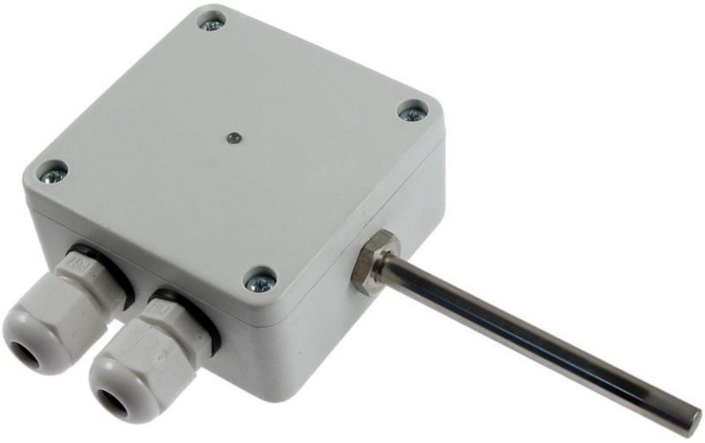
Intelligent Temperature Sensor TQS4

Measuring range -40°C to +125°C

Communication: Modbus or Spinel, RS485



TQS4

Datasheet

Created: 1.8.2018

Last update: 21.07.2021 13:33

Number of pages: 32

Papouch s.r.o.

Address:
Strasnicka 3164
102 00 Prague 10
Czech Republic

Tel:
+420 267 314 268

Internet:
www.papouch.com

E-mail:
info@papouch.com

TABLE OF CONTENTS

Changes overview

Version 1

- First version.

BASIC INFORMATION

Module is a digital temperature sensor. It measures temperature within the range from -40°C to +125°C and sends the measured value directly in degrees Celsius. The TQS4 thermometer has a very low consumption and communicates via an RS485 bus line using the Spinel protocol. These features enable connecting more sensors with a four-wire bus line containing an RS485 link and supply cable to the distance of up to 1200 m.

TQS4 thermometers are a successor of TQS3 line. TQS4 have wider power voltage range and lower consumption along with more accurate sensor. Mechanically these are the same. Communication protocols do not support ID readout for a Dallas sensor as that is no longer used. Otherwise it is identical as well.

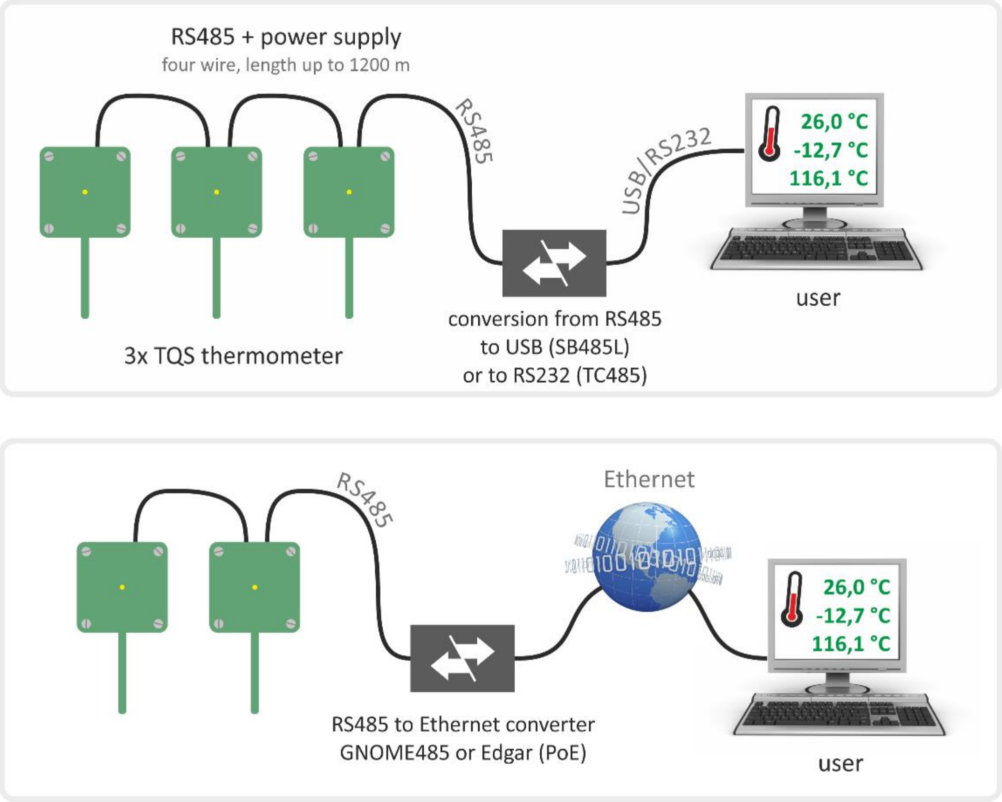


fig. 1 – Example of connecting multiple TQS4 sensors on a single RS485 bus to a PC or to Ethernet

The module measures temperature using an integrated sensor with the accuracy of ±0,5 °C within 0 to 65 °C, otherwise its accuracy is ±1,0 °C. The TQS4 module contains an indicator which blinks to signalize the ongoing temperature measurement.

TQS4 is available in the following designs:

- **Outdoor** with IP65 cover and sensor in metal rod of ø 6 mm (TQS4 O) (This design can also be ordered with a holder for wall mounting. See the picture on the right.)
- **Indoor** (TQS4 I)
- **Pipe-mount version** (TQS4 P)
- **Board with electronics** (TQS4 E)

Usage

- Comprehensive temperature measuring systems
- Industrial measurement and regulation

- Temperature measuring in warehouses, manufacturing and dwelling space
- Home automation

Features

- Temperature measuring within the range of **-40 °C to +125 °C**
- Transmission of measured temperature directly in degrees Celsius
- Communication via an RS485 link
- **Power range from 4.5 to 36 V**
- Very low **consumption** – typically **only 1.2 mA at 12 V**
- Measurement indication
- Small size
- Standardized metal cover diameter (outdoor design)
- Spinel or Modbus RTU communication protocols (can be switched by the user)
- Temperature monitoring by Wix software

SIGNALISATION

The thermometer features a yellow indicator light that lights up for a few seconds after switching the device on – this indicates the initialization of the thermometer. Then the light goes off and flashes when receiving and processing instructions.

CONNECTION

TQS4 communicates over a standard two-wire RS485 industrial bus bar. It is powered by a DC voltage of 4.5 - 36 V. The input is protected against reverse polarity.

Wago 236 terminal block is used to connect the power supply and RS485. Figure 2 shows the terminal block inside the TQS4 O box. TQS4 I uses Wago 2060-452 terminal for connections.

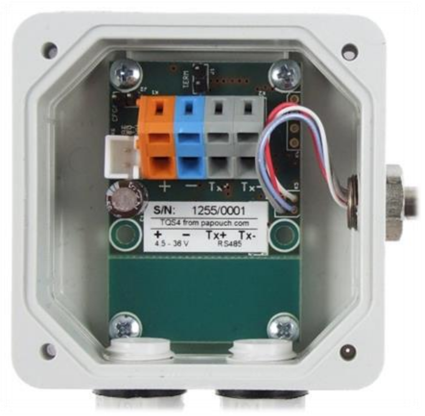


fig. 2 – Terminal block inside the box

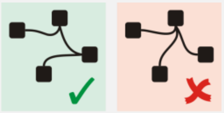
Power supply is connected to terminals + (orange) and – (blue), RS485[1] is connected to terminals Tx+ and Tx- (both grey).

[1] Manufacturers also use RS485 connections labelling as „A“ or „RxTx+“ (for Tx+) and „B“ or „RxTx-“ (for Tx-).

RS485 Connection

Some basic recommendations for connecting the RS485 line:

- It is recommended to use a standard TP cable for computer networks (UTP, FTP or STP) and to use one twisted pair from this cable as the conducting wires for RS485.
- All devices on the line must be connected “one after the other” and not in a “star” (see right). The maximum length of the line is 1.2 km.
- Cable shielding is to be connected on one side only.



The recommended cable for computer network contains four pairs of twisted wires:

- The first pair should be used for data wires. Select one wire as **Tx+** (RxTx+) and the second one as **Tx-** (RxTx-).
- The second pair: Connect the two wires and use them for the positive pole (**PWR**).
- The third pair: Connect the two wires and use them for grounding (**GND**).
- The fourth pair: Leave unconnected for possible future use.

With other devices, RS485 communication wires are connected 1:1, which means Tx+ (RxTx+) of TQS4 to RxTx+ of the other device, and similarly Tx- (RxTx-) to RxTx-.

The following picture shows an example of TQS4 connected with GNOME485 converter and other devices.

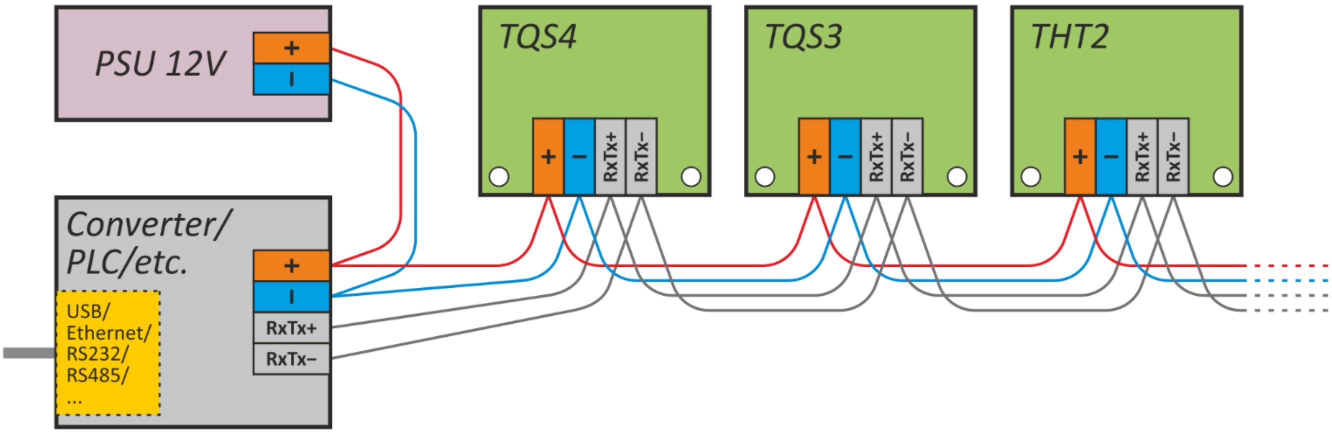


fig. 3 – example of TQS4, TQS4 and other devices connected together

COMMUNICATION PROTOCOLS

The default parameters of the communication line are:

Speed	9600 Baud
Communication protocol	Spinel
Address	31H („1“) [2]
Number of data bits	8
Parity	none
Number of stop-bits	1

Spinel

Basic communication protocol available in a text (ASCII) version (labelled 66) and binary version (labelled 97). The documentation of this communication protocol begins on page 10.

The default communication protocol of the device is Spinel. To **switch between** the Modbus and Spinel protocols, use Modbus Configurator – a utility downloadable from papouch.com.

Modbus RTU

Standard industry protocol.

(Note: The default communication protocol of the device is Spinel. To **switch between** the Modbus and Spinel protocols, use Modbus Configurator – a utility downloadable from papouch.com.)

Quick ModBus switch option

TQS4 can be switched to ModBus using the setup jumper CFG (see fig. 4).

If the unit is set to Spinel protocol (default) and setup jumper is shorted (detects only after startup), TQS4 switches to ModBus RTU regardless of the software setting stored in it.[2]

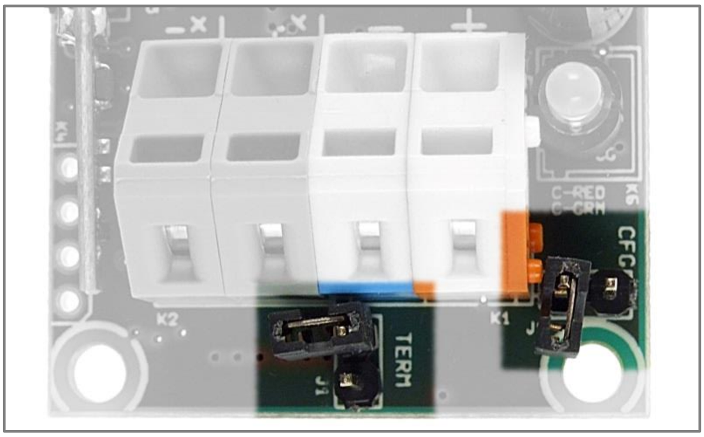


fig. 4 - if the setup jumper is shorted, TQS4 communicates via ModBus RTU (TERM jumper is used for RS485 termination)

[2] Spinel protocol address stored in a different memory location than the ModBus protocol address. Both are set to “1” in default (31H hexadecimal, 49 decadic). Change in one protocol will not affect the other one (each protocol has different address rules.)

MODBUS RTU COMMUNICATION PROTOCOL

For the initial configuration of the address, etc. we recommend Modbus Configurator.

Address

- 0x31: Default device address (49 decimal). Address can be changed in register 1 (see below).
- 0x00: Universal address of ModBus RTU protocol (0 decimal). If the device accepts this address, Instruction will be completed, but the device will not respond.
- 0xF8: Universal TQS4 address (248 decimal). If the device accepts this address, Instruction will be completed and the device will respond. This address can only be used when a single device is connected to the RS485!

List of Function Codes

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

- 0x03 .....reading of holding registers
- 0x04 .....reading of input registers
- 0x06 .....configuration of one holding register
- 0x10 .....writing in several holding registers (multiply write)
- 0x11 .....identification

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 (switched on)
Data	N Byte	String is the same as in the Spinel protocol. For example: *TQS4; v1255.01.01; f97 f67 fModbus*

Holding Register

Address	*Access*	*Function*	*Description*
0 [3]	write	0x06	**Allow configuration** Writing the 0x00FF value to this memory location must precede all instructions that write in the addresses of 0-5 in the holding register. It is used to protect against accidental configuration changes. Configuration Enabled must not be written using the 0x10 function code along with other parameters.

[3] It is possible to come across different register numbering – starting from one or zero, this first register starts at 0.

<table>
<tr><th><i>Address</i></th><th><i>Access</i></th><th><i>Function</i></th><th><i>Description</i></th></tr>
<tr><td>1</td><td>read, write</td><td>0x03, 0x06, 0x10</td><td><b>Address (ID)</b> [4]
Unique address of the device in the Modbus protocol. Number from 1 to 247 is expected. The address is unique to the Modbus protocol. <i>The default address is 0x0031.</i></td></tr>
<tr><td>2</td><td>read, write</td><td>0x03, 0x06, 0x10</td><td><b>Communication speed</b> [4]
The speeds and their corresponding codes:
1 200 Bd 0x0003
2 400 Bd 0x0004
4 800 Bd 0x0005
<i>9 600 Bd ....0x0006 (default)</i>
19 200 Bd 0x0007
38 400 Bd 0x0008
57 600 Bd 0x0009
115 200 Bd ....0x000A</td></tr>
<tr><td>3</td><td>read, write</td><td>0x03, 0x06, 0x10</td><td><b>Data word</b> [4]
Data word is always eight-bit.
<table>
<tr><th>Value</th><th>Parity</th><th>No of stop-bits</th></tr>
<tr><td><i>0x0000 (default)</i></td><td><i>none (N)</i></td><td><i>1</i></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>
</table></td></tr>
<tr><td>4</td><td>read, write</td><td>0x03, 0x06, 0x10</td><td><b>Identification of the end of the packet</b> [4]
To configure how long the delay between the bytes must be to be considered the end of the packet. The delay is specified in the number of bytes. You can specify a value ranging from 4 to 100. The default value is 10.</td></tr>
<tr><td>5</td><td>read, write</td><td>0x03, 0x06, 0x10</td><td><b>Communication protocol</b> [4]
Allows the user to switch over to Spinel protocol. After sending the response, the device switches over to the desired protocol. (Each protocol is equipped with an instruction for switching between protocols.)
Code for Spinel: 0x0001 (default)
Code for Modbus RTU: 0x0002
If the CFG jumper on the PCB is shorted, the device communicates via ModBus regardless of the state of this register.</td></tr>
<tr><td>99</td><td>read</td><td>0x03</td><td><b>Temperature Status</b>
0x0000 … the value is valid
others … the value is invalid</td></tr>
</table>

[4] Writing to this memory location must be preceded by entering the "0x00FF" value (Allow c*onfiguration*) to address 0. This prevents undesirable configuration changes. It is not allowed to enter *Allow configuration* by using Multiply Write together with other parameters.

Address	*Access*	*Function*	*Description*
100	read	0x03	**Current Temperature** This value can be used to calculate the currently measured temperature (signed integer [5]): *temperature = value / 10* The increment of the resulting temperature is 0.1°C.
101	read	0x03	**RAW value** Value as it was received from the sensors.

Input Register

Address	*Access*	*Function*	*Description*
0	read	0x04	**Temperature Status** 0x0000 … the value is valid others … the value is invalid
1	read	0x04	**Current Temperature** This value can be used to calculate the currently measured temperature (signed integer [5]): *temperature = value / 10* The increments of the resulting temperature are 0.1°C.

GETTING STARTED WITH SPINEL PROTOCOL – BASIC EXAMPLES

The following examples are based upon communication with a module in default setup. The control program sends a string indicated in the Request column. (Individual characters must not be separated by a delay longer than 5 sec.) If everything is all right, the module responds in a way indicated in the following line under the Response column.

The examples are written in a **simpler 66 format**, which is suitable for understanding the module, tuning and communication via a terminal. For the control via your application it is more suitable to use 97 format, which is described in greater detail in the chapter starting on page 17.

Temperature measuring

The following instructions will read the temperature from the thermometer with address 5.[6]

Request	**Response**	**Explanation**	
`*B5TR↵`		`*B`	Prefix
		`5`	Address It is also possible to use the $ symbol as an address. This symbol represents a universal address and works when there is only one module on the line.

[5] Negatives are in the form of two's complement. Detailed explanation can be seen for example on Wikipedia: Two's complement. In principle it is enough to add this condition to the code: `if (value > 32767) value = value - 65536;`

Example: Temperature -13,8 °C is represented as a number -138 (decimal), which is FF76H. FF76H is 65398 decadically. And 65398 minus 65536 equals -138.)

[6] The default address is „1“ (31H), unless stated otherwise on the label..

		TR	Code of instruction for temperature measurement
		↵	Final mark (enter)
	***B50+024.3C↵**	*B	Prefix
		5	Module address
		0	Confirmation
		+024.3C	Temperature; 7 characters starting with + or – sign and ending with the symbol of temperature.
		↵	Final mark (enter)

Change of address

The instruction changes the module address from **5** to **f**.

Request	Response	Explanation	
First it is necessary to enter a special instruction to enable the configuration. This instruction enables configuration for the immediately following instruction. Then the configuration is again disabled after any immediately following instruction.			
***B5E↵**		*B	Prefix
		5	Address
		E	Code of instruction for configuration authorization
		↵	Final mark (enter)
	***B50↵**	*B	Prefix
		5	Module address
		0	Confirmation
		↵	Final mark (enter)
Now the configuration is enabled. You can change the address.			
***B5ASf↵**		*B	Prefix
		5	Old address
		AS	Code of instruction for address change
		f	New address
		↵	Final mark (enter)
	***B50↵**	*B	Prefix
		5	Old address
		0	Confirmation
		↵	Final mark (enter)

SPINEL: LIST OF BASIC INSTRUCTIONS

Description	Code [Request] [Response]	Example (the address in the example is always 1)

Temperature reading	`*B[address]TR↵`
	`*B[address]0[temperature]↵`
		`*B1TR↵`
		`*B10+016.5C↵`

Device name and type request	`*B[address]?↵`
	`*B[address]0TQS4; v.199.01; F66 97↵`	

Configuration authorization[7]	`*B[address]E↵`
	`*B[address]0↵`
		`*B1E↵`
		`*B10↵`

Address setup[8]	`*B[old address]AS[new address]↵`
	`*B[old address]0↵`
		`*B1AS5↵`
		`*B10↵`

Communication speed setup[8]	`*B[address]SS[code]↵`
	`*B[address]0↵`
		`*B1SS5↵`
		`*B10↵`

Notes:

`[address]` … It is also possible to use the $ symbol as an `[address]`, which represents a universal address. It can be used when there is only one module on the line. In this case it is not necessary to address it.

`[address]` … It is also possible to use the `%` symbol as an address, which means a so called “broadcast”. It means that all modules on the line are addressed, and all of them perform the entered instruction but do not respond to prevent any collision on the line.

Communication speed Bd	Code
1200	3
2400	4
4800	5
9600	6
19200	7
38400	8
57600	9
115200	A

[7] It is not possible to use the `$` universal address for this instruction.

[8] This instruction must be preceded by the instruction Allow configuration

COMMUNICATION PROTOCOL SPINEL

The TQS4 module contains the implemented Spinel standardized protocol, as well as 66 (ASCII) and 97 (binary) formats.

Format 97

Structure

Request:

```
PRE FRM NUM NUM ADR SIG INST DATA… SUMA CR
```

Response:

```
PRE FRM NUM NUM ADR SIG ACK  DATA… SUMA CR
```

PRE Prefix, 2AH (“*“ sign).

FRM Number of 97 format (61H).

NUM Number of instruction bytes from the following bit to the end of the frame.

ADR Address of the module to which the request is being sent or which is responding to it.

SIG Message description – any number form 00H to FFH. The same number, which was sent in the request, is returned in the response, which makes it easy to see which request the response belongs to.

INST[9] Instruction code – Module instructions are described in great detail in chapter Preview of TQS4 Module Instructions on page 17.

ACK Request acknowledgement of whether and how it was executed. ACK can be 00H to 0FH.

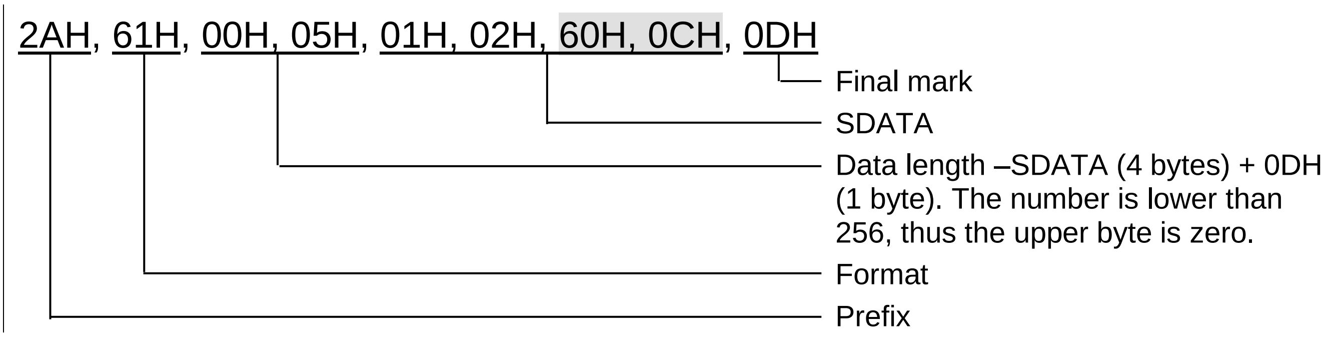
DATA[9] Data. They are described in great detail in chapter Preview of TQS4 Module Instructions (page 17) for each instruction.

SUMA Check sum.

CR Final mark (0DH).

Explanatory notes

Example


2AH, 61H, 00H, 05H, 01H, 02H, 60H, 0CH, 0DH
Final mark
SDATA
Data length –SDATA (4 bytes) + 0DH (1 byte). The number is lower than 256, thus the upper byte is zero.
Format
Prefix

[9] For easy orientation the instructions and data in the examples of following pages are highlighted this way.

Data Length (NUM)

Sixteen-bit value defining the number of bytes until the end of the instruction; number of all bytes found after NUM up to CR (including). It takes the values from 5 to 65535. If lower than 5, the instruction is considered faulty and it is answered (if intended for the relevant device) with ACK "Invalid Data" instruction.

Process of NUM creation:

Ad up the number of bytes after both NUM bytes (i.e. the number of SDATA bytes + 1 CR byte). The resulting sum view as a sixteen-bit. Divide it into the upper and lower byte. The first NUM byte id the upper byte of the number, the second NUM byte is the lower byte of the number. (If the number of bytes is lower than 256, the first NUM byte is 00H.)

Address (ADR)

The FFH address is reserved for broadcast. If the request contains the FFH address, the device operates as if its own address is entered. No response is sent to enquiries with this address.

The FEH address in the universal address. If the request contains the FEH address, the device operates as if its own address is entered. The device enters real, currently set address into the response. The universal address is used in cases where only one device is connected on the line.

Request Acknowledgement (ACK)

ACK informs the superior device on the way of the received instruction processing. Acknowledgement codes:

00H ....EVERYTHING OK
The instruction was properly received and completely executed.

01H ....ANOTHER ERROR
Unspecified device error.

02H ....INVALID CODE OF INSTRUCTION
The received instruction code is unknown.

03H ....INVALID DATA
Data are of invalid length or contain invalid value.

04H ....ENTRY NOT ALLOWED/ACCESS REFUSED
- The request was not performed, as some conditions had not been fulfilled.
- Attempt to enter data into inaccessible memory.
- Attempt to activate a device function requiring a different configuration (e.g. higher communication speed).
- Attempt to change configuration without immediately preceding setup acknowledgement.
- Access into memory protected by a password.

05H ....DEVICE FAILURE
- Device failure requiring service action.
- Device internal memory error or setup memory error.
- Device internal error (operation error or start-up error).
- Any other error affecting the device proper functioning.

06H ....NO DATA AVAILABLE

0EH ....INSTRUCTION SENT AUTOMATICALLY – CONTINUOUS MEASURING
- recurring transfer of measured values

Check Sum (SUMA)

Sum of all instruction bytes (sum of all transferred data except CR) subtracted from 255.

Calculation: SUMA = 255 – (PRE + FRM + NUM + ADR + SIG + ACK (INST) + DATA)

No response is made to messages with faulty check sum. (The system waits for the receipt of CR even if a faulty check sum is received.)

Format 66

Format 66 uses only decimal variables or characters, which can be typed using a usual keyboard.

Structure

Request:

```
PRE FRM ADR INST  DATA… CR
```

Response:

```
PRE FRM ADR ACK   DATA… CR
```

PRE Prefix, 2AH (“*“ sign).

FRM Number of 66 format (“B“ sign).

ADR Address of the module to which the request is being sent or which is responding to it.

INST[9] Instruction code – Device instruction codes. These are ASCII consisting of “A” to “Z” and “a” to “z” letter and “0” to “9” numbers. Module instructions are described in great detail in chapter Preview of TQS4 Module Instructions on page 17.

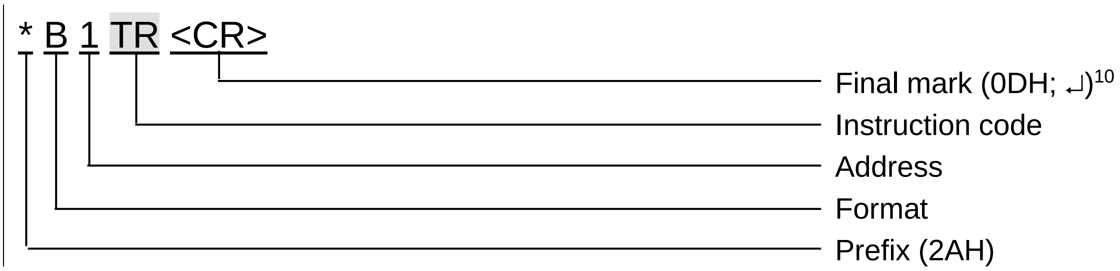
ACK Request acknowledgement of whether and how it was executed. ACK can be 00H to 0FH.

DATA[9] Data. ASCII version of transferred variables. It is recommended to transfer data in their common structure and units. Must not include prefix nor CR. They are described in great detail in chapter Preview of TQS4 Module Instructions (page 17) for each instruction.

CR Final mark (0DH).

Explanatory notes

Example – single measurement



Address (ADR)

Address is one character which unambiguously identifies and distinguishes a particular device from others on a single communication line. A device always uses this number for its identification in responses to enquiries from the superior system. The following ASCII characters can form an address: numbers “0“ to “9“, lower case letters “a“ to “z“ and capital letters “A“ to “Z“. The address must not identical with a prefix or CR.

The “%“ address is reserved for broadcast. If the request contains the “%“ address, the device operates as if its own address is entered. No response is sent to enquiries with this address.

[10] **No final mark <CR> code is given** in the **examples** of instructions in chapter Overview of TQS4 Module Instructions**!** (it is replaced with the ↵ sign.)

The “$“ address in the universal address. If the request contains the “$“ address, the device operates as if its own address is entered. The device enters real, currently set address into the response. The universal address is used in cases where only one device is connected on the line.

Instruction Code (INST)

Device instruction code.

If a valid instruction is received (correct ADR) and a flag of the received message is set, the device must respond to such instruction.

Request Acknowledgement (ACK)

ACK informs the superior device on the way of the received instruction processing. Acknowledgement codes:

0...........EVERYTHING OK
The instruction was properly received and completely executed.

1...........ANOTHER ERROR
Unspecified device error.

2...........INVALID CODE OF INSTRUCTION
The received instruction code is unknown.

3...........INVALID DATA
Data is of invalid length or contain invalid value.

4...........ENTRY NOT ALLOWED/ACCESS REFUSED
- The request was not performed, as some conditions had not been fulfilled.
- Attempt to enter data into inaccessible memory.
- Attempt to activate a device function requiring a different configuration (e.g. higher communication speed).
- Attempt to change configuration without immediately preceding setup acknowledgement.
- Access into memory protected by a password

5...........DEVICE FAILURE
- Device failure requiring service action.
- Device internal memory error or setup memory error.
- Device internal error (operation error or start-up error).
- Any other error affecting the device proper functioning.

6...........NO DATA AVAILABLE

E ....INSTRUCTION SENT AUTOMATICALLY – CONTINUOUS MEASURING
- recurring transfer of measured values.

Data (DATA)

Instruction data.

Spinel: Instructions overview

Instruction	Code 97	66	Request data (97)	Response data (97)	Page
Basic instructions					
Temperature measuring	51H	TR	---	(value)	18
Configuration					
Communication Parameters Reading	F0H	---	---	(address)(speed)	20
Communication parameters Setup	E0H	AS a SS	(address)(speed)	---	19
Additional					
Communication Errors Reading	F4H	---	---	(errors)	25
Name and Version Reading	F3H	?	---	TQS4; v1255.01.01; f97 f67	22
Status Reading	F1H	SR	---	(state)	22
Saved User Data Reading	F2H	DR	---	---	24
Manufacturing Data Reading	FAH	---	---	(product-number)(serial)(factory-data)	26
Read RAW value	5FH			(raw)	25
Check Sum – Setup Reading	FEH	---	---	(state)	23
Address Setup using Serial Number	EBH	---	(new-address) (product-number)(serial)		25
Status Setup	E1H	SW	(state)	---	21
Allow configuration	E4H	E	---	---	21
Allow checksum	EEH	---	(state)	---	23
Reset	E3H	RE	---	---	23
User Data Saving	E2H	DW	---	---	24
Switching between Communication Protocols	EDH				26

Only instructions (INST), acknowledgement (ACK) and data (DATA) are described in detail to preserve the clarity. Other instructions are described in detail on a separate Spinel documentation at papouch.com).

However – examples for the 01H address and 02H signature are described in their full form. The indexes [97] or [66] before some paragraphs indicate which format of Spinel protocol they are intended for. If no index appears before a paragraph the given information applies to both protocols 97 as well as 66. (See also note 10 on page 13.)

Basic Instruction

Temperature Measuring

Description: Performs a single temperature measurement.

[97]Request: 51H

[97]Response: (ACK 00H) (value)

[97]Legend: (value) temperature in the signed int format [5]

temperature = value / 32 resulting temperature with resolution of 0,1°C.

[97]*Example: Request: Address 1*

```
2AH,61H,00H,05H,01H,02H,51H,1BH,0DH
```

Response:

```
2AH,61H,00H,07H,01H,02H,00H,01H,05H,64H,0DH
```

The temperature in the response is in the format of signed int: 0105H. By dividing by 32 we will get the measured value in degrees Celsius.
Conversion into decimal value: `0105H = 261DEC`
Division by 32: `261 / 32 = 8,15625`
The measured temperature is (after rounding up) 8,2 °C.

[66]Request: „TR“ (Temperature Read)

[66]Response: (ACK „0“) (value)

[66]Legend: (value) Temperature as an ASCII string (always 7 characters justifies to the right). Unused characters are filled in with a space (20H).

[66]*Example: Request: Address 1*

```
*B1TR↵
```

Response: 123,4°C

```
*B10+123.4C↵
```

Configuration

Communication parameters Setup

Description: Set the address and communications speed. This instruction must be immediately preceded by the instruction of Allow configuration (see page. 21). This instruction can not be used with universal or broadcast addresses.

^{97}Request: E0H (address) (speed)

^{97}Response: (ACK 00H)

^{97}Legend: (address) 1 byte; Can be of value between 00H to FDH, if the 66 protocol is also used for communication it is necessary to use only such addresses, which can be formulated as a displayable ASCII character (see paragraph Address on page 15).

(speed) 1 byte; communication speed, speed codes can be found in Table 1.

^{97}*Example:* *Setup of the 04H address and communication speed of 19200Bd; old address 01H, signature 02H*

```
2AH,61H,00H,07H,01H,02H,E0H,04H,07H,7FH,0DH
```

Response

```
2AH,61H,00H,05H,01H,02H,00H,6CH,0DH
```

Notes: The new address and communication speed are set after response sending.

The setup of configuration parameters must be preceded by the instruction of Allow configuration (see page. 21). After the communication parameters are set, the configuration is again disabled.

Other communication parameters are: 8 bits, no parity, 1 stop-bit. The default communication speed set by the manufacturer is 9600Bd, the address is indicated as an ASCII sign on the thermometer label.

In case the address is not known and no other device is connected on the line, the address can be found out using the instruction of Communication Parameters Reading. The universal FEH address is used as the device address.

In case the communication speed is unknown, it is necessary to try out all communication speeds available.

^{66}Request: „AS“(address)[11] (Address Set)

^{66}Response: (ACK „0“)

^{66}Legend: (address) see the paragraph Address on page 15.

^{66}*Example:* *Request: Address 4*

```
*B1AS4↵
```

Response

```
*B10↵
```

[11] The address and communication speed must be set in the protocol 66 via two separated instructions (Protocol 97 this can be done using just a single instruction.)

66Request: „SS“(code)11 (Speed Set)

66Response: (ACK „0“)

66Legend: (code) communication speed as defined in Table 1

66*Example:* *Request: Speed 19200Bd*

**B1SS7↵*

Response

**B10↵*

Communication Parameters Reading

Description: Returns the address and communication speed.

97Request: F0H

97Response: (ACK 00H) (adr) (speed)

97Legend: (address) 1 byte; device address

(speed) 1 byte; communication speed; speed codes can be found in Table 1.

97*Example:* *Communication parameters reading; FEH universal address, signature: 02H*

2AH,61H,00H,05H,FEH,02H,F0H,7FH,0DH

Response- address: 04H, communication speed: 9600Bd

2AH,61H,00H,07H,04H,02H,00H,04H,06H,5DH,0DH

Communication speed Bd	Code	
	97	66
1200	03H	3
2400	04H	4
4800	05H	5
9600	06H	6
19200	07H	7
38400	08H	8
57600	09H	9
115200	0AH	A

Table 1 – communication speeds codes

97Notes: This instruction is designed for the detection of the set address of the device in case it is unknown. The request is sent to the FEH universal address. If even the communication speed is not known it is necessary to try out all communication speeds available for the particular device. However, no other device can be connected on the line in this case.

Other communication parameters are: 8 bits, no parity, 1 stop-bit. The default communication speed set by the manufacturer is 9600Bd, the address is indicated as an ASCII sign on the thermometer label.

Additional

Allow configuration

Description: Enables configurations to be carried out. It must immediately precede some instructions (Communication Parameters Setup and Check Sum Acknowledgement). After a following instruction (even an invalid one) the configuration is again automatically disabled. This instruction can not be used with the universal or broadcast address.

97Request: E4H

97Response: (ACK 00H)

97Example: Allow configuration

`2AH,61H,00H,05H,01H,02H,E4H,88H,0DH`

Response

`2AH,61H,00H,05H,01H,02H,00H,6CH,0DH`

66Dotaz: „E“ *(Enable)*

66Odpověď: (ACK „0“)

66Příklad: Request

`*B1E↵`

Response

`*B10↵`

Status Setup

Description: Sets the device status. User-defined byte, which can be used to find out the device condition.

97Request: E1H (status)

97Response: (ACK 00H)

97Legend: (status) 1 byte; device status. After the device is switched on or reset (even software) the status of 00H is set automatically. If a new value is set using the Status Setup instruction it is subsequently easy to identify the current status of the device.

97Example: Status of 12H setup; address: 01H, signature: 02H

`2AH,61H,00H,06H,01H,02H,E1H,12H,78H,0DH`

Response

`2AH,61H,00H,05H,01H,02H,00H,6CH,0DH`

66Request: „SW“(status) (Status Write)

66Response: (ACK „0“)

66Legend: (status) character from the interval of “space“ to “~“ (32 – 126)

66Example: Request – A character

`*B1SWA↵`

Response

`*B10`

Status Reading

Description: Read the device status. User-defined byte, which can be used to find out the device condition.

^{97}Request: F1H

^{97}Response: (ACK 00H)(status)

^{97}Legend: (status) 1 byte; device status, for more info see Status Setup.

^{97}*Example: Status reading; address: 01H, signature: 02H*

```
2AH,61H,00H,05H,01H,02H,F1H,7BH,0DH
```

Response- status: 12H

```
2AH,61H,00H,06H,01H,02H,00H,12H,59H,0DH
```

^{66}Request: „SR“ (Status Read)

^{66}Response: (ACK „0“)(character)

^{66}Legend: (character) character from the interval of “space“ to “~“ (32 – 126)

^{66}*Example: Request*

```
*B1SR↵
```

Response

```
*B10A↵
```

Name and Version Reading

Description: Reads the name of the device, version of the internal software and list of possible communication formats (for TQS4 97 and 66). Set by the manufacturer.

^{97}Request: F3H

^{97}Response: (ACK 00H) (string)

^{97}Legend: (string) Text in the form of: “*TQS4; v1255.01.01; f97 f67 fModbus“*.

^{97}*Example: Request*

```
2AH,61H,00H,05H,31H,02H,F3H,49H,0DH
```

Response

```
2AH,61H,00H,1BH,31H,02H,00H,54H,51H,53H,33H,3BH,20H,76H,30H,31H,39H
,39H,2EH,30H,31H,3BH,20H,46H,36H,36H,20H,39H,37H,2BH,0DH
```

^{66}Request: „?“

^{66}Response: (ACK „0“)

^{66}*Example: Request*

```
*B1?↵
```

Response

```
*B10 TQS4; v1255.01.01; f97 f67 fModbus↵
```

Reset

Description: Carries out the device reset. The module enters the same condition as after supply switching on.

^{97}Request: E3H

^{97}Response: (ACK 00H)

^{97}*Example:* *Reset; address: 01H, signature: 02H*

```
2AH,61H,00H,05H,01H,02H,E3H,89H,0DH
```

Response

```
2AH,61H,00H,05H,01H,02H,00H,6CH,0DH
```

Note: Reset is carried out after the response is sent.

^{66}Request: „RE“ *(REset)*

^{66}Response: (ACK „0“)

^{66}*Example:* *Request*

```
*B1RE↵
```

Response

```
*B10↵
```

Allow checksum

Description: Enables the verification of check sum in the incoming messages. This instruction must be immediately preceded by the instruction of Allow configuration (see page. 21).

^{97}Request: EEH (status)

^{97}Response: (ACK 00H)

^{97}Legend: (status) 1 byte; 01H for check-up switching on; 00H for switching off

^{97}*Example:* *Allow configuration*

```
2AH,61H,00H,06H,01H,02H,EEH,01H,7CH,0DH
```

Response

```
2AH,61H,00H,05H,01H,02H,00H,6CH,0DH
```

Check Sum – Setup Reading

Description: Finds out the current set up of the check sum verification.

^{97}Request: FEH

^{97}Response: (ACK 00H) (status)

^{97}Legend: (status) 1 byte; 01H for verification switching on; 00H for switching off

^{97}*Example:* *Setup request*

```
2AH,61H,00H,05H,01H,02H,FEH,6EH,0DH
```

Response – verification switched on

```
2AH,61H,00H,06H,01H,02H,00H,01H,6AH,0DH
```

User Data Saving

Description: The instruction saves user data. The device remembers the data after supply disconnection.

97Request: E2H (position)(data)

97Response: (ACK 00H)

97Legend: (position) 1 byte; address of the memory where the data are to be saved. 00H to 0FH

(data) 1 to 16 bytes; any user data.

97Example: Saving the expression "BOILER ROOM 1" on the memory address of 00H; address: 01H, signature: 02H

```
2AH,61H,00H,12H,01H,02H,E2H,42H,4FH,49H,4CH,45H,52H,20H,52H,4FH,4FH
,4DH,20H,31H,12H,0D
```

Response

```
2AH,61H,00H,05H,01H,02H,00H,6CH,0DH
```

Notes: The memory for user data has the capacity of 16 bytes. In case the data are being written to the memory address e.g. 0CH, it is possible to write 4 bytes maximum.

66Request: „DW“(position)(data) *(Data Write)*

66Response: (ACK „0“)

66Legend: (position) address of the memory position to which the data will be written. Interval 0-9 or A-F.

(data) 1 to 16 bytes; any user data. Interval 0-9 or A-F.

66Example: Request

```
*B1DW0BOILER ROOM 1↵
```

Response

```
*B10↵
```

Saved User Data Reading

Description: The instruction reads saved user data. The device remembers the data after supply disconnection.

97Request: F2H

97Response: (ACK 00H)(data)

97Legend: (data) 16 bytes; saved user data.

97Example: User data reading; address: 01H, signature: 02H

```
2AH,61H,00H,05H,01H,02H,F2H,7AH,0DH
```

Response - "BOILER ROOM 1   "

```
2AH,61H,00H,15H,01H,02H,00H,42H,4FH,49H,4CH,45H,52H,20H,52H,4FH,4FH
,4DH,20H,31H,20H,20H,20H,91H,0DH
```

66Request: „DR“ *(Data Read)*

66Response: (ACK „0“)(data)

66Legend: (data) 1 to 16 bytes; User data.

66Example: Request

```
*B1DR↵
```

Response

```
*B10BOILER ROOM 1↵
```

Communication Errors Reading

Description: The instruction returns the number of communication errors which have occurred since the device switching on or since the last communication errors reading.

^{97}Request: F4H

^{97}Response: (ACK 00H) (errors)

^{97}Legend: (errors) 1 byte; the number of errors which have occurred since the device switching on or since the last errors reading. The following events are considered communication errors:

Prefix is expected but another byte is received

SUMA check sum does not agree

Message is incomplete

^{97}*Example:* *Communication errors reading; address: 01H, signature: 02H*

```
2AH,61H,00H,05H,01H,02H,F4H,78H,0DH
```

Response - 5 errors

```
2AH,61H,00H,06H,01H,02H,00H,05H,66H,0DH
```

Read RAW value

Description: Instruction reads value as it is from the sensor.

^{97}Request: 5FH

^{97}Response: (ACK 00H)

^{97}Legend: (raw) 2 bytes; value from the sensor.

^{97}*Example:* *Request*

```
2AH,61H,00H,05H,31H,02H,5FH,DDH,0DH
```

Response – 25,3 °C

```
2AH,61H,00H,07H,31H,02H,00H,01H,96H,A3H,0DH
```

Address Setup using Serial Number

Description: The instruction enables the module address to be set using the serial number only.

^{97}Request: EBH(new-address)(product-number)(serial-number)

^{97}Response: (ACK 00H)

^{97}Legend: (new-address) 1 byte; new address of the module.

(product-number) 2 bytes; product number; for TQS4 thermometer it is always decimally: 199, thus hexadecimally: 00C7.

(serial-number) 2 bytes; the TQS4 thermometer serial number is indicated on the label after the 0199.01/ text. This number can also be found out via the Manufacturing Data Reading instruction.

^{97}*Example:* *Request – new address: 32H, product-number: 199 (= 00C7H),*
serial number: 101 (= 0065H)

```
2AH,61H,00H,0AH,FEH,02H,EBH,32H,00H,C7H,00H,65H,21H,0DH
```

Response – the thermometer already responds with the new address

```
2AH,61H,00H,05H,32H,02H,00H,3BH,0DH
```

Manufacturing Data Reading

Description: The instruction reads the manufacturing data from the TQS4 thermometer

^{97}Request: FAH

^{97}Response: (ACK 00H)(product-number)(serial-number)(manufacturing-data)

^{97}Legend: (product-number) 2 bytes; product number; for TQS4 thermometer it is always decimally: 199, thus hexadecimally: 00C7.

(serial-number) 2 bytes; the TQS4 thermometer serial number is indicated on the label after the 0199.01/ text.

(manufacturing-data) 4 bytes

^{97}*Example:* *Request*

```
2AH,61H,00H,05H,FEH,02H,FAH,75H,0DH
```

Response – PN: 199 (=00C7H), serial number 101 (=0065H), manufacturing-data 20050923H

```
2AH,61H,00H,0DH,35H,02H,00H,00H,C7H,00H,65H,20H,05H,09H,23H,B3H,0DH
```

Switching between Communication Protocols

Description: This instruction switches between the types of the communication protocols. (It must be preceded by the instruction Allow configuration on page 21.)

To switch between the protocols, it is possible to use e.g. Modbus Configurator, downloadable from papouch.com.

^{97}Request: EDH (id)

^{97}Response: (ACK 00H)

^{97}Legend: (id) 1 byte; protocol identification number:
01H – Spinel protocol, format 97 (binary) and 66 (ascii)
02H – MODBUS RTU protocol

^{97}*Example:* *Request*

```
2AH,61H,00H,06H,31H,02H,EDH,FFH,4FH,0DH
```

Response

```
2AH,61H,00H,05H,31H,02H,00H,3CH,0DH
```

TECHNICAL PARAMETERS

Common

Electronics operating temperature	-40 °C to +85 °C
Sensor temperature range	-40 °C to +125 °C; resolution 0.1 °C
Accuracy	± 0.5 °C between 0°C and +65°C, otherwise ± 1 °C
Measuring unit	TMP112
Termination	resistor 120 Ω (can be enabled using the TERM jumper
Idle state definition	resistors 22 kΩ
Supply voltage	4,5 V to 36 V DC with reverse polarity protection
Consumption	typ. 1,2 mA at 12 V; typ. 0,7 mA at 24V

Communication line:

Type	RS485
Addressability	software
Response time	2.5 ms
Communication protocol	Spinel or Modbus RTU *(switched by the user)*
Default communication protocol	Spinel
Speed	up to 115.2 kBd
default address	31H (character: „1“) [2]
Number of data bits	8
Parity	none
Number of stop-bits	1

Outdoor Design – TQS4 O

This design is suitable for measurements in outdoor environments, places exposed to water or various weather conditions. The probe rod can also be put in a sensor basin.

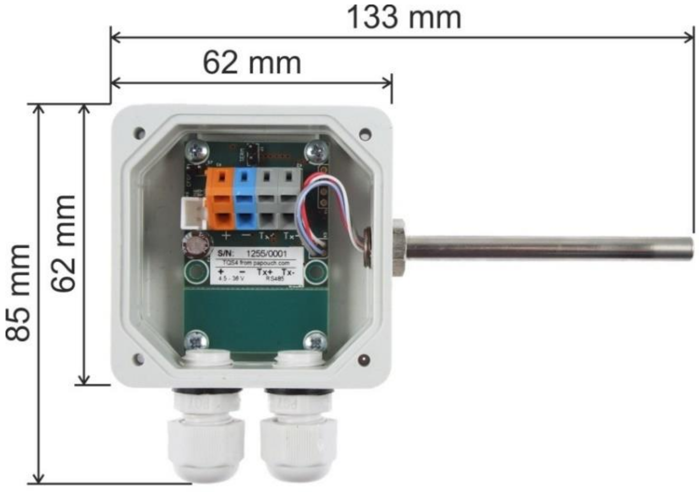


fig. 5 – Outdoor design of TQS4 O

Housing type	IP65
Sensor mechanical design	Metal rod: 6 mm in diameter, 70 mm long
Dimensions	83 (62) mm x 138 (62) mm x 33 mm
Connection	2× PG7 cable bushing
Lines connection	Wago 236 terminal block

Mounting options

1) *Default make: No mounting holes, placement into sensor basin or free installation*.
2) Wall holder for the enclosure (∅ 4mm holes distance is 73mm): [12]

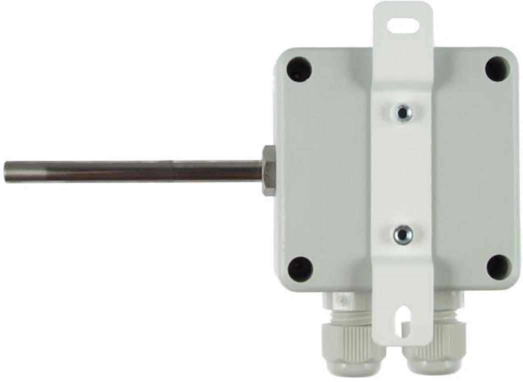


fig. 6 - Thermometer TQS with wall holder mounted on the enclosure

3) Wall holder mounted on the probe rod (∅ 3mm holes distance is 30mm): [12]

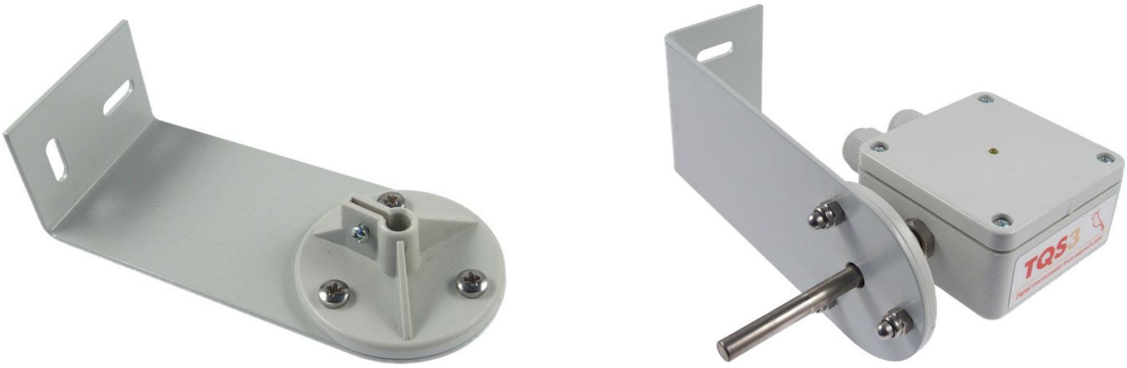


fig. 7 – Probe rod Holder with TQS sensor

4) Rod holder for measurements within closed spaces: [12]

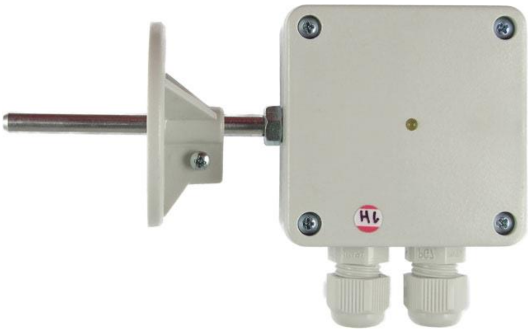


fig. 8 – TQS Thermometer with probe rod mount.

[12] This accessory is sold separately.

5) DIN rail holder: [12]

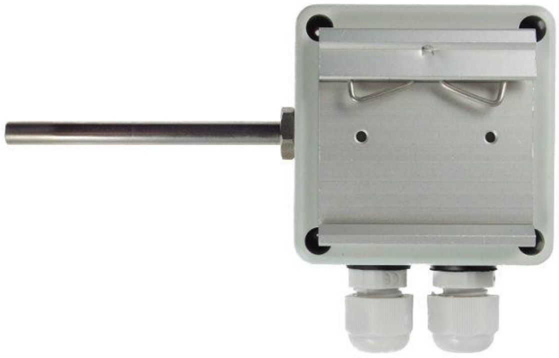


fig. 9 – Thermometer TQS with DIN rail holder

Indoor Design – TQS4 I

Design for temperature measurements in the interiors where there is no contact with water or excessive moisture.

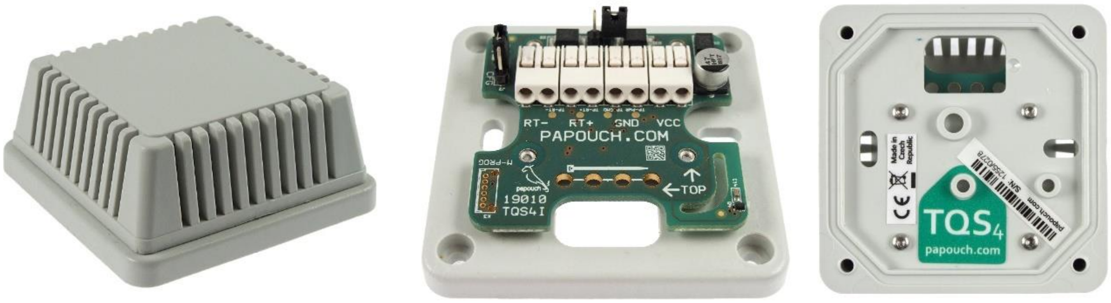


fig. 10 - TQS4 I: boxed; PCB inside; mount holes placement with cable feed-through holes

Housing type	IP20
Sensor mechanical design	SMD sensor directly on the PCB
Dimensions	62 mm x 62 mm x 29 mm
mount holes diameter	4 mm
Cables connections	Wago 2060-452 terminal

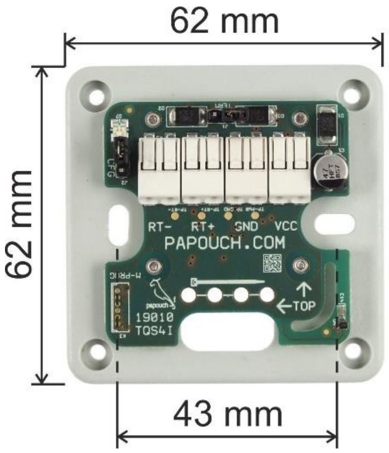


fig. 11 – Indoor design – TQS4 I

Surface Design – TQS4 P

Design for measuring the temperature of pipes or other curved objects.

Housing type	IP65
Sensor mechanical design	To be placed on pipes
Dimensions	62 mm x 62 mm x 45 mm
Cables connections	Wago 236 terminal

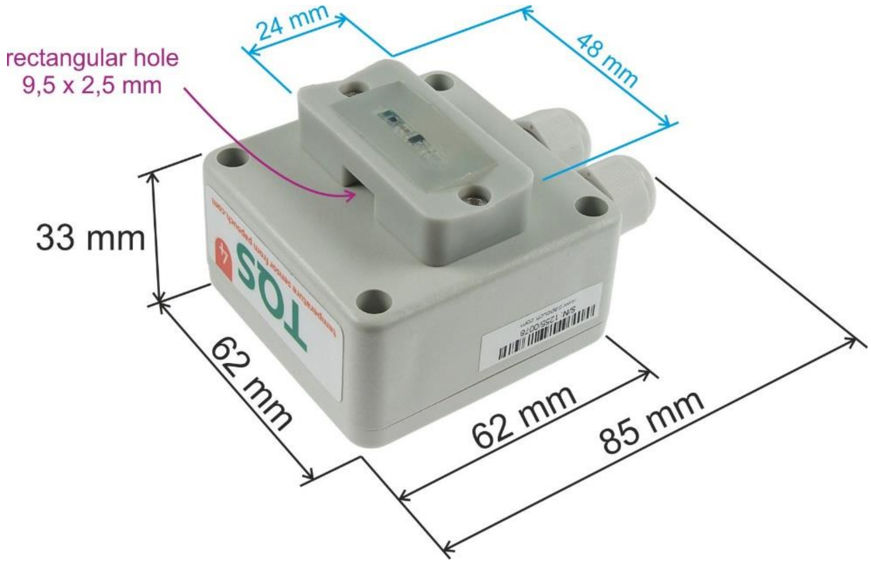


fig. 12 – Pipe-mount version TQS4 P

Board with Electronics – TQS4 E

A separate electronics board where the temperature sensor is mounted directly on the board.

Board dimensions	35(50) mm x 36 mm x 17 mm
Mounting holes diameter	3 mm
Sensor mechanical design	sensor on the PCB

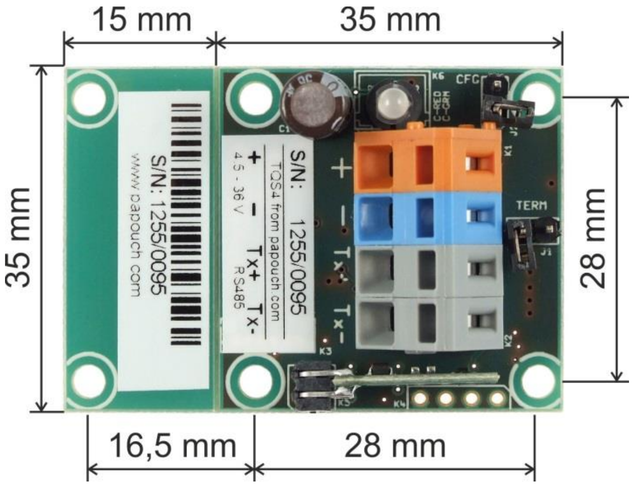


fig. 13 – PCB dimensions and mounting holes placement

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:

+420 267 314 268

Internet:

www.papouch.com

E-mail:

info@papouch.com