

**Manual de formación  
para soluciones generales en automatización  
Totally Integrated Automation (T I A)**

***Anexo I***

**Fundamentos de la programación de PLC con  
SIMATIC S7-300**

Estos documentos están elaborados por Siemens A&D FEA (Investigación, desarrollo y formación en Automatización automática) con la finalidad que su uso sea el de la formación.

Siemens no se compromete a garantizar a lo que concierne al contenido.

La publicación de estos documentos, así como la utilización y el anuncio de éstos, está permitida dentro de la formación pública. Con la salvedad de que se precisa la autorización escrita por Siemens A&D FEA (Hr. Knust: E-mail: michael.knust@hvr.siemens.de).

Las infracciones serán sometidas a una indemnización. Todos los derechos de las traducciones están también condicionados, especialmente para el caso de la patentación ó del registro GM.

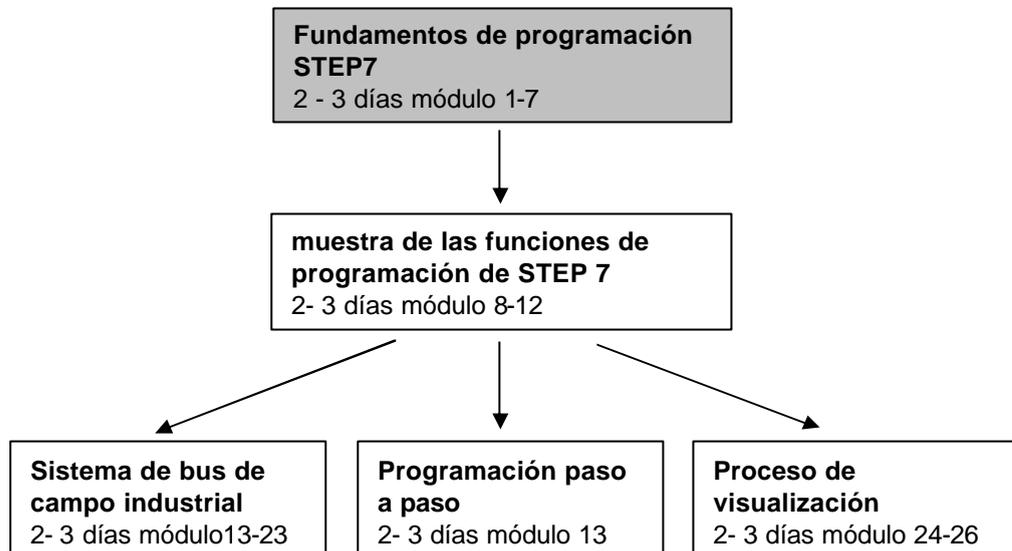
Agradecemos al Ingeniero Fa. Michael Dziallas y a los profesores, así como a las personas que han apoyado la elaboración de estos documentos.

		PÁGINA:
<b>1.</b>	<b>Preámbulo</b> .....	5
<b>2.</b>	<b>Estructura y función de un PLC</b> .....	6
<b>2.1</b>	<b>Clase de señales en automatización</b> .....	6
2.1.1	Señal binario.....	6
2.1.2	Señal analógico .....	8
<b>2.2</b>	<b>Sistema numérico</b> .....	10
2.2.1	Sistema decimal .....	10
2.2.2	Sistema binario.....	11
2.2.3	Código BCD (Código 8-4-2-1).....	11
2.2.4	Sistema hexadecimal .....	12
2.2.5	Descripción del sistema numérico .....	13
2.2.6	Reglas de conversión.....	14
<b>2.3</b>	<b>Conceptos sobre la informática</b> .....	16
2.3.1	Bit .....	16
2.3.2	Byte .....	16
2.3.3	Palabra .....	16
2.3.4	Doble palabra.....	17
2.3.5	Dirección de bit.....	17
2.3.6	Dirección de byte.....	17
2.3.7	Dirección de palabra.....	18
2.3.8	Dirección de doble palabra .....	18
<b>2.4</b>	<b>Estructura de un PLC</b> .....	19
<b>2.5</b>	<b>Partes fundamentales en un PLC</b> .....	21
2.5.1	Módulo central de proceso CPU .....	21
2.5.2	Sistema de bus.....	22
2.5.3	Fuente de alimentación.....	23
2.5.4	Memoria de programa.....	23
2.5.5	RAM .....	23
2.5.6	Flash- EPROM .....	23
<b>2.6</b>	<b>Sistema de automatización SIMATIC S7</b> .....	24
2.6.1	SIMATIC S7-300 .....	24

		PÁGINA:
<b>2.7</b>	<b>Ejecución del programa .....</b>	30
2.7.1	Memoria de programa.....	30
2.7.2	Ejecución del programa en línea .....	30
2.7.3	Estructura de la ejecución del programa.....	31
2.7.4	Bloque de usuario .....	32
2.7.5	Bloque del sistema para funciones standard y funciones del sistema.....	32
<b>3.</b>	<b>Lenguaje de programación STEP7.....</b>	<b>33</b>
<b>3.1</b>	<b>Conceptos generales del lenguaje de programación STEP7.....</b>	<b>33</b>
<b>3.2</b>	<b>Conversión STEP5 a STEP7.....</b>	<b>33</b>
<b>3.3</b>	<b>Norma IEC 61131.....</b>	<b>34</b>
<b>3.4</b>	<b>Estructura del directorio .....</b>	<b>35</b>
<b>3.5</b>	<b>Configurar y parametrizar .....</b>	<b>37</b>
<b>3.6</b>	<b>La instrucción de control .....</b>	<b>38</b>
3.6.1	Parte de la operación.....	38
3.6.2	Parte del operando .....	40
<b>3.7</b>	<b>Direccionamiento .....</b>	<b>41</b>
3.7.1	Direccionamiento simbólico.....	41
3.7.2	Direccionamiento absoluto.....	41
<b>3.8</b>	<b>Descripción del programa .....</b>	<b>43</b>
3.8.1	Esquema de contactos KOP.....	43
3.8.2	Diagrama de funciones FUP .....	43
3.8.3	Lista de instrucciones AWL .....	44
<b>3.9</b>	<b>Marcas.....</b>	<b>45</b>
3.9.1	Marcas remanentes .....	45
3.9.2	Marcas no remanentes .....	45

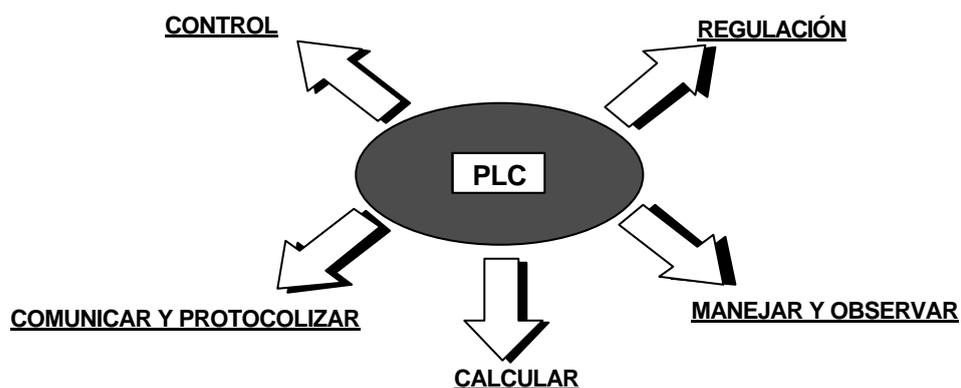
## 1. PREÁMBULO

El apéndice A es la condición para la realización del módulo ,Fundamentos en la programación STEP 7'.



### Objetivo:

Controlador lógico programable, PLC, es considerado hoy en día como pieza fundamental en cada automatización. Con estos controladores se encuentran soluciones económicas a diferentes problemas en las tareas de automatización.



Estos documentos deben dar una visión general sobre el Autómata S7-300 y también sobre el software de programación STEP 7.

### Condiciones:

En este apéndice de fundamentos se sobrepone, que tampoco son precisas condiciones especiales.

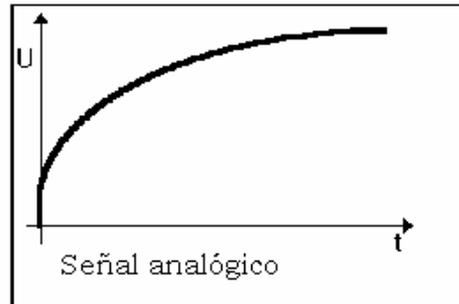
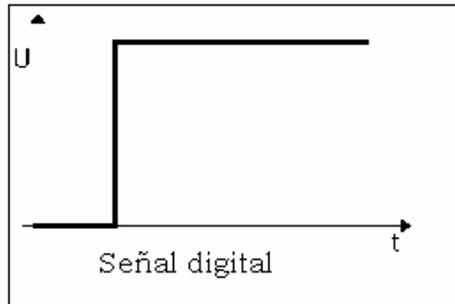
<b>Preámbulo</b>	Estructura y función de un PLC	Lenguaje de programación STEP 7
------------------	--------------------------------	---------------------------------

## 2. ESTRUCTURA Y FUNCIONES DE UN PLC

La regulación tiene la tarea de ejecutar operaciones específicas en una máquina o equipo, después de simular el desarrollo del funcionamiento en función del tipo de señal de entrada.

### 2.1 CLASE DE SEÑALES EN AUTOMATIZACION

Las entradas y salidas de las señales eléctricas se pueden dividir fundamentalmente en dos grupos :



#### 2.1.1 SEÑAL BINARIO

La señal binaria sólo puede admitir dos estados. Estos son:

Estado „1,,	=	presencia de tensión	=	p.e. interruptor ON
Estado „0,,	=	ninguna presencia de tensión	=	p.e. interruptor OFF

En regulación técnica se utiliza con frecuencia una tensión continua de 24 V como “tensión de regulación”. Un nivel de tensión de +24 V en un borne de entrada indica un estado „1„. Análogamente, una tensión de 0V indica un estado „0„. Se tiene que tener conocimiento, si un transmisor se refiere a un contacto abierto a un contacto cerrado. Un contacto abierto suministra en “caso activo”, es decir si está ocupado, una señal „0„. Se designa esta conexión como „activo 0„ o „activo low„. Análogamente un contacto cerrado es „activo 1„ / „activo high„, y entrega una señal „1„, si esta ocupado. Generalmente los transmisores de señal están en „activo 1„. Una aplicación típica para un transmisor „activo 0„ es un interruptor de emergencia. Un interruptor de emergencia está siempre desaccionado ( botón de emergencia sin apretar), es decir, entrega una señal „1„, cuando se cierra la entrada (seguridad por rotura). Se ha de provocar la actuación de un interruptor de emergencia en determinadas reacciones (p.e. cerrar todas las válvulas), donde la señal tiene que ser „0„.

### Posición binaria:

Una señal binaria sólo puede tener 2 valores, „0„ o „1„. Se caracteriza también por su posición binaria y recibe en el lenguaje técnico el nombre de „Bit„.

La agrupación de más señales binarias representan, en una determinada codificación (código), una señal digital. Mientras una señal binaria sólo comprende dos posibilidades (p.e. puerta cerrada / puerta abierta), a través de la combinación de las posiciones binarias se pueden formar número y cifras (información digital).

La combinación de n posiciones binarias permite la representación de  $2^n$  diferentes combinaciones.

Así pues por ejemplo, con 2 posiciones binarias  $2 \times 2$ , se obtienen 4 posibles soluciones:

0	0	Información1	(p.e. ambos interruptores abiertos)
0	1	Información2	(interruptos 1 cerrado / interruptor 2 abierto)
1	0	Información3	(interruptos 1 abierto / interruptor 2 cerrado)
1	1	Información4	(ambos interruptores cerrados)

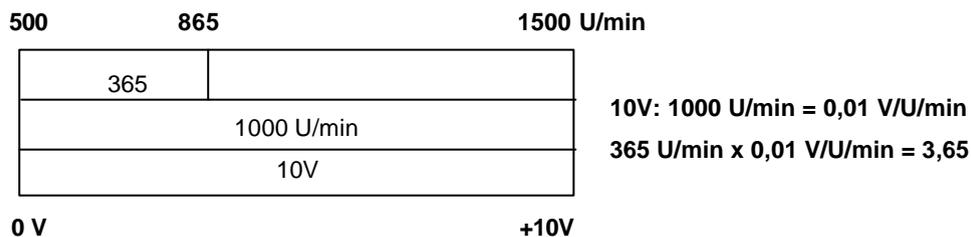
## 2.1.2 SEÑAL ANALÓGICO

Al contrario que para una señal binaria, que sólo puede tener dos estados, „presencia de tensión +24V,, y „ninguna presencia de tensión 0V,, las señales analógicas pueden comprender un determinado dominio de valores. Un típico ejemplo para un transmisor analógico es un potenciómetro. Con el posicionamiento del botón de giro se puede poder una resistencia hasta su valor máximo.

Ejemplos para medidas analógicos en la regulación técnica :

- Temperatura -50 ... +150°C
- Paso de circulación 0 ... 200l/min
- Número de revoluciones 500 ... 1500 U/min
- etc

Estas medidas se convierten, con la ayuda de un convertidor de medición, en tensiones eléctricas, corrientes o resistencias. P.e. para captar un número de revoluciones, se puede transforma un rango de revoluciones de 500 ... 1500 U/min, con un convertidor, a un rango de tensión de 0...+10V. Para la medida de las revoluciones de 865 U/min, un convertidor de valor de tensión debería dar una tensión de +3,65 V.



Para trabajar con medidas analógicas en un PLC, se han de convertir los valores de tensión, corriente o resistencia en una información digital. Esta conversión se obtiene con un conversor analógico – digital (Conversor A/D). Esto significa, que p.e. el valor de tensión de 3,65 V en una línea en posición binaria se guarda como información. Cuantas más posiciones binarias se utilicen para la representación digital, más precisa será la resolución. Si se dispusiera p.e. para el rango de tensión 0...+10V de un sólo 1 Bit, sólo se podría realizar una declaración, si el rango de tensión está comprendido entre 0...+5V o entre +5V...+10V. Con 2 bits se puede dividir el rango en 4 partes, es decir 0 ... 2,5 / 2,5 ... 5 / 5 ... 7,5 / 7,5 ... 10V. Los convertidores de corriente A/D en la regulación técnica convierten con 8 o 11 bit.

A continuación se obtiene con 8 bit 256 partes y con 11 bit una resolución de 2048.



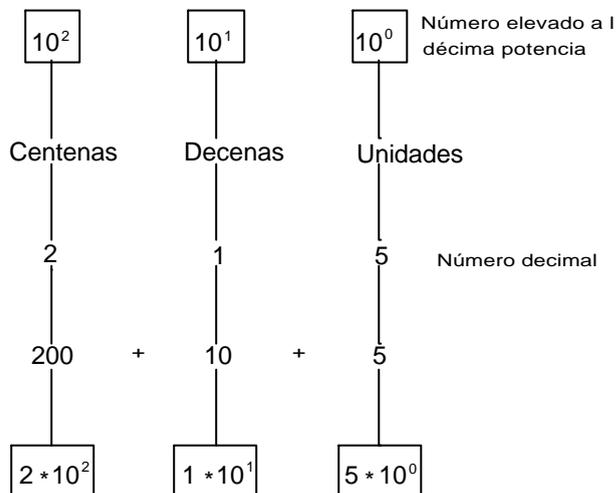
## 2.2 SISTEMA NUMÉRICO

Para la adaptación de la dirección en una célula de memoria, entradas ,salidas, tiempos, marcas, etc.. a través de un controlador lógico programable (autómata) no se utiliza el sistema decimal, sino el sistema binario.

### 2.2.1 SISTEMA DECIMAL

Para comprender el sistema numérico binario es necesario antes de nada, considerar el sistema decimal.

Aquí se ha de clasificar el número 215. El 2 corresponde a la centenas, el 1 a las decenas y el 5 a las unidades. En realidad se ha de escribir 215 como:  $200+10+5$ . Se ha de escribir la expresión  $200+10+5$ , como abajo está representada, con ayuda del número elevado a 10.

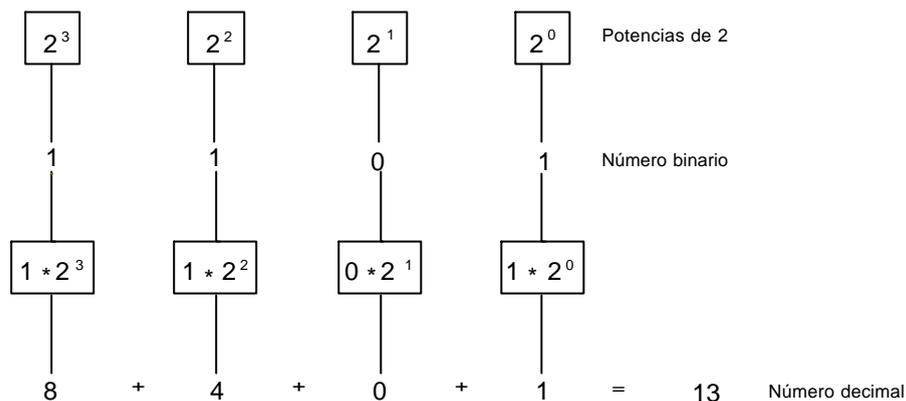


Cada cifra dentro del número decimal está clasificada en un número elevado a la décima potencia.

## 2.2.2 SISTEMA NUMÉRICO BINARIO

El sistema numérico binario reconoce sólo las cifras 0 y 1, que en la elaboración de datos técnicos es fácil representable y utilizable. Por lo tanto, esto es un sistema numérico binario.

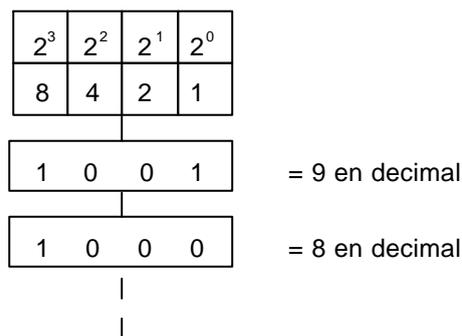
La posiciones de los valores en el sistema binario se clasifican, como se describe a continuación, en potencias de dos.



Cada cifra dentro del número binario está clasificada en un número elevado a la décima de 2.

## 2.2.3 CÓDIGO BCD (CÓDIGO 8-4-2-1)

Para representar importantes valores numéricos ordenadamente, se utiliza con frecuencia el código BCD (codificación-binaria-decimal). Los números decimales se representan con la ayuda del sistema numérico binario. La cifra decimal con el valor más alto es el 9. Para la representación del número 9 se necesita la potencia de dos hasta  $2^3$ , por lo tanto, en total se necesitan 4 posiciones.



Para la descripción de la cifra decimal más grande de 4 posiciones binarias, se designa para cada cifra decimal una unidad de 4 posiciones, llamada tetrada. El código BCD es por lo tanto un código de 4 bits.

Cada número decimal se codifica por separado. El número 255 está formado p.e. por 3 cifras decimales. Cada cifra decimal esta representada en código BCD como una unidad con 4 posiciones (tetra).

2	8	5
0010	1000	0101

Cada cifra decimal se representa con un código propio a través de una tetra.

## 2.2.4 SISTEMA HEXADECIMAL

El sistema hexadecimal pertenece a los sistemas numéricos. Como valor se utilizan las potencias de 16 números. El sistema hexadecimal es por lo tanto un sistema numérico dieciseisavo.

Cada posición dentro de un número hexadecimal se clasifica en una potencia dieciseisava. Se necesitan con el 0 16 cifras. Para las cifras de 0 a 9 se utiliza el sistema decimal y para las cifras de 10 a 15 las letras A, B, C, D, E y F.

Cada cifra dentro del número hexadecimal está clasificada en una potencia de número 16.

## 2.2.5 DESCRIPCIÓN DEL SISTEMA NUMÉRICO

Decimal	Binario					Hexadecimal
	16	8	4	2	1	
0					0	0
1					1	1
2				1	0	2
3				1	1	3
4			1	0	0	4
5			1	0	1	5
6			1	1	0	6
7			1	1	1	7
8		1	0	0	0	8
9		1	0	0	1	9
10		1	0	1	0	A
11		1	0	1	1	B
12		1	1	0	0	C
13		1	1	0	1	D
14		1	1	1	0	E
15		1	1	1	1	F
16	1	0	0	0	0	1 0
17	1	0	0	0	1	1 1
18	1	0	0	1	0	1 2
19	1	0	0	1	1	1 3



## 2.2.6 REGLAS DE CONVERSION

La conversión entre diferentes sistemas numéricos se base en reglas sencillas.

El usuario del PLC debería controlarlas, ya que estas conversiones se usan con frecuencia.

La identificación del sistema numérico, sobre el cual se fundamenta el número entregado, es un subíndice que se coloca a la derecha del número.

Con eso se coloca una D, para decimales, una „B, para binarios y una „H, para hexadecimales. Esta identificación es a menudo necesaria, ya que conforme se van considerando sistemas numéricos, es precisa una completa distinción de los valores.

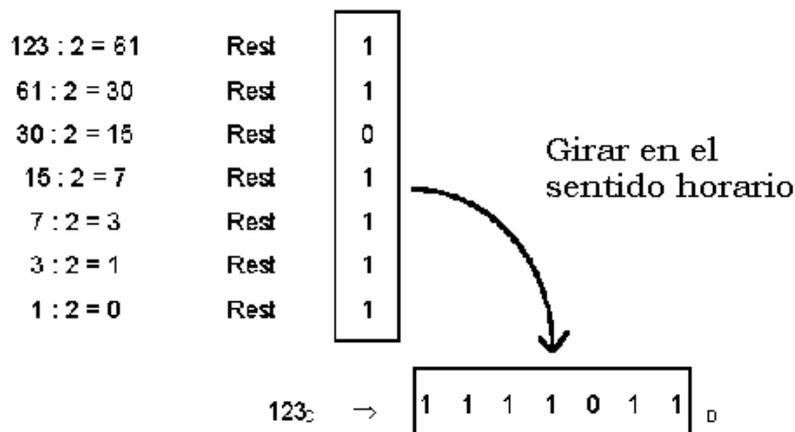
Así es p.e. „111,, el valor en sistema decimal es  $111_D$  (ciento once), en binario  $111_B$ , en valor decimal sería  $7 (1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2)$  y el número hexadecimal  $111_H$ , en valor decimal sería  $273 (1 \times 16^0 + 1 \times 16^1 + 1 \times 16^2)$ .

### Conversión decimal → binario

El número completo decimal se divide en base 2, hasta que el resultado sea nulo.

De ahí que el resto de la división (0 o 1) resulta un número binario. Hay que observar el contenido del „Resto“. El resto de la primera división es el primer bit de la derecha. (el valor más bajo del bit).

P.e.: El número decimal 123 se debe transformar en su correspondiente número binario:



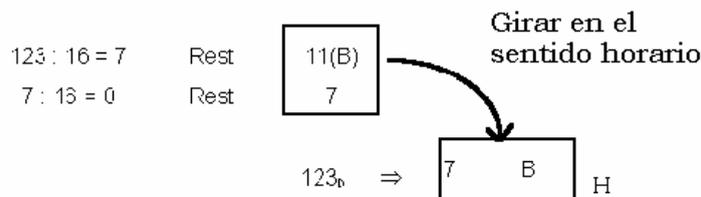
Prueba:

$$\begin{array}{cccccccc}
 1 & 1 & 1 & 1 & 0 & 1 & 1 & \\
 1 \times 2^6 + & 1 \times 2^5 + & 1 \times 2^4 + & 1 \times 2^3 + & 0 \times 2^2 + & 1 \times 2^1 + & 1 \times 2^0 & \\
 64 + & 32 + & 16 + & 8 + & 0 + & 2 + & 1 & = & \underline{123}
 \end{array}$$

## Conversión decimal → hexadecimal

La conversión se realiza igual que la conversión decimal → binario. Con la diferencia, que en vez de utilizar base 2 se utiliza base 16. Por eso se ha de dividir por 16, en vez de por 2.

P.e.: El número decimal 123 se debe transformar en su correspondiente número hexadecimal:



Prueba:

$$\begin{array}{r}
 7 \qquad \qquad B \\
 7 \times 16^1 + 11 \times 16^0 \\
 112 \quad + \quad 11 \qquad = \quad \underline{123}
 \end{array}$$

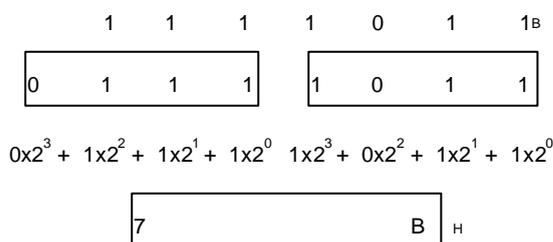
## Conversión binario → hexadecimal

Para la conversión de un número binario en uno hexadecimal se podría en primer lugar examinar el valor decimal del número binario. Este número decimal se podría entonces transformar con ayuda de la división por 16 en un número hexadecimal.

Pero también existe otra posibilidad, se puede deducir directamente desde el número binario el valor en hexadecimal.

Para esto se ha de dividir primero el número binario empezando por la derecha en cuatro grupos. Cada uno de estos grupos da como resultado una cifra en el sistema hexadecimal. En el casos necesarios se ha de rellenar en el grupo de la izquierda las carencias de bits con ceros.

P.e.: El número binario 1111011 se ha de transformar directamente en un número Hex.:





### 2.3.4 DOBLE PALABRA

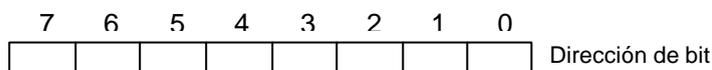
A una doble palabra le corresponde una longitud de palabra de 32 signos binarios.

Una doble palabra tiene por lo tanto un tamaño de 2 palabras, 4 bytes o 32 bits.

Otras unidades son Kilo-bit o Kilo-byte para  $2^{10}$ , es decir 1024 Bit o en su caso 1024 Byte y el Mega-bit o el Mega-byte para 1024 Kilo-bit o en su caso 1024 Kilo-byte.

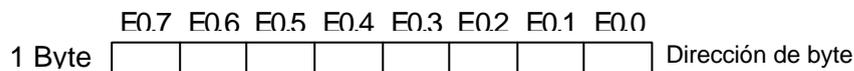
### 2.3.5 DIRECCION DE BIT

En cada byte se obtiene la dirección de bit 0 a la derecha y a la izquierda se obtiene la dirección de bit 7.



### 2.3.6 DIRECCION DE BYTE

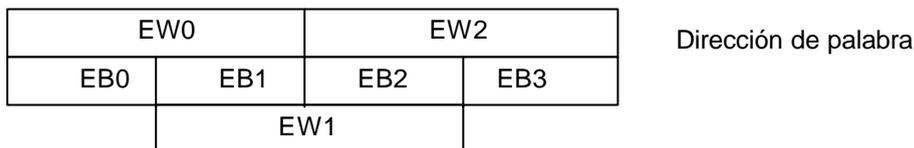
También los bytes individuales contienen números, la dirección de byte. Además el operando está caracterizado, de modo que p.e. se da EB2 para la entrada byte 2 o AB4 para la salida byte 5. Los bits individuales son direccionados inequívocamente a través de la combinación de la dirección de bits y bytes. Aparte de eso se separa la dirección de bit de la dirección de byte con un punto. A la derecha del punto está la dirección de bit y a la izquierda la dirección de byte.



## 2.3.7 DIRECCION DE PALABRA

La numeración de las palabras corresponde a la dirección de palabra.

**Nota:** Para la utilización de palabras, p.e. entradas palabra (EW), salidas palabra (AW), marca palabra (MW) ,etc la dirección de palabra es siempre la dirección más pequeña de los bytes pertinentes. P.e. en una palabra, existen EB2 y EB3 y la dirección es EW2.



**Nota:** Para ejecutar una palabra se ha de tener especial cuidado, en que p.e. la entrada palabra 0 y la entrada palabra 1 en un byte coincidan. Además se empieza a contar el número de bits por la derecha.

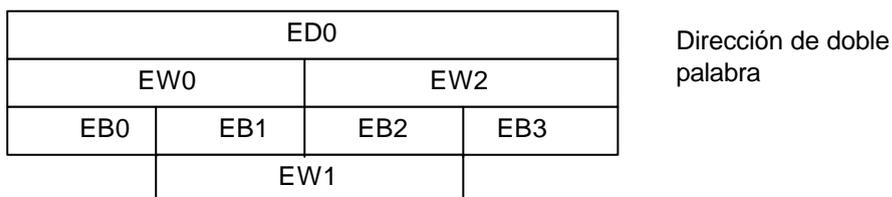
Por ejemplo el bit0 de EW1 es E2.0. El bit1 de E2.1..... el bit7 de E2.7. El bit8 de E1.0.... el bit15 de E1.7.

Entre los bits 7 y 8 existe por ello un salto.

## 2.3.8 DIRECCIÓN DE DOBLE PALABRA

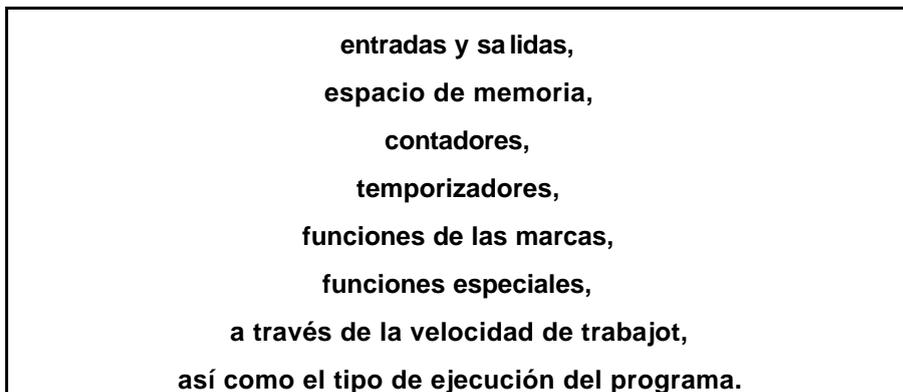
La numeración de dobles palabras resultada la dirección de doble palabra.

**Nota:** Para la utilización de dobles palabras, p.e. ED, AD, MD etc. la dirección de doble palabra es siempre la dirección de palabra más pequeña de las dos palabras pertenecientes.



## 2.4 ESTRUCTURA DE UN PLC

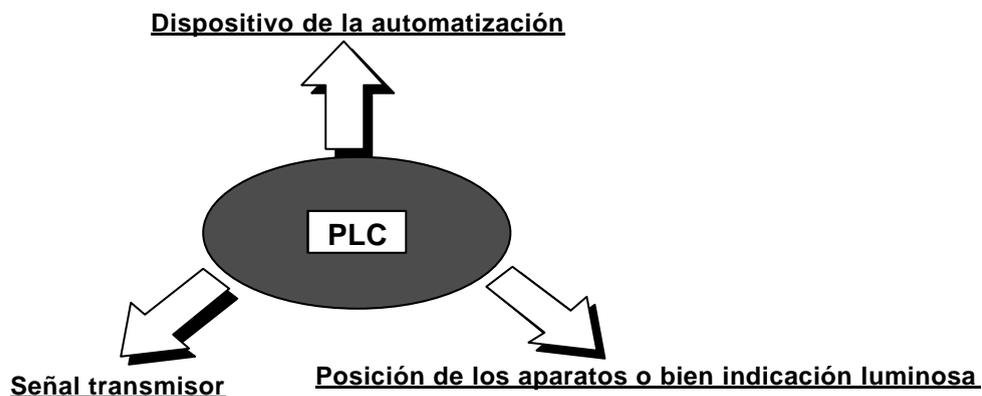
Los controladores lógicos programables son aparatos acabados en serie. Todos ellos necesitan para la regulación técnica uniones entre grupos, funciones de memoria, temporizadores, contadores, etc. Son planificados por el fabricante y se conectan por la programación par las funciones de control. Los aparatos de control se ofrecen en diferentes unidades de funciones. Se diferencian principalmente a través del número de posibilidades.



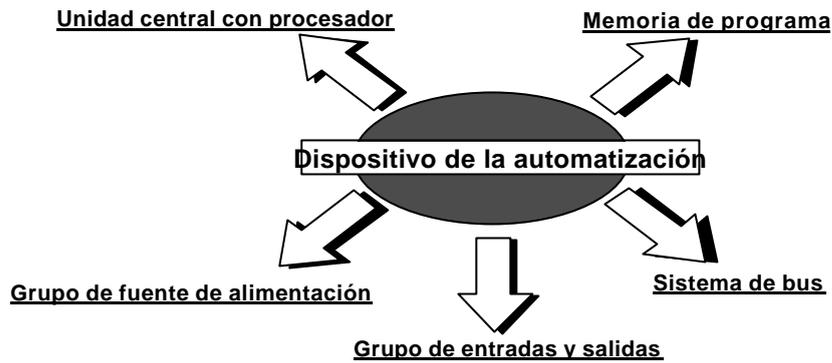
Los aparatos de control más grandes se montan en grupos separado por módulos individuales.

Para las tareas de control más pequeñas, se ofrecen equipos de control ya construidos y compactos. Se representan en una unidad cerrada con un número de entradas y salidas establecidas.

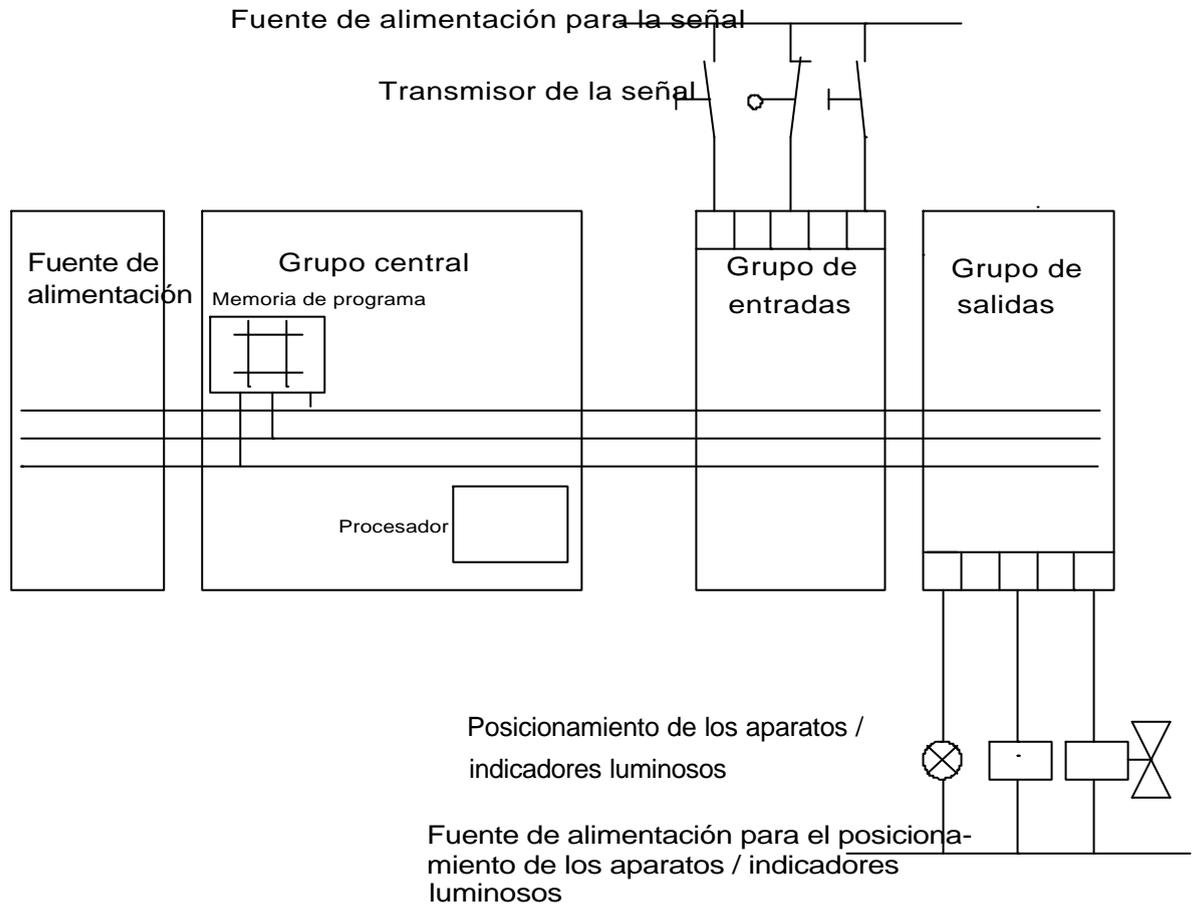
A un controlar lógico programable le pertenecen fundamentalmente:



El dispositivo de la automatización contiene principalmente:



Configuración de un controlador lógico programable:

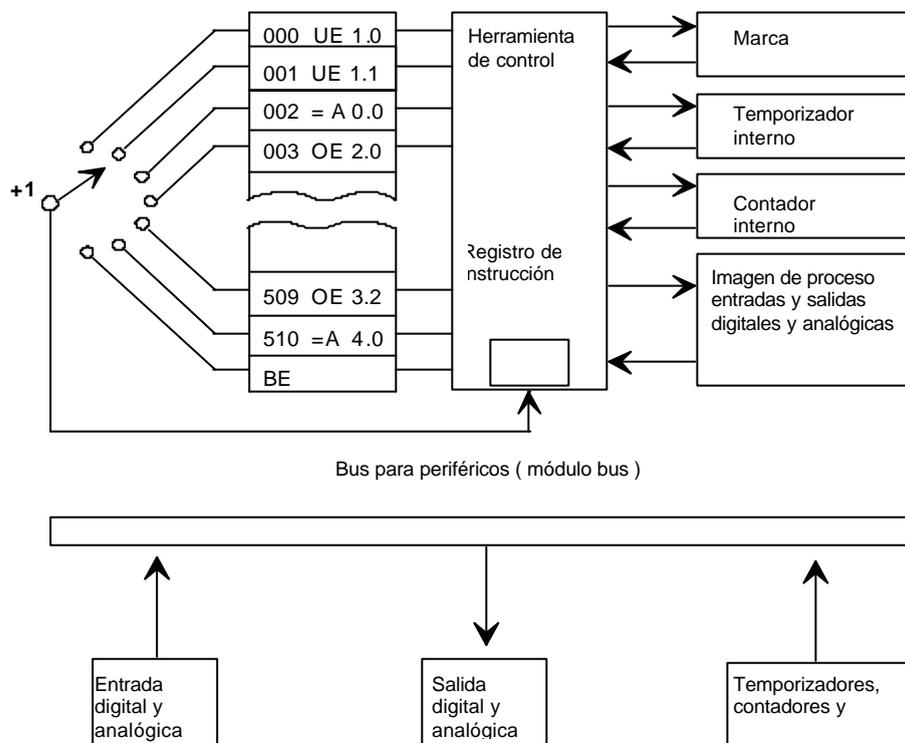


## 2.5 PARTES FUNDAMENTALES DE UN PLC

### 2.5.1 MÓDULO CENTRAL DE PROCESO CPU

A través de los transmisores de señal de tensión se conecta, sobre la regleta de terminales, el grupo de entrada. En la CPU (grupo de trabajo), el procesador ejecuta el programa que está en la memoria y éste pregunta, si las entradas de los aparatos tienen tensión o no. Dependiendo del estado de las entradas y de la posición del programa en la memoria, el procesador del grupo de trabajo, ejecuta las instrucciones en los módulos internos. Dependiendo del estado de tensión en las entradas de los grupos de trabajo se conectan o desconectan el posicionamiento de los aparatos así como los indicadores luminosos.

Unidad central (CPU) del PLC:



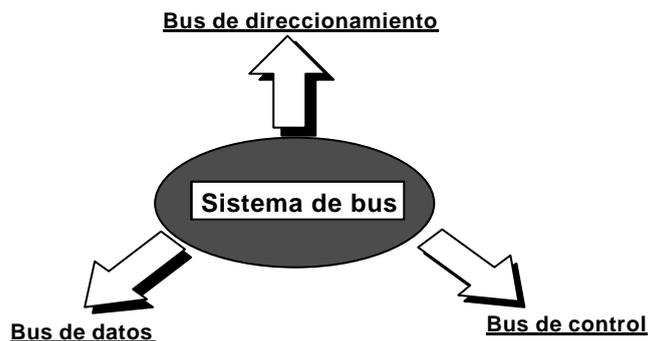
La dirección del contador solicita a la memoria de programa instrucción por instrucción una a continuación de la otra ( serie) y provoca la dependencia del programa de la transmisión de información de la memoria de programa al registro de instrucción. Todas las memorias en un procesador se llaman generalmente Registros. La unidad de control recibe sus instrucciones del registro de instrucción. Mientras la unidad de control ejecuta la instrucción actual, desplaza la dirección del contador a la próxima instrucción en el registro de instrucción. En el estado de transmisión de entradas en la imagen de proceso de las entradas (PAE) se ejecuta la unión, la entrada del temporizador, contador, acumuladores y la transferencia de los resultados lógicos (VKE) en la imagen de proceso de las salidas (PAA). Se reconoce después de la ejecución del programa de usuario (BE), que se ha efectuado la transferencia del correspondiente estado a la PAA en las salidas.

El bus de periferia lleva a cabo el intercambio de datos entre la unidad de proceso y la periferia. A la periferia pertenecen los módulos de entradas y salidas digitales, los módulos de entradas y salidas analógicas y los módulos de los temporizadores, contadores y módulos de valor limitado.

## 2.5.2 SISTEMA DE BUS

El sistema de bus es una canalización para transferir los señales. Este permite el intercambio de señales en dispositivos de la automatización entre el procesador y los módulos de entradas / salidas a través de un proceso en el sistema de bus.

El bus contiene tres direcciones paralelas de señales :



- Con el bus de direccionamiento se controlan la direcciones en cada módulo.
- Con el bus de datos se transfieren datos p.e. del módulo de entrada al de salida.
- Sobre el bus de control se transmiten las señales para el control y vigilancia de las funciones dentro de los dispositivos de la automatización.

## 2.5.3 FUENTE DE ALIMENTACION

La fuente de alimentación genera la tensión para la módulos eléctricos de los dispositivos de automatización. El nivel de esta tensión es de 24 voltios. Para las tensiones de los transmisores de señal, posicionamiento de los aparatos y indicadores luminosos, los cuales necesitan tensiones por encima de los 24 voltios, se suministran transformadores como complemento.

## 2.5.4 MEMORIA DE PROGRAMA

Los elementos de memoria son elementos, en los cuales se pueden guardar informaciones en forma de señales binarios.

Como memorias de programa se emplean en su mayoría memorias semiconductoras. La memoria contiene celdas de programa de 512, 1024, 2048 etc.. Es usual, declarar la capacidad de la memoria de programa ( es decir, el número de celdas de programa) en múltiplos de 1 K ( 1 K contiene 1024). En cada celda de programa se puede escribir (programar) con la ayuda de un aparato de programación una instrucción de control. Cada elemento binario en una celda de programa puede admitir el estado de señal "1" o "0".

## 2.5.5 RAM

En las celdas de programa se pueden escribir a menudo diferentes informaciones. La información se selecciona, sin que el contenido de la información se pierda.

Las memorias RAM son sin embargo memorias volátiles, es decir, que el contenido de la información se pierde, a caso de caída de tensión. Las memorias RAM se borran eléctricamente.

La memoria de trabajo interna de un S7-300 está en RAM. El respaldo por pila en los PLC ofrece seguridad a la memoria.

## 2.5.6 FLASH- EPROM

La EPROM es BORRABLE, PROGRAMABLE de sólo lectura. El contenido de la EPROM se borra en su totalidad a través de un rayo violeta o una tensión y a continuación se programa de nuevo. Por esto es adecuada para ser transportada sin pérdida de datos. En el S7-300 existe la posibilidad con el PG de guardar el programa en un tarjeta de memoria (Flash-EPROM) y con eso, en caso de una caída de tensión, se puede cargar rápidamente de nuevo el sistema en el modo de servicio.

Para borrar y sobrescribir la Flash- EPROM se utiliza una tensión de 5V.

## 2.6 SISTEMA DE AUTOMATIZACIÓN SIMATIC S7

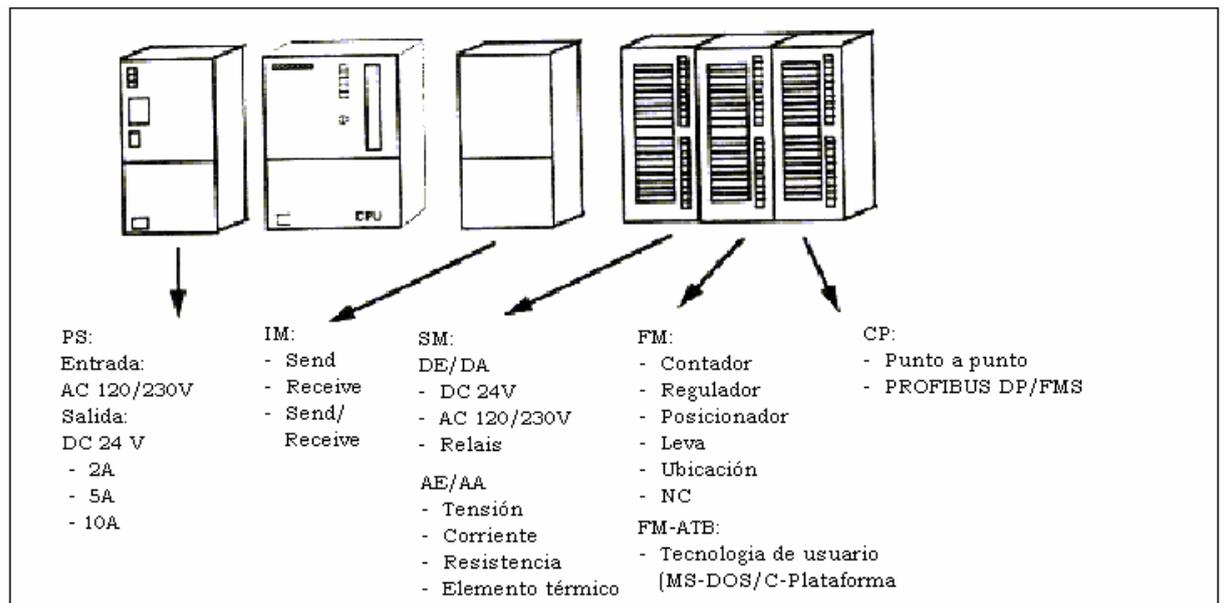
SIMATIC S7 es un producto de la empresa SIEMENS y responde a la serie actual de los controladores lógicos programables.

Esta familia de sistemas SIMATIC S7 es un bloque general de los conceptos de automatización para la fabricación y procesos técnicos. Totally Integrated Automation.

### 2.6.1 SIMATIC S7 - 300

#### Gamma del módulo:

Las siguientes clases de módulos forman, en conjunto con una CPU, la estructura de un SIMATIC S7-300:



### Elegir una CPU:

A continuación se relacionan las CPU's más relevantes que por el momento existen para la formación. No obstante existen otras CPU's con otras capacidades de rendimiento y también se están desarrollando otras nuevas CPU's. No obstante éstas sólo se caracterizan por una mejora en los cálculos aritméticos. Estas no son necesarias en los pequeños programas para la formación.

CPU 312 IFM	CPU 313	CPU 314 IFM	CPU 314	CPU 315	CPU315-2DP
2K Instrucciones 6Kbyte Memoria de trabajo 20Kbyte Ladesp.	4K Instrucciones 12Kbyte Memoria de trabajo 20Kbyte Ladesp.	8K Instrucciones 24Kbyte Memoria de trabajo 40Kbyte Ladesp.	8K Instrucciones 24Kbyte Memoria de trabajo 40Kbyte Ladesp.	16K Instrucción. 48Kbyte Memoria de trabajo 80Kbyte Ladesp.	16K Instrucción. 48Kbyte Memoria de trabajo 80Kbyte Ladesp.
128 Byte DE/DA	128 Byte DE/DA	512 Byte DE/DA	512 Byte DE/DA	1024 Byte DE/DA	1024 Byte DE/DA
32 Byte AE/AA	32 Byte AE/AA	64 Byte AE/AA	64 Byte AE/AA	128 Byte AE/AA	128 Byte AE/AA
0,6 ms / 1K Instrucción	0,6 ms / 1K Instrucción	0,3 ms / 1 K Instrucción	0,3 ms / 1K Instrucción	0,3 ms / 1K Instrucción	0,3 ms / 1K Instrucción
1024 Marcabit	2048 Marcabit	2048 Marcabit	2048 Marcabit	2048 Marcabit	2048 Marcabit
32 Contador	64 Contador	64 Contador	64 Contador	64 Contador	64 Contador
64 Temporizador	128 Temporizador	128 Temporizador	128 Temporizador	128 Temporizador	128 Temporizador
10 DE/6DA a bordo, de ellos 4 DE para el proceso de alarma o integración de funciones 20KByte EPROM		20 DE/16DA a bordo, de ellos 4DE para el proceso de alarma o funciones integradas 4 AI / 1AA a bordo Resolución: 11Bit + Signo integración 40KByte EPROM			PROFIBUS DP Conexión (Master / Slave) Asignación de direcciones parametrizable

## CPUs compactas:

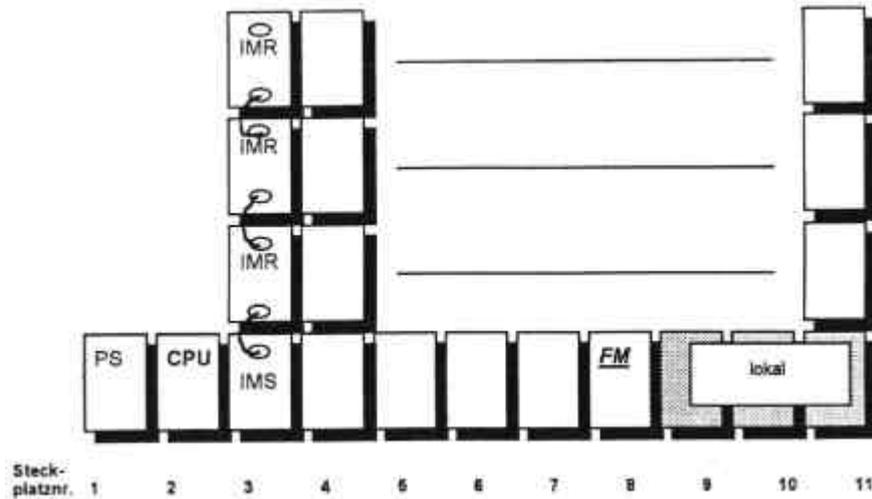
Entretanto existen también un número de CPUs compactas, que a través de módulos compactos con interface de comunicaciones integradas, destacan por tener unos precios especialmente asequibles y que son apropiadas para la formación.

Estas son comparables en funcionalidad y manipulación con las CPUs estándares.

La siguiente tabla muestra una selección de CPUs compactas.

	CPU 312C	CPU 313C	CPU 313C-2DP	CPU 314C-2DP
<b>Tiempo de instrucción</b> (Binario/Punto fijo/Punto flotante)	0.2 / 4 / 40µs	0.1 / 2 / 20µs	0.1 / 2 / 20µs	0.1 / 2 / 20µs.
<b>Memoria de trabajo</b>	16 KB	32 KB	32 KB	48 KB
<b>SIMATIC Micro Tarjeta de memoria</b>	64 KB bis 4 MB	64 KB bis 4 MB	64 KB bis 4 MB	64 KB bis 4 MB
<b>Periferia a bordo</b> DE/DA AE/AA *PT100	10 / 6 -/-	24 / 16 4+1*/2	16 / 16 -/-	24 / 16 4+1* / 2
<b>Funciones tecnológicas</b> - Contador/frecuencia - Impulso de salida - Regulación - Posicionamiento	2 (10KHz) 2 (2,5 KHz) no no	3 (30KHz) 4 (2,5 KHz) si no	3 (30KHz) 4 (2,5 KHz) si no	4 (60KHz) 4 (2,5 KHz) si 1
<b>Interface</b> - MPI 187,5 kBaud - DP-Funcionalidad	si no	si no	si Master/Slave	si Master/Slave

## Posibilidades de expansión de una S7 - 300



Esta figura muestra la extensión máxima de una S7-300/CPU314. La ampliación completa permite hasta 32 módulos(CPU 312IFM y CPU 313 con sólo 8 Módulos), cada estante hasta 8 módulos.

### Para la ocupación de un lugar en el bastidor se dan las siguientes reglas:

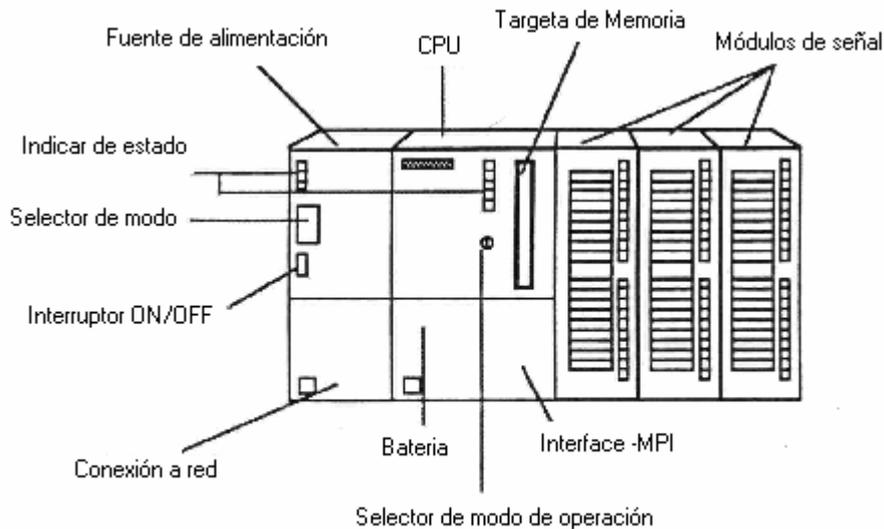
- Posición 1: Fuente de alimentación PS (=asignación fija)
- Posición 2: Módulo central de proceso CPU (=asignación fija)
- Posición 3: Módulo de acoplamiento IM (=asignación fija)
- Posición 4-11: Módulos de señales SM, Módulo de funciones FM y Procesadores de comunicaciones CP (=ninguna asignación fija)

La conexión del IMS representa al emisor y el IMR al receptor. Al módulo de conexión se le ha de asignar una posición en el bastidor.

En cada extensión del bastidor se ha de preveer una fuente de alimentación.

Los módulos de funciones, p.e. SINUMERIK FM NC, se puede clasificar en una particular periferia. Con esto, un módulo FM tiene su propio campo de periferia y por lo tanto se puede utilizar con rapidez. Este campo de periferia se clasifica como un segmento local. Pro bastidor se puede basar en un segmento local.

## Elementos importantes en la fuente de alimentación y en la CPU:

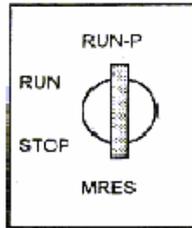


## Indicar de estados y errores en la CPU:

Indicación	Significado	Aclaración
SF (rojo)	Error general	Muestra un error general en el diagnóstico del módulo.
BAF (rojo)	Error de suministro	Indica que el respaldo de batería no suministra suficiente tensión o la ausencia de ésta.
DC5V (verde)	Suministro de DC5V para la CPU y retransformación del bus	Indica la capacidad de la función interna de suministro de 5V a la CPU
FRCE (amarillo)	Forzar	Indica para el estado de la CPU, si las entradas y salidas están forzadas a través de una función de test.
RUN (verde)	Estado de servicio RUN	reluce cuando la CPU está funcionando, cuando la CPU se encuentra en estado RUN
STOP (amarillo)	Estado de servicio STOP	Reluce cuando se realiza un reset. Cuando la CPU se encuentra en estado STOP

## Concepto de protección de la CPU:

Cada CPU posee un conmutador para seleccionar el tipo de funcionamiento. Los tipos de funcionamiento posibles son los siguientes:



- RUN-P: Se ejecuta el programa, todas las funciones PG están permitidas.
- RUN: Se ejecuta el programa, sólo se permite la lectura de las funciones PG.
- STOP: El programa no se ejecuta. Todas las funciones PG están permitidas.
- MRES: En esta posición se puede ejecutar un Reset, como se describe a continuación.

El concepto de protección para el S7-300 posibilita la protección de determinadas partes de los sistemas de automatización. Estas son:

- la CPU y todos los módulos programables
- todos los objetos (como p.e. módulos)

Esta protección de piezas puede ser influenciada a través del PG o a través del B&B.

El concepto de protección de la CPU está subdividido en tres niveles de protección :

- Nivel 1 – Posición del conmutador en Run-P o Stop: ninguna protección, todas las funciones están permitidas.
- Nivel 2 - Posición del conmutador en Run: Protección de escritura, la lectura de las funciones está permitida, p.e. observación de la funciones, funciones de información, cargar de la CPU.
- Nivel 3 – Configuración del S7 parametrizable(protección con password). Sólo está permitida la observación y información de las funciones, si el usuario del password no lo conoce.

Con el conmutador también se puede ejecutar un reset a la CPU de la siguiente manera:

Paso	Ejecución	Resultado
1	Girar el conmutador a la posición <b>STOP</b> .	El indicador STOP brilla
2	Mantener el conmutador en la posición MRES hasta que el indicador STOP brille, volver a la posición STOP.	El indicador STOP brilla una o dos veces y después de aprox. 3 segundos brilla de nuevo
3	Volver a girar la llave a la posición MRES (sólo un momento)	El indicador STOP brilla aprox. 3 segundos y después vuelve a alumbrar de nuevo: todo está en orden; <b>la CPU ha sido reseteada</b>

## 2.7 EJECUCIÓN DEL PROGRAMA

### 2.7.1 MEMORIA DE PROGRAMA

Para la ejecución del programa existen dos posibilidades diferentes, la que se utiliza en el control de un aparato y la programación independiente.

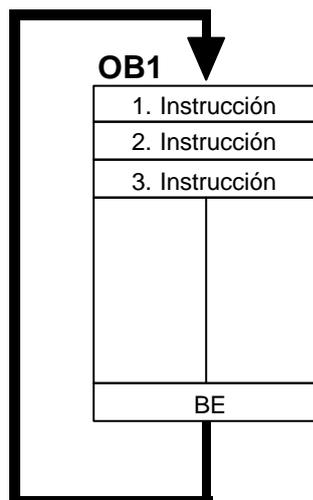
La ejecución de una única instrucción necesita un determinado tiempo (microsegundos). El tiempo para una única ejecución de todas las instrucciones se llama tiempo de ciclo y un recorrido por todo el programa, ciclo de programa.

### 2.7.2 EJECUCIÓN DEL PROGRAMA EN LÍNEA

Para la ejecución del programa en línea, se ejecutan en línea las instrucciones de control de un aparato, las cuales están guardadas en la memoria de programa. Cuando la ejecución del programa alcanza el final (BE), éste se ejecuta de nuevo desde el principio.

Se llama ejecución del ciclo.

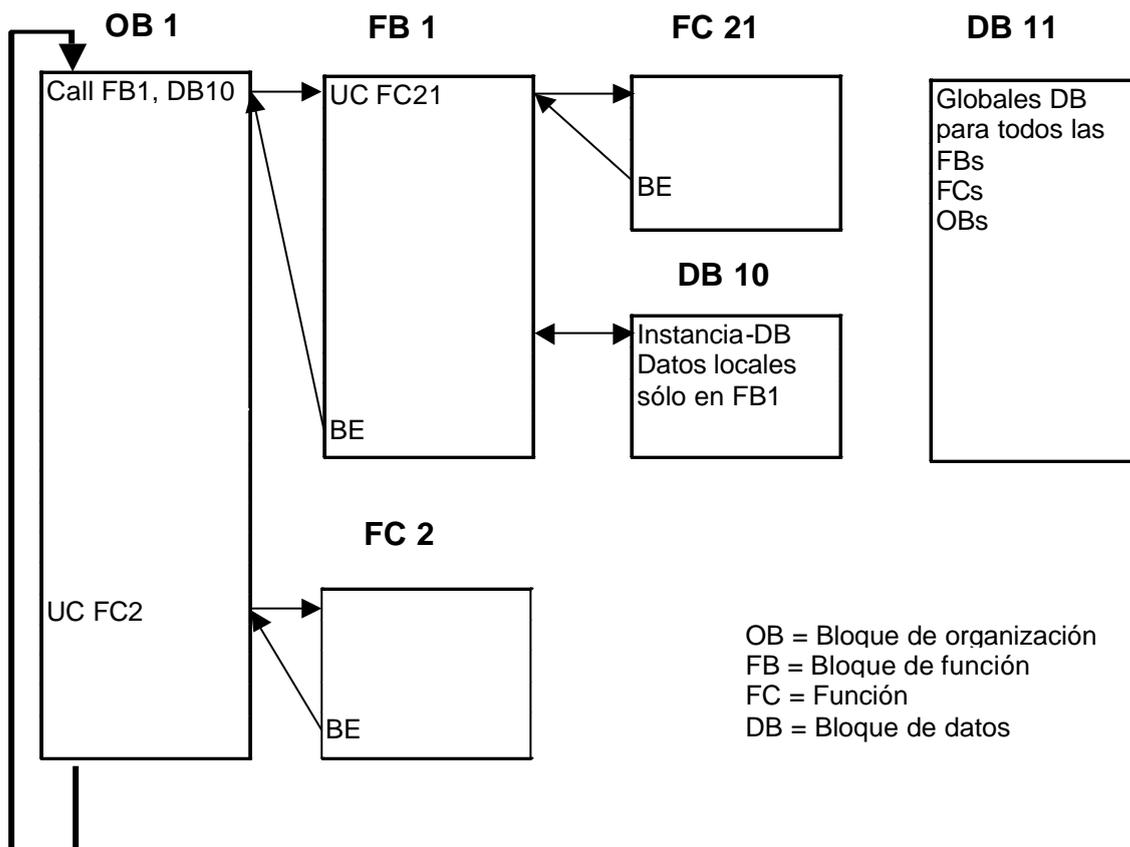
El tiempo que un aparato necesita para ejecutar una sólo vez todas las instrucciones, se llama tiempo de ciclo.



## 2.7.3 ESTRUCTURA DE LA EJECUCIÓN DEL PROGRAMA

Un programa de gran envergadura se subdivide y se reorganiza en pequeños y visibles bloques de programa. Esto tiene la ventaja de que los subprogramas se pueden testear y se pueden unir por medio de funciones.

El módulo de programa se ha de llamar con las instrucciones del módulo de llamamiento(Call xx / UC xx / CC xx). Al reconocerse el final del módulo, el programa ejecuta de nuevo una llamada al siguiente módulo.



## 2.7.4 BLOQUE DE USUARIO

STEP 7 ofrece para la estructuración de la programación los siguientes bloques de usuario:

- **OB (Bloque de organización):**

un OB se llama cíclicamente por el sistema operativo y configura con esto la interface entre el programa de usuario y el sistema operativo. En este OB se comunica a la unidad de control del AG a través de instrucciones de llamamiento, que bloque de programa debe ejecutarse.

- **FB (Bloque de función):**

El FB dispone de una organización en el área de memoria. Cuando se llama un FB se puede organizar un bloque de organización (DB). Para acceder a los datos en esta instancia DB se llama a un FB. Un FB puede organizar diferentes DBs. Otras FBs y FCs se pueden llamar en un bloque de función a través de instrucciones de llamamiento.

- **FC (Función):**

Una FC no posee ninguna organización en el área de memoria. Los datos locales de una función se pierden después de la ejecución de la función. Otras FBs y FCs se pueden llamar en una función a través de instrucciones de llamamiento.

- **DB (Bloque de datos):**

Las DBs se utilizan para suministrar espacio de memoria a las variables de datos. Existen dos tipos de bloques de datos. Los globales DBs, donde todos los OBs, FBs y FCs pueden leer datos almacenados o los que pueden escribir datos en el DB y instancias DBs, las cuales están ordenadas en una determinada FB.

## 2.7.5 BLOQUES DEL SISTEMA PARA FUNCIONES STANDARD Y FUNCIONES DEL SISTEMA

Los bloques de sistemas contienen funciones ya diseñadas, las cuales se guardan en la CPU. Estos bloques pueden ser llamados por el usuario para utilizarlos en el programa.

STEP 7 ofrece los siguientes bloques de sistema:

- **SFB (Bloque de funciones del sistema):**

Es un bloque de funciones integrado en el sistema operativo de la CPU y el usuario lo llama con el grupo de función.

- **SFC (Bloque de función):**

Es un bloque de funciones integrado en el sistema operativo de la CPU y el usuario lo llama con la función.

- **SDB (Bloque de datos del sistema):**

En el área de memoria del programa se programan diferentes Tools del STEP 7 (p.e.: configuración S7, configuración de la comunicación ... ), donde también se guardan los datos del sistema de automatización.

### 3. LENGUAJE DE PROGRAMACION STEP 7

#### 3.1 CONCEPTOS GENERALES DEL LENGUAJE DE PROGRAMACION STEP 7

A consecuencia del relevo de **SIMATIC S5** por **SIMATIC S7** se desarrolló un nuevo software de programación(**STEP7**), basado en la norma **IEC 61131**.

**STEP 7** funciona bajo **WINDOWS 95, 98** o **NT** y posee una superficie gráfica.



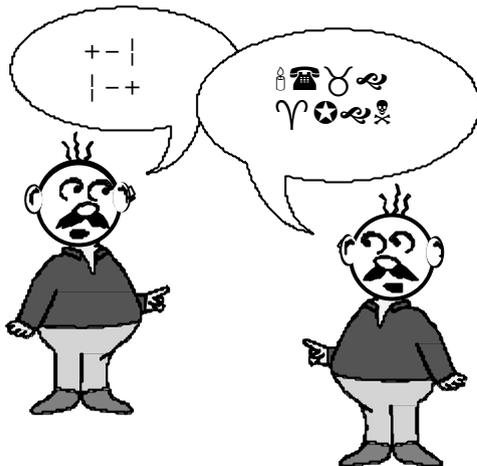
#### 3.2 CONVERSIÓN STEP 5 $\rightarrow$ STEP 7

El STEP 7 ofrece en el Tool 'S5 Datei konvertieren' la posibilidad de convertir la programación STEP 5 en STEP 7.



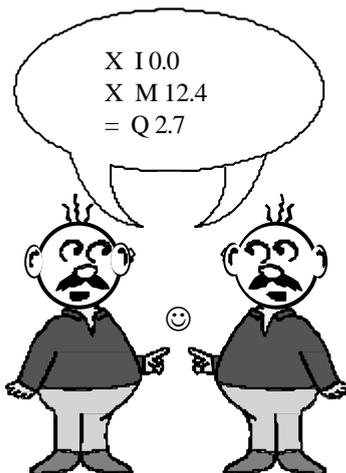
Para más información sobre la conversión de STEP 5 se ofrecen manuales de referencia STEP 7 o instrucciones de conversión en el 'Módulo 13 - Conversión STEP 5  $\rightarrow$  STEP 7'.

## 3.3 NORMA IEC 61131



Hasta hace poco, para la programación de los PLC, existían gran variedad de especificaciones de fabricante en los lenguajes y dialectos.

Esta variedad de lenguajes de programación provocaba problemas a la hora de querer comunicar diferentes PLCs entre ellos.



Con la norma IEC 61131 se crea por primera vez a nivel mundial unas bases uniformes para la programación de PLC. Bajo la presidencia de los USA se aprobó la comisión internacional electrónica (ICE) de la normalización de los PLC en 5 partes:

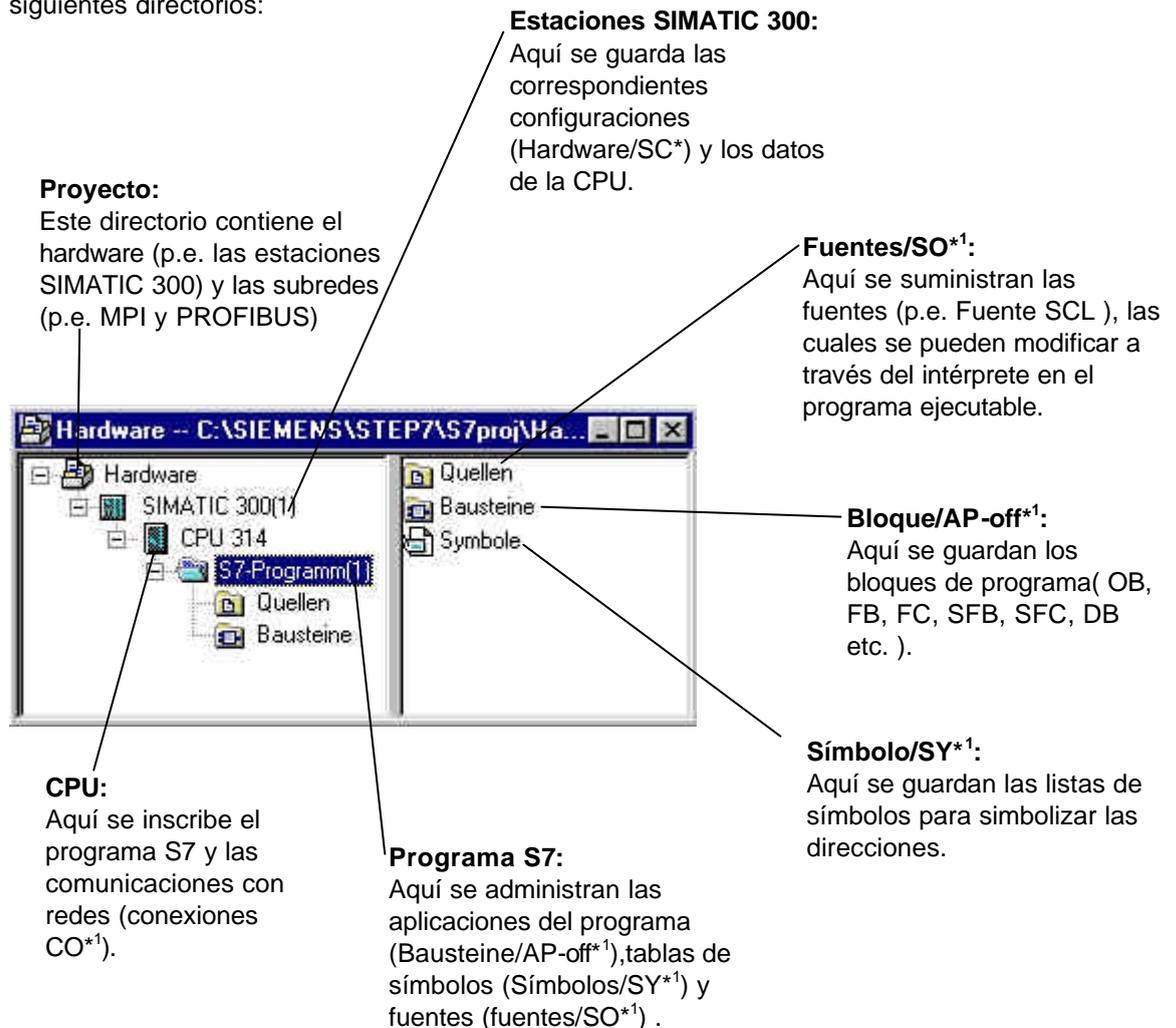
- Parte 1: Ordenanzas de los conceptos generales y propiedades de la funciones
- Parte 2: Exigencias funcionales elec. mec. en los aparatos.
- Parte 3: Cinco lenguajes de programación
- Parte 4: Instrucciones iguales para todas las fases del proyecto.
- Parte 5: Comunicación entre PLCs de diferentes fabricantes.

## 3.4 ESTRUCTURA DEL DIRECTORIO

La administración del documento se efectua en STEP 7 con '**SIMATIC Manager**'. Aquí se pueden p.e. copiar bloques de programas o seguir trabajando con otras herramientas.

La condición corresponde al standard WINDOWS 95/98/NT. Para esto p.e. existe la posibilidad de elegir en el menú cada componente clicando en el botón de la derecha del maus.

En STEP 7 se coloca cada proyecto en una sólida estructura. Los programas se guardan en los siguientes directorios:



### Símbolo/SY\*<sup>1</sup>:

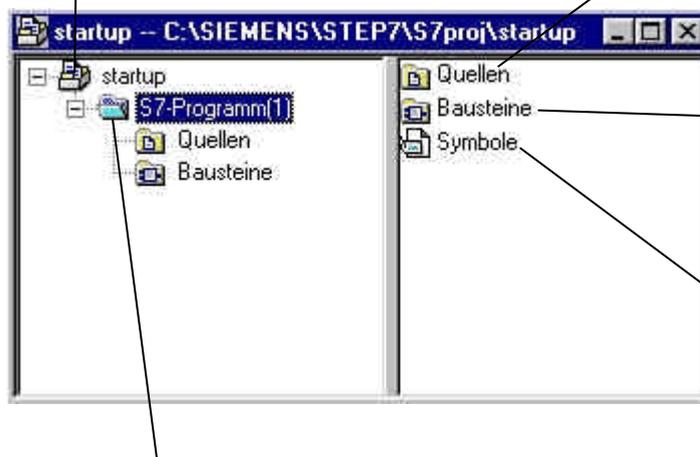
Aquí se guardan la lista de símbolos para simbolizar las direcciones.

En el archivo ,**SIMATIC 300 Station**' y ,**CPU**' se representa la configuración del hardware del PLC. Por consiguiente está también un proyecto en el cual se ven las especificaciones del hardware. Para escribir un proyecto independientemente de la configuración del hardware hay además la posibilidad de colocar un proyecto, que este directorio no lo contega.

Este tiene la siguiente estructura:

**Proyecto:**

Este directorio contiene el Hardware (p.e. las estaciones SIMATIC 300) y las subredes (p.e. MPI y PROFIBUS)



**Fuentes/SO\*1:**

Aquí se suministran las fuentes (p.e. Fuente SCL ), la cuales se pueden modificar a través del intérprete en el programa ejecutable.

**Bloque/AP-off\*1:**

Aquí se guardan los bloques de programa( OB, FB, FC, SFB, SFC, DB etc. ).

**Símbolo/SY\*1:**

Aquí se guardan las listas de símbolos para simbolizar las direcciones.

**Programa S7:**

Aquí se administran las aplicaciones del programa (Bausteine/AP-off\*1), tablas de símbolos (Símbolos/SY\*1) y fuentes (fuentes/SO\*1) .

**Símbolo/SY\*1:**

Aquí se guardan la lista de símbolos para simbolizar las direcciones.

## 3.5 CONFIGURAR Y PARAMETRIZAR

Durante la ejecución de la CPU se produce una configuración teórica y se guarda en un bloque de datos del sistema (SDB).

Con la Tool 'Hardware konfigurieren' se pueden modificar las discrepancias con la configuración real y con esto crear una nueva configuración de la estructura de control. Aparte de eso se puede también cargar una configuración existente en la CPU. Los bloques como la CPU pueden además simular parámetros (p.e. comportamiento de ejecución y ciclo de actuación de una CPU).

The screenshot shows the Siemens HW Config software interface. The window title is "HW Konfig - [Hardware konfigurieren: vernetz\_posiMot\_POSI\_CPU]". The interface includes a menu bar (Station, Bearbeiten, Einfügen, Zielsystem, Ansicht, Extras, Fenster, Hilfe) and a toolbar. On the left, a rack configuration list shows slots 1 through 11. The main area displays a network diagram with a PROFIBUS(1) DP-Mastersystem (180) connected to a (3) ET 200M and a (5) MICRO. Below the diagram is a detailed table for the rack configuration.

Steckplatz	Baugruppe	Bestellnummer	MPI-Adresse	E-Adresse	A-Adresse	Kommentar
1	PS 307 2A	6ES7 307-1BA00-0AA0				
2	CPU314(1)	6ES7 314-1AE01-0AB0	4			
3						
4	DI16xDC24V	6ES7 321-1BH00-0AA0		0...1		
5	DO16xDC24V/0.5A	6ES7 322-1BH00-0AA0			4...5	
6	CP 342-5(2)	6GK7 342-5DA00-0XE0	5	288...303	288...303	
7						
8						
9						
10						
11						

Drücken Sie F1, um Hilfe zu erhalten. NUM

### 3.6 LA INSTRUCCIÓN DE CONTROL

Para la ejecución de un controlador lógico programable se analiza el control de la tarea con instrucciones de control por separado. Una instrucción de control es la unidad independiente del control del programa. Representa la regla de trabajo para una unidad de control.

Denominaciones, características y símbolos para las instrucciones de control están fijadas en la norma DIN 19239.

Una instrucción de control se basa en lo siguiente:

Instrucción de control		
Parte de la operación	Parte del operando	
	Característica	Parámetro
<b>U</b>	<b>E</b>	<b>0.0</b>

#### 3.6.1 PARTE DE LA OPERACIÓN

La operación describe la ejecución de la función. Diferencia de la norma DIN 19 239:

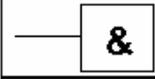
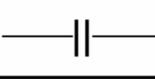
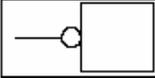
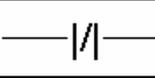
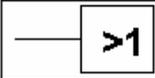
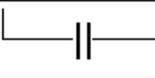
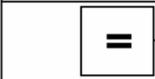
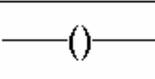


**Ejemplo para una operación digital:**

L	.....	Cargar
T	.....	Transferir
>I	.....	Comparación mayor de un número entero
==R	.....	Comparación igual de un número real
etc		

**Ejemplos de operaciones binarias:**

Resumen de la norma DIN 19 239

FUP	KOP	AWL	
		U	<b>AND</b>
		N	<b>NOT</b>
		O	<b>OR</b>
		=	<b>ASSIGNMENT</b>

etc

**Ejemplos para la organización de operaciones:**

CC	.....	Bloque de llamamiento condicional
UC	.....	Bloque de llamamiento incondicional
AUF	.....	Abrir bloque de datos
SPA	.....	Salto absoluto
SPB	.....	Salto condicionado
BEA	.....	Final de bloque incondicional
BEB	.....	Final de bloque condicional
etc		

## 3.6.2 PARTE DEL OPERANDO

La parte del operando contiene todas las instrucciones necesarias para la ejecución de la operación. Las características de los operandos incluyen la clase del operando.

Por ejemplo:

<p><b>E</b> para entradas</p> <p><b>A</b> para salidas</p> <p><b>M</b> para marcas</p> <p><b>L</b> para datos locales (Bloque interno de variables)</p> <p><b>T</b> para temporizadores</p> <p><b>Z</b> para contadores</p> <p><b>OB</b> para bloques de organización)</p> <p><b>FB</b> para bloques de funciones</p> <p><b>FC</b> para funciones</p> <p><b>DB</b> para bloques de datos</p> <p><b>SFB</b> para bloques de función del sistema</p> <p><b>SFC</b> para bloques de función</p> <p><b>L#..</b> para constantes de 32-Bit etc.</p>
--



El parámetro del operando indica la dirección del operando.

## 3.7 DIRECCIONAMIENTO

### 3.7.1 DIRECCIONAMIENTO SIMBÓLICO

El direccionamiento simbólico es una mejora servicial a la comprensibilidad. Permite clasificar determinadas direcciones absolutas en nombres simbólicos. Se puede p.e. nombrar la entrada E0.0 con el nombre END\_STOP y clasificarla con el tipo de dato BOOL. El nombre del símbolo sólo puede existir una vez. La organización se efectúa con el Tool editor de símbolos, el cual se puede ejecutar en el SIMATIC Manager.

### 3.7.2 DIRECCIONAMIENTO ABSOLUTO

En STEP 7 se ofrecen los siguientes tipos de direccionamiento absoluto:

- Direccionamiento inmediato
- Direccionamiento directo
- Direccionamiento indirecto por memoria

#### Direccionamiento inmediato:

Para el direccionamiento inmediato se codifica directamente el operando en la operación, es decir, la operación implica al operando.

#### Ejemplo:

SET	Se ejecuta el VKE (resultado lógico) a 1.
+D	Se suma el contenido del ACU 1 con el contenido del ACU2 y el resultado se deposita en el ACU1

#### Direccionamiento directo:

Para el direccionamiento directo se codifica la dirección del operando en la operación, es decir, el operando indica la dirección del valor que la operación debe ejecutar.

El operando contiene unas características del operando y unos parámetros y muestra directamente la dirección del valor.

#### Ejemplo:

U E 0.0	Ejecuta la unión UND de la entrada bit E 0.0 .
L EB 0	Carga la entrada byte EB 0 en el ACU 1.
= A 4.0	El VKE asigna la salida bit A 4.0.

## Direccionamiento indirecto por memoria:

Para el direccionamiento indirecto por memoria se ejecuta el direccionamiento indirecto del operando sobre un otro operando, el cual contiene la dirección del primero, es decir, el operando indica la dirección del valor que la operación debe ejecutar.

El operando consta de una operando característico y uno de los siguientes puntos:

- una palabra, la cual contiene un temporizador(T), un contador(Z), un bloque de datos(DB), una función(FC) o un bloque de función(FB).
- una doble palabra, la cual contiene exactamente la dirección de un valor dentro del área de memoria y que indica el operando característico.

La dirección del valor o el número indica el operando indirecto sobre el punto. La palabra o la doble palabra se pueden hallarse en una marca (M), bloque de datos (DB), instancia de bloque de datos (DI) o en datos locales (L).

### Ejemplo:

U E [MD 3]	Ejecuta la unión UND de la entrada bit. La dirección se encuentra en la marca de la doble palabra MD 3.
L EB [DID 4]	Carga la entrada byte en ACU1. La dirección se encuentra en la instancia de datos de la doble palabra DID 4.
AUF DB [MW 2]	Abre el bloque de datos. El número del bloque de datos se encuentra en la marca de la palabra MW 2.

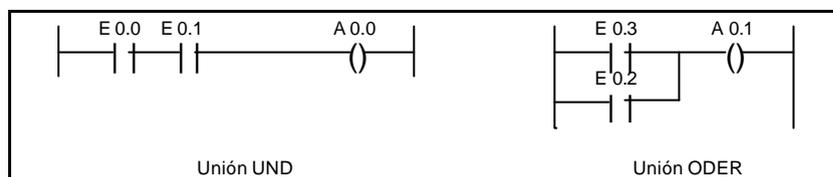
## 3.8 DESCRIPCIÓN DEL PROGRAMA

STEP 7 permite representar y programar un programa con las TOOL 'KOP/AWL/FUP Bloques de programación' en tres clases diferentes:

- Esquema de contactos KOP
- Diagrama de funciones FUP
- Lista de instrucciones AWL

### 3.8.1 ESQUEMA DE CONTACTOS KOP

El esquema de contactos es la representación gráfica de las tareas de control con símbolos (norma DIN 19239), método también habitual en USA. Tiene muchas similitudes con los esquemas de corriente, pero sin embargo esta representación no se ordena en sentido vertical, si no que se ordena en sentido horizontal.



### 3.8.2 DIAGRAMA DE FUNCIONES FUP ( PARA STEP 7 A PARTIR DE VERSION 3.x )

El diagrama de funciones es la representación gráfica de las tareas de control con símbolos (normas DIN 19239 y DIN 40 700). Las funciones individuales se representan a través de un símbolo con una función característica. En la parte izquierda del símbolo se ordenan las entradas y en la parte derecha las salidas.



### 3.8.3 LISTA DE INSTRUCCIONES AWL

En la lista de instrucciones se describe la tarea de control con instrucciones individuales de control. Las instrucciones de control (operación y operando) representan la tarea con mnemónicos de las designaciones de las funciones ( DIN 19 239).

<b>Parte de la operación:</b>	<b>Parte del operando:</b>		
	<b>Característica</b>	<b>Parámetro</b>	
U	E	0.0	Unión UND
U	E	0.1	
=	A	4.0	Unión ODER
O	E	0.2	
O	E	0.3	
=	A	4.1	

Cada clase de representación contiene propiedades específicas y determinadas limitaciones. El control de programas que se programan en KOP o en FUP se pueden traducir normalmente a lista de instrucciones AWL.

En la memoria de programa de los aparatos se guarda siempre el programa en AWL ( realmente en código máquina).

## 3.9 MARCAS

Para las uniones dentro de la regulación, cuyas señales no necesitan señales al exterior, se utilizan marcas.

Las marcas son elementos de memoria electrónicos (elementos de memoria R-S), los cuales pueden guardar los estados de señal "0" y "1".

Cada PLC dispone de un gran número de marcas para las uniones. Se programan como salidas. En las pérdidas de tensión del sistema se pierde el contenido guardado en las marcas.

### 3.9.1 MARCAS REMANENTES

Sin embargo una parte de estas marcas es remanente. A través del respaldo por pila en el PLC se soluciona la caída de tensión. Por lo tanto ambos estados lógicos se conservan.

#### Marca remanente

- En caso de desconexión conserva el último estado
- En caso de cambio de modo de trabajo "RUN>STOP" se conserva el último estado
- Se puede poner a cero excepto a través del program de usuario "AG>URLÖSCHEN"



A través de la utilización de la marca remanente se puede guardar el último estado de la máquina o equipo. Se puede arrancar de nuevo el equipo o la máquina en el estado en el que fue detenido. El área de remanencia se determina a través de la parametrización de la CPU en la Tool de configuración S7.

### 3.9.2 MARCAS NO REMANENTES



se resetean en el cambio de modo de trabajo „RUN>STOP,, así como al ser conectado.