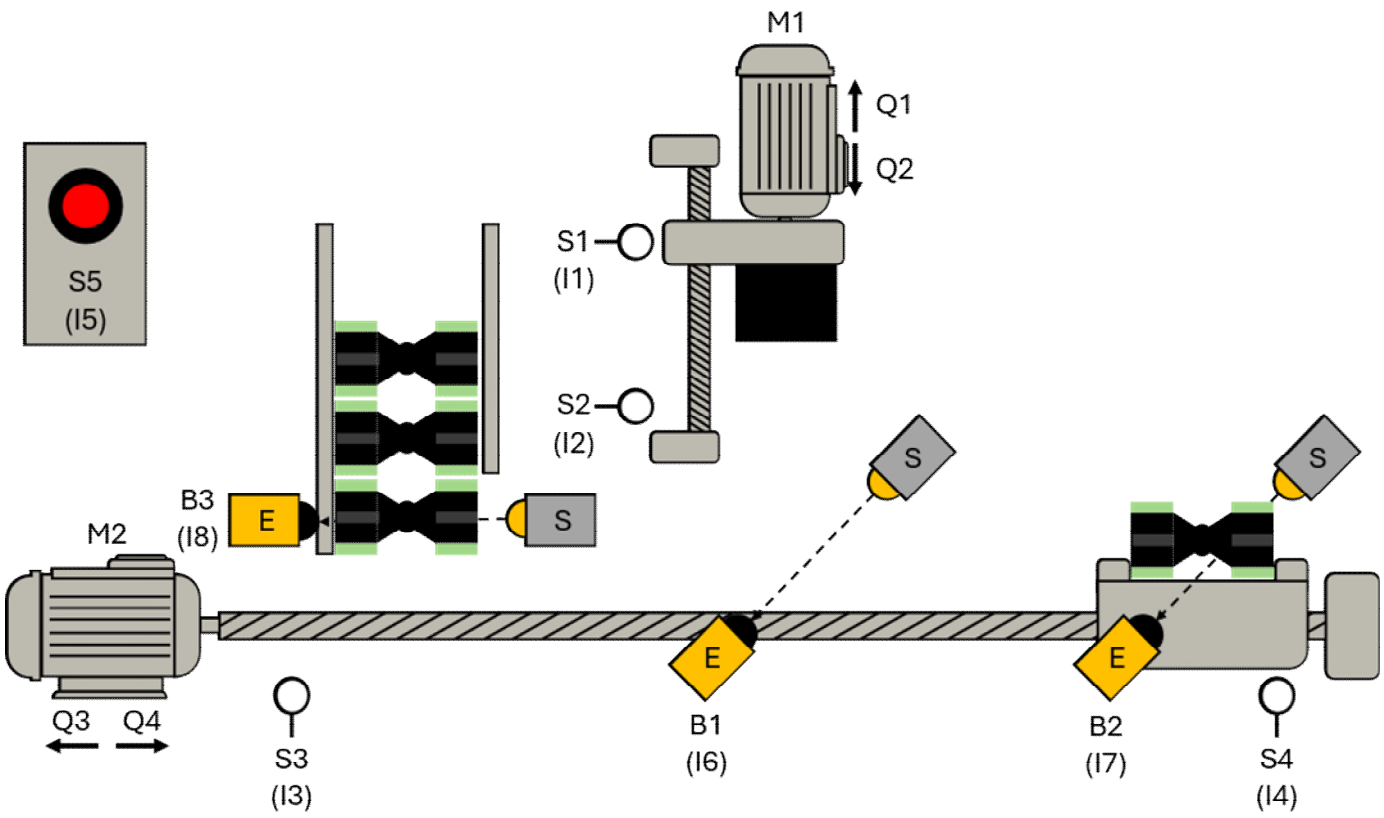
fischertechnik

GROLLMUS
KNOWLEDGE | SOFTWARE | AUTOMATION

Prensa Dobladora 24V

Programación estructurada



Índice

5 Programación estructurada

5.1 Introducción

La programación estructurada en sistemas PLC se utiliza para organizar programas complejos dividiéndolos en bloques de construcción más pequeños y claros. Esto 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 de construcción como las funciones (FC) y los módulos de función (FB) se utilizan para estructurar las partes del programa.

Los building blocks deben comunicarse entre sí a través de sus interfaces de building block, en lugar de acceder directamente a las variables globales. La transferencia de parámetros se realiza a través de entradas y salidas, así como de parámetros InOut.

Para ejecutar los módulos de código en el programa de control, es necesario llamarlos.

5.2 Función

Las funciones (FC) son bloques de construcción de código sin memoria. No disponen de un almacén de datos en el que puedan guardarse los valores de los parámetros del building block. Por lo tanto, todos los parámetros de la interfaz deben conmutarse cuando se llama a una función. Para almacenar datos de forma permanente, primero deben crearse módulos de datos globales.

Las funciones son ideales para tareas que no requieren memoria durante varios ciclos, como cálculos matemáticos o enlaces lógicos.

Una función contiene un programa que se ejecuta cada vez que la función es llamada por otro fragmento de código.

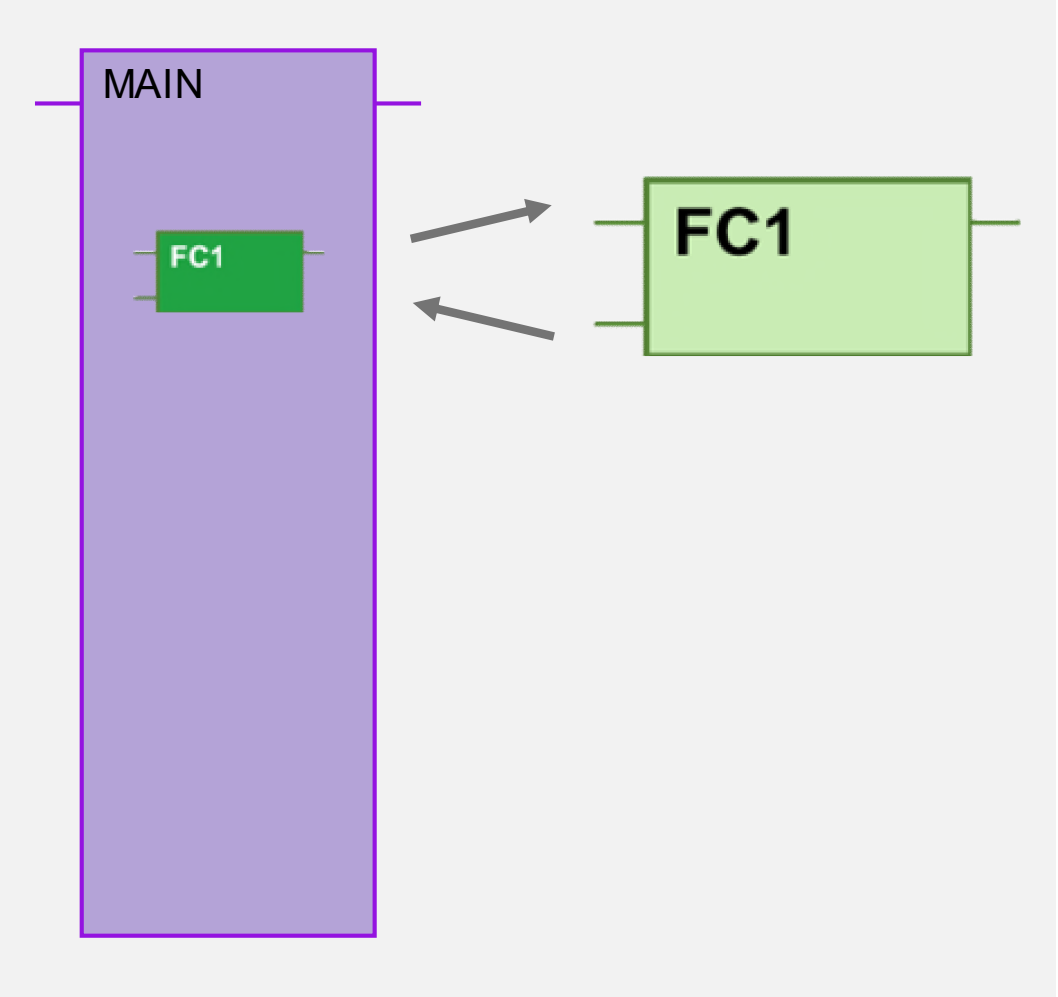


Imagen 1 Ejemplo: Llamada a una función desde MAIN

Una función también puede ser llamada varias veces en diferentes lugares dentro de un programa.

5.3 Función

Los módulos de funciones (FBs) son módulos de código que almacenan permanentemente sus variables de entrada, variables de salida, variables pass-through y también las variables estáticas en módulos de datos de instancia para que también estén disponibles después de que el módulo haya sido editado. Por eso también se les llama bloques de construcción con memoria.

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

- Siempre que se necesiten tiempos y contadores en los bloques de construcción o
- si es necesario almacenar 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.

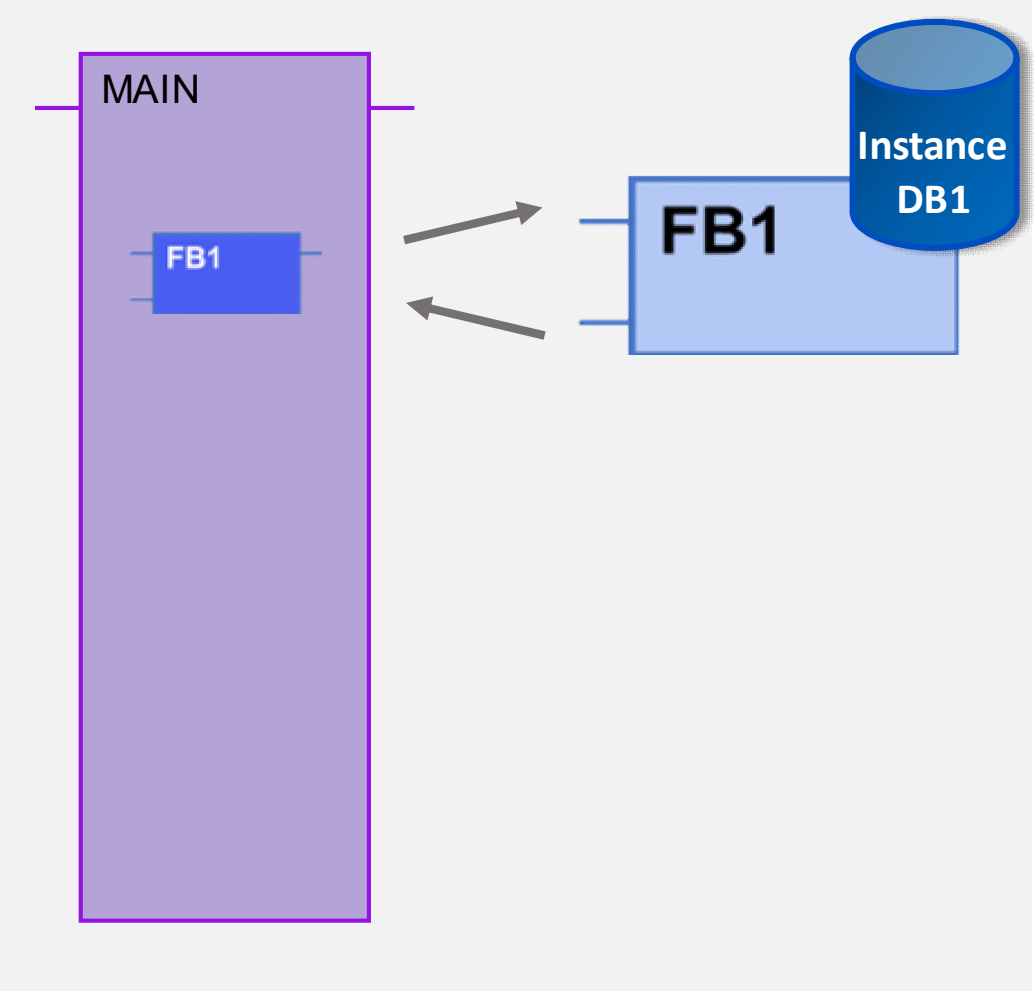


Imagen 2 Ejemplo: Llamada a un módulo de funciones desde MAIN

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

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

5.4 Añadir un nuevo bloque de construcción

En el TIA Portal, los módulos se gestionan en la navegación del proyecto, debajo del PLC, en la carpeta "Program Modules".
Haciendo doble clic en el comando "Añadir nuevo bloque" dentro de la carpeta "Bloques de programa" se abre el cuadro de diálogo "Añadir nuevo bloque", que puede utilizarse para crear un nuevo bloque.

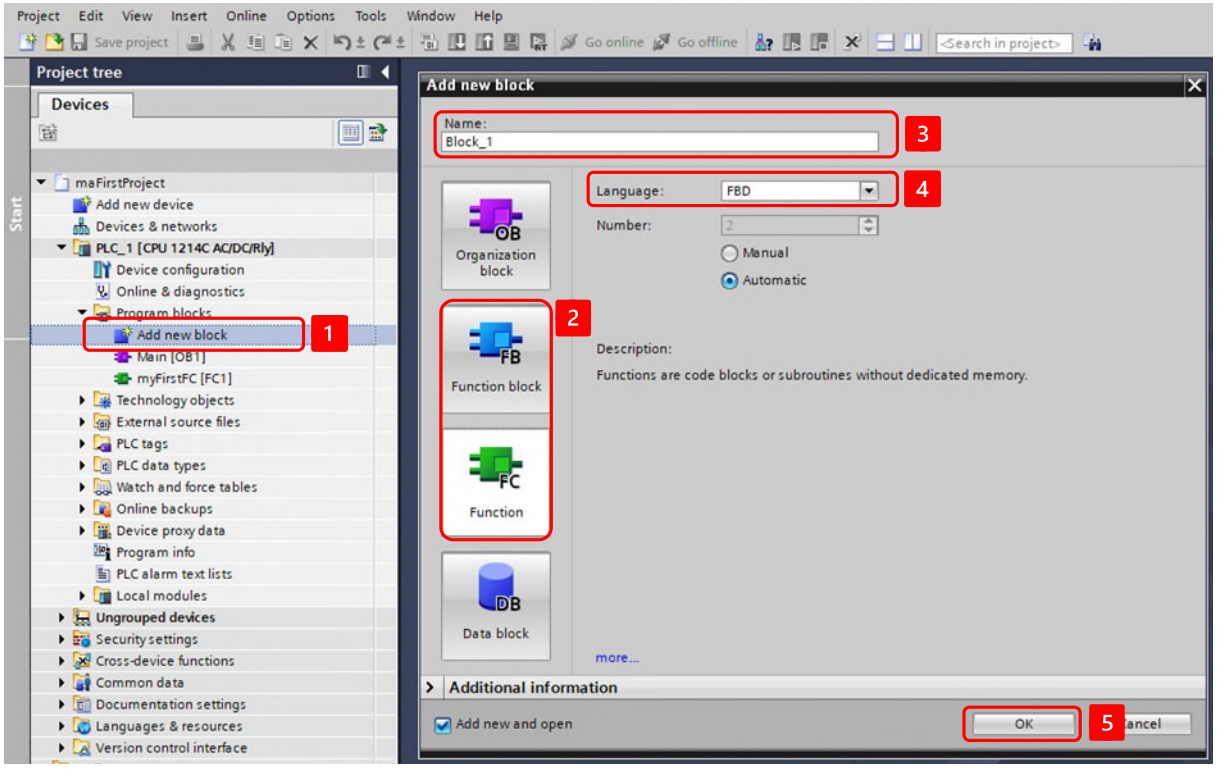


Imagen 3 Añadir un nuevo bloque de construcción

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

5.5 Convocatoria de módulos

Para ejecutar los módulos de código en el programa de control, es necesario 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 el "MAIN". Dentro del "MAIN" se pueden llamar las partes del programa estructuradas en funciones y módulos de función.

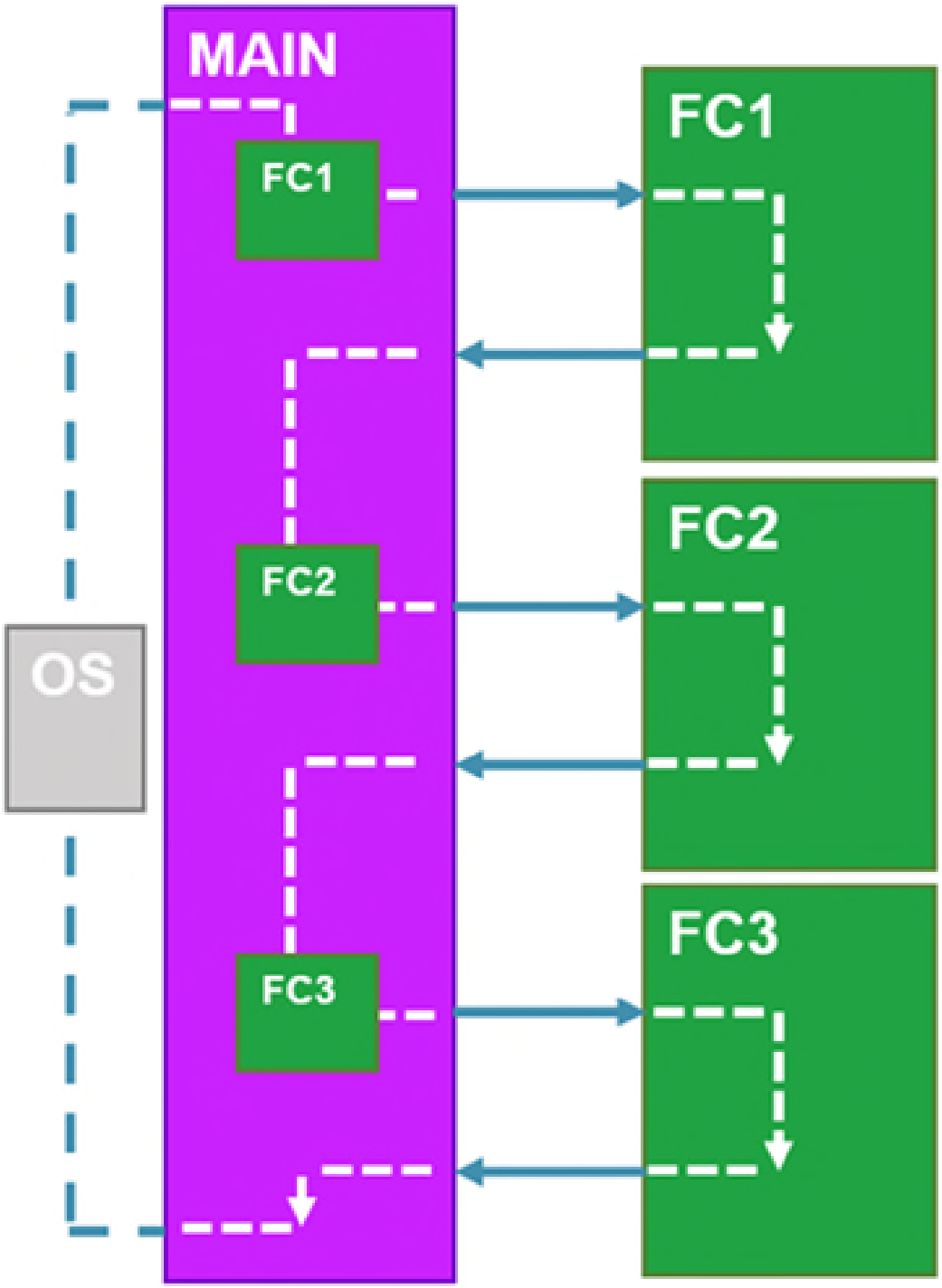


Figura 4 Llamada al bloque de construcción en el PRINCIPAL

Las funciones y los módulos de funciones estructuran el programa, haciéndolo más legible y fácil de mantener.
Todos los módulos llamados se procesan uno tras otro.

El sistema operativo de la CPU vuelve a llamar al " MAIN " después del ciclo de programa, ejecutando de nuevo todos los comandos programados en él.

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

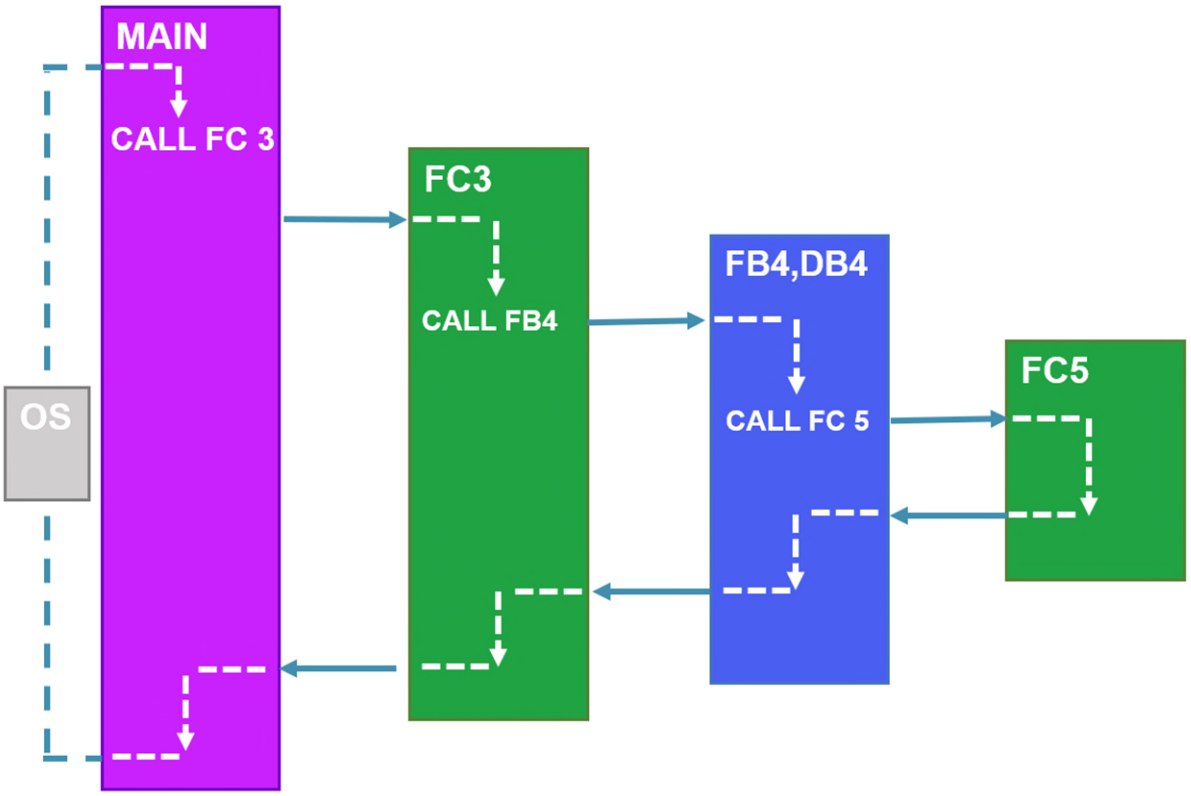


Imagen 5 Llamada a los bloques de construcción

5.6 Transferencia de parámetros

Al llamar a funciones y módulos de funciones, se pueden pasar parámetros (variables y valores de variables). El código del programa dentro del módulo de código trabaja entonces con los valores pasados de los parámetros formales y puede devolver resultados mediante parámetros iniciales o el valor de la función.

El intercambio de datos se realiza a través de la interfaz del módulo. Se declaran en la interfaz de módulo de la función (FC) o del módulo de función (FB) y pueden utilizarse localmente en el módulo.

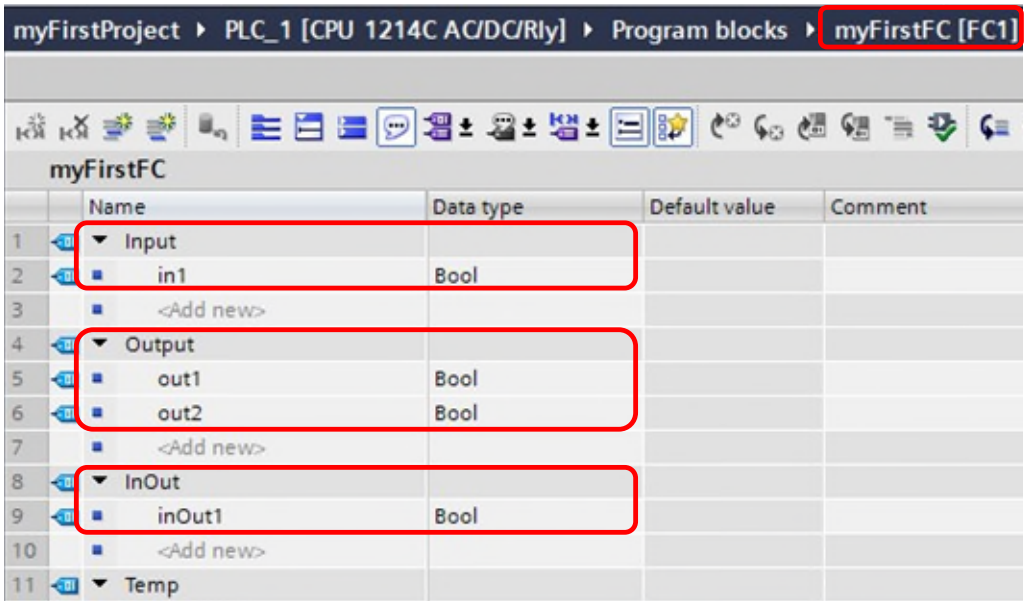


Imagen 6 Parámetros formales en la interfaz de ladrillos

Cuando se llama al módulo, estos parámetros formales se conectan a los parámetros actuales (variables globales del PLC).

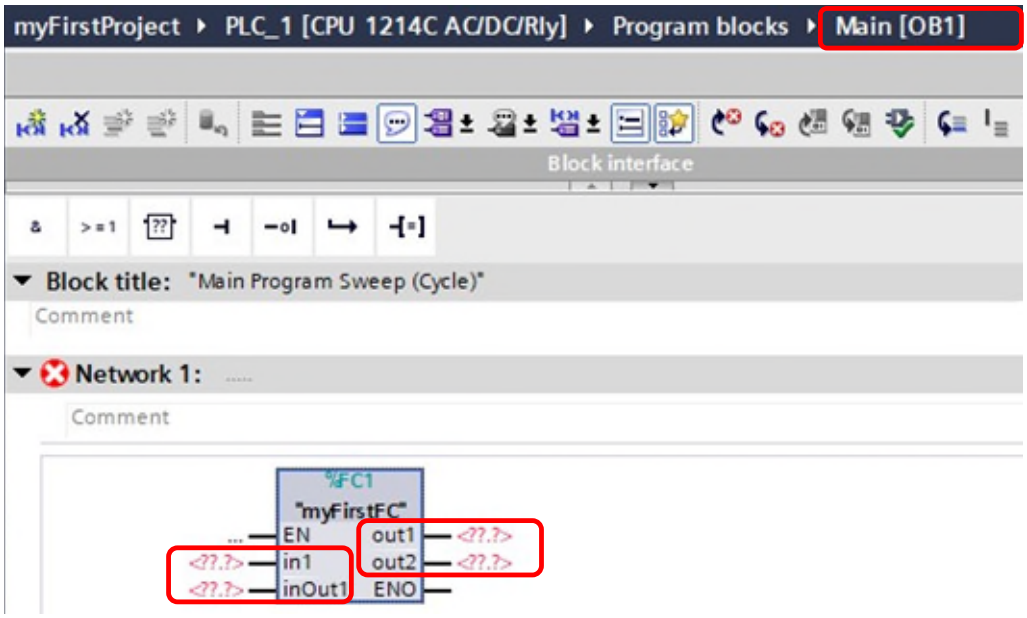


Imagen 7 Entrega del Akualparamter en la convocatoria

5.7 Llamada a una función (FC) en FUP

Para que la función sea procesada, debe ser llamada por el programa. La llamada puede realizarse insertando una casilla vacía (atajo de teclado TIA "F8"). Después de insertar la casilla vacía (1), sustituimos (2) los marcadores de posición ("???") de la casilla vacía por el nombre simbólico del bloque, de modo que la casilla vacía se sustituye por la llamada al bloque correspondiente.

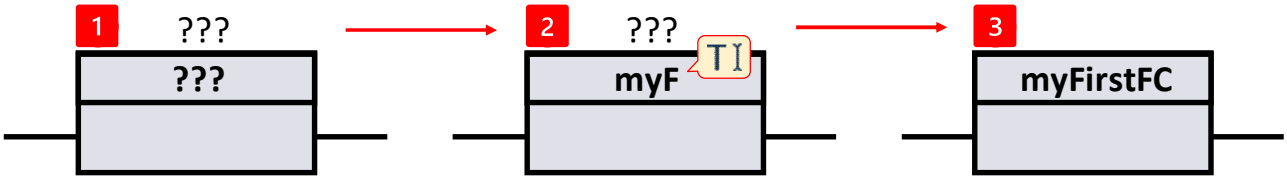


Imagen 8: Bloquear llamada desde buzón vacío

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

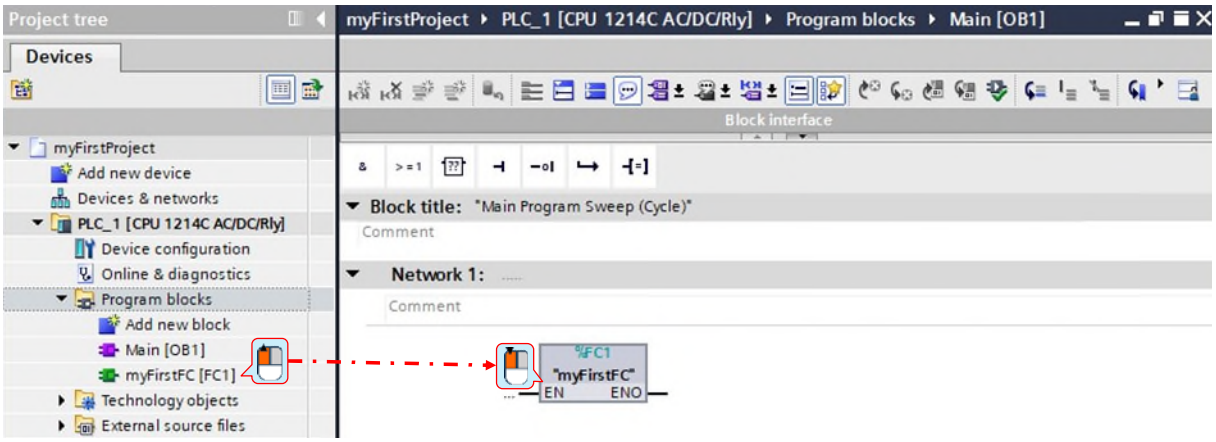


Imagen 9 Llamada al Building Block en el Portal TIA

Transferencia de parámetros

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

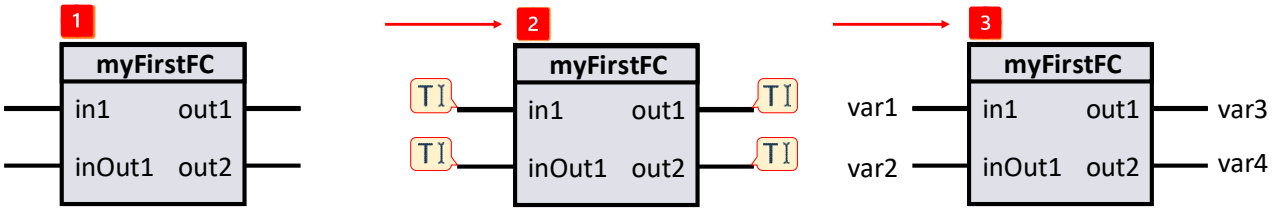


Imagen 10 Función con interfaz de componentes

Los parámetros del tipo "Entrada" e "InOut" se visualizan a la izquierda del módulo mediante una pata de conexión. Los parámetros del tipo "Salida" deben conectarse a la derecha del dispositivo.

Ejemplo

Después de crear la función "miPrimeraFC", se la llamó en la "Principal" (OB1). La transferencia de parámetros aún no ha tenido lugar; los parámetros formales a combinar se marcaron inicialmente con marcadores de posición "<??? >".

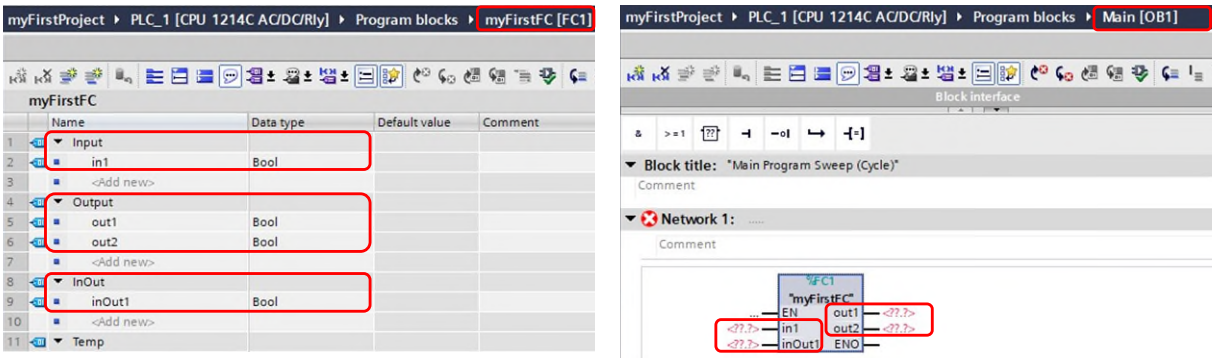


Imagen 11 Interfaz del módulo en el portal TIA

5.8 Llamada a un módulo de funciones (FB) en FUP

Para que el módulo de funciones sea procesado, 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).

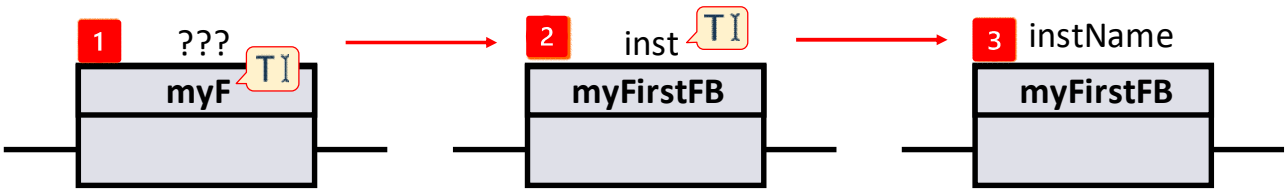
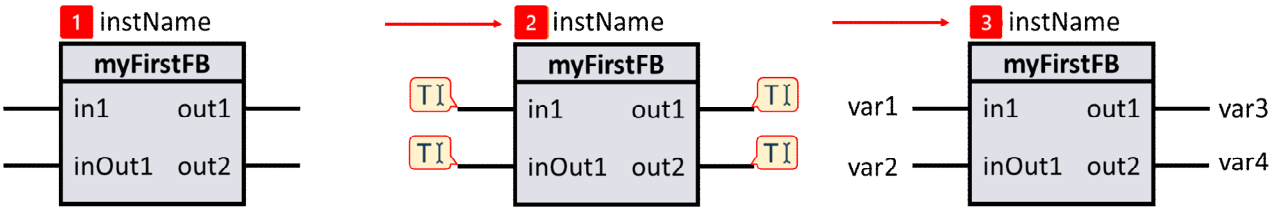


Bild 12 Funktionsbausteinaufruf aus Leerbox

También puede llamar al Building Block arrastrándolo y soltándolo, como se muestra en Llamada a una función.

Transferencia de parámetros

Si el módulo de funciones llamado tiene parámetros formales en la interfaz del módulo, éstos se muestran. En el caso de los bloques de funciones, el paso de parámetros no siempre es necesario porque la instancia ya tiene asignada una zona de memoria privada.



Fotografía 13 Módulo de funciones con transferencia de parámetros

Instanciación múltiple de un módulo de funciones

Si un módulo de función (FB) es llamado (instanciado) dos veces en el programa de usuario, se crean dos instancias separadas. En las que pueden almacenar sus datos durante todo el ciclo.

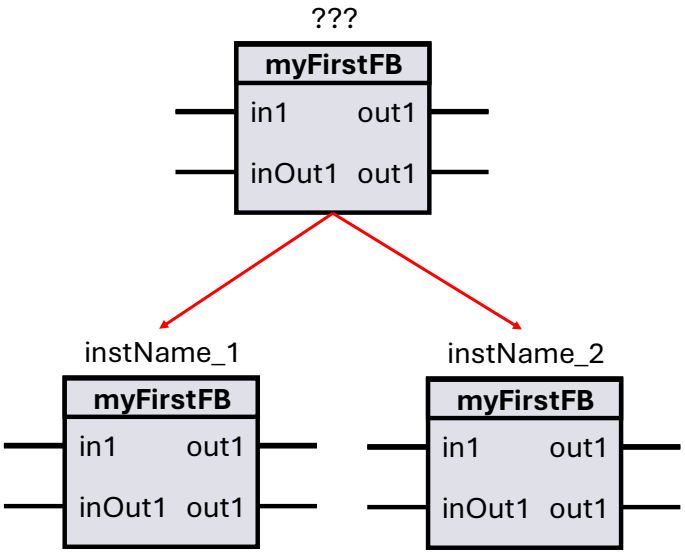


Imagen 14 Instanciación de dos FB

Opciones de compra

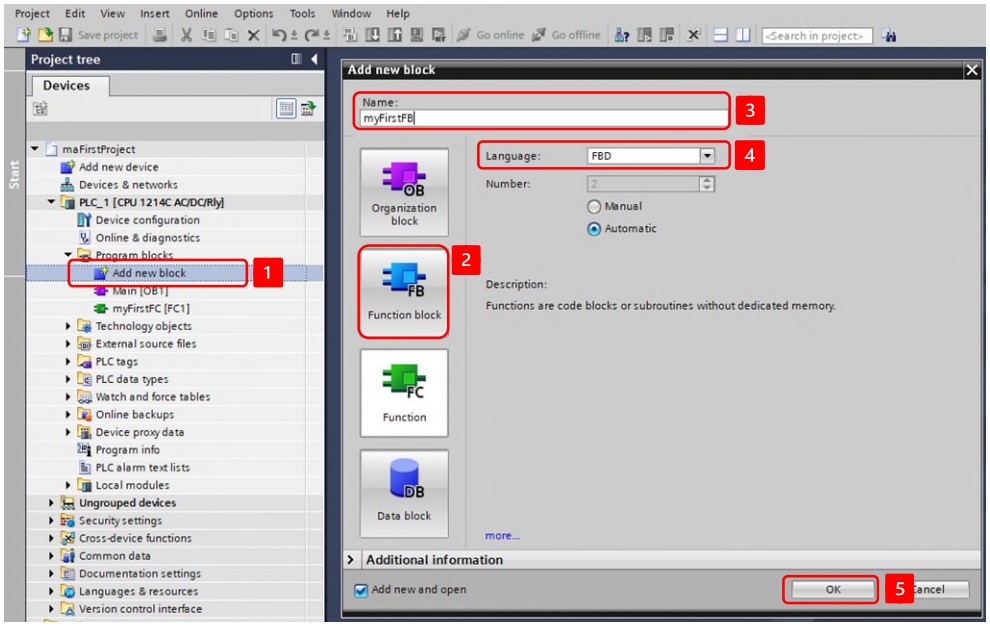
En función del tipo de módulo de llamada, las instancias pueden ubicarse directamente en la interfaz del módulo (= multiinstancia en el Portal TIA) o almacenarse como instancias globales (= instancia única en el Portal TIA).

Ejemplo

Por ejemplo, si el FB contiene el programa de usuario para un cálculo de 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.
Así, todos los datos (información) que pertenecen a este cálculo están disponibles en esta instancia asignada.
Para poder crear una instancia, el FB asignado ya debe existir.
En él 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 módulo de funciones "miPrimerFB" se muestra ahora paso a paso en el portal TIA:
Creación del módulo de función "miPrimerFB":



Fotografía 15 Añadir un nuevo bloque de construcción

Declaración de la interfaz del módulo e interconexión de las variables en la parte de instrucción del módulo:

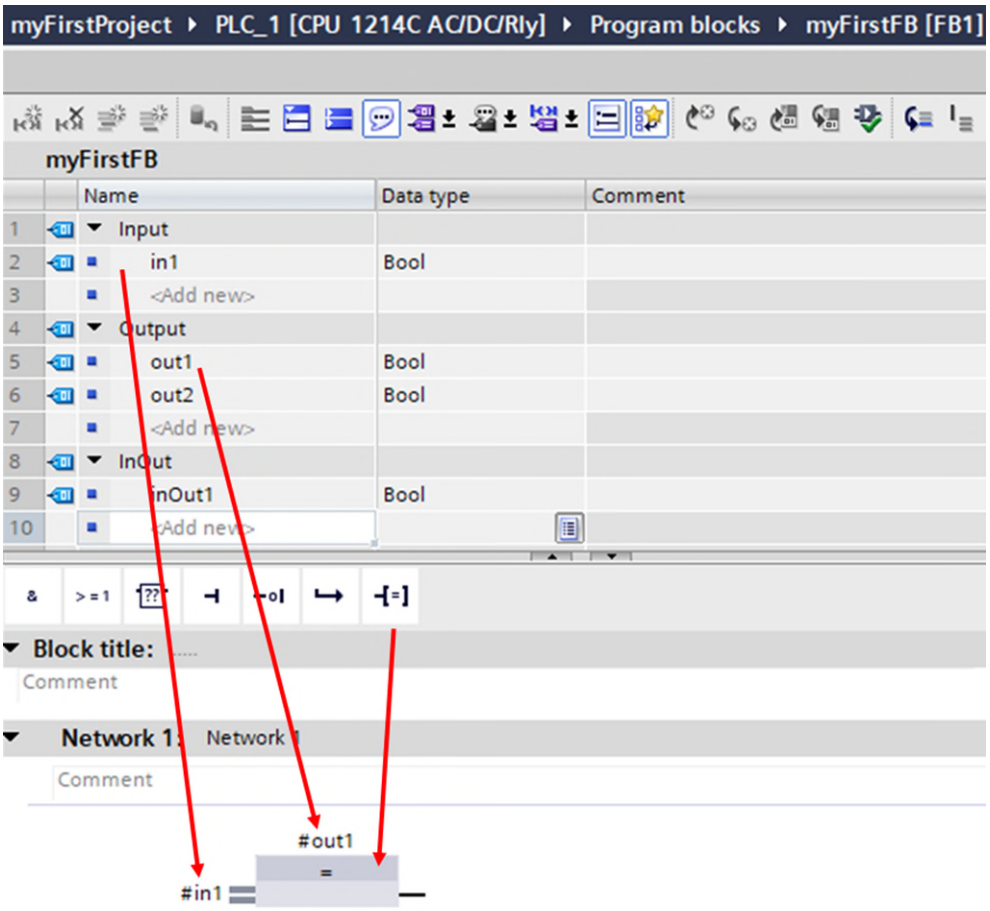
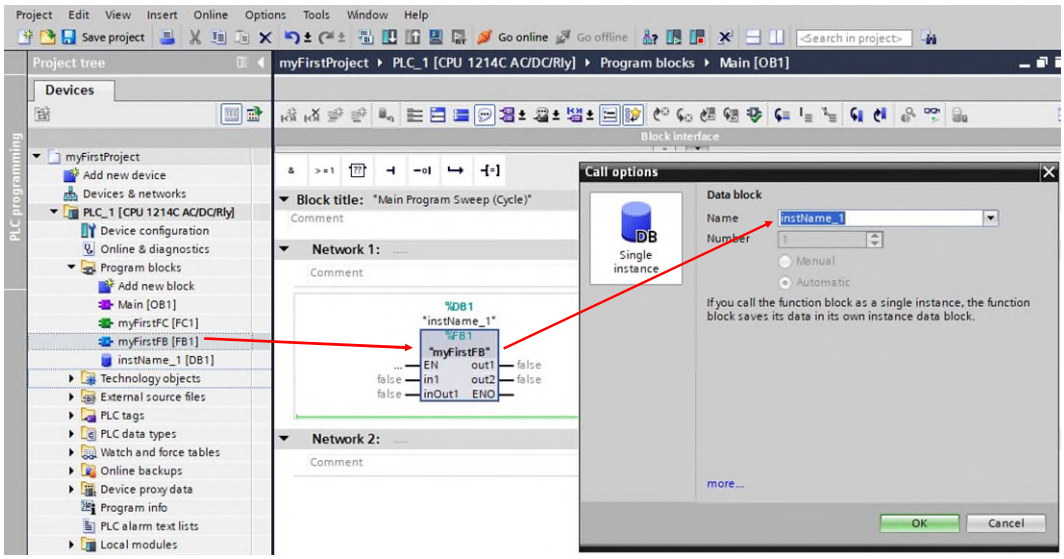


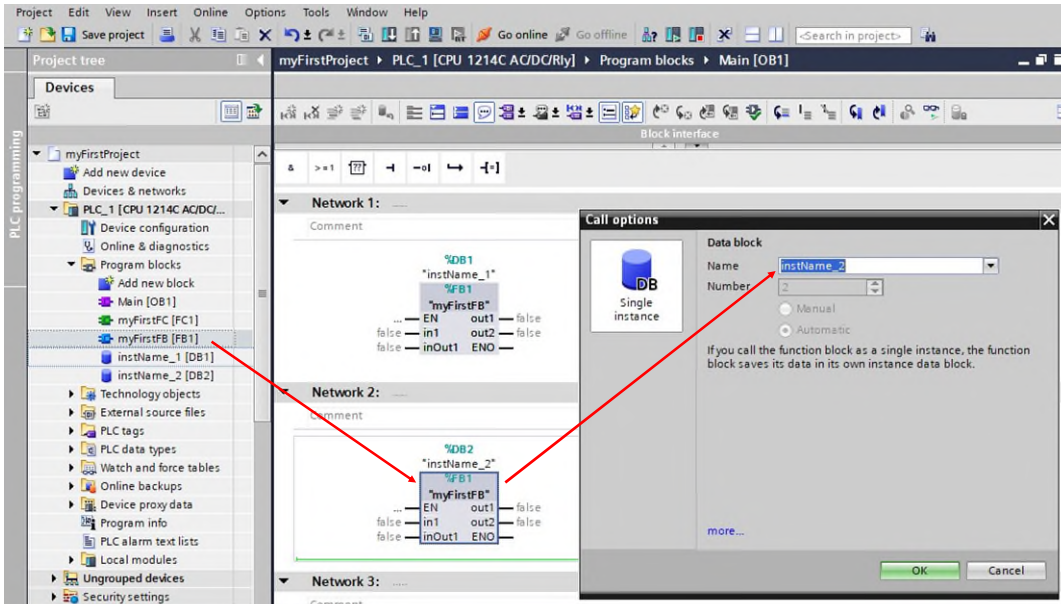
Imagen 16 FB con interfaz de módulo en el Portal TIA

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



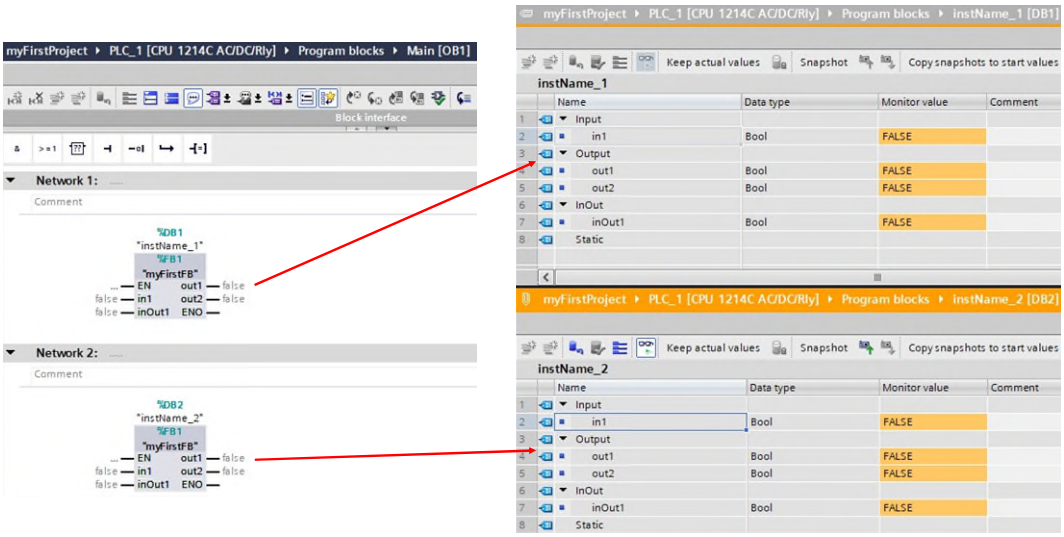
Fotografía 17 Instanciación de la primera llamada en bloque

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



Fotografía 18 Instanciación de la segunda llamada de bloque

5. Se pueden observar los dos módulos de datos de instancia:



Fotografía 19 Valores actuales en las instancias

Los módulos de datos de instancia, así como la posibilidad de observarlos y controlarlos, se describirán en detalle en el capítulo siguiente (Módulos de datos).

5.8.2 Opción de llamada como multi-instancia (TIA-Portal)

Si un módulo de funciones se llama en otro módulo de funciones, también se puede seleccionar la multiinstancia en las opciones de llamada.
El uso de multi-instancias permite reducir el número de módulos de datos de instancia. Cuando se crean subrutinas, el uso de multi-instancias es a menudo obligatorio. Si, por ejemplo, desea implementar un contador de tiempo de ejecución en un módulo de control para un motor, cada instancia de motor necesita su propia instancia de contador IEC.
Las multi-instancias se colocan en la interfaz del building block del building block llamante.

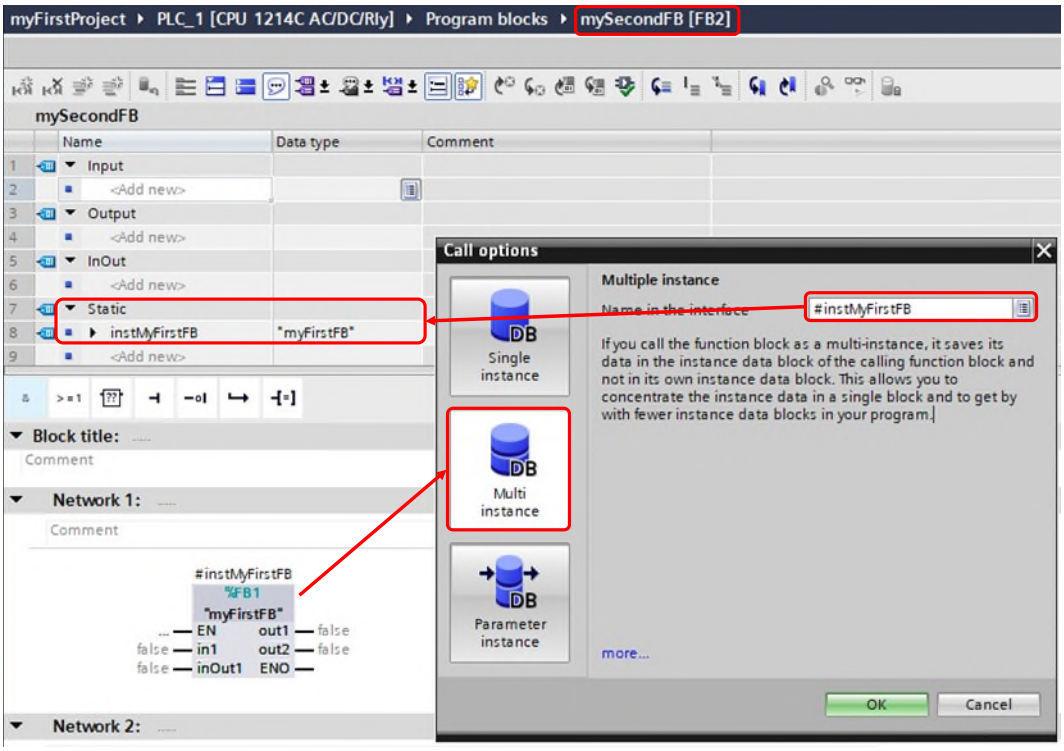


Imagen 20 Opciones de llamada en el Portal TIA

5.8.3 Declaración Textual en 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 of instances
VAR
  instName_1 : myFirstFB;   //Instance 1
  instName_2 : myFirstFB;   //Instance 2
END_VAR
```

Imagen 21 Declaración de instancias

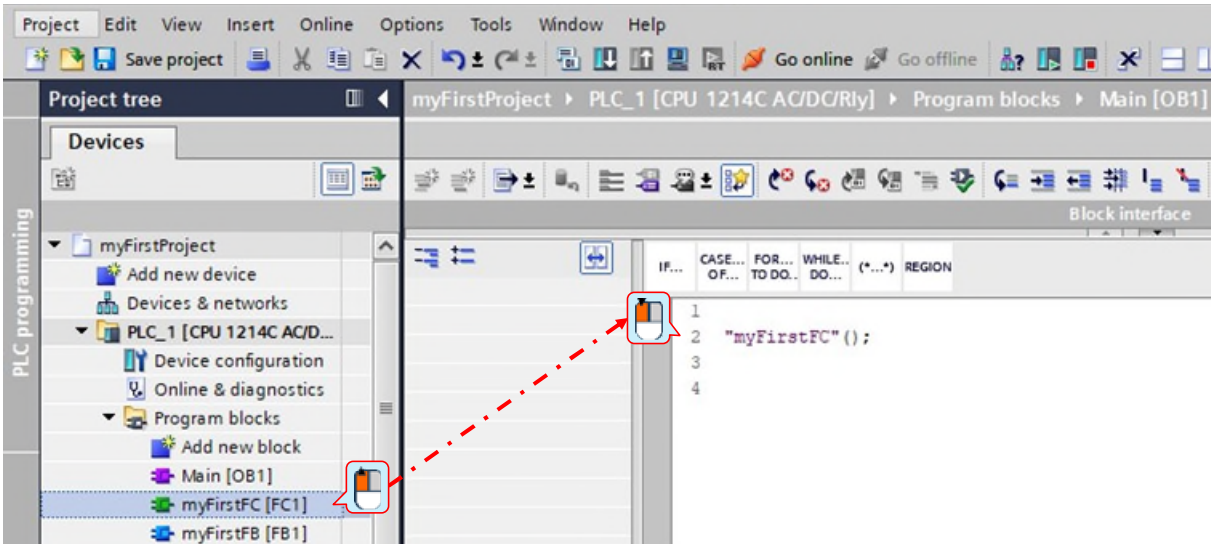
5.9 Llamada a una función (FC) en ST/SCL

Para que la función sea procesada, queremos llamarla en el programa. Una llamada a función sin valor de retorno en ST/SCL se realiza mediante el nombre de la función, seguido de "()" y un punto y coma. Entre paréntesis se pasa el parámetro, y el punto y coma concluye 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 ubicación deseada:



Fotografía 23 Llamada al Building Block en el Portal TIA

Transferencia de parámetros

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

El parámetro se pasa entre corchetes, los parámetros se separan entre sí con ",".

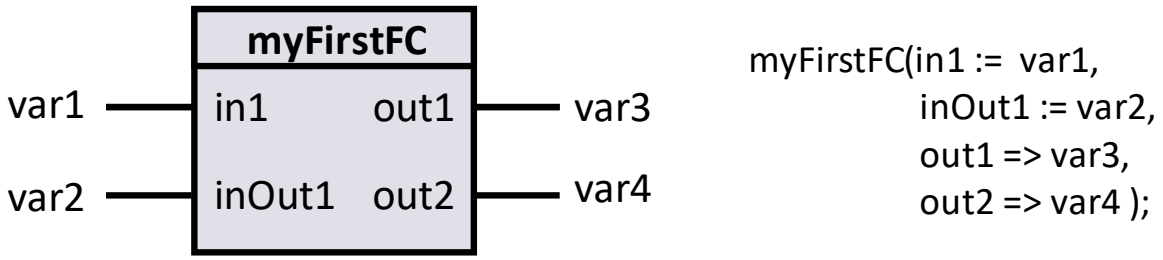


Imagen 24 Sintaxis de llamada a función, con paso de parámetros en SCL

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

Ejemplo

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

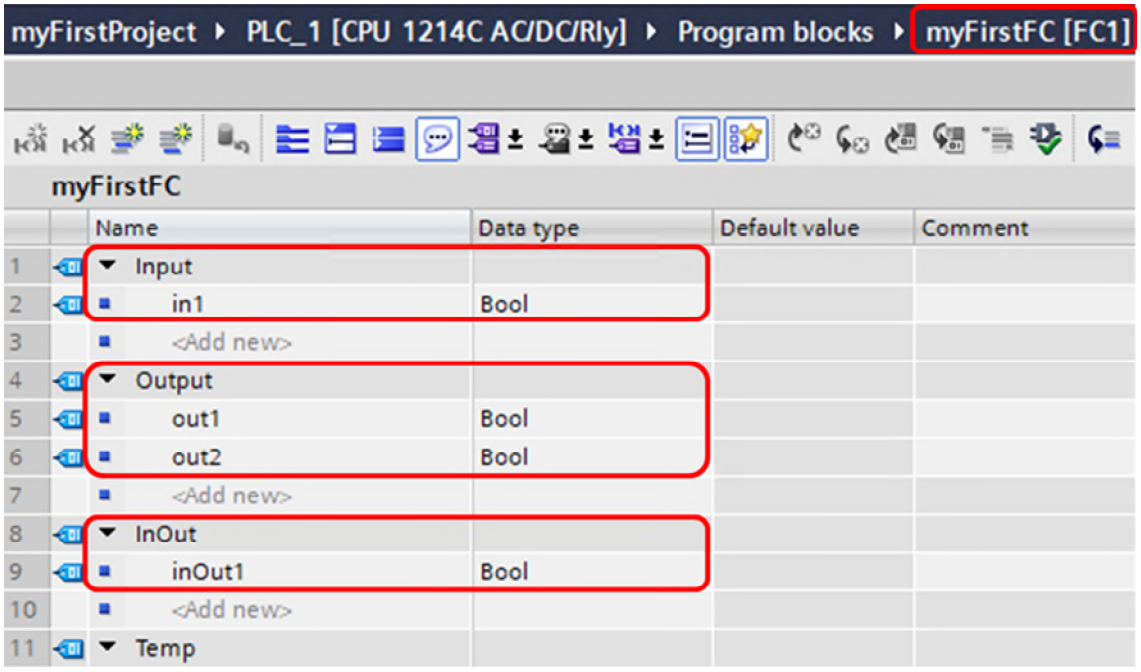
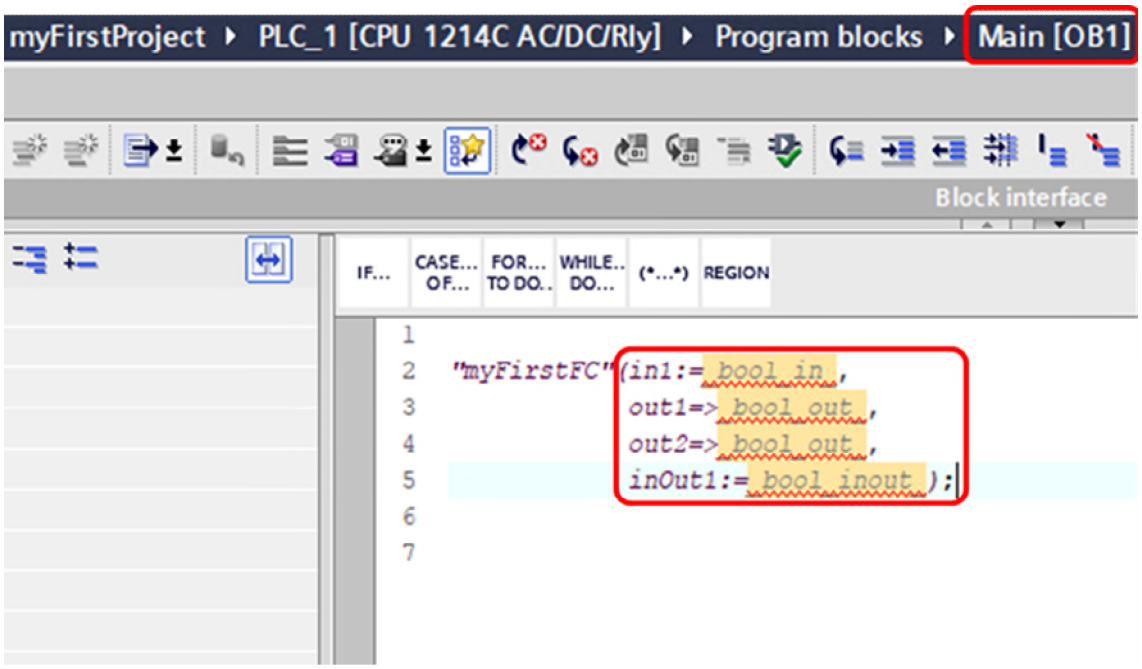


Imagen 25 Interfaz de componentes de una función en el Portal TIA

Después de crear la función "miPrimeraFC", se ha llamado en el "PRINCIPAL" (OB1). La transferencia de parámetros aún no se ha llevado a cabo, y a los parámetros formales que se van a conectar se les asignan por el momento 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 módulo de funciones (FB) en ST / SCL

Para trabajar con el módulo de funciones, lo llamamos en el programa. La principal diferencia entre llamar a un módulo 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 building block se utiliza el nombre de la instancia previamente declarada, seguido de "()" y un punto y coma. Entre paréntesis se pasa el parámetro y el punto y coma cierra la sentencia.

```
instMyFirstFB();
```

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

Transferencia de parámetros

Si la instancia llamada tiene parámetros de interfaz, éstos se muestran. En el caso de los bloques funcionales, el paso de parámetros no siempre es necesario porque la instancia ya tiene asociada una zona de memoria privada.

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

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

Opciones de compra

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

Instanciación múltiple de un módulo de funciones

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 almacenar sus datos durante todo el ciclo.

Ejemplo

Por ejemplo, si el FB contiene el programa de usuario para un cálculo de 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.
Así, todos los datos (información) que pertenecen a este cálculo están disponibles en esta instancia asignada.
Para poder crear una instancia, el FB asignado ya debe existir.
En él 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 módulo de funciones "myFirstFB" se explica ahora paso a paso en el Portal TIA.

1. Creación del módulo de función "miPrimerFB":

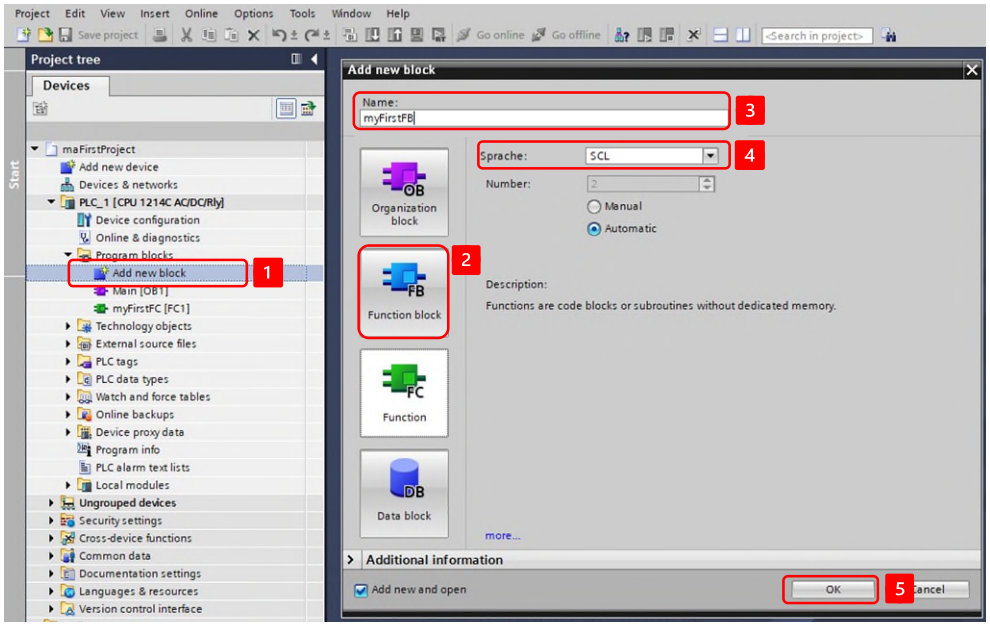


Bild 29 Nuevo Baustein hinzufügen

2. Declaración de la interfaz del módulo e interconexión de las variables en la parte de instrucción del módulo:

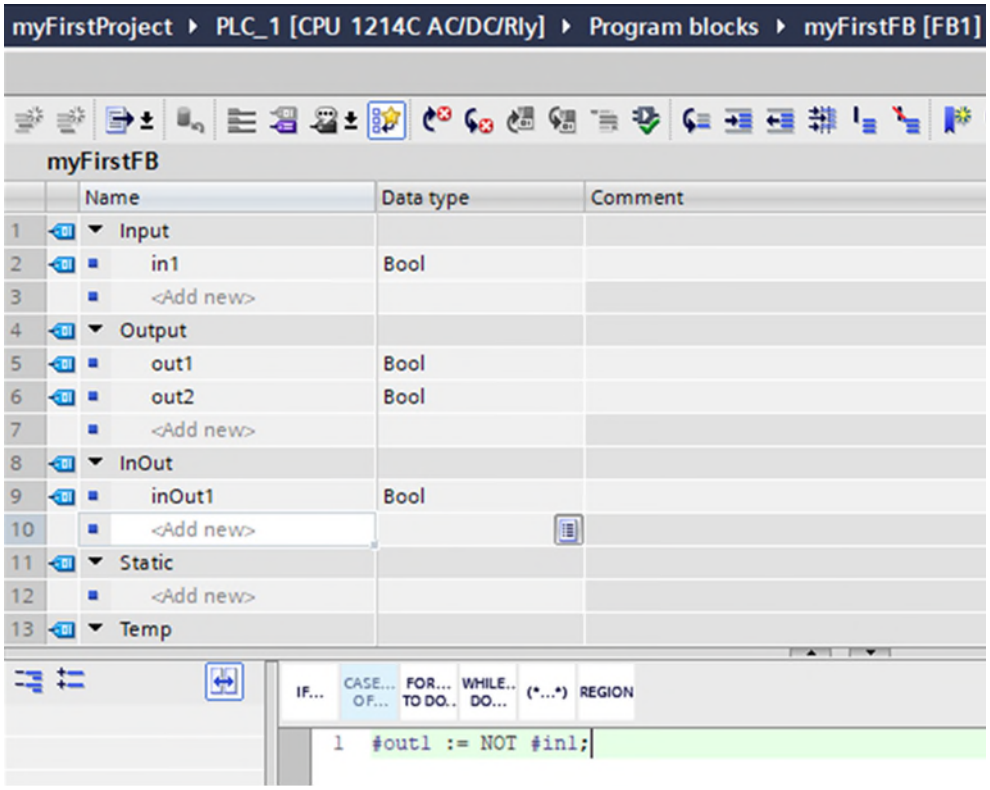
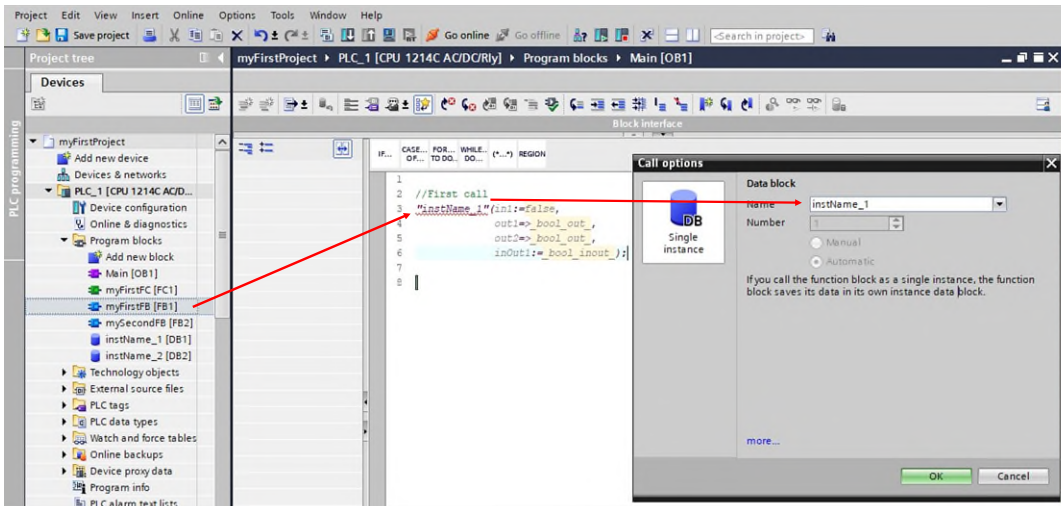


Imagen 30 FB con interfaz de módulo en el Portal TIA

3. 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

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

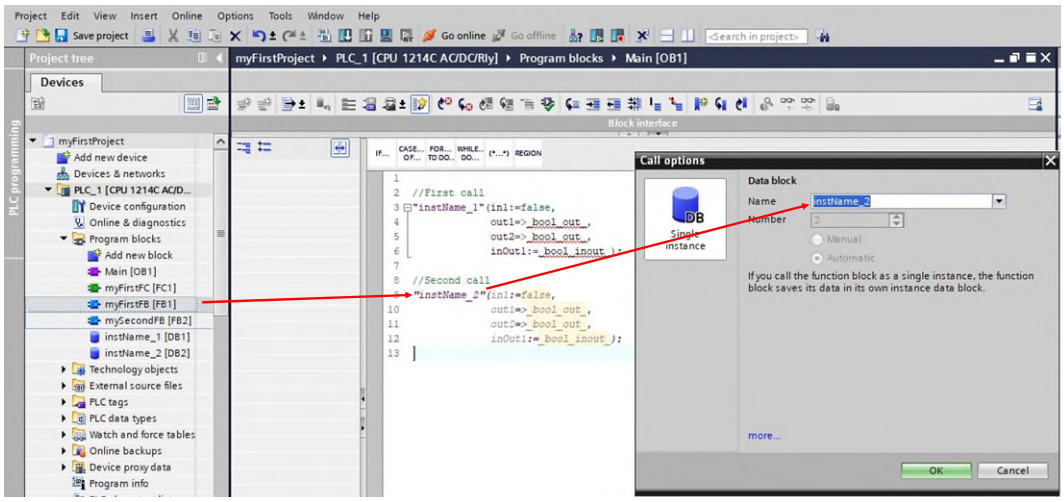
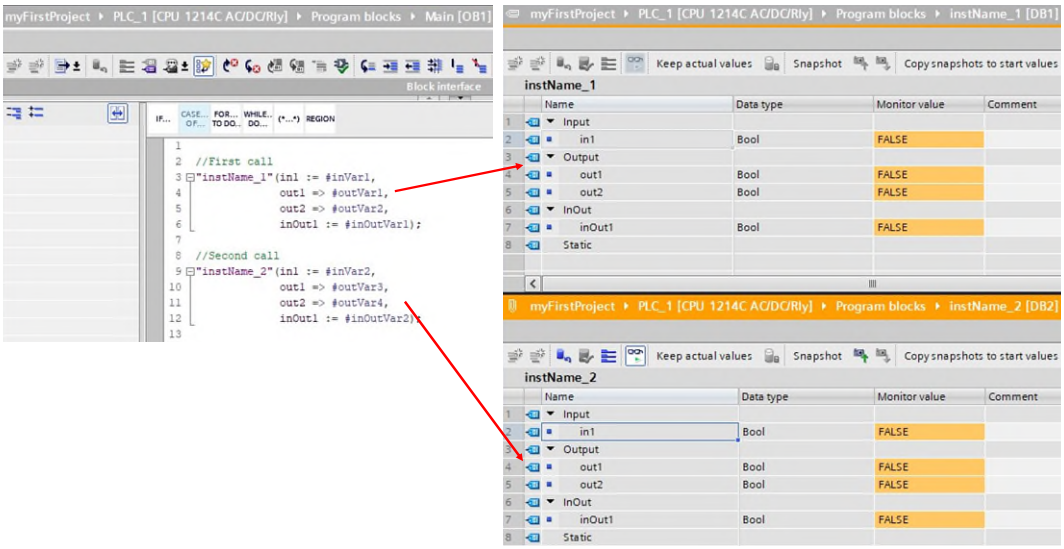


Imagen 32 Instanciación de la segunda llamada de bloque

5. Se pueden observar los dos módulos de datos de instancia:



Fotografía 33 Valores actuales en las instancias

Los módulos de datos de instancia, así como la posibilidad de observarlos y controlarlos, se describirán en detalle en el capítulo siguiente (Módulos de datos).

5.10.2 Opción de llamada como multi-instancia (TIA-Portal)

Si un módulo de funciones se llama en otro módulo de funciones, también se puede seleccionar la multiinstancia en las opciones de llamada.
El uso de multi-instancias permite reducir el número de módulos de datos de instancia. Cuando se crean subrutinas, el uso de multi-instancias es a menudo obligatorio. Si, por ejemplo, desea implementar un contador de tiempo de ejecución en un módulo de control para un motor, cada instancia de motor necesita su propia instancia de contador IEC.
Las multi-instancias se colocan en la interfaz del building block que llama.

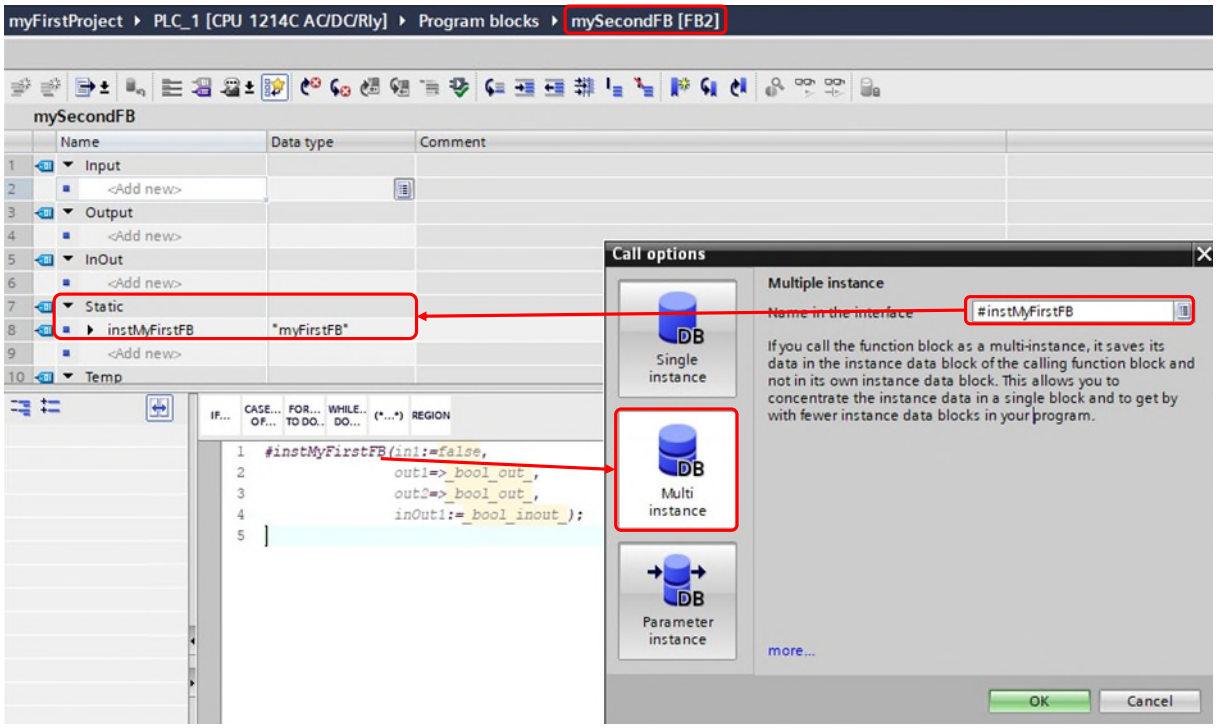


Imagen 34 Opciones de llamada en el Portal TIA

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);

Por 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

Instanciación múltiple de un módulo de funciones

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 almacenar 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