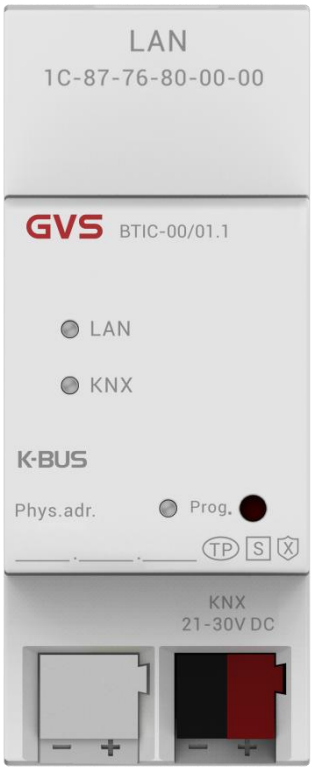
User Manual

K-BUS KNX-MQTT-HA Gateway_V1.1

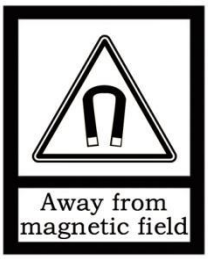
BTIC-00/01.1



KNX/EIB Home and Building Control System

Attentions

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







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



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



4.Please do not disassemble the devices.

Contents

Chapter 1 Summary 1
Chapter 2 Technical Data 2
Chapter 3 Dimension and Structural Diagram 3
3.1. Dimension Diagram 3
3.2. Structural Diagram 3
Chapter 4 Project Design and Programming 4
Chapter 5 Parameter setting description in the ETS 5
5.1. Parameter window“General” 5
5.1.1. MQTT Server Configuration 5
5.1.2. AC exchanges data with the KNX bus 6
5.2. Parameter window“Select Options” 8
5.2.1. Parameter window“Select Template x（x=1~8）” 8
5.3. Parameter window“Sensor, Select...” 10
5.3.1. Binary input 11
5.3.2. Value sender 12
5.3.3. Sensor 13
5.3.4. Select 14
5.4. Parameter window“Lighting...x（x=1~200）” 15
5.4.1. Lighting 16
5.4.2. Curtain 16
5.4.3. Scene 17
5.4.4. Valve 18
5.5. Parameter window“Switch...x（x=1~100）” 19
5.6. Parameter window“AC...x（x=1~16）” 20
5.7. Using the Integrated Tunneling Server 21
5.8. KNX Secure 23
5.9. Removing the Device 29
Chapter 6 Description of Communication Object 30
6.1. “Sensor...x（x=1~100）”Communication Object 30
6.2. “Lighting...x（x=1~200）”Communication Object 32
6.3. “Switch...x（x=1~100）”Communication Object 36
6.4. “AC...x（x=1~16）”Communication Object 37
Chapter 7 Factory settings 39

Chapter 1 Summary

The KNX-MQTT-HA Gateway as a protocol bridge, strictly complies with the Home Assistant (HA) MQTT protocol standard to implement protocol adaptation, fulfilling customers' need to integrate with the gateway via the HA protocol. The gateway performs bidirectional conversion between the KNX bus protocol and the HA standard MQTT protocol, enabling stable and compliant interconnection between the KNX smart control system and the Home Assistant platform.

If customers need a KNX-MQTT gateway, they can also use this gateway in accordance with the Home Assistant (HA) specification.

This manual provides detailed technical information about KNX-MQTT-HA Gateway for users as well as assembly and programming details, and explains how to use the device by the application examples.

KNX-MQTT-HA Gateway is powered from KNX bus. It is available to assign the physical address and configure the parameters by engineering design tools ETS with .knxprod (support edition ETS5.7 or higher).

The functions are summarized as follows:

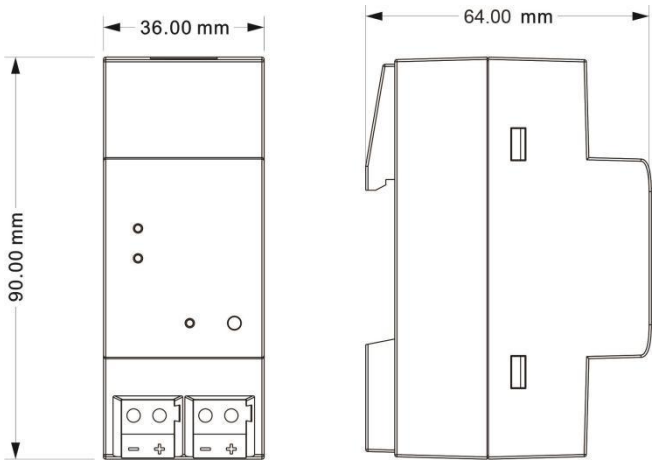
- ●Support MQTT 3.0 protocol, work in MQTT client mode, connect with the HA gateway via the MQTT server
- ●Only support MQTT plaintext communication, so pay attention to data security during use, and prohibit use in public network
- ●Support up to 5 KNXnet/IP connections at the same time
- ●Support up to 8 KNX-MQTT-HA gateway operation templates
- ● Support 100 channels of binary input/value sender/sensor/select functions
- ● Support 200 channels of light/curtain/scene/fan/valve/button functions
- ●Support 100 channels of switch function
- ● Support 16 channels of AC function
- ●Support the KNX Data Secure and KNXnet/IP Secure

Chapter 2 Technical Data

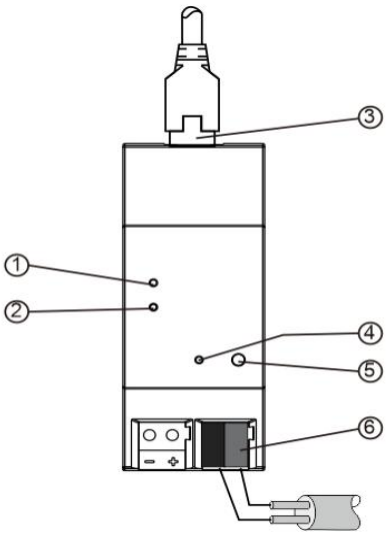
Power supply	Bus voltage	21-30V DC, via KNX bus
	Bus current Bus consumption	19.5mA/24V; 15.5mA/30V 470mW
Connections	KNX	Bus connection terminal (red/black)
	LAN	RJ45 socket for 10/100 Base-T
Operation and display	LAN LED	On: The network connect normally Flashing: The telegram traffic between device and network
	KNX LED	On: The KNX bus connect normally Flashing: The telegram traffic between KNX bus and device
	Programming button and red LED	For assigning the physical address
Temperature	Operation	–5 °C ... + 45 °C
	Storage Transportation	–25 °C ... + 55 °C – 25 °C ... + 70 °C
Ambient	Humidity	<93%, except dewing
Design	Standard 35mm DIN rail installation	
Dimension	36 × 90 × 64mm	
Weight	0.1kg	

Chapter 3 Dimension and Structural Diagram

3.1. Dimension Diagram



3.2. Structural Diagram



①LAN LED

②KNX LED

③LAN connection

④Programming LED

⑤Programming button

⑥KNX bus connection terminal

Reset the device to the factory configuration: press the programming button and hold for 4 seconds then release, repeat the operation for 4 times, and the interval between each operation is less than 3 seconds.

Chapter 4 Project Design and Programming

Application	Maximum of communication objects	Maximum number of group addresses	Maximum number of associations	Secure group addresses
KNX-MQTT-HA Gateway/00D8 1.0	**1691**	**2048**	**2048**	**768**

Chapter 5 Parameter setting description in the ETS

5.1. Parameter window“General”

5.1.1. MQTT Server Configuration

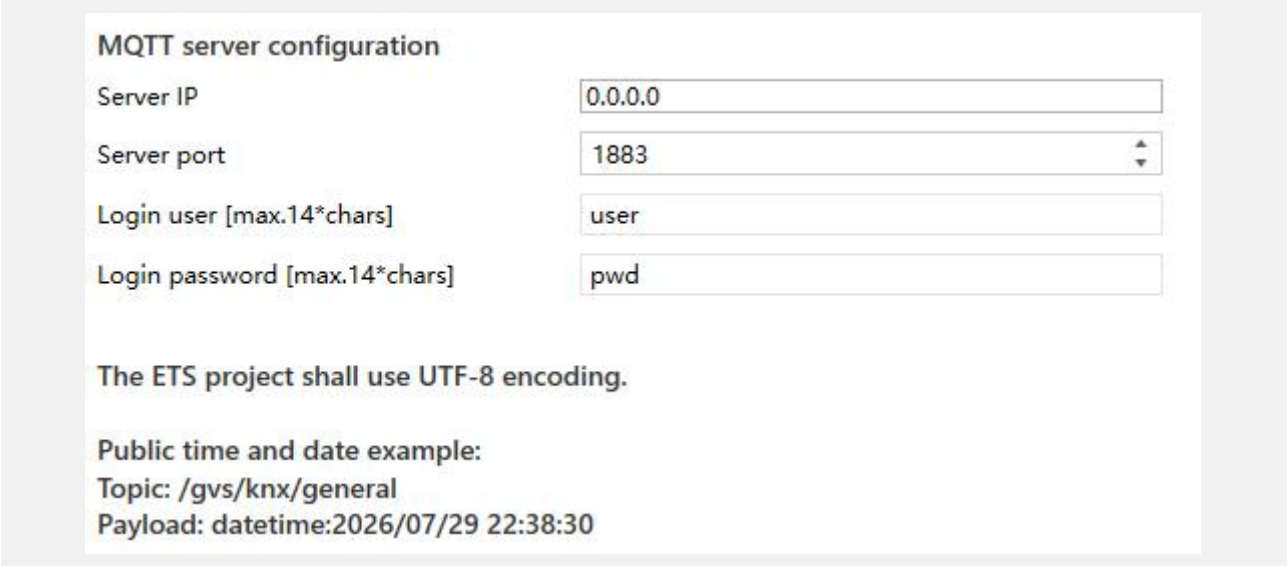


Fig.5.1.1 “General setting” Parameter window

MQTT server configuration

Parameter "Server ip"

This parameter is for setting the IP address of the MQTT server.

Parameter "Server port"

This parameter is for setting the port of the MQTT server.

Parameter "Login user [14*chars]"

This parameter is for setting the login user name for the MQTT server, up to input 14 characters.

Parameter "Login password [14*chars]"

This parameter is for setting the login password for the MQTT server, up to input 14 characters.

Note: The ETS project shall use UTE-8 encoding

5.1.2. AC exchanges data with the KNX bus

The following AC exchanges data with the KNX bus

Mode - auto [0..255]	0
Mode - heat [0..255]	1
Mode - cool [0..255]	3
Mode - fan only [0..255]	9
Mode - dry [0..255]	14
Mode - eMode 1 [0..255]	18
Mode - eMode 2 [0..255]	19
Mode - eMode 1 description	refresh
Mode - eMode 2 description	sleep
Fan Speed - auto [0..255]	0
Fan Speed - low [0..255]	1
Fan Speed - medium [0..255]	2
Fan Speed - high [0..255]	3
Fan Speed - eMode 1 [0..255]	4
Fan Speed - eMode 2 [0..255]	5
Fan Speed - eMode 3 [0..255]	6
Fan Speed - eMode 1 description	VeryLow
Fan Speed - eMode 2 description	quiet
Fan Speed - eMode 3 description	VeryHigh
Min. setpoint temperature [10..32]	16
Max. setpoint temperature [10..32]	30

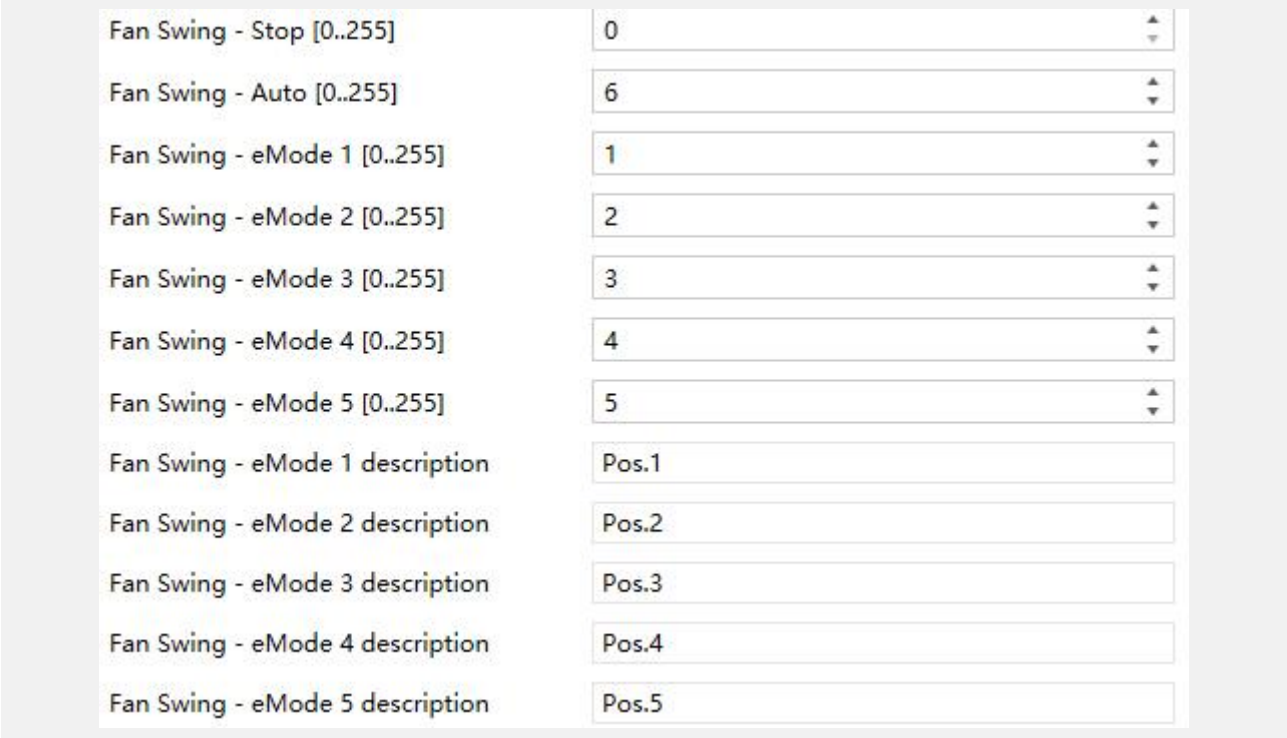


Fig.5.1.2 “General setting” Parameter window

The following AC exchanges data with the KNX bus

Parameter "Mode - ... [0..255]"

These parameters are for setting the output and status values of each mode. Options: **0..255**

Parameter "Fan Speed - ... [0..255]"

These parameters are for setting the output and status values of each fan speed. Options: **0..255**

Parameter "Fan Swing - ... [0..255]"

These parameters are for setting the output and status values of each fan swing. Options: **0..255**

Parameter "Temperature Min.[10..32]"

Parameter "Temperature Max.[10..32]"

These parameters are for setting the adjustable range of the temperature. If the temperature beyond the limited range, it will output the limited temperature. Options: **10..32 °C**

Note: the minimum value should be less than the maximum value.

Parameter "...eMode.x desc."

These parameters are for setting the description of each custom mode, up to input 8 characters.

5.2. Parameter window“Select Options”



Fig.5.2 “Select Options” Parameter window

Parameter "Number of templates"

This parameter is for setting the number of parameter templates for the KNX-MQTT-HA gateway operation. Options: **1...8**

5.2.1. Parameter window“Select Template x（x=1~8）”

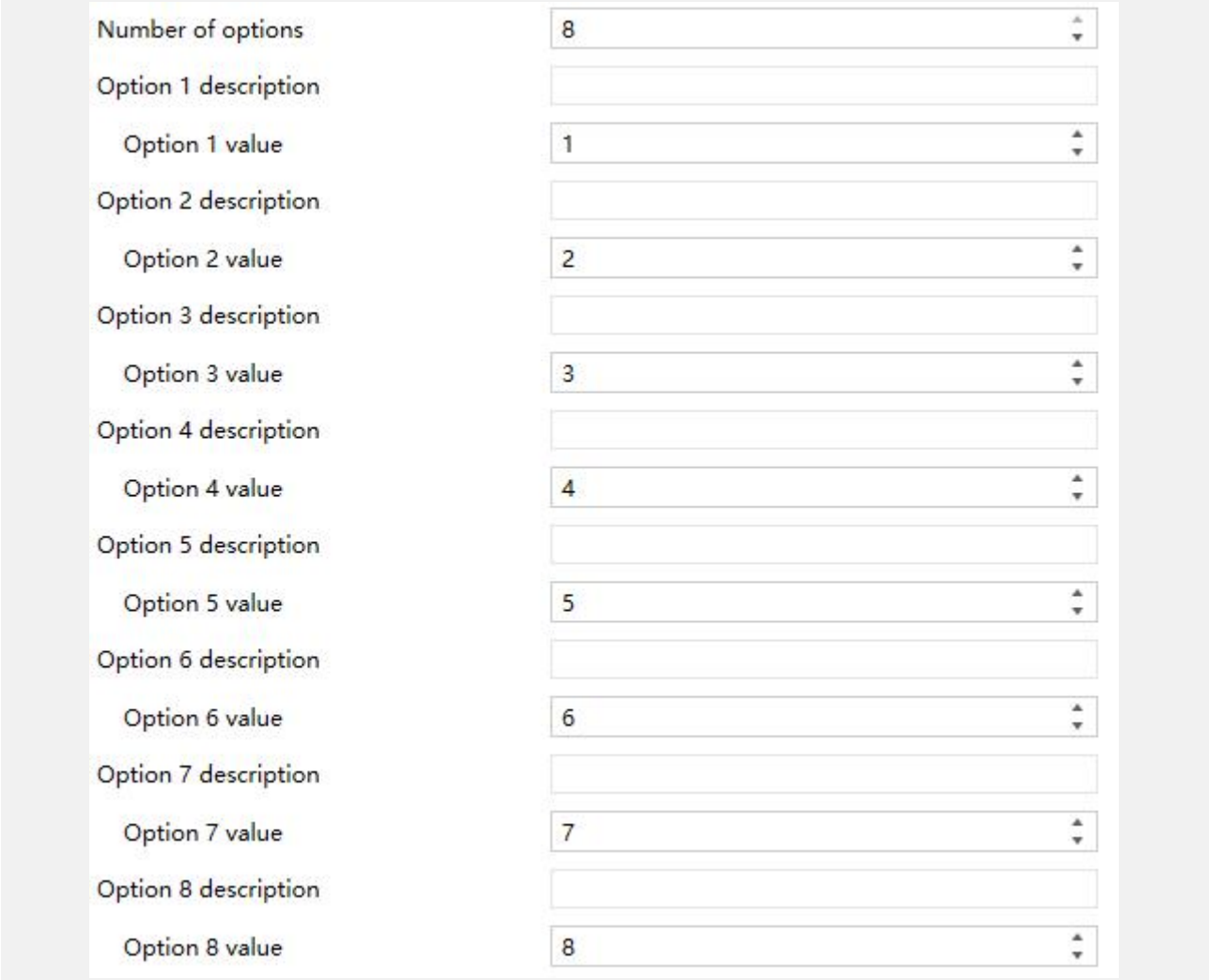


Fig.5.2.1 “Select Template x” Parameter window

Parameter "Number of options"

This parameter is for setting the number of options for each template. Options: **1...8**

Parameter "Option x description"

This parameter is for setting the custom description for each template option, up to input 14 characters.

---Parameter "Option x value"

This parameter is for setting the KNX value corresponding to each template option.

Options: **0...255**

5.3. Parameter window“Sensor, Select...”

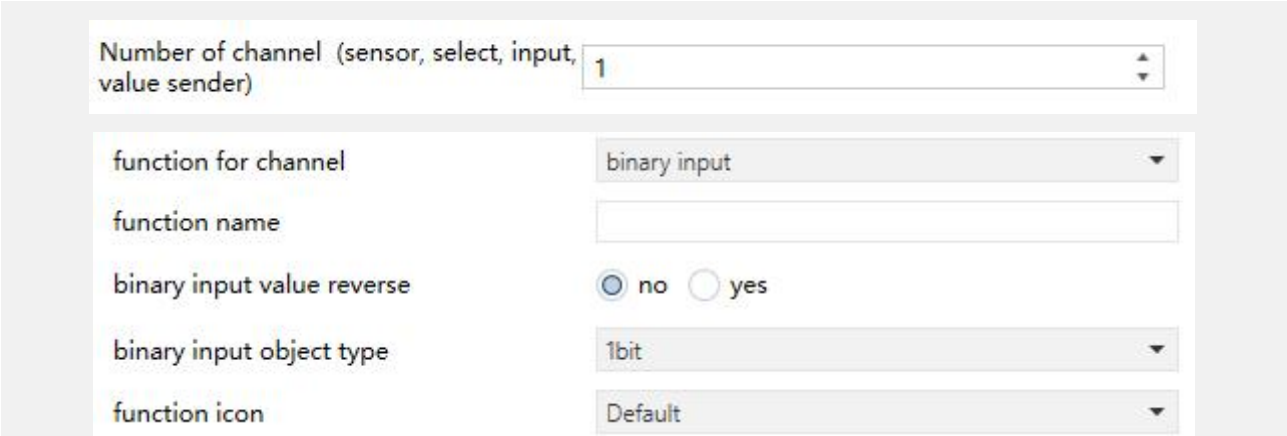


Fig.5.3 “Sensor, Select...” Parameter window

Parameter "Number of channel (sensor, select, input, value sender)"

This parameter is for setting the number of sensor channels. Options: **0...100**

Parameter "Function for channel"

This parameter is for setting the function of the sensor channel. Options:

No function

Binary input

Value sender

Sensor

Select

Parameter "Function name"

This parameter is for setting the custom description of the sensor channel, up to input 14 characters.

Parameter "Function icon"

This parameter is for setting the icon used for the channel function.

The icons corresponding to the options are described in the "Home Assistant" icon library.

Chapters as follow explain the channel function separately.

5.3.1. Binary input

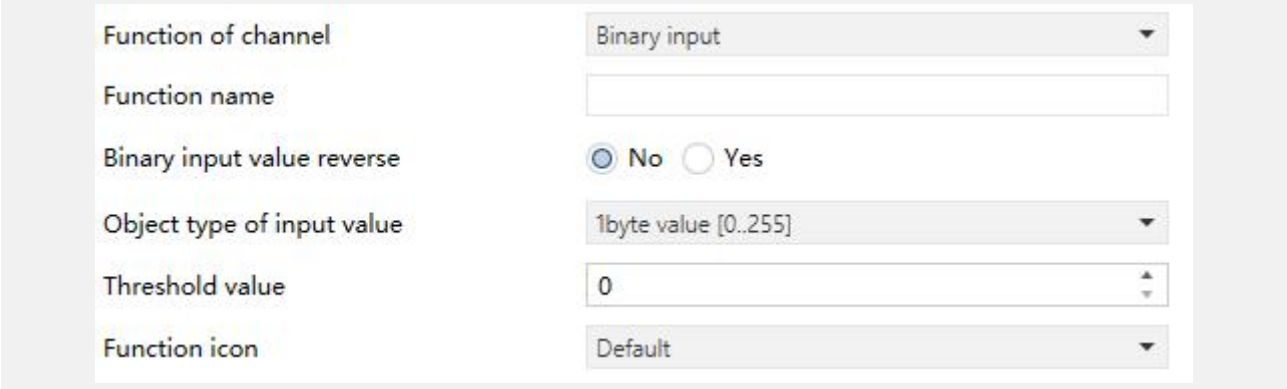


Fig.5.3.1 “binary input” Parameter window

Parameter “Binary input value reverse”

This parameter is for setting whether to reverse the binary input value. Options:

No

Yes

Parameter “Binary input object type”

This parameter is for setting the binary input object type. Options:

1bit

1byte value[0...255]

1byte value[-128...127]

2bytes value [0...65535]

2bytes value[-23768...32767]

2bytes float value

4bytes float value

4bytes value[0...4294967295]

4bytes value[-2147483648...2147483647]

Parameter “Threshold value”

This parameter is visible when “1bit” is not selected. Set the threshold value. Options according to binary input object type.

5.3.2. Value sender

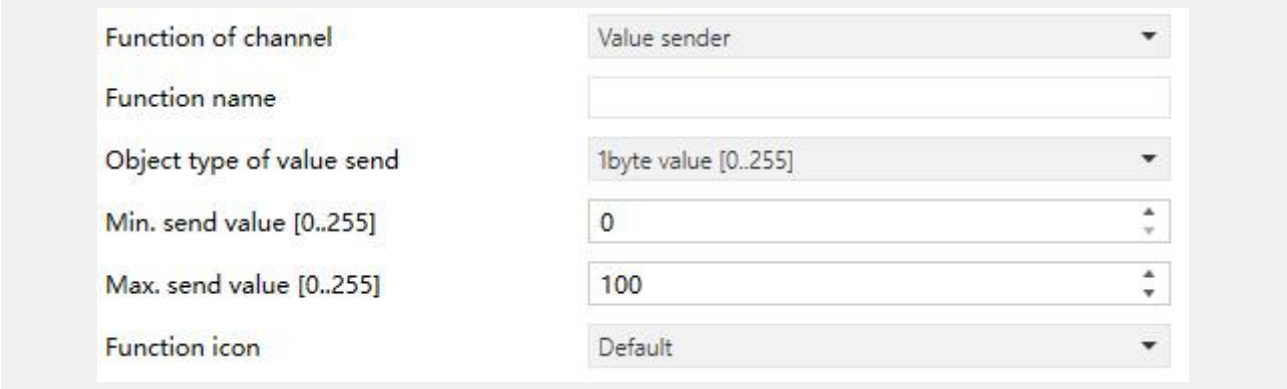


Fig.5.3.2 “number” Parameter window

Parameter "Object type of value send"

This parameter is for setting the object type of value send. Options:

1bit

1byte value[0...255]

2bytes value[0...65535]

4bytes value[0...4294967295]

Parameter "Min.send value[...]"

Parameter "Max.send value[...]"

These parameters are visible when “1bit” is not selected. Set the adjustable range of the send value. If the send value beyond the limited range, it will output the limited value.

Options according to value send object type.

Note: the minimum send value should be less than the maximum send value.

5.3.3. Sensor

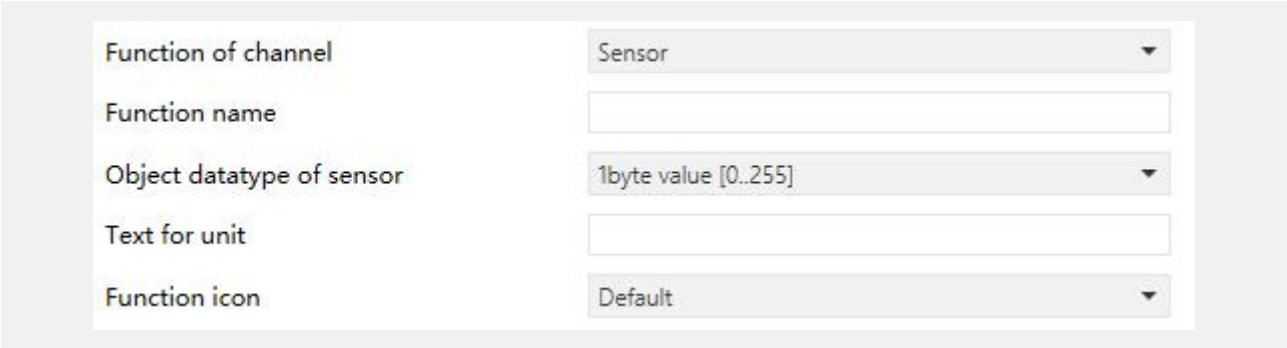


Fig. 5.3.3 “sensor” Parameter window

Parameter "Object datatype of sensor"

This parameter is for setting the object datatype of sensor. Options:

1bit

1byte value[0...255]

1byte value[-128...127]

2bytes value [0...65535]

2bytes value[-23768...32767]

2bytes float value

4bytes float value

4bytes value[0...4294967295]

4bytes value[-2147483648...2147483647]

Parameter "Text for unit"

This parameter is for setting the text for unit, up to input 8 characters.

5.3.4. Select

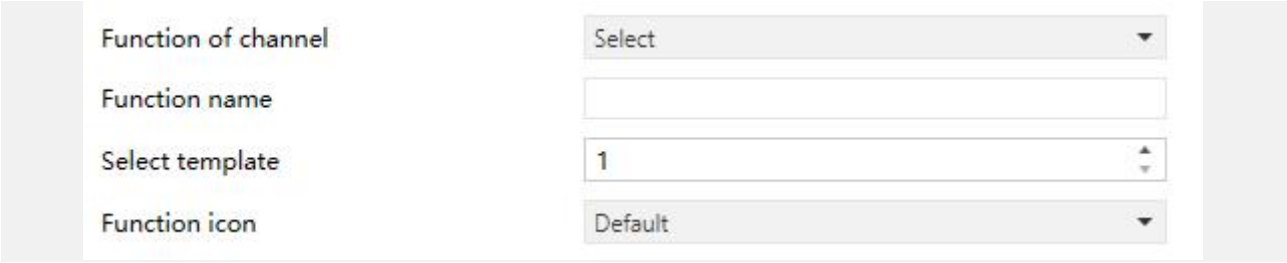


Fig.5.3.4 “select” Parameter window

Parameter "Select template"

This parameter is for setting the selected template number. Options: **1...8**

5.4. Parameter window“Lighting...x（x=1~200）”

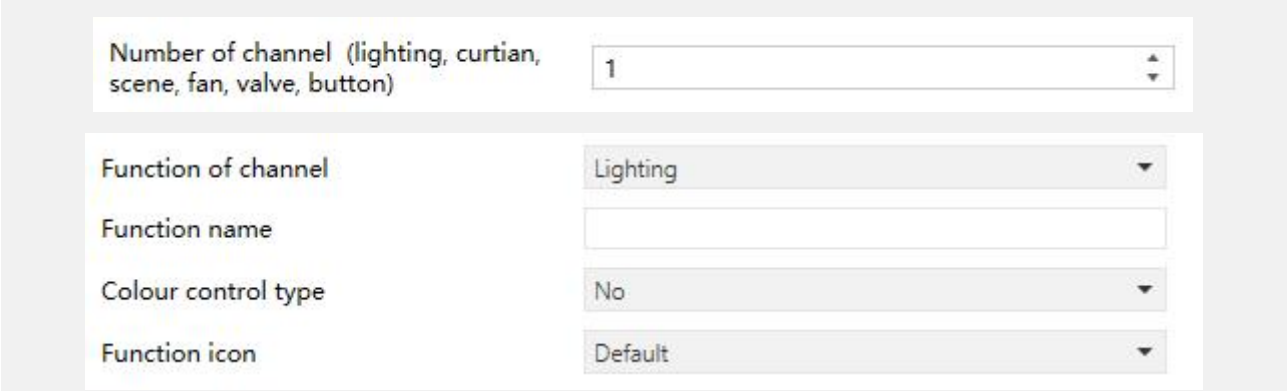


Fig.5.4 “Lighting...x” Parameter window

Parameter "Number of channel (lighting, curtain, scene, fan, valve, button)"

This parameter is for setting the number of light channels. Options: **0...200**

Parameter "Function for channel"

This parameter is for setting the function of the light channel. Options:

No function

Lighting

Curtain

Scene

Fan

Valve

Button

Parameter "Function name"

This parameter is for setting the custom description of the light channel, up to input 14 characters.

Parameter "Function icon"

This parameter is for setting the icon used for the channel function.

The icons corresponding to the options are described in the "Home Assistant" icon library.

Chapters as follow explain the channel function separately.

5.4.1. Lighting



Fig.5.4.1 “lighting” Parameter window

Parameter "Colour control type"

This parameter is for setting the colour control type. Options:

No

Colour temperature

RGB

RGBW

5.4.2. Curtain

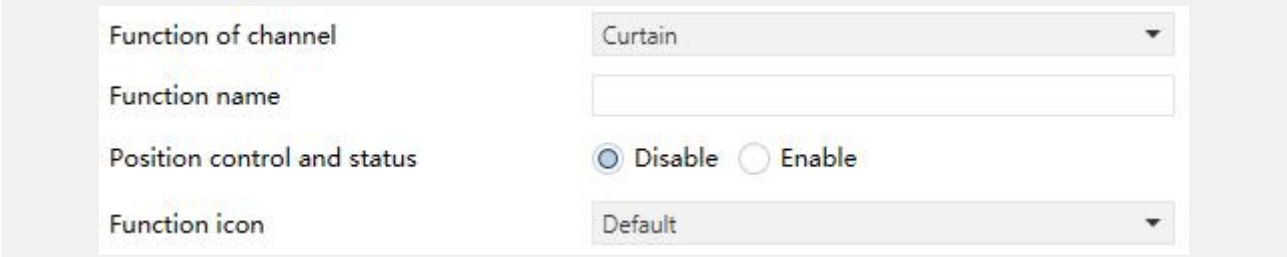


Fig.5.4.2 “curtain” Parameter window

Parameter "Position control and status"

This parameter is for setting whether to enable position control and status function. Options:

Disable

Enable

5.4.3. Scene

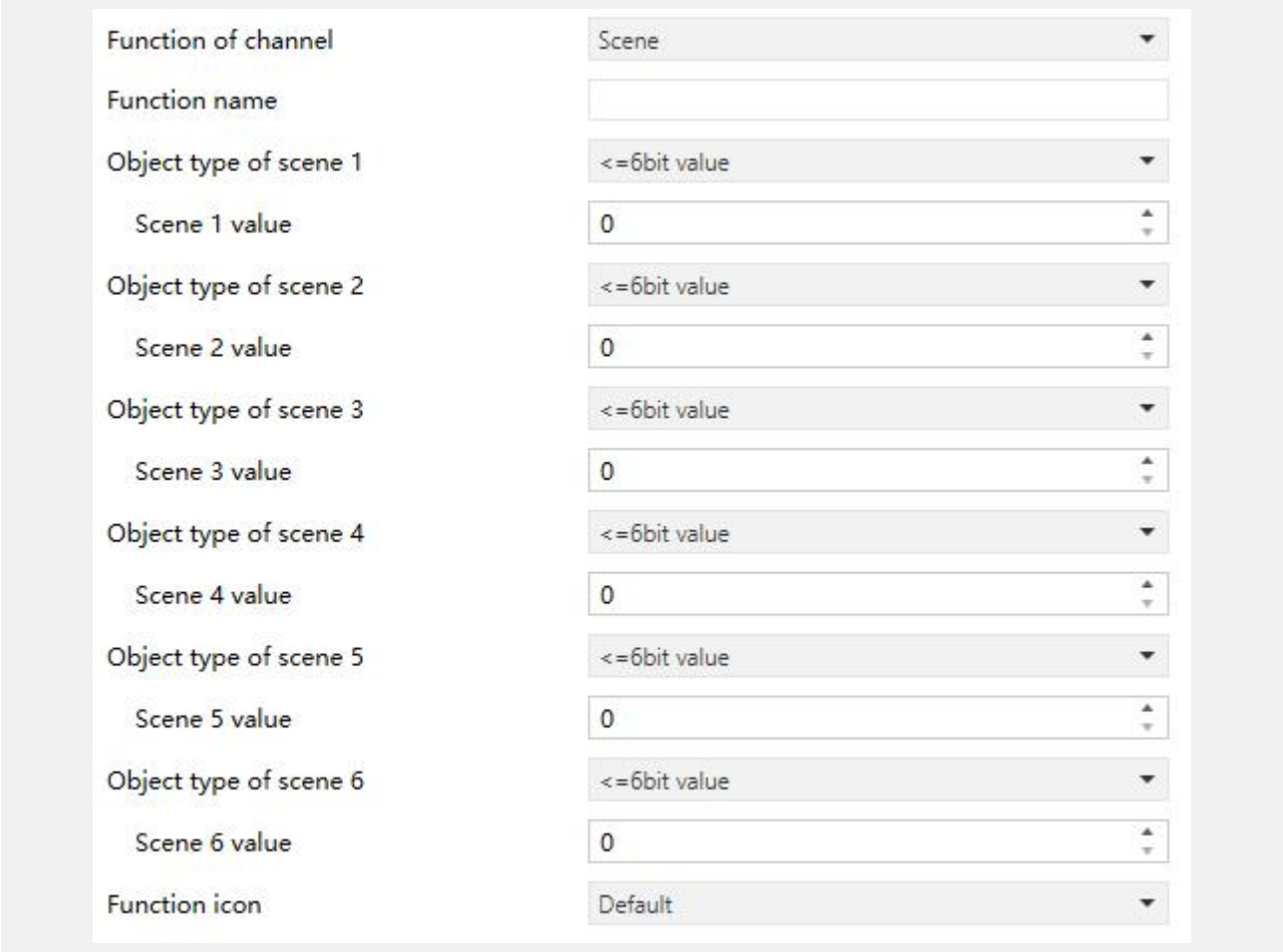


Fig.5.4.3 “scene” Parameter window

Parameter "Object type of scene x"

This parameter is for setting the object type of scene x. Options:

no

<=6bit value

1byte value[0...255]

2byte value[0...65535]

2byte float value

Parameter "Scene x value"

This parameter is for setting the scene x value. Options according to object type of scene x.

5.4.4. Valve

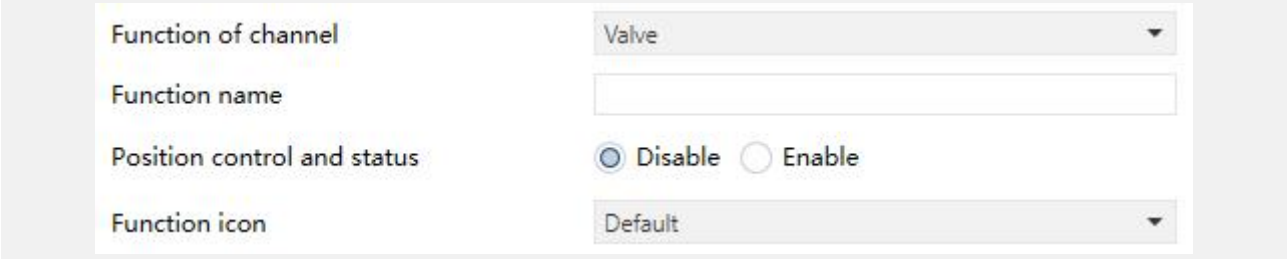


Fig.5.4.4 “valve” Parameter window

Parameter "Position control and status"

This parameter is for setting whether to enable position control and status function. Options:

Disable

Enable

5.5. Parameter window“Switch...x（x=1~100）”

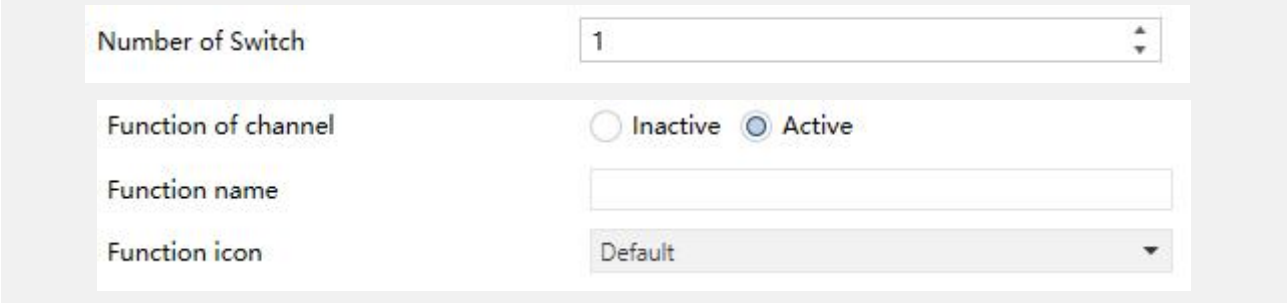


Fig.5.5 “Switch...x” Parameter window

Parameter "Number of Switch"

This parameter is for setting the number of enabled switch channels. Options: **0...100**

Parameter "Function for channel"

This parameter is for setting whether to enable the switch channel function. Options:

Inactive

Active

Parameter "Function name"

This parameter is for setting the custom description of the switch channel, up to input 14 characters.

Parameter "Function icon"

This parameter is for setting the icon used for the channel function.

The icons corresponding to the options are described in the “Home Assistant” icon library .

5.6. Parameter window“AC...x（x=1~16）”

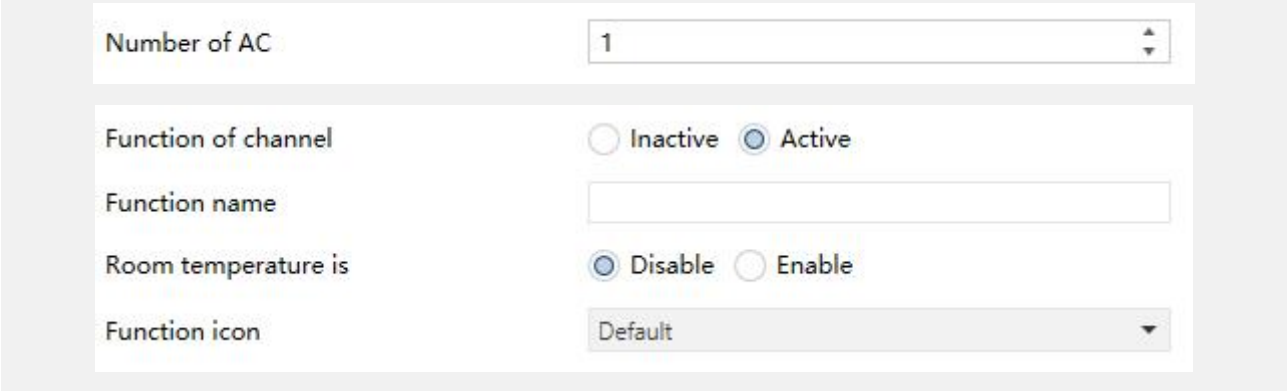


Fig.5.6 “AC...x” Parameter window

Parameter "Number of AC"

This parameter is for setting the number of AC channels. Options: **0...16**

Parameter "Function for channel"

This parameter is for setting whether to enable the AC channel function. Options:

Inactive

Active

Parameter "Function name"

This parameter is for setting the custom description of the AC channel, up to input 14 characters.

Parameter "Room temperature is"

This parameter is for setting whether to enable room temperature function. Options:

Disable

Enable

Parameter "Function icon"

This parameter is for setting the icon used for the channel function.

The icons corresponding to the options are described in the "Home Assistant" icon library.

5.7. Using the Integrated Tunneling Server

The KNX-MQTT-HA gateway provides 5 additional physical addresses, as shown in Fig.5.7.1, for tunnel connections, such as: programming interface (ETS), or for connecting visualization clients, smartphone apps, tablets, bus tools, etc.

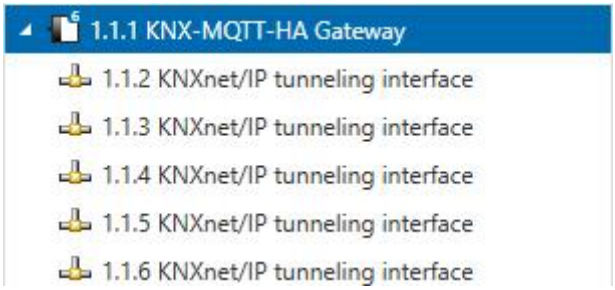


Fig.5.7.1

The physical address of each tunnel connection can be modified in the settings properties window, and their physical addresses must comply with the topology requirements and be located on the same line. In ETS5, when assigning physical addresses to the device, the first 5 free physical addresses on that line are automatically assigned to the individual tunnel connections.

If you do not want to assign a physical address to a tunnel, activate “Park” (as shown in Fig.5.7.2); after the download, this tunnel will receive address 15.15.255. If all tunnels are “Park” , all tunnels will be assigned address 15.15.255. (15.15.255 is the default address of devices without an assigned physical address)

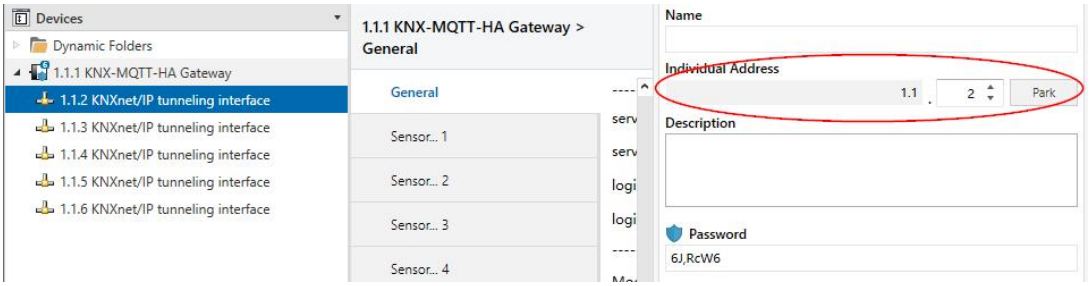


Fig.5.7.2

In addition, tunnel connections can also be encrypted using KNX Security. First activate “Secure Commissioning”, then activate “Secure Tunneling”, as shown in Fig.5.7.3. After activating “Secure Tunneling”, the password of each tunnel connection can be set in ETS, as shown in Fig.5.7.4, users can change this password as required.

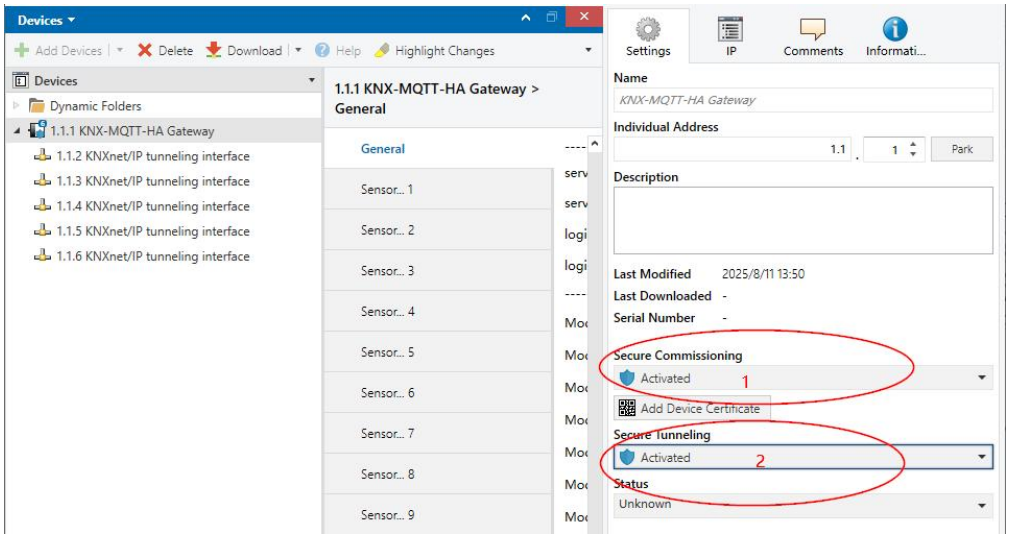


Fig.5.7.3

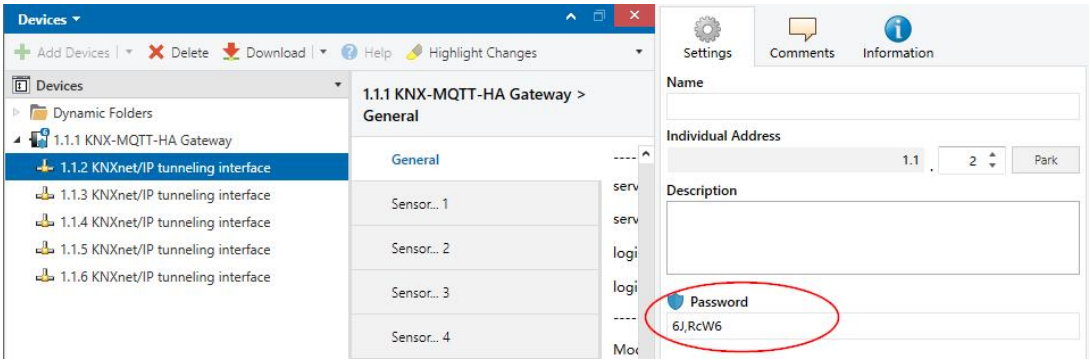


Fig.5.7.4

If the project has no password assigned, manually activating “Secure Commissioning” will prompt you to assign a project password, as shown in Fig.5.7.5. Without a project password, “Secure Commissioning” cannot be activated.

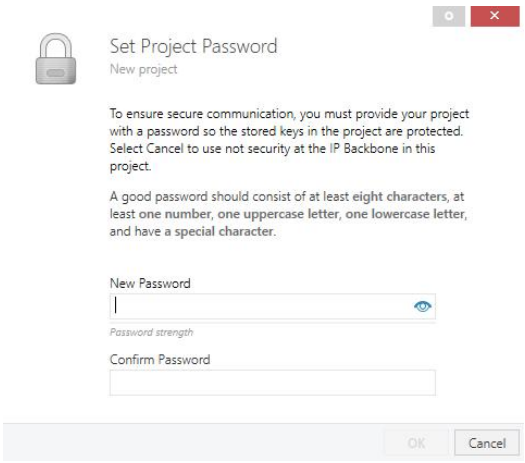


Fig.5.7.5

5.8. KNX Secure

KNX-MQTT-HA gateway is a KNX device that complies with the KNX secure standard. That is, you can run the device in safe way.

If secure commissioning is activated in ETS project, the following information must be considered during device debugging:



❖It is essential to assign a project password as soon as a KNX Secure device is imported into a project. This will protect the project against unauthorized access.

The password must be kept in a safe place – access to the project is not possible without it (not even the KNX Association or device manufacturer will be able to access it)!

Without the project password, the commissioning key will not be able to be imported.

❖A commissioning key is required when commissioning a KNX Secure device (first download). This key (FDSK = Factory Default Setup Key) is included on a sticker on the side of the device, and it must be imported into the ETS prior to the first download:

✧On the first download of the device, a window pops up in the ETS to prompt the user to enter the key, as shown in Fig.5.8.1 below.

The certificate can also be read from the device using a QR scanner (recommended).

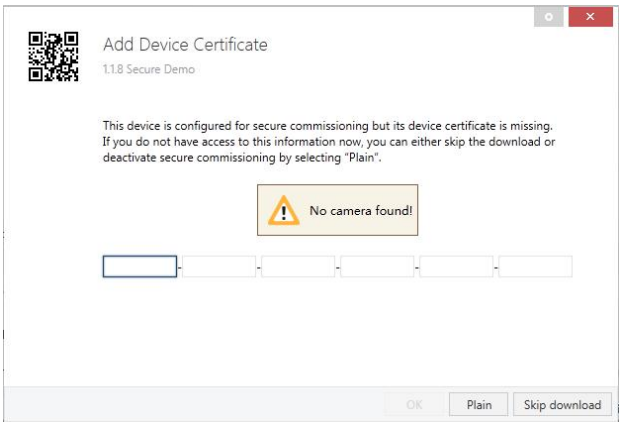


Fig.5.8.1 Add Device Certificate window

✧Alternatively, the certificates of all Secure devices can be entered in the ETS beforehand.

This is done on the "Security" tab on the project overview page, as shown in Fig.5.8.2 below.

The certificates can be also added to the selected device in the project, as shown in Fig.5.8.3.

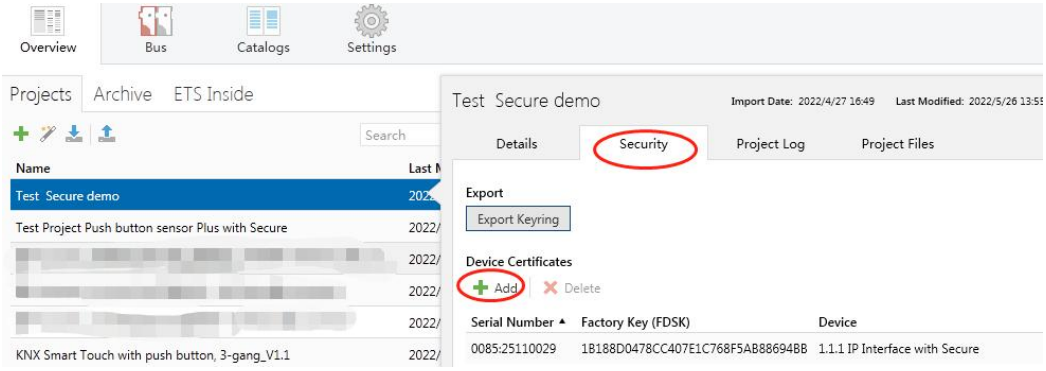


Fig.5.8.2 Add Device Certificate

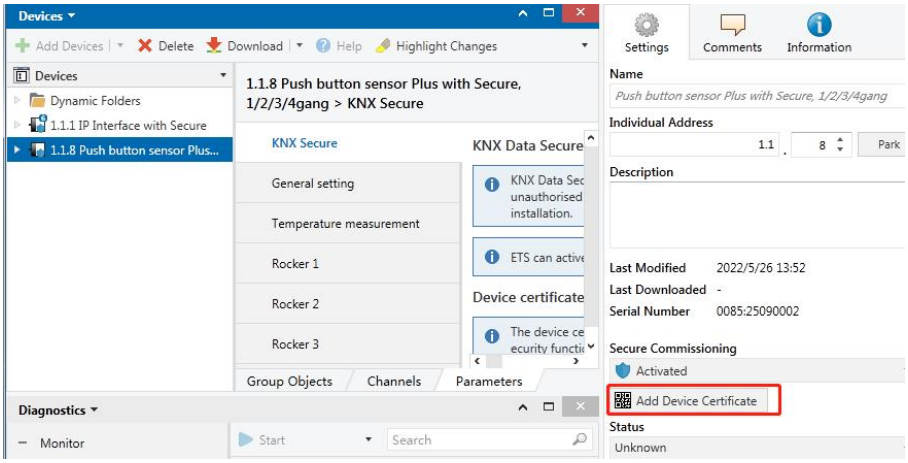


Fig.5.8.3 Add Device Certificate

✧There is a FDSK sticker on the device, which is used for viewing FDSK number.

Without the FDSK, it will no longer be possible to operate the device in KNX Secure mode after a reset.

The FDSK is required only for initial commissioning. After entering the initial FDSK, the ETS will assign a new key, as shown in Fig.5.8.4 below.

The FDSK will be required again only if the device was reset to its factory settings (e.g. If the device is to be used in a different ETS project).

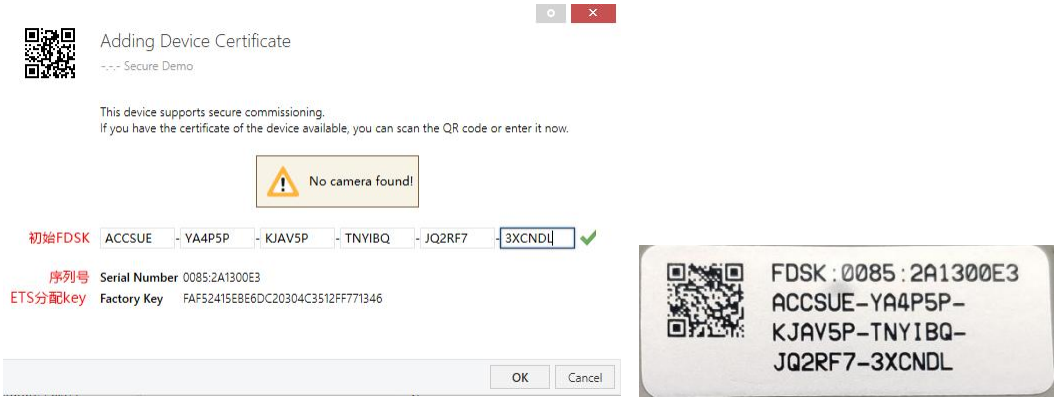


Fig.5.8.4

Example:

If this application in the project needs to be tried with another device, it is no longer the original device. When the application is downloaded to a new device, the following prompt will appear on the left of Fig.5.8.5, click yes, the Add Device Certificate window will appear, then enter the initial FDSK of the new device, and you need to reset the device to the factory settings (it is not required if the device is still factory default; If it has been used, it will be required to reset, otherwise the following error message will appear on the right of Fig.5.8.5 and then the device can be successfully downloaded again.

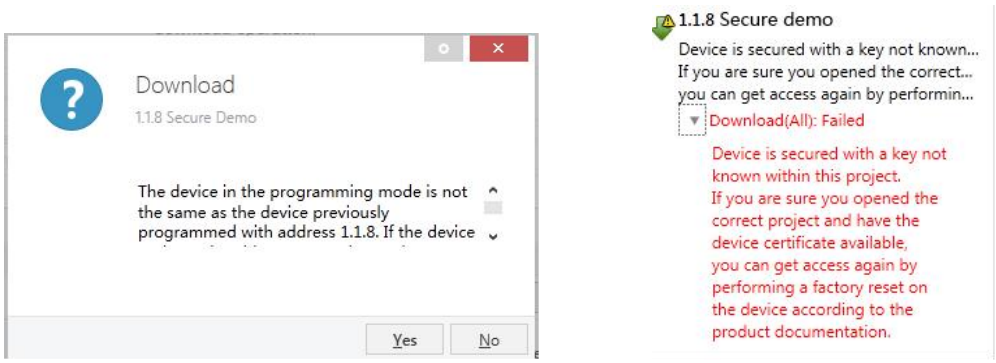


Fig.5.8.5 Example

Whether the device is replaced in the same project, or the device is replaced in a different project, the processing is similar: **Reset the device to the factory settings, then reassign the FDSK.**

After the device is downloaded successfully, the label Add Device Certificate turns gray, indicating that the key for this device has been assigned successfully, as shown in Fig.5.8.6 below.



Fig.5.8.6

ETS generates and manages keys:

Keys and passwords can be exported as needed to the use of security keys outside of the associated ETS projects. As shown in Fig.5.8.7 below, the file extension is .knxkeys.

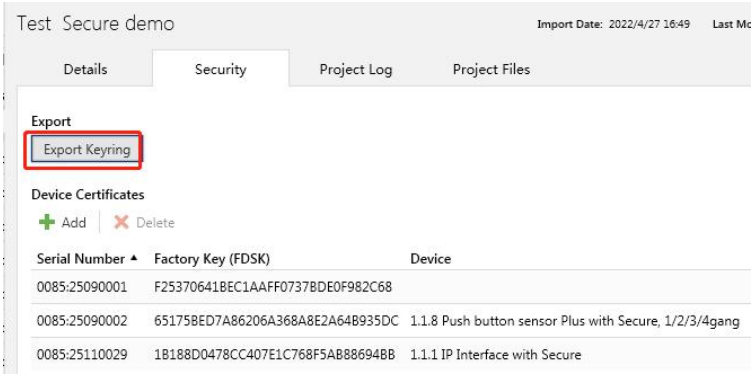


Fig.5.8.7

Example of ETS connecting via an IP tunnel:

Select the IP device, select one of the tunnels (e.g. physical address 1.1.2), and click Test. A dialog for entering the password and authentication code appears (the password and authentication code can be viewed in the properties of the IP device in the project). Enter the password and authentication code and click OK; the word “Ok” appears next to the Test button. Click Select to connect. The whole process is shown in Fig.5.8.8.

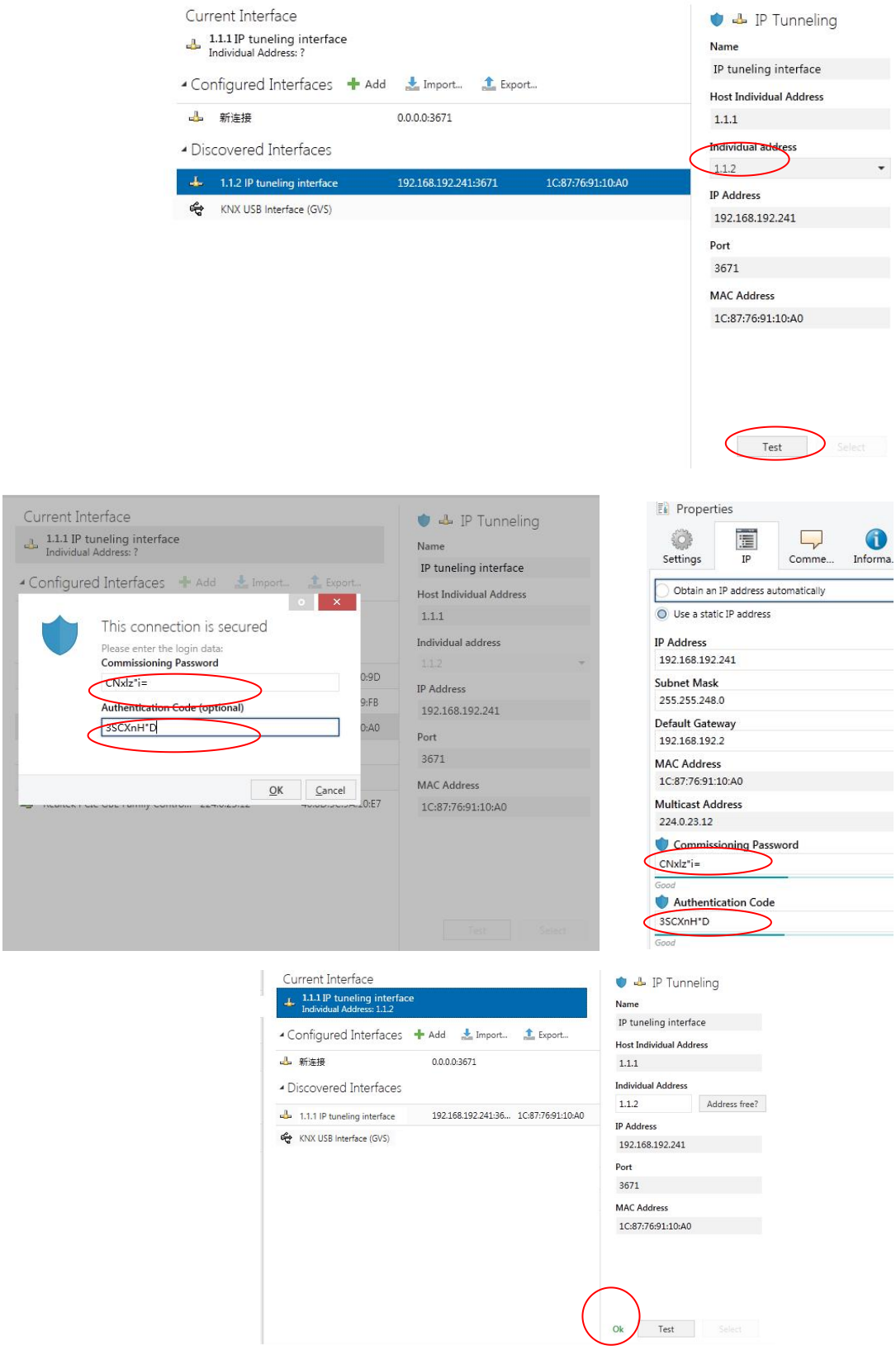


Fig.5.8.8 Example of an IP tunnel connection

In Fig.5.8.8, if “Secure Tunneling” is not activated, no commissioning password or authentication code is required when connecting as an interface. If “Secure Tunneling” is activated, ETS will prompt for the commissioning password and authentication code when connecting.

If necessary, the KNX-MQTT-HA gateway can be reset to factory settings; please refer to Chapter 7.

Note: Any USB interface used for programming a KNX Secure device must support "long frames". Otherwise ETS will report a download failure information, as shown in Fig.5.8.9.

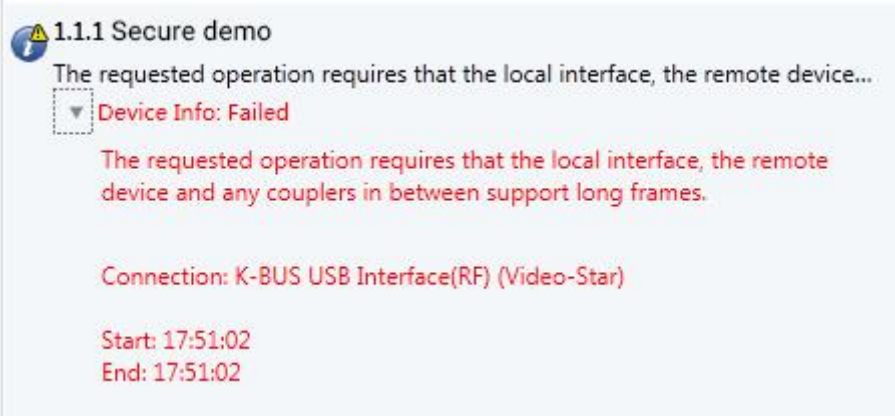


Fig.5.8.9

5.9. Removing the Device

The device can be reset to factory settings. Since it is a secure device, the following information must be observed:

When the device is operated in KNX Secure mode, it can only be reset via ETS when ETS uses the project in which this device is parameterized or when the project provides its commissioning key.

The device can be removed by right-clicking the device in its project in ETS.

Removing the application program:

· The IP address and IP configuration will be retained

· The password for the Tunneling server will be deleted, the commissioning password and authentication code will not be required when connecting (leave the dialog empty if it appears)

· The key assigned by ETS will be retained, no FDSK is required for reprogramming

· The physical address will be retained

Removing the physical address and application program:

· The device will be reset to the factory state

· Recommissioning will require the FDSK, unless the device is in its original commissioning project and its FDSK is still present in the project; then the interface can be reprogrammed directly.

Chapter 6 Description of Communication Object

The communication object is the medium to communicate other device on the bus, namely only the communication object can communicate with the bus.

NOTE: "C" in "Flag" column in the below table means enable the communication function of the object; "W" means value of object can be written from the bus; "R" means the value of the object can be read by the other devices; "T" means the object has the transmission function; "U" means the value of the object can be updated.

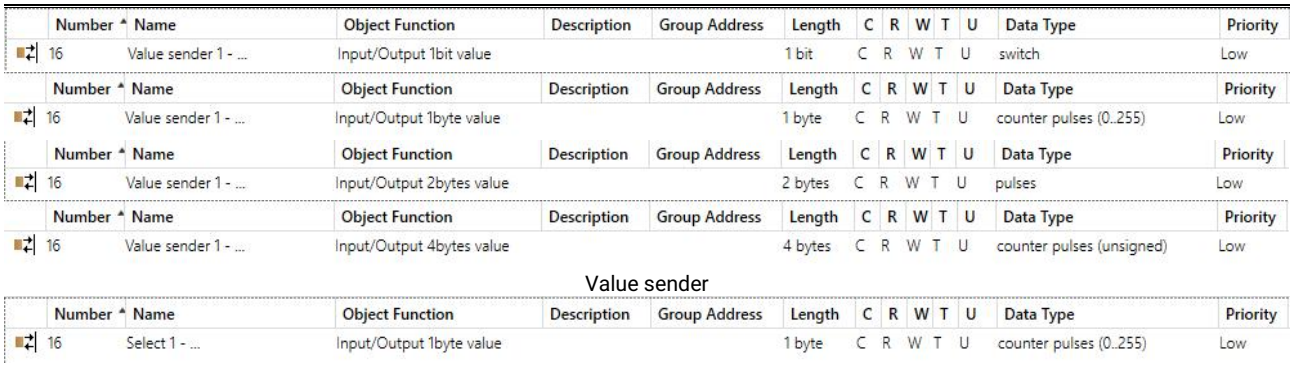
6.1. "Sensor...x（x=1~100）"Communication Object

Number	Name	Object Function	Description	Group Address	Length	C	R	W	T	U	Data Type	Priority
16	Binary 1 - ...	Input 1bit value			1 bit	C	-	W	-	U	switch	Low
16	Binary 1 - ...	Input 1byte unsigned value			1 byte	C	-	W	-	U	counter pulses (0..255)	Low
16	Binary 1 - ...	Input 1byte signed value			1 byte	C	-	W	-	U	counter pulses (-128..127)	Low
16	Binary 1 - ...	Input 2bytes unsigned value			2 bytes	C	-	W	-	U	pulses	Low
16	Binary 1 - ...	Input 2bytes signed value			2 bytes	C	-	W	-	U	pulses difference	Low
16	Binary 1 - ...	Input 2bytes float value			2 bytes	C	-	W	-	U	2-byte float value	Low
16	Binary 1 - ...	Input 4bytes float value			4 bytes	C	-	W	-	U	4-byte float value	Low
16	Binary 1 - ...	Input 4bytes unsigned value			4 bytes	C	-	W	-	U	counter pulses (unsigned)	Low
16	Binary 1 - ...	Input 4bytes signed value			4 bytes	C	-	W	-	U	counter pulses (signed)	Low

binary input

Number	Name	Object Function	Description	Group Address	Length	C	R	W	T	U	Data Type	Priority
16	Sensor 1 - ...	Input 1bit value			1 bit	C	-	W	-	U	switch	Low
16	Sensor 1 - ...	Input 1byte unsigned value			1 byte	C	-	W	-	U	counter pulses (0..255)	Low
16	Sensor 1 - ...	Input 1byte signed value			1 byte	C	-	W	-	U	counter pulses (-128..127)	Low
16	Sensor 1 - ...	Input 2bytes unsigned value			2 bytes	C	-	W	-	U	pulses	Low
16	Sensor 1 - ...	Input 2bytes signed value			2 bytes	C	-	W	-	U	pulses difference	Low
16	Sensor 1 - ...	Input 2bytes float value			2 bytes	C	-	W	-	U	2-byte float value	Low
16	Sensor 1 - ...	Input 4bytes float value			4 bytes	C	-	W	-	U	4-byte float value	Low
16	Sensor 1 - ...	Input 4bytes unsigned value			4 bytes	C	-	W	-	U	counter pulses (unsigned)	Low
16	Sensor 1 - ...	Input 4bytes signed value			4 bytes	C	-	W	-	U	counter pulses (signed)	Low

sensor



Number	Name	Object Function	Description	Group Address	Length	C	R	W	T	U	Data Type	Priority
16	Value sender 1 - ...	Input/Output 1bit value			1 bit	C	R	W	T	U	switch	Low
16	Value sender 1 - ...	Input/Output 1byte value			1 byte	C	R	W	T	U	counter pulses (0..255)	Low
16	Value sender 1 - ...	Input/Output 2bytes value			2 bytes	C	R	W	T	U	pulses	Low
16	Value sender 1 - ...	Input/Output 4bytes value			4 bytes	C	R	W	T	U	counter pulses (unsigned)	Low

Value sender

Number	Name	Object Function	Description	Group Address	Length	C	R	W	T	U	Data Type	Priority
16	Select 1 - ...	Input/Output 1byte value			1 byte	C	R	W	T	U	counter pulses (0..255)	Low

Select

Fig.6.1 “Sensor...x”communication object

NO.	Object Function	Name	Data Type	Flag	DPT
16	**Input ...**	**Binary or Sensor1 - {{}}**	**1bit** **1byte** **1byte** **2byte** **2byte** **2byte** **4byte** **4byte** **4byte**	**C,W,U**	**1.001 switch** **5.010 counter pulses (0..255)** **6.010 counter pulses (-128..127)** **7.001 pulses** **8.001 pulses difference** **9.*2-byte float value** **14.*4-byte float value** **12.001 counter pulses(unsigned)** **13.001 counter pulses(signed)**
The communication object is used to receive binary values or sensor values from the bus. The name in parentheses changes with the parameter “Function name”. If description is empty, display “Binary or sensor.1 - ” by default. The same below.					
16	**Input/Out ...**	**Value sender or Select 1 - {{}}**	**1bit** **1byte** **2byte** **4byte**	**C,W,T,U,R**	**1.001 switch** **5.010 counter pulses (0..255)** **7.001 pulses** **12.001 counter pulses(unsigned)**
This communication object is used to send output values to the bus.					

Table 6.1 “Sensor...x”communication object

6.2. “Lighting...x（x=1~200）”Communication Object

Number	Name	Object Function	Description	Group Address	Length	C	R	W	T	U	Data Type	Priority
116	Lighting 1 - ...	On/Off			1 bit	C	R	-	T	-	switch	Low
117	Lighting 1 - ...	On/Off, status			1 bit	C	-	W	T	U	switch	Low
118	Lighting 1 - ...	Brightness value			1 byte	C	R	-	T	-	percentage (0..100%)	Low
119	Lighting 1 - ...	Brightness, staus			1 byte	C	-	W	T	U	percentage (0..100%)	Low
120	Lighting 1 - ...	Colour temperature			2 bytes	C	R	-	T	-	absolute colour temperature (K)	Low
121	Lighting 1 - ...	Colour temperature, status			2 bytes	C	-	W	T	U	absolute colour temperature (K)	Low

Number	Name	Object Function	Description	Group Address	Length	C	R	W	T	U	Data Type	Priority
116	Lighting 1 - ...	On/Off			1 bit	C	R	-	T	-	switch	Low
117	Lighting 1 - ...	On/Off, status			1 bit	C	-	W	T	U	switch	Low
118	Lighting 1 - ...	Brightness value			1 byte	C	R	-	T	-	percentage (0..100%)	Low
119	Lighting 1 - ...	Brightness, staus			1 byte	C	-	W	T	U	percentage (0..100%)	Low
120	Lighting 1 - ...	RGB dimming value			3 bytes	C	R	-	T	-	RGB value 3x(0..255)	Low
121	Lighting 1 - ...	RGB brightness, status			3 bytes	C	-	W	T	U	RGB value 3x(0..255)	Low

Number	Name	Object Function	Description	Group Address	Length	C	R	W	T	U	Data Type	Priority
116	Lighting 1 - ...	On/Off			1 bit	C	R	-	T	-	switch	Low
117	Lighting 1 - ...	On/Off, status			1 bit	C	-	W	T	U	switch	Low
118	Lighting 1 - ...	Brightness value			1 byte	C	R	-	T	-	percentage (0..100%)	Low
119	Lighting 1 - ...	Brightness, staus			1 byte	C	-	W	T	U	percentage (0..100%)	Low
120	Lighting 1 - ...	RGBW dimming value			6 bytes	C	R	-	T	-	RGBW value 4x(0..100%)	Low
121	Lighting 1 - ...	RGBW brightness, status			6 bytes	C	-	W	T	U	RGBW value 4x(0..100%)	Low

Lighting

Number	Name	Object Function	Description	Group Address	Length	C	R	W	T	U	Data Type	Priority
116	Curtain 1 - ...	Open/Close			1 bit	C	R	-	T	-	open/close	Low
117	Curtain 1 - ...	Stop			1 bit	C	R	-	T	-	step	Low
118	Curtain 1 - ...	Postion			1 byte	C	R	-	T	-	percentage (0..100%)	Low
119	Curtain 1 - ...	Position, status			1 byte	C	-	W	T	U	percentage (0..100%)	Low
120	Curtain 1 - ...	Open/Close, status			1 bit	C	-	W	T	U	open/close	Low

curtain

Number	Name	Object Function	Description	Group Address	Length	C	R	W	T	U	Data Type	Priority
116	Scene 1-1 - ...	2byte float value			2 bytes	C	-	-	T	-	2-byte float value	Low
117	Scene 1-2 - ...	1bit value (<=6bit)			1 bit	C	-	-	T	-	switch	Low
118	Scene 1-3 - ...	1bit value (<=6bit)			1 bit	C	-	-	T	-	switch	Low
119	Scene 1-4 - ...	1bit value (<=6bit)			1 bit	C	-	-	T	-	switch	Low
120	Scene 1-5 - ...	1bit value (<=6bit)			1 bit	C	-	-	T	-	switch	Low
121	Scene 1-6 - ...	1bit value (<=6bit)			1 bit	C	-	-	T	-	switch	Low

scene

Number	Name	Object Function	Description	Group Address	Length	C	R	W	T	U	Data Type	Priority
116	Fan 1 - ...	On/Off			1 bit	C	R	-	T	-	switch	Low
117	Fan 1 - ...	On/Off, status			1 bit	C	-	W	T	U	switch	Low
118	Fan 1 - ...	Fan speed			1 byte	C	R	-	T	-	percentage (0..100%)	Low
119	Fan 1 - ...	Fan speed, status			1 byte	C	-	W	T	U	percentage (0..100%)	Low

fan

Number	Name	Object Function	Description	Group Address	Length	C	R	W	T	U	Data Type	Priority
116	Valve 1 - ...	On/Off			1 bit	C	R	-	T	-	switch	Low
117	Valve 1 - ...	On/Off, status			1 bit	C	-	W	T	U	switch	Low
118	Valve 1 - ...	Position			1 byte	C	R	-	T	-	percentage (0..100%)	Low
119	Valve 1 - ...	Position, status			1 byte	C	-	W	T	U	percentage (0..100%)	Low

valve

Fig.6.2 “Lighting...x”communication object

NO.	Object Function	Name	Data Type	Flag	DPT
116	**On/Off**	**Lighting1 - {{}} Fan1 - {{}} Valve1 - {{}}**	**1bit**	**C,R,T**	**1.001 switch**
117	**On/Off,status**	**Lighting1 - {{}} Fan1 - {{}} Valve1 - {{}}**	**1bit**	**C,W,T,U**	**1.001 switch**

These two communication objects apply to lights, fans and valves function. Telegrams:

0 — Off

1 — On

Obj.116: Used to send on/off telegrams to the bus to control the on/off state of lights, fans or valves.

Obj.117: Used to receive the on/off status responded by other bus devices, such as: the on/off status of lights, fans or valves.

The name in parentheses changes with the parameter “Function name”. If description is empty, display “Lighting1 - ” by default. The same below.

118	**Brightness value**	**Lighting1 - {{}}**	**1byte**	**C,R,T**	**5.001 percentage(0..100%)**
119	**Brightness,status**	**Lighting1 - {{}}**	**1byte**	**C,W,T,U**	**5.001 percentage(0..100%)**

These two communication objects apply to light function. Telegram: 0…100%

Obj.118: Used to send the dimming telegram to the bus, that is, sending the brightness value.

Obj.119: Used to receive status of brightness value from bus.

120	**Colour temperature**	**Lighting1 - {{}}**	**2byte**	**C,R,T**	**7.600 absolute color temperature**
121	**Colour temperature,status**	**Lighting1 - {{}}**	**2byte**	**C,W,T,U**	**7.600 absolute color temperature**

These two communication objects apply to colour temperature function. Telegram: 2000..7000 K

Obj.120: Used to send the control telegram of the colour temperature to the bus.

Obj.121: Used to receive colour temperature status from bus.

120	**RGB dimming value**	**Lighting1 - {{}}**	**3byte**	**C,R,T**	**232.600 RGB value 3x(0..255)**
121	**RGB brightness,status**	**Lighting1 - {{}}**	**3byte**	**C,W,T,U**	**232.600 RGB value 3x(0..255)**

These two communication objects apply to RGB function.

Obj.120: Used to send brightness value of RGB three-colour lamp to the bus.

Obj.121: Used to receive brightness telegram of RGB three-colour lamp from bus.

3-Byte Code for RGB Dimming Object Data Type: U8 U8 U8, as follows:

3_{MSB}	2	1_{LSB}
R	G	B
UUUUUUUU	UUUUUUUU	UUUUUUUU

R: red dimming value; G: green dimming value; B: blue dimming value.

120	**RGBW dimming value**	**Lighting1 - {{}}**	**6byte**	**C,R,T**	**251.600 DPT_Colour_RGBW**
121	**RGBW brightness,status**	**Lighting1 - {{}}**	**6byte**	**C,W,T,U**	**251.600 DPT_Colour_RGBW**

These two communication objects apply to RGBW function.

Obj.120: Used to send brightness value of RGBW four-colour lamp to the bus.

Obj.121: Used to receive brightness telegram of RGBW four-colour lamp from bus.

Encoding of the data type of the 6-byte RGBW dimming object: U8 U8 U8 U8 R8 R4 B4, as follows:

6_{MSB}	5	4	3	2	1_{LSB}
R	G	B	W	Reserve	r r r r mR mG mB mW
UUUUUUUU	UUUUUUUU	UUUUUUUU	UUUUUUUU	00000000	0000BBBB

R: red dimming value; G: green dimming value; B: blue dimming value; W: white dimming value;

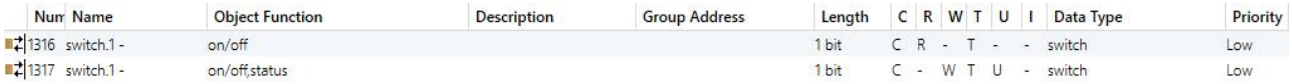
mR: determines whether the red dimming value is valid, 0 = invalid, 1 = valid;

mG: determines whether the green dimming value is valid, 0 = invalid, 1 = valid;

mB: determines whether the blue dimming value is valid, 0 = invalid, 1 = valid; mW: Determines whether the white dimming value is valid,0 = invalid,1 =valid.					
116	**Open/Close**	**Lighting1 - {{}}**	**1bit**	**C,R,T**	**1.008 up/down** **1.009 open/close**
120	**Open/Close,status**	**Lighting1 - {{}}**	**1bit**	**C,W,T,U**	**1.008 up/down** **1.009 open/close**
These two communication objects apply to curtain function. Telegrams: 1——close 0——open Obj.116: Used to send the telegram to the bus, to control curtain open/close. Obj.120: Used to receive the open/close status of the curtain responded by other bus devices.					
117	**Stop**	**Curtain1 - {{}}**	**1bit**	**C,R,T**	**1.007 step**
This communication object apply to curtain function, used to stop curtain movement. Telegrams: 1——stop					
118	**Position**	**Curtain1 - {{}}** **Valve1 - {{}}**	**1byte**	**C,R,T**	**5.001 percentage(0..100%)**
119	**Position,status**	**Curtain1 - {{}}** **Valve1 - {{}}**	**1byte**	**C,W,T,U**	**5.001 percentage(0..100%)**
These two communication objects apply to curtain and valve function. Telegrams: 0…100% Obj.118: Used to send a telegram to control the position of the curtain or valve to the bus. Obj.119: Used to receive a curtain position status in response to the actuator on the bus.					
116	**1 bit value(<=6bit)** **1byte value** **2byte value** **2byte float value**	**Scene1 - {{}}**	**1byte 1bit** **2byte** **2byte**	**C,R,T**	**1.001 switch** **5.001 percentage(0..100%)** **7.001 pulses** **9.001 temperature**
This communication object apply to scene function, used to send telegrams to the bus. The range of the telegram value depends on the selected object type.					

Table 6.2 “Lighting...x”communication object

6.3. “Switch...x（x=1~100）”Communication Object



Num	Name	Object Function	Description	Group Address	Length	C	R	W	T	U	I	Data Type	Priority
1316	switch.1 -	on/off			1 bit	C	R	-	T	-	-	switch	Low
1317	switch.1 -	on/off,status			1 bit	C	-	W	T	U	-	switch	Low

Fig.6.3 “Switch...x”communication object

NO.	Object Function	Name	Data Type	Flag	DPT
1316	**On/Off**	**Switch1 - {{}}**	**1bit**	**C,R,T**	**1.001 switch**
1317	**On/Off,status**	**Switch1 - {{}}**	**1bit**	**C,W,T,U**	**1.001 switch**

These two communication objects apply to switch function. Telegrams:

0——OFF

1——ON

Obj.1316: Used to send on/off telegrams to the bus to control on/off.

Obj.1317: Used to receive the on/off status responded by other bus devices.

The name in parentheses changes with the parameter “Function name”. If description is empty, display “Switch1 -” by default. The same below.

Table 6.3 “Switch...x”communication object

6.4. "AC...x（x=1~16）"Communication Object

Number	Name	Object Function	Description	Group Address	Length	C	R	W	T	U	Data Type	Priority
1516	AC 1 - ...	On/Off			1 bit	C	R	-	T	-	switch	Low
1517	AC 1 - ...	Current setpoint adjustment			2 bytes	C	R	-	T	-	temperature (°C)	Low
1518	AC 1 - ...	Room temperature			2 bytes	C	-	W	T	U	temperature (°C)	Low
1519	AC 1 - ...	Swing position			1 byte	C	R	-	T	-		Low
1520	AC 1 - ...	Control mode			1 byte	C	R	-	T	-	HVAC control mode	Low
1521	AC 1 - ...	Fan speed			1 byte	C	R	-	T	-	percentage (0..100%)	Low
1522	AC 1 - ...	Current temperature setpoint...			2 bytes	C	-	W	T	U	temperature (°C)	Low
1523	AC 1 - ...	Swing postion, status			1 byte	C	-	W	T	U		Low
1524	AC 1 - ...	Control mode, status			1 byte	C	-	W	T	U	HVAC control mode	Low
1525	AC 1 - ...	Fan speed, status			1 byte	C	-	W	T	U	percentage (0..100%)	Low
1526	AC 1 - ...	On/Off, status			1 bit	C	-	W	T	U	switch	Low

Fig.6.4 "AC...x"communication object

NO.	Object Function	Name	Data Type	Flag	DPT
1516	**On/Off**	**AC1 - {{}}**	**1bit**	**C,R,T**	**1.001 switch**
1526	**On/Off,status**	**AC1 - {{}}**	**1bit**	**C,W,T,U**	**1.001 switch**
These two communication objects apply to switch function. Telegrams: 0——OFF 1——ON Obj.1516: Used to send on/off telegrams to the bus to control on/off. Obj.1526: Used to receive the on/off status responded by other bus devices. The name in parentheses changes with the parameter "Function name". If description is empty, display "AC1 -" by default. The same below.					
1517	**Current setpoint temperature**	**AC1 - {{}}**	**2bytes**	**C,R,T**	**9.001 temperature**
1522	**Current setpoint temperature, status**	**AC1 - {{}}**	**2bytes**	**C,W,T,U**	**9.001 temperature**
These two communication objects apply to setpoint temperature function. The telegram value range depends on the parameter settings. Obj.1517: Used to send the actual setpoint temperature value of the air conditioner to the bus. Obj.1522: Used to receive the current setpoint temperature status fed back by the air conditioner on the bus.					

1518	**Room temperature**	**AC1 - {{}}**	**2bytes**	**C,W,T,U**	**9.001 temperature**
This communication object is used to receive the temperature measurement values sent by external temperature sensors on the bus and to periodically send read requests.					
1519	**Swing position**	**AC1 - {{}}**	**1byte**	**C,R,T**	**5.001 percentage**
1523	**Swing position,status**	**AC1 - {{}}**	**1byte**	**C,W,T,U**	**5.001 percentage**
These two communication objects apply to air swing function. The telegram value depends on the parameter settings. Obj.1519: Used to send telegrams to the bus to control the fan swing. Obj.1523: Used to receive the position status of the fan swing on the bus.					
1520	**Control mode**	**AC1 - {{}}**	**1byte**	**C,R,T**	**20.105 HVAC control mode**
1524	**Control mode,status**	**AC1 - {{}}**	**1byte**	**C,W,T,U**	**20.105 HVAC control mode**
These two communication objects apply to control mode function. The telegram value depends on the parameter settings. Obj.1520: Used to send control telegrams of each air conditioner mode to the bus. Obj.1524: Used to receive the current control mode on the bus.					
1521	**Fan speed**	**AC1 - {{}}**	**1byte**	**C,R,T**	**20.105 HVAC control mode**
1525	**Fan speed,status**	**AC1 - {{}}**	**1byte**	**C,W,T,U**	**20.105 HVAC control mode**
These two communication objects apply to fan speed function. The telegram value depends on the parameter settings. Obj.1521: Used to send control telegrams of each fan speed level to the bus. Obj.1525: Used to receive the current fan speed on the bus.					

Table 6.4 “AC...x”communication object

Chapter 7 Factory settings

The default parameter settings of the KNX-MQTT-HA gateway at delivery are as follows:

Physical address	15.15.254
Physical address of the IP tunneling connection	15.15.241 15.15.242 15.15.243 15.15.244 15.15.245
IP configuration	
IP address	192.168.2.200
Subnet mask	255.255.255.0
Default gateway	192.168.2.1

When the IP address needs to be reset, the device can be reset to the factory configuration.

The procedure to reset the device to the factory configuration is as follows:

Press the programming button and hold for 4 seconds then release, repeat the operation for 4 times, and the interval between each operation is less than 3 seconds.The LAN, KNX, and programming LED will turn off, and then the LAN and KNX LED will return to normal status and the device restarts. After the restart is completed, the device is restored to the factory settings.

For more information about the FDSK (Factory Default Setting Key), please refer to Chapter 5.8 .