Manual
Software SPECTRO-T-1-Scope V1.0

(PC software for Microsoft® Windows® 7, 8, 10)

for sensors of SPECTRO-T-1 Series

This manual describes the installation of the PC software for the sensors of SPECTRO-T-1 series. As a support for commissioning of the color sensor this manual explains the functional elements of the Windows® user interface.

The sensors of the SPECTRO-T-1 series are of single-channel design, i.e. they acquire the analog signal that comes from a receiver and evaluate this signal. Various light sources such as white light, UV light, IR light, or a laser can be used as a transmitter. The receiver is correspondingly matched to the transmitter.

The software can be used to select various evaluation modes for the analog signal. The status of the analog signal is provided through 2 digital outputs in accordance with the selected evaluation mode.
A digital input allows external "teaching" of the sensor.
Another input allows that the digital outputs are set at different edges.

The SPECTRO-T-1 sensor allows highly flexible signal acquisition.
Two receiver modes are available. On the one hand, the analogue signal can be detected directly via a so-called transimpedance converter (current/voltage converter). The sensor, for example, can be operated in alternating-light mode (AC mode), which means the sensor is not influenced by external light, or in constant-light mode (DC mode), which provides outstanding high-speed sensor operation. With the stepless adjustment of the integrated light source, the selectable gain of the receiver signal, and an INTEGRAL function the sensor can be adjusted to almost any surface.
The second receiver mode is an integrator. Here, the receiver signal is summed up over an **adjustable** "exposure time". The advantage of this method is that extremely weak signals can be detected.

A micro-controller performs 12-bit analog/digital conversion of the analog signal, which allows recording and evaluation of the signal. Furthermore, the SPECTRO-T-1 sensor offers various options for intelligent signal processing such as e.g. dirt accumulation compensation.

Parameters and measurement values can be exchanged between PC and sensor either through RS232 or Ethernet (using an Ethernet converter). Through the interface all the parameters can be stored in the non-volatile EEPROM of the sensor.

The PC software facilitates the parameterisation, diagnostics, and adjustment of the sensor system (oscilloscope function). The software furthermore provides a data recorder function that automatically records data and stores them on the hard disk of the PC.

SPECTRO-T-1 sensors are temperature-compensated over a range of 0°C to 80°C.

When parameterisation is finished, the sensor continues to operate with the current parameters in STAND-ALONE mode without a PC.

Contents

<table>
<tr><td colspan="2">Shortcuts:</td></tr>
<tr><td>SEND</td><td>F9</td></tr>
<tr><td>GET</td><td>F10</td></tr>
<tr><td>GO</td><td>F11</td></tr>
<tr><td>STOP</td><td>F12</td></tr>
</table>

1. Installation of the SPECTRO-T-1-Scope software

The following requirements must be fulfilled for successful installation of the software:

- Microsoft® Windows® 7, 8, 10
- IBM PC AT or compatible
- VGA graphics
- Microsoft-compatible mouse
- Serial RS232 interface at the PC or USB slot or RJ45 connector
- Cable *cab-las4/PC* for the RS232 interface or *cab-4/USB* USB converter or *cab-4/ETH* Ethernet converter

Please install the software as described below:

1. You can download the software via a provided download link or, if applicable, install it via the provided software DVD. To install the software, start the 'SETUP' program in the 'SOFTWARE' folder.

2. The installation program displays a dialog and suggests to install the software in the C:\"FILENAME" directory on the hard disk.
   You may accept this suggestion with OK or [ENTER], or you may change the path as desired. Installation is then performed automatically.

3. During the installation process a new program group for the software is created in the Windows Program Manager. In the program group an icon for starting the software is created automatically. When installation is successfully completed the installation program displays "Setup OK".

4. After successful installation the software can be started with a left mouse button double-click on the icon.

Windows^{TM} is a registered trademark of Microsoft Corp.
VGA^{TM} is a trademark of International Business Machines Corp.

2. Operation of the SPECTRO-T-1-Scope software

Please read this chapter first before you start to adjust and parameterise the sensor system.

When the SPECTRO-T-1-Scope software is started, the following window appears on the Windows interface:

TIP! To avoid problems with the handling of the file path, it is advisable to run the software as administrator. You can either set this in the **Properties** under **Compatibility** or you start the software with a right click and choose **"Run as administrator"**.
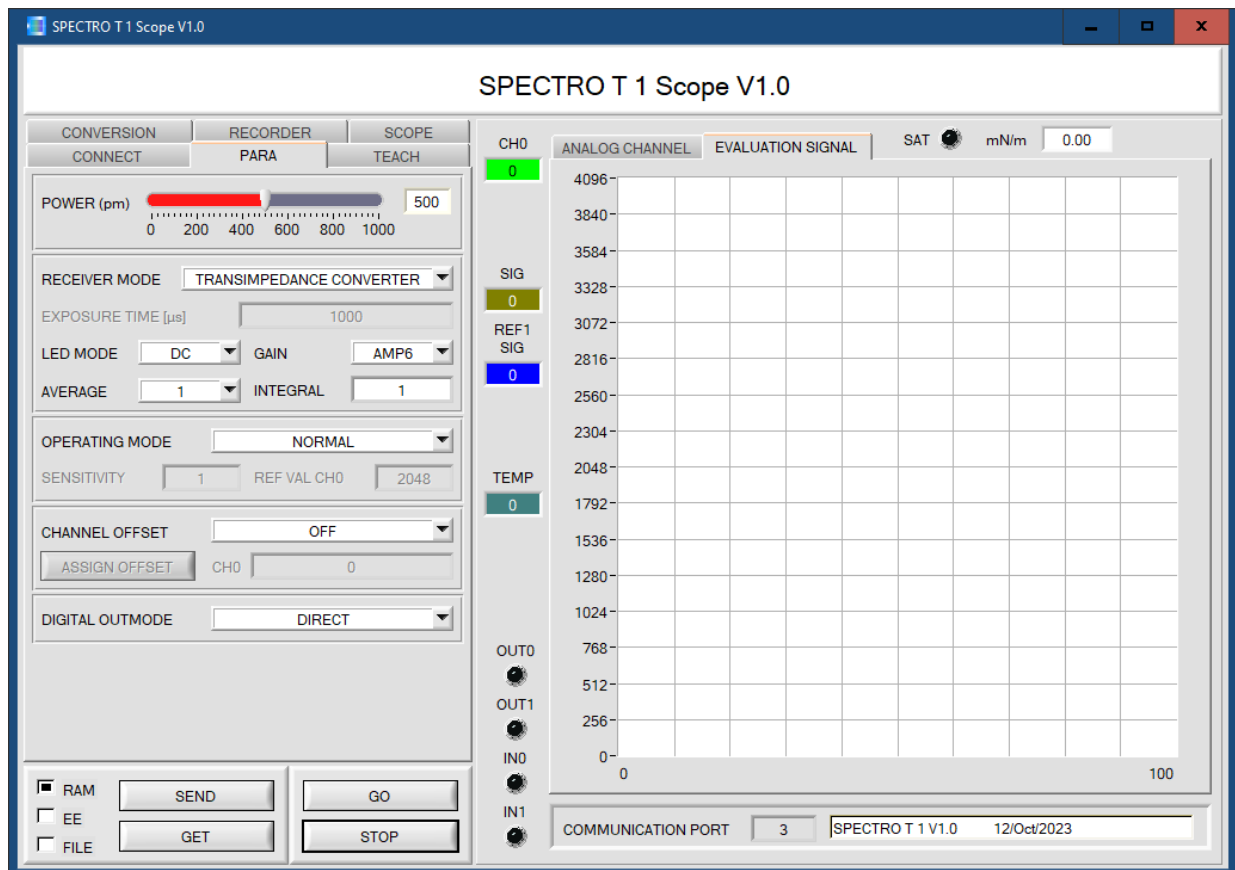
The window size and position will be the same as when the software was last closed. A double-click with the right mouse button e.g. under the minimise symbol places the window centrally in its original size.

If a connection is not established automatically, e.g. if no sensor is connected, the software can be run in OFFLINE mode. In offline mode it only is possible to exchange parameters with a file on a storage medium, which often is helpful for the purpose of analysing parameter files.

If a sensor is connected and a connection still cannot be established, either the SCOPE version (program at the PC) and the firmware version (program in the sensor) do not match, or the interface to the sensor must be correctly configured.
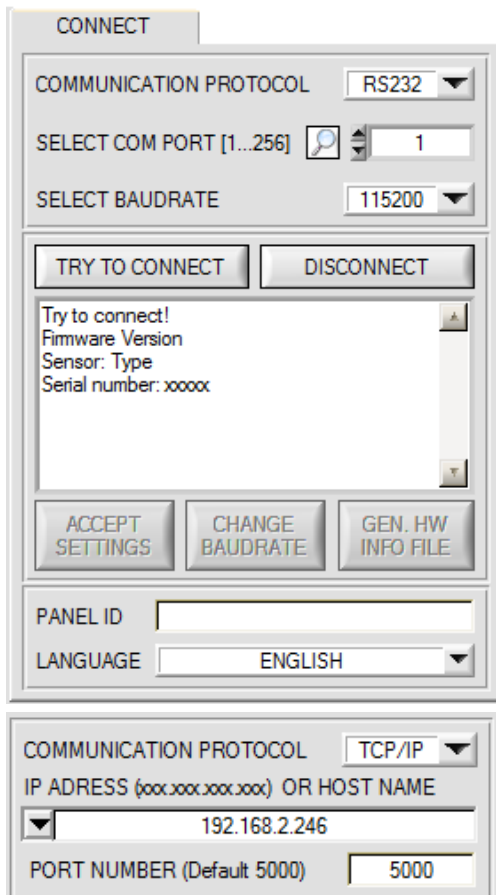If different Scope and firmware versions should be the problem, please get the Scope version that matches the firmware from your supplier.
The interface configuration is described in the CONNECT tab chapter.

Pressing the right mouse button on an individual element will call up a short help text.

Due to a better overview, parameters that are not required, displays, graphs, etc., are greyed out or switched invisible depending on the parameterization.

2.1 Tab CONNECT

CONNECT:
Pressing the **CONNECT** tab opens a window for selecting and configuring the interface.

The **COMMUNICATION PROTOCOL** function field is used for selecting either an **RS232** or a **TCP/IP** protocol.

If **RS232** is selected, a port from 1 to 256 can be selected with **SELECT COM PORT**, depending on which port the sensor is connected to. The sensor operates with a set baudrate that can be modified with **CHANGE BAUDRATE** (see below). The sensor and the user interface both must operate with the same baudrate.

At the user interface the baudrate is set with **SELECT BAUDRATE**. If after starting the software should not automatically establish a connection, the correct baudrate can be found with **SELECT BAUDRATE**.

If an converter is used, the **COM PORT** number can be determined by way of the hardware manager in the system control panel.
A click on the magnifier symbol opens a list with all the possible COM ports in the display.

An RS232 to Ethernet converter **(cab-4/ETH)** is needed if the sensor should communicate through a local network. With this converter a connection to the sensor can be established using the **TCP/IP** protocol.
Parameterisation of the **cab-4/ETH** converter (assigning of IP address, baudrate setting, ...) can be done with the **SensorFinder software** that is available free of charge on the internet.

In order to establish a connection via the converter, its IP address or HOST name must be entered in the field **IP ADDRESS (xxx.xxx.xxx.xxx) OR HOST NAME**. The DROP DOWN menu (down arrow) shows the last 10 IP addresses that were used. An address from this list can be directly selected by clicking on the respective item. The DROP DOWN list is saved and is thus always available when the software is closed.
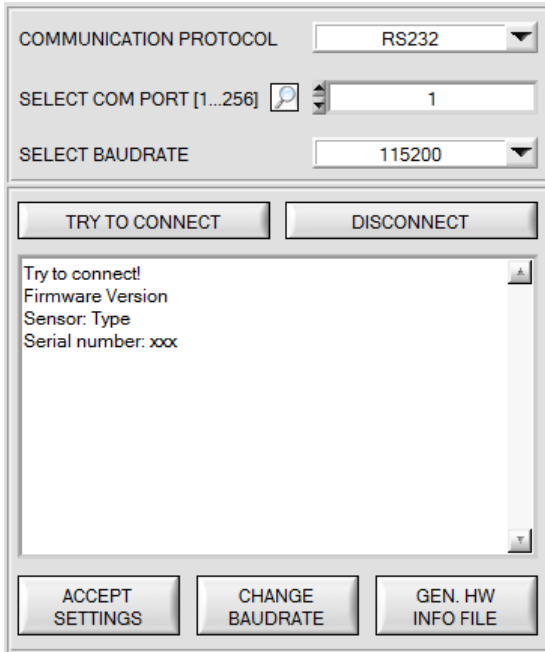The **PORT NUMBER** for the cab-4/ETH is 5000. This port number must not be changed.
When you press the **TRY TO CONNECT** button, the software tries to establish a connection with the set parameters. The communication status is shown in the display field. If the sensor answers with its FIRMWARE ID, the set connection type can be accepted by pressing **ACCEPT SETTINGS**. You will then be returned to the **PARA** tab. If you get a **TIMEOUT** message, the software could not establish a connection to the sensor. In this case please check if the interface cable is correctly connected, if the sensor is supplied with power, and if the set parameters are correct.
If a connection has been accepted by pressing **ACCEPT SETTINGS**, the software starts automatically with these settings when called the next time.

DISCONNECT disconnects the connection between sensor and PC. The software then switches to OFFLINE mode, where it is only possible to exchange parameters with a file on a storage medium.

Under **PANEL ID** a name can be entered that will be displayed at different points in the program window, and that will be recorded in different files (e.g. Record File) as well. With the input field **LANGUAGE** a language can be set with which the individual controls are displayed on the surface. This also applies to the help function that is actuated with the right mouse button.

| Please note:	The stable function of the interface is a basic prerequisite for measured value transfer from the PC to the sensor.
⚠️	Due to the limited data transfer rate through the serial RS232 interface only slow changes of the raw signals at the sensor front end can be observed in the graphic output window of the PC.
ATTENTION !	For maintaining maximum switching frequency at the sensor data communication with the PC must be stopped (press the STOP button).

The baudrate for data transfer through the RS232 interface can be set by means of the **SELECT BAUDRATE** drop down menu and **CHANGE BAUDRATE** function field.

If the baudrate should be changed, a connection must first be established by clicking on **TRY TO CONNECT**.
The **CHANGE BAUDRATE** button will then be active.

Now a new baudrate can be selected under **SELECT BAUDRATE**.
A click on **CHANGE BAUDRATE** sends the new baudrate information to the sensor.
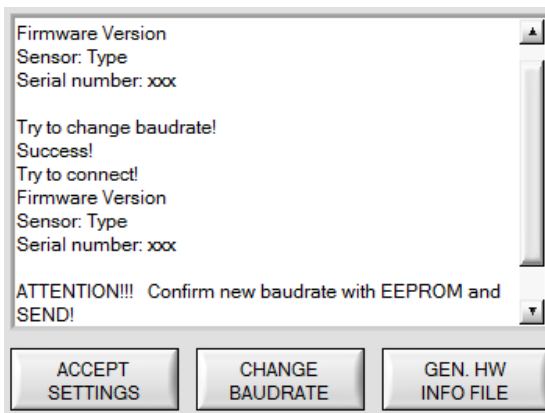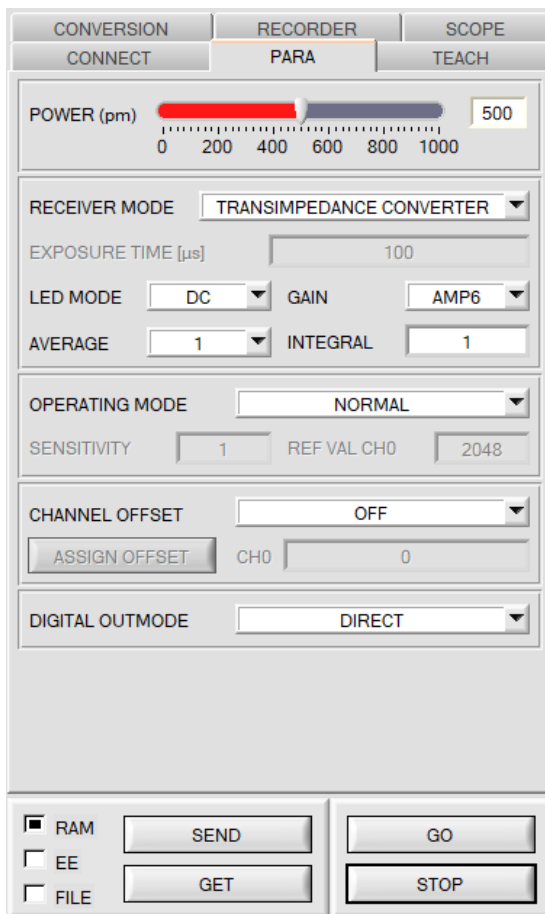
When the new baudrate information has been successfully sent, the sensor operates with the new baudrate. A window will pop up, prompting you to select **EEPROM** and then to press **SEND**. After a hardware reset the new baudrate only will be used when **EEPROM** and **SEND** have been pressed.

A click on **ACCEPT SETTINGS** saves the current interface settings, which will then be automatically set when the software is restarted.
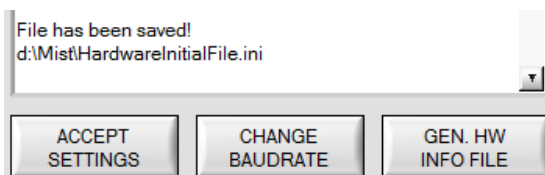
A click on the **GEN. HW INFO FILE** generates a file in which all the important sensor data are stored in encrypted form.
This file can be sent to the manufacturer for diagnostic purposes.

2.2 Tab PARA, button SEND, GET, GO, STOP

PARA:
Pressing the **PARA** tab opens a window for setting the sensor parameters.

ATTENTION!
A change of the parameter function groups only becomes effective at the sensor after actuation of the SEND button in the MEM function field!

SEND [F9]:
When the **SEND** button is clicked (or shortcut key button F9 is pressed), all the currently set parameters are transferred between PC and sensor. The target of the respective parameter transfer is determined by the selected button (**RAM, EEPROM,** or **FILE**).

GET [F10]:
The currently set values can be interrogated from the sensor by clicking on the **GET** button (or with shortcut key button F10). The source of data exchange is determined by the selected button (**RAM**, **EEPROM**, or **FILE**).

RAM:
The **RAM** is a **volatile** memory in the sensor's micro-controller, i.e. when the power at the sensor is turned off, these parameters will be lost again.
The sensor always operates with the parameters in its RAM.
If the **RAM** option is selected, a click on **SEND** writes the current parameters to the sensor's **RAM** memory, and a click on **GET** reads the parameters from the sensor's **RAM** memory.

EEPROM:
The **EEPROM** is a **non-volatile** memory in the sensor's micro-controller. When the power at the sensor is turned off the parameters in the **EEPROM** will not be lost. When power is turned on again, the parameters are loaded from the **EEPROM** to the **RAM** memory. Figuratively speaking the **EEPROM** thus is a level lower than the **RAM**. Data exchange between **PC** and **EEPROM** automatically is performed through the **RAM** memory, which means that parameters that are written to the **EEPROM** automatically are also written to the **RAM,** and data that are read from the **EEPROM** automatically are also read to the **RAM**.
If the **EEPROM** option is selected, a click on **SEND** writes the current parameters to the sensor's non-volatile **EEPROM** memory, and a click on **GET** reads the parameters from the sensor's **EEPROM**.
The **RAM** memory should always be used for parameterising the sensor. When suitable parameters have been found for the respective application, these parameters must be written to the sensor's **EEPROM** so that after restarting the sensor these parameters can be loaded from the **EEPROM** into the **RAM** memory.

FILE:
After pressing **SEND**, the current parameters can be written to a selectable file on the hard disk. With **GET** parameters can be read from such a file. When the **SEND** or **GET** button is pressed, a dialog box opens for selecting the desired file.
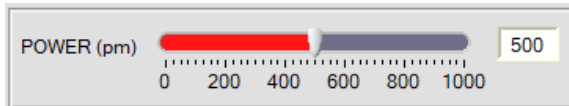**TIP!** Once suitable parameters have been found for a certain application, these should always be saved in a file on the PC.

GO [F11]:
A click on this button starts data transfer from the sensor to the PC through the serial RS232 interface.

STOP [F12]:
A click on this button stops data transfer from the sensor to the PC through the serial RS232 interface.

POWER:
In diesem Funktionsfeld kann die Senderleistung an der Sendeeinheit eingestellt werden.

The transmitter power is constantly kept at the value set with the **POWER [pm]** slider The **POWER** can be set with the slider or by entering a value in the edit-box. A value of 1000 means full intensity at the transmitter unit, a value of 0 sets the lowest intensity at the transmitter.
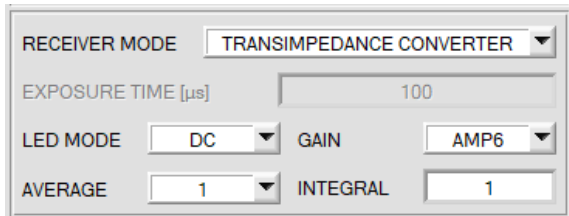
RECEIVER MODE:
In this function field, you can set how the signal present at the receiver is detected.

RECEIVER MODE = TRANSIMPEDANCE CONVERTER:
The receiver signal is detected via a so-called transimpedance converter (current/voltage converter).

In this mode, the parameters **LED MODE, GAIN, a**nd **INTEGRAL** are available.

LED MODE:
This item serves for setting the control mode for the integrated light source of the sensor.
DC: In this mode the sensor operates extremely fast. Unfortunately, the sensor is somewhat sensitive to extraneous light in DC mode, but if the extraneous light source does not directly shine into the sensor's receiver, the signal only is influenced to a very small extent.
**AC:** In this mode the sensor is insensitive to extraneous light, which is achieved by "modulating" the integrated light source, i.e. by turning the light on and off. The extraneous content in the signal is determined in off status and is simply subtracted from the on status.

GAIN:
This item is used for setting the gain of the receiver. 16 different gain levels can be set (AMP1 to AMP16). **GAIN** should be set such that with a medium **POWER** value the sensor operates in its dynamic range.

INTEGRAL:
This function field is used to set the number of scan values (measurement values) over which the raw signal measured at the receiver is summed up. This integral function allows the reliable detection even of extremely weak signals. A higher **INTEGRAL** value increases the noise of the raw signals of the receiver unit, and simultaneously decreases the maximum achievable switching frequency of the sensor.
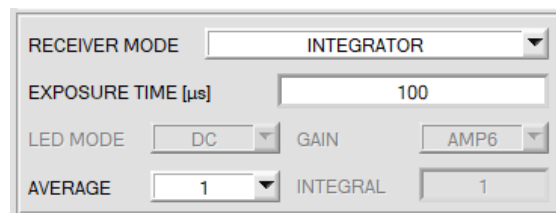
RECEIVER MODE = INTEGRATOR:
In this mode, the receiver signal is read in via an integrator. Use **EXPOSURE TIME [µs]** to set how long the receiver is "exposed". The longer, the stronger the signal.
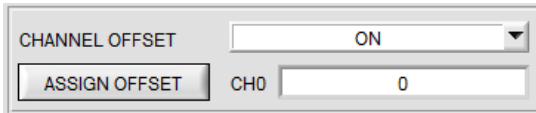
AVERAGE:
This function field is used for adjusting the number of scanning values (measurement values) over which the raw signal measured at the receiver is averaged. A higher **AVERAGE** default value reduces noise of the raw signals at the receiver unit and there will be a decrease of the maximal available switching frequency of the sensor.

TIP:
RECEIVER MODE = TRANSIMPEDANCE CONVERTER is used when the receiver signal is relatively strong, and the evaluation must be fast.
RECEIVER MODE = INTEGRATOR is used when you need to detect very weak signals.

CHANNEL OFFSET:
Some optics units may show a certain level of intrinsic reflection.
To avoid an increase of this offset when using the integral function (parameter **INTEGRAL**), the offset can be eliminated by means of offset calibration.

With **CHANNEL OFFSET = ON** the offset value for **CH0** is subtracted from the current signal.

To determine the current offset, first work with **CHANNEL OFFSET = OFF**.
Place the surface with the offset to be subtracted in front of the sensor.
If you wish to compensate the intrinsic reflection of an optics unit, let it look into empty space.
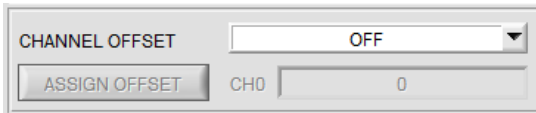Press **GO** to start the data exchange.
Make sure that the sensor is correctly parameterised (**POWER**, **RECEIVER MODE**, etc.).
Press **STOP** and then select **CHANNEL OFFSET = ON**.
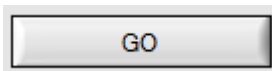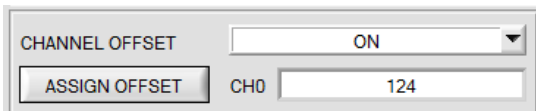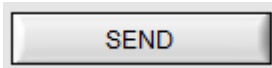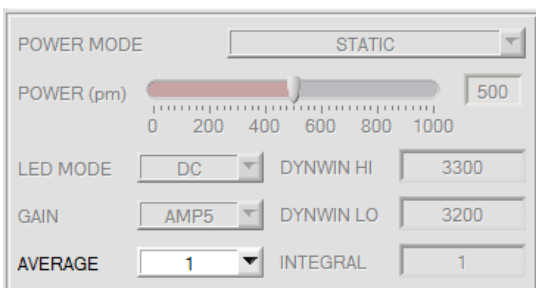
With **ASSIGN OFFSET** the current channel value is adopted as the offset value.
Then press **SEND** to save the data to the sensor.

ATTENTION!
If the sensor should already run with an offset, this offset must first be set to zero.
This is not necessary if **CHANNEL OFFSET** was **OFF** before.

With **CHANNEL OFFSET = ON** certain setting possibilities that may influence the offset are grayed out. If you wish to change such settings you must first select **CHANNEL OFFSET = OFF**, then make the respective settings, and finally determine the offset anew.

OPERATING MODE can be used to set the sensor's operating mode.
The sensor works normally with **NORMAL**.
i.e. the signal CH0 recorded on the receiver is assigned to the evaluation signal SIG.
SIG=CH0

DIFFERENTIATOR means that the analog input signal CH0 is picked up differentially by the receiver.
i.e. you determine a change in **CH0** and assign this to **SIG**.
SIG = mean value of **CH0** via number **SENSITIVITY** values **–** current value of **CH0 + 2048**

$$\text{SIG} = \frac{\sum_0^{Sensitivity} CH0}{Sensitivity} - CH0 + 2048$$

The **SENSITIVITY** parameter is used to set the sensitivity of the differentiator.

Example:
If e.g. **SENSITIVITY=32** is set, a mean **of CH0** is constituted from 32 determined values.
The difference of this average from the current value is added to 2048.
This means that for **SIG** the resulting value is 2048 if there is no change.
If for example the distance from the surface or the surface structure changes, the resulting value will show a deflection that may lie below or above the value of 2048.
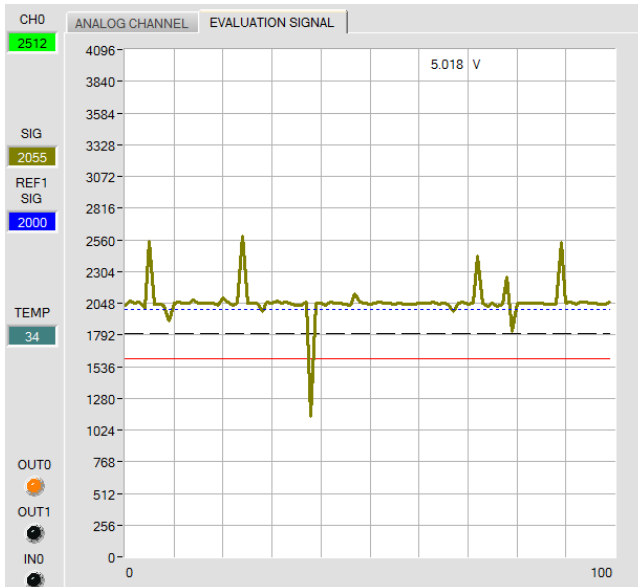
With the DELTA CH0 INTEGRATOR, the deviation of CH0 to a reference value for CH0 (REF VAL CH0) is determined, standardised to 4096 and added with 2048.
SIG=(((REF CH0 – CH0) over number of **SENSITIVITY** values**) * 4096 / REF CH0) + 2048**

$$\text{SIG} = \frac{\sum_0^{Sensitivity} (REF\ CH0 - CH0)}{REF\ CH0} * 4096 + 2048$$

A summation factor **REF CH0 – CH0** is determined with **SENSITIVITY**.

Example:
If e.g. **SENSITIVITY**=32 has been set, 32 detected values **REF CH0 - CH0** will be added.
The sum is standardised to 4096.
The standardised value is added with 2048.
You receive a value of 2048 for **SIG** if the sum of **REF CH0 – CH0 = 0**.
If e.g. the clearance to the surface or the surface property changes, you will receive a peak below or above 2048.
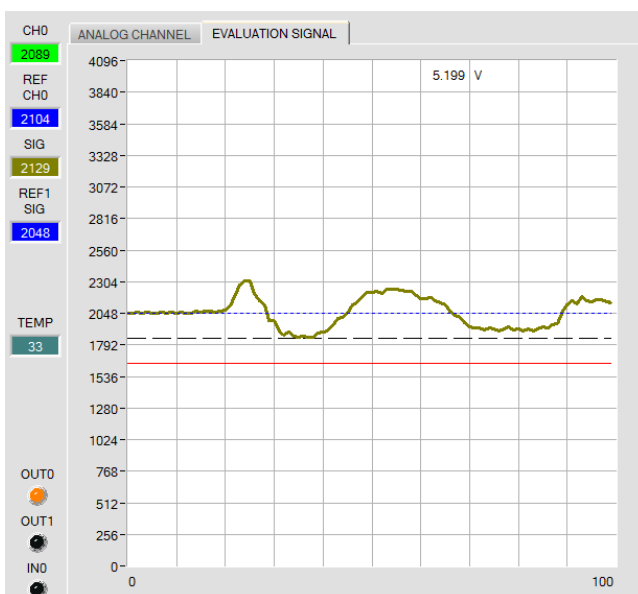
| DIGITAL OUTMODE | DIRECT ▼ |

DIGITAL OUTMODE:
This function field is used to determine the operating mode of digital outputs **OUT0** and **OUT1** when the tolerance threshold is **exceeded**.

Depending on **OUTMODE** and **THRESHOLD MODE** the status of **OUT0** and **OUT1** is as follows:

	THRESHOLD MODE LOW and HI	THRESHOLD MODE WIN
OUTMODE OFF	OUT0=0 VDC OUT1=0 VDC	OUT0=0 VDC OUT1=0 VDC
OUTMODE DIRECT	OUT0=0 VDC OUT1=0 VDC	OUT0=0 VDC OUT1=0 VDC if **SIG** is lower than the tolerance window OUT1=0+24 VDC if **SIG** is higher than the tolerance window
OUTMODE INVERSE	OUT0=+24 VDC OUT1=0 VDC	OUT0=+24 VDC OUT1=0 VDC if **SIG** is higher than the tolerance window OUT1=0+24 VDC if **SIG** is lower than the tolerance window

DIR RIS EDG of IN1 (Direct at a rising edge of IN1) **INV RIS EDG of IN1** (Inverse at a rising edge of IN1) **DIR FAL EDG of IN1** (Direct at a falling edge of IN1) **INV FAL EDG of IN1** (Inverse at a falling edge of IN1)	Unlike **DIRECT** and **INVERSE**, the outputs here are updated only at a positive (RIS) or a negative (FAL) edge at input IN1.

On the user interface the respective status of outputs is visualised by way of the **OUT0** and **OUT1** LEDs.

2.3 Tab TEACH

When the **GO** button is pressed, data transfer from the sensor to the PC is started.
The analog signal **CH0** and the the evaluation signal **SIG** are shown in a graph and in a numerical display.
Using the **TEACH** button is the easiest method for "teaching" an analog signal.
With a click on the **TEACH** button the current **SIG** value is accepted as the **TEACH VAL 1 SIG**.
With a click on **SEND** the **TEACH VAL 1 SIG** becomes the new reference value **REF1 SIG**.
The switching threshold and the hysteresis threshold are then set based on **REF1 SIG**.
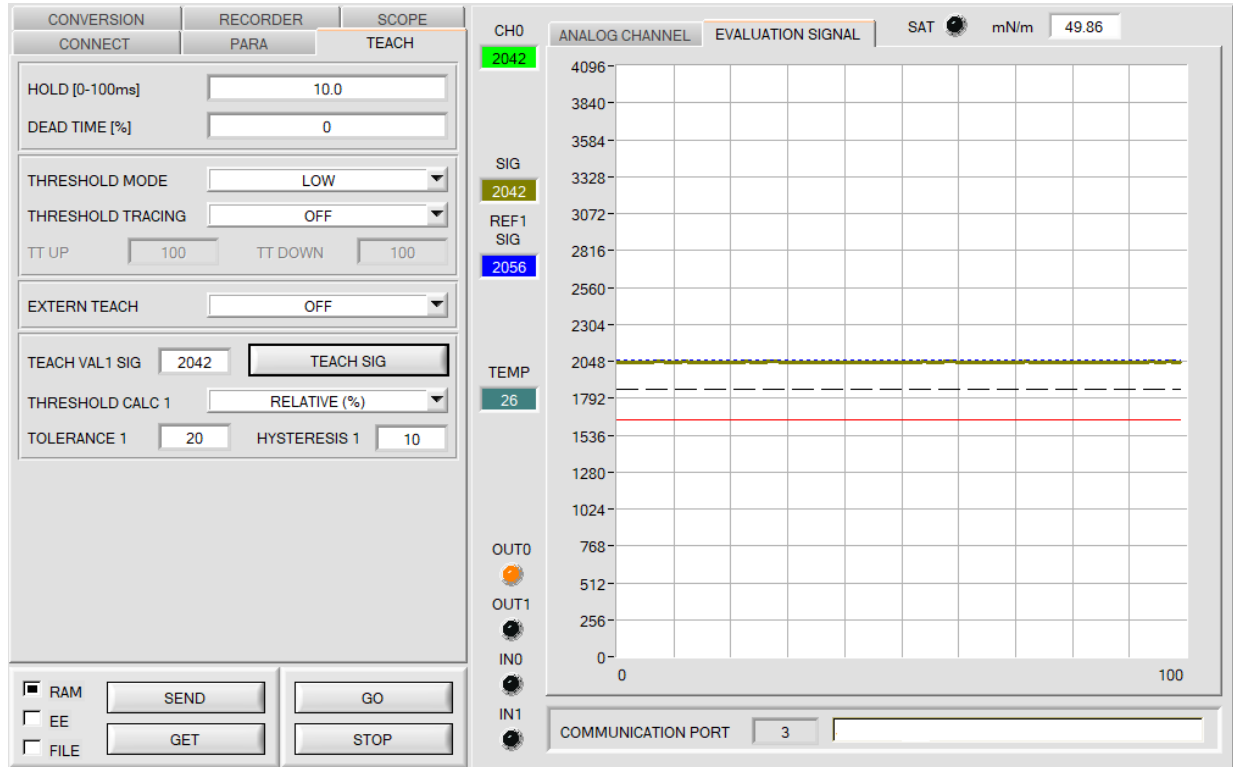
HOLD:
The sensor operates with minimum scanning times in the magnitude of less than 10μs. Entering a suitable value in the **HOLD** edit box sets a pulse lengthening of up to 100 ms at the sensor's digital output. This allows a PLC to reliably detect short changes of switching states.

DEAD TIME [%]:
This function field can be used to activate a dynamic dead time.

The dead time function can best be explained by way of an example.
Because of the application the sensor provides a regular switching signal of 100ms. These 100ms are interpreted as 100%. If a **DEAD TIME** of 20% is set, the sensor is "dead" for 20% after every switching signal. It ignores any fault that occurs within 20ms after the last switching process, and consequently also does not switch. For example this allows the correct detection of unclear edges without the sensor providing multiple pulses.
If the speed changes from 100ms to 200ms, 20ms dead time will become 40ms due to percentage calculation, which is why this is referred to as a dynamic dead time.

It is advisable to use a dead time, if multiple pulses cannot be suppressed with a suitable **HYSTERESIS 1**.

DEAD TIME [%] = 0 deactivates the dead time.

THRESHOLD MODE:
In this function field one of the possible positions of the switching threshold and hysteresis threshold with respect to the reference value **REF1 SIG** can be selected.
The reference value is learned either by means of the software or through the external input IN0. Furthermore, automatic threshold tracing can be activated with **THRESHOLD TRACING**, which means that **REF1 SIG** is cyclically re-determined.

Based on **REF1 SIG** and with the help of **TOLERANCE 1** and **HYSTERESIS 1** the switching threshold and the hysteresis threshold are calculated differently with **THRESHOLD MODE LOW, HI** and **WIN**.

If **THRESHOLD CALC 1 = ABSOLUTE(digit)** is selected, the thresholds are calculated absolutely in digits to **REF1 SIG**, i.e. **TOLERANCE 1** and **HYSTERESIS 1** are directly added to or subtracted from **REF1 SIG**.

If **THRESHOLD CALC 1 = RELATIVE(%)** is selected, the thresholds are calculated relatively to **REF1 SIG** = 100%, i.e. **TOLERANCE 1** and **HYSTERESIS 1** are relatively added to or subtracted from **REF1 SIG**.

The different lines in the following graphic windows have the following meaning:
Dark yellow line = current signal **SIG**
Blue dotted line = reference value **REF1 SIG**
Red line = switching threshold
Black dashed line = hysteresis threshold
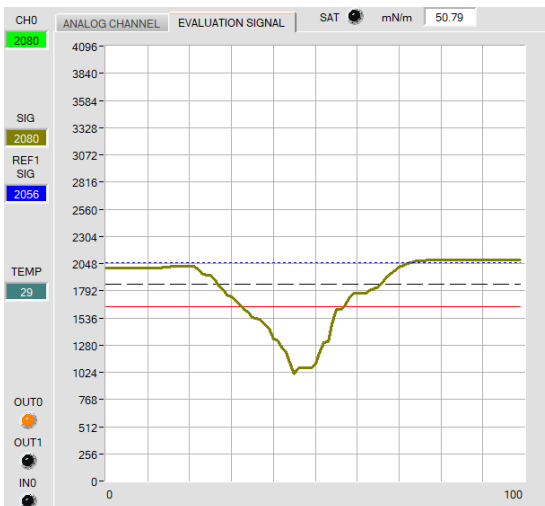
THRESHOLD MODE = LOW:
Switching threshold = **REF1 SIG** – **TOLERANCE 1**
Hysteresis threshold = **REF1 SIG** - **HYSTERESIS 1**

When the current signal value **SIG** falls below the switching threshold, the digital output **OUT0** is set to error.
When the current signa value rises above the hysteresis threshold again, the error output is reset again.
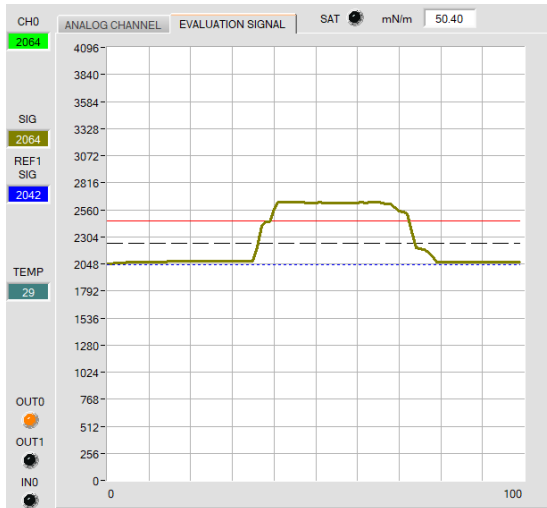
THRESHOLD MODE = HI:
Switching threshold = **REF1 SIG + TOLERANCE 1**
Hysteresis threshold = **REF1 SIG + HYSTERESIS 1**

When the current signal value **CH0** rises above the switching threshold, the digital output **OUT0** is set to error.
When the current measurement value falls below the hysteresis threshold again, the error output is reset again.
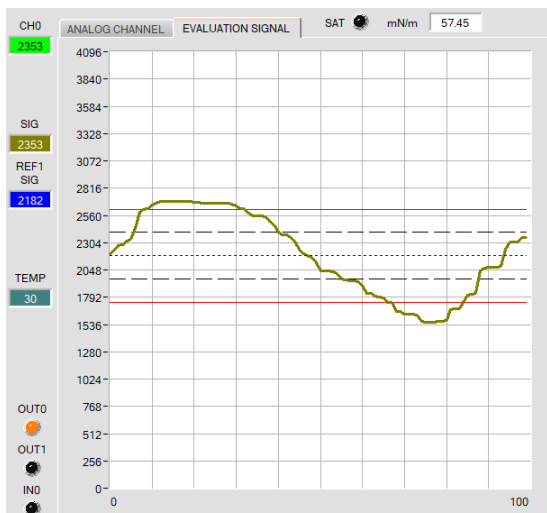
THRESHOLD MODE = WIN:
Upper switching threshold = **REF1 SIG + TOLERANCE 1**
Upper hysteresis threshold = **REF1 SIG + HYSTERESIS 1**

Lower switching threshold = **REF1 SIG - TOLERANCE 1**
Lower hysteresis threshold = **REF1 SIG - HYSTERESIS 1**

The switching thresholds form a symmetrical tolerance band around the current reference value. When the current signal value **SIG** leaves this tolerance band, the digital output **OUT0** is set to error.
The error output is reset again when the current measurement value lies below/above the hysteresis threshold again.
Depending on **OUTMODE**, the output **OUT1** is set or reset when the signal leaves the tolerance window upwards or downwards.
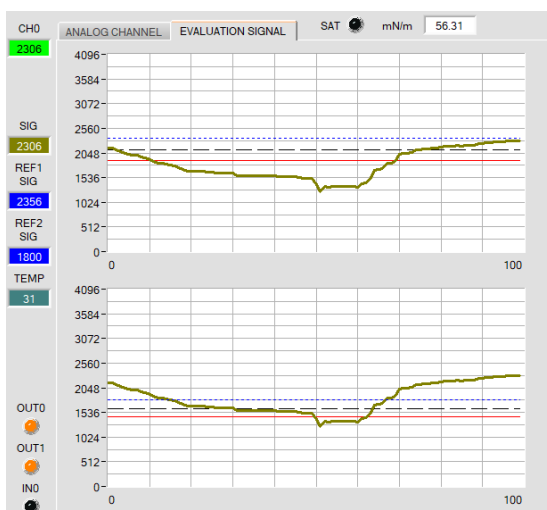
THRESHOLD MODE = 2 TRSH:
Two switching thresholds are available in this mode.

Switching threshold 1 (2) = **REF1 SIG(2) – TOLERANCE 1(2)**
Hysteresis threshold 1 (2) = **REF1 SIG(2) – HYSTERESIS 1(2)**

When the current signal **SIG** measurement value drops below switching threshold 1 or 2, digital output **OUT0** or **OUT1** is set to error.
When the current measurement value becomes higher than hysteresis threshold 1 or 2 again, the error output is reset again.

THRESHOLD TRACING	ON TOL	
TT UP	50	TT DOWN	1000

THRESHOLD CALC 1	RELATIVE (%)	
TOLERANCE 1	20	HYSTERESIS 1	10

THRESHOLD TRACING:
Automatic threshold tracing can be activated in this function field.
Actually the reference value **REF1 SIG** cyclically follows a changing **SIG** value.
Switching threshold and hysteresis threshold then are set based on **REF1 SIG**.
Generally this is referred to as threshold tracing.

OFF: Automatic threshold tracing is deactivated.

ON TOL and **ON CONT:** Automatic threshold tracing is activated. The current reference value **REF1 SIG** is cyclically adapted if the current **SIG** value decreases, e.g. due to increasing dirt accumulation.
With **ON TOL** tracing is performed when the current signal lies within the tolerance.
With **ON CONT** tracing is performed continuously.

TT UP and **TT DOWN**: In this function field a time constant for the speed of automatic threshold tracing can be set.
TT UP (THRESHOLD TRACING UP): When the current **SIG** value rises, **REF1 SIG** is increased by one digit with the set delay.
TT DOWN (THRESHOLD TRACING DOWN): When the current **SIG** value falls, **REF1 SIG** is decreased by one digit with the set delay.

A value between 0 and 60000 can be selected for **TT UP** and **TT DOWN**.
One increment means a delay of 100 microseconds. However, tracing cannot be faster than the scan frequency that can be achieved with **AVERAGE**.
Value 0:   Minimum time delay, fastest tracing.
Value 60000: Maximum time delay, slowest tracing.

The calculation of the switching thresholds depends on **THRESHOLD MODE** (see **THRESHOLD MODE**).
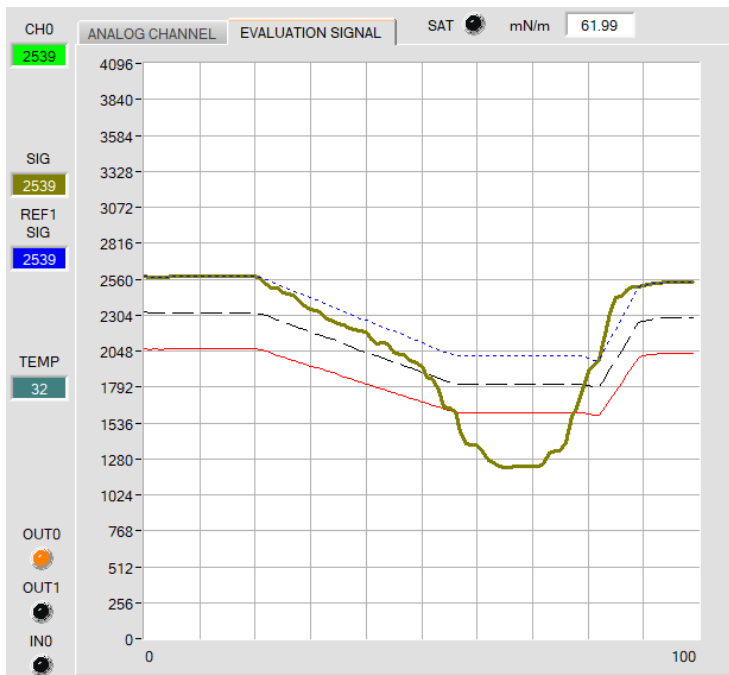
In this example a value of 50 was specified for **TT DOWN**. This means that downward threshold tracing is performed at a relatively slow speed.
In the chart this can be clearly seen in the blue dotted reference line.

The measurement value **SIG** (dark yellow line) falls relatively fast, compared to this the reference value **REF1 SIG** follows much slower. This is a typical case that occurs when increasing dirtying of the sensor must be compensated.

With **ON TOL**, threshold tracing is stopped when the current measurement value falls below the switching threshold. When **SIG** rises above the hysteresis threshold again, threshold tracing becomes active again. In this example upward threshold tracing is fast because a small value was specified for **TT UP**.

EXTERN TEACH:
This function field is used to specify how sensor "teaching" should be performed.

EXTERN TEACH = OFF:
Using the **TEACH** button is the easiest method for "teaching" an analog signal.
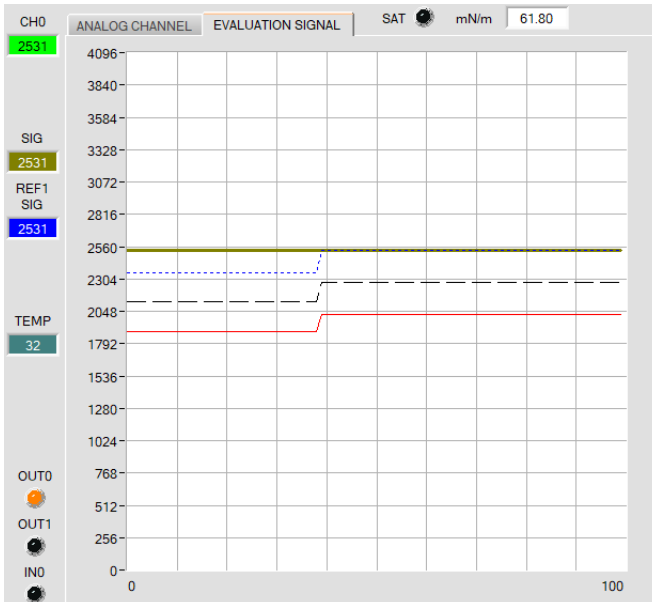
With a click on the **TEACH** button the current **SIG** value is accepted as the **TEACH VAL 1 SIG**.

With a click on **SEND** the **TEACH VAL 1 SIG** becomes the new reference value **REF1 SIG**.

The switching threshold and the hysteresis threshold are then set based on **REF1 SIG**.

The **TEACH** button is not active if either **EXTERN TEACH** or **THRESHOLD TRACING** is not **OFF**.

Teach via +24V to IN0
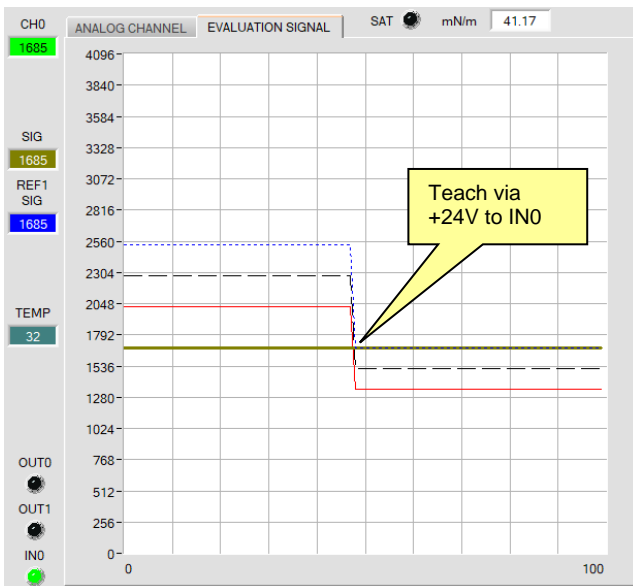
EXTERN TEACH = DIRECT:
When there is a positive edge at input IN0, the current **SIG** value becomes the new reference **REF1 SIG**.

The switching threshold and the hysteresis threshold are then set based on **REF1 SIG**.

The new reference **REF1 SIG** only is saved in the sensor's **RAM** and not in its **EEPROM**.
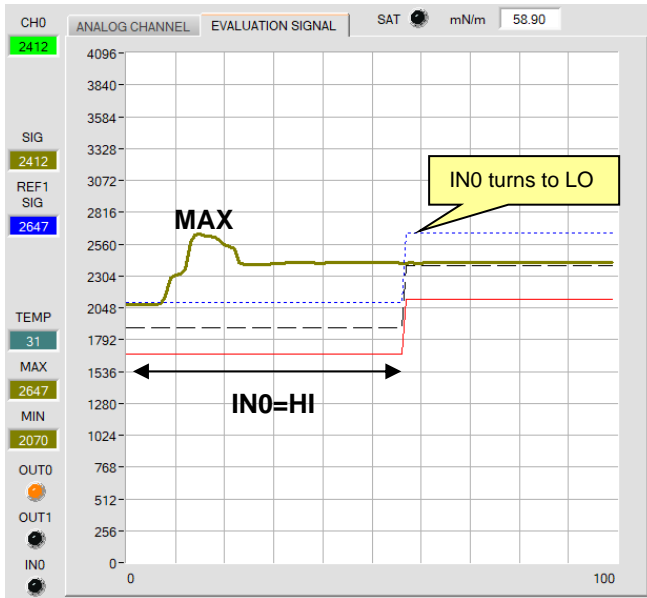
EXTERN TEACH = MAX:
While input IN0=HI (+24V), the maximum **SIG** value is determined and becomes the new reference **REF1 SIG** when IN0 changes to low.

The switching threshold and the hysteresis threshold are then set based on **REF1 SIG**.

The new reference **REF1 SIG** only is saved in the sensor's **RAM** and not in its **EEPROM**
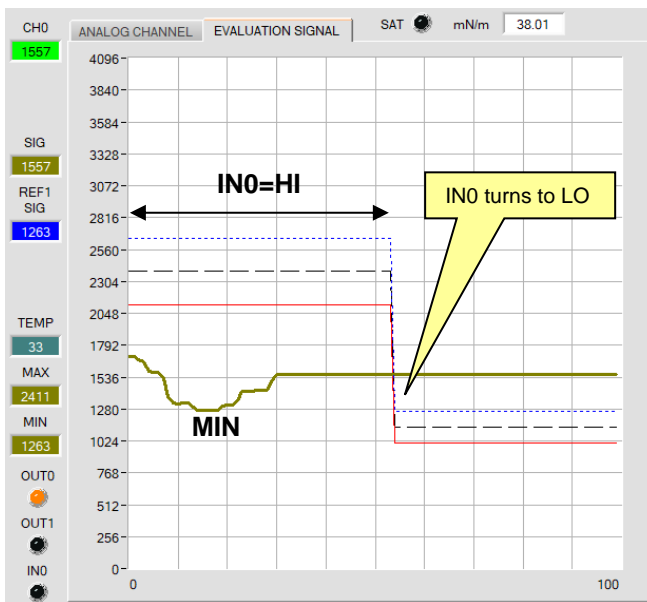
EXTERN TEACH = MIN:
While input IN0=HI (+24V), the minimum **SIG** value is determined and becomes the new reference **REF1 SIG** when IN0 changes to low.

The switching threshold and the hysteresis threshold are then set based on **REF1 SIG**.

The new reference **REF1 SIG** only is saved in the sensor's **RAM** and not in its **EEPROM**.
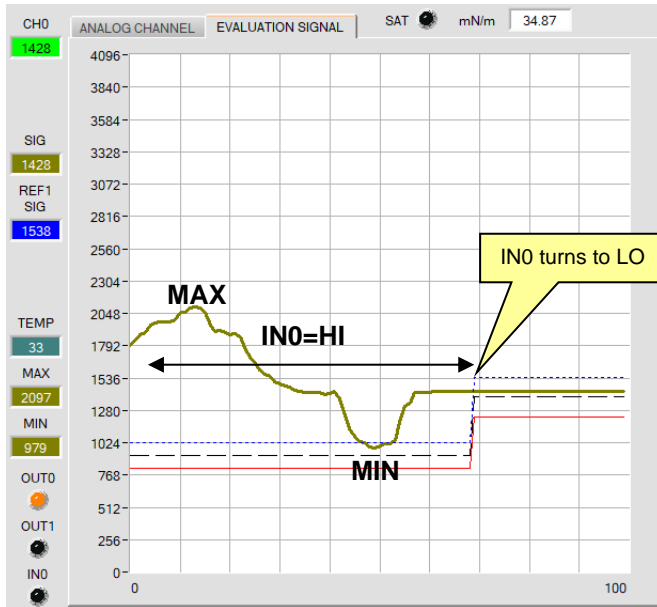
EXTERN TEACH = (MAX+MIN)/2:
While input IN0=HI (+24V), the maximum and minimum **SIG** values are determined.
When IN0 changes to low, the new reference **REF1 SIG** is set exactly between MAX and MIN.

The switching threshold and the hysteresis threshold are then set based on **REF1 SIG**.

The new reference **REF1 SIG** only is saved in the sensor's **RAM** and not in its **EEPROM**

2.4   Graphic display elements

The software provides various display elements and a graphic window for the visualisation of all the data that are important for parameterisation. The individual display elements and the graph are explained in the chapter below.
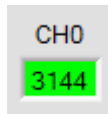
CH0:
This display shows the current measurement value.

In the graph **CH0** is visualised as a green line.
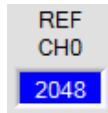
REF CH0:
The reference value for **CH0** with **OPERATING MODE = DELTA CH0 INTEGRATOR** is shown in this display. This value is usually **REF VAL CH0.** If **EXTERN TEACH = DIRECT** or **DYN** is set, then the value actually measured during the learning procedure will change from **CH0** to **REF CH0.**.

SIG:
The current signal value **SIG** is shown in this display.
In graphs, **SIG** is visualised as a dark yellow line.
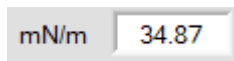
The **EVALUATION SIGNAL SIG** can be attributed a specific conversion value in a table.
This value is displayed here.
See CONVERSION tab (cf. chapter 2.5).
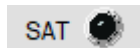
SAT:
The **SAT LED** indicates whether channel CH0 is in saturation (overdriven). It is then red.

It must always be ensured that the sensor is parameterised so that the SAT saturation LED is off (black).
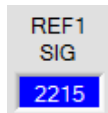
REF1 SIG:
This display shows the current reference value for **SIG**. This value is the basis for the calculation of switching threshold and hysteresis threshold.
If **THRESHOLD TRACING** and **EXERN TEACH** are set to **OFF**, the **TEACH VAL 1 SIG** is the reference value.
If **THRESHOLD TRACING** is set to **ON**, automatic threshold tracing is activated. The current reference value **REF1 SIG** is cyclically adapted to compensate a lowering of the current **SIG** value, e.g. due to increasing dirt accumulation.
Depending on the setting of **EXERN TEACH** the reference **REF1 SIG** can also be taught through input IN0 with various methods.

In the graph the reference value **REF1 SIG** is represented as a blue dotted line.

MAX and MIN:
If with **EXTERN TEACH** a search for a minimum and/or maximum **SIG** value is necessary, this value is respectively these values are shown in these displays.
The two displays only are active when they are needed.
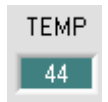
TEMP:
This display shows the temperature prevailing in the sensor housing.
The display **DOES NOT** show degrees Centigrade or Fahrenheit.

OUT0 und OUT1:
These LEDs visualise the physical status of outputs OUT0 and OUT1.
When the LED is black, the output value is 0V.
When the LED is orange, the output value is +24V.

IN0 and IN1:
These LEDs visualise the physical status of inputs IN0 and IN1.
When the LED is black, the input value is 0V.
When the LED is green, the input value is +24V.

GRAPH:

In the graphic display window, the EVALUATION SIGNAL tab displays the switching threshold (red line), the hysteresis threshold (black dashed line), the reference **REF1 SIG** (blue dotted line) as well as the current signal **SIG** (dark yellow line) depending on the set parameters.

In the graph below the parameter settings are as follows:

THRESHOLD MODE = LOW
THRESHOLD TRACING = OFF
EXTERN TEACH = OFF
TEACH VAL 1 SIG = 2000 (→ REF1 SIG=2000)
THRESHOLD CALC 1 = RELATIVE (%)
TOLERANCE 1 = 20
HYSTERESIS 1 = 10

This results in a switching threshold of **TEACH VAL 1 SIG** minus **TOLERANCE 1** (red line), and in a hysteresis threshold of **TEACH VAL 1 SIG** minus **HYSTERESIS 1** (black dashed line).

In the **ANALOG CHANNEL** tab, only the registered analogue signal **CH0** is displayed.
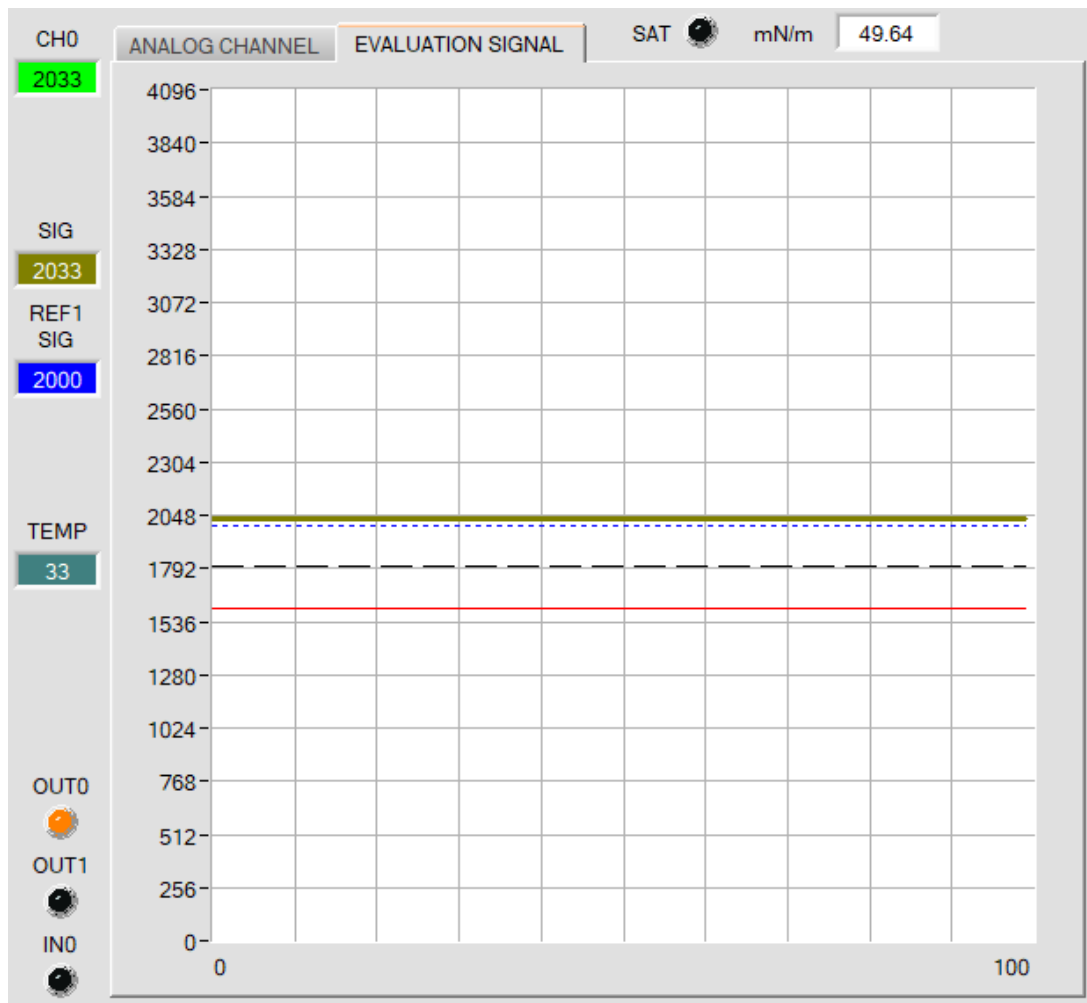
2.5 Tab CONVERSION

In the **CONVERSION** tab a certain conversion value can be assigned to the **SIG** signal.
SIG is the measured value calculated from **CH0**. It is a measure for the condition of the surface that is presented to the sensor and is displayed digitally from 0 to 4095 digits.
In order to characterise the surface in a more comprehensible way, a specific physical value can be assigned to the value **SIG** in the **CONVERSION TABLE**. The unit of measurement of this value is selected in the parameter **SIG UNIT** (selectable units for **SIG UNIT** are: mN/m, µm, g/m², mg/m², 10RFU, 100RFU, 1000RFU). **SIG** can take values between 0.00 and 100.00 to two decimal places.
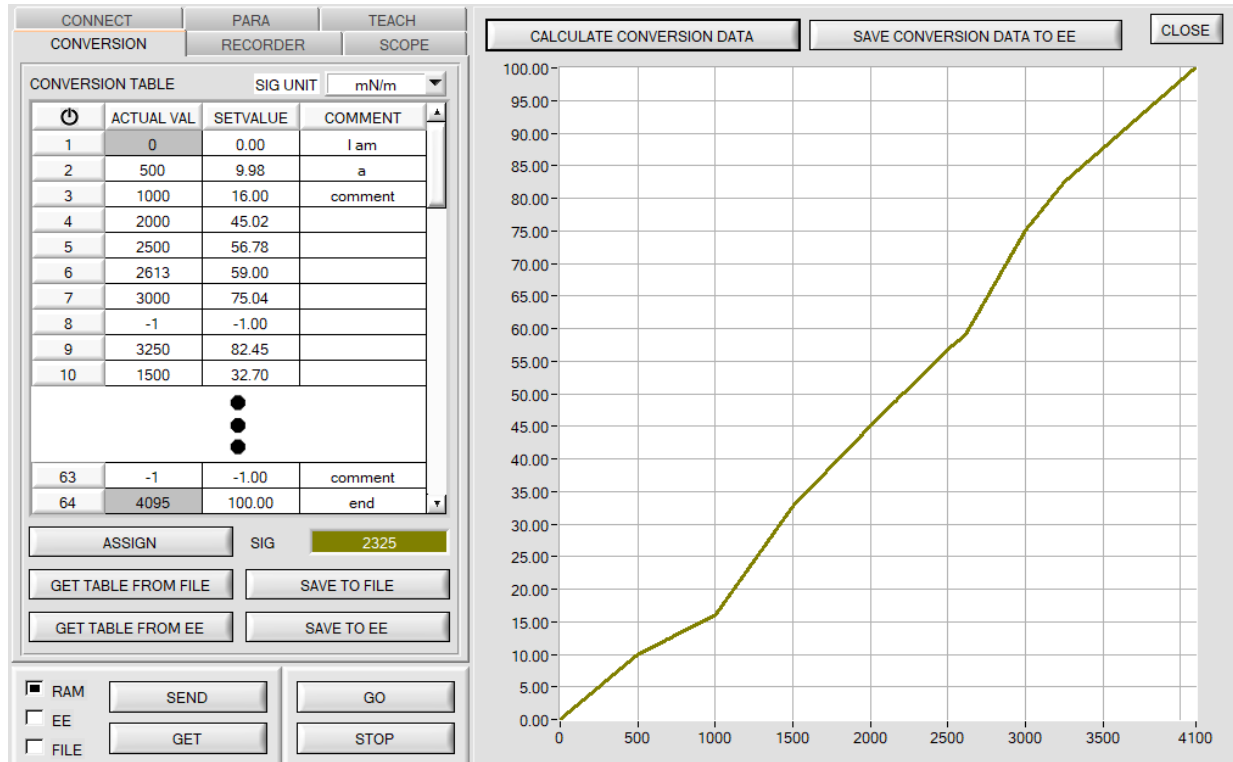
When the **GO** button is pressed the current **SIG** value is shown in the **SIG** display.
Use the mouse to select a cell in the **ACTUAL VALUE** column and press **ASSIGN**. This value is then written to the cell in the table. Then enter the physical value to which the current **SIG** value corresponds in the **SETVALUE** column. It can have values between 0.00 and 100.00.
A total of 64 conversion values can be entered this way.
NOTE: Instead of the comma, a dot must be entered as a separator.

The first and the last row must be filled in (see **CONVERSION TABLE** row 1 and 64).
Rows that are not used must be disabled with **-1** (see **CONVERSION TABLE** row 8 or 63).
The **ACTUAL VALUE** column has a value range of 0 to 4095.
The **SETVALUE** column has a value range of 0.00 to 100.00.

With a click on **CALCULATE CONVERSION DATA**, a corresponding conversion value for each **SIG** value from 0-4095 is calculated from the interpolation points of the conversion table and displayed graphically. The X-axis is the value of **SIG (ACTUAL VAL)**. The Y-axis shows the corresponding set value (**SETVALUE**).
The graph can be zoomed by holding down the control key and drawing a window with the left mouse button. Click on **ZOOM 1:1** to display the graph in normal size again.

With **SAVE CONVERSION DATA TO EE** the conversion data can be sent to the EEPROM of the sensor.

Moreover, **GET TABLE FROM FILE** and **SAVE TO FILE** can be used to load the conversion table from a file or to save it in a file.
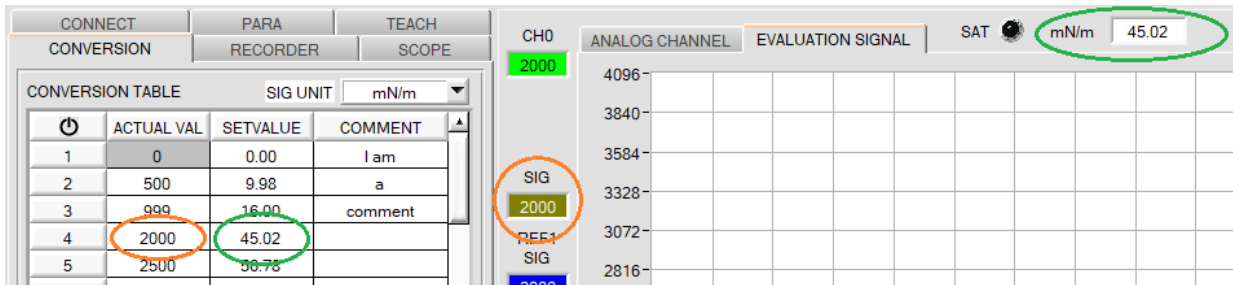
With **GET TABLE FROM EE** and **SAVE TO EE** the conversion table can be loaded from the sensor's EEPROM or saved to the sensor's EEPROM.

An individual comment for every line can be entered in the **COMMENT** column.

The comment is stored in a file on the PC via **SAVE TO FILE** and in the EEPROM of the sensor with **SAVE TO EE**. Memory space issues mean that only the first 8 characters are entered in EEPROM from the respective comment line.

A click on the ⏻ symbol resets the table.

The conversion value corresponding to the **SIG** value is shown in a display on the user interface in **GO** mode. It is also saved when recording data (see **RECORDER** tab).

NOTE: **SIG UNIT** is a parameter. It is stored in the **RAM** or **EEPROM** of the sensor together with the other parameters of the **PARA** and **TEACH** tabs.
The buttons **GET TABLE FROM FILE**, **SAVE TO FILE**, **GET TABLE FROM EE** and **SAVE TO EE** do not influence the parameter **SIG UNIT**.

2.6 Tab RECORDER

The SPECTRO-T-1-Scope software features a data recorder that allows the saving of **CH0, TEMP, SIG** and **SIG UNIT**. The recorded file is saved to the hard disk of the PC and can then be evaluated with a spreadsheet program.

The file that is created has four columns and as many rows as data frames were recorded.
A row is structured as follows: **Date**, **time**, **CH0, TEMP, SIG**, **SIG UNIT**.
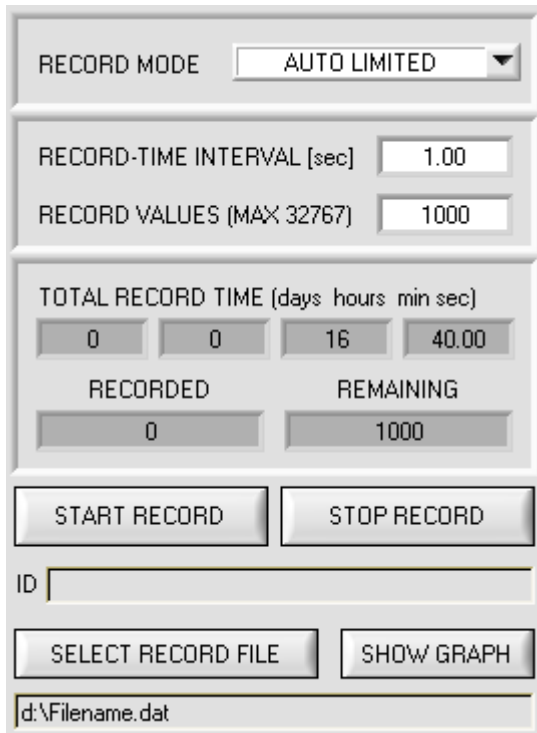
The following steps describe how data frames are recorded with the recorder:

Step 1:
When the **RECORDER** button is pressed, the following window will be displayed:
When the **SHOW GRAPH** button is pressed, a panel will be displayed that allows the user to monitor the different signals.
The individual signals can be activated from the **SIGNAL** drop-down menu.

RECORD MODE   AUTO LIMITED   ▼

RECORD-TIME INTERVAL [sec]   1.00

RECORD VALUES (MAX 32767)   1000

TOTAL RECORD TIME (days hours min sec)

| 0 | 0 | 16 | 40.00 |

RECORDED   REMAINING

| 0 | 1000 |

START RECORD   STOP RECORD

ID

SELECT RECORD FILE   SHOW GRAPH

d:\Filename.dat

Step 2:
If you want to automatically record several data frames, please select **AUTO LIMITED** under **RECORD MODE**.
Enter a time interval for recording under **RECORD-TIME INTERVAL [sec]**, in this example: 1, i.e. a new value is called from the sensor every second).
Then enter the maximum number of values you wish to record in the **RECORD VALUES [MAX 32767]** field.
Please note: Recording can also be stopped earlier by clicking **STOP RECORD**, the data recorded so far will not be lost.

The **TOTAL RECORD TIME** field indicates how long recording will take (in days, hours, minutes, and seconds) if all data are recorded.

Step 3:
By pressing the button **SELECT RECORD FILE** a file can be selected in which the data frame will be stored.
If you select an already existing file name, you will be asked whether you want to overwrite the existing file or not.

Step 4:
Pressing the **START RECORD** button starts automatic data recording.

The recorder starts to record data, and the button **START RECORD** is red to indicate that recording is active.
The respective data frames are shown in the display windows.

In the two display fields **RECORDED** and **REMAINING** you can check how many data frames have been recorded, and how many frames remain to be recorded.

Please note:
During recording the two input fields RECORD-TIME INTERVAL and VALUES TO BE RECORDED are inactive.

Step 5:
When as many data frames as set under **RECORD VALUES [MAX 32767]** have been recorded, or when the **STOP AUTO RECORD** button is pressed, a pop-up window will appear which confirms that the file is stored.

If you want to record an unlimited number of data, select the **AUTO UNLIMITED** function under **RECORD MODE**. Then select the desired recording interval and press **START RECORD**.

If you want to record data "manually", select the **MANUAL RECORDING** function under **RECORD MODE**.
You can start reading data from the sensor by pressing the **GO** button. These data are visualised in the display window. Pressing the **CAPTURE DATA FRAME** button saves a data frame in the file that was selected under **SELECT RECORD FILE**. The **RECORDED** field shows the sum of the frames already recorded.

If **RECORD MODE** = **AUTO TRIGGERED** is selected and a function that needs **IN1** is selected under **DIGITAL OUTMODE**, the sensor, when **START RECORD** is pressed, automatically sends a data frame after every drop of input **IN1**. This data frame is captured and recorded by the recorder.
STOP RECORD stops the automatic sending of the sensor.

Please note:
When you select a file with **SELECT RECORD FILE** this file is created as a new file. Then a file header with the meaning of the individual columns is written to the file.
When data are then recorded, these data are appended to the selected file, even when data recording is stopped and then resumed again.

2.7 Tab SCOPE

The SCOPE tab visualises an oscilloscope.

A click on **GET CYCLE TIME** displays the current sensor scan frequency in **[Hz]** and **[ms]**. The current scan frequency must be determined for the correct calculation of **deltaX[ms]**. Please give the sensor 8 seconds to determine the correct scan frequency before you click on **GET CYCLE TIME**.

In **TRIGGER MODE = SINGLE SHOT** a click on **SCAN** records a data frame and displays it in the graph.

In **TRIGGER MODE = FALLING EDGE** and **RISING EDGE** a click on **SCAN** starts triggered recording. A trigger start can be defined with **TRIGGER LEVEL**.

In **TRIGGER MODE= INTERN OUT0** recording starts stand-alone once OUT0 is HI.

In **TRIGGER MODE= EXTERN IN0** recording can be started external via input IN0.

SCAN-RATE can be used to delay or accelerate recording. This corresponds with the TIMEBASE function known in oscilloscopes. **PRE TRIGGER VALUES** can be used to define how many values should still be displayed before the actual trigger start.
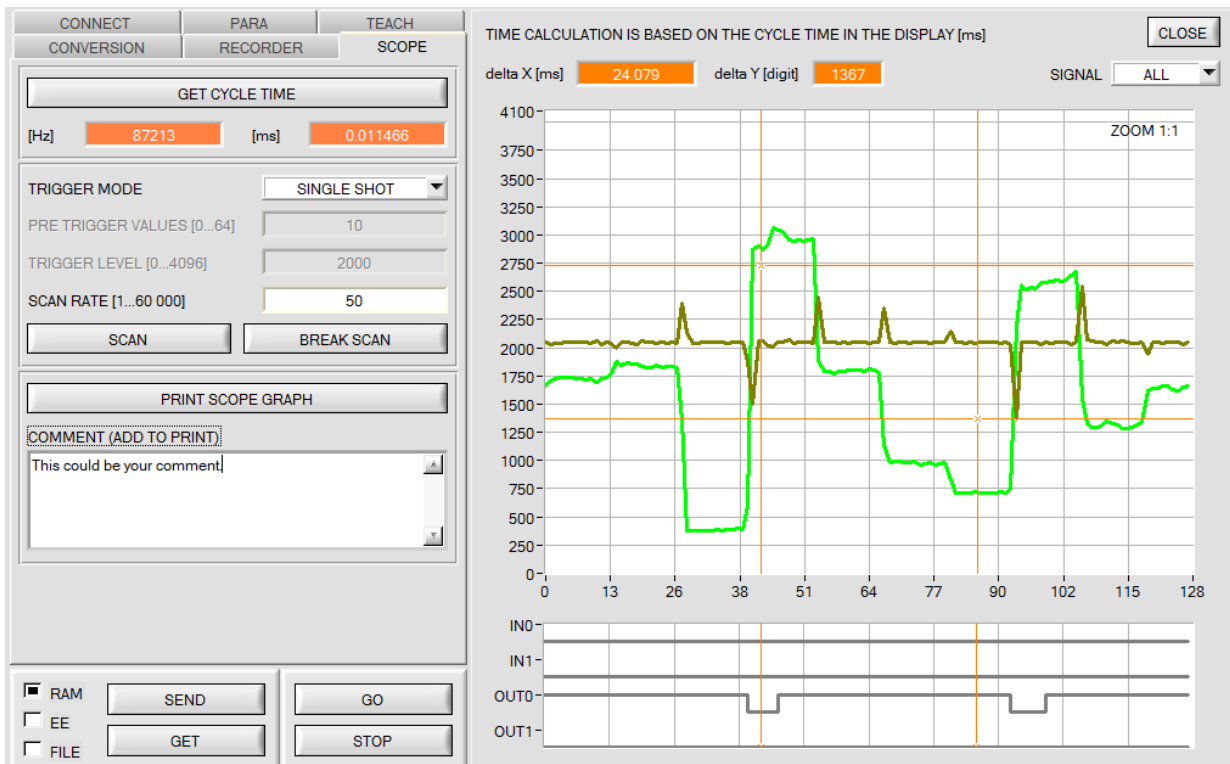
The zoom function in the graph can be activated by holding the control key (CTRL) and drawing a window with the mouse.

A click on **ZOOM 1:1** cancels the zoom function again.

The two orange cursors can be moved with the mouse. The displays **deltaX[ms]**, and **deltaY[digit]** will be updated.

deltaX[ms] shows the time between the cursors in X direction.

deltaY[digit] shows the difference between the two cursors in Y direction in digits or in Volt.

PRINT SCOPE GRAPH prints the current screen together with the text in the **COMMENT** text field.

The bottom graph represents the states of the two outputs **OUT0** and **OUT1** as well as of the inputs **IN0** and **IN1.**

2.8 Offset calibration

To avoid an increase of the electronic offset when using the integral function (**INTEGRAL** parameter), this offset can be eliminated by way of offset calibration or zero-point calibration. The corresponding tab is password-protected to prevent inadvertent incorrect settings.

Offset calibration can be accessed by double-clicking with the right mouse button exactly at the frame below **INTEGRAL** in the **PARA** tab.

e.g. here:
Double-click with the right mouse button.

You will then be prompted to enter a password.
The password is: mellon
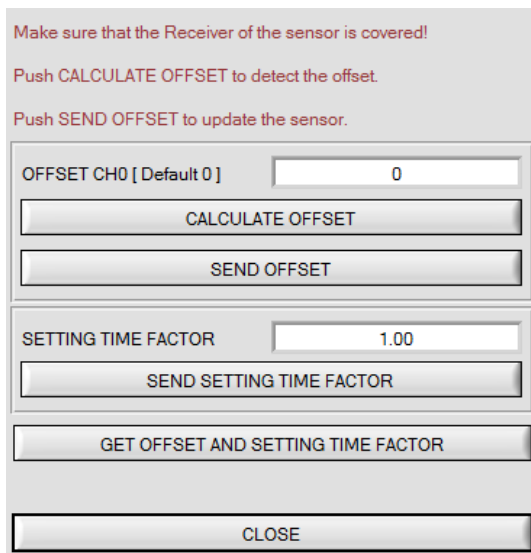
Then follow the instructions provided in the tab.

ATTENTION!
With offset calibration it is of utmost importance that the receiver does not see any extraneous light.
Therefore cover the receiver of the sensor, e.g. with a black cloth that is impermeable to light.

This is absolutely necessary for proper and perfect offset calibration.

Now press **CALCULATE OFFSET**. The offset value always should be clearly less than 100.
The value automatically is written to the EEPROM of the sensor.

GET OFFSET AND SETTING TIME FACTOR can be used to check the value that is saved as the offset value.
If necessary, **SEND OFFSET** can be used to save an offset value manually. (! not recommended !).

SETTING TIME FACTOR is a value that is defined when the sensor is manufactured.
For sensors with infrared light source this value is 2.5.
For all other sensors this value should be 1.0.
It is possible to change this value, but for safety reasons the procedure to change this value is not described in this manual.
If it should be necessary to change this value, please contact your supplier.

3   Operation of the TEMPCOMP-Scope software

If a firmware update should go wrong and the temperature characteristics that are stored in the EEPROM should be lost, these characteristics must be created anew. For this purpose you will need a file with the corresponding data. This file can be obtained from your supplier.

To perform temperature compensation please start the corresponding **TEMPCOMP-Scope software** (that can also be downloaded via the download link provided or made available via the DVD supplied, if applicable).

Please make sure that you have a functioning sensor connection. It may be necessary to select the connection with **CONNECT**. Set the correct sensor under **SELECT SENSOR**, if this is not done automatically.
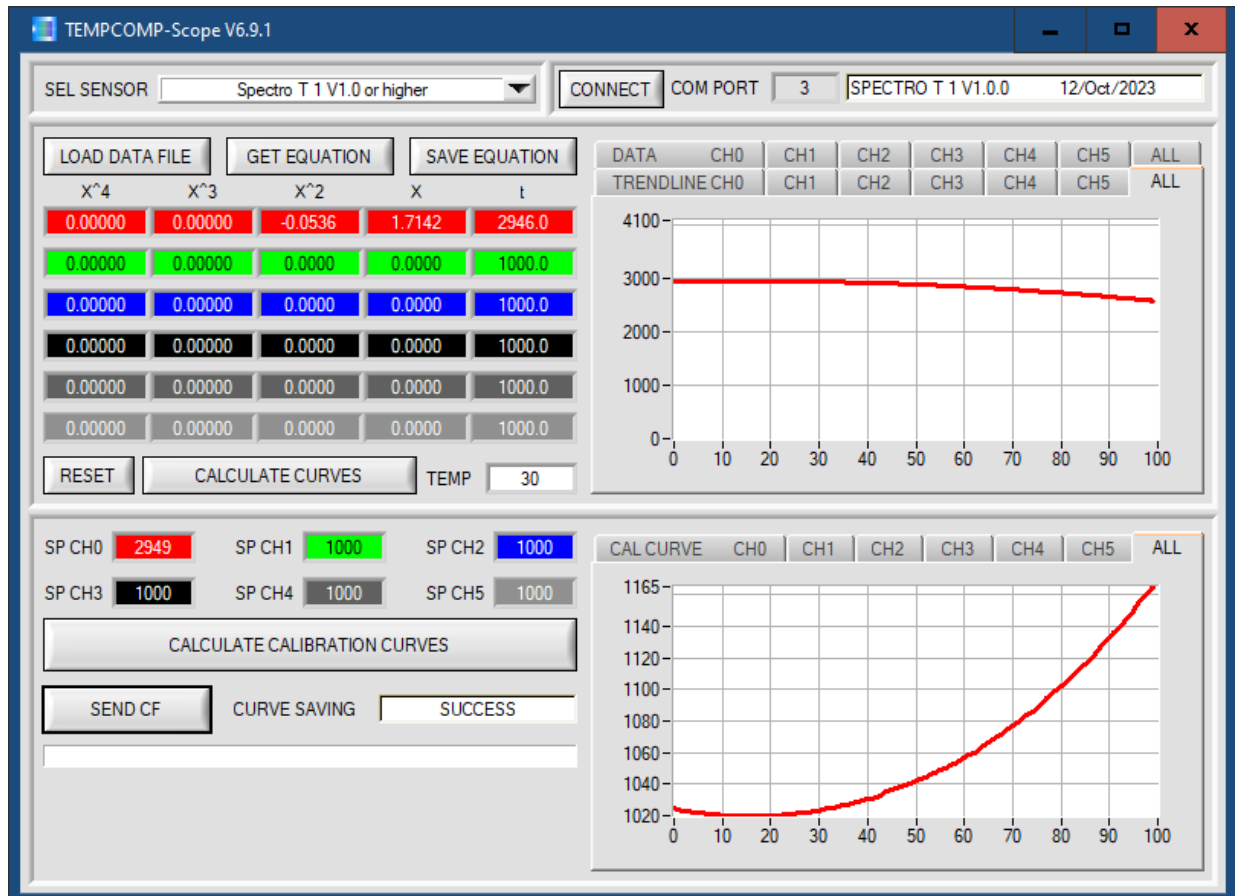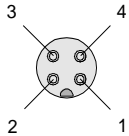
Step 1:   Load the temperature compensation file that you have received from your supplier with **GET EQUATION** or **LOAD DATA FILE**.

Step 2:   Press **CALCULATE CURVES** to display the data in the graph.

Step 3:   Select the sensor-internal operating temperature (not in °C) that the sensor has at an ambient temperature of 20°, if this has not already been done automatically. The value should be included in the file designation.

Step 4:   Press **CALCULATE CALIBRATION CURVES** to calculate the mean straight line.

Step 5:   Pressing the **SEND CF** button saves the mean straight lines in the **EEPROM** of the sensor.

Step 6:   Temperature compensation is successfully finished if the **SUCCESS** status message is then displayed.

Comment! If you do not immediately have the temperature compensation file at hand, simply start the **TempComp-Scope software**. Establish a connection, if it is not already established, and simply press **SEND-CF**. The sensor then functions as before, but it is not temperature-compensated.

4. Connector assignment of the SPECTRO-T-1 sensors
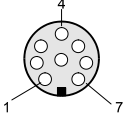
Connection of SPECTRO-T-1 to PC:

Pin No.:		Assignment:
4-pole M5 fem. connector (type Binder 707) *SPECTRO-T-1/PC-RS232*		
1		+24VDC (+Ub)
2		0V (GND)
3		Rx0
4		Tx0

Connecting cables to choose from:
cab-las4/PC-…
cab-4/USB-…
cab-4/ETH-…

Connection of SPECTRO-T-1 to PLC:

Pin No.:	Color of wire: (cab-las8/SPS)	Assignment:
8-pole fem. connector (type Binder 712) *SPECTRO-T-1/PLC*		
1	white	0V (GND)
2	brown	+24V (± 10 %)
3	green	IN0 (Digital 0: 0 … 1V, Digital 1: +Ub – 10%)
4	yellow	OUT0 (Digital 0: 0 … 1V, Digital 1: +Ub – 10%)
5	grey	OUT1 (Digital 0: 0 … 1V, Digital 1: +Ub – 10%)
6	pink or black	Not connected
7	blue	Not connected
8	red	IN1 (Digital 0: 0 … 1V, Digital 1: +Ub – 10%)

Connecting cable:
cab-las8/SPS-…

5   RS232 communication protocol

The sensors of the SPECTRO-T-1 series operate with the following **parameters** that are sent to the sensor or read from the sensor in the stated sequence.
Info! 2 bytes (8bit) are one word (16bit).

TABLE PARAMETER			
Parameter		**Type**	**Meaning**
Para1:	POWER [pm]	Word	Transmitter intensity (0 ... 1000) Attention intensity in thousendth!
Para2:	RECEIVER MODE	Word	Receiver Mode TRANSIMPEDANCE CONVERTER or INTEGATOR coded to (0,1)
Para3:	EXPOSURE TIME μs]	Word	Exposure Time in μs (1…65 000). Sensor calculates *10 →Resolution = 10μs
Para4:	LED MODE	Word	Control for the internal light source DC, AC, OFF coded to (0,1,2)
Para5:	GAIN	Word	Amplification of the integrated receiver AMP1…AMP16 coded to (1…16)
Para6:	AVERAGE	Word	Signal averaging 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384 or 32768
Para7:	INTEGRAL	Word	Signal integration (1…250)
Para8:	DIGITAL OUTMODE	Word	Function of the digital output: OFF, DIRECT, INVERSE, DIR RIS EDG of IN1, INV RIS EDG of IN1, DIR FAL EDG of IN1, INV FAL EDG of IN1 coded to (0,1,2,3,4,5,6)
Para9:	HOLD	Word	Hold time for minimum pulse length coded to (0…100 [ms]) send value*10
Para10:	THRESHOLD MODE	Word	Select Threshold Mode LOW, HI, WIN, 2 TRSH coded to (0,1,2,3)
Para11:	THRESHOLD TRACING	Word	Select Threshold Tracing OFF, ON TOL, ON CONT coded to (0,1,2)
Para12:	TT UP	Word	Time delay for Threshold Tracing Up (0…60000)
Para13:	TT DOWN	Word	Time delay for Threshold Tracing Down (0…60000)
Para14:	REF VAL CH0	Word	Reference Value for CH0 when OPERATING MODE = DELTA CH0 INTEGRATOR (0…4096)
Para15:	THRESHOLD CALC 1	Word	Threshold Calculation ABSOLUTE (digit), RELATIVE (%) coded to (0,1)
Para16:	TEACH VAL 1 SIG	Word	TEACH VAL 1 SIG (Reference) for threshold calculation (0…4095)
Para17:	TOLERANCE 1	Word	Tolerance Value for threshold calculation (0…4095)
Para18:	HYSTERESIS 1	Word	Hysteresis Value for threshold calculation (0…4095)
Para19:	THRESHOLD CALC 2	Word	Threshold Calculation ABSOLUTE (digit), RELATIVE (%) coded to (0,1)
Para20:	TEACH VAL 2 SIG	Word	TEACH VAL 2 SIG (Reference) for threshold calculation (0…4095)
Para21:	TOLERANCE 2	Word	Tolerance Value for threshold calculation (0…4095)
Para22:	HYSTERESIS 2	Word	Hysteresis Value for threshold calculation (0…4095)
Para23:	EXTERN TEACH	Word	External teach mode: OFF, DIRECT, DYN, MAX, MIN, (MAX-MIN)/2+MIN coded to (0,1,2,3,4,5)
Para24:	DEAD TIME	Word	Dead Time in [%] (0…100)
Para25:	OPERATING MODE	Word	Operating mode: NORMAL, DIFFERENTIATOR, DELTA CH0 INTEGRATROR coded to (0,1,2)
Para26:	SENSITIVITY	Word	Sensitivity of Differentiator (0…512)
Para27:	CHANNEL OFFSET	Word	Channel offset OFF, ON coded to (0,1)
Para28:	CH0 OFFSET	Wort	CH0 offset (0…4095)
Para29:	SIG UNIT	Word	SIG UNIT: mN/m, μm, g/m², mg/m², 10RFU, 100RFU, 1000RFU coded to (0,1,2,3,4,5,6)

Upon request, the data acquired and processed by the sensor are sent by the sensor in the following sequence.

TABLE DATA VALUE			
DATA VALUE		**Type**	**Meaning**
DatVal1:	CH0	Word	Analogue value CH0 of the receiver
DatVal2:	SIG	Word	Signal Value SIG
DatVal3:	REF1 SIG	Word	Reference value for threshold 1 calculation
DatVal4:	REF2 SIG	Word	Reference value for threshold 2 calculation
DatVal5:	TEMP	Word	Sensor internal temperature (not in °C or F)
DatVal6:	REF CH0	Word	Reference for CH0 when OPERATING MODE = DELTA CH0 INTEGRATOR
DatVal7:	DIGITAL OUT	Word	Bit 0 is 0 when signal is out of tolerance Bit 0 is 1 when signal is in tolerance Bit 1 is 0 when THRESHOLD MODE = LO or HI Bit 1 is 0 when THRESHOLD MODE = WIN and signal is under the window. Bit 1 is 1 when THRESHOLD MODE = WIN and signal is above window.
DatVal8:	DIGITAL IN	Word	Bit 0 is 0 when IN0 is LO Bit 0 is 1 when IN0 is HI Bit 1 is 0 when IN1 is LO Bit 1 is 1 when IN1 is HI
DatVal9:	MIN	Word	Minimum value of CH0 signal during IN0 was HI
DatVal10:	MAX	Word	Maximum value of CH0 signal during IN0 was HI
DatVal11:	SAT	Word	Saturation (SAT=0: no Saturation, SAT>0: Saturation of CH0)
DatVal12:	SIG UNIT	Word	SIG UNIT value (0…10000). Must be divided by 100.

Digital serial communication is used for the exchange of data between the software running on the PC and the sensor.
For this purpose the control unit features an EIA-232 compatible interface that operates with the (fixed) parameters "**8 data bits, 1 stop bit, no parity bit, no handshake**".
Five values are available for the baudrate: 9600baud, 19200baud, 38400baud, 57600baud and 115200baud. As an option the PC software also can communicate through TCP/IP or USB. In these cases transparent interface converters must be used that allow a connection to the RS232 interface.
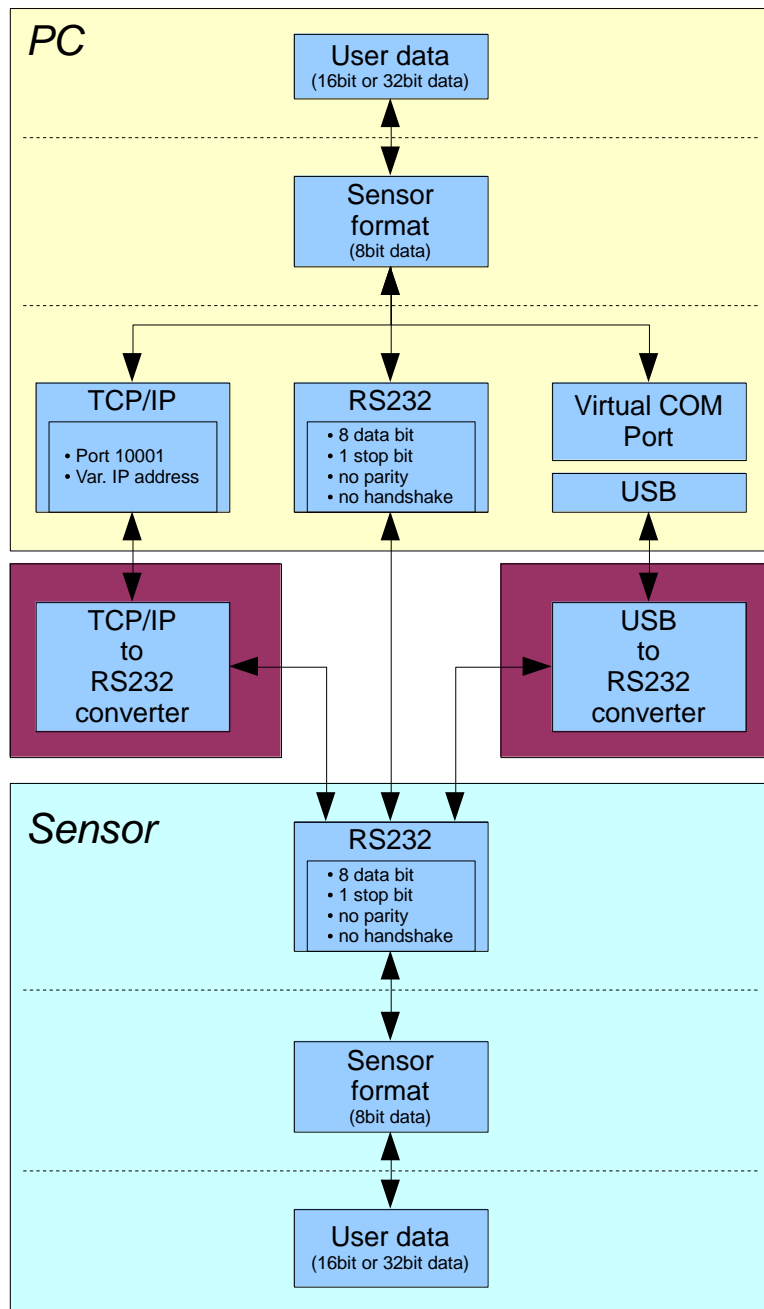
A proprietary protocol format that organises and bundles the desired data is used for all physical connection variants between PC software and control unit. Depending on their type and function the actual data are 16- or 32-bit variables and represent integer or floating-point values. The protocol format consists of 8-bit wide unsigned words ("bytes"). The actual data therefore sometimes must be distributed to several bytes.

The control unit always behaves passively (except if another behaviour has been specifically activated). Data exchange therefore always is initiated by the PC software. The PC sends a data package ("frame") corresponding to the protocol format, either with or without appended data, to which the control unit responds with a frame that matches the request.

The protocol format consists of two components:
A "header" and an optional appendant ("data").
The header always has the same structure.
The first byte is a synchronisation byte and always is 85_{dez} (55_{hex}).
The second byte is the so-called order byte. This byte determines the action that should be performed (send data, save data, etc.).
A 16-bit value (argument) follows as the third and fourth byte. Depending on the order, the argument is assigned a corresponding value.
The fifth and sixth byte again form a 16-bit value. This value states the number of appended data bytes. Without appended data both these bytes are 0_{dez} or 00_{hex}, the maximum number of bytes is 512.
The seventh byte contains the CRC8 checksum of all data bytes (data byte 1 up to and incl. data byte n).
The eight byte is the CRC8 checksum for the header and is formed from bytes 1 up to and incl. 7.
The header always has a total length of 8 bytes. The complete frame may contain between 8 and 520 bytes.

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header	Byte9 Data	Byte10 Data	…	Byte n+7 Data	Byte n+8 Data
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	Data1 (lo byte)	Data1 (hi byte)	…	Data n/2 (lo byte)	Data n/2 (hi byte)

The following **orders** can be sent to the sensor.

Number	ORDER (header byte no. 2)	Example
0	Sensor answers with order=0 if a communication error occurs. ARG=1: Invalid order number was sent to the sensor ARG=2: General communication error (wrong baudrate, overflow, …)	
1	Write parameter to the RAM of the sensor	order=1
2	Read parameter from the RAM of the sensor	order=2
3	Load parameter and actual Baudrate from RAM to EEPROM of the sensor	order=3
4	Load parameter from EEPROM to RAM of the sensor	order=4
5	Read CONNECTION OK and serial number from sensor	order=5
6	Free	
7	Read Firmware String and firmware number from sensor	order=7
8	Read data values from sensor	order=8
30	Start and Stop triggered sending of data frames	order=30
105	Get cycle time from sensor	order=105
190	Write new baud rate to the sensor	order=190

CRC8 Checksum
The so-called "Cyclic Redundancy Check" or CRC is used to verify data integrity. This algorithm makes it possible to detect individual bit errors, missing bytes, and faulty frames. For this purpose a value - the so-called checksum - is calculated over the data (bytes) to be checked and is transmitted together with the data package. Calculation is performed according to an exactly specified method based on a generator polynomial. The length of the checksum is 8 bit (= 1 byte). The generator polynomial is:

$$X^8+X^5+X^4+X^0$$

To verify the data after they have been received, CRC calculation is performed once again. If the sent and the newly calculated CRC values are identical, the data are without error.
The following pseudo code can be used for checksum calculation:

```
calcCRC8 (data[ ], table[ ])
Input:   data[ ], n data of unsigned 8bit
         table[ ], 256 table entries of unsigned 8bit
Output:  crc8, unsigned 8bit

crc8  := AA_hex
for I := 1 to n do
        idx := crc8 EXOR data[ i ]
        crc8 := table[ idx ]
endfor
return  crc8
```

table[ ]

0	94	188	226	97	63	221	131	194	156	126	32	163	253	31	65
157	195	33	127	252	162	64	30	95	1	227	189	62	96	130	220
35	125	159	193	66	28	254	160	225	191	93	3	128	222	60	98
190	224	2	92	223	129	99	61	124	34	192	158	29	67	161	255
70	24	250	164	39	121	155	197	132	218	56	102	229	187	89	7
219	133	103	57	186	228	6	88	25	71	165	251	120	38	196	154
101	59	217	135	4	90	184	230	167	249	27	69	198	152	122	36
248	166	68	26	153	199	37	123	58	100	134	216	91	5	231	185
140	210	48	110	237	179	81	15	78	16	242	172	47	113	147	205
17	79	173	243	112	46	204	146	211	141	111	49	178	236	14	80
175	241	19	77	206	144	114	44	109	51	209	143	12	82	176	238
50	108	142	208	83	13	239	177	240	174	76	18	145	207	45	115
202	148	118	40	171	245	23	73	8	86	180	234	105	55	213	139
87	9	235	181	54	104	138	212	149	203	41	119	244	170	72	22
233	183	85	11	136	214	52	106	43	117	151	201	74	20	246	168
116	42	200	150	21	75	169	247	182	232	10	84	215	137	107	53

Example order=1: Write parameter to the RAM of the sensor.

Example is with 5 parameter (Para1=500, Para2=0; Para3=3200, Para4=3300, Para5=1)
Have a look at the **TABLE PARAMETER** to check out how much parameter you have to send.

DATA FRAME PC → Sensor

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	\<order\>	\<ARG\> (lo byte)	\<ARG\> (hi byte)	\<LEN\> (lo byte)	\<LEN\> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	1	0	0	10	0	130	107
		ARG=0		LEN=10			

Byte9 Data	Byte10 Data	Byte11 Data	Byte12 Data	Byte13 Data	Byte14 Data	Byte15 Data	Byte16 Data	Byte17 Data	Byte18 Data
Para1 (lo byte)	Para1 (hi byte)	Para2 (lo byte)	Para2 (hi byte)	Para3 (lo byte)	Para3 (hi byte)	Para4 (lo byte)	Para4 (hi byte)	Para5 (lo byte)	Para5 (hi byte)
244	1	0	0	128	12	228	12	1	0
Para1=500		Para2=0		Para3=3200		Para4=3300		Para5=1	

DATA FRAME Sensor → PC

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	\<order\>	\<ARG\> (lo byte)	\<ARG\> (hi byte)	\<LEN\> (lo byte)	\<LEN\> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	1	0	0	0	0	170	224
		ARG=0		LEN=0			

If you receive an argument greater 0, ARG parameter where out of range and have been set to a default value.

Example order=2: Read parameter from the RAM of the sensor.

Example is with 5 parameter (Para1=500, Para2=0; Para3=3200, Para4=3300, Para5=1)
Have a look at the **TABLE PARAMETER** to check out how much parameter you will receive.

DATA FRAME PC → Sensor

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	2	0	0	0	0	170	185
		ARG=0		LEN=0			

DATA FRAME Sensor → PC

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	2	0	0	10	0	130	50
		ARG=0		LEN=10			

Byte9 Data	Byte10 Data	Byte11 Data	Byte12 Data	Byte13 Data	Byte14 Data	Byte15 Data	Byte16 Data	Byte17 Data	Byte18 Data
Para1 (lo byte)	Para1 (hi byte)	Para2 (lo byte)	Para2 (hi byte)	Para3 (lo byte)	Para3 (hi byte)	Para4 (lo byte)	Para4 (hi byte)	Para5 (lo byte)	Para5 (hi byte)
244	1	0	0	128	12	228	12	1	0
Para1=500		Para2=0		Para3=3200		Para4=3300		Para5=1	

Example order=3: Load parameter and actual Baudrate from RAM to EEPROM of the sensor.

DATA FRAME PC → Sensor

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	3	0	0	0	0	170	142
		ARG=0		LEN=0			

DATA FRAME Sensor → PC

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	3	0	0	0	0	170	142
		ARG=0		LEN=0			

Example order=4: Load parameter from EEPROM to RAM of the sensor.

DATA FRAME PC → Sensor

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	4	0	0	0	0	170	11
		ARG=0		LEN=0			

DATA FRAME Sensor → PC

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	4	0	0	0	0	170	11
		ARG=0		LEN=0			

Example order=5: Read CONNECTION OK from sensor.

DATA FRAME PC → Sensor

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	5	0	0	0	0	170	60
		ARG=0		LEN=0			

DATA FRAME Sensor → PC
ARG determines the serial number of the sensor

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	5	170	0	0	0	170	178
		ARG=170		LEN=0			

Example order=7: Read Firmware String from sensor

DATA FRAME PC → Sensor

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	\<order>	\<ARG> (lo byte)	\<ARG> (hi byte)	\<LEN> (lo byte)	\<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	7	0	0	0	0	170	82
		ARG=0		LEN=0			

DATA FRAME Sensor → PC
ARG determines the firmware number of the sensor

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header	Byte9 Data	Byte10 Data	Byte11 Data	Byte12 Data
0x55	\<order>	\<ARG> (lo byte)	\<ARG> (hi byte)	\<LEN> (lo byte)	\<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	ASCII	ASCII	ASCII	ASCII
85 (dec)	7	0	0	72	0	183	38	F	I	R	M
		ARG=0		LEN=72							

Byte13 Data	Byte14 Data	Byte15 Data	Byte16 Data	Byte17 Data	Byte18 Data	Byte19 Data	Byte20 Data	Byte21 Data	Byte22 Data	Byte23 Data	Byte24 Data
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
W	A	R	E		S	T	R	I	N	G	

Byte25 Data	Byte26 Data	Byte27 Data	Byte28 Data	Byte29 Data	Byte30 Data	Byte31 Data	Byte32 Data	Byte33 Data	Byte34 Data	Byte35 Data	Byte36 Data
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
											R

Byte37 Data	Byte38 Data	Byte39 Data	Byte40 Data	Byte41 Data	Byte42 Data	Byte43 Data	Byte44 Data	Byte45 Data	Byte46 Data	Byte47 Data	Byte48 Data
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
T	:	K	W	x	x	/	x	x			

Byte49 Data	Byte50 Data	Byte51 Data	Byte52 Data	Byte53 Data	Byte54 Data	Byte55 Data	Byte56 Data	Byte57 Data	Byte58 Data	Byte59 Data	Byte60 Data
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII

Byte61 Data	Byte62 Data	Byte63 Data	Byte64 Data	Byte65 Data	Byte66 Data	Byte67 Data	Byte68 Data	Byte69 Data	Byte70 Data	Byte71 Data	Byte72 Data
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII

Byte73 Data	Byte74 Data	Byte75 Data	Byte76 Data	Byte77 Data	Byte78 Data	Byte79 Data	Byte80 Data	Byte81 Data	Byte82 Data
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII

Example order=8: Read data values from sensor.

Example is with 5 data values (DataVal1=2000, DataVal2=4; DataVal3=3000, DataVal4=3500, DataVal5=18)
Have a look at the **TABLE DATA VALUE** to check out how much data values you will receive.

DATA FRAME PC → Sensor

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	8	0	0	0	0	170	118
		ARG=0		LEN=0			

DATA FRAME Sensor → PC

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	8	0	0	10	0	28	243
		ARG=0		LEN=10			

Byte9 Data	Byte10 Data	Byte11 Data	Byte12 Data	Byte13 Data	Byte14 Data	Byte15 Data	Byte16 Data	Byte17 Data	Byte18 Data
DataVal1 (lo byte)	DataVal1 (hi byte)	DataVal2 (lo byte)	DataVal2 (hi byte)	DataVal3 (lo byte)	DataVal3 (hi byte)	DataVal4 (lo byte)	DataVal4 (hi byte)	DataVal5 (lo byte)	DataVal5 (hi byte)
208	7	4	0	184	11	172	13	18	0
DatVal1 = 2000		DatVal2 = 4		DatVal3 = 3000		DatVal4 = 3500		DatVal5 = 18	

Example order=30: Start and Stop triggered sending of data frames

Start triggered sending of data frames
DATA FRAME PC → Sensor
ARG = 1 --> starts triggered sending

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	30	1	0	0	0	170	82
		ARG=1		LEN=0			

DATA FRAME Sensor → PC

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	30	1	0	0	0	170	82
		ARG=1		LEN=0			

Stop triggered sending of data frames
DATA FRAME PC → Sensor
ARG = 0 --> stops triggered sending.

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	30	0	0	0	0	170	159
		ARG=0		LEN=0			

DATA FRAME Sensor → PC

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	30	0	0	0	0	170	159
		ARG=0		LEN=0			

Example order=105: Get cycle time from sensor

DATA FRAME PC → Sensor

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header
0x55	\<order\>	\<ARG\> (lo byte)	\<ARG\> (hi byte)	\<LEN\> (lo byte)	\<LEN\> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	105	0	0	0	0	170	130
		ARG=0		LEN=0			

DATA FRAME Sensor → PC

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Data	Byte9 Data	Byte10 Data	Byte11 Data
0x55	\<order\>	\<ARG\> (lo byte)	\<ARG\> (hi byte)	\<LEN\> (lo byte)	\<LEN\> (hi byte)	CRC8 (Data)	CRC8 (Header)	lo word lo byte	lo word hi byte	hi word lo byte	hi word hi byte
85 (dec)	105	0	0	8	0	82	17	23	140	8	0
		ARG=0		LEN=8				CYCLE COUNT = 560151			

Byte12 Data	Byte13 Data	Byte14 Data	Byte15 Data
lo word lo byte	lo word hi byte	hi word lo byte	hi word hi byte
64	156	0	0
COUNTER TIME = 40000			

Cycle Time [Hz] = CYCLE COUNT / (COUNTER TIME * 0,0001)

Cycle Time [ms] = (COUNTER TIME * 0,01) / CYCLE COUNT

Example order=190: Write new baud rate to the sensor.

DATA FRAME PC → Sensor

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	\<order\>	\<ARG\> (lo byte)	\<ARG\> (hi byte)	\<LEN\> (lo byte)	\<LEN\> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	190	1	0	0	0	170	14
		ARG=1		LEN=0			

New baud rate is determined by argument.
ARG=0: baud rate = 9600
ARG=1: baud rate = 19200
ARG=2: baud rate = 38400
ARG=3: baud rate = 57600
ARG=4: baud rate = 115200

DATA FRAME Sensor → PC

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header
0x55	\<order\>	\<ARG\> (lo byte)	\<ARG\> (hi byte)	\<LEN\> (lo byte)	\<LEN\> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	190	0	0	0	0	170	195
		ARG=0		LEN=0			

A Firmware update via software Firmware Loader

A.1 Manual software Firmware Loader V1.1

This manual describes the installation of the Firmware Loader PC software and as a support for starting up the Firmware Loader software explains the individual functional elements of the graphic Windows® user interface.

The software allows the user to perform an automatic firmware update. The update will be carried out through the RS232 interface.

An initialisation file (xxx.ini) and a firmware file (xxx.elf.S) are required for performing a firmware update. These files can be obtained from your supplier. In some cases an additional firmware file for the program memory (xxx.elf.p.S) is also needed, and this file will be automatically provided together with the other two files.

Important! For a firmware update the two or three files must always be located in the same folder.

A plausibility check is performed after the initialisation file has been loaded with the Firmware Loader. If the initialisation file was changed or damaged, it will not be possible to perform a firmware update.

When the plausibility check is successfully completed, the instructions contained in the initialisation file will be carried out step by step.

The complete memory contents of the micro-controller in the sensor will be deleted in a firmware update. This means that both the program in the program memory and the data in the data memory will be lost.

The new firmware automatically writes the correct data to the program memory again.

However, the parameter settings, temperature curves, linearization curves, etc. that are stored in the data memory (EEPROM) will be deleted.

With the Firmware Loader V1.1 the data will be saved in the EEPROM, and can be written back again after successful firmware update.

For this purpose the software creates an EEPROM backup file.

A.2 Installation of the software Firmware Loader V1.1

Hardware requirements for successful installation of the Firmware Loader software:

- Microsoft® Windows® 7, 8, 10
- IBM PC AT or compatible
- VGA graphics
- Microsoft® compatible mouse
- Serial RS232 interface at the PC or USB slot or RJ45 connector
- Cable **cab-las4/PC** for the RS232 interface or **cab-4/USB** USB converter or **cab-4/ETH** Ethernet converter

Please install the software as described below:

1. You can download the software via a provided download link or, if applicable, install it via the provided software DVD. To install the software, start the 'SETUP' program in the 'SOFTWARE' folder.

2. The installation program displays a dialog and suggests to install the software in the C:\"FILENAME" directory on the hard disk. You may accept this suggestion with **OK** or [**ENTER**], or you may change the path as desired. Installation is then performed automatically.

3. During the installation process a new program group for the software is created in the Windows Program Manager. In the program group an icon for starting the software is created automatically. When installation is successfully completed the installation program displays "Setup OK".

4. After successful installation the software can be started with a left mouse button double-click on the icon.

Windows® is a trademark of the Microsoft Corp.
VGA™ is a trademark of the International Business Machines Corp.

Please read this chapter before you start!

In this example a software update is performed from SPECTRO3 V4.0 to SPECTRO3 V4.1.

Step 1:

When the Firmware Loader software is started, this window opens on the Windows® user interface.

Immediately after starting, the software attempts to establish a connection to the connected sensor. If the sensor should not be connected at **COM PORT 1,** please select the corresponding **COM PORT.**
Please make sure that the correct **BAUDRATE** is selected.
Now try to establish a connection by clicking on **TRY TO CONNECT**. When the connection has been established, the sensor sends back information about the current firmware.
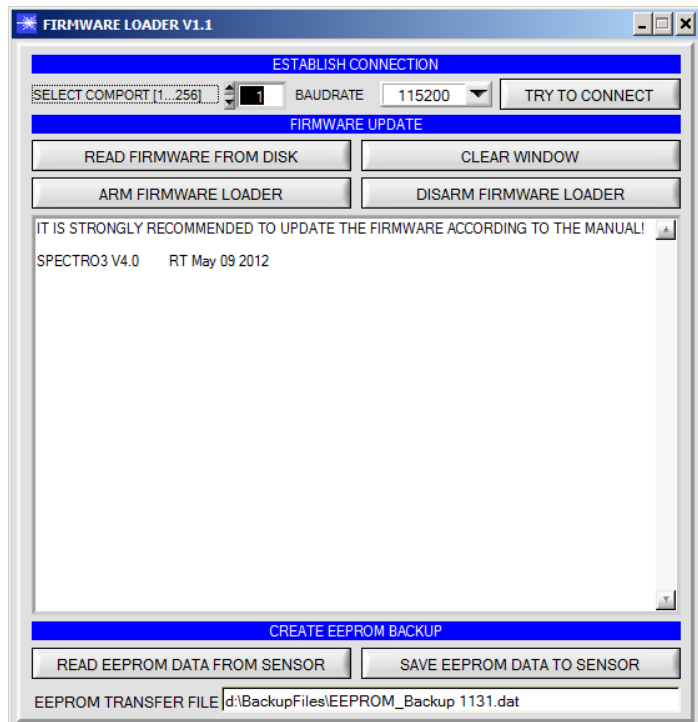

Step 2:
Press the **READ FIRMWARE FROM DISK** button and load the **xxx.ini** file.

The uploaded initialization file will be displayed in the status window.
As described above, a plausibility check of the initialisation file will be performed first.
If the file is OK, the following message will be displayed:
File read OK!
Press **ARM FIRMWARE LOADER** to start the firmware update.

Please read the comments that are shown in the display window. These comments allow you to make sure that you have loaded the correct initialisation file.
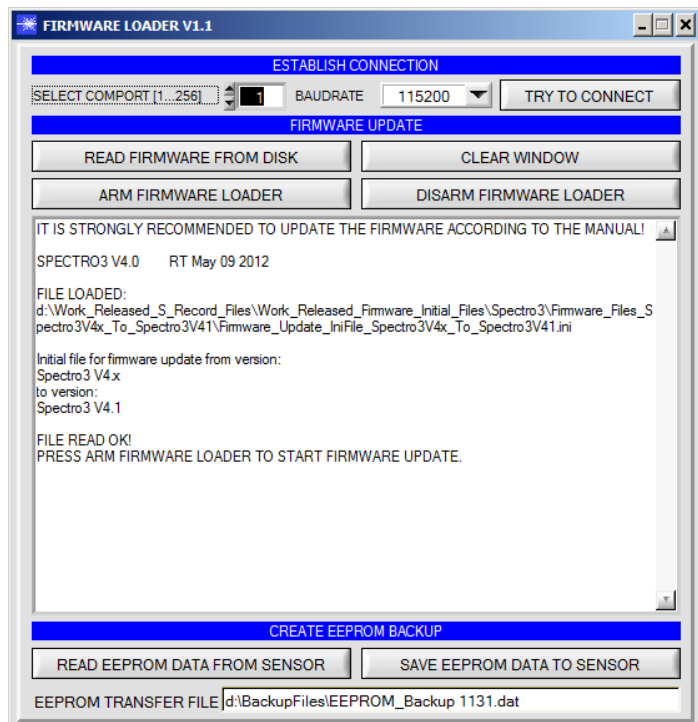
Step 3:
Now click on the **ARM FIRMWARE LOADER** button. The program now attempts to send a software command that interrupts the normal program run and jumps to the start address of the boot sector.
If this is successful, the sensor displays a prompt for loading the S-Record file to the sensor.

When you press the **ARM FIRMWARE LOADER** button the firmware update will be performed automatically.

In the course of the update process you will only be prompted to enter a name for the EEPROM backup file. If the firmware update should run perfectly until the EEPRM data are read out, but should then go wrong for any reason whatsoever, the EEPROM backup file can always be written back with **SAVE EEPROM DATA TO SENSOR**.

The file name for your **EEPROM backup file** should be chosen such that the names for several sensors cannot be mixed up. Using a file name that contains the sensor serial number might be advisable. Saving this file for future updates also might be a good idea.

After a successful update the sensor displays the status line of the new firmware.

The complete update process may take up to 1 minute.
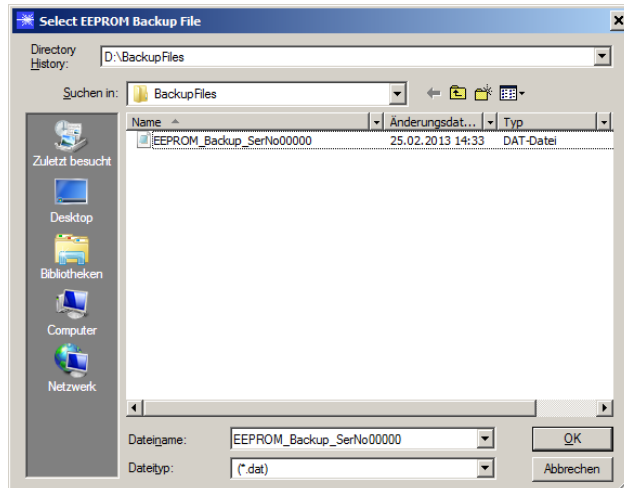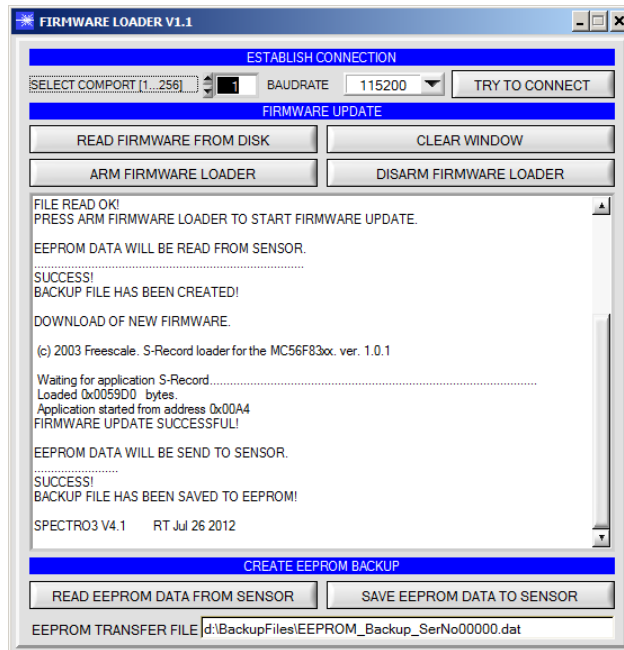
If, contrary to expectations, there should be any trouble with the update of the program memory, it will still be possible to perform an update, even though it may look like the sensor was "killed".

Please make sure that you have selected the correct **COM PORT** and the correct **BAUDRATE**.

You will not get any connection when you click on **TRY TO CONNECT**.

Load the corresponding **xxx.ini** file from the hard disk.

Then click on the **ARM FIRMWARE LOADER** button.

The program will try to send the software command for the update. This will not work, however, and you will get a **CONNECTION FAILURE** message.

However, the Firmware Loader software now is "armed" for 30 seconds.

If you perform a hardware reset within these 30 seconds, the firmware update will be performed.

After a successful update the sensor displays the status line of the new firmware.

The complete update process may take up to 1 minute.

INFO! In case that the sensor was "killed", the sensor will work with a BAUDRATE of 115200.

You may at any time create an EEPROM backup file for archiving it on your hard disk.

To do this, click on **READ EEPROM DATA FROM SENSOR.** You will be prompted to chose an initialization file in case that there has not yet been loaded any. Afterwards you will be asked to enter a file name. The selected name will be shown in the **EEPROM TRANSFER FILE** display.

The file name for your **EEPROM backup file** should be chosen such that the names for several sensors cannot be mixed up. Using a file name that contains the sensor serial number might be advisable.

The Firmware Loader then reads all the EEPROM data from the data memory and saves these data in the selected file.

Upon successful completion the following message will be displayed:
Success!
Backup File has been created!

If something should go wrong in a firmware update, any you have created the **backup file**, the saved **EEPROM backup file** can at any time be uploaded to the sensor again with **SAVE EEPROM DATA TO SENSOR.**
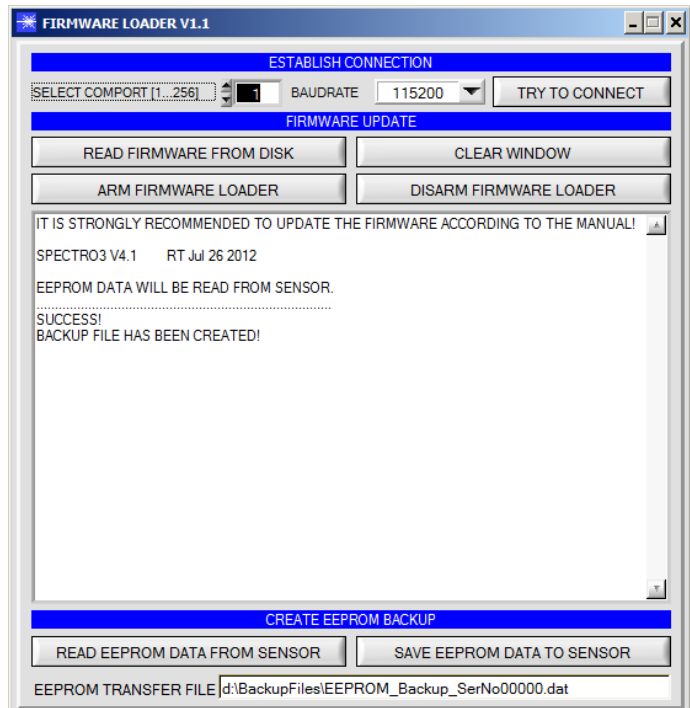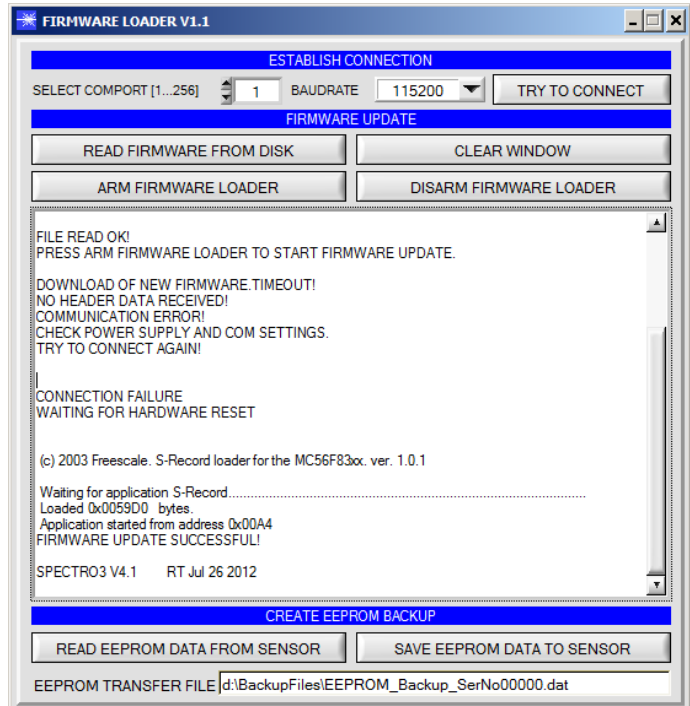
CLEAR WINDOW resets the display window.

If you should not get any response for a longer time, or if messages should be displayed in the status line, **DISARM FIRMWARE LOADER** can be used to cancel the firmware update process.
However, you should always wait for approx. 1 minute before you press this button.

FIRMWARE LOADER V1.1

ESTABLISH CONNECTION

SELECT COMPORT [1...256]   1   BAUDRATE   115200   ▼   TRY TO CONNECT

FIRMWARE UPDATE

| READ FIRMWARE FROM DISK | CLEAR WINDOW |
| ARM FIRMWARE LOADER | DISARM FIRMWARE LOADER |

IT IS STRONGLY RECOMMENDED TO UPDATE THE FIRMWARE ACCORDING TO THE MANUAL!

CREATE EEPROM BACKUP

| READ EEPROM DATA FROM SENSOR | SAVE EEPROM DATA TO SENSOR |

EEPROM TRANSFER FILE d:\BackupFiles\EEPROM_Backup_SerNo00000.dat