QPT Sensor, Application Note



be
think
innovate

GRUNDFOS

1 VERSION HISTORY 3

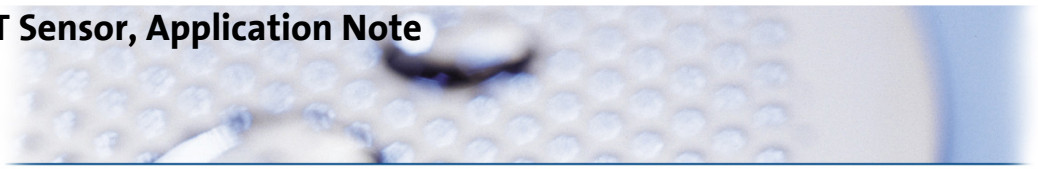
2 PURPOSE 4

3 ELECTRICAL INTERFACE 5

3.1 POWER SUPPLY 5
3.2 OUTPUT SIGNALS 5
3.3 ELECTRICAL CONNECTIONS 6
3.4 GROUNDING TECHNIQUE 7

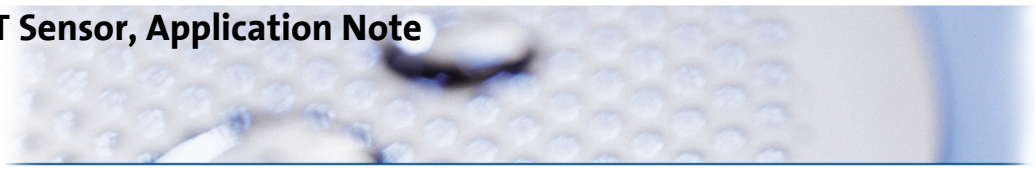
4 DATA COMMUNICATION 8

4.1 QPT DATA PROTOCOL 8
4.2 PROTOCOL ARCHITECTURE 8
Octet Format 8
Start 8
Stop 8
STX 8
Telegram length 8
Sensor ID 9
Tele type 9
RX (telegrams received by sensor) 9
TX (telegrams send by sensor) 9
Data 0 .. Data n 9
Checksum 9
4.3 TELEGRAMS FLOWS 10
Single Measurement 10
Continuous Measurements (start and stop) 10
Request Runtime Parameters 10
Set Runtime Parameters 11
For requesting sensor status 11
NACK 11
4.4 CONTENT 12
Read data 12
Measurement data 13
Set Runtime Parameter 14
Request Runtime parameter 15
Sensor Status 16
NACK 17



1 Version History

Version:	Date:	Changes:	Initials:
00.00.01	15-08-2013	Draft Document	LPED



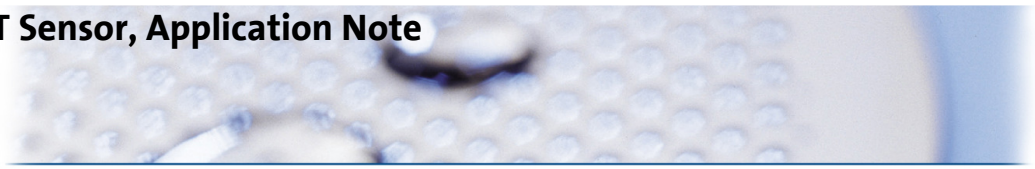
2 Purpose

This document describes the recommendations for implementing the digital flow, pressure and temperature sensor (QPT) in new products; as an example in heat pump systems, plastic injection moulding and solar thermal systems or when used with controller devices.

Focuses are at electrical interface and protocol description, whereas more general recommendations regarding the Grundfos Direct Sensors can be found in:
Design Guide Implementing Grundfos Sensors In New Products V02.00.00.pdf

In case of any additional technical questions not covered in this document, please do not hesitate to contact Grundfos Direct Sensors.

http://www.grundfos-directsensors.com



3 Electrical Interface

3.1 Power Supply

Table 1: Power Supply Recommendations

Nominal Supply Voltage	Min. Supply Voltage	Max. Supply Voltage	Max. Ripple
5.00 V_{DC}	4.75 V_{DC}	5.25 V_{DC}	± 10 mV_{PP}

Please note that the use of Switch Mode Power Supplies can cause problems with the sensor operation, dependent on the rate of the conducted and radiated electrical noise.
In general, Grundfos Direct Sensors does not recommend the use of Switch Mode Power Supplies.

3.2 Output Signals

The QPT Sensor is solely digital using TTL levels. It must be noted that the digital output reading is *not* ratio-metric.

For easy connection to a USB port a simple USB to TTL cable may be used. E.g. FTDI, TTL-232R, TTL to USB Serial Converter – Farnell 174-0364.

3.3 Electrical Connections

The digital flow, pressure and temperature sensor (QPT) has a four wire electrical interface:

Table 2: QPT Pin Configuration and Colour Code for standard QPT cable

Pin Number	Name	Description
1 (Yellow)	Rx	Receive Data
2 (White)	Tx	Transmit Data
3 (Green)	GND	Ground connected to PE
4 (Brown)	V_{CC}	+5V_{DC}, Supply relative to GND

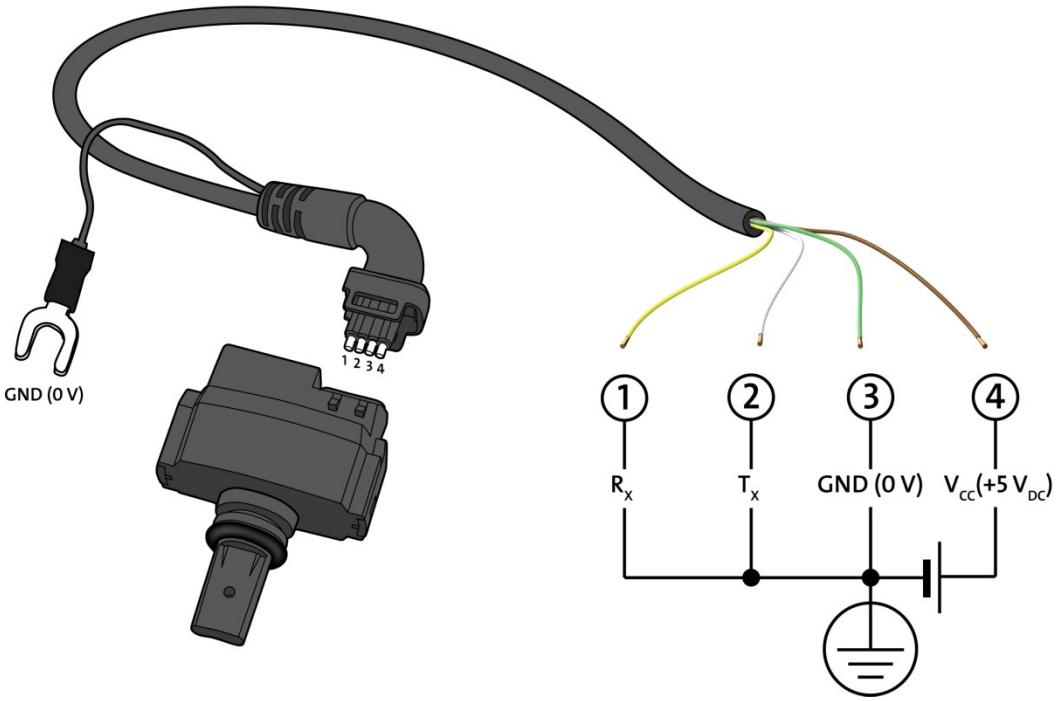
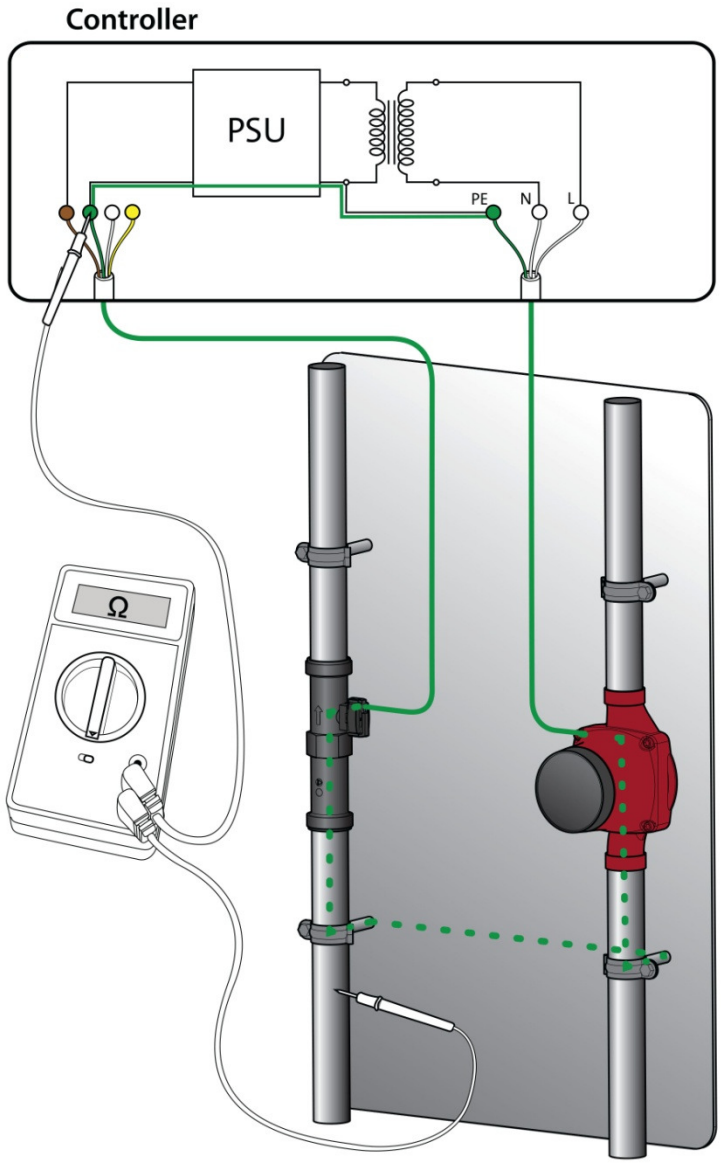


Figure 1: QPT Electrical Connections.

3.4 Grounding Technique

Using proper grounding technique is essential for correct sensor functionality. An erroneous sensor ground potential will lead to incorrect measurements.



Current path by sensor, wiring and controller

Current path by fluid, pump house, piping and mechanical fixture

Figure 2: Reading at the Multimeter must be less than 10 Ω. The illustration shows Grundfos standard colour-coded cable.

4 Data Communication

This chapter describes the QPT data protocol in details.

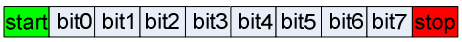
4.1 QPT Data Protocol

The QPT sensor supports a flexible sampling time, which may be as low as 200ms.

4.2 Protocol Architecture

Octet Format

The sent telegrams will have following structure (8 bit, no parity, 1 stop bit, LSB is sent first):

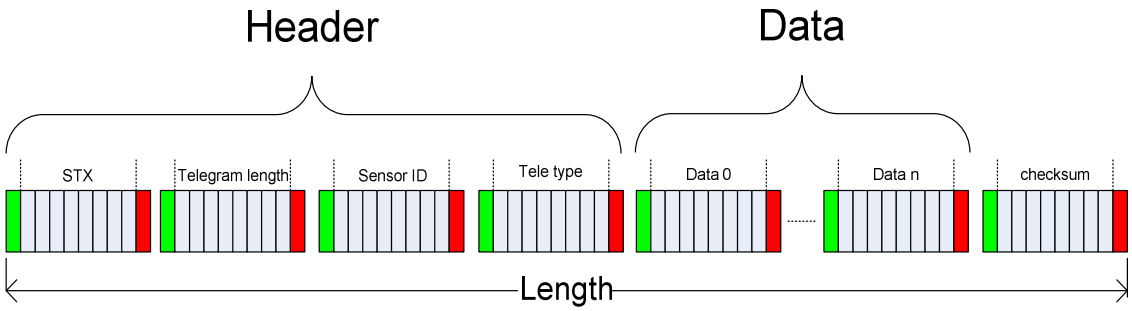


Start

One low bit.

Stop

One high bit.
Telegrams are put together as a sequence of STX, length, sensor ID, telegram type, data subsequence and checksum:



STX

Start of telegram delimiter fixed value that is always 0x72

Telegram length

1 byte for indication of the total length of the sent telegram.
The length is given by all the bytes in the telegram including STX and checksum.

Sensor ID

The Redwolf pumps may handle more sensors. To distinguish between the different sensors in the system the sensor can be given a Sensor ID. The 8 bits “Sensor ID” is used for this purpose.
“Sensor ID” = 0xEE means “broadcast” and will address every sensor no matter what sensor ID they are given. By a request to “Sensor ID” = 0xEE any sensor will respond. The sensor ID is part of the runtime parameters that can be changed while the sensor is running, but is reset to default value (0xEE) at power up.

Tele type

1 byte indicating what kind of telegram is sent. Telegrams from host to sensor have a *tele type* in the range 0x00-0x7f whereas telegrams from sensor to host have a *tele type* in the range 0x80-0xff.

RX (telegrams received by sensor)

Name	Function	Tele type
Read data	Request single measurement. Request measurements with given sample rate Stop measurement	0x00
Set runtime parameter	Host requests setting runtime parameter (e.g. communication baudrate)	0x01
Get runtime parameter	Request for runtime parameter	0x02
Request Sensor status	Request for sensor status (OK / Not OK)	0x03

TX (telegrams send by sensor)

Name	Function	Tele type
Measurement data	Sending measurement data for specific measurement.	0x80
Runtime parameter	Sending sensor runtime parameter	0x81
Sensor status	Sending sensor status.	0x83
NACK	Not Acknowledge telegram	0x8F

Data 0 .. Data n

The data octets contain the payload (or content) of the telegram. Its encoding depends on the given *tele type* field and is described in content.

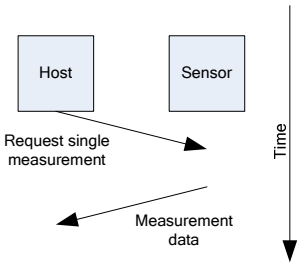
Checksum

1 byte for error check of the telegram.
The checksum is calculated by summation of all the bytes in the telegram except the checksum byte itself.

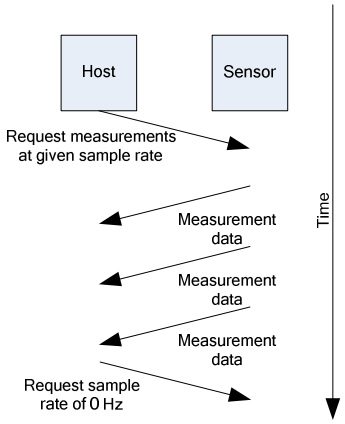
4.3 Telegrams Flows

The telegram flow will be:

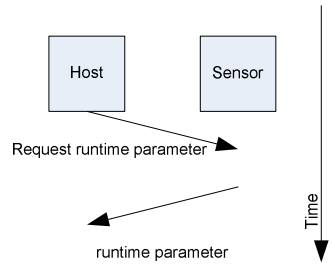
Single Measurement



Continuous Measurements (start and stop)



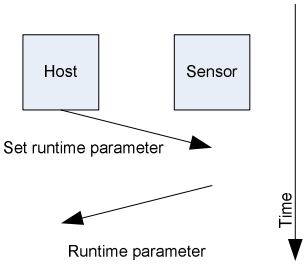
Request Runtime Parameters



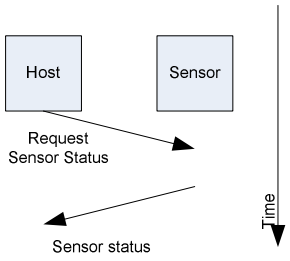
be think innovate

GRUNDFOS

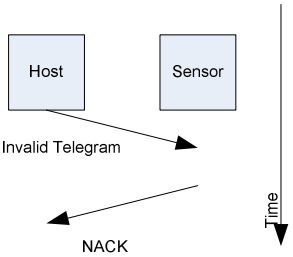
Set Runtime Parameters



For requesting sensor status



NACK

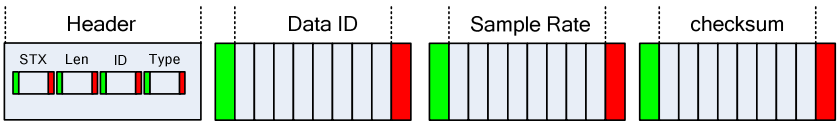


4.4 Content

Telegrams with and without content are described below.

Read data

The sensor can be setup to transmit different data at different sample rates.
The read data telegram has the following format:



Data ID is a bit field that that identifies the different measurements available in the sensor. If a bit is set the sample rate of the corresponding measurement will be updated, otherwise the sample rate will remain unchanged.

Data ID bit no.	Corresponding measurement
Bit 0	Differential pressure
Bit 1	System pressure
Bit 2	Temperature
Bit 3	Not used
Bit 4	Flow
Bit 5 – 7	Not used

The sample rate byte specifies how often the sensor shall transmit the requested measurement(s)

Sample rate value	Corresponding Sample rate
0	Single sample. Hereafter idle.
1	10 sec.
2	5 sec.
3	2 sec.
4	1 sec.
5	500 msec.
6	200 msec.
Other values	Not Allowed

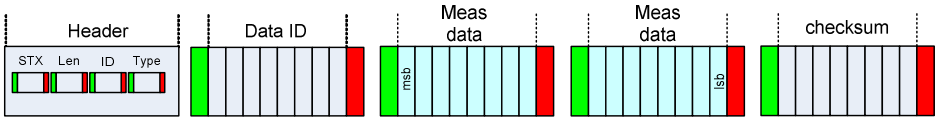
To stop the sensor sending measurement data the host shall send a telegram with the bit in data ID that corresponds to the measurement set and sample rate set to 0. By a “Read data” telegram the sensor responds with a “Measurement data” telegram. For each of the measurements given by Data ID
Note that if the baud rate is set to 1200bps then it is not possible to sample all three measurements every 200ms.

Examples for using read data telegram

Data ID value	Sample Rate Value	Sensor reaction
0x01	0	Send Differential pressure once
0x03	4	Send Differential and system pressure every second
0x02	0	Stop sending System pressure, continue sending differential pressure every second
0x04	1	Start sending Temperature every 10 seconds, continue sending differential pressure every second
0x0F	0	Stop transmitting measurement data.

Measurement data

The measurement telegram sent from the sensor has the following structure



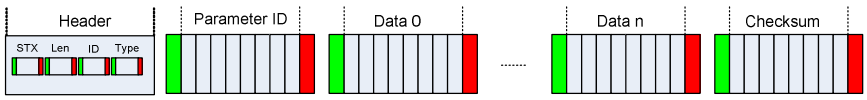
The Data ID specifies which data is being sent in the same way as in the Read data telegram. Each measurement telegram can only have one bit set in the Data ID.
The sent pressure and temperature (Meas data) are calibrated data and are scaled such that ±fullscale refers to ±16384. Meas data is a signed 16 bit integer in the range -32767..32767 (0x8001..0x7fff); -32768 (0x8000) is reserved for signaling invalid measurement or defect sensor.
Sensor returns Meas data = **0x8000** if the requested measurement is not valid or sensor is defect.
The pressure and temperature limits are defined by the runtime parameters FullScaleTemp, FullScaleDiffPress and FullScaleSysPress that define the absolute limits.

Meas data	Meaning
0x3FFF	Highest fullscale value
0x7FFF	Highest possible value (2 · fullscale)
0xC000	Lowest fullscale value
0x8001	Lowest possible value (-2 · fullscale)
0x8000	Measurement invalid or sensor defect

Examples:
If FullScaleDiffPress is 2.0 and the sensor reports 0xD8F0 (-10000), the sensor has measured that differential pressure is -1.2207
If FullScaleDiffPress is 2.0 and the sensor reports 0x5000 (20480), the sensor has measured that differential pressure is 2.5, which is outside the fullscale limit
sensor reports 0x8000, the measurement is invalid or the sensor is defect (see *Sensor Status* telegram).

Set Runtime Parameter

The runtime parameters can be written to the sensor by this telegram.
The telegrams have the following format:



The number of data bytes in the telegram depends on the parameter ID
The parameter no. describes what parameter is sent:

Parameter no.	Data byte	Parameter name	Description
0x00	1	BaudRate	The baud rate of the UART can be changed by this parameter. Value / BaudRate [bps]: 0 / 1200 1 / 4800 2 / 9600 [1] 3 / 19200 4 / 56000 5 / 115200
0x01	1	SensorID	The sensor ID

For each parameter setting telegram the sensor responds by returning the parameter to the host with the telegram described in next section.

[1] 9600 bit/sec. is default.

Request Runtime parameter

The runtime parameters can be read by the host by this telegram.
Except for the header the telegram is identical with the Set Runtime Parameter telegram described in the previous section.
Notice that some parameters are read-only: SensorID, SensorSerialNumber, FullRangeTemp, FullScaleDiffPress and FullScaleSysPress.

<table>
<tr><th>Parameter no.</th><th>Data byte</th><th>Parameter name</th><th>Description</th></tr>
<tr><td>0x00</td><td>1</td><td>BaudRate</td><td>The baud rate of the UART can be changed by this parameter.
<table>
<tr><th>Value</th><th>BaudRate [bps]</th></tr>
<tr><td>0</td><td>1200</td></tr>
<tr><td>1</td><td>4800</td></tr>
<tr><td>2</td><td>9600 [2]</td></tr>
<tr><td>3</td><td>19200</td></tr>
<tr><td>4</td><td>56000</td></tr>
<tr><td>5</td><td>115200</td></tr>
</table></td></tr>
<tr><td>0x01</td><td>1</td><td>SensorID</td><td>The sensor ID</td></tr>
<tr><td>0x02</td><td>16</td><td>SensorSerialNumber</td><td>The sensor serial number. First nine bytes are BCD encoded serial number (18 digits). The remaining bytes are reserved and shall be 0xFF to indicate invalid BCD. High nibble is first byte, low nibble is second byte...
E.g. [0x01,0x23,0x45,0x67,0x89,0x01,0x23,0x45,0x67] ➔ “012345678901234567”</td></tr>
<tr><td>0x03</td><td>4</td><td>FullRangeTemp</td><td>Floating point (IEEE754) defining absolute limits for temperature measurement. Byte order, most significant byte first.
Unit is °C.
Example: [0x00, 0x00, 0x02, 0x43] ➔ 130 °C.</td></tr>
<tr><td>0x04</td><td>4</td><td>FullScaleDiffPress</td><td>Floating point (IEEE754) defining absolute limits for differential pressure measurement. Byte order, most significant byte first.
Unit is bar.
Example: [0x00, 0x00, 0x00, 0x40] ➔ 2 bar</td></tr>
<tr><td>0x05</td><td>4</td><td>FullScaleSysPress</td><td>Floating point (IEEE754) defining absolute limits for system pressure measurement. Byte order, most significant byte first.
Unit is bar.
Example: [0x40, 0x00, 0x80, 0x41] ➔ 16 bar</td></tr>
<tr><td>0x06</td><td>32</td><td>FwVersion</td><td>Zero-terminated string containing firmware version. Format is “Vxx.yy.zz” – e.g. “V01.02.01”.
Space has been left for unofficial release information e.g. “V01.02.01 CQ30712”.</td></tr>
<tr><td>0x07</td><td>32</td><td>ProductId</td><td>Zero-terminated string containing sensor product name as written on label e.g. “DPS 0-2 GF”.</td></tr>
</table>

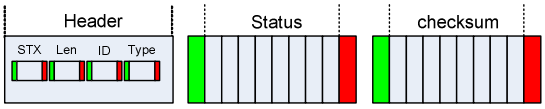
[2] 9600 bit/sec. is default.

0x08	4	FullScaleFlow	Floating point (IEEE754) defining absolute limits for temperature measurement. Byte order, most significant byte first. Unit is l/min. Example: [0x00, 0x00, 0x02, 0x43] ➔ 130 l/min.

NOTE: the *data byte* field of the telegram is ignored by the sensor in the *Request Runtime Parameter* telegram.

Sensor Status

The sensor status response telegram defines whether the sensor has detected a fault or is in the act of *power on self test*.
When sensor status is requested the sensor responds with:



Status is defined as a bitvector where each bit indicates a certain sub-status:

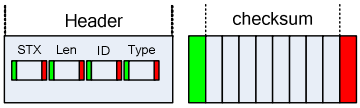
0x01	Die broken
0X02	Reset due to LVI
0x04	Reset due to watchdog
0x08	Reserved
0x10	Reserved
0x20	Reserved
0x40	Reserved
0x80	Acquiring status

Notice

- This field can only be cleared by power-cycling the sensor
- For instance if status = 0x06, sensor has been reset due to LVI and watchdog since it was powered-up
- If the sensor has been reset due to a maintenance reset, it will not be reflected by the “Reset due to watchdog” bit
- When *acquiring status* is set, the remaining bits cannot be trusted and the procedure must be repeated until *acquiring status* is cleared

NACK

If the sensor receives invalid or incomplete telegrams it shall send a Not Acknowledge telegram with the following format:



The NACK telegram does not have any content.
The sensor shall send NACK in the following situations:
The received checksum does not match the one calculated by the sensor
The sensor does not receive the next byte of a telegram within a certain time.
The timeout depends on the baud rate, and shall be set to 3 times the time it takes to transmit one byte.
The received *length* or *tele type* field is invalid.
The sensor receives bytes that are different from STX while waiting for STX