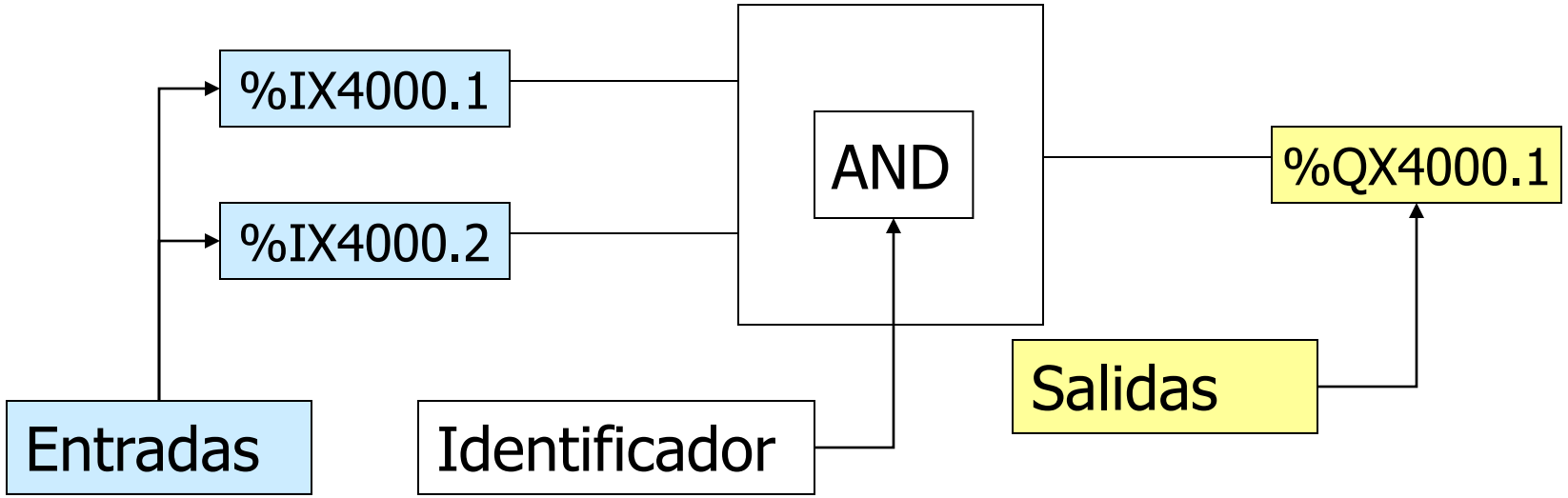
Lenguaje FBD
(Function Block Diagram)

Controladores Lógicos Programables

El Bloque Funcional

Bloque funcional: rectángulo con entradas, salidas e identificador



Los tipos de datos dependen del bloque

Estructura de programa FBD

Programa FBD: consiste en conexiones entre bloques funcionales y datos, por líneas de conexión

El programa se ejecuta de arriba abajo, y de izquierda a derecha

Se permite conectar la salida de un bloque a la entrada de otro

Bloques Funcionales

- Se definen sobre la base de “templates” de bloques
- Template de bloque: programa que define el bloque
- Para utilizar un template de bloque, la tarea declara una instancia del template
- Bibliotecas propias de usuario (re-uso)

Funciones

- Aceptan múltiples parámetros de entrada
- Salida única

Función vs Bloque Funcional

- Diferencia entre bloque funcional y función:
  - Número de salidas:
    - Función permite sólo una salida
    - Bloque funcional permite más de una
  - Variables persistentes (conservan valor entre ejecuciones):
    - Función no
    - Bloque funcional sí

Grupos de instrucciones

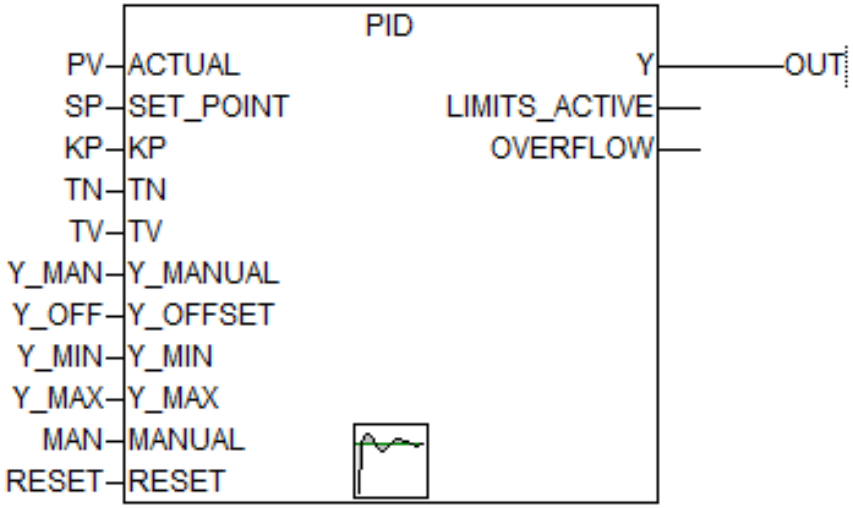
Clasificación:

Funciones binarias	Entradas y salidas de tipo binario: AND, OR, XOR
	Número de entradas variable; posibilidad de negación de entradas
Funciones de timers y contadores	Detalladas en la clase sobre lenguaje LD
Funciones de palabras/reales	Funciones de comparación: el resultado es un bit
	Funciones aritméticas: el resultado es una palabra/real (+, -, *, /)
	Funciones lógicas bit a bit: AND, OR, XOR, Shift, Rotate
Funciones de control de programa	JUMP; lectura o escritura inmediata de E/S
Funciones de control	PI, PID, etc. *
Funciones de comunicación	Comunicación por protocolos, por ejemplo MODBUS
Funciones de conversión de formato	PACK y UNPACK

El 90% de los programas se resuelve con el 20% de las instrucciones

Grupos de instrucciones

- Funciones de control: PI, PID, etc.



$$OUT = Y = KP(e + \frac{1}{TN}\int e \cdot \partial t + TV\frac{\partial e}{\partial t}) + Y\_OFFSET; e = SP - PV$$

$$PID(s) = KP\left(1 + \frac{1}{TN \cdot s} + TV \cdot s\right)$$

LD o FBD

Algunas diferencias entre un programa FBD y un programa LD:

La implementación de las funciones lógicas de bits

La concatenación de bloques funcionales:

- Permitida en FBD
- No permitida en LD

Ejemplo Bomba

Se desea escribir un programa que controle el encendido - apagado de una bomba.

La bomba será encendida si:

1) Se pulsa el botón de arranque.

2) La protección térmica está deshabilitada.

3) Está abierto el botón de alarma.

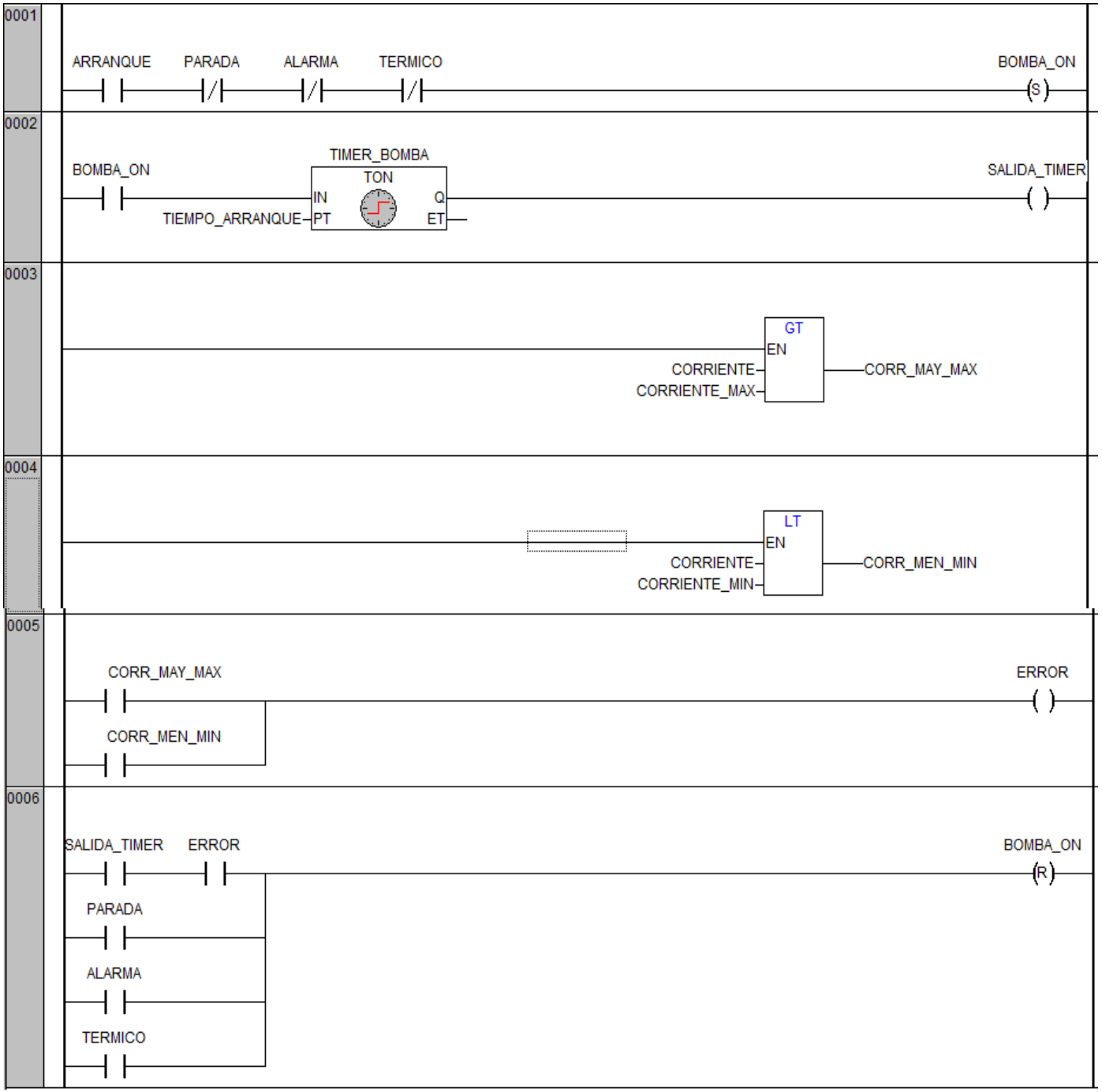
4) Está abierto el botón de parada.

Desde un tiempo T después del encendido, no puede haber ni sobre corriente ni baja corriente. Expresado de otra forma, desde un tiempo T después del arranque, la corriente I debe cumplir $I_{MIN} < I < I_{MAX}$, siendo I_{MIN} e I_{MAX} límites prefijados.

Ejemplo Bomba

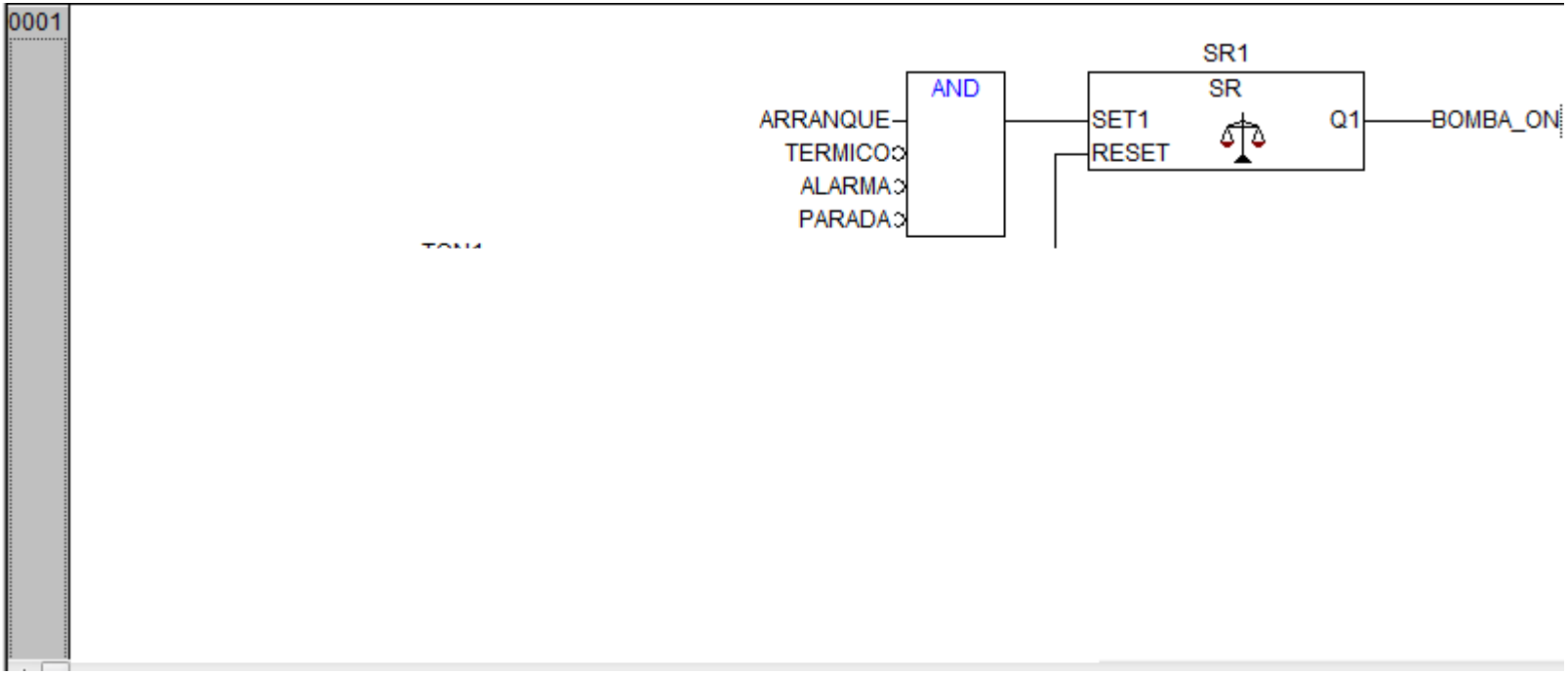
El motor de la bomba se apagará si:

1) Se pulsa el botón de parada.
2) Se cierra la protección térmica.
3) Se pulsa el botón de alarma.
4) Los límites de corriente no son los correctos.

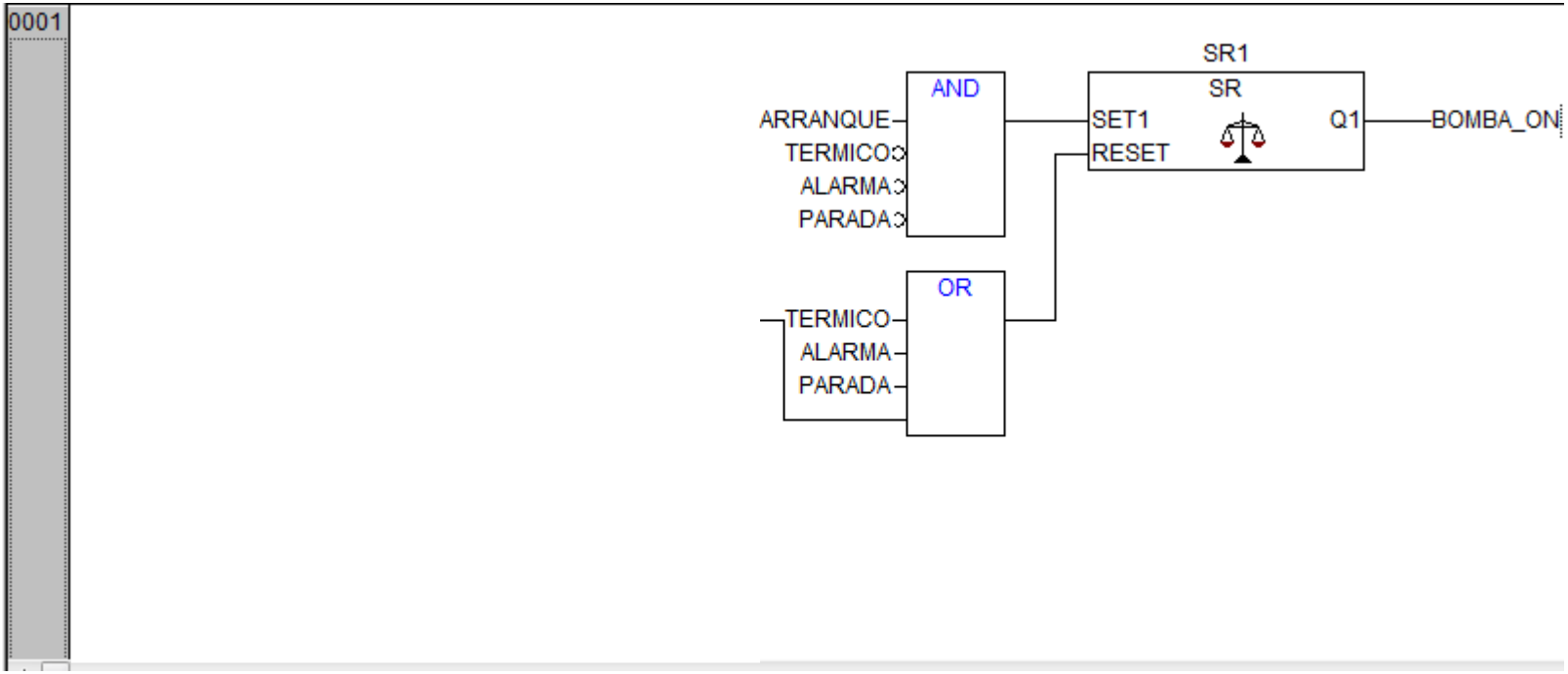

0001
ARRANQUE
PARADA
ALARMA
TERMICO
BOMBA_ON
S
0002
BOMBA_ON
TIMER_BOMBA
TON
IN
PT
Q
ET
TIEMPO_ARRANQUE
SALIDA_TIMER
0003
GT
EN
CORRIENTE
CORRIENTE_MAX
CORR_MAY_MAX
0004
LT
EN
CORRIENTE
CORRIENTE_MIN
CORR_MEN_MIN
0005
CORR_MAY_MAX
CORR_MEN_MIN
ERROR
0006
SALIDA_TIMER
ERROR
PARADA
ALARMA
TERMICO
BOMBA_ON
R

Ejemplo Bomba en FBD

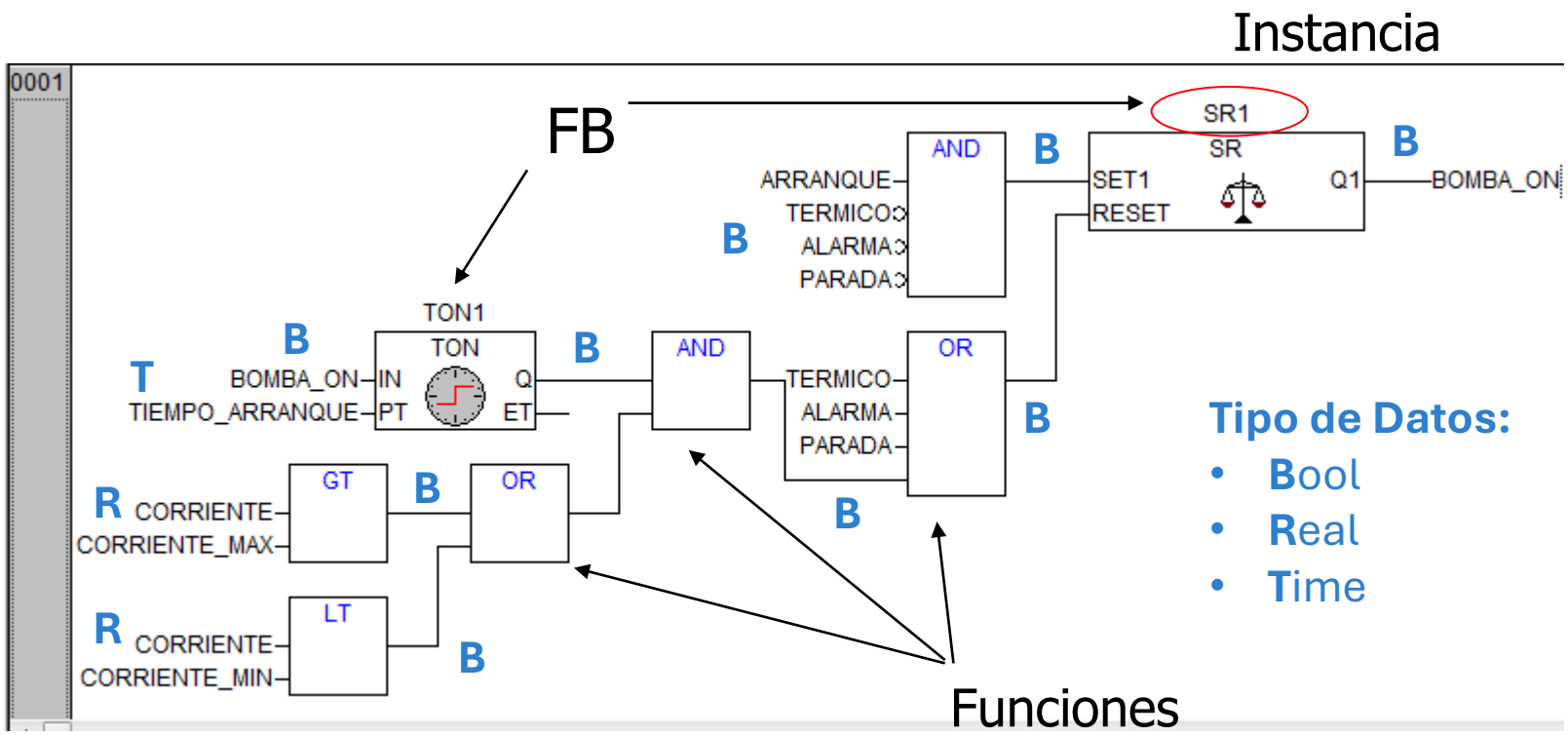
Ejemplo Bomba en FBD



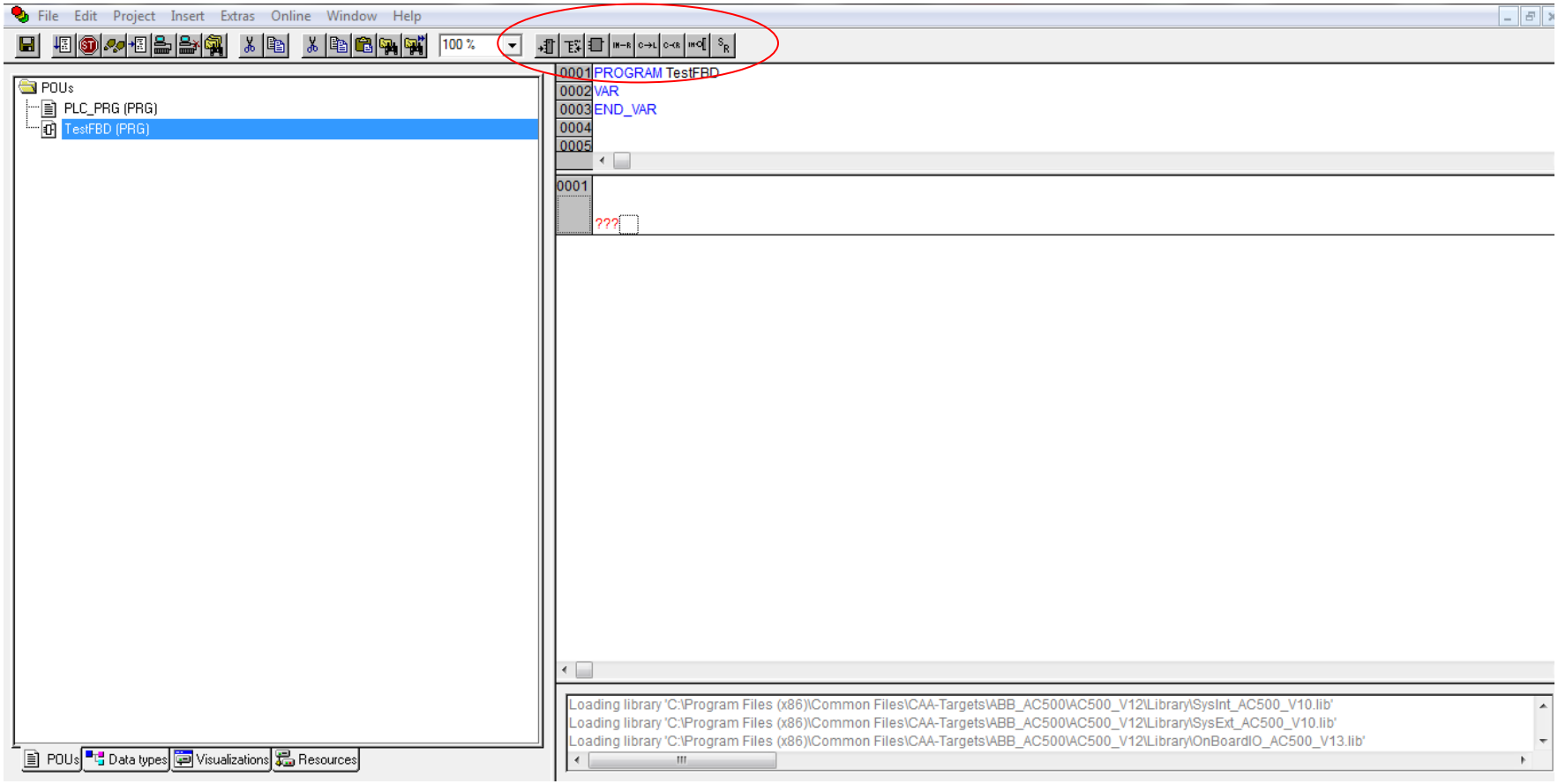
Ejemplo Bomba en FBD



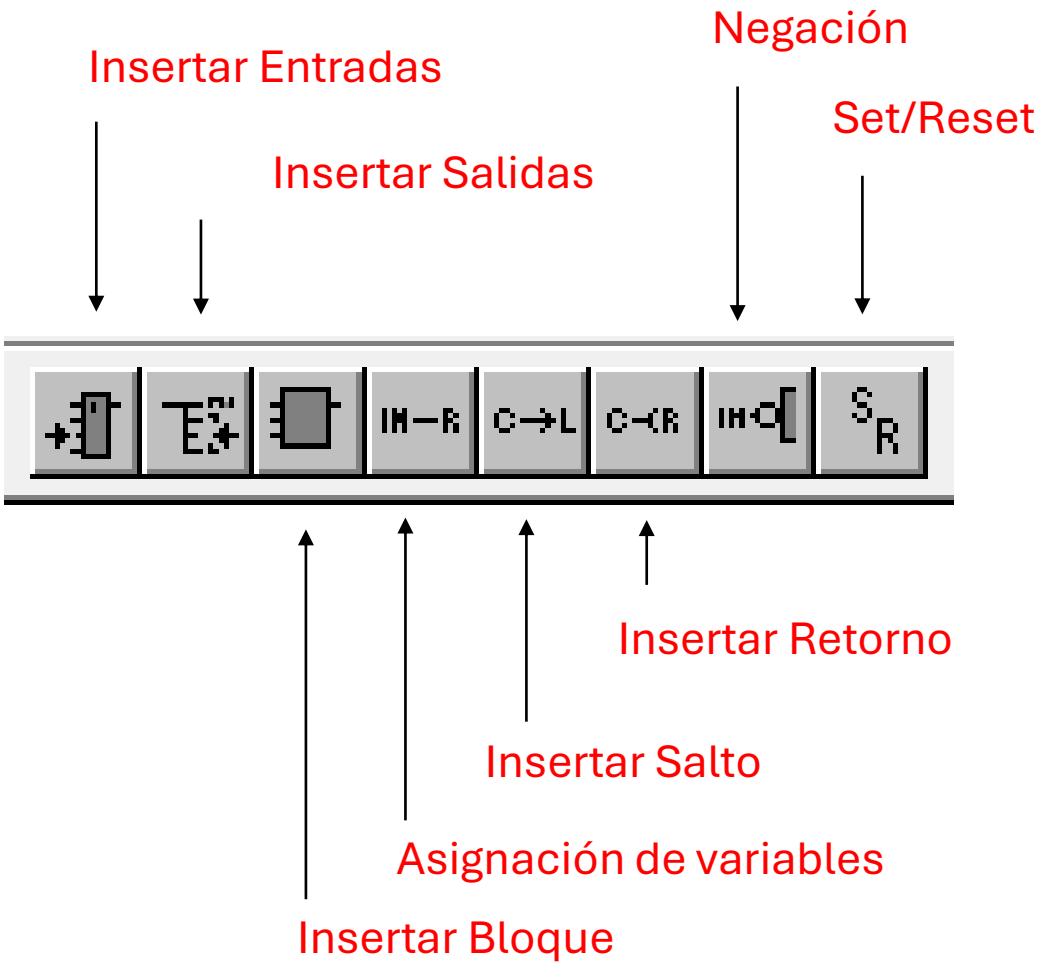
Ejemplo Bomba en FBD



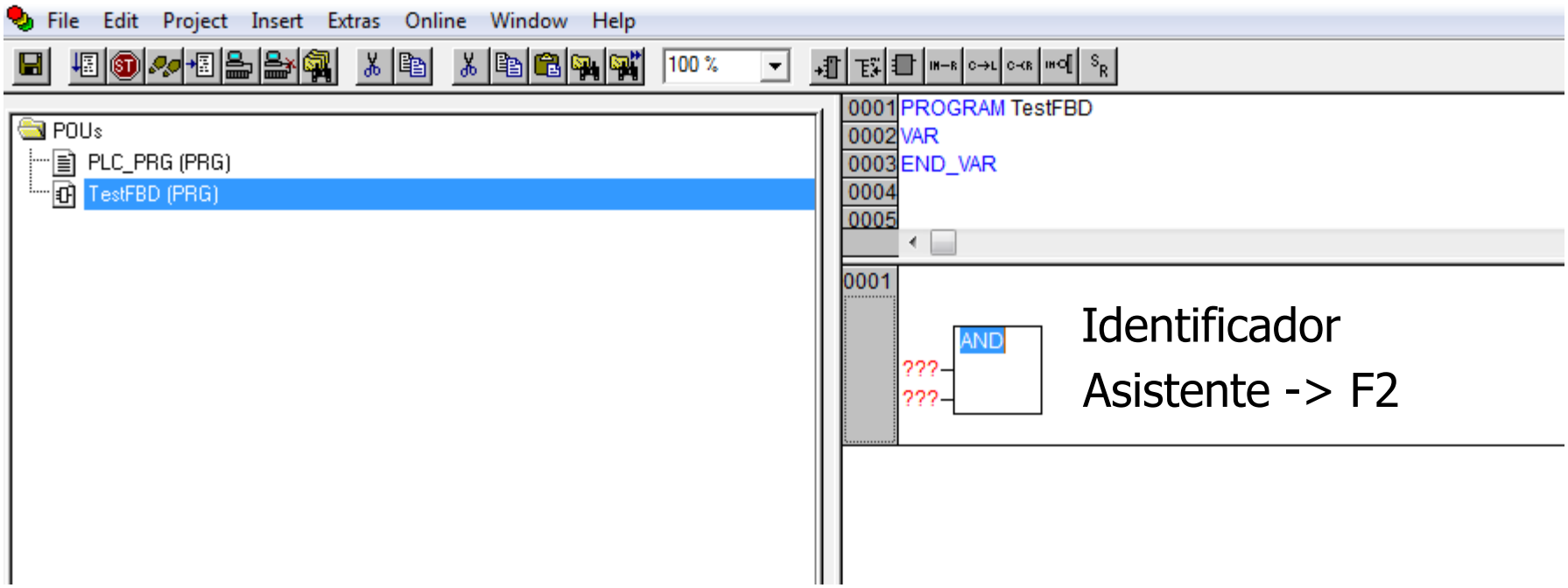
Ambiente en FBD



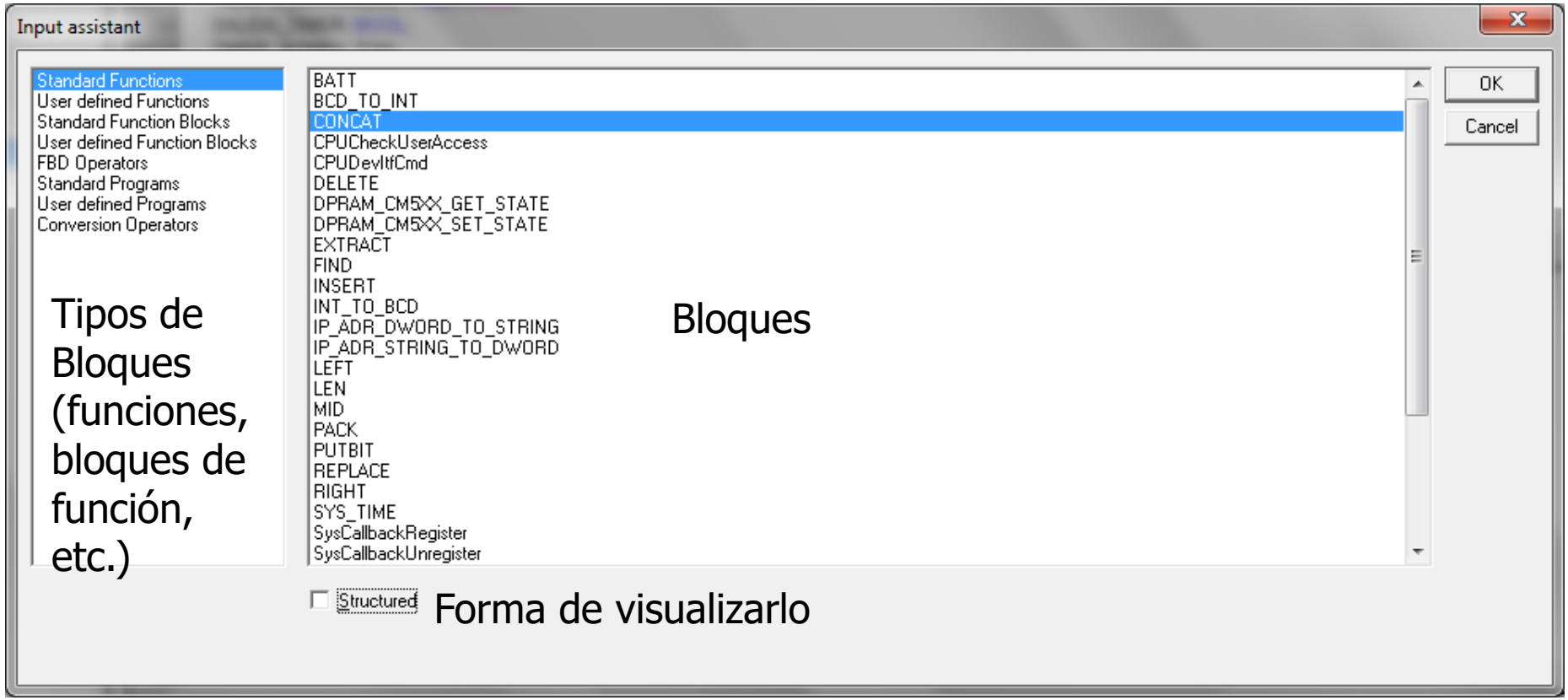
Herramientas FBD



Ambiente en FBD

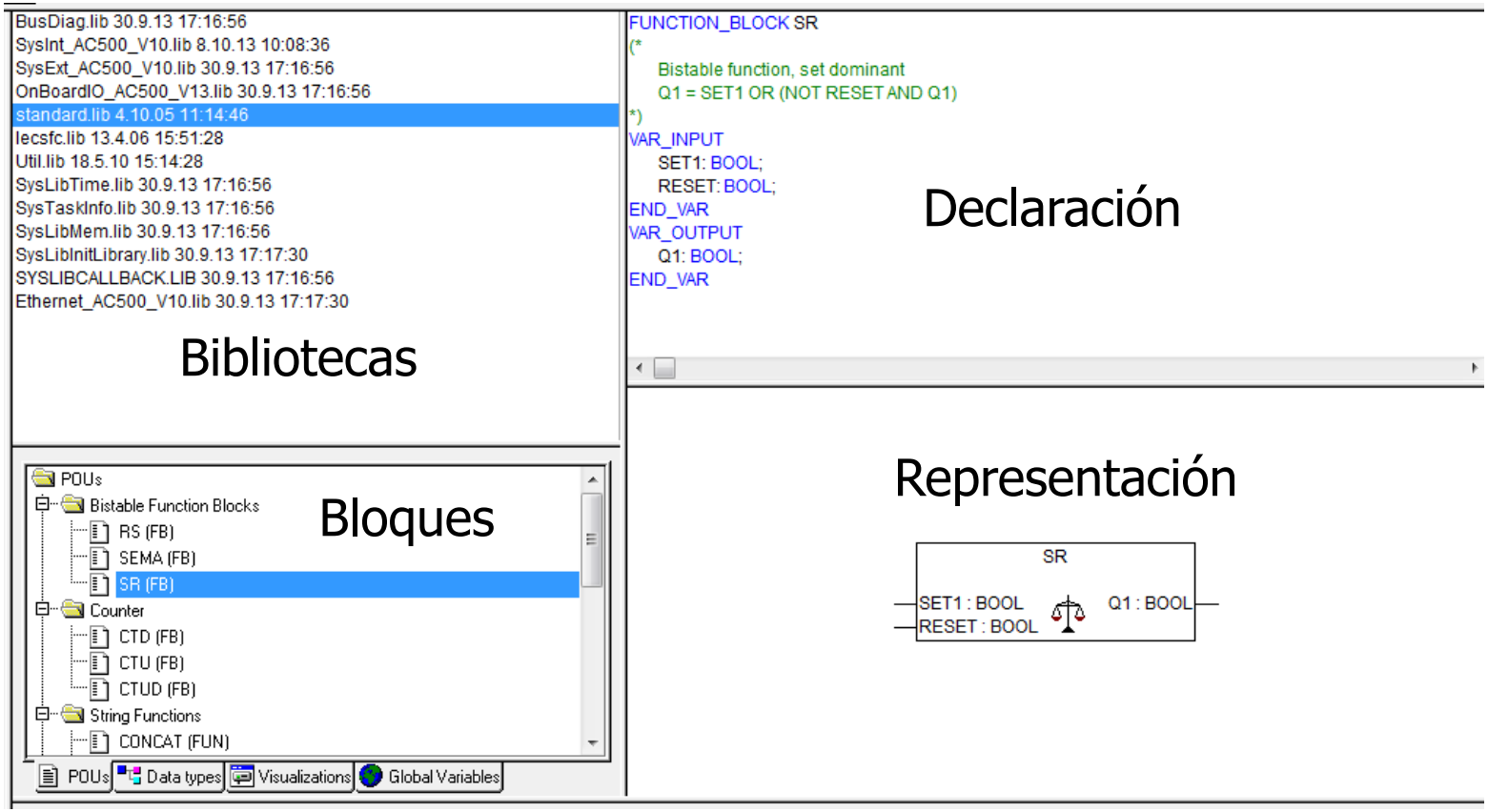


Bibliotecas de Bloques



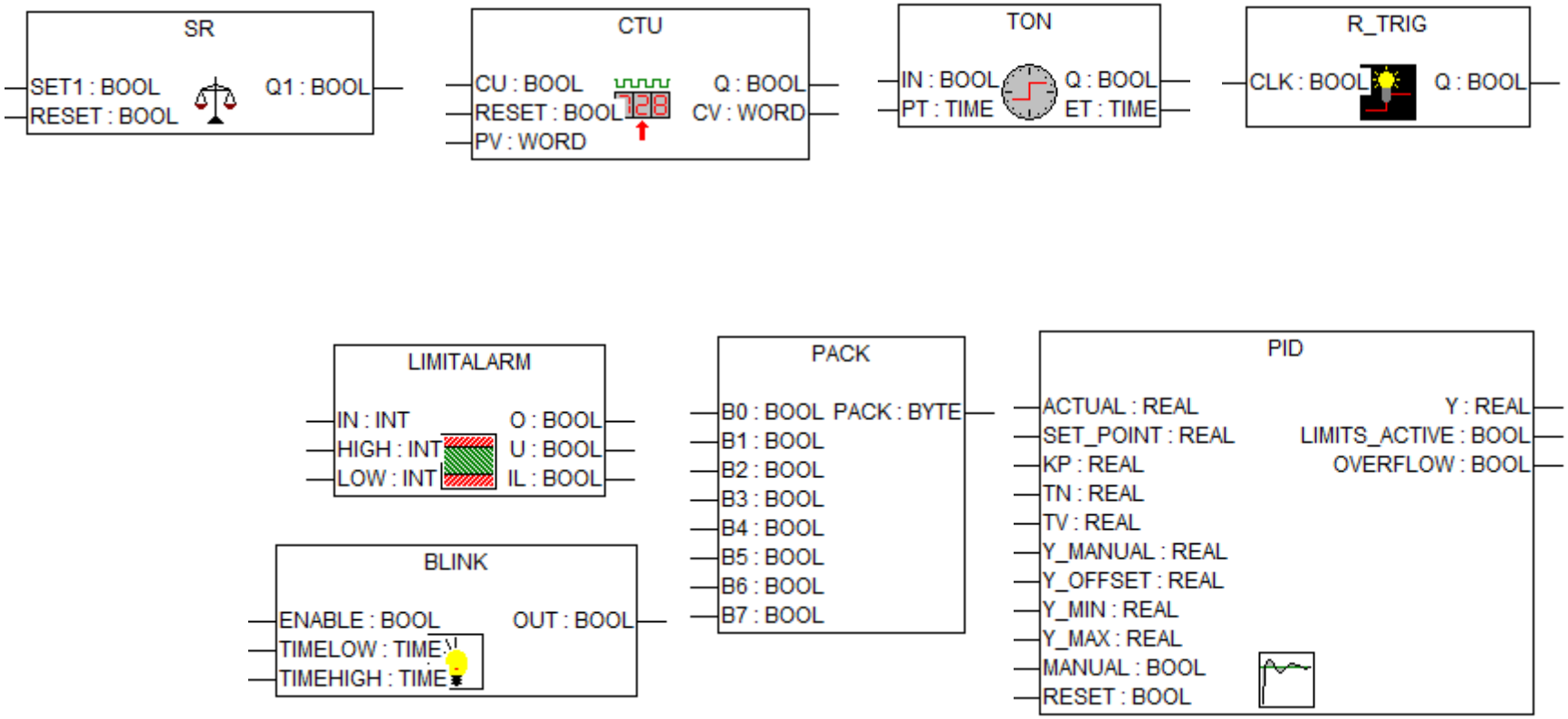
Library Manager

- Menú “Window” -> “Library Manager”

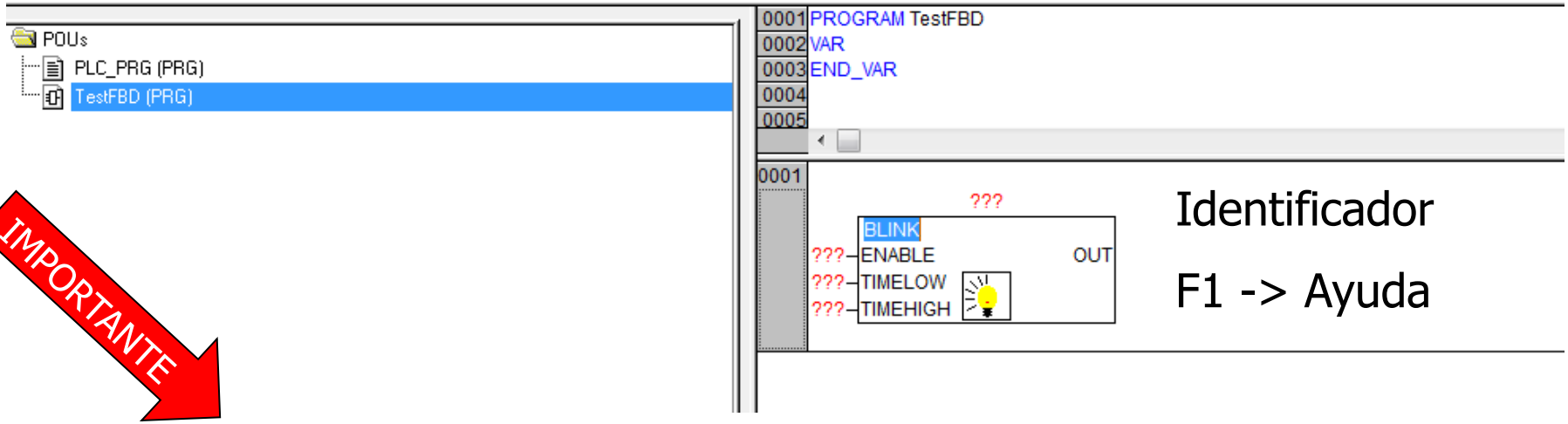


Library Manager

- Standard.lib:



Ayuda para Bloques – F1



IMPORTANTE

BLINK

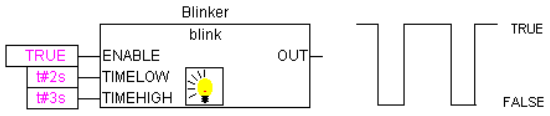
Provided by util.lib.

The function block BLINK generates a pulsating signal. The input consists of ENABLE of the type BOOL, as well as TIMELOW and TIMEHIGH of the type TIME. The output OUT is of the type BOOL.

If ENABLE is set to TRUE, BLINK begins to set the output for the time period TIMEHIGH to TRUE and afterwards to set it for the time period TIMELOW to FALSE.

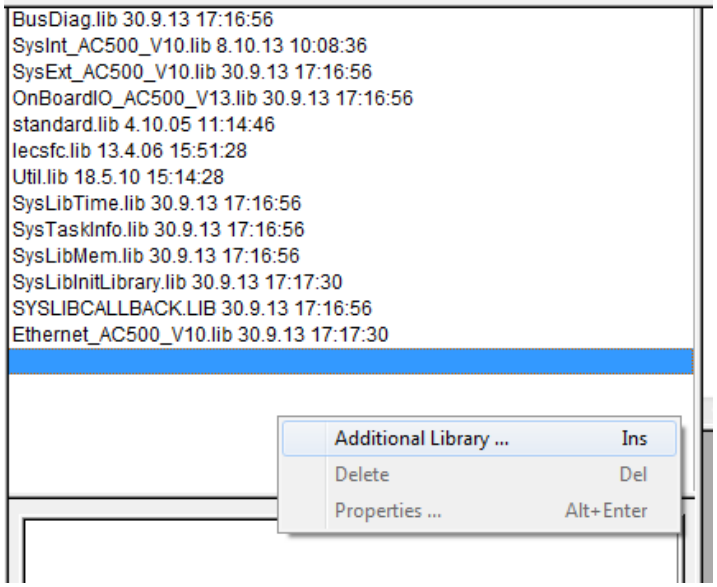
When ENABLE is reset to FALSE, output OUT will not be changed, i.e. no further pulse will be generated. If you explicitly also want to get OUT FALSE when ENABLE is reset to FALSE, you might use "OUT AND ENABLE" (i.e. adding an AND box with parameter ENABLE) at the output.

Example in CFC:

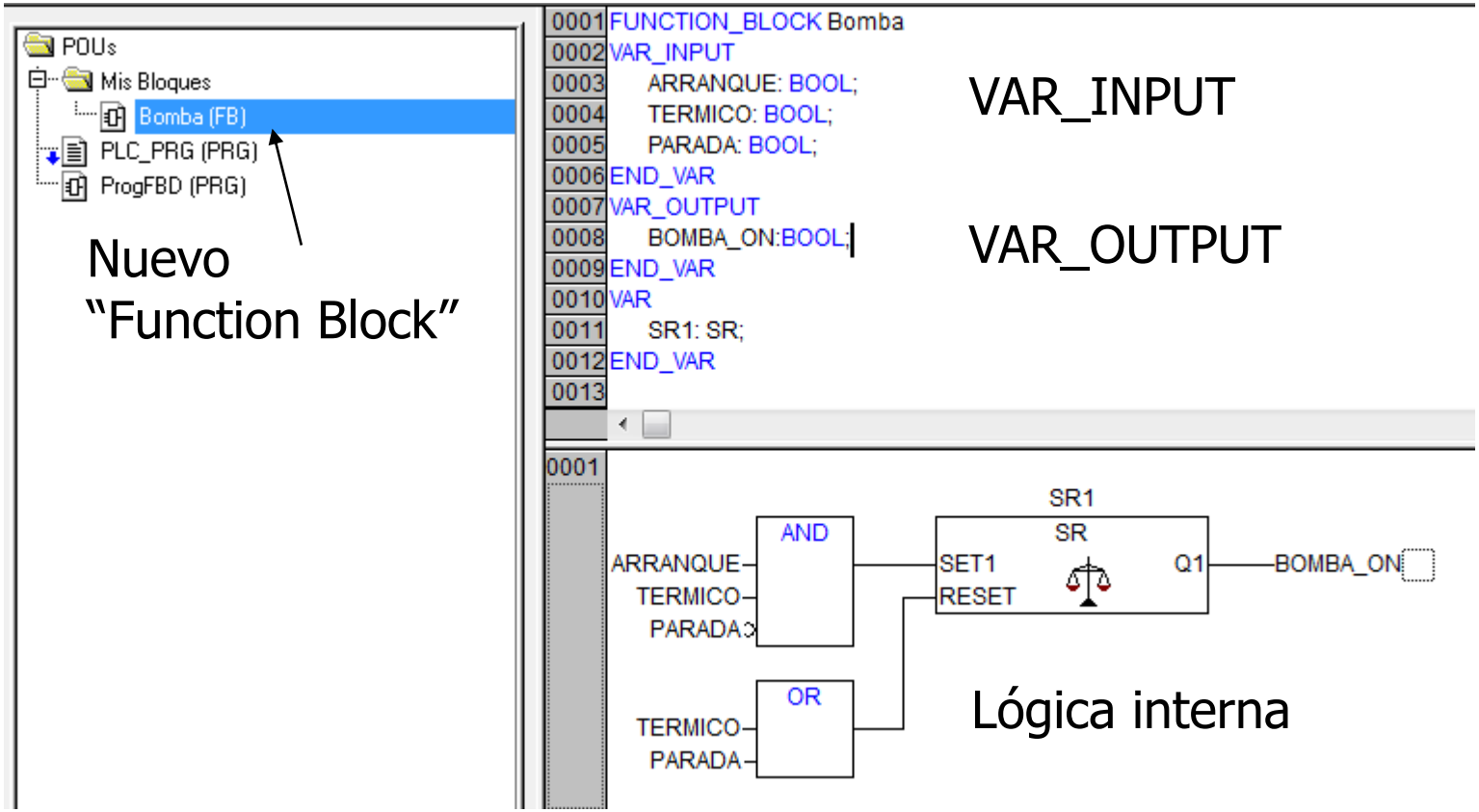


Library Manager

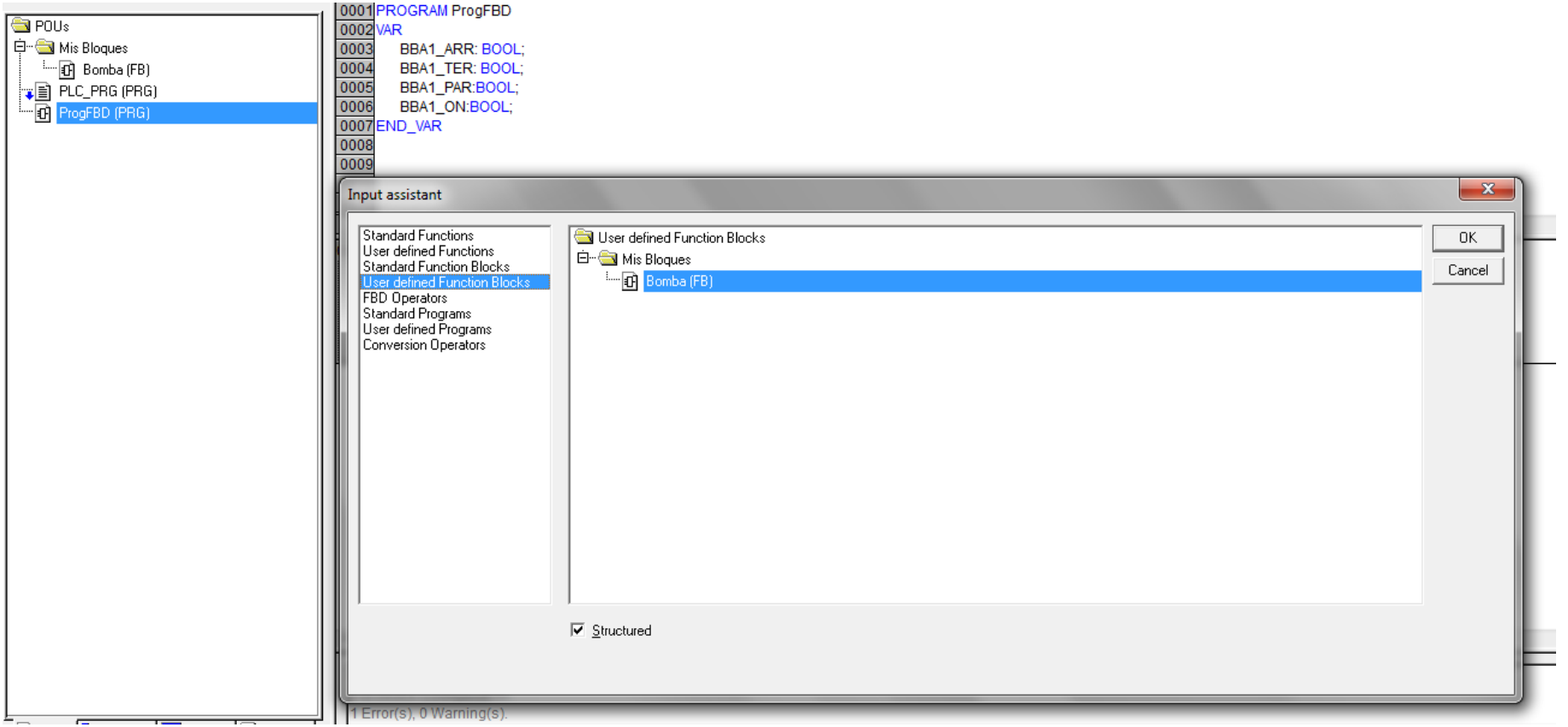
- Agregar bibliotecas:



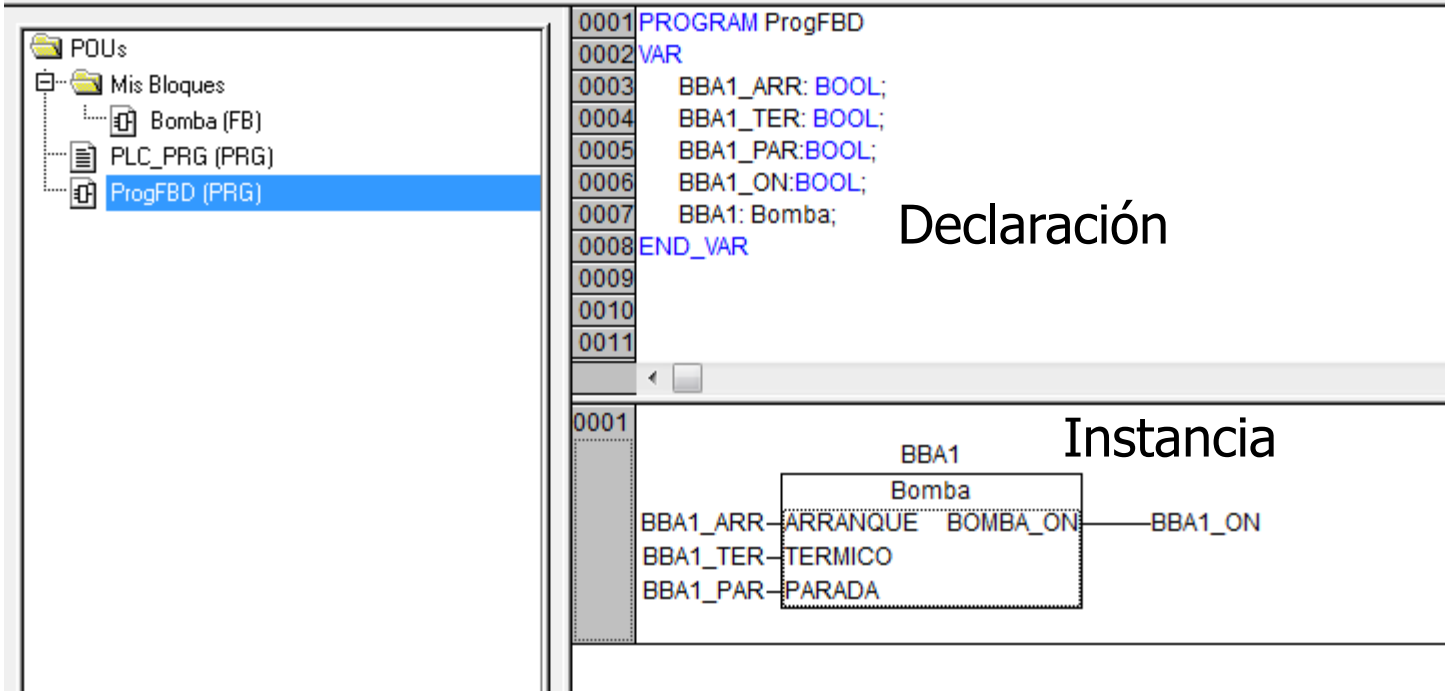
Bloques de Usuario



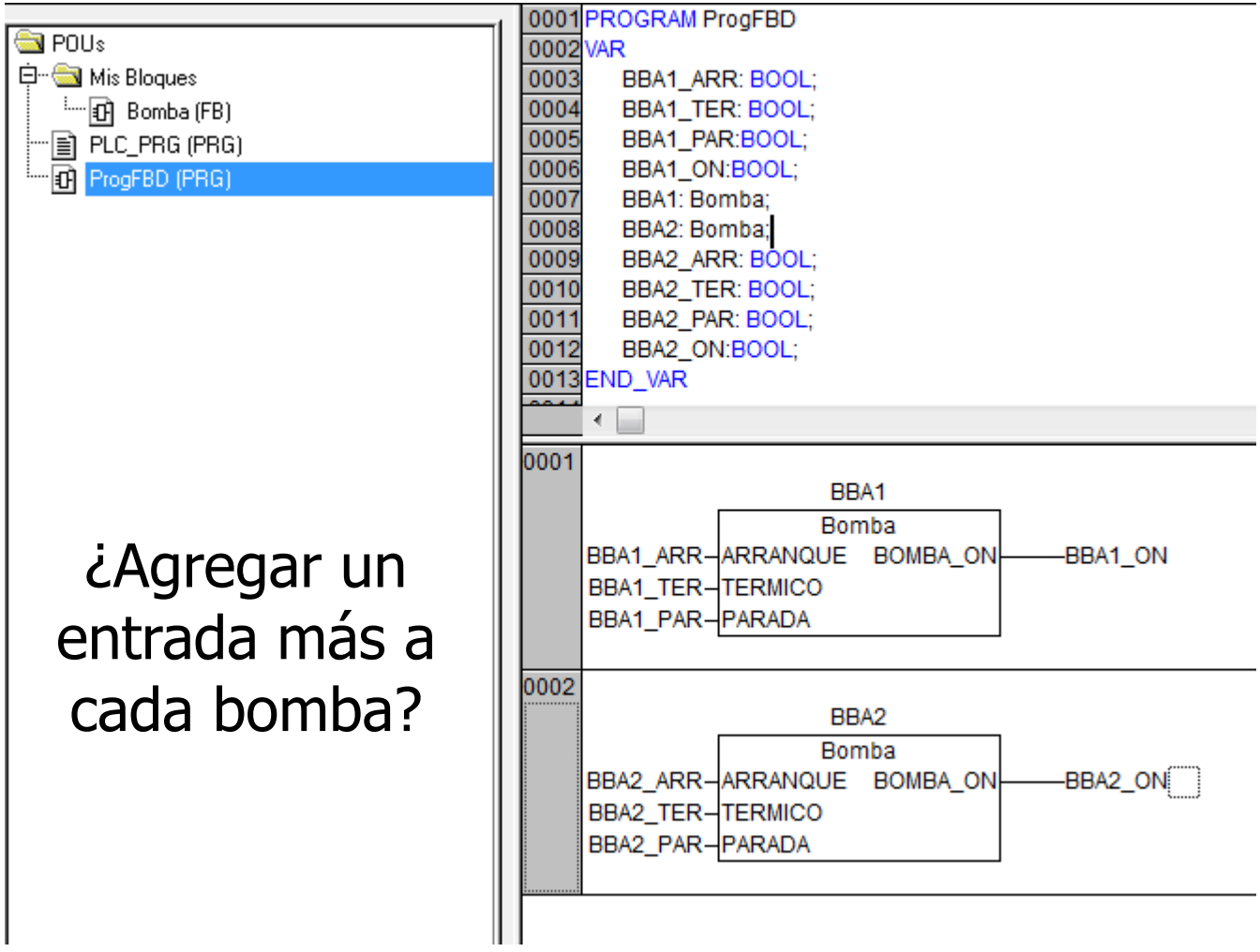
Bloques de Usuario



Bloques de Usuario

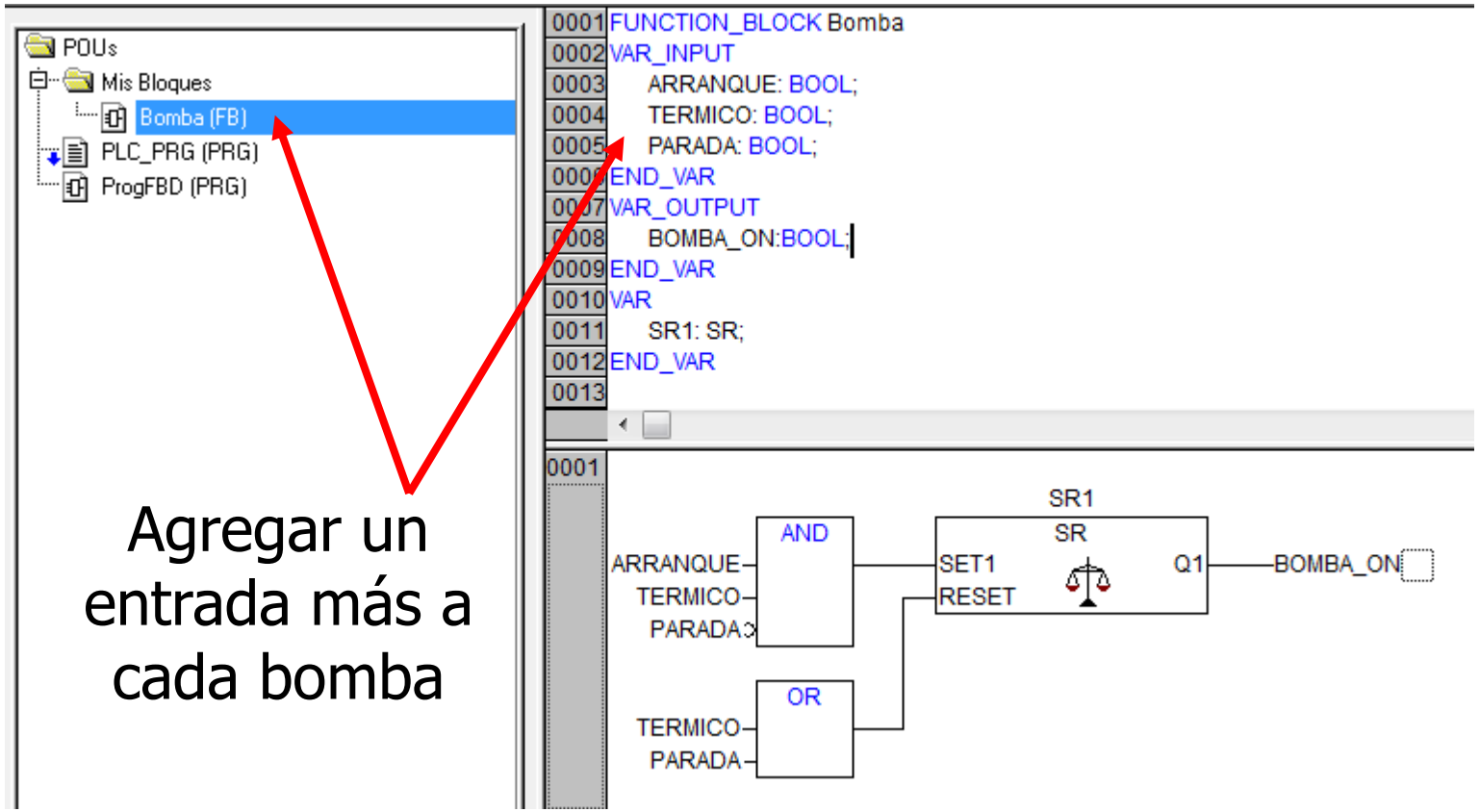


Bloques de Usuario



¿Agregar un entrada más a cada bomba?

Bloques de Usuario



Agregar un entrada más a cada bomba

Uso de bloques de funciones

- Re-uso de soluciones probadas
- Fácil modificación general
- Cajas negras con comportamiento conocido
- Simplifica el programa general

Elemento clave de la norma IEC 61131 para una alta calidad de software