Technical Manual

State 04/2024 - Version 1.0

KNX Modbus Gateway RTU485

SCN-MBGRTU.01

Further Documents:

Datasheet:
https://www.mdt.de/EN_Downloads_Datasheets.html

Assembly and Operation Instructions:
https://www.mdt.de/EN_Downloads_Instructions.html

Solution Proposals for MDT products:
https://www.mdt.de/en/for-professionals/tips-tricks.html

MDT technologies GmbH · Papiermühle 1 · 51766 Engelskirchen · Germany
Telephone: +49 (0) 2263 880 · knx@mdt.de · www.mdt.de

1 Content

2 Overview

2.1 Overview devices

This manual refers to the following devices (order number in bold).

- **SCN-MBGRTU.01** KNX Modbus Gateway RTU485, 2SU MDRC

2.2 Functions

Modbus operating mode

Comprehensive setting options enables the connection to the existing Modbus as a master or additional slave. The Modbus protocol can be RTU or ASCII. Detailed error and diagnostic objects are available to the user for commissioning the gateway.

Templates for Modbus device settings

Modbus participants devices can be individually pre-configured and used as a template to configure each channel. This saves time-consuming reconfiguration of each individual channel. No limit to the number of devices, each channel can be used individually without a device template.

Channel templates

Channel templates simplify the channel configuration. Settings such as the communication direction (Modbus to KNX, KNX to Modbus or bi-directional), the KNX data point type, the send condition and the function type can be reused in each channel.

Channels

Each of the 200 channels can be set individually or based on the user configured templates. When using the channel and device templates, only the Modbus register address to be read or written and the priority of the channel need to be specified. If the templates are not applied, each channel can be individually configured.

Bi-directional communication

In addition to the directions “Modbus -> KNX” and “KNX -> Modbus”, each channel can also be operated bi-directionally. In this operation mode values can be transmitted over a single channel, written to a single Modbus register address and status read out without the use of a second channel.

Multi-channel read

A multi-channel read function can be activated for consecutive register addresses of a Modbus slave. This function is necessary whenever a value is associated with a changing factor. In this case, multi-channel reading ensures that the associated information is read in one cycle.

Mathematics

The MDT Modbus Gateway offers a wide range of mathematical functions to convert the values received or to be transmitted. These include scaling, range conversion, range limitation, binary functions and the basic arithmetic operations: addition, subtraction, multiplication and division. An external logic module to convert the values is not required.

Comparators

A comparator with up to 4 comparison values (equal, unequal, greater than, less than) is available for each channel. The output values for a fulfilled or not fulfilled comparison can be transmitted via a common or separate objects. The data point type of each comparator can be set.

Message texts

Up to 10 message text functions can be activated, each with up to 10 different 14-byte message texts. The value of a channel is compared with a comparable value. If a comparison (equal, unequal, greater than, less than) is fulfilled, the specified message text is written to the 14-byte output object. This function can be used, for example, to display various states of a Modbus device as plain text via KNX.

Updateable via DCA app

If necessary, the device can be updated using the MDT update tool (DCA).
The download is available free of charge at www.mdt.de and www.knx.org.

Long Frame Support

The device supports “long frames” (longer telegrams). These contain more user data per telegram, which significantly reduces the programming time.

2.3 Wiring diagram

The following figure shows an exemplary wiring diagram:

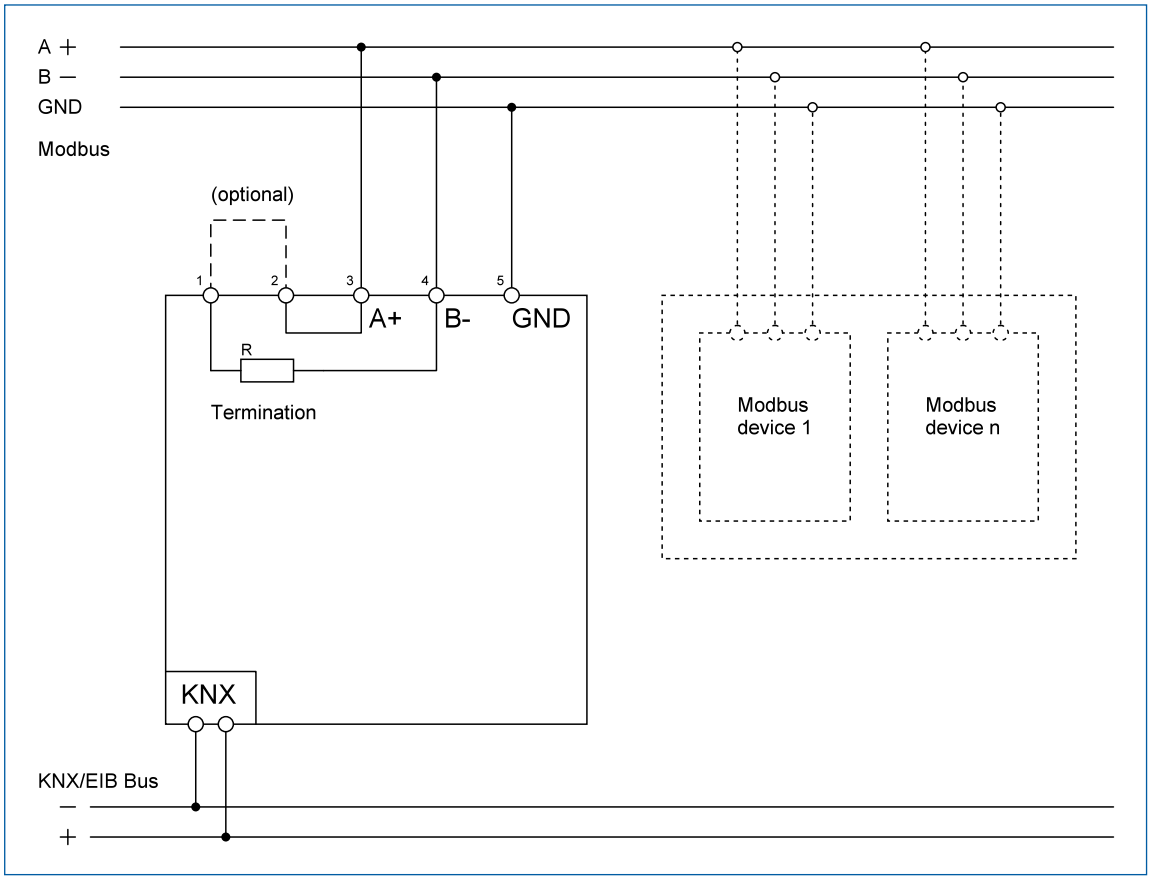

Figure 1: Wiring diagram

2.4 Structure & Handling

The following picture shows the structure of the device:

Figure 2: Structure & Handling

1 = KNX Bus connection terminal
2 = Programming button and LED
3 = Modbus connection terminal
4 = Status-LEDs

2.5 Commissioning

1. Wire the device according to the wiring diagram.
2. Connect programming interface to the bus.
3. Switch on bus voltage.
4. Press the programming button on the device (red programming LED lights up continuously).
5. Set and programme the individual address in the ETS. (Programming LED turns off)
6. Configure and programme the settings in the application programme.

Note: Due to the large number of configuration options, the database may take a long time to load. Partial programming is possible without any problems once the database has been fully programmed with the physical address. This considerably reduces the loading times.

3 Communication objects

3.1 Standard settings of the communication objects

Standard settings								
No.	Name	Object Function	Length	C	R	W	T	U
1	Channel 1:	Input	1 Bit 1 Byte 2 Byte 4 Byte 14 Byte	■		■		
2	Channel 1:	Output	1 Bit 1 Byte 2 Byte 4 Byte	■	■		■	
2	Channel 1:	Status/Output	1 Bit 1 Byte 2 Byte 4 Byte	■	■		■	
+2	**next Channel**							
401	Comparator 1:	Common output value	1 Bit 1 Byte 2 Byte 4 Byte	■	■		■	
401	Comparator 1:	Output value 1	1 Bit 1 Byte 2 Byte 4 Byte	■	■		■	
402	Comparator 1:	Output value 2	1 Bit 1 Byte 2 Byte 4 Byte	■	■		■	
403	Comparator 1:	Output value 3	1 Bit 1 Byte 2 Byte 4 Byte	■	■		■	
404	Comparator 1:	Output value 4	1 Bit 1 Byte 2 Byte 4 Byte	■	■		■	
+4	**next Comparator**							

Standard settings								
No.	Name	Object Function	Length	C	R	W	T	U
441	Message text 1:	Output	14 Byte	■	■		■	
+1	**next Message text**							
452	In operation	Output	1 Bit	■	■		■	
453	Diagnosis object	Output	14 Byte	■	■		■	
454	Error	Output	1 Bit	■	■		■	

Table 1: Communication objects – Standard settings

The table above shows the preset default settings. The priority of the individual communications objects and the flags can be adjusted by the user as required. The flags assign the communication objects their respective tasks in programming, where C stands for communication, R for read, W for write, T for transmit and U for update.

4 ETS Parameter

4.1 General Settings

The following table shows the available settings:

ETS Text	Dynamic range **[Default value]**	Comment
Startup time	2 ... 240 s **[2 s]**	Sets the time between restart and functional start-up of the device.
Send „In operation“ cyclically	**not active** 1 min – 24 h	Activation of a cyclical “In operation” telegram.

Table 2: General settings

Startup time
This time defines when the device “boots up” after a restart (reset, reprogramming, bus voltage return). This can be important if, for example, a bus reset is carried out. If there are many devices on a line, all devices would start at the same time and place a load on the bus. With a variable time, the devices can start differently.

Send „In operation“ cyclically
The “In operation” telegram is used to show on the bus that the device is “alive”. If activated, an ON telegram is sent cyclically.

The following table shows the associated communication objects:

Number	Name / Object function	Length	Usage
452	In operation – Output	1 Bit	Sending a cyclical “In operation” telegram

Table 3: Communication object – General settings

4.2 General Modbus settings

The following table shows the available settings:

ETS Text	Dynamic range **[Default value]**	Comment
Operating mode	■ **Master** ■ Slave	Setting in which operating mode the device should work.
Modbus protocol	■ **RTU** ■ ASCII	Setting according to which protocol the device should work.
Baud rate	1200, 2400, 4800, **9600**, 19200, 38400, 56000, 115200 Bit/s	Definition of the transmission speed.
Parity	■ even / 1 Stop bit ■ odd / 1 Stop bit ■ none / 1 Stop bit ■ even / 2 Stop bits ■ odd / 2 Stop bits ■ **none / 2 Stop bits**	Setting the parity.
Timeout after request	10 / 20 / 30 / 50 / 100 / 200 / 300 / **500** ms	Setting the time between the request from the master and the response from the slave. **Only for "Master" operating mode.**
Minimum time between requests	10 / 20 / 30 / 50 / **100** / 200 / 300 / 500 ms	Definition of the minimum distance between several requests. **Only for "Master" operating mode.**
Minimum cycle time	■ minimum ■ **500 ms** ■ 1 s ■ 2 s : ■ 10 s ■ 20 s ■ 30 s ■ 1 min	Setting the minimum time that a cycle lasts before the next cycle begins. **Only for "Master" operating mode.**
Own Modbus slave address	0 ... 247 **[1]**	Assigning an address for the device. **Only for "Slave" operating mode.**
Error and diagnosis object	■ **not active** ■ active	Activation of 2 objects to output an error message and a diagnosis text. A table with the codes is displayed at the same time.

Table 4: General Modbus settings

The basic Modbus settings are defined in this menu:

The “**Operating mode**” determines whether the gateway works as a “Master” or “Slave” within the system.

The protocol communication is defined in the “**Modbus protocol**”. Either “RTU” for binary representation of the data, or “ASCII” for the use of ASCII characters.

The communication parameters are set under “**Baud rate**” and “**Parity**”.

Own Modbus slave address
In “Slave” operating mode, the device is assigned an individual slave address here, which is then addressed by the master.

General: Write/read processes and time behaviour

The MDT Modbus Gateway offers up to 200 channels (with up to 400 channel-related objects), which can be used either as a read **or** write channel or as a read **and** write channel respectively.
Single operations are processed sequentially, i.e. the channels are processed one after the other.
The timing of a cycle (each channel is processed once) is influenced by the following conditions:

- Number of activated channels.
- Parameter setting “Minimum time between requests”.
- Parameter setting „Minimum cycle time“.
- Number of channels summarised as a “multi-channel operation”.

A read-channel triggers a read operation on the Modbus each time it is in line in the processing sequence.
A send-channel triggers a sending operation on the Modbus each time it is in line in the processing sequence **and** if a write operation has previously been triggered on the object assigned to the channel.

The „**Timeout after request**“ parameter describes the waiting time for the response from the slave (after a request from the master). If no response is received after this time has elapsed, an error message is output as a diagnostic text and the channel is skipped.

The “**Minimum time between requests**” describes the time between the response of a slave and the next request. This means that all channels are processed one after the other.

Each active channel is requested once in a cycle. The “**Minimum cycle time**” is the minimum time that an entire cycle takes. This can be exceeded, but not fallen short of.
Example: Minimum cycle time = 200 ms. If the cycle has already ended after 50 ms, the system waits a further 150 ms until the next cycle.
The “**Minimum**” setting does not mean a fixed time. The next cycle starts immediately after the end of the last cycle.

Error and diagnosis object
When activated, two new objects appear. The diagnosis object sends a 14 byte diagnosis text, the error object sends a 1-Bit status (“1” = error, “0” = no error).

The diagnosis object (14-byte string) outputs a pending error as follows:

„EXX YYYYY cZZZ“

EXX: Internal error code.
YYYYY: Error message (see table below).
cZZZ: “c” stands for “Channel”. “ZZZ” indicates the channel number.

The following table shows the possible error messages:

Message	Comment
ARG / VALUE	Data error, e.g. value out of range
INT	Internal Modbus gateway error
BUS	Modbus communication error (interfering signals, wrong parameters, etc.)
NO_ANS	No response from the addressed Modbus device
FUNC	Unknown function code
ADDR	Unknown register address
SLAVE	Modbus slave returns an internal error
GATW	Error of an additionally used Modbus gateway / repeater

Table 5: Diagnosis texts

Alternatively, the diagnosis object can also output an information text about a “NaN value”.
“INFO1 NaN cZZZ” means that the channel has received a value configured as a “NaN value”.

The following table shows the associated communication objects:

Number	Name / Object function	Length	Usage
453	Diagnosis object – Output	14 Byte	Output of a text message
454	Error – Output	1 Bit	Output of the error status

Table 6: Communication objects – General Modbus settings

4.3 Presets: Modbus devices

Up to 10 presets can be activated here via the "**Number of presets**" parameter. Settings can be defined in a preset, which can later be used to simplify the configuration of the channels. A separate submenu appears in the menu tree for each activated preset (1 - 10).

4.3.1 Preset 1 - 10

The following table shows the available settings:

ETS Text	Dynamic range **[Default value]**	Comment
Preset 1:	■ not active ■ **active**	Activation/deactivation of the preset.
Description	any text (30 bytes allowed)	Enter a name for the corresponding device.
Modbus Slave address	1 ... 247 **[1]**	Definition of the slave address for the device. **Only in "Master" operating mode.**
Register address	■ **first address „0"** ■ first address „1"	Setting whether the first register address is a "0" or a "1".
Byte order - 2 Byte	■ LSB first (BA) ■ **MSB first (AB)**	Defines the order in which the bytes are to be sent.
Byte order - 4 Byte	■ LSB first (DCBA) ■ **MSB first (ABCD)** ■ LSB first / Bytes swapped (CDAB) ■ MSB first / Bytes swapped (BADC)	Defines the order in which the bytes are to be sent.

Table 7: Settings – Modbus devices

A text field for free labelling is available for each "Preset".
Meaningful labelling simplifies assignment in the ETS project:

Figure 3: Text field – Description

A text of up to 30 characters can be stored for the field. The text entered in "**Device description**" appears in the menu behind the corresponding preset.

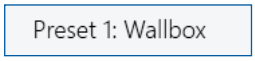

Figure 4: Presentation – Description

Register address
The addressing is set here. The start address can be a “1” (1-based) or a “0” (0-based). The information on this can be found in the technical documentation for the device and must be set accordingly.

Byte order (2 Byte / 4 Byte)
These parameters define the order in which the bytes are transferred.

LSB stands for “Least Significant Bit” - The sequence begins with the lowest significant bit.
MSB stands for “Most Significant Bit” - The sequence begins with the highest significant bit.

4.4 Presets: Channels

Up to 20 presets can be activated with the "**Number of presets**" parameter.
Settings can be defined in a preset which can later be used to simplify the configuration of the channels.
A separate menu appears in the menu tree for each activated preset.

4.4.1 Preset 1 - 20

The following table shows the available settings:

ETS Text	Dynamic range **[Default value]**	Comment
Preset	■ not active ■ **active**	Activation/deactivation of the preset.
Description	any text (30 bytes allowed)	Free text input to describe the preset.
Direction	■ **Modbus (R) to KNX** ■ KNX to Modbus (W) ■ Modbus / KNX bidirectional (R/W)	Setting the send/receive direction KNX <-> Modbus. **„ ... bidirectional (R/W)" only with operation mode „Master" .**
KNX		
KNX Datapoint type	■ **1 Bit DPT 1.001 On/Off** ■ 1 Bit DPT 1.003 Enable ■ 1 Byte DPT 5.001 Percent (0...100%) ■ 1 Byte DPT 5.005 Decimal value (0...255) ■ 1 Byte DPT 6.010 Value (-128...127) ■ 1 Byte DPT 17.001 Scene number ■ 2 Byte DPT 7.012 Current [mA] ■ 2 Byte DPT 7.013 Brightness [Lux] ■ 2 Byte DPT 9.001 Temperature [°C] ■ 2 Byte DPT 9.004 Brightness [Lux] ■ 2 Byte DPT 9.005 Speed [m/s] ■ 2 Byte DPT 9.007 Humidity [%] ■ 2 Byte DPT 9.008 air quality [ppm] ■ 2 Byte DPT 9.020 Voltage [mV] ■ 2 Byte DPT 9.021 Current [mA] ■ 2 Byte DPT 9.024 Power [kW] ■ 2 Byte DPT 9.025 Flow rate [l/h] ■ 4 Byte DPT 12.1200 Volume [l] ■ 4 Byte DPT 12.1201 Volume [m³] ■ 4 Byte DPT 13.002 Flow rate [m³/h] ■ 4 Byte DPT 13.010 Active energy [Wh] ■ 4 Byte DPT 13.013 Active energy [kWh]	Setting the KNX datapoint type.

ETS Text	Dynamic range **[Default value]**	Comment
KNX datapoint type	■ 4 Byte DPT 14.019 Current [A] ■ 4 Byte DPT 14.027 Electric potential [V] ■ 4 Byte DPT 14.033 Frequency [Hz] ■ 4 Byte DPT 14.056 Power [W] ■ 4 Byte DPT 14.065 Speed [m/s] ■ 4 Byte DPT 14.076 Volume [m³] ■ 4 Byte DPT 14.077 Flow rate [m³/s] ■ 4 Byte DPT 14.1200 Flow rate [m³/h] ■ 14 Byte DPT 16.000 String ■ 2 Byte DPT 7.* unsigned ■ 2 Byte DPT 8.* signed ■ 2 Byte DPT 9.* floating point value ■ 4 Byte DPT 12 * unsigned ■ 4 Byte DPT 13 * signed ■ 4 Byte DPT 14.* floating point value	Setting the KNX datapoint type.
KNX sending condition	■ read only ■ on change ■ cyclic ■ **on change and cyclic**	Setting whether and when a value is to be sent. **Only for direction „Modbus (R) to KNX“.**
Send on change of ...	Any value according to the set datapoint type	Setting as of which change the value should be sent again. **Only „on change“ and KNX datapoint type 1/2/4 Byte.**
Send cyclically every ...	10 s – 24 h **[10 min]**	Setting the interval at which cyclical transmission should take place. **Only if “... cyclic” is selected.**
Modbus		
Modbus function code	■ **0x01 read coil status** ■ 0x02 read inputs status) ■ 0x03 read holding register) ■ 0x04 read input register)	Setting the function code. **With direction „Modbus (R) to KNX“ and KNX DPT „1 Bit“.**
Modbus function code	■ **0x03 read holding register** ■ 0x03 read holding register, multiread ■ 0x04 read input register	Setting the function code. **With direction “Modbus (R) to KNX” and KNX DPT “1/2/4 byte”.**
Modbus function code	■ **0x05 write single coil** ■ 0x06 / 0x16 write single or multiple register	Setting the function code. **With direction “KNX to Modbus (W)” and KNX DPT “1 bit”.**
Modbus function code	0x06 / 0x16 write single or multiple register	Fixed setting, cannot be changed. **With direction “KNX to Modbus (W)” and KNX DPT “1/2/4 byte”.**

ETS Text	Dynamic range **[Default value]**	Comment
Modbus data type	■ **1 Byte unsigned** ■ 1 Byte signed ■ 2 Byte unsigned ■ 2 Byte signed ■ 4 Byte unsigned ■ 4 Byte signed ■ 4 Byte floating point value	Setting of the Modbus data type. **Only visible when KNX DPT "1/2/4 byte" is also set.**
Output function with binary input object	■ **Bit in Register** ■ Send value	Setting of the function that is to be executed when a binary input object is received. **Only available if:** ■ **Direction „KNX to Modbus"** ■ **KNX DPT „1 Bit"** ■ **Function code „0x06/0x16 ...".**
Value for ON / OFF	0 ... 255 / 0 ... 65535 / 0 ... 4294967295 **[0]**	Setting the value to be sent. ■ **If "Send value" is selected** ■ **Value range depends on the Modbus data type.**
Bit position in register	0 ... 7 / 0 ... 15 / 0 ... 31 **[0]**	Setting the bit position. ■ **Value range depends on the Modbus data type.** ■ **Only for KNX DPT "1 Bit".**
Bit inverted	■ **not active** ■ active	Setting whether the outgoing Bit should be inverted. **Only with KNX DPT „1 Bit".**
Not a Number (invalid value)	■ **not active** ■ active	Activation of a NaN value. **Only with KNX DPT „1/2/4 Byte".**
Value	0xFF / 0xFFFF / 0xFFFFFFFF	Input of the NaN value. Number of "F" depending on the data type. **Only if "Not a Number (Invalid value)" is active.**
Error message for Modbus value = NaN value (invalid value)	■ **not active** ■ active	Activation of an error message.
String length	■ **2** ■ 4 ... ■ 14	Setting of how many characters (bytes) the string consists of. **For KNX DPT „14 Byte String" and direction „Modbus (R) to KNX".**

Table 8: Settings – Channel presets

Text field „Description"
A text field for free labelling is available for each "Preset". Meaningful labelling simplifies assignment in the ETS project:

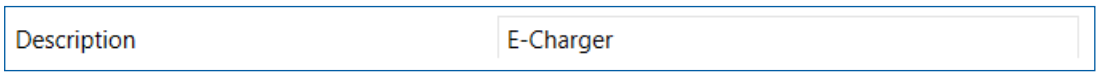

Figure 5: Text field – Description

A text of up to 30 characters can be stored for the field. The text entered in "Description" appears in the menu behind the corresponding preset:

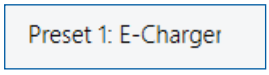

Figure 6: Channel preset – Description

KNX
The KNX datapoint type and the KNX sending conditions are defined in this parameter block.

Modbus
The Modbus specific settings for the preset are defined in this parameter block:

Modbus Function code
The MDT Modbus Gateway supports various Modbus function codes. The function codes define how the Modbus master accesses the Modbus register addresses of the slave.
The slave and the corresponding register address that is accessed must support the respective function code. Otherwise, the slave sends back an error code. In the ETS, the function code that can be used sensibly in each case is offered for configuration according to the KNX data type.

Special feature – Multi-channel read function
In some applications, it makes sense to request and evaluate read operations on the Modbus side in one Modbus frame. The MDT Modbus Gateway supports this when using the function code "0x03 read holding register, multiread".
If this Modbus function code is selected in the KNX application, the gateway automatically forms a read sequence of up to 10 channels under the following conditions:

- The channels have consecutive register addresses (address "+1" for Modbus data types "1 Byte/2 Byte". Address "+2" for Modbus data type "4 Byte").
- The channels are configured with the function code "read holding register, multiread (0x03)".
- The channels are configured with the direction "Modbus (R) to KNX".
- The set KNX data types are numerical data types.

In addition to the automatic creation of a multi-channel read sequence, the user can determine the first channel of a desired sequence himself by activating the parameter "**Start channel for function code "multiread""**.
Important: „Start channel for function code "multiread"" must be activated in the corresponding channel.

Output function with binary input object
With this function, it is possible to send a fixed output value (corresponding to the input values “1” and “0”) when a binary object is received. Alternatively, the bit position can be defined in the register.

Important: This function requires certain settings. These are:

- Direction „KNX to Modbus (W)“
- KNX Datapoint type „1 Bit DPT 1.x ...“
- Modbus function code „write single or multiple register (0x06 / 0x16 ...)

Bit position in register
If the KNX datapoint type is set to “1 Bit”, the position of the bit in the register can be determined here. The parameter therefore only appears if a Modbus function code “... register” is selected.

Bit inverted
The parameter appears if the KNX datapoint type is set to “1 Bit”. The bit can be displayed normally or inverted.

Not a Number (invalid value)
The NaN (Not a Number) function can be activated to filter out values sent by the Modbus slave in an error state.
If, for example, a participant always sends a “0xFFFF” in sleep mode (this would be an invalid value here), this value is not processed further and is not transmitted to the KNX output object.

Error message for Modbus value = NaN value (invalid value)
If activated, an error message (Warning) would be issued via the diagnosis object in the event of an error.
See 4.2 General Modbus settings , Diagnosis object.

4.4.2 Mathematics

With the “KNX datapoint type - 1/2/4 Byte” setting, a “Mathematics” parameter block appears. Various mathematical functions can be realised there. The following table shows the available settings:

ETS Text	Dynamic range **[Default value]**	**Comment**
Function	■ **not active** ■ Scaling ■ Arithmetics ■ Binary function ■ Range conversion ■ Range limitation	Setting a mathematical function. **Binary function only with DPT selection “KNX 5.005 and Modbus 1 Byte unsigned”**
Scaling		
Scaling	■ x 100,000 ■ x 10,000 ... ■ **x 1** ■ x 0.1 ... ■ x 0.00001	Determination of the multiplier for scaling. **For direction “Modbus (R) to KNX” and “KNX to Modbus (W)”.**
Scaling	■ Modbus (W) x 100,000 / Modbus (R) x 0.00001 ■ Modbus (W) x 10,000 / Modbus (R) x 0.0001 : ■ **Modbus (W) x 1 / Modbus (R) x 1** ■ Modbus (W) x 0.1 / Modbus (R) x 10 : ■ Modbus (W) x 0.00001 / Modbus (R) x 100,000	Determination of the multiplier for scaling. **For direction “Modbus / KNX bidirectional (R/W)”.**
Arithmetics		
Arithmetics	■ Addition: channel value + value 1 ■ Subtraction: channel value - value 1 ■ Subtraction: value 1 - channel value ■ **Multiplication: channel value x value 1** ■ Multiplication with Offset: value 1 + channel value x value 2 ■ Multiplication: channel value x 10^value 1 ■ Multiplication: 10^channel value x value 1 ■ Division: channel value / value 1 ■ Division: value 1 / channel value	Definition of the calculation type and the terms involved. **Only for direction “Modbus (R) to KNX” and “KNX to Modbus (W)”.**

ETS Text	Dynamic range **[Default value]**	Comment
Arithmetics	■ Modbus (W): channel value + value 1; Modbus (R): channel value - value 1 ■ Modbus (W): channel value - value 1; Modbus (R): channel value + value 1 ■ Modbus (W): value 1 - channel value; Modbus (R): value 1 + channel value ■ **Modbus (W): channel value x value 1; Modbus (R): channel value / value 1** ■ Modbus (W): value 1 + channel value x value 2; Modbus (R): (channel value - value 1) / value 2 ■ Modbus (W): 10^value 1 x channel value; Modbus (R): channel value / 10^value 1 ■ Modbus (W): 10^channel value x value 1; Modbus (R): value 1 / 10^channel value ■ Modbus (W): channel value / value 1; Modbus (R): channel value x value 1 ■ Modbus (W): value 1 / channel value; Modbus (R): channel value x value 1	Definition of the calculation type and the terms involved. **Only for direction „Modbus / KNX bidirectional (R/W)“.**
Operation with ...	■ **value 1** ■ value 1 is output value from other channel	Setting whether the value required for the operation is defined in this channel or whether the value of another channel should be used.
Value 1	-10000000000000 ... 10000000000000 **[0]**	Definition of value 1.
Value 2	-10000000000000 ... 10000000000000 **[0]**	Definition of value 2. **Only if selected with “ ... value 2”**
Binary function (only available under certain conditions - see description below the table)		
Binary function	■ **Bit masking with fixed value** ■ Bitshift left ■ Bitshift right ■ AND ■ OR ■ XOR	Setting the corresponding function.

ETS Text	Dynamic range **[Default value]**	Comment
Number system	■ **Decimal** ■ Hex	Setting the number system for the fixed value. **Only for "Bit masking with fixed value", „AND", „OR", „XOR".**
Value	0 ... 255 **[0]**	Definition of the fixed value. **Only if "Decimal" is selected.**
Value	free Input (4 Bytes allowed) **[0xFF]**	Definition of the fixed values. **Only if "Hex" is selected.**
Bitshift	0 ... 8 **[0]**	Setting for how many positions to be shifted to the left or right. **For "Bitshift left/right".**
Range conversion		
KNX: Lower limit value	Any value according to the set KNX datapoint type	Definition of the lower limit value for the range.
KNX: Upper limit value	Any value according to the set KNX datapoint type	Definition of the upper limit value for the range.
Modbus: Lower limit value	Any value according to the set Modbus data type	Definition of the lower limit value for the range.
Modbus: Upper limit value	Any value according to the set Modbus data type	Definition of the upper limit value for the range.
Range limitation		
KNX: Lower limit value	Any value according to the set KNX datapoint type	Definition of the lower limit value for the range.
KNX: Upper limit value	Any value according to the set KNX datapoint type	Definition of the upper limit value for the range.
Modbus: Lower limit value	Any value according to the set Modbus data type	Definition of the lower limit value for the range.
Modbus: Upper limit value	Any value according to the set Modbus data type	Definition of the upper limit value for the range.

Table 9: Settings – Mathematics

Scaling

The function can be used to realise a decimal point shift.

Arithmetics

Operations can be implemented here according to basic arithmetic operations. An extensive selection is available in the drop-down menu.

With „**Value 1 is output value of other channel**“, the channel selection is defined in the corresponding channel.

Binary function

The binary functions enable simple bit-by-bit binary calculation functions for the Modbus value read in or to be sent.

Important: The binary function is only available with the following selection:

- KNX DPT 1/2/4 byte unsigned
- Modbus data type 1/2/4 byte unsigned
- Both data types must have the same length
- Direction: “Modbus (R) to KNX” or “KNX to Modbus (W)”

The following applies in **master mode** of the Modbus gateway:
If the “Modbus (R) to KNX” direction is set, the data sent by the Modbus participant is linked bit by bit with the fixed value entered under “Value” and written to the KNX send object.
If the “KNX to Modbus (W)” direction is set, the data in the KNX receive object is linked bit by bit with the fixed value entered under “Value” and sent to the Modbus.

The following applies in **slave mode** of the Modbus gateway:
If the “KNX to Modbus (W)” direction is set, the data sent by the Modbus master is linked bit by bit with the fixed value entered under “Value” and written to the KNX send object.
If the “Modbus (R) to KNX” direction is set, the data in the KNX receive object is linked bit by bit with the fixed value entered under “Value” and sent to the Modbus master.

The “Bit masking”, “AND”, “OR” and “XOR” operations involve a bit-by-bit linking of the data.

The following table shows examples of the master mode:

Value from Modbus (R) or value from KNX receiver object (W)	Entered fixed value	Binary operation	Value sent to the KNX bus (R) or value sent to the Modbus (W)
0x1234	0x00FF	Bit masking with fixed value	0x0034
0x01 (1 dec)	8	Bitshift right	0x10 (16 dec)
0x10 (16 dec)	8	Bitshift left	0x01 (1 dec)
0X1234	0x00FF	AND	0x0034
0X1234	0x00FF	OR	0X12FF
0xAA (10101010 bin)	0xFF	XOR	0X55 (01010101 bin)

Table 10: Example – Binary functions in Master Mode

Range conversion

A value is converted between 2 different scale ranges (KNX and Modbus side). A lower and an upper limit value are defined in each case.

Example with KNX (1 byte DPT 5.005 decimal value) and Modbus (2 byte unsigned decimal value).

KNX upper limit: 200 Modbus upper limit: 50000

KNX lower limit: 0 Modbus lower limit: 0

A KNX value of 100 would be converted to 25000 on the Modbus side.

Range limitation

The range limitation can be used to suppress the exceeding of a value range accepted by the bus participant.

Example:

A Modbus slave allows a value range of “1-100” and sends an error message in the event of an overflow or underflow.
If you want to prevent the error message (here in the direction KNX to Modbus (W)), you can enter a “1” for “Modbus: Lower limit value” and a “100” for “Modbus: Upper limit value”.
In this example, if a “0” or “101” is sent to the KNX receive object, the value sent to the Modbus is corrected to “1” or “100”.
The example can also be applied to the KNX bus in the other direction (R).

Note: The range limitation can always be set on both sides. In practice, the limit would be set on the side with the smaller value range.

Example:
KNX datapoint type 8.* and Modbus data type 4 byte signed decimal value:
“KNX: Upper limit value” would be used here.

4.5 Channels

Up to 200 channels can be activated with the "**Number of channels**" parameter.
A separate menu appears in the menu tree for each activated channel.

4.5.1 Channel 1 - 200

The following parameters are available for each channel:

ETS Text	Dynamic range **[Default value]**	Comment
General		
Channel	■ not active ■ **active**	Activation and setting of whether channel-specific parameters should be configured individually or via presets.
Channel/Object description	Any text (30 bytes allowed)	Free text input to describe the objects and the channel.
Modbus device		
Modbus device configuration	■ **individual** ■ Modbus device 1 : ■ Modbus device 10	Setting whether the Modbus device should be configured individually or whether the settings should be adopted from an existing device.
Modbus Slave address	0 ... 247 **[1]**	Setting of the slave address. **For "Master" operating mode and "Individual" device configuration.**
Register address	■ **First address „0"** ■ First address „1"	Setting whether the first register address is a "0" or a "1". **Only if "individual" is selected.**
Byte order - 2 Byte	■ LSB first (BA) ■ **MSB first (AB)**	Definition of the order in which the bits are to be sent.
Byte order - 4 Byte	■ LSB first (DCBA) ■ **MSB first (ABCD)** ■ LSB first / bytes swapped (CDAB) ■ MSB first / bytes swapped (BADC)	Definition of the order in which the bits are to be sent.
Device description	Display of the device description	Fixed text according to the device description of the Modbus device. **For Modbus device configuration "Modbus device x".**

ETS Text	Dynamic range **[Default value]**	Comment
Display device setting	☐	When activated (tick the box with a mouse click), the device settings are displayed. **For Modbus device configuration “Modbus device x”.**
Channel		
Channel configuration	■ **individual** ■ Preset 1 : ■ Preset 20	Setting whether a preset should be used or the channel should be configured individually.
Direction	■ **Modbus (R) to KNX** ■ KNX to Modbus (W) ■ Modbus / KNX bidirectional (R/W)	Setting the send/receive direction KNX <-> Modbus. ■ **Only if „not active (individual)“.** ■ **„ ... bidirectional (R/W)“ only for operating mode „Master“ .**
Description of preset	Description of the preset is displayed	Displayed text cannot be changed. **For channel preset „Preset x“.**
Display preset setting	☐	When activated (tick the box with a mouse click), the preset settings are displayed. **For channel preset „Preset x“.**
KNX (only available if channel preset is “not active (individual)”)		
KNX Datapoint type	■ **1 Bit DPT 1.001 On/Off** ■ 1 Bit DPT 1.003 Enable ■ 1 Byte DPT 5.001 Percent (0...100%) ■ 1 Byte DPT 5.005 Decimal value (0...255) ■ 1 Byte DPT 6.010 Value (-128...127) ■ 1 Byte DPT 17.001 Scene number ■ 2 Byte DPT 7.012 Current [mA] ■ 2 Byte DPT 7.013 Brightness [Lux] ■ 2 Byte DPT 9.001 Temperature [°C] ■ 2 Byte DPT 9.004 Brightness [Lux] ■ 2 Byte DPT 9.005 Speed [m/s] ■ 2 Byte DPT 9.007 Humidity [%] ■ 2 Byte DPT 9.008 air quality [ppm] ■ 2 Byte DPT 9.020 Voltage [mV] ■ 2 Byte DPT 9.021 Current [mA] ■ 2 Byte DPT 9.024 Power [kW] ■ 2 Byte DPT 9.025 Flow rate [l/h]	Setting the KNX datapoint type.

ETS Text	Dynamic range **[Default value]**	Comment
KNX datapoint type	■ 4 Byte DPT 12.1200 Volume [l] ■ 4 Byte DPT 12.1201 Volume [m³] ■ 4 Byte DPT 13.002 Flow rate [m³/h] ■ 4 Byte DPT 13.010 Active energy [Wh] ■ 4 Byte DPT 13.013 Active energy [kWh] ■ 4 Byte DPT 14.019 Current [A] ■ 4 Byte DPT 14.027 Electric potential[V] ■ 4 Byte DPT 14.033 Frequency [Hz] ■ 4 Byte DPT 14.056 Power [W] ■ 4 Byte DPT 14.065 Speed [m/s] ■ 4 Byte DPT 14.076 Volume [m³] ■ 4 Byte DPT 14.077 Flow rate [m³/s] ■ 4 Byte DPT 14.1200 Flow rate [m³/h] ■ 14 Byte DPT 16.000 String ■ 2 Byte DPT 7.* unsigned ■ 2 Byte DPT 8.* signed ■ 2 Byte DPT 9.* floating point value ■ 4 Byte DPT 12 * unsigned ■ 4 Byte DPT 13 * signed ■ 4 Byte DPT 14.* floating point value	Setting the KNX datapoint type.
KNX sending condition	■ read only ■ on change ■ cyclic ■ **on change and cyclic**	Setting whether and when a value is to be sent. **Only for direction „Modbus (R) to KNX“.**
Send on change of ...	Any value according to the set datapoint type	Setting as of which change the value should be sent again. **Only „on change“ and KNX data-point type 1/2/4 Byte.**
Send cyclically every ...	10 s – 24 h **[10 min]**	Setting the interval at which cyclical transmission should take place. **Only if “... cyclic” is selected.**
Modbus		
Modbus register address	Any value (6 bytes allowed) **[0x0001]**	Enter the corresponding address.
Modbus function code	■ **0x01 read coil status** ■ 0x02 read inputs status) ■ 0x03 read holding register) ■ 0x04 read input register)	Setting the function code. **With direction „Modbus (R) to KNX“ and KNX DPT „1 Bit“.**
Modbus function code	■ **0x03 read holding register** ■ 0x03 read holding register, multiread ■ 0x04 read input register	Setting the function code. **With direction “Modbus (R) to KNX” and KNX DPT “1/2/4 byte”.**

ETS Text	Dynamic range **[Default value]**	Comment
Modbus function code	■ **0x05 write single coil** ■ 0x06 / 0x16 write single or multiple register	Setting the function code. **With direction "KNX to Modbus (W)" and KNX DPT "1 bit".**
Modbus function code	0x06 / 0x16 write single or multiple register	Fixed setting, cannot be changed. **With direction "KNX to Modbus (W)" and KNX DPT "1/2/4 byte".**
Modbus data type	■ **1 Byte unsigned** ■ 1 Byte signed ■ 2 Byte unsigned ■ 2 Byte signed ■ 4 Byte unsigned ■ 4 Byte signed ■ 4 Byte floating point value	Setting of the Modbus data type. **Only visible when KNX DPT "1/2/4 byte" is also set.**
Output function with binary input object	■ **Bit in Register** ■ Send value	Setting of the function that is to be executed when a binary input object is received. **Only available if:** ■ **Direction „KNX to Modbus"** ■ **KNX DPT „1 Bit"** ■ **Function code „0x06/0x16 ...".**
Value for ON / OFF	0 ... 255 / 0 ... 65535 / 0 ... 4294967295 **[0]**	Setting the value to be sent. ■ **If "Send value" is selected** ■ **Value range depends on the Modbus data type.**
Bit position in register	0 ... 7 / 0 ... 15 / 0 ... 31 **[0]**	Setting the bit position. ■ **Value range depends on the Modbus data type.** ■ **Only for KNX DPT "1 Bit".**
Bit inverted	■ **not active** ■ active	Setting whether the outgoing Bit should be inverted. **Only with KNX DPT „1 Bit".**
Not a Number (invalid value)	■ **not active** ■ active	Activation of a NaN value. **Only with KNX DPT „1/2/4 Byte".**
Value	0xFF / 0xFFFF / 0xFFFFFFFF	Input of the NaN value. Number of "F" depending on the data type. **Only if "Not a Number (Invalid value)" is active.**
Error message for Modbus value = NaN value (invalid value)	■ **not active** ■ active	Activation of an error message.

ETS Text	Dynamic range **[Default value]**	Comment
String length	■ **2** ■ 4 ... ■ 14	Setting of how many characters (bytes) the string consists of. **For KNX DPT „14 Byte String“ and direction „Modbus (R) to KNX“.**
Priority	■ low ■ **medium** ■ high	Setting the priority (see below for description). **Only for KNX DPT “1 Bit”.**

Table 11: Settings – Channels

A text field for free labelling is available for each channel:

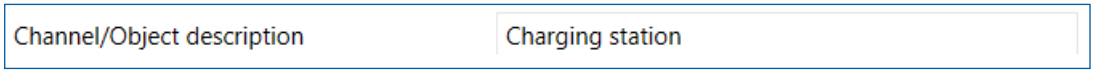

Figure 7: Text field – Channel/Object description

A text of up to 30 characters can be stored for the field.

The text entered in “**Channel/Object description**” appears both in the menu behind the corresponding channel and in the channel’s communication objects:

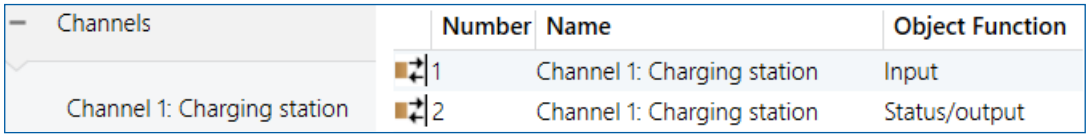

Figure 8: Presentation – Channel/Object description

Modbus device

A Modbus device can be configured separately via the “**Modbus device configuration**” parameter with the “**individual**” setting. Alternatively, a Modbus device already configured under “Presets: Modbus devices” can be used as a preset. In this case, only the device description assigned in the preset is displayed. In addition, the settings made there can be displayed by activating the check box.

Register address

The addressing is set here. The start address can be a “1” (1-based) or a “0” (0-based). The information on this can be found in the technical documentation for the device and must be set accordingly.

Byte order (2 Byte/4 Byte)

These parameters define the order in which the bytes are transferred.

LSB stands for “Least Significant Bit” - The sequence begins with the lowest significant bit.
MSB stands for “Most Significant Bit” - The sequence begins with the highest significant bit..

Channel
The channel can be configured individually or it is possible to use an already created “Channel preset”. This reduces the programming effort considerably if several channels are configured in exactly the same way.
The channel can be configured separately using the “**Channel configuration**” parameter with the setting “**(individual)**”. If an already configured preset is selected, only the description text assigned in the preset is displayed. In addition, the settings made there can be displayed by activating the check box.

KNX
The KNX datapoint type and the KNX sending conditions are defined in this parameter block.

Modbus
The Modbus specific settings for the preset are defined in this parameter block:

Modbus Function code
The MDT Modbus Gateway supports various Modbus function codes. The function codes define how the Modbus master accesses the Modbus register addresses of the slave.
The slave and the corresponding register address that is accessed must support the respective function code. Otherwise, the slave sends back an error code. In the ETS, the function code that can be used sensibly in each case is offered for configuration according to the KNX data type.

Special feature - Multi-channel read function
In some applications, it makes sense to request and evaluate read operations on the Modbus side in one Modbus frame. The MDT Modbus Gateway supports this when using the function code “0x03 read holding register, multiread”.
If this Modbus function code is selected in the KNX application, the gateway automatically forms a read sequence of up to 10 channels under the following conditions:

- All channels within the “multiread” function require the same slave address.
- The channels have consecutive register addresses (address “+1” for Modbus data types “1 Byte/2 Byte”. Address “+2” for Modbus data type “4 Byte”).
- The channels are configured with the function code “read holding register, multiread (0x03)”.
- The channels are configured with the direction “Modbus (R) to KNX”.
- The set KNX data types are numerical data types.

In addition to the automatic creation of a multi-channel read sequence, the user can determine the first channel of a desired sequence himself by activating the parameter “**Start channel for function code “multiread””**.

Output function with binary input object
With this function, it is possible to send a fixed output value (corresponding to the input values “1” and “0”) when a binary object is received. Alternatively, the bit position can be defined in the register.

Important: This function requires certain settings. These are:

- Direction „KNX to Modbus (W)“
- KNX Datapoint type „1 Bit DPT 1.x ...“
- Modbus function code „write single or multiple register (0x06 / 0x16 ...)

Bit position in register
If the KNX datapoint type is set to “1 Bit”, the position of the bit in the register can be determined here. The parameter therefore only appears if a Modbus function code “... register” is selected.

Bit inverted
The parameter appears if the KNX datapoint type is set to “1 Bit”. The bit can be displayed normally or inverted.

Not a Number (invalid value)
The NaN (Not a Number) function can be activated to filter out values sent by the Modbus slave in an error state.
If, for example, a participant always sends a “0xFFFF” in sleep mode (this would be an invalid value here), this value is not processed further and is not transmitted to the KNX output object.

Error message for Modbus value = NaN value (invalid value)
If activated, an error message would be issued via the diagnosis object in the event of an error.
See 4.2 General Modbus settings , Diagnosis object.

Priority
The differences in the settings are as follows:

- high: Execution of Modbus read operations in every cycle.
- medium: Execution of Modbus read operations after every 10th cycle.
- low: Execution of Modbus read operations after every 100th cycle.

The following table shows the associated communication objects:

Number	Name / Object function	Length	Usage
1	Channel 1: – Input		Input object (if “KNX to Modbus (W)” or “bidirectional”).
2	Channel 1: – Output		Output object (if Modbus (R) to KNX)
2	Channel 1: – Status/Output		Status/output object (if “bidirectional”)

Table 12: Communication objects – Channels

4.5.2 Mathematics

With the “KNX datapoint type - 1/2/4 Byte” setting, a “Mathematics” parameter block appears. Various mathematical functions can be realised there. The following table shows the available settings:

ETS Text	Dynamic range **[Default value]**	Comment
Function	■ **not active** ■ Scaling ■ Arithmetics ■ Binary function ■ Range conversion ■ Range limitation	Setting a mathematical function. **Binary function only with DPT selection “KNX 5.005 and Modbus 1 Byte unsigned”**
Scaling		
Scaling	■ x 100,000 ■ x 10,000 : ■ **x 1** ■ x 0.1 : ■ x 0.00001	Determination of the multiplier for scaling. **For direction “Modbus (R) to KNX” and “KNX to Modbus (W)”.**
Scaling	■ Modbus (W) x 100,000 / Modbus (R) x 0.00001 ■ Modbus (W) x 10,000 / Modbus (R) x 0.0001 : ■ **Modbus (W) x 1 / Modbus (R) x 1** ■ Modbus (W) x 0.1 / Modbus (R) x 10 : ■ Modbus (W) x 0.00001 / Modbus (R) x 100,000	Determination of the multiplier for scaling. **For direction “Modbus / KNX bidirectional (R/W)”.**
Arithmetics		
Arithmetics	■ Addition: channel value + value 1 ■ Subtraction: channel value - value 1 ■ Subtraction: value 1 - channel value ■ **Multiplication: channel value x value 1** ■ Multiplication with Offset: value 1 + channel value x value 2 ■ Multiplication: channel value x 10^value 1 ■ Multiplication: 10^channel value x value 1 ■ Division: channel value / value 1 ■ Division: value 1 / channel value	Definition of the calculation type and the terms involved. **Only for direction “Modbus (R) to KNX” and “KNX to Modbus (W)”.**

ETS Text	Dynamic range **[Default value]**	Comment
Arithmetics	■ Modbus (W): channel value + value 1; Modbus (R): channel value - value 1 ■ Modbus (W): channel value - value 1; Modbus (R): channel value + value 1 ■ Modbus (W): value 1 - channel value; Modbus (R): value 1 + channel value ■ **Modbus (W): channel value x value 1; Modbus (R): channel value / value 1** ■ Modbus (W): value 1 + channel value x value 2; Modbus (R): (channel value - value 1) / value 2 ■ Modbus (W): 10^value 1 x channel value; Modbus (R): channel value / 10^value 1 ■ Modbus (W): 10^channel value x value 1; Modbus (R): value 1 / 10^channel value ■ Modbus (W): channel value / value 1; Modbus (R): channel value x value 1 ■ Modbus (W): value 1 / channel value; Modbus (R): channel value x value 1	Definition of the calculation type and the terms involved. **Only for direction „Modbus / KNX bidirectional (R/W)“.**
Operation with ...	■ **value 1** ■ value 1 is output value from other channel	Setting whether the value required for the operation is defined in this channel or whether the value of another channel should be used.
Value 1	-10000000000000 ... 10000000000000 **[0]**	Definition of value 1.
Value 2	-10000000000000 ... 10000000000000 **[0]**	Definition of value 2. **Only if selected with “ ... value 2”**
Binary function (only available under certain conditions - see description after the table)		
Binary function	■ **Bit masking with fixed value** ■ Bitshift left ■ Bitshift right ■ AND ■ OR ■ XOR	Setting the corresponding function.

ETS Text	Dynamic range **[Default value]**	Comment
Number system	■ Decimal ■ Hex	Setting the number system for the fixed value. **Only for "Bit masking with fixed value", „AND", „OR", „XOR".**
Value	0 ... 255 **[0]**	Definition of the fixed value. **Only if "Decimal" is selected.**
Value	free Input (4 Bytes allowed) **[0xFF]**	Definition of the fixed values. **Only if "Hex" is selected.**
Bitshift	0 ... 8 **[0]**	Setting for how many positions to be shifted to the left or right. **For "Bitshift left/right".**
Range conversion		
KNX: Lower limit value	Any value according to the set KNX datapoint type	Definition of the lower limit value for the range.
KNX: Upper limit value	Any value according to the set KNX datapoint type	Definition of the upper limit value for the range.
Modbus: Lower limit value	Any value according to the set Modbus data type	Definition of the lower limit value for the range.
Modbus: Upper limit value	Any value according to the set Modbus data type	Definition of the upper limit value for the range.
Range limitation		
KNX: Lower limit value	Any value according to the set KNX datapoint type	Definition of the lower limit value for the range.
KNX: Upper limit value	Any value according to the set KNX datapoint type	Definition of the upper limit value for the range.
Modbus: Lower limit value	Any value according to the set Modbus data type	Definition of the lower limit value for the range.
Modbus: Upper limit value	Any value according to the set Modbus data type	Definition of the upper limit value for the range.

Table 13: Settings – Mathematics

Scaling

The function can be used to realise a decimal point shift.

Arithmetics

Operations can be implemented here according to basic arithmetic operations. An extensive selection is available in the drop-down menu.

With „**Value 1 is output value of other channel**“, this channel is defined with the following “Channel selection” parameter.

Binary function

The binary functions enable simple bit-by-bit binary calculation functions for the Modbus value read in or to be sent.

Important: The binary function is only available with the following selection:

- KNX DPT 1/2/4 byte unsigned
- Modbus data type 1/2/4 byte unsigned
- Both data types must have the same length.
- Direction: “Modbus (R) to KNX” or “KNX to Modbus (W)”

The following applies in **master mode** of the Modbus gateway:
If the “Modbus (R) to KNX” direction is set, the data sent by the Modbus participant is linked bit by bit with the fixed value entered under “Value” and written to the KNX send object.
If the “KNX to Modbus (W)” direction is set, the data in the KNX receive object is linked bit by bit with the fixed value entered under “Value” and sent to the Modbus.

The following applies in **slave mode** of the Modbus gateway:
If the “KNX to Modbus (W)” direction is set, the data sent by the Modbus master is linked bit by bit with the fixed value entered under “Value” and written to the KNX send object.
If the “Modbus (R) to KNX” direction is set, the data in the KNX receive object is linked bit by bit with the fixed value entered under “Value” and sent to the Modbus master.

The “Bit masking”, “AND”, “OR” and “XOR” operations involve a bit-by-bit linking of the data.

The following table shows examples of the master mode:

Value from Modbus (R) or value from KNX receiver object (W)	Entered fixed value	Binary operation	Value sent to the KNX bus (R) or value sent to the Modbus (W)
0x1234	0x00FF	Bit masking with fixed value	0x0034
0x01 (1 dec)	8	Bitshift right	0x10 (16 dec)
0x10 (16 dec)	8	Bitshift left	0x01 (1 dec)
0X1234	0x00FF	AND	0x0034
0X1234	0x00FF	OR	0X12FF
0xAA (10101010 bin)	0xFF	XOR	0X55 (01010101 bin)

Table 14: Example – Binary functions in Master Mode

Range conversion

A value is converted between 2 different scale ranges (KNX and Modbus side). A lower and an upper limit value are defined in each case.

Example with KNX (1 byte DPT 5.005 decimal value) and Modbus (2 byte unsigned decimal value).

KNX upper limit: 200 Modbus upper limit: 50000

KNX lower limit: 0 Modbus lower limit: 0

A KNX value of 100 would be converted to 25000 on the Modbus side.

Range limitation

The range limitation can be used to suppress the exceeding of a value range accepted by the bus participant.

Example:

A Modbus slave allows a value range of “1-100” and sends an error message in the event of an overflow or underflow.
If you want to prevent the error message (here in the direction KNX to Modbus (W)), you can enter a “1” for “Modbus: Lower limit value” and a “100” for “Modbus: Upper limit value”.
In this example, if a “0” or “101” is sent to the KNX receive object, the value sent to the Modbus is corrected to “1” or “100”.
The example can also be applied to the KNX bus in the other direction (R).

Note: The range limitation can always be set on both sides. In practice, the limit would be set on the side with the smaller value range.

Example:
KNX datapoint type 8.* and Modbus data type 4 byte signed decimal value:
“KNX: Upper limit value” would be used here.

4.6 Comparators

Up to 10 comparators can be activated with the "**Number of comparators**" parameter. A separate menu appears in the menu tree for each activated comparator.

The comparator compares KNX objects with comparison values according to mathematical operations.

4.6.1 Comparator 1 - 10

The following parameters are available for each comparator:

ETS Text	Dynamic range **[Default value]**	Comment
Comparator 1 - 10	■ **not active** ■ active	Activation/deactivation of the respective comparator.
Description	any text (30 bytes allowed)	Free text input to describe the comparator.
Comparison channel	1 ... 200 **[1]**	Selection of the channel to be compared with.
KNX sending condition	■ read only ■ on change ■ cyclic ■ **on change and cyclic**	Setting whether and when a value is to be sent.
Send cyclically every ...	10 s – 24 h **[10 min]**	Setting the interval at which an output value is to be sent. **Only if "... cyclic" is selected.**
Output objects	■ **common** ■ single	Setting whether to send to a common output object or to separate objects (per comparator).
Datapoint type - Output	■ **1 Bit DPT 1.002 Bool** ■ 1 Byte DPT 5.001 Percent (0...100%) ■ 1 Byte DPT 5.005 Decimal unsigned (0...255) ■ 1 Byte DPT 6.010 Decimal signed (-128...127) ■ 1 Byte DPT 17.001 Scene number ■ 2 Byte DPT 7.* unsigned ■ 2 Byte DPT 8.* signed ■ 2 Byte DPT 9.* floating point value ■ 4 Byte DPT 12 * unsigned ■ 4 Byte DPT 13 * signed ■ 4 Byte DPT 14.* floating point value	Selection of the datapoint type that is to be sent when the operation is fulfilled/not fulfilled.

ETS Text	Dynamic range **[Default value]**	Comment
Comparison 1 – 4		
Hysteresis/Tolerance range for all comparisons	-3,4E+38 ... 3,4E+38 **[0]**	Definition of a hysteresis or a tolerance range.
Comparison 1 – 4	■ not active ■ **equal (=)** ■ unequal (!=) ■ smaller (<) ■ greater (>)	Setting the comparison operation.
Comparison value	-3,4E+38 ... 3,4E+38 **[0]**	Determination of the comparison value.
Output value	Any value according to the set datapoint type	Setting of the value to be sent when the operation is fulfilled.
Output Scene number	1 – 64 **[1]**	Setting the scene that is to be sent when the operation is fulfilled. **Only for DPT "Scene number".**
Value if no comparison is fulfilled	Any value according to the set datapoint type	Setting of the value to be sent if the operation is not fulfilled.
Scene number if no comparison is fulfilled	1 – 64 **[1]**	Setting the scene that is to be sent when the operation is not fulfilled. **Only for DPT "Scene number".**

Table 15: Settings – Comparator

A text field for free labelling is available for each comparator:

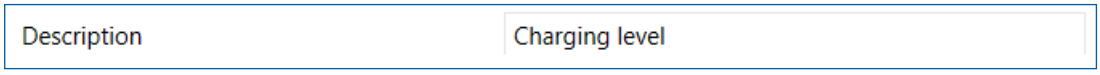

Figure 9: Text field – Description

A text of up to 30 characters can be stored for the field.

The text entered in "**Description**" appears both in the menu behind the corresponding comparator and in the comparator's communication object:

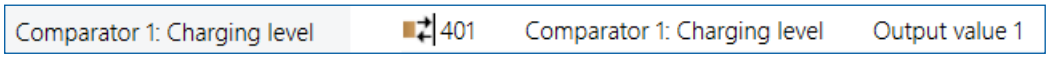

Figure 10: Presentation – Description

The “**Comparison channel**” is used to select the channel with which the comparison is to be made.

The **KNX sending condition** is also defined. The output object can be sent either when the comparison condition changes and/or also cyclically.

With the “**Output objects - common**” setting, only one output object appears. All (up to 4) comparisons send to this object.
Important: The 4 comparisons are OR-linked. If one of the 4 comparisons is fulfilled, another comparison cannot change the output object.

With the “**Output objects - single**” setting, a separate output object appears for each activated “Comparison”.

The set “**Datapoint type - Output**” determines the output object and applies to all activated “Comparisons”.

“**Hysteresis/Tolerance range for all comparisons**”

The hysteresis is intended for real, noisy measured values in order to prevent the output from “fluttering”.

The dynamic behaviour of the comparators with hysteresis / tolerance is as follows:

Comparison → **“equal (=)“**:

- The output value is set as soon as the channel value is within “Comparison value - tolerance” to “Comparison value + tolerance”.
- The output value is reset as soon as the channel value is outside “Comparison value - tolerance” to “Comparison value + tolerance”.

Comparison → **“unequal (!=)“:**

- The output value is set as soon as the channel value is outside “Comparison value - tolerance” to “Comparison value + tolerance”.
- The output value is reset as soon as the channel value is within “Comparison value - tolerance” to “Comparison value + tolerance”.

Comparison → **“smaller (<)“:**

- The output value is set as soon as the channel value falls below the comparison value and the output value was previously reset.
- The output value is reset as soon as the channel value exceeds the “Comparison value + tolerance”.

Comparison → **“greater (>)“:**

- The output value is set as soon as the channel value exceeds the comparison value and the output value was previously reset.
- The output value is reset as soon as the channel value falls below the “Comparison value - tolerance”.

Up to 4 independent comparisons are now available. Each comparison can be activated and configured individually.

The operation (comparison condition) is defined with the "**Comparison x**" parameter.

The "**Comparison value**" defines the value that acts as the basis for the comparison.

The "**Output value**" defines the value that is sent when the comparison condition is fulfilled. The type set under "Datapoint type - Output" applies here.

Under "**Value if no comparison is fulfilled**", a value can be set that is to be sent if none of the comparisons are fulfilled.

The following table shows the associated communication objects:

Number	Name / Object function	Length	Usage
401	Comparator 1 – Common output value		Sending the value (if common). DPT according to parameter setting.
401	Comparator 1 – Output value 1		Sending the value of comparison 1. DPT according to parameter setting.
402	Comparator 1 – Output value 2		Sending the value of comparison 2. DPT according to parameter setting.
403	Comparator 1 – Output value 3		Sending the value of comparison 3. DPT according to parameter setting.
404	Comparator 1 – Output value 4		Sending the value of comparison 4. DPT according to parameter setting.

Table 16: Communication objects – Comparators

4.7 Message texts

Up to 10 message texts can be activated with the “Number of message texts” parameter.
A separate menu appears in the menu tree for each activated message text.

The function is similar to that of the comparator. Message texts are sent here instead of values.

4.7.1 Message text 1 - 10

The following table shows the available settings:

ETS Text	Dynamic range **[Default value]**	Comment
Message text 1 - 10	■ **not active** ■ active	Activation/deactivation of the respective message text.
Description	any text (30 bytes allowed)	Free text input to describe the message text.
Comparison channel	1 ... 200 **[1]**	Selection of the channel to be compared with.
KNX sending condition	■ read only ■ on change ■ cyclic ■ **on change and cyclic**	Setting whether and when a value is to be sent.
Send cyclically every ...	10 s – 24 h **[10 min]**	Setting the interval at which a message text is to be sent. **Only if “... cyclic” is selected.**
Message if no comparison is fulfilled	■ not active ■ **active**	Activation of an additional message text.
Message text	Free text input (14 bytes allowed)	Enter a free text. **Only if “Message if no comparison fulfilled” is active.**

Table 17: Settings – Message texts

A text field for free labelling is available for each message text:

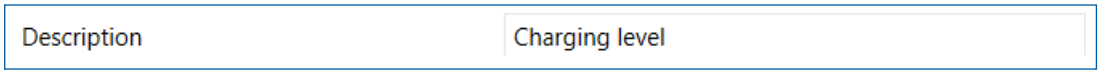

Figure 11: Text field – Description

A text of up to 30 characters can be stored for the field.

The text entered in “**Description**” appears both in the menu after the corresponding message text and in the communication object of the message text:

Figure 12: Presentation – Description

The “**Comparison channel**” is used to select the channel with which the comparison is to be made.

The **KNX sending condition** is also defined. The output object can be sent either when the comparison condition changes and/or also cyclically.

In the following table, up to 10 different conditions can be defined in the form of comparisons with the output of an associated text.

The following settings are available for a “comparison”:

- not active
- **equal (=)**
- unequal (!=)
- smaller (<)
- greater (>)

Message text	Comparison	Comparison value	Text
1	equal (=)	0	Equal
2	unequal (!=)	0	Unequal
3	smaller (<)	0	Smaller
4	greater (>)	0	Greater
5	not active		
6	not active		
7	not active		
8	not active		
9	not active		
10	not active		

Table 18: Definition of message texts via comparisons

As soon as a value is received, a comparison is carried out. Whenever one of the conditions is met, the text defined for it is sent.

If the “**Message if no comparison is fulfilled**” parameter is activated, an additional text can be defined. This is sent if none of the comparisons defined in the table are fulfilled.

The following table shows the associated communication objects:

Number	Name / Object function	Length	Usage
441	Message text 1 – Output value	14 Byte	Sending a text
+ 1	**next message text**		

Table 19: Communication objects – Message texts

5 Index

5.1 List of figures

5.2 List of tables

6 Appendix

6.1 Statutory requirements

The devices described above must not be used in conjunction with devices which directly or indirectly serve human, health, or life-safety purposes. Furthermore, the devices described must not be used if their use may cause danger to people, animals, or property.
Do not leave the packaging material carelessly lying around. Plastic foils/ bags etc. can become a dangerous toy for children.

6.2 Disposal

Do not dispose of the old devices in the household waste. The device contains electrical components that must be disposed of as electronic waste. The housing is made of recyclable plastic.

6.3 Assembly

Danger to life from electric current!

The device may only be installed and connected by qualified electricians. Observe the country-specific regulations and the applicable KNX guidelines

The devices are approved for operation in the European Union and in the United Kingdom. The products are respectively marked with the CE and UKCA symbols.

Use in the USA and Canada is prohibited!

Before starting work on the device, always de-energise it using the upstream fuses. After installation, all live terminals and connections must be completely covered by the switch panel cover to prevent accidental contact. It must not be possible to open the switch panel cover without tools.

6.4 History

V1.0	First Version of Technical Manual	DB V1.0 04/2024