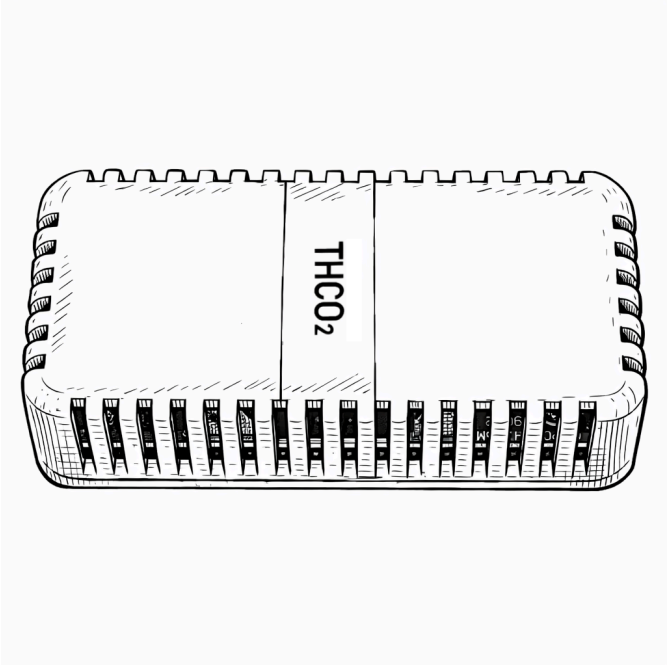
THCO2

CO2, Temperature and Humidity Sensor

Communication Protocols Spinel and Modbus RTU



Version 2.4

March 6, 2026

Table of Contents

1 Changelog ... 3
2 Communication Parameters ... 4
2.1 Hexadecimal Value Notation ... 4
2.2 Spinel ... 4
2.3 Modbus RTU ... 4
2.4 Fast Switch to Modbus ... 4
3 Device Behavior ... 5
3.1 Button Behavior ... 5
3.2 Sound Indication ... 5
3.3 LED Indication Modes ... 5
4 Modbus RTU Communication Protocol ... 6
4.1 Address ... 6
4.1.1 How to Change the Address Using Serial Number? ... 6
4.2 Function Codes List ... 6
4.3 Device Identification ... 6
4.4 Holding Register ... 7
4.5 Input Register ... 9
5 Spinel Communication Protocol ... 10
5.1 Format 97 ... 10
5.1.1 Structure ... 10
5.1.2 Explanatory Notes ... 10
5.2 Instruction Overview ... 12
5.3 Basic Instructions ... 13
5.3.1 Single Measurement ... 13
5.3.2 Single Measurement – Strings ... 13
5.4 Configuration Instructions ... 14
5.4.1 Setting Communication Parameters ... 14
5.4.2 Read Communication Parameters ... 15
5.4.3 Set Indicator Mode ... 16
5.4.4 Read Indicator Mode ... 16
5.4.5 Write Altitude ... 16
5.4.6 Read Altitude ... 17
5.4.7 Write CO2 Calibration Type ... 17
5.4.8 Read CO2 Calibration Type ... 18
5.4.9 Set Sound Indication ... 18
5.4.10 Read Sound Indication Settings ... 18
5.5 Miscellaneous Instructions ... 19
5.5.1 Allow Configuration ... 19
5.5.2 CO2 Sensor Calibration ... 19
5.5.3 Set Status ... 19
5.5.4 Read Status ... 20
5.5.5 Read Name and Version ... 20
5.5.6 Reset ... 21
5.5.7 Allow Checksum ... 21
5.5.8 Read Checksum Settings ... 21
5.5.9 Save User Data ... 22
5.5.10 Read Saved User Data ... 22
5.5.11 Read Communication Errors ... 22
5.5.12 Address Setup Using Serial Number ... 23

5.5.13 Read Factory Data 23
5.5.14 Switching the Communication Protocol 24

1 Changelog

Version	Date	Description
2.4	2026-03-06	New sensor type. Added instructions and registers for altitude, calibration type and sound indication.
1.1	2020-12-02	First version.

2 Communication Parameters

Default communication line parameters:

Parameter	Value
Speed	9600 Bd
Communication protocol	Spinel
Address	0x31 (character "1", dec. 49)
Data bits	8
Parity	none
Stop bits	1

2.1 Hexadecimal Value Notation

Hexadecimal values in this document are shown with the `0x` prefix (e.g. `0x31`). In tables where the hexadecimal context is obvious and in communication examples, the prefix is omitted (e.g. `31`).

2.2 Spinel

Binary communication protocol for machine use (format number 97). The protocol is described in detail in chapter Spinel Communication Protocol.

The default communication protocol set in the device is Spinel. Use the Modbus Configurator utility to switch between Modbus and Spinel protocols. It is available for download at papouch.com.

2.3 Modbus RTU

Standardized industrial protocol. The protocol is described in detail in chapter Modbus RTU Communication Protocol.

2.4 Fast Switch to Modbus

THCO2 can also be switched to Modbus RTU using the CFG jumper. If THCO2 is set to Spinel protocol (i.e. default setting) and the shorted jumper is detected upon power-up, THCO2 switches to Modbus RTU protocol regardless of the stored settings.

3 Device Behavior

3.1 Button Behavior

Press	Action	Result
Short	Device test	Red LED flashes, device emits a sound
2 seconds	Suppress sound indication	Sound indication is disabled until CO_2 returns to a normal level
5 seconds	Manual calibration	Device enters manual calibration mode and stores 400 ppm as a reference
10 seconds	Restore factory settings	Restores default values (address, speed, protocol, automatic calibration)

3.2 Sound Indication

Situation	Behavior
$CO_2 \leq 1000$ ppm	No sound
CO_2 1000–1500 ppm	1× short beep every 10 minutes
$CO_2 \geq 1500$ ppm	2× short beeps every 10 minutes
Test button confirmation	1× beep
Suppression confirmation	2× beeps
Manual calibration confirmation	3× beeps
Factory reset confirmation	4× beeps

Sound indication can be enabled/disabled via register 7 (Modbus) or instruction `0x66` (Spinel).

3.3 LED Indication Modes

The device has three LED indication modes, configurable via register 6 (Modbus) or instruction `0xE5` (Spinel).

Mode	State	Color	Behavior
Initialization	Modbus protocol	Red	Long flash
Initialization	Spinel protocol	Green	Long flash
Mode 1	$CO_2 \leq 1000$ ppm	Green	Lit
	CO_2 1000–1500 ppm	Red	Lit
	$CO_2 \geq 1500$ ppm	Red	Flashing
Mode 2	Communication	Green	Long flash
	Device activity	Green	Short flash every 10 s
	Sensor fault	Red	Flashing
Mode 3	Sensor fault	Red	Lit
	Normal state	Green	Off

4 Modbus RTU Communication Protocol

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

4.1 Address

Address	Dec.	Description
`0x31`	49	Default device address. Can be changed in holding register 1.
`0x00`	0	Universal Modbus RTU address. The instruction is executed, but the device **does not respond** (broadcast).
`0xF8`	248	Universal device address. The instruction is executed and the device **does respond**. Usable only when a single device is on the line.

4.1.1 How to Change the Address Using Serial Number?

The following procedure allows connecting multiple devices with the same address to an RS485 line and then changing their addresses one by one:

1. Note the serial number on the device label (format *1395/0069* — the number before the slash is the Product type, the number after the slash is the Serial number).
2. Using function code `0x10` and universal address `0xF8`, write the following holding registers to the device at once:
   - **Product type** (address 10) — write the product type from the label.
   - **Serial number** (address 11) — write the serial number from the label.
   - **Address** (address 12) — write the new address.
3. From this point on, the device communicates with the new address.

4.2 Function Codes List

Code	Function
`0x03`	Read holding registers
`0x04`	Read input registers
`0x06`	Set one holding register
`0x10`	Write to multiple holding registers
`0x11`	Device identification (Report Slave ID)

4.3 Device Identification

Read the device identification string (Report Slave ID).

Function code: `0x11`

Legend:

Field	Length [B]	Description
`ID`	1	Same as the device address
`RI`	1	Run Indicator — always `0xFF` (powered on)
`Data`	N	Identification string, e.g.: *THCO2; v1395.02.04; f97 fModbus*

4.4 Holding Register

Registers for device configuration. Writing to registers 1–5 requires prior configuration enable (writing `0x00FF` to register 0).

Address	Access	Function	Name
0[1]	write	`0x06`	**Allow configuration** Writing the value `0x00FF` must precede writes to registers 1–5 and 13–15. This protects the device from accidental configuration changes. *Note: Writing together with other parameters (code `0x10`) is not allowed.*
1	read, write	`0x03`, `0x06`, `0x10`	**Address (ID)**[2] Unique device address on the Modbus bus. Range 1–247. Default: `0x0031` (49).
2	read, write	`0x03`, `0x06`, `0x10`	**Communication speed** [2] `0x0003` = 1 200 Bd `0x0004` = 2 400 Bd `0x0005` = 4 800 Bd `0x0006` = *9 600 Bd (default)* `0x0007` = 19 200 Bd `0x0008` = 38 400 Bd `0x0009` = 57 600 Bd `0x000A` = 115 200 Bd
3	read, write	`0x03`, `0x06`, `0x10`	**Data word** [2] Always eight-bit. `0x0000` = no parity (N), 1 stop bit *(default)* `0x0001` = even parity (E), 1 stop bit `0x0002` = odd parity (O), 1 stop bit `0x0003` = no parity (N), 2 stop bits `0x0004` = even parity (E), 2 stop bits `0x0005` = odd parity (O), 2 stop bits
4	read, write	`0x03`, `0x06`, `0x10`	**End packet delay** [2] Delay between bytes considered as end of packet, in number of bytes. Range: 4–100. Default: 10.
5	read, write	`0x03`, `0x06`, `0x10`	**Communication protocol** [2] After sending the response, the device switches to the selected protocol. `0x0001` = Spinel *(default)* `0x0002` = Modbus RTU *Note: If the CFG jumper is shorted, the device always communicates using Modbus.*

[1] Register numbering starting from one or from zero may be encountered, because this first register has address 0.

[2] Writing to this register must be preceded by writing the value 0x00FF to address 0 (Allow configuration). This protects the device from accidental configuration changes. Writing Allow configuration together with other parameters using multiple write is not allowed.

Address	Access	Function	Name
6	read, write	`0x03`, `0x06`, `0x10`	**Indicator mode** LED indicator mode on the PCB. Range: 1–3. See LED Indication Modes.
7	read, write	`0x03`, `0x06`, `0x10`	**Sound indication** Enable or disable sound indication (beeping). See Sound Indication. `0x0000` = disabled `0x0001` = enabled *(default)*
10	read, write	`0x03`, `0x10`	**Product type**[3] Product type number. For THCO2, this is always 1395.
11	read, write	`0x03`, `0x10`	**Serial number**[3] Unique serial number.
12	read, write	`0x03`, `0x10`	**Address (serial number)**[3]
13	read, write	`0x03`, `0x06`, `0x10`	**Altitude** [2] Altitude for CO2 measurement compensation in meters. Range: 0–3000 m. Default: 0.
14	write	`0x06`, `0x10`	**CO2 sensor calibration** [2] Writing the number 400 calibrates the CO2 sensor to 400 ppm and switches to manual calibration mode (see register 15). Place THCO2 for at least 5 minutes in an environment with 400 ppm CO2 (or outside) before calibrating! *Note: THCO2 is set to automatic calibration from the factory.*
15	read, write	`0x03`, `0x06`, `0x10`	**CO2 calibration type** [2] Selection between automatic and manual CO2 sensor calibration. `0x0000` = automatic *(default)* `0x0001` = manual
99	read	`0x03`	**Status** `0x0000` = measured values are valid (CO2 may not yet be settled after power-up — see reg. 104) `0x0001` = waiting for first measurement other = values in measurement registers are not valid
100	read	`0x03`	**CO2 level** Positive integer directly in ppm (validity per reg. 99).
101	read	`0x03`	**Temperature** Signed integer[4] . Calculation: *temperature = value / 10*. Resolution 0.1 °C.
102	read	`0x03`	**Humidity** Positive integer (0–1000). Calculation: *humidity = value / 10*. Resolution 0.1 %.

[3]Registers 10 to 12 must be written at once. Writing does not overwrite the Product type and Serial number values. Writing to these registers only serves the function of setting the device address using serial number. See changing the address using serial number.

[4]Signed integer means the value can be negative. In Modbus RTU protocol, it is transmitted as a 16-bit number in big-endian format (two's complement). Example: Temperature −13.8 °C is represented as −138 (decimal), which is 0xFF76.

Address	Access	Function	Name
103	read	`0x03`	**Dew point** Signed integer [4]. Calculation: *dew_point = value / 10*. Resolution 0.1 °C.
104	read	`0x03`	**Time since power-up** CO2 reading becomes accurate after several tens of seconds from power-up. Integer 0–3600 (the counter stops after reaching 3600).
105–109	read	`0x03`	Always `0xFFFF` (reserved for future use).

4.5 Input Register

The input register contains **read-only** copies of the same values as holding registers starting from address 99. This duplication is provided so that the data is easily accessible to devices that can only read Modbus using function `0x04`.

Address	Access	Function	Name
0	read	`0x04`	**Status**
1	read	`0x04`	**CO2 level**
2	read	`0x04`	**Temperature**
3	read	`0x04`	**Humidity**
4	read	`0x04`	**Dew point**
5	read	`0x04`	**Time since power-up**

5 Spinel Communication Protocol

THCO2 implements the standardized Spinel protocol, format 97 (binary). Detailed information about Spinel, an online parser, etc. are available at papouch.com/spinel-parser-online.

For C# developers, Spinel.NET is also available on GitHub — a set of libraries and sample applications for working with devices communicating via the Spinel protocol.

5.1 Format 97

5.1.1 Structure

Request:

PRE	FRM	NUM	NUM	ADR	SIG	INST	DATA	SUM	CR

Response:

PRE	FRM	NUM	NUM	ADR	SIG	ACK	DATA	SUM	CR

Field	Description
PRE	Prefix, 0x2A (character `*`).
FRM	Format number 97 (0x61).
NUM	Number of bytes in the instruction from the following byte to the end of the frame.
ADR	Address of the device to which the request is sent or which sends the response.
SIG	Message signature — any number from 0x00 to 0xFF. The SIG from the request is returned in the response, making it easy to match requests and responses.
INST	Instruction code — see chapter Instruction Overview.
ACK	Request acknowledgement — whether and how it was executed. ACK values range from 0x00 to 0x0F.
DATA	Described in detail in chapter Instruction Overview.
SUM	Check sum.
CR	Final mark (0x0D).

5.1.2 Explanatory Notes

5.1.2.1 Example

```
→ 2A 61 00 05 01 02 60 0C 0D
```

Byte(s)	Meaning
`2A`	Prefix
`61`	Format
`00 05`	Number of bytes after NUM through `0D` (inclusive).
Rest	Remaining bytes

Byte(s)	Meaning
`0D`	Final mark

5.1.2.2 Data Length (NUM)

A sixteen-bit value defining the number of bytes until the end of the instruction; the count of all bytes following NUM, up to CR (inclusive). Values range from 5 to 65535. If less than 5, the instruction is considered faulty and is answered (if intended for the given device) with an ACK "invalid data" response.

Calculation:

Add up the number of bytes following both NUM bytes, up to CR (inclusive). Treat the resulting count as a sixteen-bit number. Split it into upper and lower bytes. The first NUM byte is the upper byte, the second NUM byte is the lower byte. (If the byte count is less than 256, the first NUM byte is 0x00.)

5.1.2.3 Address (ADR)

- Address 0xFF is reserved for **broadcast**. If the request contains address 0xFF, the device behaves as if its own address was specified. No response is sent to requests with this address.
- Address 0xFE is the **universal address**. If the request contains address 0xFE, the device behaves as if its own address was specified. In the response, the device provides its actual currently set address. The universal address is only used when a single device is connected to the line.

5.1.2.4 Request Acknowledgement (ACK)

ACK informs the master device about how the received instruction was processed. Acknowledgement codes:

Code	Meaning	Description
`00`	Everything OK	Instruction was properly received and completely executed.
`01`	Other error	Unspecified device error.
`02`	Invalid instruction code	The received instruction code is unknown.
`03`	Invalid data	Data has an invalid length or contains an invalid value.
`04`	Write not allowed / access refused	The request was not performed because certain conditions had not been met. Attempt to write data to inaccessible memory. Attempt to change configuration without immediately preceding Allow configuration. Access to password-protected memory.
`05`	Device failure	Device failure requiring service intervention. Internal memory or settings memory error. Any other error affecting proper device function.
`06`	No data available	—
`0A-0F`	Automatically sent instruction	Periodic transmission of measured values, notifications, etc.

5.1.2.5 Check Sum (SUM)

```
SUM = 0xFF – ((PRE + FRM + NUM_HI + NUM_LO + ADR + SIG + INST/ACK + DATA) & 0xFF)
```

NUM enters the sum split into two bytes (upper and lower). The sum is masked to 8 bits (`& 0xFF`) before subtraction, so the result is always in range 0x00–0xFF.

> No response is made to messages with a faulty check sum. (The system waits for CR reception even if an incorrect check sum is received.)

5.2 Instruction Overview

Instruction	Code	Category	Page
Single measurement	51	Basic	13
Single measurement – strings	58	Basic	13
Setting communication parameters	E0	Configuration	14
Read communication parameters	F0	Configuration	15
Set indicator mode	E5	Configuration	16
Read indicator mode	F5	Configuration	16
Write altitude	37	Configuration	16
Read altitude	38	Configuration	17
Write CO2 calibration type	39	Configuration	17
Read CO2 calibration type	3A	Configuration	18
Set sound indication	66	Configuration	18
Read sound indication settings	67	Configuration	18
Allow configuration	E4	Miscellaneous	19
CO2 sensor calibration	1E	Miscellaneous	19
Set status	E1	Miscellaneous	19
Read status	F1	Miscellaneous	20
Read name and version	F3	Miscellaneous	20
Reset	E3	Miscellaneous	21
Allow checksum	EE	Miscellaneous	21
Read checksum settings	FE	Miscellaneous	21
Save user data	E2	Miscellaneous	22
Read saved user data	F2	Miscellaneous	22
Read communication errors	F4	Miscellaneous	22
Address setup using serial number	EB	Miscellaneous	23
Read factory data	FA	Miscellaneous	23
Switching the communication protocol	ED	Miscellaneous	24

For clarity, only the instruction (INST), acknowledgement (ACK) and data (DATA) are described in detail below. Address (ADR), signature (SIG) and check sum (SUM) are described above in the protocol description and in the detailed Spinel protocol documentation (available for download at papouch.com/spinel).

5.3 Basic Instructions

5.3.1 Single Measurement

Performs a single measurement of CO2, temperature and humidity.

Request: `0x51`
Response: `ACK 0x00` `status` `co2` `temperature` `humidity` `dew-point` `time`

Legend:

Field	Length [B]	Description
`status`	1	0 = values are valid 1 = waiting for first measurement 2 and 3 = a value is out of range 4 = sensor fault
`co2`	2	CO2 level as a positive integer directly in ppm[5]
`temperature`	2	Temperature multiplied by ten in signed int format
`humidity`	2	Humidity multiplied by ten as a positive integer
`dew-point`	2	Dew point temperature multiplied by ten in signed int format
`time`	2	Time since power-up in seconds [5]

Example:

```
→  2A 61 00 05 31 02 51 EB 0D
```

```
←  2A 61 00 10 31 02 00 00 01 6F 01 04 00 DD 00 1A 00 38 8D 0D
```

Field	Hex	Decimal	Value
`status`	`00`	—	Everything OK
`co2`	`01 6F`	367	367 ppm
`temperature`	`01 04`	260	26.0 °C
`humidity`	`00 DD`	221	22.1 %RH
`dew-point`	`00 1A`	26	2.6 °C
`time`	`00 38`	56	56 sec

5.3.2 Single Measurement – Strings

Performs a single measurement of CO2, temperature and humidity. Unlike the previous instruction, this one returns values as strings.

Request: `0x58` `quantity`
Response – all: `ACK 0x00` `status` `co2` `temperature` `humidity` `dew-point` `time`
Response – one value: `ACK 0x00` `status` `value` `time`

[5] The CO2 level reading becomes accurate after several tens of seconds from sensor power-up. The `time` field can be used to evaluate whether the CO2 measurement has settled. `time` is an integer from 0 to 3600 (the counter stops after reaching 3600).

Legend:

Field	Length [B]	Description
`quantity`	1	0 = all quantities 1–4 = only one selected quantity (1: CO2, 2: Temperature, 3: Humidity, 4: Dew point)
`status`	1	0 = values are valid
`co2`	10	CO2 level as a right-aligned string
`temperature`	10	Temperature with one decimal place as a right-aligned string
`humidity`	10	Humidity with one decimal place as a right-aligned string
`dew-point`	10	Dew point temperature with one decimal place as a right-aligned string
`time`	10	Time since power-up as a string [5]

Example: request for one quantity (CO2, quantity = 0x01)

```
→  2A 61 00 06 31 02 58 01 E2 0D
```

```
←  2A 61 00 1A 31 02 00 00 20 20 20 20 20 20 20 38 30 39 20 20 20 20 20 20 20 20 20 34 52 0D
```

Field	Value
`status`	0x00 = everything OK
`co2`	″ 809″ = 809 ppm
`time`	″ 4″ = 4 sec

5.4 Configuration Instructions

5.4.1 Setting Communication Parameters

Sets the address and communication speed. This instruction must be immediately preceded by Allow configuration. The instruction cannot be used with the universal address or broadcast address.

Request: `0xE0` `address` `speed`
Response: `ACK 0x00`

Legend:

Field	Length [B]	Description
`address`	1	Can range from 0x00 to 0xFD
`speed`	1	Communication speed — codes shown in the table below

Communication speed codes:

Speed	Code
1 200 Bd	`03`
2 400 Bd	`04`
4 800 Bd	`05`

Speed	Code
9 600 Bd	`06` (default)
19 200 Bd	`07`
38 400 Bd	`08`
57 600 Bd	`09`
115 200 Bd	`0A`

Example:

Setting address 0x04 and communication speed 19200 Bd; old address 0x01, signature 0x02

```
→  2A 61 00 07 01 02 E0 04 07 7F 0D
```

```
←  2A 61 00 05 01 02 00 6C 0D
```

Notes:

- The new address and communication speed are set after the response is sent.
- Allow configuration instruction must precede this instruction. After setting communication parameters, configuration is automatically disabled.
- Other communication parameters are: 8 bits, no parity, 1 stop bit. The factory default communication speed is 9600 Bd.
- If the address is unknown and no other device is connected to the line, the address can be found using Read communication parameters. Use the universal address 0xFE.

5.4.2 Read Communication Parameters

Returns the address and communication speed.

Request: `0xF0`
Response: `ACK 0x00` `address` `speed`

Legend:

Field	Length [B]	Description
`address`	1	Device address
`speed`	1	Communication speed — codes shown in the table above

Example:

Read communication parameters; universal address 0xFE, signature 0x02

```
→  2A 61 00 05 FE 02 F0 7F 0D
```

```
←  2A 61 00 07 04 02 00 04 06 5D 0D
```

- Address 0x04, communication speed 9600 Bd

Notes:

- This instruction is intended for finding the set address when it is unknown. The request is sent to the universal address 0xFE. No other device must be connected to the line.
- Other communication parameters are: 8 bits, no parity, 1 stop bit. The factory default communication speed is 9600 Bd.

5.4.3 Set Indicator Mode

Sets the LED indicator mode on the device to one of three modes. See LED Indication Modes.

Request: `0xE5` `mode`
Response: `ACK 0x00`

Legend:

Field	Length [B]	Description
`mode`	1	Number 0x01 to 0x03 — LED indicator mode. See LED Indication Modes.

Example:

```
→ 2A 61 00 06 31 02 E5 02 54 0D
```

```
← 2A 61 00 05 31 02 00 3C 0D
```

5.4.4 Read Indicator Mode

Reads the currently set indicator mode.

Request: `0xF5`
Response: `ACK 0x00` `mode`

Legend:

Field	Length [B]	Description
`mode`	1	Number 0x01 to 0x03 — LED indicator mode. See LED Indication Modes.

Example:

```
→ 2A 61 00 05 31 02 F5 47 0D
```

```
← 2A 61 00 06 31 02 00 02 39 0D
```

5.4.5 Write Altitude

Sets the altitude for CO2 measurement compensation. The value is saved to non-volatile memory.

Request: `0x37` `altitude`
Response: `ACK 0x00`

Legend:

Field	Length [B]	Description
`altitude`	2	Altitude in meters. Range: 0–3000.

Example: write altitude 1500 m; address 0x31, signature 0x0D

```
→ 2A 61 00 07 31 0D 37 05 DC 17 0D
```

```
← 2A 61 00 05 31 0D 00 31 0D
```

5.4.6 Read Altitude

Reads the currently set altitude.

Request: `0x38`
Response: `ACK 0x00` `altitude`

Legend:

Field	Length [B]	Description
`altitude`	2	Altitude in meters. Range: 0–3000.

Example: read altitude; address 0x31, signature 0x0C

```
→ 2A 61 00 05 31 0C 38 FA 0D
```

```
← 2A 61 00 07 31 0C 00 00 00 30 0D
```

- Altitude 0x0000 = 0 m

5.4.7 Write CO2 Calibration Type

Sets the CO2 sensor calibration type (automatic or manual).

Request: `0x39` `type`
Response: `ACK 0x00`

Legend:

Field	Length [B]	Description
`type`	1	`0x00` = automatic calibration *(default)* `0x01` = manual calibration

Example: write manual calibration; address 0x31, signature 0x0F

```
→ 2A 61 00 06 31 0F 39 01 F4 0D
```

```
← 2A 61 00 05 31 0F 00 2F 0D
```

5.4.8 Read CO2 Calibration Type

Reads the currently set CO2 sensor calibration type.

Request: `0x3A`
Response: `ACK 0x00` `type`

Legend:

Field	Length [B]	Description
`type`	1	`0x00` = automatic calibration `0x01` = manual calibration

Example: read calibration type; address 0x31, signature 0x0E

```
→ 2A 61 00 05 31 0E 3A F6 0D
```

```
← 2A 61 00 06 31 0E 00 00 2F 0D
```

- `0x00` = automatic calibration

5.4.9 Set Sound Indication

Enables or disables the sound indication (beeping) of the device.

Request: `0x66` `state`
Response: `ACK 0x00`

Legend:

Field	Length [B]	Description
`state`	1	`0x00` = sound indication disabled `0x01` = sound indication enabled *(default)*

Example: disable sound indication; address 0x31, signature 0x11

```
→ 2A 61 00 06 31 11 66 00 C6 0D
```

```
← 2A 61 00 05 31 11 00 2D 0D
```

5.4.10 Read Sound Indication Settings

Reads the current sound indication setting.

Request: `0x67`
Response: `ACK 0x00` `state`

Legend:

Field	Length [B]	Description
`state`	1	`0x00` = sound indication disabled `0x01` = sound indication enabled

Example: read sound indication state; address 0x31, signature 0x10

```
→ 2A 61 00 05 31 10 67 C7 0D
```

```
← 2A 61 00 06 31 10 00 01 2C 0D
```

- `0x01` = sound indication enabled

5.5 Miscellaneous Instructions

5.5.1 Allow Configuration

Allows configuration changes. Must immediately precede certain instructions (Setting communication parameters and Allow checksum). After the following instruction (even an invalid one), configuration is automatically disabled. The instruction cannot be used with the universal address or broadcast address.

Request: `0xE4`
Response: `ACK 0x00`

Example:

```
→ 2A 61 00 05 01 02 E4 88 0D
```

```
← 2A 61 00 05 01 02 00 6C 0D
```

5.5.2 CO2 Sensor Calibration

Writing the number 400 calibrates the CO2 sensor to 400 ppm and switches to manual calibration mode (see Read CO2 calibration type). Place THCO2 for at least 5 minutes in an environment with 400 ppm CO2 (or outside) before calibrating! THCO2 is set to automatic calibration from the factory.

This instruction must be immediately preceded by Allow configuration.

Request: `0x1E`
Response: `ACK 0x00`

Example:

```
→ 2A 61 00 05 31 02 1E 1E 0D
```

```
← 2A 61 00 05 31 02 00 3C 0D
```

5.5.3 Set Status

Sets the device status. A user-defined byte that can be used to identify the device state.

Request: `0xE1` `status`
Response: `ACK 0x00`

Legend:

Field	Length [B]	Description
`status`	1	Device status. After power-up or reset, the status is automatically set to 0x00. If changed to a different value using this instruction, the device state can be easily identified later.

Example: set status 0x12; address 0x01, signature 0x02

```
→  2A 61 00 06 01 02 E1 12 78 0D
```

```
←  2A 61 00 05 01 02 00 6C 0D
```

5.5.4 Read Status

Reads the device status. A user-defined byte — see Set status.

Request: `0xF1`
Response: `ACK 0x00` `status`

Legend:

Field	Length [B]	Description
`status`	1	Device status — see Set status.

Example: read status; address 0x01, signature 0x02

```
→  2A 61 00 05 01 02 F1 7B 0D
```

```
←  2A 61 00 06 01 02 00 12 59 0D
```

5.5.5 Read Name and Version

Reads the device name, internal software version and list of available communication formats (format 97 for THCO2). Factory set.

Request: `0xF3`
Response: `ACK 0x00` `string`

Legend:

Field	Description
`string`	Text in format: `THCO2; v1395.02.04; f97 fModbus`

Example:

```
→  2A 61 00 05 31 02 F3 49 0D
```

```
←  2A 61 00 24 31 04 00 54 48 43 4F 32 3B 20 76 31 33 39 35 2E 30 32 2E 30 34 3B 20 66 39 37
20 66 4D 6F 64 62 75 73 D5 0D
```

5.5.6 Reset

Resets the device. The module returns to the same state as after a power-up. The reset is performed after the response is sent!

Request: `0xE3`
Response: `ACK 0x00`

Example: reset; address 0x01, signature 0x02

```
→ 2A 61 00 05 01 02 E3 89 0D
```

```
← 2A 61 00 05 01 02 00 6C 0D
```

5.5.7 Allow Checksum

Enables checksum verification for incoming messages. This instruction must be immediately preceded by Allow configuration.

Request: `0xEE` `state`
Response: `ACK 0x00`

Legend:

Field	Length [B]	Description
`state`	1	0x01 = enable checksum verification 0x00 = disable

Example:

```
→ 2A 61 00 06 01 02 EE 01 7C 0D
```

```
← 2A 61 00 05 01 02 00 6C 0D
```

5.5.8 Read Checksum Settings

Reads the current checksum verification setting.

Request: `0xFE`
Response: `ACK 0x00` `state`

Legend:

Field	Length [B]	Description
`state`	1	0x01 = checksum verification enabled 0x00 = checksum verification disabled

Example:

```
→ 2A 61 00 05 01 02 FE 6E 0D
```

```
← 2A 61 00 06 01 02 00 01 6A 0D
```

5.5.9 Save User Data

Saves user data. The device retains the data when powered off.

Request: `0xE2` `position` `data`
Response: `ACK 0x00`

Legend:

Field	Length [B]	Description
`position`	1	Memory address where the data will be saved. 0x00 to 0x0F
`data`	1–16	Any user data.

Example: save the word "BASEMENT 1" to memory address 0x00; address 0x01, signature 0x02

```
→ 2A 61 00 0F 01 02 E2 00 4B 4F 54 45 4C 4E 41 20 31 61 0D
```

```
← 2A 61 00 05 01 02 00 6C 0D
```

> **Note:** User data memory is 16 bytes. When writing to memory address e.g. 0x0C, a maximum of 4 bytes can be saved.

5.5.10 Read Saved User Data

Reads saved user data. The device retains the data when powered off.

Request: `0xF2`
Response: `ACK 0x00` `data`

Legend:

Field	Length [B]	Description
`data`	16	Saved user data.

Example: read user data; address 0x01, signature 0x02

```
→ 2A 61 00 05 01 02 F2 7A 0D
```

```
← 2A 61 00 15 01 02 00 4B 4F 54 45 4C 4E 41 20 31 20 20 20 20 20 20 20 5D 0D
```

- Response with string "KOTELNA 1 "

5.5.11 Read Communication Errors

Returns the number of communication errors that occurred since the device was powered up or since the last read of communication errors.

Request: `0xF4`
Response: `ACK 0x00` `errors`

Legend:

Field	Length [B]	Description
`errors`	1	Number of communication errors. Errors include: prefix expected but a different byte received; check sum mismatch; incomplete message.

Example: read communication errors; address 0x01, signature 0x02

```
→  2A 61 00 05 01 02 F4 78 0D
```

```
←  2A 61 00 06 01 02 00 05 66 0D
```

- 5 errors

5.5.12 Address Setup Using Serial Number

Allows setting the module address using the serial number only.

Request: `0xEB` `new-address` `product-number` `serial-number`
Response: `ACK 0x00`

Legend:

Field	Length [B]	Description
`new-address`	1	New module address.
`product-number`	2	Product number; for THCO2, this is always 1395 decimal (0x0573 hex).
`serial-number`	2	THCO2 serial number is on the label after `1395/`. This number can also be found using Read factory data.

Example: new address 0x32, product number 199 (0x00C7), serial number 101 (0x0065)

```
→  2A 61 00 0A FE 02 EB 32 00 C7 00 65 21 0D
```

```
←  2A 61 00 05 32 02 00 3B 0D
```

- In the response, the device already uses the newly assigned address.

5.5.13 Read Factory Data

Reads factory data from the device.

Request: `0xFA`
Response: `ACK 0x00` `product-number` `serial-number` `factory-data`

Legend:

Field	Length [B]	Description
`product-number`	2	Product number; for THCO2, this is always 1395 decimal (0x0573 hex).
`serial-number`	2	THCO2 serial number is on the label after `1395/`.
`factory-data`	4	Factory data.

Example:

```
→ 2A 61 00 05 FE 02 FA 75 0D
```

```
← 2A 61 00 0D 35 02 00 00 C7 00 65 20 05 09 23 B3 0D
```

- Product number 199 (0x00C7), serial number 101 (0x0065), factory data 0x20050923.

5.5.14 Switching the Communication Protocol

Switches the communication protocol type. This instruction must be preceded by Allow configuration.

Request: `0xED` `id`
Response: `ACK 0x00`

Legend:

Field	Length [B]	Description
`id`	1	Protocol identification number: 0x01 = Spinel format 97 0x02 = Modbus RTU

> Modbus Configurator, available for download at papouch.com, can also be used to switch protocols.

> **Warning:** If the CFG jumper on the PCB is shorted, the device always communicates using Modbus, regardless of this instruction's setting!

Example:

```
→ 2A 61 00 06 31 02 ED FF 4F 0D
```

```
← 2A 61 00 05 31 02 00 3C 0D
```

Notes:

Notes:

Notes:

Papouch s.r.o.

Industrial data transfer, line and protocol converters, RS232, RS485, RS422, USB, Ethernet, LTE, WiFi, measurement modules, intelligent temperature sensors, I/O modules, custom development and manufacturing.

Address: Strašnická 3164/1a, 102 00 Prague 10, Czech Republic
Phone: +420 267 314 268
Internet: www.papouch.com
E-mail: papouch@papouch.com

© 2026 Papouch s.r.o.