



SKX OPEN

SKX ADVANCE

ZN1RX-SKXOPEN



TABLE DES MATIERES

1.	INTRODUCTION	3
1.1.	INTERFACE SKX	3
1.2.	INSTALLATION SKX	5
1.3.	PROGRAMME D'APPLICATION: SKX ADVANCE	5
1.3.1.	SPECIFICATIONS BASIQUES DU SKX ADVANCE.....	6
1.4.	DIFFERENCES ENTRE LES APPLICATIONS SKX OPEN ET SKX ADVANCE	6
1.5.	AVANTAGES DU PRODUIT ET DE L'APPLICATION	7
2.	PARAMETRAGE ETS	8
2.1.	CONFIGURATION DE LA COMMUNICATION.	8
2.2.	CONFIGURATION DES TRAMES	12
2.2.1.	DEFINITION DES TRAMES	12
2.2.2.	TRAMES SPECIALES.....	12
2.2.3.	CARACTERES SPECIAUX	15
2.3.	GROUPES DE PARAMETRES.....	18
2.3.1.	OBJETS DE 1 BIT	19
2.3.2.	OBJETS DE 1 BYTE.....	21
2.3.3.	OBJETS DE 14 BYTES	26
2.4.	OBJET D'ERREUR.....	28
2.4.1.	EXEMPLES D'ERREURS.....	29
2.5.	EXEMPLES DE CONFIGURATION.....	35
2.5.1.	OBJETS DE 1 BIT	35
2.5.2.	OBJETS DE 1 BYTE	36
2.5.3.	OBJETS DE 14 BYTES	39
3.	RESUME DU PRODUIT	41

1. INTRODUCTION

1.1. INTERFACE SKX

SKX et une interface **ZENNIO**, qui permet la connexion du BUS KNX avec d'autres dispositifs via une communication série de type **RS-232** bidirectionnelle.



Figure 1.1. Schéma connexion SKX

Nomenclature

Voici une définition des termes utilisés fréquemment dans ce manuel:

- **SKX:** A partir de maintenant, afin d'éviter les confusions entre le hardware et le programme d'application du même nom, l'interface SKX-Open sera appelé SKX.
- **SKX Advance:** Programme d'application à télécharger dans le SKX et qui permet de gérer la communication KNX - RS232.
- **RS-232:** Type de communication série.
- **Dispositif final:** Dispositif externe à intégrer via le BUS série RS-232.

Voici une présentation des caractéristiques principales de l'interface, ainsi qu'un schéma de ses éléments (figure 1.2):

- Petite taille: 45 x 45 x 14 mm
- Plusieurs vitesses de communication et mécanismes de détection d'erreurs.
- Idéal pour les applications M2M.
- Sauvegarde des données en cas de perte d'alimentation.

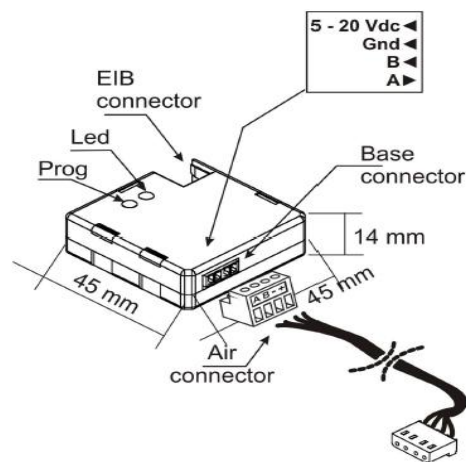


Figure 1.2. SKX: Schéma des éléments

Le bouton "Prog" est utilisé pour placer le SKX en mode programmation. Si ce bouton est maintenu appuyé au moment d'appliquer la tension de BUS, le SKX, se met en mode sûr.

SKX possède une LED bicouleur sur la face avant du dispositif. Ce signal lumineux a une double fonction. D'un côté, la LED fixe indique que l'interface se trouve en mode de programmation et la LED clignotante, chaque 0.5s, indique que l'interface se trouve en mode sûr. D'un autre côté, la LED peut également servir d'indicateur de transmission. Si la LED est verte, cela indique que des données ont été envoyées ou reçues du port série, restant allumée 0.3 secondes à chaque fois qu'il y a communication.

Cette interface peut être programmée avec deux programmes d'application différents: **SKX Open** et **SKX Advance**, dont les différences sont expliquées dans le chapitre 1.4.

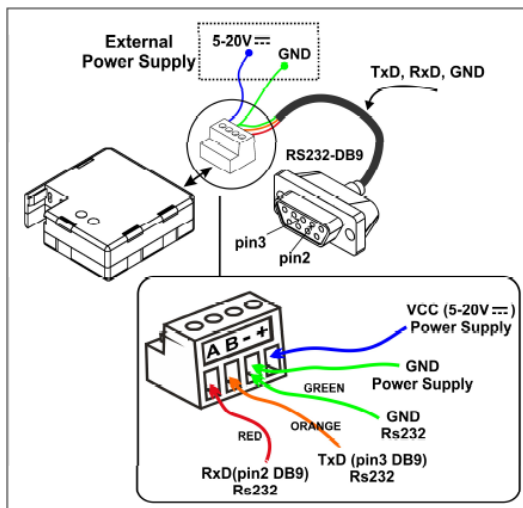
1.2. INSTALLATION SKX

L'installation du SKX est très facile: connecter simplement l'interface au BUS KNX via son connecteur spécifique, il est alors prêt à être programmé.

SKX ne nécessite pas d'alimentation externe; il fonctionne directement avec l'alimentation du BUS KNX. En revanche, il faudra alimenter le bus RS-232 avec une alimentation indépendante à celle du BUS KNX (En règle général, l'alimentation est délivrée par le dispositif final connecté sur le BUS série).

La connexion du SKX avec le RS-232 se réalise via un connecteur spécifique, ce qui facilite sa manipulation et installation.

La figure suivante présente un schéma de connexion typique SKX – RS232.



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

Figure 1.3. Connexion SKX vers RS-232

1.3. PROGRAMME D'APPLICATION: SKX ADVANCE

SKX Advance est une application qui se charge dans l'interface SKX et dont la fonction principale est la gestion de la communication entre KNX et le protocole RS-232, permettant de configurer toute l'information échangée entre le BUS et le dispositif final. **Cet échange d'information est bidirectionnel, c'est-à-dire qu'il est possible d'envoyer des données du BUS KNX vers le dispositif final et viceversa.** La figure 1.4 présente un exemple ce type de communication.

SKX Advance peut envoyer ou recevoir n'importe quel type de trame de communications dans les limites définies dans le chapitre 2.2.1.

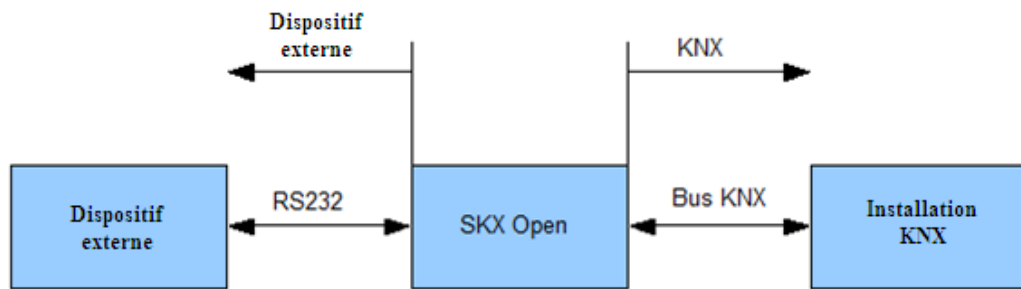


Figure 1.4. Communication bidirectionnelle

1.3.1. SPECIFICATIONS BASIQUES DU SKX ADVANCE

Voici une présentation des caractéristiques fondamentales de l'application SKX Advance:

- **Vitesse de transmission:** 1200, 1400, 4800, 9600 ou 19200 Bauds.
- **Parité:** Paire, impaire, sans parité
- **Type de distinction des trames:** Time Out, byte fin de trame
- **Nombre d'objets de communication:** 65 (40 de 1bit, 20 de 1byte et 5 de 14 bytes)
- **Identificateur d'erreurs:** Plusieurs objets de 1 bit
- **Longueur du protocole:** Les trames configurées par paramètre dans le SKX-Advance peuvent avoir une longueur supérieure à 25 bytes, grâce à la l'utilisation de caractères spéciaux dans sa configuration. Au total, il est possible d'**envoyer/recevoir des trames de jusqu'à 29 bytes**.

1.4. DIFFERENCES ENTRE LES APPLICATIONS SKX OPEN ET SKX ADVANCE

L'interface SKX peut être programmée avec deux programmes d'application différents: **SKX Open** et **SKX Advance**





Le tableau suivant présente les principales différences entre les deux programmes d'application:

	SKX OPEN	SKX ADVANCE
Nombre d'Objets	48	65
Type d'Objets	1 bit	1 bit 1 byte 14 bytes
Type de Trame	Fixes	Variables (en fonction de la valeur de l'objet)
Longueur des Trames	Jusqu'à 10 bytes	Jusqu'à 29 bytes
Reconnaissance des Trames	Le message entier doit coïncider	Uniquement reconnaissance des parties indiquées (omission du reste)
Checksum	Non	Oui
Confirmation (ACK)	Non	Oui
Configuration	- Communication - Groupes paramètres	- Communication - Des Trames - Groupes paramètres

Tableau 1.1. Différences entre SKX Open et SKX Advance

1.5. AVANTAGES DU PRODUIT ET DE L'APPLICATION

Le fait de charger le programme d'application SKX Advance dans l'interface SKX offre, entre autres, les avantages suivants:

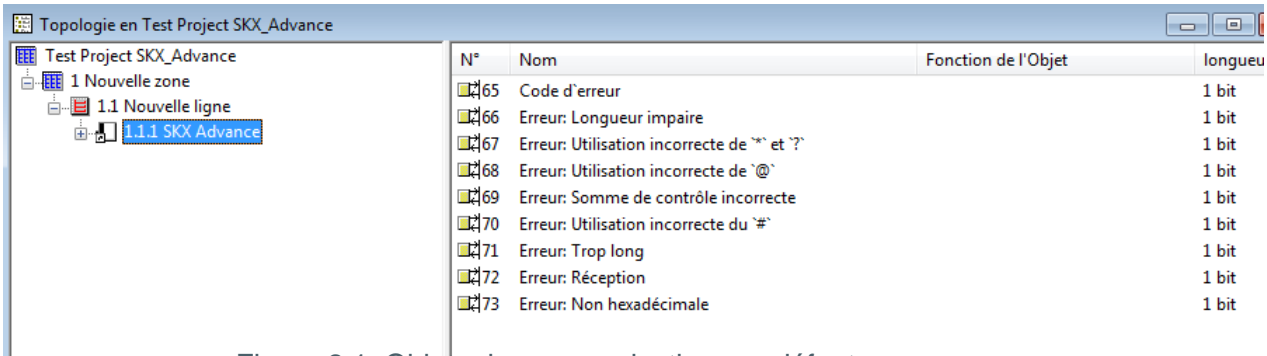
-  **Installation plus importante:** Permet d'intégrer dans une installation KNX d'autres dispositifs ne disposant pas de communication KNX mais d'un port série.
-  **Adaptabilité:** SKX Advance est capable de s'adapter parfaitement au protocole qui dirige le fonctionnement du dispositif final, indépendamment de la complexité de celui-ci. C'est-à-dire, SKX Advance s'**adapte au dispositif final** et non le contraire.
-  **Polyvalence de la communication:** SKX Advance possède des objets de différents types (plus concrètement, de 1 bit, 1 byte et 14 bytes) qui permettent l'échange d'information, avec le dispositif final, dans les deux directions (Communication bidirectionnelle).
-  **Configuration:** Les messages échangés, dans la communication RS-232, peuvent être configurés avec toutes les caractéristiques habituelles d'une communication série et avec la possibilité d'activer des en-têtes, pieds de trame, checksum, ACK, etc.; permettant ainsi l'envoi de message dynamique de longueur variable.

2. PARAMETRAGE ETS

Avec le programme d'application SKX Advance, il est possible d'intégrer n'importe quel dispositif final avec l'interface RS-232 dans une installation domotique KNX, dont les trames de communication de chaque ordre sont connues.

SKX Advance dispose de **65 objets de communication de différentes tailles** avec lesquels il sera possible d'échanger entre le protocole KNX et RS-232, en plus de plusieurs objets de 1 bit pour détection des erreurs. Chaque trame activée aura son objet de communication associé. Tous ces objets seront étudiés plus en détail dans les chapitres suivants.

Voici une présentation des paramètres ETS, qui seront à configurer pour obtenir les résultats espérés, ainsi que la présentation visuelle, sous ETS, du programme d'application.



N°	Nom	Fonction de l'Objet	longueu
65	Code d'erreur		1 bit
66	Erreur: Longueur impaire		1 bit
67	Erreur: Utilisation incorrecte de '*' et '?'		1 bit
68	Erreur: Utilisation incorrecte de '@'		1 bit
69	Erreur: Somme de contrôle incorrecte		1 bit
70	Erreur: Utilisation incorrecte du '\$'		1 bit
71	Erreur: Trop long		1 bit
72	Erreur: Réception		1 bit
73	Erreur: Non hexadécimale		1 bit

Figure 2.1. Objets de communication par défaut

Dans les chapitres suivants sont détaillée chacune des pages de configuration qui s'affiche lors du clique "Éditer paramètres".

2.1. CONFIGURATION DE LA COMMUNICATION.

La page de configuration par défaut qui apparaît lors de l'importation du programme d'application SKX Advance se présente comme suit:

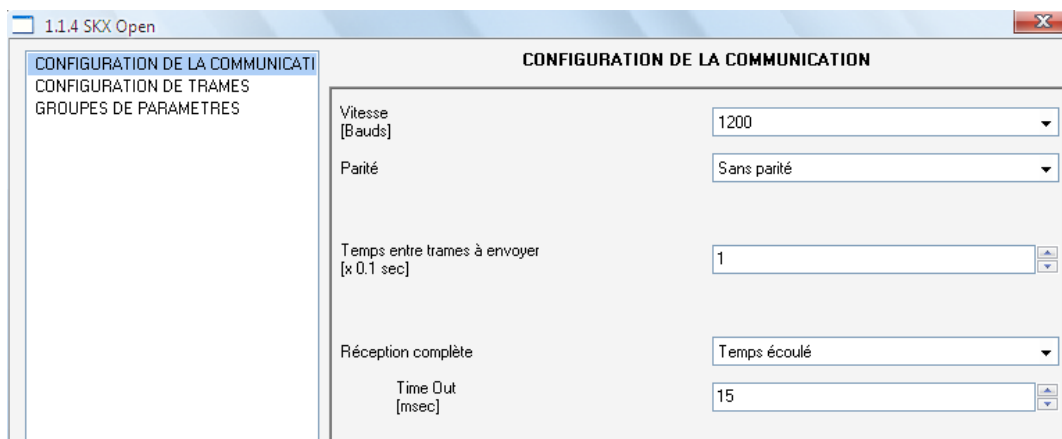


Figure 2.2. SKX Advance: Configuration de la communication

Dans celle-ci il sera possible de configurer les paramètres généraux, en lien avec la transmission des trames, suivants:

- **Vitesse (bauds):** 1200, 2400, 9600, 19200
- **Parité:** Sans parité, paire, impaire
- **Temps entre trames à envoyer (en dixièmes de seconde):**
 Configuration d'un temps d'attente minimum entre les trames à envoyer sur le port série. Ce paramètre est utilisé pour séparer différentes trames consécutives envoyées au dispositif final. Ceci peut être utile, par exemple, si plusieurs objets de communication sont associés dans la même adresse de groupe. Le SKX Advance pourra ainsi envoyer ces trames de manière ordonnée, permettant une parfaite acquisition et interprétation des données par le dispositif final. (**Note:** *Faire attention aux caractéristiques du dispositif final pour configurer le paramètre "Temps entre trame"*).
- **Réception complète:** Indique la manière qu'utilise le SKX Advance pour interpréter la bonne réception de tous les caractères de la trame. Pour la reconnaissance de la fin de trame, deux options sont possibles:
 - **Temps écoulé (ou Time Out):** Un temps minimum (en millisecondes) entre les trames est ici établi. SKX Advance interprétera la bonne réception de toute la trame une fois passé ce temps, consécutif à la réception du dernier bit de celle-ci.
 - **Byte de fin de trame:** un byte avec une valeur dont l'interprétation sera univoque est défini, de telle façon que le SKX Advance interprétera comme conclu la réception de la trame une fois reçu ce byte. Si cette option est choisie, une nouvelle case apparaîtra avec le paramètre "Time Out de sécurité". Ce temps correspond à l'attente maximale, du SKX Advance, avant réception du byte de fin de trame.

Note: Au moment de définir un Time Out pour la réception de trames, il faudra prendre en compte la durée de l'envoi de chaque byte, spécifié dans la table suivante:

Vitesse (bauds)	Parité	Temps par byte (msec)
1200	Oui	9167
	Non	8333
2400	Oui	4583
	Non	4167
4800	Oui	2292
	Non	2083
9600	Oui	1146
	Non	1042
19200	Oui	0573
	Non	0521

Si, **par exemple**, il est choisi un Time Out de 3 millisecondes et une vitesse de communication de 2400 bauds, le SKX Advance générera une "erreur de Time Out" pour chaque donnée d'entrée. En effet, il est nécessaire 4.583 millisecondes (dans le cas d'avoir un bit de parité) pour compléter le transfert d'un byte, or cette valeur est supérieure au Time Out configuré. (**Note:** Cette erreur de Time Out n'est pas indiquée sur les objets de détections d'erreurs).

Afin de s'assurer la bonne compréhension du concept du time out, il est conseillé de lire l'exemple suivant:

Exemple: Un dispositif final met 80ms pour envoyer une trame complète.

Première cas (Time Out trop long): Le "Time Out" défini est de 30ms. Le dispositif final envoie une deuxième trame immédiatement après la première. La figure suivante schématise ce fonctionnement.

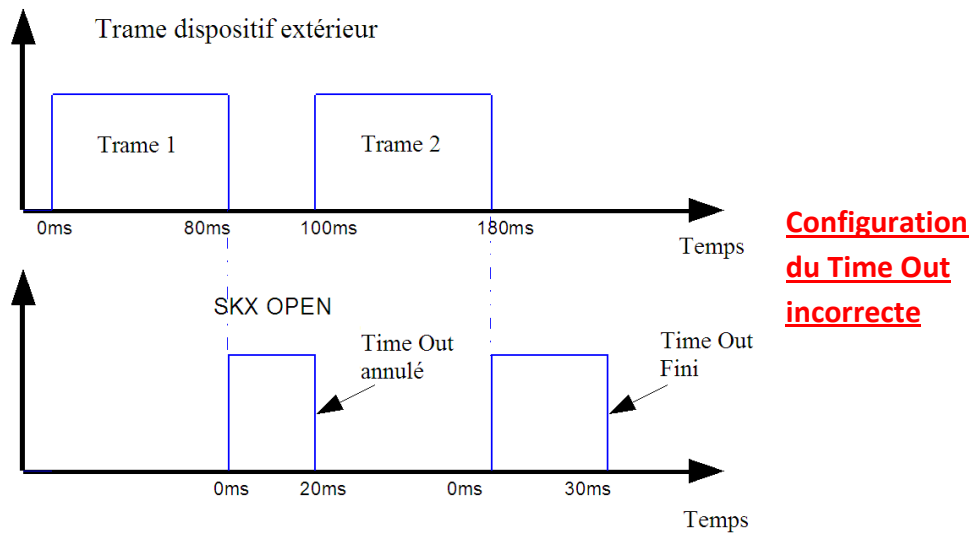


Figure 2.3. Time Out trop long

A la fin de la première trame, le Time Out commence à compter, mais une trame arrive avant la fin de son cycle et recommence à compter à partir de la fin de la deuxième trame. Maintenant, comme le Time Out arrive à sa fin (30ms), le SKX Advance considère que la trame est finie. Mais, comme deux trames sont arrivées avant la fin du Time Out, le SKX Advance considère la trame inconnue et n'enverra rien.

- 🌐 **Deuxième cas (Time Out correctement défini):** Le "Time Out" défini dans ce cas est de 10ms. La figure suivante schématise ce fonctionnement:

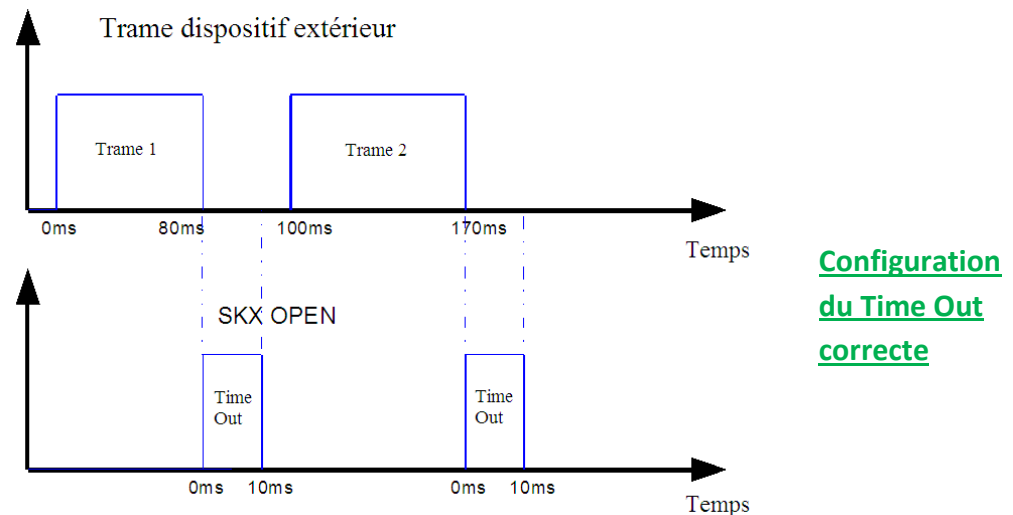


Figure 2.4. Time Out bien défini

Dans ce cas, le Time Out est bien défini et le SKX Advance est capable de reconnaître les deux trames envoyées.

C'est pourquoi, il est particulièrement important de définir correctement le Time Out en fonction du temps existant entre deux trames envoyées par le système final. Comme nous l'avons vu, un temps mal défini (trop long ou trop court) peut engendrer des erreurs dans la réception des messages.

2.2. CONFIGURATION DES TRAMES

2.2.1. DEFINITION DES TRAMES

La configuration des trames de communication est réalisé avec par des **caractères hexadécimaux** (2 caractères par byte). C'est pourquoi, il est uniquement permis d'utiliser les caractères compris entre 0-9 et A-F (excepté dans le cas des caractère spéciaux), et, les caractères A-F en lettres **majuscules**.

Note: *Il faut savoir qu'une trame hexadécimale, par exemple: "0x2B 0x7F 0x34" devra être rentré, dans l'ETS, sous le format "2B7F34".*

Dans les paramètres de l'ETS, l'utilisateur dispose d'une case de paramétrage, de type texte admettant **jusqu'à 20 caractères**. De façon à augmenter la taille des trames de communication et obtenir une plus grande flexibilité dans la communication (données variable, trames de taille non fixe, etc.) il est possible de configurer plusieurs sections de la trame. Grâce à cela, les trames reçues/envoyées pourront atteindre une longueur de **jusqu'à 29 bytes**.

Les trames de communication peuvent être **impaires**; ce qui doit être obligatoirement **paire** est l'ensemble des caractères introduits par paramètre (2 caractères par byte).

Le non respect des règles de paramétrage des trames impliquera l'envoi **d'états d'erreurs**, sur le BUS KNX via les objets de communication d'erreur activés à cet effet, après la programmation du SKX Advance. Les différents objets seront expliqués plus tard dans ce manuel.

2.2.2. TRAMES SPECIALES

Il existe la possibilité de configurer une série de trames spéciales permettant une communication complète avec l'appareil à intégrer dans l'installation KNX via RS-232.

L'utilisation de ces trames est complètement **optionnelle**. L'objectif de ces trames spéciales est l'obtention des ressources suffisantes pour générer des messages dynamique dans un protocole de communication, permettant ainsi l'utilisation d'en-tête, de pieds de trame, de sous-trames et l'envoi d'accusés de réception.

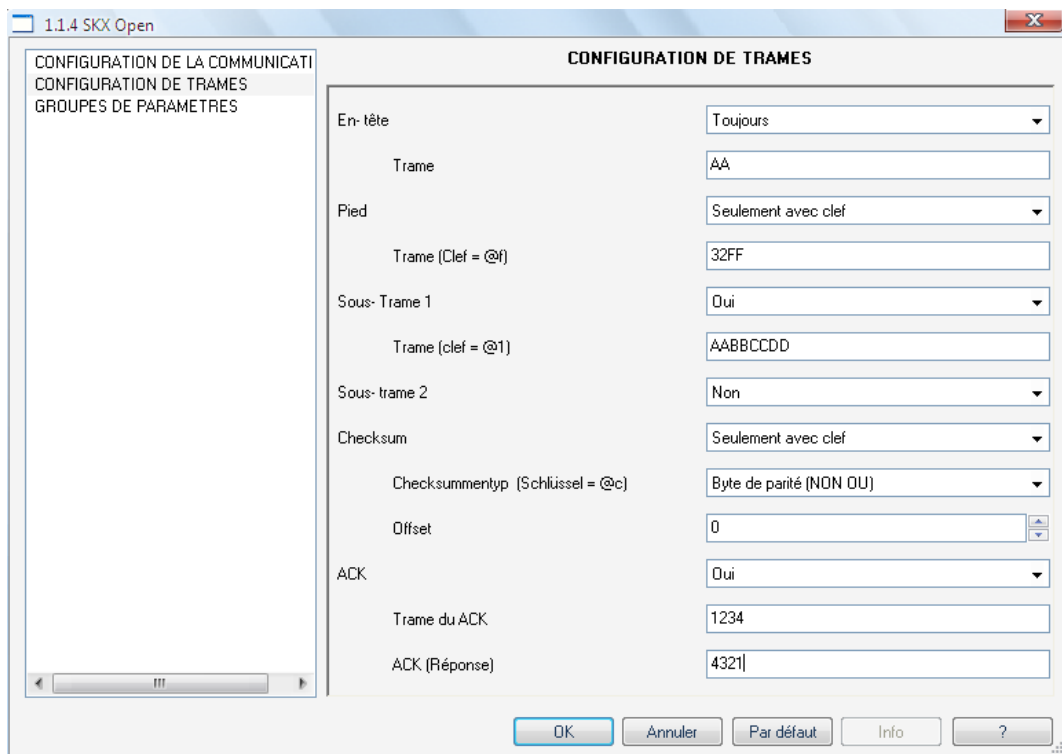


Figure 2.5. SKX Advance: Configuration des trames (trames spéciales)

Voici une explication détaillée de chacune des trames spéciales:

- **En-tête:** Cette trame pourra apparaître au début de chacune des trames (reçues et envoyées) de manière automatique ("Toujours") ou uniquement lorsqu'elle est nécessaire ("seulement avec clef") via le caractère spécial '@h'. S'il est choisi l'option "Non", l'en-tête ne sera pas inclus dans la trame.
- **Pied:** Cette trame spéciale est introduite à la fin des trames définies par paramètre. Comme pour l'En-tête, il sera possible de désactiver l'insertion de cette trame ("Non"), de l'insérer à la fin de chaque trame ("Toujours"), ou l'insérer uniquement lorsqu'elle est nécessaire ("Uniquement avec clef") via le caractère spécial '@f'.
- **Sous-trames 1 et 2:** Elles pourront être insérées à n'importe quel endroit de la trame définie par paramètre via les clefs '@1' et '@2' respectivement.
- **Checksum:** (ou Somme de contrôle). Cette trame spéciale permet d'introduire un checksum calculé pour chaque cas (réception et envoi), du premier bit jusqu'au bit antérieur au propre checksum. Il sera possible de l'insérer à la fin de chaque trame ou uniquement sur demande via la clef '@c' (après le pied, s'il y en a un). Enfin, il existe un paramètre additionnel (Offset) qui indique au programme que le checksum devra être calculé à partir du byte indiqué par cette valeur.

Les types de checksum supportés sont les suivants:

- **Mot de parité:** Opération OU EXCLUSIF byte à byte
- **Somme modulaire:** Somme de tous les bytes de la trame avec modula 256.
- **Somme modulaire avec complément à 2:** Identique au précédent, mais dans ce cas, au résultat lui est effectué un complément à 2.
- **CRC-8 et CRC-16:** Code de redondance cyclique de 8 ou 16 bits respectivement. Dans tous les cas, il sera nécessaire d'introduire un numéro (en décimal) qui défini le polynôme caractéristique du CRC.

Accusé de réception (ACK) SKX advance offre la possibilité d'envoyer automatiquement des trames d'accusées de réception fixes après chaque trame reçue du port série. Cette trame de bonne réception se définira dans le champ "Trame du ACK". Le programme d'application permet également de configurer une trame spécifique (via le champ "ACK (Réponse)") correspondant à l'accusé de réception émis par le système final de tel façon que, dans le cas de recevoir la trame "ACK (réponse)", le SKX-Advance n'enverra aucun accusé de réception. Il faut donc connaître et écrire dans ce champ les trames ACK envoyées par le dispositif final.

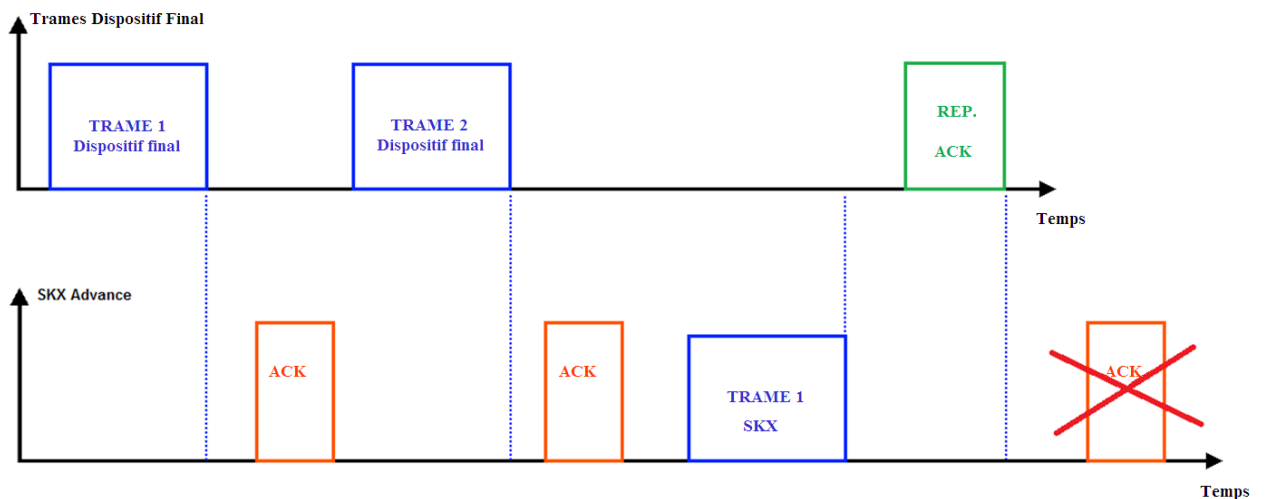



Figure 2.6. Trames ACK et réponse ACK

2.2.3. CARACTERES SPECIAUX

Comme vu précédemment, une liste de caractères spéciaux, permettant la génération et détection d'un plus grand nombre de messages, pourra être ajoutée aux trames définies par paramètre, les rendant ainsi plus polyvalente.

De plus, l'utilisation de ce type de caractères permet d'utiliser des expressions régulières dans la définition des trames.

Deux groupes peuvent se distinguer: Caractères spéciaux pour réception et Caractères spéciaux pour transmission et réception. Voici une explication de chacun d'eux:

 **Caractères spéciaux pour réception:** Ces caractères peuvent uniquement être utilisés dans la spécification des trames reçues depuis le port série; leur utilisation dans les trames à envoyer générera un message d'erreur.

- **?1, ?2,... , ?9:** Indiquent que les 1, 2,..., 9 byte suivants seront ignorés au moment de l'analyse de la trame.

Exemple: On définit l'envoi d'un "1", sur l'objet de communication numéro 2 de 1bit, si le SKX Advance reçoit, via le port série, la trame défini par paramètre: "**AA?223**". Dans les trames envoyées par le dispositif final, se trouve les trames suivantes:

"AA857D23" → Envoi de l'objet correspondant à "1". En effet, les deux bytes 857D sont ignorés, tel qu'il l'a été spécifié, et la partie fixe de la trame coïncide avec celle paramétrée.

"AA112223" → Identique au cas précédent. Cette fois il ignore les bytes 1122.


"AA685AB923" → Dans ce cas, le SKX Advance n'envoie pas la valeur sur l'objet. En effet, la partie fixe reçue ne coïncide pas avec celle paramétrée: il ignorera les 2 bytes après le AA (685A), tel qu'il l'était défini, et interprétera B923 comme la partie fixe de la trame qui, ne correspondant pas avec le byte 23 défini, n'enverra pas la valeur 1 sur l'objet numéro 2.

- ****:** Apparition de zéro ou plusieurs bytes de n'importe quel valeur jusqu'à obtenir la partie fixe de la trame défini par paramètre (excluant les clefs @h, @f, @1, @2). Les caractères ** représentent l'ensemble minimum de caractères qui devra y avoir avant de détecter la partie fixe de la trame.

Exemple: Est paramétré l'envoi de la valeur "50", sur l'objet de communication de 1 byte numéro 41, lors de la réception par le port série de la trame: "**23**AB**". Le dispositif final envoi les trames suivantes:

“231214AB” → Le SKX Advance commence à analyser la trame: 23 et AB sont les parties fixes de celle-ci, et tout ce qui se trouve avant AB (partie constante postérieure à ** dans la définition de la trame) seront les caractères définis comme **. La trame coïncide avec celle paramétrée, l'objet 41 enverra donc la valeur 50.

“23ABAB” → Dans ce cas, le SKX avance interprète que les ** est un ensemble vide. En effet, il n'y a aucun caractère entre 23 et le premier AB rencontré dans la trame. Même si, en continuant à analyser la trame, apparaît une autre fois les caractères AB, celle-ci ne sera pas reconnue et le SKX Advance n'enverra pas la valeur 50 sur l'objet 41.

 **Caractères spéciaux pour transmission et réception:** Ces caractères peuvent être utilisés dans la définition des trames entrantes ou sortantes.

➤ **Clef des sous-trames:** Elles sont définies de la forme "@ + caractère" et représentent l'insertion des certaines des trames définies dans l'onglet "Configuration des trames", vu précédemment. Les différentes possibilités sont:

- **@h:** Insertion de la trame d'en-tête
- **@f:** Insertion de la trame de pied.
- **@1, @2:** Insertion des sous-trames 1 et 2 respectivement.
- **@c:** Insertion du byte, ou bytes, de checksum.

➤ **##:** Ce caractère indique la partie variable qui sera associé à l'objet de communication.

Objets de 1 bit: Ne peut pas être utilisé.

Objets de 1 byte: les ## représenteront toujours un byte. Il existe quatre cas d'utilisation:

- Envoi de trame variable: Le SKX Advance remplacera automatique les caractères ##, dans la trame, par la valeur de l'objet de communication correspondant.

Exemple: Est défini par paramètre la trame variable suivante: "1234##55" L'objet de communication de 1 byte numéro 44 reçoit une valeur égale à **16** (décimal) sur la BUS KNX. Le SKX-Advance enverra au dispositif final la trame "1234**10**55".

(Note: Le SKX convertit le 16 décimal en son équivalent hexadécimal (0x10) et l'insère à la position occupée par ##, dans la définition de la trame, avant l'envoi).

- **Obtenir objet variable:** L'objet de communication prendra la valeur de la portion de trame, reçu du dispositif final, se trouvant à l'emplacement des ##.

Exemple: Dans cet exemple, l'objet de communication de 1byte numéro 48 est associé avec la trame suivante: "AA##E8". Le dispositif final envoie la trame suivante: "AA02E8". L'objet de communication prendra alors la valeur 2 en décimal, et l'enverra sur le BUS KNX.

- **Envoi de trame variable (pourcentage):** Identique à l'envoi de trame variable, mais la valeur de l'objet de communication sera préalablement convertie de pourcentage KNX (0-255) à pourcentage standard (0-100).

- **Obtenir objet variable (pourcentage):** Identique que pour obtenir un objet variable mais le byte variable sera convertie de pourcentage standard (0-100) à pourcentage KNX (0-255).

Note: Dans l'envoi et réception de trames fixes, le caractère ## ne peut être utilisé.

Objets de 14 bytes: Les ## représenteront toujours une chaîne de caractères. Dans le cas de transmission de trame, ceci est l'endroit où le SKX Advance devra copier les caractères reçus du bus KNX; dans le cas de réception, ceci sera l'endroit où commencera la trame de caractère que SKX Advance devra copier dans l'objet de communication associé.

Pour voir des exemples d'utilisation des caractères spéciaux associés aux différents objets du SKX Advance, se référer aux chapitres 2.3.1, 2.3.2 et 2.3.3.

2.3. GROUPES DE PARAMETRES

Dans cet onglet, il sera possible d'activer les groupes des objets de communication à utiliser.

Le SKX Advance dispose d'un total de 65 objets de communication distribués de la façon suivante:

Taille	Nombre de groupes	Nombre d'objets de communication par groupe
1 bit	4	10
1 byte	2	10
14 bytes	1	5

Il y a 40 objets de communication de 1 bit, dont leur numéro est compris entre 0 et 39: 20 objets de 1 byte, entre 40 et 59, et 5 objets de 14 bytes, entre 60 et 64.

Lors de l'activation des paramètres, les onglets correspondants à chacun d'eux s'affichent:

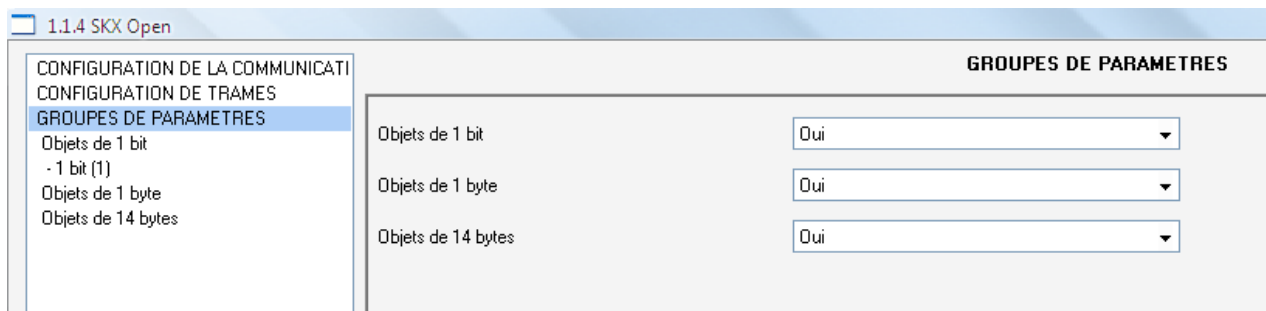


Figure 2.7. SKX Advance: Groupes de paramètres

Dans chacun des groupes des objets de communication, il faudra activer les objets avec lesquels il est souhaité travailler, permettant ainsi d'accéder aux options disponibles pour chaque type d'objet de communication.

La suite explique les différents types d'objets de communication disponibles:

2.3.1. OBJETS DE 1 BIT

Les objets de 1 bit permettent d'envoyer, depuis le SKX, une trame de données au dispositif final, par le port RS-232, quand le SKX reçoit du BUS la valeur préalablement paramétrée dans ETS (0 ou 1) sur l'objet configuré (objet 0 à 39). Ce type d'objet permet également à l'interface d'envoyer une valeur déterminée (0 ou 1) sur un objet de communication spécifique (0-39) quand le SKX reçoit une trame fixe, depuis la liaison RS-232, provenant du dispositif final.

Objet	Objet	Type de contrôle	Trame
Objet 0:	Oui	Envoyer trame si l'objet vaut 0	15301A
Objet 1:	Oui	Envoyer trame si l'objet vaut 1	5200AD
Objet 2:	Oui	Objet égal à 0 si la trame coïncide	AABBCCFF121233444455
Objet 3:	Non		
Objet 4:	Oui	Objet égal à 1 si la trame coïncide	0206AB
Objet 5:	Non		
Objet 6:	Non		
Objet 7:	Non		
Objet 8:	Non		
Objet 9:	Non		

Figure 2.8. Groupe 1 d'objets de communication de 1 bit

Par exemple, avec le paramétrage ci-dessus, le SKX enverra sur la liaison RS-232 la trame "15301A" s'il est reçu, depuis le BUS KNX, un 0 sur l'objet de 0. Si l'objet 1 reçoit un 1, le SKX enverra la trame "5200AD" au dispositif final.

De même, si le dispositif final envoie la trame "AABBCCFF121233444455", le SKX écrira un 0 sur l'objet de 1 bit numéro 2, alors que s'il envoie la trame "0206AB", depuis la liaison RS232, le SKX écrira un 1 sur l'objet numéro 4.

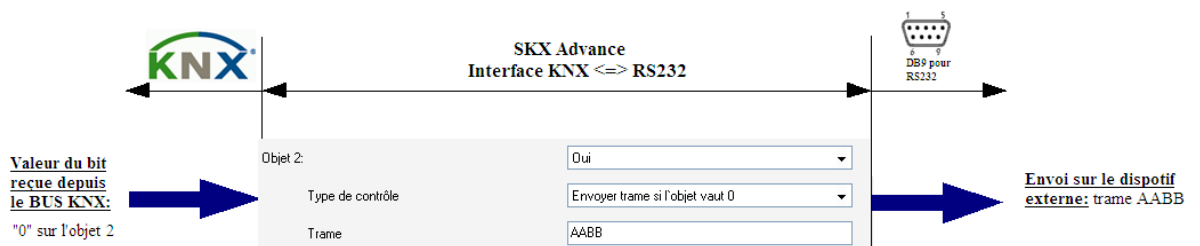
Lors de l'activation d'un objet de communication de 1 bit, deux options apparaissent: une pour sélectionner le type de contrôle et une autre pour définir la trame.

- **Objet X. Type de contrôle:** Pour chaque objet, il existe les quatre types de contrôle suivants:

Pour la Communication KNX → RS232

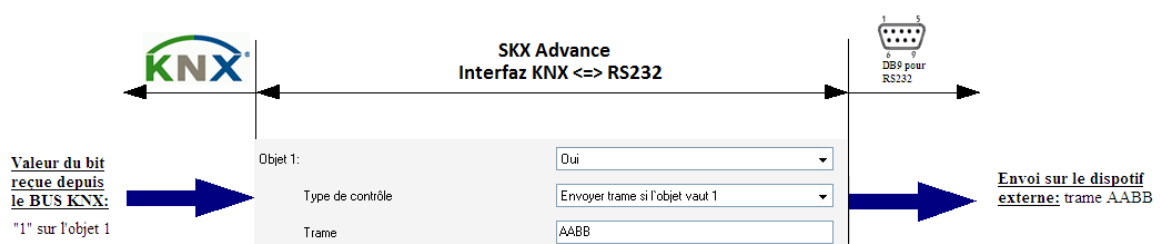
- **Envoyer trame si l'objet vaut 0:** Envoyer la trame (rentré dans le paramètre "Objet X: Trame") au dispositif final lors de la **réception d'un "0"**, depuis le BUS KNX, sur l'objet de communication correspondant:

Exemple:



- **Envoyer trame si l'objet vaut 1:** Envoyer la trame (rentré dans le paramètre "Objet X: Trame") au dispositif final lors de la **réception d'un "1"**, depuis le BUS KNX, sur l'objet correspondant.

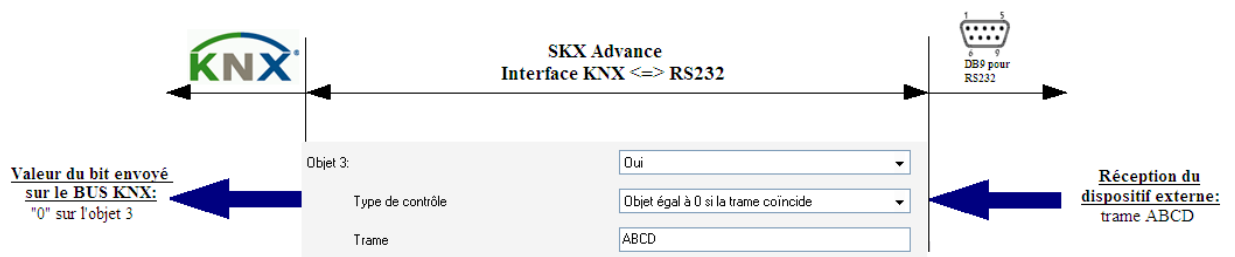
Exemple:



Pour la Communication RS232 → KNX

- **Objet égal à 0 si la trame coïncide:** Envoyer un "0" sur l'objet de communication lorsqu'il est reçu, depuis le port série, une trame qui coïncide avec celle rentrer par paramètre.

Exemple:



- **Objet égal à 1 si la trame coïncide:** Envoyer un "1" sur l'objet de communication lorsqu'il est reçu, depuis le port série, une trame qui coïncide avec celle rentrer par paramètre.

Exemple:



- **Objet X. Trame:** Dans ce paramètre sont définies les trames nécessaires à la communication. Celles-ci devront respecter les normes commentées dans le chapitre "2.2.1. Définition des trames".

2.3.2. OBJETS DE 1 BYTE

Les objets de 1 byte permettent d'envoyer, du SKX vers RS232, une trame de données lorsque le SKX reçoit du BUS la valeur préalablement paramétrée dans ETS (0-255) sur l'objet configuré (objet 40 à 59). Ce type d'objet permet également à l'interface d'envoyer une valeur déterminée (0-255), sur un objet de communication choisi (40-59), lorsque le SKX reçoit une trame spécifique provenant du dispositif final. Les trames pourront être fixes ou variables, en fonction de la valeur du byte de l'objet. De même, la valeur de l'objet pourra être fixe ou fonction de la trame reçue.

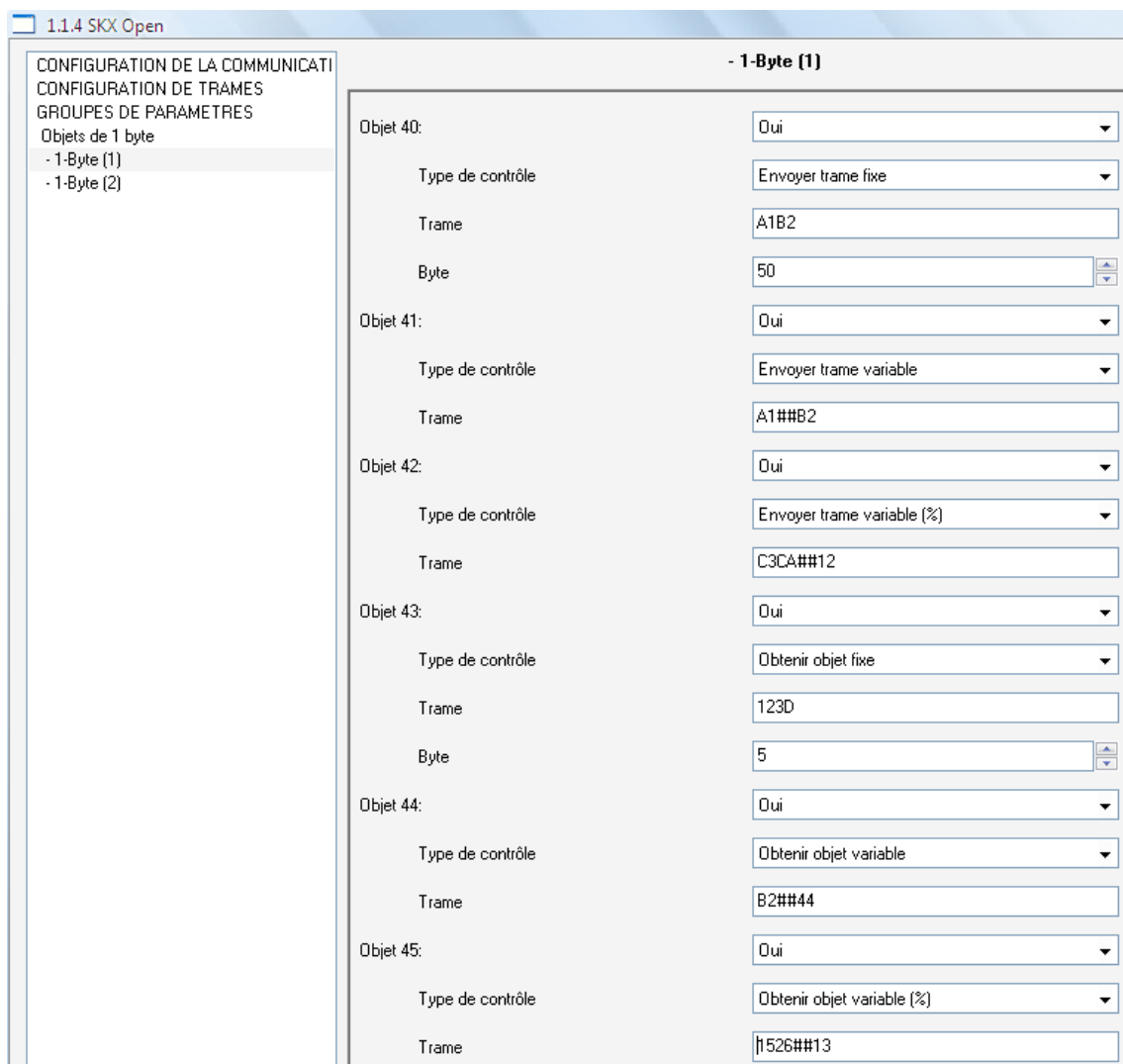


Figure 2.9. Groupe 1 d'objets de communication de 1 byte

Par exemple, avec la configuration ci-dessus, le SKX enverra la trame "A1B2", sur la liaison RS-232, quand il recevra, depuis le BUS KNX, la valeur 50 (décimale) sur l'objet de communication numéro 40. De même, quand le SKX reçoit la trame "123D", provenant du dispositif final, il écrira la valeur 5 sur l'objet de 1 byte numéro 43.

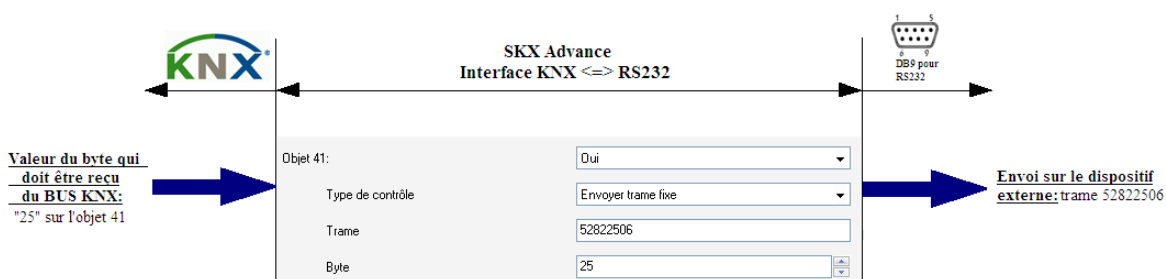
Lors de l'activation d'un objet de communication de 1 byte, deux options apparaissent: une pour sélectionner le type de contrôle et une autre pour définir la trame.

- 🌐 **Objet X. Type de contrôle:** Pour chaque objet, il existe six types de contrôles différents:

Pour la Communication KNX → RS232

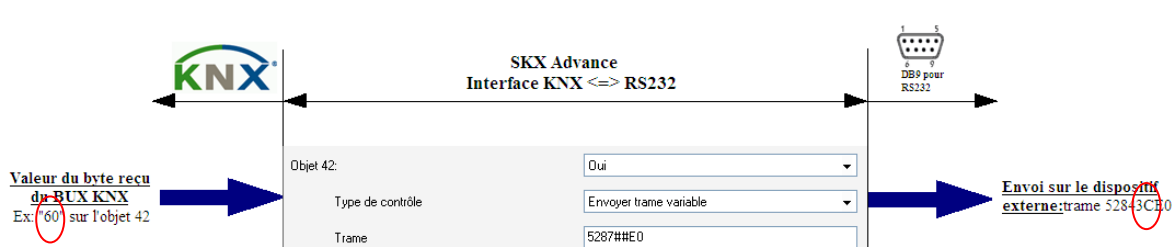
- **Envoyer trame fixe:** Envoyer la trame (rentrer dans le paramètre "Objet X: Trame") au dispositif final lors de la réception de la valeur définie dans l'objet. Lors de l'activation de ce type de contrôle, une nouvelle option apparaît: "**Objet X. Byte**": Ici est définie la valeur du byte que le SKX Advance devra recevoir, sur l'objet de communication, pour envoyer la trame définie.

Exemple:



- **Envoyer trame variable:** Envoyer la trame (rentrer dans le paramètre "Objet X: Trame") vers le dispositif final lors de la réception d'une valeur sur l'objet de communication (valeur dans la clef ##). La trame envoyée changera en fonction de la valeur reçue sur cet objet et sera la même que celle définie par paramètre, en remplaçant les ## par la valeur du byte reçu sur l'objet de communication correspondant.

Exemple:

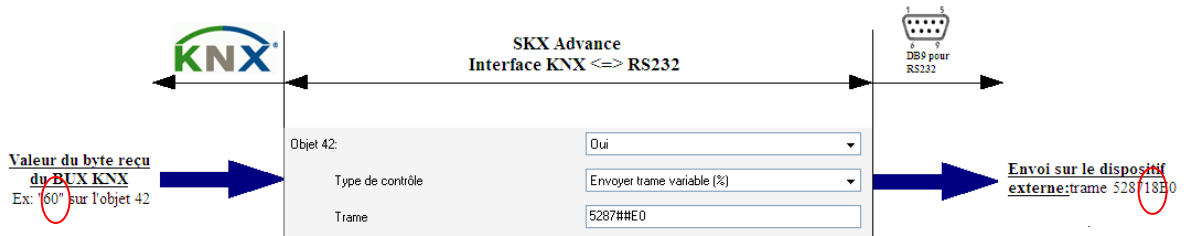


Note: Dans cet exemple, le SKX Advance reçoit une valeur "60" via le BUS KNX. Cette valeur est en **décimale**. La trame envoyée au dispositif final apparaît sous la valeur "3C" en **hexadécimale**, qui correspond à 60 en décimale. La conversion décimale/hexadécimale s'effectue en interne dans le SKX Advance.

- **Envoyer trame variable (%):** Envoyer la trame (rentrer dans le paramètre "Objet X: Trame"), vers le dispositif final, à la réception d'une valeur sur l'objet de communication correspondant (valeur dans la clef ##). La valeur reçue sera convertie en un nombre de 0 à 100 (%) et sera incorporée à la trame définie par paramètre. La trame envoyée changera en fonction de la valeur reçue sur cet objet

(en pourcentage) et sera la même que celle définie par paramètre, en substituant les ## par la valeur du byte reçu sur l'objet de communication correspondant (en pourcentage).

Exemple:

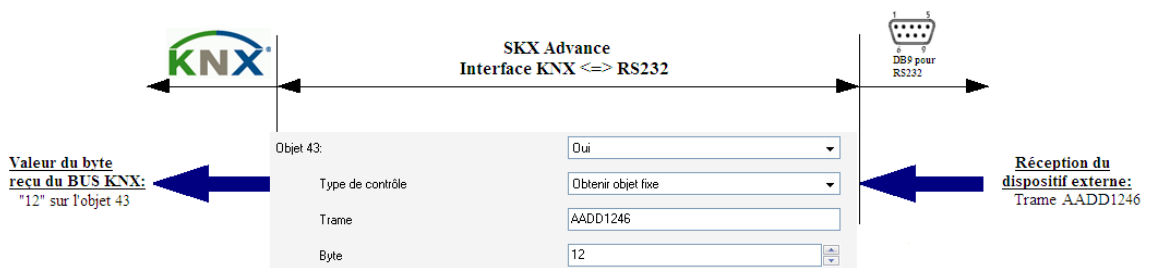


Note: Dans cet exemple, le SKX Advance reçoit la valeur "60" en décimale. Cette valeur est un pourcentage KNX (0-255), qui est convertie à un pourcentage standard (0-100%) via une simple règle de trois ($60 \cdot 100 / 255$), soit une valeur équivalente à 24%. Cette valeur est convertie en hexadécimale, donnant comme résultat "18", valeur qui remplacera les caractères ## dans la trame définie par paramètre.

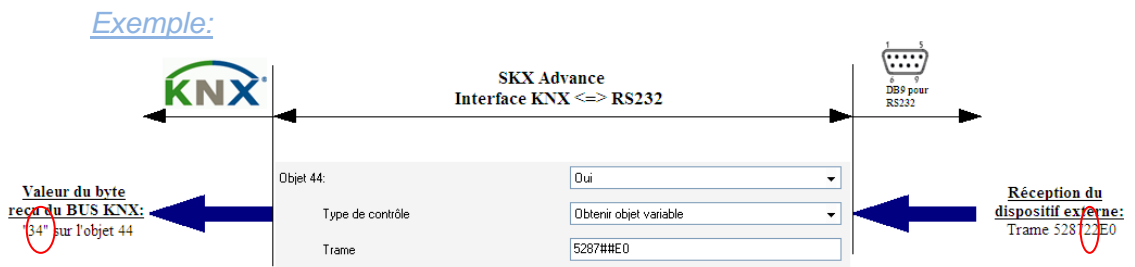
Pour la Communication RS232 → KNX

- **Obtenir objet fixe:** Envoyer sur le BUS KNX, via l'objet de communication associé, la valeur définie dans "byte" lorsqu'il est reçu, depuis le port série, une trame qui coïncide avec celle rentrée dans le paramètre "Trame". Lors de l'activation de ce type de contrôle, une nouvelle option apparaît: "Objet X. Byte": Ici est définie la valeur qui sera envoyé sur le BUS KNX, via l'objet de communication associé, à la réception de la trame définie.

Exemple:

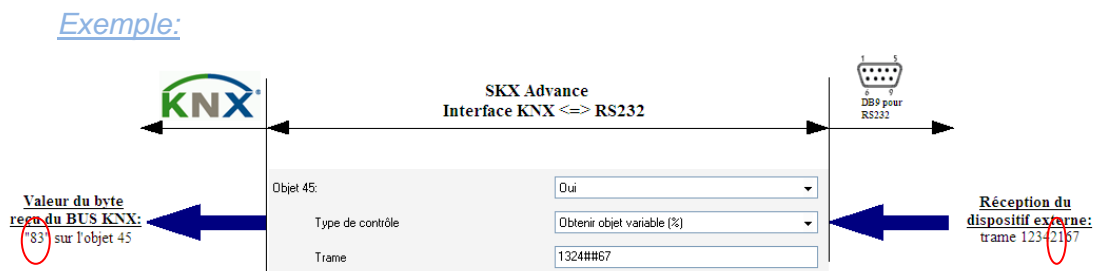


- **Obtenir objet variable:** Envoyer sur le BUS KNX, via l'objet de communication associé, une valeur lorsqu'il est reçu, depuis le port série, une trame coïncidant avec celle rentrée dans le paramètre "Trame". La valeur de l'objet de communication sera celle correspondant à la partie ## de la trame reçue.



Note: Dans cet exemple, le dispositif final envoie la trame 528722E0 en hexadécimale. La valeur que prendra l'objet de communication, numéro 41, sera celle correspondant à la partie ## reçue, qui, dans ce cas est 22 (en hexadécimale), ou "34" en décimale.

- **Obtenir objet variable (%):** Envoyer une valeur sur un objet de communication lorsqu'il est reçu, depuis le port série, une trame qui coïncide avec celle rentrée par paramètre. La valeur de l'objet de communication sera celle correspondant à la partie ## de la trame reçue (en pourcentage).



Note: Dans ce cas, le SKX Advance reçoit la valeur "21" en hexadécimale, partie de la trame correspondant à ##. Cette valeur correspond à la valeur décimale "33", interprétée, cette fois, comme un pourcentage standard (0-100%), qui sera convertie en pourcentage KNX (0-255) grâce à une simple règle de trois ($33 * 255 / 100$)

- **Objet X. Trame:** Dans ce paramètre sont définies les trames nécessaires à la communication. Celles-ci devront respecter les normes commentées dans le chapitre "2.2.1. Définition des trames".

2.3.3. OBJETS DE 14 BYTES

Les objets de 14 bytes permettent l'envoi ou la détection de chaîne de texte dans des trames de port série.

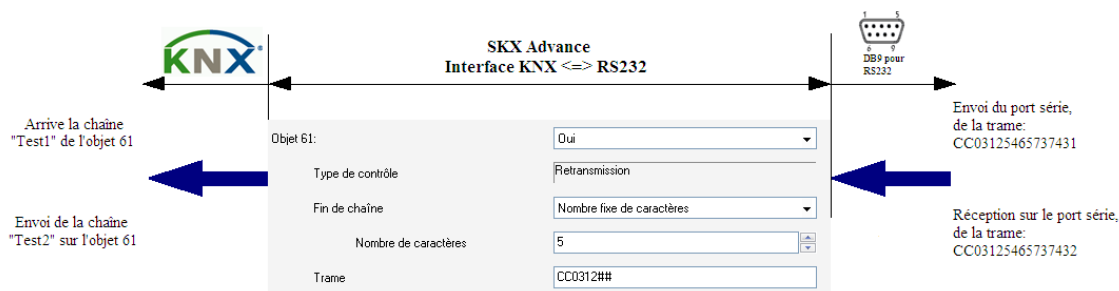
Objet	Type de contrôle	Fin de chaîne	Identificateur Fin de chaîne	Nombre de caractères	Trame
Objet 60:	Dui	Identificateur de fin de chaîne	0		
Objet 61:	Dui	Nombre fixe de caractères		1	
Objet 62:	Non				
Objet 63:	Non				
Objet 64:	Non				

Figure 2.11. Groupe 1 d'objets de communication de 14 bytes

Lors de l'activation d'un objet de 14 bytes, apparaissent trois nouvelles options permettant de choisir le type de contrôle, le type de fin de chaîne et définir la trame correspondante.

- **Objet X. Type de contrôle:** Il existe uniquement le type de contrôle "Retransmission". Ce type de contrôle permet de retransmettre une chaîne de texte variable, reçue du port série, vers le BUS KNX et vice versa (à partir des caractères ##).
- **Objet X. Fin de chaîne:** Ce champ permet de choisir la manière de confirmer la fin de chaîne. Il y a deux options disponibles:
 - **Nombre fixe de caractères:** Choix d'une valeur fixe de caractères (de 1 à 14) que devra avoir la chaîne de texte envoyée ou reçue.

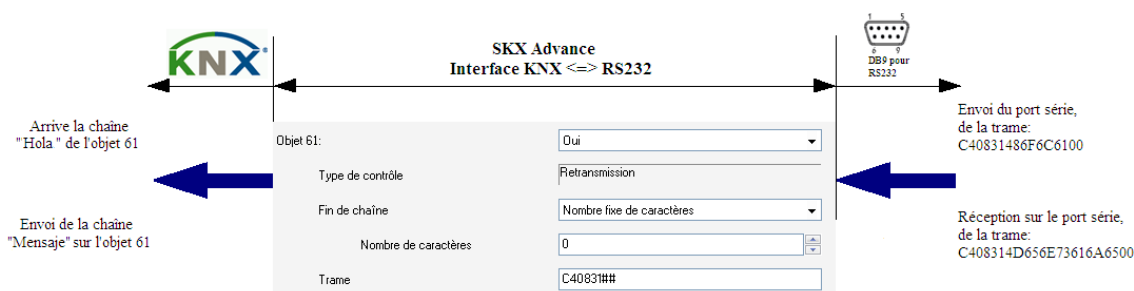
Exemple:



Note: Dans cet exemple, arrive la chaîne "Test1", de 5 caractères, sur l'objet de communication numéro 61. La réaction associée est l'envoi, par le port série de la trame suivante: "CC0312**5465737431**", avec CC0312 la partie fixe de la trame définie par paramètre, et "5465737431" la codification en ASCII, caractère par caractère, de la chaîne "Test1". De même, lorsque le SKX Advance recevra, depuis le port série, la trame "CC0312**5465737432**", la chaîne de caractère "Test2", de 5 caractères, sera envoyée sur l'objet de communication 61.

- **Identificateur fin de chaîne:** Définir le byte qui indique la fin de la trame. Ce byte possède en général la valeur 0x00.

Exemple:



Note: Ici, il a été défini un identificateur de fin de trame 0x00 (écrire un 0 dans la case correspondante). Sur l'objet numéro 61 arrive la chaîne de texte "Hola", qui impliquera un envoi, sur le port série, de la trame "C40831**486F6C6100**" où C40831 est la partie fixe de la trame définie par paramètre, et 486F6C61 est la codification ASCII, caractère par caractère, de la chaîne "Hola", et enfin, 00 qui est l'identificateur de fin de chaîne définie. De même, si le SKX Advance reçoit par le port série, la trame "C40831**4D656E73616A6500**", la chaîne de texte "Mensaje" sera envoyé sur l'objet de communication 61.

- **Objet X. Trame:** Dans ce paramètre sont définies les trames nécessaires à la communication. Elles devront respecter les normes commentées dans le chapitre "2.2.1. Définition des trames".

2.4. OBJET D'ERREUR

Le SKX Advance dispose de plusieurs objets de 1 bit qui informent sur les erreurs dans le fonctionnement du programme. Ces erreurs sont principalement dues à un mauvais paramétrage, mais, il arrive que celles-ci se produisent pendant la communication. A chaque fois que le SKX Advance détectera une erreur, il enverra l'objet de 1 bit correspondant et, en plus, un objet d'erreur général de 1 bit ("Code d'erreur").

Voici une description des erreurs qui peuvent se produire:

- **Erreur: Longueur Impaire.** L'ensemble des caractères de certaines des trames introduites par paramètre sont impaires. (**Note:** *Se rappeler que cela ne signifie pas que le nombre de bytes de chaque trame - Longueur de trame - devra être pair, mais que, comme chaque byte possède 2 caractères, il sera nécessaire d'avoir un nombre pair de caractères*).
- **Erreur: Utilisation incorrecte de "*" ou "?".** Il y a une erreur après les caractères ** ou ?
- **Erreur: Utilisation incorrecte de "@".** Erreur dans l'utilisation du caractère spécial @.
- **Erreur: Checksum incorrecte.** Il n'y a rien à calculer pour le checksum, par exemple parce que la trame a été configurée uniquement avec "@c" ou parce que l'offset du checksum est trop grand.
- **Erreur: Utilisation incorrecte de "#".** Erreur dans l'utilisation de ##. Possible erreur de syntaxe ou impossibilité d'associer ## avec une donnée variable.
- **Erreur: Non hexadécimal.** Dans certaines des trames définies par paramètre, dans lesquelles il était attendu un caractère de valeur 0-9 ou A-F, le SKX Advance en a rencontré une avec une valeur différente (par exemple, une lettre en minuscule)

Note: Toutes les erreurs précédentes sont des erreurs de paramétrage.

- **Erreur: Trop long.** La longueur de la trame à envoyée ou à recevoir est supérieure au maximum autorisé: 29 bytes. (Cette erreur peut être de paramétrage ou de communication).
- **Erreur: réception.** Erreur indéterminée de réception du port série. (Erreur de communication).

A chaque fois que revient la tension de BUS, le SKX réalise une analyse de toutes les trames introduites par paramètre et s'il détecte une erreur dans les trames, il enverra les erreurs correspondantes.

Il se produira un envoi uniquement pour chaque type d'erreur, c'est-à-dire, que s'il est détecté sur une ou plusieurs trames le même type d'erreur (par exemple, mauvaise utilisation du caractère spécial "*"), celle-ci sera signifiée une seule fois.


Si une trame possède plusieurs erreurs, elles seront toutes notifiées sauf dans le cas de la détection d'une trame impaire, dans cette situation sera notifiée l'erreur de trame impaire uniquement. Par exemple, la trame "AaB" possède une erreur de trame impaire et une erreur de caractère non hexadécimal; le SKX Advance détectera et notifiera uniquement l'erreur de la trame impaire, l'autre sera ignorée.

Le SKX Advance notifie également lorsqu'il essaie d'envoyer, sur le port série RS-232, une trame mal définie. Dans ce cas, **il notifiera la première erreur détectée dans la trame**. Par exemple, en supposant que le SKX Advance tente d'envoyer la trame "AAaB?B", trame qui contient deux erreurs (il y a une valeur non hexadécimal et une mauvaise utilisation de '?'), le SKX Advance arrêtera l'envoi dès détection de la première erreur du caractère non hexadécimal et enverra UNIQUEMENT cette erreur sur le BUS.

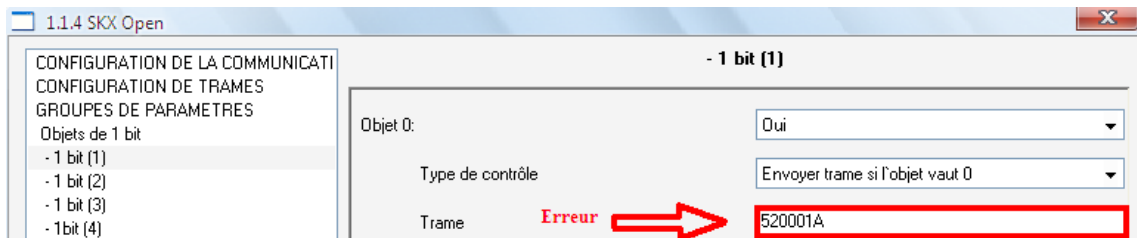
A savoir, le SKX Advance notifie les erreurs détectées dans les trames après la programmation et chaque fois que seront transmises ou reçues des trames erronées.

2.4.1. EXEMPLES D'ERREURS

La suite présente une série d'exemple des erreurs que SKX Advance peut détecter.

-  **Longueur Impaire.** Dans le cas d'essayer d'envoyer une trame avec un ensemble de caractère défini par paramètre impair, l'objet de communication de 1bit "Erreur: Longueur impaire" s'activera, et enverra un "1". L'objet de 1 bit "Code d'erreur" sera également activé et envoyé sur le BUS.

Exemple: Dans l'objet 0 est défini la trame "520001A", avec une longueur impaire (7 caractères).



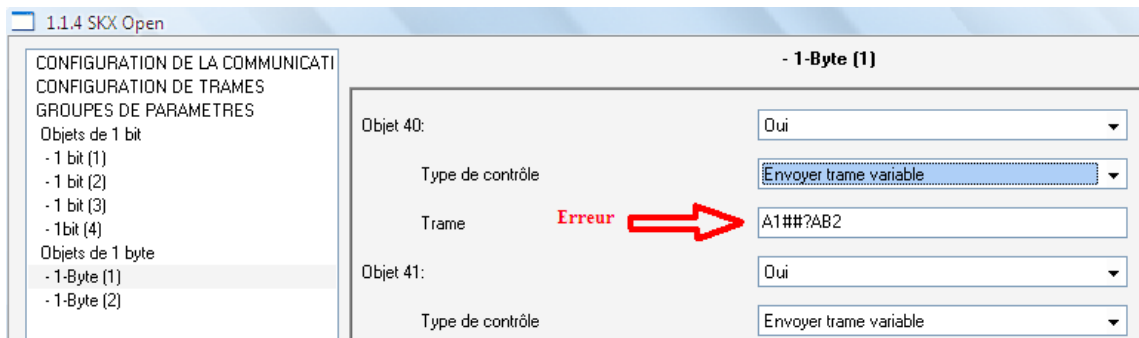
Ceci provoquera l'indication d'erreur suivante:

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	CODE ERREUR	6	1 bit	Write	\$01
367	14:25:34.504	desde ...	L	1.1.20		1/0/3	IMPAIRE	6	1 bit	Write	\$01

Figure 2.11. Objets d'erreur pour une trame de longueur impaire

Utilisation incorrecte des caractères "*" ou "?". Si, suite au caractère spécial '?', il n'est pas détecté une valeur se trouvant dans l'intervalle 0 à 9, le SKX Advance activera l'objet de communication de 1 bit "Erreur: Utilisation incorrecte de '*' ou '?'", et l'erreur "Code d'erreur". Ces erreurs seront également activées si, suite à l'utilisation des caractères '**', n'apparaît pas une constante.

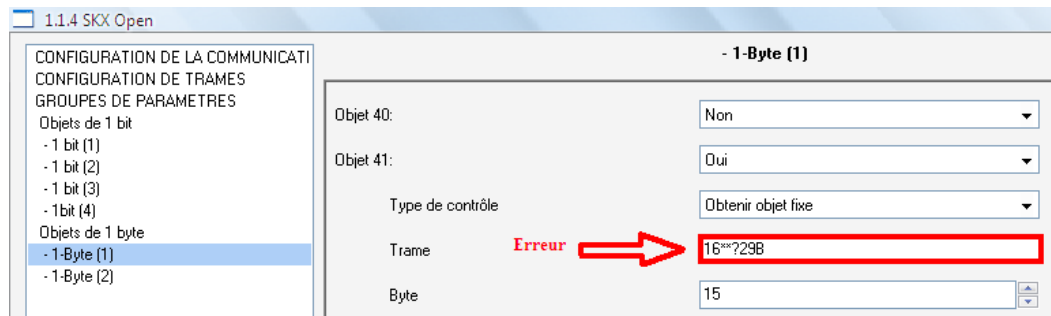
Exemple I: Dans l'objet de 1 byte numéro 41 est défini l'envoi de la trame "A1##?AB2", où apparaît le caractère 'A' après le '?', valeur erronée (en effet, il est attendu un nombre entre 0 et 9).



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	CODE ERREUR	6	1 bit	Write	\$01
367	14:25:34.504	desde ...	L	1.1.20		1/0/3	* OU ?	6	1 bit	Write	\$01

Figure 2.12. Objets d'erreurs pour utilisation incorrecte du caractère '?'.

Exemple II: Dans l'objet de 1 byte numéro 41 est défini l'écriture de la valeur 15 quand le SKX recevra, depuis le dispositif final, la trame "16**?29B". Le SKX générera une erreur d'utilisation incorrecte de **. En effet, après les caractères **, il n'y a pas de valeur constante, sinon des caractères "?2" qui indique l'attente de deux valeurs variables.

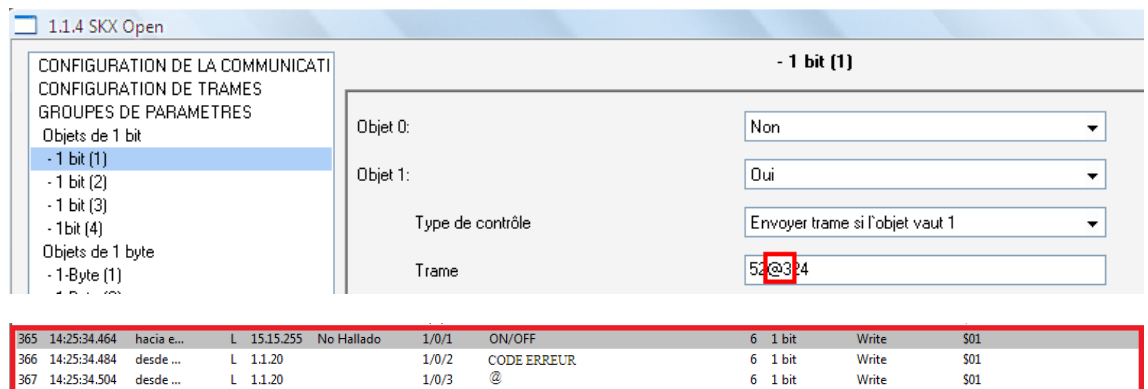


365	14:25:34.464	hacia e...	L	15.15.255	No Hallado	1/0/1	ON/OFF	6	1 bit	Write	\$01
366	14:25:34.484	desde ...	L	1.1.20		1/0/2	CODE ERREUR	6	1 bit	Write	\$01
367	14:25:34.504	desde ...	L	1.1.20		1/0/3	* OU ?	6	1 bit	Write	\$01

Figure 2.13. Objets d'erreurs pour utilisation incorrecte du caractère '***'

Utilisation incorrecte de "@". Le caractère spécial '@' permet d'introduire des en-têtes, sous-trames ou pieds dans une trame. Les codes, ou clefs, pour les insérés dans les trames sont des caractères prédéfinis: @h, @f, @1, @2, @c. S'il est entré un mauvais caractère après un @, le SKX Advance activera l'objet de communication de 1 bit "Erreur: Utilisation incorrect de '@"', ainsi que l'objet "Code d'erreur".

Exemple: Si l'objet de communication 1 reçoit la valeur 1 du BUS, envoyer la trame suivante: "52 @324". Le caractère 3 après @ n'est pas valide, le SKX Advance activera l'objet d'erreur correspondant et la trame de sera pas envoyée.



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	CODE ERREUR	6	1 bit	Write	\$01
367	14:25:34.504	desde ...	L	1.1.20		1/0/3	@	6	1 bit	Write	\$01

Figure 2.14. Objets d'erreur pour utilisation incorrecte du caractère '@'

Checksum incorrecte. Cette erreur informe qu'il n'est pas possible de réaliser le checksum d'une certaine trame parce qu'il n'y a pas de données (la trame définie est vide) ou parce que l'offset configuré est trop grand. Si une de ces situations se présente, le SKX Advance activera l'objet de communication de 1 bit "Erreur: Somme de contrôle incorrecte", ainsi que l'objet "Code d'erreur".

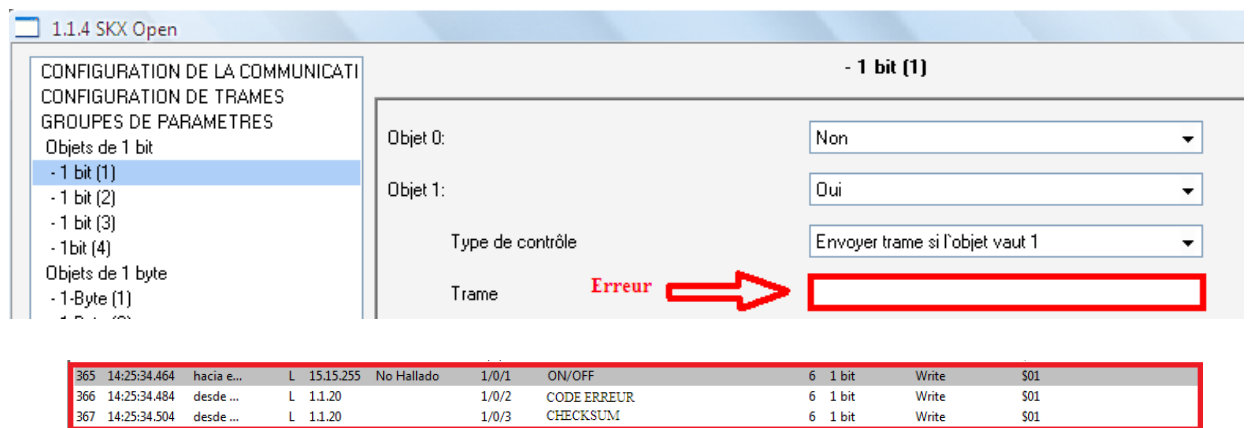


Figure 2.15. Définition de trame vide, qui générera une erreur de checksum

- Utilisation incorrecte de "#".** A chaque fois que ce caractère, qui permet d'ajouter à la trame une valeur variable, sera mal utilisé, le SKX Advance activera l'objet de 1 bit "Erreur: Utilisation incorrecte du '#'" et le code "Code d'erreur".

Exemple: Dans la trame de l'objet de 1bit numéro 0 sont utilisés les caractères ##. Le SKX Advance active l'objet d'erreur associé car il n'est pas possible d'ajouter une partie variable pour ce genre d'objets.

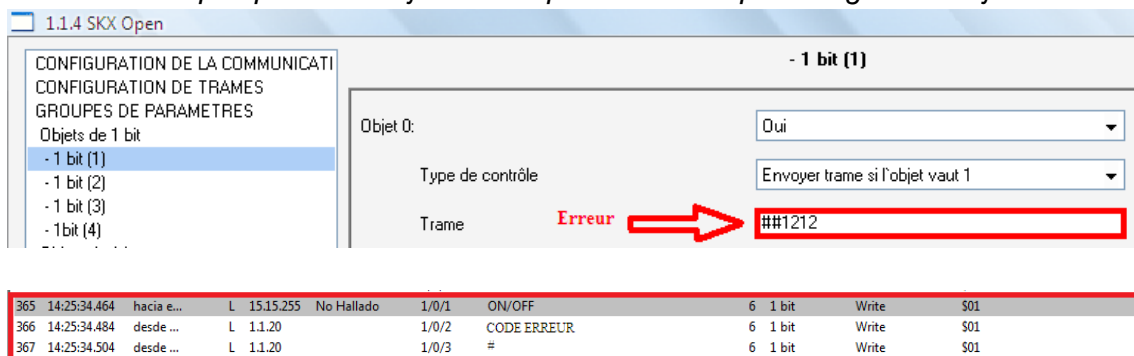


Figure 2.16. Définition de trame avec utilisation incorrecte de '##'

- Trame trop longue.** Dû au fait que les en-têtes, les pieds et sous-trames permettent d'entrer des trames de n bytes, il est possible d'atteindre une trame supérieure à 29 bytes. Si cela arrive, le SKX Advance activera l'objet de 1 bit "Erreur: Trop long" et l'objet "Code d'erreur".

Exemple: S'il est configuré un en-tête de 10 bytes, un pied de 10 bytes et une sous-trame de 4byte, et qu'en plus il est ajouté une trame de 10 bytes, la limite permise est dépassée et le SKX Advance activera l'objet d'erreur correspondant.

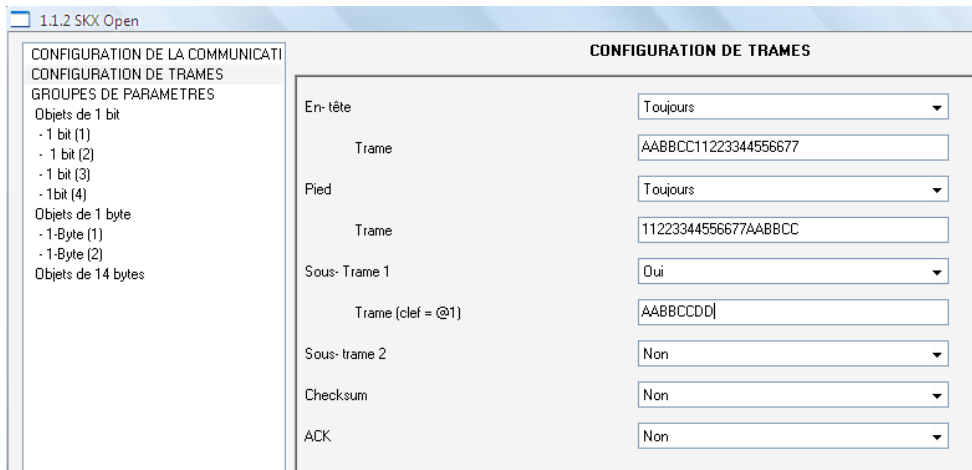


Figure 2.17. Définition des trames d'en-tête, pied et sous-trame 1

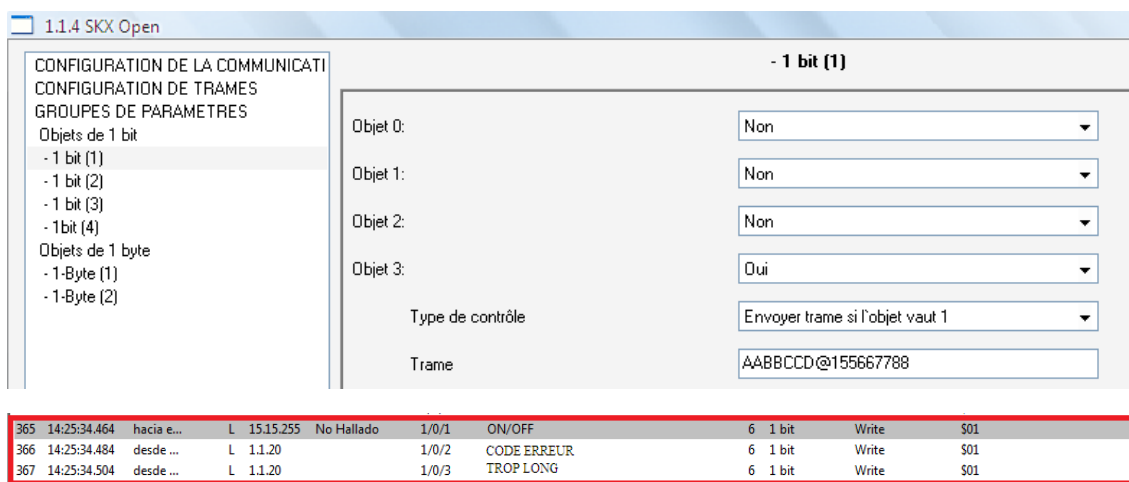


Figure 2.18. Définition de trame trop longue, qui provoquera l'erreur correspondante

- **Erreur dans la réception.** Si la configuration des paramètres de communication série ne coïncide pas avec la configuration des trames reçues (différente vitesse, parité, etc.) le SKX Advance activera l'objet de communication de 1 bit "Erreur: Réception" et l'objet "Code d'erreur".

Exemple: Le dispositif final est configuré avec une vitesse de 9600 bauds et le SKX Advance avec une vitesse de 1200 bauds. Ceci provoquera que le SKX Advance active l'objet d'erreur de réception.

- **Caractère non hexadécimal.** S'il est configuré un caractère non hexadécimal au moment de rentrer certaine trame de communication (comme une lettre minuscule), le SKX Advance activera les objets de communication "Erreur: Non hexadécimal" et "Code d'erreur".

Exemple: Un des caractères utilisés pour définir la trame de l'objet de communication numéro 0 est écrite en minuscule: "ABcD". Ceci provoquera une activation de l'objet d'erreur correspondant.

1.1.4 SKX Open

CONFIGURATION DE LA COMMUNICATI
 CONFIGURATION DE TRAMES
 GROUPE DE PARAMETRES

Objets de 1 bit
 - 1 bit (1)
 Objets de 1 byte
 - 1-Byte (1)
 - 1-Byte (2)

- 1 bit (1)

Objet 0: Oui

Type de contrôle Envoyer trame si l'objet vaut 1

Trame **Erreur** ABcD

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	CODE ERREUR	6	1 bit	Write	\$01
367	14:25:34.504	desde ...	L	1.1.20		1/0/3	NON HEXADECIMALE	6	1 bit	Write	\$01

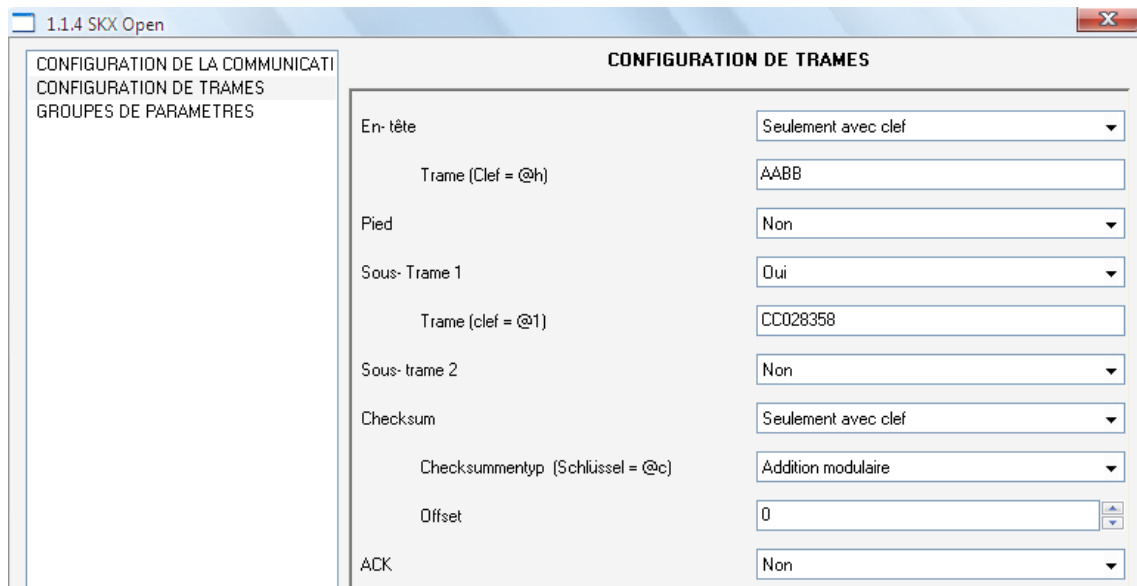
Figure 2.18. Définition de trame avec caractère non hexadécimal

2.5. EXEMPLES DE CONFIGURATION

Pour résumé et clarifier le fonctionnement du SKX Advance, voici des exemples de configurations de trames.

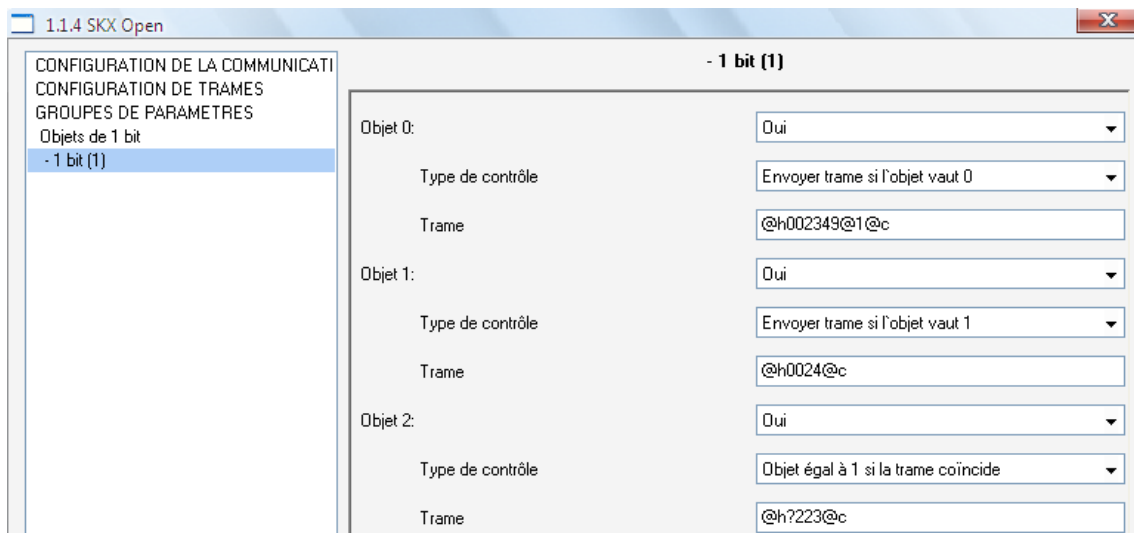
2.5.1. OBJETS DE 1 BIT

Depuis l'onglet "Configuration de trames" dans l'ETS, sont définis, dans les trames spéciales, les valeurs suivantes:



Paramètre	Valeur
En-tête	Seulement avec clef
Trame (Clef = @h)	AABB
Pied	Non
Sous- Trame 1	Oui
Trame (clef = @1)	CC028358
Sous- trame 2	Non
Checksum	Seulement avec clef
Checksummentyp (Schlüssel = @c)	Addition modulaire
Offset	0
ACK	Non

Après ce premier paramétrage, il faut définir les objets de 1 bit qu'il est souhaité utiliser; dans ce cas, il est activé l'objet numéro 0 et numéro 1 pour envoyer les trames au dispositif final, via RS-232, et l'objet numéro 2 pour recevoir des trames arrivant du dispositif final. Le paramétrage de ces objets sera le suivant:



Objet	Statut	Type de contrôle	Trame
Objet 0:	Oui	Envoyer trame si l'objet vaut 0	@h002349@1@c
Objet 1:	Oui	Envoyer trame si l'objet vaut 1	@h0024@c
Objet 2:	Oui	Objet égal à 1 si la trame coïncide	@h?223@c

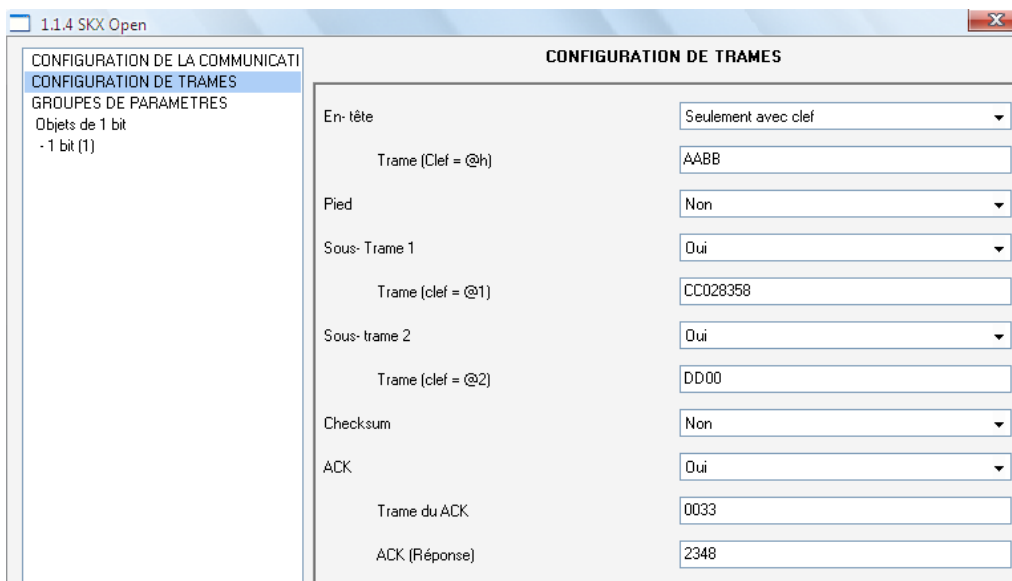
La table suivante montre une séquence de fonctionnement, avec les trames qui sont envoyées et reçues dans chaque cas. (**Note:** les caractères en gras indiquent le checksum calculé avec somme modulaire des caractères qui compose la trame).

EVENEMENT	REACTION
Arrive depuis le BUS KNX un "1" sur l'objet 0	Aucune action n'est réalisée. En effet, l'objet 0 envoie uniquement la trame définie s'il reçoit un "0".
Arrive depuis le BUS KNX un "0" sur l'objet 0	Le SKX envoie la trame "AABB00239CC0283587A", sur le port RS-232*
Arrive depuis le BUS KNX un "1" sur l'objet 1	Le SKX envoie la trame, sur le port série RS-232, vers le dispositif final "AABB002489"
Le SKX reçoit la trame suivante, arrivant de l'appareil final: "0024"	Aucune action n'est réalisée. En effet, la trame ne coïncide pas avec celle paramétrée.
Le SKX reçoit la trame suivante, arrivant de l'appareil final: "AABB38322349"	Aucune action n'est réalisée parce que le checksum reçu n'est pas correct (ce devrait être F2)
Le SKX reçoit la trame suivante, arrivant de l'appareil final: "AABB383223 F2 "	Le SKX écrit un "1" sur l'objet numéro 2
Le SKX reçoit la trame suivante, arrivant de l'appareil final: "AABB856D23 7A "	Le SKX écrit un "1" sur l'objet numéro 2

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

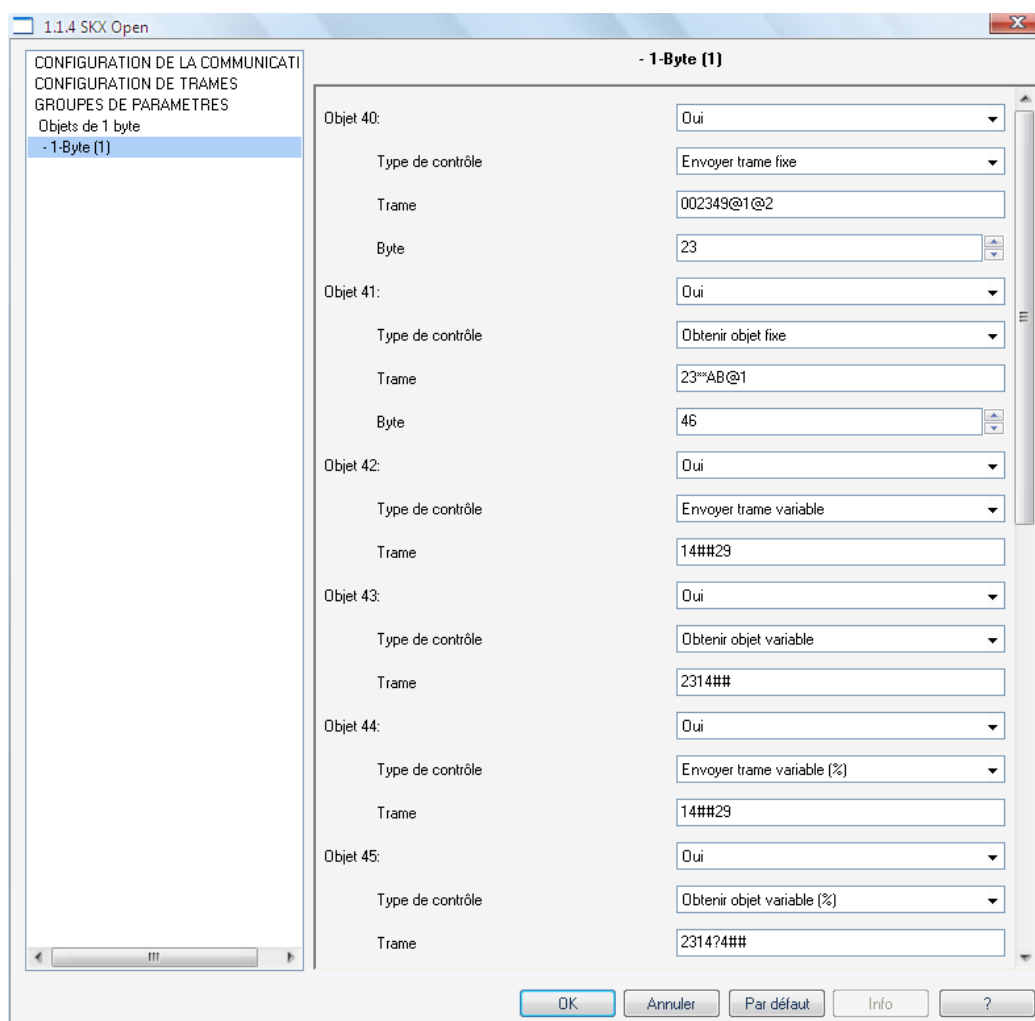
2.5.2. OBJETS DE 1 BYTE

Depuis l'onglet "Configuration de trames" dans l'ETS, sont définis, dans les trames spéciales, les valeurs suivantes:



Paramètre	Valeur
En-tête	Seulement avec clef
Trame (Clef = @h)	AABB
Pied	Non
Sous-Trame 1	Oui
Trame (clef = @1)	CC028358
Sous-trame 2	Oui
Trame (clef = @2)	DD00
Checksum	Non
ACK	Oui
Trame du ACK	0033
ACK (Réponse)	2348

Après ce premier paramétrage, il faut définir les objets de 1 byte qu'il est souhaité utiliser; dans ce cas, sont activés les objets numéro 40, 41, 42, 43, 44, 45 et 46, chacun d'eux pour une fonction différente. Le paramétrage de ces objets sera le suivant:



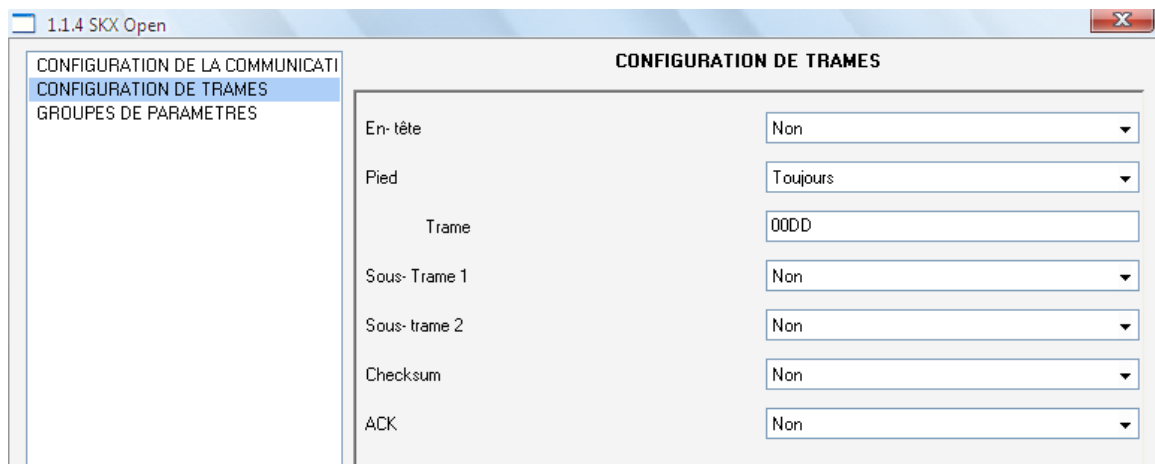
La table suivante montre une séquence de fonctionnement, avec les trames envoyées et reçues dans chaque cas.

EVENEMENT	REACTION
Arrive depuis le BUS KNX un 12 sur l'objet 40	Aucune action n'est réalisée. En effet, l'objet 40 envoie uniquement la trame définie s'il reçoit un "23".
Arrive depuis le BUS KNX un 23 sur l'objet 40	Le SKX envoie la trame, sur le port série RS-232, vers le dispositif final "AABB002349CC028358DD00"
Le SKX reçoit la trame suivante, arrivant de l'appareil final: "3232"	Aucun objet n'envoie de valeur. En effet, la trame ne coïncide avec aucune de celle paramétrée. Le SKX envoie un accusé de réception (ACK) = "0033"
Le SKX reçoit la trame suivante, arrivant de l'appareil final: "2348"	Aucun objet n'envoie de valeur. En effet, la trame ne coïncide avec aucune de celle paramétrée dans les objets. Dans ce cas, aucune trame d'accusé de réception (ACK) ne sera envoyé car "2348" coïncide avec la réponse ACK paramétré.
Le SKX reçoit la trame suivante, arrivant de l'appareil final: "23ABCC028358"	Aucun objet ne répond, parce que, même si la trame reçue coïncide avec celle définie pour l'objet 41, l'envoi de l'en-tête (AABB) est obligatoire Le SKX envoie un accusé de réception (ACK) = "0033"
Le SKX reçoit la trame suivante, arrivant de l'appareil final: "AABB23ABCC028358"	Le SKX écrira un "46" sur l'objet numéro 41. En effet, la trame coïncide avec celle configurée (dans ce cas les ** sont un ensemble vide). Le SKX envoie un accusé de réception ACK ("0033")
Le SKX reçoit la trame suivante, arrivant de l'appareil final: "AABB233333ABCC028358"	Le SKX écrira un "46" sur l'objet numéro 41. En effet, la trame coïncide avec celle configurée (dans ce cas les ** = '3333'). Le SKX envoie un accusé de réception ACK ("0033")
Le SKX reçoit la trame suivante, arrivant de l'appareil final: "AABB23ABABCC028358"	Le SKX n'enverra aucune valeur sur aucun objet, parce que dans ce cas **="AB', mais comme il coïncide avec la partie fixe de la trame après les **, le SKX Advance l'interprète comme un ensemble vide et continu l'analyse de la trame, qui ne coïncide plus avec celle configurée. le SKX n'enverra donc rien, sauf l'accusé de réception.
Arrive depuis le BUS KNX un 16 (décimal) sur l'objet 42.	Le SKX enverra au dispositif final, via la liaison RS-232, la trame: "AABB14 <u>10</u> 29" (il a transformé internement le <u>16</u> reçu par sa valeur hexadécimale: 0x <u>10</u>)
Le SKX reçoit la trame suivante, arrivant de l'appareil final: "AABB2314"	Le SKX n'envoie aucune valeur sur l'objet parce que, même si la trame coïncide avec celle configurée dans l'objet 43, aucun caractère, de la partie variable indiquée par les ## n'a été détecté. Le SKX envoie un accusé de réception.
Le SKX reçoit la trame suivante, arrivant de l'appareil final: "AABB2314 <u>93</u> "	Le SKX écrit dans l'objet numéro 43 la valeur décimale équivalente à celle hexadécimale reçue dans la partie de la trame correspondant à ## (0x <u>93</u> → <u>147</u>). Le SKX écrit 147 dans l'objet 43 et enverra, en plus, un accusé de réception ACK.

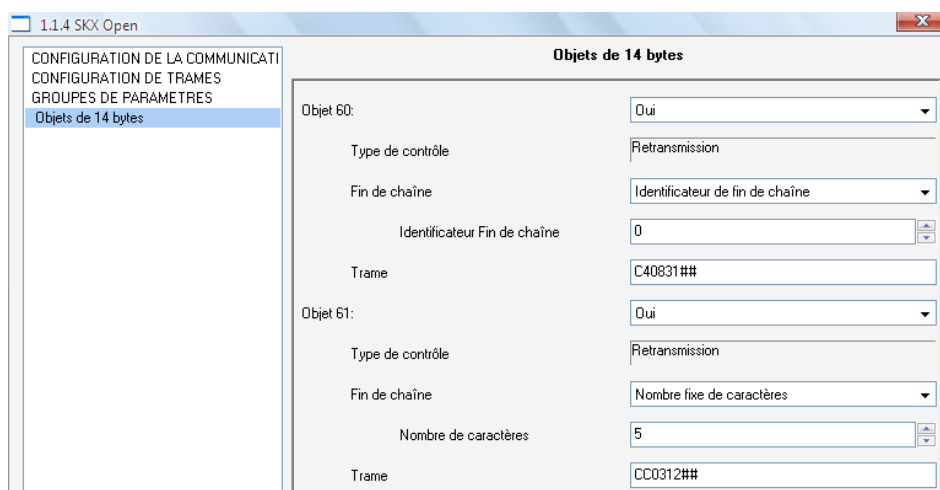
EVENEMENT	REACTION
Le SKX reçoit la trame suivante, arrivant de l'appareil final: "AABB2314 <u>28</u> "	Le SKX écrit dans l'objet numéro 43 la valeur décimale équivalente à celle hexadécimale reçue dans la partie de la trame correspondant à ## (0x <u>28</u> → 40). Le SKX écrit 40 dans l'objet 43 et enverra, en plus, un accusé de réception ACK.
Arrive depuis le BUS KNX un 83 sur l'objet 44	Le SKX envoie la trame, sur le port série RS-232, vers le dispositif final "AABB14 <u>21</u> 29" (la valeur <u>83</u> qui arrive au SKX est une valeur en décimale et fait référence à un pourcentage KNX. Tout d'abord, le SKX le transforme en un pourcentage standard → $83 \cdot 100 / 255 \approx 33\%$; et ensuite en hexadécimal: 33d → 0x <u>21</u>)
Arrive depuis le BUS KNX un 255 sur l'objet 44	Le SKX envoie la trame, sur le port série RS-232, vers le dispositif final "AABB14 <u>64</u> 29" (la valeur <u>255</u> qui arrive au SKX est une valeur en décimale et fait référence à un pourcentage KNX. Tout d'abord, le SKX le transforme en un pourcentage standard → $255 \cdot 100 / 255 = 100\%$; et ensuite en hexadécimal: 100d → 0x <u>64</u>)
Le SKX reçoit la trame suivante, arrivant de l'appareil final: "AABB2314AABBCCDD <u>83</u> "	Le SKX écrit dans l'objet numéro 45 la valeur décimale équivalente à celle où se trouvent les ## → <u>83</u> . Ceci est un pourcentage standard. Le SKX le transformera en pourcentage KNX ($83 \cdot 255 / 100 = 212$) et celui-ci sera la valeur qui sera écrite dans l'objet 45.

2.5.3. OBJETS DE 14 BYTES

Depuis l'onglet "Configuration de trames" dans l'ETS, sont définis, dans les trames spéciales, les valeurs suivantes:



Après ce premier paramétrage, il faut définir les objets de 14 bytes qu'il est souhaité utiliser; dans ce cas, sont activés les objets numéro 60 et 61. Le paramétrage de ces objets sera le suivant:



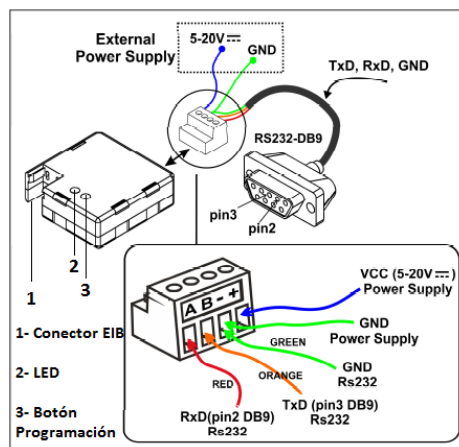
La table suivante montre une séquence de fonctionnement, avec les trames envoyées et reçues dans chaque cas.

EVENEMENT	REACTION
Arrive depuis le BUS KNX la chaîne de caractères "Hola" sur l'objet 60	Le SKX enverra, sur le port série RS-232, la trame "C40831 486F6C61 0000DD", où: Hola = " 486F6C61 " (codification ASCII de chaque caractère) 00 = Identificateur fin de trame 00DD = Pied
Le SKX reçoit la trame suivante, arrivant de l'appareil final: "C40831 4D656E73616A6500 "	Le SKX n'effectue aucune action. En effet, la trame reçue n'inclus pas le pied.
Le SKX reçoit la trame suivante, arrivant de l'appareil final: "C40831 4D656E73616A650000DD "	Le SKX écrira la chaîne de caractères "Mensaje" sur l'objet 60.
Arrive depuis le BUS KNX la chaîne de caractères "Test" sur l'objet 61	Le SKX ne réalise aucune action. En effet, la chaîne n'a pas la longueur fixée par paramètre, qui est de 5 caractères.
Arrive depuis le BUS KNX la chaîne de caractères "Test1" sur l'objet 61	Le SKX envoie, via le port série RS-232, la trame "CC0312 5465737431 00DD", où: Test1 = " 5465737431 " (codification ASCII de chaque caractère) 00DD = Pied
Le SKX reçoit la trame suivante, arrivant de l'appareil final: "CC0312 50727565626100DD "	Le SKX écrira la chaîne de caractères "Prueba" sur l'objet 61.
Le SKX reçoit la trame suivante, arrivant de l'appareil final: "CC0312 5465737400DD "	Le SKX ne réalise aucune action car, étant une trame de longueur fixe (5 caractères), le SKX l'analyse et l'interprète comme la chaîne entrante "5465737400"; il cherche ensuite le pied mais ne le trouve pas (en effet, il reste seulement les caractères DD).

3. RESUME DU PRODUIT



Hardware du dispositif



Connexion SKX - RS-232

SKX: Interface ZENNIO.

- Communication bidirectionnelle entre le BUS KNX et les dispositifs externes via le port série RS-232 (tel que TV, système de surveillance, système audio, etc.)
- Possibilité d'étendre une installation aux dispositifs ne disposant pas de communication KNX mais RS-232.

SKX Advance: Programme d'application.

- Trames dynamiques de longueur variable (jusqu'à **29 Bytes**)
- Objets de communication de plusieurs types
- Détection d'erreurs de communication:
- Plusieurs vitesses de transmission
- **S'adapte** aux différents protocoles de fonctionnement des appareils électroniques.
- Très **polyvalent**.



Devenez membre!

<http://zenniofrance.zendesk.com/portal>

SUPPORT TECHNIQUE