THCO2

CO_2 level sensor,

thermometer and hygrometer

Communication: RS485 line, Modbus or Spinel

THCO2

Datasheet

Created: 9.12.2019

Last update: 22.05.2020 10:44

Number of pages: 28

© 2020 Papouch s.r.o.

Papouch s.r.o.

Address:

Strasnicka 3164
102 00 Prague 10
Czech Republic

Tel:

+420 267 314 267

Internet:

www.papouch.com

E-mail:

info@papouch.com

TABLE OF CONTENTS

Changelog

Version 1

- First version.

BASIC INFORMATION

THCO2 is a CO_2 level sensor that also measures temperature and humidity. It measures from -40°C to +70°C and CO_2 level from 0 to 40 000 ppm. Communication is possible on an RS485 line using Spinel or ModBus RTU protocols. All of the above is possible with very little power consumption. The make of THCO2 makes it ideal to measure air quality in lecture halls, offices and all indoor areas where CO_2 level needs to be monitored.

THCO2 is available as an interior sensor for areas protected from water and condensation.

Areas of use

- Ventilation automation
- Air quality control
- Ventilation to comply with legislation guidance.

Features

- **Measurement of the CO_2 level from 0 to 40 000 ppm**
- CO_2 measurements carried out by optical NDIR sensor [1]
- CO_2 level indicator on the device
- **Temperature measurement** in range from **-40 to +70 °C** [1]
- **Humidity measurement** from 0 to 100 %RH [1]
- Automatic dew point calculation
- Measured-out values transmitted directly in ppm, degrees Celsius a percent
- RS485 communication
- **Power voltage from 4,5 to 36 V**
- Low **consumption** – typically **only 8 mA @ 12 V**
- Communication protocols: **Spinel** or **Modbus** RTU *(user selectable)*

fig. 1 – comparison of the THCO2 against an AA battery

[1] Exact accuracy shown in technical parameters section at the end of the document.

INDICATIONS

An indicator is visible in one of the corners through the vents. It indicates several basic states using different colors. Different modes of the indicator can be selected using both communication protocols.

After a power-up:

- Long green flash – indicates the set protocol is Spinel
- Long red flash – indicates the set protocol is ModBus RTU

During operation – mode 1 (default):

- green: is lit when CO_2 level is below 1000 ppm
- red: is lit when CO_2 level is from 1000 to 1500 ppm
- red: flashes when CO_2 concentration is above 1500 ppm [2]
- The indicator dims shortly when the device is communicating in this mode [3]

During operation – mode 2:

- green: long flash during the communication[3] or short flash once every 10 seconds to indicate the operation
- red: flashing as indication of a sensor fault

During operation – mode 3:

- green: off
- red: is lit only as an indication of a sensor fault

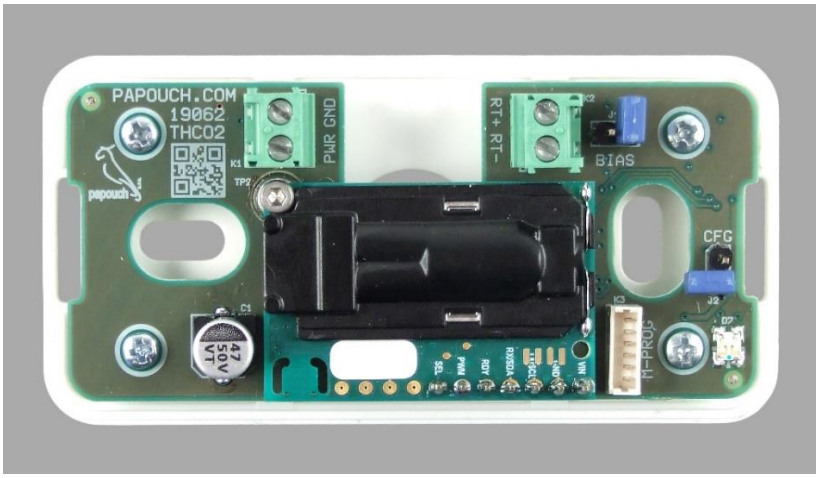


fig. 2 – indicator is at the bottom right corner of the PCB inside the enclosure

SENSOR PLACEMENT

We recommend placing the sensor on a wall at eye level. Universal recommended height is approximately 1.5m (5 ft). Sensor should not be near any heat source, in a cold spot or in direct sunlight. These can affect the measurements.

[2] „The **UK standards** for schools say that carbon dioxide in all teaching and learning spaces, when measured at seated head height and averaged over the whole day should **not exceed 1,500 ppm**." (Quote from Wikipedia)

[3] Communication means receiving an instruction meant for THCO2 device.

CONNECTIONS

THCO2 communicates using a standard two-wire RS485 industrial bus. It uses 4,5 – 36V DC for power, its power input is protected against polarity reversal.

Screw terminals are used both for power and RS485 line. (2.5 mm flat screwdriver)

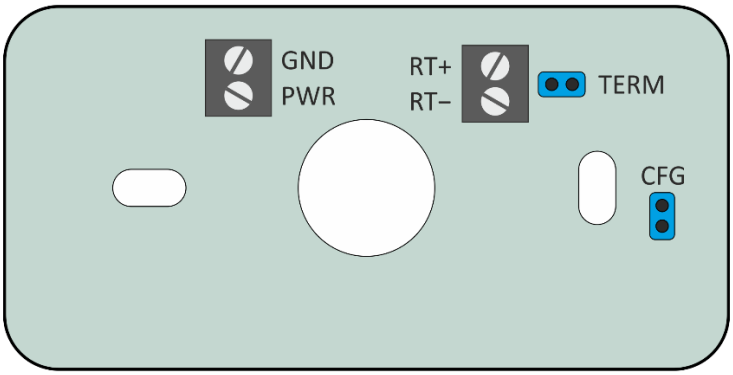


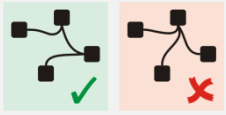
fig. 3 – terminal and jumper placement diagram

PWR and GND terminals are for power, RT+ and RT- are for **RS485 line.** [4]

Connecting the RS485 line

Some basic recommendations for connecting RS485 line (bus):

- We recommend using common TP cable used for computer network and using one twisted pair as RS485 communication cables
- All devices should be connected from one to another and not in a star topology (see picture on the right). Maximum length of such line should be 1,2 km
- Use termination at each end of the line (use Termination jumper if available).
- Shielding, if available, connect to the ground on one side of the line 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-.

[4] Alternative designations are used for RS485 wires: „A“ or „RxTx+“ (for RT+) and „B“ or „RxTx-“ (for RT-).

The following picture shows THCO2, a converter and other devices all connected properly.

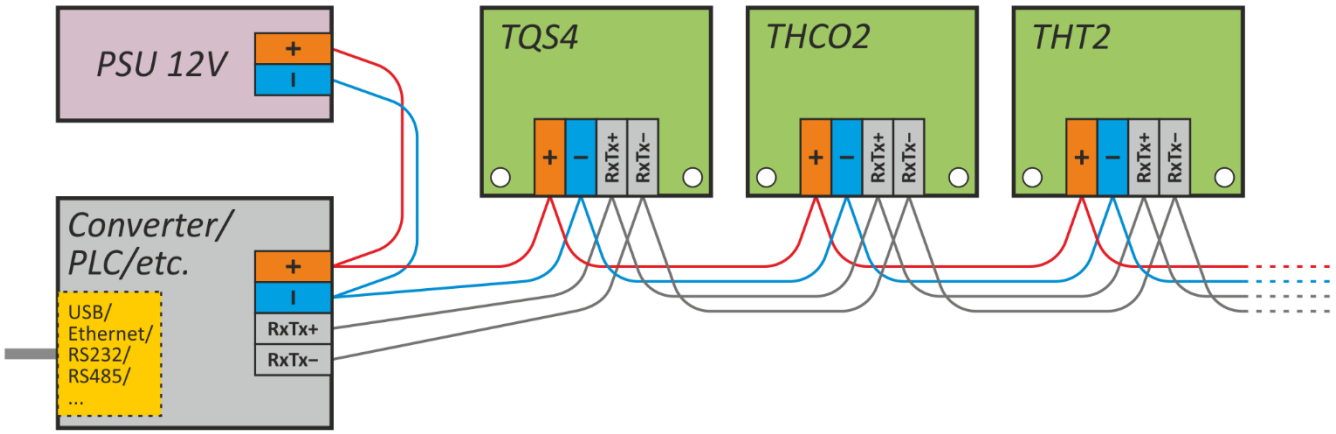


fig. 4 – an example of a proper connection of RS485 line

COMMUNICATION PROTOCOLS

Default communication parameters are following:

Speed .....9600 Baud
Communication protocol.....Spinel
Address .....31H („1“) [5]
Data bits 8
Parity .....none
Stop bits.....1

Spinel

The basic communication protocol is Spinel in the text (ASCII) version (Spinel 66 format) and binary version for machine use (Spinel 97 format). Spinel documentation starts on page 12. [5]

Default protocol set in the device is Spinel. Use Modbus Configurator utility to **switch between Modbus RTU and Spinel protocols.** it is available on papouch.com.

Modbus RTU

Standardized industrial protocol.

Default protocol set in the device is Spinel. Use Modbus Configurator utility to **switch between Modbus RTU and Spinel protocols.** it is available on papouch.com.

Fast switch to ModBus

THCO2 can also be switched to ModBus RTU using the jumper CFG (see fig. 3 on page 6).

If the THCO2 is set to Spinel protocol (default) and the CGF jumper short is detected upon power-up, THCO2 switches to Modbus RTU protocol regardless of the settings. [5]

[5] Address for Spinel protocol is in a different memory space than ModBus RTU address. Both are set to „1“ (hexadecimal: 31H, decadic: 49) in default. Change on one protocol will not affect the other protocol. (Each protocol has different addressing rules.)

MODBUS RTU PROTOCOL

We recommend using ModbusConfigurator to set-up address and communication parameters.

Address

- 0x31: Default device address (49 decadic). Address can be changed in register1 (see below).
- 0x00: Universal address of ModBus RTU protocol (0 decadic). When the device receives a request with this address, an instruction will be done, but the device will not respond.
- 0xF8: Universal device address (248 decadic). When the device receives a request with this address, an instruction will be done and the device will respond. This can only be used when a single device is on the line!

Changing the address using serial number

Thanks to the following method, multiple devices with the same address can be connected to the same RS485 line and have their address changed one by one:

1) Write down serial numbers of your devices. These are on a label on the side of the device in format *1395/0069*
   The first number is Product type, the second is its serial number.
2) Using function code 0x10 and universal address 0xF8 write these holding registers to the device at once:
   a. *Product type* (address 10) – write product type according to the label.
   b. *Serial number* (address 11) – write serial number according to the label.
   c. *Address* (address 12) – write the new address you want to set.
3) Your device now has the new address.

Function codes list

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

- 0x03 .....read holding register
- 0x04 .....read input register
- 0x06 .....set one holding register
- 0x10 .....write to multiple holding registers
- 0x11 .....identification

Device identification

Read identification string from the device (Report slave ID).

Function codes:

0x11 – Report slave ID

Parameters:

No. of bytes	1 Byte	According to the string
ID	1 Byte	ID is the same as device address
RI	1 Byte	Run Indicator – always 0xFF (powered on)
Data	N Byte	String identical to Spinel protocol string, for example: *THCO2; v1395.01.01; f97 fModbus*

Holding Register

Address	*Access*	*Function*	*Designation*
0 [6]	write	0x06	**Allow configuration** Writing value0x00FF to this memory space must precede all instructions writing to holding registers 0 to 5. This is to protect the device from accidental changing its settings. Function code 0x10 along with other parameters is not allowed to allow configuration.
1	read, write	0x03, 0x06, 0x10	**Address (ID)** [7] Unique address of the device in ModBus protocol. Expected number is from 1 to 247. Address is unique for ModBus protocol. *Default address is 0x0031.* See page 8 to set up address using devices serial number.
2	read, write	0x03, 0x06, 0x10	**Communication speed** [7] Speeds and their respective codes: 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** [7] Data word is always 8-bit. **Value** / **Parity** / **Stop bits** *0x0000 (default)* / *none (N)* / *1* 0x0001 / even (E) / 1 0x0002 / odd (O) / 1
4	read, write	0x03, 0x06, 0x10	**End packet delay** [7] Determines length of the delay between packets. Delay is determined as number of bytes. It can range from 4 to 10, default value is 10.

[6] Some manufacturers use register numbering from 1 while other start from 0 because the first register has an address of 0.

[7] This write has to be preceded by writing 0x00FF into register 0 to Allow configuration position. This is to protect the device from accidental writing into settings registers. Function code 0x10 along with other parameters is not allowed to allow configuration.

Address	*Access*	*Function*	*Designation*
5	read, write	0x03, 0x06, 0x10	**Communication protocol** [7] This register allows to switch the device to Spinel protocol. After the device responds, it sets itself to Spinel protocol and no longer communicates on ModBus protocol. (Each protocol has an instruction to switch the protocol to the other one.) Spinel protocol code: *0x0001 (default)* ModBus RTU protocol code: 0x0002 If the CFG jumper is shorted on the PCB the device will communicate using ModBus RTU regardless of this register!
6	read, write	0x03, 0x06, 0x10	**Indicator mode** Indication mode for the internal LED indicator. Number from 1 to 3. Indication modes are described in In section on page 5.
10	read, write	0x03, 0x10 [8]	**Product type** This is always 1395 as an identification of the product.
11	read, write	0x03, 0x10 [8]	**Serial number** Unique serial number.
12	read, write	0x03, 0x10 [8]	**Address** See changing the address using serial number on page 8.
16	write	0x06, 0x10	**CO_2 sensor calibration** User calibration at 400 ppm CO_2 can be performed by writing a number 400 in this register. Place the sensor for at least 5 minutes outside or to a place with a CO_2 level of 400 and then write 400 in this register. (Sensor does not require calibration from after purchase.)
99	read	0x03	**Status** 0x0000 … Values are valid (CO2 does not have to be settled after a power-up – see value *Time since power-up in seconds in* reg. 5) Other… Values are invalid (registers do not contain current values)
100	read	0x03	**CO_2 level** Value as a positive integer directly in ppm.
101	read	0x03	**Temperature** This value (signed integer[9]) can be calculated to temperature: *temperature = value / 10* Temperature then has a resolution of 0.1°C.

[8] Registers 10 through 12 have to be written to at once. Writing to these registers only serves to change the device address, product type and serial number will not be re-written (see page 8).

[9] Negatives are in the form of two's complement. Detailed explanation can be seen for example on Wikipedia: Two's complement. You can use windows scientific calculator to convert those numbers. *Example:* Temperature -13,8 °C is represented as a number -138 (decimal), which is FF76H.

Address	*Access*	*Function*	*Designation*
102	read	0x03	**Humidity** Positive integer ranging from 0 to 1000. Calculation: *humidity = value / 10* Humidity then has a resolution of 0.1 %
103	read	0x03	**Dew point** This value (signed integer[9]) can be calculated into dew point: *dew_point = value / 10* Dew point then has a resolution of 0.1°C
104	read	0x03	**Time since power-up in seconds** CO_2 level reading becomes accurate after several tens of seconds from power-up. Time from this registry can be used to see how long since the power up it has been. It is a whole number from 0 to 3600. (The timer stops after reaching 3600.)
105–109	read	0x03	Always 0xFFFF (reserved for future use)

Input Register

Input register contains same values as holding registers starting from address 99 except these are **read-only**. This is to allow devices that can only use function code 0x04 to read from THCO2.

Address	*Access*	*Function*	*Designation*
0	read	0x04	**Status** 0x0000 … Values are valid (CO2 does not have to be settled after a power-up – see value *Time since power-up in seconds in* reg. 5) Other… Values are invalid (registers do not contain values)
1	read	0x04	**CO_2 level** Value as a positive integer directly in ppm.
2	read	0x04	**Temperature** This value (signed integer[9]) can be calculated to temperature: *temperature = value / 10* Temperature then has a resolution of 0.1 °C.
3	read	0x04	**Humidity** Positive integer ranging from 0 to 1000. Calculation: *humidity = value / 10* Humidity then has a resolution of 0.1 %
4	read	0x04	**Dew point** This value (signed integer[9]) can be calculated to dew point: *dew_point = value / 10* Dew point then has a resolution of 0.1 °C
5	read	0x04	**Time since power-up in seconds** CO_2 level reading becomes accurate after several tens of seconds from power-up. Time from this registry can be used to see how long since the power up it has been. It is a whole number from 0 to 3600. (The timer stops after reaching 3600.)

SPINEL COMMUNICATION PROTOCOL

THCO2 has Spinel protocol implemented in format 97 (binary).

Format 97

Structure

Request:

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

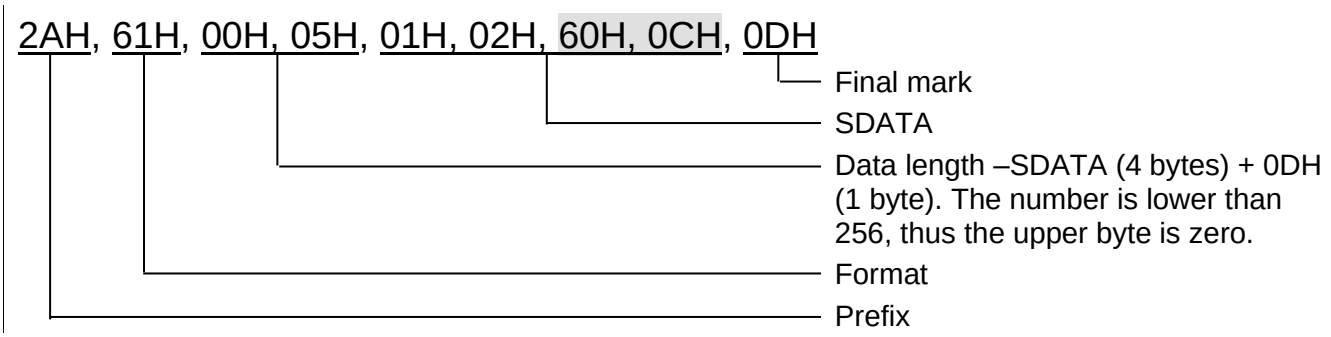
Response:

```
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[10]	Instruction code – Module instructions are described in great detail in chapter Instruction overview on page 14.
ACK	Request acknowledgement of whether and how it was executed. ACK can be 00H to 0FH.
DATA[10]	Data. They are described in great detail in chapter Instruction overview on page 14 for each instruction.
SUMA	Check sum.
CR	Final mark (0DH).

Explanatory notes

Example



[10] 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.)

Instruction overview

To make the instructions more legible, only Instruction (INST) portion is shown. Acknowledge (ACK), Address (ADR), signature (SIG) and CheckSum (SUMA) are described elsewhere.

Basic instructions

Single measurement

Description: This does a single measurement of CO_2, temperature and humidity.

Request: 51H

Response: (ACK 00H) (status) (co2) (temperature) (humidity) (dew-point) (time)

Legend:
(status) 1 byte, 0 = values are valid; 1 = waiting for first measurement; 2 a 3 = one of values is out of range; 4 = sensor fault
(co2) 2 bytes, CO_2 level as a positive integer
(temperature) 2 bytes, temperature multiplied by 10 in signed int format
(humidity) 2 bytes, humidity multiplied by 10 as a positive integer
(dew-point) 2 bytes, temperature multiplied by 10 in signed int format
(time) 2 bytes, see information about register 5 on page 11

Example: *Request*

2AH,61H,00H,05H,31H,02H,51H,EBH,0DH

Response

2AH,61H,00H,0FH,31H,02H,00H,04H,BBH,01H,3CH,00H,C1H,00H,33H,0EH,10H,24H,0DH

Status: 00H, all lis ok

CO_2: 016FH = 367 decadic = 367 ppm

Temperature: 0104H = 260 decadic = 26.0 °C

Humidity: 00DDH = 221 decadic = 22.1 %RH

Dew point: 001AH = 26 decadic = 2.6 °C

Time since power-up: 0038H = 56 decadic = 56 sec from startup

Single measurement - strings

Description: This does a single measurement of CO_2, temperature and humidity. Unlike the previous instruction, this one returns values as strings.

Request: 58H (quantity)

Response – all: (ACK 00H) (status) (co2) (temperature) (humidity) (dew-point) (time)

Response – one value: (ACK 00H) (status) (quantity) (time)

Legend:
(quantity) 1 byte; all quantities are sent if set to 0, 1 – 4 only sends the given quantity
(status) 1 byte; 0 = values are valid
(co2) 10 byte, CO_2 level as a string aligned to right
(temperature) 10 byte, temperature with a decimal place as a string aligned right
(humidity) 10 byte, humidity with a decimal place as a string aligned right
(dew-point) 10 byte, temperature with a decimal place as a string aligned right
(time) 10 byte, string, see information of register 5 on page 11

Example: *Request*

2AH,61H,00H,06H,31H,02H,58H,01H,E2H,0DH

Response

2AH,61H,00H,1AH,31H,02H,00H,00H,20H,20H,20H,20H,20H,20H,20H,38H,30H,39H,20H,20H,20H,20H,20H,20H,20H,20H,34H,52H,0DH

Status: 00H

CO_2: 20H,20H,20H,20H,20H,20H,20H,38H,30H,39H = 809 ppm

Time since power-up: 20H,20H,20H,20H,20H,20H,20H,20H,20H,34H = 4 sec

Configuration instructions

Setting communication parameters

Description: Sets address and communication speed (baudrate). Allow configuration (see page 17) instruction must precede this instruction. Universal address or broadcast address cannot be used with this instruction.

Request: E0H (address) (speed)

Response: (ACK 00H)

Legend: (address) 1 byte; Can range from 00H to FDH

(speed) 1 byte; communication speed, speed codes shown in tab. 1.

Example: Set address 04H and communication speed 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: New address a communication speed sets after sending the response.

Allow configuration (see page 17) instruction must precede this instruction. Allow configuration is disregarded when those new settings are written.

Other communication parameters are: 8 bits, no parity, 1 stop bit. Default communication speed is 9600Bd, default address is 1 (31H).

In case the address is unknown and there is a single device on the line, address can be found using the instruction Read communication parameters. Use universal address FEH for this instruction

In case the communication speed is unknown, all communication speeds must be tested in order to find the set one.

Read communication parameters

Description: This instruction reads address and baudrate.

Request: F0H

Response: (ACK 00H)(Address)(baudrate)

Legend: (Address) 1 byte; Devices address

(baudrate) 1 byte; Comm. speed codes are described in tab. 1.

Example: Read communication parameters; universal address FEH, signature 02H

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

Response – Address 04H, comm. speed 9600Bd

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

Communication speed Bd	Code
1200	03H
2400	04H
4800	05H
9600	06H
19200	07H
38400	08H
57600	09H
115200	0AH

tab. 1 – comm. speed codes

Notes: Use this instruction once the address is unknown. Request is sent to the universal address FEH. If the communication speed is unknown as well, all communication speeds must be tried. This only works in a single device is connected to the line.

Other communication parameters are: 8 bits, no parity, 1 stop bit. Default communication speed is 9600Bd, default address is 1 (31H).

Set indicator mode

Description: Sets indicator mode to one of three options.

Request: E5H (mode)

Response: (ACK 00H)

Legend: (mode) 1 byte; number 01H to 03H; More is described in In section on page 5.

Example: *Request*

2AH,61H,00H,06H,31H,02H,E5H,02H,54H,0DH

Response

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

Read indicator mode

Description: Reads currently set indicator mode.

Request: F5H

Response: (ACK 00H)(mode)

Legend: (mode) 1 byte; number 01H to 03H; More is described in In section on page 5.

Example: *Request*

2AH,61H,00H,05H,31H,02H,F5H,47H,0DH

Response

2AH,61H,00H,06H,31H,02H,00H,02H,39H,0DH

Other

Allow configuration

Description: This instruction allows the user to make configuration changes. It has to precede some instruction to set communication parameters. It is invalidated after a following instruction (even if said instruction is invalid) and configuration is automatically disabled. (Universal address cannot be used with this instruction.)

Request: E4H

Response: (ACK 00H)

[7]*Example:* *Allow configuration*

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

Response: *2AH,61H,00H,05H,01H,02H,00H,6CH,0DH*

CO_2 sensor calibration

Description: This instruction performs user calibration at CO_2 level of 400. Place the sensor for at least 5 minutes outside or to a place with a CO_2 level of 400 and then use this instruction. (Sensor does not require calibration from after purchase.) Allow configuration (see page 17) instruction must precede this instruction.

Request: 1EH

Response: (ACK 00H)

Example: *Request:*

2AH,61H,00H,05H,31H,02H,1EH,1EH,0DH

Response:

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

Set status

Description: Sets device status. This is user-defined byte used to get the device state. User can freely set this byte at any time.

Request: E1H (status)

Response: (ACK 00H)

Legend: (status) 1 byte; device status. After a power cycle (and even software reset) byte is set to 00H. If it was set using the instruction Set status before, it is then easy to identify the device state.

Example: *Set status 12H; address 01H, signature 02H*

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

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

Read status

Description: Reads user defined status byte.

[97]Request: F1H

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

[97]Legend: (status) 1 byte; device status, meaning – see “Description: This instruction performs user calibration at CO_2 level of 400. Place the sensor for at least 5 minutes outside or to a place with a CO_2 level of 400 and then use this instruction. (Sensor does not require calibration from after purchase.) Allow configuration (see page 17) instruction must precede this instruction.

Request: 1EH

Response: (ACK 00H)

Example: *Request:*

`2AH,61H,00H,05H,31H,02H,1EH,1EH,0DH`

Response:

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

Set status” instruction.

[97]*Example:* *Read status; 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`

Read name and version

Description: Reads device name, internal software version and a list of possible communication formats (only format 97 in case of THCO2). These settings are factory set.

Request: F3H

Response: (ACK 00H) (string)

Legend: (string) Text in format: „*THCO2; v1395.01.01; f97 fModbus“*.

Example: *Request*

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

Response

2AH,61H,00H,24H,31H,02H,00H,54H,48H,43H,4FH,32H,3BH,20H,76H,31H,33H,39H,35H,2EH,30H,31H,2EH,30H,31H,3BH,20H,66H,39H,37H,20H,66H,4DH,6FH,64H,62H,75H,73H,DBH,0DH

Reset

Description: This instruction resets the device. It returns to the exact same state it was after a power-up.

Request: E3H

Response: (ACK 00H)

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 a response is sent.

Allow checksum

Description: Allows checksum control for incoming instructions. Allow configuration (see page 17) instruction must precede this instruction.

Request: EEH (state)

Response: (ACK 00H)

Legend: (state) 1 byte; 01H to turn checksum ON; 00H to turn it OFF

Example: Request

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

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

Read checksum settings

Description: This instruction reads the current checksum setting.

Request: FEH

Response: (ACK 00H) (state)

Legend: (state) 1 byte; 01H to switch checksum on; 00H to switch it off

Example: Request setting state

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

Response – checksum on

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

Save user data

Description: Instruction saves user data. Data is stored in non-volatile memory.

Request: E2H(position)(data)

Response: (ACK 00H)

Legend: (position) 1 byte; address of the memory where the data will be saved. Number ranging from 00H to 0FH.

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

Example: Save words "Basement 1" to memory address 00H; Address 01H, signature 02H

2AH,61H,00H,0FH,01H,02H,E2H,00H,"BASEMENT 1",61H,0DH

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

Notes: Memory for user data is 16 bytes. In case data is written for example to 0CH, maximum of 4bytes can be saved.

Read saved user data

Description: Instruction reads the saved user data. Data is stored in non-volatile memory.

Request: F2H

Response: (ACK 00H)(data)

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

[7]*Example: Read user data; Address 01H, signature 02H*

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

Response - "Basement 1 "

```
2AH,61H,00H,15H,01H,02H,00H,"BASEMENT 1        ",5DH,0DH
```

Read communication errors

Description: Instruction reads the number of communication errors that occurred since the device was powered up or since the last use of instruction read communication errors.

Request: F4H

Response: (ACK 00H) (errors)

Legend: (errors) 1 byte; Number of errors of communication since the device power up or last read. Communication errors can be following events:

Another byte comes instead of prefix

Checksum (SUMA) does not match

Incomplete message

Example: Read communication errors; 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

Address Setup using Serial Number

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

Request: EBH (new-address)(product-number)(serial-number)

Response: (ACK 00H)

Legend:

(new-address)	1 byte; new address of the module.
(product-number)	2 bytes; product number; for THCO2 thermometer it is always decimally 1395 or 0573 hexadecimally.
(serial-number)	2 bytes; THCO2 serial number is indicated on the label after the 1395/ text. This number can also be found out via the Manufacturing Data Reading instruction.

Example: Request – new address: 32H, product-number: 1395 (= 0573H), serial number: 101 (= 0065H)

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

Response – THCO2 already responds with the new address

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

Read factory data

Description: Instruction reads saved factory data from device.

Request: FAH

Response: (ACK 00H)(device-number)(serial-number)(factory-data)

Legend: (device-number) 2 bytes; device number; this is always 1395 decadic for THCO2 or 0573 hexadecimally.

(serial-number) 2 Bytes; serial number

(factory-data) 4 Bytes

Example: Request

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

Response – device number 1357 (=0573H), serial number 101 (=0065H), factory data 20050923H

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

Switching the communication protocol

Description: This instruction switches between the types of the communication protocols. Allow configuration (see page 17) instruction must precede this instruction.

Modbus Configurator available on papouch.com can also be used to switch protocols.

[97]Request: EDH (id)

[97]Response: (ACK 00H)

[97]Legend: (id) 1 byte; protocol identification number:
01H – Spinel protocol, format 97 (binary)
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

CO_2 level measurements

Range	400 to 10 000 ppm (0 to 40 000 ppm)
Accuracy [11]	±(30 ppm + 3% measured value) (at 25 °C, 400 to 10 000 ppm, humidity 0 to 95 %)
Repeatability	±10 ppm
Temperature stability	±2.5 ppm / °C (0 to 50 °C)
Measurement delay ($\tau_{63\%}$)	20 s
sensor type	optical (NDIR)

Temperature measurements

Range	10 to 30 °C (-40 to +70 °C)
Resolution	0.1 °C
Accuracy [11]	±1 °C in range 10 to 30 °C

Humidity measurement

Range	0 – 100 %RH (non-condensing)
Resolution	0.1 %RH
Accuracy [11]	±3% RH
Measurement delay ($\tau_{63\%}$)	8 s

Communication line

Type	RS485
Termination	resistor 120 Ω user connected using TERM jumper
BIAS-ing	resistors 22 kΩ
Response delay	2.5 ms
Communication protocol	Spinel or Modbus RTU *(user selectable)*
Default communication protocol	Spinel
Speed	to 115.2 kBd (default: 9600 Bd)
Default address	31H (character: „1“, decadic: 49) [5]
Data bits	8
Parity	none
Stop bits	1

Mechanical parameters

IP coverage	IP 20
Mechanical sensor placement	directly on the PCB
Dimensions	80 × 40 × 20.5 mm
Connections	screw terminal for wires 0.2 - 1 mm^2
Terminal tool	flat-head screwdriver 0.5 × 2.5 mm

[11] Accuracy can also be negatively affected by heat and cold sources, air circulation and direct sunlight.

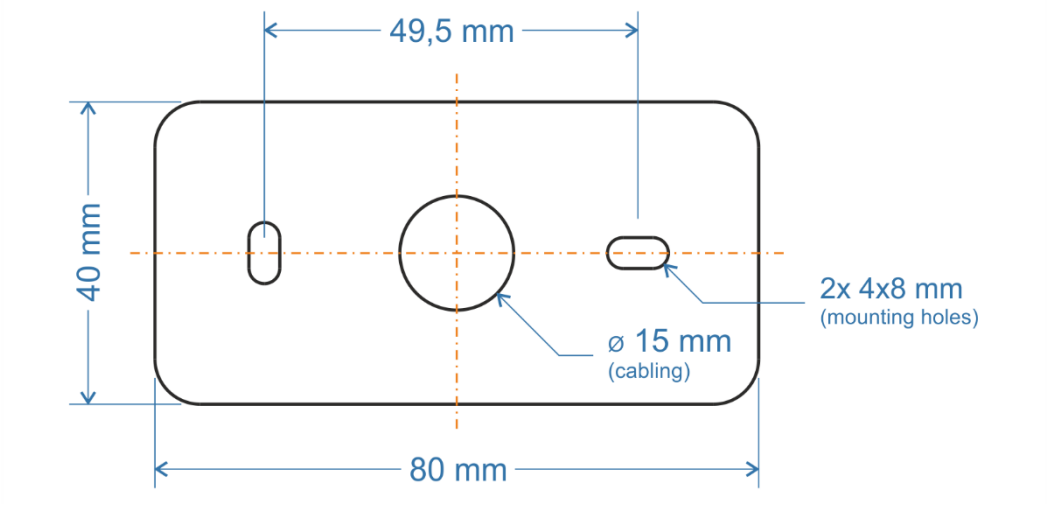


fig. 5 – mount holes and central opening for wiring

Other parameters

Power voltage ..... 4.5 to 36 V DC with polarity reversal protection

Current draw at 12 V ..... typ. 8 mA; max. 25 mA

Current draw at 24 V ..... typ. 4 mA; max. 13 mA

Operation temperatures ..... -40 to +70 °C

Sensor lifetime ..... 15 years

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:

www.papouch.com

E-mail:

info@papouch.com