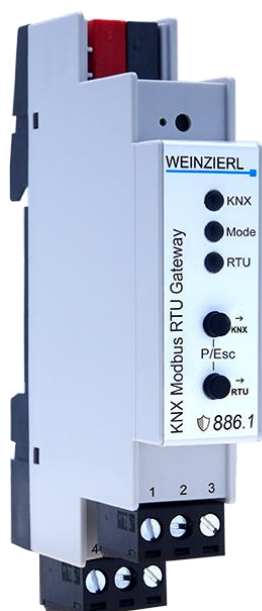


For Modbus RS-485, bus powered

## **KNX Modbus RTU Gateway 886.1 *secure***

Operation and installation manual



(Art. # 5498)

WEINZIERL ENGINEERING GmbH  
Achatz 3-4  
84508 Burgkirchen an der Alz  
GERMANY

Tel.: +49 8677 / 916 36 – 0

E-Mail: [info@weinzierl.de](mailto:info@weinzierl.de)

Web: [www.weinzierl.de](http://www.weinzierl.de)

## Content

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Application</b>                               | <b>3</b> |
| <b>2</b> | <b>Installation and connection</b>               | <b>3</b> |
| 2.1      | KNX programming mode                             | 4        |
| 2.2      | Manual operation and status display              | 4        |
| <b>3</b> | <b>Reset to factory default settings</b>         | <b>5</b> |
| <b>4</b> | <b>Wiring scheme</b>                             | <b>6</b> |
| 4.1      | Pluggable screw terminals                        | 6        |
| 4.2      | Pin assignment                                   | 6        |
| <b>5</b> | <b>KNX Security</b>                              | <b>7</b> |
| <b>6</b> | <b>ETS database</b>                              | <b>8</b> |
| 6.1      | Secure commissioning                             | 8        |
| 6.2      | Secure group communication                       | 10       |
| 6.3      | Description                                      | 12       |
| 6.4      | General settings                                 | 13       |
| 6.5      | Modbus settings                                  | 15       |
| 6.6      | Modbus test environment                          | 19       |
| 6.7      | Datapoints N – M                                 | 27       |
| 6.8      | Channel function “DPT 01 – Binary – 1 bit”       | 33       |
| 6.9      | Channel function “DPT 03 – Dimming – 4 bits”     | 37       |
| 6.10     | Channel function “DPT 05 – Percent – 1 byte”     | 45       |
| 6.11     | Channel function “DPT 05 – Configured – 1 byte”  | 46       |
| 6.12     | Channel function “DPT 05 – Unsigned – 1 byte”    | 55       |
| 6.13     | Channel function “DPT 06 – Signed – 1 byte”      | 57       |
| 6.14     | Channel function “DPT 07 – Configured – 2 bytes” | 58       |
| 6.15     | Channel function “DPT 07 – Unsigned – 2 bytes”   | 66       |
| 6.16     | Channel function “DPT 08 – Signed – 2 bytes”     | 68       |
| 6.17     | Channel function “DPT 09 – Float – 2 bytes”      | 68       |
| 6.18     | Channel function “DPT 12 – Configured – 4 bytes” | 72       |
| 6.19     | Channel function “DPT 12 – Unsigned – 4 bytes”   | 80       |
| 6.20     | Channel function “DPT 13 – Signed – 4 bytes”     | 82       |
| 6.21     | Channel function “DPT 14 – Float – 4 bytes”      | 84       |
| 6.22     | Channel function “DPT 29 – Signed – 8 bytes”     | 87       |
| 6.23     | Channel function “Fixed value – 1 bit”           | 88       |
| 6.24     | Channel function “Fixed value – 2 bytes”         | 89       |
| 6.25     | Converter N – M                                  | 90       |
| 6.26     | Converter function “Binary”                      | 94       |
| 6.27     | Converter function “Threshold”                   | 95       |
| 6.28     | Converter function “Percent”                     | 96       |
| 6.29     | Converter function “Scaling”                     | 97       |
| 6.30     | General information                              | 98       |

## 1 Application

The KNX Modbus RTU Gateway 886.1 *secure* is a compact Gateway between KNX TP and Modbus RTU with 250 freely configurable datapoints.

The device enables easy integration of Modbus devices that support the RTU protocol via RS-485 and can act as a Modbus master or slave. As Master the device can address up to 25 slave devices.

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

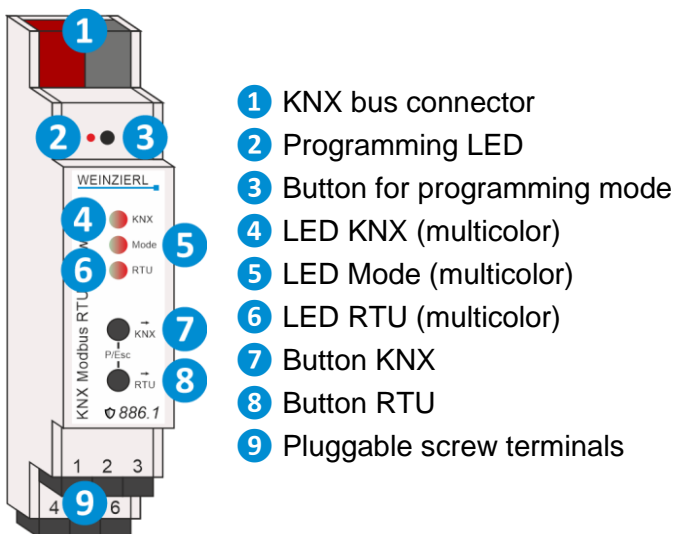
Two buttons and three LEDs allow local operation and visualization of the device status.

The gateway provides a galvanic isolation between KNX bus and Modbus.

Power is supplied via the KNX bus. The device supports KNX Security.

## 2 Installation and connection

The KNX Modbus RTU Gateway 886.1 *secure* is designed for installation on a DIN rail (35 mm) with a width of 1 unit (18 mm). An installation friendly design with pluggable screw terminals helps to reduce the cost of commissioning. It features the following controls and displays:



- 1 KNX bus connector
- 2 Programming LED
- 3 Button for programming mode
- 4 LED KNX (multicolor)
- 5 LED Mode (multicolor)
- 6 LED RTU (multicolor)
- 7 Button KNX
- 8 Button RTU
- 9 Pluggable screw terminals

The device has galvanic isolation between Modbus and KNX.



*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 (P/Esc) **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 RTU **6** lights up green when KNX bus voltage is present. When this LED flickers, telegram traffic is taking place on the Modbus.

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.

Furthermore, LED RTU **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.

Summary of the states of the LED RTU **6**:

| LED Status         | Meaning   |
|--------------------|---|
| LED lights green   | KNX bus voltage present.  |
| LED lights red     | Incorrect configuration of the register addresses in the ETS database (static error). |
| LED flickers green | Telegram traffic on the Modbus.   |
| LED briefly red    | Error in the Modbus communication.  |

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

The synchronization of the KNX objects is triggered by pressing the button KNX **7** for a long time. This is indicated by the LED Mode **5** lighting up in orange.

Pressing the button RTU **7** for a long time 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 unit operates in normal operating mode.  |
| LED lights red    | The programming mode is active.  |
| LED lights orange | The programming mode is not active.<br>Synchronization is active.  |
| LED flashes red   | The programming mode is not active.<br>Synchronization is not active.<br>The device is not loaded correctly, e.g. after aborting a download. |
| LED flashes green | The device is currently loaded by the ETS.   |

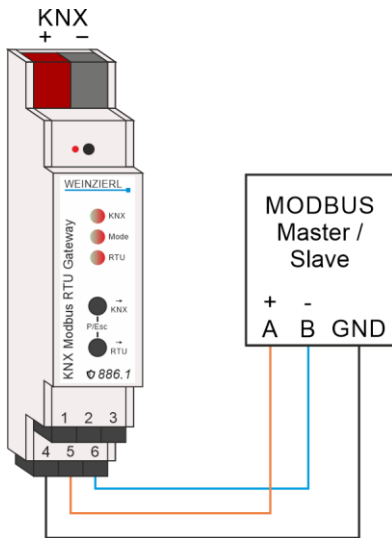
## 3 Reset to factory default settings

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 physical address 15.15.255 and no group addresses are connected. Also, KNX Data Security is disabled and the initial key (FDSK) must be used for secure commissioning.

## 4 Wiring scheme



### 4.1 Pluggable screw terminals

The upper terminal is used to connect the terminating resistor, the lower terminal to connect the Modbus (or vice versa). The terminals are identical:

|   |   |   |
|---|---|---|
| ⊥ | A | B |
| ⊥ | A | B |

### 4.2 Pin assignment

| Connection | Icon | Description  |
|------------|------|--|
| 1          | ⊥    | Ground connection for Modbus (connected to connection 4) |
| 2          | A    | Data line A (+) for Modbus (connected to connection 5)   |
| 3          | B    | Data line B (-) for Modbus (connected to connection 6)   |
| 4          | ⊥    | Ground connection for Modbus (connected to connection 1) |
| 5          | A    | Data line A (+) for Modbus (connected to connection 2)   |
| 6          | B    | Data line B (-) for Modbus (connected to connection 3)   |
| KNX        | +    | Positive connection for KNX bus                          |
| KNX        | -    | Ground connection for KNX bus                            |

The transmission line must be terminated at the last participant of the Modbus transmission line with a 120 Ohm / 0.25 W resistor. This resistor must be inserted directly between the two signal lines before the input of the last device. No terminating resistor is installed in the device itself. It can be inserted on the upper or lower screw terminal

Only shielded and twisted cables should be used for Modbus.

## 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 TP – Communication encrypted (KNX Data Security)  
Modbus RTU – Communication unencrypted*

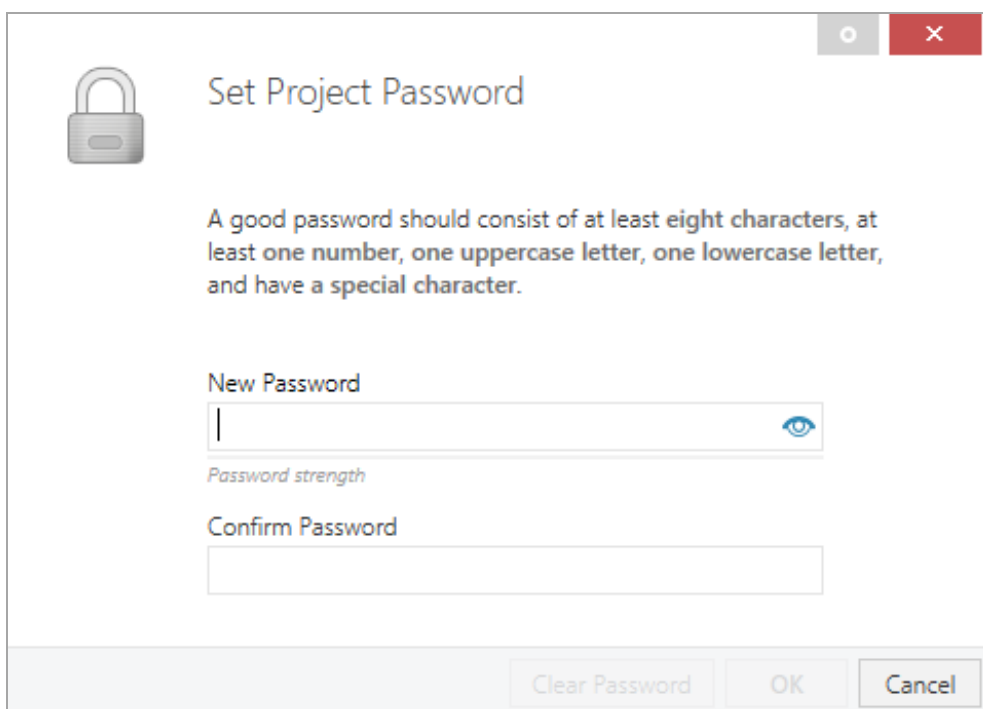
## 6 ETS database

The ETS 5 database (for ETS 5.7 or newer) can be downloaded from the product website of the KNX Modbus RTU Gateway 886.1 *secure* ([www.weinzierl.de](http://www.weinzierl.de)) or from the ETS online catalogue.

The KNX Modbus RTU Gateway 886.1 *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.

### 6.1 Secure commissioning

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



**Set Project Password**

A good password should consist of at least **eight characters**, at least **one number**, **one uppercase letter**, **one lowercase letter**, and have a **special character**.

New Password

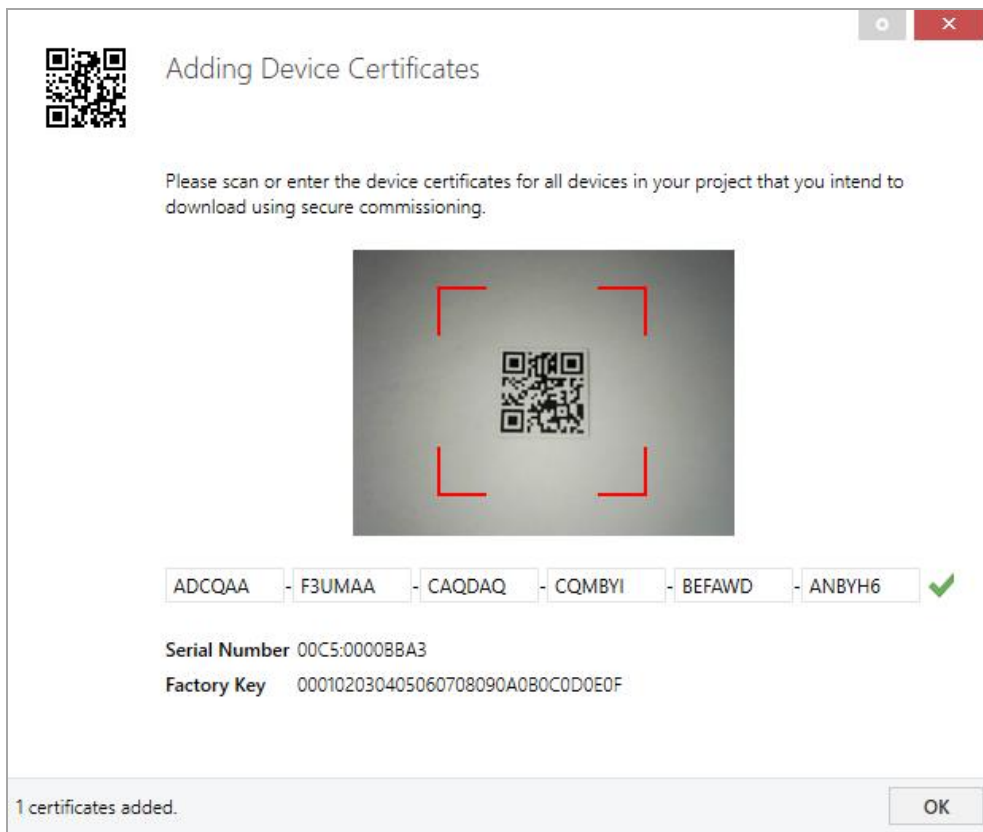
Confirm Password

Clear Password OK Cancel

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).





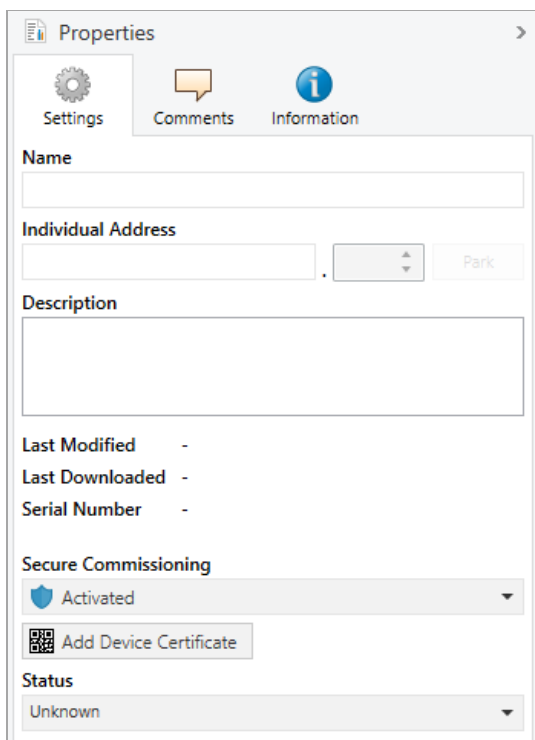
The certificate is printed as text on the device. It can also be scanned from the printed QR code via a webcam.

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.

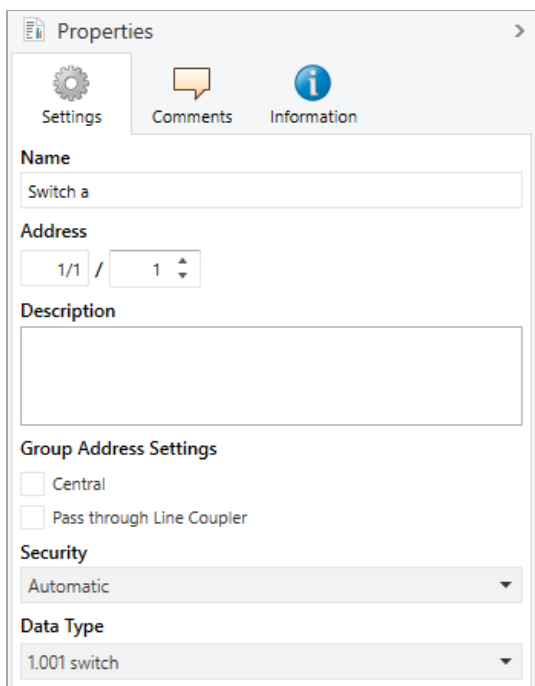
The serial number in the certificate enables the ETS to assign the correct key to a device during a download.

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









## 6.2 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:



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.

## 6.3 Description

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > Description |  |
|---|--|
| <b>Description</b>                                      |  |
| General settings  | KNX Modbus RTU Gateway 886.1 secure<br>For Modbus RS-485, bus powered  |
| Modbus settings   |  |
| + Datapoints  |  |
| + DPT converter   |  |
|   | <p>The KNX Modbus RTU Gateway 886.1 secure is a compact Gateway between KNX TP and Modbus RTU with 250 freely configurable datapoints.</p> <p>The device enables easy integration of Modbus devices that support the RTU protocol via RS-485 and can act as a Modbus master or slave. As Master the device can address up to 25 slave devices.</p> <p>The assignment between KNX objects and Modbus registers can be configured via parameters in the ETS without an additional tool.</p> <p>Two buttons and three LEDs allow local operation and visualization of the device status.</p> <p>The gateway provides a galvanic isolation between KNX bus and Modbus.</p> <p>Power is supplied via the KNX bus. The device supports KNX Security.</p> |
|   | <p><b>Wiring scheme:</b></p> <p>Please consult device data sheet and manual for further information.</p>   |
|   | <p>Contact:</p> <p>WEINZIERL ENGINEERING GmbH<br/>           Achatz 3-4<br/>           84508 Burgkirchen an der Alz<br/>           GERMANY<br/> <a href="http://www.weinzierl.de">www.weinzierl.de</a><br/> <a href="mailto:info@weinzierl.de">info@weinzierl.de</a></p>   |

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

## 6.4 General settings

1.1.1 KNX Modbus RTU Gateway 886.1 secure > General settings

| Description      | General settings   |
|------------------|--|
| General settings | <p>Device name: KNX Modbus RTU Gateway 886.1</p> <p>Send delay after bus power return: 5 s</p> <p>Heartbeat: <input checked="" type="radio"/> Disabled <input type="radio"/> Enabled</p> <p>Prog. mode on device front: <input type="radio"/> Disabled <input checked="" type="radio"/> Enabled</p> <p>Synchronization on device (Manual operation): <input type="radio"/> Disabled <input checked="" type="radio"/> Enabled</p> <p>Synchronization via objects: <input checked="" type="radio"/> Disabled <input type="radio"/> Enabled</p> <p>Synchronization via buttons:</p> <p>Sync KNX - Start      Sync modbus - Start</p> <p>Sync KNX - Stop      Sync modbus - Stop</p> |
| Modbus settings  |  |
| + Datapoints     |  |
| + DPT converter  |  |
| KNX settings     | Telegram rate limitation: 1.0 s  |

### Device name (30 characters)

Any arbitrary name can be assigned to the KNX Modbus RTU Gateway 886.1 *secure*. The device name should be meaningful, e.g. "Living room ground floor". This helps the clarity of the ETS project.

### 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.

### 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 477 Heartbeat – Trigger | 1.001    | 1 bit | To KNX    |

### 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.

## Synchronization on device (Manual operation)

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 RTU **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.

## Synchronization via object

Objects for synchronization in direction KNX and direction Modbus can be displayed here. Synchronization can be started or stopped via the respective object.

Direction KNX:

All data points of the channels that are configured “Modbus to KNX” send their current value to the KNX bus.

Direction Modbus:

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

| Group object                    | Type KNX | Size  | Direction |
|---------------------------------|----------|-------|-----------|
| GO 478 Sync KNX – Start/Stop    | 1.010    | 1 bit | From KNX  |
| GO 479 Sync Modbus – Start/Stop | 1.010    | 1 bit | From KNX  |

## Synchronization via buttons

Sync KNX – Start:

This button can be used to start the synchronization in direction KNX. All datapoints of the channels configured as “Modbus to KNX” send their current value on the KNX bus.

Sync KNX – Stop:

This button can be used to stop the synchronization in direction KNX.

Sync modbus – Start:

This button can be used to start the synchronization in direction modbus. If the gateway is operating as Modbus master, all registers of the channels configured “KNX to Modbus” are written to Modbus again.

Sync modbus – Stop:

This button can be used to stop the synchronization in direction modbus.

## 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.*

## 6.5 Modbus settings

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > Modbus settings |   |
|---|---|
| Description   | Modbus settings   |
| General settings  |   |
| <b>Modbus settings</b>                                      |   |
| + Datapoints  |   |
| + DPT converter   |   |
|   | KNX Gateway <input checked="" type="radio"/> Modbus master <input type="radio"/> Modbus slave<br>Slave address (common) <input type="text" value="1"/><br>Baudrate <input type="text" value="19200 bits/s"/><br>Parity <input type="text" value="Even (1 stop bit)"/><br>Byte order (data in word register) <input checked="" type="radio"/> MSB first <input type="radio"/> LSB first<br>Register address <input checked="" type="radio"/> 0 based <input type="radio"/> 1 based<br><hr/> Request settings<br>Time till next request <input type="text" value="Automatic"/><br>Time till next cycle <input type="text" value="Automatic"/><br>Multi read requests <input type="radio"/> Disabled <input checked="" type="radio"/> Enabled<br>Bit register write requests <input checked="" type="radio"/> Write single coil - 05 <input type="radio"/> Write multi coils - 15<br>Word register write requests <input checked="" type="radio"/> Write single holding register - 06 <input type="radio"/> Write multi holding registers - 16<br><hr/> Diagnostic settings<br>Diagnostic objects <input checked="" type="radio"/> Disabled <input type="radio"/> Enabled<br>Modbus test environment <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

## Slave address (common)

Here the general slave address (0 ... 247) is set and used according to KNX Gateway configuration.

Modbus master:

The slave address of the Modbus communication partner is entered here. If several Slave devices are to be addressed, a specific slave address can be assigned per parameter page (e.g. datapoints 1-10).

Modbus slave:

The slave address of the KNX gateway is entered here.

## Baudrate

Configures the baudrate of the Modbus communication. The following are available:

- 1200 bits/s
- 2400 bits/s
- 4800 bits/s
- 9600 bits/s
- 19200 bits/s
- 38400 bits/s
- 56000 bits/s
- 57600 bits/s
- 115200 bits/s

## Parity

Here the Modbus frame is parameterized with regard to parity and stop bit. The following options are available:

- Even (1 stop bit)
- Odd (1 stop bit)
- None (2 stop bits)
- None (1 stop bit)

## 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

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

This parameter defines the minimum time for the next request. Used to slow down the master.

## Time till next cycle *(only in Master Mode)*

This parameter defines the time after which a new request cycle has to begin. If the time is too short, the cycle will not be completed. This is indicated by the red flashing of the LED RTU **6**.

## Multi read requests *(only in Master Mode)*

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



*The slave address 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:

*Type Slave address: Common*

*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



*Some modbus devices on the market support only one of these function codes. This is usually the function code for multi write access. This can be set here.*

## 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



*Some modbus devices on the market support only one of these function codes. This is usually the function code for multi write access. This can be set here.*

## 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 diagnostic object.

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 476 Diagnostic: Slave (common) – No communication | 1.001    | 1 bit | To KNX    |

## Modbus test environment

This parameter can be used to activate the modbus test environment. The Modbus communication can then be tested in advance here.

## 6.6 Modbus test environment

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Modbus settings > Modbus test environment

|   |                                    |  |
|---|------------------------------------|--|
| Description   | Modbus test environment            |  |
| General settings  | Baudrate                           | 19200 bits/s   |
| Modbus settings   | Parity                             | Even (1 stop bit)  |
| <b>Modbus test environment</b>  |                                    |  |
| + Datapoints  | Modbus request                     |  |
| + DPT converter   | Slave address                      | 1  |
|   | Function                           | Read holding registers - 03  |
|   | Address                            | 2  |
|   | Type                               | Word register  |
|   | Position in word register          | High/Low byte - unsigned   |
|   | Byte order (data in word register) | <input checked="" type="radio"/> MSB first <input type="radio"/> LSB first |
|   | Modbus request (preview)           | 01 03 00 02 00 01h   |
| <div style="border: 1px solid #ccc; padding: 2px; margin: 5px 0;"> <span style="font-size: 0.8em;">i</span> The testing of modbus telegrams can take several seconds due to ETS structure.         </div> |                                    |  |
|   |                                    | <input type="button" value="Send modbus request"/>                         |
|   | Modbus request (sent)              | 01 03 00 02 00 01h   |
|   |                                    | <input type="button" value="Check modbus response"/>                       |
|   | Modbus response (received)         | 01 03 02 00 2Dh  |
|   | Modbus response                    |  |
|   | Slave address                      | 1  |
|   | Function                           | Read holding registers - 03  |
|   | Address                            | 2  |
|   | Value                              | 45   |
|   | Date / Time                        | 2024-06-25 / 14:26:19  |

In the Modbus test environment, Modbus telegrams can be created and sent for test purposes. The range of functions corresponds to the settings that can also be used for the individual data points. The names of the parameters are identical.



*All settings only apply in the test environment and are not loaded into the device.*

### Baudrate

Configures the baud rate of the Modbus communication. The following are available:

- 1200 bits/s
- 2400 bits/s
- 4800 bits/s
- 9600 bits/s
- 19200 bits/s
- 38400 bits/s
- 56000 bits/s
- 57600 bits/s
- 115200 bits/s

## Parity

The Modbus frame is parameterized here with regard to parity and stop bit. The following options are available:

- Even (1 stop bit)
- Odd (1 stop bit)
- Keine (2 stop bits)
- Keine (1 stop bit)

## Slave address

The slave address (1 ... 247) of the Modbus request is set here.

## Function

The Modbus function code of the Modbus request is parameterized here. The following options are available:

- Read coils – 01
- Read discrete inputs – 02
- Read holding registers – 03
- Read input registers – 04
- Write single coil – 05
- Write multi coils – 15
- Write single holding register – 06
- Write multi holding registers – 16

Depending on this parameter, various functions are possible here, which are described in more detail in the following sections.

## Address

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

## Byte order (data in word register)

Defines the order of the two bytes per word register:

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

## Modbus request (preview)

The Modbus telegram created is displayed here.

## Send Modbus request

### Check Modbus response

The button “Send modbus request” sends the configured request and automatically waits for the response. If no response is received within 1 second, the button “Check modbus response” can be used to manually check for a response again. However, this is not necessary in most cases.

## Modbus request (sent)

The Modbus telegram sent is displayed here.

## Modbus response (received)

The received Modbus telegram is displayed here.

## Modbus response

In the section “Modbus response”, the data of the modbus response is listed and interpreted according to the settings in the section “Modbus request”

### 6.6.1 Function – Read coils – 01 Function – Read discrete inputs – 02

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Modbus settings > Modbus test environment

|  |                            |                            |                       |
|--|----------------------------|----------------------------|-----------------------|
| Description  | Modbus test environment    |                            |                       |
| General settings   | Baudrate                   | 19200 bits/s               |                       |
| Modbus settings  | Parity                     | Even (1 stop bit)          |                       |
| <b>Modbus test environment</b>   |                            |                            |                       |
| +  | Datapoints                 |                            |                       |
| +  | DPT converter              |                            |                       |
|  | Modbus request             | Slave address              | 1                     |
|  |                            | Function                   | Read coils - 01       |
|  |                            | Address                    | 3                     |
|  |                            | Count                      | 3                     |
|  | Modbus request (preview)   | 01 01 00 03 00 03h         |                       |
| <div style="border: 1px solid #007bff; padding: 5px; display: inline-block; margin: 5px 0;"> <span style="color: #007bff; font-weight: bold;">i</span> The testing of modbus telegrams can take several seconds due to ETS structure.         </div> |                            |                            |                       |
|  |                            | Send modbus request        |                       |
|  | Modbus request (sent)      | 01 01 00 03 00 03h         |                       |
|  |                            | Check modbus response      |                       |
|  | Modbus response (received) | 01 01 01 03h               |                       |
|  | Modbus response            | Slave address              | 1                     |
|  |                            | Function                   | Read coils - 01       |
|  |                            | Address                    | 3                     |
|  |                            | Bit register @ address     | Value '1'             |
|  |                            | Bit register @ address + 1 | Value '1'             |
|  |                            | Bit register @ address + 2 | Value '0'             |
|  |                            | Date / Time                | 2024-06-25 / 14:28:06 |

## Count

The number of bit registers to be read (1 ... 16) is set here.

## 6.6.2 Function – Read holding registers – 03 Function – Read input registers – 04

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Modbus settings > Modbus test environment

|   |                                    |  |
|---|------------------------------------|--|
| Description   | Modbus test environment            |  |
| General settings  | Baudrate                           | 19200 bits/s   |
| Modbus settings   | Parity                             | Even (1 stop bit)  |
| Modbus test environment   | Modbus request                     |  |
| + Datapoints  | Slave address                      | 1  |
| + DPT converter   | Function                           | Read holding registers - 03  |
|   | Address                            | 3  |
|   | Type                               | Word register  |
|   | Position in word register          | High/Low byte - unsigned   |
|   | Byte order (data in word register) | <input checked="" type="radio"/> MSB first <input type="radio"/> LSB first |
|   | Modbus request (preview)           | 01 03 00 03 00 01h   |
| <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p><span style="color: #007bff;">i</span> The testing of modbus telegrams can take several seconds due to ETS structure.</p> </div> |                                    |  |
|   |                                    | Send modbus request  |
|   | Modbus request (sent)              | 01 03 00 03 00 01h   |
|   |                                    | Check modbus response  |
|   | Modbus response (received)         | 01 03 02 00 15h  |
|   | Modbus response                    |  |
|   | Slave address                      | 1  |
|   | Function                           | Read holding registers - 03  |
|   | Address                            | 3  |
|   | Value                              | 21   |
|   | Date / Time                        | 2024-06-25 / 14:29:44  |

### Type

The following types can be configured:

- Word register  
1 word register (Modbus) is read
- Double word register  
2 word registers (Modbus) are read
- Four word register  
4 word registers (Modbus) are read

## **Position in word register** *(only for word register)*

This parameter defines the area of the word register that is read. The following areas 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

## **Word order** *(only for double/four word register)*

This parameter defines the word order in which the value is set in the word registers (Modbus).

The following options are available:

- Hi word @ address / Lo word @ address + 1 *(for double word register)*
- Lo word @ address / Hi word @ address + 1 *(for double word register)*
- Hi word @ address / Lo word @ address + 3 *(for four word register)*
- Lo word @ address / Hi word @ address + 3 *(for four word register)*

## **Type register value** *(only for double/four word register)*

This defines how the register value is interpreted. The following options are available:

- Modbus holds integer value – unsigned
- Modbus holds integer value – 2th complement
- Modbus holds float value – IEEE *(only for four word register)*

## 6.6.3 Function – Write single coil – 05 Function – Write multi coils – 15

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Modbus settings > Modbus test environment

|  |                            |  |
|--|----------------------------|--|
| Description  | Modbus test environment    |  |
| General settings   | Baudrate                   | 19200 bits/s   |
| Modbus settings  | Parity                     | Even (1 stop bit)  |
| <b>Modbus test environment</b>   | Modbus request             |  |
| + Datapoints   | Slave address              | 1  |
| + DPT converter  | Function                   | Write multi coils - 15   |
|  | Address                    | 3  |
|  | Count                      | 1  |
|  | Bit register               | <input checked="" type="radio"/> Value '1' <input type="radio"/> Value '0' |
|  | Modbus request (preview)   | 01 0F 00 03 00 01 01 01h   |
| <div style="border: 1px solid #add8e6; padding: 5px; display: inline-block;"> <i>i</i> The testing of modbus telegrams can take several seconds due to ETS structure.         </div> |                            |  |
|  |                            | <input type="button" value="Send modbus request"/>                         |
|  | Modbus request (sent)      | 01 0F 00 03 00 01 01 01h   |
|  |                            | <input type="button" value="Check modbus response"/>                       |
|  | Modbus response (received) | 01 0F 00 03 00 01h   |
|  | Modbus response            |  |
|  | Slave address              | 1  |
|  | Function                   | Write multi coils - 15   |
|  | Address                    | 3  |
|  | Value                      | Value '1'  |
|  | Date / Time                | 2024-06-25 / 14:30:37  |

### Count

Only one bit register can be written per request.



## 6.6.4 Function – Write single holding register – 06 Function – Write multi holding registers – 16

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Modbus settings > Modbus test environment

|                                |   |  |
|--------------------------------|---|--|
| Description                    | Modbus test environment   |  |
| General settings               | Baudrate  | 19200 bits/s   |
| – Modbus settings              | Parity  | Even (1 stop bit)  |
| <b>Modbus test environment</b> | Modbus request  |  |
| + Datapoints                   | Slave address   | 1  |
| + DPT converter                | Function  | Write multi holding registers - 16   |
|                                | Address   | 3  |
|                                | Type  | Word register  |
|                                | Position in word register   | High/Low byte - unsigned   |
|                                | Value   | 21   |
|                                | Byte order (data in word register)  | <input checked="" type="radio"/> MSB first <input type="radio"/> LSB first |
|                                | Modbus request (preview)  | 01 10 00 03 00 01 02 00 15h  |
|                                | <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p><span style="color: #007bff;">i</span> The testing of modbus telegrams can take several seconds due to ETS structure.</p> </div> |  |
|                                |   | <input type="button" value="Send modbus request"/>                         |
|                                | Modbus request (sent)   | 01 10 00 03 00 01 02 00 15h  |
|                                |   | <input type="button" value="Check modbus response"/>                       |
|                                | Modbus response (received)  | 01 10 00 03 00 01h   |
|                                | Modbus response   |  |
|                                | Slave address   | 1  |
|                                | Function  | Write multi holding registers - 16   |
|                                | Address   | 3  |
|                                | Value   | 21   |
|                                | Date / Time   | 2024-06-25 / 14:31:59  |

### Type

The following types can be configured:

- Word register  
1 word register (Modbus) is written
- Double word register (*only for write multi holding registers – 16*)  
2 word registers (Modbus) are written
- Four word register (*only for write multi holding registers – 16*)  
4 word registers (Modbus) are written

## **Position in word register** *(only for word register)*

This parameter defines the area of the word register that is written. The following areas 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

## **Word order** *(only for double/four word register)*

This parameter defines the word order in which the value is set in the word registers (Modbus). The following options are available:

- Hi word @ address / Lo word @ address + 1 *(for double word register)*
- Lo word @ address / Hi word @ address + 1 *(for double word register)*
- Hi word @ address / Lo word @ address + 3 *(for four word register)*
- Lo word @ address / Hi word @ address + 3 *(for four word register)*

## **Type register value** *(only for double/four word register)*

This defines how the register value is interpreted. The following options are available:

- Modbus holds integer value – unsigned
- Modbus holds integer value – 2th complement
- Modbus holds float value – IEEE *(only for four word register)*

## **Value**

The register value to be written.

## 6.7 Datapoints N – M

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints

|                  |  |  |
|------------------|--|--|
| Description      | Datapoints 1 - 10  |  |
| General settings | Description for this page <input style="width: 100%;" type="text"/>  |  |
| Modbus settings  | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |  |
| - Datapoints     | Slave address (common) 1   |  |

|   |   |  |
|---|---|--|
| <ul style="list-style-type: none"> <li style="background-color: #e0e0e0; padding: 2px;">Datapoints 1 - 10</li> <li style="padding: 2px;">Datapoints 11 - 20</li> <li style="padding: 2px;">Datapoints 21 - 30</li> <li style="padding: 2px;">Datapoints 31 - 40</li> <li style="padding: 2px;">Datapoints 41 - 50</li> <li style="padding: 2px;">Datapoints 51 - 60</li> <li style="padding: 2px;">Datapoints 61 - 70</li> <li style="padding: 2px;">Datapoints 71 - 80</li> <li style="padding: 2px;">Datapoints 81 - 90</li> <li style="padding: 2px;">Datapoints 91 - 100</li> <li style="padding: 2px;">Datapoints 101 - 110</li> <li style="padding: 2px;">Datapoints 111 - 120</li> <li style="padding: 2px;">Datapoints 121 - 130</li> <li style="padding: 2px;">Datapoints 131 - 140</li> <li style="padding: 2px;">Datapoints 141 - 150</li> <li style="padding: 2px;">Datapoints 151 - 160</li> <li style="padding: 2px;">Datapoints 161 - 170</li> <li style="padding: 2px;">Datapoints 171 - 180</li> <li style="padding: 2px;">Datapoints 181 - 190</li> <li style="padding: 2px;">Datapoints 191 - 200</li> <li style="padding: 2px;">Datapoints 201 - 210</li> <li style="padding: 2px;">Datapoints 211 - 220</li> <li style="padding: 2px;">Datapoints 221 - 230</li> <li style="padding: 2px;">Datapoints 231 - 240</li> <li style="padding: 2px;">Datapoints 241 - 250</li> </ul> | <p>Channel 1</p> <p>Datapoint type <span style="border: 1px solid #ccc; padding: 2px;">Disabled</span></p> <hr/> <p>Channel 2</p> <p>Datapoint type <span style="border: 1px solid #ccc; padding: 2px;">DPT 01 - Binary - 1 bit</span></p> <hr/> <p>Channel 3</p> <p>Datapoint type <span style="border: 1px solid #ccc; padding: 2px;">DPT 07 - Configured - 2 bytes</span></p> <hr/> <p>Channel 4</p> <p>Datapoint type <span style="border: 1px solid #ccc; padding: 2px;">DPT 09 - Float - 2 bytes</span></p> <hr/> <p>Channel 5</p> <p>Datapoint type <span style="border: 1px solid #ccc; padding: 2px;">DPT 12 - Unsigned - 4 bytes</span></p> <hr/> <p>Channel 6</p> <p>Datapoint type <span style="border: 1px solid #ccc; padding: 2px;">Disabled</span></p> <hr/> <p>Channel 7</p> <p>Datapoint type <span style="border: 1px solid #ccc; padding: 2px;">Disabled</span></p> <hr/> <p>Channel 8</p> <p>Datapoint type <span style="border: 1px solid #ccc; padding: 2px;">Disabled</span></p> <hr/> <p>Channel 9</p> <p>Datapoint type <span style="border: 1px solid #ccc; padding: 2px;">Disabled</span></p> <hr/> <p>Channel 10</p> <p>Datapoint type <span style="border: 1px solid #ccc; padding: 2px;">Disabled</span></p> | <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">Disabled</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">Disabled <span style="float: right;">✓</span></div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">DPT 01 - Binary - 1 bit</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">DPT 03 - Dimming - 4 bits</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">DPT 05 - Percent - 1 byte</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">DPT 05 - Configured - 1 byte</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">DPT 05 - Unsigned - 1 byte</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">DPT 06 - Signed - 1 byte</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">DPT 07 - Configured - 2 bytes</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">DPT 07 - Unsigned - 2 bytes</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">DPT 08 - Signed - 2 bytes</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">DPT 09 - Float - 2 bytes</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">DPT 12 - Configured - 4 bytes</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">DPT 12 - Unsigned - 4 bytes</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">DPT 13 - Signed - 4 bytes</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">DPT 14 - Float - 4 bytes</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">DPT 29 - Signed - 8 bytes</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">Fixed value - 1 bit</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">Fixed value - 2 bytes</div> <div style="border: 1px solid #ccc; padding: 2px; margin-bottom: 5px;">Disabled</div> |
|---|---|--|

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 address used (only in master mode)

This parameter determines whether the general slave address, which is configured in "Modbus settings", or an individual slave address is to be used for the configured channels on this page.

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

Here the individual slave address (0 ... 247) of the channels of this page is set.

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

If the diagnostic is activated, the diagnostic object of this page can be named here.

| Group object   | Type KNX | Size  | Direction |
|--|----------|-------|-----------|
| GO 451 Diagnostic: Slave (page 1) – No communication | 1.001    | 1 bit | To KNX    |

## 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 RTU 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.

## **Actual register value**

This button can be used to check the current modbus register value. The configuration must be loaded into the device to do this.



## 6.8 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)

### 6.8.1 Type – Bit register

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints

|                    |  |  |
|--------------------|--|--|
| Description        | Datapoints 1 - 10  |  |
| General settings   | Description for this page <input style="width: 100%;" type="text"/>  |  |
| Modbus settings    | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |  |
| - Datapoints       | Slave address (common) 1   |  |
| Datapoints 1 - 10  | Channel 1  |  |
| Datapoints 11 - 20 | Datapoint type <span style="float: right;">▼</span> DPT 01 - Binary - 1 bit                                    |  |
| Datapoints 21 - 30 | Description <input style="width: 100%;" type="text"/>  |  |
| Datapoints 31 - 40 | Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX                   |  |
| Datapoints 41 - 50 | Type <span style="float: right;">▼</span> Bit register   |  |
| Datapoints 51 - 60 | Value inverted <input checked="" type="radio"/> No <input type="radio"/> Yes                                   |  |
| Datapoints 61 - 70 | Function Write single coil - 05  |  |
| Datapoints 71 - 80 | Address <input style="width: 100%;" type="text" value="0"/>  |  |
| Datapoints 81 - 90 | Actual register value  |  |

### Value inverted

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

## 6.8.2 Type – Bit in word register

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints

|                          |  |  |
|--------------------------|--|--|
| Description              | <b>Datapoints 1 - 10</b>   |  |
| General settings         | Description for this page <input type="text"/>   |  |
| Modbus settings          | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |  |
| - Datapoints             | Slave address (common) 1   |  |
| <b>Datapoints 1 - 10</b> | Channel 1  |  |
| Datapoints 11 - 20       | Datapoint type <span style="float: right;">▼</span> DPT 01 - Binary - 1 bit                                    |  |
| Datapoints 21 - 30       | Description <input type="text"/>   |  |
| Datapoints 31 - 40       | Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX                   |  |
| Datapoints 41 - 50       | Type <span style="float: right;">▼</span> Bit in word register   |  |
| Datapoints 51 - 60       | Position in word register <span style="float: right;">▼</span> Bit 00  |  |
| Datapoints 61 - 70       | Valid bit in word register -----Vb   |  |
| Datapoints 71 - 80       | Value inverted <input checked="" type="radio"/> No <input type="radio"/> Yes                                   |  |
| Datapoints 81 - 90       | Function Write single holding register - 06  |  |
| Datapoints 91 - 100      | Address <input style="width: 100px;" type="text" value="0"/> <span style="float: right;">▲▼</span>             |  |
| Datapoints 101 - 110     | <input type="text" value="Actual register value"/>   |  |

### 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.

## 6.8.3 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).

### 6.8.3.1 Direction – KNX to modbus

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints

|                      |  |   |
|----------------------|--|---|
| Description          | Datapoints 1 - 10  |   |
| General settings     | Description for this page <input type="text"/>   |   |
| Modbus settings      | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |   |
| - Datapoints         | Slave address (common) 1   |   |
| Datapoints 1 - 10    | Channel 1  |   |
| Datapoints 11 - 20   | Datapoint type   | DPT 01 - Binary - 1 bit   |
| Datapoints 21 - 30   | Description  | <input type="text"/>  |
| Datapoints 31 - 40   | Direction  | <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX    |
| 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   | Behavior on receiving data 'ON'  | <input type="radio"/> No reaction <input checked="" type="radio"/> Set register value |
| Datapoints 81 - 90   | Register value   | 1   |
| Datapoints 91 - 100  | Register value (binary)  | 0000 0000 0000 0001b  |
| Datapoints 101 - 110 | Register value (hexadecimal)   | 0001h   |
| Datapoints 111 - 120 | Behavior on receiving data 'OFF'   | <input type="radio"/> No reaction <input checked="" type="radio"/> Set register value |
| Datapoints 121 - 130 | Register value   | 0   |
| Datapoints 131 - 140 | Register value (binary)  | 0000 0000 0000 0000b  |
| Datapoints 141 - 150 | Register value (hexadecimal)   | 0000h   |
| Datapoints 151 - 160 | Function   | Write single holding register - 06  |
| Datapoints 161 - 170 | Address  | 0   |
| Datapoints 171 - 180 | Actual register value  |   |

### 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**.

### 6.8.3.2 Direction – Modbus to KNX

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints |   |
|--|---|
| Description  | Datapoints 1 - 10   |
| General settings                                       | Description for this page <input type="text"/>  |
| Modbus settings  | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page        |
| - Datapoints   | Slave address (common) 1  |
| Datapoints 1 - 10                                      | Channel 1   |
| Datapoints 11 - 20                                     | Datapoint type DPT 01 - Binary - 1 bit  |
| Datapoints 21 - 30                                     | Description <input type="text"/>  |
| Datapoints 31 - 40                                     | Direction <input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX                          |
| Datapoints 41 - 50                                     | Send condition On change  |
| Datapoints 51 - 60                                     | Type Value in word register   |
| Datapoints 61 - 70                                     | Bit count 16 bit  |
| Datapoints 71 - 80                                     | Offset from right 00 bit  |
| Datapoints 81 - 90                                     | Threshold 1   |
| Datapoints 91 - 100                                    | Threshold (binary) 0000 0000 0000 0001b   |
| Datapoints 101 - 110                                   | Threshold (hexadecimal) 0001h   |
| Datapoints 111 - 120                                   | Behavior on value higher Send 'OFF'   |
| Datapoints 121 - 130                                   | Behavior on value match Send 'ON'   |
| Datapoints 131 - 140                                   | Behavior on value lower Send 'OFF'  |
| Datapoints 141 - 150                                   | Function <input checked="" type="radio"/> Read holding registers - 03 <input type="radio"/> Read input registers - 04 |
| Datapoints 151 - 160                                   | Address 0   |
| Datapoints 161 - 170                                   | Polling interval Every cycle  |
| Datapoints 171 - 180                                   | Actual register value <input type="text"/>  |
| Datapoints 181 - 190                                   |   |

### 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'

## 6.9 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)

## 6.9.1 Type – Bit register

### 6.9.1.1 Direction – KNX to modbus

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints |  |
|--|--|
| Description  | Datapoints 1 - 10  |
| General settings                                       | Description for this page <input type="text"/>   |
| Modbus settings  | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |
| – Datapoints   | Slave address (common) 1   |
| <b>Datapoints 1 - 10</b>                               | Channel 1  |
| Datapoints 11 - 20                                     | Datapoint type DPT 03 - Dimming - 4 bits   |
| Datapoints 21 - 30                                     | Description <input type="text"/>   |
| Datapoints 31 - 40                                     | Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX                   |
| Datapoints 41 - 50                                     | Type Bit register  |
| Datapoints 51 - 60                                     | Behavior on receiving data 'DIM UP' Bit in register - Value '1'  |
| Datapoints 61 - 70                                     | Behavior on receiving data 'DIM DOWN' No reaction  |
| Datapoints 71 - 80                                     | Behavior on receiving data 'DIM STOP' Bit in register - Value '0'  |
| Datapoints 81 - 90                                     | Function Write single coil - 05  |
| Datapoints 91 - 100                                    | Address <input type="text" value="0"/>   |
| Datapoints 101 - 110                                   | Actual register value <input type="text"/>   |
| Datapoints 111 - 120                                   |  |

#### 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'

### 6.9.1.2 Direction – Modbus to KNX

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints |  |
|--|--|
| Description  | Datapoints 1 - 10  |
| General settings                                       | Description for this page <input type="text"/>   |
| Modbus settings  | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |
| - Datapoints   | Slave address (common) 1   |
| <b>Datapoints 1 - 10</b>                               | Channel 1  |
| Datapoints 11 - 20                                     | Datapoint type <b>DPT 03 - Dimming - 4 bits</b>  |
| Datapoints 21 - 30                                     | Description <input type="text"/>   |
| Datapoints 31 - 40                                     | Direction <input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX                   |
| Datapoints 41 - 50                                     | Send condition <b>On change</b>  |
| Datapoints 51 - 60                                     | Type <b>Bit register</b>   |
| Datapoints 61 - 70                                     | Behavior on bit in register - Value '1' <b>Send 'DIM UP'</b>   |
| Datapoints 71 - 80                                     | Behavior on bit in register - Value '0' <b>Send 'DIM STOP'</b>   |
| Datapoints 81 - 90                                     | Function <input checked="" type="radio"/> Read coils - 01 <input type="radio"/> Read discrete inputs - 02      |
| Datapoints 91 - 100                                    | Address <b>0</b>   |
| Datapoints 101 - 110                                   | Polling interval <b>Every cycle</b>  |
| Datapoints 111 - 120                                   | <input type="text" value="Actual register value"/>   |
| Datapoints 121 - 130                                   |  |

#### 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'

## 6.9.2 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.

### 6.9.2.1 Direction – KNX to modbus

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints |  |
|--|--|
| Description  | Datapoints 1 - 10  |
| General settings                                       | Description for this page <input type="text"/>   |
| Modbus settings  | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |
| - Datapoints   | Slave address (common) 1   |
| Datapoints 1 - 10                                      | Channel 1  |
| Datapoints 11 - 20                                     | Datapoint type DPT 03 - Dimming - 4 bits   |
| Datapoints 21 - 30                                     | Description <input type="text"/>   |
| Datapoints 31 - 40                                     | Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX                   |
| 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 receiving data 'DIM UP' Bit in register - Value '1'  |
| Datapoints 81 - 90                                     | Behavior on receiving data 'DIM DOWN' No reaction  |
| Datapoints 91 - 100                                    | Behavior on receiving data 'DIM STOP' Bit in register - Value '0'  |
| Datapoints 101 - 110                                   | Function Write single holding register - 06  |
| Datapoints 111 - 120                                   | Address 0  |
| Datapoints 121 - 130                                   | Actual register value  |

### 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'

### 6.9.2.2 Direction – Modbus to KNX

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints

|                          |  |   |
|--------------------------|--|---|
| Description              | Datapoints 1 - 10  |   |
| General settings         | Description for this page <input type="text"/>   |   |
| Modbus settings          | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |   |
| – Datapoints             | Slave address (common) 1   |   |
| <b>Datapoints 1 - 10</b> | Channel 1  |   |
| Datapoints 11 - 20       | Datapoint type   | DPT 03 - Dimming - 4 bits   |
| Datapoints 21 - 30       | Description  | <input type="text"/>  |
| Datapoints 31 - 40       | Direction  | <input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX                              |
| Datapoints 41 - 50       | Send condition   | On change   |
| Datapoints 51 - 60       | Type   | Bit in word register  |
| Datapoints 61 - 70       | Position in word register  | Bit 00  |
| Datapoints 71 - 80       | Valid bit in word register   | ---- ---- ---- ---Vb  |
| Datapoints 81 - 90       | Behavior on bit in register - Value '1'  | Send 'DIM UP'   |
| Datapoints 91 - 100      | Behavior on bit in register - Value '0'  | Send 'DIM STOP'   |
| Datapoints 101 - 110     | Function   | <input checked="" type="radio"/> Read holding registers - 03<br><input type="radio"/> Read input registers - 04 |
| Datapoints 111 - 120     | Address  | 0   |
| Datapoints 121 - 130     | Polling interval   | Every cycle   |
| Datapoints 131 - 140     | Actual register value  |   |
| Datapoints 141 - 150     |  |   |

## 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’

## 6.9.3 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).

### 6.9.3.1 Direction – KNX to modbus

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints

|                          |   |  |
|--------------------------|---|--|
| Description              | Datapoints 1 - 10   |  |
| General settings         | Description for this page <input type="text"/>  |  |
| Modbus settings          | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page              |  |
| - Datapoints             | Slave address (common) 1  |  |
| <b>Datapoints 1 - 10</b> | Channel 1   |  |
| Datapoints 11 - 20       | Datapoint type <input type="text" value="DPT 03 - Dimming - 4 bits"/>   |  |
| Datapoints 21 - 30       | Description <input type="text"/>  |  |
| Datapoints 31 - 40       | Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX                                |  |
| Datapoints 41 - 50       | Type <input type="text" value="Value in word register"/>  |  |
| Datapoints 51 - 60       | Bit count <input type="text" value="16 bit"/>   |  |
| Datapoints 61 - 70       | Offset from right <input type="text" value="00 bit"/>   |  |
| Datapoints 71 - 80       | Behavior on receiving data 'DIM UP' <input type="radio"/> No reaction <input checked="" type="radio"/> Set register value   |  |
| Datapoints 81 - 90       | Register value <input type="text" value="1"/>   |  |
| Datapoints 91 - 100      | Register value (binary) 0000 0000 0000 0001b  |  |
| Datapoints 101 - 110     | Register value (hexadecimal) 0001h  |  |
| Datapoints 111 - 120     | Behavior on receiving data 'DIM DOWN' <input checked="" type="radio"/> No reaction <input type="radio"/> Set register value |  |
| Datapoints 121 - 130     | Behavior on receiving data 'DIM STOP' <input type="radio"/> No reaction <input checked="" type="radio"/> Set register value |  |
| Datapoints 131 - 140     | Register value <input type="text" value="0"/>   |  |
| Datapoints 141 - 150     | Register value (binary) 0000 0000 0000 0000b  |  |
| Datapoints 151 - 160     | Register value (hexadecimal) 0000h  |  |
| Datapoints 161 - 170     | Function <input type="text" value="Write single holding register - 06"/>  |  |
| Datapoints 171 - 180     | Address <input type="text" value="0"/>  |  |
| Datapoints 181 - 190     | Actual register value <input type="text"/>  |  |

## **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**.

## 6.9.3.2 Direction – Modbus to KNX

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints

|                      |   |  |
|----------------------|---|--|
| Description          | Datapoints 1 - 10   |  |
| General settings     | Description for this page <input type="text"/>  |  |
| Modbus settings      | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page        |  |
| - Datapoints         | Slave address (common) 1  |  |
| Datapoints 1 - 10    | Channel 1   |  |
| Datapoints 11 - 20   | Datapoint type <span style="border: 1px solid #ccc; padding: 1px;">DPT 03 - Dimming - 4 bits</span>                   |  |
| Datapoints 21 - 30   | Description <input type="text"/>  |  |
| Datapoints 31 - 40   | Direction <input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX                          |  |
| Datapoints 41 - 50   | Send condition <span style="border: 1px solid #ccc; padding: 1px;">On change</span>                                   |  |
| Datapoints 51 - 60   | Type <span style="border: 1px solid #ccc; padding: 1px;">Value in word register</span>                                |  |
| Datapoints 61 - 70   | Bit count <span style="border: 1px solid #ccc; padding: 1px;">16 bit</span>   |  |
| Datapoints 71 - 80   | Offset from right <span style="border: 1px solid #ccc; padding: 1px;">00 bit</span>                                   |  |
| Datapoints 81 - 90   | Threshold <span style="border: 1px solid #ccc; padding: 1px;">1</span>  |  |
| Datapoints 91 - 100  | Threshold (binary) 0000 0000 0000 0001b   |  |
| Datapoints 101 - 110 | Threshold (hexadecimal) 0001h   |  |
| Datapoints 111 - 120 | Behavior on value higher <span style="border: 1px solid #ccc; padding: 1px;">Send 'DIM STOP'</span>                   |  |
| Datapoints 121 - 130 | Behavior on value match <span style="border: 1px solid #ccc; padding: 1px;">Send 'DIM UP'</span>                      |  |
| Datapoints 131 - 140 | Behavior on value lower <span style="border: 1px solid #ccc; padding: 1px;">Send 'DIM STOP'</span>                    |  |
| Datapoints 141 - 150 | Function <input checked="" type="radio"/> Read holding registers - 03 <input type="radio"/> Read input registers - 04 |  |
| Datapoints 151 - 160 | Address <span style="border: 1px solid #ccc; padding: 1px;">0</span>  |  |
| Datapoints 161 - 170 | Polling interval <span style="border: 1px solid #ccc; padding: 1px;">Every cycle</span>                               |  |
| Datapoints 171 - 180 | <span style="border: 1px solid #ccc; padding: 2px 10px;">Actual register value</span>                                 |  |
| Datapoints 181 - 190 |   |  |

### 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'

## 6.10 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)

### 6.10.1 Type – Word register

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints |  |
|--|--|
| Description  | Datapoints 1 - 10  |
| General settings                                       | Description for this page <input type="text"/>   |
| Modbus settings  | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |
| – Datapoints   | Slave address (common) 1   |
| <b>Datapoints 1 - 10</b>                               | Channel 1  |
| Datapoints 11 - 20                                     | Datapoint type DPT 05 - Percent - 1 byte   |
| Datapoints 21 - 30                                     | Description <input type="text"/>   |
| Datapoints 31 - 40                                     | Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX                   |
| Datapoints 41 - 50                                     | Type Word register   |
| Datapoints 51 - 60                                     | Position in word register Low byte   |
| Datapoints 61 - 70                                     | Value minimum (register) 0   |
| Datapoints 71 - 80                                     | Value maximum (register) 255   |
| 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  |
| Datapoints 121 - 130                                   | Actual register value <input type="text"/>   |

### 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.*

## 6.11 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)

## 6.11.1 Type – Bit register

### 6.11.1.1 Direction – KNX to modbus

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints

|                          |  |   |
|--------------------------|--|---|
| Description              | Datapoints 1 - 10                              |   |
| General settings         | Description for this page <input type="text"/> |   |
| Modbus settings          | Slave address used                             | <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |
| - Datapoints             | Slave address (common)                         | 1   |
| <b>Datapoints 1 - 10</b> | Channel 1                                      |   |
| Datapoints 11 - 20       | Datapoint type                                 | DPT 05 - Configured - 1 byte  |
| Datapoints 21 - 30       | Description                                    | <input type="text"/>  |
| Datapoints 31 - 40       | Direction                                      | <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX          |
| Datapoints 41 - 50       | Type   | Bit register  |
| Datapoints 51 - 60       | Threshold                                      | 255   |
| Datapoints 61 - 70       | Behavior on value higher                       | Bit in register - Value '0'   |
| Datapoints 71 - 80       | Behavior on value match                        | Bit in register - Value '1'   |
| Datapoints 81 - 90       | Behavior on value lower                        | Bit in register - Value '0'   |
| Datapoints 91 - 100      | Function                                       | Write single coil - 05  |
| Datapoints 101 - 110     | Address  | 0   |
| Datapoints 111 - 120     | Actual register value <input type="text"/>     |   |

#### 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'

### 6.11.1.2 Direction – Modbus to KNX

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints |   |
|--|---|
| Description  | Datapoints 1 - 10   |
| General settings                                       | Description for this page <input type="text"/>  |
| Modbus settings  | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page        |
| – Datapoints   | Slave address (common) 1  |
| <b>Datapoints 1 - 10</b>                               | Channel 1   |
| Datapoints 11 - 20                                     | Datapoint type DPT 05 - Configured - 1 byte   |
| Datapoints 21 - 30                                     | Description <input type="text"/>  |
| Datapoints 31 - 40                                     | Direction <input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX                          |
| Datapoints 41 - 50                                     | Send condition On change  |
| Datapoints 51 - 60                                     | Type Bit register   |
| Datapoints 61 - 70                                     | Behavior on bit in register - Value '1' <input type="radio"/> No reaction <input checked="" type="radio"/> Send value |
| Datapoints 71 - 80                                     | Object value 255  |
| Datapoints 81 - 90                                     | Behavior on bit in register - Value '0' <input type="radio"/> No reaction <input checked="" type="radio"/> Send value |
| Datapoints 91 - 100                                    | Object value 0  |
| Datapoints 101 - 110                                   | Function <input checked="" type="radio"/> Read coils - 01 <input type="radio"/> Read discrete inputs - 02             |
| Datapoints 111 - 120                                   | Address 0   |
| Datapoints 121 - 130                                   | Polling interval Every cycle  |
| Datapoints 131 - 140                                   | Actual register value <input type="text"/>  |
| Datapoints 141 - 150                                   |   |

## 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.

## 6.11.2 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.

### 6.11.2.1 Direction – KNX to modbus

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints

|                          |  |   |
|--------------------------|--|---|
| Description              | Datapoints 1 - 10                              |   |
| General settings         | Description for this page <input type="text"/> |   |
| Modbus settings          | Slave address used                             | <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |
| – Datapoints             | Slave address (common)                         | 1   |
| <b>Datapoints 1 - 10</b> | Channel 1                                      |   |
| Datapoints 11 - 20       | Datapoint type                                 | DPT 05 - Configured - 1 byte  |
| Datapoints 21 - 30       | Description                                    | <input type="text"/>  |
| Datapoints 31 - 40       | Direction                                      | <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX          |
| Datapoints 41 - 50       | Type   | Bit in word register  |
| Datapoints 51 - 60       | Threshold                                      | 255   |
| Datapoints 61 - 70       | Position in word register                      | Bit 00  |
| Datapoints 71 - 80       | Valid bit in word register                     | ---- -- -- -- --Vb  |
| Datapoints 81 - 90       | Behavior on value higher                       | Bit in register - Value '0'   |
| Datapoints 91 - 100      | Behavior on value match                        | Bit in register - Value '1'   |
| Datapoints 101 - 110     | Behavior on value lower                        | Bit in register - Value '0'   |
| Datapoints 111 - 120     | Function                                       | Write single holding register - 06  |
| Datapoints 121 - 130     | Address  | 0   |
| Datapoints 131 - 140     | Actual register value                          |   |

## 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'

### 6.11.2.2 Direction – Modbus to KNX

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints

|                          |   |  |
|--------------------------|---|--|
| Description              | <b>Datapoints 1 - 10</b>  |  |
| General settings         | Description for this page <input style="width: 90%;" type="text"/>  |  |
| Modbus settings          | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page        |  |
| - Datapoints             | Slave address (common) 1  |  |
| <b>Datapoints 1 - 10</b> | Channel 1   |  |
| Datapoints 11 - 20       | Datapoint type <input type="text" value="DPT 05 - Configured - 1 byte"/>  |  |
| Datapoints 21 - 30       | Description <input style="width: 90%;" type="text"/>  |  |
| Datapoints 31 - 40       | Direction <input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX                          |  |
| Datapoints 41 - 50       | Send condition <input type="text" value="On change"/>   |  |
| Datapoints 51 - 60       | Type <input type="text" value="Bit in word register"/>  |  |
| Datapoints 61 - 70       | Position in word register <input type="text" value="Bit 00"/>   |  |
| Datapoints 71 - 80       | Valid bit in word register -----Vb  |  |
| Datapoints 81 - 90       | Behavior on bit in register - Value '1' <input type="radio"/> No reaction <input checked="" type="radio"/> Send value |  |
| Datapoints 91 - 100      | Object value <input type="text" value="255"/>   |  |
| Datapoints 101 - 110     | Behavior on bit in register - Value '0' <input type="radio"/> No reaction <input checked="" type="radio"/> Send value |  |
| Datapoints 111 - 120     | Object value <input type="text" value="0"/>   |  |
| Datapoints 121 - 130     | Function <input checked="" type="radio"/> Read holding registers - 03 <input type="radio"/> Read input registers - 04 |  |
| Datapoints 131 - 140     | Address <input type="text" value="0"/>  |  |
| Datapoints 141 - 150     | Polling interval <input type="text" value="Every cycle"/>   |  |
| Datapoints 151 - 160     | <input type="text" value="Actual register value"/>  |  |
| Datapoints 161 - 170     |   |  |

## 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.

## **6.11.3 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).

## 6.11.3.1 Direction – KNX to modbus

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints

|                  |  |
|------------------|--|
| Description      | Datapoints 1 - 10  |
| General settings | Description for this page <input type="text"/>   |
| Modbus settings  | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |
| - Datapoints     | Slave address (common) 1   |

|   |   |
|---|---|
| <ul style="list-style-type: none"> <li style="background-color: #e0e0e0; padding: 2px 5px;">Datapoints 1 - 10</li> <li style="padding: 2px 5px;">Datapoints 11 - 20</li> <li style="padding: 2px 5px;">Datapoints 21 - 30</li> <li style="padding: 2px 5px;">Datapoints 31 - 40</li> <li style="padding: 2px 5px;">Datapoints 41 - 50</li> <li style="padding: 2px 5px;">Datapoints 51 - 60</li> <li style="padding: 2px 5px;">Datapoints 61 - 70</li> <li style="padding: 2px 5px;">Datapoints 71 - 80</li> <li style="padding: 2px 5px;">Datapoints 81 - 90</li> <li style="padding: 2px 5px;">Datapoints 91 - 100</li> <li style="padding: 2px 5px;">Datapoints 101 - 110</li> <li style="padding: 2px 5px;">Datapoints 111 - 120</li> <li style="padding: 2px 5px;">Datapoints 121 - 130</li> <li style="padding: 2px 5px;">Datapoints 131 - 140</li> <li style="padding: 2px 5px;">Datapoints 141 - 150</li> <li style="padding: 2px 5px;">Datapoints 151 - 160</li> <li style="padding: 2px 5px;">Datapoints 161 - 170</li> <li style="padding: 2px 5px;">Datapoints 171 - 180</li> <li style="padding: 2px 5px;">Datapoints 181 - 190</li> <li style="padding: 2px 5px;">Datapoints 191 - 200</li> <li style="padding: 2px 5px;">Datapoints 201 - 210</li> <li style="padding: 2px 5px;">Datapoints 211 - 220</li> <li style="padding: 2px 5px;">Datapoints 221 - 230</li> </ul> | <p>Channel 1</p> <p>Datapoint type <span style="float: right;">▼</span><br/>DPT 05 - Configured - 1 byte</p> <p>Description <input type="text"/></p> <p>Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX</p> <p>Type <span style="float: right;">▼</span><br/>Value in word register</p> <p>Threshold <input type="text" value="255"/></p> <p>Bit count <span style="float: right;">▼</span><br/>16 bit</p> <p>Offset from right <span style="float: right;">▼</span><br/>00 bit</p> <p>Behavior on value higher <input type="radio"/> No reaction <input checked="" type="radio"/> Set register value</p> <p>Register value <input type="text" value="0"/></p> <p>Register value (binary) 0000 0000 0000 0000b</p> <p>Register value (hexadecimal) 0000h</p> <p>Behavior on value match <input type="radio"/> No reaction <input checked="" type="radio"/> Set register value</p> <p>Register value <input type="text" value="1"/></p> <p>Register value (binary) 0000 0000 0000 0001b</p> <p>Register value (hexadecimal) 0001h</p> <p>Behavior on value lower <input type="radio"/> No reaction <input checked="" type="radio"/> Set register value</p> <p>Register value <input type="text" value="0"/></p> <p>Register value (binary) 0000 0000 0000 0000b</p> <p>Register value (hexadecimal) 0000h</p> <p>Function <span style="float: right;">▼</span><br/>Write single holding register - 06</p> <p>Address <input type="text" value="0"/></p> <p style="text-align: center; background-color: #e0e0e0; padding: 2px;">Actual register value</p> |
|---|---|

### 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**.

## 6.11.3.2 Direction – Modbus to KNX

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints

|                          |  |   |
|--------------------------|--|---|
| Description              | Datapoints 1 - 10  |   |
| General settings         | Description for this page <input type="text"/>   |   |
| Modbus settings          | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |   |
| - Datapoints             | Slave address (common) 1   |   |
| <b>Datapoints 1 - 10</b> | Channel 1  |   |
| Datapoints 11 - 20       | Datapoint type   | DPT 05 - Configured - 1 byte  |
| Datapoints 21 - 30       | Description  | <input type="text"/>  |
| Datapoints 31 - 40       | Direction  | <input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX                              |
| Datapoints 41 - 50       | Send condition   | On change   |
| Datapoints 51 - 60       | Type   | Value in word register  |
| Datapoints 61 - 70       | Bit count  | 16 bit  |
| Datapoints 71 - 80       | Offset from right  | 00 bit  |
| Datapoints 81 - 90       | Threshold  | 1   |
| Datapoints 91 - 100      | Threshold (binary)   | 0000 0000 0000 0001b  |
| Datapoints 101 - 110     | Threshold (hexadecimal)  | 0001h   |
| Datapoints 111 - 120     | Behavior on value higher   | <input type="radio"/> No reaction <input checked="" type="radio"/> Send value                                   |
| Datapoints 121 - 130     | Object value   | 0   |
| Datapoints 131 - 140     | Behavior on value match  | <input type="radio"/> No reaction <input checked="" type="radio"/> Send value                                   |
| Datapoints 141 - 150     | Object value   | 255   |
| Datapoints 151 - 160     | Behavior on value lower  | <input type="radio"/> No reaction <input checked="" type="radio"/> Send value                                   |
| Datapoints 161 - 170     | Object value   | 0   |
| Datapoints 171 - 180     | Function   | <input checked="" type="radio"/> Read holding registers - 03<br><input type="radio"/> Read input registers - 04 |
| Datapoints 181 - 190     | Address  | 0   |
| Datapoints 191 - 200     | Polling interval   | Every cycle   |
| Datapoints 201 - 210     | Actual register value  |   |
| Datapoints 211 - 220     |  |   |

### 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).

## 6.12 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)

## 6.12.1 Type – Word register

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10

|                          |  |   |
|--------------------------|--|---|
| Description              | Datapoints 1 - 10                              |   |
| General settings         | Description for this page <input type="text"/> |   |
| Modbus settings          | Slave address used                             | <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |
| – Datapoints             | Slave address (common)                         | 1   |
| <b>Datapoints 1 - 10</b> | Channel 1                                      |   |
| Datapoints 11 - 20       | Datapoint type                                 | DPT 05 - Unsigned - 1 byte  |
| Datapoints 21 - 30       | Description                                    | <input type="text"/>  |
| Datapoints 31 - 40       | Direction                                      | <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX          |
| Datapoints 41 - 50       | Type   | Word register   |
| Datapoints 51 - 60       | Position in word register                      | Configured  |
| Datapoints 61 - 70       | Bit count                                      | 08 bit  |
| Datapoints 71 - 80       | Offset from right                              | 00 bit  |
| Datapoints 81 - 90       | Valid bits in word register                    | ---- ---- VVVV VVVVb  |
| Datapoints 91 - 100      | Valid values                                   | 0 ... 255   |
| Datapoints 101 - 110     | Function                                       | Write single holding register - 06  |
| Datapoints 111 - 120     | Address  | <input type="text" value="0"/>  |
| Datapoints 121 - 130     | Actual register value                          | <input type="text"/>  |

### 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.



## 6.13 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)

### 6.13.1 Type – Word register

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10 |  |
|--|--|
| Description  | Datapoints 1 - 10  |
| General settings   | Description for this page <input type="text"/>   |
| Modbus settings  | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |
| - Datapoints   | Slave address (common) 1   |
| <b>Datapoints 1 - 10</b>   | Channel 1  |
| Datapoints 11 - 20   | Datapoint type <b>DPT 06 - Signed - 1 byte</b>   |
| Datapoints 21 - 30   | Description <input type="text"/>   |
| Datapoints 31 - 40   | Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX                   |
| Datapoints 41 - 50   | Type <b>Word register</b>  |
| Datapoints 51 - 60   | Position in word register <b>Configured</b>  |
| Datapoints 61 - 70   | Offset from right <b>00 bit</b>  |
| Datapoints 71 - 80   | Valid bits in word register ----- VVVV VVVVb   |
| Datapoints 81 - 90   | Valid values -128 ... 127  |
| Datapoints 91 - 100  | Function <b>Write single holding register - 06</b>   |
| Datapoints 101 - 110   | Address <input type="text" value="0"/>   |
| Datapoints 111 - 120   | <input type="button" value="Actual register value"/>   |

### 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.

## 6.14 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)

### 6.14.1 Type – Bit register

#### 6.14.1.1 Direction – KNX to modbus

The screenshot shows a configuration page for 'Datapoints 1 - 10'. The left sidebar lists various datapoint ranges, with 'Datapoints 1 - 10' selected. The main configuration area includes the following fields:

- Description:** Datapoints 1 - 10
- General settings:** Description for this page (empty text box)
- Modbus settings:** Slave address used (radio buttons for 'from "Modbus settings"' and 'for this page')
- Datapoints:** Slave address (common) set to 1
- Datapoints 1 - 10 (selected):**
  - Channel 1
  - Datapoint type: DPT 07 - Configured - 2 bytes
  - Description: (empty text box)
  - 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
  - Actual register value: (button)

### 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’

### 6.14.1.2 Direction – Modbus to KNX

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10

|                      |   |  |
|----------------------|---|--|
| Description          | Datapoints 1 - 10   |  |
| General settings     | Description for this page <input style="width: 90%;" type="text"/>  |  |
| Modbus settings      | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page        |  |
| - Datapoints         | Slave address (common) 1  |  |
| Datapoints 1 - 10    | Channel 1   |  |
| Datapoints 11 - 20   | Datapoint type <span style="float: right;">▼</span> DPT 07 - Configured - 2 bytes                                     |  |
| Datapoints 21 - 30   | Description <input style="width: 90%;" type="text"/>  |  |
| Datapoints 31 - 40   | Direction <input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX                          |  |
| Datapoints 41 - 50   | Send condition <span style="float: right;">▼</span> On change   |  |
| Datapoints 51 - 60   | Type <span style="float: right;">▼</span> Bit register  |  |
| Datapoints 61 - 70   | Behavior on bit in register - Value '1' <input type="radio"/> No reaction <input checked="" type="radio"/> Send value |  |
| Datapoints 71 - 80   | Object value <span style="float: right;">▲▼</span> 65535  |  |
| Datapoints 81 - 90   | Behavior on bit in register - Value '0' <input type="radio"/> No reaction <input checked="" type="radio"/> Send value |  |
| Datapoints 91 - 100  | Object value <span style="float: right;">▲▼</span> 0  |  |
| Datapoints 101 - 110 | Function <input checked="" type="radio"/> Read coils - 01 <input type="radio"/> Read discrete inputs - 02             |  |
| Datapoints 111 - 120 | Address <span style="float: right;">▲▼</span> 0   |  |
| Datapoints 121 - 130 | Polling interval <span style="float: right;">▼</span> Every cycle   |  |
| Datapoints 131 - 140 | Actual register value <input style="width: 100%;" type="text"/>   |  |
| Datapoints 141 - 150 |   |  |

## 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.

## 6.14.2 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.

### 6.14.2.1 Direction – KNX to modbus

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10 |  |
|--|--|
| Description  | Datapoints 1 - 10  |
| General settings   | Description for this page <input type="text"/>   |
| Modbus settings  | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |
| - Datapoints   | Slave address (common) 1   |
| <b>Datapoints 1 - 10</b>   | Channel 1  |
| Datapoints 11 - 20   | Datapoint type DPT 07 - Configured - 2 bytes   |
| Datapoints 21 - 30   | Description <input type="text"/>   |
| Datapoints 31 - 40   | Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX                   |
| Datapoints 41 - 50   | Type Bit in word register  |
| Datapoints 51 - 60   | Threshold 65535  |
| Datapoints 61 - 70   | Position in word register Bit 00   |
| Datapoints 71 - 80   | Valid bit in word register ---- - - - - - - - - - - Vb   |
| Datapoints 81 - 90   | Behavior on value higher Bit in register - Value '0'   |
| Datapoints 91 - 100  | Behavior on value match Bit in register - Value '1'  |
| Datapoints 101 - 110   | Behavior on value lower Bit in register - Value '0'  |
| Datapoints 111 - 120   | Function Write single holding register - 06  |
| Datapoints 121 - 130   | Address 0  |
| Datapoints 131 - 140   | Actual register value <input type="text"/>   |

## 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'

## 6.14.2.2 Direction – Modbus to KNX

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10

|                          |  |   |
|--------------------------|--|---|
| Description              | Datapoints 1 - 10  |   |
| General settings         | Description for this page  |   |
| Modbus settings          | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |   |
| - Datapoints             | Slave address (common) 1   |   |
| <b>Datapoints 1 - 10</b> | Channel 1  |   |
| Datapoints 11 - 20       | Datapoint type   | DPT 07 - Configured - 2 bytes   |
| Datapoints 21 - 30       | Description  |   |
| Datapoints 31 - 40       | Direction  | <input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX                              |
| Datapoints 41 - 50       | Send condition   | On change   |
| Datapoints 51 - 60       | Type   | Bit in word register  |
| Datapoints 61 - 70       | Position in word register  | Bit 00  |
| Datapoints 71 - 80       | Valid bit in word register   | ---- -- -- -- --Vb  |
| Datapoints 81 - 90       | Behavior on bit in register - Value '1'  | <input type="radio"/> No reaction <input checked="" type="radio"/> Send value                                   |
| Datapoints 91 - 100      | Object value   | 65535   |
| Datapoints 101 - 110     | Behavior on bit in register - Value '0'  | <input type="radio"/> No reaction <input checked="" type="radio"/> Send value                                   |
| Datapoints 111 - 120     | Object value   | 0   |
| Datapoints 121 - 130     | Function   | <input checked="" type="radio"/> Read holding registers - 03<br><input type="radio"/> Read input registers - 04 |
| Datapoints 131 - 140     | Address  | 0   |
| Datapoints 141 - 150     | Polling interval   | Every cycle   |
| Datapoints 151 - 160     | Actual register value  |   |
| Datapoints 161 - 170     |  |   |

### 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.

## 6.14.3 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).

### 6.14.3.1 Direction – KNX to modbus

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10

|                      |  |  |
|----------------------|--|--|
| Description          | Datapoints 1 - 10  |  |
| General settings     | Description for this page <input type="text"/>   |  |
| Modbus settings      | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |  |
| - Datapoints         | Slave address (common) 1   |  |
| Datapoints 1 - 10    | Channel 1  |  |
| Datapoints 11 - 20   | Datapoint type DPT 07 - Configured - 2 bytes   |  |
| Datapoints 21 - 30   | Description <input type="text"/>   |  |
| Datapoints 31 - 40   | Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX                   |  |
| Datapoints 41 - 50   | Type Value in word register  |  |
| Datapoints 51 - 60   | Threshold 65535  |  |
| Datapoints 61 - 70   | Bit count 16 bit   |  |
| Datapoints 71 - 80   | Offset from right 00 bit   |  |
| Datapoints 81 - 90   | Behavior on value higher <input type="radio"/> No reaction <input checked="" type="radio"/> Set register value |  |
| Datapoints 91 - 100  | Register value 0   |  |
| Datapoints 101 - 110 | Register value (binary) 0000 0000 0000 0000b   |  |
| Datapoints 111 - 120 | Register value (hexadecimal) 0000h   |  |
| Datapoints 121 - 130 | Behavior on value match <input type="radio"/> No reaction <input checked="" type="radio"/> Set register value  |  |
| Datapoints 131 - 140 | Register value 1   |  |
| Datapoints 141 - 150 | Register value (binary) 0000 0000 0000 0001b   |  |
| Datapoints 151 - 160 | Register value (hexadecimal) 0001h   |  |
| Datapoints 161 - 170 | Behavior on value lower <input type="radio"/> No reaction <input checked="" type="radio"/> Set register value  |  |
| Datapoints 171 - 180 | Register value 0   |  |
| Datapoints 181 - 190 | Register value (binary) 0000 0000 0000 0000b   |  |
| Datapoints 191 - 200 | Register value (hexadecimal) 0000h   |  |
| Datapoints 201 - 210 | Function Write single holding register - 06  |  |
| Datapoints 211 - 220 | Address 0  |  |
| Datapoints 221 - 230 | Actual register value  |  |

### 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**.

### 6.14.3.2 Direction – Modbus to KNX

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10

|                      |  |   |
|----------------------|--|---|
| Description          | Datapoints 1 - 10  |   |
| General settings     | Description for this page <input type="text"/>   |   |
| Modbus settings      | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |   |
| - Datapoints         | Slave address (common) 1   |   |
| Datapoints 1 - 10    | Channel 1  |   |
| Datapoints 11 - 20   | Datapoint type   | DPT 07 - Configured - 2 bytes   |
| Datapoints 21 - 30   | Description  | <input type="text"/>  |
| Datapoints 31 - 40   | Direction  | <input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX                              |
| Datapoints 41 - 50   | Send condition   | On change   |
| Datapoints 51 - 60   | Type   | Value in word register  |
| Datapoints 61 - 70   | Bit count  | 16 bit  |
| Datapoints 71 - 80   | Offset from right  | 00 bit  |
| Datapoints 81 - 90   | Threshold  | 1   |
| Datapoints 91 - 100  | Threshold (binary)   | 0000 0000 0000 0001b  |
| Datapoints 101 - 110 | Threshold (hexadecimal)  | 0001h   |
| Datapoints 111 - 120 | Behavior on value higher   | <input type="radio"/> No reaction <input checked="" type="radio"/> Send value                                   |
| Datapoints 121 - 130 | Object value   | 0   |
| Datapoints 131 - 140 | Behavior on value match  | <input type="radio"/> No reaction <input checked="" type="radio"/> Send value                                   |
| Datapoints 141 - 150 | Object value   | 65535   |
| Datapoints 151 - 160 | Behavior on value lower  | <input type="radio"/> No reaction <input checked="" type="radio"/> Send value                                   |
| Datapoints 161 - 170 | Object value   | 0   |
| Datapoints 171 - 180 | Function   | <input checked="" type="radio"/> Read holding registers - 03<br><input type="radio"/> Read input registers - 04 |
| Datapoints 181 - 190 | Address  | 0   |
| Datapoints 191 - 200 | Polling interval   | Every cycle   |
| Datapoints 201 - 210 | Actual register value  |   |
| Datapoints 211 - 220 |  |   |

## 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).

## 6.15 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)

## 6.15.1 Type – Word register

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10 |  |
|--|--|
| Description  | Datapoints 1 - 10  |
| General settings   | Description for this page <input type="text"/>   |
| Modbus settings  | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |
| - Datapoints   | Slave address (common) 1   |
| <b>Datapoints 1 - 10</b>   | Channel 1  |
| Datapoints 11 - 20   | Datapoint type DPT 07 - Unsigned - 2 bytes   |
| Datapoints 21 - 30   | Description <input type="text"/>   |
| Datapoints 31 - 40   | Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX                   |
| Datapoints 41 - 50   | Type Word register   |
| Datapoints 51 - 60   | Position in word register <input type="radio"/> High/Low byte <input checked="" type="radio"/> Configured      |
| Datapoints 61 - 70   | Bit count 16 bit   |
| Datapoints 71 - 80   | Offset from right 00 bit   |
| Datapoints 81 - 90   | Valid bits in word register VVVV VVVV VVVV VVVVb   |
| Datapoints 91 - 100  | Valid values 0 ... 65535   |
| Datapoints 101 - 110   | Function Write single holding register - 06  |
| Datapoints 111 - 120   | Address 0  |
| Datapoints 121 - 130   | Actual register value <input type="text"/>   |

### 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.

## 6.16 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)

### 6.16.1 Type – Word register

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10 |  |
|--|--|
| Description  | Datapoints 1 - 10  |
| General settings   | Description for this page <input type="text"/>   |
| Modbus settings  | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |
| - Datapoints   | Slave address (common) 1   |
| <b>Datapoints 1 - 10</b>   | Channel 1  |
| Datapoints 11 - 20   | Datapoint type <b>DPT 08 - Signed - 2 bytes</b>  |
| Datapoints 21 - 30   | Description <input type="text"/>   |
| Datapoints 31 - 40   | Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX                   |
| Datapoints 41 - 50   | Type Word register   |
| Datapoints 51 - 60   | Position in word register High/Low byte  |
| Datapoints 61 - 70   | Function Write single holding register - 06  |
| Datapoints 71 - 80   | Address <input type="text" value="0"/>   |
| Datapoints 81 - 90   | <input type="text" value="Actual register value"/>   |

### 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

## 6.17 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)

## 6.17.1 Type – Word register

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10

|                          |  |   |
|--------------------------|--|---|
| Description              | Datapoints 1 - 10                                  |   |
| General settings         | Description for this page                          | <input type="text"/>  |
| Modbus settings          | Slave address used                                 | <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |
| - Datapoints             | Slave address (common)                             | 1   |
| <b>Datapoints 1 - 10</b> | Channel 1  |   |
| Datapoints 11 - 20       | Datapoint type                                     | DPT 09 - Float - 2 bytes  |
| Datapoints 21 - 30       | Description  | <input type="text"/>  |
| Datapoints 31 - 40       | Direction  | <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX          |
| Datapoints 41 - 50       | Type   | <input checked="" type="radio"/> Word register <input type="radio"/> Double word register   |
| Datapoints 51 - 60       | Position in word register                          | High/Low byte - 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   |
| Datapoints 121 - 130     | <input type="text" value="Actual register value"/> |   |
| Datapoints 131 - 140     |  |   |

### 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.*

## 6.17.2 Type – Double word register

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10

|                      |   |  |
|----------------------|---|--|
| Description          | Datapoints 1 - 10   |  |
| General settings     | Description for this page <input style="width: 100%;" type="text"/>   |  |
| Modbus settings      | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page  |  |
| Datapoints           | Slave address (common) 1  |  |
| Datapoints 1 - 10    | Channel 1   |  |
| Datapoints 11 - 20   | Datapoint type <span style="float: right;">▼</span> DPT 09 - Float - 2 bytes  |  |
| Datapoints 21 - 30   | Description <input style="width: 100%;" type="text"/>   |  |
| Datapoints 31 - 40   | Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX  |  |
| Datapoints 41 - 50   | Type <input type="radio"/> Word register <input checked="" type="radio"/> Double word register  |  |
| Datapoints 51 - 60   | Word order <input checked="" type="radio"/> Hi word @ address / Lo word @ address + 1 <input type="radio"/> Lo word @ address / Hi word @ address + 1 |  |
| Datapoints 61 - 70   | Type register value <span style="float: right;">▼</span> Modbus holds integer value - unsigned  |  |
| Datapoints 71 - 80   | Value minimum (register) <input style="width: 100%;" type="text" value="0"/>  |  |
| Datapoints 81 - 90   | Value maximum (register) <input style="width: 100%;" type="text" value="100"/>  |  |
| Datapoints 91 - 100  | Value minimum (KNX) <input style="width: 100%;" type="text" value="0"/>   |  |
| Datapoints 101 - 110 | Value maximum (KNX) <input style="width: 100%;" type="text" value="100"/>   |  |
| Datapoints 111 - 120 | Function Write single holding register - 06   |  |
| Datapoints 121 - 130 | Address <input style="width: 100%;" type="text" value="0"/>   |  |
| Datapoints 131 - 140 | <input type="button" value="Actual register value"/>  |  |
| Datapoints 141 - 150 |   |  |

### 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.

## 6.18 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)

### 6.18.1 Type – Bit register

#### 6.18.1.1 Direction – KNX to modbus

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10

|                      |   |
|----------------------|---|
| Description          | Datapoints 1 - 10   |
| General settings     | Description for this page <input style="width: 90%;" type="text"/>  |
| Modbus settings      | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page                      |
| - Datapoints         | Slave address (common) 1  |
| Datapoints 1 - 10    | Channel 1   |
| Datapoints 11 - 20   | Datapoint type <span style="border: 1px solid #ccc; padding: 2px;">DPT 12 - Configured - 4 bytes</span>                             |
| Datapoints 21 - 30   | Description <input style="width: 90%;" type="text"/>  |
| Datapoints 31 - 40   | Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX  |
| Datapoints 41 - 50   | Type <span style="border: 1px solid #ccc; padding: 2px;">Bit register</span>  |
| Datapoints 51 - 60   | Threshold <span style="border: 1px solid #ccc; padding: 2px;">4294967295</span>   |
| Datapoints 61 - 70   | Behavior on value higher <span style="border: 1px solid #ccc; padding: 2px;">Bit in register - Value '0'</span>                     |
| Datapoints 71 - 80   | Behavior on value match <span style="border: 1px solid #ccc; padding: 2px;">Bit in register - Value '1'</span>                      |
| Datapoints 81 - 90   | Behavior on value lower <span style="border: 1px solid #ccc; padding: 2px;">Bit in register - Value '0'</span>                      |
| Datapoints 91 - 100  | Function <span style="border: 1px solid #ccc; padding: 2px;">Write single coil - 05</span>  |
| Datapoints 101 - 110 | Address <span style="border: 1px solid #ccc; padding: 2px;">0</span>  |
| Datapoints 111 - 120 | <span style="border: 1px solid #ccc; padding: 2px; display: inline-block; width: 100px; height: 15px;">Actual register value</span> |

### 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’

### 6.18.1.2 Direction – Modbus to KNX

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10

|                      |   |  |
|----------------------|---|--|
| Description          | Datapoints 1 - 10   |  |
| General settings     | Description for this page <input type="text"/>  |  |
| Modbus settings      | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page        |  |
| - Datapoints         | Slave address (common) 1  |  |
| Datapoints 1 - 10    | Channel 1   |  |
| Datapoints 11 - 20   | Datapoint type <input type="text" value="DPT 12 - Configured - 4 bytes"/>   |  |
| Datapoints 21 - 30   | Description <input type="text"/>  |  |
| Datapoints 31 - 40   | Direction <input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX                          |  |
| Datapoints 41 - 50   | Send condition <input type="text" value="On change"/>   |  |
| Datapoints 51 - 60   | Type <input type="text" value="Bit register"/>  |  |
| Datapoints 61 - 70   | Behavior on bit in register - Value '1' <input type="radio"/> No reaction <input checked="" type="radio"/> Send value |  |
| Datapoints 71 - 80   | Object value <input type="text" value="4294967295"/>  |  |
| Datapoints 81 - 90   | Behavior on bit in register - Value '0' <input type="radio"/> No reaction <input checked="" type="radio"/> Send value |  |
| Datapoints 91 - 100  | Object value <input type="text" value="0"/>   |  |
| Datapoints 101 - 110 | Function <input checked="" type="radio"/> Read coils - 01 <input type="radio"/> Read discrete inputs - 02             |  |
| Datapoints 111 - 120 | Address <input type="text" value="0"/>  |  |
| Datapoints 121 - 130 | Polling interval <input type="text" value="Every cycle"/>   |  |
| Datapoints 131 - 140 | Actual register value <input type="text"/>  |  |
| Datapoints 141 - 150 |   |  |

## 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.

## 6.18.2 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.

### 6.18.2.1 Direction – KNX to modbus

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10 |  |
|--|--|
| Description  | Datapoints 1 - 10  |
| General settings   | Description for this page <input type="text"/>   |
| Modbus settings  | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |
| - Datapoints   | Slave address (common) 1   |
| <b>Datapoints 1 - 10</b>   | Channel 1  |
| Datapoints 11 - 20   | Datapoint type DPT 12 - Configured - 4 bytes   |
| Datapoints 21 - 30   | Description <input type="text"/>   |
| Datapoints 31 - 40   | Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX                   |
| Datapoints 41 - 50   | Type Bit in word register  |
| Datapoints 51 - 60   | Threshold 4294967295   |
| Datapoints 61 - 70   | Position in word register Bit 00   |
| Datapoints 71 - 80   | Valid bit in word register ---- - - - - -Vb  |
| Datapoints 81 - 90   | Behavior on value higher Bit in register - Value '0'   |
| Datapoints 91 - 100  | Behavior on value match Bit in register - Value '1'  |
| Datapoints 101 - 110   | Behavior on value lower Bit in register - Value '0'  |
| Datapoints 111 - 120   | Function Write single holding register - 06  |
| Datapoints 121 - 130   | Address 0  |
| Datapoints 131 - 140   | Actual register value <input type="text"/>   |

## 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'

## 6.18.2.2 Direction – Modbus to KNX

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10

|                          |   |   |
|--------------------------|---|---|
| Description              | Datapoints 1 - 10                       |   |
| General settings         | Description for this page               |   |
| Modbus settings          | Slave address used                      | <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page                     |
| - Datapoints             | Slave address (common)                  | 1   |
| <b>Datapoints 1 - 10</b> | Channel 1                               |   |
| Datapoints 11 - 20       | Datapoint type                          | DPT 12 - Configured - 4 bytes   |
| Datapoints 21 - 30       | Description                             |   |
| Datapoints 31 - 40       | Direction                               | <input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX                              |
| Datapoints 41 - 50       | Send condition                          | On change   |
| Datapoints 51 - 60       | Type                                    | Bit in word register  |
| Datapoints 61 - 70       | Position in word register               | Bit 00  |
| Datapoints 71 - 80       | Valid bit in word register              | ---- -- -- -- --Vb  |
| Datapoints 81 - 90       | Behavior on bit in register - Value '1' | <input type="radio"/> No reaction <input checked="" type="radio"/> Send value                                   |
| Datapoints 91 - 100      | Object value                            | 4294967295  |
| Datapoints 101 - 110     | Behavior on bit in register - Value '0' | <input type="radio"/> No reaction <input checked="" type="radio"/> Send value                                   |
| Datapoints 111 - 120     | Object value                            | 0   |
| Datapoints 121 - 130     | Function                                | <input checked="" type="radio"/> Read holding registers - 03<br><input type="radio"/> Read input registers - 04 |
| Datapoints 131 - 140     | Address                                 | 0   |
| Datapoints 141 - 150     | Polling interval                        | Every cycle   |
| Datapoints 151 - 160     | Actual register value                   |   |
| Datapoints 161 - 170     |   |   |

### 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.

## 6.18.3 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).

### 6.18.3.1 Direction – KNX to modbus

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10

|                      |                              |   |
|----------------------|------------------------------|---|
| Description          | Datapoints 1 - 10            |   |
| General settings     | Description for this page    | <input type="text"/>  |
| Modbus settings      | Slave address used           | <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |
| - Datapoints         | Slave address (common)       | 1   |
| Datapoints 1 - 10    | Channel 1                    |   |
| Datapoints 11 - 20   | Datapoint type               | DPT 12 - Configured - 4 bytes   |
| Datapoints 21 - 30   | Description                  | <input type="text"/>  |
| Datapoints 31 - 40   | Direction                    | <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX          |
| Datapoints 41 - 50   | Type                         | Value in word register  |
| Datapoints 51 - 60   | Threshold                    | 4294967295  |
| Datapoints 61 - 70   | Bit count                    | 16 bit  |
| Datapoints 71 - 80   | Offset from right            | 00 bit  |
| Datapoints 81 - 90   | Behavior on value higher     | <input type="radio"/> No reaction <input checked="" type="radio"/> Set register value       |
| Datapoints 91 - 100  | Register value               | 0   |
| Datapoints 101 - 110 | Register value (binary)      | 0000 0000 0000 0000b  |
| Datapoints 111 - 120 | Register value (hexadecimal) | 0000h   |
| Datapoints 121 - 130 | Behavior on value match      | <input type="radio"/> No reaction <input checked="" type="radio"/> Set register value       |
| Datapoints 131 - 140 | Register value               | 1   |
| Datapoints 141 - 150 | Register value (binary)      | 0000 0000 0000 0001b  |
| Datapoints 151 - 160 | Register value (hexadecimal) | 0001h   |
| Datapoints 161 - 170 | Behavior on value lower      | <input type="radio"/> No reaction <input checked="" type="radio"/> Set register value       |
| Datapoints 171 - 180 | Register value               | 0   |
| Datapoints 181 - 190 | Register value (binary)      | 0000 0000 0000 0000b  |
| Datapoints 191 - 200 | Register value (hexadecimal) | 0000h   |
| Datapoints 201 - 210 | Function                     | Write single holding register - 06  |
| Datapoints 211 - 220 | Address                      | 0   |
|                      | Actual register value        | <input type="text"/>  |

### 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**.

### 6.18.3.2 Direction – Modbus to KNX

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10

|                      |  |   |
|----------------------|--|---|
| Description          | Datapoints 1 - 10  |   |
| General settings     | Description for this page <input type="text"/>   |   |
| Modbus settings      | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |   |
| - Datapoints         | Slave address (common) 1   |   |
| Datapoints 1 - 10    | Channel 1  |   |
| Datapoints 11 - 20   | Datapoint type   | DPT 12 - Configured - 4 bytes   |
| Datapoints 21 - 30   | Description  | <input type="text"/>  |
| Datapoints 31 - 40   | Direction  | <input type="radio"/> KNX to modbus <input checked="" type="radio"/> Modbus to KNX                              |
| Datapoints 41 - 50   | Send condition   | On change   |
| Datapoints 51 - 60   | Type   | Value in word register  |
| Datapoints 61 - 70   | Bit count  | 16 bit  |
| Datapoints 71 - 80   | Offset from right  | 00 bit  |
| Datapoints 81 - 90   | Threshold  | 1   |
| Datapoints 91 - 100  | Threshold (binary)   | 0000 0000 0000 0001b  |
| Datapoints 101 - 110 | Threshold (hexadecimal)  | 0001h   |
| Datapoints 111 - 120 | Behavior on value higher   | <input type="radio"/> No reaction <input checked="" type="radio"/> Send value                                   |
| Datapoints 121 - 130 | Object value   | 0   |
| Datapoints 131 - 140 | Behavior on value match  | <input type="radio"/> No reaction <input checked="" type="radio"/> Send value                                   |
| Datapoints 141 - 150 | Object value   | 4294967295  |
| Datapoints 151 - 160 | Behavior on value lower  | <input type="radio"/> No reaction <input checked="" type="radio"/> Send value                                   |
| Datapoints 161 - 170 | Object value   | 0   |
| Datapoints 171 - 180 | Function   | <input checked="" type="radio"/> Read holding registers - 03<br><input type="radio"/> Read input registers - 04 |
| Datapoints 181 - 190 | Address  | 0   |
| Datapoints 191 - 200 | Polling interval   | Every cycle   |
| Datapoints 201 - 210 | Actual register value  |   |
| Datapoints 211 - 220 |  |   |

## 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).

## 6.19 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)



## 6.19.1 Type – Double word register

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10

|                          |  |   |
|--------------------------|--|---|
| Description              | Datapoints 1 - 10  |   |
| General settings         | Description for this page <input type="text"/>   |   |
| Modbus settings          | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |   |
| - Datapoints             | Slave address (common) 1   |   |
| <b>Datapoints 1 - 10</b> | Channel 1  |   |
| Datapoints 11 - 20       | Datapoint type   | DPT 12 - Unsigned - 4 bytes   |
| Datapoints 21 - 30       | Description  | <input type="text"/>  |
| Datapoints 31 - 40       | Direction  | <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX  |
| Datapoints 41 - 50       | Type   | Double word register  |
| Datapoints 51 - 60       | Word order   | <input checked="" type="radio"/> Hi word @ address / Lo word @ address + 1<br><input type="radio"/> Lo word @ address / Hi word @ address + 1 |
| Datapoints 61 - 70       | Position in double word register   | <input type="radio"/> All 4 bytes <input checked="" type="radio"/> Configured   |
| Datapoints 71 - 80       | Bit count  | 32 bit  |
| Datapoints 81 - 90       | Offset from right  | 00 bit  |
| Datapoints 91 - 100      | Valid bits in word register @ address  | VVVV VVVV VVVV VVVVb  |
| Datapoints 101 - 110     | Valid bits in word register @ address + 1  | VVVV VVVV VVVV VVVVb  |
| Datapoints 111 - 120     | Valid values   | 0 ... 4294967295  |
| Datapoints 121 - 130     | Function   | Write single holding register - 06  |
| Datapoints 131 - 140     | Address  | 0   |
| Datapoints 141 - 150     | Actual register value  | <input type="text"/>  |

### 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.

**6.20 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)

## 6.20.1 Type – Double word register

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10 |  |
|--|--|
| Description  | Datapoints 1 - 10  |
| General settings   | Description for this page <input type="text"/>   |
| Modbus settings  | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |
| - Datapoints   | Slave address (common) 1   |
| <b>Datapoints 1 - 10</b>   | Channel 1  |
| Datapoints 11 - 20   | Datapoint type DPT 13 - Signed - 4 bytes   |
| Datapoints 21 - 30   | Description <input type="text"/>   |
| Datapoints 31 - 40   | Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX                   |
| Datapoints 41 - 50   | Type Double word register  |
| Datapoints 51 - 60   | Word order <input checked="" type="radio"/> Hi word @ address / Lo word @ address + 1                          |
| Datapoints 61 - 70   | <input type="radio"/> Lo word @ address / Hi word @ address + 1  |
| Datapoints 71 - 80   | Position in double word register All 4 bytes   |
| Datapoints 81 - 90   | Function Write single holding register - 06  |
| Datapoints 91 - 100  | Address <input type="text" value="0"/>   |
| Datapoints 101 - 110   | <input type="text" value="Actual register value"/>   |

### 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.

## 6.21 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)

### 6.21.1 Type – Word register

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10

|                      |  |   |
|----------------------|--|---|
| Description          | Datapoints 1 - 10  |   |
| General settings     | Description for this page <input type="text"/>   |   |
| Modbus settings      | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |   |
| - Datapoints         | Slave address (common) 1   |   |
| Datapoints 1 - 10    | Channel 1  |   |
| Datapoints 11 - 20   | Datapoint type   | DPT 14 - Float - 4 bytes  |
| Datapoints 21 - 30   | Description  | <input type="text"/>  |
| Datapoints 31 - 40   | Direction  | <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX        |
| Datapoints 41 - 50   | Type   | <input checked="" type="radio"/> Word register <input type="radio"/> Double word register |
| Datapoints 51 - 60   | Position in word register  | High/Low byte - unsigned  |
| Datapoints 61 - 70   | Value minimum (register)   | <input type="text" value="0"/>  |
| Datapoints 71 - 80   | Value maximum (register)   | <input type="text" value="100"/>  |
| Datapoints 81 - 90   | Value minimum (KNX)  | <input type="text" value="0"/>  |
| Datapoints 91 - 100  | Value maximum (KNX)  | <input type="text" value="100"/>  |
| Datapoints 101 - 110 | Function   | Write single holding register - 06  |
| Datapoints 111 - 120 | Address  | <input type="text" value="0"/>  |
| Datapoints 121 - 130 | Actual register value  |   |
| Datapoints 131 - 140 |  |   |

### 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.*

## 6.21.2 Type – Double word register

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10 |  |
|--|--|
| Description  | Datapoints 1 - 10  |
| General settings   | Description for this page <input type="text"/>   |
| Modbus settings  | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |
| - Datapoints   | Slave address (common) 1   |
| <b>Datapoints 1 - 10</b>   | Channel 1  |
| Datapoints 11 - 20   | Datapoint type DPT 14 - Float - 4 bytes  |
| Datapoints 21 - 30   | Description <input type="text"/>   |
| Datapoints 31 - 40   | Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX                   |
| Datapoints 41 - 50   | Type <input type="radio"/> Word register <input checked="" type="radio"/> Double word register                 |
| Datapoints 51 - 60   | Word order <input checked="" type="radio"/> Hi word @ address / Lo word @ address + 1                          |
| Datapoints 61 - 70   | <input type="radio"/> Lo word @ address / Hi word @ address + 1  |
| Datapoints 71 - 80   | Type register value Modbus holds integer value - unsigned  |
| Datapoints 81 - 90   | Value minimum (register) 0   |
| Datapoints 91 - 100  | Value maximum (register) 100   |
| Datapoints 101 - 110   | Value minimum (KNX) 0  |
| Datapoints 111 - 120   | Value maximum (KNX) 100  |
| Datapoints 121 - 130   | Function Write single holding register - 06  |
| Datapoints 131 - 140   | Address 0  |
| Datapoints 141 - 150   | Actual register value <input type="text"/>   |

### 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.

## 6.22 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)

### 6.22.1 Type – Four word register

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10 |  |
|--|--|
| Description  | Datapoints 1 - 10  |
| General settings   | Description for this page  |
| Modbus settings  | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page |
| - Datapoints   | Slave address (common) 1   |
| <b>Datapoints 1 - 10</b>   | Channel 1  |
| Datapoints 11 - 20   | Datapoint type DPT 29 - Signed - 8 bytes   |
| Datapoints 21 - 30   | Description  |
| Datapoints 31 - 40   | Direction <input checked="" type="radio"/> KNX to modbus <input type="radio"/> Modbus to KNX                   |
| Datapoints 41 - 50   | Type Four word register  |
| Datapoints 51 - 60   | Word order <input checked="" type="radio"/> Hi word @ address / Lo word @ address + 3                          |
| Datapoints 61 - 70   | <input type="radio"/> Lo word @ address / Hi word @ address + 3  |
| Datapoints 71 - 80   | Position in four word register All 8 bytes   |
| Datapoints 81 - 90   | Function Write single holding register - 06  |
| Datapoints 91 - 100  | Address 0  |
| Datapoints 101 - 110   | Actual register value  |

### 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.

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

### Type

The following type is configured:

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

### 6.23.1 Type – Bit register

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10 |   |
|--|---|
| Description  | Datapoints 1 - 10   |
| General settings   | Description for this page <input type="text"/>  |
| Modbus settings  | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page                |
| - Datapoints   | Slave address (common) 1  |
| <b>Datapoints 1 - 10</b>   | Channel 1   |
| Datapoints 11 - 20   | Datapoint type Fixed value - 1 bit  |
| Datapoints 21 - 30   | Description <input type="text"/>  |
| Datapoints 31 - 40   | Direction KNX to modbus   |
| Datapoints 41 - 50   | Type Bit register   |
| Datapoints 51 - 60   | Register value <input checked="" type="radio"/> Bit in register - Value '1' <input type="radio"/> Bit in register - Value '0' |
| Datapoints 61 - 70   | Function Write single coil - 05   |
| Datapoints 71 - 80   | Address <input type="text" value="0"/>  |
| Datapoints 81 - 90   | Write interval <input checked="" type="radio"/> Once on startup <input type="radio"/> On startup and cyclically               |
| Datapoints 91 - 100  | <input type="button" value="Actual register value"/>  |
| Datapoints 101 - 110   |   |



## 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’

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

### Type

The following type is configured:

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

### 6.24.1 Type – Word register

1.1.1 KNX Modbus RTU Gateway 886.1 secure > Datapoints > Datapoints 1 - 10

|                      |   |
|----------------------|---|
| Description          | Datapoints 1 - 10   |
| General settings     | Description for this page <input type="text"/>  |
| Modbus settings      | Slave address used <input checked="" type="radio"/> from "Modbus settings" <input type="radio"/> for this page  |
| - Datapoints         | Slave address (common) 1  |
| Datapoints 1 - 10    | Channel 1   |
| Datapoints 11 - 20   | Datapoint type Fixed value - 2 bytes  |
| Datapoints 21 - 30   | Description <input type="text"/>  |
| Datapoints 31 - 40   | Direction KNX to modbus   |
| Datapoints 41 - 50   | Type Word register  |
| Datapoints 51 - 60   | Bit count 16 bit  |
| Datapoints 61 - 70   | Offset from right 00 bit  |
| Datapoints 71 - 80   | Register value 0  |
| Datapoints 81 - 90   | Register value (binary) 0000 0000 0000 0000b  |
| Datapoints 91 - 100  | Register value (hexadecimal) 0000h  |
| Datapoints 101 - 110 | Function Write single holding register - 06   |
| Datapoints 111 - 120 | Address 0   |
| Datapoints 121 - 130 | Write interval <input checked="" type="radio"/> Once on startup <input type="radio"/> On startup and cyclically |
| Datapoints 131 - 140 | <input type="text" value="Actual register value"/>  |

### 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**.

## 6.25 Converter N – M

1.1.1 KNX Modbus RTU Gateway 886.1 secure > DPT converter > Converters 1 - 10

|                  |                        |   |
|------------------|------------------------|---|
| Description      | Converters 1 - 10      |   |
| General settings | Converter 1            |   |
| Modbus settings  | Description            | <input type="text"/>  |
| + Datapoints     | Datapoint type - Input | Disabled  |
| - DPT converter  | Converter 2            | <ul style="list-style-type: none"> <li>Disabled ✓</li> <li>DPT 01 - Binary - 1 bit</li> <li>DPT 05 - Percent - 1 byte</li> <li>DPT 05 - Unsigned - 1 byte</li> <li>DPT 06 - Signed - 1 byte</li> <li>DPT 07 - Unsigned - 2 bytes</li> <li>DPT 08 - Signed - 2 bytes</li> <li>DPT 09 - Float - 2 bytes</li> <li>DPT 12 - Unsigned - 4 bytes</li> <li>DPT 13 - Signed - 4 bytes</li> <li>DPT 14 - Float - 4 bytes</li> <li>DPT 29 - Signed - 8 bytes</li> </ul> |
|                  | Description            | <input type="text"/>  |
|                  | Datapoint type - Input | Disabled  |
|                  | Converter 3            |   |
|                  | Description            | <input type="text"/>  |
|                  | Datapoint type - Input | Disabled  |
|                  | Converter 4            |   |
|                  | Description            | <input type="text"/>  |
|                  | Datapoint type - Input | 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 251 Converter 1: Input – Binary – 1 bit | 1.001    | 1 bit | From KNX  |

- DPT 05 – Percent – 1 byte

| Group object                                 | Type KNX | Size   | Direction |
|--|----------|--------|-----------|
| GO 251 Converter 1: Input – Percent – 1 byte | 5.001    | 1 byte | From KNX  |

- DPT 05 – Unsigned – 1 byte

| Group object                                  | Type KNX | Size   | Direction |
|---|----------|--------|-----------|
| GO 251 Converter 1: Input – Unsigned – 1 byte | 5.010    | 1 byte | From KNX  |

- DPT 06 – Signed – 1 byte

| Group object                                | Type KNX | Size   | Direction |
|---|----------|--------|-----------|
| GO 251 Converter 1: Input – Signed – 1 byte | 6.010    | 1 byte | From KNX  |

- DPT 07 – Unsigned – 2 bytes

| Group object                                   | Type KNX | Size    | Direction |
|--|----------|---------|-----------|
| GO 251 Converter 1: Input – Unsigned – 2 bytes | 7.001    | 2 bytes | From KNX  |

- DPT 08 – Signed – 2 bytes

| Group object                                 | Type KNX | Size    | Direction |
|--|----------|---------|-----------|
| GO 251 Converter 1: Input – Signed – 2 bytes | 8.001    | 2 bytes | From KNX  |

- DPT 09 – Float – 2 bytes

| Group object                                | Type KNX | Size    | Direction |
|---|----------|---------|-----------|
| GO 251 Converter 1: Input – Float – 2 bytes | 9.001    | 2 bytes | From KNX  |

- DPT 12 – Unsigned – 4 bytes

| Group object                                   | Type KNX | Size    | Direction |
|--|----------|---------|-----------|
| GO 251 Converter 1: Input – Unsigned – 4 bytes | 12.001   | 4 bytes | From KNX  |

- DPT 13 – Signed – 4 bytes

| Group object                                 | Type KNX | Size    | Direction |
|--|----------|---------|-----------|
| GO 251 Converter 1: Input – Signed – 4 bytes | 13.001   | 4 bytes | From KNX  |

- DPT 14 – Float – 4 bytes

| Group object                                | Type KNX | Size    | Direction |
|---|----------|---------|-----------|
| GO 251 Converter 1: Input – Float – 4 bytes | 14.000   | 4 bytes | From KNX  |

- DPT 29 – Signed – 8 bytes

| Group object                                 | Type KNX | Size    | Direction |
|--|----------|---------|-----------|
| GO 251 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 252 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 252 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 252 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 252 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 252 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 252 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 252 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 252 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 252 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 252 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 252 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 252 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**.

## 6.26 Converter function “Binary”

### Condition:

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

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > DPT converter > Converters 1 - 10 |                                   |
|---|-----------------------------------|
| Description   | Converters 1 - 10                 |
| General settings  | Converter 1                       |
| Modbus settings   | Description                       |
| + Datapoints  | Datapoint type - Input            |
| - DPT converter   | Datapoint type - Output           |
|   | Behavior on receiving data 'ON'   |
|   | Value                             |
|   | Behaviour on receiving data 'OFF' |
|   | 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.

## 6.27 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”.

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > DPT converter > Converters 1 - 10 |                          |
|---|--------------------------|
| Description   | Converters 1 - 10        |
| General settings  | Converter 1              |
| Modbus settings   | Description              |
| + Datapoints  | Datapoint type - Input   |
| - DPT converter   | Datapoint type - Output  |
| Converters 1 - 10   | Threshold                |
| Converters 11 - 20  | Behavior on value higher |
| Converters 21 - 30  | Behavior on value match  |
| Converters 31 - 40  | Behavior on value lower  |
| Converters 41 - 50  | Send cyclically          |

### 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'

## 6.28 Converter function “Percent”

### Condition:

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

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

| 1.1.1 KNX Modbus RTU Gateway 886.1 secure > DPT converter > Converters 1 - 10 |   |
|---|---|
| Description   | Converters 1 - 10   |
| General settings  | Converter 1   |
| Modbus settings   | Description <input type="text"/>  |
| + Datapoints  | Datapoint type - Input <input type="text" value="DPT 05 - Percent - 1 byte"/>                               |
| - DPT converter   | <input type="radio"/> DPT 01 - Binary - 1 bit<br><input checked="" type="radio"/> DPT 05 - Percent - 1 byte |
| Converters 1 - 10   | Percent value output [%] for input = 0% <input type="text" value="0 / 0x00 / 0.0%"/>                        |
| Converters 11 - 20  | Percent value output [%] for input = 100% <input type="text" value="255 / 0xFF / 100.0%"/>                  |
| Converters 21 - 30  | Send cyclically <input checked="" type="radio"/> Disabled <input type="radio"/> Enabled                     |
| Converters 31 - 40  |   |

### 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 %).*



## 6.29 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”

1.1.1 KNX Modbus RTU Gateway 886.1 secure > DPT converter > Converters 1 - 10

|                          |   |
|--------------------------|---|
| Description              | Converters 1 - 10   |
| General settings         | Converter 1   |
| Modbus settings          | Description <input type="text"/>  |
| + Datapoints             | Datapoint type - Input <span>DPT 09 - Float - 2 bytes</span>                            |
| - DPT converter          | Datapoint type - Output <span>DPT 09 - Float - 2 bytes</span>                           |
| <b>Converters 1 - 10</b> | Scaling factor <input type="text" value="1"/>   |
| Converters 11 - 20       | <i>i</i> Input * Scaling factor = Output  |
| Converters 21 - 30       | Send cyclically <input checked="" type="radio"/> Disabled <input type="radio"/> Enabled |

### Scaling factor

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

## 6.30 General information

### 6.30.1 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.***

### 6.30.2 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.

### 6.30.3 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.

## 6.30.4 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*



## 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.



### Product database for ETS 5/6

[www.weinzierl.de/en/products/886.1/ets6](http://www.weinzierl.de/en/products/886.1/ets6)

### Data sheet

[www.weinzierl.de/en/products/886.1/datasheet](http://www.weinzierl.de/en/products/886.1/datasheet)

### CE Declaration

[www.weinzierl.de/en/products/886.1/ce-declaration](http://www.weinzierl.de/en/products/886.1/ce-declaration)

### WEINZIERL ENGINEERING GmbH

Achatz 3-4  
84508 Burgkirchen an der Alz  
GERMANY

Tel.: +49 8677 / 916 36 – 0

E-Mail: [info@weinzierl.de](mailto:info@weinzierl.de)

Web: [www.weinzierl.de](http://www.weinzierl.de)

2024-08-05