M212572EN-D

Configuration Guide

Vaisala Air Quality Transmitter

AQT530

VAISALA

PUBLISHED BY

Vaisala Oyj
Vanha Nurmijärventie 21, FI-01670 Vantaa, Finland
P.O. Box 26, FI-00421 Helsinki, Finland
+358 9 8949 1

Visit our Internet pages at www.vaisala.com.

Table of contents

List of figures

List of tables

1. About this document

1.1 Version information

This document provides instructions for configuring Vaisala Air Quality Transmitter AQT530.

Table 1 Document versions (English)

Document code	Date	Description
M212572EN-D	December 2021	For device SW version 3.3: • Modbus version 1.3 • CLI version 1.3 Changes to Modbus: • Changed the representation of the hexadecimal numbers from specific format 0xFFFF to generic format FFFFh • Added time units to calibration registers 0013h - 0015h • Added time units to system time registers 0057h - 005Ch • Added new LPC update interval register 007Fh • Added unit resolution (%, per cent) for register values 0086h - 008Bh • Added unit resolution (‰, per mille) for register values 0092h, 0093h and 0096h • Added new 32-bit device uptime register 0098h + 0099h
M212572EN-C	September 2021	• Modbus register address type updated for 0x0066 - 0x006B • Default parity for `rs485_parity` updated
M212572EN-B	April 2021	For device SW version 3.1: • Modbus version 1.2 • CLI version 1.2 Main changes: • Added support for PM_1 observations (CLI, Modbus, ASCII CSV) • Added support for measurement values without linear correction (CLI and Modbus) • Added humidity invalidation flags for each PM value (Modbus) • Added device status monitoring (CLI and Modbus) • Updated default Modbus serial port settings • Updated and added some examples Changes to CLI: • Added commands **`status`** and **`meas --nolc`** • Added PM_1 to **`lpc --meas`** output • Added parameters `pm1_zero` and `pm1_span` Changes to Modbus: • Added device status registers 0x004B - 0x004C • Added measurement data registers 0x006C - 0x0075 • Added humidity invalidation flag registers 0x007C - 0x007E for PM_1, $PM_{2.5}$, and PM_{10} • Updated LPC humidity invalidation flag register 0x007B • Added span and zero registers 0x0096 - 0x0097 for PM_1

1.2 Related manuals

Table 2 AQT530 manuals

Document code	Name
M212573EN	*Vaisala Air Quality Transmitter AQT530 Setup Guide*
M212572EN	*Vaisala Air Quality Transmitter AQT530 Configuration Guide*
M212580EN	*Vaisala Air Quality Transmitter AQT530 Maintenance Guide*

The documentation is available online at http://www.vaisala.com/aqt530.

1.3 Documentation conventions

WARNING! Warning alerts you to a serious hazard. If you do not read and follow instructions carefully at this point, there is a risk of injury or even death.

CAUTION! Caution warns you of a potential hazard. If you do not read and follow instructions carefully at this point, the product could be damaged or important data could be lost.

Note highlights important information on using the product.

Tip gives information for using the product more efficiently.

Lists tools needed to perform the task.

Indicates that you need to take some notes during the task.

1.4 Trademarks

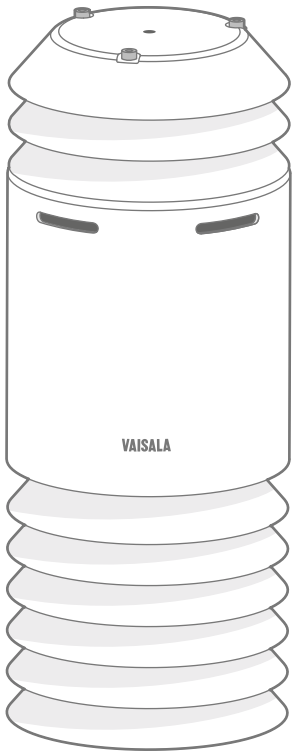
Vaisala® is a registered trademark of Vaisala Oyj.

Modbus® is a registered trademark of Schneider Automation Inc.

Microsoft® and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

2. Product overview

2.1 Vaisala Air Quality Transmitter AQT530



Vaisala Air Quality Transmitter AQT530 measures the pollution content of ambient air. AQT530 is available in different models for measuring gases, particles, or both.

AQT530 is specifically designed for air quality monitoring networks in areas with traffic, road networks, or around transportation hubs.

Thanks to its small weight, compact size, and good precision it is ideally suited for deployment especially in large air quality networks. The measurement data can be sent wirelessly to a web-based database with a gateway solution and it is also available locally through a serial interface.

AQT530 is available as a standalone instrument for integration to customer system. AQT530 can also be paired with Vaisala Beacon Station BWS500 or Vaisala Road Weather Station RWS200 to provide a professional-grade complete network solution with best-in-class accuracy and reliability.

2.2 Safety

This product has been tested for safety. Note the following precautions:

WARNING! Do not replace components when the system is powered up. Disconnect all power sources before performing maintenance procedures.

WARNING! Do not substitute parts or modify the system, or install unsuitable parts in the system. Improper modification can damage the product or lead to malfunction.

WARNING! Assess the risks from the installation work. Consider also the effects of local weather conditions.

WARNING! Failure to comply with these precautions or with specific warnings elsewhere in these instructions violates safety standards of design, manufacture, and intended use of the product. Vaisala assumes no liability for the customer's failure to comply with these requirements.

WARNING! If the equipment is used in a manner not specified by Vaisala, the protection provided by the equipment may be impaired.

WARNING! Follow local and state legislation and regulations on occupational safety.

WARNING! Do not attempt to open the laser particle counter (LPC) for service. Removing the cover may cause exposure to harmful class 3B laser radiation and electrical shock.

CAUTION! Improper modification can damage the product or lead to malfunction. Any modification voids your warranty.

2.2.1 Eye safety

Vaisala Air Quality Transmitter AQT530 is classified as a Class 1 laser device in accordance with International Standard IEC 60825-1. It complies with 21 CFR 1040.10 and 1040.11 except for conformance with IEC 60825-1 ED. 3., as described in the Laser Notice No. 56, dated May 8,2019. A Class 1 laser device is safe under all conditions of normal use.

AQT530 incorporates a Class 3B laser. The laser is contained in an enclosure, preventing direct physical access to laser radiation.

Table 3 Incorporated laser module

Property	Description/Value
Laser module	Class 3B
Laser wavelength	658 nm
Maximum power	22 mW

The device is equipped with the following laser product labels.

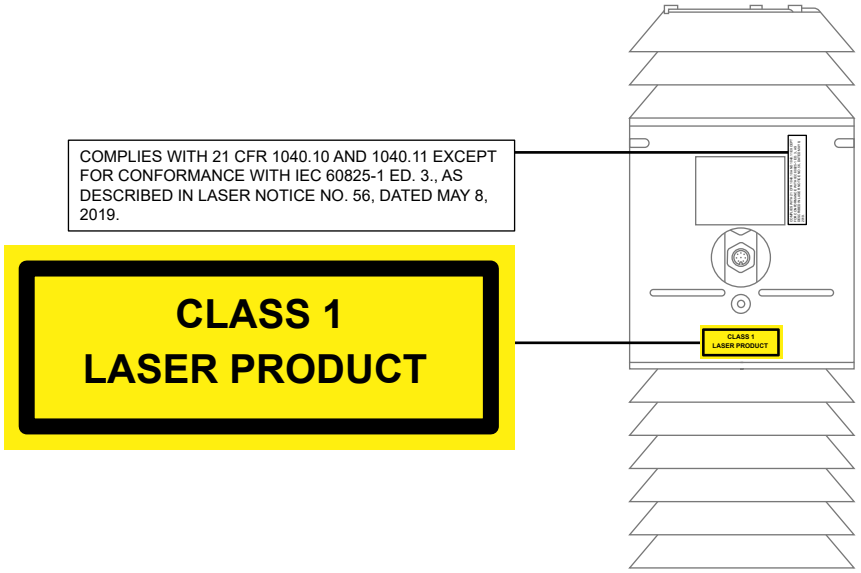


Figure 1 Location of Class 1 laser product labels on AQT530

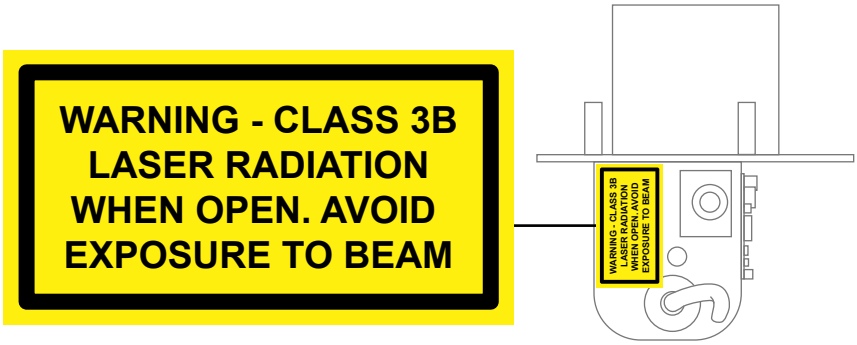


Figure 2 Location of Class 3B laser radiation label on AQT530 inside laser particle counter (LPC)

Follow the safety information to avoid exposure to laser radiation.

2.2.2 ESD protection

Electrostatic Discharge (ESD) can damage electronic circuits. Vaisala products are adequately protected against ESD for their intended use. However, it is possible to damage the product by delivering electrostatic discharges when touching, removing, or inserting any objects in the equipment housing.

To avoid delivering high static voltages to the product:

- Handle ESD-sensitive components on a properly grounded and protected ESD workbench or by grounding yourself to the equipment chassis with a wrist strap and a resistive connection cord.
- If you are unable to take either precaution, touch a conductive part of the equipment chassis with your other hand before touching ESD-sensitive components.
- Hold component boards by the edges and avoid touching component contacts.

3. Configuration

3.1 Interface overview

The Modbus interface is an application programming interface (API) for remote connection to AQT530 using the Modbus protocol.

By default, AQT530 outputs data in the Modbus RTU format.

AQT530 has a command line interface (CLI) and a Modbus interface:

- Modbus interface properties are configured over the CLI interface.
- All measurement values are available from both CLI and Modbus.
- All configuration parameters are available through CLI, while Modbus supports the most important configuration parameters.

Alternatively, AQT530 outputs data in ASCII CSV format over the RS-485 interface. ASCII CSV format uses the same physical channel as Modbus format, so only one format can be used at a time.

The CLI interface is for a local connection over RS-232.

Table 4 AQT530 data connection specifications

Property	Description/Value
Data output	Modbus® ASCII, Modbus® RTU, ASCII CSV
Serial data interface	RS-485
Maintenance interface [1)]	RS-232

1) Recommended Vaisala USB maintenance cable kit (253163SET).

More information

- Overview of CLI commands and parameters (page 29)
- Modbus register addresses for AQT530 (page 46)
- ASCII CSV data message (page 23)

3.2 Configuration overview

You can do the operations listed in the following table using a local connection with CLI commands and/or remotely over the Modbus interface.

Table 5 Overview of configuration operations

Action	CLI command and parameter	CLI reference	Modbus register addresses
View measurement data	**meas**	Printing measurements and changing measurement output unit (meas command) (page 33)	• 0000h ... 000Ch (gas values in ppb with linear correction) • 0066h ... 006Bh (gas values in ug/m^3) • 006Ch ... 0075h (gas values in ppb without linear correction)
Configure Modbus interface	**set** • rs485_mode • rs485_addr • rs485_baud • rs485_databits • rs485_parity • rs485_stopbits	Configuring Modbus interface (page 21)	–
Change temperature unit (°C/°F)	**set tempunit**	Changing temperature unit (page 25)	001Ch
Adjust linear correction	**set** • co_zero • co_span • no_zero • no_span • no2_zero • no2_span • so2_zero • so2_span • h2s_zero • h2s_span • o3_zero • o3_span • pm1_zero • pm1_span • pm25_zero • pm25_span • pm10_zero • pm10_span	Adjusting linear correction (page 26)	0086h ... 0097h
Set system time	**date**	Setting date (date command) (page 31)	0057h ... 005Ch See Setting up system time over Modbus interface (page 53)

Action	CLI command and parameter	CLI reference	Modbus register addresses
AQT530, base module, LPC, and HMP identification (read-only)	**show** • serial • sw_ver • hw_ver • model • unit • cal_date • base_serial • lpc_serial • hmp_serial • hmp_version **lpc** --info	Printing LPC information (lpc command) (page 32)	Modbus device identification: • Model • SW version • Sensor model • Serial number • Calibration date • Hardware version Modbus registers: • Calibration date: 0013h - 0015h • Base firmware version: 004Fh -0053h • HMP firmware version: 0054h - 0056h • AQT serial number: 00B4h - 00B7h • HMP serial number: 00B8h - 00BBh • LPC serial number: 00BC - 00BFh • LPC firmware version: 00F4 - 00F8h
Get ug/m^3 converted values	**meas** --metric	Printing measurements and changing measurement output unit (meas command) (page 33)	Modbus register addresses are available for both metric (ug/m^3) and parts per billion (ppb) values
Reset device	**reboot** --really	Restarting device (reboot command) (page 36)	00FAh

To use the CLI commands, establish a local maintenance connection to AQT530 over RS-232.

More information

- Overview of CLI commands and parameters (page 29)
- Modbus register addresses for AQT530 (page 46)

3.3 Measurement output

The AQT530 measurement data is available over RS-485 by means of Modbus (ASCII and RTU modes), or alternatively in ASCII CSV format.

The measurement data can be monitored also through CLI.

The following tables present the measurements, their units, and resolution.

Set of available gases depends on gas cell setup.
Particle measurement results are included when an LPC device is installed and enabled.

Table 6 Temperature

Value	Modbus	CLI	ASCII CSV
Air temperature	**000Ah**: 0.1 °C / 0.1 °F	• **meas**: °C / °F • **meas --metric**: °C / °F • **meas --ppb**: °C / °F	°C / °F

Table 7 Humidity

Value	Modbus	CLI	ASCII CSV
Air humidity	**000Bh**: 0.1 %RH	• **meas**: %RH • **meas --metric**: %RH • **meas --ppb**: %RH	%RH

Table 8 Pressure

Value	Modbus	CLI	ASCII CSV
Air pressure	**000Ch**: 0.1 hPa	• **meas**: hPa • **meas --metric**: hPa • **meas --ppb**: hPa	hPa

Table 9 Gas concentration

Value	Modbus	CLI	ASCII CSV
NO_2	• **0000h**: ppb • **0066h**: 0.1 µg / m^3	• **meas**: ppm • **meas --metric**: µg/m^3 • **meas --ppb**: ppb	ppm
SO_2	• **0001h**: ppb • **0067h**: 0.1 µg/m^3	• **meas**: ppm • **meas --metric**: µg/m^3 • **meas --ppb**: ppb	ppm
CO	• **0002h**: ppb • **0068h**: 1 µg/m^3	• **meas**: ppm • **meas --metric**: µg/m^3 • **meas --ppb**: ppb	ppm
H_2S	• **0004h**: ppb • **0069h**: 0.1 µg/m^3	• **meas**: ppm • **meas --metric**: µg/m^3 • **meas --ppb**: ppb	ppm
O_3	• **0005h**: ppb • **006Ah**: 0.1 µg/m^3	• **meas**: ppm • **meas --metric**: µg/m^3 • **meas --ppb**: ppb	ppm

Value	Modbus	CLI	ASCII CSV
NO	• `0006h`: ppb • `006Bh`: 0.1 µg/m^3	• **meas**: ppm • **meas** `--metric`: µg/m^3 • **meas** `--ppb`: ppb	ppm

Table 10 Mass concentration without linear correction

Value	Modbus	CLI	ASCII CSV
PM_1	0075h: 0.1 µg/m^3	**meas** `--nolc`: µg/m^3	–
$PM_{2.5}$	0073h: 0.1 µg/m^3	**meas** `--nolc`: µg/m^3	–
PM_{10}	0074h: 0.1 µg/m^3	**meas** `--nolc`: µg/m^3	–

Table 11 Mass concentration with linear correction

Value	Modbus	CLI	ASCII CSV
PM_1	`0037h`: 0.1 µg/m^3	• **meas**: µg/m^3 • **meas** `--metric`: µg/m^3 • **meas** `--ppb`: µg/m^3	µg/m^3
$PM_{2.5}$	`0008h`: 0.1 µg/m^3	• **meas**: µg/m^3 • **meas** `--metric`: µg/m^3 • **meas** `--ppb`: µg/m^3	µg/m^3
PM_{10}	`0009h`: 0.1 µg/m^3	• **meas**: µg/m^3 • **meas** `--metric`: µg/m^3 • **meas** `--ppb`: µg/m^3	µg/m^3

More information

- Modbus register addresses for AQT530 (page 46)
- Printing measurements and changing measurement output unit (meas command) (page 33)
- ASCII CSV data message (page 23)

3.3.1 Conversion factors for gases

Conversion from ppb to µg/m^3 is calculated with the following formula:

µg/m^3 = ppb * conversion_factor

Table 12 Conversion factors for conversion from ppb to µg/m^3

Gas	Conversion factor
NO_2	1.912
SO_2	2.66

Gas	Conversion factor
CO	1.16
H_2S	1.417
O_3	2.00
NO	1.247

At +20 °C (+68 °F) temperature (European standard)

4. CLI operation

4.1 Connecting to AQT530 over RS-232

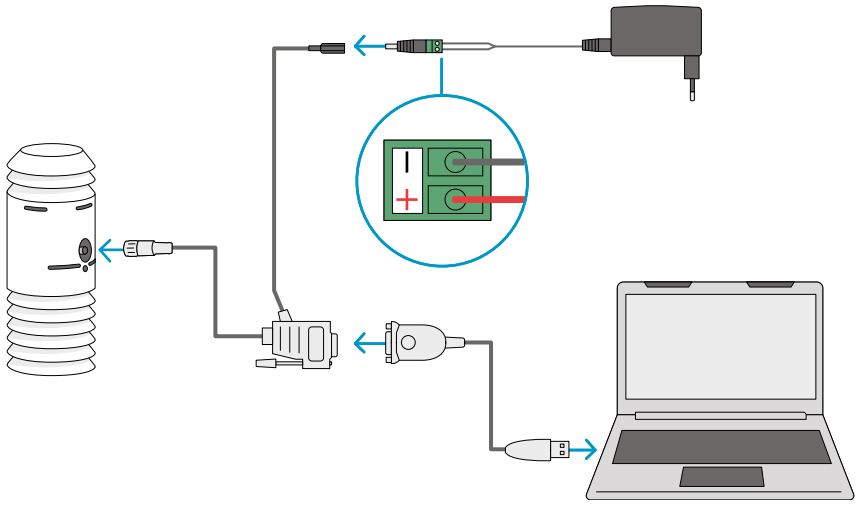
- Computer
- Service cable kit
- Slothead screwdriver
- Terminal block
- Terminal program with a command line interface (CLI), such as PuTTY or TeraTerm.

This instruction and examples use and refer to PuTTY. Another program with a command line interface can also be used, but the user interface looks different.

Once you have installed and set up AQT530 as instructed in *Vaisala Air Quality Transmitter AQT530 Setup Guide*, you may sometimes need to connect to AQT530 locally, for example, to change the settings.

To connect to AQT530, create a local maintenance connection over the serial RS-232 interface.

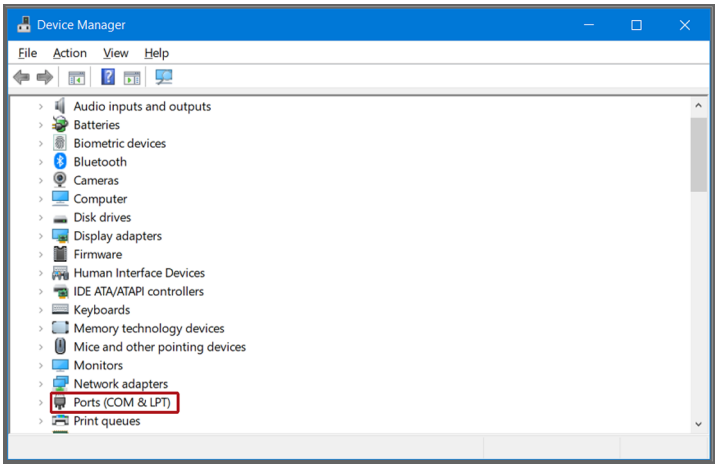
1. Connect your computer to AQT530 with the service cable kit.



Connect the open-ended cables of the power source to the adapter and attach the adapter to the DC power connector (10 ... 25 V DC).
Alternatively, connect your own power source directly to the DC power connector.

Wire color	Power
Red	+
Black	–

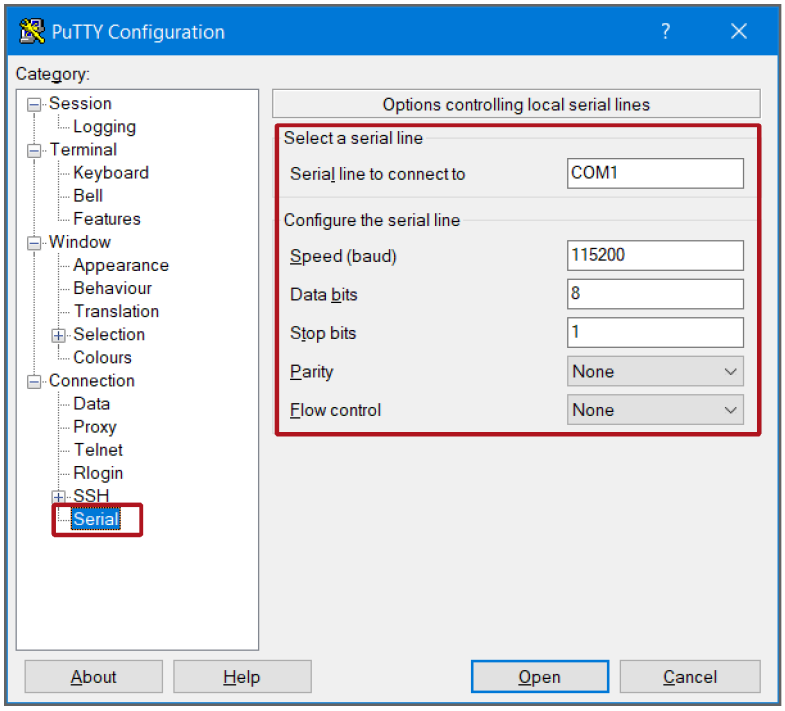
2. On your computer, select **Start > PuTTY**.
3. On your computer, select **Start > Control Panel > Device Manager > Ports (COM & LPT)**.



In **Device Manager**, check to which port AQT530 is connected with name **ATEN USB to Serial Bridge**. For example: ATEN USB to Serial Bridge (**COM1**).

4. In PuTTY, select **Serial** or **Serial & USB**.

> The screen may look different, depending on the computer and Windows operating system version.



5. In **PuTTY Configuration**, select the COM port where you connected ATEN USB to Serial Bridge.
   Use the following COM port settings.

Parameter	Value
Speed (baud)	**115200**
Data bits	**8**
Stop bits	**1**
Parity	**None**
Flow control	**None**

   These settings are applied on your computer so that it can connect to AQT530.

6. Select **Open**.
   Connection to AQT530 is now open and you can proceed to read and set parameters.
   To start, press **ENTER** or type command **`motd`**.

More information

- CLI command reference (page 29)

4.2 Configuring Modbus interface

The default communication settings are listed in the following table. Use the settings that are suitable for you.

Table 13 Modbus interface configuration parameters

Parameter	Type	Range	Default	Unit	Description
`rs485_mode`	uint8	• 0 = ASCII CSV • 1 = Modbus ASCII • 4 = Modbus RTU	4	–	RS-485 mode
`rs485_addr`	uint8	1 ... 253	1	–	Modbus address (not used in ASCII CSV mode)
`rs485_baud`	uint32	4800 ... 115200	19200	bps	RS-485 baud rate
`rs485_databits`	uint8	7, 8	8	bits	RS-485 data bits
`rs485_parity`	Text	• **`N`** = None • **`E`** = Even • **`O`** = Odd	E	–	RS-485 parity Case insensitive
`rs485_stopbits`	uint8	1, 2	1	bits	RS-485 stop bits

1. Establish a connection to AQT530.

2. Using a terminal program, define the settings for the RS-485 interface.
   - To use the RS-485 port in Modbus ASCII mode, type the following:

```
set rs485_baud=9600
set rs485_mode=1
set rs485_databits=7
set rs485_parity=e
set rs485_stopbits=1
write --really
```

     - Bit rate: 9600 bps
     - Mode: ASCII
     - Port settings: 7E1 (7 data bits, even parity, 1 stop bit)
   - To use the RS-485 port in Modbus RTU mode, type the following:

```
set rs485_baud=19200
set rs485_mode=4
set rs485_databits=8
set rs485_parity=e
set rs485_stopbits=1
write --really
```

     - Bit rate: 19200
     - Mode: RTU
     - Port settings: 8E1 (8 data bits, even parity, 1 stop bit)
3. The device is ready.

More information

- Changing configuration parameter (set command) (page 36)
- Saving to non-volatile memory (write command) (page 43)
- Modbus register addresses for AQT530 (page 46)

4.3 Configuring ASCII CSV data sending

The ASCII CSV output mode is an alternative to AQT530 Modbus API. It is a simplified measurement monitoring output for users that are only collecting data, without the ability to control and monitor the AQT530 device otherwise.

The comma-separated values (CSV) output is sent automatically every 60 seconds. The output contains 1 measurement from all measurement outputs on one line and with a timestamp. The output data does not contain validity or device health information.

1. Establish a connection to AQT530.

2. Set the RS-485 interface to ASCII CSV mode.

```
set rs485_mode=0
set rs485_baud=115200
set rs485_databits=8
set rs485_parity=n
set rs485_stopbits=1
write --really
```

   - Port settings: 8N1 (8 data bits, no parity, 1 stop bit)

   Data sending starts automatically after selecting the mode.
3. Close the local RS-232 maintenance connection.
4. You can view the data through the RS-485 connection. For example, if the RS-485 port is connected to the serial port of your computer, the ASCII CSV data is available there.

More information

- Changing configuration parameter (set command) (page 36)
- Saving to non-volatile memory (write command) (page 43)

4.3.1 ASCII CSV data message

The ASCII CSV data message is used for outputting AQT530 measurement data.

```
<Timestamp>,<Conditions>,<Gases>,<Particles>,<Config>,<Uptime>
```

Table 14 ASCII CSV message data fields

Field	Description	Example
Timestamp	UTC date and time in ISO 8601 format	2020-12-11T03:00:41
Conditions	Environment conditions separated by commas (included always): • Air temperature (°C or °F, depending on the configuration [1]) • Air humidity (%RH) • Air pressure (hPa)	7.3,92.3,990.1
Gases	Gas measurement results in parts per million (ppm) separated by commas Possible gases, depending on gas cell setup: • NO_2, SO_2, CO, H_2S, O_3, NO Max. number of gas cells at a time: 4	0.009,0.101,0.013,0.03

Field	Description	Example
Particles	Particle measurement results in µg/m^3, separated by a comma: • PM_1, $PM_{2.5}$, PM_{10} The particle measurement results are included when an LPC device is installed and enabled	0.1,1.1,1.9
Config	Device setup represented with symbols separated by colon (:): • T = Air temperature (included always) • H = Air humidity (included always) • P = Air pressure (included always) • NO_2, SO_2, CO, H_2S, O_3, NO (included according to gas cell setup) • PM_1, $PM_{2.5}$, PM_{10} (included when LPC is installed and enabled) Order matches order of results in fields Conditions, Gases, and Particles	T:H:P:NO2:CO:O3:NO:PM1:PM2.5:PM10
Uptime	Time from last reboot in seconds	2735641

1) *The temperature unit can be configured with CLI using* **`tempunit`** *parameter. For example, setting Celsius units with* **`set tempunit=0`**.

Example of CSV ASCII data message with 4 gas sensors and LPC

```
2022-01-22T07:37:38,22.3,24.1,999.3,0.182,2.920,0.575,0.140,0.1,1.1,1.9,T:H:P:N
O2:CO:O3:NO:PM1:PM2.5:PM10,3185
2022-01-22T07:38:38,22.3,24.1,999.3,0.170,2.921,0.551,0.131,0.1,1.1,1.9,T:H:P:N
O2:CO:O3:NO:PM1:PM2.5:PM10,3245
2022-01-22T07:39:38,22.3,24.1,999.3,0.159,2.919,0.527,0.123,0.1,1.1,1.9,T:H:P:N
O2:CO:O3:NO:PM1:PM2.5:PM10,3305
```

Example of CSV ASCII data message with 4 gas sensors, without LPC

```
2022-01-22T08:07:38,22.3,24.1,999.4,0.108,2.926,0.416,0.084,T:H:P:NO2:CO:O3:NO,
4983
2022-01-22T08:08:38,22.3,24.0,999.4,0.101,2.927,0.402,0.079,T:H:P:NO2:CO:O3:NO,
5043
2022-01-22T08:09:38,22.3,24.0,999.4,0.095,2.927,0.389,0.074,T:H:P:NO2:CO:O3:NO,
5103
```

Example of CSV ASCII data message with LPC only

```
2022-01-22T07:40:38,22.4,24.1,999.3,0.1,1.1,1.9,T:H:P:PM1:PM2.5:PM10,3364
2022-01-22T07:41:38,22.4,24.1,999.3,0.1,1.1,1.9,T:H:P:PM1:PM2.5:PM10,3424
2022-01-22T07:42:38,22.4,24.1,999.3,0.1,1.1,1.9,T:H:P:PM1:PM2.5:PM10,3484
```

The actual message does not contain line feeds.

4.4 Changing temperature unit

1. Establish a connection to AQT530.
2. To change the temperature unit to Celsius, type:

```
set tempunit=0
write --really
```

   To change to Fahrenheit, type:

```
set tempunit=1
write --really
```

3. To check that the settings are correct, type:

```
show tempunit
```

More information

- Changing configuration parameter (set command) (page 36)
- Saving to non-volatile memory (write command) (page 43)

4.5 Setting system time

1. Establish a connection to AQT530.
2. To set the system date and time, type:

```
date [YYYY-MM-DDTHH:II:SS]
```

   For example:

```
date 2020-05-22T12:34:11
```

3. To check the setting, type:

```
show time
```

More information

- Setting date (date command) (page 31)

4.6 Adjusting linear correction

Span and zero values are used for the linear correction of the measured values. Zero parameter corrects the offset and span the sensitivity (gain).

You can set the zero and span correction for gas and particle measurements.

The adjusted values affect the measured values of the following Modbus register addresses.

To adjust the span and zero through the Modbus interface, use registers 0086h - 0097h.

Table 15 Modbus registers for measurement data

Address	Description
0000h - 0006h	Gas concentrations in parts per billion (ppb) with linear correction
0066h - 006Bh	Gas concentrations in µg/m^3 with linear correction
0008h - 0009h 0037h	Particle matter PM_1, $PM_{2.5}$, and PM_{10} in µg/m^3 with linear correction

After making these changes, AQT530 outputs data corrected by these zero and span correction factors.

Table 16 Linear correction parameters - gases

Parameter	Type	Range	Default	Unit	Description
`co_zero`	int16	−10000 ... 10000	0	ppb	CO sensor zero correction
`co_span`	uint8	1 ... 255	100	%	CO sensor span correction (1 ... 255 = 0.01 ... 2.55)
`no_zero`	int16	−10000 ... 10000	0	ppb	NO sensor zero correction
`no_span`	uint8	1 ... 255	100	%	NO sensor span correction (1 ... 255 = 0.01 ... 2.55)
`no2_zero`	int16	−10000 ... 10000	0	ppb	NO_2 sensor zero correction

Parameter	Type	Range	Default	Unit	Description
no2_span	uint8	1 ... 255	100	%	NO_2 sensor span correction (1 ... 255 = 0.01 ... 2.55)
so2_zero	int16	–10000 ... 10000	0	ppb	SO_2 sensor zero correction
so2_span	uint8	1 ... 255	100	%	SO_2 sensor span correction (1 ... 255 = 0.01 ... 2.55)
h2s_zero	int16	–10000 ... 10000	0	ppb	H_2S sensor zero correction
h2s_span	uint8	1 ... 255	100	%	H_2S sensor span correction (1 ... 255 = 0.01 ... 2.55)
o3_zero	int16	–10000 ... 10000	0	ppb	O_3 sensor zero correction
o3_span	uint8	1 ... 255	100	%	O_3 sensor span correction (1 ... 255 = 0.01 ... 2.55)

Table 17 Linear correction parameters - particles

Parameter	Type	Range	Default	Unit	Description
pm1_zero	int16	–10000 ... 10000	0	0.1 µg/m³	PM_1 offset (zero) correction (–10000 ... 10000 = –1000.0 ... 1000.0)
pm1_span	int16	1 ... 10000	1000	‰	PM_1 span (gain) correction (1 ... 10000 = 0.001 ... 10.000)
pm25_zero	int16	–10000 ... 10000	0	0.1 µg/m³	$PM_{2.5}$ offset (zero) correction (–10000 ... 10000 = –1000.0 ... 1000.0)
pm25_span	int16	1 ... 10000	1000	‰	$PM_{2.5}$ span (gain) correction (1 ... 10000 = 0.001 ... 10.000)
pm10_zero	int16	–10000 ... 10000	0	0.1 µg/m³	PM_{10} offset (zero) correction (–10000 ... 10000 = –1000.0 ... 1000.0)
pm10_span	int16	1 ... 10000	1000	‰	PM_{10} span (gain) correction (1 ... 10000 = 0.001 ... 10.000)

1. Establish a connection to AQT530.
2. Adjust the gain and offset of the parameters.
   For example, to adjust offset up by 100 ppb and sensitivity 1.2 times (120 %) higher, type:

   ```
   set o3_zero=100
   set o3_span=120
   ```

3. To save the changes:

   ```
   write --really
   ```

   The new values overwrite the previous zero and span values.

Values without linear correction are available with Modbus registers 006Ch - 0075h.

More information

- Changing configuration parameter (set command) (page 36)
- Saving to non-volatile memory (write command) (page 43)

5. CLI commands and parameters

5.1 Overview of CLI commands and parameters

AQT530 CLI commands include the following:

- AQT530 identification commands: **motd**, **uname**
- System control command: **reboot**
- Configuration commands: **date**, **initconf**, **set**, **show**, **write**
- Measurement output command: **meas**
- LPC identification command: **lpc**
- Device status command: **status**

The AQT530 parameters include a set of read-only and read/write parameters.

Read/write parameters:

- Modbus interface configuration parameters Table 25 (page 37)
- Temperature unit parameter Table 26 (page 38)
- Linear correction parameters for gases Table 27 (page 38)
- Linear correction parameters for particles Table 28 (page 39)

Read-only parameters:

- Device identification parameters Table 29 (page 39)
- Base module identification parameters Table 30 (page 39)
- Base module monitoring parameters Table 31 (page 40)
- Gas measurement configuration and diagnostics parameters Table 32 (page 40)
- LPC parameters Table 33 (page 40)
- Temperature and humidity probe (TH) probe parameters Table 34 (page 40)

More information

- CLI command reference (page 29)
- CLI parameters (page 37)

5.2 CLI command reference

The commands are for use in a standard terminal program (command-line interface, CLI).

Table 18 List of commands

Command	Description	Reference
date [<YYYY-MM-DDThh:mm:ss>]	Prints or changes current date and time.	Setting date (date command) (page 31)

Command	Description	Reference
initconf --really	Loads default configuration, including: • Communication settings • Configuration and linear correction	Restoring default configuration (initconf command) (page 31)
lpc --info	Shows LPC module information, including: • Serial number • Firmware version	Printing LPC information (lpc command) (page 32)
meas [--metric|--ppb|--nolpc]	Outputs current measurements: • Gas measurements with linear correction in metric (µg/m^3) units • Gas measurements with linear correction in parts per billion (ppb) units • Gas measurements without linear correction in parts per million (ppm) units	Printing measurements and changing measurement output unit (meas command) (page 33)
motd	Clears screen and prints a welcome banner.	Clearing screen (motd command) (page 35)
reboot --really	Restarts the device. If gas cells are in use, rebooting the device restarts the 24 hour stabilization time.	Restarting device (reboot command) (page 36)
set <parameter>= <value>	Changes a configuration parameter.	Changing configuration parameter (set command) (page 36)
show <parameter>	Prints the value of a configuration parameter.	Printing value of configuration parameter (show command) (page 41)
status	Prints the device status and possible reason.	Printing device status (status command) (page 41)
uname [-a]	Prints the Unix name of the system (short or long name).	Printing Unix name (uname command) (page 42)
write --really	Writes the configuration changes to non-volatile memory of the device.	Saving to non-volatile memory (write command) (page 43)

To use the commands, establish a connection to AQT530.

More information

- Connecting to AQT530 over RS-232 (page 18)
- CLI parameters (page 37)

5.3 Setting date (**date** command)

To display or change the device date, use the **date** command.

```
date [YYYY-MM-DDThh:mm:ss]
```

Table 19 Parameters for **date** command

Parameter	Value	Mandatory	Description
(none)	–	–	Shows the current date and time.
YYYY-MM-DDThh:mm:ss	Year, month, day, hour, minute, second	No	Sets a new date.

Example of changing the date:

```
date 2020-12-02T13:50:36
```

Example response:

```
Date updated to 2020-12-02T13:50:36
$
```

More information

- Setting system time (page 25)

5.4 Restoring default configuration (**initconf** command)

To restore the default configuration, use the **initconf** command.

```
initconf --really
```

Table 20 Parameters for **initconf** command

Parameter	Value	Mandatory	Description
--really	Fixed	Yes	Loads the default configuration, including: • Communication settings • Configuration and linear correction

The command does not affect the following configurations:
- Device identification information
- Factory calibration information

The command does not affect connected devices, such as laser particle counter (LPC) and the temperature and humidity (TH) probe.

Example response:

```
Loading default configuration... OK!
Write the default configuration to non-volatile memory by using "write"
command.
$
```

Use the **`write`** command to save the configuration.

```
write --really
```

5.5 Printing LPC information (**`lpc`** command)

To display information about the laser particle counter (LPC), use the **`lpc`** command.

```
lpc --info
```

Table 21 Parameters for **`lpc`** command

Parameter	Value	Mandatory	Description
`--info`	Fixed	Yes	Shows LPC module information: • Serial number • Firmware version

The LPC must be connected to the system.

Querying LPC information:

```
lpc --info
```

Example response 1:

```
Serial: B3245009
SW ver: 3.0.243.ad102e1
$
```

Example response 2 (no LPC connected):

```
No LPC option installed!
$
```

5.6 Printing measurements and changing measurement output unit (**meas** command)

To display the measurements with specific units, use the **meas** command.

```
meas [--metric|--ppb|--nolc]
```

Table 22 Parameters for **meas** command

Parameter	Value	Mandatory	Description
(none)	–	No	Prints out all measurements with gases in parts per million (ppm) units and with linear correction.
`--metric`	Fixed	No	Prints out all measurements with gases in metric (µg/m^3) units and with linear correction.
`--ppb`	Fixed	No	Prints out all measurements with gases in parts per billion (ppb) units and with linear correction.
`--nolc`	Fixed	No	Prints out all measurements, without linear correction and with gases in parts per million (ppm).

```
meas
```

Example response:

```
NO2 (ppm): 0.004
SO2 (ppm): 0.562
CO (ppm): 0.077
O3 (ppm): -0.002
PM1 (ug/m3): 0.1
PM2.5 (ug/m3): 0.4
PM10 (ug/m3): 2.2
TEMP (C): 22.2
HUM (%RH): 31.2
PRES (mbar): 1012.4
Uptime (s): 10850
$
```

Example of setting measurement output to use metric (µg/m^3) units.

```
meas --metric
```

Example response:

```
NO2 (ug/m3): 5.9
SO2 (ug/m3): 912.5
CO (ug/m3): 63.6
O3 (ug/m3): -1.8
PM1 (ug/m3): 0.1
PM2.5 (ug/m3): 0.4
PM10 (ug/m3): 2.2
TEMP (C): 22.1
HUM (%RH): 31.5
PRES (mbar): 1012.3
Uptime (s): 10684
$
```

Example of setting measurement output to use parts per billion (ppb) units.

```
meas --ppb
```

Example response:

```
NO2 (ppb): 4.5
SO2 (ppb): 698.6
CO (ppb): 97.5
O3 (ppb): -0.4
PM1 (ug/m3): 0.1
PM2.5 (ug/m3): 0.4
PM10 (ug/m3): 2.2
TEMP (C): 22.3
HUM (%RH): 30.9
PRES (mbar): 1012.4
Uptime (s): 10923
$
```

Example of setting measurement output to print without linear correction, using parts per million (ppm) units.

```
meas --nolc
```

Example response:

```
NO2 (ppm): 0.004
SO2 (ppm): 0.562
CO (ppm): 0.077
O3 (ppm): -0.002
PM1 (ug/m3): 0.2
PM2.5 (ug/m3): 0.4
PM10 (ug/m3): 2.2
TEMP (C): 22.2
HUM (%RH): 31.2
PRES (mbar): 1012.4
Uptime (s): 10850
$
```

More information

- Measurement output (page 14)

5.7 Clearing screen (**motd** command)

To clear the screen and print a welcome text, use the **motd** command.

```
motd
```

Example response:

```
Vaisala Air Quality Transmitter AQT530
FW version 3.0.783.a51b18d
Copyright (c) 2020 Vaisala Oyj

$
```

5.8 Restarting device (**reboot** command)

To restart the device, use the **reboot** command.

```
reboot --really
```

If gas cells are in use, rebooting the device restarts the 24 hour stabilization time.

Table 23 Parameters for **reboot** command

Parameter	Value	Mandatory	Description
`--really`	Fixed	Yes	Confirms the restart.

Example response:

```
Resetting device in 5 seconds...


Vaisala Air Quality Transmitter AQT530
FW version 3.1.0.d5535d2
Copyright (c) 2020 Vaisala Oyj


$
```

5.9 Changing configuration parameter (**set** command)

To change device configuration, use the **set** command.

```
set <parameter>=<value>
```

Table 24 Parameters for **set** command

Parameter	Mandatory	Description
For the list of parameters, see CLI parameters (page 37).	Yes	Changes the value of selected parameter.

Example of changing the baud rate:

```
set rs485_baud=19200
```

Example response:

```
set: rs485_baud=19200
$
```

For more information, see the full list of CLI parameters.

More information

- Configuring Modbus interface (page 21)
- Configuring ASCII CSV data sending (page 22)
- Changing temperature unit (page 25)
- Adjusting linear correction (page 26)

5.9.1 CLI parameters

The following configuration and status parameters are available with AQT530.

Use the parameters with commands **set** and **show**.

Table 25 Modbus interface configuration parameters

Parameter	Type	Range	Default	Unit	Description
rs485_mode	uint8	• 0 = ASCII CSV • 1 = Modbus ASCII • 4 = Modbus RTU	4	–	RS-485 mode
rs485_addr	uint8	1 ... 253	1	–	Modbus address (not used in ASCII CSV mode)
rs485_baud	uint32	4800 ... 115200	19200	bps	RS-485 baud rate

Parameter	Type	Range	Default	Unit	Description
`rs485_databits`	uint8	7, 8	8	bits	RS-485 data bits
`rs485_parity`	Text	• `N` = None • `E` = Even • `O` = Odd	E	–	RS-485 parity Case insensitive
`rs485_stopbits`	uint8	1, 2	1	bits	RS-485 stop bits

Table 26 Temperature unit parameter

Parameter	Type	Range	Default	Unit	Description
`tempunit`	uint8	• `0` = °C • `1` = °F	0	–	Temperature unit for Modbus, ASCII CSV, and CLI interfaces

Table 27 Linear correction parameters - gases

Parameter	Type	Range	Default	Unit	Description
`co_zero`	int16	–10000 … 10000	0	ppb	CO sensor zero correction
`co_span`	uint8	1 … 255	100	%	CO sensor span correction (1 … 255 = 0.01 … 2.55)
`no_zero`	int16	–10000 … 10000	0	ppb	NO sensor zero correction
`no_span`	uint8	1 … 255	100	%	NO sensor span correction (1 … 255 = 0.01 … 2.55)
`no2_zero`	int16	–10000 … 10000	0	ppb	NO_2 sensor zero correction
`no2_span`	uint8	1 … 255	100	%	NO_2 sensor span correction (1 … 255 = 0.01 … 2.55)
`so2_zero`	int16	–10000 … 10000	0	ppb	SO_2 sensor zero correction
`so2_span`	uint8	1 … 255	100	%	SO_2 sensor span correction (1 … 255 = 0.01 … 2.55)
`h2s_zero`	int16	–10000 … 10000	0	ppb	H_2S sensor zero correction
`h2s_span`	uint8	1 … 255	100	%	H_2S sensor span correction (1 … 255 = 0.01 … 2.55)
`o3_zero`	int16	–10000 … 10000	0	ppb	O_3 sensor zero correction

Parameter	Type	Range	Default	Unit	Description
o3_span	uint8	1 ... 255	100	%	O_3 sensor span correction (1 ... 255 = 0.01 ... 2.55)

Table 28 Linear correction parameters - particles

Parameter	Type	Range	Default	Unit	Description
pm1_zero	int16	–10000 ... 10000	0	0.1 µg/m^3	PM_1 offset (zero) correction (–10000 ... 10000 = –1000.0 ... 1000.0)
pm1_span	int16	1 ... 10000	1000	‰	PM_1 span (gain) correction (1 ... 10000 = 0.001 ... 10.000)
pm25_zero	int16	–10000 ... 10000	0	0.1 µg/m^3	$PM_{2.5}$ offset (zero) correction (–10000 ... 10000 = –1000.0 ... 1000.0)
pm25_span	int16	1 ... 10000	1000	‰	$PM_{2.5}$ span (gain) correction (1 ... 10000 = 0.001 ... 10.000)
pm10_zero	int16	–10000 ... 10000	0	0.1 µg/m^3	PM_{10} offset (zero) correction (–10000 ... 10000 = –1000.0 ... 1000.0)
pm10_span	int16	1 ... 10000	1000	‰	PM_{10} span (gain) correction (1 ... 10000 = 0.001 ... 10.000)

Table 29 Device identification parameters (read-only)

Parameter	Type	Range	Unit	Description
serial	Text	–	–	AQT530 device serial number in Vaisala format

Table 30 Base module identification parameters (read-only)

Parameter	Type	Range	Unit	Description
swver	Text	–	–	Base module firmware version
hwver	Text	–	–	Base module hardware version
base_serial	Text	–	–	Base control board with gas cells
boardid	Text	–	–	Base control board without gas cells

Parameter	Type	Range	Unit	Description
`cal_date`	Text	–	–	Base module calibration date in ISO 8601 format (YYYY-MM-DD) Combination of parameter values: • `cal_day` • `cal_month` • `cal_year`
`cal_day`	uint16	1 - 31	DD	Base module calibration day of month
`cal_month`	uint16	1 - 12	MM	Base module calibration month
`cal_year`	uint16	2000 - 2100	YYYY	Base module calibration year

Table 31 Base module monitoring parameters (read-only)

Parameter	Type	Range	Unit	Description
`time`	Text	–	–	Current timestamp in ISO 8601 format
`model`	Text	–	–	Device model
`unit`	Text	–	–	Combination of • Model • Serial number • Configuration

Table 32 Gas measurement configuration and diagnostics parameters (read-only)

Parameter	Type	Range	Unit	Description
`health`	uint8	0 ... 100	%	Device health status
`meas_state`	uint8	• `0` = Gas cell stabilization ongoing • `1` = Gas cell stabilization finished	–	Gas cell stabilization status During gas cell stabilization, measurement values are invalid

Table 33 LPC parameters (read-only)

Parameter	Type	Range	Unit	Description
`lpc_serial`	Text	–	–	Serial number of laser particle counter (LPC) board

Table 34 TH probe parameters (read-only)

Parameter	Type	Range	Unit	Description
`hmp_serial`	Text	–	–	HMP110 serial number

Parameter	Type	Range	Unit	Description
`hmp_version`	Text	–	–	HMP110 firmware version number

5.10 Printing value of configuration parameter (**show** command)

To view the value of a configuration parameter, use the **show** command.

```
show <parameter>
```

Table 35 Parameters for **show** command

Parameter	Mandatory	Description
For the list of parameters, see CLI parameters (page 37).	Yes	Shows the value of selected parameter.

Example of checking the software version:

```
show rs485_baud
```

Example response:

```
show: rs485_baud=19200
$
```

For more information, see the full list of CLI parameters.

5.11 Printing device status (**status** command)

To print the device status, use the **status** command.

```
status
```

The following responses are possible:

- `Unknown (initializing)`

- `OK`
- `Degraded:` [reason]
- `Faulty:` [reason]

Example responses:

```
Device status:
OK
```

```
Device status:
Faulty: LPC malfunction
```

5.12 Printing Unix name (**uname** command)

To display the Unix name of the system, use the **uname** command.

```
uname [-a]
```

Table 36 Parameters for **uname** command

Parameter	Value	Mandatory	Description
(none)	–	–	Prints the Unix name of the system in short format.
-a	Fixed	No	Prints the Unix name of the system in long format.

Querying short name:

```
uname
```

Example response:

```
$ uname
Vaisala aqt-fw

$
```

Querying long name:

```
uname -a
```

Example response:

```
Vaisala aqt-fw 3.0.783.a51b18d 2020-12-17T08:10:40Z B210887-E avr-m1280/328p

$
```

5.13 Saving to non-volatile memory (**write** command)

To write the configuration changes to the non-volatile memory of the device, use the **write** command.

```
write --really
```

Use the **write** command together with the other commands, for example **set**.

Table 37 Parameters for **write** command

Parameter	Value	Mandatory	Description
--really	Fixed	Yes	Confirms the action.

Example response:

```
Writing current configuration to non-volatile memory... 1165 bytes written!
$
```

More information

- Configuring Modbus interface (page 21)
- Configuring ASCII CSV data sending (page 22)
- Changing temperature unit (page 25)
- Adjusting linear correction (page 26)

6. Modbus interface description

6.1 Supported Modbus protocols

AQT530 interface supports:

- Modbus ASCII mode
- Modbus RTU mode

For the general full Modbus specification, see http://www.modbus.org/.

6.2 Function codes

AQT530 supports the following Modbus function codes.

Function code	Function name
03h	Read multiple registers
06h	Write single register
2Bh / 0Eh	Read Device Identification

6.3 Device identification

AQT530 supports the following device identification objects.

AQT530 supports only individual access of device identification objects (read device ID code 04h).

Object ID	Category	Description	Example
00h	Basic	Vendor name	Vaisala
01h	Basic	Product code	AQT530
02h	Basic	Version	3.0.783.a51b18d
03h	Regular	Vendor URL	https://www.vaisala.com
04h	Regular	Product name	Vaisala Air Quality Transmitter AQT530
05h	Regular	Model name	Model: CO, NO_2, NO, O_3, LPC
80h	Extended	Serial number	A0110001

Object ID	Category	Description	Example
81h	Extended	Calibration date	2020-11-04
85h	Extended	Hardware version	B210887-E

6.4 Byte order

Individual registers follow the Modbus standard big endian byte order for 16-bit values.

Values greater than 16 bits comprise multiple registers. These values are organized as follows.

32-bit values

In this application programming interface (API), 32-bit values are supported by the means of 2 consecutive 16-bit registers. The byte order for the 32-bit values is middle-endian: bytes within words are in big-endian order, while words in multi-word values are in little-endian order.

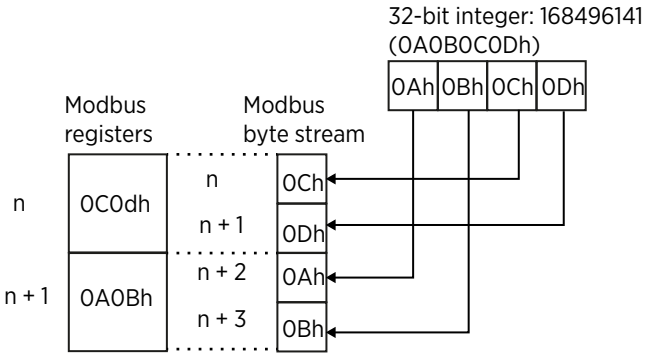


Figure 3 Byte order for 32-bit values

String fields

Several string type fields are available in this API, for example serial numbers. String fields consist of 1-byte ASCII characters packed into multiple 16-bit registers, 2 characters each.

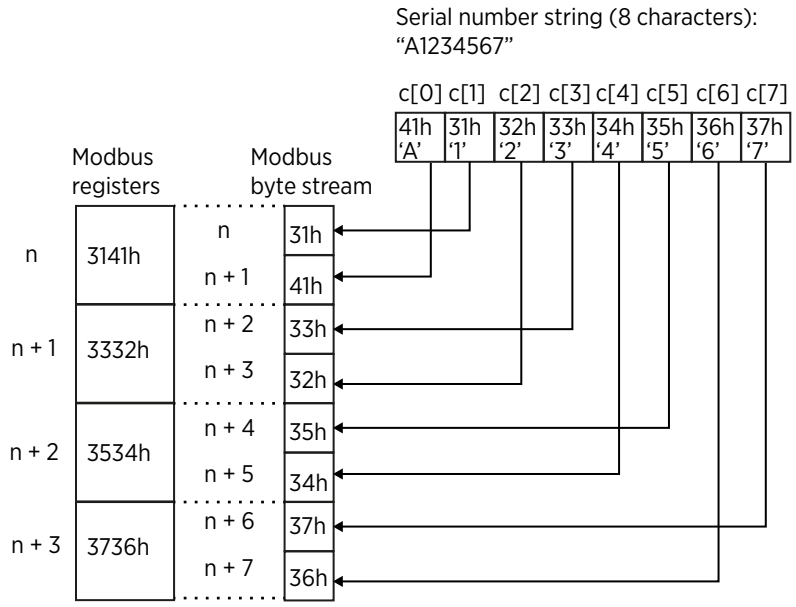


Figure 4 Byte order for string fields

6.5 Modbus register addresses for AQT530

The following register addresses are available in AQT530. The register addresses are:

- R = Read only
- RW = Read/write
- W = Write only

Table 38 Modbus register addresses

Address	RW	Type	Unit	Register count	Description
0000h	R	int16	ppb	1	Nitrogen dioxide (NO_2) concentration in parts per billion (ppb) with linear correction • For a value without linear correction, use register 006Ch • For µg/m^3 value, use register 0066h
0001h	R	int16	ppb	1	Sulfur dioxide (SO_2) concentration in parts per billion (ppb) with linear correction • For a value without linear correction, use register 006Dh • For µg/m^3 value, use register 0067h

Address	RW	Type	Unit	Register count	Description
0002h	R	int16	ppb	1	Carbon monoxide (CO) concentration in parts per billion (ppb) with linear correction • For a value without linear correction, use register 006Eh • For µg/m^3 value, use register 0068h
0004h	R	int16	ppb	1	Hydrogen sulfide (H_2S) concentration in parts per billion (ppb) with linear correction • For a value without linear correction, use register 0070h • For µg/m^3 value, use register 0069h
0005h	R	int16	ppb	1	Trioxygen (ozone) (O_3) concentration in parts per billion (ppb) with linear correction • For a value without linear correction, use register 0071h • For µg/m^3 value, use register 006Ah
0006h	R	int16	ppb	1	Nitric oxide (NO) concentration in parts per billion (ppb) with linear correction • For a value without linear correction, use register 0072h • For µg/m^3 value, use register 006Bh
0008h	R	int16	0.1 µg/m^3	1	Particulate matter $PM_{2.5}$ with linear correction • For a value without linear correction, use register 0073h
0009h	R	int16	0.1 µg/m^3	1	Particulate matter PM_{10} • For a value without linear correction, use register 0074h
000Ah	R	int16	0.1 °C / 0.1 °F	1	Air temperature (HMP110) in Celsius or Fahrenheit, depending on configuration Current configuration available from register 001Ch
000Bh	R	int16	0.1 %RH	1	Air humidity (HMP110)
000Ch	R	int16	0.1 hPa	1	Air pressure from an on-board analog pressure sensor
0013h	R	uint16	y	1	Calibration year
0014h	R	uint16	mo	1	Calibration month
0015h	R	uint16	d	1	Calibration day

<table>
<tr><th>Address</th><th>RW</th><th>Type</th><th>Unit</th><th>Register count</th><th>Description</th></tr>
<tr><td>0016h</td><td>R</td><td>uint16</td><td>-</td><td>1</td><td>Unit configuration flags (bits/description)<table><tr><th>Bits</th><th>Description</th></tr><tr><td>15:2</td><td>Reserved</td></tr><tr><td>1</td><td>LPC configuration:
• 0 = No LPC
• 1 = LPC</td></tr><tr><td>0</td><td>Reserved</td></tr></table></td></tr>
<tr><td>001Ah</td><td>R</td><td>int16</td><td>s</td><td>1</td><td>Seconds elapsed after last measurement data update</td></tr>
<tr><td>001Bh</td><td>R</td><td>int16</td><td>-</td><td>1</td><td>Gas measurement validity
• 0 = Measurement is not valid
• 1 = Measurement is valid (includes 24-hour stabilization time after power-up and temperature is within valid range (below 38.0 °C)
Flags can be read separately from registers 0033h and 0034h</td></tr>
<tr><td>001Ch</td><td>R</td><td>uint16</td><td>-</td><td>1</td><td>Temperature unit configuration
• 0 = Celsius (default)
• 1 = Fahrenheit</td></tr>
<tr><td>001Fh</td><td>R</td><td>int16</td><td>%</td><td>1</td><td>Device health index (combined percentage of usage of sensor cells, decreases from 100 %)
• 100 % = Full health
• 0 % = All sensors over-aged</td></tr>
<tr><td>0033h</td><td>R</td><td>int16</td><td>-</td><td>1</td><td>Gas cell stabilization invalidation flag
• 0 = 24-h stabilization time after power-up has passed
• 1 = 24-h stabilization time after power-up not passed and measurement is invalid</td></tr>
<tr><td>0034h</td><td>R</td><td>int16</td><td>-</td><td>1</td><td>Gas cell temperature invalidation flag
• 0 = Cell temperature valid
• 1 = Cell temperature too high (≥ 38.0 °C) and measurement invalid</td></tr>
<tr><td>0036h</td><td>R</td><td>int16</td><td>-</td><td>1</td><td>Humidity compensation (gas compensation)
• 0 = Humidity compensation disabled
• 1 = Humidity compensation enabled</td></tr>
<tr><td>0037h</td><td>R</td><td>int16</td><td>0.1 µg/m^3</td><td>1</td><td>Particulate matter PM_1 with linear correction
• For a value without linear correction, use register 0075h</td></tr>
</table>

Address	RW	Type	Unit	Register count	Description	
004Bh	R	int16	-	1	Status reported by device • 0 = Unknown. Status information not yet available, starting up • 1 = Ok. Operating normally • 2 = Degraded. Operating but functionality degraded • 3 = Faulty For more information, see status code from register 004Ch	
004Ch	R	int16	-	1	Status codes • 0 = No specific status to report • 1 = LPC malfunction • 2 = HMP110 malfunction	
004Fh	R	uint16	-	1	AQT base module firmware version major number	
0050h	R	uint16	-	1	AQT base module firmware version minor number	
0051h	R	uint16	-	1	AQT base module firmware version build number	
0052h	R	uint32	-	2	AQT base module firmware version hash number	
0054h	R	uint16	-	1	HMP device firmware version major number	
0055h	R	uint16	-	1	HMP device firmware version minor number	
0056h	R	uint16	-	1	HMP device firmware version revision number	
0057h	RW	uint16	y	1	System time year	• Read = System time is read into cache when 0057h is read. Other registers return values from cache • Write = Register value is written to cache, and system time is updated when 005Ch is written
0058h	RW	uint16	mo	1	System time month	
0059h	RW	uint16	d	1	System time day	
005Ah	RW	uint16	h	1	System time hours	
005Bh	RW	uint16	min	1	System time minutes	
005Ch	RW	uint16	s	1	System time seconds	
0065h	RW	uint16	-	1	Select the ppb to µg/m^3 conversion type for registers 0066h - 006Bh • 0 = EU standard (temperature t = +20 °C, pressure p = 1013.25 hPa) (default) • 1 = International standard (WHO, temperature t = +25 °C, pressure p = 1013.25 hPa) • 2 = Temperature based dynamic (ambient temperature from HMP110, pressure p = 1013.25 hPa)	

Address	RW	Type	Unit	Register count	Description
0066h	R	int16	0.1 µg/m^3	1	Nitrogen dioxide (NO_2) concentration calculated from ppb value with linear correction (register 0000h) Calculated by means of selected conversion type . Calculation method is selected with register 0065h
0067h	R	int16	0.1 µg/m^3	1	Sulfur dioxide (SO_2) concentration calculated from ppb value with linear correction (register 0001)h Calculated by means of selected conversion type . Calculation method is selected with register 0065h
0068h	R	int16	1 µg/m^3	1	Carbon monoxide (CO) concentration calculated from ppb value with linear correction (register 0002h) Calculated by means of selected conversion type. Calculation method is selected with register 0065h
0069h	R	int16	0.1 µg/m^3	1	Hydrogen sulfide (H_2S) concentration calculated from ppb value with linear correction (register 0003h) Calculated by means of selected conversion type. Calculation method is selected with register 0065h
006Ah	R	int16	0.1 µg/m^3	1	Trioxygen (ozone) (O_3) concentration calculated from ppb value with linear correction (register 0005h) Calculated by means of selected conversion type. Calculation method is selected with register 0065h
006Bh	R	int16	0.1 µg/m^3	1	Nitric oxide (NO) concentration calculated from ppb value with linear correction (register 0006h) Calculated by means of selected conversion type. Calculation method is selected with register 0065h
006Ch	R	int16	ppb	1	Nitrogen dioxide (NO_2) concentration in parts per billion (ppb), without linear correction • For a value with linear correction, use register 0000h
006Dh	R	int16	ppb	1	Sulfur dioxide (SO_2) concentration in parts per billion (ppb), without linear correction • For a value with linear correction, use register 0001h
006Eh	R	int16	ppb	1	Carbon monoxide (CO) concentration in parts per billion (ppb), without linear correction • For a value with linear correction, use register 0002h

Address	RW	Type	Unit	Register count	Description
0070h	R	int16	ppb	1	Hydrogen sulfide (H_2S) concentration in parts per billion (ppb), without linear correction • For a value with linear correction, use register 0004h
0071h	R	int16	ppb	1	Trioxygen (ozone) (O_3) concentration in parts per billion (ppb), without linear correction • For a value with linear correction, use register 0005h
0072h	R	int16	ppb	1	Nitric oxide (NO) concentration in parts per billion (ppb), without linear correction • For a value with linear correction, use register 0006h
0073h	R	int16	0.1 µg/m^3	1	Particulate matter $PM_{2.5}$, without linear correction • For a value with linear correction, use register 0008h
0074h	R	int16	0.1 µg/m^3	1	Particulate matter PM_{10}, without linear correction • For a value with linear correction, use register 0009h
0075h	R	int16	0.1 µg/m^3	1	Particulate matter PM_1, without linear correction • For a value without linear correction, use register 0037h
0076h	R	int16	-	1	LPC data state • 0 = LPC data not ready • 1 = LPC data ready
007Bh	R	int16	-	1	LPC humidity invalidation flag Combines all PM values from registers 007Ch - 007Eh • 0 = LPC humidity ok • 1 = LPC measurement may be invalid due to high humidity
007Ch	R	int16	-	1	LPC humidity invalidation flag for PM_1 value • 0 = LPC humidity is ok • 1 = LPC PM_1 measurement may be invalid due to high humidity
007Dh	R	int16	-	1	LPC humidity invalidation flag for $PM_{2.5}$ value • 0 = LPC humidity is ok • 1 = LPC $PM_{2.5}$ measurement may be invalid due to high humidity
007Eh	R	int16	-	1	LPC humidity invalidation flag for PM_{10} value • 0 = LPC humidity is ok • 1 = LPC PM_{10} measurement may be invalid due to high humidity
007Fh	RW	int16	min	1	LPC measurement interval in minutes (2 - 255) Default: 10 minutes

Address	RW	Type	Unit	Register count	Description
0086h	RW	int16	%	1	CO linear correction gain value (1 ... 255, corresponding gain values 0.01 ... 2.55)
0087h	RW	int16	%	1	O_3 linear correction gain value (1 ... 255, corresponding gain values 0.01 ... 2.55)
0088h	RW	int16	%	1	NO_2 linear correction gain value (1 ... 255, corresponding gain values 0.01 ... 2.55)
0089h	RW	int16	%	1	SO_2 linear correction gain value (1 ... 255, corresponding gain values 0.01 ... 2.55)
008Ah	RW	int16	%	1	NO linear correction gain value (1 ... 255, corresponding gain values 0.01 ... 2.55)
008Bh	RW	int16	%	1	H_2S linear correction gain value (1 ... 255, corresponding gain values 0.01 ... 2.55)
008Ch	RW	int16	ppb	1	CO linear correction offset –10000 ... 10000
008Dh	RW	int16	ppb	1	O_3 linear correction offset –10000 ... 10000
008Eh	RW	int16	ppb	1	NO_2 linear correction offset –10000 ... 10000
008Fh	RW	int16	ppb	1	SO_2 linear correction offset –10000 ... 10000
0090h	RW	int16	ppb	1	NO linear correction offset –10000 ... 10000
0091h	RW	int16	ppb	1	H_2S linear correction offset –10000 ... 10000
0092h	RW	int16	‰	1	$PM_{2.5}$ linear correction gain value (per mille) Valid range: 1 ... 10000 (0.001 ... 10.000)
0093h	RW	int16	‰	1	PM_{10} linear correction gain value (per mille) Valid range: 1 ... 10000 (0.001 ... 10.000)
0094h	RW	int16	0.1 µg/m^3	1	$PM_{2.5}$ linear correction offset Valid range: -10000 ... 10000 (-1000.0 ... 1000.0)
0095h	RW	int16	0.1 µg/m^3	1	PM_{10} linear correction offset Valid range: -10000 ... 10000 (-1000.0 ... 1000.0)
0096h	RW	int16	‰	1	PM_1 linear correction gain value (per mille) Valid range: 1 ... 10000 (0.001 ... 10.000)
0097h	RW	int16	0.1 µg/m^3	1	PM_1 linear correction offset Valid range: -10000 ... 10000 (-1000.0 ... 1000.0)
0098h	R	uint32	s	2	Device uptime in seconds from last power-up

Address	RW	Type	Unit	Register count	Description
00B4h	R	char[8]	-	4	AQT product device serial number (8-character string)
00B8h	R	char[8]	-	4	HMP serial number (8-character string)
00BCh	R	char[8]	-	4	LPC serial number (8-character string)
00C3h	R	int16	-	1	Circuit board serial number, part 1
00C4h	R	int16	-	1	Circuit board serial number, part 2
00C5h	R	int16	-	1	Circuit board serial number, part 3
00F4h	R	uint16	-	1	LPC firmware version major number
00F5h	R	uint16	-	1	LPC firmware version minor number
00F6h	R	uint16	-	1	LPC firmware version build number
00F7h	R	uint32	-	2	LPC firmware version hash number
00FAh	W	uint16	-	1	Reset device • 1 = Reset device • Other values = no effect
00FEh	W	uint16	-	1	Write configuration to the onboard EEPROM memory • 1 = Write configuration • Other values = no effect

6.6 Setting up system time over Modbus interface

You can read and write the system time over the Modbus interface.

1. To read system time, read register 0057h first.
   This reads the current system time to a cache and prevents the value from changing when other system time registers are accessed.
2. After reading the register 0057h, read registers 0058h ... 005Ch to get month, day of month, hours, minutes, and seconds.
   The reading order of these registers is free.
3. To write the system time, set the year, month, day of month, hours, and minutes by the means of the registers 0057h ... 005Bh.
   This writes the date to the cache.
4. To finish updating the system time, write seconds to register 005Ch.
   This updates the system time immediately by applying all the recent values from the cache.

7. Troubleshooting

If the product does not work as it should, check all cables and connectors.

Check the maintenance needs.

If the failure persists, contact Vaisala technical support.

Table 39 Troubleshooting AQT530

Problem	Probable cause	Remedy
Measurement failure or irrelevant data values.	The settings are incorrect.	Contact Vaisala technical support.
	There is a hardware failure.	
No response to any commands.	Wiring is wrong.	Check the wiring.
	Operational power is not connected.	Check the operating voltage.
	Baud rate or other serial property of the device is different from the host.	Connect the serial cable and check the serial port settings of the device with a terminal program.
No response to Modbus query.	RS-485 wiring is wrong.	Do the following: 1. Configure human-readable ASCII CSV report, see Configuring ASCII CSV data sending (page 22). 2. Verify that ASCII CSV report is sent every 60 s to RS-485 port. If data is not coming through, change the wiring of pins 5 and 6. See M12 pinout and wiring (page 56). 3. Configure original Modbus mode (Modbus RTU or Modbus ASCII) back to the device, see Configuring Modbus interface (page 21).

7.1 Writing a problem report

When troubleshooting the product, write a problem report including:

- What failed (what worked / did not work)?
- Where did it fail (location and environment)?
- When did it fail (date, immediately / after a while / periodically / randomly)?
- How many failed (only one defect / other same or similar defects / several failures in one unit)?
- What was done when the failure was noticed?
- What was connected to the product and to which connectors?

- Input power source type, voltage, and list of other items (such as lighting, heaters, and motors) that were connected to the same power output.
- Are all parts connected and grounded properly? Take a photo to help the troubleshooting.

Appendix A. M12 pinout and wiring

The M12 connector provides RS-232 and RS-485 serial interfaces.

The 8-pin M12 connector is located on the side of the air quality transmitter and provides power and data to the transmitter.

The following pinout is from the side of AQT530, not the cable.

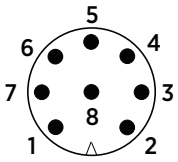


Figure 5 Pinout of M12 connector

Table 40 Pinout of M12 connector

M12 pin	Wire color	RS-232	RS-485
1	White	Data GND	Data GND
2	Brown	RX (input)	–
3	Green	TX (output)	–
4	Yellow	No connection (floating)	
5	Gray	–	B (-)
6	Pink	–	A (+)
7	Blue	Power GND	Power GND
8	Red	Power 10 ... 25 V DC	Power 10 ... 25 V DC
–	Black	Cable GND	Cable GND

RS-232 serial interface

The RS-232 interface is used mainly for the maintenance connection.

RS-485 serial interface

The RS-485 interface is used mainly for the data connection.

RS-485 supports Modbus ASCII and Modbus RTU, and ASCII formatted text with comma separated values (CSV).

Appendix B. OSS licenses

This product contains open source software (OSS) components. Such OSS is governed by the terms and conditions of the applicable OSS licenses, and you are bound by the terms and conditions of such licenses in connection with your use and distribution of the OSS in this product.

If you have any questions about the open source software, contact us at opensourcesw@vaisala.com.

To request a copy of certain open source code as required by certain applicable OSS licenses, send an email to opensourcesw@vaisala.com with the subject line OSS source code request.

List of installed packages and their respective versions and licenses:

Table 41 AQT530 OSS license

Library	Version	License
avr-lic	2.0.0	Modified BSD License / https://www.nongnu.org/avr-libc/LICENSE.txt

License text

```
The contents of avr-libc are licensed with a Modified BSD License.

All of this is supposed to be Free Software, Open Source, DFSG-free,
GPL-compatible, and OK to use in both free and proprietary applications.

See the license information in the individual source files for details.

Additions and corrections to this file are welcome.

****************************************************************************
Portions of avr-libc are Copyright (c) 1999-2010
Keith Gudger,
Bjoern Haase,
Steinar Haugen,
Peter Jansen,
Reinhard Jessich,
Magnus Johansson,
Artur Lipowski,
Marek Michalkiewicz,
Colin O'Flynn,
Bob Paddock,
Reiner Patommel,
Michael Rickman,
Theodore A. Roth,
Juergen Schilling,
Philip Soeberg,
Anatoly Sokolov,
Nils Kristian Strom,
Michael Stumpf,
Stefan Swanepoel,
Eric B. Weddington,
Joerg Wunsch,
Dmitry Xmelkov,
The Regents of the University of California.
All rights reserved.

   Redistribution and use in source and binary forms, with or without
   modification, are permitted provided that the following conditions are met:

   * Redistributions of source code must retain the above copyright
     notice, this list of conditions and the following disclaimer.

   * Redistributions in binary form must reproduce the above copyright
     notice, this list of conditions and the following disclaimer in
     the documentation and/or other materials provided with the
     distribution.

   * Neither the name of the copyright holders nor the names of
     contributors may be used to endorse or promote products derived
```

```
      from this software without specific prior written permission.

   THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
   AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
   IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
   ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
   LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
   CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
   SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
   INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
   CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
   ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
   POSSIBILITY OF SUCH DAMAGE.

*****************************************************************************
```

Warranty

For standard warranty terms and conditions, see www.vaisala.com/warranty.

Please observe that any such warranty may not be valid in case of damage due to normal wear and tear, exceptional operating conditions, negligent handling or installation, or unauthorized modifications. Please see the applicable supply contract or Conditions of Sale for details of the warranty for each product.

Technical support

Contact Vaisala technical support at helpdesk@vaisala.com. Provide at least the following supporting information as applicable:

- Product name, model, and serial number
- Software/Firmware version
- Name and location of the installation site
- Name and contact information of a technical person who can provide further information on the problem

For more information, see www.vaisala.com/support.

Recycling

Recycle all applicable material.

Follow the statutory regulations for disposing of the product and packaging.

VAISALA

www.vaisala.com