

## 5 Los editores

### 5.1 Aspectos comunes a todos los editores

#### Estructura de un editor

Los editores para todos los lenguajes de programación en IndraLogic constan de una parte de declaración y un tronco. El tronco puede consistir en un editor de texto o gráfico, mientras que la parte de declaración es siempre un editor de texto. El tronco y la parte de declaración están separados por un divisor de pantalla horizontal que se puede desplazar a voluntad, haciendo clic sobre él con el ratón y moviéndolo hacia arriba o hacia abajo manteniendo pulsado el botón del ratón.

#### Márgenes de impresión

Los márgenes de página verticales y horizontales que se aplican al imprimir el contenido del editor están representados por líneas punteadas rojas, en caso de que haberse seleccionado la opción "**Mostrar márgenes de impresión**" en las opciones de proyecto en el diálogo "**Área de trabajo**". Se aplican las preferencias de la impresora configurada, así como el tamaño de la plantilla de impresión seleccionado en el menú "**Archivo**" "**Ajustes documentación**". Si no hay configurada ninguna impresora o plantilla de impresión, se parte de una configuración por defecto (Default.DFR e impresora estándar). Los márgenes de impresión horizontales se representan como si en "Ajustes documentación" se hubieran seleccionado las opciones "Página nueva para cada objeto" o "Página nueva para cada subobjeto". No se muestra el margen inferior.

**Nota:** La representación exacta de los márgenes de impresión sólo está garantizada con un factor de zoom ajustado al 100%.

#### Comentario

Los comentarios del usuario se deben escribir dentro de las secuencias de caracteres especiales "(\*" y "\*)". *Ejemplo:* (\*Esto es un comentario.\*)

Los comentarios están permitidos en todos los editores de texto y en cualquier posición dentro de éstos, es decir, en todas las declaraciones, en los lenguajes AWL y ST y en tipos de datos definidos por el propio usuario. Si se imprime el proyecto utilizando una **plantilla de documento**, en partes del programa basadas en texto aparece detrás de la variable el comentario introducido durante la declaración de variables.

En los editores gráficos FUP y KOP se pueden introducir comentarios para cada red. Para ello, localice la red que desea comentar y active "**Insertar**" "**Comentario**". Además, se pueden añadir comentarios allí donde se introduzcan nombres de variables.

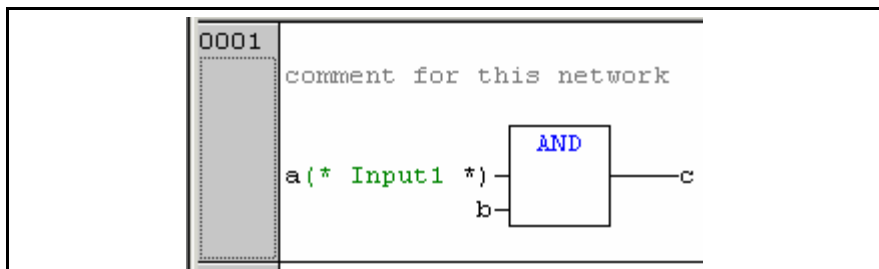


Fig. 5-1 : Ejemplo en FUP de un comentario de red y un comentario detrás de una variable de entrada



En el editor KOP también se puede añadir un comentario a cada contacto o cada bobina, si se han ajustado en consecuencia las opciones de visualización en el menú "Extras" "Opciones".

En el editor CFC existen componentes de comentario especiales que pueden ubicarse a voluntad.

En AS puede introducir un comentario para un paso en el diálogo para la edición de atributos de paso.

También se permiten **comentarios encuadrados**, si está activada la opción correspondiente en el diálogo "**Proyecto**" "**Opciones**" "**Opciones de traducción**".

En el modo online, si mantiene el puntero del ratón durante un breve lapso sobre una variable, se indica en un **tooltip** el tipo y, si procede, la dirección y el comentario de la variable.

## **Zoom al componente llamado**

**Forma abreviada: <Alt>+<Intro>**

Esta orden está disponible en el menú contextual (<F2>) o en el menú Extras, cuando el cursor en los editores de texto se halla sobre el nombre de un componente llamado o si, en los editores gráficos, está marcada la casilla de un componente. El zoom abre el componente en cuestión en su ventana de editor.

Si procede de una biblioteca, se llama el administrador de bibliotecas y se muestra el componente en cuestión.

## **"Extras" "Abrir instancia"**

Esta orden equivale a la orden "Proyecto" "Abrir instancia". Está disponible en el menú contextual o en el menú Extras cuando el cursor se halla sobre el nombre de un bloque de función en los editores de texto o cuando está marcada la casilla de un bloque de función en los editores gráficos.

## **La función "Listar componentes"**

Si en la categoría Editor de las opciones de proyecto está activada la opción "Listar componentes", en todos los editores, en el Administrador watch y de fórmulas, en la visualización y en la configuración de seguimiento está disponible la "función Intellisense".

- Si en lugar de un identificador se introduce un punto ".", se abre una lista de selección de todas las variables locales y globales. A partir de esta lista se puede seleccionar un elemento, que puede insertarse detrás del punto pulsando la tecla Intro. La inserción también funciona después de un doble clic sobre el elemento de la lista.

- Si se introduce como identificador una instancia de bloque de función o una variable definida como estructura seguida de un punto, al introducir el punto se abre una lista de selección de las variables de entrada y salida del bloque de función o de los componentes de la estructura.

Ejemplo:

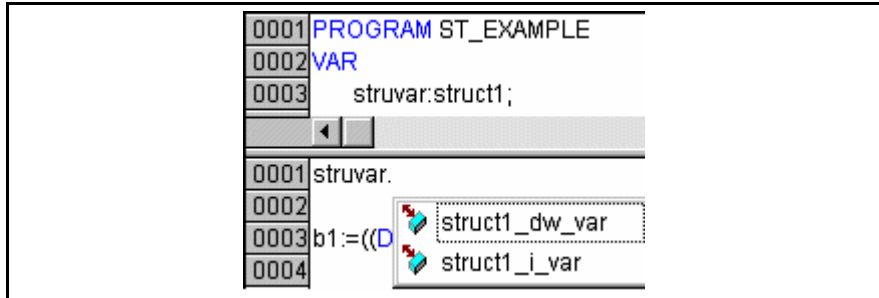


Fig. 5-2: Si se introduce "struvar" se ofrecen los componentes de la estructura struct1

- Si se introduce como identificador una secuencia de caracteres cualquiera y se pulsa <Ctrl> + <barra espaciadora>, aparece una lista de selección de todos los componentes y variables globales disponibles en el proyecto, en la cual la primera que empieza con esta secuencia de caracteres está marcada y puede ser adoptada en el programa pulsando la tecla Intro.

## 5.2 El editor de declaraciones

### Trabajar en el editor de declaraciones

El editor de declaraciones se utiliza durante la declaración de variables de componentes y variables globales, para la declaración de tipos de datos y en el Administrador watch y de fórmulas. Cuenta con las funciones habituales de Windows, y también se pueden utilizar las del ratón IntelliMouse si está instalado el controlador correspondiente.

En el modo Sobrescribir se indica "**SOB**" en negro en la barra de estado, y mediante la tecla <Insert> se puede alternar entre los modos Sobrescribir e Insertar.

La declaración de variables es asistida por **coloración sintáctica**.

Las órdenes más importantes se encuentran en el menú contextual (botón derecho del ratón).

**Nota:** Tenga en cuenta la posibilidad de influir mediante pragmas en las propiedades de una variable durante la traducción o la precompilación (ver capítulo "**Instrucciones pragma en el editor de declaraciones**" a partir de la página 5-13).

#### Parte de declaración

En la parte de declaración de un componente se declaran todas las variables que sólo se utilizan en este componente. Puede tratarse de variables de entrada, variables de salida, variables EntradaSalida, variables remanentes y constantes. La sintaxis de declaración se basa en el estándar de la IEC 61131-3.

Tenga en cuenta la posibilidad de utilizar **plantillas de objetos** para el llenado inicial de la parte de declaración al crear un nuevo objeto del tipo

**5-4 Los editores**

**IndraLogic**

"Variables globales", "Tipo de archivo", "Función", "Componente de función" o "Programa" (ver capítulo 4.3, "Archivo" "Abrir desde plantilla". Además, existe la posibilidad de insertar instrucciones pragma (ver capítulo "Instrucciones pragma en el editor de declaraciones" a partir de la página 5-13).



Fig. 5-3: Ejemplo de una declaración de variable correcta en el editor IndraLogic

**Variables de entrada**

Entre las palabras clave **VAR\_INPUT** y **END\_VAR** se declaran todas las variables que sirven como variables de entrada de un componente, esto es, en la posición de llamada se puede introducir el valor de las variables junto con la llamada.

```
VAR_INPUT
  in1:INT; (* 1. variable de entrada*)
END_VAR
```

Fig. 5-4: Declaración de una variable de entrada

**Ejemplo de acceso a una variable de entrada de un bloque de función:**

El bloque de función FUB tiene una variable de entrada in1 del tipo int

```
PROGRAM prog
VAR
  inst:FUB;
END_VAR
```

Fig. 5-5: Declaración

```
LD 17
ST inst.in1
CAL inst
```

Fig. 5-6: Parte de programa en AWL

```
inst(in1:=17);
```

Fig. 5-7: Parte de programa en ST

### Variables de salida

Entre las palabras clave **VAR\_OUTPUT** y **END\_VAR** se declaran todas las variables que sirven como variables de salida de un componente, es decir, estos valores se devuelven al componente que realiza la llamada, donde pueden ser consultados y utilizados posteriormente.

```
VAR_OUTPUT
  out1:INT; (* 1. variable de salida*)
END_VAR
```

Fig. 5-8 : Declaración de una variable de salida

### Variables de EntradaSalida

Entre las palabras clave **VAR\_IN\_OUT** y **END\_VAR** se declaran todas las variables que sirven como variables de EntradaSalida de un componente.

**Nota:** En estas variables de modifica directamente el valor de la variable transferida ("Transferencia como pointer", Call-by-Reference). Por este motivo, el valor de introducción para una variable de este tipo no puede ser una constante. Por esta razón, las variables VAR\_IN\_OUT de un bloque de función no pueden ser leídas ni escritas desde fuera directamente mediante <Instancia de bloque de función>.<Variable de entrada/salida>

```
VAR_IN_OUT
  inout1:INT; (* 1. variable EntradaSalida *)
END_VAR
```

Fig. 5-9 : Declaración de una variable de EntradaSalida

### Variables locales

Entre las palabras clave **VAR** y **END\_VAR** se declaran todas las variables locales de un componente. Éstas carecen de conexión hacia el exterior, es decir, no es posible escribir en ellas desde fuera.

```
VAR
  loc1:INT; (* 1. variable local*)
END_VAR
```

Fig. 5-10 : Declaración de una variable local

### Variables remanentes

Las variables remanentes pueden conservar su valor más allá del tiempo de ejecución del programa habitual. Entre ellas se cuentan las variables Retain y las variables Persistentes.

```
VAR RETAIN
  rem1:INT; (* 1. variable Remanente*)
END_VAR
```

Fig. 5-11 : Declaración de una variable Retain

- Las variables Retain se identifican con la palabra clave **RETAIN**. Estas variables conservan su valor después de una parada incontrolada, así como después del apagado y encendido normal del control (mediante el comando "Online" "Reset", respectivamente). Al



volver a ejecutar el programa se continúa trabajando con los valores almacenados. Un ejemplo de aplicación sería un contador de unidades en una planta de producción, que debe seguir contando después de un fallo de la corriente. Todas las demás variables se reinician, ya sea con sus valores inicializados o con las inicializaciones estándar.

A diferencia de las variables persistentes, las variables Retain se reinician al volver a descargar el programa.

- Las variables persistentes se identifican con la palabra clave **PERSISTENT**. Al contrario que las variables Retain, conservan su valor sólo después de una nueva descarga ("Online" "Cargar"), pero no después de "Online" "Reset", "Online" "Reset origen" u "Online" "Reset frío" (ver capítulo 4.6), dado que no se guardan en el "área Retain". Por lo tanto, si desea que también las variables persistentes conserven sus valores previos después de un fallo incontrolado del control, es preciso declararlas adicionalmente como VAR RETAIN. Un ejemplo de aplicación de "**variables Retain persistentes**" sería un contador de horas de funcionamiento que debe seguir contando incluso después de un fallo de corriente.

**Nota:** Si una variable local está declarada como RETAIN en un **programa**, precisamente esa variable se guardará en el área Retain (como variable Retain global).

Si una variable local está declarada como RETAIN en un **bloque de función**, la instancia completa de ese bloque de función se guardará en el área Retain (todos los datos del componente), si bien sólo la variable Retain declarada se tratará como tal.

Si una variable local se declara como RETAIN en una **función**, esto no tiene consecuencias. ¡La variable no se guarda en el área Retain! Si una variable local se declara como PERSISTENT en una función, tampoco esto tiene consecuencias.

tras la orden Online	VAR	VAR RETAIN	VAR PERSISTENT	VAR RETAIN PERSISTENT VAR PERSISTENT RETAIN
Reset	-	x	-	x
Reset (frío)	-	-	-	-
Reset (origen)	-	-	-	-
Cargar (=descarga)	-	-	x	x
Cambio online	x	x	x	x

Fig. 5-12 : Comportamiento de remanencia



## Constantes, Typed Literals

Las constantes se identifican con la palabra clave **CONSTANT**. Pueden declararse local o globalmente.

```
VAR CONSTANT o VAR_GLOBAL CONSTANT
  <Identificador>:<Tipo>:= <Inicializacion>;
END_VAR
```

Fig. 5-13 : Sintaxis para la declaración de constantes

```
VAR CONSTANT
  con1:INT:=12; (* 1. constante*)
END_VAR
```

Fig. 5-14 : Declaración de una constante

Encontrará un listado de constantes posibles e información sobre la utilización de constantes tipificadas (Typed Literals) en el Apéndice C: Tipos de datos en IndraLogic.

## Variables externas

Las variables globales que se deben importar en un componente se identifican con la palabra clave **EXTERNAL**. Aparecen online también en la ventana Watch del editor de declaraciones.

Si la declaración en **VAR\_EXTERNAL** no coincide con la declaración global, aparece el siguiente mensaje de error al traducir:

"¡La declaración de "<Nombre>" no coincide con la declaración global!"

Si la variable global no existe, se muestra el siguiente mensaje:

"¡Variable global desconocida: "<variable>!"

```
VAR_EXTERNAL
  var_ext1:INT:=12; (* 1. variable externa *)
END_VAR
```

Fig. 5-15 : Declaración de una variable externa

## Palabras clave

Las palabras clave se deben escribir en mayúsculas en todos los editores. No se deben utilizar las palabras clave como nombres de variables.

## Declaración de variables

Una declaración de variables presenta la siguiente sintaxis:

```
<Identificador> {AT <Dirección>} :<Tipo>{:<Inicialización>};
```

Las partes entre llaves {} son opcionales.

En cuanto al **identificador**, es decir, el nombre de la variable, es preciso recordar que no debe contener espacios ni diéresis, no debe ser declarado por duplicado y no debe ser idéntico a palabras clave. Para las variables no se tienen en cuenta las mayúsculas/minúsculas, de modo que VAR1, Var1 y var1 no son variables distintas. Los guiones bajos son significativos en los identificadores, p. ej. "A\_BCD" y "AB\_CD" se interpretan como identificadores distintos. No se permiten varios guiones bajos seguidos al principio de un identificador o en un identificador. La longitud del identificador, así como su parte significativa, son ilimitadas.

Todas las declaraciones de variables y elementos de tipos de datos pueden contener **inicializaciones** (asignación de un valor inicial). Éstas tienen lugar mediante el operador de asignación " := ". Para variables de

tipos elementales, estas inicializaciones son constantes. La inicialización predeterminada para todas las declaraciones es 0.

```
var1:INT:=12; (* Variable entera con valor inicial 12*)
```

Fig. 5-16 : Inicialización de una variable

Si desea vincular una variable directamente a una dirección determinada, debe declarar la variable con la palabra clave **AT**.

Para la introducción más rápida de declaraciones, utilice el **modo de forma abreviada**.

En bloques de función se pueden especificar variables incluso con datos de dirección incompletos. Para utilizar tales variables en una instancia local, para ello se debe realizar una entrada en la configuración de variables (Recursos).

Tenga en cuenta las posibilidades de la **declaración automática (ver abajo)**, así como de la **utilización de pragmas** para influir en las propiedades de variables durante el proceso de traducción (ver capítulo "Instrucciones pragma en el editor de declaraciones" en la página 5-13).

### Declaración AT

Si desea vincular una variable directamente a una dirección determinada, debe declarar la variable con la palabra clave **AT**. La ventaja de este procedimiento reside en la posibilidad de dar a una dirección un nombre más informativo, y en el hecho de que si fuera preciso modificar una señal de entrada o salida, basta con hacerlo en una sola posición (concretamente en la declaración).

Tenga en cuenta que no es posible el acceso de escritura a variables asignadas a una entrada.

```
interruptor_calefaccion7 AT %QX0.0: BOOL;  
impulso_barrera_de_luz AT %IW2: WORD;  
descarga AT %MX2.2: BOOL;
```

Fig. 5-17 : Ejemplos de declaraciones AT

**Nota:** ¡Si se asignan variables booleanas a una dirección de Byte, Word o DWORD, éstas ocupan 1 byte con TRUE o FALSE, no sólo el primer bit tras el offset!

### "Insertar" "Palabras clave de declaración"

Mediante esta orden puede abrir una lista de todas las palabras clave que pueden utilizarse en la parte de declaración de un componente. Una vez seleccionada una palabra clave y confirmada la selección, se inserta la palabra en la posición actual del cursor.

También obtendrá la lista llamando a la ayuda de entrada (<F2>) y seleccionando la categoría **Declaraciones**.

### "Insertar" "Tipos"

Mediante esta orden se le muestra una selección de los tipos posibles para una declaración de variable. También obtendrá la lista llamando a la ayuda de entrada (<F2>).

Los tipos están divididos en las categorías:

- Tipos estándar BOOL, BYTE, etc.
- Tipos definidos Estructuras, tipos de enumeración, etc.





- Bloques de función estándar para declaraciones de instancia
- Bloques de función definidos para declaraciones de instancia

IndraLogic soporta todos los tipos estándar de la norma IEC 61131-3.

Hallará ejemplos de la utilización de los diversos tipos en el "Apéndice C: Tipos de datos en IndraLogic", página 12-1.

### Coloración sintáctica

En todos los editores contará con ayuda visual durante la implementación y la declaración de variables. Mediante la visualización del texto en color se evitan o se detectan más rápidamente los errores.

Se detecta inmediatamente un comentario no cerrado, y que por tanto comenta instrucciones excluyéndolas; no se escriben incorrectamente palabras clave por descuido, etc.

Color	Significado
Azul	Palabras clave
Verde	Comentarios en los editores de texto
Rosa	Constantes especiales (p. ej. TRUE/FALSE, T#3s, %IX0.0)
Rojo	Entrada errónea (p. ej. constante de tiempo inválida, palabra clave escrita en minúsculas, ...)
Negro	Variables, constantes, operadores de asignación, ...

Fig. 5-18 : Coloración

### Modo de forma abreviada

El editor de declaraciones de IndraLogic le ofrece la posibilidad del modo de forma abreviada. Dicho modo se activa al terminar una línea con <Ctrl><Intro>.

Se admiten las siguientes formas abreviadas:

- Todos los identificadores hasta el último identificador de una línea se convierten en identificadores de variable de la declaración.
- El tipo de declaración viene determinado por el último identificador de la línea. En este contexto, se aplica lo siguiente:  
 B o BOOL arroja BOOL  
 I o INT arroja INT  
 R o REAL arroja REAL  
 S o STRING arroja STRING
- Si no se ha podido establecer ningún tipo mediante estas reglas, el tipo es BOOL y el último identificador no se utiliza como tipo (ejemplo 1).
- Según el tipo de declaración, cada constante se convierte en una inicialización o una longitud de string (ejemplos 2 y 3).
- Una dirección (como en %MD12) se amplía con el atributo AT ... (ejemplo 4).
- Un texto tras un punto y coma (;) se convierte en un comentario (ejemplo 4).
- Se ignoran todos los demás caracteres en la línea (como p. ej. el signo de exclamación en el último ejemplo).

Forma abreviada	Declaración
A	A: BOOL;
A B I 2	A, B: INT := 2;

ST S 2; Un string	ST: STRING(2); (* Un string *)
X %MD12 R 5; Número real	X AT %MD12: REAL := 5.0; (*Número real *)
B !	B: BOOL;

Fig. 5-19 : Ejemplos de declaraciones en el modo de forma abreviada

### Declarar automáticamente

Si se ha elegido la opción **Declarar automáticamente** en la categoría **Editor** en el diálogo "Proyecto" "Opciones", al introducir una variable todavía no declarada aparece en todos los editores un diálogo mediante el cual se puede declarar dicha variable.

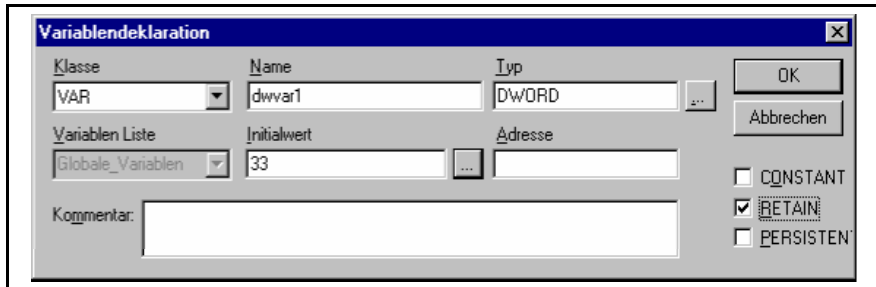


Fig. 5-20 : Diálogo para la declaración de variables

Mediante el campo combinado **Clase**, seleccione si se trata de una variable local (**VAR**) una variable de entrada (**VAR\_INPUT**), una variable de salida (**VAR\_OUTPUT**), una variable de EntradaSalida(**VAR\_IN\_OUT**) o una variable global (**VAR\_GLOBAL**).

Mediante las opciones **CONSTANT**, **RETAIN**, **PERSISTENT** puede definir si se trata de una constante o una variable remanente.

En el campo **Nombre** ya está ocupado con el nombre de variable introducido en el editor, y en el campo **Tipo** se ha introducido BOOL.

Mediante el botón accederá al diálogo de la ayuda de entrada para seleccionar todos los tipos de datos posibles.

### Declaración de **Arrays** (campos):

Si se selecciona **ARRAY** (campo) como tipo de la variable, aparece el diálogo para la introducción de los límites de array.

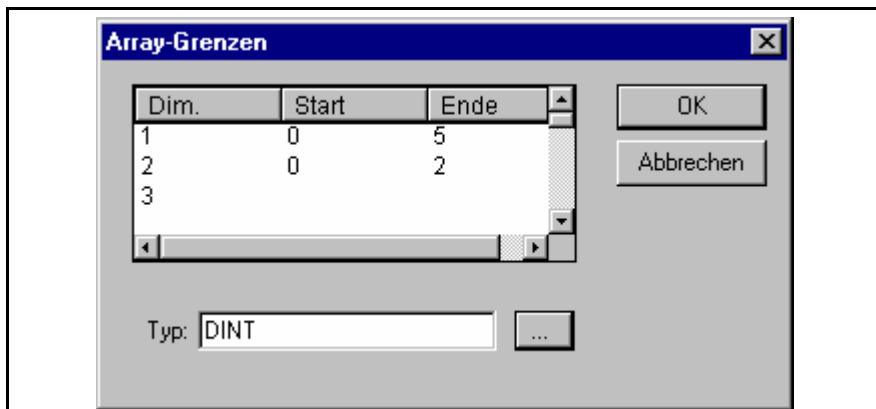



Fig. 5-21 : Editor de declaraciones para arrays

Para cada una de las tres dimensiones posibles (**Dim.**) se pueden introducir en **Inicio** y **Fin** los límites de array (¡intervalo de valores DINT!), haciendo clic con el ratón sobre el campo correspondiente para abrir un marco de edición. En el campo **Tipo** se introduce el tipo de dato del



array. Para ello, mediante el botón  se puede llamar una ayuda de entrada.

Al salir del diálogo de límites de array mediante el botón **OK**, a partir de las entradas se ocupa en el formato IEC el campo "Tipo" en el diálogo Declaración de variables. Ejemplo: ARRAY [1..5, 1..3] OF INT

En el campo **Valor inicial** puede introducir el valor inicial de la variable a declarar. Si es un array o una estructura válida, puede abrir un diálogo de inicialización especial mediante el botón  o bien abrir el diálogo de la ayuda de entrada para otros tipos de variables.

En el diálogo de inicialización para un array se le mostrará un listado de los elementos de array, y haciendo clic con el ratón en la posición detrás de "[:=" puede abrir un campo de edición para introducir el valor inicial de un elemento.

En el diálogo de inicialización para una **estructura** se representan los distintos componentes en la estructura de árbol. En paréntesis detrás del nombre de la variable se encuentran el tipo y el valor inicial predeterminado del componente, seguido de "[:=". Al hacer clic con el ratón sobre el campo detrás de "[:=" se abre un campo de edición en el que puede introducir el valor inicial deseado. Si un componente es un array, mediante un clic con el ratón sobre el signo "más" delante del nombre del array en el diálogo de inicialización se pueden desplegar los distintos campos del array para editarlos con valores iniciales.

Después de salir del diálogo de inicialización mediante **OK** aparece en el campo **Valor inicial** del diálogo de declaración la inicialización del array o de la estructura en formato IEC.

Ejemplo: x:=5,campo:=2,3,estruct2:=(a:=2,b:=3)

En el campo **Dirección** puede vincular a una dirección IEC la variable a declarar (declaración AT).

Si procede, introduzca un **comentario**. El comentario puede dotarse de saltos de línea mediante la combinación de teclas <Ctrl>+<Intro>.

Al pulsar **OK** se cierra el diálogo de declaración y la variable se introduce conforme a la sintaxis IEC en el editor de declaraciones correspondiente.

---

**Nota:** El diálogo para la declaración de variables también se obtiene mediante la orden "Editar" "Declaración" (ver Funciones de edición generales). Si el cursor se encuentra sobre una variable, en el modo Offline se puede abrir mediante <Mayúsculas> <F2> la ventana Autodeclare con los ajustes actuales relacionados con la variable.

---

## Números de línea en el editor de declaraciones

En el modo Offline, un solo clic sobre un número de línea concreto marca la línea de texto entera.

En el modo Online, un solo clic sobre un número de línea concreto permite abrir o cerrar la variable en esta línea, en caso de que se trate de una variable estructurada.

## Declaraciones como tabla

Si está activada la opción **Declaraciones como tabla** en el diálogo de opciones en la categoría **Editor**, accederá al editor de declaraciones en una representación tabular. Al igual que en un archivador, puede seleccionar individualmente las pestañas de los distintos tipos de variables y editar las variables.

5-12 Los editores

IndraLogic

Para cada variable se le mostrarán los siguientes campos de entrada:

<b>Nombre</b>	Introduzca el identificador de la variable.
<b>Dirección</b>	Introduzca, si procede, la dirección de la variable (declaración AT).
<b>Tipo</b>	Introduzca el tipo de la variable. (En caso de instanciación de un bloque de función, introduzca el bloque de función.)
<b>Inicial</b>	Introduzca una posible inicialización de la variable. (correspondiente al operador de asignación " := ").
<b>Comentario</b>	Introduzca aquí un comentario.

Fig. 5-22 : Campos de entrada para una variable en la declaración como tabla. Los dos tipos de representación del editor de declaraciones pueden alternarse sin problemas. En el modo Online no existen diferencias en la representación del editor de declaraciones.

Para editar una nueva variable, ejecute la orden "Insertar" "Nueva declaración".

	Name	Adresse	Typ	Initial	Kommt
0001	AMPEL1		AMPEL		
0002	AMPEL2		AMPEL		
0003	VERZ		WARTEN		
0004	ZAEHLER		INT		

Fig. 5-23 : Editor de declaraciones como tabla

**"Insertar" "Nueva declaración"**

Mediante esta orden se introduce una nueva variable en la tabla de declaración del editor de declaraciones. Si la posición actual del cursor se halla en un campo de la tabla, la nueva variable se inserta antes de esta línea, de lo contrario se inserta al final de la tabla. Además, puede insertar una nueva declaración al final de la tabla, accionando en el último campo de la tabla la tecla de flecha derecha o la tecla de tabulación.

Obtendrá una variable que tiene como valores predeterminados en el campo **Nombre** "Nombre", y en el campo **Tipo** "Bool". Debe cambiar estos valores por los valores deseados. El nombre y el tipo bastan para una declaración de variable completa.

**Editores de declaraciones en el modo Online**

En el modo Online, el editor de declaraciones se convierte en una ventana de monitor. En cada línea se encuentra una variable, seguida de un signo de igualdad (=) y el valor de la variable. Si la variable todavía está indefinida en ese momento, aparecen tres signos de interrogación (???). En el caso de los bloques de función sólo se muestran los valores para instancias abiertas (orden "Proyecto" "Abrir instancia").

Antes de cada variable de varios elementos se halla un signo "más". Pulsando la tecla <Intro> o haciendo doble clic en una de estas variables, ésta se despliega; en el ejemplo estaría desplegada la estructura semáforo1:

```

-AMPEL1
  ...STATUS = 3
  ...GRLEN = FALSE
  ...GEL3 = FALSE
  ...ROT = TRUE
  ...AUS = FALSE
  
```

Fig. 5-24 : Variables en el modo Online

Cuando una variable está desplegada, todos sus componentes están listados a continuación. Delante de la variable aparece un signo "menos". Mediante un nuevo doble clic o pulsando la tecla <Intro>, la variable se pliega y vuelve a aparecer el signo "más".

Pulsando la tecla <Intro> o haciendo doble clic sobre una variable de un solo elemento, se abre el diálogo para escribir una variable (ver "Funciones online generales"). Aquí es posible modificar el valor actual de la variable. En las variables booleanas no aparece ningún diálogo, sino que su valor se conmuta.

El nuevo valor se indica detrás de la variable en color turquesa y entre corchetes y permanece inalterado.	bvar = TRUE < := FALSE > ivar = 509 < := 65 >
Si se ejecuta la orden "Online" "Escribir valores", se ajustan todas las variables a los valores escogidos y se vuelven a mostrar en negro.	bvar = TRUE ivar = 65
Si se ejecuta la orden "Online" "Forzar valores", se ajustan todas las variables a los valores escogidos hasta que se ejecuta la orden "Anular forzar". En este caso, el color del valor forzado cambia a rojo.	bvar = FALSE ivar = 4

Fig. 5-25 : Asignaciones de valor en el modo Online

## Instrucciones pragma en el editor de declaraciones

La instrucción pragma sirve para controlar el proceso de traducción. Se encuentra con texto adicional en una línea del programa o en una línea propia del editor de declaraciones.

La instrucción pragma se encierra entre corchetes (no se tiene en cuenta la escritura en mayúsculas o minúsculas):

```
{ <Texto de la instrucción> }
```

En caso de que el compilador no pueda interpretar con sentido el texto de la instrucción, el pragma en su conjunto se trata como un comentario y se omite. Sin embargo, se muestra un aviso: "Ignorar directiva de compilador "< Texto de la instrucción>!".

Dependiendo del tipo y el contenido del pragma, un pragma actúa sobre la línea en la que se encuentra o, si se da el caso, sobre todas las líneas siguientes, hasta que es anulado por el pragma correspondiente o hasta que se ejecuta el mismo pragma con otros parámetros o se alcanza el final del archivo. En este contexto se entiende por archivo: parte de declaración, parte de implementación, lista de variables globales, declaración de tipo.

El corchete de apertura puede seguir inmediatamente a un nombre de variable. Los corchetes de apertura y de cierre deben hallarse en la misma línea.

Actualmente, en IndraLogic se pueden utilizar los siguientes pragmas:

- Pragma {flag} para inicialización, monitorización, creación de símbolos
- Pragma {bitaccess...} para acceso de bit



- Pragma {parameter..}, {template...}, {instance...} para la creación de entradas del administrador de parámetros
- Pragma {library ...} para mostrar/ocultar partes de declaración de una biblioteca en el administrador de bibliotecas

### Pragma para inicialización, monitorización, creación de símbolos, acceso de bit

**Pragma {flag} para inicialización, monitorización, creación de símbolos:**

**Sintaxis:** {flag [<flags>] [off/on]}

Mediante este pragma se puede influir en las propiedades de una declaración de variable:

noinit:	No se inicializa la variable.
nowatch:	No se monitoriza la variable
noread:	La variable se exporta sin derecho de lectura al archivo de símbolos
nowrite:	La variable se exporta sin derecho de escritura al archivo de símbolos
noread, nowrite:	La variable no se exporta al archivo de símbolos

Fig. 5-26 : <flags> puede ser una combinación de estos flags

Mediante la modificación "on", el pragma actúa sobre todas las declaraciones de variables subsiguientes, hasta que es anulado por el pragma {flag **off**}, o hasta que es sobrescrito por otro pragma {flag <flags> on}.

Sin la modificación con "on" u "off", el pragma actúa únicamente sobre la declaración de variable actual (esto es, la declaración que se cierra con el siguiente punto y coma).

#### Ejemplos de utilización del pragma {flag}:

##### Inicialización y monitorización de variables:

```

VAR
  a :INT {flag noinit, nowatch};
  b : INT {flag noinit };
END_VAR

VAR
  {flag noinit, nowatch on}
  a : INT;
  {flag noinit on}
  b : INT;
  {flag off}
END_VAR

```

Fig. 5-27 : No se inicializa ni monitoriza la variable a. No se inicializa la variable b.

```
{flag noinit on}
VAR
  a : INT;
  b : INT;
END_VAR

{flag off}
VAR
{flag noinit on}
  a : INT;
  b : INT;
{flag off}
END_VAR
```

Fig. 5-28 : No se inicializa ninguna de ambas variables

Exportación de variables al archivo de símbolos:

```
VAR
  a : INT {flag noread};
  b : INT {flag noread, nowrite};
END_VAR

VAR
  { flag noread on}
  a : INT;
  { flag noread, nowrite on}
  b : INT;
  {flag off}
END_VAR
```

Fig. 5-29 : Si se dota al componente de derecho de lectura y escritura, mediante los siguientes pragmas se puede exportar la variable a sólo con derecho de escritura, mientras que no se puede exportar en absoluto la variable b

```
{ flag noread, nowrite on }
VAR
  a :INT;
  b : INT;
END_VAR
{flag off}

VAR
  { flag noread, nowrite on }
  a : INT;
  b : INT;
  {flag off}
END_VAR
```

Fig. 5-30 : Ninguna de ambas variables a y b se exporta al **archivo de símbolos**

El pragma actúa aditivamente sobre todas las declaraciones de variables subordinadas

Ejemplo: (todos los componentes utilizados se exportan con derecho de lectura y escritura)

```
a : afb;

...

FUNCTION_BLOCK afb
VAR
  b : bfb {flag nowrite};
  c : INT;
END_VAR

...

FUNCTION_BLOCK bfb
VAR
  d : INT {flag noread};
  e : INT {flag nowrite};
END_VAR
```

Fig. 5-31 : Ejemplo: El pragma actúa aditivamente

"a.b.d": No se exporta.

"a.b.e": Se exporta sólo con derecho de lectura.

"a.c": Se exporta con derecho de lectura y escritura.

**Pragma {bitaccess...} para el acceso de bit:**

Mediante este pragma se pueden definir accesos de bit simbólicos válidos a **estructuras** que tienen lugar con ayuda de una constante global. Estos símbolos se ofrecen entonces tanto en la ayuda de entrada como en la "función Intellisense" y se utilizan para la representación de los accesos de bit durante la monitorización en el editor de declaraciones. Allí se muestran entonces también las constantes globales utilizadas.

---

**Nota:** ¡Debe estar activada la opción de proyecto "Sustituir constantes" (categoría Opciones de traducción, ver capítulo 4.2)!

---

Se debe insertar el pragma en la definición de la estructura en una línea aparte. La línea no se cierra mediante un punto y coma.

**Sintaxis:** {bitaccess <Constante global> <Número de bit> '<Comentario>'}  
<Constante global>: Nombre de la constante global que debe estar definida en una lista de variables globales.

<Número de bit>: Valor de la constante global tal como está definida en la lista de variables globales.

Ver un ejemplo en el "Apéndice B: Operandos en IndraLogic", Operandos en IndraLogic, Direccionamiento de bits en variables.



## Pragmas para la creación de entradas en el Administrador de parámetros

Mediante pragmas dentro de declaraciones de variables se pueden crear automáticamente entradas en listas de parámetros que se gestionan en el Administrador de parámetros. El Administrador de parámetros está disponible en el sistema de programación en función del sistema de destino, esto es, debe estar activado en los ajustes del sistema de destino (funciones de red).

### Generalidades sobre la sintaxis:

- Un pragma se encierra entre llaves. No se tiene en cuenta la escritura en mayúsculas o minúsculas. Si se inserta en declaraciones de variables "normales", debe situarse delante del punto y coma final de la declaración de variable sobre la que debe actuar.
- ¡Los pragmas que se utilizan en ventanas VAR\_CONFIG se encuentran cada uno en una única línea y no se cierran con un punto y coma!
- <name>: nombre de la lista de parámetros en el Administrador de parámetros. Si todavía no existe la lista de variables, se creará automáticamente.
- <key>: nombre del atributo, esto es, título de la columna en la lista de parámetros; p. ej. "Name", "Value", "Accesslevel" etc.; las keys que pueden introducirse dependen de la definición del tipo de lista de parámetros específica del sistema de destino. Todas las definiciones de key se encuentran en el pragma una tras otra, separadas por espacios, entre corchetes. Tenga en cuenta la sintaxis para las entradas en listas de instancias para componentes de array, de estructura o de bloque de función (ver 3.).
- <value>: valor que debe introducirse en la lista para el atributo definido mediante <key>. Para ello, tenga en cuenta que los valores que contienen espacios deben encerrarse entre comillas dobles. Ejemplo: ...accesslevel="read only"....

---

**Nota:** Las instrucciones de pragma actúan ya durante un cambio de foco (Precompilar), es decir, al salir del editor de declaraciones. Las entradas de pragma defectuosas no se comunican hasta la traducción del proyecto.

---

Se pueden generar las siguientes entradas:

### 1. Entradas en listas de parámetros del tipo "Lista de variables"

#### (a) desde la parte de declaración de programas y listas de variables globales

Para una variable dentro de una declaración PROGRAM o VAR\_GLOBAL se puede generar una entrada en una lista de parámetros del tipo "Variables" si se declara de la siguiente forma: (Si todavía no existe la lista de parámetros, se creará una).

**Sintaxis:** {parameter list=<name> [ <key>=<value> <key>=<value> ... keys adicionales ] }

**Ejemplo:** En un programa se declara la variable bvar, la cual debe introducirse en la lista de parámetros parlist1 del tipo Lista de variables con el nombre bvar1, el valor 102, el índice 16#1200 y el subíndice 16#2.

```
VAR
bvar:INT{parameter list=parlist1 [name=bvar1 value=102
index=16#1200 subindex=16#1 ] };
END_VAR
```

Fig. 5-32 : Ejemplo de lista de variables

(b) mediante una declaración en la interfaz VAR\_CONFIG:

Para variables se puede generar una entrada en una lista de parámetros del tipo "Variables" si se declara de la siguiente forma en una ventana VAR\_CONFIG: (Si todavía no existe la lista de parámetros, se creará una).

**Sintaxis:** {parameter list=<name> path=<path> [ <key>=<value> <key>=<value> ... keys adicionales ] }

<path> Ruta de la variable para la que se debe generar la entrada, p. ej. "PLC\_PRG.act1.var\_x"

**Ejemplo:** para la variable var\_x se genera una entrada en la lista de parámetros varlist1, y como nombre simbólico se introduce xvar.

```
VAR
VAR_CONFIG
{parameter list=varlist1 path=PLC_PRG.act1.var_x [
name=xvar ] }
END_VAR
```

Fig. 5-33 : Ejemplo de declaración de una entrada en la lista de parámetros

**2. Entradas en listas de parámetros del tipo "Plantilla" desde bloques de función y estructuras**

En declaraciones de variables en bloques de función y estructuras se pueden generar entradas en listas de parámetros del tipo "Plantilla". (Si todavía no existe la plantilla, se creará una).

**Sintaxis:** {template list=<name> [ <key>=<value> <key>=<value> ... keys adicionales ] }

**Ejemplo:** la variable strvar, que es un elemento de la estructura stru1, debe ser introducida con el nombre (member) "struvar1" y el Accesslevel=low en la plantilla "vor11" en el Administrador de parámetros:

```
TYPE stru :
STRUCT
ivar:INT;
strvar:STRING{template list=vor11 [member=struvar1
accesslevel=low] };
END_STRUCT
END_TYPE
```

Fig. 5-34 : Ejemplo de lista de parámetros del tipo "Plantilla"





### 3. Entradas en listas de parámetros del tipo "Instancia" (para variables de array, de bloques de función o de estructuras)

#### (a) desde programas y listas de variables globales

En la declaración de variables de array, de bloques de función o de estructuras dentro de un programa o de una lista de variables globales, se puede generar directamente una lista de instancias en el Administrador de parámetros.

**Sintaxis:**{instance list=<name> template=<template> baseindex=<index> basesubindex=<subindex> [ <key>=<value para el primer elemento > <key>=<value para el primer elemento > ... keys adicionales para el primer elemento] | [ <key>=<value para el segundo elemento > <key>=<value para el segundo elemento > ... keys adicionales para el segundo elemento] | [keys para elementos adicionales]}

Para arrays se define la key "template" con la plantilla siempre disponible implícitamente "ARRAY"; para estructuras y bloques de función debe existir la plantilla correspondiente en el Administrador de parámetros y ser introducida aquí.

Para cada elemento individual de array y estructura o bloque de función se puede predefinir una entrada individual en la lista de parámetros: Por ejemplo, por cada elemento se puede introducir una definición propia [name=<elementname>]. Las definiciones de key de los elementos individuales (¡por cada elemento dentro de los mismos corchetes!) se introducen en sucesión separadas por espacios y se refieren automáticamente a los elementos en el orden ascendente del índice (Member). ¡Si no existen tantas definiciones de key como elementos o variables contiene el array, estructura o bloque de función, los elementos remanentes reciben los mismos valores que el último elemento definido individualmente (excepción para la key "name" en arrays, ver abajo) ! (Ver abajo, ejemplo 1b).

#### Automatismos para la introducción de arrays en listas de parámetros en relación con la key "name":

- Si no predefine en el pragma ningún nombre para un elemento de array, el elemento y todos los elementos siguientes en la lista de parámetros reciben automáticamente el nombre <nombre de componente>\_<nombre de variable de array>\_<índice de arrays correspondiente>.

Ejemplo: Se debe introducir en una lista de parámetros la variable de array ARRVAR [1..8] of INT en el componente PLC\_PRG mediante pragma. Si no se introduce ninguna definición para la key "name", los elementos de array individuales en la lista de parámetros reciben automáticamente los nombres "PLC\_PRG\_arrvar\_1" a "PLC\_PRG\_arrvar\_8".

- Si predefine para el primer elemento de array un nombre cualquiera "<nombre>\_<primer índice del área de array>" en el pragma, los demás elementos de array en la lista de parámetros reciben automáticamente el nombre "<nombre>\_<índice correspondiente>".

Ejemplo: Para la variable de array ARRVAR [1..8] se predefine en el pragma para el primer elemento "[name=xyz\_1]" -> en la lista de parámetros aparecen los nombres xyz\_1 a xyz\_8.

---

**Nota:** Si en el caso de variables de array no especifica ningún valor para la key "Member", éste se genera automáticamente para cada elemento de array del índice de arrays.

---

#### Ejemplos:

Ejemplo1a: una variable de array arr\_1 se declara de la siguiente manera para que en el Administrador de parámetros se cree una lista de

instancias "arrinst" (en caso de que aún no exista!), en la que se introducirán los elementos del array, de tal forma que cada elemento sea introducido en principio con el nombre simbólico xname\_<Index> (editable en la lista) y el subíndice se incremente en 1 por cada entrada partiendo de 0 (basesubindex). Accesslevel=low se adopta para todos los elementos.

```
arr_1: ARRAY [1..8] OF INT{instance list=arrinst
template=ARRAY baseindex=16#0 basesubindex=16#0
[name=xname_1]};
```

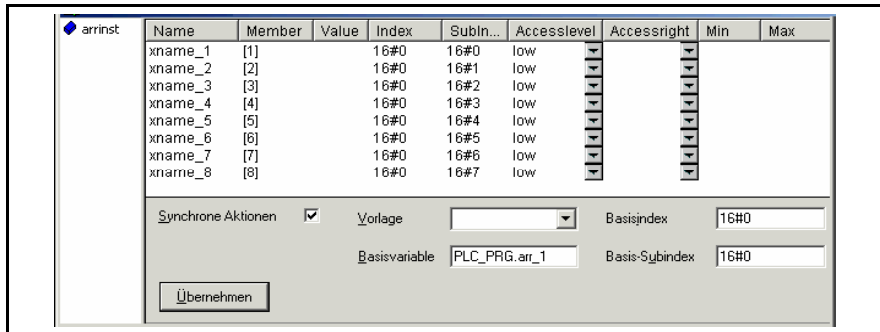


Fig. 5-35 : Editor del Administrador de parámetros para el ejemplo 1a, introducir array en lista de instancias

Ejemplo1b: para una variable de array arr\_1 se predefinen en la lista de parámetros ya nombres distintos sólo para los elementos 1 a 4, así que los elementos 5 a 8 reciben el nombre del elemento 4, al que se añaden un guión bajo y el índice correspondiente, esto es, xname\_5 a xname\_8. Tenga en cuenta que debe introducir definiciones de key adicionales para un elemento determinado dentro de las mismas corchetes, tal como se muestra aquí para el primer y cuarto elementos en relación con el Accesslevel:

```
arr_1: ARRAY [1..8] OF INT{instance list=arrinst
template=ARRAY baseindex=16#0
basesubindex=16#0 [name=aname accesslevel=high]
[name=bname] [name=cname] [name=xname
accesslevel=medium]};
```

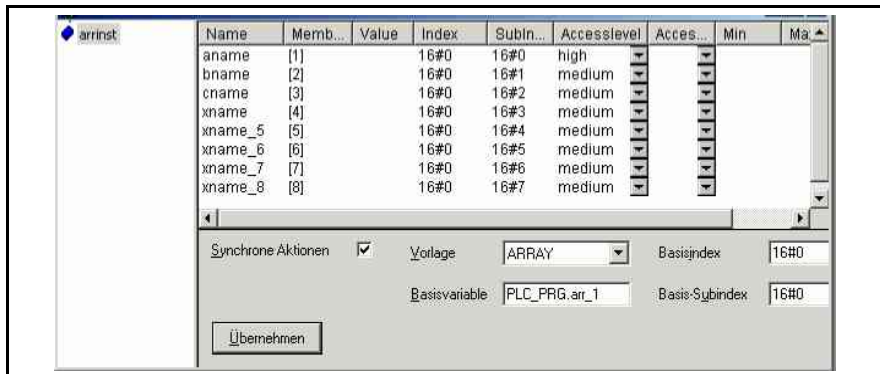


Fig. 5-36 : Editor del Administrador de parámetros para el ejemplo 1b, introducir array en lista de instancias

Ejemplo 2: una variable de estructura struvar del tipo stru1 se declara de la siguiente manera para que en el Administrador de parámetros se cree una lista de instancias "strulist " (en caso de que aún no exista) basada en la plantilla strulist\_temp y en la que aparecerán como entradas las variables a,b,c de la estructura ya existente stru1. Cada variable no recibe todavía un nombre simbólico al ser introducida, el Accesslevel se

ajusta en High y al índice definido por la plantilla se le suma 2. Asegúrese de que la plantilla de instancia especificada exista en el Administrador de parámetros:

```
struvar:stru1{instance list=strulist
template=strulist_templ baseindex=16#2 basesubindex=16#0
[accesslevel=high] };
```

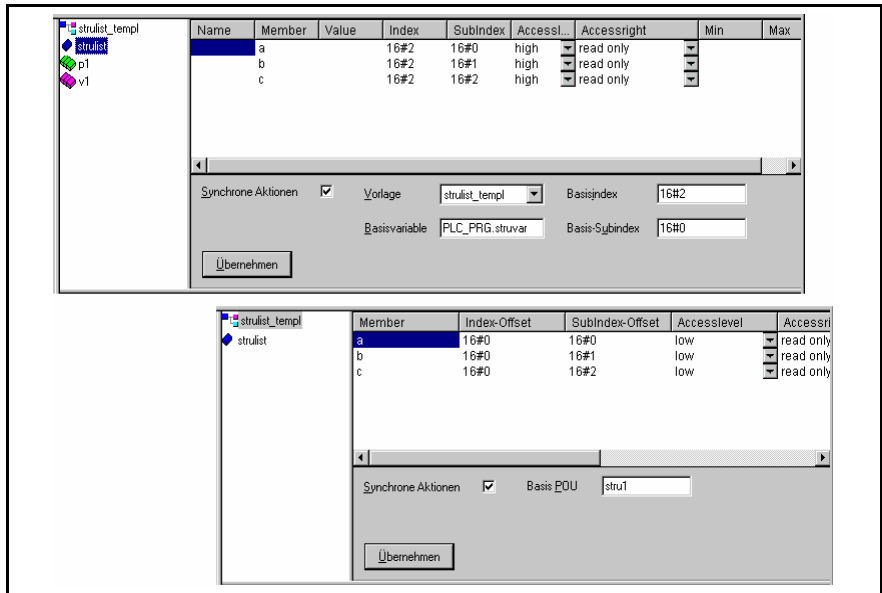


Fig. 5-37 : Introducir variable de estructura en lista de instancias

**(b) mediante una declaración en la interfaz VAR\_CONFIG**

Para variables instanciables se pueden definir entradas en una lista de instancias en el Administrador de parámetros directamente mediante una declaración en una ventana VAR\_CONFIG. ¡Esta declaración es independiente de eventuales configuraciones de variables! (Si todavía no existe la lista de instancias, se creará una.)

Asegúrese de que la plantilla especificada (<template>) exista en el Administrador de parámetros.

**Syntaxis:**{instance list=<name> path=<path> template=<template> baseindex=<index> basesubindex=<subindex>[ <key>=<value> <key>=<value> ... keys adicionales ] }

<path>: la ruta de instancia de la variable; p. ej. "PLC\_PRG.fb1inst", donde fb1inst es una instancia del bloque de función fb1.

**Ejemplo:** con la siguiente entrada en una ventana VAR\_CONFIG (¡independientemente de eventuales configuraciones de variables!) se crean en una lista de instancias "varinst1" entradas para todas las variables del bloque de función fb1 sobre la base de la plantilla "fb1\_templ" (la cual ya debe existir). Para cada entrada se suma 2 al Index-Offset, predefinido por la plantilla (baseindex), mientras que no se suma nada al Subindex-Offset (basesubindex). Cada entrada recibe un nombre simbólico "fb1var" que todavía debe ser editado en la lista.

```
VAR_CONFIG
{instance list=varinst1 path=PLC_PRG.fb1
template=fb1_templ baseindex=16#2 basesubindex=16#0 [
name=fb1var ]}
END_VAR
```



## Pragma para mostrar/ocultar partes de declaración en el administrador de bibliotecas

Mediante los pragmas {library public} y {library private} se puede definir, en una biblioteca creada en IndraLogic, qué líneas/partes de línea de la parte de declaración se mostrarán o no posteriormente en el administrador de bibliotecas durante la utilización de la biblioteca en un proyecto. La visualización de la parte de implementación no se ve afectada por esto.

De este modo, por ejemplo, se pueden hacer invisibles para el usuario comentarios o determinadas declaraciones de variable de la biblioteca. Los pragmas se aplican para el resto de la siguiente línea o para las líneas siguientes, hasta que sean anulados por el otro pragma.

**Sintaxis:** {library public} *El siguiente texto se muestra en el administrador de bibliotecas.*  
{library private} *El siguiente texto no se muestra.*

**Ejemplo:** observe abajo la parte de declaración de una biblioteca creada en IndraLogic. El comentario "(\* this is for all \*)" debe mostrarse tras la integración de la biblioteca en el administrador de bibliotecas, mientras que el comentario "(\* but this is not for all \*)" no debe mostrarse. Las variables local e in2 tampoco deben ser visibles:

```
{library public}(*this is for all*){library private}(*this
is not for all*)

{library public}
FUNCTION afun : BOOL

VAR_INPUT
  in: BOOL;
END_VAR

{library private}

VAR
  local: BOOL;
END_VAR

{library public}

VAR_INPUT
  in2: BOOL;
{library private}
  in3: BOOL;
{library public}
END_VAR
```



## 5.3 Editores de los lenguajes de programación textuales

### Trabajar en los editores de texto

Los editores de texto utilizados para la parte de implementación (el editor de Listas de instrucciones y el editor para Texto estructurado) de IndraLogic cuentan con las funciones habituales de los editores de texto de Windows.

La implementación en los editores de texto se facilita mediante coloración sintáctica.

Si está trabajando en el modo de sobrescritura, en la barra de estado se indica en negro "**SOB**". Pulsando la tecla <Insert> puede alternar entre los modos de sobrescritura y de inserción.

Las órdenes más importantes se encuentran en el menú contextual (botón derecho del ratón).

#### "Insertar" "Operador" en editores de texto

Mediante esta orden se muestran en un diálogo todos los operadores que están disponibles en el lenguaje actual.

Si se selecciona uno de los operadores y se cierra la lista con **OK**, el operador marcado se inserta en la posición actual del cursor. (La manipulación es la misma que en la ayuda de entrada.)

#### "Insertar" "Operando" en editores de texto

Mediante esta orden se muestran en un diálogo todas las variables que están disponibles como operandos. Puede elegir si desea visualizar una lista de las variables globales, las locales o las del sistema.

Si se selecciona uno de los operandos y se cierra el diálogo con **OK**, el operando marcado se inserta en la posición actual del cursor. (La manipulación es la misma que en la ayuda de entrada.)

#### "Insertar" "Función" en editores de texto

Mediante esta orden se muestran en un diálogo todas las funciones. Puede elegir si desea visualizar una lista de las funciones definidas por el usuario o de las funciones estándar.

Si se selecciona una de las funciones y se cierra el diálogo con **OK**, la función marcada se inserta en la posición actual del cursor. (La manipulación es la misma que en la ayuda de entrada.)

Si se ha seleccionado en el diálogo la opción **Con argumentos**, se insertan también las variables de entrada de la función necesarias.

#### "Insertar" "Bloque de función" en editores de texto

Mediante esta orden se muestran en un diálogo todos los bloques de función. Puede elegir si desea visualizar una lista de los bloques de función definidos por el usuario o de los bloques de función estándar.

Si se selecciona uno de los bloques de función y se cierra el diálogo con **OK**, el bloque de función marcado se inserta en la posición actual del cursor. (La manipulación es la misma que en la ayuda de entrada.)

Si se ha seleccionado en el diálogo la opción **Con argumentos**, se insertan también las variables de entrada del bloque de función necesarias. Sin embargo, no es imprescindible definir dichas variables.



### Llamada de componente con parámetros de salida en editores de texto

En los lenguajes textuales AWL y ST se pueden asignar directamente ya en la llamada los parámetros de salida de un componente llamado.

Ejemplo: el parámetro de salida out1 de afbinst se asigna a la variable a.

AWL: CAL afbinst(in1:=1, out1=>a)

ST: afbinst(in1:=1, out1=>a);

Si se inserta el componente utilizando la ayuda de entrada (<F2>) con la opción **Con argumentos** en la ventana de implementación de un componente ST o AWL, se representa automáticamente en esta sintaxis con sus parámetros. Pero no se deben asignar necesariamente los parámetros.

### Los editores de texto en el modo Online

Las funciones online en los editores son Insertar breakpoint y Procesamiento por pasos individuales. Junto con la monitorización, el usuario dispone así de la capacidad de depuración de un moderno depurador de lenguaje estándar de Windows.

En el modo Online, la ventana del editor de texto está dividida en dos verticalmente. En el lado izquierdo de la ventana se encuentra el texto de programa normal, y en el lado derecho se muestran las variables cuyos valores se están cambiando en la línea actual.

La representación es la misma que en la parte de declaración, esto es, cuando el control está en funcionamiento, se muestran los valores actuales de las variables correspondientes.

Durante la monitorización de expresiones o variables direccionadas por bit se debe tener en cuenta lo siguiente: En las expresiones se indica siempre el valor de la expresión en su conjunto. Ejemplo: a AND b se representa como azul o con ":=TRUE", si a y b son TRUE. En caso de variables direccionadas por bit se monitoriza siempre el valor de bit diseccionado (p. ej. a.3 se representa en azul o con :=TRUE si a tiene el valor 4). Si mantiene el puntero del ratón brevemente sobre una variable, se muestran en un tooltip (información de herramientas) el tipo, la dirección y el comentario de la variable.

### "Extras" "Ajustes de monitorización"

Esta orden le permite configurar su ventana de monitorización. En los editores de texto, durante la monitorización se divide la ventana en una mitad izquierda en la que se encuentra el programa, y una mitad derecha en la que se monitorizan todas las variables presentes en la línea de programa en cuestión.

Puede ajustar la **anchura** que debe adoptar el área de monitorización en la ventana de texto, así como la **distancia** que deben presentar dos variables de monitorización en una línea. La especificación de distancia 1 corresponde a una altura de línea en la fuente de escritura seleccionada.

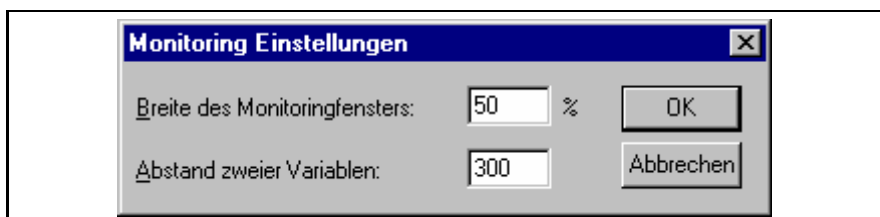


Fig. 5-38 : Diálogo de ajustes de monitorización

### Posiciones de breakpoint en el editor de texto

Dado que en IndraLogic se agrupan internamente varias líneas AWL en una línea de Code C, no se pueden insertar breakpoints en cada línea. Las posiciones de breakpoint son todos los puntos del programa en los que pueden variar valores de variable o en los que se ramifica el flujo del programa (excepción: llamadas de función. En este caso, si fuera preciso, se debe insertar un breakpoint en la función). Tampoco tiene sentido insertar un breakpoint en las posiciones intermedias, dado que no puede haber cambiado nada en los datos desde la posición de breakpoint anterior.

De este modo, en AWL se dan las siguientes posiciones de breakpoint:

- Al principio del componente
- En cada LD, LDN (o, en caso de que un LD se halle directamente detrás de una marca, en ésta)
- En cada JMP, JMPC, JMPCN
- En cada marca
- En cada CAL, CALC, CALCN
- En cada RET, RETC, RETCN
- Al final del componente

Para Textos estructurados se dan las siguientes posiciones de breakpoint:

- En cada asignación
- En cada instrucción de RETURN y EXIT
- En líneas en las que se evalúan condiciones (WHILE, IF, REPEAT)
- Al final del componente

Las posiciones de breakpoint se caracterizan por el hecho de que el campo de número de línea se representa en un gris más oscuro.

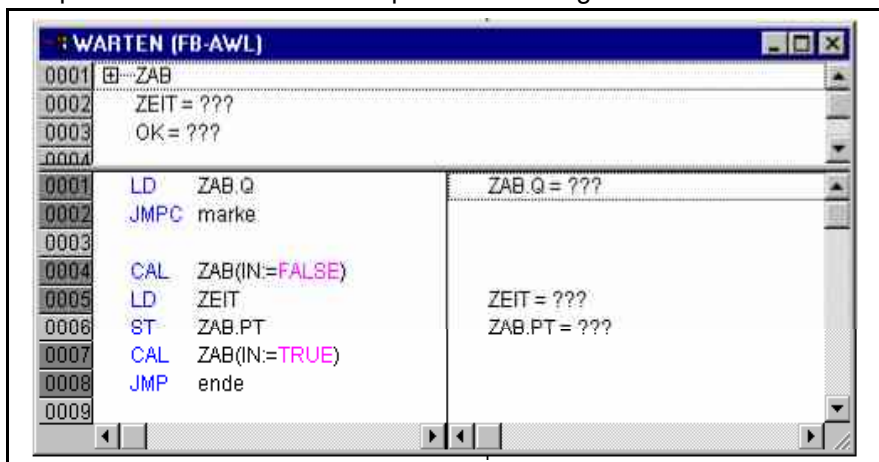


Fig. 5-39 : Editor AWL con posiciones de breakpoint posibles (campos de número de línea más oscuros)

Para insertar un breakpoint, el usuario hace clic con el ratón sobre el campo de número de la línea en la que desea insertar el breakpoint. Si el campo seleccionado es una posición de breakpoint, el color del campo de número de línea cambia de gris oscuro a azul claro y se activa el breakpoint en el control.

De forma análoga, para borrar un breakpoint se hace clic sobre el campo de número de la línea que contiene el breakpoint a eliminar.

La inserción y el borrado de breakpoints también puede seleccionarse mediante menú ("**Online**" "**Colocar/suprimir breakpoint**"), mediante la tecla de función <F9> o mediante el símbolo en la barra de funciones.

### ¿Qué pasa en un breakpoint?

Si en el control se ha alcanzado un breakpoint, se muestra en la pantalla el segmento con la línea correspondiente. El campo de número de la línea en la que se encuentra el control se muestra en rojo. En el control se detiene la ejecución del programa del usuario.

Si el programa se encuentra en un breakpoint, se puede reanudar el procesamiento mediante "Online" "Inicio".

Además, mediante "Online" "Paso individual sobre" o "Paso individual en" se puede avanzar sólo hasta la siguiente posición de breakpoint. Si la instrucción sobre la que nos encontramos es una orden CAL, o si en las líneas hasta la siguiente posición de breakpoint se halla una llamada de función, se omite mediante "Paso individual sobre", mientras que con "Paso individual en" se ramifica al componente llamado.

### Números de línea del editor de texto

Los números de línea del editor de texto indican el número de cada línea de texto de una implementación de un componente.

En el modo Offline, un solo clic sobre un número de línea concreto marca la línea de texto entera.

En el modo Online, el color del fondo del número de línea indica el estado de breakpoint de cada línea:

- gris oscuro: esta línea es una posición posible para un breakpoint.
- azul claro: en esta línea se ha insertado un breakpoint.
- rojo: la ejecución del programa se encuentra en este punto.

En el modo Online, un clic simple con el ratón cambia el estado de breakpoint de la línea en cuestión.

## El editor de listas de instrucciones

Éste es el aspecto que presenta un componente escrito en AWL en el correspondiente editor IndraLogic:

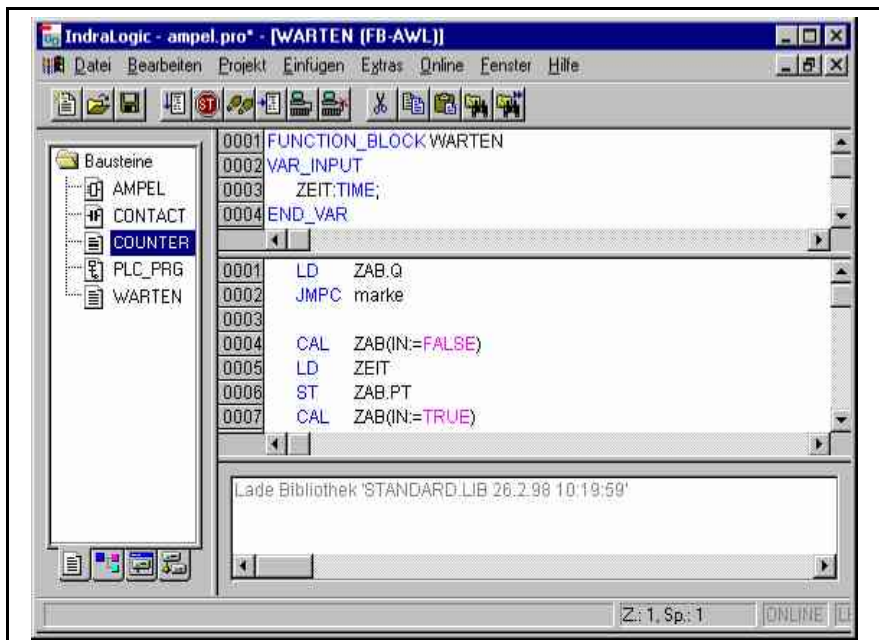


Fig. 5 -40 : Componente escrito en AWL en el editor IndraLogic

Todos los editores para componentes constan de una parte de declaración y un tronco. Ambas áreas están separadas por un divisor de pantalla.

El editor de listas de instrucciones es un editor de texto con las funciones habituales de los editores de texto de Windows. Las órdenes más importantes se encuentran en el menú contextual (botón derecho del ratón).

Es posible una llamada de componente de varias líneas: Ejemplo:

```

CAL CTU_inst (
CU:=%IX10,
PV:=(
LD A
ADD 5
)
)

```

Fig. 5 -41 : Llamada de componente de varias líneas

Para información sobre el lenguaje, ver Lista de instrucciones (AWL).

### AWL en el modo Online

Mediante la orden "Online" "Control de proceso", en el editor AWL se inserta en el lado izquierdo de cada línea un campo adicional en el que se muestra el contenido del acumulador.

Para información adicional sobre el editor AWL en el modo Online, ver el capítulo "Los editores de texto en el modo Online".

## El editor para Texto estructurado

Éste es el aspecto que presenta un componente escrito en ST en el correspondiente editor IndraLogic:

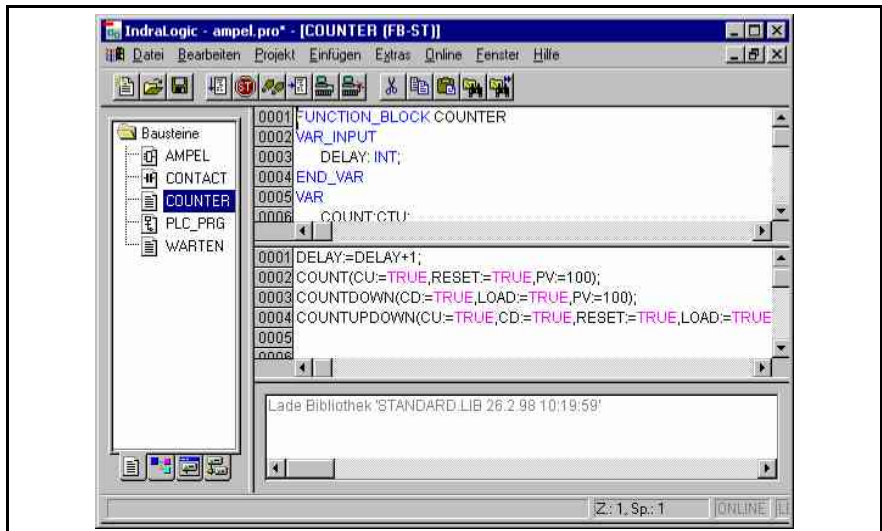


Fig. 5 -42 : Componente escrito en ST en el editor IndraLogic

Todos los editores para componentes constan de una parte de declaración y un tronco. Ambas áreas están separadas por un divisor de pantalla.

El editor para Texto estructurado es un editor de texto con las funciones habituales de los editores de texto de Windows. Las órdenes más importantes se encuentran en el menú contextual (botón derecho del ratón).

Para información adicional sobre el editor ST en el modo Online, consulte "Los editores de texto en el modo Online".

Para información adicional sobre el lenguaje, ver "Texto estructurado (ST)".





## 5.4 Editores de los lenguajes de programación gráficos

### Trabajar en los editores gráficos

Los editores de los lenguajes orientados gráficamente Lenguaje de proceso AS, Esquema de contactos KOP, Esquema de funciones FUP y Editor de esquemas de funciones continuo CFC presentan numerosos aspectos en común. Estos aspectos se describen en las secciones (ver abajo) Zoom, Red, Marcas de salto, Comentarios de red, Insertar red, Editores de red en el modo Online. Además se ofrecen las descripciones específicas de KOP, FUP y CFC, así como de la Lenguaje de proceso AS.

La implementación en los editores gráficos se facilita mediante coloración sintáctica.

#### Zoom

Objetos tales como componentes, acciones, transiciones, etc. en los lenguajes AS, KOP, FUP, CFC y en visualizaciones pueden ampliarse o reducirse mediante una función de zoom. La función abarca todos los elementos del contenido de la ventana de la parte de implementación, mientras que la parte de declaración permanece inalterada.

Como ajuste estándar, cada objeto se muestra con el nivel de zoom 100%. El nivel de zoom ajustado se guarda en el proyecto como propiedad del objeto.

¡La impresión de la documentación del proyecto tiene lugar siempre en la representación 100%!

El nivel de zoom puede ajustarse mediante una lista de selección en la barra de símbolos. Se pueden escoger valores entre 25% y 400%, así como especificar manualmente valores concretos entre 10% y 500%.

La selección del nivel de zoom sólo está disponible si el cursor se halla en un objeto creado en un lenguaje gráfico o en un objeto de visualización.

Las posiciones del cursor en los editores pueden seguir siendo seleccionadas y alcanzadas mediante las teclas de flecha cuando está activo el zoom para el objeto. El tamaño del texto se rige por el factor de zoom y el tamaño de fuente ajustado.

La ejecución de todas las opciones de menú para el manejo del editor (p. ej. inserción de una casilla) en relación con la posición del cursor es posible en cualquier nivel de zoom y conservando dicho nivel.

En el modo Online, cada objeto se representa conforme al nivel de zoom ajustado, y las funciones Online están disponibles sin restricciones.

Si se utiliza el IntelliMouse, es posible ampliar/reducir un objeto pulsando la tecla <Ctrl> mientras se gira la rueda hacia delante/atrás.

#### Red

En los editores KOP y FUP, el programa se organiza en una lista de redes. Cada red está identificada en el lado izquierdo por un número de red correlativo, y contiene una estructura que representa una expresión lógica o aritmética, una llamada de programa, de función o de bloque de función, un salto o una instrucción de retorno.



### Marcas de salto

Cada red cuenta con una marca de salto, que opcionalmente puede estar vacía. Para editar esta marca se debe hacer clic en la primera línea de la red, directamente junto al número de red. Ahora se puede introducir una marca seguida de dos puntos

### Comentarios, saltos de línea, "Extras" "Opciones"

Se puede asignar un comentario de varias líneas a cada red. En el diálogo "Opciones del Esquemas de funciones y de contactos", que se abre mediante la orden "Extras" "Opciones", se pueden realizar ajustes en relación con los comentarios:

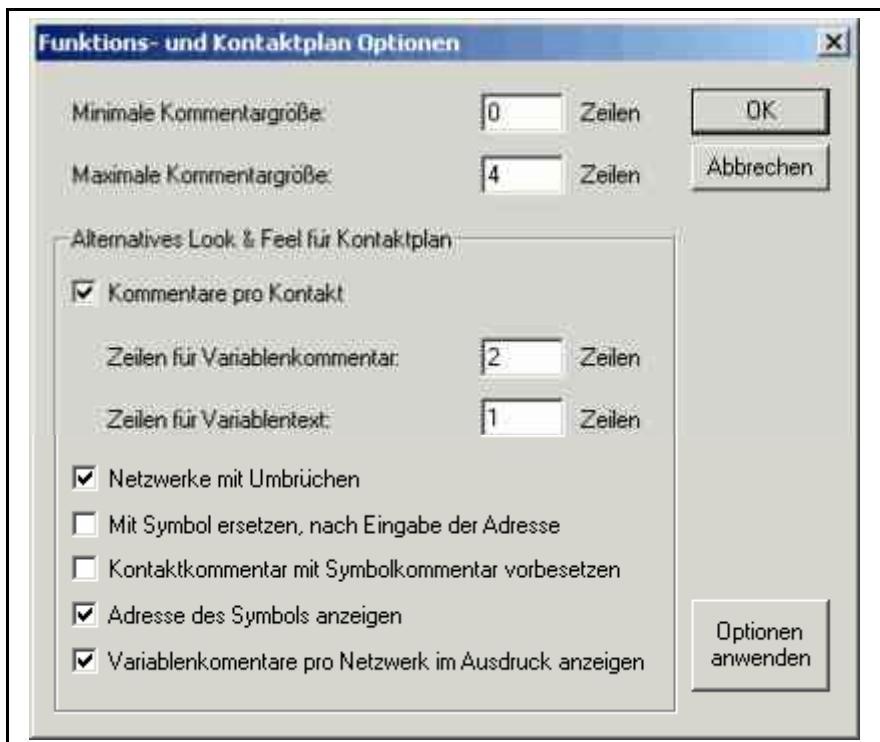


Fig. 5 -43 : Diálogo Opciones del Esquema de funciones y de contactos

**Tamaño máximo del comentario:** Número máximo de líneas que estarán disponibles para un comentario de red (el valor predeterminado aquí es 4).

**Tamaño mínimo del comentario:** Número de líneas que generalmente deberán reservarse o mostrarse para comentarios. Por ejemplo, si aquí se especifica 2, en cada principio de red se mostrarán dos líneas de comentario vacías detrás de la línea de etiqueta. El valor predeterminado aquí es 0, lo cual ofrece la ventaja de que caben más redes en el área de la pantalla.

Si el tamaño mínimo del comentario de red es mayor que 0, para introducir un comentario basta con hacer clic en la línea de comentario mostrada e introducir el comentario. De lo contrario, primero se debe seleccionar la red para la cual se desea introducir el comentario e insertar una línea de comentario mediante "**Insertar**" "**Comentario**". A diferencia del texto del programa, los comentarios se muestran en color gris.

Visualización alternativa: Las siguientes opciones permiten una visualización alternativa de las redes.

**Comentarios por contacto** (sólo para el Esquema de contactos): Si está activada esta opción se pueden asignar comentarios para contactos y bobinas individuales. Introduzca en el campo **Líneas para comentario de variable** el número deseado de líneas que deben ser reservadas y mostradas al efecto. A continuación aparece un campo de comentario encima del contacto o la bobina, y se puede introducir texto.

Si está activada la opción "Comentarios por contacto", además se puede definir el número de líneas (**Líneas para texto de variable:**) que se utilizarán para el nombre de variable del contacto o la bobina, para que sea posible visualizar por completo también nombres largos mediante la utilización de varias líneas. En el siguiente ejemplo se han previsto 2 líneas para el comentario del contacto y 1 línea para el texto de la variable:

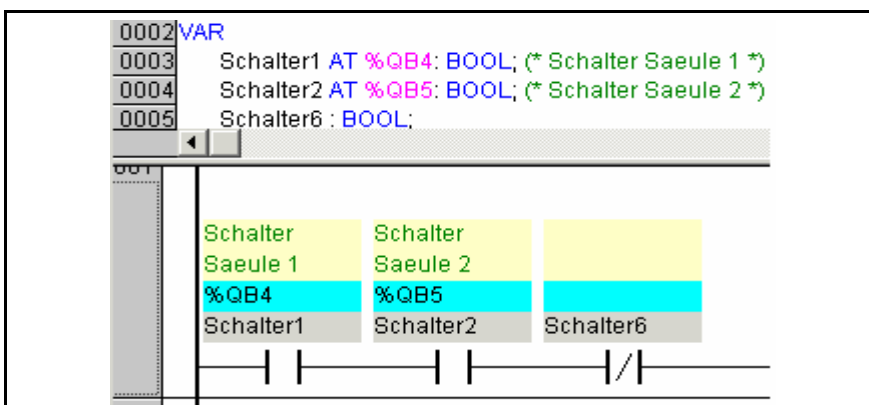


Fig. 5 -44 : Red con visualización de comentario de variable y dirección por contacto

**Redes con saltos de línea** (sólo para el esquema de contactos): si está activada esta opción, se insertan en las redes saltos de línea cuando la anchura de ventana ajustada ya no permite mostrar todos los elementos de la red.

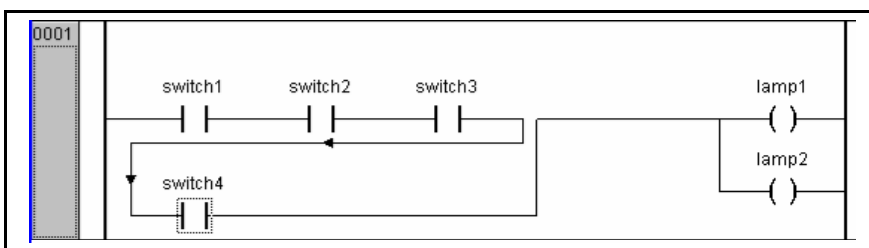


Fig. 5 -45 : Red con salto de línea

**Sustituir por símbolo, tras la introducción de la dirección:** Si está activada esta opción, puede introducir una dirección (p. ej. "%QB4") junto al componente, el contacto o la bobina, e inmediatamente después de la introducción ésta será sustituida por el nombre de la variable asignada a dicha dirección. Si una dirección no tiene asignada ninguna variable, se muestra inalterada.

**Preasignar comentario de contacto con comentario de símbolo** (sólo para el esquema de contactos): Si está activada esta opción, en el campo de comentario del contacto o de la bobina se muestra el comentario que se definió para la variable utilizada durante la declaración de ésta, y allí puede ser editado (ver arriba, figura para la opción "Comentarios por contacto"). Sin embargo, para ello debe estar activada también la opción "Comentarios por contacto" (ver arriba). Nota: ¡En este caso, un comentario ya introducido localmente en un campo de comentario es sustituido automáticamente por el comentario de variable, o bien por

espacios en blanco si no existe ningún comentario en la declaración de variable!

**Mostrar dirección del símbolo** (sólo para el esquema de contactos): Si la variable introducida en el contacto o en la bobina está asignada a una dirección, se muestra adicionalmente encima del nombre de la variable (ver arriba, figura para la opción "Comentarios por contacto").

**Mostrar en la expresión comentarios de variable por red:** Si está activada esta opción, por cada red para cada variable utilizada en la red se muestra una línea que indica el nombre de la variable, la dirección, el tipo de dato y el comentario de variable, tal como se definieron en la declaración de variable. Esto puede resultar útil para la documentación del proyecto (impresión). Ejemplo:

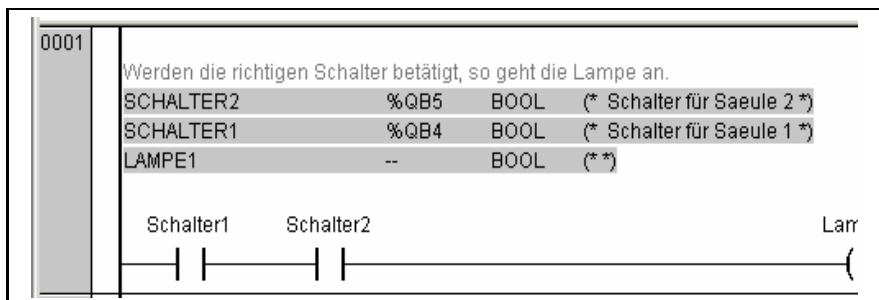


Fig. 5 -46 : Visualización de una línea con informaciones para cada variable de la red

Aplicación de las opciones:

**OK:** mediante este botón se aplican en el componente actual las opciones ajustadas y se cierra el diálogo.

**Aplicar opciones:** mediante este botón se pueden aplicar en todo el proyecto las opciones ajustadas. Aparece un diálogo de confirmación en el que deberá confirmar explícitamente la opción.

**"Insertar" "Red (detrás)" o "Insertar" "Red (delante)"**

**Forma abreviada:** <Mayúsculas>+<T>

Para insertar una nueva red en el editor FUP o KOP, se selecciona la orden **"Insertar" "Red (detrás)"** o **"Insertar" "Red (delante)"**, en función de si se desea insertar la nueva red delante o detrás de la red actual. La red actual se cambia haciendo clic sobre el número de red. Se identifica por el rectángulo punteado visible debajo del número. Mediante la tecla <Mayúsculas> y un clic del ratón se selecciona toda el área de redes entre la red actual y la red sobre la que se ha hecho el clic.

### Los editores de redes en el modo Online

En los editores FUP y KOP se pueden insertar breakpoints sólo en redes. El campo de número de una red en la que se ha insertado un breakpoint se muestra en azul. En ese caso, el procesamiento se detiene delante de la red en la que se encuentra el breakpoint. En este caso, el campo de número de red se muestra en rojo. En el procesamiento por pasos individuales se salta de una red a otra.

Todos los valores se monitorizan en las entradas y salidas de los componentes de la red.

Durante la monitorización de expresiones o variables direccionadas por bit se debe tener en cuenta lo siguiente: En expresiones, p. ej. a AND b como condición de transición o entrada de bloque de función, se muestra siempre el valor de la expresión entera (a AND b se muestra en azul o con :=TRUE si a y b son TRUE). En caso de variables direccionadas por bit se monitoriza siempre el valor de bit disecionado (p. ej. a.3 se representa en azul o con :=TRUE si a tiene el valor 4).

Inicie el control de proceso mediante la orden de menú "Online" "Control de proceso". Mediante esta orden puede visualizar los valores actuales que se transportan en las redes por las líneas de conexión. Si las líneas de conexión no transportan valores booleanos, se muestra el valor en un campo insertado adicionalmente. Los campos de monitor para variables que no se utilizan (p. ej. en el caso de la función SEL) se visualizan sombreados en gris. Si las líneas transportan valores booleanos, se colorean en azul en caso de que transporten TRUE. De este modo se puede seguir el flujo de información durante el funcionamiento del control.

Si mantiene el puntero del ratón brevemente sobre una variable, se muestran en un tooltip (información de herramientas) el tipo, la dirección y el comentario de la variable.

## El editor de esquemas de funciones

El editor de esquemas de funciones es un editor gráfico. Trabaja con una lista de redes, cada una de las cuales contiene una estructura que representa una expresión lógica o aritmética, la llamada a un bloque de función, una función, un programa, un salto o una instrucción de retorno.

Éste es el aspecto que presenta un componente escrito en FUP en el correspondiente editor IndraLogic:

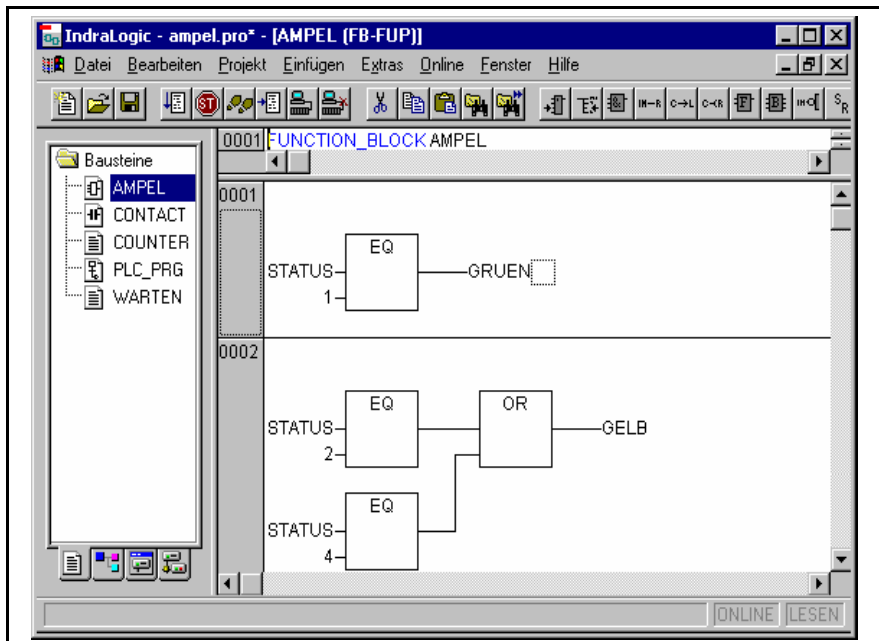


Fig. 5 -47 : Componente escrito en FUP en el editor IndraLogic  
Las órdenes más importantes se encuentran en el menú contextual (botón derecho del ratón).

Tenga en cuenta la posibilidad de alternar entre la representación en el editor FUP y el editor KOP para un componente creado en FUP, tanto en modo Offline como Online (ver más abajo "Extras" "Vista"). Tenga en cuenta también las posibilidades de ajuste para comentarios, introducción de dirección, etc., mediante el diálogo de opciones, ver capítulo "Comentarios, saltos de línea, "Extras" "Opciones",", página 5-29.

### Posiciones del cursor en FUP

Cada texto es una posible posición del cursor. El texto seleccionado se muestra con fondo azul y ahora puede ser modificado.

Por lo demás, la posición actual del cursor está identificada por un rectángulo punteado. A continuación se enumeran todas las posiciones del cursor posibles con un ejemplo:

1. Cada campo de texto:

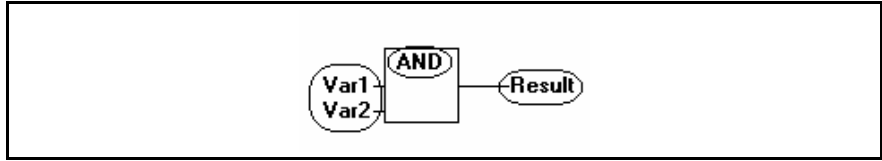


Fig. 5 -48 : Posición del cursor Campo de texto (las posiciones del cursor posibles se muestran con un marco negro)

2. Cada entrada:

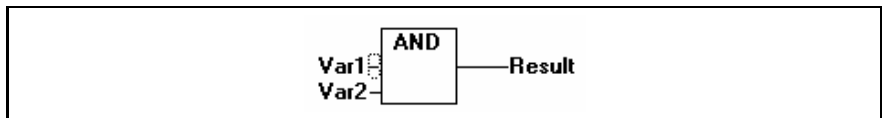


Fig. 5 -49 : Posición del cursor Entrada

3. Cada operador, cada función o cada módulo de función:

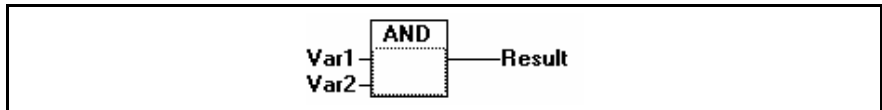


Fig. 5 -50 : Posición del cursor Operador

4. Salidas, cuando a continuación viene una asignación o un salto:

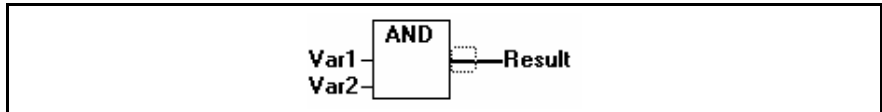


Fig. 5 -51 : Posición del cursor Salida

5. La intersección de líneas sobre una asignación, un salto o una instrucción de retorno:



Fig. 5 -52 : Posición del cursor Intersección de líneas

6. Detrás del objeto situado más a la derecha de cada red ("última posición del cursor"; es también la posición del cursor cuando se ha seleccionado una red):

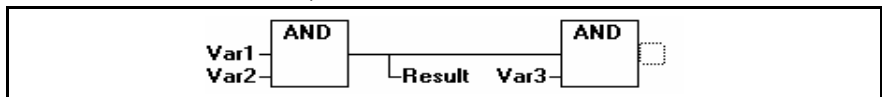


Fig. 5 -53 : Posición del cursor detrás del objeto situado más a la derecha

7. La intersección de líneas inmediatamente delante de una asignación:

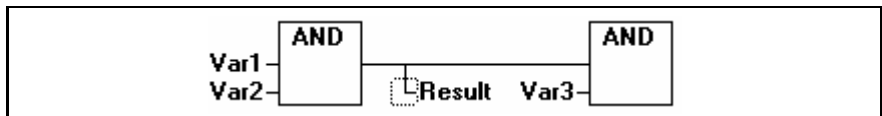


Fig. 5-54 : Posición del cursor Intersección de líneas inmediatamente delante de una asignación



### Cómo situar el cursor en FUP

El cursor puede situarse en una posición determinada haciendo clic con el ratón o utilizando el teclado.

Mediante las teclas de flecha se salta hasta la siguiente posición del cursor en la dirección seleccionada. De este modo se puede acceder a todas las posiciones del cursor, incluidos los campos de texto. Si se ha seleccionado la última posición del cursor, mediante las teclas de flecha <hacia arriba> o <hacia abajo> se puede seleccionar la última posición del cursor de la red anterior o de la siguiente, respectivamente.

Una red vacía contiene sólo tres signos de interrogación "???". Haciendo clic detrás de ésta se selecciona la última posición del cursor.

### "Insertar" "Asignación" en FUP

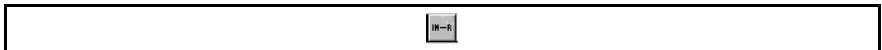


Fig. 5-55 : Símbolo: "Insertar" "Asignación"

#### Forma abreviada: <Ctrl>+<A>

Esta orden inserta una asignación.

La inserción tiene lugar, en función de la posición seleccionada (ver "Posiciones del cursor en FUP"), inmediatamente delante de la entrada seleccionada, inmediatamente detrás de la salida seleccionada o al final de la red.

Para una asignación insertada, a continuación es posible seleccionar el texto "???" introducido y sustituirlo por la variable a la que se desea asignar. Para ello también puede utilizar la ayuda de entrada. Tenga en cuenta la posibilidad de introducir direcciones si así se ha ajustado en el diálogo de opciones (ver capítulo: Comentarios, saltos de línea, "Extras" "Opciones").

Para añadir una asignación adicional a una asignación existente, utilice la orden "Insertar" "Salida".

### "Insertar" "Salto" en FUP

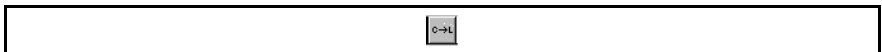


Fig. 5-56 : Símbolo: "Insertar" "Salto"

#### Forma abreviada: <Ctrl>+<L>

Esta orden inserta un salto.

La inserción tiene lugar, en función de la posición seleccionada (ver "Posiciones del cursor en FUP"), inmediatamente delante de la entrada seleccionada, inmediatamente detrás de la salida seleccionada o al final de la red.

Para un salto insertado, a continuación es posible seleccionar el texto "???" introducido y sustituirlo por la marca de salto a la que se desea saltar.

### "Insertar" "Retorno" en FUP



Fig. 5-57 : Símbolo: "Insertar" "Retorno"

#### Forma abreviada: <Ctrl>+<R>

Esta orden inserta una instrucción RETURN.

La inserción tiene lugar, en función de la posición seleccionada (ver "Posiciones del cursor en FUP"), inmediatamente delante de la entrada



seleccionada, inmediatamente detrás de la salida seleccionada o al final de la red.

### "Insertar" "Componente" en FUP



Fig. 5-58 : Símbolo: "Insertar" "Componente"

#### Forma abreviada: +

Mediante esta orden se pueden insertar operadores, funciones, bloques de función y programas. En primer lugar se inserta siempre un operador "AND". Seleccionando y sobrescribiendo el texto de tipo ("AND"), dicho operador puede transformarse en cualquier otro operador, en cualquier función, en cualquier bloque de función y en cualquier programa. Mediante la ayuda de entrada (<F2>) puede seleccionar el componente deseado. Si el nuevo componente seleccionado tiene otra cantidad mínima de entradas, se adjuntan éstas. Si el nuevo componente tiene una cantidad máxima de entradas menor, se borran las últimas entradas. En funciones y bloques de función se indican los nombres formales de las entradas y salidas.

En los bloques de función existe un campo de instancia editable encima de la caja. Si mediante la modificación del texto de tipo se llama otro bloque de función desconocido, se muestra una caja de operador con dos entradas y el tipo especificado. Si se selecciona el campo de instancia, mediante <F2> se puede llamar a la ayuda de entrada con las categorías para la selección de variables.

El nuevo componente se inserta dependiendo de la posición seleccionada (ver "Posiciones del cursor en FUP"):

- Si se ha seleccionado una entrada, el componente se inserta delante de dicha entrada. La primera entrada de este componente se conecta a la rama situada a la izquierda de la entrada seleccionada. La salida del nuevo componente se conecta a la entrada seleccionada.
- Si se ha seleccionado una salida, el componente se inserta detrás de dicha salida. La primera entrada del componente se conecta a la salida seleccionada. La salida del nuevo componente se conecta a la rama a la que estaba conectada la salida seleccionada.
- Si está seleccionado un componente, una función o un bloque de función, el elemento antiguo es reemplazado por el nuevo componente. En la medida de lo posible, las ramas se conectan como antes de la sustitución. Si el elemento antiguo tenía más entradas que el nuevo, se borran las ramas no conectables. Lo mismo rige para las salidas.
- Si se ha seleccionado un salto o un retorno, el componente se inserta delante de dicho salto o retorno. La primera entrada del componente se conecta a la rama situada a la izquierda del elemento seleccionado. La salida del componente se conecta a la rama situada a la derecha del elemento seleccionado.
- Si se ha seleccionado la última posición del cursor de una red, el componente se inserta detrás del último elemento. La primera entrada del componente se conecta a la rama situada a la izquierda de la posición seleccionada.

Todas las entradas del componente que no hayan podido conectarse reciben el texto "???". Es preciso hacer clic sobre este texto y cambiarlo a la constante o variable deseada.

Si a la derecha de un componente insertado se encuentra una rama, se asigna a ésta la primera salida del componente. De lo contrario, las salidas permanecen sin asignar.

### "Insertar" "Entrada"



Fig. 5-59 : Símbolo: "Insertar" "Entrada"

#### **Forma abreviada: <Ctrl>+<U>**

Esta orden inserta una entrada de operador. En muchos operadores, el número de entradas es variable (p. ej. ADD puede tener 2 o más entradas).

Para ampliar en uno el número de entradas en tal operador, es preciso seleccionar la entrada delante de la cual se debe insertar otra, o bien el propio operador, en caso de que se deba insertar una entrada inferior (ver Posiciones del cursor en FUP).

La entrada insertada está ocupada con el texto "???". Es preciso hacer clic sobre este texto y cambiarlo a la constante o variable deseada. Para ello también puede utilizar la ayuda de entrada. Tenga en cuenta la posibilidad de introducir direcciones si así se ha ajustado en el diálogo de opciones (ver capítulo: Comentarios, saltos de línea, "Extras" "Opciones" en la página 5-29).

### "Insertar" "Salida"

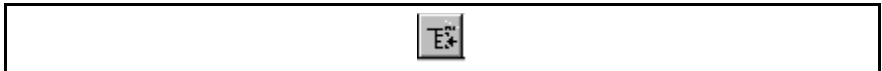


Fig. 5-60 : Símbolo: "Insertar" "Salida"

Esta orden añade una asignación adicional a una asignación ya existente. Esta función sirve para crear los denominados "peines de asignación", esto es, la asignación a varias variables del valor actualmente existente en la línea.

Si está seleccionada la intersección de líneas sobre una asignación o la salida situada inmediatamente delante, se añade una nueva asignación detrás de las ya existentes.

Si está seleccionada la intersección de líneas directamente delante de una asignación, se inserta una nueva asignación delante de esta asignación.

La salida insertada está ocupada con el texto "???". Es preciso hacer clic sobre este texto y cambiarlo a la variable deseada. Para ello también puede utilizar la ayuda de entrada. Tenga en cuenta la posibilidad de introducir direcciones si así se ha ajustado en el diálogo de opciones (ver capítulo: Comentarios, saltos de línea, "Extras" "Opciones" en la página 5-29).

### "Extras" "Negación"



Fig. 5-61 : Símbolo: "Extras" "Negación"

#### **Forma abreviada: <Ctrl>+<N>**

El símbolo de la negación es un pequeño círculo en una conexión.

Si está seleccionada una entrada, se deniega dicha entrada.

Si está seleccionada una salida, se deniega dicha salida.

Si está seleccionado un salto o un retorno, se deniega la entrada de dicho salto o retorno.

Una negación puede borrarse mediante una nueva negación.

**"Extras" "Set/Reset"**

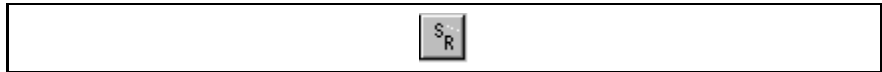


Fig. 5-62 : Symbol: "Extras" "Set/Reset"

Mediante esta orden se pueden definir salidas como salidas Set o Reset. Una rejilla con salida Set se representa con [S] y una rejilla con salida Reset se representa con [R].

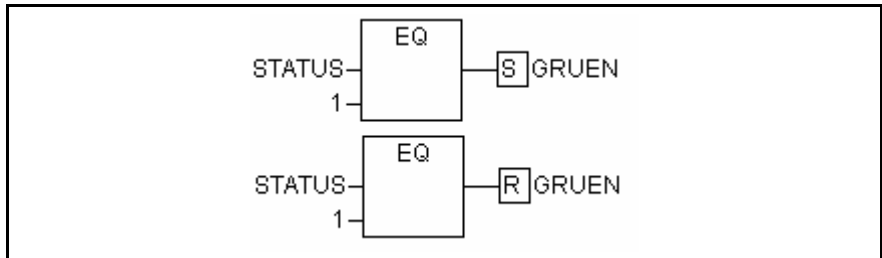


Fig. 5-63 : Salidas Set/Reset en FUP

Una salida Set se ajusta a TRUE si la rejilla correspondiente arroja TRUE. La salida conserva este valor aunque la rejilla vuelva a saltar a FALSE.

Una salida Reset se ajusta a FALSE si la rejilla correspondiente arroja TRUE. La salida conserva su valor aunque la rejilla vuelva a saltar a FALSE.

Si se ejecuta esta orden varias veces, la salida alterna entre salida Set, Reset y normal.

**"Extras" "Vista"**

Mediante esta orden se puede seleccionar entre la representación en el editor de Esquema de contactos y el editor de Esquema de funciones para un componente creado en el editor de Esquema de funciones. Esto es posible tanto en el modo Offline como en el Online.

**Zoom al componente llamado**

Forma abreviada: <Alt>+<Intro>

Esta orden está disponible en el menú contextual (<F2>) o en el menú Extras cuando el cursor se halla sobre el nombre de un componente llamado en los editores de texto o cuando está marcada la casilla de un componente en los editores gráficos. El zoom abre el componente en cuestión en su ventana del editor.

Si el componente procede de una biblioteca, se llama el administrador de bibliotecas y se muestra el componente en cuestión.

**Cortar, copiar, insertar y borrar en FUP**

Las órdenes para "Cortar", "Copiar", "Insertar" o "Borrar" se encuentran en la opción de menú "Editar".

Si está seleccionada una intersección de líneas, se cortan, se borran o se copian las asignaciones, los saltos o las instrucciones RETURN situadas debajo.

Si está seleccionado un componente, se corta, se borra o se copia el propio objeto seleccionado, así como todas las ramas dependientes de las entradas, a excepción de la primera rama (posición más alta).

En los demás casos, se corta, se borra o se copia toda la rama situada delante de la posición del cursor.

Después de copiar o cortar, la parte borrada o copiada se halla en el portapapeles y puede insertarse tantas veces como se desee.



Para ello, primero se debe seleccionar la posición de inserción. Las posiciones de inserción válidas son las entradas y salidas.

Si se ha cargado un componente en el portapapeles (recuerde: en este caso todas las ramas dependientes, excepto la primera, se hallan también en el portapapeles), la primera entrada se conecta a la rama delante de la posición de inserción.

De lo contrario, la totalidad de la rama situada delante de la posición de inserción es sustituida por el contenido del portapapeles.

En cualquier caso, el último elemento insertado se conecta a la rama situada a la derecha de la posición de inserción.

---

**Nota:** Mediante cortar e insertar se puede solucionar el siguiente problema: se inserta un nuevo operador en mitad de una red. La rama situada a la derecha del operador está ahora conectada a la primera entrada, pero debe estar conectada a la segunda entrada. Entonces se selecciona la primera entrada y se ejecuta "**Editar**" "**Cortar**". A continuación se selecciona la segunda entrada y se ejecuta "**Editar**" "**Insertar**". Ahora, la rama cuelga de la segunda entrada.

---

## El esquema de funciones en el modo Online

En el esquema de funciones se pueden insertar breakpoints sólo en redes. Si se ha insertado un breakpoint en una red, se muestra en azul el campo de número de la red. En ese caso, el procesamiento se detiene delante de la red en la que se encuentra el breakpoint. En este caso, el campo de número de red se vuelve rojo. En el procesamiento por pasos individuales se salta de una red a otra.

Se indica el valor actual para cada variable. Excepción: Si la entrada de un bloque de función es una expresión, se monitoriza sólo la primera variable de la expresión.

Un doble clic sobre una variable abre el diálogo para escribir una variable. Aquí es posible modificar el valor actual de la variable. En las variables booleanas no aparece ningún diálogo, sino que su valor se conmuta.

El nuevo valor se vuelve rojo y permanece inalterado. Si se ejecuta la orden "Online" "Escribir valores", se ajustan todas las variables a los valores escogidos y se vuelven a mostrar en negro.

Inicie el control de proceso mediante la orden de menú "Online" "Control de proceso". Mediante esta orden puede visualizar los valores actuales que se transportan en las redes por las líneas de conexión. Si las líneas de conexión no transportan valores booleanos, se muestra el valor en un campo insertado al efecto. Si las líneas transportan valores booleanos, se colorean en azul en caso de que transporten TRUE. De este modo se puede seguir el flujo de información durante el funcionamiento del control.

Si mantiene el puntero del ratón brevemente sobre una variable, se muestran en un tooltip (información de herramientas) el tipo, la dirección y el comentario de la variable.

## El editor de esquemas de contactos

Éste es el aspecto que presenta un componente escrito en KOP en el editor IndraLogic:

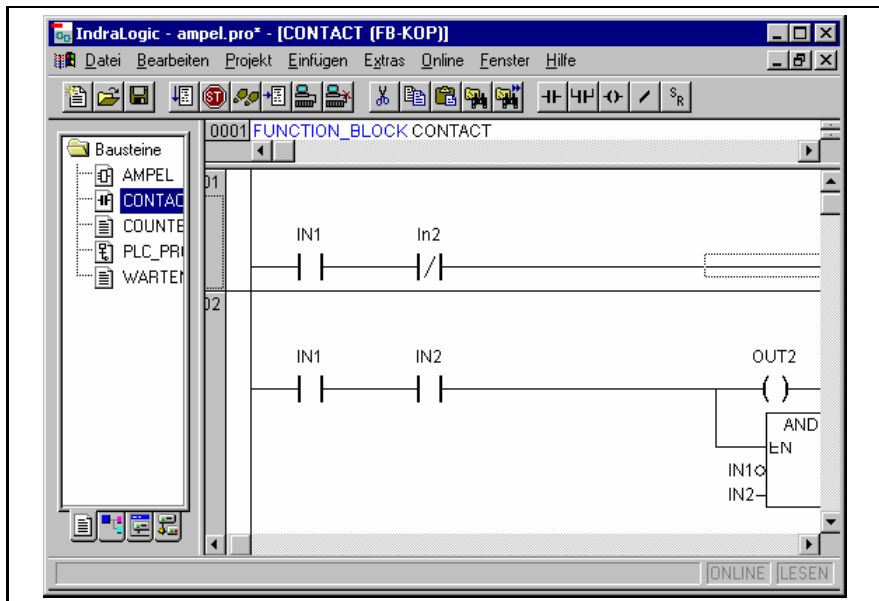


Fig. 5-64 : Componente escrito en KOP en el editor IndraLogic

Todos los editores para componentes constan de una parte de declaración y un tronco. Ambas áreas están separadas por un divisor de pantalla.

El editor KOP es un editor gráfico. Las órdenes más importantes se encuentran en el menú contextual (botón derecho del ratón).

Para hallar información sobre los elementos, consulte el capítulo: Esquema de contactos (KOP), página 2-26.

### Posiciones del cursor en el editor KOP

Los siguientes puntos pueden ser posiciones del cursor, teniendo en cuenta que las llamadas de bloque de función y de programa pueden tratarse como contactos. Los componentes con entradas EN y otros componentes vinculados a ellas se tratan igual que en el esquema de funciones. Hallará información sobre la edición de estas partes de la red en "Editor FUP", en el capítulo Esquema de funciones (FUP) a partir de la página 5-32).

8. Cada campo de texto:

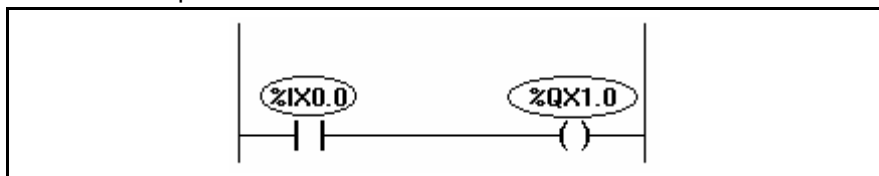


Fig. 5-65 : Posición del cursor Campo de texto (las posiciones del cursor posibles enmarcadas en negro)

9. Cada contacto o bloque de función:

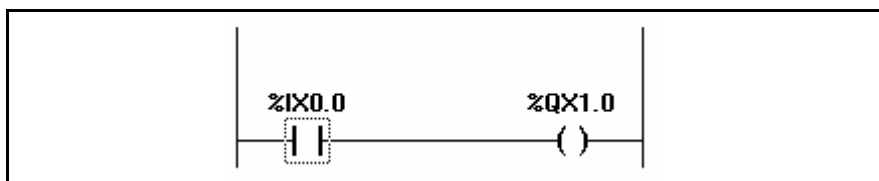


Fig. 5-66 : Posición del cursor Contacto

10. Cada bobina

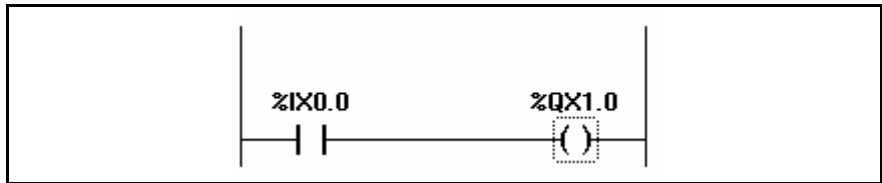


Fig. 5-67 : Posición del cursor Bobina

11. La línea de conexión entre los contactos y las bobinas:

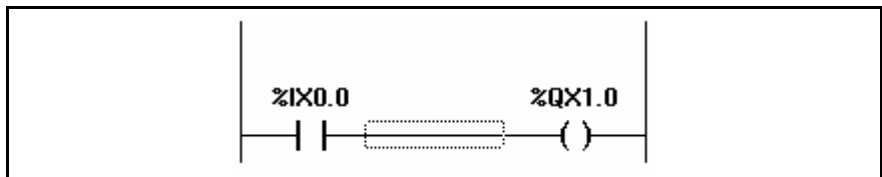


Fig. 5-68 : Posición del cursor Línea de conexión

### Desplazar elementos, nombres en el editor KOP

Mediante "Drag&Drop" (arrastrar y soltar) se puede desplazar a otra posición dentro del componente tanto un elemento entero (contacto, bobina, bloque de función) de un componente KOP como tan sólo el nombre (nombre de variable, dirección, comentario) de un elemento.

Para ello, seleccione el contacto, la bobina o el bloque de función deseado y arrástrelo fuera de su posición actual manteniendo pulsado el botón del ratón. A continuación se indican con rectángulos rellenos de gris todas las posiciones posibles dentro de las redes del componente a las que se puede desplazar el elemento.

En cuanto se arrastra el elemento a una de estas marcas, ésta se muestra rellena de verde. Al soltar entonces el botón del ratón se inserta el elemento en la nueva posición.

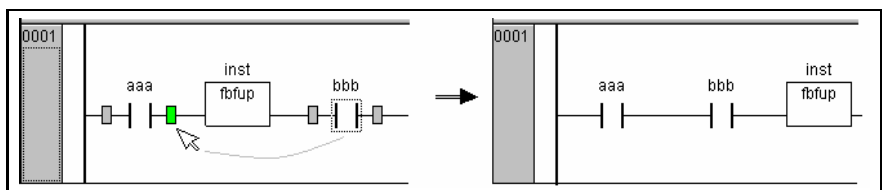


Fig. 5-69 : Desplazar elementos

Si, por el contrario, arrastra el elemento sobre la inscripción (nombre de variable) de otro elemento, éste se muestra con un fondo verde. Al soltar entonces el botón del ratón, el nombre actual es sustituido por el nombre "arrastrado". En caso de que se indiquen además la dirección y el comentario, la copia también incluye a ambos.

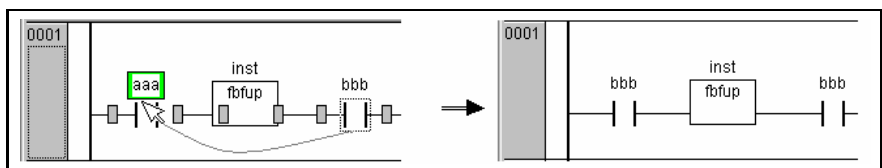


Fig. 5-70 : Desplazar nombres



### "Insertar" "Contacto" en KOP



Fig. 5-71 : Símbolo: "Insertar" "Contacto"

#### Forma abreviada: <Ctrl>+<O>

Mediante esta orden se inserta en el editor KOP un contacto delante de la posición marcada en la red.

Si la posición marcada es una bobina o la línea de conexión entre los contactos y las bobinas, el nuevo contacto se conectará serialmente a la conexión de contacto previa.

El contacto recibe como preasignación el texto "???". Puede hacer clic en este texto y cambiarlo al nombre de la variable o la constante deseada. Para ello también puede utilizar la ayuda de entrada. Tenga en cuenta la posibilidad de introducir direcciones si así se ha ajustado en el diálogo de opciones; ver capítulo: Comentarios, saltos de línea, "Extras" "Opciones" en la página 5-29.

Si ha activado la opción **Comentarios por contacto** también en el diálogo Opciones, puede especificar allí, junto a una cantidad deseada de **Líneas para el comentario de variable**, también una cantidad determinada de **Líneas para el nombre de variable**. Esto es recomendable en caso de nombres de variable largos, a fin de mantener la red compacta horizontalmente.

Tenga en cuenta además la opción **Redes con saltos de línea**, que también puede activar mediante "Extras" "Opciones".

### "Insertar" "Contacto paralelo" en KOP



Fig. 5-72 : Symbol: "Insertar" "Contacto paralelo"

#### Forma abreviada: <Ctrl>+<R>

Mediante esta orden se inserta en el editor KOP un contacto paralelo a la posición marcada en la red.

Si la posición marcada es una bobina o la conexión entre los contactos y las bobinas, el nuevo contacto se conectará en paralelo a la conexión de contacto previa.

El contacto recibe como preasignación el texto "???". Puede hacer clic en este texto y cambiarlo a la variable o la constante deseada. Para ello también puede utilizar la ayuda de entrada. Acerca de la posible introducción de direcciones, de la representación del nombre de variable en varias líneas y de un comentario específico para el contacto, consulte más arriba en "Insertar" "Contacto".

### "Insertar" "Bloque de función" en KOP

#### Forma abreviada: <Ctrl>+<B>

Utilice esta orden para insertar un bloque de función o un programa como componente. Para ello debe estar marcada la conexión entre los contactos y las bobinas o bien una bobina. Se abrirá el diálogo de la ayuda de entrada, donde podrá escoger entre los componentes estándar y definidos por el usuario disponibles.

La primera entrada del nuevo componente insertado se sitúa en la conexión de entrada, y la primera salida en la conexión de salida; por lo tanto, estas variables deben ser necesariamente del tipo BOOL. Todas las demás entradas y salidas del componente se ocupan con el texto

"???". Estas preasignaciones pueden cambiarse a otras constantes, variables o direcciones. Para ello también puede utilizar la ayuda de entrada.

Acerca de la posible introducción de direcciones, de la representación del nombre de variable en varias líneas y de un comentario específico para el bloque de función, consulte la descripción de las opciones de esquema de funciones y de contactos.

### "Insertar" "Bobina" en KOP



Fig. 5-73 : Símbolo: "Insertar" "Bobina"

#### Forma abreviada: <Ctrl>+<L>

Mediante esta orden se inserta en el editor KOP una bobina en paralelo a las bobinas existentes.

Si la posición marcada es la conexión entre los contactos y las bobinas, la nueva bobina se inserta en último lugar. Si la posición marcada es una bobina, la nueva bobina se inserta directamente sobre ella.

La bobina recibe como preasignación el texto "???". Puede hacer clic en este texto y cambiarlo a la variable deseada. Para ello también puede utilizar la ayuda de entrada. Acerca de la posible introducción de direcciones, de la representación del nombre de variable en varias líneas y de un comentario específico para el contacto, consulte más arriba en "Insertar" "Contacto".

### Componentes con entradas EN

Si desea utilizar su red KOP para controlar llamadas a otros componentes, debe insertar un componente con una entrada EN. Dicho componente se conecta en paralelo a las bobinas. Partiendo de ella, puede seguir desarrollando la red igual que en esquema de funciones. Las órdenes para la inserción en un componente EN se encuentran en la opción de menú "Insertar" "Insertar en componente".

Un operador, un bloque de función, un programa o una función con entrada EN se comporta como el componente correspondiente en el esquema de funciones, con la diferencia de que su ejecución se controla mediante la entrada EN. Esta entrada se conecta a la línea de conexión entre las bobinas y los contactos. Si esta conexión transporta la información "TRUE", se evalúa el componente.

Una vez se ha creado un componente con entrada EN, mediante este componente se puede crear una red como en el esquema de funciones. Esto significa que pueden fluir a un componente EN datos de operadores, funciones, bloques de función habituales, y que un componente EN puede transportar datos a tales componentes habituales.

Por lo tanto, si desea programar una red en el editor KOP igual que en FUP, debe insertar primero un operador EN sólo en una nueva red, y a continuación puede continuar desarrollando su red desde este

componente como en el editor FUP. Una red así creada se comporta como la red equivalente en FUP.

### "Insertar" "Componente con EN" en KOP

Mediante esta orden puede insertar en una red KOP un bloque de función, un operador, una función o un programa con entrada EN.

La posición marcada debe ser la conexión entre los contactos y las bobinas o bien una bobina. El nuevo componente se inserta en paralelo a las bobinas y debajo de éstas, y en principio contiene la designación "AND". Puede cambiar esta designación por la deseada. Para ello



también puede utilizar la ayuda de entrada. Están disponibles componentes estándar y definidos por el usuario.

### "Insertar" "Insertar en componente" en KOP

Mediante esta orden puede añadir elementos adicionales a un componente ya insertado (también un componente con entrada EN). Las órdenes en esta opción de menú son ejecutables en las mismas posiciones del cursor que las órdenes correspondientes en el esquema de funciones.

Mediante **Entrada** puede añadir una nueva entrada al componente.

Mediante **Salida** puede añadir una nueva salida al componente.

Mediante **Componente** puede añadir un nuevo componente. El procedimiento es análogo al descrito en "Insertar" "Componente".

Mediante **Asignación** puede insertar una asignación a una variable. En principio se representa ésta mediante tres signos de interrogación "???", que usted puede editar y sustituir por la variable deseada. Para ello está disponible la ayuda de entrada.

Acerca de la posible introducción de direcciones, de la representación del nombre de variable en varias líneas y de un comentario específico para el contacto, consulte más arriba en "Insertar" "Contacto".

### "Insertar" "Salto" en KOP

Mediante esta orden se inserta en el editor KOP un salto en paralelo al final de las bobinas existentes. Si la línea entrante arroja el valor "TRUE", se ejecuta el salto a la marca indicada.

La posición marcada debe ser la conexión entre los contactos y las bobinas o bien una bobina.

El salto recibe como preasignación el texto "Label". Puede hacer clic en este texto y cambiarlo a la marca de salto deseada.

### "Insertar" "Retorno" en KOP

Mediante esta orden se inserta en el editor KOP una instrucción RETURN en paralelo al final de las bobinas existentes. Si la línea entrante arroja el valor "Activado", se cancela el procesamiento del componente en esta red.

La posición marcada debe ser la conexión entre los contactos y las bobinas o bien una bobina.

### "Extras" "Insertar detrás" en KOP

Mediante esta orden se inserta en el editor KOP el contenido del portapapeles como contacto serial detrás de la posición marcada. Esta orden sólo es posible si el contenido del portapapeles y la posición marcada son redes de contactos.

### "Extras" "Insertar debajo" en KOP

**Forma abreviada:** <Ctrl>+<U>

Mediante esta orden se inserta en el editor KOP el contenido del portapapeles como contacto paralelo debajo de la posición marcada. Esta orden sólo es posible si el contenido del portapapeles y la posición marcada son redes de contactos.

### "Extras" "Insertar encima" en KOP

Mediante esta orden se inserta en el editor KOP el contenido del portapapeles como contacto paralelo encima de la posición marcada. Esta orden sólo es posible si el contenido del portapapeles y la posición marcada son redes de contactos.

### "Extras" "Negación" en KOP



Fig. 5-74 : Symbol: "Extras" "Negación"

#### Forma abreviada: <Ctrl>+<N>

Mediante esta orden puede denegar un contacto, una bobina, una instrucción de salto o de RETURN, o una entrada o salida de componentes EN en la posición actual del cursor.

Entre los paréntesis de la bobina o entre las rayas rectas del contacto aparece una barra oblicua (/) y |/, respectivamente. En saltos, retornos, entradas o salidas de componentes EN aparece, al igual que en el editor FUP, un pequeño círculo en la conexión.

La bobina escribe entonces en la variable booleana correspondiente el valor negado de la conexión de entrada. Un contacto negado conmuta el estado de la entrada a la salida cuando la variable booleana correspondiente arroja el valor FALSE.

Si está seleccionado un salto o un retorno, se niega la entrada de dicho salto o retorno.

Una negación puede borrarse mediante una nueva negación.

### "Extras" "Set/Reset" en KOP

Si ejecuta esta orden en una bobina, obtendrá una bobina Set. Tal bobina nunca sobrescribe el valor TRUE en la variable booleana correspondiente. Esto significa que una vez que se ha ajustado el valor de esta variable a TRUE, éste permanece para siempre en TRUE. Una bobina Set se identifica con una "S" en el símbolo de bobina.

Si ejecuta de nuevo esta orden, obtendrá una bobina Reset. Tal bobina nunca sobrescribe el valor FALSE en la variable booleana correspondiente. Esto significa que una vez que se ha ajustado el valor de esta variable a FALSE, éste permanece para siempre en FALSE. Una bobina Reset se identifica con una "R" en el símbolo de bobina.

Si ejecuta esta orden varias veces, esta bobina alterna entre bobina Set, Reset y normal.

### El esquema de contactos en el modo Online

En el modo Online, en el esquema de contactos se colorean en **azul** todos los contactos y las bobinas que se encuentran en el estado "Activado" (TRUE), y también se colorean en azul todas las líneas por las que se transporta "Activado". En las entradas y salidas de bloques de función se indican los **valores** de las variables correspondientes.

Sólo se pueden insertar **breakpoints** en redes; en el **procesamiento por pasos individuales** se salta de una red a otra.

Si está activado el **Control de proceso** ("Online" "Control de proceso"), los campos de número de las redes recorridas se marcan en verde.

Si mantiene el puntero del ratón brevemente sobre una variable, se muestran en un **tooltip** (información de herramientas) el tipo, la dirección y el comentario de la variable.

## El editor de lenguajes de proceso

Éste es el aspecto que presenta un componente escrito en AS en el editor IndraLogic:

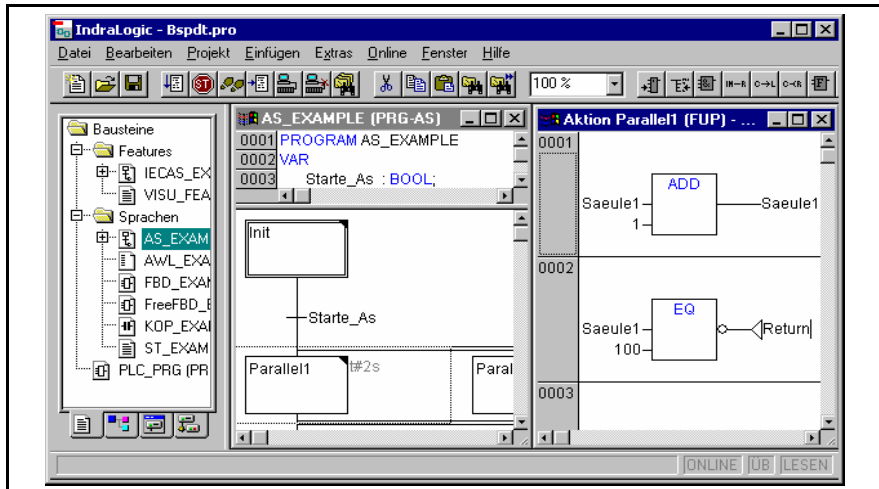


Fig. 5-75 : Componente escrito en AS en el editor IndraLogic

Todos los editores para componentes constan de una parte de declaración y un tronco. Ambas áreas están separadas por un divisor de pantalla.

El editor de lenguajes de proceso es un editor gráfico. Las órdenes más importantes se encuentran en el menú contextual (botón derecho del ratón). Las informaciones de herramientas indican, tanto en el modo Offline como Online y con el zoom activado, los nombres y expresiones completos de pasos, transiciones, saltos, marcas de salto, calificadores o acciones asociadas.

Para hallar información sobre el lenguaje de proceso, consulte el capítulo "Lenguaje de proceso (AS)" en la página 2-18.

El editor de lenguajes de proceso debe concordar con las particularidades del lenguaje AS. Para ello sirven las siguientes opciones de menú:

### Marcar bloques

Un bloque marcado es un conjunto de elementos AS que están rodeados por un rectángulo punteado.

Se puede seleccionar un elemento (un paso, una transición o un salto) colocando el puntero del ratón sobre dicho elemento y pulsando el botón izquierdo del ratón, o bien utilizando las teclas de flecha. Para seleccionar un grupo de varios elementos, pulse <Mayúsculas> para un bloque ya marcado y seleccione el elemento en la esquina inferior izquierda o derecha del grupo. La selección resultante es el grupo coherente mínimo de elementos que contiene estos dos elementos.

¡Tenga en cuenta que sólo puede borrar un paso junto con la transición precedente o posterior!

### "Insertar" "Transición de paso (delante)"



Fig. 5-76 : Símbolo: "Insertar" "Transición de paso (delante)"

**Forma abreviada:** <Ctrl>+<T>

Esta orden inserta en el editor AS un paso seguido de una transición delante del bloque marcado.



### "Insertar" "Transición de paso (detrás)"



Fig. 5-77 : Símbolo: "Insertar" "Transición de paso (detrás)"

#### **Forma abreviada: <Ctrl>+<E>**

Esta orden inserta en el editor AS un paso seguido de una transición detrás de la primera transición en el bloque marcado.

### **Borrar paso y transición**

Sólo se puede borrar un paso junto con la transición precedente o la posterior. Marque para ello el paso y la transición y ejecute la orden "Editar" "Borrar" o pulse la tecla <Supr>.

### "Insertar" "Rama alternativa (derecha)"



Fig. 5-78 : Símbolo: "Insertar" "Rama alternativa (derecha)"

#### **Forma abreviada: <Ctrl>+<A>**

Esta orden inserta en el editor AS una ramificación alternativa como ramificación derecha del bloque marcado. Para ello, el bloque marcado debe empezar y terminar con una transición. La nueva rama consiste entonces en una transición.

### "Insertar" "Rama alternativa (izquierda)"



Fig. 5-79 : Símbolo: "Insertar" "Rama alternativa (izquierda)"

Esta orden inserta en el editor AS una ramificación alternativa como ramificación izquierda del bloque marcado. Para ello, el bloque marcado debe empezar y terminar con una transición. La nueva rama consiste entonces en una transición.

### "Insertar" "Rama paralela (derecha)"



Fig. 5-80 : Símbolo: "Insertar" "Rama paralela (derecha)"

#### **Forma abreviada: <Ctrl>+<L>**

Esta orden inserta en el editor AS una ramificación paralela como ramificación derecha del bloque marcado. Para ello, el bloque marcado debe empezar y terminar con un paso. La nueva rama consiste entonces en un paso. Para posibilitar saltos a la ramificación paralela creada, es preciso dotarla de una marca de salto.

### "Insertar" "Rama paralela (izquierda)"



Fig. 5-81 : Símbolo: "Insertar" "Rama paralela (izquierda)"

Esta orden inserta en el editor AS una ramificación paralela como ramificación izquierda del bloque marcado. Para ello, el bloque marcado debe empezar y terminar con un paso. La nueva rama consiste entonces



en un paso. Para posibilitar saltos a la ramificación paralela creada, es preciso dotarla de una marca de salto.

### "Insertar" "Salto"



Fig. 5-82 : Símbolo: "Insertar" "Salto"

**Forma abreviada: <Ctrl>+<U>**

Esta orden inserta en el editor AS un salto al final de la rama a la que pertenece el bloque marcado. Para ello, la ramificación debe ser una ramificación alternativa.

Para un salto insertado, a continuación es posible seleccionar el texto "Step" introducido y sustituirlo por el nombre del paso o la marca de salto de una ramificación paralela a la que se desea saltar.

### "Insertar" "Salto de transición"



Fig. 5-83 : Símbolo: "Insertar" "Salto de transición"

Esta orden inserta en el editor AS una transición seguida de un salto al final de la ramificación seleccionada. Para ello, la ramificación debe ser una ramificación paralela.

Para un salto insertado, a continuación es posible seleccionar el texto "Step" introducido y sustituirlo por el nombre del paso o la marca de salto de una ramificación paralela a la que se desea saltar.

### "Insertar" "Añadir acción de entrada"

Mediante esta orden puede añadir una acción de entrada a un paso. Una acción de entrada se ejecuta una sola vez, inmediatamente después de la activación del paso. La acción de entrada puede implementarse en cualquier lenguaje.

Un paso con acción de entrada se identifica mediante una "E" en la esquina inferior izquierda.

### "Insertar" "Añadir acción de salida"

Mediante esta orden puede añadir una acción de salida a un paso. Una acción de salida se ejecuta una sola vez, antes de que el paso sea desactivado. La acción de salida puede implementarse en cualquier lenguaje.

Un paso con acción de salida se identifica mediante una "X" en la esquina inferior derecha.

### "Extras" "Insertar rama paralela (derecha)"

Esta orden inserta el contenido del portapapeles como ramificación paralela derecha del bloque marcado. Para ello, el bloque marcado debe empezar y terminar con un paso. El contenido del portapapeles también debe ser un bloque AS que empiece y termine con un paso.

### "Extras" "Añadir marca a la rama paralela"

Para dotar de una marca de salto a una ramificación paralela recién insertada, es preciso marcar la transición situada delante de la ramificación paralela y se debe ejecutar la orden "Añadir marca a la rama paralela". Entonces se dotará a la ramificación paralela de un nombre estándar "Parallel" y un número correlativo adjunto, los cuales pueden editarse conforme a las reglas para nombres identificadores. En el

siguiente ejemplo se sustituyó "Parallel" por "Par\_1\_2" y el salto tras la transición "Fin" se dirigió a esta marca de salto.

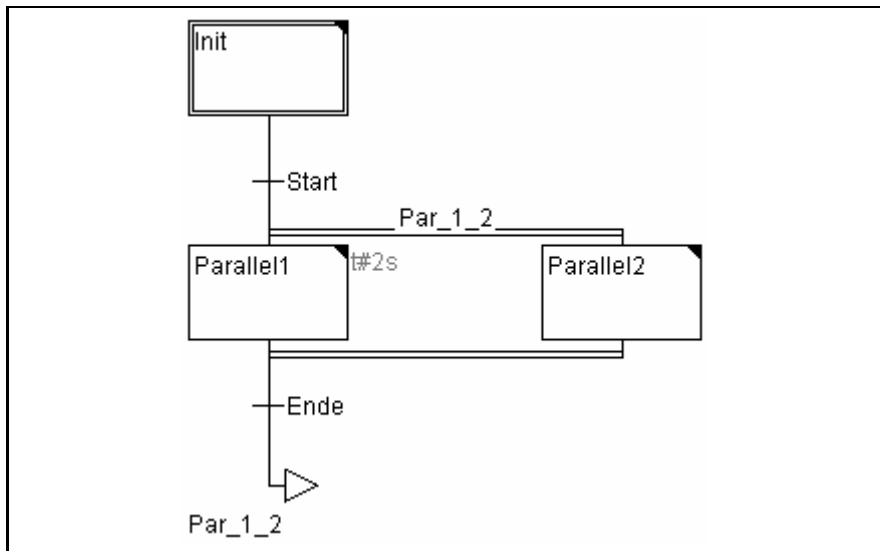


Fig. 5-84 : Añadir marca a la rama paralela

### Borrar marca de salto

Una marca de salto se borra mediante el borrado del texto de la marca de salto.

### "Extras" "Insertar detrás"

Esta orden inserta el bloque AS en el portapapeles detrás del primer paso o de la primera transición del bloque marcado (el copiado normal lo inserta delante del bloque marcado). Esto sólo se ejecuta si la estructura AS resultante es correcta según las normas del lenguaje.

### "Extras" "Zoom acción/transición"

**Forma abreviada: <Alt>+<Intro>**

La acción del primer paso del bloque marcado o del tronco de la transición de la primera transición del bloque marcado se carga en el editor en el lenguaje en el que está escrito. Si la acción o el tronco de la transición están vacíos, se debe seleccionar el lenguaje en el que debe escribirse.

En el caso de las transiciones, tenga en cuenta que la condición escrita en el editor tiene prioridad sobre una condición escrita directamente en la marca de transición. Ejemplo: Si aquí  $i > 100$ , para la condición de transición se aplica: ¡FALSE, pese a que en la marca esté TRUE!

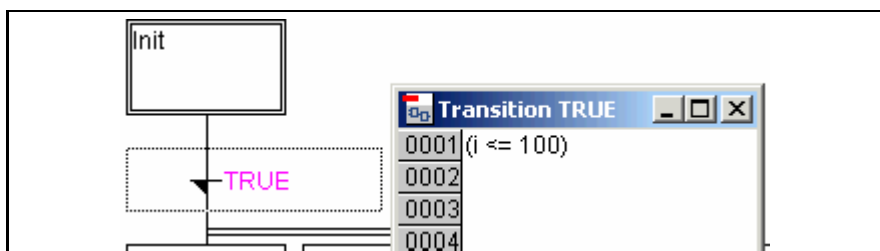


Fig. 5-85 : Prioridad de una condición de transición

### "Extras" "Borrar acción/transición"

Mediante esta orden puede borrar las acciones del primer paso del bloque marcado o la primera transición del bloque marcado.

Si en un paso sólo se implementa la acción, la acción de entrada o la acción de salida, ésta se borra mediante la orden. De lo contrario, aparece un diálogo en el que se puede escoger qué acción o acciones deben borrarse.

Si el cursor se encuentra en una acción de un paso IEC, se borra sólo esta asociación. Si se selecciona un paso IEC con una acción asociada, se borra esta asociación. En caso de un paso IEC con varias acciones, aparece un diálogo para la selección.

### "Extras" "Atributos de paso"

Mediante esta orden se abre un diálogo en el que puede editar atributos para el paso marcado.

Puede realizar tres entradas distintas en el diálogo de atributos de paso. Introduzca en **Tiempo mínimo** el tiempo mínimo que debe durar el procesamiento de este paso. Introduzca en **Tiempo máximo** el tiempo máximo que debe durar el procesamiento del paso. Tenga en cuenta que las entradas son del tipo **TIME**, es decir, utilice una constante TIME (p. ej. T#3s) o una variable del tipo TIME.

En **Comentario** puede introducir un comentario sobre el paso. En el diálogo "Opciones de lenguajes de proceso", que puede abrir mediante "**Extras**" "**Opciones**", puede establecer entonces si en el editor AS se deben mostrar los comentarios o el ajuste de tiempo para sus pasos. A la derecha junto al paso aparece entonces el comentario o los ajustes de tiempo.



Fig. 5-86 : Diálogo para la edición de atributos de paso

Al superarse el tiempo máximo, aparecen flags AS que el usuario puede consultar.



Fig. 5-87 : Tiempo de ejecución de un paso

En el ejemplo se muestra un paso cuya ejecución debe durar un mínimo de dos y un máximo de diez segundos. En el modo Online se indica, además de estos dos tiempos, el tiempo que el paso lleva ya activo.

### "Extras" "Visión general del tiempo"

Mediante esta orden se abre una ventana en la que puede editar los ajustes de tiempo de sus pasos AS:



Fig. 5-88 : Visión general de los límites de tiempo para un componente AS

En la visión general de los límites de tiempo se representan todos los pasos de su componente AS. Si ha especificado una limitación de tiempo para un paso, ésta se indica a la derecha del paso (primero el límite inferior, después el límite superior). Además puede editar las limitaciones de tiempo. Para ello, haga clic sobre el paso deseado en la visión general. Entonces se indica el **Nombre del paso** en la parte inferior de la ventana; acceda al campo **Tiempo mínimo** o **Tiempo máximo** e introduzca allí la limitación de tiempo deseada. Tenga en cuenta que las entradas son del tipo **TIME**, es decir, utilice una constante TIME (p. ej. T#3s) o una variable del tipo TIME. Si cierra la ventana con **OK** se guardan todos los cambios.

En el ejemplo, los pasos 2 y 6 tienen una limitación de tiempo. Cambio1 dura como mínimo dos y como máximo diez segundos. Cambio2 dura como mínimo siete y como máximo ocho segundos.

### "Extras" "Opciones"

Mediante esta orden se abre un diálogo en el que puede ajustar diversas opciones para su componente AS.

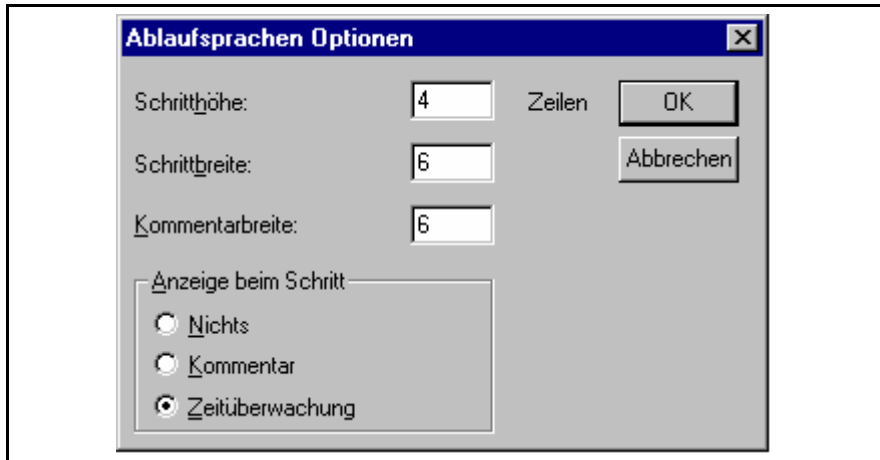


Fig. 5-89 : Diálogo para opciones de lenguajes de proceso

En el diálogo de opciones AS puede realizar cinco entradas. En **Altura del paso** puede especificar cuántas líneas de altura debe tener un paso AS en su editor AS. El ajuste estándar aquí es 4. En **Anchura del paso** puede especificar cuántas columnas de anchura debe tener un paso. El ajuste estándar aquí es 6. La **Anchura del comentario** define el número de columnas que se muestran si se incluye en la visualización el comentario del paso.

En **Mostrar en paso** puede establecer cuáles de las entradas que ha realizado en "Extras" "**Atributos de paso**" se deben mostrar. Puede establecer que no se muestre **Nada**, o bien que se muestre el **Comentario** o la **Vigilancia del tiempo**.

### "Extras" "Asociar acción"

Mediante esta orden puede asociar a pasos IEC acciones y variables booleanas.

A la derecha junto al paso IEC se adjunta una casilla adicional dividida en dos para la asociación de una acción. La casilla está preasignada en el campo izquierdo con el calificador "N" y el nombre "Action". Se pueden modificar ambas preasignaciones. Para ello puede utilizar la ayuda de entrada.

¡A un paso IEC se le pueden asignar un máximo de nueve acciones!

Nuevas acciones para pasos IEC se crean en el Object Organizer para crear un componente AS mediante la orden "**Proyecto**" "**Añadir acción**".

### "Extras" "Utilizar pasos IEC"



Fig. 5-90 : Símbolo: "Extras" "Utilizar pasos IEC"

Si está activada esta orden (reconocible por la marca delante de la opción de menú y por el símbolo pulsado en la barra de funciones), al insertar transiciones de paso y ramas paralelas se insertan pasos IEC en lugar de los pasos simplificados.

Si está seleccionada esta opción, al crear un componente AS se crea el paso Init como paso IEC.



**Nota:** Este ajuste se guarda en el archivo "IndraLogic.ini" y se restablece la próxima vez que se inicie IndraLogic.

### El lenguaje de proceso en el modo Online

En el editor de lenguajes de proceso, en el funcionamiento Online se indican en azul los pasos actualmente activos. Si así lo ha ajustado en "Extras" "Opciones", se muestra junto a los pasos la vigilancia del tiempo. Bajo los límites inferior y superior especificados por usted, aparece una tercera indicación de tiempo que le informa de cuánto tiempo lleva ya activo el paso.

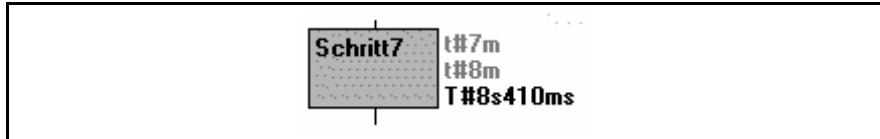


Fig. 5-91 : Tiempo de ejecución de un paso en el modo Online

En la figura superior, el paso representado ya lleva activo 8 segundos y 410 milisegundos. Pero debe estar activo como mínimo 7 minutos antes de salir del paso.

Mediante "Online" "Colocar/suprimir breakpoint" se puede insertar un breakpoint en un paso, y además en una acción en las posiciones permitidas para el lenguaje utilizado. El procesamiento detiene entonces la acción antes de la ejecución de este paso o punto del programa. Los pasos o puntos del programa en los que se ha colocado un breakpoint están marcados en azul claro.

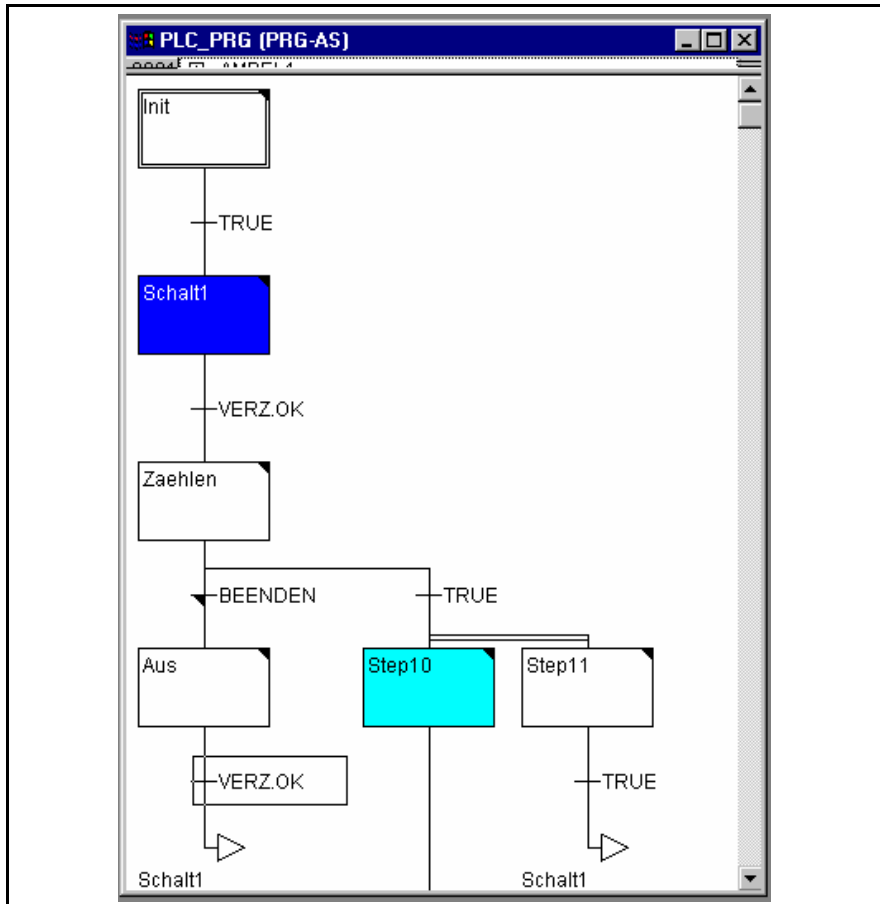


Fig. 5-92 : Lenguaje de proceso en modo Online con un paso activo (Cambio1) y un breakpoint (Step10).



Si en una ramificación paralela están activos varios pasos, se muestra en rojo el paso activo cuya acción será la próxima en procesarse.

Si se han utilizado pasos IEC, en el modo Online se muestran en azul todas las acciones activas.

También en AS es posible el procesamiento por pasos individuales:

Mediante la orden "**Online**" "**Paso individual sobre**" se pasa siempre al siguiente paso cuya acción se está ejecutando. Si la posición actual es:

- un paso en un procesamiento lineal de un componente o un paso en la rama paralela situada más a la derecha en un componente, se sale del componente AS y se vuelve al llamante. Si el componente es el programa principal, se inicia el siguiente ciclo.
- un paso en la rama no situada más a la derecha de una ramificación paralela, se salta al paso activo en la siguiente rama paralela.
- la última posición de breakpoint dentro de una acción, se salta al llamante del SFC.
- la última posición de breakpoint dentro de una acción de entrada o de salida, se salta al primer paso activo.

Mediante "**Online**" "**Paso individual en**" se puede pasar además al interior de acciones. Si se debe saltar al interior de una acción de entrada, de salida o IEC, debe haber un breakpoint colocado allí. Dentro de las acciones, el usuario dispone de todas las capacidades de depuración del editor correspondiente.

Si mantiene el puntero del ratón brevemente sobre una variable en el editor de declaraciones, se muestran en un **tooltip** (información de herramientas) el tipo, la dirección y el comentario de la variable.

---

**Nota:** ¡Si renombra un paso y ejecuta Cambio Online mientras precisamente este paso está activo, el programa se detiene en estado indefinido!

---

#### **Secuencia de procesamiento de los elementos de una cadena de pasos:**

1. En primer lugar se restauran todos los Action Control Block Flags de las acciones IEC que se utilizan en esta cadena de pasos. (Excepto los flags de acciones IEC que se llaman dentro de acciones.)
2. Para todos los pasos se comprueba, en el orden en que se integran en la cadena de pasos (de arriba a abajo y de izquierda a derecha), si se cumple la condición para la ejecución de la acción de salida, y en caso afirmativo se ejecuta ésta.
3. Para todos los pasos se comprueba, en el orden en que se integran en la cadena de pasos, si se cumple la condición para la ejecución de la acción de entrada, y en caso afirmativo se ejecuta ésta.
4. Para todos los pasos se ejecuta lo siguiente, en el orden en que se integran en la cadena de pasos:
  - Si se da el caso, se copia en la variable de paso correspondiente el tiempo transcurrido.
  - Si se da el caso, se comprueba una eventual superación del tiempo y se manejan en consecuencia los flags de error AS.
  - En caso de pasos que no sean IEC, se ejecuta entonces la acción correspondiente.

5. Las acciones IEC que se utilizan en la cadena de pasos se ejecutan en orden alfabético. Esto tiene lugar en dos pasadas por la lista de acciones. En la primera pasada se ejecutan todas las acciones IEC desactivadas en el ciclo actual. En la segunda pasada se ejecutan todas las acciones IEC activas en el ciclo actual.
6. Las transiciones se evalúan: si el paso en el ciclo actual estaba activo y la siguiente transición arroja TRUE (y, cuando sea aplicable, si ya ha transcurrido el tiempo activo mínimo), entonces se activa el siguiente paso.

**Aspectos a tener en cuenta para la implementación de acciones:**

Puede ocurrir que una acción se ejecute varias veces en un ciclo porque está asociada a varias cadenas de pasos. (Por ejemplo, un SFC podría poseer dos acciones IEC A y B, las cuales están implementadas en SFC y llaman la acción IEC C; entonces las acciones IEC A y B pueden estar activas en el mismo ciclo y, a su vez, en ambas acciones IEC puede estar activa la acción IEC C, con lo cual C se llamaría dos veces).

Si se utiliza la misma acción IEC simultáneamente en diversos niveles de un SFC, esto podría dar lugar a efectos indeseados debido a la secuencia de procesamiento anteriormente descrita. Por este motivo, en este caso se emite un mensaje de error. ¡Esto podría suceder durante el procesamiento de proyectos creados con versiones antiguas de IndraLogic!

**Nota:** Durante la monitorización de expresiones (p. ej. A AND B) en transiciones, se indica sólo el "valor total" de la transición.

**El Editor de esquemas de funciones continuo (CFC)**

Éste es el aspecto que presenta un componente creado mediante el editor de esquemas de funciones continuo (CFC):

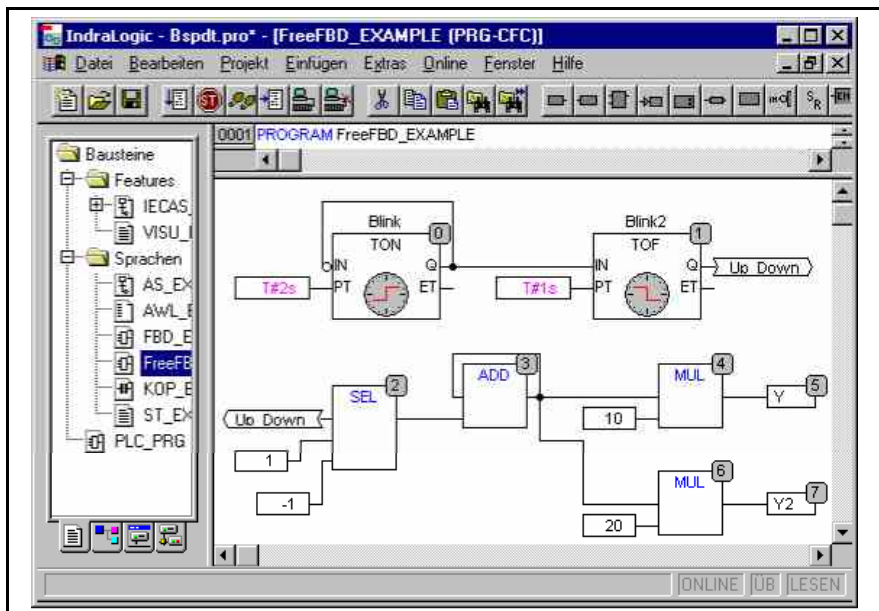


Fig. 5-93 : Componente escrito en CFC en el editor IndraLogic

En el editor de esquemas de funciones continuo no se utilizan redes, sino que se pueden colocar libremente los elementos. Entre los elementos de la lista de procesamiento se cuentan componente, entrada, salida, salto,

etiqueta, retorno y comentario. Las entradas y salidas de estos elementos pueden conectarse estirando una conexión mediante el ratón. La línea de conexión se dibuja automáticamente. Se dibuja la línea de conexión más corta posible, teniendo en cuenta las conexiones existentes. Al desplazar elementos, se adaptan automáticamente las líneas de conexión. Si por motivos de espacio no se puede dibujar una línea de conexión, se muestra una línea roja entre la entrada y la salida correspondiente. En cuanto existe espacio suficiente, se transforma esta línea en una línea de conexión.

Una ventaja del editor de esquemas de funciones continuo sobre el editor de esquemas de funciones FUP convencional es que se pueden insertar directamente realimentaciones.

Las órdenes más importantes se encuentran en el menú contextual.

### Posiciones del cursor

Cada texto es una posible posición del cursor. El texto seleccionado se muestra con fondo azul y puede ser modificado.

Por lo demás, la posición actual del cursor está identificada por un rectángulo punteado. A continuación se enumeran todas las posiciones del cursor posibles con ejemplos:

1. Troncos de los elementos componente, entrada, salida, salto, etiqueta, retorno y comentario:

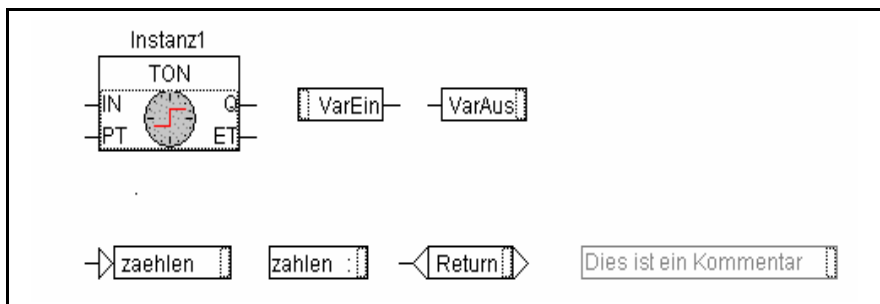


Fig. 5-94 : Posición del cursor Tronco de los elementos

2. Campos de texto de los elementos componente, entrada, salida, salto, etiqueta, retorno y comentario, además de los campos de texto de las marcas de conexión:

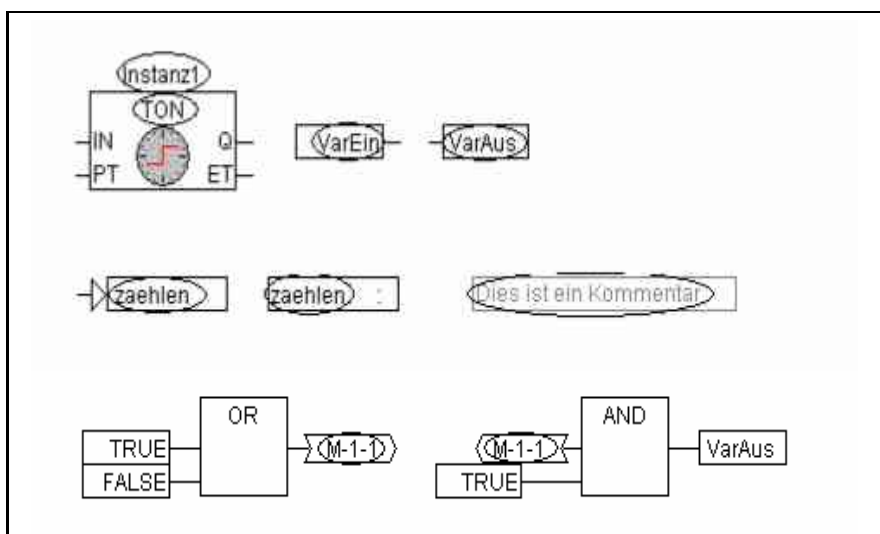


Fig. 5-95 : Posición del cursor Campos de texto

3. Entradas de los elementos componente, salida, salto y retorno:

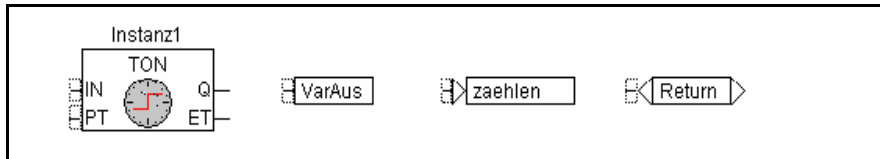


Fig. 5-96 : Posición del cursor Entradas

4. Salidas de los elementos componente y entrada:

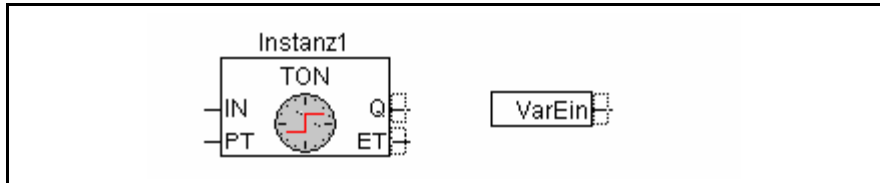


Fig. 5-97 : Posición del cursor Salidas

**"Insertar" "Componente"**

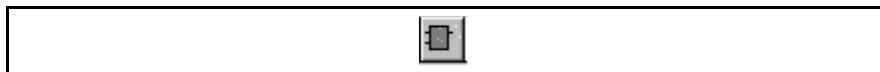


Fig. 5-98 : Símbolo: "Insertar" "Componente"

**Forma abreviada: <Ctrl>+<B>**

Mediante esta orden se pueden insertar operadores, funciones, bloques de función y programas. En primer lugar se inserta siempre un operador "AND". Seleccionando y sobrescribiendo el texto, dicho operador puede transformarse en cualquier otro operador, en cualquier función, en cualquier bloque de función y en cualquier programa. Mediante la ayuda de entrada puede seleccionar el componente deseado de la lista de componentes soportados. Si el nuevo componente tiene otra cantidad mínima de entradas, se adjuntan éstas. Si el nuevo componente tiene una cantidad máxima de entradas menor, se borran las últimas entradas.

**"Insertar" "Entrada"**



Fig. 5-99 : Símbolo: "Insertar" "Entrada"

**Forma abreviada: <Ctrl> + <E>**

Mediante esta orden se inserta una entrada. Es posible seleccionar el texto introducido "???" y sustituirlo por una variable o constante. Para ello también puede utilizar la ayuda de entrada.

**"Insertar" "Salida"**



Fig. 5-100 : Símbolo: "Insertar" "Salida"

**Forma abreviada: <Ctrl>+<A>**

Mediante esta orden se inserta una salida. Es posible seleccionar el texto introducido "???" y sustituirlo por una variable. Para ello también puede utilizar la ayuda de entrada. Se asigna a esta variable el valor asociado a la entrada de la salida.



### "Insertar" "Salto"



Fig. 5-101 : Símbolo: "Insertar" "Salto"

**Forma abreviada: <Ctrl>+<J>**

Mediante esta orden se inserta un salto. Es posible seleccionar el texto "???" introducido y sustituirlo por la marca de salto a la que se desea saltar.

La marca de salto se inserta con la orden "Insertar" "Marca".

### "Insertar" "Marca"



Fig. 5-102 : Símbolo: "Insertar" "Salto"

**Forma abreviada: <Ctrl>+<L>**

Mediante esta orden se inserta una marca de salto. Es posible seleccionar el texto introducido "???" y sustituirlo por la marca de salto. En el modo Online se inserta automáticamente una etiqueta RETURN para marcar el final del componente.

El salto se inserta con la orden "Insertar" "Salto".

### "Insertar" "Retorno"



Fig. 5-103 : Símbolo: "Insertar" "Retorno"

**Forma abreviada: <Ctrl> + <R>**

Mediante esta orden se inserta una instrucción RETURN. Tenga en cuenta que en el modo Online se inserta automáticamente una marca de salto con la designación RETURN en la primera columna y detrás del último elemento en el editor, a la que se salta durante el procesamiento por pasos individuales antes de salir del componente.

### "Insertar" "Comentario"

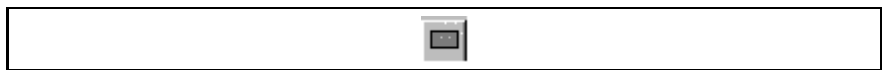


Fig. 5-104 : Símbolo: "Insertar" "Comentario"

**Forma abreviada: <Ctrl> + <K>**

Mediante esta orden se inserta un comentario.

Mediante <Ctrl> + <Intro> obtendrá una nueva línea dentro del comentario.

### "Insertar" "Entrada de componente"

**Forma abreviada: <Ctrl> + <U>**

Esta orden inserta una entrada de componente. En muchos operadores, el número de entradas es variable (p. ej. ADD puede tener 2 o más entradas).

Para incrementar en uno el número de entradas en tal operador, es preciso seleccionar el propio operador (posición del cursor 1).

### "Insertar" "In-Pin", "Insertar" "Out-Pin"



Fig. 5-105 : Símbolo: "Insertar" "In-Pin"



Fig. 5-106 : Símbolo: "Insertar" "Out-Pin"

Estas órdenes están disponibles en cuanto se abre una macro para la edición. Sirven para la inserción de In-Pins y Out-Pins como entradas y salidas de la macro. Se diferencian de las entradas y salidas normales de los componentes por la forma de representación y por el hecho de que no reciben ningún índice de posición.

### "Extras" "Negar"



Fig. 5-107 : Símbolo: "Extras" "Negar"

#### Forma abreviada: <Ctrl> + <N>

Mediante esta orden puede denegar entradas, salidas, saltos o instrucciones RETURN. El símbolo de la negación es un pequeño círculo en una conexión.

Si está seleccionada la entrada de un componente, salida, salto o retorno (posición del cursor 3), se deniega dicha entrada.

Si está seleccionada la salida de un componente o entrada (posición del cursor 4), se deniega dicha salida.

Una negación puede borrarse mediante una nueva negación.

### "Extras" "Set/Reset"

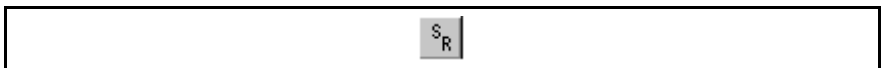


Fig. 5-108 : Símbolo: "Extras" "Set/Reset"

#### Forma abreviada: <Ctrl> + <T>

Esta orden sólo puede ejecutarse para entradas seleccionadas del elemento salida (posición del cursor 3).

El símbolo para Set es S y para Reset es R.

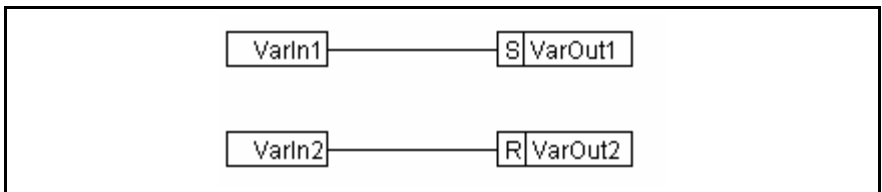


Fig. 5-109 : Ejemplo de "Set/Reset"

VarOut1 se ajusta a TRUE en caso de que VarIn1 arroje TRUE. VarOut1 conserva este valor aunque VarIn1 vuelva a saltar a FALSE.

VarOut2 se ajusta a FALSE en caso de que VarIn2 arroje TRUE. VarOut2 conserva este valor aunque VarIn2 vuelva a saltar a FALSE.

Si se ejecuta esta orden varias veces, la salida alterna entre los estados Set, Reset y normal.

**"Extras" "EN/ENO"**



Fig. 5-110 : Símbolo: "Extras" "EN/ENO"

Mediante esta orden, un componente seleccionada (posición del cursor 3) recibe una entrada de habilitación booleana adicional EN (Enable In) y una salida booleana ENO (Enable Out).

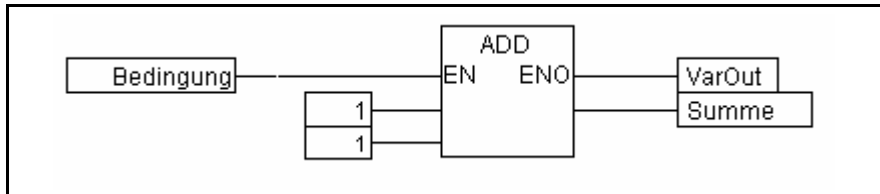


Fig. 5-111 : Inserción de una entrada EN y una salida ENO

En este ejemplo (Fig. 5-111 ) sólo se ejecuta ADD si la condición de la variable booleana es TRUE. En ese caso, también se ajusta VarOut a TRUE tras la ejecución de ADD. ¡En caso de que la condición de la variable cambie a FALSE, ya no se ejecutará ADD y VarOut conservará el valor TRUE! El siguiente ejemplo muestra cómo se puede utilizar el valor de ENO para otros componentes.

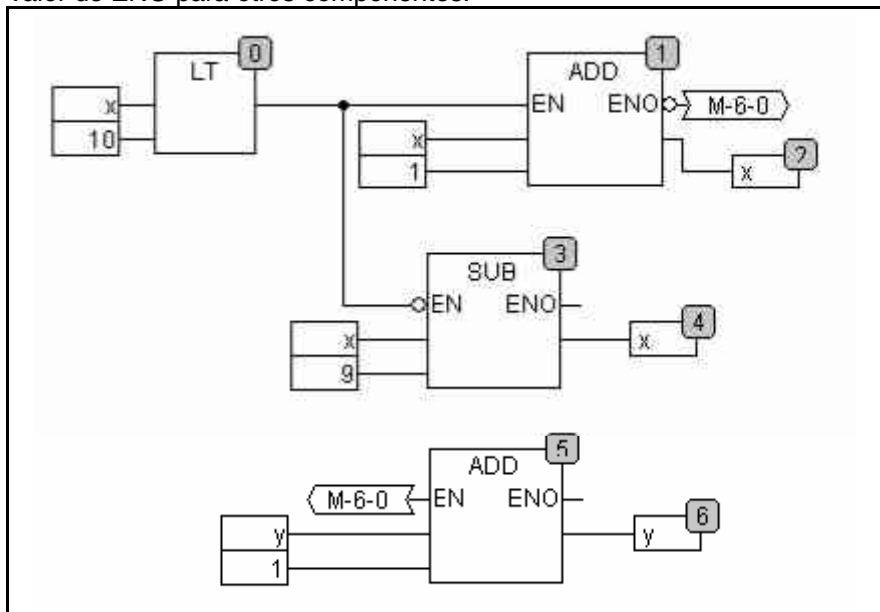


Fig. 5-112 : Ejemplo de CFC

Para ello, x debe estar inicializado con 1 e y con 0. Los números en la esquina derecha de los componentes indican la secuencia de procesamiento.

x se incrementa en uno hasta que adopta el valor 10. Dado que entonces la salida del componente arroja LT(0) FALSE, se ejecutan SUB(3) y ADD(5). De este modo, se ajusta x al valor 1 y se incrementa y en 1. A continuación se ejecuta de nuevo LT(0) mientras x sea menor que 10. Así pues, y cuenta las veces que x recorre los valores 1 hasta 10.

**"Extras" "Propiedades... "**

En el Editor de esquemas de funciones continuo no se muestran directamente parámetros de entrada constantes (VAR\_INPUT CONSTANT) de funciones y bloques de función. Éstos pueden ser visualizados y su valor modificado si se selecciona el tronco del

componente en cuestión (posición del cursor 1) y se ejecuta la orden "Extras" "Propiedades..." o simplemente se hace doble clic sobre el tronco. Se abre el diálogo Editar parámetros:

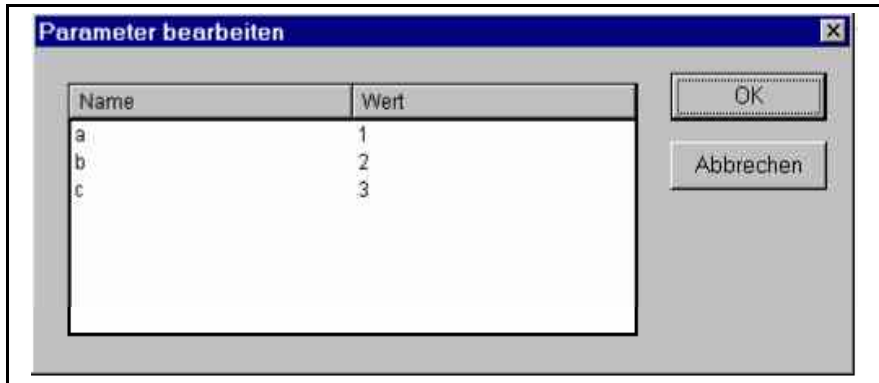


Fig. 5-113 : Diálogo de propiedades

Si el valor de parámetro está marcado en la columna Valor, puede ser editado tras hacer de nuevo clic con el ratón o tras pulsar la barra espaciadora. La modificación de un valor se confirma pulsando la tecla <Intro>, mientras que pulsando la tecla <Escape> se desechan los cambios. Mediante el botón **OK** se guardan todos los cambios.

**Nota:** Esta función, y por ende la declaración con palabra clave "VAR\_INPUT CONSTANT" sólo es relevante para el editor CFC. En el editor FUP se indican siempre todas las variables INPUT en el componente, independientemente de si la declaración se ha realizado como VAR\_INPUT o como VAR\_INPUT CONSTANT. Esto tampoco afecta a los editores de texto. VAR\_INPUT CONSTANT sólo es válido para la utilización en CFC.

### Seleccionar elementos

Para seleccionar un elemento, haga clic con el ratón sobre el tronco del elemento (posición del cursor 1).

Para seleccionar varios elementos, pulse la tecla <Mayúsculas> y haga clic con el ratón sucesivamente sobre los elementos en cuestión o arrastre una ventana sobre los elementos a seleccionar mientras mantiene pulsado el botón izquierdo del ratón.

Mediante la orden "**Extras**" "**Marcar todo**" puede seleccionar todos los elementos.

### Desplazar elementos

Se pueden desplazar uno o varios elementos seleccionados mediante las teclas de flecha manteniendo pulsada la tecla <Mayúsculas>. Otra posibilidad consiste en desplazar los elementos mientras se mantiene pulsado el botón izquierdo del ratón. Estos elementos se depositan soltando el botón izquierdo del ratón, siempre y cuando no solapen otros elementos ni superen el tamaño previsto del editor. En este caso, los elementos marcados vuelven a su posición original y suena un aviso.

### Copiar elementos

Uno o varios elementos seleccionados se copian mediante "**Editar**" "**Copiar**" y se insertan mediante "**Editar**" "**Insertar**".

### Crear conexiones

Una entrada de un elemento puede conectarse con una sola salida de otro elemento. Una salida de un elemento puede conectarse con varias entradas de elementos.

Existen varias posibilidades para conectar una entrada de un elemento E2 con la salida de un elemento E1.



Fig. 5-114 : Elementos que se desea conectar

Hacer clic con el botón izquierdo del ratón sobre la salida del elemento E1 (posición del cursor 4), mantener pulsado el botón izquierdo del ratón, arrastrar el puntero del ratón hasta la entrada del elemento E2 (posición del cursor 3) y allí soltar el botón izquierdo del ratón. Durante el proceso de arrastre con el ratón se dibuja una conexión desde la salida del elemento E1 hasta el puntero del ratón.

Hacer clic con el botón izquierdo del ratón sobre la entrada del elemento E2, mantener pulsado el botón izquierdo del ratón, arrastrar el puntero del ratón hasta la salida del elemento E1 y allí soltar el botón izquierdo del ratón.

Desplazar uno de los elementos E1 o E2 (posición del cursor 1) y, soltando el botón izquierdo del ratón, depositarlo de forma que la salida del elemento E2 y la entrada del elemento E1 se toquen.

En caso de que el elemento E2 sea un componente con una entrada libre, mediante el ratón también se puede crear una conexión desde una entrada de E1 hasta el tronco de E2. Al soltar el botón del ratón se establece automáticamente una conexión con la entrada libre de E2 situada en una posición más alta. Si el componente E2 carece de entrada libre pero es un operador cuyo número de entradas puede incrementarse en uno, se crea automáticamente una nueva entrada.

Por medio de estos métodos también se pueden conectar entre sí la salida y la entrada de un componente (realimentación). Para crear una conexión entre dos pins, haga clic con el botón izquierdo del ratón sobre un pin, mantenga el botón pulsado y arrastre la conexión hasta el pin deseado, donde deberá soltar el botón. Si durante el arrastre de la conexión se sale del área de trabajo del editor, se produce un desplazamiento automático. Para tipos de datos sencillos tiene lugar una comprobación del tipo durante la conexión. Si los tipos de ambos pins no son compatibles, el cursor cambia a "prohibido". Para tipos de datos complejos no tiene lugar ninguna comprobación.

### Cambiar conexiones

Una conexión entre la salida de un elemento E1 y la entrada de un elemento E2 puede cambiarse fácilmente a una conexión entre la salida de E1 y una entrada de un elemento E3. Para ello se hace clic con el ratón sobre la entrada de E2 (posición del cursor 3) mientras se mantiene apretado el botón izquierdo del ratón, se desplaza el puntero del ratón hasta la entrada de E3 y se suelta allí.

### Borrar conexiones

Existen varias posibilidades para borrar una conexión entre la salida de un elemento E1 y una entrada de un elemento E2:

Seleccione la salida de E1 (posición del cursor 4) y pulse la tecla <Supr> o ejecute la orden "**Editar**" "**Borrar**". Si la salida de E1 está conectada a varias entradas, se borran varias conexiones.



Seleccione la entrada de E2 (posición del cursor 4) y pulse la tecla <Supr> o ejecute la orden "Editar" "Borrar".

Seleccione con el ratón la entrada de E2 mientras mantiene apretado el botón izquierdo del ratón y arrastre la conexión retirándola de la entrada de E2. Si entonces se suelta el botón izquierdo del ratón en un área libre, se borra la conexión.

### "Extras" "Marca de conexión"

En lugar de mediante líneas de conexión, las conexiones también se pueden representar mediante conectores (marcas de conexión). Para ello, la salida y la entrada correspondiente son dotadas de un conector que tiene un nombre unívoco.

Si ya existe una conexión entre dos elementos que ahora se desea visualizar en la representación de conectores, en primer lugar se marca la salida de la línea de conexión (posición del cursor 3) y se selecciona la opción de menú "Extras" "Marca de conexión". La representación subsiguiente muestra una conexión antes y después de la selección de dicha opción de menú.

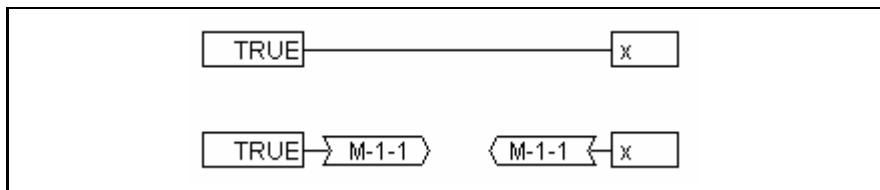


Fig. 5-115 : Conexión antes y después de la selección de "Extras" "Marca de conexión"

El programa asigna como estándar un nombre de conector unívoco que empieza por M pero puede ser modificado. El nombre del conector se guarda como parámetro de la salida, pero puede editarse tanto en la entrada como en la salida:

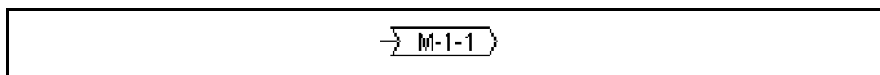


Fig. 5-116 : Edición del nombre del conector en la salida

Si se sustituye el texto en el conector, el nuevo nombre del conector es adoptado por todos los conectores asociados en las entradas. Sin embargo, no se puede escoger un nombre que ya pertenezca a **otra** marca de conexión, dado que ello vulneraría la unicidad del nombre del conector. En este caso se emite el mensaje pertinente.

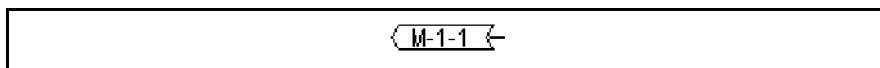


Fig. 5-117 : Edición del nombre del conector en la entrada

Si se sustituye el texto en el conector, se sustituye también en la marca de conexión correspondiente en el otro componente. Las conexiones en representación de conector pueden convertirse de nuevo en conexiones normales, marcando para ello las salidas de las conexiones (posición del cursor 4) y seleccionando de nuevo la opción de menú "Extras" "Marca de conexión".

### Insertar inputs/outputs "al vuelo"

Si está seleccionado un solo pin de input o de output de un elemento, mediante la introducción de una secuencia de caracteres con el teclado



se puede insertar directamente el elemento de input u output correspondiente y llenar su campo del editor con la secuencia de caracteres.

### **Secuencia de procesamiento**

En el editor de esquemas de funciones continuo CFC, los elementos componente, salida, salto, retorno y etiqueta reciben cada uno un número de procesamiento. En este orden se ejecutan los distintos elementos en el tiempo de ejecución.

Al insertar un elemento, el número se asigna automáticamente en orden topológico (de izquierda a derecha y de arriba a abajo). Si ya se ha modificado el orden, el nuevo elemento recibe el número de su sucesor topológico y todos los números más altos se incrementan en uno.

Al desplazar un elemento se mantiene el número.

El orden influye en el resultado y en determinados casos debe ser modificado.

Si se muestra el orden, en los elementos aparece en la esquina superior derecha el número de procesamiento correspondiente.

### **"Secuencia" "Mostrar"**

Mediante esta orden se activa o desactiva en el editor CFC la indicación de la secuencia de procesamiento. El ajuste predeterminado es mostrar la secuencia de procesamiento (reconocible por la marca delante de la opción de menú).

En los elementos componente, salida, salto, retorno y etiqueta aparece su número de procesamiento correspondiente en la esquina superior derecha.

### **"Extras" "Secuencia" "Ordenar topológicamente"**

Los elementos están ordenados topológicamente en el editor CFC si el procesamiento tiene lugar de izquierda a derecha y de arriba a abajo, es decir, en caso de elementos ordenados topológicamente, el número se incrementa de izquierda a derecha y de arriba a abajo. Las conexiones son irrelevantes a este respecto. Sólo cuenta la posición de los elementos.

Si se ejecuta la orden **"Extras" "Secuencia" "Ordenar topológicamente"**, se ordenan topológicamente todos los elementos **seleccionados**. Todos los elementos de la selección se toman de la lista de procesamiento. A continuación, los elementos de la selección se insertan individualmente de nuevo desde abajo a la derecha hacia arriba a la izquierda en la lista de procesamiento remanente. Cada elemento marcado se inserta en la lista de procesamiento delante del sucesor topológico, es decir, se inserta delante del elemento que en un orden topológico se procesaría a continuación si todos los elementos del editor estuvieran dispuestos en orden topológico. Esto se ilustra mediante el siguiente ejemplo:

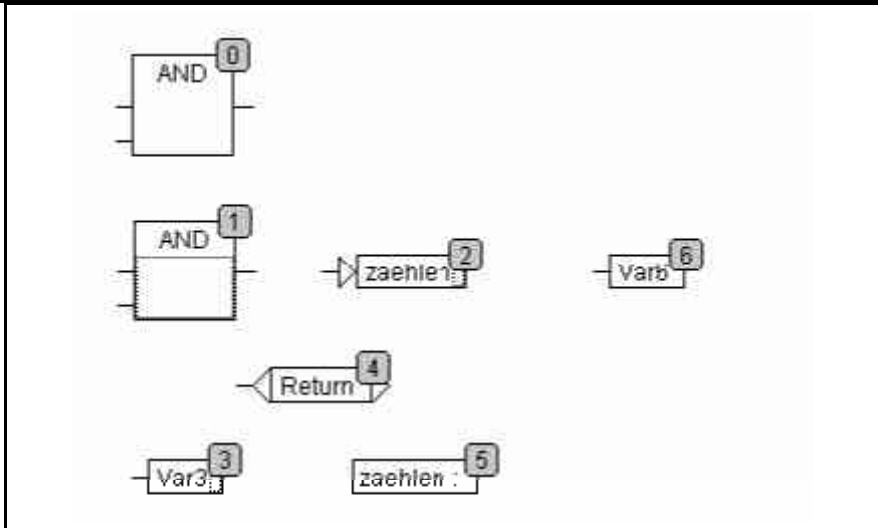


Fig. 5-118 : Orden topológico

Los elementos con el número 1, 2 y 3 están seleccionados. Si ahora se ejecuta la orden "**Ordenar topológicamente**", en primer lugar se toman de la lista de procesamiento los tres elementos seleccionados. A continuación se insertan de nuevo sucesivamente Var3, el salto y el operador AND. Var3 se ordena delante de la etiqueta y recibe el número 2. A continuación se ordena el salto, el cual recibe en principio el 4, y tras la inserción del AND el 5. Resulta la siguiente nueva secuencia de procesamiento:

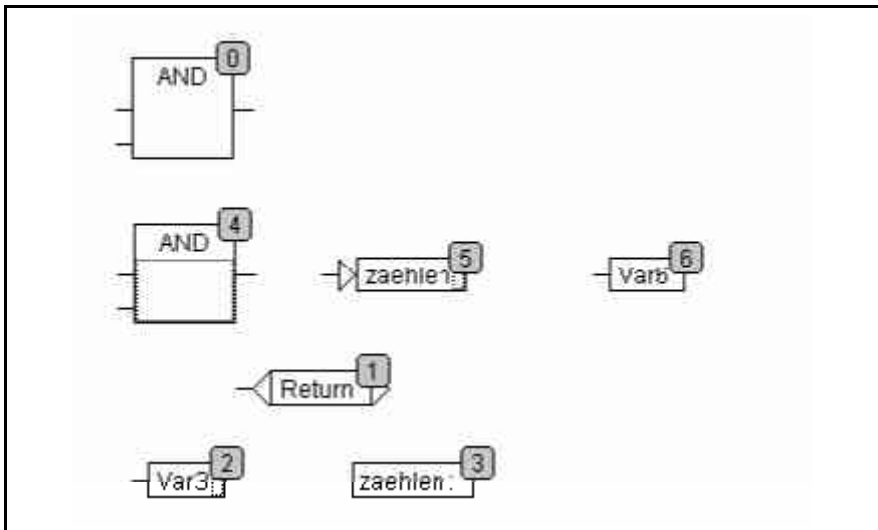


Fig. 5-119 : Nueva secuencia de procesamiento

Al guardar un componente de nueva creación, como ajuste estándar éste se ordena delante de su sucesor topológico en la lista de procesamiento.

### "Extras" "Secuencia" "Uno adelante"

En el editor CFC, mediante esta orden, todos los elementos seleccionados, excepto el elemento que se encuentra al principio de la secuencia de procesamiento, son desplazados una posición hacia delante dentro de la secuencia de procesamiento.

### "Extras" "Secuencia" "Uno atrás"

En el editor CFC, mediante esta orden, todos los elementos seleccionados, excepto el elemento que se encuentra al final de la secuencia de procesamiento, son desplazados una posición hacia atrás dentro de la secuencia de procesamiento.

### "Extras" "Secuencia" "Al principio"

En el editor CFC, mediante esta orden, todos los elementos seleccionados son desplazados hasta el principio de la secuencia de procesamiento, manteniéndose el orden dentro de los elementos seleccionados. Asimismo, se conserva el orden dentro de los elementos no seleccionados.

### "Extras" "Secuencia" "Al final"

En el editor CFC, mediante esta orden, todos los elementos seleccionados son desplazados hasta el final de la secuencia de procesamiento, manteniéndose el orden dentro de los elementos seleccionados. Asimismo, se conserva el orden dentro de los elementos no seleccionados.

### "Extras" "Secuencia" "Ordenar todo según el flujo de datos"

Esta orden se aplica a **todos** los elementos en el editor CFC. La secuencia de procesamiento es determinada por el flujo de datos de los elementos y no por su posición.

La siguiente figura muestra elementos ordenados topológicamente.

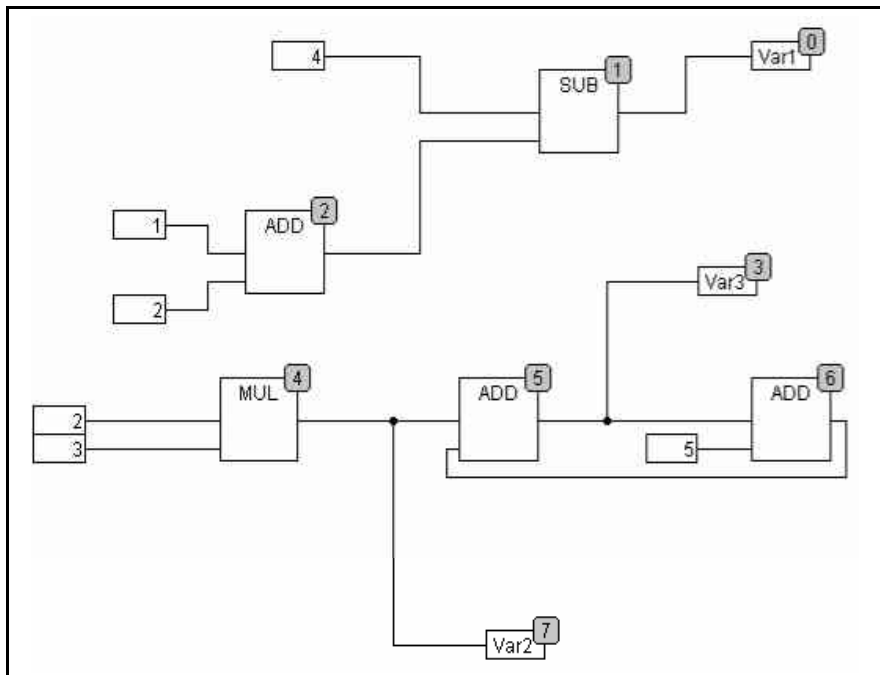


Fig. 5-120 : Ejemplo de orden topológico

Al ejecutar la orden se obtiene la siguiente secuencia:

5-66 Los editores

IndraLogic

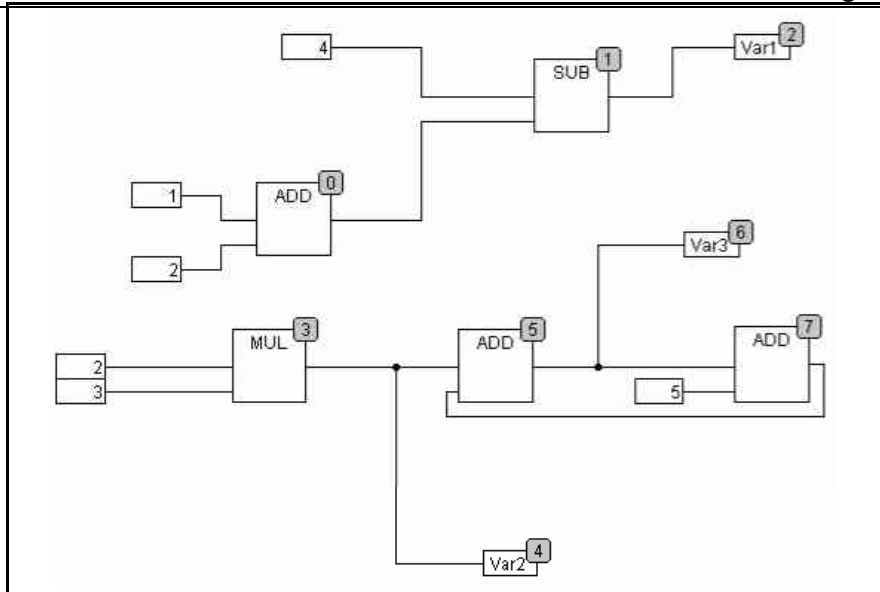


Fig. 5-121 : Nueva secuencia tras ejecutar "Extras" "Secuencia" "Ordenar todo según el flujo de datos"

Al ejecutar la orden, en primer lugar se ordenan todos los elementos topológicamente. A continuación se crea una nueva lista de procesamiento. A partir de los valores conocidos de las entradas, se determina cuáles de los elementos todavía no numerados pueden ser procesados a continuación. Por ejemplo, en la "red" superior se puede procesar inmediatamente el componente ADD, dado que los valores presentes en sus entradas (1 y 2) son conocidos. Sólo a continuación se puede procesar el componente SUB, dado que antes se debe conocer el resultado de ADD, y así sucesivamente.

En cualquier caso, las realimentaciones se insertan en último lugar.

La ventaja de la secuencia conforme al flujo de datos reside en el hecho de que una caja de salida conectada a la salida de un componente sigue inmediatamente a ésta en la secuencia conforme al flujo de datos, lo cual no siempre es el caso en el orden topológico. Así pues, en determinadas circunstancias la secuencia topológica arroja un resultado distinto a la secuencia según el flujo de datos, como se observa en los ejemplos anteriores.

**"Extras" "Crear macro"**



Fig. 5-122 : Symbol: "Extras" "Crear macro"

Mediante esta orden puede agrupar varios componentes seleccionados simultáneamente en el editor CFC creando un bloque que puede ser dotado de un nombre como macro. Las macros sólo se pueden reproducir mediante copiar/pegar, y cada copia constituye una macro propia cuyo nombre puede escogerse independientemente. Así pues, las macros no son referencias. Todas las conexiones que se "cortan" debido a la creación de la macro generan in-pins y out-pins en la macro. Las conexiones a inputs generan un in-pin. Como nombre junto al pin aparece un nombre por defecto con la forma In<n>. Para conexiones a outputs aparece Out<n>. Las conexiones afectadas que tenían marcas de conexión antes de la creación de la macro conservan la marca de conexión en el PIN de la macro.

Una macro recibe en principio el nombre por defecto "MACRO". Este nombre se puede modificar en el campo de nombre de la utilización de la macro. Si se edita la macro, se muestra el nombre de la macro en la



barra de título de la ventana del editor, adjuntado al nombre del componente.

**Ejemplo:**

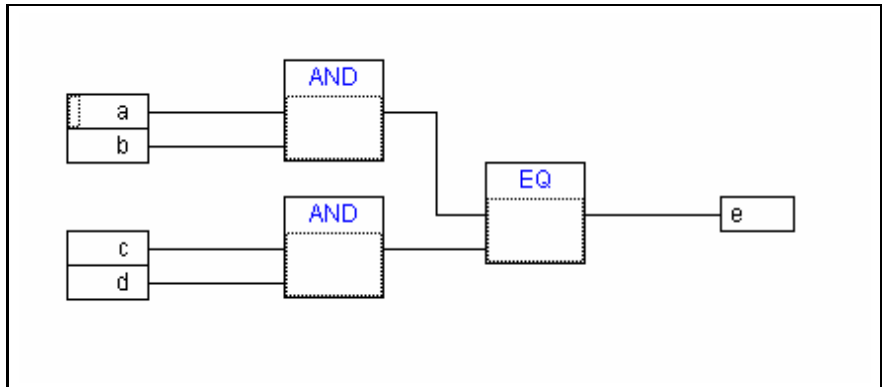


Fig. 5-123 : Selección

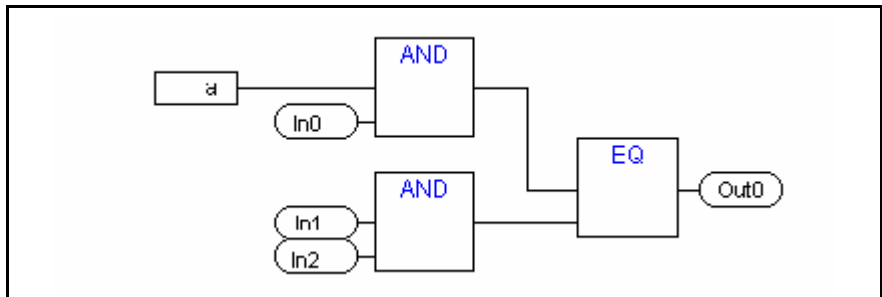


Fig. 5-124 : Macro

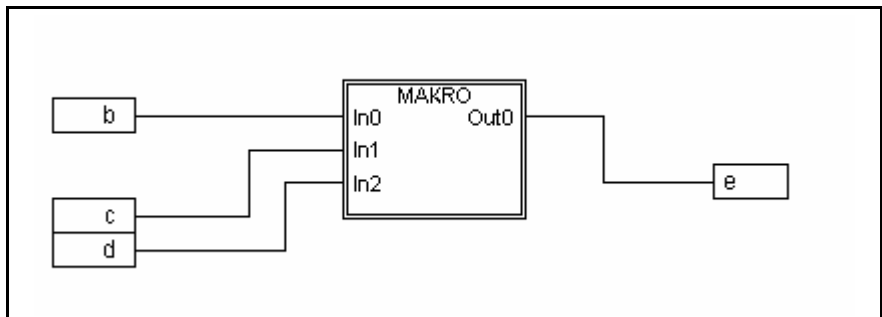


Fig. 5-125 : En el editor

**"Extras" "Saltar a macro"**



Fig. 5-126 : Símbolo: "Extras" "Saltar a macro"

Mediante esta orden o bien haciendo doble clic sobre el tronco de la macro en el editor CFC se abre para la edición la macro en la ventana del editor del componente correspondiente. El nombre de la macro se adjunta al nombre del componente en la barra del título.

Las cajas de pin generadas para las entradas y salidas de la macro durante la creación pueden ser desplazadas, borradas, añadidas, etc., al igual que las entradas y salidas de componentes normales. Sólo se diferencian en la representación y carecen de índice de posición. Para

añadir puede utilizar los botones (entrada) y (salida), los cuales se ofrecen en la barra de símbolos. Las cajas de pin poseen esquinas

redondeadas. El texto de la caja de pin coincide con el nombre del pin en la representación de macro.

El orden de los pins en la caja de macro se rige por la secuencia de procesamiento de los elementos de la macro. Un índice de secuencia bajo delante de uno alto, un pin en posición elevada delante de uno en posición baja.

La secuencia de procesamiento dentro de la macro está cerrada, es decir, la macro se procesa como un bloque, concretamente en la posición de la macro en el componente de orden superior. Así pues, las órdenes para la manipulación de la secuencia sólo operan dentro de la macro.

### "Extras" "Expandir macro"

Mediante esta orden se expande de nuevo la macro seleccionada en el editor CFC y se insertan los elementos contenidos en la posición de la macro en el componente. Las conexiones a los pins de la macro se representan de nuevo como conexiones a las entradas y salidas de los elementos. Si por falta de espacio no pudiera tener lugar la expansión de la macro en la posición de la caja de macro, se desplaza la macro hacia la derecha y hacia abajo hasta que se dispone del espacio suficiente.

**Nota:** Si se almacena el proyecto como versión del proyecto 2.1, se expanden también todas las macros. También se expanden todas las macros antes de la conversión a otros lenguajes.

### "Extras" "Retroceder un nivel de macro", "Extras" "Retroceder todos los niveles de macro"



Fig. 5-127 : Símbolo: "Extras" "Retroceder un nivel de macro"



Fig. 5-128 : Símbolo: "Extras" "Retroceder todos los niveles de macro"

Estas órdenes están disponibles también en la barra de símbolos en cuanto se abre una macro para la edición en el editor CFC. Si las macros están encajadas unas dentro de otras, se puede conmutar de nuevo al nivel de representación inmediatamente superior o al más alto.

### Realimentaciones

A diferencia del editor de esquemas de funciones convencional, en el Editor de esquemas de funciones continuo se pueden representar directamente realimentaciones. En este contexto, se debe tener en cuenta que para la salida de un componente se crea generalmente una variable intermedia interna. En los operadores, el tipo de dato de la variable intermedia se deriva del tipo de dato más grande de las entradas.

El tipo de dato de una constante se determina a partir del tipo de dato más pequeño posible, esto es, para la constante "1" se adopta el tipo de dato SINT. Si a continuación se efectúa una adición con realimentación y la constante "1", la primera entrada arroja el tipo de dato SINT y la segunda es indefinida debido a la realimentación. De este modo, la variable intermedia también es del tipo SINT. Sólo posteriormente se asigna el valor de la variable intermedia a las variables de salida.

La figura **Fig. 5-129** a continuación muestra una adición con realimentación y una directamente con una variable. Para ello, las variables x e y deben ser del tipo INT.

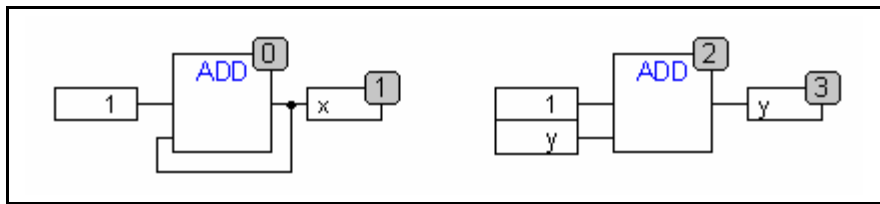


Fig. 5-129 : Adición con realimentación y adición con una variable

Existen diferencias entre ambas sumas:

La variable y puede inicializarse con un valor distinto a 0, pero no así la variable intermedia de la adición izquierda.

La variable intermedia de la adición izquierda tiene el tipo de datos SINT, y la de la derecha el tipo de datos INT. Las variables x e y tienen valores distintos a partir de la 129ª llamada. La variable x, pese a ser del tipo INT, recibe el valor -127, dado que la variable intermedia presenta un desbordamiento. En cambio, la variable y recibe el valor 129.

### CFC en modo Online

#### Monitorización:

Los valores para entradas y salidas se representan dentro de las cajas de input y output, respectivamente. Las constantes no se monitorizan. Para variables no booleanas se amplían las cajas conforme a los valores indicados. Para conexiones booleanas, tanto el nombre de la variable como la conexión se muestran en azul si el valor es TRUE, de lo contrario permanecen en negro.

Las conexiones booleanas internas se muestran Online también en azul en estado TRUE, de lo contrario en negro. El valor de las conexiones internas no booleanas se muestra en una pequeña caja con esquinas redondeadas en el pin de salida de la conexión.

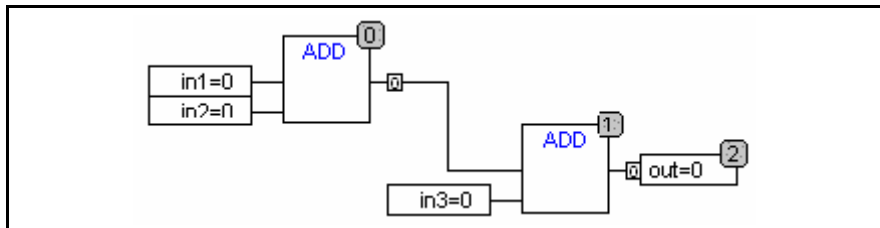


Fig. 5-130 : Conexiones en modo Online

Los PIN en macros se monitorizan como cajas de entrada y salida.

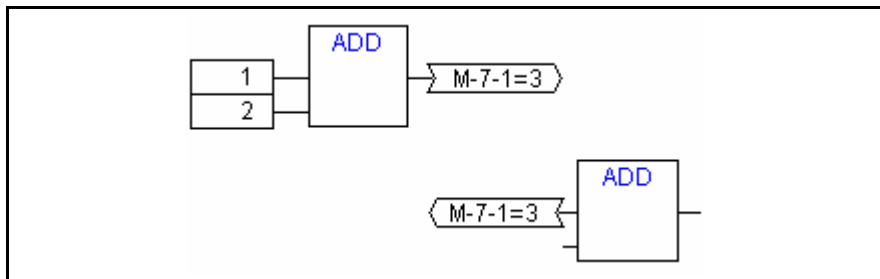


Fig. 5-131 : PIN en macros en modo Online

Las conexiones no booleanas con marcas de conexión muestran su valor dentro de la marca de conexión. Para conexiones booleanas, tanto las líneas como los nombres de las marcas se muestran también en azul si la línea transporta TRUE, de lo contrario permanecen en negro.



### **Control de proceso:**

Si está activado el control de proceso, las conexiones recorridas se marcan con el color especificado en las opciones de proyecto.

### **Breakpoints:**

Los puntos de interrupción pueden colocarse en todos los elementos que también poseen un índice de secuencia de procesamiento. El procesamiento del programa se detiene antes de ejecutarse el elemento en cuestión, esto es, para componentes y salidas, antes de la asignación de las entradas, para marcas de salto antes de la ejecución del elemento con el siguiente índice. Como posición del punto de interrupción en el diálogo de Breakpoint se utiliza el índice de secuencia de procesamiento del elemento.

Los puntos de interrupción se insertan en un elemento seleccionado mediante la tecla F9 o utilizando la opción de menú "Colocar/suprimir breakpoint" en el menú "Online" o "Extras", o bien en el menú contextual del editor. Si se ha insertado un punto de interrupción en un elemento, la próxima vez que se ejecute la orden "Colocar/suprimir breakpoint" éste se borrará, y viceversa. Además, se puede conmutar el punto de interrupción en un elemento haciendo doble clic sobre él.

La representación del breakpoint tiene lugar con los colores definidos en las opciones de proyecto.

### **Marca RETURN:**

En el modo Online se crea automáticamente una marca de salto con la designación "RETURN" en la primera columna y detrás del último elemento en el editor. Esta marca señala el final del componente y durante el procesamiento por pasos individuales se salta antes de salir del componente. En las macros no se insertan marcas RETURN.

### **Pasos individuales:**

En "Paso individual sobre" se salta siempre al elemento con el índice de secuencia inmediatamente superior. Si el elemento actual es una macro o un componente, en "Paso individual en" se produce una ramificación hacia su implementación. Si desde allí se ejecuta "Paso individual sobre", se salta al elemento cuyo índice de secuencia sigue al de la macro.

### **"Extras" "Zoom"**

#### **Forma abreviada: <Alt> + <Intro>**

Mediante esta orden se puede abrir la implementación de un componente cuando está seleccionado el componente en el editor CFC.