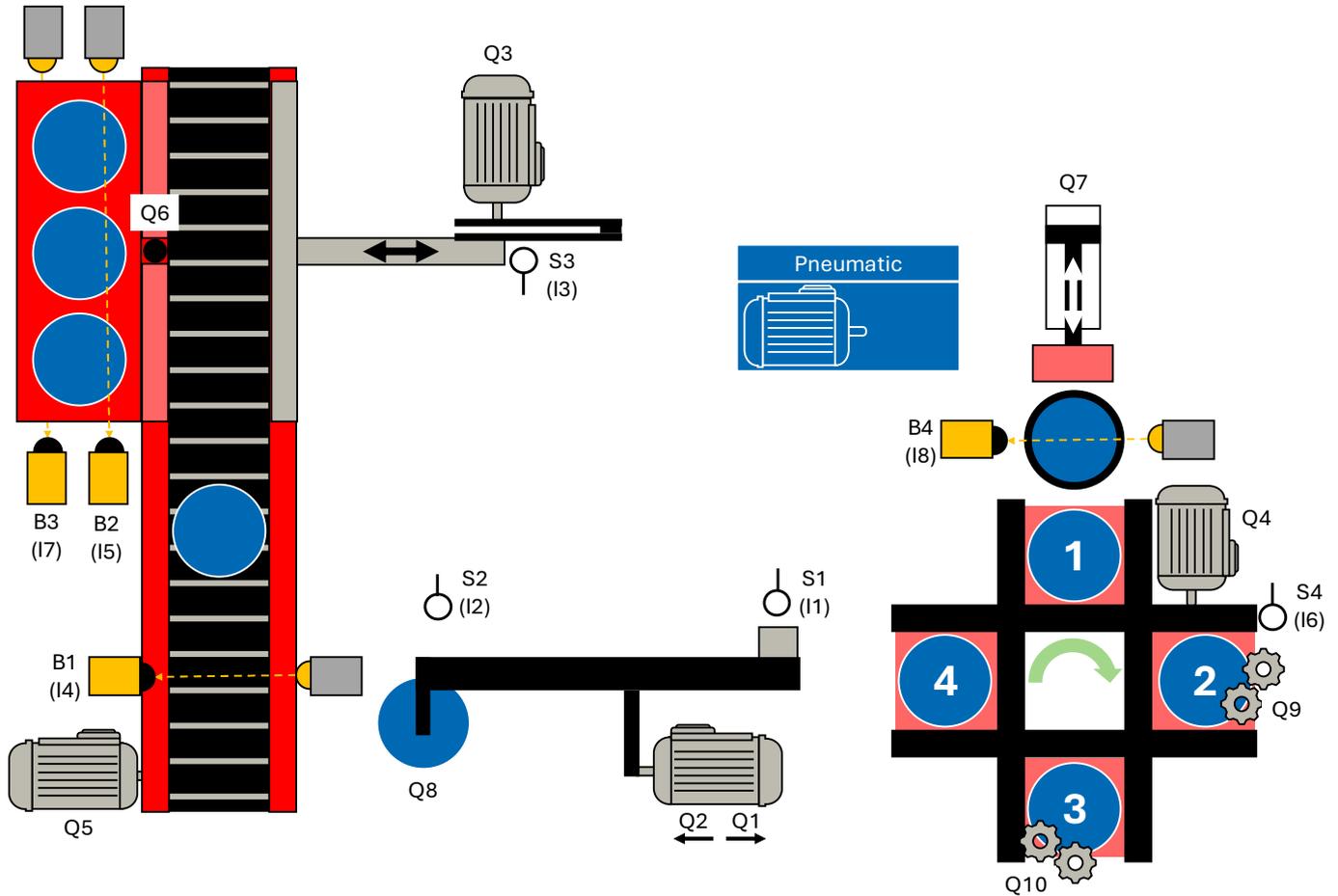


Línea de producción de 24 V

Programación estructurada



Índice

5	Programación estructurada	1
5.1	Introducción.....	1
5.2	Función.....	2
5.3	Bloque de funciones.....	3
5.4	Añadir un nuevo módulo	5
5.5	Llamada al módulo.....	6
5.6	Transferencia de parámetros.....	8
5.7	Llamada a una función (FC) en FBD.....	9
5.8	Llamada a un bloque de función (FB) en FBD.....	11
5.8.1	Llamada a procedimiento con instancia única.....	13
5.8.2	Opción de llamada como multiinstancia (portal TIA).....	16
5.8.3	Declaración textual como multiinstancia (CODESYS / Beckhoff)	16
5.9	Llamada a una función (FC) en ST / SCL.....	17
5.10	Llamada a un bloque de funciones (FB) en ST / SCL	19
5.10.1	Llamada a procedimiento con instancia única.....	20
5.10.2	Opción de llamada como multiinstancia (portal TIA)	23
5.10.3	Declaración textual como multiinstancia (CODESYS / Beckhoff).....	24

5 Programación estructurada

5.1 Introducción

La programación estructurada en sistemas PLC se utiliza para organizar programas complejos dividiéndolos en bloques más pequeños y claros. De este modo se mejora la legibilidad, el mantenimiento y la reutilización del código. El programa de usuario puede estructurarse en función de aspectos tecnológicos o funcionales.

En un programa PLC, los bloques como las funciones (FC) y los bloques de función (FB) se utilizan para estructurar partes del programa.

Los bloques de funciones deben comunicarse entre sí a través de sus interfaces de bloque de funciones en lugar de acceder directamente a las variables globales. Los parámetros se transfieren mediante entradas y salidas, así como parámetros InOut.

Para ejecutar los bloques de código del programa de control, hay que llamarlos.

5.2 Función

Las funciones (FC) son bloques de código sin memoria. **No disponen de memoria de datos** en la que puedan almacenarse los valores de los parámetros del bloque de funciones. Por este motivo, todos los parámetros de interfaz deben conectarse cuando se llama a una función. Para guardar datos de forma permanente, primero deben crearse bloques de datos globales.

Las funciones son ideales para tareas que no requieren memoria durante varios ciclos, como los cálculos matemáticos o las operaciones lógicas.

Una función contiene un programa que siempre se ejecuta cuando la función es llamada por otro bloque de código.

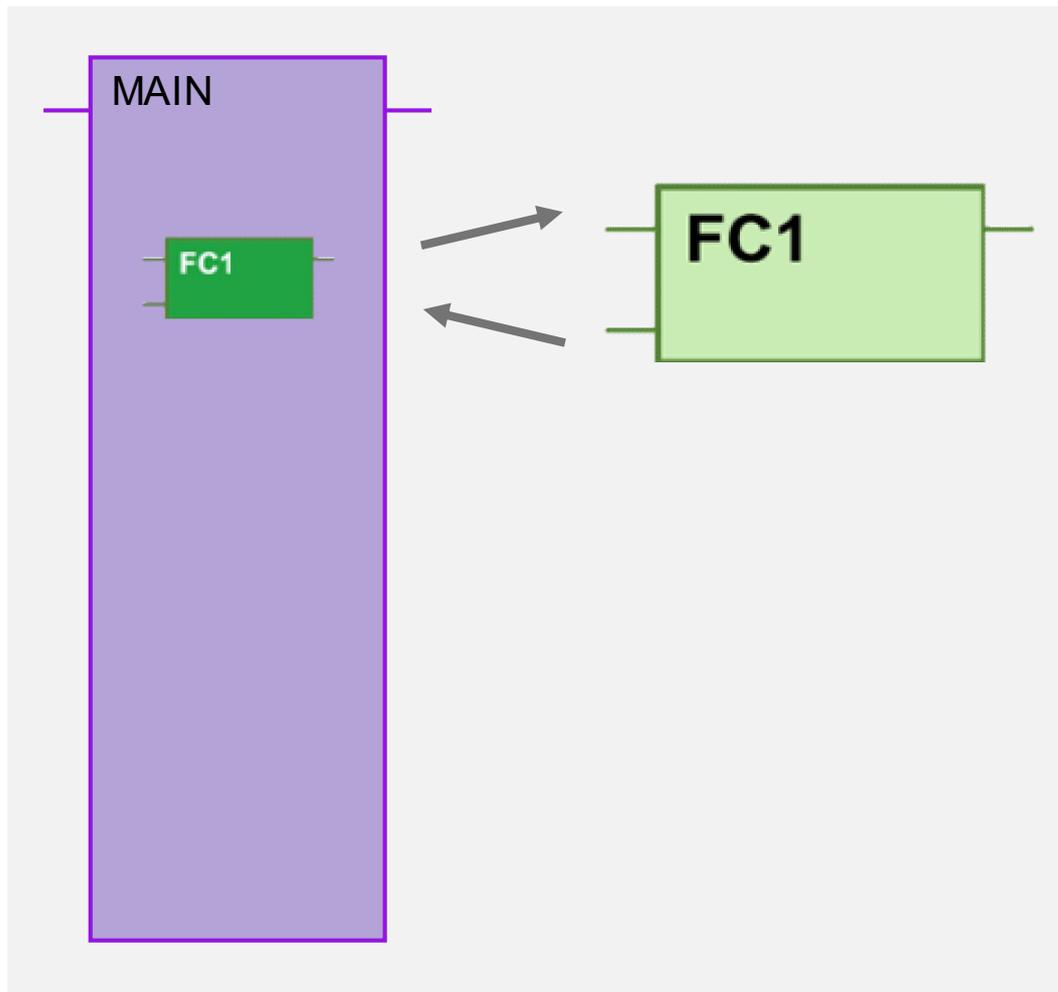


Imagen 1 Ejemplo: Llamada a una función desde MAIN

Una función también puede llamarse varias veces en distintos puntos de un programa.

5.3 Bloque de funciones

Los bloques de función (FB) son bloques de código que almacenan permanentemente sus variables de entrada, variables de salida, variables pass-through y también las variables estáticas en bloques de datos de instancia para que también estén disponibles tras el procesamiento del bloque. Por eso también se denominan bloques con memoria.

Los bloques de funciones se utilizan para tareas que no pueden realizarse con funciones:

- Siempre que se requieran tiempos y contadores en los bloques de construcción o
- si es necesario guardar información en el programa (por ejemplo, el estado de la cadena de pasos).

Los módulos de funciones siempre se ejecutan cuando un módulo de funciones es llamado por otro módulo de código.

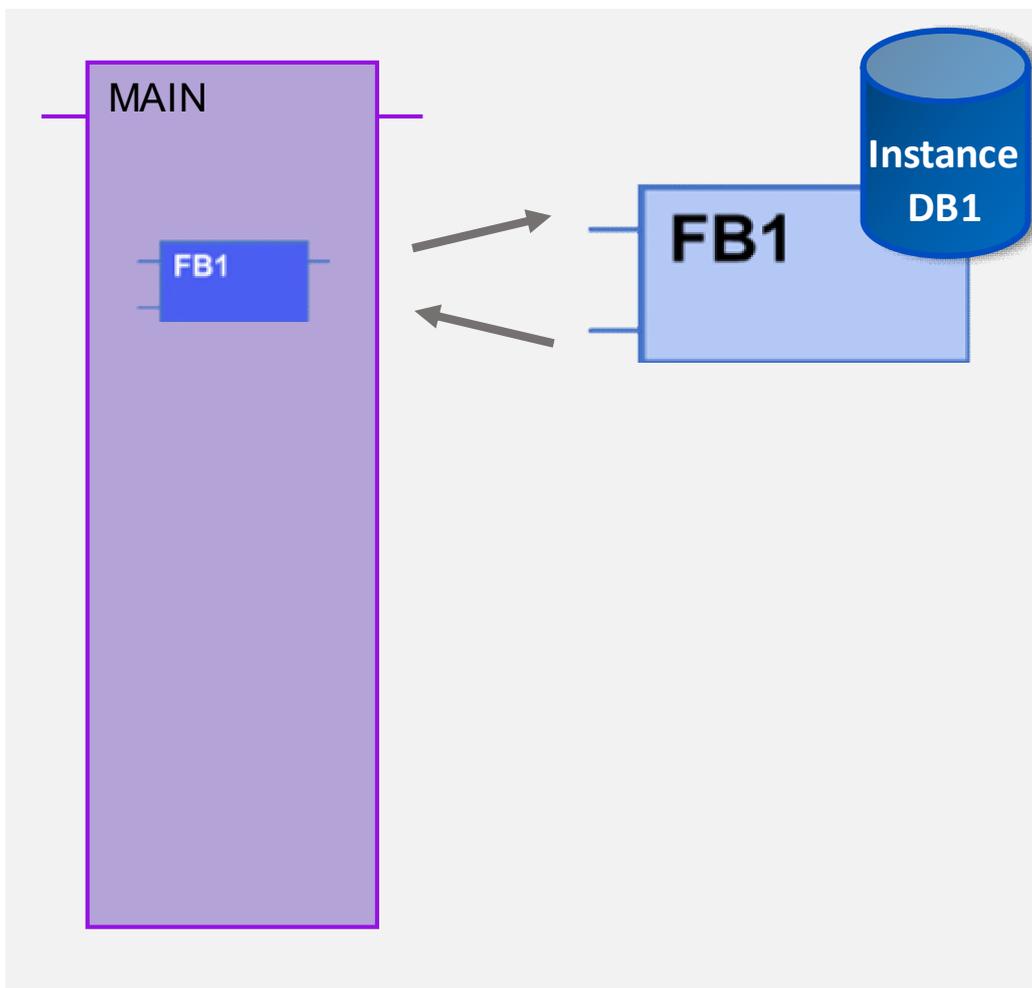


Imagen 2 Ejemplo: Llamada a un módulo de funciones desde MAIN

Un módulo de funciones también puede llamarse varias veces en distintos puntos de un programa.

Una llamada a un módulo de funciones se denomina instancia. A cada instancia de un módulo de funciones se le asigna un área de memoria que contiene los datos con los que trabaja el módulo de funciones.

5.4 Añadir nuevo módulo

En el TIA Portal, los bloques se gestionan en la navegación del proyecto, debajo del PLC, en la carpeta "Bloques de programa".

Haga doble clic en el comando "Añadir nuevo bloque" dentro de la carpeta "Bloques de programa" para abrir el cuadro de diálogo "Añadir nuevo bloque", con el que podrá crear un nuevo bloque.

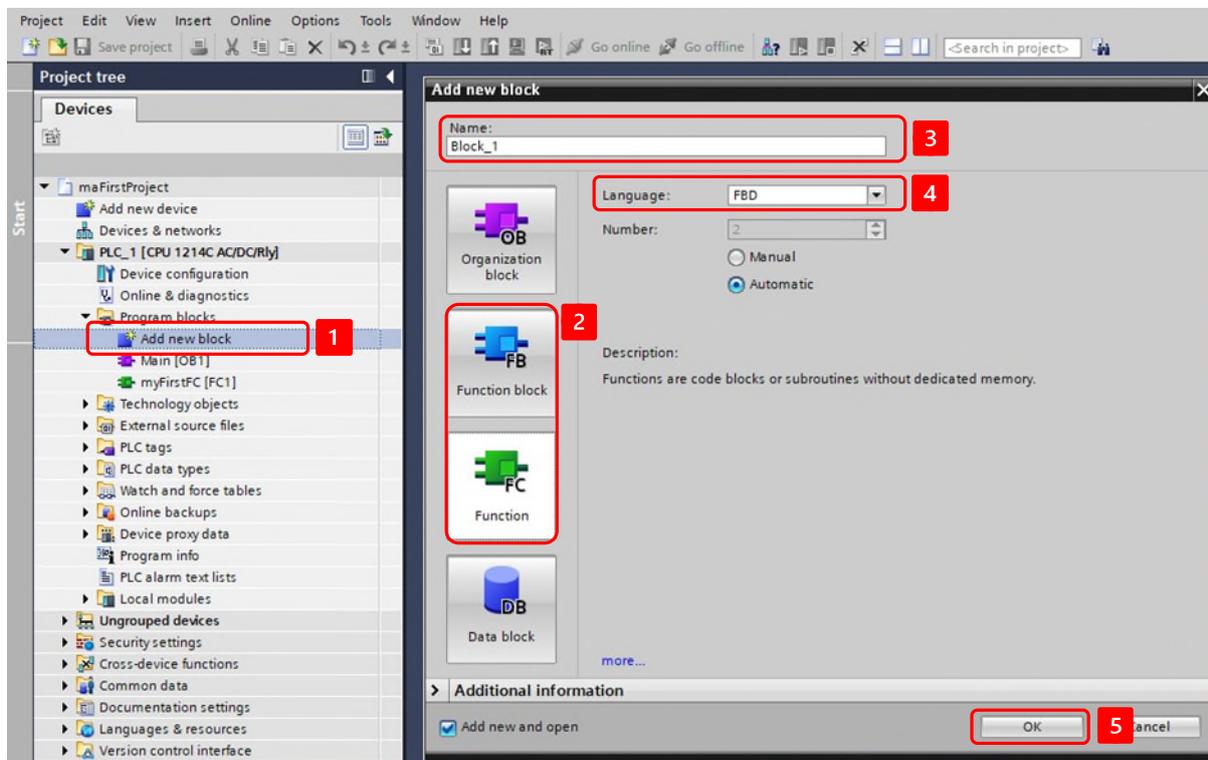


Imagen 3 Añadir nuevo módulo

Aquí debe seleccionarse el tipo de bloque (2), el nombre (3) y el lenguaje de programación deseado (4).

5.5 Llamada al módulo

Para ejecutar los bloques de código del programa de control, hay que llamarlos. El módulo de código responsable del procesamiento cíclico del programa suele denominarse "MAIN". Éste es iniciado por el sistema operativo y constituye la interfaz con el mismo. La CPU procesa el código de programa que se encuentra en "MAIN". Las partes del programa estructuradas en funciones y bloques de funciones pueden llamarse dentro de "MAIN".

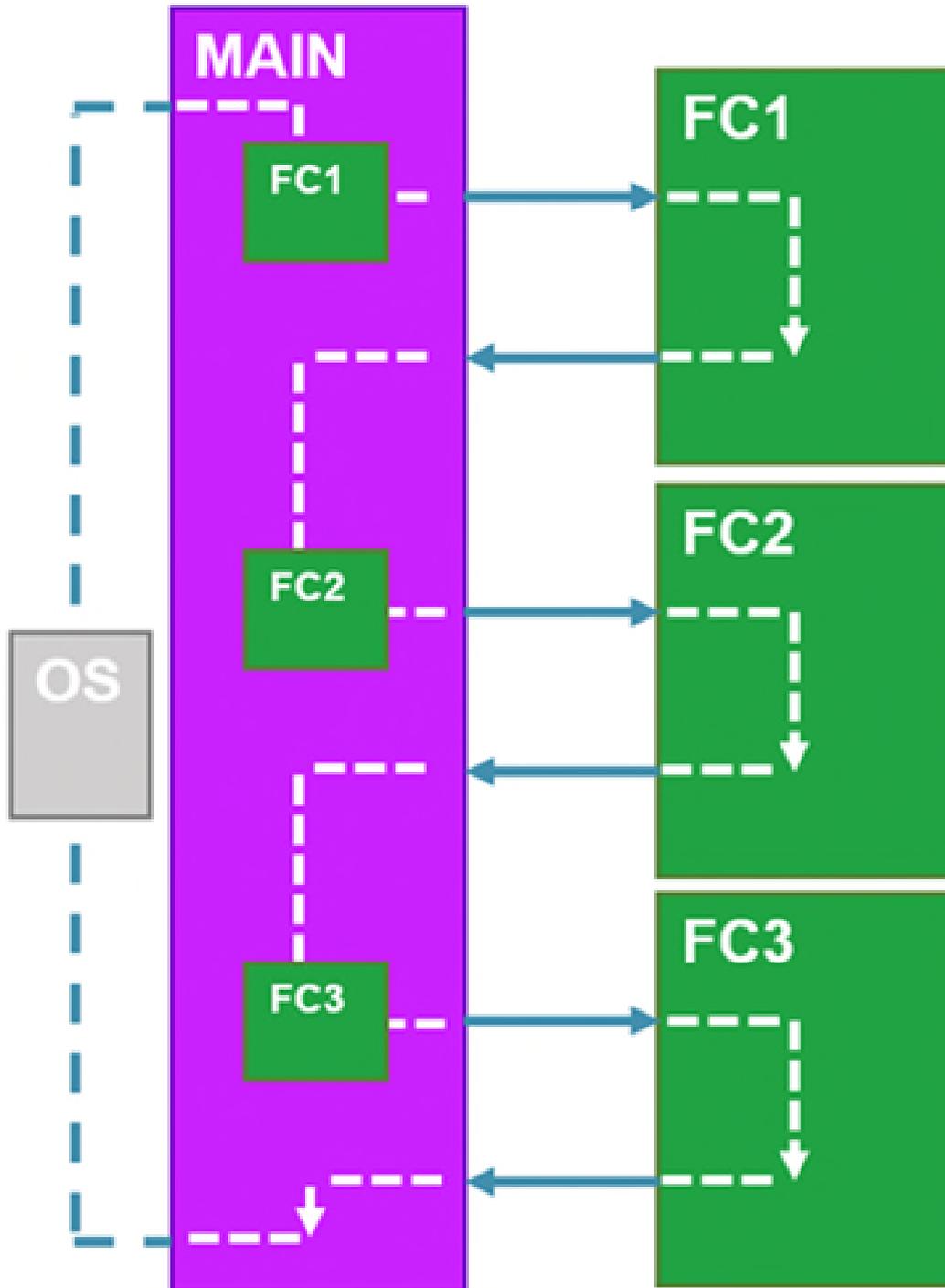


Imagen 4 Llamada a módulo en MAIN

Las funciones y los bloques de funciones estructuran el programa, facilitando su lectura y mantenimiento. Todos los bloques llamados se procesan uno tras otro.

El sistema operativo de la CPU vuelve a llamar al " PRINCIPAL " después del ciclo de programa, con lo que se ejecutan de nuevo todos los comandos programados en él.

Un bloque puede procesarse llamándolo desde la "PRINCIPAL", por ejemplo. Alternativamente, también puede llamarse desde un FB o FC, que a su vez se llaman en el "MAIN".

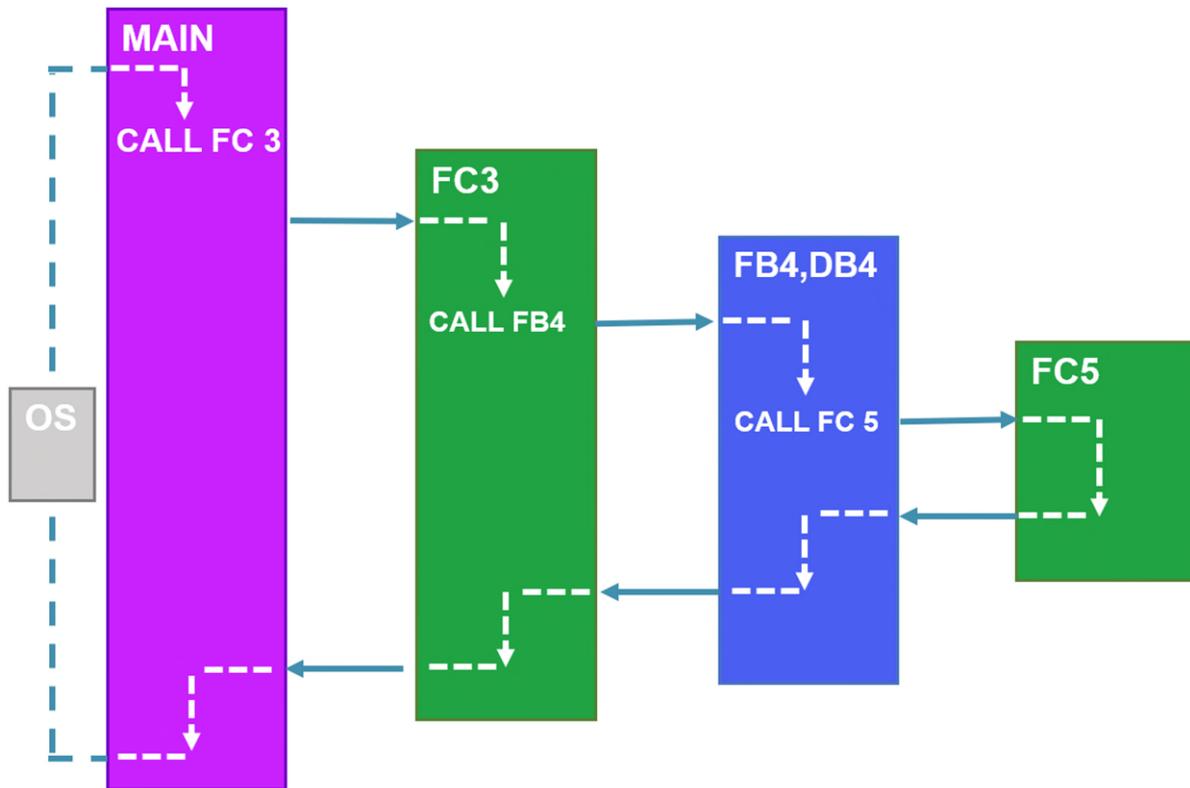


Imagen 5 Llamada al módulo

5.6 Transferencia de parámetros

Los parámetros (variables y valores de variables) pueden transferirse al llamar a funciones y módulos de funciones. El código del programa dentro del módulo de código trabaja entonces con los valores transferidos de los parámetros formales y puede devolver resultados mediante parámetros de salida o el valor de la función.

Los datos se intercambian a través de la interfaz del bloque de funciones. Se declaran en la interfaz de bloque de funciones de la función (FC) o del bloque de funciones (FB) y pueden utilizarse localmente en el bloque de funciones.

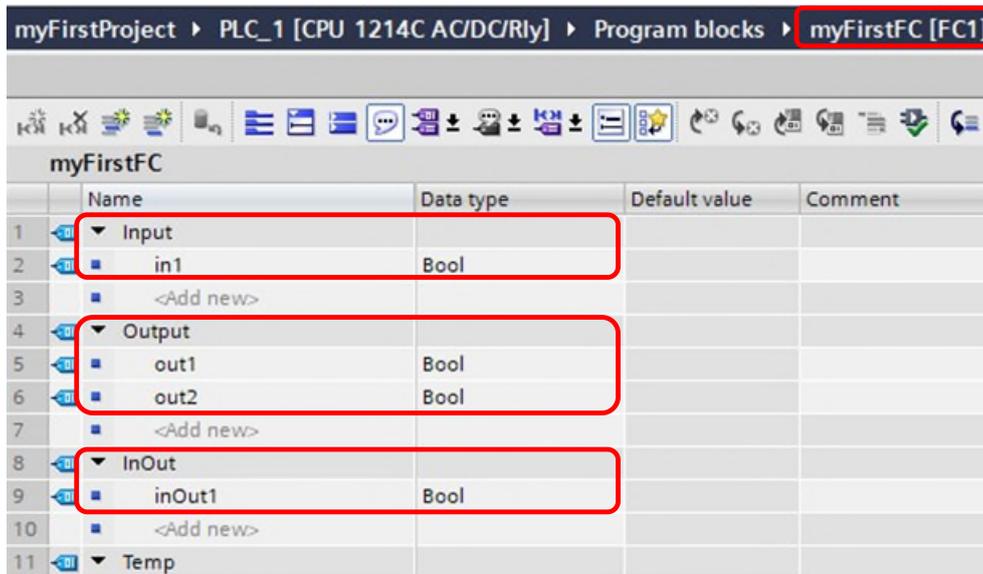


Imagen 6 Parámetros formales en la interfaz del bloque de funciones

Cuando se llama al bloque de funciones, estos parámetros formales se vinculan con parámetros reales (variables globales del PLC).

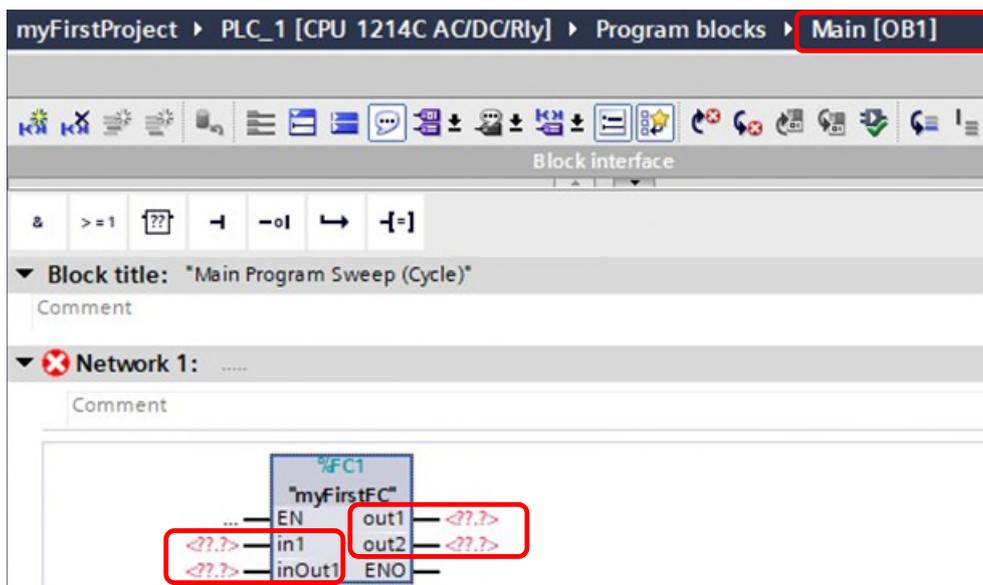


Imagen 7 Transferencia de los parámetros actuales al llamar

Transferencia de parámetros

Si el bloque de funciones llamado tiene parámetros de interfaz, éstos se muestran. En el caso de las funciones, los parámetros formales deben suministrarse con parámetros reales.

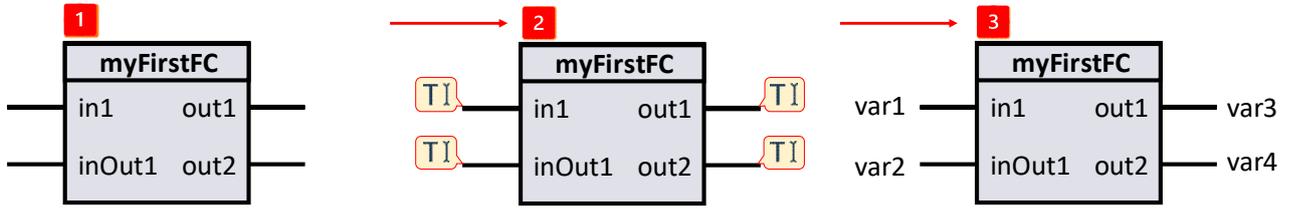


Imagen 10 Función con interfaz de bloque de funciones

Los parámetros de tipo "Entrada" e "InOut" se visualizan a la izquierda del bloque de funciones mediante patas de conexión. Los parámetros de tipo "Salida" deben conectarse a la derecha del bloque de funciones.

Ejemplo

Después de crear la función "myFirstFC", se la llamó en el "Main" (OB1). Los parámetros aún no se han transferido; los parámetros formales que debían vincularse se proporcionaron inicialmente con marcadores de posición "<??.?>".

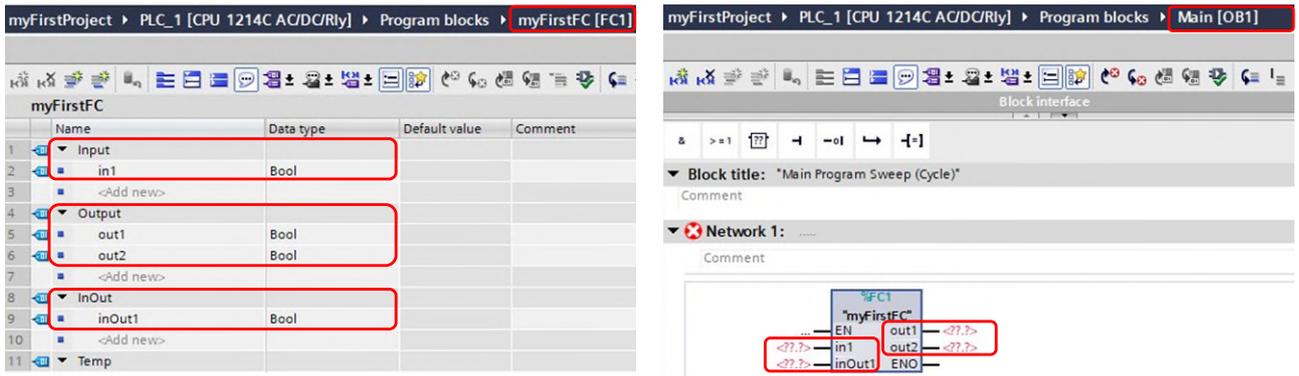


Imagen 11 Interfaz del módulo en el Portal TIA

5.8 Llamada a un bloque de función (FB) en FBD

Para que el bloque de funciones pueda procesarse, debe ser llamado por el programa. La llamada puede realizarse insertando una casilla vacía (atajo de teclado TIA "F8"). Una vez insertada la casilla vacía (1) y sustituidos los marcadores de posición de la casilla vacía por el nombre del módulo de funciones, sustituimos el marcador de posición situado encima de la casilla vacía por el nombre de la instancia (2).

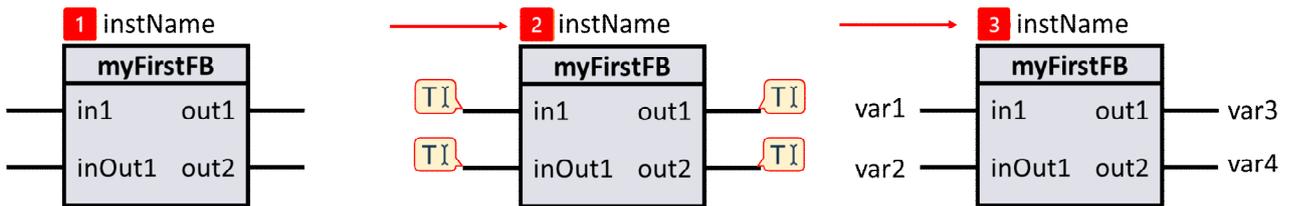


Imagen 12 Llamada a bloque de función desde caja vacía

Alternativamente, puede llamar el bloque de función mediante arrastrar y soltar, como se muestra en Llamar a una función.

Transferencia de parámetros

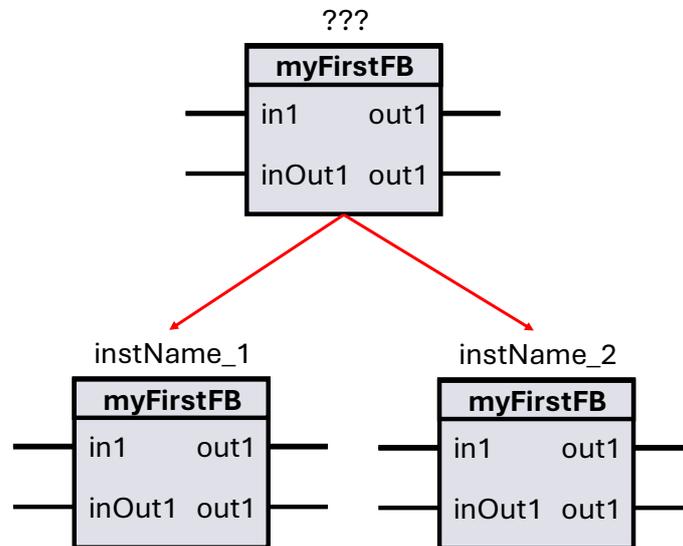
Si el bloque de funciones llamado tiene parámetros formales en la interfaz del bloque, éstos se muestran. No siempre es necesario pasar parámetros para los bloques de funciones, puesto que ya se asigna un área de memoria privada a la instancia.



Fotografía 13 Bloque de función con transferencia de parámetros

Instanciar el módulo de funciones varias veces

Si un bloque de función (FB) se llama dos veces (instanciado) en el programa de usuario, se crean dos instancias separadas. En ellas pueden guardar sus datos a lo largo del ciclo.



Fotografía 14 Instanciación de dos FB

Opciones de compra

Dependiendo del tipo de bloque de funciones de llamada, las instancias pueden ubicarse directamente en la interfaz del bloque de funciones (= multiinstancia en el Portal TIA) o almacenarse como instancias globales (= instancia única en el Portal TIA).

Ejemplo

Por ejemplo, si el FB contiene el programa de usuario para calcular operaciones de conmutación, cada llamada representa una instancia que sólo utiliza estados en el tiempo de ejecución de esta llamada. Se necesita una instancia distinta para cada llamada.

Esto significa que todos los datos (información) relativos a este cálculo están disponibles en esta instancia asignada.

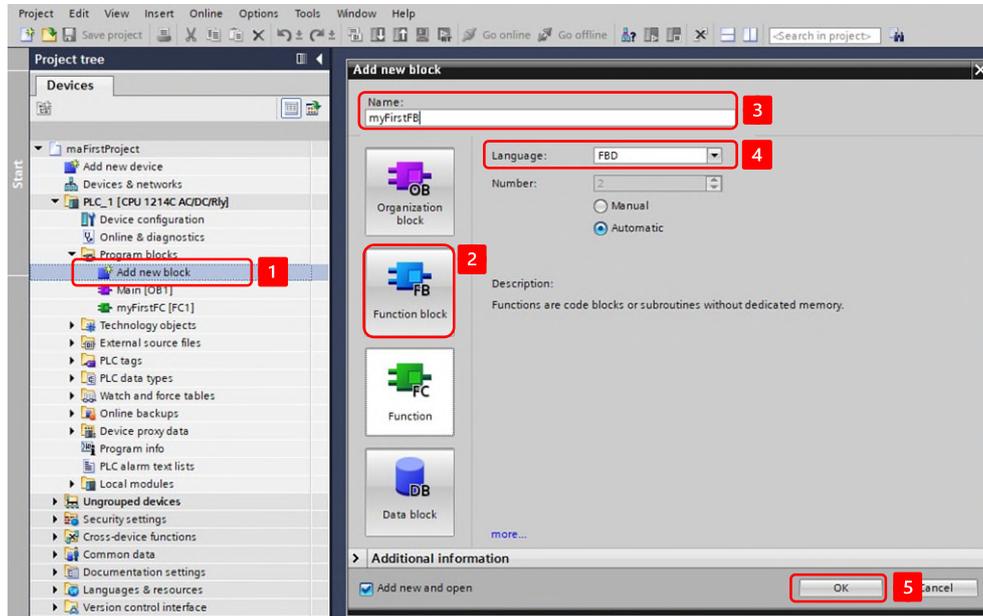
Para poder crear una instancia, el FB asignado ya debe existir.

En esta instancia se pueden observar las variables de la instancia correspondiente.

5.8.1 Llamada a procedimiento con instancia única

El procedimiento para llamar dos veces al bloque de funciones "myFirstFB" se muestra ahora paso a paso en el portal TIA:

1. Crear el módulo de función "miPrimerFB":



Fotografía 15 Añadir un nuevo módulo

2. Declaración de la interfaz del bloque de funciones y conexión de las variables en la parte de instrucciones del bloque de funciones:

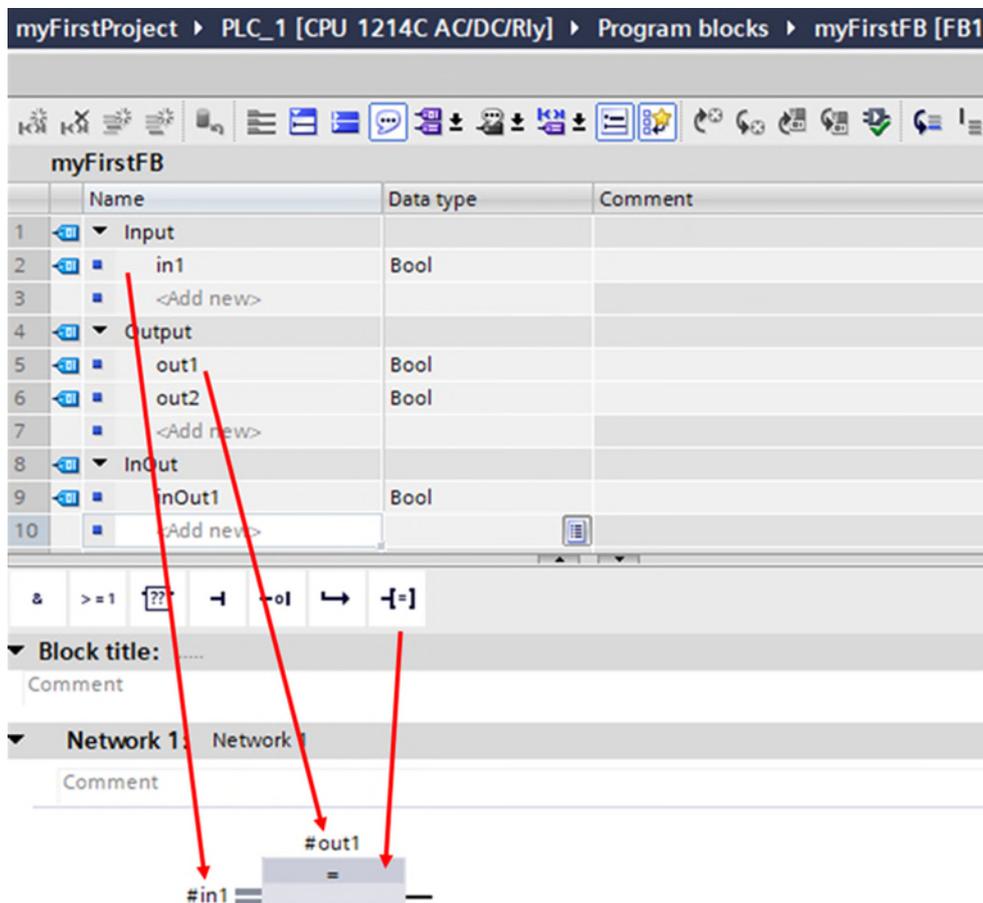
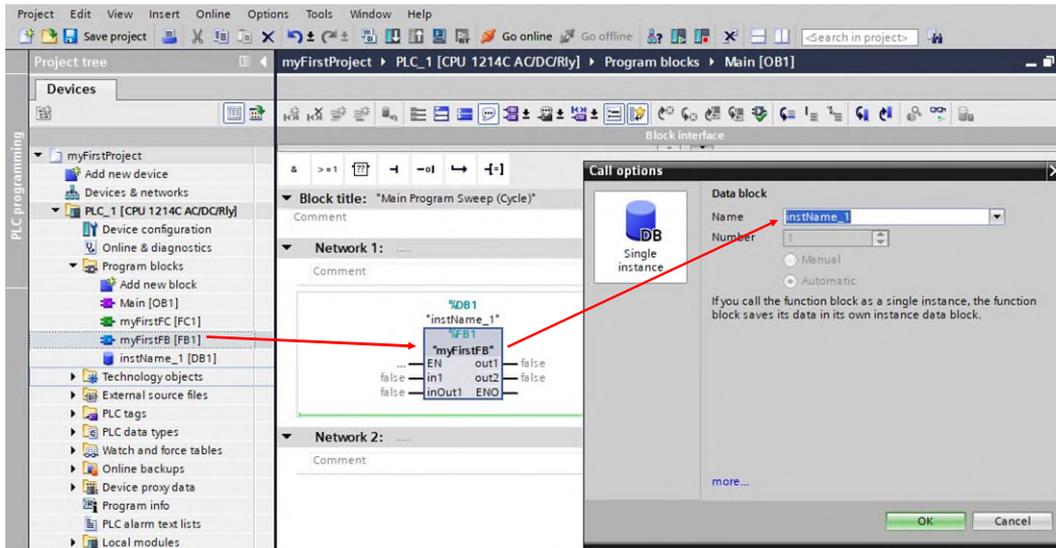


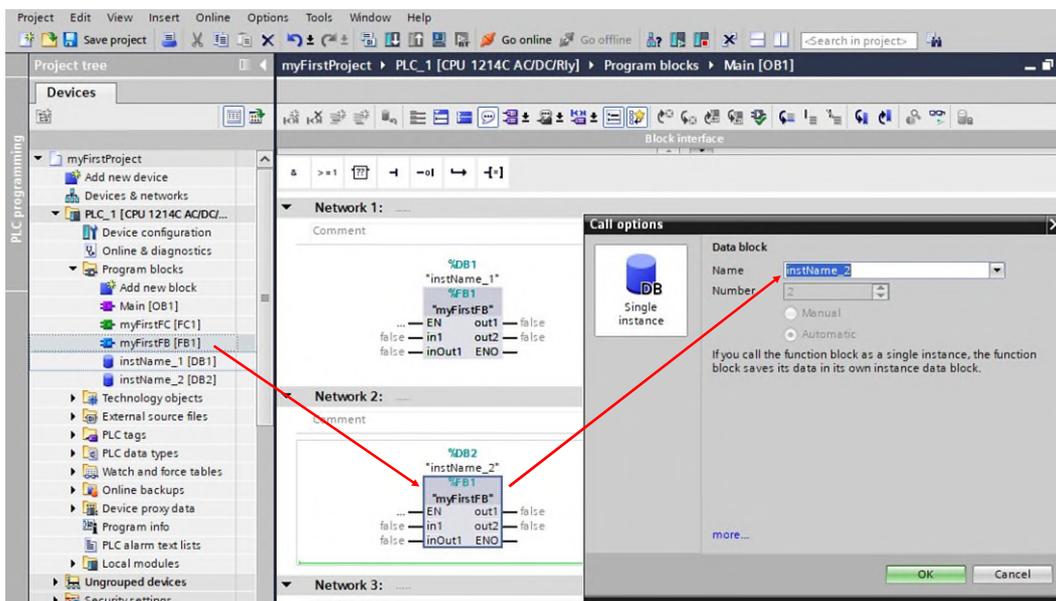
Imagen 16 FB con interfaz de bloque de funciones en el Portal TIA

3. primera llamada de "myFirstFB" en el primer grafo del OB "MAIN" mediante drag & drop y declaración de la primera instancia como instancia única:



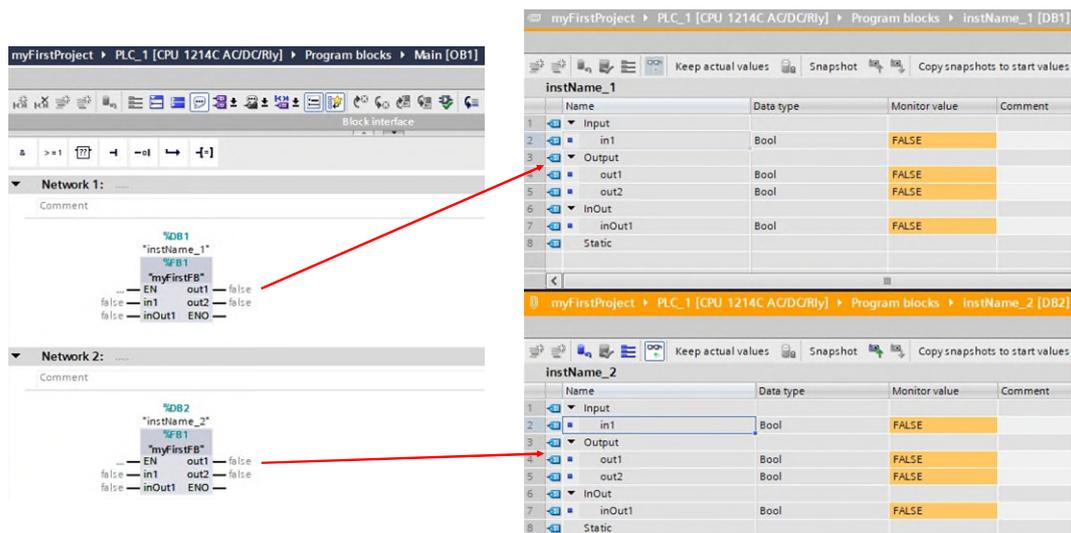
Fotografía 17 Instanciación de la primera llamada de bloque

4. segunda llamada de "myFirstFB" en el segundo grafo del OB "MAIN" mediante drag & drop y declaración de la segunda instancia como instancia única:



Fotografía 18 Instanciación segunda llamada de bloque

5. se pueden observar los dos bloques de datos de instancia:



Fotografía 19 Valores reales en las instancias



Los bloques de datos de instancia y la posibilidad de supervisarlos y controlarlos se describen detalladamente en el capítulo siguiente (Bloques de datos).

5.8.2 Opción de llamada como multiinstancia (portal TIA)

Si un bloque de funciones se llama en otro bloque de funciones, también puede seleccionarse la multiinstancia en las opciones de llamada.

El número de bloques de datos de instancia puede reducirse utilizando multiinstancias. El uso de multi-instancias es a menudo esencial cuando se crean subprogramas. Por ejemplo, si desea realizar un contador de tiempo de ejecución en un módulo de control para un motor, cada instancia de motor requiere su propia instancia de contador IEC.

Las multi-instancias se colocan en la interfaz del módulo de llamada.

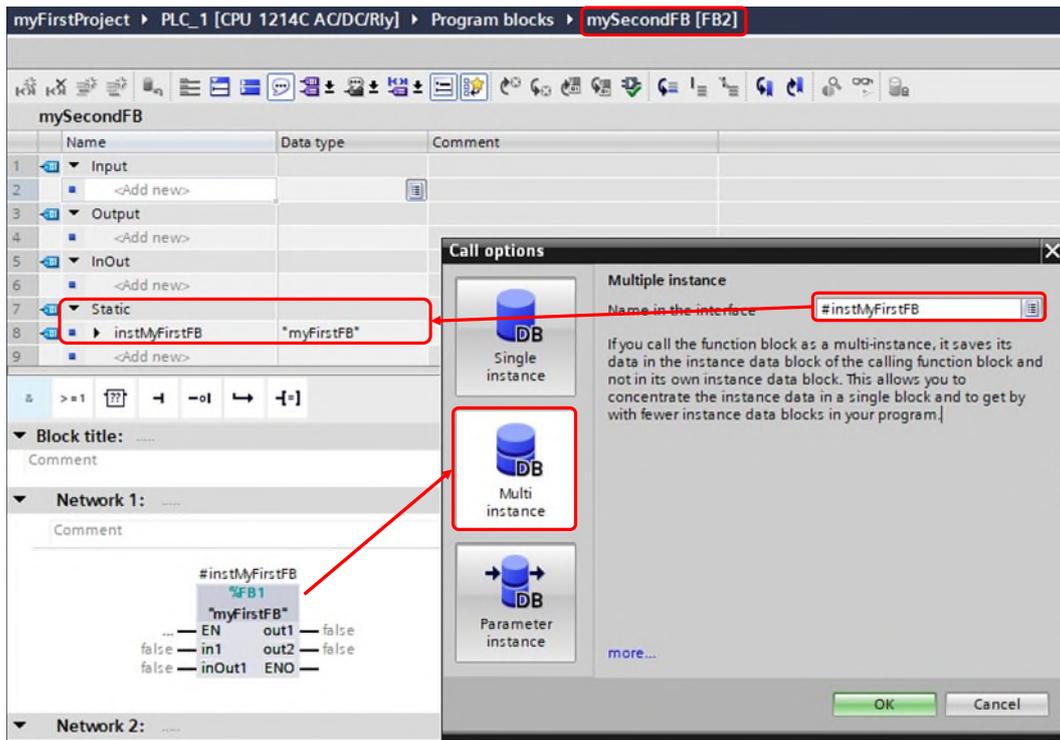


Imagen 20 Opciones de llamada en el Portal TIA

5.8.3 Declaración textual como multi-instancia (CODESYS / Beckhoff)

La declaración textual de las instancias se realiza según el siguiente esquema.

Sintaxis:

Nombre de la instancia (nombre de la variable) : Nombre del módulo (tipo de datos);

Ejemplo:

```
//Declaration of instances
VAR
    instName_1 : myFirstFB; //Instance 1
    instName_2 : myFirstFB; //Instance 2
END_VAR
```

Imagen 21 Declaración de las instancias

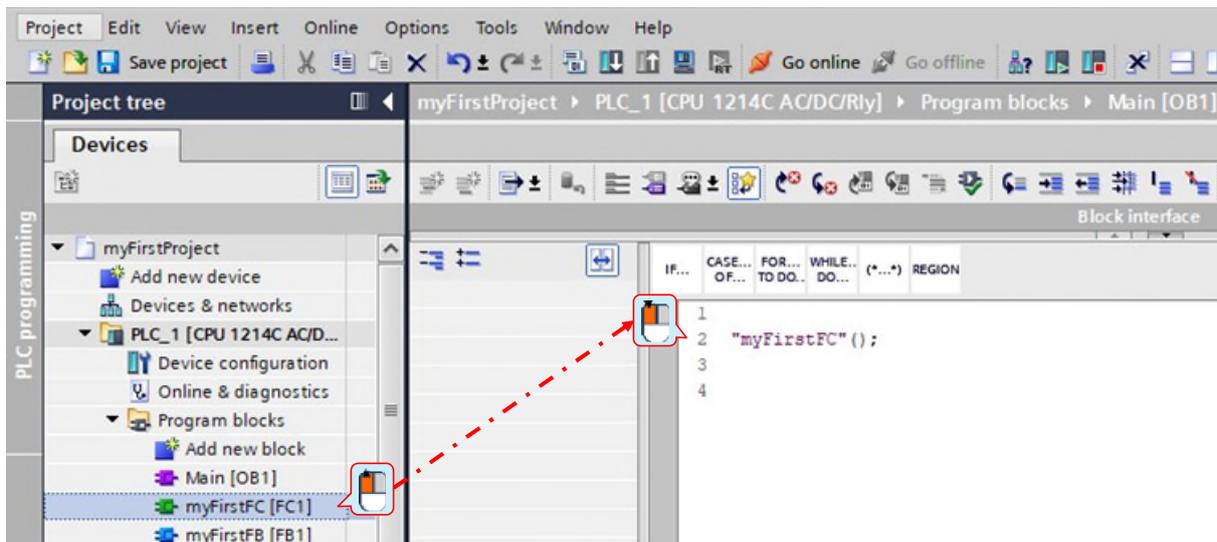
5.9 Llamada a una función (FC) en ST / SCL

Para garantizar que la función se procesa, queremos llamarla en el programa. Una llamada a una función sin valor de retorno en ST / SCL se realiza utilizando el nombre de la función, seguido de "()" y un punto y coma. Los parámetros se pasan entre corchetes y el punto y coma cierra la sentencia.

myFirstFC();

Imagen 22 Llamada a función ST / SCL

En el Portal TIA, el módulo deseado también puede llamarse mediante arrastrar y soltar, arrastrándolo desde la navegación del proyecto hasta la posición deseada:



Fotografía 23 Llamada en bloque en el Portal TIA

Transferencia de parámetros

Si el bloque de funciones llamado tiene parámetros de interfaz, éstos se muestran. En el caso de las funciones, los parámetros formales deben suministrarse con parámetros reales.

Los parámetros se pasan entre corchetes, los parámetros se separan entre sí con ",".

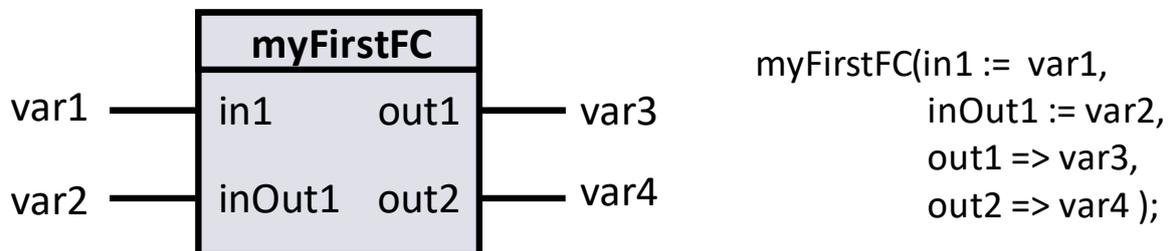


Imagen 24 Sintaxis de llamada a función, con transferencia de parámetros en SCL

Los parámetros de tipo "Entrada" e "InOut" se asignan mediante ":=". Los parámetros de tipo "Output" deben conectarse con "=>".

Ejemplo

En esta pantalla, se han declarado los siguientes parámetros de interfaz en la función "myFirstFC".

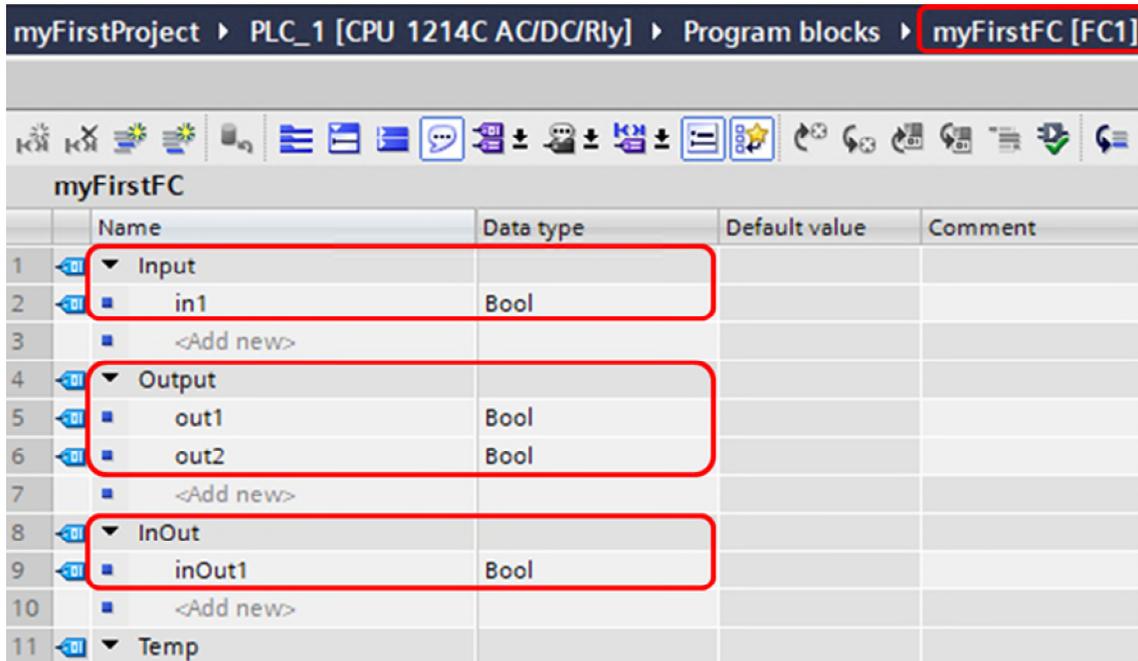
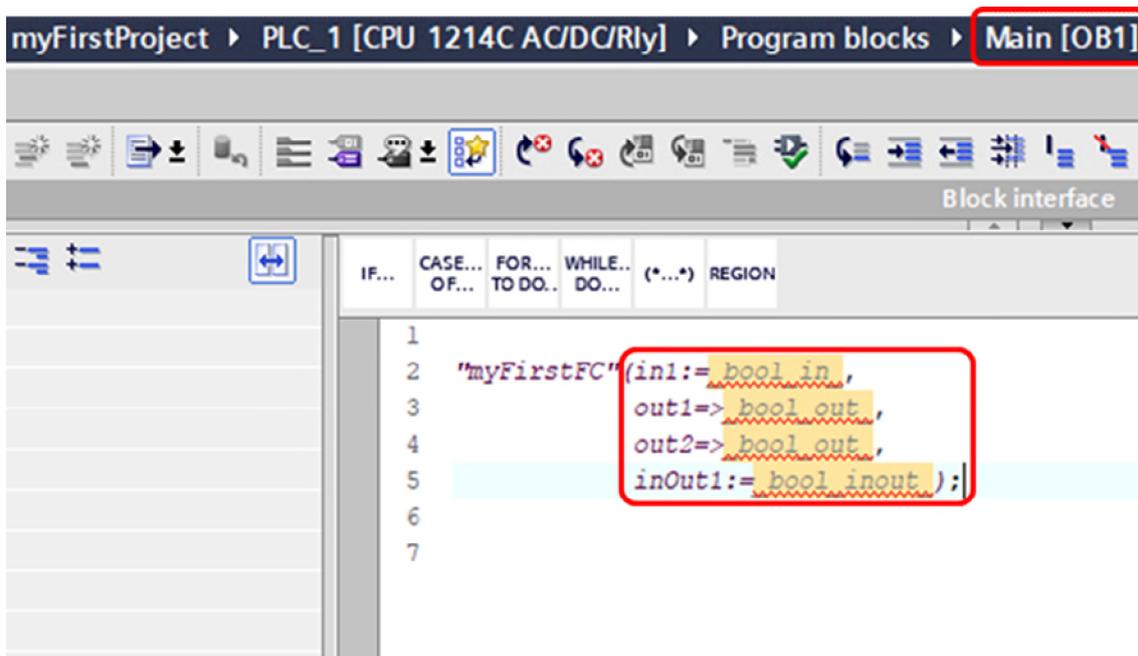


Imagen 25 Interfaz de bloques de función de una función en el Portal TIA

Después de crear la función "miPrimeraFC", se llamó en "PRINCIPAL" (OB1). Los parámetros aún no se han transferido y a los parámetros formales que se van a conectar se les asignan inicialmente marcadores de posición. Estos marcadores de posición proporcionan información sobre el tipo de datos y el tipo de parámetro (Input, Output, InOut).



Fotografía 26 Interfaz del módulo en el Portal TIA

5.10 Llamada a un bloque de funciones (FB) en ST / SCL

Para procesar el bloque de funciones, lo llamamos en el programa. La principal diferencia de la llamada a un bloque de funciones (FB) es que se le asigna una instancia. La llamada es similar a la de una función (FC), pero en lugar del nombre del bloque de funciones se utiliza el nombre de la instancia previamente declarada, seguido de "(" y un punto y coma. Los parámetros se pasan entre corchetes y el punto y coma cierra la instrucción.

```
instMyFirstFB();
```

Fotografía 27 Ejemplo de sintaxis, sin transferencia de parámetros en SCL

Transferencia de parámetros

Si la instancia llamada tiene parámetros de interfaz, éstos se muestran. No siempre es necesario pasar parámetros para los bloques de función, ya que la instancia ya tiene asignada una zona de memoria privada.

```
instMyFirstFB(in1 := var1,  
              inOut1 := var2,  
              out1 => var3,  
              out2 => var4 );
```

Fotografía 28 Ejemplo de sintaxis, con transferencia de parámetros en SCL

Opciones de compra

Dependiendo del tipo de bloque de funciones de llamada, las instancias pueden ubicarse directamente en la interfaz del bloque de funciones (= multiinstancia en el Portal TIA) o almacenarse como instancias globales (= instancia única en el Portal TIA).

Instanciar el módulo de funciones varias veces

Si un FB se llama dos veces en el programa de usuario, existen dos instancias. A cada instancia se le asigna su propia zona de memoria en la que la instancia puede guardar sus datos durante todo el ciclo.

Ejemplo

Por ejemplo, si el FB contiene el programa de usuario para calcular operaciones de conmutación, cada llamada representa una instancia que sólo utiliza estados en el tiempo de ejecución de esta llamada. Se necesita una instancia distinta para cada llamada.

Esto significa que todos los datos (información) relativos a este cálculo están disponibles en esta instancia asignada.

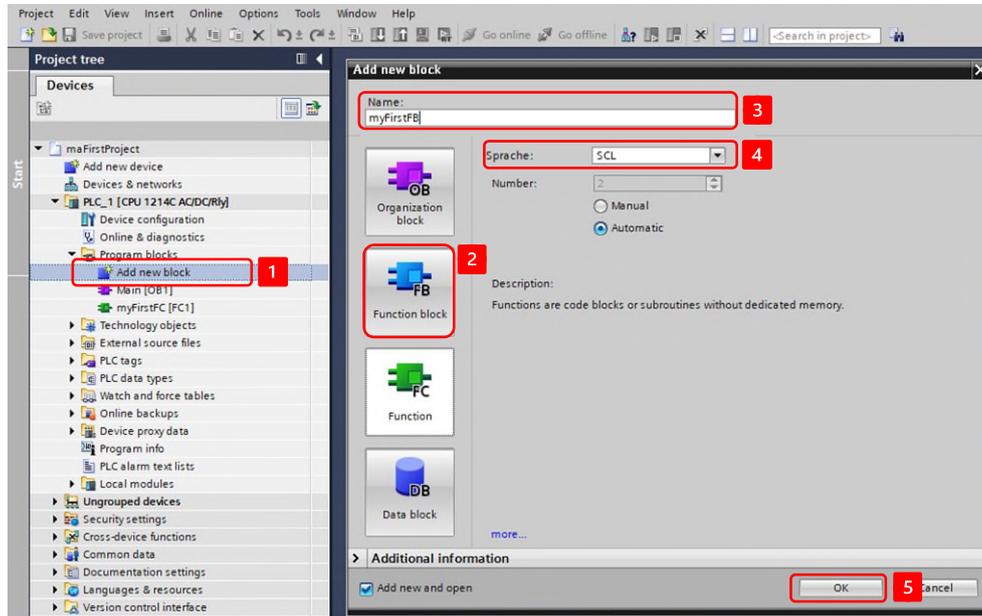
Para poder crear una instancia, el FB asignado ya debe existir.

En esta instancia se pueden observar las variables de la instancia correspondiente.

5.10.1 Llamada a procedimiento con instancia única

El procedimiento para llamar dos veces al bloque de funciones "myFirstFB" se explica ahora paso a paso en el portal TIA.

1. Crear el módulo de función "miPrimerFB":



Fotografía 29 Añadir un nuevo módulo

2. Declaración de la interfaz del bloque de funciones y conexión de las variables en la parte de instrucciones del bloque de funciones:

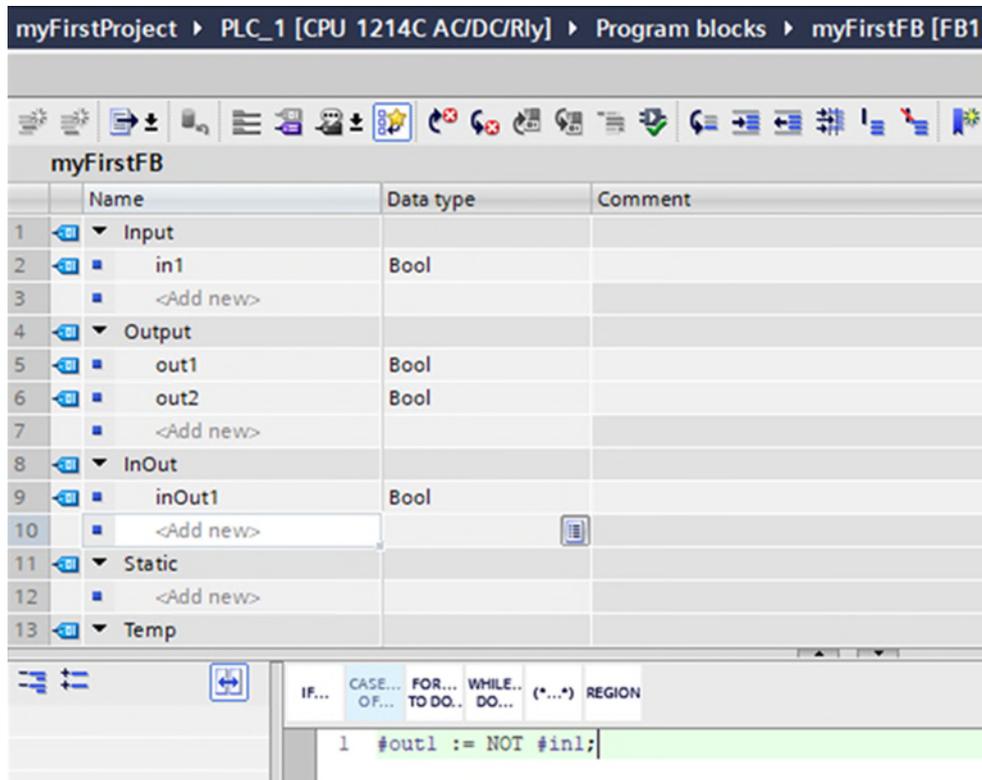
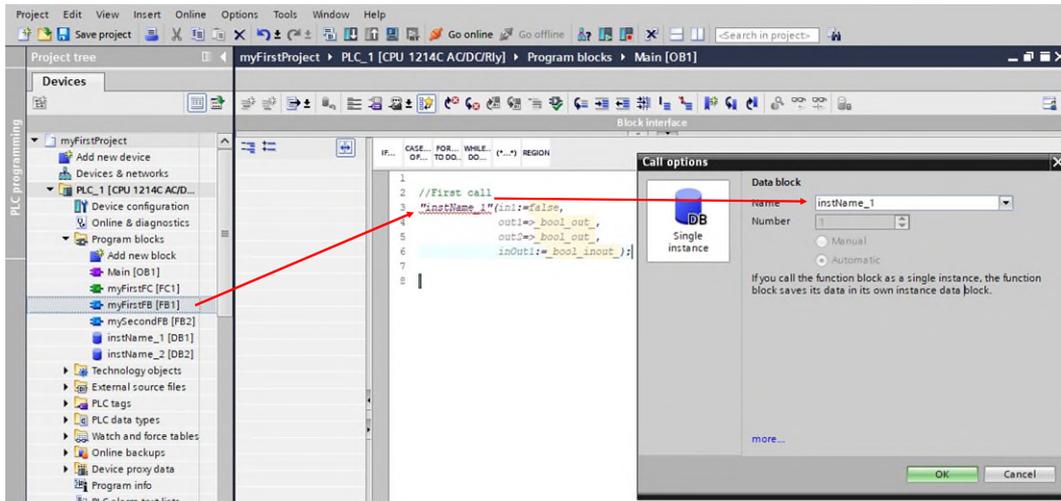


Imagen 30 FB con interfaz de bloque de funciones en el Portal TIA

- primera llamada de "myFirstFB" en el OB "MAIN" mediante drag & drop y declaración de la primera instancia como instancia única:



Fotografía 31 Instanciación de la primera llamada de bloque

- segunda llamada de "myFirstFB" en el OB "MAIN" mediante drag & drop y declaración de la segunda instancia como instancia única:

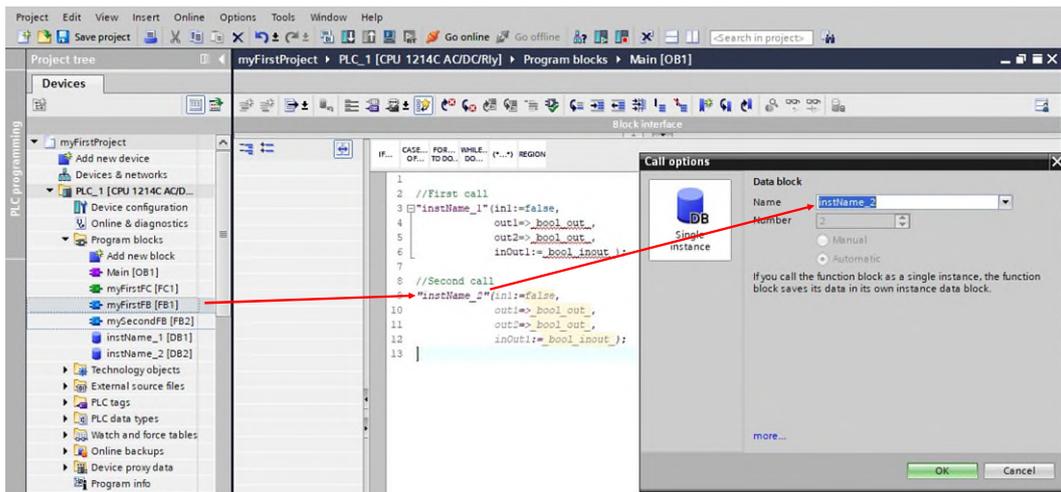


Imagen 32 Instanciación segunda llamada de bloque

5. se pueden observar los dos bloques de datos de instancia:

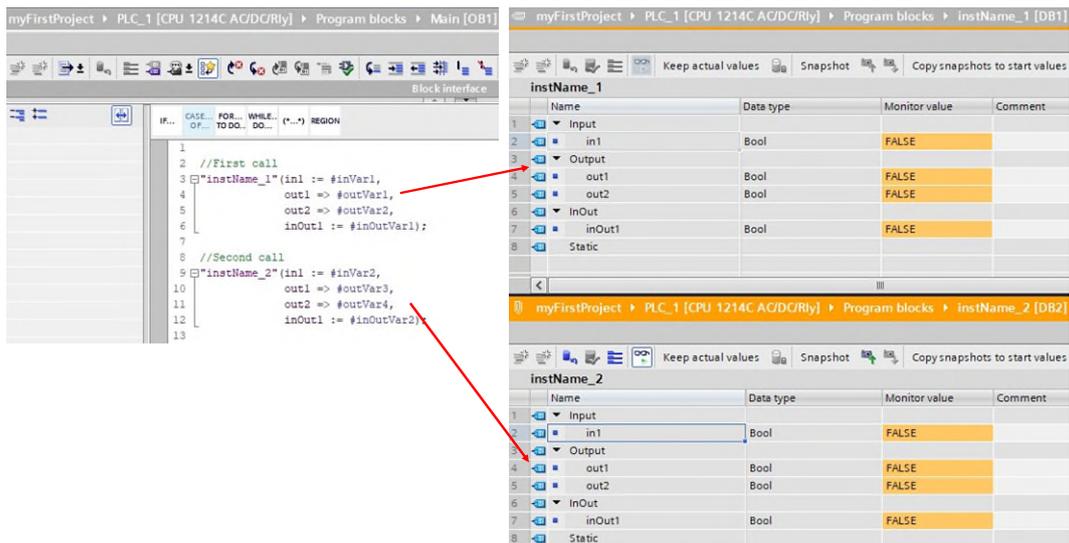


Imagen 33 Valores reales en las instancias



Los bloques de datos de instancia y la posibilidad de supervisarlos y controlarlos se describen con detalle en el capítulo siguiente (Bloques de datos).

5.10.2 Opción de llamada como multiinstanciada (portal TIA)

Si un bloque de funciones se llama en otro bloque de funciones, también puede seleccionarse la multiinstanciada en las opciones de llamada.

El número de bloques de datos de instancia puede reducirse utilizando multiinstanciadas. El uso de multiinstanciadas es a menudo esencial cuando se crean subprogramas. Por ejemplo, si desea realizar un contador de tiempo de ejecución en un módulo de control para un motor, cada instancia de motor requiere su propia instancia de contador IEC.

Las multiinstanciadas se colocan en la interfaz del módulo de llamada.

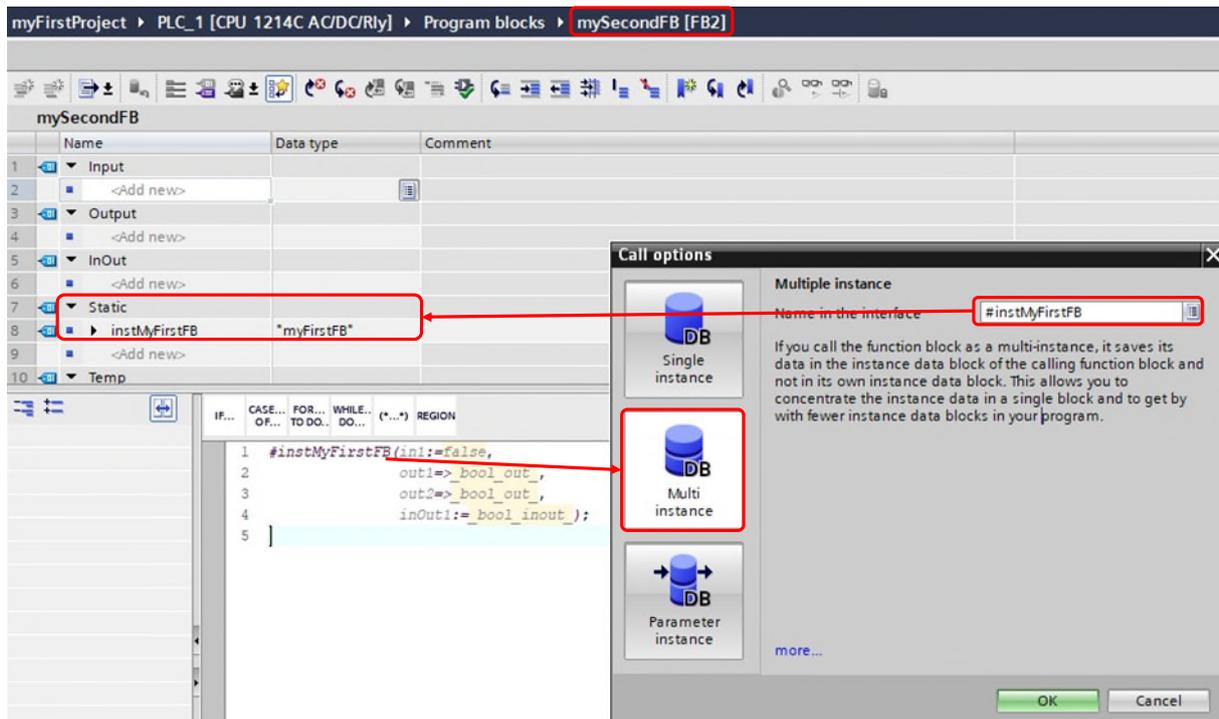


Imagen 34 Opciones de llamada en el Portal TIA

5.10.3 Declaración textual como multi-instancia (CODESYS / Beckhoff)

La declaración textual de las instancias se realiza según el siguiente esquema.

Sintaxis:

Nombre de la instancia (nombre de la variable) : Nombre del bloque (tipo de datos);

Ejemplo:

```
//Declaration in the function block interface
```

```
VAR
```

```
  instMyFirstFB : myFirstFB; //declaration of the instance
```

```
END_VAR
```

```
//Implementation in the program
```

```
instMyFirstFB(in1:=      ,  
              inOut1:=   ,  
              out1=>     ,  
              out2=>    );
```

Imagen 35 Declaración y llamada de la instancia

Instanciar el módulo de funciones varias veces

Si un FB se llama dos veces en el programa de usuario, existen dos instancias. A cada instancia se le asigna su propia zona de memoria en la que la instancia puede guardar sus datos durante todo el ciclo.

//Declaration in the function block interface

VAR

instName_1 : myFirstFB; //Instance 1

instName_2 : myFirstFB; //Instance 2

END_VAR

//Implementation in the program

//Call 1

```
instName_1(in1:=      ,
           inOut1:=   ,
           out1=>     ,
           out2=>     );
```

//Call 2

```
instName_2(in1:=      ,
           inOut1:=   ,
           out1=>     ,
           out2=>     );
```

Imagen 36 Instanciación de dos FB