

## Tema 2: Programación de PLCs



### 1. STEP 7

### 2. PROGRAMACIÓN BÁSICA AWL

- Introducción
- Operaciones lógicas a nivel de bit
- Marcas
- Simbólicos
- Flancos
- Temporizadores
- Diagnósis e información del sistema
- Contadores
- **Carga y Transferencia**
- Operaciones con acumuladores
- Saltos

### 3. PROGRAMACIÓN ESTRUCTURADA

# Introducción a los Tipos de Datos en STEP 7



Tipos de datos elementales (hasta 32 bits)

- Tipos de Datos de bit (BOOL, BYTE, WORD, DWORD, CHAR)
- Tipos de Datos Matemáticos (INT, DINT, REAL)
- Tipos de Tiempo (S5TIME, TIME, DATE, TIME\_OF\_DAY)

Tipos de datos complejos (mayores de 32 bits)

- Tiempo (DATE\_AND\_TIME)
- Array (ARRAY)
- Estructura (STRUCT)
- Cadena de Caracteres (STRING)

Tipos de datos definidos por el usuario (mayores de 32 bits)

Tipo de Datos UDT (User Defined Type)

PLC's

## TIPOS DE DATOS DE BIT – TAMAÑO



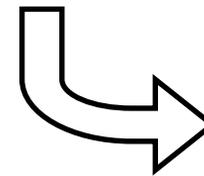
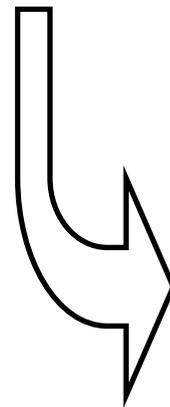
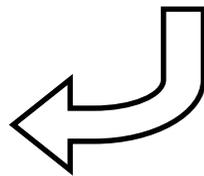
- La memoria del Autómata está estructurada en bytes.
- Estos bytes podemos agruparlos/descomponerlos, obteniendo los siguientes tamaños de datos:
  - Bit \_\_\_ .Nº
  - Byte (8 bits) \_\_\_ B
  - Palabra/word (16 bits) \_\_\_ W
  - Doble palabra/double word (32 bits) \_\_\_ D
- Se puede acceder a cualquier área de la memoria: marcas, entrada, salida, etc. con cualquiera de los tamaños de datos:



## Zona de memoria:

### E Tam BBB.b

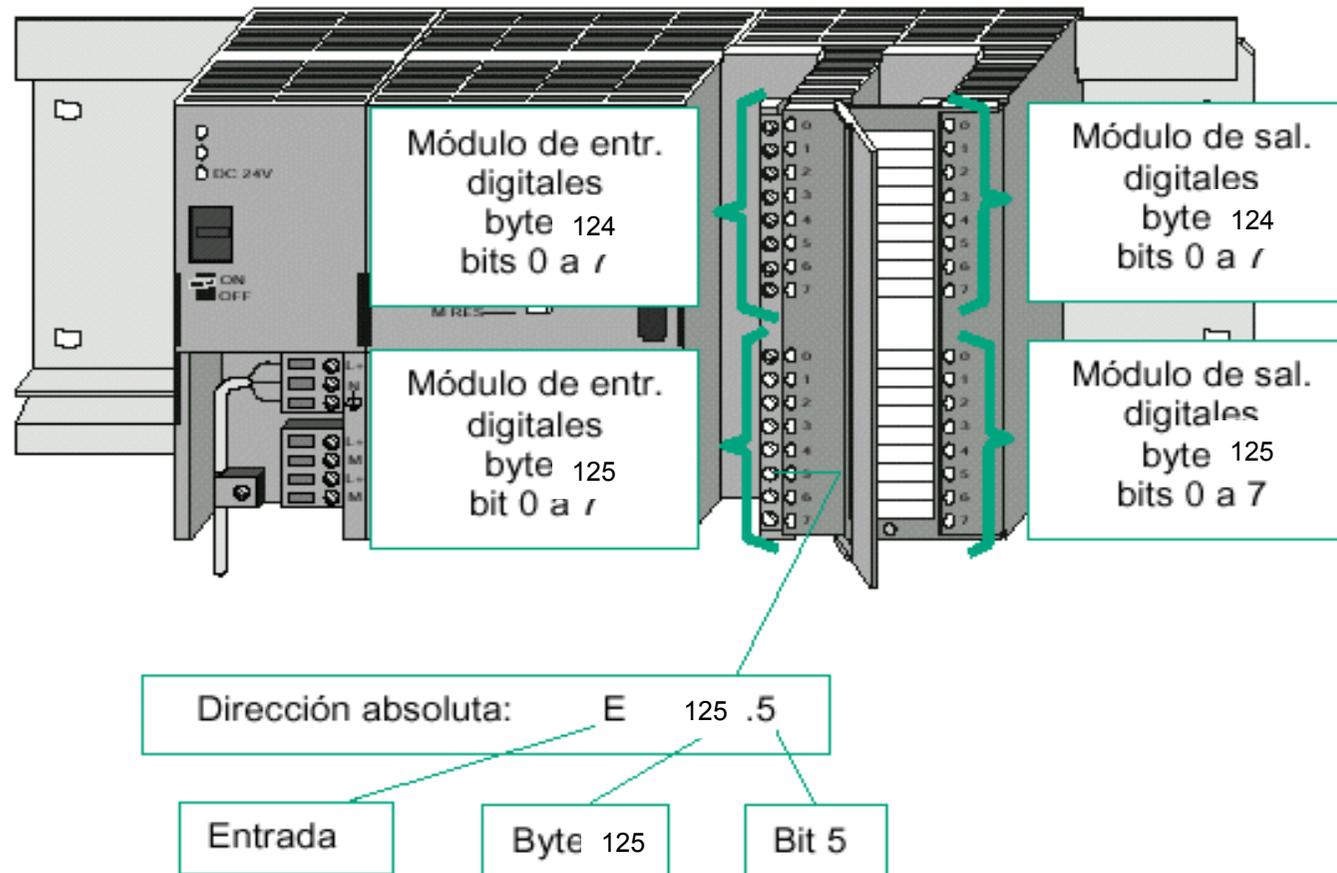
- E -> Entradas
- A -> Salidas
- T -> Temporizadores
- Z -> Contadores
- M -> Marcas
- DB -> Módulos de datos
- P -> Acceso directo (PA, PE)



Número de byte y bit (0-7), en su caso

Tamaño del Dato que vamos a usar (bit por omisión): B, W, D

# DIRECCIONAMIENTO 2/3



## DIRECCIONAMIENTO 3/3



- Dentro de cada byte, palabra o doble palabra, los bits se numeran de derecha a izquierda, de 0 al 7.      E 124.0

- Dir. byte: *área de memoria* + **B** + *nº de byte*      EB124

- Dir. palabra: *área de memoria* + **W** + *nº de su 1<sup>er</sup> byte*

EW124 => EB124 EB125

- Dir. doble palabra: *área de memoria* + **D** + *nº de su 1<sup>er</sup> byte*

MD0 => MB0 MB1 MB2 MB3

PLC's

# SISTEMAS NUMÉRICOS



- **Sistemas numéricos soportados:**
  - **Decimal** (enteros y fraccionarios), **binario**, **hexadecimal**, **BCD**
- **Tipos de datos:**
  - Entero (en cualquiera de los sistemas numéricos anteriores)
  - Real (coma flotante)
  - Carácter
  - Tiempo: cantidad de tiempo, hora, S5T, fecha
  - Puntero

Ver ayuda

**Admin. Simatic**

Tipos de datos simples

# Tipos de Datos Elementales en STEP 7



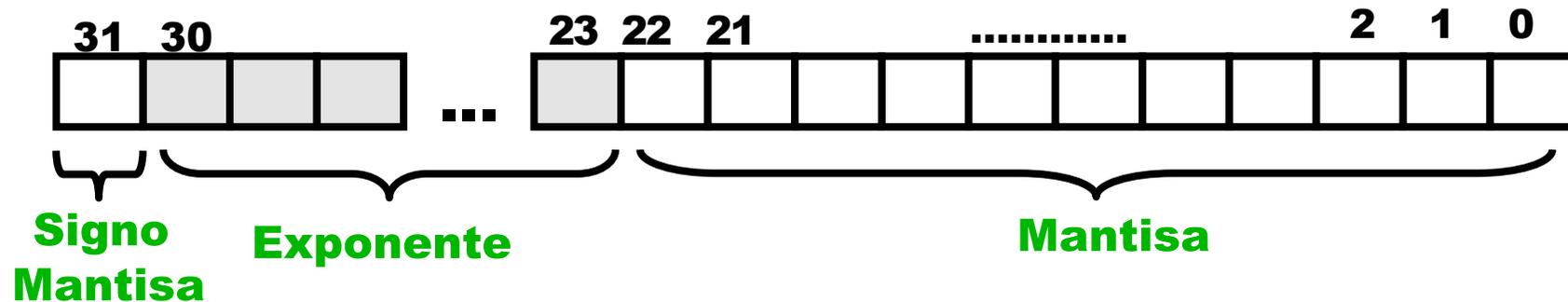
Palabra Clave	Long. (en bits)	Ejemplo de una constante de este tipo
BOOL	1	1 ó 0
BYTE	8	B#16#A9      o    B169
WORD	16	W#16#12AE    o    W4782
DWORD	32	DW#16#ADAC   o    DW44460
CHAR	8	' w '
S5TIME	16	S5T#5s_200ms
INT	16	123
DINT	32	65539
REAL	32	1.2      o      34.5E-12
TIME	32	T#2D_1H_3M_45S_12MS
DATE	16	D#1993-01-20
TIME_OF_DAY	32	TOD#12:23:45.12

# Números en COMA FLOTANTE (1/2)



**Estándar IEEE754:** Expresa números reales en 32 bits

$$\text{N}^\circ \text{ decimal} = (-1)^{\text{signo}} \cdot 2^{\text{exponente}-127} \cdot (1 + \text{Mantisa} \cdot 2^{-23})$$



- **Signo (S)** ⇒ 1: negativo , 0: positivo (bit 31)
- **Mantisa (M)** ⇒ La mantisa incluye 23 bits (bit 0.. 22).
- **Exponente (E)** ⇒ El exponente incluye 8 bits (bit 23..30).

# Números en COMA FLOTANTE (2/2)



**Estándar IEEE754:** Expresa números reales en 32 bits

$$\text{N}^\circ \text{ decimal} = (-1)^{\text{signo}} \cdot 2^{\text{exponente}-127} \cdot (1 + \text{Mantisa} \cdot 2^{-23})$$

- **Ejemplo:**

1#10000000#11000000000000000000000000000000

- Parte ligada al signo:  $(-1)^1 = -1$
- Parte ligada al exponente:  $2^{128-127}=2^1 = 2$
- Parte ligada a la mantisa:  $1 + 6291456 \cdot 2^{-23} = 1 + 0.75 = 1.75$
- Resultado:  $-1 \cdot 2 \cdot 1.75 = -3.5$

## ACUMULADORES (1/2)



- Registros auxiliares de 32 bits para intercambio de datos y operaciones de comparación y aritméticas.

ACCU-H (16 bits)	ACCU-L (16 bits)
------------------	------------------

- CPU 314IFM tiene 2 acumuladores.
- Carga al ACCU1,  
y el antiguo ACCU1 pasa al ACCU2.
- Si cargamos o transferimos tamaños de datos menores a la doble palabra ( bits, bytes (B), word (W) ),  
**se empieza a leer y escribir por la derecha** (byte menos significativo)

## ACUMULADORES (2/2)



### Operaciones entre acumuladores:

- TAK: intercambia contenido ACCU1 y ACCU2 .
- PUSH: copia el contenido de ACCU1 al ACCU2, sin cambiar el 1º.
- POP: copia el contenido de ACCU2 al ACCU1, sin cambiar el 2º.
- ENT (sólo S7-400): ACCU2 a ACCU3 y ACCU3 a ACCU4.
- LEAVE (sólo S7-400): ACCU3 a ACCU2 y ACCU4 a ACCU3.

## CARGA Y TRANSFERENCIA 1/7



Permiten programar intercambio de información  
entre módulos de E/S y áreas de memoria.

L --> Instrucción de **carga a ACCU-1**

T --> Instrucción de **transferencia desde ACCU-1**

- Cargan y transfieren del acumulador ACCU1, **independientemente del valor RLO, y tampoco lo modifican.**
- A dichas instrucciones **les sigue un código** en el que se indica el tamaño, formato y el número/posición a cargar o transferir.

## CARGA Y TRANSFERENCIA 2/7



- Ejemplos:

- **LC Z0** --> Carga el contenido del contador en el acumulador en BCD.

Si Z0=10                      ACU1: 00 00 00 ..... 0001 0000                      (BCD)

- **L Z0** --> Idem pero en Binario

Si Z0=10                      ACU1: 00 00 00 ..... 0000 1010 (Binario auténtico)

- **L S5T#5S** --> Herencia de versiones previas a S7: máximo número 9990sg,

## CARGA Y TRANSFERENCIA 3/7



- **Ejemplos:**

- **L ED124** (Cargar en el acumulador la doble palabra 124 de la PAE)

ACU1: EB124 EB125 EB126 EB127



Obsérvese: el byte más significativo del acumulador (parte izquierda) queda relleno con el byte de dirección menor, y el de mayor dirección, a la dcha.

- **L EW8** (Cargar en el ACU1 la palabra 8 de la PAE)

00 00 EB8 EB9



- **L EB8** (Cargar en el ACU1 el byte 8 de la PAE)

00 00 00 EB8



## CARGA Y TRANSFERENCIA 4/7



- Ejemplos (con los Registros de Direcciones, AR1 y AR2):
  - LAR1 ----> Carga el reg.de dirección AR1 con el contenido de ACU1
  - LAR1 R2 ----> Carga el reg. de dirección AR1 con el contenido de reg AR2
  - TAR ----> Intercambia los contenidos de registros AR1 y AR2
  - TAR1 R2 ----> Transfiere el contenido de AR1 a AR2

El listado AWL siguiente:

L ED 124

LAR1

L ED 128

LAR2

TAR

Tendrá como  
resultado:

ED124 ----> AR2

ED128 ----> AR1

# CARGA Y TRANSFERENCIA 5/7



- Ejemplos (L y T con diferentes tamaños):

– L EB124 (Cargar en el ACU1 el byte 124 de las entradas)

00 00 00 EB124

– Si ahora hacemos T MW32

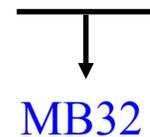
00 00 00 EB124



MB32 MB33

• L ED124 EB124 EB125 EB126 EB127

T MB32



PLC's

## CARGA Y TRANSFERENCIA 6/7



L +5

L L#523123

L B#16#EF

L 2#0011\_0110\_1110\_0011

T MB0

T AD256

### NOTA:

Ver ayuda  
Adm. Simatic

“Tipos de datos  
simples”

# CARGA Y TRANSFERENCIA 7/7



## Ejemplo

AWL	Explicación
L	EB10
L	MB120
L	DBB12
L	DIW15
L	LD252
L	P# E 8.7

## Ejemplo

AWL	Explicación
T	AB10
T	MW14
T	DBD2

## EJERCICIO (Entregable 4)



- Tenemos una mezcladora que produce 2 tipos de productos, magdalenas y donuts, para lo cual controla la receta, modificando las proporciones de harina (A124.0) y leche (A124.1), y el tiempo que está mezclándolos (A124.2).
- La producción empezará cuando E124.0 se ponga a 1
- Si E124.1=1 produce magdalenas, para lo cual A124.0 está encendido 3 segundos, A124.1 está encendido 1 segundo y A124.2 durante 2 sg.
- Si E124.1=0 produce donuts, para lo cual los tiempos son 4, 3 y 2 segundos respectivamente.
- El sistema debe ser robusto de forma que si está en medio de una receta no debe interrumpirse aunque cualquiera de los dos botones de entrada se modifique