

Funciones lógicas

Módulo de funciones lógico-matemáticas

Edición del manual: [1.0]_a

www.zennio.com

CONTENIDO

Contenido.....	2
Actualizaciones del documento	3
1 Introducción	4
1.1 Módulo de funciones lógicas.....	4
2 Configuración	5
2.1 Funcionamiento general	5
2.2 Disparadores	6
2.3 Condición de ejecución	6
2.4 Operaciones	7
2.5 Objetos de entrada	8
2.6 Variables internas.....	8
2.7 Objetos de resultado.....	8
3 Parametrización ETS.....	10
3.1 Configuración	10
3.2 Función n	12
3.2.1 Disparadores.....	13
3.2.2 Condición de ejecución.....	14
3.2.3 Operaciones.....	15
3.2.4 Resultado	17
ANEXO I: Listado de operaciones	20
Operaciones en lógica booleana (1 bit)	20
Operaciones aritméticas	21
Operaciones de comparación	22
Operaciones de conversión.....	23
Observaciones varias.....	25

ACTUALIZACIONES DEL DOCUMENTO

Versión	Modificaciones	Página(s)
[1.0]_a	Cambios en el software: <ul style="list-style-type: none">• Optimización interna.	-
[0.3]_a	Cambios en el software: <ul style="list-style-type: none">• Retardo interno entre envíos simultáneos.• Cancelación de envíos periódicos o retardados de resultados previos en caso de ejecutarse la función de nuevo.	-

1 INTRODUCCIÓN

1.1 MÓDULO DE FUNCIONES LÓGICAS

Muchos de los dispositivos Zennio (entre ellos, toda la familia de actuadores ACTinBOX y MAXinBOX) incorporan un módulo de funciones lógicas, lo que les permite efectuar operaciones **matemáticas** o en **lógica binaria** con datos procedentes del bus KNX, así como enviar los resultados mediante objetos de comunicación de diferentes tamaños.

Los operandos de estas funciones pueden ser de los siguientes tipos:

- **Objetos de comunicación** recibidos a través del bus KNX.
- **Variables internas** con resultados parciales de operaciones previas.
- **Valores constantes**, establecidos por parámetro en ETS.

Es posible definir diez funciones lógicas diferentes e independientes, cada una de las cuales, a su vez, podrá consistir en hasta cuatro operaciones sucesivas, que podrán compartir variables entre sí de tal modo que el resultado de una sea entrada de la siguiente.

Importante: *dependiendo de cada dispositivo, el número de funciones lógicas disponibles y la funcionalidad cambian ligeramente, por lo que el manual de usuario de este módulo se ha particularizado para cada dispositivo. Así pues, se recomienda encarecidamente utilizar los enlaces de descarga que figuran en la ficha del dispositivo concreto que se desee configurar, dentro la página web de Zennio (www.zennio.com).*

2 CONFIGURACIÓN

2.1 FUNCIONAMIENTO GENERAL

El módulo de funciones lógicas permite habilitar y configurar hasta diez funciones numéricas independientes. El funcionamiento de cada una de estas funciones se divide en cuatro etapas:

- **Llamada:** el primer paso para que la función configurada se ejecute consiste en *llamarla*. Para ello, se podrá configurar uno o varios objetos de comunicación, de modo que cada vez que alguno de ellos actualice su valor desde el bus, se disparará automáticamente la ejecución de la función (siempre y cuando se cumpla la condición de ejecución).
- **Condición de ejecución:** tras haberse activado un disparador asignado a una función lógica se comprobará si se cumple la condición de ejecución. Esta condición consiste en la comparación de dos operandos de entrada. En caso de que la comparación sea cierta, se procederá a la ejecución de las operaciones. Si no se configura ninguna condición de ejecución, la función se ejecutará siempre que se la llame.
- **Operaciones:** el disparo de la función desencadenará, a su vez, la ejecución sucesiva de hasta cuatro operaciones matemáticas o binarias. Para cada una de ellas deberá configurarse lo siguiente:
 - **Tipo de operación:** acción deseada (suma, resta, negación, etc.).
 - **Operandos:** valores con los que se operará. Podrán ser objetos de comunicación de entrada, variables internas donde se haya guardado el resultado de alguna operación previa, o constantes predefinidas en ETS.
 - **Resultado:** variable interna donde almacenar el resultado de la operación.
- **Resultado:** se deberá seleccionar qué variable interna contiene el resultado global de la función. Su valor se enviará, al término de la ejecución de todas las operaciones, a través del correspondiente objeto de comunicación.

En la Figura 1 se muestra un diagrama del funcionamiento general de las funciones lógicas.

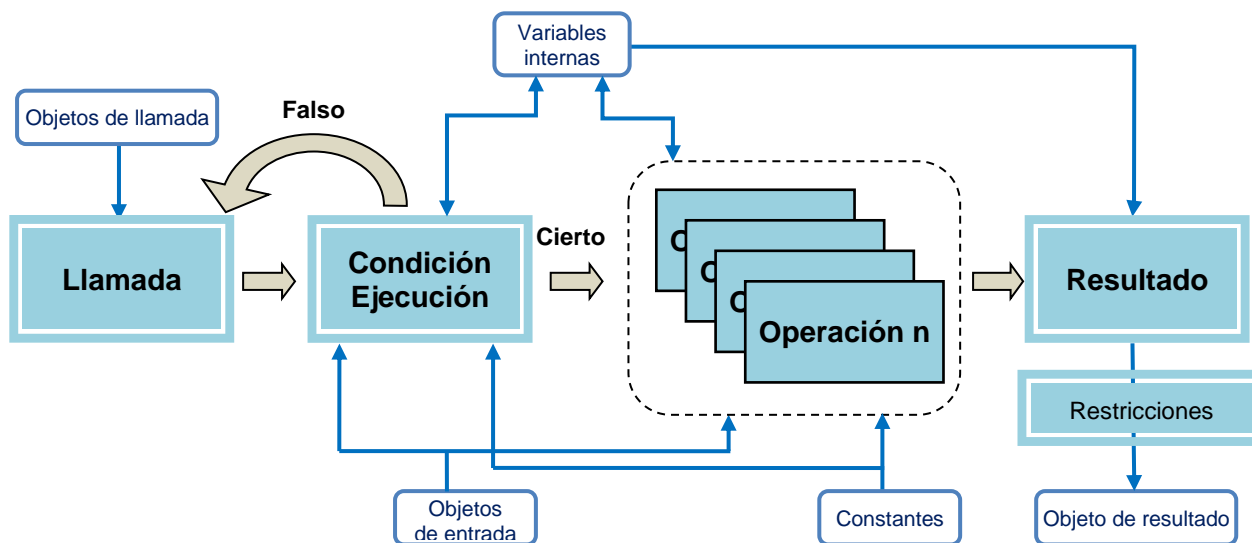


Figura 1. Diagrama de funcionamiento de las funciones lógicas.

2.2 DISPARADORES

Para cada función, se podrá disponer de hasta ocho objetos de llamada (de un bit, un byte, dos bytes o cuatro bytes), cada uno de los cuales disparará la función automáticamente cada vez que reciba un valor desde el bus. Estos objetos no necesariamente tienen por qué coincidir con objetos utilizados como operandos.

Nótese que la ejecución en sí de las operaciones una vez disparada la función dependerá del resultado de evaluar la condición de ejecución, si la hay.

2.3 CONDICIÓN DE EJECUCIÓN

Una vez que se ha activado el disparador, el siguiente paso es comprobar si la situación actual reúne las condiciones necesarias para que el cálculo de las operaciones se realice.

Si no se ha configurado ninguna condición de ejecución, las operaciones se efectúan directamente. En cambio, si está habilitada, las operaciones se realizarán sólo si se cumple la comparación de **dos operandos de entrada**, que pueden ser valores constantes, objetos de comunicación o variables internas.

Las **operaciones de comparación** disponibles son (en función del tamaño de comparación elegido): igual a; no igual a; mayor que; mayor o igual que; menor que; menor o igual que.

2.4 OPERACIONES

Como ya se ha explicado, cada función lógica consiste en la ejecución de hasta cuatro operaciones consecutivas, que pueden ser de los siguientes tipos:

- **Lógica:** ID, NOT, AND, OR, XOR, NAND, NOR y NXOR.
- **Aritmética:** ID, suma, resta, multiplicación, división, máximo y mínimo.
- **Comparación:** mayor, mayor o igual, menor, menor o igual, igual, desigual.
- **Conversión:** operaciones de conversión del tamaño (*cast*) de un determinado operando. Por ejemplo: de un bit a un byte.

El módulo de funciones lógicas puede operar con los siguientes rangos de valores (ya sean objetos, variables internas con resultados intermedios previos, o constantes establecidas por parámetro en ETS):

- Binarios: **0** y **1**.
- Enteros sin signo (un byte): **0 – 255**.
- Valores de porcentaje (un byte): **0 – 100**.
- Enteros sin signo (dos bytes): **0 – 65535**.
- Enteros con signo (dos bytes): **-32768 – 32767**.
- Valores en coma flotante (dos bytes): **-671088,64 – 671760,96**.
- Entero con signo (cuatro bytes): **-2147483648 – 2147483647**.

Para más información sobre estas operaciones, sobre la conversión entre tamaños y sobre el truncado de valores, consúltese el *ANEXO I: Listado de operaciones*.

2.5 OBJETOS DE ENTRADA

Podrán habilitarse múltiples objetos específicos para usarlos con las funciones:

- Hasta 32 objetos de un bit,
- Hasta 16 objetos de un byte,
- Hasta 16 objetos de dos bytes.
- Hasta 8 objetos de cuatro bytes.

Los valores de estos objetos podrán intervenir, por ejemplo, como operandos dentro de las operaciones de las funciones que se habiliten.

2.6 VARIABLES INTERNAS

Asimismo, el integrador podrá disponer de:

- 32 variables internas de un bit (b_1, \dots, b_{32}),
- 16 variables internas de un byte (n_1, \dots, n_{16}),
- 16 variables internas de dos bytes (x_1, \dots, x_8).
- 8 variables internas de cuatro bytes (y_1, \dots, y_8).

Todas ellas permitirán el almacenamiento temporal de resultados intermedios, que a su vez podrán emplearse como valores de entrada de operaciones posteriores.

2.7 OBJETOS DE RESULTADO

Cada función lógica habilitada dispondrá de un objeto específico (que será de un bit, un byte, dos bytes o cuatro bytes, en función de la parametrización de la función) a través del cual enviará al bus el valor de la variable interna que se seleccione por parámetro como resultado de la secuencia de operaciones que conforman la función.

El integrador podrá configurar si el envío de este objeto al bus se efectuará una vez cada vez que la función se ejecute, o bien periódicamente, o bien sólo cuando la función arroje un resultado diferente del de la anterior ejecución.

De igual modo, podrán restringirse los resultados que se envían, de tal modo que sólo se notifique el resultado al bus cuando se encuentre dentro de un determinado rango. Por último, también es posible configurar en ETS un retardo para el envío del resultado.

3 PARAMETRIZACIÓN ETS

3.1 CONFIGURACIÓN

La pestaña 'Configuración' del módulo de funciones lógicas contiene las opciones que se muestran abajo.

Nota: puede ser que las pestañas de parámetros del módulo de funciones lógicas no se muestren por defecto en ETS y que resulte necesario habilitar este módulo desde la pestaña General del propio dispositivo. Por favor, consúltese el manual de usuario del dispositivo específico para más detalles.

The screenshot shows the 'Configuración' (Configuration) tab for logical functions. On the left, a sidebar contains a tree view with 'GENERAL' expanded and 'Funciones lógicas' selected, with 'Configuración' as the active sub-tab. The main area is titled 'ACTIVAR FUNCIONES LÓGICAS' and lists ten functions (Función 1 to Función 10), each with an unchecked checkbox. Below this, a section titled 'NÚMERO TOTAL DE OBJETOS DE ENTRADA' (Total Number of Input Objects) contains four dropdown menus for '1 bit', '1 byte', '2 bytes', and '4 bytes', all currently set to '0'.

Figura 2. Pestaña general del módulo de funciones lógicas.

Como puede verse, ninguna de las diez funciones aparece habilitada por defecto. A medida que se habiliten, irán apareciendo pestañas adicionales en el listado de la izquierda.

Además, en esta pestaña se determina el número de objetos de entrada necesarios de cada tipo para el conjunto de todas las funciones lógicas que vayan a ser utilizadas.

Como ya se vio en la sección 2.5, se podrán habilitar hasta 32 objetos de un bit, 16 de un byte, 16 de dos bytes u 8 de cuatro bytes.

En las siguientes secciones se explica la finalidad de cada pestaña y de los parámetros que contiene.

3.2 FUNCIÓN n

Para cada función habilitada (ver sección 3.1) se mostrará en el listado de la izquierda una pestaña específica, que a su vez se subdivide en otras cuatro.

ACTIVAR FUNCIONES LÓGICAS	
Función 1	<input checked="" type="checkbox"/>
Función 2	<input type="checkbox"/>
Función 3	<input type="checkbox"/>
Función 4	<input type="checkbox"/>
Función 5	<input type="checkbox"/>
Función 6	<input type="checkbox"/>
Función 7	<input type="checkbox"/>
Función 8	<input type="checkbox"/>
Función 9	<input type="checkbox"/>
Función 10	<input type="checkbox"/>

NÚMERO TOTAL DE OBJETOS DE ENTRADA	
1 bit	<input type="text" value="0"/>
1 byte	<input type="text" value="0"/>
2 bytes	<input type="text" value="0"/>
4 bytes	<input type="text" value="0"/>

Figura 3. Función 1 habilitada.

3.2.1 DISPARADORES

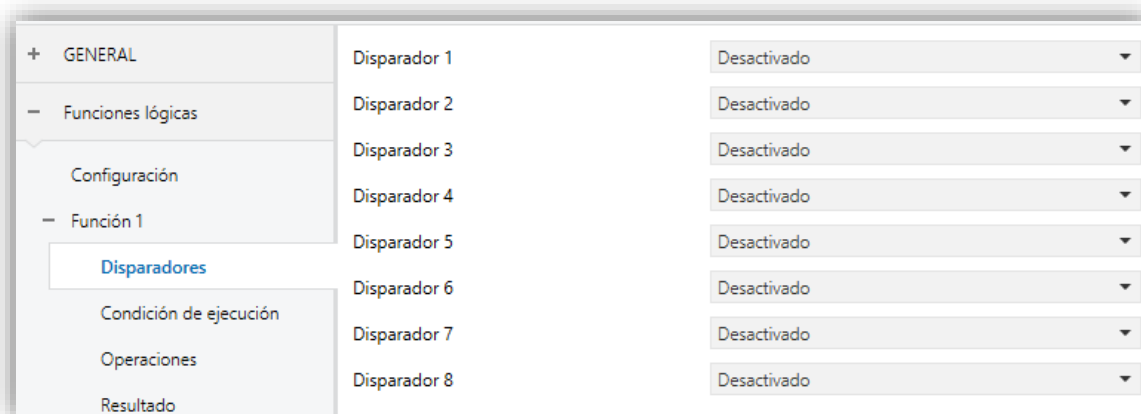


Figura 4. Pestaña "Disparadores"

Desde esta sección se podrá seleccionar hasta **ocho disparadores** que actuarán como objetos de llamada. Los objetos de llamada serán los responsables de disparar la ejecución de la función cada vez que uno de ellos reciba algún valor desde el bus, siempre y cuando se cumpla la condición de ejecución. Igualmente, si se utiliza un mismo objeto como objeto de llamada de varias funciones, todas ellas se dispararán consecutivamente cuando se escriba algún valor en el objeto.

Nota: los objetos que se desee utilizar como disparadores deberán haberse habilitado previamente (ver 3.1).

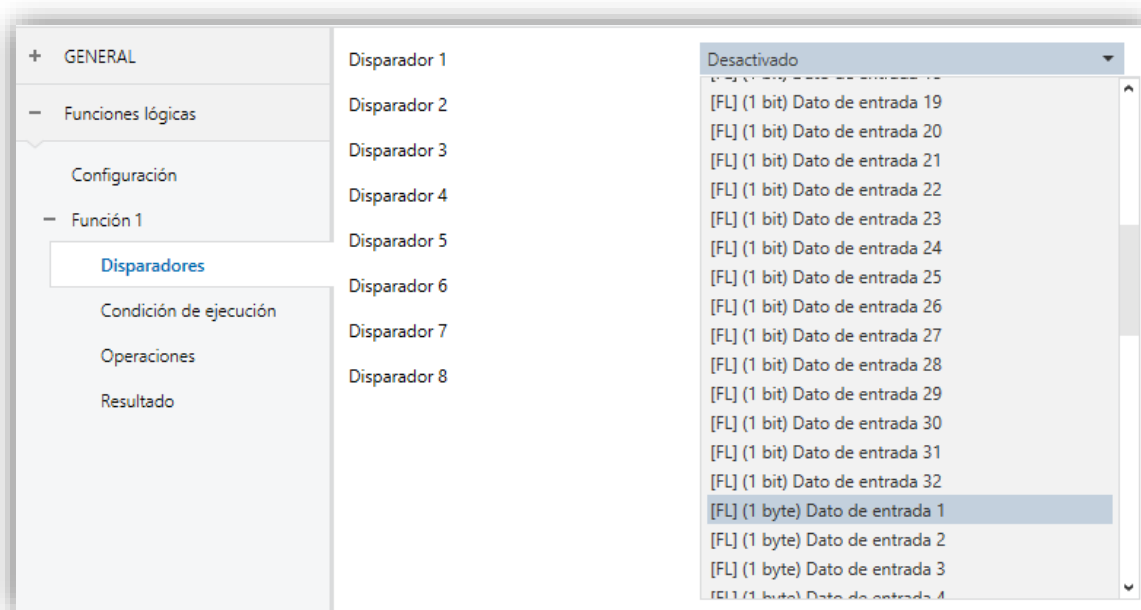
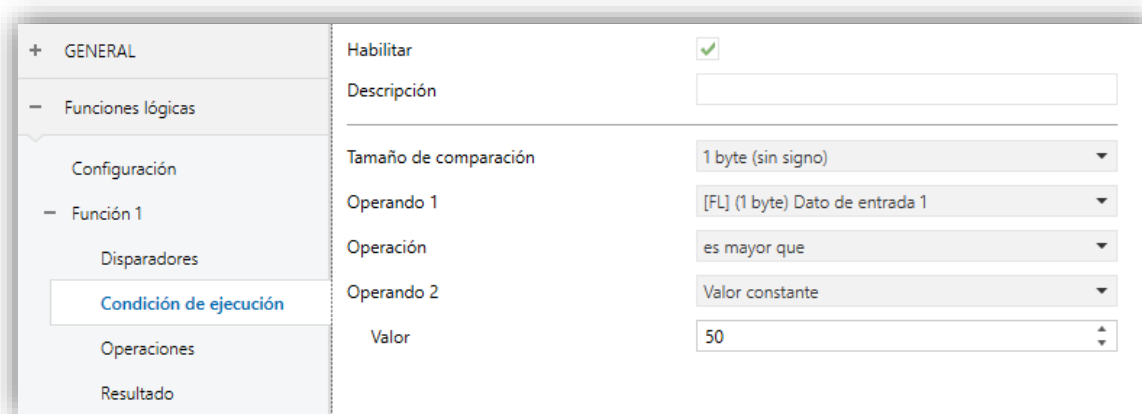


Figura 5. Selección de los objetos disparadores

3.2.2 CONDICIÓN DE EJECUCIÓN

Desde esta sección se puede configurar la condición de ejecución mediante los siguientes parámetros:



+ GENERAL	Habilitar	<input checked="" type="checkbox"/>
- Funciones lógicas	Descripción	<input type="text"/>
Configuración	Tamaño de comparación	1 byte (sin signo)
- Función 1	Operando 1	[FL] (1 byte) Dato de entrada 1
Disparadores	Operación	es mayor que
Condición de ejecución	Operando 2	Valor constante
Operaciones	Valor	50
Resultado		

Figura 6. Pestaña "Condición de ejecución"

- **Habilitar:** establece si se desea o no configurar una condición de ejecución.
- **Descripción:** permite especificar una breve descripción (hasta cien caracteres). Este campo no tiene ninguna implicación práctica; simplemente facilita al integrador la identificación de la función.
- **Tamaño de comparación:** tamaño de los operandos de entrada: 1 bit, 1 byte (sin signo), 1 byte (porcentaje), 2 bytes (sin signo), 2 bytes (con signo), 2 bytes (flotante), 4 bytes (con signo).
- **Operando 1:** origen del valor: constante, objeto o variable interna. En caso de seleccionarse "Valor constante", se mostrará un nuevo campo ("**Valor**") mediante el que especificar el valor constante deseado.
- **Operación:** el listado dependerá del campo **Tamaño de comparación**. De ser de un bit, sólo se mostrarán las opciones "es igual a" y "no es igual a". Si es mayor a 1 bit, se mostrarán todas las posibilidades (ver apartado 2.4).
- **Operando 2:** origen del valor: constante, objeto o variable interna. En caso de seleccionarse "Valor constante", se mostrará un nuevo campo ("**Valor**") mediante el que especificar el valor constante deseado.

3.2.3 OPERACIONES

El número máximo de operaciones por función lógica es cuatro. Éstas se activan individualmente y se ejecutarán de forma secuencial. Si hay alguna operación desactivada en medio, será ignorada.

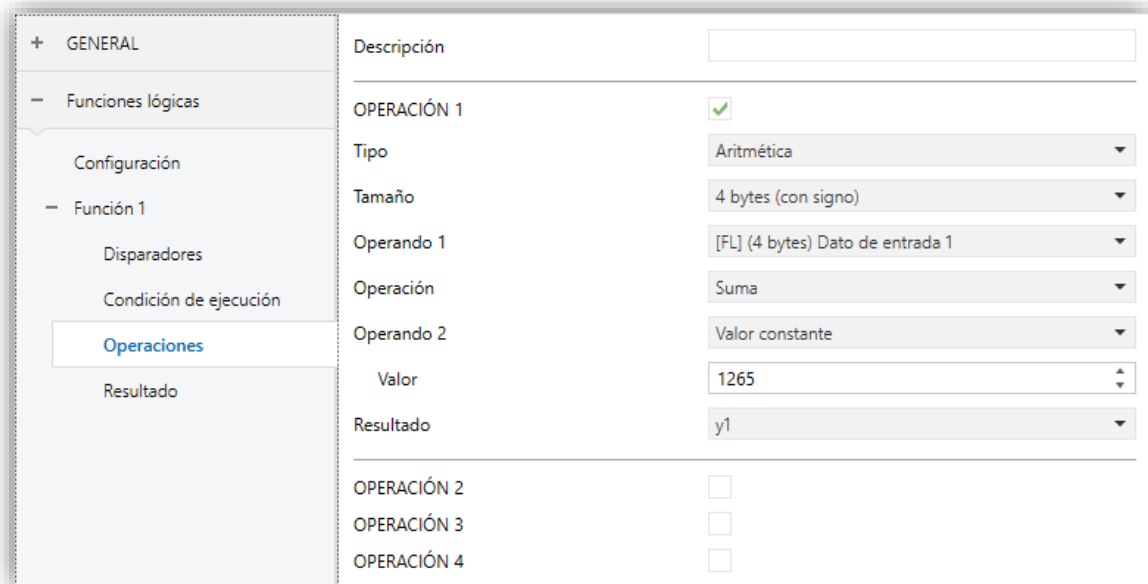


Figura 7. Pestaña “Operaciones”.

En cada operación se pueden configurar los siguientes parámetros:

- **Descripción:** permite especificar una breve descripción (hasta cien caracteres) de la función lógica. Este campo no tiene ninguna implicación práctica; simplemente facilita al integrador la identificación de la función.
- **Operación “i”:** permite habilitar o inhabilitar la operación “i” (1-4). Cada operación habilitada mostrará a su vez los siguientes parámetros:
 - **Tipo:** permite seleccionar el tipo de operación (lógica, aritmética, de comparación o de conversión). Salvo que se seleccione “lógica”, se mostrará un parámetro adicional (“**Tamaño**”) para la selección del tamaño del resultado: “1 bit”, “1 byte (sin signo)”, “1 byte (porcentaje)”, “2 bytes (sin signo)”, “2 bytes (con signo)”, “2 bytes (coma flotante)” y “4 bytes (con signo)”.
 - **Operación:** permite seleccionar la acción que ejecutará la operación “i”. En función del tipo de operación seleccionado (lógica, aritmética, de

comparación o de conversión), este parámetro mostrará unas opciones u otras. Para más información, ver [ANEXO I: Listado de operaciones](#).

- **Operando “j”**: dependiendo de la opción seleccionada en el parámetro anterior, se habilitarán uno o varios parámetros de nombre “Operando j”, mediante los que se puede establecer los valores de entrada (operandos) de la operación. Éstos pueden ser objetos de comunicación, variables internas o valores constantes. Véanse la sección 2.4 y las siguientes.
- **Resultado** de la operación: permite seleccionar la variable interna en la que se almacenará el resultado de la operación. Este resultado intermedio podrá ser utilizado como operando de entrada en las operaciones sucesivas o como resultado final de la función, si se desea.

Nota: *las variables internas son comunes para todas las funciones lógicas. Por ejemplo, si la función 1 almacena un resultado intermedio en la variable interna “n1” y posteriormente la función 2 utiliza esa variable interna como dato de entrada, el valor leído será el que escribió la función 1.*

3.2.4 RESULTADO

Desde esta sección se podrá seleccionar qué variable interna es la que determina el resultado final de la función, de tal modo que tras ejecutar las operaciones que conforman la función se podrá consultar el valor de esta variable y enviarlo al bus a través del objeto “[FL] Función *n* – Resultado”.

+	GENERAL	Tamaño	2 bytes (sin signo)
-	Funciones lógicas	Valor	x1
	Configuración	Condición de envío	es mayor o igual que
-	Función 1	Valor	0
	Disparadores	Modo de envío	Siempre
	Condición de ejecución	Retardo	0
	Operaciones		s
	Resultado		

Figura 8. Pestaña “Resultado”.

- **Tamaño:** establece el tamaño del resultado de la función. Las opciones son: “1 bit”, “1 byte (sin signo)”, “1 byte (porcentaje)”, “2 bytes (sin signo)”, “2 bytes (con signo)”, “2 bytes (flotante)” o “4 bytes (con signo)”.
- **Valor:** determina la variable interna cuyo valor se enviará al bus, a través del objeto de resultado de la función, al término de las operaciones.
- **Condición de envío:** establece restricciones sobre los resultados que se envían al bus, de tal modo que sólo se envíen aquellos que cumplan esa restricción. Las opciones disponibles son:
 - **Para resultados de tamaño 1 bit:**
 - Sin restricciones.
 - Es igual a,
 - No es igual a.
 - **Para resultados de otros tamaños (1 byte, 2 bytes, 4 bytes):**
 - Sin restricciones.
 - Es igual a,

- No es igual a,
- Es mayor que,
- Es mayor o igual que,
- Es menor que,
- Es menor o igual que.

Cuando se elige una restricción, deberá también cumplimentarse el parámetro “**Valor**”. El rango de este parámetro depende del *Tamaño* elegido (ver sección 2.4).

- **Modo de envío:** en este campo se define bajo qué condiciones se enviará el resultado al bus KNX, una vez que las restricciones anteriores se cumplan.

- Siempre.
- Resultado distinto al anterior: sólo se enviará al bus KNX el resultado de la función cuando éste sea distinto del de la ejecución anterior.
- Envío periódico: el objeto de resultado se reenviará al bus repetidamente cada cierto tiempo a partir de ejecutarse la función por primera vez, según se establezca en el parámetro “**Periodicidad**”. El rango de este parámetro depende de la unidad de tiempo elegida: 10 a 600 décimas de segundo, 1 a 3600 segundos, 1 a 1440 minutos o 1 a 24 horas.

Téngase en cuenta que reenviar el resultado periódicamente **no implica que se calcule nuevamente** antes de cada reenvío, por lo que los cambios en los objetos de entrada no alterarán el valor que se envíe al bus; para ello es necesario volver a disparar la función. Nótese además que al disparar la función nuevamente se interrumpirá el envío periódico anterior y se reiniciará el contador del tiempo de período, incluso aunque el nuevo resultado no cumpla las condiciones para enviarse al bus (en tal caso, el anterior envío periódico se interrumpirá igualmente).

- **Retardo:** establece un tiempo de retardo para el envío del resultado tras la ejecución de la función. Si se desea un envío inmediato, deberá establecerse el valor “0”.

El rango de este parámetro depende de la unidad de tiempo elegida: 0 a 600 décimas de segundo, 0 a 3600 segundos, 0 a 1440 minutos, o 0 a 24 horas.

Tal como se ha explicado, el retardo se aplica en el momento de enviar el objeto al bus: la función en sí se ejecuta inmediatamente después de la llamada, independientemente de que el resultado se envíe o no en diferido, por lo que éste no estará afectado por los cambios de valor de los operandos que tengan lugar durante el tiempo de retardo.

Nota: *en caso de recibirse una nueva orden de disparo de una función cuyos resultados aún no se hayan enviado (debido al retardo), ésta se ejecutará de nuevo, obviando el resultado pendiente de envío y reiniciándose la temporización. En el caso de que el nuevo resultado no cumpla las condiciones para enviarse al bus, el envío pendiente se cancelará igualmente.*

ANEXO I: LISTADO DE OPERACIONES

OPERACIONES EN LÓGICA BOOLEANA (1 BIT)

- ID (identidad)

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
0	-	0
1	-	1

- AND (producto lógico o conjunción)

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
0	0	0
0	1	0
1	0	0
1	1	1

- OR (suma lógica o disyunción)

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
0	0	0
0	1	1
1	0	1
1	1	1

- XOR (OR exclusivo)

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
0	0	0
0	1	1
1	0	1
1	1	0

- NOT (negación)

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
0	-	1
1	-	0

• NAND (AND negada)

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
0	0	1
0	1	1
1	0	1
1	1	0

• NOR (OR negada)

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
0	0	1
0	1	0
1	0	0
1	1	0

• NXOR (XOR negada)

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
0	0	1
0	1	0
1	0	0
1	1	1

OPERACIONES ARITMÉTICAS

• ID (identidad)

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
$v1$	-	$v1$

• SUMA

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
$v1$	$v2$	$v1 + v2$

• RESTA

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
$v1$	$v2$	$v1 - v2$

• **MULTIPLICACIÓN**

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
<i>v1</i>	<i>v2</i>	$v1 * v2$

• **DIVISIÓN**

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
<i>v1</i>	<i>v2</i>	$v1 / v2$

• **MÁXIMO**

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
<i>v1</i>	<i>v2</i>	$\text{máx}(v1, v2)$

• **MÍNIMO**

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
<i>v1</i>	<i>v2</i>	$\text{mín}(v1, v2)$

Nota: se recomienda leer la sección Observaciones varias para más información sobre determinados casos específicos y desbordamientos.

OPERACIONES DE COMPARACIÓN

• **MAYOR**

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
<i>v1</i>	<i>v2</i>	$1 \leftrightarrow v1 > v2.$ 0 en otro caso.

• **MAYOR O IGUAL**

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
<i>v1</i>	<i>v2</i>	$1 \leftrightarrow v1 \geq v2.$ 0 en otro caso.

• **MENOR**

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
<i>v1</i>	<i>v2</i>	$1 \leftrightarrow v1 < v2.$ 0 en otro caso.

● **MENOR O IGUAL**

Operando 1	Operando 2	Resultado
v1	v2	1 ↔ v1 ≤ v2. 0 en otro caso.

● **IGUAL**

Operando 1	Operando 2	Resultado
v1	v2	1 ↔ v1 = v2. 0 en otro caso.

● **DISTINTO**

Operando 1	Operando 2	Resultado
v1	v2	1 ↔ v1 ≠ v2. 0 en otro caso.

Nota: se recomienda leer la sección Observaciones varias para más información sobre determinados casos específicos y desbordamientos.

OPERACIONES DE CONVERSIÓN

Operando de un bit

	1 byte (sin signo)	2 bytes (sin sig.)	2 bytes (con s.)	2 bytes (flot.)	4 bytes (con s.)
0	0	0	0	0,00	0
1	1	1	1	1,00	1

Operando de un byte sin signo

	1 bit	2 bytes (sin signo)	2 bytes (con s.)	2 bytes (flot.)	4 bytes (con s.)
0	0	0	0	0,00	0
10	1	10	10	10,00	10
180	1	180	180	180,00	180
255	1	255	255	255,00	255

Operando de un byte tipo porcentaje

	1 bit	1 byte (sin signo)	2 bytes (sin signo)	2 bytes (con s.)	2 bytes (flot.)	4 bytes (con s.)
0	0	0	0	0	0,00	0
10	1	10	10	10	10,00	10
80	1	80	80	80	80,00	80
100	1	100	100	100	100,00	100

Operando de dos bytes sin signo

	1 bit	1 byte (sin signo)	2 bytes (con s.)	2 bytes (flot.)	4 bytes (con s.)
0	0	0	0	0,00	0
10	1	10	10	10,00	10
500	1	255	500	500,00	500
65535	1	255	32767	65535,00	65535

Operando de dos bytes con signo

	1 bit	1 byte (sin signo)	2 bytes (sin s.)	2 bytes (flot.)	4 bytes (con s.)
-32768	0	0	0	-32768,00	-32768
-12000	0	0	0	-12000,00	-12000
0	0	0	0	0,00	0
12345	1	255	12345	12345,00	12345

Operando de dos bytes en coma flotante

	1 bit	1 byte (sin signo)	2 bytes (sin s.)	2 bytes (con s.)	4 bytes (con s.)
-671088,64	0	0	0	-32768	-671088
-321654,98	0	0	0	-32768	-321654
0,5	0	0	0	0	0
0,00	0	0	0	0	0
12345,67	1	255	12345	12345	12345

Operando de cuatro bytes con signo

	1 bit	1 byte (sin signo)	2 bytes (sin s.)	2 bytes (con s.)	2 bytes (flot.)
-2147483648	0	0	0	-32768	0xF800 (*)
-1	0	0	0	-1	-1,00
0	0	0	0	0	0,00
123456	1	255	65535	32767	123456,00
2147483647	1	255	65535	32767	0x7FFF (*)

(*) Los valores mínimo y máximo de dos bytes en coma flotante permitidos en el estándar KNX se operan, respectivamente, como $-\infty$ y $+\infty$, tal según se indica en Observaciones varias.

OBSERVACIONES VARIAS

Como ya se ha explicado, el módulo de funciones lógicas puede trabajar con los siguientes tipos de datos:

- Binarios: **0** y **1**.
- Enteros sin signo.
 - Un byte: **0 – 255**.
 - Dos bytes: **0 – 65535**
- Valores de porcentaje (un byte): **0 – 100**.
- Enteros con signo.
 - Dos bytes: **-32768 – 32767**.
 - Cuatro bytes: **-2147483648 – 2147483647**.
- Coma flotante (dos bytes): **-671088,64 – 671760,96**.

Estos operandos podrán ser objetos de comunicación, variables internas donde ir almacenando resultados intermedios, o incluso, en según qué operaciones, constantes numéricas establecidas por parámetro en ETS.

Por otro lado, es importante tener presentes las siguientes observaciones:

- Los **desbordamientos en las operaciones matemáticas** se atienden devolviendo el valor del límite que haya sido superado. Por ejemplo, una suma de tamaño un byte entre los valores 250 y 10 devolverá 255, al ser 255 el límite del rango permitido para un byte.
- La **pérdida de tensión** en el bus no supone la pérdida de los valores que tuvieran los objetos ni las variables internas: a la vuelta de la tensión, recuperan el valor anterior.
- Las operaciones de **división entre cero** no devuelven ningún resultado; si alguna función contiene una operación de división entre cero, se interrumpirá al término de la anterior operación.

- La **multiplicación y la división de valores de porcentaje** entre sí se efectúa conforme a los siguientes ejemplos:

- Dos constantes:

La multiplicación y la división de dos constantes entre sí se hace asumiendo que una es un entero. Por ejemplo, si el operando 1 se fija en “25” y el operando 2 en “2”, el resultado es “50%” (nótese que el resultado de esta operación se conoce de antemano).

- Una constante y un objeto:

Esta opción se destina a los casos en que se desee, por ejemplo, que la posición de una persiana sea siempre “k” veces (siendo “k” una constante) la de otra. Por ejemplo, si se le asigna a “k” el valor “3” y la posición de la segunda persiana es “30%”, la primera se moverá al 90%.

- Dos objetos:

- Si se desea, como en el caso inmediatamente anterior, que una persiana B adquiera “k” veces la posición de otra persiana A, siendo “k” en este caso un objeto entero de un byte sin signo (y no una constante), será necesario convertir “k” en una variable interna, para después multiplicarla por el objeto de estado de B.
- En la misma situación, si “k%” es el valor de un objeto de un byte tipo porcentaje (y no una constante), para que B adquiera un “k%” de la posición de A (ej.: el 50% de 40%, esto es, 20%), será necesario convertir ambos en variables de dos bytes coma flotante, multiplicarlas, y finalmente convertir el resultado en un valor de porcentaje de un byte.

- Los límites superior (más positivo) e inferior (más negativo) del rango de valores en coma flotante permitidos por KNX (esto es, -671088,64 representado como 0xF800, y +670760,96 representado como 0x7FFF) se operan como **$+\infty$ (más infinito)** y **$-\infty$ (menos infinito)** respectivamente, como en los siguientes ejemplos:

- $(670760,96) * (-671088,64) = (-671088,64)$
- $(-671088,64) * (-671088,64) = (670760,96)$
- $(670760,96) * (34,00) = (670760,96)$

- $(670760,96) * (-34,80) = (-671088,64)$
 - $(670760,96) / (65,20) = (670760,96)$
 - $(670760,96) / (-341,12) = (-671088,64)$
 - $(120000,00) / (670760,96) = (0,00)$
 - $(-671088,64) / (670760,96) = (-1,00)$
 - $(-671088,64) / (3500,66) = (-671088,64)$
- En caso de parametrizar un **período de envío** del resultado y, además, un **tiempo de retardo**, éste sólo se aplicará antes del primer envío al bus, al término de las operaciones. Transcurrido el retardo se enviará el resultado al bus y empezará a contar el período de reenvío. En los sucesivos reenvíos no se aplicará el retardo, a no ser que se dispare de nuevo una función y se genere un nuevo resultado que deba enviarse al bus, en cuyo caso se volverá a aplicar el retardo en el primer envío.

Únete y envíanos tus consultas
sobre los dispositivos Zennio:
<http://support.zennio.com>

Zennio Avance y Tecnología S.L.
C/ Río Jarama, 132. Nave P-8.11
45007 Toledo (Spain).

Tel. +34 925 232 002.

www.zennio.com
info@zennio.com



RoHS