

# **Modbus**

## **USER MANUAL**

Version: 1.1

Date: 20/Apr/2026

## Contents

1.	Modbus .....	4
	General .....	4
	Modbus Device Templates (1 ÷ 8) .....	5
	Channel Templates (1 ÷ 16) .....	5
	Channels (1 ÷ 25) .....	6

VERSION	DATE	CHANGES
1.0	23/Jun/2025	-
1.1	20/Apr/2026	Mask parameter added

### Translation of the Original Instructions


Any information in this manual is subject to change without notice.


This handbook can be download freely from the website:  
[www.eelectron.com](http://www.eelectron.com)

#### Exclusion of liability:

Despite checking that the contents of this document match the hardware and software, deviations cannot be completely excluded. We therefore cannot accept any liability for this.

Any necessary corrections will be incorporated into newer versions of this manual.

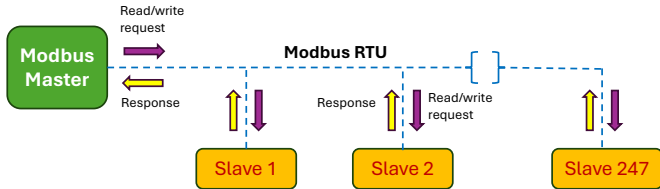
Symbol for relevant information 

Symbol for warning 



# 1. Modbus

Modbus is a serial communication protocol used to transmit information between electronic devices. Modbus operates on a client-server architecture: the device requesting the information is called Modbus Master and the devices providing the information are Modbus Slaves. In a standard Modbus network, there is only one master and up to 247 slaves.

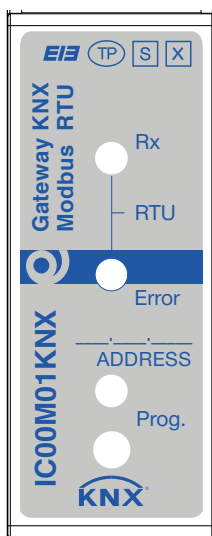


Main features:

- data organization
- simplicity
- reliability
- scalability

Data are organized into four categories: discrete inputs, coils, input registers, and holding registers, allowing very easy management of a variety of data types.

LED	FUNCTION	
RTU	Rx	Green flashing: Receiving data on RS485
	Error	Red flashing: RTU error
		Red off: No RTU error present
Prog	Button	Press to activate program mode
	LED	Steady red: device ready for programming



## General

Communication objects involved:

"<Modbus> Alarm"	1 Bit	CRT
"<Modbus> Alarm Text"	14 Bytes	CRT

KNX PARAMETER	SETTINGS
<b>Modbus type</b>	modbus master modbus slave
This parameter defines whether the device is used as master or device.	
<b>Requests timeout</b>	100/250/500 ms 1 s
Only if device is master, this parameter defines the waiting time for a response from slave. Communication alarm is set if no response received.	
<b>Time between cycles</b>	500 ms 1/2/5/8/10/15/20/25/30/45 s 1/2/3/4/5 m
Only if device is master, this parameter defines the delay between two consecutive cycles of requests.	
<b>Time between requests</b>	minimal 10/25/50/100/150/250/500 ms
Only if device is master, this parameter defines the delay between two consecutive requests.	
<b>Own address</b>	1 ÷ 247
Only if device is slave, this parameter defines the physical Modbus address of the device.	
<b>Baudrate</b>	1200/2400/4800/9600/ 14400/19200/28800/38400/ 56000/57600/115200
This parameter defines the speed at which data is transmitted over the Modbus RTU serial bus.	
<b>Parity</b>	none/even/odd
This parameter defines the error control method used in Modbus RTU serial communication.	
<b>Stop bits</b>	1 / 2
This parameter defines the number of bits that signal the end of each data frame in Modbus RTU serial communication, ensuring correct synchronization between devices.	
<b>Alarm object</b>	disabled/enabled
This parameter is used to enable object '<Modbus> Alarm' to signal presence of any alarm on the Modbus network.	
<b>Alarm telegram</b>	telegram "0"/telegram "1"
This parameter defines the value of the object '<Modbus> Alarm' in case of alarm.	
<b>Alarm text</b>	disabled/enabled
This parameter is used to enable object '<Modbus> Alarm Text' to describe the alarm detected on the Modbus network.	
<b>Requests timeout [min]</b>	1 ÷ 255
Only if device is slave, this parameter defines the time within which a request from master is expected before activating alarm.	

**Modbus Device Templates (1 ÷ 8)**

Use templates to repeat the same setting in more of one channel; each template can be loaded in any channel for faster setup.

KNX PARAMETER	SETTINGS
<b>Slave name</b>	Free field (alphanumerical)
This parameter defines the name of the module; the name can be used to rapidly identify the slave.	
<b>Slave address</b>	1 ÷ 247
Only if device is master, this parameter defines the physical Modbus address of the slave device.	
<b>Register address</b>	first address "0" first address "1"
This parameter defines whether the gateway interprets the Modbus register addresses of the slave device as zero-based (0) or one-based (1).	
<b>Byte order - 2 Bytes</b>	LSB first (BA) MSB first (AB)
This parameter specifies how the gateway interprets the sequence of the two bytes (least significant or most significant byte first).	
<b>Byte order - 4 Bytes</b>	LSB first (DCBA) MSB first (ABCD) LSB first / bytes swapped (CDAB) MSB first / bytes swapped (BADC)
This parameter specifies how the gateway interprets the sequence of the four bytes (least significant or most significant byte first with or without swap).	

**Channel Templates (1 ÷ 16)**

KNX PARAMETER	SETTINGS
<b>Template name</b>	Free field (alphanumerical)
This parameter defines the name of the module; the name can be used to rapidly identify the template.	
<b>Direction</b>	KNX → modbus KNX ← modbus KNX ↔ modbus
The "direction" parameter defines the data flow for an object, specifying if information goes from KNX to Modbus, from Modbus to KNX, or in both directions.	
<b>DPT type</b>	1 bit 1 byte signed 1 byte unsigned 2 bytes signed 2 bytes unsigned 2 bytes float 4 bytes signed 4 bytes unsigned 4 bytes float 4 bytes float 14 bytes string
This parameter defines the DPT of the object. The telegram can be: <ul style="list-style-type: none"> <li>• 1 bit</li> <li>• 1 byte (signed, unsigned)</li> <li>• 2 bytes (signed, unsigned, float)</li> <li>• 4 bytes (signed, unsigned, float)</li> <li>• 14 bytes string</li> </ul>	
<b>On variation</b>	no / yes
This parameter defines whether to send the KNX object telegram when its value changes.	

<b>Variation value</b>	1 ÷ 255 1 ÷ 65535 0,001 ÷ 670760 1 ÷ 4294967295 0,001 ÷ 3,4E+38
Only if parameter "On variation" is set to "yes", this parameter defines the minimum value of change on the KNX object "<Modbus> Object" to be sent on the bus.	
<b>Cyclic</b>	no cyclic sending 1/2/5/10/15/30/45m 1/2/3/4/5/6/8/12h
This parameter defines the time interval to send cyclically on the BUS the object "<Modbus> Object".	
<b>Modbus data type</b>	1 bit 1 byte signed 1 byte unsigned 2 bytes signed 2 bytes unsigned 4 bytes signed 4 bytes unsigned 4 bytes float
This parameter defines the type of Modbus data.	
<b>Mask is available only in case value of "DPT type" is 1 ÷ 4 bytes and value of "Modbus data type" is 1 / 2 byte unsigned. The register is always converted in MSB format before applying the mask.</b>	
<b>Mask</b>	disabled / enabled
When enabled, this parameter allows you to select the bit mask to be applied within the Modbus register.	
<b>Mask value (MSB)</b>	bitmask (when 0 bit disabled, when 1 bit enabled)
This parameter defines the bitmask sequence.	
<b>Function code</b>	01 - read coil 02 - read input 03 - read holding registers 04 - read internal registers 05 - write coil 06 - write single or multiple registers
This parameter defines the Modbus operation to execute.	
<b>Bit inverted</b>	no / yes
Only if parameter "DPT type" is set to "1 bit", this parameter defines whether to invert bit when it is transferred from KNX to Modbus and viceversa.	
<b>Arithmetics</b>	no / yes
Only if parameter "DPT type" is different from "1 bit" or "14 bytes string", this parameter defines whether a mathematical operation is applied when data is transferred from KNX to Modbus and viceversa.	
<b>Offset</b>	-3,4E+38 ÷ 3,4E+38
Only when "Arithmetics" is set to "yes", this parameter defines a value to add or subtract when data is transferred from KNX to Modbus and viceversa.	
<b>Factor</b>	-3,4E+38 ÷ 3,4E+38
Only when "Arithmetics" is set to "yes", this parameter defines a value to multiply when data is transferred from KNX to Modbus and viceversa.	
<b>Round</b>	Unchecked / Checked
Only when "Arithmetics" is set to "yes", this parameter determines whether the result must be rounded.	
<b>Round type</b>	up / down
Only if parameter "Round" is set to "yes", when automatic round is not possible, this parameter defines direction of rounding.	
<b>Round value</b>	0,001 ÷ 3,4E+38
Only if parameter "Round" is set to "yes", this parameter defines the value of which the result must be a multiple of.	

<b>String length</b>	2/4/6/8/10/12/14
Only if parameter "DPT type" set to "14 bytes string", this parameter defines the number of characters in the string.	

## Channels

Communication objects involved:

"<Modbus x> Object"	1 Bit ÷ 14 Bytes	CW/CWTU CRT CRWT/CRWTU
"<Modbus x> Alarm"	1 Bit	CRT

The ETS parameters are the same described in the templates. Additional parameters are listed below.

KNX PARAMETER	SETTINGS
<b>Channel</b>	disabled/enabled
This parameter enable or disable the channel.	
<b>Channel name</b>	Free field (alphanumeric)
This parameter defines the name of the channel; the name can be used to rapidly identify the configuration.	
<b>Device modbus</b>	independent template 1 ÷ 8
This parameter allows configuration of the device or use of pre-set template.	
<b>Channel modbus</b>	independent template 1 ÷ 16
This parameter allows to select the way each channel can be managed: independent or using a preset template.	
<b>Address</b>	0 ÷ 65535
This parameter defines the physical address of modbus register.	
<b>Read KNX value at power ON</b>	disabled/enabled
When this parameter is set on "enabled" the value at power up of object is recovered through an operation of "read request" on KNX bus.	
<b>Alarm object</b>	disabled/enabled
This parameter is used to enable object '<Modbus x> Alarm' to signal presence of alarm on the specific Modbus channel.	
<b>Alarm telegram</b>	telegram "0"/telegram "1"
This parameter defines the value of the object '<Modbus x> Alarm' in case of alarm.	