

Nothing can stop automation



Tema 5. Control Automático de Procesos Industriales mediante PLC

Pirámide de la automatización ANSI/ISA-95

- ✓ El estándar ANSI/ISA-95 (**Integración de los Sistemas de Control Empresarial**) divide jerárquicamente el proceso de automatización industrial en **cinco capas** o **niveles tecnológicos diferenciados**:

Nivel 4:

ERP



Enterprise Level

Nivel 3:

MES



Management Level

Nivel 2:

SCADA



Supervision Level

Nivel 1:

PLC/PAC



Control Level

Nivel 0:

Sensors,
Actuators,
etc.



Field Level

<https://youtu.be/IAhxYsMi4e8>

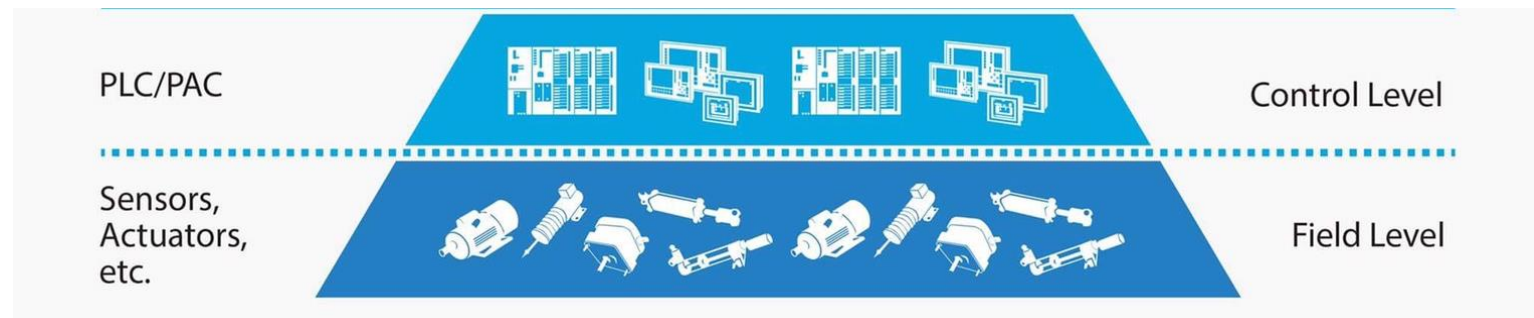


Nivel 0: Dispositivos de campo (proceso industrial)

- ✓ Conjunto de **sensores, actuadores y resto de hardware** que forman parte del equipo productivo.
- ✓ Se comunican mediante los denominados “**protocolos de comunicaciones de campo**” con el nivel de control.

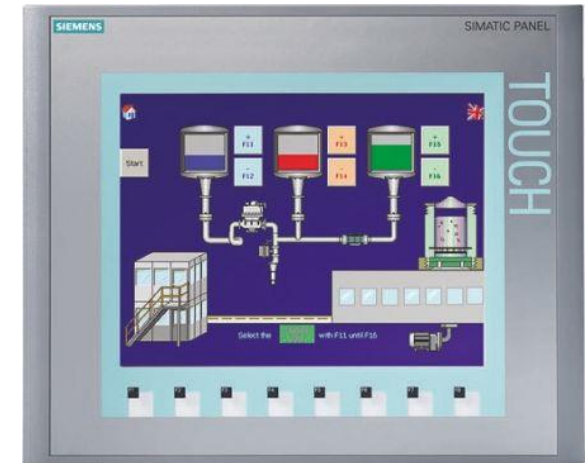
Nivel 1: Control de proceso (PLC)

- ✓ Formado por los **equipos de control** (fundamentalmente PLC, PAC e IPC). Reciben información y envían órdenes al nivel 0. Se comunican con el nivel superior (supervisión)



Nivel 2: Supervisión mediante SCADA

- ✓ Un **SCADA (Supervisory Control And Data Acquisition)**: conjunto de **hardware y software** que permite controlar y supervisar a distancia procesos industriales automatizados, incluyendo gestión de alarmas, generación de informes, control de usuarios y permisos, etc.
- ✓ Está **formado generalmente por uno o varios HMI (Human-Machine Interfaces)**, generalmente en forma de pantallas interactivas.



Nivel 3: Gestión de la Producción

- ✓ **MES** viene de **Manufacturing Execution System** (Sistemas de Ejecución de la Producción): software de control de la producción, documenta la gestión de la planta y rastrea la transformación de materias primas...
- ✓ *Ejemplos: Siemens SIMATIC IT, SAP Manufacturing Execution*

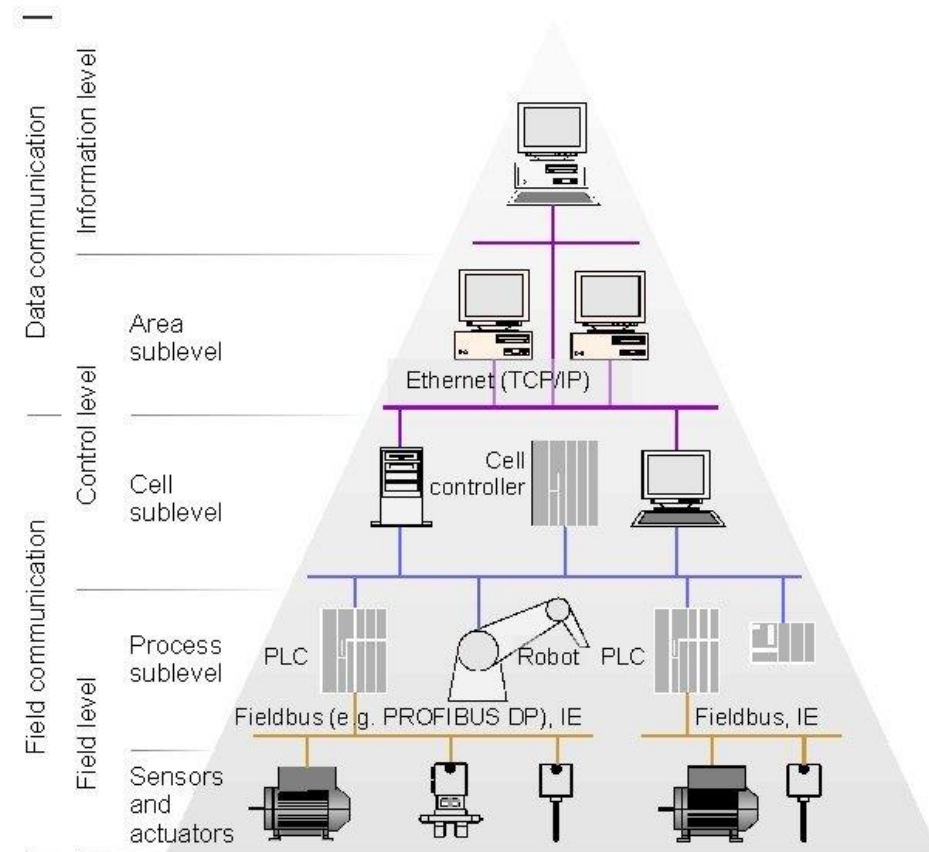
Nivel 4: Administración Empresarial

- ✓ **ERP (Enterprise Resource Planning)**, “Sistema de Planificación de recursos Empresariales”. Se hace cargo de gestionar y monitorizar compras, ventas, distribución, pedidos, recursos humanos...
- ✓ *Ejemplos: SAP ERP, Oracle ERP, Microsoft Dynamics.*



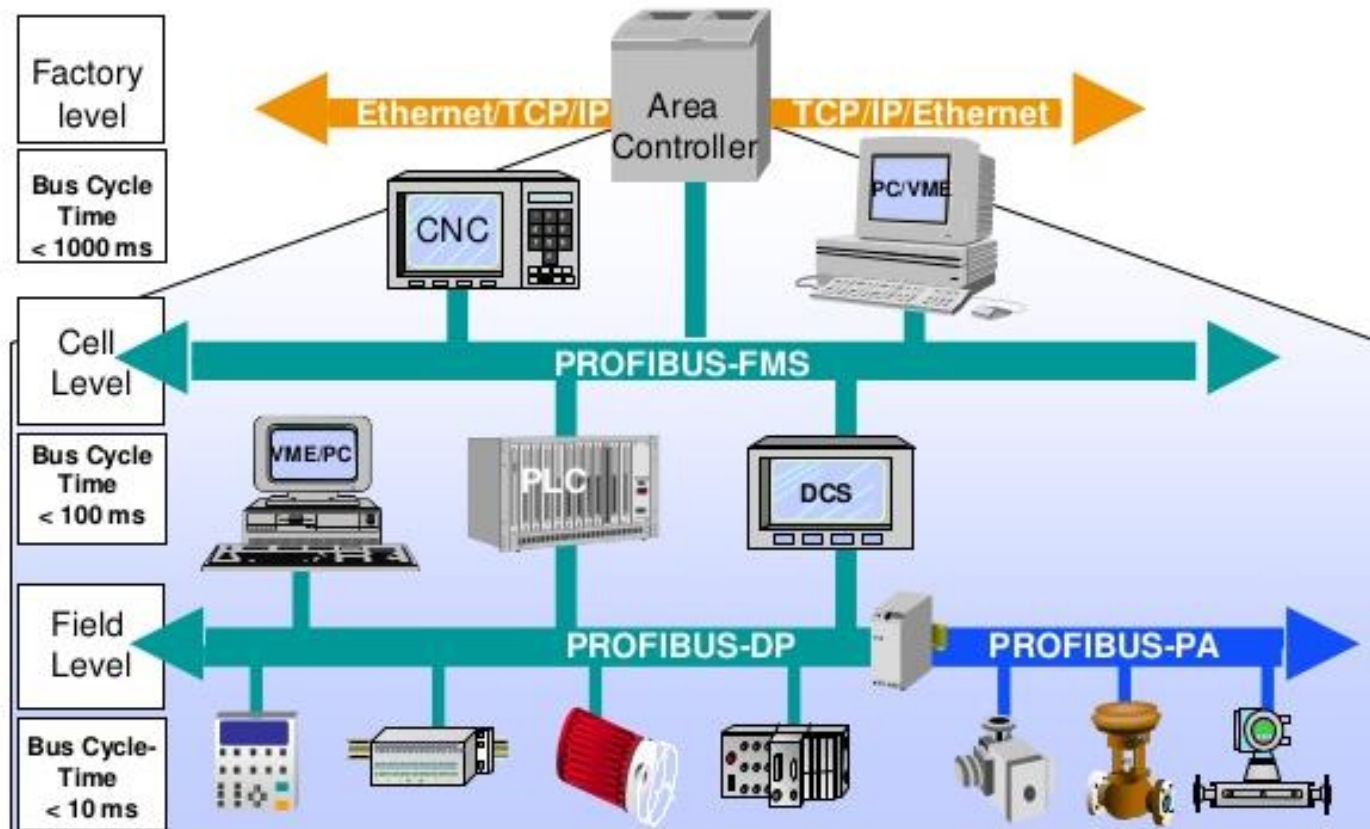
Redes de comunicaciones industriales

- ✓ En la pirámide, las comunicaciones se realizan tanto **horizontalmente** (dentro de cada nivel) como **verticalmente** (entre varios niveles jerárquicos)
- ✓ Cada comunicación tiene unas **necesidades diferentes** (volumen de datos, tiempo de reacción, seguridad). **Tres niveles o capas:**
 1. **Nivel de campo.** Sensores/ actuadores y proceso (PLC)
 2. **Nivel de control** (celda ó área)
 3. **Nivel de planta (factory level):** información
- ✓ No hay un tipo de red que cubra todas las necesidades. **Soluciones adaptadas** a cada nivel



Ejemplo de red de comunicación industrial

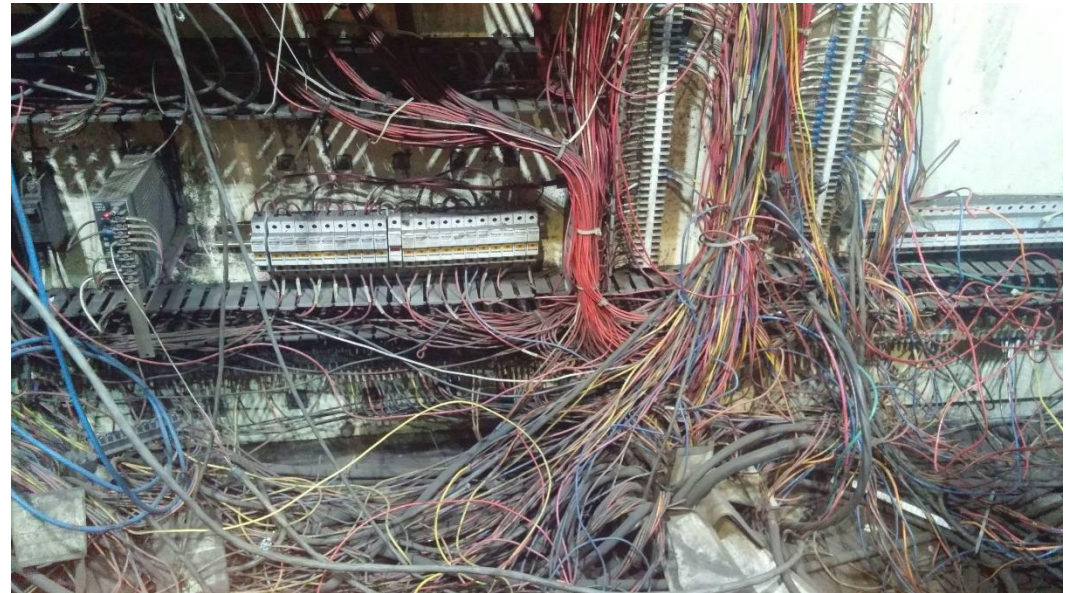
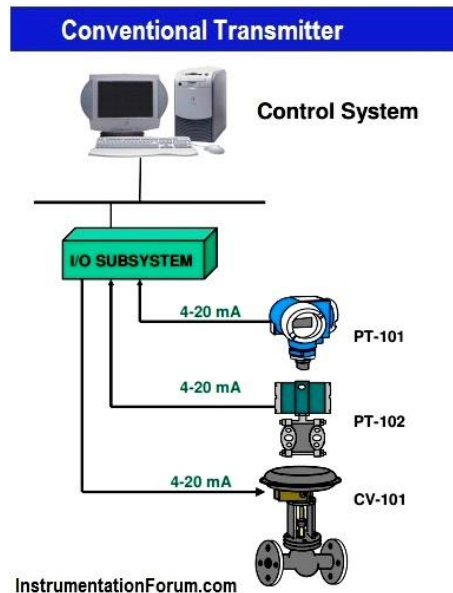
- ✓ Cada uno de los niveles utiliza un protocolo adecuado a sus necesidades:



Comunicaciones tradicionales (punto a punto)

Tradicionalmente se han usado **conexiones eléctricas de cada sensor/actuador** al sistema E/S del PLC de tipo **punto a punto** (E/S digitales o analógicas mediante lazos de corriente 4-20 mA)

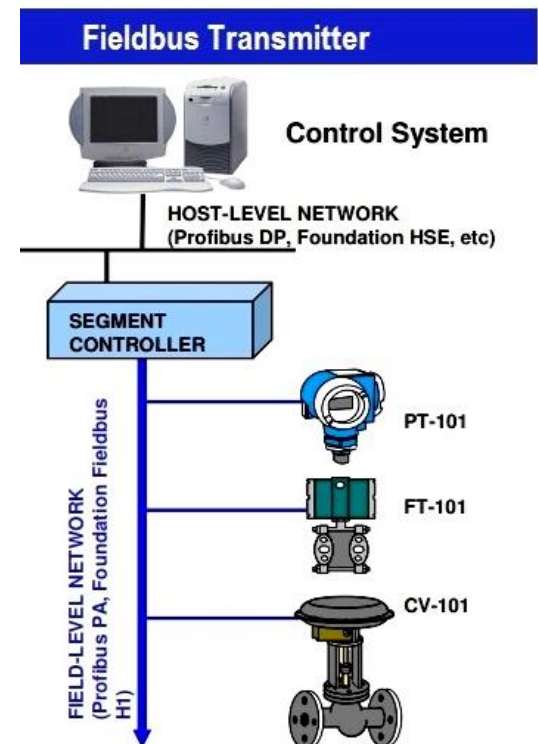
- ✗ Generan **muchísimo cableado** en sistemas de tamaño elevado
- ✗ El **diagnóstico de fallos** es muy tedioso



Buses de campo (Fieldbus)

Familia de protocolos de red industriales (IEC 61158) para sustituir las conexiones punto a punto. **Multipunto y bidireccionales**. Los datos se digitalizan y se envían a través de un bus serie (cable común).

- ✓ Reducen **costes de instalación** (menos cableado y elementos intermedios).
- ✓ Son más **eficientes, robustos y escalables**.



Ejemplo: PROFIBUS

Bus de campo de tipo abierto creado en 1987-1990 por ABB, AEG, Bosch, Honeywell, Moeller, Landis & Gyr, Phoenix Contact, Rheinmetall, RMP, Sauter-cumulus, Siemens y varios institutos alemanes de investigación.



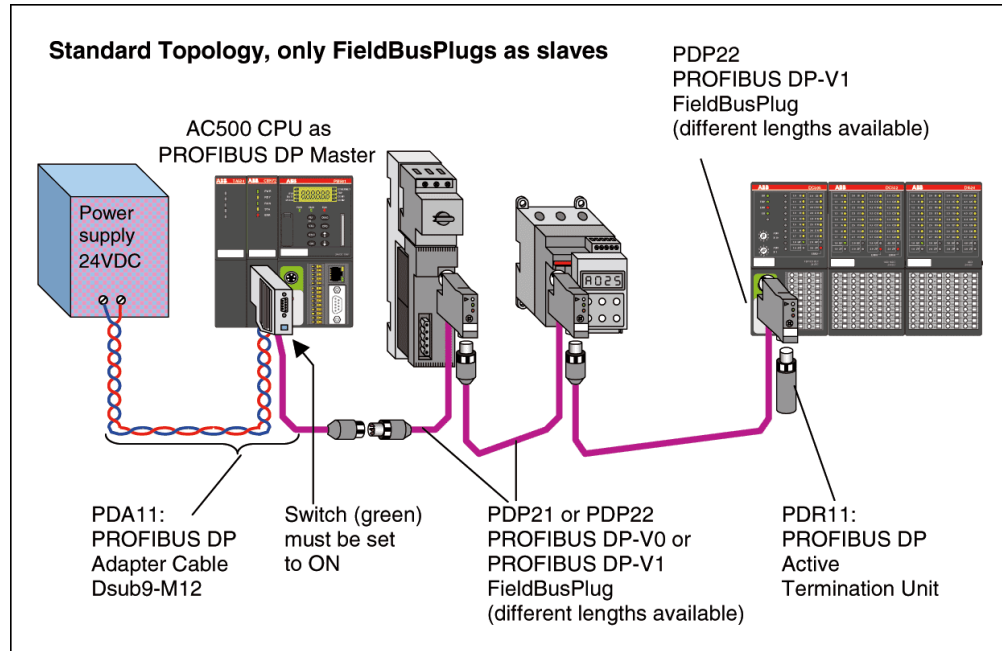
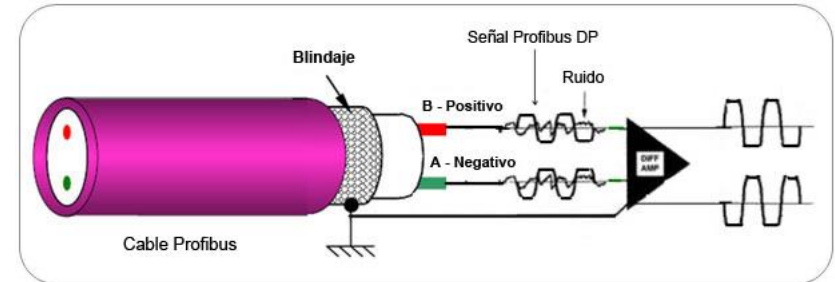
Ha sido el más utilizado en el mundo (especialmente en Europa)

Formado por tres variantes compatibles entre sí:

1. **PROFIBUS DP (Decentralized Periphery):** optimizado para ofrecer mayor velocidad (hasta 12 Mb/s) y tiempo de respuesta. Usado a nivel de campo.
2. **PROFIBUS FMS (Filed Message Specification):** Mismas velocidades que DP, peores tiempos. Multimáster. Usado en el nivel de control.
3. **PROFIBUS PA (Process Automation):** nivel de campo. Lento (síncrono). Ofrece suministro de energía a través del bus: seguridad en áreas peligrosas (por ejemplo con riesgo de explosión).

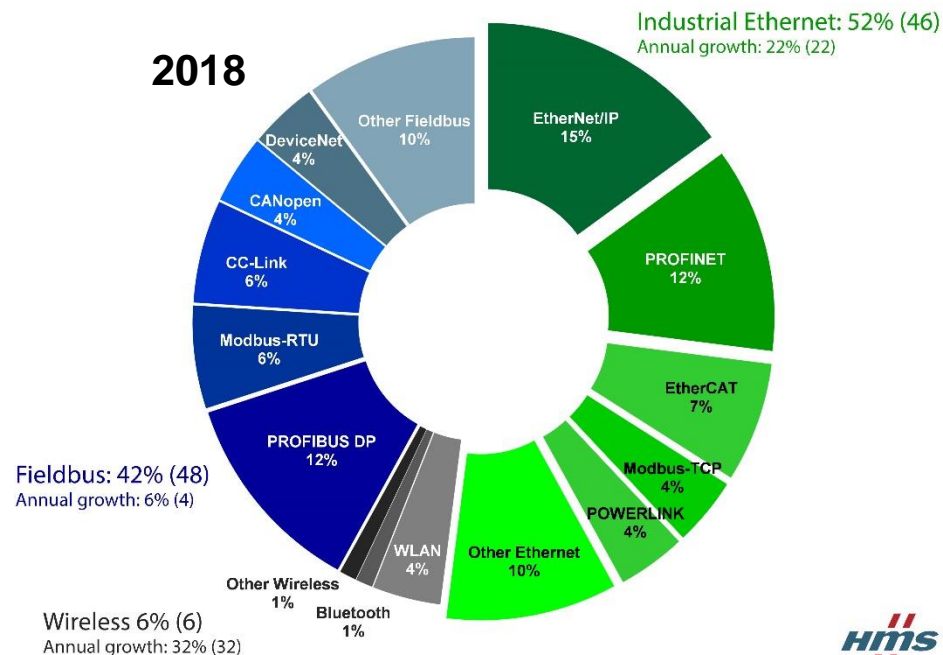
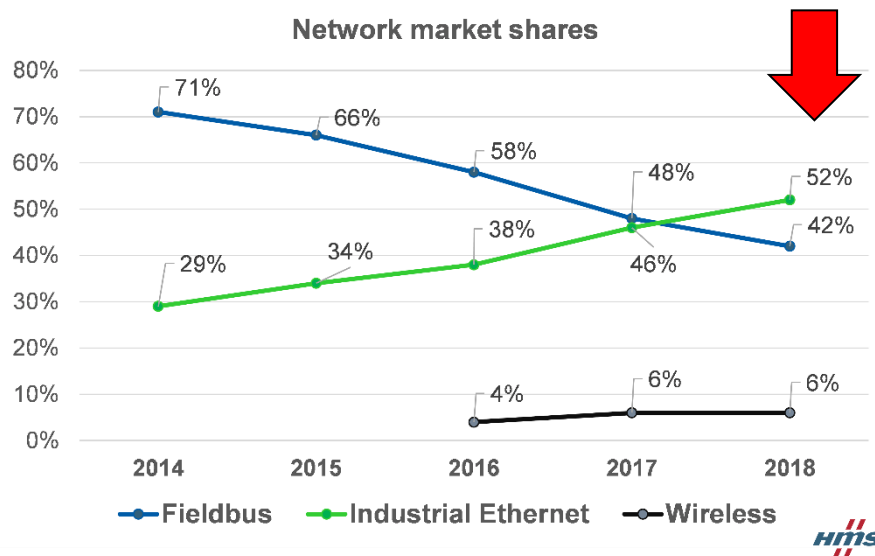
Ejemplo: PROFIBUS DP

- Es el más utilizado de los tres
- Configuración típica con un dispositivo maestro y varios esclavos.
- Topología en bus lineal o tipo árbol
- Basado en capa física RS-485



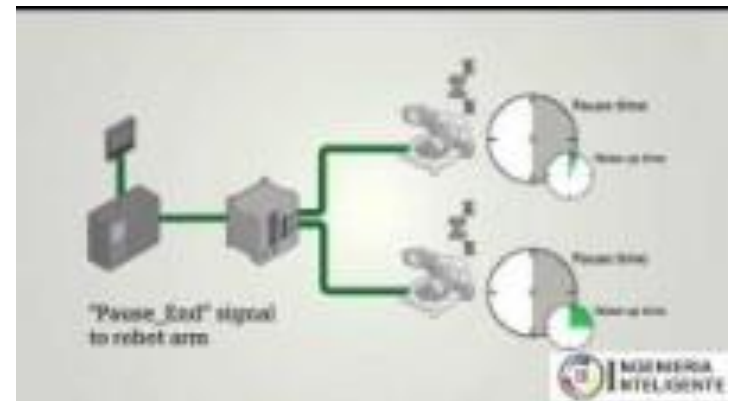
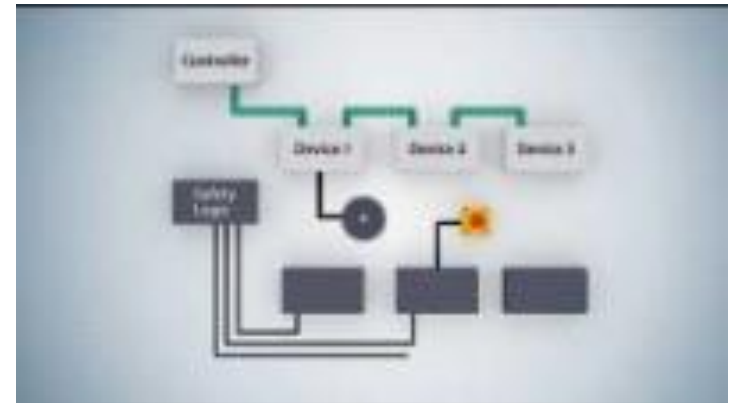
Buses de campo vs Industrial (real-time) Ethernet

- ✓ A nivel de red de campo, los protocolos de bus de campo también se están viendo sustituidos por **protocolos de comunicación tipo Ethernet Industrial** (PROFINET, EtherCAT, EtherNet/IP)
- ✓ Los protocolos **Ethernet Industrial** ofrecen **mayor velocidad, pueden ser inalámbricos y más flexibles** (más opciones de topología de red, direcciones...)



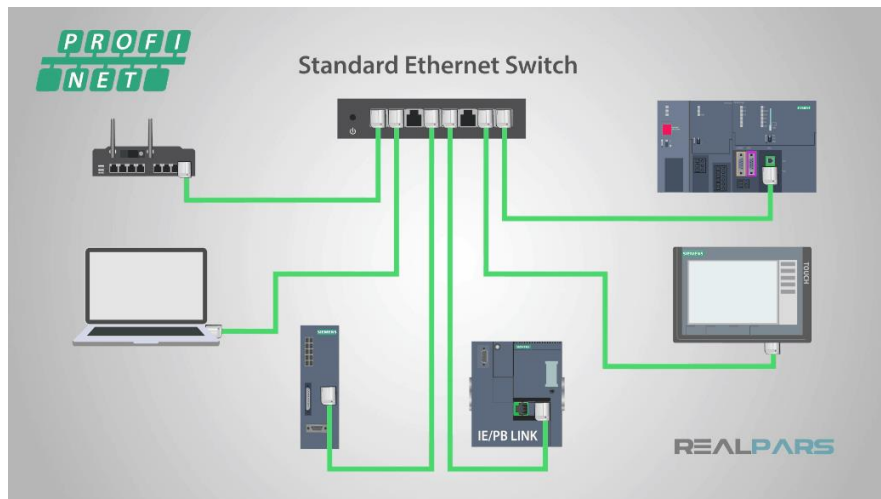
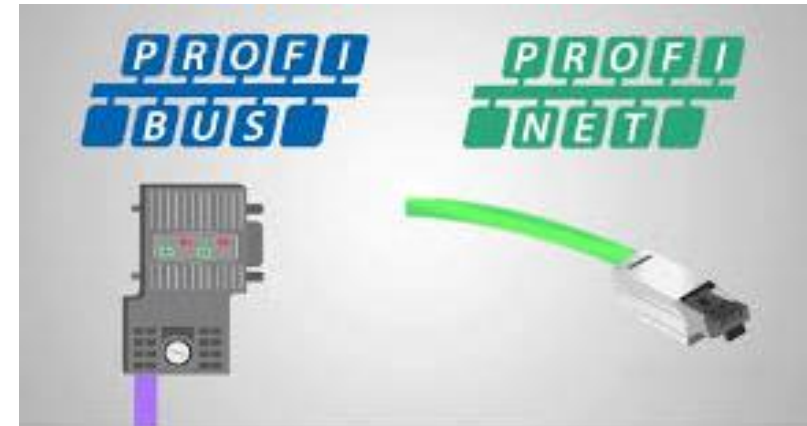
PROFINET

- ✓ **PROFINET** es un estándar abierto de Ethernet Industrial de la asociación **PROFIBUS-PROFINET Internacional** según **IEC 61784-2** (*Performance indicators supporting classification schemes for Real-Time Ethernet (RTE)*).
- ✓ **Protocolo de tiempo real**: permite tiempos de ciclo muy rápidos, de tan sólo **31,25µs** (32.000 muestras por segundo) con PROFINET v2.3.
- ✓ Permite transmitir **datos de seguridad** (safety) mediante **PROFISAFE**, y datos y comandos para el **ahorro y control energético** mediante **PROFIENERGY**.



PROFINET (I.E.) vs PROFIBUS (bus de campo)

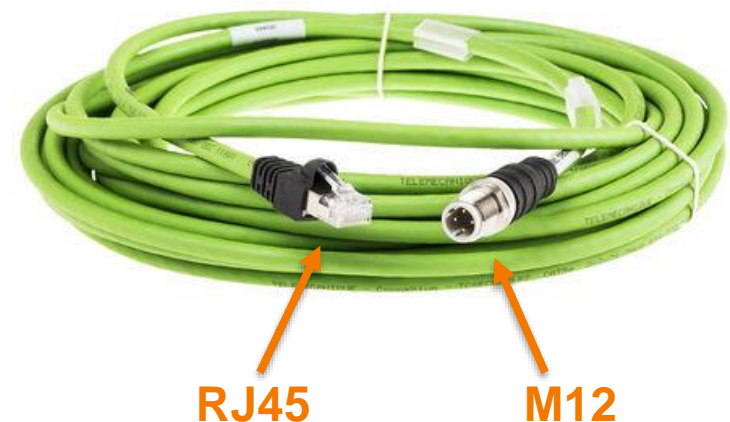
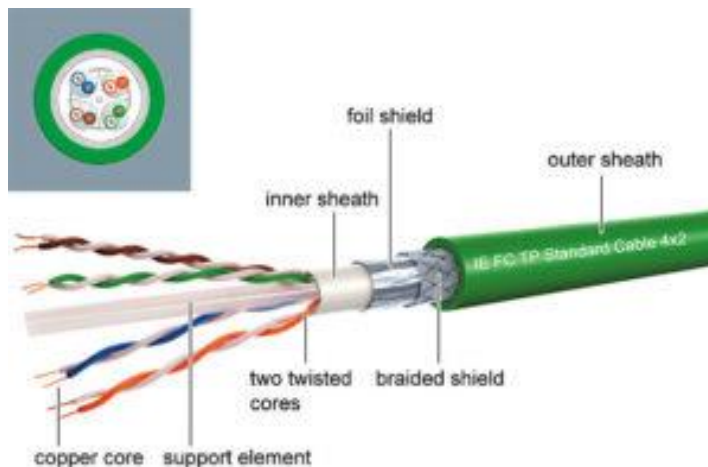
- ✓ **PROFINET** puede usarse con switches de red convencionales (capa física Ethernet), lo que facilita la conexión con PCs
- ✓ **PROFINET** proporciona velocidades mucho mayores y espacio de direccionamiento ilimitado



	Profibus	Profinet
Organización	PI	PI
Definición de hardware	Archivos GSD	Archivos GSD
Perfiles de aplicación	Iguales	Iguales
Capa física	RS 485	Ethernet
Velocidad de tx de datos	12 Mbit/s	1 Gbit/s o 100 Mbit/s
Telegrama	244 bytes	1.440 bytes (cíclico)
Espacio de direccionamiento	126	Ilimitado
Tecnología	Maestro/esclavo	Proveedor/consumidor
Wireless	Posible	IEEE 802.11, IS-1
Motion	32 ejes	>150 ejes
Machine-to-machine	No	Si
Integración vertical	No	Si
Conectividad	PA + otros	Muchos buses

PROFINET: cableado y conectores

- ✓ **PROFINET** usa cableado Ethernet con blindaje (funda de aluminio y malla trenzada) para proteger contra interferencias EMC. **Color verde**
- ✓ Se usan conectores **RJ45 industriales** (robusto, apantallado, con protección IP). A veces también conectores **M12**



¿Qué es un PLC? Norma internacional IEC 61131

- El **IEC 61131** es un conjunto de normas e informes técnicos publicados por la **Comisión Electrotécnica Internacional** con el objetivo de **estandarizar** los Autómatas o Controladores Lógicos Programables (PLC) (<https://webstore.iec.ch/publication/4550>)
- La norma está formada por 10 partes: 8 secciones y 2 “technical reports (TR)”:



Project	Title
61131- 1, Ed 2.0	General information, 2003-05
61131- 2, Ed 3.0	Equipment requirements and tests, 2007-07
61131- 3, Ed 3.0	Programming languages (Currently CDV - Committee Draft for Voting)
61131- 4, Ed 2.0	User guidelines (TR), 2004-07
61131- 5, Ed 1.0	Communications, 2000-11
61131- 6, Ed 1.0	Functional safety for PLC (Currently CDV - Committee Draft for Voting)
61131- 7, Ed 1.0	Fuzzy control programming, 2000-08
61131- 8, Ed 2.0	Guidelines applic. & implem. progr. languages (TR), 2003-09
61131- 9, Ed 1.0	Single-drop digital communication interface for small sensors and actuators (SDCI) aka “IO-Link” (Currently CD - Committee Draft)

61131-10 Ed 1.0 Programmable controllers - Part 10: PLC open XML exchange format **(2019)**

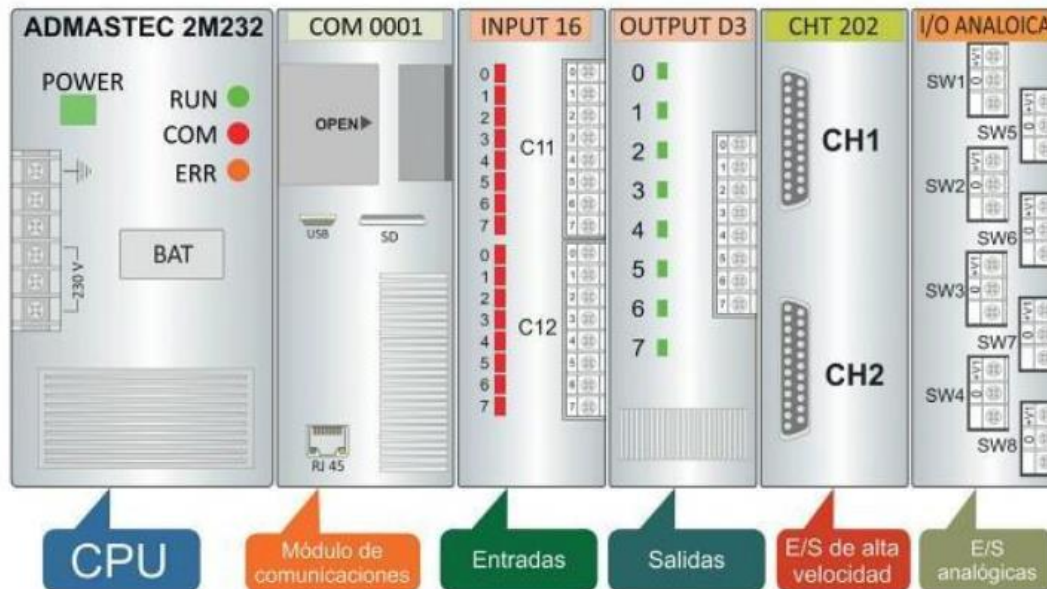
Definición de PLC según IEC 61131-1

- Un **PLC** es una **máquina electrónica programable**:
 - Diseñada para ser utilizada en un **entorno industrial** (hostil)
 - Que utiliza una memoria programable para el almacenamiento interno de instrucciones orientadas al usuario
 - Proporciona soluciones específicas tales como funciones lógicas, secuencias, temporizaciones, recuentos y funciones aritméticas
 - Cuyo fin es **controlar mediante entradas y salidas, digitales y analógicas diversos tipos de máquinas o procesos**



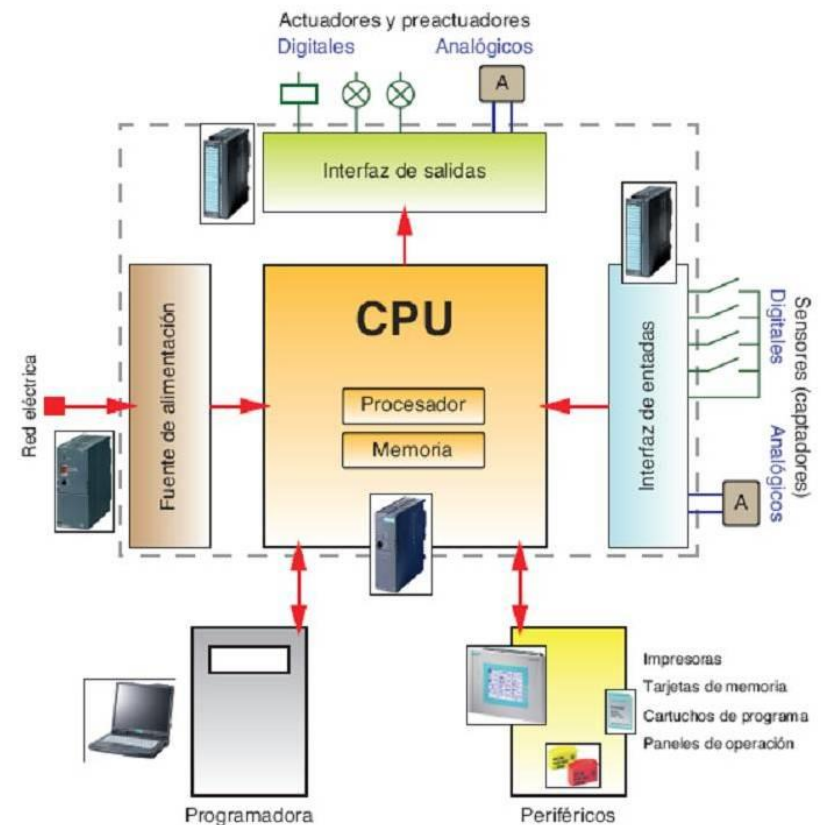
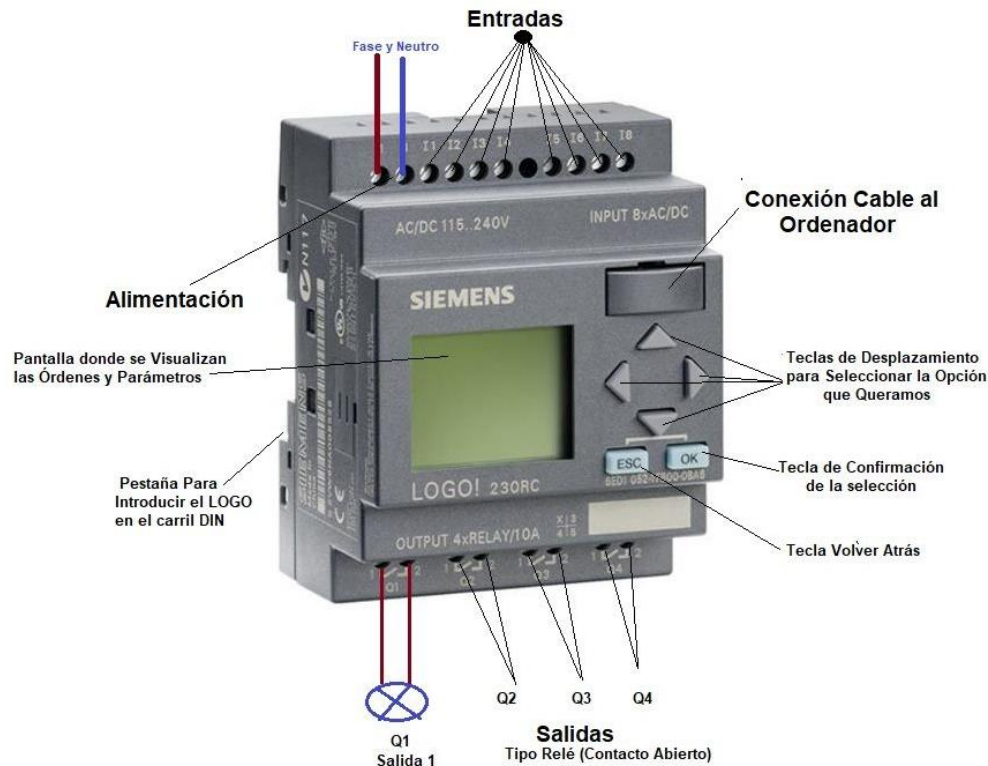
Componentes típicos de un PLC

- CPU (*Unidad Central de Proceso*) + Memoria
- **Módulo de entradas** (digitales, analógicas).
- **Módulo de salidas** (digitales, analógicas, por relé)
- Módulo de comunicaciones.
- Fuente de alimentación.
- **(Opcional)** Periféricos: panel de operación, tarjeta externa de memoria



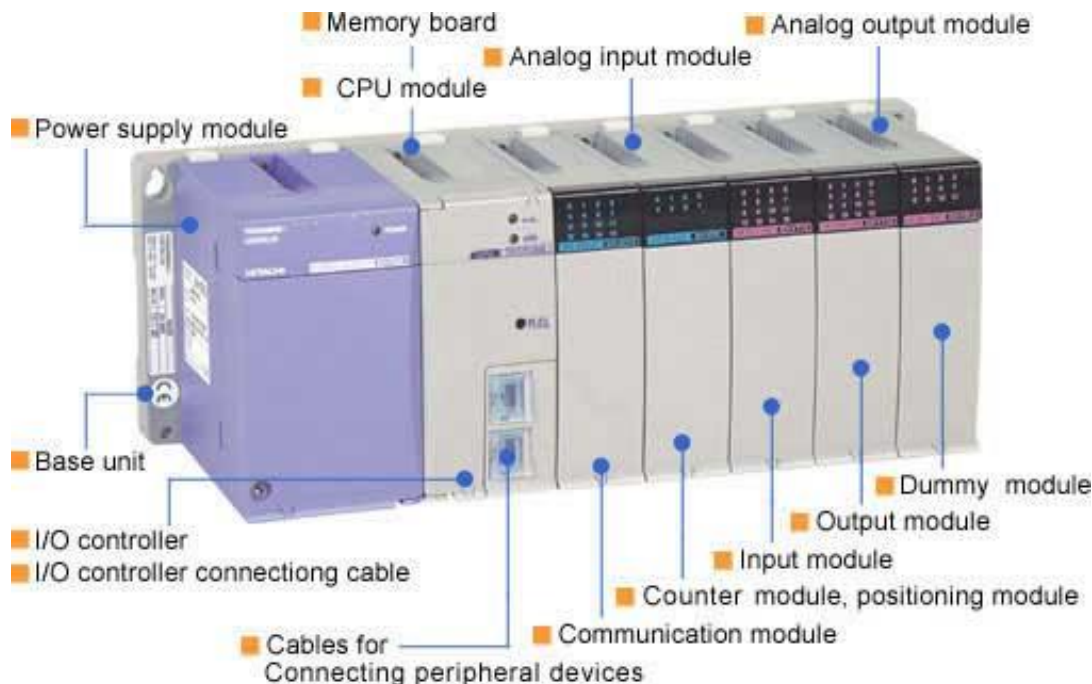
Tipos de PLC según su estructura (I)

- ✓ **PLC Compactos:** También llamados integrados o de E/S fija. CPU, E/S y comunicaciones en el mismo dispositivo. No se pueden expandir.



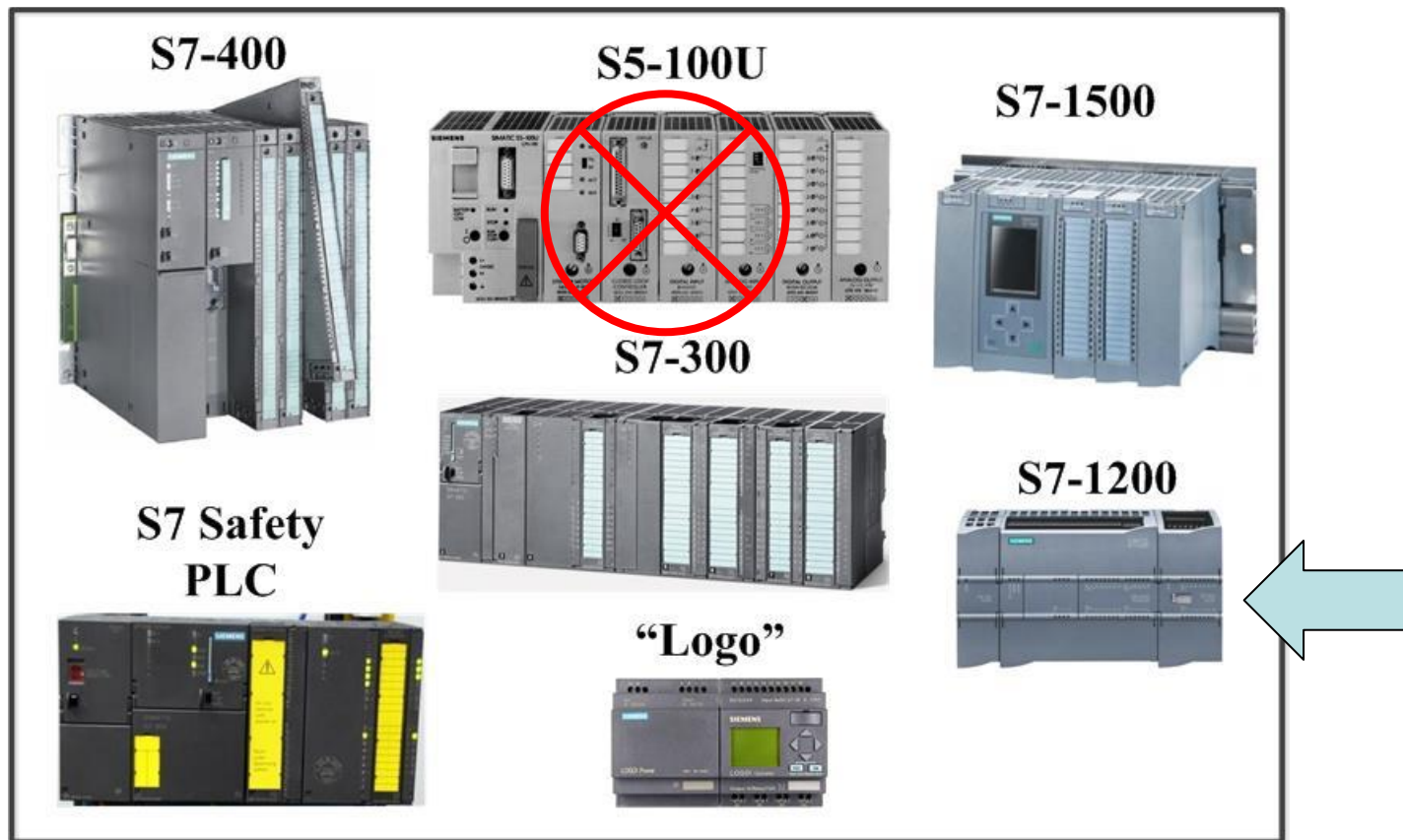
Tipos de PLC según su estructura (II)

- ✓ **PLC Modulares:** Admiten módulos conectables a la CPU que permiten ampliar las E/S o las comunicaciones. **Escalable** según necesidad. Mantenimiento más fácil. A veces se montan en rack. Los módulos se pueden conectar de manera **distribuida** (E/S descentralizadas)



Gama actual de PLC SIEMENS

SIEMENS
SIMATIC



<https://new.siemens.com/es/es/productos/automatizacion/sistemas/simatic.html>

Ejemplo: PLC Siemens SIMATIC S7-1215C AC/DC/RLY



6ES7215-1BG40-0XB0

SIMATIC S7-1200, CPU 1215C,
COMPACT CPU, AC/DC/RELAY,

2 PROFINET PORT,

ONBOARD I/O: 14 DI 24V DC;

10 DO RELAY 2A, 2 AI 0-10V DC,

2 AO 0-20MA DC.

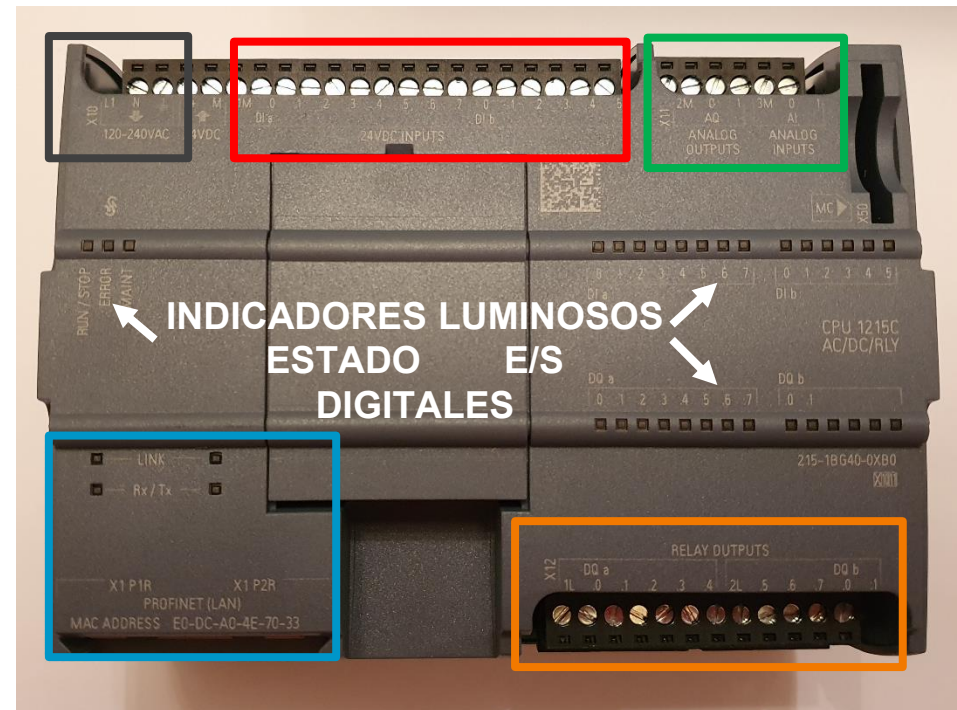
POWER SUPPLY: AC 85 - 264 V AC AT 47 - 63 HZ,

PROGRAM/DATA MEMORY: 100 KB

ALIMENTACIÓN
ELÉCTRICA AC

14 ENTRADAS
DIGITALES 24V DC

2 ENTRADAS +
2 SALIDAS
ANALÓGICAS

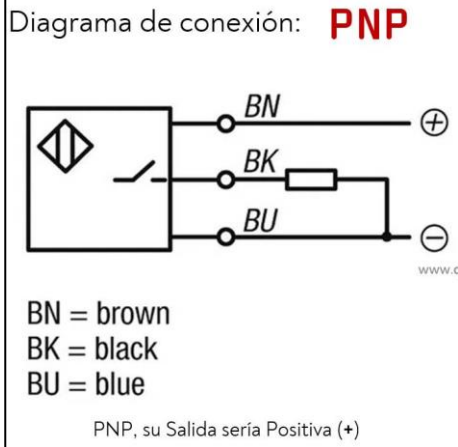


INDICADORES LUMINOSOS
ESTADO E/S
DIGITALES

2 PUERTOS DE
COMUNICACIONES
PROFINET

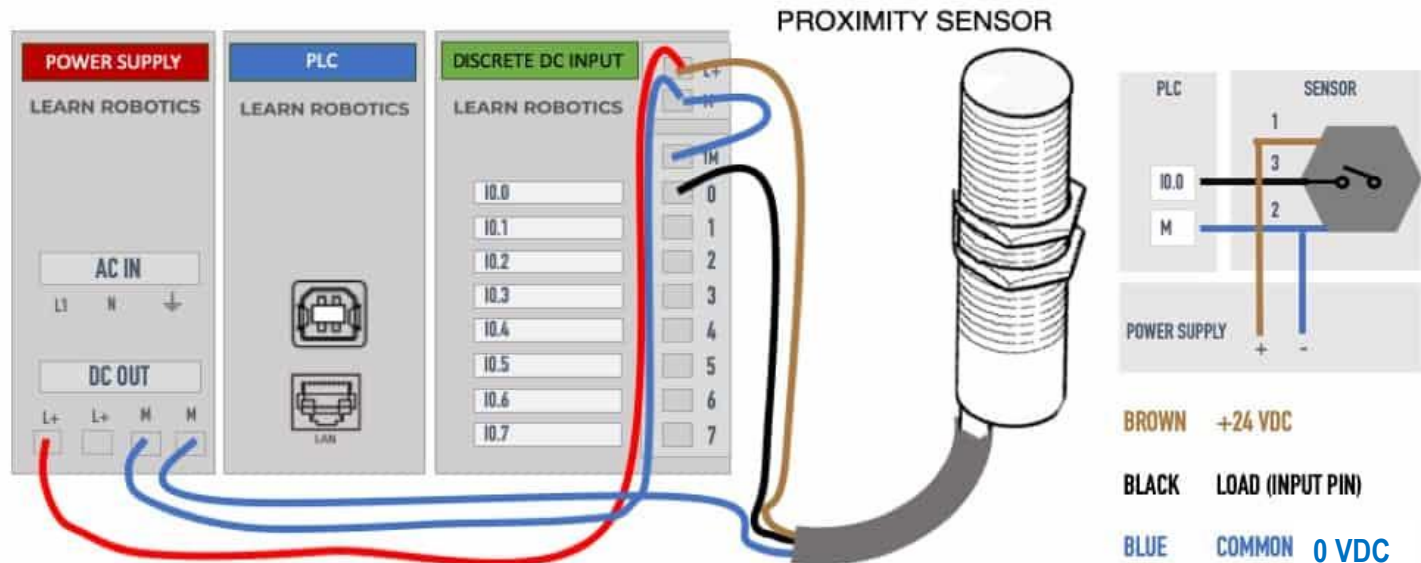
10 SALIDAS
DIGITALES A
RELÉ (HASTA 2 A)

Ejemplo: Cableado de un sensor de proximidad PNP

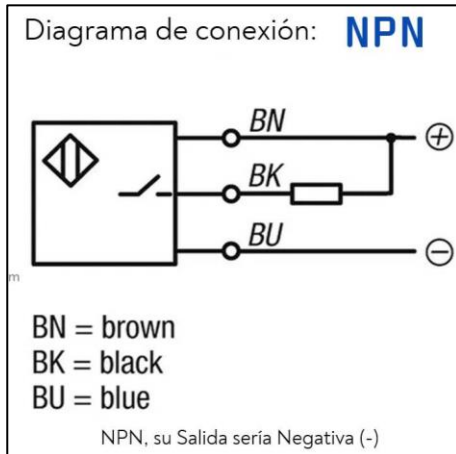


- Un sensor de proximidad en configuración PNP da **salida positiva** al efectuar una detección
- Se cablea la **salida a una entrada digital** del PLC, y el **negativo del sensor al común** de las entradas del PLC

PNP WIRING

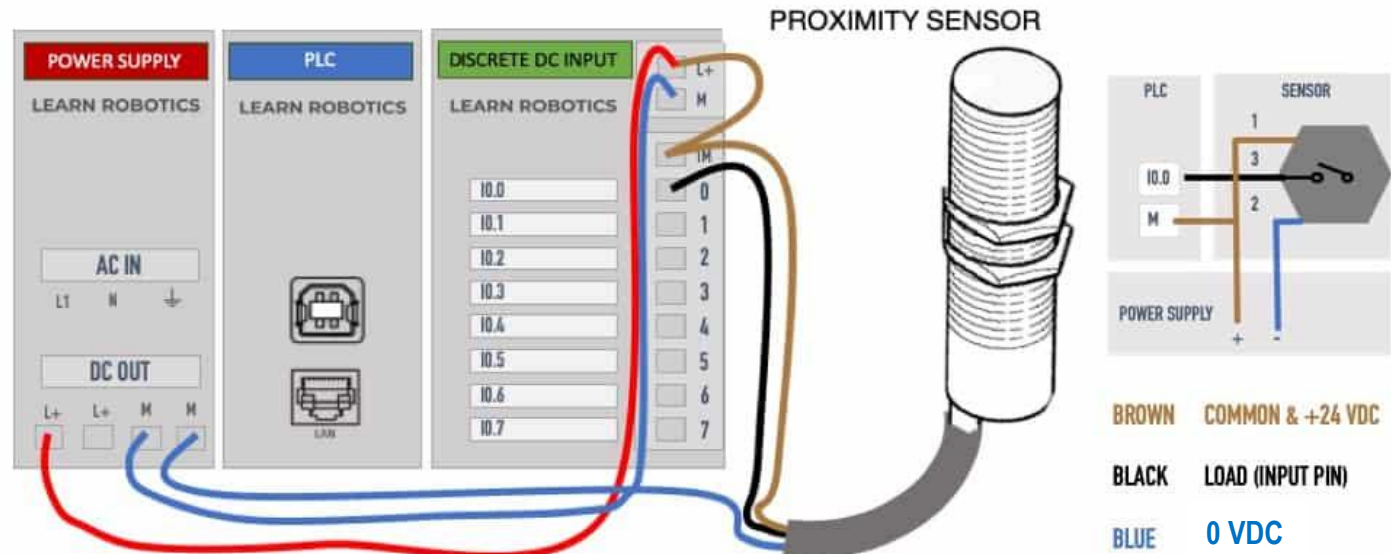


Ejemplo: Cableado de un sensor de proximidad NPN



- Un sensor de proximidad en configuración PNP tiene **salida negativa** al efectuar una detección
- Se cablea la **salida a una entrada digital del PLC**, y el **positivo del sensor al común** de las entradas del PLC

NPN WIRING



Alternativas al PLC: PC Industrial (I-PC)

Un **PC Industrial (I-PC)**:

- Es un PC con resistencia a factores externos (protecciones IP frente a temperatura, atmósferas industriales)
- Típicamente modular y de dimensiones reducidas (para rack)
- Pueden usar un **sistema operativo convencional** (Windows, Linux..)
- Eso les hace **compatibles con otras aplicaciones de PC** (programación, BBDD, visualización)
- Se pueden conectar a **E/S descentralizadas** mediante interfaces de comunicaciones industriales

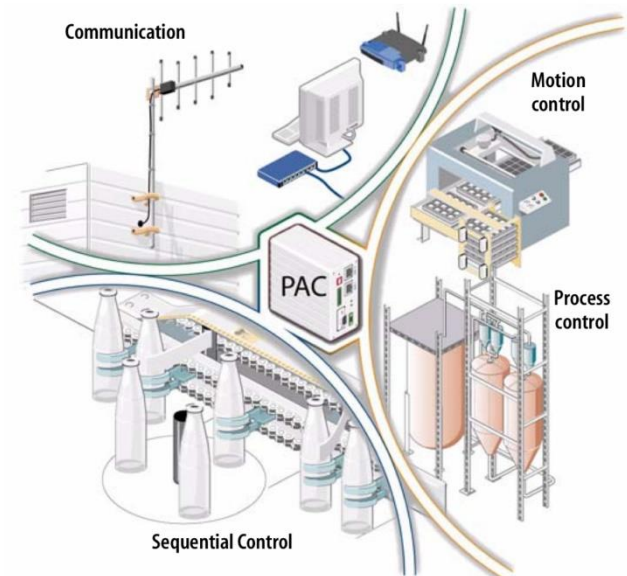


<https://new.siemens.com/global/en/products/automation/pc-based.html>

Alternativas al PLC: PAC

Un **Programmable Automation Controller (PAC)**:

- Se puede entender como un **equipo de mayores prestaciones** que combina la funcionalidad del **PLC** y la capacidad de proceso y versatilidad del **PC**
- Se pueden **programar como PLC** (LD, ST, etc) y **como un PC** (C/C++, etc)
- Permiten **controlar ejes de movimiento** (controlador de robot)
- Pueden almacenar **grandes cantidades de datos**
- Contienen **procesadores más avanzados** y, a veces, matrices de puertas programables (FPGA)

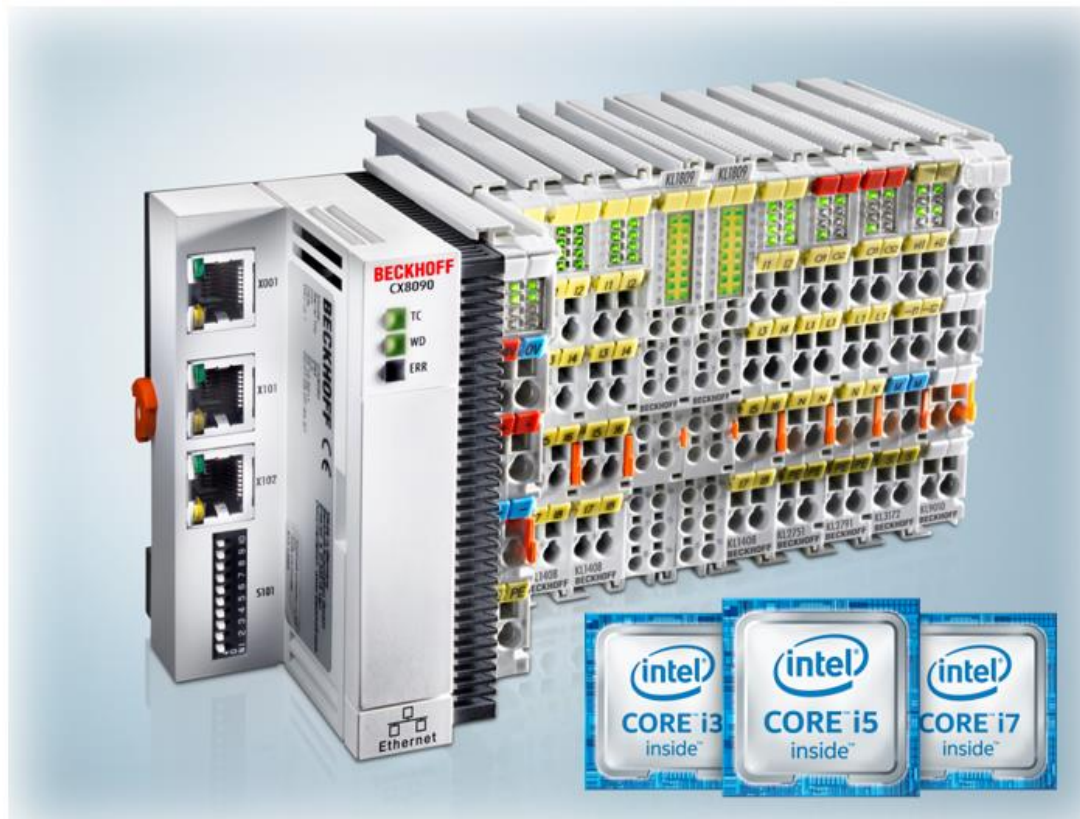


PAC Component Compare

	PLC	PAC	PC
High Shock and Vibration Ratings	☺	☺	☹
Safety and Haz Loc Certifications	☺	☺	☹
Industrial Temperature Range	☺	☺	☹
Real-Time OS	☺	☺	☹
Redundant Power Supply Inputs	☺	☺	☹
Full Windows Driver Set	☹	☹	☺
Floating Point Processor	☹	☺	☺
Non-Volatile Data Storage	☹	☺	☺
Full Featured Programming SW	☹	☺	☺
Ethernet Connectivity with Web	☹	☺	☺

Ejemplo: PAC BECKHOFF CX8090

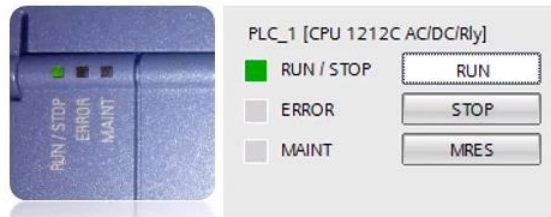
- Consta de un PLC (con sistema de E/S) y una CPU potente independiente, con almacenamiento. A veces se llaman “embedded PCs”.



<https://www.beckhoff.com/es-es/products/ipc/embedded-pcs/cx8000-arm9/cx8090.html>

Modos de operación de un PLC

The CPU has three modes of operation: STOP mode, STARTUP mode, and RUN mode. Status LEDs on the front of the CPU indicate the current mode of operation



In RUN mode, the scan cycle is executed repeatedly. Interrupt events can occur and be processed at any point within the program cycle phase.



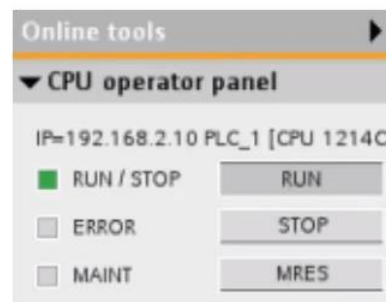
In STOP mode, the CPU is not executing the program, and you can download a project.



In STARTUP mode, the CPU executes any startup logic (if present). Interrupt events are not processed during the startup mode.



You can download your project only when the CPU is in STOP mode.



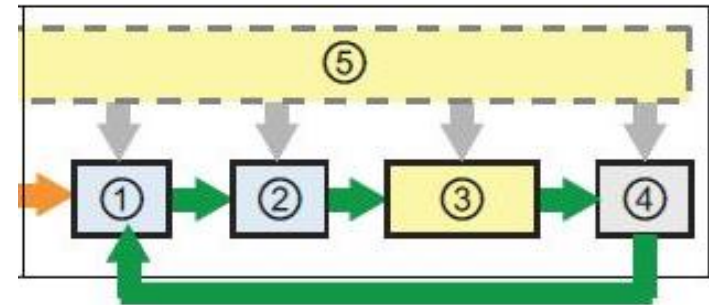
Use the button on the operator panel to change the operating mode (STOP or RUN). The operator panel also provides an MRES button for resetting the memory.



Ciclo de trabajo (scan) de un PLC. Watchdog

En modo RUN, el PLC ejecuta repetidamente un ciclo de operaciones, entre las que se incluye la ejecución del programa de usuario:

1. Los datos de la memoria de salidas “Q” se escriben en las salidas físicas del PLC
2. Los valores de las entradas físicas se copian en la memoria de entradas “I”.
3. Se lee el programa de usuario (bloques de operación – OB) y se ejecutan todas sus funciones.
4. Se realiza el autodiagnóstico del PLC.
5. Las alarmas y comunicaciones se procesan en cualquier parte del ciclo.



- ✓ El tiempo que tarda un ciclo en ejecutarse se denomina tiempo de ciclo. Depende del tamaño del programa y de la potencia de la CPU del PLC.
- ✓ El watchdog (tiempo de vigilancia de ciclo) es un mecanismo (fijo o programable) que vigila que el tiempo de ciclo esté por debajo de un valor

Mapa de memoria de un PLC. Tipos

Cada PLC tiene un mapa de memoria, que se puede dividir en dos categorías principales: **tabla de datos** y **programas de usuario**

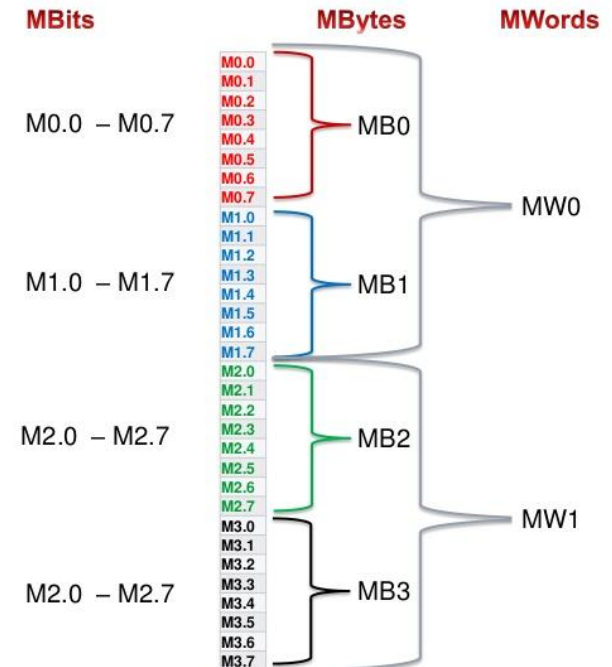
La tabla de datos se divide en 5 tipos de memoria:

- **Memoria de salida (Q):** la CPU copia el contenido de esta memoria en las salidas físicas al inicio del ciclo
- **Memoria de entrada (I):** la CPU copia el estado de las entradas físicas a esta memoria al inicio del ciclo
- **Memoria de bits o Marcas (M):** memoria general para almacenar datos del programa de usuario. Accesible desde todos los programas. De tamaño limitado
- **Memoria temporal (L):** para el almacenamiento de datos temporales de ejecución de ciertos bloques de función (FB).
- **Memoria de Bloques de Datos (DB):** memoria general para almacenar datos de fuera de la memoria de Marcas M. Cada FB tiene asociado un DB.

Direccionamiento de memoria un PLC

Se puede **acceder a los datos** en memoria de diferentes maneras (bit, Byte, palabra)

Componente	Elemento	Descripción
Símbolo	%	El símbolo de porcentaje siempre precede a una dirección interna.
Tipo de objeto	I	Entrada digital (objeto de bit).
	Q	Salida digital (objeto de bit).
	IW	Valor de la entrada analógica (objeto de palabra).
	QW	Valor de la salida analógica (objeto de palabra).
	IWS	Estado de canal de entrada analógica (objeto de palabra).
Número de módulo	y	Canal de E/S en un módulo conectado directamente con el controlador.
	Z	Número de canal de E/S del módulo.



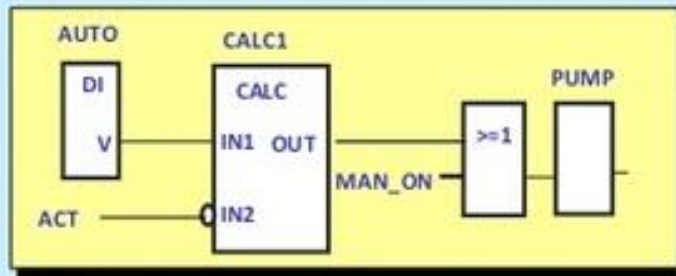
Ejemplos:

Objeto de E/S	Descripción
%I0.5	Número de canal 5 de la entrada digital del controlador del módulo 0.
%Q3.4	Número de canal 4 de la salida digital del módulo 3.
%IW0.1	Entrada analógica 1 en el módulo 0.
%QW2.1	Salida analógica 1 en el módulo 2.
%QWS1.1	Estado de canal de salida analógica del canal de salida 1 del módulo 1.

SOFTWARE: Lenguajes de programación de PLC

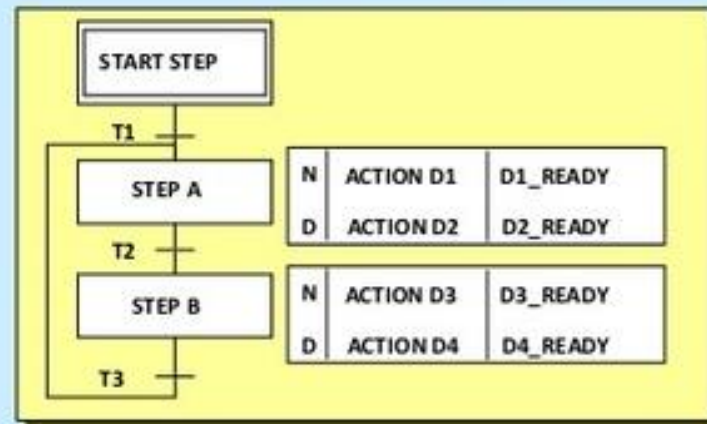
LA IEC 61131-3 estandariza **cinco lenguajes**, tres gráficos y dos textuales:

Function Block Diagram (FBD)

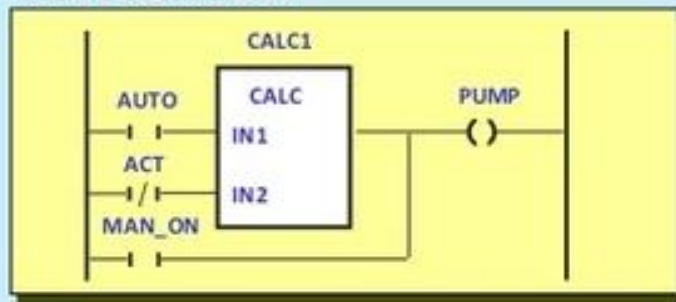


graphical languages

Sequential Flow Chart (SFC)



Ladder Diagram (LD)



textual languages

Structured Text (ST)

```

VAR CONSTANT X : REAL := 53.8 ;
Z : REAL; END_VAR
VAR aFB, bFB : FB_type; END_VAR

bFB(A:=1, B:='OK');
Z := X - INT_TO_REAL (bFB.OUT1);
IF Z>57.0 THEN aFB(A:=0, B:="ERR");
ELSE aFB(A:=1, B:="Z is OK");
END_IF
  
```

Instruction List (IL)

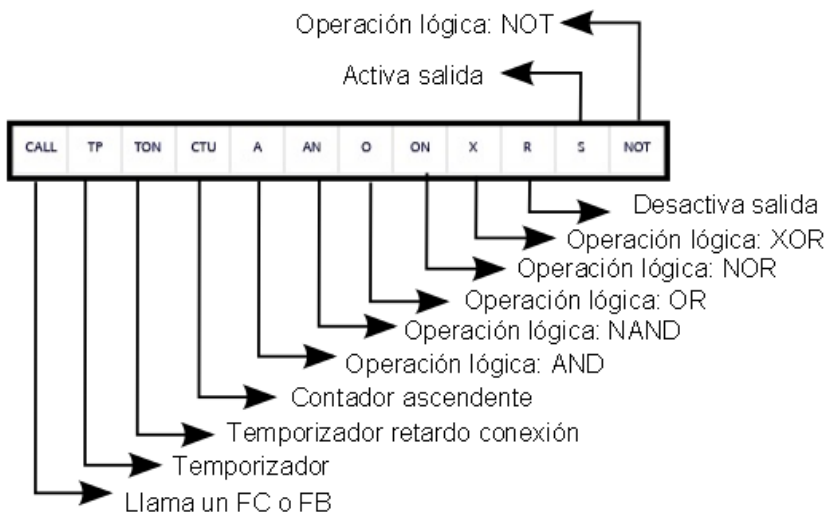
```

A: LD    %IX1 (* PUSH BUTTON *)
ANDN    %MX5 (* NOT INHIBITED *)
ST      %QX2 (* FAN ON *)
  
```


Lenguajes textuales de programación de PLC (I)

Lista de Instrucciones (IL):

- ✓ Lenguaje de **bajo nivel** similar al lenguaje **ensamblador**.
- ✓ El flujo de programa se controla con instrucciones de salto y llamadas a subrutinas. Para procesos complejos se hace grande y difícil de depurar.
- ✓ A veces, los fabricantes lo adaptan a sus necesidades. **SIEMENS** lo denomina **STL** (*Statement List*) o **AWL** (del alemán *Anweisungsliste*)



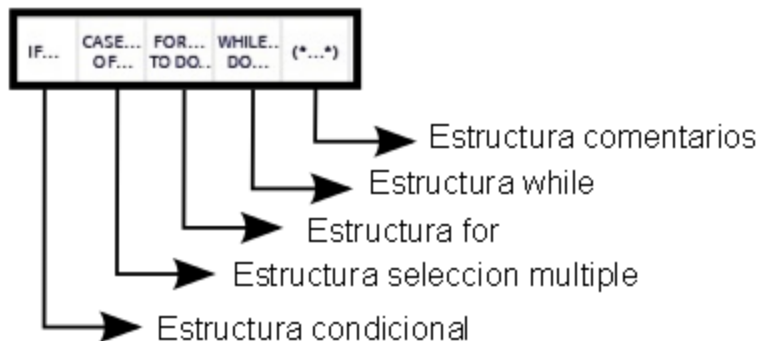
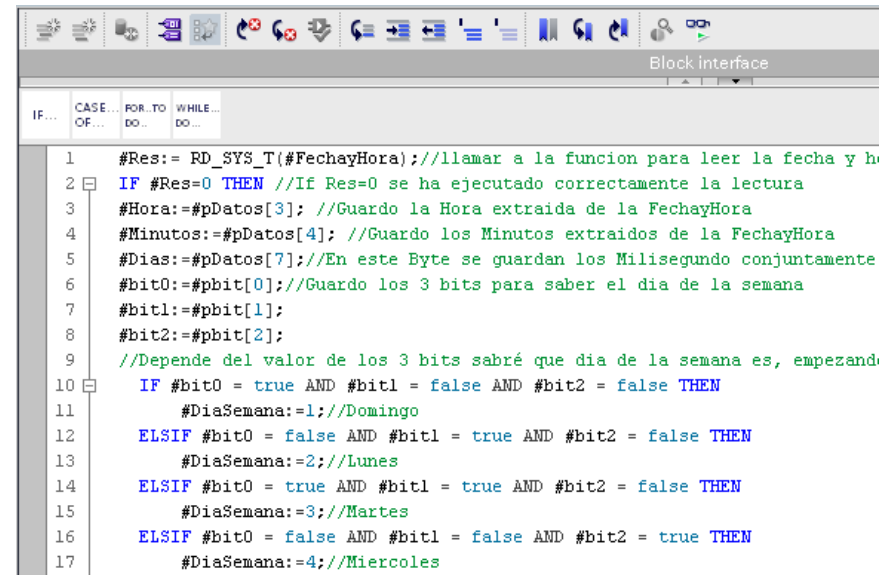
```

Segmento 1: .....
Comentario
1      A      "M1_START"
2      =      %L20.0
3      BLD    103
4      A      "M1_STOP"
5      =      %L20.1
6      BLD    103
7      CALL   "Bloque_1", "Bloque_1_DB"
8      ARRANQUE :=%L20.0
9      PARO     :=%L20.1
10     SALIDA  := "M1_ON"
11     NOP 0
  
```

Lenguajes textuales de programación de PLC (II)

Texto Estructurado (ST).

- ✓ Lenguaje de **alto nivel basado** en Pascal, que permite programación estructurada: **bucles de iteración** (FOR, WHILE) y **ejecución condicional** (IF-THEN-ELSE, CASE) para control de flujo de programa más complejo
- ✓ Con funciones matemáticas. Muy útil para implementar de forma sencilla desde **cálculos algebraicos** hasta **algoritmos de optimización**
- ✓ **SIEMENS** lo denomina **SCL** (*Structured Control Language*)

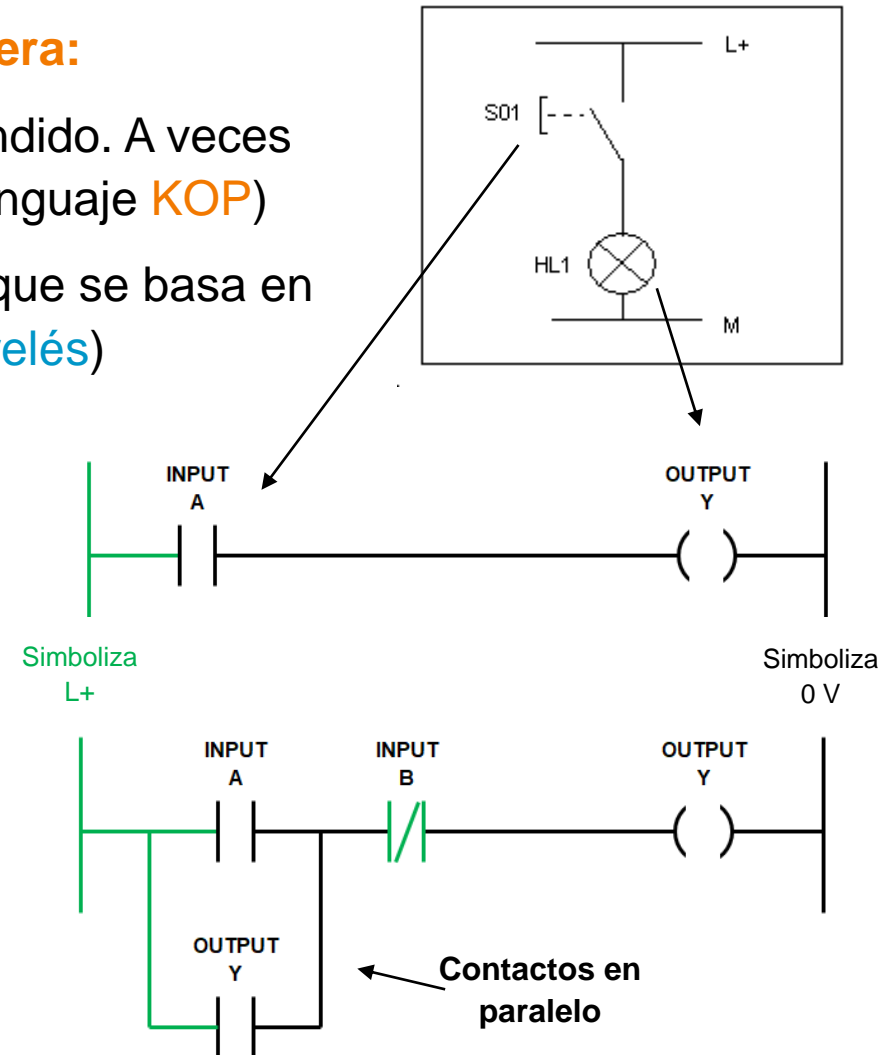
```

1  #Res:= RD_SYS_T(#FechayHora); //llamar a la funcion para leer la fecha y h
2  IF #Res=0 THEN //If Res=0 se ha ejecutado correctamente la lectura
3    #Hora:=#pDatos[3]; //Guardo la Hora extraida de la FechayHora
4    #Minutos:=#pDatos[4]; //Guardo los Minutos extraidos de la FechayHora
5    #Dias:=#pDatos[7]; //En este Byte se guardan los Milisegundo conjuntamente
6    #bit0:=#pbit[0]; //Guardo los 3 bits para saber el dia de la semana
7    #bit1:=#pbit[1];
8    #bit2:=#pbit[2];
9    //Depende del valor de los 3 bits sabré que dia de la semana es, empezand
10 IF #bit0 = true AND #bit1 = false AND #bit2 = false THEN
11   #DiaSemana:=1; //Domingo
12 ELSIF #bit0 = false AND #bit1 = true AND #bit2 = false THEN
13   #DiaSemana:=2; //Lunes
14 ELSIF #bit0 = true AND #bit1 = true AND #bit2 = false THEN
15   #DiaSemana:=3; //Martes
16 ELSIF #bit0 = false AND #bit1 = false AND #bit2 = true THEN
17   #DiaSemana:=4; //Miercoles
  
```

Lenguaje Ladder (LD). Diagramas de escalera. KOP

El lenguaje Ladder o diagrama de escalera:

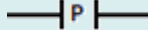
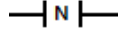
- ✓ Lenguaje visual básico. Es el más extendido. A veces se denomina **esquema de contactos** (lenguaje **KOP**)
- ✓ Es una abstracción de una red eléctrica que se basa en la antigua **lógica cableada** (o **lógica de relés**)
- ✓ Los programas se dividen en **segmentos** (escalones) que conectan dos líneas verticales, que **representan la alimentación eléctrica** (izquierda: **conductor con tensión**, derecha: 0 V)
- ✓ Sus elementos básicos son cables, **contactos** $-| |-$ y **bobinas** $-()-$
- ✓ Los contactos se conectan **en serie** o **en paralelo** en cada segmento



Elementos básicos en diagramas de escalera

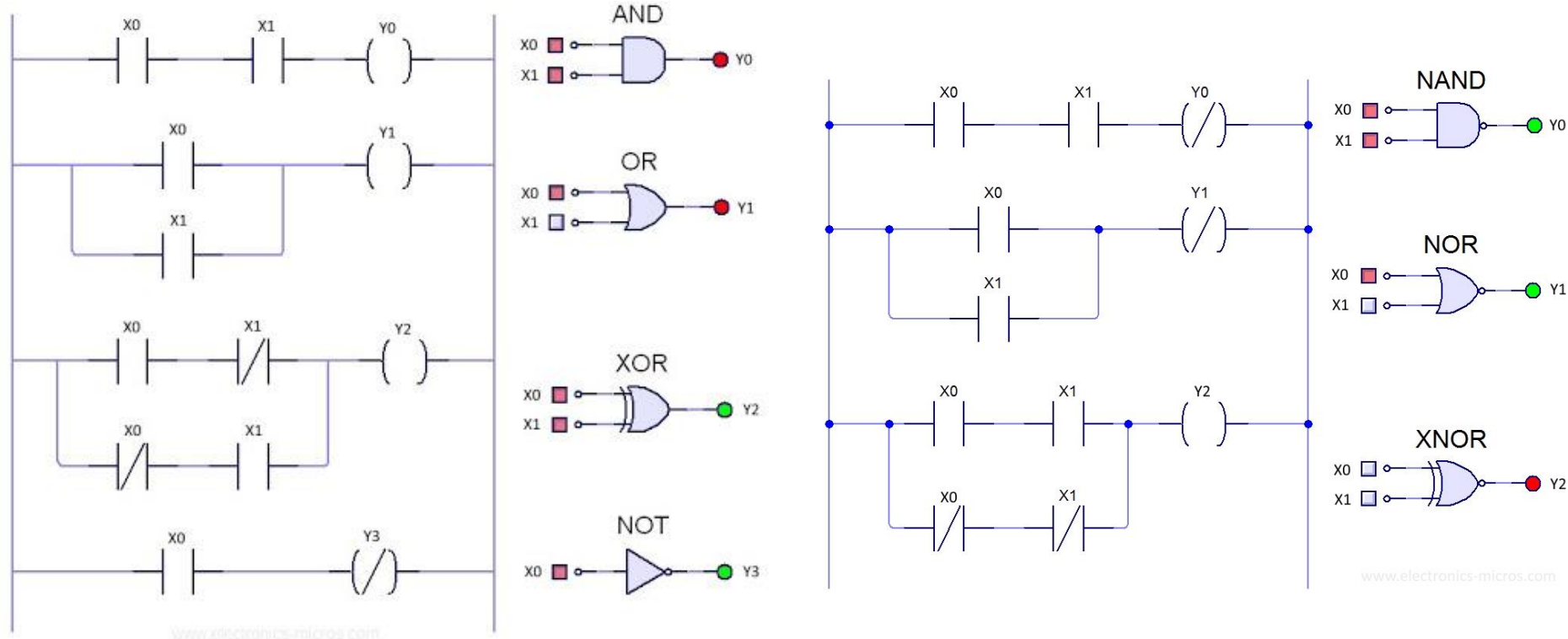
- ✓ Los contactos y las bobinas tienen dos estados (0 ó 1) y pueden ser **normalmente abiertos** (NA) o **normalmente cerrados** (NC)
- ✓ También se utilizan **bobinas S** (SET) y **R** (RESET), equivalentes a biestables SR

Símbolo	Nombre	Descripción
	Contacto NA	Se activa cuando hay un uno lógico en el elemento que representa. Un bit de sistema o variable interna.
	Contacto NC	Su función es similar al contacto NA anterior, pero en este caso se activa cuando hay un cero lógico.
	Bobina NA	Se activa cuando la combinación que hay a su entrada (izquierda) da un uno lógico. Representa elementos de salida, o variables internas.
	Bobina NC	Se activa cuando la combinación que hay a su entrada (izquierda) da un cero lógico.
	Bobina SET	Una vez activa (puesta a 1) no se puede desactivar (puesta a 0) si no es por su correspondiente bobina de RESET.
	Bobina RESET	Permite desactivar una bobina SET previamente activada.

Elemento	Entrada	Salida
	0	0
	1	1
	0	1
	1	0
	1	1
	1	0
 %I0.7 "Tag_2" %M1.0 "Tag_3"	0→1	1
 %I0.7 "Tag_2" %M1.1 "Tag_4"	1→0	0

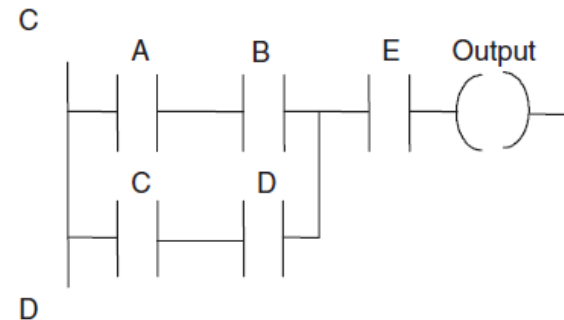
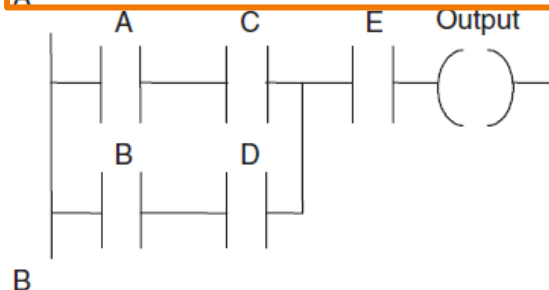
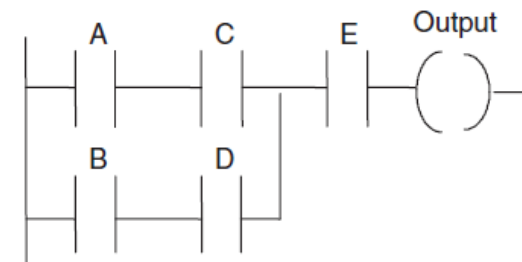
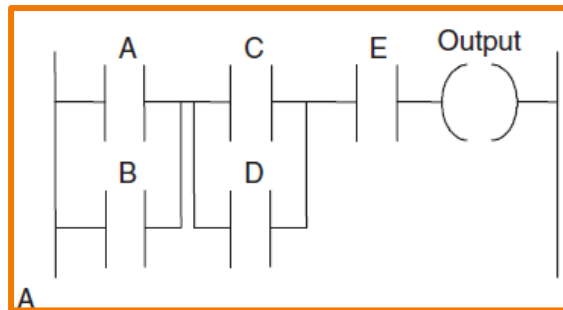
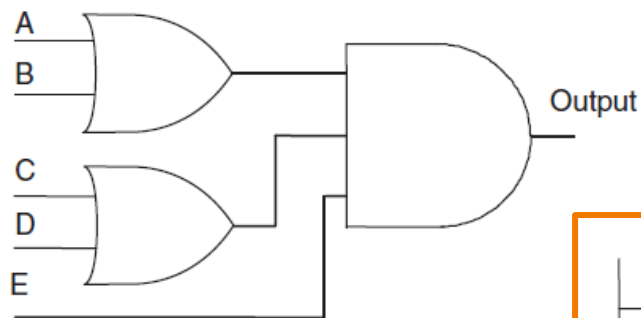
Puertas lógicas y diagramas LD equivalentes

- ✓ Todas las **operaciones lógicas básicas** tienen su equivalencia en un diagrama de contactos:



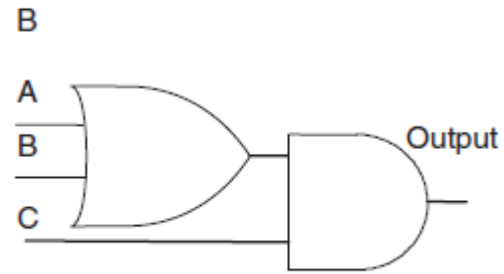
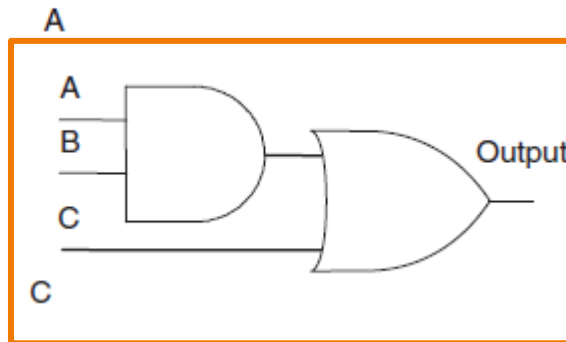
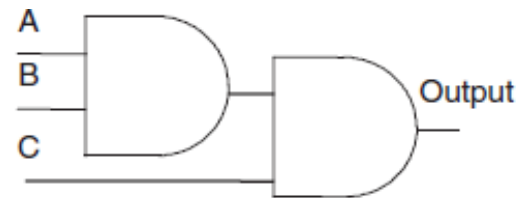
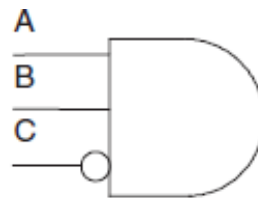
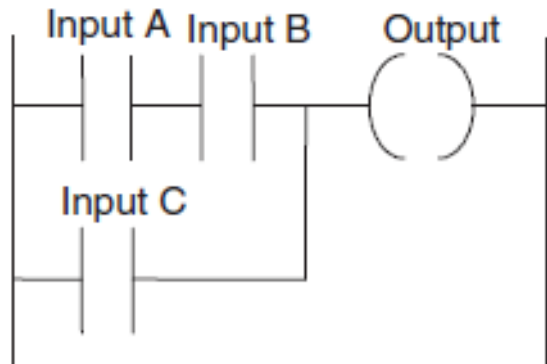
Puertas lógicas y diagramas LD equivalentes

Dado el siguiente circuito lógico compuesto por dos puertas OR y una puerta AND, ¿cuál de las opciones es su equivalente en diagrama de contactos?



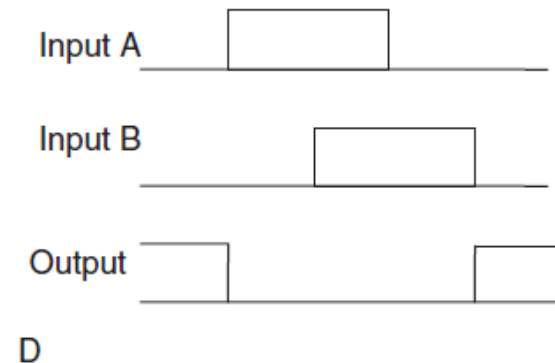
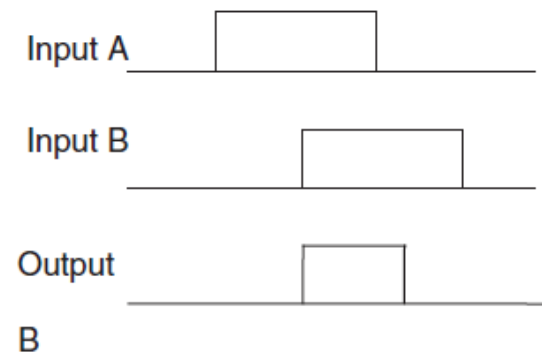
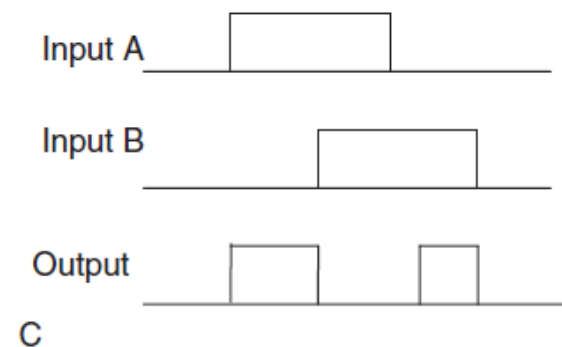
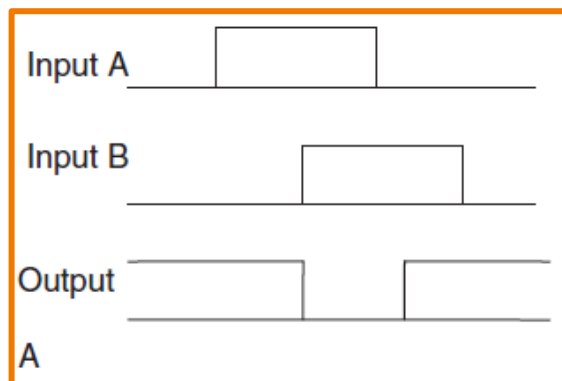
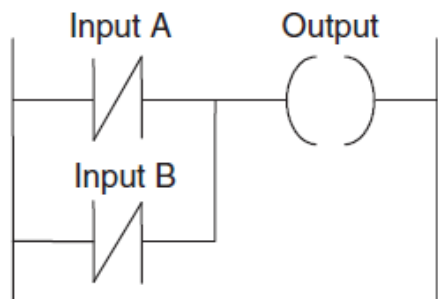
Puertas lógicas y diagramas LD equivalentes

Dado el siguiente diagrama de contactos, ¿cuál de los siguientes es su circuito lógico equivalente?



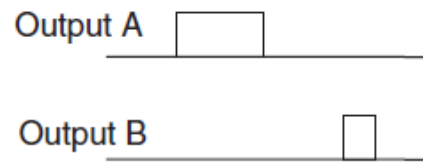
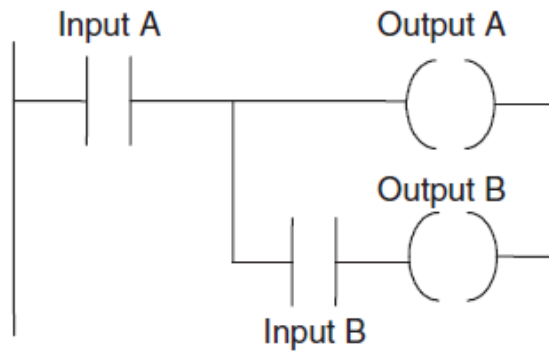
Secuencias temporales de diagramas LD

Dado el siguiente diagrama de contactos, ¿cuál de los siguientes es su diagrama temporal de entrada y salidas?

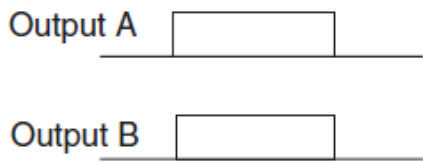
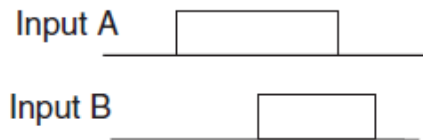


Secuencias temporales de diagramas LD

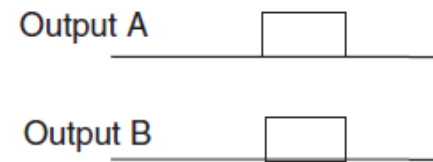
Dado el siguiente diagrama de contactos, ¿cuál de los siguientes es su diagrama temporal de entrada y salidas?



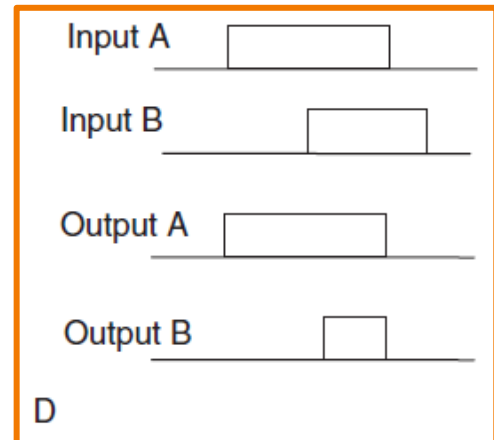
A



C



B



D

Lenguaje Ladder (LD). Entradas y salidas. Ejemplo 1

- ✓ Un programa compuesto de diferentes segmentos los ejecuta de **arriba hacia abajo**. Cada segmento se ejecuta de **izquierda a derecha**.
- ✓ Típicamente, los contactos $-| |-$ están conectados a las **entradas físicas** y las bobinas $-()-$ a las **salidas físicas**, aunque no siempre es así.

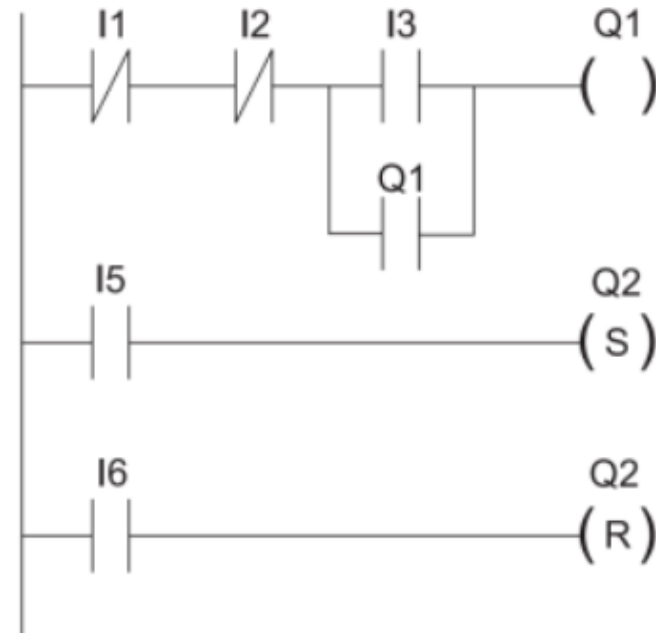
Ejemplo 1. Mando de dos salidas

Salida 1:

- Se activa con la entrada I3
- Mecanismo de autoenclavamiento para mantener la salida activa
- Se desactiva cuando se acciona I1 o I2

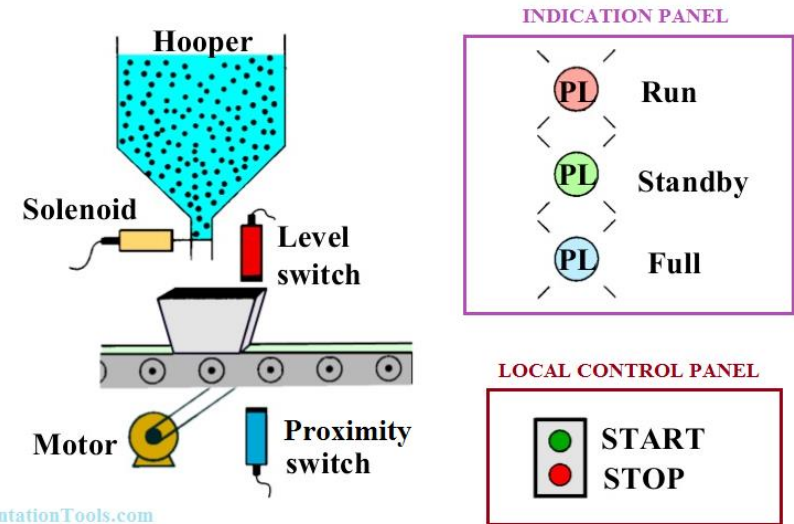
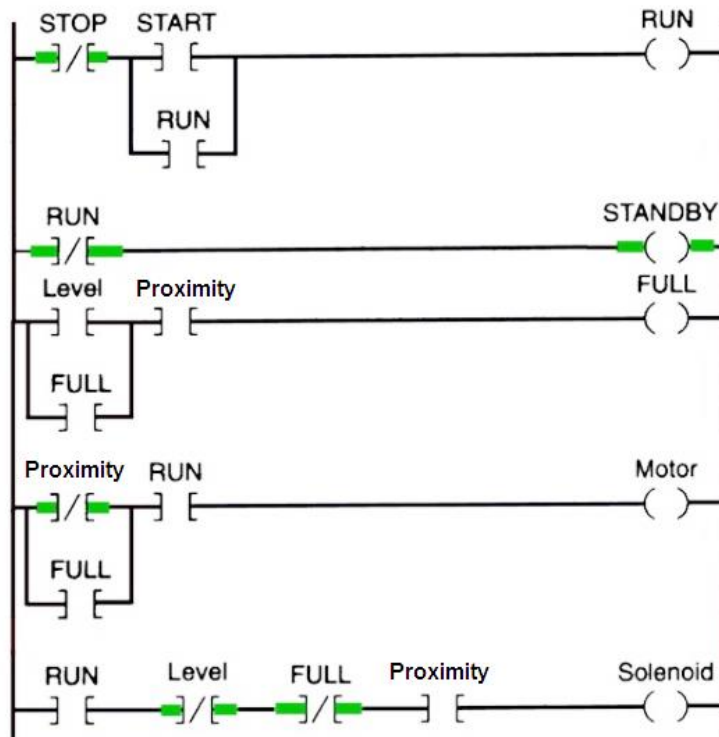
Salida 2:

- Se activa con entrada I5
- Se mantiene activo por el uso de bobinas Set/Reset
- Se desactiva cuando se acciona I6



Lenguaje Ladder (LD). Entradas y salidas. Ejemplo 2

➤ *Ejemplo 2. Estudia y explica el comportamiento del sistema programado mediante el siguiente diagrama de escalera:*



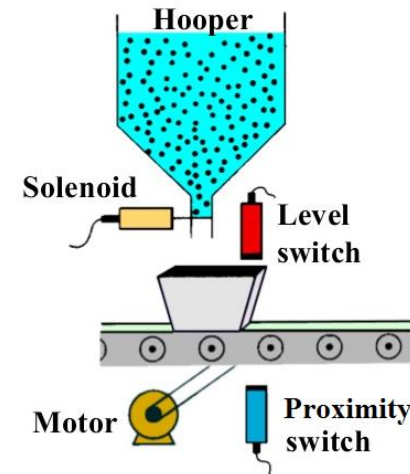
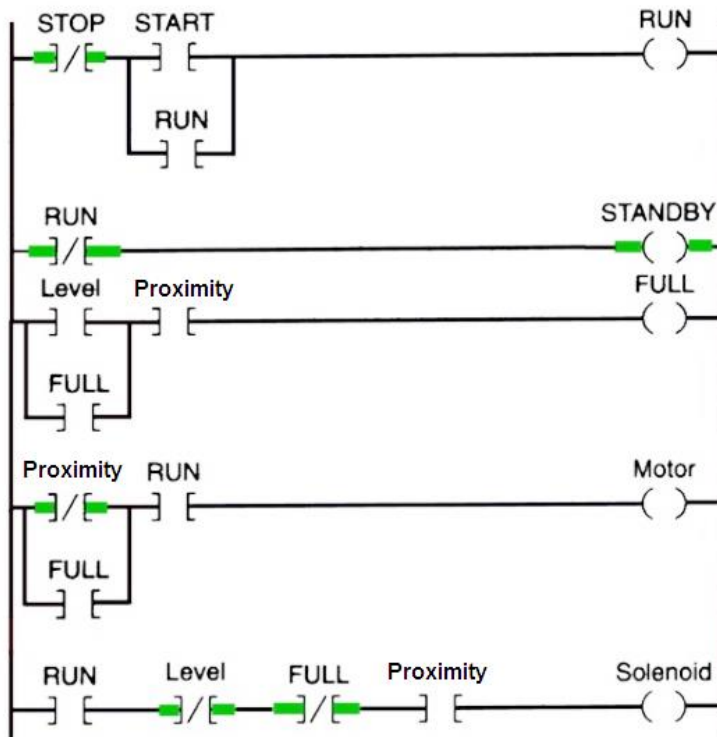
RUN:

- Se activa con la entrada START
- Mantiene activo hasta que se presiona STOP (autoenclavamiento)

STANDBY: se activa cuando no está activa RUN

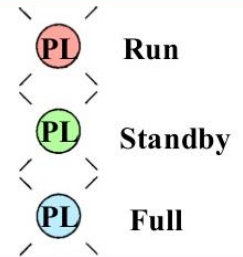
Lenguaje Ladder (LD). Entradas y salidas. Ejemplo 2

Ejemplo 2. Estudia y explica el comportamiento del sistema programado mediante el siguiente diagrama de escalera:



InstrumentationTools.com

INDICATION PANEL



LOCAL CONTROL PANEL

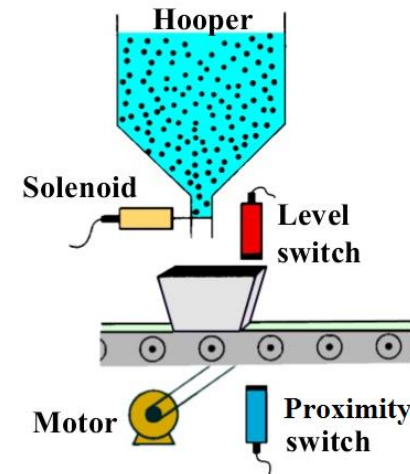
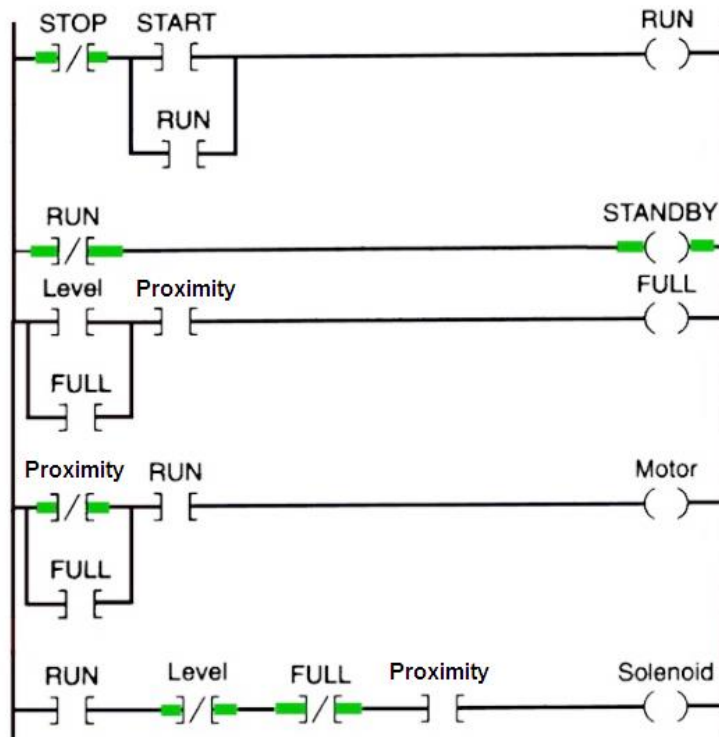


FULL:

- Se activa si el sensor de proximidad detecta recipiente y se alcanza el nivel correcto
- Una vez activo, permanece activo independientemente del nivel
- Se desactiva cuando se deja de detectar el recipiente

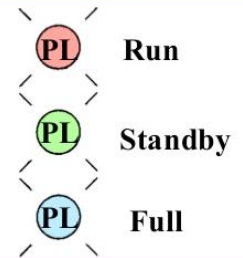
Lenguaje Ladder (LD). Entradas y salidas. Ejemplo 2

Ejemplo 2. Estudia y explica el comportamiento del sistema programado mediante el siguiente diagrama de escalera:



InstrumentationTools.com

INDICATION PANEL



LOCAL CONTROL PANEL



Motor:

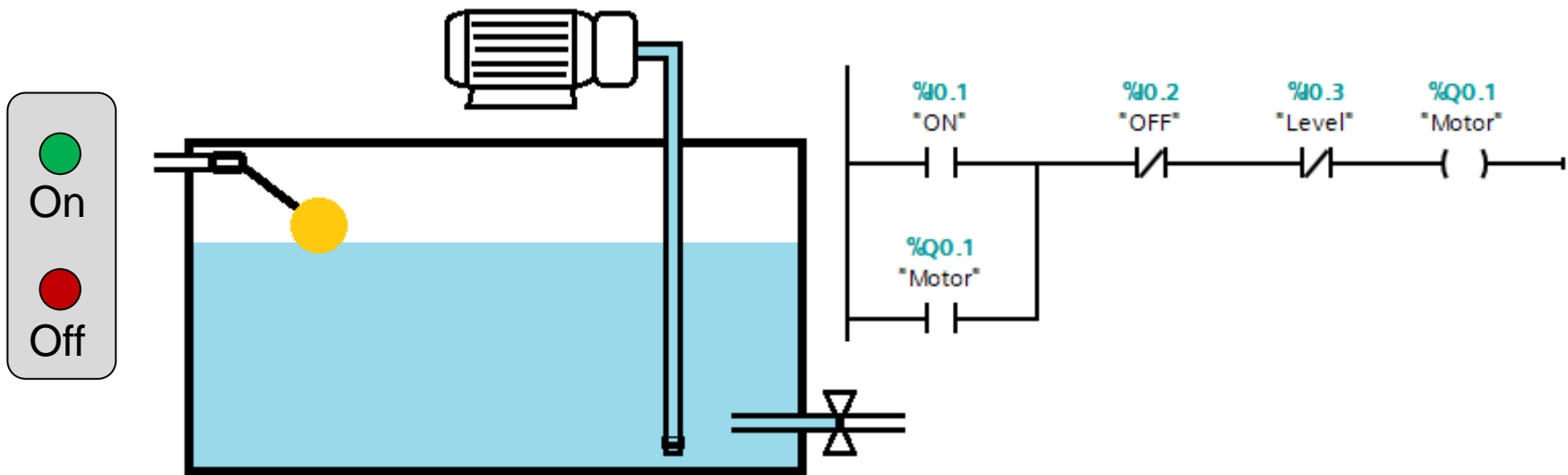
- Se activa cuando el sistema está en RUN y aún no se ha detectado el recipiente o cuando este alcanza el nivel correcto

Solenoid:

- Se activa con el sistema en RUN, cuando se detecta el recipiente y mientras que este no esté lleno

Lenguaje Ladder (LD). Diseño de automatismos 1

Diseño 1. Se pretende controlar el llenado de un depósito. La bomba comenzará a funcionar cuando se accione el Pulsador y se parará cuando se accione el pulsador Off o cuando el sensor de nivel detecte que el líquido ha alcanzado el nivel máximo



Entradas:

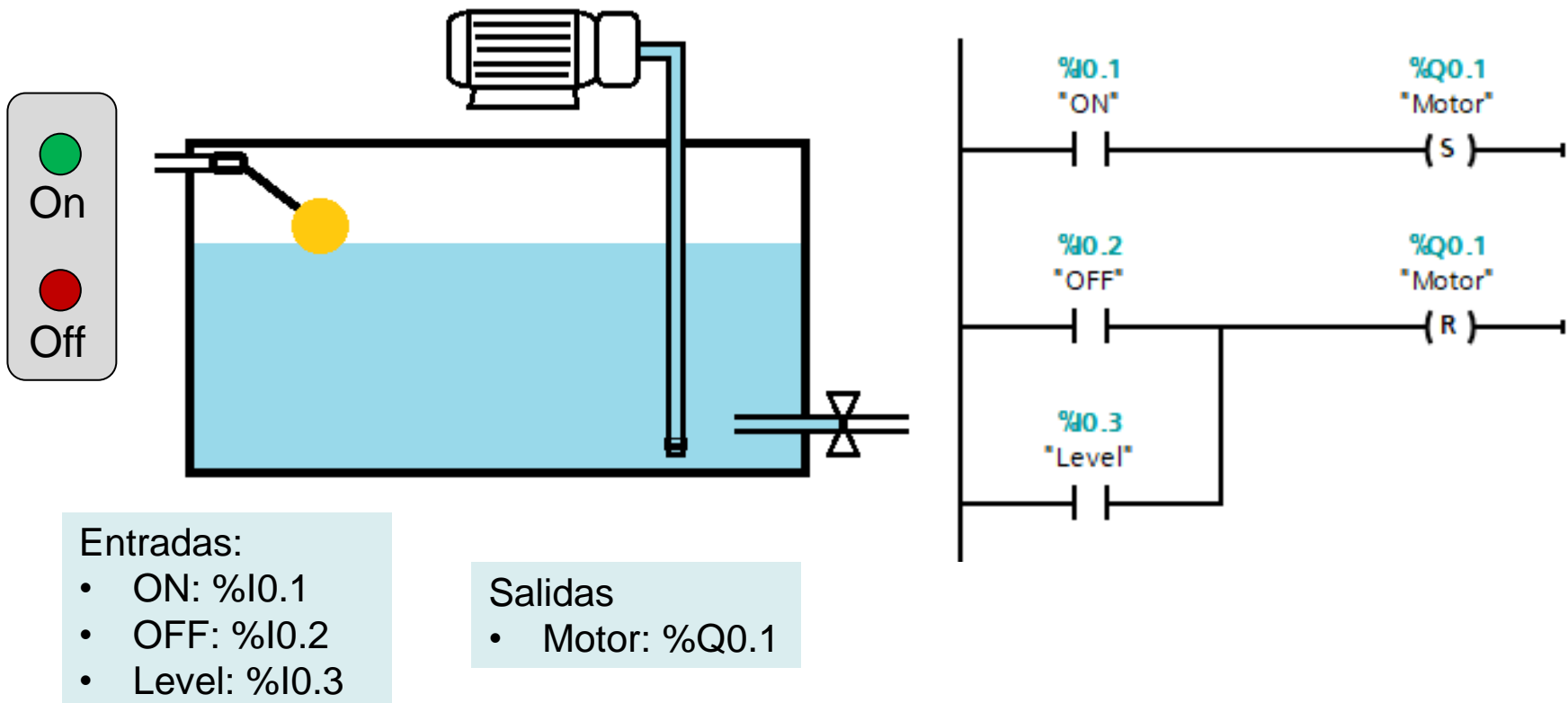
- ON: %I0.1
- OFF: %I0.2
- Level: %I0.3

Salidas

- Motor: %Q0.1

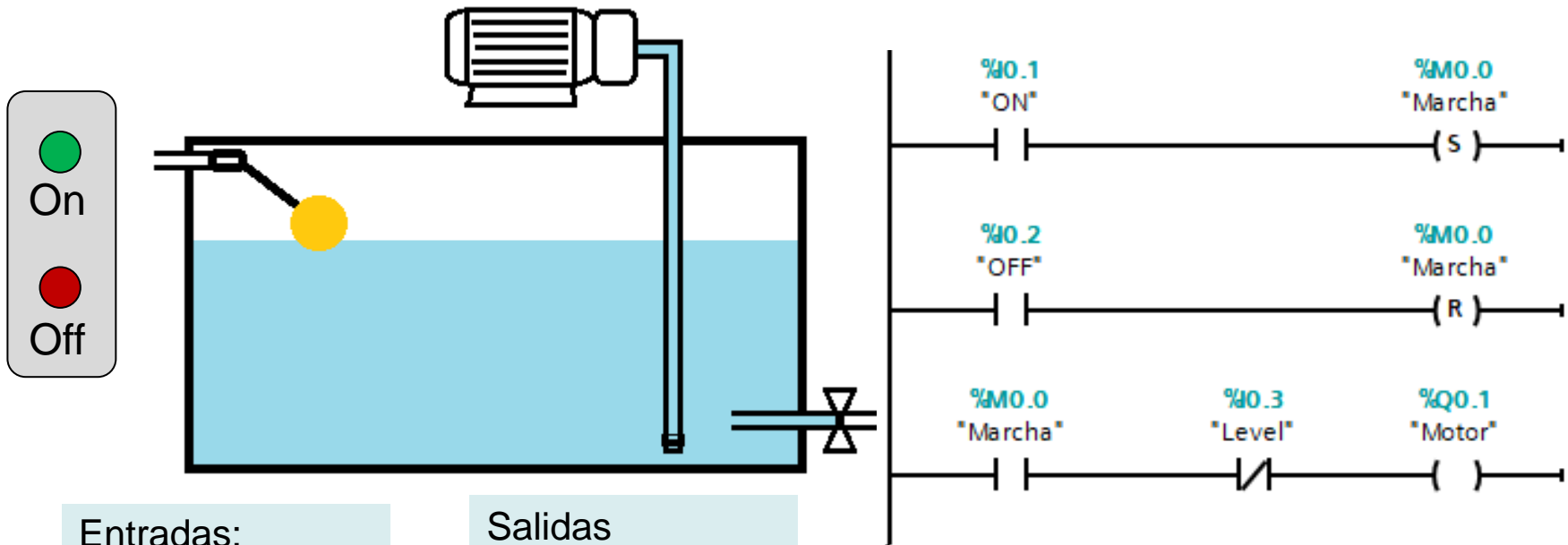
Lenguaje Ladder (LD). Diseño de automatismos 1

Diseño 1b. Se pretende controlar el llenado de un depósito. La bomba comenzará a funcionar cuando se accione el Pulsador y parará cuando se accione el pulsador Off o cuando el sensor de nivel detecte que el líquido ha alcanzado el nivel máximo



Lenguaje Ladder (LD). Diseño de automatismos 1

Diseño 2. Se pretende controlar el llenado de un depósito. La bomba comenzará a funcionar cuando se accione el Pulsador y parará cuando se accione el pulsador Off o cuando el sensor de nivel detecte que el líquido ha alcanzado el nivel máximo. Si se reduce el nivel del depósito y aún no se ha actuado sobre el pulsador Off, la bomba deberá volver a ponerse en marcha.



Entradas:

- ON: %I0.1
- OFF: %I0.2
- Level: %I0.3

Salidas

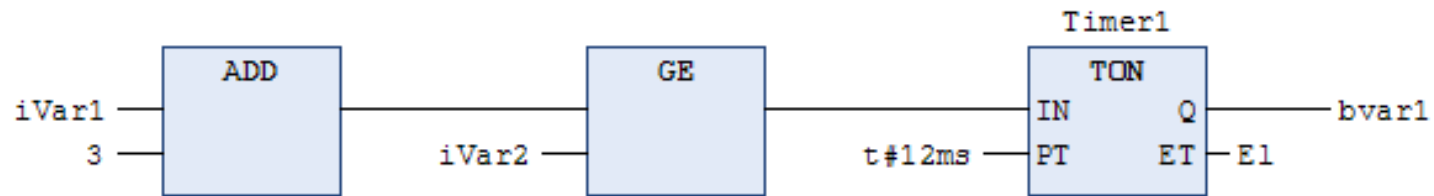
- Motor: %Q0.1

Marca

- Marcha: %M0.0

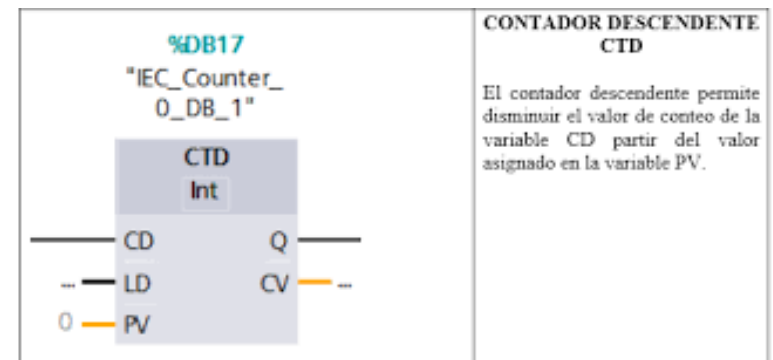
Funcion Block Diagrams (FBD).

Otro de los métodos de programación gráfica para PLC según el IEC 61131-3 son los **Diagramas de Bloques de Función (FBD)**. Se basan en usar bloques con una funcionalidad específica y varias entradas y salidas.



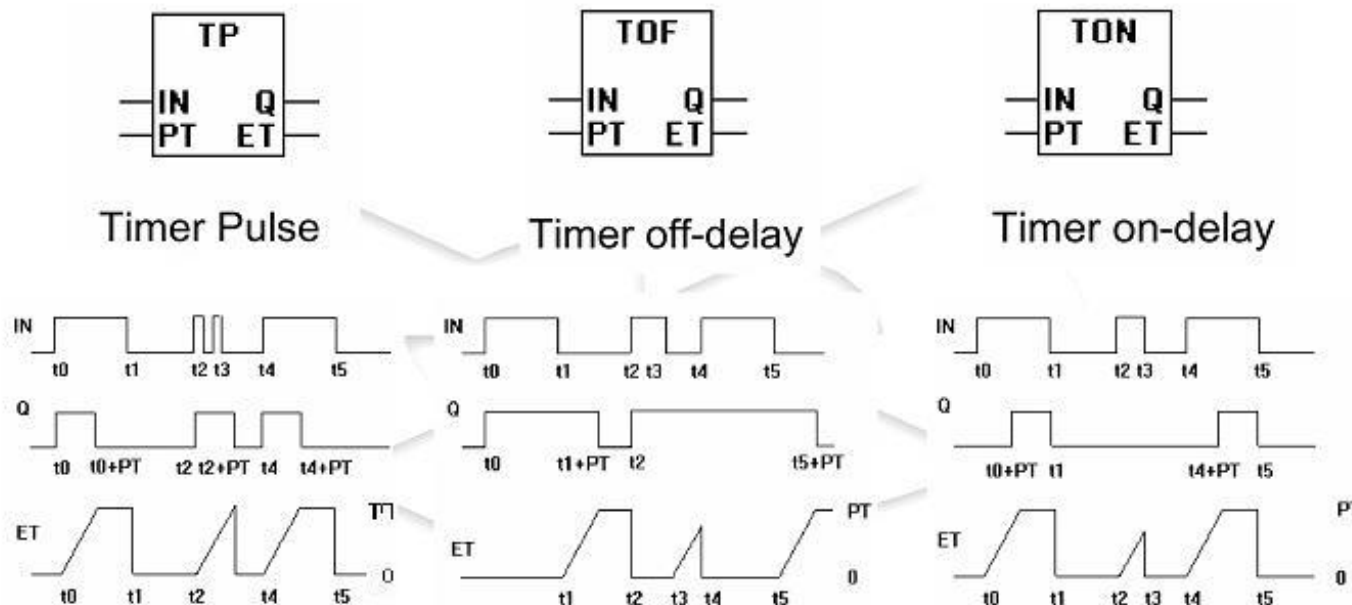
La IEC define varios bloques básicos. El usuario puede crear bloques adicionales con funcionalidades más complejas.

- ✓ Temporizadores (TP, TON, TOF)
- ✓ Contadores (CTU, CTD, CTUD)
- ✓ Biestables (RS, SR)
- ✓ Detección de flancos (F_TRIG, R_TRIG)
- ✓ Bloques de comunicación (IEC 61131-5)



Contabilización de tiempo: Temporizadores

- ✓ **Temporizadores:** bloques software para **activaciones retardadas** de la salida booleana Q. El tiempo de retardo se especifica en **PT (Preset Time)**. **ET (Elapsed Time)** es una salida que da acceso al tiempo transcurrido:
 - **TP:** temporizador de impulso (de duración **PT** desde **flanco subida IN=1**)
 - **TOFF:** temporizador retardo a la desconexión (**Q=0** tras **PT** segundos si **IN=0**)
 - **TON:** temporizador de retardo a la conexión (**Q=1** tras **PT** segundos si **IN=1**)



Curiosidad
 En Siemens TIA Portal el valor máximo de **PT** es de **24 días, 20h y 31m**.
 Además hay un temporizador acumulativo **TONR**.

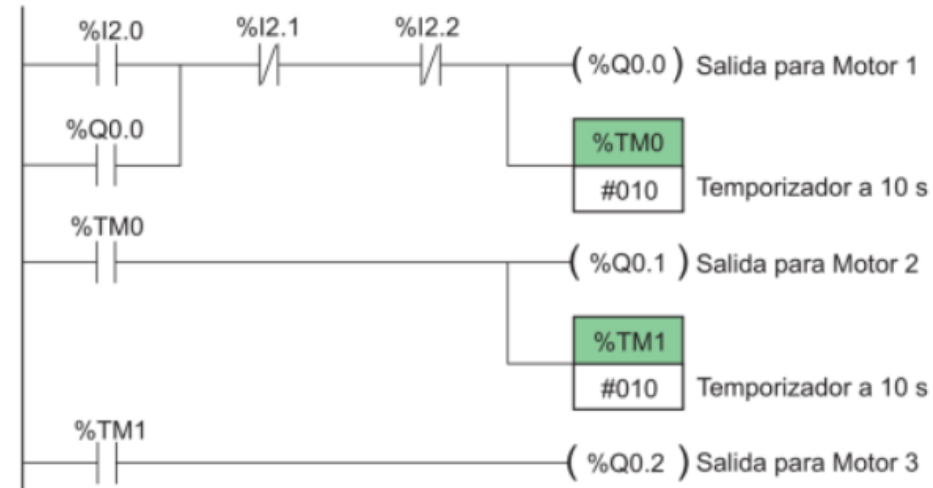
Ejemplo análisis de temporizadores

Ejemplo 1. Arranque secuencial de tres motores eléctricos:

Leyenda

%I2.0: Pulsador Marcha
%I2.1: Pulsador Paro
%I2.2: Relés de protección
%Q0.0: Preactuador QA1
%Q0.1: Preactuador QA2
%Q0.2: Preactuador QA3

Supongamos TM0 y TM1 temporizadores retardo a la conexión (TON)



Motor 1:

- Se activa con la entrada I2.0
- Inicia la cuenta del temporizador TM0
- Mecanismo de autoenclavamiento para mantener la salida activa

Motor 2:

- Se activa cuando el temporizador TM0 cumple su conteo
- Inicia la cuenta del temporizador TM1

Motor 3:

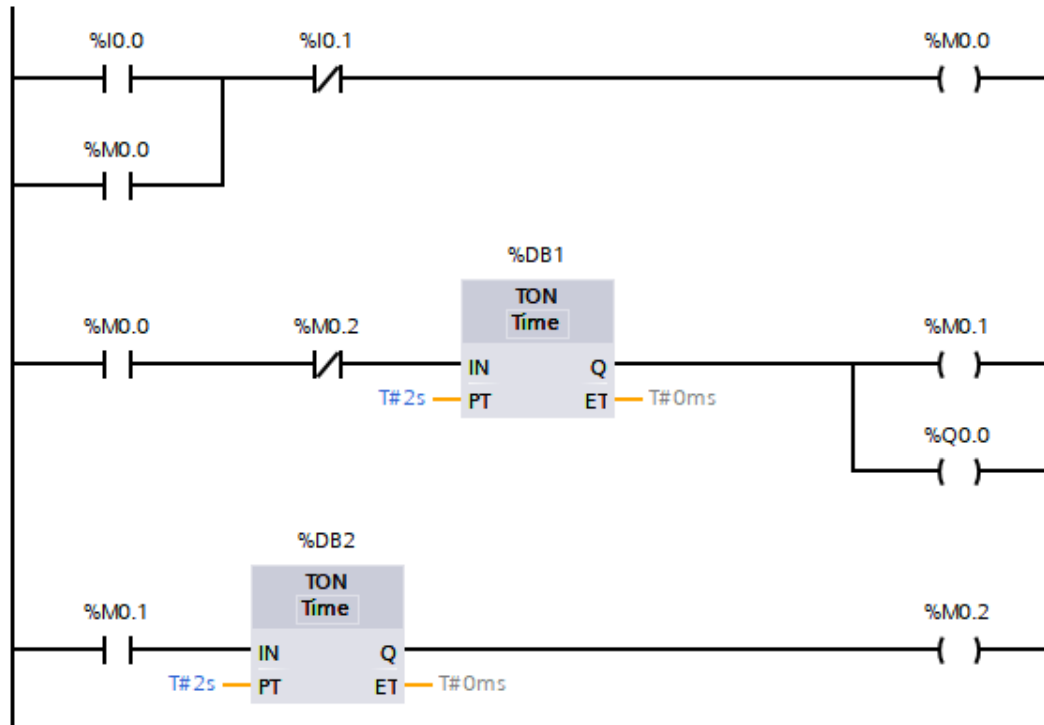
- Se cuando el temporizador TM1 cumple su conteo

Motores 1, 2 y 3:

- Se desactivan cuando se actúa sobre I2.1 o I2.2
- La desactivación de la primera salida se propaga a las demás

Ejemplo análisis de temporizadores

Ejemplo 2. Intermitencia de salida



Marca M0.0:

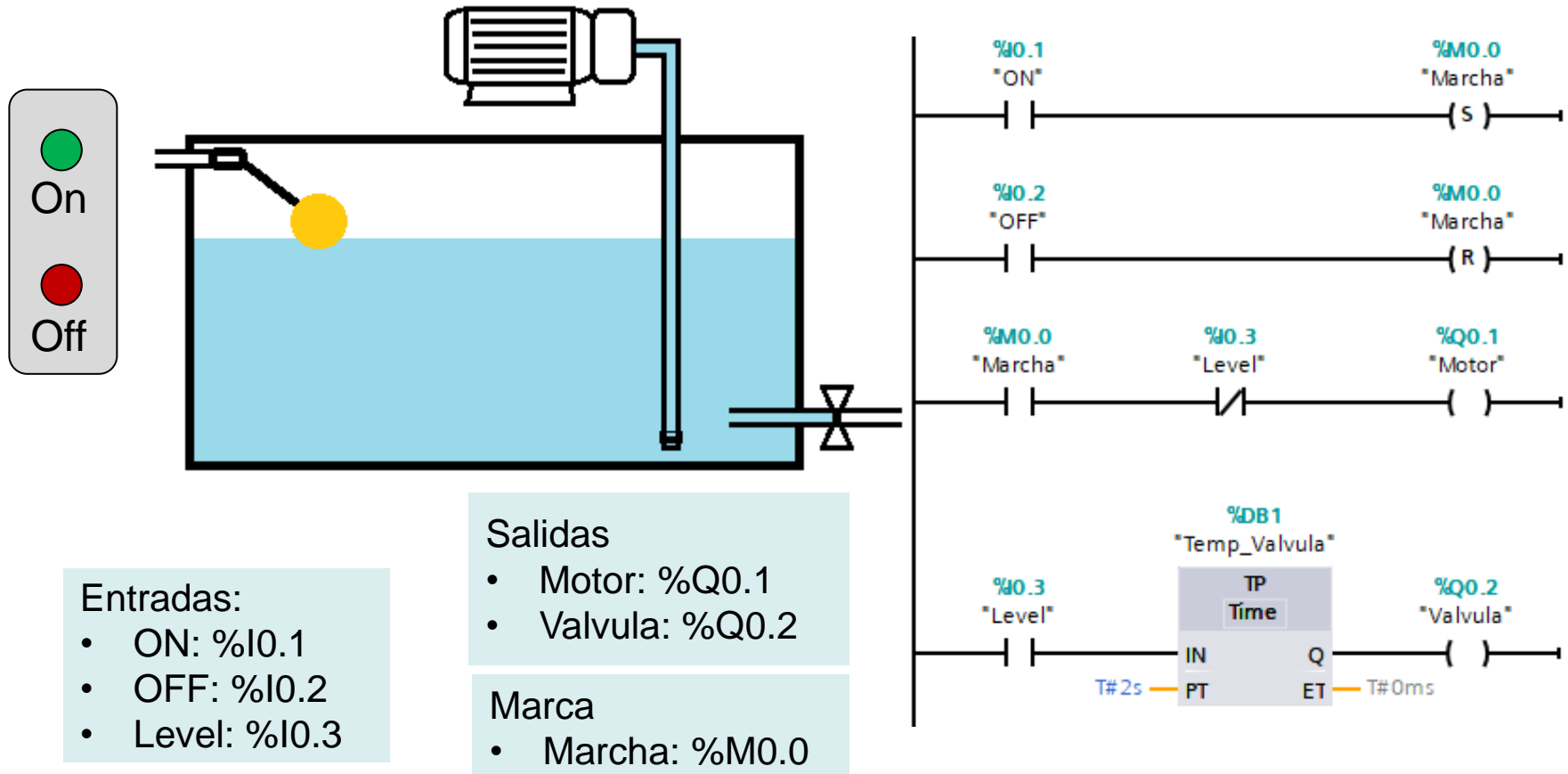
- Se activa con la entrada I0.0 y se autoenclava.
- Permanece activa hasta que se actúa I0.0

- Al activarse M0.0, con M0.2 inactiva, se inicia el temporizador DB1 (TON).
- Tras 2s, se activan tanto la marca M0.1 como la salida Q0.0
- Al activarse M0.1, se arranca el temporizador DB2 (TON). Tras 2 segundos se activa M0.2
- Al activarse M0.2, se desactiva la entrada a DB1, desactivándose M0.1, Q0.0, DB2 y la propia M0.2
- Al desactivarse M0.2, vuelve a activarse DB1, reiniciando el ciclo
- El ciclo continua hasta que se desactiva M0.0 (con la activación de I0.1)

Resultado: Parpadeo de Q0.0 (2s Off, 2s On) desde que se activa I0.0 hasta que se activa I0.1

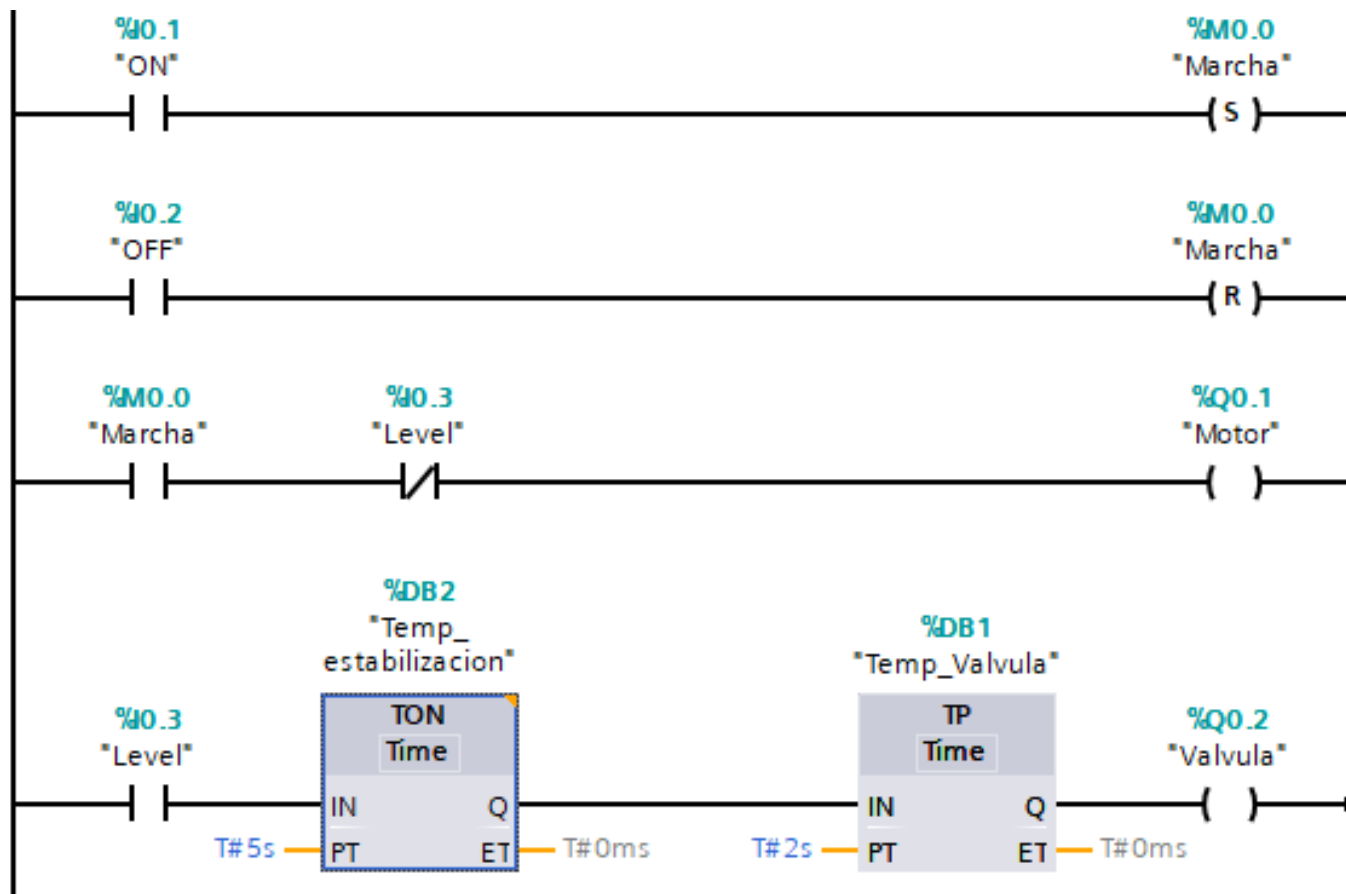
Diseño de automatismos 2, uso de temporizadores

Diseño 3. Se pretende controlar el llenado de un depósito. Manteniendo la funcionalidad del diseño 2, cada vez que el líquido active el sensor de nivel, la válvula de salida deberá abrirse durante un tiempo igual a 2 segundos



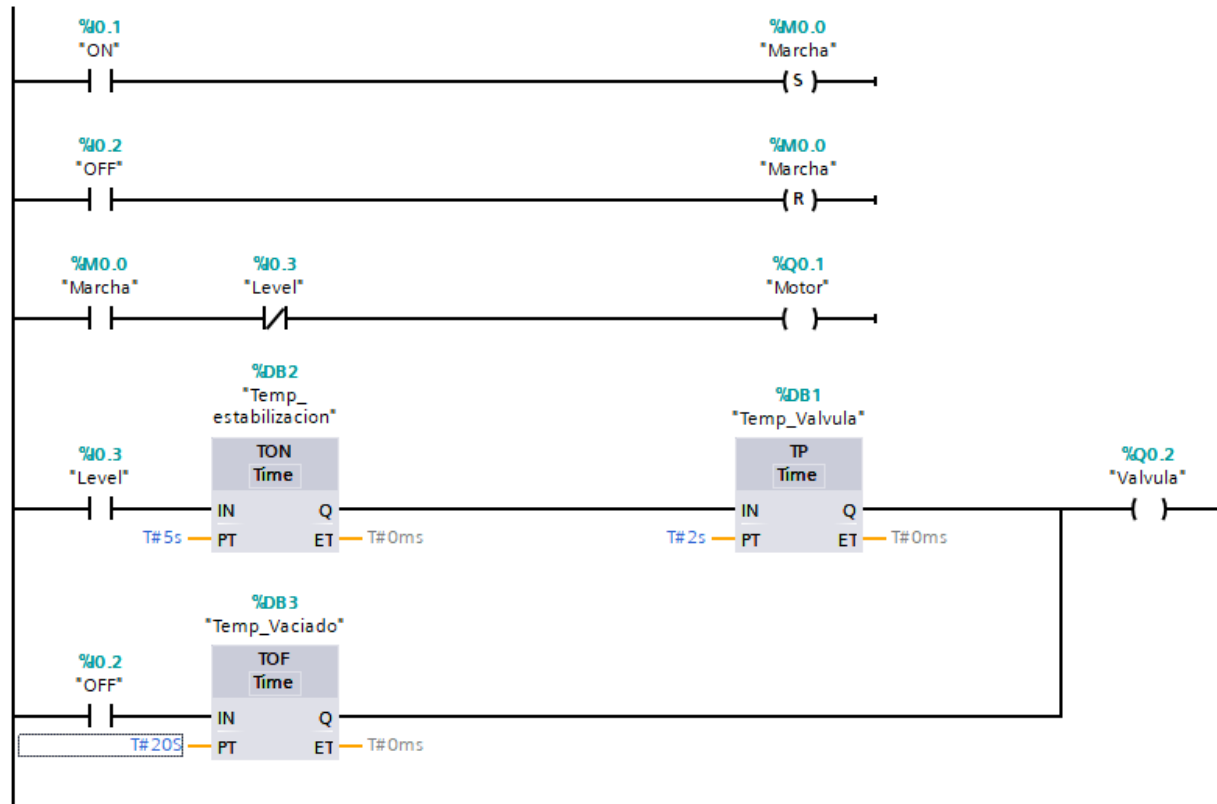
Diseño de automatismos 2, uso de temporizadores

Diseño 4. Se pretende controlar el llenado de un depósito. Manteniendo la funcionalidad del diseño 3. Para asegurar que el contenido del tanque se ha estabilizado, la válvula sólo se abrirá una vez transcurridos 5s tras alcanzar el nivel máximo



Diseño de automatismos 2, uso de temporizadores

Diseño 5. Se pretende controlar el llenado de un depósito, manteniendo la funcionalidad del diseño 4. Una vez apagado el sistema (mediante el pulsador OFF) hay que disminuir el nivel del depósito por motivos de seguridad, para ello, la válvula de salida deberá permanecer abierta durante 20 segundos.



Contabilización de tiempo mediante marcas de ciclo

- ✓ **Marcas de ciclo:** son posiciones de memoria que cambian periódicamente de estado con una relación 1:1 entre estado '1' y '0'.

Bits de marcas de ciclo

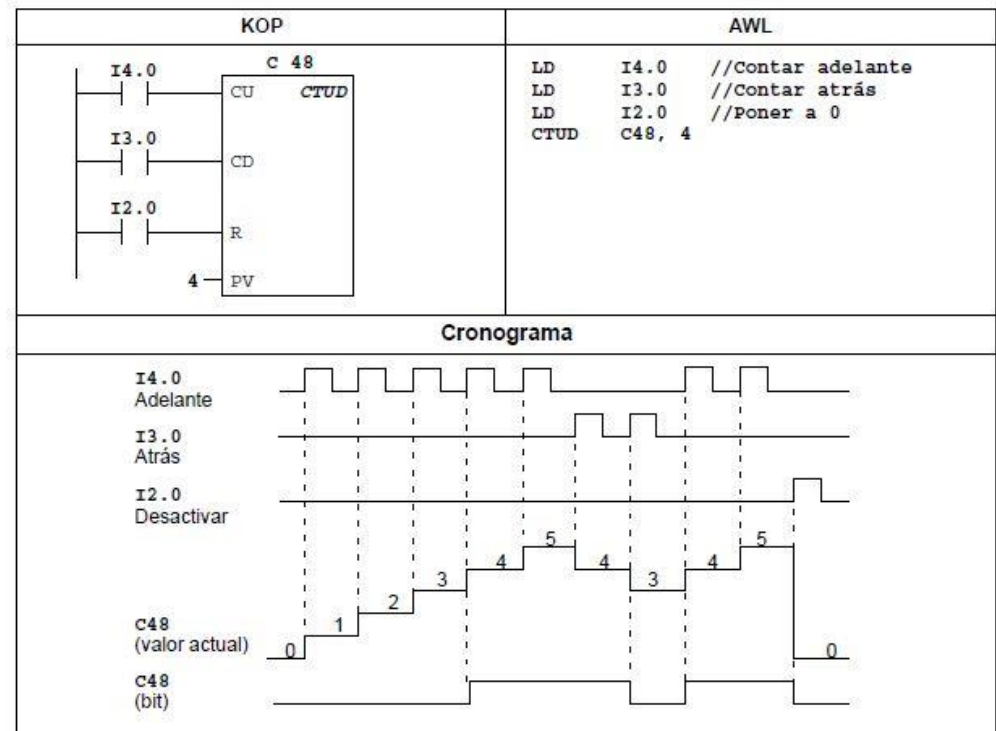
☒ Activar la utilización del byte de marcas de ciclo

Dirección del byte de marcas de ciclo (MBx):

Reloj 10 Hz:	%M1.0 (Clock_10Hz)
Reloj 5 Hz:	%M1.1 (Clock_5Hz)
Reloj 2.5 Hz:	%M1.2 (Clock_2.5Hz)
Reloj 2 Hz:	%M1.3 (Clock_2Hz)
Reloj 1.25 Hz:	%M1.4 (Clock_1.25Hz)
Reloj 1 Hz:	%M1.5 (Clock_1Hz)
Reloj 0.625 Hz:	%M1.6 (Clock_0.625Hz)
Reloj 0.5 Hz:	%M1.7 (Clock_0.5Hz)

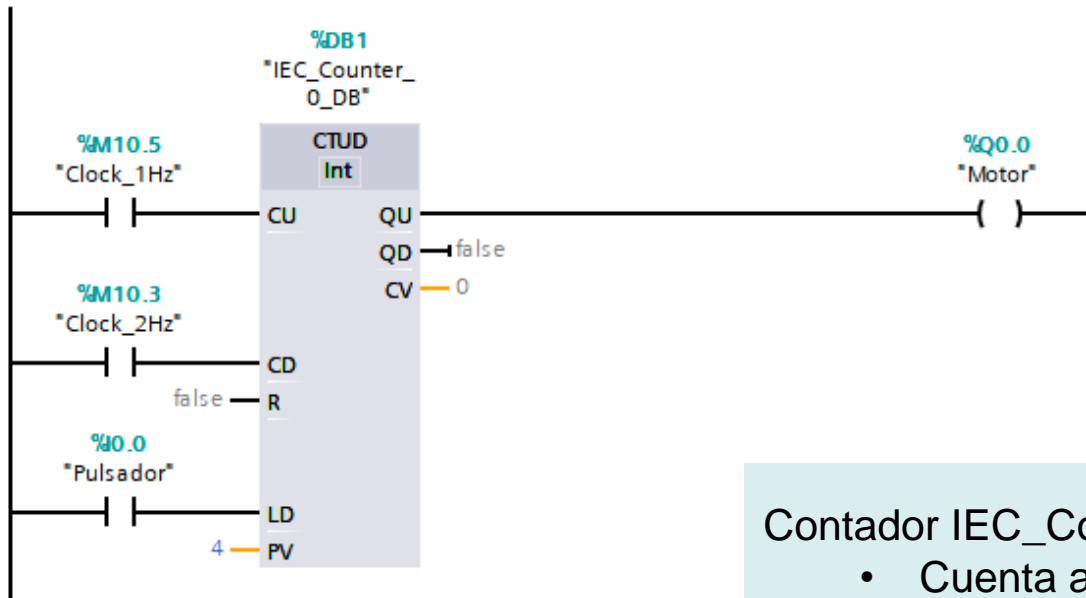
Contabilización de eventos: Contadores

- ✓ **Contadores:** permiten contabilizar el número de activaciones (flancos de subida, típicamente) de sus entradas.
- **Entradas típicas:** **CU** (Counter Up), **CD** (Counter Down), **LD** (Load: carga del valor dado en PV, **PV** (Valor Preajustado) y **R** (Reset)
- **Salidas:** **Q** (booleana, = 1 cuando se alcanza **PV**). Y **CV** (valor actual de conteo)
- **Tipos:**
 - **CTU:** contador ascendente
 - **CTD:** contador descendente
 - **CTUD:** contador ascendente y descendente



Ejemplo análisis de contadores

Ejemplo 1. Contadores y marcas de ciclo



Resultado:

La salida conectada al motor se activa con el pulsador, pero se desactiva con el primer flanco ascendente en la marca de ciclo a 2 Hz.

Marcas de ciclo M10.x:

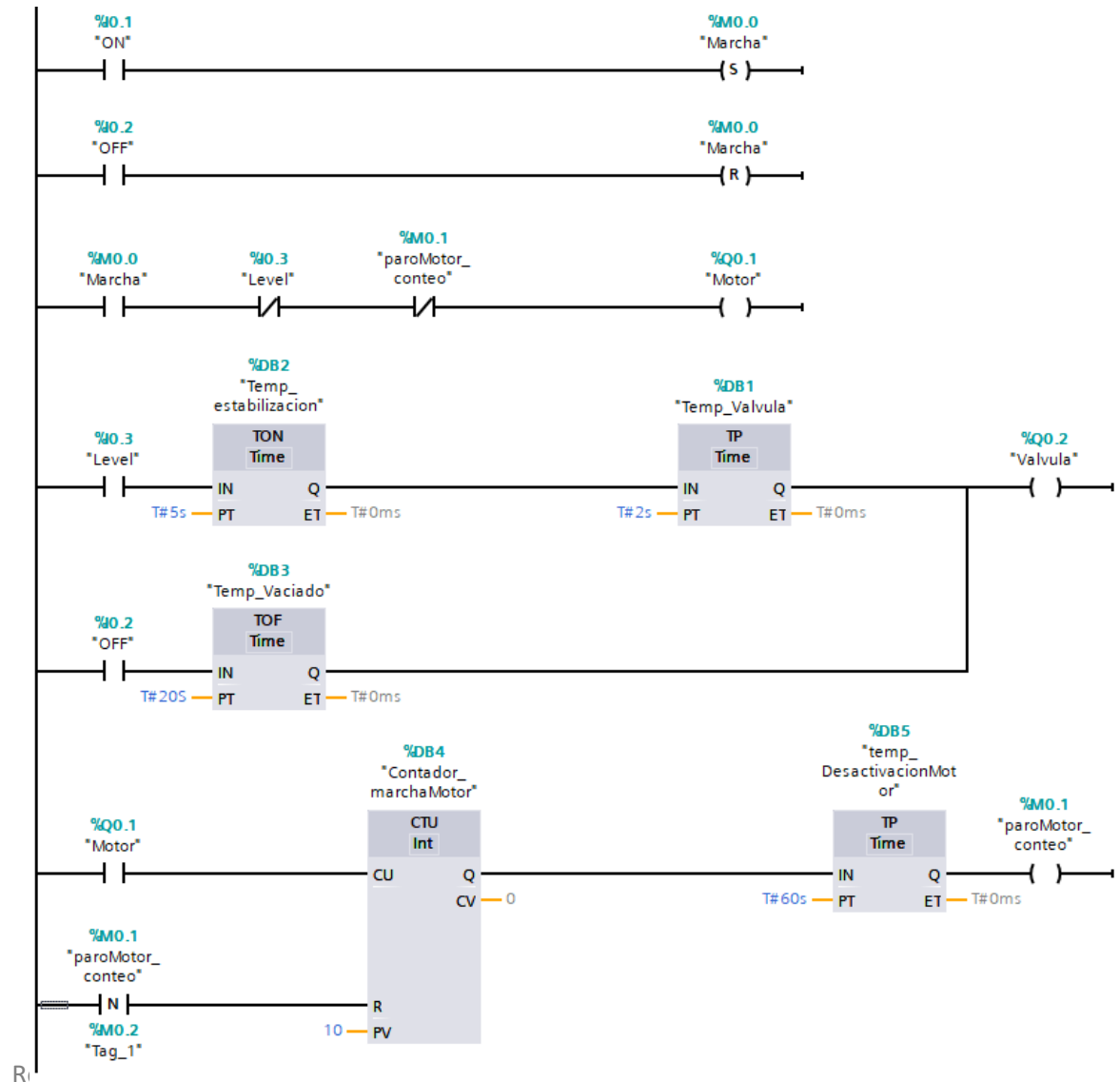
- M10.5 genera una señal cuadrada de 1Hz (1 pulso por segundo)
- M10.3 genera una señal cuadrada de 2 Hz (2 pulsos por segundo)

Contador IEC_Counter_0_DB:

- Cuenta ascendente con la marca a 1Hz
- Cuenta descendente con la marca a 2Hz
 - Por cada cuenta positiva se obtienen dos negativas
- Salida a 1 cuando el conteo llegue a 4
- El contador se carga con el PV al accionar el pulsador (entrada LD del contador)

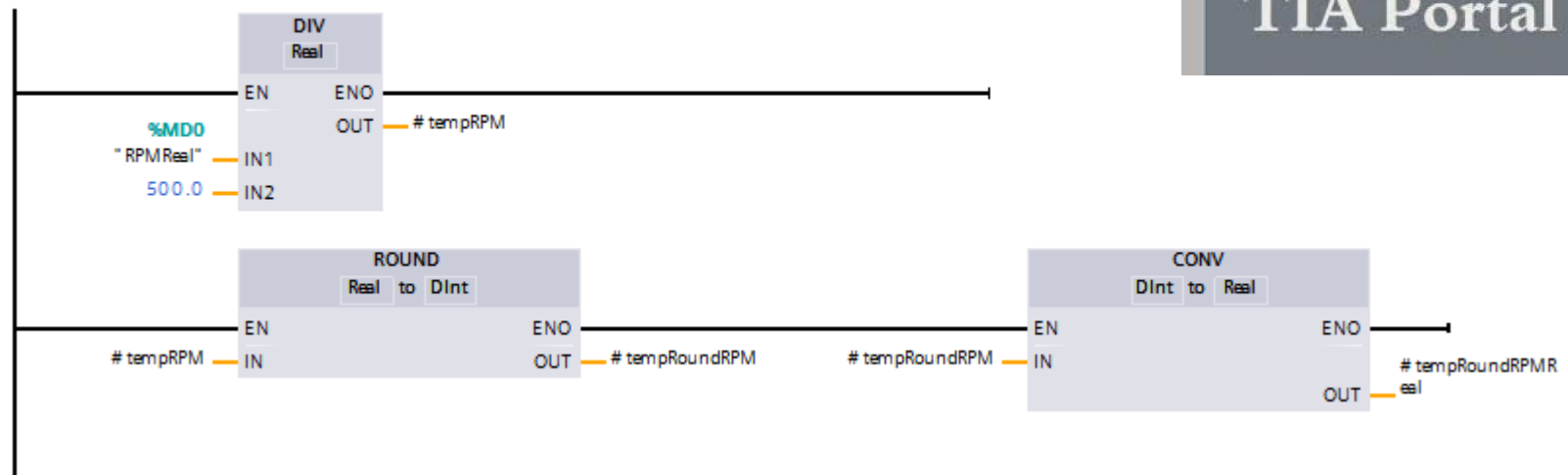
Diseño de automatismos 3, uso de contadores

Diseño 6. Se pretende controlar el llenado de un depósito, manteniendo la funcionalidad del diseño 5. Para evitar dañar la bomba, se monitorizará el número de veces que el motor se enciende, de manera que tras haberse activado 10 veces, este deberá permanecer desactivado durante un minuto.



Bloques en entorno SIEMENS TIA Portal

El entorno de programación de **STEP 7 de SIEMENS** (denominado **TIA Portal**) contiene muchos más bloques predefinidos. Alguno se verán en las prácticas:



Network 2:

calculate how close TemproundRMReal=RPM/500 is to a whole number and save in #tempdiff

