

User Manual

K-BUS® KNX USB Interface_V1.1

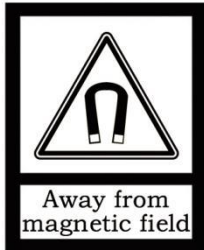
BNTP-USB/00.1



KNX/EIB Home and Building Control System

Attentions

1. Please keep devices away from strong magnetic field, high temperature, wet environment;



2. Do not fall the device to the ground or make them get hard impact;



3. Do not use wet cloth or volatile reagent to wipe the device;



4. Do not disassemble the devices.

Contents

Chapter 1 Summary	1
Chapter 2 Technical data	2
Chapter 3 Dimension and Connection Diagram	3
3.1. Dimension diagram	3
3.2. Connection diagram	3
Chapter 4 Operational description	4

Chapter 1 Summary

The KNX TP-USB Interface is designed for an intelligent building control system, which is used for facilitating communication between the PC and the KNX system. The device can connect the USB communication interface to the computer through a standard type A USB2.0 interface extension line, and the other end of the device can be connected to the KNX bus via a bus connection terminal. If the KNX bus on the device side is long enough, it can also be directly connected to the computer without USB extension line.

After the USB communication interface and the KNX bus are connected normally, the device can work and do not need an additional power supply.

Through the ETS software (version ETS5 and above) in the computer of this device, the individual address, configuration parameters, debugging and bus monitoring of the KNX device can be assigned.

The KNX TP-USB Interface has not an application program, and its individual address is able to be allocated in the individual address field of the bus connection window of the ETS. The factory default individual addresses are 10.15.255.

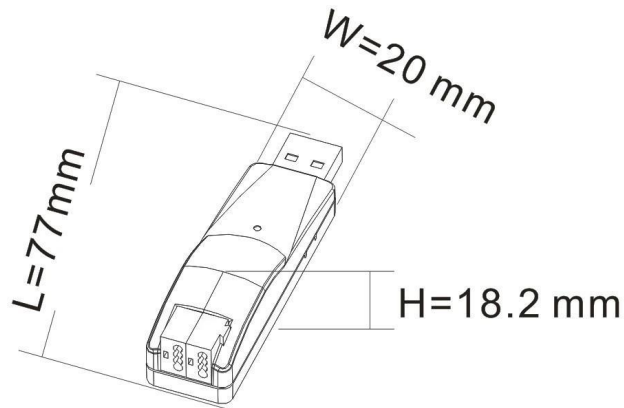
The KNX TP-USB Interface supports extended frames and long telegrams with up to 55 bytes APDU length. Due to HID profile support, no specific USB driver is required. The protocol used for communication between interface and host is the flexible “cEMI” protocol.

Chapter 2 Technical data

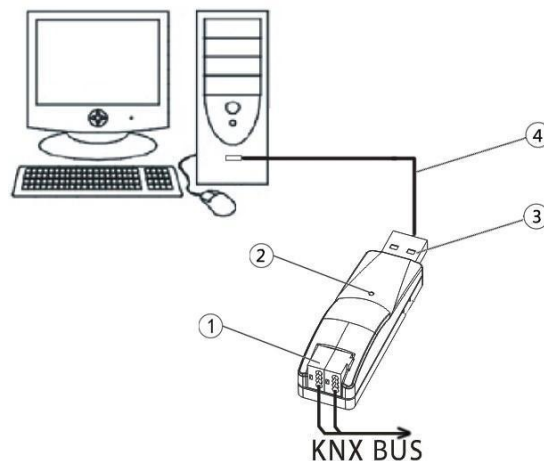
Power supply	Operating voltage	21-30V DC, via the KNX bus
	Current consumption, bus	<3.5mA/24V, <3mA/30V
	Power consumption, bus	<90mW
	USB voltage	5V DC
	Current consumption, USB	<100mA
	Power consumption, USB	<500mW
	Total power consumption, KNX and USB	<600 mW
Interface	USB Standard 2.0	
Connections	KNX	Bus connecting terminal (red/black)
	PC-connection	Via USB socket type A
		Max. cable length 5m (standardized)
Indication	Green LED ON	A PC is linked to the device
	Green LED flashing	Data transfer between USB interface and PC
	Red LED ON	The device is linked to the KNX bus
	Red LED flashing	Telegram traffic on the bus
Temperature	Operation	-5 °C ... + 45 °C
	Storage	-25 °C ... + 55 °C
	Transport	- 25 °C ... + 70 °C
Ambient	Humidity	<93%, except dewing
Mounting	As required	
Dimensions	18mm×20mm×77mm	
Weight	0.1kg	

Chapter 3 Dimension and Connection Diagram

3.1. Dimension diagram



3.2. Connection diagram



- ① KNX bus connection terminal
- ② The green LED lights on as soon as the USB interface and PC are connected and ready for operation. Flashing means that there is telegram communication between the USB interface and PC. The red LED lights on as soon as the USB interface and KNX bus are connected and ready for operation. Flashing means that there is telegram traffic on the bus.
- ③ USB serial ports connection terminal
- ④ USB extension cable

Chapter 4 Operational description

After the USB interface and KNX bus connection of this device are normal, we will see its default configuration in the bus connection window of the ETS, as follows:

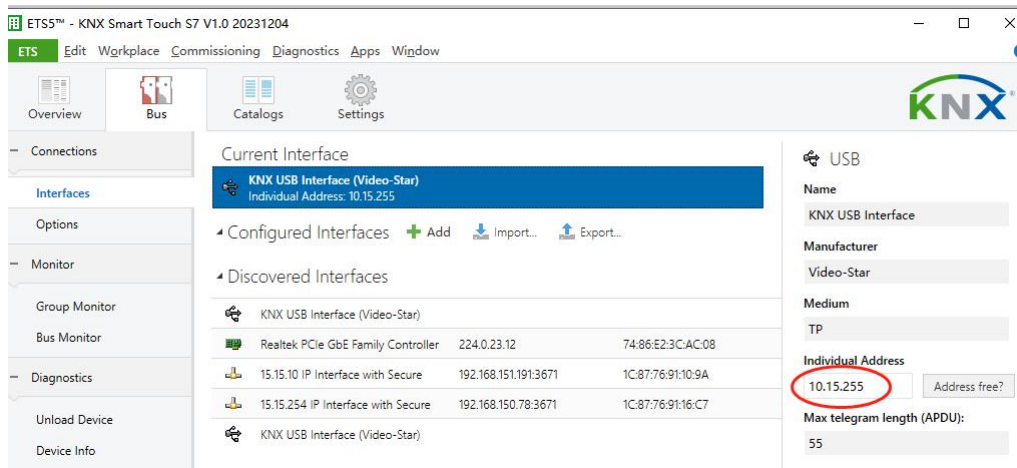


Fig.4.1 Bus connection window for ETS5--Individual address assignment (1)

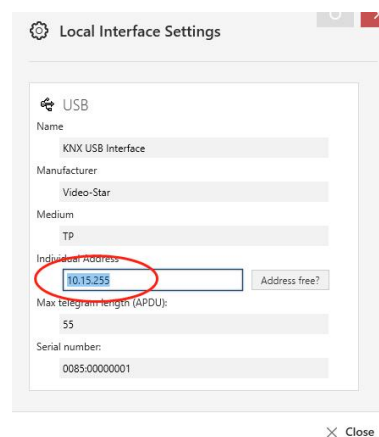
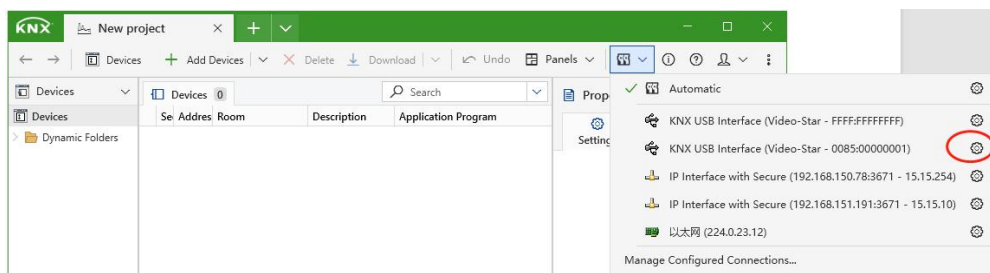


Fig.4.1 Bus connection window for ETS6--Individual address assignment (2)

In general, in the KNX topology system, it is necessary to set the correct individual address for the device according to the topology location of the device in the ETS bus connection window in Figure 4.1.

The device has an ETS database and can be used with ETS 5 or above version. But the device has not an application program, so there won't be parameters and objects in the database. The database is mainly convenient for us to add the device to a project and set itself individual address(as shown in Figure 4.2 below), to clarify the topological location or line where the device is located.

Note: The database cannot be used to program the individual address, and the individual address can only be modified via the bus connection window of the ETS.

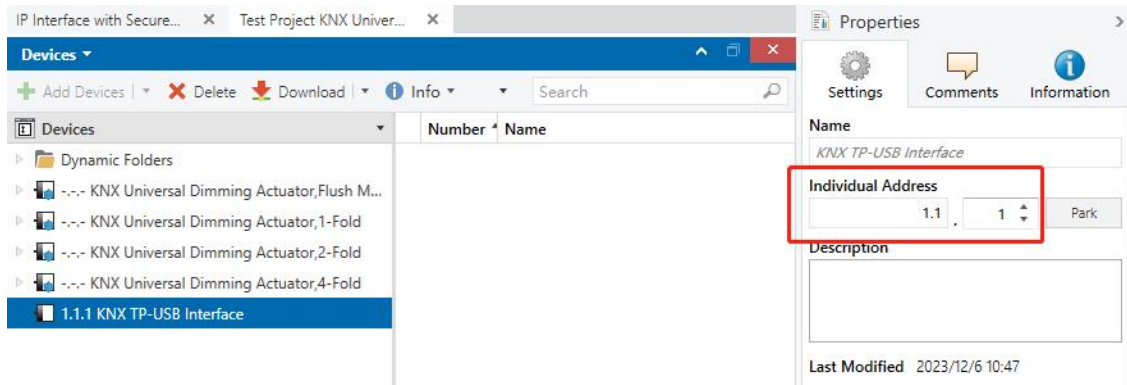


Fig.4.2 ETS project properties window--setting the desired individual address