# Combinational and sequential systems

Prof. Cesar de Prada
Dpt. of Systems Engineering and Automatic Control
UVA
prada@autom.uva.es

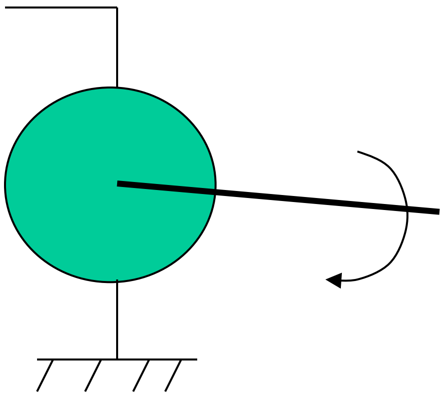
# Outline

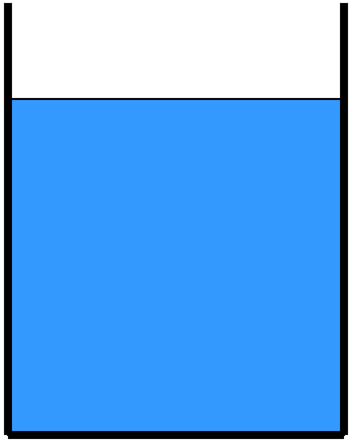
- ✓ Discrete events systems
- ✓ Combinational logic
- ✓ Sequential systems
- ✓ Programmable Logic Controllers (PLC)
  - Functions and architecture
  - Software
- ✓ Batch process Control
- ✓ Safety systems

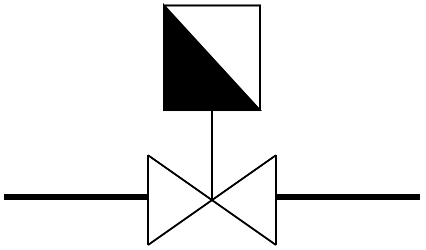
# Discrete events systems

- ✓ Many processes involve discontinuous elements
- ✓ Some of its variables only take an integer number of values
- ✓ The values of some variables only change at certain time instants (events)
- ✓ Logic and sequential control problems

# Discrete states







Motor:

Stop / Running

Tank:

Full / Empty

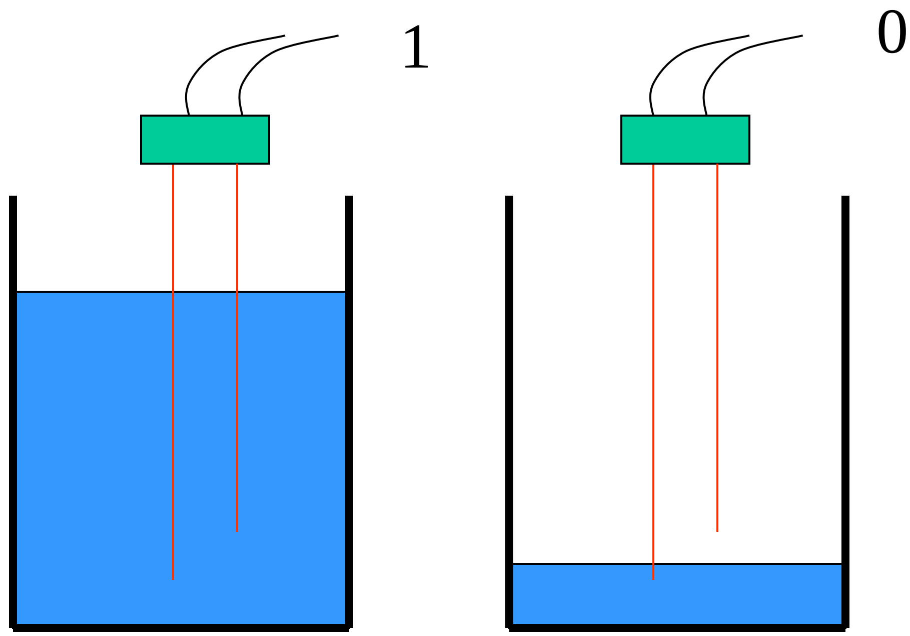
Valve:

Open / Closed

# Instrumentation

The signal from the instrument takes only two values and changes when the event takes place:

Minimum **Level detector**: When the level is above the minimal level, the signal is activated



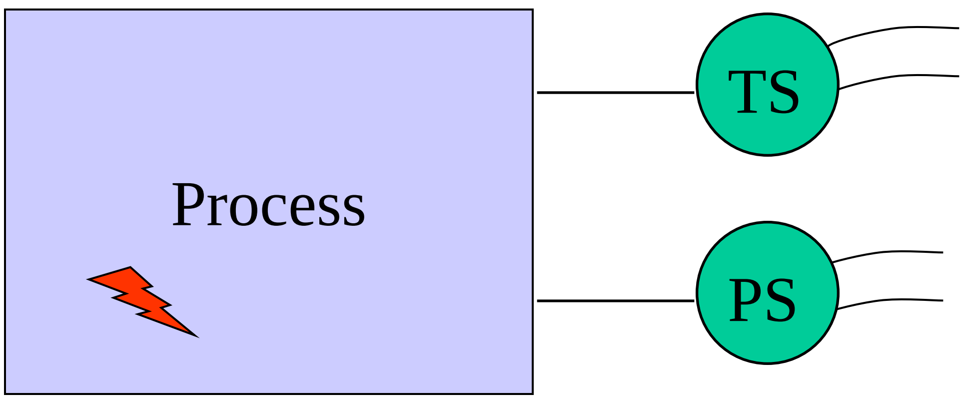

Closed circuit

Open circuit

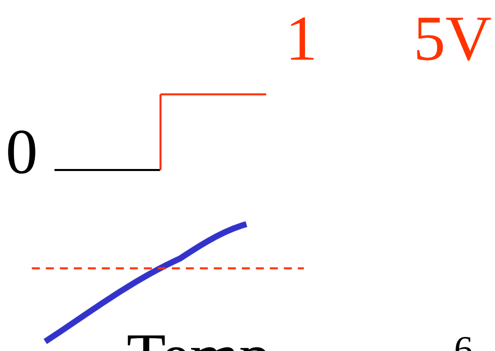
NO / NC  normally open /close

# Instrumentation

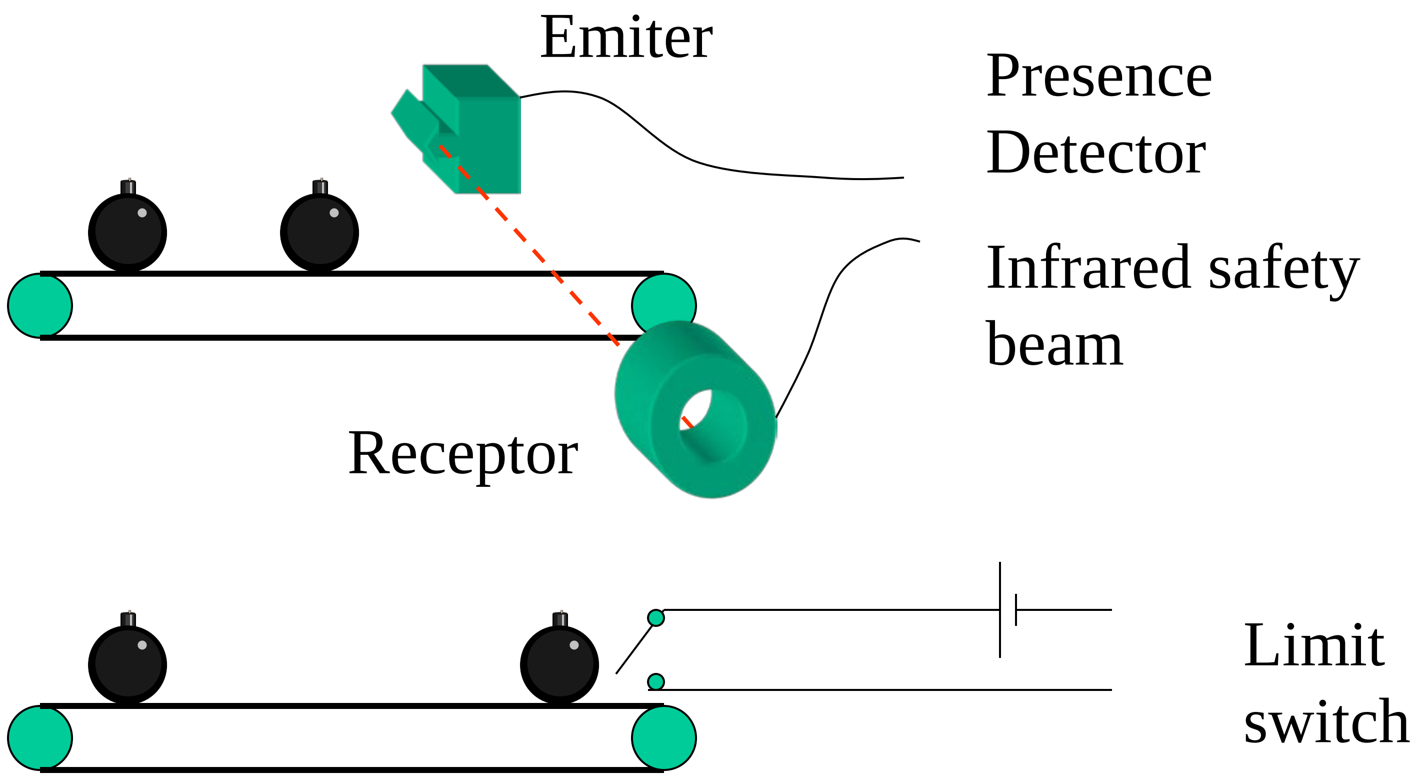
**Thermostat**:
When the temperature rises up to a certain limit the sensor is activated



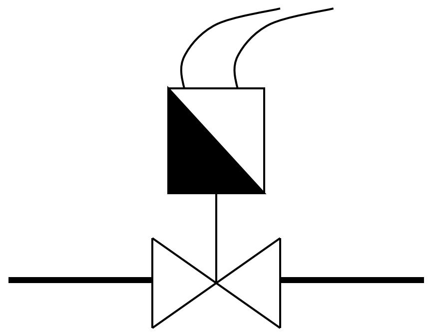

**Pressure switch**

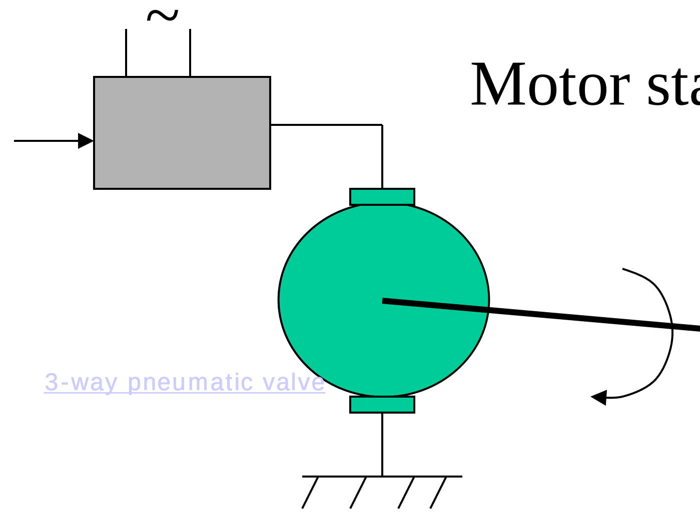


# Instrumentation (Detectors)



# Instrumentation (Actuators)






Motor starter

On/off valve

Electrovalve

Solenoid valve

Pneumatic valve

# Combinational systems

- The value of the system output depends only on the current value of the system inputs, through combinations of the logic functions AND, OR, NOT
- IF ( Logic statements )

  THEN ( actions)
- Associated to alarms or logic of operation
- How to represent the logic and perform the actions?

# Combinational Logic

| AND | 1 | 0 |
|---|---|---|
| 1 | 1 | 0 |
| 0 | 0 | 0 |

| OR | 1 | 0 |
|---|---|---|
| 1 | 1 | 1 |
| 0 | 1 | 0 |

| NOT | 1 | 0 |
|---|---|---|
| | 0 | 1 |

$A.B$ AND

$A+B$ OR

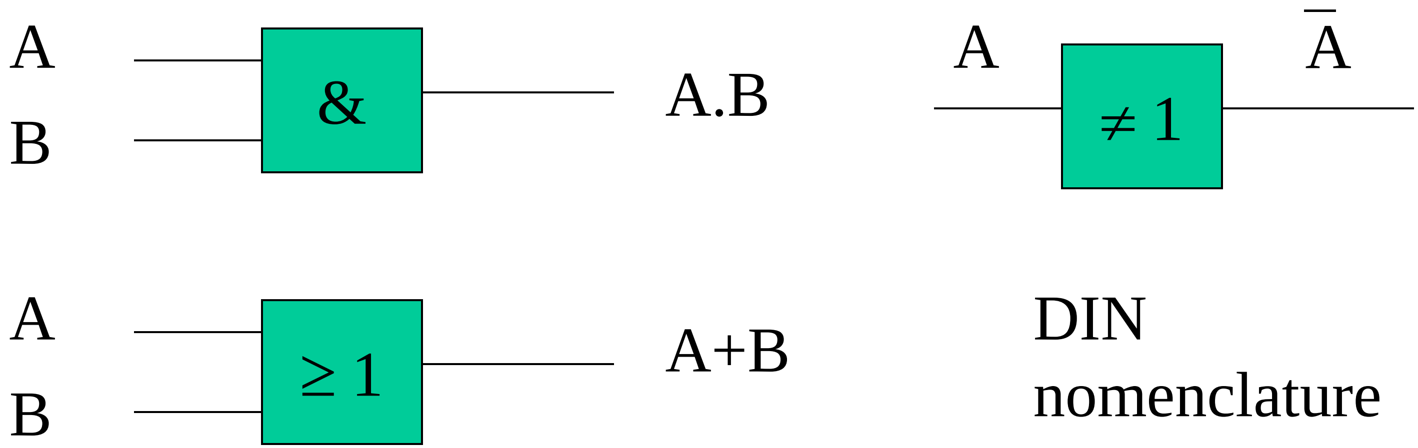
$\overline{A}$ NOT

Morgan Laws

$$\overline{(A+B)} = \overline{A}.\overline{B}$$

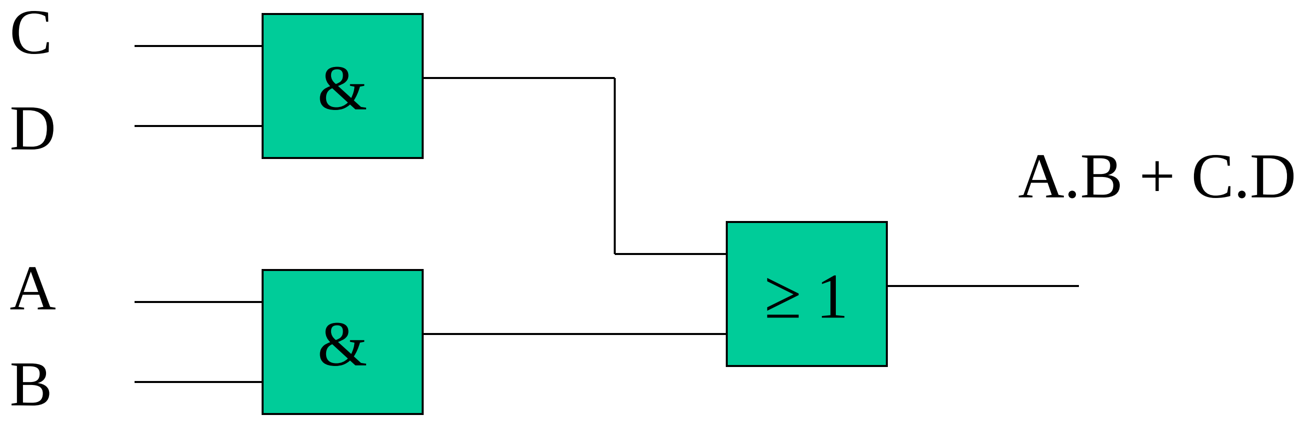
$$\overline{A.B} = \overline{A} + \overline{B}$$

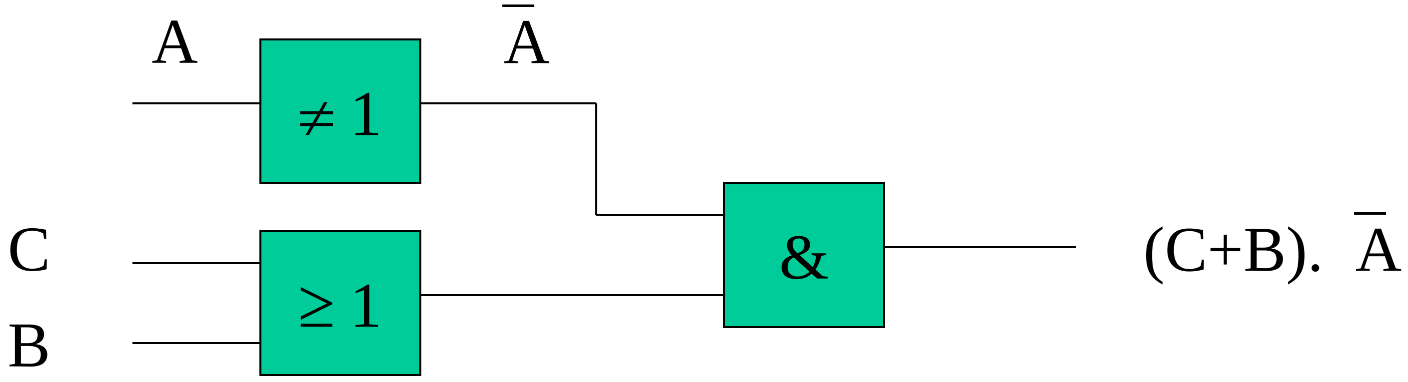
# Logic gates



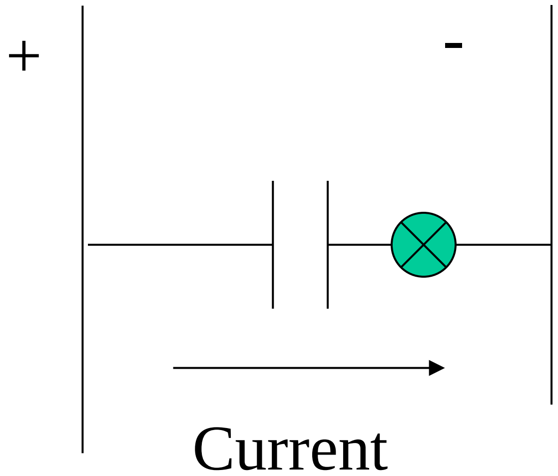

Logic expressions can be assimilated to electrical circuits where true/false can be represented by the presence or not of an electrical signal and the conclusion is expressed in terms of the value of the output signal

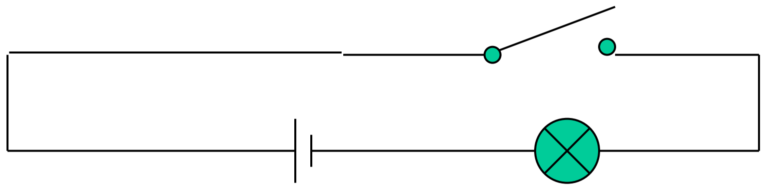
# Logic gates (Block functions)



# Contact (Ladder) diagrams



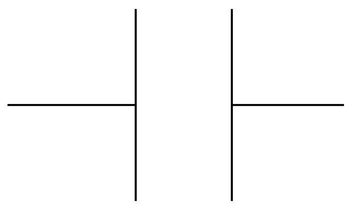



If the switch is closed, then the current flows through the circuit and the light bulb is activated

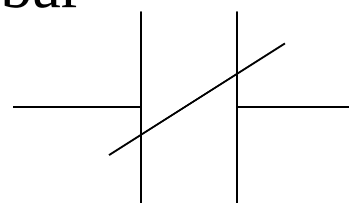
Logic expressions can be assimilated to electrical circuits where false/true can be represented by open or closed switches and the conclusion is expressed in terms of  the current the flows or not in the circuit

# Ladder diagrams

Load



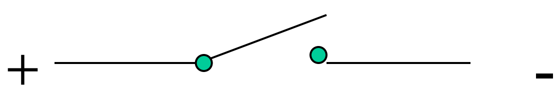
Load bar



Contactor

Normally Open Contact

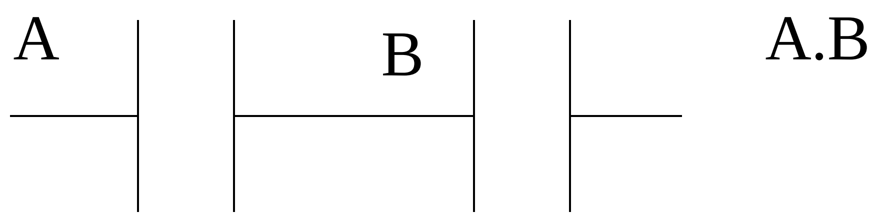
Normally Closed Contact



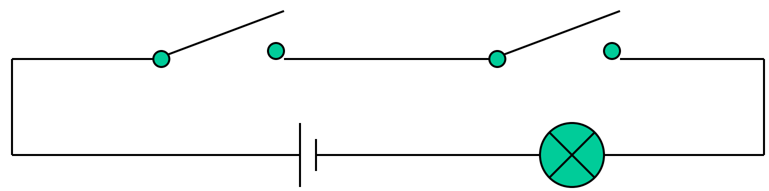

If the event takes place, the contact will close and the current will flow

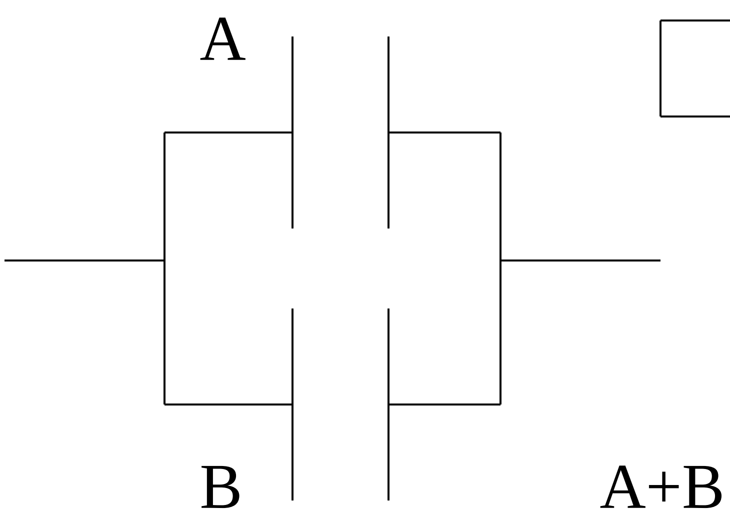
If the event takes place, the contact will open and the current will stop flowing

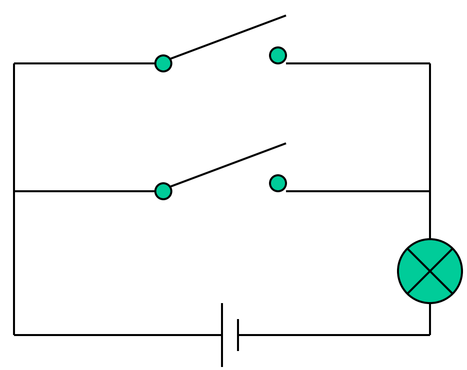
# Combinational logic using contacts




The ligh bulb is on if both, A and B, are closed

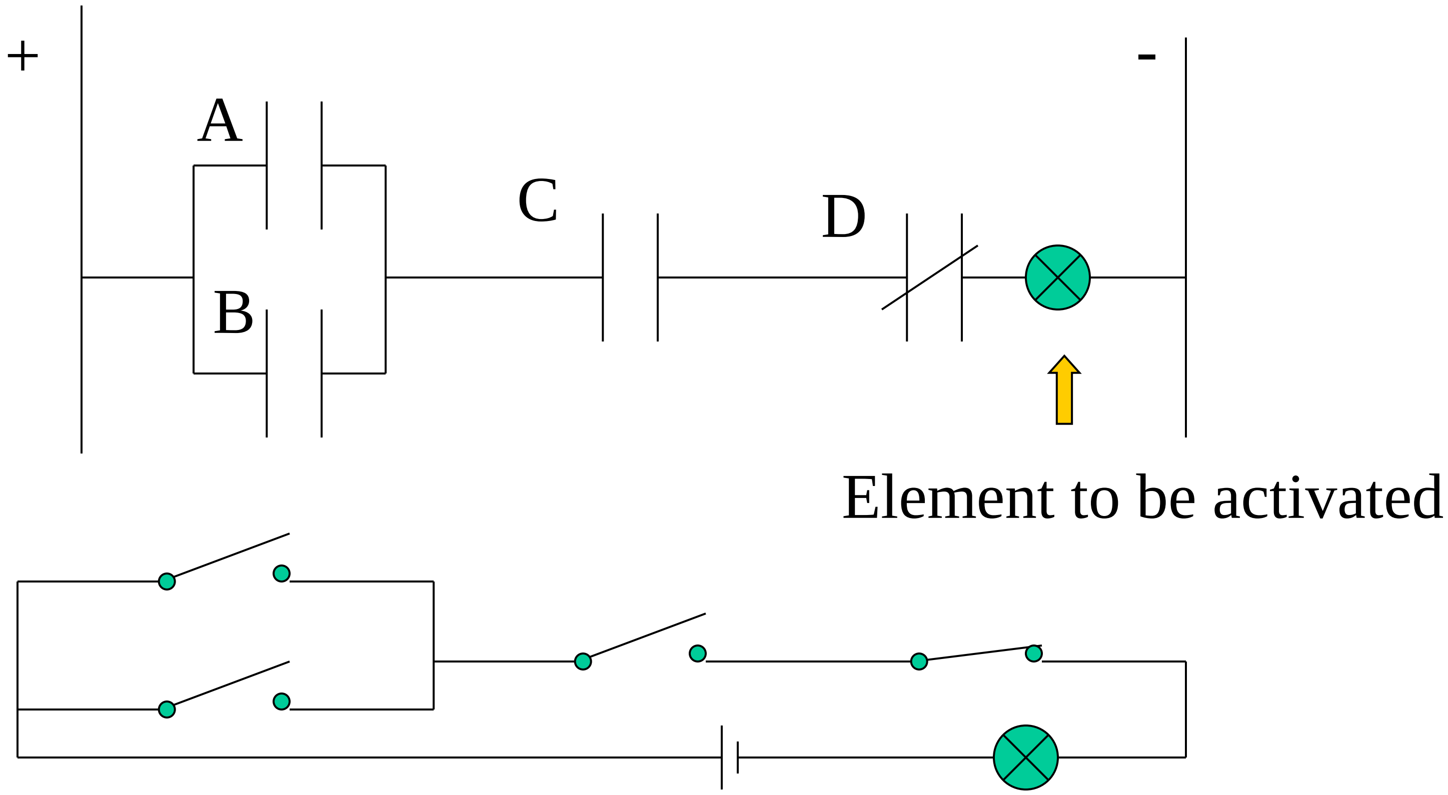




The ligh bulb is on if any, A or B, is closed

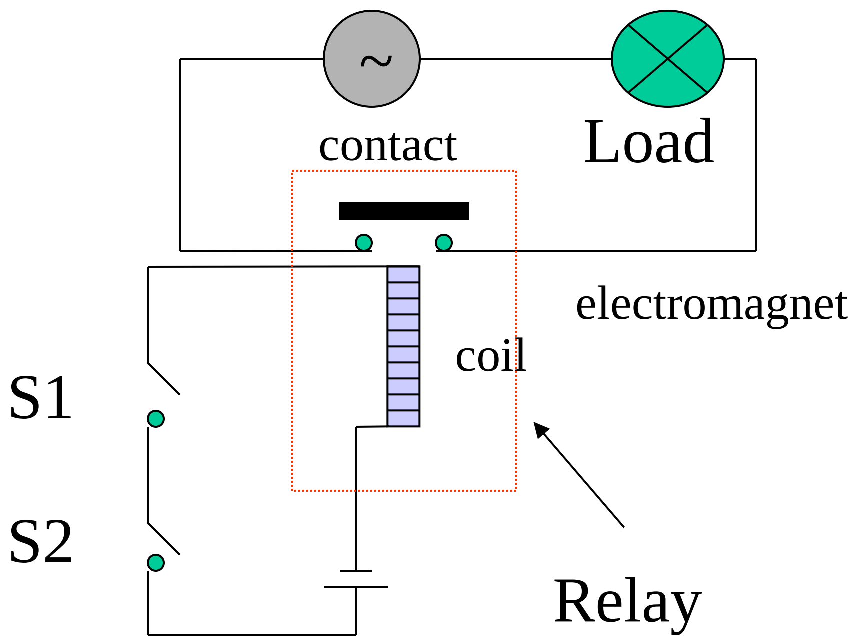
# Contact diagrams




Logic function: $(A+B).C.\ \overline{D}$

# Relay

Most of the times the current circulating on the circuit is too small to activate a dispositive (light bulb, motor, horns,..). For this purpose, a relay is used.




Mechanical

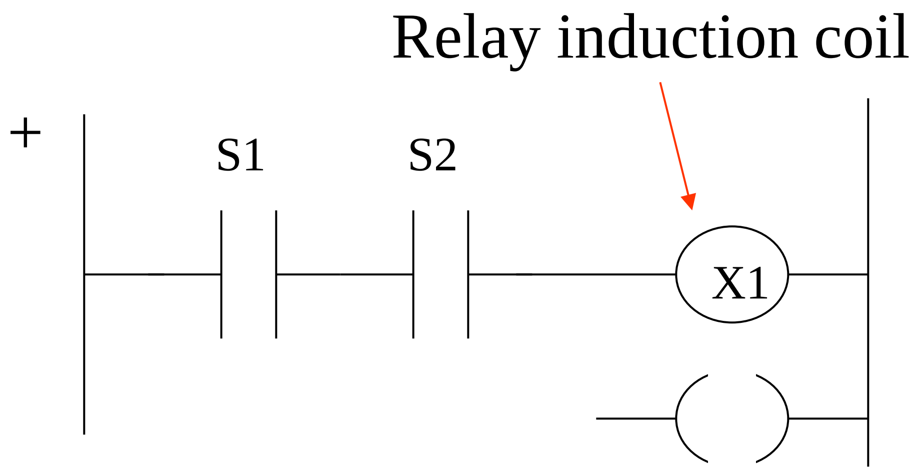
Solid state

IF (S1= closed AND S2= closed)

THEN coil activated

The relay will let activate the load with a higher current

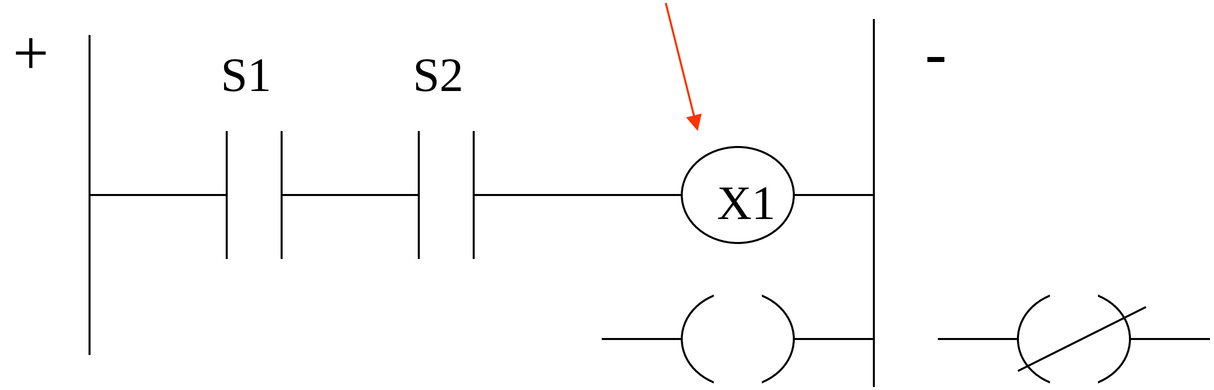
# Contact diagrams




In the contact (ladder) diagram only the coil of the relay is represented

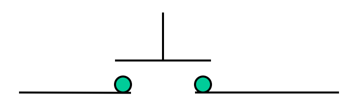
# Contact diagrams

Relay induction coil

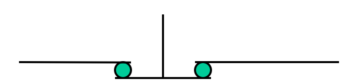



S1 or S2 can be any element providing a 0 – 1 signal: timers, counters, detectors, switches, etc.

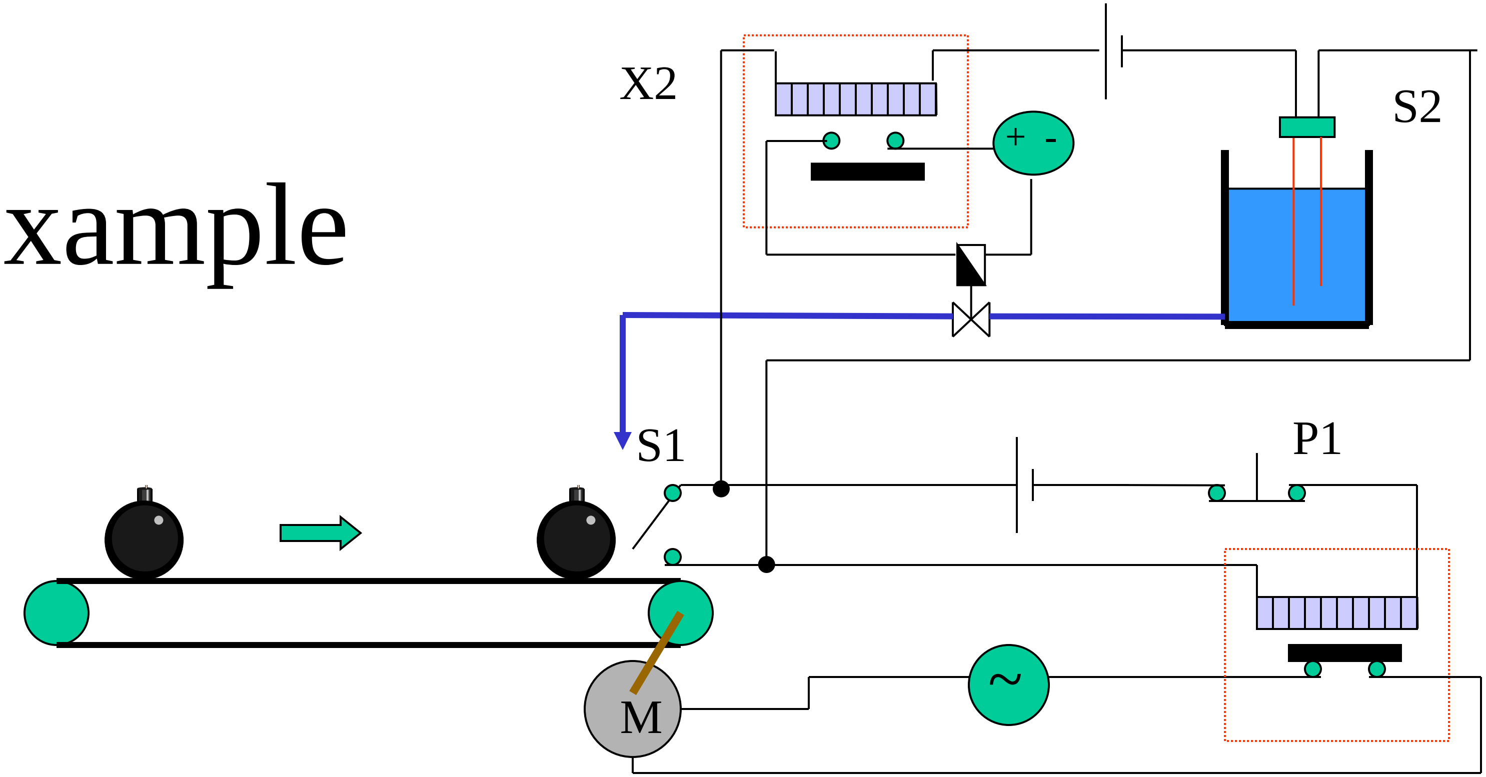
Push-button switch normally open



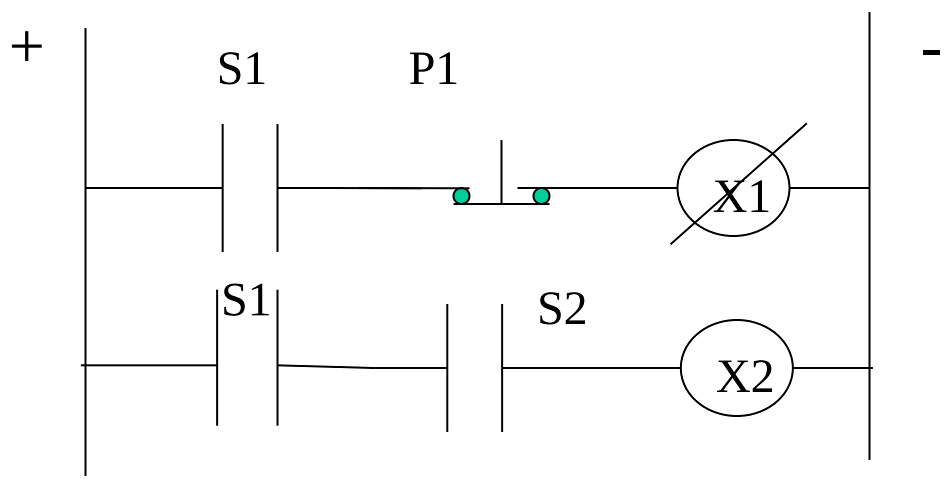
Push-button switch normally closed



# Example

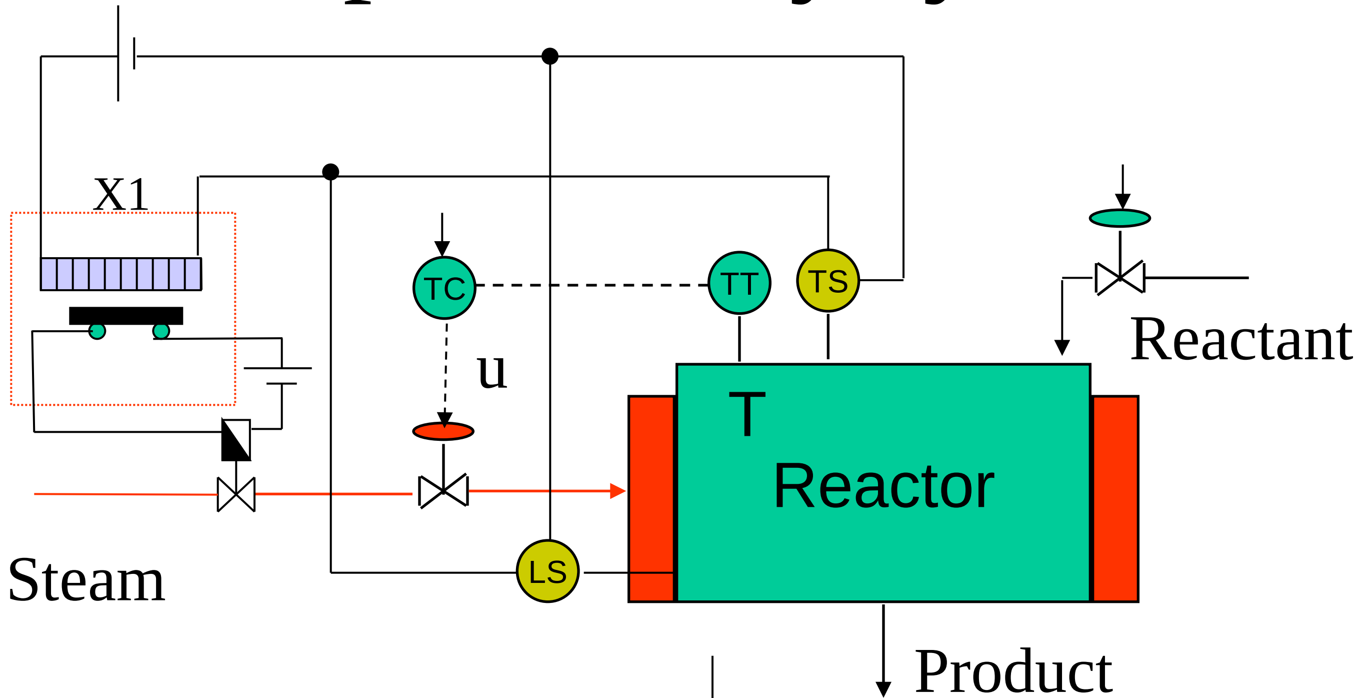


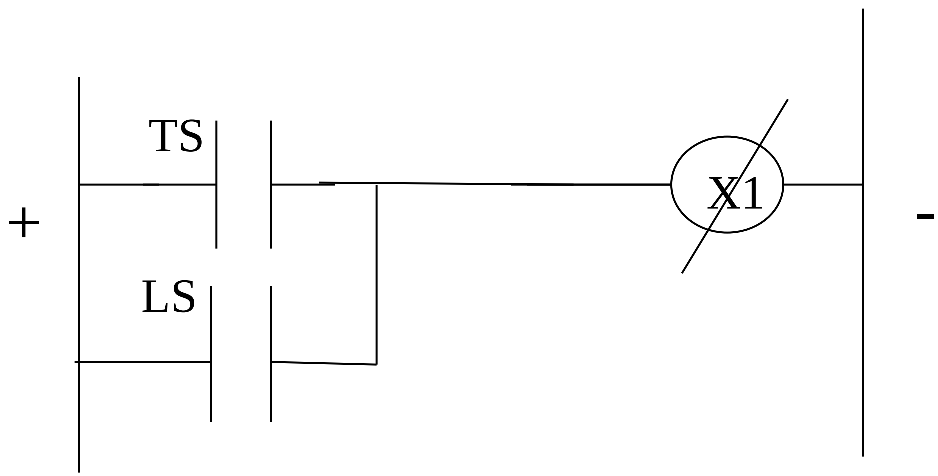

The bottle should stop al the end of the conveyor and be filled with a certain amount of product. P1 starts the conveyor again




Inverse relay

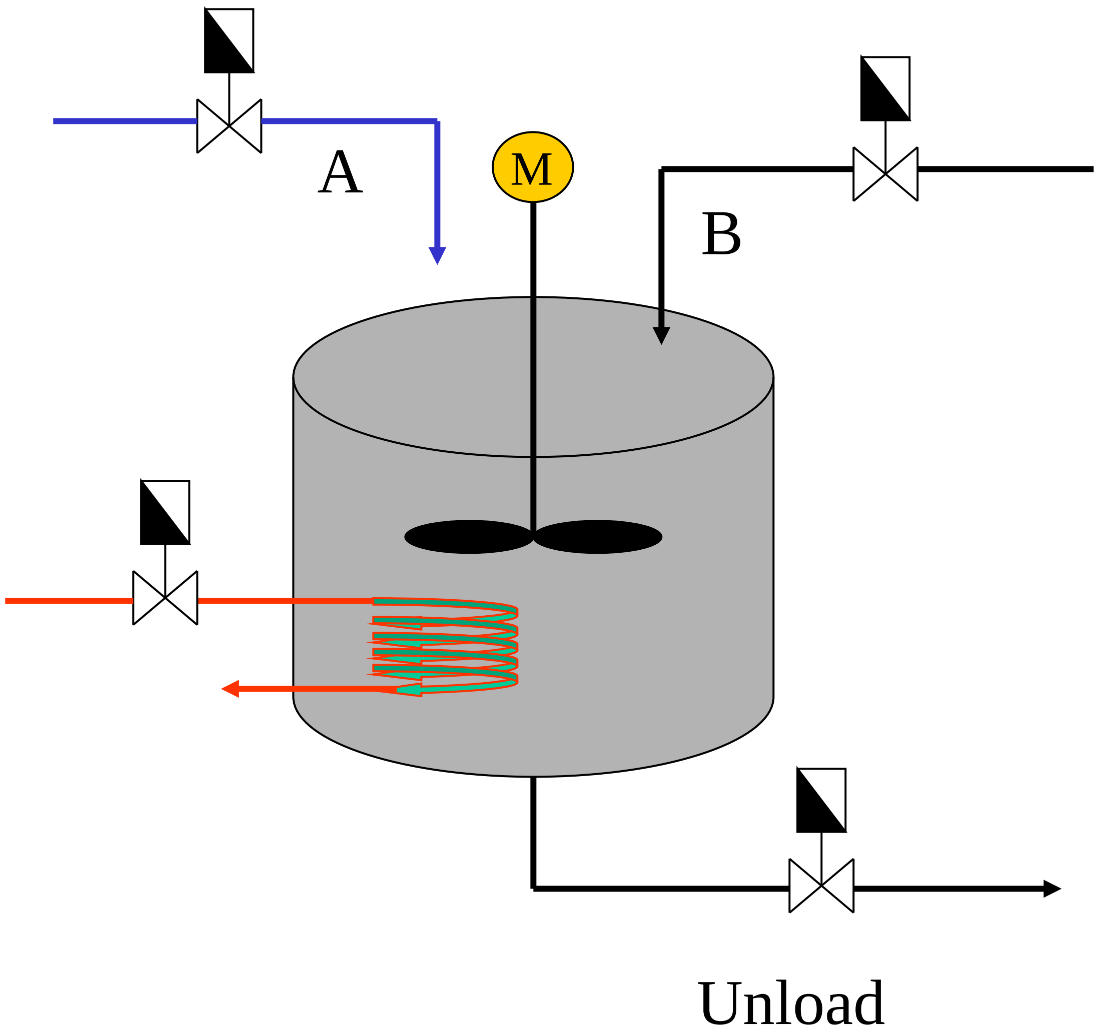
# Example: Safety system




The steam input is stopped if the temperature or the level are out of limits

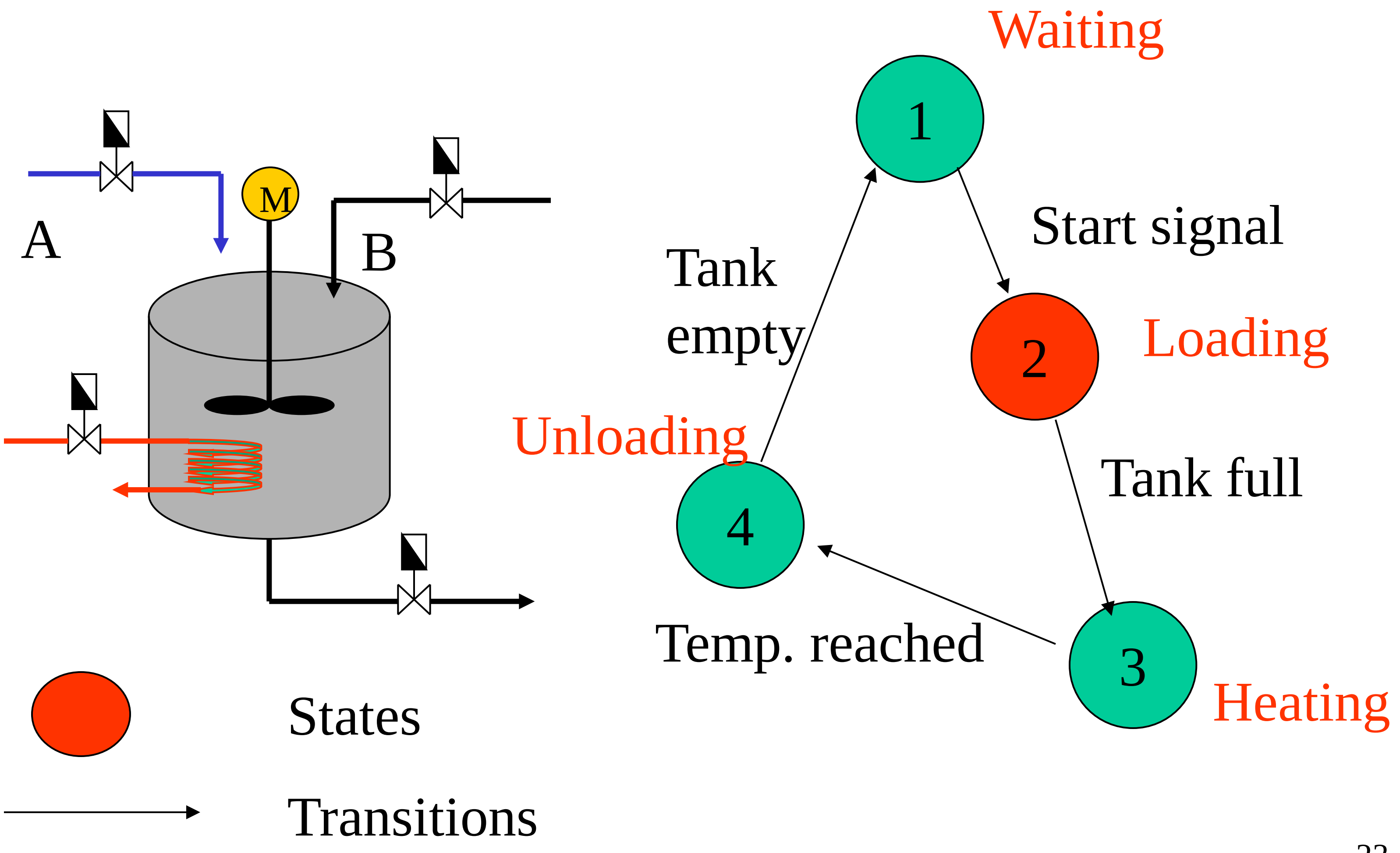
# Batch processes /Sequential systems



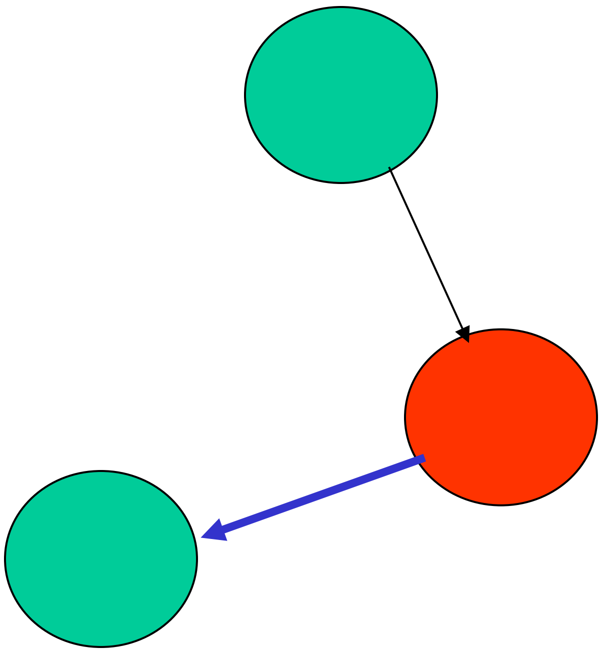

Sequence of stages with specific actions associated and transition conditions between them:

1 Waiting
2 Loading
3 Heating
4 Unloading

# State transition graphs



# State transition graphs



Transitions between states are formulated as logic functions of the system variables or time. A transition can be activated when the system is in the previous state and the logic condition is true. Each state have a set of associated actions.

# Synchronous and asynchronous processes

- Synchronous: State changes take place only at precise time instants marked by the pulses of a clock
- Asynchronous: State changes take place at any time as a function of the values of its input variables.

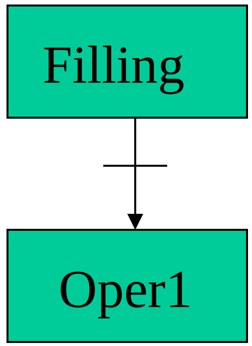
# SFC / Grafcet

- ✓ SFC Sequential Function Chart
- ✓ Graphical description of a sequential system
- ✓ Predecessor: Petri Nets
- ✓ They can be used at different levels
- ✓ Stages, transitions, actions

# SFC

Stage

Transition



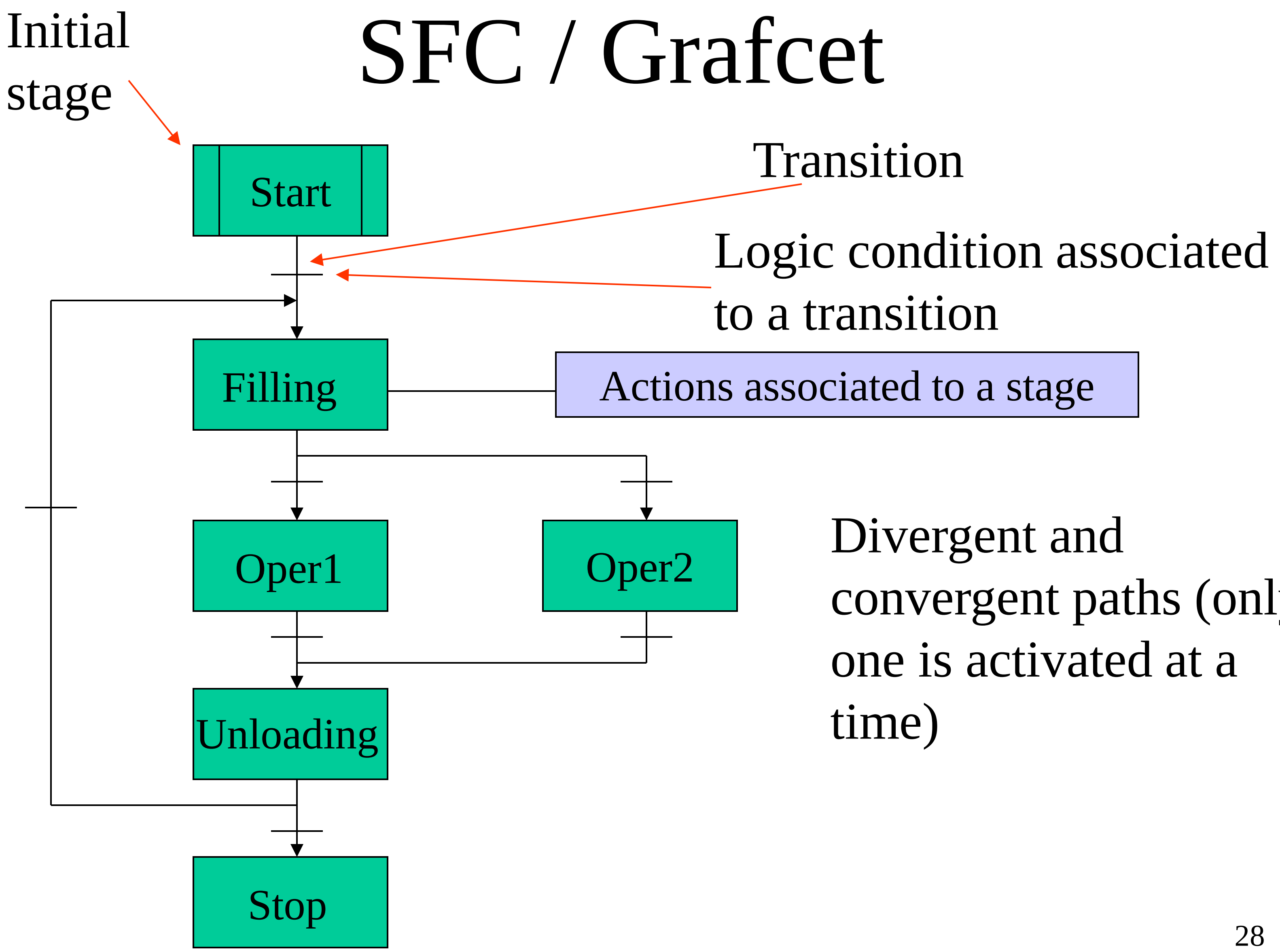

When the logic condition associated to a transition becomes true (and the process is in the corresponding stage), the current stage is deactivated and the following one is activated, besides executing all its associated actions. The logic condition can be formulated in any of the IEC 61131 languages

Variables associated by default to a stage:

stage.X = 1 if the process is in this stage, 0 if not

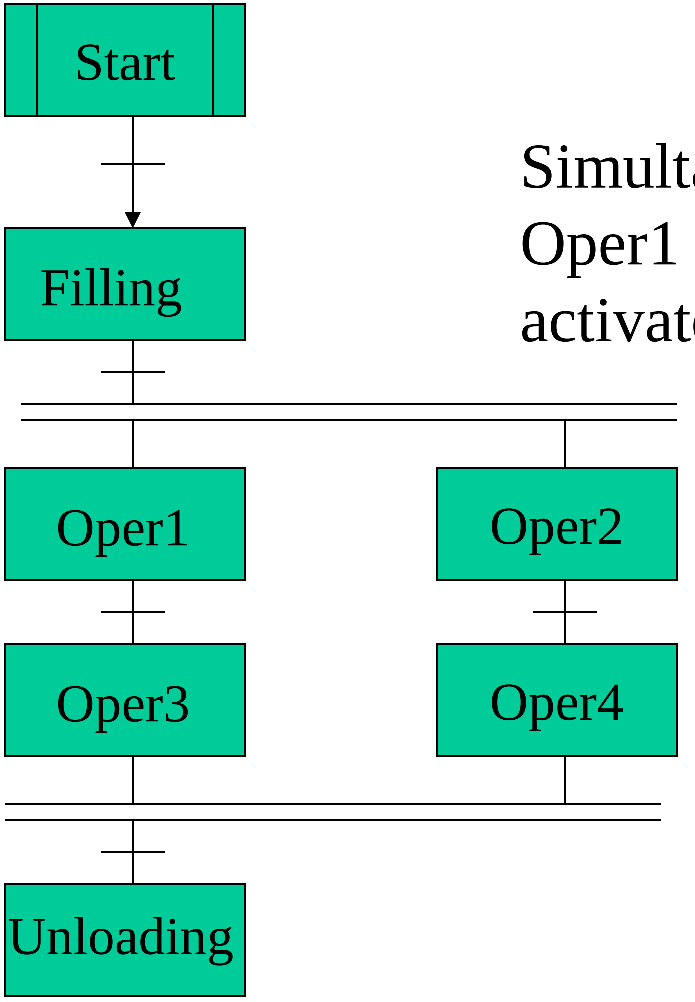
stage.T = elapsed time from the moment the stage was activated

# SFC / Grafcet




Divergent and convergent paths (only one is activated at a time)

# Parallel sequences




Simultaneous divergence: Oper1 and Oper2 are activated in parallel

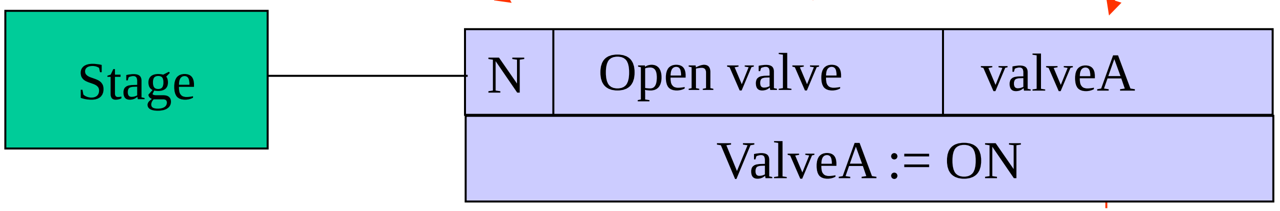
Simultaneous convergence: Only when both, Oper3 and Oper4, are active and the transition condition is fulfilled, the Unloading stage is activated.

# Actions

Action: Single name describing the action, either in the SFC or using an IEC language

Qualification: It describes when the action will take place

Optional associated variable

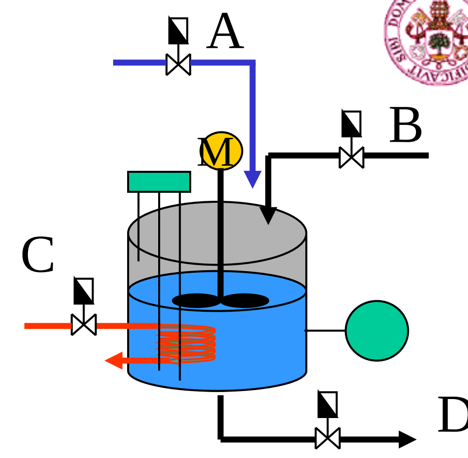


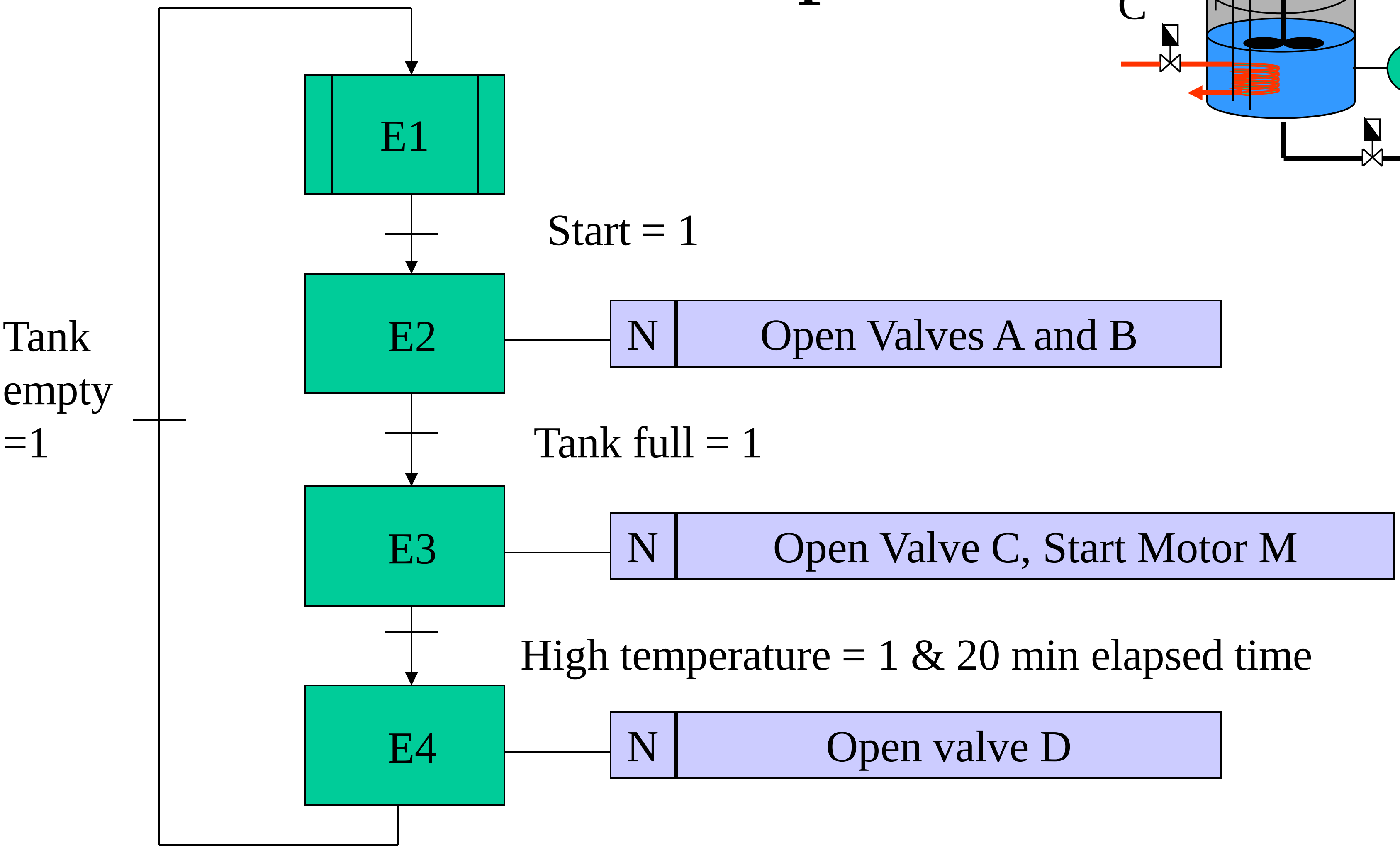
# Actions

| | | |
|---|---|---|
| N | Open valve | valveA |

| | |
|---|---|
| N | The action is executed while the stage is active |
| S | The execution of the action continues until a reset is activated |
| R | reset of a previous action |
| D x | The action is executed x sec. after the stage is activated and while it remains active |
| L | The action is executed only once when the stage is activated |

# Example


A
B
M
C
D


E1
Start = 1
E2
N
Open Valves A and B
Tank full = 1
E3
N
Open Valve C, Start Motor M
High temperature = 1 & 20 min elapsed time
E4
N
Open valve D
Tank
empty
=1

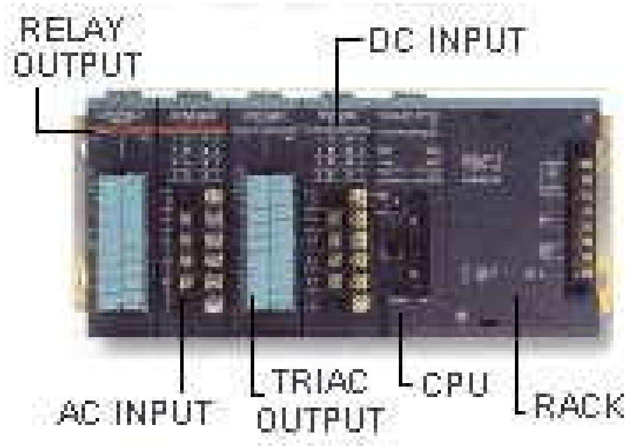
# Programmable Logic Controllers (PLC) (Autómatas programables)

Computerized devices that implement combinational and sequential functions connected to a process.

Late 1960's
Modicon

(High end PLC with many more functions)

- CPU
- Communications
- I/O cards
- Power supply

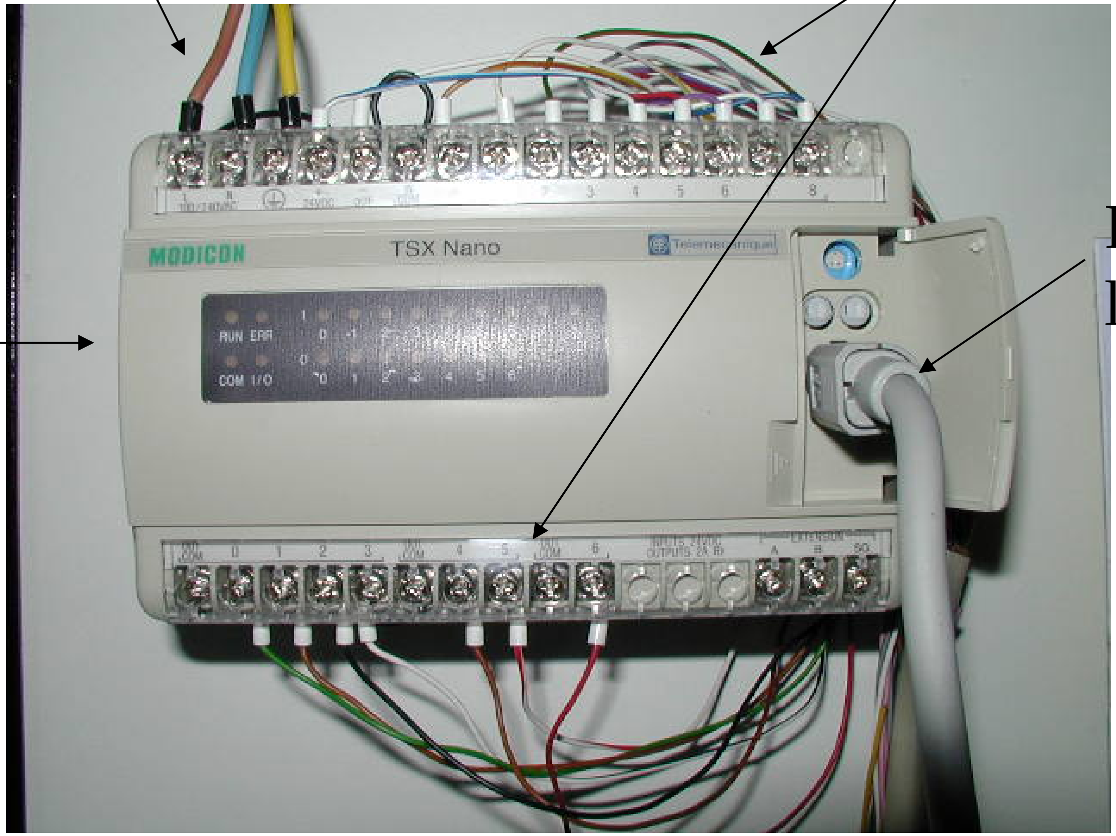



http://www.plcs.net/contents.shtml

# TSX Nano (Modicon)

Power supply

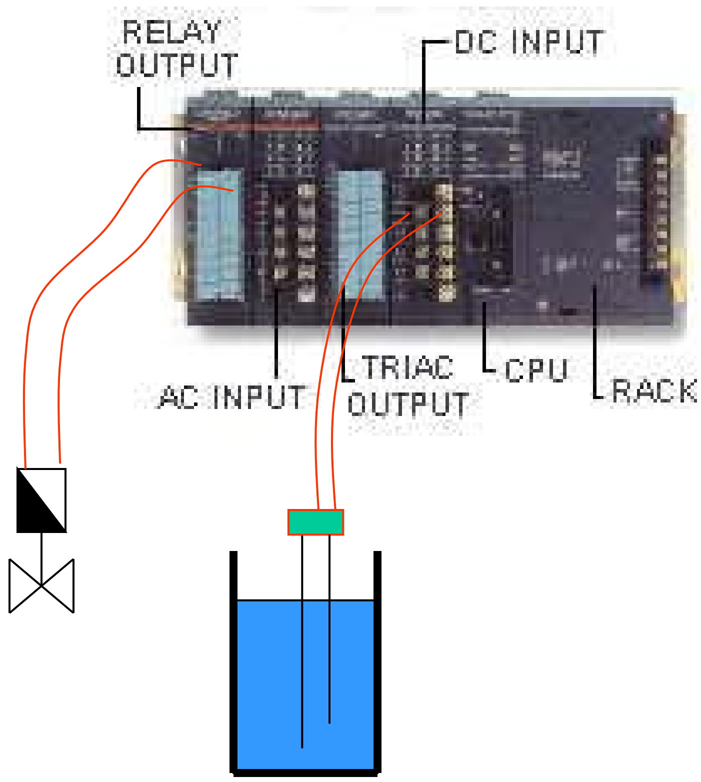
I/O signals

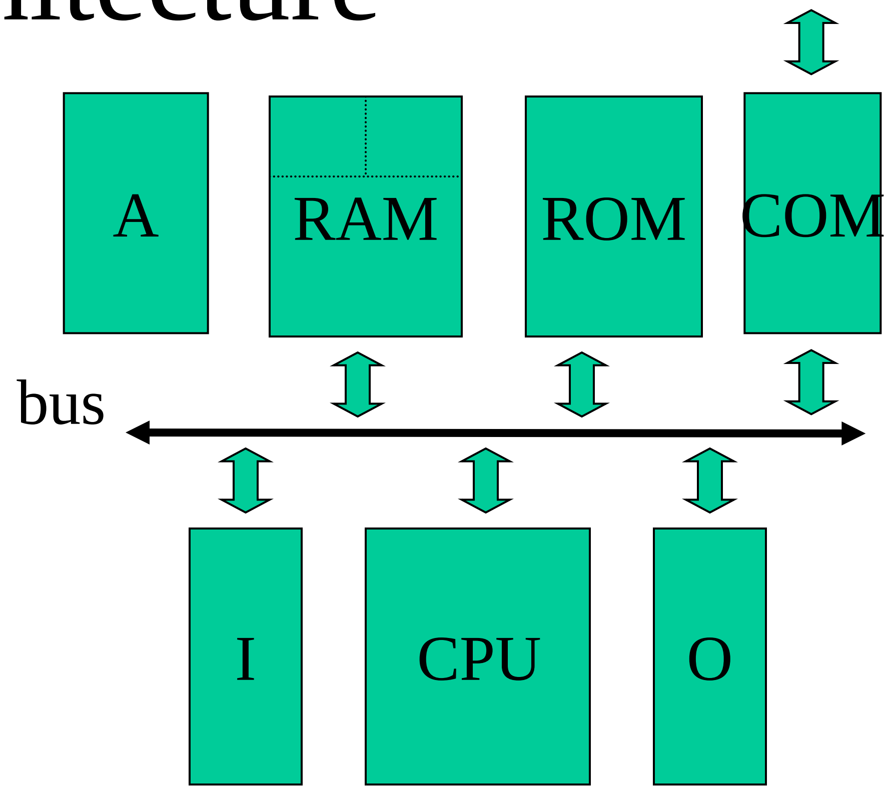



Display

RS232 link

# PLC Architecture



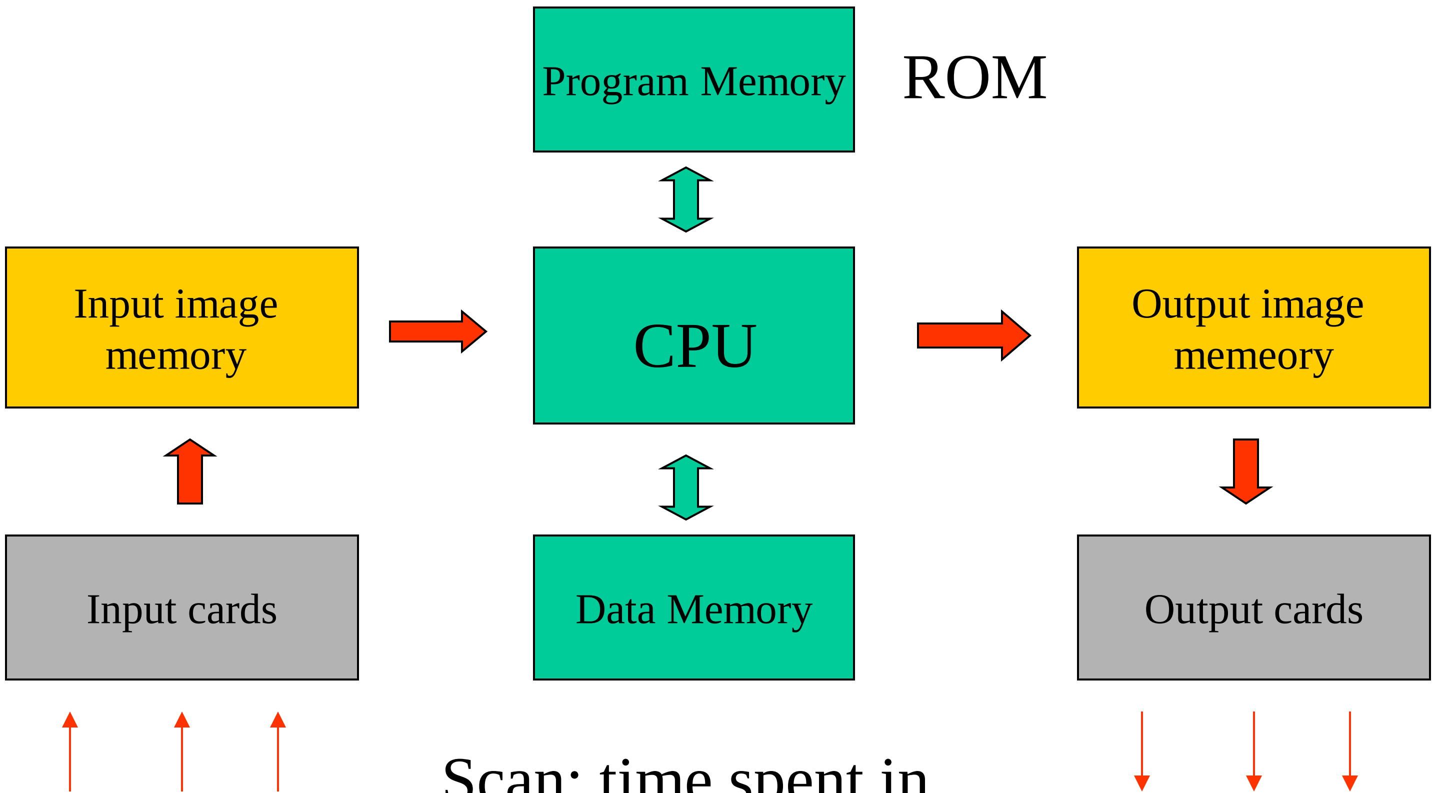

A: Power supply and battery

Different types of I/O cards

# I / O cards

- Input cards contain input relays, transistors, etc. (contacts) connected to the external world: sensors, switches, etc. that receive the fields signals and convert them to 0/1 values in the PLC memory.
- Output cards contain output relays, triacs, transistors, etc. (coils) connected to the external world: solenoids, lights,, etc. They send to them on/off signals according to the 0/1 values in the PLC memory.
- The PLC software contains virtual relays, counters, etc., used to implement the required logic and sequential functions.

# PLC Operation / Scan cycle



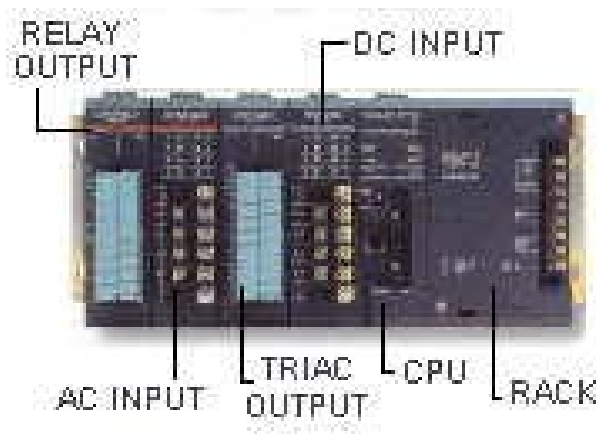

Sensors

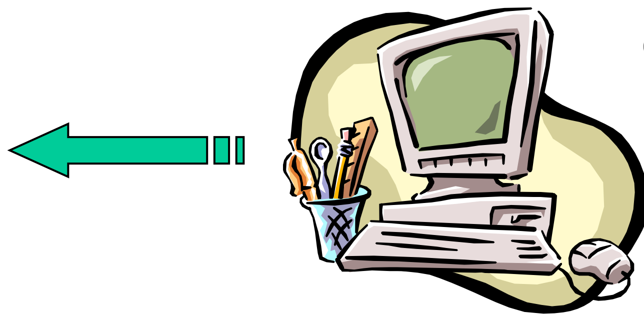
Actuators

Scan: time spent in executing a cycle
Registers: sets of bits for information storage

# Programming

PC ó programming console



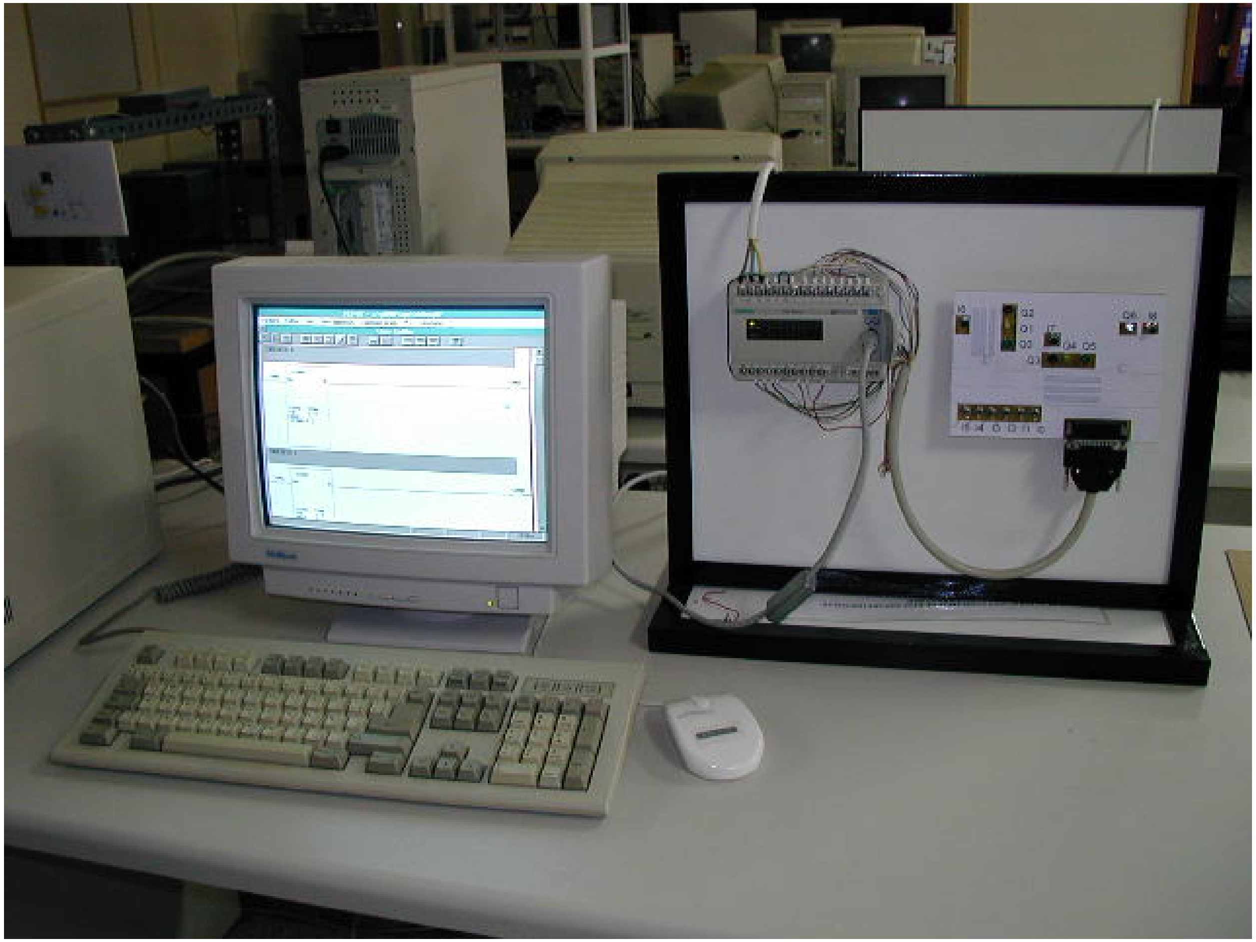
Firmware + configuration

Transfer to the PLC by RS-232 or network link

The program can be executed in different ways: cyclic operation, at a given time, when an event takes place, etc.

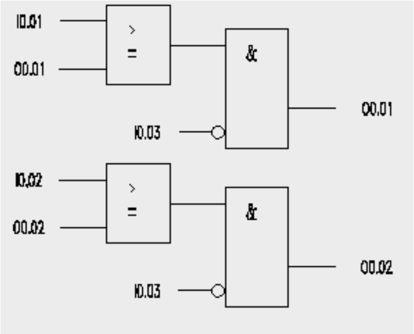
The PLC operation can be supervised from a PC

# PC + PLC



# IEC 61131-3 Norm

- ✓ Sequential Function Chart (SFC) (Grafcet) structures the internal organization of a program. Four interoperable programming languages:
- ✓ Structured Text (ST) ~ Pascal
- ✓ Function Block Diagram (FBD)
- ✓ Ladder Diagram (LD)
- ✓ Instruction List (IL)



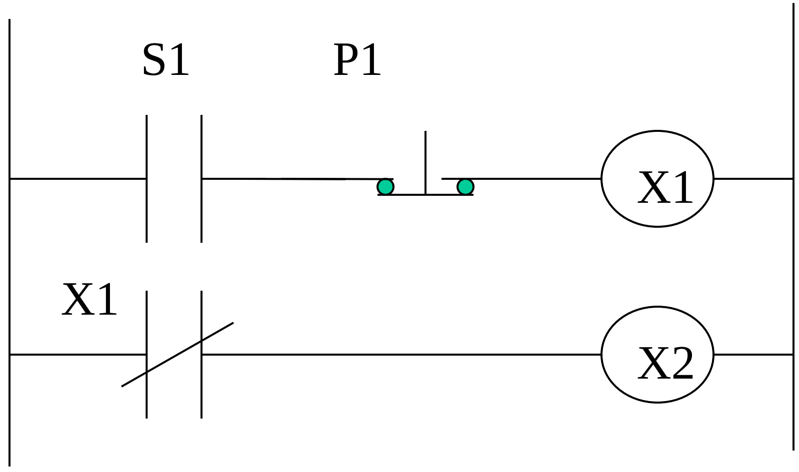

```
000  LD    %I0.1   Bp. inicio ciclo
     AND   %I0.0   Dp. presencia vehículo
     AND   %M3     Bit autorización reloj calendario
     AND   %I0.5   Fc. alto rodillo
     AND   %I0.4   Fc. detrás pórtico
005  S     %M0     Memo inicio ciclo
     LD    %M2
     AND   %I0.5
     OR    %I0.2   Bp. parada ciclo
     R     %M0
010  LD    %M0
     ST    %Q0.0   Piloto ciclo
```

http://www.plcopen.org/

# Ladder Diagrams

- Graphic programming
- It tries to imitate the electrical circuit diagrams with relays, timers, etc. used by electricians in the past.
- The steps are executed sequentially from top to bottom, from left to right

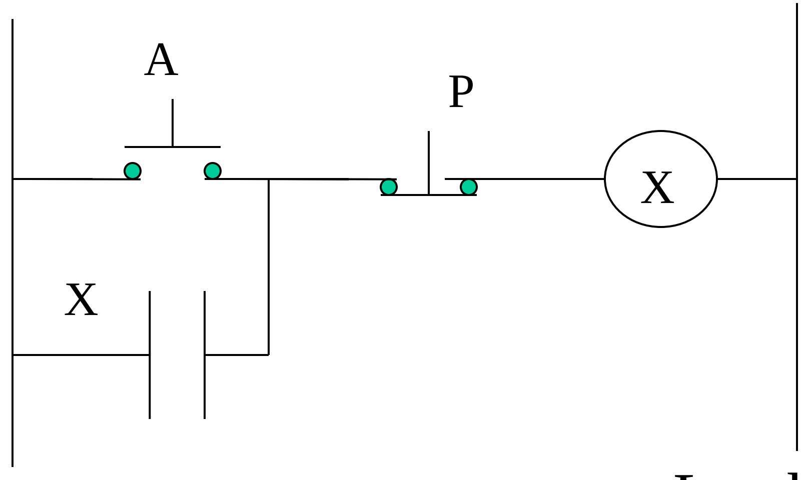
Ladder

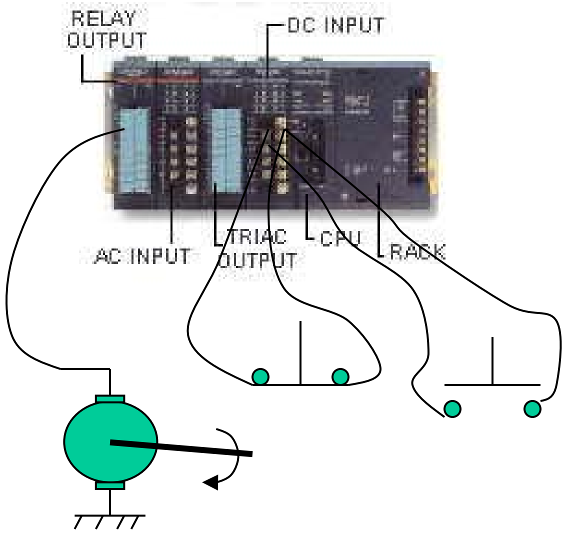



rungs

# Self-maintenance

Starting and stopping a motor
with two switches



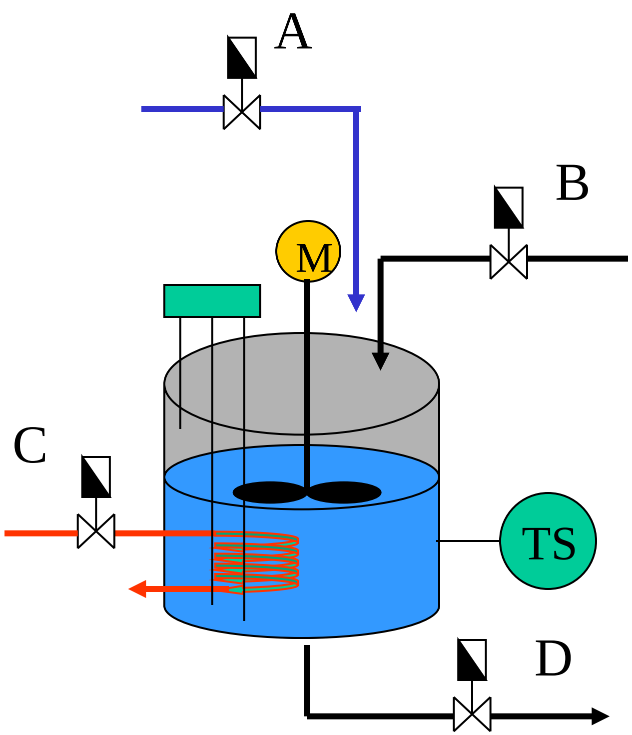




Also: SET RES Latching instructions

# Programming sequential systems with Ladder diagrams

✓ Three groups of rungs:
- Rungs to activate stages
- Rungs to activate transitions between stages
- Rungs to activate actions associated to each stage

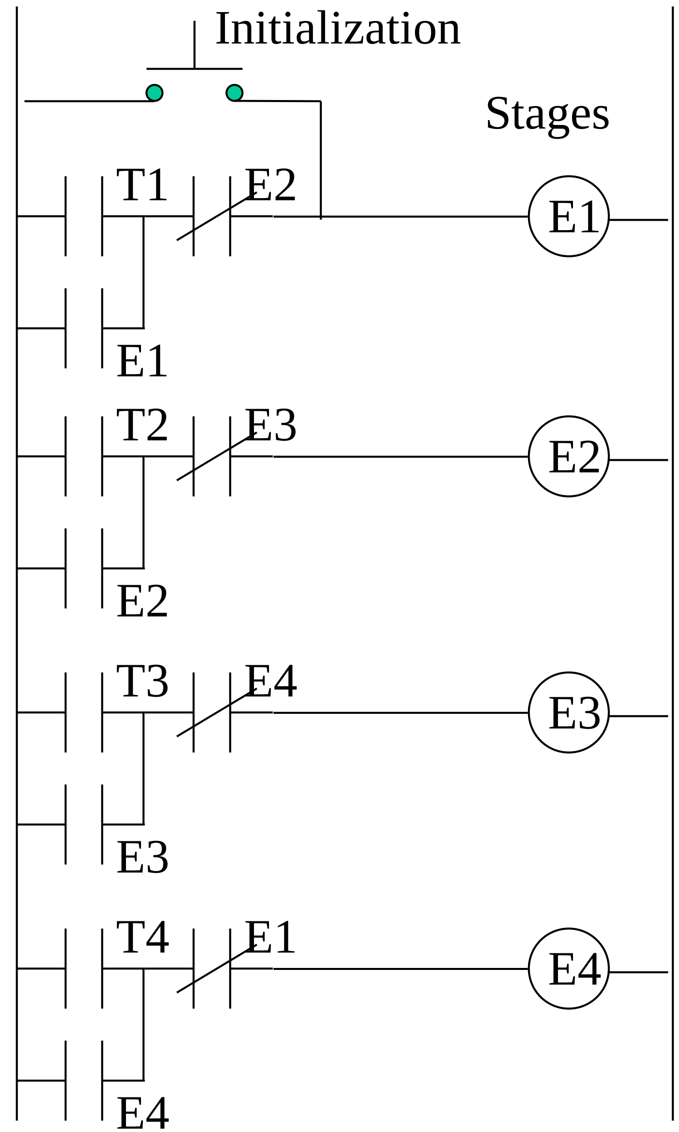
# Example: Stages



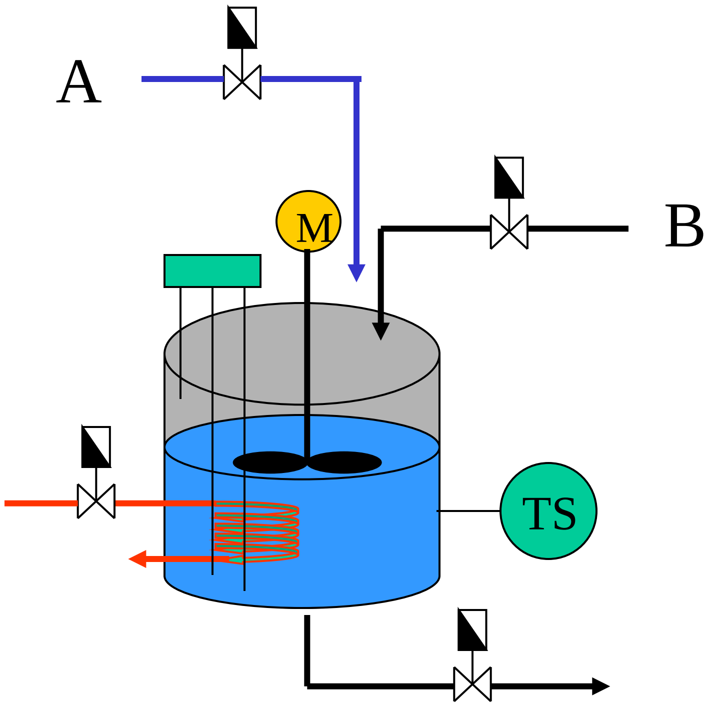

E Stage

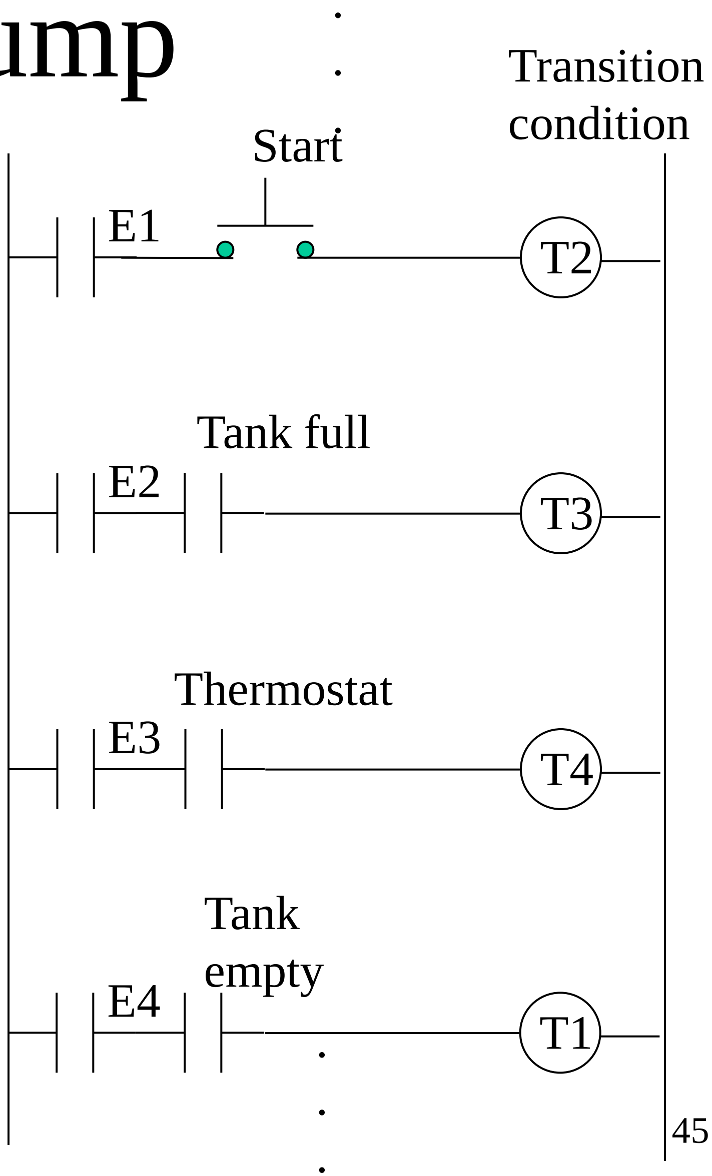
T Transition crossing condition

Initialization



# Stage jump

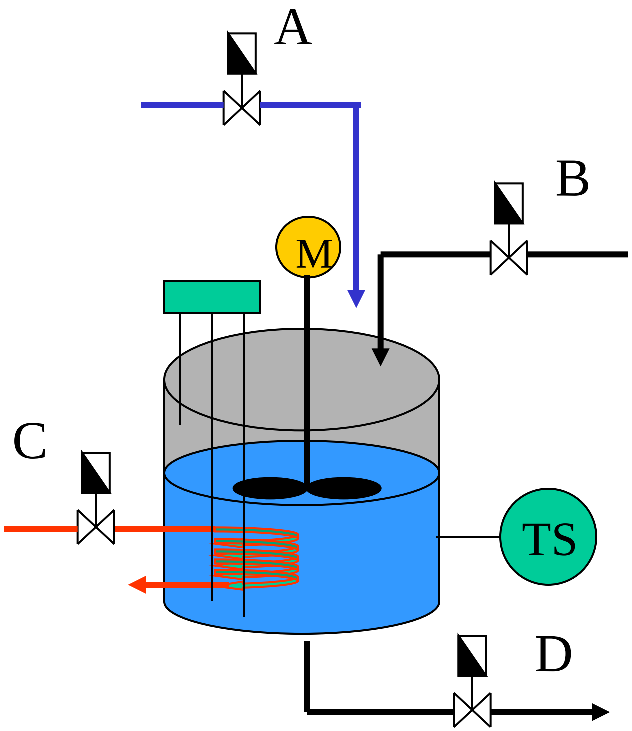


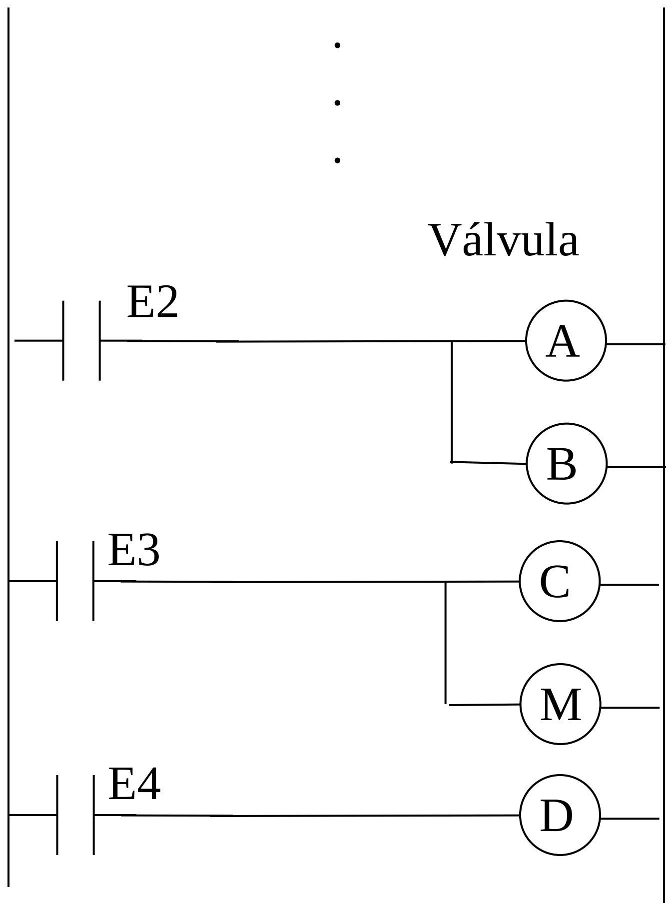




E  Stage

T  Transition condition

# Actions

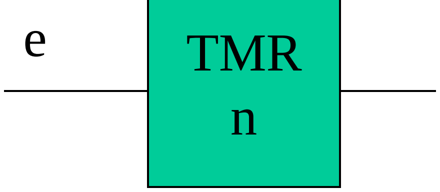


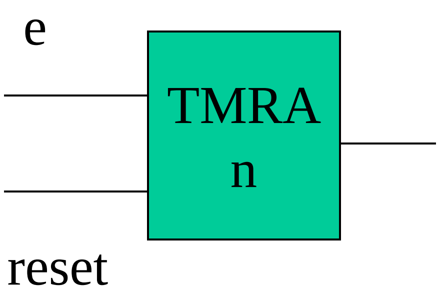

E  Stage

A,B,C,D    Electrovalves

# Timers

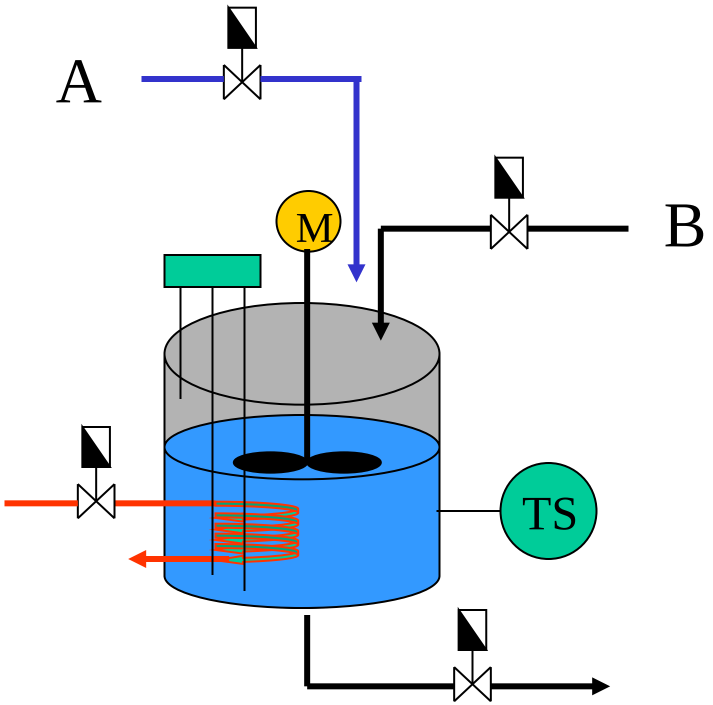



**Ton, Tof** Timer on/off Delay. The output is activated n time units after the input is activated. The timer is reset if the input does not keep active for n seconds

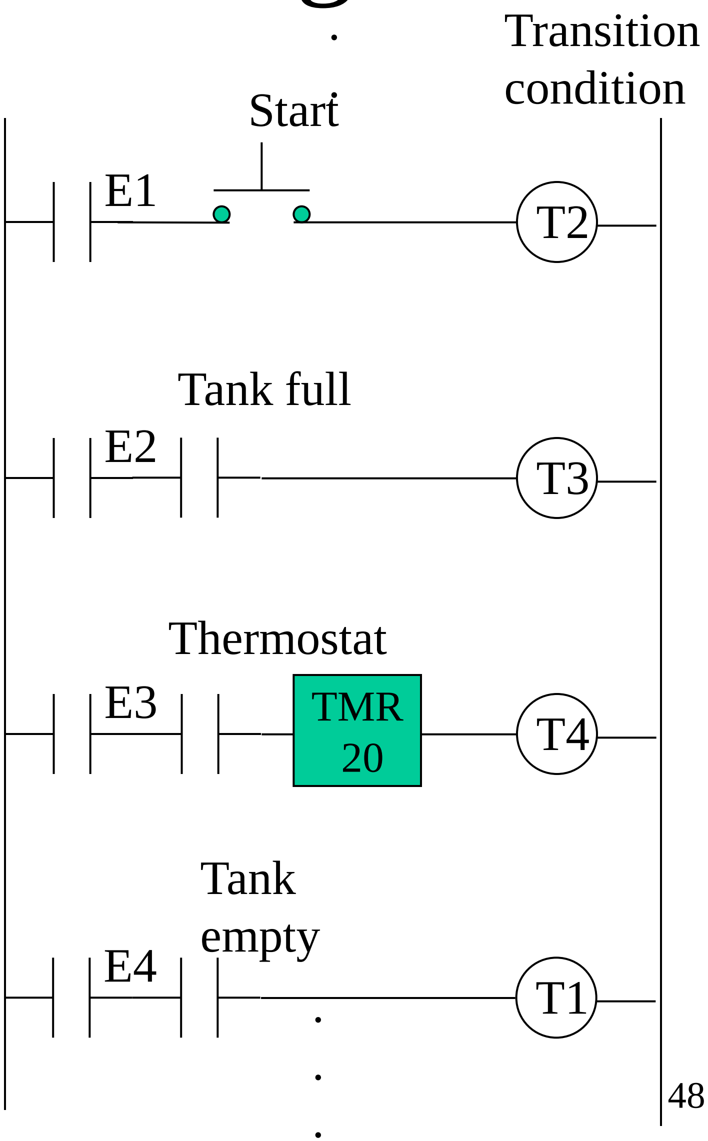



The output is activated n time units after the input is activated. If e is deactivated before n, the timer keep the accumulated time. The timer is reset only if the reset signal is activated

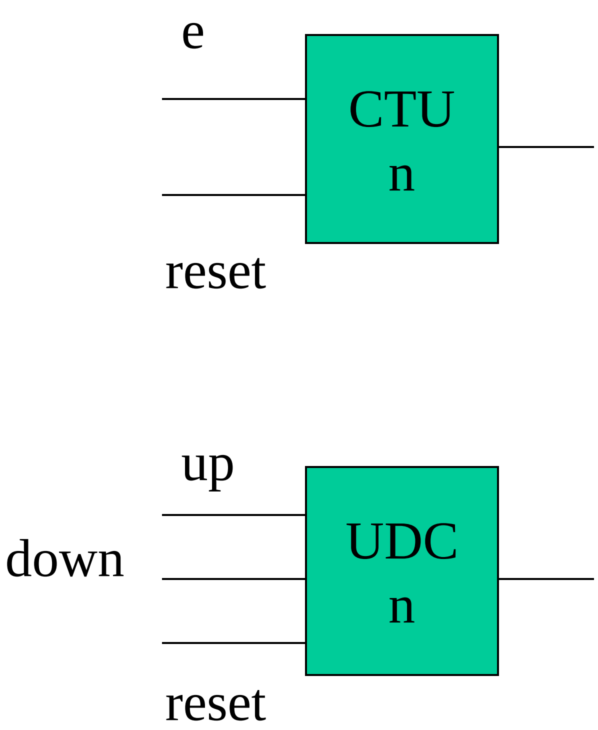
# A small change in stage 3




After reaching the required temperature, one wish to maintain the operation for 20sec.



# Counters

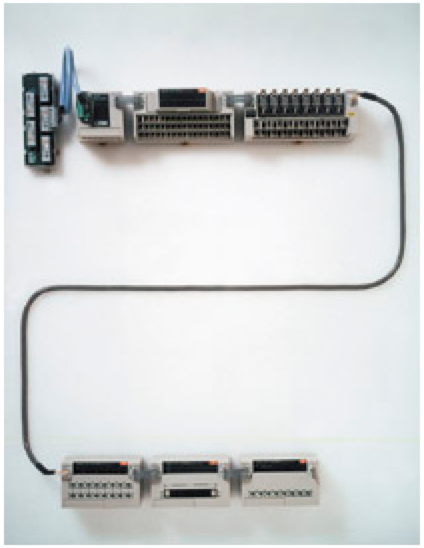



The output is activated when the input changes from false to true n times. The counter is reset to zero when the reset input is activated.

# IEC SFC

- ✓ SFC Sequential Function Chart
- ✓ Graphical description of a sequential system
- ✓ Predecessor: Petri Nets
- ✓ Very similar to Grafcet
- ✓ It can be used at different levels
- ✓ Stages, transitions, actions

# PLC networks /buses



**ASI**
**BITBUS**
**MODBUS**
**UNITELWAY**
**OPC**
**…..**

Among PLCs

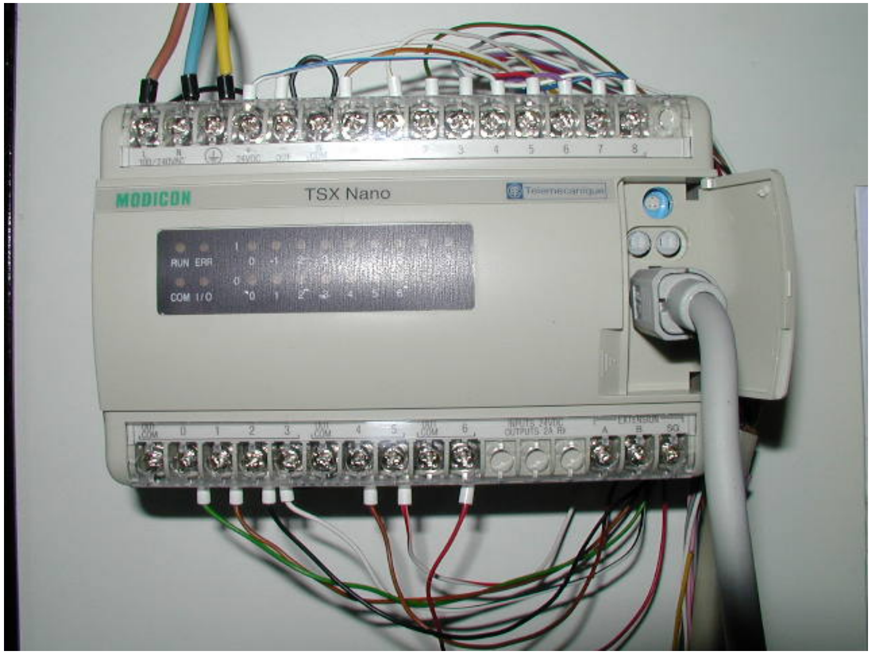
With the instrumentation

# TSX Nano

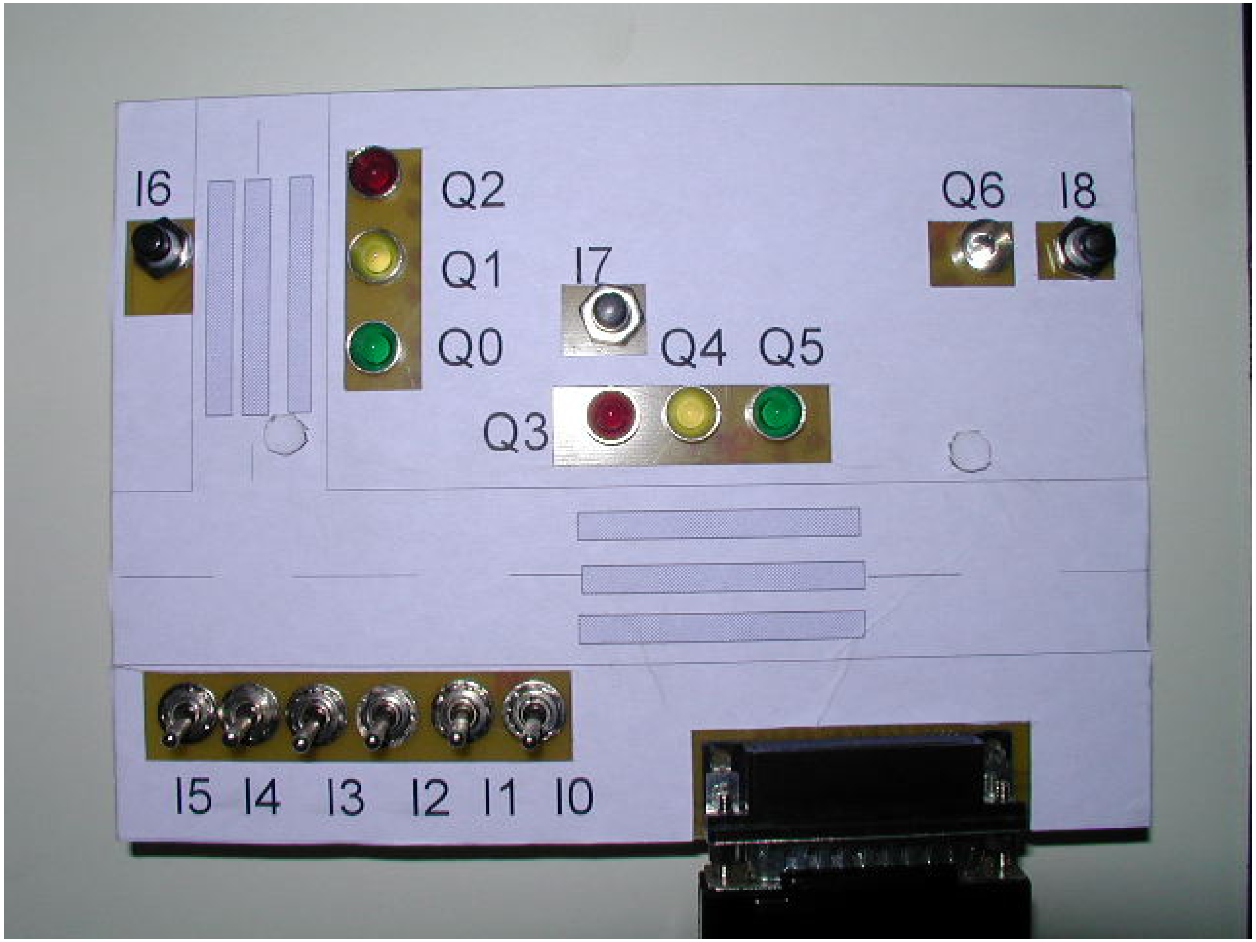
**<u>Number od inputs</u>**: 9 (%I0.0 to %I0.8). (positive logic)
**<u>Number of outputs</u>**: 7 (%Q0.0 to %Q0.6), relays.

Links/Operating modes:

- Stand alone
- Up to 1 Input/output extension.
- Up to 3 Automaton extension.



# Traffic lights



# Batch process

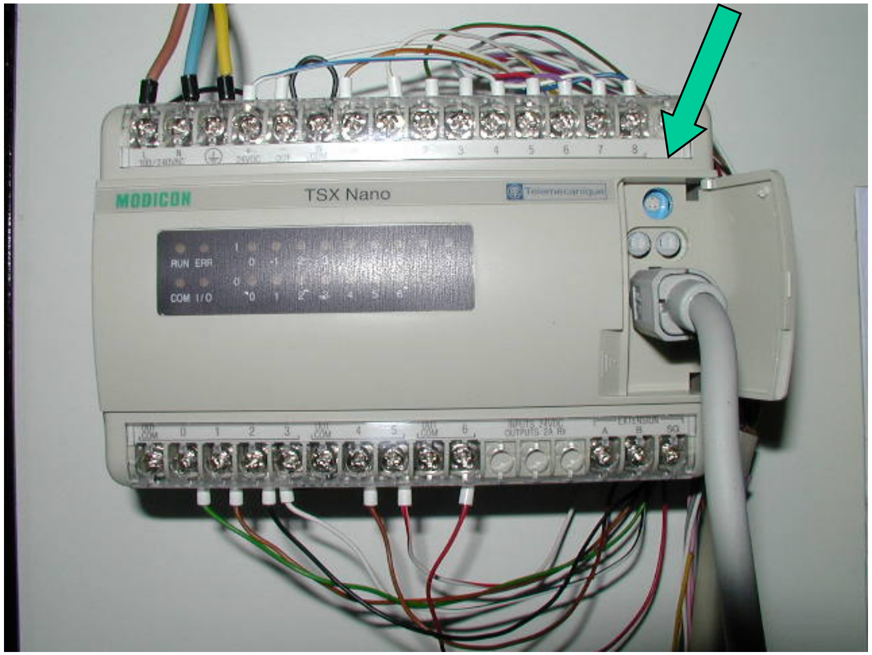
# TSX Nano

Each automata have a selector to choose an operating mode:

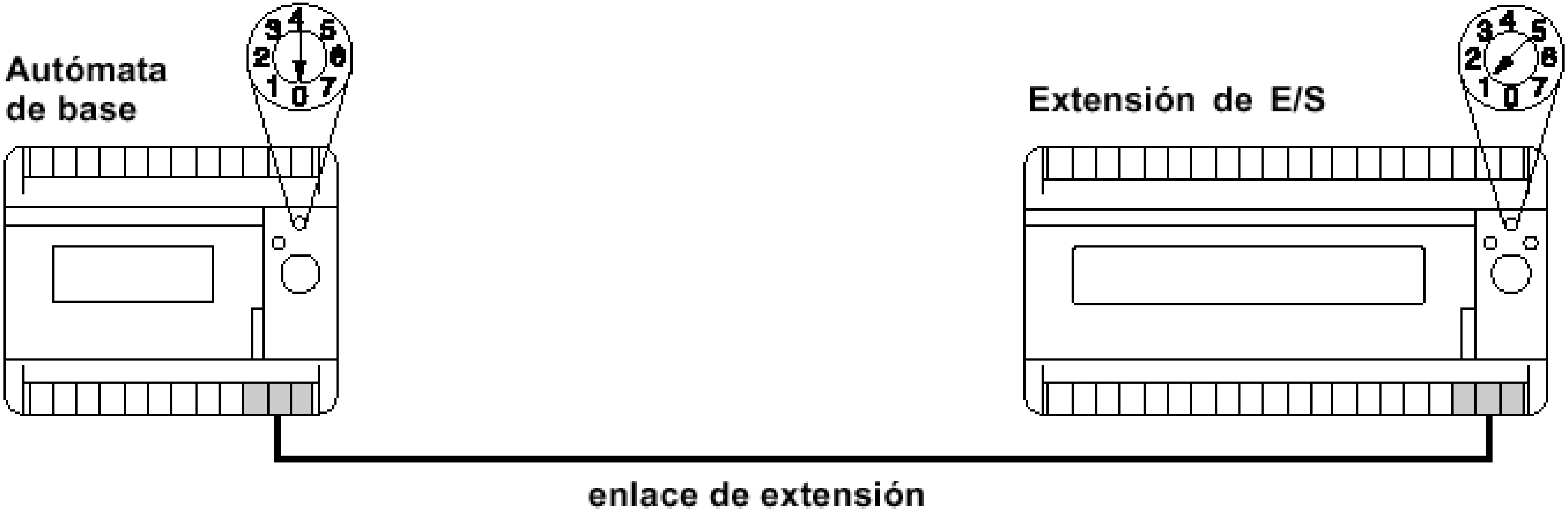
Switch to 0: Master.

Switch to 1: Input/output extension of the master.

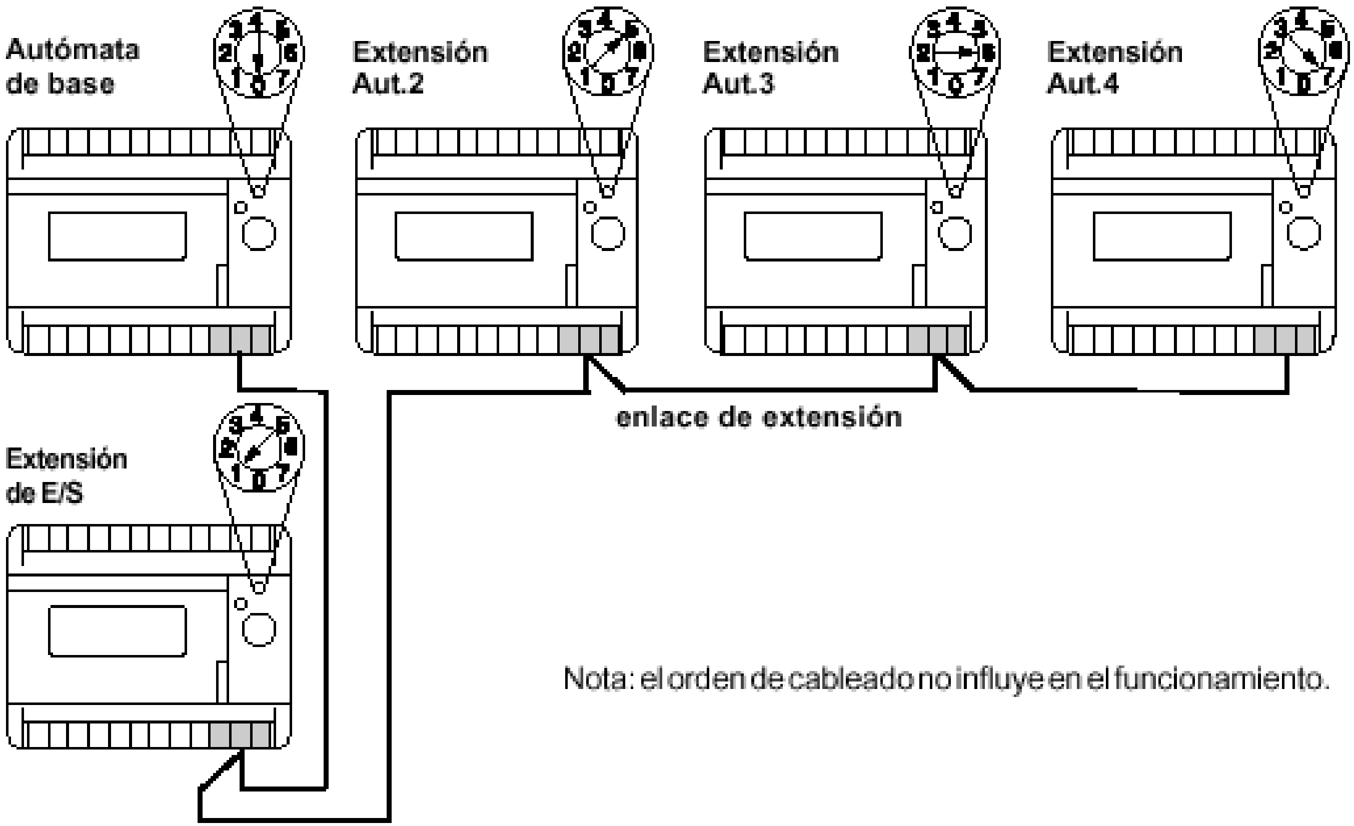
Switch to 5, 6 y 7: The PLC works as an extension of the master



# I/O Extension



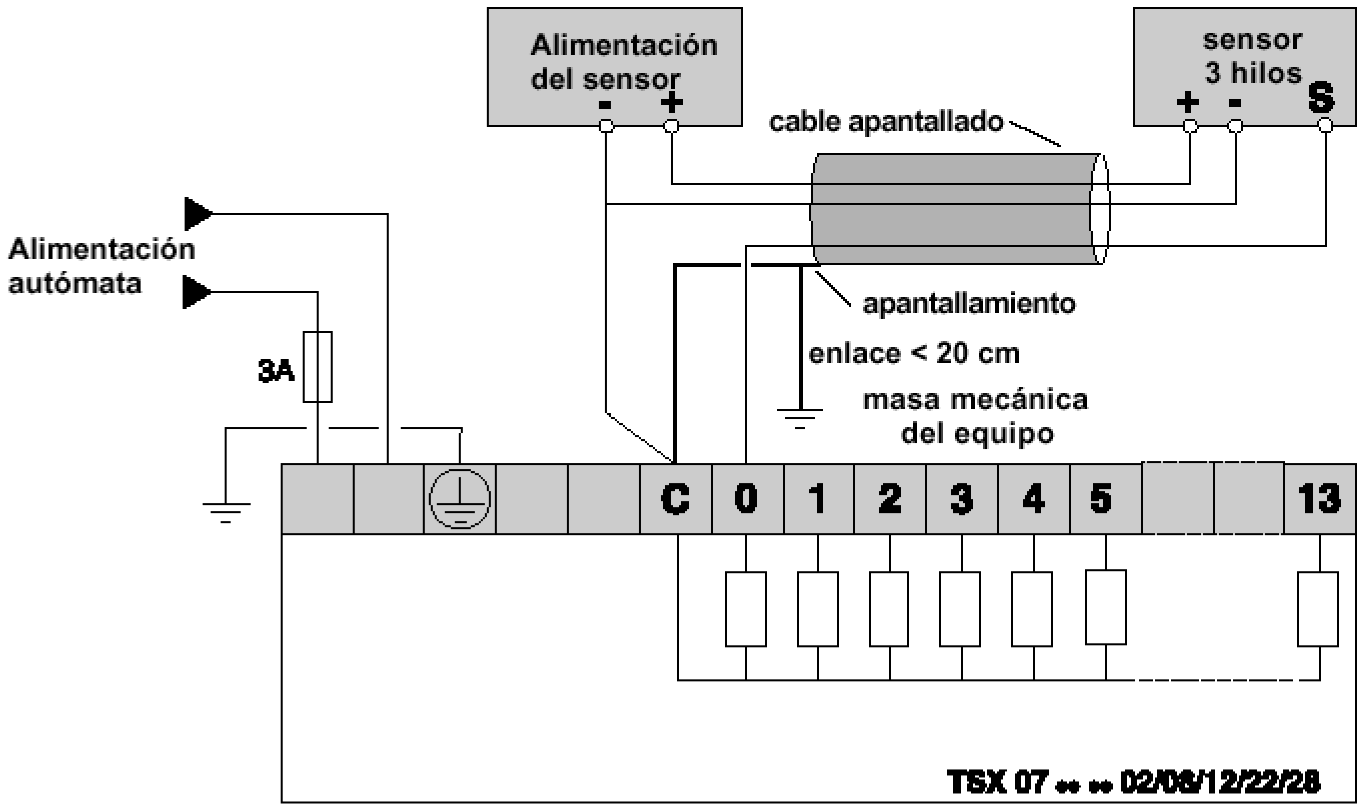
# PLC and I/O extension



# I/O naming

| % | I o Q | 0 ó 1 | • | i |
|---|---|---|---|---|
| símbolo | **I** = entrada<br>**Q** = salida | **0** = autómata de base o extensión autómata<br>**1** = extensión de las entradas/ salidas | punto | **i** = número de vía (véase la siguien-te tabla) |

# Wiring



Alimentación del sensor
- +
sensor 3 hilos
+ - S
cable apantallado
Alimentación autómata
3A
apantallamiento
enlace < 20 cm
masa mecánica del equipo
C 0 1 2 3 4 5 13
TSX 07 •• •• 02/06/12/22/28

# Names of variables

| Tipo | Dirección (o valor) | Número máximo | Acceso en escritura(1) | Ver apart. |
|---|---|---|---|---|
| Valor inmediato | 0 ó 1 | - | - | - |
| Bits de entrada<br>de salida | %I0.i o %I1.i (2)<br>%Q0.i o %Q1.i (2) | 28<br>20 | no<br>sí | 1.5<br>Sec.A |
| Bits internos | %Mi | 128 (3) | sí | |
| Bits de sistema | %Si | 128 | según i | 5.1 |
| Bits de etapa Grafcet | %Xi | 62 | sí | 2.3-1 |
| Bits de bloques función | %TMi.Q %DRi.F..... | | no (4) | 2.2-1 |
| Bits bloques función reversible | E,D,F,Q,TH0,TH1 | | no | 3.3-1 |
| Bits extraídos palabr. | | | | 3.1-1 |

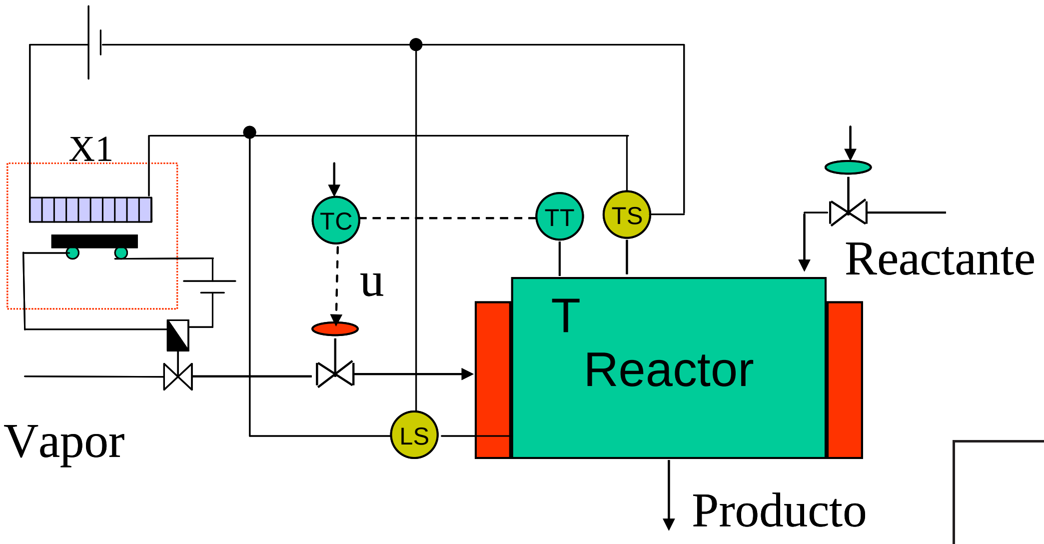
# Functional blocks

| Bloques función estándares | Palabras y bits asociados | | Dirección | Acceso escritura | Ver Ap. |
|---|---|---|---|---|---|
| Temporizador %TMi (i = 0 a 31) | Palabra | Valor actual | %TMi.V | no | 2.2-3 |
| | | Valor de preselección | %TMi.P | sí | |
| | Bit | Salida temporizador | %TMi.Q | no | |
| Contador/ descontador %Ci (i = 0 a 15) | Palabra | Valor actual | %Ci.V | no | 2.2-4 |
| | | Valor de preselección | %Ci.P | sí | |
| | Bit | Salida desbordam.(vacío) | %Ci.E | no | |
| | | Salida preselec. alcanzada | %Ci.D | no | |
| | | Salida desbordam. (lleno) | %Ci.F | no | |
| Registro palabra %Ri (i = 0 a 3) | Palabra | Acceso al registro | %Ri.I | sí | 2.2-5 |
| | | Salida del registro | %Ri.O | sí | |
| | Bit | Salida del registro lleno | %Ri.F | no | |
| | | Salida del registro vacío | %Ri.E | no | |
| Programador cíclico %DRi (i = 0 a 3) | Palabra | Nº de paso en curso | %DRi.S | sí | 2.2-6 |
| | Bit | Último paso definido en curso | | %DRi.F | no |

# SIF SIS SIL

- They are systems oriented to guarantee safe operation of the process or a controlled shut-down if necessary.
- IEC 61508 (ISA S84.01), IEC61511 standards
- SIF Safety Instrumented Function (Set of actions that protect a process against a particular risk)
- SIS Safety Instrumented Systems (composed of several SIF)
- SIL Safety Integrity Level (1, 2, 3) (Level of protection of a SIF)
- The design of the control system of a process and its safety system must be performed jointly, but they must be implemented separately

# SIS



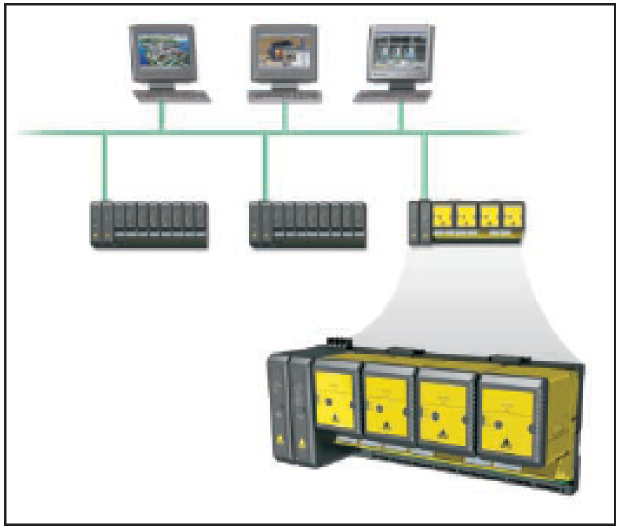

SIF are focused on preventing catastrophic incidents
Interlocking

1 Sensors (different from the ones of the control systems and with separate wiring)

2 Associated safety logic implemented in an independent PLC

3 Actuators

Key information: Mean Time Between Faults MTBF



Per IEC 61508, the DeltaV SIS system and the DeltaV system are separate, yet have integrated engineering software.

# SIS

¿ Qué hace un SIS ?
Boom!
Control del SIS
Acción Mecánica de Paro
Punto de Trip
Acción de Paro de Emergencia
ESD (Emergency Shutdown)
Acción del Operador
Alarma de Alto Nivel
Alto Nivel
Variable de proceso
Comportamiento normal
Bajo Nivel
Tiempo

# SIL of a SIF

Table 1: Risk Based on Frequency

| Risk level | Descriptor | Frequency of Occurrence |
|---|---|---|
| 5 | Frequent | One per year |
| 4 | Probable | One per 10 years |
| 3 | Occasional | One per 100 years |
| 2 | Remote | One per 1,000 years |
| 1 | Improbable | One per 10,000 years |

Table 2: Risk Levels Based on Severity

| Risk level | Descriptor | Potential consequences |
|---|---|---|
| 5 | Catastrophic | Multiple deaths |
| 4 | Severe | Death |
| 3 | Serious | Lost time accident |
| 2 | Minor | Medical treatment |
| 1 | Negligible | No injury |

Table 3: Safety Integrity Levels: Target Failure Measures

| SIL | Risk Reduction Factor | Average PFD |
|---|---|---|
| 1 | 10 to 100 | 0.1 to 0.01 |
| 2 | 100 to 1,000 | 0.01 to 0.001 |
| 3 | 1,000 to 10,000 | 0.001 to 0.0001 |

# Sistemas Instrumentados de Seguridad SIS -SIL