

With KNX IP interface, bus powered

# **KNX Modbus Gateway TCP secure**

## Operation and installation manual



(Art. # PNx41-20001)

PEAKnx GmbH  
Leydheckerstraße 10  
DE-64293 Darmstadt

Tel.: +49 6151 / 279 18 25  
support@peaknx.com  
www.peaknx.com

# Content

<b>1</b>	<b>Application .....</b>	<b>4</b>
<b>2</b>	<b>Installation and connection .....</b>	<b>4</b>
2.1	KNX programming mode .....	5
2.2	Manual operation and status display .....	5
<b>3</b>	<b>Reset to factory default settings .....</b>	<b>6</b>
3.1	Factory default settings .....	7
<b>4</b>	<b>Wiring scheme .....</b>	<b>7</b>
<b>5</b>	<b>KNX Security .....</b>	<b>7</b>
<b>6</b>	<b>Interface settings in the ETS .....</b>	<b>8</b>
6.1	ETS 5 .....	8
6.2	ETS 6 .....	9
6.3	Common .....	10
<b>7</b>	<b>Programming.....</b>	<b>11</b>
7.1	Via the KNX bus .....	11
7.2	Via KNXnet/IP tunneling.....	11
<b>8</b>	<b>ETS database .....</b>	<b>12</b>
8.1	Secure commissioning .....	12
8.2	Secure group communication .....	14
8.3	Additional individual addresses .....	16
8.4	IP settings.....	17
8.5	Description.....	19
8.6	General settings .....	20
8.7	Modbus settings .....	22
8.8	Datapoints N – M .....	25
8.9	Channel function “DPT 01 – Binary – 1 bit” .....	30
8.10	Channel function “DPT 03 – Dimming – 4 bits” .....	34
8.11	Channel function “DPT 05 – Percent – 1 byte” .....	42
8.12	Channel function “DPT 05 – Configured – 1 byte” .....	43
8.13	Channel function “DPT 05 – Unsigned – 1 byte” .....	51
8.14	Channel function “DPT 06 – Signed – 1 byte” .....	53
8.15	Channel function “DPT 07 – Configured – 2 bytes” .....	54
8.16	Channel function “DPT 07 – Unsigned – 2 bytes” .....	62
8.17	Channel function “DPT 08 – Signed – 2 bytes” .....	64
8.18	Channel function “DPT 09 – Float – 2 bytes” .....	64
8.19	Channel function “DPT 12 – Configured – 4 bytes” .....	68
8.20	Channel function “DPT 12 – Unsigned – 4 bytes” .....	76
8.21	Channel function “DPT 13 – Signed – 4 bytes” .....	78
8.22	Channel function “DPT 14 – Float – 4 bytes” .....	80
8.23	Channel function “DPT 29 – Signed – 8 bytes” .....	83
8.24	Channel function “Fixed value – 1 bit” .....	84
8.25	Channel function “Fixed value – 2 bytes” .....	85
8.26	Converter N – M.....	86

8.27	Converter function “Binary” .....	90
8.28	Converter function “Threshold” .....	91
8.29	Converter function “Percent” .....	92
8.30	Converter function “Scaling” .....	93
8.31	General information.....	94
<b>9</b>	<b>Open source licenses .....</b>	<b>99</b>

# 1 Application

The KNX Modbus Gateway TCP *secure* is a compact gateway between KNX TP and modbus TCP with 250 freely configurable datapoints.

The device enables easy integration of modbus devices that support the TCP protocol via IP and can act as a modbus master or slave. As master the device can address up to 5 slave devices.

The assignment between KNX objects and modbus registers can be configured via parameters in the ETS without an additional tool.

Two buttons and three LEDs allow local operation and visualization of the device status.

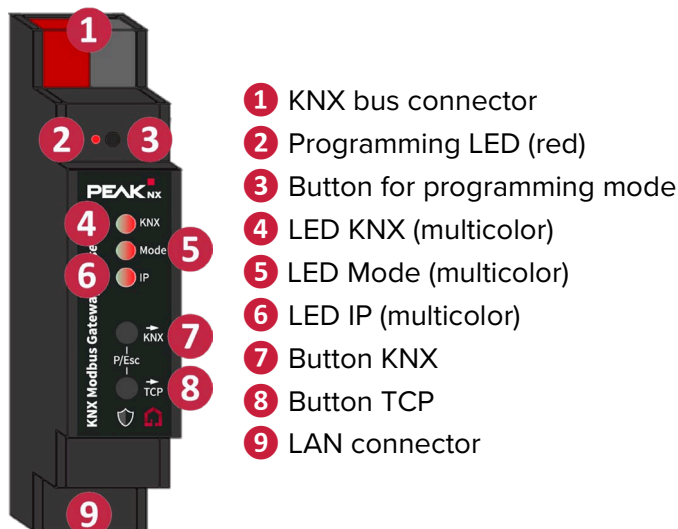
Furthermore, the device can be used as a programming interface to connect one or more PCs to the KNX bus (e.g. for ETS®).

Power is supplied via the KNX bus.

The device supports KNX Security.

# 2 Installation and connection

The KNX Modbus Gateway TCP *secure* is designed for installation on a DIN rail (35 mm) with a width of 1 unit (18 mm). It features the following controls and displays:



*If the bus voltage is missing, the device is without function.*

## 2.1 KNX programming mode

The KNX programming mode is activated/deactivated either by pressing the recessed KNX programming button **3** or by simultaneously pressing the buttons **7** and **8**.

When the programming mode is active, the programming LED **2** and the LED Mode **5** light up red.

The operation/visualization of the programming mode on the front can be activated/deactivated in the ETS® on page general settings.

## 2.2 Manual operation and status display

The LED KNX **4** lights up green when KNX bus voltage is present. When this LED flickers, telegram traffic is taking place on the KNX bus.

Errors in communication (e.g. telegram repetitions or telegram fragments) are indicated by a brief color change to red.

Summary of the states of the LED KNX **4**:

LED Status	Meaning
LED lights green	KNX bus voltage present.
LED flickers green	Telegram traffic on the KNX bus.
LED briefly red	Error in the communication on the KNX bus.

The LED IP **6** lights up when an ethernet link is active. This LED is green if the device has valid IP settings (IP address, subnet and gateway). This LED is red if the IP settings are invalid or non-existent. This is also the case, for example, if the device has not yet received the IP settings from the DHCP server. When this LED flickers, IP telegram traffic is taking place.

Furthermore, LED IP **6** lights up red if a static error has been configured in the ETS database. This is the case if “1 based” is parameterized in the Modbus settings and additionally address 0 is parameterized in at least one channel.

Errors in the Modbus communication (*gateway is Modbus master*) are indicated by a brief color change to red. These errors are:

- In the Modbus settings, the **Time till next cycle** set too short.  
Not all channels can be run through in the specified time.
- The Modbus gateway receives an exception response.  
But not the exception “Acknowledge” (code 0x05).
- The Modbus gateway does not receive a response.

Summary of the states of the LED IP 6:

LED Status	Meaning
LED lights green	The device has an active ethernet link and valid IP settings.
LED lights red	The device has an active ethernet link and invalid IP settings or has not yet received IP settings from the DHCP server. Incorrect configuration of the register addresses in the ETS database (static error).
LED flickers green	IP telegram traffic.
LED briefly red	Error in the Modbus communication.

The LED Mode 5 lights up or flashes when KNX bus voltage is present.

Pressing and holding the button KNX 7 triggers the synchronization of the KNX objects. This is indicated by the LED Mode 5 lighting up in orange.

Pressing and holding the button TCP 8 triggers the synchronization of the Modbus registers. This is indicated by the LED Mode 5 lighting up in orange.

Summary of the states of the LED Mode 5:

LED Status	Meaning
LED lights green	The device operates in normal operating mode.
LED lights red	The programming mode is active.
LED lights orange	The programming mode is not active. Synchronization is active.
LED flashes red	The programming mode is not active. The device is not loaded correctly, e.g. after aborting a download.
LED flashes green	The device is currently loaded by the ETS.

### 3 Reset to factory default settings

It is possible to reset the device to its factory default settings.

- Disconnect the KNX bus connector 1 from the device.
- Press the KNX programming button 3 and keep it pressed down.
- Reconnect the KNX bus connector 1 to the device.
- Keep the KNX programming button 3 pressed for at least another 6 seconds.
- A short flashing of all LEDs (2 4 5 6) visualizes the successful reset of the device to factory default settings.

In the factory default settings, the device has the individual address 15.15.255 and no group addresses are connected. The IP address assignment is done via DHCP. Also, KNX Data Security is disabled and the initial key (FDSK) must be used for secure commissioning.

## 3.1 Factory default settings

### Individual addresses and KNXnet/IP tunneling connections

Individual address: 15.15.255

Active KNXnet/IP tunneling connections: 1

Individual address of the tunneling connection: 15.15.240

## 4 Wiring scheme



## 5 KNX Security

The KNX standard was extended by KNX Security to protect KNX installations from unauthorized access. KNX Security reliably prevents the monitoring of communication as well as the manipulation of the system.

The specification for KNX Security distinguishes between KNX IP Security and KNX Data Security. KNX IP Security protects the communication over IP while on KNX TP the communication remains unencrypted. Thus, KNX IP Security can also be used in existing KNX systems and with non-secure KNX TP devices.

KNX Data Security describes the encryption on telegram level. This means that the telegrams on the twisted pair bus or via RF (radio frequency) are also encrypted.



*Encrypted telegrams are longer than the previously used unencrypted ones. For secure programming via the bus, it is therefore necessary that the interface used (e.g. USB) and any intermediate line couplers support the so called KNX long frames.*



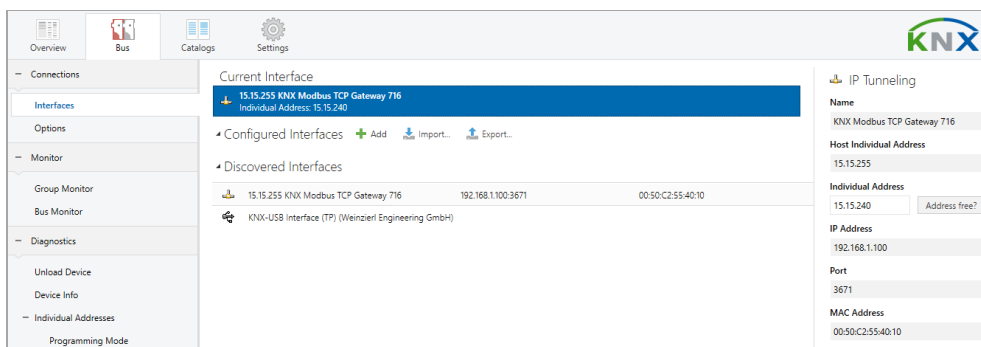
*KNX IP Security only refers to the KNX communication via IP. The Modbus communication remains unencrypted.*

- KNX TP – Communication encrypted (KNX Data Security)
- KNX IP – Communication encrypted (KNX IP Security)  
The entire KNX communication on application and management level is encrypted.
- Modbus TCP – Communication unencrypted  
Theoretically, the modbus communication could be intercepted and manipulated.

## 6 Interface settings in the ETS

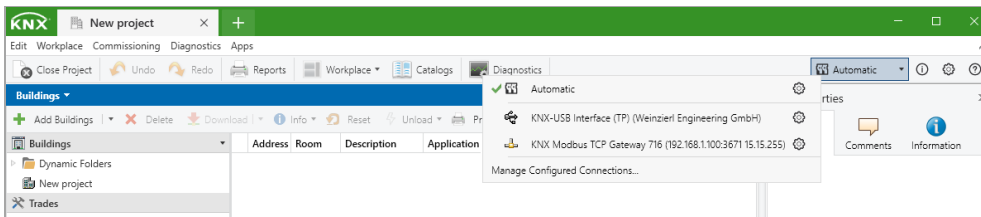
### 6.1 ETS 5

In the ETS 5, interfaces can be selected and configured via the ETS menu “Bus – Interfaces”. All available connections are listed under “Discovered Interfaces”. After clicking on the desired connection, connection specific information and options appear on the right side of the ETS window. The selected connection can be selected as the “Current Interface” via the “Select” button.



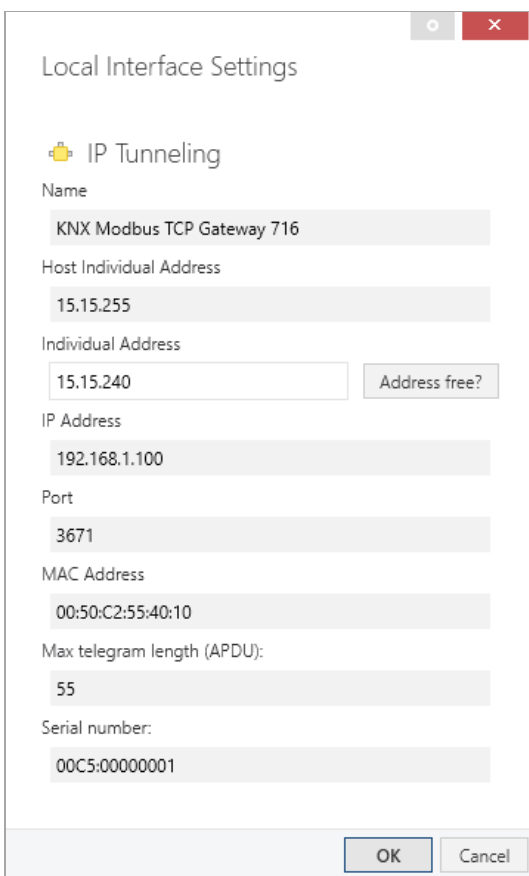
## 6.2 ETS 6

In the ETS 6, interfaces can be selected and configured in the ETS project via the “Interface” button. All available connections are listed here.



By clicking on a connection, this is selected as the desired interface.

By clicking the gear next to the desired connection, the connection specific information and options appear.



The 'Local Interface Settings' dialog box is shown, displaying configuration details for the 'KNX Modbus TCP Gateway 716' connection. The settings include:

- Name:** KNX Modbus TCP Gateway 716
- Host Individual Address:** 15.15.255
- Individual Address:** 15.15.240 (with a button 'Address free?')
- IP Address:** 192.168.1.100
- Port:** 3671
- MAC Address:** 00:50:C2:55:40:10
- Max telegram length (APDU):** 55
- Serial number:** 00C5:00000001

At the bottom of the dialog, there are 'OK' and 'Cancel' buttons.

## 6.3 Common

The displayed device name and the “Host Individual Address” (individual address of the device) can then be changed via the database entry within your ETS project.

In the “Individual address” section, the individual KNX address of the currently used KNXnet/IP tunneling connection can be changed. In order to check whether the desired individual address is not already available in your KNX installation, the button “Address free?” can be pressed.

The KNX Modbus Gateway TCP secure supports up to 8 connections simultaneously. A separate individual address is used for each connection.

The ETS can access configured IP interfaces even without a database entry. If the configuration does not correspond to the conditions of the installation, it must be adapted via the database entry within your ETS project.

The individual KNX device address as well as the individual KNX addresses for the additional tunneling connections can be changed via the database entry within the ETS project after the device has been added to the project. See section “ETS database – Additional individual addresses”.

Like all programmable KNX devices, the Das KNX Modbus Gateway TCP secure has a individual address with which the device can be addressed. This is used, for example, by the ETS when downloading the interface via the KNX bus.

For the interface function, the device uses additional individual addresses that can be set in the ETS (for ETS5.7 or newer). If a client (e.g. ETS) sends telegrams to the KNX bus via the Das KNX Modbus Gateway TCP secure, these contain one of the additional addresses as the sender address. Each address is assigned to a connection. In this way, response telegrams can be forwarded clearly to the respective client.

The additional individual addresses must be from the address range of the bus line in which the interface is located and must not be used by another device.

### Example:

<i>Individual address:</i>	<i>1.1.10 (device address in the topology)</i>
<i>KNXnet/IP tunneling connection 1:</i>	<i>1.1.240 (1. additional individual address)</i>
<i>KNXnet/IP tunneling connection 2:</i>	<i>1.1.241 (2. additional individual address)</i>
<i>KNXnet/IP tunneling connection 3:</i>	<i>1.1.242 (3. additional individual address)</i>
<i>KNXnet/IP tunneling connection 4:</i>	<i>1.1.243 (4. additional individual address)</i>
<i>KNXnet/IP tunneling connection 5:</i>	<i>1.1.244 (5. additional individual address)</i>
<i>KNXnet/IP tunneling connection 6:</i>	<i>1.1.245 (6. additional individual address)</i>
<i>KNXnet/IP tunneling connection 7:</i>	<i>1.1.246 (7. additional individual address)</i>
<i>KNXnet/IP tunneling connection 8:</i>	<i>1.1.247 (8. additional individual address)</i>

## 7 Programming

The Das KNX Modbus Gateway TCP secure can be programmed by the ETS in various ways.

### 7.1 Via the KNX bus

For this, the device must only be connected to the KNX bus. The ETS requires an additional interface (e.g. USB) to the KNX bus. This way, both the individual address and the entire application including IP configuration can be programmed. Programming via the KNX bus is recommended if no IP connection can be established.

### 7.2 Via KNXnet/IP tunneling

No additional interface is required here. Programming via KNXnet/IP tunneling is possible if the device already has a valid IP configuration (e.g. via DHCP). In this case, the device is displayed in the interfaces in the ETS and must be selected. The download takes place from the ETS project, as with other devices.

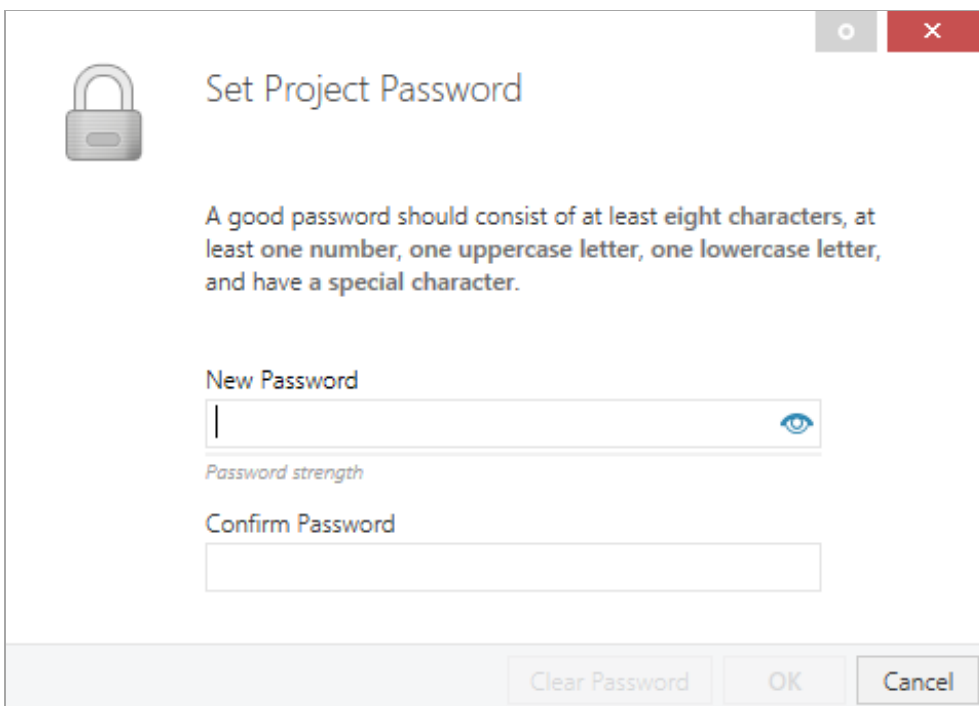
## 8 ETS database

The ETS5 database (for ETS 5.7 or newer) can be downloaded from the product website of KNX Modbus Gateway TCP secure ([www.peaknx.com](http://www.peaknx.com)) or from the ETS online catalogue.

The KNX Modbus Gateway TCP secure supports KNX Data Security to protect the device against unauthorized access from the KNX bus. If the device is programmed via the KNX bus, this is done with encrypted telegrams.

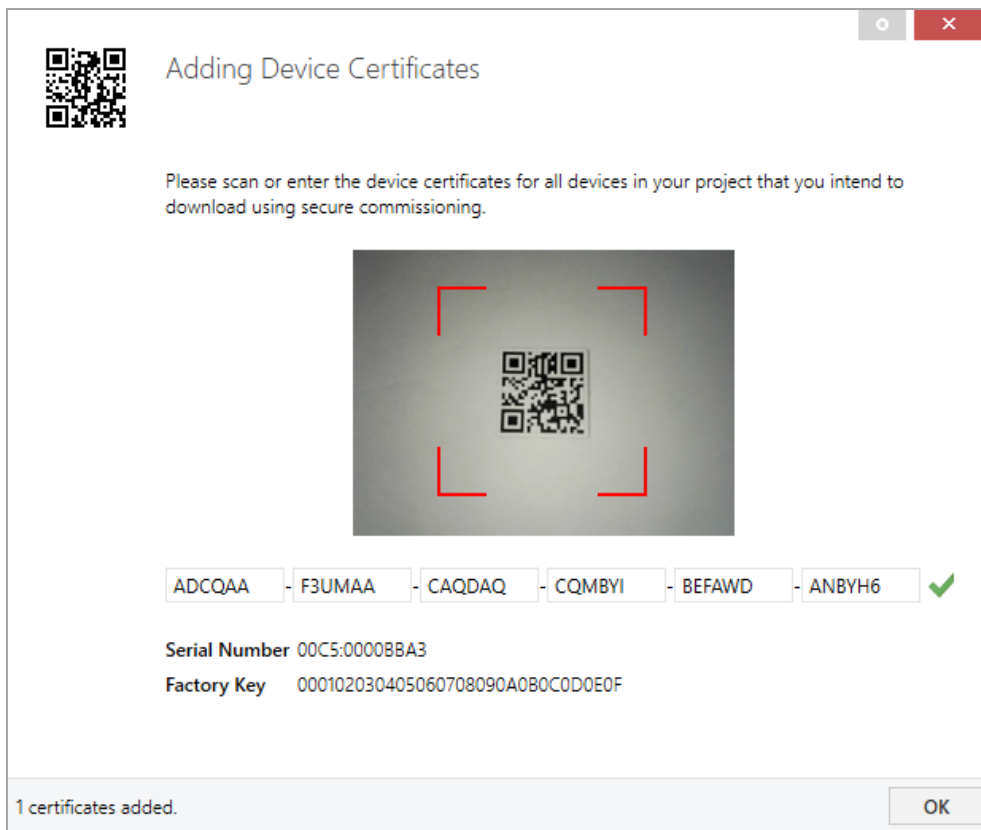
### 8.1 Secure commissioning

If the first product is inserted into a project with KNX Security, the ETS prompts you to enter a project password.



This password protects the ETS project from unauthorized access. This password is not a key that is used for KNX communication. The entry of the password can be bypassed with “Cancel”, but this is not recommended for security reasons.

ETS requires a device certificate for each device with KNX Security that is created in the ETS. This certificate contains the serial number of the device as well as an initial key (FDSK = Factory Default Setup Key).



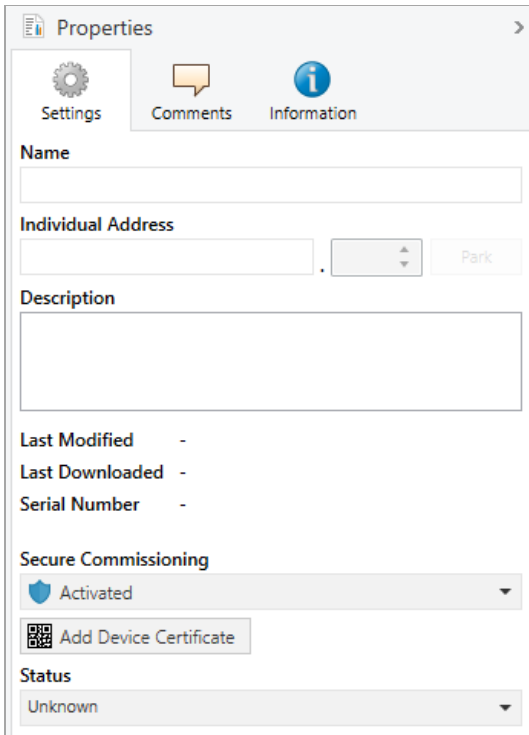
The certificate is printed as text on the device. It can also be scanned from the printed QR code via a webcam.

The list of all device certificates can be managed in the ETS panel Reports – Project Security.

This initial key is required to safely put a device into operation from the start. Even if the ETS download is recorded by a third party, the third party has no access to the secured devices afterwards. During the first secure download, the initial key is replaced by the ETS with a new key that is generated individually for each device. This prevents persons or devices who may know the initial key from accessing the device. The initial key is reactivated after a reset to factory default settings.

The serial number in the certificate enables the ETS to assign the correct key to a device during a download.

In the ETS project in the properties of the device, secure commissioning can be activated and the device certificate can be added:



**Properties**

Settings | Comments | Information

**Name**

**Individual Address**

**Description**

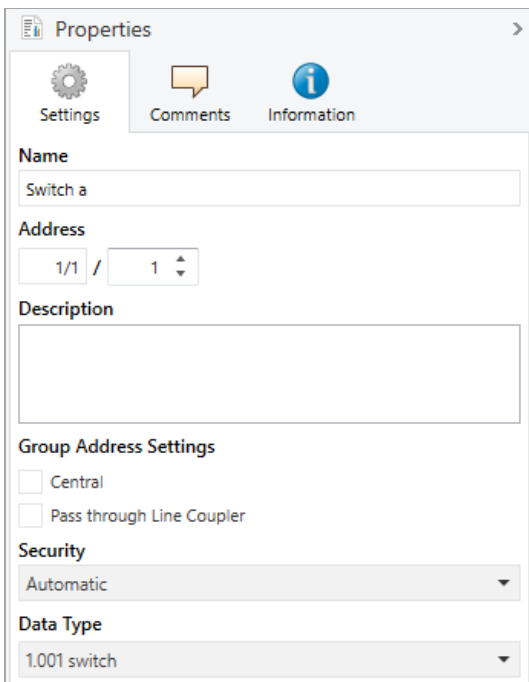
**Last Modified** -  
**Last Downloaded** -  
**Serial Number** -

**Secure Commissioning**  
 Activated

**Status**  
 Unknown

## 8.2 Secure group communication

Each object of the device can communicate either encrypted or unencrypted. The encryption is set under “Security” in the properties of the used group address:



**Properties**

Settings | Comments | Information

**Name**  
 Switch a

**Address**  
 1/1 / 1

**Description**





**Group Address Settings**  
☐ Central  
☐ Pass through Line Coupler

**Security**  
 Automatic

**Data Type**  
 1.001 switch

The setting “Automatic” activates encryption if both objects to be connected can communicate encrypted. Otherwise encrypted communication between the objects is not possible.

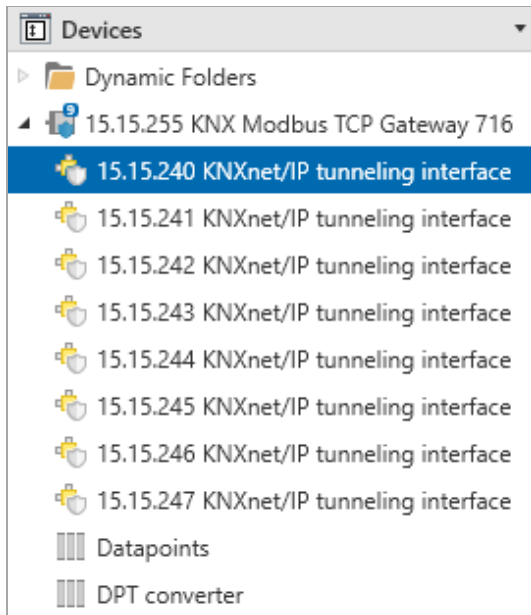
In the overview of communication objects in the ETS project, secured objects can be recognized by a shield symbol:

	Security	Number ^	Name	Object Function	Description	Group Address
		11	Button A0: Object a	Switch	Switch a	1/1/1
		12	Button A0: Object b	Switch	Switch b	1/1/2
		21	Button A1: Object a	Switch	Switch a	1/1/1
		22	Button A1: Object b	Switch	Switch b	1/1/2

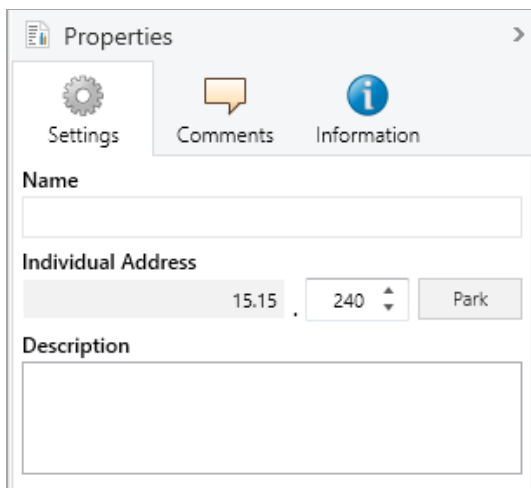
A separate key is automatically generated by the ETS for each secured group address. These keys can also be checked in the ETS panel Reports – Project Security. To enable all devices to communicate with a secure group address, the keys must be known to all. Therefore a download must be made into all devices that use this group address when a key is created or changed. A key is changed by the ETS e.g. when the encryption of a group address is switched off and on again.

## 8.3 Additional individual addresses

The additional individual addresses appear in the topology view.



To change the individual addresses, select the corresponding entry in the list and enter the desired address in the text field under “Properties – Settings”. If the frame of the text field changes its colour to red after entry, this indicates that the address entered is already being used. The changes are only applied in the device after download.



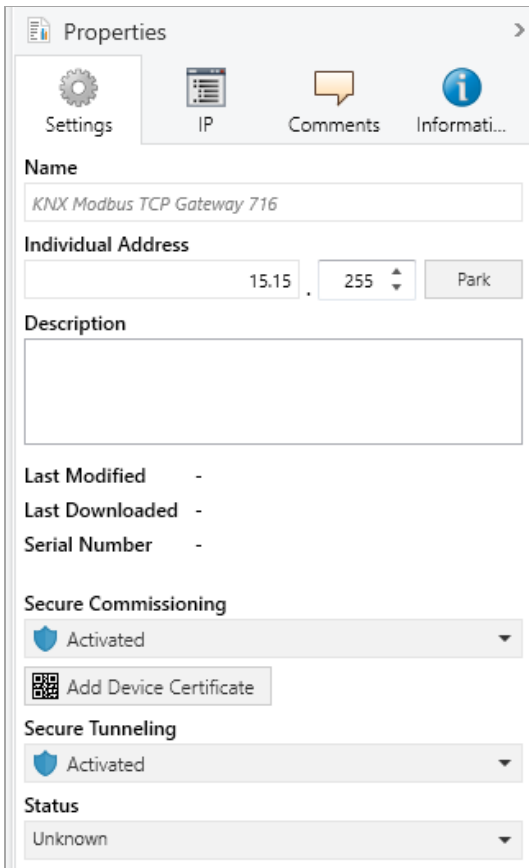
*Make sure that none of the specified addresses are already used in your KNX installation.*

## 8.4 IP settings

By marking the KNX Modbus Gateway TCP secure in the tree structure of the topology view of the ETS project, the overview “Properties” appears on the right side of the ETS window.

### 8.4.1 Device name (user friendly name)

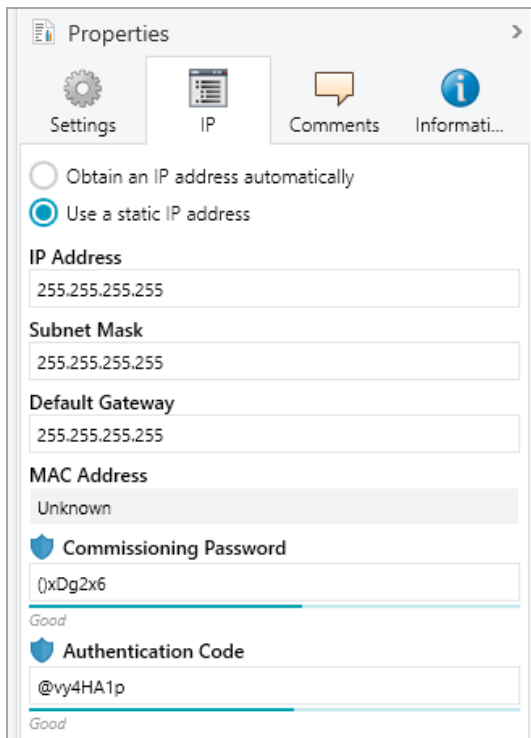
Under Properties menu item “Settings”, the device name (user friendly name) of the KNX Modbus Gateway TCP secure can be changed. The first 30 characters are used.




*The change made will only take effect after an ETS download.*

## 8.4.2 IP configuration

Under Properties menu item “IP”, the IP specific options of the KNX Modbus Gateway TCP secure can be changed.




*The change made will only take effect after an ETS download.*

By switching from “Obtain an IP address automatically” (via DHCP) to “Use a static IP address” (static IP address), the IP address, subnet mask and standard gateway can be freely selected.

### IP Address

The IP address of the KNX Modbus Gateway TCP secure must be entered here. This is used to address the device via the IP network (Wi-Fi). The IP addressing should be coordinated with the administrator of the network.

### Subnet Mask

The subnet mask must be entered here. This mask is used by the device to determine whether a communication partner is located in the local network. If a partner is not in the local network, the device does not send the telegrams directly to the partner, but to the standard gateway, which takes over the forwarding.

### Default Gateway

Enter the IP address of the gateway here, e.g. the access point of the installation.

## Example for the assignment of IP addresses

The KNX Modbus Gateway TCP secure is to be accessed with a PC.

*IP address of PC: 192.168.1.30*

*Subnet mask of PC: 255.255.255.0*

The KNX Modbus Gateway TCP secure is located in the same local network, i.e. it uses the same subnet. The assignment of the IP address is restricted by the subnet, i.e. in this example the IP address of the IP interface must be 192.168.1.xx, xx can be a number from 1 to 254 (with the exception of 30, which has already been used). Care must be taken not to assign addresses twice.

*IP address of KNX Modbus Gateway TCP secure: 192.168.1.31*

*Subnet mask of KNX Modbus Gateway TCP secure: 255.255.255.0*

## 8.5 Description

KNX Modbus Gateway TCP secure > Description

Description

General settings


Modbus settings

+ Datapoints

+ DPT converter

KNX Modbus Gateway TCP secure

With KNX IP interface, bus powered



The KNX Modbus Gateway TCP secure is a compact gateway between KNX TP and modbus TCP with 250 freely configurable datapoints.

The device enables easy integration of modbus devices that support the TCP protocol via IP and can act as a modbus master or slave. As master the device can address up to 5 slave devices.

The assignment between KNX objects and modbus registers can be configured via parameters in the ETS without an additional tool.


Two buttons and three LEDs allow local operation and visualization of the device status.

Furthermore, the device can be used as a programming interface to connect one or more PCs to the KNX bus (e.g. for ETS®).

Power is supplied via the KNX bus.

The device supports KNX Security.

Wiring scheme:



Please consult device data sheet and manual for further information.

This page shows the device description and the corresponding connection diagram.

## 8.6 General settings

KNX Modbus TCP Gateway 716 > General settings	
Description	<i>i</i> For device name and IP settings see dialog "Properties"
General settings	<i>i</i> For configuration of the database a DCA is available
Modbus settings	Send delay after bus power return: 5 s
+ Datapoints	Prog. mode on device front: <input type="radio"/> Disabled <input checked="" type="radio"/> Enabled
+ DPT converter	Manual operation (sync) on device: <input type="radio"/> Disabled <input checked="" type="radio"/> Enabled
	Heartbeat: <input type="radio"/> Disabled <input checked="" type="radio"/> Enabled
	Cycle time: 5 min
	KNX settings
	Telegram rate limitation: 1.0 s

### Send delay after bus power return

The send delay of telegrams after the return of the bus voltage can be set via this parameter. In this case, telegrams from the device are sent to the KNX bus in a delayed manner by the set time. This reduces the bus load when the bus voltage returns. Other functions such as receiving telegrams are not affected by this parameter.

### Prog. mode on device front

In addition to the normal programming button **3** the device allows activating the programming mode on the device front without opening the switchboard cover. The programming mode can be activated and deactivated via pressing simultaneously both buttons **7** and **8**.

This feature can be enabled and disabled via the parameter Prog. mode on device front. The recessed programming button **3** (next to the programming LED **2**) is always enabled and is not affected by this parameter.

## Manual operation (sync) on device

This parameter is used to configure manual operation on the device. Manual operation can be activated or deactivated.

Manual operation enables synchronization of all channels in the direction to KNX (button KNX **7**) and in the direction to Modbus (button TCP **8**).

Direction KNX:

All datapoints of the channels configured as “Modbus to KNX” send their current value on the KNX bus.

Direction Modbus:

If the gateway is operating as Modbus master, all registers of the channels configured “KNX to modbus” are written to Modbus again.

Synchronization can be cancelled by pressing the keys **7** and **8** simultaneously.

## Heartbeat

Cyclic sending of values to the KNX bus, to indicate that the device is operational. For the cycle time values between 1 min and 24 h are selectable.

Group object	Type KNX	Size	Direction
GO 456 Heartbeat – Trigger	1.001	1 bit	To KNX

## Telegram rate limitation

With this parameter the telegram rate limitation can be activated and the time between telegrams can be configured. Times between 0.1 s and 1.0 s can be selected.



*The telegram rate limitation only occurs when the bus load is increased.*

## 8.7 Modbus settings

--.- KNX Modbus TCP Gateway 716 secure > Modbus settings

Description	KNX Gateway	<input checked="" type="radio"/> Modbus master <input type="radio"/> Modbus slave
General settings	Byte order (data in word register)	<input checked="" type="radio"/> MSB first <input type="radio"/> LSB first
Modbus settings	Register address	<input checked="" type="radio"/> 0 based <input type="radio"/> 1 based
+ Datapoints	Slave count	2
+ DPT converter	Slave 1	
	Name	
	IP address	255.255.255.255
	IP port	502
	Slave address	Disabled
	Slave 2	
	Name	
	IP address	255.255.255.255
	IP port	502
	Slave address	Disabled
	Request settings	
	Time till next request	Automatic
	Time till next cycle	Automatic
	Multi read requests	<input checked="" type="radio"/> Disabled <input type="radio"/> Enabled
	Bit register write requests	<input checked="" type="radio"/> Write single coil - 05 <input type="radio"/> Write multi coils - 15
	Word register write requests	<input checked="" type="radio"/> Write single holding register - 06 <input type="radio"/> Write multi holding registers - 16
	Diagnostics settings	
	Diagnostics objects	<input checked="" type="radio"/> Disabled <input type="radio"/> Enabled

### KNX Gateway

This parameter defines the role of the KNX gateway within the client/server architecture of the Modbus environment. Available are:

- Modbus master
- Modbus slave

### Byte order (data in word register)

Defines the order in which 2 byte values are transmitted:

- MSB first (*high byte is sent first*)
- LSB first (*low byte is sent first*)

### Register address

Here it is set on which address basis the register address is defined.

- 0 based
- 1 based

### **Slave count** *(only in master mode)*

The number of Modbus communication partners is entered here. 1 to 5 slaves can be configured.

### **Name** *(30 characters, only in master mode)*

Any name can be assigned to a slave. However, this name should be unique and meaningful. This facilitates the work in the ETS project later, since the assigned name is displayed for the respective slave. This applies to the diagnostic objects as well as to the assignment of the channel pages. If no name is assigned, the designation "Slave 1 – 5" is used.

### **IP address** *(only in master mode)*

The IP address of the communication partner (slave) must be entered here.

### **IP port**

Modbus master:

The IP port of the communication partner (slave) must be entered here.

Modbus slave:

Here the own used IP port is to be entered. Port 502 is reserved for Modbus and is always active. If e.g. port 501 is parameterized here, the gateway uses ports 501 and 502.


### **Slave address** *(only in master mode)*

The slave address (0 ... 247) can be set here. This is required if the communication partner is a TCP/RTU gateway in order to address the slaves on the RTU side.

### **Time till next request** *(only in master mode)*

This parameter defines the minimum time to the next request. Serves to brake the master.

### **Time till next cycle** *(only in master mode)*

This parameter defines the time after which a new request cycle should start. If the time is too short, the cycle will not be completed. This is indicated by the LED IP  flashing red.

### **Multi read requests** *(only in master mode)*

When this parameter is activated, Modbus registers that are located one after the other in the channels are combined in a multi read request.



*The respective slave settings and the function code must be identical. The register address must be continuous, but repetitions may occur. A maximum of 16 channels can be combined.*

### Example:

*Slave for this page: Slave 1*

*Channel 1 – MB to KNX – Read Holding Register – Address 0*

*Channel 2 – MB to KNX – Read Holding Register – Address 1*

*Channel 3 – MB to KNX – Read Holding Register – Address 2*

*Channel 4 – MB to KNX – Read Holding Register – Address 2*

*Channel 5 – MB to KNX – Read Holding Register – Address 3*

*Channel 6 – MB to KNX – Read Holding Register – Address 4*

*Channel 7 – MB to KNX – Read Holding Register – Address 4*

The result is a read-multi-holding-register request for addresses 0 – 4.

### Bit register write requests *(only in master mode)*

Here the function code for writing bit registers is set.

- Write single coil – 05
- Write multi coils – 15

### Word register write requests *(only in master mode)*

Here the function code for writing word registers is set.

- Write single holding register – 06
- Write multi holding registers – 16

### Diagnostic objects

Activated, objects for diagnostic are displayed and provide information about the communication with the respective slave device.

Modbus master:

If no response is received from the slave device, “No communication – On” is sent to the KNX bus via the corresponding diagnostic object.

Group object	Type KNX	Size	Direction
GO 251 Diagnostic: Slave 1 – No communication	1.001	1 bit	To KNX
GO 252 Diagnostic: Slave 2 – No communication	1.001	1 bit	To KNX
GO 253 Diagnostic: Slave 3 – No communication	1.001	1 bit	To KNX
GO 254 Diagnostic: Slave 4 – No communication	1.001	1 bit	To KNX
GO 255 Diagnostic: Slave 5 – No communication	1.001	1 bit	To KNX

Modbus slave:

If no request is received from the master within the time interval, “No communication – On” is sent to the KNX bus via the diagnostic object.

Group object	Type KNX	Size	Direction
GO 251 Diagnostic: Slave – No communication	1.001	1 bit	To KNX

## 8.8 Datapoints N – M

--- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	<input type="text"/>
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
Datapoints	Channel 1	
	Datapoint type	Disabled
	Channel 2	
	Datapoint type	<div>Disabled ✓</div> <div>DPT 01 - Binary - 1 bit</div> <div>DPT 03 - Dimming - 4 bits</div> <div>DPT 05 - Percent - 1 byte</div> <div>DPT 05 - Configured - 1 byte</div> <div>DPT 05 - Unsigned - 1 byte</div> <div>DPT 06 - Signed - 1 byte</div> <div>DPT 07 - Configured - 2 bytes</div> <div>DPT 07 - Unsigned - 2 bytes</div> <div>DPT 08 - Signed - 2 bytes</div> <div>DPT 09 - Float - 2 bytes</div> <div>DPT 12 - Configured - 4 bytes</div> <div>DPT 12 - Unsigned - 4 bytes</div> <div>DPT 13 - Signed - 4 bytes</div> <div>DPT 14 - Float - 4 bytes</div> <div>DPT 29 - Signed - 8 bytes</div> <div>Fixed value - 1 bit</div> <div>Fixed value - 2 bytes</div>
	Channel 3	
	Datapoint type	
	Channel 4	
	Datapoint type	
	Channel 5	
	Datapoint type	
Channel 6		
Datapoint type		
	Channel 7	
	Datapoint type	Disabled
	Channel 8	
	Datapoint type	Disabled
	Channel 9	
	Datapoint type	Disabled
	Channel 10	
	Datapoint type	Disabled

10 channels are combined per page.

### Description for this page (30 characters)

Any name can be assigned to the parameter page. This facilitates the work in the ETS project. If no name is assigned, the channel page is designated e.g. with “Datapoints 1 – 10”.

### Slave for this page (only in master mode)

This parameter determines to which slave the configured channels of this page are to be assigned. The selection consists of the communication partners active in **Slave count** (1 – 5).

## Datapoint type

This parameter activates and defines the KNX interface and the function of this channel. The following options are available:

- Disabled
- DPT 01 – Binary – 1 bit

Group object	Type KNX	Size	Direction
GO 1 Channel 1: ... – Switch – 1 bit	1.001	1 bit	...

- DPT 03 – Dimming – 4 bits

Group object	Type KNX	Size	Direction
GO 1 Channel 1: ... – Dimming – 4 bits	3.007	4 bits	...

- DPT 05 – Percent – 1 byte

Group object	Type KNX	Size	Direction
GO 1 Channel 1: ... – Percent – 1 byte	5.001	1 byte	...

- DPT 05 – Configured – 1 byte

Group object	Type KNX	Size	Direction
GO 1 Channel 1: ... – Configured – 1 byte	5.010	1 byte	...

- DPT 05 – Unsigned – 1 byte

Group object	Type KNX	Size	Direction
GO 1 Channel 1: ... – Unsigned – 1 byte	5.010	1 byte	...

- DPT 06 – Signed – 1 byte

Group object	Type KNX	Size	Direction
GO 1 Channel 1: ... – Signed – 1 byte	6.010	1 byte	...

- DPT 07 – Configured – 2 bytes

Group object	Type KNX	Size	Direction
GO 1 Channel 1: ... – Configured – 2 bytes	7.001	2 bytes	...

- DPT 07 – Unsigned – 2 bytes

Group object	Type KNX	Size	Direction
GO 1 Channel 1: ... – Unsigned – 2 bytes	7.001	2 bytes	...

- DPT 08 – Signed – 2 bytes

Group object	Type KNX	Size	Direction
GO 1 Channel 1: ... – Signed – 2 bytes	8.001	2 bytes	...

- DPT 09 – Float – 2 bytes

Group object	Type KNX	Size	Direction
GO 1 Channel 1: ... – Float – 2 bytes	9.001	2 bytes	...

- DPT 12 – Configured – 4 bytes

Group object	Type KNX	Size	Direction
GO 1 Channel 1: ... – Configured – 4 bytes	12.001	4 bytes	...

- DPT 12 – Unsigned – 4 bytes

Group object	Type KNX	Size	Direction
GO 1 Channel 1: ... – Unsigned – 4 bytes	12.001	4 bytes	...

- DPT 13 – Signed – 4 bytes

Group object	Type KNX	Size	Direction
GO 1 Channel 1: ... – Signed – 4 bytes	13.001	4 bytes	...

- DPT 14 – Float – 4 bytes

Group object	Type KNX	Size	Direction
GO 1 Channel 1: ... – Float – 4 bytes	14.000	4 bytes	...

- DPT 29 – Signed – 8 bytes

Group object	Type KNX	Size	Direction
GO 1 Channel 1: ... – Signed – 8 bytes	29.010	8 bytes	...

- Fixed value – 1 bit
- Fixed value – 2 bytes

### Description (30 characters)

Any name can be assigned to the channel. However, this should be unique and meaningful, this facilitates later work with the associated group objects, since the assigned name is displayed there as a designation. If no name is assigned, the group objects are designated with “Channel N: ...”.

### Direction *(only for channels with DPT)*

The KNX specific communication direction is set here:

- KNX to modbus *(group object is input)*

Group object	Type KNX	Size	Direction
GO 1 Channel 1: Input – ...	Dependent on DPT		From KNX

- Modbus to KNX *(group object is output)*

Group object	Type KNX	Size	Direction
GO 1 Channel 1: Output – ...	Dependent on DPT		To KNX

### Send condition *(only for channels with DPT)*

If the group object is defined as an output, it is parameterised here when the object sends the value to the KNX bus. The following are available for selection:

- Read only  
Object sends only on read requests
- On change  
Object sends on value change
- Cyclically  
Object sends after cycle time
- On change and cyclically  
Object sends after cycle time and on value change

### Cycle time *(only for channels with DPT)*

The time of the cyclic **Send condition**.

### Type

This parameter defines the function of the channel and the size of the modbus register used.

Depending on the parameter **Datapoint type N**, various channel functions are possible here, which are described in more detail in the following chapters.

For “Value in word register” it should be noted that if the channel is misconfigured, it will not work.



**Bit count** and **Offset from right** must not be greater than 16 together.  
The value must fit into **Bit count**, e.g. **Bit count** = 1 → “Value” = 0 or 1.

## Function

Here the modbus function code for this channel is parameterized.

Depending on **KNX Gateway** (Modbus master/slave), the **Direction** and the **Type**, different function codes can be configured.

Word register:

Modbus master | KNX to modbus

Set via **Word register write requests**.

- Write single holding register – 06
- Write multi holding registers – 16

Modbus master | Modbus to KNX

- Read holding registers – 03
- Read input registers – 04

Modbus slave | KNX to modbus

- Read holding registers – 03
- Read input registers – 04

Modbus slave | Modbus to KNX

- Write single/multi holding registers – 06, 16

Bit register:

Modbus master | KNX to modbus

Set via **Bit register write requests**.

- Write single coil – 05
- Write multi coils – 15

Modbus master | Modbus to KNX

- Read coils – 01
- Read discrete inputs – 02

Modbus slave | KNX to modbus

- Read coils – 01
- Read discrete inputs – 02

Modbus slave | Modbus to KNX

- Write single/multi coils – 05, 15

## Address

The address of the modbus register is parameterized here. An address range of 0 ... 65535 is available.



*If the address 0 is parameterized for “1 based”, this is a static error, which deactivates the channel function and is indicated by the LED IP 6 lighting up red.*

### Polling interval *(only in master mode with Modbus to KNX)*

The cyclic intervals at which read requests are to be made for the respective register are defined here. The following options are available:

- Every cycle
- Every second cycle
- Every fourth cycle
- Every sixth cycle
- Every eighth cycle

### Write interval *(only in master mode for channels without DPT)*

Here it is defined when the fixed value is written to Modbus. The following options are available:

- Once on startup
  - On startup and cyclically
- A cycle time of 1 min to 24 h can be parameterized.

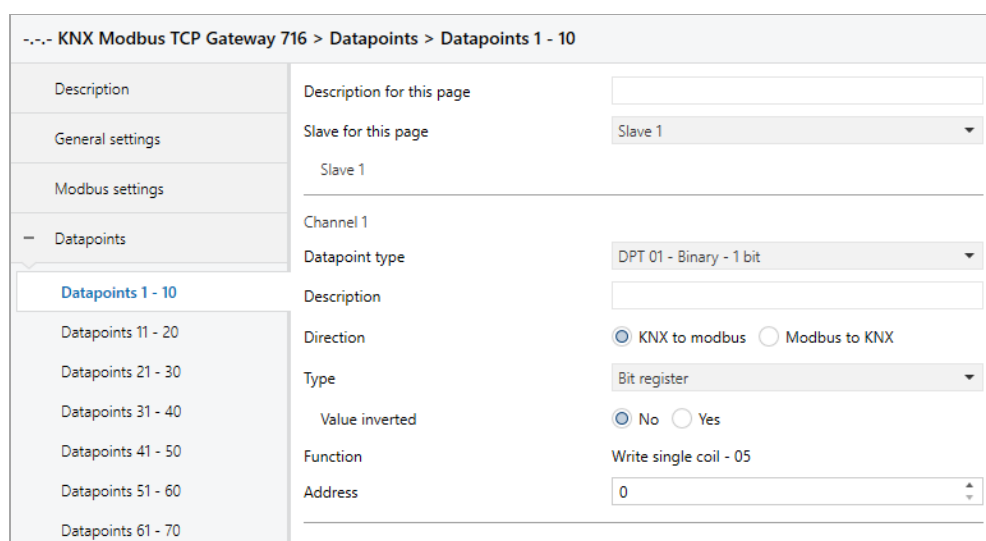
## 8.9 Channel function “DPT 01 – Binary – 1 bit”

### Type

The following types can be configured:

- Bit register  
1 bit (KNX) sets bit register (modbus)
- Bit in word register  
1 bit (KNX) sets 1 bit in word register (modbus)
- Value in word register  
1 bit (KNX) is mapped to value in word register (modbus)

### 8.9.1 Type – Bit register



--- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	DPT 01 - Binary - 1 bit
Datapoints 11 - 20	Description	
Datapoints 21 - 30	Direction	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX
Datapoints 31 - 40	Type	Bit register
Datapoints 41 - 50	Value inverted	<input checked="" type="radio"/> No <input type="radio"/> Yes
Datapoints 51 - 60	Function	Write single coil - 05
Datapoints 61 - 70	Address	0

### Value inverted

If set, the inverted value of the group object corresponds to the value of the bit register.

## 8.9.2 Type – Bit in word register

--- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	<input type="text"/>
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
— Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	DPT 01 - Binary - 1 bit
Datapoints 11 - 20	Description	<input type="text"/>
Datapoints 21 - 30	Direction	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX
Datapoints 31 - 40	Type	Bit in word register
Datapoints 41 - 50	Position in word register	Bit 00
Datapoints 51 - 60	Valid bit in word register	---- -- -- --Vb
Datapoints 61 - 70	Value inverted	<input checked="" type="radio"/> No <input type="radio"/> Yes
Datapoints 71 - 80	Function	Write single holding register - 06
	Address	0

### Position in word register

Defines the bit in the word register.

### Valid bit in word register

Indicates which bit has been defined in the word register.

### Value inverted

If set, the inverted value of the group object corresponds to the value of the bit in the word register.

## 8.9.3 Type – Value in word register

### Bit count

This parameter defines the size of the value in the word register (in bits).

### Offset from right

This parameter defines the position of the value in the word register (offset from right in bits).

### 8.9.3.1 Direction – KNX to modbus

--.- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description

General settings

Modbus settings

-- Datapoints

Datapoints 1 - 10

Datapoints 11 - 20

Datapoints 21 - 30

Datapoints 31 - 40

Datapoints 41 - 50

Datapoints 51 - 60

Datapoints 61 - 70

Datapoints 71 - 80

Datapoints 81 - 90

Datapoints 91 - 100

Datapoints 101 - 110

Datapoints 111 - 120

Datapoints 121 - 130

Datapoints 131 - 140

Datapoints 141 - 150

Description for this page

Slave for this page

Slave 1

Slave 1

Channel 1

Datapoint type

DPT 01 - Binary - 1 bit

Description

Direction

☒ KNX to modbus
☐ Modbus to KNX

Type

Value in word register

Bit count

16 bit

Offset from right

00 bit

Behavior on receiving data 'ON'

☐ No reaction
☒ Set register value

Register value

1

Register value (binary)

0000 0000 0000 0001b

Register value (hexadecimal)

0001h

Behavior on receiving data 'OFF'

☐ No reaction
☒ Set register value

Register value

0

Register value (binary)

0000 0000 0000 0000b

Register value (hexadecimal)

0000h

Function

Write single holding register - 06

Address

0

### Behavior on receiving data 'ON'

Here you can parameterize whether a value is to be set in the register when an 'ON' telegram is received.

### Register value (for 'ON' telegram)

The value that is set in the register when an 'ON' telegram is received.

### Register value (binary) (for 'ON' telegram)

Binary representation of the valid bits in the register as well as the parameterized value. Dependent on **Bit count**, **Offset from right** and **Register value**.

### Register value (hexadecimal) (for 'ON' telegram)

Hexadecimal representation of the valid bits in the register as well as the parameterized value. Dependent on **Bit count**, **Offset from right** and **Register value**.

## Behavior on receiving data 'OFF'

Here you can parameterize whether a value is to be set in the register when an 'OFF' telegram is received.

### Register value (for 'OFF' telegram)

The value that is set in the register when an 'OFF' telegram is received.

### Register value (binary) (for 'OFF' telegram)

Binary representation of the valid bits in the register as well as the parameterized value. Dependent on **Bit count**, **Offset from right** and **Register value**.

### Register value (hexadecimal) (for 'OFF' telegram)

Hexadecimal representation of the valid bits in the register as well as the parameterized value. Dependent on **Bit count**, **Offset from right** and **Register value**.

## 8.9.3.2 Direction – Modbus to KNX

KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description

General settings

Modbus settings

Datapoints

Datapoints 1 - 10

Datapoints 11 - 20

Datapoints 21 - 30

Datapoints 31 - 40

Datapoints 41 - 50

Datapoints 51 - 60

Datapoints 61 - 70

Datapoints 71 - 80

Datapoints 81 - 90

Datapoints 91 - 100

Datapoints 101 - 110

Datapoints 111 - 120

Datapoints 121 - 130

Datapoints 131 - 140

Datapoints 141 - 150

Datapoints 151 - 160

Description for this page

Slave for this page

Slave 1

Channel 1

Datapoint type

DPT 01 - Binary - 1 bit

Description

Direction

☐ KNX to modbus
 ☒ Modbus to KNX

Send condition

On change

Type

Value in word register

Bit count

16 bit

Offset from right

00 bit

Threshold

1

Threshold (binary)

0000 0000 0000 0001b

Threshold (hexadecimal)

0001h

Behavior on value higher

Send 'OFF'

Behavior on value match

Send 'ON'

Behavior on value lower

Send 'OFF'

Function

☒ Read holding registers - 03
 ☐ Read input registers - 04

Address

0

Polling interval

Every cycle

## Threshold

This parameter defines the value for which the word register is checked.

### Threshold (binary)

Binary representation of the valid bits in the register as well as the parameterized value. Dependent on **Bit count**, **Offset from right** and **Threshold**.

### **Threshold (hexadecimal)**

Hexadecimal representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Threshold**.

### **Behavior on value higher**

This parameter defines the behavior of the group object in the event that the register value is greater than the parameterized value. The following options are available:

- No reaction
- Send 'ON'
- Send 'OFF'

### **Behavior on value match**

This parameter defines the behavior of the group object in case the register value corresponds to the parameterized value. The following options are available:

- No reaction
- Send 'ON'
- Send 'OFF'

### **Behavior on value lower**

This parameter defines the behavior of the group object in case the register value is smaller than the parameterized value. The following options are available:

- No reaction
- Send 'ON'
- Send 'OFF'

## **8.10 Channel function “DPT 03 – Dimming – 4 bits”**

### **Type**

The following types can be configured:

- Bit register  
4 bit dimming command (KNX) sets bit register (modbus)
- Bit in word register  
4 bit dimming command (KNX) sets 1 bit in word register (modbus)
- Value in word register  
4 bit dimming command (KNX) is mapped to value in word register (modbus)

## 8.10.1 Type – Bit register

### 8.10.1.1 Direction – KNX to modbus

--- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	DPT 03 - Dimming - 4 bits
Datapoints 11 - 20	Description	
Datapoints 21 - 30	Direction	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX
Datapoints 31 - 40	Type	Bit register
Datapoints 41 - 50	Behavior on receiving data 'DIM UP'	Bit in register - Value '1'
Datapoints 51 - 60	Behavior on receiving data 'DIM DOWN'	No reaction
Datapoints 61 - 70	Behavior on receiving data 'DIM STOP'	Bit in register - Value '0'
Datapoints 71 - 80	Function	Write single coil - 05
Datapoints 81 - 90	Address	0

#### Behavior on receiving data 'DIMM UP'

This parameter defines the behavior of the bit register in the event that a 'DIMM UP' telegram was received at the group object. The following options are available:

- No reaction
- Bit in register – Value '1'
- Bit in register – Value '0'

#### Behavior on receiving data 'DIMM DOWN'

This parameter defines the behavior of the bit register in the event that a 'DIMM DOWN' telegram was received at the group object. The following options are available:

- No reaction
- Bit in register – Value '1'
- Bit in register – Value '0'

#### Behavior on receiving data 'DIMM STOP'

This parameter defines the behavior of the bit register in the event that a 'DIMM STOP' telegram has been received at the group object. The following options are available:

- No reaction
- Bit in register – Value '1'
- Bit in register – Value '0'

### 8.10.1.2 Direction – Modbus to KNX

--- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	<input type="text"/>
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
– Datapoints	Channel 1	
– Datapoints 1 - 10	Datapoint type	DPT 03 - Dimming - 4 bits
Datapoints 11 - 20	Description	<input type="text"/>
Datapoints 21 - 30	Direction	<input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX
Datapoints 31 - 40	Send condition	On change
Datapoints 41 - 50	Type	Bit register
Datapoints 51 - 60	Behavior on bit in register - Value '1'	Send 'DIM UP'
Datapoints 61 - 70	Behavior on bit in register - Value '0'	Send 'DIM STOP'
Datapoints 71 - 80	Function	<input checked="" type="radio"/> Read coils - 01 <input type="radio"/> Read discrete inputs - 02
Datapoints 81 - 90	Address	0
Datapoints 91 - 100	Polling interval	Every cycle

#### Behavior on bit in register – Value '1'

This parameter defines the behavior of the group object in case the register value corresponds to the parameterized value. The following options are available:

- No reaction
- Send 'DIMM UP'
- Send 'DIMM DOWN'
- Send 'DIMM STOP'

#### Behavior on bit in register – Value '0'

This parameter defines the behavior of the group object in case the register value corresponds to the parameterized value. The following options are available:

- No reaction
- Send 'DIMM UP'
- Send 'DIMM DOWN'
- Send 'DIMM STOP'

## 8.10.2 Type – Bit in word register

### Position in word register

Defines the bit in the word register.

### Valid bit in word register

Indicates which bit has been defined in the word register.

### 8.10.2.1 Direction – KNX to modbus

KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10		
Description	Description for this page	
General settings	Slave for this page	
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	
Datapoints 11 - 20	DPT 03 - Dimming - 4 bits	
Datapoints 21 - 30	Description	
Datapoints 31 - 40	Direction	
Datapoints 41 - 50	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX	
Datapoints 51 - 60	Type	
Datapoints 61 - 70	Bit in word register	
Datapoints 71 - 80	Position in word register	
Datapoints 81 - 90	Bit 00	
Datapoints 91 - 100	Valid bit in word register	
	---- ---- ---- --Vb	
	Behavior on receiving data 'DIM UP'	
	Bit in register - Value '1'	
	Behavior on receiving data 'DIM DOWN'	
	No reaction	
	Behavior on receiving data 'DIM STOP'	
	Bit in register - Value '0'	
	Function	
	Write single holding register - 06	
	Address	
	0	

### Behavior on receiving data 'DIMM UP'

This parameter defines the behavior of the bit register in the event that a 'DIMM UP' telegram was received at the group object. The following options are available:

- No reaction
- Bit in register – Value '1'
- Bit in register – Value '0'

### Behavior on receiving data 'DIMM DOWN'

This parameter defines the behavior of the bit register in the event that a 'DIMM DOWN' telegram was received at the group object. The following options are available:

- No reaction
- Bit in register – Value '1'
- Bit in register – Value '0'

## Behavior on receiving data 'DIMM STOP'

This parameter defines the behavior of the bit register in the event that a 'DIMM STOP' telegram has been received at the group object. The following options are available:

- No reaction
- Bit in register – Value '1'
- Bit in register – Value '0'

### 8.10.2.2 Direction – Modbus to KNX

KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description

General settings

Modbus settings

Datapoints

Datapoints 1 - 10

Datapoints 11 - 20

Datapoints 21 - 30

Datapoints 31 - 40

Datapoints 41 - 50

Datapoints 51 - 60

Datapoints 61 - 70

Datapoints 71 - 80

Datapoints 81 - 90

Datapoints 91 - 100

Datapoints 101 - 110

Datapoints 111 - 120

Description for this page

Slave for this page

Slave 1

Channel 1

Datapoint type

DPT 03 - Dimming - 4 bits

Description

Direction

☐ KNX to modbus
 ☒ Modbus to KNX

Send condition

On change

Type

Bit in word register

Position in word register

Bit 00

Valid bit in word register

-----Vb

Behavior on bit in register - Value '1'

Send 'DIM UP'

Behavior on bit in register - Value '0'

Send 'DIM STOP'

Function

☒ Read holding registers - 03
 ☐ Read input registers - 04

Address

0

Polling interval

Every cycle

## Behavior on bit in register – Value '1'

This parameter defines the behavior of the group object in case the register value corresponds to the parameterized value. The following options are available:

- No reaction
- Send 'DIMM UP'
- Send 'DIMM DOWN'
- Send 'DIMM STOP'

## Behavior on bit in register – Value '0'

This parameter defines the behavior of the group object in case the register value corresponds to the parameterized value. The following options are available:

- No reaction
- Send 'DIMM UP'
- Send 'DIMM DOWN'
- Send 'DIMM STOP'

### 8.10.3 Type – Value in word register

#### Bit count

This parameter defines the size of the value in the word register (in bits).

#### Offset from right

This parameter defines the position of the value in the word register (offset from right in bits).

#### 8.10.3.1 Direction – KNX to modbus

KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description

General settings

Modbus settings

Datapoints

Datapoints 1 - 10

Datapoints 11 - 20

Datapoints 21 - 30

Datapoints 31 - 40

Datapoints 41 - 50

Datapoints 51 - 60

Datapoints 61 - 70

Datapoints 71 - 80

Datapoints 81 - 90

Datapoints 91 - 100

Datapoints 101 - 110

Datapoints 111 - 120

Datapoints 121 - 130

Datapoints 131 - 140

Datapoints 141 - 150

Datapoints 151 - 160

Description for this page

Slave for this page

Slave 1

Channel 1

Datapoint type

DPT 03 - Dimming - 4 bits

Description

Direction

☒ KNX to modbus
 ☐ Modbus to KNX

Type

Value in word register

Bit count

16 bit

Offset from right

00 bit

Behavior on receiving data 'DIM UP'

☐ No reaction
 ☒ Set register value

Register value

1

Register value (binary)

0000 0000 0000 0001b

Register value (hexadecimal)

0001h

Behavior on receiving data 'DIM DOWN'

☒ No reaction
 ☐ Set register value

Behavior on receiving data 'DIM STOP'

☐ No reaction
 ☒ Set register value

Register value

0

Register value (binary)

0000 0000 0000 0000b

Register value (hexadecimal)

0000h

Function

Write single holding register - 06

Address

0

#### Behavior on receiving data 'DIMM UP'

Here you can parameterize whether a value is to be set in the register when a 'DIMM UP' telegram is received.

#### Register value (for 'DIMM UP' telegram)

The value that is set in the register when a 'DIMM UP' telegram is received.

#### Register value (binary) (for 'DIMM UP' telegram)

Binary representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

#### Register value (hexadecimal) (for 'DIMM UP' telegram)

Hexadecimal representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

### **Behavior on receiving data 'DIMM DOWN'**

Here you can parameterize whether a value is to be set in the register when a 'DIMM DOWN' telegram is received.

#### **Register value** *(for 'DIMM DOWN' telegram)*

The value that is set in the register when a 'DIMM DOWN' telegram is received.

#### **Register value (binary)** *(for 'DIMM DOWN' telegram)*

Binary representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

#### **Register value (hexadecimal)** *(for 'DIMM DOWN' telegram)*

Hexadecimal representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

### **Behavior on receiving data 'DIMM STOP'**

Here you can parameterize whether a value is to be set in the register when a 'DIMM STOP' telegram is received.

#### **Register value** *(for 'DIMM STOP' telegram)*

The value that is set in the register when a 'DIMM STOP' telegram is received.

#### **Register value (binary)** *(for 'DIMM STOP' telegram)*

Binary representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

#### **Register value (hexadecimal)** *(for 'DIMM STOP' telegram)*

Hexadecimal representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

### 8.10.3.2 Direction – Modbus to KNX

--> KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	
General settings	Slave for this page	
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	
Datapoints 11 - 20	DPT 03 - Dimming - 4 bits	
Datapoints 21 - 30	Description	
Datapoints 31 - 40	Direction	
Datapoints 41 - 50	<input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX	
Datapoints 51 - 60	Send condition	
Datapoints 61 - 70	On change	
Datapoints 71 - 80	Type	
Datapoints 81 - 90	Value in word register	
Datapoints 91 - 100	Bit count	
Datapoints 101 - 110	16 bit	
Datapoints 111 - 120	Offset from right	
Datapoints 121 - 130	00 bit	
Datapoints 131 - 140	Threshold	
Datapoints 141 - 150	1	
Datapoints 151 - 160	Threshold (binary)	
	0000 0000 0000 0001b	
	Threshold (hexadecimal)	
	0001h	
	Behavior on value higher	
	Send 'DIM STOP'	
	Behavior on value match	
	Send 'DIM UP'	
	Behavior on value lower	
	Send 'DIM STOP'	
	Function	
	<input checked="" type="radio"/> Read holding registers - 03 <input type="radio"/> Read input registers - 04	
	Address	
	0	
	Polling interval	
	Every cycle	

#### Threshold

This parameter defines the value for which the word register is checked.

#### Threshold (binary)

Binary representation of the valid bits in the register as well as the parameterized value. Dependent on **Bit count**, **Offset from right** and **Threshold**.

#### Threshold (hexadecimal)

Hexadecimal representation of the valid bits in the register as well as the parameterized value. Dependent on **Bit count**, **Offset from right** and **Threshold**.

#### Behavior on value higher

This parameter defines the behavior of the group object in case the register value is greater than the parameterized value. The following options are available:

- No reaction
- Send 'DIMM UP'
- Send 'DIMM DOWN'
- Send 'DIMM STOP'

### Behavior on value match

This parameter defines the behavior of the group object in case the register value corresponds to the parameterized value. The following options are available:

- No reaction
- Send 'DIMM UP'
- Send 'DIMM DOWN'
- Send 'DIMM STOP'

### Behavior on value lower

This parameter defines the behavior of the group object in case the register value is smaller than the parameterized value. The following options are available:

- No reaction
- Send 'DIMM UP'
- Send 'DIMM DOWN'
- Send 'DIMM STOP'

### 8.11 Channel function “DPT 05 – Percent – 1 byte”

## Type

The following type is configured:

- Word register  
1 byte percent value (KNX) is mapped to value in word register (modbus)

### 8.11.1 Type – Word register

**---> KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10**

Description	Description for this page	<input type="text"/>
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
- Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	DPT 05 - Percent - 1 byte
Datapoints 11 - 20	Description	<input type="text"/>
Datapoints 21 - 30	Direction	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX
Datapoints 31 - 40	Type	Word register
Datapoints 41 - 50	Position in word register	Low byte
Datapoints 51 - 60	Value minimum (register)	0
Datapoints 61 - 70	Value maximum (register)	255
Datapoints 71 - 80	Value minimum (KNX)	0
Datapoints 81 - 90	Value maximum (KNX)	100
Datapoints 91 - 100	Function	Write single holding register - 06
	Address	0

### Position in word register

This parameter defines the range of the word register which is mapped. The following ranges are available:

- Low byte
- High byte
- High/Low byte

### Value minimum (register)

Register value, which corresponds to **Value minimum (KNX)**.

### Value maximum (register)

Register value, which corresponds to **Value maximum (KNX)**.

### Value minimum (KNX)

KNX value, which corresponds to the **Value minimum (register)**.

### Value maximum (KNX)

KNX value, which corresponds to the **Value maximum (register)**.



*The conversion is always transferred to the entire register range.  
**Value minimum/maximum (register)** does not define any limits.*

## 8.12 Channel function “DPT 05 – Configured – 1 byte”

### Type

The following types can be configured:

- Bit register  
1 byte unsigned value (KNX) sets bit register (modbus)
- Bit in word register  
1 byte unsigned value (KNX) sets 1 bit in word register (modbus)
- Value in word register  
1 byte unsigned value (KNX) is mapped to value in word register (modbus)

## 8.12.1 Type – Bit register

### 8.12.1.1 Direction – KNX to modbus

--- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	
General settings	Slave for this page	
Modbus settings	Slave 1	
– Datapoints	Channel 1	
– Datapoints 1 - 10	Datapoint type	
Datapoints 11 - 20	DPT 05 - Configured - 1 byte	
Datapoints 21 - 30	Description	
Datapoints 31 - 40	Direction	
Datapoints 41 - 50	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX	
Datapoints 51 - 60	Type	
Datapoints 61 - 70	Bit register	
Datapoints 71 - 80	Threshold	
Datapoints 81 - 90	255	
Datapoints 91 - 100	Behavior on value higher	
	Bit in register - Value '0'	
	Behavior on value match	
	Bit in register - Value '1'	
	Behavior on value lower	
	Bit in register - Value '0'	
	Function	
	Write single coil - 05	
	Address	
	0	

#### Threshold

This parameter defines the value for which the group object (KNX) is checked.

#### Behavior on value higher

This parameter defines the behavior of the bit register in the event that the value received (KNX) is greater than the parameterised value. The following options are available:

- No reaction
- Bit in register – Value '1'
- Bit in register – Value '0'

#### Behavior on value match

This parameter defines the behavior of the bit register in the event that the value received (KNX) corresponds to the parameterised value. The following options are available:

- No reaction
- Bit in register – Value '1'
- Bit in register – Value '0'

#### Behavior on value lower

This parameter defines the behavior of the bit register in the event that the value received (KNX) is less than the parameterised value. The following options are available:

- No reaction
- Bit in register – Value '1'
- Bit in register – Value '0'

### 8.12.1.2 Direction – Modbus to KNX

KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	DPT 05 - Configured - 1 byte
Datapoints 11 - 20	Description	
Datapoints 21 - 30	Direction	<input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX
Datapoints 31 - 40	Send condition	On change
Datapoints 41 - 50	Type	Bit register
Datapoints 51 - 60	Behavior on bit in register - Value '1'	<input type="radio"/> No reaction <input checked="" type="radio"/> Send value
Datapoints 61 - 70	Object value	255
Datapoints 71 - 80	Behavior on bit in register - Value '0'	<input type="radio"/> No reaction <input checked="" type="radio"/> Send value
Datapoints 81 - 90	Object value	0
Datapoints 91 - 100	Function	<input checked="" type="radio"/> Read coils - 01 <input type="radio"/> Read discrete inputs - 02
Datapoints 101 - 110	Address	0
Datapoints 111 - 120	Polling interval	Every cycle

#### Behavior on bit in register – Value '1'

Here the behavior of the group object can be defined, in case the register is set. The following options are available:

- No reaction
- Send value

#### Object value

The value that is sent when the register is set on KNX.

#### Behavior on bit in register – Value '0'

Here the behavior of the group object can be defined in case the register is not set. The following options are available:

- No reaction
- Send value

#### Object value

The value that is sent on KNX when the register is not set.

## 8.12.2 Type – Bit in word register

### Position in word register

Defines the bit in the word register.

### Valid bit in word register

Indicates which bit has been defined in the word register.

### 8.12.2.1 Direction – KNX to modbus

KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	DPT 05 - Configured - 1 byte
Datapoints 11 - 20	Description	
Datapoints 21 - 30	Direction	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX
Datapoints 31 - 40	Type	Bit in word register
Datapoints 41 - 50	Threshold	255
Datapoints 51 - 60	Position in word register	Bit 00
Datapoints 61 - 70	Valid bit in word register	-----Vb
Datapoints 71 - 80	Behavior on value higher	Bit in register - Value '0'
Datapoints 81 - 90	Behavior on value match	Bit in register - Value '1'
Datapoints 91 - 100	Behavior on value lower	Bit in register - Value '0'
Datapoints 101 - 110	Function	Write single holding register - 06
	Address	0

### Threshold

This parameter defines the value for which the group object (KNX) is checked.

### Behavior on value higher

This parameter defines the behavior of the bit in the word register in the event that the value received (KNX) is greater than the parameterised value. The following possibilities are available:

- No reaction
- Bit in register – Value '1'
- Bit in register – Value '0'

## Behavior on value match

This parameter defines the behavior of the bit in the word register in the event that the value received (KNX) corresponds to the parameterised value. The following options are available:

- No reaction
- Bit in register – Value ‘1’
- Bit in register – Value ‘0’

## Behavior on value lower

This parameter defines the behavior of the bit in the word register in the event that the value received (KNX) is less than the parameterised value. The following possibilities are available:

- No reaction
- Bit in register – Value ‘1’
- Bit in register – Value ‘0’

### 8.12.2.2 Direction – Modbus to KNX

-- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description

General settings

Modbus settings

-- Datapoints

Datapoints 1 - 10

Datapoints 11 - 20

Datapoints 21 - 30

Datapoints 31 - 40

Datapoints 41 - 50

Datapoints 51 - 60

Datapoints 61 - 70

Datapoints 71 - 80

Datapoints 81 - 90

Datapoints 91 - 100

Datapoints 101 - 110

Datapoints 111 - 120

Datapoints 121 - 130

Datapoints 131 - 140

Description for this page

Slave for this page

Slave 1

Channel 1

Datapoint type

DPT 05 - Configured - 1 byte

Description

Direction

☐ KNX to modbus
☒ Modbus to KNX

Send condition

On change

Type

Bit in word register

Position in word register

Bit 00

Valid bit in word register

---- -- -- --Vb

Behavior on bit in register - Value '1'

☐ No reaction
☒ Send value

Object value

255

Behavior on bit in register - Value '0'

☐ No reaction
☒ Send value

Object value

0

Function

☒ Read holding registers - 03
☐ Read input registers - 04

Address

0

Polling interval

Every cycle

## Behavior on bit in register – Value ‘1’

Here the behavior of the group object can be defined, in case the bit in the word register is set. The following options are available:

- No reaction
- Send value

## Object value

The value that is sent on KNX when the bit in the word register is set.

## Behavior on bit in register – Value ‘0’

Here the behavior of the group object can be defined in the event that the bit in the word register is not set. The following options are available:

- No reaction
- Send value

## Object value

The value that is sent on KNX when the bit in the word register is not set.

## 8.12.3 Type – Value in word register

### Bit count

This parameter defines the size of the value in the word register (in bits).

### Offset from right

This parameter defines the position of the value in the word register (offset from right in bits).

### 8.12.3.1 Direction – KNX to modbus

-- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	DPT 05 - Configured - 1 byte
Datapoints 11 - 20	Description	
Datapoints 21 - 30	Direction	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX
Datapoints 31 - 40	Type	Value in word register
Datapoints 41 - 50	Threshold	255
Datapoints 51 - 60	Bit count	16 bit
Datapoints 61 - 70	Offset from right	00 bit
Datapoints 71 - 80	Behavior on value higher	<input type="radio"/> No reaction <input checked="" type="radio"/> Set register value
Datapoints 81 - 90	Register value	0
Datapoints 91 - 100	Register value (binary)	0000 0000 0000 0000b
Datapoints 101 - 110	Register value (hexadecimal)	0000h
Datapoints 111 - 120	Behavior on value match	<input type="radio"/> No reaction <input checked="" type="radio"/> Set register value
Datapoints 121 - 130	Register value	1
Datapoints 131 - 140	Register value (binary)	0000 0000 0000 0001b
Datapoints 141 - 150	Register value (hexadecimal)	0001h
Datapoints 151 - 160	Behavior on value lower	<input type="radio"/> No reaction <input checked="" type="radio"/> Set register value
Datapoints 161 - 170	Register value	0
Datapoints 171 - 180	Register value (binary)	0000 0000 0000 0000b
Datapoints 181 - 190	Register value (hexadecimal)	0000h
Datapoints 191 - 200	Function	Write single holding register - 06
	Address	0

## Threshold

This parameter defines the value for which the group object (KNX) is checked.

### Behavior on value higher

This parameter defines the behavior of the word register in the event that the value received (KNX) is greater than the parameterised value. The following options are available:

- No reaction
- Set register value

### Register value *(for value higher)*

The value which is set in the word register.

### Register value (binary) *(for value higher)*

Binary representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

### Register value (hexadecimal) *(for value higher)*

Hexadecimal representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

### Behavior on value match

This parameter defines the behavior of the word register in the event that the value received (KNX) corresponds to the parameterised value. The following options are available:

- No reaction
- Set register value

### Register value *(for value match)*

The value which is set in the word register.

### Register value (binary) *(for value match)*

Binary representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

### Register value (hexadecimal) *(for value match)*

Hexadecimal representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

### Behavior on value lower

This parameter defines the behavior of the word register in the event that the value received (KNX) is less than the parameterised value. The following options are available:

- No reaction
- Set register value

### Register value *(for value lower)*

The value which is set in the word register.

### Register value (binary) *(for value lower)*

Binary representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

### Register value (hexadecimal) *(for value lower)*

Hexadecimal representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

## 8.12.3.2 Direction – Modbus to KNX

KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description

General settings

Modbus settings

Datapoints

Datapoints 1 - 10

Datapoints 11 - 20

Datapoints 21 - 30

Datapoints 31 - 40

Datapoints 41 - 50

Datapoints 51 - 60

Datapoints 61 - 70

Datapoints 71 - 80

Datapoints 81 - 90

Datapoints 91 - 100

Datapoints 101 - 110

Datapoints 111 - 120

Datapoints 121 - 130

Datapoints 131 - 140

Datapoints 141 - 150

Datapoints 151 - 160

Datapoints 161 - 170

Datapoints 171 - 180

Datapoints 181 - 190

Description for this page

Slave for this page

Slave 1

Channel 1

Datapoint type

DPT 05 - Configured - 1 byte

Description

Direction

☐ KNX to modbus
 ☒ Modbus to KNX

Send condition

On change

Type

Value in word register

Bit count

16 bit

Offset from right

00 bit

Threshold

1

Threshold (binary)

0000 0000 0000 0001b

Threshold (hexadecimal)

0001h

Behavior on value higher

☐ No reaction
 ☒ Send value

Object value

0

Behavior on value match

☐ No reaction
 ☒ Send value

Object value

255

Behavior on value lower

☐ No reaction
 ☒ Send value

Object value

0

Function

☒ Read holding registers - 03
 ☐ Read input registers - 04

Address

0

Polling interval

Every cycle

### Threshold

This parameter defines the value for which the word register is checked.

### Threshold (binary)

Binary representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Threshold**.

### Threshold (hexadecimal)

Hexadecimal representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Threshold**.

**Behavior on value higher**

This parameter defines the behavior of the group object (KNX) in the event that the register value is greater than the parameterised value. The following options are available:

- No reaction
- Send value

**Object value** *(for value higher)*

The value that is sent via the group object (KNX).

**Behavior on value match**

This parameter defines the behavior of the group object (KNX) in the event that the register value corresponds to the parameterised value. The following options are available:

- No reaction
- Send value

**Object value** *(for value match)*

The value that is sent via the group object (KNX).

**Behavior on value lower**

This parameter defines the behavior of the group object (KNX) in the event that the register value is less than the parameterised value. The following options are available:

- No reaction
- Send value

**Object value** *(for value lower)*

The value that is sent via the group object (KNX).

## 8.13 Channel function “DPT 05 – Unsigned – 1 byte”

**Type**

The following type is configured:

- Word register  
1 byte unsigned value (KNX) is written/read to/from area in word register (modbus)

### 8.13.1 Type – Word register

--- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	<input type="text"/>
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	DPT 05 - Unsigned - 1 byte
Datapoints 11 - 20	Description	<input type="text"/>
Datapoints 21 - 30	Direction	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX
Datapoints 31 - 40	Type	Word register
Datapoints 41 - 50	Position in word register	Configured
Datapoints 51 - 60	Bit count	08 bit
Datapoints 61 - 70	Offset from right	00 bit
Datapoints 71 - 80	Valid bits in word register	---- -- VVVV VVVb
Datapoints 81 - 90	Valid values	0 ... 255
Datapoints 91 - 100	Function	Write single holding register - 06
	Address	0

#### Position in word register

This parameter defines the range of the word register which is written/read. The following areas are available:

- Low byte
- High byte
- Configured

#### Bit count (only for configured)

This parameter defines the size of the area in the word register (in bits).

#### Offset from right (only for configured)

This parameter defines the position of the range in the word register (offset from right in bits).

#### Valid bits in word register (only for configured)

Indicates which bits have been defined in the word register.

Depending on **Bit count** and **Offset from right**.

#### Valid values (only for configured)

Indicates which values fit into the defined bits.

Depending on **Bit count** and **Offset from right**.



**Bit count** and **Offset from right** must not be greater than 16 together.  
The value must fit into **Bit count**, e.g. **Bit count** = 1 → "Value" = 0 or 1.

## 8.14 Channel function “DPT 06 – Signed – 1 byte”

### Type

The following type is configured:

- Word register  
1 byte signed value (KNX) is written/read to/from area in word register (modbus)

### 8.14.1 Type – Word register

--- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10		
Description	Description for this page	
General settings	Slave for this page	
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	
Datapoints 11 - 20	DPT 06 - Signed - 1 byte	
Datapoints 21 - 30	Description	
Datapoints 31 - 40	Direction	
Datapoints 41 - 50	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX	
Datapoints 51 - 60	Type	
Datapoints 61 - 70	Word register	
Datapoints 71 - 80	Position in word register	
Datapoints 81 - 90	Configured	
	Offset from right	
	00 bit	
	Valid bits in word register	
	---- ---- VVVV VVVVb	
	Valid values	
	-128 ... 127	
	Function	
	Write single holding register - 06	
	Address	
	0	

### Position in word register

This parameter defines the range of the word register which is written/read. The following areas are available:

- Low byte
- High byte
- Configured

### Offset from right (only for configured)

This parameter defines the position of the range in the word register (offset from right in bits).

### Valid bits in word register (only for configured)

Indicates which bits have been defined in the word register.

Dependent on **Offset from right**.

### Valid values (only for configured)

Indicates which values fit into the defined bits.

## 8.15 Channel function “DPT 07 – Configured – 2 bytes”

### Type

The following types can be configured:

- Bit register  
2 byte unsigned value (KNX) sets bit register (modbus)
- Bit in word register  
2 byte unsigned value (KNX) sets 1 bit in word register (modbus)
- Value in word register  
2 byte unsigned value (KNX) is mapped to value in word register (modbus)

### 8.15.1 Type – Bit register

#### 8.15.1.1 Direction – KNX to modbus

KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description

General settings

Modbus settings

Datapoints

Datapoints 1 - 10

Datapoints 11 - 20

Datapoints 21 - 30

Datapoints 31 - 40

Datapoints 41 - 50

Datapoints 51 - 60

Datapoints 61 - 70

Datapoints 71 - 80

Datapoints 81 - 90

Datapoints 91 - 100

Description for this page

Slave for this page

Slave 1

Channel 1

Datapoint type

DPT 07 - Configured - 2 bytes

Description

Direction

☒ KNX to modbus
 ☐ Modbus to KNX

Type

Bit register

Threshold

65535

Behavior on value higher

Bit in register - Value '0'

Behavior on value match

Bit in register - Value '1'

Behavior on value lower

Bit in register - Value '0'

Function

Write single coil - 05

Address

0

### Threshold

This parameter defines the value for which the group object (KNX) is checked.

### Behavior on value higher

This parameter defines the behavior of the bit register in the event that the value received (KNX) is greater than the parameterised value. The following options are available:

- No reaction
- Bit in register – Value ‘1’
- Bit in register – Value ‘0’

### Behavior on value match

This parameter defines the behavior of the bit register in the event that the value received (KNX) corresponds to the parameterised value. The following options are available:

- No reaction
- Bit in register – Value ‘1’
- Bit in register – Value ‘0’

### Behavior on value lower

This parameter defines the behavior of the bit register in the event that the value received (KNX) is less than the parameterised value. The following options are available:

- No reaction
- Bit in register – Value ‘1’
- Bit in register – Value ‘0’

## 8.15.1.2 Direction – Modbus to KNX

KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description

General settings

Modbus settings

Datapoints

Datapoints 1 - 10

Datapoints 11 - 20

Datapoints 21 - 30

Datapoints 31 - 40

Datapoints 41 - 50

Datapoints 51 - 60

Datapoints 61 - 70

Datapoints 71 - 80

Datapoints 81 - 90

Datapoints 91 - 100

Datapoints 101 - 110

Datapoints 111 - 120

Description for this page

Slave for this page

Slave 1

Channel 1

Datapoint type

Description

Direction

Send condition

Type

Behavior on bit in register - Value '1'

Object value

Behavior on bit in register - Value '0'

Object value

Function

Address

Polling interval

Slave 1

DPT 07 - Configured - 2 bytes

KNX to modbus

Modbus to KNX

On change

Bit register

No reaction

Send value

65535

No reaction

Send value

0

Read coils - 01

Read discrete inputs - 02

0

Every cycle

### Behavior on bit in register – Value ‘1’

Here the behavior of the group object can be defined, in case the register is set. The following options are available:

- No reaction
- Send value

### Object value

The value that is sent when the register is set on KNX.

## Behavior on bit in register – Value '0'

Here the behavior of the group object can be defined in case the register is not set. The following options are available:

- No reaction
- Send value

## Object value

The value that is sent on KNX when the register is not set.

## 8.15.2 Type – Bit in word register

### Position in word register

Defines the bit in the word register.

### Valid bit in word register

Indicates which bit has been defined in the word register.

### 8.15.2.1 Direction – KNX to modbus

KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description

General settings

Modbus settings

Datapoints

Datapoints 1 - 10

Datapoints 11 - 20

Datapoints 21 - 30

Datapoints 31 - 40

Datapoints 41 - 50

Datapoints 51 - 60

Datapoints 61 - 70

Datapoints 71 - 80

Datapoints 81 - 90

Datapoints 91 - 100

Datapoints 101 - 110

Description for this page

Slave for this page

Slave 1

Channel 1

Datapoint type

DPT 07 - Configured - 2 bytes

Description

Direction

☒ KNX to modbus
 ☐ Modbus to KNX

Type

Bit in word register

Threshold

65535

Position in word register

Bit 00

Valid bit in word register

-----Vb

Behavior on value higher

Bit in register - Value '0'

Behavior on value match

Bit in register - Value '1'

Behavior on value lower

Bit in register - Value '0'

Function

Write single holding register - 06

Address

0

## Threshold

This parameter defines the value for which the group object (KNX) is checked.

### Behavior on value higher

This parameter defines the behavior of the bit in the word register in the event that the value received (KNX) is greater than the parameterised value. The following possibilities are available:

- No reaction
- Bit in register – Value '1'
- Bit in register – Value '0'

### Behavior on value match

This parameter defines the behavior of the bit in the word register in the event that the value received (KNX) corresponds to the parameterised value. The following options are available:

- No reaction
- Bit in register – Value '1'
- Bit in register – Value '0'

### Behavior on value lower

This parameter defines the behavior of the bit in the word register in the event that the value received (KNX) is less than the parameterised value. The following possibilities are available:

- No reaction
- Bit in register – Value '1'
- Bit in register – Value '0'

## 8.15.2.2 Direction – Modbus to KNX

--- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	
General settings	Slave for this page	
Modbus settings	Slave 1	
- Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	
Datapoints 11 - 20	DPT 07 - Configured - 2 bytes	
Datapoints 21 - 30	Description	
Datapoints 31 - 40	Direction	
Datapoints 41 - 50	<input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX	
Datapoints 51 - 60	Send condition	
Datapoints 61 - 70	On change	
Datapoints 71 - 80	Type	
Datapoints 81 - 90	Bit in word register	
Datapoints 91 - 100	Position in word register	
Datapoints 101 - 110	Bit 00	
Datapoints 111 - 120	Valid bit in word register	
Datapoints 121 - 130	-----Vb	
Datapoints 131 - 140	Behavior on bit in register - Value '1'	
		<input type="radio"/> No reaction <input checked="" type="radio"/> Send value
		Object value
		65535
		Behavior on bit in register - Value '0'
		<input type="radio"/> No reaction <input checked="" type="radio"/> Send value
		Object value
		0
		Function
		<input checked="" type="radio"/> Read holding registers - 03 <input type="radio"/> Read input registers - 04
		Address
		0
		Polling interval
		Every cycle

### **Behavior on bit in register – Value ‘1’**

Here the behavior of the group object can be defined, in case the bit in the word register is set. The following options are available:

- No reaction
- Send value

### **Object value**

The value that is sent on KNX when the bit in the word register is set.

### **Behavior on bit in register – Value ‘0’**

Here the behavior of the group object can be defined in the event that the bit in the word register is not set. The following options are available:

- No reaction
- Send value

### **Object value**

The value that is sent on KNX when the bit in the word register is not set.

### 8.15.3 Type – Value in word register

#### Bit count

This parameter defines the size of the value in the word register (in bits).

#### Offset from right

This parameter defines the position of the value in the word register (offset from right in bits).

#### 8.15.3.1 Direction – KNX to modbus

--> KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	DPT 07 - Configured - 2 bytes
Datapoints 11 - 20	Description	
Datapoints 21 - 30	Direction	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX
Datapoints 31 - 40	Type	Value in word register
Datapoints 41 - 50	Threshold	65535
Datapoints 51 - 60	Bit count	16 bit
Datapoints 61 - 70	Offset from right	00 bit
Datapoints 71 - 80	Behavior on value higher	<input type="radio"/> No reaction <input checked="" type="radio"/> Set register value
Datapoints 81 - 90	Register value	0
Datapoints 91 - 100	Register value (binary)	0000 0000 0000 0000b
Datapoints 101 - 110	Register value (hexadecimal)	0000h
Datapoints 111 - 120	Behavior on value match	<input type="radio"/> No reaction <input checked="" type="radio"/> Set register value
Datapoints 121 - 130	Register value	1
Datapoints 131 - 140	Register value (binary)	0000 0000 0000 0001b
Datapoints 141 - 150	Register value (hexadecimal)	0001h
Datapoints 151 - 160	Behavior on value lower	<input type="radio"/> No reaction <input checked="" type="radio"/> Set register value
Datapoints 161 - 170	Register value	0
Datapoints 171 - 180	Register value (binary)	0000 0000 0000 0000b
Datapoints 181 - 190	Register value (hexadecimal)	0000h
Datapoints 191 - 200	Function	Write single holding register - 06
	Address	0

#### Threshold

This parameter defines the value for which the group object (KNX) is checked.

### **Behavior on value higher**

This parameter defines the behavior of the word register in the event that the value received (KNX) is greater than the parameterised value. The following options are available:

- No reaction
- Set register value

### **Register value** *(for value higher)*

The value which is set in the word register.

### **Register value (binary)** *(for value higher)*

Binary representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

### **Register value (hexadecimal)** *(for value higher)*

Hexadecimal representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

### **Behavior on value match**

This parameter defines the behavior of the word register in the event that the value received (KNX) corresponds to the parameterised value. The following options are available:

- No reaction
- Set register value

### **Register value** *(for value match)*

The value which is set in the word register.

### **Register value (binary)** *(for value match)*

Binary representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

### **Register value (hexadecimal)** *(for value match)*

Hexadecimal representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

### **Behavior on value lower**

This parameter defines the behavior of the word register in the event that the value received (KNX) is less than the parameterised value. The following options are available:

- No reaction
- Set register value

### **Register value** *(for value lower)*

The value which is set in the word register.

### Register value (binary) *(for value lower)*

Binary representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

### Register value (hexadecimal) *(for value lower)*

Hexadecimal representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

## 8.15.3.2 Direction – Modbus to KNX

KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description

General settings

Modbus settings

Datapoints

Datapoints 1 - 10

Datapoints 11 - 20

Datapoints 21 - 30

Datapoints 31 - 40

Datapoints 41 - 50

Datapoints 51 - 60

Datapoints 61 - 70

Datapoints 71 - 80

Datapoints 81 - 90

Datapoints 91 - 100

Datapoints 101 - 110

Datapoints 111 - 120

Datapoints 121 - 130

Datapoints 131 - 140

Datapoints 141 - 150

Datapoints 151 - 160

Datapoints 161 - 170

Datapoints 171 - 180

Datapoints 181 - 190

Description for this page

Slave for this page

Slave 1

Channel 1

Datapoint type

DPT 07 - Configured - 2 bytes

Description

Direction

☐ KNX to modbus
 ☒ Modbus to KNX

Send condition

On change

Type

Value in word register

Bit count

16 bit

Offset from right

00 bit

Threshold

1

Threshold (binary)

0000 0000 0000 0001b

Threshold (hexadecimal)

0001h

Behavior on value higher

☐ No reaction
 ☒ Send value

Object value

0

Behavior on value match

☐ No reaction
 ☒ Send value

Object value

65535

Behavior on value lower

☐ No reaction
 ☒ Send value

Object value

0

Function

☒ Read holding registers - 03
 ☐ Read input registers - 04

Address

0

Polling interval

Every cycle

### Threshold

This parameter defines the value for which the word register is checked.

### Threshold (binary)

Binary representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Threshold**.

### Threshold (hexadecimal)

Hexadecimal representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Threshold**.

**Behavior on value higher**

This parameter defines the behavior of the group object (KNX) in the event that the register value is greater than the parameterised value. The following options are available:

- No reaction
- Send value

**Object value** *(for value higher)*

The value that is sent via the group object (KNX).

**Behavior on value match**

This parameter defines the behavior of the group object (KNX) in the event that the register value corresponds to the parameterised value. The following options are available:

- No reaction
- Send value

**Object value** *(for value match)*

The value that is sent via the group object (KNX).

**Behavior on value lower**

This parameter defines the behavior of the group object (KNX) in the event that the register value is less than the parameterised value. The following options are available:

- No reaction
- Send value

**Object value** *(for value lower)*

The value that is sent via the group object (KNX).

## **8.16 Channel function “DPT 07 – Unsigned – 2 bytes”**

**Type**

The following type is configured:

- Word register  
2 byte unsigned value (KNX) is written/read to/from area in word register (modbus)

## 8.16.1 Type – Word register

--- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	<input type="text"/>
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	DPT 07 - Unsigned - 2 bytes
Datapoints 11 - 20	Description	<input type="text"/>
Datapoints 21 - 30	Direction	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX
Datapoints 31 - 40	Type	Word register
Datapoints 41 - 50	Position in word register	<input type="radio"/> High/Low byte <input checked="" type="radio"/> Configured
Datapoints 51 - 60	Bit count	16 bit
Datapoints 61 - 70	Offset from right	00 bit
Datapoints 71 - 80	Valid bits in word register	VVVV VVVV VVVV VVVVb
Datapoints 81 - 90	Valid values	0 ... 65535
Datapoints 91 - 100	Function	Write single holding register - 06
	Address	0

### Position in word register

This parameter defines the range of the word register which is written/read. The following areas are available:

- High/Low byte
- Configured

### Bit count (only for configured)

This parameter defines the size of the area in the word register (in bits).

### Offset from right (only for configured)

This parameter defines the position of the range in the word register (offset from right in bits).

### Valid bits in word register (only for configured)

Indicates which bits have been defined in the word register.

Depending on **Bit count** and **Offset from right**.

### Valid values (only for configured)

Indicates which values fit into the defined bits.

Depending on **Bit count** and **Offset from right**.



**Bit count** and **Offset from right** must not be greater than 16 together.  
The value must fit into **Bit count**, e.g. **Bit count** = 1 → "Value" = 0 or 1.

## 8.17 Channel function “DPT 08 – Signed – 2 bytes”

### Type

The following type is configured:

- Word register  
2 byte signed value (KNX) is written/read to/from area in word register (modbus)

### 8.17.1 Type – Word register

--- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	<input type="text"/>
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
– Datapoints	Channel 1	
– Datapoints 1 - 10	Datapoint type	DPT 08 - Signed - 2 bytes
Datapoints 11 - 20	Description	<input type="text"/>
Datapoints 21 - 30	Direction	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX
Datapoints 31 - 40	Type	Word register
Datapoints 41 - 50	Position in word register	High/Low byte
Datapoints 51 - 60	Function	Write single holding register - 06
	Address	0

### Position in word register

This parameter defines the range of the word register which is written/read. The following range is configured:

- High/Low byte

## 8.18 Channel function “DPT 09 – Float – 2 bytes”

### Type

The following type is configured:

- Word register  
2 byte float value (KNX) is mapped to area in word register (modbus)
- Double word register  
2 byte float value (KNX) is mapped to two word registers (modbus)

## 8.18.1 Type – Word register

--- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	<input type="text"/>
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	DPT 09 - Float - 2 bytes
Datapoints 11 - 20	Description	<input type="text"/>
Datapoints 21 - 30	Direction	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX
Datapoints 31 - 40	Type	<input checked="" type="radio"/> Word register <input type="radio"/> Double word register
Datapoints 41 - 50	Position in word register	High/Low byte - unsigned
Datapoints 51 - 60	Value minimum (register)	0
Datapoints 61 - 70	Value maximum (register)	100
Datapoints 71 - 80	Value minimum (KNX)	0
Datapoints 81 - 90	Value maximum (KNX)	100
Datapoints 91 - 100	Function	Write single holding register - 06
Datapoints 101 - 110	Address	0

### Position in word register

This parameter defines the range of the word register which is mapped. The following ranges are available:

- Low byte – unsigned
- High byte – unsigned
- High/Low byte – unsigned
- Low byte – 2th complement
- High byte – 2th complement
- High/Low byte – 2th complement

### Value minimum (register)

Register value, which corresponds to **Value minimum (KNX)**.

### Value maximum (register)

Register value, which corresponds to **Value maximum (KNX)**.

### Value minimum (KNX)

KNX value, which corresponds to **Value minimum (register)**.

### Value maximum (KNX)

KNX value, which corresponds to **Value maximum (register)**.



*The conversion is always transferred to the entire register range.  
**Value minimum/maximum (register)** does not define any limits.*

## 8.18.2 Type – Double word register

KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	DPT 09 - Float - 2 bytes
Datapoints 11 - 20	Description	
Datapoints 21 - 30	Direction	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX
Datapoints 31 - 40	Type	<input type="radio"/> Word register <input checked="" type="radio"/> Double word register
Datapoints 41 - 50	Word order	<input checked="" type="radio"/> Hi word @ address / Lo word @ address + 1 <input type="radio"/> Lo word @ address / Hi word @ address + 1
Datapoints 51 - 60	Type register value	Modbus holds integer value - unsigned
Datapoints 61 - 70	Value minimum (register)	0
Datapoints 71 - 80	Value maximum (register)	100
Datapoints 81 - 90	Value minimum (KNX)	0
Datapoints 91 - 100	Value maximum (KNX)	100
Datapoints 101 - 110	Function	Write single holding register - 06
Datapoints 111 - 120	Address	0

### Word order

This parameter defines the byte order in which the value of the group object (KNX) is distributed to the two word registers (modbus). The following options are available:

- Hi word @ address / Lo word @ address + 1
- Lo word @ address / Hi word @ address + 1

### Type register value

Here you define how the float value is to be mapped to modbus. The following options are available:

- Modbus holds integer value – unsigned
- Modbus holds integer value – 2th complement
- Modbus holds float value – IEEE

**Value minimum (register)** *(only for integer value)*

Register value, which corresponds to **Value minimum (KNX)**.

**Value maximum (register)** *(only for integer value)*

Register value, which corresponds to **Value maximum (KNX)**.

**Value minimum (KNX)** *(only for integer value)*

KNX value, which corresponds to **Value minimum (register)**.

**Value maximum (KNX)** *(only for integer value)*

KNX value, which corresponds to **Value maximum (register)**.



*The conversion is always transferred to the entire register range.*

**Value minimum/maximum (register)** does not define any limits.

**Scaling factor** *(only for float value)*

A scaling factor can be specified here, which is applied when converting from KNX to modbus and from modbus to KNX.

**Function** *(as “Modbus master”, for “KNX to modbus” and “Double word register”)*

By means of **Word register write requests** the transmission type of the double word register can be configured. The following options are available:

- Write single holding register – 06  
One word register in one request
- Write multi holding registers – 16  
Both word registers in one request



*As “Modbus master”, for “Modbus to KNX” and “Double word register”, multi read requests should be enabled to read both word registers in one request.*

**Address** *(for “Double word register”)*

Double word registers use the register address specified here plus this register address + 1.

## 8.19 Channel function “DPT 12 – Configured – 4 bytes”

### Type

The following types can be configured:

- Bit register  
4 byte unsigned value (KNX) sets bit register (modbus)
- Bit in word register  
4 byte unsigned value (KNX) sets 1 bit in word register (modbus)
- Value in word register  
4 byte unsigned value (KNX) is mapped to value in word register (modbus)

### 8.19.1 Type – Bit register

#### 8.19.1.1 Direction – KNX to modbus

--.- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	DPT 12 - Configured - 4 bytes
Datapoints 11 - 20	Description	
Datapoints 21 - 30	Direction	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX
Datapoints 31 - 40	Type	Bit register
Datapoints 41 - 50	Threshold	4294967295
Datapoints 51 - 60	Behavior on value higher	Bit in register - Value '0'
Datapoints 61 - 70	Behavior on value match	Bit in register - Value '1'
Datapoints 71 - 80	Behavior on value lower	Bit in register - Value '0'
Datapoints 81 - 90	Function	Write single coil - 05
Datapoints 91 - 100	Address	0

### Threshold

This parameter defines the value for which the group object (KNX) is checked.

### Behavior on value higher

This parameter defines the behavior of the bit register in the event that the value received (KNX) is greater than the parameterised value. The following options are available:

- No reaction
- Bit in register – Value ‘1’
- Bit in register – Value ‘0’

### Behavior on value match

This parameter defines the behavior of the bit register in the event that the value received (KNX) corresponds to the parameterised value. The following options are available:

- No reaction
- Bit in register – Value ‘1’
- Bit in register – Value ‘0’

### Behavior on value lower

This parameter defines the behavior of the bit register in the event that the value received (KNX) is less than the parameterised value. The following options are available:

- No reaction
- Bit in register – Value ‘1’
- Bit in register – Value ‘0’

## 8.19.1.2 Direction – Modbus to KNX

-.- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page <input style="width: 90%;" type="text"/>	
General settings	Slave for this page <div style="border: 1px solid #ccc; padding: 2px;">Slave 1</div>	
Modbus settings	Slave 1	
- Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type <div style="border: 1px solid #ccc; padding: 2px;">DPT 12 - Configured - 4 bytes</div>	
Datapoints 11 - 20	Description <input style="width: 90%;" type="text"/>	
Datapoints 21 - 30	Direction <div style="display: flex; align-items: center;"> <input type="radio"/> KNX to modbus           <input checked="" type="radio"/> Modbus to KNX         </div>	
Datapoints 31 - 40	Send condition <div style="border: 1px solid #ccc; padding: 2px;">On change</div>	
Datapoints 41 - 50	Type <div style="border: 1px solid #ccc; padding: 2px;">Bit register</div>	
Datapoints 51 - 60	Behavior on bit in register - Value '1' <div style="display: flex; align-items: center;"> <input type="radio"/> No reaction           <input checked="" type="radio"/> Send value         </div>	
Datapoints 61 - 70	Object value <div style="border: 1px solid #ccc; padding: 2px;">4294967295</div>	
Datapoints 71 - 80	Behavior on bit in register - Value '0' <div style="display: flex; align-items: center;"> <input type="radio"/> No reaction           <input checked="" type="radio"/> Send value         </div>	
Datapoints 81 - 90	Object value <div style="border: 1px solid #ccc; padding: 2px;">0</div>	
Datapoints 91 - 100	Function <div style="display: flex; align-items: center;"> <input checked="" type="radio"/> Read coils - 01           <input type="radio"/> Read discrete inputs - 02         </div>	
Datapoints 101 - 110	Address <div style="border: 1px solid #ccc; padding: 2px;">0</div>	
Datapoints 111 - 120	Polling interval <div style="border: 1px solid #ccc; padding: 2px;">Every cycle</div>	

### Behavior on bit in register – Value ‘1’

Here the behavior of the group object can be defined, in case the register is set. The following options are available:

- No reaction
- Send value

### Object value

The value that is sent when the register is set on KNX.

### Behavior on bit in register – Value '0'

Here the behavior of the group object can be defined in case the register is not set. The following options are available:

- No reaction
- Send value

### Object value

The value that is sent on KNX when the register is not set.

## 8.19.2 Type – Bit in word register

### Position in word register

Defines the bit in the word register.

### Valid bit in word register

Indicates which bit has been defined in the word register.

### 8.19.2.1 Direction – KNX to modbus

KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description

General settings

Modbus settings

Datapoints

Datapoints 1 - 10

Datapoints 11 - 20

Datapoints 21 - 30

Datapoints 31 - 40

Datapoints 41 - 50

Datapoints 51 - 60

Datapoints 61 - 70

Datapoints 71 - 80

Datapoints 81 - 90

Datapoints 91 - 100

Datapoints 101 - 110

Description for this page

Slave for this page

Slave 1

Channel 1

Datapoint type

DPT 12 - Configured - 4 bytes

Description

Direction

☒ KNX to modbus
 ☐ Modbus to KNX

Type

Bit in word register

Threshold

4294967295

Position in word register

Bit 00

Valid bit in word register

-----Vb

Behavior on value higher

Bit in register - Value '0'

Behavior on value match

Bit in register - Value '1'

Behavior on value lower

Bit in register - Value '0'

Function

Write single holding register - 06

Address

0

### Threshold

This parameter defines the value for which the group object (KNX) is checked.

### **Behavior on value higher**

This parameter defines the behavior of the bit in the word register in the event that the value received (KNX) is greater than the parameterised value. The following possibilities are available:

- No reaction
- Bit in register – Value '1'
- Bit in register – Value '0'

### **Behavior on value match**

This parameter defines the behavior of the bit in the word register in the event that the value received (KNX) corresponds to the parameterised value. The following options are available:

- No reaction
- Bit in register – Value '1'
- Bit in register – Value '0'

### **Behavior on value lower**

This parameter defines the behavior of the bit in the word register in the event that the value received (KNX) is less than the parameterised value. The following possibilities are available:

- No reaction
- Bit in register – Value '1'
- Bit in register – Value '0'

### 8.19.2.2 Direction – Modbus to KNX

KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	DPT 12 - Configured - 4 bytes
Datapoints 11 - 20	Description	
Datapoints 21 - 30	Direction	<input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX
Datapoints 31 - 40	Send condition	On change
Datapoints 41 - 50	Type	Bit in word register
Datapoints 51 - 60	Position in word register	Bit 00
Datapoints 61 - 70	Valid bit in word register	-----Vb
Datapoints 71 - 80	Behavior on bit in register - Value '1'	<input type="radio"/> No reaction <input checked="" type="radio"/> Send value
Datapoints 81 - 90	Object value	4294967295
Datapoints 91 - 100	Behavior on bit in register - Value '0'	<input type="radio"/> No reaction <input checked="" type="radio"/> Send value
Datapoints 101 - 110	Object value	0
Datapoints 111 - 120	Function	<input checked="" type="radio"/> Read holding registers - 03 <input type="radio"/> Read input registers - 04
Datapoints 121 - 130	Address	0
Datapoints 131 - 140	Polling interval	Every cycle

#### Behavior on bit in register – Value '1'

Here the behavior of the group object can be defined, in case the bit in the word register is set. The following options are available:

- No reaction
- Send value

#### Object value

The value that is sent on KNX when the bit in the word register is set.

#### Behavior on bit in register – Value '0'

Here the behavior of the group object can be defined in the event that the bit in the word register is not set. The following options are available:

- No reaction
- Send value

#### Object value

The value that is sent on KNX when the bit in the word register is not set.

## 8.19.3 Type – Value in word register

### Bit count

This parameter defines the size of the value in the word register (in bits).

### Offset from right

This parameter defines the position of the value in the word register (offset from right in bits).

### 8.19.3.1 Direction – KNX to modbus

-- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description

General settings

Modbus settings

-- Datapoints

Datapoints 1 - 10

Datapoints 11 - 20

Datapoints 21 - 30

Datapoints 31 - 40

Datapoints 41 - 50

Datapoints 51 - 60

Datapoints 61 - 70

Datapoints 71 - 80

Datapoints 81 - 90

Datapoints 91 - 100

Datapoints 101 - 110

Datapoints 111 - 120

Datapoints 121 - 130

Datapoints 131 - 140

Datapoints 141 - 150

Datapoints 151 - 160

Datapoints 161 - 170

Datapoints 171 - 180

Datapoints 181 - 190

Datapoints 191 - 200

Description for this page

Slave for this page

Slave 1

Channel 1

Datapoint type

DPT 12 - Configured - 4 bytes

Description

Direction

☒ KNX to modbus
☐ Modbus to KNX

Type

Value in word register

Threshold

4294967295

Bit count

16 bit

Offset from right

00 bit

Behavior on value higher

☐ No reaction
☒ Set register value

Register value

0

Register value (binary)

0000 0000 0000 0000b

Register value (hexadecimal)

0000h

Behavior on value match

☐ No reaction
☒ Set register value

Register value

1

Register value (binary)

0000 0000 0000 0001b

Register value (hexadecimal)

0001h

Behavior on value lower

☐ No reaction
☒ Set register value

Register value

0

Register value (binary)

0000 0000 0000 0000b

Register value (hexadecimal)

0000h

Function

Write single holding register - 06

Address

0

### Threshold

This parameter defines the value for which the group object (KNX) is checked.

### Behavior on value higher

This parameter defines the behavior of the word register in the event that the value received (KNX) is greater than the parameterised value. The following options are available:

- No reaction
- Set register value

**Register value** *(for value higher)*

The value which is set in the word register.

**Register value (binary)** *(for value higher)*

Binary representation of the valid bits in the register as well as the parameterized value.

Dependent on **Bit count**, **Offset from right** and **Register value**.

**Register value (hexadecimal)** *(for value higher)*

Hexadecimal representation of the valid bits in the register as well as the parameterized value.

Dependent on **Bit count**, **Offset from right** and **Register value**.

**Behavior on value match**

This parameter defines the behavior of the word register in the event that the value received (KNX) corresponds to the parameterised value. The following options are available:

- No reaction
- Set register value

**Register value** *(for value match)*

The value which is set in the word register.

**Register value (binary)** *(for value match)*

Binary representation of the valid bits in the register as well as the parameterized value.

Dependent on **Bit count**, **Offset from right** and **Register value**.

**Register value (hexadecimal)** *(for value match)*

Hexadecimal representation of the valid bits in the register as well as the parameterized value.

Dependent on **Bit count**, **Offset from right** and **Register value**.

**Behavior on value lower**

This parameter defines the behavior of the word register in the event that the value received (KNX) is less than the parameterised value. The following options are available:

- No reaction
- Set register value

**Register value** *(for value lower)*

The value which is set in the word register.

**Register value (binary)** *(for value lower)*

Binary representation of the valid bits in the register as well as the parameterized value.

Dependent on **Bit count**, **Offset from right** and **Register value**.

**Register value (hexadecimal)** *(for value lower)*

Hexadecimal representation of the valid bits in the register as well as the parameterized value.

Dependent on **Bit count**, **Offset from right** and **Register value**.

### 8.19.3.2 Direction – Modbus to KNX

--- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	DPT 12 - Configured - 4 bytes
Datapoints 11 - 20	Description	
Datapoints 21 - 30	Direction	<input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX
Datapoints 31 - 40	Send condition	On change
Datapoints 41 - 50	Type	Value in word register
Datapoints 51 - 60	Bit count	16 bit
Datapoints 61 - 70	Offset from right	00 bit
Datapoints 71 - 80	Threshold	1
Datapoints 81 - 90	Threshold (binary)	0000 0000 0000 0001b
Datapoints 91 - 100	Threshold (hexadecimal)	0001h
Datapoints 101 - 110	Behavior on value higher	<input type="radio"/> No reaction <input checked="" type="radio"/> Send value
Datapoints 111 - 120	Object value	0
Datapoints 121 - 130	Behavior on value match	<input type="radio"/> No reaction <input checked="" type="radio"/> Send value
Datapoints 131 - 140	Object value	4294967295
Datapoints 141 - 150	Behavior on value lower	<input type="radio"/> No reaction <input checked="" type="radio"/> Send value
Datapoints 151 - 160	Object value	0
Datapoints 161 - 170	Function	<input checked="" type="radio"/> Read holding registers - 03 <input type="radio"/> Read input registers - 04
Datapoints 171 - 180	Address	0
Datapoints 181 - 190	Polling interval	Every cycle

#### Threshold

This parameter defines the value for which the word register is checked.

#### Threshold (binary)

Binary representation of the valid bits in the register as well as the parameterized value.

Dependent on **Bit count**, **Offset from right** and **Threshold**.

#### Threshold (hexadecimal)

Hexadecimal representation of the valid bits in the register as well as the parameterized value.

Dependent on **Bit count**, **Offset from right** and **Threshold**.

#### Behavior on value higher

This parameter defines the behavior of the group object (KNX) in the event that the register value is greater than the parameterised value. The following options are available:

- No reaction
- Send value

**Object value** *(for value higher)*

The value that is sent via the group object (KNX).

**Behavior on value match**

This parameter defines the behavior of the group object (KNX) in the event that the register value corresponds to the parameterised value. The following options are available:

- No reaction
- Send value

**Object value** *(for value match)*

The value that is sent via the group object (KNX).

**Behavior on value lower**

This parameter defines the behavior of the group object (KNX) in the event that the register value is less than the parameterised value. The following options are available:

- No reaction
- Send value

**Object value** *(for value lower)*

The value that is sent via the group object (KNX).

## 8.20 Channel function “DPT 12 – Unsigned – 4 bytes”

**Type**

The following type is configured:

- Double word register  
4 byte unsigned value (KNX) is written/read to/from area in double word register (modbus)

## 8.20.1 Type – Double word register

--- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	DPT 12 - Unsigned - 4 bytes
Datapoints 11 - 20	Description	
Datapoints 21 - 30	Direction	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX
Datapoints 31 - 40	Type	Double word register
Datapoints 41 - 50	Word order	<input checked="" type="radio"/> Hi word @ address / Lo word @ address + 1 <input type="radio"/> Lo word @ address / Hi word @ address + 1
Datapoints 51 - 60	Position in double word register	<input type="radio"/> All 4 bytes <input checked="" type="radio"/> Configured
Datapoints 61 - 70	Bit count	32 bit
Datapoints 71 - 80	Offset from right	00 bit
Datapoints 81 - 90	Valid bits in word register @ address	VVV VVV VVV VVVb
Datapoints 91 - 100	Valid bits in word register @ address + 1	VVV VVV VVV VVVb
Datapoints 101 - 110	Valid values	0 ... 4294967295
Datapoints 111 - 120	Function	Write single holding register - 06
	Address	0

### Word order

This parameter defines the byte order in which the value of the group object (KNX) is distributed to the two word registers (modbus). The following options are available:

- Hi word @ address / Lo word @ address + 1
- Lo word @ address / Hi word @ address + 1

### Position in double word register

This parameter defines the range of the double word register which is written/read. The following areas are available:

- All 4 bytes
- Configured

### Bit count *(only for configured)*

This parameter defines the size of the area in the Double word register (in bits).

### Offset from right *(only for configured)*

This parameter defines the position of the range in the Double word register (offset from right in bits).

### Valid bits in word register @ address *(only for configured)*

Indicates which bits have been defined in the word register (address).

Depending on **Word order**, **Bit count** and **Offset from right**.

### Valid bits in word register @ address + 1 (only for configured)

Indicates which bits have been defined in the word register (address + 1).  
Depending on **Word order**, **Bit count** and **Offset from right**.

### Valid values (only for configured)

Indicates which values fit into the defined bits.  
Depending on **Bit count** and **Offset from right**.



**Bit count** and **Offset from right** must not be greater than 32 together.  
The value must fit into **Bit count**, e.g. **Bit count** = 1 → “Value” = 0 or 1.

### Function (as “Modbus master”, for “KNX to modbus” and “Double word register”)

By means of **Word register write requests** the transmission type of the Double word register can be configured. The following options are available:

- Write single holding register – 06  
One word register in one request
- Write multi holding registers – 16  
Both word registers in one request



As “Modbus master”, for “Modbus to KNX” and “Double word register”, multi read requests should be enabled to read both word registers in one request.

### Address (for “Double word register”)

Double word registers use the register address specified here plus this register address + 1.

## 8.21 Channel function “DPT 13 – Signed – 4 bytes”

### Type

The following type is configured:

- Double word register  
4 byte signed value (KNX) is written/read to/from area in double word register (modbus)

## 8.21.1 Type – Double word register

--- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	<input type="text"/>
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
– Datapoints	Channel 1	
– Datapoints 1 - 10	Datapoint type	DPT 13 - Signed - 4 bytes
Datapoints 11 - 20	Description	<input type="text"/>
Datapoints 21 - 30	Direction	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX
Datapoints 31 - 40	Type	Double word register
Datapoints 41 - 50	Word order	<input checked="" type="radio"/> Hi word @ address / Lo word @ address + 1 <input type="radio"/> Lo word @ address / Hi word @ address + 1
Datapoints 51 - 60	Position in double word register	All 4 bytes
Datapoints 61 - 70	Function	Write single holding register - 06
Datapoints 71 - 80	Address	0

### Word order

This parameter defines the byte order in which the value of the group object (KNX) is distributed to the two word registers (modbus). The following options are available:

- Hi word @ address / Lo word @ address + 1
- Lo word @ address / Hi word @ address + 1

### Position in double word register

This parameter defines the range of the double word register which is written/read. The following range is configured:

- All 4 bytes

### Function (as “Modbus master”, for “KNX to modbus” and “Double word register”)

By means of **Word register write requests** the transmission type of the double word register can be configured. The following options are available:

- Write single holding register – 06  
One word register in one request
- Write multi holding registers – 16  
Both word registers in one request



As “Modbus master”, for “Modbus to KNX” and “Double word register”, multi read requests should be enabled to read both word registers in one request.

### Address (for “Double word register”)

Double word registers use the register address specified here plus this register address + 1.

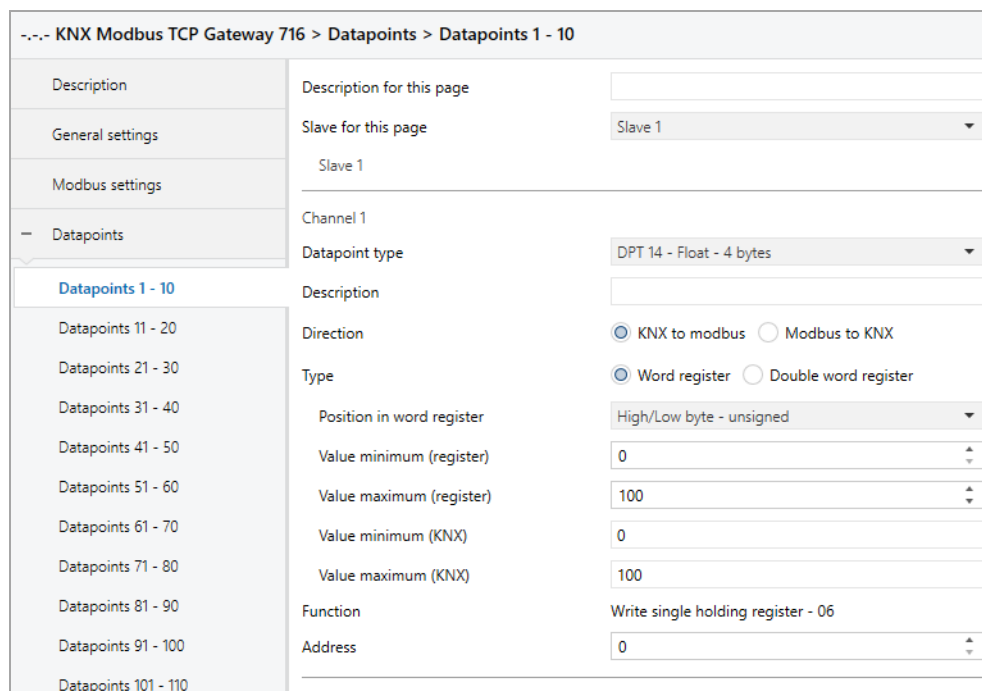
## 8.22 Channel function “DPT 14 – Float – 4 bytes”

### Type

The following types can be configured:

- Word register  
4 byte float value (KNX) is mapped to area in word register (modbus)
- Double word register  
4 byte float value (KNX) is mapped to two word registers (modbus)

### 8.22.1 Type – Word register



KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10		
Description	Description for this page	
General settings	Slave for this page: Slave 1	
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type: DPT 14 - Float - 4 bytes	
Datapoints 11 - 20	Description	
Datapoints 21 - 30	Direction: <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX	
Datapoints 31 - 40	Type: <input checked="" type="radio"/> Word register <input type="radio"/> Double word register	
Datapoints 41 - 50	Position in word register: High/Low byte - unsigned	
Datapoints 51 - 60	Value minimum (register): 0	
Datapoints 61 - 70	Value maximum (register): 100	
Datapoints 71 - 80	Value minimum (KNX): 0	
Datapoints 81 - 90	Value maximum (KNX): 100	
Datapoints 91 - 100	Function: Write single holding register - 06	
Datapoints 101 - 110	Address: 0	

### Position in word register

This parameter defines the range of the word register which is mapped. The following ranges are available:

- Low byte – unsigned
- High byte – unsigned
- High/Low byte – unsigned
- Low byte – 2th complement
- High byte – 2th complement
- High/Low byte – 2th complement

### Value minimum (register)

Register value, which corresponds to **Value minimum (KNX)**.

### Value maximum (register)

Register value, which corresponds to **Value maximum (KNX)**.

### Value minimum (KNX)

KNX value, which corresponds to **Value minimum (register)**.

### Value maximum (KNX)

KNX value, which corresponds to **Value maximum (register)**.



*The conversion is always transferred to the entire register range.  
**Value minimum/maximum (register)** does not define any limits.*

## 8.22.2 Type – Double word register

--- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	DPT 14 - Float - 4 bytes
Datapoints 11 - 20	Description	
Datapoints 21 - 30	Direction	<input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX
Datapoints 31 - 40	Type	<input type="radio"/> Word register <input checked="" type="radio"/> Double word register
Datapoints 41 - 50	Word order	<input checked="" type="radio"/> Hi word @ address / Lo word @ address + 1 <input type="radio"/> Lo word @ address / Hi word @ address + 1
Datapoints 51 - 60	Type register value	Modbus holds integer value - unsigned
Datapoints 61 - 70	Value minimum (register)	0
Datapoints 71 - 80	Value maximum (register)	100
Datapoints 81 - 90	Value minimum (KNX)	0
Datapoints 91 - 100	Value maximum (KNX)	100
Datapoints 101 - 110	Function	Write single holding register - 06
Datapoints 111 - 120	Address	0

### Word order

This parameter defines the byte order in which the value of the group object (KNX) is distributed to the two word registers (modbus). The following options are available:

- Hi word @ address / Lo word @ address + 1
- Lo word @ address / Hi word @ address + 1

### Type register value

Here you define how the float value is to be mapped to modbus. The following options are available:

- Modbus holds integer value – unsigned
- Modbus holds integer value – 2th complement
- Modbus holds float value – IEEE

**Value minimum (register)** *(only for integer value)*

Register value, which corresponds to **Value minimum (KNX)**.

**Value maximum (register)** *(only for integer value)*

Register value, which corresponds to **Value maximum (KNX)**.

**Value minimum (KNX)** *(only for integer value)*

KNX value, which corresponds to **Value minimum (register)**.

**Value maximum (KNX)** *(only for integer value)*

KNX value, which corresponds to **Value maximum (register)**.



*The conversion is always transferred to the entire register range.*

**Value minimum/maximum (register)** does not define any limits.

**Scaling factor** *(only for float value)*

A scaling factor can be specified here, which is applied when converting from KNX to modbus and from modbus to KNX.

**Function** *(as “Modbus master”, for “KNX to modbus” and “Double word register”)*

By means of **Word register write requests** the transmission type of the double word register can be configured. The following options are available:

- Write single holding register – 06  
One word register in one request
- Write multi holding registers – 16  
Both word registers in one request



*As “Modbus master”, for “Modbus to KNX” and “Double word register”, multi read requests should be enabled to read both word registers in one request.*

**Address** *(for “Double word register”)*

Double word registers use the register address specified here plus this register address + 1.

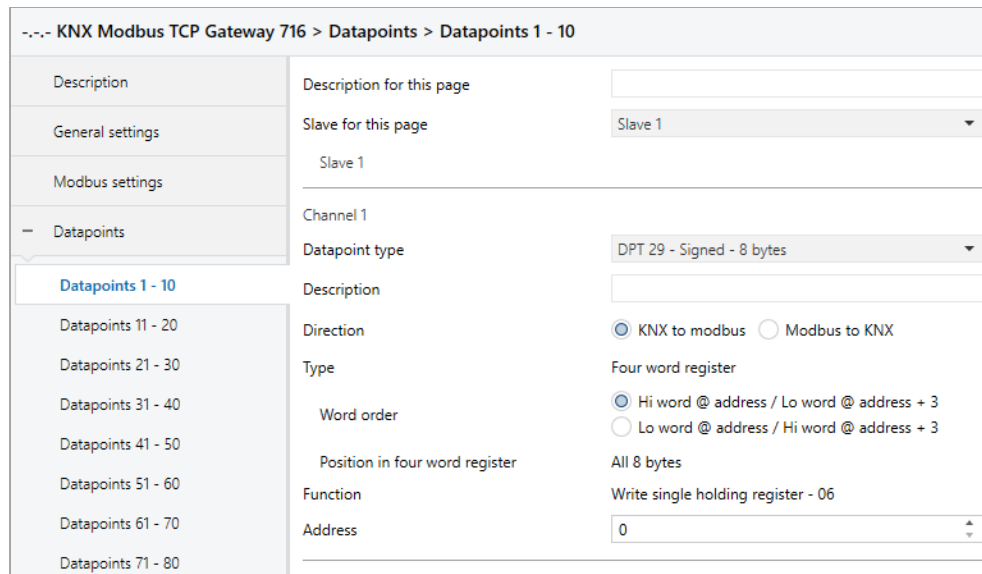
## 8.23 Channel function “DPT 29 – Signed – 8 bytes”

### Type

The following type is configured:

- Four word register  
8 byte signed value (KNX) is written/read to/from area in four word register (modbus)

### 8.23.1 Type – Four word register



-.- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10	
Description	Description for this page
General settings	Slave for this page Slave 1
Modbus settings	Channel 1
Datapoints	Datapoint type DPT 29 - Signed - 8 bytes
Datapoints 1 - 10	Description
Datapoints 11 - 20	Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX
Datapoints 21 - 30	Type Four word register
Datapoints 31 - 40	Word order <input checked="" type="radio"/> Hi word @ address / Lo word @ address + 3 <input type="radio"/> Lo word @ address / Hi word @ address + 3
Datapoints 41 - 50	Position in four word register All 8 bytes
Datapoints 51 - 60	Function Write single holding register - 06
Datapoints 61 - 70	Address 0
Datapoints 71 - 80	

### Word order

This parameter defines the byte order in which the value of the group object (KNX) is distributed to the four word registers (modbus). The following options are available for selection:

- Hi word @ address / Lo word @ address + 3
- Lo word @ address / Hi word @ address + 3

### Position in four word register

This parameter defines the range of the four word register which is written/read. The following range is configured:

- All 8 bytes

**Function** (as “Modbus master”, for “KNX to modbus” and “Four word register”)

By means of **Word register write requests** the transmission type of the four word register can be configured. The following options are available:

- Write single holding register – 06  
One word register in one request
- Write multi holding registers – 16  
Four word registers in one request



As “Modbus master”, for “Modbus to KNX” and “Four word register”, multi read requests should be enabled to read all four word registers in one request.

**Address** (for “Four word register”)

Four word registers use the register address specified here, register address + 1, register address + 2 and register address + 3.

## 8.24 Channel function “Fixed value – 1 bit”

### Type

The following type is configured:

- Bit register  
1 bit (parameter) sets bit register (modbus)

### 8.24.1 Type – Bit register

--- KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10		
Description	Description for this page	
General settings	Slave for this page	
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	Fixed value - 1 bit
Datapoints 11 - 20	Description	
Datapoints 21 - 30	Type	Bit register
Datapoints 31 - 40	Register value	<input checked="" type="radio"/> Bit in register - Value '1' <input type="radio"/> Bit in register - Value '0'
Datapoints 41 - 50	Function	Write single coil - 05
Datapoints 51 - 60	Address	0
Datapoints 61 - 70	Write interval	<input checked="" type="radio"/> Once on startup <input type="radio"/> On startup and cyclically

### Register value

The value which is set in the register. The following are available for selection:

- Bit in register – Value ‘1’
- Bit in register – Value ‘0’

## 8.25 Channel function “Fixed value – 2 bytes”

### Type

The following type is configured:

- Word register  
Value (parameter) sets word register (modbus)

### 8.25.1 Type – Word register

KNX Modbus TCP Gateway 716 > Datapoints > Datapoints 1 - 10

Description	Description for this page	
General settings	Slave for this page	Slave 1
Modbus settings	Slave 1	
Datapoints	Channel 1	
Datapoints 1 - 10	Datapoint type	Fixed value - 2 bytes
Datapoints 11 - 20	Description	
Datapoints 21 - 30	Type	Word register
Datapoints 31 - 40	Bit count	16 bit
Datapoints 41 - 50	Offset from right	00 bit
Datapoints 51 - 60	Register value	0
Datapoints 61 - 70	Register value (binary)	0000 0000 0000 0000b
Datapoints 71 - 80	Register value (hexadecimal)	0000h
Datapoints 81 - 90	Function	Write single holding register - 06
Datapoints 91 - 100	Address	0
	Write interval	<input checked="" type="radio"/> Once on startup <input type="radio"/> On startup and cyclically

### Bit count

This parameter defines the size of the value in the word register (in bits).

### Offset from right

This parameter defines the position of the value in the word register (offset from right in bits).

### Register value

The value which is set in the register.

### Register value (binary)

Binary representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

### Register value (hexadecimal)

Hexadecimal representation of the valid bits in the register as well as the parameterized value.  
Dependent on **Bit count**, **Offset from right** and **Register value**.

8.26 Converter N – M

KNX Modbus TCP Gateway 716 > DPT converter > Converters 1 - 10

Description

General settings

Modbus settings

+ Datapoints

- DPT converter

Converters 1 - 10

Converters 11 - 20

Converters 21 - 30

Converters 31 - 40

Converters 41 - 50

Converters 51 - 60

Converters 61 - 70

Converter 1

Description

Datapoint type - Input

Converter 2

Description

Datapoint type - Input

Converter 3

Description

Datapoint type - Input

Converter 4

Description

Datapoint type - Input

Disabled

Disabled

DPT 01 - Binary - 1 bit

DPT 05 - Percent - 1 byte

DPT 05 - Unsigned - 1 byte

DPT 06 - Signed - 1 byte

DPT 07 - Unsigned - 2 bytes

DPT 08 - Signed - 2 bytes

DPT 09 - Float - 2 bytes

DPT 12 - Unsigned - 4 bytes

DPT 13 - Signed - 4 bytes

DPT 14 - Float - 4 bytes

DPT 29 - Signed - 8 bytes

Disabled

10 converters are combined per page.

Description (30 characters)

Any name can be assigned to the converter. However, this should be unique and meaningful, this makes it easier to work with the associated group objects later, as the assigned name is displayed there as a designation. If no name is assigned, the group objects are designated with “Converter N: ... ”.

## Datapoint type – Input

This parameter activates and defines the input object and the function of this converter. The following options are available:

- Disabled
- DPT 01 – Binary – 1 bit

Group object	Type KNX	Size	Direction
GO 256 Converter 1: Input – Binary – 1 bit	1.001	1 bit	From KNX

- DPT 05 – Percent – 1 byte

Group object	Type KNX	Size	Direction
GO 256 Converter 1: Input – Percent – 1 byte	5.001	1 byte	From KNX

- DPT 05 – Unsigned – 1 byte

Group object	Type KNX	Size	Direction
GO 256 Converter 1: Input – Unsigned – 1 byte	5.010	1 byte	From KNX

- DPT 06 – Signed – 1 byte

Group object	Type KNX	Size	Direction
GO 256 Converter 1: Input – Signed – 1 byte	6.010	1 byte	From KNX

- DPT 07 – Unsigned – 2 bytes

Group object	Type KNX	Size	Direction
GO 256 Converter 1: Input – Unsigned – 2 bytes	7.001	2 bytes	From KNX

- DPT 08 – Signed – 2 bytes

Group object	Type KNX	Size	Direction
GO 256 Converter 1: Input – Signed – 2 bytes	8.001	2 bytes	From KNX

- DPT 09 – Float – 2 bytes

Group object	Type KNX	Size	Direction
GO 256 Converter 1: Input – Float – 2 bytes	9.001	2 bytes	From KNX

- DPT 12 – Unsigned – 4 bytes

Group object	Type KNX	Size	Direction
GO 256 Converter 1: Input – Unsigned – 4 bytes	12.001	4 bytes	From KNX

- DPT 13 – Signed – 4 bytes

Group object	Type KNX	Size	Direction
GO 256 Converter 1: Input – Signed – 4 bytes	13.001	4 bytes	From KNX

- DPT 14 – Float – 4 bytes

Group object	Type KNX	Size	Direction
GO 256 Converter 1: Input – Float – 4 bytes	14.000	4 bytes	From KNX

- DPT 29 – Signed – 8 bytes

Group object	Type KNX	Size	Direction
GO 256 Converter 1: Input – Signed – 8 bytes	29.010	8 bytes	From KNX

### Datapoint type – Output

This parameter defines the output object of this converter. The selection depends on the selected input object. The following options are available:

- DPT 01 – Binary – 1 bit

Not for **Datapoint type – Input** “DPT 29 – Signed – 8 bytes”.

Group object	Type KNX	Size	Direction
GO 257 Converter 1: Output – Binary – 1 bit	1.001	1 bit	To KNX

- DPT 05 – Percent – 1 byte

For **Datapoint type – Input** “DPT 1 – Binary – 1 bit”

For **Datapoint type – Input** “DPT 5 – Percent – 1 byte”.

Group object	Type KNX	Size	Direction
GO 257 Converter 1: Output – Percent – 1 byte	5.001	1 byte	To KNX

- DPT 05 – Unsigned – 1 byte

Not for **Datapoint type – Input** “DPT 5 – Percent – 1 byte”.

Group object	Type KNX	Size	Direction
GO 257 Converter 1: Output – Unsigned – 1 byte	5.010	1 byte	To KNX

- DPT 06 – Signed – 1 byte

Not for **Datapoint type – Input** “DPT 5 – Percent – 1 byte”.

Group object	Type KNX	Size	Direction
GO 257 Converter 1: Output – Signed – 1 byte	6.010	1 byte	To KNX

- DPT 07 – Unsigned – 2 bytes

Not for **Datapoint type – Input** “DPT 5 – Percent – 1 byte”.

Group object	Type KNX	Size	Direction
GO 257 Converter 1: Output – Unsigned – 2 bytes	7.001	2 bytes	To KNX

- DPT 08 – Signed – 2 bytes

Not for **Datapoint type – Input** “DPT 5 – Percent – 1 byte”.

Group object	Type KNX	Size	Direction
GO 257 Converter 1: Output – Signed – 2 bytes	8.001	2 bytes	To KNX

- DPT 09 – Float – 2 bytes

Not for **Datapoint type – Input** “DPT 5 – Percent – 1 byte”.

Group object	Type KNX	Size	Direction
GO 257 Converter 1: Output – Float – 2 bytes	9.001	2 bytes	To KNX

- DPT 12 – Unsigned – 4 bytes

Not for **Datapoint type – Input** “DPT 5 – Percent – 1 byte”.

Group object	Type KNX	Size	Direction
GO 257 Converter 1: Output – Unsigned – 4 bytes	12.001	4 bytes	To KNX

- DPT 13 – Signed – 4 bytes

Not for **Datapoint type – Input** “DPT 5 – Percent – 1 byte”.

Group object	Type KNX	Size	Direction
GO 257 Converter 1: Output – Signed – 4 bytes	13.001	4 bytes	To KNX

- DPT 14 – Float – 4 bytes

Not for **Datapoint type – Input** “DPT 5 – Percent – 1 byte”.

Group object	Type KNX	Size	Direction
GO 257 Converter 1: Output – Float – 4 bytes	14.000	4 bytes	To KNX

- DPT 18 – Scene – 1 Byte

For **Datapoint type – Input** “DPT 1 – Binary – 1Bit”.

Group object	Type KNX	Size	Direction
GO 257 Converter 1: Output – Scene – 1 byte	18.001	1 byte	To KNX

- DPT 29 – Signed – 8 bytes

Not for **Datapoint type – Input** “DPT 1 – Binary – 1 bit ”

Not for **Datapoint type – Input** “DPT 5 – Percent – 1 byte”.

Group object	Type KNX	Size	Direction
GO 257 Converter 1: Output – Signed – 8 bytes	29.010	8 bytes	To KNX

### Send cyclically

If this parameter is activated, the output is sent cyclically.

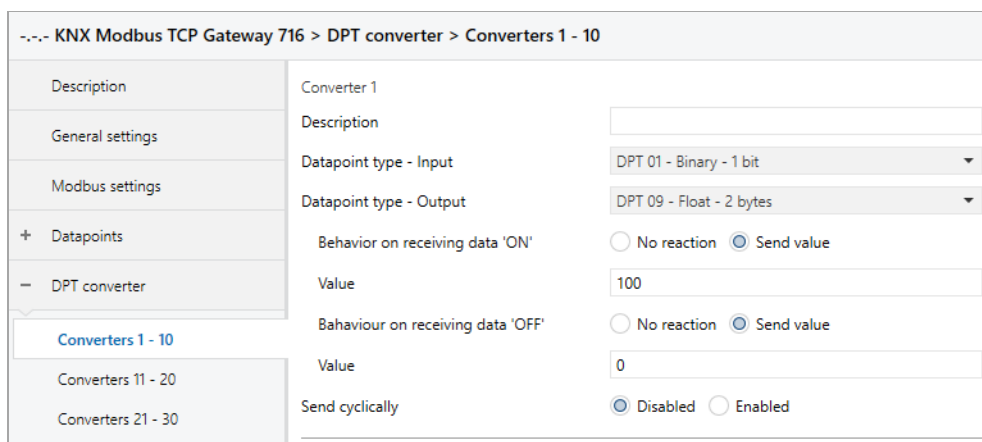
### Cycle time *(only for send cyclically)*

The time for **Send cyclically**.

## 8.27 Converter function “Binary”

### Condition:

Datapoint type – Input = “DPT 01 – Binary – 1 bit”.



--- KNX Modbus TCP Gateway 716 > DPT converter > Converters 1 - 10	
Description	Converter 1
General settings	Description
Modbus settings	Datapoint type - Input
+ Datapoints	Datapoint type - Output
- DPT converter	Behavior on receiving data 'ON'
Converters 1 - 10	Value
Converters 11 - 20	Behaviour on receiving data 'OFF'
Converters 21 - 30	Value
	Send cyclically

### Behavior on receiving data ‘ON’ (only for output not “DPT 01 – Binary – 1 bit”)

The behavior on receiving an ‘ON’ telegram is parameterised here.

The following options are available:

- No reaction
- Send value

### Value / Scene (only for output not “DPT 01 – Binary – 1 bit”)

Here you parameterise which value / scene is sent on receipt of an ‘ON’ telegram at the output.

### Behavior on receiving data ‘OFF’ (only with output not “DPT 01 – Binary – 1 bit”)

The behavior on receiving an ‘OFF’ telegram is parameterised here.

The following options are available:

- No reaction
- Send value

### Value / Scene (only for output not “DPT 01 – Binary – 1 bit”)

Here you parameterise which value / scene is sent on receipt of an ‘OFF’ telegram at the output.

### Value inverted (only for output “DPT 01 – Binary – 1 bit”)

This parameter defines whether the input value is to be sent inverted at the output.

## 8.28 Converter function “Threshold”

### Condition:

Datapoint type – Input != “DPT 01 – Binary – 1 bit”.

Datapoint type – Input != “DPT 29 – Signed – 8 bytes”.

Datapoint type – Output = “DPT 01 – Binary – 1 bit”.

--- KNX Modbus TCP Gateway 716 > DPT converter > Converters 1 - 10	
Description	Converter 1
General settings	Description <input type="text"/> Datapoint type - Input <span>DPT 09 - Float - 2 bytes</span> Datapoint type - Output <span>DPT 01 - Binary - 1 bit</span> Threshold <input type="text" value="0"/> Behavior on value higher <span>Send 'ON'</span> Behavior on value match <span>Send 'OFF'</span> Behavior on value lower <span>Send 'OFF'</span> Send cyclically <input checked="" type="radio"/> Disabled <input type="radio"/> Enabled
Modbus settings	
+ Datapoints	
- DPT converter	
Converters 1 - 10	
Converters 11 - 20	
Converters 21 - 30	

### Threshold

This parameter defines the value for which the input object is checked.

### Behavior on value higher

This parameter defines the behavior at the output object in the event that the object value at the input is greater than the parameterised value. The following options are available:

- No reaction
- Send 'ON'
- Send 'OFF'

### Behavior on value match

This parameter defines the behavior at the output object in the event that the object value at the input is equal to the parameterised value. The following options are available:

- No reaction
- Send 'ON'
- Send 'OFF'

### Behavior on value lower

This parameter defines the behavior at the output object in the event that the object value at the input is less than the parameterised value. The following options are available:

- No reaction
- Send 'ON'
- Send 'OFF'

## 8.29 Converter function “Percent”

### Condition:

Datapoint type – Input = “DPT 05 – Percent – 1 byte”.

Datapoint type – Output = “DPT 05 – Percent – 1 byte”.

--- KNX Modbus TCP Gateway 716 > DPT converter > Converters 1 - 10

Description	Converter 1	
General settings	Description	<input type="text"/>
Modbus settings	Datapoint type - Input	DPT 05 - Percent - 1 byte
+ Datapoints	Datapoint type - Output	<input type="radio"/> DPT 01 - Binary - 1 bit <input checked="" type="radio"/> DPT 05 - Percent - 1 byte
- DPT converter	Percent value output [%] for input = 0%	0 / 0x00 / 0.0%
	Percent value output [%] for input = 100%	255 / 0xFF / 100.0%
Converters 1 - 10	Send cyclically	<input checked="" type="radio"/> Disabled <input type="radio"/> Enabled
Converters 11 - 20		

### Percent value output [%]

#### for input = 0%

Percent value for output object, which corresponds to 0 % at the input object.

### Percent value output [%]

#### for input = 100%

Percent value for output object, which corresponds to 100 % at the input object.

### Example:

Percent value output [%] for input = 0% = “51 / 0x33 / 20.0%”.

Percent value output [%] for input = 100% = “204 / 0xCC / 80.0%”.

The value range of the input (0 % ... 100 %) is mapped to the value range of the output (20 % ... 80 %).

## 8.30 Converter function “Scaling”

### Condition:

Datapoint type – Input != “DPT 01 – Binary – 1 bit”.

Datapoint type – Input != “DPT 05 – Percent – 1 byte”.

Datapoint type – Output != “DPT 01 – Binary – 1 bit”

--- KNX Modbus TCP Gateway 716 > DPT converter > Converters 1 - 10

Description	Converter 1	
General settings	Description	<input type="text"/>
Modbus settings	Datapoint type - Input	DPT 09 - Float - 2 bytes
+ Datapoints	Datapoint type - Output	DPT 09 - Float - 2 bytes
- DPT converter	Scaling factor	<input type="text" value="1"/>
	<div> <i>i</i> Input * Scaling factor = Output         </div>	
Converters 1 - 10	Send cyclically	<input checked="" type="radio"/> Disabled <input type="radio"/> Enabled
Converters 11 - 20		

### Scaling factor

A scaling factor can be specified here, which is applied when converting from input to output.

## 8.31 General information

### 8.31.1 Scaling

With the respective minimum/maximum values the scaling factor can be defined.

#### Example:

*Value minimum (register) = 0*

*Value maximum (register) = 100*

*Value minimum (KNX) = 0*

*Value maximum (KNX) = 10*

This gives a scaling \* 10 of the KNX value:

Value KNX = 10.5 Value → Register = 105



*The conversion is always transferred to the entire register range.*

***Value minimum/maximum (register) does not define any limits.***

### 8.31.2 2th complement

The 2th complement is used with modbus registers for the representation of negative numbers. Thus, for example, a range of -32768...32767 can be represented on a word register.

### 8.31.3 Modbus communication

If the KNX gateway (master mode) does not receive a response from the slave within 1 second, the request is repeated twice. If these are not successful, all channels of this parameter page are skipped.

If the slave needs more than 1 second to process the data, it can send an acknowledge telegram, which restarts the time interval at the master.

### 8.31.4 Modbus specification

In Modbus there are various types of register address specification.

#### Variant 1:

Register type	Access	Size	Address range
Coil	RW	1 bit	00001 – 09999
Discrete inputs	R	1 bit	10001 – 19999
Input register	R	2 bytes	30001 – 39999
Holding register	RW	2 bytes	40001 – 49999



*The address range additionally defines the register type.  
The address range is 1 based.*

#### Example – Coil:

00005 is the fifth coil.

*Register address = 1 based*

*Function = Read coils – 01*

*Function = Write single coil – 05*

*Function = Write multi coils – 15*

*Address = 5*

#### Example – Discrete input:

10001 is the first discrete input.

*Register address = 1 based*

*Function = Read discrete inputs – 02*

*Address = 1*

#### Example – Input register:

30002 is the second input register.

*Register address = 1 based*

*Function = Read input register – 04*

*Address = 2*

#### Example – Holding register:

40004 is the fourth holding register.

*Register address = 1 based*

*Function = Read holding register – 03*

*Function = Write single holding register – 06*

*Function = Write multi holding register – 16*

*Address = 4*

## Variant 2:

Register type	Access	Size	Address range
Coil	RW	1 bit	0x0001 – 0x9999
Discrete inputs	R	1 bit	1x0001 – 1x9999
Input register	R	2 bytes	3x0001 – 3x9999
Holding register	RW	2 bytes	4x0001 – 4x9999



*The address range additionally defines the register type.  
The address range is 1 based.*

### Example – Coil:

0x0005 is the fifth coil.

*Register address = 1 based*

*Function = Read coils – 01*

*Function = Write single coil – 05*

*Function = Write multi coils – 15*

*Address = 5*

### Example – Discrete input:

1x0001 is the first discrete input.

*Register address = 1 based*

*Function = Read discrete inputs – 02*

*Address = 1*

### Example – Input register:

3x0002 is the second input register.

*Register address = 1 based*

*Function = Read input register – 04*

*Address = 2*

### Example – Holding register:

4x0004 is the fourth holding register.

*Register address = 1 based*

*Function = Read holding register – 03*

*Function = Write single holding register – 06*

*Function = Write multi holding register – 16*

*Address = 4*

### Variant 3:

Register type	Access	Size	Address range
Coil	RW	1 bit	0 – 65535
Discrete inputs	R	1 bit	0 – 65535
Input register	R	2 bytes	0 – 65535
Holding register	RW	2 bytes	0 – 65535



*The address range defines the address that is actually sent.  
The address range is 0 based.*

#### Example – Coil:

5 is the sixth coil.

*Register address = 0 based*

*Function = Read coils – 01*

*Function = Write single coil – 05*

*Function = Write multi coils – 15*

*Address = 5*

#### Example – Discrete input:

0 is the first discrete input.

*Register address = 0 based*

*Function = Read discrete inputs – 02*

*Address = 0*

#### Example – Input register:

2 is the third input register.

*Register address = 0 based*

*Function = Read input register – 04*

*Address = 2*

#### Example – Holding register:

4 is the fifth holding register.

*Register address = 0 based*

*Function = Read holding register – 03*

*Function = Write single holding register – 06*

*Function = Write multi holding register – 16*

*Address = 4*

#### Variant 4:

Register type	Access	Size	Address range
Coil	RW	1 bit	1 – 65535
Discrete inputs	R	1 bit	1 – 65535
Input register	R	2 bytes	1 – 65535
Holding register	RW	2 bytes	1 – 65535



*The address range is 1 based.*

#### Example – Coil:

5 is the fifth coil.

*Register address = 1 based*

*Function = Read coils – 01*

*Function = Write single coil – 05*

*Function = Write multi coils – 15*

*Address = 5*

#### Example – Discrete input:

1 is the first discrete input.

*Register address = 1 based*

*Function = Read discrete inputs – 02*

*Address = 1*

#### Example – Input register:

2 is the second input register.

*Register address = 1 based*

*Function = Read input register – 04*

*Address = 2*

#### Example – Holding register:

4 is the fourth holding register.

*Register address = 1 based*

*Function = Read holding register – 03*

*Function = Write single holding register – 06*

*Function = Write multi holding register – 16*

*Address = 4*

## 9 Open source licenses

The firmware used in this product uses the following open source software package:

- curve25519-donna (<http://code.google.com/p/curve25519-donna/>)  
Curve25519 elliptic curve, public key function

Copyright 2008, Google Inc. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**WARNING**

- The device must be mounted and commissioned by an authorized electrician.
- The prevailing safety rules must be heeded.
- The device must not be opened.
- For planning and construction of electric installations, the relevant guidelines, regulations and standards of the respective country are to be considered.

**PEAKnx GmbH**

Leydheckerstraße 10  
DE-64293 Darmstadt

Tel.: +49 6151 / 279 18 25  
[support@peaknx.com](mailto:support@peaknx.com)  
[www.peaknx.com](http://www.peaknx.com)

2025-07-01