SISTEMAS INFORMATICOS INDUSTRIALES

PRÁCTICA 4: *PROGRAMACIÓN ESTRUCTURADA*

1 Objetivos:

- Aprender cómo configurar el hardware del equipo de control.
- Comprender los conceptos de programación estructurada.

2 Configuración del Hardware

Cada alumno debe de configurar el Hardware del equipo de control que hay en su puesto de trabajo:

En STEP 7 Archivo → Nuevo y se le indica el nombre de proyecto deseado(en el ejemplo *profibusdp*). Una vez creado el proyecto, hay que insertar un equipo SIMATIC 300 haciendo clic con el botón derecho del ratón y eligiendo en el menú emergente insertar nuevo objeto → Equipo SIMATIC 300 como muestra la figura 1.

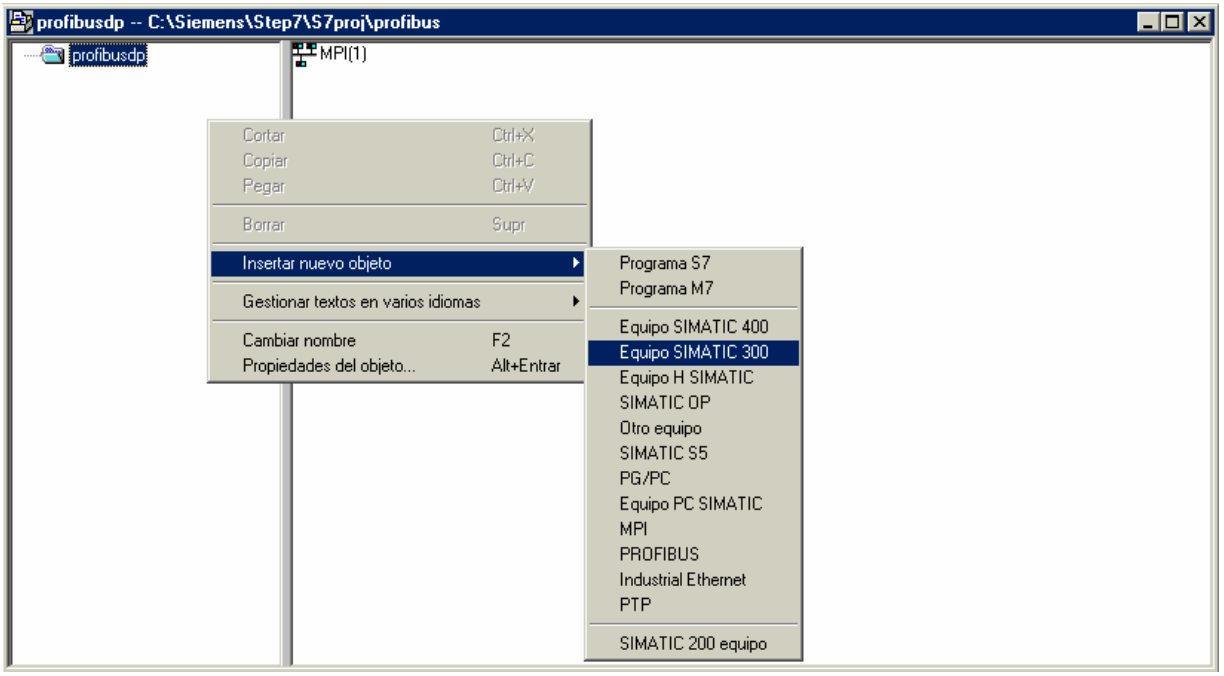


Figura 1. Inserción de un equipo SIMATIC 300 en un proyecto de Step 7

En el equipo SIMATIC 300 creado, hay que configurar el hardware que reproduzca el hardware del autómata con el que se va a realizar la práctica. Para ello se hace un doble click sobre el icono hardware dentro del equipo SIMATIC 300 Hardware.

En la aplicación configuración de hardware hay que configurar el equipo. Lo primero será insertar el bastidor donde se va a alojar el autómata. En el catálogo(Figura 2) existente la aplicación configurar hardware, se selecciona SIMATIC 300 → BASTIDOR 300 → Perfil Soporte. Una vez insertado el bastidor hay que indicar que hay conectado a cada uno de los slots del bastidor:

- En el slot 1 insertamos la fuente de alimentación SIMATIC 300 → PS-300 → PS 307 5A.
- En el slot 2 insertamos la CPU SIMATIC 300 → CPU-300 → CPU xxxx
- En el slot 3 está reservado para el módulo interface.
- En el slot 4 insertamos ……

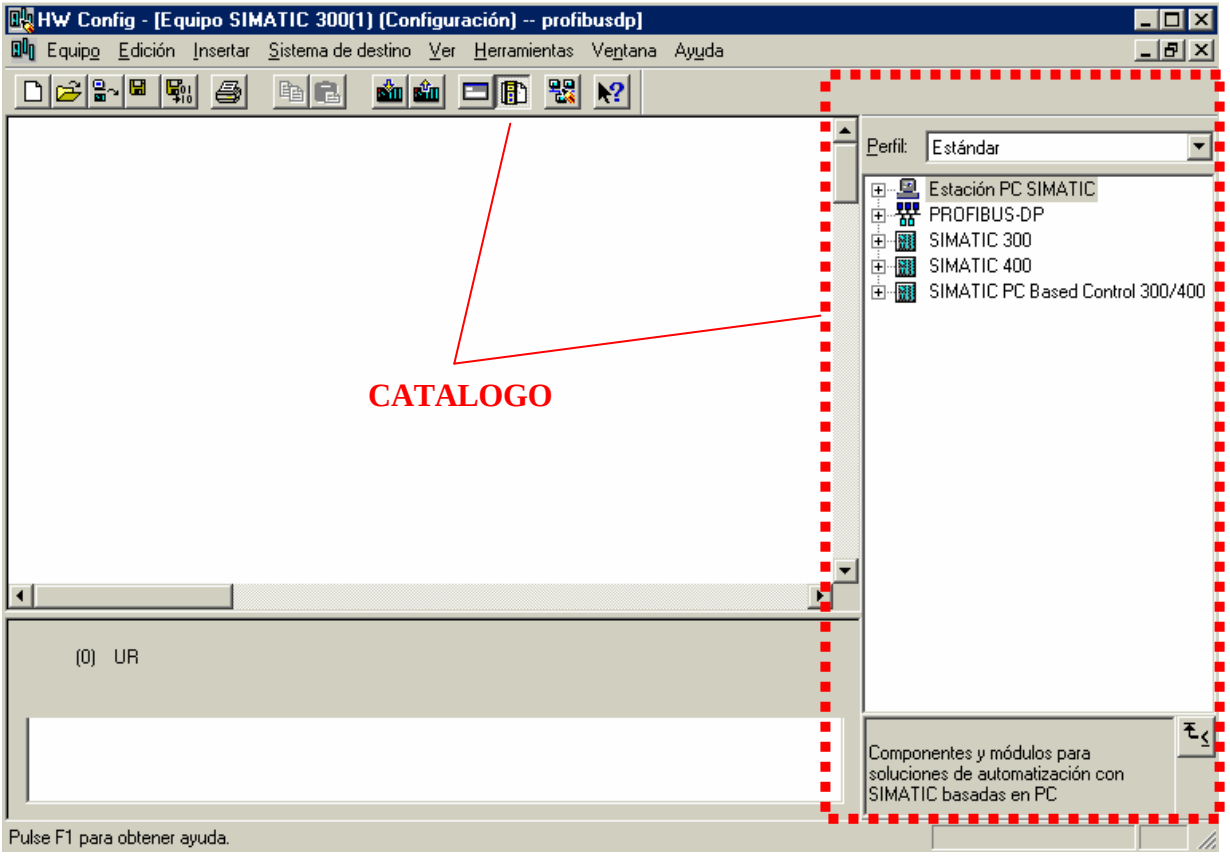


Figura 2. Catalogo de la aplicación configuración de hardware.

3 Programación Simbólica

Direcciones absolutas

Todas las entradas y salidas tienen asignada, de forma estándar, una dirección absoluta. Esta dirección se adopta directamente en el software. Las direcciones absolutas se pueden sustituir por nombres simbólicos definidos por el usuario.

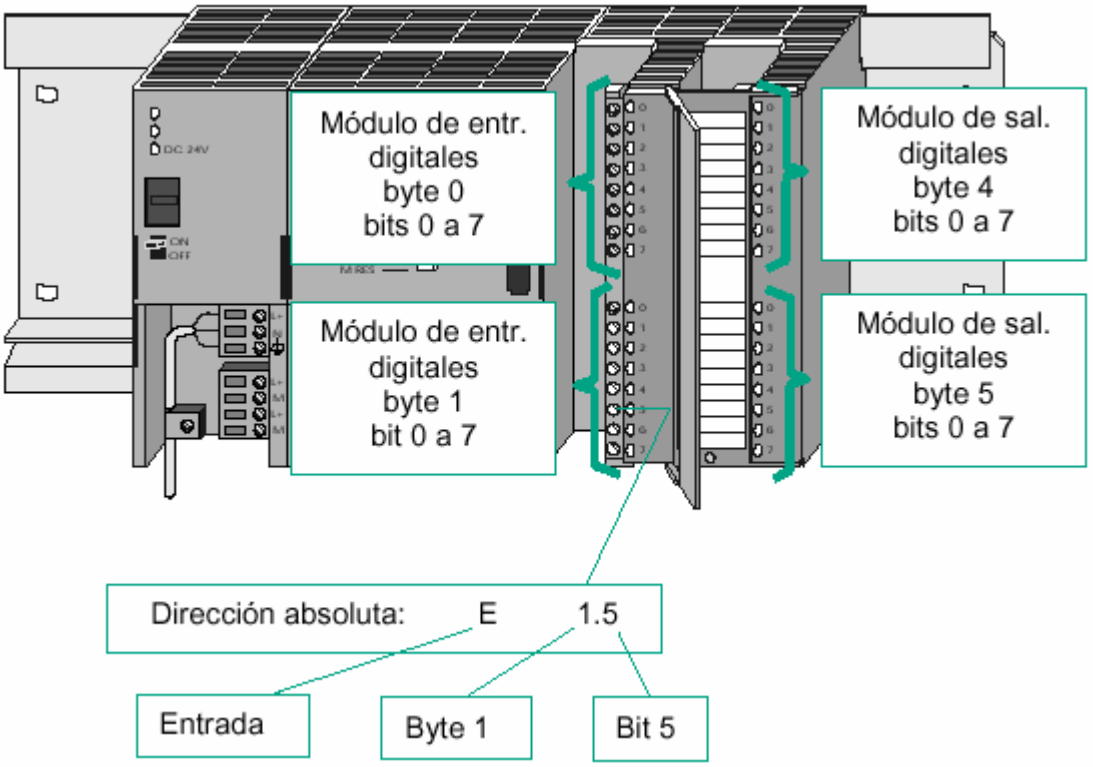


Figura 3. Direcciones absolutas

Programación simbólica

En la tabla de símbolos asigne, a todas las direcciones absolutas que utilizará más tarde en el programa un nombre simbólico y un tipo de datos; p.ej.: el símbolo 'Pulsador 1' a la entrada 'E0.1'. Estos nombres, denominados variables globales, son válidos para todos los componentes del programa. La programación simbólica permite mejorar considerablemente la legibilidad del programa S7.

Uso del editor de símbolos

Navegue en la ventana del proyecto hasta la carpeta **Programas S7 (1)** y abra el icono **Símbolos** con un doble clic.

Por el momento nuestra tabla de símbolos contiene sólo el bloque de organización OB1.

Haga clic en el símbolo **Cycle Execution** y sobreescríbalo para nuestro ejemplo con "Programa principal“. En la línea 2 introduzca "Lámpara verde“ y "A 4.0“ o la dirección correspondiente de la primera salida libre. El tipo de datos se añade automáticamente. Para introducir el comentario del símbolo haga clic en la columna 'Comentario' de la línea 1 ó 2. Al finalizar una línea pulse la tecla **Entrar** para insertar otra. En la línea 3 introduzca "Lámpara roja" y "A 4.1“ y confirme con **Entra**r.

Para evitar copiar de nuevo todos los nombres simbólicos de la memoria del autómata que se van a utilizar, se copiará la tabla de símbolos de un proyecto ejemplo. Para ello, abrir el proyecto ejemplo: ZEs01_01_STEP7__AWL_1-9 (Ver Figura 4).

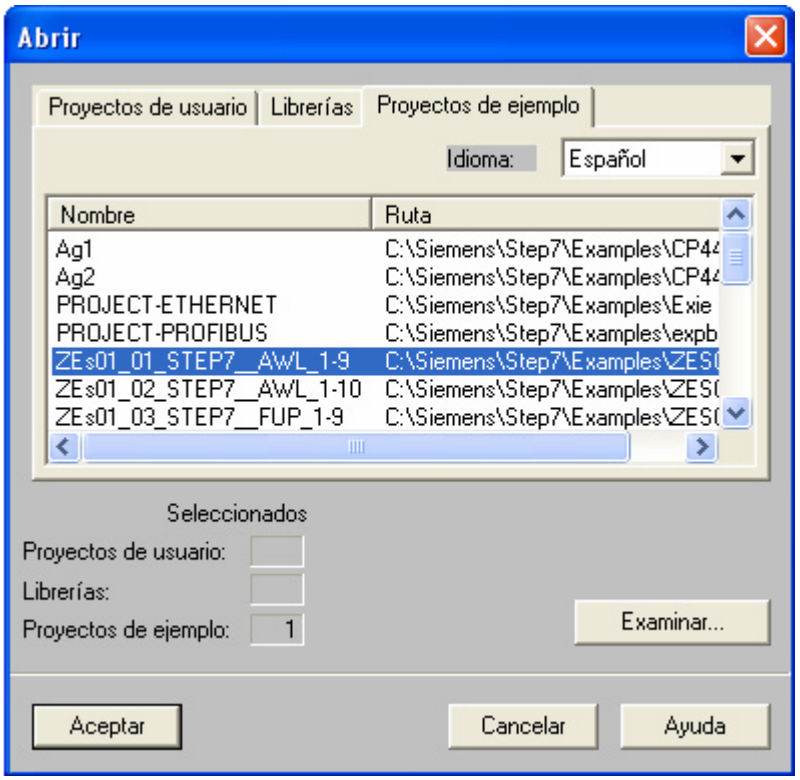


Figura 4. Ventana de Diálogo Abrir. Pestaña Proyectos de Ejemplo.

Una vez abierto copiar la tabla de símbolos de su programa **Programa S7(1)** que tiene el aspecto que se muestra en la Figura 5.

4 Crear un programa con FBs y DBs

Crear y abrir un bloque de función

El bloque de función (FB) está subordinado al bloque de organización y contiene una parte del programa a la que se puede acceder siempre que se desee desde el OB1. Los parámetros formales y los datos estáticos del bloque de función se guardan por separado en un bloque de datos DB asignado al FB.

Navegue hasta la carpeta **Bloques** y ábrala. Haga clic con la tecla derecha del ratón en la mitad derecha de la ventana. El menú contextual que aparece al pulsar la tecla derecha del ratón contiene los comandos más utilizados de la barra de menús. Inserte el objeto **Bloque de función**.

ZEs01_01_STEP7__AWL_1-9\SIMATIC 3...\Símbolos

	Símbolo	Dirección	Tipo de datos	Comentario
1	Arrancar_MD	E 1.4	BOOL	Arrancar el motor diesel
2	Arrancar_MG	E 1.0	BOOL	Arrancar motor de gasolina
3	Automático ON	E 0.5	BOOL	para la función de memorización (activar)
4	Conectar_ventilador_MD	A 5.6	BOOL	Comando para conectar ventilador motor diese
5	Conectar_ventilador_MG	A 5.2	BOOL	Comando para conectar ventilador motor gaso
6	Datos_globales	DB 3	DB 3	Bloque de datos globales
7	Debe_revol_alcanzado_MD	A 5.5	BOOL	Indicador motor diesel alcanza núm. revol. teó
8	Debe_revol_alcanzado_MG	A 5.1	BOOL	Indicador motor gasolina alcanza núm. revol. t
9	Diesel	DB 2	FB 1	Datos del motor diesel
10	Fallo_DM	E 1.6	BOOL	Fallo del motor diesel
11	Fallo_MG	E 1.2	BOOL	Fallo del motor de gasolina
12	Gasolina	DB 1	FB 1	Datos del motor de gasolina
13	Lámpara roja	A 4.1	BOOL	Resultado de la consulta O
14	Lámpara verde	A 4.0	BOOL	Resultado de la consulta Y
15	Manual ON	E 0.6	BOOL	para la función de memorización (desactivar)
16	MD_ON	A 5.4	BOOL	Comando para arrancar el motor diesel
17	MG_ON	A 5.0	BOOL	Comando para arrancar el motor de gasolina
18	Modo automático	A 4.2	BOOL	Salida con memorización
19	Motor	FB 1	FB 1	Control del motor
20	Parar_MD	E 1.5	BOOL	Parar motor diesel
21	Parar_MG	E 1.1	BOOL	Parar el motor de gasolina
22	Programa principal	OB 1	OB 1	Este bloque contiene el programa de usuario
23	Pulsador 1	E 0.1	BOOL	para la consulta Y
24	Pulsador 2	E 0.2	BOOL	para la consulta Y
25	Pulsador 3	E 0.3	BOOL	para la consulta O
26	Pulsador 4	E 0.4	BOOL	para la consulta O
27	Retardo_descon_MD	T 2	TIMER	Retardo para el ventilador del motor diesel
28	Retardo_descon_MG	T 1	TIMER	Retardo para el ventilador del motor de gasolin
29	Revol_reales_MD	MW 4	INT	Revoluciones reales del motor diesel
30	Revol_reales_MG	MW 2	INT	Revoluciones reales del motor de gasolina
31	Ventilador	FC 1	FC 1	Control del ventilador
32				

Figura 5. Tabla de símbolos.

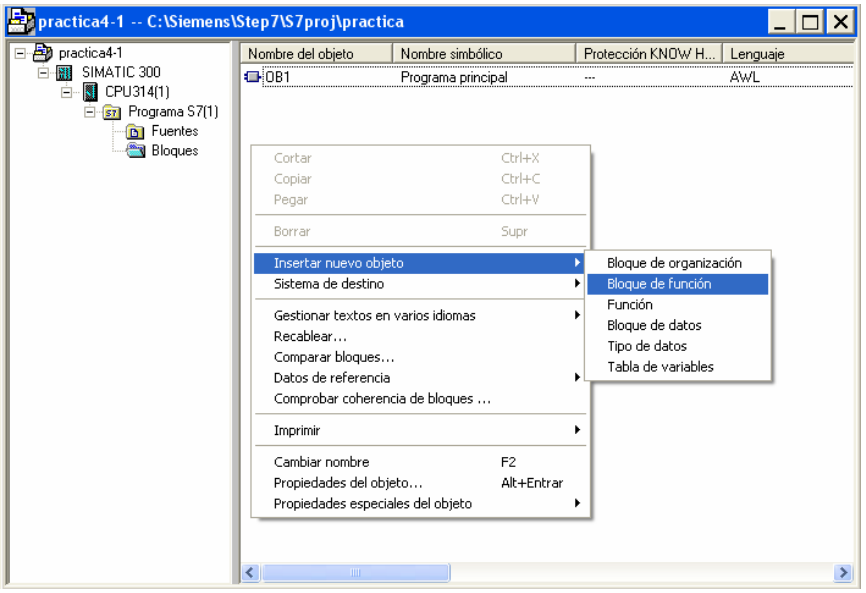


Figura 6. Crear un Bloque de Función.

Haciendo doble clic en el FB1 se abrirá la ventana KOP/AWL/FUP. En el cuadro de diálogo "Propiedades– Bloque de función“ seleccione el lenguaje en el que desea programar el bloque (AWL), active la opción **FB multiinstancia** y pulse **Aceptar** para que se guarden los ajustes realizados. El bloque de función **FB1** se insertará en la carpeta 'Bloques'.

Programar el FB1 en AWL

Ahora se mostrará cómo programar un bloque de función que controla y supervisa, a modo de ejemplo, un motor de gasolina y un motor diesel, a saber: con un bloque de datos distinto para cada motor.

Las señales "específicas del motor" son transferidas por el bloque de organización al bloque de función en calidad de parámetros del bloque, por lo que deben figurar en la tabla de declaración de variables como parámetros de entrada y salida (declaración "in" y "out").

Primero rellenaremos la tabla de declaración de variables

La ventana KOP/AWL/FUP está abierta y el comando **Ver > AWL** (lenguaje de programación) activado.

Fíjese en el encabezado; verá que aparece el FB1, ya que ha abierto la ventana del programa haciendo doble clic en el mismo (Ver Figura 7).

Introduzca las declaraciones siguientes en la tabla de declaración de variables. Para ello haga clic en una de las celdas y escriba el nombre y el comentario que ve en la Figura 8.

Seleccione el tipo del menú contextual **Datos simples** que aparece al pulsar la tecla derecha el ratón. Pulse **Entra**r; así el cursor saltará a la columna siguiente y se insertará otra línea.

Dirección	Declaración	Nombre	Tipo	Valor inicial	Comentario
0.0	in	Switch_On	BOOL	FALSE	Conectar motor
0.1	in	Switch_Off	BOOL	FALSE	Desconectar motor
0.2	in	Failure	BOOL	FALSE	Fallo del motor; conduce a una desconexión
2.0	in	Actual_Speed	INT	0	revoluciones reales del motor
4.0	out	Engine_On	BOOL	FALSE	Conexión del motor
4.1	out	Preset_Speed_Reached	BOOL	FALSE	Número de revoluciones alcanzado
	in_out				
6.0	stat	Preset_Speed	INT	1500	Número de revoluciones solicitado

Figura 8. Tabla de declaración de variables de FB1

Programación de la FB1

En los segmentos de FB1 van a programarse dos secuencias de control básicas: el arranque y paro del motor y la supervisión del número de revoluciones. Podemos resumir estas secuencias de control en las siguientes especificaciones:

- **¿Cuándo se arranca y cuándo se para el motor?** Cuando el estado de señal de la variable #Arrancar es "1" y el de la variable "Modo automático" es "0", arranca el

motor. Para ello es imprescindible que se niegue (contacto normalmente cerrado) la variable "Modo_automático". Cuando el estado de señal de la variable #Parar es "1" o cuando el de la variable #Fallo es "0", se para el motor. Para ello es imprescindible que se niegue la variable #Fallo (#Fallo es una señal "activa a cero"; es decir, normalmente está a "1", sólo en el caso de que se produzca un fallo se pone a "0").

- **¿Cómo supervisa el comparador las revoluciones del motor?** El comparador compara la variable #Actual_Speed (#Revol_reales) con la variable #Preset_Speed (#Revol teóricas) y arroja el resultado por medio de la variable #Preset_Speed_Reached (#Revol_alcanzado) (el estado de señal es "1").

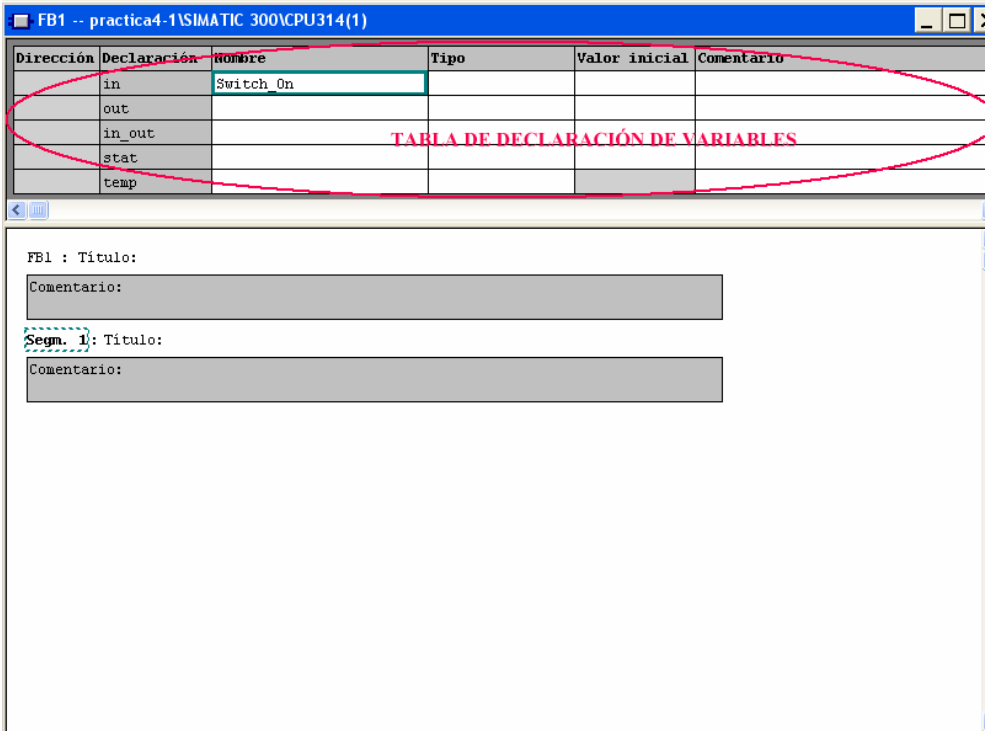
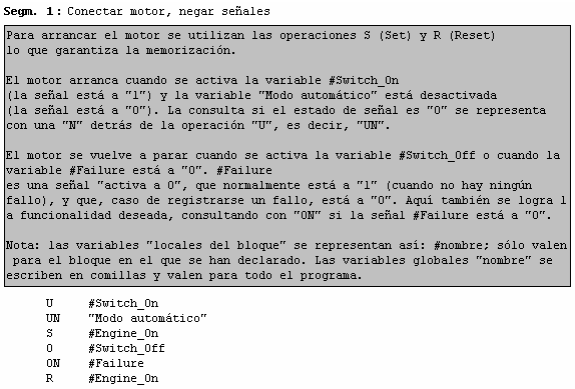
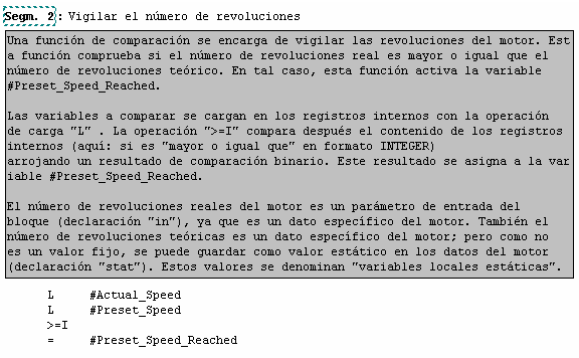


Figura 7. Bloque de Funcióne FB1.

Por tanto el segmento 1 puede contener la secuencia de arranque y paro del motor:



y el segmento 2 la supervisión de las revoluciones del motor:



5 Crear bloques de datos de instancia y modificar valores actuales

Se ha programado el bloque de función FB1 ("Motor") definiendo, entre otros, parámetros específicos del motor en la tabla de declaración de variables.

Para poder programar más tarde la llamada (CALL) del OB1 al FB, debemos crear el bloque de datos correspondiente. Generalmente se asigna a cada FB un bloque de datos de instancia (DB).

El FB controla y supervisa un motor de gasolina y un motor diesel. El número teórico de revoluciones del motor de gasolina se registra en un DB y el del motor diesel en otro, por lo que tendremos que modificar el valor actual en cada caso (#N°_teór_rev).

Nos podemos ahorrar un tiempo valioso si programamos un único bloque de función para ambos DB.

Inserte un **bloque de datos** con ayuda del menú contextual que aparece al pulsar la tecla derecha del ratón (De la misma forma que se insertó el bloque de función).

Confirme todos los ajustes del cuadro de diálogo "Propiedades" con **Aceptar** (Recuerde que se trata de un DB de instancia). Así se insertará el bloque de datos DB1 en el proyecto. Abra el **DB1** con un doble clic.

Así se abrirá la ventana KOP/AWL/FUP del programa con los datos de la tabla de declaración de variables del FB1. Ahora debemos introducir en el DB1 los datos específicos del motor de gasolina. Pero primero elegiremos el comando de menú **Ver > Datos**.

Introduzca, para el motor de gasolina, el valor "1500" en la columna 'Valor actual' (de la línea "N°_teór_rev"). Así definirá el número máximo de revoluciones para este motor. Guarde el DB1 y cierre la ventana del programa (Ver Figura 9).

Cree el bloque de datos DB2 del mismo modo que hemos creado el DB1. Para el motor diesel introduzca el valor "1200" como valor actual (Ver Figura 10).

Dirección	Declaración	Nombre	Tipo	Valor inicial	Valor actual	Comentario
0.0	in	Switch_On	BOOL	FALSE	FALSE	Conectar motor
0.1	in	Switch_Off	BOOL	FALSE	FALSE	Desconectar motor
0.2	in	Failure	BOOL	FALSE	FALSE	Fallo del motor; conduce a una desconexión
2.0	in	Actual_Speed	INT	0	0	revoluciones reales del motor
4.0	out	Engine_On	BOOL	FALSE	FALSE	Conexión del motor
4.1	out	Preset_Speed_Reached	BOOL	FALSE	FALSE	Número de revoluciones alcanzado
6.0	stat	Preset_Speed	INT	1500	1500	Número de revoluciones solicitado

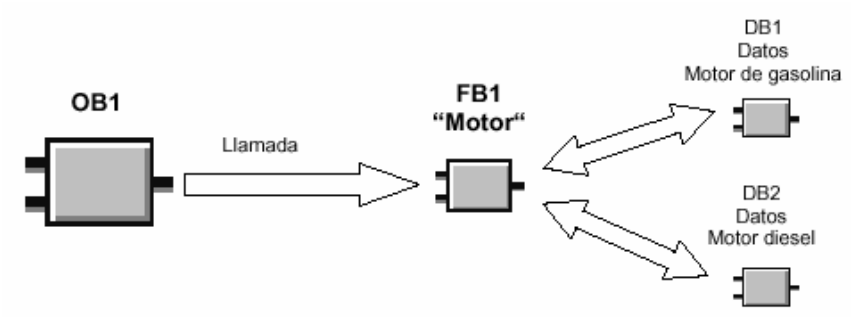
Figura 9. Bloque de datos DB1

Dirección	Declaración	Nombre	Tipo	Valor inicial	Valor actual	Comentario
0.0	in	Switch_On	BOOL	FALSE	FALSE	Conectar motor
0.1	in	Switch_Off	BOOL	FALSE	FALSE	Desconectar motor
0.2	in	Failure	BOOL	FALSE	FALSE	Fallo del motor; conduce a una desconexión
2.0	in	Actual_Speed	INT	0	0	revoluciones reales del motor
4.0	out	Engine_On	BOOL	FALSE	FALSE	Conexión del motor
4.1	out	Preset_Speed_Reached	BOOL	FALSE	FALSE	Número de revoluciones alcanzado
6.0	stat	Preset_Speed	INT	1500	1200	Número de revoluciones solicitado

Figura 10. Bloque de datos DB2

6 Programar la llamada del bloque en KOP

La programación realizada en un FB no sirve de nada si este FB no es llamado por el OB1. Cada vez que se llama al FB se utiliza un bloque de datos distinto, controlándose así ambos motores.



Navegue hasta la carpeta **Bloques** y abra el OB1. Inserte en el segmento 1 en la ventana KOP/AWL/FUP.

En el área de instrucciones escriba **CALL "Motor", "Gasolina"** y, a continuación, pulse la tecla **Entra**r. Así se mostrarán todos los parámetros del bloque de función "Gasolina".

Posicione el cursor detrás del signo '=' de 'Arrancar' y pulse la tecla derecha del ratón. Abra el menú contextual que aparece al pulsar la tecla derecha del ratón y elija **Insertar símbol**o. Entonces se abrirá una lista desplegable. La primera vez tardará un poco más en abrirse.

```
OB1 :  "Main Program Sweep (Cycle)"
Comentario:

Segm. 1: Conectar motor gasolina
Llamada del bloque de función FB 1 ("Motor") con los datos de un motor de
gasolina (bloque de datos "Gasolina" DB 1).

      CALL  "Motor" , "Gasolina"
       Switch_On            :=
       Switch_Off           :=  Arrancar_MD      E   1.4
       Failure              :=  Arrancar_MG      E   1.0
       Actual_Speed         :=  Automático ON    E   0.5
       Engine_On            :=  Conectar_ventilad... A  5.6
       Preset_Speed_Reached:=   Conectar_ventilad... A  5.2
                                Datos_globales      DB  3
                                Debe_revol_alcan... A  5.5
                                Debe_revol_alcan... A  5.1
```

Finalmente la llamada a la función con los datos del motor de gasolina tendrá el siguiente aspecto:

```
Segm. 1: Conectar motor gasolina
Llamada del bloque de función FB 1 ("Motor") con los datos de un motor de
gasolina (bloque de datos "Gasolina" DB 1).

      CALL  "Motor" , "Gasolina"
       Switch_On            :="Arrancar_MG"
       Switch_Off           :="Parar_MG"
       Failure              :="Fallo_MG"
       Actual_Speed         :="Revol_reales_MG"
       Engine_On            :="MG_ON"
       Preset_Speed_Reached:="Debe_revol_alcanzado_MG"
```

En otro segmento programe la llamadada el bloque de función "Motor" (FB1) al bloque de datos "Diesel" (DB2).Proceda como hemos hecho anteriormente.

```
Segm. 2: Conectar motor diesel
Llamada del bloque de función FB 1 ("Motor") con los datos de un motor diesel
(bloques de datos "diesel" DB 2).

      CALL  "Motor" , "Diesel"
       Switch_On            :="Arrancar_MD"
       Switch_Off           :="Parar_MD"
       Failure              :="Fallo_DM"
       Actual_Speed         :="Revol_reales_MD"
       Engine_On            :="MD_ON"
       Preset_Speed_Reached:="Debe_revol_alcanzado_MD"
```

7 Probar el programa con la tabla de variables

El test de las variables del programa consiste en observar y forzar dichas variables. Para poder realizar este test tiene que existir una conexión online con la CPU, ésta tiene que estar en RUN-P y el programa se tiene que haber cargado ya.

Del mismo modo que en el test observando el estado del programa, al probar las variables puede observar las entradas y salidas en la tabla de variables. Además, ajustando un número REAL de revoluciones puede probar el comparador previsto para el número de revoluciones del motor en el FB1.

Inserte un **Tabla de variables** con ayuda del menú contextual que aparece al pulsar la tecla derecha del ratón (De la misma forma que se insertó el bloque de función).

Acepte los ajustes estándar cerrando el cuadro de diálogo "Propiedades" con **Aceptar**. Se creará una VAT1 (tabla de variables) en la carpeta 'Bloques'. Abra la **VAT1** (con un doble clic) y se abrirá la ventana "Observar y forzar variable".

Para evitar tener que crear un nueva tabla de variables se copiará la tabla de variables de un proyecto ejemplo. Para ello, abrir el proyecto ejemplo: ZEs01_01_STEP7__AWL_1-9 y copiar su tabla de variables.

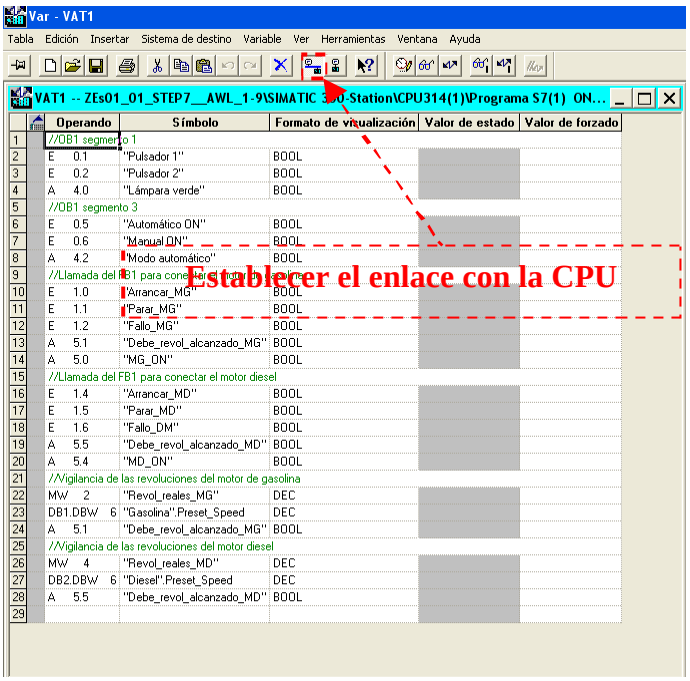
VAT1 -- ZEs01_01_STEP7__AWL_1-9\SIMATIC 300-Station\CPU314(1)\Programa S7(1)

	Operando	Símbolo	Formato de visualización	Valor de estado	Valor de forzado
1	//OB1 segmento 1				
2	E 0.1	"Pulsador 1"	BOOL		
3	E 0.2	"Pulsador 2"	BOOL		
4	A 4.0	"Lámpara verde"	BOOL		
5	//OB1 segmento 3				
6	E 0.5	"Automático ON"	BOOL		
7	E 0.6	"Manual ON"	BOOL		
8	A 4.2	"Modo automático"	BOOL		
9	//Llamada del FB1 para conectar el motor de gasolina				
10	E 1.0	"Arrancar_MG"	BOOL		
11	E 1.1	"Parar_MG"	BOOL		
12	E 1.2	"Fallo_MG"	BOOL		
13	A 5.1	"Debe_revol_alcanzado_MG"	BOOL		
14	A 5.0	"MG_ON"	BOOL		
15	//Llamada del FB1 para conectar el motor diesel				
16	E 1.4	"Arrancar_MD"	BOOL		
17	E 1.5	"Parar_MD"	BOOL		
18	E 1.6	"Fallo_DM"	BOOL		
19	A 5.5	"Debe_revol_alcanzado_MD"	BOOL		
20	A 5.4	"MD_ON"	BOOL		
21	//Vigilancia de las revoluciones del motor de gasolina				
22	MW 2	"Revol_reales_MG"	DEC		
23	DB1.DBW 6	"Gasolina".Preset_Speed	DEC		
24	A 5.1	"Debe_revol_alcanzado_MG"	BOOL		
25	//Vigilancia de las revoluciones del motor diesel				
26	MW 4	"Revol_reales_MD"	DEC		
27	DB2.DBW 6	"Diesel".Preset_Speed	DEC		
28	A 5.5	"Debe_revol_alcanzado_MD"	BOOL		
29					

Figura 11. Tabla de Variables VAT1.

Conectar la tabla de variables online

Haga clic en la ventana "Observar y forzar variable" en **ON** para establecer la conexión con la CPU configurada. En la barra de estado aparecerá "Online". Posicione el selector de modo de la CPU en **RUN-P** (si aún no lo está).



Observar variables

Haga clic en **Observar variable** . En la barra de estado se visualizará el funcionamiento de la CPU. Oprima los pulsadores 1 y 2 y observe el resultado en la tabla de variables. Verá que los valores de estado de la tabla cambian de 'false' a 'true'.

Forzar variables

En la columna 'Valor de forzado' introduzca el valor "1500" para el operando MW2 y "1300" para el operando MW4.

Transfiera los valores de estado a la CPU .

Var - @VAT1

Tabla Edición Insertar Sistema de destino Variable Ver Herramientas Ventana Ayuda

@VAT1 -- ZEs01_01_STEP7__AWL_1-9\SIMATIC 300-Station\CPU314(1)\Programa S7(1) O...

	Operando	Símbolo	Formato de visualización	Valor de estado	Valor de forzado
1	//OB1 segmento 1				
2	E 0.1	"Pulsador 1"	BOOL	false	
3	E 0.2	"Pulsador 2"	BOOL	false	
4	A 4.0	"Lámpara verde"	BOOL	false	
5	//OB1 segmento 3				
6	E 0.5	"Automático ON"	BOOL	true	
7	E 0.6	"Manual ON"	BOOL	false	
8	A 4.2	"Modo automático"	BOOL	false	
9	//Llamada del FB1 para conectar el motor de gasolina				
10	E 1.0	"Arrancar_MG"	BOOL	true	
11	E 1.1	"Parar_MG"	BOOL	false	
12	E 1.2	"Fallo_MG"	BOOL	true	
13	A 5.1	"Debe_revol_alcanzado_MG"	BOOL	false	
14	A 5.0	"MG_ON"	BOOL	true	
15	//Llamada del FB1 para conectar el motor diesel				
16	E 1.4	"Arrancar_MD"	BOOL	false	
17	E 1.5	"Parar_MD"	BOOL	false	
18	E 1.6	"Fallo_DM"	BOOL	false	
19	A 5.5	"Debe_revol_alcanzado_MD"	BOOL	false	
20	A 5.4	"MD_ON"	BOOL	false	
21	//Vigilancia de las revoluciones del motor de gasolina				
22	MW 2	"Revol_reales_MG"	DEC	0	
23	DB1.DBW 6	"Gasolina".Preset_Speed	DEC	1500	
24	A 5.1	"Debe_revol_alcanzado_MG"	BOOL	false	
25	//Vigilancia de las revoluciones del motor diesel				
26	MW 4	"Revol_reales_MD"	DEC	0	
27	DB2.DBW 6	"Diesel".Preset_Speed	DEC	1200	
28	A 5.5	"Debe_revol_alcanzado_MD"	BOOL	false	
29					

Figura 12. Tabla de Variables VAT1.Observar Variables.

Var - @VAT1

Tabla Edición Insertar Sistema de destino Variable Ver Herramientas Ventana Ayuda

@VAT1 -- ZEs01_01_STEP7__AWL_1-9\SIMATIC 300-Station\CPU314(1)\Programa S7(1) O...

	Operando	Símbolo	Formato de visualización	Valor de estado	Valor de forzado
1	//OB1 segmento 1				
2	E 0.1	"Pulsador 1"	BOOL	false	
3	E 0.2	"Pulsador 2"	BOOL	false	
4	A 4.0	"Lámpara verde"	BOOL	false	
5	//OB1 segmento 3				
6	E 0.5	"Automático ON"	BOOL	true	
7	E 0.6	"Manual ON"	BOOL	false	
8	A 4.2	"Modo automático"	BOOL	false	
9	//Llamada del FB1 para conectar el motor de gasolina				
10	E 1.0	"Arrancar_MG"	BOOL	true	
11	E 1.1	"Parar_MG"	BOOL	false	
12	E 1.2	"Fallo_MG"	BOOL	true	
13	A 5.1	"Debe_revol_alcanzado_MG"	BOOL	true	
14	A 5.0	"MG_ON"	BOOL	true	
15	//Llamada del FB1 para conectar el motor diesel				
16	E 1.4	"Arrancar_MD"	BOOL	false	
17	E 1.5	"Parar_MD"	BOOL	false	
18	E 1.6	"Fallo_DM"	BOOL	false	
19	A 5.5	"Debe_revol_alcanzado_MD"	BOOL	true	
20	A 5.4	"MD_ON"	BOOL	false	
21	//Vigilancia de las revoluciones del motor de gasolina				
22	MW 2	"Revol_reales_MG"	DEC	1500	1500
23	DB1.DBW 6	"Gasolina".Preset_Speed	DEC	1500	
24	A 5.1	"Debe_revol_alcanzado_MG"	BOOL	true	
25	//Vigilancia de las revoluciones del motor diesel				
26	MW 4	"Revol_reales_MD"	DEC	1300	1300
27	DB2.DBW 6	"Diesel".Preset_Speed	DEC	1200	
28	A 5.5	"Debe_revol_alcanzado_MD"	BOOL	true	
29					

Figura 13. Tabla de Variables VAT1. Forzar Variables.

8 Programar una función (FC)

Crear y abrir la función

La función, como el bloque de función, está subordinada al bloque de organización. Para que pueda ser procesada por la CPU, ésta tiene que ser llamada también desde el bloque de orden superior. Para ello, al contrario que en el caso del bloque de función, no se requiere ningún bloque de datos.

En las funciones, los parámetros también se listan en la tabla de declaración de variables, pero no se admiten datos locales estáticos. Las funciones, como los bloques de función, se programan en la ventana KOP/FUP/AWL.

Inserte un **Función** (FC1) con ayuda del menú contextual que aparece al pulsar la tecla derecha del ratón (De la misma forma que se insertó el bloque de función).

Programar la función

A continuación se programará una función de temporización en nuestro ejemplo. Esta función de temporización activará un retardo a la desconexión, p.ej., para que el ventilador que funciona automáticamente al arrancar el motor siga funcionando unos cuatro segundos más tras desconectarse el motor.

Como hemos mencionado anteriormente, los parámetros de entrada y salida de la función (declaración "in" y "out") deben indicarse en la tabla de declaración de variables.

Haga la siguiente tabla de declaración de variables:

Dirección	Declaración	Nombre	Tipo	Valor inicial	Comentario
0.0	in	Engine_On	BOOL		Señal para la conexión del motor
2.0	in	Timer_Function	TIMER		Función de temporización utilizada para el retardo a la desconexión
4.0	out	Fan_On	BOOL		Señal para la conexión del ventilador
	in_out				
	temp				

Programar la función de temporización en AWL

Si programa en AWL, seleccione el campo de entrada que se encuentra debajo del segmento e introduzca la instrucción que le mostramos a continuación:

```
U     #Engine_On
L     S5T#4S
SA    #Timer_Function

U     #Timer_Function
=     #Fan_On
```

Llamada de la función en el OB1

La llamada de la función FC1 en el OB1 se realiza del mismo modo que la llamada del bloque de función. Todos los parámetros de la función reciben en el OB1 los valores de los correspondientes operandos del motor de gasolina o del motor diesel.

Segm. 3 : Control del ventilador para motor de gasolina

```
Llamada de la función FC 1 para el motor de gasolina.

      CALL  "Ventilador"
       Engine_On      :="MG_ON"
       Timer_Function:="Retardo_descon_MG"
       Fan_On         :="Conectar_ventilador_MG"
```

Segm. 4 : Control del ventilador para motor diesel

```
Llamada de la función FC 1 para el motor diesel.

      CALL  "Ventilador"
       Engine_On      :="MD_ON"
       Timer_Function:="Retardo_descon_MD"
       Fan_On         :="Conectar_ventilador_MD"
```