Intelligent Temperature Sensor TQS3

Measuring range -55°C to +125°C

Communication: Modbus or Spinel, RS485

TQS3

Datasheet

Created: 5.3.2007

Last update: 11.04.2019 10:54

Number of pages: 32

Papouch s.r.o.

Address:

Strasnicka 3164
102 00 Prague 10
Czech Republic

Tel:

+420 267 314 268

Fax:

+420 267 314 269

Internet:

www.papouch.com

E-mail:

info@papouch.com

CONTENT

Changes overview

Version 4.4

- Fixed negative temperature readout error.

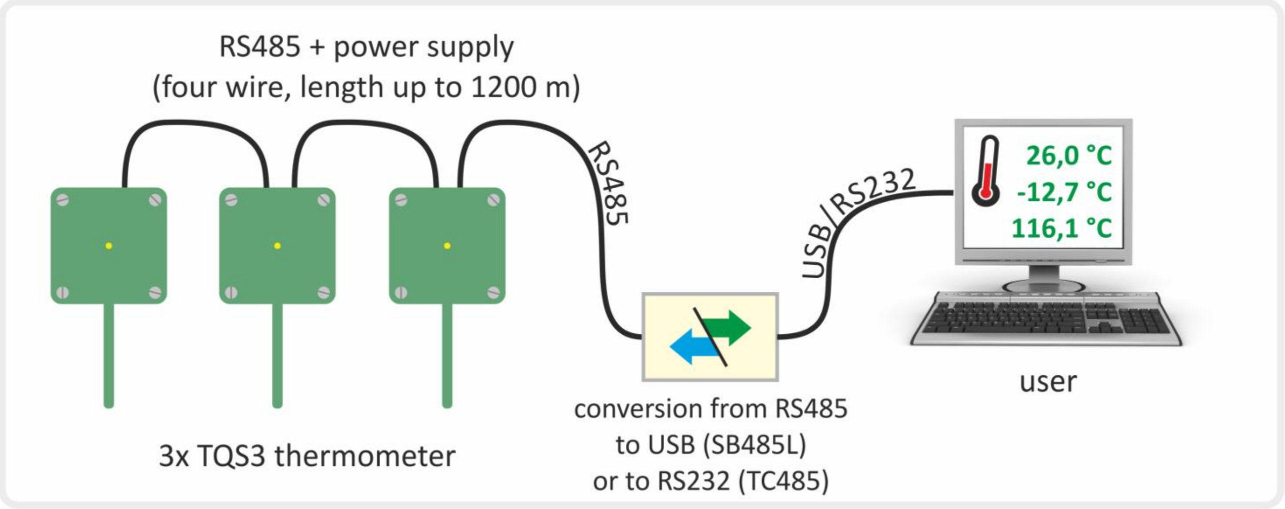
Version 4.3

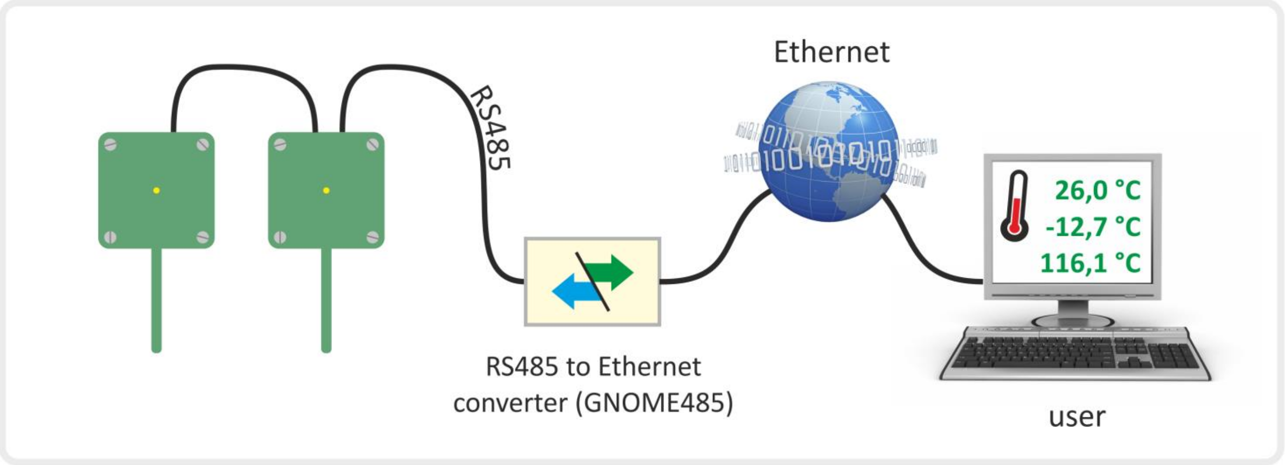
- Possibility of the sensors ID and a “RAW value” directly from the sensor added.
- Holding registers 100 and 101 added. (Mirrored values from input registers).

BASIC INFORMATION

Description

TQS3 Module is a digital temperature sensor. It measures temperature within the range from -55°C to +125°C and sends the measured value directly in degrees Celsius. The TQS3 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.





The module measures temperature using an integrated unit with the accuracy of ± 0,5°C. The TQS3 module contains an indicator which blinks to signalize the ongoing temperature measurement.

TQS3 is available in the following designs:

- **Outdoor** with IP65 cover and sensor in metal rod of ø 6 mm (TQS3 O) (This design can also be ordered with a holder for wall mounting. See the picture on the right.)
- **Indoor** (TQS3 I)
- **Surface** to measure temperature of pipes (TQS3 P)
- **Board with electronics** (TQS3 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 **-55°C to +125°C**
- Transmission of measured temperature directly in degrees Celsius
- Communication via an RS485 link
- Very low **consumption** – typically **only 2 mA**
- Measurement indication
- Small size
- Unstabilized supply voltage
- 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

TQS3 communicates over a standard two-wire RS485 industrial bus bar. It is powered by a DC voltage of 7-20 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 TQS3 O box.

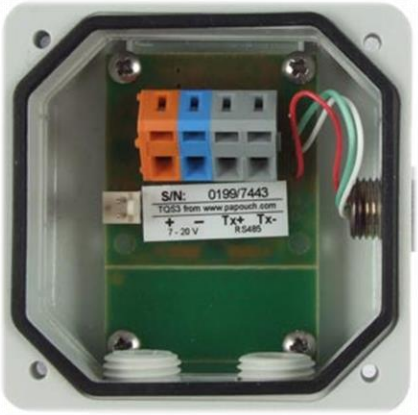


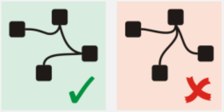
Fig. 1 – Terminal block inside the box

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

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 TQS3 to RxTx+ of the other device, and similarly Tx- (RxTx-) to RxTx-.

The following picture shows an example of TQS3 connected with GNOME485 converter and other devices.

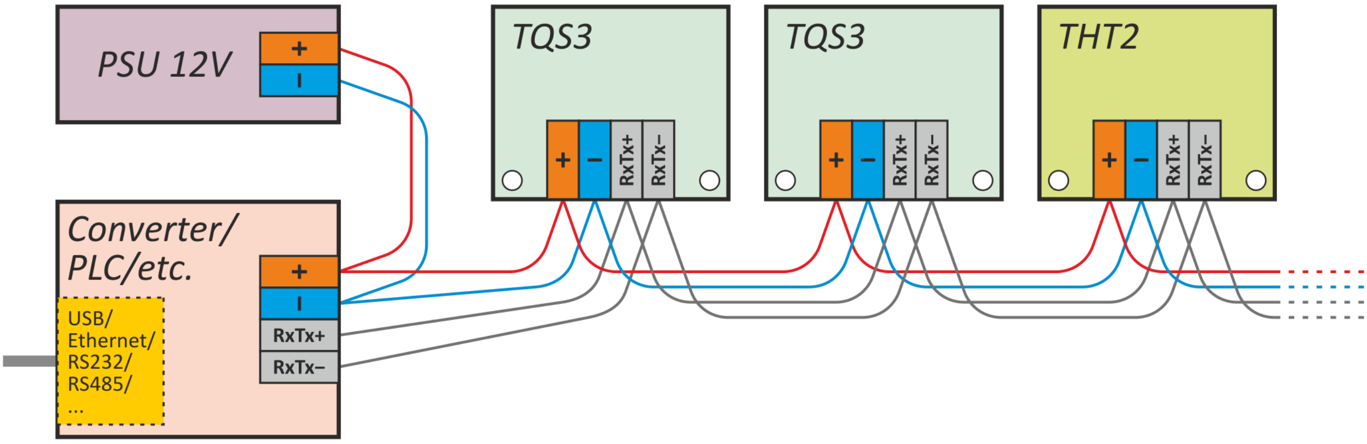


Fig. 2 – example of TQS3 connected to a converter and other devices

COMMUNICATION PROTOCOLS

The default parameters of the communication line are:

Speed	9600 Baud
Communication protocol	Spinel
Address	31H („1“)
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 www.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 www.papouch.com.)

Quick ModBus switch option

TQS3 can be switched to ModBus using the setup jumper (see fig. 3).

If the unit is set to Spinel protocol (default) and setup jumper is shorted (detects only after startup), TQS3 switches to ModBus RTU regardless of the software setting stored in it.

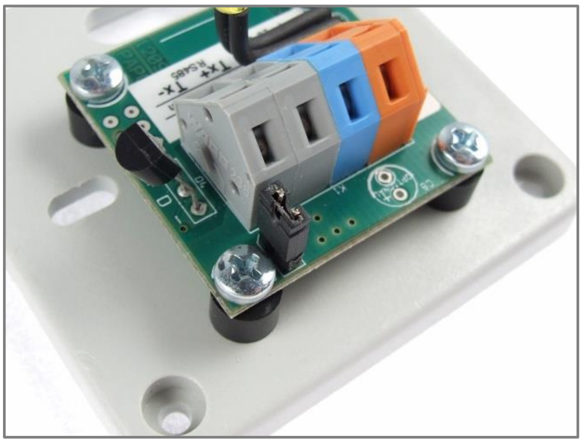


fig. 3 - if the setup jumper is shorted, TQS3 communicates via ModBus RTU

MODBUS RTU COMMUNICATION PROTOCOL

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

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: *TQS3; v0199.04.02; F66 97*

Holding Register

Address	*Access*	*Function*	*Description*
0	write	0x06	**Configuration Enabled** 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.
1	read, write	0x03, 0x06, 0x10	**Address (ID)**[1] Unique address of the device in the Modbus protocol. Number from 1 to 247 is expected. The address is unique to the Modbus protocol. *The default address is 0x0031.*

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

<table>
<tr><td>2</td><td>read, write</td><td>0x03, 0x06, 0x10</td><td><b>Communication speed</b>1
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>[1]
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> [1]
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>[1]
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 <i>Spinel: 0x0001 (default)</i>
Code for Modbus RTU: 0x0002</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>
<tr><td>100</td><td>read</td><td>0x03</td><td><b>Current Temperature</b>
This value can be used to calculate the currently measured temperature:
<i>temperature = value / 10</i>
The increment of the resulting temperature is 0.1°C.</td></tr>
<tr><td>101</td><td>read</td><td>0x03</td><td><b>RAW value</b>
Value as it was received from the sensors.</td></tr>
<tr><td>105</td><td>read</td><td>0x03</td><td><b>Sensor status ID</b>
It can be of these values:
0x0000..........Error, ID is invalid.
0x0001..........ID is just being read.
0x00FF..........ID is valid</td></tr>
<tr><td>106 – 109</td><td>read</td><td>0x03</td><td><b>Sensor ID</b>
Temperature sensor unique identifier. Validity is in the previous register.</td></tr>
</table>

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: *temperature = value / 10* The increments of the resulting temperature is 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 Enquiry 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 18.

Temperature measuring

The following instructions will read the temperature from the thermometer with address 5.[2]

Enquiry	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.
		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)

[2] The default address set by the manufacturer is indicated on the label of the TQS3 sensor. (The address is indicated as an ASCII sign.)

Change of address

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

Enquiry	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 [Enquiry] [Response]	Example (the address in the example is always 1)
Temperature reading	`*B[address]TR↵`	`*B1TR↵`
	`*B[address]0[temperature]↵`	`*B10+016.5C↵`
Device name and type enquiry	`*B[address]?↵`	
	`*B[address]0TQS3; v.199.01; F66 97↵`	
Configuration authorization[3]	`*B[address]E↵`	`*B1E↵`
	`*B[address]0↵`	`*B10↵`
Address setup[4]	`*B[old address]AS[new address]↵`	`*B1AS5↵`
	`*B[old address]0↵`	`*B10↵`
Communication speed setup[4]	`*B[address]SS[code]↵`	`*B1SS5↵`
	`*B[address]0↵`	`*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

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

[4] This instruction must be preceded by the instruction for Configuration Authorization

COMMUNICATION PROTOCOL SPINEL

The TQS3 module contains the implemented Spinel standardized protocol, as well as 66 (ASCII) and 97 (binary) formats.

Format 97

Structure

Enquiry:

```
  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 enquiry 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 enquiry, is returned in the response, which makes it easy to see which enquiry the response belongs to.

INST[5] Instruction code – Module instructions are described in great detail in chapter Preview of TQS3 Module Instructions on page 18.

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

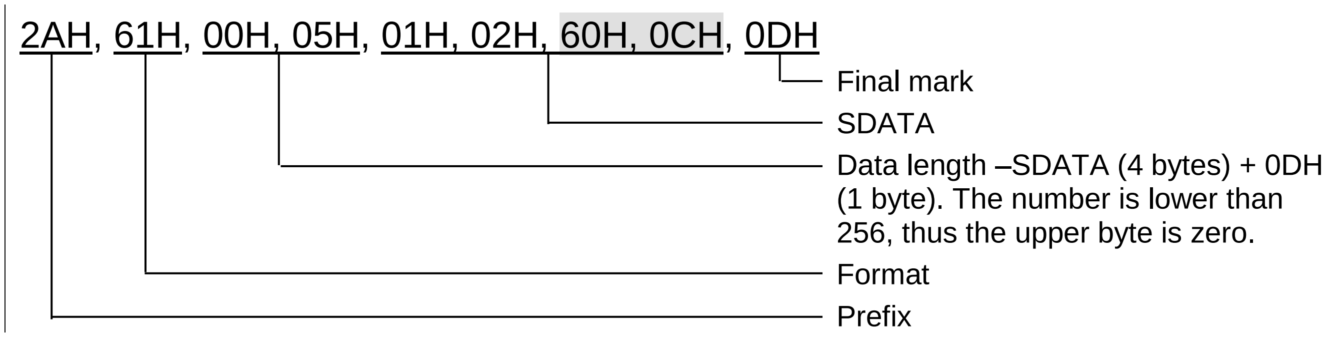
DATA[5] Data. They are described in great detail in chapter Preview of TQS3 Module Instructions (page 18) for each instruction.

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

[5] 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 enquiry 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 enquiry 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.

Enquiry 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 enquiry 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

Enquiry:

```
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 enquiry is being sent or which is responding to it.

INST[5] 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 TQS3 Module Instructions on page 18.

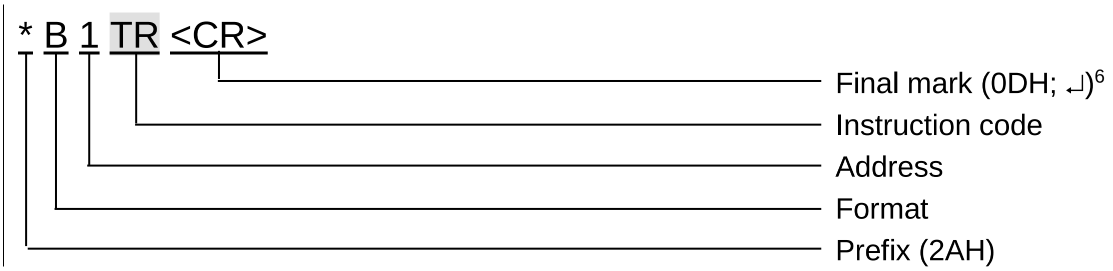
ACK Enquiry acknowledgement of whether and how it was executed. ACK can be 00H to 0FH.

DATA[5] 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 TQS3 Module Instructions (page 18) for each instruction.

CR Final mark (0DH).

Explanatory notes

Example – single measurement



* B 1 TR <CR>

Final mark (0DH; ↵)[6]

Instruction code

Address

Format

Prefix (2AH)

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 enquiry contains the "%" address, the device operates as if its own address is entered. No response is sent to enquiries with this address.

[6] **No final mark <CR> code is given** in the **examples** of instructions in chapter Overview of TQS3 Module Instructions**!** (it is replaced with the ↵ sign.)

The “$“ address in the universal address. If the enquiry contains the “$“ 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.

Enquiry 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 enquiry 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	Enquiry data (97)	Response data (97)	Page
Basic instructions					
Temperature Measuring	51H	TR	---	(value)	19
Configuration					
Communication Parameters Reading	F0H	---	---	(address)(speed)	21
Communication Parameters Setup	E0H	AS a SS	(address)(speed)	---	20
Additional					
Configuration Acknowledgement	E4H	E	---	---	22
Name and Version Reading	F3H	?	---	*TQS3; v0199.00; F66 97*	23
Status Reading	F1H	SR	---	(status)	23
Status Setup	E1H	SW	(status)	---	22
Saved User Data Reading	F2H	DR	---	---	25
User Data Saving	E2H	DW	---	---	25
Read ID from Dallas sensor	A0H			(status)(id)	26
Read RAW value	5FH			(raw)	26
Check Sum Acknowledgement	EEH	---	(status)	---	24
Check Sum Acknowledgement	FEH	---	---	(status)	24
Manufacturing Data Reading	FAH	---	---	(product No)(serial No)(manufacturing data)	27
Address Setup using Serial Number	EBH	---	(new address)(product No)(serial No)		27
Communication Errors Reading	F4H	---	---	(errors)	26
Reset	E3H	RE	---	---	24

On the other hand, the examples are presented in their full form – for the 01H address and 02H signature. 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 6 on page 16.)

Basic Instruction

Temperature Measuring

Description: Performs a single temperature measurement.[7]

^{97}Enquiry: 51H

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

^{97}Legend: (value) temperature in the signed int format

temperature = value / 32 resulting temperature with resolution of 0,1°C.[8]

^{97}*Example: Enquiry: 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}Enquiry: „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: Enquiry: Address 1*

`*B1TR↵`

Response: 123,4°C

`*B10+123.4C↵`

[7] TQS3 thermometer dynamically changes the internal interval of temperature measuring by the temperature sensor according to the frequency of the instruction of “Temperature Measuring”. If you wish to minimize the thermometer current consumption use a measurement interval longer than 30 sec (The internal temperature conversion takes approx. 700 ms. During this time the TQS3 thermometer consumption increases by 0,5 mA.)

[8] After the division the temperature is given with the accuracy of 1/32. The guaranteed accuracy of the temperature sensor is only 1/10. Thus the resulting value is a number rounded to decimals. Rounding of the calculated temperature value is illustrated in the example to this instruction.

Configuration

Communication Parameters Setup

Description: Set the address and communications speed. This instruction must be immediately preceded by the instruction of Configuration Acknowledgement (see page 22). This instruction can not be used with universal or broadcast addresses.

^{97}Enquiry: 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 16).

(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 Configuration Acknowledgement (page 22). 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}Enquiry: „AS“(address)[9] (Address Set)

^{66}Response: (ACK „0“)

^{66}Legend: (address) see the paragraph Address on page 16.

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

```
*B1AS4↵
```

Response

```
*B10↵
```

[9] 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.)

^{66}Enquiry: „SS“(code)[9] (Speed Set)

^{66}Response: (ACK „0“)

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

^{66}*Example:* *Enquiry: Speed 19200Bd*

`*B1SS7↵`

Response

`*B10↵`

Communication Parameters Reading

Description: Returns the address and communication speed.

^{97}Enquiry: F0H

^{97}Response: (ACK 00H) (adr) (speed)

^{97}Legend: (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

^{97}Notes: This instruction is designed for the detection of the set address of the device in case it is unknown. The enquiry 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

Configuration Acknowledgement

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.

97Enquiry: E4H

97Response: (ACK 00H)

97Example: Configuration acknowledgement

`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: Enquiry

`*B1E↵`

Response

`*B10↵`

Status Setup

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

97Enquiry: 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`

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

66Response: (ACK „0“)

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

66Example: Enquiry – 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}Enquiry: 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}Enquiry: „SR“ (Status Read)

^{66}Response: (ACK „0“)(character)

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

^{66}*Example: Enquiry*

```
*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 TQS3 97 and 66 modules). Set by the manufacturer.

^{97}Enquiry: F3H

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

^{97}Legend: (string) Text in the form of: „*TQS3; v0199.04.02; F66 97*“.

^{97}*Example: Enquiry*

```
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}Enquiry: „?“

^{66}Response: (ACK „0“)

^{66}*Example: Enquiry*

```
*B1?↵
```

Response

```
*B10TQS3; V0199.01; F66 97↵
```

Reset

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

^{97}Enquiry: 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}Enquiry: „RE“ *(REset)*

^{66}Response: (ACK „0“)

^{66}*Example:* *Enquiry*

```
*B1RE↵
```

Response

```
*B10↵
```

Check Sum Acknowledgement

Description: Enables the verification of check sum in the incoming messages. This instruction must be immediately preceded by the instruction of Configuration Acknowledgement (see page 22).

^{97}Enquiry: EEH (status)

^{97}Response: (ACK 00H)

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

^{97}*Example:* *Configuration acknowledgement*

```
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}Enquiry: FEH

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

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

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

```
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.

^{97}Enquiry: E2H (position)(data)

^{97}Response: (ACK 00H)

^{97}Legend: (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.

^{97}*Example: 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.

^{66}Enquiry: „DW“(position)(data) *(Data Write)*

^{66}Response: (ACK „0“)

^{66}Legend: (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.

^{66}*Example: Enquiry*

```
*B1DW0BOILER ROOM 1↵
```

Response

```
*B10↵
```

Saved User Data Reading

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

^{97}Enquiry: F2H

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

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

^{97}*Example: 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
```

^{66}Enquiry: „DR“ *(Data Read)*

^{66}Response: (ACK „0“)(data)

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

^{66}*Example: Enquiry*

```
*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}Enquiry: 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 ID from Dallas sensor

Description: Instruction ready unique ID from the temperature sensor. This ID cannot be changed and is hard coded in the sensor during manufacturing process.

^{97}Enquiry: A0H

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

^{97}Legend: (status) 1 byte; 00H = error, 01H = reading in progress, FFH = follows ID

(id) 8 byte; unique ID

^{97}*Example:* *read communication errors; address 01H, signature 02H*

```
2AH,61H,00H,05H,31H,02H,A0H,9CH,0DH
```

Response

```
2AH,61H,00H,0EH,31H,02H,00H,FFH,28H,00H,00H,07H,9DH,60H,A0H,55H,13H
,0DH
```

Read RAW value

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

^{97}Enquiry: 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}Enquiry: 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 TQS3 thermometer it is always decimally: 199, thus hexadecimally: 00C7.

(serial-number) 2 bytes; the TQS3 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: Enquiry – 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 TQS3 thermometer

^{97}Enquiry: FAH

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

^{97}Legend: (product-number) 2 bytes; product number; for TQS3 thermometer it is always decimally: 199, thus hexadecimally: 00C7.

(serial-number) 2 bytes; the TQS3 thermometer serial number is indicated on the label after the 0199.01/ text.

(manufacturing-data) 4 bytes

^{97}*Example: Enquiry*

```
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 Configuration Enabled described on page 22.)

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

^{97}Enquiry: 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: Enquiry*

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

Response

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

TECHNICAL PARAMETERS

Electronics operating temperature	-40 °C to +85 °C
Sensor temperature range	-55 °C to +125 °C; resolution 0.1 °C
Accuracy	± 0.5 °C between -10°C and +85°C, otherwise ± 2 °C
Measuring unit	DS18B20
Termination	No; only 10 kΩ resistances defining the line standby state
Supply voltage	7 V to 20 V DC with protection against polarity reversal
Consumption	typically 2 mA, 3 mA maximum (see note 7 on page 19)
Lines connection	Wago 236 terminal block

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
Number of data bits	8
Parity	none
Number of stop-bits	1

Available Designs

Outdoor Design – TQS3 O

This design is suitable for measurements in outdoor environments, places exposed to water or various weather conditions.

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 gland
Lines connection.................................	Wago 236 terminal block

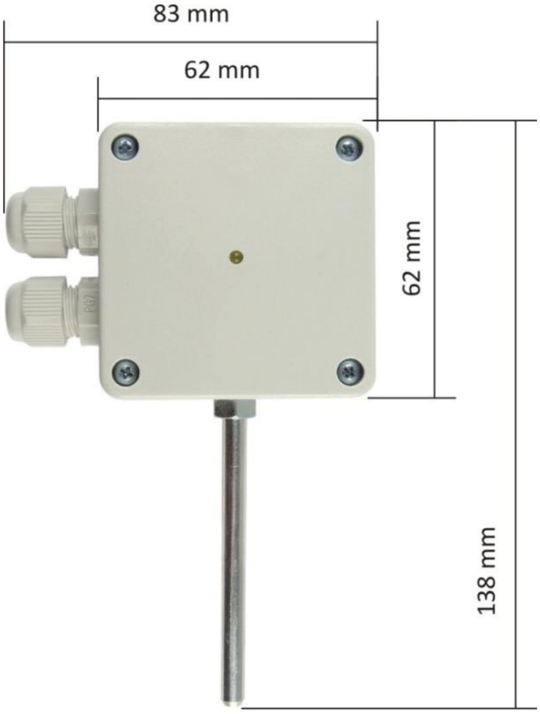


Figure 1 – Outdoor Design

Indoor Design – TQS3 I

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

Housing type......................................	IP20
Sensor mechanical design.................	Plastic sensor 5x5x5 mm
Dimensions..	62 mm x 62 mm x 28 mm

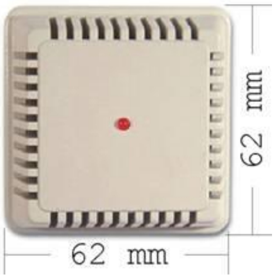


Figure 2 – Indoor Design

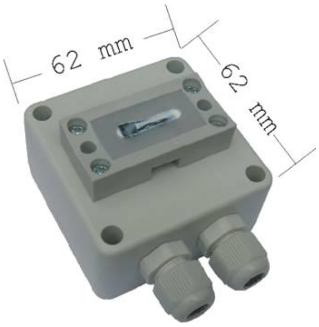
Surface Design – TQS3 P

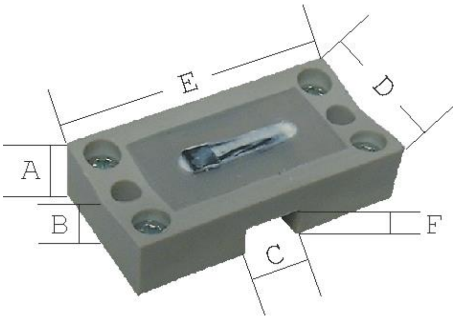
Design for measuring the temperature of pipes or other curved objects.

Housing type IP65

Sensor mechanical design .... To be placed on pipes

Dimensions .... 62 mm x 62 mm x 45 mm





A = 11,5 mm
B = 10 mm
C = 9,5 mm
D = 24 mm
E = 48 mm
F = 2,5 mm

Figure 3 – Surface Design

Figure 4 – Application Part Detail

Board with Electronics – TQS3 E

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

Board dimensions .... 35(51) mm x 36 mm x 15 mm

Holes location .... in the corners of 28(44,4) mm x 28 mm rectangle; 3 mm diameter

Sensor mechanical design .... plastic sensor 5 mm x 5 mm x 5 mm

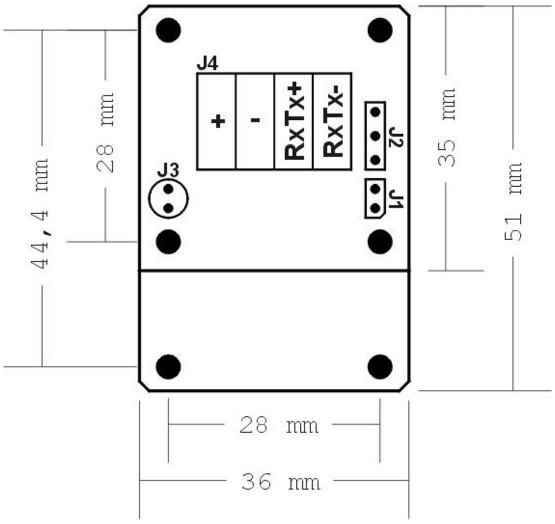


Figure 5 – Board with Electronics Drawing

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

Fax:

+420 267 314 269

Internet:

www.papouch.com

E-mail:

info@papouch.com