



Modbus RTU communication protocol in THT/THT2 sensors

Complete protocol description



Modbus RTU in THT

Datasheet

Created: 6.4.2009

Translated: 20.10 2022 14:25

Total pages: 12

© 2022 Papouch s.r.o.

Papouch s.r.o.

Address:

**Strasnicka 3164
102 00 Prague 10
Czech Republic**

Tel:

+420 267 314 267

Internet:

en.papouch.com

E-mail:

info@papouch.com



TABLE OF CONTENTS

Description	4
Basic communication parameters	4
Firmware change log	4
Switching the protocols	5
Spinel → Modbus RTU	5
Allow configuration	5
Switching	5
MODBUS RTU → Spinel	5
Memory layout	6
List of function codes	6
Identification of the Device	6
Holding Register	6
Input Register	7

DESCRIPTION

This document describes Modbus RTU communication protocol embedded into an intelligent sensor for temperature and humidity, THT, [THT2](#) and [THT2i](#). Documentation of the sensors' hardware and a description of their functions is available at papouch.com.

Basic communication parameters

Communication line.....	RS485
Communication speed	from 1.2 kBd to 115.2 kBd (default: 9.6 kBd)
Data bits.....	8
Parity.....	no parity
Stop bits.....	1
Delay before response is sent.....	2 ms ¹
Default address.....	0x31
Default protocol.....	Spinel

Firmware change log

Version 04

Modbus RTU protocol added. THT is capable of communication via one protocol at once. Active protocol can be set by special instruction described below. (Default protocol is Spinel described in other document.)

¹ Delay is added in order to allow the device to switch the direction of RS485 communication.

SWITCHING THE PROTOCOLS

Default communication protocol is Spinel. For switching the protocol to Modbus RTU, following Spinel instruction must be sent to the device. For the initial configuration of the address, etc., we recommend using the [ModbusConfigurator](#) (for Windows).

Spinel → Modbus RTU

Allow configuration

This instruction allows service instructions. It has to forego the switching instruction. Instruction cannot be used with universal or broadcast address.

Request:

Instruction code: E4H

Response:

Acknowledge code: ACK 00H

Examples:

Request:
2AH, 61H, 00H, 05H, 31H, 02H, E4H, 58H, 0DH
Allow configuration.
Response:
2AH, 61H, 00H, 05H, 31H, 02H, 00H, 3CH, 0DH
Instruction acknowledged.

Switching

Switching the protocol is executed by special Spinel 97 instruction. Address of a specific module has to be used (the instruction cannot be used with universal or broadcast address.) Allow configuration instruction must forego this instruction.

Request:

Instruction code: EDH

Response:

Acknowledge code: ACK 00H

Examples:

Request:
2AH, 61H, 00H, 06H, 31H, 02H, EDH, 02H, 4CH, 0DH
Switching instruction: Spinel to Modbus RTU.
Response:
2AH, 61H, 00H, 05H, 31H, 02H, 00H, 3CH, 0DH
Instruction acknowledged. From this point on THT communicates via Modbus RTU.

MODBUS RTU → Spinel

Method of switching is documented on page 7 of this document.

MEMORY LAYOUT

For the initial configuration of the address, etc., we recommend using the [ModbusConfigurator](#).

List of function codes

The device allows access to its memory – depending on the type of registry – using the following instructions:

- 0x03Read Holding register
- 0x04Read Input Registers
- 0x06Write Single Register
- 0x10Write Multiple registers
- 0x11Identification

Identification of the Device

Reading of the device identification string (Report slave ID).

Function codes:

0x11 – Report slave ID

Parameters:

Number of bytes	1 Byte	According to the string
ID	1 Byte	ID is the same as the device address
RI	1 Byte	Run Indicator – here always 0xFF (on)
Data	N Bytes	String the same as in Spinel protocol. For example: <i>THT; v0301.01.01; f66 97</i>

Holding Register

Address	Access	Function	Name
0 ²	write	0x06	Enable Configuration Writing the 0x00FF value to this memory location must precede all instructions that write in the addresses of 0 to 5 in the holding register. It is used to protect against accidental configuration changes. The <i>Enable Configuration</i> instruction must not be written using the 0x10 function code together with other parameters!
1	read, write	0x03, 0x06, 0x10	Address (ID)³ A unique address of the device in the Modbus protocol. A number ranging from 1 to 247 is expected. The address is unique to the Modbus protocol. <i>The default address is 0x0031.</i>

² You can see register numbered from 0 or from 1 as the first register has an address of 0.

³ Writing to this memory location must be preceded by writing the 0x00FF value into the address of 0 in the Enable Configuration position. This prevents accidental configuration changes. It is not allowed to write Enable Configuration using Multiply write simultaneously with other parameters.

Address	Access	Function	Name																					
2	read, write	0x03, 0x06, 0x10	Communication speed³ Supported communication speeds are: ⁴ 1 200 Bd.....0x0003 2 400 Bd.....0x0004 4 800 Bd.....0x0005 9 600 Bd.....0x0006 (default) 19 200 Bd.....0x0007 38 400 Bd.....0x0008 57 600 Bd.....0x0009 115 200 Bd.....0x000A																					
3	read, write	0x03, 0x06, 0x10	Data word format³ Data word is always eight-bit. <table border="1"> <thead> <tr> <th>Code</th> <th>Parity</th> <th>Stop bits</th> </tr> </thead> <tbody> <tr> <td>0x0000 (default)</td> <td>none (N)</td> <td>1</td> </tr> <tr> <td>0x0001</td> <td>even (E)</td> <td>1</td> </tr> <tr> <td>0x0002</td> <td>odd (O)</td> <td>1</td> </tr> <tr> <td>0x0003</td> <td>none (N)</td> <td>2</td> </tr> <tr> <td>0x0004</td> <td>even (E)</td> <td>2</td> </tr> <tr> <td>0x0005</td> <td>odd (O)</td> <td>2</td> </tr> </tbody> </table>	Code	Parity	Stop bits	0x0000 (default)	none (N)	1	0x0001	even (E)	1	0x0002	odd (O)	1	0x0003	none (N)	2	0x0004	even (E)	2	0x0005	odd (O)	2
Code	Parity	Stop bits																						
0x0000 (default)	none (N)	1																						
0x0001	even (E)	1																						
0x0002	odd (O)	1																						
0x0003	none (N)	2																						
0x0004	even (E)	2																						
0x0005	odd (O)	2																						
4	read, write	0x03, 0x06, 0x10	Packet end distinction³ Configures what delay between bytes is considered an end of the packet. Delay is entered as bytes count. You can enter 4 to 100 bytes. <i>Default value is 10.</i>																					
5	read, write	0x03, 0x06, 0x10	Communication protocol³ This instruction allows THT to be switched to Spinel protocol. After sending the response, THT switches to Spinel protocol and communicates with it from this point on. Code for <i>Spinel</i> : 0x0001 (default) Code for Modbus RTU: 0x0002																					

Input Register

Address	Access	Function	Name
<i>Values and states sorted by channel order</i>			
0 ²	read	0x04	Temperature status 0x0000 ... value is valid Other ... value is invalid
1	read	0x04	Current temperature – integer Current temperature as signed integer ⁵ (16bit value) multiplied by a factor of ten. The final temperature has a resolution of 0.1°C.

⁴ Lower or higher communication speeds can be added on request.

⁵ Negative numbers are a two's complement. For a detailed explanation of this method, see, for example, the [Wikipedia entry on Two's complement](#). In principle, all you need to do to convert is to add a condition along these lines to the code: if (value > 32767) value = value - 65536;

Example: Temperature -13.8 °C is represented as the number -138 (decadically), which is FF76H.

Address	Access	Function	Name
2, 3	read	0x04	Current temperature – float Current temperature as 32 bit float number according to IEEE 754.
4	read	0x04	Humidity status 0x0000 ... value is valid Other ... value is invalid
5	read	0x04	Current humidity – integer Current humidity as signed integer (16bit value) multiplied by a factor of ten.
6, 7	read	0x04	Current humidity – float Current temperature as 32 bit float number according to IEEE 754.
8	read	0x04	Dew point status 0x0000 ... value is valid Other ... value is invalid
9	read	0x04	Current dew point – integer Current dew point as signed integer (16bit value) multiplied by a factor of ten.
10, 11	read	0x04	Current dew point – float Current dew point as 32 bit float number according to IEEE 754.
Values and states sorted by value type			
29	read	0x04	Temperature status
30	read	0x04	Humidity status
31	read	0x04	Dew point status
32	read	0x04	Current temperature – integer
33	read	0x04	Current humidity – integer
34	read	0x04	Current dew point – integer
35, 36	read	0x04	Current temperature – float
37, 38	read	0x04	Current humidity – float
39, 40	read	0x04	Current dew point – float
41	read	0x04	Current temperature – RAW value Value are 16bit number directly from A/D converter.
42	read	0x04	Current humidity – RAW value Value are 16bit number directly from A/D converter.

Papouch s.r.o.

Data transmission in industry, line and protocol conversions, RS232/485/422/USB/Ethernet/GPRS/WiFi, measurement modules, intelligent temperature sensors, I/O modules, and custom-made electronic applications.

Address:

**Strasnicka 3164
102 00 Prague 10
Czech Republic**

Tel:

+420 267 314 267

Internet:

en.papouch.com

E-mail:

info@papouch.com

