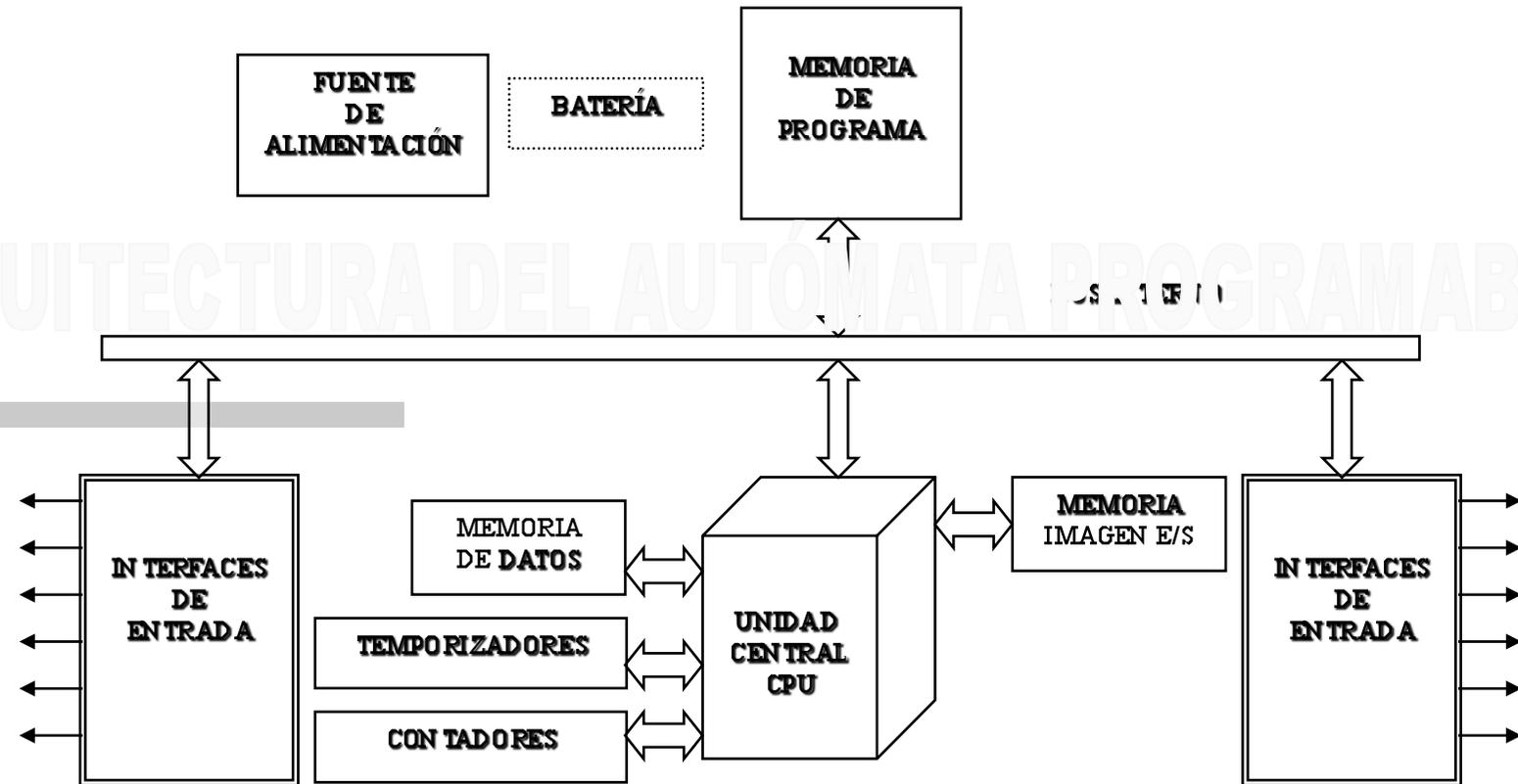


AUTÓMATAS PROGRAMABLES



Arquitectura del autómata programable

- 
- Introducción
 - Estructura externa del autómata
 - Bloques que forman un autómata programable
 - Arquitectura interna de un autómata programable
 - Unidad central de proceso
 - Memoria
 - Memorias internas
 - Memoria de programa
 - Interfaces de entrada salida
 - Ejemplo

Arquitectura del autómata programable

Introducción

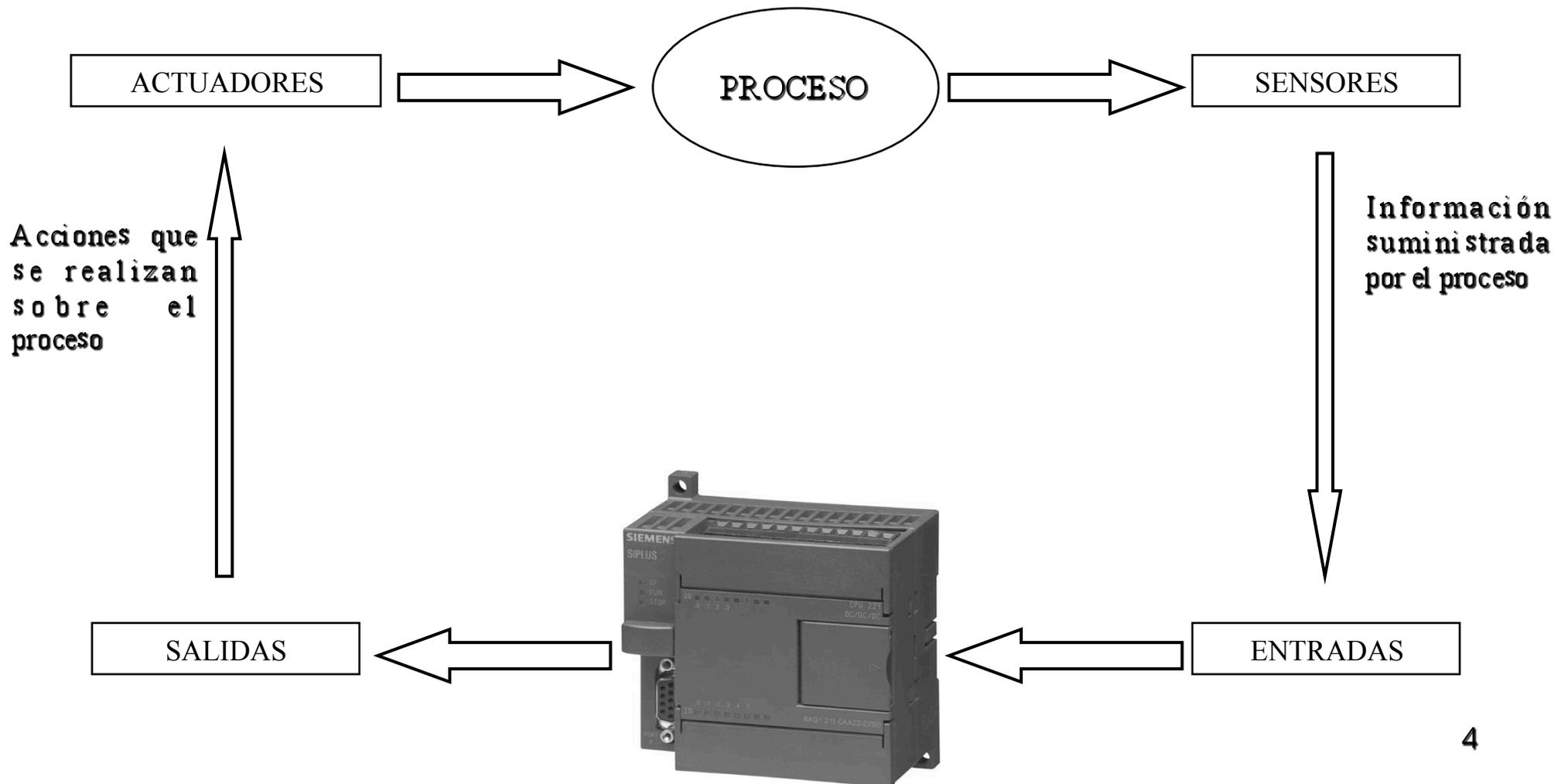
- Un autómata programable es:

“Un equipo electrónico, basado en un microprocesador o microcontrolador, que tiene generalmente una configuración modular, puede programarse en lenguaje no informático y está diseñado para controlar procesos en tiempo real y en ambiente agresivo (ambiente industrial)”

- Una característica diferenciadora del autómata programable frente a otros sistemas de control programables está en la estandarización de su hardware.

Arquitectura del autómata programable

Introducción



Arquitectura del autómata programable

Introducción

■ Clasificación (estructura externa)

- Estructura compacta

- Incluye todos sus elementos en un solo bloque: fuente de alimentación, CPU, memorias, entrada/salida, etc.
- Se utiliza en aplicaciones en el que el número de entradas/salidas es pequeño, poco variable y conocido a priori.
- Cuentan con una carcasa de carácter estanco, que permite su empleo en ambientes industriales especialmente hostiles.

- Estructura modular

- Permite que se añadan módulos para más funcionalidades: P.e. E/S digitales, E/S analógicas, módulos de comunicación para redes.
- Se adapta mejor a las necesidades del diseño, y a las posteriores actualizaciones. Configuración del sistema variable.
- Permite un funcionamiento parcial del sistema frente a averías localizadas y una rápida reparación al poderse sustituir los módulos averiados.

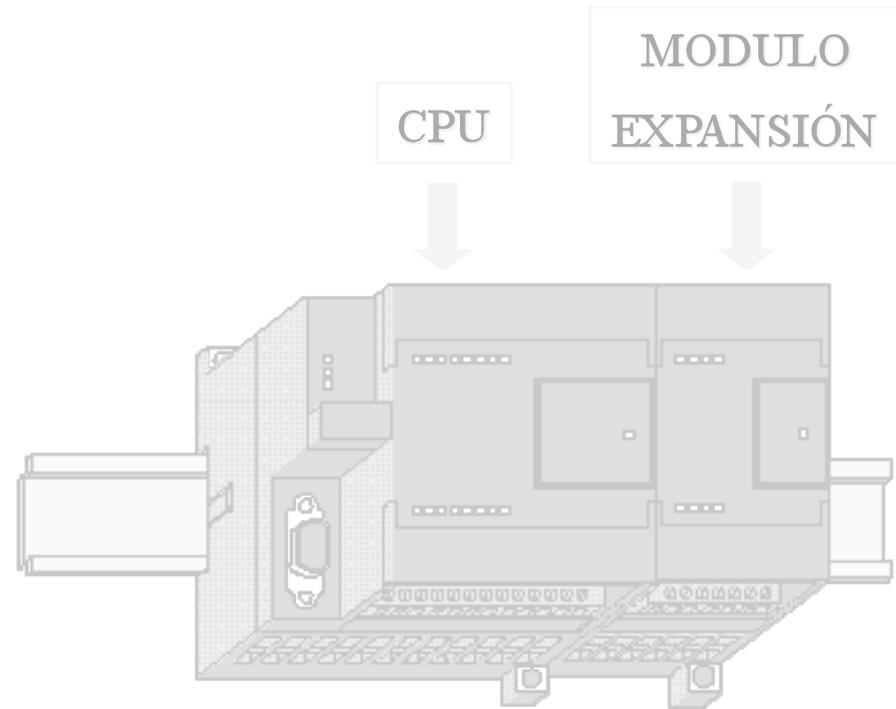
Arquitectura del autómata programable

Introducción

- Estructura modular. Del autómata S7-224.



COMPACTOS

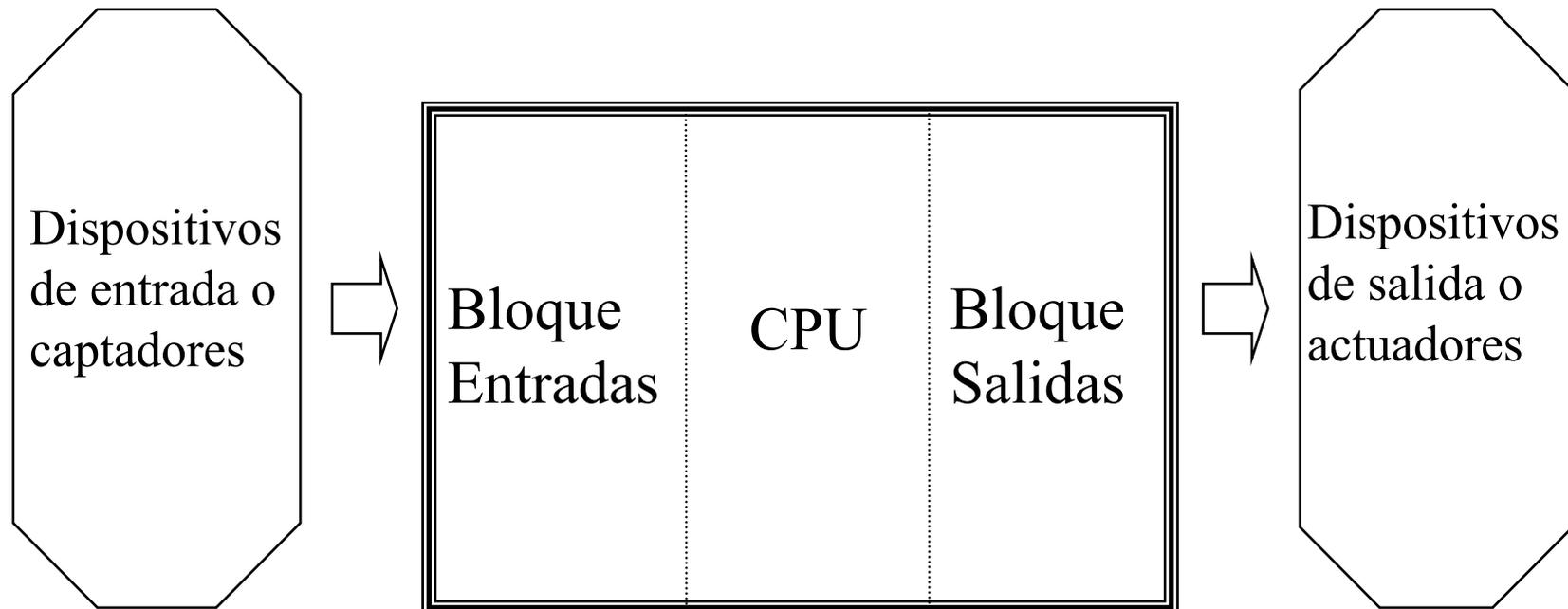


MODULAR

Arquitectura del autómata programable

Introducción

■ Bloques principales



Arquitectura del autómata programable

Bloques que forman un autómata programable

■ Bloques principales

- Bloque de entradas

- Adapta y codifica de forma comprensible para la CPU las señales procedentes de los dispositivos de entrada o captadores, como, por ejemplo, pulsadores, finales de carrera, sensores, etc.
- Normalmente proporciona una separación eléctrica, de manera que se protegen los circuitos internos del autómata.

- Bloque de salidas

- Decodifica las señales procedentes de la CPU, las amplifica y las envía a los dispositivos de salida o actuadores, como lámparas, relés, contactores, arrancadores, electroválvulas, etc.

- Unidad central de proceso (CPU)

- Este bloque es el "cerebro" del autómata.
- Su función es la de interpretar las instrucciones del programa de usuario y en función de las entradas, activa las salidas deseadas.

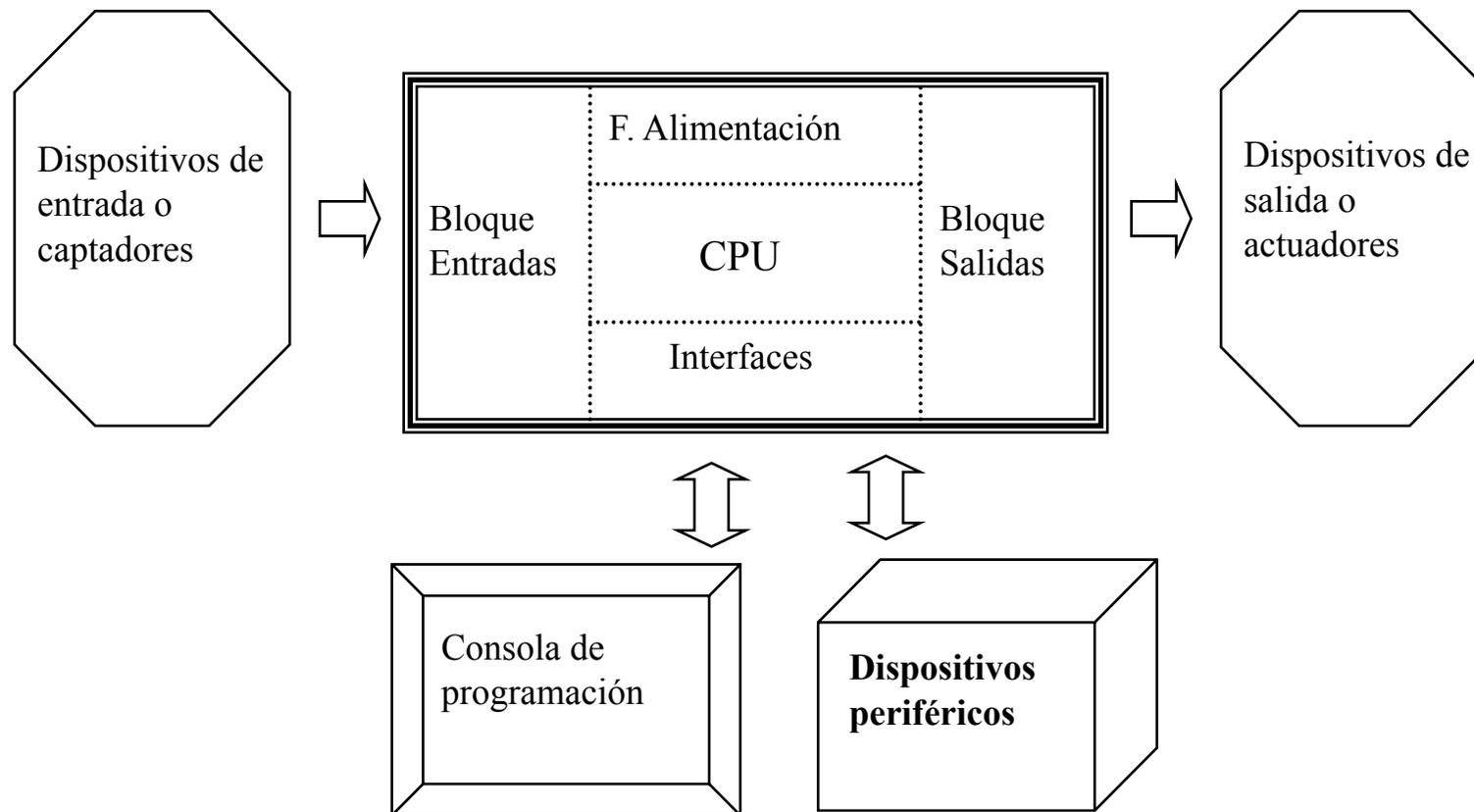
Arquitectura del autómata programable

Bloques que forman un autómata programable

- Otros bloques necesarios para el funcionamiento del autómata
 - Fuente de alimentación
 - A partir de una tensión exterior proporciona las tensiones necesarias para el funcionamiento de los distintos circuitos electrónicos del autómata.
 - Batería o condensador de alta potencia: para mantener el programa y algunos datos en la memoria si hubiera un corte de la tensión exterior.
 - Periféricos
 - Son aquellos elementos auxiliares, físicamente independientes del autómata, que se unen al mismo para realizar una función específica. Como tales no intervienen directamente ni en la elaboración ni en la ejecución del programa.
 - PC o consolas de programación
 - Interfaces
 - Circuitos o dispositivos electrónicos que permiten la conexión a la CPU de los elementos periféricos descritos.

Arquitectura del autómata programable

Bloques que forman un autómata programable



Arquitectura del autómata programable

Bloques que forman un autómata programable

- Introducción
 - Estructura externa del autómata
 - Bloques que forman un autómata programable
-  - Arquitectura interna de un autómata programable
 - Unidad central de proceso
 - Elementos que integran la CPU
 - Memoria
 - Tipos de memorias
 - Memorias internas
 - Memoria de programa
 - Interfaces de entrada salida
 - Ejemplo

Arquitectura interna de un autómata programable

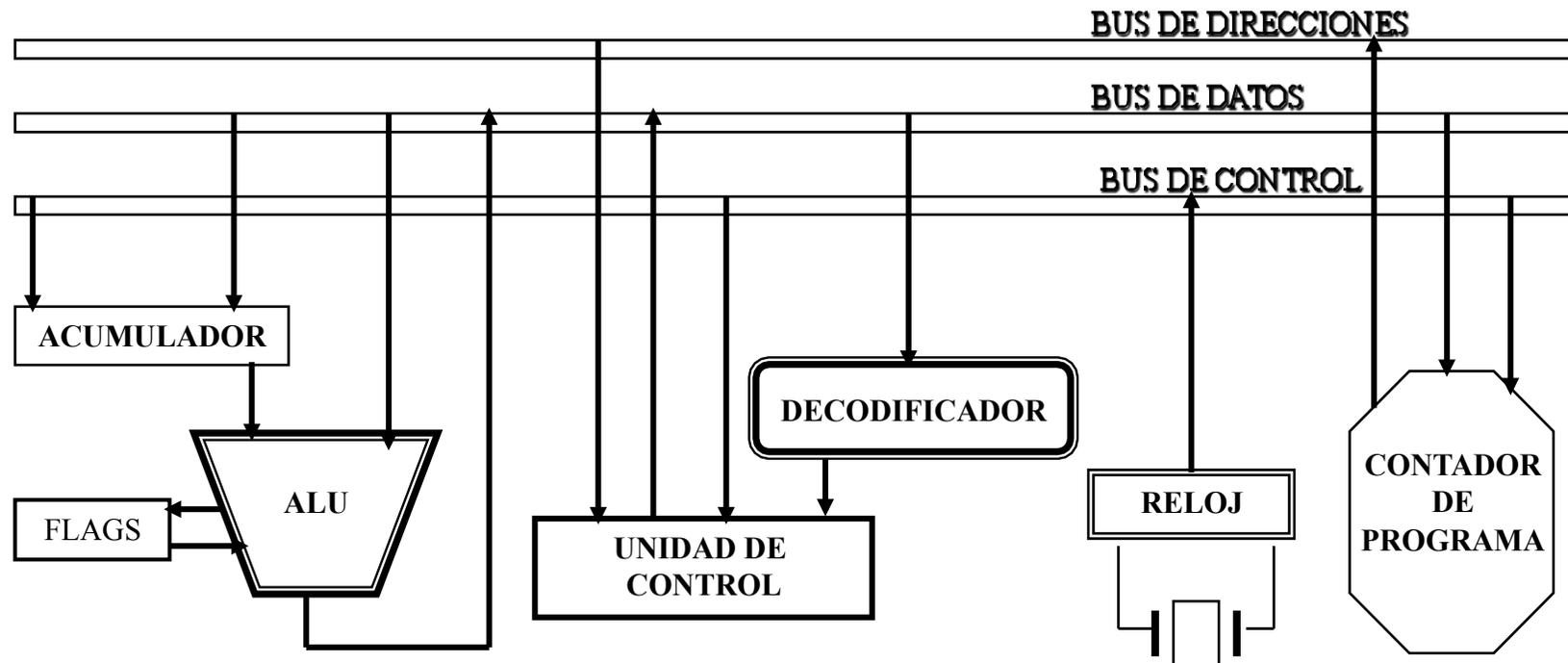
Unidad central de proceso (CPU)

- Veremos más en detalle todos los bloques de un autómata.
- La CPU (**Control Processing Unit**) es la encargada de ejecutar instrucciones una a una y activar el sistema de entradas y salidas. Adquiere las instrucciones una a una del programa residente en memoria.
- Bloques fundamentales de una CPU
 - ALU (**Arithmetic Logic Unit**, Unidad Aritmético Lógica)
 - Acumulador
 - Flags (p.e. acarreo)
 - Contador de programa (PC)
 - Unidad de control
 - Reloj (clock)

Arquitectura interna de un autómata programable

Unidad central de proceso (CPU)

- Esquema de los bloques fundamentales que componen una CPU



Arquitectura interna de un autómata programable

Unidad central de proceso (CPU)

■ ALU

- Unidad aritmético lógica, es la parte de la CPU donde se realizan los cálculos y las decisiones lógicas (combinaciones Y, O, sumas, comparaciones, etc.).

■ ACUMULADOR

- Almacena el resultado de la última operación realizada por la ALU.

■ FLAGS

- Indicadores de resultado de operación (mayor que, positivo, negativo, resultado cero, etc.). El estado de estos flags puede ser consultado por el programa.

■ CONTADOR DE PROGRAMA (PC)

- Se utiliza para el direccionamiento de la memoria donde se encuentran las instrucciones del programa de control, y del cual depende la secuencia de ejecución de ellas.

■ UNIDAD DE CONTROL

- Se encarga de leer las instrucciones y de decodificarlas. Genera las señales de control necesarias para coordinar el resto de acciones (von Neumann).

■ RELOJ (CLOCK)

- Genera una secuencia de impulsos que se utilizan para generar las señales de control y coordinar los diferentes elementos.

Arquitectura interna de un autómata programable

Memoria

- Las instrucciones del programa de usuario, así como los resultados parciales que genere, deben almacenarse en algún dispositivo → es necesario disponer de memoria.
- Distinguimos, principalmente, dos tipos diferentes de memoria:
 - **RAM (Random Access Memory)**, memoria de acceso aleatorio. Se conoce así porque permite acceder de forma aleatoria a cualquier elemento almacenado en un tiempo constante. Permite la lectura y la escritura por procedimientos eléctricos. La información almacenada desaparece al faltarle la alimentación.
 - **ROM (Read Only Memory)**, o memoria de sólo lectura. En estas memorias se puede leer su contenido, pero no se puede escribir en ellas; los datos e instrucciones los graba el fabricante y el usuario no puede alterar su contenido. Mantienen la información almacenada aunque haya un fallo en la alimentación.

Arquitectura interna de un autómata programable

Memoria

■ Clasificación (tecnología empleada)

- Memorias de lectura/escritura,
 - RAM (volátil), lectura/escritura.
 - FLASH (no volátil), lectura/escritura. Tecnología similar a EEPROM. Tienen un número máximo de ciclos de borrado/grabado.
- Memorias de sólo lectura, ROM (no volátil)
 - Memorias de sólo lectura, reprogramables, con borrado por ultravioletas, EPROM
 - Memorias de sólo lectura, reprogramables, alterables por medios eléctricos, EEPROM. Tienen un número máximo de ciclos de borrado/grabado. Menor capacidad de integración que FLASH.

Arquitectura interna de un autómata programable

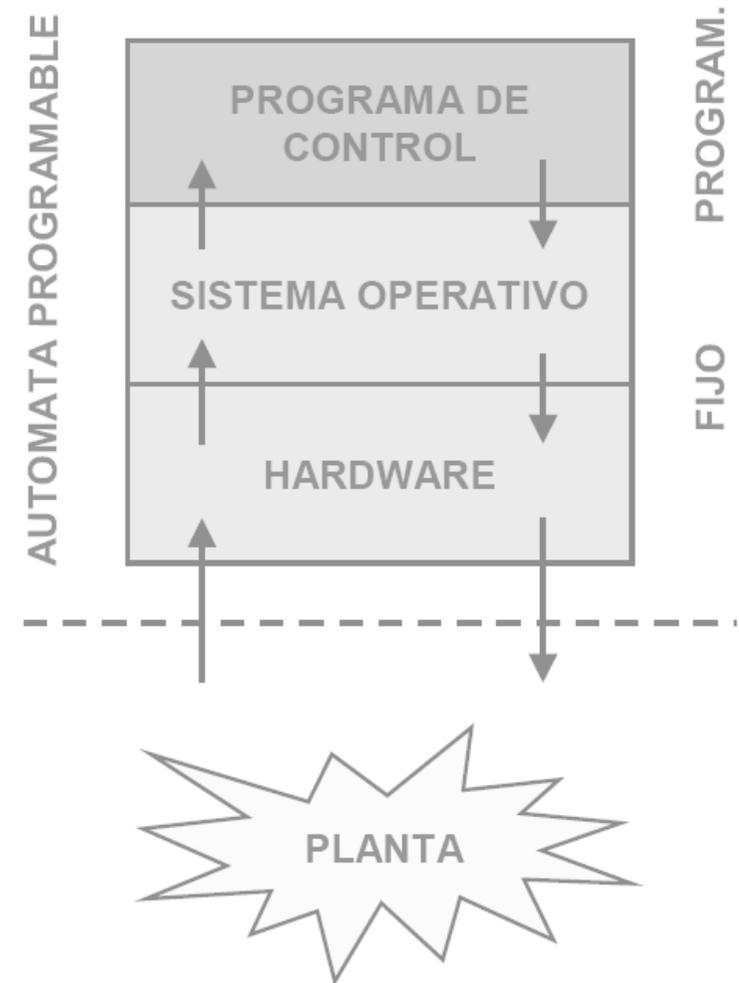
Memoria

- Los PLCs cuentan con una combinación de memorias ROM y RAM.
- Las memorias no volátiles resultan interesantes para almacenar el programa de usuario, ya que, de esta manera, no se borran si falla la alimentación. Para esta función existen diferentes alternativas:
 - Memoria RAM con batería. Fácilmente modificable, respaldada por batería en caso de fallo de alimentación.
 - Memoria EPROM, cuando el programa de usuario ya está depurado y estamos seguros de que funciona correctamente.
 - Mediante una memoria FLASH (EEPROM no es habitual, al no tener demasiada capacidad).

Arquitectura interna de un autómata programable

Sistema operativo

- Sistema operativo
 - Los PLC cuentan con una forma de Sistema operativo. Realiza la función de interfaz entre el programa de control del usuario y el hardware del autómata.
 - Está grabado por el fabricante y no puede ser modificado por el usuario.
 - Se almacena en ¿?
 - ROM por el fabricante.
 - Lo podemos definir como un conjunto de funciones que permiten manejar fácilmente los recursos hardware del autómata (E/S, contadores, temporizadores).
 - Permite acceder a los recursos de forma sencilla desde el programa de control.
- Programa de control
 - El funcionamiento del sistema es de tipo interpretado.
 - El S.O. cuenta con un programa que se encarga de interpretar el programa del usuario y manejar el hardware en consecuencia.
 - Se almacena en ¿?
 - FLASH/RAM+batería o EPROM cuando el programa ya está depurado.
 - Se modifica por el usuario para adaptarlo al proceso de control.



Arquitectura interna de un autómata programable

Lenguaje interpretado

■ Lenguaje compilado vs. interpretado

- Compilado

- programa fuente \Rightarrow compilación \Rightarrow programa objeto \Rightarrow enlazado ó linkado \Rightarrow programa ejecutable (en código máquina)

- Interpretado

- Existe un programa especial que analiza y ejecuta el programa del usuario sentencia a sentencia.

■ ¿Qué lenguajes de programación conocéis?

- C/C++

- \rightarrow lenguaje compilado

- Java

- \rightarrow lenguaje interpretado.
- Existe un programa (JVM) que realiza la función de interfaz código \leftrightarrow máquina
- Las aplicaciones Java se "compilan" a un bytecode. El bytecode es un código intermedio más abstracto que el código máquina (que depende de cada procesador).
- La JVM interpreta (traduce) cada bytecode del programa al código nativo de cada máquina y S.O. particular.

- Matlab \rightarrow

- \rightarrow lenguaje interpretado. Matlab

Arquitectura interna de un autómata programable

Lenguaje interpretado

- Un lenguaje interpretado tiene generalmente las siguientes ventajas:
 - Portabilidad. P.e. mismos juegos en Java en diferentes móviles. El mismo programa de control puede funcionar en diferentes modelos de autómatas (del mismo fabricante...).
 - Comprobación de acceso ilegal a memoria.
 - **Watchdog**, comprobación periódica de que el programa no se ha "colgado".
 - Añade seguridad al funcionamiento del autómata (muy deseable en un proceso industrial)
- Como contrapartida, el lenguaje interpretado suele ser más lento que el compilado, ya que el intérprete debe traducir cada línea del programa por separado.

Arquitectura interna de un autómata programable

Memoria

■ Clasificación de memoria según el uso:

Nombre	Uso	Tipo
Sistema Operativo	Sist. Operativo del fabricante. Firmware	EPROM, FLASH
Memoria temporal (S.O.)	Cálculos intermedios del S.O.	RAM
Programa de control	Programa del usuario	RAM+batt. o FLASH
Memoria imagen de entradas	almacena las últimas señales leídas en la entrada, actualizándose tras cada ejecución completa del programa.	RAM
Memoria imagen de salidas	almacena las últimas señales enviadas a la salida, actualizándose tras cada ejecución completa del programa.	RAM
Memoria de variables internas	Relés internos, Contadores internos, temporizadores, marcas internas.	RAM

Memoria de datos

Memoria de programa

Arquitectura interna de un autómata programable

Memoria

- Podemos clasificarlas en función del tipo de variables que almacena y el número de bits que ocupa la variable:
 - Posiciones de 1 bit.
 - Memoria imagen entradas/salidas
 - Relés internos
 - Posiciones de 8, 16 o más bits.
 - Temporizaciones
 - Contadores
 - Otros registros de uso general

Arquitectura interna de un autómata programable

Memorias internas

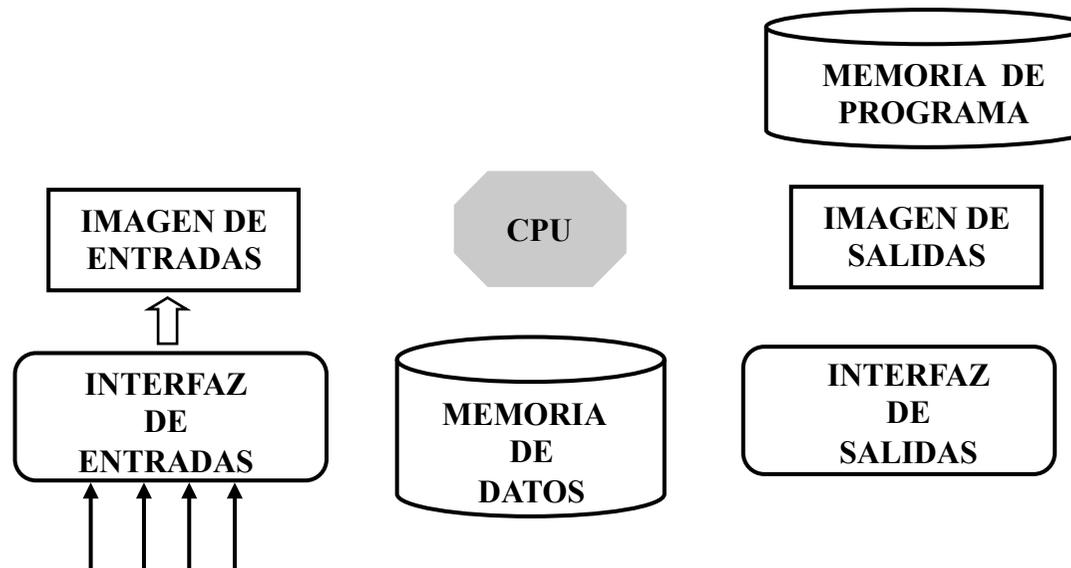
■ Otras funciones:

- **relés internos** ocupan posiciones RAM de 1 bit, y son utilizados como área de datos temporales, como salida de resultados de operaciones intermedias.
- **relés auxiliares/especiales** se guardan en posiciones de 1 bit, y mantienen información sobre señales necesarias para el sistema, como relojes, bits de control, flags de estados de la CPU, e información sobre el autómata (Run, Stop, Halt, errores, etc.). Estos relés pueden consultarse y utilizarse desde el programa usuario.
- Área de **temporizadores y contadores** ocupa posiciones de 16 bits o más bits, capaces de almacenar los valores de preselección y estado actualizados de estos elementos.

Arquitectura interna de un autómata programable

Ciclo de tratamiento de E/S

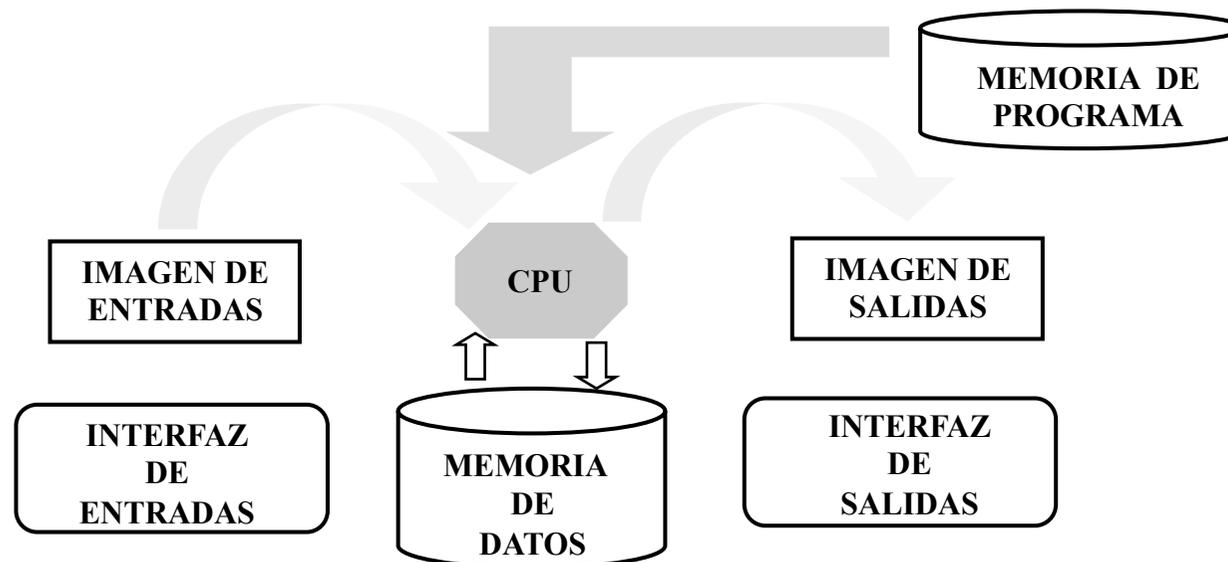
- Ciclo de tratamiento de las señales de entrada/salida a través de las **memorias imagen**:
 - Antes de la ejecución del programa de usuario, la CPU consulta los estados de las **entradas físicas** y carga con ellos la **memoria imagen de entradas**.



Arquitectura interna de un autómata programable

Ciclo de tratamiento de E/S

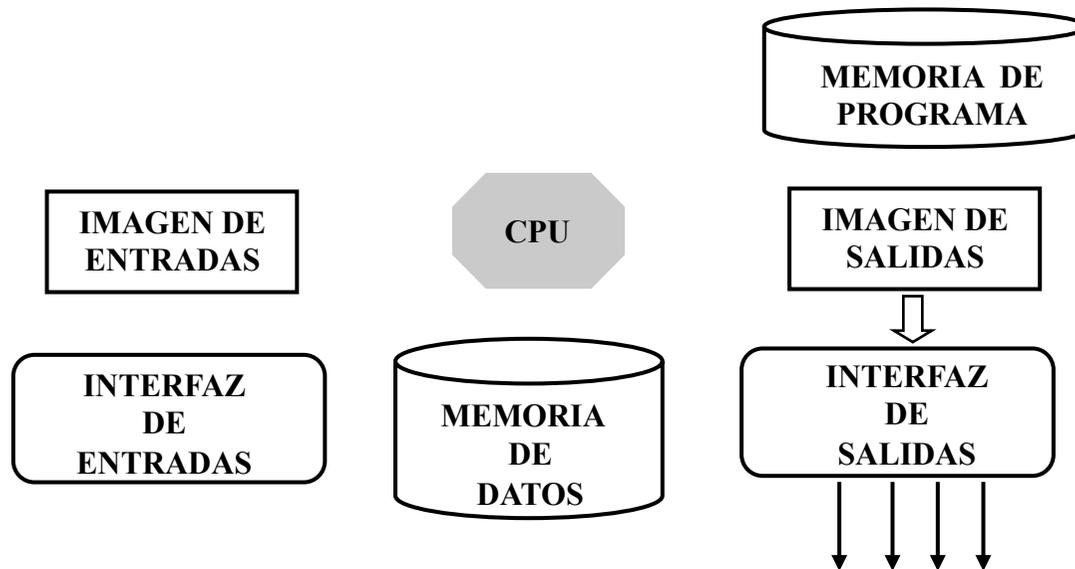
- Durante la ejecución del programa de usuario, la CPU realiza los cálculos a partir de los datos de la memoria imagen y del estado de los temporizadores, contadores y relés internos. El resultado de estos cálculos queda depositado en la memoria imagen de salidas.



Arquitectura interna de un autómata programable

Memorias internas

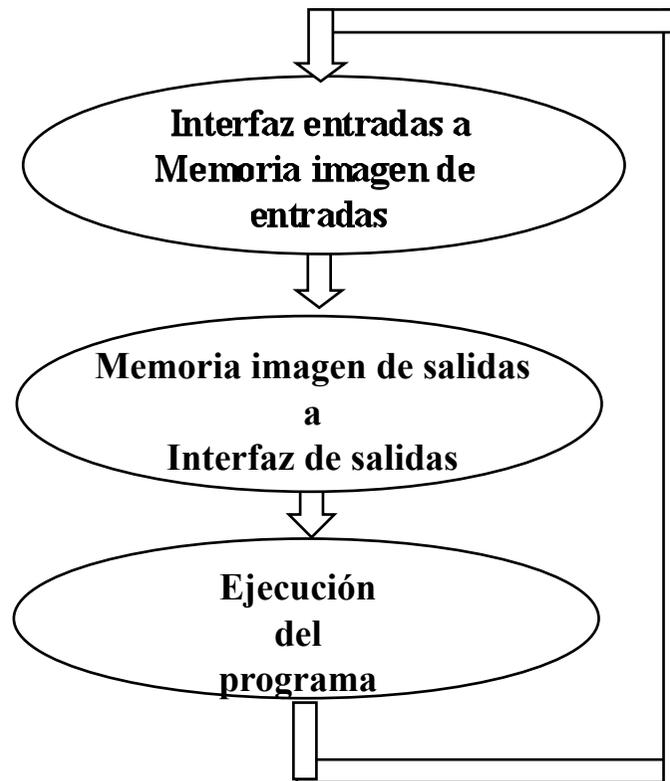
- Finalizada la ejecución, la CPU transfiere a las interfaces de salida los estados de las señales contenidos en la memoria imagen de salidas, quedando el sistema preparado para comenzar un nuevo ciclo



Arquitectura interna de un autómata programable

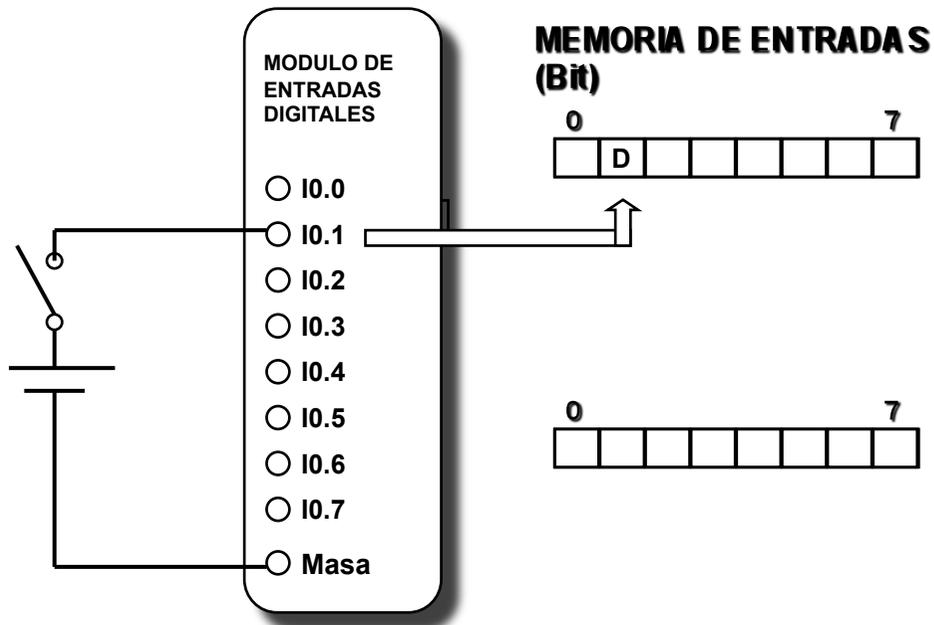
Memorias internas

- Ciclo real (más rápido)

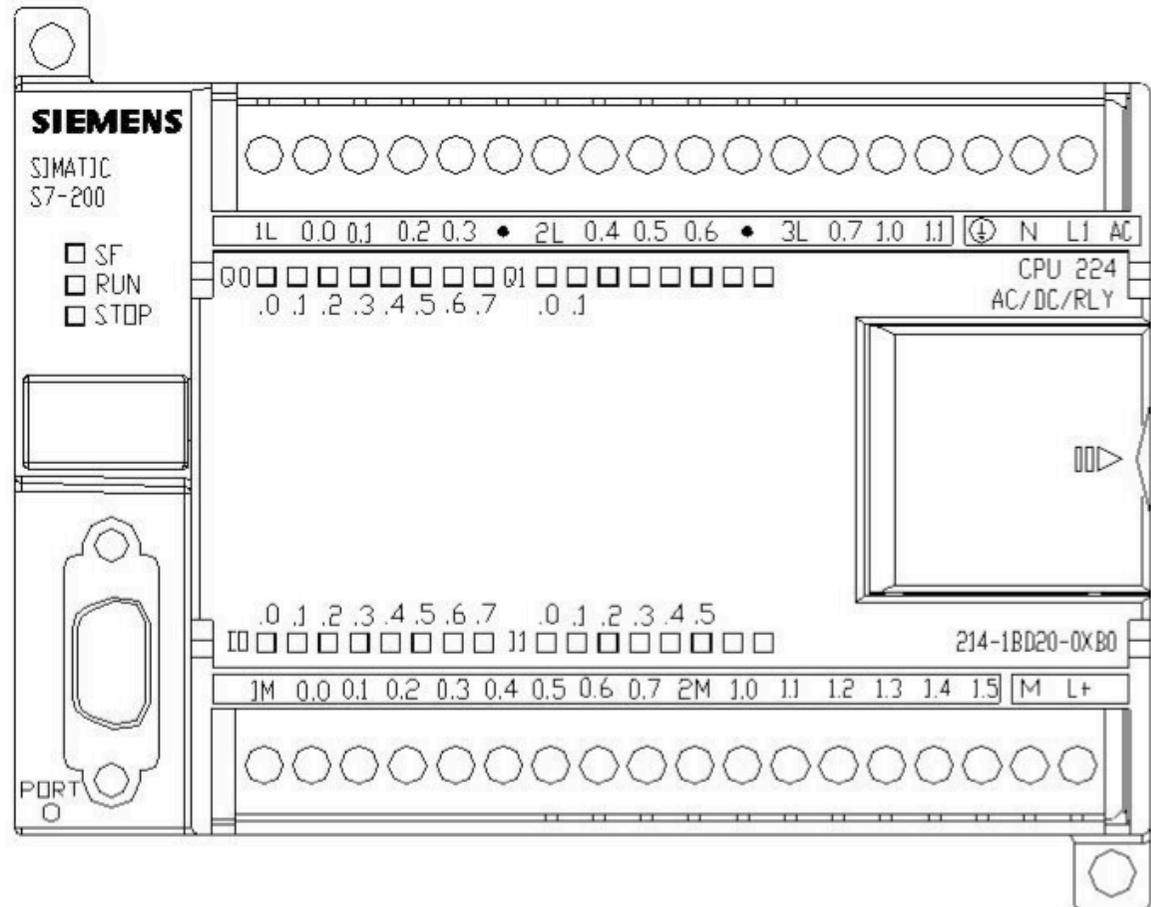


Memorias internas

- Ejemplo: Carga en la memoria imagen de entradas de las entradas digitales.

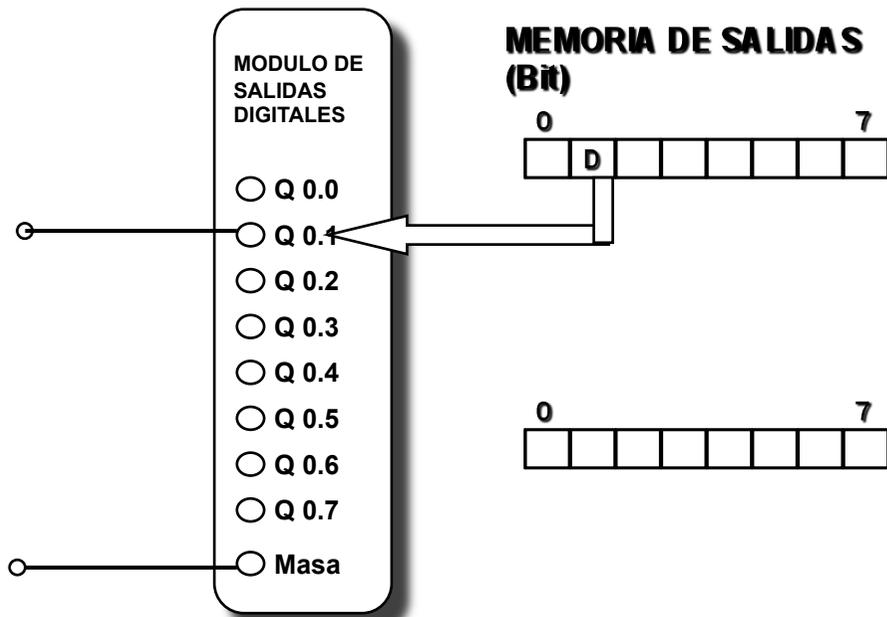


Cuando se cierra el interruptor se guarda un 1 en el Bit 1 del Byte 0

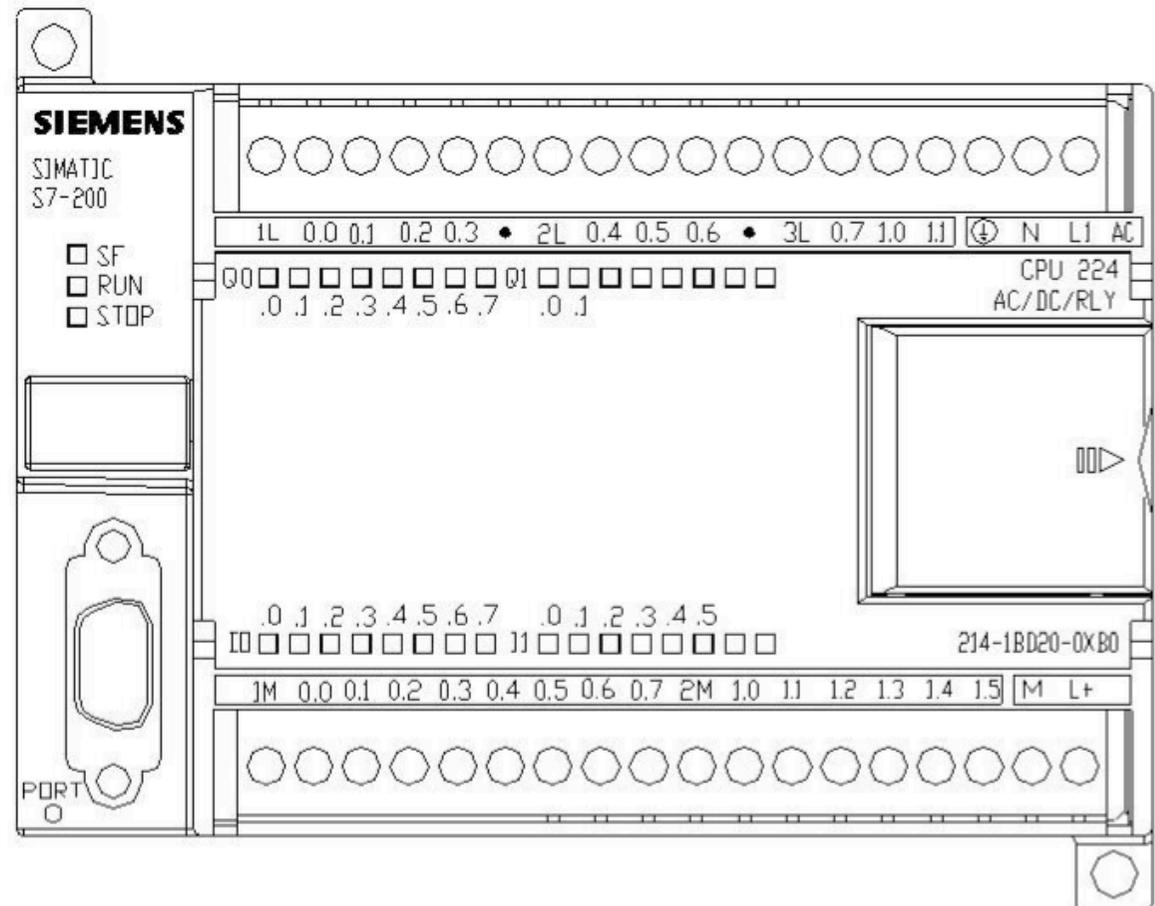


Memorias internas

- Ejemplo: Esquema de la transferencia de los datos contenidos en la memoria de salidas a las salidas digitales

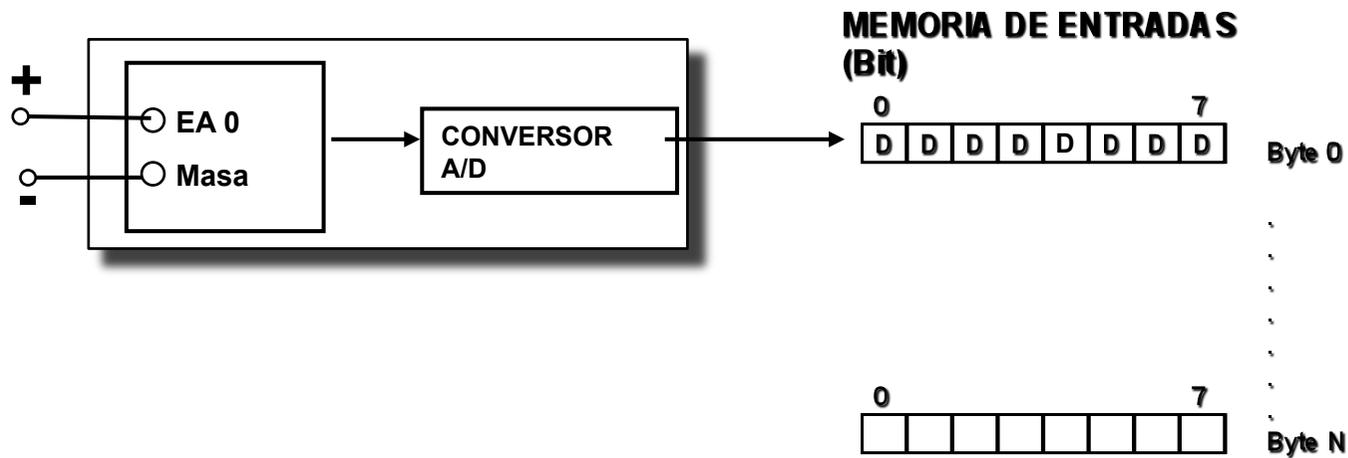


Si el dato almacenado es un 1, el contacto Q1 esta cerrado



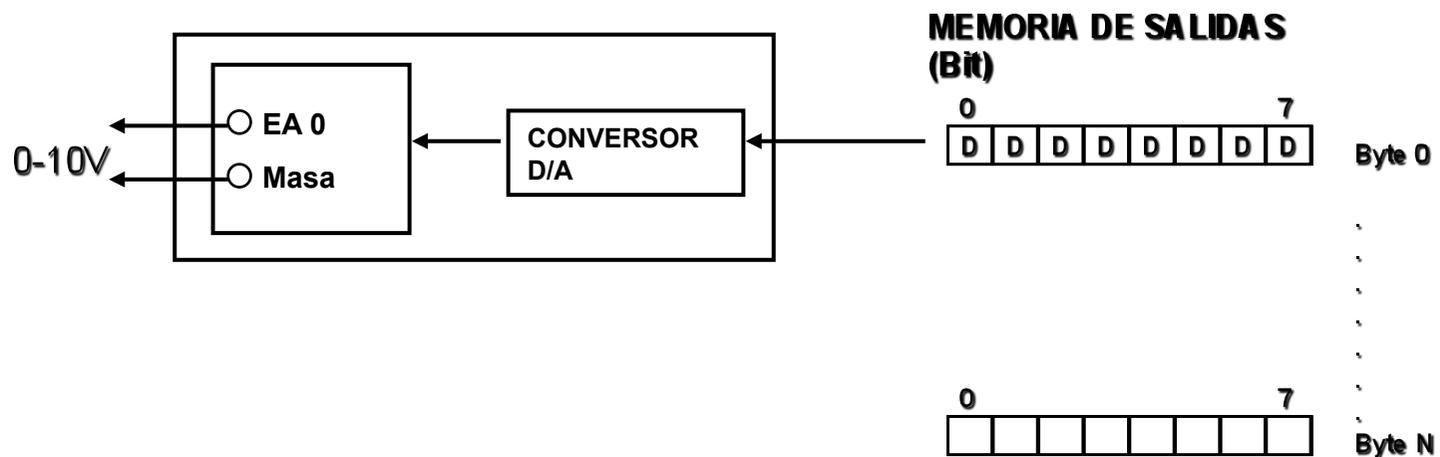
Memorias internas

- Ejemplo: Carga en la memoria imagen de entradas de las entradas analógicas.



Memorias internas

- Ejemplo: Esquema de la transferencia de los datos contenidos en la memoria de salidas a las salidas analógicas



Arquitectura del autómata programable

Bloques que forman un autómata programable

- Introducción
 - Estructura externa del autómata
 - Bloques que forman un autómata programable
- Arquitectura interna de un autómata programable
 - Unidad central de proceso
 - Elementos que integran la CPU
 - Memoria
 - Tipos de memorias
 - Memorias internas
 - Memoria de programa
 - Interfaces de entrada salida
 - Ejemplo



Arquitectura interna de un autómata programable

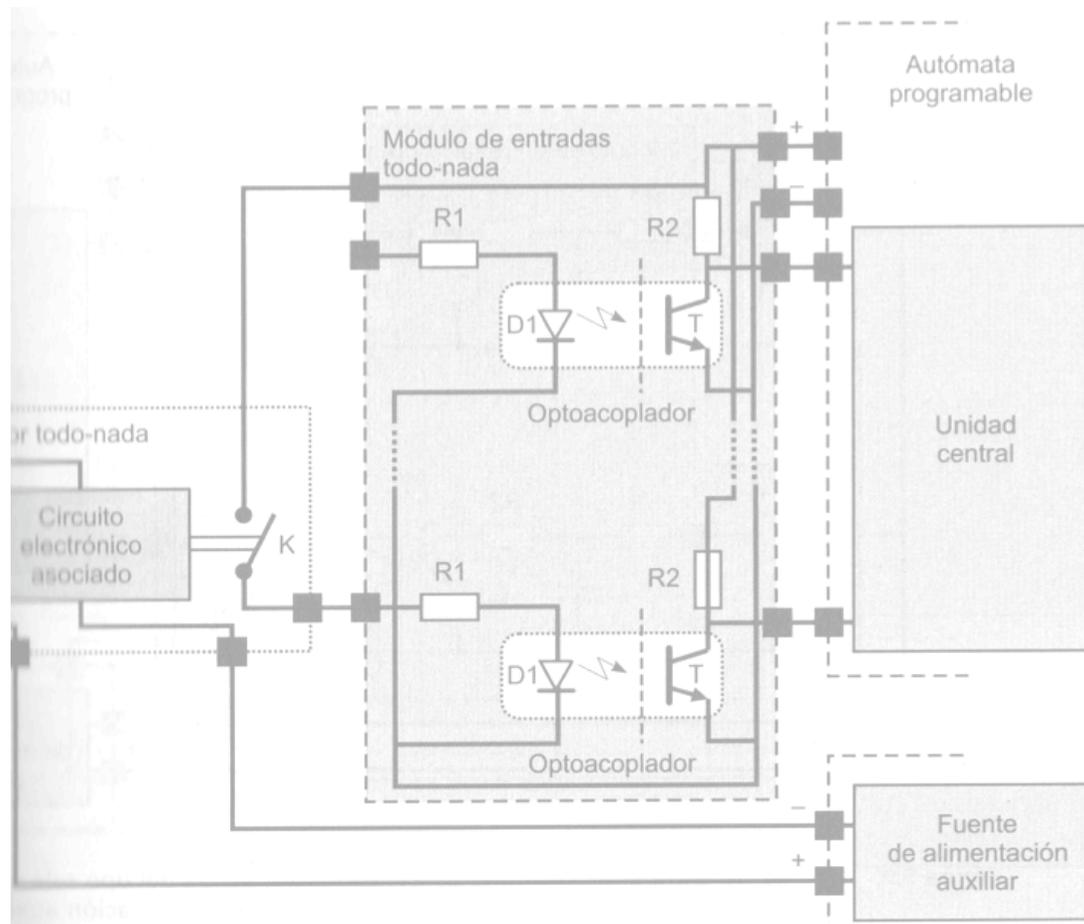
Interfaces de entrada y salida

- A continuación trataremos las interfaces de E/S
- Su misión consiste en establecer un puente de comunicación entre el autómata y el proceso
 - Las interfaces de entrada filtran, adaptan y codifican de forma comprensible para la CPU las señales procedentes de los elementos de entrada.
 - Las interfaces de salida son las encargadas de decodificar, y amplificar las señales generadas durante la ejecución del programa antes de enviarlas a los elementos de salida.
- En los autómatas pequeños, el tipo de interfaces disponibles suele ser limitado, siendo las más frecuentes:
 - Entradas
 - Corriente continua a 24 o 48 Vcc. Corriente alterna a 110 o 220 Vca. Analógicas de 0-10 Vcc o 4-20 mA
 - Salidas
 - Por relé. Estáticas por triac a 220 Vca máximo. Analógicas de 0-10 V o 4-20 mA.

Arquitectura interna de un autómata programable

Interfaces E/S

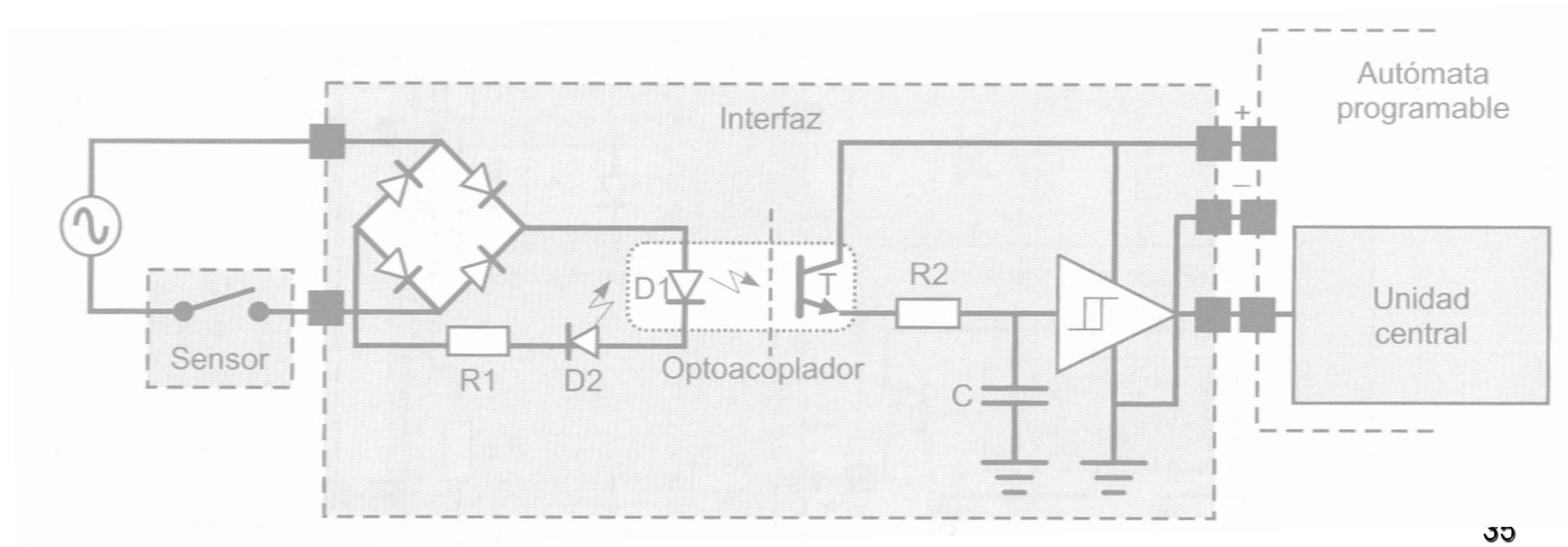
- Entrada digital optoacoplada (protección contra sobrevoltajes/sobreintensidades)
- El sensor (todo/nada) actúa como un ruptor, aplicando voltaje a la entrada física del autómata. Típicamente 24 o 48 Vcc.



Arquitectura interna de un autómata programable

Interfaces E/S

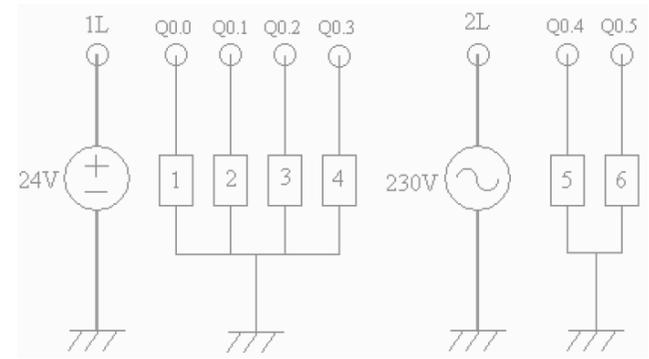
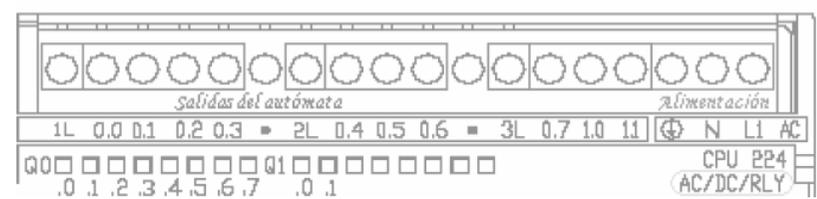
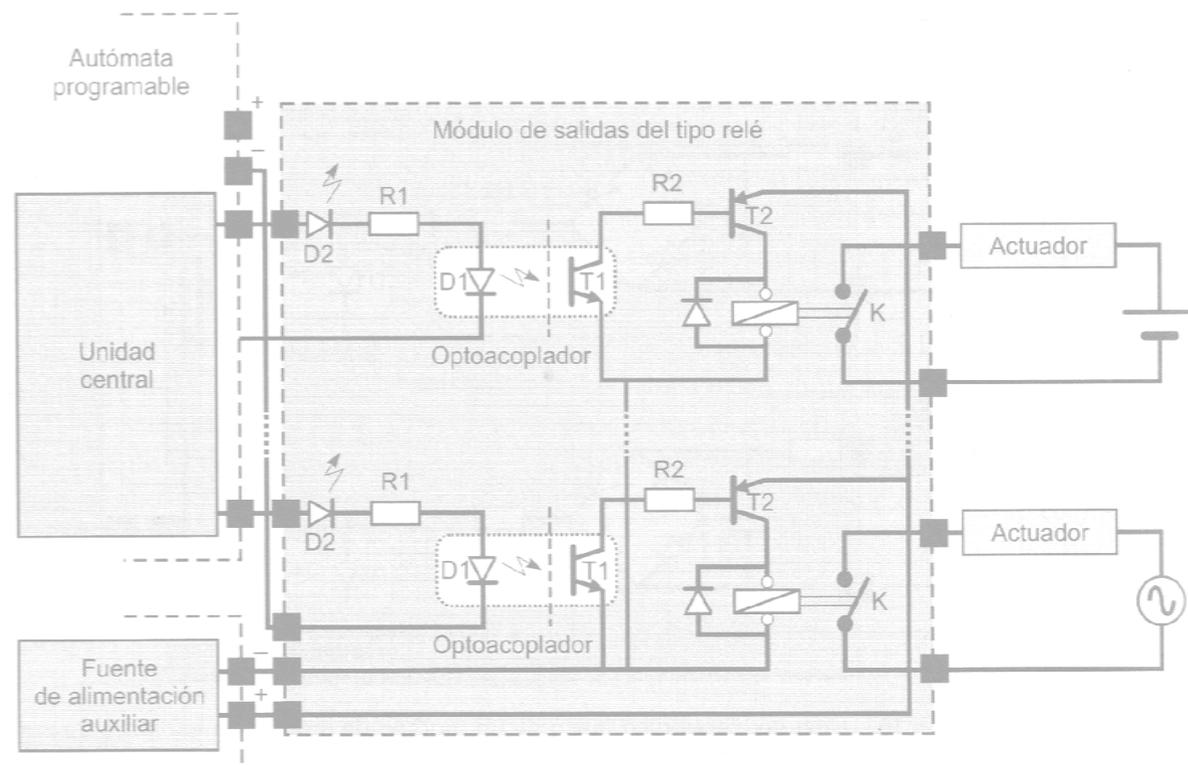
- Es posible también tener entradas en alterna.
- En este caso, el voltaje alterno se rectifica mediante un puente de diodos.
- El diodo D2 advierte al usuario de la activación de la entrada.
- La red R2-C es un filtro paso-bajo que filtra la componente alterna de la señal (50-60Hz).
- Normal: Corriente alterna a 110 o 220 Vca.



Arquitectura interna de un autómata programable

Interfaces E/S

- Se muestra una interfaz de salidas por relé optoacopladas.
- La principal ventaja de tener las salidas por relé es la capacidad de manejar cargas en alterna y continua de forma muy sencilla.
- Por ejemplo, en el caso del S7-200, existen varios terminales unidos a potencial común, cada uno accionado por un relé.

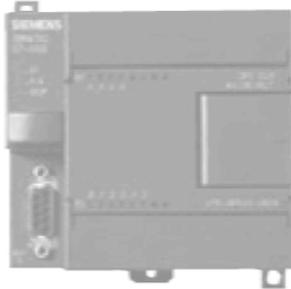


Ejemplos

■ Ejemplos: Gama de autómatas Siemens S7-200

compacto

CPU 221



6 inputs / 4 outputs
not expandable
10 I/O max.



Input modules

CPU 222

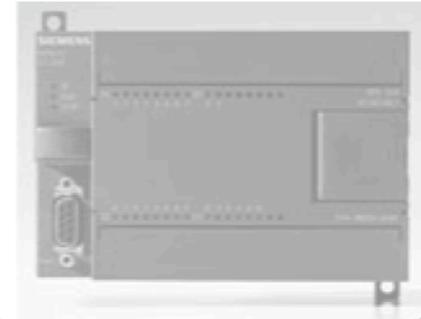


8 inputs / 6 outputs
+ 2 expansion modules max.
94 I/O max.

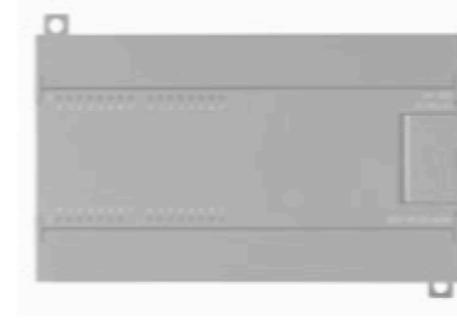


Output modules

CPU 224



14 inputs / 10 outputs
+ 7 expansion modules max.
224 I/O max.



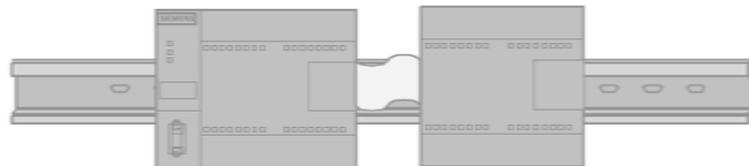
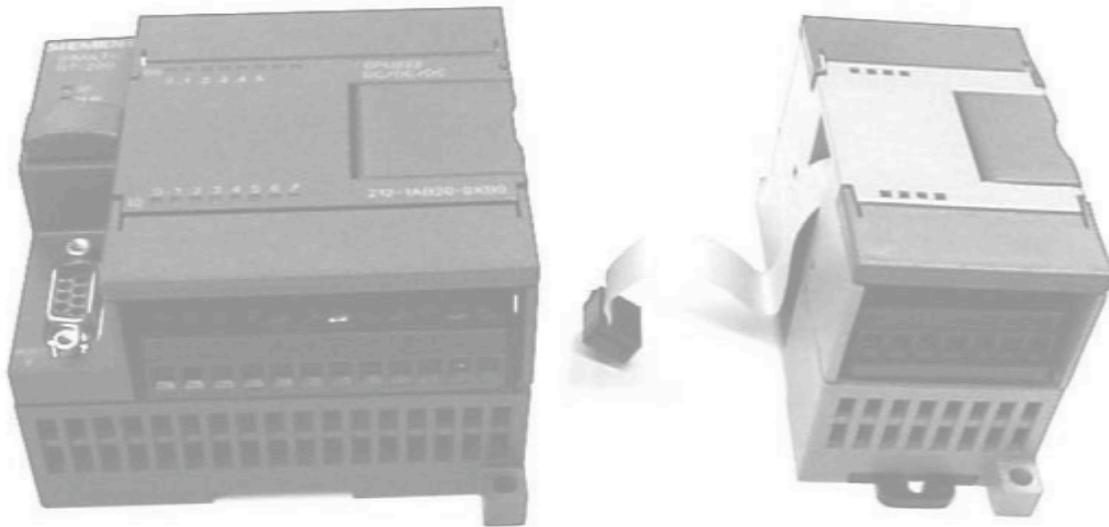
Input/output modules

prácticas

- Se pueden considerar semi-modulares: Fuente de alimentación+CPU+E/S. Se pueden añadir módulos adicionales³⁷

Ejemplos

■ Módulos



CPU 224



EM221



- EM 221, módulo que permite añadir 8 entradas digitales

Ejemplos

	CPU 221	CPU 222	CPU 224
E/S integradas	6 DI/4 DO	8 DI/6 DO	14 DI/16 DO
Memoria programa	4 KB	4 KB	8 KB
Memoria de datos	2 KB	2 KB	8 KB
Puerto de comunicaciones	RS-485	RS-485	RS-485

- Nos permite elegir la versión que se adapte mejor al proceso a controlar.
- Debemos tener en cuenta si el proceso puede necesitar más E/S.
- También se debe tener en cuenta la capacidad de memoria
 - P.e. cpu 224. 1 inst=2bytes→4000 instrucciones...

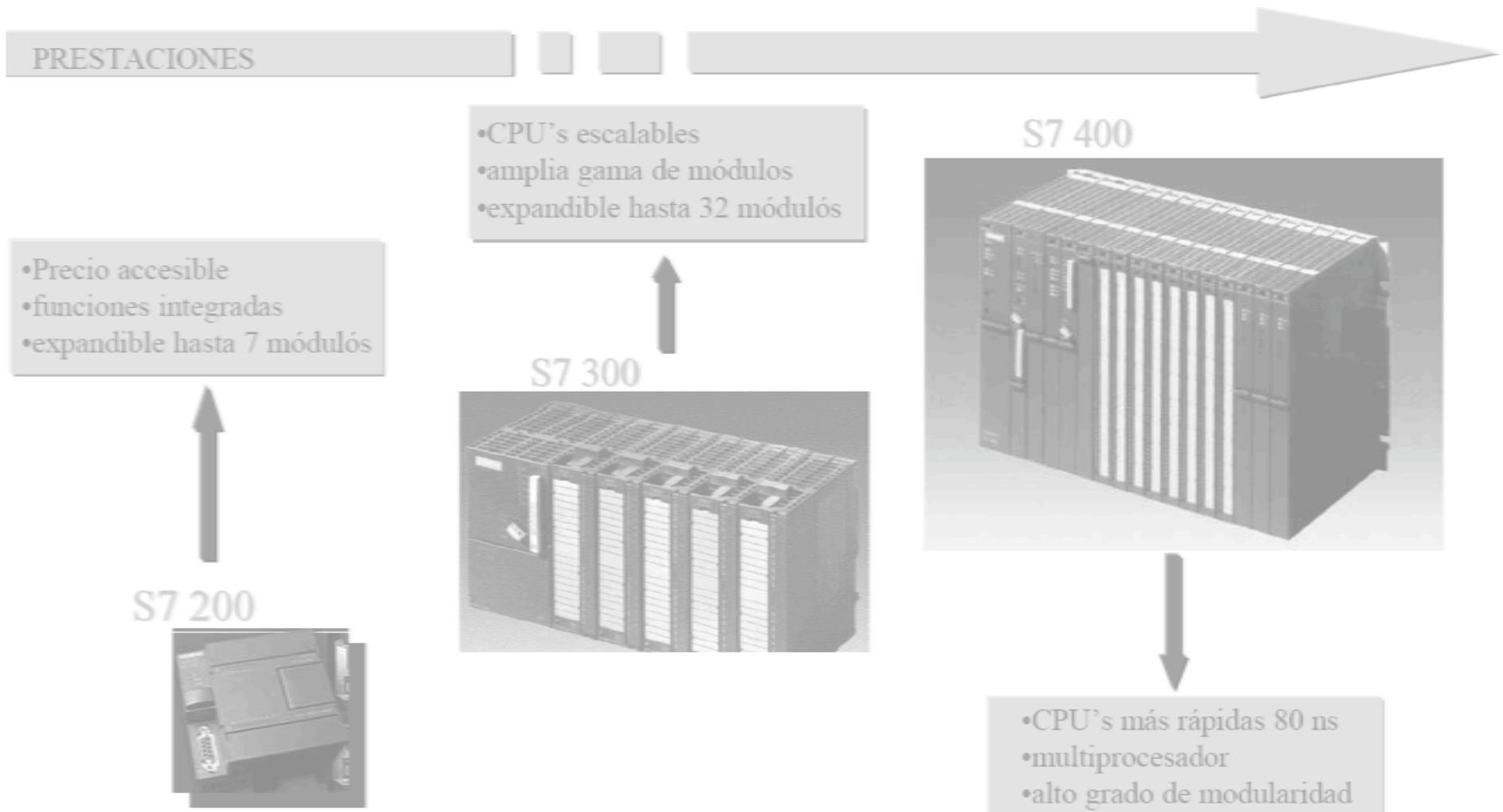
Ejemplos

- Periféricos: Paneles de control



Ejemplos

- Gama de Autómatas Siemens



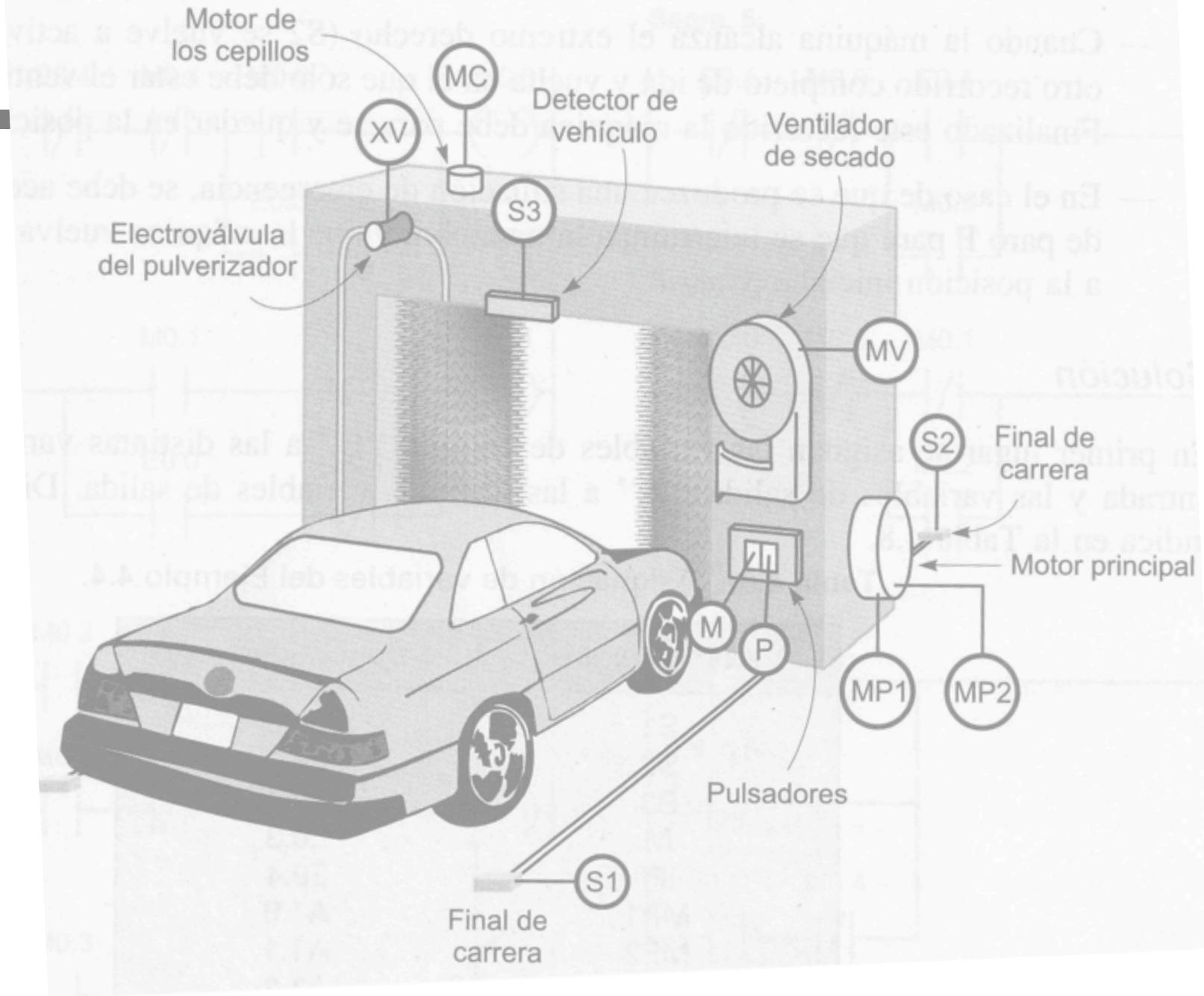
Arquitectura del autómata programable

Bloques que forman un autómata programable

- Introducción
 - Estructura externa del autómata
 - Bloques que forman un autómata programable
- Arquitectura interna de un autómata programable
 - Unidad central de proceso
 - Elementos que integran la CPU
 - Memoria
 - Tipos de memorias
 - Memorias internas
 - Memoria de programa
 - Interfaces de entrada salida
 - Ejemplo



Ejemplo: conexión a un



-
- Descripción del sistema: El sistema cuenta con los siguientes elementos.
 - **MP**, motor principal que mueve la estructura (en dos direcciones izquierda y derecha).
 - **MC**, motor que mueve los cepillos de limpieza.
 - **MV**, motor del ventilador de aire, para secado del coche.
 - **XV**, Una electroválvula, que permite la salida del líquido de limpieza.
 - **S1** y **S2**, dos finales de carrera (interruptores), que detectan cuando la estructura llega al final de su movimiento.
 - **S3**, un sensor que detecta la presencia del vehículo.
 - **M**, botón de marcha.
 - **P**, botón de parada de emergencia.

- Repasamos... diseño del automatismo:
 - Lógica programada (autómata) o lógica cableada.
 - Sup. que queremos utilizar un autómata.
 - ¿Cuál elegimos?
 - ¿Qué debemos tener en cuenta?

Número de entradas?

Precio?

Número de salidas?

Conexión con periféricos?

Selección del autómata

Capacidad para manejar salidas en AC y DC?

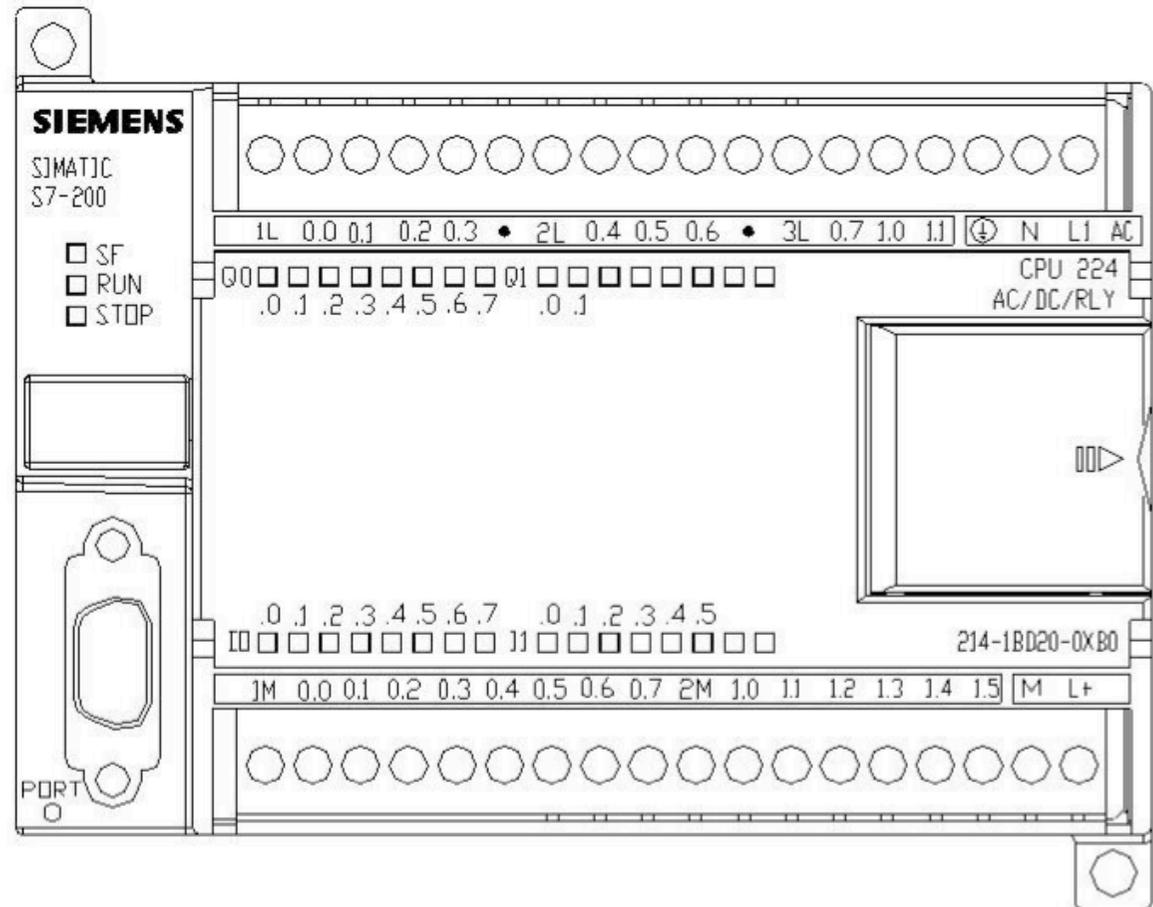
Añadir módulos E/S?

Puede almacenar el programa de usuario?

E/S analógicas?

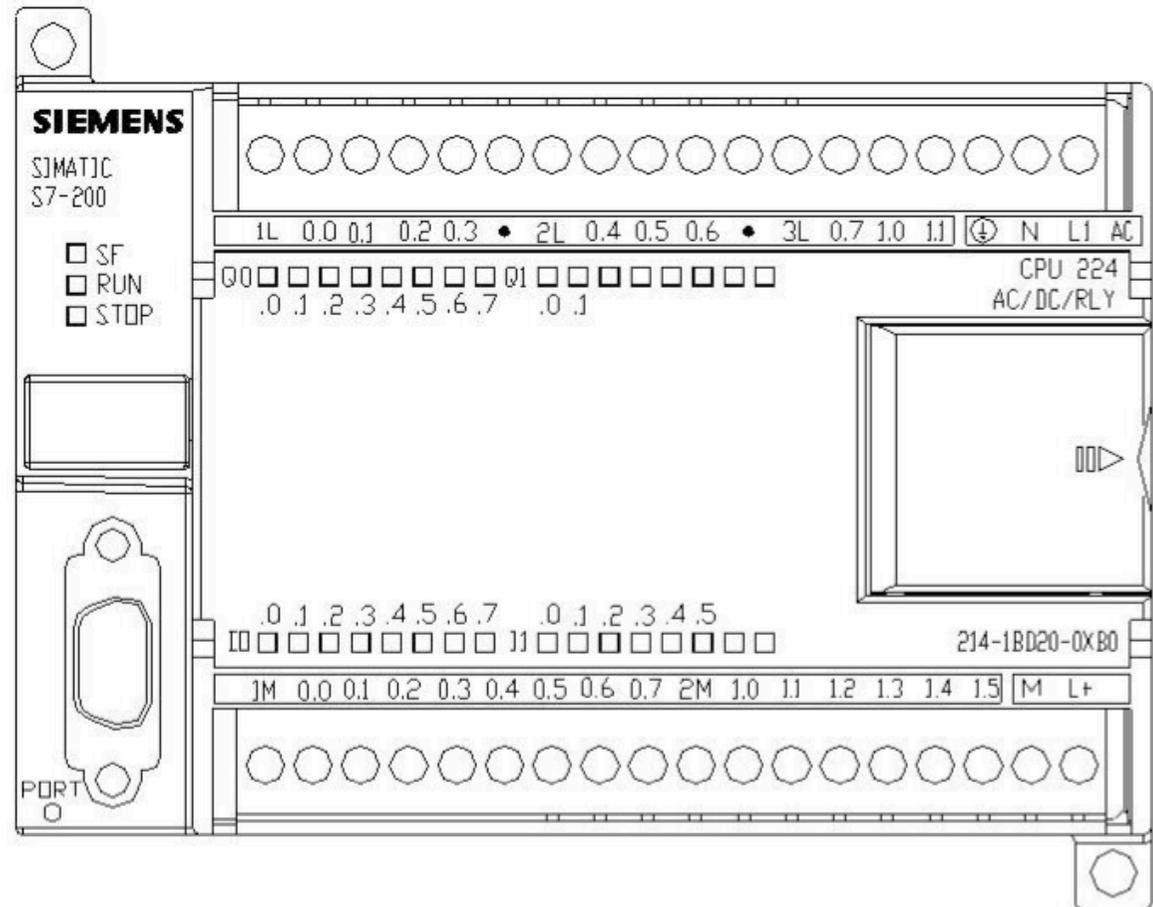
- Conexión al proceso:

ENTRADAS AUT.	SALIDAS AUT.



- Conexión al proceso:

ENTRADAS AUT.	SALIDAS AUT.
S1	MP1, MP2 (por medio de un relé)
S2	MC (por medio de un relé)
S3	XV
M	
P	



- ¿Cómo asignamos las E/S?

ENTRADAS disponibles AUT.	Conectado a
I0.0	
I0.1	
I0.2	
I0.3	
I0.4	

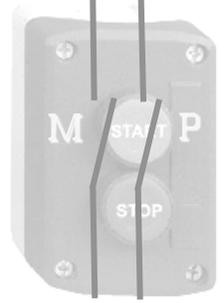
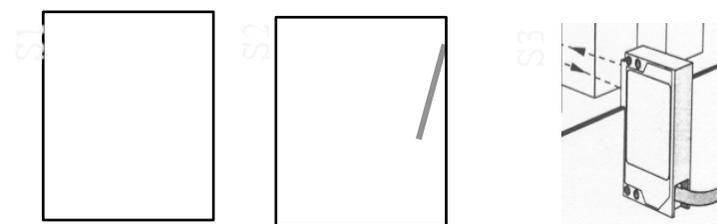
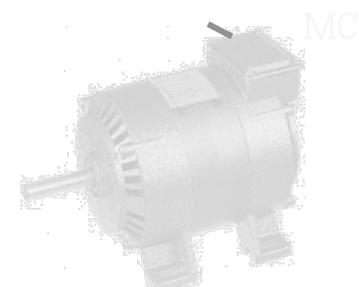
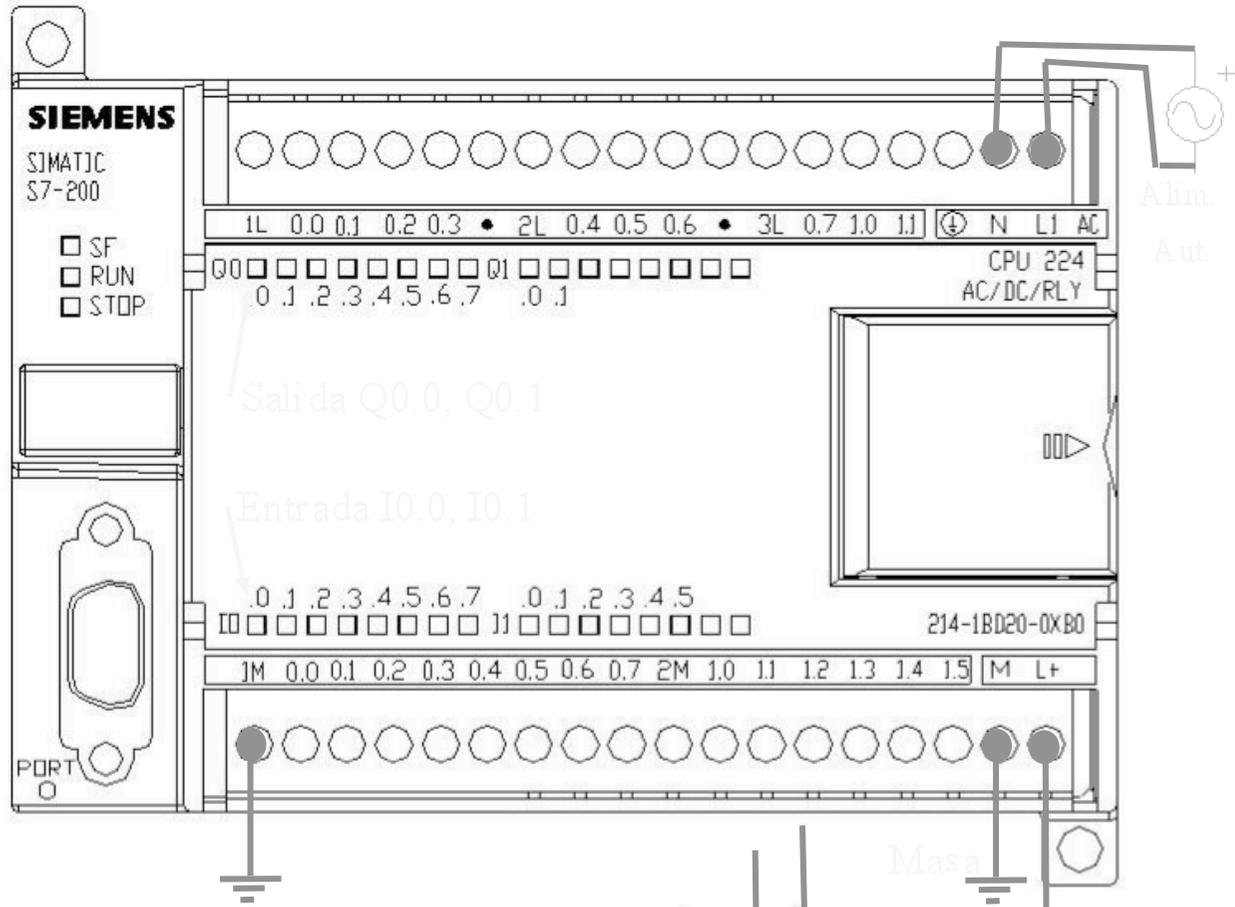
SALIDAS disponibles AUT.	Conectado a
Q0.0	
Q0.1	
Q0.2	
Q0.3	

- ¿Cómo asignamos las E/S?

ENTRADAS disponibles AUT.	Conectado a
I0.0	S1
I0.1	S2
I0.2	S3
I0.3	M
I0.4	P

SALIDAS disponibles AUT.	Conectado a
Q0.0	MP1
Q0.1	MP2
Q0.2	MC
Q0.3	XV

- ¿Cómo cableamos las E/S con el proceso?



24 VDC

