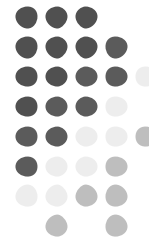


# Tema 3

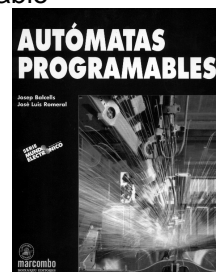


## Principios de programación. Visión práctica con Simatic S7



## Bibliografía

- Título: "Autómatas Programables"
  - Autor: Josep Balcells, Jose Luis Romeral
  - Editorial: Marcombo
  - Año: 1997
  - Temas Parte II: El Autómata Programable



# Índice

---



- Introducción a la programación
- Organización de programas
- Programación lineal
- Programación estructurada
- Tipos de módulos
- Bit, Byte, Palabra, Doble palabra
- Direccionamiento
- Elaboración cíclica del programa
- Procesamiento de entradas y salidas
- Imágenes de proceso
- Organización de la secuencia de trabajo
- Tiempo de ciclo
- Sistemas de representación del programa



# Índice

---



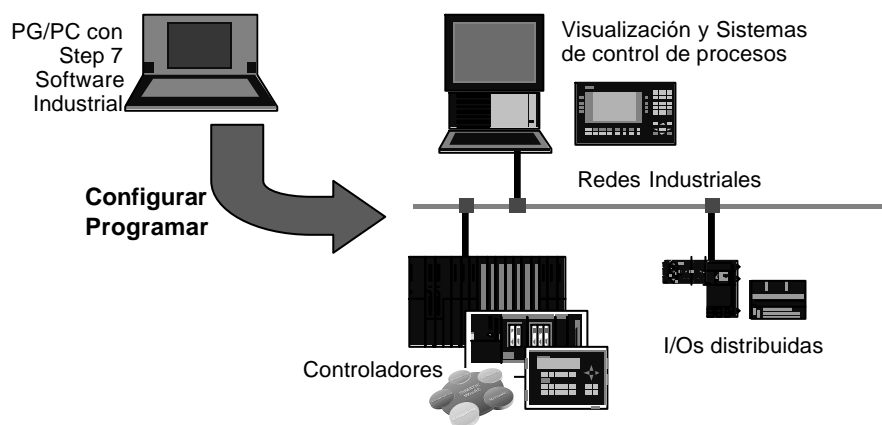
- Introducción a la programación
- Organización de programas
- Programación lineal
- Programación estructurada
- Tipos de módulos
- Bit, Byte, Palabra, Doble palabra
- Direccionamiento
- Elaboración cíclica del programa
- Procesamiento de entradas y salidas
- Imágenes de proceso
- Organización de la secuencia de trabajo
- Tiempo de ciclo
- Sistemas de representación del programa



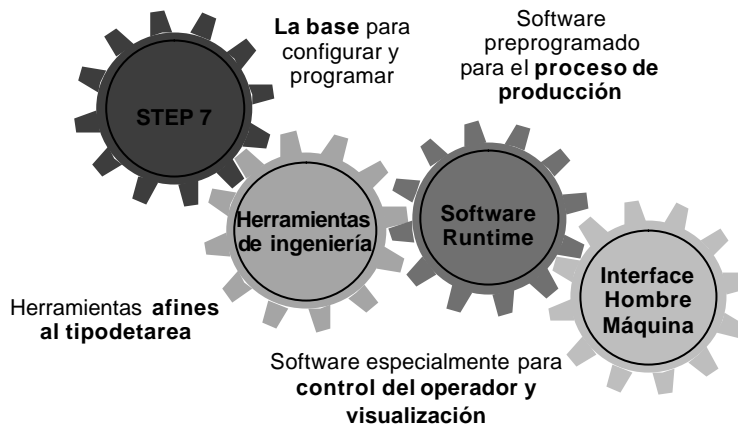
# Introducción

- ¿Qué es STEP 7?
  - Es el software estándar de SIMATIC para crear programas PLC en los lenguajes de programación KOP, FUP o AWL para dispositivos de control SIMATIC S7-300 y S7-400.
- ¿Por qué?
  - Es la herramienta que se utilizará para la implementación de las tareas de comunicación

# Introducción



# Introducción



Principios de programación. Visión práctica con Simatic S7

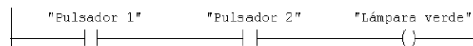
7

# Introducción

- STEP 7 permite crear programas PLC en los lenguajes de programación KOP, AWL o FUP. En la práctica debe decidirse por uno de los tres lenguajes.

## KOP (esquema de contactos)

apropiado, entre otros, para usuarios que provienen de la industria electrotécnica



## AWL (lista de instrucciones)

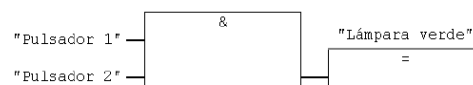
apropiada, entre otros, para usuarios que provienen del campo de la informática

```

U   "Pulsador 1"
U   "Pulsador 2"
=   "Lámpara verde"
    
```

## FUP (diagrama de funciones)

apropiado, entre otros, para usuarios que estén familiarizados con la técnica de circuitos



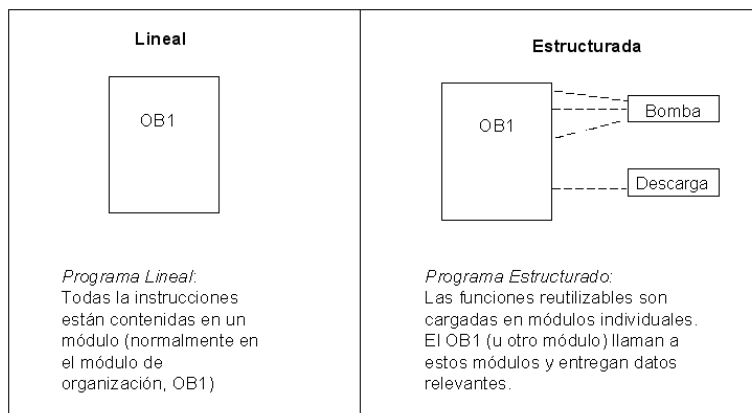
Principios de programación

# Índice

- Introducción a la programación
- Organización de programas
- Programación lineal
- Programación estructurada
- Tipos de módulos
- Bit, Byte, Palabra, Doble palabra
- Direccionamiento
- Elaboración cíclica del programa
- Procesamiento de entradas y salidas
- Imágenes de proceso
- Organización de la secuencia de trabajo
- Tiempo de ciclo
- Sistemas de representación del programa

# Organización de programas

- STEP 7 proporciona dos posibilidades para organizar un programa. Basado en estas líneas-guía, puede decidir cuál es el procedimiento más apropiado para proyectar su programa.



## Organización de programas



- **Programa Lineal**
  - El programa completo se localiza en un módulo (OB1) con todas las instrucciones juntas. Este modelo es similar al esquema de relés fijo reemplazado por el controlador lógico programable. El sistema procesa sucesivamente las instrucciones individuales.
- **Programa Dividido**
  - El programa se divide en módulos, con lo que cada bloque contiene las operaciones lógicas de un cierto grupo de dispositivos o tareas. Las instrucciones integradas en el módulo de organización OB1 determina la secuencia en que se procesan los módulos. Un programa dividido puede, por ejemplo, contener módulos de instrucciones con los que se controlan los modos de operación individuales para un proceso industrial.



## Organización de programas



- **Programa Estructurado**
  - Un programa estructurado contiene módulos con parámetros (módulos parametrizables). Estos módulos se programan de forma que se puedan usar universalmente. Durante la llamada se especifican los parámetros actuales (las direcciones concretas de las entradas y salidas).
  - Un ejemplo de módulo parametrizable:
    - Un "módulo bomba" contiene las instrucciones para una bomba y una serie de entradas y salidas que cualquier otra bomba puede usar en un proceso.
    - Los módulos de instrucciones lógicas, son los responsables de controlar la bomba concreta, llamar (abrir) el "módulo bomba" y proporcionar información sobre qué bomba debe ser controlada.
    - Cuando el "módulo bomba" concluya de realizar sus instrucciones, el programa retorna al módulo que le llamó (por ejemplo, OB1) el cual continúa procesando sus instrucciones.



# Índice

- Introducción a la programación
- Organización de programas
- Programación lineal
- Programación estructurada
- Tipos de módulos
- Bit, Byte, Palabra, Doble palabra
- Direccionamiento
- Elaboración cíclica del programa
- Procesamiento de entradas y salidas
- Imágenes de proceso
- Organización de la secuencia de trabajo
- Tiempo de ciclo
- Sistemas de representación del programa



# Programación lineal

- Antes de diseñar un programa es necesario entender los diferentes modos en que se puede estructurar, distinguir los distintos tipos de módulo utilizables así como la forma de trabajo del PLC.
- Estructura Programa: La mayoría de los PLC permiten diseñar el programa de forma lineal o estructurada..
- Programación Lineal
  - Todo el programa está contenido en un único módulo.
  - Normalmente, este módulo es el OB1 (módulo de elaboración cíclica). Las instrucciones se van ejecutando secuencialmente una detrás de otra.
  - Este sistema de programación sólo está recomendado en proyectos pequeños.

OB 1

Nivel 0



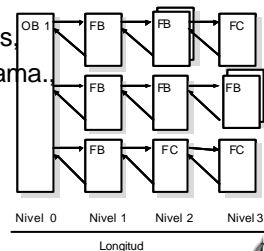
# Índice

- Introducción a la programación
- Organización de programas
- Programación lineal
- Programación estructurada
- Tipos de módulos
- Bit, Byte, Palabra, Doble palabra
- Direccionamiento
- Elaboración cíclica del programa
- Procesamiento de entradas y salidas
- Imágenes de proceso
- Organización de la secuencia de trabajo
- Tiempo de ciclo
- Sistemas de representación del programa

# Programación estructurada

- Programación **estructurada**:
  - La programación estructurada implica la división del programa en diversos bloques. Cada uno de estos bloques constituye al mismo tiempo un subprograma. Por lo tanto, se recomienda utilizar este tipo de programación cuando en el programa haya partes que se repitan.
- Ventajas de la programación estructurada
  - Simplificación y claridad de programas largos.
  - Posibilidad de estandarizar partes del programa.
  - Facilidad en la modificación
  - Simplificación del test
  - Simplificación del comienzo

Programación estructurada



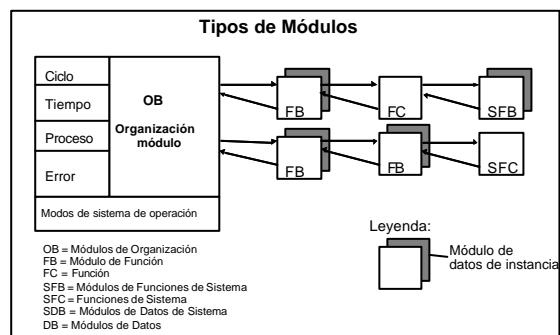


# Índice

- Introducción a la programación
- Organización de programas
- Programación lineal
- Programación estructurada
- Tipos de módulos
- Bit, Byte, Palabra, Doble palabra
- Direccionamiento
- Elaboración cíclica del programa
- Procesamiento de entradas y salidas
- Imágenes de proceso
- Organización de la secuencia de trabajo
- Tiempo de ciclo
- Sistemas de representación del programa

# Tipos de módulos

- Módulos de sistema
  - Los módulos de sistema son funciones predefinidas o módulos integrados en el sistema operativo de la CPU.
- Módulos de usuario



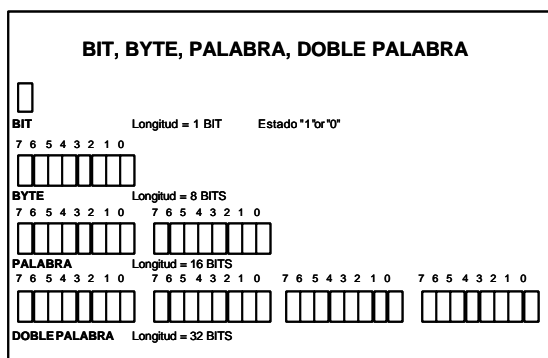
## Tipos de módulos

- **Módulos**
  - La programación estructurada permite la utilización de diferentes tipos de módulos, en los que el usuario puede programar o almacenar datos.
- **Módulo de organización**
  - **Los módulos de organización (OBs)** constituyen los módulos ejecutables del sistema. Todo el programa podrá ser almacenado en el módulo OB1 (módulo de elaboración cíclica). Por otra parte, el programa podrá ser almacenado en distintos bloques, en cuyo caso el módulo OB1 es utilizado para realizar la llamada del resto de los módulos en el momento que se cumpla una determinada condición.
- **Módulo de función FB**
  - Un módulo de función (**FB**) es un bloque que contiene una parte del programa y que controla una determinada área de la memoria. Este módulo ofrece la posibilidad de utilizar parámetros. Estos módulos se emplean para tareas repetitivas o funciones complejas.
- **Función FC**
  - Una **función (FC)** es, de acuerdo a la norma **IEC 1131-3** un módulo de datos estáticos. Te ofrece la posibilidad de transferir datos al programa de usuario.
- **Instrucciones**
  - Todas las operaciones son utilizables en todos los módulos (**FB, FC y OB**).
- **Módulos de DB**
  - Los **módulos de datos (DB)** constituyen áreas de datos en el programa de usuario. Sólo contiene datos.

## Índice

- Introducción a la programación
- Organización de programas
- Programación lineal
- Programación estructurada
- Tipos de módulos
- Bit, Byte, Palabra, Doble palabra
- Direccionamiento
- Elaboración cíclica del programa
- Procesamiento de entradas y salidas
- Imágenes de proceso
- Organización de la secuencia de trabajo
- Tiempo de ciclo
- Sistemas de representación del programa

# Bit, Byte, Palabra, Doble palabra

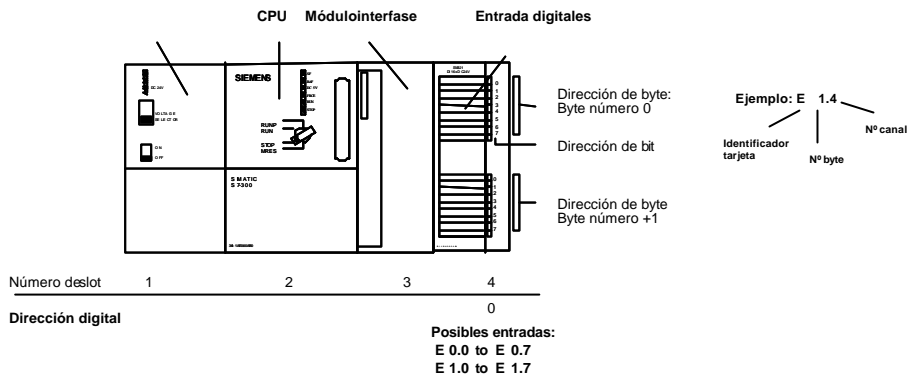


# Índice

- Introducción a la programación
- Organización de programas
- Programación lineal
- Programación estructurada
- Tipos de módulos
- Bit, Byte, Palabra, Doble palabra
- Direccionamiento
- Elaboración cíclica del programa
- Procesamiento de entradas y salidas
- Imágenes de proceso
- Organización de la secuencia de trabajo
- Tiempo de ciclo
- Sistemas de representación del programa

# Direccionamiento

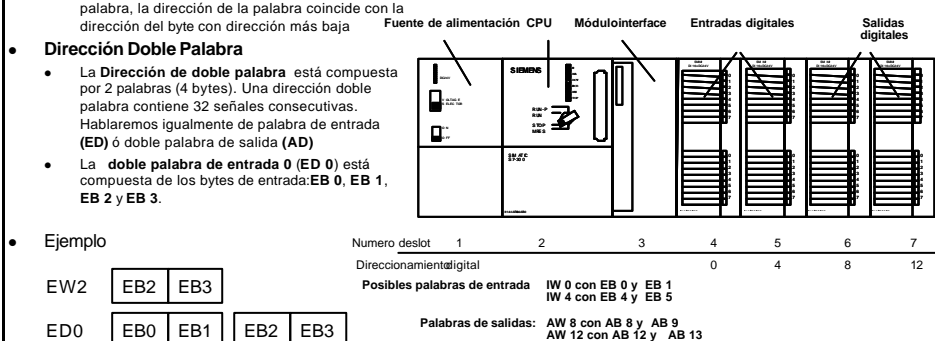
## Direccionamiento (I)



# Direccionamiento

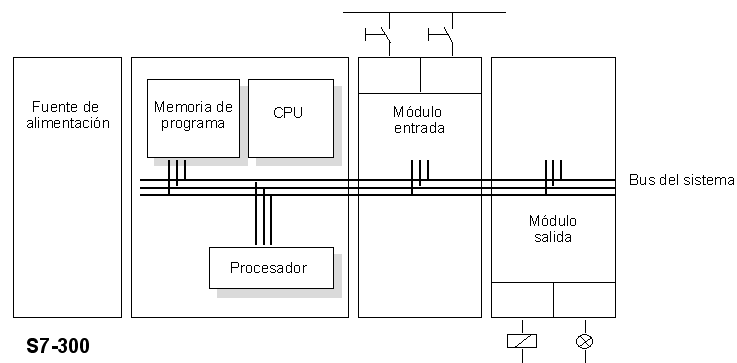
- **Dirección Palabra**
  - Si combinamos dos direcciones de byte, obtenemos una **Dirección de Palabra**. Por ejemplo, dos bytes de palabra de entrada, constituyen una palabra de entrada (**EW**). La palabra de entrada 2 (**EW 2**) está compuesta de los bytes de entrada 2 y 3 (**EB 2 y EB 3**). En una palabra, la dirección de la palabra coincide con la dirección del byte con dirección más baja
- **Dirección Doble Palabra**
  - La **Dirección de doble palabra** está compuesta por 2 palabras (4 bytes). Una dirección doble palabra contiene 32 señales consecutivas. Hablaremos igualmente de palabra de entrada (**ED**) ó doble palabra de salida (**AD**)
  - La **doble palabra de entrada 0 (ED 0)** está compuesta de los bytes de entrada: **EB 0, EB 1, EB 2 y EB 3**.

## Direccionamiento (II)



## Direccionamiento

- **Dirección del bus**
  - Un bus permite el intercambio entre la CPU y los módulos de la estructura del PLC. Podemos considerar un bus como varias líneas de señales en paralelo a través de todo el sistema.
- **Rack**
  - En una estructura modular de un PLC podemos encontrar un "rack", que permite considerar al PLC como un sistema totalmente compacto.



S7-300

25

## Direccionamiento

- **Direccionamiento fijo**
  - El término **direccionamiento fijo** es empleado cuando el direccionamiento está establecido previamente. Podemos encontrar este tipo de direccionamiento en los autómatas compactos S7200 y S7300 (CPUs con periferia integrada: 312-IFM y 314-IFM)
  - En los sistemas modulares, el direccionamiento viene establecido por la posición de slot. Por ejemplo, el slot próximo a la CPU tiene una dirección ED 0 ó AD 0, dependiendo del tipo de módulo conectado.
- **Direccionamiento variable**
  - El direccionamiento variable podemos encontrarlos en los autómatas de la gama alta (S7-400).
  - Los módulos pueden ser direccionados de forma independiente a su posición en la estructura lineal del PLC. La dirección viene establecida a través de la herramienta de configuración del Hardware del STEP 7.

# Direccionamiento

**Configuración de Direcciones de E/A**

**Dirección digital (byte de dirección 0 - 127)**

CPU y fuente de alimentación	IM (emisor)	0.0	4.0	8.0	12.0	16.0	20.0	24.0	28.0
		a 3.7	a 7.7	a 11.7	a 15.7	a 19.7	a 23.7	a 27.7	a 31.7

IM (receptor y fuente de alimentación)	32.0	36.0	40.0	44.0	48.0	52.0	56.0	60.0
	a 35.7	a 39.7	a 43.7	a 47.7	a 51.7	a 55.7	a 59.7	a 63.7

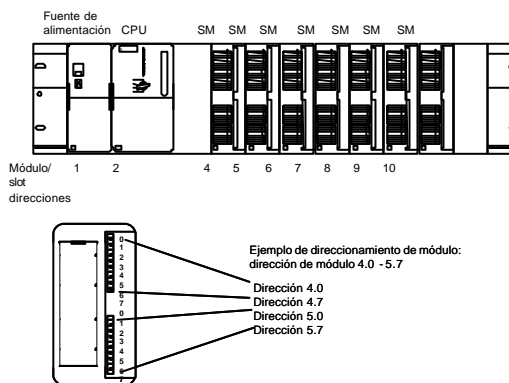
las restantes direcciones digitales se dividen en grupo de cuatro bytes sucesivos por módulo desde 64,0 a 127,7 (16 módulos adicionales en dos racks adicionales).

**Direcciones analógicas (byte de dirección 256 a 752)**

CPU y fuente de alimentación	IM	256	272	288	304	320	336	352	368
		a 271	a 287	a 303	a 319	a 335	a 351	a 367	a 383

# Direccionamiento

- Los números de slot en el rack de un S7-300 hace más fácil determinar la dirección esquemática dentro del entorno S7-300. La primera dirección del módulo se especifica por su ubicación en el rack.
  - Slot 1: Fuente de alimentación. Por defecto, está en el primer slot. No hay dirección de módulo asignada a la fuente de alimentación.
  - Slot 2: CPU. Debe estar ubicada junto a la fuente de alimentación. No está asignada dirección de módulo a la CPU.
  - Slot 3: Módulo interfase (IM). Para la conexión de un rack de expansión. No está asignada dirección de módulo a la IM. Incluso si la IM no está presente, debe ser considerada en el esquema de direccionamiento para los slots. El slot 3 lógicamente está reservado para la IM dentro de la CPU.
  - Slots 4-11: Módulos de señal. El slot 4 es aplicable como primer slot para módulos de E/A. El direccionamiento esquemático, visualizado en la figura inferior, empieza con el primer módulo de E/A e incrementa dependiendo del tipo del módulo.



# Direccionamiento

## Volumen Posible de Direccionamiento para STEP 7

Área de direc. Desig.	Acceso al área	Abrev.	Máx. direcc. área
<b>Imagen de proceso E/A</b>	entrada / salida bit	E / A	0.0 a 65,535.7
	entrada / salida byte	E / AB	0 a 65,535
	entrada / salida palabra	EW / AW	0 a 65,534
	entrada / salida doble palabra	ED / AD	0 a 65,532
<b>Marcas</b>	bit de memoria	M	0.0 a 255.7
	byte de memoria	MB	0 a 255
	palabra de memoria	MW	0 a 254
	doble palabra de memoria	MD	0 a 252
<b>E/A externa entrada/salida</b>	byte E/A, periferia	PEB / PAB	0 a 65,535
	palabra E/A, periferia	PEW/PAW	0 a 65,534
	doble palabra E/A, periferia	PED/PAD	0 a 65,532
	<b>Temporizador</b>	Temporizador (T)	T
Contador (C)		C	0 a 255
<b>Módulo de datos</b>	Bit,byte,palabra,doble palabra	DBX,DBB DBW,DBD	0 a 65,532
	<b>Módulo de datos de instancia</b>	Bit,byte,palabra,doble palabra	DIX,DI8 DIW,DI16

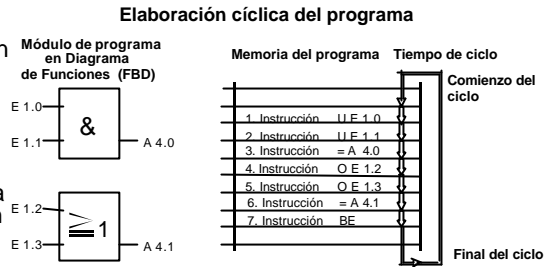
# Índice

- Introducción a la programación
- Organización de programas
- Programación lineal
- Programación estructurada
- Tipos de módulos
- Bit, Byte, Palabra, Doble palabra
- Direccionamiento
- Elaboración cíclica del programa
- Procesamiento de entradas y salidas
- Imágenes de proceso
- Organización de la secuencia de trabajo
- Tiempo de ciclo
- Sistemas de representación del programa

# Elaboración cíclica del programa

## • Elaboración cíclica del programa

- La programadora se encarga de traducir esta información en un conjunto de instrucciones legible para el PLC. Cada instrucción está localizada en una determinada área de la memoria de programa.
- El procesador del PLC elabora el programa usuario escrito en la memoria de forma continuada y repetitiva. Esto es lo que conocemos como elaboración cíclica del programa



# Elaboración cíclica del programa

## • Imagen de proceso

- La imagen de proceso constituye el espejo del estado de la periferia de entrada y salida. Se localiza en una determinada área de la memoria de la CPU

## • Imagen de proceso de entrada (PAE)

- La imagen de proceso de entrada se genera al comienzo del ciclo. El procesador escanea los estados de señal de la periferia de entrada generando con esta información la imagen de proceso de entradas (PAE)

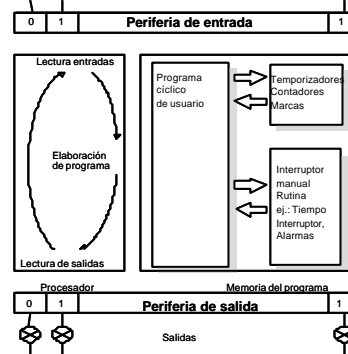
## • Imagen de proceso de salida (PAA)

- La imagen de proceso de salida (PAA) se genera durante la elaboración del programa de usuario. En el mismo instante en el que se elabora la última instrucción del programa de usuario, se transfiere la imagen de proceso de salida (PAA) a la periferia.

- Mientras se elabora el programa de usuario, podemos afirmar que nuestro PLC está aislado de la periferia.

- Esto no es completamente cierto ya que, como veremos más adelante, existe la posibilidad de hacer llamadas directas a la periferia en cualquier momento de elaboración del programa.

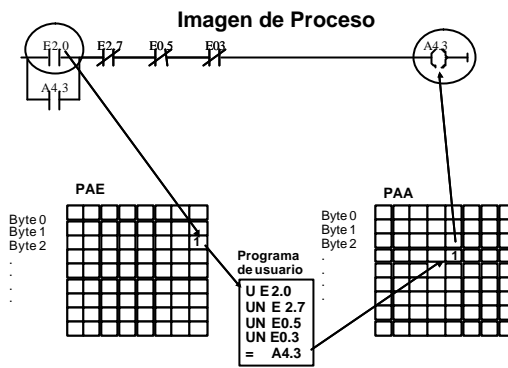
## Procesamiento de entradas y salidas





# Elaboración cíclica del programa

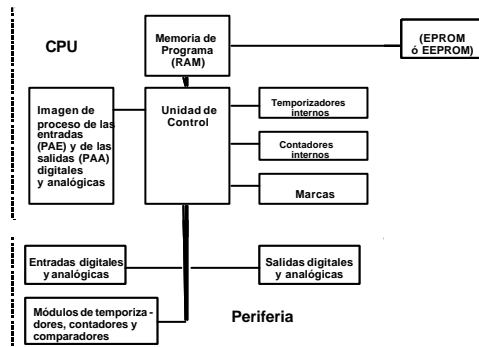
- La CPU pregunta el estado de las entradas y las salidas en cada ciclo. Hay ciertas áreas de memoria en las cuales se almacenan los datos binarios de los módulos: PAE y PAA. El programa accede a estos registros durante el proceso.
- **PAE**
  - La tabla de la imagen de proceso de entradas está localizada en el área de memoria de la CPU y el estado de señal de todas las entradas se almacena allí.
- **PAA**
  - La tabla de la imagen de proceso de salidas contiene los valores de salida resultado del proceso del programa. Al final de el ciclo, estos valores de salida son enviados a las salidas reales (A).
- **Programa de Usuario**
  - Cuando examina las entradas en el programa de usuario con U E2.0, por ejemplo, lo que se evalúa es el último estado en la PAE. Esto garantiza que para todas las consultas de entradas en un mismo ciclo se utiliza el mismo estado de señal.



# Organización de la secuencia de trabajo

- **ORGANIZACIÓN DE LA SECUENCIA DE TRABAJO**
  - Un programa se compone de una secuencia de instrucciones que se almacenan luego en la memoria de programa interna de la unidad central (CPU).
  - Además de la memoria interna, el autómata programable contiene también la Unidad de control y la periferia que son necesarias para que pueda cumplir su función.
  - Los elementos que integran la CPU son: Unidad de control, memoria, imagen de proceso, temporizadores internos, contadores internos y marcas (memorias intermedias).

## Organización de la secuencia de trabajo



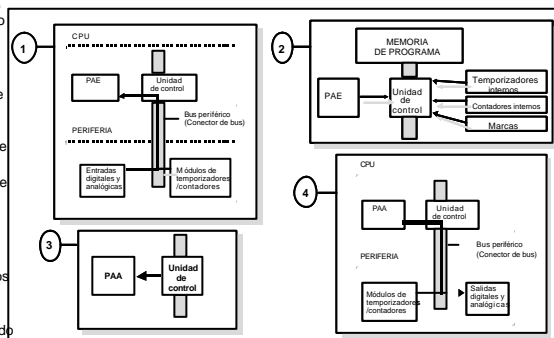
# Índice

- Introducción a la programación
- Organización de programas
- Programación lineal
- Programación estructurada
- Tipos de módulos
- Bit, Byte, Palabra, Doble palabra
- Direccionamiento
- Elaboración cíclica del programa
- Procesamiento de entradas y salidas
- Imágenes de proceso
- Organización de la secuencia de trabajo
- Tiempo de ciclo
- Sistemas de representación del programa

# Organización de la secuencia de trabajo

Procesamiento cíclico:

- 1) El autómata programable funciona cíclicamente, esto es, una vez finalizado un recorrido completo del programa, comienza a procesar nuevamente su primera instrucción.
- 2) La unidad de control comienza el procesamiento del programa. La unidad de control toma de la memoria interna el programa (instrucción a instrucción) y las ejecuta. Para ello establece combinaciones, efectúa cálculos con los datos de la PAE y considera los estados de los temporizadores y de los contadores, así como de las marcas.
- 3) La unidad de control deposita el resultado del procesamiento del programa en la imagen de proceso de las salidas "PAA"
- 4) La unidad de control sólo transfiere a los módulos de salida, temporizadores y contadores los estados de señal contenidos en la imagen de proceso de las salidas (PAA) cuando ha finalizado el recorrido del programa, esto es, al final de un ciclo. Ahora puede comenzar un nuevo ciclo.



# Índice

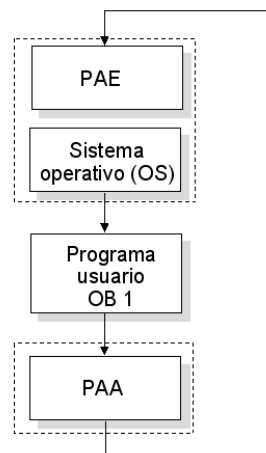
- Introducción a la programación
- Organización de programas
- Programación lineal
- Programación estructurada
- Tipos de módulos
- Bit, Byte, Palabra, Doble palabra
- Direccionamiento
- Elaboración cíclica del programa
- Procesamiento de entradas y salidas
- Imágenes de proceso
- Organización de la secuencia de trabajo
- Tiempo de ciclo
- Sistemas de representación del programa



# Tiempo de Ciclo

- **Tiempo de scan**
  - Se define como tiempo de scan el tiempo de elaboración del ciclo.
  - Comprende los siguientes procesos:
    - Lectura de la periferia de entrada y creación de la PAE (Imagen de proceso de entrada).
    - Elaboración del programa
    - Transferencia de la imagen de proceso de salida a la periferia de salida.
    - Tiempo de funcionamiento del sistema.
- **Tiempo de respuesta**
  - Se define como tiempo de respuesta el tiempo correspondiente a la lectura de una señal de entrada y la actuación sobre la salida correspondiente.
  - El tiempo de respuesta comprende:
    - Tiempo de espera al inicio del tiempo de scan
    - Tiempo de lectura de las entradas
    - Tiempo de transferencia de la imagen de proceso
    - Tiempo de funcionamiento del sistema
    - Tiempo de elaboración del programa de usuario.
    - Comunicación con la interfase multipunto (MPI)

## Tiempo de ciclo

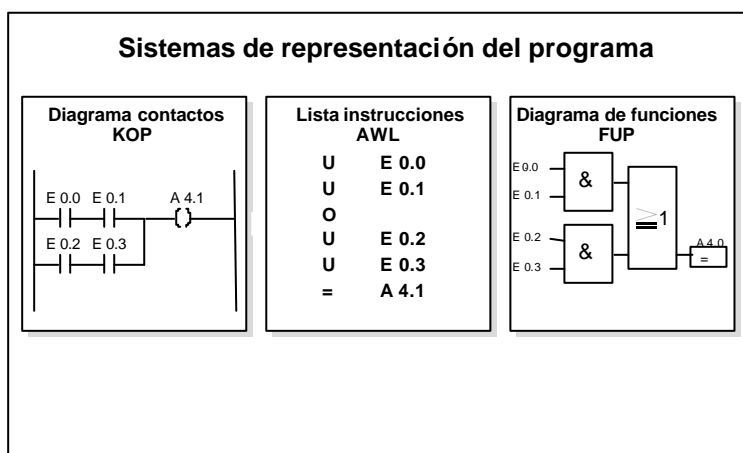


# Índice

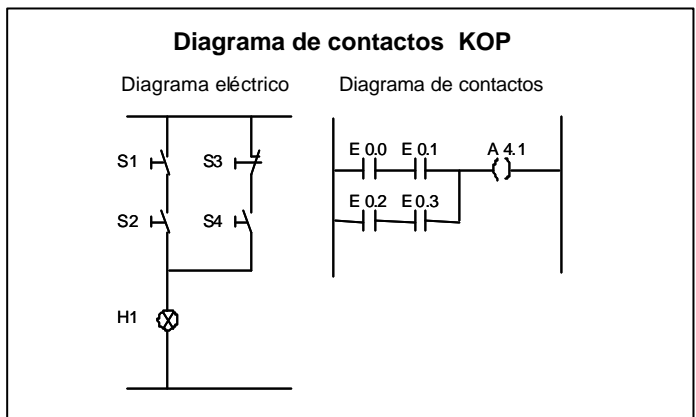
- Introducción a la programación
- Organización de programas
- Programación lineal
- Programación estructurada
- Tipos de módulos
- Bit, Byte, Palabra, Doble palabra
- Direccionamiento
- Elaboración cíclica del programa
- Procesamiento de entradas y salidas
- Imágenes de proceso
- Organización de la secuencia de trabajo
- Tiempo de ciclo
- Sistemas de representación del programa

# Sistemas de representación del programa

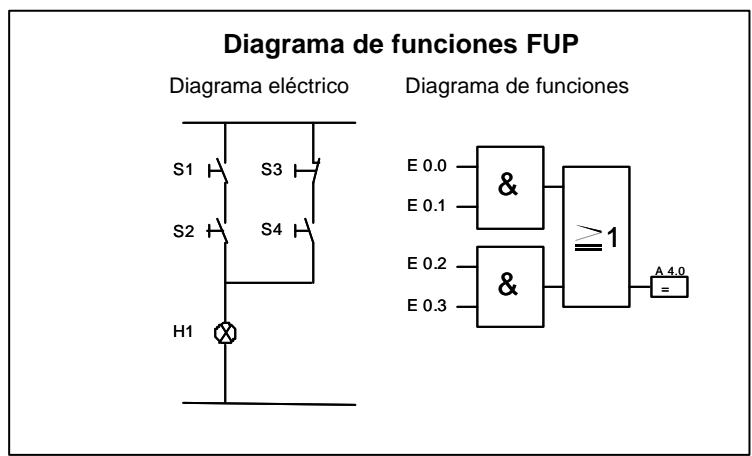
## Sistemas de representación del programa



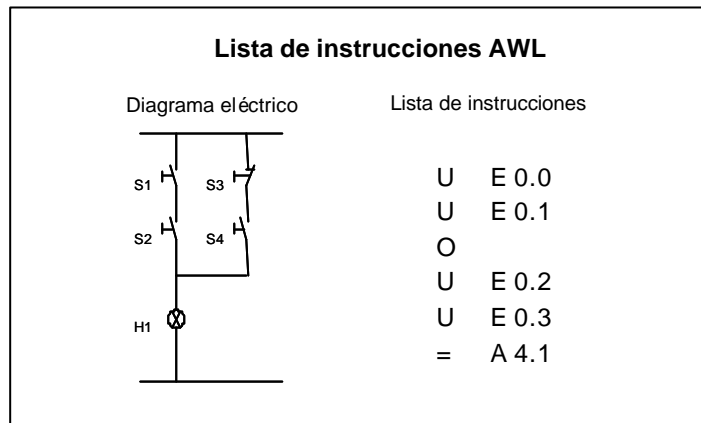
# Sistemas de representación del programa



# Sistemas de representación del programa



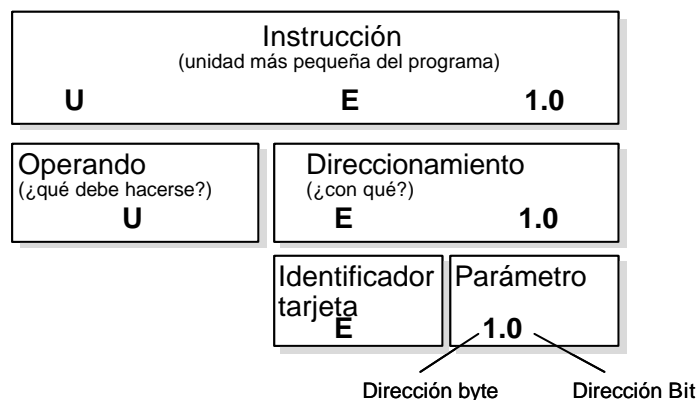
# Sistemas de representación del programa



# Sistemas de representación del programa



## Concepto de Instrucción



## Sistemas de representación del programa



- La programación no obliga a utilizar un lenguaje común para todo el proyecto.
- La decisión no es pues **qué** lenguaje de programación utilizar, sino **cuando** utilizar cada uno.
- Para conocer dicha respuesta, deberemos aprender primero las ventajas de cada uno de ellos (y también los inconvenientes).



## Sistemas de representación del programa



- **VENTAJAS DEL AWL**, el más utilizado en Simatic:
  - Es el lenguaje que necesita menos instrucciones de programación de todos, con lo cual el programa ocupa igualmente menos código compilado. Esto permite optimizar la cantidad de memoria disponible en el PLC, pero sobre todo el tiempo de ciclo es menor, por lo que se pueden utilizar CPU's "lentas" para procesos relativamente rápidos, aún con programas considerables.
  - Es el más indicado para operaciones de saltos condicionales en función de valores analógicos. Cualquier tratamiento analógico que sobrepase la simple comparación es el terreno del AWL.
  - Permite introducir una gran cantidad de sentencias en la misma pantalla, con lo cual los test status no requieren de desplazamientos en la misma.



## Sistemas de representación del programa



- **INCONVENIENTES DEL AWL:**

- La programación de secuencias de pasos en procesos (set y reset de etapas) carece de sentido en este lenguaje, ya que no se gana memoria y el programa se convierte en ininteligible hasta para el propio programador.
- El programar una línea debajo de otra lleva a los programadores a implementar LINEAS Y LINEAS DE CODIGO SIN SER DIVIDIDAS EN SEGMENTOS Y SIN COMENTAR. Programas ininteligibles.



## Sistemas de representación del programa



- **VENTAJAS DEL KOP**

- Es muy sencillo ver en él los pasos de programa que no cumplen, y seguir las condiciones del proceso.
- Totalmente indicado para programadores más cercanos al mundo eléctrico que al informático en tareas de tratamiento digital (bobinas, set, reset...)

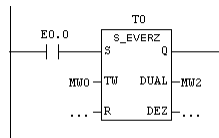




## Sistemas de representación del programa

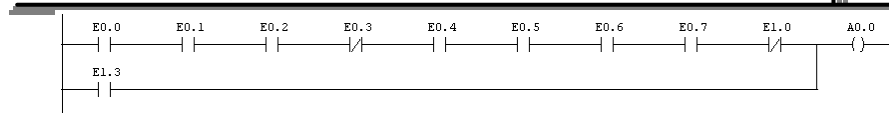
- INCONVENIENTES EN KOP

- Las series y paralelos requieren tanto espacio en pantalla que enseguida se nos salen de la misma, por lo que en status nos debemos de desplazar a menudo. La solución a dicho problema es sencilla: utilizar marcas.
- El realizar comparaciones de salto analógicas es misión imposible, a poco que se compliquen.
- Y el principal problema: las cajas de KOP necesitan una sistemática de proceso por parte del Step 7 que hace que no se optimice el código de las mismas, por lo que el programa haciendo lo mismo va más lento.



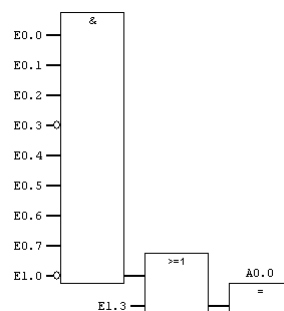
U	E	0.0
L	MW	0
SE	T	0
NOP	0	
L	T	0
T	MW	2
NOP	0	
NOP	0	

## Sistemas de representación del programa



- VENTAJAS DEL FUP

- Permite realizar gran cantidad de series y paralelos en la misma pantalla, con lo cual se acerca a la ventaja del AWL, pero con mayor claridad en el diagnóstico.
- Es el indicado para los programadores electrónicos.



## Sistemas de representación del programa



- INCONVENIENTES FUP
  - No es útil, al igual que le pasaba al KOP, para tratar valores analógicos ni condiciones de salto.
  - Sufre el mismo problema de optimización de código en el tratamiento que realiza del mismo el Step 7.
  - Los programadores de KOP suelen no identificar de una manera rápida las combinaciones *and* y *or* en un status de programa.



## Sistemas de representación del programa



- ¿En que lenguaje debo programar?
  - La respuesta es evidente, cada vez uno.
  - Para las tareas que no sean setear, resetear o activar bits, el AWL es sin dudas el lenguaje a utilizar. Las ventajas del mismo sobrepasan ampliamente los inconvenientes.
  - Sin embargo para todas las activaciones (series y contactos que van a parar a bobinas, ya sean enclavadas o no) la decisión debe de ser KOP o FUP, dependiendo del gusto del programador.
  - Si el programador tiene la misma destreza con KOP que con FUP. Decantarse por FUP tiene más ventajas.

