



SKX OPEN

SKX OPEN

ZNRX-SKXOPEN

1.	Introduction	3
1.1.	SKX OPEN.....	3
1.2.	Application OPEN	4
1.3.	Spécifications basiques du SKX OPEN	4
2.	Installation.....	5
2.1.	Installation SKX OPEN, BUS KNX.....	5
2.2.	Connexion du SKX OPEN sur le port RS232.....	5
3.	Paramétrage.....	6
3.1.	Ecran de configuration général	7
3.2.	Groupes de paramètre (parameter groups)	10
3.3.	L'objet d'erreur	12
3.3.1.	Les différentes erreurs	13
3.3.2.	Exemples d'erreurs.....	13

1. INTRODUCTION



1.1.SKX OPEN

Le SKX OPEN est le produit de Zennio qui permet la connexion du BUS KNX avec le BUS de données type RS232.

Caractéristique du dispositif

- Petite taille : 45 x 45 x 14mm
- Dessiné pour être placé, soit dans une boîte de dérivation (ou boîte encastrable), soit dans un tableau électrique.
- Protocole de communication propre.
- Diverses vitesses de communication et mécanisme de correction d'erreurs
- Idéal pour les applications M2M.
- Basé sur le noyau EIB/KNX BIMM112
- Sauvegarde complète des données
- Conforme aux directives CE.

Description des éléments :

- **Prog :** Bouton utilisé pour mettre l'appareil en position « mode de programmation ». S'il est pressonné pendant la connexion de la tension de BUS, l'appareil est forcé en position « mode sûr ».
- **LED :** Indique que l'appareil est en « mode de programmation ». Quand l'appareil entre en « mode sûr », la LED clignote chaque 0.5s.

1.2. APPLICATION OPEN

L'objectif de ce manuel est d'expliquer le fonctionnement du programme d'application spécifique OPEN, développé pour l'interconnexion de dispositif externe avec le BUS KNX, via une liaison RS232 pour tous les appareils acceptant la possibilité de contrôle par cette liaison.

Le SKX OPEN est un produit utilisé pour la communication entre KNX et le protocole série RS232 en toute liberté. En effet, indépendamment du code hexadécimal généré par n'importe quel appareil pour réaliser une action, il sera possible d'associer tel code hexadécimal à un objet de communication et contrôler cet ordre depuis le KNX.

Note : Il est possible d'intégrer n'importe quel code hexadécimal dans la limite de la longueur acceptée par le SKX OPEN. (Voir paragraphe "3-2. Groupes de paramètres")

La communication se fait via le SKX OPEN qui permet d'échanger des informations de manière bidirectionnelle. En effet, il est possible d'envoyer des données, du bus KNX vers le BUS de données RS232 connecté sur votre terminal, ou vice-versa, reconnaître des données envoyées par le terminal (via le protocole RS232) vers le BUS KNX.

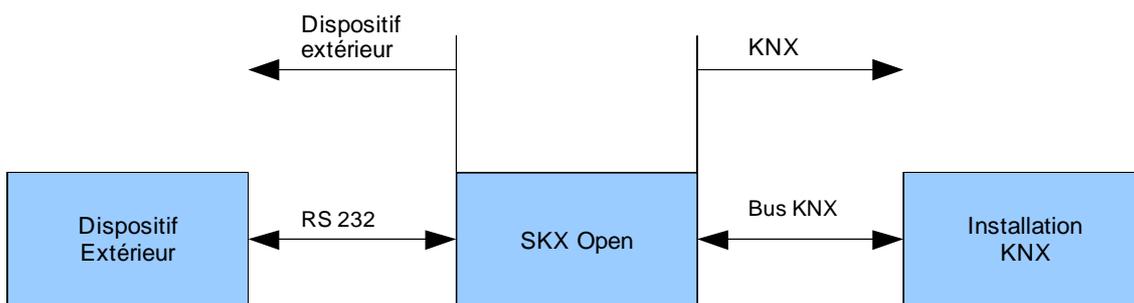


Figure 1. Communication SKX OPEN

1.3. SPECIFICATIONS BASIQUES DU SKX OPEN

Dans ce chapitre nous vous présentons les spécifications basiques du SKX OPEN :

- **Vitesse de transmission** : 1200, 2400, 4800, 9600, 19200
- **Types de distinction des trames** : TimeOut, Byte de fin de trame
- **Numéro d'objets de communication** : 44
- **Identification d'erreurs** : 1 byte – Masque de bits.
- **Longueur maximum du protocole** : 10 bytes / 20 caractères HEX (Byte de fin de trame non inclus).

2. INSTALLATION

2.1. INSTALLATION SKX OPEN, BUS KNX

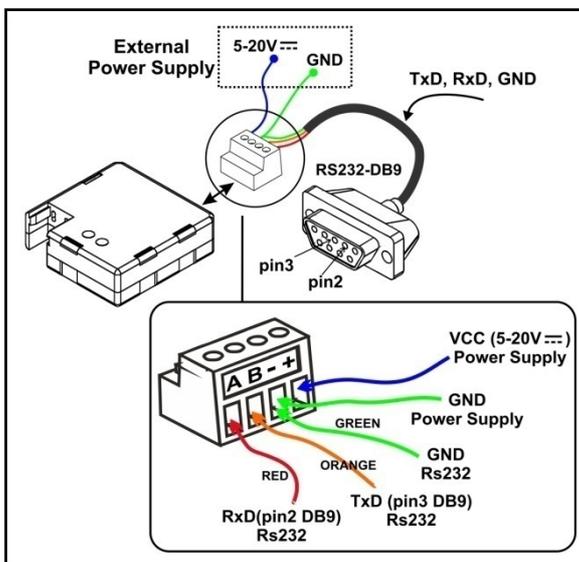
L'installation du SKX OPEN s'effectue de la même manière que n'importe quel dispositif KNX. Pour cela, il suffit de connecter le dispositif au BUS KNX via son connecteur spécifique, après quoi il est prêt à être programmé.

Une fois le dispositif alimenté via le BUS KNX, il est possible de charger l'adresse physique ainsi que le programme d'application spécifique du SKX OPEN.

Ce dispositif ne nécessite pas d'alimentation extérieure, il fonctionne uniquement avec l'alimentation du BUS. Par contre, il faudra alimenter le bus RS232 avec une alimentation indépendante de celle du BUS KNX, tel que le définit le standard.

2.2. CONNEXION DU SKX OPEN SUR LE PORT RS232

La connexion avec le réseau RS232 se réalise via un connecteur spécifique non fixé sur le SKX OPEN, ce qui facilite sa manipulation et son installation. A continuation est spécifiée la connexion entre les deux protocoles :



Connecteur SKX Open	BUS RS232
A	RSA
B	RSB
-	Terre
+	+12V

Figure 2. Connexion du SKX OPEN

3. PARAMETRAGE

Grâce à ce programme d'application, et si le code hexadécimal utilisé pour chaque ordre est connu, il sera possible d'intégrer n'importe quel appareil avec une interface RS232.

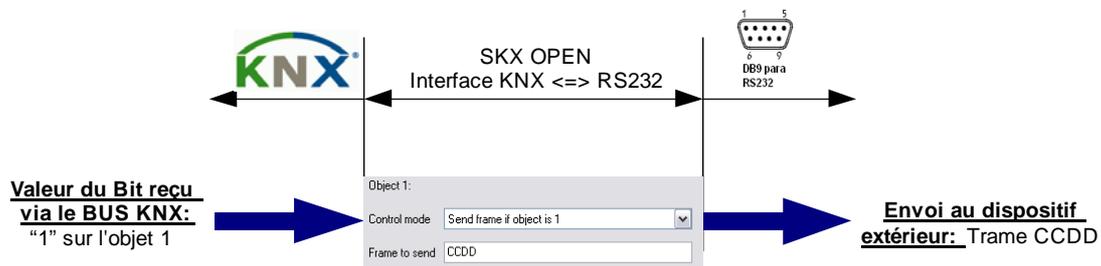
Le SKX OPEN dispose de 48 objets de communication de 1 bit, avec lesquels il sera possible d'interagir entre le protocole KNX et RS232, et d'un objet de contrôle d'erreurs de 1 byte donnant des informations sur les problèmes pouvant survenir durant l'exécution de l'application (valeurs introduites non hexadécimales, minuscules, la trame introduite par l'installateur est impaire, trame reçue par le port série trop longue ou avec une erreurs, l'erreur se trouve sur la dernière trame envoyée ou la trame à comparer avec celle reçue.)

Il existe 4 possibilités de contrôle sur chaque objet, via les paramètres :

- **Communication KNX => RS232 :**

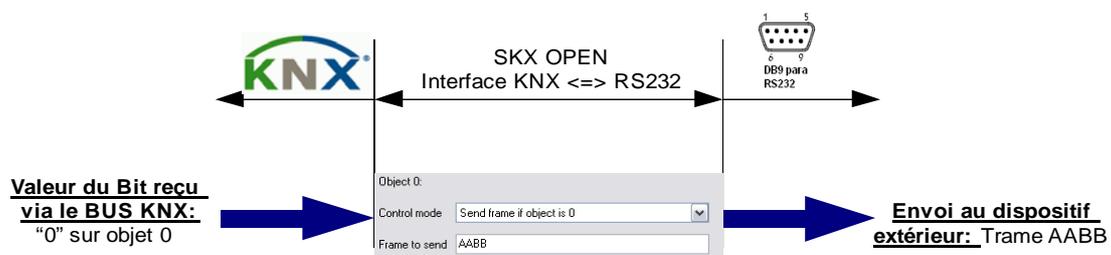
- Envoyer une trame (introduite comme paramètre) vers l'appareil à intégrer, via le port série, après la réception d'un 1 sur l'objet de communication.

Exemple :



- Envoyer une trame (introduite comme paramètre) vers l'appareil à intégrer, via le port série, après la réception d'un 0 sur l'objet de communication.

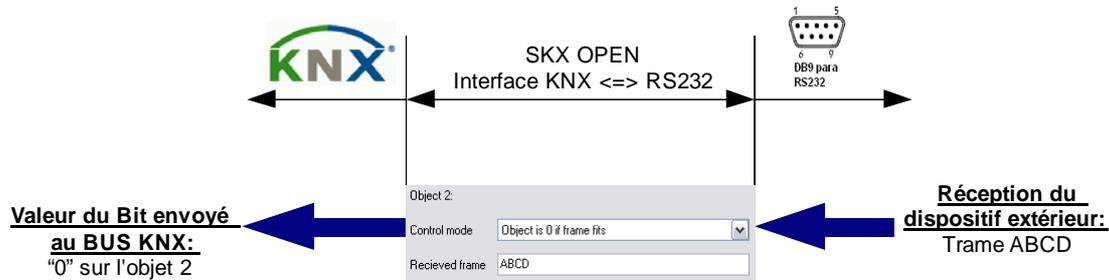
Exemple :



- **Communication RS232 => KNX:**

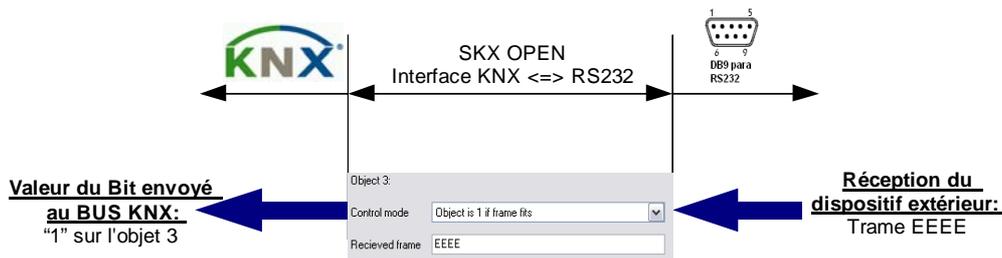
- Envoyer un 0 sur l'objet de communication si la trame reçue via le port RS232 coïncide avec la trame introduite par l'installateur dans les paramètres.

Exemple :



- Envoyer un 1 sur l'objet de communication si la trame reçue via le port RS232 coïncide avec la trame introduite par l'installateur dans les paramètres.

Exemple :



3.1.ECRAN DE CONFIGURATION GENERAL

Sur cette page il faut configurer les différents paramètres de communications du SKX OPEN :

- **Vitesse :** 1200-2400-4800-9600-19200
- **Parité :** Paire ou impaire
- **Temps d'attente entre les trames (en dixièmes de seconde) :** Il s'agit d'un temps d'attente, configurable par paramètre, à laisser entre deux trames à envoyer par le port série. Ce paramètre est utilisé dans le cas où plusieurs objets de communication seraient unis dans une même adresse de groupe, permettant au SKX OPEN d'envoyer ces trames de manière ordonnée, et ainsi d'obtenir une parfaite acquisition et interprétation des données par le récepteur. Ce paramètre dépendra des caractéristiques du récepteur.

- **Réception complète :** Pour la reconnaissance de fin de trame, deux solutions sont possibles :
 - **TimeOut :** C'est le temps pendant lequel est prise en compte la trame entrante, interprétant la réception complète de cette trame
 - **End frame byte :** Il existe aussi la possibilité de marquer la fin de la trame avec un byte spécifique dont l'interprétation est univoque. Dans ce cas, il faudra également déterminer un temps maximum de réception (TimeOut de sécurité) de ce byte de fin de trame afin d'éviter des erreurs de communication. Dans le cas de recevoir une trame dont la longueur dépasse les 10 Bytes, ces données seront ignorées et l'erreur sera indiquée sur l'objet de communication d'erreurs envoyé sur le BUS KNX.

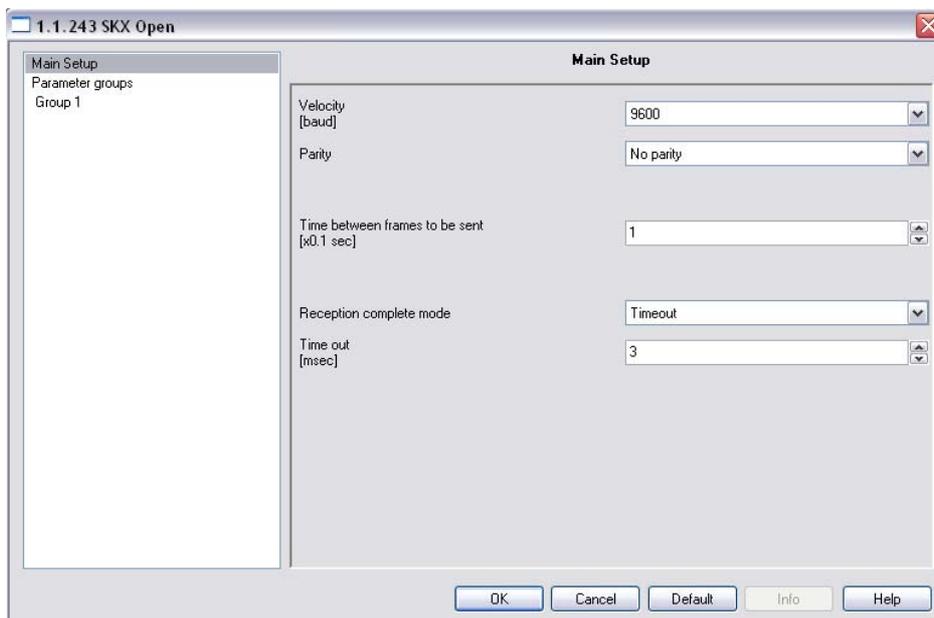


Figure 3. SKX OPEN – Configuration Générale

Exemple :

Premier cas : L'utilisateur définit un temps de "TimeOut" de 30ms. Imaginons que le dispositif extérieur veut envoyer une seconde trame immédiatement après la première (20ms après la première). La figure 4 représente ce fonctionnement

A la fin de la première trame, le TimeOut commence à compter, mais une autre trame arrive avant la fin du temps de TimeOut, ce qui arrête le comptage qui recommence à zéro après la fin de la seconde trame (en réalité il commence à compter après l'envoi de chaque bit mais ce temps est très court donc nous n'en parlons pas). Après cette seconde trame, nous constatons que le TimeOut arrive à sa fin (30ms) donc le SKX OPEN considèrera la trame comme complète. Mais vu que deux trames se sont superposées, le SKX OPEN considèrera la trame comme inconnue et n'enverra rien sur ces objets de communication.

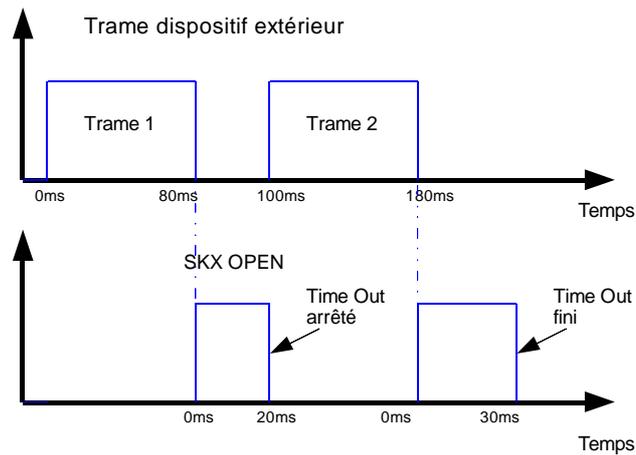


Figure 4. TimeOut trop long

***Second cas :** L'utilisateur définit un TimeOut de 10ms. La Figure 5 représente ce fonctionnement :*

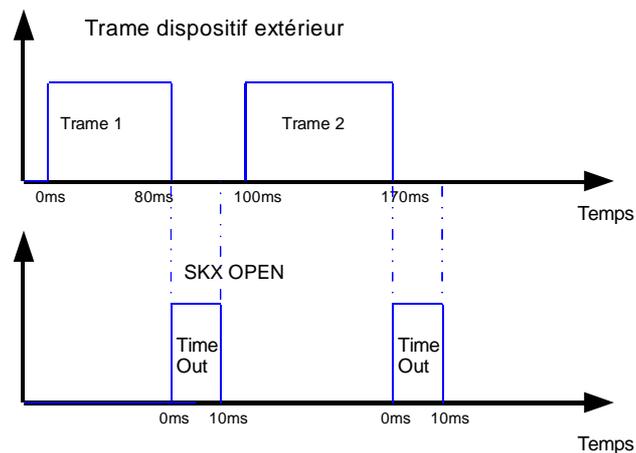


Figure 5. TimeOut bien défini.

Dans ce cas, le temps est bien défini et le SKX OPEN reconnaît les deux trames envoyées.

Le TimeOut devra être défini correctement en fonction des temps d'envoi des trames transmises depuis les appareils extérieurs. Comme nous l'avons vu un temps mal défini peut générer des erreurs dans la réception des messages.

***Exemple :** A continuation est décrit un exemple concret dans lequel il sera nécessaire d'ajuster le TimeOut. Le SKX OPEN (à droite), à réception d'un ordre depuis le SKX OPEN (à gauche) via le port RS232, enverra une trame de confirmation (ACK) et un état (status) avec une séparation de 60ms. Si le TimeOut établi dans le SKX OPEN est supérieur à 60ms, celui-ci ne détectera aucune d'elles, et donc n'actualisera pas l'état.*

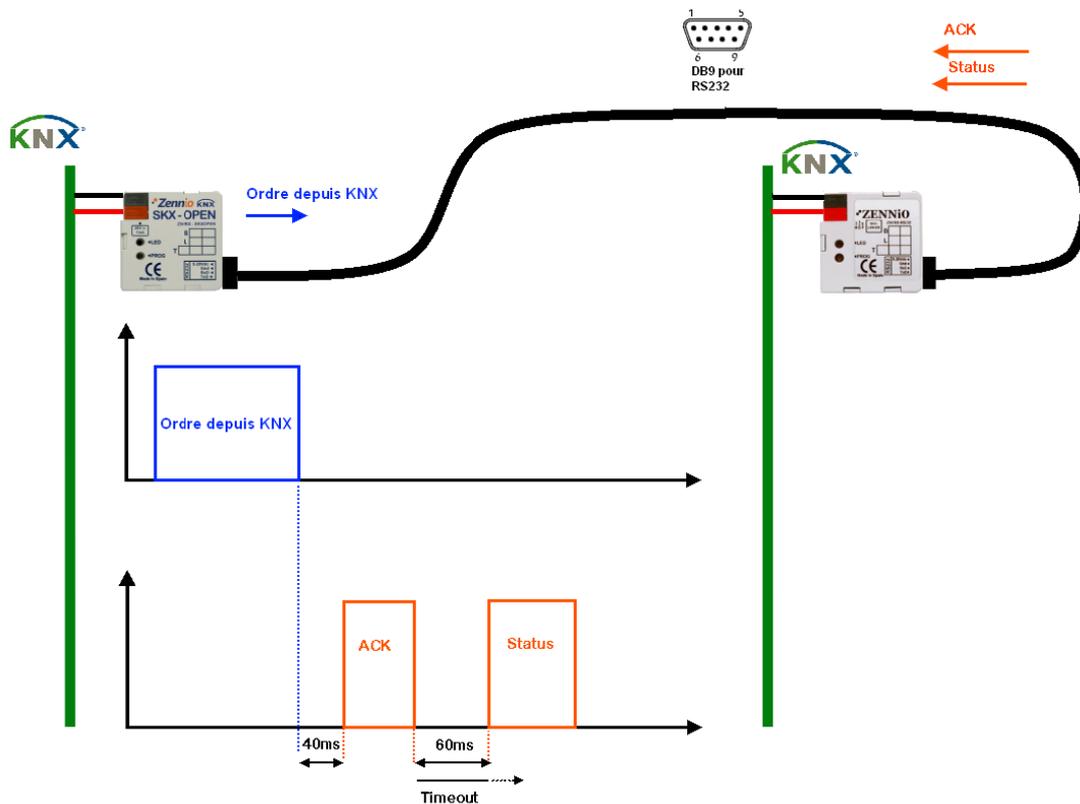


Figure 6.Exemple avec SKX OPEN y SKX OPEN

3.2.GROUPES DE PARAMETRE (PARAMETER GROUPS)

Dans cet onglet, nous devons activer les groupes d'objets de communication que l'on souhaite utiliser. Il y a 4 groupes avec 12 objets de communication de 1 bit chacun.

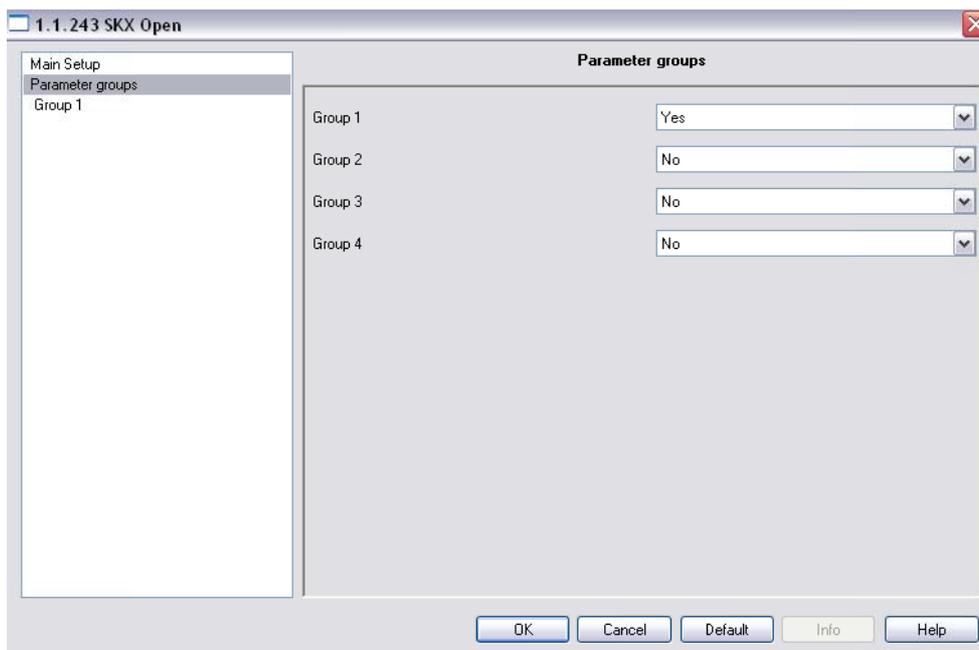


Figure 7. SKX OPEN – Groupes de paramètres

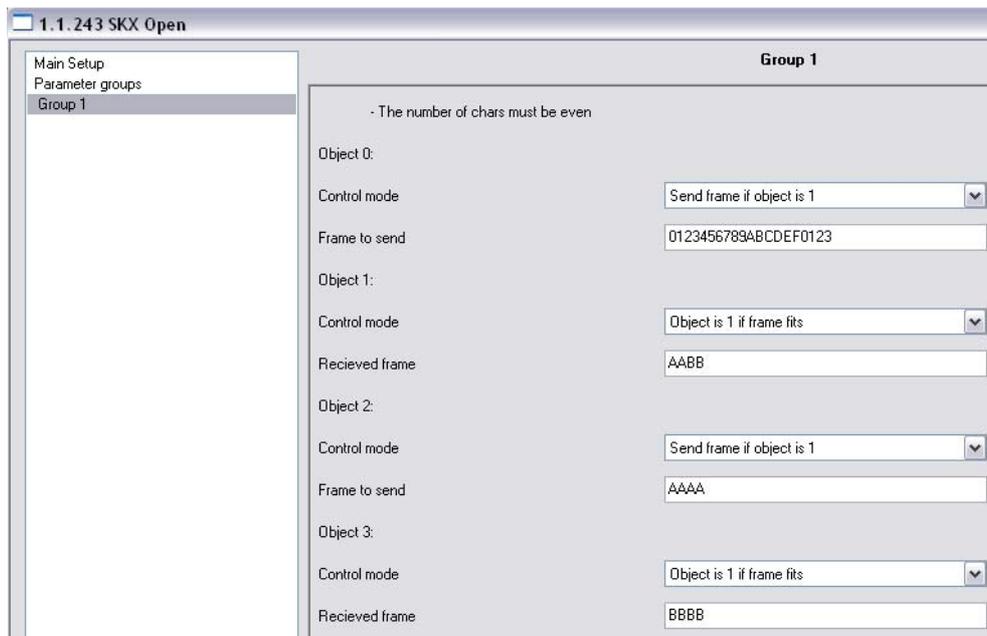


Figure 8. SKX OPEN – Groupe X

Dans les onglets "Group X", il faut sélectionner les types de contrôle et les trames utilisés pour chacun des objets de communication.

- **Obj. X. Control Type (Type de contrôle) :** Ce paramètre permet de définir le type de communication qu'aura l'objet et son action.

○ **Pour la communication KNX => RS232**

- ✓ **Send frame if object is 0 (envoyer la trame si l'objet est 0) :** Envoyer la trame (introduite dans le paramètre "frame to send") vers l'appareil intégré, via le port série, à la réception d'un 0 sur l'objet.
- ✓ **Send frame if object is 1 (envoyer la trame si l'objet est 1) :** Envoyer la trame (introduite dans le paramètre "frame to send") vers l'appareil intégré, via le port série, à la réception d'un 1 sur l'objet.

○ **Pour la communication RS232 => KNX**

- ✓ **Object is 0 if frame fits (Objet à 0 si la trame coïncide) :** Envoyer un 0 sur l'objet dans le cas de recevoir sur le port série une trame qui coïncide avec la trame introduite par l'installateur dans les paramètres.
- ✓ **Object is 1 if frame fits (Objet à 1 si la trame coïncide) :** Envoyer un 1 sur l'objet dans le cas de recevoir, par le port série, une trame qui coïncide avec la trame introduite par l'installateur dans les paramètres.

Selon la communication utilisée KNX => RS232 ou RS232 => KNX, il faut configurer le paramètre correspondant.

- **”Obj. X. Frame to send” ou ”Received frame” (”Trame à envoyer” ou ”Trame à recevoir”)** : Dans ce champs, il faut définir les trames qui devront être prises en compte pendant la communication KNX => RS232 ou RS232 => KNX respectivement. Les trames introduites devront remplir les conditions suivantes :

- Les caractères introduits devront correspondre avec des valeurs hexadécimales (0-9, A-F)
- Les caractères A-F devront être des caractères en majuscule.
- La longueur de la trame devra être paire. Deux caractères pour chaque valeur hexadécimale de 1 byte.

Note I : L’objet de communication ”Error Code” réalise un contrôle des données entrées dans les paramètres. Ce contrôle sera réalisé à l’initialisation du dispositif.

Note II : Une trame hexadécimale de 2 bytes, par exemple, 0x2B 0x7F, devra être introduite dans l’ETS avec le format ”2B7F”.

3.3.L’OBJET D’ERREUR

Number	Name	G...	Length	C	C	R	W	T	U	D.	Priority
48	Error code		1 Byte	C	R	-	T	-			Low
5	Object 5		1 bit	C	-	W	T	-			Low

Figure 9. Objet de communication – Code d’erreur

L’objet d’erreur (1 byte) indiquera, via une accumulation d’erreurs, les problèmes existants. Chaque bit de l’objet a une signification précise.

- **Bit 0 :** Donnée non hexadécimale dans une trame.
- **Bit 1 :** Les données d’une trame sont en minuscules.
- **Bit 2 :** La longueur d’une trame introduite par l’installateur est impaire.
- **Bit 3 :** L’erreur, défini par les bits antérieurement expliqués, se trouve dans la trame en cours d’utilisation.
- **Bit 4 :** Erreur dans la réception du port série : Vitesse de transmission, parité, longueur de la donnée...
- **Bit 5 :** La trame reçue par le port série dépasse les 10 bytes.

3.3.1. LES DIFFERENTES ERREURS

L'objet d'erreur sera actualisé et transmis au BUS KNX dans les cas suivant :

Définition de l'erreur	N° de l'erreur (Après le téléchargement ou si l'erreur ne se trouve pas dans la trame en cours)	N° de l'erreur si elle se trouve dans la trame en cours.
Erreur: Objet associé à une trame vide.	NON	09h (bit0 y bit3)
Erreur: Byte fin de trame non envoyée	NON	28h (bit5 y bit3)
Erreur: Paramètres de communication	18h (bit4 y bit3)	18h (bit4 y bit3)
Erreur: Valeur non hexadécimale	01h (bit0)	09h (Bit0 y bit3)
Erreur: Valeur impaire	04h (bit2)	0Ch (bit2 y bit3)
Erreur: Caractère minuscule	02h (bit1)	0Ah (bit1 et bit3)
Erreur: Longueur de trame	NON	28h (Bit5 y Bit3)

3.3.2. EXEMPLES D'ERREURS

- **Objet associé à une trame vide :** Si, par inadvertance, il est activé un objet et aucune trame n'est entrée dans le champ prévu à cet effet, apparaîtra l'erreur 09h.

Exemple : Dans ce cas, quand l'objet 0 passe à 0, l'erreur de la trame vide sera indiquée dans l'objet d'erreur

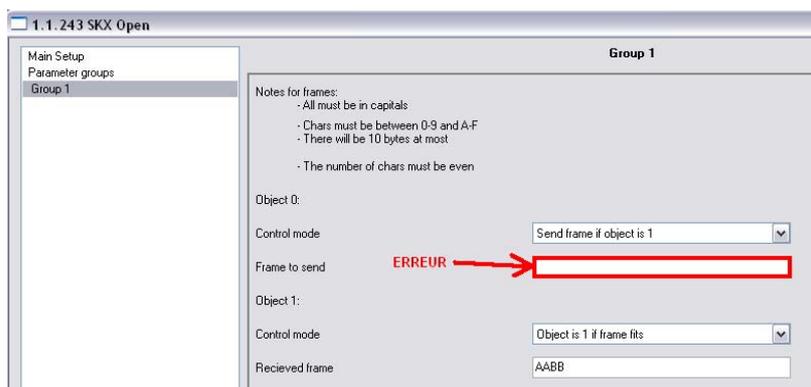


Figure 10. Trame à envoyée vide

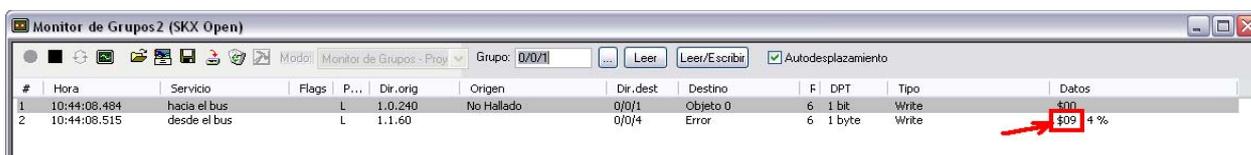


Figure 11. Objet d'erreur avec trame vide

- **Byte fin de trame non envoyé :** Si la méthode de détection de fin de trame est un caractère de fin de trame et que celui-ci n'est pas été envoyé avant la fin du temps maximum (TimeOut), ou qu'il existe une erreur dans la valeur reçue, et donc ne détecte pas le caractère de fin de trame, apparaîtra l'erreur 28h.

Exemple : Si l'on définit un byte de fin de trame FF, la trame est considérée comme bonne uniquement après la réception du byte FF. Si l'appareil externe envoie la trame : 02ADFA ou la trame 02ADFF, mais avec un temps de transmission de plus de 60ms (dans notre exemple), le SKX OPEN signalera l'erreur avec le code 28.

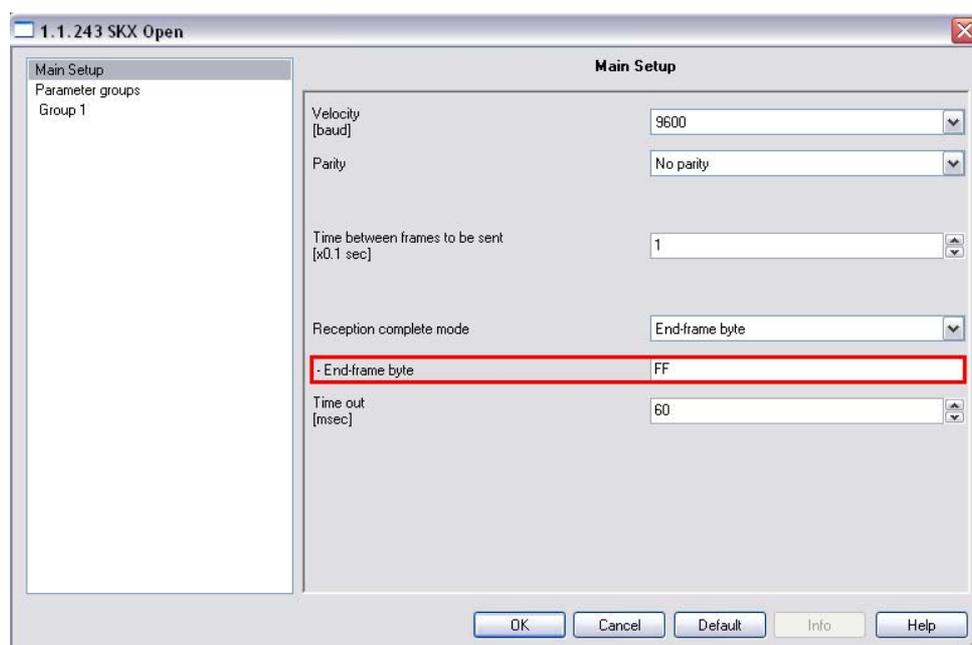


Figure 12 : Fin de trame

#	Time	Service	Flags	Prio	Src.addr	S...	Dest.addr	Destination	Rout	DPT	Type	Data
11...	12:16:55.950	from bus		S	1.1.2		15.15.255	Not Found	6	-	MemoryResponse (S=10)	# bytes: 1 address: B6 E
11...	12:16:55.979	to bus		S	15.15.255	N...	1.1.2		6	-	ACK (S=10)	
11...	12:16:56.043	to bus		S	15.15.255	N...	1.1.2		6	-	T_Disconnect	
11...	12:17:20.749	from bus		L	1.1.2		0/0/3	Error projector	6	1 byte	Write	528 16 %

Figure 13 : Objet d'erreur avec byte de fin de trame non reçu

- **Erreur dans les paramètres de communication :** Si la configuration des paramètres de communication série ne coïncide pas avec la configuration des trames reçues (vitesse différente, parité, bits de stop...).

***Exemple :** L'appareil externe à une vitesse de transmission de 9600 Bauds et le SKX OPEN de 1200 Bauds. Toutes les erreurs faisant référence aux paramètres de communication sont signalées sous l'erreur numéro 18.*

- **Valeur non hexadécimale :** Si l'installateur entre une valeur non hexadécimale dans les paramètres du SKX OPEN.

***Exemple :** La valeur entrée dans l'objet est 4D5Gh. Dans ce cas, après le téléchargement, l'objet d'erreur indiquera l'erreur numéro 01.*

Object 0:

Control mode

Frame to send

Figure 14 : Valeur entrée non hexadécimale.

- **Valeur impaire :** Si l'installateur entre un nombre impair de caractère dans les paramètres du SKX OPEN.

***Exemple :** Dans ce cas, après le téléchargement, l'objet d'erreur indiquera l'erreur numéro 04h.*

Object 0:

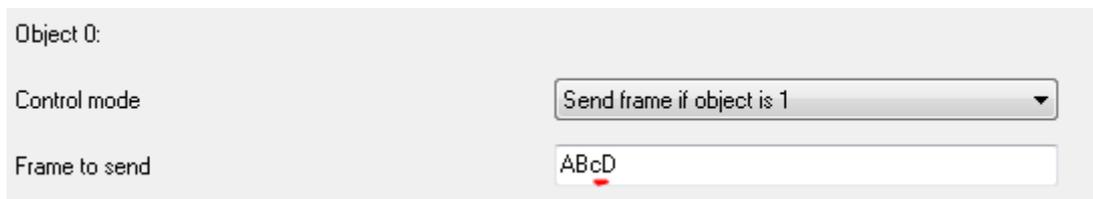
Control mode

Frame to send

Figure 15 : Valeur entrée impaire

- **Caractère minuscule :** Si l'installateur entre un caractère minuscule dans les paramètres du SKX OPEN.

Exemple : Dans ce cas, après le téléchargement, l'objet d'erreur indiquera l'erreur numéro 02.



Object 0:

Control mode

Frame to send

Figure 16. Caractère en minuscule

- **Longueur de la trame :** Si la longueur de la trame reçue est supérieurs à 10 Bytes.

Exemple : Si l'appareil extérieur envoi la trame "0123456789ABCDEF012345", de 12 bytes, l'objet d'erreur indiquera l'erreur numéro 28.

- **Erreur dans la trame en cours :** Si lors de l'envoi d'une trame de la table des paramètres, celle-ci contient une erreur (longueur impaire, valeur non hexadécimale...), cette dernière est notifiée et aucune trame n'est envoyée.



Devenez Membre!

<http://zenniofrance.zendesk.com>

SUPPORT TECHNIQUE