



**SKX OPEN**

**SKX ADVANCE**

**ZN1RX-SKXOPEN**



Edición 2  
Versión 1.1

## ÍNDICE

1.	Introducción.....	3
1.1.	Interfaz SKX .....	3
1.2.	Instalación SKX .....	5
1.3.	Programa de aplicación: SKX Advance.....	5
1.3.1.	Especificaciones básicas SKX Advance .....	6
1.4.	Diferencias entre las aplicaciones SKX Open y SKX Advance .....	6
1.5.	Ventajas del producto y la aplicación .....	7
2.	Parametrización ETS.....	8
2.1.	Configuración de la comunicación .....	8
2.2.	Configuración de tramas.....	12
2.2.1.	Definición de tramas.....	12
2.2.2.	Tramas especiales.....	12
2.2.3.	Caracteres especiales.....	15
2.3.	Grupos de parámetros.....	18
2.3.1.	Objetos de 1 bit .....	19
2.3.2.	Objetos de 1 byte.....	21
2.3.3.	Objetos de 14 bytes .....	26
2.4.	Objetos de error .....	28
2.4.1.	Ejemplos de errores .....	29
2.5.	Ejemplos de configuración.....	34
2.5.1.	Objetos de 1 bit .....	34
2.5.2.	Objetos de 1 byte.....	35
2.5.3.	Objetos de 14 bytes .....	38
3.	Resumen del producto .....	40

# 1. INTRODUCCION

## 1.1. INTERFAZ SKX

**SKX** es un interfaz **ZENNIO** que permite la conexión del bus KNX con otros dispositivos a través de una comunicación serie de tipo **RS-232** bidireccional.



Figura 1.1. Esquema conexionado SKX

### Nomenclatura

Se definen a continuación los términos utilizados con más frecuencia a lo largo de este manual:

- **SKX:** a partir de este momento se denominará SKX al interfaz SKX Open, para no generar confusiones entre el hardware y el programa de aplicación del mismo nombre.
- **SKX Advance:** programa de aplicación que se descarga en SKX y que permite gestionar la comunicación KNX – RS232.
- **RS-232:** tipo de comunicación serie.
- **Equipo terminal (de datos):** dispositivo externo que se está integrando por el puerto serie RS-232.

A continuación se presentan las características más relevantes del interfaz, así como un esquema de sus elementos (figura 1.2):

- Tamaño reducido: 45 x 45 x 14 mm.
- Varias velocidades de comunicación y mecanismos de detección de errores.
- Ideal para aplicaciones M2M.
- Salvado de datos tras pérdida de alimentación.

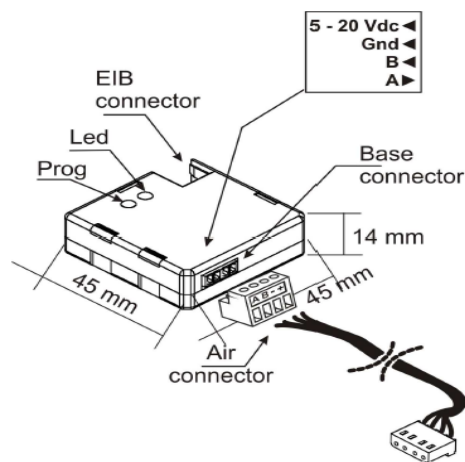


Figura 1.2. SKX. Esquema de elementos

El botón "**Prog**" se utiliza para situar a SKX en modo programación. Si este botón se mantiene pulsado en el momento en que se aplica la tensión de bus, SKX entra en modo seguro.

SKX incorpora un **LED bicolor** en la parte frontal del hardware. Esta señal luminosa tiene una doble funcionalidad. Por un lado, si está iluminando en rojo, indica que el interfaz se encuentra en modo programación y si el LED parpadea en color rojo cada 0.5 segundos, el interfaz se encontrará en modo seguro. Por otro lado, el LED puede actuar como **indicador de transmisión**. Si se ilumina en verde, estará indicando que se han enviado o recibido datos a través del puerto serie correspondiente, quedándose encendido durante 0.3 segundos cada vez que hay comunicación.

Este interfaz permite ser programado con dos programas de aplicación diferentes: **SKX Open** y **SKX Advance**, cuyas diferencias se explican en el apartado 1.4.

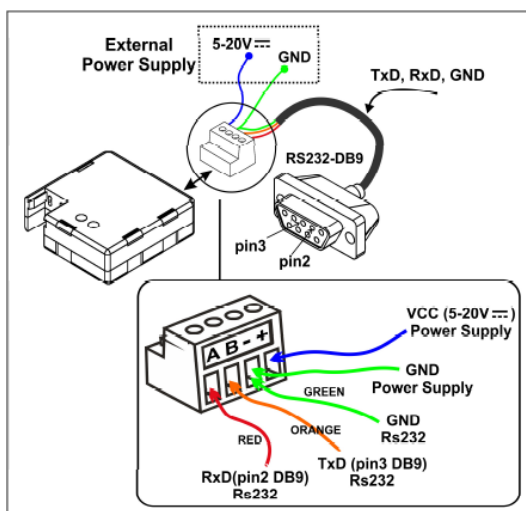
## 1.2. INSTALACIÓN SKX

La instalación de SKX se realiza de manera muy **sencilla**: basta con unir el interfaz al bus KNX a través de su conector específico, quedando listo para ser programado.

SKX no necesita alimentación externa; es capaz de funcionar exclusivamente con la alimentación propia del bus KNX. Sin embargo, será necesario alimentar el bus RS-232 de una manera independiente a la del bus KNX (normalmente, la alimentación necesaria suele obtenerse del propio equipo terminal conectado al bus serie).

La conexión de SKX con RS-232 se realiza a través de una clema específica, que facilita su manipulación e instalación.

En la siguiente figura puede verse un esquema típico de conexión de **SKX – RS232**.



Clema SKX Open	BUS RS232
A	RSA
B	RSB
-	Tierra
+	+12V

Figura 1.3. Conexión SKX a RS-232

## 1.3. PROGRAMA DE APLICACIÓN: SKX ADVANCE

**SKX Advance** es una aplicación que se descarga en el interfaz **SKX** y cuya principal función es la de gestionar la comunicación entre KNX y el protocolo RS-232, permitiendo configurar toda la información intercambiada entre el bus y el equipo terminal. Este intercambio de información es **bidireccional**, es decir, es posible enviar datos desde el bus KNX al equipo terminal y viceversa. En la figura 1.4 puede verse un ejemplo de este tipo de comunicación.

SKX Advance puede enviar o recibir cualquier tipo de trama de comunicaciones, siempre que cumpla con los criterios de mensajes RS-232 permitidos (véase apartado 2.2.1).

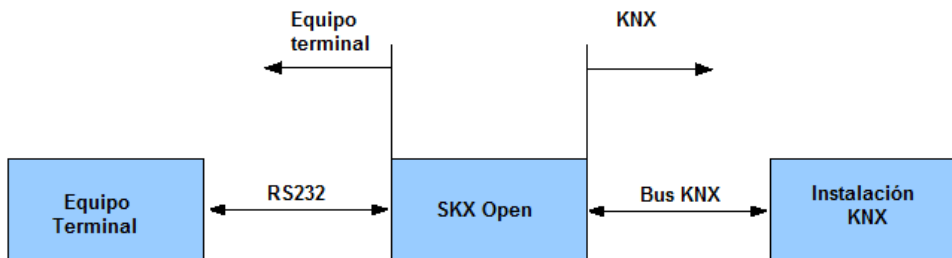


Figura 1.4. Comunicación bidireccional

### 1.3.1. ESPECIFICACIONES BÁSICAS SKX ADVANCE

A continuación se presentan las características fundamentales de la aplicación SKX Advance:

- **Velocidad de transmisión:** 1200, 1400, 4800, 9600 o 19200 baudios.
- **Paridad:** par, impar, sin paridad
- **Tipos de distinción de tramas:** Time Out, byte fin de trama
- **Número de objetos de comunicación:** 65 (40 de 1 bit, 20 de 1 byte y 5 de 14 bytes)
- **Identificación de errores:** varios objetos de 1 bit
- **Longitud del protocolo:** las tramas configuradas por parámetro en SKX Advance pueden tener una longitud mayor de 25 bytes, gracias a la posibilidad de poder usar caracteres especiales en su configuración. En total, se pueden **enviar/recibir hasta 29 bytes** por mensaje.

### 1.4. DIFERENCIAS ENTRE LAS APLICACIONES SKX OPEN Y SKX ADVANCE

El interfaz SKX admite ser programado con dos programas de aplicación diferentes: **SKX Open** y **SKX Advance**.





En la siguiente tabla pueden observarse las principales diferencias entre ambos programas de aplicación:

	SKX OPEN	SKX ADVANCE
<b>Número Objetos</b>	48	65
<b>Tipo de Objetos</b>	1 bit	1 bit 1 Byte 14 Bytes
<b>Tipo de Tramas</b>	Fijas	Variables (en función del valor del objeto)
<b>Longitud de Tramas</b>	Hasta 10 bytes	Hasta 29 bytes
<b>Reconocimiento de tramas</b>	El mensaje entero debe coincidir	Sólo reconocimiento de partes indicadas (omisión del resto)
<b>Checksum</b>	No	Sí
<b>Confirmación (ACK)</b>	No	Sí
<b>Configurabilidad</b>	- Comunicación - Grupos parámetros	- Comunicación - De tramas - Grupos parámetros

Tabla 1.1. Diferencias entre SKX Open y SKX Advance

## 1.5. VENTAJAS DEL PRODUCTO Y LA APLICACIÓN

El hecho de cargar el programa de aplicación SKX Advance en el interfaz SKX trae asociadas, entre otras, las siguientes ventajas:

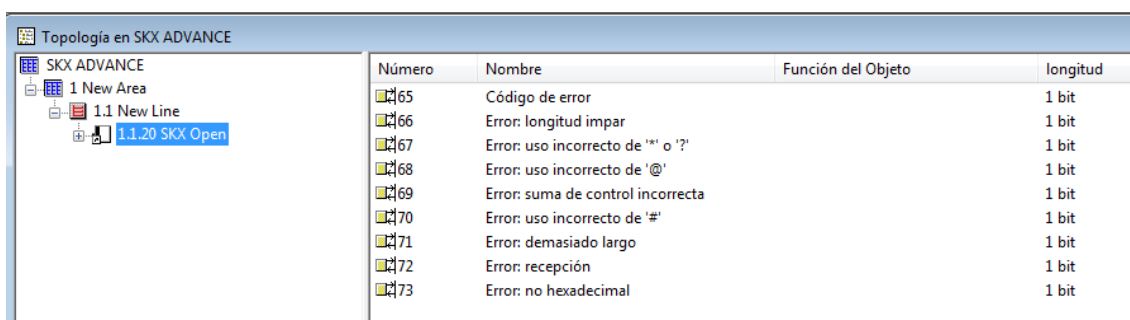
-  Ampliación de la instalación. Permite **integrar** en una instalación KNX otros dispositivos que no posean comunicación KNX pero que dispongan de un puerto serie RS-232.
-  Adaptabilidad. SKX Advance es capaz de adaptarse perfectamente al protocolo que rige el funcionamiento del equipo terminal, independientemente de la complejidad de este. Es decir, SKX Advance **se adapta al equipo terminal** y no al contrario.
-  Versatilidad de la comunicación. SKX Advance posee objetos de diferentes tipos (en concreto, de 1 bit, 1 byte y 14 bytes) con los que intercambia información con el equipo terminal en ambas direcciones (bidireccionalidad).
-  Configurabilidad. los mensajes presentes en la comunicación RS-232 se pueden configurar con todas las características habituales de una comunicación serie, como la posibilidad de habilitar cabeceras, pies de trama, checksum, ACK, etc.; permitiendo así el envío de mensajes dinámicos de longitud variable.

## 2. PARAMETRIZACIÓN ETS

Con el programa de aplicación SKX Advance, es posible integrar cualquier equipo terminal con interfaz RS-232 en una instalación domótica KNX, conociendo las tramas de comunicación que el equipo terminal utiliza para cada orden.

SKX Advance dispone de **65 objetos de comunicación de diferentes tamaños** con los que será posible interactuar entre el protocolo KNX y RS-232, además de varios objetos de 1 bit para detección de errores. Por tanto, cada trama definida llevará siempre asociado un objeto de comunicación. Todos estos objetos se estudiarán con más detenimiento en los apartados correspondientes.

A continuación se presentan los parámetros ETS que se pueden modificar para obtener los resultados esperados, así como el aspecto en que son presentados en ETS al cargar la aplicación.



Número	Nombre	Función del Objeto	longitud
65	Código de error		1 bit
66	Error: longitud impar		1 bit
67	Error: uso incorrecto de '*' o '?'		1 bit
68	Error: uso incorrecto de '@'		1 bit
69	Error: suma de control incorrecta		1 bit
70	Error: uso incorrecto de '#'		1 bit
71	Error: demasiado largo		1 bit
72	Error: recepción		1 bit
73	Error: no hexadecimal		1 bit

Figura 2.1. Objetos de comunicación por defecto

En los siguientes apartados se explica con detalle cada una de las pantallas de configuración que se muestran al hacer click sobre “Editar parámetros”.

### 2.1. CONFIGURACIÓN DE LA COMUNICACIÓN

A continuación se muestra la pantalla de configuración general que aparece por defecto al cargar el programa de aplicación SKX Advance.



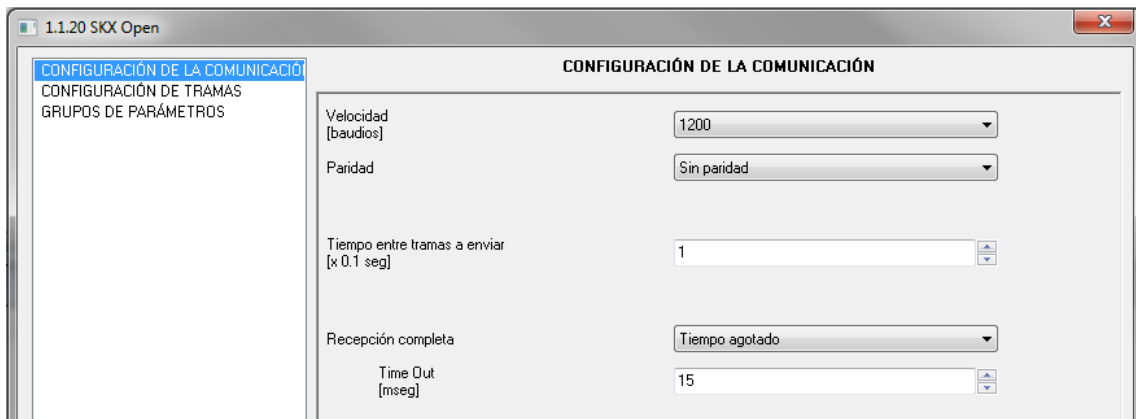


Figura 2.2. SKX Advance: Configuración de la comunicación

En ella se podrán configurar los siguientes parámetros generales relacionados con la transmisión de tramas:

- **Velocidad (baudios):** 1200, 2400, 9600, 19200
- **Paridad:** Sin paridad, par, impar
- **Tiempo entre tramas a enviar (en décimas de segundo):** se trata de un tiempo mínimo configurable de espera entre las tramas a enviar por el puerto serie. Este parámetro se utiliza para separar varias tramas consecutivas enviadas al equipo terminal. Esto puede ser útil, por ejemplo, si se enlaza más de un objeto a la misma dirección de grupo. Así, SKX Advance podrá enviar dichas tramas de manera ordenada, permitiendo una perfecta adquisición e interpretación de los datos por parte del equipo terminal. (**Nota:** *tener en cuenta las características del equipo terminal para configurar el parámetro “Tiempo entre tramas”*).
- **Modo de recepción completa:** indica la forma en que SKX Advance detecta cuándo se han recibido todos los caracteres de la trama. Para el reconocimiento de fin de trama se ofrecen dos opciones:
  - **Tiempo agotado (o Time Out):** se establece un tiempo mínimo (en milisegundos) entre tramas. SKX Advance interpreta que la trama se ha recibido completa una vez transcurrido este tiempo tras recibir el último bit de la misma.
  - **Byte fin de trama:** se establece un byte con un valor específico cuya interpretación sea unívoca, de tal manera que SKX Advance dará por concluida la recepción de la trama una vez que haya recibido este byte. En caso de elegir esta opción, aparecerá una nueva casilla con el parámetro “Time Out de seguridad”, que se define como el tiempo máximo que SKX Advance espera la recepción del byte fin de trama.

**Nota:** A la hora de especificar un Time Out para la recepción de tramas habrá que tener en cuenta la duración del envío de cada byte, especificado en la tabla siguiente:

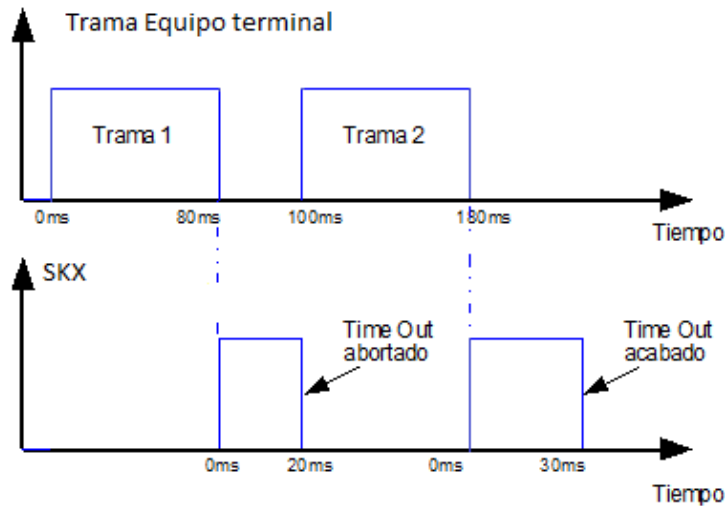
Velocidad (baudios)	Paridad	Tiempo por byte (mseg)
1200	Sí	9.167
	No	8.333
2400	Sí	4.583
	No	4.167
4800	Sí	2.292
	No	2.083
9600	Sí	1.146
	No	1.042
19200	Sí	0.573
	No	0.521

Si, **por ejemplo**, se introduce un Time Out de 3 milisegundos y una velocidad de comunicación de 2400 baudios, SKX Advance provocará un “error de Time Out” por cada dato de entrada, ya que para completar la transferencia de un byte son necesarios 4.583 milisegundos (en caso de haber bit de paridad), valor mayor que el Time Out configurado. (Nota: este error de Time Out no se indica mediante ninguno de los objetos de detección de errores).

Es conveniente leer el siguiente ejemplo para que el concepto de Time Out quede más claro:

**Ejemplo:** Un equipo terminal toma 80ms para enviar su trama completa.

**Primer caso (Time Out demasiado largo):** El “tiempo agotado” definido es de 30ms. El equipo terminal envía una segunda trama inmediatamente después de la primera. La siguiente figura representa este funcionamiento.

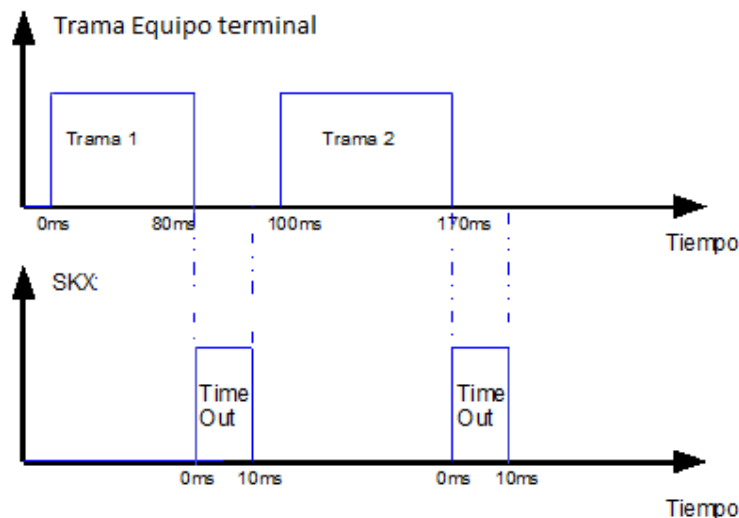


**Configuración de TimeOut incorrecta**

Figura 2.3. Tiempo agotado demasiado largo

Al final de la primera trama, el Time Out empezó a contar, pero llegó otra trama antes de acabar su tiempo, así que se abortó el Time Out y volvió a empezar a contar después del final de la segunda trama. En este caso, como el Time Out llega a su fin (30ms), SKX Advance considera que la trama acaba. Pero como han llegado dos tramas hasta que se ha detectado el fin de trama, SKX Advance considerará la trama desconocida, y no enviará nada.

- **Segundo caso (Time Out bien definido):** El “tiempo agotado” definido en este caso es de 10ms. La siguiente figura representa este funcionamiento:



**Configuración de TimeOut correcta**

Figura 2.4. Tiempo agotado bien definido

En este caso, el Time Out está bien definido y SKX Advance es capaz de reconocer las dos tramas enviadas.

Por tanto, es de especial **importancia definir de manera correcta el Time Out**, teniendo en cuenta el tiempo existente entre trama y trama enviada desde el equipo terminal, pues ya se ha visto en el ejemplo que un tiempo mal definido (demasiado largo o demasiado corto) puede generar errores en la recepción de los mensajes.

## 2.2. CONFIGURACIÓN DE TRAMAS

### 2.2.1. DEFINICIÓN DE TRAMAS

La definición/parametrización de las tramas de comunicación se realiza mediante **caracteres hexadecimales** (2 caracteres por byte); por tanto, sólo están permitidos los caracteres comprendidos entre 0-9 y A-F (excepto en el caso de los caracteres especiales), siendo **obligatorio** que los caracteres A-F estén escritos en **mayúsculas**.

**Nota:** *Es conveniente saber que una trama hexadecimal, por ejemplo: "0x2B 0x7F 0x34" deberá ser introducida en ETS con el formato "2B7F34".*

A la hora de definir una trama de puerto serie, está disponible en ETS una casilla de parametrización de tipo texto que admite **hasta 20 caracteres**. De cara a aumentar el tamaño de las tramas de comunicación y conseguir una mayor versatilidad en la comunicación (datos variables, tramas de tamaño no fijo, etc.) se permite configurar diferentes secciones de la trama. De esta manera, se pueden llegar a transmitir/recibir tramas de **hasta 29 bytes**.

Las tramas de comunicación pueden ser **impares**; lo que debe ser obligatoriamente **par** es el conjunto de caracteres que se introducen por parámetro (2 caracteres por byte).

El incumplimiento de algún requisito de parametrización de tramas provocará que SKX Advance envíe **estados de error** al bus KNX tras ser programado, a través de los objetos de comunicación de error habilitados a tal efecto y que se explicarán más adelante en este manual.

### 2.2.2. TRAMAS ESPECIALES

Existe la posibilidad de configurar una serie de tramas especiales que permiten una **comunicación completa** con el equipo que se desea integrar en la instalación KNX mediante RS-232.

La utilización de estas tramas es completamente **opcional**. El objetivo de las mismas es proveer de los mecanismos suficientes para la generación de mensajes dinámicos en un protocolo de comunicación, permitiendo el uso de cabeceras, pies, subtramas y el envío de asentimientos automatizados.

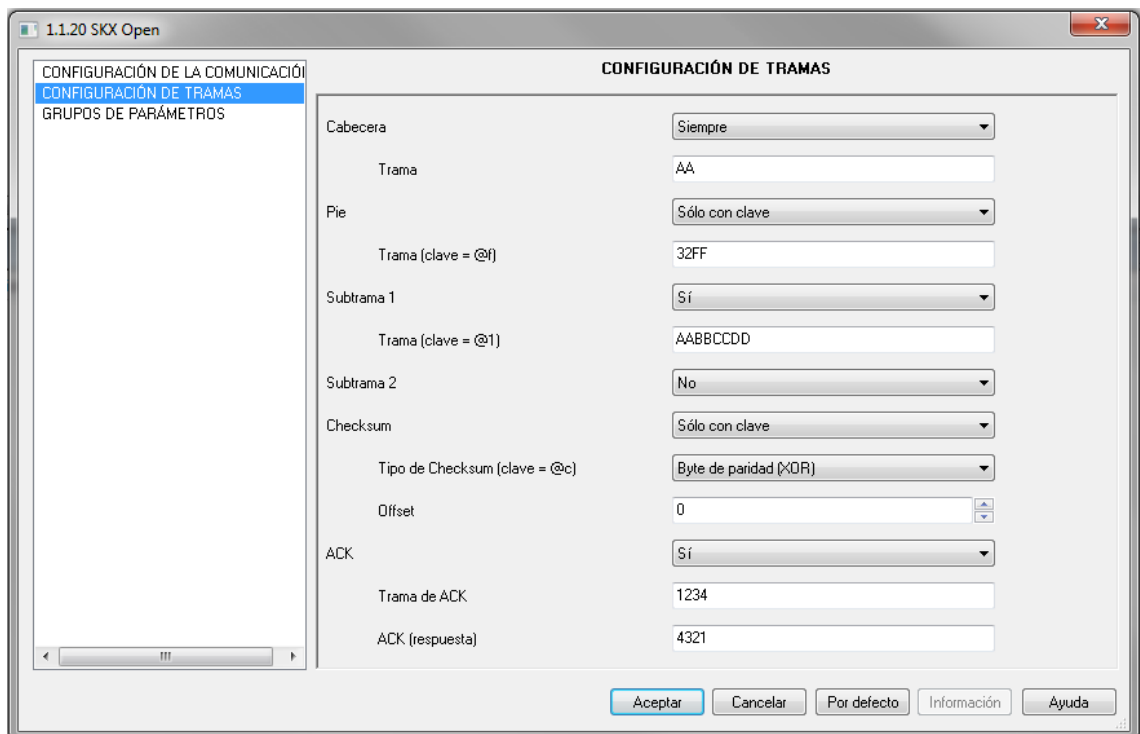



Figura 2.5. SKX Advance: Configuración de tramas (tramas especiales)

A continuación se detallan cada una de estas tramas especiales:

- **Cabecera:** esta trama podrá aparecer al principio de cada una de las tramas (recibidas y enviadas) de manera automática (“Siempre”) o sólo cuando se indique en la propia trama (“Sólo con clave”) mediante el carácter especial ‘@h’. Si se elige la opción “No”, la cabecera no se incluirá en la trama.
- **Pie:** esta trama especial se incluye al final de las tramas definidas por parámetro. Al igual que con la cabecera, se podrá deshabilitar la inserción de esta trama (“No”), insertarla siempre al final de cada trama (“Siempre”) o sólo cuando se indique (“Sólo con clave”) mediante el carácter especial ‘@f’.
- **Subtramas 1 y 2:** podrán insertarse en cualquier punto de la trama definida por parámetro, mediante las claves ‘@1’ y ‘@2’ respectivamente.
- **Checksum:** (o suma de control). Esta trama especial permite introducir un checksum, calculado en cada caso (recepción o envío), desde el primer byte de la trama hasta el byte anterior al propio checksum. Podrá insertarse en la trama mediante la clave ‘@c’ o indicar que cada trama lleve siempre un checksum determinado al final de la misma (tras el pie, si lo hubiera). Además, existe un parámetro adicional (Offset) que indica al programa que el checksum deberá calcularse a partir del byte indicado por este valor.

Los tipos de checksum soportados son los siguientes:

- **Palabra de paridad:** operación XOR byte a byte.
- **Suma modular:** suma todos los bytes de la trama con módulo 256.
- **Suma modular con complemento a 2:** similar al anterior, pero en este caso, al resultado se le realiza el complemento a 2.
- **CRC-8 y CRC-16:** códigos de redundancia cíclicos de 8 o 16 bits respectivamente. En cada caso será necesario introducir un número (en decimal) que defina el polinomio característico del CRC.

 **Asentimientos (ACK):** SKX Advance permite enviar automáticamente tramas de asentimiento fijas ante cualquier trama recibida por el puerto serie. Esta trama de asentimiento se definirá en el campo “Trama de ACK”. Por otro lado, el programa de aplicación también permite configurar una trama específica (mediante el campo “Respuesta ACK”) como asentimiento emitido por el equipo terminal de manera que, en el caso de recibir la trama “Respuesta ACK”, SKX Advance no enviará ningún tipo de asentimiento. Se hace necesario por tanto definir en este punto las posibles tramas ACK que pueda enviar el equipo terminal.

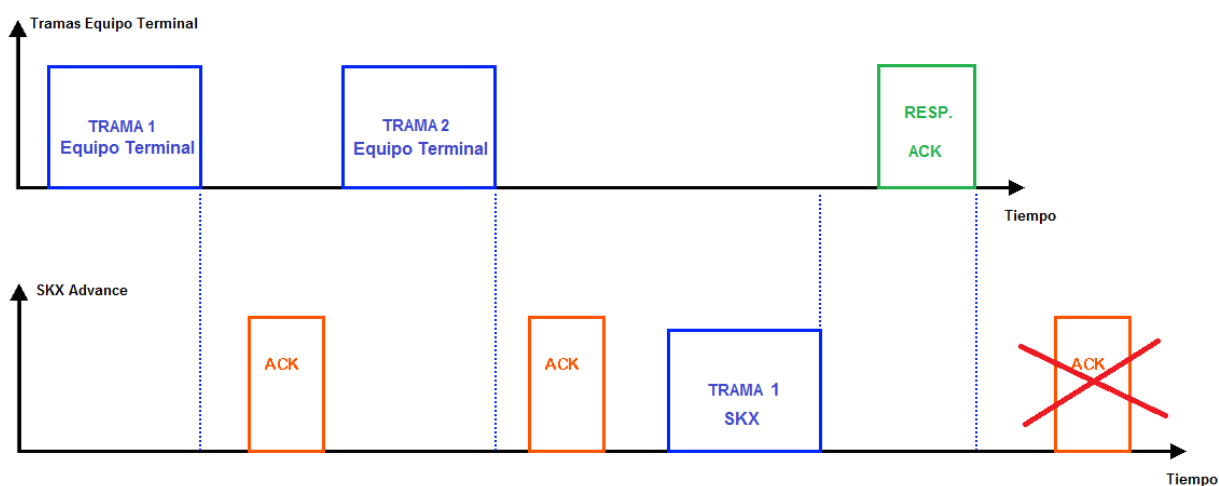



Figura 2.6. Tramas ACK y respuesta ACK

## 2.2.3. CARACTERES ESPECIALES

Como ya se ha comentado con anterioridad, existe un juego de caracteres especiales que podrán ser añadidos a las tramas definidas por parámetro para dotarlas de una mayor versatilidad, permitiendo la generación y detección de un conjunto más amplio de mensajes.

Además, el uso de este tipo de caracteres pretende permitir la utilización de expresiones regulares en la definición de las tramas.

Pueden distinguirse dos grupos: Caracteres especiales para recepción y Caracteres especiales para transmisión y recepción. A continuación se explican cada uno de ellos:

 **Caracteres especiales para recepción:** estos caracteres sólo pueden ser utilizados en la especificación de tramas que serán recibidas a través del puerto serie; su utilización en tramas a enviar generará un mensaje de error.

- **?1, ?2,... , ?9:** indican que los siguientes 1, 2,..., 9 bytes son ignorados a la hora de analizar la trama.

*Ejemplo:* se parametriza el envío de un “1” por el objeto de comunicación de 1 bit número 2 si SKX Advance recibe a través del puerto serie la siguiente trama definida por parámetro: “AA?223”. El equipo terminal empieza a enviar tramas, entre las que se encuentran las siguientes:

“AA857D23” → envía el objeto correspondiente a “1”, ya que los dos bytes 857D son ignorados, tal y como estaba especificado, y la parte fija de la trama coincide con la parametrizada.

“AA112223” → Igual que en el caso anterior. Esta vez ignora los bytes 1122.


“AA685AB923” → en este caso, SKX Advance no envía el valor del objeto, puesto que la parte fija recibida no coincide con la parametrizada: ignorará los 2 bytes siguientes a AA (685A), tal y como estaba definido, e interpretará B923 como parte fija de la trama que, al no coincidir con el 23 definido, hará que el objeto número 2 no envíe el valor 1.

- **\*\*:** aparición de **cero o más bytes** de cualquier valor hasta encontrar la parte fija de la trama definida por parámetro (excluyendo las claves @h, @f, @1, @2) Los caracteres \*\* representarán por tanto al conjunto mínimo de caracteres que se encuentren previos a la detección de la parte constante posterior a \*\* en la definición de la trama.

Ejemplo: se parametriza el envío del valor “50” a través del objeto de comunicación de 1 byte número 41 al recibir por el puerto serie la trama: “23\*\*AB”. El equipo terminal envía lo siguiente:

“231214AB” → SKX Advance comienza a analizar la trama: 23 y AB son las partes fijas de la misma, y todo lo que haya hasta que encuentre AB (parte constante posterior a \*\* en la definición de la trama) serán los caracteres definidos como \*\*. La trama coincide con la parametrizada, por lo que el objeto 41 envía el valor 50.

“23ABAB” → en este caso, SKX Advance interpreta que \*\* es el conjunto vacío, ya que no existen caracteres entre 23 y el primero de los AB de la trama. Al seguir analizando la trama, se encuentra con los caracteres AB, por lo que interpreta que la trama no coincide con la definida por parámetro y no envía el valor 50 por el objeto 41.

 **Caracteres especiales para transmisión y recepción:** estos caracteres pueden ser utilizados en la definición de tramas entrantes o salientes.

➤ **Claves de subtramas:** se definen de la forma “@ + carácter” y representan la inclusión de alguna de las tramas definidas en la pestaña “Configuración de tramas”, vista con anterioridad. Las diferentes posibilidades son:

- @h: inclusión de la trama de cabecera
- @f: inclusión de la trama de pie
- @1, @2: inclusión de las subtramas 1 y 2, respectivamente.
- @c: inclusión del byte, o bytes, de checksum

➤ **##:** este carácter indica la parte variable que irá asociada al objeto de comunicación.

**Objetos de 1 bit:** no puede ser utilizado.

**Objetos de 1 byte:** ## siempre ocupará un byte. Se distinguen cuatro casos en los que puede utilizarse:

- Envío de trama variable: SKX Advance insertará automáticamente el valor del objeto de comunicación correspondiente en la posición de la trama que ocupa el carácter ##.



*Ejemplo: se define por parámetro la siguiente trama variable: "1234###55". El objeto de comunicación de 1 byte número 44 recibe un valor igual a 16 (decimal) a través del bus KNX. Por tanto, SKX Advance enviará al equipo terminal la trama "12341055".*

*(Nota: SKX convierte 16 decimal en su equivalente hexadecimal (0x10) y lo inserta en la posición que ocupa ## en la definición de la trama en ETS para poder enviarla correctamente).*

- Obtener objeto variable: el objeto de comunicación adquirirá el valor recibido en la posición que ocupa el carácter especial dentro de la trama enviada desde el equipo terminal.

*Ejemplo: se define en este caso la siguiente trama asociada al objeto de 1 byte número 48: "AA##E8". Desde el equipo terminal llega lo siguiente: "AA02E8", por lo que el objeto de comunicación adquirirá el valor 2, en decimal, y lo enviará a través del bus KNX.*

- Envío de trama variable (porcentaje): similar al envío de trama variable, pero el valor del objeto de comunicación será convertido previamente desde porcentaje KNX (0-255) a porcentaje estándar (0-100).

- Obtener objeto variable (porcentaje): similar a obtener objeto variable pero el byte variable será convertido desde porcentaje estándar (0-100) a porcentaje KNX (0-255).

**Nota:** en el envío y recepción de tramas fijas el carácter ## no puede ser utilizado.

**Objetos de 14 bytes:** ## siempre representará una cadena de caracteres. En el caso de transmisión de la trama, este será el punto donde SKX Advance deberá copiar los caracteres recibidos a través del bus KNX; en el caso de recepción, este será el punto donde comience la trama de caracteres que SKX Advance tendrá que copiar en el objeto de comunicación correspondiente.

En los apartados 2.3.1, 2.3.2 y 2.3.3 pueden obtenerse ejemplos de uso de los caracteres especiales asociados a los diferentes objetos de SKX Advance.

## 2.3. GRUPOS DE PARÁMETROS

En esta pestaña podrán habilitarse los grupos de objetos de comunicación a utilizar.

SKX Advance dispone de un total de 65 objetos de comunicación, distribuidos de la siguiente forma:

Tamaño	Número de grupos	Número de objetos por grupo
1 bit	4	10
1 byte	2	10
14 bytes	1	5

Hay 40 objetos de comunicación de 1 bit, cuyos números de objeto están comprendidos entre 0 y 39; 20 objetos de 1 byte, entre 40 y 59, y 5 objetos de 14 bytes, con números comprendidos entre 60 y 64.

Al habilitar los grupos de parámetros, se muestran las pestañas correspondientes de cada uno de ellos:

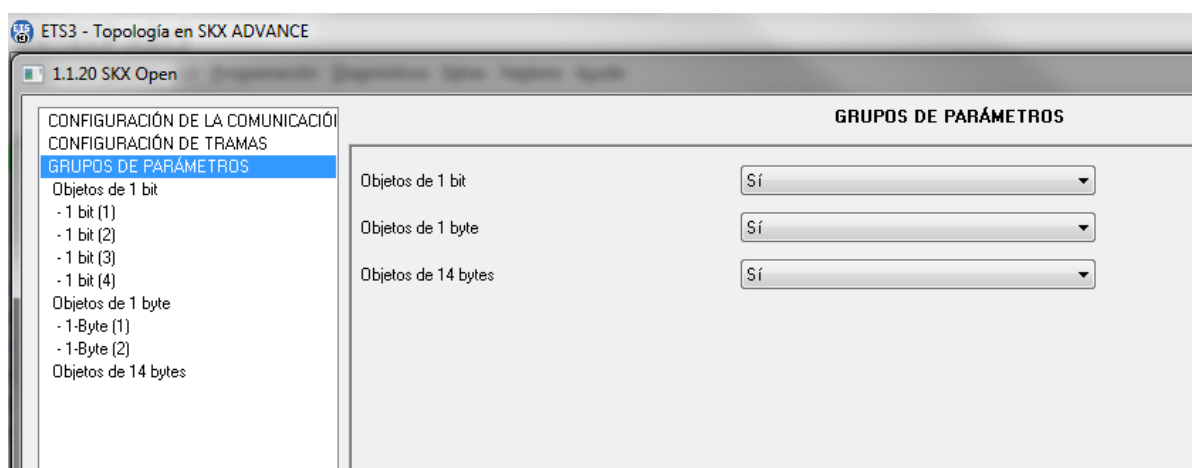


Figura 2.7. SKX Advance: Grupos de parámetros

Dentro de cada uno de los grupos de objetos de comunicación habrá que habilitar los objetos con los que se desea trabajar, accediendo así a las opciones disponibles en cada tipo de objeto de comunicación.

A continuación se explican cada uno de los tipos de objetos de comunicación disponibles:

### 2.3.1. OBJETOS DE 1 BIT

Los objetos de 1 bit permiten enviar desde SKX una trama de datos al equipo terminal, a través de RS-232, cuando SKX recibe por el bus KNX el valor previamente parametrizado en ETS (0 o 1) para el objeto configurado (objeto número 0 a objeto número 39). Además, este tipo de objetos también permiten al interfaz enviar un determinado valor (0 o 1) a través de un objeto de comunicación específico (0-39) cuando SKX recibe a través del bus RS-232 una trama fija procedente del equipo terminal.

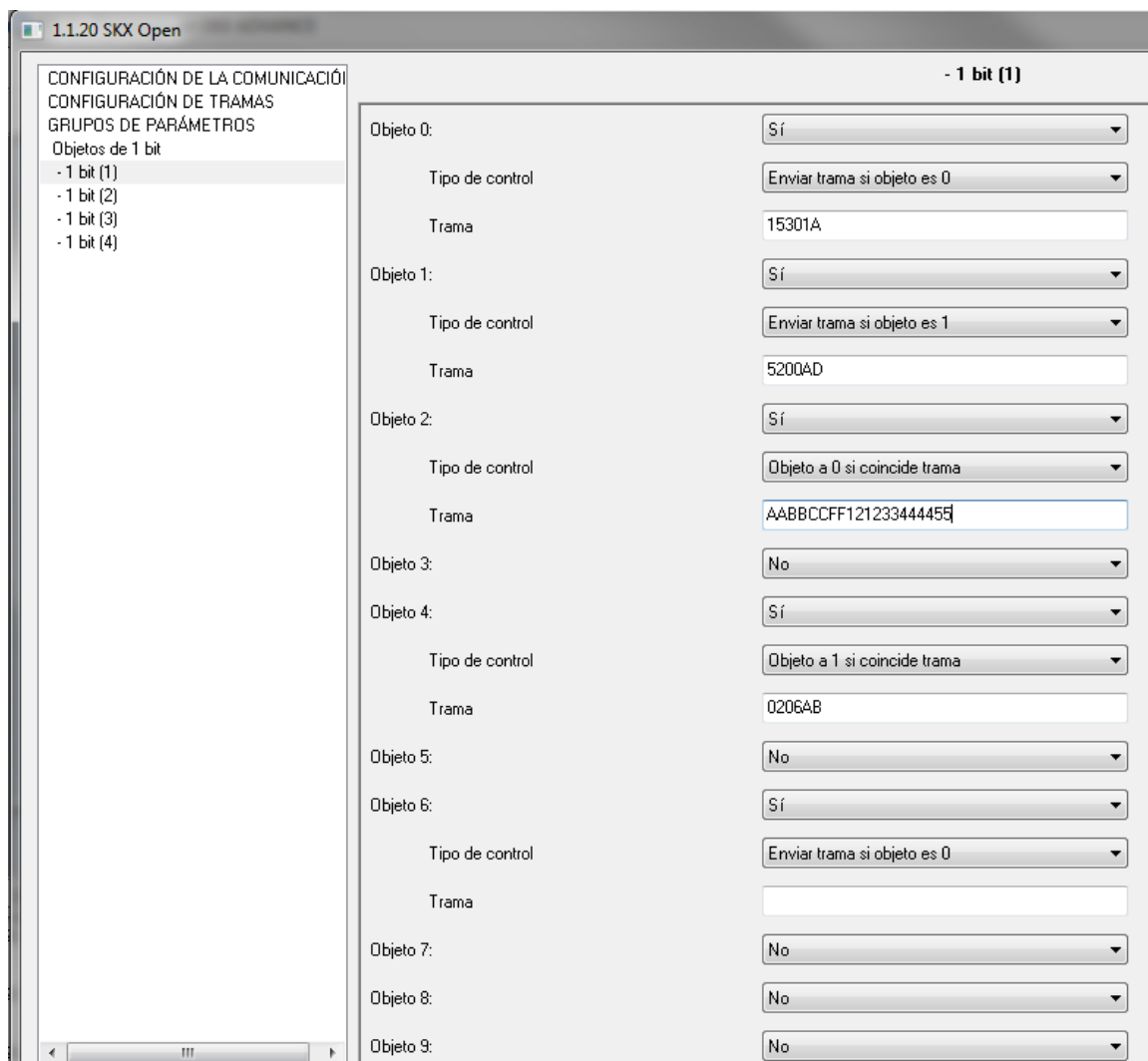


Figura 2.8. Grupo 1 de objetos de comunicación de 1 bit

Por ejemplo, con la parametrización anterior, SKX enviará, a través de RS-232, la trama “15301A” al equipo terminal si recibe por el bus KNX el objeto 0 con valor 0, mientras que si el objeto 1 llega con valor 1, SKX enviará la trama “5200AD” al equipo terminal.

Así mismo, si a través de RS-232 llega al interfaz la trama "AABCCFF121233444455" procedente del equipo terminal, SKX escribirá un 0 en el objeto de 1 bit número 2, mientras que si recibe la trama "0206AB" a través de RS-232, SKX escribirá un 1 en el objeto número 4.

Al habilitar un objeto de comunicación de 1 bit se muestran dos opciones: una para seleccionar el tipo de control y otra para definir la trama.

- **Objeto X. Tipo de control:** Existen cuatro posibilidades de control sobre cada objeto, a través de los siguientes parámetros:

Para la Comunicación KNX → RS232

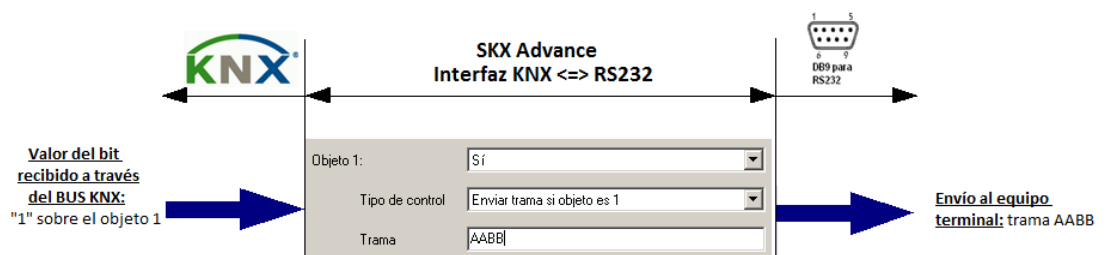
- **Enviar trama si objeto es 0:** Enviar la trama (introducida dentro del parámetro "Objeto X. Trama") hacia el equipo terminal **al recibir un "0"** por el bus KNX en el objeto de comunicación correspondiente.

*Ejemplo:*



- **Enviar trama si objeto es 1:** Enviar la trama (introducida dentro del parámetro "Objeto X. Trama") hacia el equipo terminal **al recibir un "1"** por el bus KNX en el objeto correspondiente.

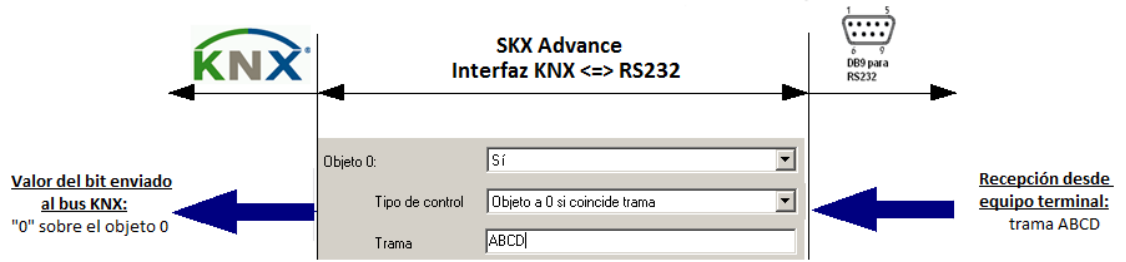
*Ejemplo:*



Para la Comunicación RS232 → KNX

- **Objeto a 0 si coincide trama:** Enviar un "0" por el objeto en caso de recibir por el puerto serie una trama que coincida con la introducida por parámetro.

*Ejemplo:*



- **Objeto a 1 si coincide trama:** Enviar un "1" por el objeto en caso de recibir por el puerto serie una trama que coincida con la introducida por parámetro.

*Ejemplo:*



- **Objeto X. Trama:** En este parámetro se definen las tramas necesarias para la comunicación. Éstas deberán cumplir los requisitos comentados en el apartado "2.2.1. Definición de tramas".

### 2.3.2. OBJETOS DE 1 BYTE

Los objetos de 1 byte permiten enviar desde SKX una trama de datos al equipo terminal, a través de RS-232, cuando SKX recibe por el bus KNX el valor parametrizado en ETS (0-255) para el objeto configurado (objeto número 40 a número 59). Además, este tipo de objetos también permiten al interfaz enviar un determinado valor (0-255) a través de un objeto de comunicación específico (40-59) cuando SKX recibe a través del bus RS-232 una trama específica procedente del equipo terminal. Las tramas podrán ser fijas o variables, según el valor del byte del objeto y lo mismo para el valor del objeto, que puede ser fijo o depender de la trama recibida.

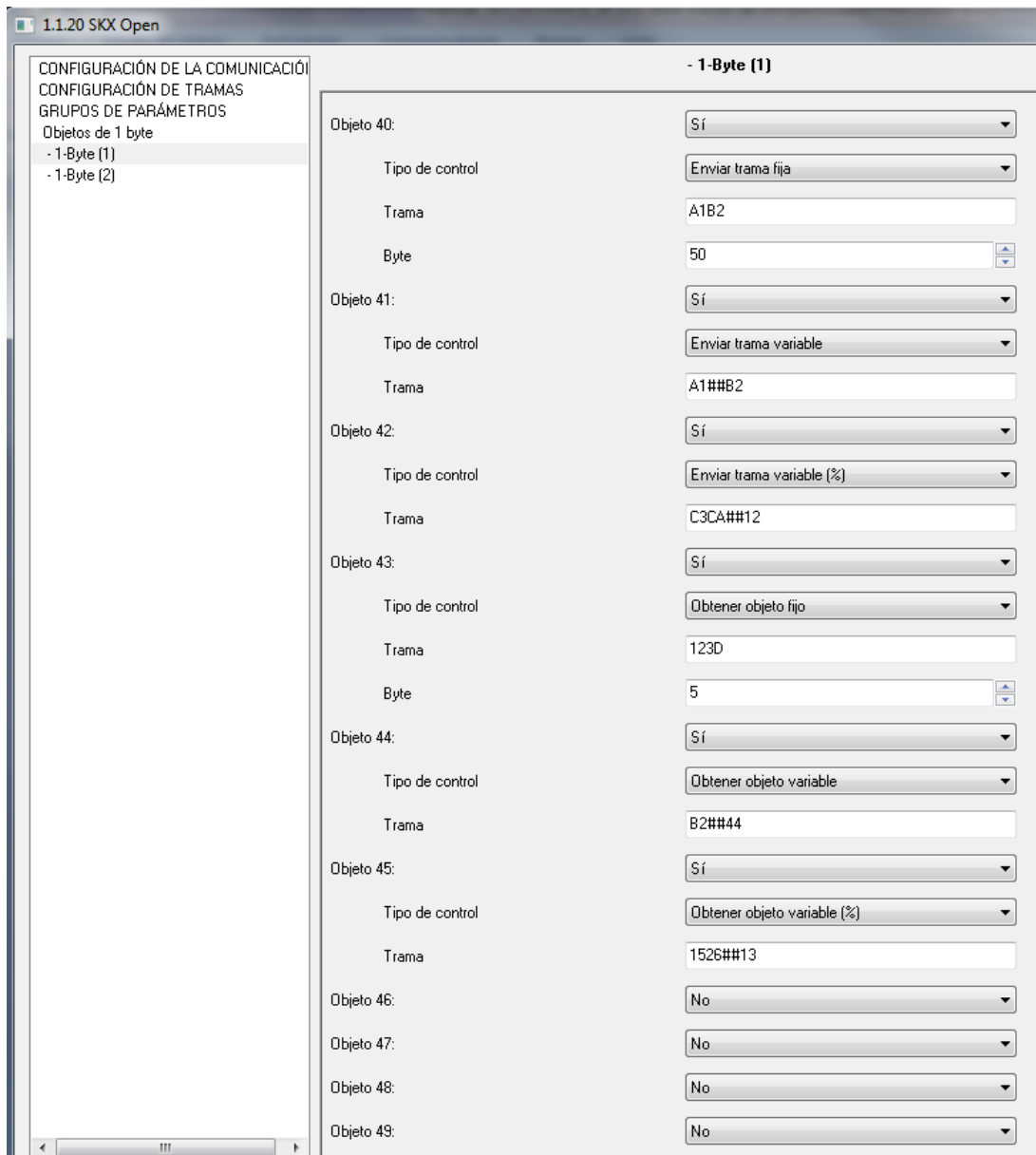


Figura 2.9. Grupo 1 de objetos de comunicación de 1 byte

Por ejemplo, con la configuración anterior, SKX enviará la trama “A1B2” al equipo terminal, a través de RS-232, cuando reciba por el bus KNX el objeto número 40 con valor 50 (decimal). Así mismo, cuando SKX reciba la trama “123D” a través de RS-232 procedente del equipo terminal, escribirá el valor 5 en el objeto de 1 byte número 43.

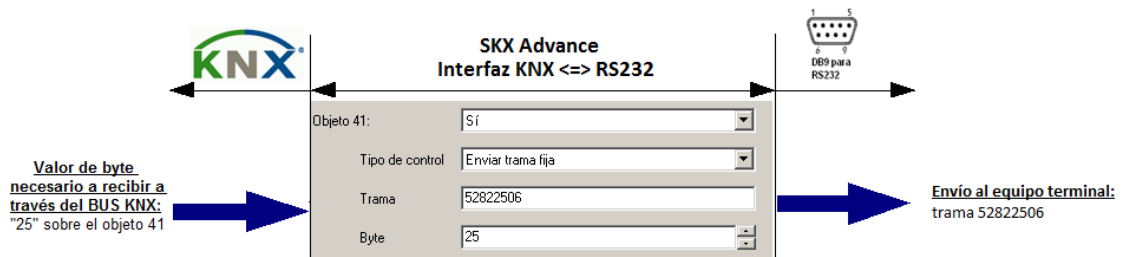
Al habilitar un objeto de comunicación de 1 byte se muestran dos opciones: una para seleccionar el tipo de control y otra para definir la trama.

- **Objeto X. Tipo de control:** existen seis posibilidades de control sobre cada objeto, a través de los siguientes parámetros:

## Para la Comunicación KNX → RS232

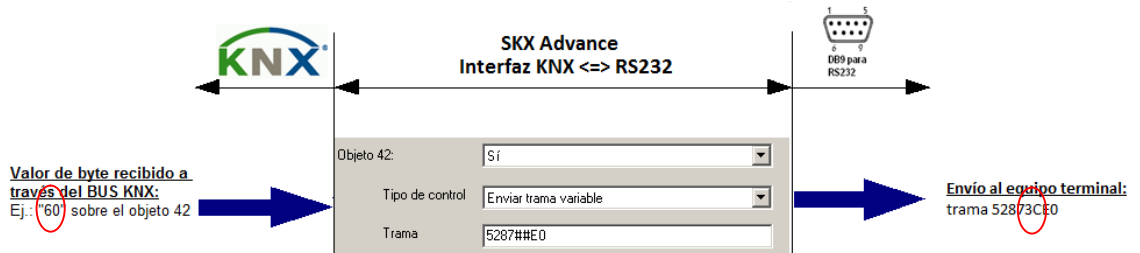
- **Enviar trama fija:** Enviar la trama (introducida dentro del parámetro “Objeto X. Trama”) hacia el equipo terminal al recibir el valor parametrizado por el objeto. Al habilitar este tipo de control, aparece una nueva opción: “**Objeto X. Byte**”, donde se define el byte que SKX Advance espera recibir por el objeto de comunicación para poder enviar la trama definida.

### Ejemplo:



- **Enviar trama variable:** Enviar la trama (introducida dentro del parámetro “Objeto X. Trama”) hacia el equipo terminal al recibir un valor por el objeto de comunicación (mediante la clave ##). La trama enviada variará en función del valor recibido por dicho objeto y será la misma que la definida por parámetro, sustituyendo ## por el valor del byte recibido por el objeto de comunicación correspondiente.

### Ejemplo:

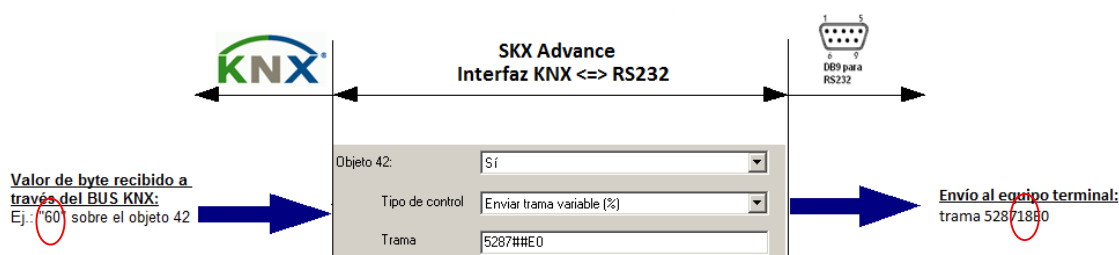


**Nota:** en este ejemplo, SKX Advance recibe un valor “60” a través del Bus KNX. Este valor está en **decimal**. Sin embargo, en la trama enviada al equipo terminal aparece el valor “3C”, que es 60 en **hexadecimal**. SKX Advance realiza esta conversión internamente, para que la trama se envíe correctamente.

- **Enviar trama variable (%):** Enviar la trama (introducida dentro del parámetro “Objeto X. Trama”) hacia el equipo terminal al recibir un valor por el objeto de comunicación correspondiente (mediante la clave ##). El valor recibido se convertirá a un número entre 0 y 100 (%) y será incorporado a la trama definida por parámetro. La trama enviada variará en función del valor recibido por dicho objeto (en

porcentaje) y será la misma que la definida por parámetro, sustituyendo ## por el valor del byte recibido por el objeto de comunicación correspondiente (en porcentaje).

Ejemplo:

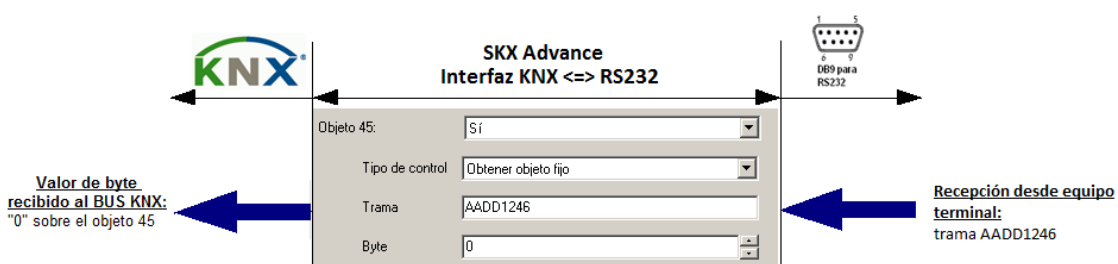


**Nota:** en este caso SKX Advance recibe un valor “60” en decimal. Este valor es un porcentaje KNX (0-255), que es convertido a porcentaje estándar (0-100%) mediante una simple regla de tres ( $60 \cdot 100 / 255$ ), siendo su valor equivalente 24%. Este valor se convierte a hexadecimal, dando como resultado “18”, valor que sustituirá a los caracteres ## en la trama definida por parámetro.

Para la Comunicación RS232 → KNX

- **Obtener objeto fijo:** Enviar el valor parametrizado por el objeto correspondiente en caso de recibir por el puerto serie una trama que coincida con la introducida por parámetro. Al habilitar este tipo de control, se muestra una nueva opción: “Objeto X. Byte”, donde se define el byte que SKX Advance escribirá en el objeto de comunicación al recibir la trama.

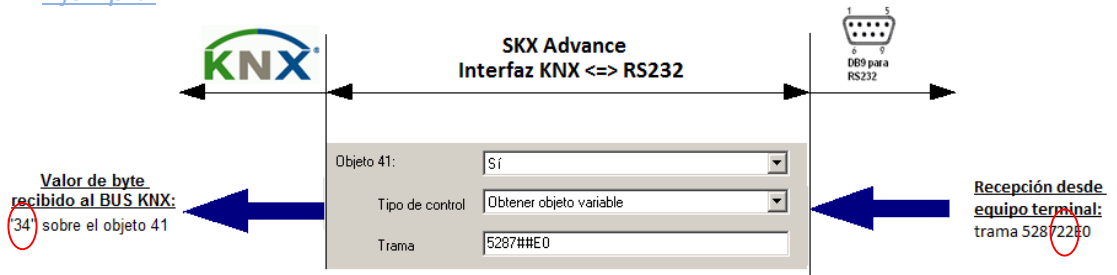
Ejemplo:



- **Obtener objeto variable:** Enviar un valor por el objeto de comunicación correspondiente en caso de recibir por el puerto serie una trama que coincida con la introducida por parámetro. El valor del objeto de comunicación será el correspondiente a la parte ## recibida en la trama.



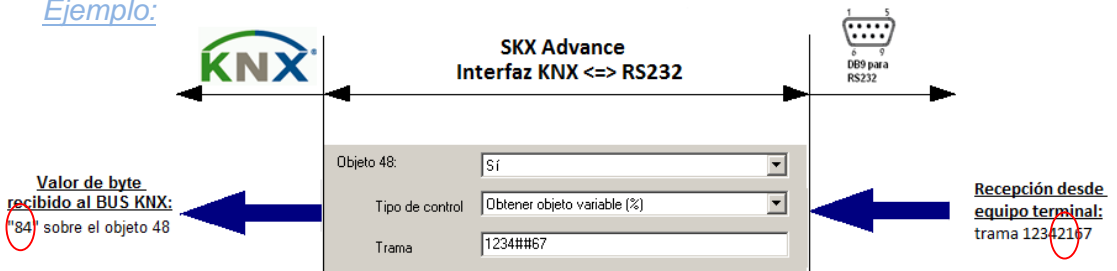
*Ejemplo:*



**Nota:** en este ejemplo, el equipo terminal envía la trama 528722E0 en hexadecimal. El valor que tomará el objeto de comunicación número 41 será el correspondiente a la parte ## recibida, que en este caso es "22" (en hexadecimal), pero convertido a decimal, de ahí que se escriba un "34" sobre el objeto 41.

- **Obtener objeto variable (%):** Enviar un valor por el objeto en caso de recibir por el puerto serie una trama que coincida con la introducida por parámetro. El valor del objeto de comunicación será el correspondiente a la parte ## recibida en la trama (en porcentaje).

*Ejemplo:*



**Nota:** en este caso SKX Advance recibe un valor "21" en hexadecimal en la parte correspondiente a ## de la trama definida. Este valor se corresponde con el valor decimal "33", interpretado esta vez como porcentaje estándar (0-100%), que será convertido a porcentaje KNX (0-255), mediante una sencilla regla de tres ( $33 \cdot 255 / 100$ ) siendo su valor equivalente "84", valor que se enviará al objeto de comunicación correspondiente.

- **Objeto X. Trama:** En este parámetro se definen las tramas necesarias para la comunicación. Éstas deberán cumplir los requisitos comentados en el apartado "2.2.1. Definición de tramas".

### 2.3.3. OBJETOS DE 14 BYTES

Los objetos de 14 bytes permiten el envío o detección de cadenas de texto dentro de tramas de puerto serie.

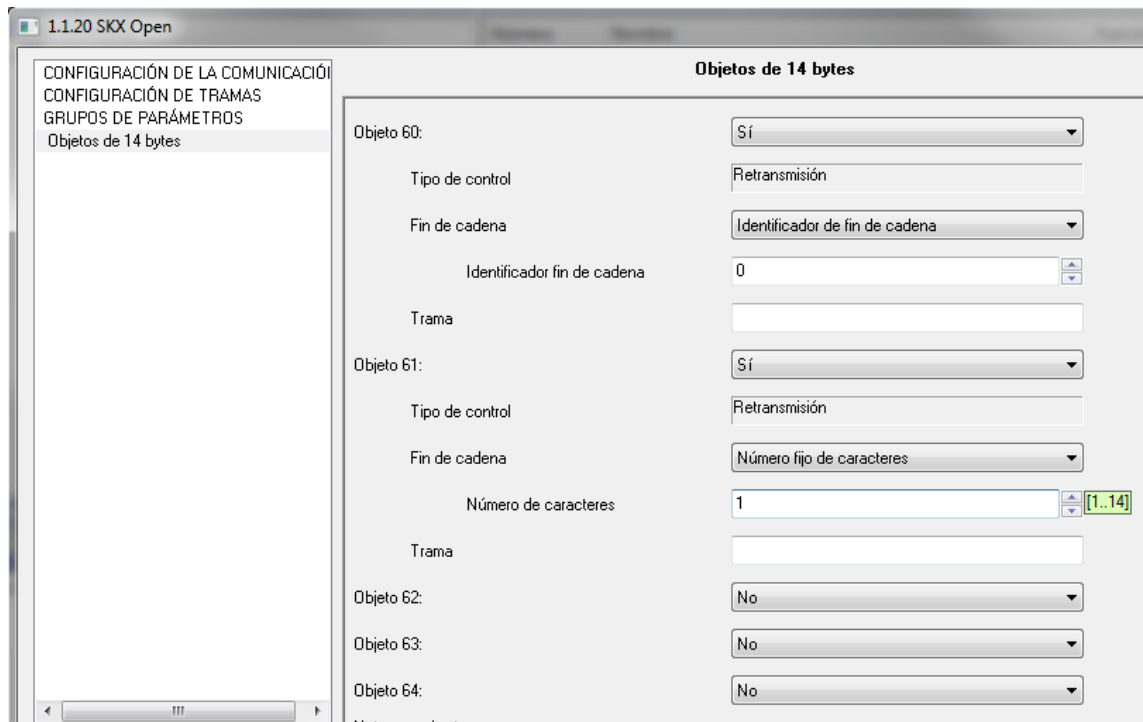
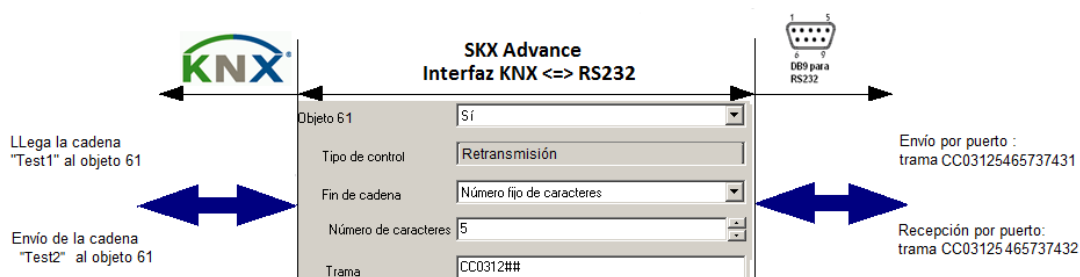


Figura 2.10. Grupo de objetos de comunicación de 14 bytes

Al habilitar un objeto de 14 bytes se muestran tres opciones que permiten seleccionar el tipo de control, el tipo de fin de cadena y definir la trama correspondiente.

- **Objeto X. Tipo de control:** existe una única forma de control sobre cada objeto, a través del parámetro **Retransmisión**, que permite volver a transmitir una cadena de texto variable recibida por el puerto serie a través del bus KNX y viceversa (mediante los caracteres ##).
- **Objeto X. Fin de cadena:** en este campo se selecciona la manera de confirmar el fin de cadena. Hay dos opciones disponibles:
  - **Número fijo de caracteres:** se define un valor fijo de caracteres (de 1 a 14) que debe tener la cadena de texto enviada o recibida.

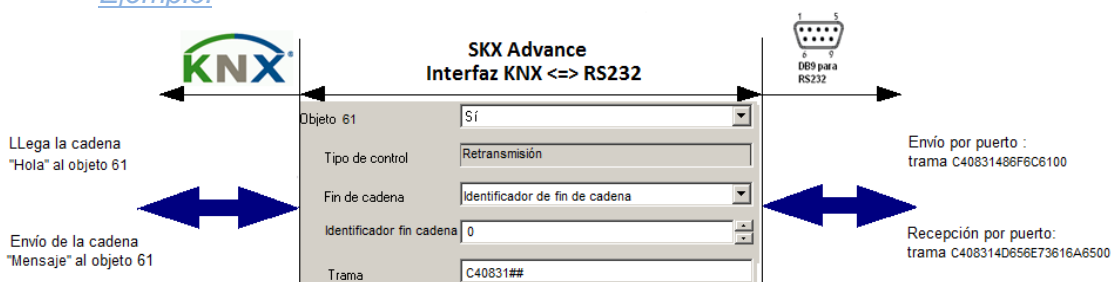
Ejemplo:



**Nota:** en este ejemplo, llega la cadena "Test1", de 5 caracteres, al objeto de comunicación número 61. La reacción asociada es un envío por el puerto serie de la siguiente trama: "CC03125465737431", donde CC0312 es la parte fija de la trama definida por parámetro y "5465737431" es la codificación ASCII, carácter a carácter, de la cadena "Test1". Después, SKX Advance recibe por el puerto serie la trama "CC03125465737432", lo que supone el envío de la cadena "Test2", de 5 caracteres, al objeto de comunicación 61.

- **Identificador fin de cadena:** se define el byte que indica el final de una trama. Este byte suele tener el valor 0x00.

Ejemplo:



**Nota:** se ha definido el identificador fin de cadena 0x00 (se escribe un 0 en la casilla correspondiente). Por el objeto número 61 llega la cadena de texto "Hola", que implicará un envío por el puerto serie de la trama "C40831486F6C6100", donde C40831 es la parte fija de la trama definida por parámetro, 486F6C61 es la codificación ASCII, carácter a carácter, de la cadena "Hola" y 00 es el identificador fin de cadena definido. Después, SKX Advance recibe por el puerto serie la trama "C408314D656E73616A6500" que genera el envío de la cadena de texto "Mensaje" al objeto de comunicación 61.

- **Objeto X. Trama:** En este parámetro se definen las tramas necesarias para la comunicación. Éstas deberán cumplir los requisitos comentados en el apartado "2.2.1. Definición de tramas".

## 2.4. OBJETOS DE ERROR

SKX Advance dispone de varios objetos de 1 bit que informan sobre errores en el funcionamiento del programa, principalmente debidos a una mala parametrización, aunque también pueden producirse durante la comunicación. Siempre que SKX Advance detecte un error, enviará el objeto de 1 bit correspondiente y, además, un objeto de error genérico de 1 bit (“Código de error”).

A continuación se describen los errores que pueden producirse:

- **Error: longitud impar.** El conjunto de caracteres de alguna de las tramas introducida por parámetro es impar. (**Nota:** recordar que esto no significa que el número de bytes de cada trama – longitud de trama – deba ser par, sino que como cada byte de la trama se parametriza con dos caracteres, el número total de caracteres debe ser siempre par).
- **Error: uso incorrecto de ‘\*’ o ‘?’.** No hay algo constante después de \*\* o el carácter posterior a ‘?’ es incorrecto.
- **Error: uso incorrecto de ‘@’.** Error en la utilización del carácter especial @.
- **Error: checksum incorrecto.** No hay nada a lo que calcularle el checksum, por ejemplo porque se ha configurado la trama simplemente con @c o porque el offset del checksum es demasiado grande.
- **Error: uso incorrecto de ‘#’.** Error en la utilización de ##. Posible error de sintaxis o que no es posible asociar ## con un dato variable.
- **Error: no hexadecimal.** En alguna trama definida por parámetro, en la que se esperaba un carácter de valor 0-9 o A-F, SKX Advance encontró otro distinto (por ejemplo, una letra en minúsculas).

**Nota:** todos los errores anteriores son errores de parametrización.

- **Error: demasiado largo.** La longitud de la trama a enviar o de la recibida es mayor que la máxima permitida: 29 bytes. (Este error puede ser de parametrización o de comunicación).
- **Error: recepción.** Error indeterminado de recepción de puerto serie. (Error de comunicación).

Cada vez que vuelve la tensión de bus, SKX realiza un análisis de todas las tramas introducidas por parámetro y en caso de detectar algún fallo en las tramas, envía los objetos de error correspondientes.

Únicamente se producirá **un envío por cada tipo de error**; es decir, que si en dos o más tramas se detecta el mismo tipo de error (por ejemplo, mala utilización del carácter especial “\*”), éste será notificado una única vez.

Si una trama contiene varios errores, todos ellos serán notificados, excepto en el caso en que la trama introducida sea impar; en dicha situación se notificará únicamente el error de trama impar. Por ejemplo, la trama “AaB” contiene un error de trama impar y otro de carácter no hexadecimal; SKX Advance únicamente detectará y notificará el error de trama impar, obviando el otro.

SKX Advance también notifica errores cuando intenta enviar por el puerto serie RS-232 una trama mal parametrizada. En este caso, **notificará únicamente el primer error detectado en la trama**. Por ejemplo, suponer que SKX Advance envía la trama “AAaB?B”; esta trama contiene dos errores (hay un carácter no hexadecimal y una mala utilización de ‘?’), pero cuando va a empezar a transmitirse, el envío se detendrá en el momento en que se detecte el error de carácter no hexadecimal, notificando únicamente este error.

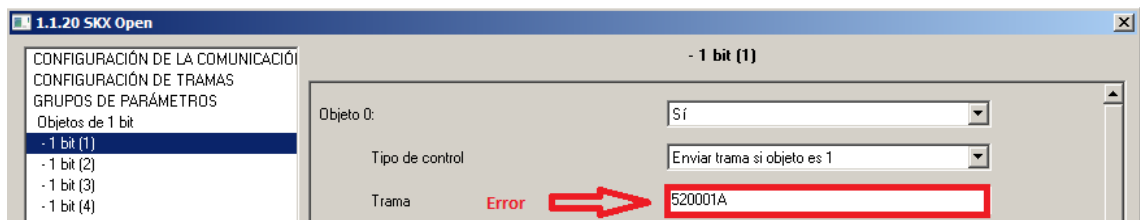
Por tanto, SKX Advance notifica los errores detectados en las tramas tras ser programado y cada vez que se transmitan o reciban tramas erróneas.

## 2.4.1. EJEMPLOS DE ERRORES

A continuación se presentan una serie de ejemplos con los errores que SKX Advance puede detectar.

- **Longitud impar.** En caso de intentar enviar una trama con un conjunto de caracteres definidos por parámetro impar, el objeto de comunicación de 1 bit “Error: Longitud impar” se activará, poniéndose a “1”. Adicionalmente, se activará y enviará al bus el objeto de 1 bit “Código de error”.

*Ejemplo:* se define dentro del objeto 0 la trama “520001A”, cuya longitud es impar (7 caracteres).



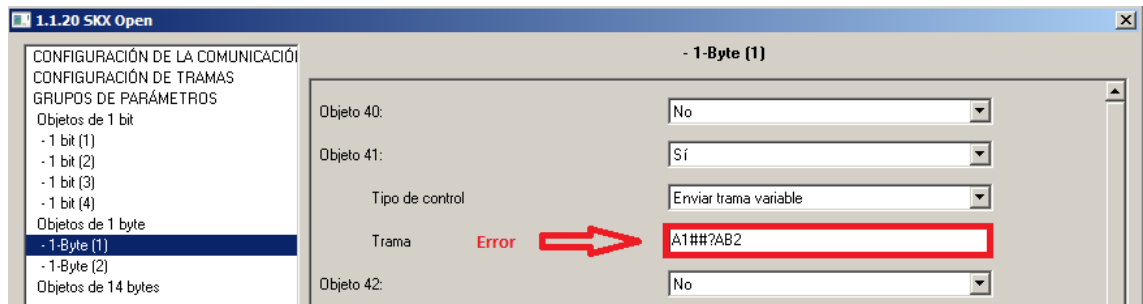
*Esto provocará el reporte de error siguiente:*

365	14:25:34.464	hacia e...	L	15.15.255	No Hallado	1/0/1	ON/OFF	6	1 bit	Write	\$01
366	14:25:34.484	desde ...	L	1.1.20		1/0/2	CODIGO ERROR	6	1 bit	Write	\$01
367	14:25:34.504	desde ...	L	1.1.20		1/0/3	IMPARE	6	1 bit	Write	\$01

Figura 2.11. Objetos de error por trama de longitud impar

**Uso incorrecto de los caracteres '\*' o '?'.** Si tras el carácter especial '?' se añade otro valor cualquiera distinto de los valores en el rango de 0 a 9, SKX Advance activará el objeto de comunicación de 1 bit "Error: Uso incorrecto de '\*' o '?'", así como el objeto "Código de error". También habilitará estos objetos si tras los caracteres '\*\*' no aparece algo constante.

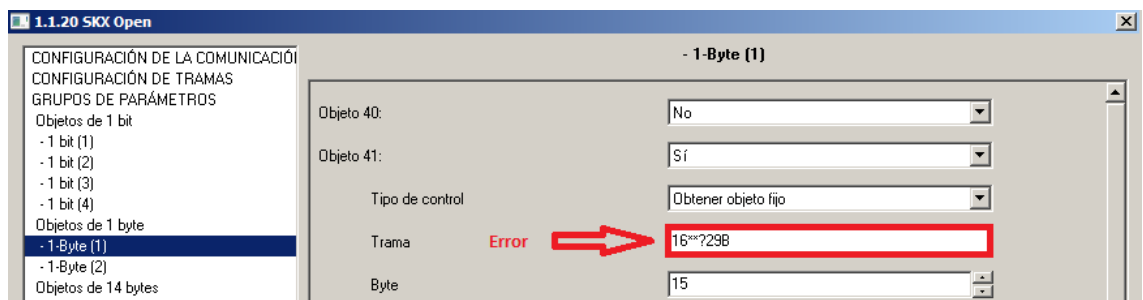
*Ejemplo I:* se define dentro del objeto de 1 byte número 41 el envío de la trama "A1##?AB2", donde aparece el carácter 'A' tras '?', valor inadecuado (ya que debería ser un número entre 0 y 9).



#	Hora	Servicio	Fl...	P	Dir.orig	Origen	Dir.dest	Destino	R	DPT	Tipo	Datos
1	14:33:34.194	hacia e...		L	15.15.255	No Hallado	1/0/41	Nueva Dirección de Grupo	6	1 byte	Write	\$03   1 %
2	14:33:34.214	desde ...		L	1.1.20		1/0/2	CODIGO ERROR	6	1 bit	Write	\$01
3	14:33:34.234	desde ...		L	1.1.20		1/0/4	*?	6	1 bit	Write	\$01


Figura 2.12. Objetos de error por uso incorrecto de carácter '?'

*Ejemplo II:* se define que en el objeto de 1 byte número 41 se escriba el valor 15 cuando SKX reciba desde el equipo terminal la trama "16\*\*?29B". SKX provocará un error de uso incorrecto de '\*\*', pues tras los caracteres '\*\*' no hay algo constante, sino que se ha definido "?2", que implica la aparición de 2 caracteres que pueden tener cualquier valor.

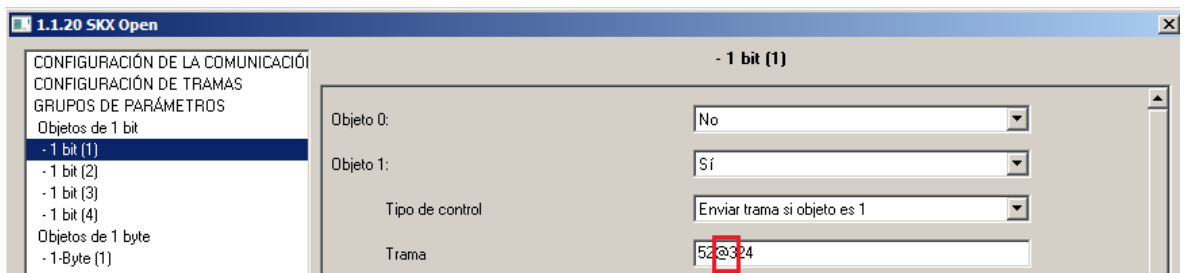


894	07:43:15.593	hacia el...		S	15.15.255	No Hallado	1.1.20		6	-	T_Disconnect	
895	07:43:15.613	desde ...		L	1.1.20		1/0/2	CODIGO ERROR	6	1 bit	Write	\$01
896	07:43:15.633	desde ...		L	1.1.20		1/0/4	*?	6	1 bit	Write	\$01

Figura 2.13. Objetos de error por uso incorrecto de '\*\*'


- 
**Uso incorrecto de '@'.** El carácter especial '@' permite introducir cabeceras, subtramas o pies dentro de una trama, y las claves para insertar cada una de ellas están ya predefinidas, no pudiendo haber otras diferentes a las siguientes: @h, @f, @1, @2 y @c. En caso de introducir un carácter erróneo tras @, SKX Advance activará el objeto de comunicación de 1 bit "Error: uso incorrecto de '@'", así como el objeto "Código de error".

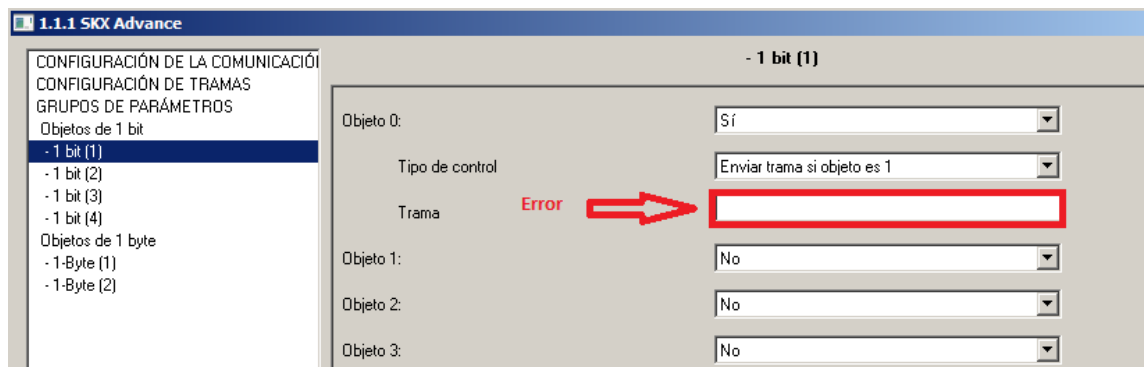
*Ejemplo:* se define la siguiente trama a enviar si el objeto de comunicación número 1 tiene el valor "1": "52 @324". El carácter 3 tras @ no es válido, por lo que SKX Advance activará el objeto de error correspondiente y la trama no se enviará.



1	14:36:54.645	hacia e...	L	15.15.255	No Hallado	1/0/1	ON/OFF	6	1 bit	Write	\$01
2	14:36:54.665	desde ...	L	1.1.20		1/0/2	CODIGO ERROR	6	1 bit	Write	\$01
3	14:36:54.685	desde ...	L	1.1.20		1/0/5	@	6	1 bit	Write	\$01


Figura 2.14. Objetos de error por uso incorrecto de '@'

- 
**Checksum incorrecto.** Este error informa de que no es posible realizar el checksum de una determinada trama porque no hay datos (la trama definida está vacía) o porque el offset configurado es demasiado grande. Si se da alguna de estas situaciones, SKX Advance activará el objeto de comunicación de 1 bit "Error: suma de control incorrecta" así como el objeto "Código de error".

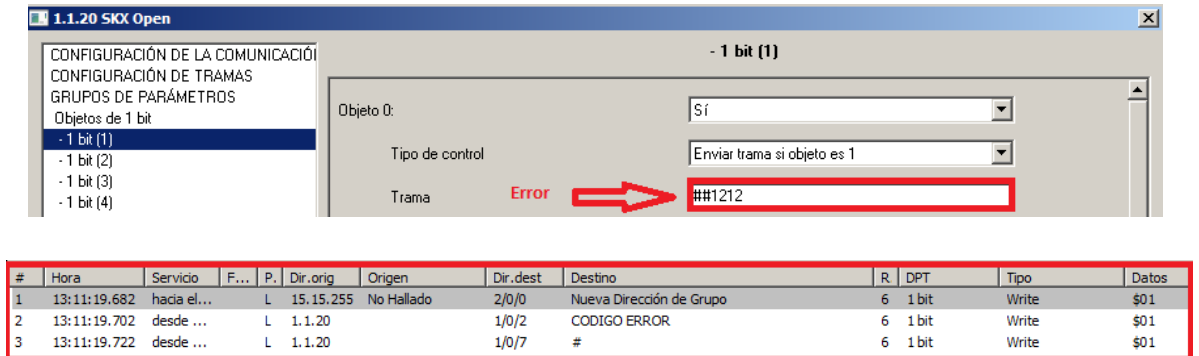


#	Hora	Servicio	Fl...	P	Dir.orig	Origen	Dir.dest	Destino	R	DPT	Tipo	Datos
1	14:27:11.128	hacia e...	L		15.15.255	No Hallado	2/0/0	Nueva Dirección de Grupo	6	1 bit	Write	\$01
2	14:27:11.149	desde ...	L		1.1.20		1/0/2	CODIGO ERROR	6	1 bit	Write	\$01
3	14:27:11.170	desde ...	L		1.1.20		1/0/6	CHEKSUM	6	1 bit	Write	\$01

Figura 2.15. Definición de trama vacía, que generará un error de checksum

- 
**Uso incorrecto de '#'**. Siempre que se utilice de forma incorrecta este carácter especial que permite añadir a la trama un valor variable, SKX Advance activará el objeto de 1 bit “Error: uso incorrecto de '#'” y el objeto “Código de error”.


*Ejemplo:* se utilizan los caracteres ## en la definición de una trama en el objeto de 1 bit número 0. SKX Advance activa el objeto de error asociado, ya que se está intentando añadir una parte variable cuando no está permitido para este tipo de objetos.



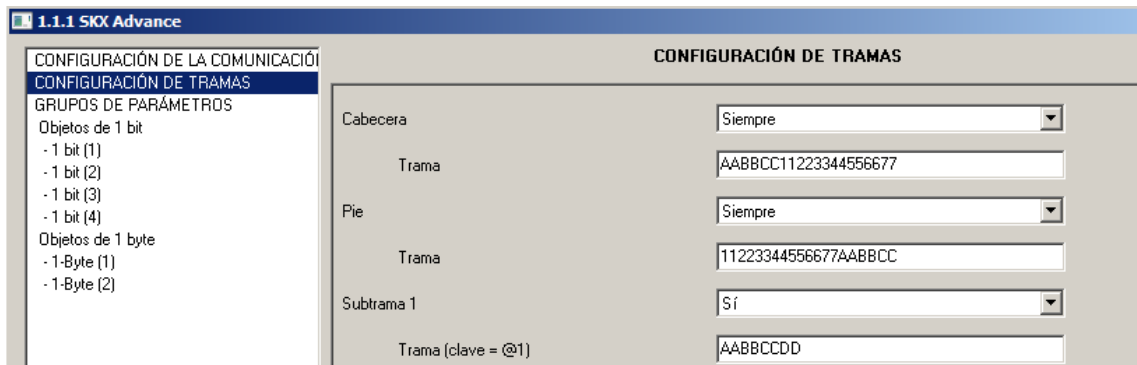
The screenshot shows the '1.1.20 SKX Open' configuration window. The left sidebar lists 'Objetos de 1 bit' with '- 1 bit (1)' selected. The main area shows configuration for '- 1 bit (1)'. The 'Objeto 0' dropdown is set to 'Sí'. The 'Tipo de control' dropdown is 'Enviar trama si objeto es 1'. The 'Trama' field contains '##1212', which is highlighted with a red box and a red arrow pointing to it from the word 'Error' written in red next to the field.

#	Hora	Servicio	F...	P.	Dir.orig	Origen	Dir.dest	Destino	R	DPT	Tipo	Datos
1	13:11:19.682	hacia el...	L	15.15.255	No Hallado	2/0/0	Nueva Dirección de Grupo	6	1 bit	Write	\$01	
2	13:11:19.702	desde ...	L	1.1.20		1/0/2	CODIGO ERROR	6	1 bit	Write	\$01	
3	13:11:19.722	desde ...	L	1.1.20		1/0/7	#	6	1 bit	Write	\$01	

Figura 2.16. Definición de trama con uso incorrecto de ##

- 
**Trama demasiado larga.** Debido a que las cabeceras, pies y subtramas permiten longitudes de n bytes, se puede llegar a obtener una definición de trama mayor de 29 bytes. Si esto sucede, SKX Advance activará el objeto de 1 bit “Error: demasiado largo” y el objeto “Código de error”.

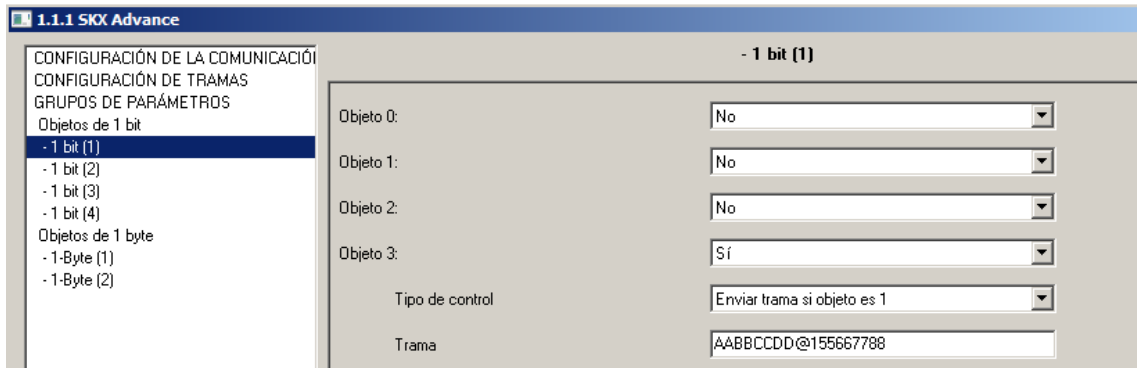
*Ejemplo:* si se definen una cabecera de 10 bytes, un pie de 10 bytes y una subtrama de 4 bytes, al añadir una trama de 10 bytes junto con la cabecera, pie y subtrama se está sobrepasando el límite permitido por lo que SKX Advance activará el objeto de error correspondiente.



The screenshot shows the '1.1.1 SKX Advance' configuration window. The left sidebar lists 'Objetos de 1 bit' with '- 1 bit (1)' selected. The main area shows configuration for 'CONFIGURACIÓN DE TRAMAS'. The 'Cabecera' dropdown is 'Siempre'. The 'Trama' field contains 'AABBCC11223344556677'. The 'Pie' dropdown is 'Siempre'. The 'Trama' field contains '11223344556677AABBCC'. The 'Subtrama 1' dropdown is 'Sí'. The 'Trama (clave = @1)' field contains 'AABBCCDD'.

Figura 2.17. Definición de las tramas de cabecera, pie y subtrama 1





#	Hora	Servicio	F...	P.	Dir.orig	Origen	Dir.dest	Destino	R	DPT	Tipo	Datos
1	13:17:14.946	hacia el...	L	15.15.255	No Hallado	2/0/3	Nueva Dirección de Grupo	6	1 bit	Write	\$01	
2	13:17:14.966	desde ...	L	1.1.20		1/0/2	CODIGO ERROR	6	1 bit	Write	\$01	
3	13:17:14.986	desde ...	L	1.1.20		1/0/8	DEMASIADO LARGO	6	1 bit	Write	\$01	

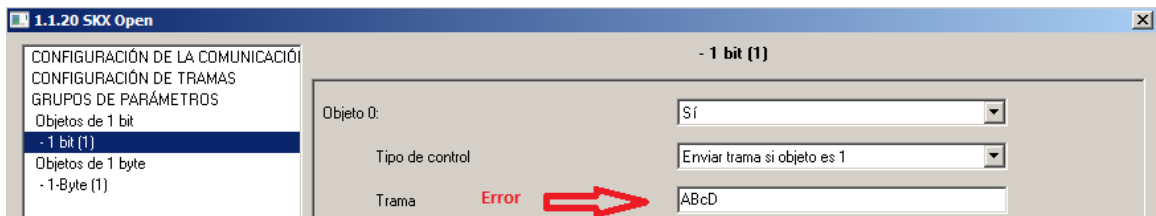
Figura 2.18. Definición de trama demasiado larga, que provocará el error correspondiente

- Error en la recepción.** Si la configuración de los parámetros de comunicación serie no coincide con la configuración de las tramas recibidas (diferente velocidad, paridad, etc.) SKX Advance habilitará el objeto de comunicación de 1 bit “Error: recepción” y el objeto “Código de error”.

*Ejemplo:* el equipo terminal tiene configurada una velocidad de 9600 baudios, mientras que la de SKX Advance es de 1200 baudios. Esto provocará que SKX Advance active el objeto de error de recepción.

- Carácter no hexadecimal.** Si se introdujo un carácter no hexadecimal a la hora de definir alguna trama de comunicación (como una letra minúscula), SKX Advance activará los objetos de comunicación “Error: no hexadecimal” y “Código de error”.

*Ejemplo:* uno de los caracteres utilizados para definir la trama del objeto de comunicación número 0 está escrito en minúsculas: “ABcD”. Esto hará que SKX Advance active el objeto de error correspondiente.



#	Hora	Servicio	F...	P.	Dir.orig	Origen	Dir.dest	Destino	R	DPT	Tipo	Datos
1	13:21:53.762	hacia el...	L	15.15.255	No Hallado	2/0/0	Nueva Dirección de Grupo	6	1 bit	Write	\$01	
2	13:21:53.782	desde ...	L	1.1.20		1/0/2	CODIGO ERROR	6	1 bit	Write	\$01	
3	13:21:53.804	desde ...	L	1.1.20		1/0/10	NO HEXADECIMAL	6	1 bit	Write	\$01	

Figura 2.19. Definición de trama con carácter no hexadecimal

## 2.5. EJEMPLOS DE CONFIGURACIÓN

A modo de resumen se muestran a continuación algunos ejemplos de configuración de tramas en SKX Advance y de su utilización, para que quede más claro su funcionamiento.

### 2.5.1. OBJETOS DE 1 BIT

Desde la pestaña “Configuración de tramas” de ETS, se establecen los siguientes valores para las tramas especiales:

The screenshot shows the 'CONFIGURACIÓN DE TRAMAS' window in the SKX Open software. The window title is '1.1.20 SKX Open'. The left sidebar contains three main sections: 'CONFIGURACIÓN DE LA COMUNICACIÓN', 'CONFIGURACIÓN DE TRAMAS' (which is selected), and 'GRUPOS DE PARÁMETROS'. The main area displays the following configuration parameters:

Cabecera	Sólo con clave
Trama (clave = @h)	AABB
Pie	No
Subtrama 1	Si
Trama (clave = @1)	CC028358
Subtrama 2	No
Checksum	Sólo con clave
Tipo de Checksum (clave = @c)	Suma modular
Offset	0
ACK	No

Tras esta primera parametrización, se pasan a definir los objetos de 1 bit que se desea utilizar; en este caso, se habilitan el objeto número 0 y el número 1 para enviar tramas al equipo terminal a través de RS-232 y el objeto número 2 para recibir tramas desde el equipo terminal. La parametrización de estos objetos queda así:

The screenshot shows the 'Objetos de 1 bit' configuration window in the SKX Open software. The window title is '1.1.20 SKX Open'. The left sidebar contains three main sections: 'CONFIGURACIÓN DE LA COMUNICACIÓN', 'CONFIGURACIÓN DE TRAMAS', and 'GRUPOS DE PARÁMETROS'. The 'Objetos de 1 bit' section is selected, and the main area displays the following configuration parameters:

- 1 bit (1)	
Objeto 0:	Si
Tipo de control	Enviar trama si objeto es 0
Trama	@h002349@1@c
Objeto 1:	Si
Tipo de control	Enviar trama si objeto es 1
Trama	@h0024@c
Objeto 2:	Si
Tipo de control	Objeto a 1 si coincide trama
Trama	@h?223@c

En la siguiente tabla se muestra una posible secuencia de funcionamiento, con las tramas que se envían y se reciben en cada caso. (**Nota:** los caracteres en *negrita indican el Checksum calculado como suma modular de los caracteres que conforman la trama*).

EVENTO	REACCIÓN
Llega desde el bus KNX un "1" al objeto número 0	No realiza acción, pues el objeto 0 sólo envía la trama parametrizada si llega un "0"
Llega desde el bus KNX un "0" al objeto número 0	SKX envía por el puerto serie RS-232 al equipo terminal la trama "AABB002349CC028358 <b>7A</b> "
Llega desde el bus KNX un "1" al objeto número 1	SKX envía por el puerto serie RS-232 al equipo terminal la trama "AABB0024 <b>89</b> "
SKX recibe desde el equipo terminal por el puerto serie la trama: "0024"	No realiza acción, pues la trama no coincide con la parametrizada
SKX recibe desde el equipo terminal por el puerto serie la trama: "AABB383223 <b>49</b> "	No realiza acción porque el checksum recibido no es correcto (debería ser <b>F2</b> )
SKX recibe desde el equipo terminal por el puerto serie la trama: "AABB383223 <b>F2</b> "	SKX escribirá un "1" sobre el objeto número 2
SKX recibe desde el equipo terminal por el puerto serie la trama: "AABB856D23 <b>7A</b> "	SKX escribirá un "1" sobre el objeto número 2

\* Checksum = **7A** = AA+BB+00+23+49+CC+02+83+58

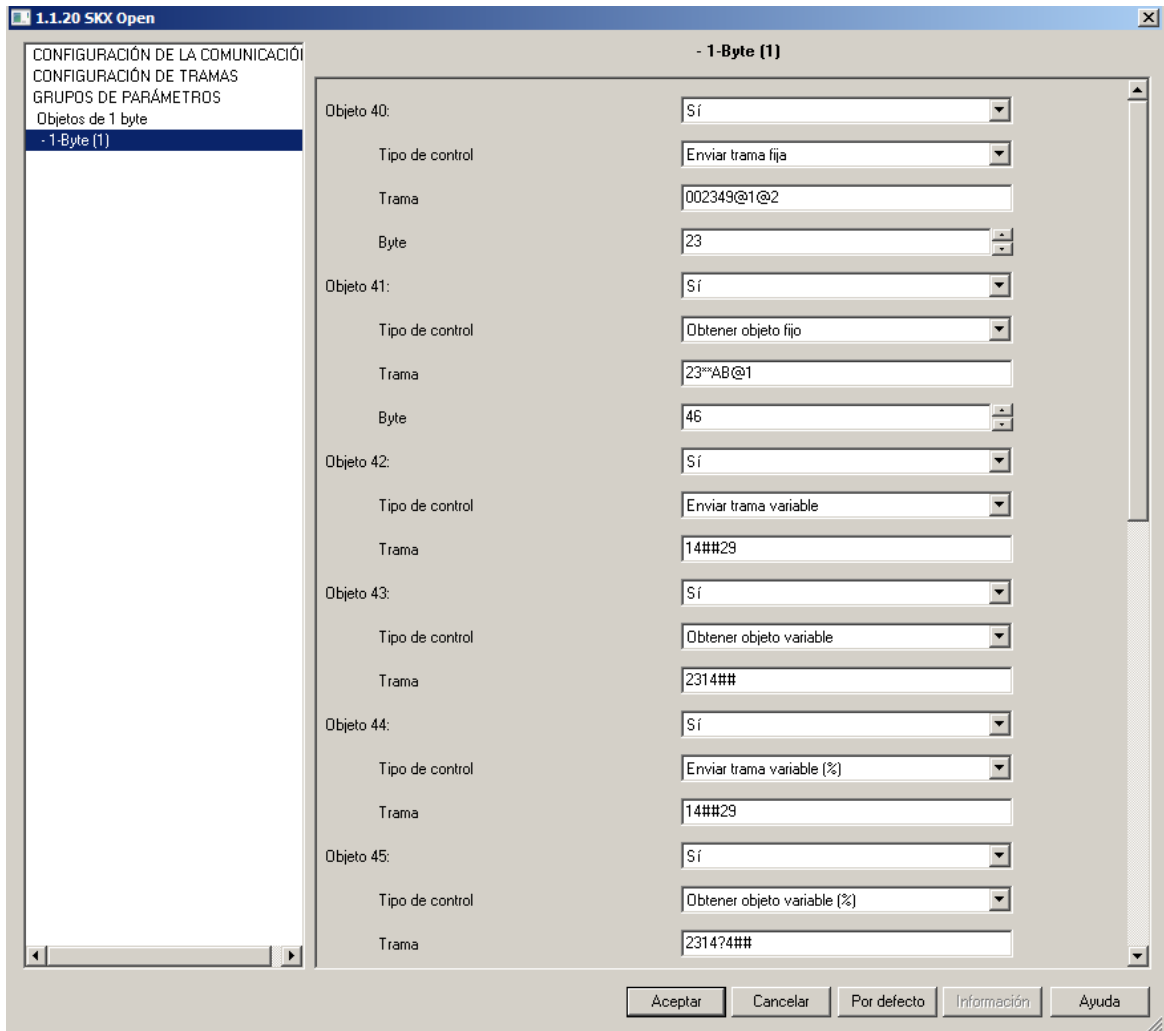
## 2.5.2. OBJETOS DE 1 BYTE

Desde la pestaña "Configuración de tramas" de ETS, se establecen los siguientes valores para las tramas especiales:

The screenshot shows the 'CONFIGURACIÓN DE TRAMAS' window with the following settings:

Parámetro	Valor
Cabecera	Siempre
Trama	AABB
Pie	No
Subtrama 1	Sí
Trama (clave = @1)	CC028358
Subtrama 2	Sí
Trama (clave = @2)	DD00
Checksum	No
ACK	Sí
Trama de ACK	0033
ACK (respuesta)	2348

Tras esta primera parametrización, se pasan a definir los objetos de 1 byte que se desea utilizar; en este caso, se habilitan los objetos número 40, 41, 42, 43, 44, 45 y 46, cada uno con un cometido diferente. La parametrización de estos objetos queda así:



En la siguiente tabla se muestra una posible secuencia de funcionamiento, con las tramas que se envían y se reciben en cada caso:

EVENTO	REACCIÓN
Llega desde el bus KNX un 12 al objeto número 40	No realiza acción, pues el objeto 40 sólo envía la trama parametrizada si llega un "23"
Llega desde el bus KNX un 23 al objeto número 40	SKX envía por el puerto serie RS-232 al equipo terminal la trama "AABB002349CC028358DD00"
SKX recibe desde el equipo terminal por el puerto serie la trama: "3232"	No se envía ningún valor a ningún objeto, pues la trama no coincide con ninguna de las parametrizadas. SKX envía asentimiento (ACK) = "0033"
SKX recibe desde el equipo terminal por el puerto serie la trama: "2348"	No se envía ningún valor a ningún objeto, pues la trama no coincide con ninguna de las parametrizadas para los objetos. En este caso tampoco se enviará la trama de ACK, pues "2348" coincide con la respuesta ACK parametrizada
SKX recibe desde el equipo terminal por el puerto serie la trama: "23ABCC028358"	No se envía ningún valor al objeto, porque aunque la trama recibida coincide con la definida para el objeto 41, el envío de la cabecera (AABB) es obligatorio y el equipo terminal no lo ha añadido. SKX envía el ACK = "0033"
SKX recibe desde el equipo terminal por el puerto serie la trama: "AABB23ABCC028358"	SKX escribirá un "46" sobre el objeto número 41, pues la trama recibida coincide con la parametrizada (siendo en este caso ** un conjunto vacío). Además, envía el ACK ("0033")
SKX recibe desde el equipo terminal por el puerto serie la trama: "AABB233333ABCC028358"	SKX escribirá un "46" sobre el objeto número 41, pues la trama recibida coincide con la parametrizada (siendo en este caso ** = '3333'). Además, envía el ACK ("0033")
SKX recibe desde el equipo terminal por el puerto serie la trama: "AABB23ABABCC028358"	SKX no envía ningún valor a ningún objeto, porque en este caso **='AB', pero al coincidir con la parte fija de la trama tras **, SKX Advance lo interpreta como un conjunto vacío y sigue analizando la trama, que no coincidirá con la parametrizada, por lo que no envía nada, excepto el ACK.
Llega desde el bus KNX un <u>16</u> (decimal) al objeto número 42	SKX enviará al equipo terminal a través de RS-232 la trama: "AABB14 <u>1029</u> " (ha transformado internamente el <u>16</u> recibido en su valor equivalente hexadecimal: 0x <u>10</u> )
SKX recibe desde el equipo terminal por el puerto serie la trama: "AABB2314"	SKX no envía ningún valor al objeto, porque aunque la trama coincide con la parametrizada para el objeto 43, no ha llegado ningún carácter en la parte ## que indique la parte variable a escribir sobre el objeto. SKX envía el ACK.
SKX recibe desde el equipo terminal por el puerto serie la trama: "AABB2314 <u>93</u> "	SKX escribe en el objeto número 43 el valor decimal equivalente al hexadecimal recibido en la parte correspondiente a ## de la trama (0x <u>93</u> → <u>147</u> ). SKX escribe 147 en el objeto 43 y además envía el ACK.

EVENTO	REACCIÓN
SKX recibe desde el equipo terminal por el puerto serie la trama: "AABB2314 <u>28</u> "	SKX escribe en el objeto número 43 el valor decimal equivalente al hexadecimal recibido en la parte correspondiente a ## de la trama (0x <u>28</u> → <u>40</u> ). SKX escribe 40 en el objeto 43 y además envía el ACK
Llega desde el bus KNX un <u>83</u> al objeto número 44	SKX envía por el puerto serie RS-232 al equipo terminal la trama "AABB14 <u>21</u> 29" (el valor <u>83</u> que llega a SKX está en decimal y hace referencia a un porcentaje KNX. SKX primero lo transforma a porcentaje estándar → $83 \cdot 100 / 255 \approx 33\%$ ; y luego a hexadecimal: 33d → 0x <u>21</u> )
Llega desde el bus KNX un <u>255</u> al objeto número 44	SKX envía por el puerto serie RS-232 al equipo terminal la trama "AABB14 <u>64</u> 29" (el valor <u>255</u> que llega a SKX está en decimal y hace referencia a un porcentaje KNX. SKX primero lo transforma a porcentaje estándar → $255 \cdot 100 / 255 = 100\%$ ; y luego a hexadecimal: 100d → 0x <u>64</u> )
SKX recibe desde el equipo terminal por el puerto serie la trama: "AABB2314AABBCCDD <u>83</u> "	SKX escribe en el objeto número 45 el valor decimal equivalente a lo que llega en la posición correspondiente a ## → <u>83</u> . Este es un porcentaje estándar. SKX lo transforma a porcentaje KNX ( $83 \cdot 255 / 100 = 212$ ) y ese será el valor que escriba en el objeto 45, además de enviar el ACK.

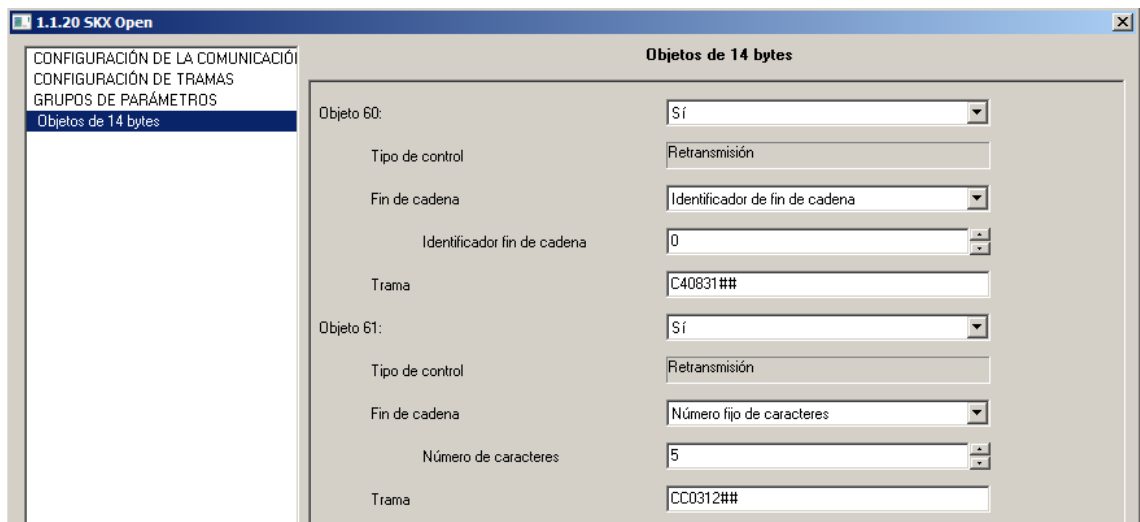
### 2.5.3. OBJETOS DE 14 BYTES

Desde la pestaña "Configuración de tramas" de ETS, se establecen los siguientes valores para las tramas especiales:

The screenshot shows a software window titled "1.1.20 SKX Open" with a sub-tab "CONFIGURACIÓN DE TRAMAS". On the left, there is a navigation pane with "CONFIGURACIÓN DE TRAMAS" selected. The main area contains the following configuration options:

- Cabecera: No
- Pie: Siempre
- Trama: 00DD
- Subtrama 1: No
- Subtrama 2: No
- Checksum: No
- ACK: No

Tras esta primera parametrización, se pasan a definir los objetos de 14 bytes que se desea utilizar; en este caso, se habilitan los objetos número 60 y 61. La parametrización de estos objetos queda así:



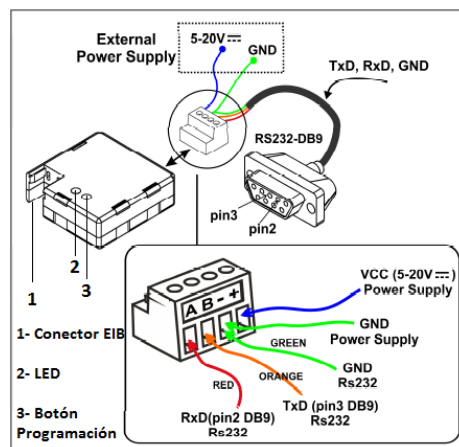
En la siguiente tabla se muestra una posible secuencia de funcionamiento, con las tramas que se envían y se reciben en cada caso:

EVENTO	REACCIÓN
Llega desde el bus KNX la cadena de caracteres "Hola" al objeto número 60	SKX envía por el puerto serie RS-232 al equipo terminal la trama "C40831 <b>486F6C61</b> 0000DD", donde: Hola = " <b>486F6C61</b> " (codificación ASCII de cada carácter) 00 = Identificador fin de trama 00DD = Pie
SKX recibe desde el equipo terminal por el puerto serie la trama: "C40831 <b>4D656E73616A6500</b> "	SKX no realiza ninguna acción, pues la trama recibida no incluye el pie.
SKX recibe desde el equipo terminal por el puerto serie la trama: "C40831 <b>4D656E73616A650000DD</b> "	SKX escribirá la cadena de caracteres "Mensaje" sobre el objeto número 60.
Llega desde el bus KNX la cadena de caracteres "Test" al objeto número 61	SKX no realiza ninguna acción, pues la cadena no tiene la longitud fija parametrizada, 5 caracteres
Llega desde el bus KNX la cadena de caracteres "Test1" al objeto número 61	SKX envía por el puerto serie RS-232 al equipo terminal la trama "CC0312 <b>5465737431</b> 00DD", donde: Test1 = " <b>5465737431</b> " (codificación ASCII de cada carácter) 00DD = Pie
SKX recibe desde el equipo terminal por el puerto serie la trama: "CC0312 <b>50727565626100DD</b> "	SKX escribirá la cadena de caracteres "Prueba" sobre el objeto número 61
SKX recibe desde el equipo terminal por el puerto serie la trama: "CC0312 <b>5465737400DD</b> "	SKX no realiza acción ninguna, porque al ser la trama de longitud fija (5 caracteres), SKX la analiza e interpreta como cadena entrante "5465737400"; lo siguiente que busca es el pie, pero no lo encuentra (ya que sólo queda DD).

### 3. RESUMEN DEL PRODUCTO



Hardware del dispositivo



Conexión SKX – RS232

**SKX:** interfaz ZENNIO.

- Comunicación **bidireccional** entre el bus KNX y dispositivos externos con puerto serie RS-232 (como televisores, sistemas de vigilancia, sistemas de audio, etc.)
- Posibilidad de ampliar una instalación con dispositivos que no dispongan comunicación KNX pero sí RS-232.

**SKX Advance:** programa de aplicación.

- Tramas dinámicas de longitud variable (hasta **29 Bytes**)
- Objetos de comunicación de varios tipos
- Detección de errores de comunicación
- Varias velocidades de transmisión
- **Se adapta** a cualquier protocolo que rija el funcionamiento del equipo terminal
- Gran **versatilidad**.





¡HAZTE USUARIO!

<http://zennio.zendesk.com>

SOPORTE TÉCNICO