

PROGRAMACIÓN INDUSTRIAL



AUTOMATIZACIÓN

OBJETIVOS PRINCIPALES DE LA AUTOMATIZACIÓN

- La automatización de los procesos de producción persigue los objetivos:
 - Mejorar la calidad y mantener un nivel de calidad uniforme.
 - Producir las cantidades necesarias en el momento preciso.
 - Mejorar la productividad y reducir costes.
 - Hacer mas flexible el sistema productivo (facilitar los cambios en la producción).

DEFINICIÓN DE AUTOMATISMO

- Se define un sistema (maquina o proceso) automatizado como aquel capaz de reaccionar de forma automática (sin la intervención del operario) ante los cambios que se producen en el mismo, realizando las acciones adecuadas para cumplir la función para la que ha sido diseñado

OBJETIVO DE LOS AUTOMATISMOS

- El objetivo de un automatismo es controlar una planta o sistema sin necesidad que un operario intervenga directamente sobre los elementos de salida. El operario solo debe intervenir sobre las variables de control y el automatismo es el encargado de actuar sobre las salidas mediante los accionamientos con el fin de poder llevar a efecto el control de la planta.

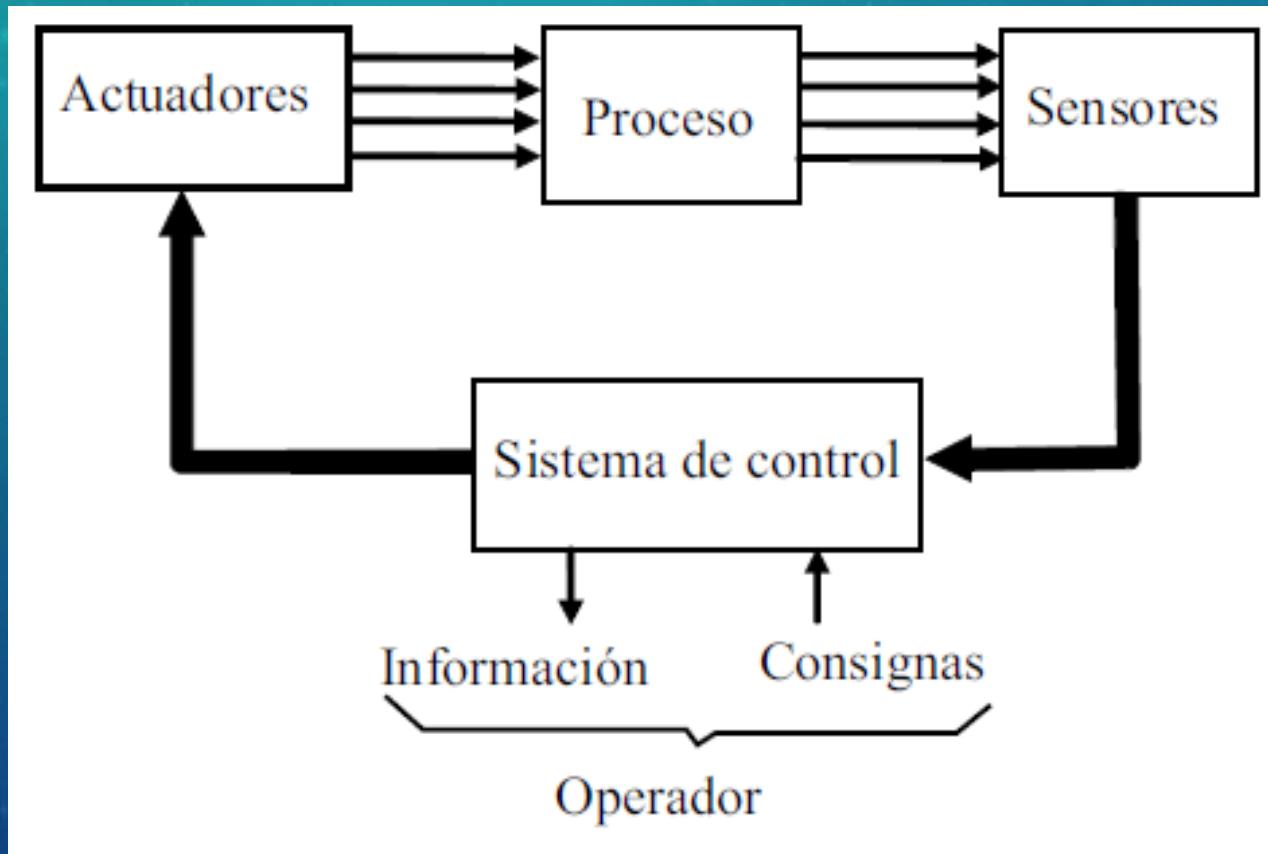
NIVELES EN LA AUTOMATIZACIÓN

- Se pueden distinguir varios niveles en la automatización de un proceso productivo:
- 1. Nivel de maquina. En este nivel se considera la automatización de una maquina que realiza una tarea productiva simple determinada.
- 2. Nivel de célula (de grupo). En este nivel se considera el control automatizado de un conjunto de maquinas que trabajan conjunta y coordinadamente para realizar un proceso de producción más complejo.

NIVELES EN LA AUTOMATIZACIÓN

- 3. Nivel de planta. En este nivel se considera el control automatizado de toda la planta de producción que trabaja de forma coordinada para cumplir unos objetivos de producción global de la fabrica.
- 4. Nivel de empresa. En este nivel se considera el conjunto de la empresa (gestión, ventas, producción).

- En la siguiente figura se muestra la estructura típica de un sistema automatizado.



- En el nivel 2 puede haber una red local de comunicación entre los distintos elementos de una célula (si las maquinas están muy separadas). La implementación de los niveles 1 y 2 se realiza mediante sensores, actuadores y equipos de control.
- Un automatismo es en esencia una maquina o un proceso automatizado.
- Los automatismos definen, por tanto, los niveles 1 y 2.

- Las señales de entrada y de salida pueden ser de cualquier tipo, sin embargo el concepto tradicional de automatismo se utiliza para sistemas de eventos discretos (también llamados sistemas secuenciales) en los que esas señales son binarias, es decir, solo pueden tomar 2 valores, activa o inactiva (estos valores suelen representarse como un 1 ó un 0).

- En ese caso el sistema de control implementa el algoritmo de lógica binaria que relaciona los valores que van tomando en cada instante las entradas (1 ó 0) con los valores que deben ir tomando en cada instante las salidas (también 1 ó 0) para que el sistema funcione adecuadamente.

VENTAJAS DE AUTOMATIZAR PROCESOS

- Una de las razones principales para el uso de sistemas automatizados fue y sigue siendo la necesidad de producir a costos cada vez menores para ser competitivos. La técnica de la automatización contribuye a ese fin de varias maneras:
 - En las secciones de fabricación automatizada se necesitan menos operarios.
 - Se puede fabricar las 24 horas del día, interrumpiendo los procesos únicamente para realizar trabajos de mantenimiento.

VENTAJAS DE AUTOMATIZAR PROCESOS

- En términos generales, las máquinas cometen menos errores que los humanos, por lo que los productos tienen un alto y constante nivel de calidad.
- Los tiempos de los procesos son menores. Es posible entregar a los clientes más productos en menos tiempo.
- Gracias a la automatización, los operarios no tiene que hacer trabajos monótonos, pesados y peligrosos o nocivos para la salud.

DESVENTAJAS

- Eliminación de puestos de trabajo, especialmente aquellos que pueden ocupar trabajadores de bajo nivel de cualificación (en vez de 10 trabajadores no cualificados, se necesita un solo operario cualificado, encargado del servicio técnico).
- La automatización de los procesos de fabricación implica que los operarios tomen decisiones específicas, cuyas consecuencias no puede apreciar en su totalidad debido al carácter complejo de las instalaciones.
- Los costos originados por un sistema automático tienen como consecuencia que los individuos asumen una mayor responsabilidad en relación con el éxito de la empresa.

LA SEGURIDAD EN LOS AUTOMATISMOS

- Un aspecto que tiene especial importancia en la implementación de automatismos es la seguridad.
- Ésta se debe tener en cuenta de dos formas. Por una parte se debe definir la secuencia de operaciones del proceso de forma que se garantice en todo momento la seguridad de los operarios.

LA SEGURIDAD EN LOS AUTOMATISMOS

- Por ejemplo, una prensa en la que el operario introduce una chapa para después darle al pulsador de marcha. La secuencia del automatismo debería permitir la puesta en marcha de la prensa solo cuando el operario pulsa de forma simultanea dos pulsadores separados.

LA SEGURIDAD EN LOS AUTOMATISMOS

- En función del nivel de riesgo puede ser necesario utilizar en la implementación una tecnología adecuada que garantice la seguridad.

LA SEGURIDAD EN LOS AUTOMATISMOS

- Por ejemplo, si el automatismo se implementa mediante un autómata programable y se quiere garantizar que la apertura de una puerta produzca la parada instantánea de la maquina, no basta con definir la secuencia para que así sea, sino que hay que utilizar un interruptor y un relé de seguridad que corte la alimentación de la maquina independientemente del autómata programable que la controla.

LA TÉCNICA DE LA AUTOMATIZACIÓN COMO PARTE DE LAS CIENCIAS DE INGENIERÍA

- Para configurar procesos automáticos modernos se necesitan tres componentes principales:
 - Sensores para captar los estados del sistema
 - Actuadores para emitir los comandos de control
 - Unidades de control para la ejecución del programa y para tomar decisiones

SENORES Y ACTUADORES INDUSTRIALES.

Sensores Industriales

- Finales de carrera
- Detectores inductivos
- Detectores capacitivos
- Detectores ultrasónicos
- Detectores fotoeléctricos
- Detectores de presión

SENsoRES Y ACTUADoRES INDUSTRIALES.

Actuadores Industriales

- Accionamientos eléctricos (Motores DC, AC, Servomotores, etc)
- Accionamientos neumáticos (Cilindros, motores)
- Accionamientos hidráulicos (Cilindros, Motores)

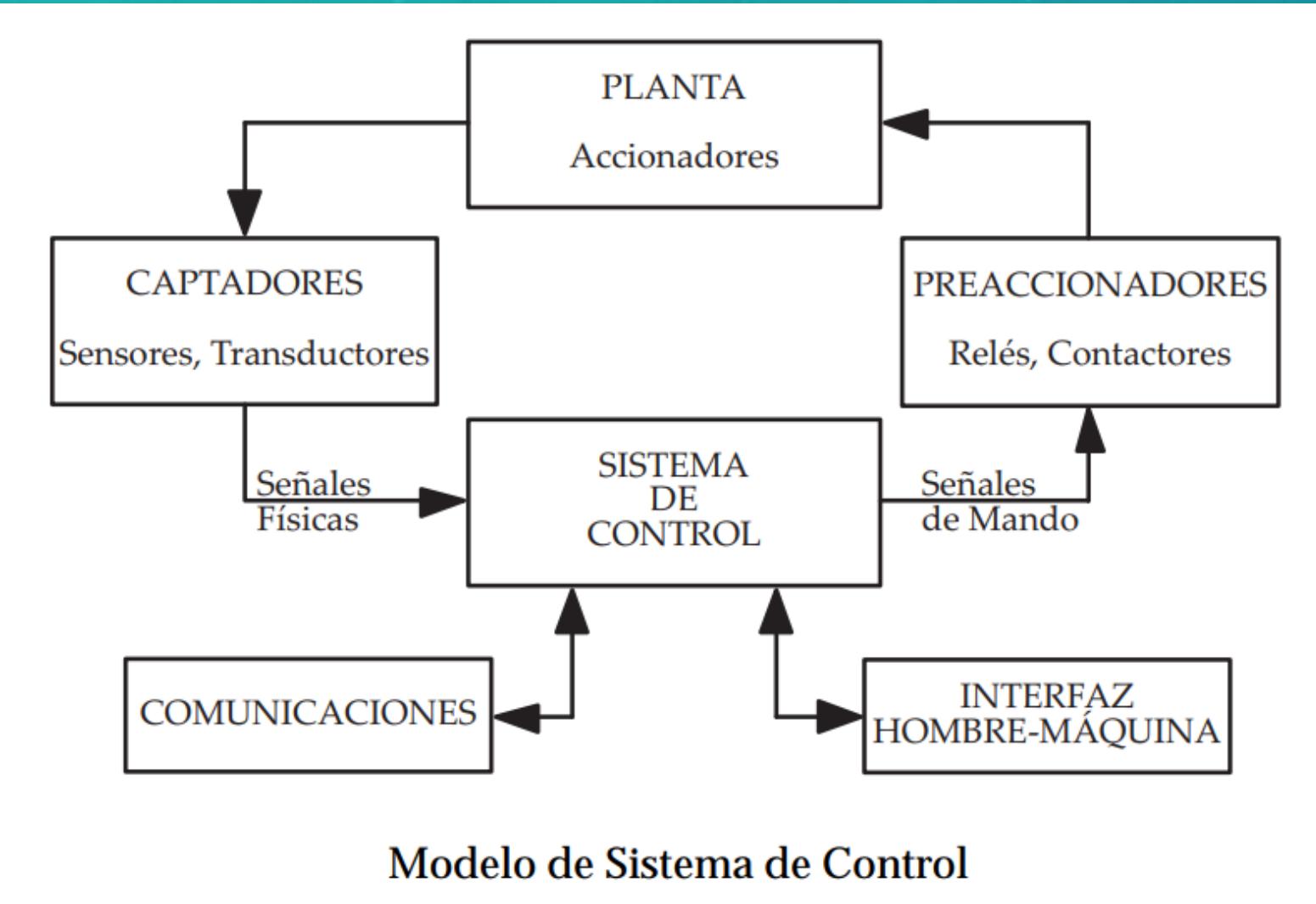
AUTOMATISMOS DESDE UN PUNTO DE VISTA ESTRUCTURAL

- Desde un punto de vista estructural, un automatismo se compone de dos partes claramente diferenciables:
- Parte Operativa
- Parte de Control

- Parte Operativa: Formada principalmente por el conjunto de dispositivos, máquinas y/o subprocesos diseñados para realizar determinadas funciones de producción y corresponden en su gran mayoría a elementos de potencia.
- Parte de Control: Formada por los elementos de procesamiento y/o mando, interfaz de comunicación y de diálogo con el hombre.

- El sometimiento de la parte operativa se logra mediante un intercambio continuo de información entre ésta y la parte de mando o control. Este flujo de información se establece mediante los captadores (sensores binarios, transductores análogos y digitales) y los preaccionadores (contactores, relés).
- Los captadores se encargan entonces de recoger datos de magnitudes físicas y de cambios de estado a controlar y envían dicha información a la parte de control para su procesamiento.

- La parte de control envía entonces acciones de mando a través de los preaccionadores, que son elementos que permiten el manejo de grandes potencias a partir de señales de baja potencia.



AUTOMATISMOS PROGRAMADOS

- Se implementan por medio de un programa que se ejecuta en un microprocesador.
- Las instrucciones de este programa determinan la función lógica que relaciona las entradas y las salidas. Se pueden distinguir 3 formas de implementación:
 - Autómata programable industrial.
 - Ordenador (PC industrial).
 - Microcontrolador.

AUTOMATISMOS PROGRAMADOS

Las ventajas más importantes de los automatismos programados son:

- Permiten una gran flexibilidad para realizar modificaciones o ampliaciones.
- Permiten implementar funciones de control y de comunicación complejas.
- Ocupan poco espacio.

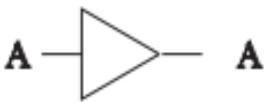
SISTEMAS DE EVENTOS DISCRETOS

- Se define un sistema dinámico de eventos discretos como aquel que evoluciona entre un numero finito de estados discretos y cuyo paso de un estado a otro depende del valor de determinadas variables binarias.

LÓGICA CABLEADA

- Los dispositivos lógicos pueden implementarse en base a componentes electrónicas, eléctricas, mecánicas, neumáticas, etc. Entre las componentes eléctricas tenemos interruptores y relés. Entre las electrónicas pueden mencionarse circuitos en base a diodos, transistores bipolares y de efecto de campo.

COMPUERTAS LÓGICAS



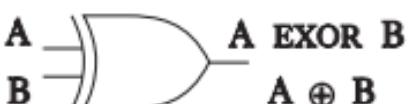
| A | A |
|---|---|
| 0 | 0 |
| 1 | 1 |



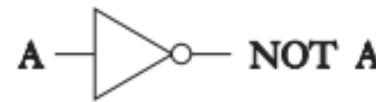
| A | B | AB |
|---|---|----|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



| A | B | A+B |
|---|---|-----|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |



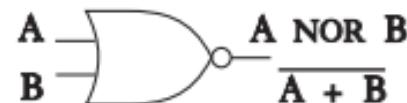
| A | B | A⊕B |
|---|---|-----|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



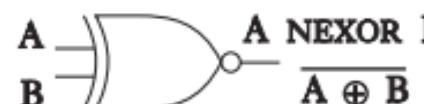
| A | NOT A |
|---|-------|
| 0 | 1 |
| 1 | 0 |



| A | B | A NAND B |
|---|---|----------|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



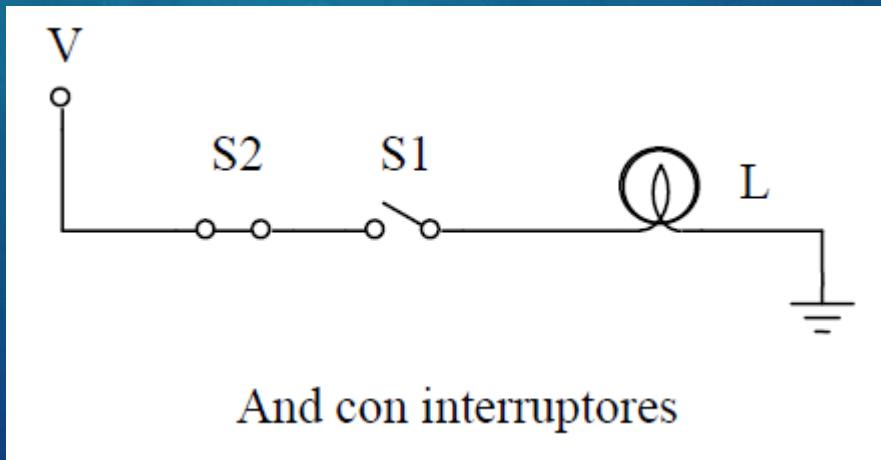
| A | B | A NOR B |
|---|---|---------|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |



| A | B | A⊕B |
|---|---|-----|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

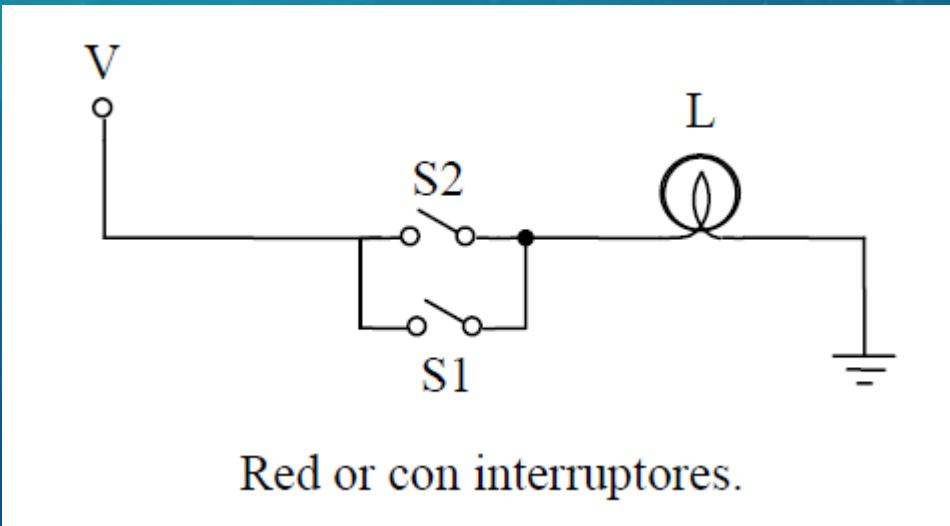
INTERRUPTORES MANUALES.

- *Circuito and.-* La figura se ilustra la implementación de la función and de dos variables, mediante interruptores manuales de dos posiciones. El foco se prende cuando ambos interruptores están cerrados. En la figura se muestra el interruptor 2 en posición cerrado.



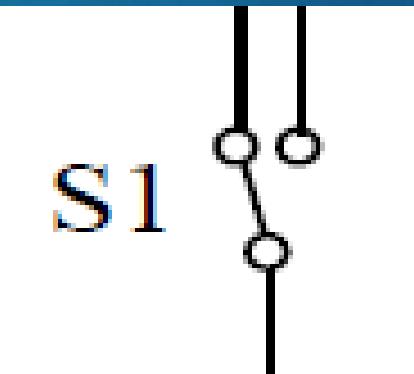
INTERRUPTORES MANUALES.

- *Circuito or.-* El siguiente diagrama ilustra la función or de dos variables, empleando interruptores manuales.



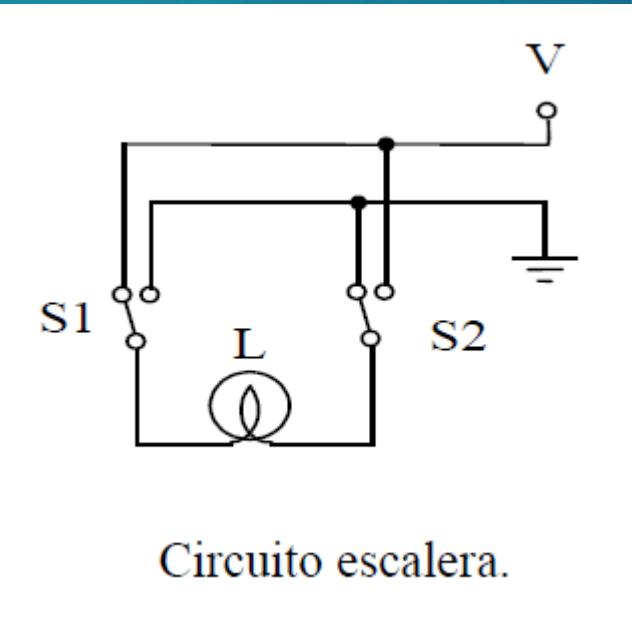
INTERRUPTORES MANUALES.

- *Circuito escalera.*
- En este caso se emplean interruptores con tres terminales, uno de ellos se denomina común. Cuando el común se conecta con el terminal izquierdo decimos que el interruptor está en valor lógico verdadero, y falso en caso contrario.



INTERRUPTORES MANUALES.

- *Circuito escalera*.- La figura se representa el circuito que se emplea en las escaleras para encender y apagar, desde dos lugares diferentes, el mismo foco.



INTERRUPTORES MANUALES.

- **Relés.** En un relé electromagnético cuando circula corriente a través de la bobina se producen fuerzas que causarán que la conexión entre el contacto nc (normalmente cerrado, closed) y el común se abra, y se efectúe la conexión entre el común y el terminal denominado no (normalmente abierto, open). Cuando no circula corriente por la bobina, existe un resorte mecánico que retorna el contacto móvil a la posición nc. En la siguiente figura se dibujan los contactos en su posición normal, con bobina desenergizada.

INTERRUPTORES MANUALES.

- Relés.

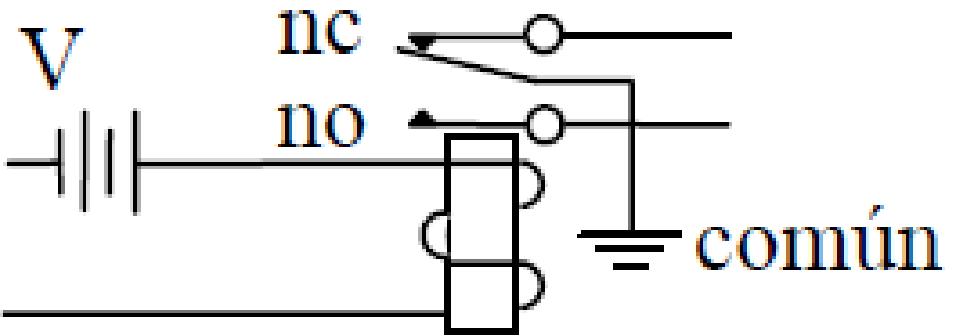


Diagrama de un relé.

ALGEBRA DE BOOLE

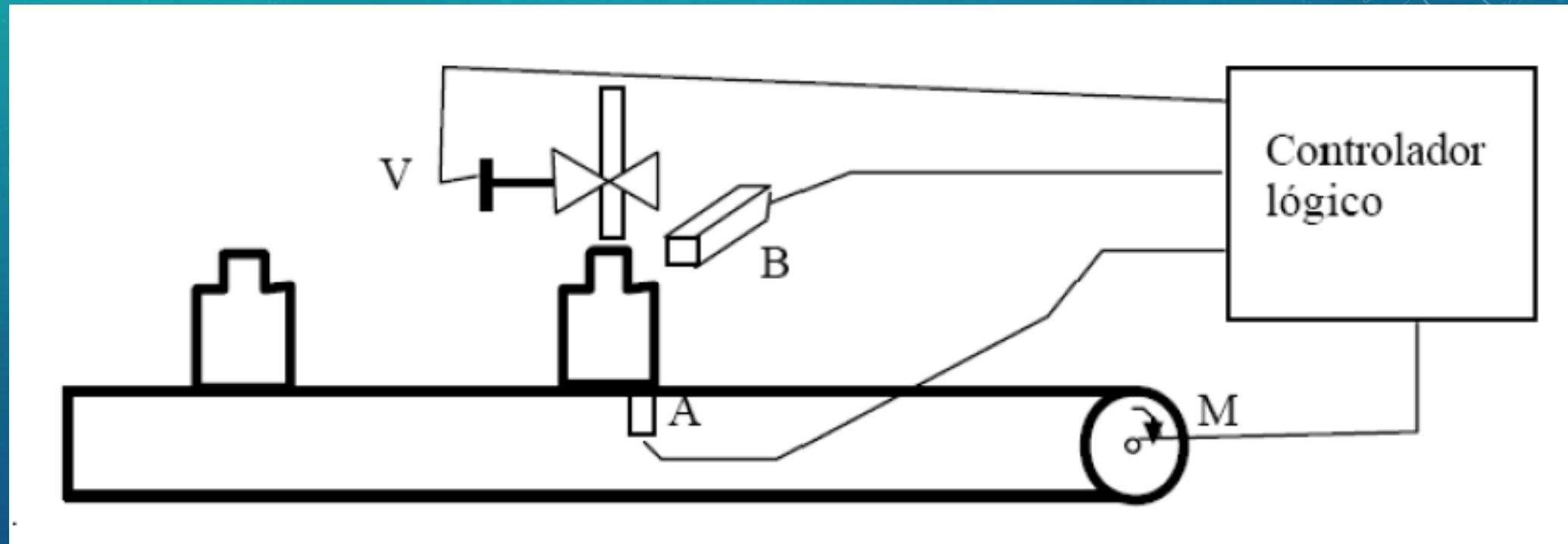
$$F = \overline{XY} + \overline{XZ} + XYZ$$

$$F = A\overline{B}(\overline{C} + \overline{D}) + ABC\overline{C} + \overline{C} \cdot \overline{D}$$

EJEMPLO: PROCESO DE LLENADO DE BOTELLAS

- Como ejemplo considérese la embotelladora que se muestra en la figura. Existen 2 señales dadas por 2 sensores. La señal A es 1 si tiene una botella encima y 0 en caso contrario. La señal B es 1 si el líquido que llena la botella ha llegado al nivel máximo y 0 en caso contrario. Asimismo existen otras 2 señales. Cuando M se pone a 1, el motor se pone en marcha, y cuando se pone a 0, se para. Cuando V se pone a 1, la electroválvula se abre empezando a llenar la botella, cerrándose cuando V = 0.

EJEMPLO: PROCESO DE LLENADO DE BOTELLAS



GRAFCET

GRAFCET - IEC 848

- El Grafcet (Graphe de commande etape-transition) es una herramienta de modelado de sistemas de eventos discretos derivada de las redes de Petri. En realidad, es como una red de Petri en la que los lugares solo pueden tener una marca. Es una herramienta adecuada para representar sistemas con evoluciones simultaneas.

SÍMBOLOS

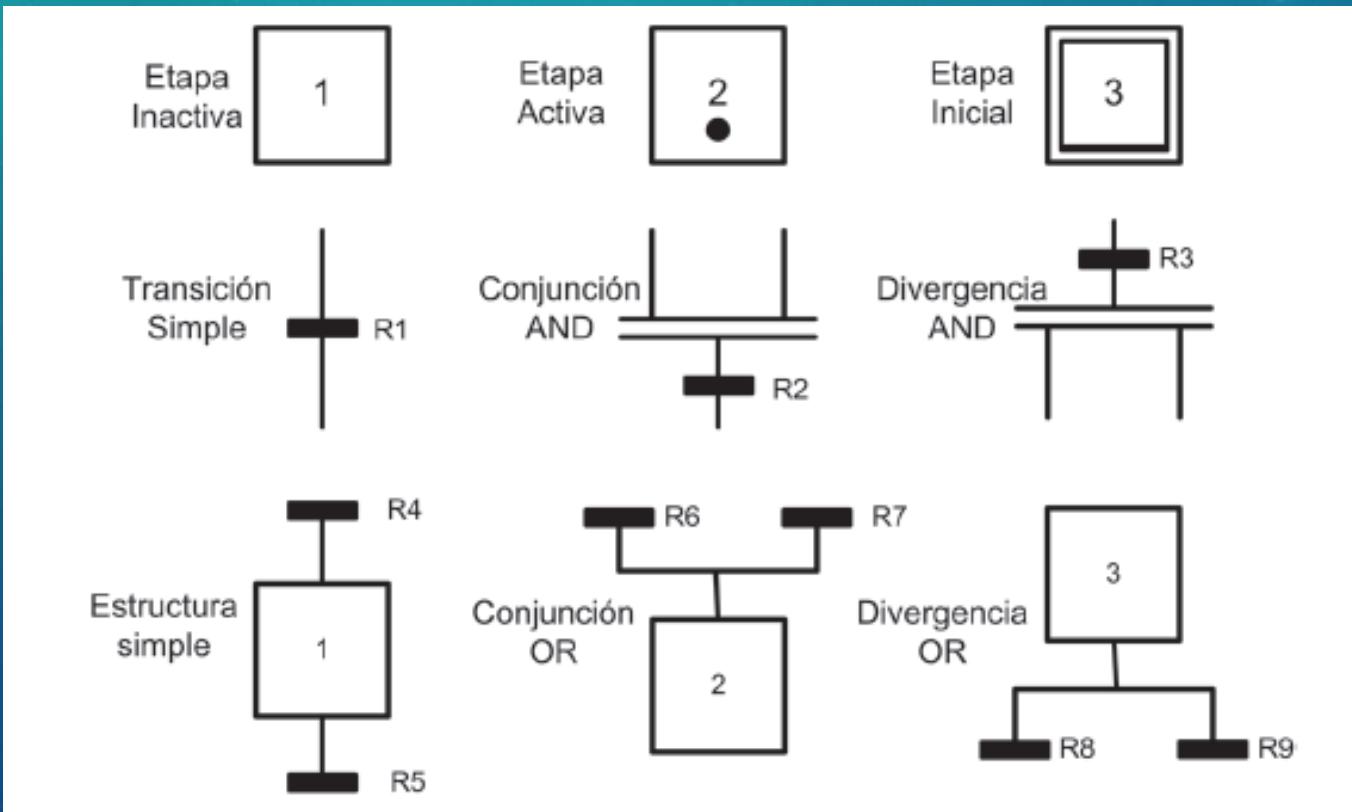
Etapas:

La evolución de un proceso representada mediante un gráfico Grafcet, está formada por una sucesión de etapas que representan cada uno de sus estados, llevando cada una de ellas asociada una o varias acciones a realizar sobre el proceso.

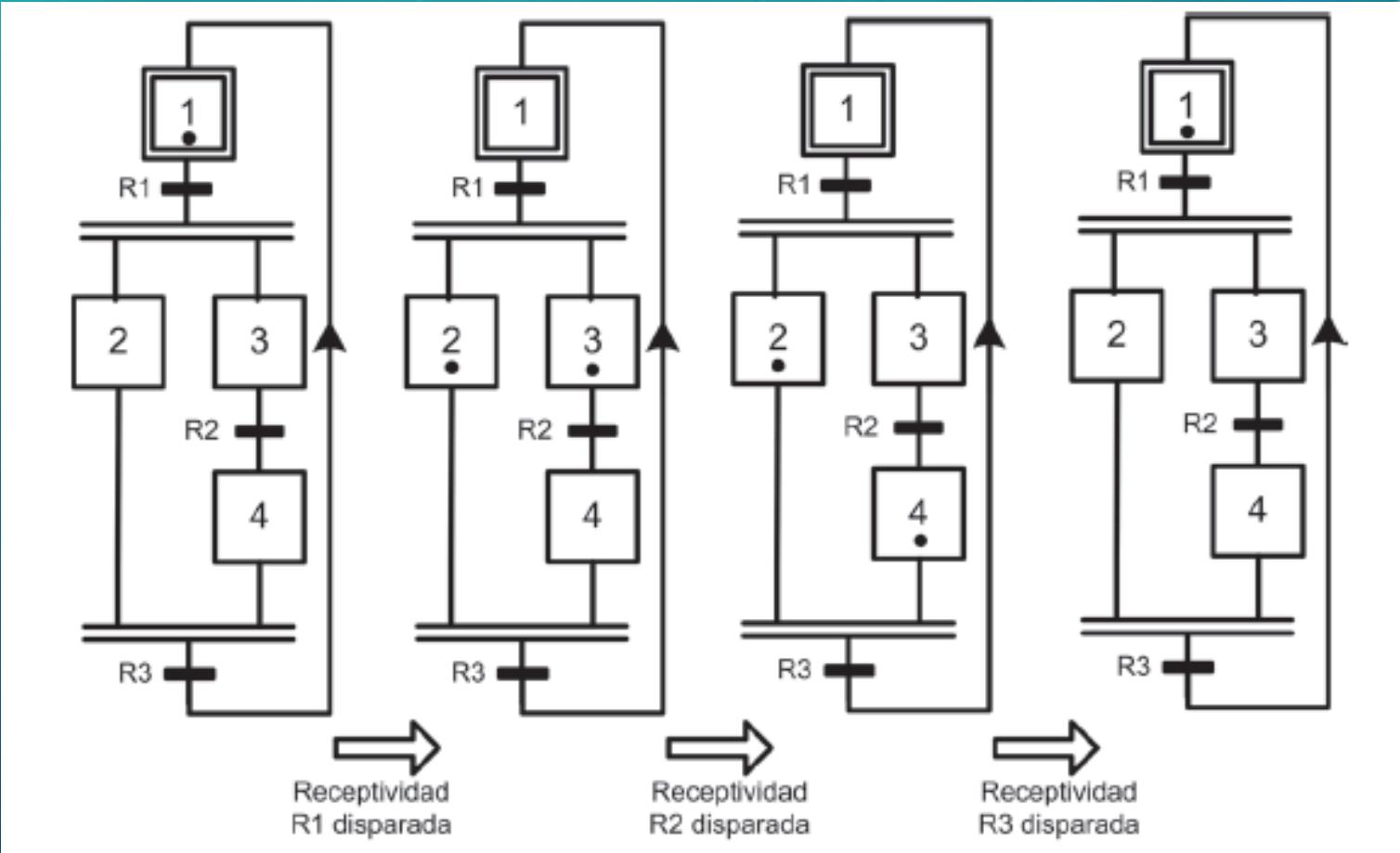
SÍMBOLOS

Las etapas se representan con un cuadro y un número o símbolo con un subíndice numérico en su interior, en ambos casos el número indica el orden que ocupa la etapa dentro del Grafcet

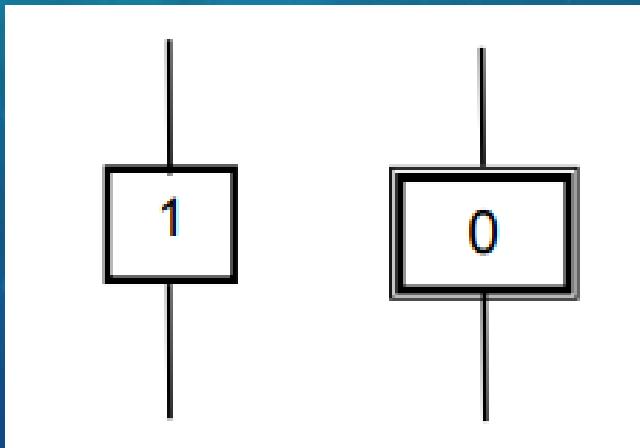
- La figura muestra la representación básica de un grafcet, los lugares de una RP se interpretan como etapas en un grafcet y las transiciones, como receptividades.



- En la figura se aprecia la evolución de un grafcet cuando se disparan las receptividades R1, R2 y R3 en el orden mencionado.



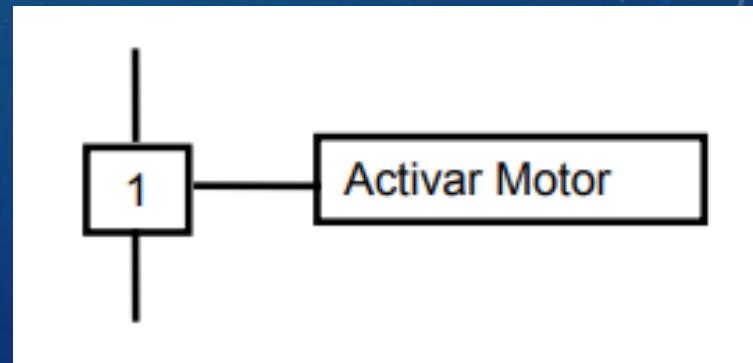
- Las etapas iniciales, aquellas en las que se posiciona el sistema al iniciarse el proceso, se representan con un cuadro doble.



Acción asociada

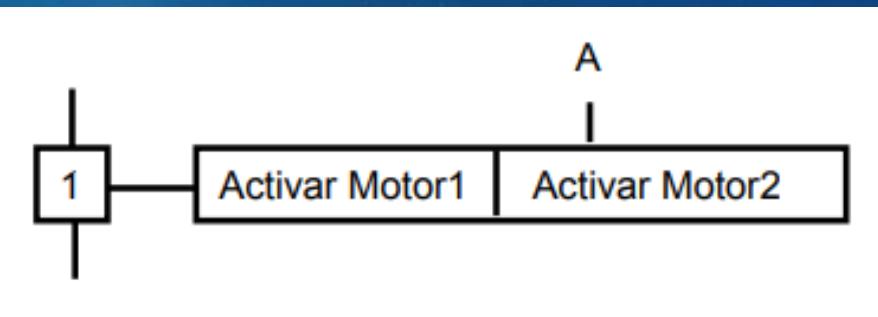
Son una o varias acciones a realizar sobre el proceso, cuando la etapa de la cual dependen dichas acciones se encuentra activada.

Dichas acciones correspondientes a una etapa, se simbolizan mediante rectángulos conectados y situados a la derecha de dicha etapa. En el interior de estos rectángulos se indica, bien de forma literal, bien de forma simbólica, las acciones a realizar.



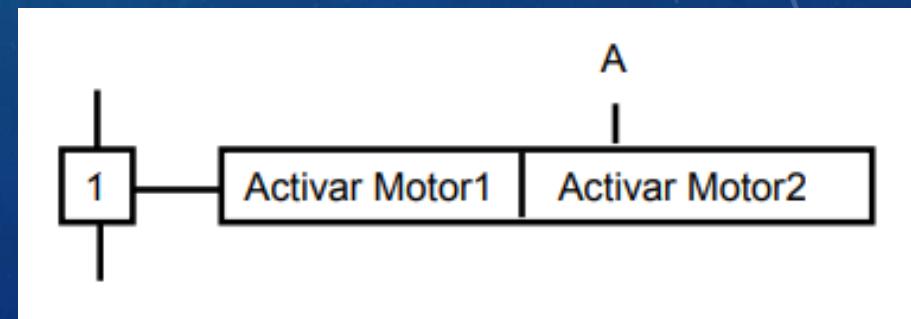
Clasificación de las acciones:

- Tipo de activación:
 - Incondicionales : acciones que se ejecutan con sólo quedar activadas las etapas correspondientes.
 - Condicionales : son las acciones que necesitan el cumplimiento de una condición además de la propia activación de la etapa correspondiente.

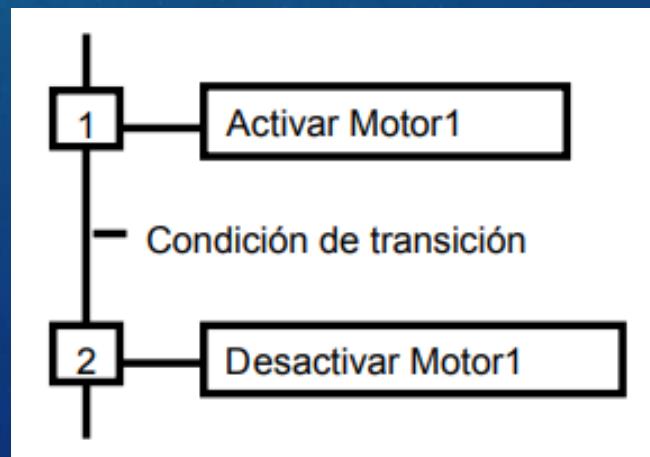


Clasificación de las acciones:

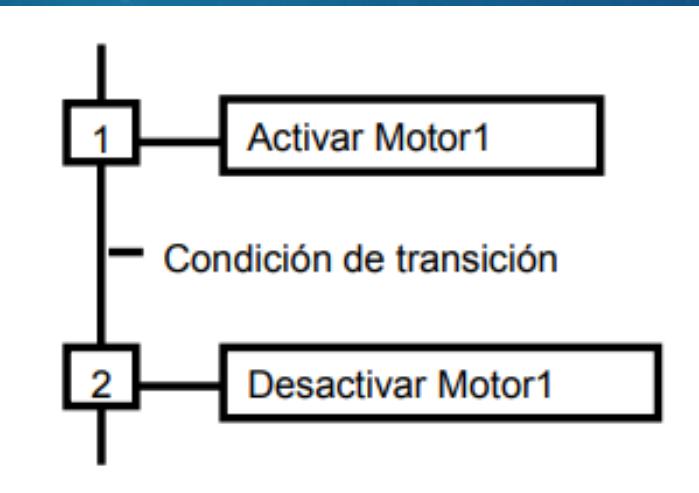
- Objeto Acción :
 - Internas : acciones que se producen en el equipo de control, por ejemplo temporizaciones, contadores, cálculos matemáticos, etc.
 - Externas : las acciones que se producen sobre el proceso, por ejemplo abrir o cerrar una válvula, activar o desactivar una bomba, etc.



- Transición y Condición de transición
- En el diagrama Grafcet, un proceso se compone de una serie de etapas secuenciales que se activan una tras otra unidas mediante una transición.
- El paso de una etapa a la siguiente se realiza dependiendo de si se cumple o no la condición de transición entre ellas.

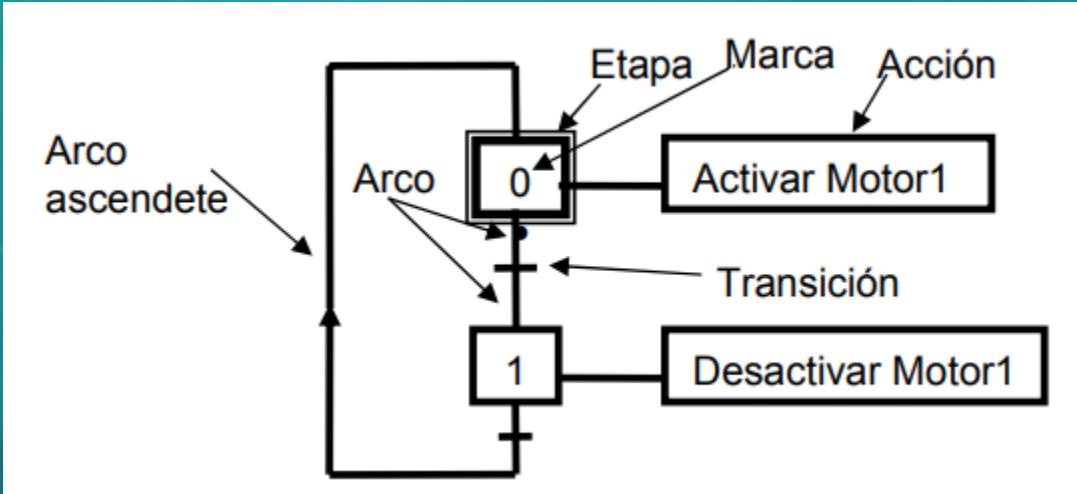


- Toda transición lleva asociada una condición de transición o función lógica booleana que se denomina receptividad, y que puede ser verdadera o falsa.
- Se dice que la transición está validada, cuando la etapa o etapas anteriores a la transición están activadas. El franqueamiento de la transición se producirá si, y sólo si, la transición esta validada y la receptividad es verdadera.



- Arco: Es el segmento de recta que une una transición (con su condición de transición) con una etapa y viceversa. Los arcos pueden ser verticales u horizontales, además los arcos verticales deben llevar una flecha indicando su sentido en el caso de ser éste ascendente.

- Esquema de elementos que componen el Grafcet



- La situación de etapa activada, se indica mediante la colocación de una marca en el interior del gráfico representativo de la etapa

REGLAS DE EVOLUCIÓN DEL GRAFCET

- Reglas
- 1) La etapa inicial de un Grafcet se activa de forma incondicional. Esta situación inicial se corresponde en general con una situación de reposo.

REGLAS DE EVOLUCIÓN DEL GRAFCET

- 2) Una transición está en disposición de ser validada cuando todas las etapas inmediatamente precedentes, unidas a dicha transición, están activadas. La activación de una transición se produce cuando está validada y la condición de transición o receptividad es verdadera. Se podría definir una etapa como activable cuando la transición precedente está validada.

REGLAS DE EVOLUCIÓN DEL GRAFCET

- 3) Franquear una transición implica la activación de todas las etapas siguientes inmediatas, y la desactivación de las inmediatas precedentes.

REGLAS DE EVOLUCIÓN DEL GRAFCET

- 4) Transiciones conectadas en paralelo, se activan de forma simultánea si se cumplen las condiciones para ello.
- 5) Una o varias acciones se asocian a cada etapa. Estas acciones sólo están activas cuando la etapa está activa.

ESTRUCTURAS BASE

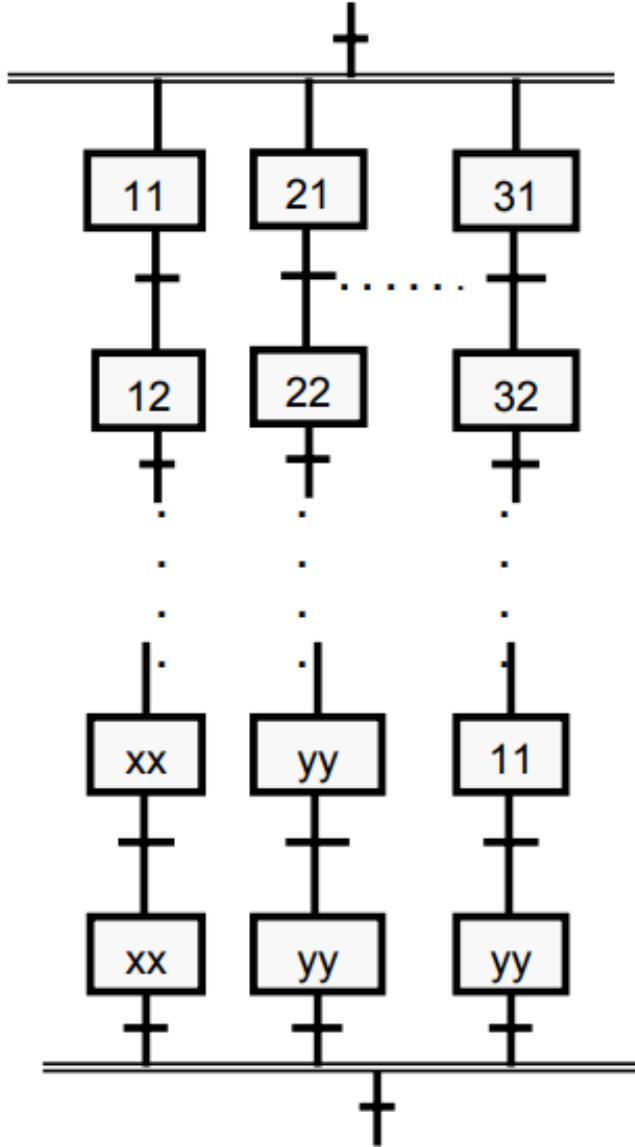
- Grafset soporta diferentes tipos de estructura secuencial:
 - Estructura base: trata conceptos de secuencialidad y concurrencia
 - Estructura lógica: trata conceptos de concatenación de estructuras

ESTRUCTURAS BASE

- Estructuras de secuencia única
 - Son estructuras formadas por secuencias de etapas que se van activando una tras otra, sin interacción con ninguna otra estructura
- Estructuras de secuencia paralela
 - Son un conjunto de estructuras únicas activadas por una misma transición de forma simultánea. Después de la activación de las distintas secuencias su evolución se produce de forma independiente



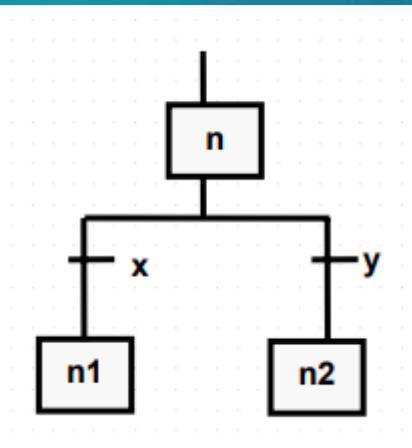
Estructura de secuencia única



Estructura de secuencias paralelas

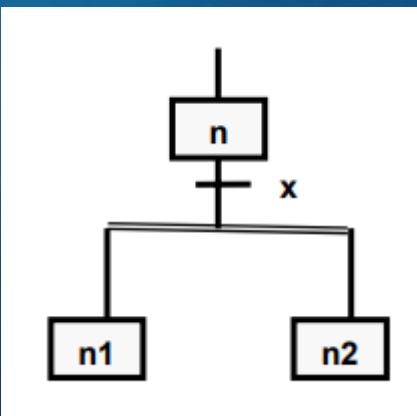
ESTRUCTURAS LÓGICAS

- Divergencia OR
- Se utiliza cuando lo que se trata es de modelar la posibilidad de tomar dos o más secuencias alternativas a partir de una etapa común.

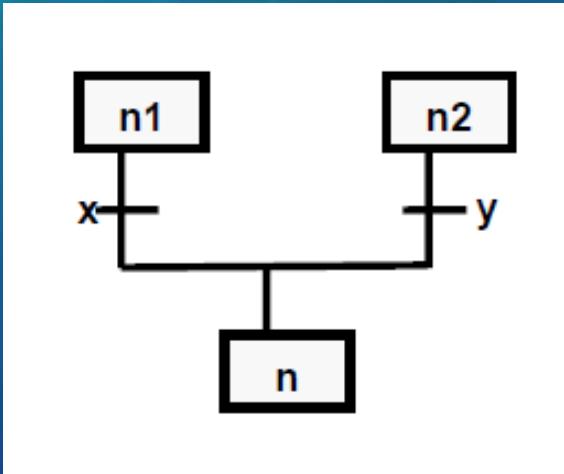


- La etapa n1 pasará a estar activa si estando activa la etapa n, se satisface la condición de transición o receptividad x. De igual forma la etapa n2 pasará a estar activa si estando activa la etapa n se satisface la condición de transición o receptividad y.

- Divergencia AND
- Permite la implementación de procesos concurrentes síncronos, de forma que dos o más subprocessos del sistema, representados por las secuencias paralelas, pueden activarse de forma sincronizada.
- La etapa n₂ y n₁ pasarán al estado activo, si estando activa la etapa n se satisface la condición de transición o receptividad x



- Convergencia OR
- La etapa n pasará a estar activa, si estando activa la etapa n1 se satisface la condición de transición o receptividad x; o si estando activa la etapa n2 se satisface la condición de transición o receptividad y.



- Convergencia en AND
- La etapa n pasará a estar activa, si estando las etapas n_1 y n_2 activas se satisface la condición de transición o receptividad x

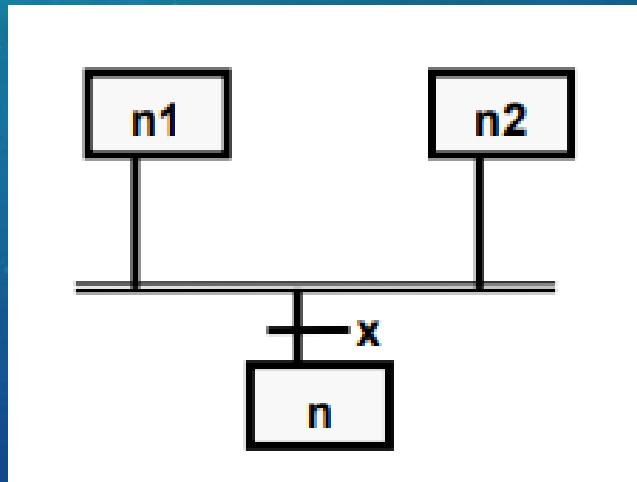


DIAGRAMA DE EJEMPLO

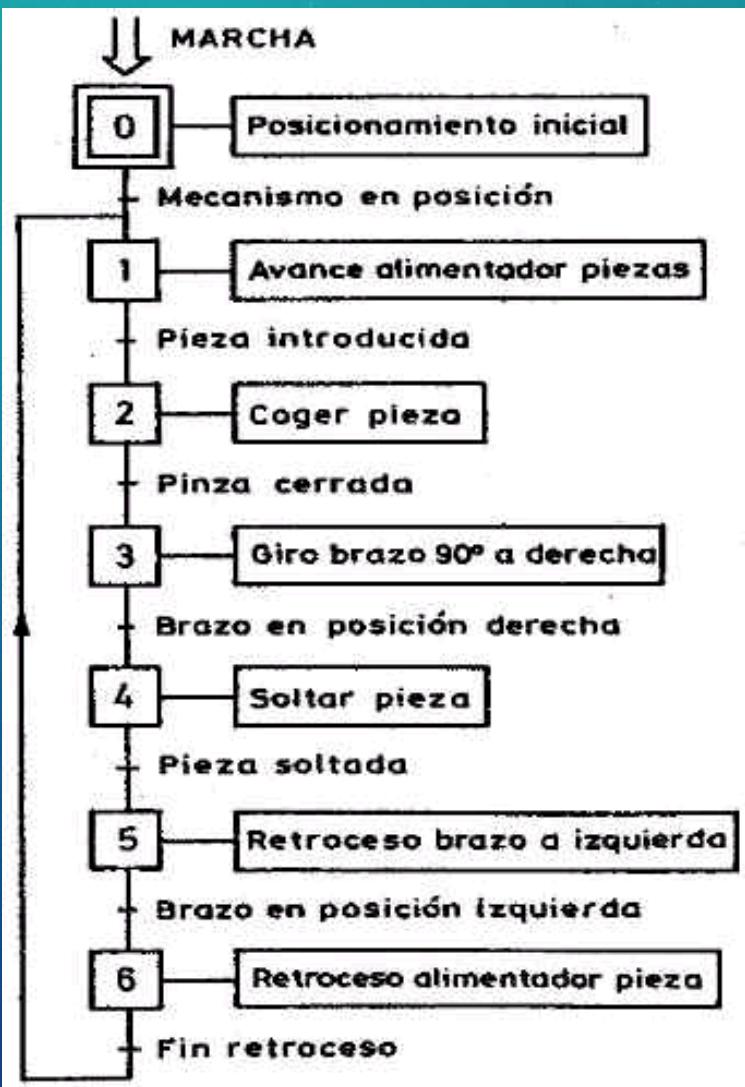


Diagrama de ejemplo

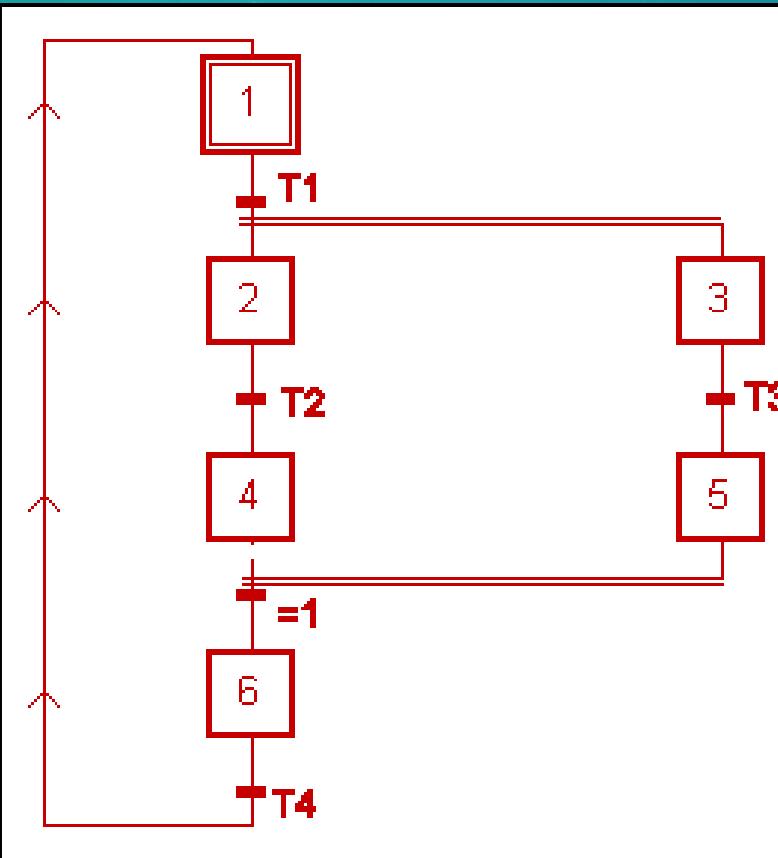
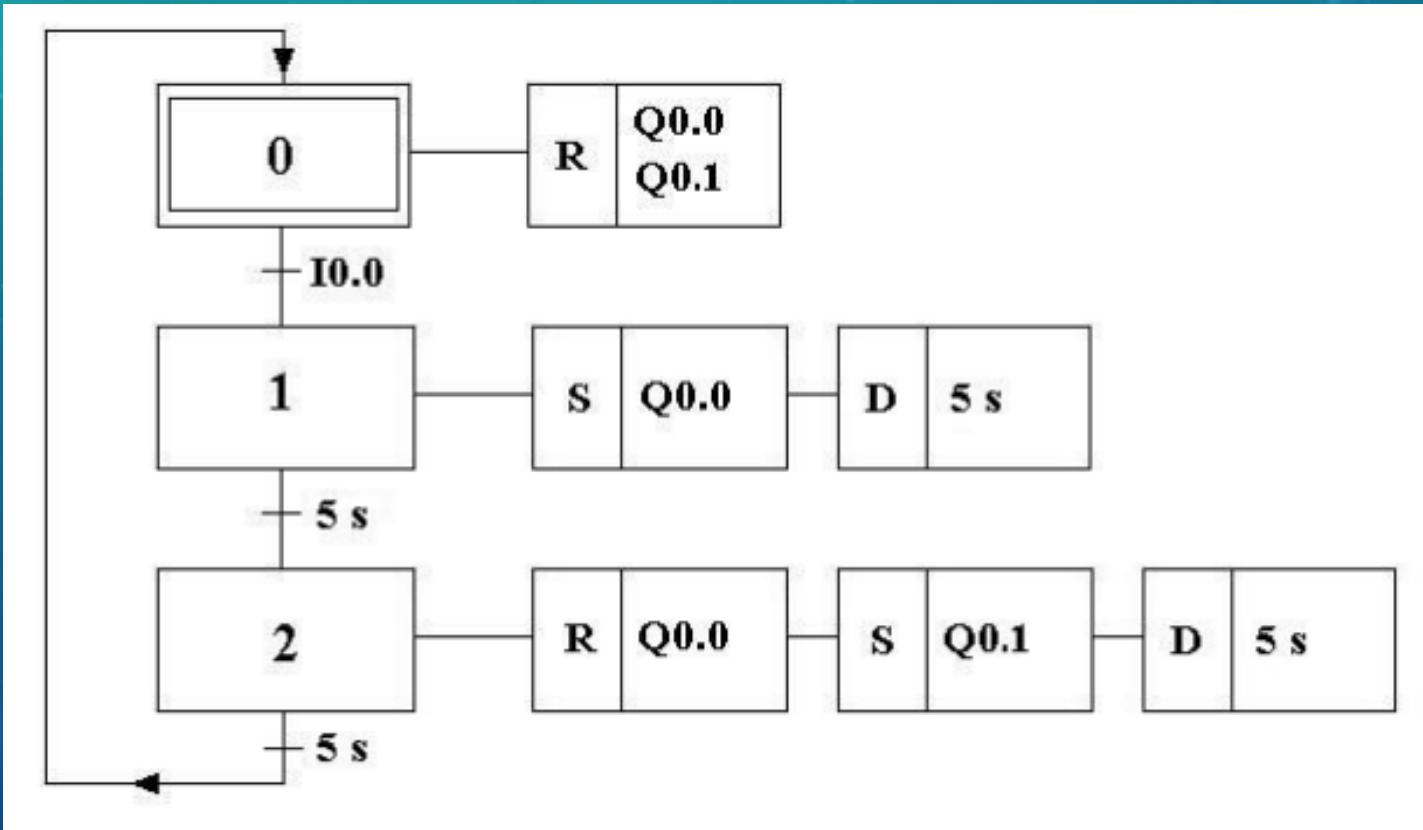
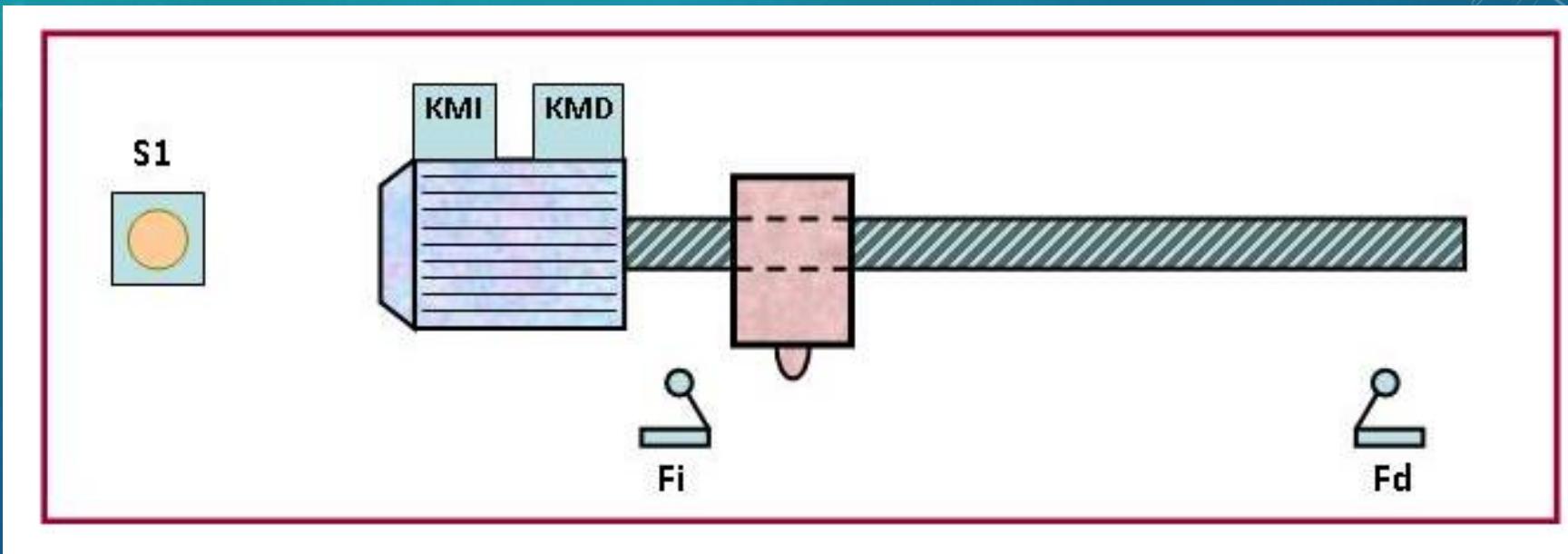
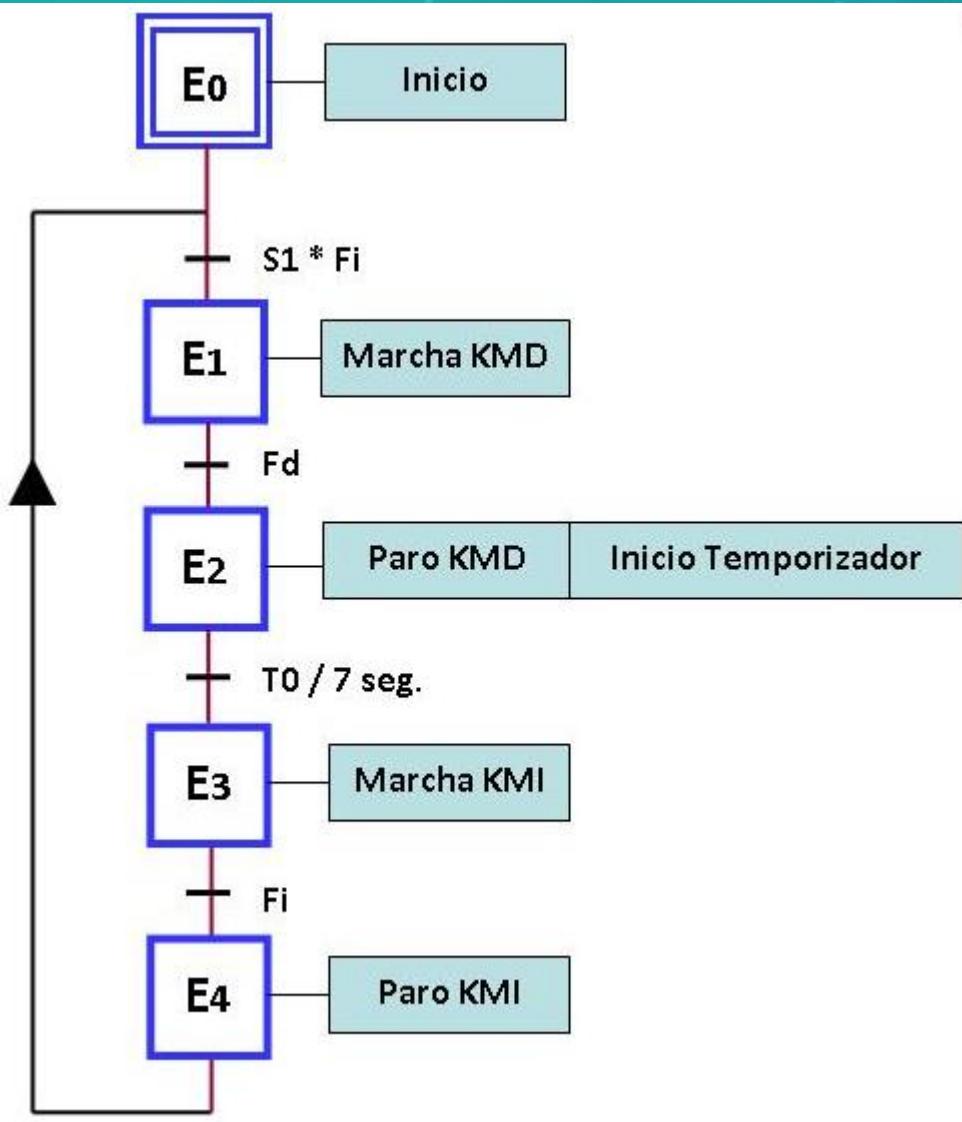


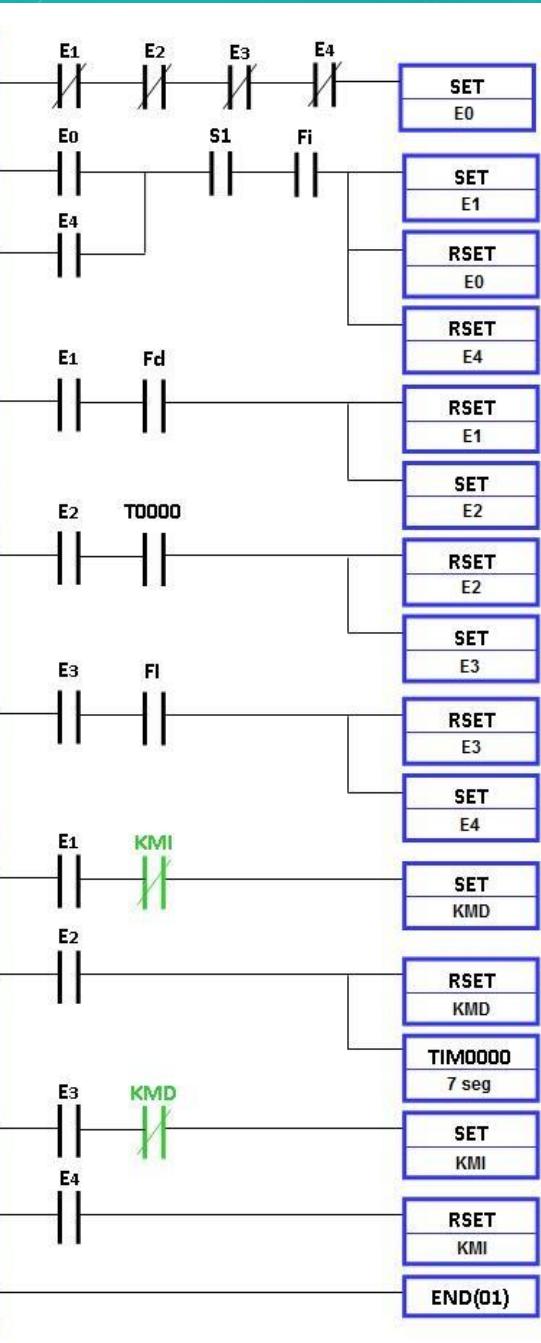
Diagrama de ejemplo



EJERCICIO DE EJEMPLO:



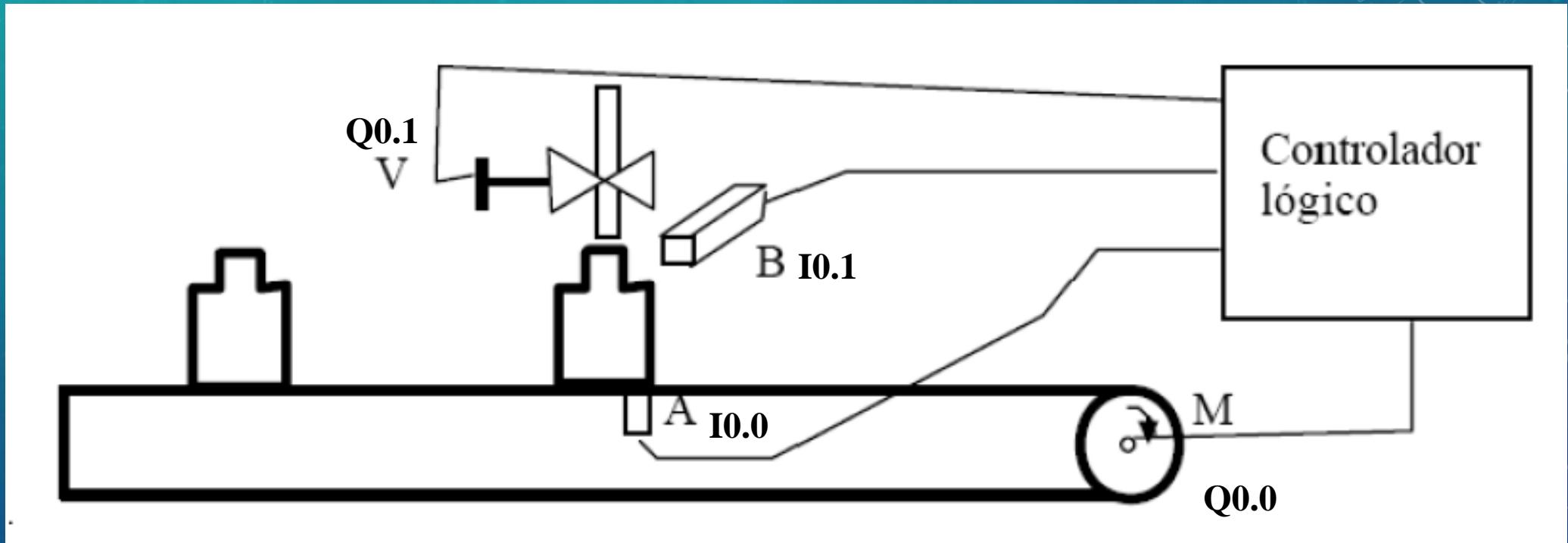




EJEMPLO: PROCESO DE LLENADO DE BOTELLAS

- Como ejemplo considérese la embotelladora que se muestra en la figura. Existen 2 señales dadas por 2 sensores. La señal A es 1 si tiene una botella encima y 0 en caso contrario. La señal B es 1 si el líquido que llena la botella ha llegado al nivel máximo y 0 en caso contrario. Asimismo existen otras 2 señales. Cuando M se pone a 1, el motor se pone en marcha, y cuando se pone a 0, se para. Cuando V se pone a 1, la electroválvula se abre empezando a llenar la botella, cerrándose cuando V = 0.

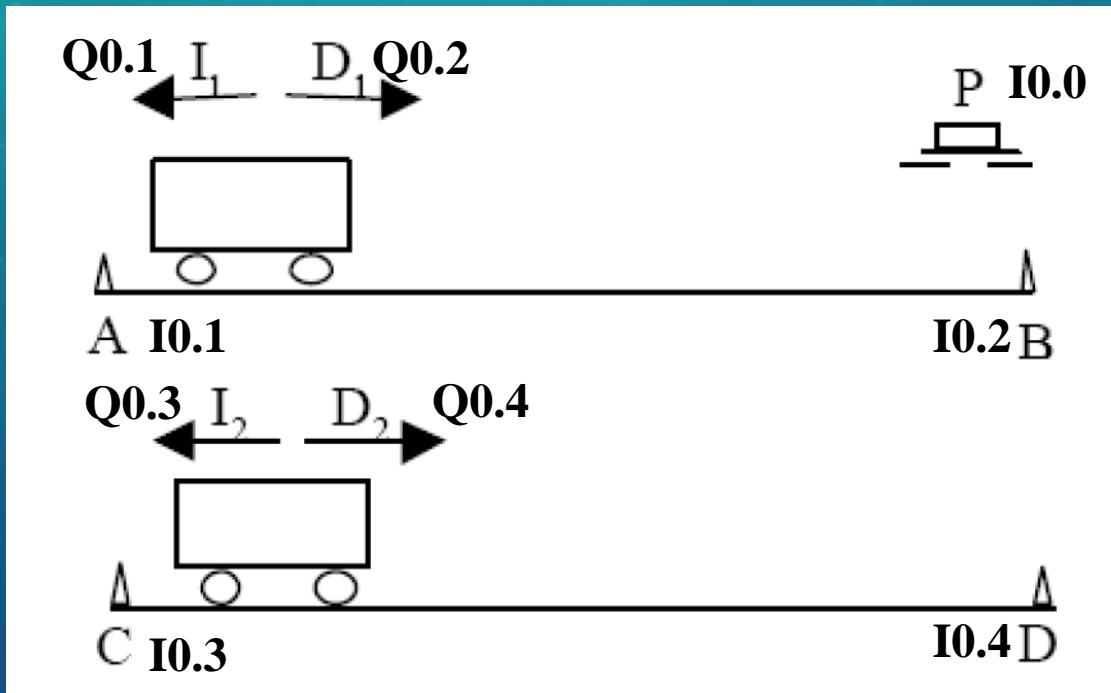
EJEMPLO: PROCESO DE LLENADO DE BOTELLAS



EJERCICIO

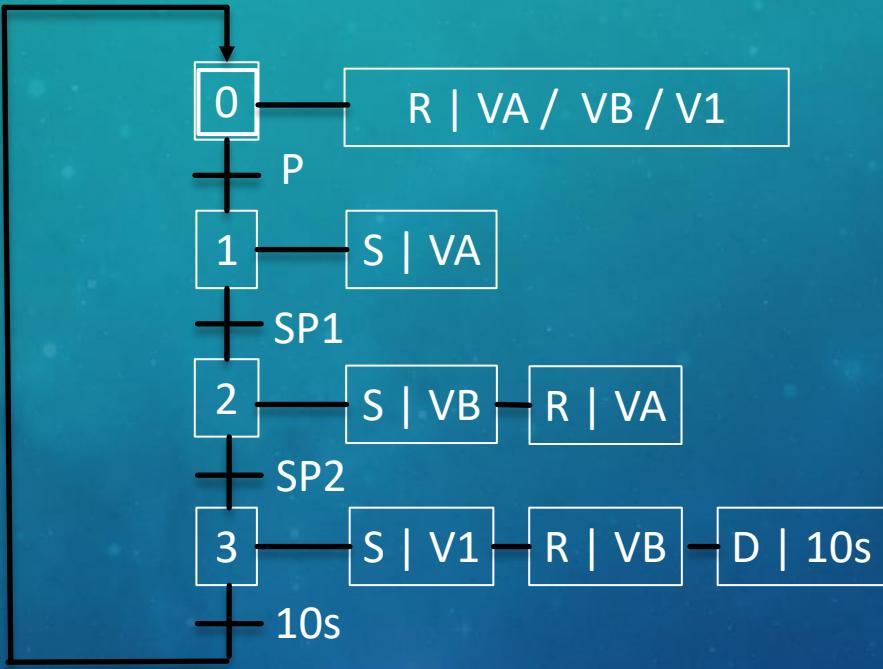
- En la siguiente figura los carros están inicialmente en A y C. Cuando se pulsa P, se ponen en marcha hacia la derecha hasta llegar a B y D. Cada carro, al llegar al extremo derecho, cambia de sentido de forma independiente poniéndose en marcha hacia la izquierda hasta llegar a C o A.

EJEMPLO:

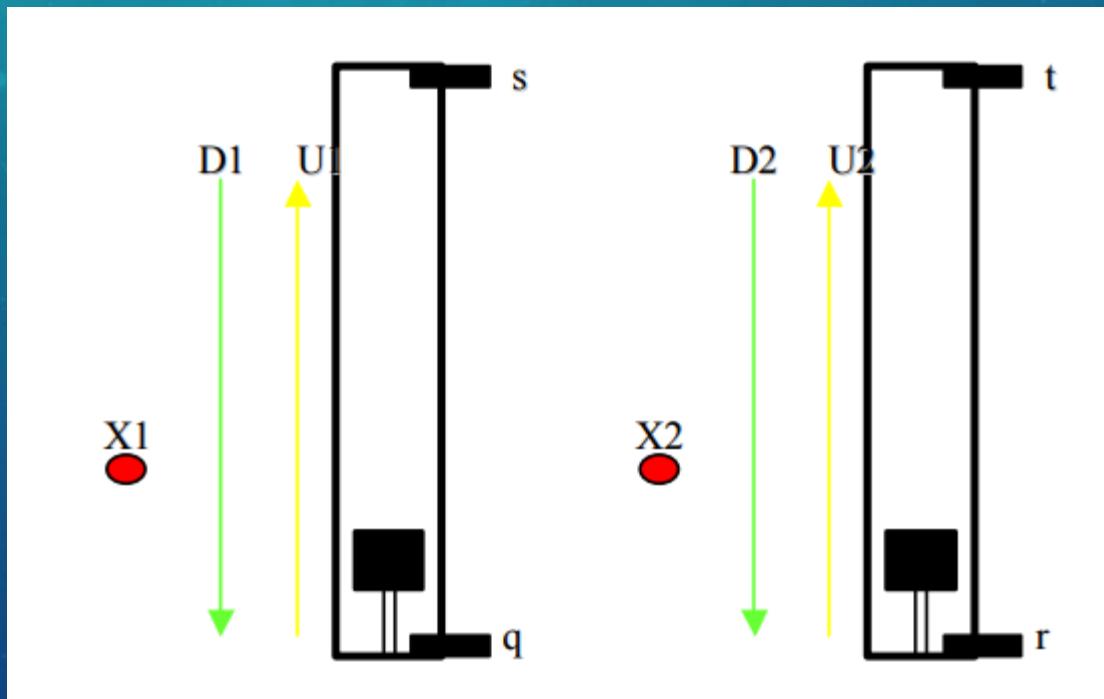


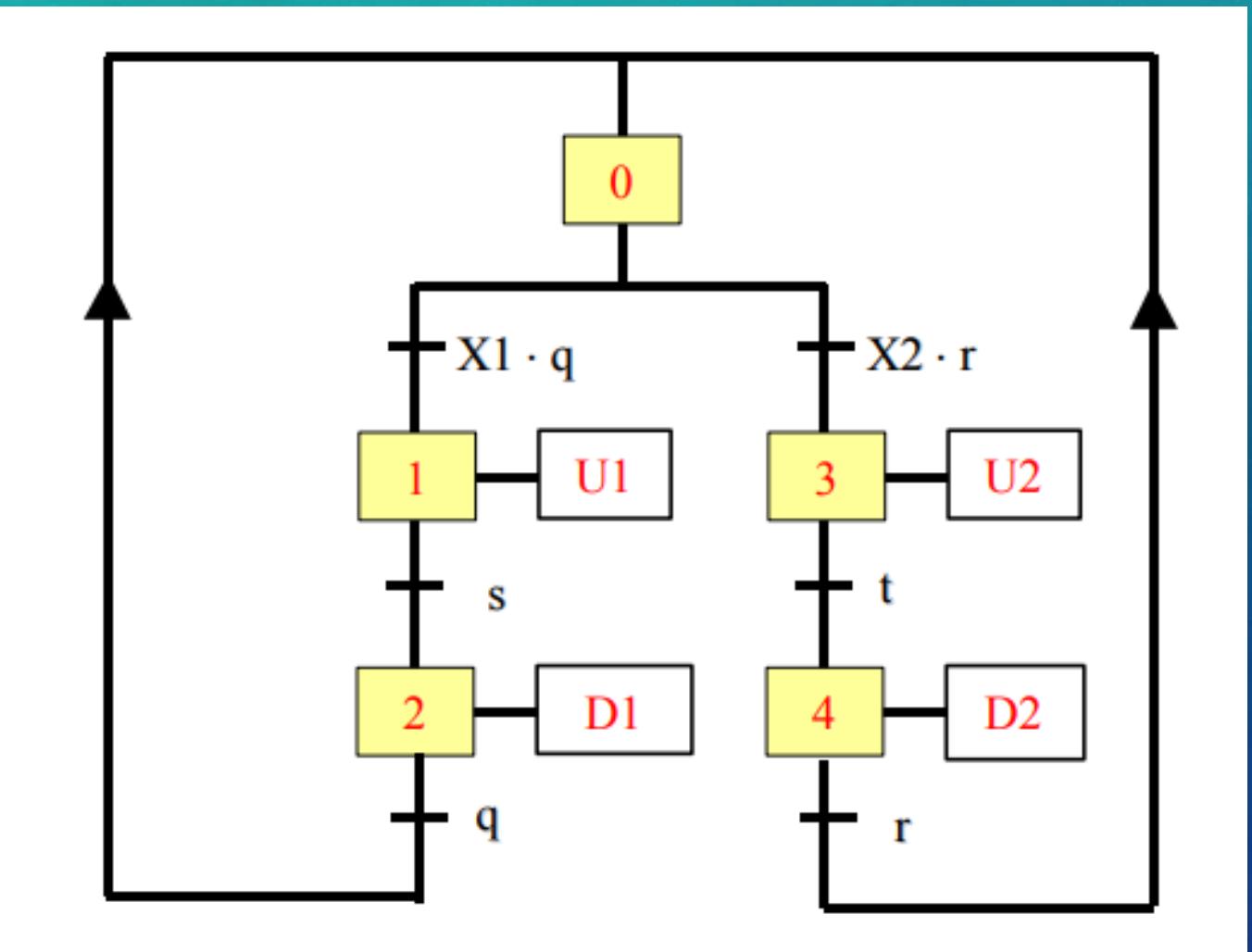
EJERCICIO

- Ejemplo Básico: Se quieren mezclar 2 productos. La dosificación de los dos productos se realiza en una tolva, primero se vierte el producto A sobre la tolva hasta que se alcanza un nivel SP1 y a continuación se añade el producto B para conseguir el nivel total de los 2 productos, SP2. Se abre la válvula v1 durante 10 segundos para vaciar el depósito y para poder iniciar un nuevo ciclo. El proceso se activa con un interruptor P.



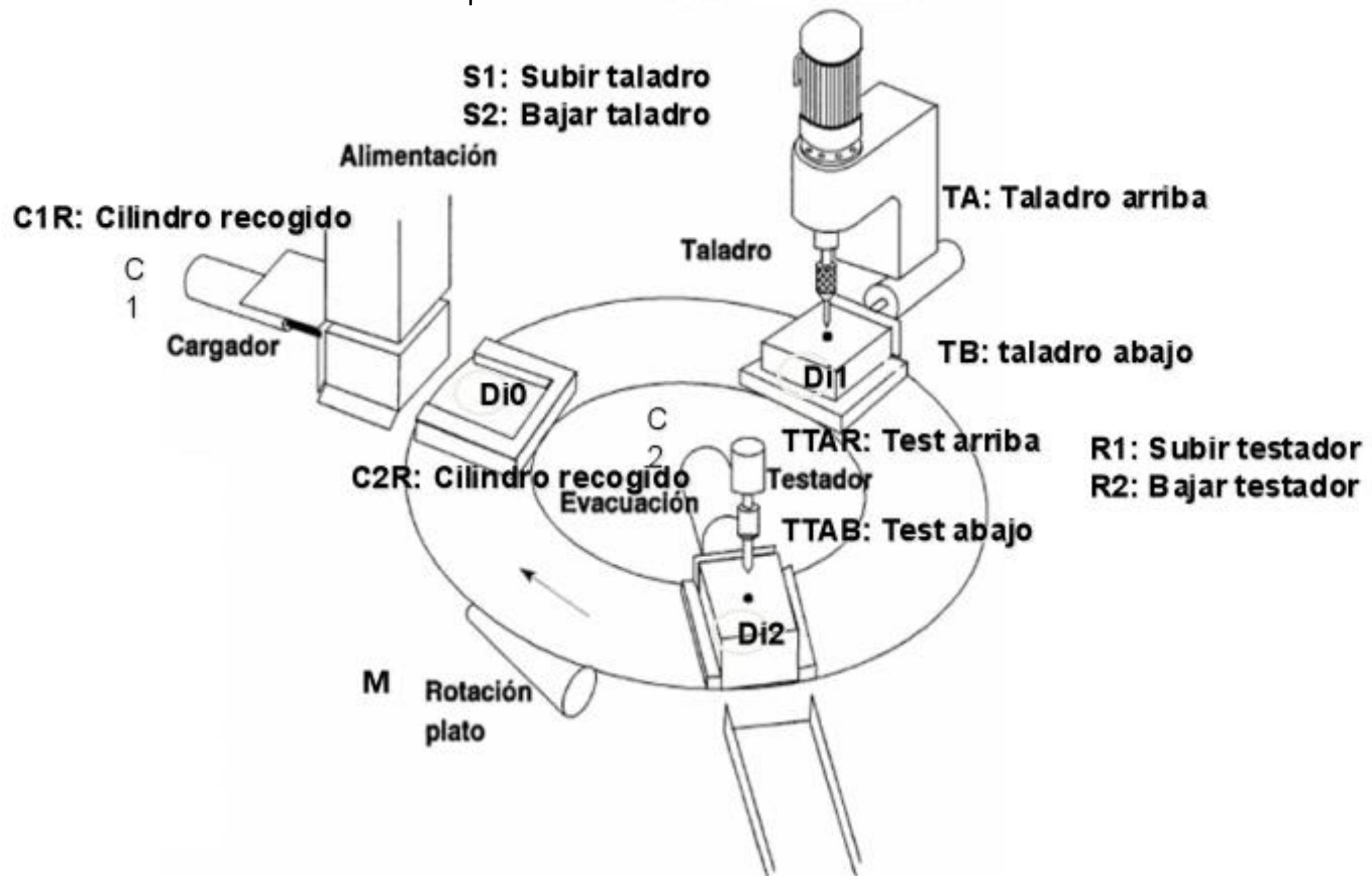
- En el siguiente ejemplo dos montacargas se mueva cuando se pulsa un botón (x_1 o x_2) en sentido ascendente hasta el final de recorrido (s,t) y de inmediato se desciende a la situación inicial (q,r).





Círculo de control de una unidad de perforación

MT: Motor taladro



La máquina dispone de tres estaciones:

- Alimentación de piezas.
- Taladrado.
- Comprobación dureza y expulsión.

Funcionamiento:

- Las piezas se van cargando en la estación de alimentación.
- A continuación, el plato de la máquina gira 120º. Cuando el movimiento se ha completado se detecta la señal T.
- Cuando existe una pieza en la estación de taladrado, se taladra.
- Finalizada la operación, la máquina vuelve a girar.
- Se acciona el cilindro para expulsar la pieza.

Nota: Nótese que los procesos se deben llevar a cabo de forma concurrente, para acelerar el proceso en general.

Botonera de mando:

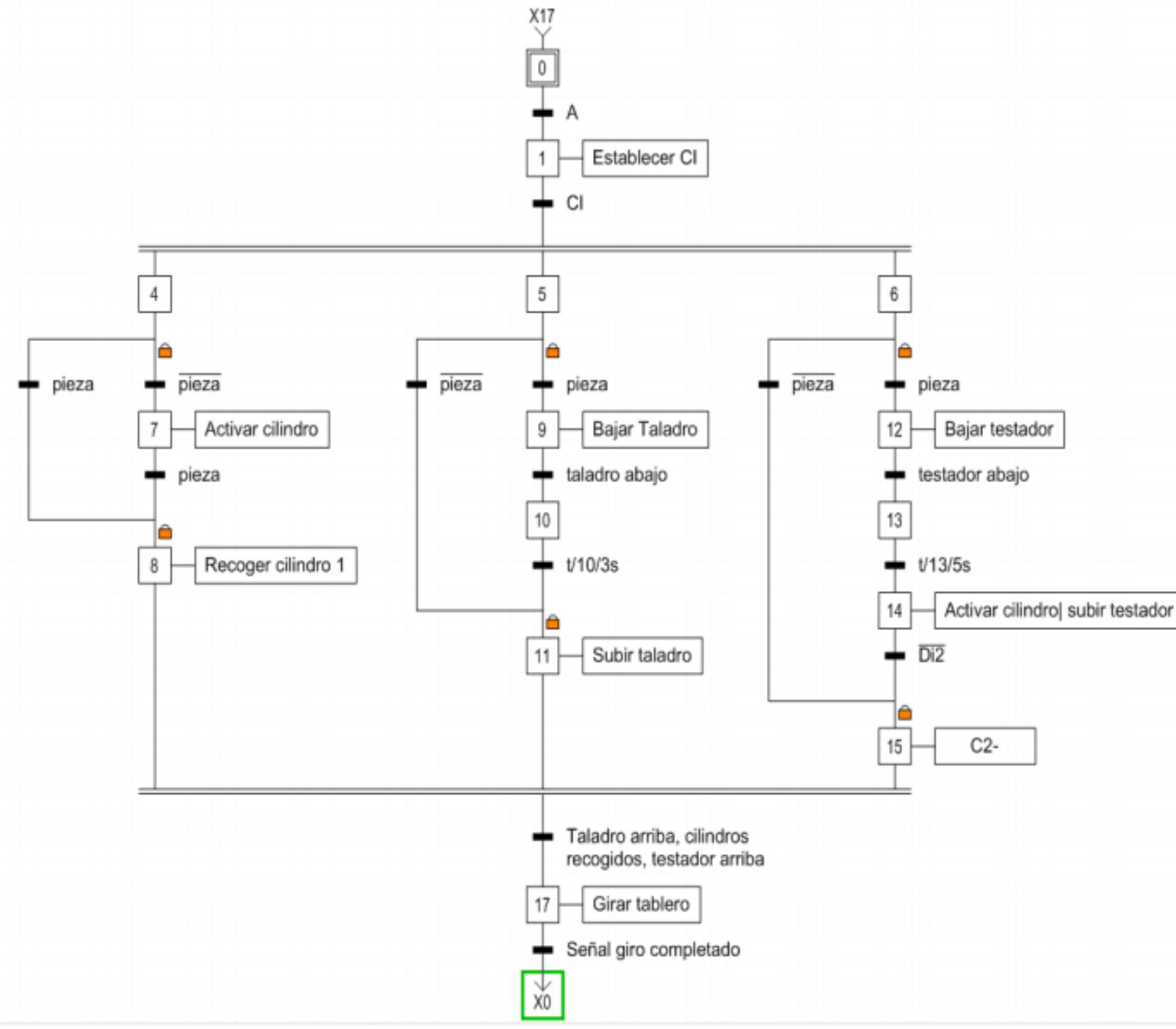
- Pulsador M
- Botón A (con enclavamiento)
- Pulsador S

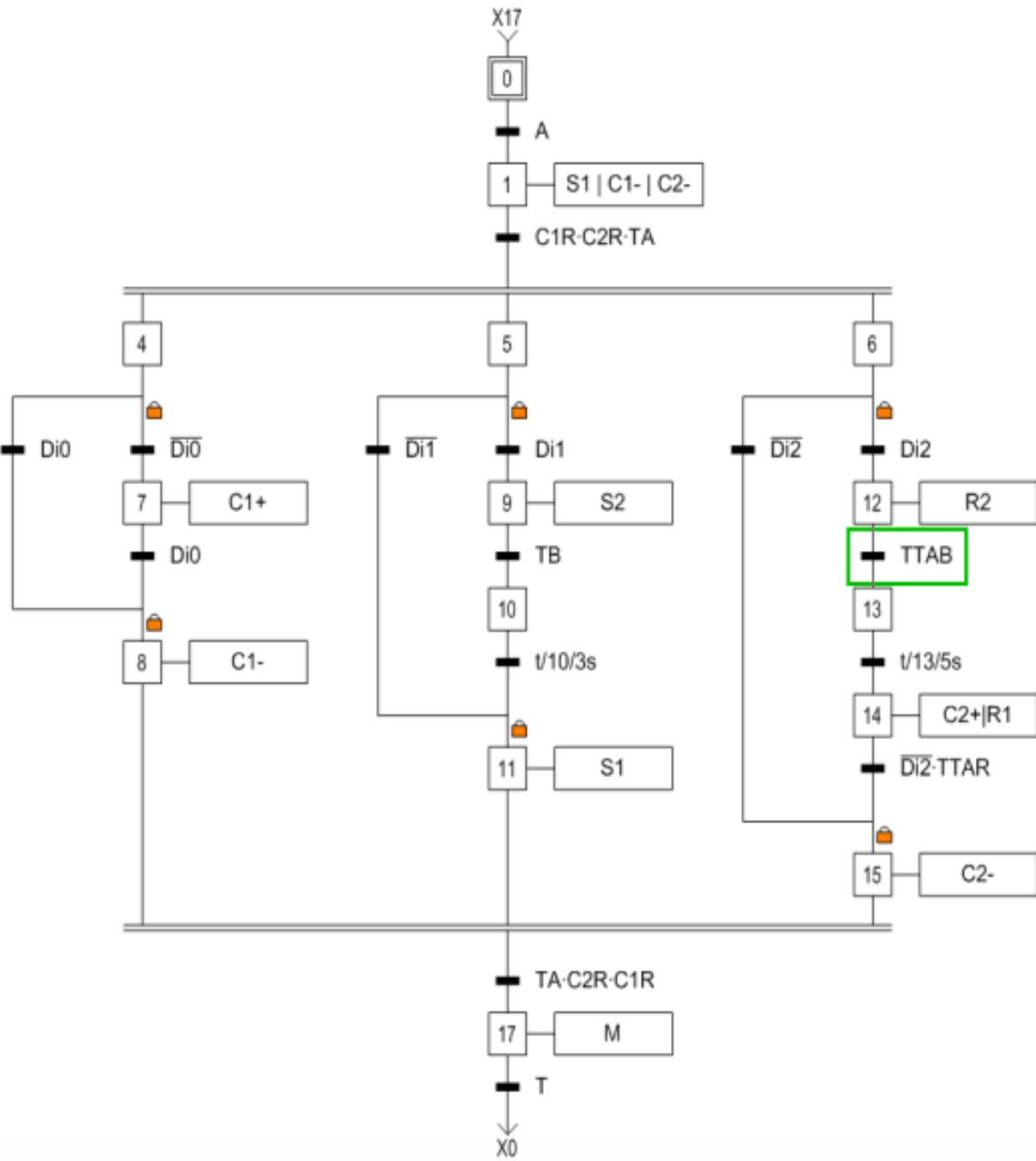
Funcionamiento (en detalle):

- Cuando la máquina se pone en marcha, se enciende el motor del taladro MT. Se asume que el motor necesita 3 segundos para llegar a una velocidad de giro adecuada. También, en el inicio de la operación, se debe subir el taladro (TA, S1). Además, se debe comprobar que el Testador se encuentra también arriba (TTA, R1).
- Se carga una pieza si no se detecta pieza en el receptor (Di0).
- Si existe pieza en el taladro (Di0), y este se encuentra arriba, entonces el taladro desciende, taladrando la pieza. Se esperan 3 segundos, y, a continuación el taladro asciende.

- Las piezas se van cargando en la estación de alimentación.
- A continuación, el plato de la máquina gira 120º. Cuando el movimiento se ha completado se detecta la señal T. Se debe esperar a que todos los procesos se terminen antes de mover el plato.
- Cuando existe una pieza en la estación de taladrado, se taladra.
- Cuando existe una pieza en la estación de test. Se baja, se mantiene durante 5 segundos y, a continuación, se levanta. Finalizada la operación, se acciona el cilindro C2 para expulsar la pieza.

Nivel 1

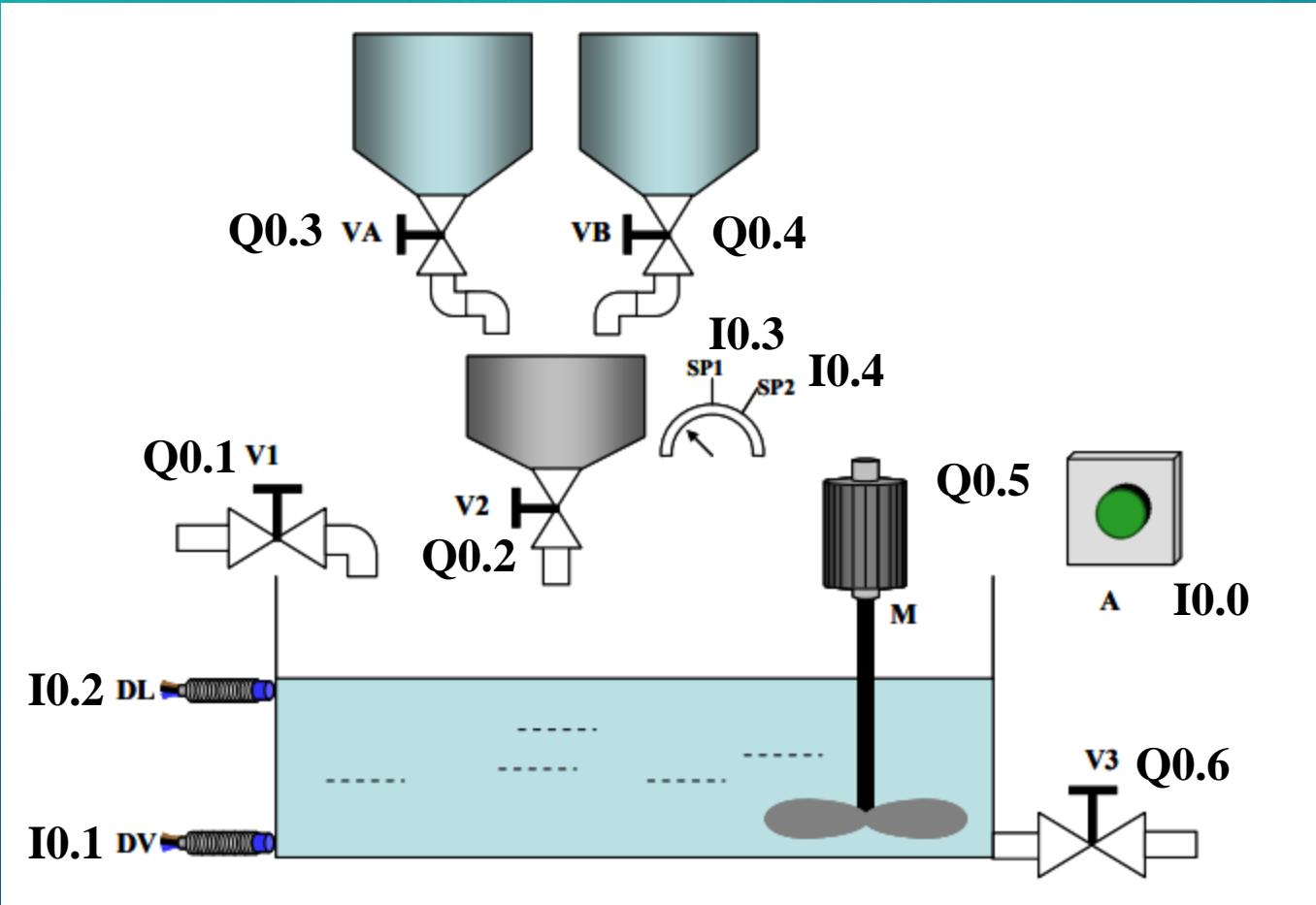




MEZCLADORA

- Se quieren mezclar 2 productos con agua. Se llena el depósito de agua abriendo la válvula V1. Al mismo tiempo la dosificación de los dos productos se realiza con una tolva acumulativa, se vierte el producto A sobre la tolva hasta que se alcanza un peso SP1 y a continuación se añade el producto B para conseguir el peso total de los 2 productos, SP2. Se abre la válvula de la tolva durante 10 segundos para dejar caer el contenido. Se realiza el proceso de mezclado durante 30 segundos accionando el agitador y se vacía el depósito para poder iniciar un nuevo ciclo. El proceso se activa con un interruptor A.

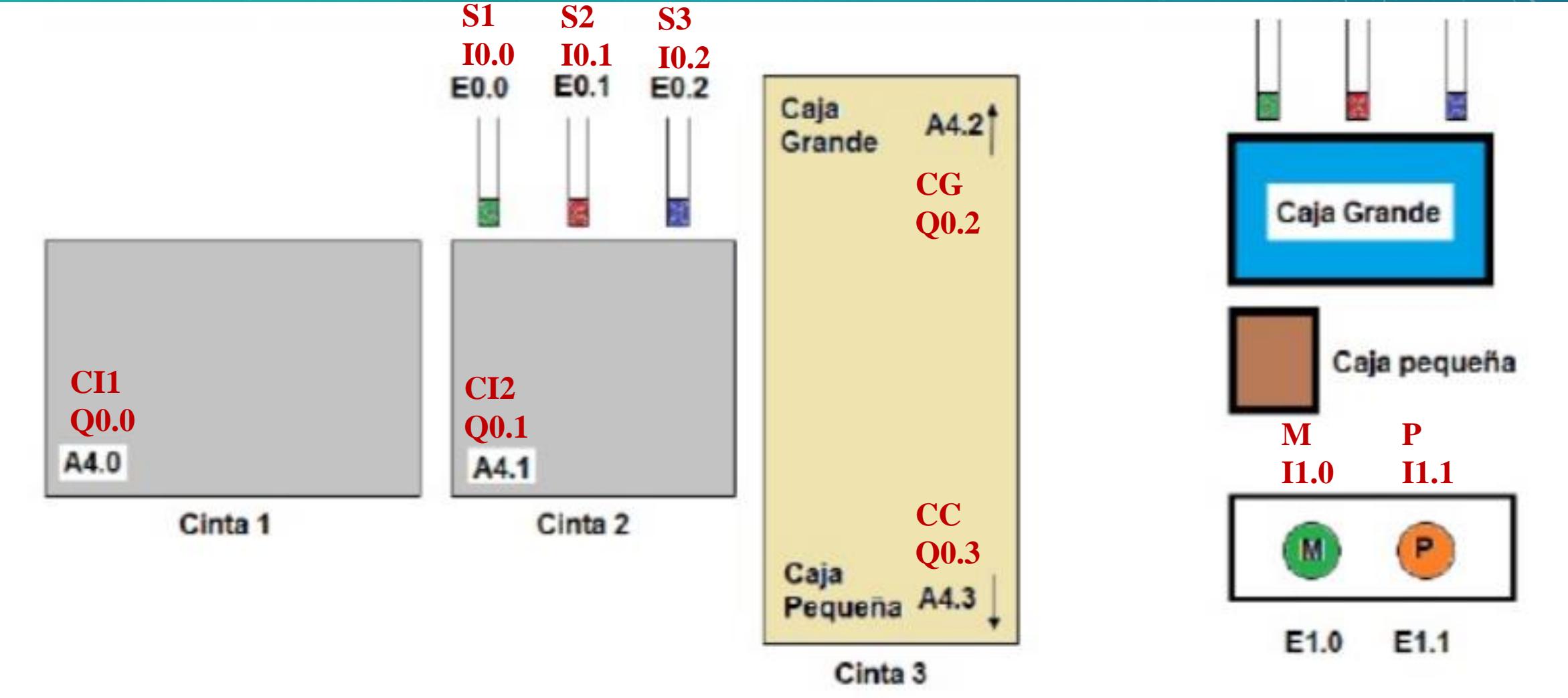
MEZCLADORA



EJEMPLO: CONTROL DE UN SISTEMA DE CINTAS TRANSPORTADORAS

- Tenemos tres cintas transportadoras dispuestas como indica la siguiente figura. Por las cintas transportadoras van a circular cajas grandes y pequeñas indistintamente. El tamaño de las cajas es detectado por tres sensores. Para cajas grandes los tres sensores se activan. Para las pequeñas sólo el primero de ellos.

Ejemplo: Control de un Sistema de Cintas transportadoras



EJEMPLO: CONTROL DE UN SISTEMA DE CINTAS TRANSPORTADORAS

- Cuando le demos al pulsador de marcha queremos que se ponga en marcha la cinta nº 1. Cuando llegue la primera caja a la cinta nº 2, queremos que se pare la cinta nº 1 y que se ponga en marcha la cinta nº 2.
- En la cinta nº 2 detectamos si la caja es grande o pequeña. Si es grande, queremos que se ponga en marcha la tercera cinta hacia arriba, y si es pequeña queremos que se ponga en marcha la tercera cinta hacia abajo.
- La cinta nº 2 se para cuando la caja ya esté abandonando la cinta nº 2. La cinta nº 3 se para a los 10 seg. de haberse puesto en marcha. A continuación se pone en marcha de nuevo la primera cinta y vuelve a comenzar el ciclo.



INTRODUCCIÓN PROGRAMMABLE LOGIC CONTROLLER - PLC

AUTOMATA PROGRAMABLE INDUSTRIAL - PLC

DEFINICIÓN

- Un autómata programable industrial (API) es un equipo electrónico, programable en lenguaje no informático, diseñado para controlar en tiempo real procesos secuenciales en ambiente de tipo industrial.
- El autómata programable también se conoce como **PLC**, que es la sigla de Programmable Logic Controller.

PROGRAMMABLE LOGIC CONTROLLER - PLC

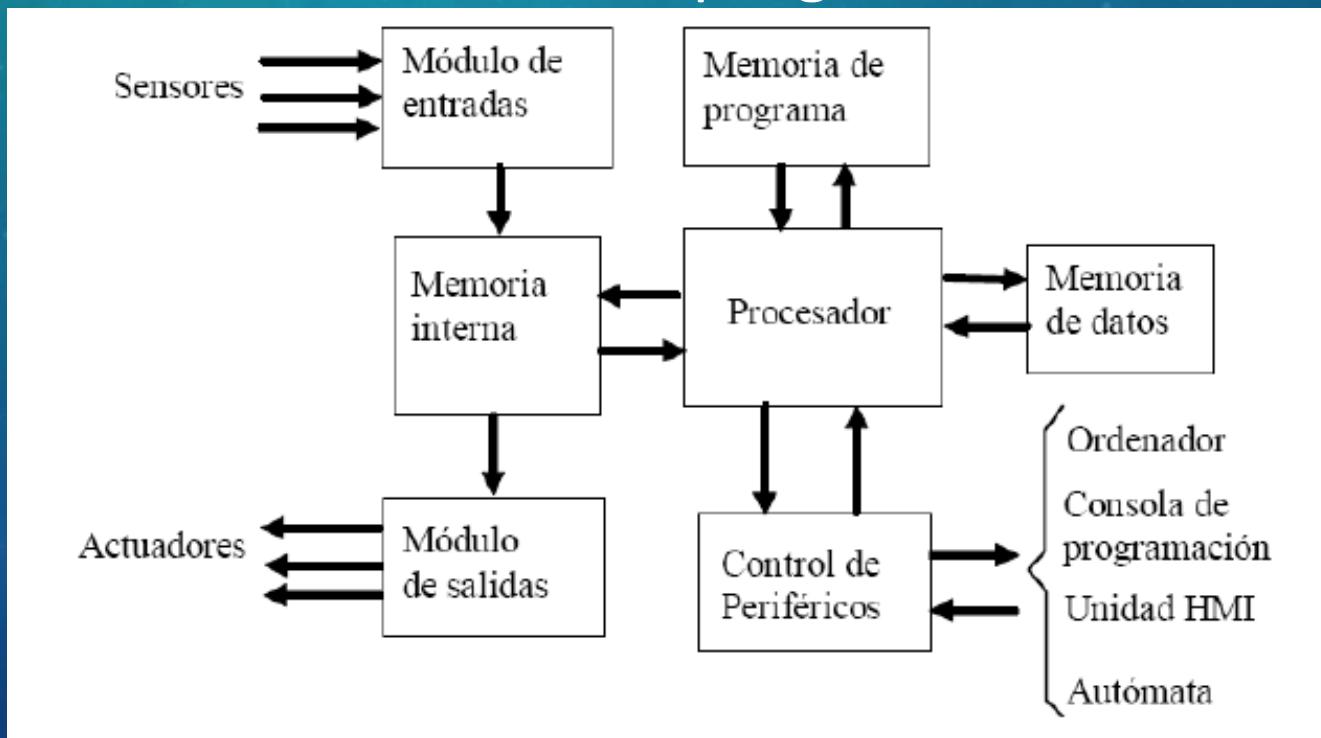
- Se programa en un lenguaje especial diseñado específicamente para generar de forma sencilla el programa que implementa el algoritmo de control de procesos secuenciales (de sistemas de eventos discretos), y porque el algoritmo de control programado es ejecutado de forma periódica en un ciclo temporal que es lo bastante breve como para poder controlar los procesos en tiempo real.

PROGRAMMABLE LOGIC CONTROLLER - PLC

- Utiliza componentes robustos que soportan condiciones de trabajo adversas, como las que se dan en ambientes industriales (polvo, temperatura, vibraciones, etc.), y porque su constitución física incluye los circuitos de interfaz necesarios para conectarlo de forma directa a los sensores y actuadores del proceso.

ARQUITECTURA DEL PLC

- La siguiente figura se muestra de forma esquemática la arquitectura interna de un PLC típico. Como computador que es, tiene un procesador que es el que ejecuta el programa almacenado en la memoria de programa.



MEMORIA DE PLC

- La memoria de programa y la de datos están físicamente separadas, constituyendo una arquitectura tipo Harvard. Además, la memoria de datos esta separada en dos tipos, que en la figura se denominan memoria de datos y memoria interna.

BLOQUES DE MEMORIA EN UN PLC

- Memoria interna.

Almacena los valores de entradas y salidas, además de otras variables internas del autómata. En ella se almacenan variables de 1 solo bit, es decir, variables que, aunque estén organizadas en bytes (grupos de 8 bits), se puede acceder a cada uno de los bits de forma independiente para leer o escribir.

En esta zona de memoria se leen los valores de las entradas (donde están conectados los sensores), y se escriben los valores de las salidas (donde están conectados los actuadores).

BLOQUES DE MEMORIA EN UN PLC

- Memoria de datos.

Contiene datos de configuración o parámetros de funcionamiento del autómata y del proceso, o datos de propósito general. En ella se almacenan variables tipo byte (8 bits) o word (16 bits).

Tanto la memoria interna como la de datos suelen tener una parte de RAM normal (volátil) y una parte de RAM con batería o EEPROM para almacenar datos que no se deben perder cuando se desconecta el autómata.

CONSTITUCIÓN FÍSICA

- La mayoría de los PLC's son modulares, es decir, están formados por módulos que pueden conectarse entre sí, permitiendo una gran flexibilidad en la configuración. En las siguientes figuras se muestran dos autómatas modulares comerciales.



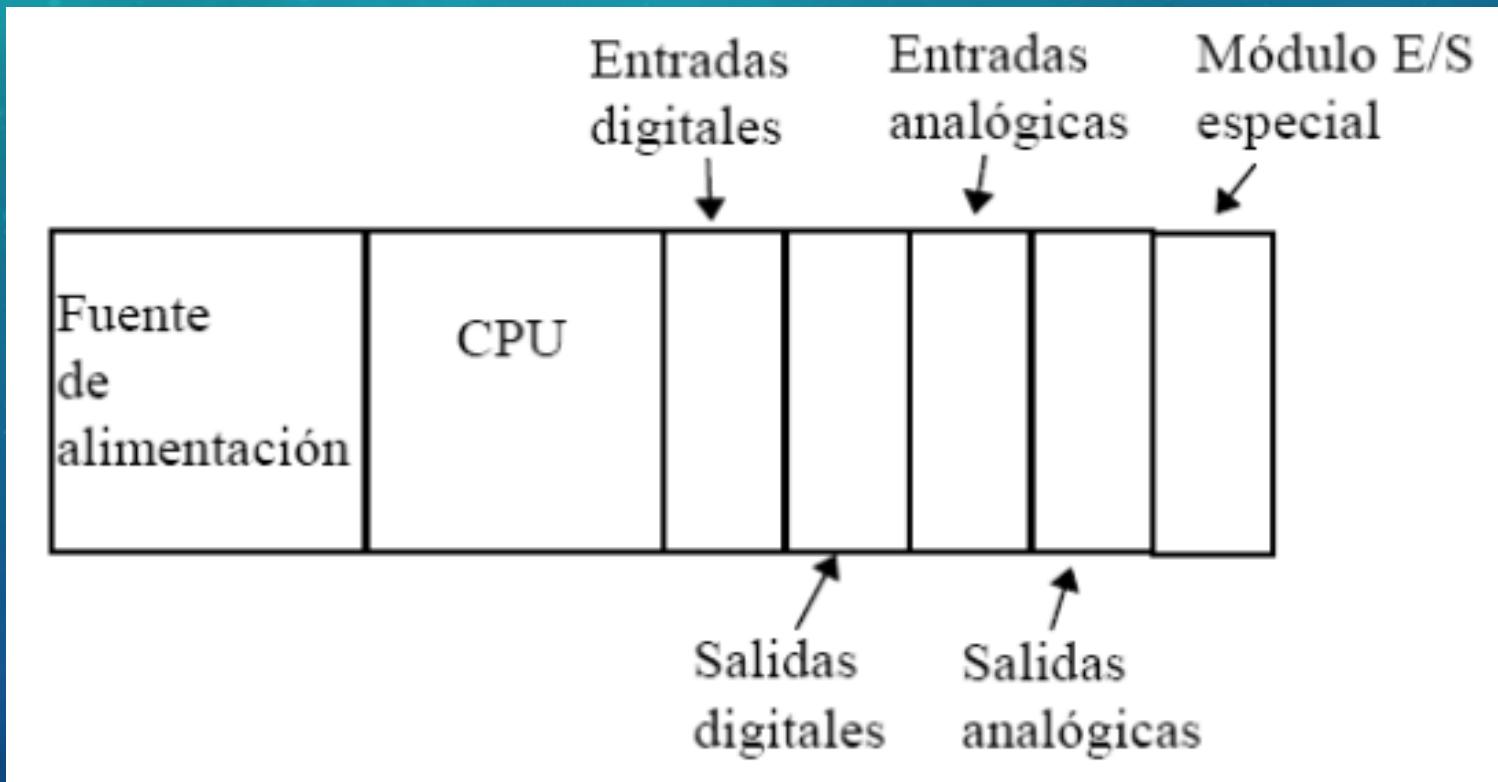
RockWell Allen-Bradley PLC SLC 500 PLC







ELEMENTOS CONSTITUTIVOS DE UN AUTÓMATA MODULAR.



CONSTITUCIÓN FÍSICA

- El modulo principal es el de CPU, que contiene el procesador, la memoria y algunos controladores de periféricos (puerto serie, por ejemplo). En algunos modelos, el modulo de CPU contiene además algunas entradas y/o salidas digitales.

CONSTITUCIÓN FÍSICA

- El modulo de fuente de alimentación da la tensión (normalmente 24 V) para alimentar a todos los módulos del equipo. Junto con el modulo de CPU forman la estructura mínima necesaria. A veces hay módulos que requieren una fuente de alimentación especial, además de la general del equipo.

- Los módulos adicionales se conectan al modulo de CPU por medio del bus. Respecto de los módulos entre los que se puede elegir para definir la configuración necesaria para controlar un proceso, se pueden distinguir:
 - Módulos de entradas digitales.
 - Módulos de salidas digitales.
 - Módulos de entradas analógicas.
 - Módulos de salidas analógicas.
 - Módulos de communication (RS232, RS485, Devicenet, Profibus, Ethernet, etc.).
 - Módulos de control PID.
 - Módulos de control de posición de ejes.

DESCRIPCIÓN FUNCIONAL

- El PLC es una computadora, y por lo tanto, su funcionamiento consiste en la ejecución de un determinado programa almacenado en la memoria de programa.
- Se puede distinguir dos modos de funcionamiento:
 - Modo de programación
 - Modo de ejecución.

DESCRIPCIÓN FUNCIONAL

- Modo de programación.

En este modo de funcionamiento el programa monitor que reside en la parte ROM de la memoria de programa comunica el PLC con el elemento de programación (normalmente un PC), para que desde éste se le traspase el programa que implementa el algoritmo de control que se desea ejecutar. En este modo, el PLC no esta controlando el proceso.

DESCRIPCIÓN FUNCIONAL

- Modo de ejecución (modo RUN).

En este modo, se ejecuta el programa del usuario que implementa al algoritmo de control, con lo que el autómata controla el proceso.

En este modo, cuando se inicializa el autómata, el programa monitor salta a la dirección donde está el programa de control, y se empieza a ejecutar éste. La ejecución del programa de control se lleva a cabo de forma cíclica, ejecutando ciclos de **scan** de forma indefinida

DESCRIPCIÓN FUNCIONAL

- Se llama ciclo de trabajo (o de scan) al conjunto de tareas que el autómata lleva a cabo (instrucciones que se ejecutan) de forma cíclica cuando está controlando un proceso.
- La inicialización se lleva a cabo una sola vez siempre que se pone en marcha el autómata.

MÓDULOS DE ENTRADA/SALIDA Y MÓDULOS DE COMUNICACIÓN

- Módulos de entradas digitales

Son módulos que permiten leer señales binarias (digitales) obtenidas a partir de detectores (cuya salida es activa o inactiva). Normalmente, van aisladas por medio de optoacopladores.

Suelen funcionar con niveles de tensión de 24 V (de continua o de alterna) o niveles de 220 V (de alterna).

- Módulos de salidas digitales

Estos módulos permiten transmitir una señal binaria (activada/desactivada) a un dispositivo externo. Hay tres tipos de salidas digitales que son las más utilizadas: salida a relé (contacto), salida a transistor (que puede ser NPN o PNP) y salida a TRIAC.

En todos los casos, el circuito interno está aislado de los componentes externos (en la salida a relé por el propio relé, en la salida a transistor por medio de un optoacoplador y en la salida a TRIAC por un optotriac).

- Módulos de entrada/salida analógica

Son módulos que permiten medir señales analógicas y dar como salida señales analógicas. De esta forma, permiten controlar procesos que requieran señales continuas.

Las entradas analógicas pueden ser de tensión (normalmente en un rango de -10 a 10 V ó de 0 a 10 V) o de corriente (en un rango de 4 a 20 mA, p. ej.).

UNIDADES DE PROGRAMACIÓN

- Las unidades de programación son los periféricos más comunes que se utilizan al trabajar con los PLC. Estas unidades son las que permiten introducir en la memoria del autómata el programa de usuario que servirá para controlar el proceso. Las más utilizadas son:
 - Consola de programación
 - PC

UNIDADES DE PROGRAMACIÓN

- Consola de programación

La consola de programación es un dispositivo específico para programar el autómata. Consta de un pequeño teclado y una pequeña pantalla.

Se comunica con el autómata por medio del puerto serie o de un puerto especial de periféricos (situados en el modulo de CPU). Introducir programas mediante la consola resulta engorroso y lento. Normalmente tienen utilidad para realizar pequeñas modificaciones en los programas de autómatas que ya están instalados en planta, debido a que son de pequeño tamaño (cómodos de transportar) y se alimentan directamente del autómata.

UNIDADES DE PROGRAMACIÓN

- Ordenador personal

El elemento más utilizado para programar los PLC es el ordenador personal. Se conecta al autómata mediante el puerto serie RS232, el puerto USB o ethernet. Cada fabricante de autómatas dispone de un software que permite la edición de programas, la depuración, la compilación y la carga del mismo en el autómata. También permite ejecutar el programa en el autómata y tener acceso al mismo tiempo a las distintas variables, lo que permite la depuración y detección de errores.

UNIDADES HMI

- Además de las unidades de programación, podemos destacar otros periféricos muy utilizados que son las unidades HMI (Human Machine Interface).
- Son unidades que disponen de un teclado y una pantalla (que puede ser bastante grande), o bien únicamente de una pantalla táctil. Se utilizan para mostrar al operador el estado del proceso y permitir la introducción de consignas (órdenes de marcha, de paro, cambio de parámetros de funcionamiento, etc.).

PROGRAMACIÓN DE LOS PLC

- La forma más común de programar un autómata es mediante un ordenador personal con el software adecuado. En primer lugar se debe configurar el equipo (seleccionar el tipo de autómata y los módulos de entradas/salidas que se tiene).
- Después se introduce el programa de control (en el lenguaje adecuado), y por ultimo, se carga este programa en el autómata para proceder a su verificación (se ejecuta el programa comprobando la evolución de las distintas variables).

PROGRAMACIÓN DE LOS PLC

- Antes de introducir el programa es conveniente hacer una lista de las variables de entrada y salida del proceso, y su asignación a variables internas en la memoria del PLC. Algunos software de programación permiten simular la ejecución del programa sin necesidad de cargarlo en el PLC, por lo que se puede hacer la depuración de forma más sencilla.

PROGRAMACIÓN DE LOS PLC

- Lenguajes de programación

El lenguaje de programación más utilizado en los autómatas programables es el diagrama de contactos (Ladder Diagram o diagrama de escalera).

Está basado en los automatismos cableados por medio de contactores, que fueron los primeros en implementarse. Gráficamente se representan dos líneas verticales largas separadas, de forma que la de la izquierda representa tensión y la de la derecha, masa. Entre esas líneas verticales, se representan las ecuaciones lógicas por medio de contactos.

PROGRAMACIÓN DE LOS PLC

Hay 3 tipos de elementos fundamentales.

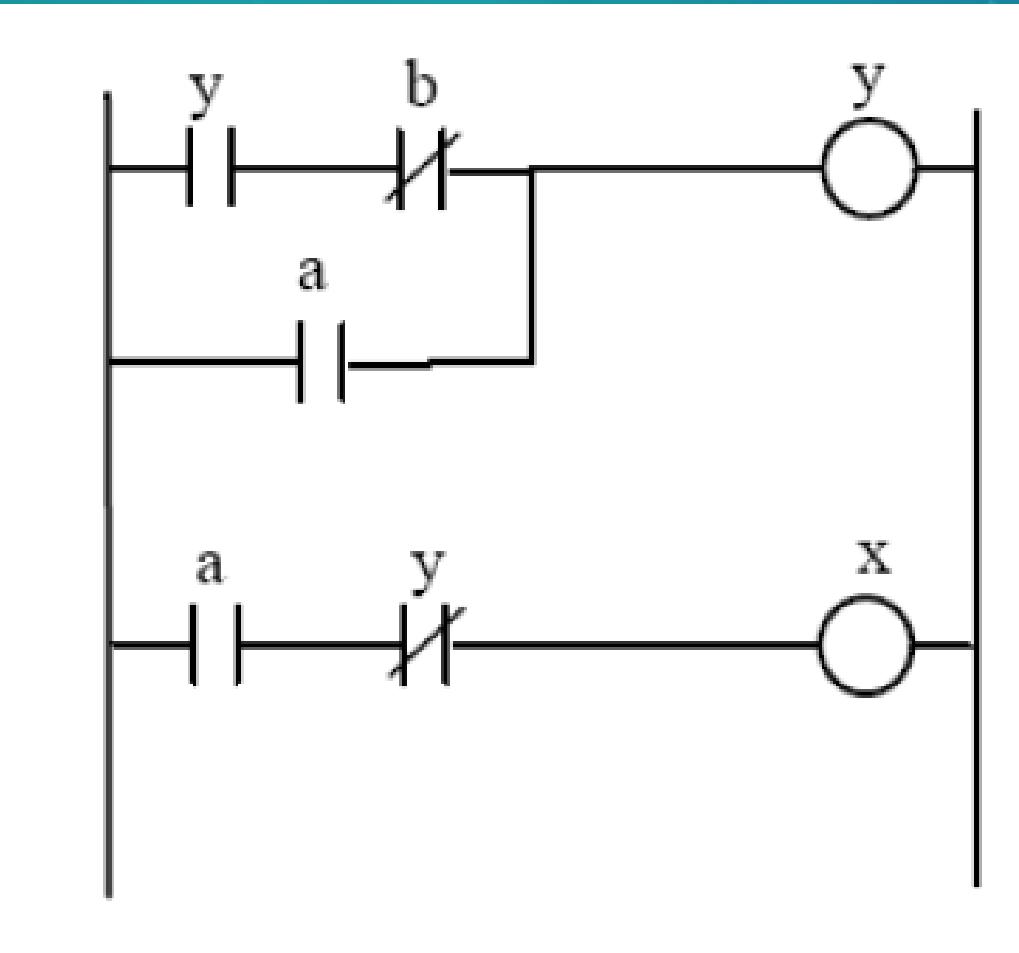
- Contacto normalmente abierto (NA): Representa un contacto que esta abierto si la variable asociada vale 0, y que se cierra si la variable asociada vale 1. Se representa con dos líneas verticales paralelas.
- Contacto normalmente cerrado (NC): Representa un contacto que esta cerrado si la variable asociada vale 0, y que se abre si la variable asociada vale 1. Se representa con dos líneas verticales paralelas cruzadas por una línea oblicua.

PROGRAMACIÓN DE LOS PLC

- Bobina. Representa el valor de una variable. Cada salida tiene asociada una bobina. Si esta bobina tiene corriente (el diagrama de contactos la activa), la salida está a 1, y si no tiene corriente, la salida está a cero.

Además puede haber bobinas asociadas a variables internas (por ejemplo, las que representan cada etapa del proceso). Se representa mediante un circulo.

EJEMPLO DE PROGRAMA EN DIAGRAMA DE CONTACTOS (LADDER DIAGRAM).



PROGRAMACIÓN DE LOS PLC

- Cada línea o red (que normalmente contiene una sola bobina, aunque puede contener varias) representa una línea del programa. El autómata ejecuta las líneas de arriba abajo.
- La función lógica implementada por los contactos de una red se suele denominar condición de activación, ya que cuando esta condición es 1, la variable de salida (bobina) asociada se activa.

PROGRAMACIÓN DE LOS PLC

Hay una serie de reglas que se debe seguir para dibujar correctamente un diagrama de contactos. Algunas de ellas son:

- A la derecha de la bobina no puede haber contactos.
- En una red puede haber varias bobinas en paralelo.
- La conducción a través de los contactos solo se produce de izquierda a derecha.

PROGRAMACIÓN DE LOS PLC

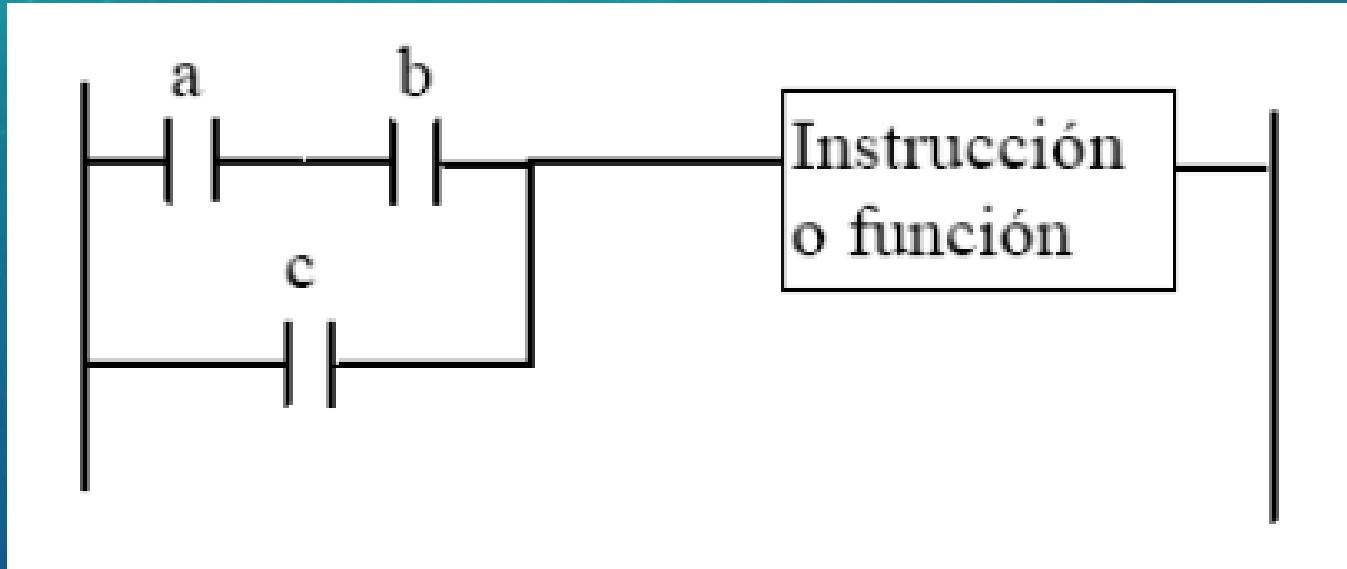
Además de la bobina simple puede haber otros dos tipos de bobinas:

- Bobina normalmente cerrada. Se representa mediante un círculo con una línea transversal. Su valor es el inverso del que resulta de la ecuación lógica de los contactos.

PROGRAMACIÓN DE LOS PLC

- Bobina de enclavamiento. En algunos autómatas se representa mediante un circulo que en el interior tiene una letra L, de Latch (o S de Set) o una letra U, de Unlatch (o R, de Reset). Si es una letra L (Latch) o S (Set) indica que la variable correspondiente se activa, quedando activa aunque después se haga cero la condición de activación. Para desactivar la variable tiene que utilizarse una bobina de des enclavamiento con la letra U (Unlatch) o R (Reset).

USO DE FUNCIONES COMPLEJAS EN DIAGRAMAS DE CONTACTOS.



- El significado es que cuando el autómata llega a la línea correspondiente, ejecuta la instrucción o función compleja en función de que los contactos den un valor 1 ó 0, es decir, en función de que la condición de activación sea 1 ó 0.

Entre estas funciones se puede destacar:

- Instrucción END. Indica el final del programa.
- Temporizadores.
- Contadores.
- Instrucciones de salto.
- Operaciones binarias (mover bytes, sumar, multiplicar, comparar, etc.).
- Actualización de E/S. Permite refrescar las E/S cuando se desee, independientemente del refresco automático que se produce cada ciclo de scan.
- Funciones matemáticas complejas (senos, cosenos, exponenciales).
- Control de interrupciones.
- Funciones especiales de comunicaciones (transmisión o recepción de datos).

PROCEDIMIENTO A SEGUIR PARA PROGRAMAR UN PLC

Una vez se tiene el programa, se debe proceder según los siguientes pasos para introducir el programa en el PLC:

1. Configurar el equipo (tipo de PLC, módulos de entrada y salida, comunicaciones, etc).
2. Identificar el programa (nombre, empresa, fecha, versión, etc.).
3. Crear la estructura básica del programa (bloques y grupos dentro de los bloques).

PROCEDIMIENTO A SEGUIR PARA PROGRAMAR UN AUTÓMATA

4. Hacer la lista de etiquetas (nombres) de las variables a utilizar (entradas, salidas, variables internas, datos, temporizadores, contadores), asignándoles su dirección en memoria.
5. Introducir el programa (diagrama de contactos).
6. Comprobar el programa. Si el software lo permite, se puede simular la ejecución del programa en el PC y verificar el funcionamiento correcto.
7. Transferir el programa al autómata.
8. Ejecutar el programa en el PLC y comprobar el funcionamiento.

CRITERIOS DE SELECCIÓN DE UN PLC

Las características fundamentales que definen un PLC, y que se deben tener en cuenta a la hora de elegir el más adecuado para una aplicación son:

- Posibilidad de entradas/salidas especiales (analógicas, PID, contador de alta velocidad, salidas de pulsos PWM).
- Lenguaje de programación (diagrama de contactos, lista de instrucciones, Grafcet,) y dispositivos de programación (ordenador, consola).
- Comunicaciones. Posibilidad de conexión en redes de comunicación (E/S remotas, bus de campo, red de área local, red de nivel superior).
- Capacidad para implementar funciones matemáticas complejas.

CRITERIOS DE SELECCIÓN DE UN PLC

Los criterios fundamentales a tener en cuenta para la selección del autómata más adecuado según las características anteriores son:

- Número y tipo de E/S a controlar. El PLC debe admitir el número de entradas y salidas que se necesitan y algunas más para posibles imprevistos o ampliaciones o modificaciones futuras.
- Complejidad (tamaño) del programa de control que se quiere implementar.

CRITERIOS DE SELECCIÓN DE UN PLC

- El programa de control debe caber en la memoria, y debe sobrar algo de espacio para imprevistos o posibles ampliaciones y modificaciones futuras.
- Potencia de las instrucciones necesarias (funciones matemáticas, rapidez de respuesta). Hay que tener en cuenta si se necesita hacer cálculos complejos (senos, raíces cuadradas, etc.) o si el algoritmo de control necesita ejecutarse especialmente rápido por alguna característica del proceso.

CRITERIOS DE SELECCIÓN DE UN PLC

- Necesidad de utilización de periféricos (como terminales táctiles, lectores de código de barras, etc.), módulos especiales de entrada o salida y módulos de comunicación. El PLC elegido debe admitir los módulos especiales y de comunicación que se necesiten.

CRITERIOS DE SELECCIÓN DE UN PLC

- Precio. Evidentemente, a mayor precio, mayores prestaciones (en tamaño de memoria, en capacidad de cálculo, en disponibilidad de módulos, etc.). Se debe llegar a un compromiso entre las prestaciones y el precio final. Este compromiso dependerá del número de unidades que se vayan a fabricar.

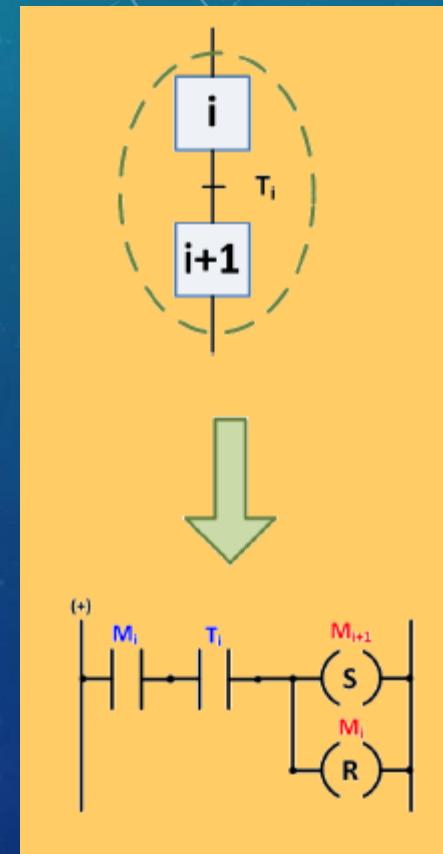
CRITERIOS DE SELECCIÓN DE UN PLC

- Posibilidad de ampliaciones futuras. Como ya se ha comentado, incluso en el caso de que no estén previstas ampliaciones futuras determinadas, es conveniente dejar algo de margen en el tamaño de memoria y número de entradas/salidas para imprevistos. No obstante, en muchas ocasiones se pueden prever ampliaciones futuras del proceso en las que se puede evaluar las necesidades en cuanto al autómata de control.

GRAFCET – LADDER

EQUIVALENCIA GRAFCET – LADDER (LENGUAJE DE ESCALERA)

- Un diagrama GRAFCET se puede “traducir” a lenguaje de contactos o lenguaje Ladder. Con esto se aprovecha la comprensión visual del ciclo de trabajo que ofrece el GRAFCET con la potencia de programación del Ladder.



EQUIVALENCIA GRAFCET - LADDER

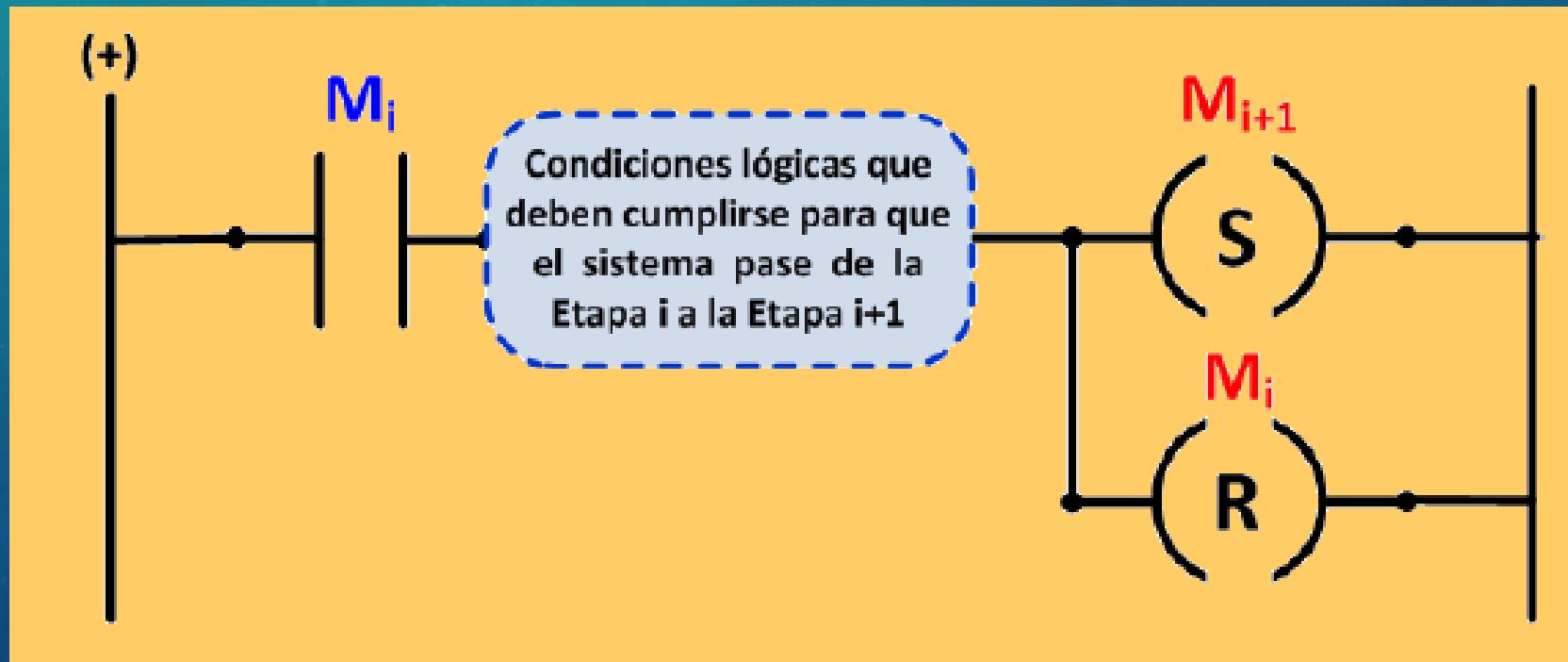
- La traducción de GRAFCET a Ladder se consideran dos aspectos:
 - Control de la Secuencia de las etapas
 - Acciones que deben realizarse en cada etapa cuando se hallen activas

CONTROL DE LA SECUENCIA

- Hay que asignar en cada etapa una marca interna (bit interno) que estará en “1” el tiempo que se encuentre activa la etapa asociada.
- Para esto se usa bobinas SET y RESET. Esta parte de Ladder controla la evolución de la secuencia del proceso, etapa por etapa, transición por transición.

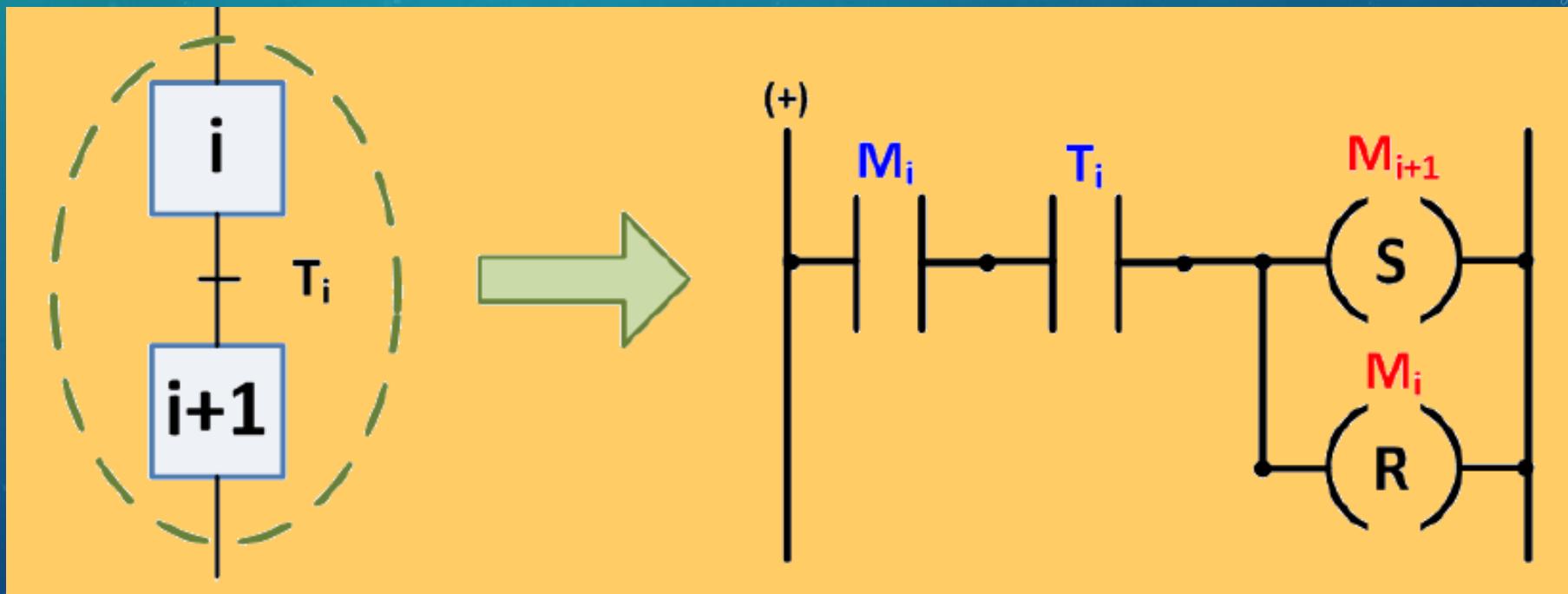
CONTROL DE LA SECUENCIA

- *Estando activa la etapa i, y si se cumple la transición, se desactiva la etapa i (RESET) y se activa la nueva (SET).*



CONTROL DE LA SECUENCIA

- *Estando activa la etapa i ($M_i = 1$) y si se cumple la transición ($T_i = 1$), se desactiva la etapa i (RESET $\rightarrow M_i = 0$) y se activa la nueva (SET $\rightarrow M_{i+1} = 1$)*



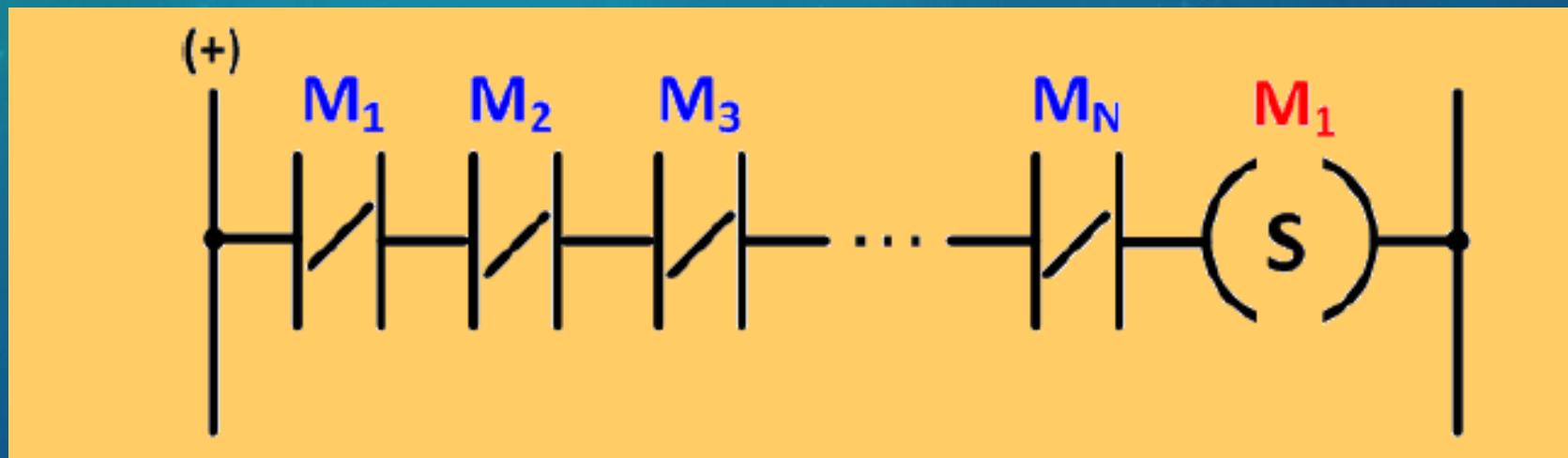
CONTROL DE LA SECUENCIA

Secuencia Inicial

- Ninguna etapa está activa, todos los bits internos asociados a etapas valen cero y por lo tanto ninguna acción se ejecuta. Por esa razón debe existir una primera línea de programación que indique cual es la primera etapa a ejecutar.

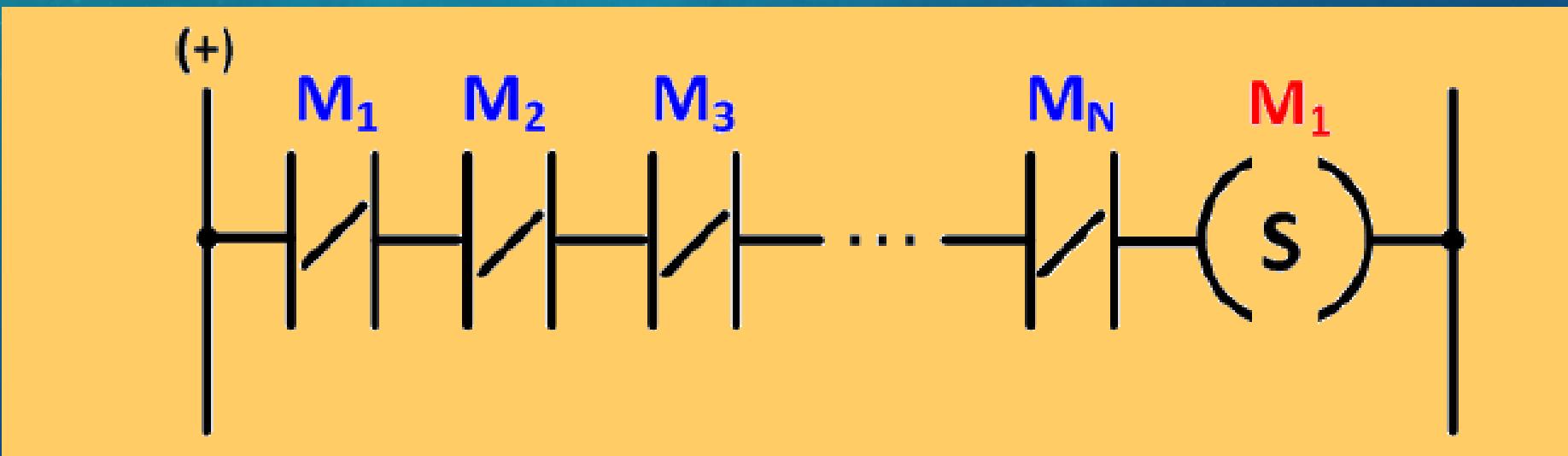
Control de la secuencia

- En el inicio, ninguna etapa está activada y por esta única vez, se activa la primera de las etapas ($SET \rightarrow M_1 = 1$)



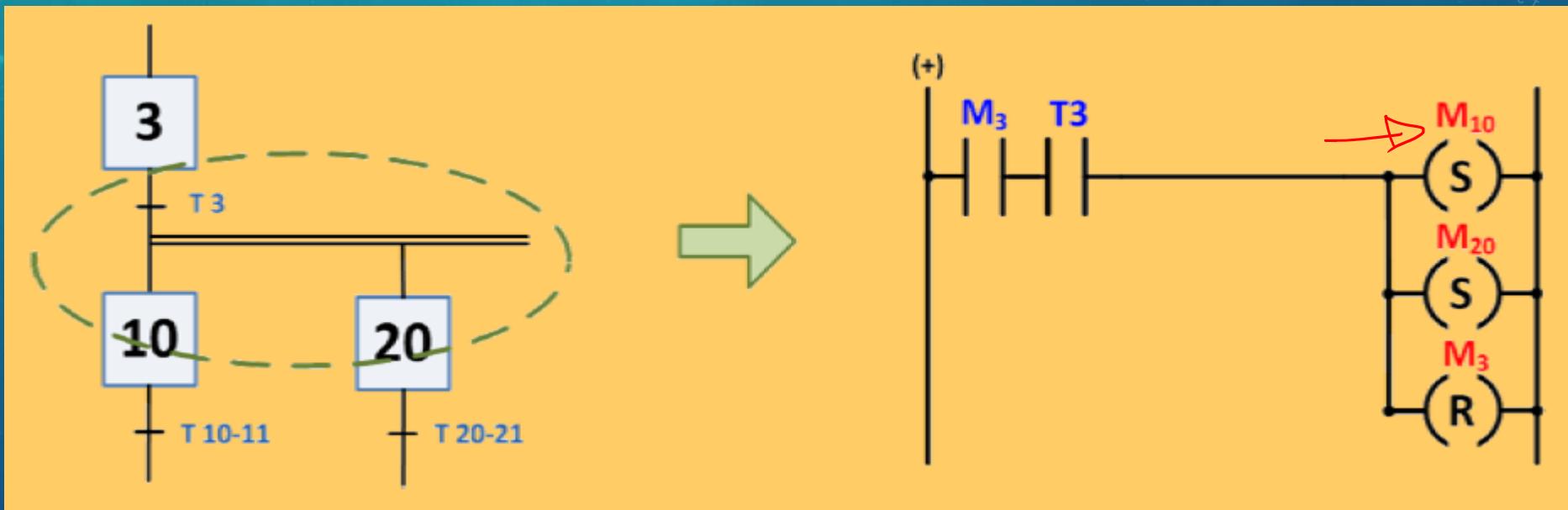
Control de la secuencia

- En el inicio, ninguna etapa está activada y por esta única vez, se activa la primera de las etapas ($SET \rightarrow M_1 = 1$).



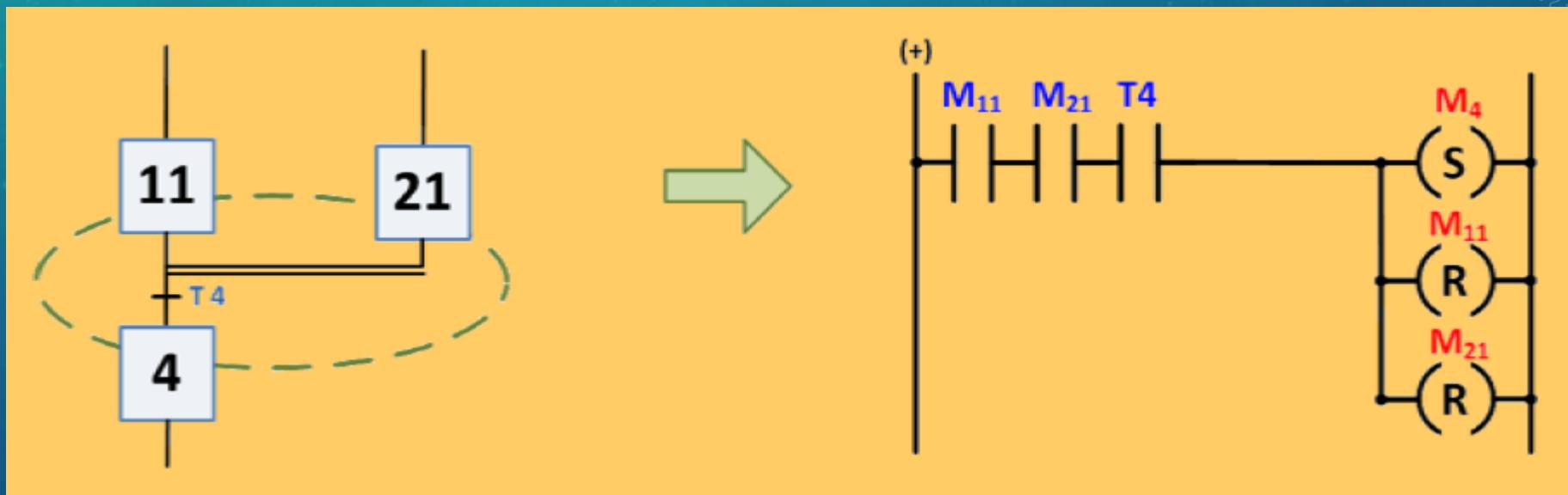
Control de la secuencia

- *Secuencias simultáneas Divergencia AND*



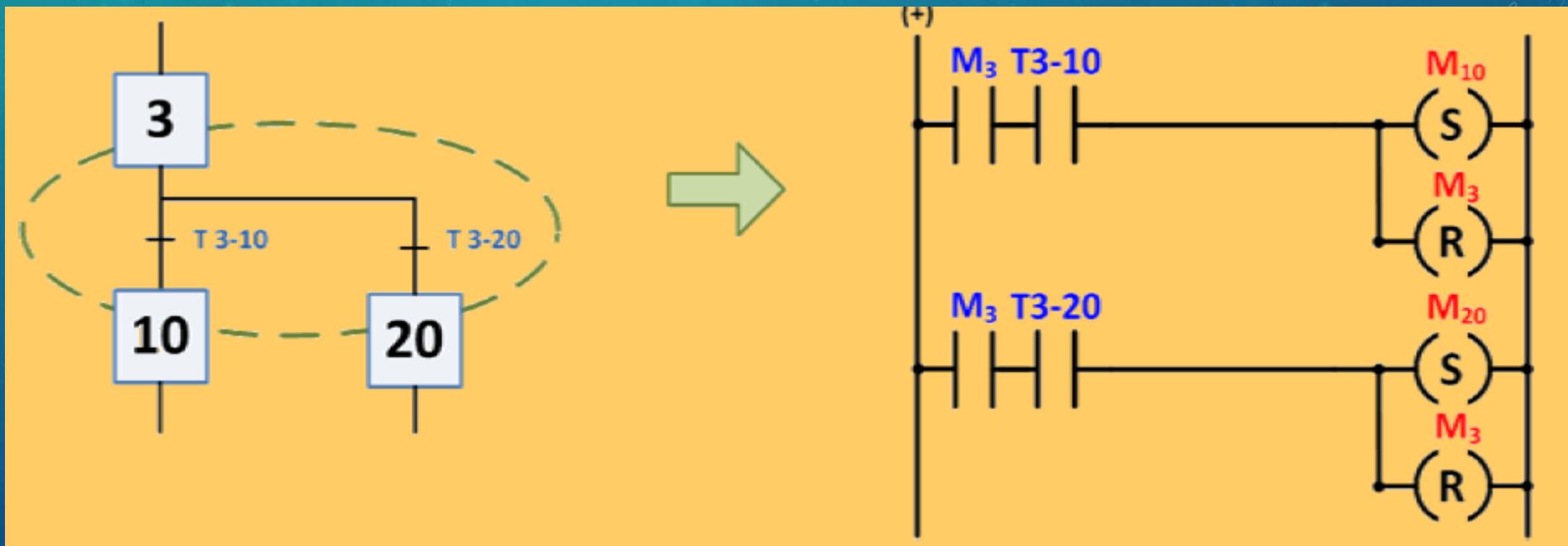
Control de la secuencia

- Secuencias simultáneas Convergencia AND



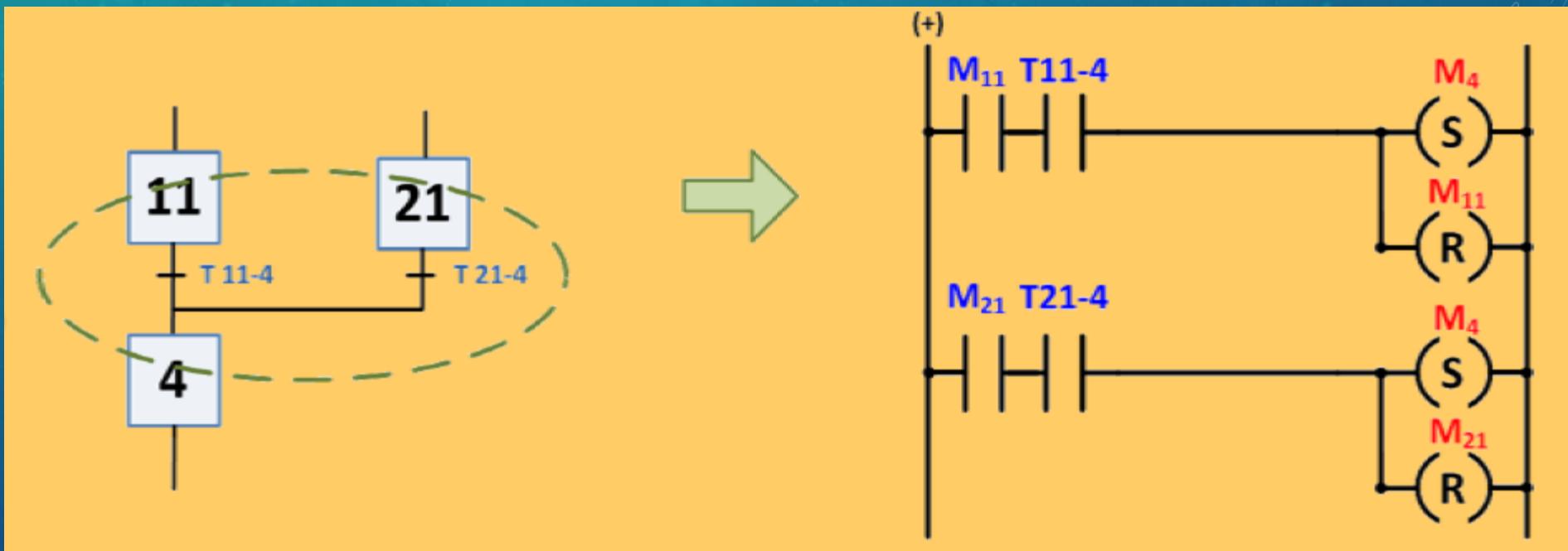
Control de la secuencia

- Secuencias alternativas Divergencia OR



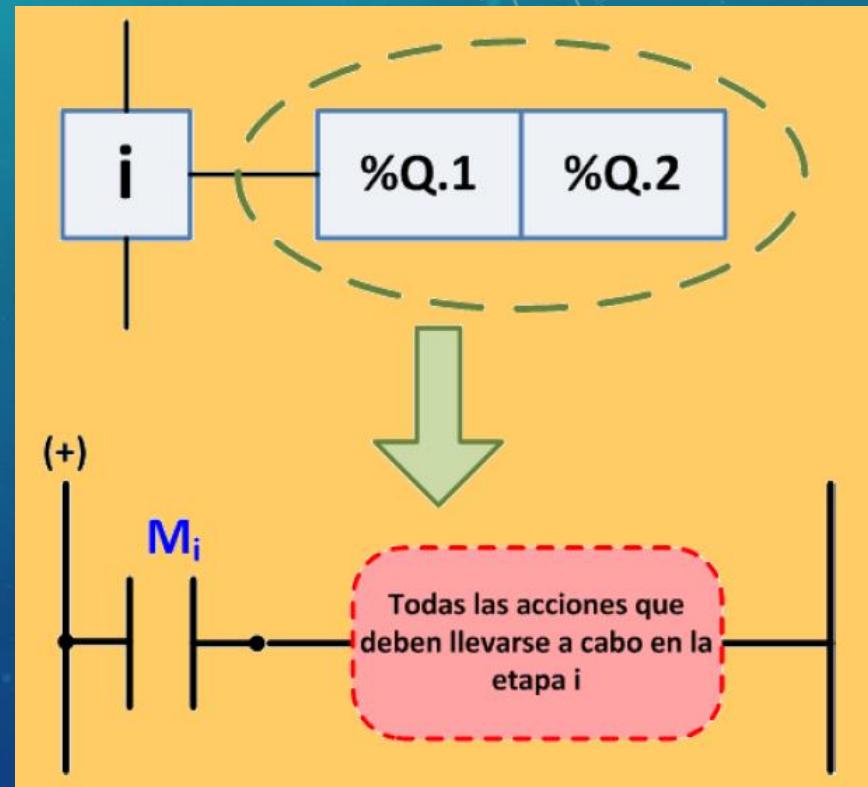
Control de la secuencia

- Secuencias alternativas Convergencia OR

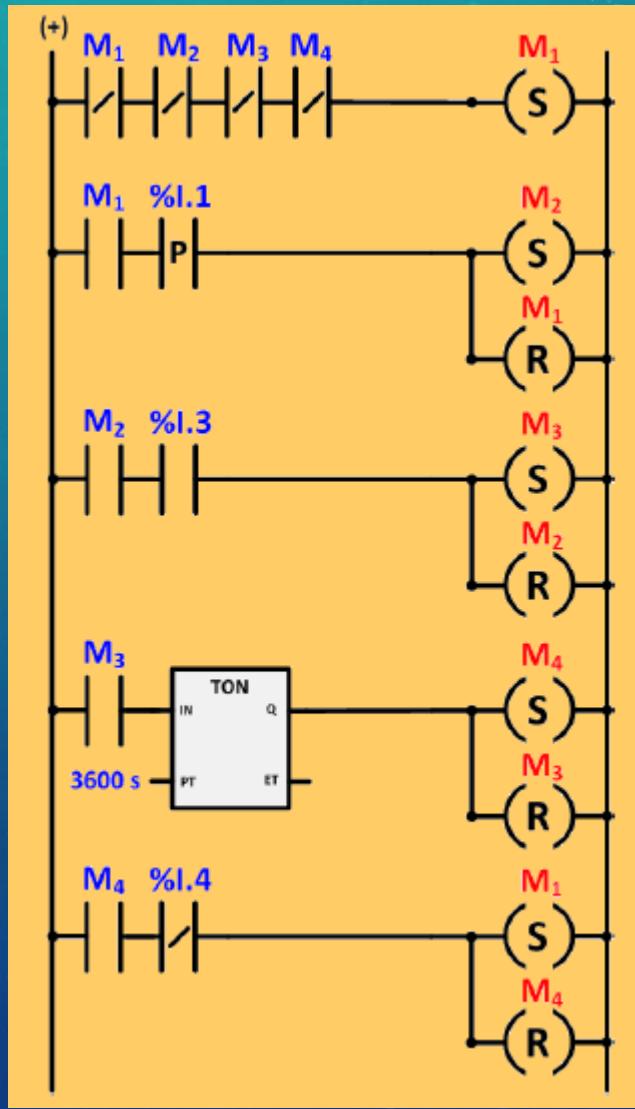
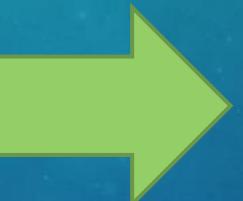
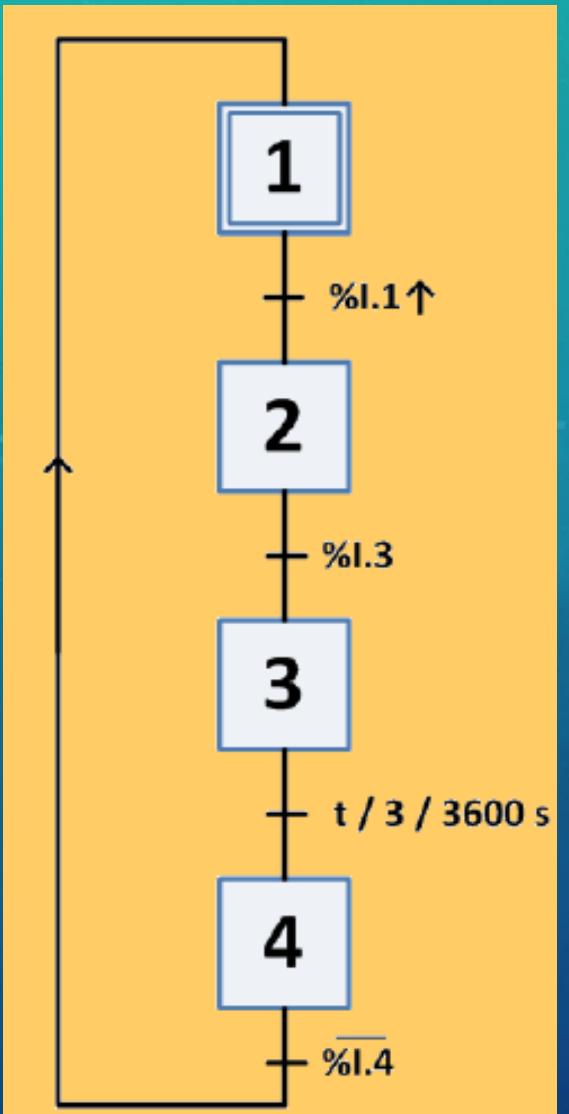


Acciones en cada etapa

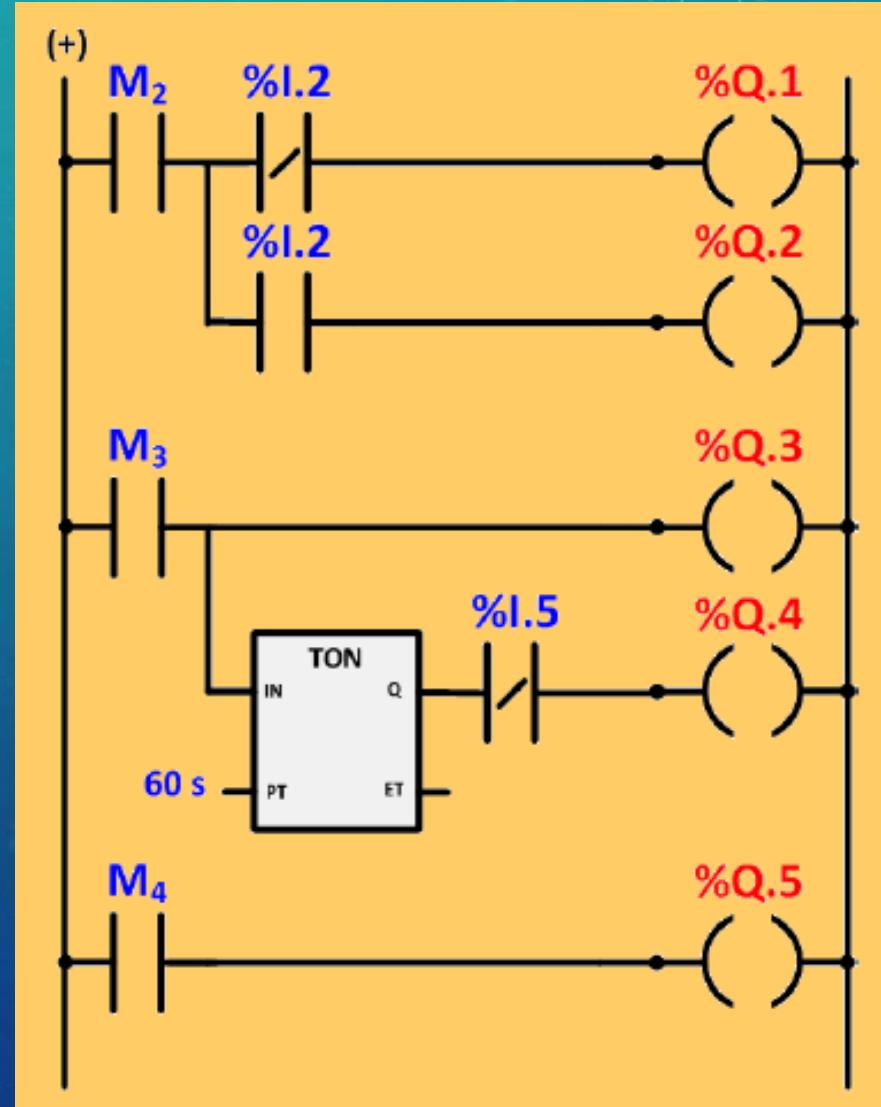
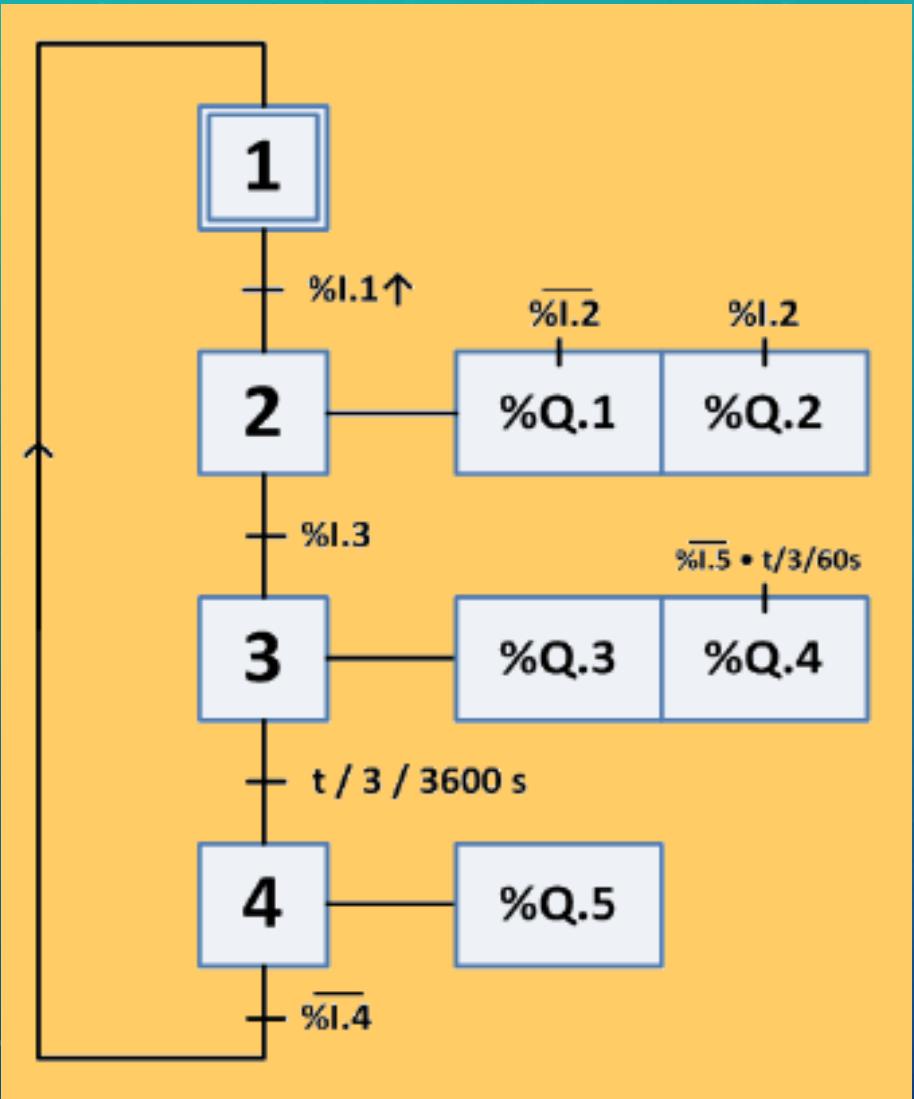
- El programa en lenguaje Ladder deberá contemplar las acciones asociadas a cada etapa, considerando las marcas internas activas que corresponden a cada una.

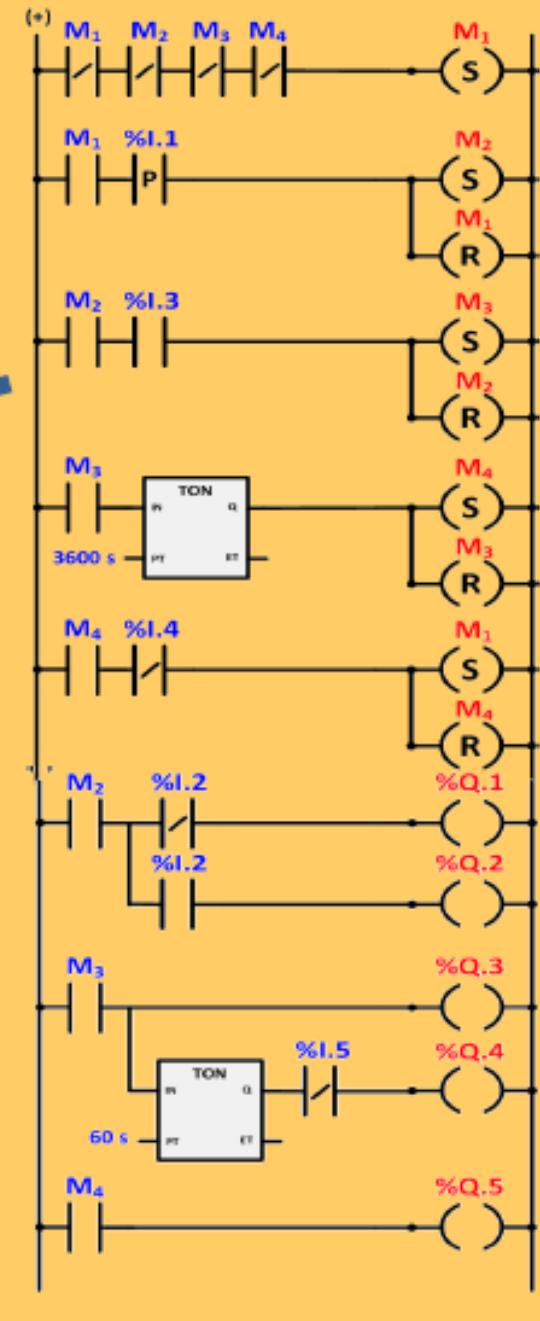
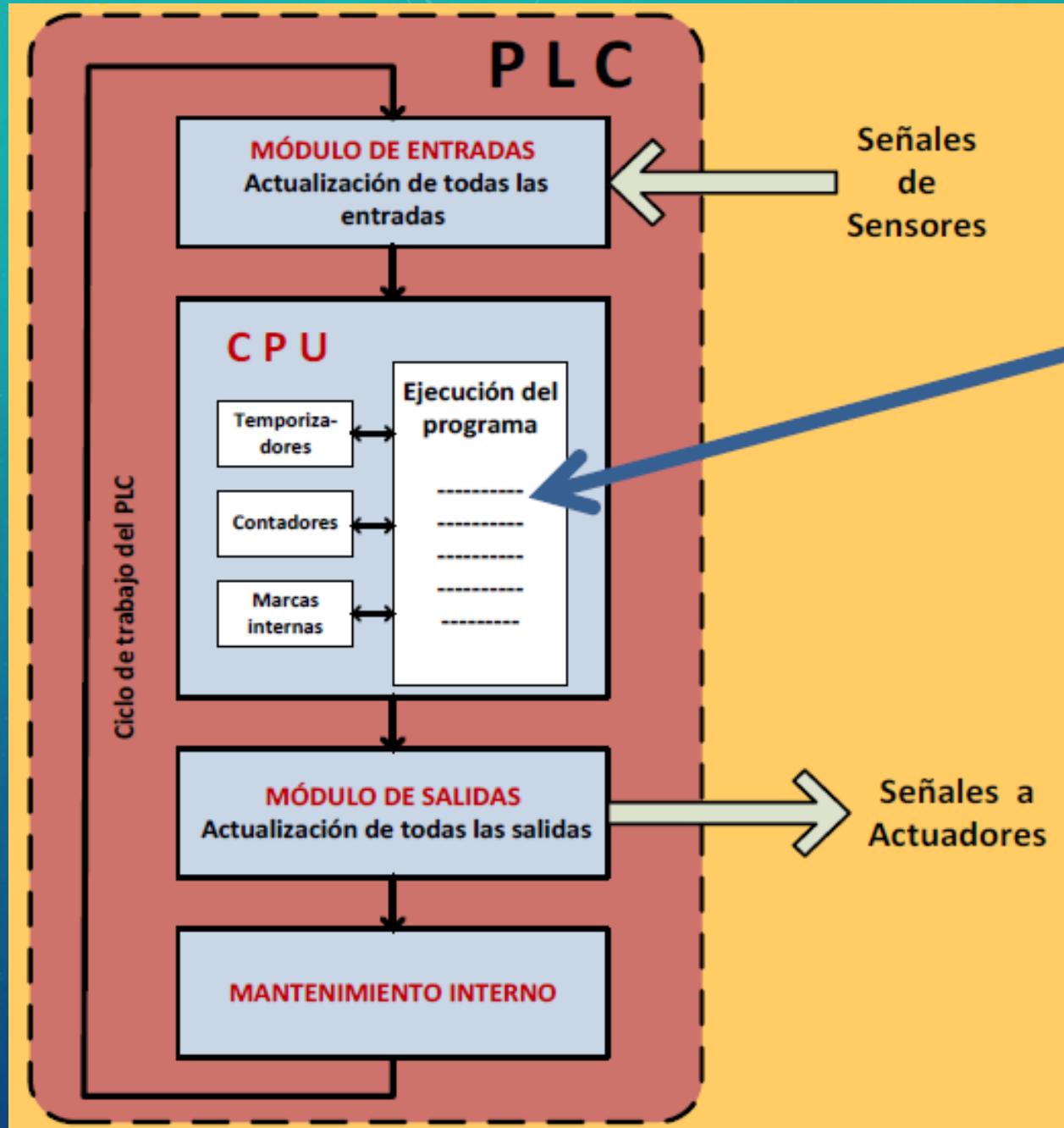


ETAPAS GRAFCET - LEADDER



Acciones Grafcet - Leadder

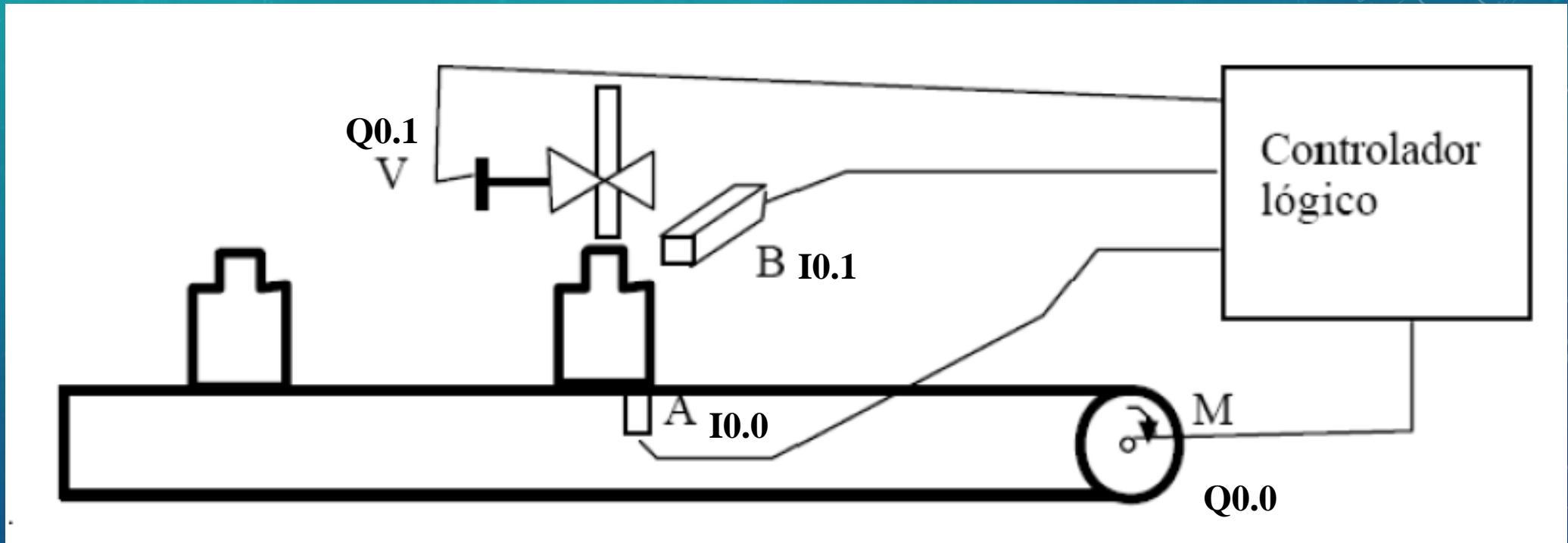


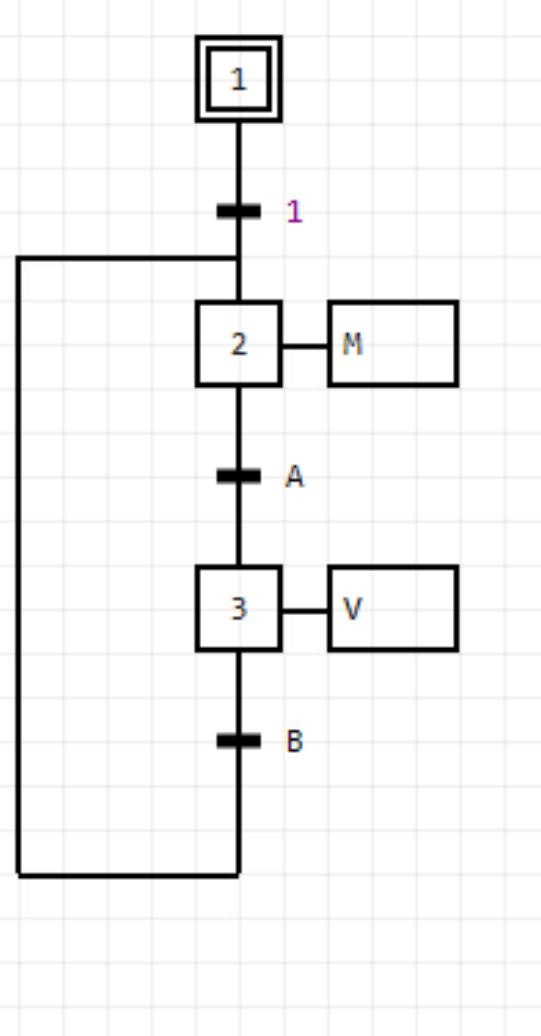


EJEMPLO: PROCESO DE LLENADO DE BOTELLAS

- Como ejemplo considérese la embotelladora que se muestra en la figura. Existen 2 señales dadas por 2 sensores. La señal A es 1 si tiene una botella encima y 0 en caso contrario. La señal B es 1 si el líquido que llena la botella ha llegado al nivel máximo y 0 en caso contrario. Asimismo existen otras 2 señales. Cuando M se pone a 1, el motor se pone en marcha, y cuando se pone a 0, se para. Cuando V se pone a 1, la electroválvula se abre empezando a llenar la botella, cerrándose cuando V = 0.

EJEMPLO: PROCESO DE LLENADO DE BOTELLAS

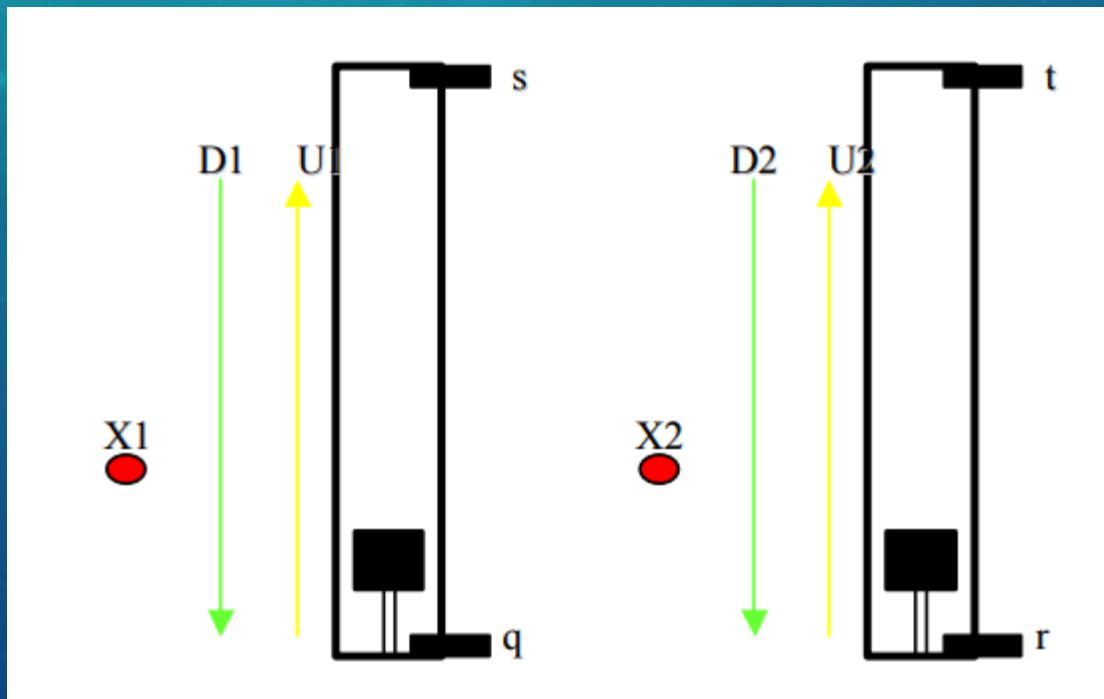


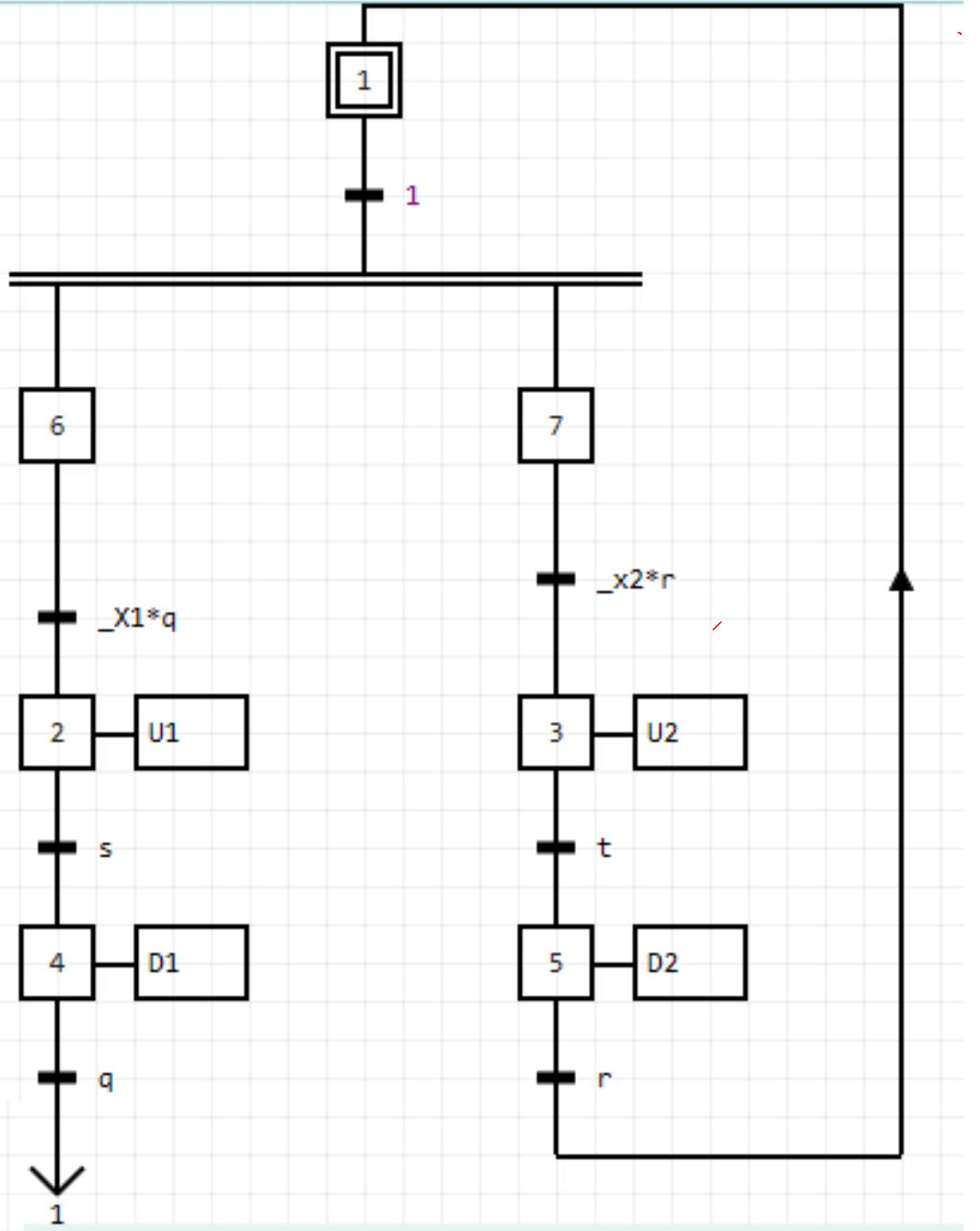


EJERCICIO

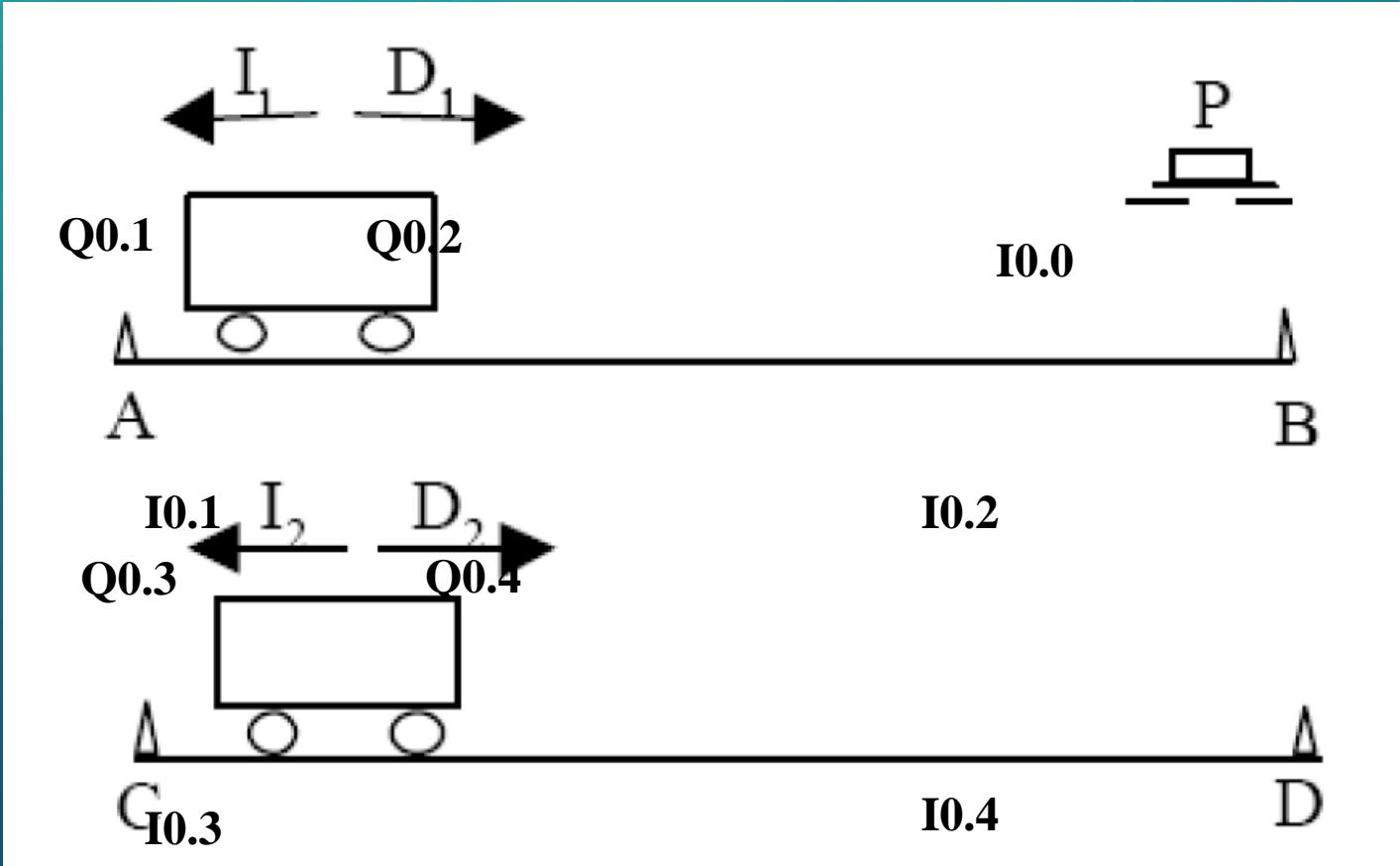
- En la siguiente figura los carros están inicialmente en A y C. Cuando se pulsa P, se ponen en marcha hacia la derecha hasta llegar a B y D. Cada carro, al llegar al extremo derecho, cambia de sentido de forma independiente poniéndose en marcha hacia la izquierda hasta llegar a C o A.

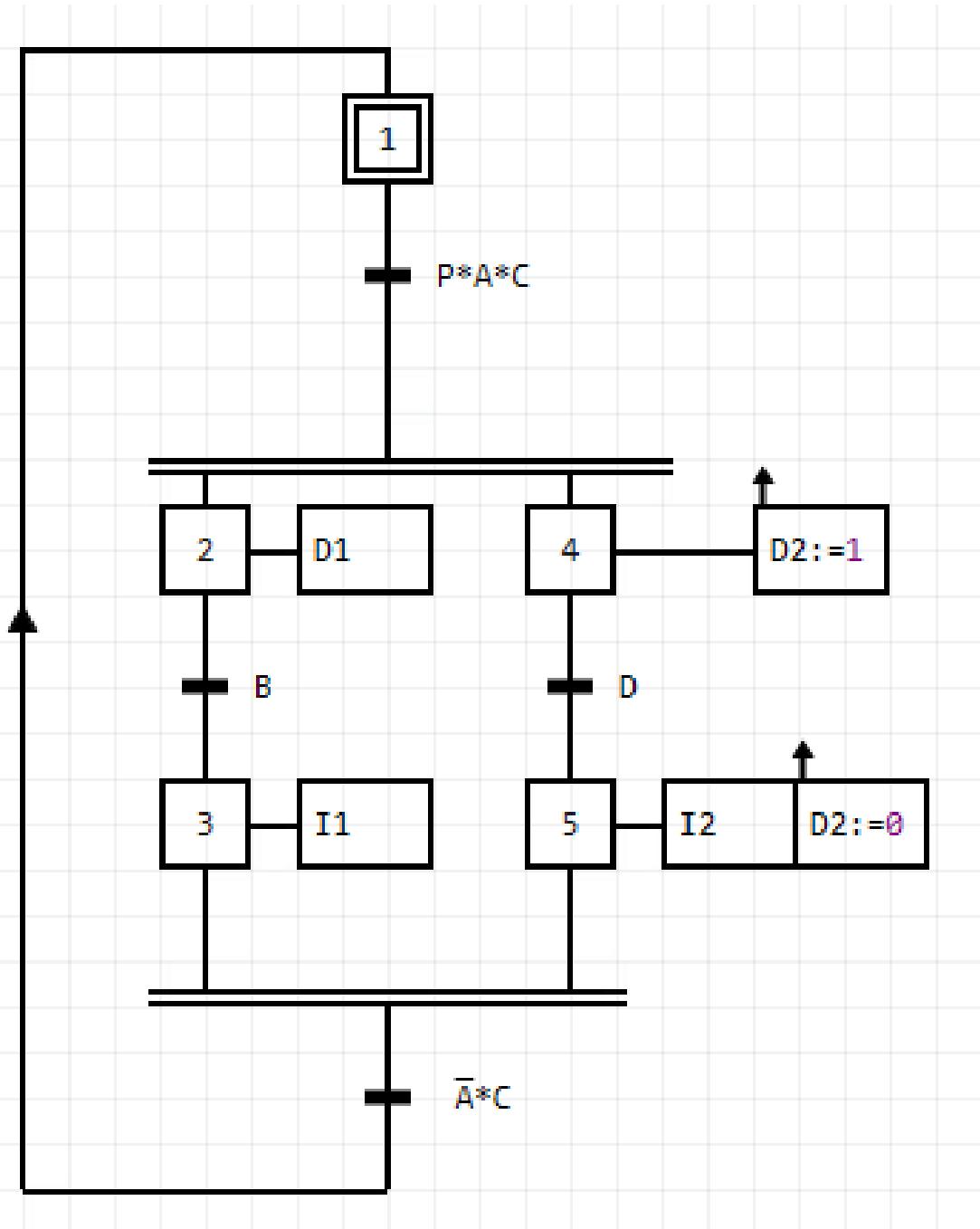
- En el siguiente ejemplo dos montacargas se mueva cuando se pulsa un botón (x_1 o x_2) en sentido ascendente hasta el final de recorrido (s,t) y de inmediato se desciende a la situación inicial (q,r).





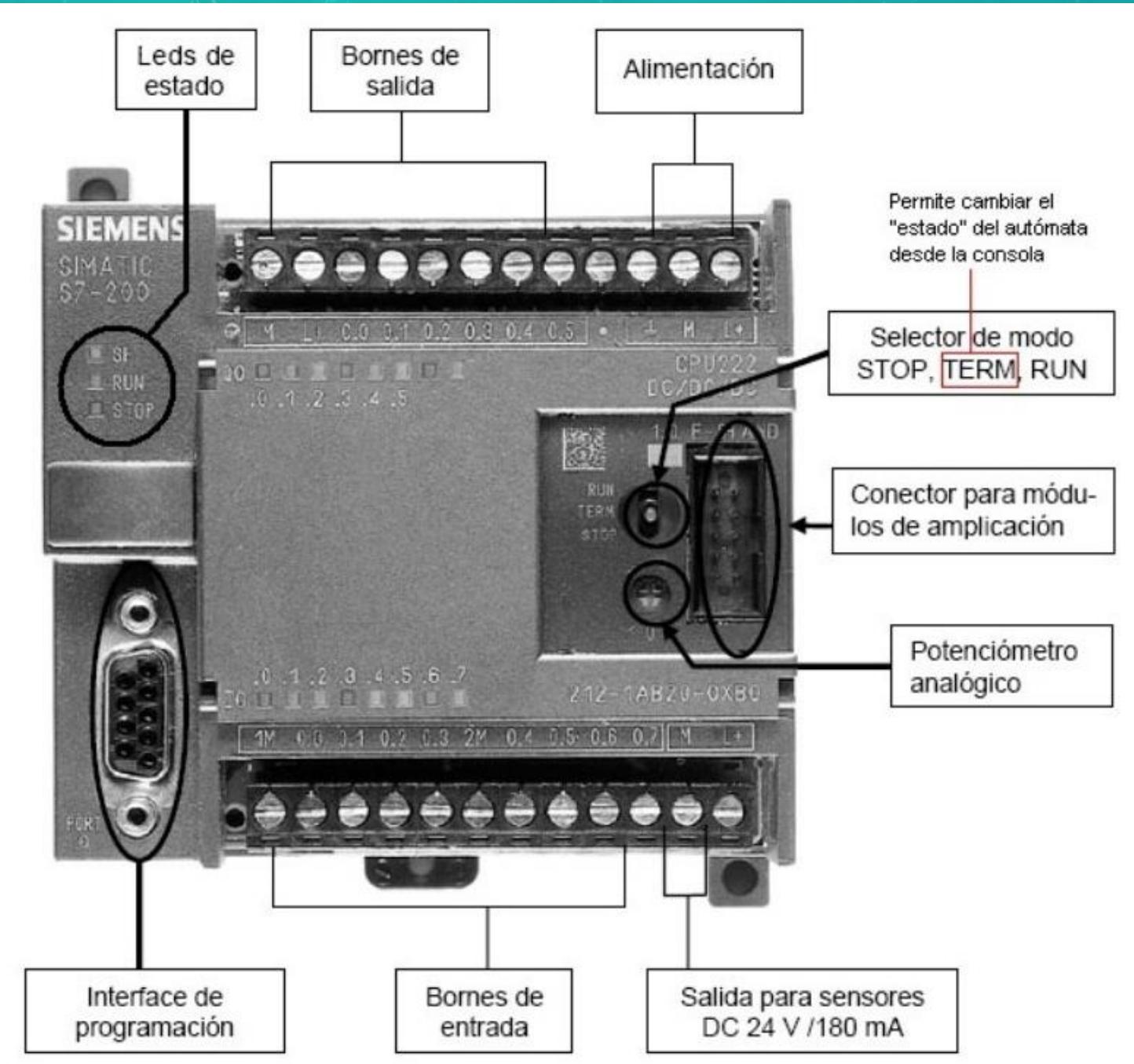
EJEMPLO:





PLC S7-200





CPU S7--200

- La CPU S7--200 incorpora en una carcasa compacta un microprocesador, una fuente de alimentación integrada, así como circuitos de entrada y de salida que conforman un potente Micro--PLC. Tras haber cargado el programa en el S7--200, éste contendrá la lógica necesaria para supervisar y controlar los aparatos de entrada y salida de la aplicación.

MICRO--PLC S7--200

LEDs de estado:

Fallo del sistema/diagnóstico
(SF/DIAG)

RUN

STOP

Cartucho opcional:

Cartucho de memoria

Reloj de tiempo real

Pila

Puerto de comunicación

Micro-PLC S7-200

LEDs de E/S

Tapa de acceso:

Selector de modo (RUN/STOP)

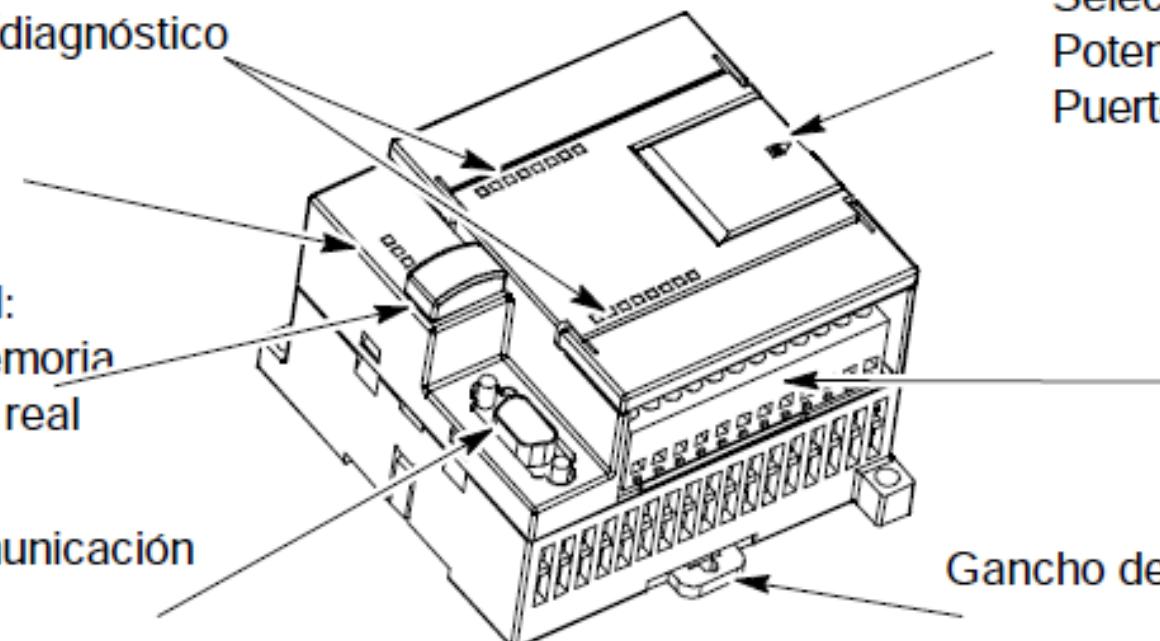
Potenciómetro(s) analógico(s)

Puerto de ampliación (en casi todas las CPUs)

Bloque de terminales

(extraíble en la CPU 224, CPU 224XP
y CPU 226)

Gancho de retención para el montaje en un rafil DIN



Comparativa de las CPUs S7-200

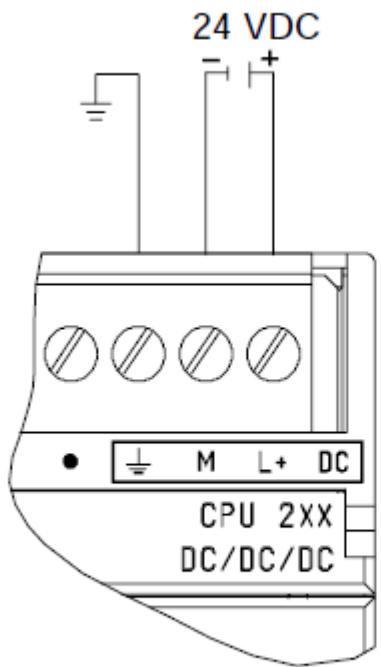
| Función | CPU 221 | CPU 222 | CPU 224 | CPU 224XP, CPU 224XPsi | CPU 226 |
|---|------------------------------|--------------------------|---------------------------|----------------------------|----------------------------|
| Dimensiones físicas (mm) | 90 x 80 x 62 | 90 x 80 x 62 | 120,5 x 80 x 62 | 140 x 80 x 62 | 190 x 80 x 62 |
| Memoria del programa: con edición en runtime sin edición en runtime | 4096 bytes 4096 bytes | 4096 bytes 4096 bytes | 8192 bytes 12288 bytes | 12288 bytes 16384 bytes | 16384 bytes 24576 bytes |
| Memoria de datos | 2048 bytes | 2048 bytes | 8192 bytes | 10240 bytes | 10240 bytes |
| Memoria de backup | 50 horas (típ.) | 50 horas (típ.) | 100 horas (típ.) | 100 horas (típ.) | 100 horas (típ.) |
| E/S integradas | | | | | |
| Digitales | 6 E/4 S | 8 E/6 S | 14 E/10 S | 14 E/10 S | 24 E/16 S |
| Analógicas | | | | 2 E/1 S | |
| Módulos de ampliación | 0 módulos | 2 módulos ¹ | 7 módulos ¹ | 7 módulos ¹ | 7 módulos ¹ |
| Contadores rápidos | | | | | |
| Fase simple | 4 a 30 kHz | 4 a 30 kHz | 6 a 30 kHz | 4 a 30 kHz 2 a 200 kHz | 6 a 30 kHz |
| Dos fases | 2 a 20 kHz | 2 a 20 kHz | 4 a 20 kHz | 3 a 20 kHz 1 a 100 kHz | 4 a 20 kHz |
| Salidas de impulsos (DC) | 2 a 20 kHz | 2 a 20 kHz | 2 a 20 kHz | 2 a 100 kHz | 2 a 20 kHz |
| Potenciómetros analógicos | 1 | 1 | 2 | 2 | 2 |
| Reloj de tiempo real | Cartucho | Cartucho | Incorporado | Incorporado | Incorporado |
| Puertos de comunicación | 1 RS-485 | 1 RS-485 | 1 RS-485 | 2 RS-485 | 2 RS-485 |
| Aritmética en coma flotante | Sí | | | | |
| Tamaño de la imagen de E/S digitales | 256 (128 E / 128 S) | | | | |
| Velocidad de ejecución booleana | 0.22 microsegundos/operación | | | | |

Módulos de ampliación S7-200

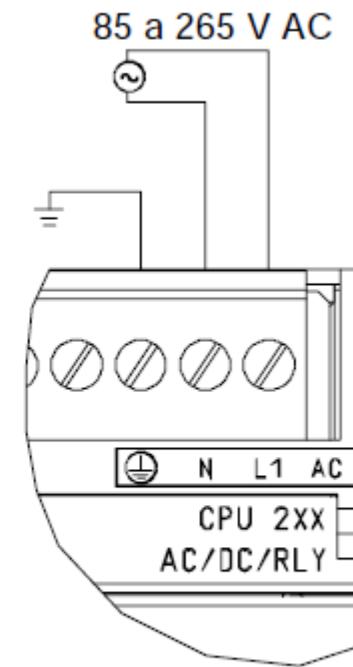
| Módulos de ampliación | Tipo de datos | | | |
|-----------------------------|---|--------------------------------------|--|--|
| Módulos digitales | | | | |
| Entrada | 8 entradas DC | 8 entradas AC | 16 entradas DC | |
| Salida | 4 salidas DC | 4 salidas de relé | 8 salidas de relé | |
| | 8 salidas DC | 8 salidas AC | | |
| Combinación | 4 entradas DC / 4 salidas DC | 8 entradas DC / 8 salidas DC | 16 entradas DC / 16 salidas DC | 32 entradas DC / 32 salidas DC |
| | 4 entradas DC / 4 salidas de relé | 8 entradas DC / 8 salidas de relé | 16 entradas DC / 16 salidas de relé | 32 entradas DC / 32 salidas de relé |
| Módulos analógicos | | | | |
| Entrada | 4 entradas analógicas | 8 entradas analógicas | 4 entradas termopar | 8 entradas termopar |
| | 2 entradas RTD | 4 entradas RTD | | |
| Salida | 2 salidas analógicas | 4 salidas analógicas | | |
| Combinación | 4 entradas analógicas 4 salidas analógicas | | | |
| Módulos inteligentes | | | | |
| | Posición | Módem | PROFIBUS-DP | |
| | Ethernet | Ethernet IT | | |

CONECTAR LA ALIMENTACIÓN DEL S7--200

Alimentación DC

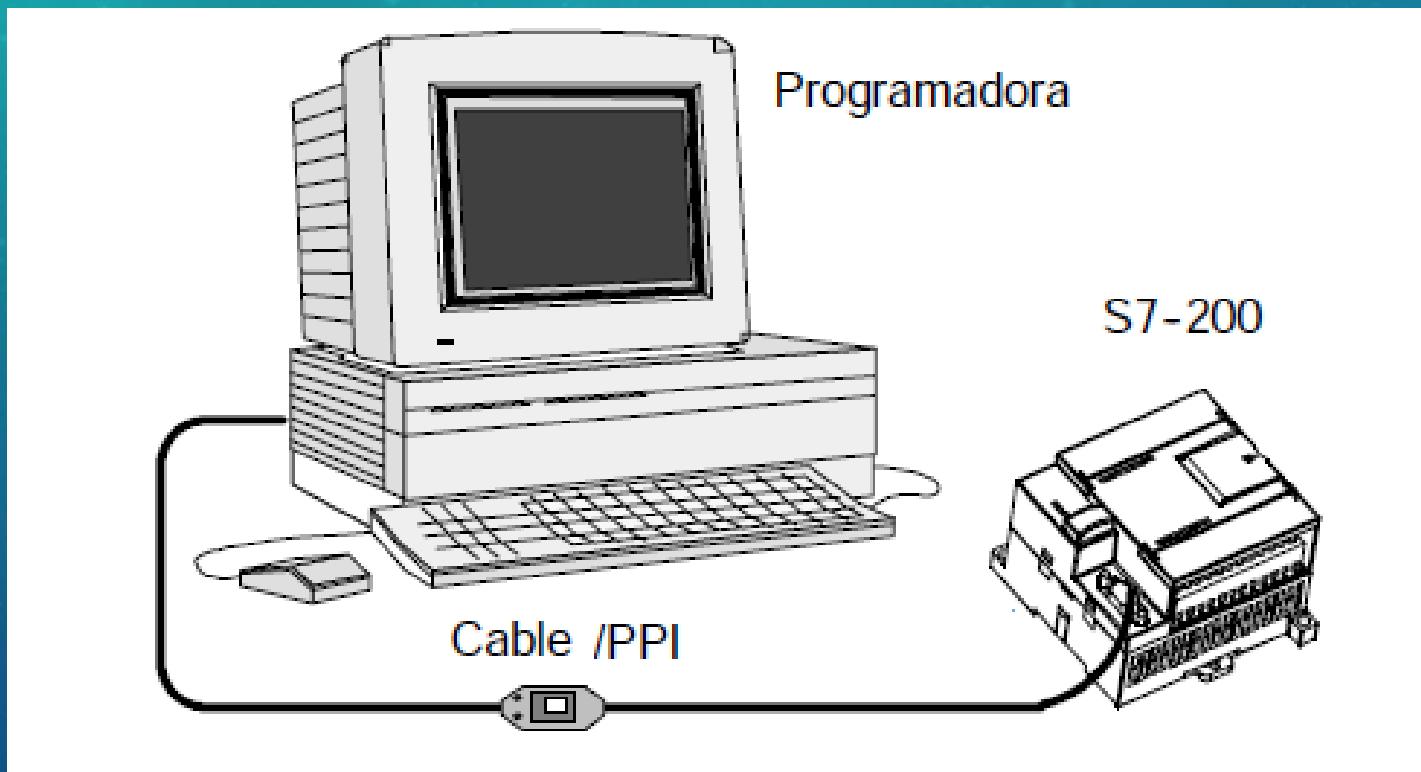


Alimentación AC



Conectar la alimentación del S7-200

CONECTAR EL CABLE USB/PPI



MPI (Multi Point Interface), PPI (Point to Point Interface)

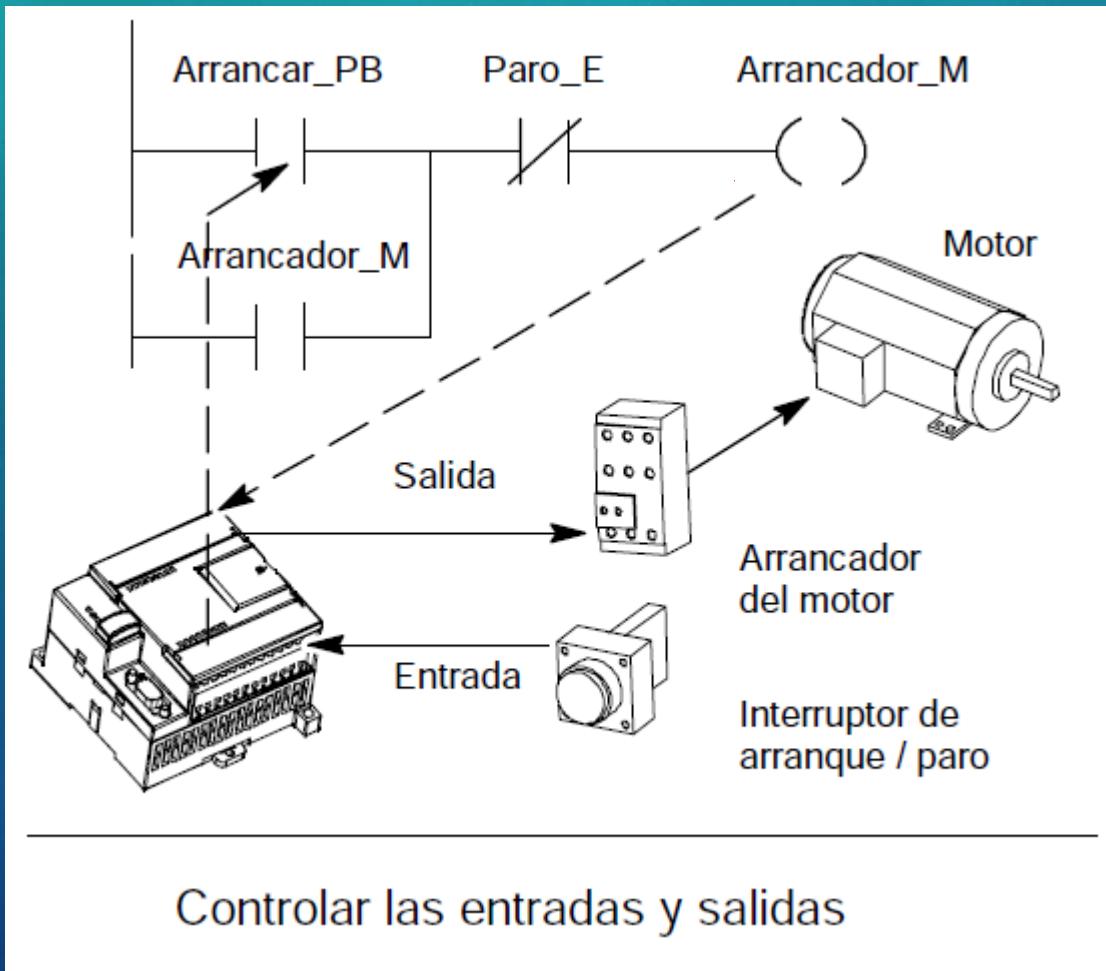
PROTOCOLO PPI

- PPI es un protocolo maestroesclavo. Los maestros envían peticiones a los esclavos y éstos responden. Los esclavos no inicián mensajes, sino que esperan a que un maestro les envíe una petición o solicite una respuesta.

PROTOCOLO PPI

- Los maestros se comunican con los esclavos vía un enlace compartido que es gestionado por el protocolo PPI. El protocolo PPI no limita el número de maestros que se pueden comunicar con un mismo esclavo Sin embargo la red no puede comprender más de 32 maestros.

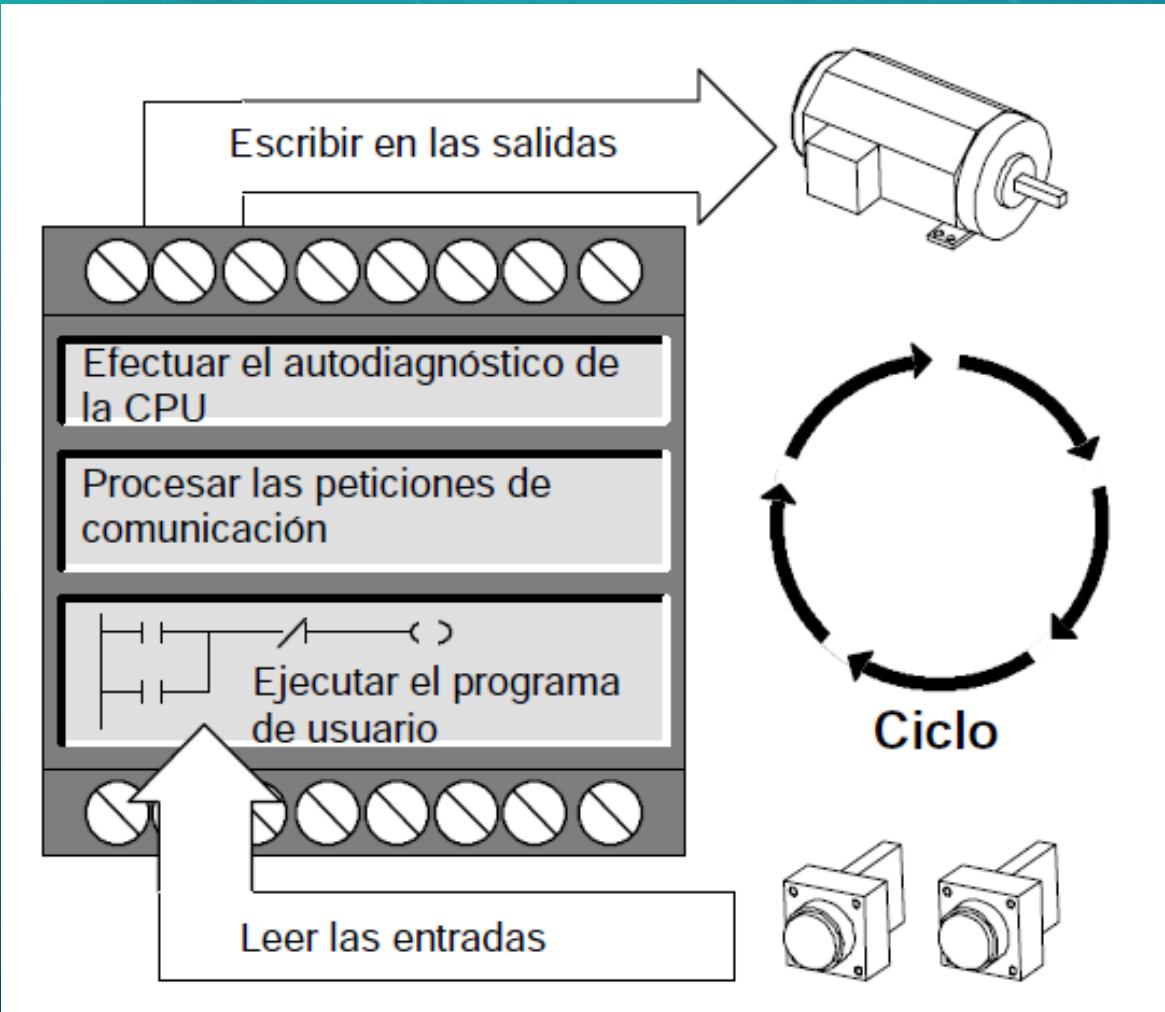
EL S7--200 RELACIONA EL PROGRAMA CON LAS ENTRADAS Y SALIDAS FÍSICAS



SCAN - EL S7 200 EJECUTA LAS SIGUIENTES TAREAS EN UN CICLO

1. Leer las entradas: el S7--200 copia el estado de las entradas físicas en la imagen de proceso de las entradas.
2. Ejecutar la lógica de control en el programa: el S7--200 ejecuta las operaciones del programa y guarda los valores en las distintas áreas de memoria.
3. Procesar las peticiones de comunicación: el S7--200 ejecuta las tareas necesarias para la comunicación.
4. Efectuar el autodiagnóstico de la CPU: el S7--200 verifica si el firmware, la memoria del programa y los módulos de ampliación están trabajando correctamente.
5. Escribir en las salidas: los valores almacenados en la imagen de proceso de las salidas se escriben en las salidas físicas.

CICLO DEL S7--200



DATOS DEL S7--200

| Representación | Byte (B) | Palabra (W) | Palabra doble (D) |
|------------------|------------|------------------|---------------------------------|
| Entero sin signo | 0 a 255 | 0 a 65.535 | 0 a 4.294.967.295 |
| | 0 a FF | 0 a FFFF | 0 a FFFF FFFF |
| Entero con signo | 128 a +127 | 32.768 a +32.767 | -2.147.483.648 a +2.147.483.647 |
| | 80 a 7F | 8000 a 7FFF | 8000 0000 a 7FFF FFFF |

I 3 . 4

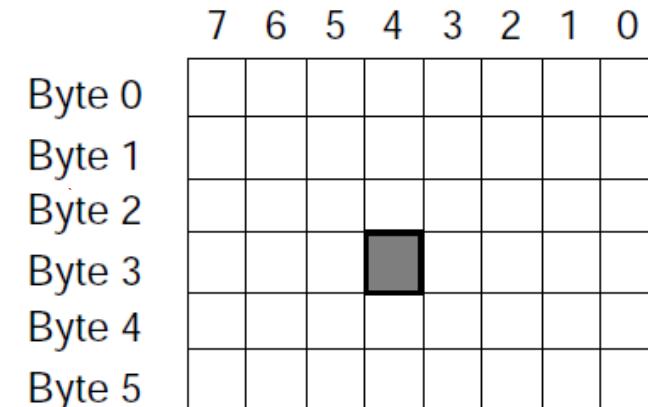
Bit del byte o número de bit: bit 4 de 8 (0 a 7)

Punto decimal que separa la dirección de byte del número de bit

Dirección de byte: byte 3 (el cuarto byte)

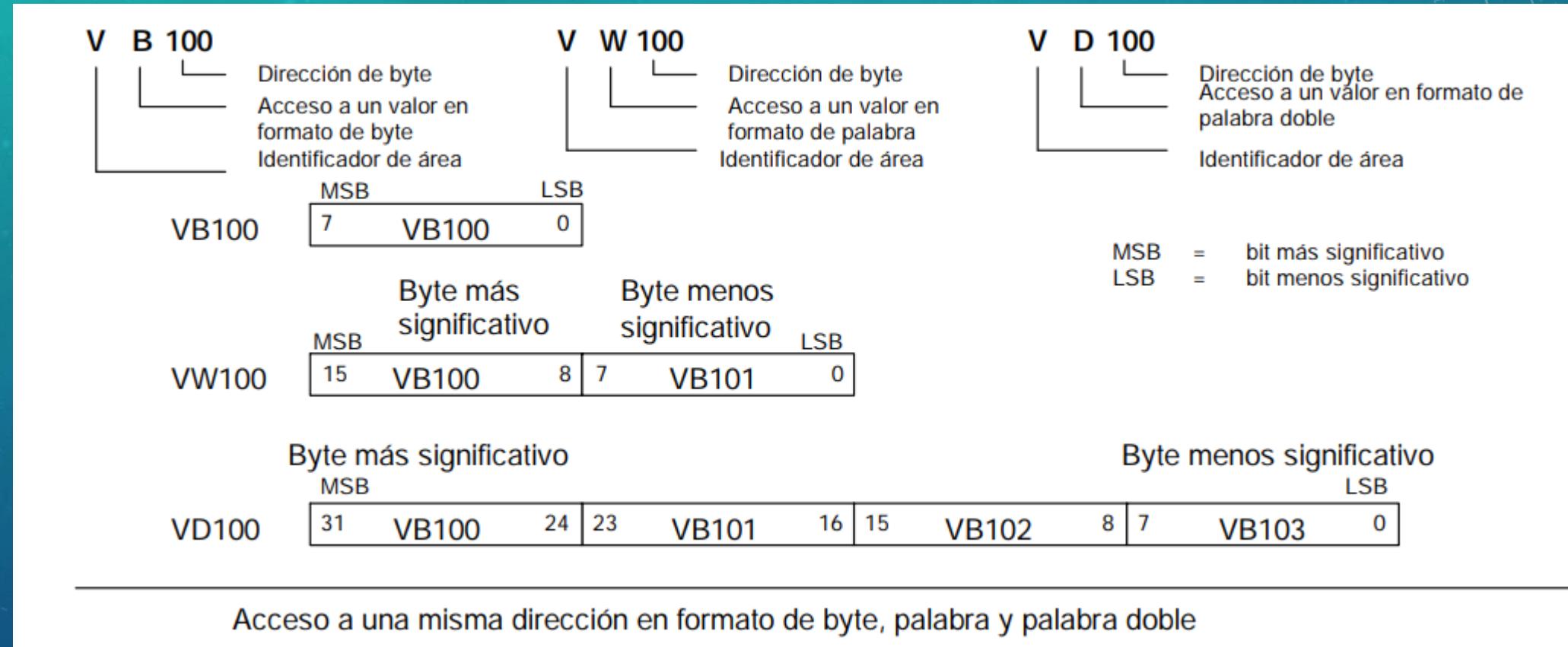
Identificador de área

Imagen de proceso de las entradas (I)



Direccionamiento byte.bit

- Utilizando el formato de dirección de byte se puede acceder a los datos de la mayoría de las áreas de memoria (V, I, Q, M, S, L y SM) en formato de bytes, palabras o palabras dobles.



TIPOS DE DATOS EN LAS ÁREAS DE MEMORIA

- Imagen de proceso de las entradas: I – Nomenclatura Inglesa, E - Nomenclatura Alemana
- Imagen de proceso de las salidas: Q - Nomenclatura Inglesa, A - Nomenclatura Alemana
- Memoria de variables: V
- Área de marcas: M
- Área de temporizadores: T
- Área de contadores: C
- Contadores rápidos: HC
- Acumuladores: AC
- Marcas especiales: SM
- Entradas analógicas: AI
- Salidas analógicas: AQ
- Relés de control secuencial (SCR): S

IMAGEN DE PROCESO DE LAS ENTRADAS: I

- El S7--200 lee las entradas físicas al comienzo de cada ciclo y escribe los correspondientes valores en la imagen de proceso de las entradas. A ésta última se puede acceder en formato de bit, byte, palabra o palabra doble.

IMAGEN DE PROCESO DE LAS SALIDAS: Q

- Al final de cada ciclo, el S7--200 copia en las salidas físicas el valor almacenado en la imagen de proceso de las salidas. A ésta última se puede acceder en formato de bit, byte, palabra o palabra doble.

MEMORIA DE VARIABLES: V

- La memoria de variables (memoria V) se puede utilizar para almacenar los resultados intermedios calculados por las operaciones en el programa. La memoria V también permite almacenar otros datos relativos al proceso o a la tarea actuales. A la memoria V se puede acceder en formato de bit, byte, palabra o palabra doble.

ÁREA DE MARCAS: M

- El área de marcas (memoria M) se puede utilizar como relés de control para almacenar el estado inmediato de una operación u otra información de control. Al área de marcas se puede acceder en formato de bit, byte, palabra o palabra doble.

ÁREA DE TEMPORIZADORES: T

- Los temporizadores del S7-200 tienen resoluciones (intervalos) de 1 ms, 10 ms ó 100 ms.

Existen dos variables asociadas a los temporizadores:

- Valor actual: en este número entero de 16 bits con signo se almacena el valor de tiempo contado por el temporizador.
- Bit del temporizador (bit T): este bit se activa o se desactiva como resultado de la comparación del valor actual con el valor de preselección. Éste último se introduce como parte de la operación del temporizador.

ÁREA DE CONTADORES: C

Los contadores del S7--200 son elementos que cuentan los cambios de negativo a positivo en la(s) entrada(s) de conteo. Hay contadores que cuentan sólo adelante, otros que cuentan sólo atrás y otros cuentan tanto adelante como atrás. Existen dos variables asociadas a los contadores:

- Valor actual: en este número entero de 16 bits con signo se almacena el valor de conteo acumulado.
- Bit del contador (bit C): este bit se activa o se desactiva como resultado de la comparación del valor actual con el valor de preselección. El valor de preselección se introduce como parte de la operación del contador.

MARCAS ESPECIALES: SM

- Las marcas especiales (SM) permiten intercambiar datos entre la CPU y el programa. Estas marcas se puede utilizar para seleccionar y controlar algunas funciones especiales de la CPU S7--200. Por ejemplo, hay una marca que se activa sólo en el primer ciclo, marcas que se activan y se desactivan en determinados intervalos, o bien marcas que muestran el estado de las operaciones matemáticas y de otras operaciones. A las marcas especiales se puede acceder en formato de bit, byte, palabra o palabra doble.

SMB0: Bits de estado

Como muestra la tabla D-1, SMB0 contiene ocho bits de estado que la CPU S7-200 actualiza al final de cada ciclo.

Tabla D-1 Byte de marcas SMB0 (SM0.0 a SM0.7)

| Bits SM | Descripción (sólo lectura) |
|---------|--|
| SM0.0 | Este bit siempre está activado. |
| SM0.1 | Este bit se activa en el primer ciclo. Se utiliza p. ej. para llamar una subrutina de inicialización. |
| SM0.2 | Este bit se activa durante un ciclo si se pierden los datos remanentes. Se puede utilizar como marca de error o como mecanismo para llamar a una secuencia especial de arranque. |
| SM0.3 | Este bit se activa durante un ciclo cuando se pasa a modo RUN tras conectarse la alimentación. Se puede utilizar durante el tiempo de calentamiento de la instalación antes del funcionamiento normal. |
| SM0.4 | Este bit ofrece un reloj que está activado durante 30 segundos y desactivado durante 30 segundos, siendo el tiempo de ciclo de 1 minuto. Ofrece un retardo fácil de utilizar o un reloj de 1 minuto. |
| SM0.5 | Este bit ofrece un reloj que está activado durante 0,5 segundos y desactivado durante 0,5 segundos, siendo el tiempo de ciclo de 1 segundo. Ofrece un retardo fácil de utilizar o un reloj de 1 segundo. |
| SM0.6 | Este bit es un reloj que está activado en un ciclo y desactivado en el ciclo siguiente. Se puede utilizar como entrada de conteo de ciclos. |
| SM0.7 | Este bit indica la posición del selector de modos de operación (OFF = TERM; ON = RUN). Si el bit se utiliza para habilitar el modo Freeport cuando el selector esté en posición RUN, se podrá habilitar la comunicación normal con la programadora cambiando el selector a TERM. |

ENTRADAS ANALÓGICAS: AI

- El S7--200 convierte valores reales analógicos (p. ej. temperatura, tensión, etc.) en valores digitales en formato de palabra (de 16 bits). A estos valores se accede con un identificador de área (AI), seguido del tamaño de los datos (W) y de la dirección del byte inicial. Puesto que las entradas analógicas son palabras que comienzan siempre en bytes pares (p. ej. 0, 2, 4, etc.), es preciso utilizar direcciones con bytes pares (p. ej. AIW0, AIW2, AIW4, etc.) para acceder a las mismas. Las entradas analógicas son valores de sólo lectura.

SALIDAS ANALÓGICAS: AQ

- El S7--200 convierte valores digitales en formato de palabra (de 16 bits) en valores reales analógicos (p. ej. intensidad o tensión). Estos valores analógicos son proporcionales a los digitales. A los valores analógicos se accede con un identificador de área (AQ), seguido del tamaño de los datos (W) y de la dirección del byte inicial. Puesto que las salidas analógicas son palabras que comienzan siempre en bytes pares (p. ej. 0, 2, 4, etc.), es preciso utilizar direcciones con bytes pares (p. ej. AQW0, AQW2, AQW4, etc.) para acceder a las mismas. Las salidas analógicas son valores de sólo escritura.

RELÉS DE CONTROL SECUENCIAL (SCR): S

- Los relés de control secuencial (SCR o bits S) permiten organizar los pasos del funcionamiento de una máquina en segmentos equivalentes en el programa. Los SCRs permiten segmentar lógicamente el programa de control. A los relés de control secuencial (SCR) se puede acceder en formato de bit, byte, palabra o palabra doble.

ÁREAS DE MEMORIA Y FUNCIONES DEL S7--200

| Descripción | CPU 221 | CPU 222 | CPU 224 | CPU 224XP, CPU 224XPsi | CPU 226 |
|--|--------------------------|--------------------------|---------------------------|----------------------------|----------------------------|
| Tamaño del programa de usuario con edición en runtime sin edición en runtime | 4096 bytes 4096 bytes | 4096 bytes 4096 bytes | 8192 bytes 12288 bytes | 12288 bytes 16384 bytes | 16384 bytes 24576 bytes |
| Tamaño de los datos de usuario | 2048 bytes | 2048 bytes | 8192 bytes | 10240 bytes | 10240 bytes |
| Imagen de proceso de las entradas | I0.0 a I15.7 | I0.0 a I15.7 | I0.0 a I15.7 | I0.0 a I15.7 | I0.0 a I15.7 |
| Imagen de proceso de las salidas | Q0.0 a Q15.7 | Q0.0 a Q15.7 | Q0.0 a Q15.7 | Q0.0 a Q15.7 | Q0.0 a Q15.7 |
| Entradas analógicas (sólo lectura) | AIW0 a AIW30 | AIW0 a AIW30 | AIW0 a AIW62 | AIW0 a AIW62 | AIW0 a AIW62 |
| Salidas analógicas (sólo escritura) | AQW0 a AQW30 | AQW0 a AQW30 | AQW0 a AQW62 | AQW0 a AQW62 | AQW0 a AQW62 |
| Memoria de variables (V) | VB0 a VB2047 | VB0 a VB2047 | VB0 a VB8191 | VB0 a VB10239 | VB0 a VB10239 |
| Memoria local (L) ¹ | LB0 a LB63 | LB0 a LB63 | LB0 a LB63 | LB0 a LB63 | LB0 a LB63 |

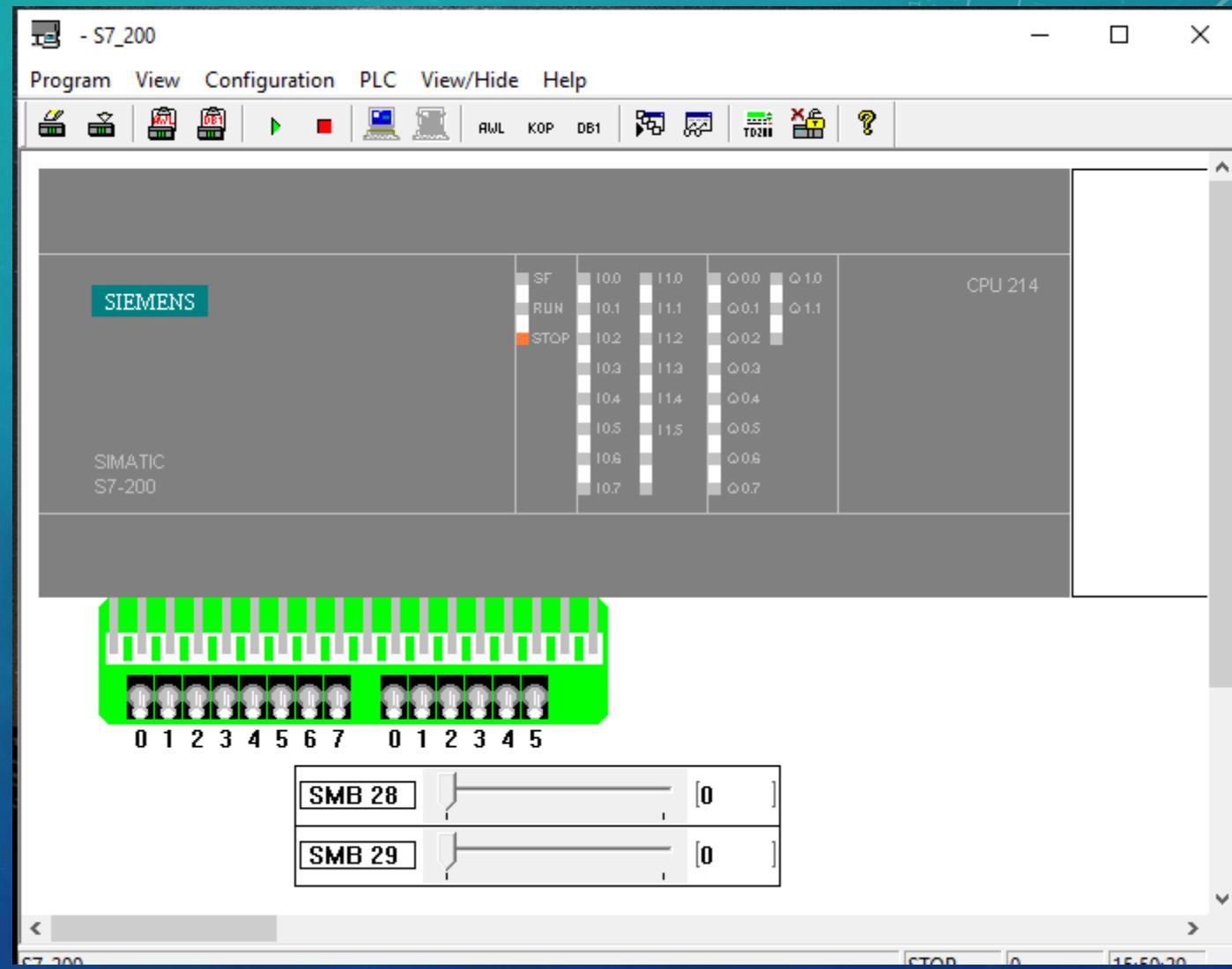
ÁREAS DE MEMORIA Y FUNCIONES DEL S7--200

| Descripción | CPU 221 | CPU 222 | CPU 224 | CPU 224XP, CPU 224XPsi | CPU 226 |
|-----------------------------------|--------------------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| Área de marcas (M) | M0.0 a M31.7 | M0.0 a M31.7 | M0.0 a M31.7 | M0.0 a M31.7 | M0.0 a M31.7 |
| Marcas especiales (SM) | SM0.0 a SM179.7 | SM0.0 a SM299.7 | SM0.0 a SM549.7 | SM0.0 a SM549.7 | SM0.0 a SM549.7 |
| Sólo lectura | SM0.0 a SM29.7 | SM0.0 a SM29.7 | SM0.0 a SM29.7 | SM0.0 a SM29.7 | SM0.0 a SM29.7 |
| Temporizadores | 256 (T0 a T255) | 256 (T0 a T255) | 256 (T0 a T255) | 256 (T0 a T255) | 256 (T0 a T255) |
| Retardo a la conexión con memoria | 1 ms T0, T64 | T0, T64 | T0, T64 | T0, T64 | T0, T64 |
| | 10 ms T1 a T4 y T65 a T68 | T1 a T4 y T65 a T68 | T1 a T4 y T65 a T68 | T1 a T4 y T65 a T68 | T1 a T4 y T65 a T68 |
| | 100 ms T5 a T31 y T69 a T95 | T5 a T31 y T69 a T95 |
| Retardo a la conexión/desconexión | 1 ms T32, T96 | T32, T96 | T32, T96 | T32, T96 | T32, T96 |
| | 10 ms T33 a T36 y T97 a T100 | T33 a T36 y T97 a T100 | T33 a T36 y T97 a T100 | T33 a T36 y T97 a T100 | T33 a T36 y T97 a T100 |
| | 100 ms T37 a T63 y T101 a T255 | T37 a T63 y T101 a T255 |
| Contadores | C0 a C255 | C0 a C255 | C0 a C255 | C0 a C255 | C0 a C255 |

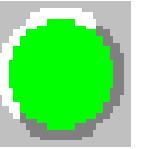
ÁREAS DE MEMORIA Y FUNCIONES DEL S7--200

| Descripción | CPU 221 | CPU 222 | CPU 224 | CPU 224XP, CPU 224XPsi | CPU 226 |
|-----------------------------------|--------------|--------------|--------------|---------------------------|--------------------|
| Contadores | C0 a C255 | C0 a C255 | C0 a C255 | C0 a C255 | C0 a C255 |
| Contadores rápidos | HC0 a HC5 | HC0 a HC5 | HC0 a HC5 | HC0 a HC5 | HC0 a HC5 |
| Relés de control secuencial (S) | S0.0 a S31.7 | S0.0 a S31.7 | S0.0 a S31.7 | S0.0 a S31.7 | S0.0 a S31.7 |
| Acumuladores | AC0 a AC3 | AC0 a AC3 | AC0 a AC3 | AC0 a AC3 | AC0 a AC3 |
| Saltos a metas | 0 a 255 | 0 a 255 | 0 a 255 | 0 a 255 | 0 a 255 |
| Llamadas a subrutinas | 0 a 63 | 0 a 63 | 0 a 63 | 0 a 63 | 0 a 127 |
| Rutinas de interrupción | 0 a 127 | 0 a 127 | 0 a 127 | 0 a 127 | 0 a 127 |
| Detectar flanco positivo/negativo | 256 | 256 | 256 | 256 | 256 |
| Lazos PID | 0 a 7 | 0 a 7 | 0 a 7 | 0 a 7 | 0 a 7 |
| Puertos | Puerto 0 | Puerto 0 | Puerto 0 | Puerto 0, puerto 1 | Puerto 0, puerto 1 |

SIMULADOR S7-200



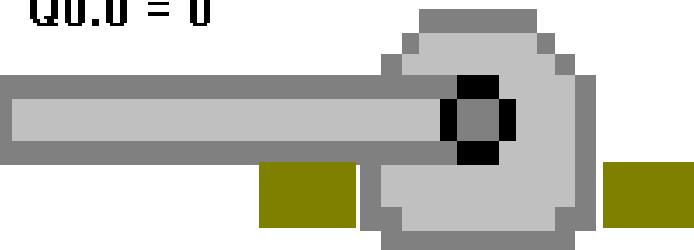
EJERCICIO 1



P - I0.0

Electroválvula monoestable

Q0.0 = 0



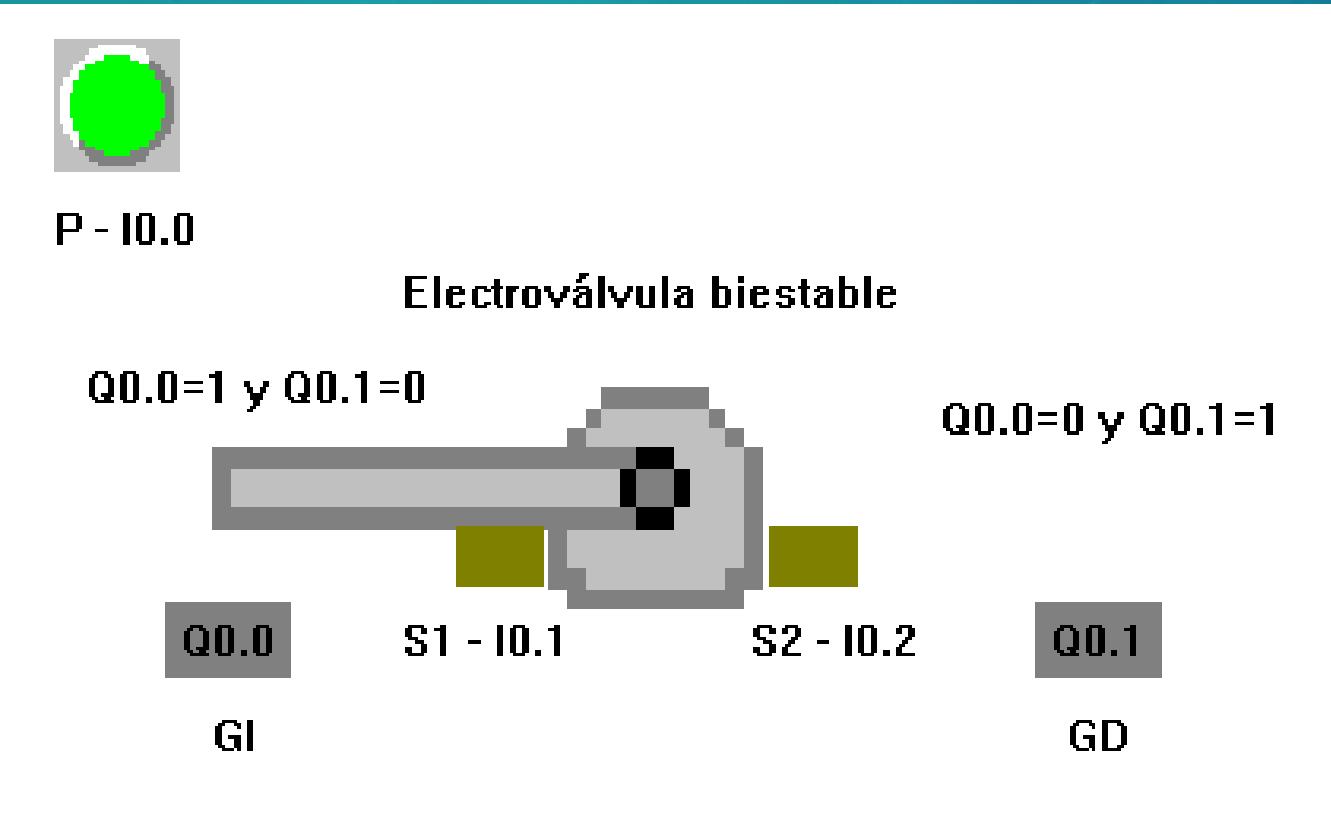
S1 - I0.1

Q0.0 = 1

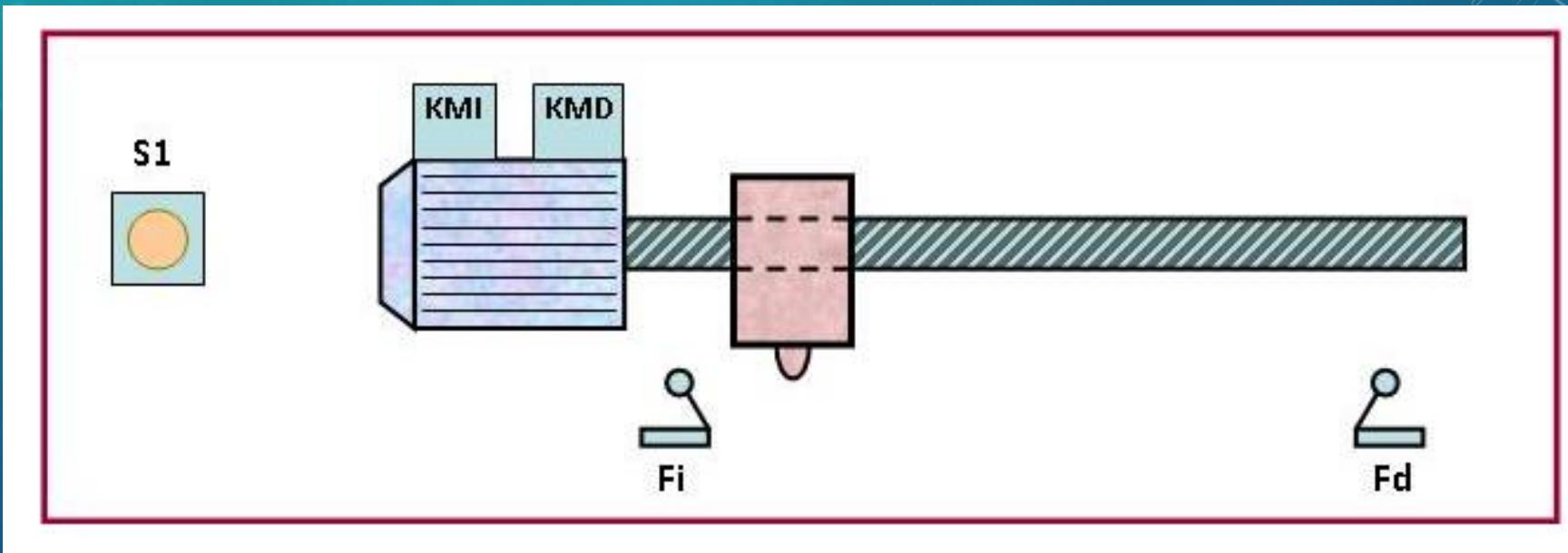
S2 - I0.2

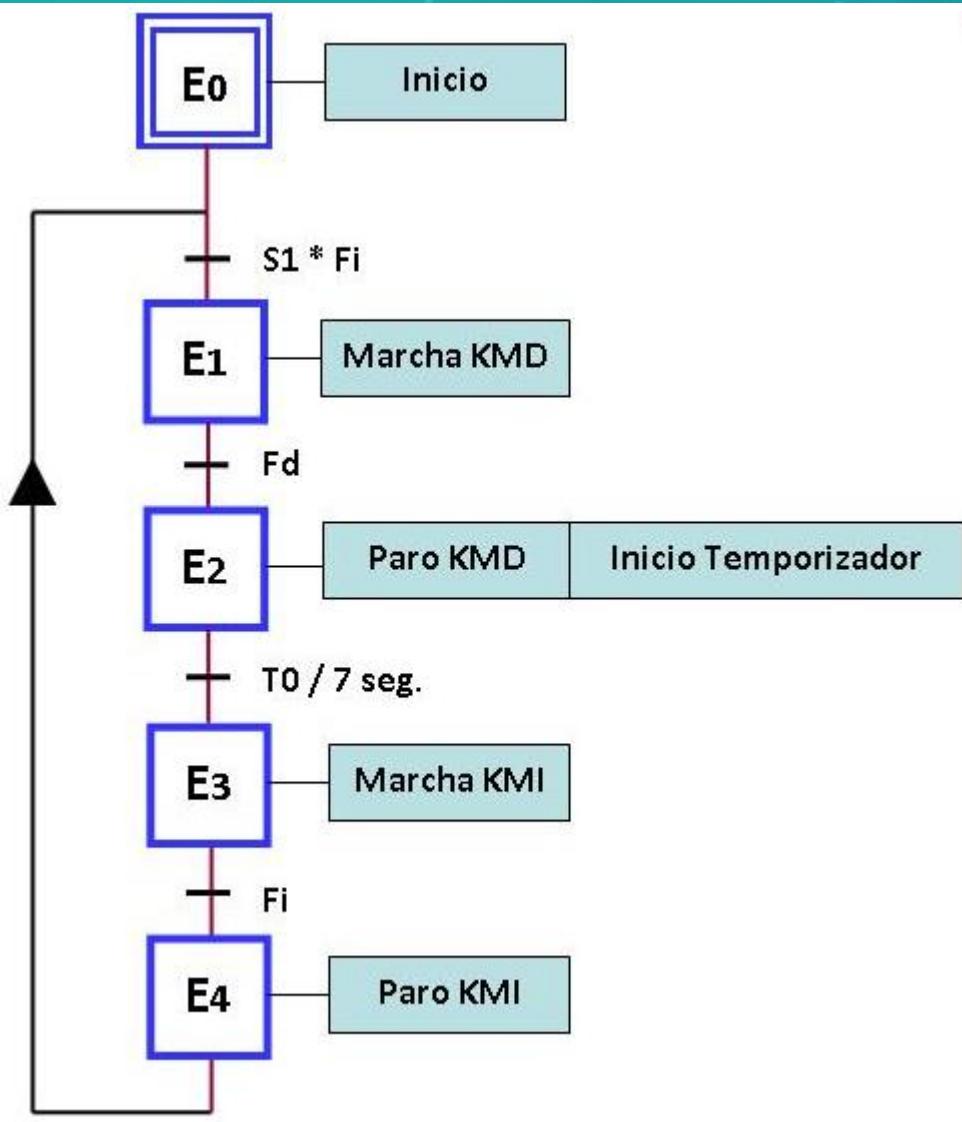
Giro - Q0.0

EJERCICIO 2

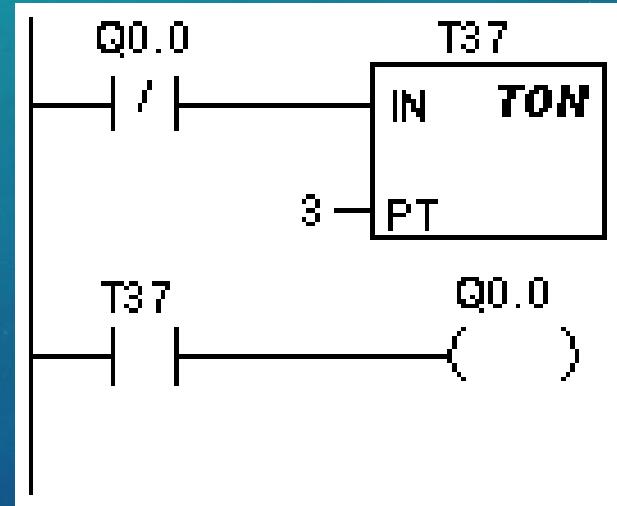
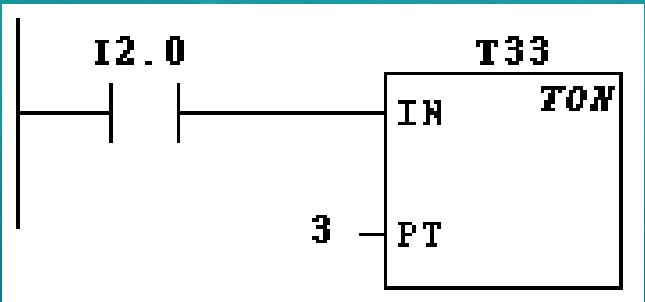
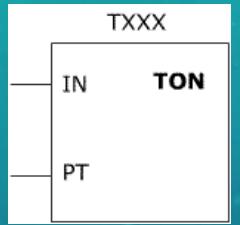


EJERCICIO TEMPORIZADOR





TEMPORIZADOR



Funcionamiento de las operaciones de temporización

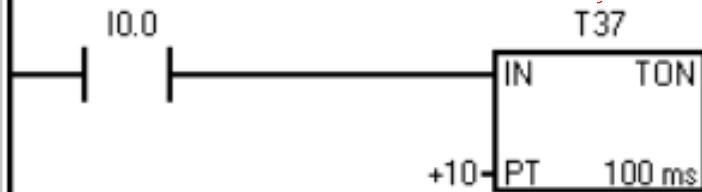
| Tipo de datos | Actual >= Preselección | Estado de la entrada de habilitación (IN) | Alimentación/primer ciclo |
|---------------|--|---|---|
| TON | Bit de temporización ON El valor actual continúa contando hasta 32.767. | ON: El valor actual cuenta el tiempo. OFF: Bit de temporización OFF, valor actual = 0 | Bit de temporización OFF Valor actual = 0 |
| TONR | Bit de temporización ON El valor actual continúa contando hasta 32.767. | ON: El valor actual cuenta el tiempo. OFF: El bit de temporización y el valor actual conservan el último estado. | Bit de temporización OFF El valor actual se puede conservar ¹ |
| TOF | Bit de temporización OFF. Valor actual = valor de preselección, se detiene el conteo. | ON: Bit de temporización ON, valor actual = 0 OFF: El temporizador cuenta tras un cambio de ON a OFF. | Bit de temporización OFF Valor actual = 0 |

Temporizadores y sus resoluciones

| Tipo de temporizador | Resolución | Valor máximo | | Nº de temporizador |
|---------------------------|------------|--------------|--------------|------------------------|
| TONR (con memoria) | 1 ms | 32,767 s | (0,546 min.) | T0, T64 |
| | 10 ms | 327,67 s | (5,46 min.) | T1 a T4, T65 a T68 |
| | 100 ms | 3276,7 s | (54,6 min.) | T5 a T31, T69 a T95 |
| TON, TOF (sin memoria) | 1 ms | 32,767 s | (0,546 min.) | T32, T96 |
| | 10 ms | 327,67 s | (5,46 min.) | T33 a T36, T97 a T100 |
| | 100 ms | 3276,7 s | (54,6 min.) | T37 a T63, T101 a T255 |

Ejemplo: Temporizador como retardo a la conexión (SIMATIC)

Network 1

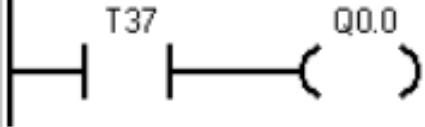


Network 1

//El temporizador T37 de 100 ms se detiene al
//cabo de (10 x 100 ms = 1s)
//I0.0 ON=T37 habilitado,
//I0.0 OFF=inhibir e inicializar T37

LD
TON
I0.0
T37, +10

Network 2



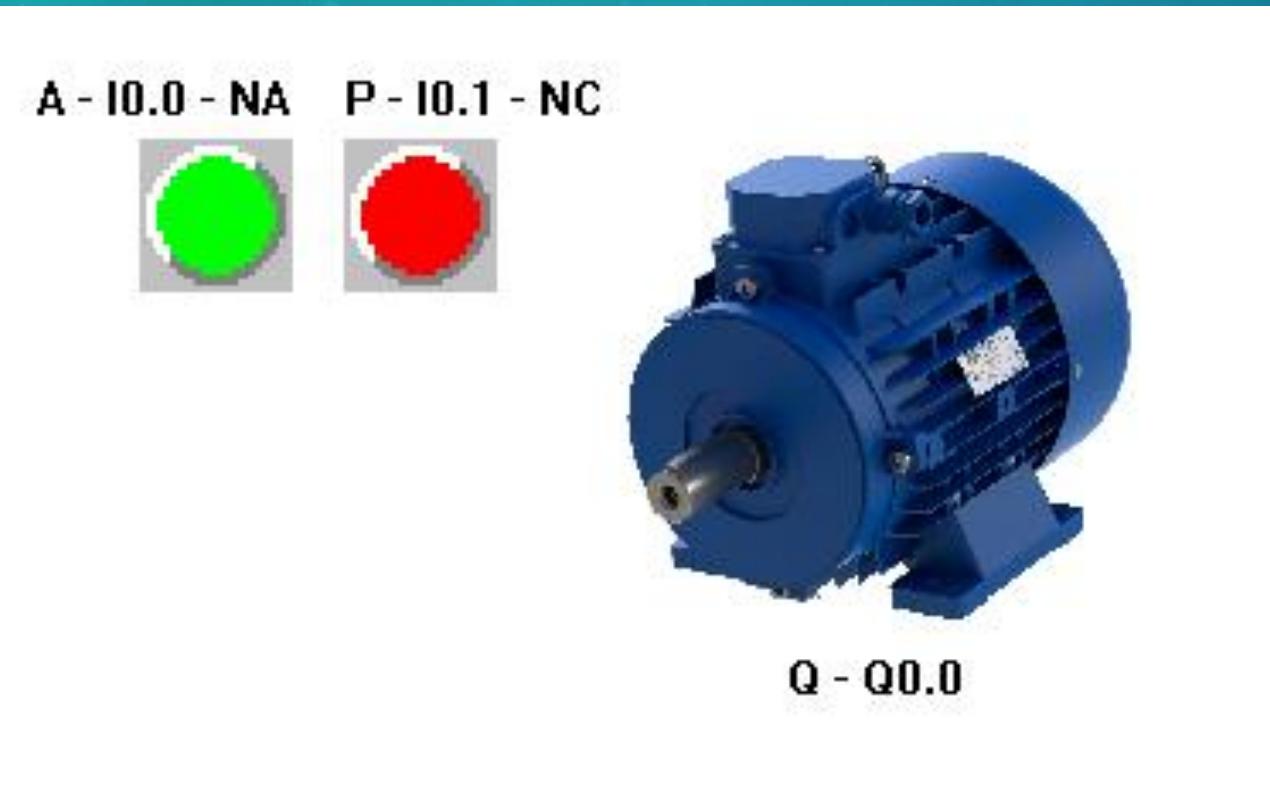
Network 2

//El temporizador T37 controla el bit T37
LD
= T37
Q0.0

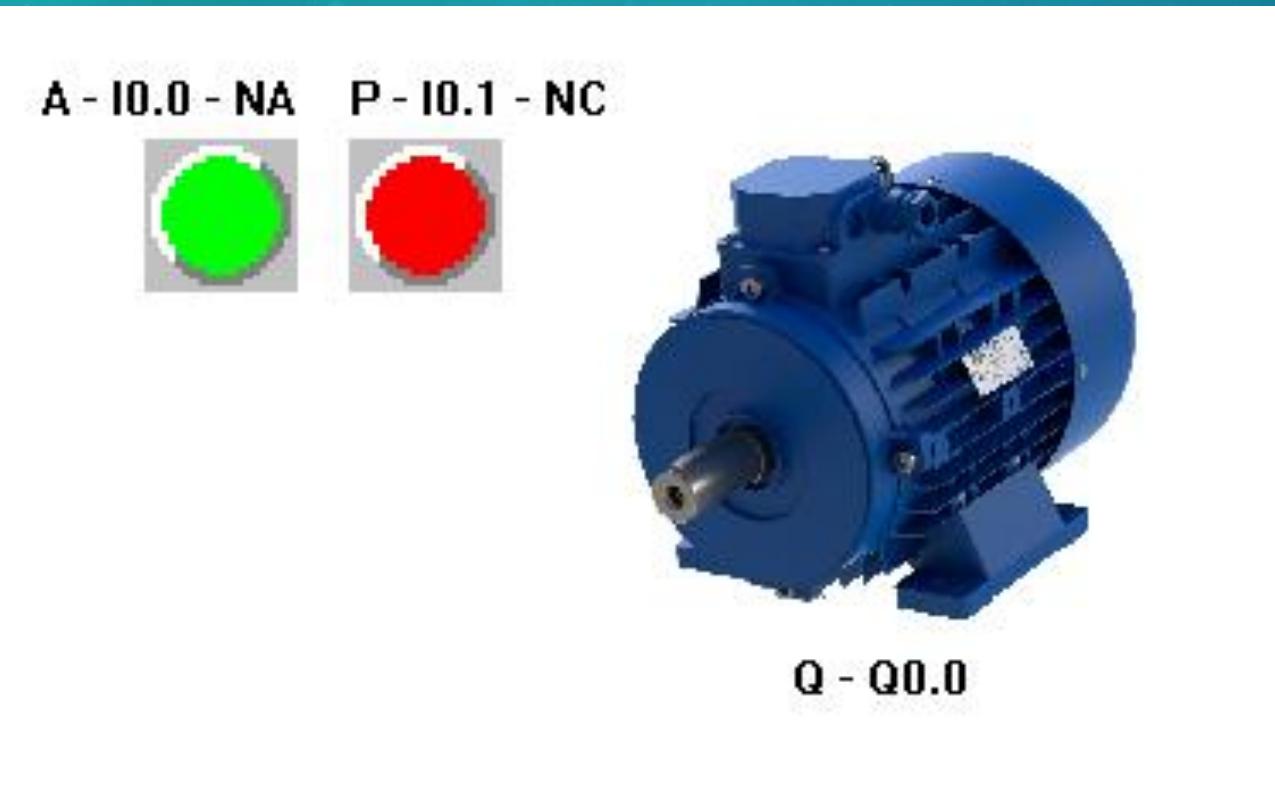
EJERCICIO 3. MARCHA Y PARO



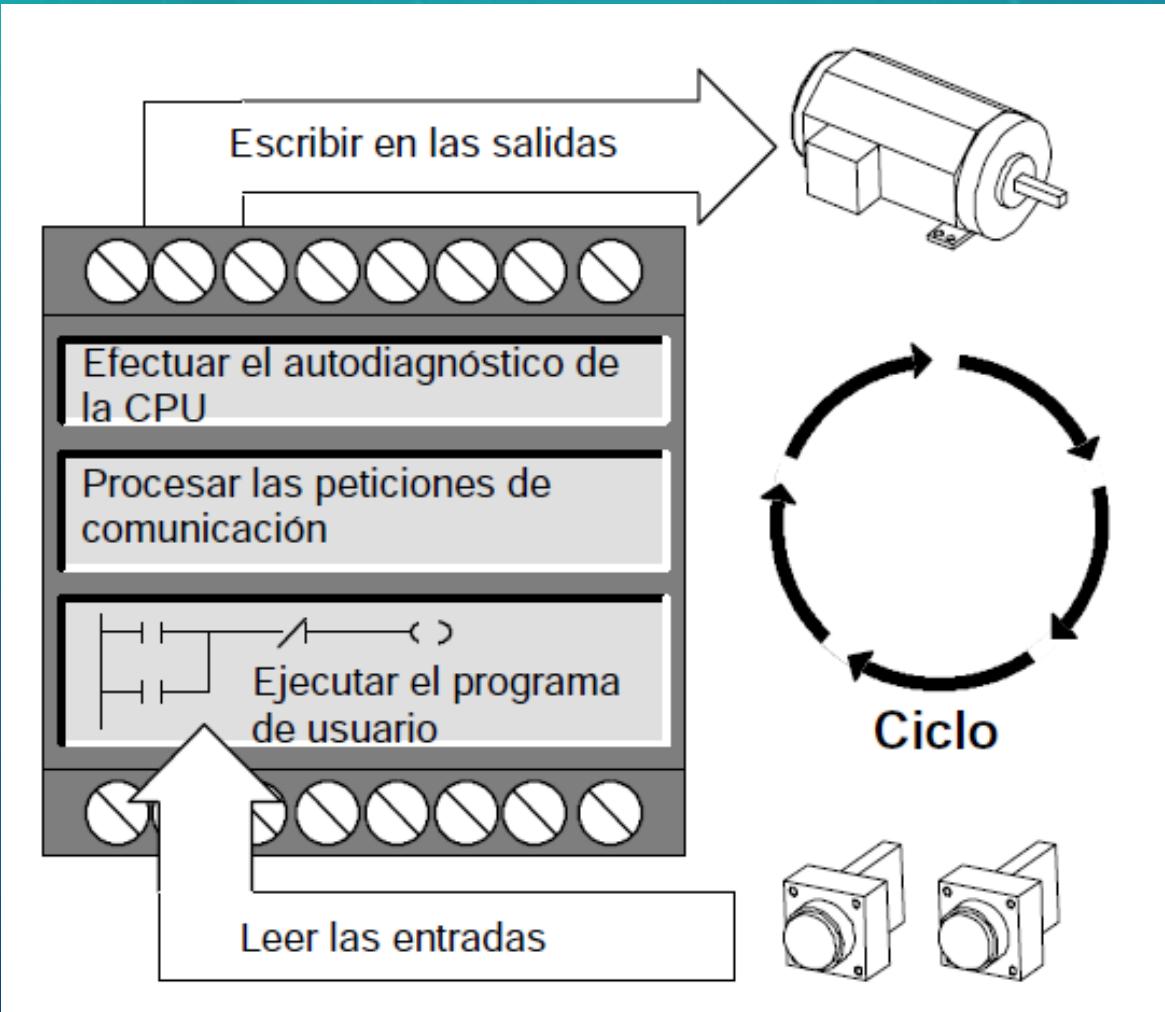
EJERCICIO 4. MARCHA Y PARO

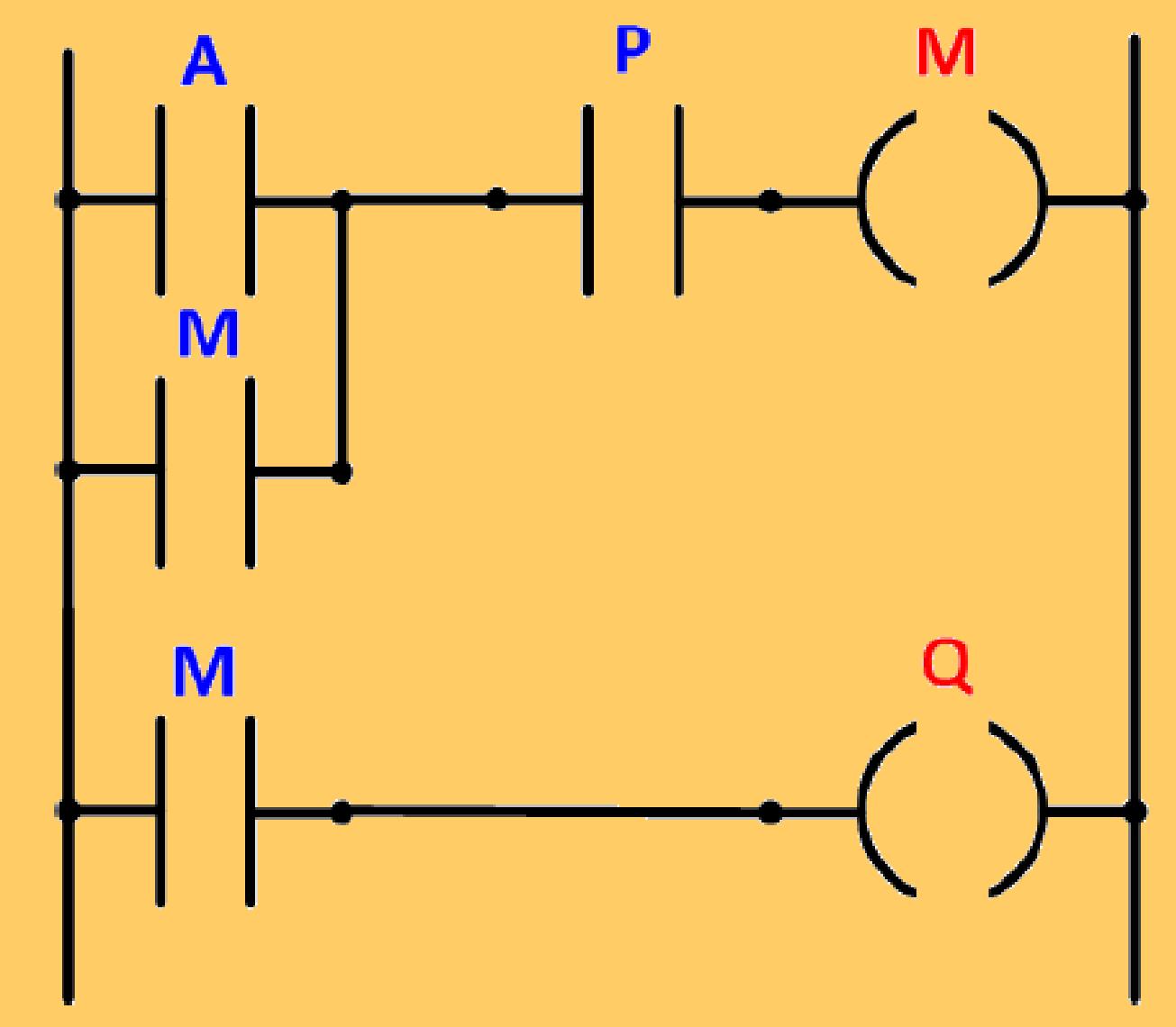


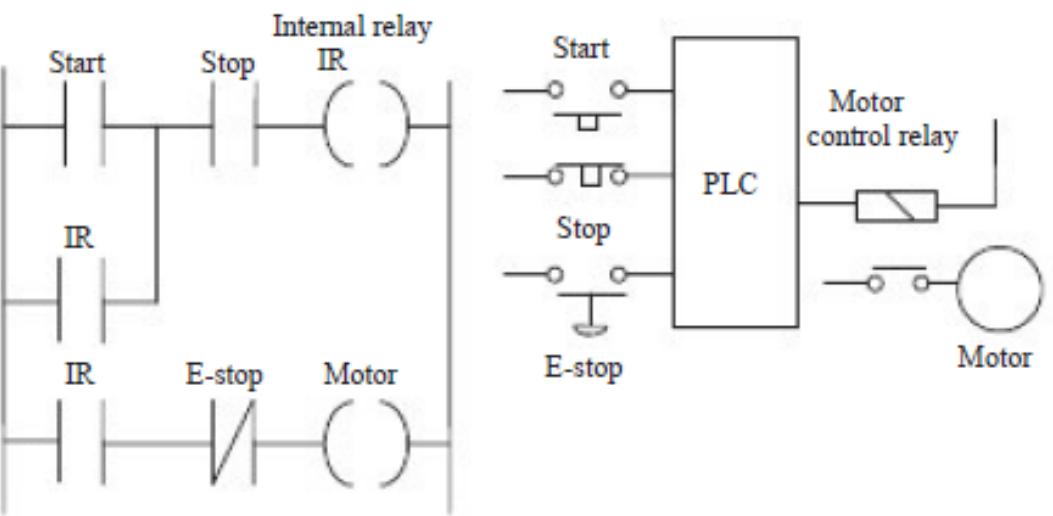
EJERCICIO 4_A: RETROALIMENTACION



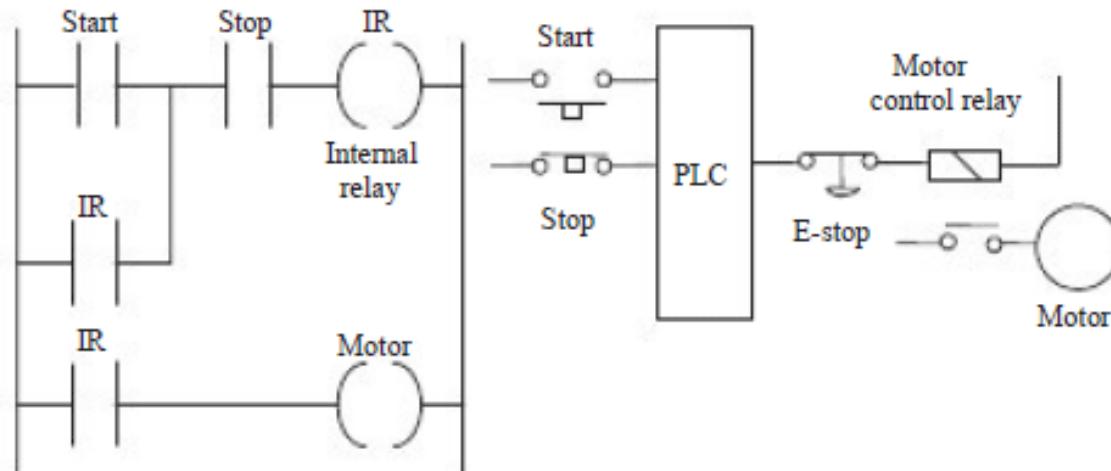
CICLO DEL S7--200



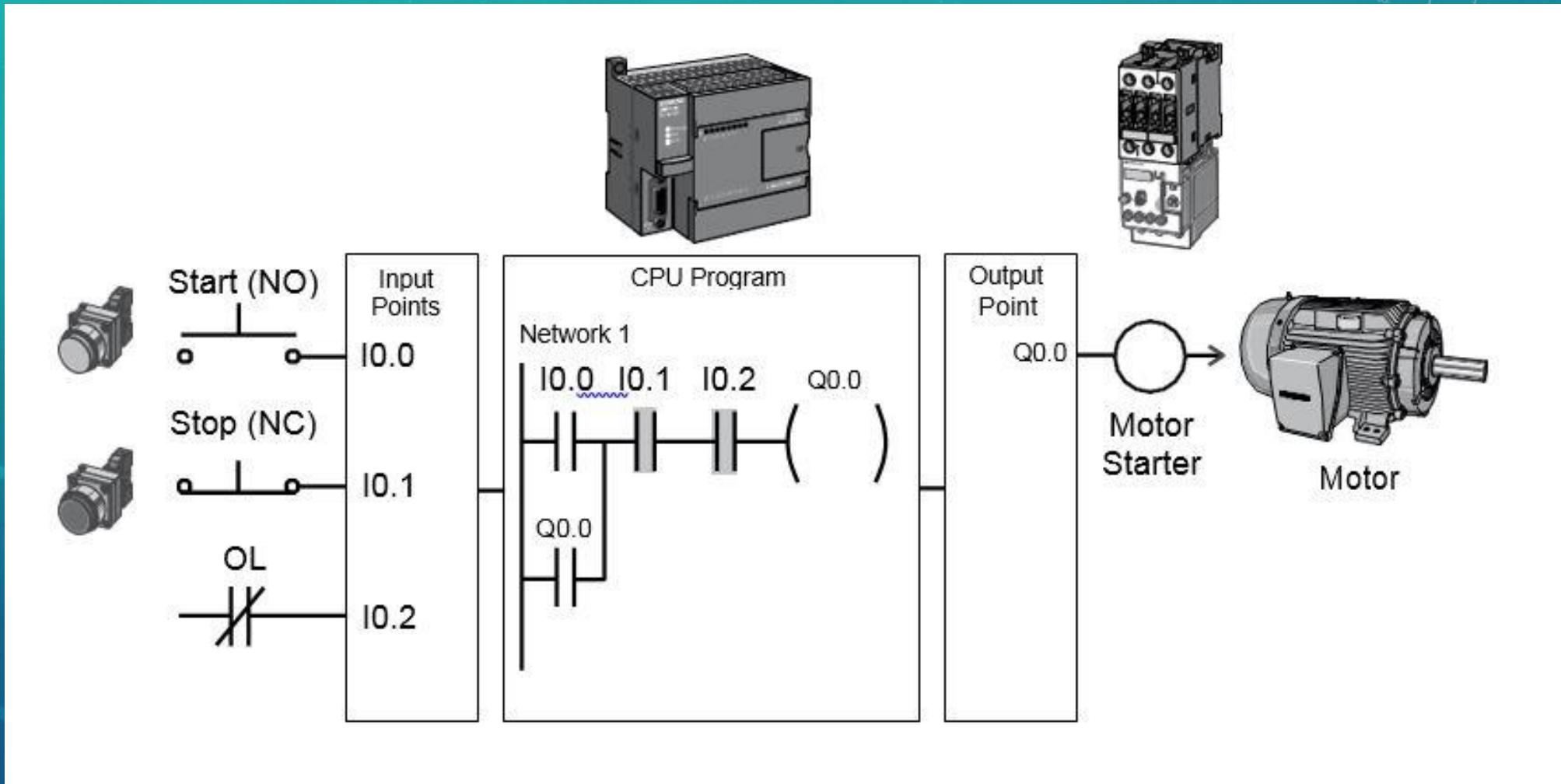


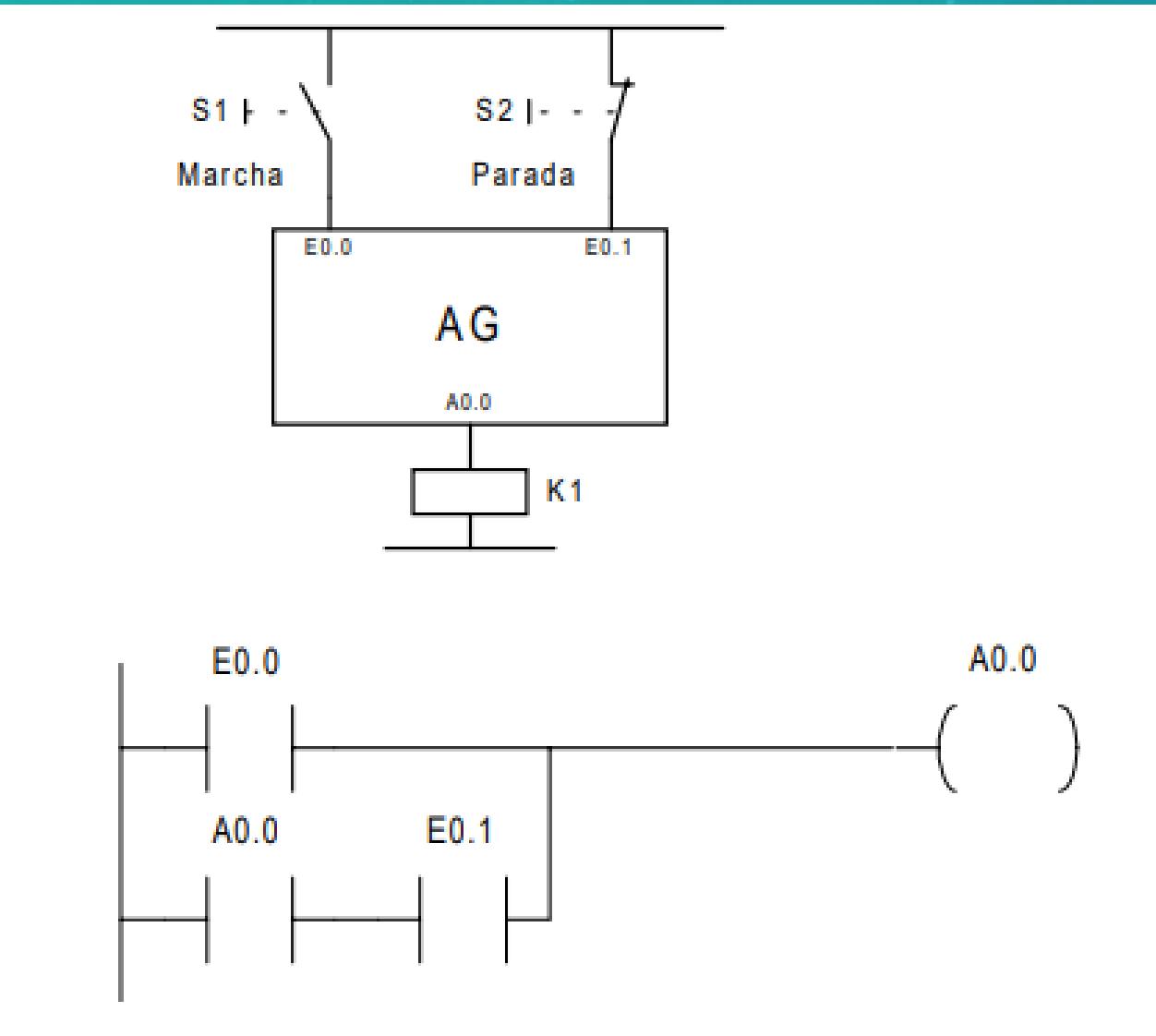


An unsafe emergency stop system.

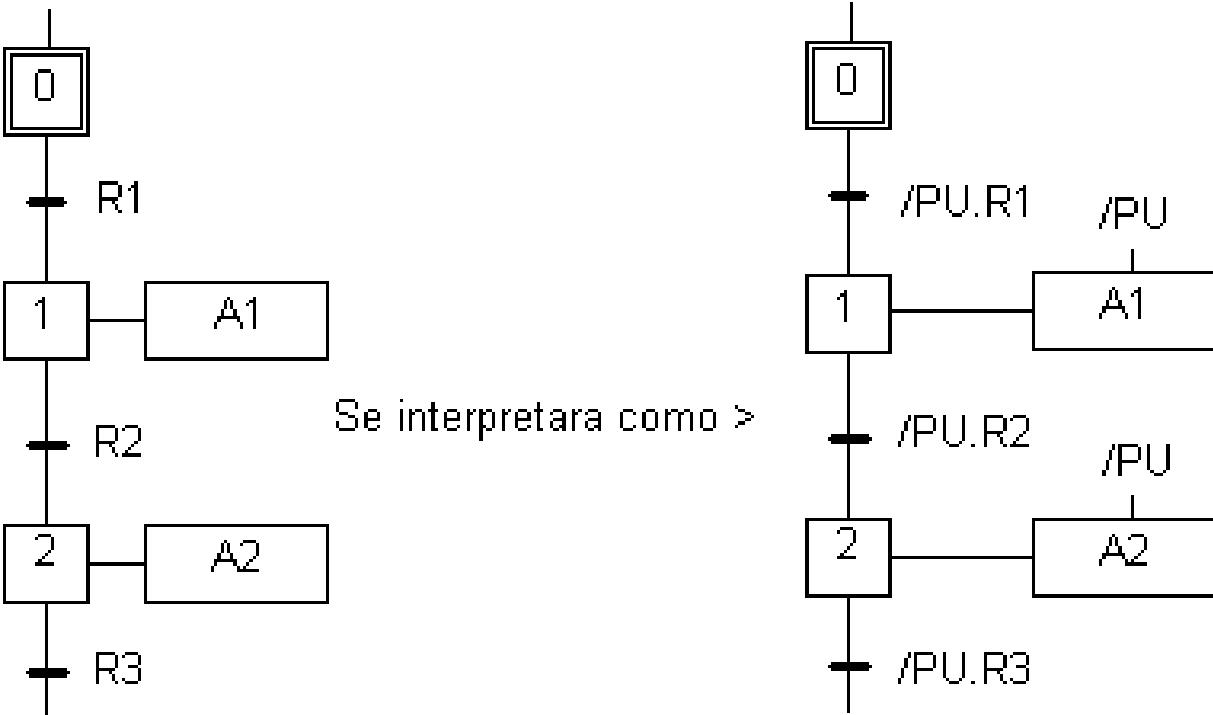


A safer emergency stop system.





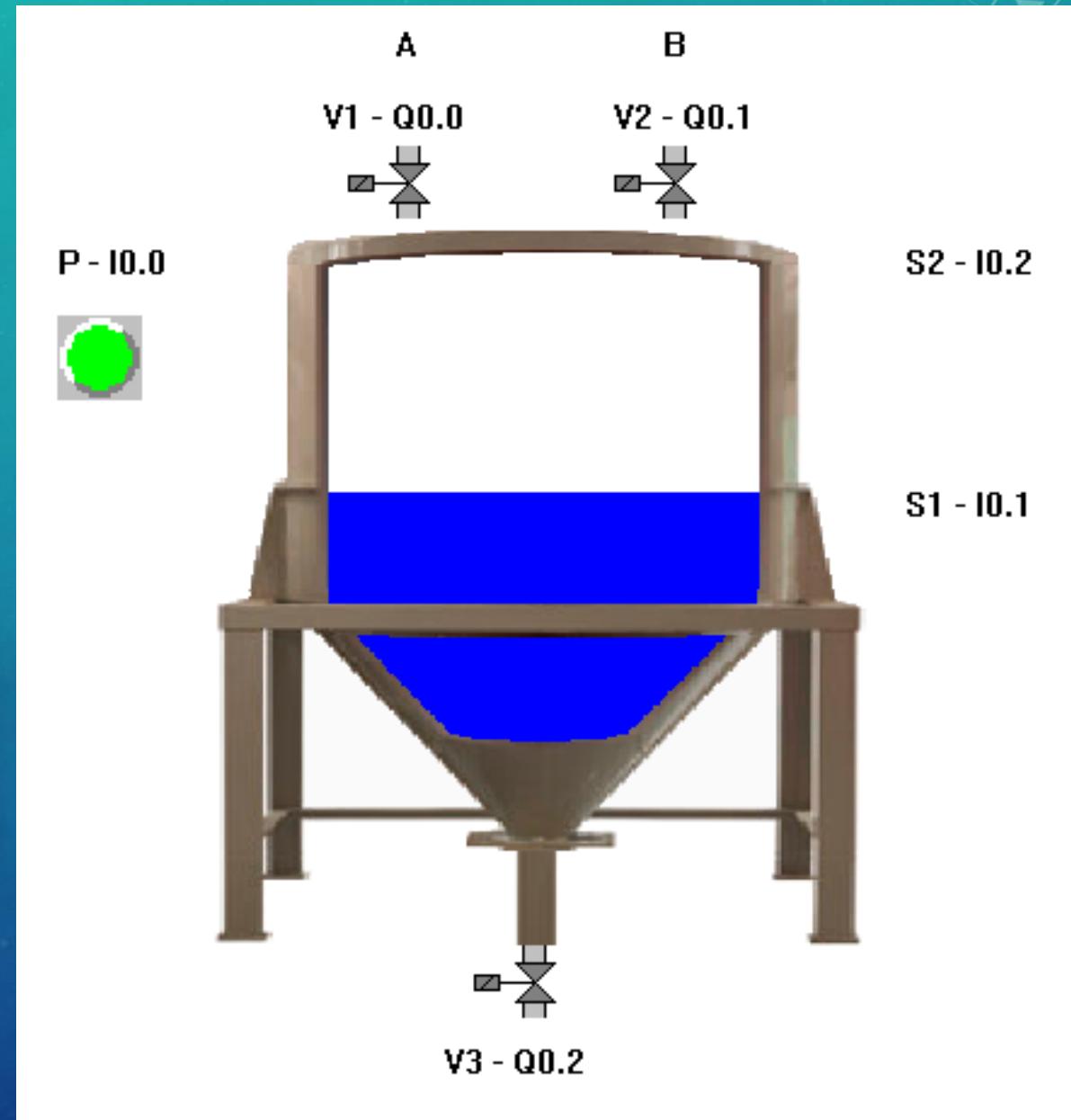
Congelación del GRAFCET convención de interpretación



En el módulo GRAFCET o chart de un PLC deberá encontrarse unas marcas o bits de control que permiten la congelación de su evolución, no suele existir por otro lado la función inhibición.

- Ejercicio 5: Se requiere mezclar 2 productos. La dosificación de los dos productos se realiza en una tolva, primero se vierte el producto A sobre la tolva activando la Válvula **V1** hasta que se alcanza un nivel activando el sensor **S1** y a continuación se añade el producto B Activando la válvula **V2** para conseguir el nivel total de los 2 productos, activando el sensor **S2**, posteriormente se abre la válvula **V3** durante 30 segundos para dejar caer el contenido y se vacíe el depósito para poder iniciar un nuevo ciclo. El proceso se activa con un interruptor P.

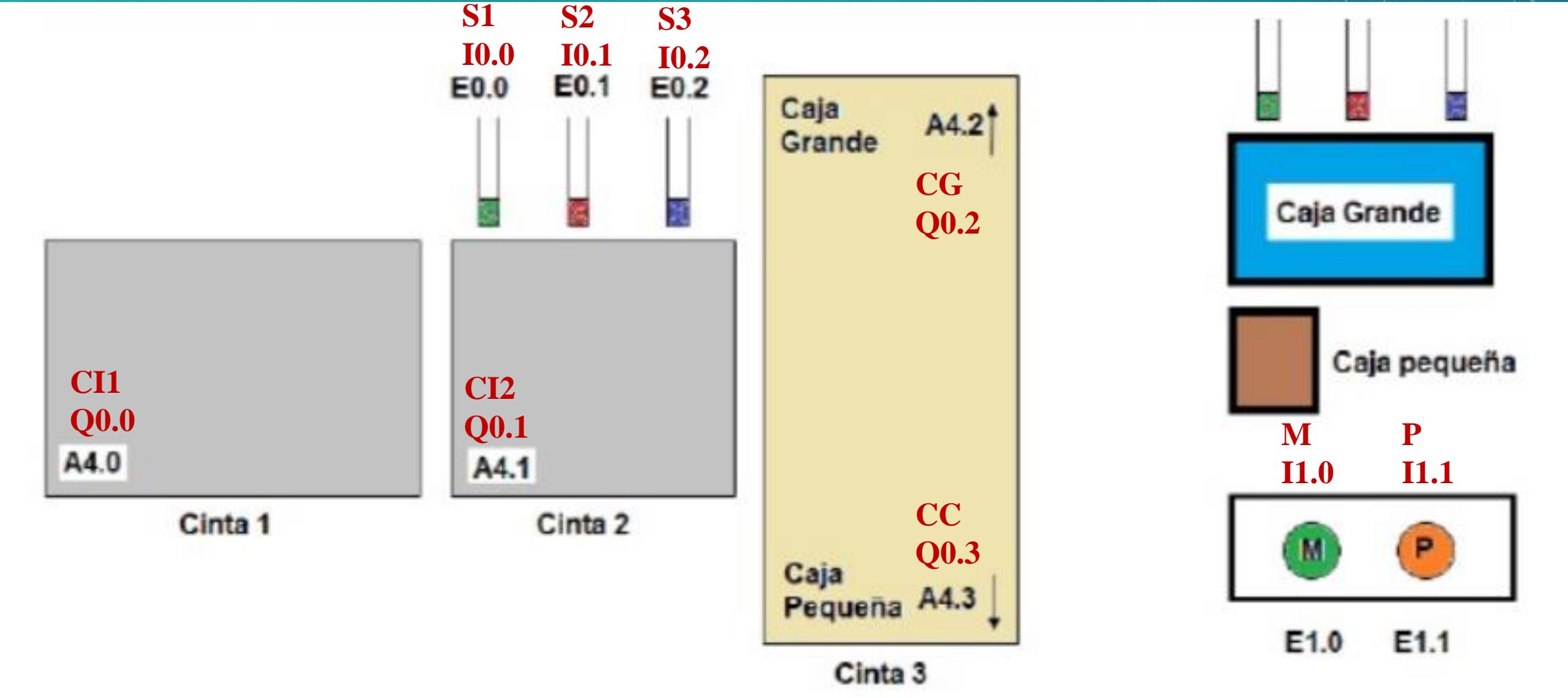
EJERCICIO 5



EJEMPLO: CONTROL DE UN SISTEMA DE CINTAS TRANSPORTADORAS

- Tenemos tres cintas transportadoras dispuestas como indica la siguiente figura. Por las cintas transportadoras van a circular cajas grandes y pequeñas indistintamente. El tamaño de las cajas es detectado por tres sensores. Para cajas grandes los tres sensores se activan. Para las pequeñas sólo el primero de ellos.

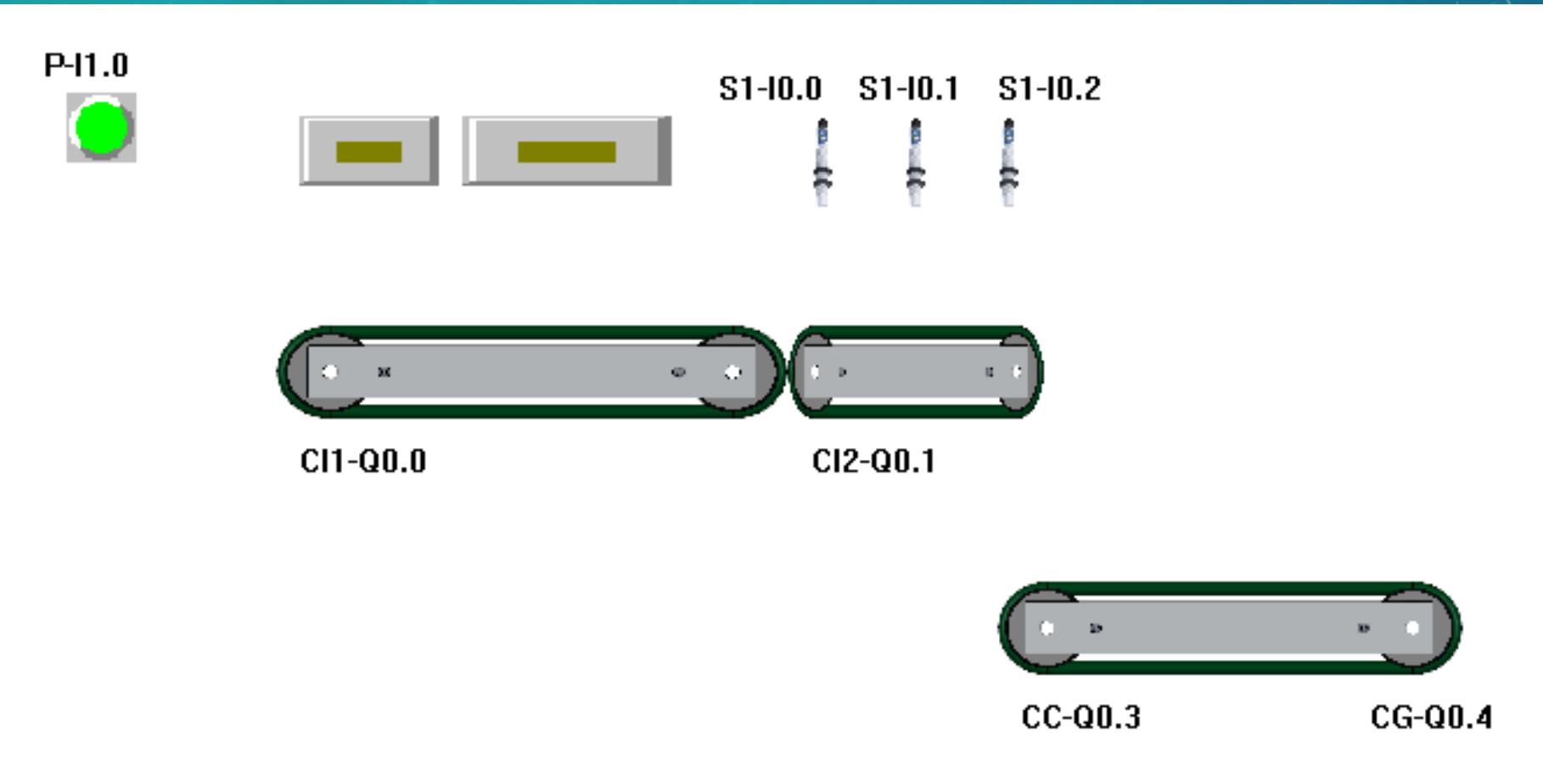
Ejemplo: Control de un Sistema de Cintas transportadoras



EJEMPLO: CONTROL DE UN SISTEMA DE CINTAS TRANSPORTADORAS

- Cuando presionemos el pulsador se pondrá en marcha la cinta nº 1. Cuando llegue la primera caja a la cinta nº 2, se detendrá la cinta nº 1 y que se pondrá en marcha la cinta nº 2.
- En la cinta nº 2 detectamos si la caja es grande o pequeña. Si es grande, se pondrá en marcha la tercera cinta hacia arriba, y si es pequeña se pondrá en marcha la tercera cinta hacia abajo.
- La cinta nº 2 se detendrá cuando la caja ya no se encuentre la caja en la cintanº2. La cinta nº 3 se detendrá a los 10 segundos de haberse puesto en marcha. A continuación se podrá poner en marcha de nuevo la primera cinta y vuelve a comenzar el ciclo.

EJERCICIO 6: CONTROL DE CAJAS



VENTAJAS DE TRABAJAR CON SECUENCIAS

- La selección de mando en la secuencia y la activación de las salidas están separadas
 - Así, si una salida no sólo debe estar activa en las etapas 1, 2 y 3 sino también en la 7, esto sólo exige modificar el programa en un punto.
 - Los cambios en la sección de mando de la secuencia no afectan a la activación de las salidas.

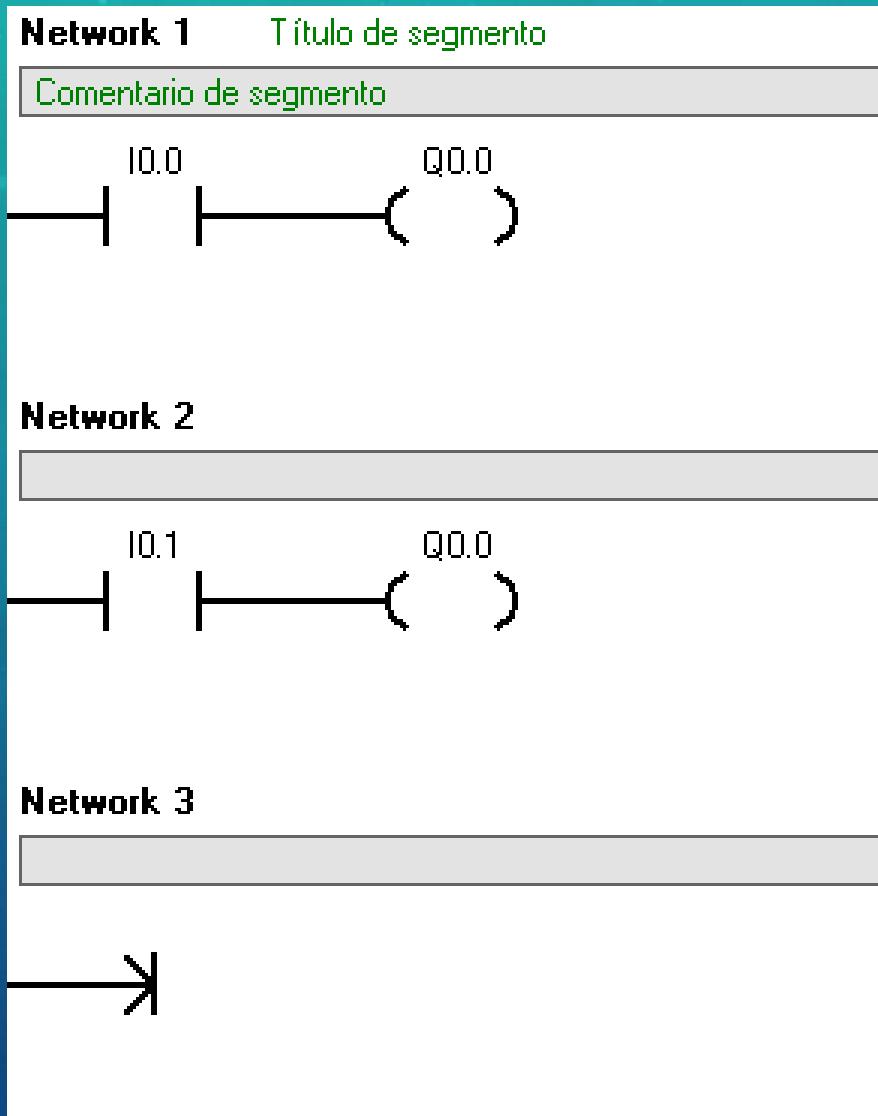
VENTAJAS DE TRABAJAR CON SECUENCIAS

- El programa puede probarse fácilmente
 - Cada etapa puede seguirse en la unidad de programación.
 - Si no avanza la secuencia puede detectarse fácilmente qué condición falta.

VENTAJAS DE TRABAJAR CON SECUENCIAS

- Se reducen los tiempos de parada de la máquina
 - Si no funciona la máquina, en base a la posición mecánica de la misma y de la marca de etapa activa es fácil detectar qué condición de transición falta.
- Menores errores de programación, puesta en marcha más rápida
 - El uso de una secuencia obliga a una estructura de programa que minimiza los errores de programación.

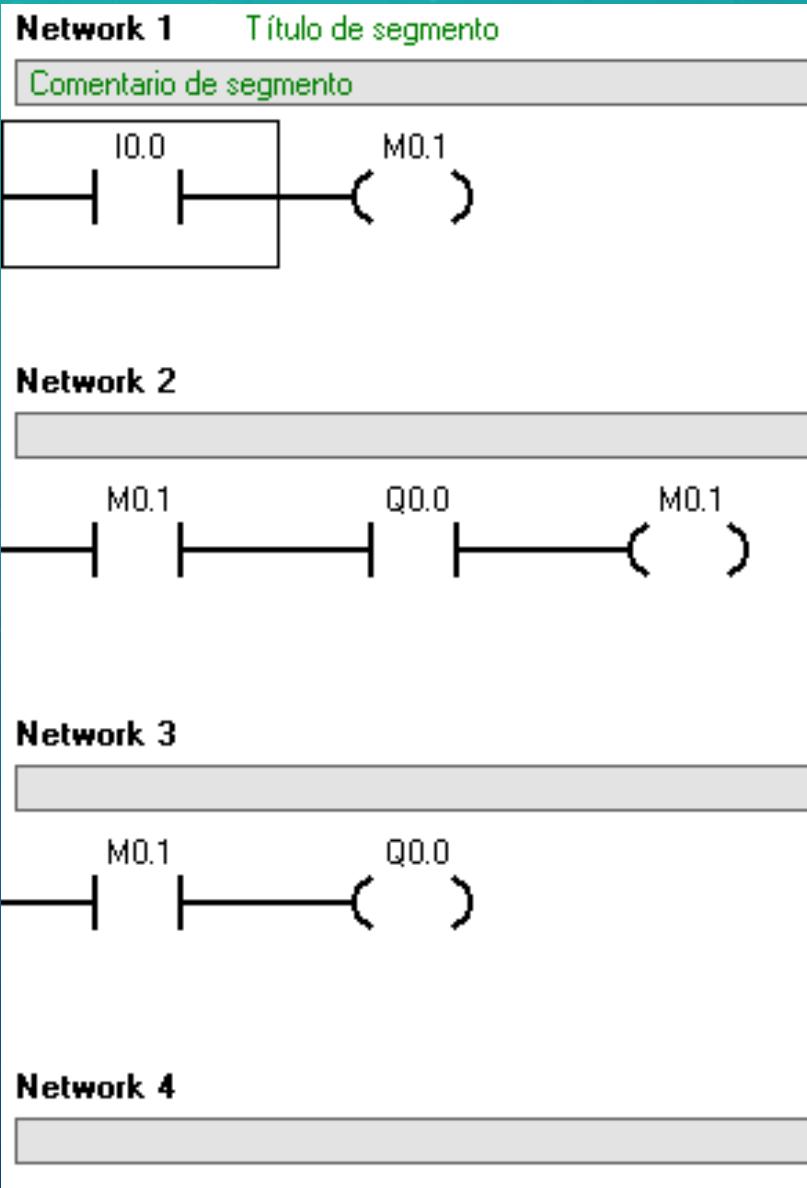
CICLO SCAN



¿Cuando se activa la salida Q0.0?

- a) Cuando se activa I0.0
- b) Cuando se activa I0.1
- c) Nunca

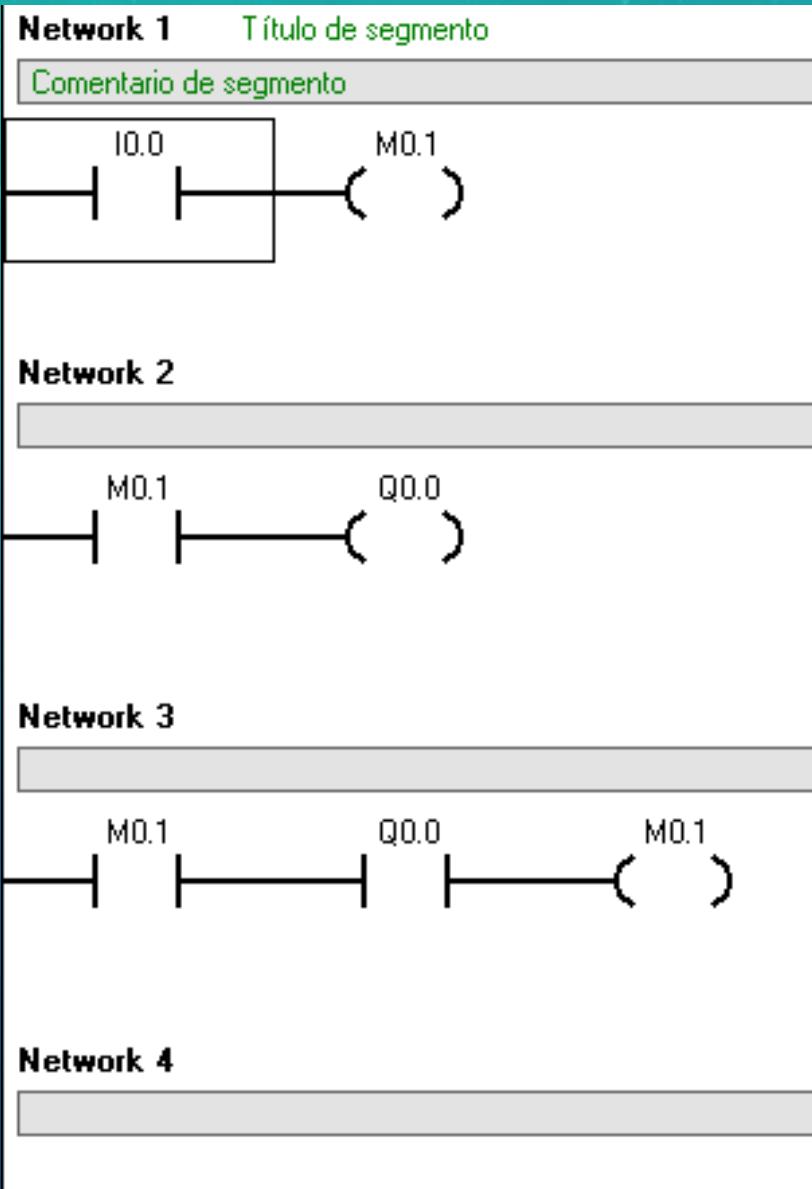
CICLO SCAN



¿Cuando se activa la salida Q0.0?

- a) Cuando se activa I0.0
- b) Nunca

CICLO SCAN

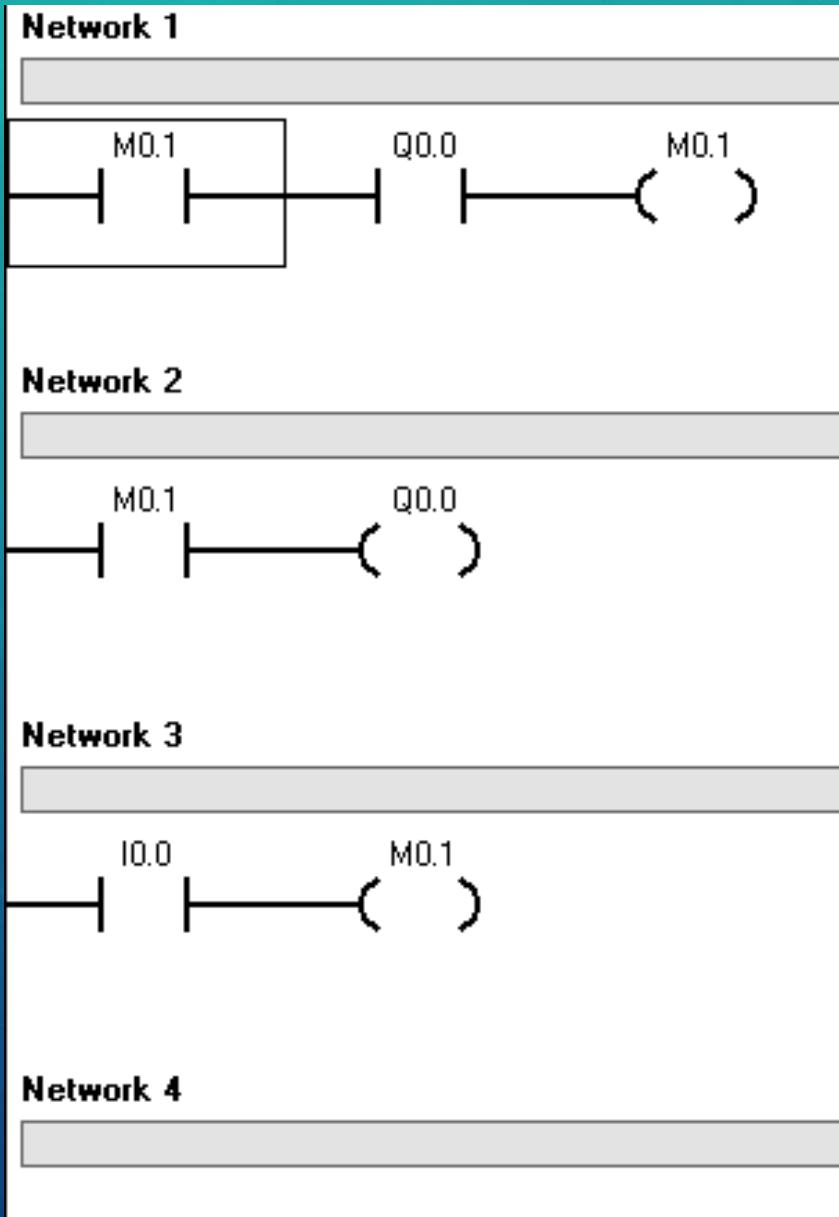


¿Cuando se activa la salida Q0.0?

- a) Cuando se activa I0.0
- b) Nunca

CICLO SCAN

c)



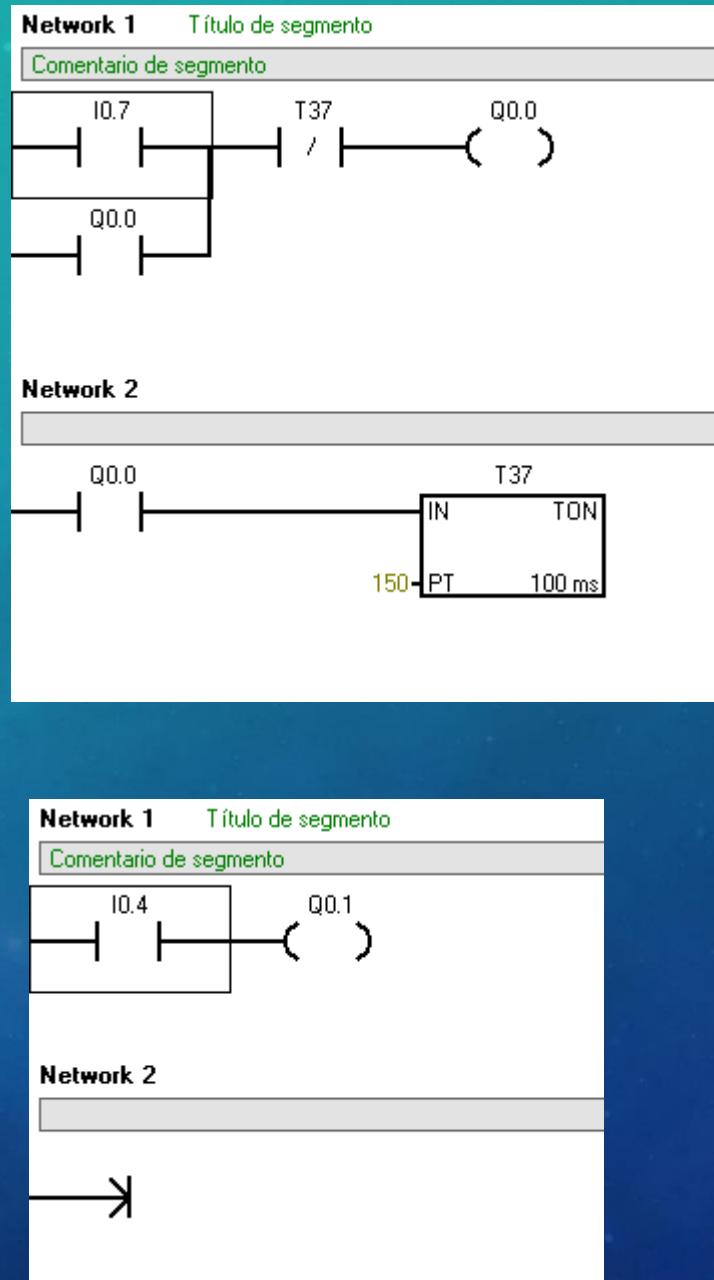
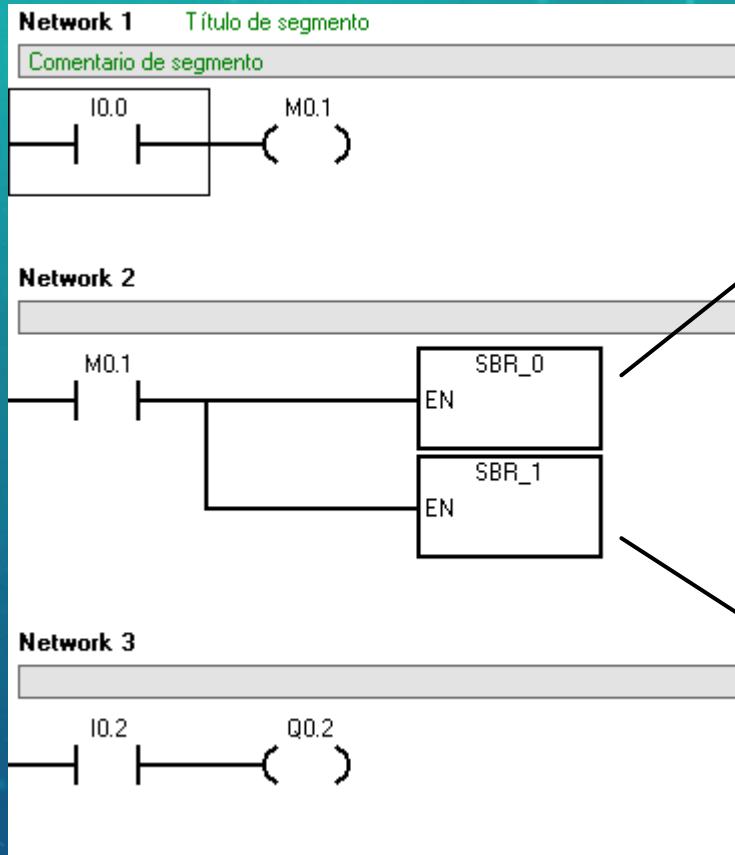
¿Cuando se activa la salida Q0.0?

- a) Cuando se activa I0.0
- b) Nunca

FUNCIONES

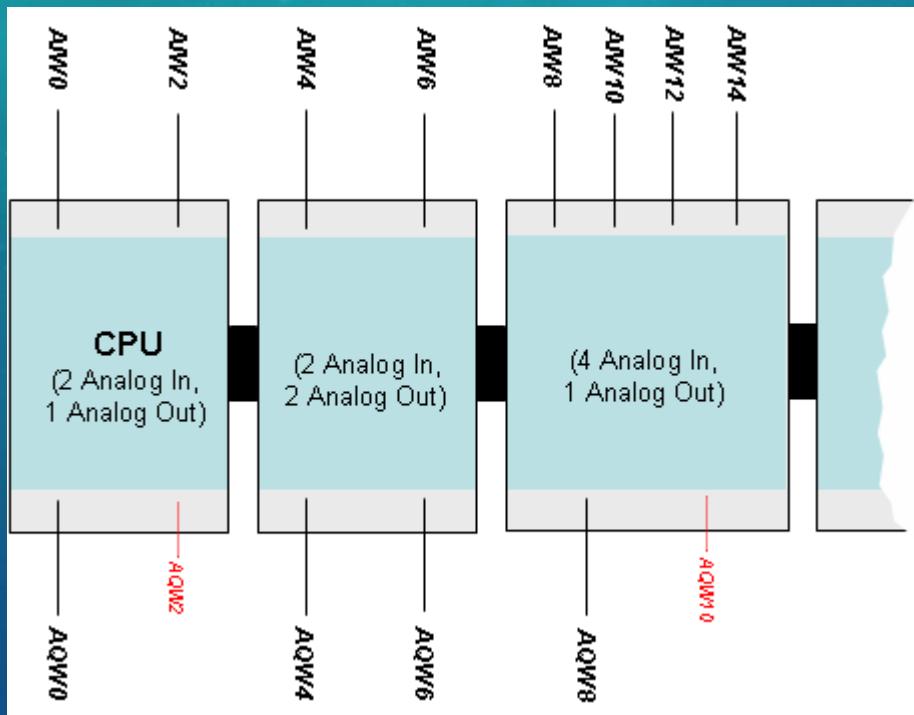
- SR
- JMP
- FOR

SUBRUTINAS



ADQUISICIÓN DE SEÑALES ANALÓGICAS

CONEXIÓN DE MÓDULOS DE ENTRADA Y SALIDAS ANALÓGICAS



ÁREAS DE MEMORIA DE LAS CPU'S S7-200

| Descripción | CPU 221 | CPU 222 | CPU 224 | CPU 224XP, CPU 224XPsi | CPU 226 |
|-------------------------------------|--------------|--------------|--------------|---------------------------|--------------|
| Imagen de proceso de las entradas | I0.0 a I15.7 | I0.0 a I15.7 | I0.0 a I15.7 | I0.0 a I15.7 | I0.0 a I15.7 |
| Imagen de proceso de las salidas | Q0.0 a Q15.7 | Q0.0 a Q15.7 | Q0.0 a Q15.7 | Q0.0 a Q15.7 | Q0.0 a Q15.7 |
| Entradas analógicas (sólo lectura) | AIW0 a AIW30 | AIW0 a AIW30 | AIW0 a AIW62 | AIW0 a AIW62 | AIW0 a AIW62 |
| Salidas analógicas (sólo escritura) | AQW0 a AQW30 | AQW0 a AQW30 | AQW0 a AQW62 | AQW0 a AQW62 | AQW0 a AQW62 |

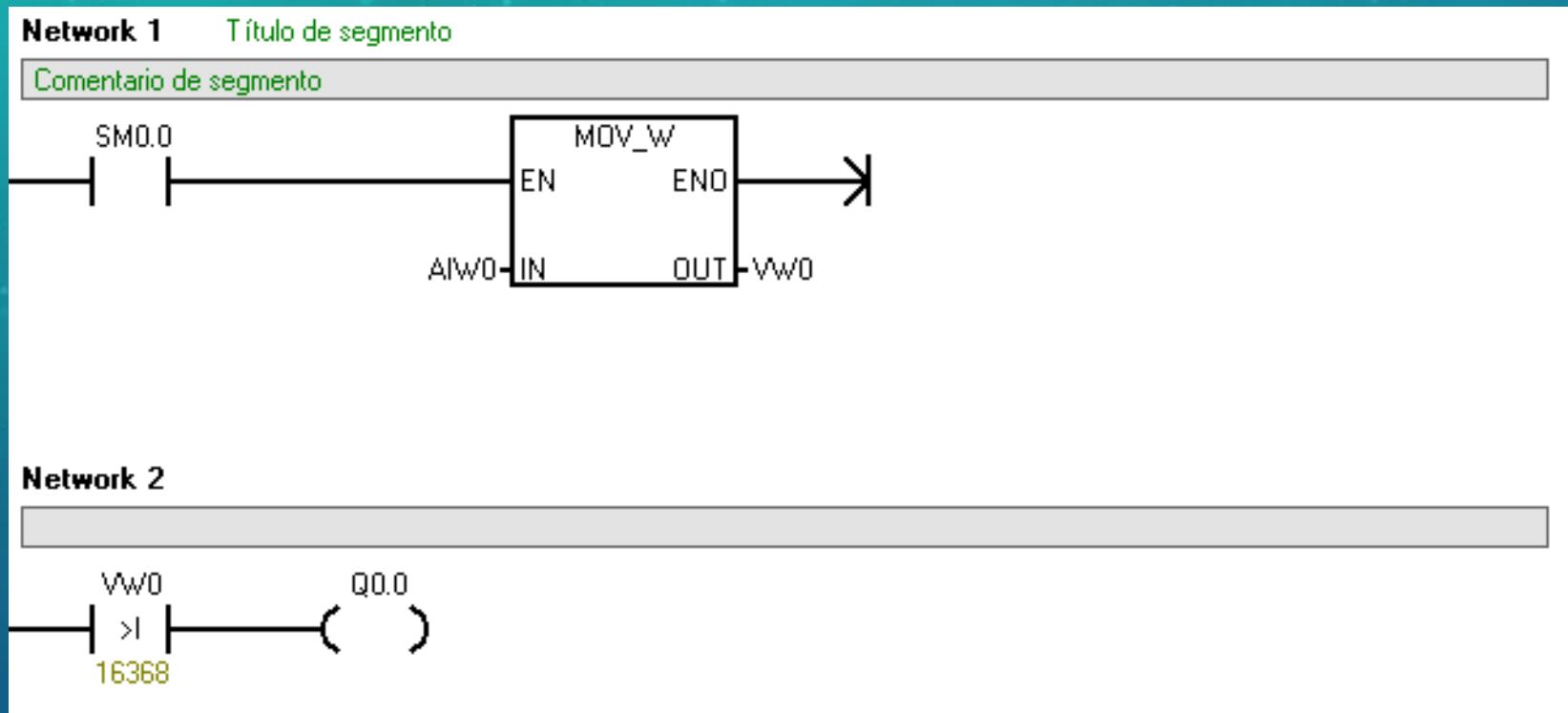
ENTRADAS ANALÓGICAS: AI

- El S7-200 convierte valores reales analógicos (p. ej. temperatura, tensión, etc.) en valores digitales en formato de palabra (de 16 bits). A estos valores se accede con un identificador de área (AI), seguido del tamaño de los datos (W) y de la dirección del byte inicial. Puesto que las entradas analógicas son palabras que comienzan siempre en bytes pares (p. ej. 0, 2, 4, etc.), es preciso utilizar direcciones con bytes pares (p. ej. AIW0, AIW2, AIW4, etc.) para acceder a las mismas. Las entradas analógicas son valores de sólo lectura.

SALIDAS ANALÓGICAS: AQ

- El S7-200 convierte valores digitales en formato de palabra (de 16 bits) en valores reales analógicos (p. ej. intensidad o tensión). Estos valores analógicos son proporcionales a los digitales. A los valores analógicos se accede con un identificador de área (AQ), seguido del tamaño de los datos (W) y de la dirección del byte inicial. Puesto que las salidas analógicas son palabras que comienzan siempre en bytes pares (p. ej. 0, 2, 4, etc.), es preciso utilizar direcciones con bytes pares (p. ej. AQW0, AQW2, AQW4, etc.) para acceder a las mismas. Las salidas analógicas son valores de sólo escritura.

CÓDIGO DE EJEMPLO



- El valor de la señal de entrada analógica **AIW0** se transfiere al registro de propósito general **VW0** para posteriormente realizar una comparación y activar la salida cuando el valor de la señal de entrada sea mayor a **16368** (aprox. 5V)

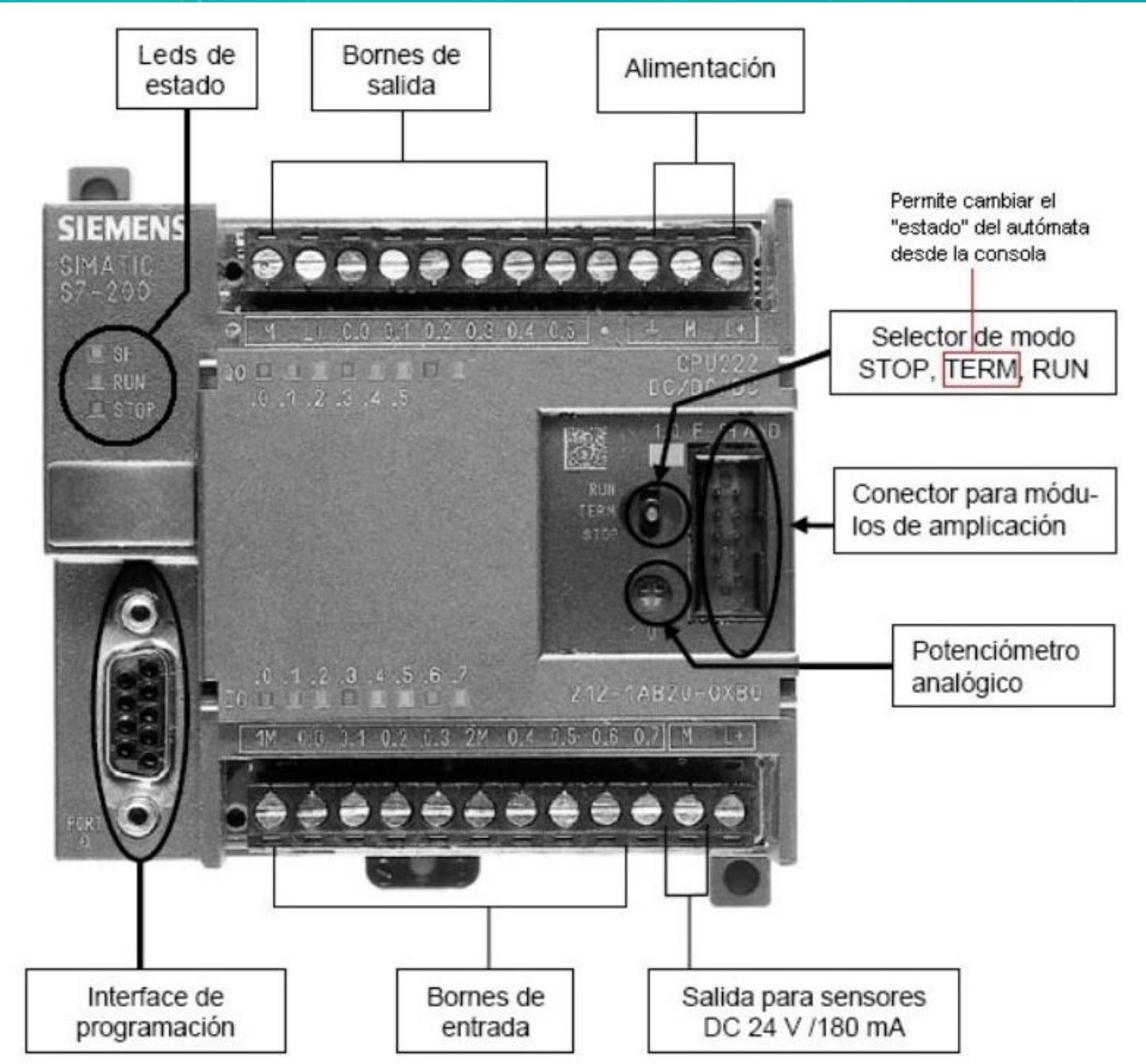
RANGO DE DATOS

Rangos de valores numéricos de constantes

| | Rango de enteros sin signo | | Rango de enteros con signo | |
|-------------------|-------------------------------|---------------|-------------------------------|-----------------------|
| Formato: | Decimal: | Hexadecimal: | Decimal: | Hexadecimal: |
| B (byte) | 0 a 255 | 0 a FF | -128 a +127 | 80 a 7F |
| W (palabra) | 0 a 65535 | 0 a FFFF | -32768 a +32767 | 8000 a 7FFF |
| D (palabra doble) | 0 a 4294967295 | 0 a FFFF FFFF | -2147483648 a +2147483647 | 8000 0000 a 7FFF FFFF |
| Formato: | Decimal real (rango positivo) | | Decimal real (rango negativo) | |
| D (palabra doble) | +1,175495E-38 a +3,402823E+38 | | -1,175495E-38 a -3,402823E+38 | |

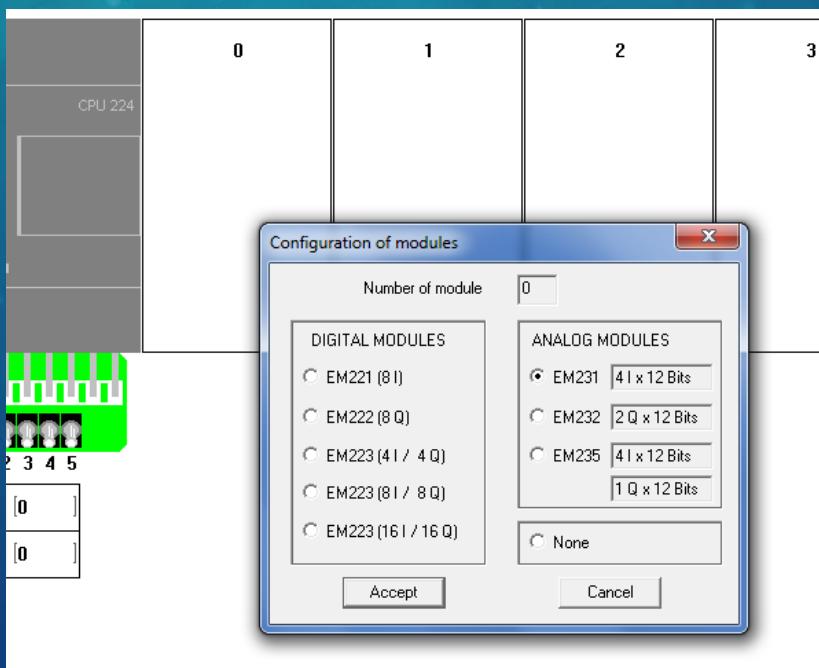
SMB28 A SMB29 POTENCIÓMETROS ANALÓGICOS S7200

- Los bytes de marcas especiales 28 y 29 contienen los valores digitales correspondientes a la posición de los potenciómetros analógicos 0 y 1, respectivamente.
- Estos potenciómetros están ubicados detrás de la tapa frontal de la CPU. Utilice un destornillador pequeño para ajustar los potenciómetros (gire hacia la derecha para incrementar el valor, o bien hacia la izquierda para decrementarlo).
- El programa puede utilizar estos valores de sólo lectura para diversas funciones, por ejemplo, para actualizar el valor actual de un temporizador o de un contador, para introducir o modificar los valores estándar, o bien, para configurar límites. Los potenciómetros analógicos tienen un rango nominal comprendido entre 0 y 255.

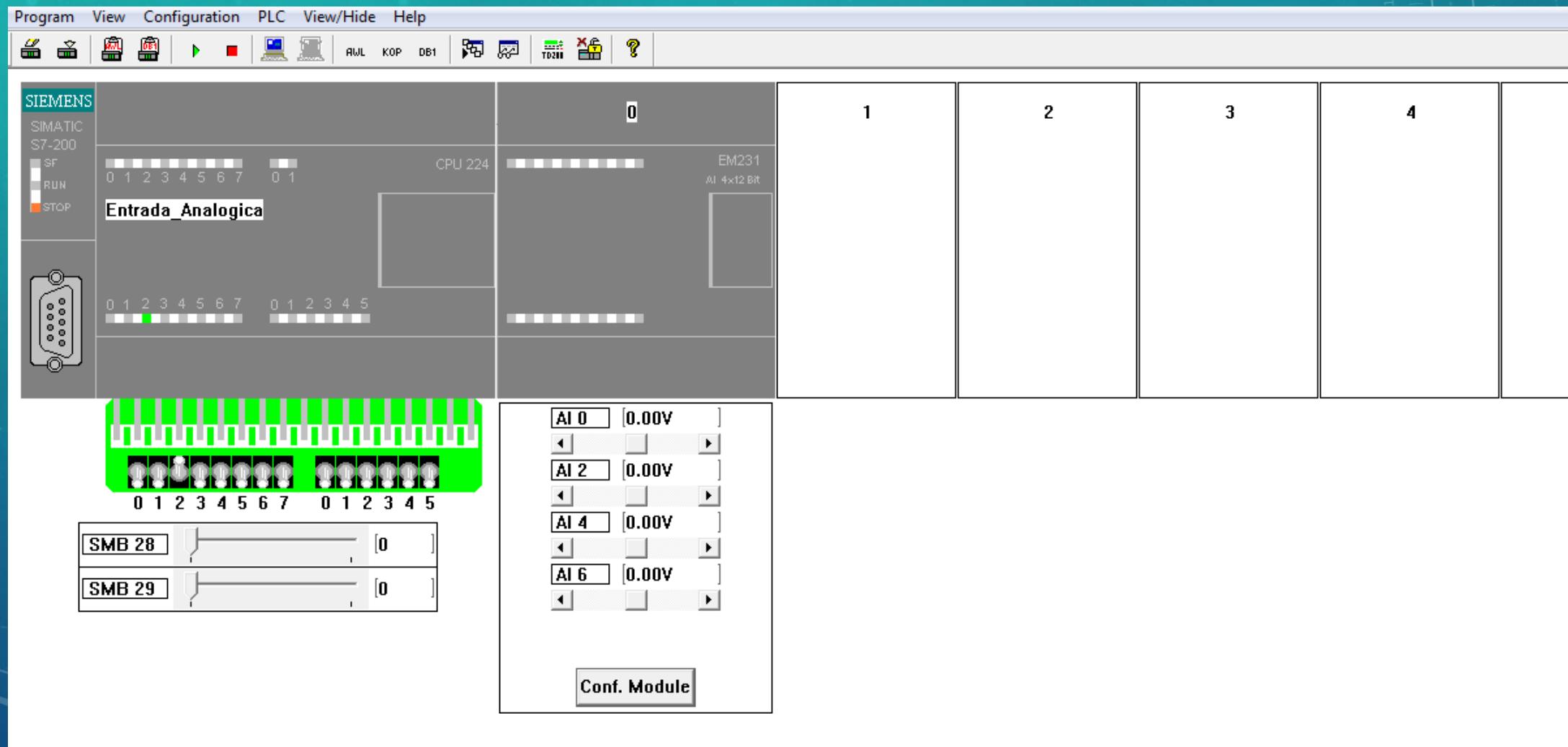


SIMULADOR DE PLC

Para poder realizar la simulación de una entrada o salida analógica, se debe agregar el modulo correspondiente para poder modificar los valores de entrada en la ejecución del programa. Solo se debe dar doble clic sobre algún modulo y aparecerán los diferentes tipos que se pueden conectar al PLC.



SIMULADOR DE PLC

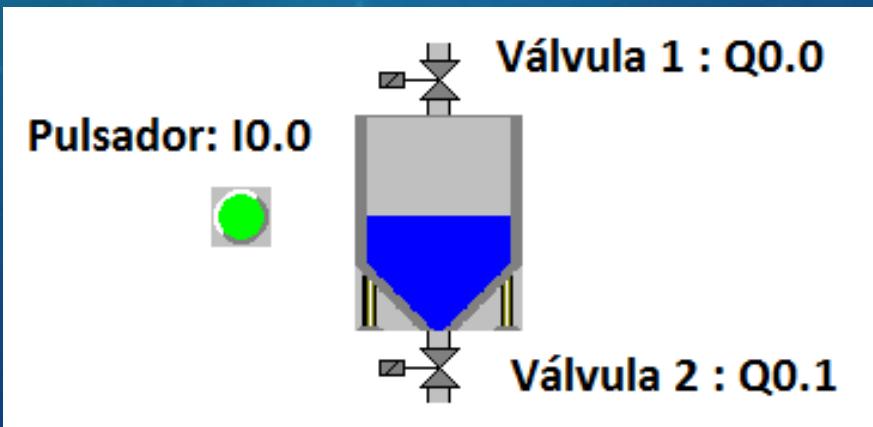


EJERCICIO 7

Condiciones iniciales: las válvulas deben estar cerradas

Al presionar el pulsador se deberá abrir la válvula 1 para empezar a llenar el deposito, al llegar a un nivel de llenado de un 75% se debe cerrar la válvula 1 y se debe abrir la válvula 2 para empezar a vaciar el deposito, cuando el nivel llegue al 25% se debe cerrar la válvula 2 y se debe abrir la válvula 1 para volver a llenar hasta el nivel de 75% y así volver a repetir la secuencia de vaciado y llenado indefinidamente.

Nota: El deposito cuenta con un sensor analógico que proporciona el nivel de llenado, calcular los valores a 75% y 25% para hacer las comparaciones.

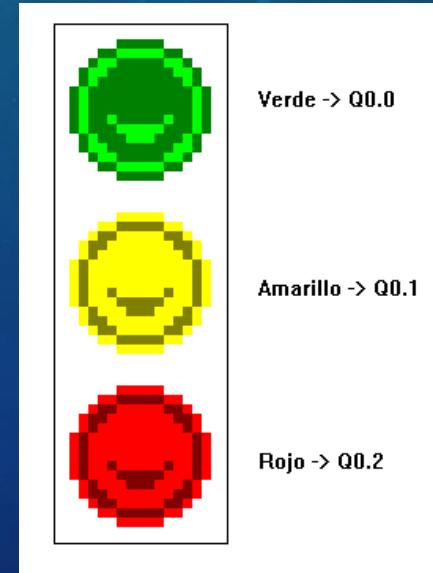


EJERCICIO 8:

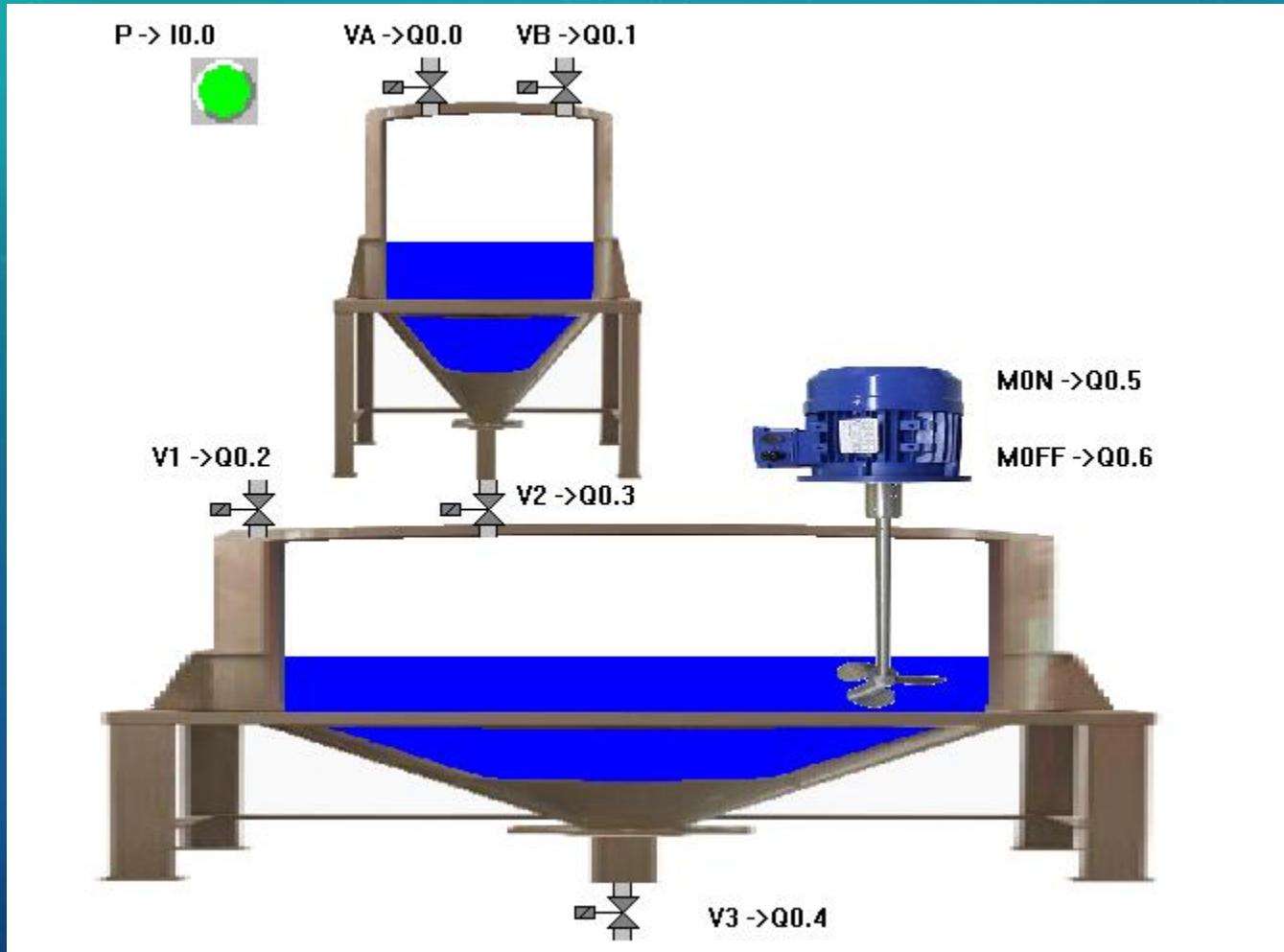
- Realizar un programa que prenda una lampara de manera intermitente y que la velocidad de transición de prendido y apagado se configure por medio de los potenciómetros internos del PLC

EJERCICIO 9: CONTROL DE UN SEMÁFORO

- Tenemos un semáforo con las tres luces verde, amarillo y rojo. El ciclo empieza en verde.
- Verde (5 seg).
- Verde y Amarillo (2 seg).
- Rojo (6 seg)



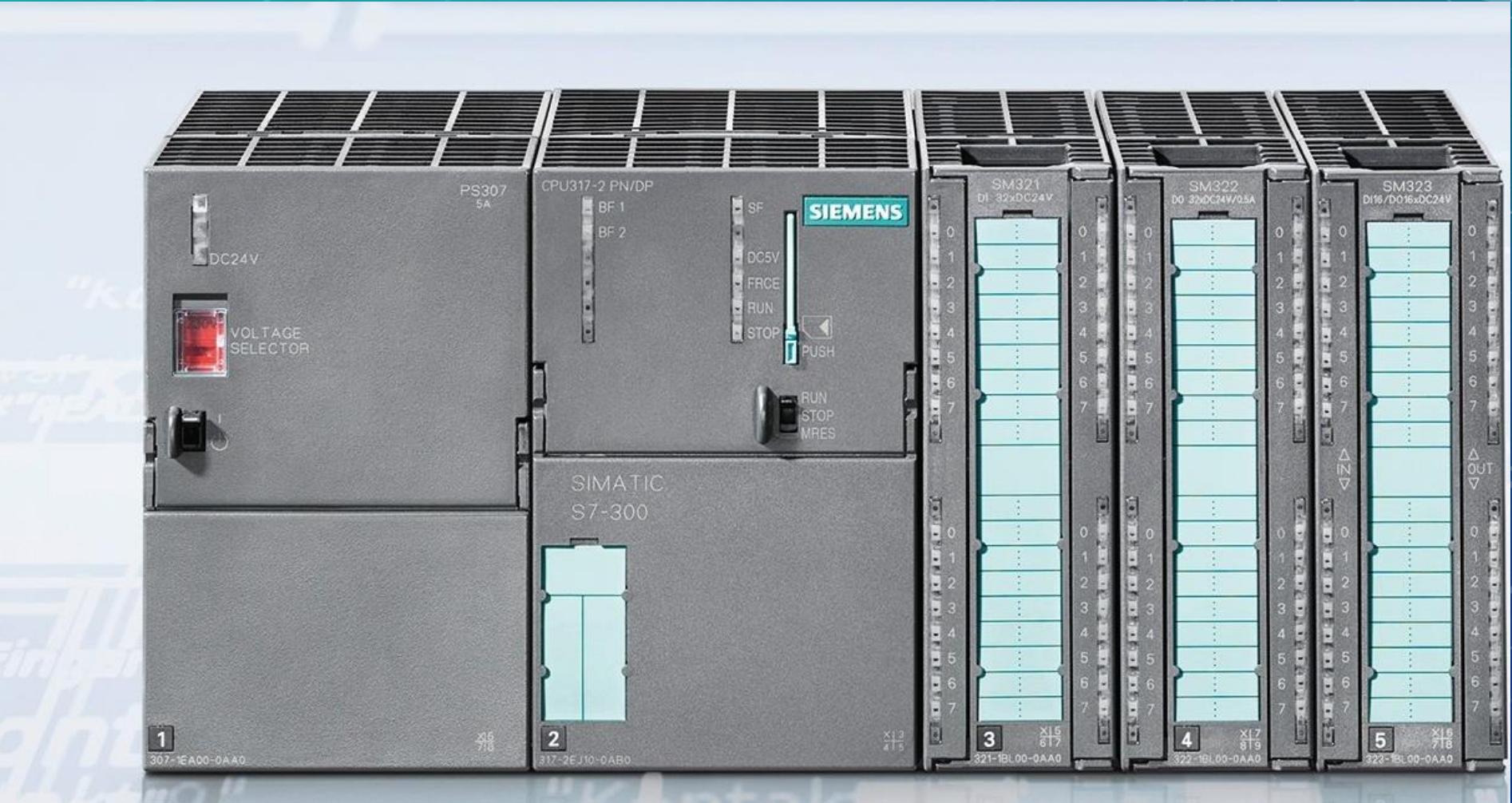
EJERCICIO 10: MEZCLADORA - SEÑALES ANALÓGICAS



EJERCICIO 9: MEZCLADORA SEÑALES ANALÓGICAS

- Se requiere mezclar 2 productos. La dosificación de los dos productos se realiza en un deposito (1), primero se vierte el producto A sobre la tolva activando la Válvula VA hasta que se alcanza un nivel de 30% y a continuación se añade el producto B Activando la válvula VB para conseguir el nivel total del 60%, al mismo tiempo se llenara de agua el deposito (2) abriendo V1 hasta alcanzar un nivel de 20%, terminado estos dos procesos se abre V2 hasta vaciar el deposito (1) sobre el contenedor (2) que contiene el agua. Posteriormente se enciende el motor para mezclar los 2 productos con el agua y después de 5s se detendrá el motor y se vaciara el depósito (2) activando V3 para poder iniciar un nuevo ciclo. El proceso se activa con un interruptor P.

PLC S7 300

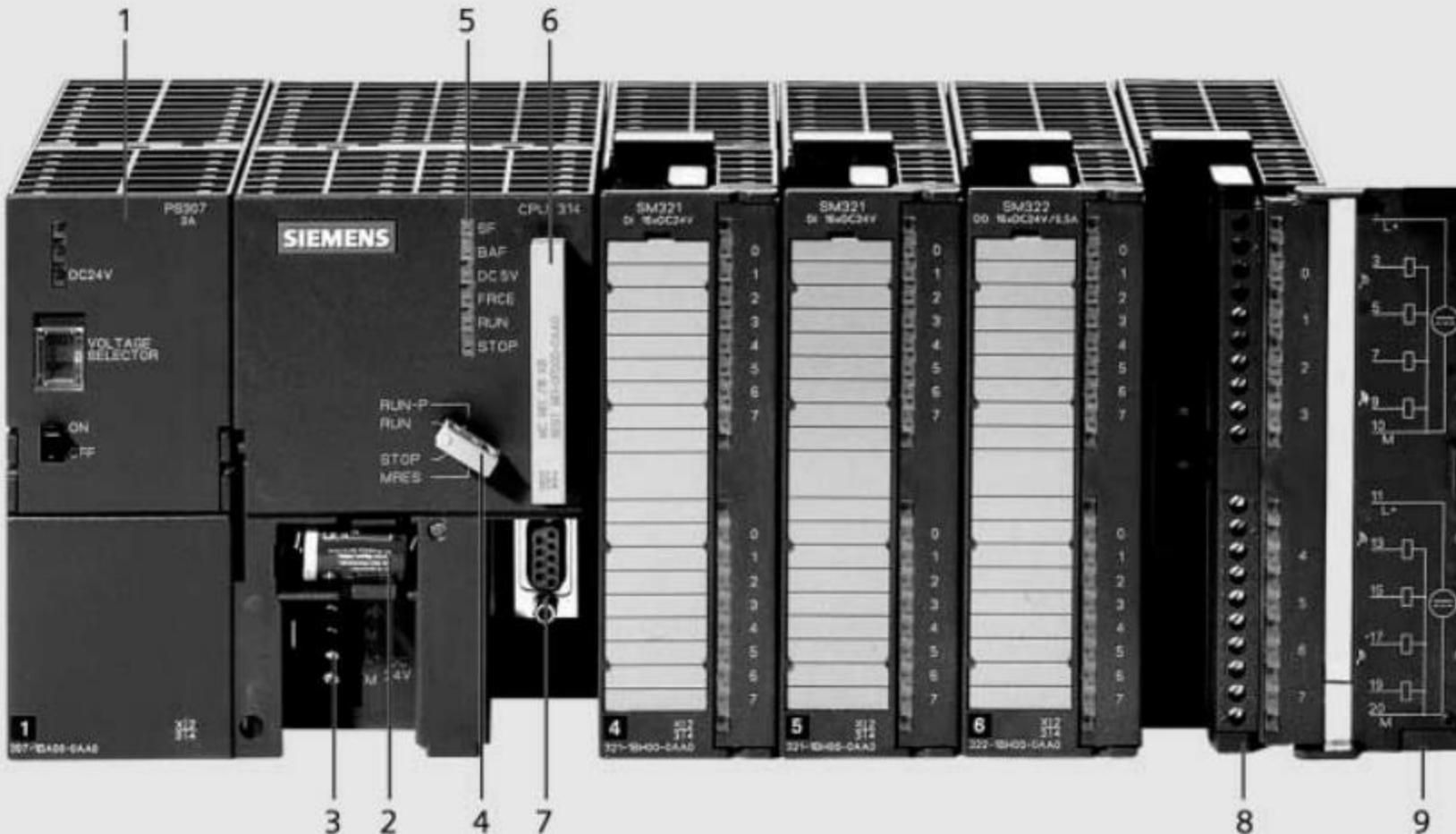


CONEXIÓN DE MODULOS



PC ADAPTER USB





1: Fuente de alimentación de carga (opción)
2: Pila tampón (a partir de CPU 313)
3: Conexión para 24 V DC
4: Selector de modo (con llave)
5: LED para señalizar estados y fallos

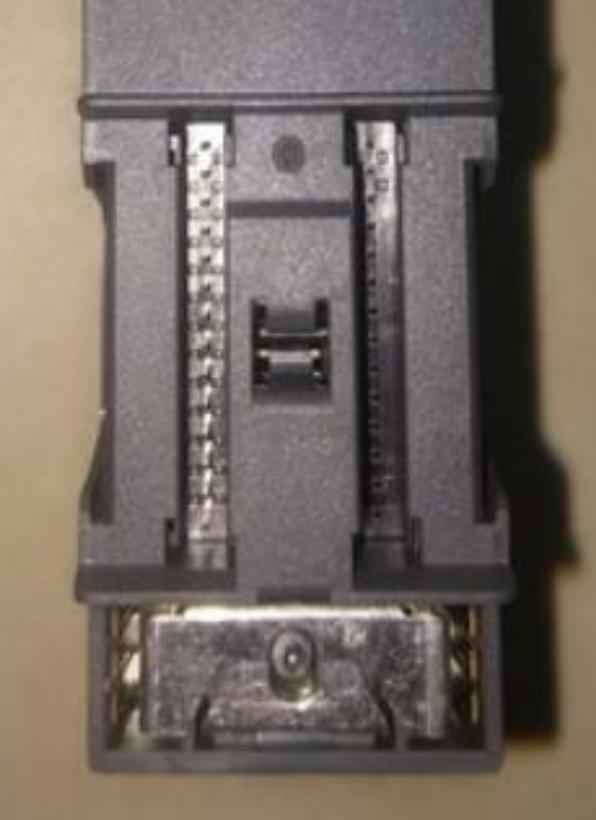
6: Memory Card (a partir de CPU 313)
7: MPI (interface multipunto)
8: Conector frontal
9: Puerta frontal

PS 307

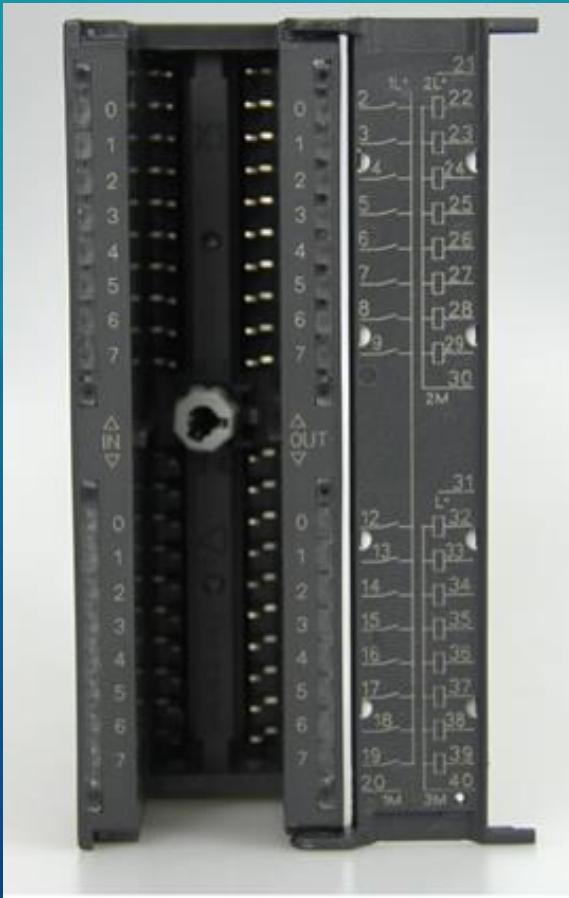
307-1EA01-0AA0

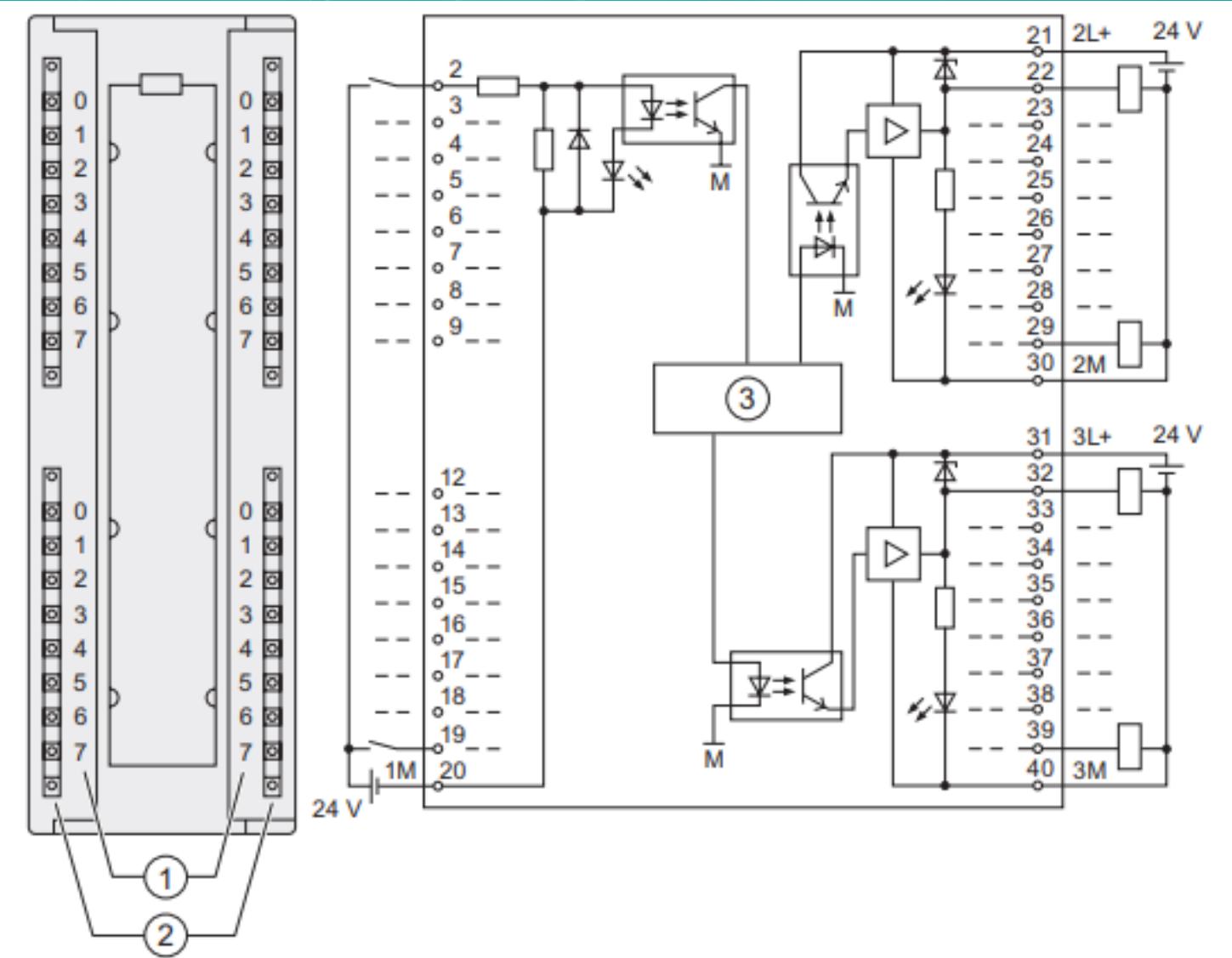


CPU 315F-2 PN/DP
6ES7 315-2FJ14-0AB0



DI16/DO16xDC24V
6ES7 323-1BL00-0AA0

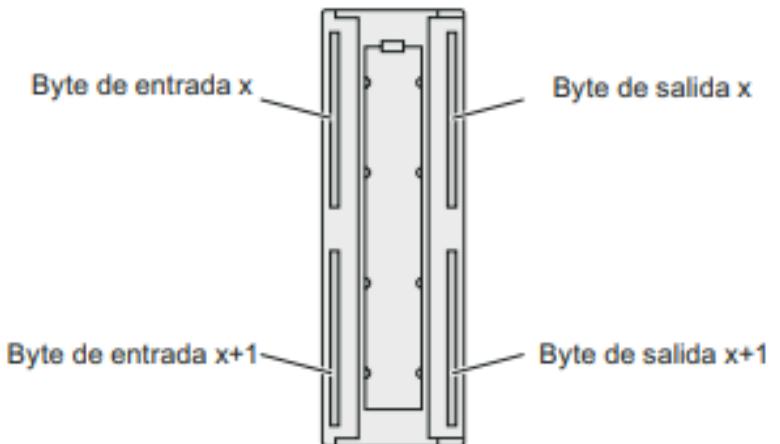




- ① Número de canal
- ② Indicación del estado – verde
- ③ Interfaz al bus posterior

Asignación de conexiones

La figura siguiente muestra la asignación de los canales respecto a las direcciones de entrada y salida.



| Inglés | Alemán | Explicación: | Tipo de datos: | Direcciones: |
|--------|--------|------------------------------|------------------------------|--------------|
| I | E | Bit de entrada | BOOL | 0..65535.7 |
| IB | EB | Byte de entrada | BYTE, CHAR | 0..65535 |
| IW | EW | Palabra de entrada | WORD, INT, S5TIME, DATE | 0..65534 |
| ID | ED | Palabra doble de entrada | DWORD, DINT, REAL, TOD, TIME | 0..65532 |
| Q | A | Bit de salida | BOOL | 0..65535.7 |
| QB | AB | Byte de salida | BYTE, CHAR | 0..65535 |
| QW | AW | Palabra de salida | WORD, INT, S5TIME, DATE | 0..65534 |
| QD | AD | Palabra doble de salida | DWORD, DINT, REAL, TOD, TIME | 0..65532 |
| M | M | Bit de marcas | BOOL | 0..65535.7 |
| MB | MB | Byte de marcas | BYTE, CHAR | 0..65535 |
| MW | MW | Palabra de marcas | WORD, INT, S5TIME, DATE | 0..65534 |
| MD | MD | Palabra doble de marcas | DWORD, DINT, REAL, TOD, TIME | 0..65532 |
| PIB | PEB | Byte de entrada de periferia | BYTE, CHAR | 0..65535 |

| | | | | |
|-----|-----|---------------------------------------|------------------------------|----------|
| PQW | PAW | Palabra de salida de periferia | WORD, INT, S5TIME, DATE | 0..65534 |
| PID | PED | Palabra doble de entrada de periferia | DWORD, DINT, REAL, TOD, TIME | 0..65532 |
| PQD | PAD | Palabra doble de salida de periferia | DWORD, DINT, REAL, TOD, TIME | 0..65532 |
| T | T | Temporizador | TIMER | 0..65535 |
| C | Z | Contadores | COUNTER | 0..65535 |
| FB | FB | Bloque de función | FB | 0..65535 |
| OB | OB | Bloque de organización | OB | 0..65535 |
| DB | DB | Bloque de datos | DB, FB, SFB, UDT | 0..65535 |
| FC | FC | Función | FC | 0..65535 |
| SFB | SFB | Bloque de función de sistema | SFB | 0..65535 |
| SFC | SFC | Función de sistema | SFC | 0..65535 |
| VAT | VAT | Tabla de variables | | 0..65535 |
| UDT | UDT | Tipo de datos de usuario | UDT | 0..65535 |

Resumen de los identificadores de bit, byte, palabra y palabra doble

| Área de memoria de la CPU | Identificador de operando | | | |
|---|---------------------------|---------|---------|-----------------|
| | Bit | Byte | Palabra | Doble palabra |
| Imagen del proceso de las entradas | E | EB | EW | ED |
| Imagen del proceso de las salidas | A | AB | AW | AD |
| Marcas | M | MB | MW | MD |
| Datos locales | L | LB | LW | LD |
| Área de periferia directa Entradas | | PEB | PEW | PED |
| Área de periferia directa Salidas | | PAB | PAW | PAD |
| Bloque de datos --(OPN) DB --(OPN) DI | | DBX DIX | DBB DIB | DBW DIW DBD DID |
| Temporizador (T) | T | | T | |
| Contador (Z) | Z | | Z | |

Indicación

Los dos operandos EB 0 y EB 1 (área de memoria E , formato: byte, bytes de dirección 0, 1) direccionan los dos mismos bytes como el operando EW 0 (área de memoria E, formato: palabra , 2 bytes, incremento del byte 0 a los bytes de dirección 0, 1).

PROGRAMAS DE UNA CPU

- En una CPU se ejecutan principalmente dos programas diferentes:
 - el sistema operativo
 - el programa de usuario.

SISTEMA OPERATIVO

El sistema operativo, que está integrado en las CPUs, organiza todas las funciones y procesos de la CPU que no están ligados a una tarea de control específica. Sus funciones son:

- gestionar el rearranque completo (en caliente) y el rearranque normal
- actualizar la imagen de proceso de las entradas y emitir la imagen de proceso de las salidas
- llamar el programa de usuario
- detectar las alarmas y llamar los OBs de tratamiento de alarmas
- detectar y tratar los errores
- administrar las áreas de memoria
- comunicar con unidades de programación y otras estaciones de comunicación

Modificando los parámetros del sistema operativo (preajustes) se puede controlar el comportamiento de la CPU en áreas determinadas.

PROGRAMA DE USUARIO

El programa de usuario primero se ha de crear y luego se ha de cargar en la CPU. Contiene todas las funciones requeridas para procesar la tarea específica de automatización. Las tareas del programa de usuario son:

- definir las condiciones del rearranque completo (en caliente) y del rearranque normal de la CPU (p.ej. preestablecer un valor determinado para las señales)
- tratar datos del proceso (p.ej. efectuar combinaciones lógicas de señales binarias, leer y evaluar valores analógicos, definir señales binarias de salida, emitir valores analógicos)
- reaccionar a alarmas
- tratamiento de perturbaciones en el desarrollo normal del programa.

BLOQUES DEL PROGRAMA DE USUARIO

El software de programación STEP 7 permite estructurar el programa de usuario, es decir, subdividirlo en distintas partes. Esto aporta las siguientes ventajas:

- los programas de gran tamaño se pueden programar de forma clara
- se pueden estandarizar determinadas partes del programa
- se simplifica la organización del programa
- las modificaciones del programa pueden ejecutarse más fácilmente
- se simplifica el test del programa, ya que puede ejecutarse por partes
- se simplifica la puesta en marcha.

TIPOS DE BLOQUES

| Bloque | Descripción breve de la función |
|---|---|
| Bloques de organización (OB) | Los OBs definen la estructura del programa de usuario. |
| Bloques de función del sistema (SFBs) y funciones de sistema (SFCs) | Los SFBs y SFCs están integrados en la CPU S7, permitiéndole acceder a importantes funciones del sistema. |
| Bloques de función (FB) | Los FBs son bloques con "memoria" que puede programar el mismo usuario. |
| Funciones (FC) | Las FCs contienen rutinas de programa para funciones frecuentes. |
| Bloques de datos de instancia (DBs de instancia) | Al llamar a un FB/SFB, los DBs de instancia se asocian al bloque. Los DBs de instancia se generan automáticamente al efectuarse la compilación. |
| Bloques de datos (DB) | Los DBs son áreas de datos para almacenar los datos de usuario. Adicionalmente a los datos asociados a un determinado bloque de función, se pueden definir también datos globales a los que pueden acceder todos los bloques. |

BLOQUES DE ORGANIZACIÓN Y ESTRUCTURA DEL PROGRAMA

- Los bloques de organización (OB) constituyen el interface entre el sistema operativo y el programa de usuario. Son llamados por el sistema operativo y controlan el procesamiento cíclico y controlado por alarmas del programa, el comportamiento de arranque del sistema de automatización y el tratamiento de los errores. Programando los bloques de organización se define el comportamiento de la CPU.

EJEMPLO OB

| Tipo de alarma | Bloque de organización | Prioridad (predeterminada) | Consulte también: |
|--------------------|---|---|--|
| Ciclo libre | OB 1 | 1 | "Bloque de organización para la ejecución cíclica del programa (OB 1)" |
| Alarmas horarias | OB 10 a OB 17 | 2 | "Bloques de organización de alarma horaria (OB 10 a OB 17)" |
| Alarmas de retardo | OB 20 OB 21 OB 22 OB 23 | 3 4 5 6 | "Bloques de organización de alarma de retardo (OB 20 a OB 23)" |
| Alarmas cíclicas | OB 30 OB 31 OB 32 OB 33 OB 34 OB 35 OB 36 OB 37 OB 38 | 7 8 9 10 11 12 13 14 15 | "Bloques de organización de alarma cíclica (OB 30 a OB 38)" |

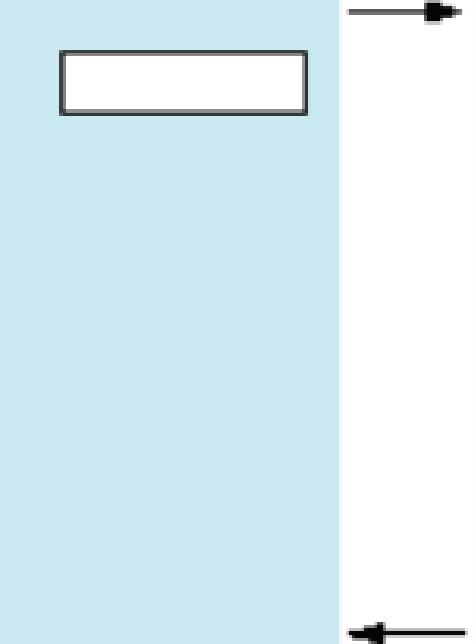
INFORMACIÓN DE ARRANQUE DE UN OB

- Cada bloque de organización contiene una información de arranque de 20 bytes de datos locales, suministrada por el sistema operativo durante el arranque de un OB. La información de arranque informa sobre el evento de arranque del OB, la fecha y hora de arranque de OB, así como errores ocurridos y eventos de diagnóstico.
- La información de arranque del OB 40 de alarma de proceso indica, por ejemplo, en la información de arranque la dirección del módulo que ha causado la alarma.

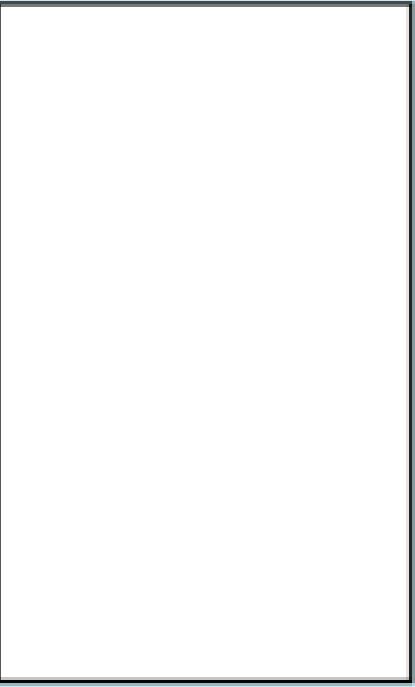
EJECUCIÓN CÍCLICA DE PROGRAMAS

- La ejecución cíclica de programas es la ejecución "normal" en PLCs, es decir, el sistema operativo se ejecuta en un bucle llamado ciclo. Cada vez que se recorre un ciclo, el sistema operativo llama al bloque de organización OB 1 en el programa principal. Por consiguiente, el programa de usuario se trata cíclicamente en el OB 1.

Sistema operativo



Programa de usuario



- La ejecución cíclica del programa puede ser interrumpida por determinados eventos de arranque (alarmas).
- El programa de usuario se puede dividir en subprogramas y repartir en diferentes bloques de organización. Si el programa de usuario debe reaccionar a una señal importante que se presente con poca frecuencia (p.ej., si el indicador de nivel de un depósito indica que se ha alcanzado el nivel de llenado), el subprograma que se deba ejecutar cuando se emita la señal se puede depositar en un OB que se ejecute de forma controlada por eventos.

PROGRAMACIÓN LINEAL O ESTRUCTURADA

Programación lineal

Programa
principal = OB 1



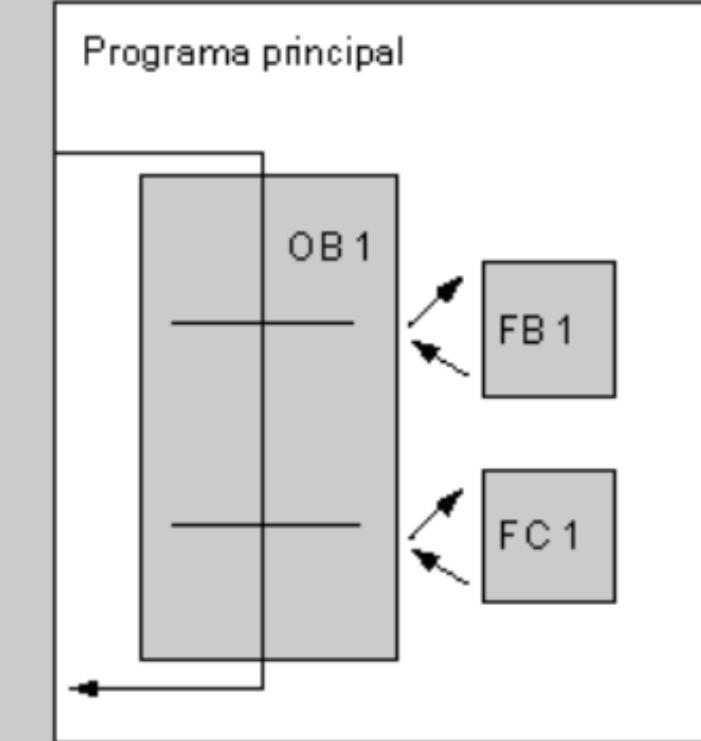
Programación estructurada

Programa principal

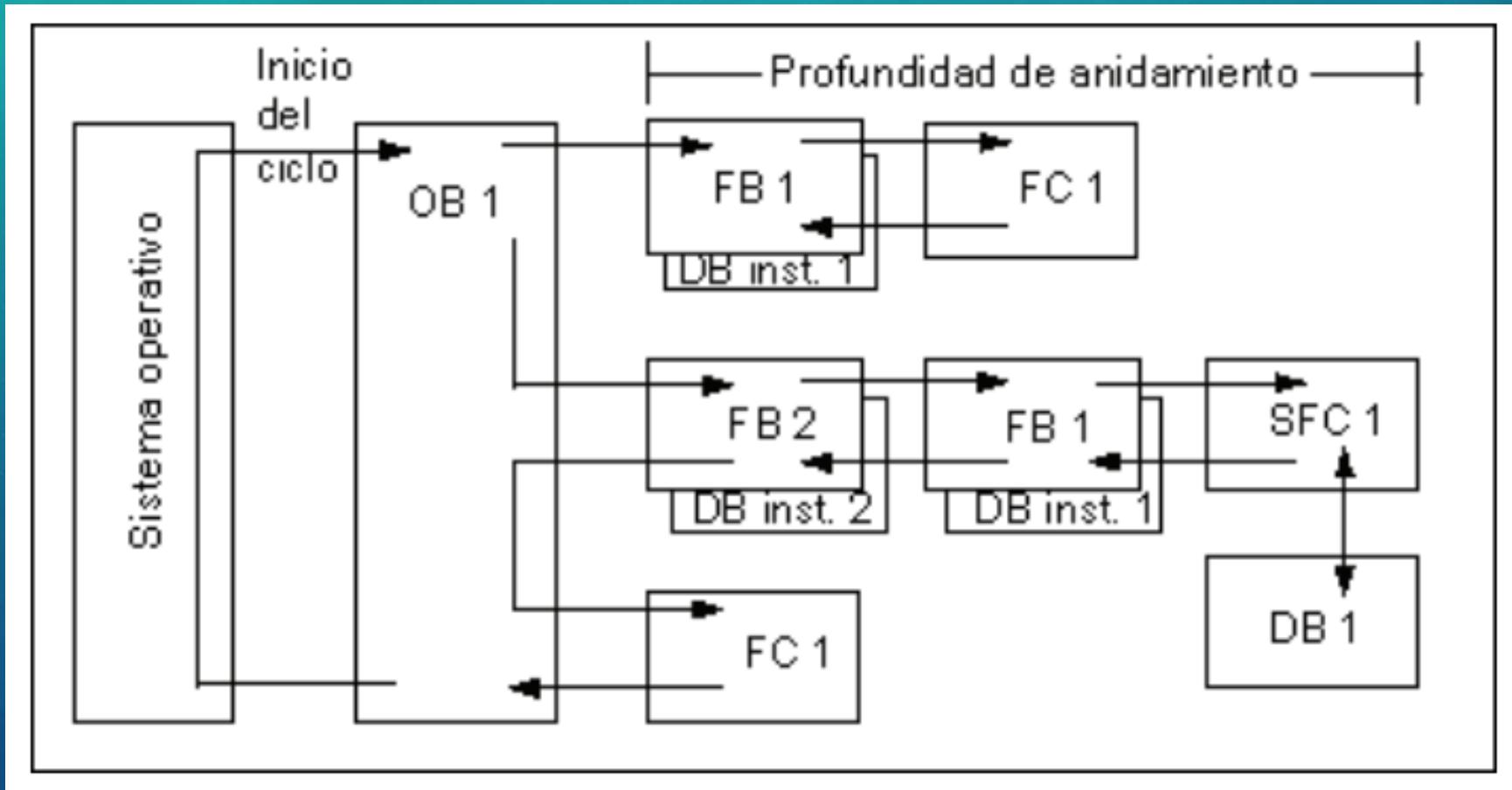
OB 1

FB 1

FC 1



SECUENCIA Y PROFUNDIDAD DE ANIDAMIENTO



EJECUCIÓN CÍCLICA DEL PROGRAMA

| Paso | Secuencia en las CPUs <small>ja partir de 10/98</small> |
|------|--|
| 1º | El sistema operativo inicia el tiempo de vigilancia del ciclo. |
| 2º | La CPU escribe los valores de la imagen de proceso de las salidas en los módulos de salida. |
| 3º | La CPU lee el estado de las entradas en los módulos de entradas y actualiza la imagen de proceso de las entradas. |
| 4º | La CPU ejecuta el programa de usuario y las operaciones indicadas en dicho programa. |
| 5º | Al final del ciclo, el sistema operativo realiza las tareas pendientes, p. ej. cargar y borrar bloques, recibir y enviar datos globales. |
| 6º | Finalmente, la CPU regresa al principio del ciclo y arranca nuevamente la vigilancia del tiempo de ciclo. |

FUNCIONES (FC)

- Las funciones son bloques programables. Una función es un bloque lógico "sin memoria". Las variables temporales de las FCs se memorizan en la pila de datos locales. Estos datos se pierden tras el tratamiento de las FCs. Para fines de memorización de datos, las funciones pueden utilizar bloques de datos globales. Como una FC no tiene asignada ninguna memoria, se han de indicar siempre parámetros actuales. A los datos locales de una FC no se pueden asignar valores iniciales.

CAMPO DE APLICACIÓN

La FC contiene un programa que se ejecuta cada vez que la FC es llamada por otro bloque lógico. Las funciones se pueden utilizar para

- devolver un valor de función al bloque invocante (ejemplo: funciones matemáticas)

BLOQUES DE FUNCIÓN (FB)

- Los bloques de función son bloques programables. Un FB es un bloque "con memoria". Dispone de un bloque de datos asignado como memoria (bloque de datos de instancia). Los parámetros que se transfieren al FB, así como las variables estáticas, se memorizan en el DB de instancia. Las variables temporales se memorizan en la pila de datos locales. Los datos memorizados en el DB de instancia no se pierden al concluir el tratamiento del FB. Los datos memorizados en la pila de datos locales se pierden al concluir el tratamiento del FB.

BLOQUES DE DATOS DE INSTANCIA

- A cada llamada de un bloque de función que transfiere parámetros está asignado un bloque de datos de instancia. En el DB de instancia están depositados los parámetros actuales y los datos estáticos del FB. Las variables declaradas en el FB definen la estructura del bloque de datos de instancia. La instancia define la llamada de un bloque de función. Si, por ejemplo, un bloque de función se llama cinco veces en el programa de usuario S7, existen cinco instancias de dicho bloque.

BLOQUES DE FUNCIÓN DE SISTEMA (SFB) Y FUNCIONES DEL SISTEMA (SFC)

Bloques preprogramados

- No es necesario programar cada función. Las CPUs S7 ofrecen bloques preprogramados que se pueden llamar desde el programa de usuario.

BLOQUES DE FUNCIONES DEL SISTEMA

- Un SFB es un bloque de funciones integrado en la CPU S7. Como los SFBs forman parte del sistema operativo, no se cargan como parte integrante del programa. Al igual que los FBs, los SFBs son bloques "con memoria". Para los SFBs se han de crear también bloques de datos de instancia y cargar en la CPU como parte integrante del programa. Las CPUs S7 ofrecen SFBs para la comunicación vía enlaces configurados para las funciones especiales integradas (p.ej. SFB 29 "HS_COUNT" en la CPU 312 IFM y en la CPU 314 IFM)

FUNCIONES DEL SISTEMA

- Una función del sistema es una función preprogramada integrada en la CPU S7. La SFC se puede llamar desde el programa. Como las SFCs forman parte del sistema operativo, no se cargan como parte integrante del programa. Al igual que las FCs, las SFCs son bloques "sin memoria".
- Las CPUs S7 ofrecen SFCs para:
- manipulación del reloj y del contador de horas de funcionamiento
- manipulación de eventos de errores síncronos, eventos de errores de alarma y asíncronos
- direccionamiento de módulos
- etc

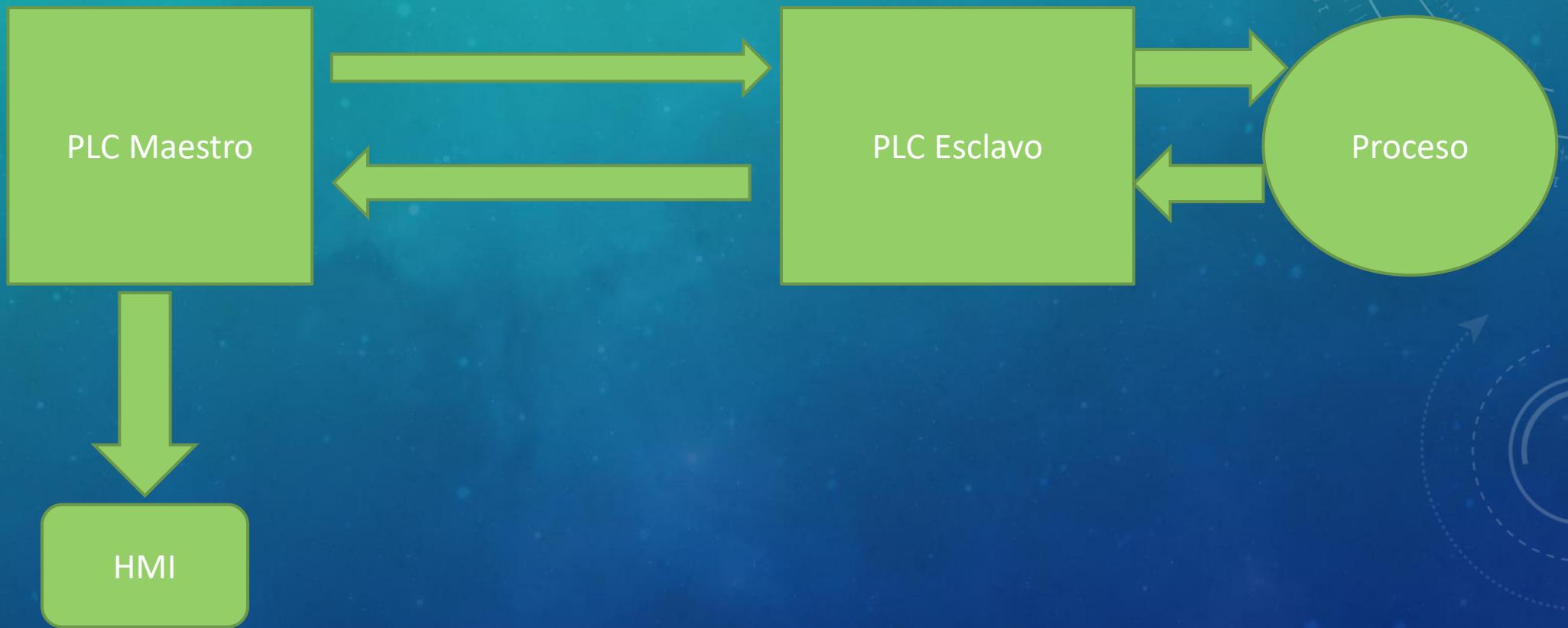
ACCIONES

| Operación | Identificador operando | Operando dirección | Significado |
|-----------|------------------------|--------------------|--|
| N[C] | A,E,M,D | m.n | Mientras esté activa la etapa [y se cumpla el interlock], la señal del operando será 1. |
| S[C] | A,E,M,D | m.n | Set (activar): tan pronto como se active la etapa [y se cumpla el interlock], el operando se pondrá a 1 y permanecerá así (con memoria) |
| R[C] | A,E,M,D | m.n | Reset (desactivar): tan pronto como se active la etapa [y se cumpla el interlock], el operando se pondrá a 0 y permanecerá así (con memoria) |
| D[C] | A,E,M,D | m.n | Retardo a la conexión (Delay): n seg. tras la activación de la etapa la señal del operando [cumpliéndose el interlock] será 1 mientras dure la activación de la etapa. Ello no rige cuando la activación de la etapa dura menos de n seg. (sin memoria). |
| | T#<const> | | Constante de tiempo |
| L[C] | A,E,M,D | m.n | Impulso limitado: si la etapa es activa [y se cumple el interlock], la señal del operando es 1 durante n segundos (sin memoria). |
| | T#<const> | | Constante de tiempo |
| CALL[C] | FB, FC, SFB, SFC | número del bloque | Llamada a bloque: cuando está activa la etapa [y se cumple el interlock], se llama al bloque indicado. |

[] = complemento opcional del interlock; m = dirección byte; n = dirección bit; área de direccionamiento: 0.0 65535.7

PROTOCOLOS DE COMUNICACIÓN





PROTOCOLOS DE COMUNICACIÓN SOPORTADOS POR LAS CPUS S7--300

- Interfaz multipunto (MPI)
- PROFIBUS
- PROFINET

INTERFAZ MULTIPUNTO (MPI)

- La interface multipunto MPI (Multi Point Interface) es un protocolo serie, propio de siemens integrado en cada autómata programable SIMATIC S7. Se puede utilizar para interconexiones en red sencillas, permitiendo conectar varios PG's o autómatas programables.

INTERFAZ MULTIPUNTO (MPI)

- El método de acceso es de bus token. La velocidad estándar es de 187.5 Kbaudios, aunque dependiendo de los participantes se puedes ajustar otras velocidades.

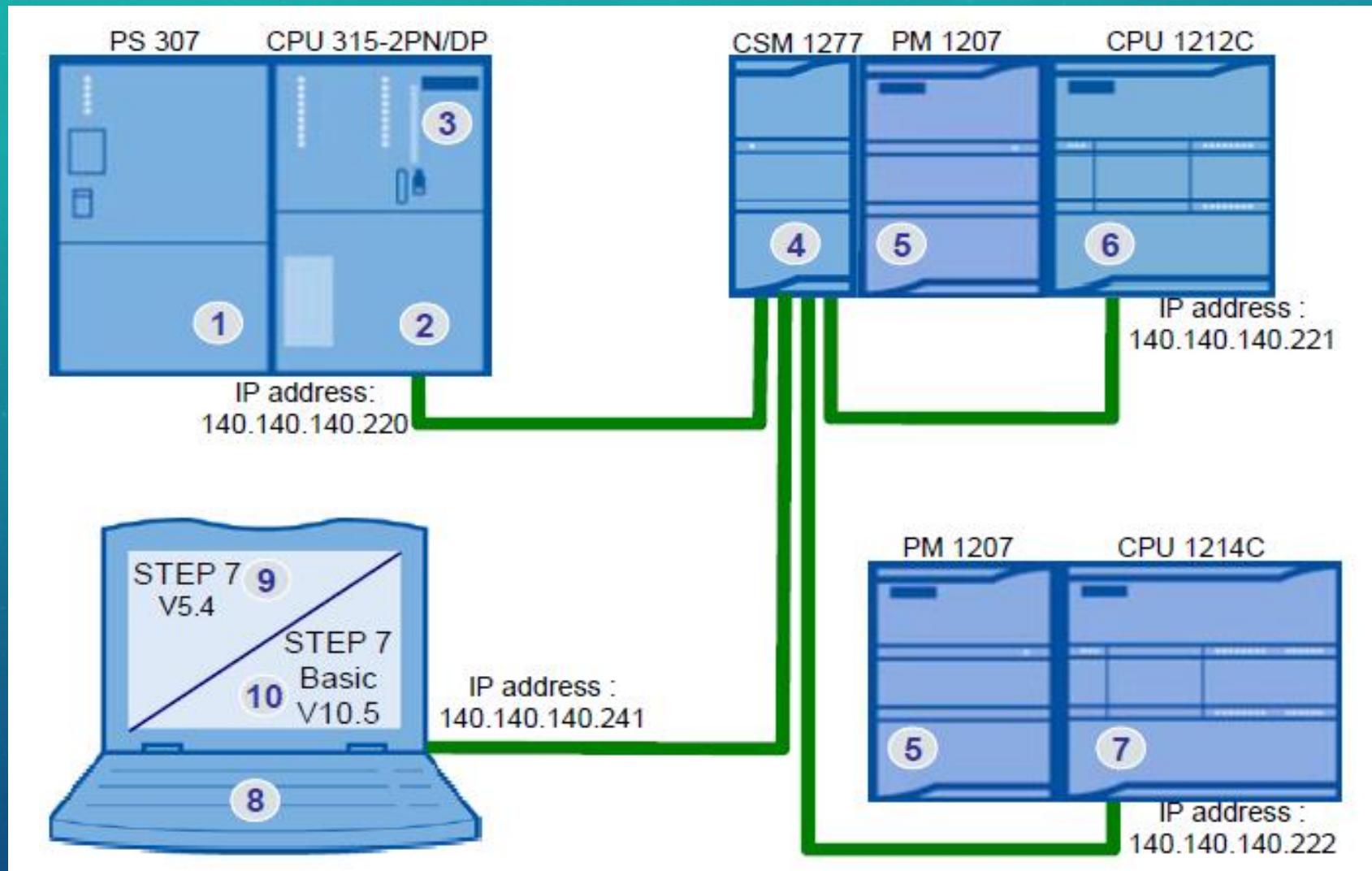
INTERFAZ MULTIPUNTO (MPI)

| | |
|----------------------------|---|
| Normalización | Específica de Siemens |
| Equipos | 32 como máximo |
| Método de acceso | Token (paso de testigo) |
| Velocidad de transferencia | 19, 2 Kbits/s 187, 5 Kbits/s o 12 Mbits/s |
| Soporte de transmisión | Cable bifilar apantallado o fibra óptica |
| Extensión de la red | Longitud de segmento 50mts. Con repetidores RS 485 hasta 1100m F.O > 100 km |
| Topología | Eléctrica: Línea Optica: Arbol, estrella, anillo |
| Servicios de comunicación | Funciones PG/OP Funciones S7 Funciones básicas S7 Comunicaciones por datos globales. |

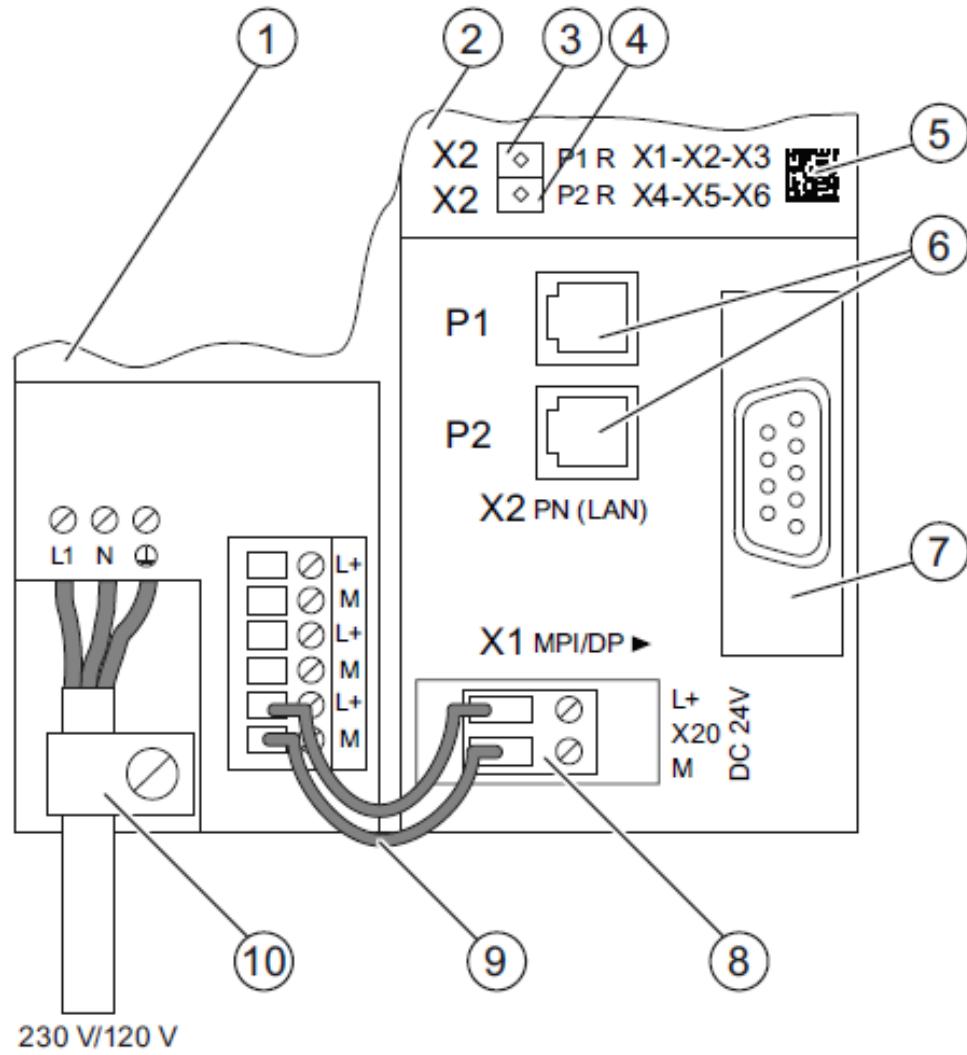
PC ADAPTER USB



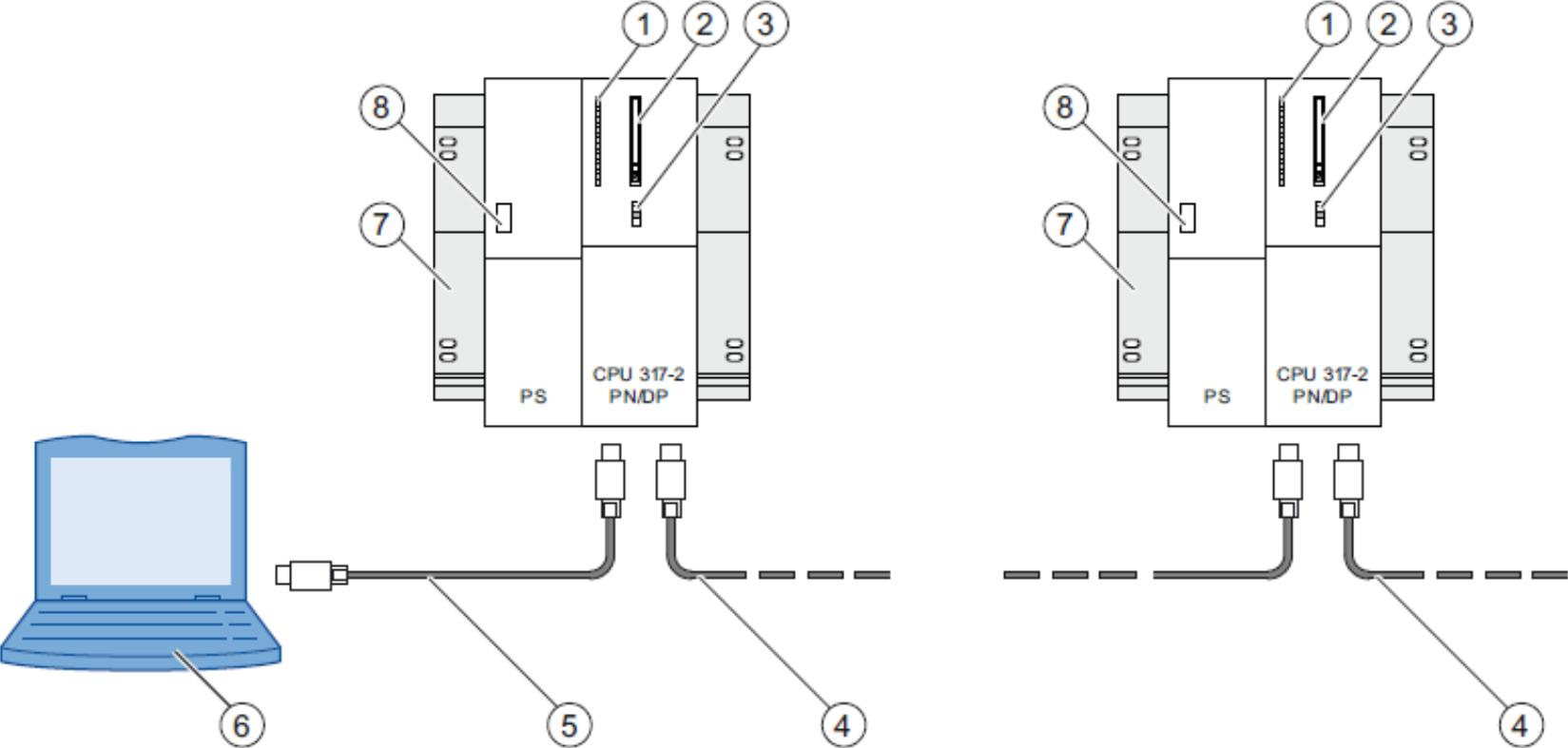
PROFINET



| | Profibus | Profinet |
|-----------------------------|-----------------|-----------------------|
| 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 |

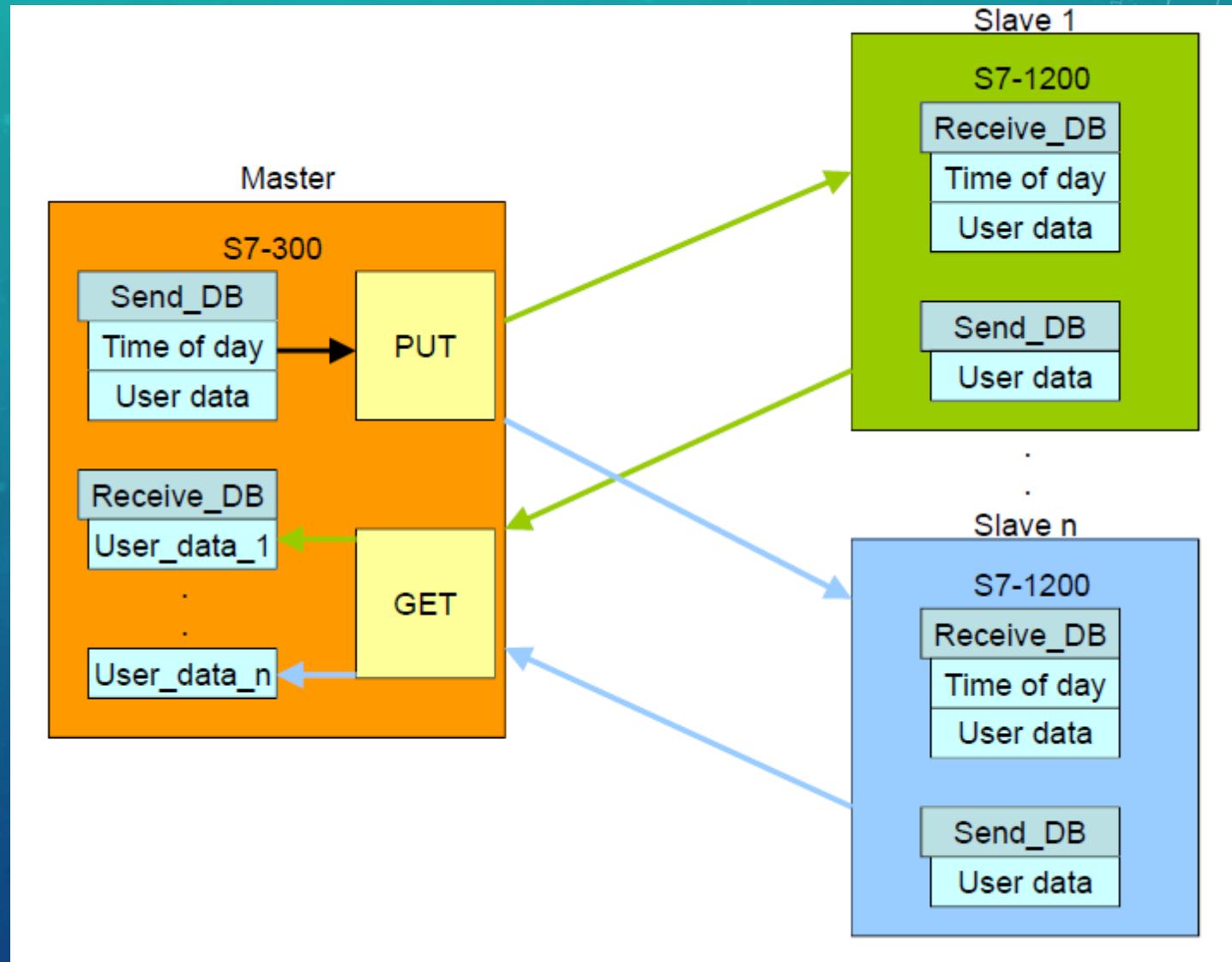


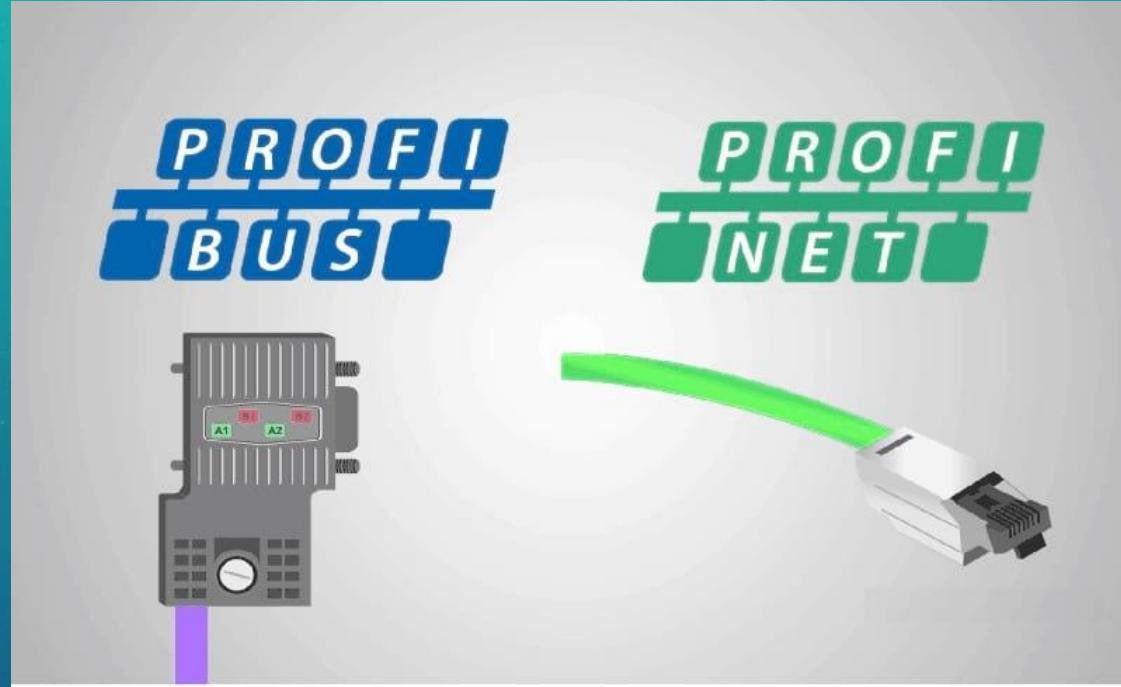
- ① Fuente de alimentación (PS)
- ② CPU 317-2 PN/DP
- ③ Puerto PROFINET 1
El estado del puerto 1 se señaliza mediante un LED de dos colores (verde/amarillo):
 - LED encendido en verde: Existe un LINK con un interlocutor
 - LED cambia a amarillo: Tráfico de datos activo (RX/TX)
 R: Puerto en anillo para crear una topología en anillo con redundancia de medios
- ④ Puerto PROFINET 2
El estado del puerto 2 se señaliza mediante un LED de dos colores (verde/amarillo):
 - LED encendido en verde: Existe un LINK con un interlocutor
 - LED cambia a amarillo: Tráfico de datos activo (RX/TX)
 R: Puerto en anillo para crear una topología en anillo con redundancia de medios
- ⑤ Dirección MAC y código de barras 2D
- ⑥ 2. Interfaz X2 (PN), con switch de 2 puertos
- ⑦ 1. interfaz X1 (MPI/DP)
- ⑧ Conexión para la fuente de alimentación
- ⑨ Cables de conexión entre la PS y la CPU
- ⑩ Alivio de tracción



- ① Indicadores de estado y error
- ② Ranura de la Micro Memory Card SIMATIC
- ③ Selector de modo
- ④ Cable Industrial Ethernet Twisted Pair para la conexión a los demás aparatos PROFINET
- ⑤ Cable Industrial Ethernet Twisted Pair para la conexión a la interfaz PN X2
- ⑥ Programadora (PG) con software STEP 7
- ⑦ Perfil soporte
- ⑧ Fuente de alimentación ON / OFF

PROFINET

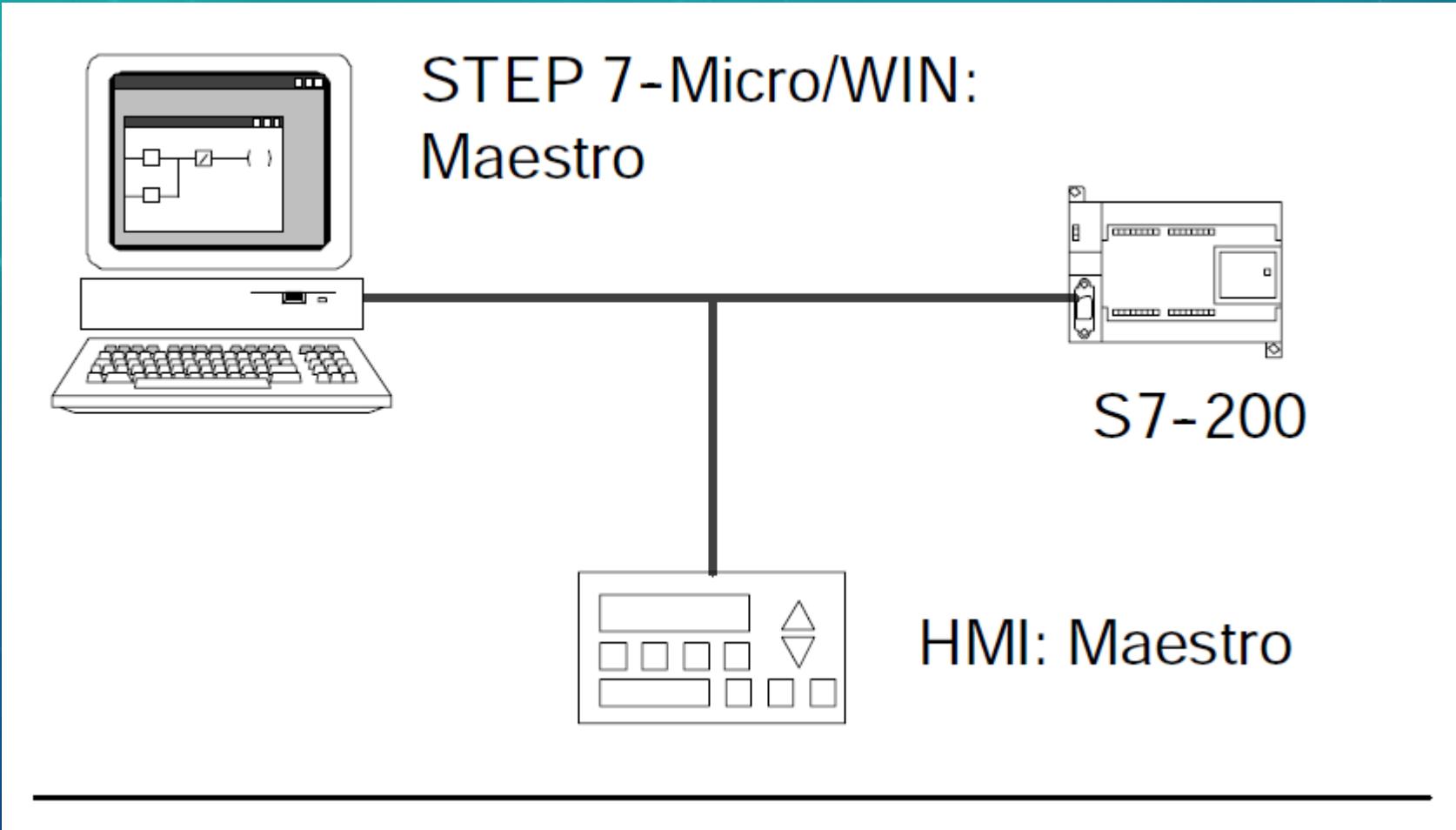




MARCAS DE CICLO

| Bit del byte de la marca de ciclo | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----------------------------------|-----|-------|-----|------|-----|-----|-----|-----|
| Duración del periodo (s) | 2,0 | 1,6 | 1,0 | 0,8 | 0,5 | 0,4 | 0,2 | 0,1 |
| Frecuencia (Hz) | 0,5 | 0,625 | 1 | 1,25 | 2 | 2,5 | 5 | 10 |

RED PPI

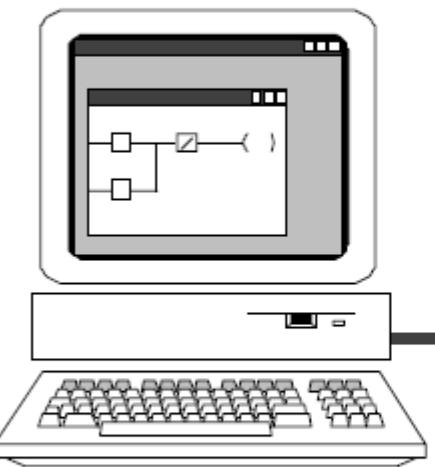


PROTOCOLO MPI

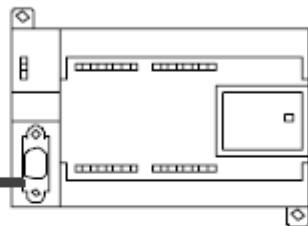
- El protocolo MPI soporta la comunicación maestro--maestro y maestro—esclavo. Para comunicarse con una CPU S7--200, STEP 7--Micro/WIN establece un enlace maestro--esclavo. El protocolo MPI no sirve para comunicarse con una CPU S7--200 que actúe de maestra.

RED MPI

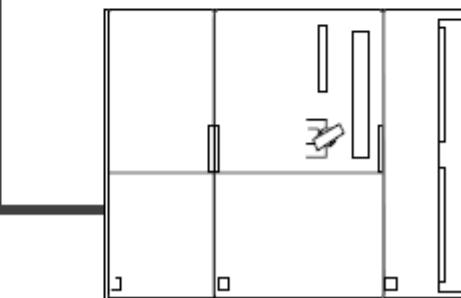
STEP 7-Micro/WIN:
Maestro



S7- 200: Esclavo



S7- 300: Maestro



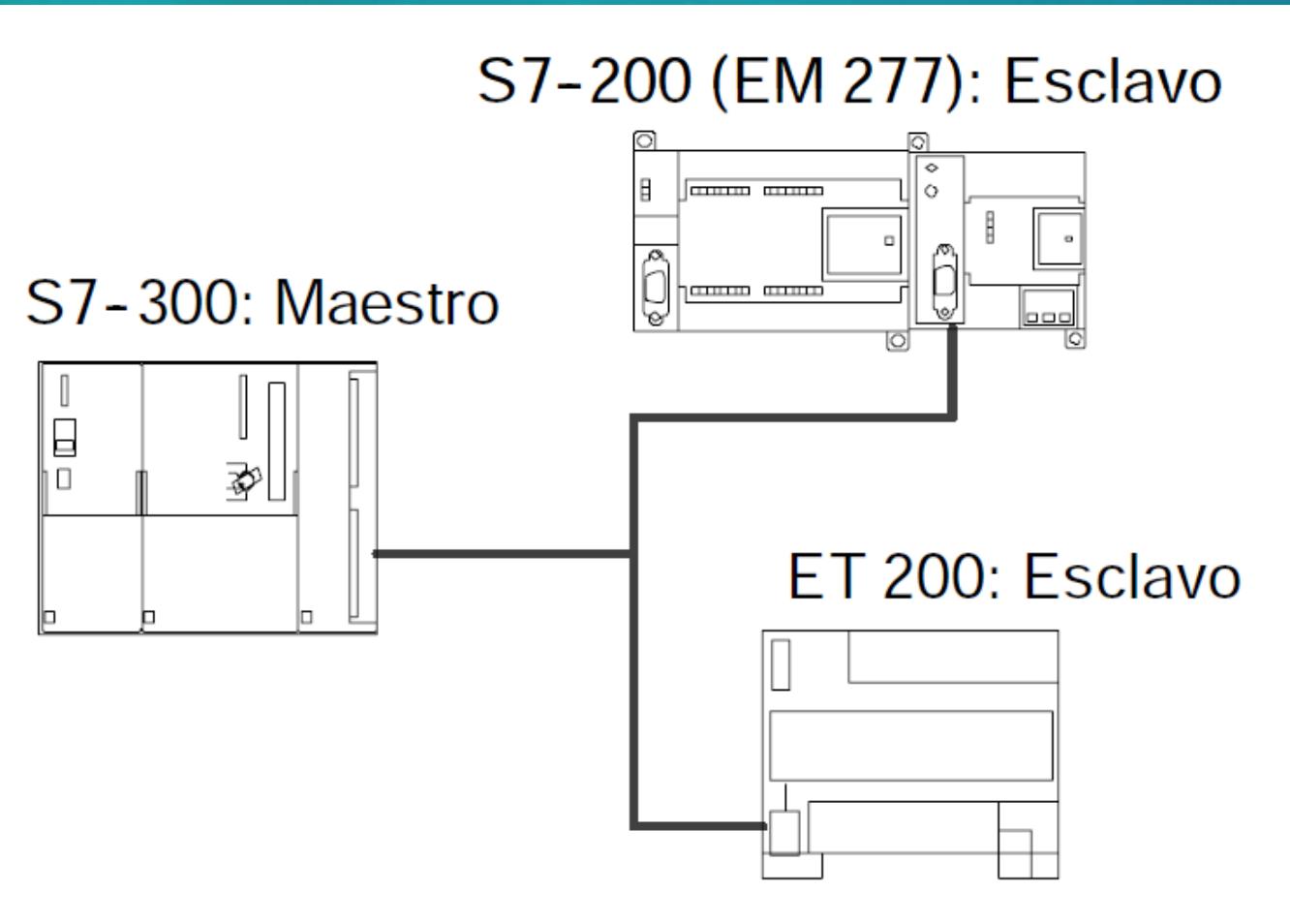
PROTOCOLO PROFIBUS

- El protocolo PROFIBUS se ha diseñado para la comunicación rápida con unidades periféricas descentralizadas (E/S remotas). Hay numerosos aparatos PROFIBUS ofrecidos por diversos fabricantes. Estos aparatos abarcan desde módulos sencillos de entradas o salidas hasta controladores de motores y sistemas de automatización (autómatas programables).

PROTOCOLO PROFIBUS

- Por lo general, las redes PROFIBUS incorporan un maestro y varios esclavos. La configuración del maestro permite detectar los tipos de esclavos conectados, así como sus respectivas direcciones. El maestro inicializa la red y verifica si los esclavos coinciden con la configuración. Continuamente, el maestro escribe los datos de salida en los esclavos y lee de allí los datos de entrada.

RED PROFIBUS



REFERENCIA

- Fundamentos de la técnica de automatización, Libro técnico, FESTO, www.festo-didactic.com
- Automatización industrial, Ingeniería de Sistemas Industriales y Diseño, ISBN: 978-84-693-0994-0
- Pérez-López, E. Los sistemas SCADA en la automatización, industrial. Tecnología en Marcha.
- Automatismos Industriales, Taller de Publicaciones- Universidad Tecnológica de Pereira, ISBN: 978-958-8272-99-3
- Programación de Autómatas, Introducción a Grafket
http://www.ieef.upm.es/moodle/pluginfile.php/946/mod_resource/content/0/Grafket.pdf
- Programación de PLC, https://catedras.facet.unt.edu.ar/aycp/wp-content/uploads/sites/88/2015/03/5_c-Programacion-de-PLC-1.pdf
- Diagrama de contactos, <https://www.yumpu.com/es/document/view/14484017/diagrama-de-contactos-ladder>
- **Manual del sistema de automatización S7-200**, Número de referencia del manual: **6ES7298--8FA24--8DH0**