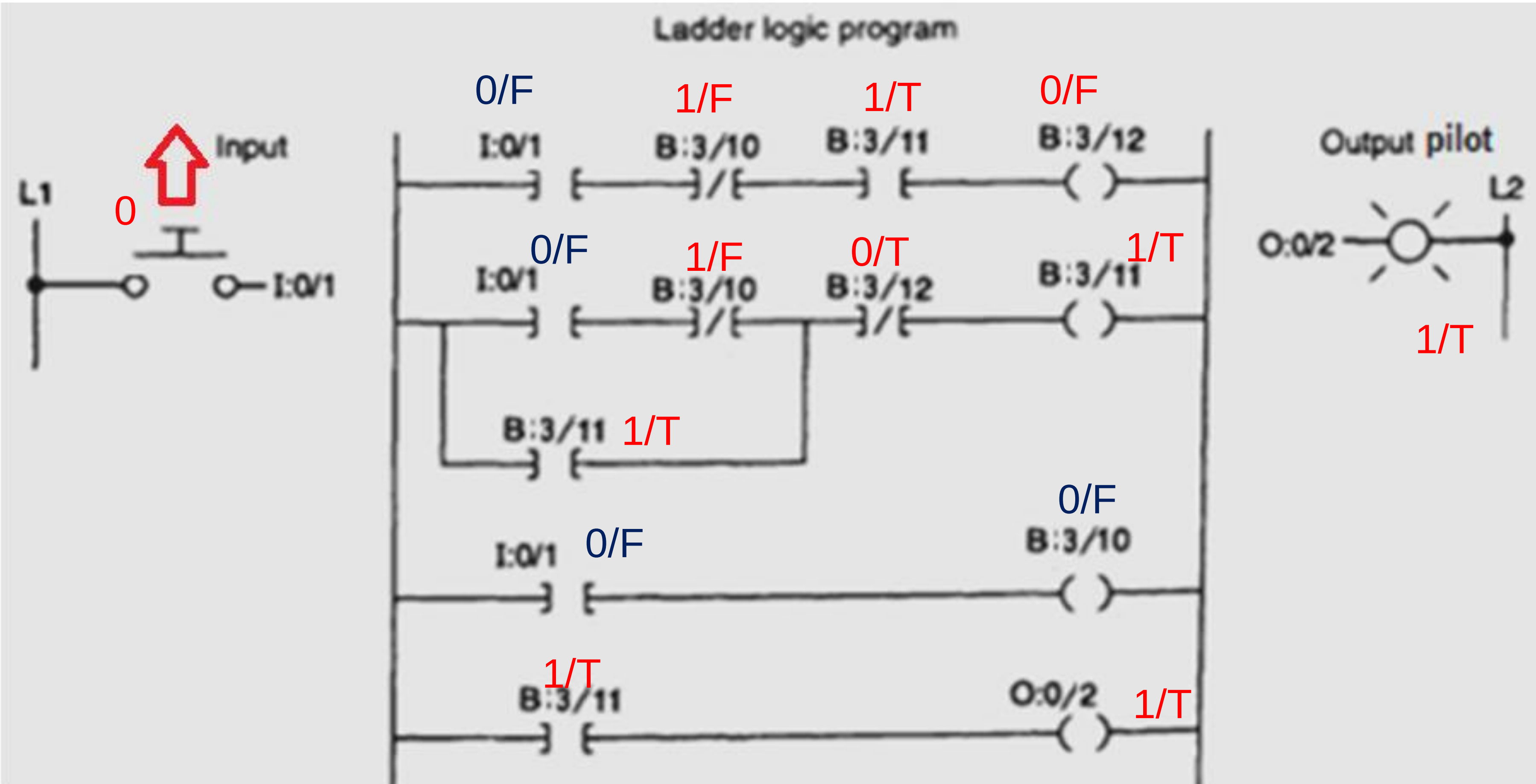


1) What will happened to the output pilot when the push button in the below ladder logic diagram is momentarily press for the **first** time?



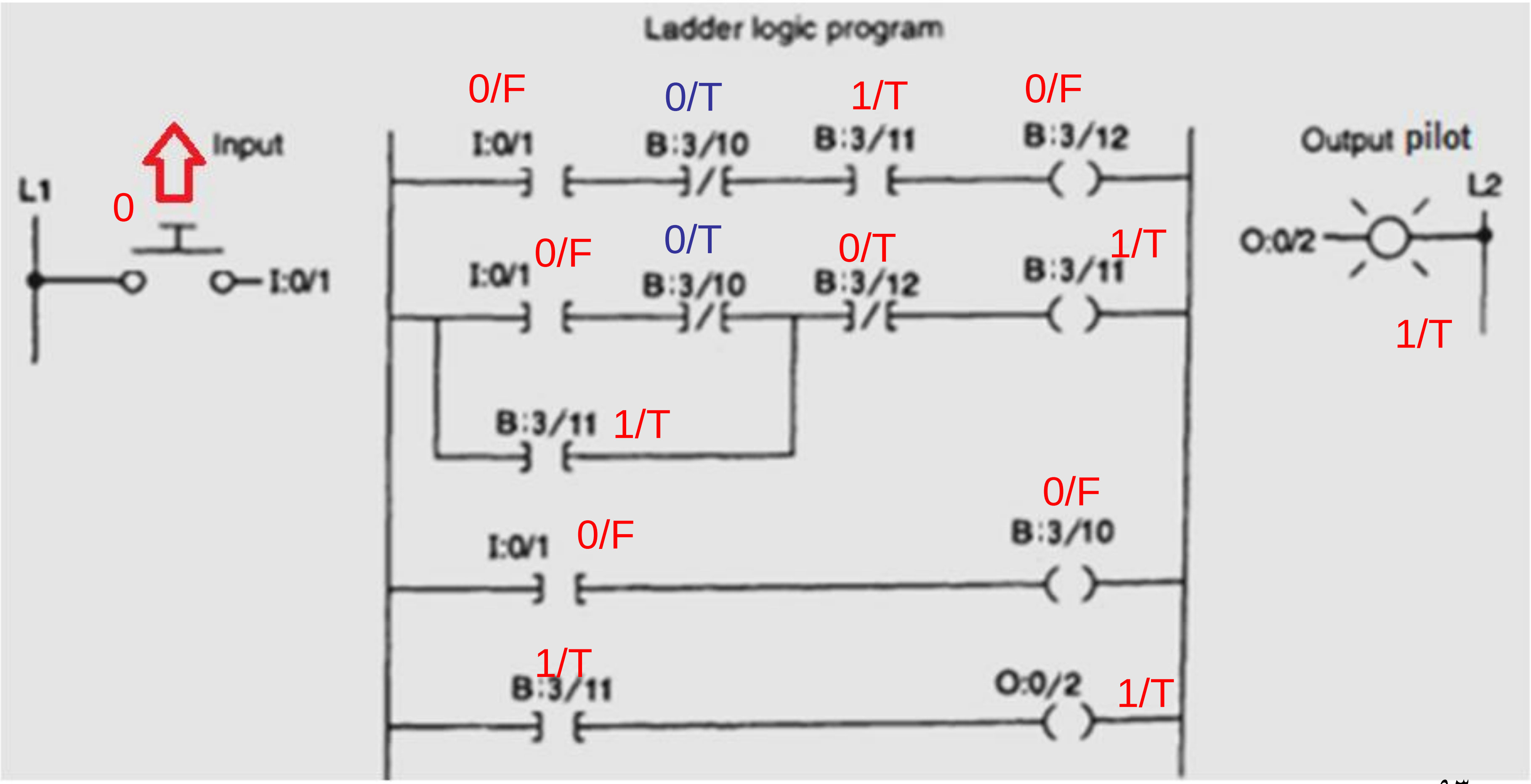
1) What will happened to the output pilot when the push button in the below ladder logic diagram is momentarily press for the **first** time?

Release push button

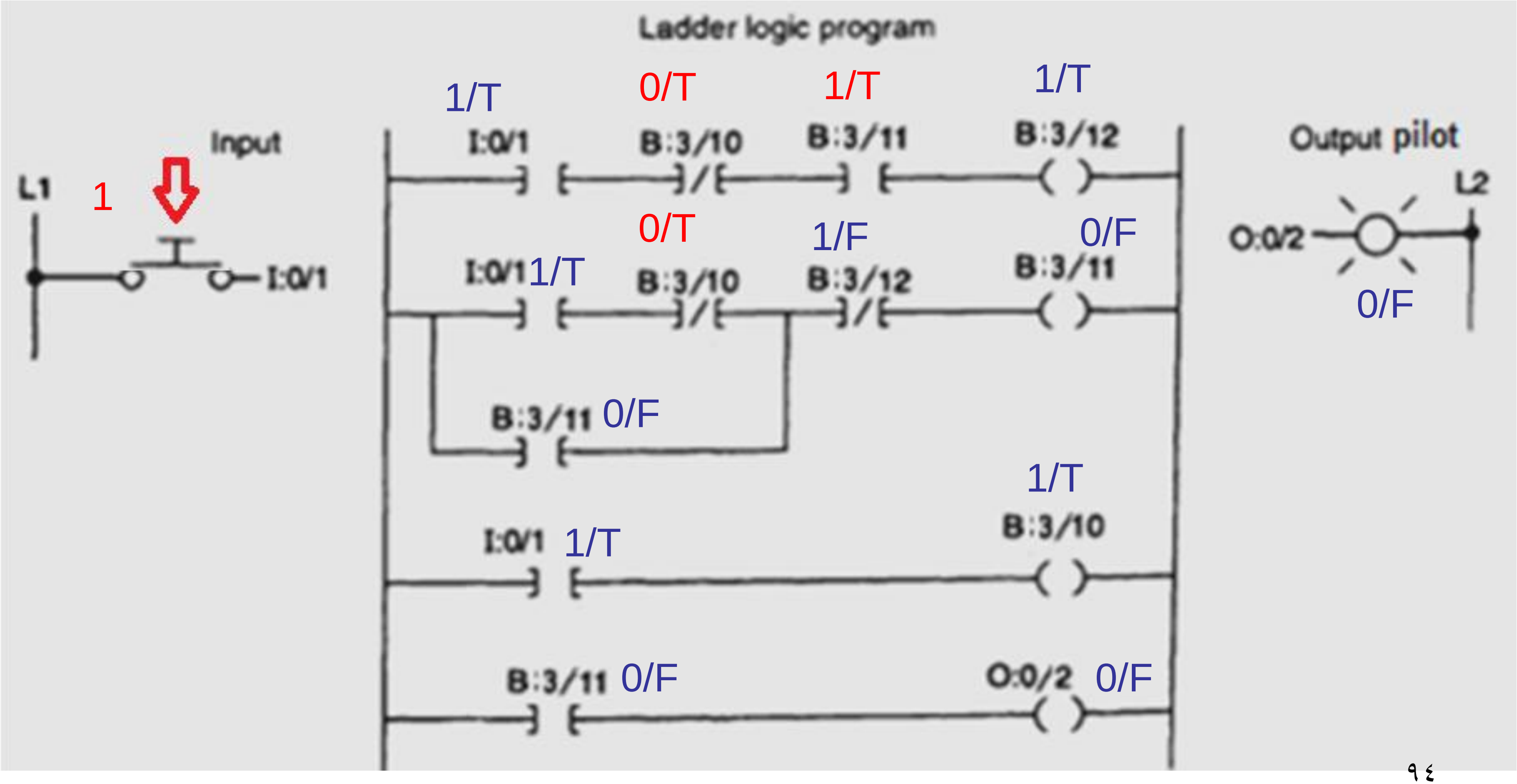


1) What will happened to the output pilot when the push button in the below ladder logic diagram is momentarily press for the **first** time? ON

Next scan

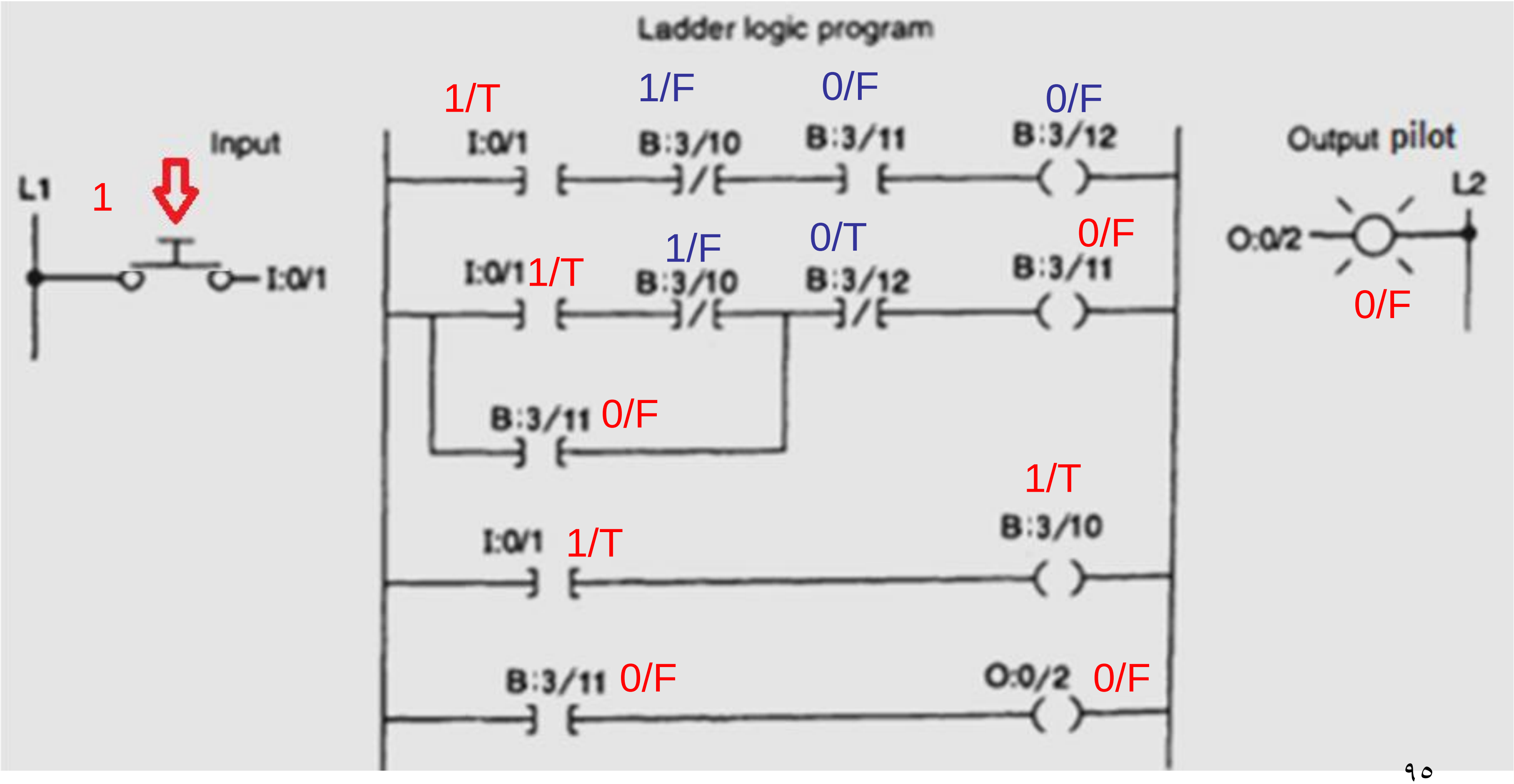


2) What will happened to the output pilot when the push button in the below ladder logic diagram is momentarily press for the **second** time?



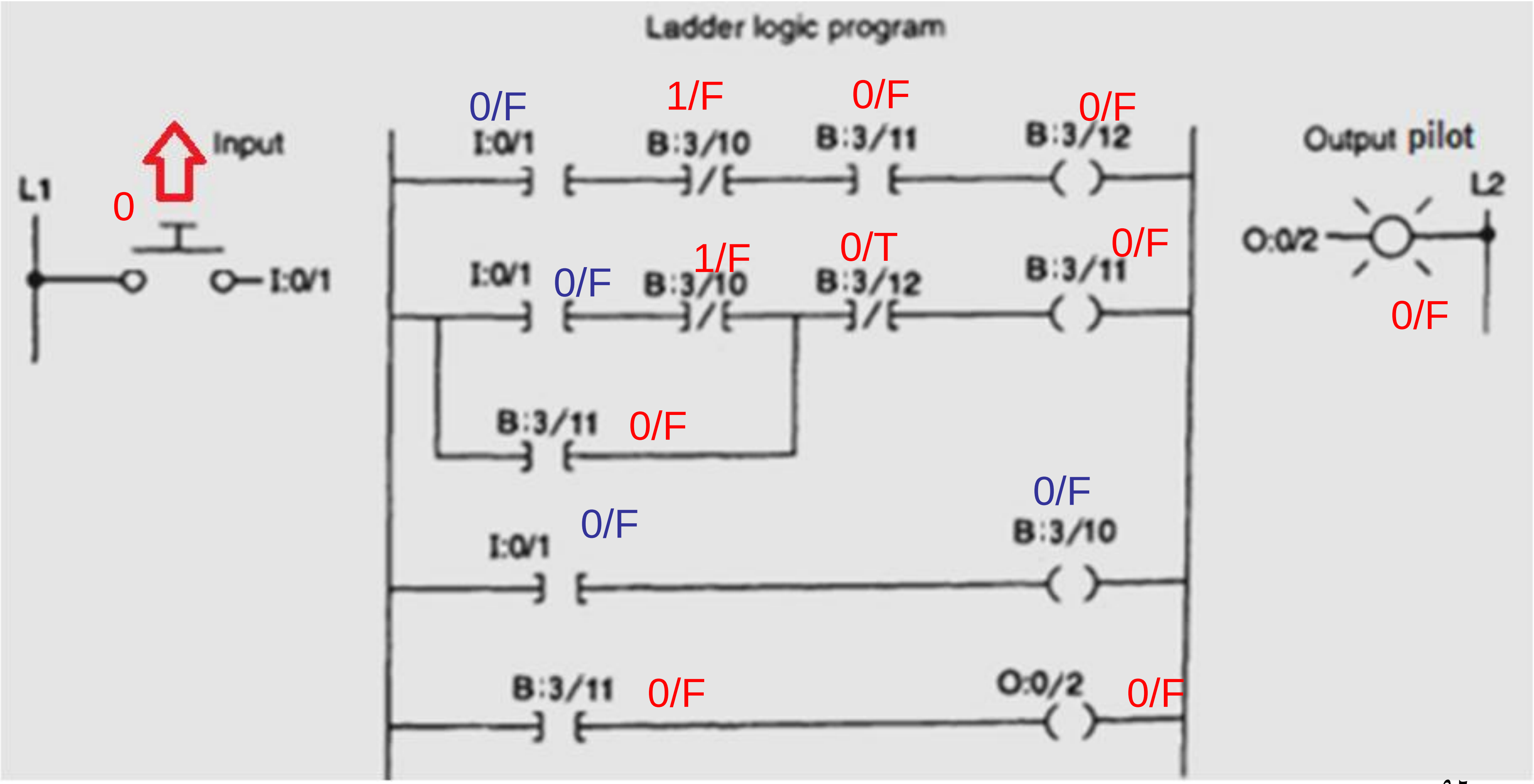
2) What will happened to the output pilot when the push button in the below ladder logic diagram is momentarily press for the **second** time?

Next scan



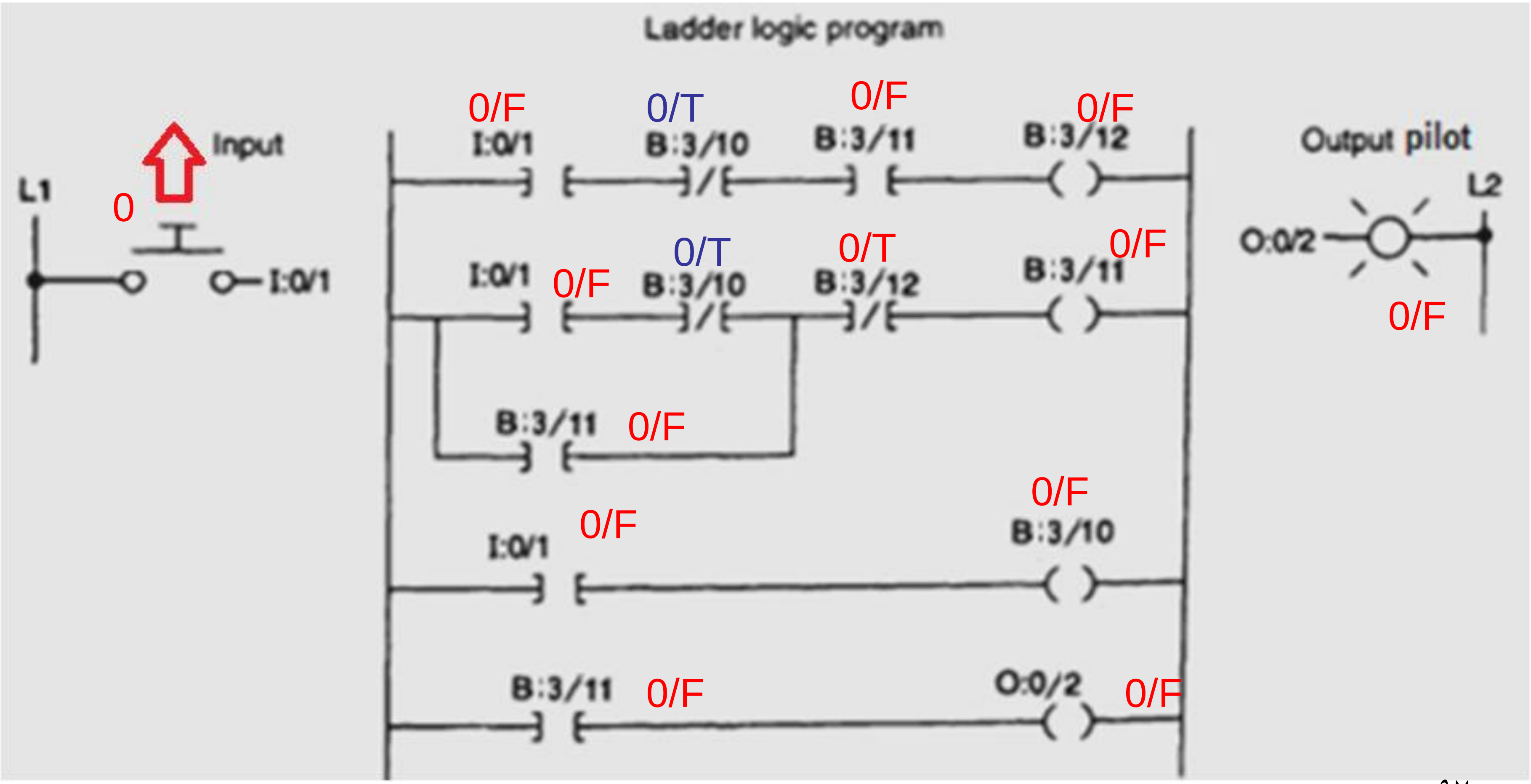
2) What will happened to the output pilot when the push button in the below ladder logic diagram is momentarily press for the **second** time?

Release push button



2) What will happened to the output pilot when the push button in the below ladder logic diagram is momentarily press for the **second** time? OFF

Next scan



10. A pump is to be used to fill two storage tanks. The pump is manually started by the operator from a START/STOP station. When the first tank is full, the control logic must be able to automatically stop flow to the first tank and direct flow to the second tank through the use of sensors and electric solenoid valves. When the second tank is full, the pump must shut down automatically. Indicator lamps are to be included to signal when each tank is full.

a. Draw a sketch of the process.

b. Prepare a typical PLC program for this control process.

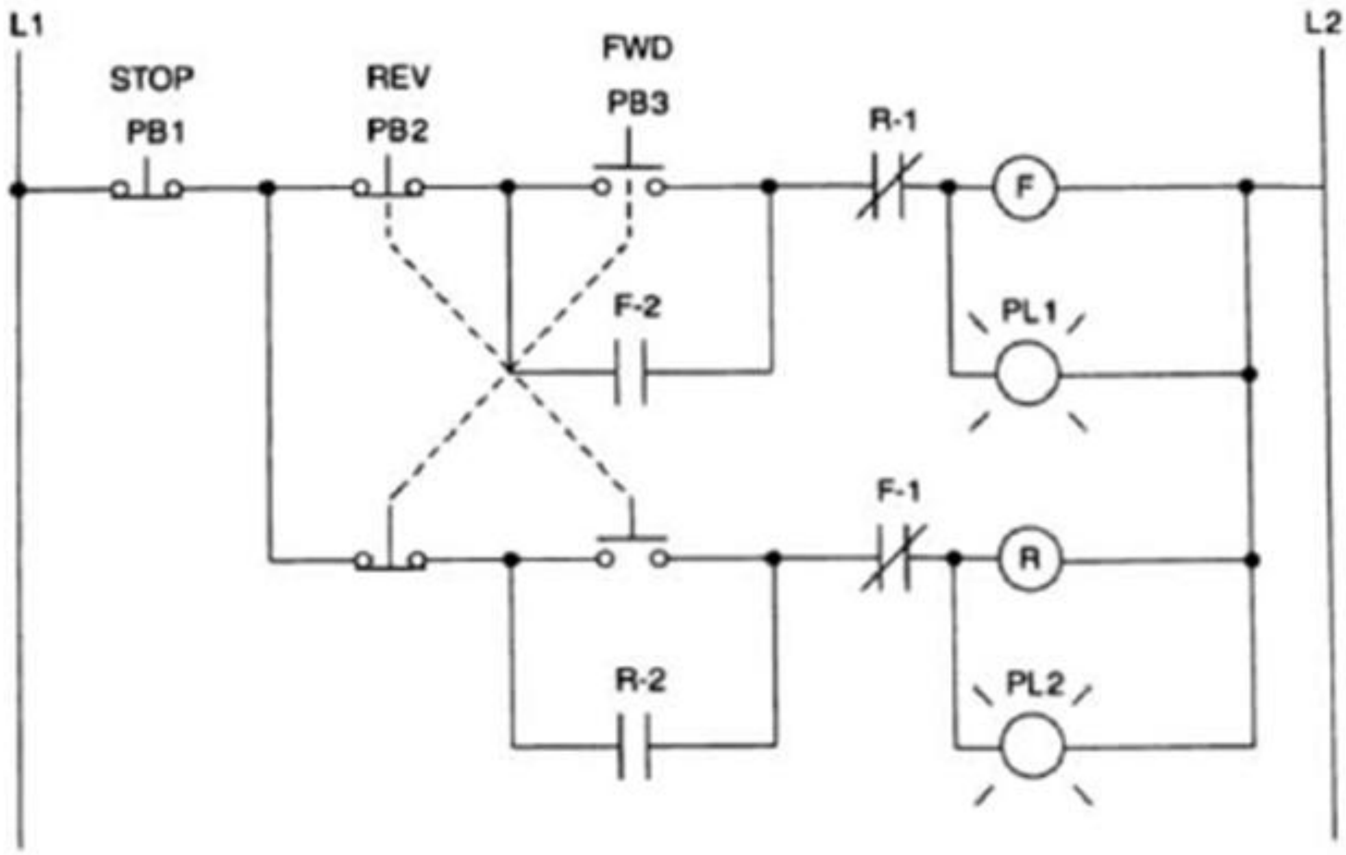
11. Write the optimum ladder logic rung for each of the following scenarios, and arrange the instructions for optimum performance:

a. If limit switches LS1 or LS2 or LS3 are on, or if LS5 and LS7 are on, turn on; otherwise, turn off. (Commonly, if LS5 and LS7 are on, the other conditions rarely occur.)

b. Turn on an output when switches SW6, SW7, and SW8 are all on, or when SW55 is on. (SW55 is an indication of an alarm state, so it is rarely on; SW7 is on most often, then SW8, then SW6.)

.

6. Design a PLC program and prepare a typical I/O connection diagram and ladder logic program that will correctly execute the hardwired control circuit in Figure 6-64.



references:

1-notes from Dr. **Jeff Jackson** *,the university of Alabama*

2-notes from Dr. Radu Muresan ,University of Guelph

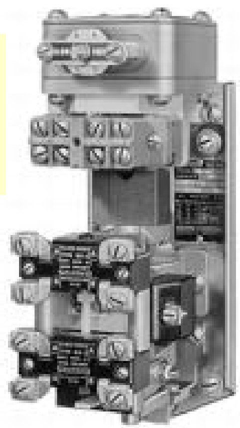
Timers

Radu Muresan

There are very few industrial control systems that do not need at least one or two timed functions. They are used to activate or de-activate a device after a preset interval of time.

Time delay relays and solid-state timers are used to provide a time delay. They may have displays, pots or other means of operator interface for time settings and electromechanical or solid state outputs.

Time Delay Relay

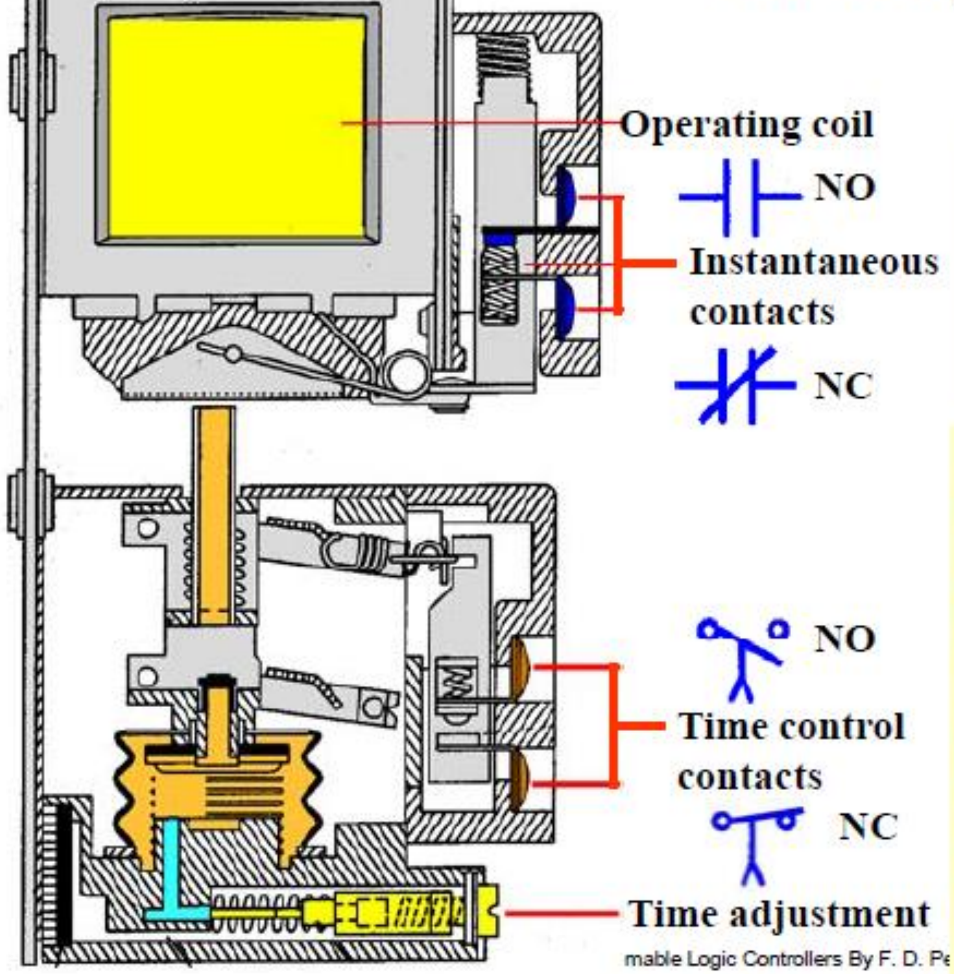


Solid-State Timer

Timer Uses

- The timing block functions are used with various contact arrangements and in multiples to accomplish various timing tasks.
- Typical industrial timing tasks include timing of the intervals for welding, painting, and heat treating.
- Timers can also predetermine the interval between two operations.
- With a PLC you can utilize as many timer blocks as you need, within the PLC memory limitations.

On-Delay Timing Relay

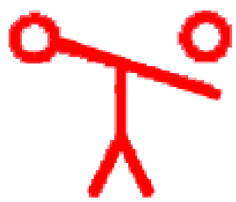


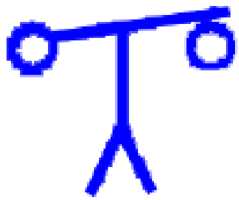
Nontimed contacts are controlled directly by the timer coil, as in a general-purpose control relay.

When the coil is energized, the timed contacts are prevented from opening or closing until the time delay period has elapsed. However, when the coil is de-energized, the timed contacts return instantaneously to their normal state.

Timed Contact Symbols

On-Delay Symbols





Normally open, timed closed contact (NOTC)

Contact is open when relay coil is de-energized

When relay is energized, there is a time delay in closing

Normally closed, timed open contact (NCTO)

Contact is closed when relay coil is de-energized

When relay is energized, there is a time delay in opening

Timed Contact Symbols

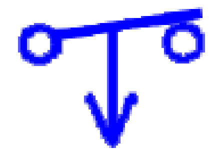
Off Delay Symbols

Normally open, timed open contacts (NOTO).

Contact is normally open when relay coil is de-energized.

When relay coil is energized, contact closes instantly.

When relay coil is de-energized, there is a time delay before the contact opens.



Normally closed, timed closed contacts (NCTC).

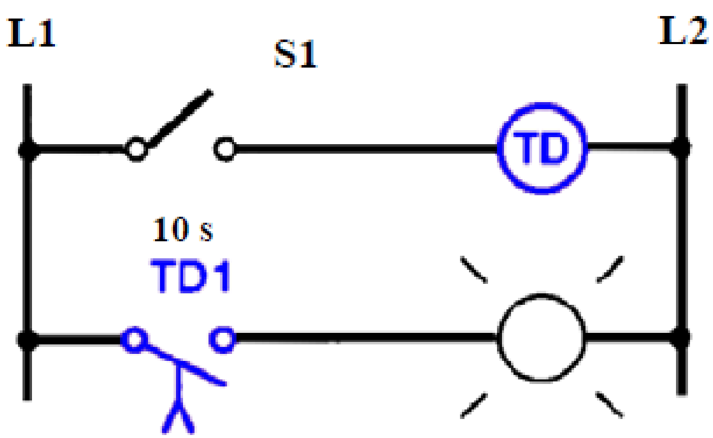
Contact is normally closed when relay coil is de-energized.

When relay coil is energized, contact opens instantly.

When relay coil is de-energized, there is a time delay before the contact closes.

On-Delay Relay Timer Circuit (NOTC Contact)

Radu Muresan



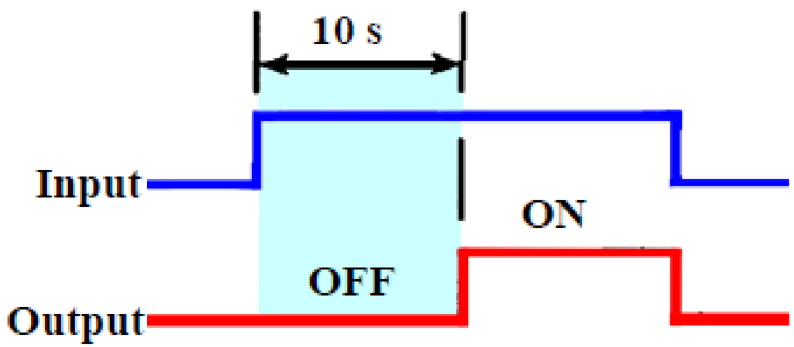
Sequence of operation

S1 open, TD de-energized, TD1 open, L1 is off.

S1 closes, TD energizes, timing period starts, TD1 still open, L1 is still off.

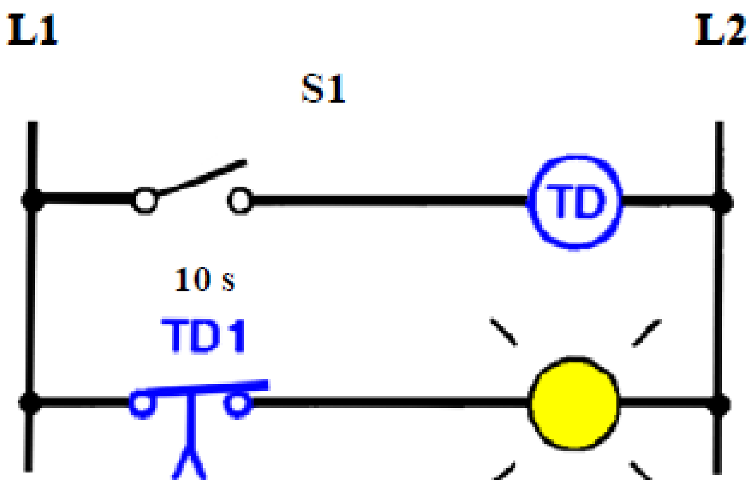
After 10 s, TD1 closes, L1 is switched on.

S1 is opened, TD de-energizes, TD1 opens instantly, L1 is switched off.



Timing Diagram

On-Delay Relay Timer Circuit (NCTO Contact)



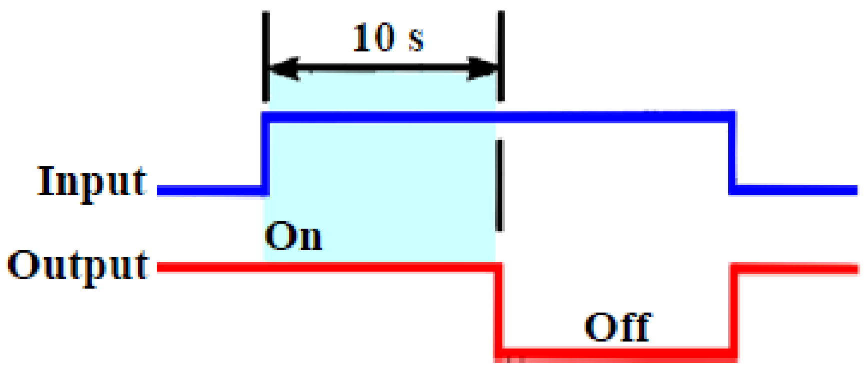
Sequence of operation

S1 open, TD de-energized, TD1 closed, L1 is on.

S1 closes, TD energizes, timing period starts, TD1 is still closed, L1 is still on.

After 10 s, TD1 opens, L1 is switched off.

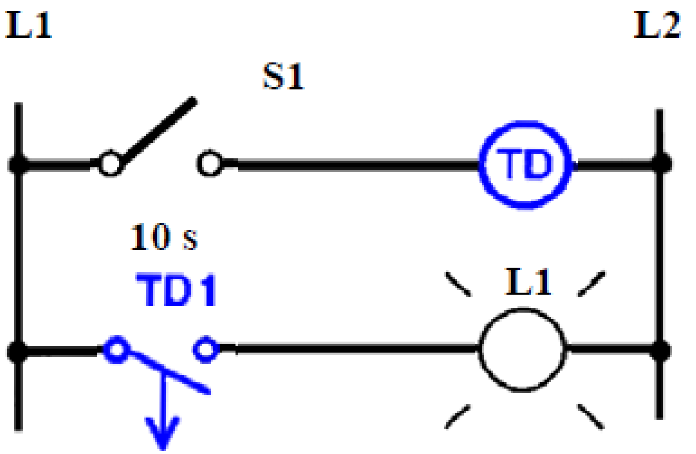
S1 is opened, TD de-energizes, TD1 closes instantly, L1 is switched on.



Timing Diagram

7. Adapted From Programmable Logic Controllers By F. D. Petruzella, McGraw-Hill

Off-Delay Relay Timer Circuit (NOTO Contact)



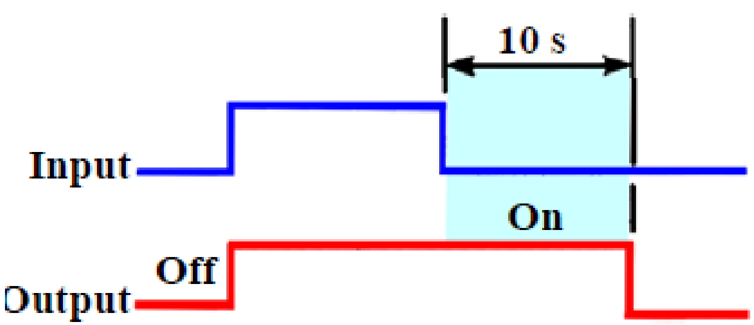
Sequence of operation

S1 open, TD de-energized, TD1 open, L1 is off.

S1 closes, TD energizes, TD1 closes instantly, L1 is switched on.

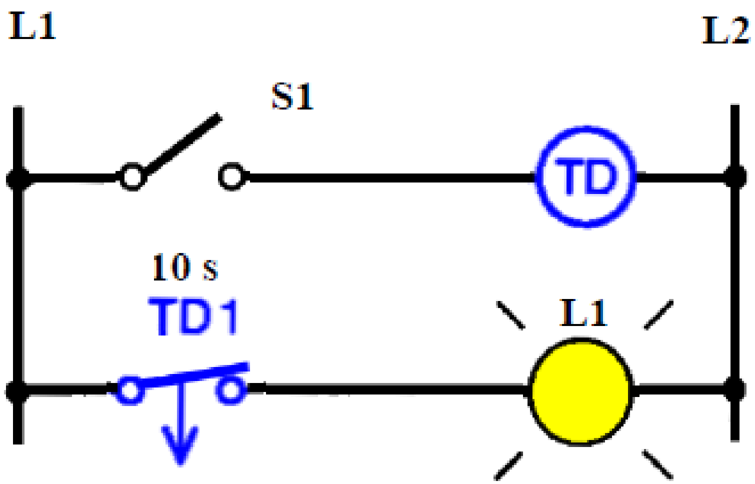
S1 is opened, TD de-energizes, timing period starts, TD1 is still closed, L1 is still on.

After 10 s, TD1 opens, L1 is switched off.



Timing Diagram

Off-Delay Relay Timer Circuit (NCTC Contact)



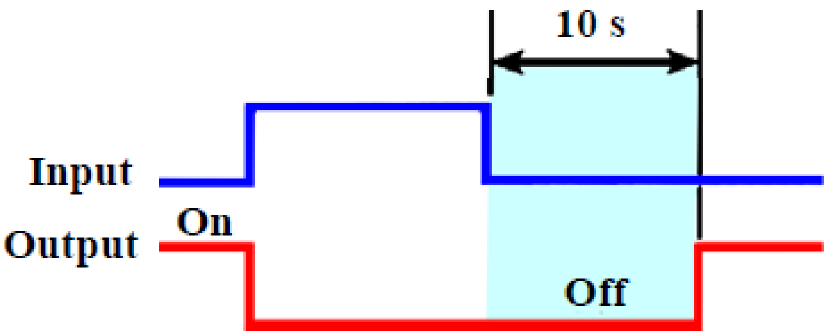
Sequence of operation

S1 open, TD de-energized, TD1 closed, L1 is on.

S1 closes, TD energizes, TD1 opens instantly, L1 is switched off.

S1 is opened, TD de-energized, timing period starts, TD1 is still open, L1 is still off.

After 10 s, TD1 closes, L1 is switched on.



Timing Diagram

Programmed Timer Instructions

PLC timers are output instructions that provide the same functions as timing relays and solid state timers.

Some advantages of PLC timers:

- their settings can be altered easily
- the number of PLC timers used can be increased or decreased by programming changes without wiring changes
- timer accuracy and repeatability are extremely high

Programmed timer command

(based on the Allen Bradley SLC-500 and its associated RSLogix software)

Command	Name	Description
TON	Timer On-Delay	Counts time base intervals when the instruction is "true"

RSLogic Timer Commands

Command	Name	Description
TOF	Timer Off-Delay	Counts time base intervals when the instruction is "false"

RSLogic Timer Commands

Command	Name	Description
RTO	Retentive Timer ON	Counts time base intervals when the instruction is "true" and retains the accumulated value when the instruction goes "false" or when power cycle occurs
RES	Reset	When this instruction is "true" it resets the count of the RTO counter

Quantities Associated with the Timer Instruction

Preset Time – Represents the time duration of the timing circuit. For example, if a time delay of 10 s is required, the timer will have a preset of 10 s.

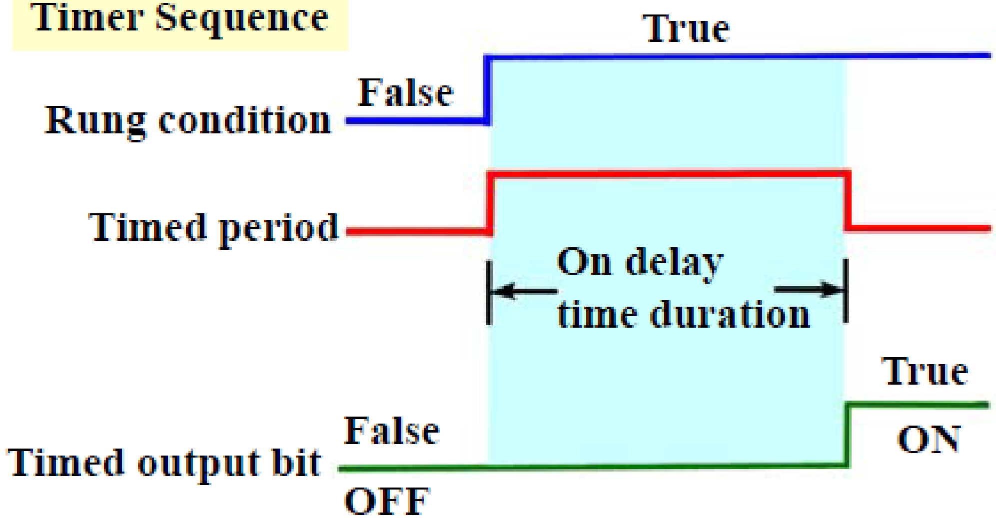
Accumulated Time – Represents the amount of time that has elapsed from the moment the timing coil became energized.

Time Base – Timers can typically be programmed with several different time bases: 1 s, 0.1 s, and 0.01 s are typical time bases. For example, if you enter 0.1 for the time base and 50 for the preset time the timer would have a 5 s delay (50 x 0.1 s = 5 s).

On-Delay Timer Instruction

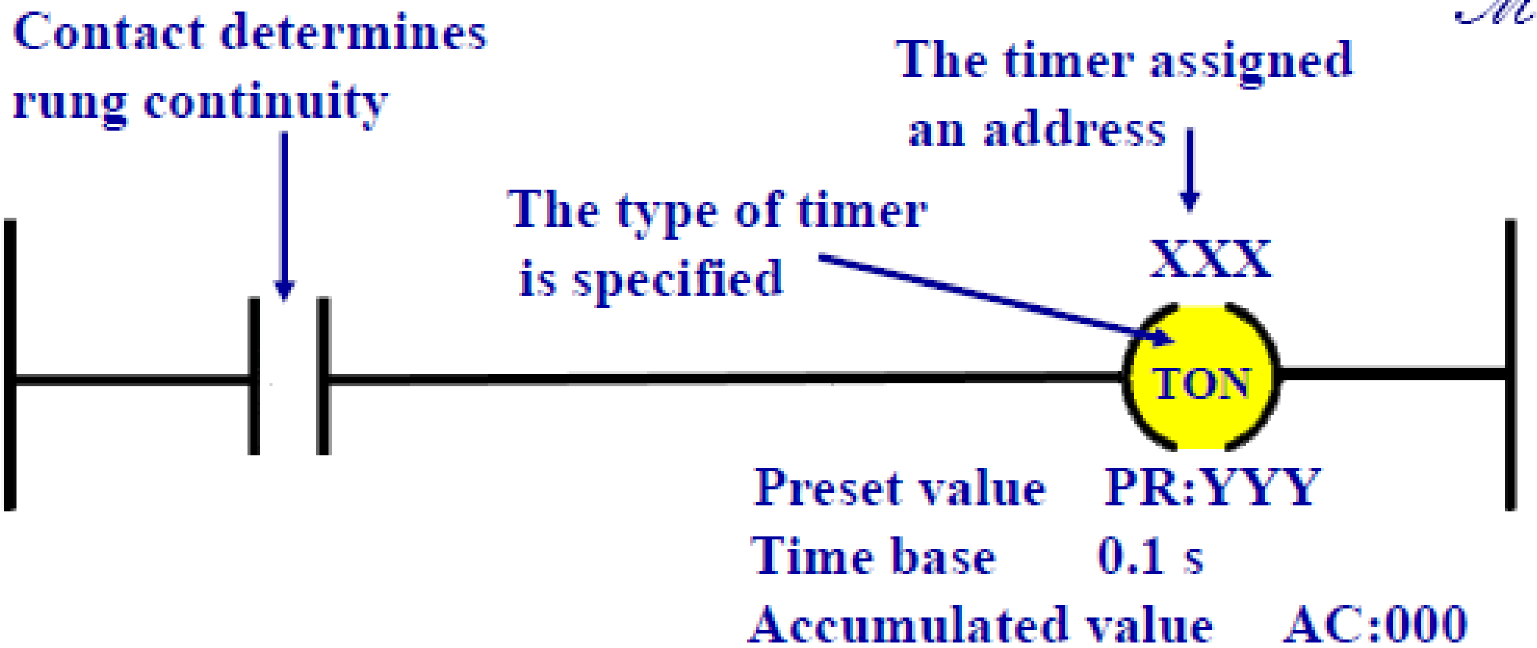
The *on-delay timer* operates so that, when the rung containing the timer is true, the timer time-out period commences.

Timer Sequence



The timed output becomes true sometimes after the timer rung becomes true; hence the timer is said to have an on delay.

Coil-Formatted Timer Instruction

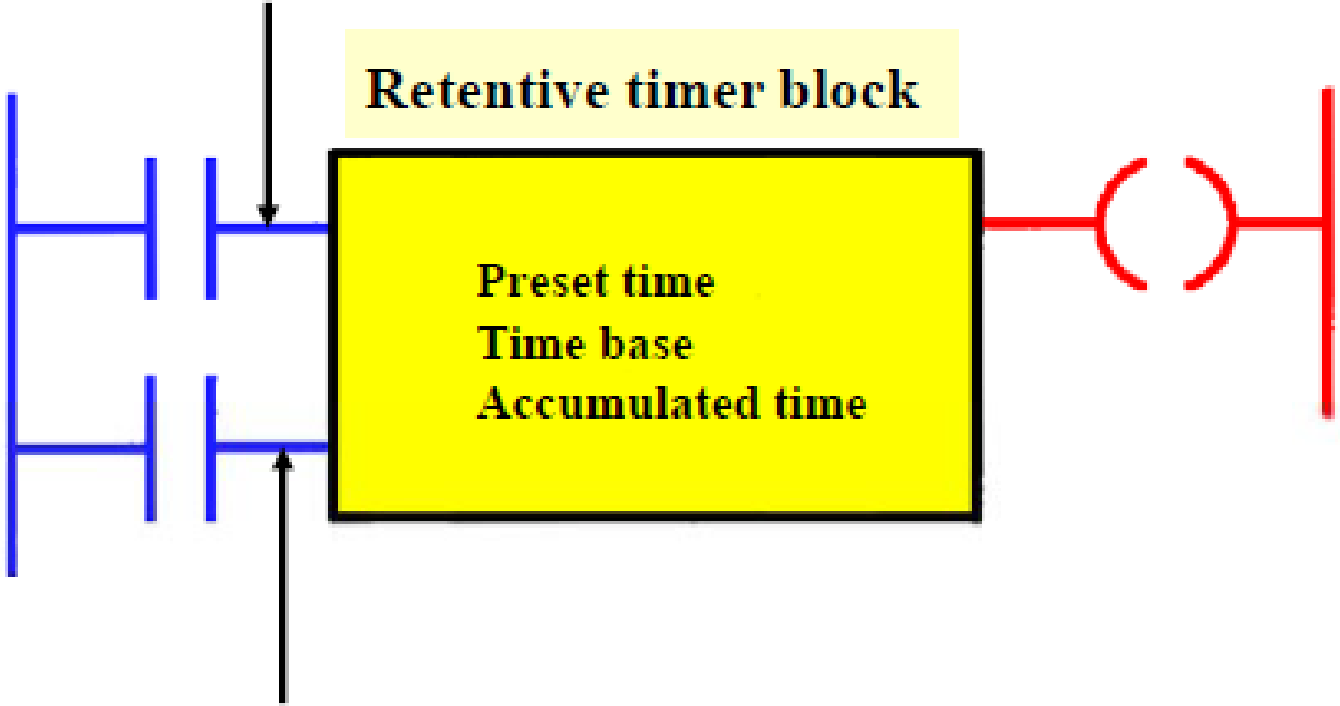


When the timer rung has logic continuity, the timer's accumulated value increases. When accumulated value equals the preset value, the output is energized and and the timed output contact associated with the output is closed. The timed contact can be used as many times as you wish throughout the program as a NO or NC contact.

Generic Block-Formatted Timer Instruction

Timers are most often represented by boxes in a ladder logic.

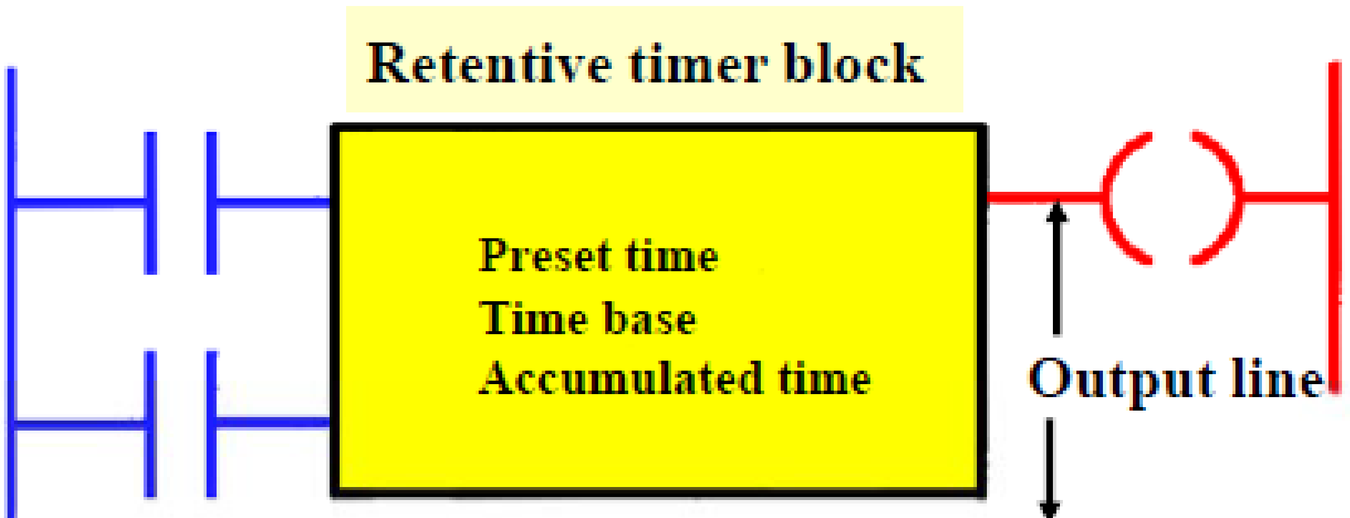
Control line controls the actual timing operation of the timer. Whenever this line is true the timer will time.



Reset line resets the the timer's accumulated value to zero.

Generic Block-Formatted Timer Instruction

Timers are most often represented by boxes in a ladder logic.



The timer continuously compares its accumulated time with its preset time. Its output is logic 0 as long as the accumulated time is less than the preset time. When the two become equal the output changes to logic 1.

Allen-Bradley On-Delay Timer Instruction

Radu Muresan

Allen-Bradley PLC-5 and SLC-500 controller timer elements each take three data table words: *the control word, preset word, and accumulated word.*

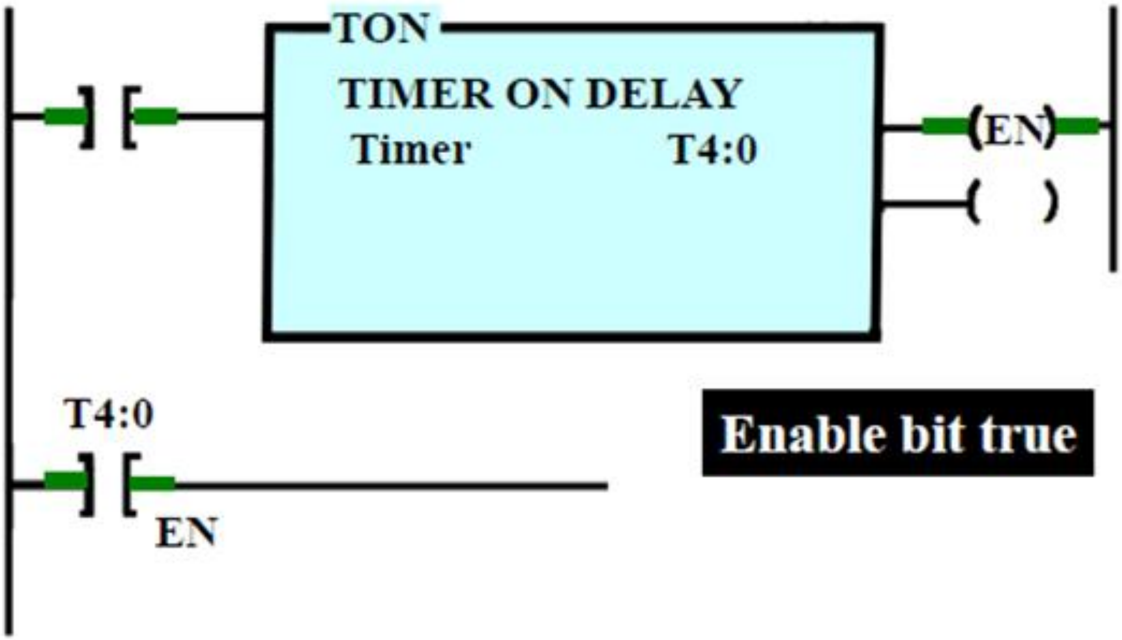
The control word uses three control bits: Enable (EN) bit, Timer-Timing (TT) bit, and Done-Bit (DN).

TIMER TABLE

	/EN	/TT	/DN
T4:0	0	0	0

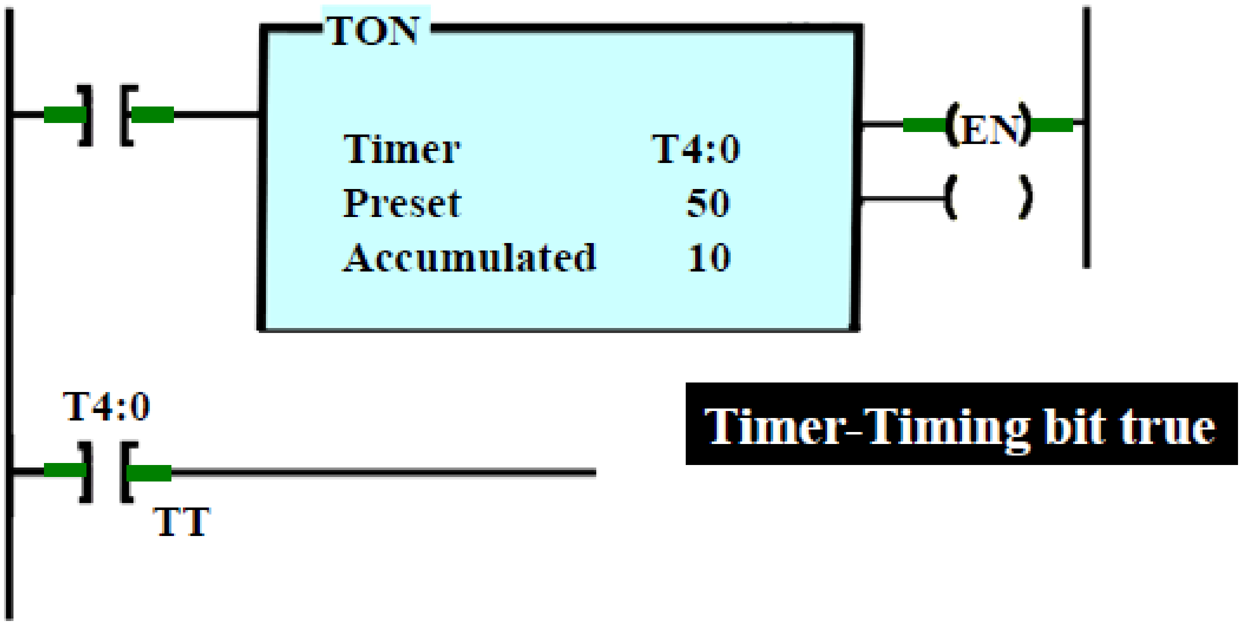
Allen-Bradley On-Delay Timer Instruction

The Enable (EN) bit is true (has a status of 1) whenever the timer instruction is true. When the timer instruction is false, the enable bit is false (has a status of 0)



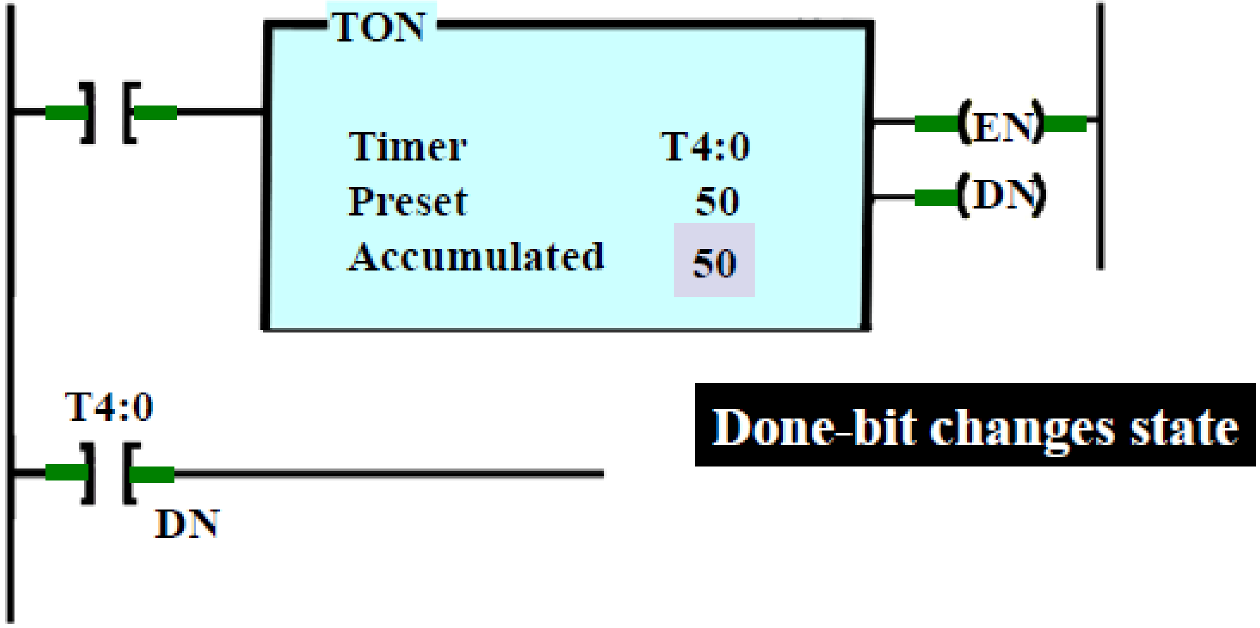
Allen-Bradley On-Delay Timer Instruction

The Timer-Timing (TT) bit is true whenever the accumulated value of the timer is changing, which means the timer is timing.



Allen-Bradley On-Delay Timer Instruction

The Done-Bit (DN) changes state whenever the accumulated value reaches the preset value. Its state depends on the type of timer being used.



Allen-Bradley On-Delay Timer Instruction

The preset value *(PRE) word* is the set point of the timer, that is, the value up to which the timer will time.

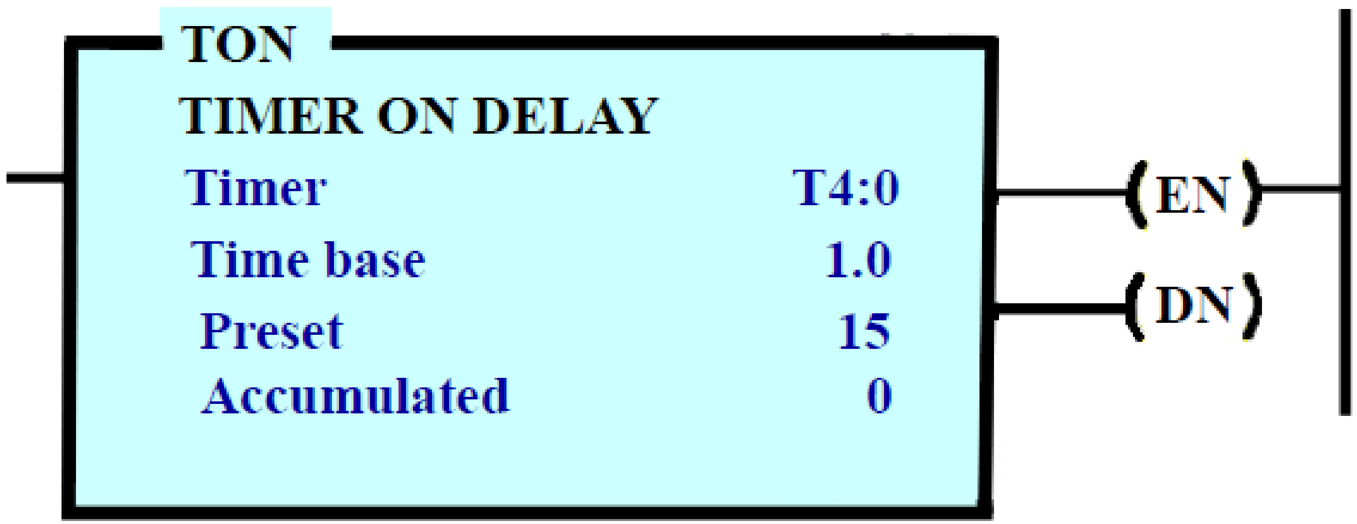
The accumulated value *(ACC) word* is the value that increments as the timer is timing. The accumulated value will stop incrementing when its value reaches the preset value.

TIMER TABLE

	/EN	/TT	/DN	.PRE	.ACC
T4:0	0	0	0	0	0

Allen-Bradley On-Delay Timer Instruction

The information to be entered includes:

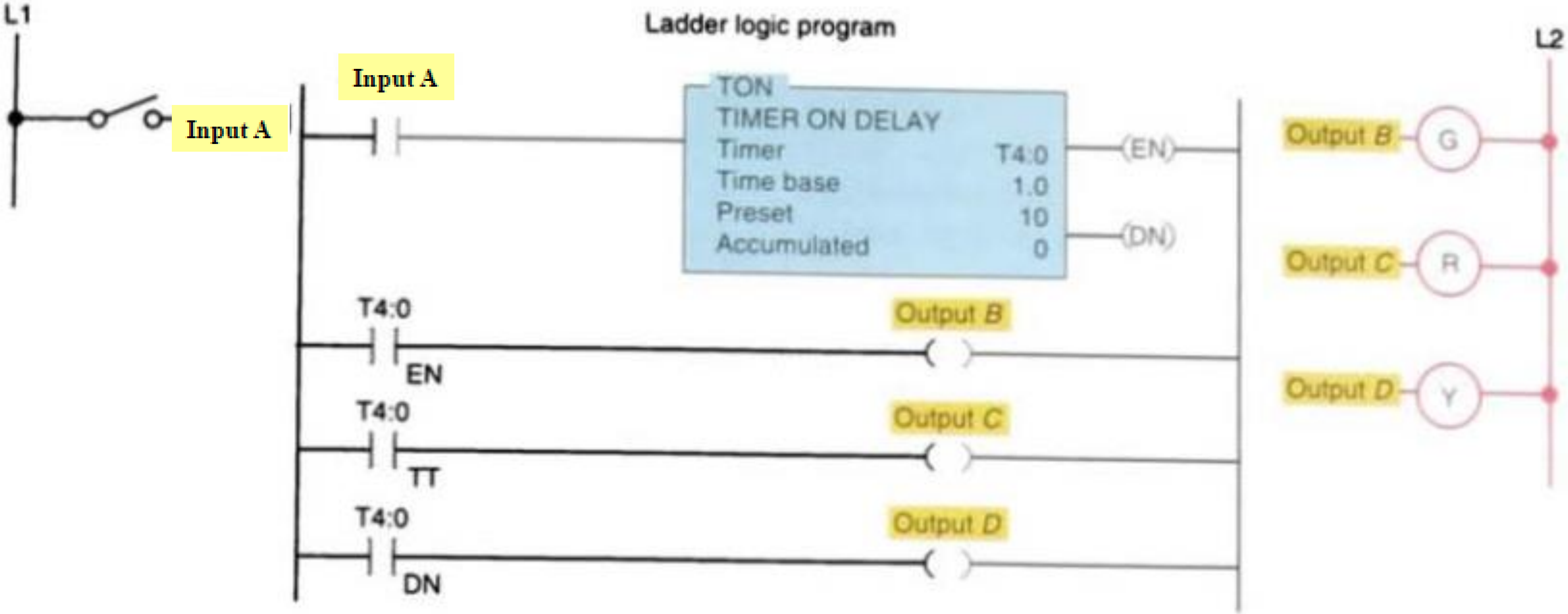


Timer number which must come from the timer file.

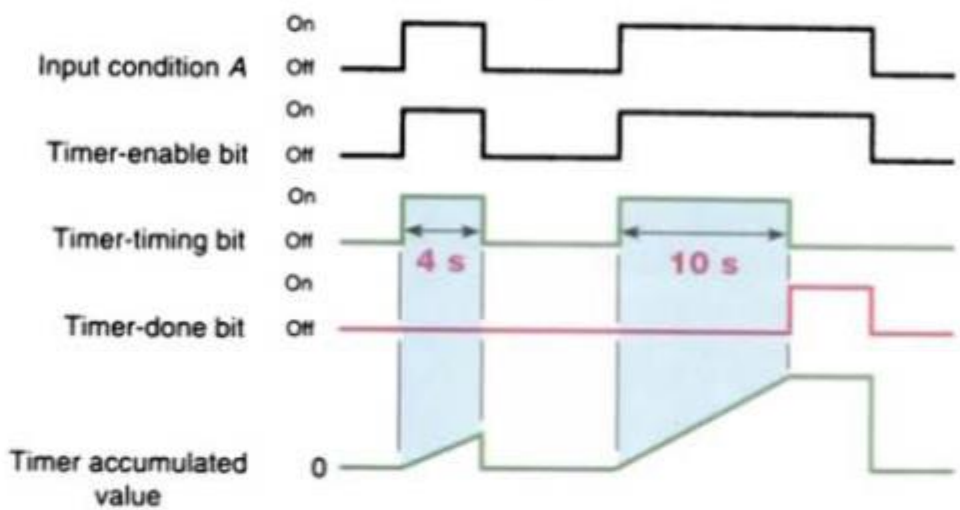
Time base which is expressed in seconds.

Preset value which is the length of the time delay.

Accumulated value which is normally entered as 0.



(a) Ladder diagram



(b) Timing diagram

On-Delay Timer Program

Timers are 3-word elements

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	Word
EN TT DN Internal use	0
Preset value PRE	1
Accumulated value ACC	2

Word 0 is the control word

Word 1 stores the preset value

Word 2 stores the accumulated value

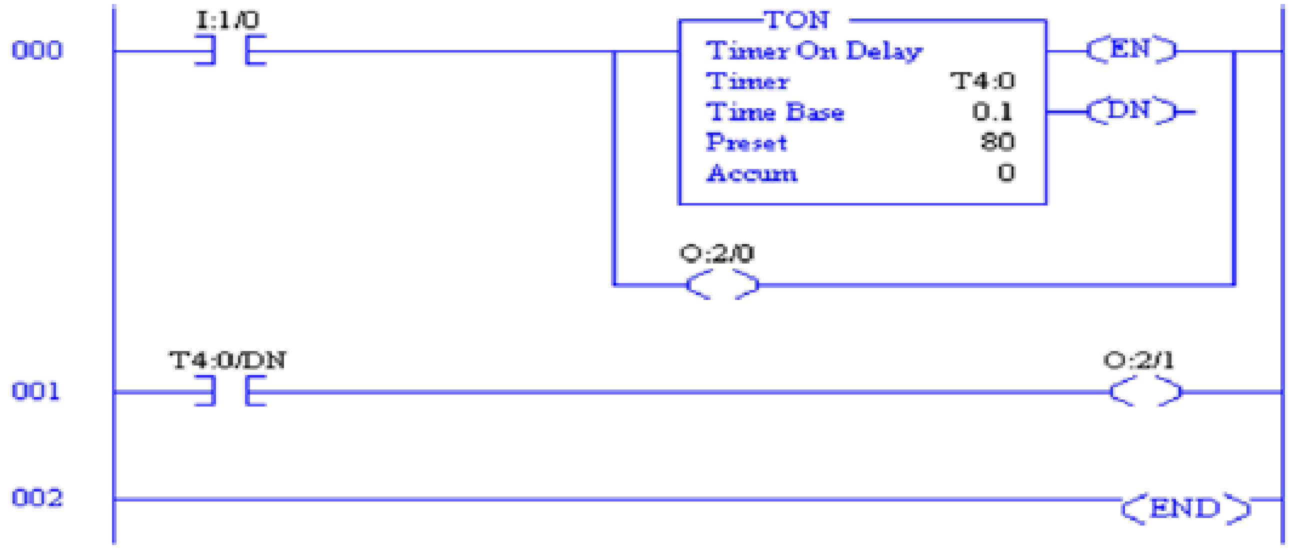
On Delay Timer Function

- The first example is the simplest form of time delay.
  - When the circuit is turned on, one action takes place.
  - A specified time later, another action occurs.
  - O:2/1 energizes exactly 8 seconds after O:2/0 energizes, provided I:1/0 remains energized

Solution to be added later

On Delay Timer Function

- The first example is the simplest form of time delay.
  - When the circuit is turned on, one action takes place.
  - A specified time later, another action occurs.
  - O:2/1 energizes exactly 8 seconds after O:2/0 energizes, provided I:1/0 remains energized



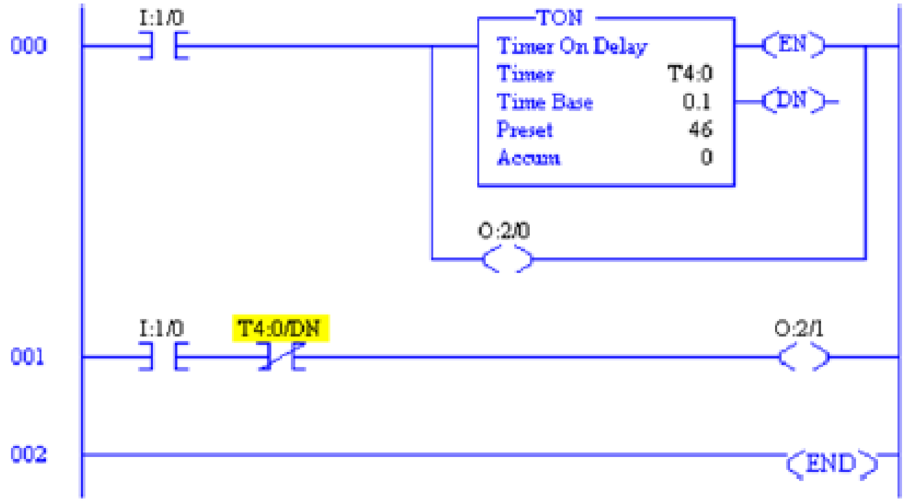
Limited On-Time Timer Function

- This example illustrates a situation in which two outputs go on at the same time.
- Then, one of them is to go off after a preset period of time.
- One output, O:2/0, stays on; the other output, O:2/1, turns off at the end of the timing interval.
- Resetting is accomplished by turning I:1/0 off.

Solution to be added later

Limited On-Time Timer Function

- This example illustrates a situation in which two outputs go on at the same time.
- Then, one of them is to go off after a preset period of time.
- One output, O:2/0, stays on; the other output, O:2/1, turns off at the end of the timing interval.
- Resetting is accomplished by turning I:1/0 off.

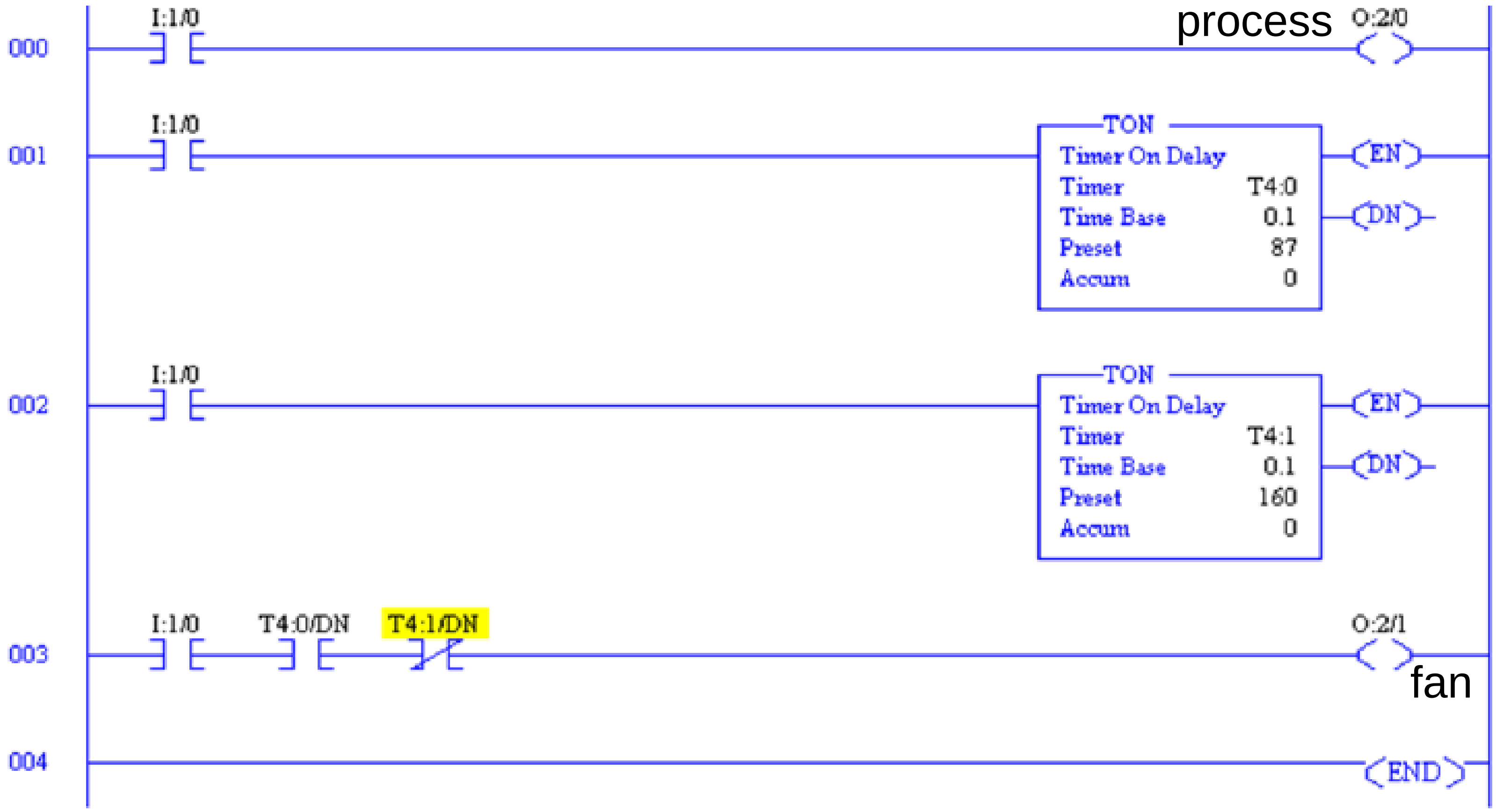


Interval Time Within a Cycle

- This example is for a timed interval of a number of seconds after the start of a process operation.
  - This time interval is sometimes called an *embedded time interval*.
  - A fan is to come on 8.7 seconds after a system is turned on.
  - It is then to run until 16 seconds after the system is turned on, which is a net time of 7.3 seconds.

Solution to be added later

Interval Time Example

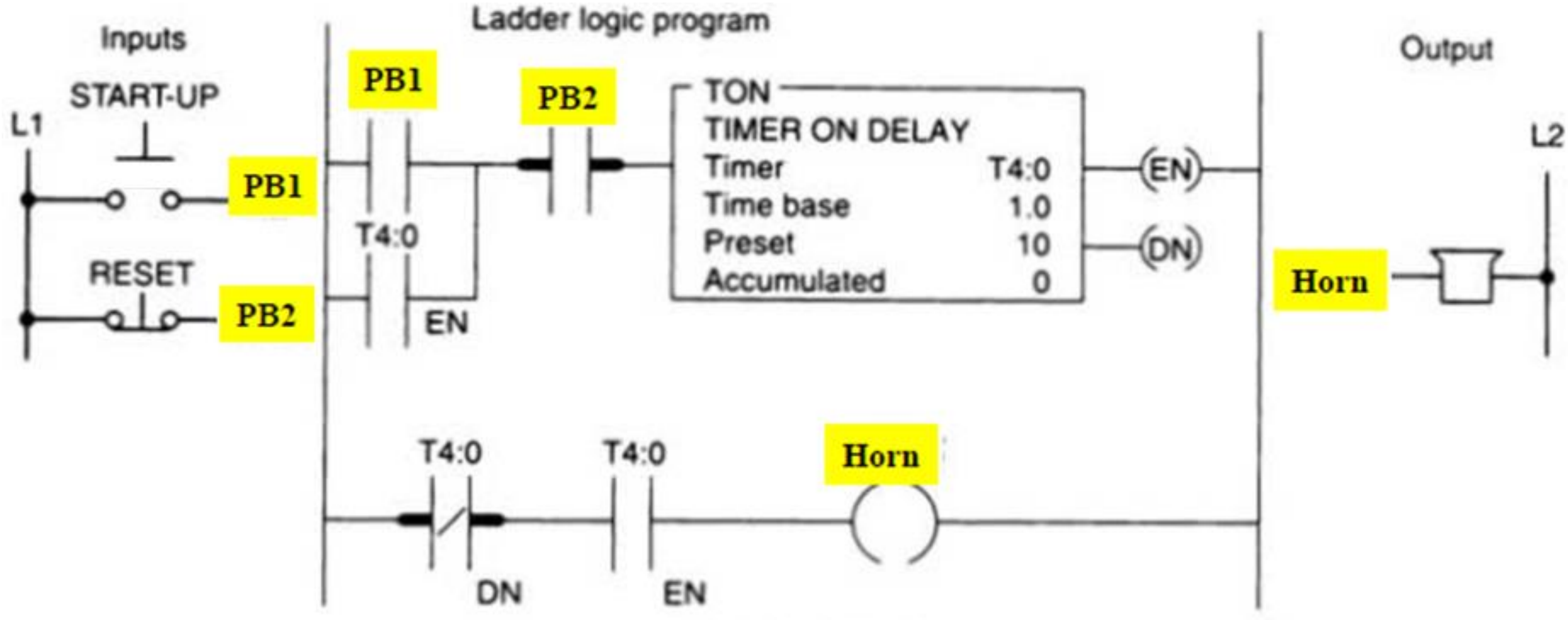


Start-Up Warning Signal Circuit

Relay Ladder Schematic Diagram

Convert to PLC ladder program?

Solution to be added later



(b) Ladder logic

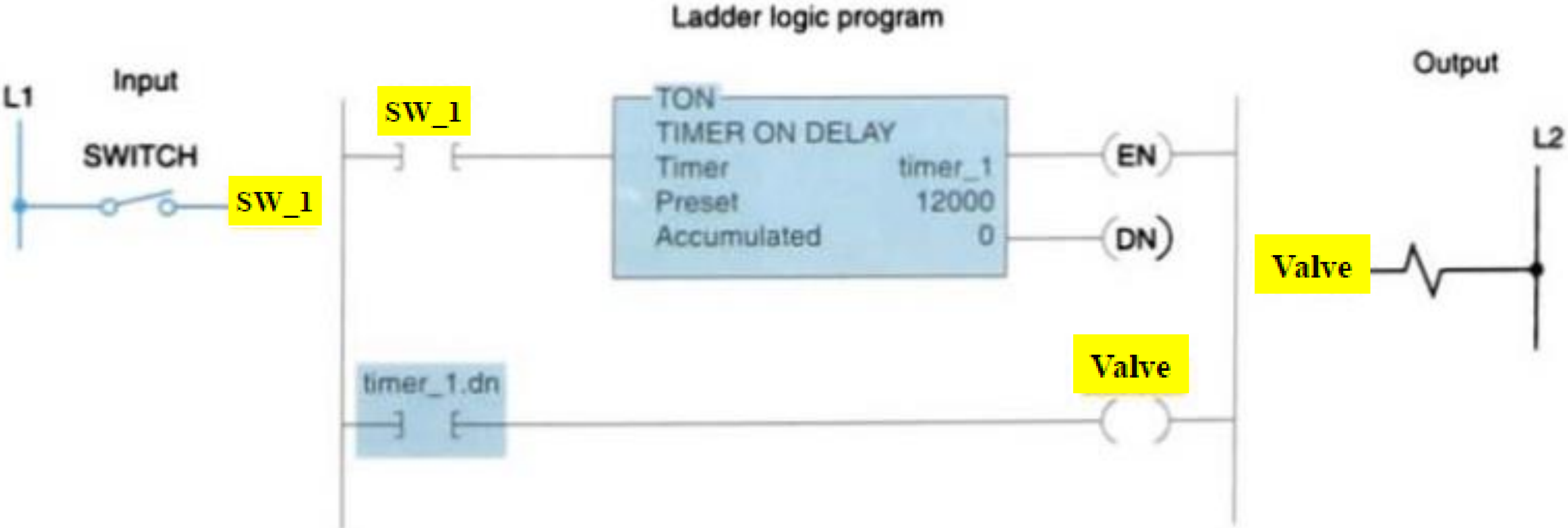


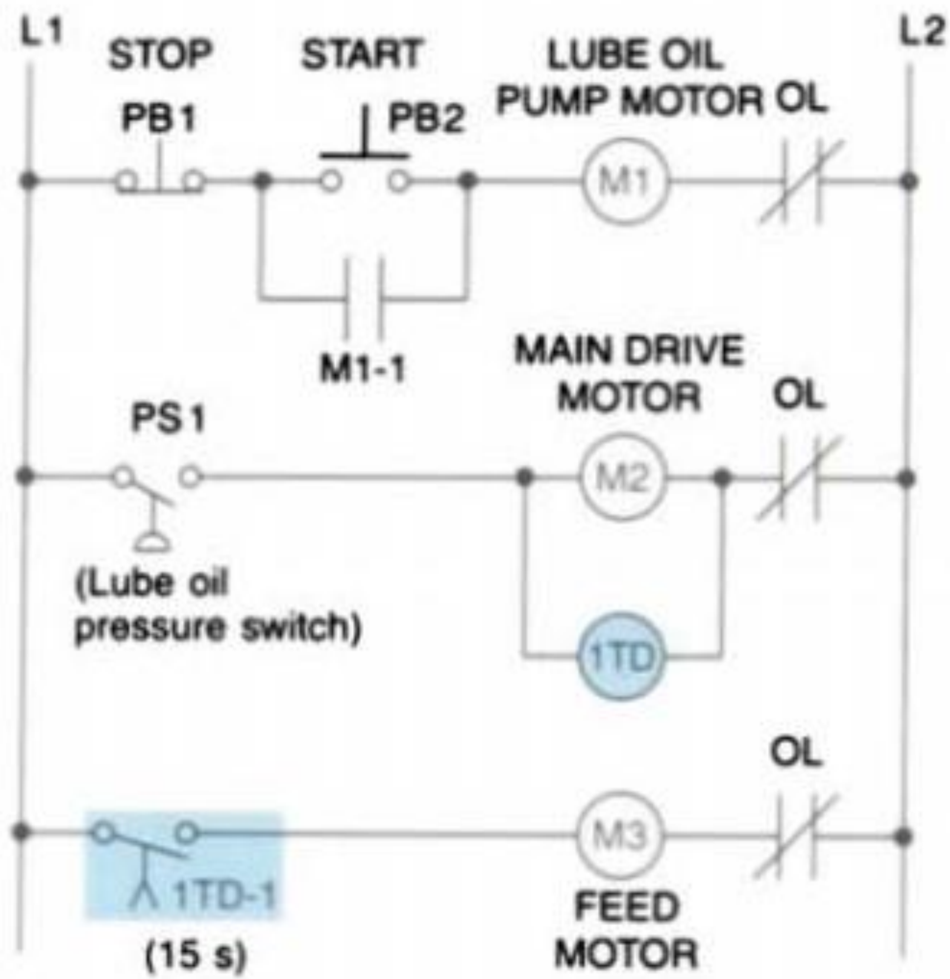
FIGURE 7-15 Program for a solenoid valve to be time-closed using the ControlLogix TON timer.

Allen-Bradley ControlLogix controller timers function in the same manner as PLC-5 and SLC-500 controllers. The differences occur in the time base

the time base is fixed at 1 ms (0.001 s).

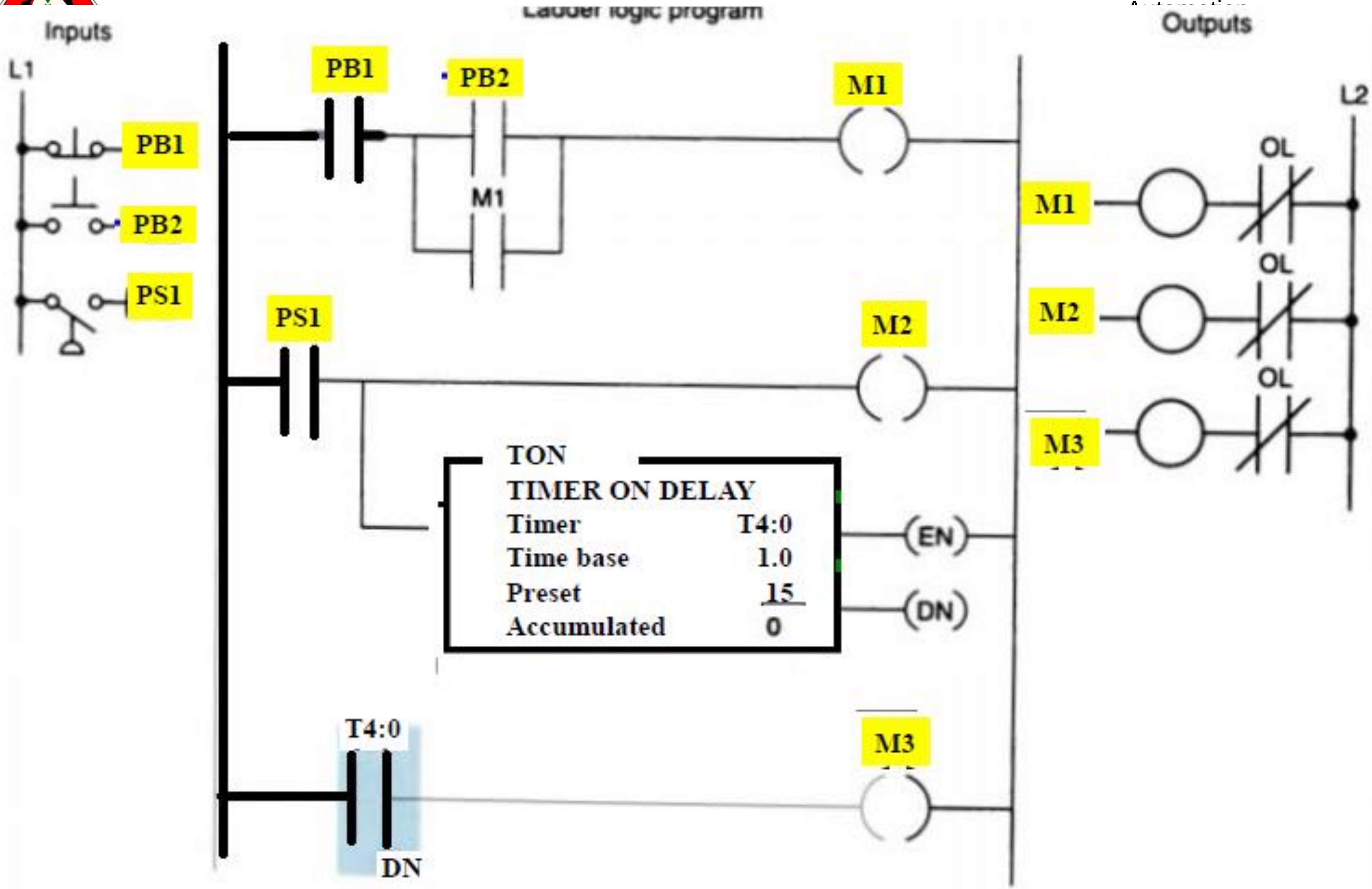
Convert to PLC ladder program?

Timers are often used as part of automatic sequential control systems. Figure 7-16 shows how a series of motors can be started automatically with only one start/stop control station. According to the relay ladder schematic, lube-oil pump motor starter coil M1 is energized when the start pushbutton PB2 is momentarily actuated. As a result, M1-1 control contact closes to seal in M1, and the lube-oil pump motor starts. When the lube-oil pump builds up sufficient oil pressure, the lube-oil pressure switch PS1 closes. This in turn energizes coil M2 to start the main drive motor and energizes coil 1TD to begin the time-delay period. After the preset time-delay period of 15 s, 1TD-1 contact closes to energize coil M3 and start the feed motor.



(a) Relay ladder schematic diagram

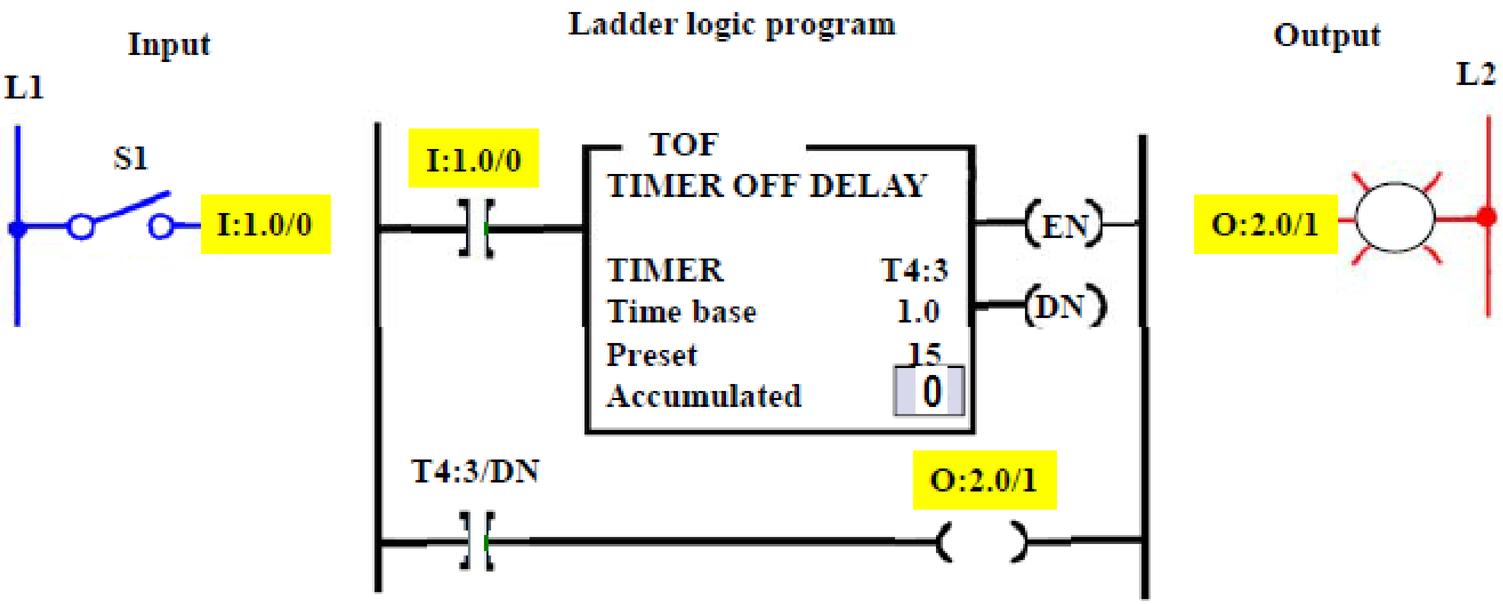
Solution to be added later

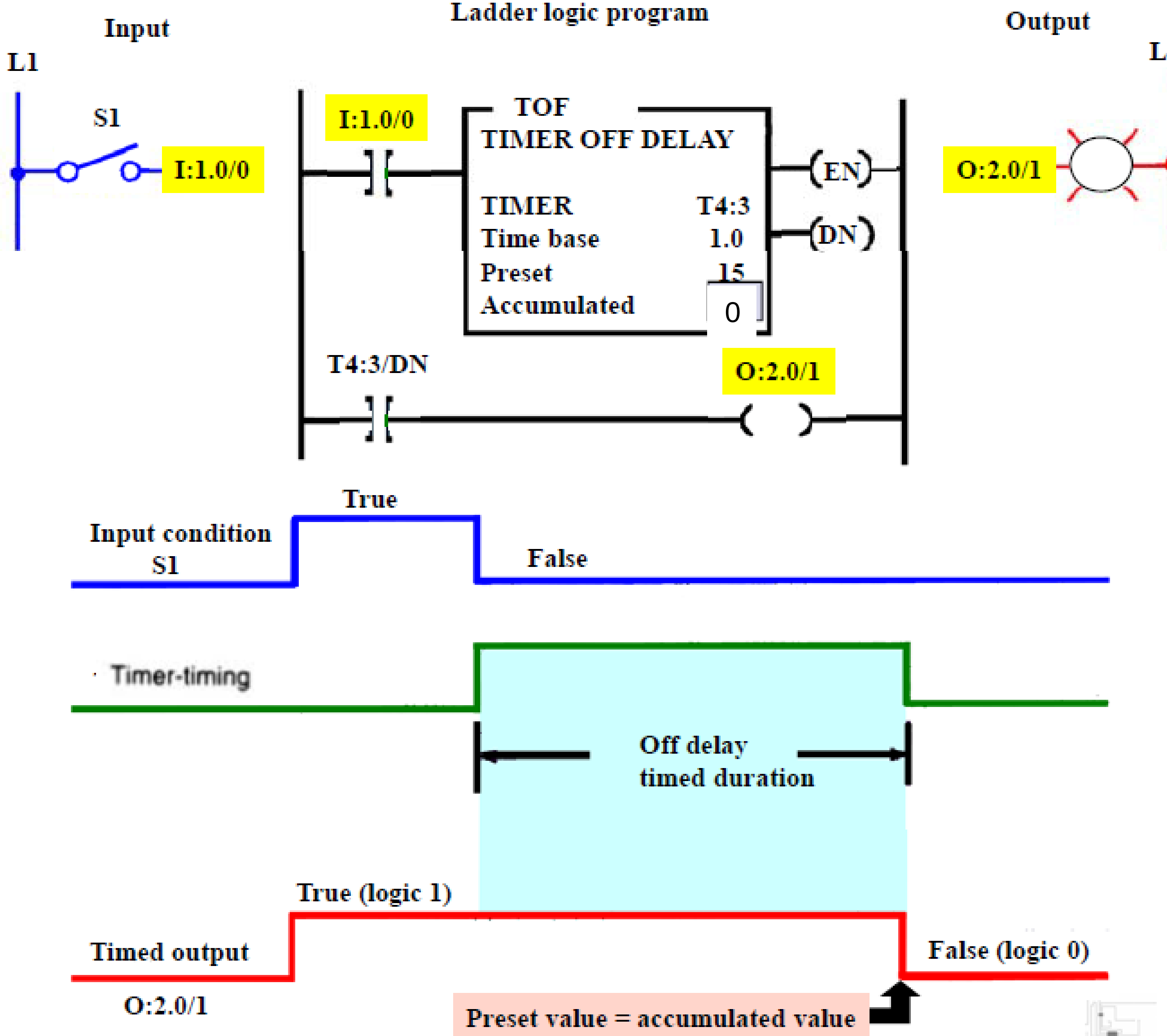


(b) Ladder logic

Off-Delay Programmed Timer

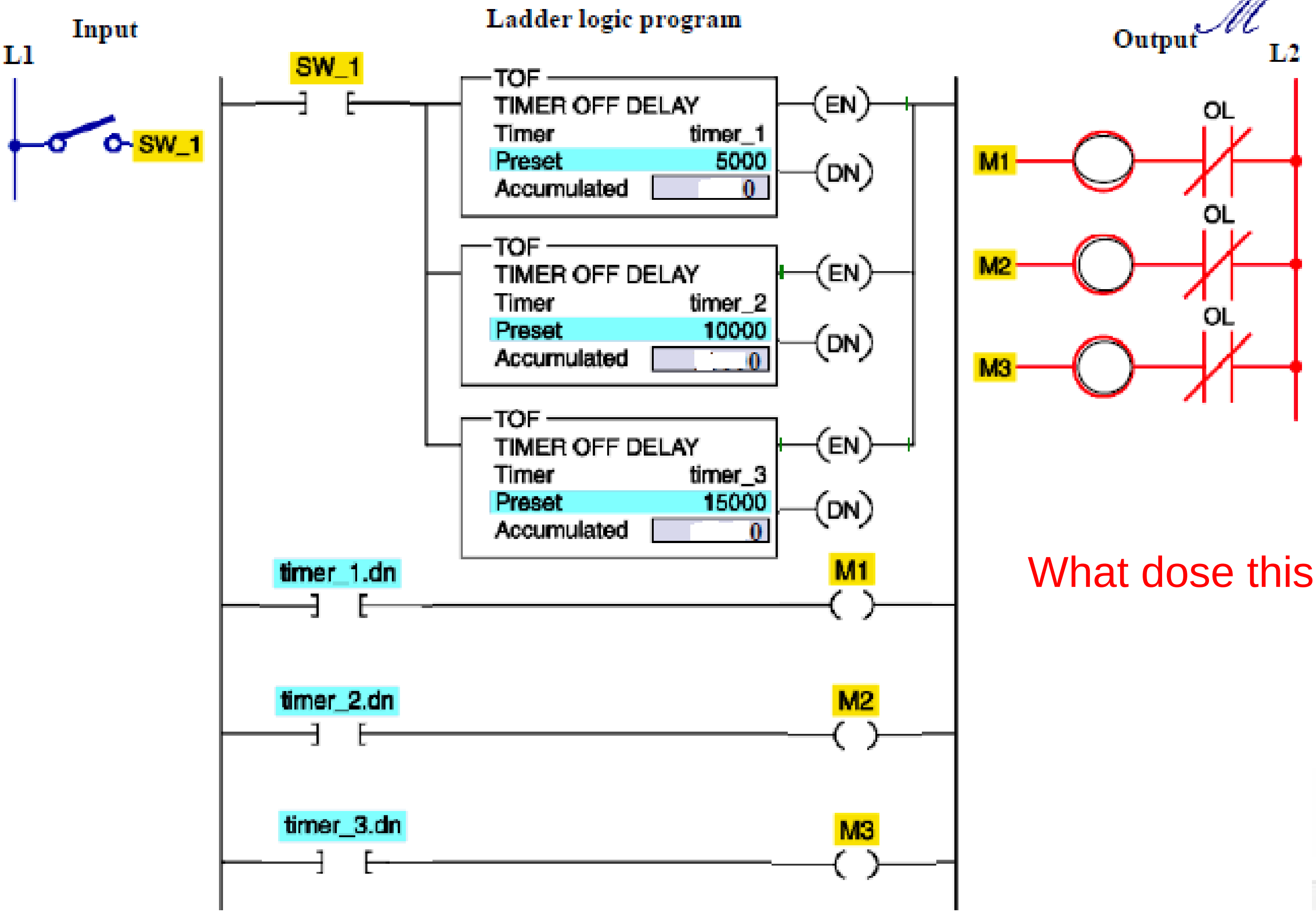
The *off-delay timer (TOF)* operation will keep the output energized for a period after the rung containing the timer has gone false.





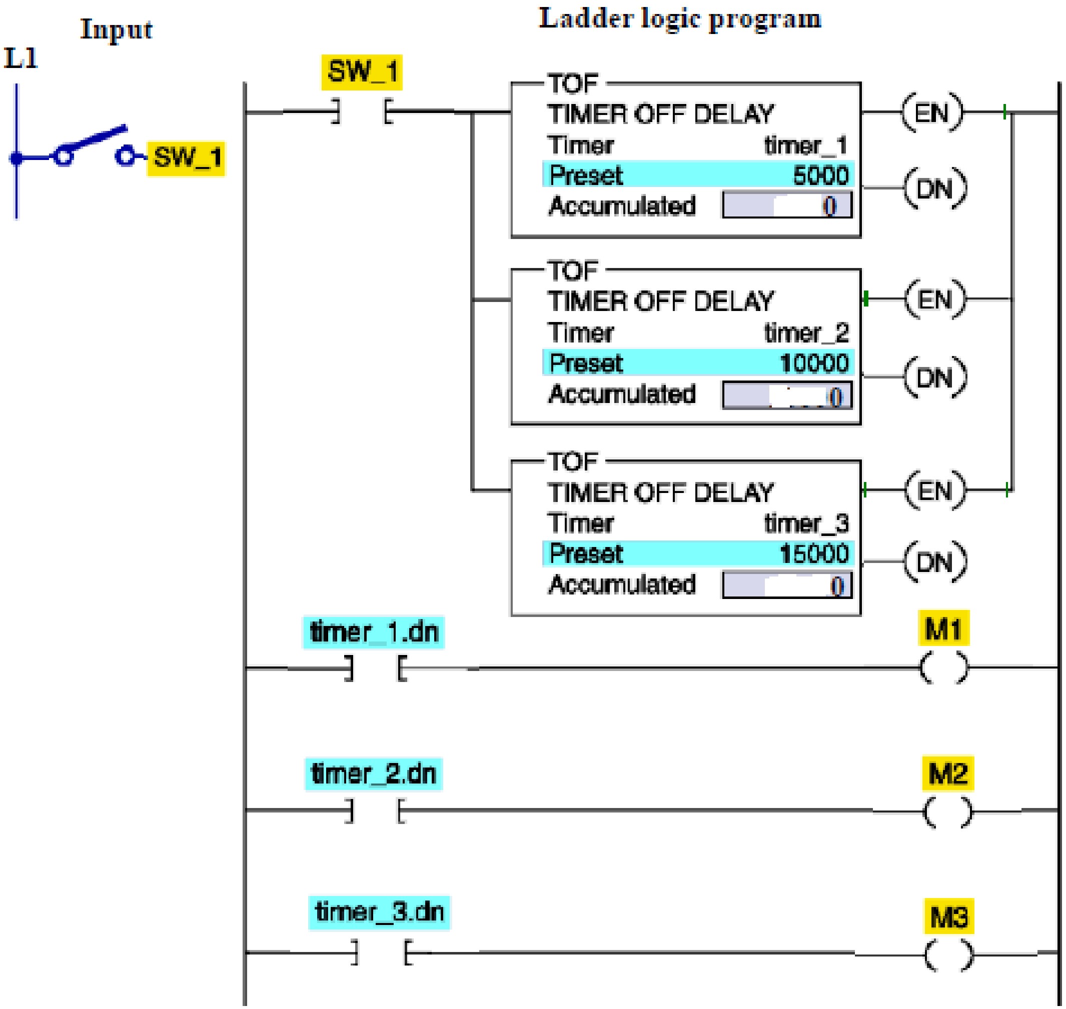
Ladder logic program
Input
Output
L1
L2
S1
I:1.0/0
I:1.0/0
TOF
TIMER OFF DELAY
TIMER T4:3
Time base 1.0
Preset 15
Accumulated 0
EN
DN
O:2.0/1
T4:3/DN
O:2.0/1
True
Input condition
S1
False
Timer-timing
Off delay
timed duration
True (logic 1)
Timed output
O:2.0/1
False (logic 0)
Preset value = accumulated value

Off-Delay Timer Used To Switch Motors Off



What dose this program do?

Off-Delay Timer Used To Switch Motors Off



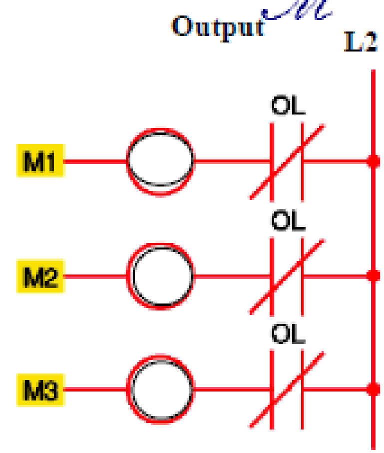
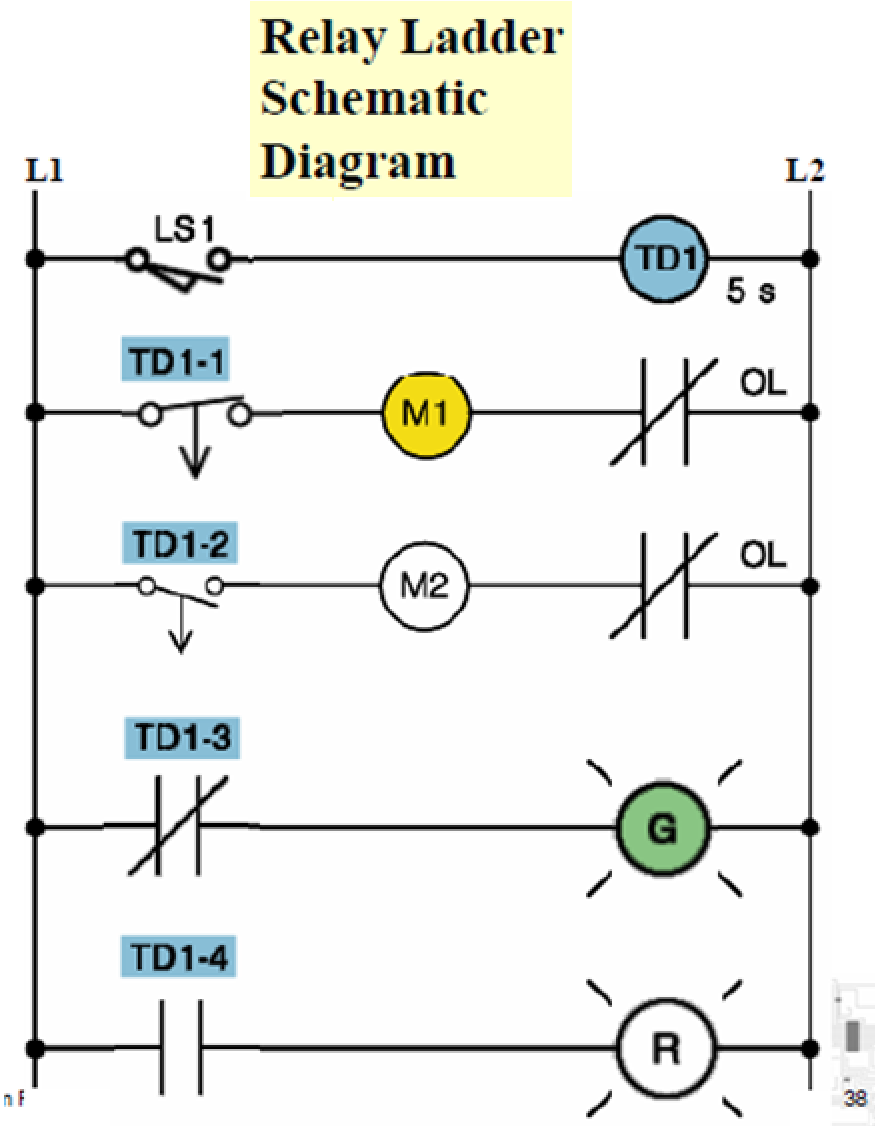


Figure 7-18 illustrates the use of the ControlLogix off-delay timer instruction. In this application, closing the switch immediately turns on motors M1, M2, and M3. When the switch is opened, motors M1, M2, and M3 turn off at 5-s intervals.

Off-Delay Timer

What dose this program do?

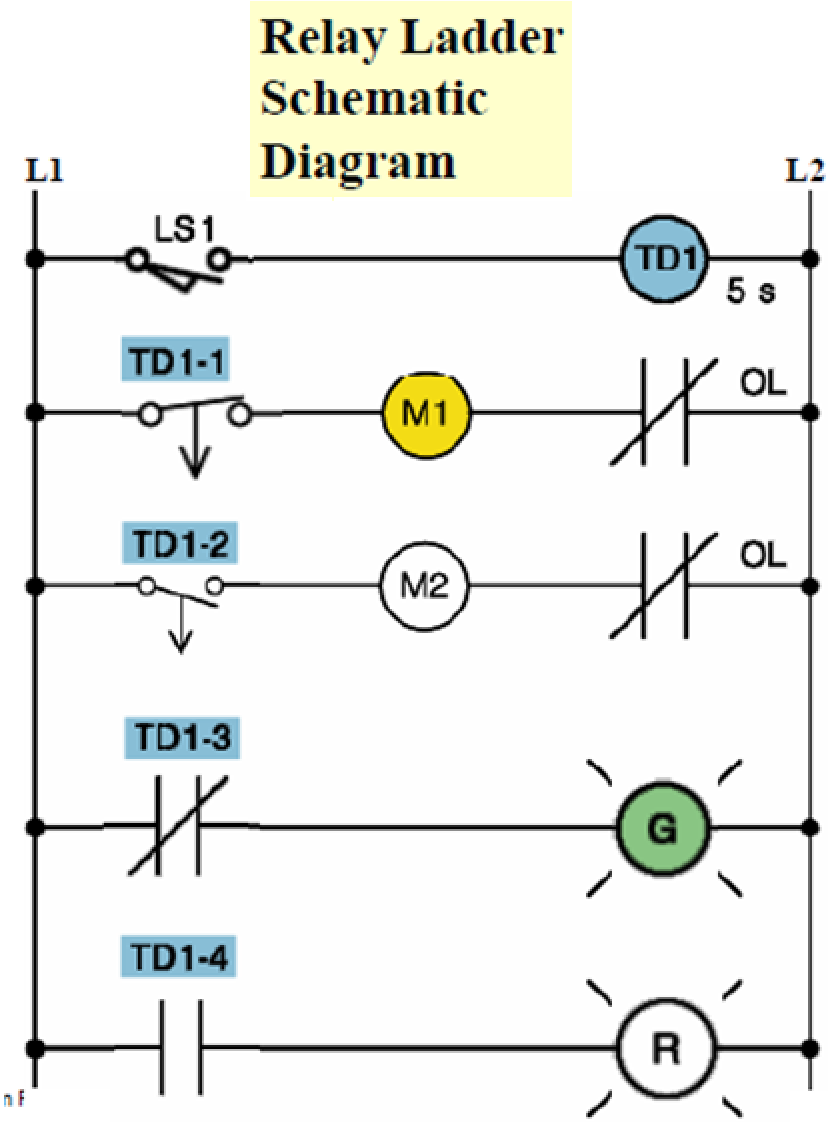


when power is first applied (limit switch LS1 open), motor starter coil M1 is energized and the green pilot light is on. At the same time, motor starter coil M2 is de-energized, and the red pilot light is off.

When limit switch LS1 closes, off-delay timer coil TD1 energizes. As a result, timed contact TD1-1 opens to de-energize motor starter coil M1, timed contact TD1-2 closes to energize motor starter coil M2, instantaneous contact TD1-3 opens to switch the green light off, and instantaneous contact TD1-4 closes to switch the red light on. The circuit remains in this state as long as limit switch LS1 is closed.

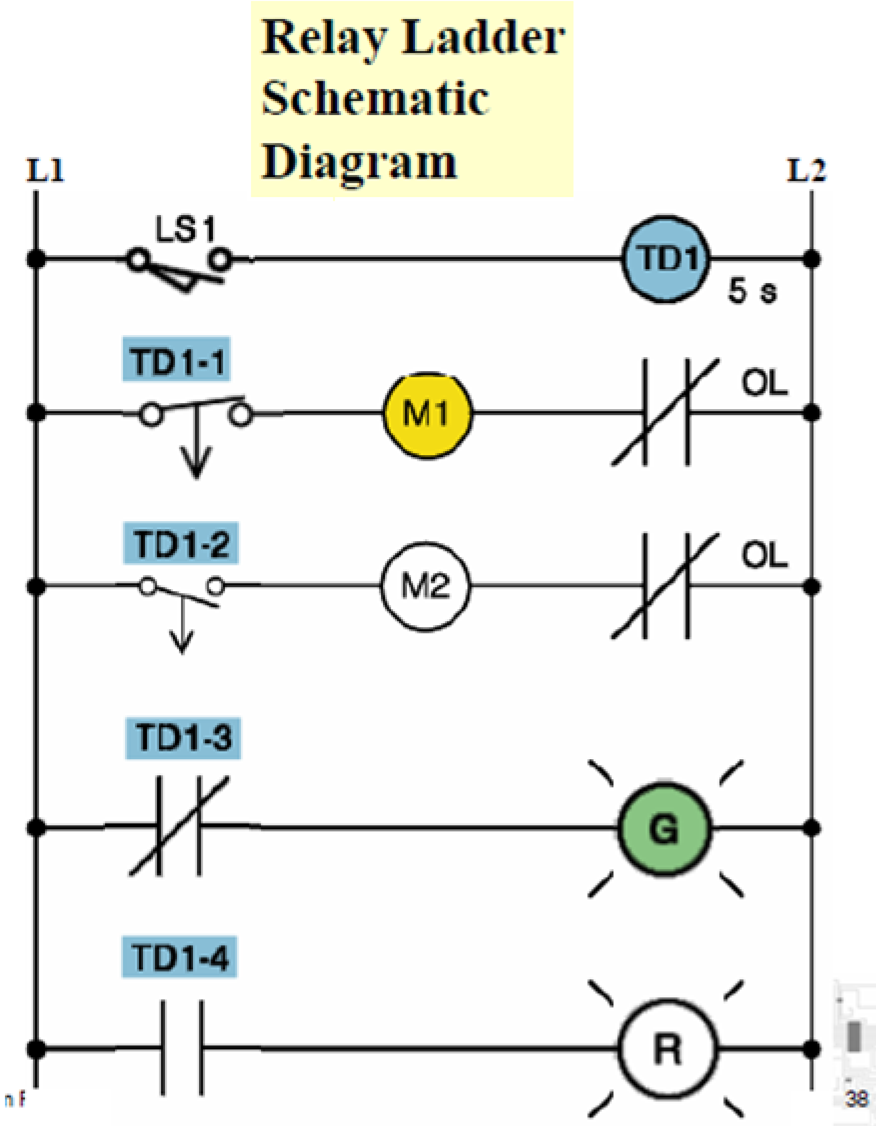
When limit switch LS1 is opened, the off-delay timer coil TD1 de-energizes. As a result, the time-delay period is started, instantaneous contact TD1-3 closes to switch the green light on, and instantaneous contact TD1-4 opens to switch the red light off. After a 5-s time-delay period, timed contact TD1-1 closes to energize motor starter M1, and timed contact TD1-2 opens to de-energize motor starter M2. Figure 7-19*b* shows how the circuit is programmed using the SLC-500 TOF timer.

Off-Delay Timer



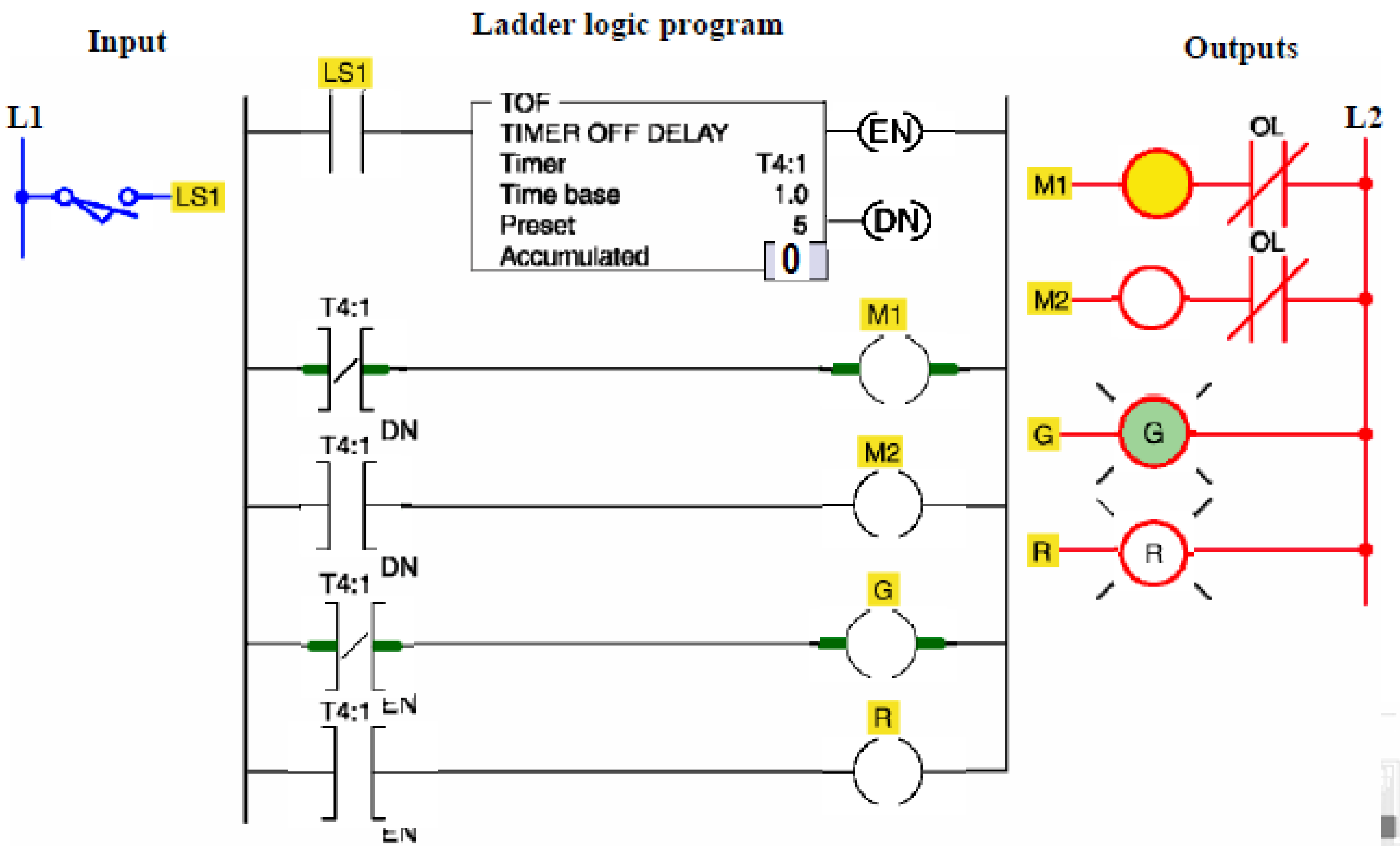
Off-Delay Timer

Convert to PLC ladder program?

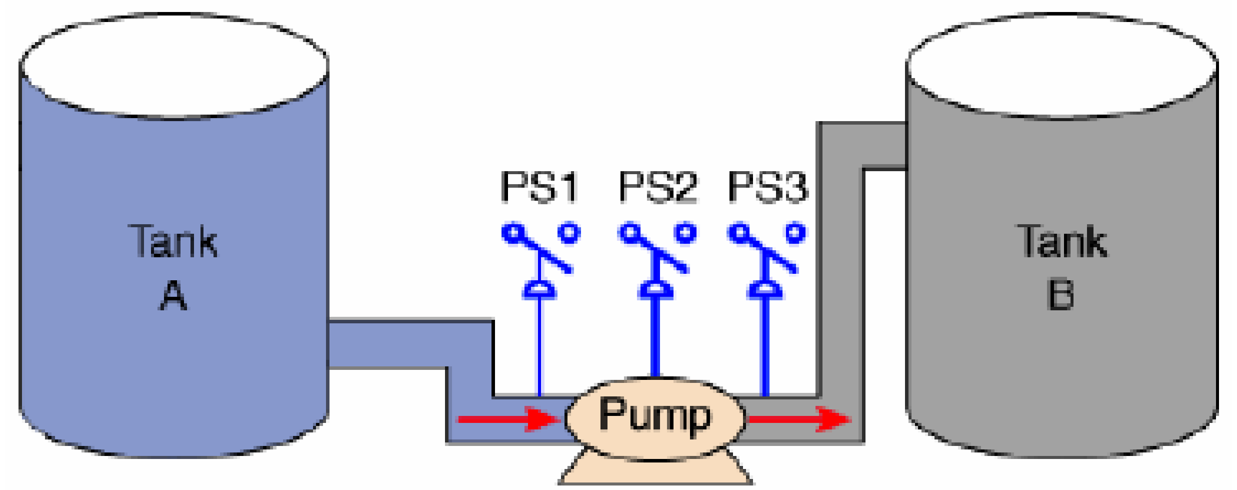


Solution to be added later

Equivalent Programmed Circuit



Fluid Pumping Process



Operation

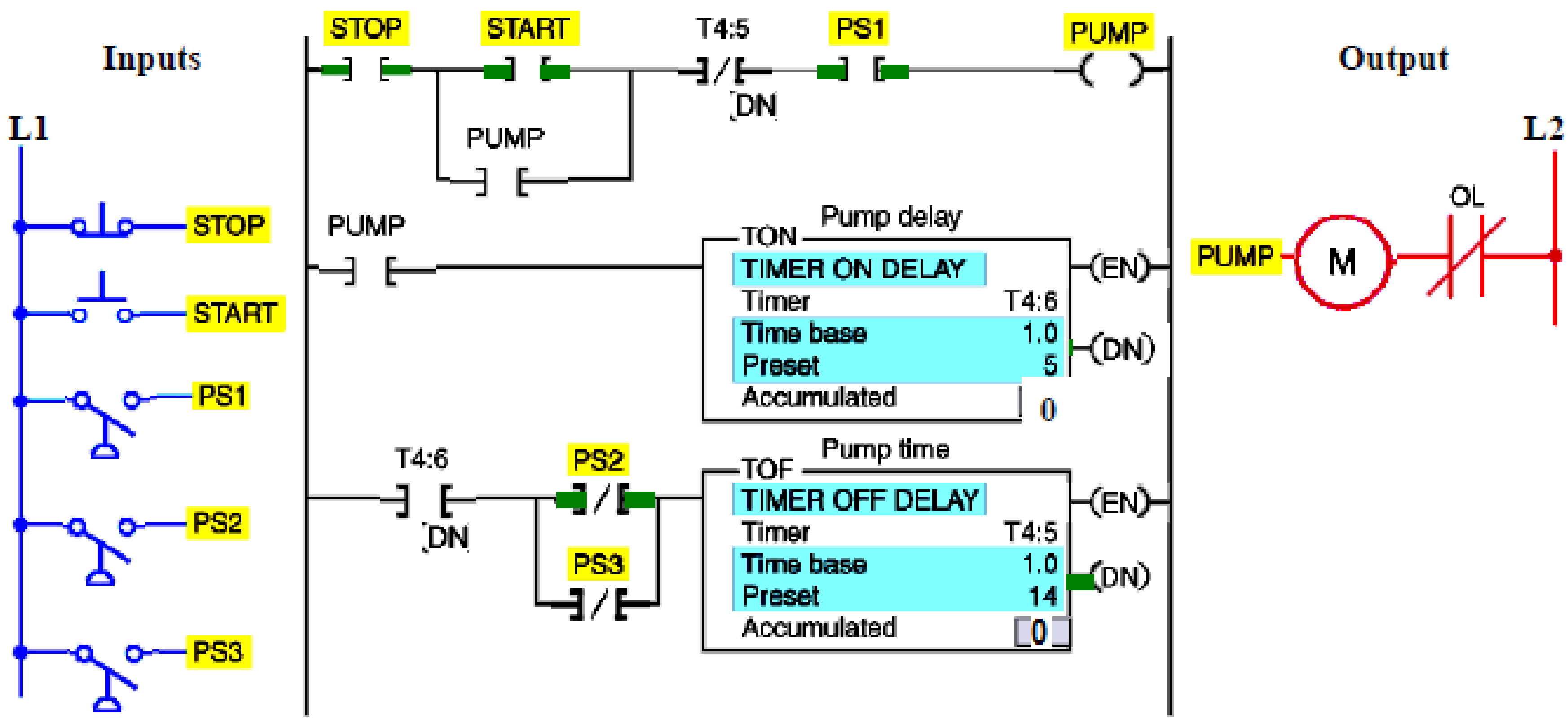
- Before starting, PS1 must be closed.
- When the pump start button is pressed, the pump starts. The button can then be released and the pump continues to operate.
- When the stop button is pushed, the pump stops.
- PS2 and PS3 must be closed for 5 s after the pump starts. If either PS2 or PS3 opens, the pump will shut off and will not not be able to start again for another 14 s.

Fluid Pumping Process Program

Solution to be added later

Fluid Pumping Process Program

Ladder logic program



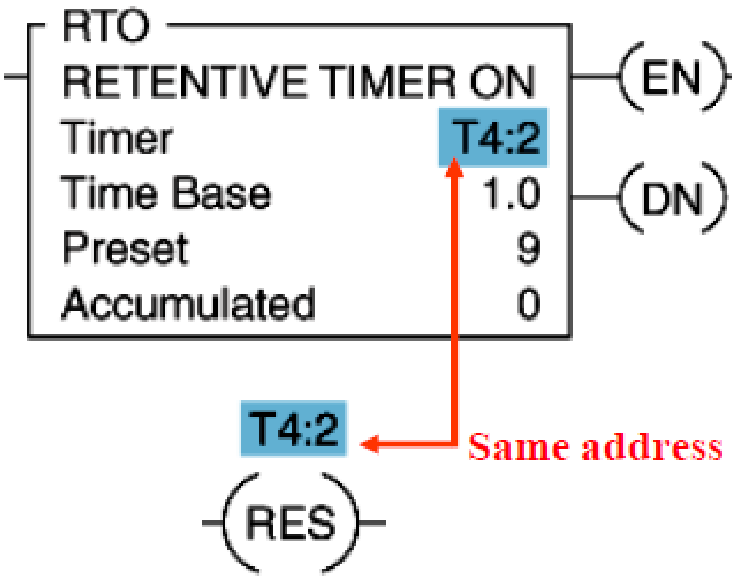
Retentive Timer

Radu Muresan

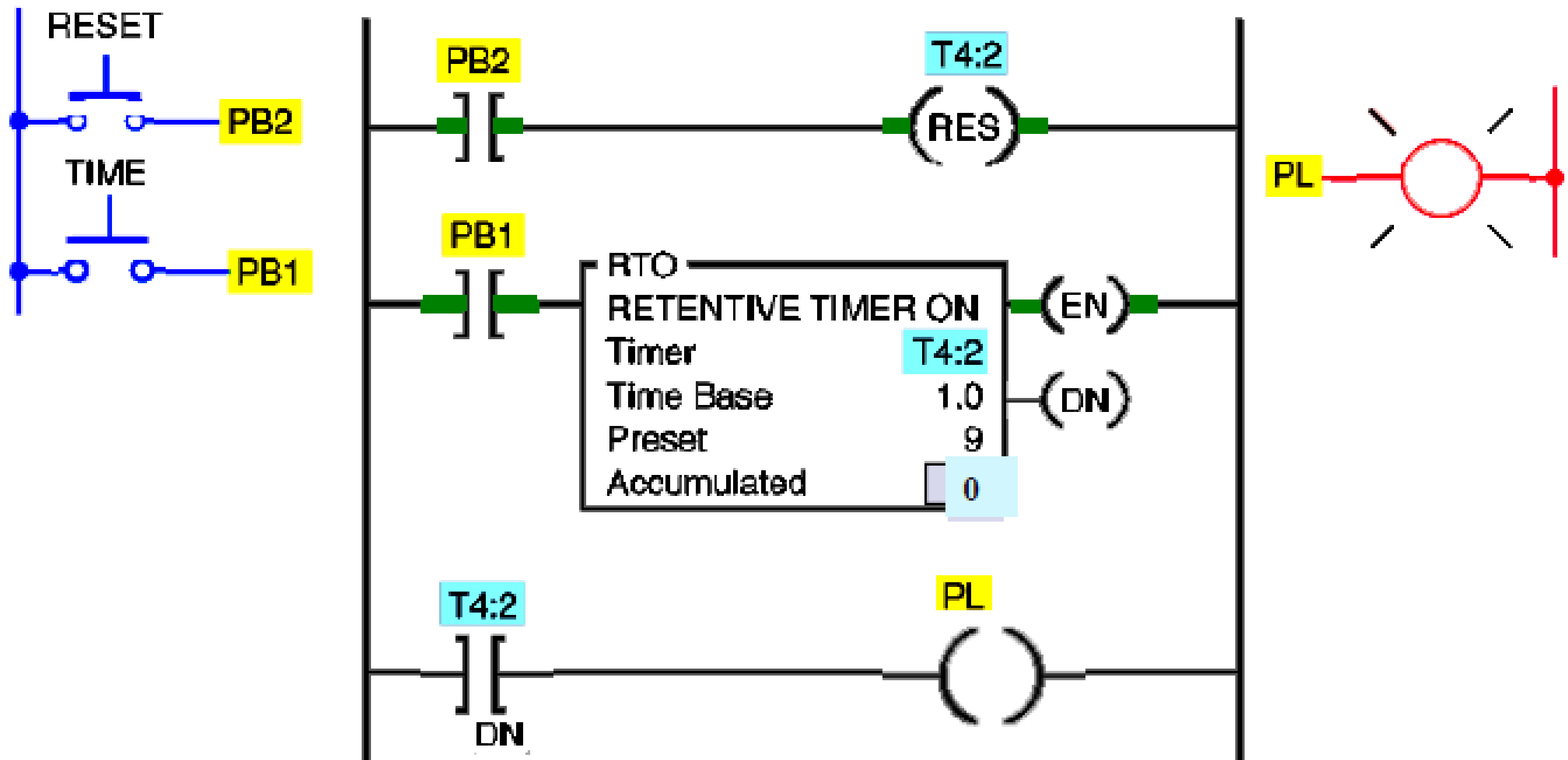
A *retentive timer* accumulates time whenever the device receives power, and maintains the current time should power be removed from the device. Once the device accumulates time equal to its preset value, the contacts of the device change state. The retentive timer must be *intentionally reset* with a separate signal for the accumulated time to be reset.

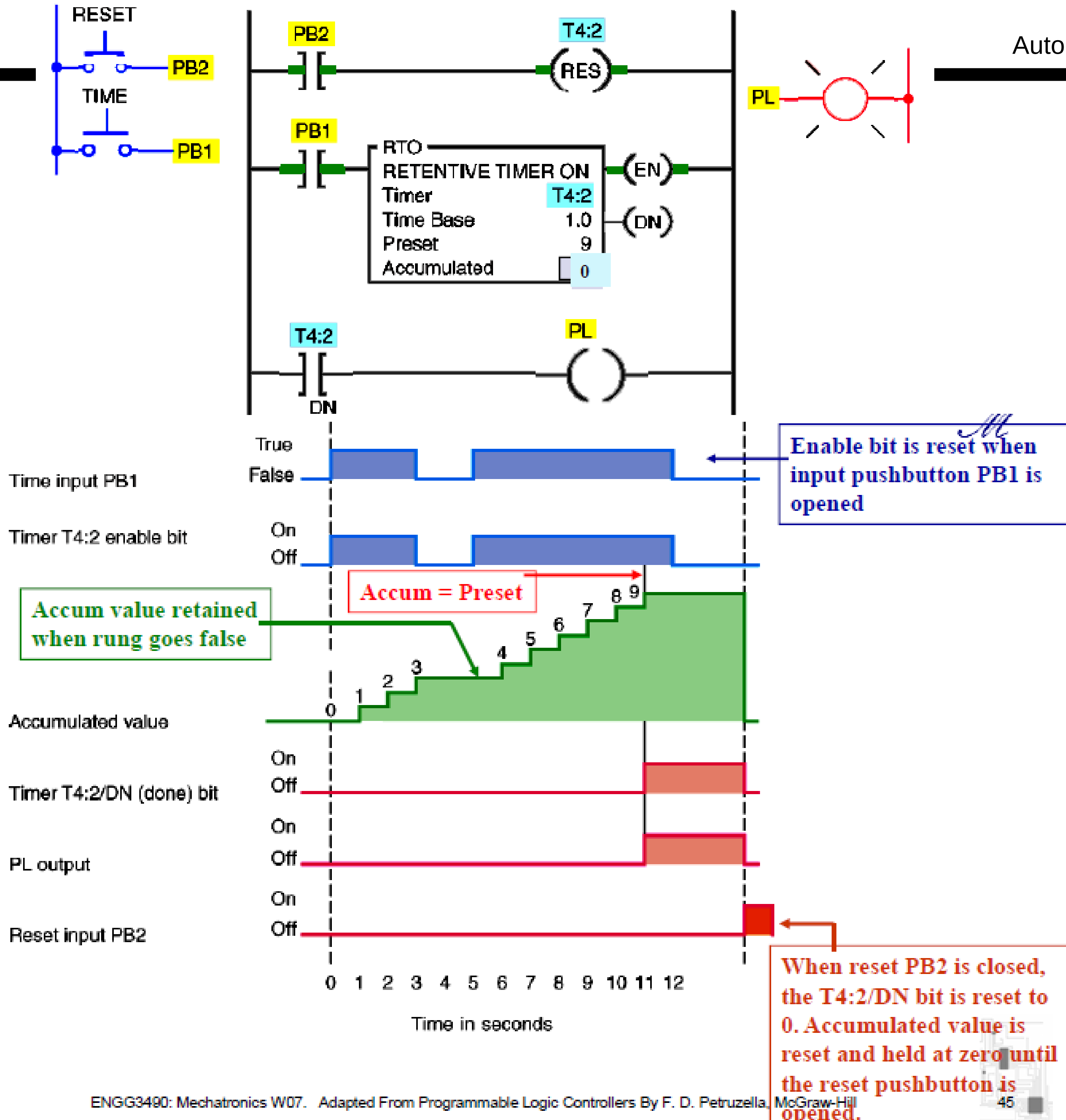
Retentive On-Delay Timer Program

The PLC-programmed RETENTIVE ON-DELAY timer (RTO) operates in the same way as the nonretentive on-delay timer (TON), with one major exception. There is a retentive timer reset (RES) instruction.

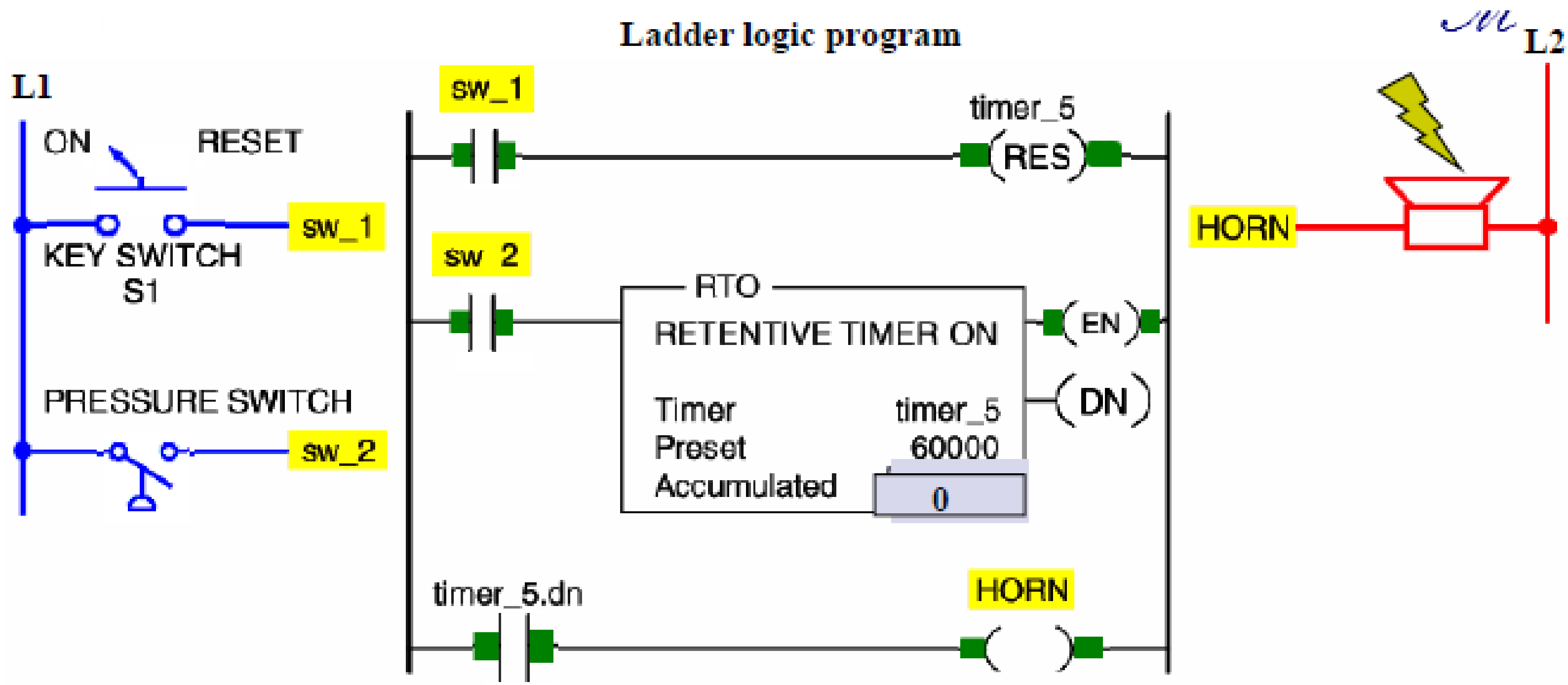


Unlike the TON, the RTO will hold its accumulated value when the timer rung goes false and will continue timing where it left off when the timer rung goes true again. This timer must be accompanied by a timer reset (RES) instruction to reset the accumulated value of the timer to zero.


RESET
PB2
TIME
PB1
PB2
T4:2
RES
PL
PB1
RTO
RETENTIVE TIMER ON
Timer T4:2
Time Base 1.0
Preset 9
Accumulated 0
EN
DN
T4:2
DN
PL

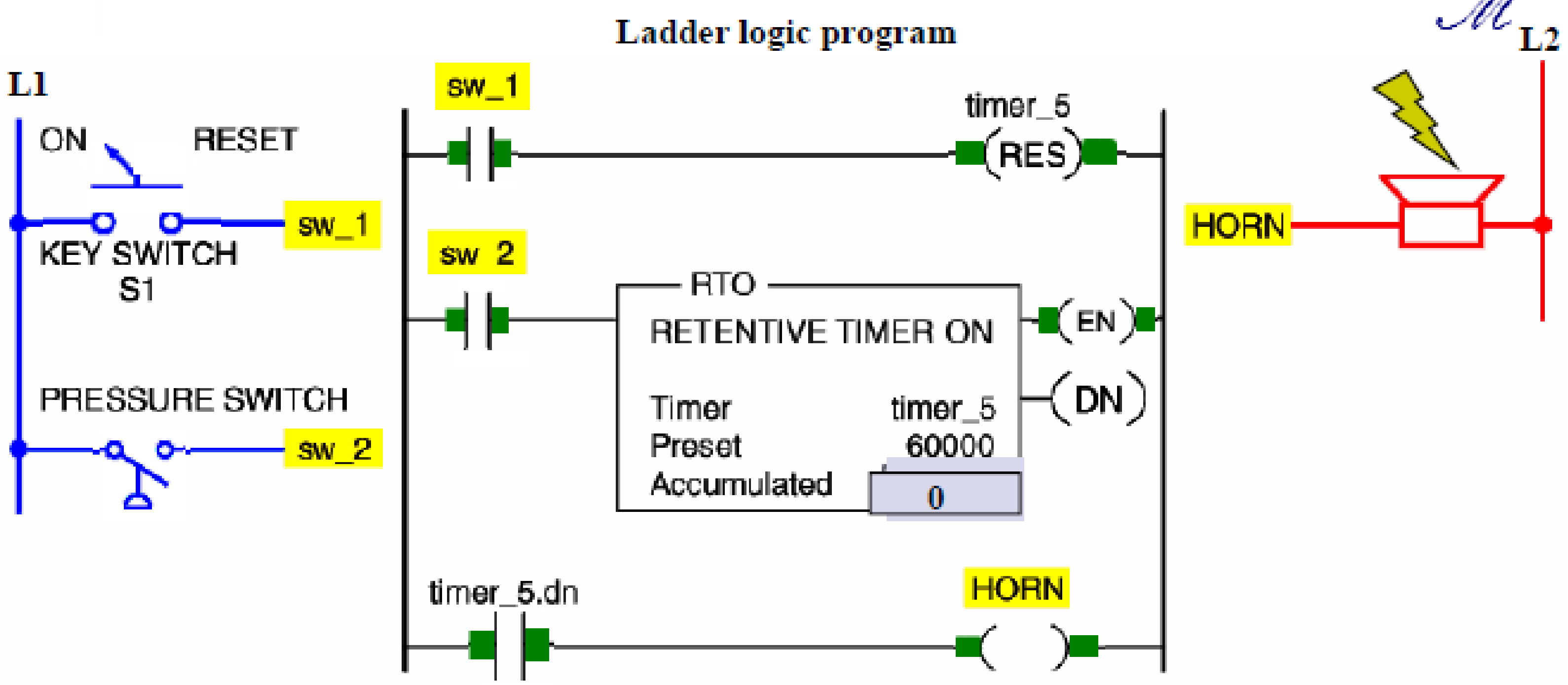


RESET
PB2
TIME
PB1
PB2
T4:2
RES
PL
PB1
RTO
RETENTIVE TIMER ON
Timer T4:2
Time Base 1.0
Preset 9
Accumulated 0
EN
DN
T4:2
DN
PL
True
False
Time input PB1
Enable bit is reset when input pushbutton PB1 is opened
Timer T4:2 enable bit
On
Off
Accum = Preset
Accum value retained when rung goes false
0 1 2 3 4 5 6 7 8 9
Accumulated value
On
Off
Timer T4:2/DN (done) bit
On
Off
PL output
On
Off
Reset input PB2
0 1 2 3 4 5 6 7 8 9 10 11 12
Time in seconds
When reset PB2 is closed, the T4:2/DN bit is reset to 0. Accumulated value is reset and held at zero until the reset pushbutton is opened.



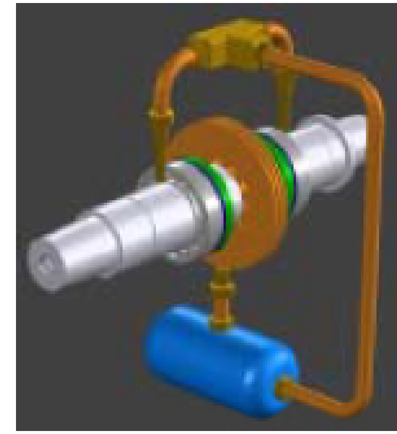
What dose this program do?

Retentive On-delay Alarm Program



The purpose of the RTO timer is to detect whenever a piping system has sustained a cumulative overpressure condition of 60 s. At that point, a horn is sounded automatically. You can silence the alarm by switching the key switch to the rest position.

Bearing Lubrication Program

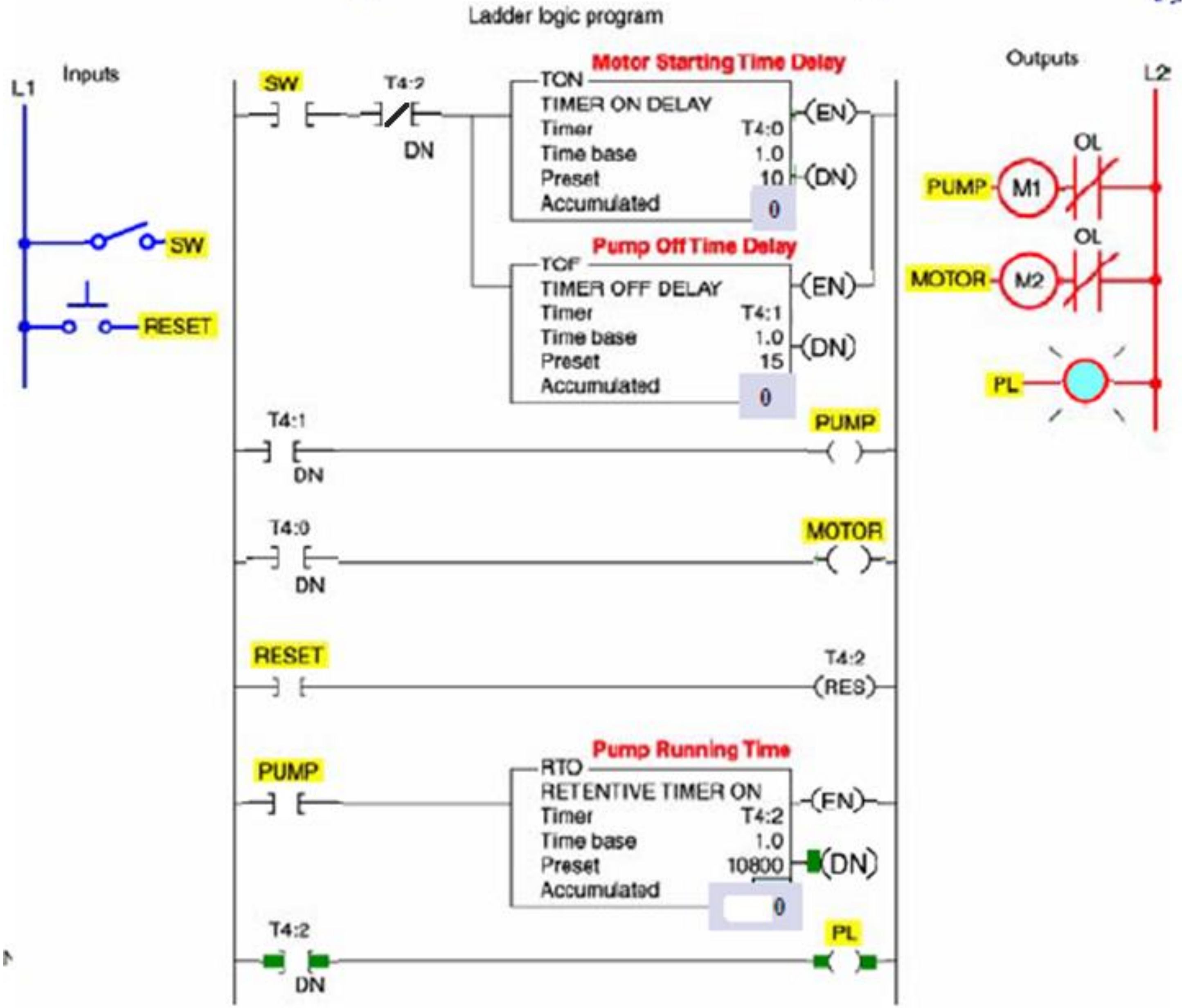


Sequence Of Operation

- To start the machine, the operator turns SW on.
- Before the motor shaft starts to turn, the bearings are supplied with oil by the pump for 10 s.
- The bearings also receive oil when the machine is running.
- When the operator turns SW off to stop the machine, the oil pump continues to supply oil for 15 s.
- A retentive timer is used to track the total running time of the pump. When the total running time is 3 h, the motor is shut down and a pilot light is turned on to indicate that the filter and oil need to be changed.
- A reset button is provided to reset the process after the filter and oil have been changed.

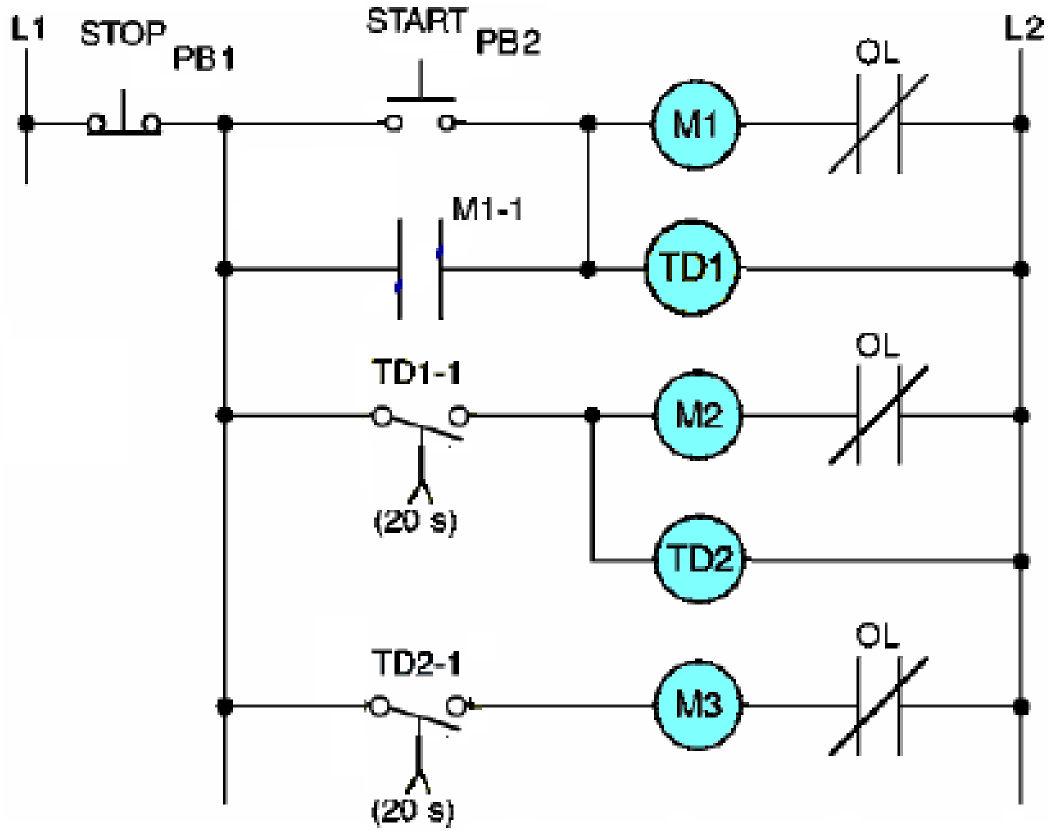
Solution to be added later

Bearing Lubrication Program



Cascading Timers

The programming of two or more timers together is called *cascading*. Timers may be interconnected, or cascaded to satisfy any required control logic.



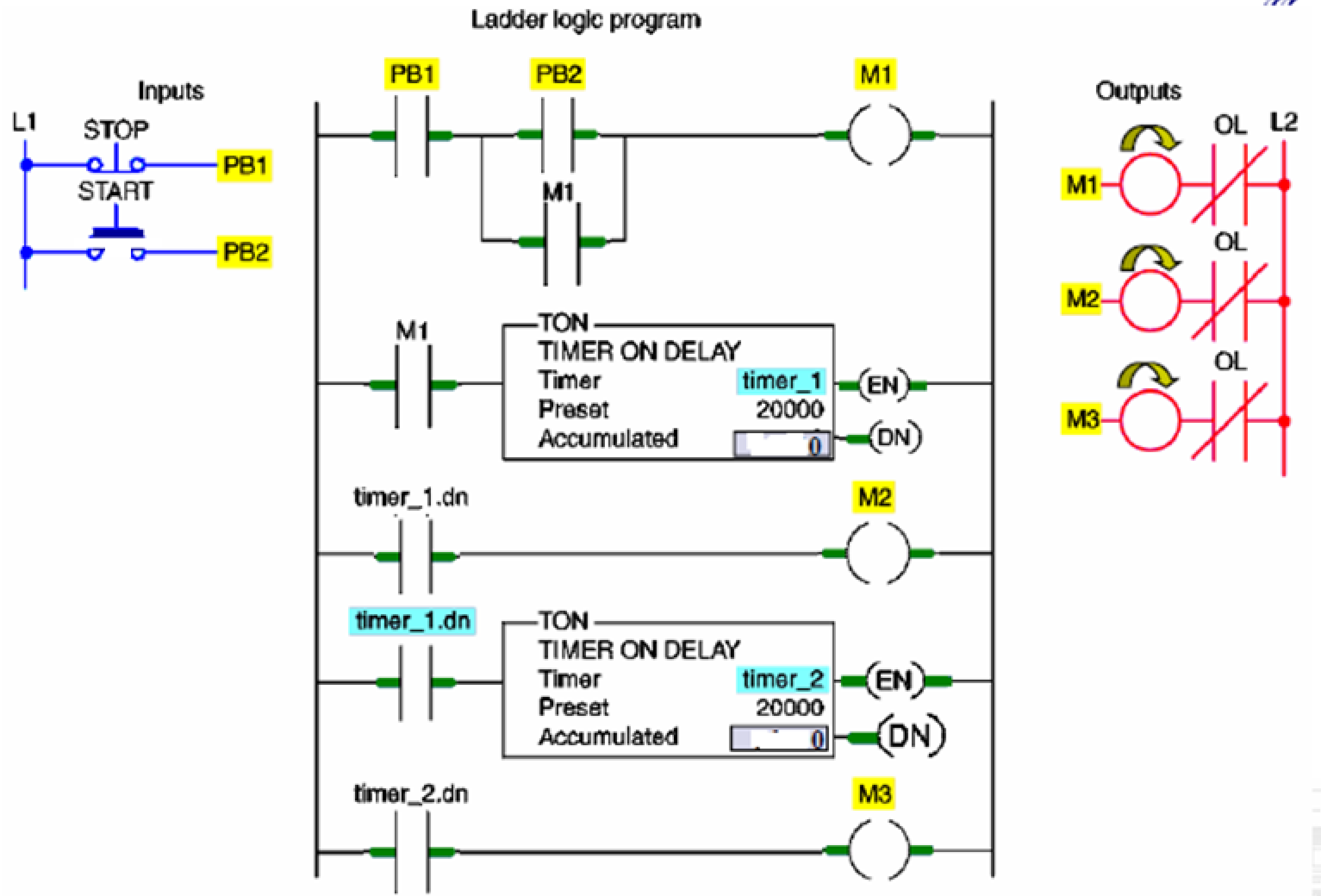
Relay Schematic Diagram

Three motors started automatically in sequence with a 20-s time delay between each motor startup.

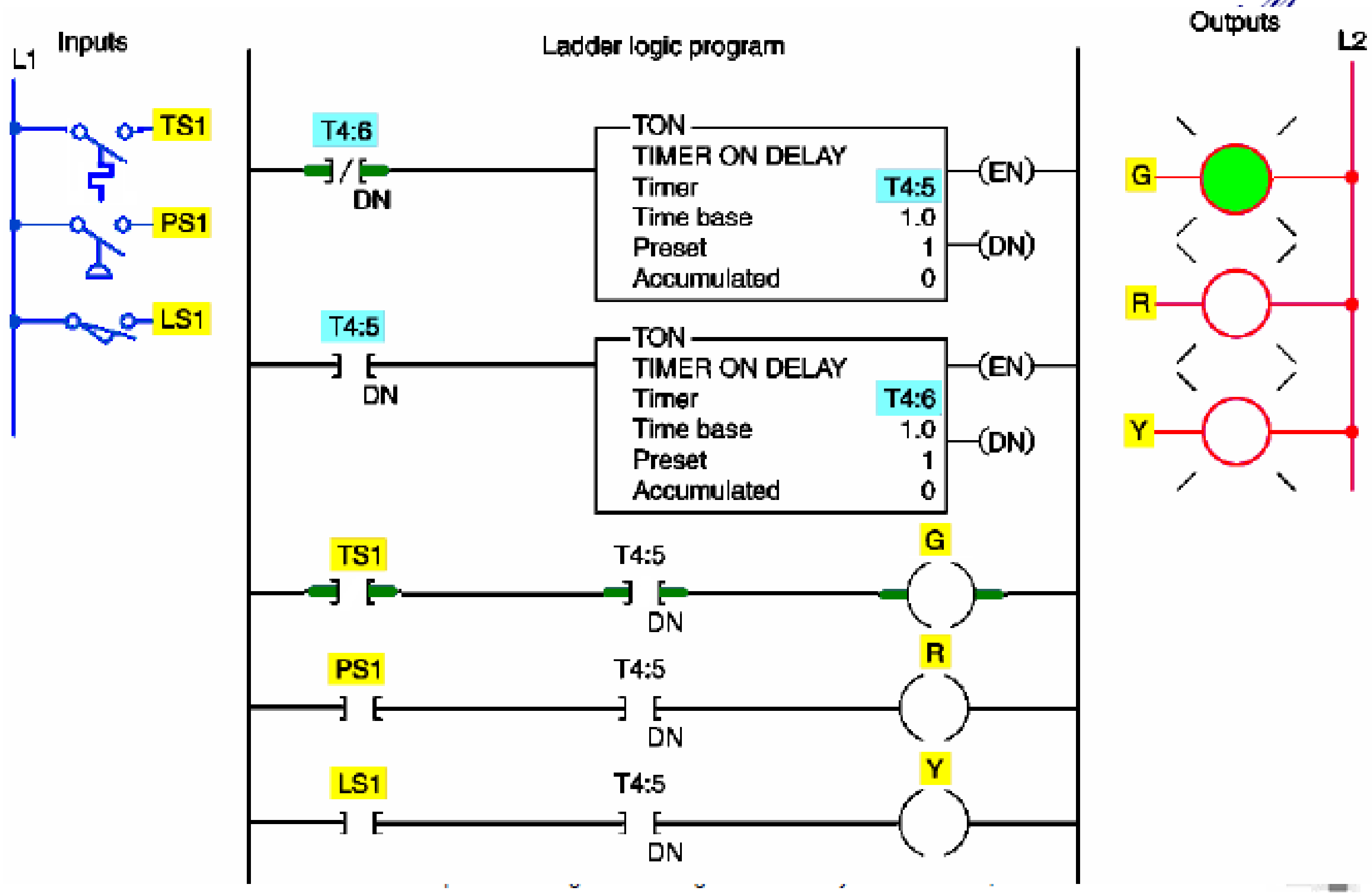
Convert to PLC ladder program?

Solution to be added later

Equivalent Time-Delayed Motor-Starting Program

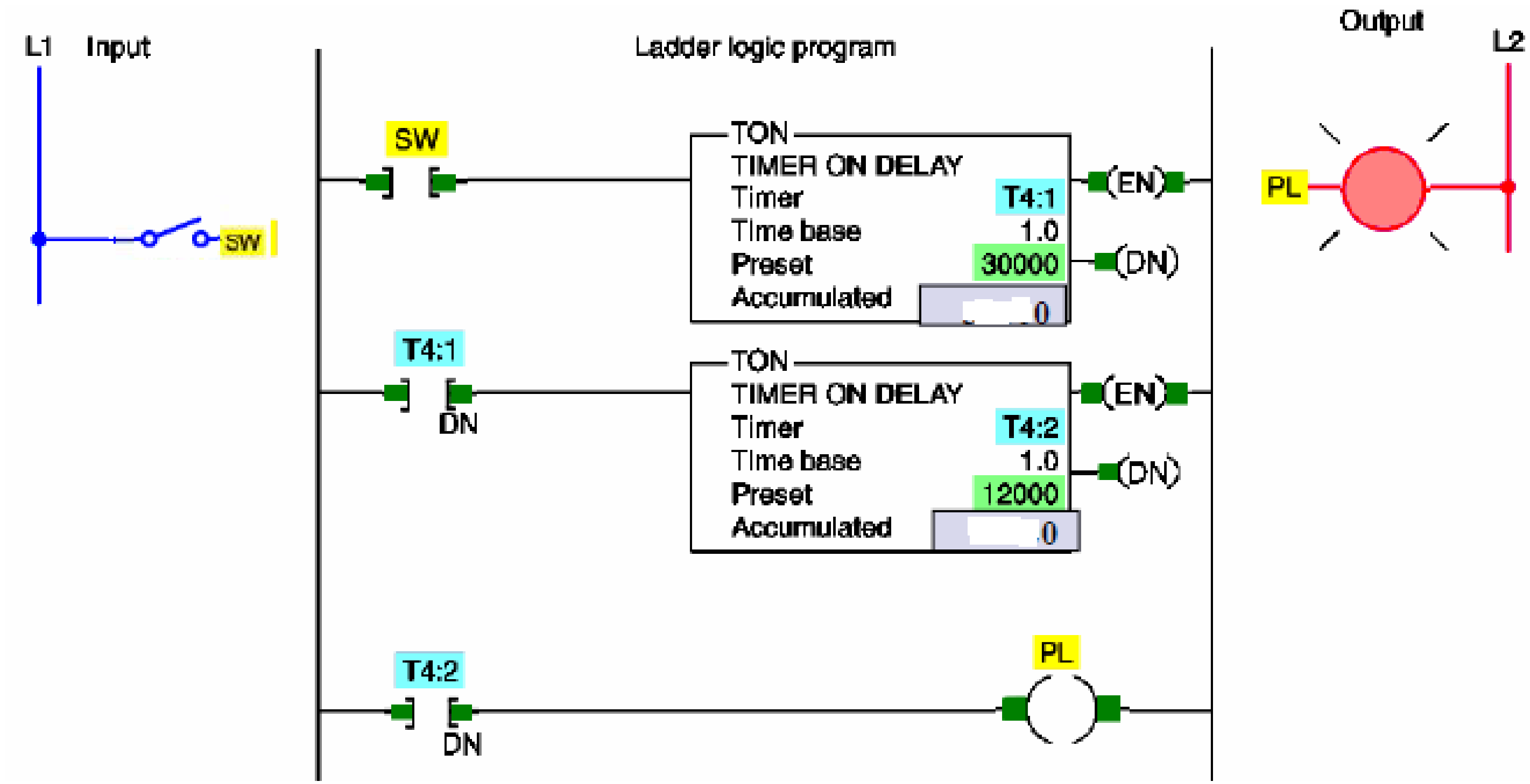


Annunciator Flasher Circuit

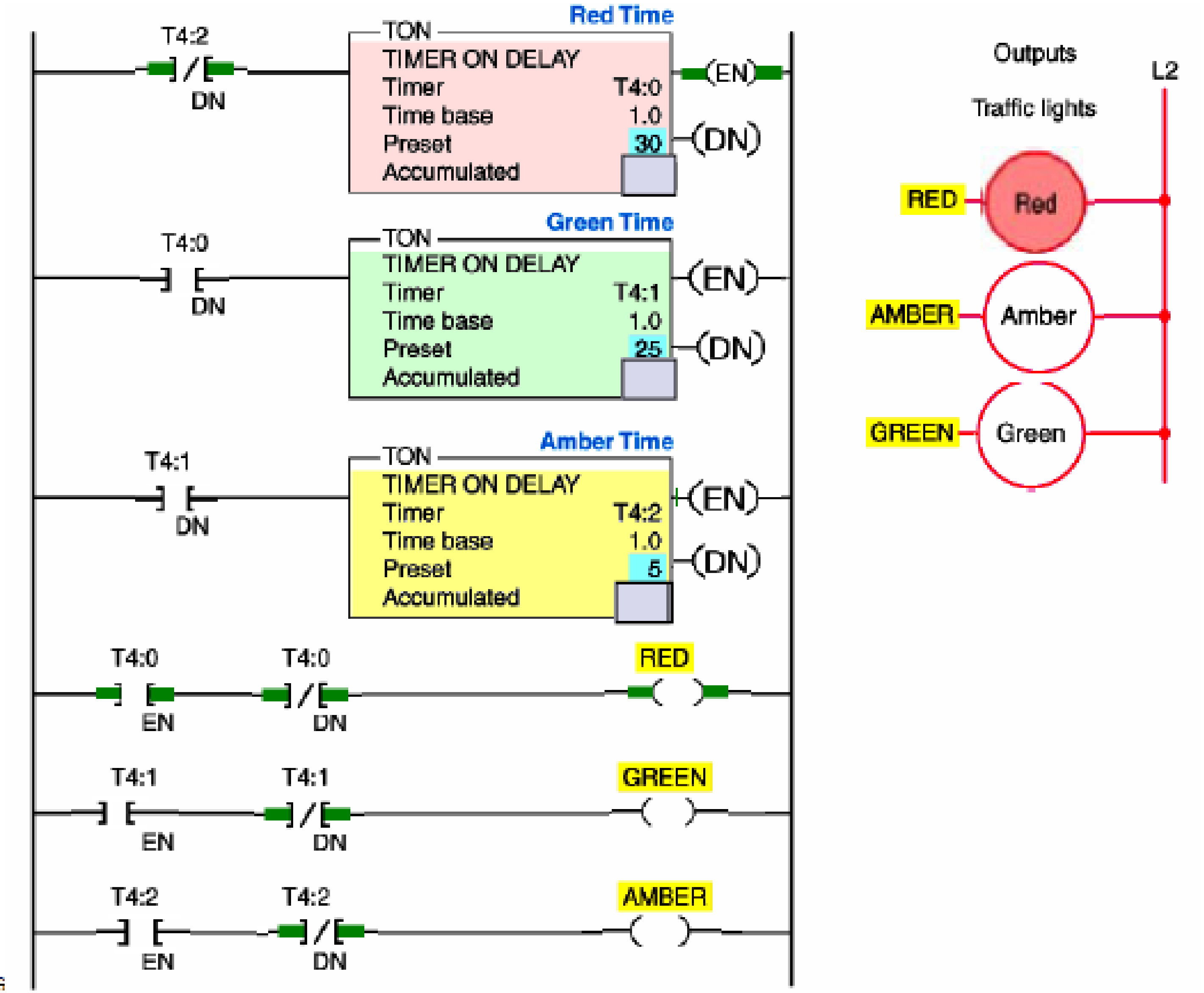


What dose this program do?

Cascading of Timers for Longer Time Delays

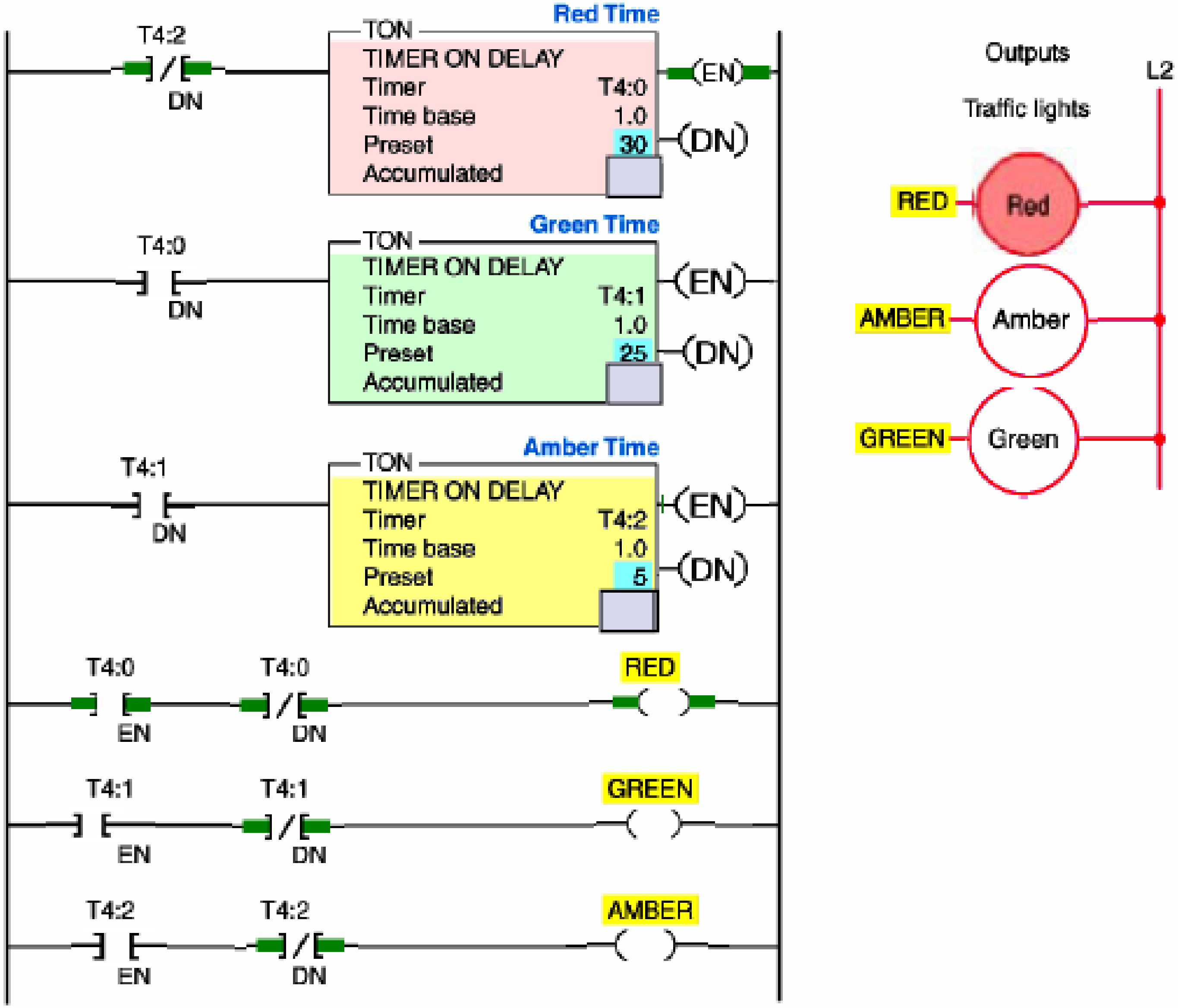


What dose this program do?



ENG

Control of Traffic Lights in One Direction



ENG

references:

1-notes from Dr. **Jeff Jackson** *,the university of Alabama*

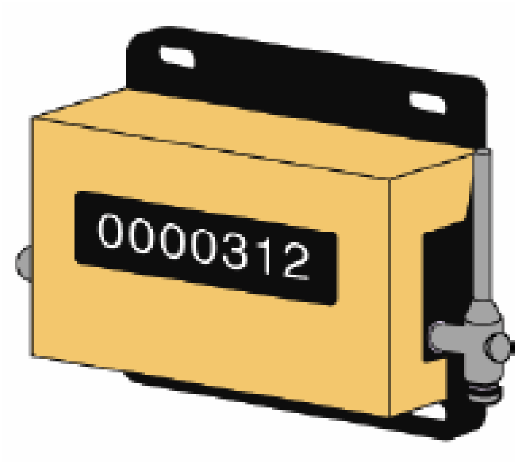
2-notes from Dr. Radu Muresan ,University of Guelph

Counters

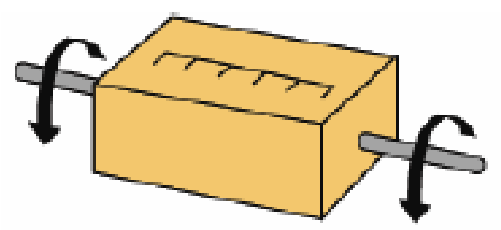
Common applications of counters include keeping track of the number of items moving past a given point, and determining the number of times a given action occurs.

A preset counter can control an external circuit when its counted total matches the user-entered preset limits.

Mechanical Counters



Programmed counters can serve the same functions as mechanical counters.



Every time the actuating lever is moved over the counter adds one number, while the actuating lever returns automatically to its original position. Resetting to zero is done with a pushbutton located on the side of the unit.

Electronic counters can count up, count down, or be combined to count up and down. They are dependent on external sources, such as parts traveling past a sensor or actuating a limit switch for counting.

Counter Counting Sequence

PLC counters are normally retentive. Whatever count was contained in the counter at the time of a processor shutdown will be restored to the counter on power-up. The counter may be reset, however, if the reset condition is activated at the time of power restoration.

PLC counters can be designed to count up to a preset value or to count down to a preset value.

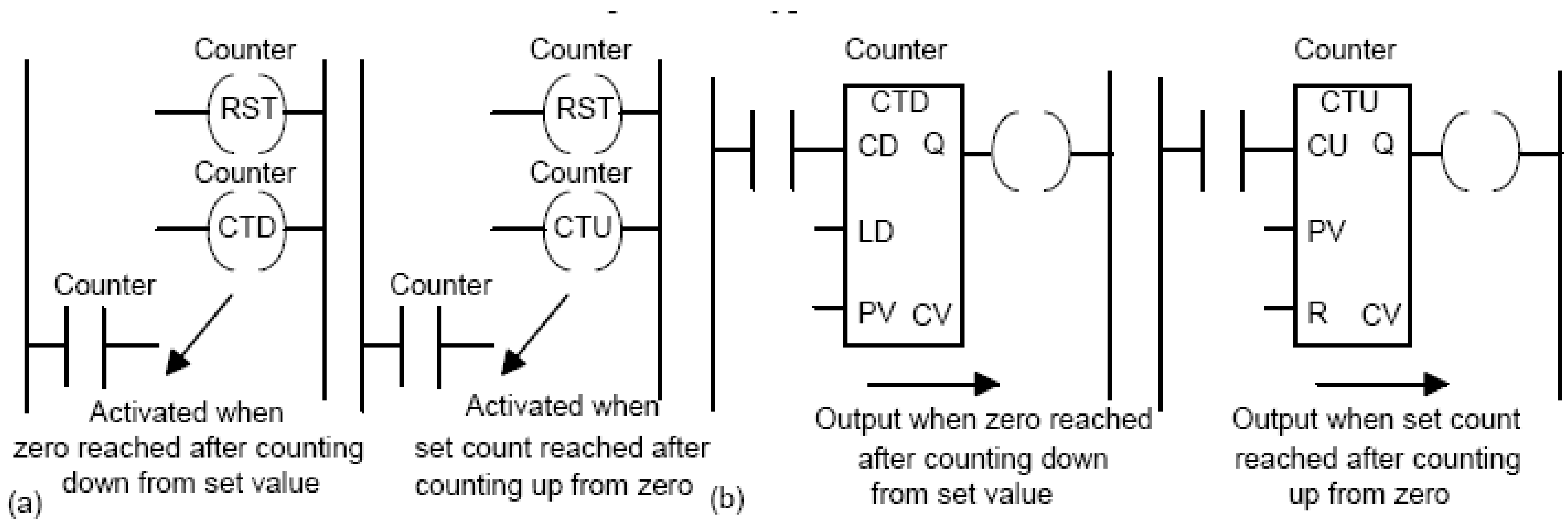
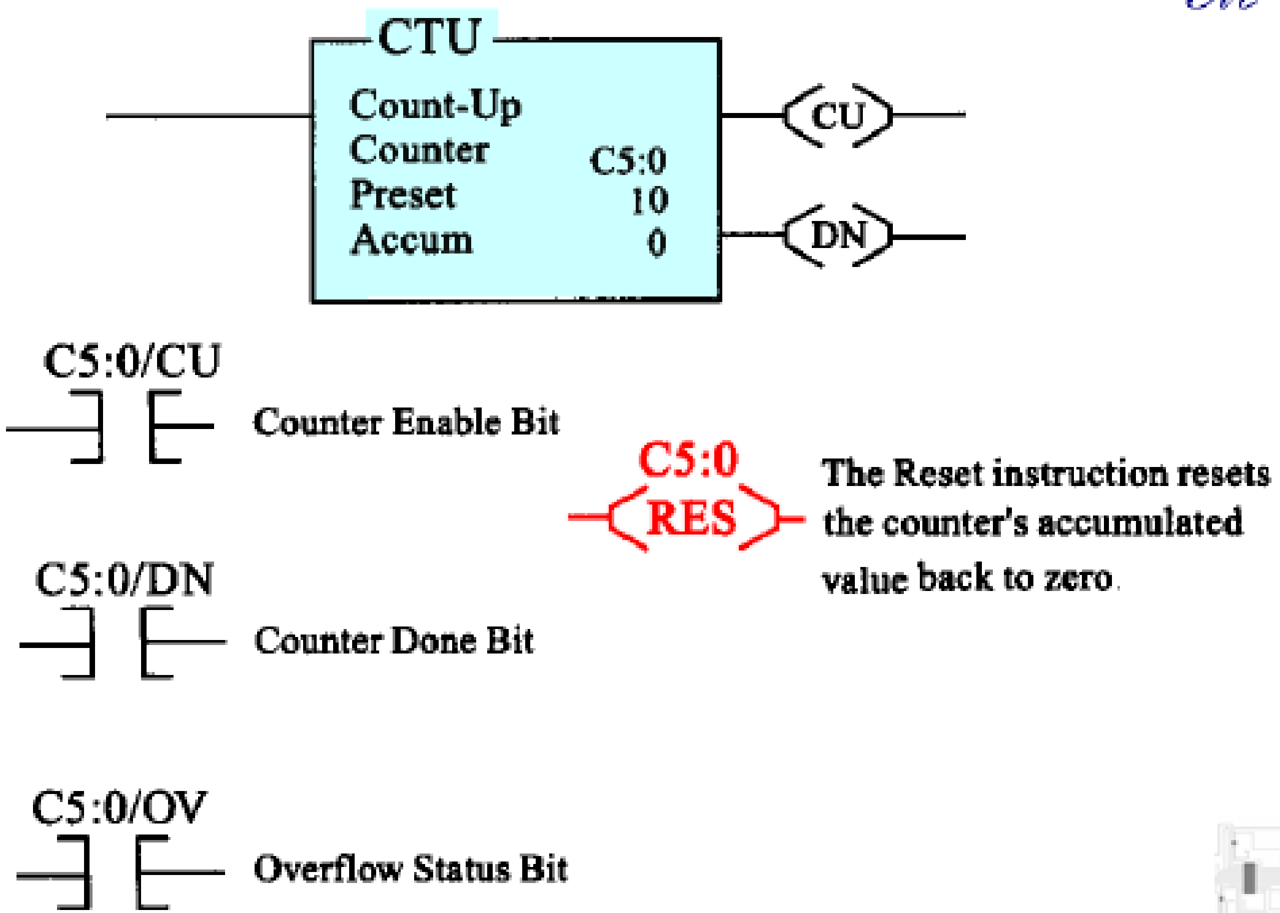


Figure 10.1 *Forms of representation of counters. In (a) RST is reset. In (b), the IEC 1131-3 representation, CD is count down input, LD is for loading the input, PV is for the preset value, CV the current count value, CU is count up input, and R is for the reset input.*

PLC-5 And SLC 500 Count-Up Counter Instruction



C5 Counter Data File

Each counter address is made of a 3-word element

Counter Address																	
C5:N	Bit	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
C5:N.0	Word 0	CU	CD	DN	OV	UN	UA	Internal Use (not addressable)									
C5:N.1																	
C5:N.2																	

- •Bit 0-9: Internal Use
- •Bit 10: UA - Update accumulation value.
- •Bit 11: UN - Underflow bit.
- •Bit 12: OV - Overflow bit.
- •Bit 13: DN - Done
- •Bit 14: CD - Count down is enabled.
- •Bit 15: CU - Count up is enabled.

- **Count-Up (CU) Enable Bit**
  The count-up enable bit is used with the count-up counter and is true whenever the count-up counter instruction is true. If the count-up counter instruction is false, the CU bit is false.

- **Count-Down (CD) Enable Bit**
  The count-down enable bit is used with the count-down counter and is true whenever the count-down counter instruction is true. If the count-down counter instruction is false, the CD bit is false.

- **Done (DN) Bit**
  The done bit is true whenever the accumulated value is equal to or greater than the preset value of the counter, for either the count-up or the count-down counter.

- **Overflow (OV) Bit**
  The overflow bit is true whenever the counter counts past its maximum value, which is 32,767. On the next count, the counter will wrap around to −32,768 and will continue counting from there toward 0 on successive false-to-true transitions of the count-up counter.

- **Underflow (UN) Bit**
  The underflow bit will go true when the counter counts below −32,768. The counter will wrap around to +32,767 and continue counting down toward 0 on successive false-to-true rung transitions of the count-down counter.

C5 Counter Data File

Each counter address is made of a 3-word element

Counter Address																	
C5:N	Bit	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
C5:N.0																	
C5:N.1	Word 1	Preset Value															
C5:N.2																	

Specifies the value, which the counter must reach before the controller sets the done bit. When the accumulated value becomes equal to or greater than the preset value, the done status bit is set. You can use this bit to control an output device.

The *preset value (PRE) word* specifies the value that the counter must count to before it changes the state of the done bit. The preset value is the set point of the counter and ranges from −32,768 through +32,767.

C5 Counter Data File

Each counter address is made of a 3-word element

Counter Address																	
C5:N	Bit	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
C5:N.0																	
C5:N.1																	
C5:N.2	Word 2	Accumulated Value															

This is the number of times of false to true transitions that have occurred since the counter was last reset.

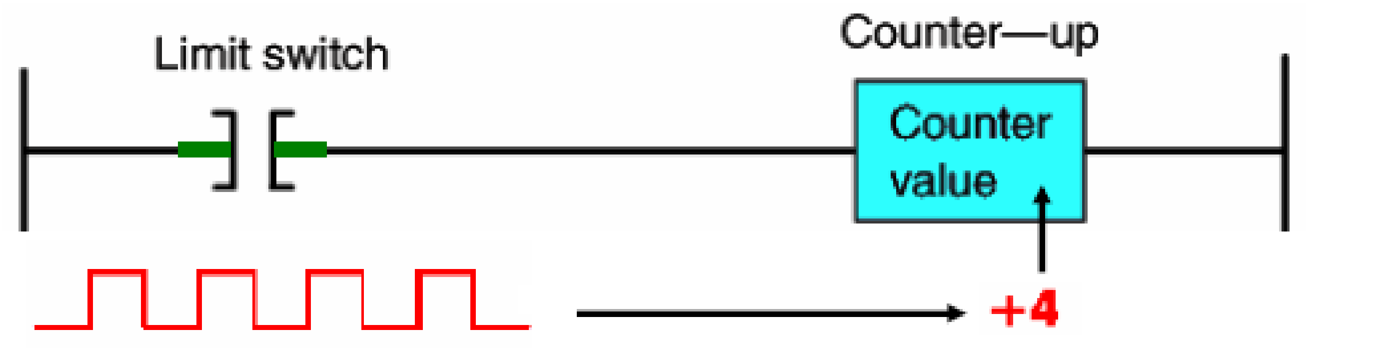
The *accumulated value (ACC) word* is the current count based on the number of times the rung goes from false to true. The accumulated value either increments with a false-to-true transition of the count-up counter instruction or decrements with a false-to-true transition of the count-down counter instruction. It has the same range as the preset: −32,768 through +32,767. The accumulated value will continue to count past the preset value instead of stopping at the preset like a timer does.

RSLogic Counter Commands

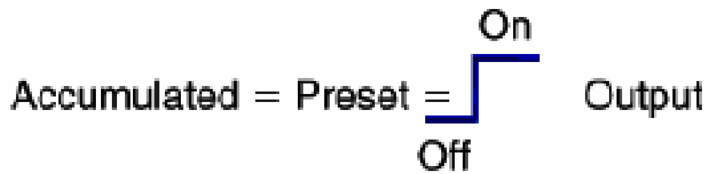
Command	Name	Description
CTU	Count-Up	Increments the accumulated value at each false-to-true transition and retains the accumulated value when power cycle occurs

Counter Counting Sequence

The up-counter is incremented by 1 each time the rung containing the counter is energized.



The counter will increment until the accumulated value is equal to or greater than the preset value, at which time an output will be produced.



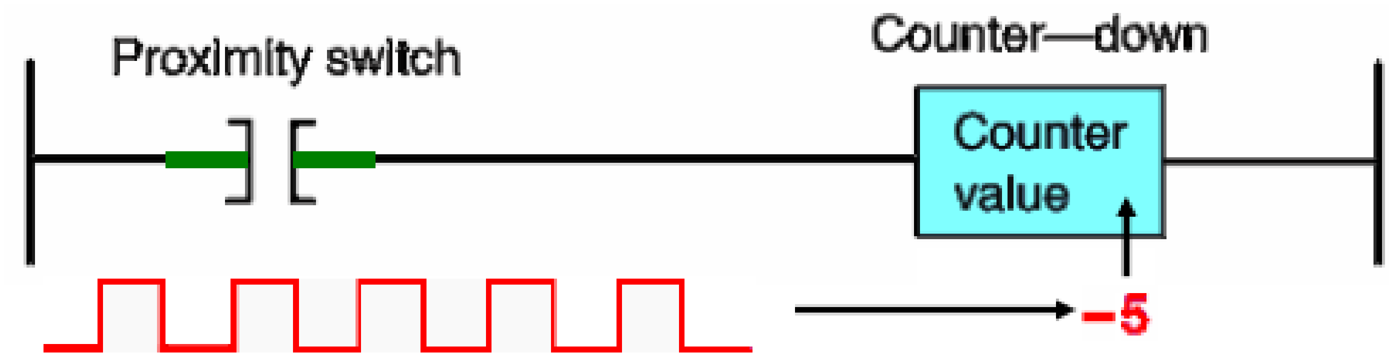
ENGG3490: Me uzella, McGraw-Hill 64

RSLogic Counter Commands

Command	Name	Description
CTD	Count-Down	Decrements the accumulated value at each false-to-true transition and retains the accumulated value when power cycle occurs

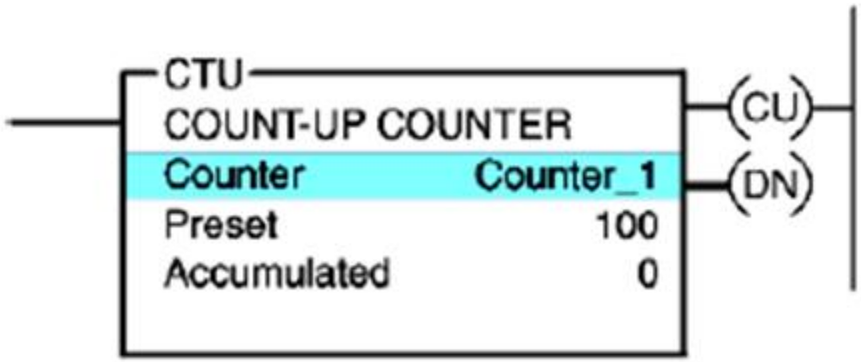
Counter Counting Sequence

The down-counter decrements by 1 each time the rung containing the counter is energized.



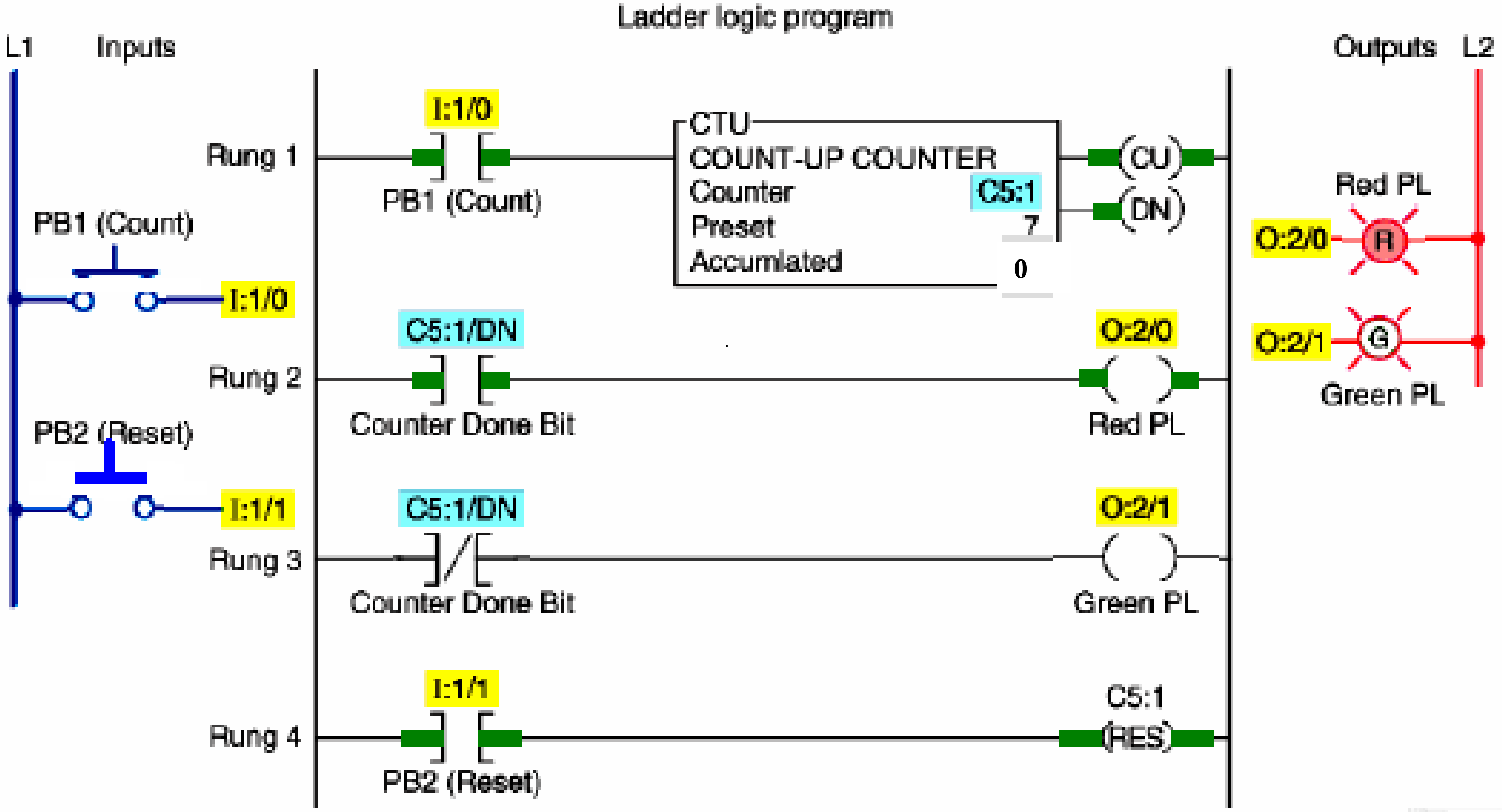
A counter reset is always provided to cause the counter accumulated value to be reset to a predetermined value.

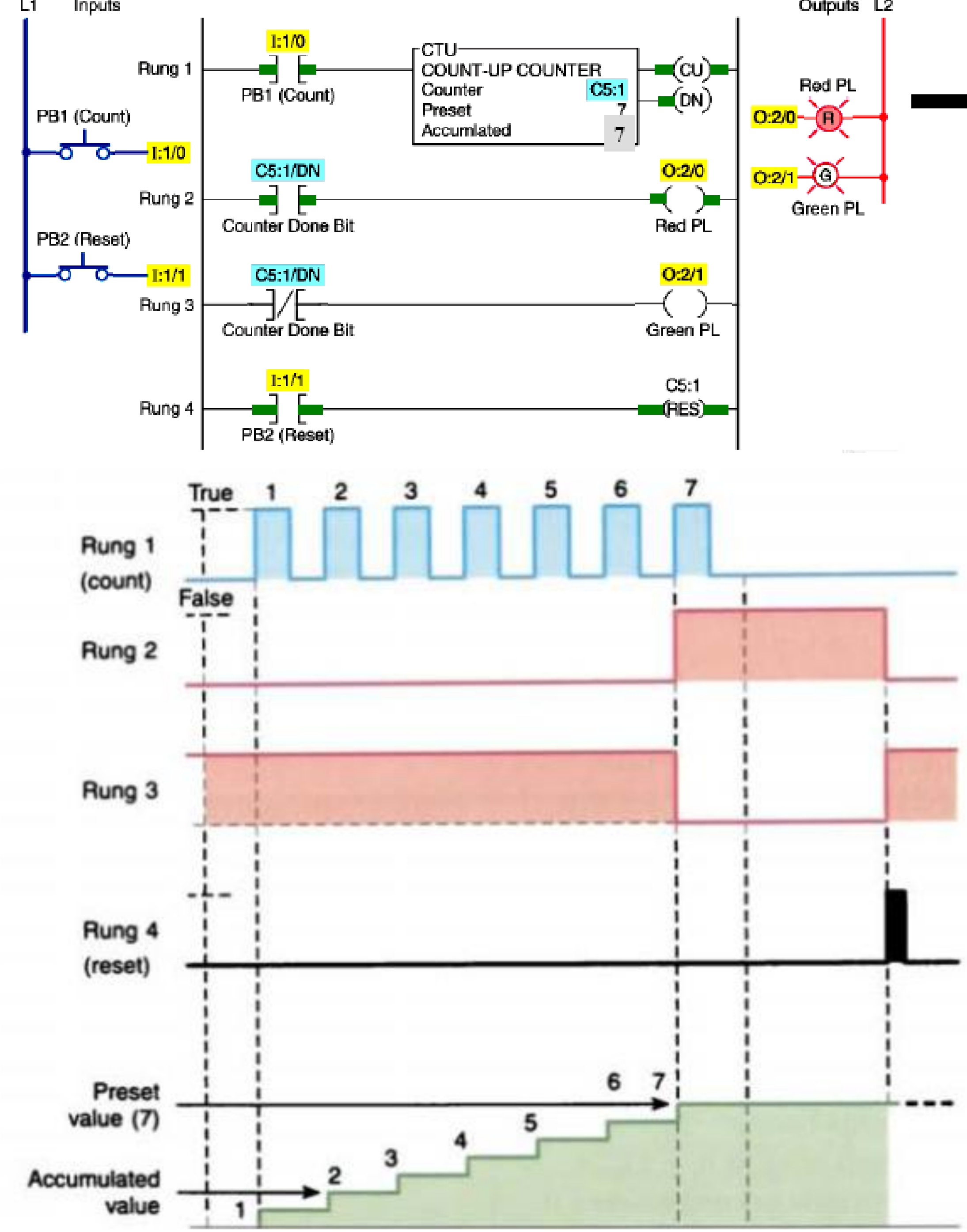
ControlLogix Count-Up Counter Instruction



In the PLC-5 and SLC 500, the max value for the preset and accumulated values is 32,767 and the min value is –32,768; for the ControlLogix controller the max value is 2,147,438,647 and the min value is –2,147,438,648.

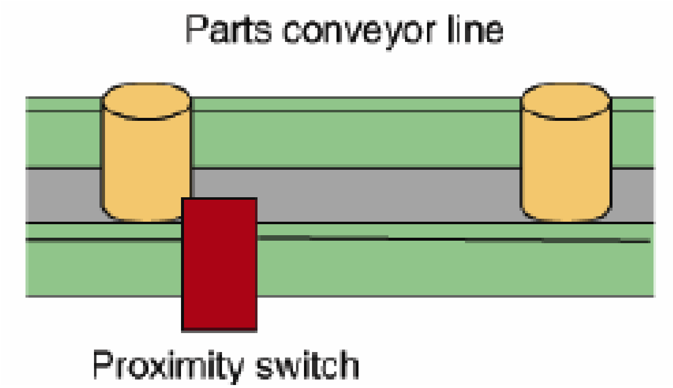
Simple Up-counter Program





L1
Inputs
Outputs
L2
Rung 1
I:1/0
PB1 (Count)
CTU
COUNT-UP COUNTER
Counter C5:1
Preset 7
Accumlated 7
(CU)
(DN)
PB1 (Count)
I:1/0
Rung 2
C5:1/DN
Counter Done Bit
O:2/0
Red PL
PB2 (Reset)
I:1/1
Rung 3
C5:1/DN
Counter Done Bit
O:2/1
Green PL
Rung 4
I:1/1
PB2 (Reset)
C5:1
(RES)
Red PL
O:2/0
R
O:2/1
G
Green PL
True
False
1 2 3 4 5 6 7
Rung 1
(count)
Rung 2
Rung 3
Rung 4
(reset)
Preset
value (7)
Accumulated
value

Parts Counting Program



Counter C5:2 counts the total number of parts coming off an assembly line for final packaging

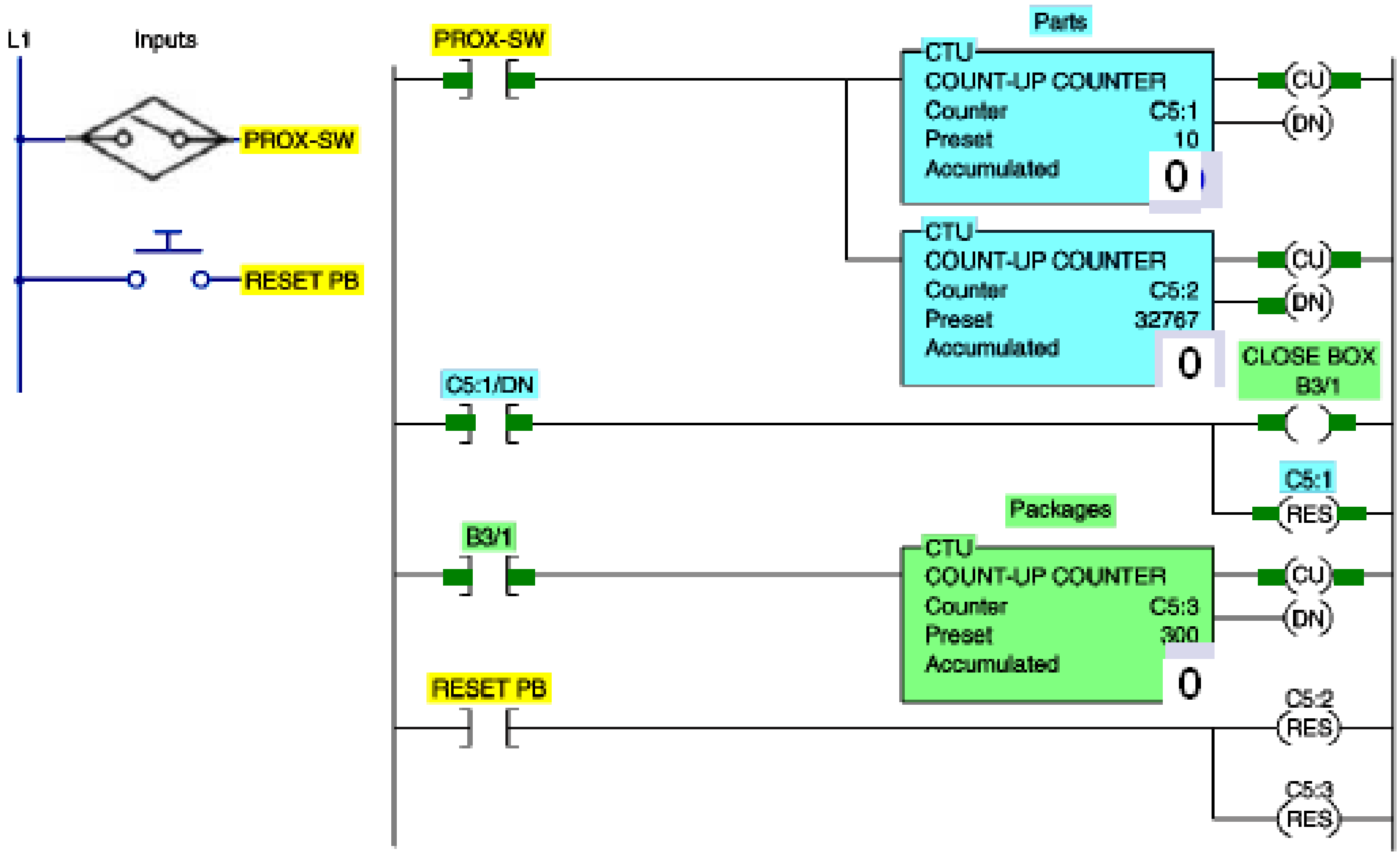
Each package must contain 10 parts

When 10 parts are detected, counter C5:1 sets bit B3/1 to initiate the box closing sequence

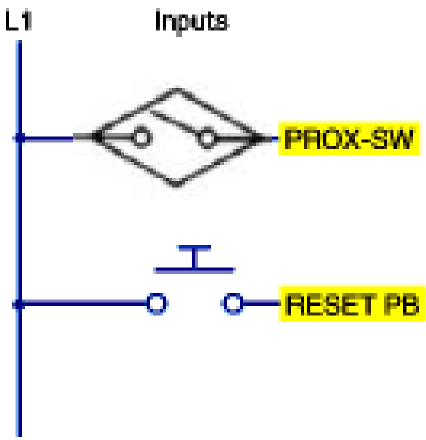
Counter C5:3 counts the total number of packages filled per day

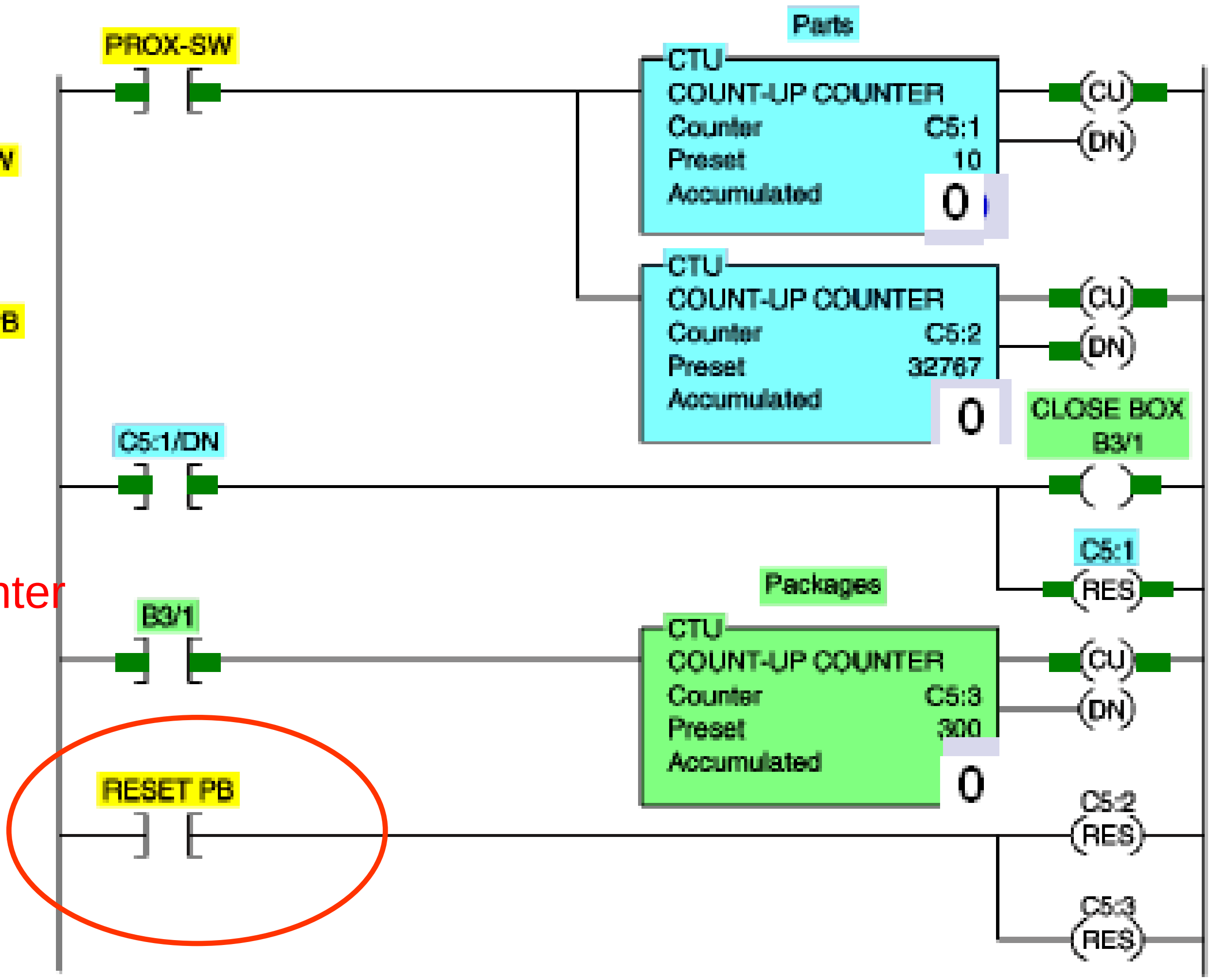
A pushbutton is used to restart the total part and package count from zero daily

Parts Counting Program



Parts Counting Program





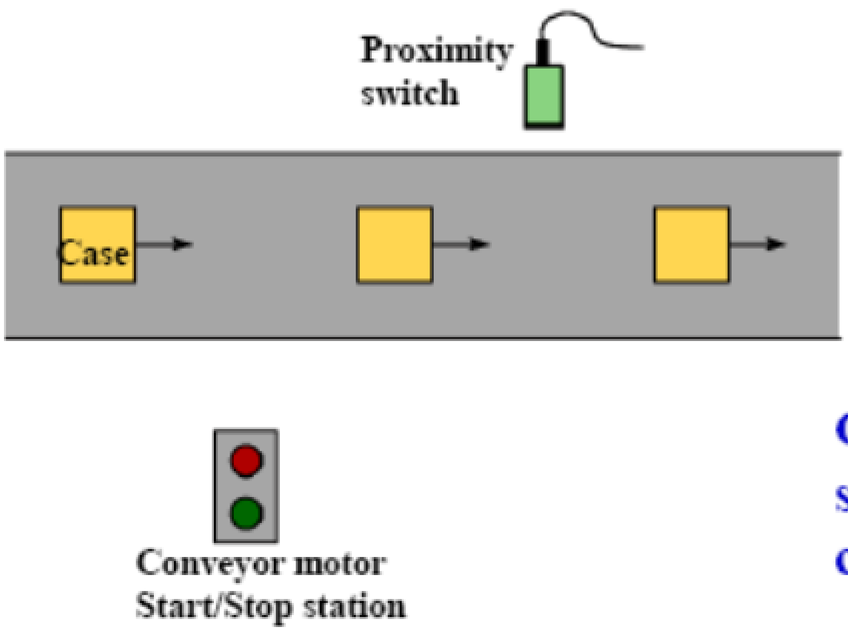
Can we rest the counter automatically?

Conveyor Motor Circuit

Ladder logic program

Inputs

Conveyor Motor Circuit That Uses A Programmed One-Shot Reset Circuit



Sequential Task:

The start button is pressed to start the conveyor motor

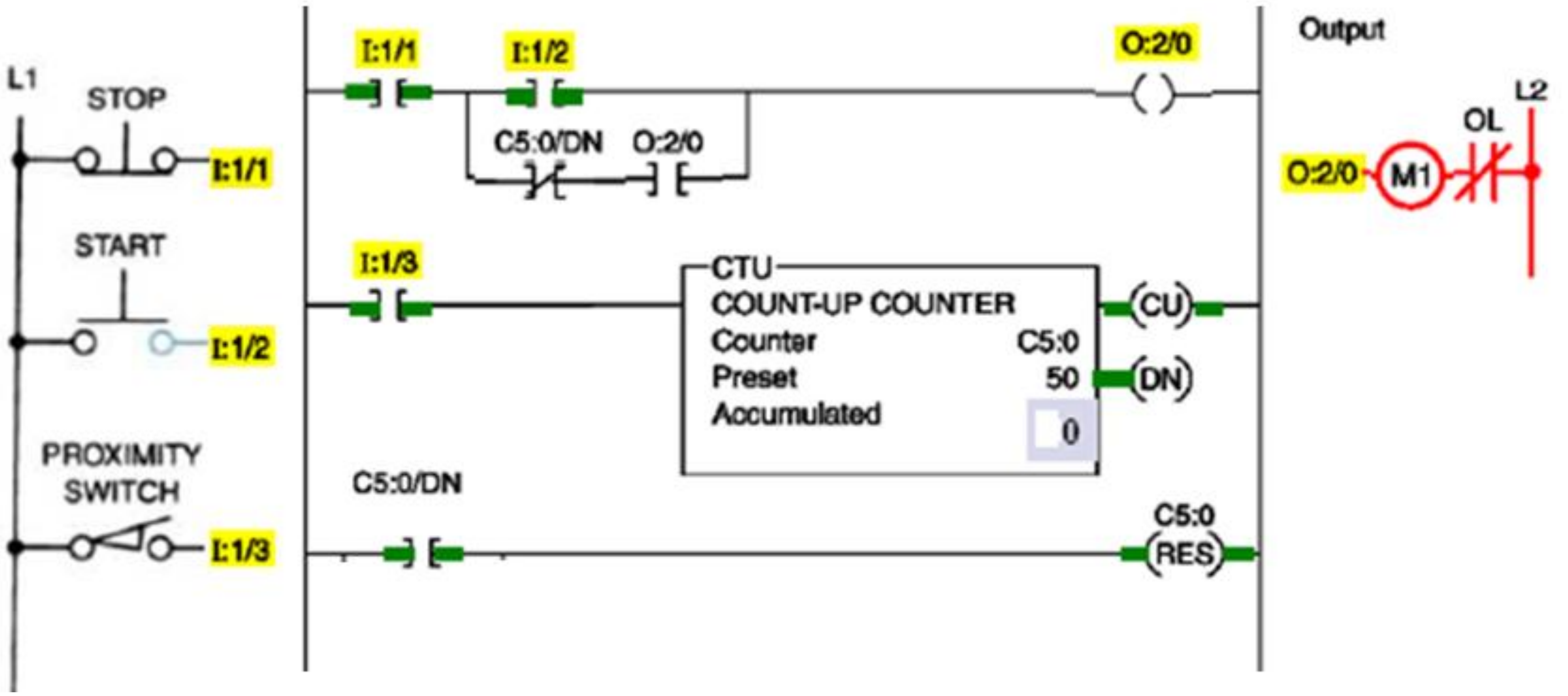
Cases move pass the proximity switch and increment the counter's accumulated value

After a count of 50, the conveyor motor stops automatically and the counter's accumulated value is reset to zero

The conveyor motor can be stopped or started manually at anytime without loss of the accumulated count

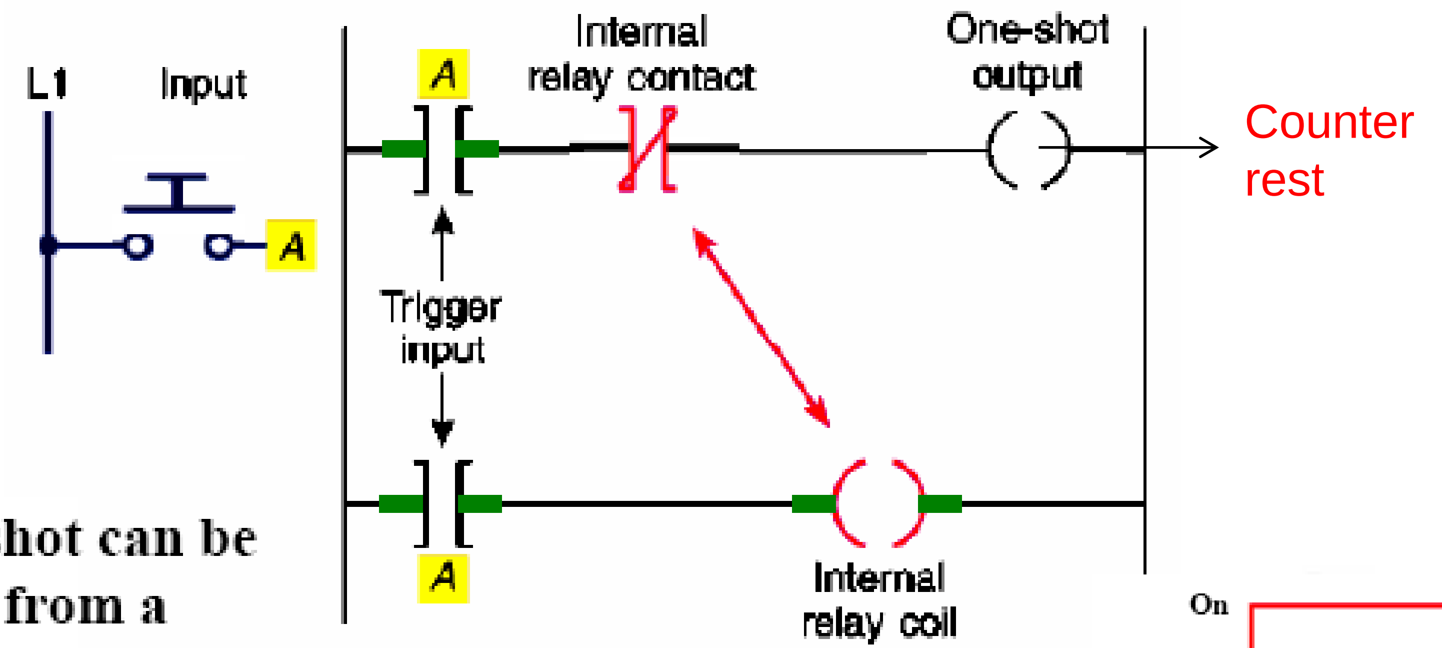
Solution to be added later

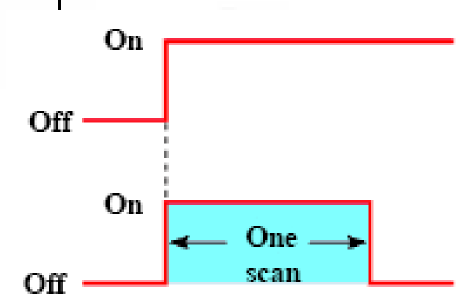
.


L1
STOP
I:1/1
START
I:1/2
PROXIMITY
SWITCH
I:1/3
I:1/1
I:1/2
O:2/0
C5:0/DN
O:2/0
I:1/3
CTU
COUNT-UP COUNTER
Counter C5:0
Preset 50
Accumulated 0
(CU)
(DN)
C5:0/DN
C5:0
(RES)
Output
L2
OL
O:2/0
M1

One-Shot, Or Transitional, Contact Program

The transitional or one shot contact program can be used to automatically clear or reset a counter. The program is designed to generate an output pulse that, when triggered, goes on for the duration of one program scan and then goes off.





The one-shot can be triggered from a momentary signal, or one that comes on and stays on for some time.

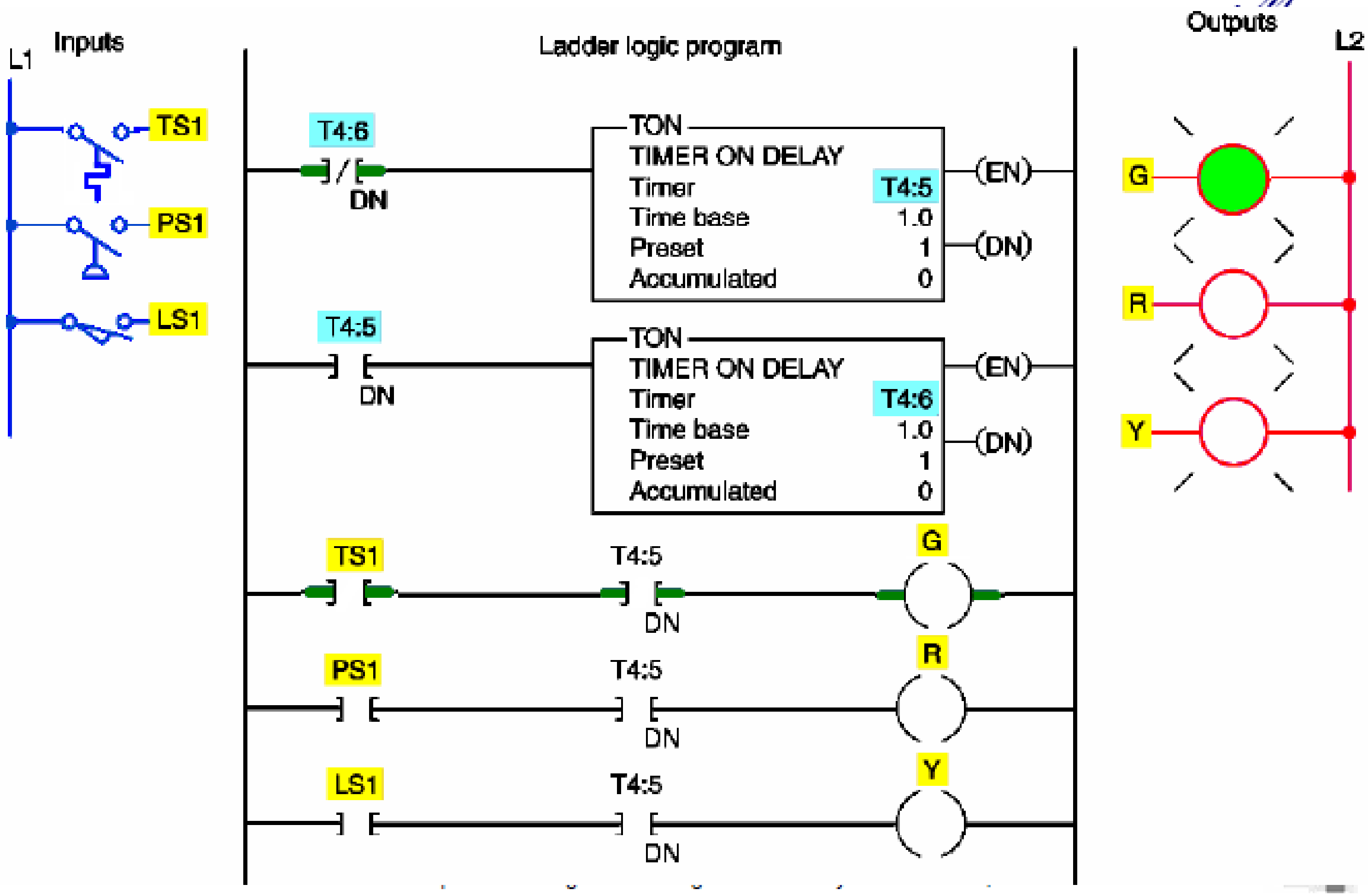
Alarm Monitor Program

- The alarm is triggered by the closing of liquid level switch LS1
- The light will flash whenever the alarm condition is triggered and has not been acknowledged, even if the alarm condition clears in the meantime
- The alarm is acknowledged by closing selector switch SS1
- The light will operate in the steady mode when the alarm trigger condition exists but has been acknowledged

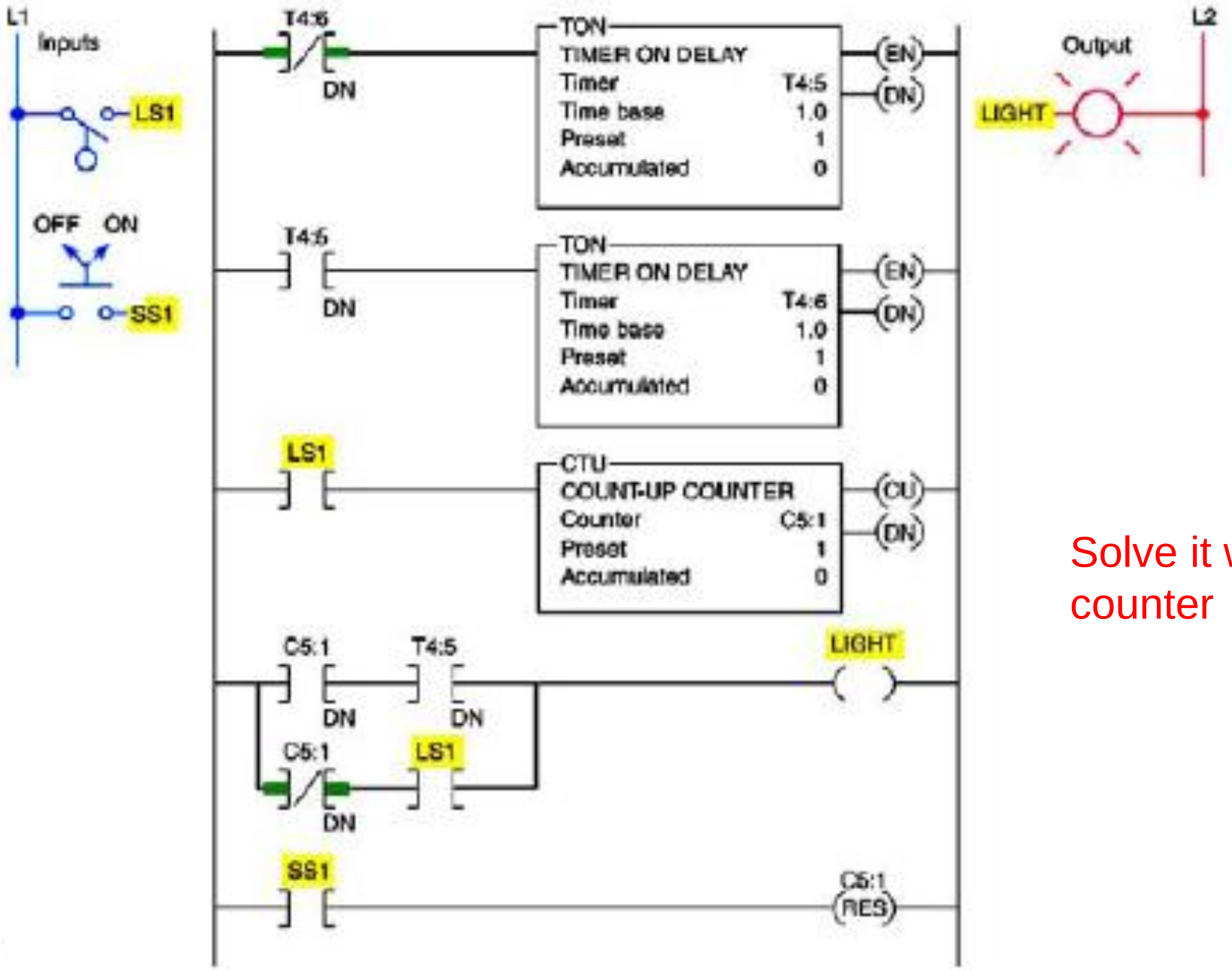
Solution to be added later

.

Annunciator Flasher Circuit

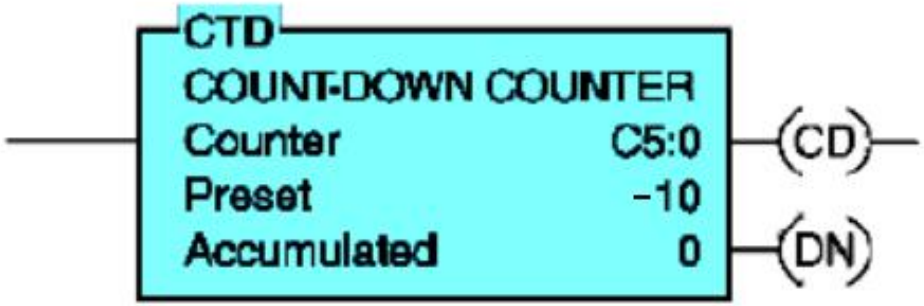


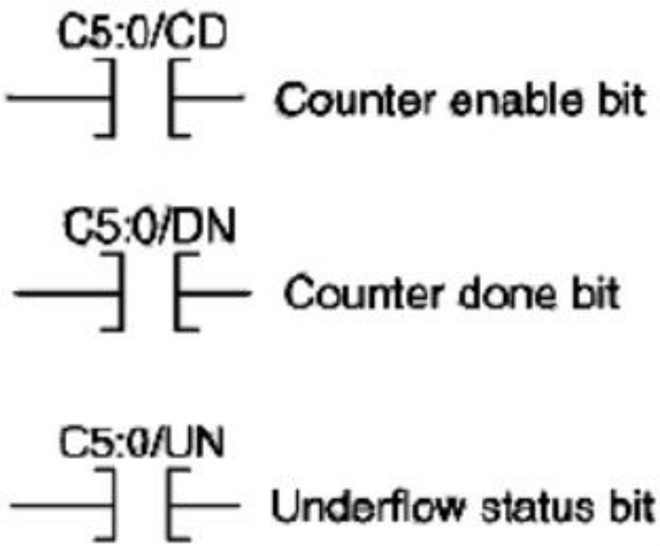
Alarm Monitor Program



Solve it without counter

PLC-5 And SLC-500 Count-Down Counter Instruction



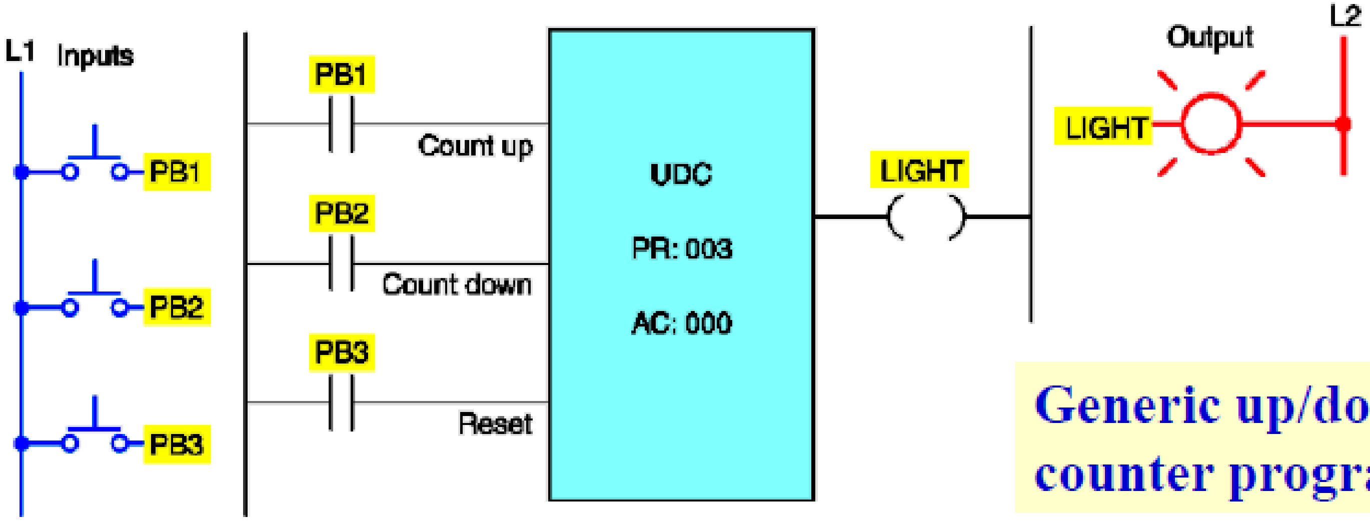


The reset instruction resets the counter's accumulated value back to zero.

If the accumulation value is below the minimum range then the underflow (UN) bit will be true.

Down-Counter

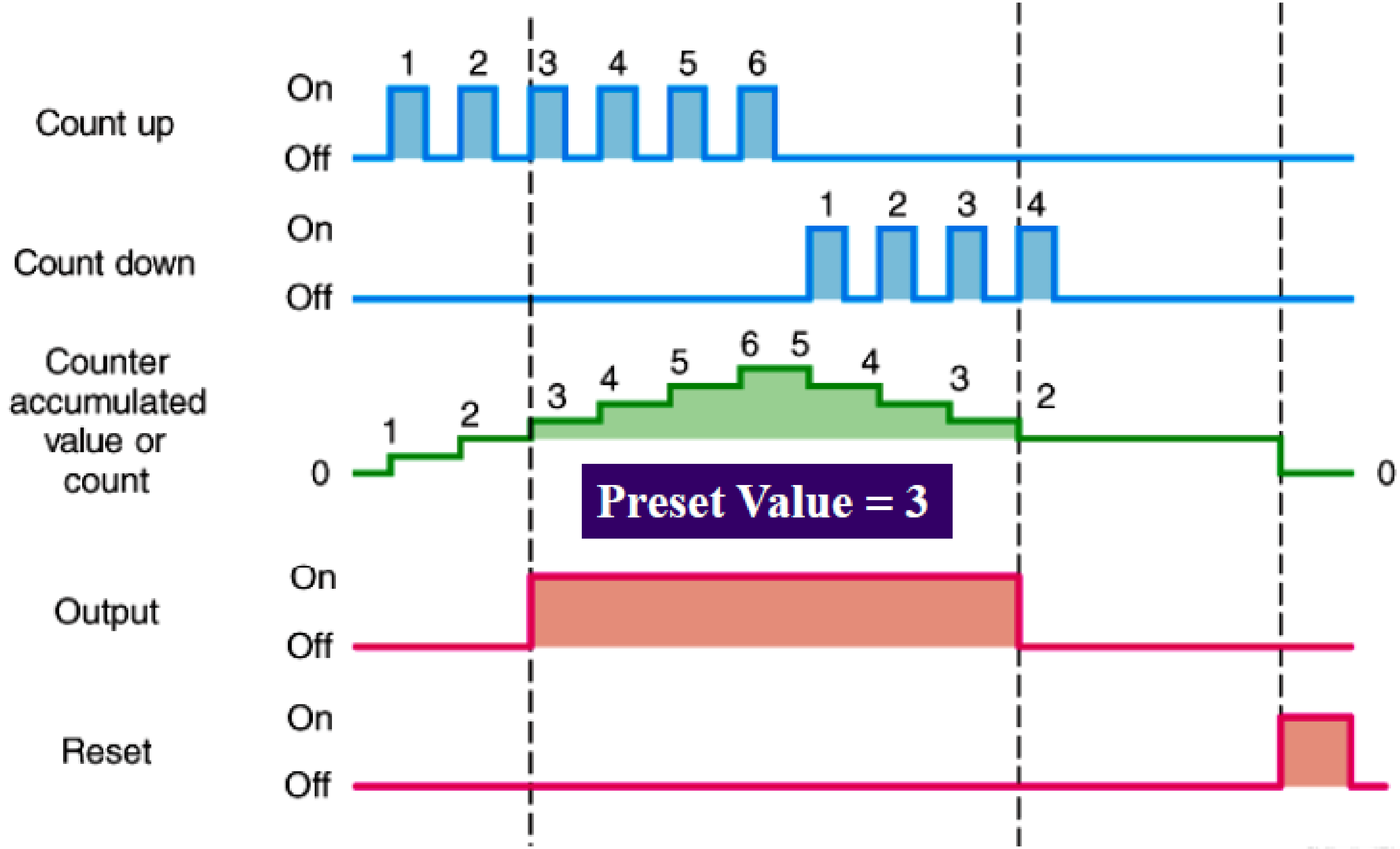
The down-counter output instruction will count down or decrement by 1 each time the counted event occurs. Each time the down-count event occurs, the accumulated value is decremented. Normally the down-counter is used in conjunction with the up counter to form an up/down counter.



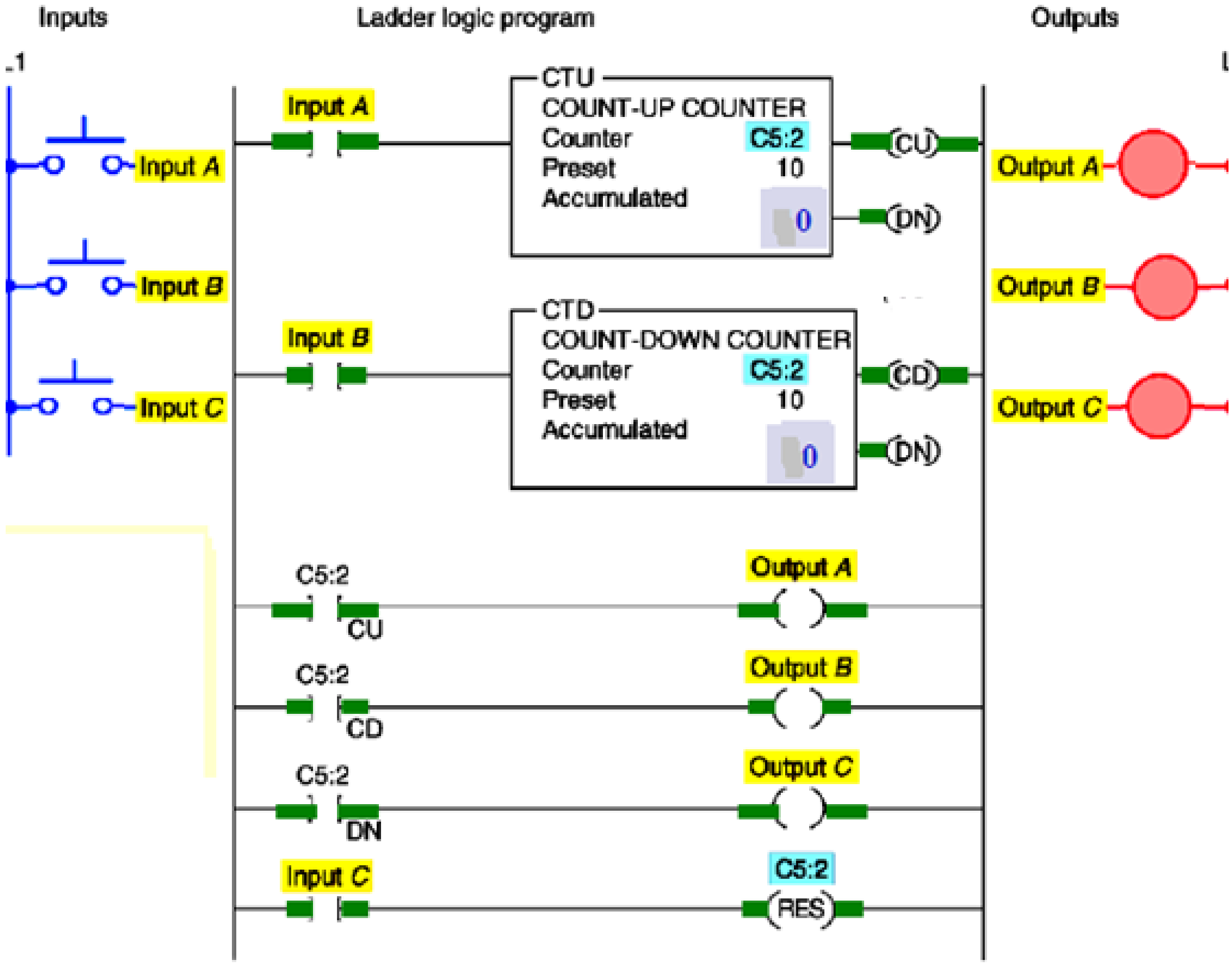
Generic up/down counter program

Up/Down Counter Timing diagram

Radu Muresan



Up/Down-Counter Program



What dose this program do?

Up/Down-Counter Program

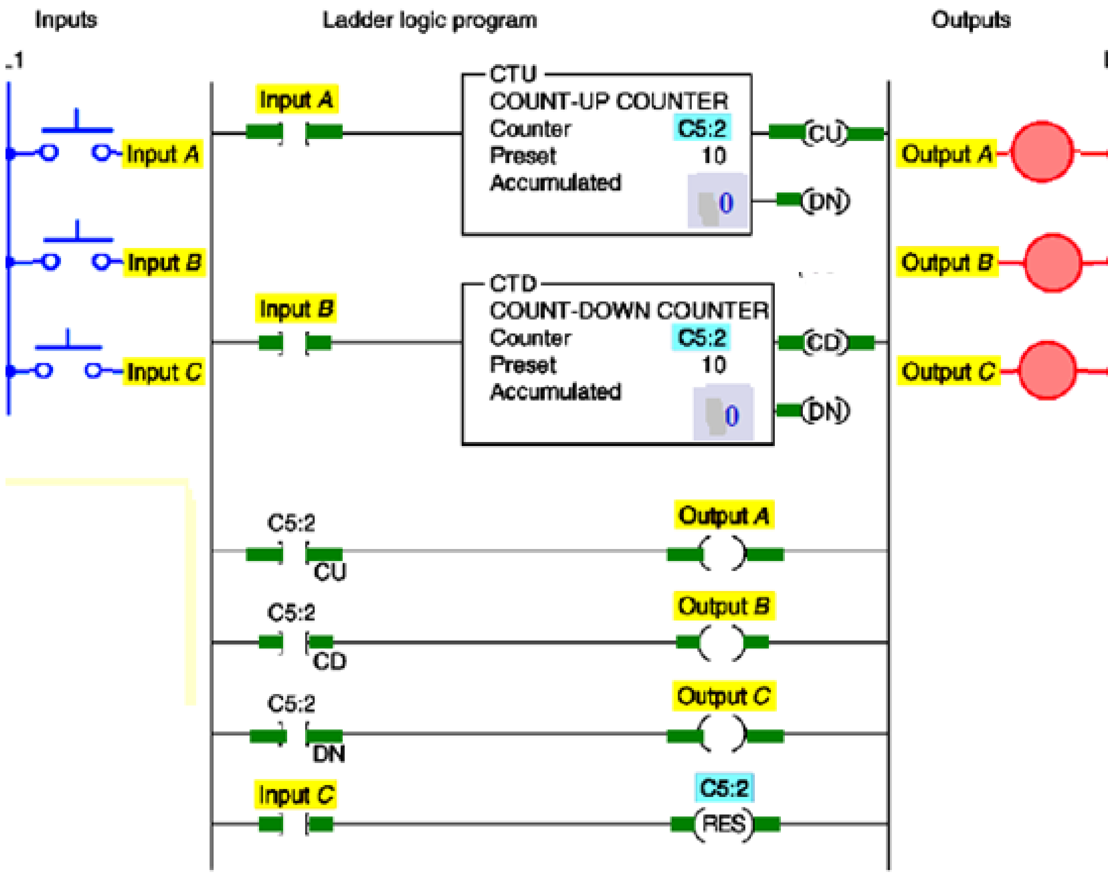


Figure 8-22 shows an up/down-counter program that will increase the counter's accumulated value when pushbutton PB1 is pressed and will decrease the counter's accumulated value when pushbutton PB2 is pressed. Note that the same address is given to the *up-counter* instruction, the down-counter instruction, and the *reset* instruction. All three instructions will be looking at the *same address* in the counter file. When input *A* goes from false to true, one count is added to the accumulated value. When input *B* goes from false to true, one count is subtracted from the accumulated value. The operation of the program can be summarized as follows:

- When the CTU instruction is true, C5:2/CU will be true, causing output *A* to be true.
- When the CTD instruction is true, C5:2/CD will be true, causing output *B* to be true.
- When the accumulated value is greater than or equal to the preset value, C5:2/DN will be true, causing output *C* to be true.

Input *C* going true will cause both counter instructions to reset. When *reset* by the RES instruction, the accumulated value will be reset to 0 and the done bit will be reset.

Parking Garage Counter Program

Parking Garage Counter Program

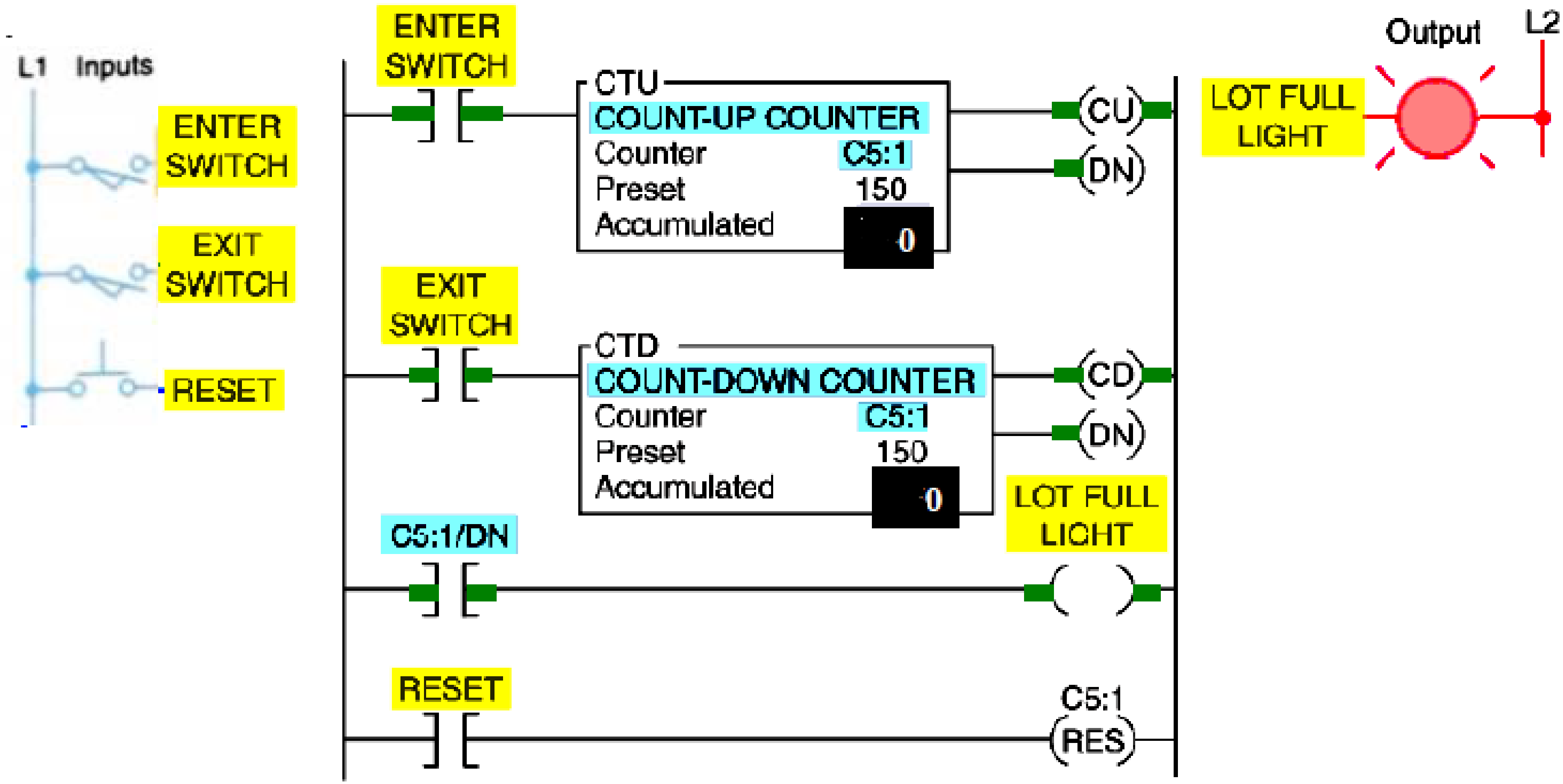
- As a car enters, it triggers the up-counter output instruction and increments the accumulated count by 1.
- As a car leaves, it triggers the down-counter output instruction and decrements the accumulated count by 1.
- Since both the up- and down-counters have the same address, the accumulated value will be the same in both.
- Whenever the accumulated value equals the preset value, the counter output is energized to light up the Lot Full sign.

Solution to be added later

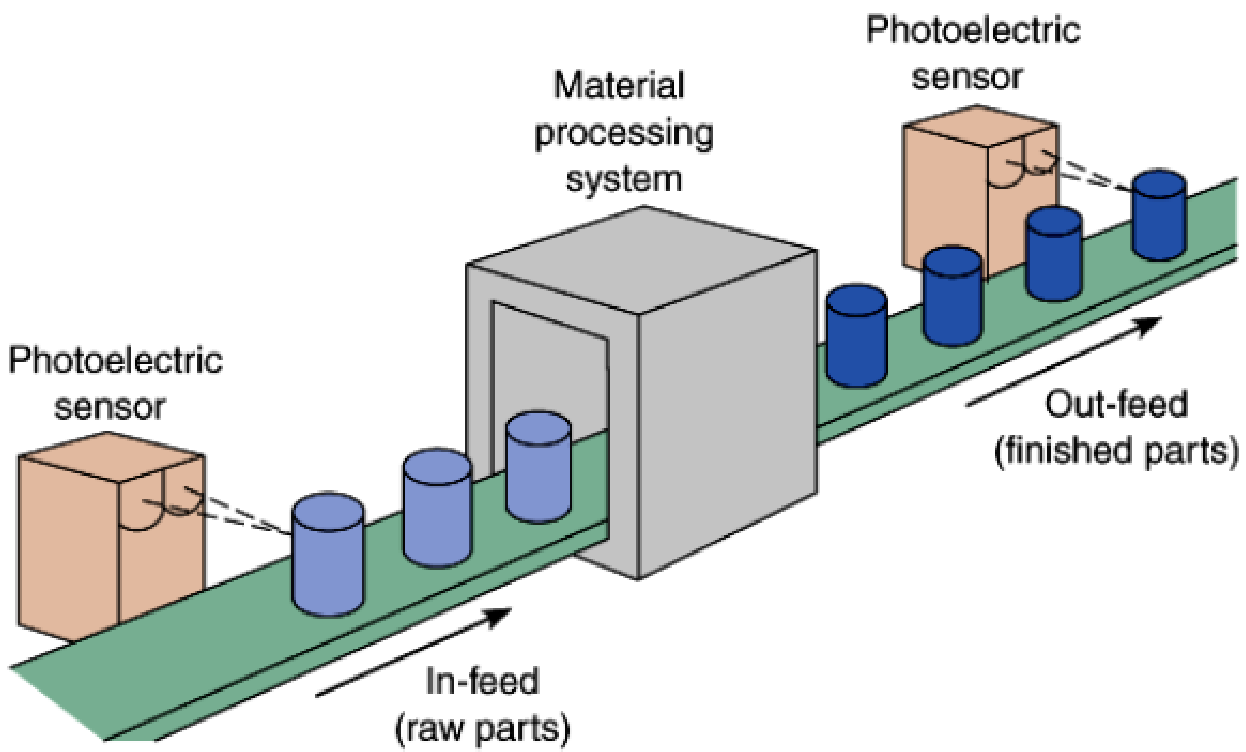
.

Parking Garage Counter Program

Ladder logic program



In-Process Monitoring System

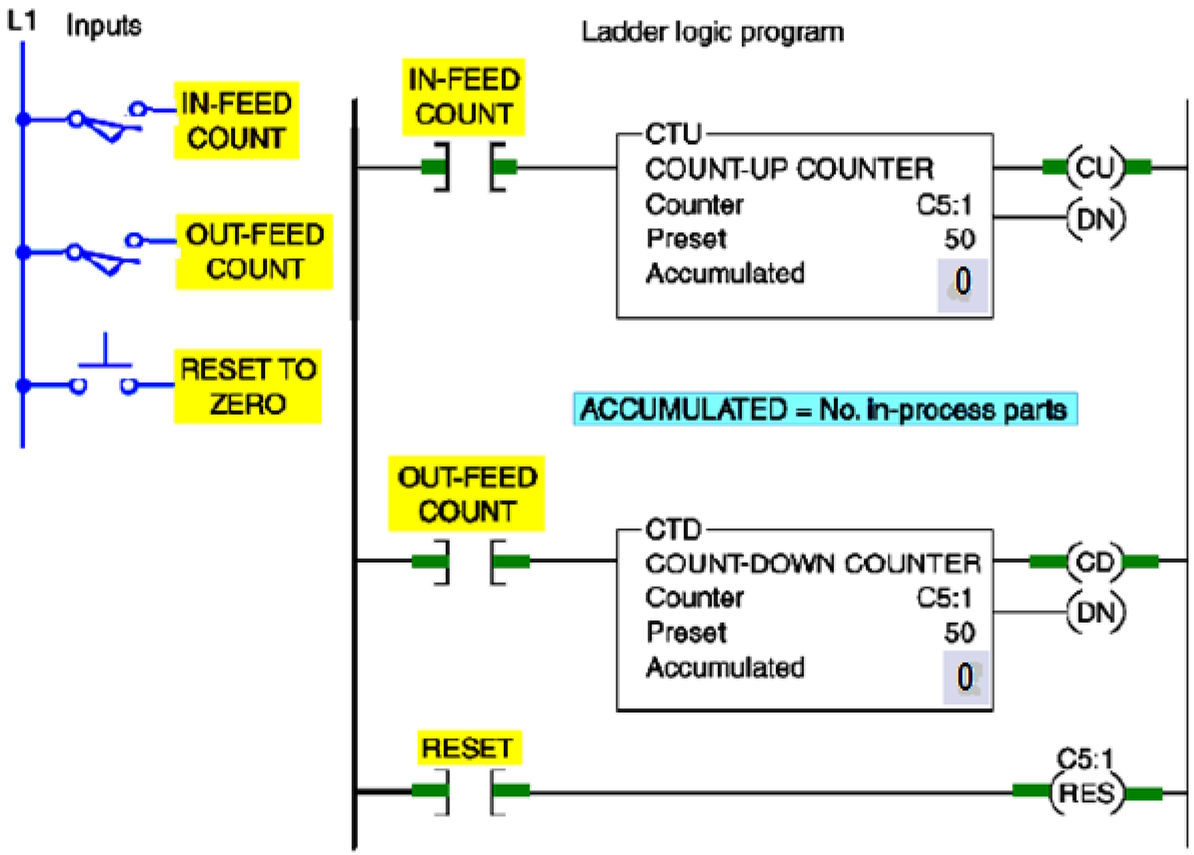


Before start-up, the system is completely empty of parts, and the counter is reset manually to zero.

Design a PLC program that find the number of in process parts?

Solution to be added later

.



Before start-up, the system is completely empty of parts, and the counter is reset manually to 0.

When the operation begins, raw parts move through the in-feed sensor, with each part generating an up count.

After processing, finished parts appearing at the out-feed sensor generate down counts, so the accumulated count of the counter continuously indicates the number of in-process parts.

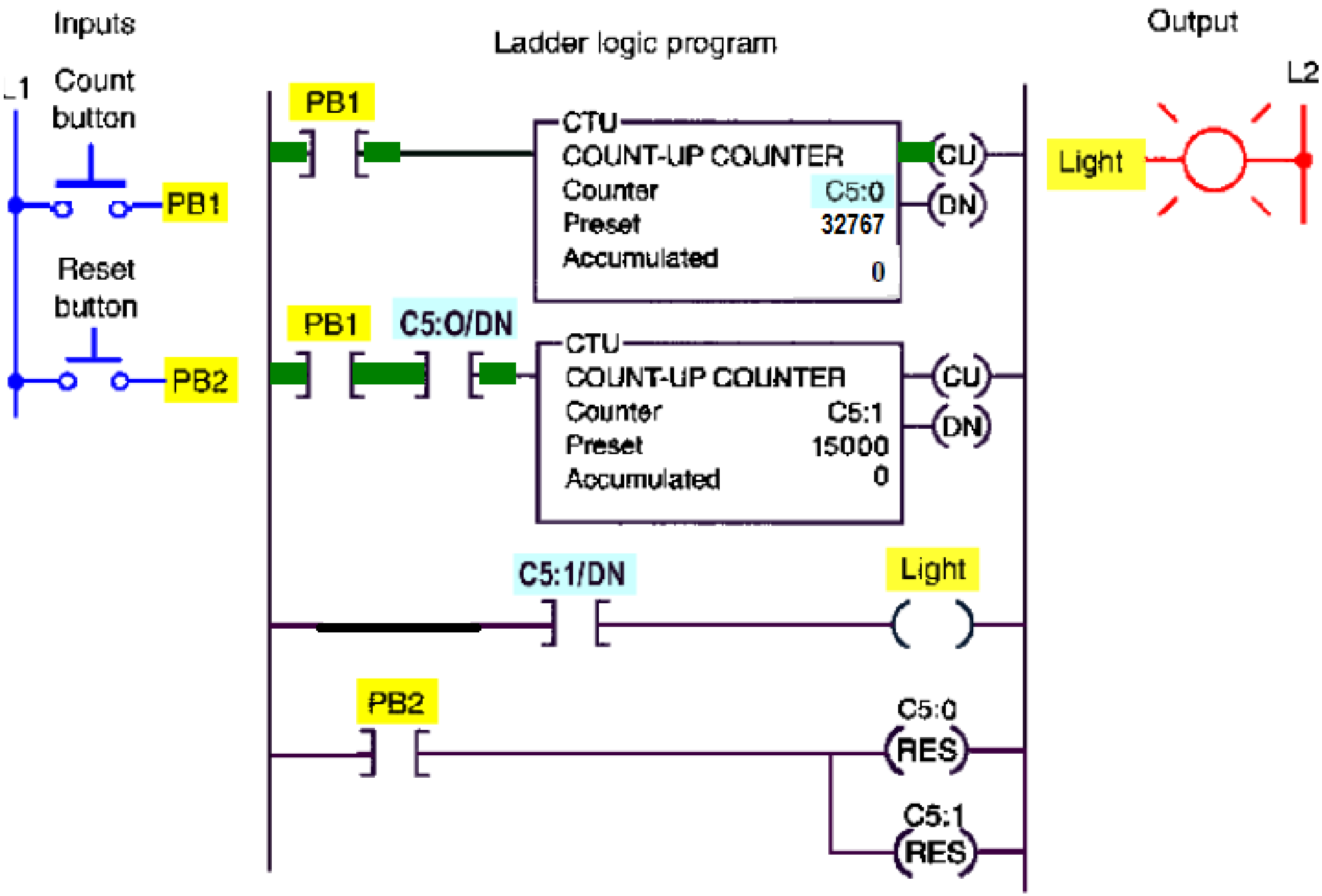
Cascading Counters

Depending on the application, it may be necessary to count events that exceed the maximum number allowable per counter instruction. One way of accomplishing this is by interconnection, or *cascading*, two counters.

Counting Beyond The Maximum Count

HOW?

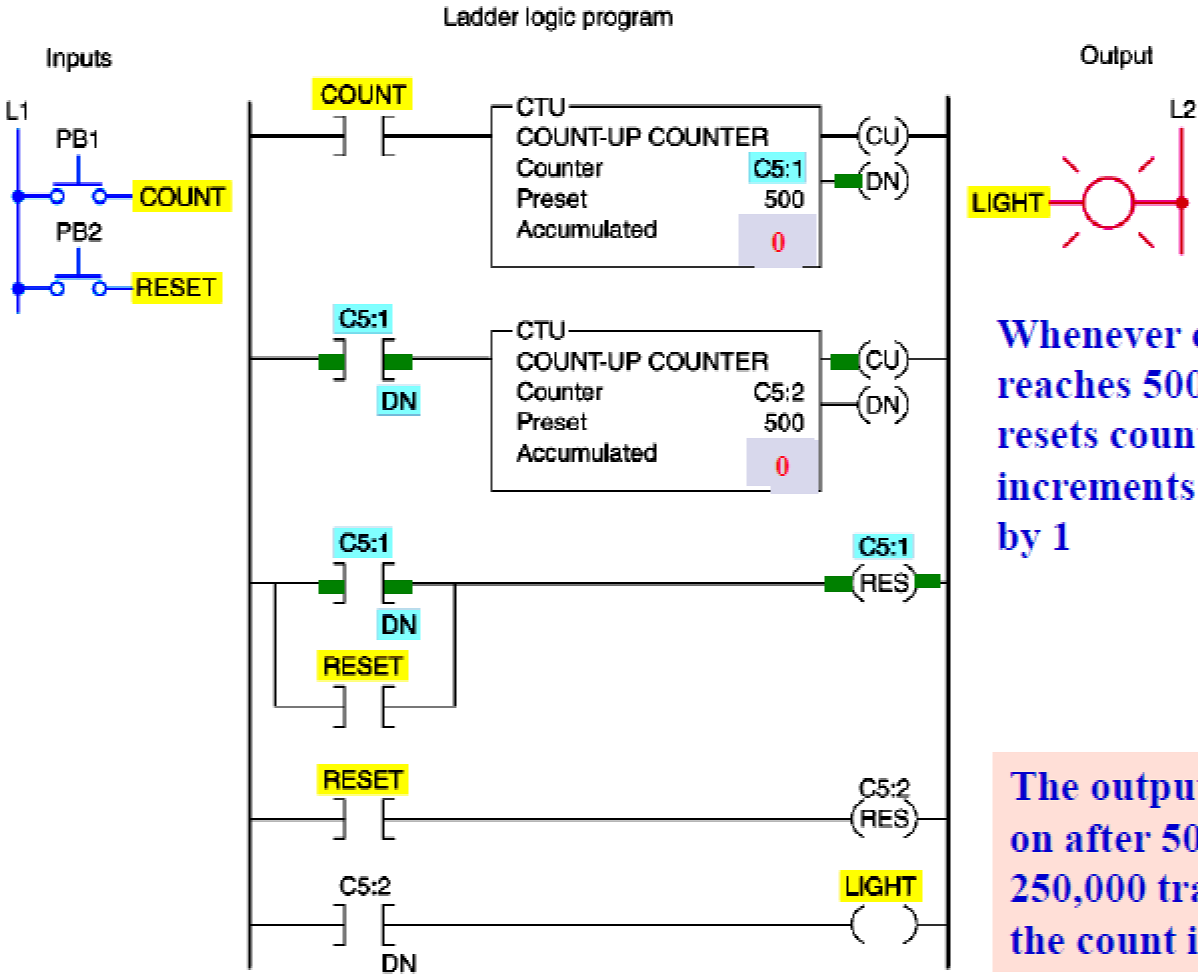
Counting Beyond The Maximum Count



Cascading Counters For Extremely Large Counts

Another way?

Cascading Counters For Extremely Large Counts

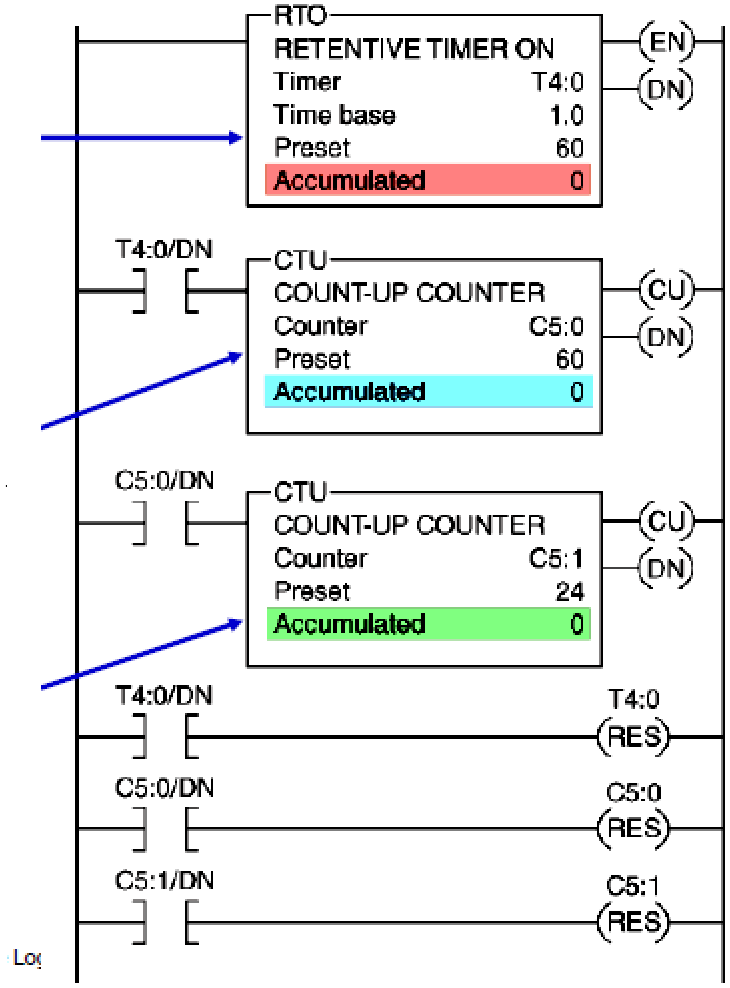


Whenever counter C5:1 reaches 500, its done bit resets counter C5:1 and increments counter C5:2 by 1

The output light turns on after 500 x 500, or 250,000 transitions of the count input

What does this program do?

Ladder logic program

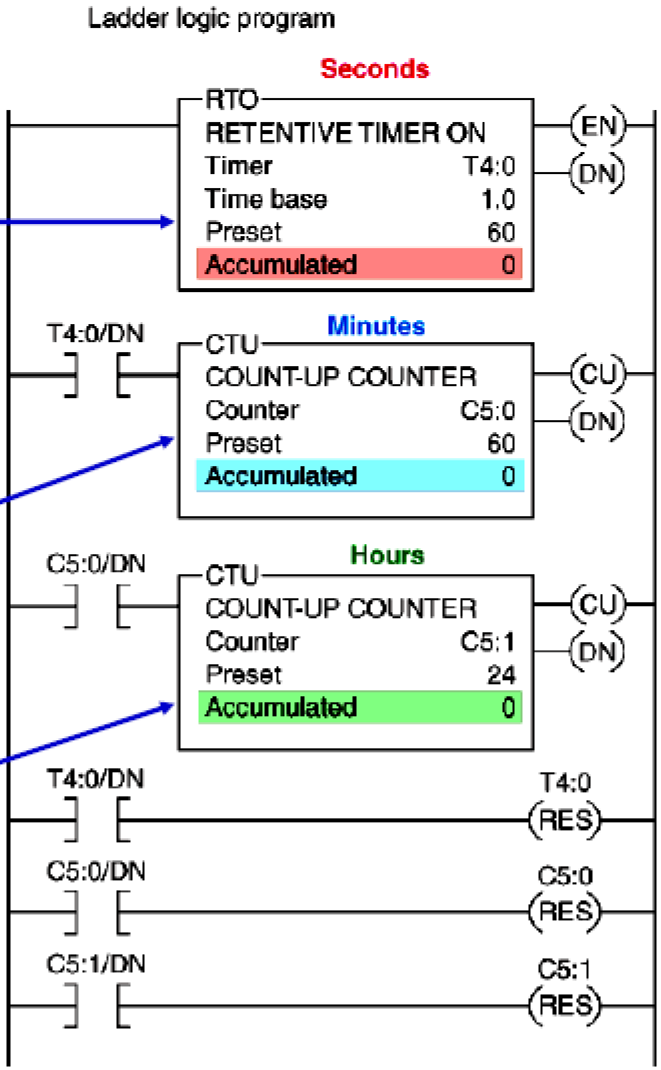


24 Hour Clock Program

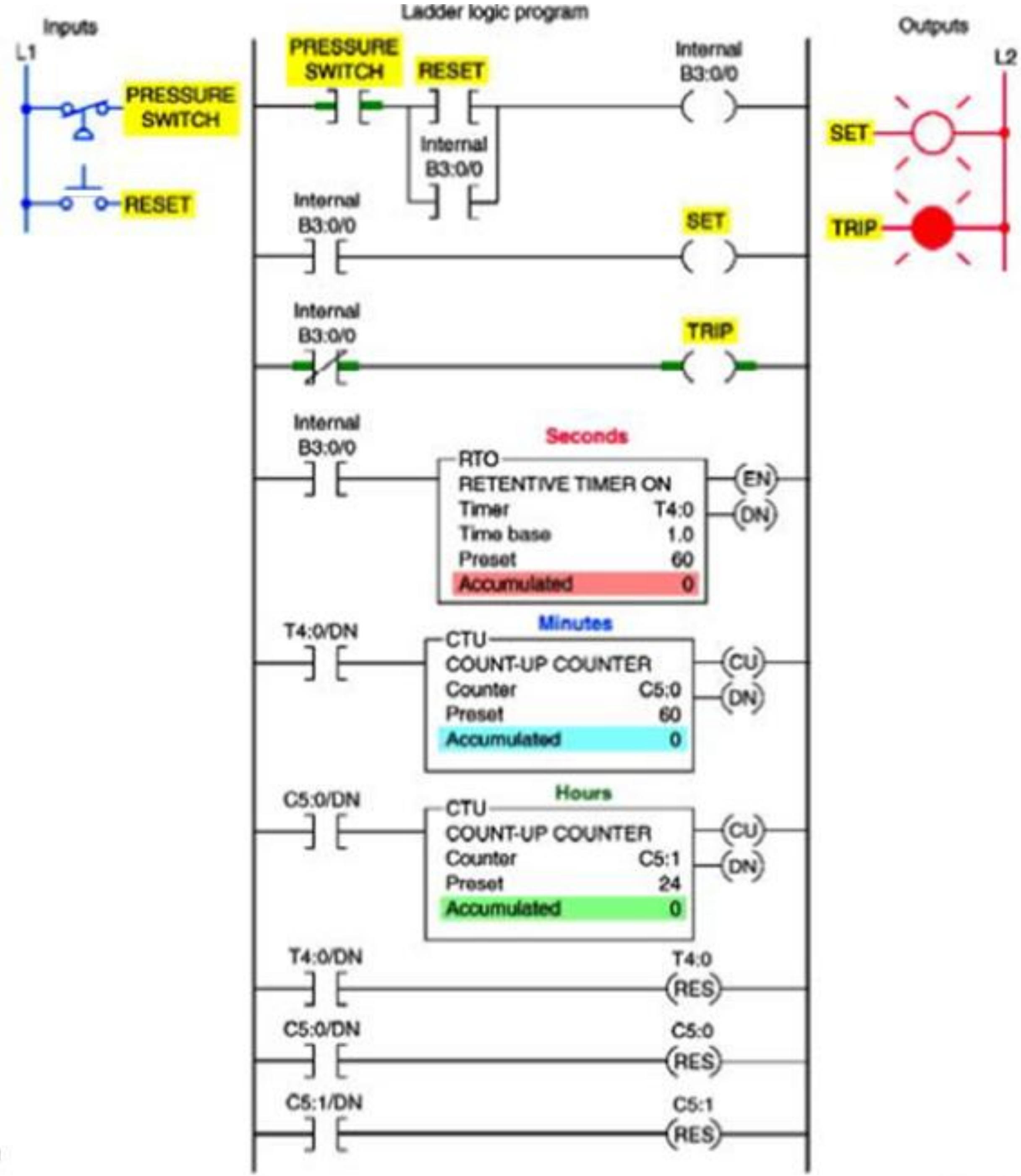
The timer times for a 60 s period, after which its done bit is set. This, in turn cases C5:0 to increment 1 count. On the next processor scan, the timer is reset and begins timing again.

Whenever C5:0 reaches its preset value of 60, its done bit is set. This, in turn causes it to reset itself and C5:1 to increment 1 count.

Whenever C5:1 reaches its preset value of 24, its done bit is set to reset itself.

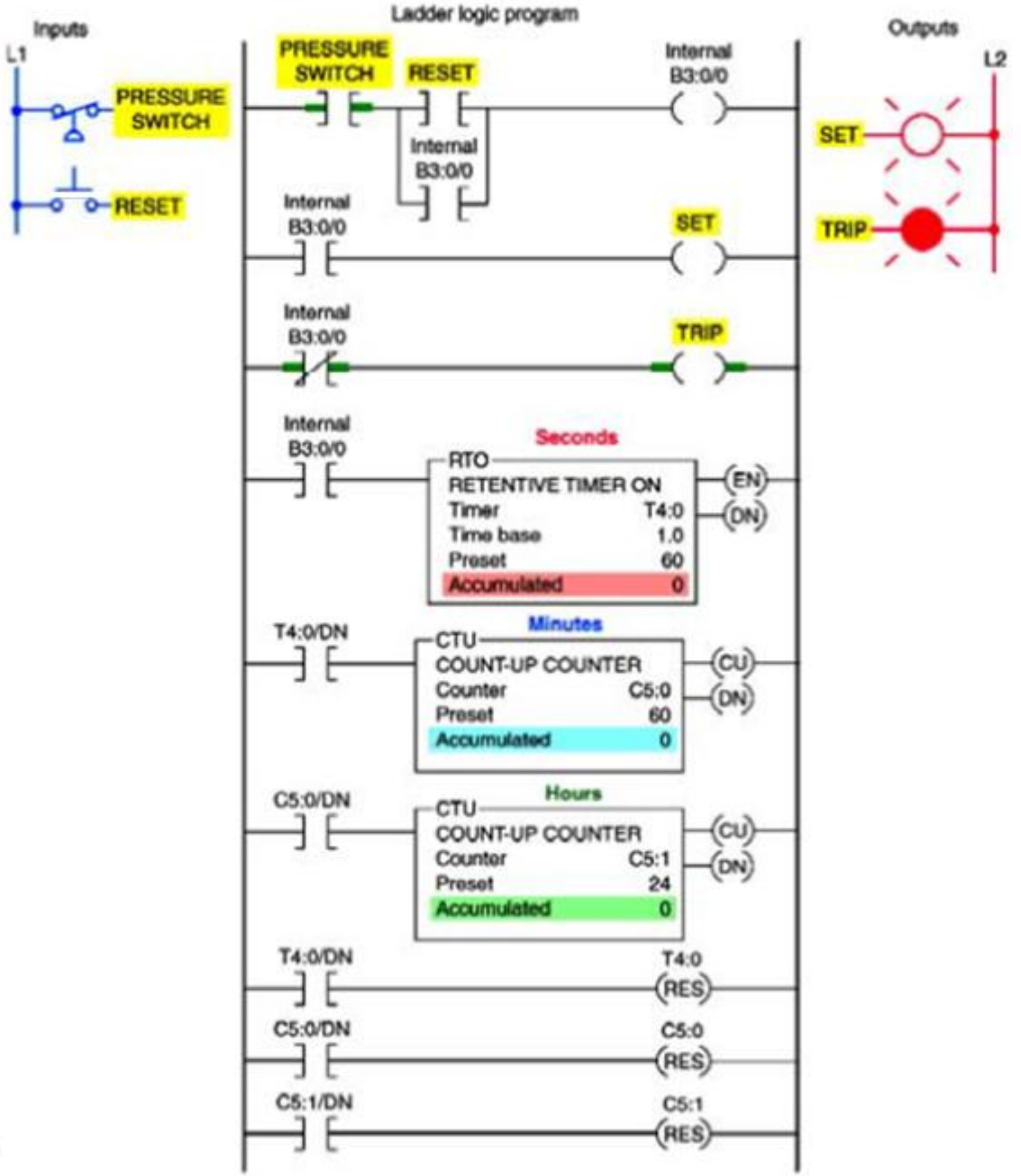


1- What is this program doing?

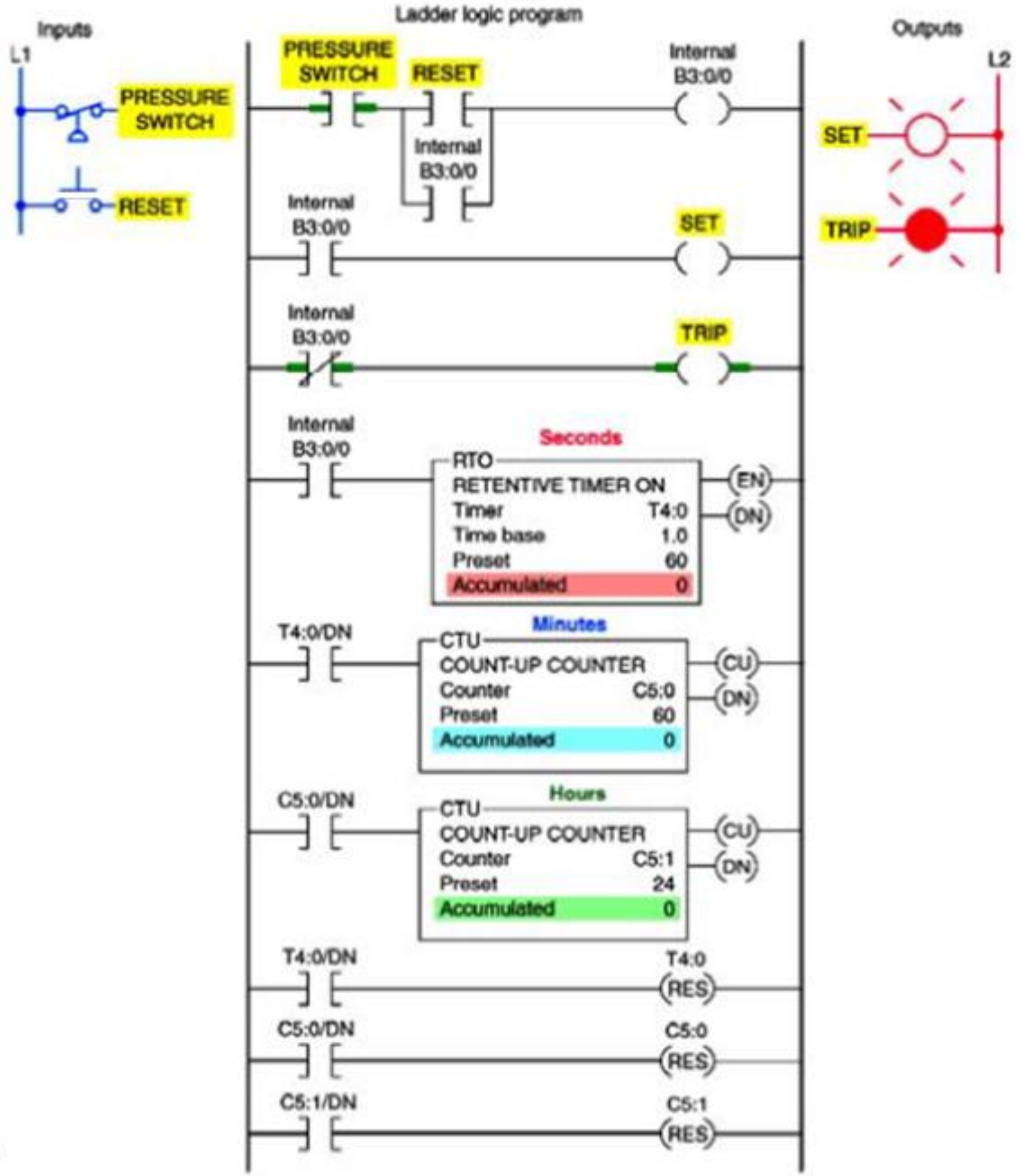


1- What is this program doing

Program For Monitoring The Time Of An Event

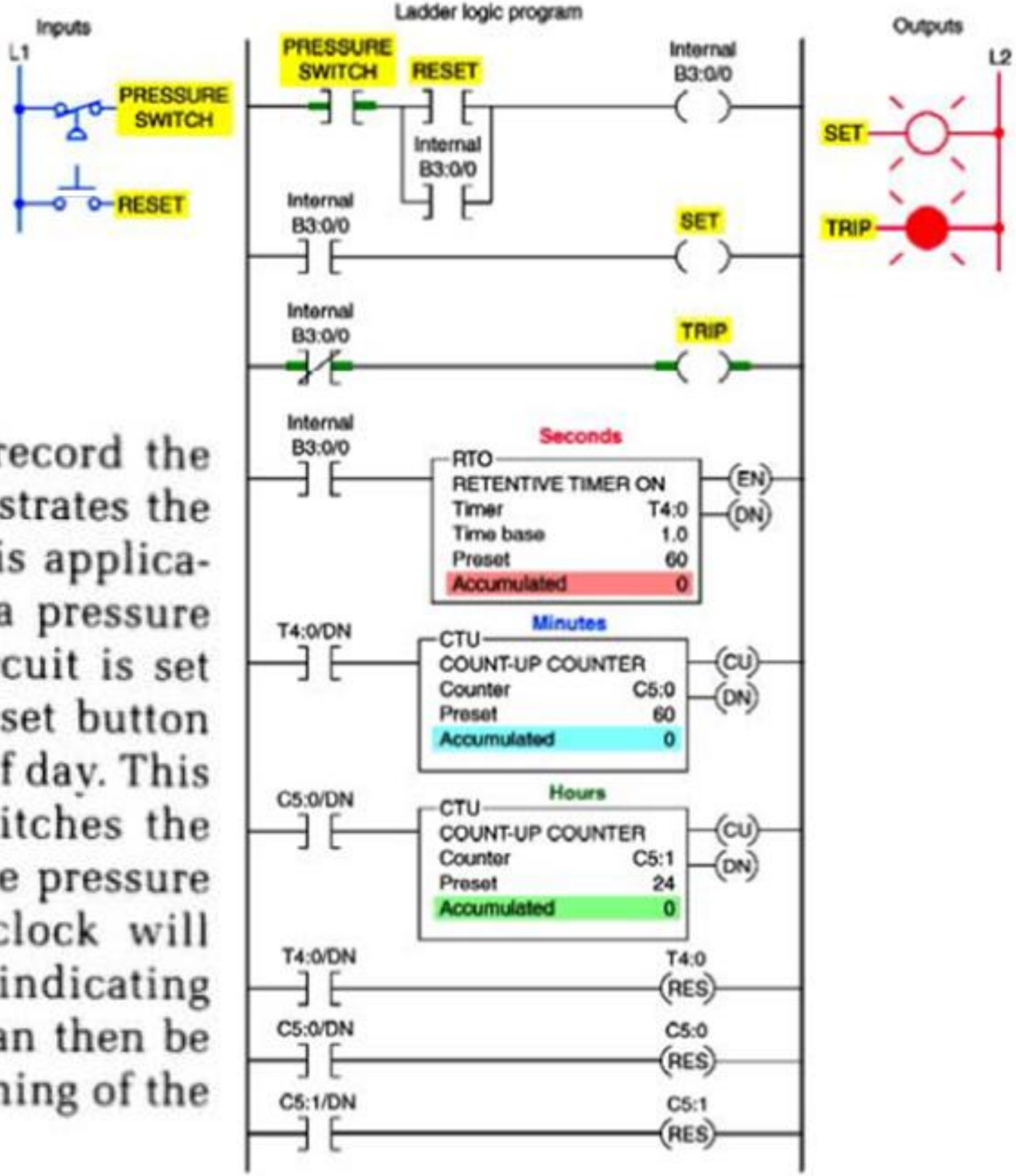


Program For Monitoring The Time Of An Event



2- How?

Program For Monitoring The Time Of An Event

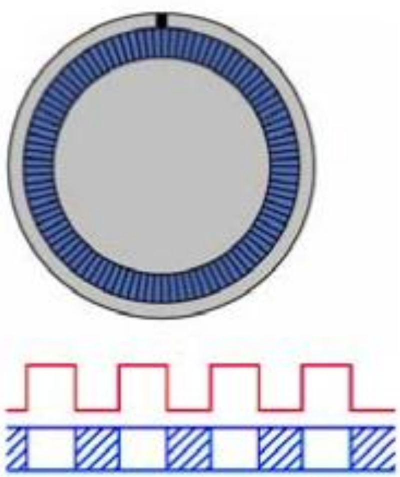


The 24-h clock can be used to record the time of an event. Figure 8-27 illustrates the principle of this technique. In this application the time of the opening of a pressure switch is to be recorded. The circuit is set into operation by pressing the reset button and setting the clock for the time of day. This starts the 24-hour clock and switches the set indicating light on. Should the pressure switch open at any time, the clock will automatically stop and the trip indicating light will switch on. The clock can then be read to determine the time of opening of the pressure switch.

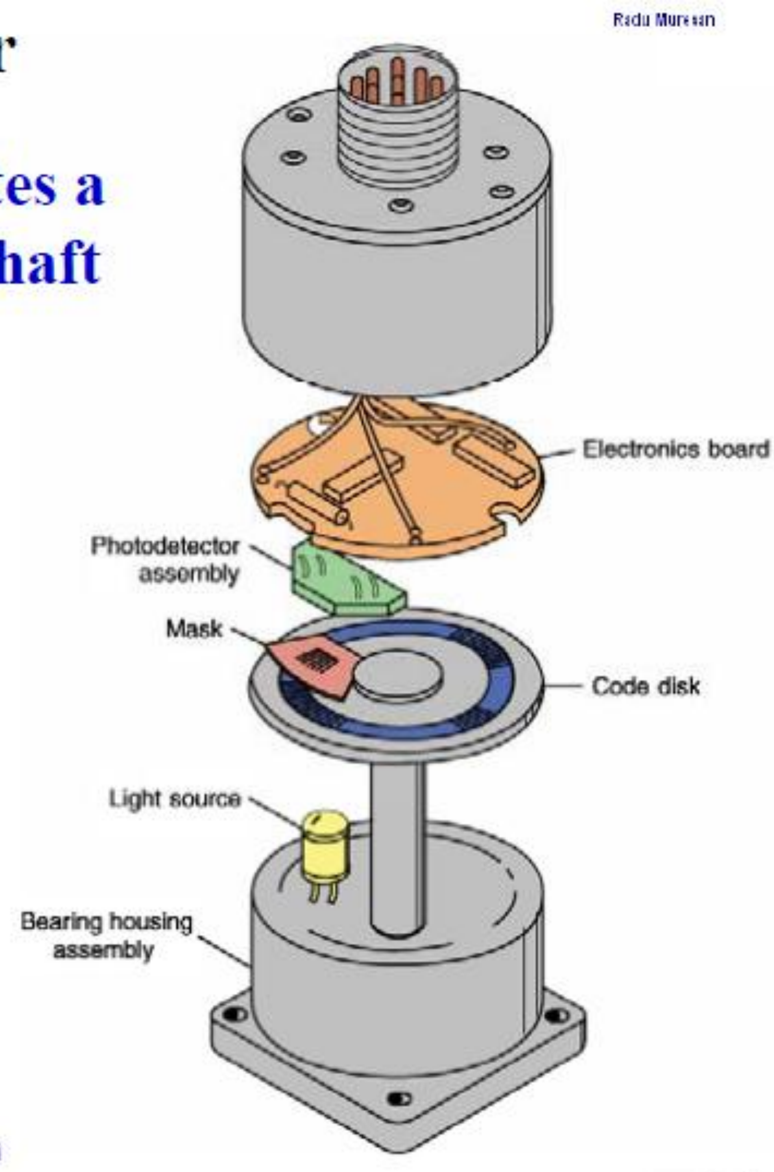
Incremental Encoder

An incremental encoder creates a series of square waves as its shaft is rotated.

The encoder disk interrupts the light as the encoder shaft is rotated to produce the square wave output waveform.

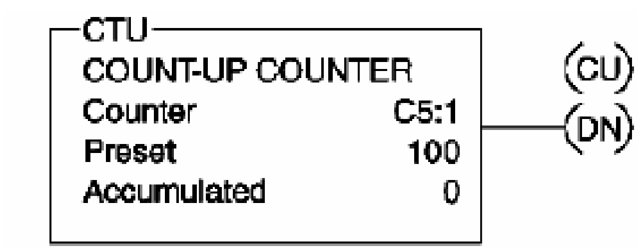


EN … n Programmable Logi



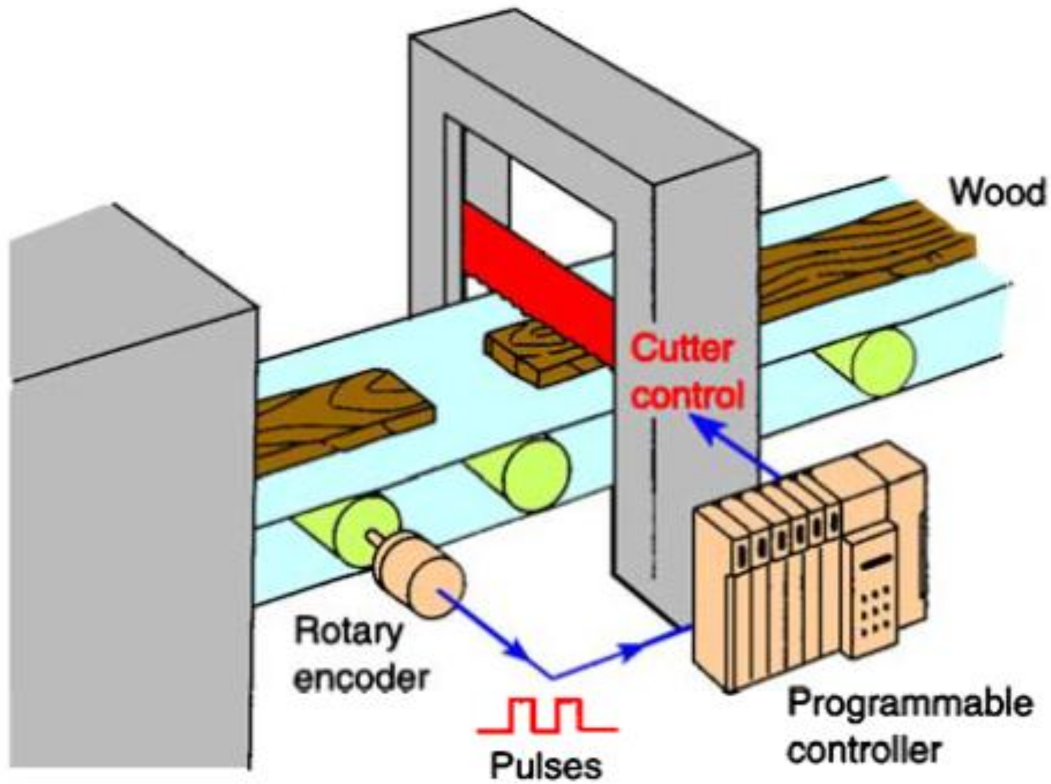
Incremental Encoder

The number of square waves obtained from the output of the encoder can be made to correspond to the mechanical movement required.



To divide a shaft revolution into 100 parts, an encoder could be selected to supply 100 square wave cycles per revolution. By using a counter to those cycles, we could tell how far the shaft has rotated.

Cutting Objects To A Specific Size

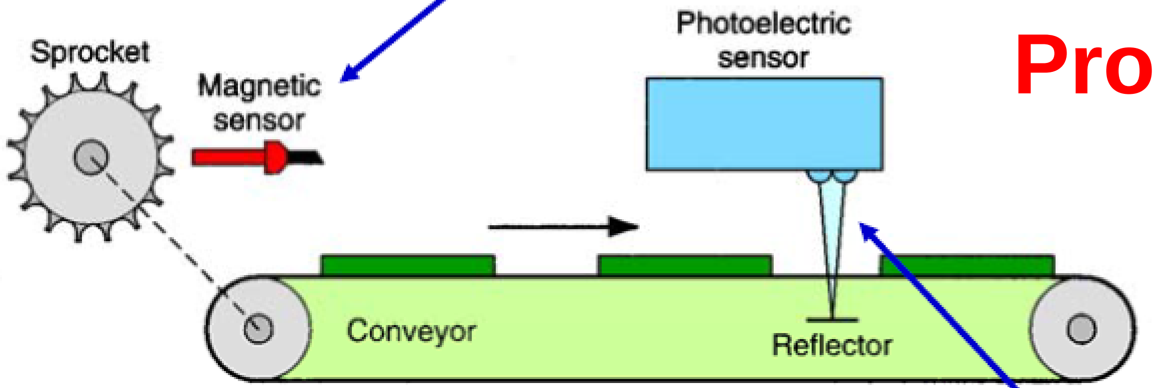


The object is advanced for a specific distance and measured by encoder pulses to determine the correct length for cutting.

Counter Used For Length Measurement

Radu Muresan

Count input pulses generated by the magnetic sensor, which detects passing teeth on a conveyor drive sprocket. If 10 teeth per foot of conveyor motion pass the sensor, the accumulated count of the counter would indicate feet in tenths.



Program?

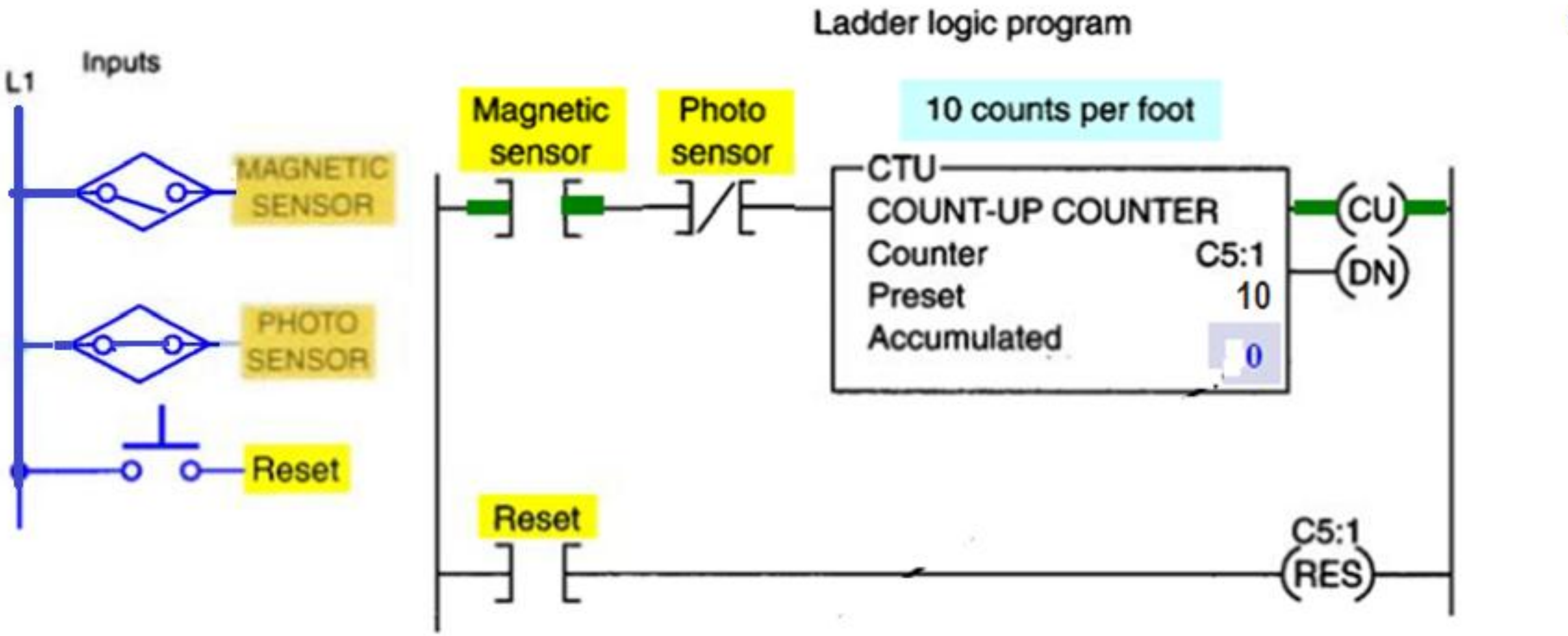
The photoelectric sensor monitors a reference point on the conveyor. When activated, it prevents the unit from counting, thus permitting the counter to accumulate counts only when bar stock is moving.

Application : could be cutting pieces of foot length long

Solution to be added later

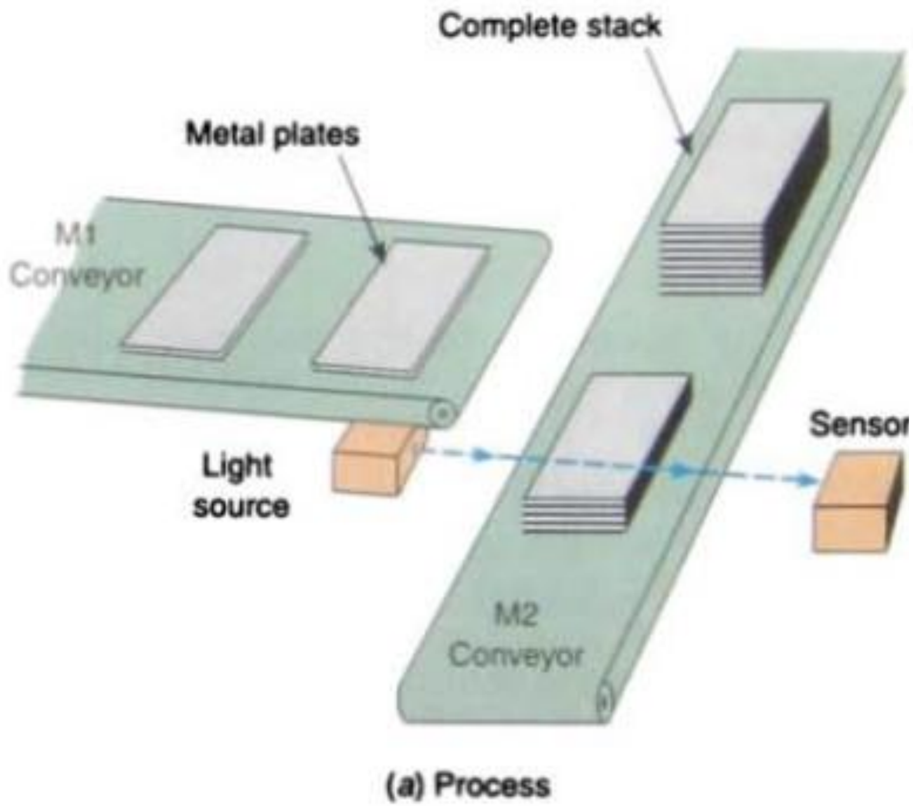
.

Counter Used For Length Measurement



Automatic Stacking program

Program?



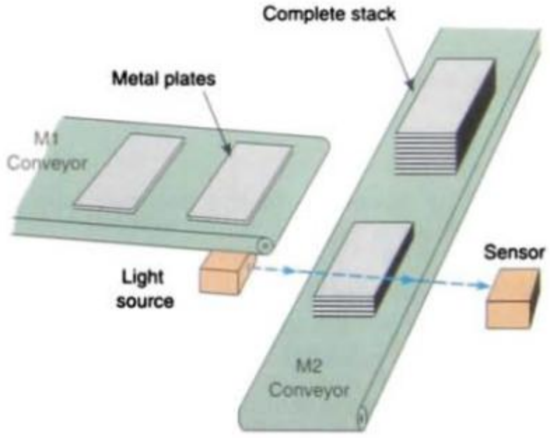
(a) Process

FIGURE 8-31 Automatic stacking program.

process, conveyor M1 is used to stack metal plates onto conveyor M2. The photoelectric sensor provides an input pulse to the PLC counter each time a metal plate drops from conveyor M1 to M2. When 15 plates have been stacked, conveyor M2 is activated for 5 s by the PLC timer. The operation of the program can be summarized as follows:

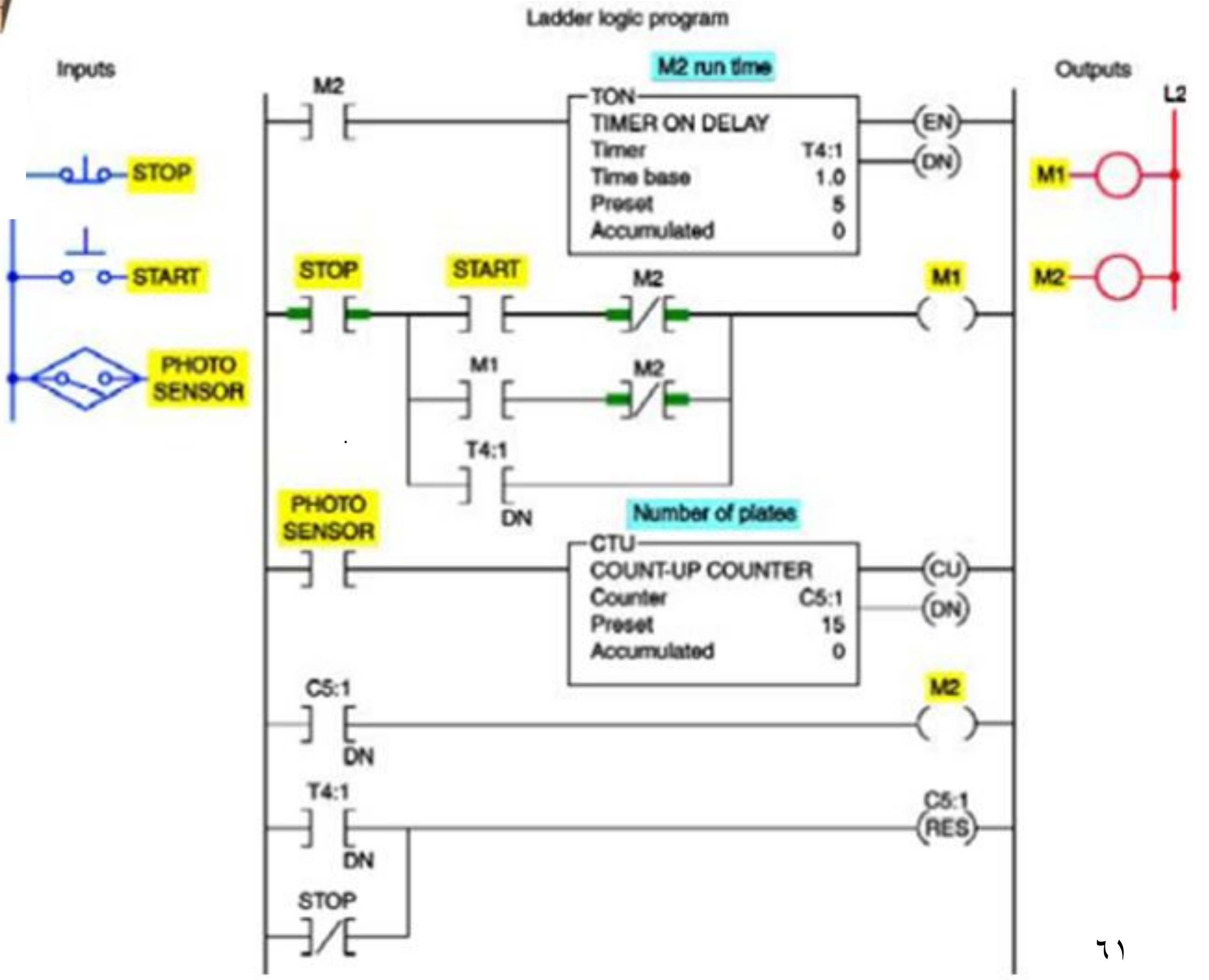
- When the start button is pressed, conveyor M1 begins running.
- After 15 plates have been stacked, conveyor M1 stops and conveyor M2 begins running.
- After conveyor M2 has been operated for 5 s, it stops and the sequence is repeated automatically.
- The done bit of the timer resets the timer and the counter and provides a momentary pulse to automatically restart conveyor M1.

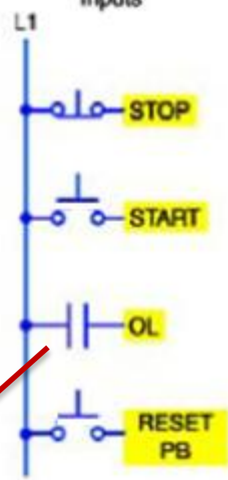
Solution to be added later

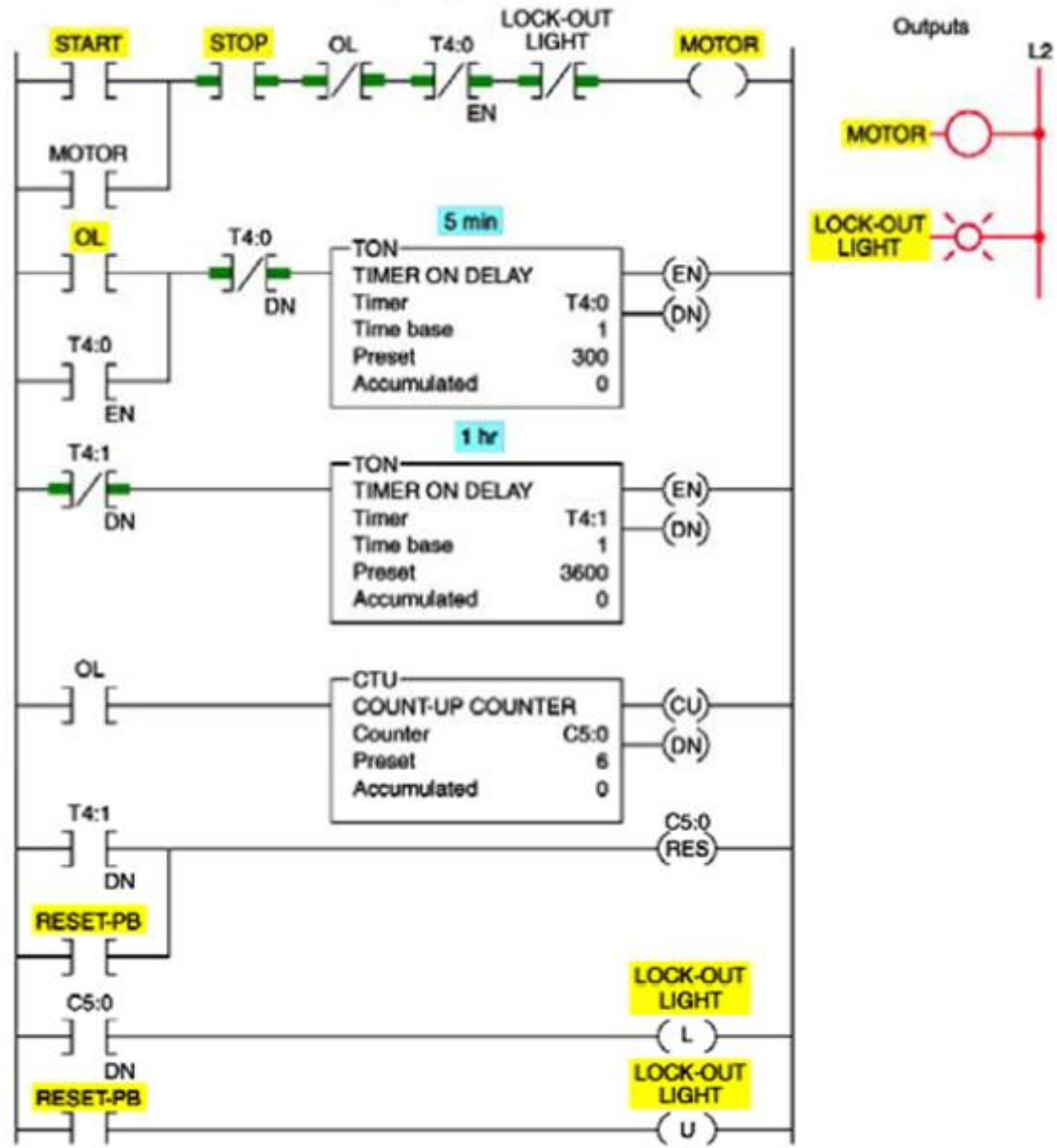


(*a*) Process

FIGURE 8-31 Automatic stacking program.





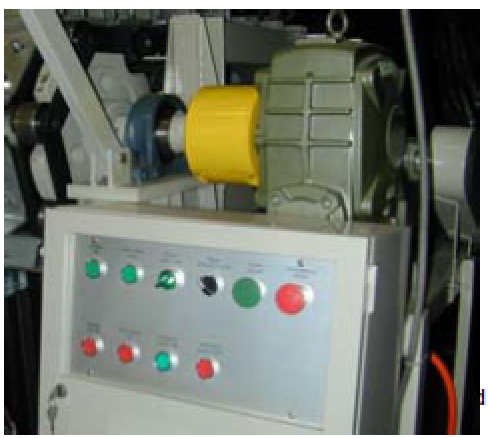


The normally open (OL) relay contact momentarily closes each time an overload current is sensed.

What does this program do?

Motor Lock-Out Program

Designed to prevent a machine operator from starting a motor that has tripped off more than 5 times in an hour.

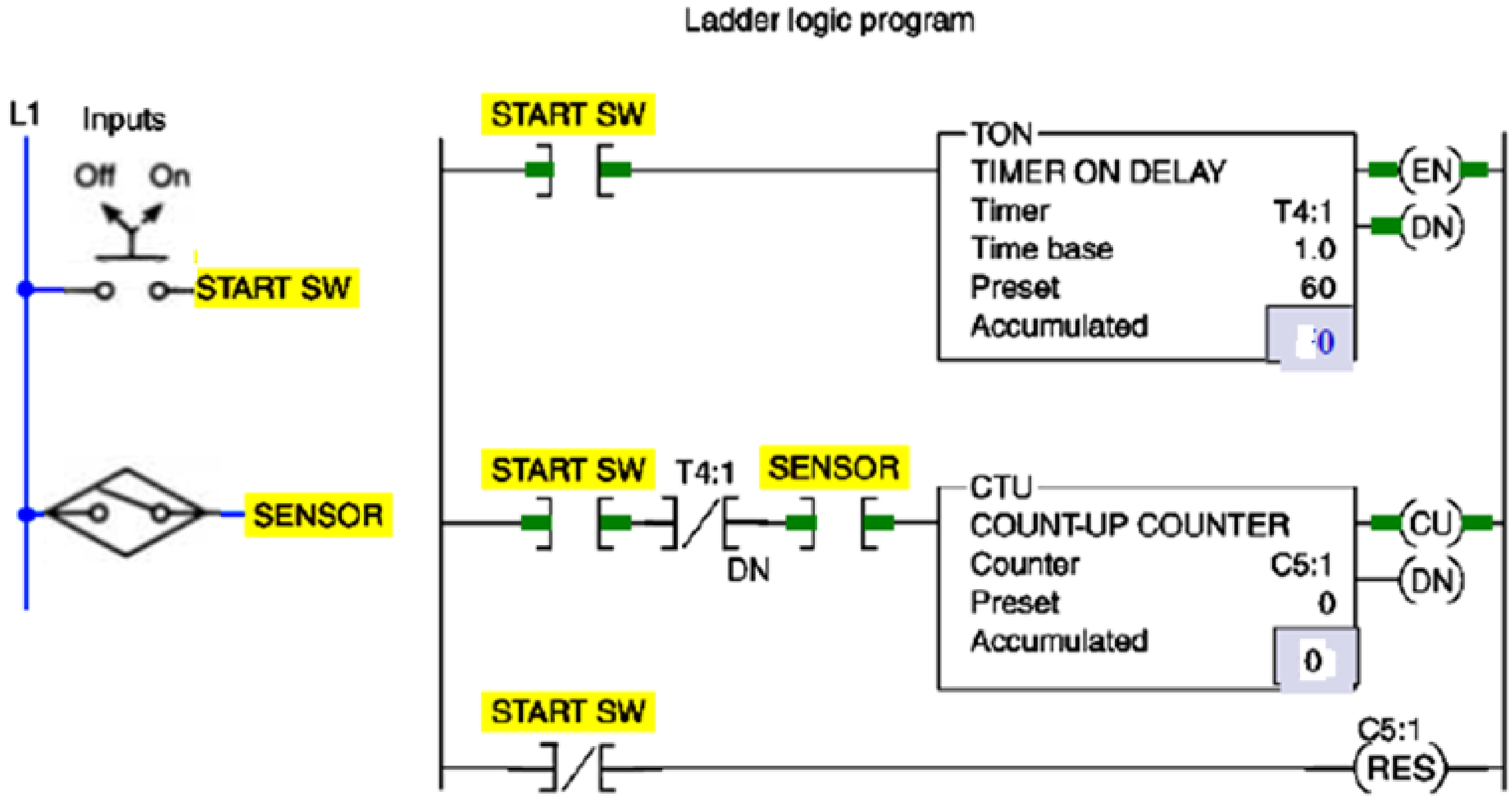


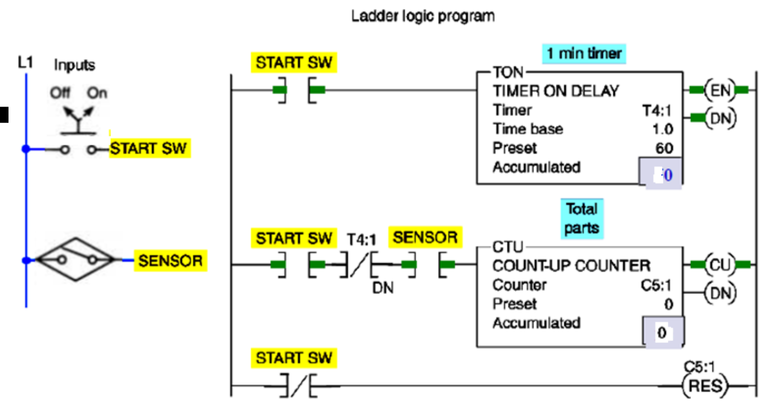
The normally open (OL) relay contact momentarily closes each time an overload current is sensed.

Every time the motor stops due to an overload condition, the motor start circuit is locked out for 5 min.

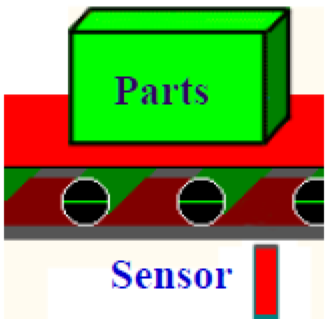
If the motor trips off more than 5 times in an hour, the motor start circuit is permanently locked out and cannot be started until the reset button is actuated.

What dose this program do?





Product Flow Rate Program



This program is designed to indicate how many parts per minute pass a given process point.

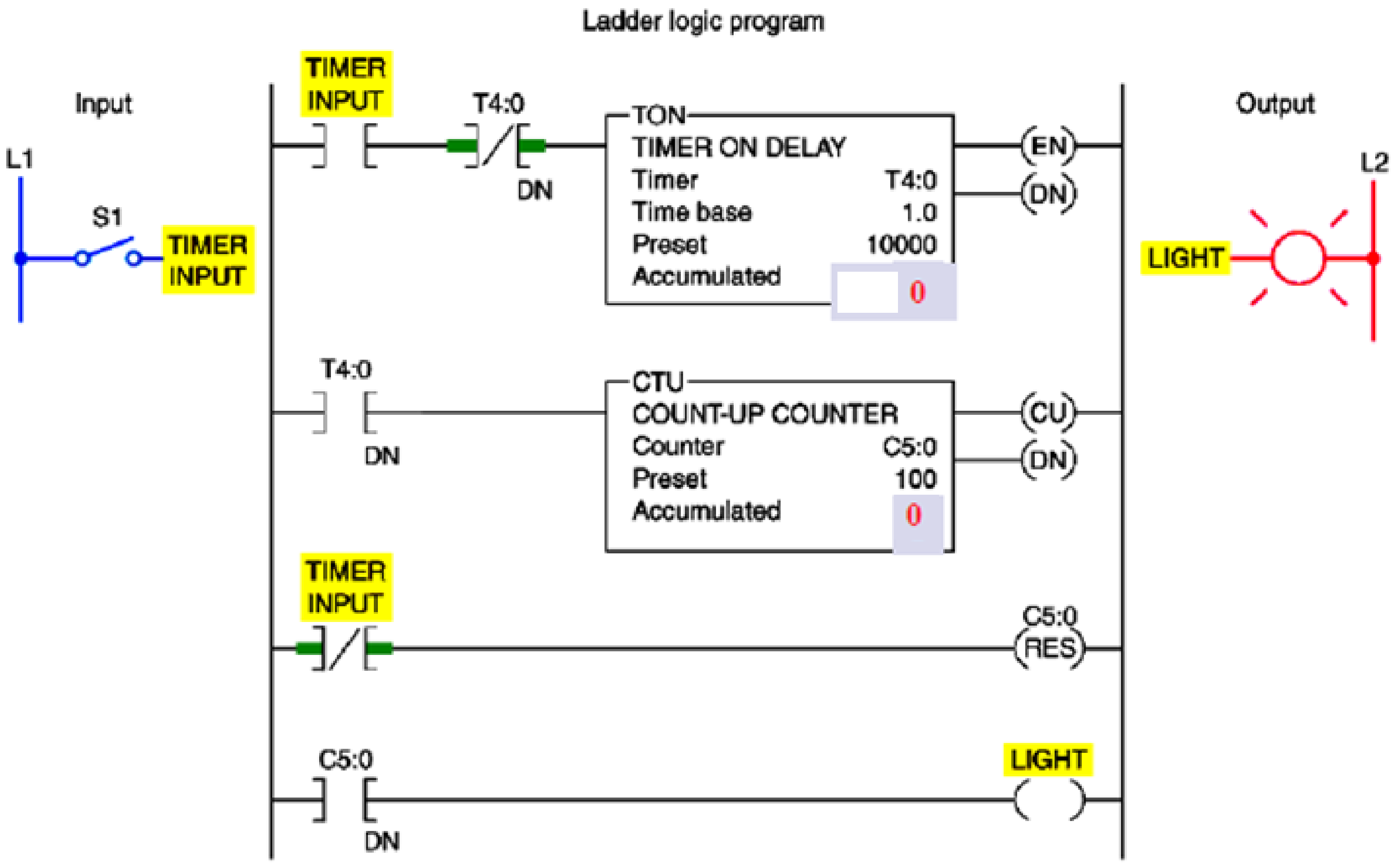
When the start switch is closed, both the counter and timer are enabled.

The counter is pulsed for each part passing the sensor.

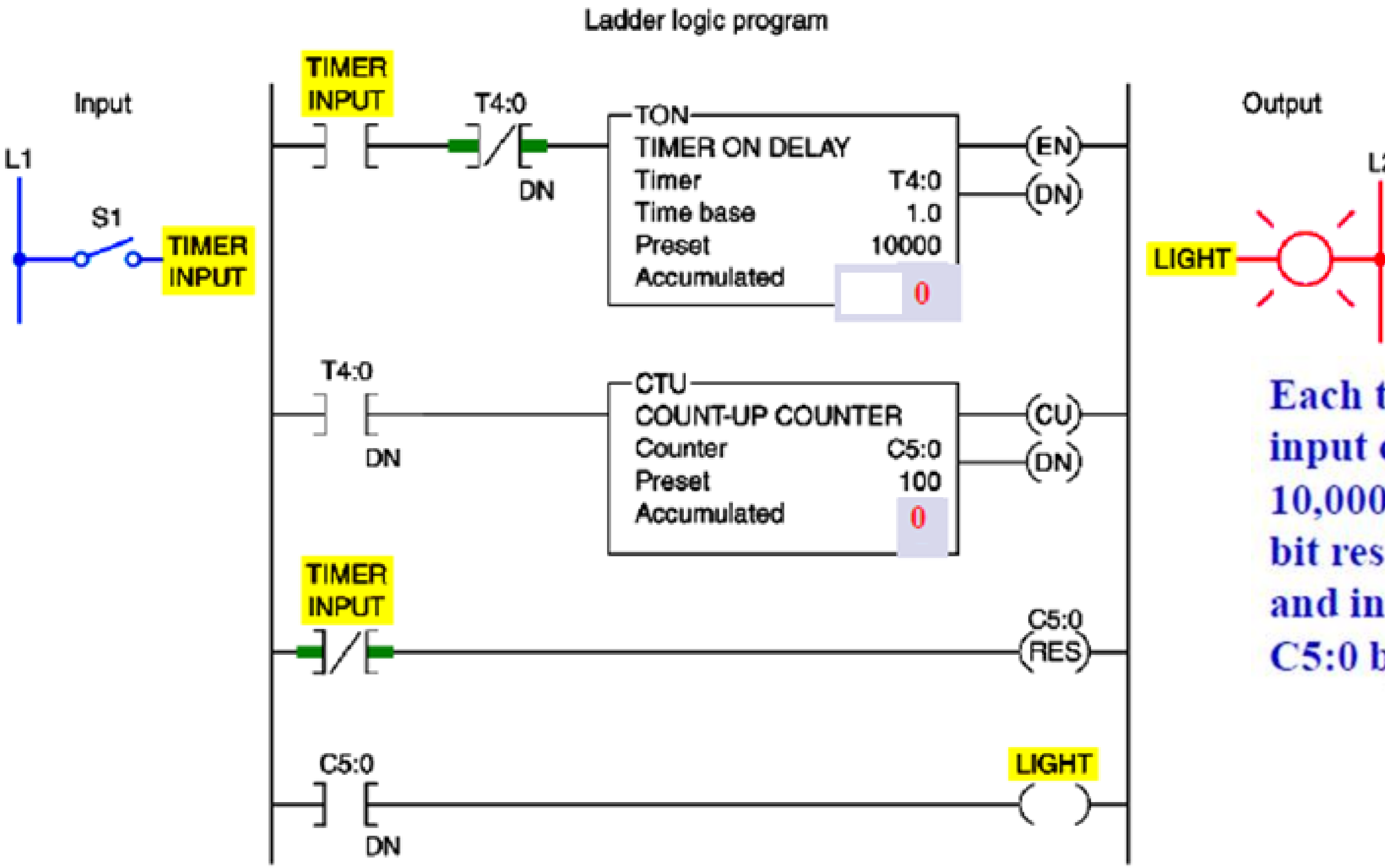
The counting begins and the timer starts timing through its 1-min time interval.

At the end of 1 min, the timer done bit causes the counter rung to go false. Sensor pulses continue but do not affect the PLC counter. The number of parts for the past minutes are represented by the accumulated value of the counter.

What dose this program do?



Timer Driving A Counter For Long Time-Delay Period



Each timer T4:0 input closes for 10,000 s, its done bit resets itself and increments C5:0 by 1.

The output light turns on 10,000 x 100, or 1,000,000 seconds after the timer input contact closes.