

Tema 7. Introducción a la Automatización Industrial

Automática

2º Curso del Grado en
Ingeniería en Tecnología Industrial



Copyright

Autor:

Juan Antonio García Fortes, 2013
Dpto. Ingeniería de Sistemas y Automática
Universidad de Málaga
jagarciaf@uma.es

Modificado y adaptado de:

victortorreslópez, 2011
Dpto. Ingeniería de Sistemas y Automática
Universidad de Málaga
vetorres@uma.es

Licencia:

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/> or send a letter to Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.

Tema 7.- Introducción a la automatización industrial

7.1. Concepto de automatización

7.2. Sistema automático de producción (SAP)

7.3. Sistemas de eventos discretos (DES)

7.4. Concepto de automatismo

7.5. Implantación de automatismos

7.6. El autómeta programable (PLC)

7.7. Lenguajes de programación de PLC IEC 61131-3

Definiciones

- **Automática:** ciencia que trata de sustituir en un proceso el operador humano por dispositivos mecánicos o electrónicos.
- **Automatización:** utilización de técnicas y equipos para que un sistema funcione de forma automática.

Ámbito de aplicación

- **Servicios:** semáforo, ascensor, puerta automática, máquina expendedora,...
- **Doméstico:** electrodomésticos, domótica,...
- **Industrial:**
 - Tareas:
 - Cortado
 - Empaquetado
 - Ensamblado
 - Procesos:
 - Plantas embotelladoras
 - Producción y control de energía
 - Sistemas de fabricación flexible

Concepto de automatización II

Ámbito de aplicación

-
-
-



- Sistemas de fabricación flexible

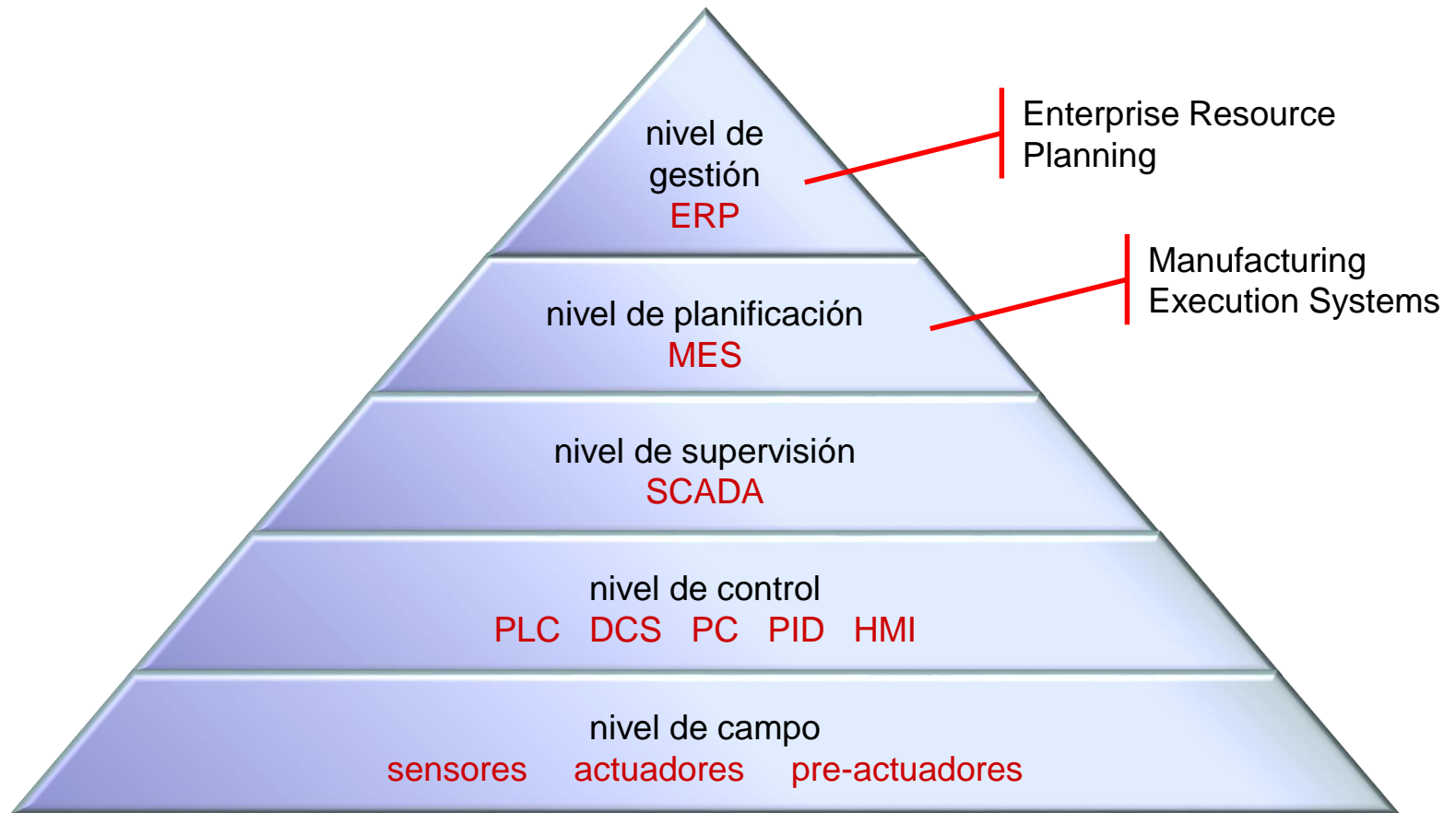
Beneficios

- Incrementa la producción
- Mejora la productividad
- Disminuye los costes de producción
- Reduce los tiempos de producción
- Mejora la calidad de los productos
- Reduce el stock y aumenta su rotación
- Mejora la seguridad
- Favorece la automatización integral

Límites

- La automatización es cara
- Dificultades técnicas
- Imposibilidad de rentabilizar la inversión
- Incremento de costes fijos
- Casi nunca es rentable sustituir completamente al operador humano

Pirámide de la automatización



Pirámide de la automatización II

- **Nivel de gestión:** optimización de los procesos empresariales, tales como producción, ventas, compras, nóminas, pedidos, inventarios, ...
- **Nivel de planificación:** optimización de los recursos de la planta para realizar la producción de la forma más eficiente posible.
- **Nivel de supervisión:** adquisición de datos y control de alto nivel de la planta.
- **Nivel de control:** control de la planta a bajo nivel.
- **Nivel de campo:** hardware de la planta.

Tema 7.- Introducción a la automatización industrial

7.1. Concepto de automatización

7.2. Sistema automático de producción (SAP)

7.3. Sistemas de eventos discretos (DES)

7.4. Concepto de automatismo

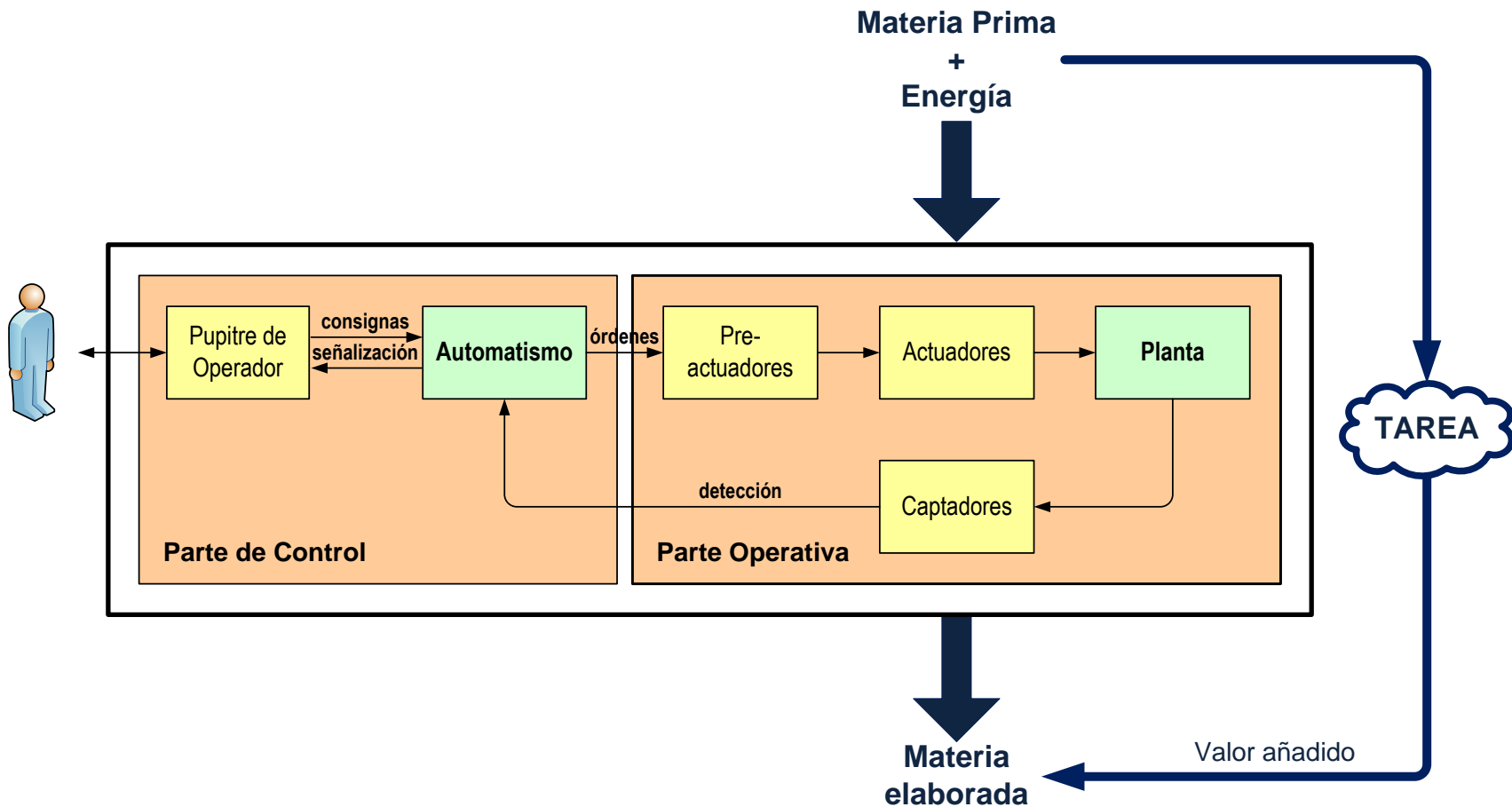
7.5. Implantación de automatismos

7.6. El autómeta programable (PLC)

7.7. Lenguajes de programación de PLC IEC 61131-3

Sistema automático de producción I

Esquema



Aspecto físico

- Un SAP es una máquina compuesta de:
 - Captadores
 - Actuadores
 - Pre-actuadores
 - Controladores
 - Interfaces Hombre Máquina (HMI)
 - Planta

Captadores



ultrasonido

distancia



codificador angular



final de carrera



fotoeléctrico

barrera



inductivo

detector metal



fibra óptica

termómetros

Actuadores



motor AC



pistón hidráulico



pinzas neumáticas



pistón neumático



pistón neumático

Pre-actuadores



relé



electroválvula neumática



contactor



electroválvula hidráulica



variador de frecuencia

Controladores



Typ FEC (Festo)



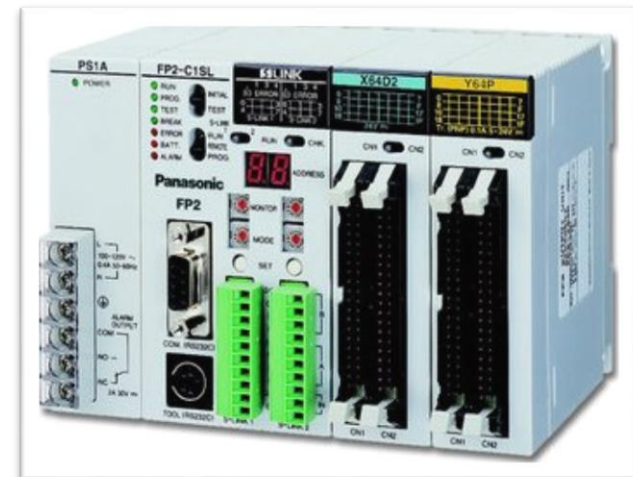
CX1010 (Beckhoff)



CPM1A (Omron)

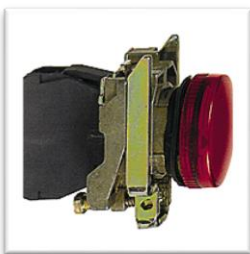


Logo (Siemens)



FP2 (Panasonic)

Interfaces Hombre Máquina (HMI)



elementos clásicos



pantalla táctil

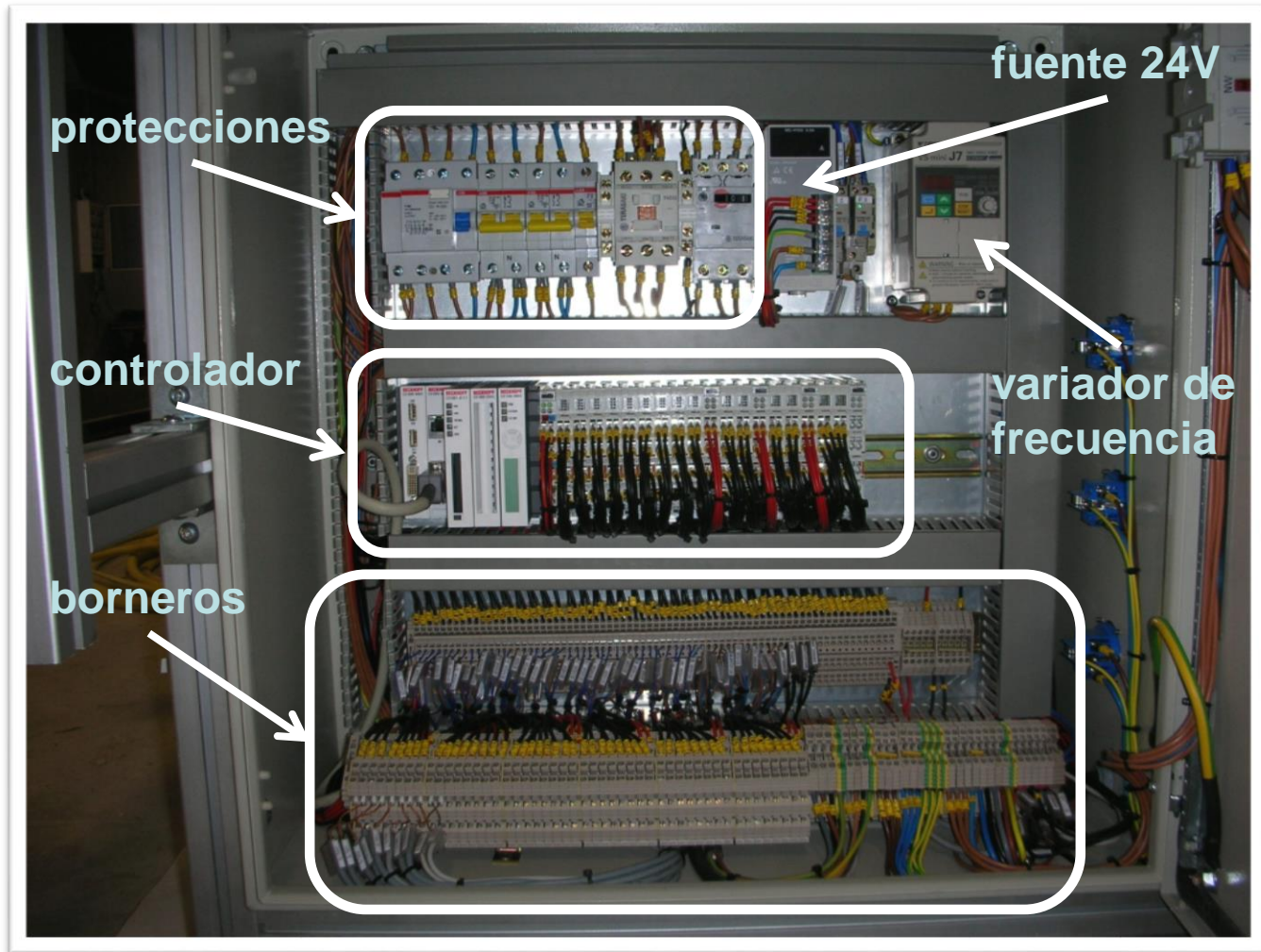


pantalla alfanumérica

Pupitres de operador

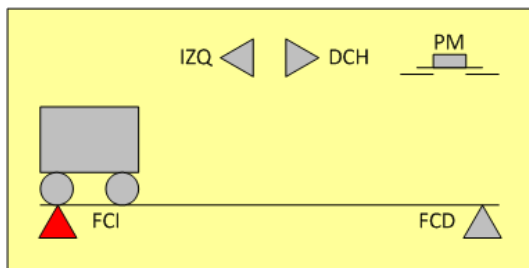


Cuadro de control



Sistema automático de producción X

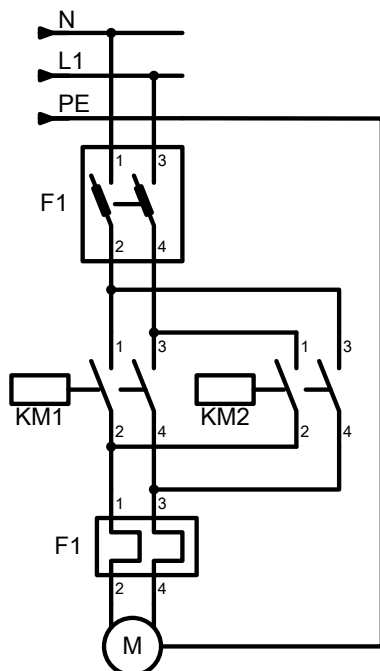
Circuitos de fuerza y mando



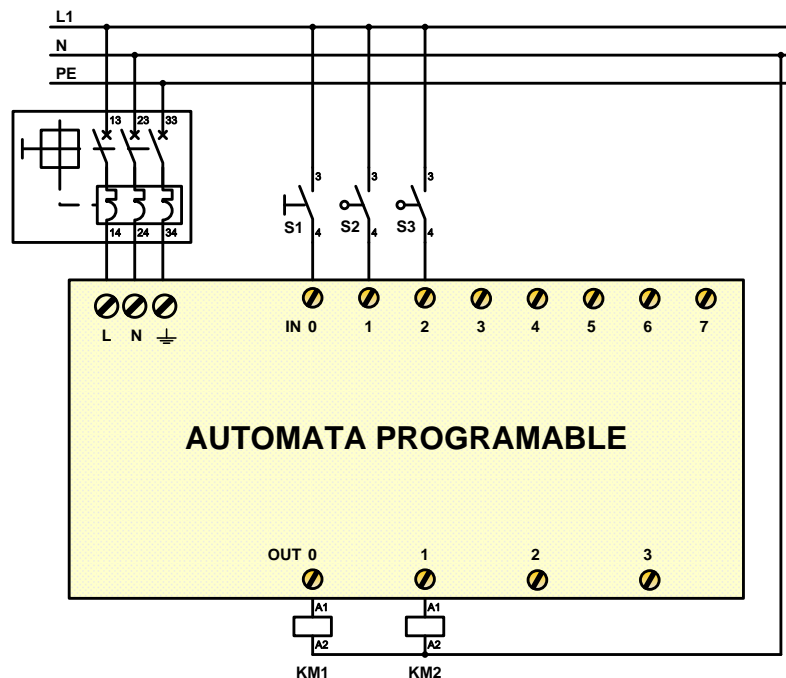
carro va y viene

Nombre	Símbolo	Descripción
PM	S1	pulsador de marcha
FCI	S2	final de carrera izquierda
FCD	S3	final de carrera derecha
IZQ	KM1	marcha hacia la izquierda
DCH	KM2	marcha hacia la derecha

tabla de entradas y salidas



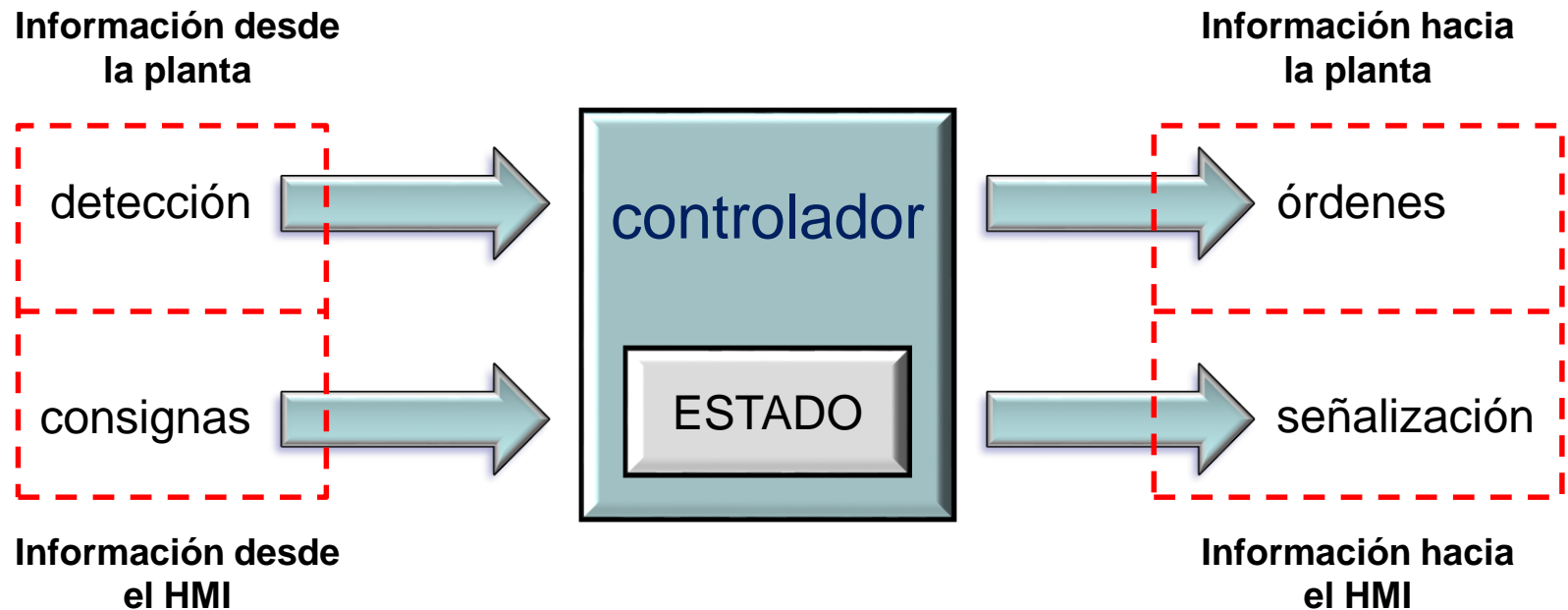
circuito de fuerza



circuito de mando

Aspecto lógico

- Un SAP se puede considerar un procesador de información



Tema 7.- Introducción a la automatización industrial

- 7.1. Concepto de automatización
- 7.2. Sistema automático de producción (SAP)
- 7.3. **Sistemas de eventos discretos (DES)**
- 7.4. Concepto de automatismo
- 7.5. Implantación de automatismos
- 7.6. El autómeta programable (PLC)
- 7.7. Lenguajes de programación de PLC IEC 61131-3

Definición

- Un DES (Discrete Event System) es un sistema **asíncrono** cuya evolución es dirigida por el acaecimiento de sucesos.
- Un DES es un sistema en el que su estado sólo cambia cuando ocurre un suceso.

Clasificación

- **DES combinacional (estático):** **la salida** del sistema en un determinado instante sólo **depende de la entrada** en ese preciso instante.

$$y(t) = f[u(t)]$$

- **DES secuencial (dinámico):** **la salida** del sistema en un determinado instante depende de **la entrada** en ese preciso instante y de la historia pasada del sistema (**estado**).

$$y(t) = f[u(t), q(t)]$$

Representación DES combinacional

$$M = P \wedge \neg CF$$

Función lógica

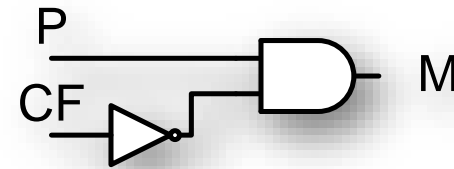


Diagrama lógico

P	CF	M
0	0	0
0	1	0
1	0	1
1	1	0

Tabla de verdad

		CF	
		0	1
P	0	0	0
	1	1	0

Mapa de Karnaugh

Representación DES secuencial

Autómata Finito (AF)

- Un AF es una máquina de estados que representa el comportamiento de un DES secuencial. Está definido por:

$$AF = \langle E, S, Q, \delta, \lambda, Q_0 \rangle$$

- Un AF establece una relación indirecta entre la entrada y la salida a través del estado.

Autómata Finito I

- **E** (alfabeto de entrada): conjunto de símbolos que recibe el AF.
- **S** (alfabeto de salida): conjunto de símbolos que emite el AF.
- **Q** (conjunto de estados): conjunto de estados en los que puede encontrarse el AF.
- **Q₀** (estado inicial): estado de partida del AF.

Autómata Finito II

- δ (función de transición entre estados):

$$\delta(Q, E) = Q'$$

- λ (función de lectura o salida):

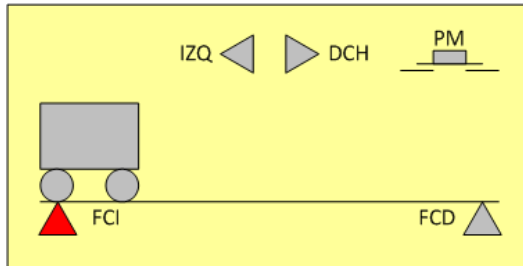
$$\lambda(Q) = S$$

máquina de Moore

$$\lambda(Q, E) = S$$

máquina de Mealy

Autómata Finito IV



carro va y viene

Nombre	Descripción
PM	pulsador de marcha
FCI	final de carrera izquierda
FCD	final de carrera derecha
IZQ	marcha hacia la izquierda
DCH	marcha hacia la derecha

tabla de entradas y salidas

Entradas = {PM, FCI, FCD} → $E = \{000, 001, 010, \dots\}$

Salidas = {IZQ, DCH} → $S = \{00, 10, 01, \dots\}$

$Q = \{Q_0, Q_1, Q_2\}$

$\delta: \delta(Q_0, \{PM, FCI\}) = Q_1$ $\lambda: \lambda(Q_0) = \{\emptyset\}$

$\delta(Q_1, \{FCD\}) = Q_2$ $\lambda(Q_1) = \{DCH\}$

$\delta(Q_2, \{FCI\}) = Q_0$ $\lambda(Q_2) = \{IZQ\}$

Tema 7.- Introducción a la automatización industrial

- 7.1. Concepto de automatización
- 7.2. Sistema automático de producción (SAP)
- 7.3. Sistemas de eventos discretos (DES)
- 7.4. Concepto de automatismo**
- 7.5. Implantación de automatismos
- 7.6. El autómeta programable (PLC)
- 7.7. Lenguajes de programación de PLC IEC 61131-3

Concepto de automatismo I

Definición

- **Automatismo (también llamado autómeta):** **dispositivo controlador** mediante el cual una máquina o proceso adquiere el carácter de automático.
- **Automatismo:** DES secuencial que controla el comportamiento de un proceso modelado como un DES secuencial.

Concepto de automatismo II

Automatismo vs Regulador

Automatismo

- Controla el estado
- Secuencia
- Sist. de eventos discretos
- Autómata finito
- Ejemplos:
 - Semáforo
 - Lavadora
 - Puerta automática
 - Ascensor

Regulador

- Controla una variable
- Consigna
- Sistemas continuos
- Ecuaciones Diferenciales
- Ejemplos:
 - Climatizador
 - Dirección asistida
 - Velocidad de crucero
 - Termostato

Representación de automatismos

- Tabla de fases
- Diagramas de relés y contactos
- GRAFCET (IEC 60848)

Concepto de automatismo IV

Tabla de fases primitiva

variables de entrada

variables de salida

		entradas (PM, FCI, FCD)									
		000	001	011	010	110	111	101	100		
estados	0				0	1				00	
	1				2	1			3	01	
	2	4			2	1				01	
	3	4						5	3	01	
	4	4	6						3	01	
	5		6					5	7	10	
	6	8	6					5		10	
	7	8			0	1			7	10	
	8	8			0	1			7	10	

estados

función de transición

función de lectura

salidas (IZQ, DCH)

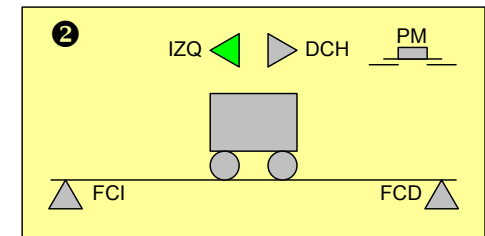
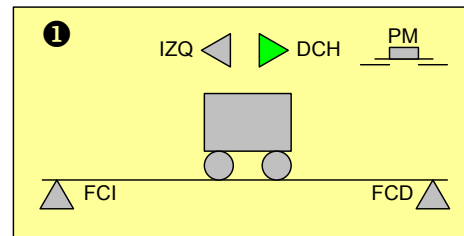
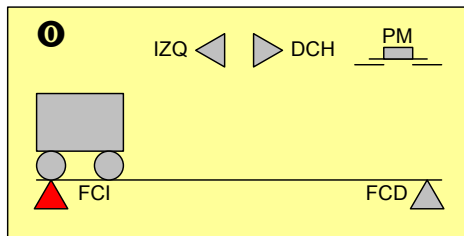
Concepto de automatismo V

Tabla de fases reducida

entradas (PM, FCI, FCD)

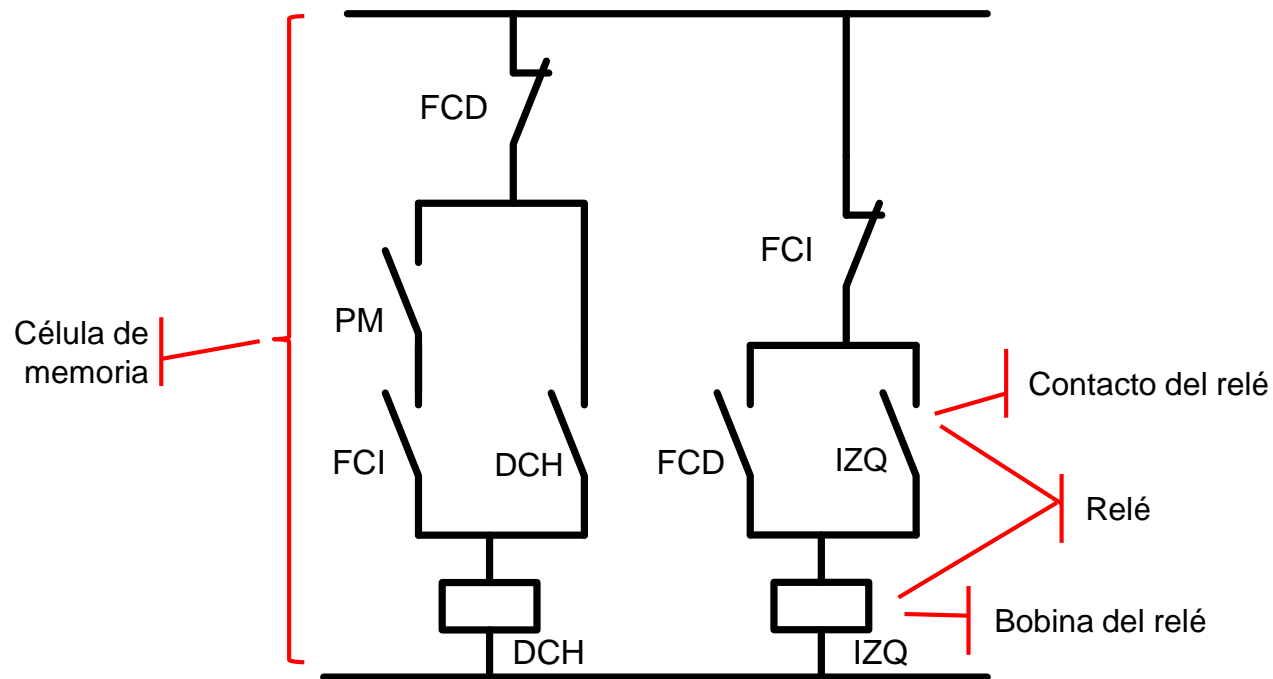
	000	001	011	010	110	111	101	100	
0				0	1				00
1	1	2		1	1		2	1	01
2	2	2		0	1		2	2	10

salidas (IZQ, DCH)



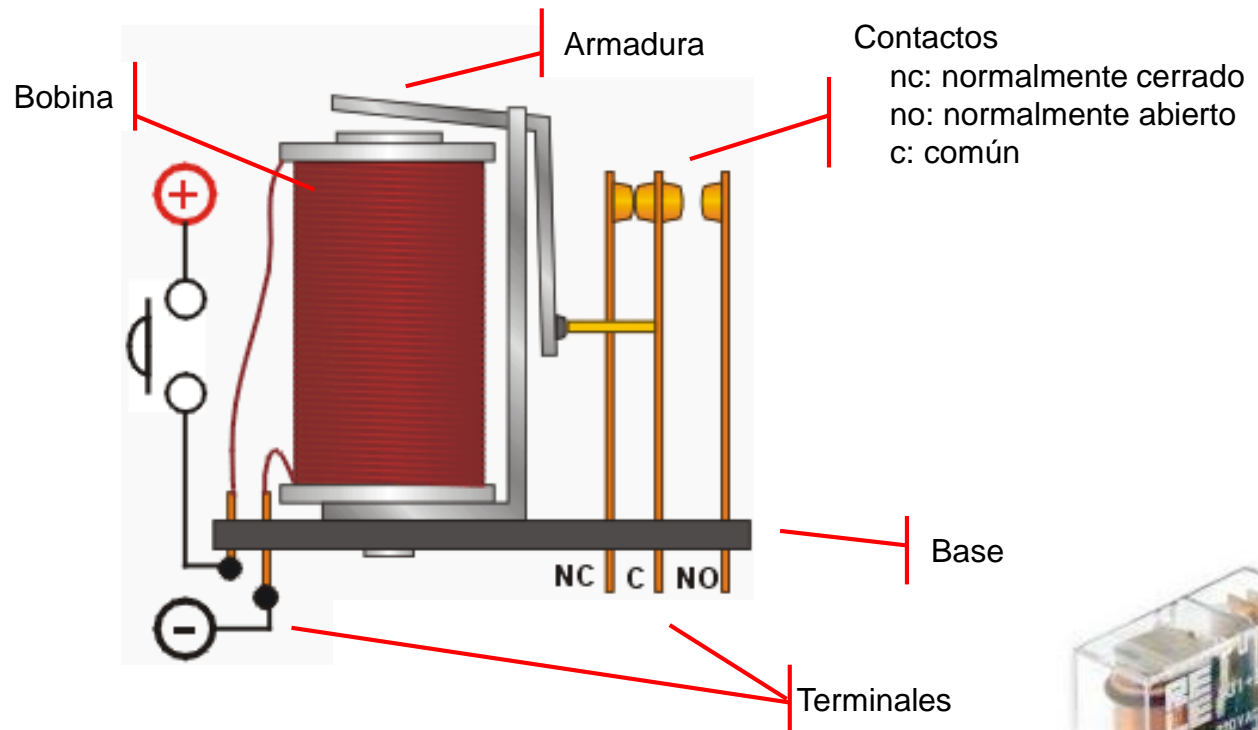
situaciones relevantes: REPOSO, DERECHA, IZQUIERDA

Diagramas de relés y contactos

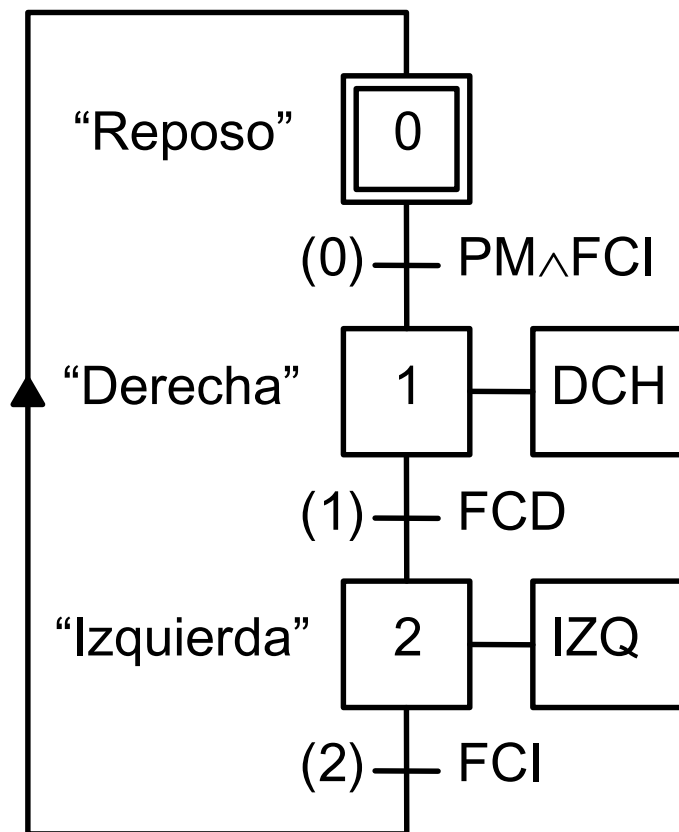


Concepto de automatismo VI (bis)

Diagramas de relés y contactos



GRAFCET (IEC 60848)



Tema 7.- Introducción a la automatización industrial

7.1. Concepto de automatización

7.2. Sistema automático de producción (SAP)

7.3. Sistemas de eventos discretos (DES)

7.4. Concepto de automatismo

7.5. Implantación de automatismos

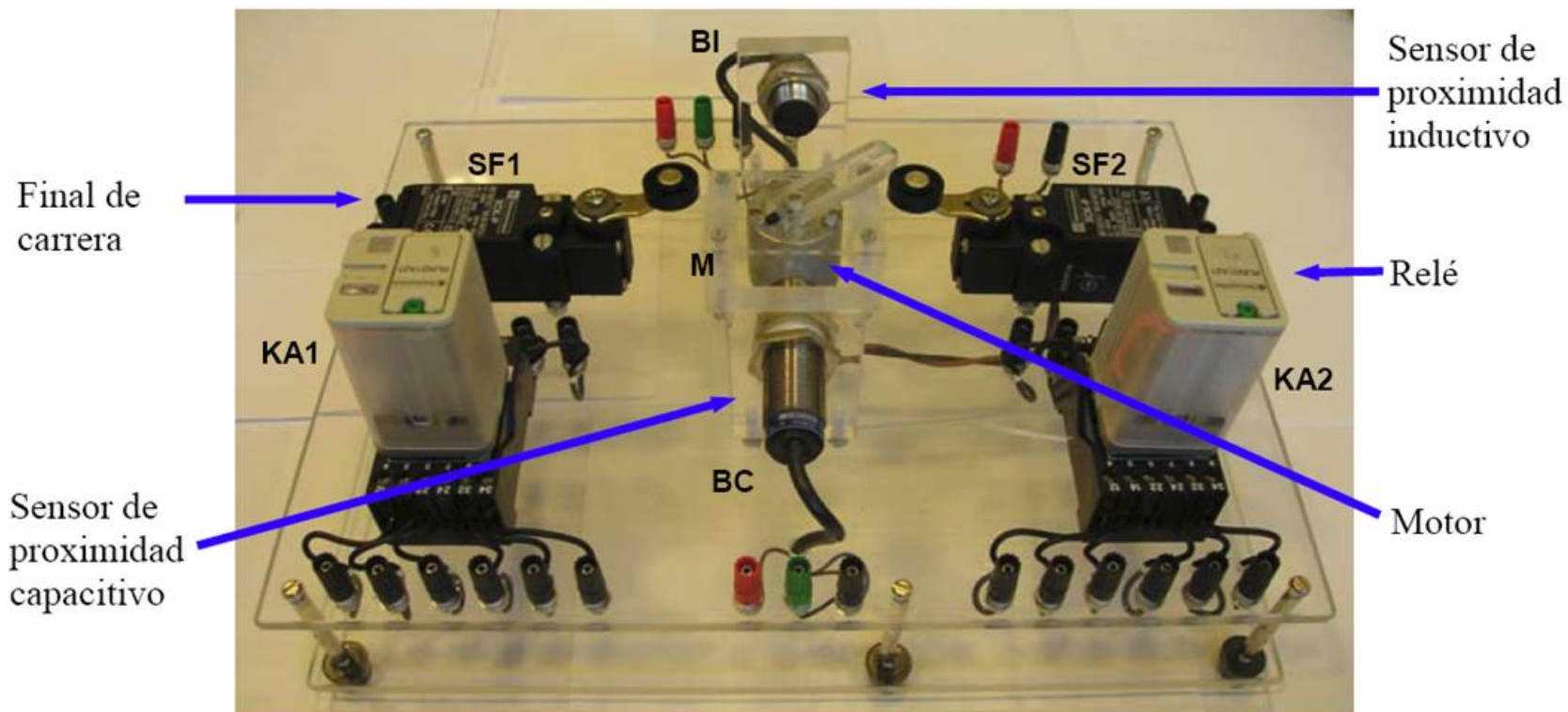
7.6. El autómeta programable (PLC)

7.7. Lenguajes de programación de PLC IEC 61131-3

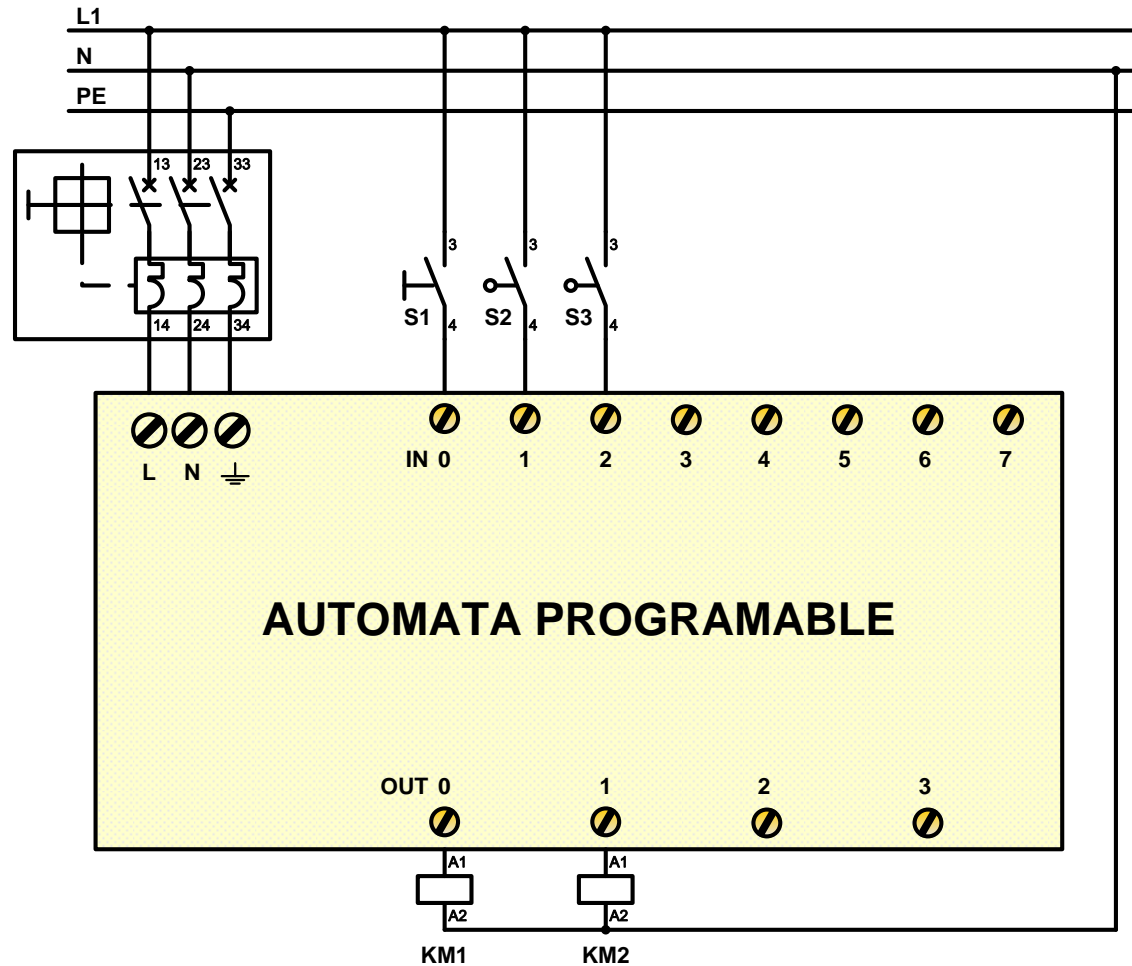
Implantación de automatismos I

- **Implantación cableada:**
 - Mecánica
 - Neumática
 - Eléctrica
 - ...
- **Implantación programada:**
 - Autómata programable (PLC)
 - Ordenador industrial
 - Micro-controlador
 - ...

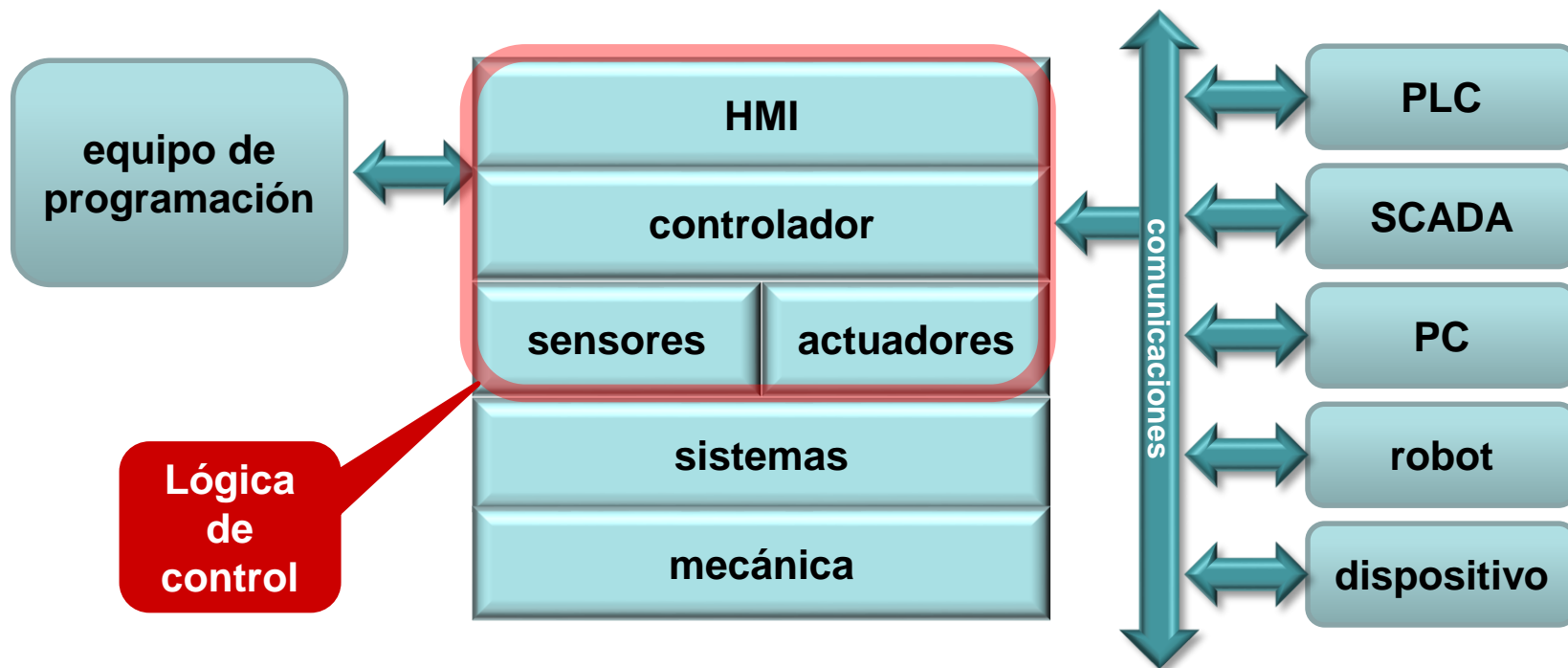
Implantación cableada



Implantación programada



Implantación programada - arquitectura



Tema 7.- Introducción a la automatización industrial

- 7.1. Concepto de automatización
- 7.2. Sistema automático de producción (SAP)
- 7.3. Sistemas de eventos discretos (DES)
- 7.4. Concepto de automatismo
- 7.5. Implantación de automatismos
- 7.6. El autómeta programable (PLC)**
- 7.7. Lenguajes de programación de PLC IEC 61131-3

Definición

- Un autómata programable industrial (PLC: Programmable Logic Controller) es **un dispositivo electrónico programable diseñado para controlar procesos secuenciales en tiempo real y en ambiente industrial.**

Características

- Flexibilidad
- Fiabilidad
- Modularidad
- Robustez
- Espacio reducido
- Realiza funciones complejas

Aspecto externo

- PLC compacto



- PLC semi-modular



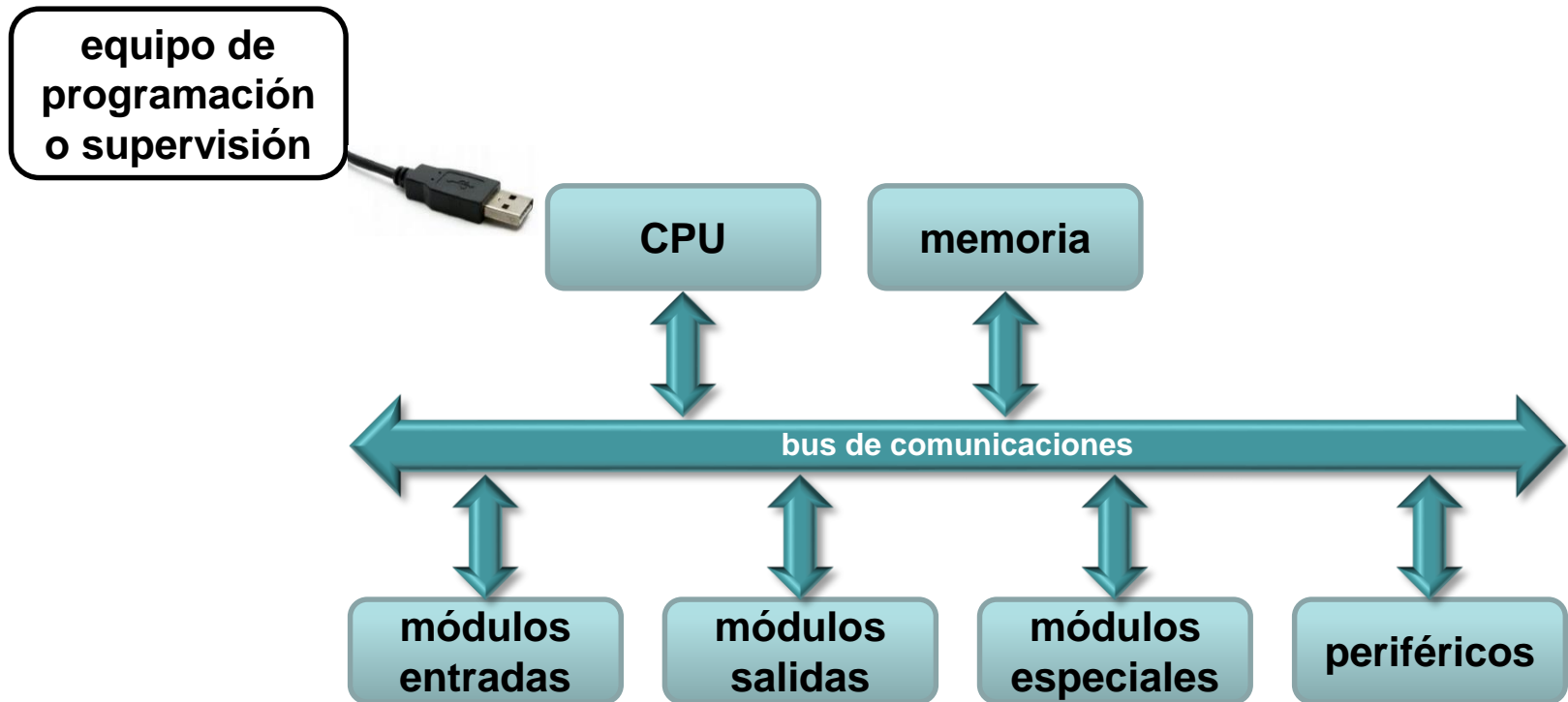
- PLC modular



Esquema PLC modular



Aspecto interno



Elementos de un PLC

- CPU
- Fuente de alimentación
- Módulos de entradas
- Módulos de salidas
- Módulos adicionales
- Elementos HMI

PLC VII

CPU

- Ejecuta los programas y comunica los distintos módulos del PLC.



Datos técnicos	CX1010
Processor	compatible with Pentium® MMX, clock frequency 500 MHz
Flash memory	64 MB Compact Flash card
Internal main memory	256 MB DDR RAM (not expandable)
Interfaces	1 x RJ 45 (Ethernet), 10/100 Mbit/s
Diagnostics LED	1 x power, 1 x LAN speed, 1 x LAN activity, TC status, 1 x flash access
Expansion slot	1 x Compact Flash type II insert with ejector
Clock	internal battery-backed clock for time and date (battery exchangeable)
Operating system	Microsoft Windows CE or Microsoft Windows Embedded Standard
Control software	TwinCAT PLC run-time or TwinCAT NC PTP run-time
System bus	16 bit ISA (PC/104 standard)

Fuente de alimentación

- Proporciona la tensión necesaria al PLC y a los distintos módulos.



Datos técnicos	CX1100-0004
Power supply	24 V DC (-15 %/+20 %)
Current supply E-bus	2 A
Display	FSTN display 2 lines x 16 characters of text, illuminated
Diagnostics LED	1 x PWR, 1 x L/A, 1 x Run
Max. power consumption	3.5 W

Módulos de entradas

- Recoge las señales de entrada al PLC. Los más usuales son de entradas digitales y analógicas.



Datos técnicos	EL1008
Number of inputs	8
Nominal voltage	24 V DC (-15 %/+20 %)
"0" signal voltage	-3...+5 V (EN 61131-2, type 3)
"1" signal voltage	15...30 V (EN 61131-2, type 3)
Input current	typ. 3 mA (EN 61131-2, type 3)

Módulos de salidas

- Envía las señales de salida a la planta. Los más usuales son de salidas digitales y analógicas.



Datos técnicos	EL2008
Number of outputs	8
Rated load voltage	24 V DC (-15 %/+20 %)
Max. output current	0.5 A (short-circuit-proof) per channel
Reverse voltage protection	yes

Módulos adicionales

- Módulos para la ampliación de las funciones o conectividad del PLC.



Elementos HMI

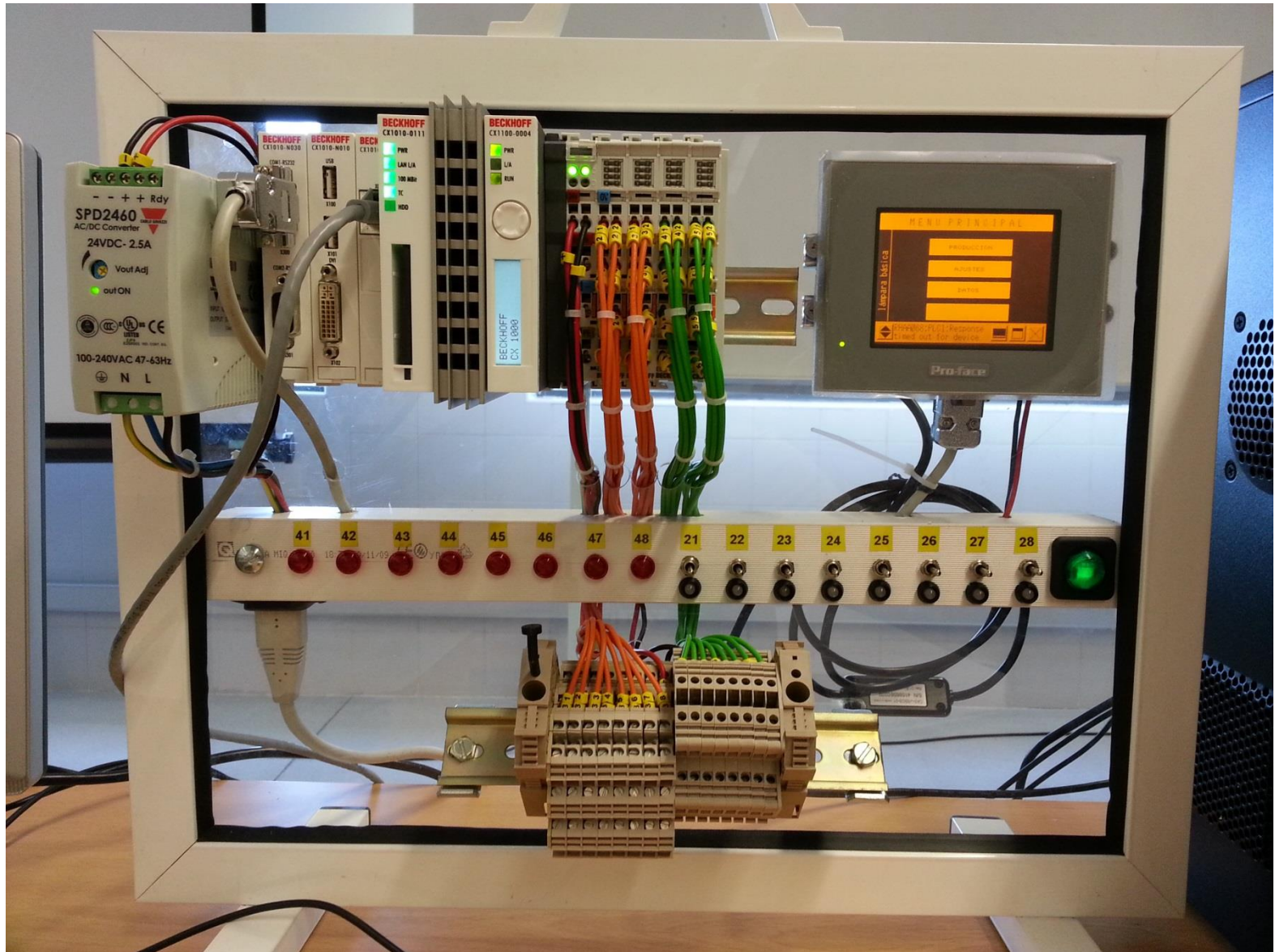
- Dispositivos de Interfaz **H**ombre **M**áquina.



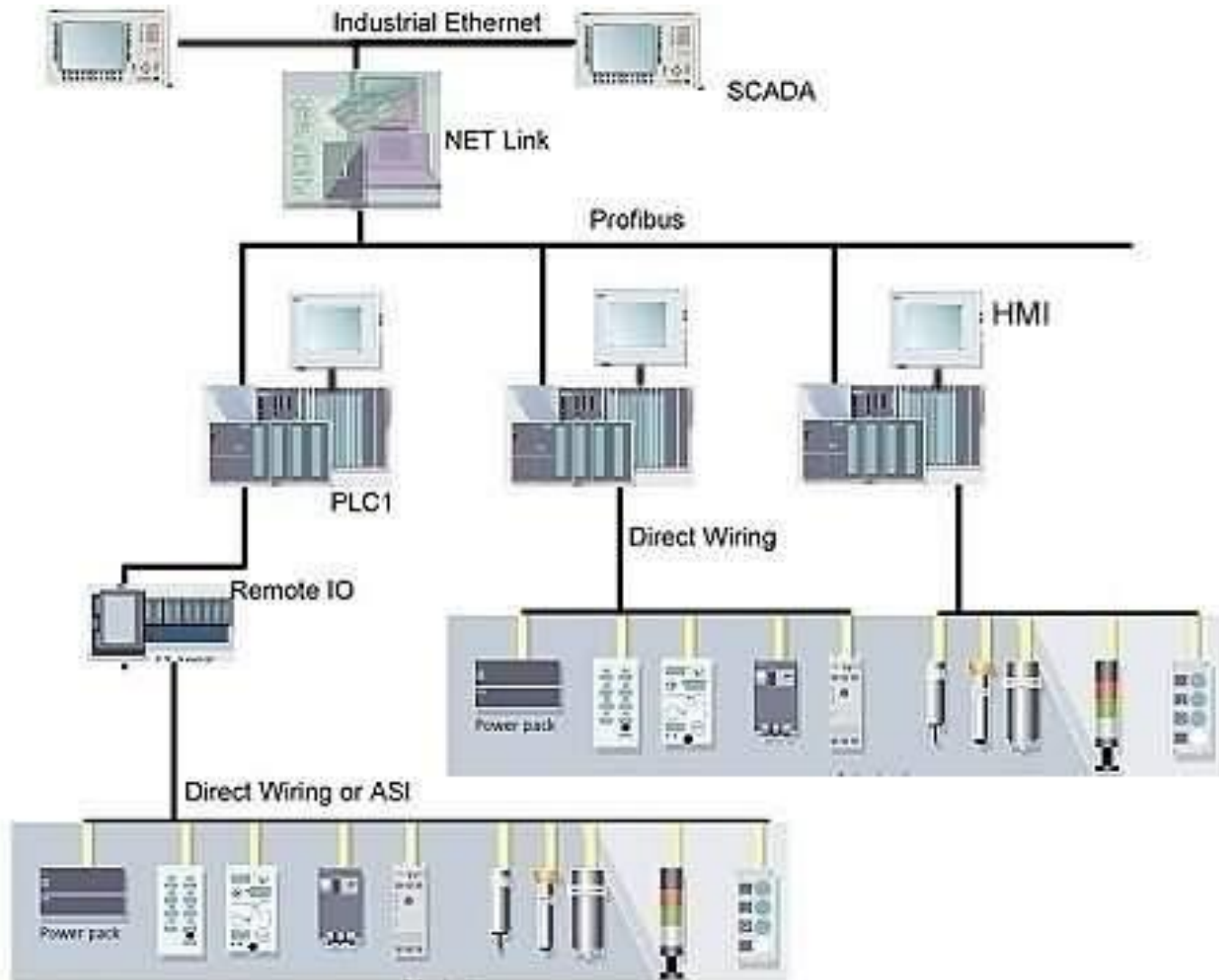
Datos técnicos

Tamaño del Display	3.8 pulgadas
Tipo de Display	Monocromo LCD
Colores de Display	8 tonos de ambar
Resolución del Display	320x240 Pixel
Backlight	backlight LED
Tipo de panel táctil	Analogico resistivo
Resolución del panel táctil	1024x1024
Interfaces Serie	2
Com 1	RS232
Com 2	RS422/485
USB I/F	Si

PLC XIII

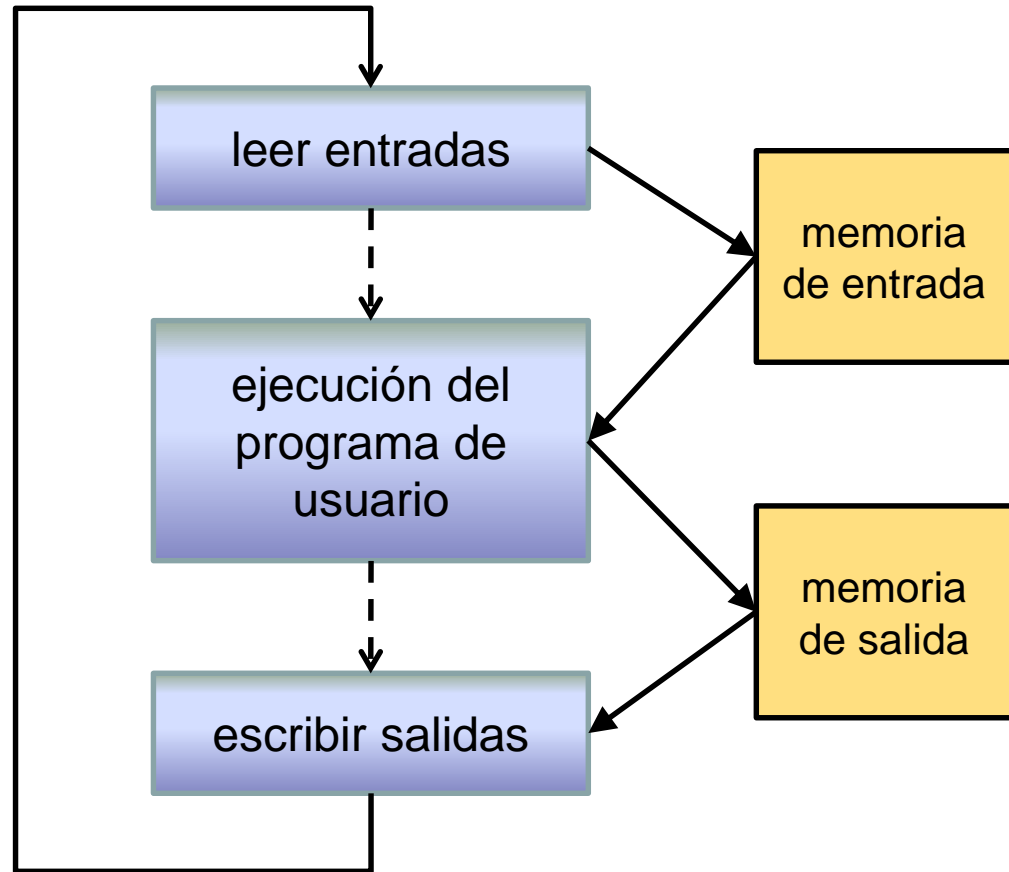


Sistemas de control distribuido



Ciclo básico de funcionamiento

tiempo de ciclo
PLC Beckhoff:
típico 1ms
máximo 50 μ s



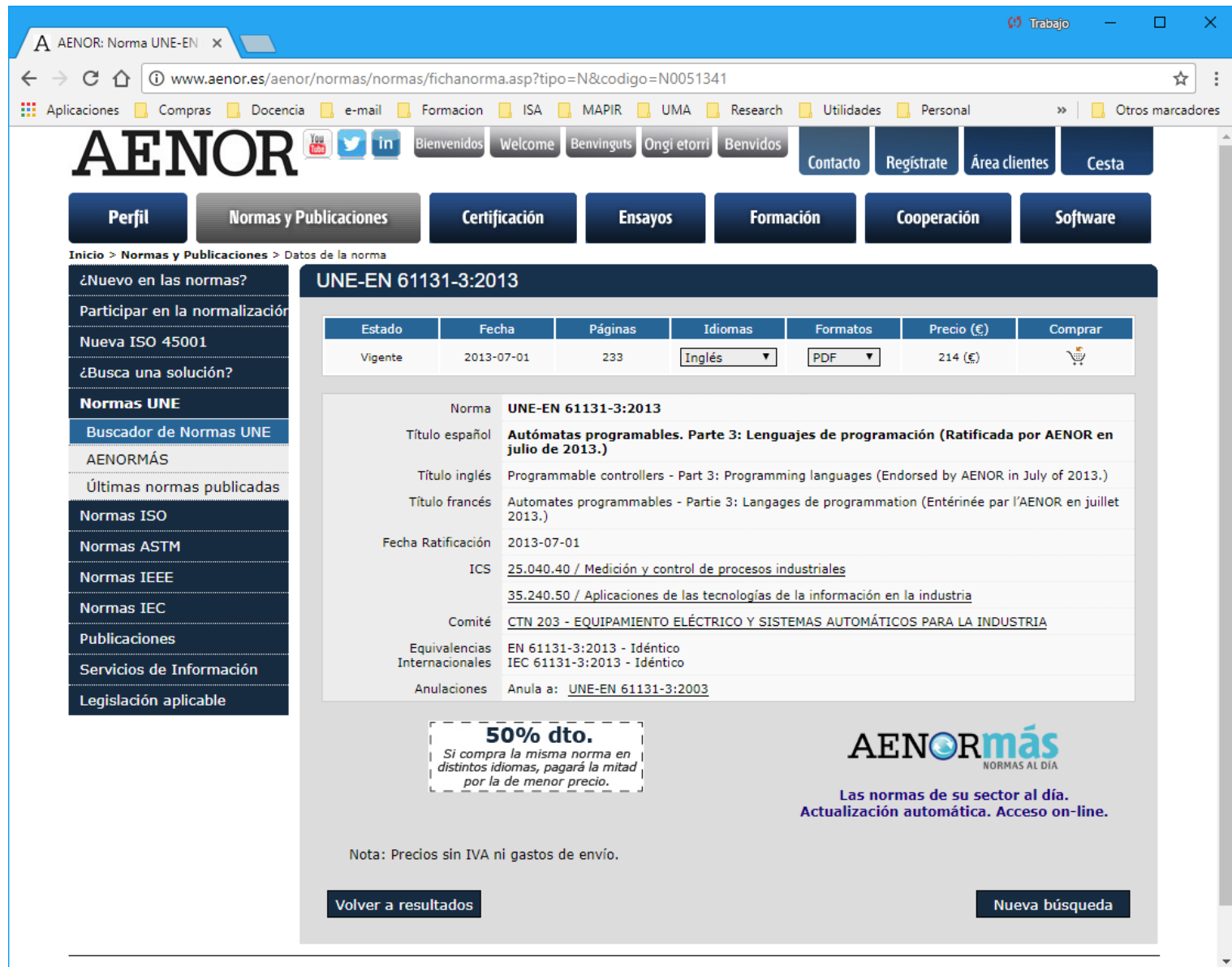
Tema 7.- Introducción a la automatización industrial

- 7.1. Concepto de automatización
- 7.2. Sistema automático de producción (SAP)
- 7.3. Sistemas de eventos discretos (DES)
- 7.4. Concepto de automatismo
- 7.5. Implantación de automatismos
- 7.6. El autómeta programable (PLC)
- 7.7. Lenguajes de programación de PLC IEC 61131-3

Introducción

- En la actualidad la mayoría de los PLCs son específicos del fabricante, con programación dependiente y conexión compleja con otros sistemas.
- La **norma IEC 61131** es el primer paso en la **estandarización** de los autómatas programables y sus periféricos, incluyendo los lenguajes de programación que se deben utilizar.


Lenguajes de PLC IEC 61131-3 I



AENOR

Inicio > Normas y Publicaciones > Datos de la norma

UNE-EN 61131-3:2013

Estado	Fecha	Páginas	Idiomas	Formatos	Precio (€)	Comprar
Vigente	2013-07-01	233	Inglés	PDF	214 (€)	

Norma **UNE-EN 61131-3:2013**

Título español **Autómatas programables. Parte 3: Lenguajes de programación (Ratificada por AENOR en julio de 2013.)**

Título inglés Programmable controllers - Part 3: Programming languages (Endorsed by AENOR in July of 2013.)

Título francés Automates programmables - Partie 3: Langages de programmation (Entérinée par l'AENOR en juillet 2013.)

Fecha Ratificación 2013-07-01

ICS [25.040.40 / Medición y control de procesos industriales](#)
[35.240.50 / Aplicaciones de las tecnologías de la información en la industria](#)

Comité **CTN 203 - EQUIPAMIENTO ELÉCTRICO Y SISTEMAS AUTOMÁTICOS PARA LA INDUSTRIA**

Equivalencias Internacionales EN 61131-3:2013 - Idéntico
IEC 61131-3:2013 - Idéntico

Anulaciones Anula a: [UNE-EN 61131-3:2003](#)

50% dto.
Si compra la misma norma en distintos idiomas, pagará la mitad por la de menor precio.

AENORMÁS
NORMAS AL DÍA

Las normas de su sector al día.
Actualización automática. Acceso on-line.

Nota: Precios sin IVA ni gastos de envío.

[Volver a resultados](#) [Nueva búsqueda](#)

son
ción
tros

n la
bles
de

Partes de la norma

- **Parte 1:** Información general
- **Parte 2:** Especificaciones de los equipos
- **Parte 3:** Lenguajes de programación
- **Parte 4:** Guías de usuario
- **Parte 5:** Comunicaciones
- **Parte 7:** Control borroso
- **Parte 8:** Guías de implantación de los lenguajes de programación

IEC 61131-3

- **Define los distintos tipos de datos** que se pueden utilizar.
- Posibilita el uso de variables simbólicas.
- **Estructura la programación** utilizando tres tipos de unidades de organización (**POU**).
- Unifica la implantación de las **funciones típicas** de los PLCs.
- Incorpora el término configuración para lograr la independencia hardware de la programación.
- **Define cinco lenguajes** de programación de PLCs distintos.

Ventajas del uso del IEC 61131-3

- Estándar aceptado internacionalmente.
- Ahorra tiempo.
- Permite una programación segura y de calidad.
- Ofrece el mejor lenguaje de programación para cada problema.

Tipos de datos

- **Booleano:** *BOOL*
- **Entero:** *INT, SINT, UINT, DINT, LINT, UDINT ...*
- **Real:** *REAL, LREAL*
- **Duración:** *TIME*
- **Fecha y hora:** *DATE, TOD, DT*
- **Carácter:** *STRING, WSTRING*
- **Cadena de bits:** *BYTE, WORD, DWORD, LWORD*
- Se permite la declaración de tipos de datos derivados y estructuras de datos.

Variables

- Al ser **declaradas se les asigna un identificador único** para su uso simbólico (independencia del hardware).
- Al ser declaradas se les asigna un tipo de dato elemental o derivado.
- Al ser declaradas se les puede asignar un valor inicial.
- Se pueden declarar de un solo elemento, tablas y estructuras.

Tipos de variables

- **De entrada:** *VAR_INPUT*
 - **De salida:** *VAR_OUTPUT*
 - **De entrada y salida:** *VAR_IN_OUT*
 - **Globales:** *VAR_GLOBAL*
 - **Externas:** *VAR_EXTERNAL*
 - **De acceso:** *VAR_ACCESS*
 - **Temporales:** *VAR_TEMP*
 - **De retención:** *RETAIN*
 - **Constantes:** *CONSTANT*
- Variables por referencia

Variables vinculadas con el hardware

- Al ser declaradas se les añade la palabra reservada *AT*, el símbolo *%*, una localización, un tamaño y uno o varios enteros sin signo separados por puntos que representa la dirección.
- Localización: *I*, *Q* o *M*
- Tamaño: nada, *X*, *B*, *W*, *D*, *L* o ***

Variables vinculadas con el hardware

- Al ser declaradas se les añade la palabra reservada *AT*, el símbolo *%*, una localización, un tamaño y uno o varios enteros sin signo separados por puntos que representa la dirección.
- Localización: *I*, *Q* o *M*
- Tamaño: nada, *X*, *B*, *W*, *D*, *L* o ***
- Ejemplos:
 - Entrada1 *AT %I3.0 : BOOL;*

Define la variable de tipo BOOL Entrada1 como variable de entrada vinculada al bit 0 del modulo 3

Variables vinculadas con el hardware

- Al ser declaradas se les añade la palabra reservada *AT*, el símbolo *%*, una localización, un tamaño y uno o varios enteros sin signo separados por puntos que representa la dirección.
- Localización: *I*, *Q* o *M*
- Tamaño: nada, *X*, *B*, *W*, *D*, *L* o ***
- Ejemplos:
 - Entrada1 *AT %I3.0* : *BOOL*;
 - Salida3 *AT %Q** : *BYTE*;

Define la variable de tipo BYTE Salida3 como variable de salida aún no vinculada

Variables vinculadas con el hardware

- Al ser declaradas se les añade la palabra reservada *AT*, el símbolo *%*, una localización, un tamaño y uno o varios enteros sin signo separados por puntos que representa la dirección.
- Localización: *I*, *Q* o *M*
- Tamaño: nada, *X*, *B*, *W*, *D*, *L* o ***
- Ejemplos:
 - Entrada1 *AT %I3.0* : *BOOL*;
 - Salida3 *AT %Q** : *BYTE*;
 - Dato2 *AT %MD12* : *REAL*;

Define la variable de tipo REAL Dato2 como variable de memoria vinculada a los bytes de memoria 12, 13, 14 y 15

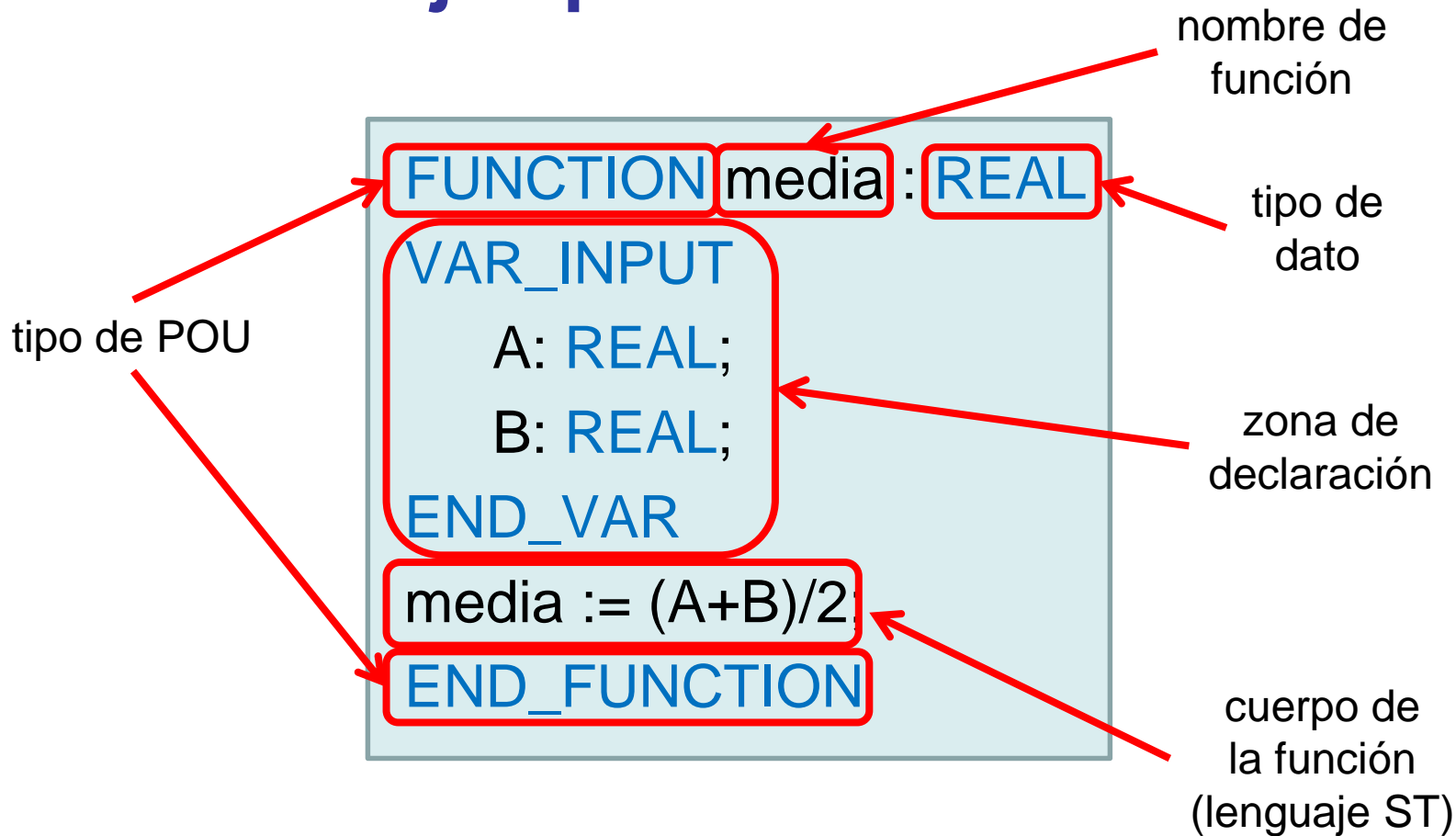
Unidades de organización (POU)

- Tres tipos de POU:
 - Funciones
 - Bloques funcionales
 - Programas
- Partes de un POU:
 - Tipo de POU, nombre y tipo de dato en funciones
 - Zona de declaración de variables
 - Cuerpo del POU
- **Sin recursividad.**

Función

- Es un POU con parámetros de entrada, que devuelve un sólo valor de cualquier tipo de dato y que no contiene variables estáticas (sin memoria).
- Las funciones no pueden invocar a bloques funcionales, sólo pueden invocar a otras funciones.

Ejemplo de función



Funciones estándar

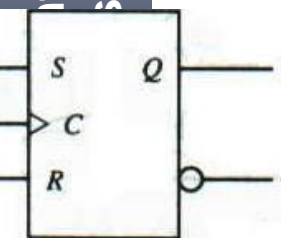
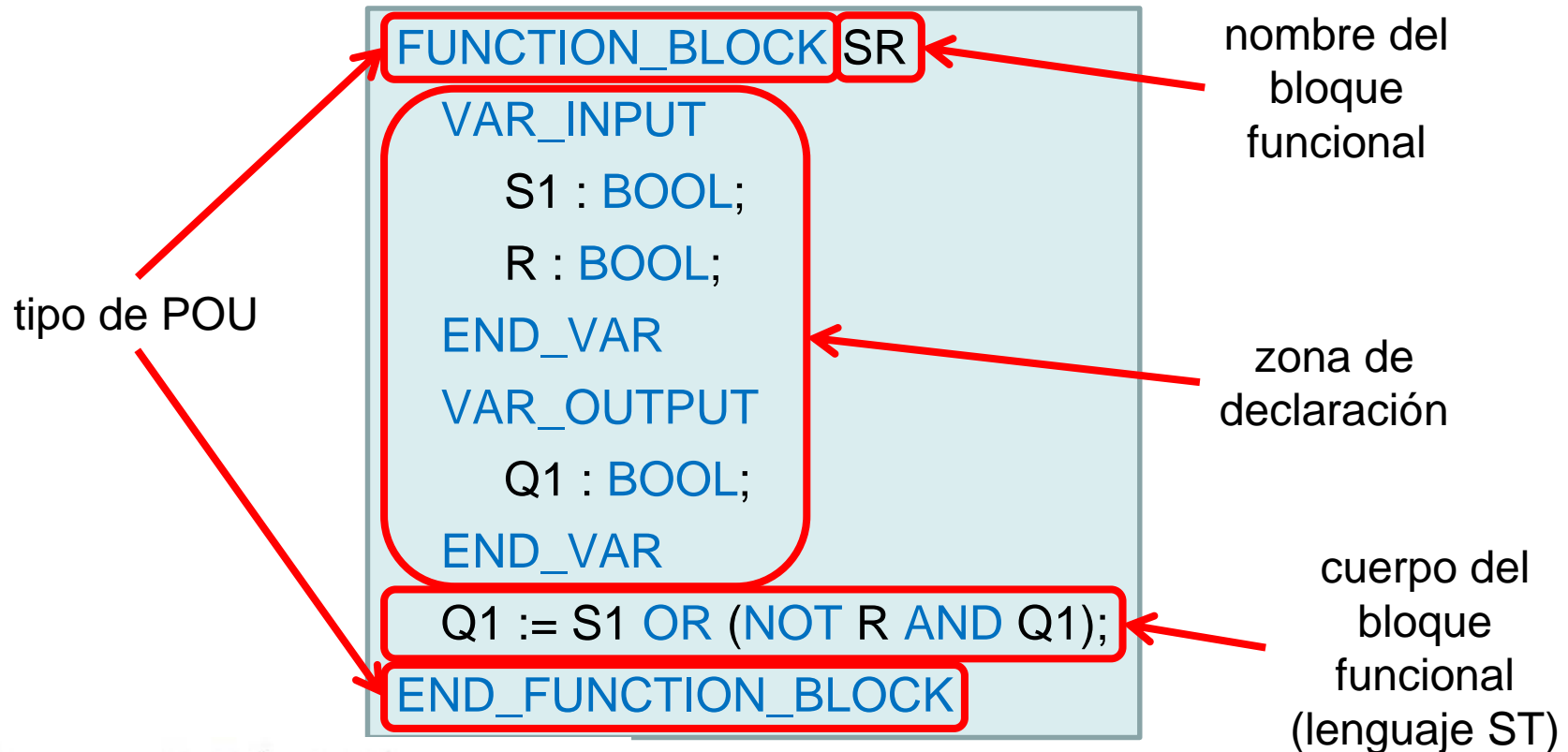
- **Operaciones de bits:** *AND, OR, NOT, SHL, ROR, ...*
- **Numéricas:** *ABS, COS, ADD, SQRT, ...*
- **Conversión de tipo:** *REAL_TO_INT, BOOL_TO_BYTE, ...*
- **Selección:** *MIN, MAX, LIMIT, SEL, MUX*
- **Comparación:** *GT, GE, EQ, LT, LE, NE*
- **Caracteres:** *LEN, LEFT, RIGHT, MID, CONCAT, FIND, ...*

Estas funciones deben estar incluidas en el lenguaje del autómatas para que cumpla con la norma IEC 61131-3

Bloque funcional

- Es un POU con parámetros de entrada, parámetros de salida y que contiene variables estáticas (con memoria).
- Los bloques funcionales se utilizan usando el concepto de instanciación.
- Los bloques funcionales pueden invocar tanto a otros bloques funcionales como a funciones.

Ejemplo de bloque funcional



b) Tabla característica

S	R	Q (t+1)	
0	0	Q (t)	Sin cambio
0	1	0	Limpiar en 0
1	0	1	Establecer en 1
1	1	X → 1	Indeterminado

biestable SR

Instanciación de bloque funcional

declaración de variables de tipo SR (instanciación)

VAR

M1 : SR;
M2 : SR;

ActivarM1 AT %I*: BOOL;
DesactivarM1 AT %I*: BOOL;
ActivarM2 AT %I*: BOOL;
DesactivarM2 AT %I*: BOOL;
Motor1 AT %Q*: BOOL;
Motor2 AT %Q*: BOOL;

END_VAR

asignación a variables de los valores de los parámetros de salida de los bloques funcionales

M1.S1 := ActivarM1;
M1.R := DesactivarM1;
M2.S1 := ActivarM2;
M2.R := DesactivarM2;

M1;
M2;

Motor1 := M1.Q1;
Motor2 := M2.Q1;

asignación de valores a los parámetros de entrada de los bloques funcionales

llamada a los bloques funcionales

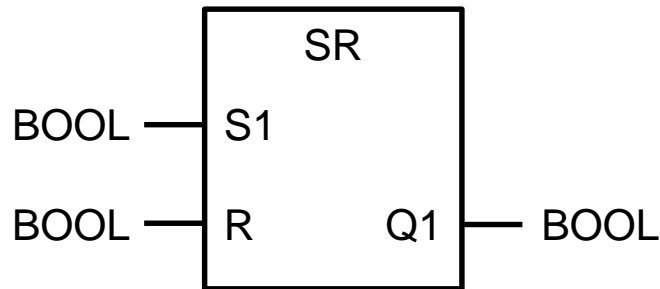
Bloques funcionales estándar

- **Biestables:** *SR, RS*
- **Detección de flancos:** *R_TRIG, F_TRIG*
- **Temporizadores:** *TP, TON, TOF*
- **Contadores:** *CTU, CTD, CTUD*

Estos bloques funcionales deben estar incluidos en el lenguaje del autómatas para que cumpla con la norma IEC 61131-3

Bloques funcionales: biestables I

Set prioritario (SR)



símbolo lógico

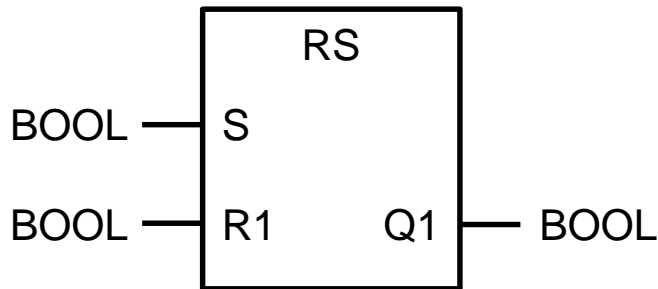
S1	R	Q1
0	0	Q1
0	1	0
1	0	1
1	1	1

tabla de verdad

- S1**: condición de activación
- R**: condición de desactivación
- Q1**: estado del biestable

Bloques funcionales: biestables II

Reset prioritario (RS)



símbolo lógico

S	R1	Q1
0	0	Q1
0	1	0
1	0	1
1	1	0

tabla de verdad

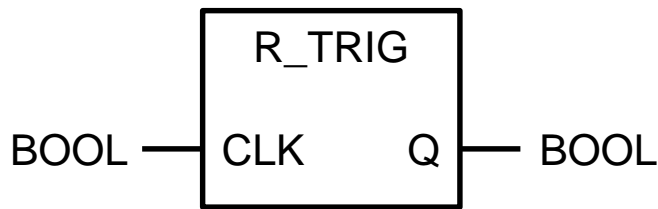
S: condición de activación

R1: condición de desactivación

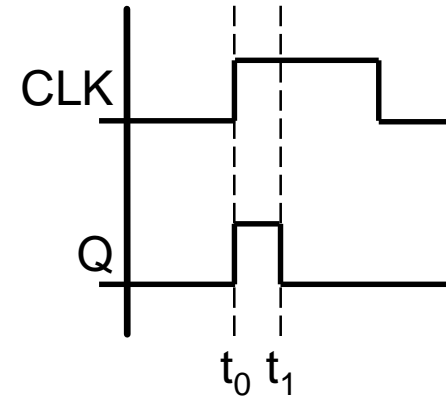
Q1: estado del biestable

Bloques funcionales: flancos I

Flanco de subida (R_TRIG)



símbolo lógico

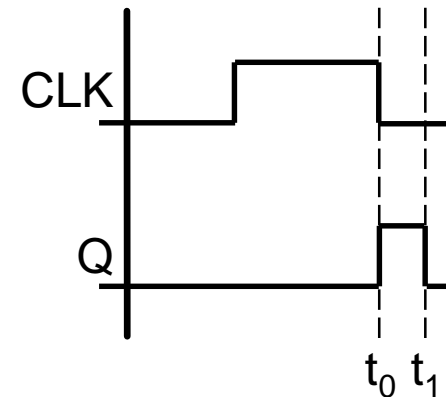
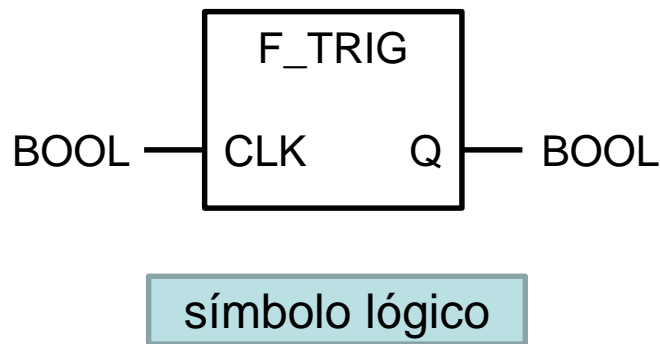


CLK: señal a monitorizar
Q: estado de la detección

t_0 : CLK cambia de 0 a 1 ($Q = 1$)
 t_1 : $Q = 0$
 $t_1 - t_0 =$ un ciclo de reloj del PLC

Bloques funcionales: flancos II

Flanco de bajada (F_TRIG)

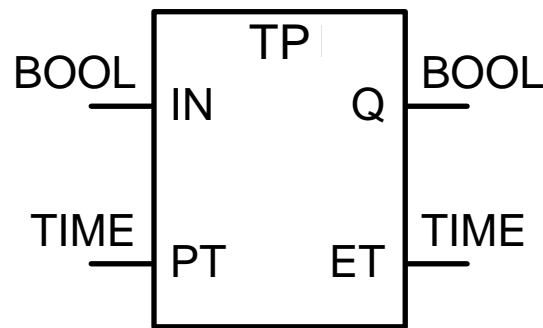


CLK: señal a monitorizar
Q: estado de la detección

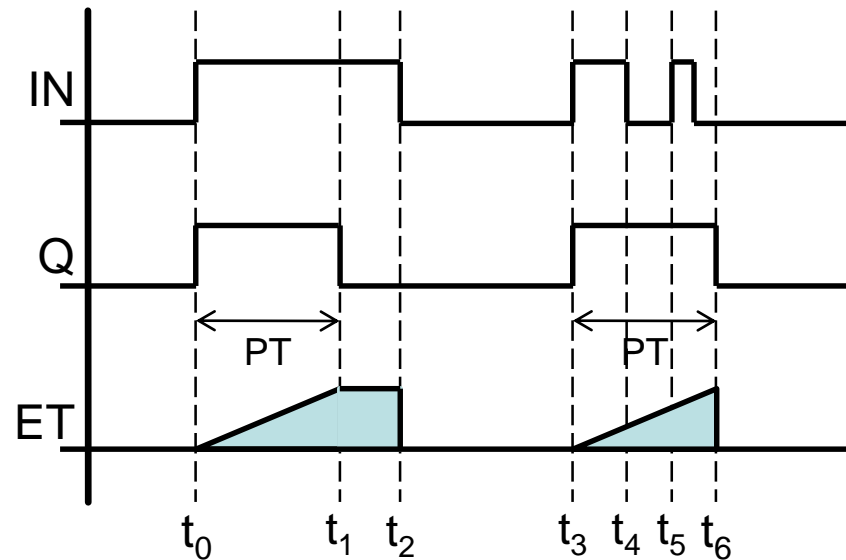
t_0 : CLK cambia de 1 a 0 ($Q = 1$)
 t_1 : $Q = 0$
 $t_1 - t_0 =$ un ciclo de reloj del PLC

Bloques funcionales: temporizadores I

Pulso (TP)



símbolo lógico



IN: condición de activación

PT: tiempo programado

Q: estado temporizador

ET: tiempo transcurrido

t_0 y t_3 : activación IN (inicio de cuenta y $Q = 1$)

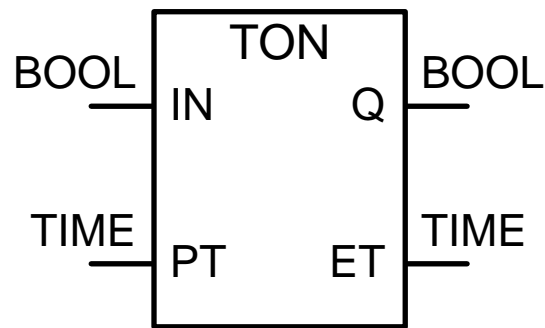
$t_1 = t_0 + PT$ ($Q = 0$) $t_6 = t_3 + PT$ ($Q = 0$)

t_2 y t_4 : desactivación IN (no afecta a Q)

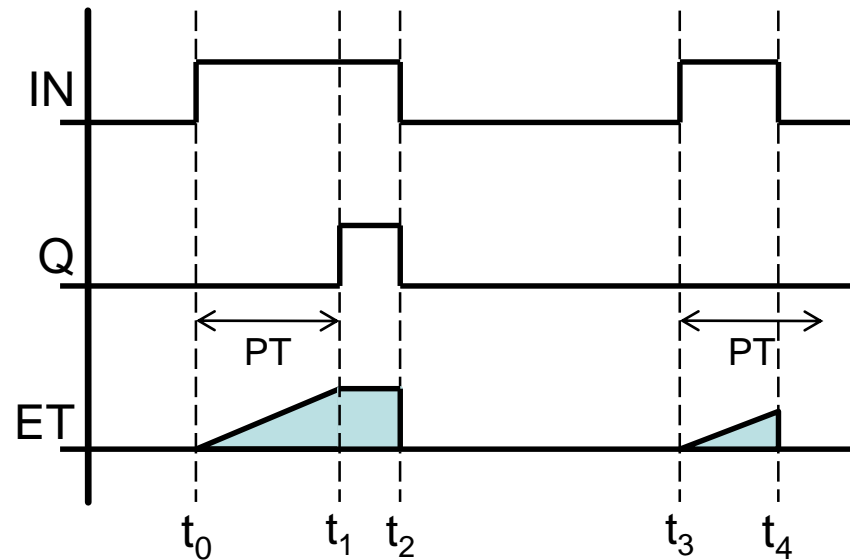
$t_5 < t_3 + PT$: activación IN (no afecta a Q ni ET)

Bloques funcionales: temporizadores II

Retardo a la conexión (TON)



símbolo lógico



IN: condición de activación

PT: tiempo programado

Q: estado temporizador

ET: tiempo transcurrido

t_0 y t_3 : activación IN (inicio de cuenta)

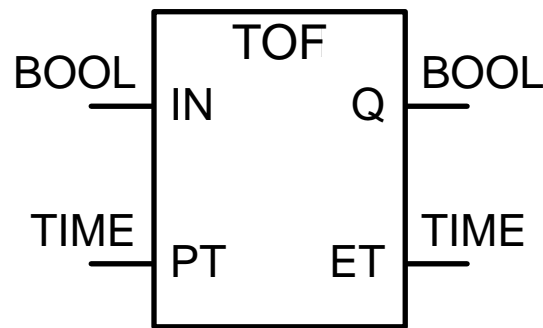
$t_1 = t_0 + PT$ ($Q = 1$)

t_2 : desactivación IN ($Q = 0$ y $ET = 0$)

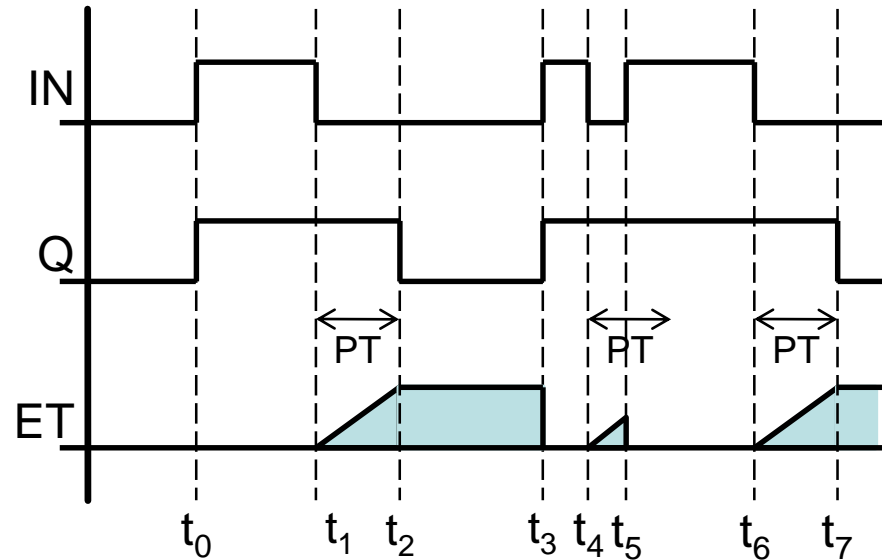
$t_4 < t_3 + PT$: desactivación IN ($ET = 0$)

Bloques funcionales: temporizadores III

Retardo a la desconexión (TOF)



símbolo lógico



IN: condición de activación

PT: tiempo programado

Q: estado temporizador

ET: tiempo transcurrido

t_0 y t_3 : activación IN ($Q = 1$)

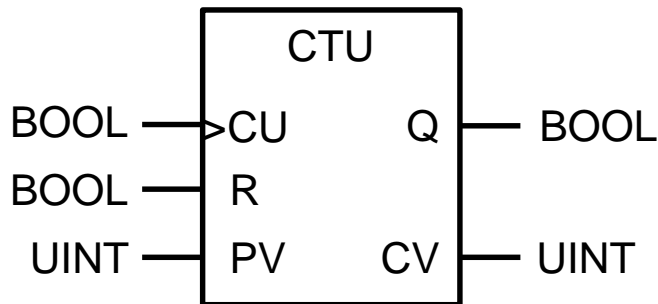
t_1 , t_4 y t_6 : desactivación IN (inicio cuenta)

$t_2 = t_1 + PT$ ($Q = 0$) $t_7 = t_6 + PT$ ($Q = 0$)

$t_5 < t_4 + PT$: activación IN ($ET = 0$)

Bloques funcionales: contadores I

Sólo incremento (CTU)



símbolo lógico

CU: incremento (flanco de subida)

R: reset (CV = 0)

PV: valor programado

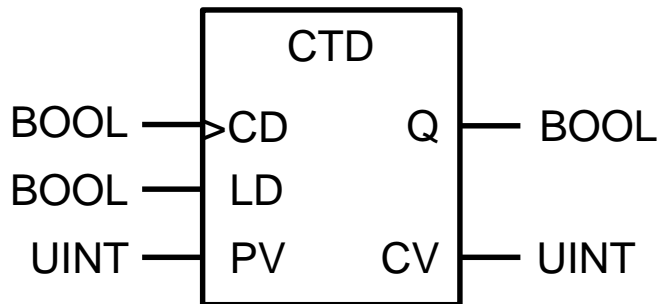
Q: estado contador

CV: valor de la cuenta

$CV = PV \Rightarrow Q = 1$

Bloques funcionales: contadores II

Sólo decremento (CTD)



símbolo lógico

CD: decremento (flanco de subida)

LD: carga (CV = PV)

PV: valor programado

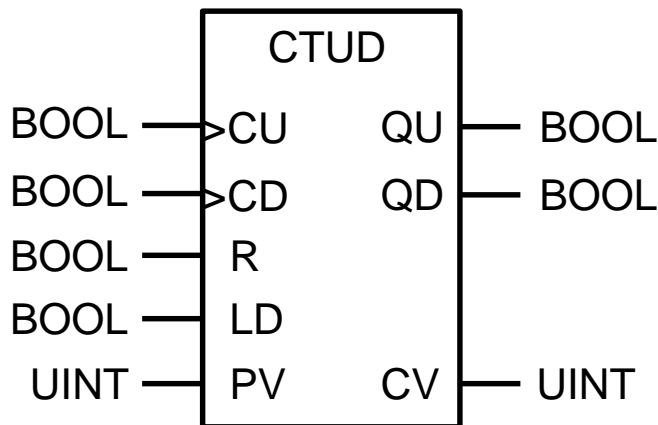
Q: estado contador

CV: valor de la cuenta

$$CV = 0 \Rightarrow Q = 1$$

Bloques funcionales: contadores III

De incremento y decremento (CTUD)



símbolo lógico

- CU**: incremento (flanco de subida)
- CD**: decremento (flanco de subida)
- R**: reset ($CV = 0$)
- LD**: carga ($CV = PV$)
- PV**: valor programado
- QU**: límite superior alcanzado
- QD**: límite inferior alcanzado
- CV**: valor de la cuenta

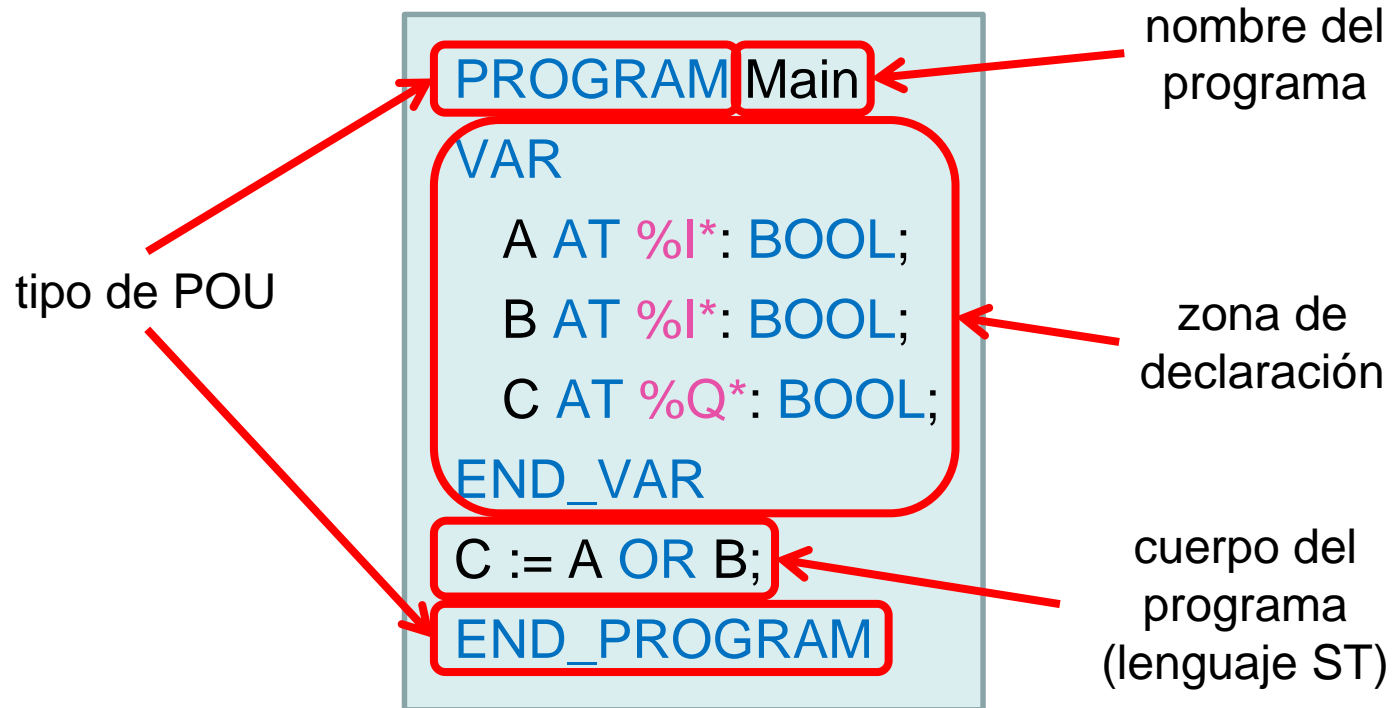
$$CV = 0 \Rightarrow QD = 1$$

$$CV = PV \Rightarrow QU = 1$$

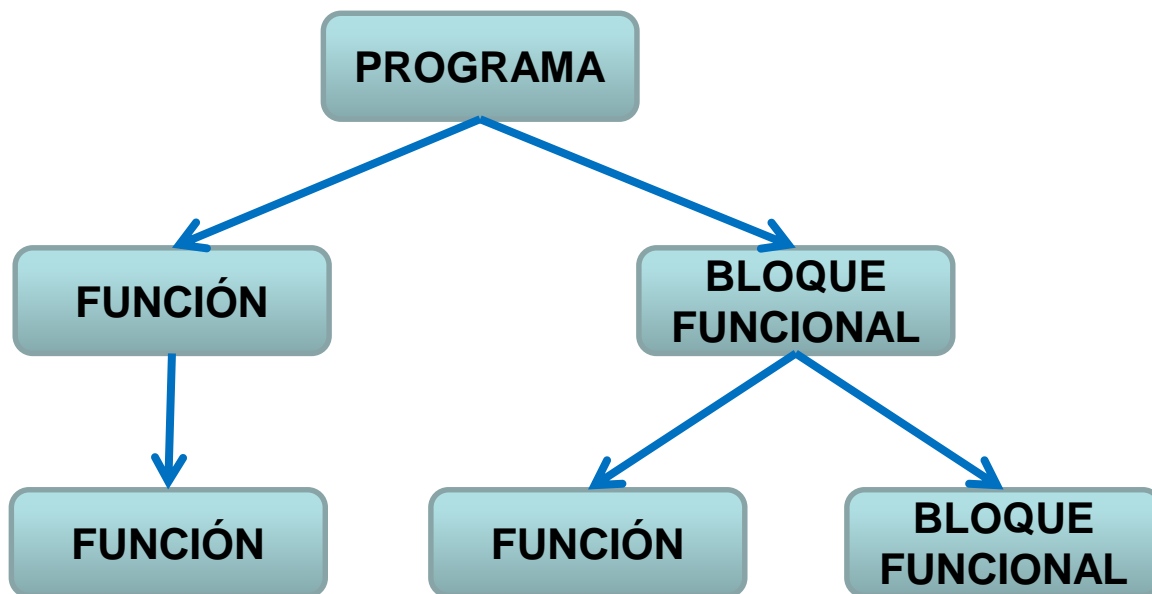
Programa

- Es un **POU sin parámetros de entrada ni de salida**. Normalmente representa al programa principal.
- Los programas pueden invocar tanto a bloques funcionales como a funciones.

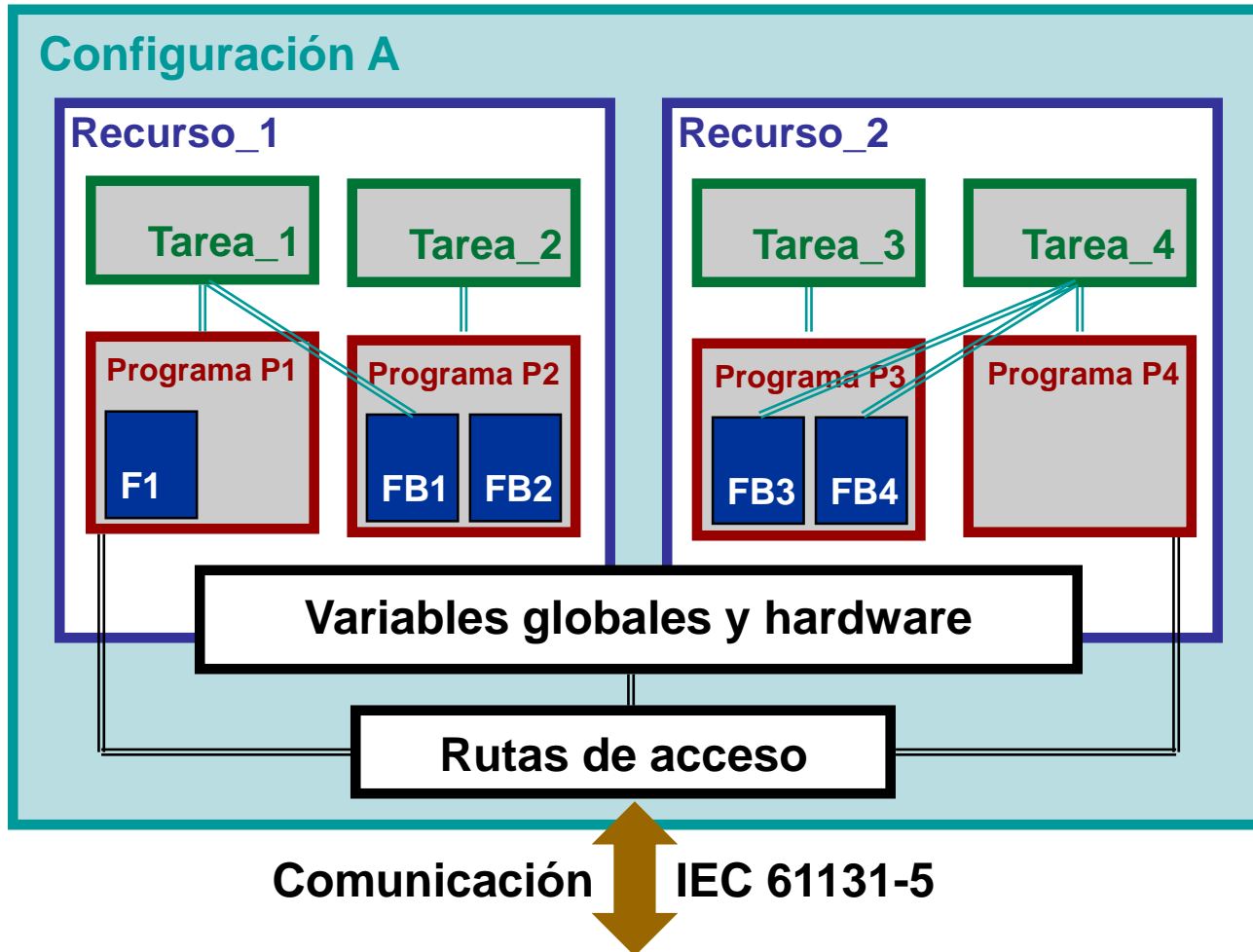
Ejemplo de programa



Llamadas entre POUs



Configuración



El IEC 61131-3 utiliza el término configuración para agrupar todos los recursos del PLC y dotarlos de medios para el intercambio de datos.

De esta forma, la programación se hace independiente del hardware.

Lenguajes de programación

- Textuales:
 - Lenguaje IL: lista de instrucciones.
 - Lenguaje ST: texto estructurado.
- Gráficos:
 - Lenguaje LD: diagrama ladder.
 - Lenguaje FBD: diagrama de bloques funcionales.
 - Lenguaje SFC: gráfico secuencial de función.

Lenguaje IL

- Lenguaje tipo ensamblador.
- Ejemplo:

```
LD      pesar
JMPC   AHORA
ST      ENO
RET
AHORA : LD      peso_bruto
        SUB     peso_tara
        ST      PESO
```

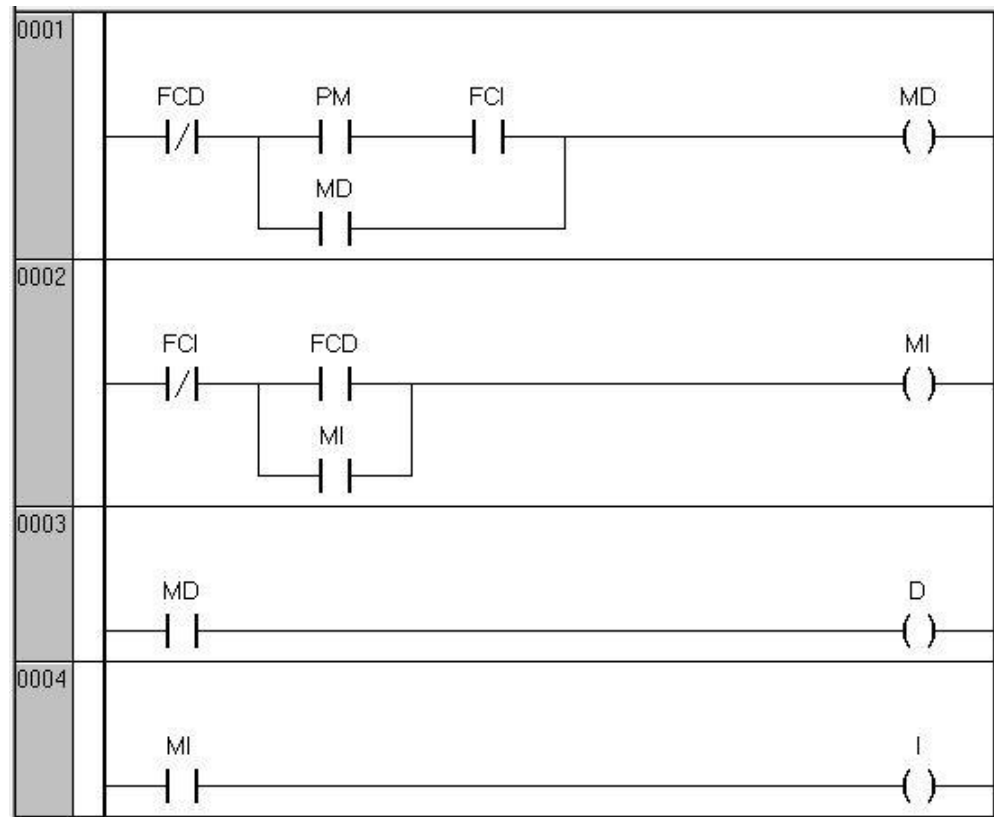
Lenguaje ST

- Lenguaje de alto nivel.
- Ejemplo:

```
IF R THEN
    CV := 0;
ELSIF CU AND (CV < PV) THEN
    CV := CV + 1;
ENDIF;
Q := (CV >= PV);
```

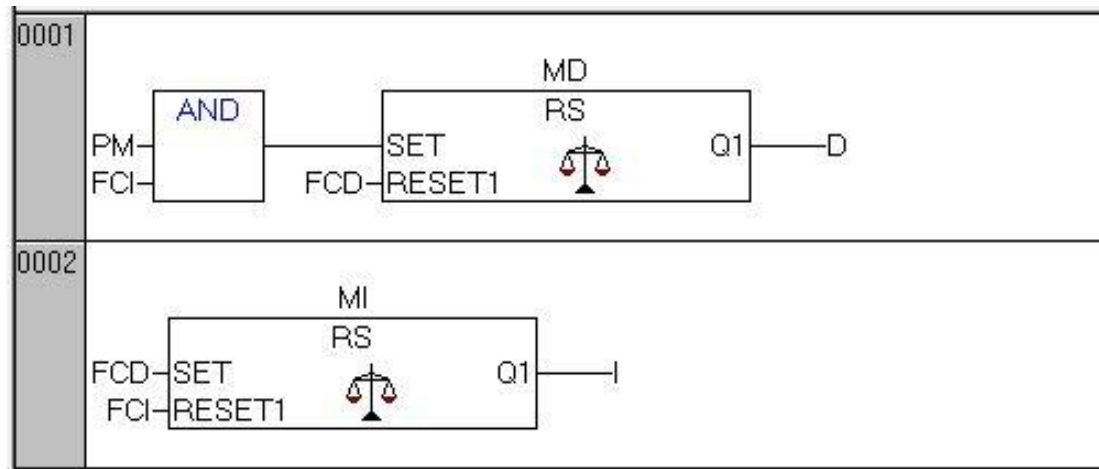
Lenguaje LD

- Representación similar a diagramas eléctricos.
- Ejemplo:



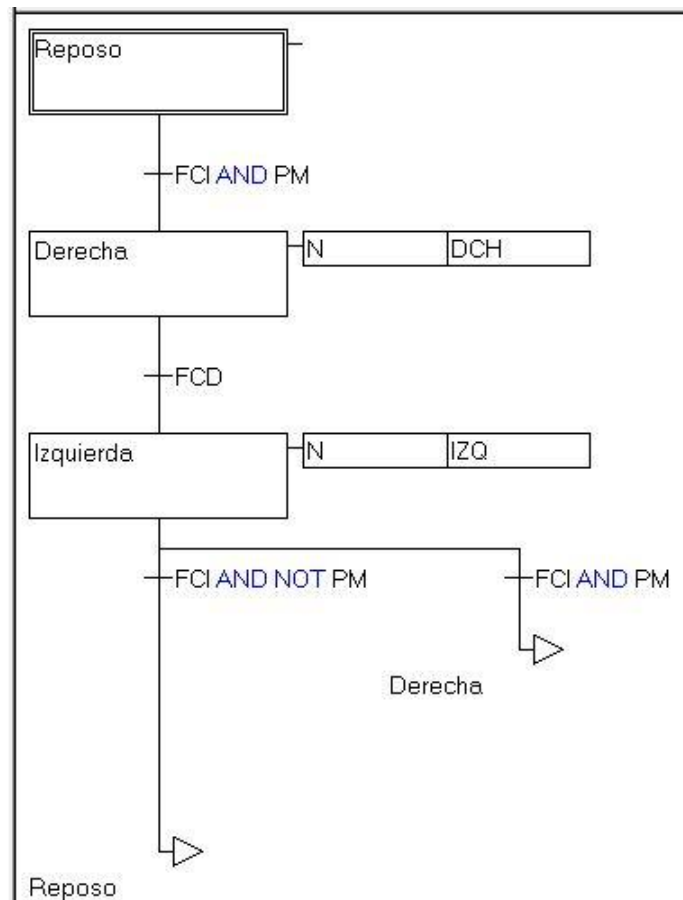
Lenguaje FBD

- Representación similar a diagramas electrónicos.
- Ejemplo:



Lenguaje SFC

- Representación de la evolución del estado del sistema.
- Ejemplo:



Tema 7.- Introducción a la automatización industrial

- 7.1. Concepto de automatización
- 7.2. Sistema automático de producción (SAP)
- 7.3. Sistemas de eventos discretos (DES)
- 7.4. Concepto de automatismo
- 7.5. Implantación de automatismos
- 7.6. El autómeta programable (PLC)
- 7.7. Lenguajes de programación de PLC IEC 61131-3

FIN

Automática

2º Curso del Grado en

Ingeniería en Tecnología Industrial