PLC & SCADA

Programmable Logic Controller (PLC):

1. PLC Hardware Environment

- Components
- Architecture
- Functionality

2. Software & Programming

- Ladder Programming
- Instruction list
- Latching and internal relays
- Sequencing
- Timers and counters
- Shift Register
- Master and jump control
- Data Handling
- Analog input & Output

3. Troubleshooting

- Field instrument troubleshooting
- Program Create/ Restore
- Signal control and Bypass

4. Industrial Control System (ICS)

- Supervisory Control and Data Acquisition (SCADA)
- Distributed Control System (DCS)

PLC & SCADA

Introduction:

A programmable logic controller (PLC) is a digital electronic device that uses a programmable memory to store instructions and to implement functions such as logic, sequencing, timing, counting and arithmetic to control machines and processes and has been specifically designed to make programming easy.

The term **logic** is used because the programming is primarily concerned with implementing logic and switching operations. Input devices, e.g., switches, and output devices, e.g. motors, being controlled are connected to the PLC and then the controller monitors the inputs and outputs according to the program stored in the PLC by the operator and so controls the machine or process.

Originally PLCs were designed as a replacement for **hard-wired relay** and **timer logic control systems**.

PLCs have the great advantage that it is possible to modify a control system without having to rewire the connections to the input and output devices, the only requirement being that an operator must key in a different set of instructions. Also, they are much faster than relay-operated systems. The result is a flexible system which can be used to control systems which vary quite widely in their nature and complexity. Such systems are widely used for the implementation of logic control functions because they are easy to use and program. PLCs are like computers but have certain features which are specific to their use as controllers. These are:

- They are rugged and designed to withstand vibrations, temperature,

  humidity and noise.
- The interfacing for inputs and outputs is inside the controller
- They are easily programmed.

PLC & SCADA

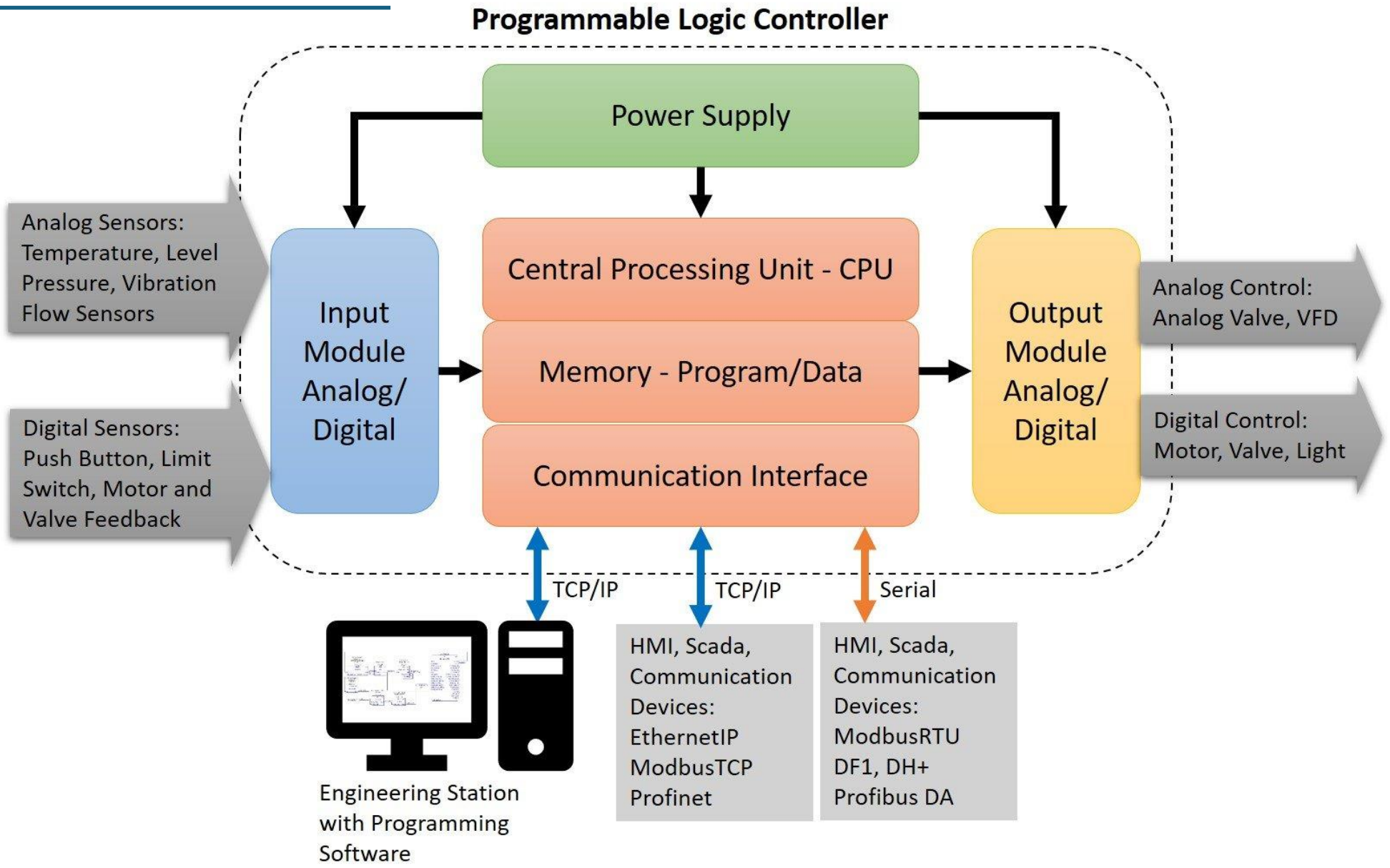
Introduction:

Nature of Industrial Automation Computation:

- Interfaced with physical signals
- Slow to fast processing time
- Complexity Level (Low to medium)
- Repetitive and Real time task.
  - Mostly static scheduling
  - Simple OS
- Critical in Nature
- Harsh physical environments (Dust, heat, moisture & Vibrations)

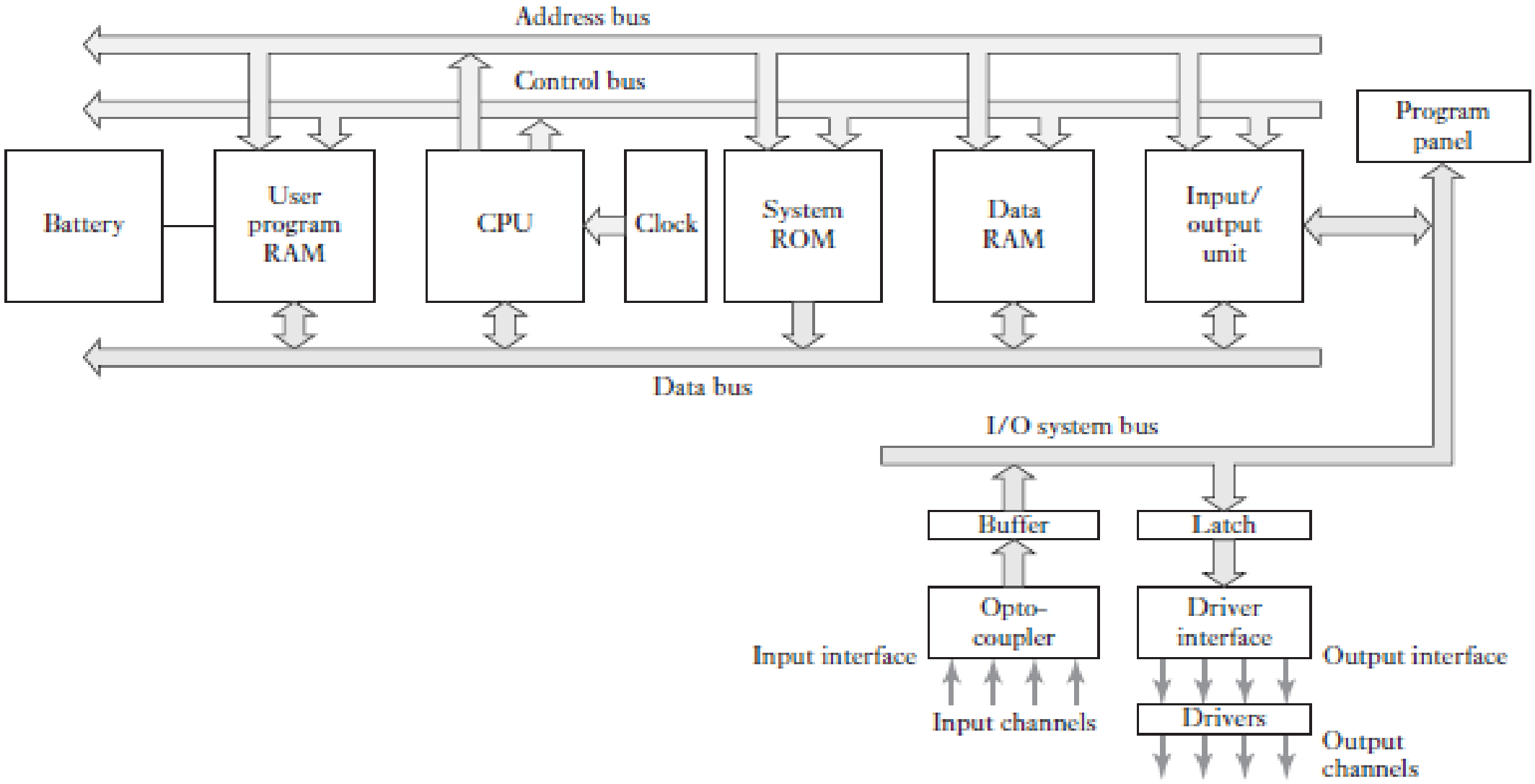
PLC & SCADA

Internal Architecture of PLC



PLC & SCADA

Hardware Architecture of PLC

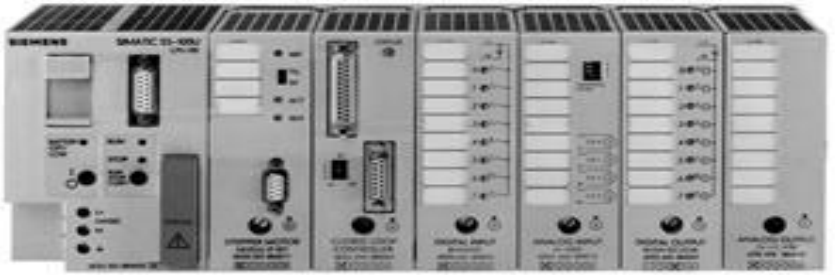


PLC & SCADA

Common versions of Industrial PLC

S7-400

S5-100U



S7-1500

S7-300

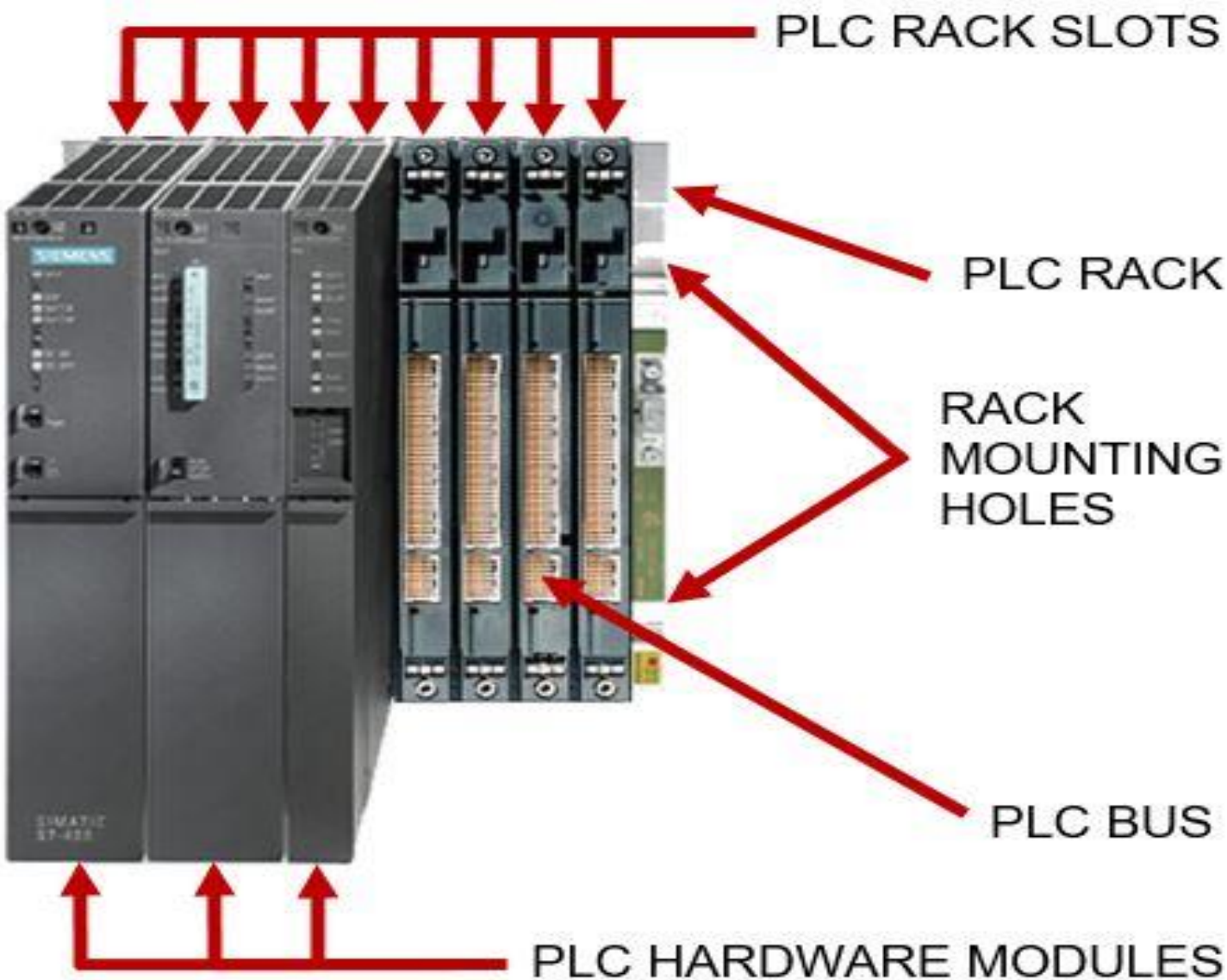
S7-1200

S7 Safety PLC

"Logo"

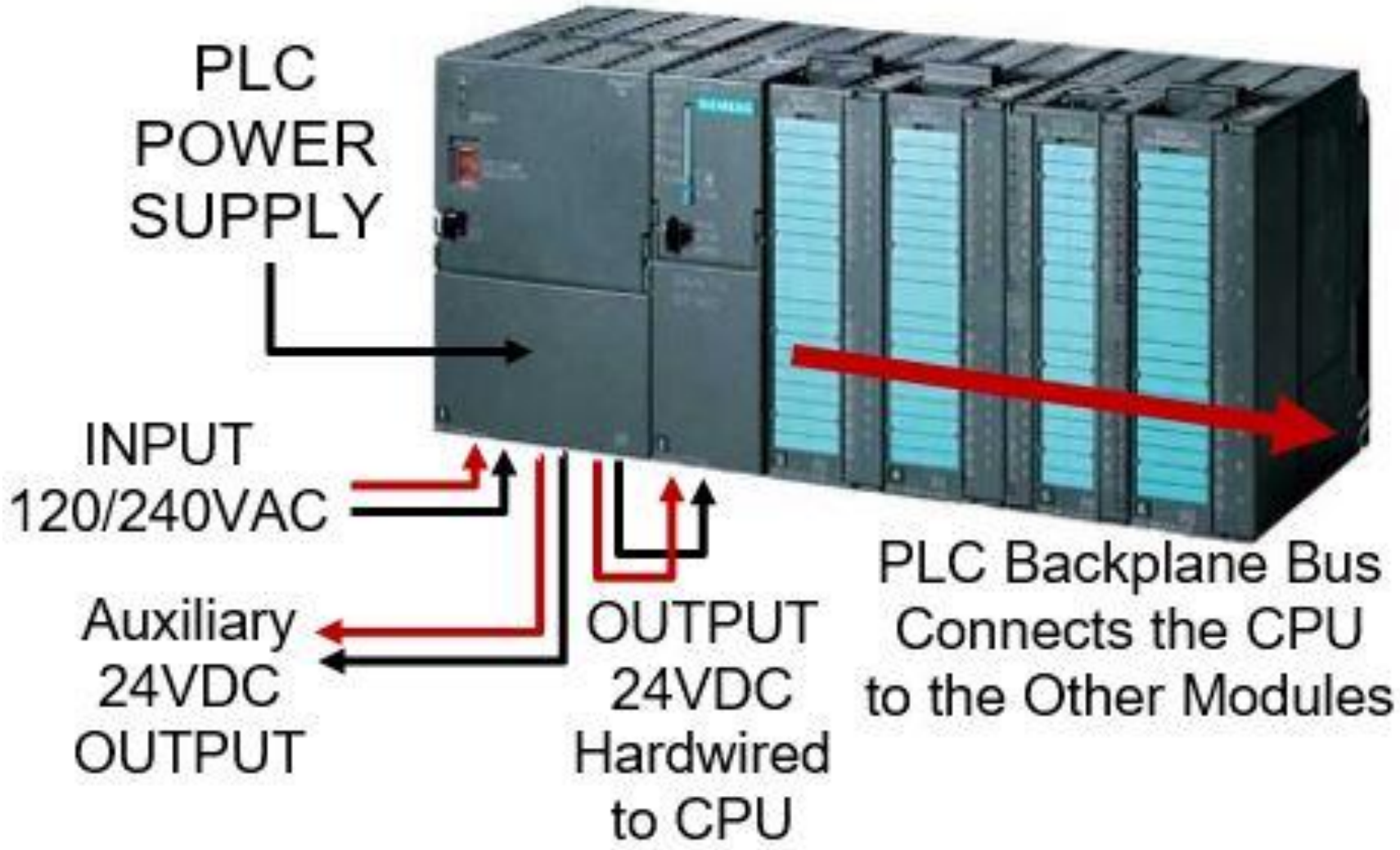
PLC & SCADA

Internal Setup of PLC



PLC & SCADA

PLC Configuration :

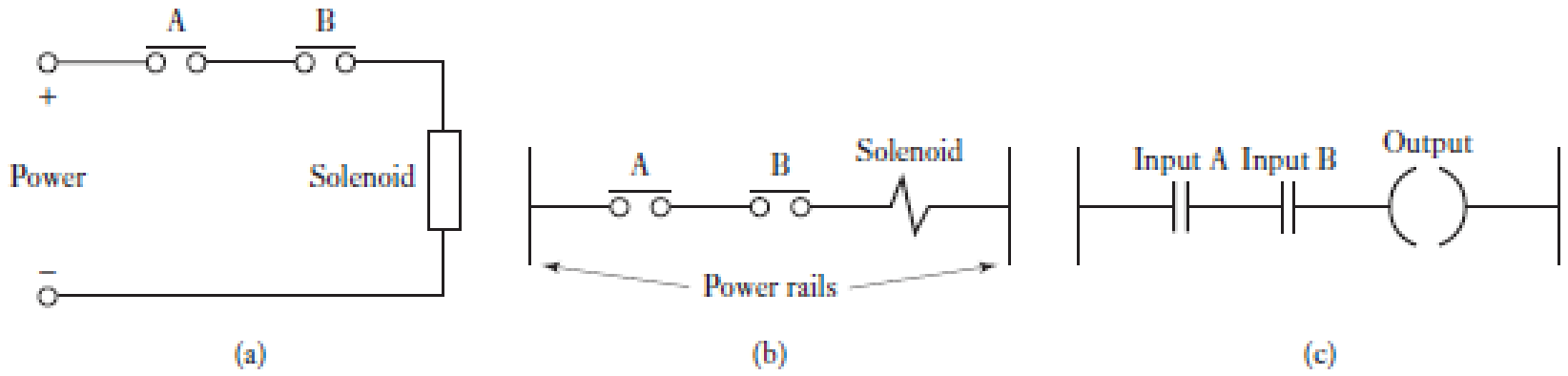


PLC & SCADA

Software & Programming

1. Ladder programming:

The form of programming commonly used with PLCs is ladder programming. This involves each program task being specified as though a rung of a ladder. Thus, such a rung could specify that the state of switches A and B, the inputs, be examined and if A and B are both closed then a solenoid, the output, is energized. Below figure illustrates this idea by comparing it with an electric circuit.



(a), (b) Alternative ways of drawing an electric circuit, (c) comparable rung

PLC & SCADA

Software & Programming

1. Ladder programming :

The sequence followed by a PLC when carrying out a program can be summarized as follows.

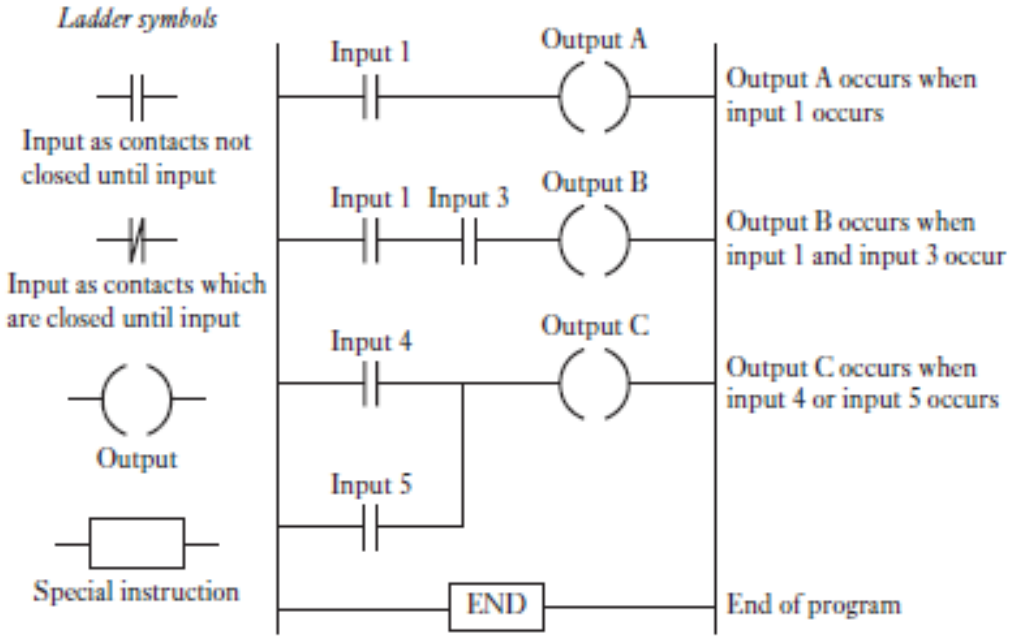
1. Scan the inputs associated with one rung of the ladder program.
2. Solve the logic operation involving those inputs.
3. Set/reset the outputs for that rung.
4. Move on to the next rung and repeat operations 1, 2, 3.
5. Move on to the next rung and repeat operations 1, 2, 3.
6. Move on to the next rung and repeat operations 1, 2, 3.
7. And so on until the end of the program with each rung of the ladder program scanned in turn. The PLC then goes back to the beginning of the program and starts again.

PLC & SCADA

Software & Programming

1. Ladder programming :

PLC programming based on the use of ladder diagrams involves writing a program in a similar manner to drawing a switching circuit. The ladder diagram consists of two vertical lines representing the power rails. Circuits are connected as horizontal lines, i.e., the rungs of the ladder, between these two verticals. Below figure shows the basic standard symbols that are used and an example of rungs in a ladder diagram. In drawing the circuit line for a rung, inputs must always precede outputs and there must be at least one output on each line. Each rung must start with an input or a series of inputs and end with an output.

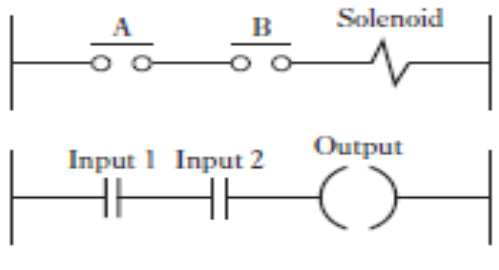


PLC & SCADA

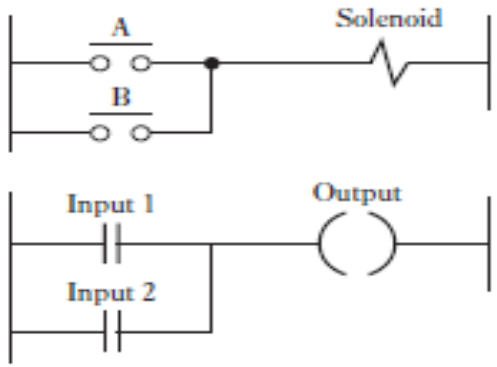
Software & Programming

2. Logic Functions :

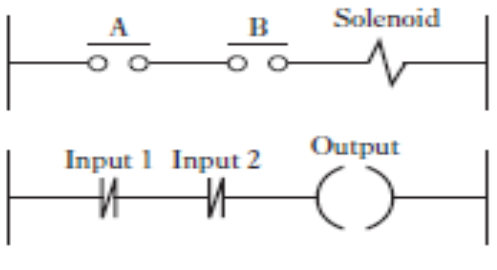
The logic functions can be obtained by combinations of switches and the following shows how we can write ladder programs for such combinations .



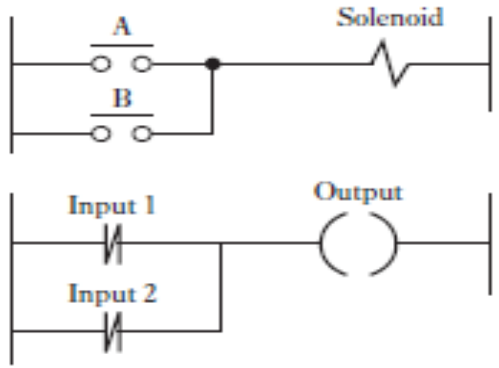
(a)



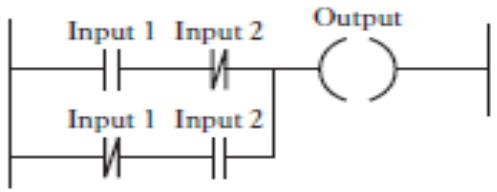
(b)



(c)



(d)



(e)

PLC & SCADA

Software & Programming

2. Logic Functions :

Question.

Consider a situation where a normally open switch A must be activated and either of two other, normally open, switches B and C must be activated for a coil to be energized. We can represent this arrangement of switches as switch A in series with two parallel switches B and C . For the coil to be energized we require A to be closed and either B or C to be closed. Switch A when considered with the parallel switches gives an AND logic situation. The two parallel switches give an OR logic situation. We thus have a combination of two gates ?

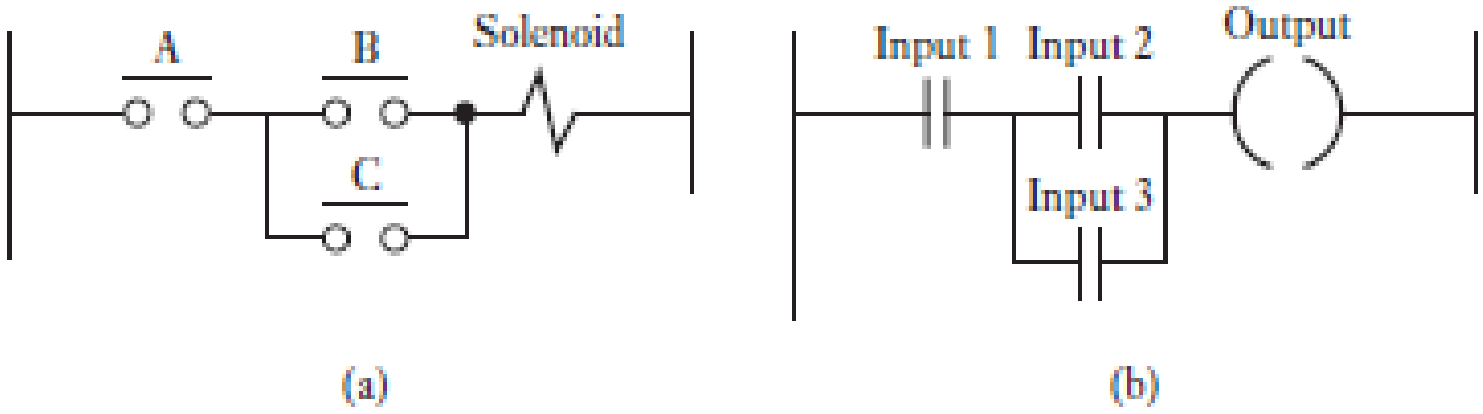
PLC & SCADA

Software & Programming

2. Logic Functions :

Consider a situation where a normally open switch A must be activated and either of two other, normally open, switches B and C must be activated for a coil to be energized. We can represent this arrangement of switches as switch A in series with two parallel switches B and C (Figure 14.9(a)). For the coil to be energized we require A to be closed and either B or C to be closed. Switch A when considered with the parallel switches gives an AND logic situation. The two parallel switches give an OR logic situation. We thus have a combination of two gates.

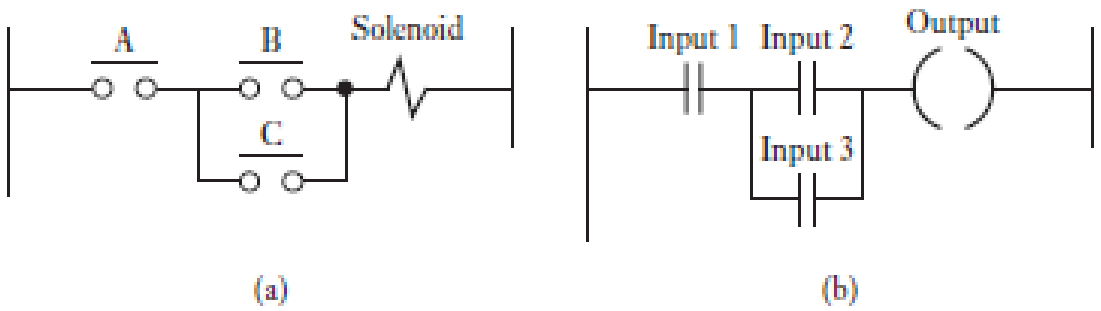
Write a truth table for below logic?



PLC & SCADA

Software & Programming

2. Logic Functions :



(a) (b)

Inputs			
A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

PLC & SCADA

Software & Programming

3. Instruction Lists :

Each horizontal rung on the ladder in a ladder program represents a line in the program and the entire ladder gives the complete program in ‘ladder language’.

The programmer can enter the program into the PLC using a keyboard with the graphic symbols for the ladder elements or using a computer screen and a mouse to select symbols, and the program panel or computer then translates these symbols into machine language that can be stored in the PLC memory.

There is an alternative way of entering a program and that is to translate the ladder program into an instruction list and then enter this into the programming panel or computer.

Instruction lists consist of a series of instructions with each instruction being on a separate line. An instruction consists of an operator followed by one or more operands, i.e. the subjects of the operator. In terms of ladder programs, each operator in a program may be regarded as a ladder element. Thus, we might have for the equivalent of an input to a ladder program:

PLC & SCADA

Software & Programming

3. Instruction Lists :

Instruction code mnemonics

IEC 1131-3	Mitsubishi	OMRON	Siemens	Operation	Ladder diagram
LD	LD	LD	A	Load operand into result register	Start a rung with open contacts
LDN	LDI	LD NOT	AN	Load negative operand into result register	Start a rung with closed contacts
AND	AND	AND	A	Boolean AND	A series element with open contacts
ANDN	ANI	AND NOT	AN	Boolean AND with negative operand	A series element with closed contacts
OR	OR	OR	O	Boolean OR	A parallel element with open contacts
ORN	ORI	OR NOT	ON	Boolean OR with negative operand	A parallel element with closed contacts
ST	OUT	OUT	=	Store result register into operand	An output from a rung

PLC & SCADA

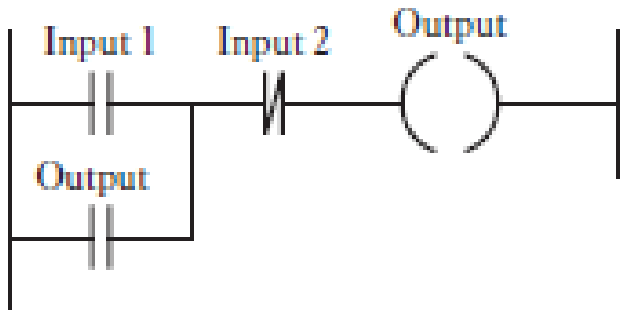
Software & Programming

4. Latching and Internal relays :

There are often situations where it is necessary to hold a coil energised, even when the input which energised it ceases. The term latch circuit is used for the circuit which carries out such an operation.

It is a self-maintaining circuit in that, after being energised, it maintains that state until another input is received. It remembers its last state.

An example of a latch circuit is shown in figure below. When Input 1 is energised and closes, there is an output. However, when there is an output, a set of contacts associated with the output is energised and closes. These contacts OR the Input 1 contacts. Thus, even if Input 1 contacts open, the circuit will still maintain the output energised. The only way to release the output is by operating the normally closed contact Input 2.



PLC & SCADA

Software & Programming

4. Latching and Internal relays :

The term internal relay, auxiliary relay or marker is used for what can be considered as an internal relay in the PLC.

These behave like relays with their associated contacts, but are not actual relays but simulations by the software of the PLC.

Some have battery back-up so that they can be used in circuits to ensure a safe shut-down of plant in the event of a power failure.

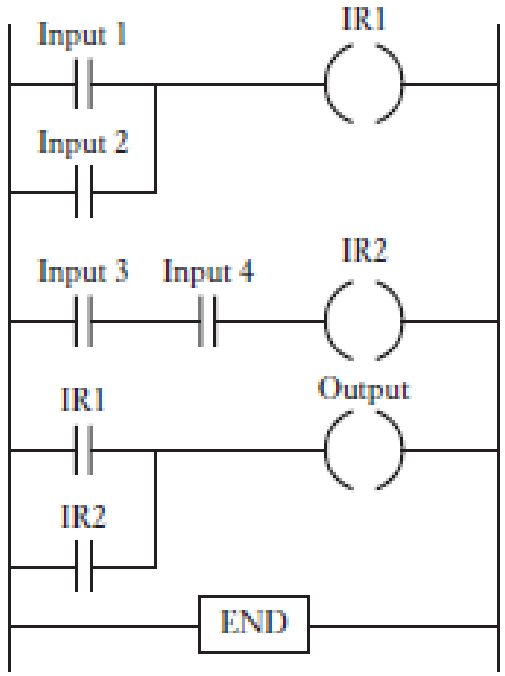
Internal relays are often used when there are programs with multiple input conditions.

PLC & SCADA

Software & Programming

4. Latching and Internal relays :

Consider the situation where the excitation of an output depends on two different input arrangements. Below figure shows how we can draw a ladder diagram using internal relays.



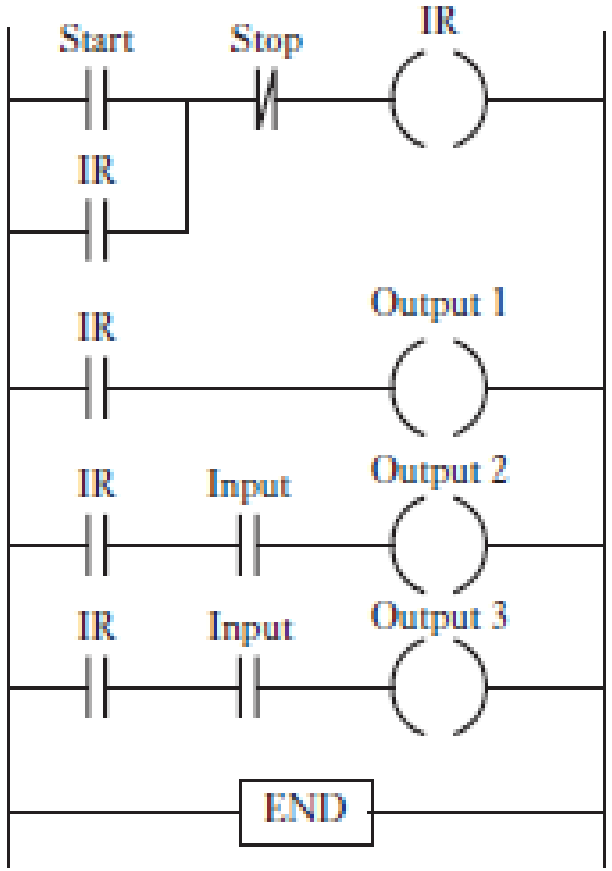
The first rung shows one input arrangement being used to control the coil of internal relay IR1. The second rung shows the other input arrangement controlling the coil of internal relay IR2. The contacts of the two relays are then put in an OR situation to control the output. Another use of internal relays is for the starting of multiple outputs.

PLC & SCADA

Software & Programming

4. Latching and Internal relays :

Below figure shows such a ladder program. When the start contacts are closed, the internal relay is activated and latches the input. It also starts Output 1 and makes it possible for Outputs 2 and 3 to be activated.



PLC & SCADA

Software & Programming

5. Sequencing :

There are often control situations where sequences of outputs are required, with the switch from one output to another being controlled by sensors.

Consider the requirement for a ladder program for a pneumatic system with double-solenoid valves controlling two double-acting cylinders A and B if limit switches a-, a+, b-, b+ are used to detect the limits of the piston rod movements in the cylinders and the cylinder activation sequence A+, B+, A- B- is required. A possible program is shown in the figure.

A start switch input has been included in the first rung. Thus, cylinder extension for A, i.e. the solenoid A+ energised, only occurs when the start switch is closed and the b-switch is closed, this switch indicating that the B cylinder is retracted. When cylinder A is extended, the switch a+, which indicates the extension of A, is activated. This then leads to an output to solenoid B+ which results in B extending. This closes the switch indicating the extension of B, i.e. the b+ switch, and leads to the output to solenoid A- and the retraction of cylinder A.

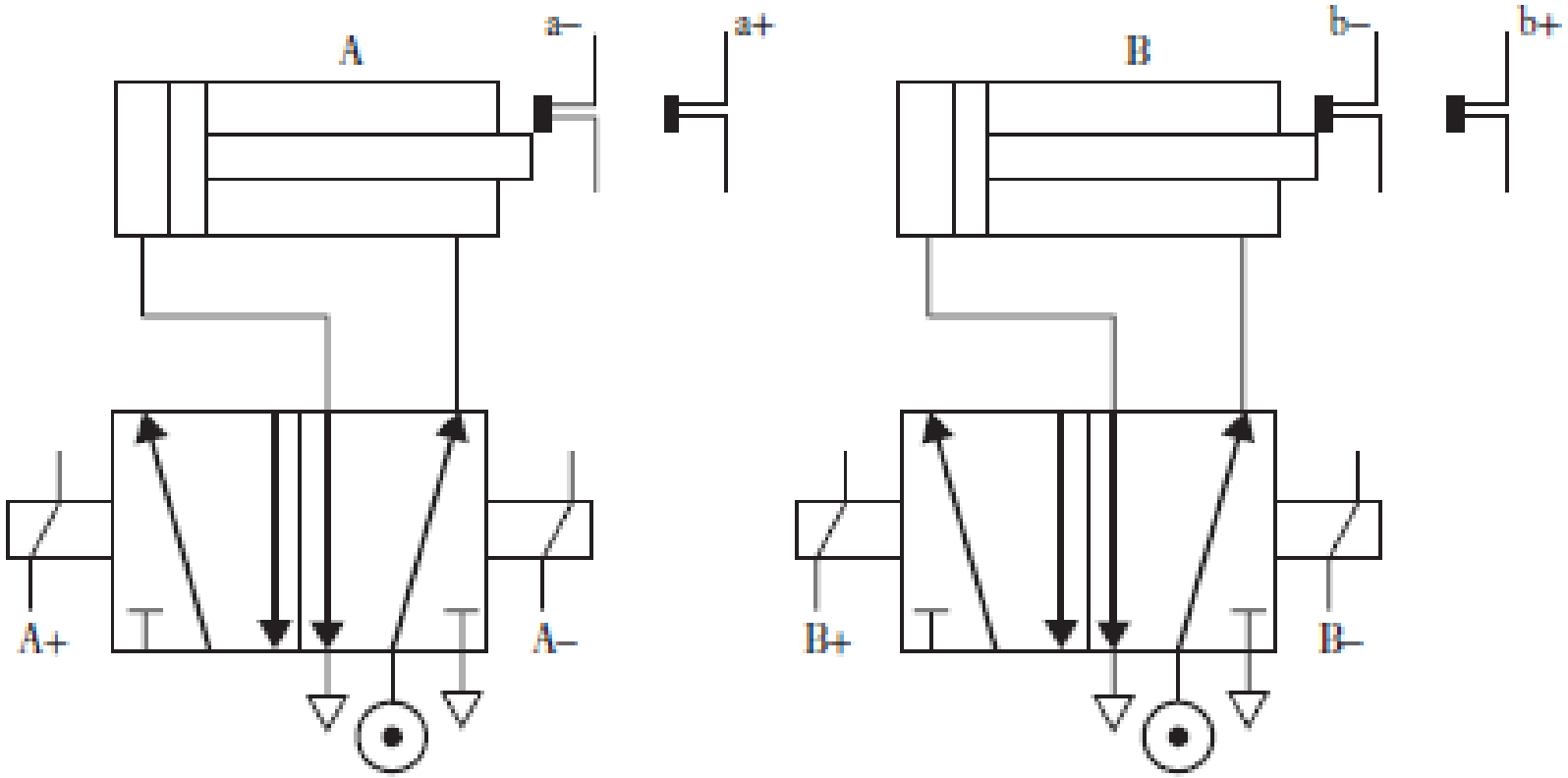
This retraction closes limit switch a- and so gives the output to solenoid B- which results in B retracting. This concludes the program cycle and leads to the first rung again, which awaits the closure of the start switch before being repeated.

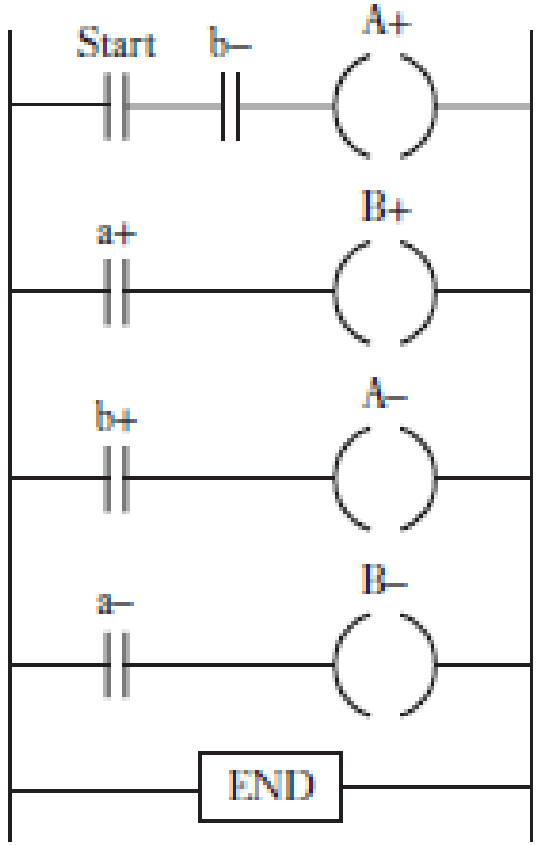
PLC & SCADA

Software & Programming

5. Sequencing :

Pneumatic Cylinder sequencing :



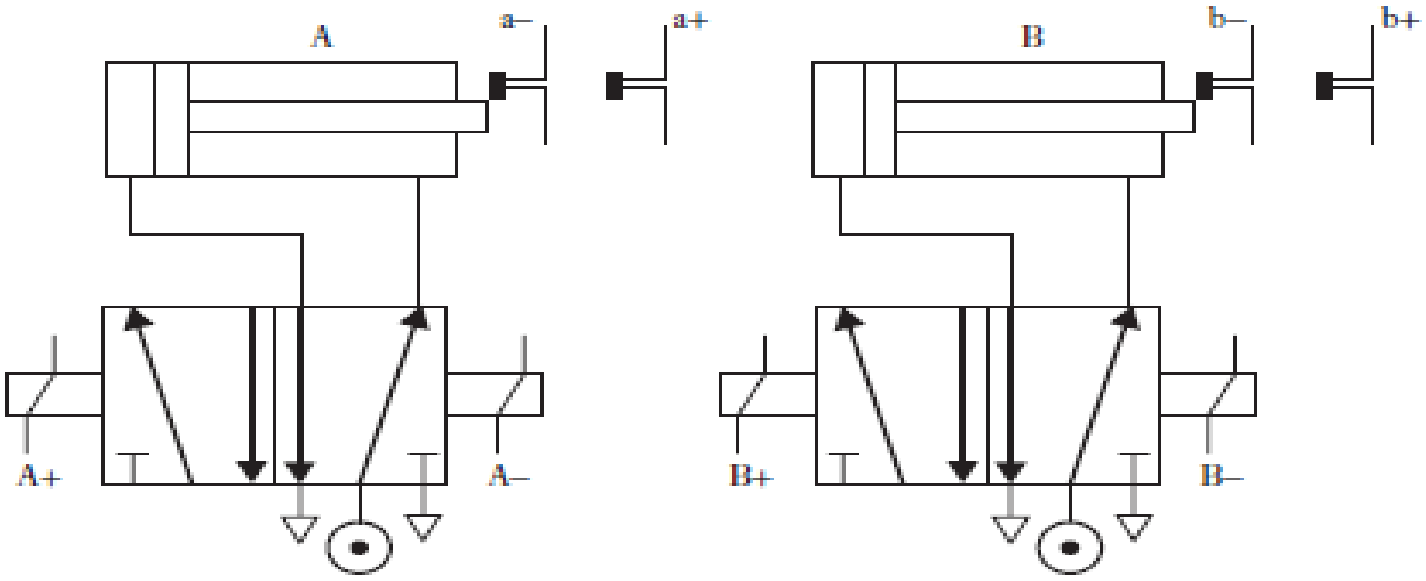


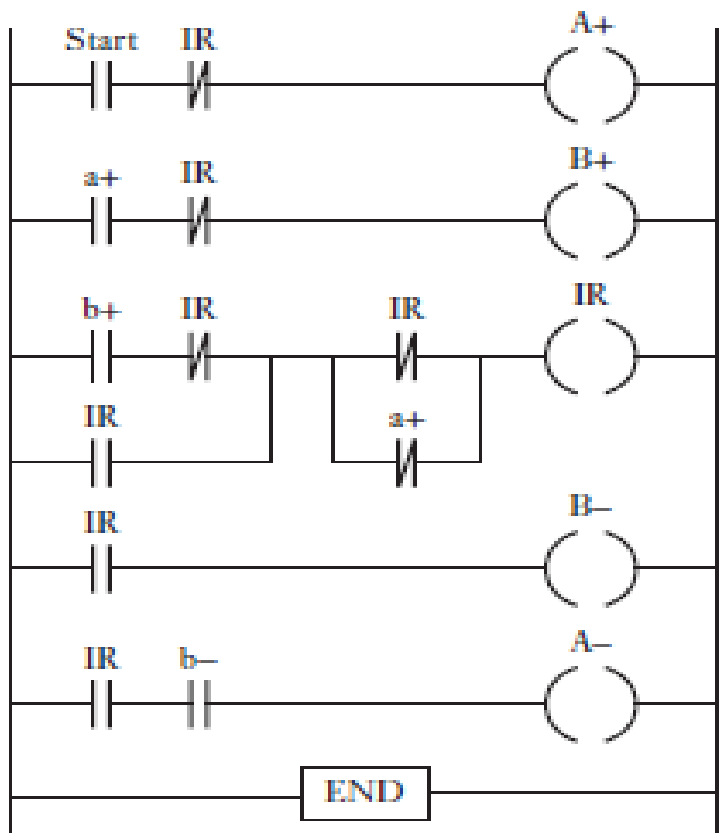
PLC & SCADA

Software & Programming

5. Sequencing :

Pneumatic Cylinder sequencing :



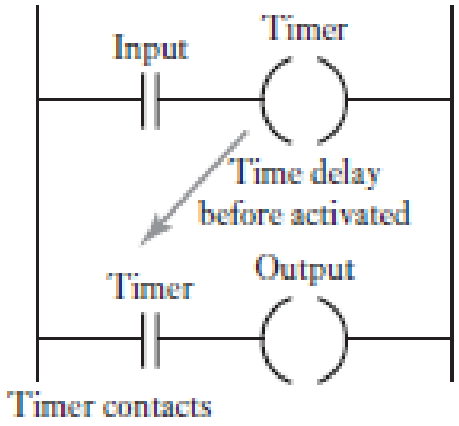


PLC & SCADA

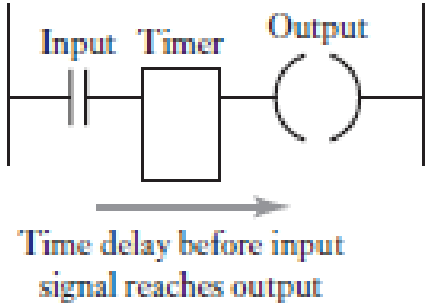
Software & Programming

6. Timers and Counters:

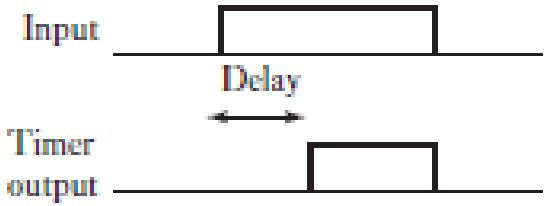
A common approach used by PLC manufacturers is to consider timers to behave like relays with coils which when energised result in the closure or opening of contacts after some **preset time.** The timer is thus treated as an output for a rung with control being exercised over pairs of contacts elsewhere. Others consider a timer as a delay block in a rung which delays signals in that rung reaching the output.



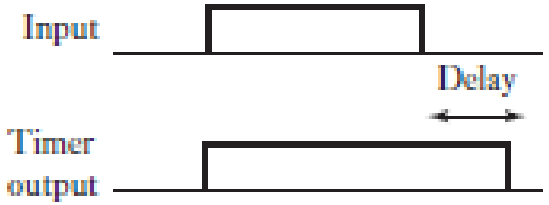
(a)



(b)



(c) On-delay timer TON



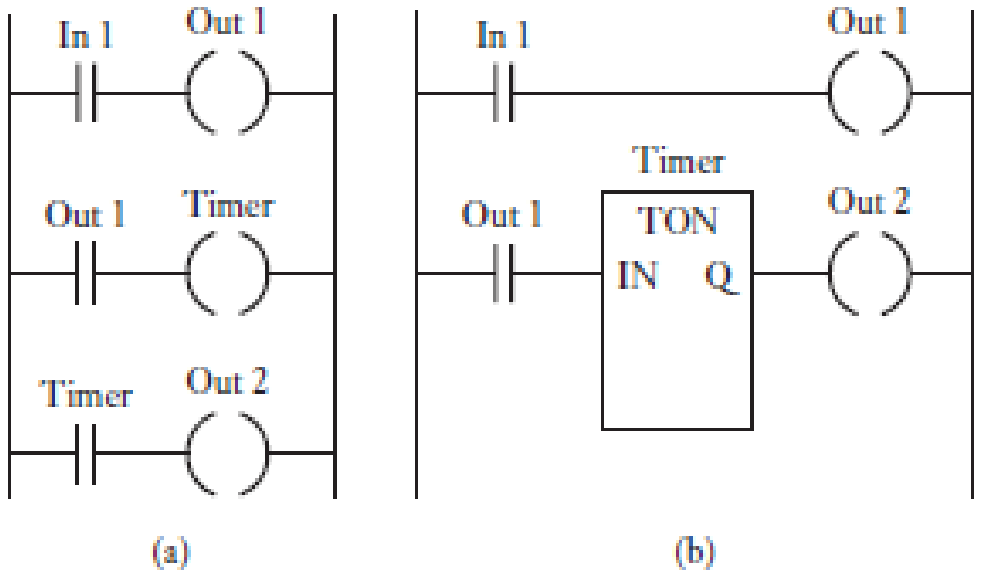
(d) Off-delay time TOFF

PLC & SCADA

Software & Programming

6. Timers and Counters:

PLCs are generally provided with only a delay-on timer (TON), small PLCs possibly having only this type. Such a timer waits for a fixed delay period before turning on, e.g., a period which can be set between 0.1 and 999 s in steps of 0.1 s. Other time-delay ranges and steps are possible.



PLC & SCADA

Software & Programming

6. Timers and Counters:

Counters are used when there is a need to count a specified number of contact operations, e.g., where items pass along a conveyor into boxes, and when the specified number of items has passed into a box, the next item is diverted into another box.

Counter circuits are supplied as an internal feature of PLCs. In most cases the counter operates as a down-counter.

This means that the counter counts down from the present value to zero, i.e., events are subtracted from the set value.

When zero is reached the counter's contact changes state. An up-counter would count to the preset value, i.e., events are added until the number reaches the set value. When the set value is reached the counter's contact changes state.

PLC & SCADA

Software & Programming

6. Timers and Counters:

Different PLC manufacturers deal with counters in different ways.

Some consider the counter to consist of two basic elements:

One output coil to count input pulses and one to reset the counter, the associated contacts of the counter being used in other rungs, e.g. Mitsubishi and Allen-Bradley.

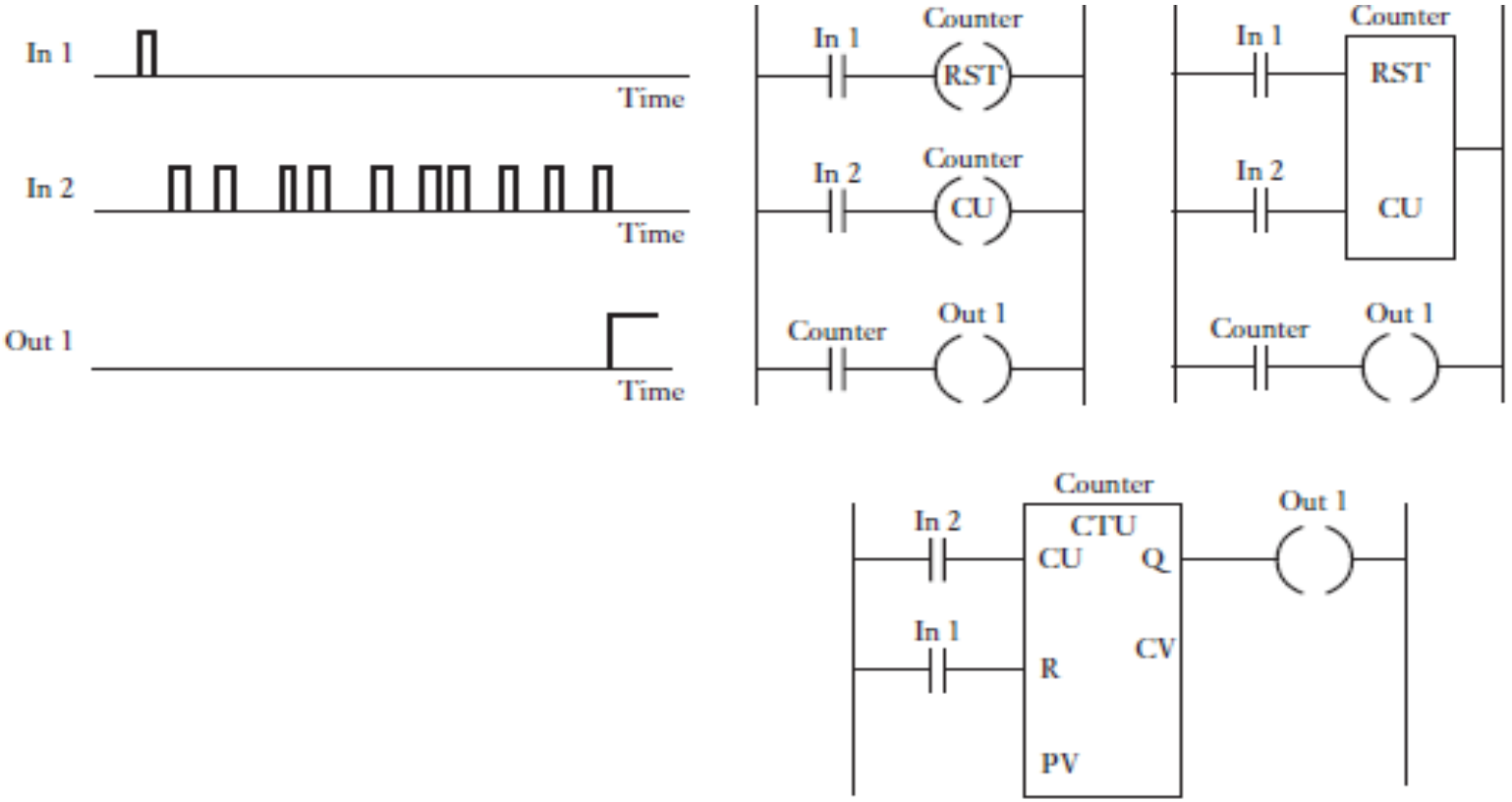
Others treat the counter as an intermediate block in a rung from which signals emanate when the count is attained, e.g. Siemens.

PLC & SCADA

Software & Programming

6. Timers and Counters:

As an illustration, Below figure shows a basic counting circuit. When there is a pulse input to In 1, the counter is reset. When there is an input to In 2, the counter starts counting. If the counter is set for, say, 10 pulses, then when 10 pulse inputs have been received at In 2, the counter's contacts will close and there will be an output from Out 1. If at any time during the counting there is an input to In 1, the counter will be reset and start all over again and count for 10 pulses.



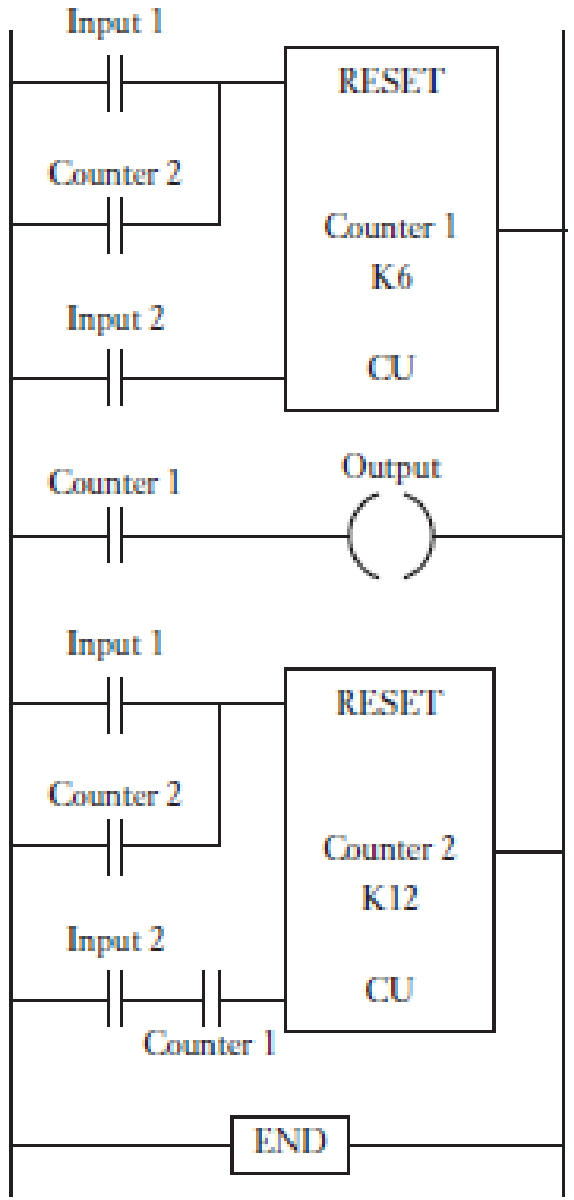
PLC & SCADA

Software & Programming

6. Timers and Counters:

As an illustration of the use of a counter, consider the problem of the control for a machine which is required to direct 6 items along one path for packaging in a box, and then 12 items along another path for packaging in another box.

Figure shows the program that could be used. It involves two counters, one preset to count 6 and the other to count 12. Input 1 momentarily closes its contacts to start the counting cycle, resetting both counters. Input 2 contacts could be activated by a microswitch, which is activated every time an item passes up to the junction in the paths. Counter 1counts 6 items and then closes its contact. This activates the output, which might be a solenoid used to activate a flap which closes one path and opens another. Counter 1 also has contacts which close and enables Counter 2 to start counting. When Counter 2 has counted 12 items it resets both the counters and opens the Counter 1 contacts, which then results in the output becoming deactivated and items no longer directed towards the box to contain 12 items.



PLC & SCADA

Software & Programming

7. Shift Register

A number of internal relays can be grouped together to form a register which can provide a storage area for a series sequence of individual bits.

A 4-bit register would be formed by using four internal relays, an 8-bit using eight.

The term shift register is used because the bits can be shifted along by 1 bit when there is a suitable input to the register.

For example, with an 8-bit register we might initially have:

1	0	1	1	0	1	0	1

Then there is an input of a 0 shift pulse:

0 → | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | → 1

with the result that all the bits shift along one place and the last bit overflows.

PLC & SCADA

Software & Programming

7. Shift Register

Shift registers have three inputs: one to load data into the first element of the register (OUT), one as the shift command (SFT) and one for resetting (RST).

With OUT, a logic level 0 or 1 is loaded into the first element of the shift register.

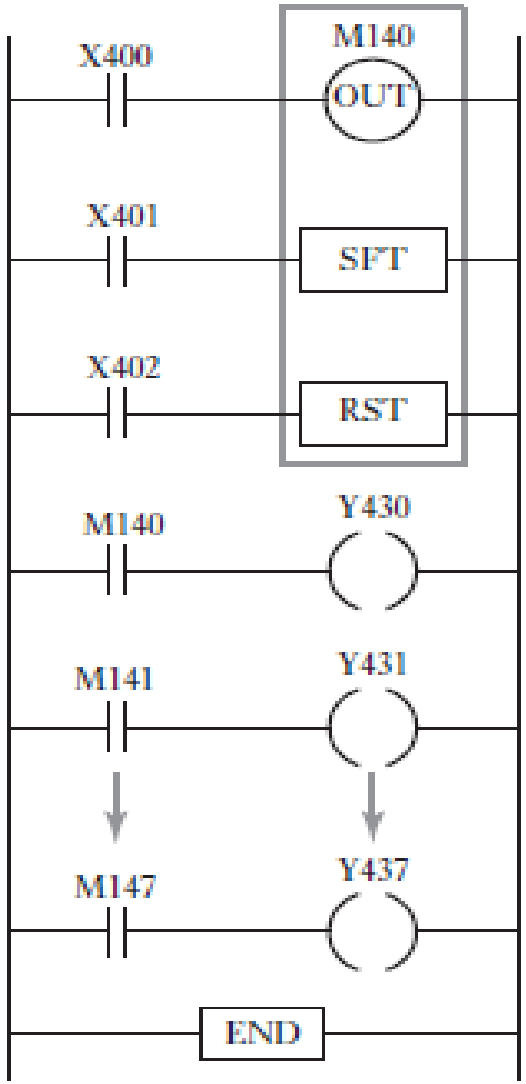
With SFT, a pulse moves the contents of the register along 1 bit at a time, the final bit overflowing and being lost.

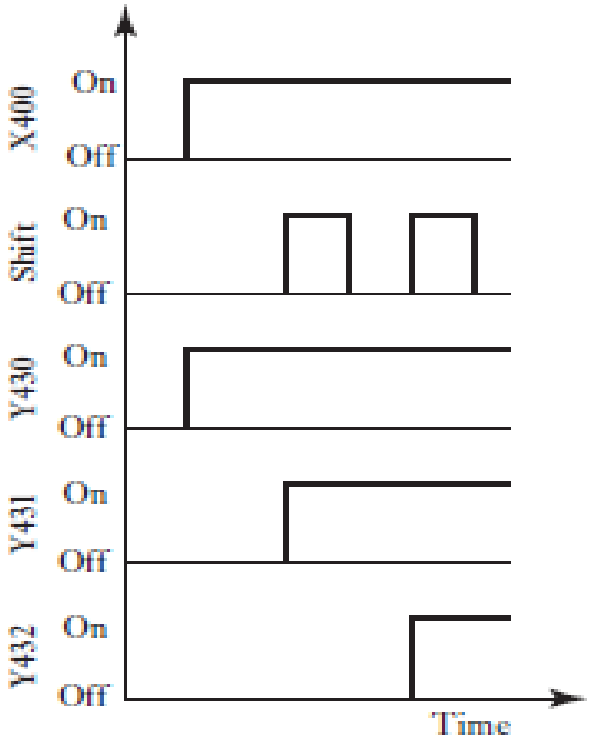
With RST, a pulse of the closure of a contact resets the register contents to all zeros.

PLC & SCADA

Software & Programming

7. Shift Register



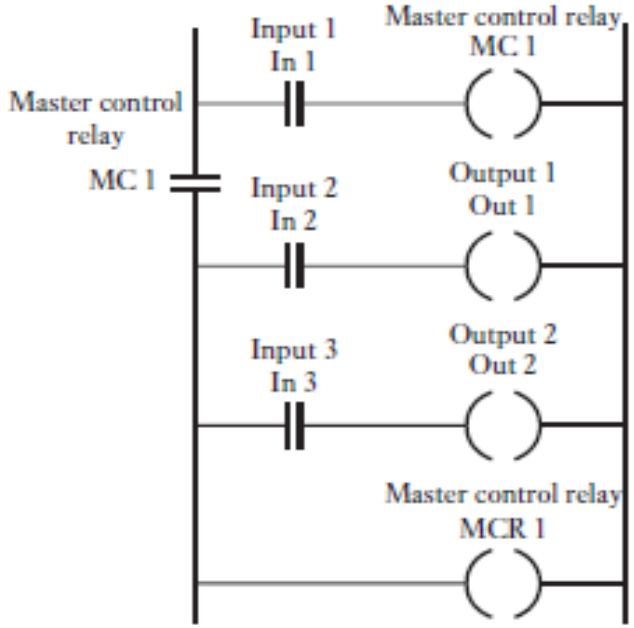


PLC & SCADA

Software & Programming

8. Master and jump controls:

A whole block of outputs can be simultaneously turned off or on by using the same internal relay contacts in each output rung so that switching it on or off affects every one of the rungs. An alternative way of programming to achieve the same effect is to use a master relay. We can think of it as controlling the power to a length of the vertical rails of the ladder. When there is an input to close Input 1 contacts, master relay MC1 is activated and then the block of program rungs controlled by that relay follows. The end of a master-relay-controlled section is indicated by the reset MCR. It is thus a branching program in that if there is Input 1 then branch to follow the MC1 controlled path; if not, follow the rest of the program and ignore the branch.



PLC & SCADA

Software & Programming

8. Master and jump controls:

With a Mitsubishi PLC, an internal relay can be designated as a master control relay by programming it accordingly. Thus, to program an internal relay M100 as a master control relay, the program instruction is:

MC M100

To indicate the end of the section controlled by a master control relay, the program instruction is:

MCR M100

PLC & SCADA

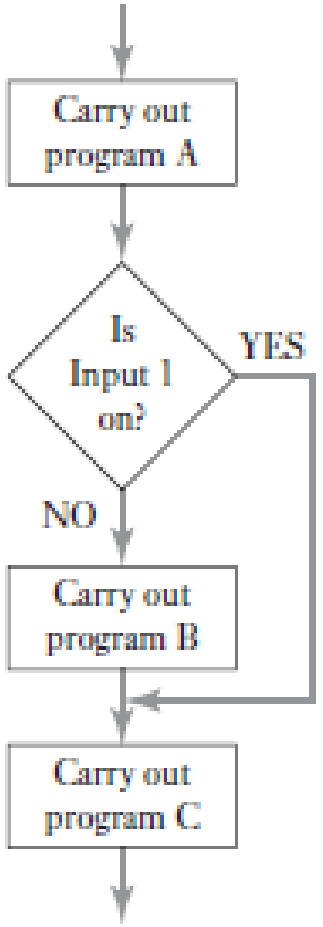
Software & Programming

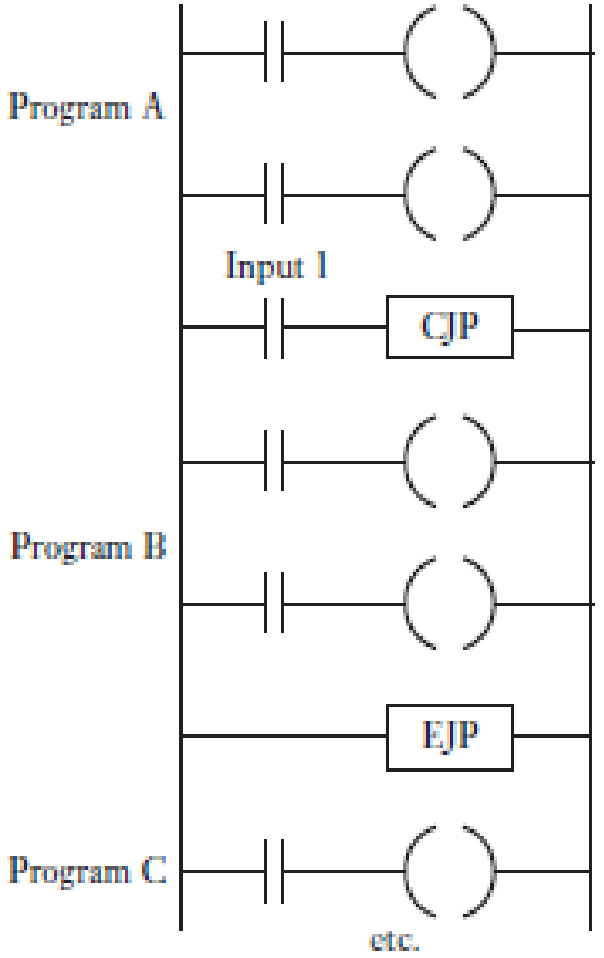
8. Master and jump controls:

A function which is often provided with PLCs is the conditional jump function. Such a function enables programs to be designed so that if a certain condition exists then a section of the program is jumped.

Figure illustrates this on a flow diagram and with a section of ladder program. Following a section of program A, the program rung is encountered with Input 1 and the conditional jump relay CJP.

If Input 1 occurs then the program jumps to the rung with the end of jump relay coil EJP and so continues with that section of the program labelled as C, otherwise it continues with the program rungs labelled as program B.





PLC & SCADA

Software & Programming

8. Data Handling:

Except for the shift register, the previous parts of this chapter have been concerned with the handling of individual bits of information, e.g., a switch being closed or not. There are, however, some control tasks where it is useful to deal with related groups of bits, e.g., a block of eight inputs, and so operate on them as a data word. Such a situation can arise when a sensor supplies an analogue signal which is converted to, say, an 8-bit word before becoming an input to a PLC.

The operations that may be carried out with a PLC on data words normally include:
1 Moving data;
2 Comparison of magnitudes of data, i.e. greater than, equal to, or less than;
3 Arithmetic operations such as addition and subtraction;
4 Conversions between binary-coded decimal (BCD), binary and octal.

As discussed earlier, individual bits have been stored in memory locations specified by unique addresses. For example, for the Mitsubishi PLC, input memory addresses have been preceded by an A, outputs by a Y, timers by a T, auxiliary relays by an M, etc. Data instructions also require memory addresses and the locations in the PLC memory allocated for data are termed data registers. Each data register can store a binary word of, usually, 8 or 16 bits and is given an address such as D0, D1, D2, etc. An 8-bit word means that a quantity is specified to a precision of 1 in 256, a 16-bit word a precision of 1 in 65536. Each instruction has to specify the form of the operation, the source of the data used in terms of its data register and the destination data register of the data.

PLC & SCADA

Software & Programming

8. Data Handling:

For data movement the instruction will contain the move data instruction, the source address of the data and the destination address of the data. Thus, the ladder rung could be of the form shown below. Such data transfers might be to move a constant into a data register, a time or count value to a data register, data from a data register to a timer or counter, data from a data register to an output, input data to a data register, etc.

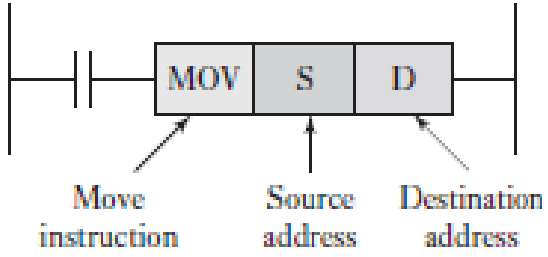


Fig: Data move.

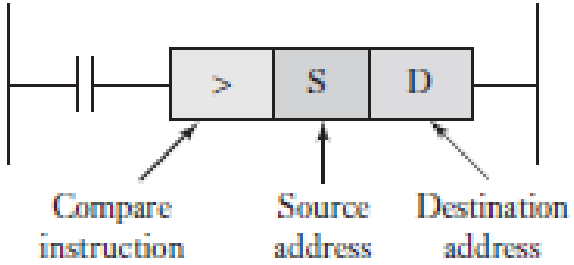


Fig: Compare Data.

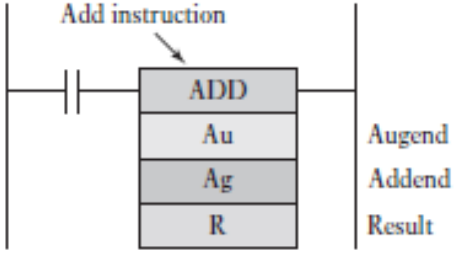


Fig: Add Data

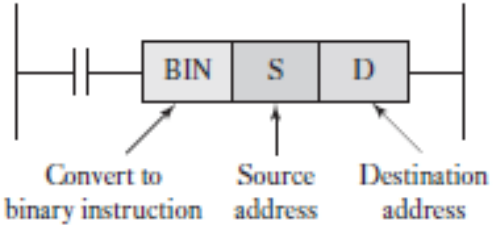


Fig: BCD to Binary

PLC & SCADA

Software & Programming

9. Analog Input & Output:

Many sensors generate analogue signals, and many actuators require analogue signals. Thus, some PLCs may have an analogue-to-digital converter (ADC) module fitted to input channels and a digital-to-analogue converter (DAC) module fitted to output channels.

An example of where such an item might be used is for the control of the speed of a motor so that its speed moves up to its steady value at a steady rate. The input is an on/off switch to start the operation. This opens the contacts for the data register and so it stores zero. Thus, the output from the controller is zero and the analogue signal from the DAC is zero and hence motor speed is zero. The closing of the start contacts gives outputs to the DAC and the data register. Each time the program cycles through these rungs on the program, the data register is incremented by 1 and so the analogue signal is increased and hence the motor speed. Full speed is realized when the output from the data register is the word 11111111. The timer function of a PLC can be used to incorporate a delay between each of the output bit signals.

PLC & SCADA

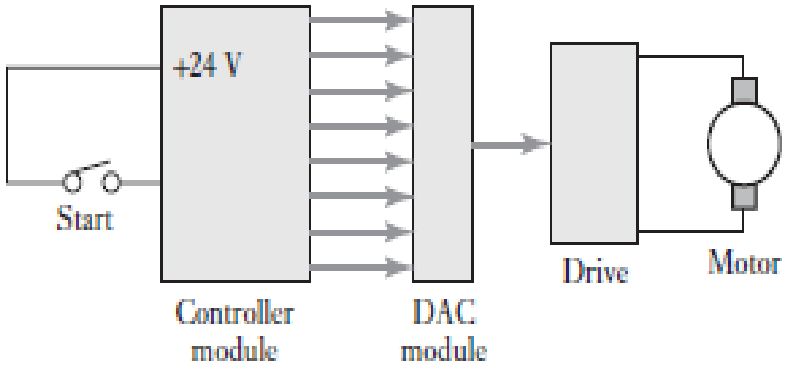
Software & Programming

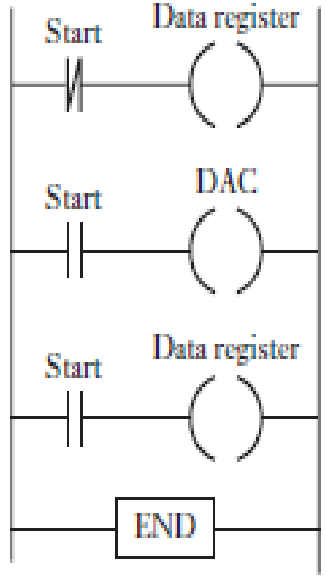
9. Analog Input & Output:

A PLC equipped with analogue input channels can be used to carry out a continuous control function, i.e., PID control .

Thus, for example, to carry out proportional control on an analogue input the following set of operations can be used:

1 Convert the sensor output to a digital signal.
2 Compare the converted actual sensor output with the required sensor value, i.e. the set point and obtain the difference. This difference is the error.
3 Multiply the error by the proportional constant *K*P.
4 Move this result to the DAC output and use the result as the correction signal to the actuator.

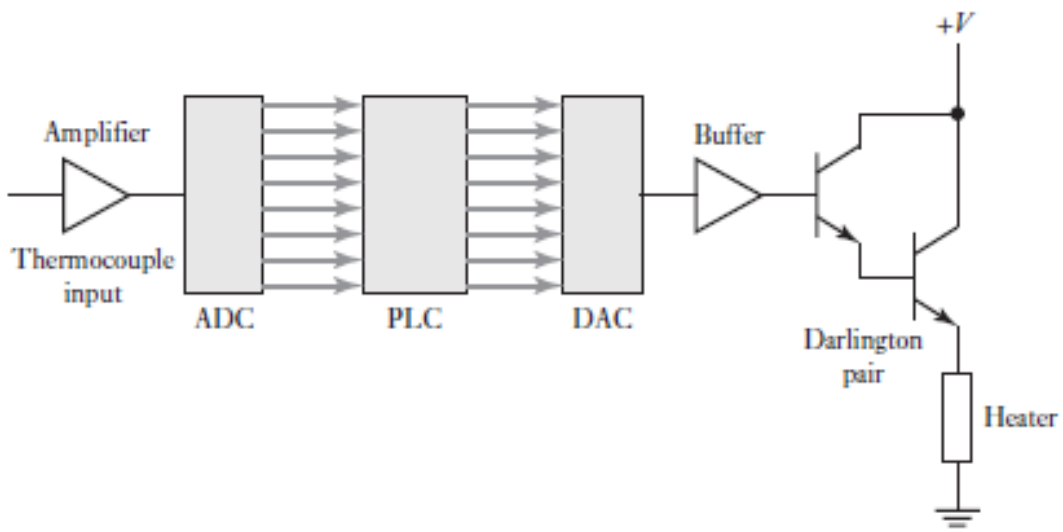




PLC & SCADA

Software & Programming

9. Analog Input & Output:



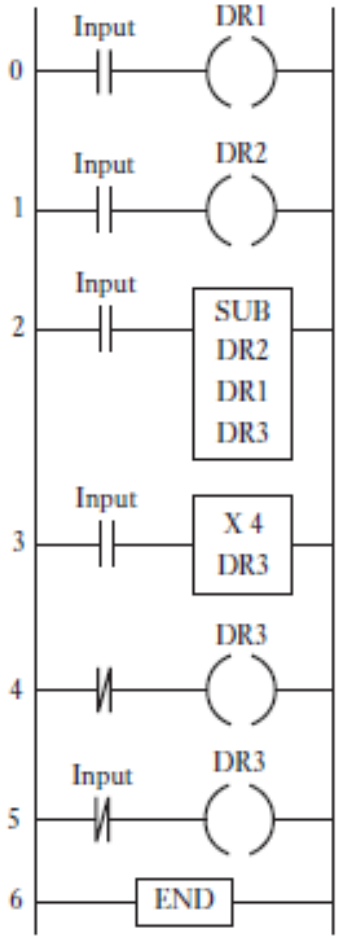


Fig: Proportional control of temperature

PLC & SCADA

Troubleshooting

Field instrument troubleshooting:

PLC systems are integral components of many industrial machines, and their failure can be catastrophic in terms of production. Let’s look at some of the most common PLC faults:

1. **Electrical Failures**
2. **Environmental Factors**
3. **Software Glitches**
4. **Component Wear and Tear**
5. **External Interference**
6. **Operator Error**

PLC & SCADA

Troubleshooting

Field instrument troubleshooting:

1. Electrical Failures:

Electrical issues can stem from power surges, voltage fluctuations, or short circuits. These can damage the PLC's internal components, leading to malfunctions. Regular electrical maintenance and surge protection can mitigate these risks.

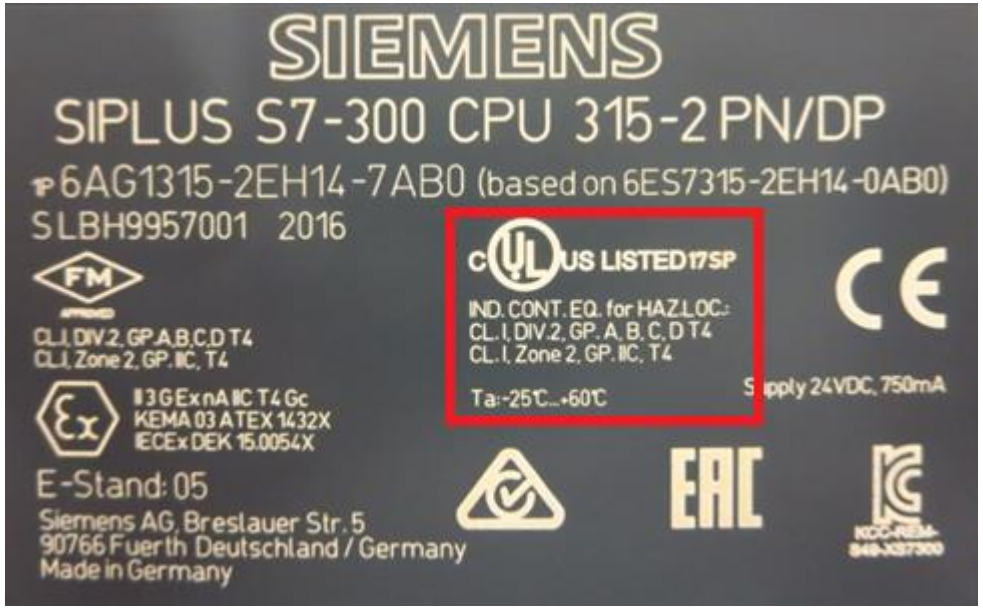
PLC & SCADA

Troubleshooting

Field instrument troubleshooting:

2. Environmental Factors:

PLCs are often exposed to harsh industrial environments, including extreme temperatures, humidity, dust, and corrosive substances. Over time, these factors can take a toll on the system's hardware, causing failures. Routine PLC repair and maintenance, including cleaning and protection measures, are essential for minimizing the impact of environmental stressors.



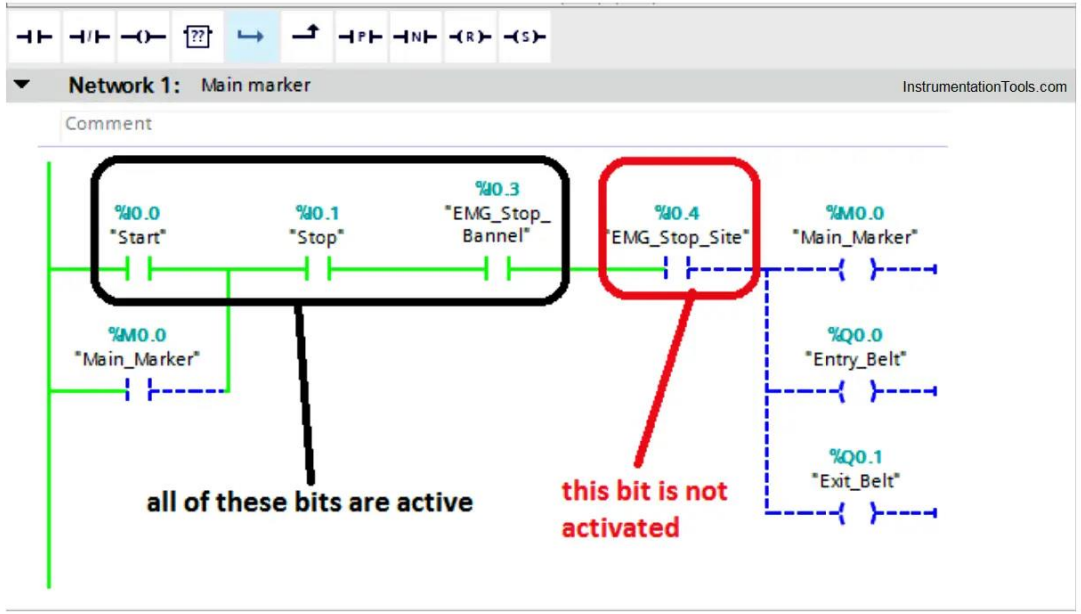
PLC & SCADA

Troubleshooting

Field instrument troubleshooting:

3. Software Glitches:

PLCs rely on software programs to execute tasks. Programming errors, software corruption, or incompatibilities can lead to system failures. Regular updates and diligent programming practices can help prevent these issues.



PLC & SCADA

Troubleshooting

Field instrument troubleshooting:

4. Component Wear and Tear:

Over time, the mechanical components of a PLC, such as fans, connectors, and cooling systems, can wear out. This can result in overheating, electrical issues, and reduced performance. Scheduled PLC repair and maintenance can identify and replace worn-out components.

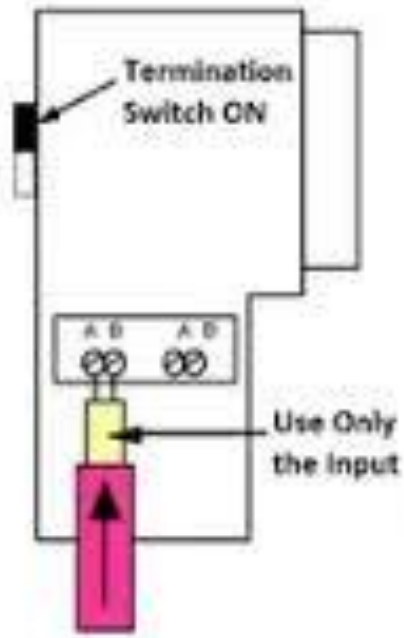
PLC & SCADA

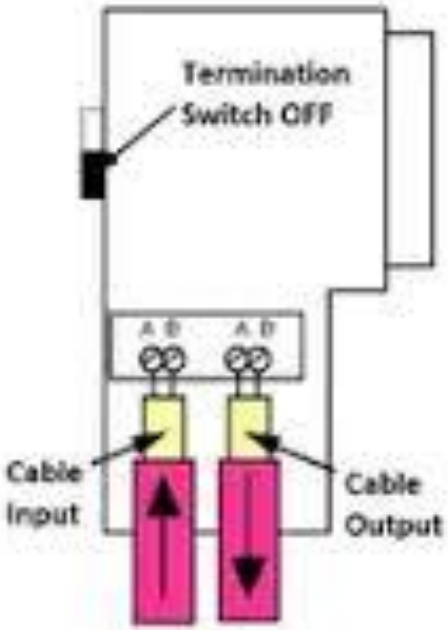
Troubleshooting

Field instrument troubleshooting:

5.External Interference:

Electromagnetic interference (EMI) or radio frequency interference (RFI) from nearby equipment or electrical sources can disrupt the operation of a PLC. Shielding, proper grounding, and isolation techniques can help mitigate these external interferences.





PLC & SCADA

Troubleshooting

Field instrument troubleshooting:

6. Operator Error

Human error, such as incorrect data input, accidental program modifications, or improper handling of the system, can lead to PLC failures. Adequate training and stringent access controls can minimize this risk.

PLC & SCADA

Signs of Impending PLC Failure:

Recognizing the symptoms of a potentially failing PLC control module is important in addressing issues promptly and efficiently. Here are some common signs to be aware of that might indicate an upcoming failure:

1. Input/Output Modules and Field Devices

A failing PLC control module may cause issues with input and output modules or field devices. Symptoms may include erratic readings, malfunctioning sensors, or actuators that fail to respond as expected.

2. Ground Integrity

Ground integrity problems can result in electrical noise and interference. Look out for erratic readings, unpredictable behavior, or voltage fluctuations in the control system, which can indicate ground-related issues.

3. Power Supply Failure

If the power supply to the PLC control module is compromised, the system may become unstable or fail to power on. This can result in intermittent failures or complete system shutdowns.

4. Electrical Noise Interference

Electrical noise interference can disrupt the normal operation of a PLC, causing incorrect readings and erratic behavior. This interference may be caused by nearby equipment or electrical sources.

5. Loss of Network Communication

A failing PLC control module might lose its ability to communicate over the network. Network communication errors, dropped connections, or missing data are telltale signs of this issue.

PLC & SCADA

Signs of Impending PLC Failure

6. Excessive Heat

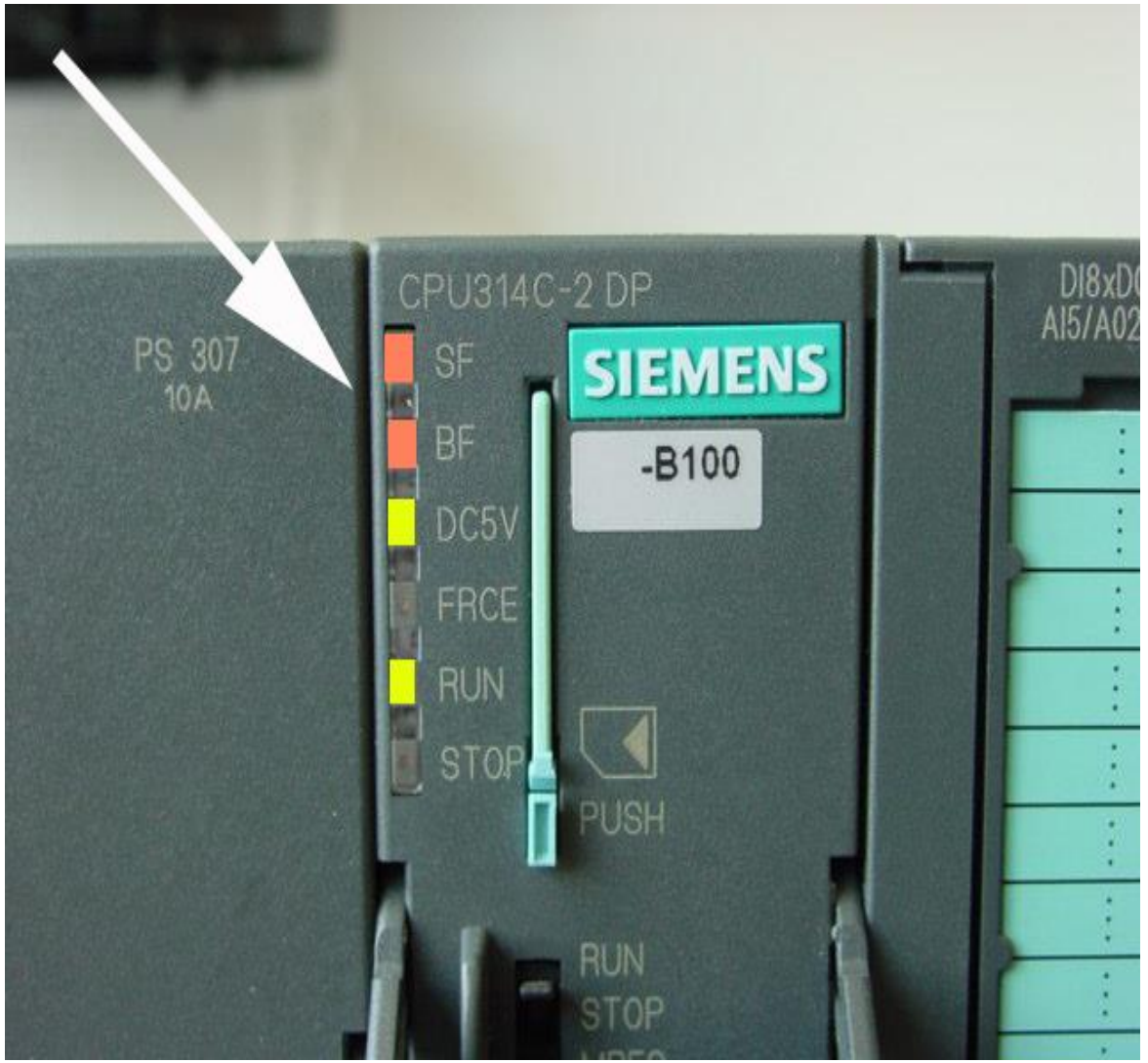
Overheating is a common symptom of a failing PLC control module. High temperatures can lead to component damage, affecting the system's overall performance and lifespan.

7. Conflict with Other Components

Incompatibility or conflicts with other components within the control system can cause the PLC to operate erratically. This might manifest as errors, unpredictable behavior, or even system crashes.

8. Corrupted Memory

Memory corruption within the PLC control module can result in unpredictable outcomes and unreliable operation. Symptoms can include program errors, data loss, or system crashes.



PLC & SCADA

Methods of troubleshooting PLC Failures:

1. Troubleshoot Ground Loops:

Ground loops are common in PLC failures, leading to electrical noise and interference. To address this issue, verify the ground integrity of your system. Ensure all components are correctly grounded, and there are no unintended ground paths. You may need to isolate sensitive components or use isolation transformers to minimize ground loop effects.

2. Check PLC Indicators:

PLC control modules often include status indicators that provide valuable information about the system's health. Examine these indicators to identify any error codes or fault messages. Consult the PLC's documentation to interpret the indicators correctly. This can help pinpoint the source of the problem and allow you to make quick PLC repairs.

3. Troubleshoot PLC Inputs:

Erratic readings or unresponsive sensors are common issues with failing PLCs. Start by checking the input devices, such as sensors and switches. Ensure they are correctly wired and functioning as expected. Testing these devices with a multimeter can help identify faulty input components.

4. Troubleshoot PLC Outputs

If actuators, relays, or other output devices fail to respond as they should, investigate the output side of the PLC. Examine the output devices for faults, damaged wiring, or loose connections. It's also crucial to verify that the PLC program logic for controlling these outputs is error-free.

5. Troubleshoot the CPU

The CPU is the brain of the PLC, and problems with the central processing unit can lead to system-wide issues. Examine the CPU for overheating, loose connections, or damaged components. If you suspect a problem with the CPU, it may be necessary to consult a PLC repair and maintenance service provider to perform a thorough diagnostic and repair.

PLC & SCADA

Create PLC backup :

Step 1 – Go online with the PLC

•Open the PLC project file with the Simatic manager application.

Go Online with the PLC, selected from the controller options dropdown

Step 2 – Save a backup

•Click on File and select Save As

•The ‘Save As’ Window appears.

•Browse to the folder where you want to save the file and give it a file name like ‘Project1’.

Step 3 – Ensure the backup has the latest variable values

•Click on Save

•A window pops up asking to upload tag values before saving

•Select ‘Yes’. This will ensure that the latest values for the variables are loaded into the project. Where this matters, is where you may have parameters that are adjusted from an HMI or SCADA, and you want these latest operator configurations to be backed up as well.

Step 4 – Close the application

•The backup has now successfully been made

•Select ‘Go offline’ to disconnect from the controller.

PLC & SCADA

Restore PLC backup :

Step 1 – Open the PLC project file to be downloaded

•Browse to the PLC project file that needs to be downloaded.

•Open the file in Simatic Manager.

Step 2 – Verify the controller

•Select Download from the controller options dropdown.

• The Download window appears.

•Verify that the correct controller has been selected, by ensuring that the name, type, path, and serial number is correct.

Step 3 – Ensure that the operation is in a safe state

•As a last check, make sure again that everyone involved is notified and that everything is safe.

•Click on the download button in the Download window.

•The ‘Downloading’ window appears, scrolling through all the different tags, programs, and routines being downloaded.

•Step 4 – Finalize the download

•After the download has successfully completed, the following popup appears.

•The controller was changed to program mode prior to the download and is still in program mode.

•Click on ‘Yes’, to change the controller to ‘Remote Run’.

•Save the project, go offline, and close the application.

PLC & SCADA

Industrial control system (ICS):

An industrial control system (or ICS) is a type of computer system that monitors and controls industrial processes and infrastructure. ICSs are used in a variety of industries, including oil and gas, chemical, water and wastewater, energy, food and beverage, pharmaceutical, automotive, and more. Each one operates differently and is designed to effectively manage duties electronically depending on the industry.

ICSs have become increasingly complex and powerful in recent years. Here, we will explore the different types of existing ICSs, the various components of an ICS environment, the most common threats associated with internal control systems, and how to keep them safe from threat factors through actions such as patching.

Types of Industrial Control Systems:

There are several types of industrial control systems used around the world, but the following are the most common ones:

1. **Supervisory Control and Data Acquisition (SCADA)**
2. **Distributed Control System (DCS)**

PLC & SCADA

Components of an ICS Environment :

•**Programmable Logic Controller (PLC)**: PLC is a type of hardware used in both DCS and SCADA systems as a control component of the overall system. Additionally, it offers local supervision of operations carried out by feedback control gadgets like sensors and actuators.

•**IT and OT:** The convergence between IT and OT provides enterprises with greater integration and visibility of the supply chain (which include critical assets, logistics, plans, and operation processes)

•**Remote Terminal Unit (RTU)**: An RTU is a field device that is microprocessor-controlled and delivers data back to the MTU in response to commands.

•**Control Loop**: Hardware components like PLCs and actuators are a part of every control loop. Signals from sensors, control valves, breakers, switches, motors, and other similar devices are interpreted by the control loop. The controller will use the variables that these sensors have measured to carry out a task or finish a process.

•**Control Server:** Hosts the DCS or PLC supervisory control software and communicates with lower-level control devices;

•**Human Machine Interface (HMI):** Graphical user interface (GUI) which allows interaction between the hardware and the human operator;

•**SCADA Server or Master Terminal Unit (MTU):** A device used to send instructions to out-of-band RTUs;

•**Intelligent Electronic Device (IED):** A gadget capable of acquiring data, perform local processing and control, and communicating with other connected devices.

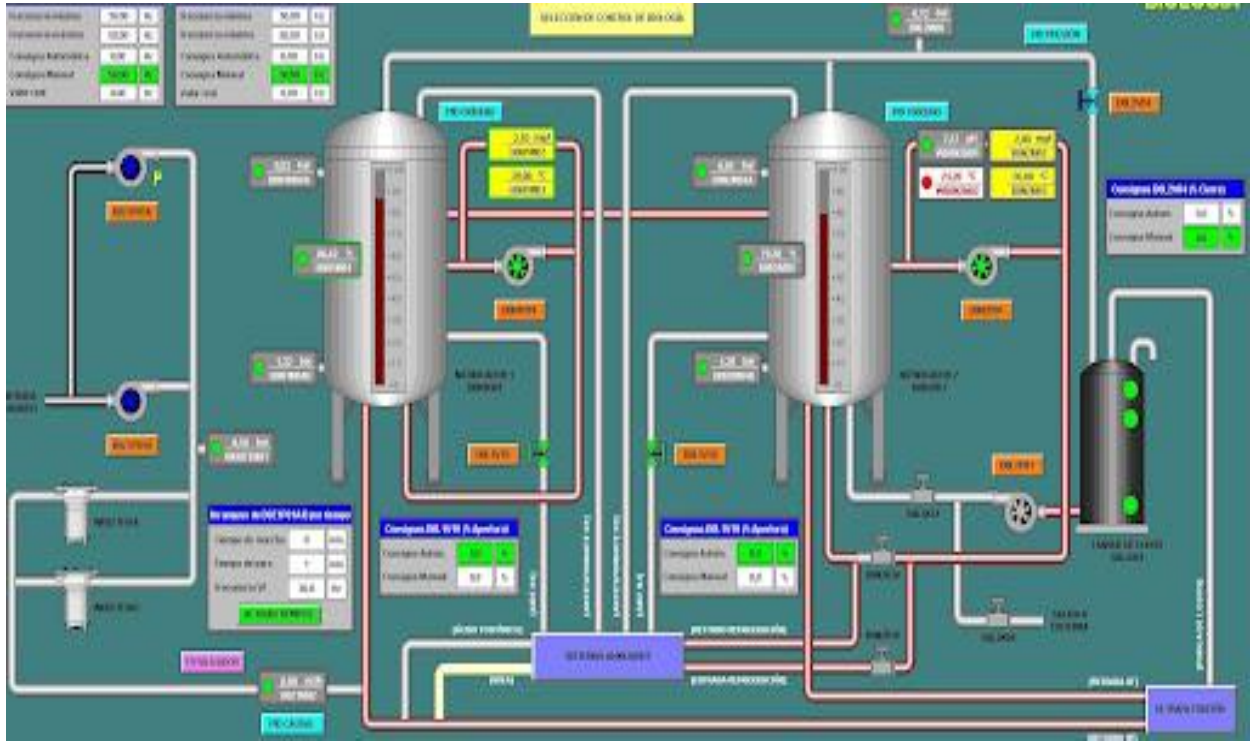
•**Data Historian:** A centralized database for logging all process information within an ICS environment and exporting data to the corporate IS.

PLC & SCADA

Supervisory control and data acquisition (SCADA):

SCADA (supervisory control and data acquisition) is a control system architecture comprising computers, networked data communications and graphical user interfaces for high-level supervision of machines and processes. It also covers sensors and other devices, such as programmable logic controllers, which interface with process plant or machinery.

The operator interfaces which enable monitoring and the issuing of process commands, such as controller setpoint changes, are handled through the SCADA computer system. The subordinated operations, e.g., the real-time control logic or controller calculations, are performed by networked modules connected to the field sensors and actuators.



PLC & SCADA

Key Components of a SCADA System:

Sensors and Manual Input

Digital or analog sensors serve as measuring tools that collect data from various parts of the plant. Sensors may range from simple binary options, such as an on or off signal, to more complex tools that measure flow rate, temperature, and pressure. In addition, technicians or operators at the remote or central location can manually input data into the system.

Conversion Units

Data collected by sensors is only useful if it can be converted into a form that is easily comprehensible. Remote terminal units (RTU) and programmable logic controllers (PLC) are the devices that can translate the collected data into usable information. Since information is collected throughout an entire system, the sheer amount of data can be great.

Supervisory System

Data feeds that are converted by the RTUs and PLCs meet at a master unit known as the supervisory system or the human-machine interface (HMI). This interface brings useful information to the maintenance team. At this point, one operator can have a complete picture of an entire process or system. The data is presented in an easily digestible format, and the employee can take control of certain pieces of equipment to make repairs or isolate failures.

Communication Network

All the SCADA components are located throughout the plant and must be linked together by a communication network. Conventionally, telephone lines and circuits have served as this network with newer wireless options now available that use radio waves or cellular satellites.

PLC & SCADA

Application of SCADA system:

A wide variety of companies, organizations, and businesses can use SCADA systems to improve efficiency, share quality data across departments, and better identify and address systems issues.

1. Both private and public sector organizations can benefit from SCADA systems including small, basic manufacturing plants to large, multi-million-dollar corporations.

2. Companies within the oil and gas, power, water, and transportation industries often employ SCADA systems, as well as businesses in energy, food, healthcare, and recycling.

3. SCADA systems can be configured to do everything from managing the operations of freezer and refrigeration systems at a food distribution company to reducing downtime on a production line at a manufacturing plant.

4. SCADA systems help businesses comply with health and safety regulations, meet government compliance requirements, boost efficiency, and save money.

PLC & SCADA

How SCADA Benefits Maintenance and Reliability Efforts:

Since SCADA systems provide flexible, scalable means to monitor what's happening throughout a complex production process, on a shop floor, or within remote substations, they can make a significant contribution to maintenance and reliability efforts.

Reliability-centered maintenance (RCM) is a maintenance strategy that boosts equipment and asset reliability to a top priority. The strategy is designed to optimize the maintenance system of an entire organization with the intent to improve efficiency and timely production. RCM typically focuses on identifying and prioritizing different failure modes. This focus helps in scheduling activities that will prevent major system failure.

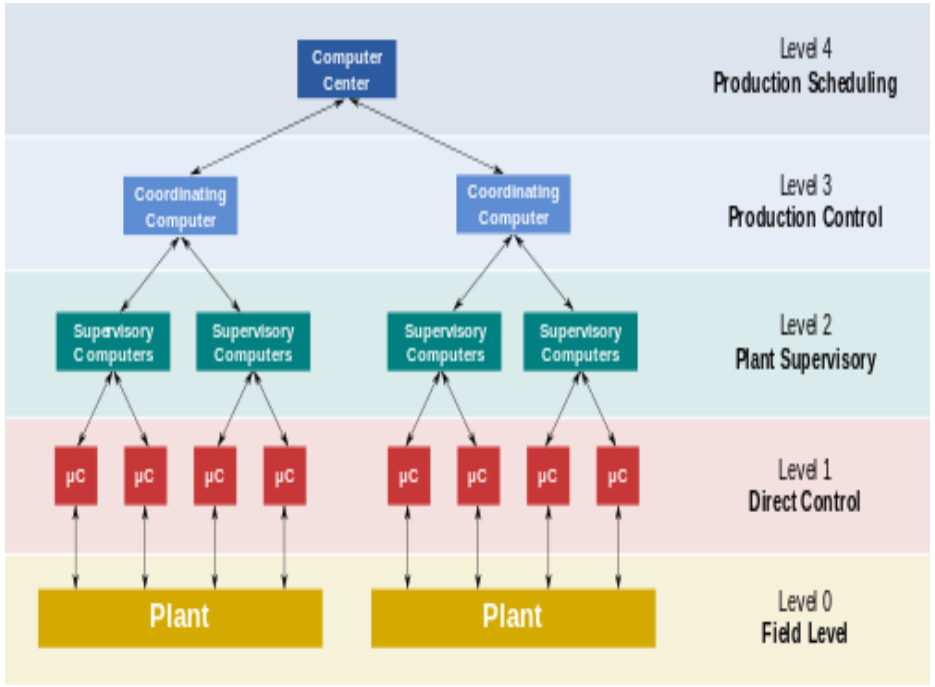
It's easy to see how SCADA can go hand in hand with an RCM system, as it provides a great deal of automated information and data on the performances of various assets and machinery in a plant. SCADA also allows human intervention early in the process, preventing the failures that an RCM strategy is designed to seek and identify.
Here are some real-life applications.

1. **Automating Electric Distribution**
2. **Identifying Failures on a Production Floor**
3. **Manage Sensitivity and Security in IT**
4. **Nerve Center of Alternative Energy**

PLC & SCADA

Distributed Control System (DCS):

A distributed control system (DCS) is a computerized control system for a process or plant usually with many control loops, in which autonomous controllers are distributed throughout the system, but there is no central operator supervisory control. This contrasts with systems that use centralized controllers; either discrete controllers located at a central control room or within a central computer. The DCS concept increases reliability and reduces installation costs by localizing control functions near the process plant, with remote monitoring and supervision.



Distributed control systems first emerged in large, high value, safety critical process industries, and were attractive because the DCS manufacturer would supply both the local control level and central supervisory equipment as an integrated package, thus reducing design integration risk. Today the functionality of Supervisory control and data acquisition (SCADA) and DCS systems are very similar, but DCS tends to be used on large continuous process plants where high reliability and security is important, and the control room is not geographically remote. Many machine control systems exhibit similar properties as plant and process control systems do.

PLC & SCADA

DCS VS SCADA: What are the Similarities?

DCS and SCADA have similarities in their basic functionalities. Let us explore the similarities between DCS and SCADA:

Control: DCS and SCADA are designed to monitor and control industrial processes. They can be used to control the operation of equipment, monitor the status of processes, and adjust parameters to maintain optimal performance.

Data acquisition: Both systems are designed to collect data from various sensors and devices in the industrial process. This data can be used to generate reports, analyze trends, and improve the overall efficiency of the process.

Human-machine interface: DCS and SCADA systems provide a human-machine interface (HMI) that allows operators to interact with the system. A graphical depiction of the process is provided by the HMI, enabling operators to monitor and control it in real time.

Alarms and alerts: These two systems can generate alarms and alerts when certain conditions are met. This can be used to notify operators of potential issues or to alert them about the need for action.

Remote access: Remote access to both DCS and SCADA systems enables operators to check and manage the procedure from a distance. This can be especially useful for large-scale industrial processes that require continuous monitoring and control.

PLC & SCADA

DCS vs SCADA: What's the Difference?

DCS and SCADA are control systems used to monitor and control industrial processes. However, they differ in their scope and functionality. DCS is a comprehensive control system designed to control the entire process from a central location, whereas SCADA is primarily used to monitor and control specific processes or equipment.

Architecture: DCS architecture is distributed, meaning that control and monitoring functions are distributed across multiple controllers connected through a network. This makes DCS systems more suitable for complex industrial processes, such as chemical plants, oil and gas refineries, and power plants. SCADA architecture, on the other hand, is typically centralized, where data is collected and monitored from a central location. This makes SCADA systems more suitable for relatively simple applications, such as water treatment plants.

Functionality: DCS systems are designed for more complex process control functions, such as closed-loop control, advanced regulatory control, and sequence control. In contrast, SCADA systems are primarily intended for monitoring and data collection, with the ability to issue commands and alerts to operators. However, some SCADA systems offer more advanced control functions, blurring the line between the two systems.

PLC & SCADA

DCS vs SCADA: What's the Difference?

Integration: DCS systems are designed to be highly integrated with other automation systems such as programmable logic controllers (PLCs), human-machine interface (HMI), and other subsystems. This integration allows DCS systems to provide a centralized control and monitoring platform to manage multiple processes across different areas of a plant or facility. On the other hand, SCADA systems are more standalone and used for monitoring and data acquisition only. SCADA systems are usually designed to be compatible with various third-party systems, but they are more tightly integrated with these systems than DCS systems.

End User: DCS is suitable for large-scale industrial processes, such as power plants, refineries, and chemical plants. These processes involve multiple units working together to produce the desired output. DCS is designed to integrate all these units into a single control system, allowing operators to monitor and control the entire process from a central location. DCS is also designed to handle large volumes of data, allowing operators to make informed decisions based on real-time data. On the other hand, SCADA is typically used in smaller-scale processes involving specific equipment or machines. SCADA systems are designed to monitor and control specific equipment, such as pumps, valves, or motors. SCADA systems have applications in sectors such as manufacturing, building automation, and water treatment plants.

PLC & SCADA

Common ICS Threats:

Attacks against ICS systems are frequently focused attacks that leverage the entry point provided by the industrial control system to obtain access to the system and laterally migrate into the company. Every ICS continuously integrates new technologies and applications in both IT and OT to enhance system functionality and productivity. Because IT and OT are now integrated, threat factors may target them more easily.

Some of the ways in which attackers may gain access to these systems are:

- **Lateral movement from IT network**
- **Direct access to Internet-facing systems**
- **Phishing attacks**
- **Vulnerable IoT and Internet-connected systems**

The lack of security solutions used in OT infrastructure to secure legacy control systems like SCADA is one of their prevalent weaknesses. Additionally, businesses must deal with the growing security issues in cutting-edge technologies like cloud computing, big data analytics, and the internet of things (IoT) .

PLC & SCADA

Protect Industrial Control Systems:

A list of recommendations you should follow if you want to protect your industrial control systems are :

- **Application Whitelisting:** Use application whitelisting to protect your infrastructure from potentially harmful programming;
- **Configuration and Patch Management:** By implementing configuration and patch management solutions your control systems will be constantly updated and secured;
- **Network Segmentation** : segment your network into logical parts and restrict host-to-host communication paths to reduce attack surface;
- **Multi-factor authentication** must be implemented, and the principle of least privileged (POLP) must be used;
- **Secure Remote Access**: implement strong authentication , access control, and encryption to prevent exploitation of the systems by unauthorized actors;
- **Traffic Monitorization** : monitor the traffic within the control networks and ICS perimeters;