

# Tema 6

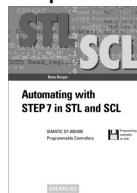


## Programación estructurada



## Bibliografía

- Título: "Step7 Avanzado"
  - Autor: José Martínez Torres
  - Descargar de la página web
- Manual Siemens "Step7-AWL para S7-300 y S7-400"
- Manual Siemens "Step7-KOP para S7-300 y S7-400"
- Manual Siemens "Step7-FUP para S7-300 y S7-400"
- Título: "Automating with Step7 in STL and SCL"
  - Autor: Hans Berger
  - ISBN: 3-89578-140-1



## Bibliografía

---



- Título: “Comunicaciones Industriales”
  - Autores: V.Sempere, J. Silvestre, J.A. Martínez
  - Editorial : SPUPV (SPUPV-2002.213)
  - Año:2002



## Índice

---



- Módulos de STEP 7
- Módulos de Organización (OBs)
- Funciones (FCs)
- Módulos de Función (FBs)
- Llamada a módulos
- Acceso y Restricciones de los Parámetros de Paso
- Uso de los Parámetros EN/ENO
- Ejercicio: Funciones con Parámetros y sin Parámetros
- Módulos de datos (DBs)



# Módulos de STEP 7

- **Módulos**
  - Los módulos son partes funcionales del programa de usuario. Se diferencian en su función, uso y estructura. Los módulos representan el código ejecutable del programa.
- El entorno STEP7 soporta los siguientes tipos de módulos:
  - Módulos lógicos:
    - OBs - Módulos de Organización
    - FCs – Funciones
    - FBs - Módulos de Función
    - SFCs - Funciones de Sistema
    - SFBs - Módulos de Función de Sistema
  - Módulos de Datos:
    - DBs - Módulos de Datos
    - SDBs - Módulos de Datos de Sistema
  - Tipos de Datos definidos por el Usuario:
    - UDTs

# Módulos de STEP 7

Nombre del objeto	Nonbr.	Protección KNOV/H...	Tipo	Lenguaje	Usuario	Tamaño en la memor...	Última modif. del inte...	Última...
☞ Datos de sistema	---	---	SDB	---	---	---	---	30.10...
☞ OB1	---	---	Bloque de organización	KOP		322	10.11.1998 09:25:34	11.12...
☞ OB82	---	---	Bloque de organización	AWL		62	10.11.1998 17:15:11	18.11...
☞ OB100	---	---	Bloque de organización	KOP		90	24.11.1998 14:43:06	09.12...
☞ FB1	---	---	Bloque de función	KOP		1116	11.12.1998 09:26:00	11.12...
☞ FB2	---	---	Bloque de función	KOP		1154	11.12.1998 09:02:02	11.12...
☞ FB3	---	---	Bloque de función	KOP		1330	09.12.1998 10:18:31	10.12...
☞ FB4	---	---	Bloque de función	KOP		1678	09.12.1998 17:37:04	09.09...
☞ FB5	---	---	Bloque de función	KOP		880	30.11.1998 12:12:00	06.09...
☞ FB6	---	---	Bloque de función	KOP		440	30.11.1998 15:24:20	26.11...
☞ FC0	SI	---	Función	AWL	AUT	296	26.01.1996 14:08:57	22.07...
☞ FC8	SI	---	Función	AWL	AUT	214	26.01.1996 14:13:01	26.01...
☞ FC10	SI	---	Función	AWL	AUT	260	26.01.1996 14:13:15	26.01...
☞ FC11	SI	---	Función	AWL	AUT	278	26.01.1996 14:13:23	22.07...
☞ FC12	SI	---	Función	AWL	AUT	322	26.01.1996 14:13:30	26.01...
☞ FC100	---	---	Función	KOP		230	18.11.1998 10:24:46	10.12...
☞ FC101	---	---	Función	KOP		386	19.11.1998 14:08:21	09.09...
☞ FC102	---	---	Función	KOP		288	23.11.1998 10:26:34	16.12...
☞ FC103	---	---	Función	KOP	Lois	120	18.12.1996 17:19:16	23.11...
☞ FC104	---	---	Función	KOP		360	23.11.1998 11:17:01	25.07...
☞ DB1	---	---	Bloque de datos	DB		166	10.11.1998 14:42:18	20.11...
☞ DB2	---	---	Bloque de datos	DB		166	10.11.1998 14:42:18	23.11...
☞ DB3	---	---	Bloque de datos	DB		38	09.12.1998 10:18:31	09.12...
☞ DB4	---	---	Bloque de datos	DB		52	09.12.1998 17:37:04	09.12...
☞ DB5	---	---	Bloque de datos	DB		44	30.11.1998 12:12:00	30.11...
☞ DB6	---	---	Bloque de datos	DB		38	30.11.1998 15:24:20	30.11...
☞ DB10	---	---	Bloque de datos	DB		40	11.12.1998 09:26:00	11.12...
☞ DB11	---	---	Bloque de datos	DB		40	11.12.1998 09:02:02	11.12...
☞ DB20	---	---	Bloque de datos	DB		286	01.01.1994 00:00:00	24.11...
☞ SFB0	SI	---	SFB	AWL	SIMATIC	---	09.08.1996 08:21:46	09.08...
☞ SFB1	SI	---	SFB	AWL	SIMATIC	---	09.08.1996 08:21:48	09.08...
☞ SFB2	SI	---	SFB	AWL	SIMATIC	---	15.08.1996 07:53:04	15.08...
☞ SFB3	SI	---	SFB	AWL	SIMATIC	---	09.08.1996 08:21:48	09.08...
☞ SFB4	SI	---	SFB	AWL	SIMATIC	---	09.08.1996 08:21:48	09.08...
☞ SFB5	SI	---	SFB	AWL	SIMATIC	---	09.08.1996 08:21:49	09.08...
☞ SFB32	SI	---	SFB	AWL	SIMATIC	---	24.09.1996 07:22:54	24.09...

# Módulos de Organización (OBs)

- **Módulos de Organización**
  - Constituyen el interface entre la CPU y el programa. El sistema operativo llama a estos módulos de código para que realicen distintas operaciones de acuerdo a su función.
- **OB1**
  - Es un módulo de código que trabaja cíclicamente, el OB1 se ejecuta en cada ciclo. Puede almacenar y ejecutar su programa completo en el OB1, o bien puede almacenar su programa en diferentes módulos lógicos y utilizar el OB1 para llamarlos cuando sea necesario. Además del OB1, el sistema operativo puede llamar a otros OBs en respuesta a ciertos eventos. Los OBs sólo pueden ser llamados por el sistema operativo, no por otros módulos lógicos.
  - El sistema operativo de la CPU del S7 ofrece 26 clases de prioridad (niveles de ejecución del programa). El sistema operativo procesa los OBs de acuerdo a su prioridad. El OB1 (prioridad clase 1) tiene la prioridad más baja de todos los OBs. Esto significa que todos los otros OBs pueden interrumpir la ejecución cíclica del OB1 si es necesario.
- Cada CPU del S7 soporta diferentes OBs y sus respectivas funciones. Los diferentes tipos de OB realizan distintas funciones:

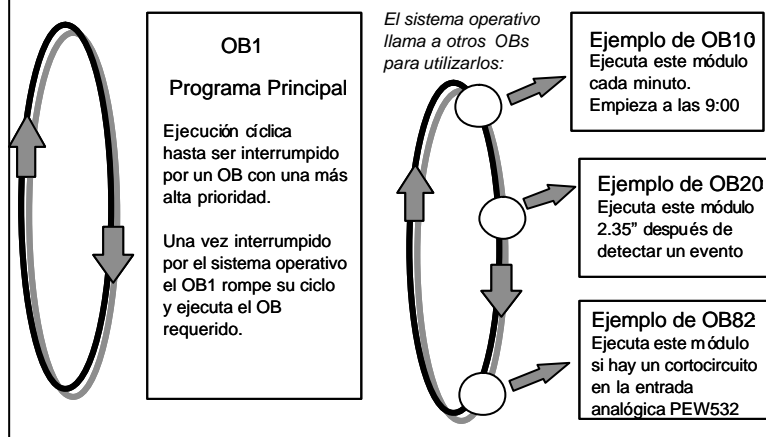
OB1	Programa principal	Ejecución en ciclos continuos
OB10	Interrupción Horaria	Empieza en un momento programado
OB20	Interrupción de Retardo	Corre después de una demora programada
OB35	Interrupción Cíclica	Se repite en intervalos programados
OB40	Interrupción Hardware	Corre al detectarse un error en un módulo
OB80 - 87	Error Asíncrono	Se ejecuta si hay errores asíncronos
OB121,122	Error Síncrono	Se ejecuta si hay errores síncronos
OB100	Rearranque Completo	Se ejecuta al pasar la CPU de STOP a RUN
OB101	Rearranque	Se ejecuta si hay un rearmar que de la CPU

Programación estructurada

7

# Módulos de Organización (OBs)

## Módulos de Organización (OBs)



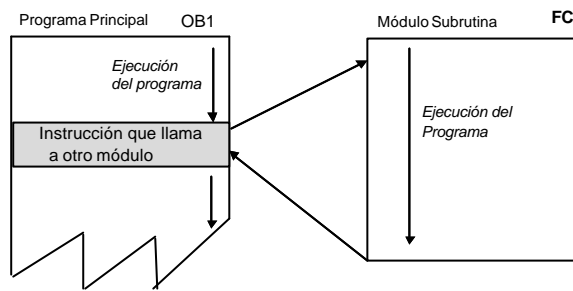
Programación estructurada

8

# Funciones (FCs)

## Funciones (FCs)

□ Las Funciones son módulos de código que se ejecutan al ser llamadas desde otro módulo. Cuando se la llama, la función actúa como una subrutina del programa. Cuando termina de ejecutar su código, el control se devuelve al módulo y al segmento desde el cual fue llamada.



Programación estructurada

9

# Funciones (FCs)

## • FC sin parámetros:

- Las funciones (FCs) actúan como subrutinas del módulo de código que las llama. Si no se pasa ningún valor (parámetro) desde el módulo que realiza la llamada, la FC ejecuta su código con sus propios valores.

```
Segmento 1 :  
L MD10  
L MD20  
+R  
L MD40  
/R  
RND+  
T MW0
```

Programación estructurada

10

## Funciones (FCs)

- **FC con parámetros:**

- En las Funciones (FC), los valores (parámetros) pasados desde otro módulo, se pueden usar con el nombre de variable local. Los valores se almacenan en la pila L, que está dinámicamente asignada en la memoria local. Una vez que acaba la ejecución de la FC, los valores almacenados en la pila L se pierden.

0.0	valor_A	real	temperatura1
4.0	valor_B	real	temperatura2
8.0	valor_C	real	temperatura3

Segmento 1 :			
L valor_A			
L valor_B			
+R			
L valor_C			
/R			
RND+			
T MW0			

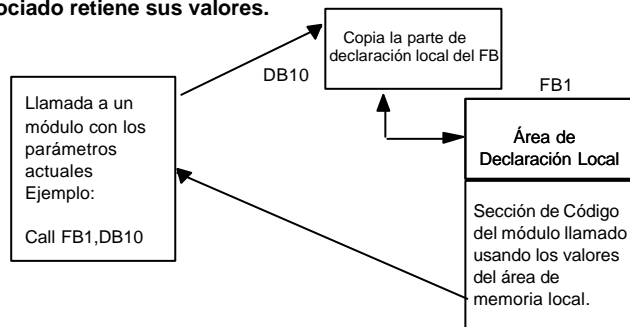
Programación estructurada

11

## Módulos de Función (FBs)

### Módulos de Función (FBs)

□ Un Módulo de Función (FB) tiene un módulo adicional de memoria asociado a él. Un Módulo de Datos (DB) mantiene una copia de los parámetros que van a pasar al área de datos locales. Después de la ejecución del FB, se borra el área de memoria local, pero el DB asociado retiene sus valores.

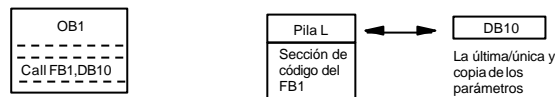


Programación estructurada

12

## Módulos de Función (FBs)

- **FB con Parámetros (Con DB a Instancia)**
  - Un Módulo de Función es un módulo lógico con un área de memoria asociada en forma de módulo de datos “a instancia”.
  - Los parámetros pasados al área de memoria local del FB se almacenan también en el DB a instancia. Los datos almacenados en el DB se mantienen aunque el FB termine.
  - El DB que está asociado a un FB debe tener la misma estructura de datos que el área de declaración local.
  - Si utiliza el mismo DB cada vez que llama al FB, sólo retendrá el último (el más reciente) conjunto de datos que pasó al FB.

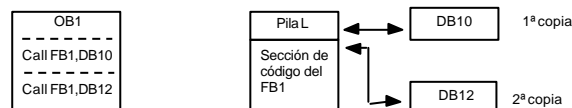


Programación estructurada

13

## Módulos de Función (FBs)

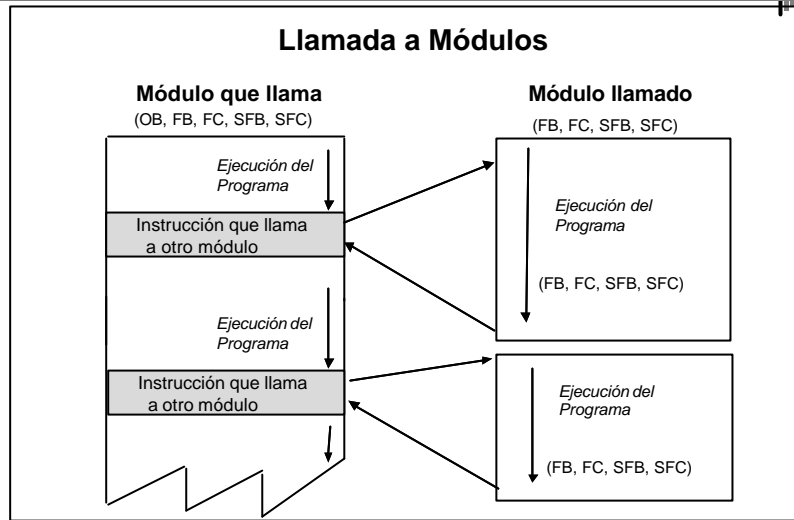
- Si quiere mantener muestras separadas de los datos, puede asociar un DB diferente cada vez que llame a un FB.



Programación estructurada

14

## Llamada a módulos



15

## Llamada a módulos

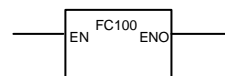
- La instrucción Call consigue que los módulos lógicos estructurados “se despierten”.
  - Como se ve en el diagrama anterior, cuando el programa llama al segundo módulo, éste ejecuta todas sus instrucciones.
  - Una vez que el módulo llamado ha terminado, el que hizo la llamada reasume la ejecución del programa continuando por la línea siguiente a la instrucción Call.

- **Llamada a una Función en AWL: Call FC100**

(sin parámetros)

- **Llamada a una Función en FUP/KOP:**

(sin parámetros)



Programación estructurada

16



## Llamada a módulos

- Llamada a una Función con parámetros en AWL:

Call FC20

Call FB30,DB10

Arranque:=MW20

Arranque:=MW20

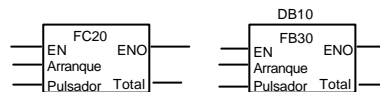
Pulsador:=PEW352

Pulsador:=PEW352

Total:=AW20

Total:=AW20

- Llamada a una Función con parámetros en FUP/KOP:

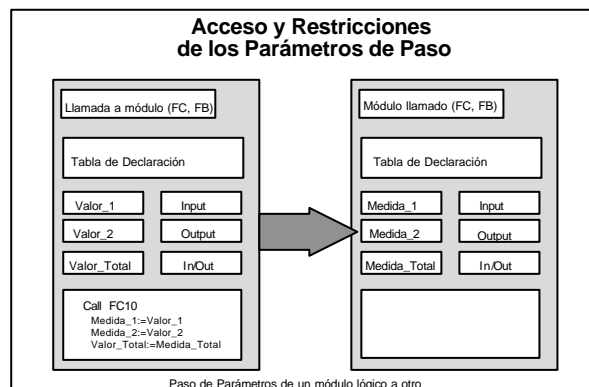


Programación estructurada

17

## Acceso y Restricciones de los Parámetros de Paso

- Acceso y restricciones de los Parámetros de Paso



Paso de Parámetros de un módulo lógico a otro

Programación estructurada

18

## Acceso y Restricciones de los Parámetros de Paso



- **Dirección Local**

- La dirección local identifica el área de memoria asignada por STEP7 a cada variable. Esas direcciones aparecen en pantalla cuando se termina de introducir una declaración y sale de la fila.
- La dirección es el offset (en formato BYTE.BIT) o el número de byte de datos en la memoria local. Es posible acceder directamente a una dirección local (Ej.: LW0), pero es más aconsejable escribir el programa usando los nombres simbólicos.

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
	temp				

Programación estructurada

19



## Acceso y Restricciones de los Parámetros de Paso



- **Tipo de Declaración** El tipo indica cómo se va utilizar la variable dentro del módulo.

- Las opciones son:

in	Parámetros de entrada de los que se lee
out	Parámetros de salida en los que se puede escribir
in_out	Pueden ser leídos o escritos
stat	Variable local cuyo valor se almacena en un DB a instancia
temp	Variable temporal cuyo valor no se guarda cuando el termina

Programación estructurada

20



## Ejercicio

- Ejercicio
  - Programe una FC con parámetros que active la salida que usted indique cuando se pulse la entrada que usted indique.
  - Programe otra FC sin parámetros que active la salida A4.0 cuando pulse la entrada E0.0 y la desactive cuando pulse la entrada E0.1.
  - Llame a ambas funciones desde el OB1.

## Ejercicio: solución

**Ejercicio 1: Funciones con parámetros y sin parámetros**

Dirección	Declaración	Nombre	Tipo
0.0	in	Entrada	BOOL
2.0	out	Salida	BOOL

FC2 : Función con parámetros  
Segm. 1: Titulo:  
U #Entrada  
= #Salida