DTR.PCE.PRE.SGE.MODBUS*(ENG)*

APLISENS

MANUFACTURE OF PRESSURE TRANSMITTERS
AND CONTROL INSTRUMENTS

Part 1:

USER'S MANUAL

Part 2:

Selected issues concerning the operation of transmitters using "Modbus Configurator"

SUPPLEMENT TO DTR.PCE.PRE-28.02(ENG); DTR.SG…05(ENG)

PRESSURE TRANSMITTERS
TYPE: **PCE-28/MODBUS, PRE-28/MODBUS**

SMART LEVEL PROBES
TYPE: **SGE-25.SMART/MODBUS;**
SGE-25S.SMART/MODBUS

Edition C2

WARSAW AUGUST 2015

APLISENS S.A., 03-192 Warszawa, ul. Morelowa 7
tel. +48 22 814 07 77; fax +48 22 814 07 78
www.aplisens.pl, e-mail: aplisens@aplisens.pl

Symbols used

Symbol	Description
⚠	Warning to proceed strictly in accordance with the information contained in the documentation in order to ensure the safety and full functionality of the device.
i	Information particularly useful during installation and operation of the device.
🗑	Information on disposal of used equipment.

BASIC REQUIREMENTS AND SAFE USE

⚠

- **The manufacturer will not be liable for damage resulting from incorrect installation, failure to maintain the device in a suitably functional condition, or use of the device other than for its intended purpose.**
- Installation should be carried out by qualified personnel having the necessary authorisation to install electrical and pressure measuring devices. The installer is responsible for performing the installation in accordance with these instructions and with the electromagnetic compatibility and safety regulations and standards applicable to the type of installation.
- The device should be configured appropriately for the purpose for which it is to be used. Incorrect configuration may cause erroneous functioning, leading to damage to the device or an accident.
- In systems with pressure transmitters there exists, in case of leakage, a risk to personnel on the side where the medium is under pressure. All safety and protection requirements must be observed during installation, operation and inspections.
- If a device is not functioning correctly, disconnect it and send it for repair to the manufacturer or to a firm authorised by the manufacturer.

⚠

In order to minimise the risk of malfunction and associated risks to personnel, the device is not to be installed or used in particularly hostile conditions, where the following risks occur:

- possibility of mechanical impacts, excessive shocks and vibration;
- excessive temperature fluctuation;
- condensation of water vapour, dust, icing.

Changes in the production of transmitters may precede a paper updating for the user. The current user manuals are available at http www.aplisens.pl

PART 1

PART 2

PART 1

1. Introduction

This User Manual concerns the **PCE-28/Modbus**, **PRE-28/Modbus** pressure transmitters and the **SGE-25.Smart/Modbus**, **SGE-25S.Smart/Modbus** level probes with the output signal Modbus RTU/ /4-20 mA and manufactured after July 2014, provided with at least **software version 14** and **PCB** (printed circuit board) **version 48**. It is a supplement to the User Manuals numbered: DTR.PCE.PRE-28.02*(ENG)*, DTR.SG..05*(ENG)*. The permissible environment and operating parameters, product materials, process connections, degree of protection of the housing and measuring ranges are the same as in **PCE-28, PRE-28, SGE-25.Smart, SGE-25S.Smart** products.

Process variables and other configuration data can be read using the protocol Modbus RTU. To perform configuration or calibration of the transmitter, you need a PC, the converter RS-232/RS-485, USB/RS-485 port and the configuration software Modbus Configurator from Aplisens. In addition to the interface RS485/Modbus RTU, the transmitters provided with at least **software version 14 and PCB version 48** can be used in current loop mode (4-20 mA).

2. Technical specifications

2.1. PCE-28/Modbus, PRE-28/Modbus, Metrological parameters

Accuracy of 4-20 mA digital and analogue output:	±0.1% (basic range)
Long-term stability	≤ accuracy per 2 years (for basic range)
Thermal error of 4-20 mA digital and analogue output:	< ±0.08% (FSO) / 10ºC max ±0.25% (FSO) (for the whole thermal compensation range)
Thermal compensation range	-25...80ºC (other ranges available on request)
Additional electronic damping	0...30 s
Pressure ranges, other parameters	see PCE-28, PRE-28 acc. to DTR.PCE.PRE.02*(ENG)*

2.2. SGE-25.Smart/Modbus, SGE-25S.Smart/Modbus, Metrological parameters

Stability, accuracy and thermal error	see PCE-28/Modbus, PRE-28/Modbus
Compensation temperature range	-25...80ºC (other ranges available on request)
Measuring ranges, other parameters	see SGE-25.Smart, SGE-25S.Smart acc. to DTR.SG…05*(ENG)*

2.3. PCE-28/Modbus, PRE-28/Modbus, SGE-25.Smart/Modbus, SGE-25S.Smart/Modbus, Electrical parameters

Power supply voltage in MODBUS RTU mode	4...28 V DC
Power supply voltage in 4-20 mA mode	5...28 V DC
Power consumption	<3.6 mA in Modbus mode
Output signal	MODBUS RTU or current loop 4-20 mA DC (Transmitter D/A 16 bit /28 Hz, PWM)
Electrical connection	5-pin M12 connector (male) or cable
Digital transmission range	1200 m (shielded twisted-pair cable)
Address space	1...247 addresses
Max devices per bus	256
Transfer rate	1200, 2400, 4800, 9600, 19200, 28800, 38400, 57600, 115200 bps
Transfer parity control	no parity, odd, even
Data load per transfer frame	11 bit (8N2, 8E1, 8O1)
Response time	3-20ms, depending on the speed of transmission

The transmitters comply with the requirements of standard EN 61000-4-2,3,4,5,6,11.

2.4. Data layout (registers) in address space

Read-only data

Register	Address (hex)	Purpose	Notes	Format	Bytes (2 bytes per register)
1	0x0000	**User specific**	% of the controlled set range	IEEE754	4 bytes (2 registers)
3	0x0002	**Pressure of sensor 1**	Pressure or level process variable	IEEE754	4 bytes (2 registers)
5	0x0004	**Pressure of sensor 2**	Constant 0	IEEE754	4 bytes (2 registers)
7	0x0006	**Temperature of sensor 1**	Sensor 1 temperature process variable at ºC	IEEE754	4 bytes (2 registers)
9	0x0008	**CPU temperature**	CPU temperature process variable at ºC	IEEE754	4 bytes (2 registers)
11	0x000A	**Temperature of sensor 2**	Constant 0ºC	IEEE754	4 bytes (2 registers)
13	0x000C	---------------------------	------------------------	-----------------------	4 bytes (2 registers)
15	0x000E	---------------------------	------------------------	-----------------------	4 bytes (2 registers)
17	0x0010	**User specific**	1/100% of the controlled set range	Signed 16-bit int	2 bytes (1 register)
18	0x0011	**Pressure of sensor 1**	Integer, 1/100 of the pressure or level unit	Signed 16-bit int	2 bytes (1 register)
19	0x0012	**Pressure of sensor 2**	Value 0 1/100 of the selected unit	Signed 16-bit int	2 bytes (1 register)
20	0x0013	**Temperature of sensor 1**	Integer, 1/100ºC	Signed 16-bit int	2 bytes (1 register)
21	0x0014	**CPU temperature**	Integer, 1/100ºC	Signed 16-bit int	2 bytes (1 register)
22	0x0015	**Temperature of sensor 2**	Value 0 1/100 at ºC	Signed 16-bit int	2 bytes (1 register)
23	0x0016	**Pressure unit**	Pressure or level unit	Unsigned 16-bit int	2 bytes (1 register) See DTR.PCE.PRE.SGE.MODBUS.*(ENG*), Part 1, table in section 2.4.1
24	0x0017	---------------------------	------------------------	-----------------------	2 bytes (1 register)
25	0x0018	**Upper sensor limit**	Upper limit of the standard range	IEEE754	4 bytes (2 registers)
27	0x001A	**Lower sensor limit**	Lower limit of the standard range	IEEE754	4 bytes (2 registers)
29	0x001C	**Damping value** seconds (s)		IEEE754	4 bytes (2 registers)
31	0x001E	**Response delay value** milliseconds (ms)		Unsigned 16-bit int	2 bytes (1 register)
32	0x001F	**Modbus address**	1…247	Unsigned 8-bit int	2 bytes (1 register)
33	0x0020	**Identity register**		Unsigned 8-bit int	6 bytes (3 registers)
36	0x0023	**Status register**		8-bit flags	2 bytes (1 register)

* Fields shaded in gray are not active in the specified models.

2.4.1. Codes of pressure units.

Unit	Value (decimal number)	Unit	Value (decimal number)
atm	14	mbar	8
bar	7	mmH2O w 4ºC	239
FtH2O	3	mmH2O	4
g/cm²	9	mH2O w 4ºC	171
InH2O w 4ºC	238	mmHg	5
InH2O	1	MPa	237
InHg	2	Pa	11
kg/cm²	10	psi	6
kPa	12	torr	13

2.5. Modbus status register

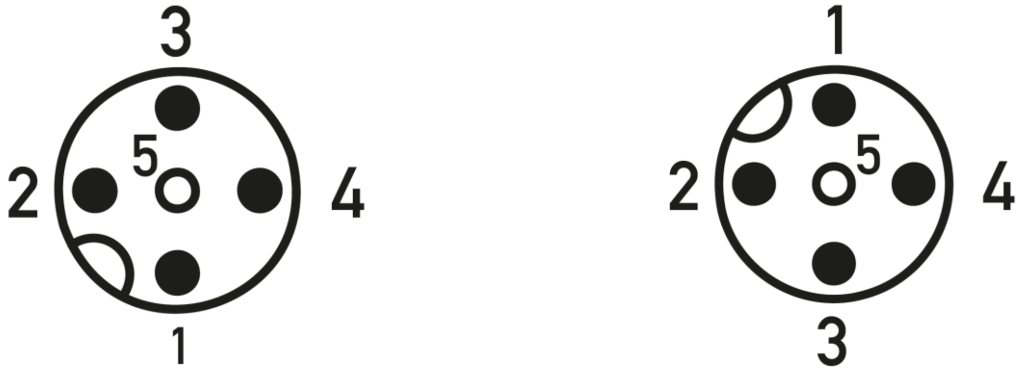
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	SV,TV,FV out of limit range	PV out of limit range	0	0	0	0	0

2.6. Error codes supported by Modbus

Error code	*Name*	*Description*
1	Invalid function code	This error shows that a function code is not supported by the device. Function code No. 3, i.e. Read Holding Registers, is currently implemented.
2	Invalid data address	This error shows that an address is out of the range 0x00÷0x23.
3	Invalid data load	Invalid number of registers to be read as specified in the frame.
4	Transmitter failure	Transmitter hardware error.
8	Memory error	RAM or FLASH memory error.

3. Connection via PM12 connector or cables

Output	Function	PC(R)E-28	SGE-25(S).Smart
		PM12 connector (male)	Cable
Power supply and 4-20 mA current loop in configuration mode	GND	3	black
	+Vcc (4-28 V)	2	red
Digital	RS-485A	4	blue
	RS-485B	5	yellow
Screen	–	–	green



M12 connector (female) **M12 connector (male)**

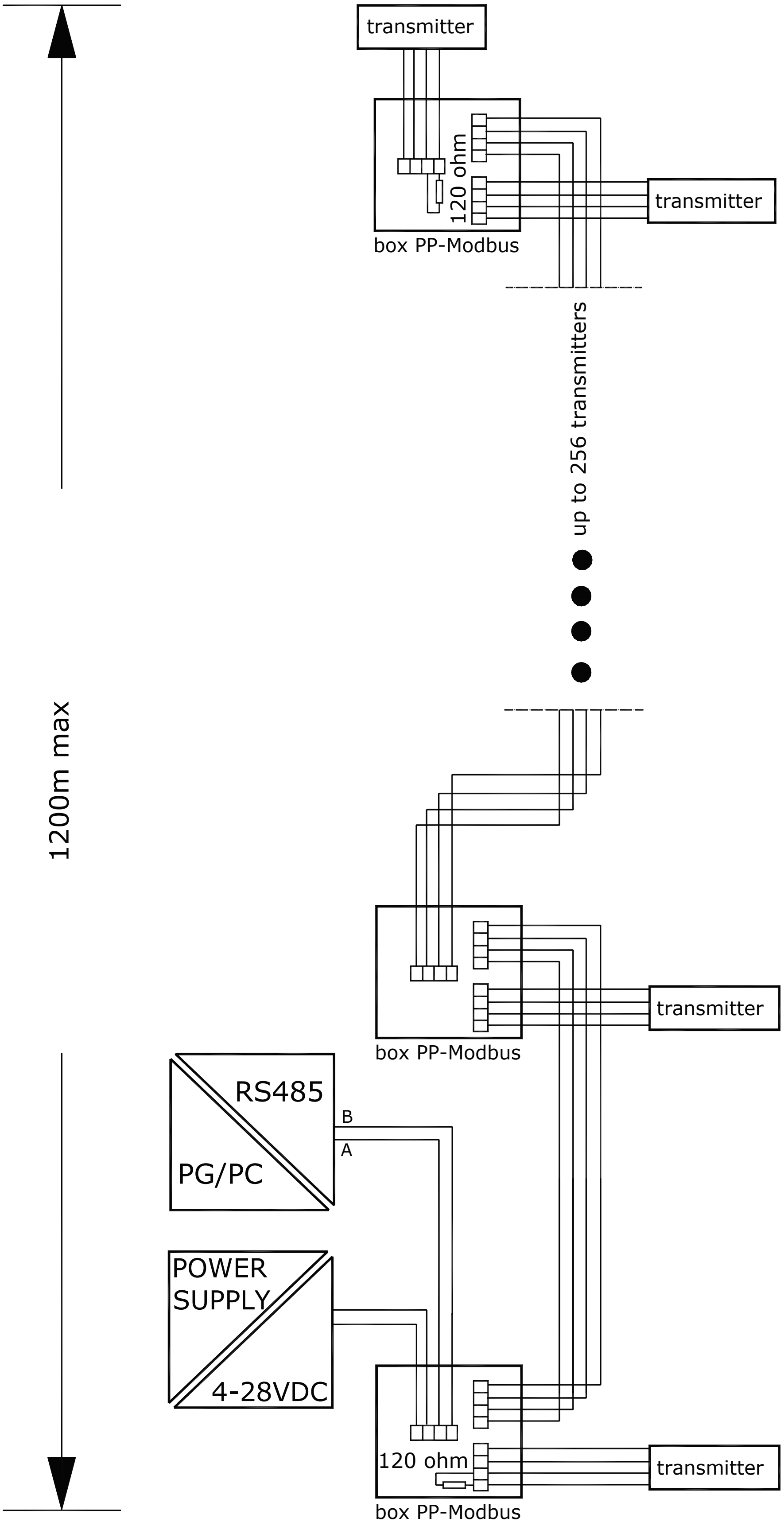
Fig. Connector leads (front view)

Notes:

- The transmitters **PC(R)E-28/Modbus and SGE-25(S).Smart/Modbus** from Aplisens have two operating modes, configuration mode and Modbus mode. The configuration mode is used for changing settings and detailed diagnostics of the transmitter. In this mode you can also activate a current loop compatible with a 4-20 mA current output in transmitters manufactured after July 2014 and provided with at least software version 14 and PCB version 48.
- If you intend to use the operating mode 4-20 mA, leave the transmitter in configuration mode after setting parameters using Modbus Configurator.
- Additional settings for the 4-20mA operating mode exceeding the range of options of "Modbus Configurator" are possible with the Raport 2 application.
- If you intend to use Modbus mode, switch the transmitter into Modbus mode after setting parameters using Modbus Configurator.
- The measuring range for 1/100 of a non-decimal unit, including its sign, lies in the range of -32767 to 32767 units. If this range is exceeded, a false value will be read. This is valid for the following units: g/cm^2, mbar, mmH$_2$O at 4ºC, mmH$_2$O, mmHg, Pa, torr.
- A resistor adapting 120 Ώ between Digital A and B outputs should be used for all transfer rates (especially for 115200 bps).

3.1. When using devices with MODBUS RTU output signal, it is convenient to use appropriate junction boxes. We recommend using Aplisens junction boxes designated PP-Modbus.

4. Connecting transmitters to Modbus network



PART 2

1. Purpose

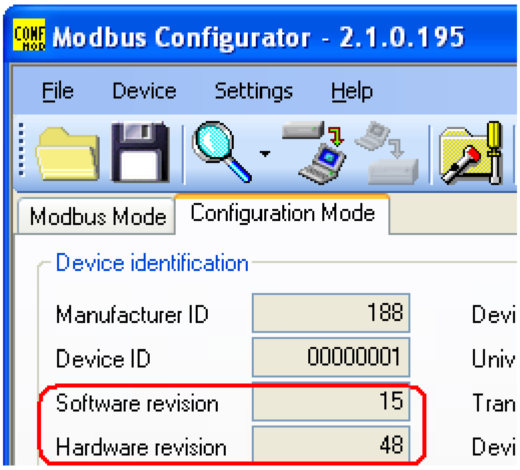
This chapter describes a communication protocol used in devices provided with a Modbus digital output, manufactured by APLISENS S.A.:

- Pressure transmitters: PCE-28/Modbus, PRE-28/Modbus;
- Level probes: SGE-25.Smart/Modbus, SGE-25S.Smart/Modbus,

with PCB version 48 and software at least from version 14.

The above protocol conforms to the standard published by Modbus organization (see http://www.modbus.org/) in the documents **MODBUS Application Protocol Specification v1.1, MODBUS over Serial Line Specification and Implementation Guide v1.02 and MODBUS Application Protocol Specification v1.1.**

The number of the transmitter's PCB and software versions can be checked in configuration mode, in the frame "**Device identification**" via "Modbus Configurator" from APLISENS.



Device identification

Manufacturer ID 188

Device ID 00000001

Software revision 15

Hardware revision 48

2. Physical layer for RS485 data transfer

2.1. Introduction

The transmitter is connected to the system via RS485 serial interface. This interface guarantees high resistance to interferences and flexible bus structure, e.g. multiple slave devices can be managed via a single master device. An RS485 half-duplex mode has been implemented to reduce the number of necessary communication cables. This means that 2 communication cables are required. The other 2 cables are used for power supply distribution.

2.2. Description

To ensure correct operation of multiple devices on one serial communication bus, appropriate leads (RS485A, RS485B, GND and +Vcc) should be connected in parallel to the bus. Before connecting to the bus, a unique address must be defined for each device.

A network of up to 1,200 meters with max 247 Modbus devices can be set up. Each cable junction from the bus can be up to 15 m long.

The cables should conform to EIA RS485.

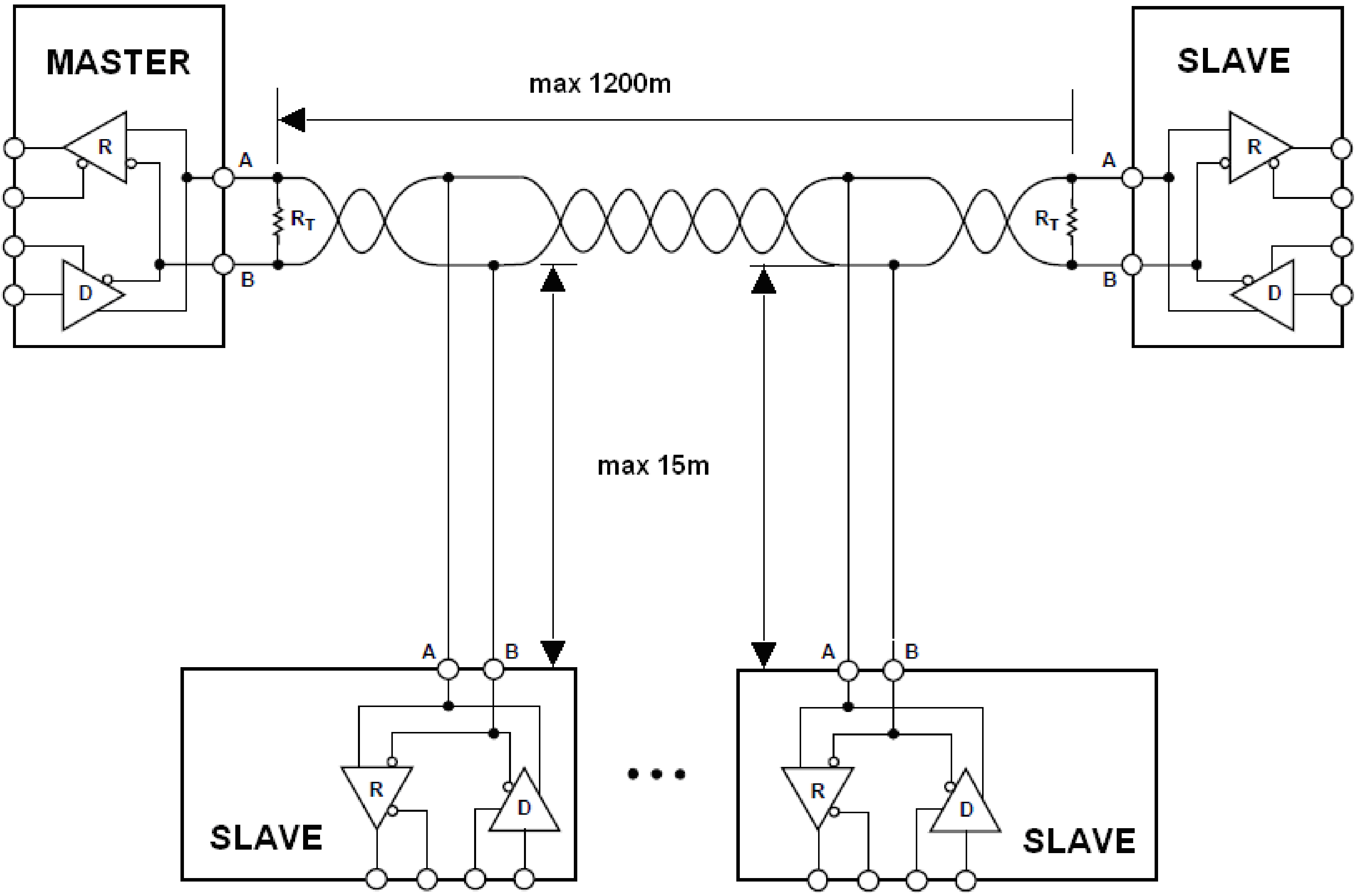


Fig. Modbus network configuration (example).

2.3. Details of RS485 „half-duplex”

To ensure the best possible efficiency in industrial environment, the transmitters from Aplisens are provided with RS485 transceivers with appropriately customized operating parameters. To provide compatibility and the best possible working conditions, the master transceiver should conform to the specifications below.

2.3.1. Limited digital signal incremental rate.

To eliminate oscillations and interferences, the output voltage slew rate of the signals from the transmitters is limited by the use of appropriate RS485 transceivers. This makes it possible to use standard cable-based connections and/or customized topologies (e.g. branches up to 15 m long).

2.3.2. „Fail safe” mode

The fail-safe mode means a strictly defined level of received signals – both in open bus connection and **closed bus** connection modes. It is very important in half-duplex mode when all the devices connected to the bus are in receiving mode. This approach eliminates the need to use external polarization resistors. Power losses in RS485 transceivers caused by e.g. **bus short-circuit**, are limited by the use of thermal safety features in transmitting and receiving circuits.

2.3.3. 1/16 of bus load

Input impedance in RS485 transceivers used in the transmitters from Aplisens, in receiving mode, is higher than the standard impedance to enable connecting up to 256 devices to the bus.

2.3.4. Line transmission

The terminating resistors on the RS485 bus should be connected between the points A and B at the beginning and at the end of the communication bus. When working with long transmission lines at the highest transfer rates, resistor value should correspond to cable impedance which typically is 120 Ω. With shorter bus segments and lower transfer rates, resister with higher values, e.g. 1 kΩ, can be used to reduce current fluctuations in the transmission line (with 2 resistors of 120 Ω, the amplitude can be as high as ca 50 mA). At least one resistor should be used to ensure stable communication.

If a fail-safe device is connected to the master bus, the electromagnetic environment in which the communication bus is located is free of interferences and the transmission line is relatively short (several meters) and terminations resistors are not required.

The APLISENS transmitters described in this document are NOT provided with terminations resistors.

2.3.5. Line polarization

Polarization resistors can be used for ensuring permanently defined voltage levels in relation to power supply on the communication bus. The resistors are connected respectively:

- between point A on the bus and +Vcc point;
- between point B on the bus and GND point.

They are necessary if any of the transceivers of the devices connected to the bus is not a fail-safe transceiver. The resistance of such resistors depends on the power supply connected to the bus, input current in the connected devices operated in receiving mode, and it should be calculated or determined experimentally. It usually ranges from 450 to 650 Ω. If the master device is in fail-safe mode and only works with Modbus transmitters from Aplisens, polarization resistors are not required.

The APLISENS transmitters described in this document are NOT integrated with polarizations resistors.

2.3.6. Common mode voltage on RS485 bus

The permissible common mode voltage levels on the transmission line compared to GND are -7/+12V. Such levels are necessary for correct operation of the devices. Always connect the master's/converter's GND terminal with the transmitter's GND terminals.

Transmission line definitions

Signal	**Designation by APLISENS and transceiver manufacturers**	**Designation acc. to EIA**
Inverted (-)	B	A
Simple (+)	A	B

3. MODBUS Data Link Layer

This chapter describes data transfer on the bus. Data and their control structures are divided into groups and make up a message. A message means the smallest communication unit and only such units can be transferred between devices. Half-duplex mode means that at a given point in time only one device can be in transmitting mode while the other devices must be in receiving mode. A PC or a controller is a master device and the connected measuring or execution devices are slave devices.

Messages are transferred at all times under the control of the master device. All messages contain a slave address.

As a result two options of data transfer are available:

- Broadcast mode

In this communication mode the master device can send a broadcast message to each slave device, whatever its network address. In the APLISENS transmitters described in this document, the broadcast mode is intended for reading the defined network address using function 3 (function code=3). Given that each connected transmitter responds to a message containing a broadcast address, function 3 in this mode should only be used for a single slave device connected to the master device.

- Data transfer mode

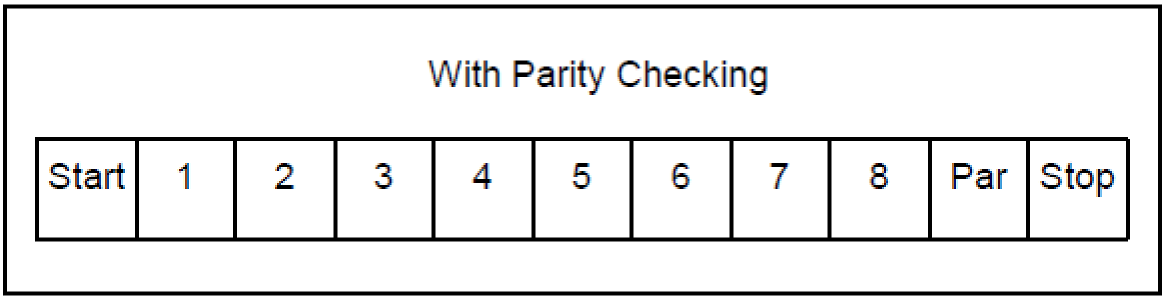
This mode enables communication between the master device and a selected slave device. The communication process involves sending a request message from the master device and a response message from an appropriate slave device. Only the master device can send request messages. The request is received by all slave devices connected to the bus, however, only a device with an address specified in the message sends a response. A slave device must respond to a correctly received request message within the defined maximum time interval, otherwise the master device recognizes that the request failed and will retry sending the message in accordance with the programmed algorithm.

3.1. Modbus RTU serial transmission mode in transmitters PCE-28/Modbus, PRE-28/Modbus and level probes SGE-25. Smart/Modbus, SGE-25S.Smart/Modbus

Data in the RS485 bus are transmitted serially. As described in the Modbus RTU standard (MODBUS over Serial Line Specification and Implementation Guide v1.02), the following formats can be used:

- **1 start bit**
- **8 data bits** (binary coding, least significant is bit sent first)
- **bits for parity checking:**
  - 1 bit **Even** (default), or
  - 1 bit **Odd**, or
  - 0 bits **None** (no bits for parity control)
- 1 **stop bit** for Even or Odd parity mode, or
- 2 **stop bits** for None parity mode

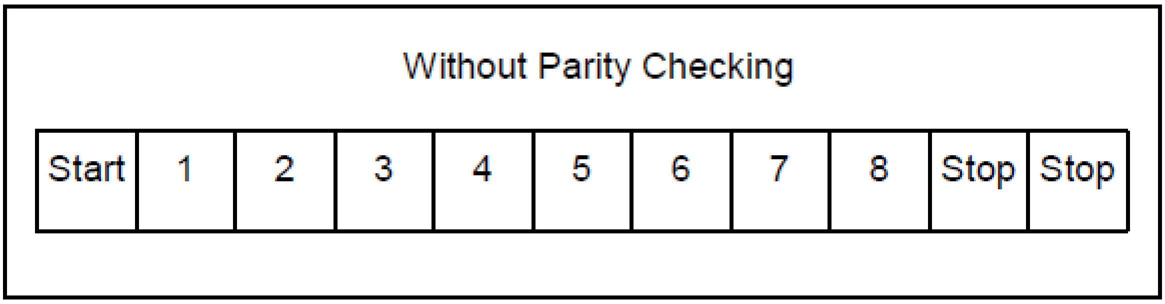
Characters are always sent using 11 bits. The number of stop bits (1 or 2) is selected by the transmitter depending on whether an Even, Odd or None parity mode is set.



With Parity Checking

Start	1	2	3	4	5	6	7	8	Par	Stop

Bit sequence with parity checking.



Without Parity Checking

Start	1	2	3	4	5	6	7	8	Stop	Stop

Bit sequence without parity checking.

3.2. Modbus RTU message format

All Modbus RTU messages are sent in the following format:

Slave Address	Function Code	Data	CRC
1 byte	1 byte	0 up to 252 byte(s)	2 bytes CRC Low | CRC Hi

Modbus RTU message field descriptions:

Slave Address – network address of a slave device. The address 0 is reserved for broadcast mode when a request is sent from the master device. All slave devices, whatever their addresses, must be capable of responding to a message sent to the address 0 by the master.
The individual slave devices are assigned addresses in the range of 1 – 247. In one network cannot be more than one device with the same address in this range. Addresses in the range of 248 – 255 are reserved for future use.

0	From 1 to 247	From 248 to 255
Broadcast address	Slave individual addresses	Reserved

Function code – A code to tell the slave to which a request is addressed what functions are to be performed. Function codes use 7 bits (0...127). The oldest 8th bit is always zero when a message is sent by the master. The same function code is returned in this field in the slave response message. If the oldest 8th bit is zero, it means that the function was performed correctly and the data returned in the message are also correct. If the oldest 8th bit is one, it means that a function error or a device error was detected.

Data – The Function Code can be followed by the field Data containing control data (master request) or read data (slave response), depending on the function number. This field can contain up to 252 bytes of data. If the oldest 8th bit of the function code in the slave response is one, the data read are not entered in the field Data in the response message, but one byte of the field is used for sending an error code, see **DTR.PCE.PRE.SGE.MODBUS*(ENG)*, Part 1, section 2.6. Error codes supported by Modbus.**

CRC – At the end of a message, there is always a field for the 2-byte control sum CRC16 sent according to the sequence CRC Low | CRC High. The control sum is calculated as defined in Supplement B to 'MODBUS over Serial Line Specification and Implementation Guide v1.02' available at http://www.modbus.org/.

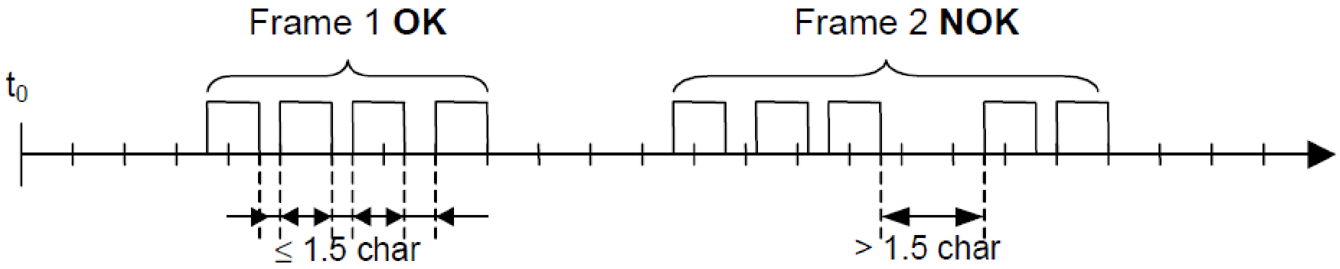
3.3. Modbus RTU message transfer rules

3.3.1. Primary rules:

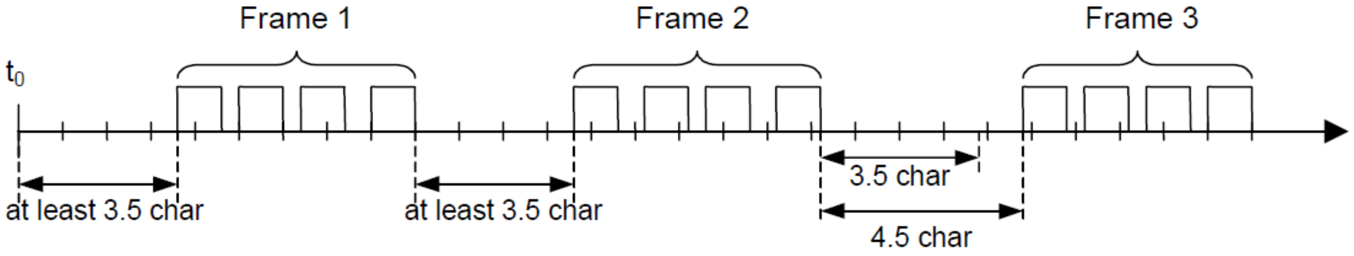
– An address in the range (1...247) can only be assigned to one slave device connected to the bus. If more devices connected to the bus are assigned the same address, all of them will send conflicting response messages at the same time.

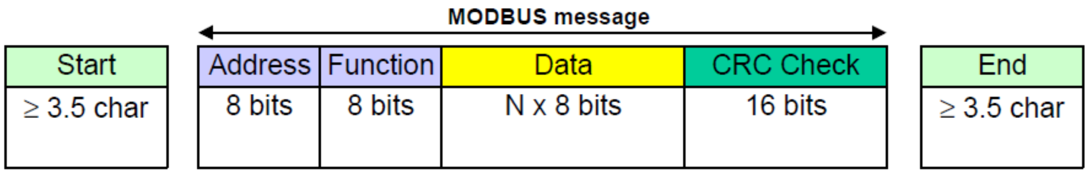
– Data transfer operations via messages are initiated at all times by the master. This means that slave devices can only send data-containing messages after receiving a properly addressed request message from the master.

– A message is made up of a number of bytes. These bytes should be sent without any in-between intervals which mean that the entire message must be transmitted as a continuous stream. The maximum permissible silent interval between two successive characters must not exceed 1.5T, where T is the time it takes to transmit one character (11 bits). If the silent interval between two successive characters exceeds 1.5T, the message can be considered invalid and rejected by the device.



– The addressed slave must respond within a defined silent interval, otherwise the response message will be considered invalid and rejected by the master. The silent interval between the request message sent by the master and the response message sent by the slave must not be shorter than 3.5T, where T is the time it takes to transmit one character (11 bits).
The maximum silent interval after which a slave device responds to the master request message depends on the function code in the request and the data load. The response time in the APLISENS transmitters described in this document should be less than 5 ms, including the most disadvantageous conditions (data load, transfer rate).





Start	Address	Function	Data	CRC Check	End
≥ 3.5 char	8 bits	8 bits	N x 8 bits	16 bits	≥ 3.5 char

3.3.2. Error handling

When messages are transferred between master and slave devices, two main types of errors may arise: transmission errors and slave device errors.

Transmission errors, causes:

- The message received is too short due to e.g. an excessively long interval between bytes contained in the message.
- The message received is longer than allowed by the device's reception buffer due to e.g. message frame syntax improperly programmed in the controller.
- The maximum character transmission time is exceeded due to an inappropriate transfer rate.
- The control sum calculated based on the message received does not correspond to the value sent in the message in the CRC field.

Slave devices do not respond to messages for which a transmission error is detected, the master device can retry sending a request message if an invalid message receipt is detected. The respective algorithm is programmed in the master device.

- Illegal function code.
- Illegal data address.
- Illegal data load.

Slave device errors, causes:

- Transmitter failure.
- RAM, FLASH memory errors.

Error codes conform to 'MODBUS over Serial Line Specification and Implementation Guide v1.02'.
For error codes and their descriptions, see **DTR.PCE.PRE.SGE.MODBUS*(ENG)*, Part 1, section 2.6**. **Error codes supported by Modbus**

4. Application layer, function 03

The transmitters PCE-28/Modbus, PRE-28/Modbus and the level probes SGE-25.Smart/Modbus, SGE-25S.Smart/Modbus are provided with the function **03 Read Holding Register**. This function code is used to read the contents of a contiguous address space in a data block. The master device specifies a starting address of the register (2 bytes) and the number of 2-byte read registers. The registers are addressed starting from zero therefore e.g. the first register's address is 0x0000, and the address of the register 17 is 0x0010, i.e. 16.
The data read from each register are sent by slave devices as 2 bytes per register, where the first byte is older than the next one. If data are read from multiple registers, the data from slave devices are sent according to the register numbers.
Registers contain data related to measurements of process values, temperatures and other transmitter settings. A list of registers, their addresses and descriptions are shown in the table below.

4.1. Data layout (registers) in address space

Read-only data.

Register	Address (hex)	Purpose	Notes	Format	Bytes (2 bytes per register)
1	0x0000	**User specific**	% of the controlled set range	IEEE754	4 bytes (2 registers)
3	0x0002	**Pressure of sensor 1**	Pressure or level process variable	IEEE754	4 bytes (2 registers)
5	0x0004	**Pressure of sensor 2**	Constant 0	IEEE754	4 bytes (2 registers)
7	0x0006	**Temperature of sensor 1**	Sensor 1 temperature process variable at ºC	IEEE754	4 bytes (2 registers)
9	0x0008	**CPU temperature**	CPU temperature process variable at ºC	IEEE754	4 bytes (2 registers)
11	0x000A	**Temperature of sensor 2**	Constant 0°C	IEEE754	4 bytes (2 registers)
13	0x000C	------------------------	------------------------	-----------------------	4 bytes (2 registers)
15	0x000E	------------------------	------------------------	-----------------------	4 bytes (2 registers)
17	0x0010	**User specific**	1/100% of the controlled set range	Signed 16-bit int	2 bytes (1 register)
18	0x0011	**Pressure of sensor 1**	Integer, 1/100 of the pressure or level unit	Signed 16-bit int	2 bytes (1 register)
19	0x0012	**Pressure of sensor 2**	Value 0 1/100 of the selected unit	Signed 16-bit int	2 bytes (1 register)
20	0x0013	**Temperature of sensor 1**	Integer, 1/100ºC	Signed 16-bit int	2 bytes (1 register)
21	0x0014	**CPU temperature**	Integer, 1/100ºC	Signed 16-bit int	2 bytes (1 register)
22	0x0015	**Temperature of sensor 2**	Value 0 1/100 at ºC	Signed 16-bit int	2 bytes (1 register)
23	0x0016	**Pressure unit**	Pressure or level unit	Unsigned 16-bit int	2 bytes (1 register) See DTR.PCE.PRE.SGE. MODBUS*(ENG)*, Part 1, table in section 2.5.1
24	0x0017	------------------------	------------------------	-----------------------	2 bytes (1 register)
25	0x0018	**Upper sensor limit**	Upper limit of the standard range	IEEE754	4 bytes (2 registers)
27	0x001A	**Lower sensor limit**	Lower limit of the standard range	IEEE754	4 bytes (2 registers)
29	0x001C	**Damping value** seconds (s)		IEEE754	4 bytes (2 registers)
31	0x001E	**Response delay value** milliseconds (ms)		Unsigned 16-bit int	2 bytes (1 register)
32	0x001F	**Modbus address**	1…247	Unsigned 8-bit int	2 bytes (1 register)
33	0x0020	**Identity register**		Unsigned 8-bit int	6 bytes (3 registers)
36	0x0023	**Status register**		8-bit flags	2 bytes (1 register)

* Fields shaded in gray are not active in the specified models.

4.1.1. Request message (example 1)

Master request and slave response, sensor 1 pressure read value

Request message							
Slave address	Function	Parameters				CRC(L)	CRC(H)
		Starting address in register (H)	Starting address in register (L)	Number of registers to be read (H)	Number of registers to be read (L)		
0x01	**0x03**	**0x00**	**0x02**	**0x00**	**0x02**	**0x65**	**0xCB**
Response message							
Slave address	Function	Parameters				CRC(L)	CRC(H)
		Number of data bytes	Data value (sensor 1 pressure in IEEE754 format), hexadecimal numbers				
0x01	**0x03**	**0x04**	**40 5F D1 BC**			**0x82**	**0x00**

4.1.2. Request message (example 2)

Master request and slave response, read total accessible address space

Request message							
Slave address	Function	**Parameters**				CRC(L)	CRC(H)
		Starting address in register (H)	Starting address in register (L)	Number of registers to be read (H)	Number of registers to be read (L)		
0x01	**0x03**	**0x00**	**0x00**	**0x00**	**0x24**	**0x45**	**0xD1**
Response message							
Slave address	Function	**Parameters**				CRC(L)	CRC(H)
		Number of data bytes	Memory content in registers read				
0x01	**0x03**	**0x48**	**00 00 00 00 40 5F F8 DD 00 00 00 00 41 C8 00 00 41 C8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 5E 00 00 09 C4 09 C4 00 00 00 0C 00 00 42 C8 00 01 00 00 00 00 00 00 00 00 00 00 00 01 00 BC 7D 00 00 01 00 00**			**0x97**	**0xCE**

4.1.3. Register list for Example 2

1	0x0000	**User specific**	IEEE754	**40 5F F8 DD**	4 bytes (2 registers)
3	0x0002	**Pressure of sensor 1**	IEEE754	**40 5F F8 DD**	4 bytes (2 registers)
5	0x0004	**Pressure of sensor 2**	IEEE754	**00 00 00 00**	4 bytes (2 registers)
7	0x0006	**Temperature of sensor 1**	IEEE754	**41 C8 00 00**	4 bytes (2 registers)
9	0x0008	**CPU temperature**	IEEE754	**41 C8 00 00**	4 bytes (2 registers)
11	0x000A	**Temperature of sensor 2**	IEEE754	**00 00 00 00**	4 bytes (2 registers)
13	0x000C	-------------------------	------------------------	**00 00 00 00**	4 bytes (2 registers)
15	0x000E	-------------------------	------------------------	**00 00 00 00**	4 bytes (2 registers)
17	0x0010	**User specific**	Signed 16-bit int	**01 5E**	2 bytes (1 register)
18	0x0011	**Pressure of sensor 1**	Signed 16-bit int	**01 5E**	2 bytes (1 register)
19	0x0012	**Pressure of sensor 2**	Signed 16-bit int	**00 00**	2 bytes (1 register)
20	0x0013	**Temperature of sensor 1**	Signed 16-bit int	**09 C4**	2 bytes (1 register)
21	0x0014	**CPU temperature**	Signed 16-bit int	**09 C4**	2 bytes (1 register)
22	0x0015	**Temperature of sensor 2**	Signed 16-bit int	**00 00**	2 bytes (1 register)
23	0x0016	**Pressure unit**	Unsigned 16-bit int	**00 0C**	2 bytes (1 register) See DTR.PCE.PRE.SGE. MODBUS*(ENG)*, Part 1, table in section 2.5.1
24	0x0017	-------------------------	------------------------	**00 00**	2 bytes (1 register)
25	0x0018	**Upper sensor limit**	IEEE754	**42 C8 00 01**	4 bytes (2 registers)
27	0x001A	**Lower sensor limit**	IEEE754	**00 00 00 00**	4 bytes (2 registers)
29	0x001C	**Damping value** seconds (s)	IEEE754	**00 00 00 00**	4 bytes (2 registers)
31	0x001E	**Response delay value** milliseconds (ms)	Unsigned 16-bit int	**00 06**	2 bytes (1 register)
32	0x001F	**Modbus address**	Unsigned 8-bit int	**00 01**	2 bytes (1 register)
33	0x0020	**Identity register**	Unsigned 8-bit int	**00 BC 7D 00 00 01**	6 bytes (3 registers)
36	0x0023	**Status register**	8-bit flags	**00 00**	2 bytes (1 register)

* Fields shaded in gray are not active in the specified models

4.2. Data from registers readable using function 03

4.2.1. Register 1, User specific, % of the controlled set range

1	0x0000	**User specific**	% of the controlled set range	IEEE754	4 bytes (2 registers)

It is a control value of the set range, expressed in percentage points. For example: if the measuring range is 0-100kPa and the currently read-out value of pressure is 50 kPa, then the control value equals 50%. This value may be applied by the user to perform additional mathematical operations on it in a resolver. This value may also be converted in a transmitter, using quadratic, cubic, root. or linear multi-section functions. It may be used, e.g. to measure the flow through orifices or to calculate volume in tanks of various shapes.

4.2.2. Register 3, Pressure of sensor 1

3	0x0002	**Pressure of sensor 1**	Pressure or level process variable	IEEE754	4 bytes (2 registers)

Basic process variable (pressure or level) standardized for the selected physical unit. This value is sent in 4-byte floating point format conforming to IEEE754. The maximum readable pressure or level range lies between:
[Lower sensor limit – 0.5 x (Upper sensor limit – Lower sensor limit)], and
[0.5 x (Upper sensor limit – Lower sensor limit) + Upper sensor limit].
In other words, if the lower sensor limit is designated Pd and the upper sensor limit is designated Pg, then the transmitter's maximum possible processing range can be described as:
P= [Pd -0,5x(Pg-Pd) ….. Pg+0,5x(Pg-Pd)]

Example:

A pressure transmitter for which Pd=0 kPa, Pg=100 kPa will process pressure in the range of -50 kPa to +150 kPa. If the pressure continues to increase beyond the permissible range, the read value will not change.

4.2.3. Register 5, Pressure of sensor 2

5	0x0004	**Pressure of sensor 2**	Constant 0	IEEE754	4 bytes (2 registers)

Standby register. Data variables to be read are always zero.

4.2.4. Register 7, Temperature of sensor 1

7	0x0006	**Temperature of sensor 1**	Sensor 1 temperature process variable at ºC	IEEE754	4 bytes (2 registers)

Value of the temperature process variable in ºC read from the transmitter's measuring head. This value reflects the temperature of the measuring sensor and an approximate temperature of the medium* (depending on the application).

4.2.5. Register 9, CPU temperature

9	0x0008	**CPU temperature**	CPU temperature process variable at ºC	IEEE754	4 bytes (2 registers)

Value of the temperature process variable in ºC read from the transmitter's CPU. This value reflects the temperature of the transmitter's electronic circuits.

4.2.6. Register 11, Temperature of sensor 2

11	0x000A	**Temperature of sensor 2**	Constant 0 ºC	IEEE754	4 bytes (2 registers)

Standby register. Data variables to be read are always zero.

4.2.7. Register 13, Standby

13	0x000C	------------------------	------------------------	-----------------------	4 bytes (2 registers)

Standby register. Data variables to be read are always zero.

4.2.8. Register 15, Standby

15	0x000E	------------------------	------------------------	-----------------------	4 bytes (2 registers)

Standby register. Data variables to be read are always zero.

4.2.9. Register 17, User value

17	0x0010	**User specific**	1/100% of the controlled set range	Signed 16-bit int	2 bytes (1 register)

A given value has been described in Register 1. Outside this index, it also occurs in a 16-bit integer version with a sign.

4.2.10. Register 18, Pressure of sensor 1– binary format, integer and sign

18	0x0011	**Pressure of sensor 1**	Integer, 1/100 of the pressure or level unit`	Signed 16-bit int	2 bytes (1 register)

Value of the primary process variable (pressure) standardized for the selected physical unit of pressure (or level). This value is an integer sent in scalable binary format as a value 100 times greater than the value of the sensor 1 pressure process variable.

Caution! The measuring range for a scalable integer, including its sign, is in the range of -32767 to 32767 units. If this range is exceeded, a false value will be read. This is valid for the following units: g/cm^2, mbar, mmH_2O in 4ºC, mmH_2O, mmHg, Pa, torr.

4.2.11. Register 19, Pressure of sensor 2

19	0x0012	**Pressure of sensor 2**	Value 0 1/100 of the selected unit.	Signed 16-bit int	2 bytes (1 register)

Standby register. Data variables to be read are always zero.

4.2.12. Register 20, Temperature of sensor 1– binary format, integer and sign

20	0x0013	**Temperature of sensor 1**	Integer, 1/100ºC	Signed 16-bit int	2 bytes (1 register)

Value of the temperature process variable in ºC read from the transmitter's measuring head. This value is an integer sent in scalable binary format as a value 100 times greater than the value of the sensor 1 temperature process variable. This value reflects the temperature of the measuring sensor and an approximate temperature of the medium* (depending on the application).

4.2.13. Register 21, CPU temperature – binary format, integer and sign

21	0x0014	**CPU temperature**	Integer, 1/100ºC	Signed 16-bit int	2 bytes (1 register)

Value of the temperature process variable in ºC read from the transmitter's CPU. This value is an integer sent in scalable binary format as a value 100 times greater than the value of the CPU temperature process variable. This value reflects the temperature of the transmitter's electronic circuits.

4.2.14. Register 23, Pressure unit

23	0x0016	**Pressure unit**	Unsigned 16-bit int	**00 0C**	2 bytes (1 register) see DTR.PCE.PRE.SGE.MODBUS *(ENG)* Part 1, tab. p.2.5.1

Binary value corresponding to the pressure unit. The table below shows the units used.

Unit	Value (decimal number)	Unit	Value (decimal number)
atm	14	mbar	8
bar	7	mmH2O in 4ºC	239
FtH2O	3	mmH2O	4
g/cm^2	9	mH2O in 4ºC	171
InH2O in 4ºC	238	mmHg	5
InHg	1	MPa	237
kg/cm^2	2	Pa	11
kPa	10	psi	6
mbar	12	torr	13

4.2.15. Register 24, Standby

24	0x0017	-------------------------	------------------------	-----------------------	2 bytes (1 register)

Standby register. Data variables to be read are always zero.

4.2.16. Register 25, Upper sensor limit

25	0x0018	**Upper sensor limit**	Upper limit of the basic range	IEEE754	4 bytes (2 registers)

An upper value of the transmitter's basic range expressed in the transmitter's pressure units. This value is sent in 4-byte floating point format conforming to IEEE754.

4.2.17. Register 27, Lower sensor limit

27	0x001A	**Lower sensor limit**	Lower limit of the basic range	IEEE754	4 bytes (2 registers)

A lower value of the transmitter's basic range expressed in the transmitter's pressure units. This value is sent in 4-byte floating point format conforming to IEEE754.

4.2.18. Register 29, Damping value

29	0x001C	**Damping value** The seconds [s]		IEEE754	4 bytes (2 registers)

A damping block time constant in the transmitter in seconds. The damping time constant can be defined in the range of 0 to 30 seconds. This value is sent in 4-byte floating point format conforming to IEEE754.

4.2.19. Register 31, Response delay

31	0x001E	**Response delay value** milliseconds (ms)		Unsigned 16-bit int	2 bytes (1 register)

It is a value of response delay subsequent to the "Master" request. It describes the time between the end of a bit stop byte of the message checksum and a bit start byte of the response message address. This time depends on the speed of transmission and always exceeds 3.5T. This time may be useful in designing the time of a measurement cycle in a Modbus network.

4.2.20. Register 32, Modbus address

32	0x001F	**Modbus address**	1 … 247	Unsigned 8-bit int	2 bytes (1 register)

Values of the transmitter's network address register. The address is sent using 2 bytes. The first older byte is always zero. The second lower byte is used for the address. The address can be set in the range of 1 to 247. Each transmitter connected to the network must have a unique network address. If the same address is assigned to more than one transmitter, the master will not be able to communicate with all present transmitters on the bus.

4.2.21. Register 33, Identity register

33	0x0020	**Identity register**		Unsigned 8-bit int	6 bytes (3 registers)

An identifier of the manufacturer, type and serial number of the transmitter. A unique identifier, different for each transmitter.

Identification of bytes according to the transmission sequence:

1st byte – insignificant, always 0

2nd byte – manufacturer's number acc. to HCF, APLISENS number: 188 dec (BC hex)

3rd byte – device type number, number for the transmitters described in this document: 125 dec (7D hex)

4th-6th byte – 24-bit binary identification number of the device. This number identifies the date of manufacture and serial number.

4.2.22. Register 36, Status register

36	0x0023	**Status register**		8-bit flags	2 bytes (1 register)

A 2-byte register monitoring the functioning of the transmitter's subsystems. Events and the transmitter's irregularities modify bits contained in the register, see: **DTR.PCE.PRE.SGE. MODBUS*(ENG)*, Part 1, p. 2.5. Modbus status register**

These irregularities can be identified in detail in configuration mode with the use of programmes such as e.g. Raport 2 from APLISENS.

5. Configuration mode

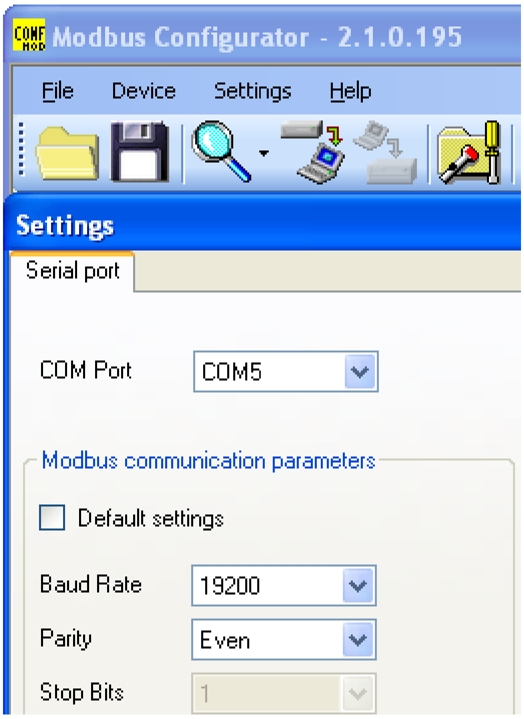
5.1. Modbus Configurator - serial port configuration, Modbus network search, searching performed by an individual transmitter.

To identify, read and modify settings or to perform calibration of the transmitters described in this document, switch the transmitter into Configuration Mode using **Modbus Configurator**. To do so, use the button on the form of the program.

- **The procedure to be followed, if it is required to reconfigure parameters of one of the transmitters operating in a Modbus network:**

It is not necessary to disconnect the Aplisens transmitters from the Modbus network to change their settings or to perform calibration. If all transmitters connected to network are in Modbus mode, stop the master process controller and connect an additional PC controller provided with Modbus Configurator to the network.

After launching the Configurator, open the tab **Serial port settings** and set the number of COM port to which the RS485 converter is connected, Modbus baud rate, parity mode and stop bits for the transmitters in the network.


Modbus Configurator - 2.1.0.195
File
Device
Settings
Help
Settings
Serial port
Modbus communication parameters

COM Port: COM5

☐ Default settings

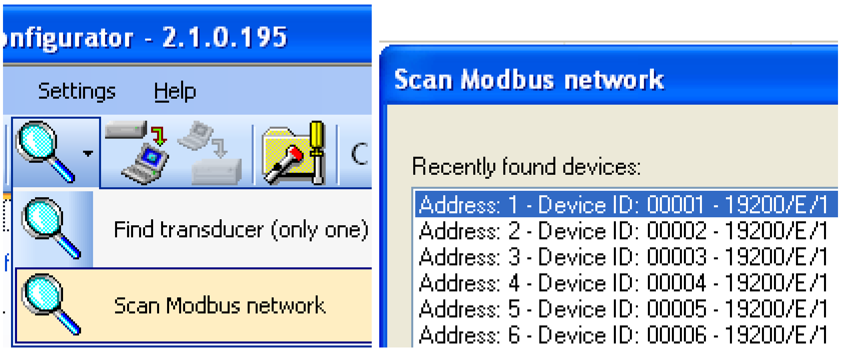
Baud Rate: 19200

Parity: Even

Stop Bits: 1

Serial port settings (example).

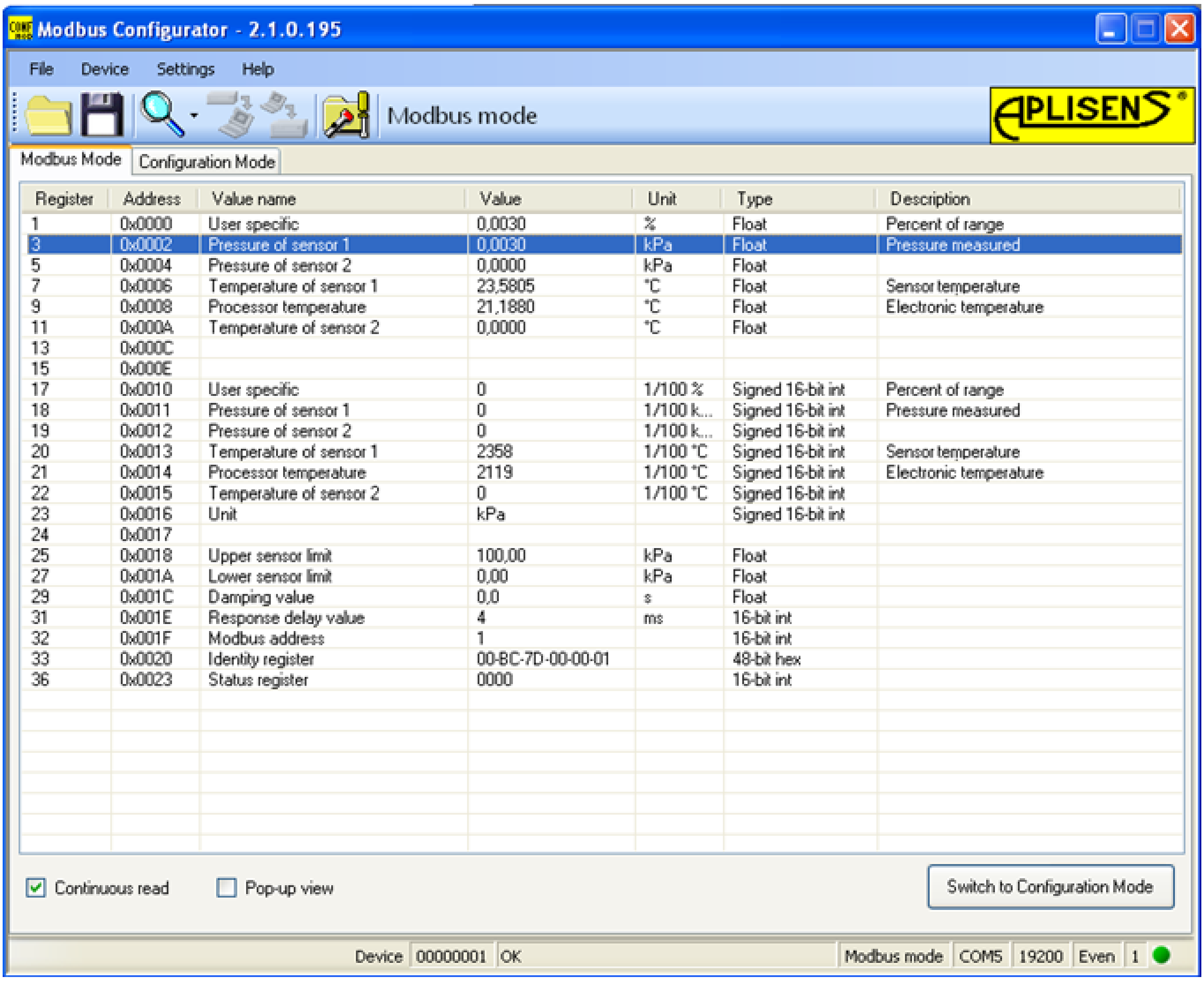
Next, search for Modbus transmitters according to address by clicking on **Scan Modbus network** (see figure below).



Recently found devices:

Address: 1 - Device ID: 00001 - 19200/E/1
Address: 2 - Device ID: 00002 - 19200/E/1
Address: 3 - Device ID: 00003 - 19200/E/1
Address: 4 - Device ID: 00004 - 19200/E/1
Address: 5 - Device ID: 00005 - 19200/E/1
Address: 6 - Device ID: 00006 - 19200/E/1

When searching the network, a list of Modbus transmitters from Aplisens, connected to the network, will be displayed. If you know the network address of the transmitter to be set up, click on the appropriate transmitter in the scanned list... A list of registers will appear with read-out parameters from Modbus registers…



Register	Address	Value name	Value	Unit	Type	Description
1	0x0000	User specific	0,0030	%	Float	Percent of range
3	0x0002	Pressure of sensor 1	0,0030	kPa	Float	Pressure measured
5	0x0004	Pressure of sensor 2	0,0000	kPa	Float	
7	0x0006	Temperature of sensor 1	23,5805	°C	Float	Sensor temperature
9	0x0008	Processor temperature	21,1880	°C	Float	Electronic temperature
11	0x000A	Temperature of sensor 2	0,0000	°C	Float	
13	0x000C					
15	0x000E					
17	0x0010	User specific	0	1/100 %	Signed 16-bit int	Percent of range
18	0x0011	Pressure of sensor 1	0	1/100 k...	Signed 16-bit int	Pressure measured
19	0x0012	Pressure of sensor 2	0	1/100 k...	Signed 16-bit int	
20	0x0013	Temperature of sensor 1	2358	1/100 °C	Signed 16-bit int	Sensor temperature
21	0x0014	Processor temperature	2119	1/100 °C	Signed 16-bit int	Electronic temperature
22	0x0015	Temperature of sensor 2	0	1/100 °C	Signed 16-bit int	
23	0x0016	Unit	kPa		Signed 16-bit int	
24	0x0017					
25	0x0018	Upper sensor limit	100,00	kPa	Float	
27	0x001A	Lower sensor limit	0,00	kPa	Float	
29	0x001C	Damping value	0,0	s	Float	
31	0x001E	Response delay value	4	ms	16-bit int	
32	0x001F	Modbus address	1		16-bit int	
33	0x0020	Identity register	00-BC-7D-00-00-01		48-bit hex	
36	0x0023	Status register	0000		16-bit int	

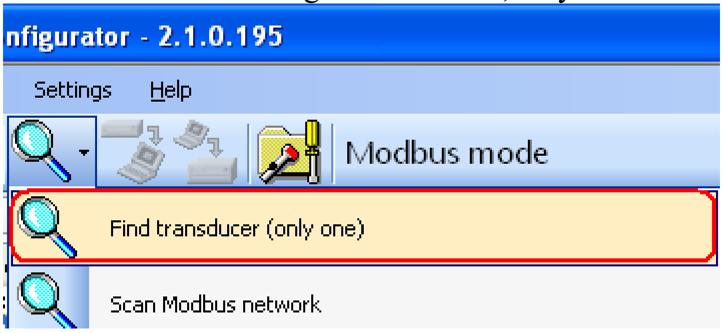
☑ Continuous read ☐ Pop-up view

Device 00000001 OK Modbus mode COM5 19200 Even 1

Switch the transmitter from Modbus into configuration mode…

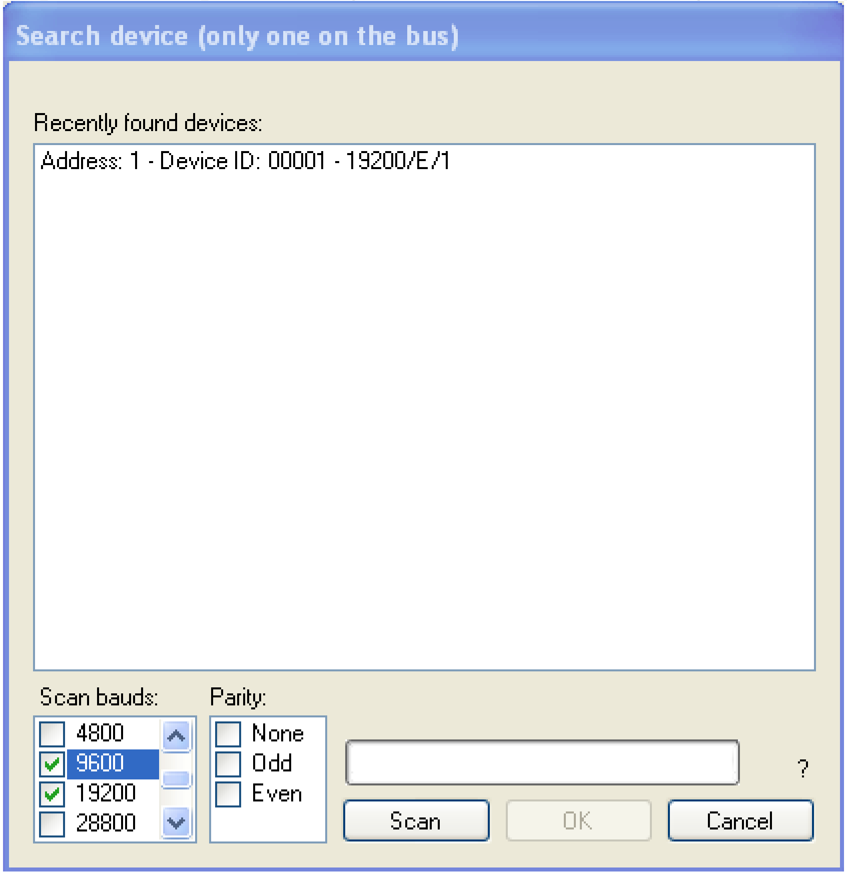
– **The procedure to be followed, if it is required to reconfigure parameters or identify an individual transmitter:**

If the user wishes to identify an individual transducer and they are not sure whether it is operating in the Modbus or Configuration modes, they shall use the **"Find transducer (only one)"** option.



Transmitter search parameters related to potential transmission settings must be configured in the opened window. When in doubt, as to the range of settings, the best solution is to tick all the options. Clicking the "Scan" button prompts the programme to start searching for a transmitter, in a sequence, starting from the Configuration mode, and then it will go through an indicated range of transmission speeds and parity in the Modbus mode.

Searching for a transmitter in the Modbus mode is carried out with a broadcast = 0. Therefore, only one transmitter may be connected, when scanning through a communication bus. Where there are a larger number of transmitters operating in the Modbus mode connected to the bus and set to the same transmission parameters - sending a broadcast address will make transmitters respond to a request message at the same time, causing data corruption and an unreadable answer.



Search device (only one on the bus)

Recently found devices:

Address: 1 - Device ID: 00001 - 19200/E/1

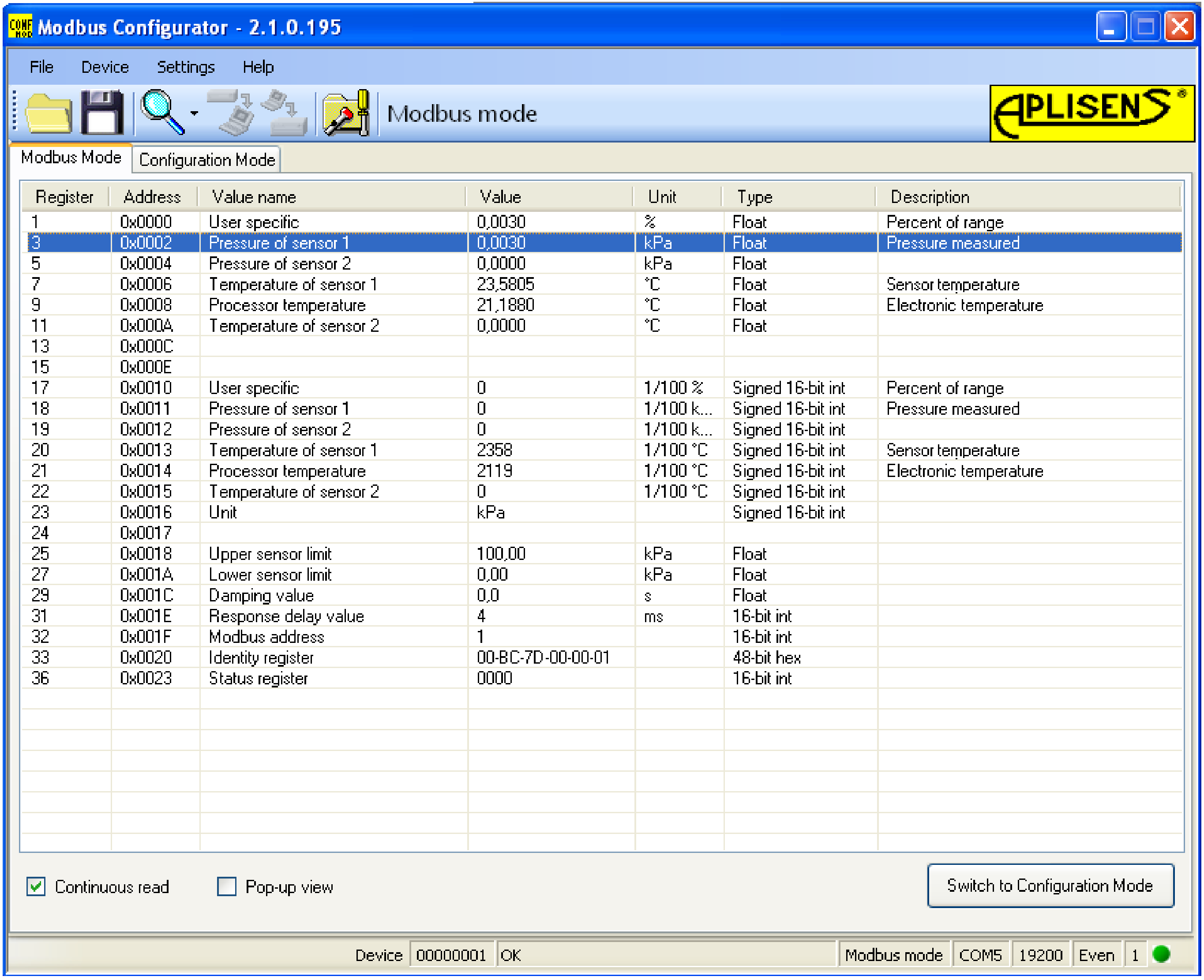
Scan bauds:

□ 4800
☑ 9600
☑ 19200
□ 28800

Parity:

□ None
□ Odd
□ Even

Click on the detected transmitter... A list of registers will appear with parameters read out from Modbus registers...



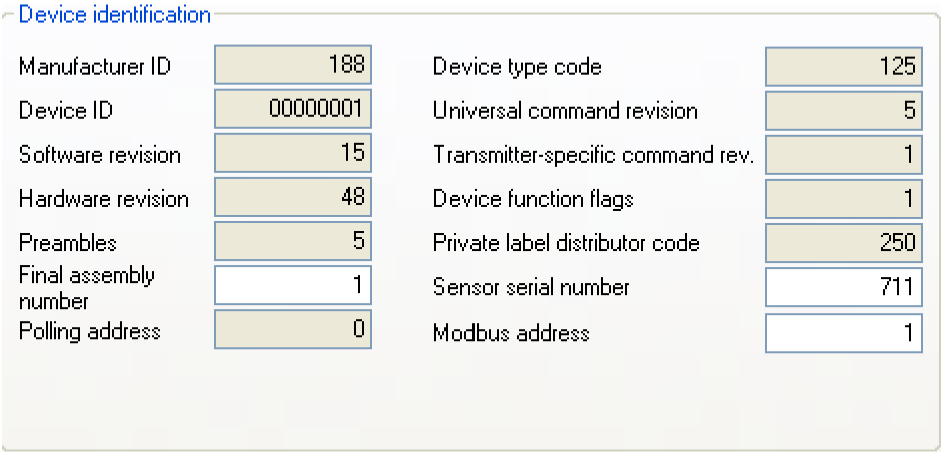
Register	Address	Value name	Value	Unit	Type	Description
1	0x0000	User specific	0,0030	%	Float	Percent of range
3	0x0002	Pressure of sensor 1	0,0030	kPa	Float	Pressure measured
5	0x0004	Pressure of sensor 2	0,0000	kPa	Float	
7	0x0006	Temperature of sensor 1	23,5805	°C	Float	Sensor temperature
9	0x0008	Processor temperature	21,1880	°C	Float	Electronic temperature
11	0x000A	Temperature of sensor 2	0,0000	°C	Float	
13	0x000C					
15	0x000E					
17	0x0010	User specific	0	1/100 %	Signed 16-bit int	Percent of range
18	0x0011	Pressure of sensor 1	0	1/100 k...	Signed 16-bit int	Pressure measured
19	0x0012	Pressure of sensor 2	0	1/100 k...	Signed 16-bit int	
20	0x0013	Temperature of sensor 1	2358	1/100 °C	Signed 16-bit int	Sensor temperature
21	0x0014	Processor temperature	2119	1/100 °C	Signed 16-bit int	Electronic temperature
22	0x0015	Temperature of sensor 2	0	1/100 °C	Signed 16-bit int	
23	0x0016	Unit	kPa		Signed 16-bit int	
24	0x0017					
25	0x0018	Upper sensor limit	100,00	kPa	Float	
27	0x001A	Lower sensor limit	0,00	kPa	Float	
29	0x001C	Damping value	0,0	s	Float	
31	0x001E	Response delay value	4	ms	16-bit int	
32	0x001F	Modbus address	1		16-bit int	
33	0x0020	Identity register	00-BC-7D-00-00-01		48-bit hex	
36	0x0023	Status register	0000		16-bit int	

☑ Continuous read □ Pop-up view

Device 00000001 OK Modbus mode COM5 19200 Even 1 ●

Then, switch the transmitter from the Modbus mode to the Configuration mode...

5.2. Transmitter identification data



Device identification

Manufacturer ID	188	Device type code	125
Device ID	00000001	Universal command revision	5
Software revision	15	Transmitter-specific command rev.	1
Hardware revision	48	Device function flags	1
Preambles	5	Private label distributor code	250
Final assembly number	1	Sensor serial number	711
Polling address	0	Modbus address	1

Identification data frames (example). Read-only values are shaded.

Read-only information fields:

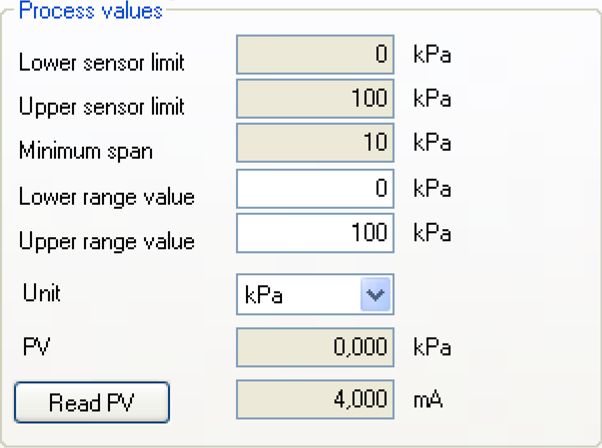
- **Manufacturer ID** - manufacturer’s code according to HCF (HART) specifications.
- **Device ID** - transmitter’s serial number.
- **Software revision** - CPU software version.
- **Hardware revision** - transmitter’s PCB version.
- **Preambles** - number of preambles used for communication in configuration mode.
- **Pooling address -** address used for communication in configuration mode.
- **Device type code** – code indicating the transmitter type.
- **Universal command revision** number in configuration mode.
- **Transmitter-specific command revision** – number in configuration mode.
- **Device function flags** – related to record handling type.
- **Private label distributor code.**

Editable information fields:

- **Final assembly number** of the device, recorded in binary format using 3 bytes, an integer in the range of 0...16777215 used for identification (modifiable by user).
- **Sensor serial number**, recorded in binary format using 3 bytes, an integer in the range of 0...16777215 used for identification (modifiable by user).
- **Modbus address**, an integer in the range of 1...247 (modifiable by user).

The field **Modbus address** used for defining the transmitter’s network address is highly significant.

5.3. Reading the transmitter’s limit values and process variables



Process values

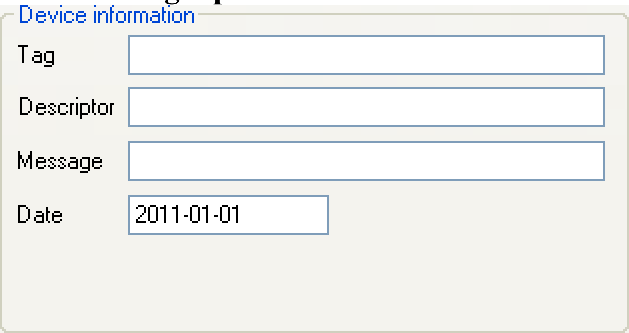
Lower sensor limit	0	kPa
Upper sensor limit	100	kPa
Minimum span	10	kPa
Lower range value	0	kPa
Upper range value	100	kPa
Unit	kPa	
PV	0,000	kPa
Read PV	4,000	mA

Frame containing process variables and limit values (example)

- **Lower sensor limit.** A read-only value. If exceeded, accuracy of measurements using the transmitter cannot be guaranteed. However, the transmitter will continue to process measuring signals up to 50% of the basic range below this point.
- **Upper sensor limit.** A read-only value. If exceeded, accuracy of measurements using the transmitter cannot be guaranteed. However, the transmitter will continue to process measuring signals up to 50% of the basic range above this point.
- **Minimum span.** A read-only value that defines the minimum width of the set range, modifiable by user by changing the start and/or the end points of the range. This parameter is also important when calibrating pressure or level. If intervals between calibration points are insufficient, the transmitter cannot be calibrated.
- **Lower range value.** Value related to the current loop mode: 4-20 mA. Specific pressure or level is assigned to the current of 4 mA.
- **Upper range value.** Value related to the current loop mode: 4-20 mA. Specific pressure or level is assigned to the current of 20 mA.
- Standard **unit** is the physical unit used for standardizing measurements performed with the transmitter (definable by user).
- **PV** is the value of the pressure or level process variable, standardized basic unit as defined. Additionally, the output current process variable (4-20 mA) is displayed.

To refresh the value, click on the button **Read PV**.

5.4. Reading alphanumeric identification data



Device information

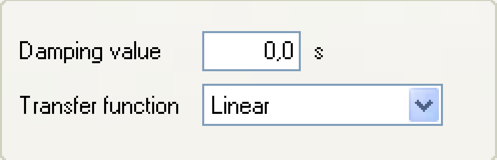
Tag	
Descriptor	
Message	
Date	2011-01-01

Alphanumeric identification data frame (example).

User can read or modify the transmitter’s alphanumeric identification data.

- Tag is an 8-character alphanumeric field where you can enter e.g. an identification number of the transmitter in the system. This field can contain digits and capital letters without national characters.
- Descriptor is a 16-character alphanumeric field where you can enter a short description of e.g. an installation site. This field can contain digits and capital letters without national characters.
- Message is a 32-character alphanumeric field where you can enter a longer description. This field can contain digits and capital letters without national characters.

5.5. Damping and transfer function



Damping value 0,0 s

Transfer function Linear

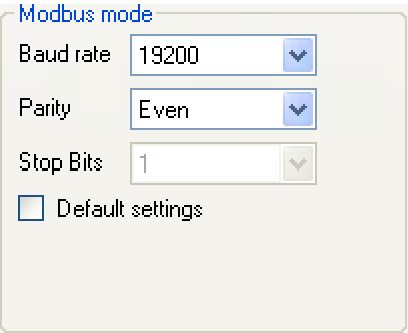
User can read or modify the measurement damping value and transfer function type.

- Damping value is expressed in seconds and describes the duration of a non-determined status after which the process variable will achieve 63.2% of the total value change. The process variable value will be nearly fully changed with an accuracy of below 1% after ca 5 time constants.
- The transfer function determines the method of transforming a process variable related to pressure measurement / pressure differences into a control value of a process output (current or a percentage value of the set range).

  The following functions are available from a drop-down list:
  - linear;
  - root;
  - quadratic;
  - user.

  The control value of a process output transformed with a transfer function is available in Modbus 1 and 17 registers.

5.6. Modbus mode



Modbus mode

Baud rate 19200

Parity Even

Stop Bits 1

□ Default settings

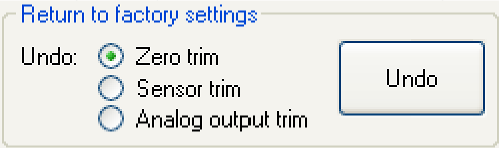
User can read or edit and save Modbus RTU baud rate and parity control mode or restore default settings.

5.7. Restore default settings

User can restore default settings for the following parameters:

- Zero-point calibration.
- Pressure sensor calibration.
- Analogue output calibration (current loop (4-20 mA) mode, configuration mode).

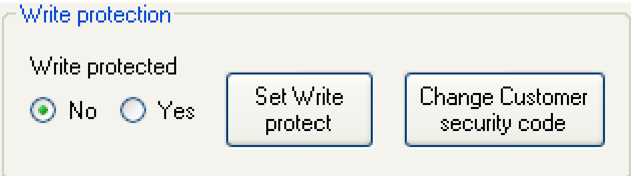
You can restore default settings if you changed some settings accidentally and you are not able to perform calibration to your own settings. Please remember that this operation will restore calibration settings defined by the manufacturer. If you have already calibrated the transmitter to your own settings, all your calibration settings will be lost as a result of this operation.



Return to factory settings

Undo: ● Zero trim
○ Sensor trim
○ Analog output trim

5.8. Write lock



Write protected ● No ○ Yes

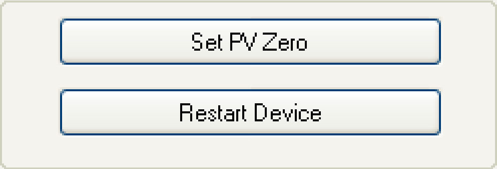
You can enable write protect to prevent modification of the transmitter's parameters or the code necessary to disable write lock. The write protect code consists of 8 hexadecimal characters, i.e. 0...9 plus A...F. Default setting: 00000000.

When write protect is enabled, the transmitter's settings cannot be changed.

5.9. Set pressure zero, restart transmitter

User can perform additional operations on the transmitter:

- **Set PV Zero** enables resetting pressure in a non-pressurized transmitter. The resetting option is used for eliminating any deviations of the zero point caused by installation (position or stress). The operation cannot be performed if deviation of the zero point exceeds the permissible limit or if the transmitter measures absolute pressure.
- **Restart Device** enables sending a command to hot restart the transmitter without the need to disconnect the power supply.



5.10. Calibrations

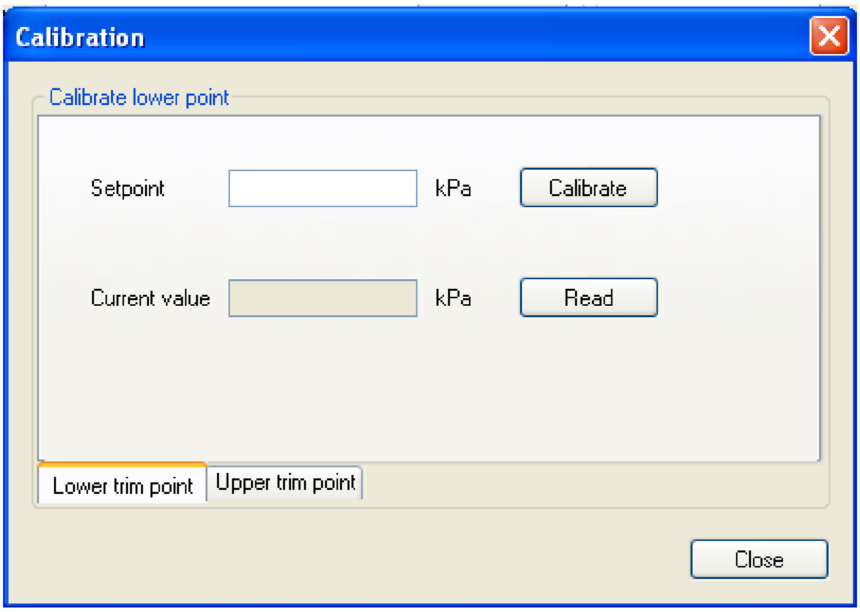
User can perform 2-point pressure calibration and 2-point current calibration.

5.10.1. Pressure or level calibration

To open the pressure or level calibration tab, select Device...Calibration on the top bar of the programme screen.

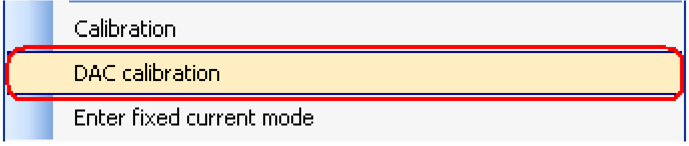
Calibration of the lower and upper pressure limits should be best performed for the transmitter's standard measuring range. For metrological reasons, if the used range is only an insignificant section of the standard range, the transmitter should be calibrated based on the lower and upper limits of the used range. The calibration procedure involves supplying reference pressure to the transmitter for the lower calibration point. The reference pressure or level should be entered into the set value field. As soon as the read value is stable, press Calibrate. If calibration is successful, an appropriate message will appear. Repeat the above steps for the upper pressure limit.

If the difference between the reference pressure and the actual pressure read by the transmitter exceeds the permissible limit, calibration will not be performed and an appropriate message will be displayed. A similar situation arises if the selected calibration point lies outside the transmitter's lower or upper limit.



5.10.2. Current calibration

If the transmitter is used in current loop mode (4-20 mA), it may be necessary to perform current calibration. To do so, select Device...DAC calibration (4-20 mA analogue output calibration) on the top bar of the screen.



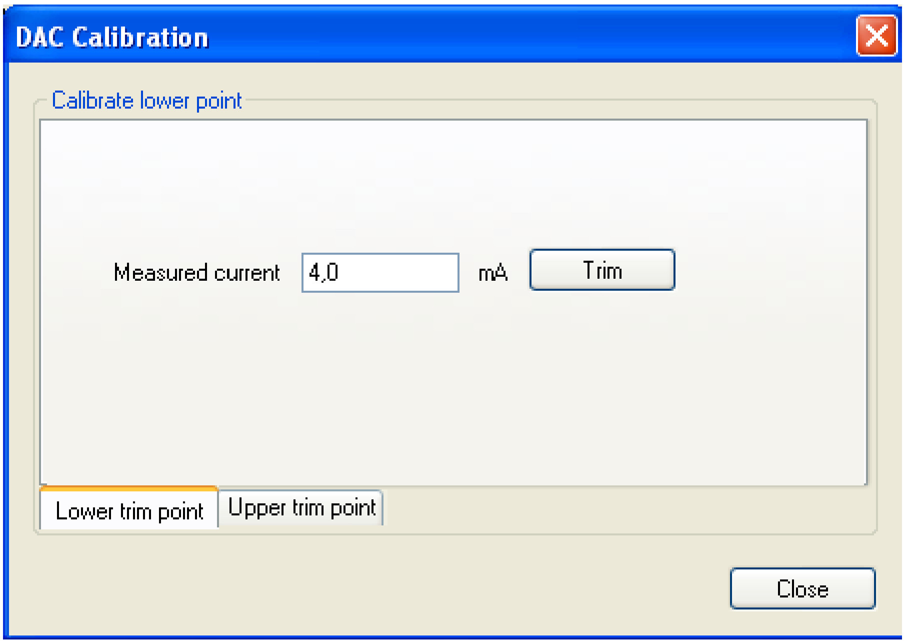
Before analogue output calibrations check that the RS485 converter is galvanically isolated. Using terminal or polarization resistors during analogue output calibration is not recommended. If the RS485 converter is a fail-safe converter, resistors are not required. The power supply to the transmitter should also be galvanically isolated.

The above procedures should be used to prevent uncontrolled current flow from the transmitter's controller branch which might cause a measurement error.

A reference milliammeter should be connected in parallel to the power supply to the transmitter.

Next, select an appropriate current calibration option in the calibration frame. Current calibration should be best performed for 2 points of the current range, 4 and 20 mA by default.

As soon as you enter the value from the milliammeter in the field Measured current for the lower calibration point (4 mA), press Trim to start calibration. Repeat the above steps for the upper calibration point (20 mA).



Measured current 4,0 mA

5.11. Current loop mode 4-20 mA

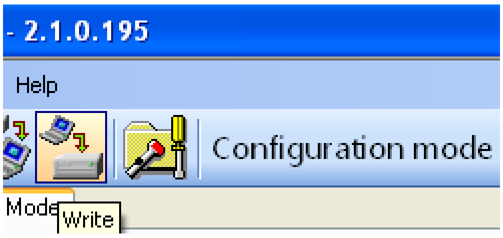
5.11.1. Set range

If you intend to use the transmitter in current loop mode (4-20 mA), you must check and set, if necessary, an appropriate measuring range.

The set range means that the lower pressure point is assigned to 4 mA (start of the set range) and the upper pressure point is assigned to 20 mA (end of the set range).

5.11.2. Configuration

If you wish to continue in configuration mode after setting parameters to use the transmitter in current loop mode (4-20 mA), you must save configuration settings. To do so, select Device...Write on the top bar of the screen or click on the save configuration icon

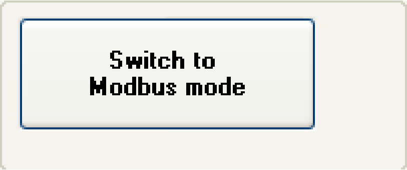


5.11.3. Recommendations

In the current loop mode (4-20 mA) the communication terminals RS485_A and RS485_B may only be used to configure the transmitter. During operation they should not be connected to any device (isolated and left loose). In an environment with very high EMI levels exceeding the exposure limits defined in EN 61000 and when using unshielded cables, it is not recommended connecting the communication line RS485_A and RS485_B into the cable.

5.12. Modbus mode

After configuring the transmitter as necessary to restore Modbus mode, you must save new settings. Click on the button Switch to Modbus mode to save new settings and switch the transmitter into Modbus communication mode.



6. PCB and software versions

01-03-2015 **- Implementation of version 15 of the transmitter firmware.**

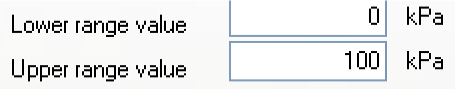
- The management of the data connection layer in the transmitter software has been improved. Under certain conditions, a transmitter may have failed to send a response to a correctly sent request message. The implemented patch has removed this fault.
- The option of reading a percentage value of the controlled set range from the transmitters in the Modbus mode has been added. A variable of the IEEE754 float type with a header is sent on 4 bytes:

Register	Address (hex)	Intended use	Notes	Format	Number of bytes (2 bytes per register)
1	0x0000	**User value**	% of the set range control	IEEE754	4 bytes (2 register)

A variable of the 16-bit integer type with a header is sent on 2 bytes:

17	0x0010	**User value**	1/100% of the set range control	Signed 16-bit int	2 bytes (1 register)

The set range is defined in the Configuration mode:



Lower range value 0 kPa

Upper range value 100 kPa

The set range may be equal to or lower than the basic range. It is defined mostly to configure a transmitter to operate in the 4-20 mA current loop mode.

In this mode, the start of the set range corresponds to the current of 4mA and the control percentage of 0%, whereas the end of the set range corresponds to the current of 20mA and the control percentage of 100%. Both the value of current in a current loop in the 4-20mA mode and the percentage of the set range control may be converted with the transfer function defined in the Configuration mode.

In the Modbus operating mode, the percentage of the set range control (0-100) may be used by the user to convert this value into any other value, by performing a mathematical operation in a resolver. At the same time, when a root function is set in the transmitter, it may be used to read out flows through orifices or reducers.

- The information entered by a transmitter to register 31 in version 15 of the firmware describes the actual time between the end of a bit stop of the last byte of the CRC checksum in a request message and a bit start of the byte of the transmitter answer message (response time). This time depends on the set speed of transmitter transmission within the range from 3ms to 20ms.
- Small corrections increasing the comfort of use have been introduced to the Modbus Configurator software dedicated to transmitters, at the same time incrementing it to version 2.1.0.195.
- Small corrections related to the afore-mentioned modifications have been introduced to the DTR.PC.PR.SG.MODBUS Operation Manual, Edition "C".

01-07-2014 **- Implementation of PCB version 48**

- Miniaturized printed circuit board with reduced dimensions.
- Frequency of the CPU clock frequency changed from 14.74 to 3.68 MHz, reduced power consumption.
- Additional D/A transmitter and current source circuits to enable control in current loop mode (4-20 mA).
- Change in the way signals are connected to the M12 electrical connection.
- Greater resistance to EMI interferences.

01-07-2014 **- Implementation of version 14 of the transmitter firmware.**

The new version 14 of the firmware was designed in parallel with the introduction of version 48 of the electronic system, and it introduced the following changes:

- Eliminated option of a 10-bit word in serial transmission. Starting from the software version 14, in accordance with the Modbus RTU standard (‘MODBUS over Serial Line Specification and Implementation Guide v1.02’) one character always consists of 11 bits. If there is no parity bit, 2 stop bits are used to achieve 11 bits in the character frame. The following formats are used: 8E1 (default), 8O1, 8N2.
- Eliminated option of modifying the transmitter’s response delay. In the version 14 this time interval is set to 0 and depends on such factors as the number of data read using function 03, defined transfer rate and coincidence of the request with the transmitter’s automated programme. This time interval should not be greater than 5 ms.
- Eliminated Modbus transfer rate of 600 b/s.
- Reduced consumption and optimized input energy management.
- Optimized time-critical procedures.
- Implemented functions related to operation of the transmitters in current loop mode.
- Implemented functions related to configuration of the transmitters in current loop mode.

24-06-2015 **- Correction of pressure units for the codes 1, 2, 10, 12 in table p. 2.4.1.**

01-09-2015 **- Complement the information regarding power supply for 4-20 mA operation mode in p.2.3**