

Funciones lógicas (X10)

Módulo de diez funciones lógico-matemáticas

Edición del manual: 1.b

www.zennio.com

Contenido

Actualizaciones del documento.....	3
1 Introducción.....	4
1.1 Módulo de funciones lógicas.....	4
2 Configuración.....	5
2.1 Funcionamiento general.....	5
2.2 Objetos de llamada.....	6
2.3 Operaciones	6
2.4 Objetos de entrada.....	8
2.5 Variables internas.....	8
2.6 Objetos de resultado	8
2.7 Objetos de habilitación (o de “gate”).....	9
3 Parametrización ETS.....	10
3.1 Pantalla general.....	10
3.2 1bit, 1byte, 2bytes.....	11
3.3 Función n.....	11
3.3.1 Llamada.....	12
3.3.2 Operaciones.....	14
3.3.3 Resultado	15
ANEXO I: Listado de operaciones.....	18
Operaciones en lógica booleana (1 bit).....	18
Operaciones aritméticas	19
Operaciones de comparación.....	20
Operaciones de conversión.....	21
Observaciones varias.....	24

ACTUALIZACIONES DEL DOCUMENTO

Version	Changes	Page(s)
1.b	Revisión general de textos y estilos.	-

1 INTRODUCCIÓN

1.1 MÓDULO DE FUNCIONES LÓGICAS

Diversos dispositivos Zennio (como por ejemplo los actuadores de las familias ACTinBOX y MAXinBOX o el economizador de energía KES) 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 un bit, un byte o dos bytes) específicos para tal efecto.

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.

Dependiendo del número de funciones independientes que permita configurar, el módulo de funciones lógicas incorporado en el dispositivo será uno de los siguientes:

- **Módulo X5**: permite definir hasta cinco funciones lógicas diferentes e independientes, cada una de las cuales, a su vez, podrá consistir en hasta cuatro operaciones sucesivas e interrelacionadas.
- **Módulo X10**: permite definir hasta diez funciones lógicas diferentes e independientes, cada una de las cuales, a su vez, podrá consistir en hasta cuatro operaciones sucesivas e interrelacionadas.

Puede consultarse el manual de usuario del dispositivo Zennio específico para confirmar el módulo concreto (X5 ó X10) que incorpora.

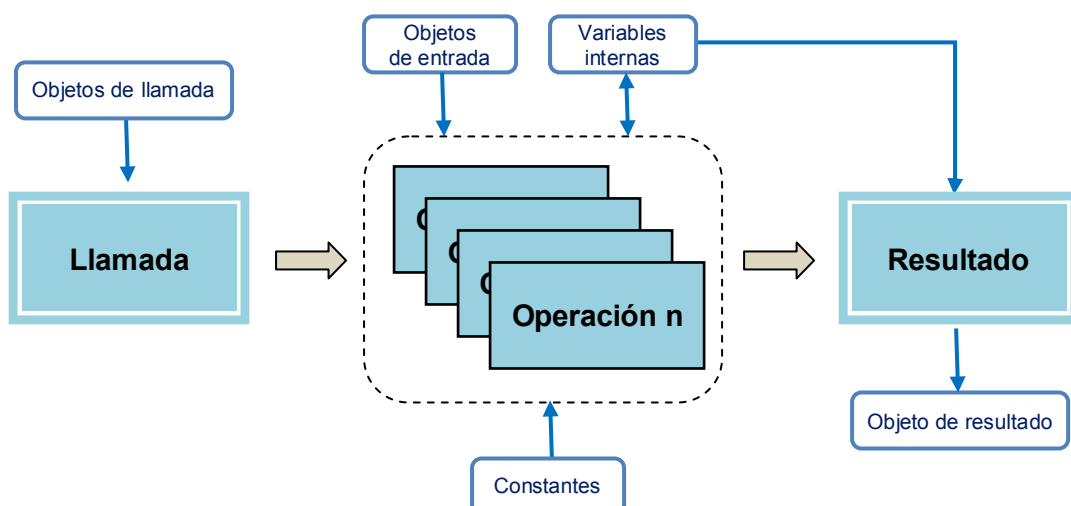
Nota: *a partir de aquí, este manual se centrará en el módulo X10. Para información específica sobre el módulo X5, por favor consúltese el correspondiente manual de usuario, disponible en la página web de Zennio.*

2 CONFIGURACIÓN

2.1 FUNCIONAMIENTO GENERAL

El módulo de funciones lógicas X10 permite habilitar y configurar hasta diez funciones lógicas independientes. El funcionamiento de cada una de estas diez funciones lógicas se divide en tres 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, los cuales, cada vez que actualicen su valor desde el bus, dispararán automáticamente la ejecución de la función.
- **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 que almacenará 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.



2.2 OBJETOS DE LLAMADA

Para cada función, el integrador podrá disponer de hasta ocho objetos de llamada (de un bit, un byte o dos 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.

2.3 OPERACIONES

Cada función lógica consiste en la ejecución de hasta cuatro operaciones consecutivas. Las operaciones disponibles pueden clasificarse de la siguiente forma:

- **Lógica:** ID, NOT, AND, OR, XOR, NAND, NOR y XNOR.
- **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 X10 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**.
- Enteros sin signo (dos bytes): **0 – 65535**.
- Números fraccionarios en coma flotante (dos bytes): **0,00 – 120,00**.
- Para más información sobre estas operaciones, sobre la conversión entre tamaños y sobre el truncado de valores, consúltese el

ANEXO I: Listado de operaciones.

2.4 OBJETOS DE ENTRADA

Podrán habilitarse múltiples objetos específicos para usarlos con las funciones lógicas:

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

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

2.5 VARIABLES INTERNAS

Asimismo, el integrador podrá disponer de:

- 32 variables internas de un bit,
- 16 variables internas de un byte,
- 16 variables internas de dos bytes.

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

2.6 OBJETOS DE RESULTADO

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

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.

2.7 OBJETOS DE HABILITACIÓN (O DE “GATE”)

Aparte de lo ya explicado, existe la opción de inhabilitar y volver a habilitar cada función lógica por separado en tiempo de ejecución, mediante el envío de un uno o un cero (según se configure) a través de su objeto de “gate” particular, que deberá ser alguno de los 32 objetos de entrada de un bit mencionados en 0. Mientras esté inhabilitada, la función ignorará cualquier valor recibido a través de los objetos de llamada, no ejecutándose en ningún caso.

3 PARAMETRIZACIÓN ETS

3.1 PANTALLA GENERAL

La pestaña general de parámetros del módulo de funciones lógicas X10 contiene las opciones que se muestran en la figura (téngase en cuenta que puede haber ligeras diferencias de un dispositivo a otro).

Nota: puede ser que la pestaña general de parámetros del módulo de funciones lógicas no se muestre por defecto en ETS y resulte necesario habilitarla desde la pestaña General de parámetros del propio dispositivo. Consúltese el manual de usuario del dispositivo específico para más detalles.

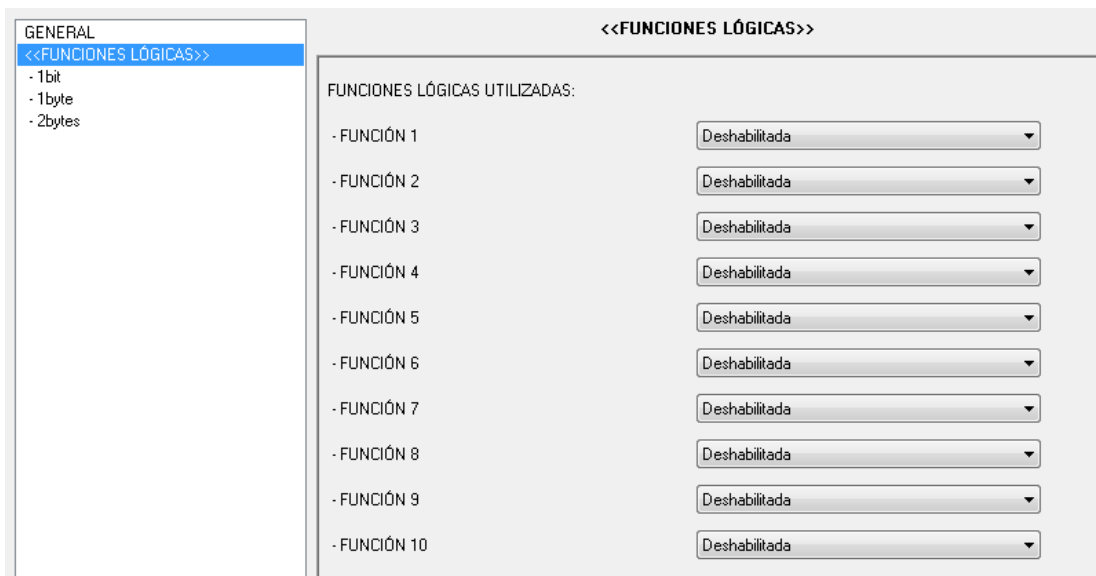


Figura 1. Pestaña general del módulo de funciones lógicas X10.

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.

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

3.2 1bit, 1byte, 2bytes

Estas tres pestañas, que sí son visibles en todo momento, permiten al integrador habilitar, uno por uno, los objetos de entrada (de un bit, un byte o dos byte) que las funciones necesiten utilizar, bien como operandos, como objetos de llamada, etc. Estos objetos tienen la siguiente nomenclatura:

- “[FL] Dato (1bit) *n*”,
- “[FL] Dato (1byte) *n*”,
- “[FL] Dato (2bytes) *n*”.

Como ya se vio en 0, se podrá habilitar hasta 32 objetos de un bit, 16 de un byte y 16 de dos bytes. Como puede verse, ninguno está habilitado por defecto.

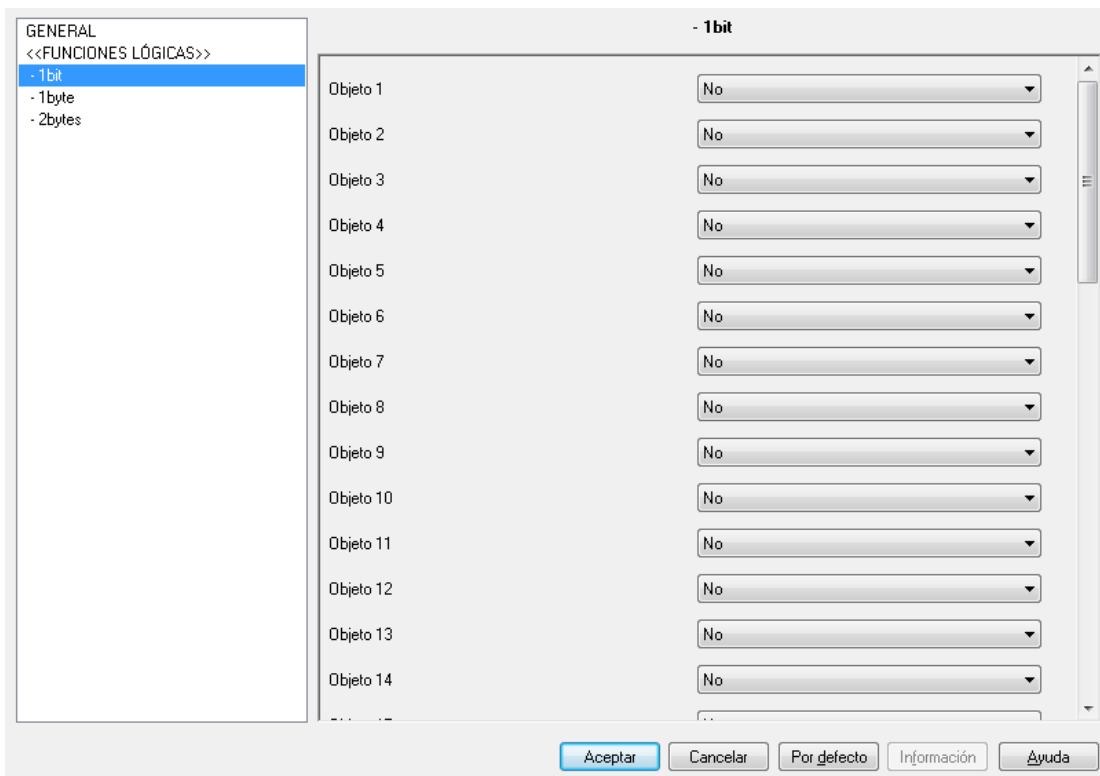


Figura 2. Habilitación de los objetos de entrada (1 bit).

3.3 FUNCIÓN *n*

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

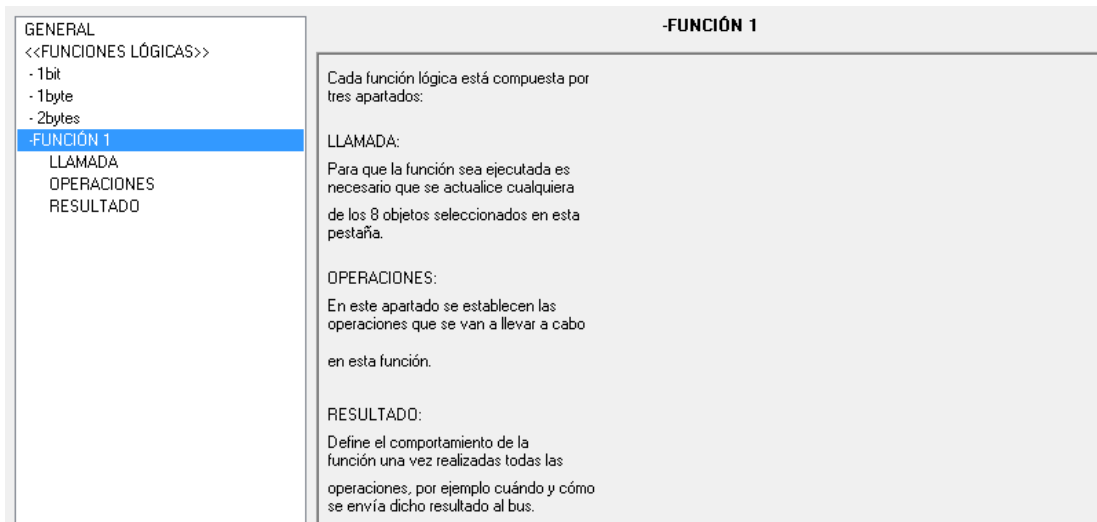


Figura 3. Función 1

3.3.1 LLAMADA

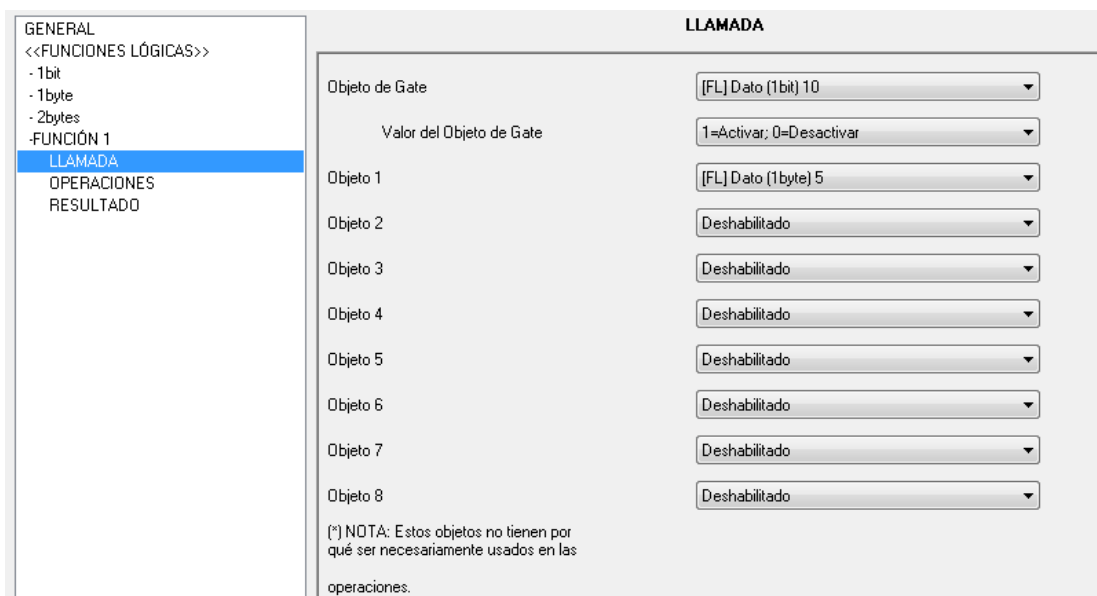


Figura 4. Pestaña "Llamada"

Desde esta sección se podrá seleccionar hasta **ocho objetos** (que deberán haberse habilitado previamente; ver sección 3.2), que actuarán como objetos de llamada. Los objetos de llamada serán los responsables de disparar la ejecución de la función cada vez que alguno de ellos reciba algún valor desde el bus. Igualmente, si se utiliza un mismo objeto como objeto de llamada de varias funciones, todas ellas se dispararán simultáneamente cuando se escriba algún valor en el objeto.

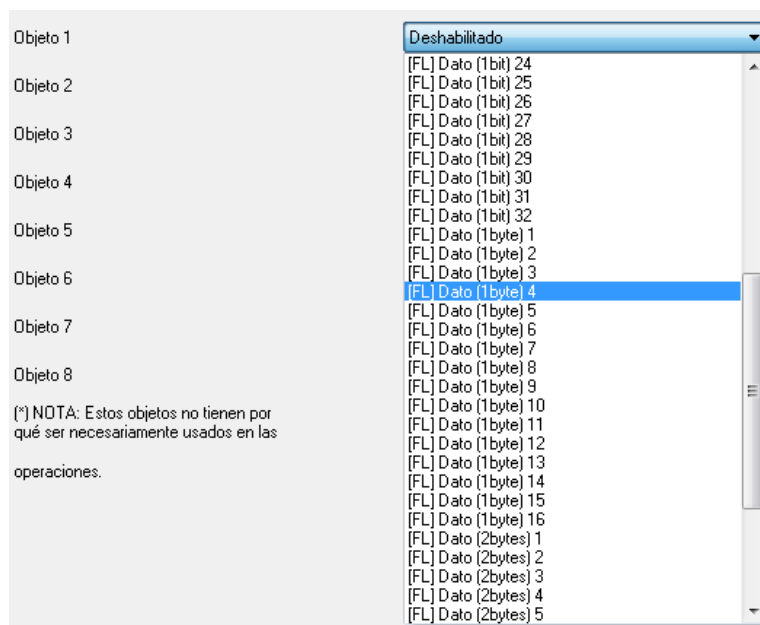


Figura 5. Selección de los objetos de llamada

Además, desde esta pestaña el integrador podrá determinar cuál de los objetos activos (ver sección 3.2) actuará como objeto de habilitación de la función (parámetro “**Objeto de gate**”) y qué significado se asignará a cada posible valor (parámetro “**Valor del objeto de gate**”, cuyas opciones son “[0=Activar; 1=Desactivar]” y “[1=Activar; 0=Desactivar]”), según se explicó en 2.7. Es posible no configurar ningún objeto de llamada, eligiendo para ello la opción “Deshabilitado” (seleccionada por defecto).

Notas:

- *Tras descarga en el dispositivo, los objetos de gate se inician con el valor correspondiente a “desactivado” (“0” ó “1”, según corresponda), mientras que tras un fallo de bus recuperarán los valores que tuvieron antes del fallo.*
- *Si se utiliza como objeto de gate un objeto que ya intervenga de algún modo en esa (o cualquier otra) función, por ejemplo como operando o como resultado, deberá prestarse atención a los valores que recibe, pues podrían habilitar/inhabilitar la función inesperadamente.*

3.3.2 OPERACIONES

Figura 6. Operaciones

En esta sección se definen las operaciones que componen la función, mediante 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) y el tamaño de los operandos que intervengan (un bit, entero sin signo de un byte, entero sin signo de dos bytes, coma flotante de dos bytes). Ver sección 2.3.
 - **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.
- **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 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.

3.3.3 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 consultará el valor de esta variable y se enviará al bus a través del objeto “[FL] RESULTADO Función n (tamaño)”, en donde “tamaño” dependerá de la configuración que se seleccione.

Figura 7. Resultado

- **Tipo:** establece el tamaño del resultado de la función. Las opciones son “1 bit”, “1 byte”, “2 bytes (entero sin signo)” y “2 bytes (coma flotante)”.
- **Valor:** determina la variable interna cuyo valor será enviado al bus, a través del objeto de resultado de la función, al término de las operaciones.

- **Restricción:** establece restricciones sobre los resultados que se deben enviar al bus. Las opciones disponibles son:

- **Un bit:**

- Sin restricción (se envía tanto el 0 como el 1),
- Sólo se envía el 0,
- Sólo se envía el 1.

- **Un byte / Dos bytes (entero sin signo) / Dos bytes (coma flotante):**

- Sin restricción,
- Enviar sólo valores iguales al de referencia,
- Enviar sólo valores distintos del de referencia,
- Enviar sólo valores menores que el de referencia.
- Enviar sólo valores mayores que el de referencia,

El valor de referencia se especifica mediante el parámetro “**Valor de referencia**”, y podrá tomar valores entre 0 y 255 (para el caso de los enteros de un byte sin signo), entre 0 y 65535 (para el caso de los enteros de dos bytes sin signo) y entre 0 y 1200 décimas, es decir, entre 0 y 120,0 (para el caso de coma flotante de dos bytes).

- **Envío:** en este campo se define bajo qué condiciones se enviará el resultado al bus KNX.

- Cada vez que se ejecuta la función.

- Cambio de resultado final: sólo se enviará al bus KNX el resultado de la función cuando éste sea distinto del de la ejecución anterior.

Nota: *la inhabilitación temporal de una función mediante su objeto de gate no tiene efecto sobre las restricciones: cuando se habilite nuevamente, la función seguirá recordando cuál fue el último resultado que envió.*

- Periódico: el objeto de resultado se enviará al bus repetidamente (actualizado en cada caso) cada cierto tiempo a partir de ejecutarse la función por primera vez, según se establezca en el parámetro “**Período de envío**” (entre 0 y 65535 segundos).

Téngase en cuenta que a partir del primer envío, el objeto ya sólo se transmitirá al expirar el período, por lo que las respuestas a las **llamadas sucesivas** a la función podrían no ser inmediatas: el temporizador no se reinicia cada vez que se llama nuevamente a la función (sí lo hará en caso de producirse una pérdida de tensión en el dispositivo).

Nota: *si mientras estaba en curso un envío periódico se recibe el valor de inhabilitación de la función mediante el objeto de gate, el envío quedará interrumpido y no se reanudará hasta que la función se habilite y sea llamada nuevamente.*

- **Retardo:** establece un tiempo de retardo (entre 0 y 65535 segundos) para el envío del objeto tras la ejecución de la función. Si se desea un envío inmediato, deberá establecerse el valor "0".

Como se ve, el retardo se aplica en el momento de enviar el objeto al bus: la función se ejecuta inmediatamente después de llamada independientemente de que el resultado se envíe en diferido, y sin verse afectada por los cambios de valor de los operandos que tengan lugar durante el tiempo de retardo.

Nota: *si durante el tiempo de retardo del envío se recibe el valor de inhabilitación de la función mediante el objeto de gate, el envío del resultado quedará cancelado y no tendrá lugar hasta que la función se habilite y sea llamada de nuevo.*

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

Operando 1	Operando 2	Resultado
v1	v2	$v1 * v2$

• DIVISIÓN

Operando 1	Operando 2	Resultado
v1	v2	$v1 / v2$

• MÁXIMO

Operando 1	Operando 2	Resultado
v1	v2	$\text{máx}(v1, v2)$

• MÍNIMO

Operando 1	Operando 2	Resultado
v1	v2	$\text{mín}(v1, v2)$

Nota: se recomienda leer la sección *Observaciones varias* para más información sobre determinados casos inusuales.

OPERACIONES DE COMPARACIÓN

• MAYOR

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

• MAYOR O IGUAL

Operando 1	Operando 2	Resultado
v1	v2	$1 \leftrightarrow v1 \geq v2.$ 0 en otro caso.

• MENOR

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

• MENOR O IGUAL

Operando 1	Operando 2	Resultado
v1	v2	1 ↔ $v1 \leq v2$. 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 ↔ $v1 \neq v2$. 0 en otro caso.

Nota: se recomienda leer la sección *Observaciones varias* para más información sobre determinados casos inusuales.

OPERACIONES DE CONVERSIÓN

Conversión a un bit

• 1 byte → 1 bit

Operando 1	Operando 2	Resultado
0	-	0
1-255	-	1

• 2 bytes (entero sin signo) → 1 bit

Operando 1	Operando 2	Resultado
0	-	0
1 - 65535	-	1

• 2 bytes (coma flotante) → 1 bit

Operando 1	Operando 2	Resultado
≤ 0,00	-	0
0,01 – 120,00	-	1

Conversión a un byte

- 1 bit → 1 byte

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

- 2 bytes (entero sin signo) → 1 byte

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
0	-	0
1	-	1
...
255	-	255
256 - 65535	-	255

- 2 bytes (coma flotante) → 1 byte

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
≤ 0,00	-	0
0,01	-	0
...
0,10	-	1
0,11	-	1
...
25,49	-	254
≥ 25,50	-	255

Conversión a dos bytes (entero sin signo)

- 1 bit → 2 bytes (entero sin signo)

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

• 1 byte → 2 bytes (entero sin signo)

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
0	-	0
1	-	1
...
255	-	255

• 2 bytes (coma flotante) → 2 bytes (entero sin signo)

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
≤ 0,00	-	0
0,01	-	0
...
0,10	-	1
0,11	-	1
...
119,98	...	1199
119,99	-	1199
≥ 120,00	-	1200

Conversión a dos bytes (coma flotante)

• 1 bit → 2 bytes (coma flotante)

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
0	-	0,00
1	-	0,10

• 1 byte → 2 bytes (coma flotante)

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
0	-	0
1	-	1
...
254	-	25,40
255	-	25,50

- **2 bytes (entero sin signo) → 2 bytes (coma flotante)**

<i>Operando 1</i>	<i>Operando 2</i>	<i>Resultado</i>
0	-	0
1	-	0
...
1199	-	119,90
≥ 1200	-	120,00

OBSERVACIONES VARIAS

Como ya se ha explicado, el módulo X10 de funciones lógicas puede trabajar con los siguientes tipos de datos:

- Binarios: **0 y 1.**
- Enteros sin signo (un byte): **0 – 255.**
- Enteros sin signo (dos bytes): **0 – 65535.**
- Números fraccionarios en coma flotante (dos bytes): **0,00 – 120,00.**

Estos operandos podrán ser objetos de comunicación, variables internas donde poder 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:

- Las operaciones en coma flotante **truncan siempre los valores fraccionarios a décimas de unidad.** Por ejemplo, si el objeto “[FL] Dato (2bytes) 1” tiene valor de 2,37 y se hace una operación “ID” (identidad) sobre él, el resultado será 2,30.
- Igualmente, las operaciones en coma flotante **truncan siempre a cero cualquier valor de entrada y cualquier resultado negativos.** Por ejemplo, si el objeto “[FL] Dato (2bytes) 1” vale -5,00 y el objeto “[FL] Dato (2bytes) 2” vale +10,00, al sumarlos se obtendrá +10,00. Igualmente, el resultado del segundo al primero será 0,00, no -15,00 ni tampoco -10,00.

- La **pérdida de tensión** en el bus no supone la pérdida de los valores que tuvieron los objetos ni las variables internas: a la vuelta de la tensión, recuperan el valor anterior.
- Las operaciones de **multiplicación y división están destinadas a operar un objeto** (de temperatura, por ejemplo) **y una constante numérica** (configurable en ETS), **o una variable interna y una constante**. Podrían llegar a observarse resultados matemáticamente incorrectos o inesperados debido a desbordamientos y otras limitaciones, si se multiplica o divide un objeto/variable por otro objeto/variable, a pesar de que sí es posible tal parametrización.
- Las operaciones de **división entre cero** no devuelven ningún resultado.

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

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

Tel. +34 925 232 002.

Fax. +34 925 337 310.

www.zennio.com

info@zennio.com



RoHS