

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

MÓDULO B4

Bloques de datos

Este documento fue suministrado por SIEMENS Siemens A&D SCE (Tecnología en Automatización y Accionamientos, Siemens A&D, coopera con la Educación) para formación. Siemens no hace ningún tipo de garantía con respecto a su contenido.

El préstamo o copia de este documento, incluyendo el uso e informe de su contenido, sólo se permite dentro de los centros de formación.

En caso de excepciones se requiere el permiso por escrito de Siemens A&D SCE (Mr. Knust: E-Mail: michael.knust@hvr.siemens.de). Cualquier incumplimiento de estas normas estará sujeto al pago de los posibles perjuicios causados. Todos los derechos quedan reservados para la traducción y posibilidad de patente.

Agradecemos al Ingeniero Michael Dziallas, a los tutores de las escuelas de formación profesional, así como a todas aquellas personas que nos han prestado su colaboración para la elaboración de este documento.

PÁGINA:

1.	Introducción.....	4
2.	Notas sobre los Bloques de Datos.....	6
3.	Generando Bloques de Datos.....	7

Los símbolos siguientes acceden a los módulos especificados:



Información



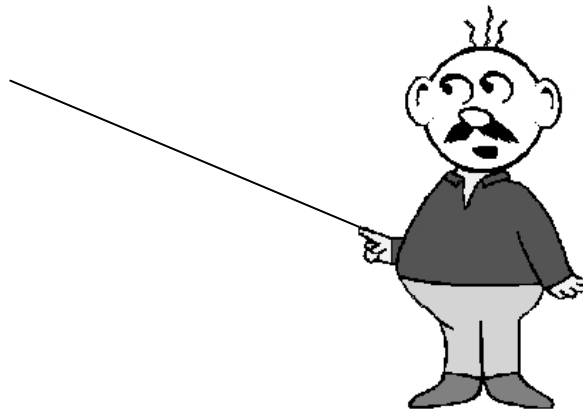
Programación



Ejercicio Ejemplo

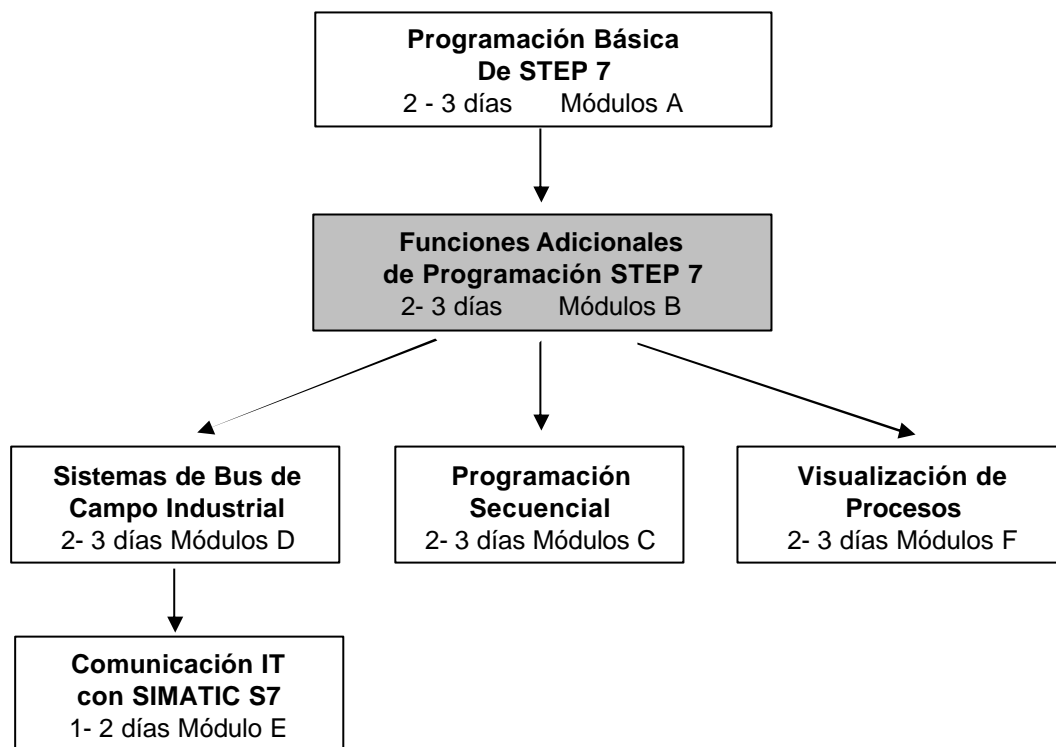


Notas



1. INTRODUCCIÓN

El módulo B4 pertenece al contenido de las **Funciones Adicionales de Programación STEP 7**.



Finalidad del Aprendizaje:

En este módulo, el lector aprenderá sobre como utilizar los Bloques de Datos para almacenar información.

- Generando Bloques de Datos
- Especificando la estructura de un Bloque de Datos
- Accediendo a un elemento de un DB en un programa STEP 7

Requirements:

Requisitos:

Para el correcto aprovechamiento de este módulo, se requieren los siguientes conocimientos:

- Conocimientos de uso de Windows 95/98/2000/ME/NT4.0
- Programación Básica de PLC con STEP 7 (Módulo A3 - 'Puesta en Marcha' programando PLC con STEP 7)
- Conocimientos Básicos de programación estructurada (Anexo I – Programación Básica de PLC – Programando con un SIMATIC S7-300)

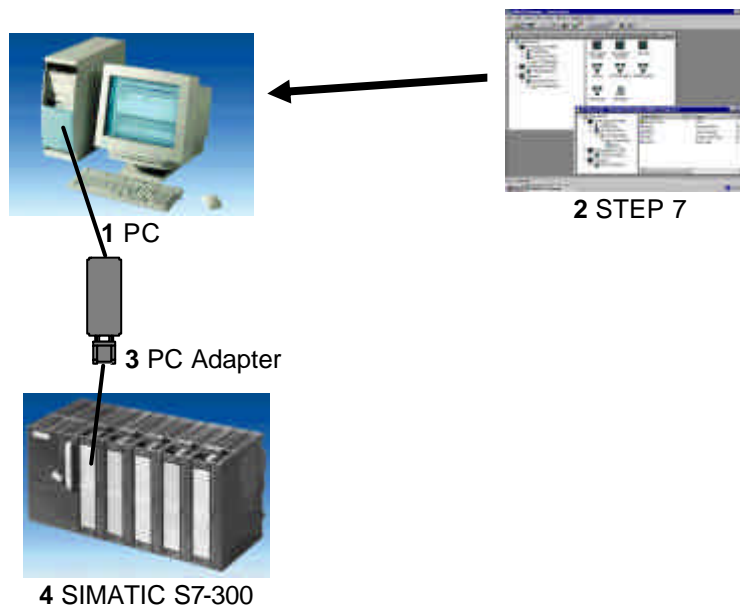
Indice	Notas	Generando Bloques de Datos
---------------	-------	----------------------------

Hardware y software Necesarios

- 1 PC, Sistema Operativo Windows 95/98/2000/ME/NT4.0 con
 - Mínimo: 133MHz y 64MB RAM, aprox. 65 MB de espacio libre en disco duro
 - Óptimo: 500MHz y 128MB RAM, aprox. 65 MB de espacio libre en disco duro
- 2 Software STEP 7 V 5.x
- 3 Interfase MPI para PC (p.e. PC- Adapter)
- 4 PLC SIMATIC S7-300 con al menos un módulo de entradas/salidas

Ejemplo de configuración:

- Fuente de Alimentación: PS 307 2A
- CPU: CPU 314
- Entradas Digitales: DI 16x DC24V
- Salidas Digitales: DO 16x DC24V / 0.5 A
-



2. NOTAS SOBRE LOS BLOQUES DE DATOS



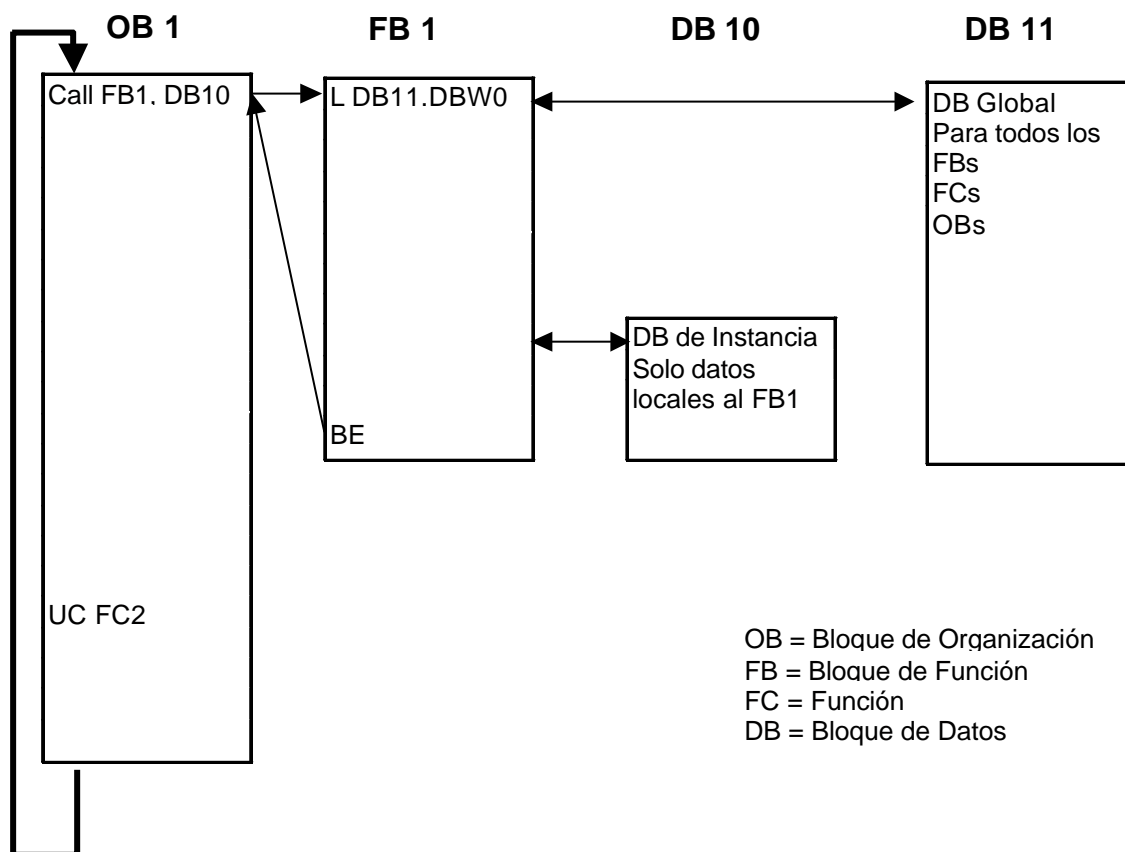
Los Bloques de Datos (DBs) pueden ser utilizados en el programa para salvar información en la CPU. Tienen una capacidad de memoria de hasta 8 KBytes (8192 Bytes).

Existen dos tipos de Bloques de Datos. DBs **Globales**, en los cuales todos los OBs, FBs y FCs pueden guardar o leer datos y DBs **de Instancia**, los cuales se encuentran asignados a un FB en particular.

En los DBs, se pueden almacenar diferentes tipos de datos (p.e. BOOL o WORD) de manera aleatoria.

La estructura de un DB se genera a través de la herramienta **Editor KOP/ AWL//FUP: Programar Bloques** de STEP7 .

En la estructura del programa STEP 7, los bloques de datos se encuentran de la siguiente manera:



3. GENERANDO BLOQUES DE DATOS



Los Bloques de Datos son generados y abiertos como cualquier otro tipo de bloque en la herramienta **Editor KOP/ AWL//FUP: Programar Bloques**. Sirven para almacenar datos e información de estados.

En el párrafo siguiente, se describe a través de un ejemplo la utilización de un bloque de datos global:



Por esto, los valores se seleccionarán con los interruptores 'S0' a 'S7' y visualizados en un módulo de 'Display' de salidas. La representación numérica del valor a almacenar se representa con estos interruptores, donde el interruptor S7 tendrá el mayor peso del valor y el interruptor S1 el menor peso.

El ejemplo se corresponde con las direcciones siguientes:

Entradas:

- Interruptor S0 = E 0.0
- Interruptor S1 = E 0.1
- Interruptor S2 = E 0.2
- Interruptor S3 = E 0.3
- Interruptor S4 = E 0.4
- Interruptor S5 = E 0.5
- Interruptor S6 = E 0.6
- Interruptor S7 = E 0.7

Salidas:

- Display = AW4

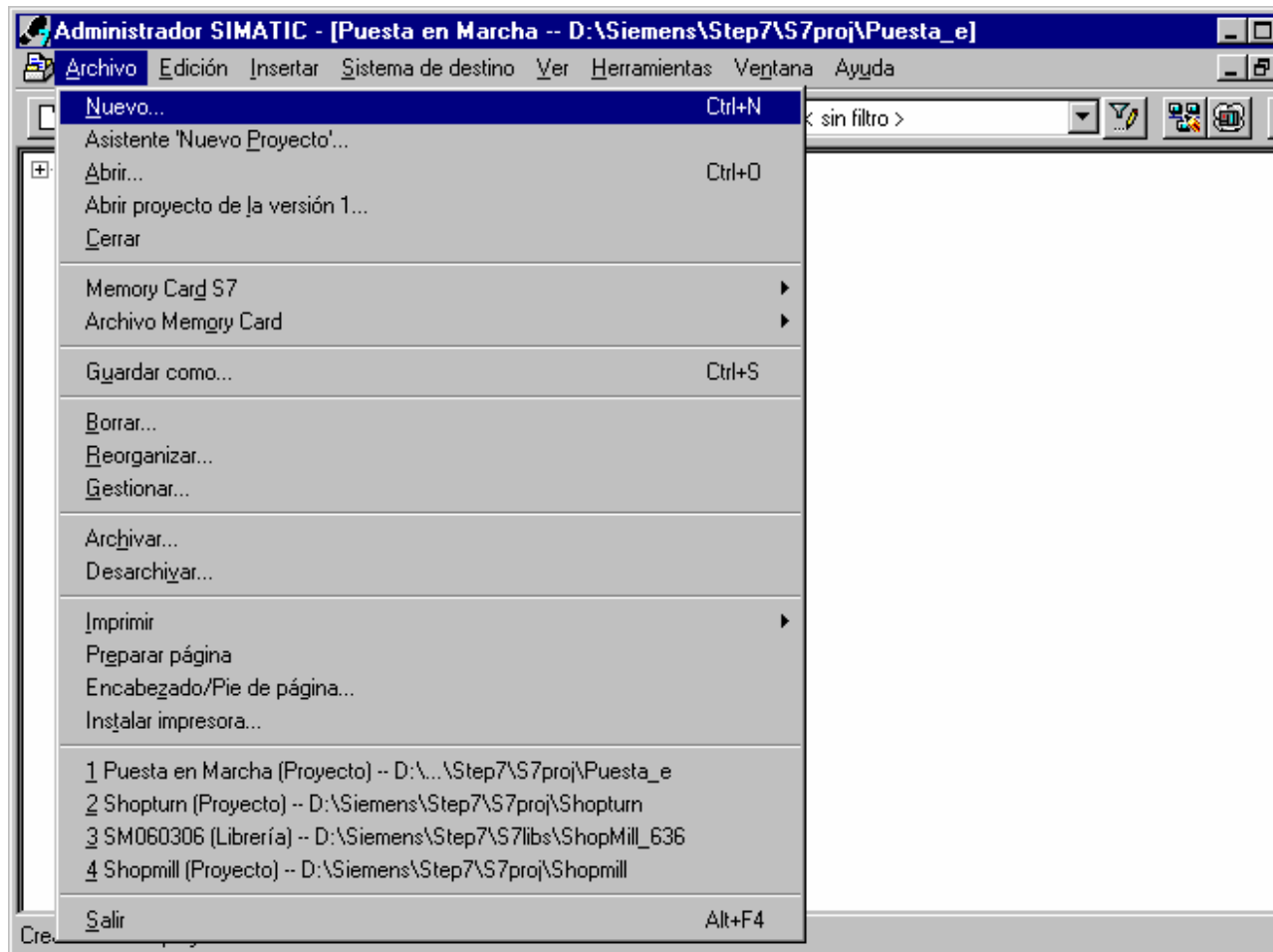
Para poder generar este ejemplo del programa, deben de los seguirse los siguientes pasos (Por eso el programa se distribuye con la creación de la configuración del hardware):

1. Llamar al **Administrador SIMATIC** con un doble click (→ Administrador SIMATIC)



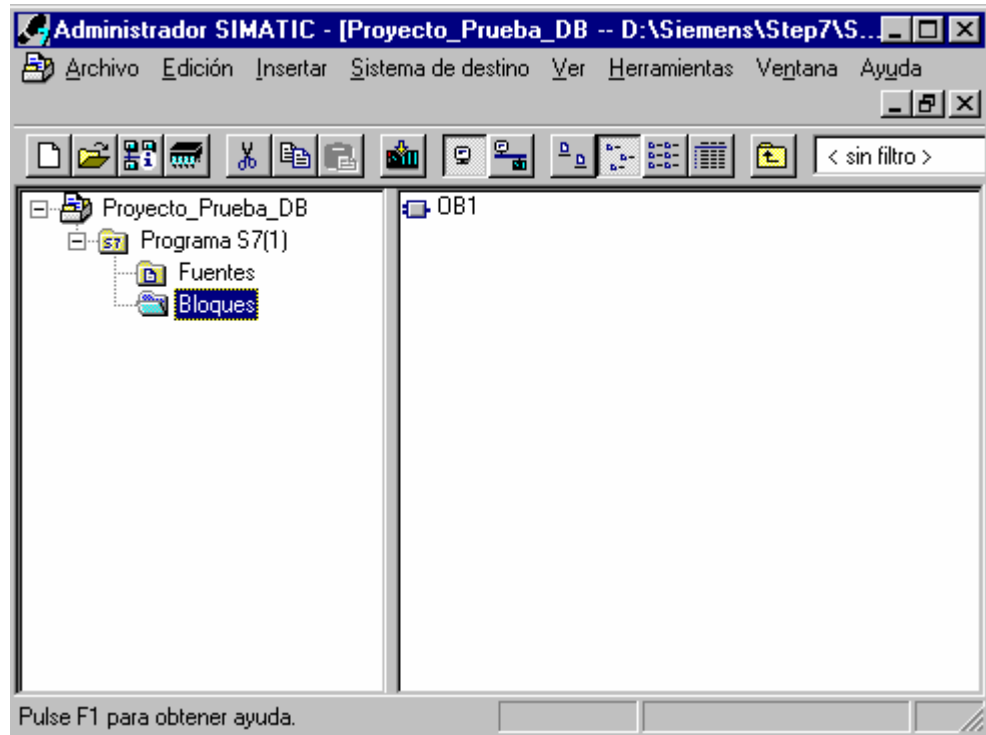
Administrador SIMATIC.lnk

2. Crear un proyecto nuevo (→ Archivo → Nuevo)

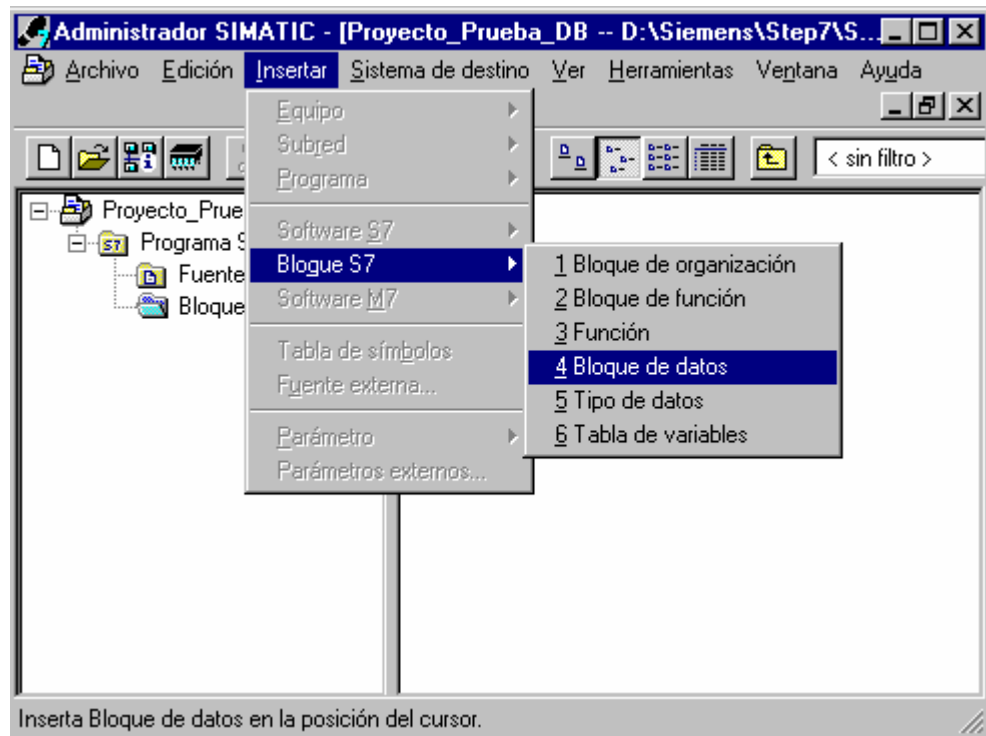




5. Seleccionar la carpeta **Bloques**. (→ Bloques)



6. Insertar un **Bloque de Datos** (→ Insertar → Bloque S7 → Bloque de Datos).





- Introducir el nombre **DB10**, seleccionar la opción DB Global y aceptar con **OK** (→ DB10 → DB Global → OK).

Propiedades - Bloque de datos

General - 1ª parte | General - 2ª parte | Llamadas | Atributos

Nombre y tipo: DB10 DB global

Nombre simbólico:

Comentario del símbolo:

Lenguaje: DB

Ruta del proyecto:

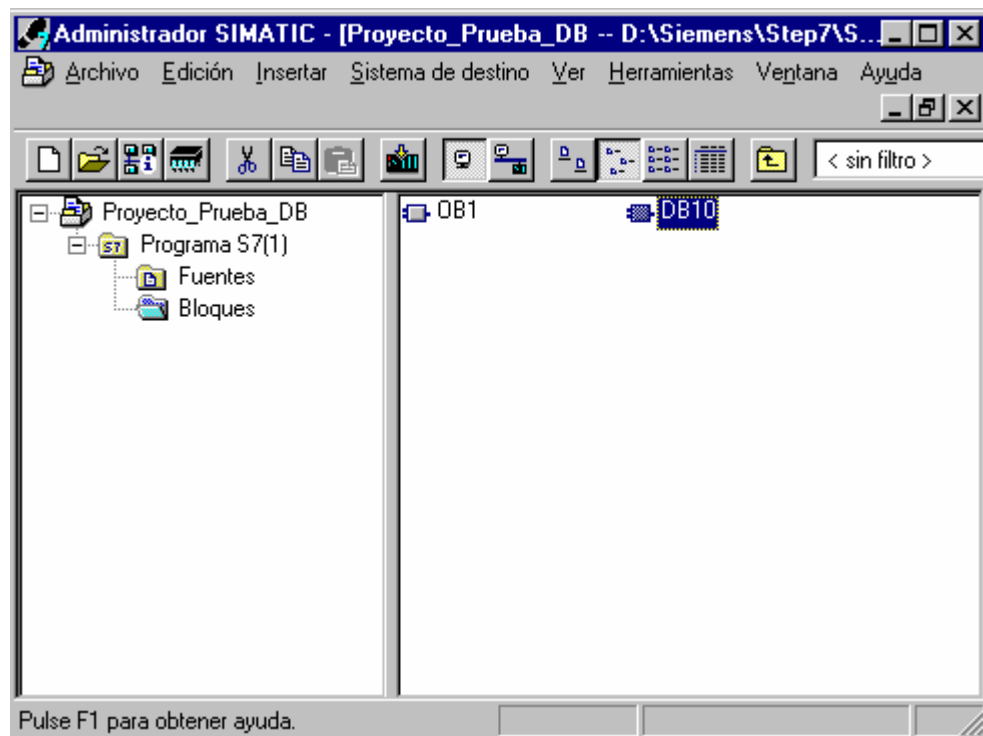
Ubicación del proyecto: D:\Siemens\Step7\S7proj\Proyecto

	Código	Interface
Fecha de creación:	10/01/2003 09:15:33	
Última modificación:	10/01/2003 09:15:33	10/01/2003 09:15:33

Comentario:

Aceptar Cancelar Ayuda





- Abrir el bloque de datos **DB10** con un doble click(→ DB10).



Nota: Los bloques de datos asignados a un FB se generan automáticamente en la llamada al FB. No tiene sentido, por tanto, generarlos antes de generar el FB. Por otro lado, los bloques de datos asociados a un tipo de datos (UDT) son DBs cuya estructura es una copia exacta del UDT.



10. Cada registro del DB tiene un **Nombre, Tipo, Valor Inicial y Comentario** (opcional). En el momento en que se pasa al campo siguiente, se le genera una dirección en le DB que no puede ser alterada.

El bloque de datos será salvado  y cargado el PLC . El selector de modo del PLC deberá estar en STOP! (→ Nombre→ Tipo → Valor Inicial → Comentario →  → )

Dirección	Nombre	Tipo	Valor inic	Comentario
0.0		STRUCT		
+0.0	Valor0	INT	0	Valor asignado a través del interruptor S0
+2.0	Valor1	WORD	W#16#0	Valor asignado a través del interruptor S1
+4.0	Valor2	WORD	W#16#0	Valor asignado a través del interruptor S2
+6.0	Valor3	WORD	W#16#0	Valor asignado a través del interruptor S3
+8.0	Valor4	WORD	W#16#0	Valor asignado a través del interruptor S4
+10.0	Valor5	INT	0	Valor asignado a través del interruptor S5
+12.0	Valor6	INT	0	Valor asignado a través del interruptor S6
+14.0	Valor7	BOOL	FALSE	Valor asignado a través del interruptor S7

Nombre simbólico con el que se accede a la dirección absoluta asignada

Tipo de dato asignado (ver tabla abajo) al registro.

Valor inicial en formato compatible con el tipo de dato seleccionado (opcional).

Comentario para documentación (opcional).

La dirección absoluta es asignada por STEP 7, cada vez que se valida un registro con ENTER.

El formato de la dirección puede ser **BIT, BYTE, WORD,**

Se puede acceder también a este elemento a través de su dirección absoluta (después de haber compilado y cargado el DB).



Nota:

Si el bloque de datos es declarado como de instancia a un FB, al abrir dicho DB ya aparece la tabla con la estructura del DB y no puede modificarse. Dicha estructura es una copia exacta de la tabla de declaraciones del FB.

Índice	Notas	Generando Bloques de Datos
--------	-------	-----------------------------------



Los datos en un bloque de datos deben de llevar asignado un tipo de datos.
 Los tipos de datos estándar STEP 7 se definen en la tabla mostrada abajo :

Tipo y descripción	Tamaño en Bits	Formato-Opciones	Rango y notación numérica (Valores máximo y mínimo)	Ejemplo
BOOL (Bit)	1	Texto Booleano	TRUE/FALSE	TRUE
BYTE (Byte)	8	Número Hexadecimal	B#16#0 a B#16#FF	B#16#10
WORD (Palabra)	16	Número Binario	2#0 a 2#1111_1111_1111_1111	2#0001_0000_0000_0000
		Número Hexadecimal	W#16#0 a W#16#FFFF	W#16#1000
		BCD	C#0 a C#999	C#998
		Número Decimal sin signo	B#(0,0) a B#(255,255)	B#(10,20)
DWORD (Doble Palabra)	32	Número Binario	2#0 a 2#1111_1111_1111_1111_1111_1111	2#1000_0001_0001_1000_1011_1011_0111_1111
		Número Hexadecimal	DW#16#0000_0000 a DW#16#FFFF_FFFF	DW#16#00A2_1234
		Número Decimal sin signo	B#(0,0,0,0) a B#(255,255,255,255)	B#(1,14,100,120)
INT (Entero)	16	Número Decimal con signo	-32768 a 32767	1
DINT (Int,32 bit)	32	Número Decimal con signo	L#-2147483648 a L#2147483647	L#1
REAL (Número en coma flotante)	32	Número en coma flotante IEEE	Máximo: +/-3.402823e+38 Mínimo: +/-1.175495e-38	1.234567e+13
S5TIME (Tiempo Simatic)	16	Tiempo S7 en pasos de 10 ms	S5T#0H_0M_0S_10MS a S5T#2H_46M_30S_0MS and S5T#0H_0M_0S_0MS	S5T#0H_1M_0S_0MS S5TIME#1H_1M_0S_0MS
TIME (Tiempo IEC)	32	Tiempo IEC en pasos desde 1ms, entero con signo	-T#24D_20H_31M_23S_648MS a T#24D_20H_31M_23S_647MS	T#0D_1H_1M_0S_0MS TIME#0D_1H_1M_0S_0MS
DATE (Fecha IEC)	16	Fecha IEC en pasos de 1 día	D#1990-1-1 a D#2168-12-31	DATE#1994-3-15
TIME_OF_DAY (Fecha y Hora)	32	Tiempo en pasos de 1ms	TOD#0:0:0.0 a TOD#23:59:59.999	TIME_OF_DAY#1:10:3.3
CHAR (Carácter)	8	Caracteres ASCII	'A', 'B' etc.	'B'

Indice	Notas	Generando Bloques de Datos
--------	-------	-----------------------------------



11. El contenido de los valores del bloque de datos se puede modificar. No es recomendable hacerlo a través del campo **Valor Inicial**. Se debe de hacer a través de la opción de menú **Datos** (→ Ver → Datos).

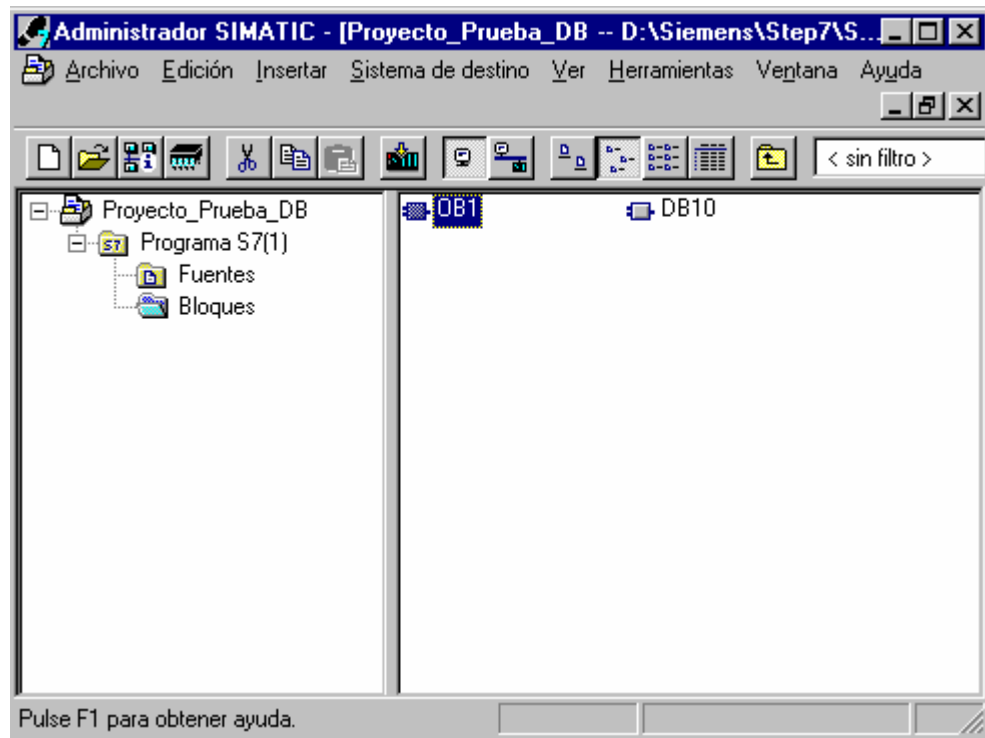
Dirección	Nombre	Tipo	Valor
0.0		STRUCT	
+0.0	Valor0	INT	0
+2.0	Valor1	WORD	W#16#
+4.0	Valor2	WORD	W#16#
+6.0	Valor3	WORD	W#16#
+8.0	Valor4	WORD	W#16#
+10.0	Valor5	INT	0
+12.0	Valor6	INT	0
+14.0	Valor7	BOOL	FALSE

12. Es ahora cuando se puede introducir un nuevo valor en el campo **Valor Actual** y salvarlo en el disco duro con así como cargarlo en la CPU con (→ Valor Actual → →).

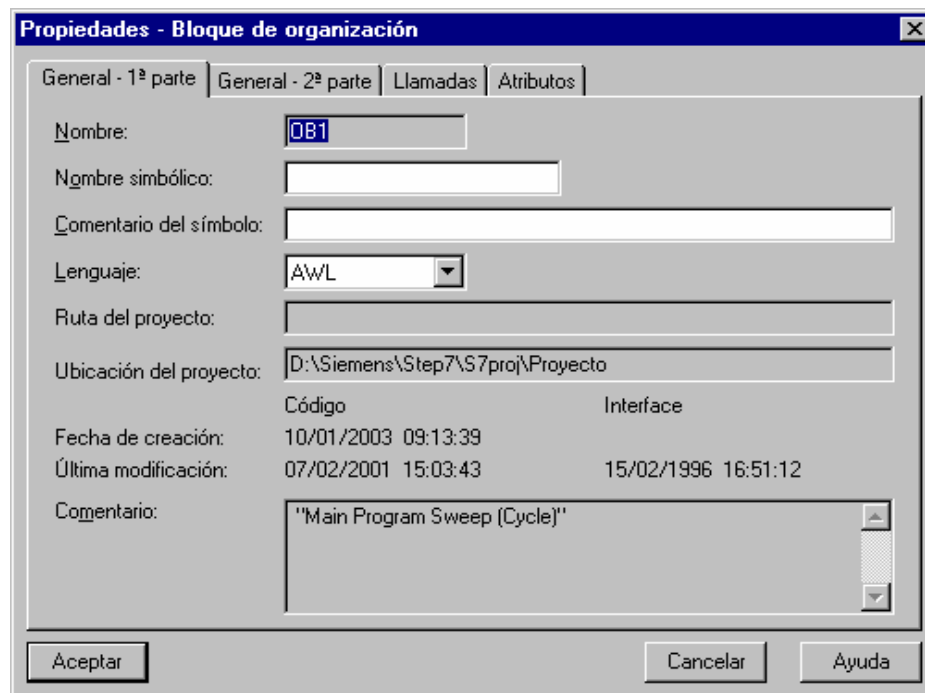
Dirección	Nombre	Tipo	Valor inicial	Valor actual	Comentario
0.0	Valor0	INT	0	0	Valor asignado a través del
2.0	Valor1	WORD	W#16#0	W#16#0	Valor asignado a través del
4.0	Valor2	WORD	W#16#0	W#16#0	Valor asignado a través del
6.0	Valor3	WORD	W#16#0	W#16#0	Valor asignado a través del
8.0	Valor4	WORD	W#16#0	W#16#0	Valor asignado a través del
10.0	Valor5	INT	0	0	Valor asignado a través del
12.0	Valor6	INT	0	0	Valor asignado a través del
14.0	Valor7	BOOL	FALSE	FALSE	Valor asignado a través del



- Para poder acceder a los registros del DB en el programa STEP 7, el **OB1** debe ser abierto con un doble click en el icono asociado del **Administrador SIMATIC** (→ Administrador SIMATIC → OB1).



- Aceptamos la pantalla con **OK** (→ OK).





Existen tres posibilidades de acceso a los datos:

1. Acceso por Direccionamiento Directo:

Se puede acceder a un elemento de un DB con las siguientes instrucciones:

Ejemplo:

L	DB 20. DBB2	Cargamos el byte de datos 2 del DB20 en el ACCU 1
L	DB 22. DBW4	Cargamos el byte de datos 4 del DB22 en el ACCU 1
A	DB 2. DBX5.6	Consultamos el estado del bit de datos 5.6 del DB2.

2. Acceso a un elemento de un DB que ya se encontraba abierto:

Para poder acceder a un elemento de datos, primero se debe abrir su correspondiente DB a través de las instrucciones OPN DB o OPN DI. Una vez hecho esto, varios bits de datos (DBX/DIX), bytes de datos (DBB/DIB), palabras de datos (DBW/DIW) o dobles palabras de datos (DBD/DID) pueden ser procesadas en diferentes operaciones binarias o aritméticas.

Al utilizar la instrucción OPN DI se abre un DB de Instancia. Esta instrucción puede utilizarse también con DBs globales, cuando 2 DBs permanecen abiertos al mismo tiempo.

Ejemplo:

OPN	DB 20	Abrimos el DB20
OPN	DI 22	Abrimos el DB22
L	DBW 0	Cargamos la palabra de datos 0 del DB20 en el ACCU 1
T	MW 1	Transferimos el contenido del ACCU 1 a la palabra de marcas 1
A	DIX 0.0	Operación AND sobre el bit de datos 0.0 del DB22 con
A	E 1.0	el bit de entradas 1.0
=	A 4.0	Asignamos el resultado al bit de salidas digitales 4.0

3. Acceso a datos de un DB de Instancia a través de la llamada de un Bloque de Función:

Los datos pueden ser transferidos a un DB de Instancia a través de la llamada a su correspondiente bloque de función con la instrucción CALL FB1, DB19. La asignación de las variables, que deberían de haber sido previamente definidas en la tabla de definiciones del FB, y cuyo contenido se almacena en el DB de instancia, son tomadas de las direcciones asignadas a los parámetros del FB (p.e. EW0, M 10.0 o AW4) con la instrucción CALL.

Ejemplo:





CALL	FB1, DB19	
CONTADOR:=	EW 0	La Variable CONTADOR corresponde con la dirección EW 0.
SALIDA:=	A 4.0	La variable SALIDA corresponde con la dirección A 4.0.

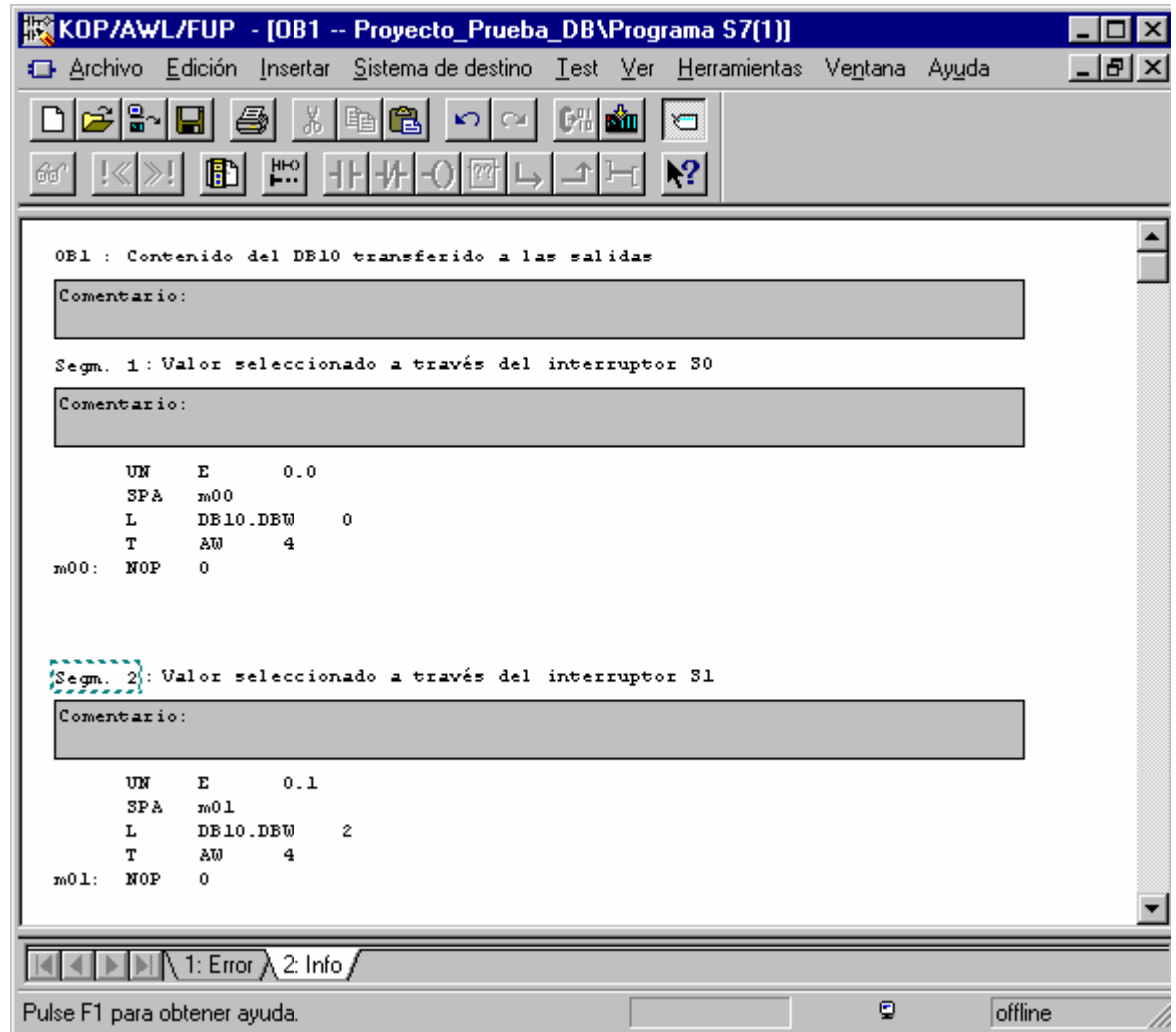


Nota: El formato de las variables del DB de Instancia y la dirección asignada en el parámetro del FB deben de ser iguales.

Indice	Notas	Generando Bloques de Datos
--------	-------	-----------------------------------



15. Con el **Editor KOP/ AWL//FUP: Programar Bloques**, se dispone de una herramienta para generar el programa STEP 7.
- La selección del elemento de datos se realiza a través de los diferentes interruptores, como se muestra en la imagen de abajo.
- Cuando se hayan contemplado todas las posibilidades de selección con los interruptores (S0 a S7) en el OB1, se deberá guardar en el disco duro  y cargado en el PLC . El interruptor de modo de la CPU deberá estar en STOP! (→  → )



16. Cuando el selector de modo del PLC se ponga en RUN, el programa comienza su ejecución. Activando sólo uno de los interruptores S0 a S7, el valor almacenado en el correspondiente registro del bloque de datos es visualizado en el módulo de salidas a través de la dirección AW4.

Indice	Notas	Generando Bloques de Datos
--------	-------	-----------------------------------