

Llamar a subrutina

Operaciones con subrutinas

Operandos Tipos de datos

n: Constante
Constante

[Acceder a la memoria de la CPU](#)

[Errores](#)

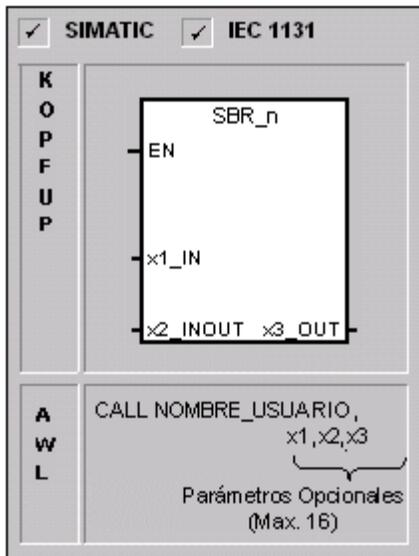
[Operaciones soportadas por las CPUs S7-200](#)

[Nemotécnica SIMATIC/Internacional](#)

[Rangos de datos](#)

[Tamaño de las operaciones en la memoria de la CPU](#)

[Direccionamiento](#)



La operación Llamar a subrutina (**CALL**) transfiere el control a la subrutina (SBR_n). Esta operación se puede utilizar con o sin parámetros. Una vez ejecutada la subrutina, el control vuelve a la operación que sigue a la llamada de la subrutina (CALL).

El límite máximo de parámetros de entrada/salida en cada llamada a subrutina es de 16. Si intenta cargar un programa que exceda este límite, ocurrirá un error.

Si ha asignado un nombre simbólico a la subrutina (por ejemplo, NOMBRE_SBR), éste aparecerá en la carpeta "Subrutinas" del árbol de operaciones.

Los valores de parámetros se asignan de la siguiente forma a la memoria Local en la subrutina:

1. Los valores de parámetros se asignan a la memoria Local en el orden indicado por la operación Llamar a subrutina, comenzando esos parámetros en L.0.
2. Uno a ocho valores binarios de parámetros consecutivos se asignan a un sólo byte comenzando en Lx.0 hasta Lx.7.
3. Los valores de byte, palabra y palabra doble se asignan a la memoria Local en bytes (LBx, LWx ó LDx).

En la operación CALL con parámetros, éstos deben concordar exactamente con las variables definidas en la tabla de variables locales de la subrutina. Los parámetros se deben organizar de la siguiente forma: primero los parámetros de entrada, luego los de entrada/salida y, por último, los de salida.

Condiciones de error que ponen ENO a 0 en una operación CALL con parámetros = 0:

0008 Excedida la profundidad máxima de anidado

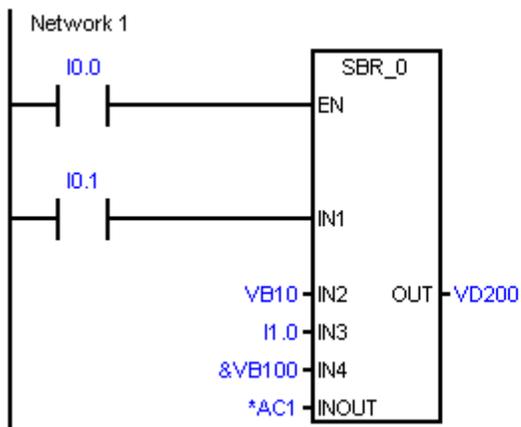
Nota: El rótulo informativo de la subrutina en el árbol de operaciones visualiza el nombre de cada parámetro.

Ejemplo de una llamada a subrutina

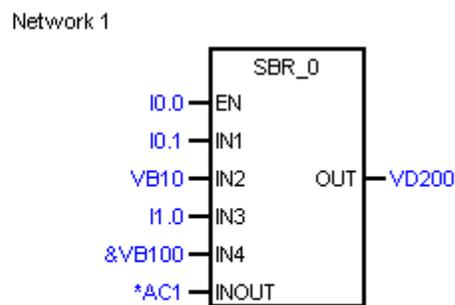
Tabla de variables locales de SBR_0

| | Símbolo | Tipo var. | Tipo de datos | Comentario |
|------|---------|-----------|---------------|------------|
| | EN | IN | BOOL | |
| L0.0 | IN1 | IN | BOOL | |
| LB1 | IN2 | IN | BYTE | |
| L2.0 | IN3 | IN | BOOL | |
| LD3 | IN4 | IN | DWORD | |
| | | IN | | |
| LD7 | INOUT | IN_OUT | REAL | |
| | | IN_OUT | | |
| LD11 | OUT | OUT | REAL | |
| | | OUT | | |

Programa principal en KOP



Programa principal en FUP



Código AWL creado por STEP 7-Micro/WIN conforme al diagrama KOP/FUP que permite visualizar el programa en KOP, FUP o AWL

Nota: STEP 7-Micro/WIN reserva y utiliza los 4 bytes más altos de la memoria L (LB60 a LB63) para cada uno de los parámetros de llamada.

NETWORK 1

// La memoria L se utiliza para guardar el estado de los parámetros de entrada BOOL
 // que se visualizan como entradas de circulación de corriente en KOP y FUP. Gracias a ello,
 // este segmento se puede visualizar en los editores KOP, FUP y AWL.

LD I0.0

= L60.0

```
LD I0.1
= L63.7
LD L60.0
CALL SBR_0 L63.7 VB10 I1.0 &VB100 *AC1 VD200
```

Ejemplo sólo en AWL (introducido con el editor AWL) para la misma llamada a subrutina que aparece arriba

Nota: Los programadores expertos en AWL pueden utilizar este método simplificado.

```
NETWORK 1
// Este segmento sólo se puede visualizar en el editor AWL,
// puesto que los parámetros BOOL se utilizan como entradas de circulación de corriente
// y no se guardan en la memoria L.
LD I0.0
CALL SBR_0 I0.1 VB10 I1.0 &VB100 *AC1 VD200
```

Consulte también:

[Subrutinas](#)