



SKX OPEN

SKX Open

ZN1RX – RS232

INHALT

1.	Einleitung.....	3
1.1.	ZN1RX-SKX OPEN.....	3
1.2.	Applikationsprogramm: SKX Open	3
1.3.	Grundlegende Spezifikationen des SKX Open.....	4
2.	Installation.....	5
2.1.	Installation des SKX OPEN KNX-BUSSYSTEMS.....	5
2.2.	Anschluss des SKX OPEN an die RS 232 Schnittstelle.....	5
3.	Parametrisierung	6
3.1.	Allgemeines Konfigurationsfenster	7
3.2.	Parametergruppen.....	10
3.3.	Die Fehlerobjekte.....	12
3.3.1.	Fehlerbeispiele.....	13

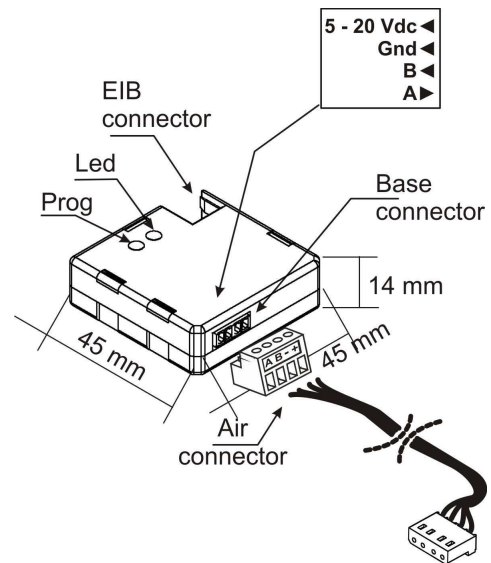
1. EINLEITUNG

1.1. ZN1RX-SKX OPEN

Der ZN1RX-SKX OPEN ist das Produkt von Zennio welches den Datenaustausch zwischen einem KNX-Bus und einem RS 232 Datenbus ermöglicht.

Geräteeigenschaften

- Reduzierte Abmessungen: 45 x 45 x 14mm.
- Geeignet zum Einbau in Abzweigdosen, Gerätedosen, oder in Schaltschränken.
- Proprietäres Kommunikationsprotokoll
- Mehrere Geschwindigkeiten und Fehlerkorrektionsmechanismen.
- Ideal für M2M-Applikationen
- Basiert auf einem B1MM112-Modul
- Kompletter Datenerhalt.
- Erfüllt CE Standard.



Beschreibung der Elemente

- **Prog:** Durch das Betätigen der Programmier Taste wird das Gerät in den Programmiermodus gebracht. Bei Erstbetätigung, nach Anlegen der Busspannung, geht das Gerät in den "Sicherheitsmodus".
- **LED:** Lichtsignal, zeigt an dass sich das Gerät im Programmiermodus befindet. Befindet sich das Gerät im Sicherheitsmodus, blinkt sie in einem Intervall von 0,5 Sek.

1.2. APPLIKATIONSPROGRAMM: SKX OPEN

Der Zweck dieser Bedienungsanleitung ist es das speziell entwickelte Applikationsprogramm zur Integration von Geräten die über eine RS 232 Schnittstelle gesteuert werden können, zu erklären.

Der SKX Open ist ein Produkt, welches zur Kommunikation zwischen einem KNX-System und einem RS 232 Protokoll dient. Unabhängig vom verwendeten Hexadezimalcode den ein Gerät mit einer Aktion assoziiert, kann dieser mit einem Kommunikationsobjekt verknüpft werden um Befehle von KNX-Geräten zu empfangen oder an diese zu senden.

Hinweis: Es können beliebige Hexadezimalcodes integriert werden, so lange sie sich innerhalb der vom SKX Open bestimmten Einschränkungen in Bezug auf die Länge dieser Codes befinden.. (Siehe Abschnitt “3-2. Parametergruppen”).

Die Kommunikation wird über den SKX Open ausgeführt, dadurch wird ein bidirektionaler Datenaustausch ermöglicht. Das heisst es können Daten von KNX-Geräten zum angeschlossenen Datenbus, und umgekehrt von diesem an die KNX-Geräte gesendet werden.

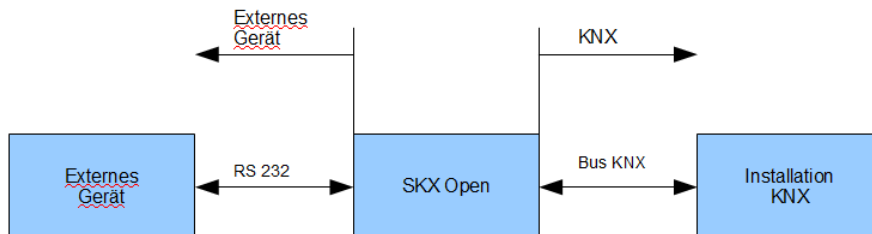


Bild 1. Kommunikation SKX Open

1.3. GRUNDLEGENDE SPEZIFIKATIONEN DES SKX OPEN

In diesem Abschnitt werden die grundlegenden Spezifikationen des SKX OPEN beschrieben:

- Übertragungsgeschwindigkeit (1200, 2400, 4800, 9600, 19200 Baud)
- Erkennung Übertragungsende der Frames (Timeout, End-Byte)
- Anzahl der Kommunikationsobjekte: 48
- Fehlererkennung: 1-bit Fehlerobjekte
- Maximale Länge des Protokolls: 10 Byte / 20 HEX-Zeichen (ohne End-Byte Datenrahmen, falls vorhanden)

WARNUNG

Es sollte mit besonderer Vorsicht bei der Integration eines beliebigen Gerätes in ein KNX-System vorgegangen werden, da dadurch eine Möglichkeit der externen Manipulation des jeweils anderen Systems geschaffen wird.

2. INSTALLATION

2.1. INSTALLATION DES SKX OPEN AN DAS KNX-BUSSYSTEM

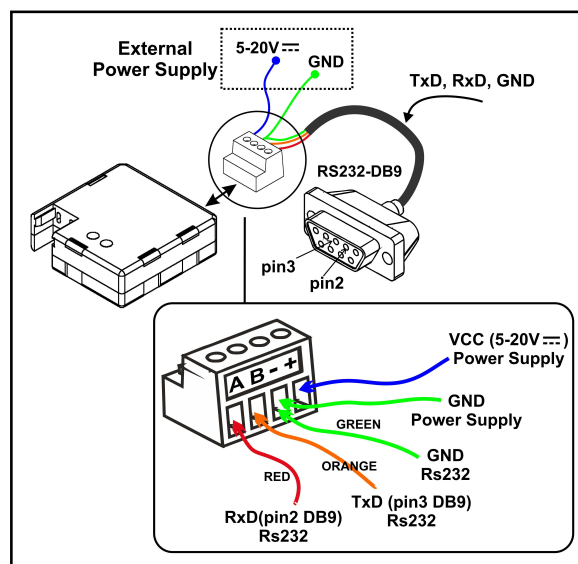
Der SKX Open wird wie jedes beliebige andere KNX-Gerät an den Bus angeschlossen. Hierzu muss das Gerät nur mit Hilfe einer Standardbusklemme mit dem KNX-Bus verbunden werden, und kann dann programmiert werden.

Ist das Gerät einmal mit der Busspannung versorgt, so kann die physikalische Adresse vergeben werden, sowie das Applikationsprogramm SKX Open aufgespielt werden.

Dieses Gerät benötigt keine externe Spannungsversorgung, es funktioniert ausschliesslich mit der KNX-Busspannung. Es ist jedoch notwendig den RS 232 Datenbus mit einer eigenen, dem Standard entsprechenden Spannung zu versorgen.

2.2. ANSCHLUSS DES SKX OPEN AN DIE RS 232 SCHNITTSTELLE

Die Verbindung mit dem RS 232 Datenbus wird über eine spezifische Steckklemme realisiert, wodurch die Installation erleichtert wird. Nachfolgend wird die Verbindung der beiden Systeme beschrieben.



Installation des SKX Linker	BUS RS232
A	RSA
B	RSB
-	Masse
+	+12V

Bild 2. Anschluss des SKX Open

3. PARAMETRISIERUNG

Das für das Applikationsprogramm SKX Open verwendete Gerät, ist das existente Zennio-Produkt ZN1RX-RS232, oder SKX Open. Dank dieses Applikationsprogramms ist es möglich jedes beliebige Gerät mit RS232-Schnittstelle zu integrieren, unter der Voraussetzung dass die mit jeder Aktion verknüpften Hexadezimalcodes bekannt sind.

Der SKX Open verfügt über 48 1-bit Kommunikationsobjekte über die eine Kommunikation zwischen KNX- und RS 232-Protokoll durchgeführt werden kann. Ausserdem stehen mehrere 1-bit Objekte zur Fehlererkennung zur Verfügung (eingegebener Wert nicht hexadezimal, Kleinbuchstaben, eingegebene Länge ist ungerade, am RS 232 Port empfangener Frame ist zu lang oder fehlerhaft, der aufgetretene Fehler befindet sich im letzten gesendeten Frame oder im mit dem empfangenen zu vergleichenden Frame).

Es stehen vier verschiedene Kontrollmöglichkeiten über Parameter zur Verfügung:

Kommunikation KNX => RS232. 1-bit Befehle.

- Senden eines Frames (als Parameter eingegeben) an das über den seriellen Port integrierte Gerät **bei Empfang einer 1** auf dem Kommunikationsobjekt.

Beispiel:



- Senden eines Frames (als Parameter eingegeben) an das über den seriellen Port integrierte Gerät **bei Empfang einer 0** auf dem Kommunikationsobjekt.

Beispiel:



Kommunikation RS232 => KNX . 1-bit Befehle.

- Senden einer 0 über das Kommunikationsobjekt bei **Empfang eines Frames über den seriellen Port**, der mit dem als Parameter angegebenen Frame übereinstimmt.

Beispiel:



- Senden einer 1 über das Kommunikationsobjekt bei **Empfang eines Frames über den seriellen Port**, der mit dem als Parameter angegebenen Frame übereinstimmt.

Beispiel:



3.1. ALLGEMEINES KONFIGURATIONSFENSTER

Dieses Fenster dient zur Konfiguration verschiedener Parameter die die Kommunikation als solches betreffen.

- Geschwindigkeit:** 1200-2400-4800-9600-19200
- Parität:** Gerade oder Ungerade
- Zeit zwischen zu sendenden Frames (in Zehntelsekunden):** Hierbei handelt es sich um eine konfigurierbare Sendepause zwischen den einzelnen Frames. Dieser Parameter ermöglicht dem SKX Open, für den Fall dass mehrere Kommunikationsobjekte mit einer Gruppenadresse verknüpft sind, die den Objekten zugeordneten Frames so zu senden, dass sie vom Empfänger perfekt verarbeitet und interpretiert werden können. Dieser Parameter ist abhängig von den Charakteristiken des Empfängers.

● **Empfang abgeschlossen:** Zur Erkennung des Frame-Endes stehen zwei Möglichkeiten zur Verfügung:

- **Timeout:** Dies ist die Zeit nach deren Ablauf ohne Empfang eines bits der Frame als gültig angesehen wird.
- **End-Byte Datenrahmen :** es besteht die Möglichkeit das Ende eines Frames mit einem eindeutigen, spezifischen Byte zu kennzeichnen. Diese Option stellt zusätzlich ein Timeout als Sicherheit zur Vermeidung von Kommunikationsfehlern zur Verfügung. Im Falle des Empfanges eines Frames mit einer Länge von mehr als 10 Byte, werden diese Daten ignoriert und über ein Kommunikationsobjekt ein Fehlertelegramm auf den Bus gesendet.

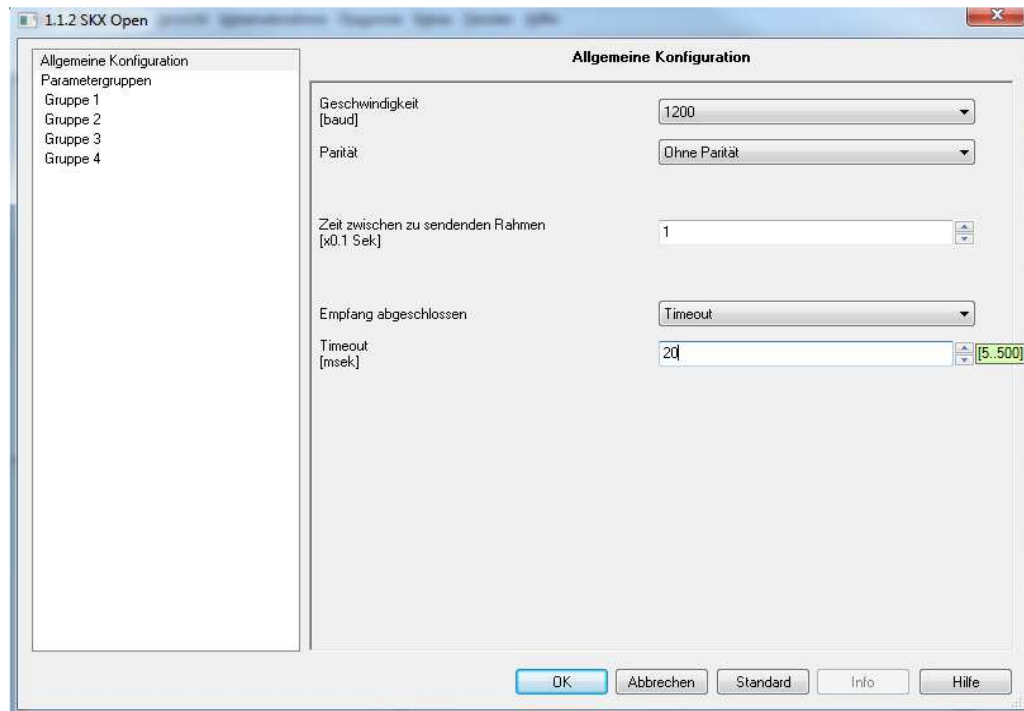


Bild 3. SKX Open – Allgemeine Konfiguration

Beispiel: Ein externes Gerät benötigt 80ms zum Senden des kompletten Frames.

Erster Fall: Der Benutzer definiert ein "Timeout" von 30 ms. Es wird angenommen, dass das externe Gerät einen zweiten Frame direkt nach dem ersten senden will. Diese Funktionsweise wird im folgenden Diagramm dargestellt:

Nach Ende des ersten Frames beginnt der Timeout, der nächste Frame startet bevor der Timeout abgeschlossen ist, d.h. der Timeout wird abgebrochen und zählt wieder von vorne nach Ende des zweiten Frames. Da der Timeout in diesem Fall abgeschlossen wurde (30 ms), betrachtet der SKX Open den Frame als beendet. Da allerdings zwei Frames gesendet wurden, bevor der Empfang als abgeschlossen betrachtet wurde, interpretiert der SKX OPEN den Frame als unbekannt und sendet nichts.

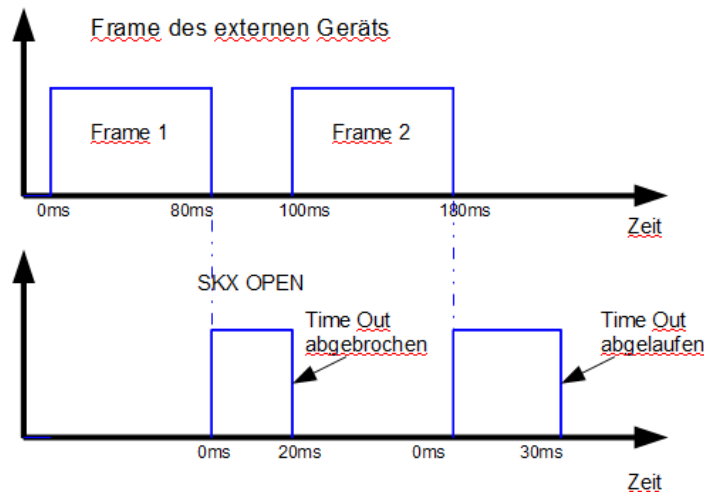


Bild 4. Timeout zu lang

Zweiter Fall: Der Benutzer definiert ein "Timeout" von 10 ms Diese Funktionsweise wird im folgenden Diagramm dargestellt:

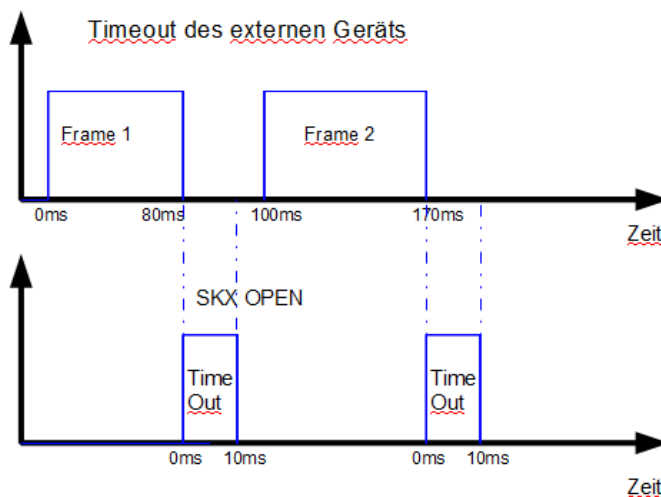


Bild 5. Timeout korrekt definiert

In diesem Fall ist der Timeout korrekt definiert, und der SKX OPEN erkennt die zwei gesendeten Frames.

Bitte beachten: Um den Timeout korrekt zu definieren, muss dies unter Berücksichtigung der Pause zwischen gesendeten Frames des externen Geräts geschehen. Wie im Beispiel gezeigt wird, kann ein schlecht definierter Timeout (zu lang) zu Kommunikationsproblemen führen.

Beispiel: Nachfolgend wird ein praktisches Beispiel aufgezeigt, bei dem es nötig ist den Timeout einzustellen. Der SKX Linker (rechts) sendet bei Empfang eines Befehls des SKX Open (links) am RS232-Port, ein Bestätigungstelegramm (ACK) und den Status in einem Abstand von 60 ms. Ist der Timeout im SKX Open grösser als 60 ms gewählt, so wird dieser keines der Telegramme mitbekommen, und so seinen Status nicht aktualisieren.

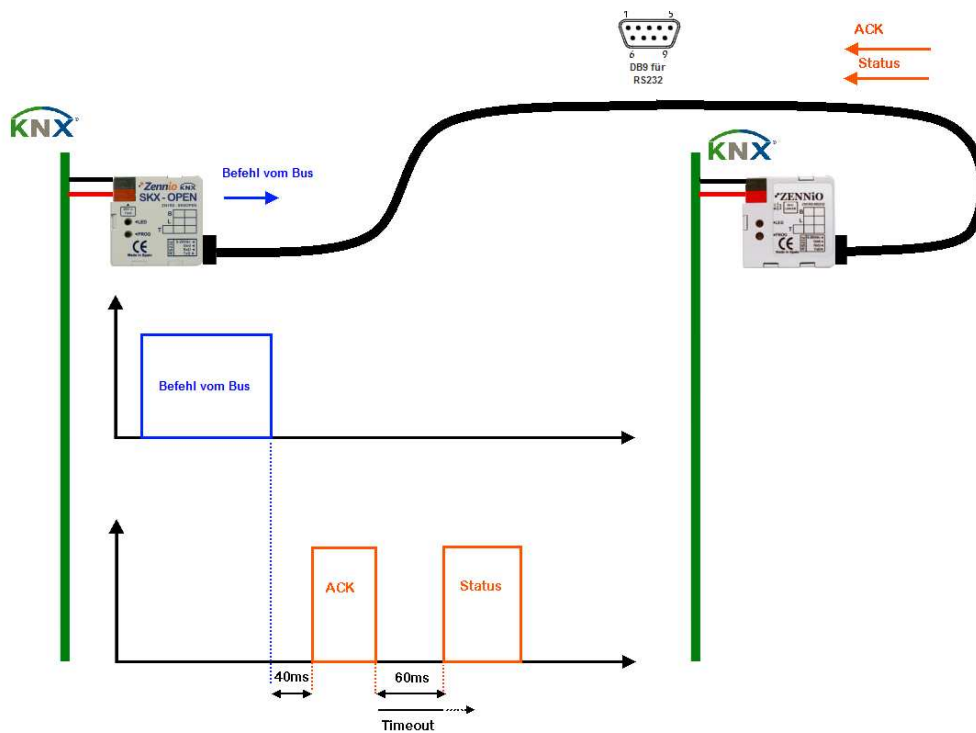


Bild 6. Beispiel eines SKX Open mit einem SKX Linker

3.2. PARAMETERGRUPPEN

In diesem Reiter die Gruppen der Kommunikationsobjekte die benutzt werden sollen, freigegeben werden. Es stehen 4 Gruppen mit je 12 1-bit Kommunikationsobjekten zur Verfügung.

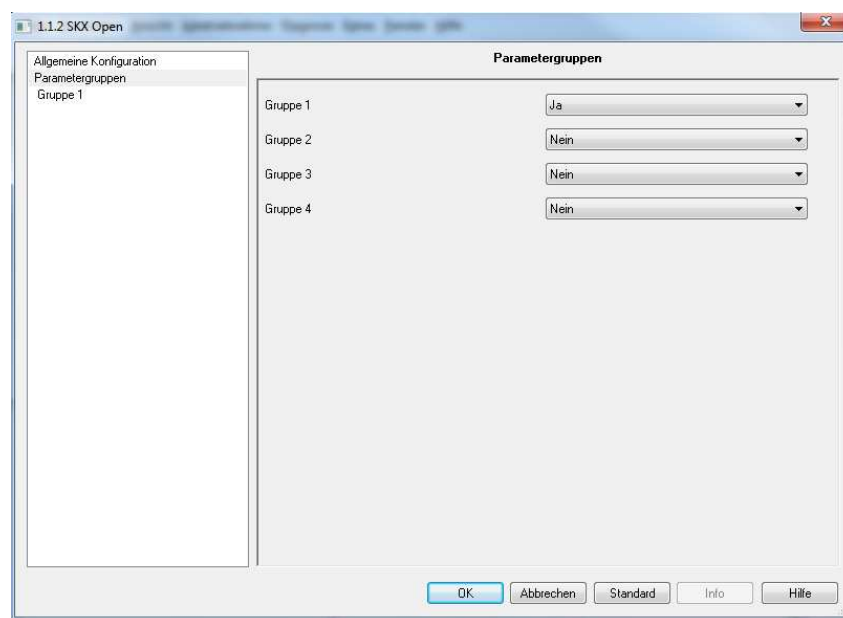


Bild 7. SKX Open – Parametergruppen

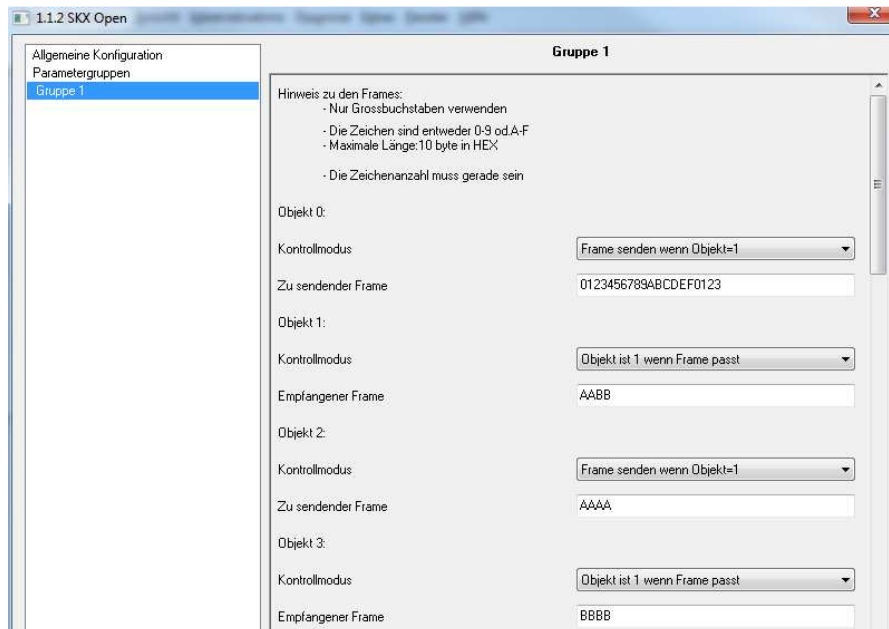


Bild 8. SKX Open – Gruppe X

Über den Reiter Gruppe X (einer je aktiver Gruppe) werden die Kontrollmodi und die benutzten Frames für die einzelnen Kommunikationsobjekte festgelegt.

- **Obj X. Kontrollmodus:** Dieser Parameter definiert welche Art der Kommunikation durch das Objekt generiert wird, und welcher Effekt ausgelöst wird.


Für die Kommunikation KNX => RS232.

- **Frame senden wenn Objekt = 0** Senden eines Frames (als Parameter in "Obj.X, Frame" eingegeben) an das über den seriellen Port integrierte Gerät bei Empfang einer 0 auf dem Kommunikationsobjekt.
- **Frame senden wenn Objekt = 1** Senden eines Frames (als Parameter in "Obj.X, Frame" eingegeben) an das über den seriellen Port integrierte Gerät bei Empfang einer 1 auf dem Kommunikationsobjekt.

Für die Kommunikation RS232 => KNX

- **Objekt ist 0 wenn Frame passt:** Senden einer 0 über das Kommunikationsobjekt bei Empfang eines Frames über den seriellen Port, der mit dem als Parameter angegebenen Frame übereinstimmt.
- **Objekt ist 1 wenn Frame passt:** Senden einer 1 über das Kommunikationsobjekt bei Empfang eines Frames über den seriellen Port, der mit dem als Parameter angegebenen Frame übereinstimmt.

Je nach gewählter Kommunikation KNX => RS232 oder RS232 => KNX, muss folgender Parameter konfiguriert werden:

 **Obj X. Zu sender Frame oder Obj X. empfangener Frame:** Mit diesem Parameter werden die Frames definiert, die für die Kommunikation, KNX => RS232, bzw. RS232 => KNX, in Betracht kommen. Die eingegebenen Frames müssen folgende Voraussetzungen erfüllen:

- Die benutzten Zeichen müssen hexadezimaler Natur sein (0-9, A-F).
- Die Zeichen A-F müssen Grossbuchstaben sein.
- Die Länge des Frames muss eine gerade Parität aufweisen. Zwei Zeichen pro 1-Byte Hexadezimalwert.
- Die maximale Länge beträgt 10 Byte in Hexadezimal.

***Hinweis I:** Es findet eine Überprüfung der eingegebenen Parameter über die Fehler- Kommunikationsobjekte statt. Diese Überprüfung wird bei Initialisierung des Gerätes durchgeführt.*

***Hinweis II:** Es wird darauf hingewiesen dass ein Hexadezimalcode von 2 byte, z.B. 0x2B 0x7F im Format "2B7F" in der ETS verarbeitet wird.*

3.3. DIE FEHLEROBJEKTE

 48	Fehler	1 bit	K	L	-	Ü	-	Niedrig
 49	Fehler: nicht hexadezimal	1 bit	K	L	-	Ü	-	Niedrig
 50	Fehler: Kleinbuchstaben	1 bit	K	L	-	Ü	-	Niedrig
 51	Fehler: Länge ungerade	1 bit	K	L	-	Ü	-	Niedrig
 52	Fehler: im aktuellen Frame	1 bit	K	L	-	Ü	-	Niedrig
 53	Fehler: Empfang	1 bit	K	L	-	Ü	-	Niedrig
 54	Fehler: zu lang	1 bit	K	L	-	Ü	-	Niedrig

Bild 9. Fehler- Kommunikationsobjekte

Das Objekt Fehler gibt an, dass ein Fehler vorliegt. Dieser wird dann durch die weiteren Objekte konkretisiert. Jedes Objekt hat eine konkrete Bedeutung:

- **Fehler: nicht hexadezimal** In einem Frame befindet sich ein nicht-hexadezimaler Wert
- **Fehler: Kleinbuchstaben** In einem Frame befindet sich ein Kleinbuchstabe
- **Fehler: Länge ungerade** Die Länge eines eingegebenen Frames ist ungerade
- **Fehler: im aktuellen Frame** Der aufgetretene Fehler befindet sich im Frame der gerade verarbeitet wird.
- **Fehler: Empfang** Fehler beim Empfang der Daten am seriellen Port: Übertragungsgeschwindigkeit, Parität, Länge,...
- **Fehler: zu lang** Der empfangene Frame ist länger als 10 Byte.

3.3.1. FEHLERBEISPIELE

- Fehler eines nicht hexadezimalen Zeichens:** Wurde vom Benutzer ein nicht hexadezimaler Zeichen in den SKX-Parametern eingegeben, so wird eine 1 über das Objekt 48 gesendet. Ausserdem wird eine 1 über das Objekt 49 gesendet (nicht hexadezimal).

***Beispiel:** Der eingegebene Wert für das 1-bit Objekt 2 ist AAAG Da in diesem Fall ein nicht-hexadezimaler Zeichen (G) eingegeben wurde, wird eine 1 über das Objekt 49 gesendet.*

The screenshot shows a software interface with the following elements:

- Objekt 2:
- Kontrollmodus: Frame senden wenn Objekt=1
- Zu sendender Frame: AAAG

A red arrow labeled "Fehler" points to the character 'G' in the "Zu sendender Frame" field, which is also highlighted with a red box.

Bild 100. Nicht-hexadezimaler Zeichen

- Fehler eines Kleinbuchstabens:** Wurde vom Benutzer ein Kleinbuchstabe als Zeichen in den SKX-Parametern eingegeben, so wird eine 1 über das Objekt 48 gesendet. Ausserdem wird eine 1 über das Objekt 50 gesendet (Kleinbuchstaben).

***Beispiel:** Der eingegebene Wert für das 1-bit Objekt 0 ist ABcD Da in diesem Fall ein Kleinbuchstabe eingegeben wurde, wird eine 1 über das Objekt 50 gesendet.*

The screenshot shows a software interface with the following elements:

- Objekt 0:
- Kontrollmodus: Frame senden wenn Objekt=1
- Zu sendender Frame: ABcD
- Objekt 1:

A red arrow labeled "Fehler" points to the character 'c' in the "Zu sendender Frame" field, which is also highlighted with a red box.

Bild 111. Kleinbuchstabe

- 🔴 **Länge ungerade:** Wird versucht einen Frame mit ungerader Zeichenanzahl zu senden, so wird eine 1 über das Objekt 48 gesendet. Ausserdem wird eine 1 über das Objekt 51 gesendet (Länge ungerade).

***Beispiel:** In diesem Fall wurde für das Objekt 0 der Frame 520001A definiert (ungerade Zeichenanzahl) Dieser Fehler wird über das Objekt 51 auf den Bus gesendet.*

The screenshot shows a configuration window with the following elements:

- Objekt 0:
- Kontrollmodus: Frame senden wenn Objekt=1 (dropdown menu)
- Zu sendender Frame: 520001A (text input field)
- A red arrow labeled "Fehler" points from the "Zu sendender Frame" field to the "Kontrollmodus" dropdown.

Bild 122. Ungerader Frame

- 🔴 **Fehler: im aktuellen Frame** Befindet sich der aufgetretene Fehler im Frame der gerade verarbeitet wird, so wird ausserdem eine "1" über das Objekt 52 gesendet.

- 🔴 **Fehler bei den Kommunikationsparametern** Die Konfiguration der seriellen Kommunikationsparameter stimmt nicht mit der Konfiguration der empfangenen Frames überein. (Unterschiedliche Geschwindigkeit, Parität, Stopbits,...) In diesem Fall wird eine 1 über das Objekt 53 gesendet

***Beispiel:** Das externe Gerät verfügt über eine definierte Geschwindigkeit von 9600 Baud und der SKX von 1200 Baud. Alle Fehler die die Kommunikation betreffen, werden über das Objekt 53 "Fehler: Empfang" gemeldet.*

- 🔴 **Falsche Länge, Frame länger als 10 byte:** Für den Fall dass ein zu langer Datenrahmen vom externen Gerät empfangen wird, so wird eine 1 über das Objekt 48 gesendet. Ausserdem wird eine 1 über das Objekt 54 gesendet (zu lang)..

***Beispiel:** Wird ein Frame mit mehr als 10 Byte bzw. 20 HEX-Zeichen vom SKX Open empfangen , so wird eine 1 über das Fehlerobjekt 54 gesendet.*

Bitte beachten: Wird ein kompletter oder partieller Download durchgeführt, so wird im Falle eines Konfigurationsfehlers eines Frames (ungerade Zeichenanzahl, exzessive Länge, nicht-hexadezimale Zeichen) unmittelbar die entsprechende(n) Fehlermeldung(en) gesendet.



WERDE BENUTZER!

<http://zennio.zendesk.com>

TECHNISCHER SUPPORT