Manual
Software SPECTRO-T-4-Scope V1.0

(PC software for Microsoft® Windows® 10)

for sensors of SPECTRO-T-4 Series

This manual describes the installation of the PC software for the SPECTRO-T-4 sensor. As a support for commissioning of the sensor this manual explains the functional elements of the Windows® user interface.

The SPECTRO-T-4 sensors were developed to identify different plastic granulates and flakes as well as textiles made of natural and synthetic fibers. The determination takes place in the visible range (VIS) and in the near infrared range (NIR).

For this purpose, the sensor has a white light source in the VIS range and an integrated triple receiver for the red, green and blue proportion. This provides 3 signals (X, Y, Z) in the VIS range.
In the NIR range, the sensor has 3 light sources with different wavelengths and a broadband photodiode which records the individual wavelengths analogously. Here you also obtain 3 signals (NIR1, NIR2, NIR3).
The 4 light sources are switched on one after the other, so that the signals do not overlap.
With the recorded signal values X, Y, Z or NIR1, NIR2, NIR3, the 3 color space coordinates L*a*b* for the visible range and the 3 NIR space coordinates N*i*r* for the near-infrared range are calculated. These coordinates each represent a point in space and uniquely characterize the surface.

Signal detection with the SPECTRO-T-4 is very flexible. The stepless adjustment possibility of the integrated light sources as well as a selectable amplification of the receiver signal and an INTEGRAL function enable the sensor to be adjusted to almost any surface.

Sensors in the SPECTRO-T-4 series have no analog or digital inputs or outputs.
Data is exchanged via the sensor's internal RS232 interface.
Parameters and measurement values can be exchanged between a PC and the SPECTRO-T-4 sensor through the serial RS232 interface. All the parameters can be saved to the non-volatile EEPROM of the SPECTRO-T-4 sensor through the serial RS232 interface. When parameterisation is finished, the sensor continues to operate with the current parameters in STAND-ALONE mode without a PC.

Possible firmware updates can be easily performed through the RS232 interface, even with the sensor system in installed condition. (→ cf. Annex „Firmware update via software Firmware Loader“).

The measuring sensors of the SPECTRO-T-4 series are calibrated at the factory. Using the calibration function, the user can calibrate independently with the calibration cards available from the sensor supplier under the menu item CALIB.

As usual, the sensor data are visualized on the PC surface.

The current space coordinates as well as their differences to their set values are also displayed.

The space coordinates as well as their deltas and two freely adjustable documentation entries can then be output to a label printer, for example. The data can also be saved in a file which can then be reloaded for viewing.

We recommend the following label printer and accessories:

Printer	Accessories	Accessories
Company: cab Produkttechnik GmbH & Co. KG Model: cab EOS2/300 Item No. 5978202	Adhesive roll label Item No. 5780036 Material: TF Satin White Width x Height: 50x30mm	Thermal transfer ribbon Item No. 5556661

You can download drivers, documentation, additional software, etc. at the following link:
https://www.cab.de/en/marking/label-printer/eos25/?ref=nav_lang
The installation of the software for the label printer is described in this manual.

Label printer and accessories can also be purchased from our company.
We have no experience with other printers and therefore cannot provide support.

Contents

Shortcuts:

SEND	F9
GET	F10
GO	F11
STOP	F12

1. Installation of the SPECTRO-T-4-Scope software

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

- Microsoft® Windows® 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 'Install' 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.

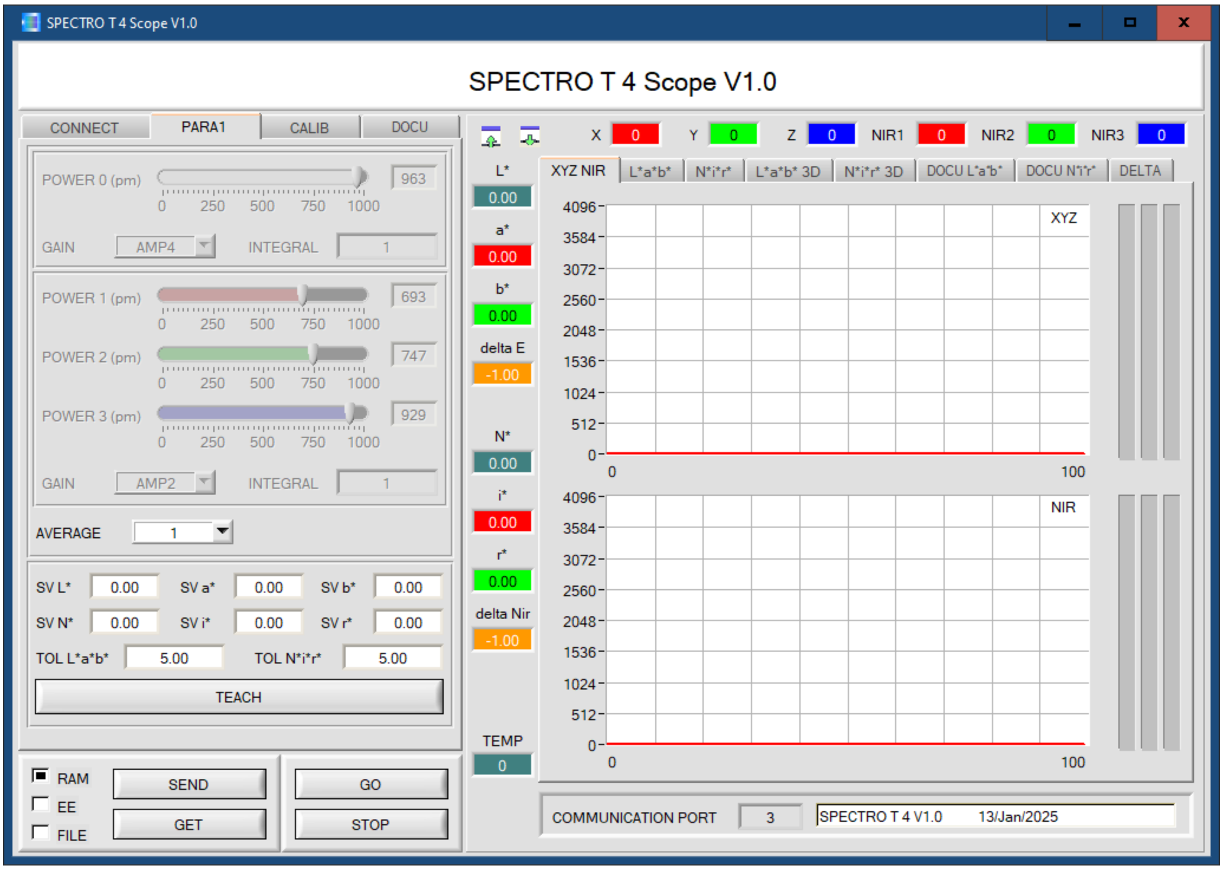
3. During the installation process a new program group for the software is created in the Windows Program Manager. In the program group an icon for starting the software is created automatically. When installation is successfully completed the installation program displays "Setup OK".

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

2. Operation of the SPECTRO-T-4-Scope software

Please read this chapter first before you start to adjust and parameterise the SPECTRO-T-4 sensor.

When the SPECTRO-T-4-Scope software is started, the following window appears on the Windows interface:



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.

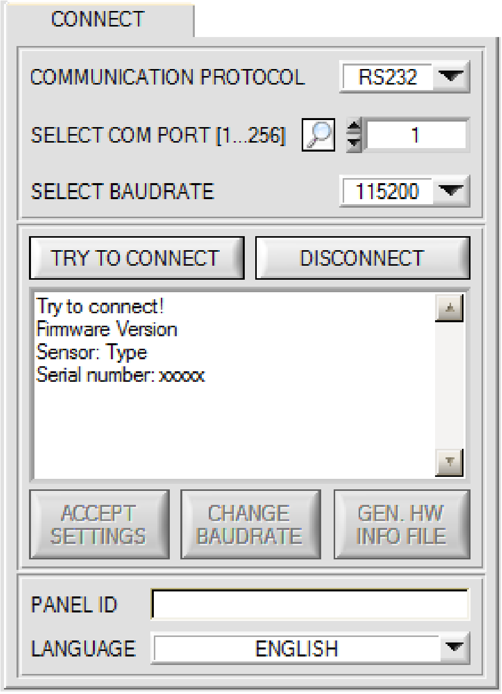
If a connection is not established automatically, e.g. if no sensor is connected, the software can be run in OFFLINE mode. In offline mode it only is possible to exchange parameters with a file on a storage medium, which often is helpful for the purpose of analysing parameter files.

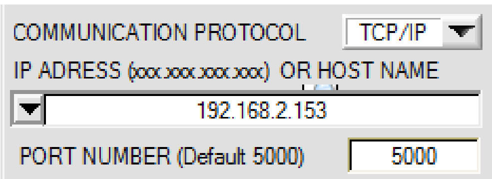
If a sensor is connected and a connection still cannot be established, either the SCOPE version (program at the PC) and the firmware version (program in the sensor) do not match, or the interface to the sensor must be correctly configured.
If different Scope and firmware versions should be the problem, 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 invisible depending on the parameterization.

2.1 Tab CONNECT (Connection setup)





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.

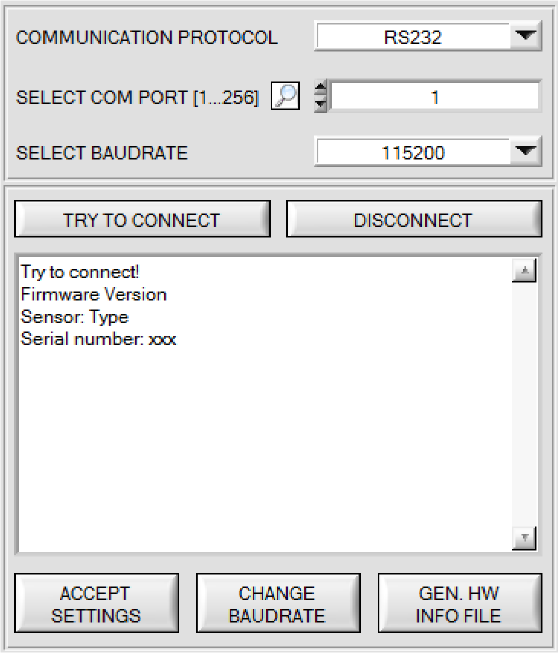
An RS232 to Ethernet converter **(cab-4/ETH)** is needed if the sensor should communicate through a local network. With this converter a connection to the sensor can be established using the **TCP/IP** protocol.
Parameterisation of the **cab-4/ETH** converter (assigning of IP address, baudrate setting,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

In order to establish a connection to 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.

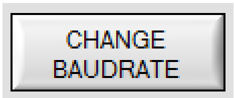
Under **PANEL ID** a name can be entered that will be displayed at different points in the program window, and that will be recorded in different files (e.g. Record File) as well. With the input field **LANGUAGE** a language can be set with which the individual controls are displayed on the surface.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

Please note: **ATTENTION !**	**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.** **For maintaining maximum switching frequency at the sensor data communication with the PC must be stopped (press the STOP button).**

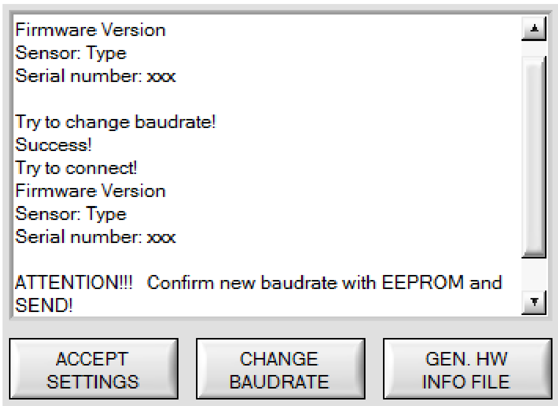


The baudrate for data transfer through the RS232 interface can be set by means of the **SELECT BAUDRATE** drop down menu and **CHANGE BAUDRATE** function field.

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.

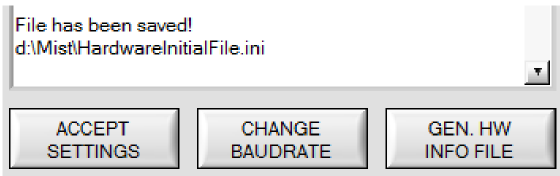


Now a new baudrate can be selected under **SELECT BAUDRATE**.
A click on **CHANGE BAUDRATE** sends the new baudrate information to the sensor.



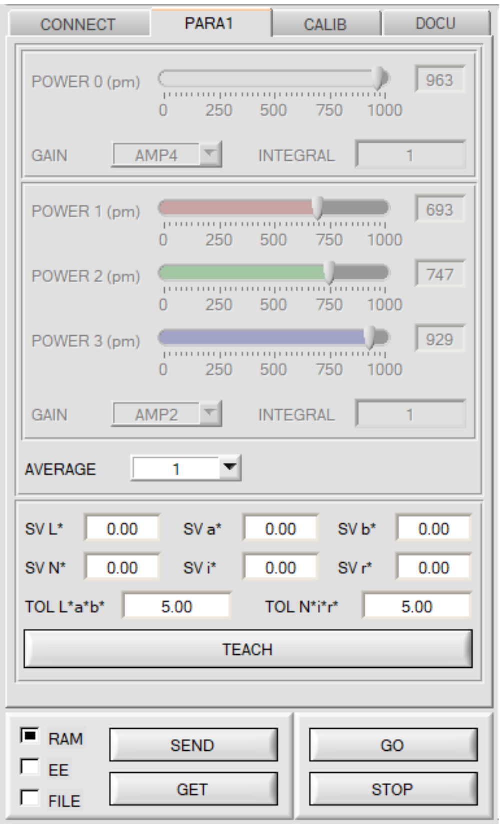
When the new baudrate information has been successfully sent, the sensor operates with the new baudrate. A window will pop up, prompting you to select **EEPROM** and then to press **SEND**. After a hardware reset the new baudrate only will be used when **EEPROM** and **SEND** have been pressed.

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 PARA1, button SEND, GET, GO, STOP (parameterization, data exchange)



PARA1:
Pressing the **PARA1** 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**).

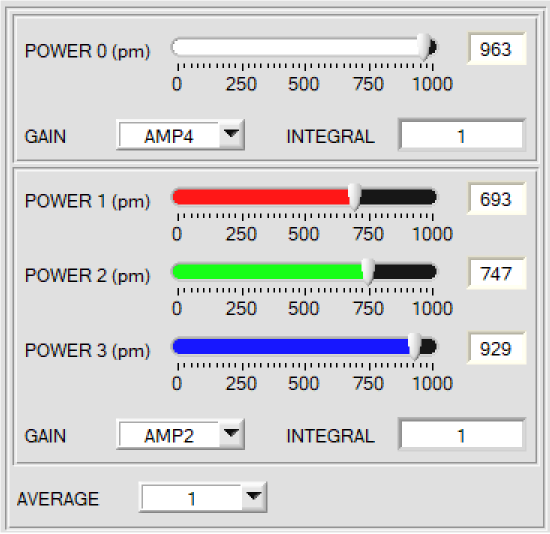
RAM:
The **RAM** is a **volatile** memory in the sensor's micro-controller, i.e. when the power at the sensor is turned off, these parameters will be lost again.
The sensor always operates with the parameters in its RAM.
If the **RAM** option is selected, a click on **SEND** writes the current parameters 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! A parameter setting that has been defined for a specific application should always be stored 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.



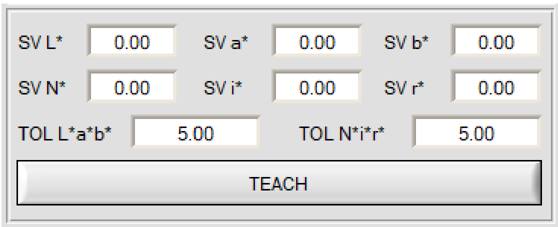
POWER:
The transmitter power of the 4 transmitters is constantly according to the value set on the respective **POWER [pm]** slider, when the light source is switched on for measuring. The **POWER** can be set with the slider or by entering a value in the edit-box. A value of 1000 means full intensity at the transmitter unit, a value of 0 sets the lowest intensity at the transmitter.
INFO: POWER only is available with **CALIB=OFF**.

GAIN:
This item is used for setting the gain. 8 different gain stages can be set (AMP1 to AMP8). **GAIN** should be set such that with a medium **POWER** value the sensor operates in its dynamic range (X, Y, Z, NIR1, NIR2, NIR3 between 2750 and 3750).
INFO: GAIN only is available with **CALIB=OFF** or **XYZ OFFSET.**

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** default value increases the noise of the raw signals of the receiver unit, and simultaneously decreases the maximum achievable switching frequency of the sensor.
INFO: INTEGRAL only is available with **CALIB=OFF.**

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.

INFO: **POWER 0** is used to set the power of the white light transmission LED for the **VIS range**. **GAIN** and **INTEGRAL** in this framed block only affect the visible range for detecting the **X, Y, Z** signals.
The same applies in the **NIR range** for **POWER 1**, **POWER 2** and **POWER 3** for the detection of **NIR1**, **NIR2** and **NIR3**. After **X, Y, Z** have been detected, the 3 transmitters with different wavelengths are switched on one after the other and supply the signals **NIR1**, **NIR2** and **NIR3.**



SETVALUES and TOL:
A set value and a tolerance for **L*a*b*** and **N*i*r*** can be specified in these input fields by entering these directly or by pressing **TEACH**.
These set values and tolerances are required at various points in the program to indicate how far the currently measured values deviate from the target values.
E.g. **DOCU** tab or **L*a*b* 3D** etc.

2.3 Calculation formulae and display elements

The spatial coordinates **L*a*b*** for the visible **(VIS)** range and **N*i*r*** for the near infrared **(NIR)** range are calculated from the recorded signals **X, Y, Z** and **NIR1, NIR2, NIR3** using the following formulae.

L*a*b*:
The L*a*b* color space (also referred to as the CIELAB space) is one of the uniform color spaces defined by the CIE in 1976.

Lightness variable	$L^* = 116\left(\frac{Y}{Y_n}\right)^{1/3} - 16$	
Chromaticity coordinates	$a^* = 500\left[\left(\frac{X}{X_n}\right)^{1/3} - \left(\frac{Y}{Y_n}\right)^{1/3}\right]$	$b^* = 200\left[\left(\frac{Y}{Y_n}\right)^{1/3} - \left(\frac{Z}{Z_n}\right)^{1/3}\right]$
Color distance	$\text{delta E} = \sqrt{(dL^*)^2 + (da^*)^2 + (db^*)^2}$	

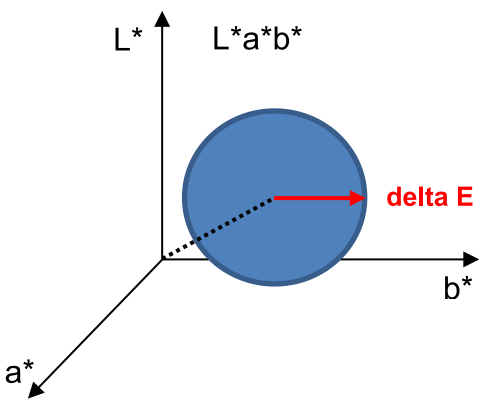
X Y Z	Normfarbwerte XYZ der Probe
Xn Yn Zn	Normfarbwerte XYZ eines vollkommen mattweißen Körpers.
dL* da* db*	Differenzen der aktuellen L*, a* und b* Werte von der Probe (aktuelle Oberfläche) zu den Sollwerten. Siehe **SETVALUES und TOL.**

N*i*r*:

Spatial coordinate N*	$N^* = 116\left(\frac{NIR2}{NIR2_n}\right)^{1/3} - 16$
Spatial coordinate i*	$i^* = 500\left[\left(\frac{NIR1}{NIR1_n}\right)^{1/3} - \left(\frac{NIR2}{NIR2_n}\right)^{1/3}\right]$
Spatial coordinate r*	$r^* = 200\left[\left(\frac{NIR2}{NIR2_n}\right)^{1/3} - \left(\frac{NIR3}{NIR3_n}\right)^{1/3}\right]$
Spatial distance	$\text{delta Nir} = \sqrt{(dN^*)^2 + (di^*)^2 + (dr^*)^2}$

NIR1 NIR2 NIR3	Norm values NIR1, NIR2, NIR3 of the sample (analog values of the current surface)
NIR1n NIR2n NIR3n	Norm values NIR1, NIR2, NIR3 of the reference surface
dN* di* dr*	Differences between the current N*, i* and r* values of the sample (current surface) and the set values. See **SETVALUES** and **TOL**.

Example with the spacial coordinates **L*a*b*** (dito **N*i*r***)



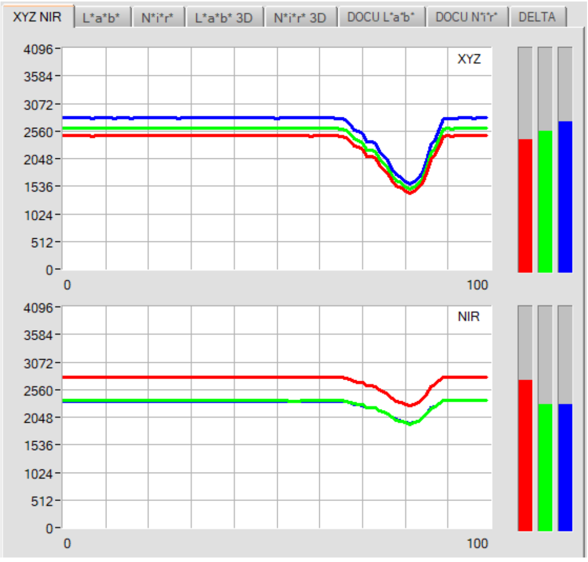
The spatial coordinates **L*a*b*** and **N*i*r*** each form a point in three-dimensional space. If a tolerance is placed around this point, this results in a sphere in space.
The sphere is therefore defined by its set values and a tolerance. See **SETVALUES** and **TOL.**

During the measuring process, **L*a*b*** and **N*i*r*** values are determined from the respective surface.
You can now calculate a spatial distance between the measured values in the room and the set values.

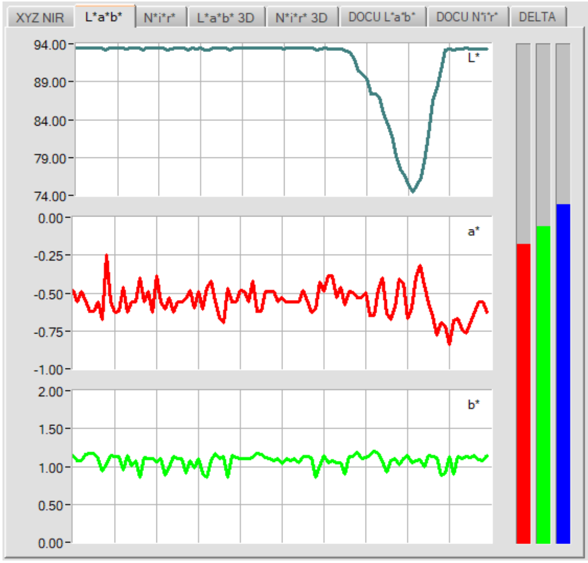
If the spatial coordinates are considered individually, a separate distance is obtained for each coordinate.
→ **dL*, da*, db*** or **dN*, di*, dr*** (d=delta)

If you consider the spatial coordinates as a point in space, you can calculate the distance between the setpoint and the current measuring point.
→ **delta E** and **delta Nir.**

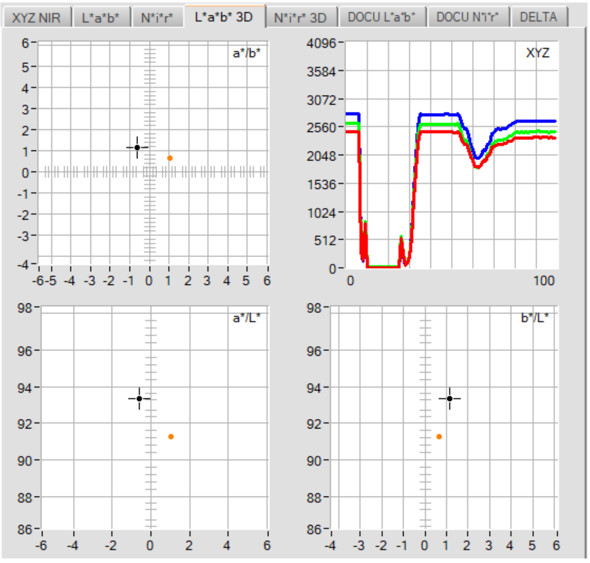
Tab **XYZ NIR:**
Current raw signals (standard color values) **X, Y, Z** and **NIR1, NIR2, NIR** are displayed.



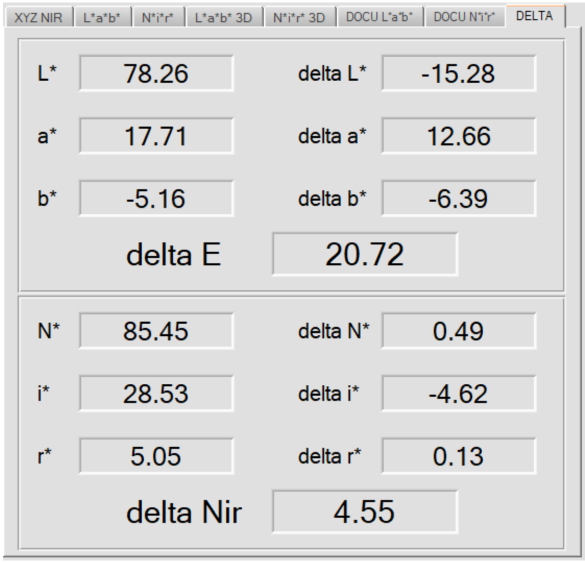
Tabs **L*a*b*** and **N*i*r*:**
The respective spatial coordinates are displayed in a line graph.



Tabs **L*a*b* 3D** and **N*i*r* 3D**:
Three-sided view of the target values (black cross) and current values in the room (orange dot).
The taught spatial coordinates and the current spatial position are displayed. A three-sided view of the space was selected for better visualization.



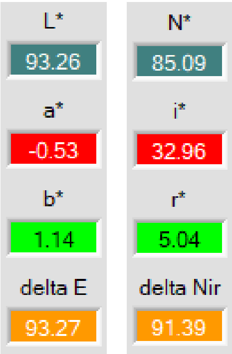
Tab **DELTA:**
The respective spatial coordinates and their deviation (delta) from the set values are shown in numerical displays.
See also **DOCU** tab.



The tabs **DOCU L*a*b*** and **DOCU N*i*r*** are described in more detail in the description of the tab **DOCU**.

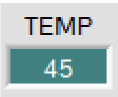
TIP! Double-clicking with the left mouse button in one of the line graphs starts an automatic zoom function.
You can stop this again with a single click.

These displays show the set values currently measured on the surface.

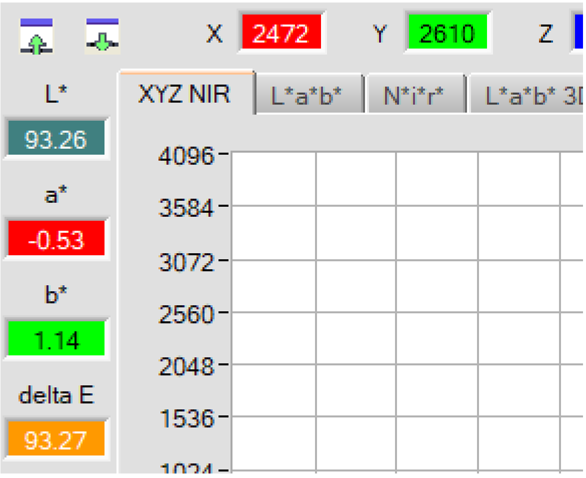


The displays in the left-hand column show the color coordinates and brightness values **L*a*b*** calculated from the standard color values **X, Y, Z** as well as the calculated spatial distance **delta E** to the set value.

The displays next to it show the spatial coordinates **N*i*r*** calculated from **NIR1, NIR2, NIR3** as well as their spatial distance **delta Nir** to the set value.



TEMP:
This display shows the temperature prevailing in the sensor housing.
The display does NOT correspond to degrees Celsius or Fahrenheit.



You can use the green arrow buttons in the top left-hand corner to zoom in and out of the right-hand part of the display window.

Please note:	**The above output fields are only updated when data transfer between the PC and the sensor is active (GO button pressed).**

2.4 Tab CALIB

2.4.1 Calibration

The sensors are calibrated at the factory. If the parameter **CALIBRATION** is set to **ON**, the sensor works with the factory calibration.

If the sensor is intended to be operated in “uncalibrated” mode, select **CALIBRATION=OFF**.

Alternatively, the user can calibrate the sensor himself to different surfaces.
If this is done, the factory calibration is overwritten, and the sensor works with the calibration performed by the user.

Press **START CALIBRATION ASSISTANT** to start the wizard, which guides you step by step through the calibration process.

Calibration can be performed with either **RAL** or **NIR** plates, the **L*a*b*** and **N*i*r*** values of which are available in a file. During calibration, you will be prompted to load this file. A **RAL** or **NIR** plate is then placed in front of the sensor. The corresponding **L*a*b*** or **N*i*r*** values are then loaded from the file by entering the corresponding serial number.

There are 3 conditions for this:

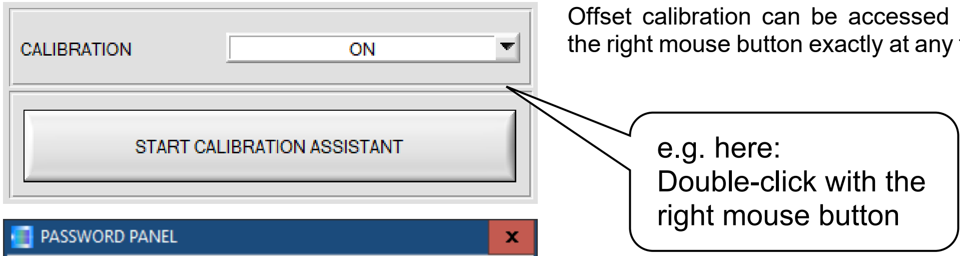
1. You need a **RAL** and a **NIR** reference plate. These plates are available from the supplier.
2. Different **RAL** color plates from the ‘RAL CLASSIC’ or ‘RAL DESIGN SYSTEM plus’ series are required. These are available from the supplier of the sensor. You also need various **NIR** plates, which are also available from the supplier.
3. You need a file (.ini file) in which the **L*a*b*** values for the various **RAL** plates are stored. This file is also available from the supplier (via download → RAL PLATE Data Files). In addition, you need the corresponding file in which all **N*i*r*** values of the **NIR** plates are stored.

Another method is to enter the **L*a*b*** and **N*i*r*** values manually (by hand). In this case, you also need a **RAL** and **NIR** reference surface and various surfaces from which the **L*a*b*** and **N*i*r*** values are known.
If you save the calibration values in a file after successful calibration when prompted, you can load them as a calibration template for the next calibration, expand them or send them directly to the sensor.

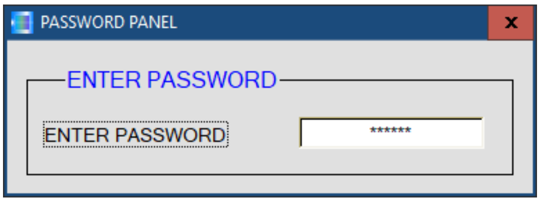
The individual tables, displays and display boxes for calculating the calibration factors are updated automatically during the calibration process. They cannot be changed and do not require any further attention.

2.4.2 Offset calibration

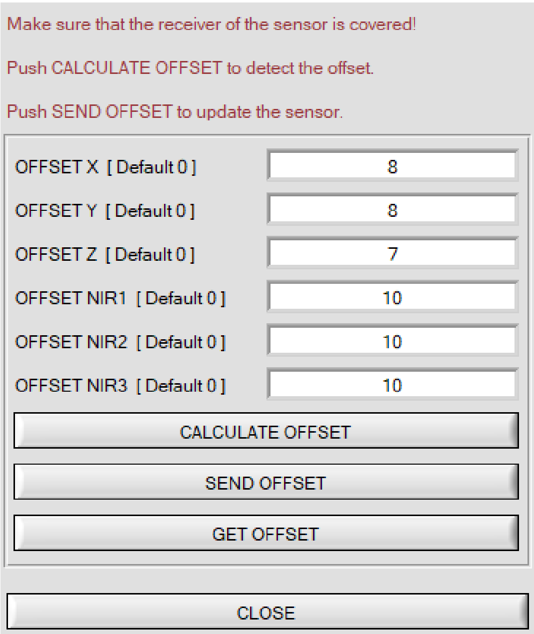
To avoid an increase of the electronic offset when using the integral function (**INTEGRAL** parameter), this offset can be eliminated by way of offset calibration or zero-point calibration. The corresponding tab is password-protected to prevent inadvertent incorrect settings.



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



You will then be prompted to enter a password.
The password is: **mellon**



Now please follow the instructions provided in the tab.

ATTENTION!
It is extremely important for offset calibration that the receiver is not exposed to any extraneous light.
Please make sure that you cover the receiver of the sensor e.g. witch a black cloth that is impervious to light.

This is absolutely necessary for proper offset calibration.

Now press **CALCULATE OFFSET**. The offset values should be significantly below 100 for all channels.

The offset values can then be sent to the sensor by pressing **SEND OFFSET**.
GET OFFSET can be used to check whether the data have been sent.

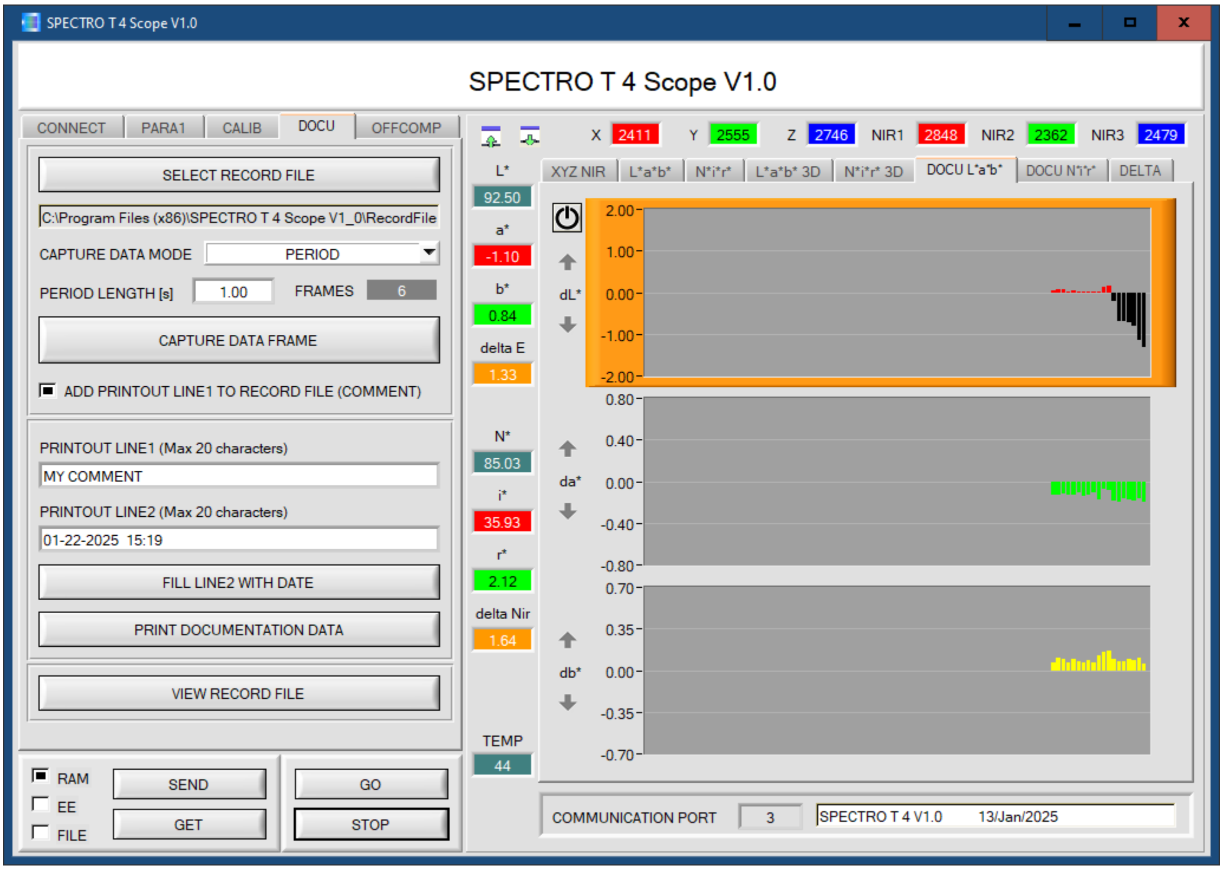
2.5 Tab DOCU

Clicking **DOCU** opens a view in which, after pressing **CAPTURE DATA FRAME**, a recording of the space coordinates starts depending on the setting in CAPTURE DATA MODE.

If **CAPTURE DATA MODE = MANUAL**, a single frame is picked and displayed.
In case of **CAPTURE DATA MODE = PERIOD**, space coordinates are sampled via a **PERIOD LENGTH [s]** long time, summed up and then averaged and displayed.

The delta values of the individual space coordinates are always calculated from the recorded space coordinates and the set values in the **PARA1** tab and are visualized in the individual displays in the **DELTA** tab and the bar graphs in the **DOCU L*a*b*** and **DOCU N*i*r*** tabs.

If the deltas leave the taught tolerance, the corresponding graphs and displays are highlighted in orange.



With the button **SELECT RECORD FILE** you select a file in which the **X, Y, Z** values, the space coordinates, the respective **delta** values as well as the hit (**V-No:**) and the sensor temperature can be stored.
If the checkbox ADD PRINTOUT LIN1 TO RECORDER FILE (COMMENT) is selected, additionally the content of PRINTOUT LINE1 is stored in the file.

The file is saved after pressing **CAPTURE DATA FRAME**.
It should be noted that the file is only regenerated after pressing **SELECT RECORD FILE**.
In any other case, the data is simply appended to the file displayed in the file path.
Even if the software has been restarted.

Pressing **PRINT DOCUMENTATION DATA** will output the current space coordinates, their **delta** values and the two text lines **PRINTOUT LINE1** and **PRINTOUT LINE2** to a printer. The button **FILL LINE2 WITH DATE** is used to enter the current date and time into **PRINTOUT LINE2**.

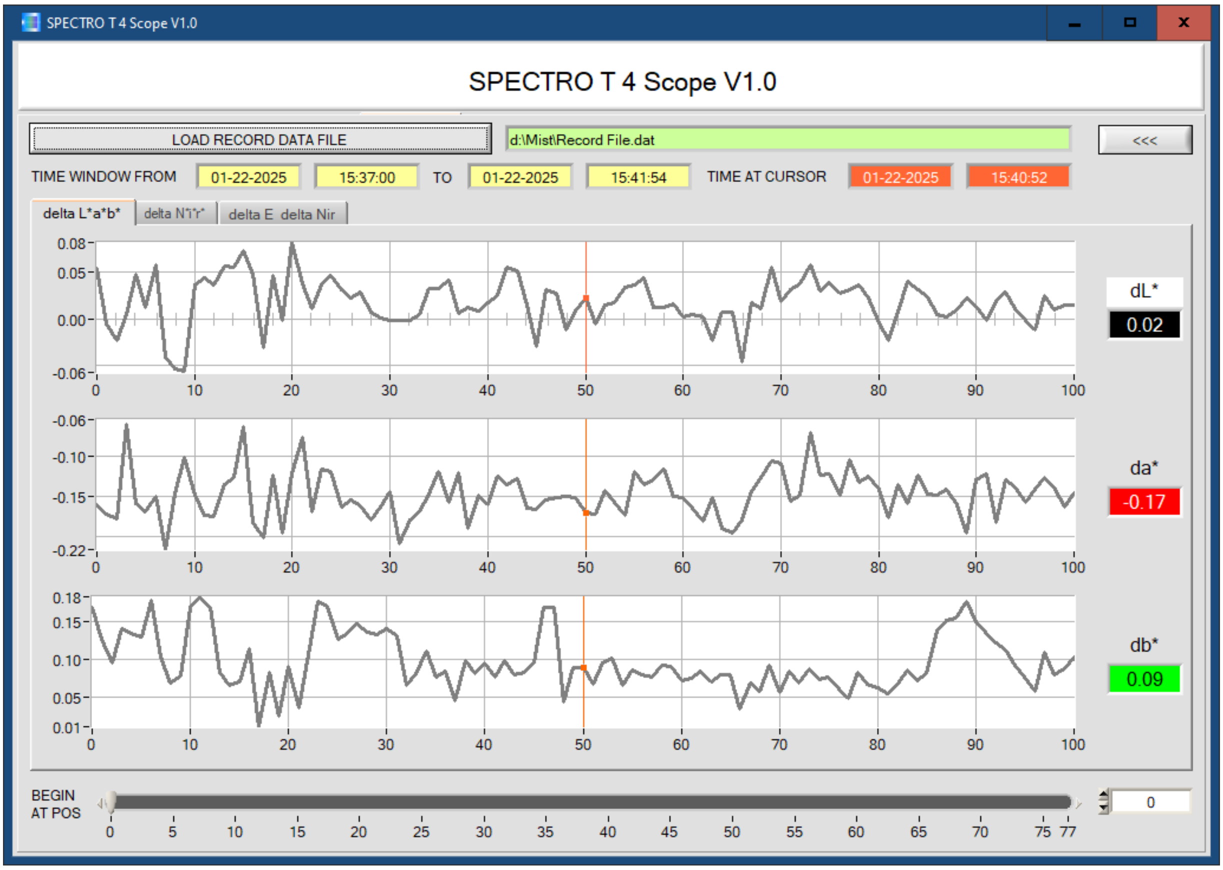
The data can also be output to a label printer. Please note that the labels must be at least 50mm in width and 30mm in height.
If the labels are not output correctly, this can be optimized in the settings of the respective printer(printer-dependent).

After pressing **VIEW RECORD FILE**, a window is opened in which you can display the data from a recorded data file by pressing **LOAD RECORD DATA FILE**.

The data saved in the file are shown in the displays and in the graphs.
From the entire file only a section of 100 values is shown in the graph.
However, you can scroll through the entire file using the **BEGIN AT POS** slider.
The yellow **TIME WINDOW** displays show the time window that the graphs are currently visualizing.

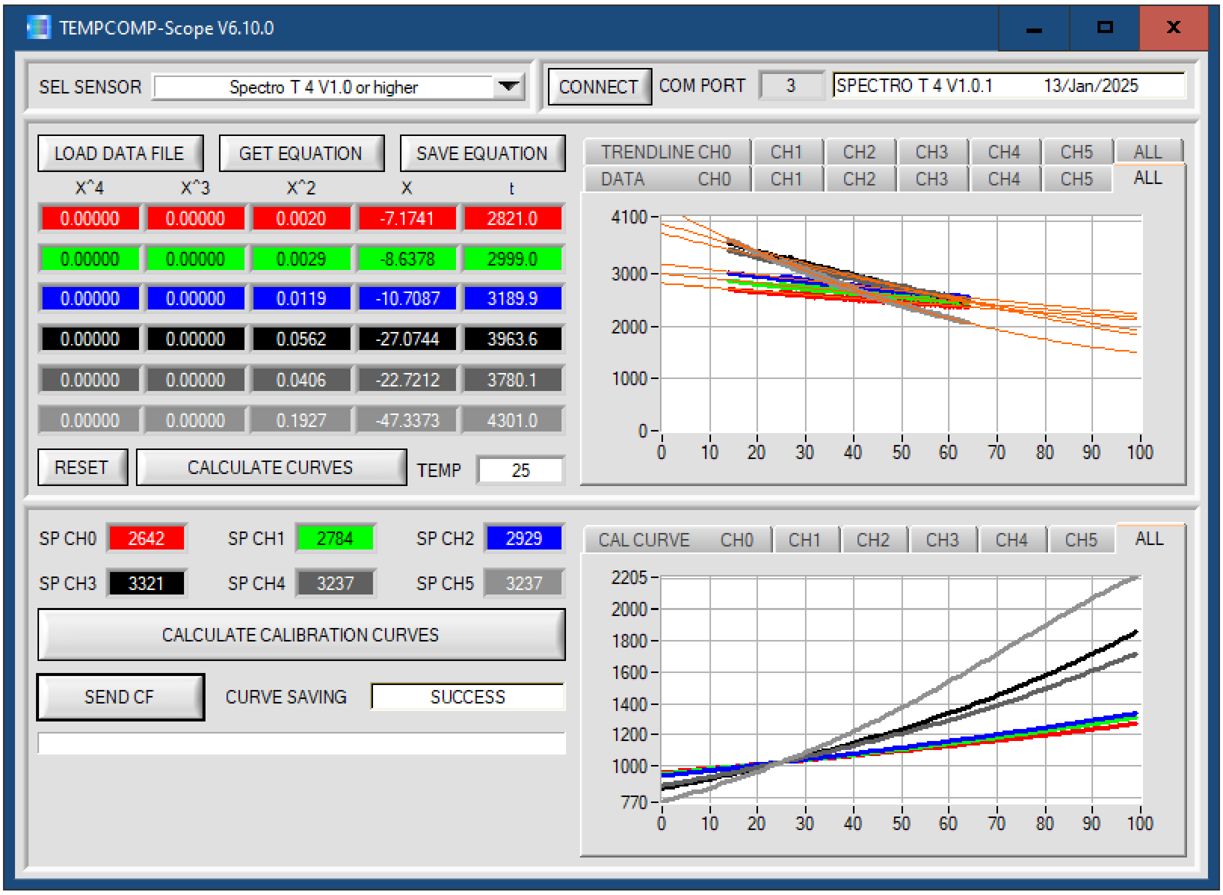
The cursor can be used to move to individual points, the value of which is shown in the displays to the right.
Moreover, the **TIME AT CURSOR** display shows when this point was recorded.

Press **<<<** to return to the main window.



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 is either available also via the provided download link or included on the supplied DVD.
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 **SEL 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.

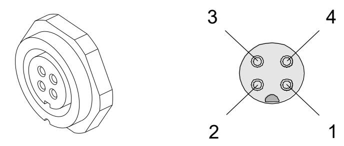
Step 6: Temperature compensation is successfully finished if the **SUCCESS** status message is then displayed.

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-4 sensors

Connection of SPECTRO-T-4 to PC:

4-pole fem. M5 connector (type Binder 707)
SPECTRO-T-4/PC-RS232

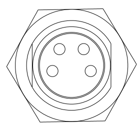


Pin-No.:		Assignment:
1		+24VDC (+Ub)
2		0V (GND)
3		Rx0
4		Tx0

Connecting cables to choose from:
cab-las4/PC-…
cab-4/USB-…
cab-4/ETH-…
cab-4-PNET-500 (PROFINET adapter)

Connection of SPECTRO-T-4 to PLC:

4-pole M8 connector (type Binder 768/718)
SPECTRO-T-4/PLC



Pin No.:	Color:	Assignment:
1	brown	+24VDC (± 10%)
2	white	Not connected
3	blue	0V (GND)
4	black	Not connected

Connecting cable:
cab-M8/4-g-…

5. RS232 communication protocol

The sensors of the SPECTRO-T-4 series operate with the following **PARAMETERS** that are sent to the sensor or read from the sensor in the stated sequence.
Info! 1 **bytes** = 8bit 1 **word** = 2 **byte** 1 **long** = 2 **word** = 4 **byte**

PARAMETERS		Type	Meaning
Para1:	POWER 0	**word**	Parameter to adjust the Intensity for the VIS transmitter to detect X, Y, Z (0 ... 1000)
Para2:	POWER 1	**word**	Parameter to adjust the Intensity for the NIR transmitter to detect NIR1 (0 ... 1000)
Para3:	POWER 2	**word**	Parameter to adjust the Intensity for the NIR transmitter to detect NIR2 (0 ... 1000)
Para4:	POWER 3	**word**	Parameter to adjust the Intensity for the NIR transmitter to detect NIR3 (0 ... 1000)
Para5:	GAIN	**word**	Amplification of the integrated receiver for VIS signals X, Y, Z AMP1, AMP2, AMP3, AMP4, AMP5, AMP6, AMP7, AMP8 coded to (1, 2, 3, 4, 5, 6, 7, 8)
Para6:	INTEGRAL	**word**	X, Y, Z Raw signal integration (1…250)
Para7:	GAIN	**word**	Amplification of the integrated receiver for NIR signals NIR1, NIR2, NIR3 AMP1, AMP2, AMP3, AMP4, AMP5, AMP6, AMP7, AMP8 coded to (1, 2, 3, 4, 5, 6, 7, 8)
Para8:	INTEGRAL	**word**	NIR1, NIR2, NIR3 Raw signal integration (1…250)
Para9:	AVERAGE	**word**	Signal averaging 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384 or 32768
Para10:	CALIB	**word**	Calibration Mode: OFF, ON coded to (0,1)

There are 8 **SETVALUES** which can be sent to the sensor or read from the sensor in the stated sequence.
The sensor does not need these values. However, they can be stored in the sensor and read if required.
Info! 1 **bytes** = 8bit 1 **word** = 2 **byte** 1 **long** = 2 **word** = 4 **byte**

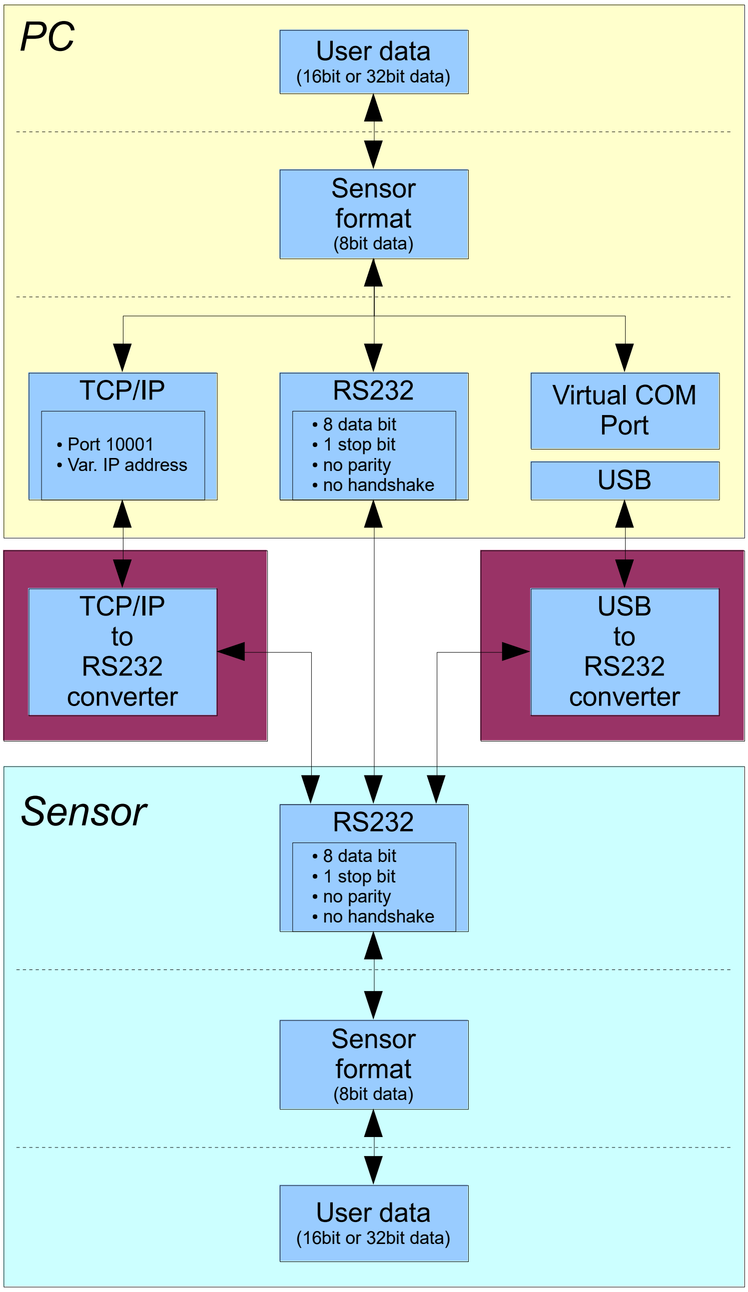
SETVALUES		Type	Meaning
SetVal1:	SV L*	**long**	Setvalue L*. Must be multiplied with 65536 for sending and divided by 65536 after reading.
SetVal2:	SV a*	**long**	Setvalue a*. Must be multiplied with 65536 for sending and divided by 65536 after reading.
SetVal3:	SV b*	**long**	Setvalue b*. Must be multiplied with 65536 for sending and divided by 65536 after reading.
SetVal4:	SV N*	**long**	Setvalue N*. Must be multiplied with 65536 for sending and divided by 65536 after reading.
SetVal5:	SV i*	**long**	Setvalue i*. Must be multiplied with 65536 for sending and divided by 65536 after reading.
SetVal6:	SV r*	**long**	Setvalue r*. Must be multiplied with 65536 for sending and divided by 65536 after reading.
SetVal7:	TOL L*a*b*	**long**	Setvalue TOL L*a*b*. Must be multiplied with 65536 for sending and divided by 65536 after reading.
SetVal8:	TOL N*i*r*	**long**	Setvalue TOL N*i*r*. Must be multiplied with 65536 for sending and divided by 65536 after reading.

Upon request, the **DATA** acquired and processed by the sensor are sent by the sensor in the following sequence.
Info! 1 **bytes** = 8bit 1 **word** = 2 **byte** 1 **long** = 2 **word** = 4 **byte**

DATA VALUES		Type	Meaning
DatVal1:	L*	long	Color Space Lightness variable L*. Must be divided by 65536.
DatVal2:	a*	long	Color Space chromaticity coordinate a*. Must be divided by 65536.
DatVal3:	b*	long	Color Space chromaticity coordinate b*. Must be divided by 65536.
DatVal4:	N*	long	Control Space coordinate N*. Must be divided by 65536.
DatVal5:	i*	long	Control Space coordinate i*. Must be divided by 65536.
DatVal6:	r*	long	Control Space coordinate r*. Must be divided by 65536.
DatVal7:	TEMP	word	Temperature in the sensor (**not** in °C or °F)
DatVal8:	X	word	Calibrated and temperature compensated value of channel X
DatVal9:	Y	word	Calibrated and temperature compensated value of channel Y
DatVal10:	Z	word	Calibrated and temperature compensated value of channel Z
DatVal11:	NIR1	word	Calibrated and temperature compensated value of channel NIR1
DatVal12:	NIR2	word	Calibrated and temperature compensated value of channel NIR2
DatVal13:	NIR3	word	Calibrated and temperature compensated value of channel NIR3
DatVal14:	RAW X	word	None Calibrated value of channel X
DatVal15:	RAW Y	word	None Calibrated value of channel Y
DatVal16:	RAW Z	word	None Calibrated value of channel Z
DatVal17:	RAW NIR1	word	None Calibrated value of channel NIR1
DatVal18:	RAW NIR2	word	None Calibrated value of channel NIR2

DatVal19: RAW NIR3		word	None Calibrated value of channel NIR3

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 0 up to and incl. data byte n).
The eight byte is the CRC8 checksum for the header and is formed from bytes 0 up to and incl. 6.
The header always has a total length of 8 bytes. The complete frame may contain between 8 and 520 bytes.

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Data	Byte9 Data	…	Byte n+6 Data	Byte n+7 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
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  := AAhex
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 **PARAMETERS** or **SETVALUES** to the RAM of the sensor.
<ARG> determines whether you want to save parameter or setvalues
ARG = 0 --> Parameters
ARG = 1 --> Setvalues

Example is with 5 parameter (Para1=500, Para2=0; Para3=3200, Para4=3300, Para5=1)
Refer to the table **PARAMETERS** or **SETVALUE**S to find out how many values need to be sent.

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.
<ARG> determines whether you want to read parameter or setvalues
ARG = 0 --> Parameters
ARG = 1 --> Setvalues

Example is with 5 parameter (Para1=500, Para2=0; Para3=3200, Para4=3300, Para5=1)
Refer to the table **PARAMETERS** or **SETVALUE**S to find out which 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)	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

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)	3	0	0	0	0	170	142
		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)	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

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)	4	0	0	0	0	170	11
		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)	4	0	0	0	0	170	11
		ARG=0		LEN=0			

Example order=5: Read CONNECTION OK 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)	5	0	0	0	0	170	60
		ARG=0		LEN=0			

DATA FRAME Sensor → PC
ARG determines the serial number of the 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)	5	170	0	0	0	170	178
		ARG=170		LEN=0			

Example order=7: Read Firmware String 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)	7	0	0	0	0	170	82
		ARG=0		LEN=0			

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

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)	ASCII	ASCII	ASCII	ASCII
85 (dec)	7	0	0	72	0	183	38	F	I	R	M
		ARG=0		LEN=72							

Byte12 Data	Byte13 Data	Byte14 Data	Byte15 Data	Byte16 Data	Byte17 Data	Byte18 Data	Byte19 Data	Byte20 Data	Byte21 Data	Byte22 Data	Byte23 Data
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
W	A	R	E		S	T	R	I	N	G	

Byte24 Data	Byte25 Data	Byte26 Data	Byte27 Data	Byte28 Data	Byte29 Data	Byte30 Data	Byte31 Data	Byte32 Data	Byte33 Data	Byte34 Data	Byte35 Data
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
											R

Byte36 Data	Byte37 Data	Byte38 Data	Byte39 Data	Byte40 Data	Byte41 Data	Byte42 Data	Byte43 Data	Byte44 Data	Byte45 Data	Byte46 Data	Byte47 Data
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
T	:	K	W	x	x	/	x	x			

Byte48 Data	Byte49 Data	Byte50 Data	Byte51 Data	Byte52 Data	Byte53 Data	Byte54 Data	Byte55 Data	Byte56 Data	Byte57 Data	Byte58 Data	Byte59 Data
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII

Byte60 Data	Byte61 Data	Byte62 Data	Byte63 Data	Byte64 Data	Byte65 Data	Byte66 Data	Byte67 Data	Byte68 Data	Byte69 Data	Byte70 Data	Byte71 Data
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII

Byte72 Data	Byte73 Data	Byte74 Data	Byte75 Data	Byte76 Data	Byte77 Data	Byte78 Data	Byte79 Data	Byte80 Data	Byte81 Data
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII	ASCII

Example order=8: Read data values 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)	8	0	0	0	0	170	118
		ARG=0		LEN=0			

DATA FRAME Sensor → PC

8 Header Bytes and all **DATA VALUES**
Refer to the table **DATA VALUES** to find out which values you will receive.

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	206	163	40	28	2	0
		ARG=0		LEN=8				CYCLE COUNT = 138280			

Byte12 Data	Byte13 Data	Byte14 Data	Byte15 Data
lo word lo byte	lo word hi byte	hi word lo byte	hi word hi byte
144	1	0	0
COUNTER TIME = 400			

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

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

Example order=190: Write new baud rate to the 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)	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
ARG=5: baud rate = 230400
ARG=6: baud rate = 460800

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)	190	0	0	0	0	170	195
		ARG=0		LEN=0			

6. Commissioning the label printer EOS2/300

We recommend the following label printer and accessories:

Printer	Accessories	Accessories
Company: cab Produkttechnik GmbH & Co. KG Model: cab EOS2/300 Item No. 5978202	Adhesive roll label Item No. 5780036 Material: TF Satin White Width x Height: 50x30mm	Thermal transfer ribbon Item No. 5556661

You can download drivers, documentation, additional software, etc. at the following link:
https://www.cab.de/en/marking/label-printer/eos25/?ref=nav_lang

The installation of the software for the label printer is described in this manual.

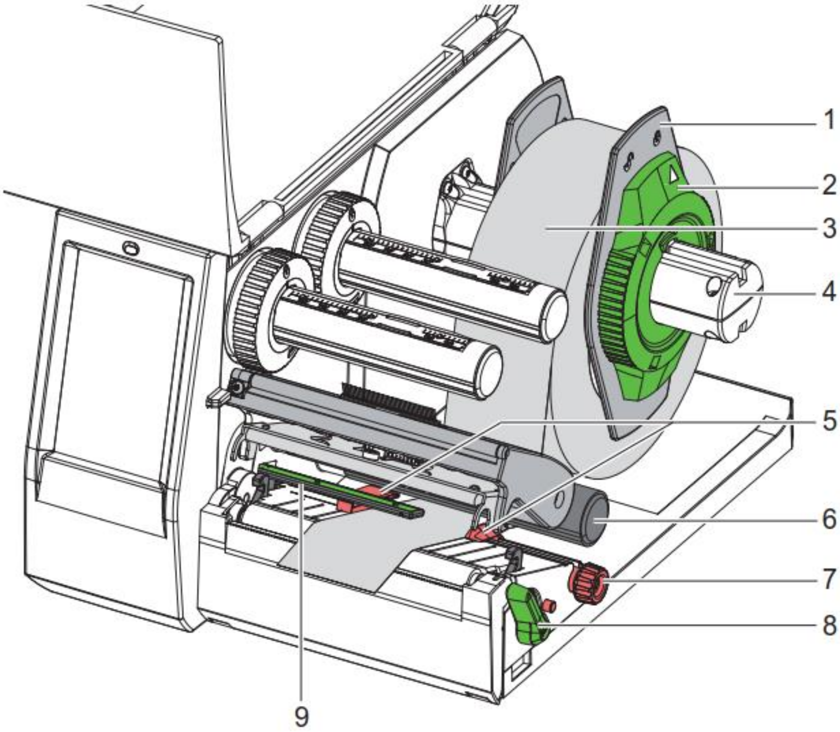
Label printer and accessories can also be purchased from our company.

We have no experience with other printers and therefore cannot provide support.

6.1 Loading the roll adhesive label and the thermal transfer ribbon

See also Operator's Manual EOS2 / EOS5, subitem **Loading Material** (▷ 4 on page 12).
https://www.cab.de/en/marking/label-printer/eos25/#dokumentation → Documentation → Operation Manuals

6.1.1 Loading media from roll



1. Turn ring (2) at the margin stop (1) counterclockwise, so that the arrow points to the symbol 🔓., and thus release the margin stop.
2. Remove the margin stop (1) from the roll retainer (4).
3. Load label roll (3) on the roll retainer (4) in such a way that the labels can be inserted into the printhead in the right position. The printing side of the labels must be visible from above.
4. Re-mount the margin stop (1) onto the roll retainer (4). Push the margin stop (1) to the roll (3) until the roll touches both margin stops and a clear resistance is encountered.
5. Turn ring (2) clockwise, so that the arrow (10) points to the symbol 🔒 and thus fix the margin stop (1) on the roll retainer (4).
6. Turn lever (8) counterclockwise to open printhead.
7. Position guides (5) by turning the knob (7) so that they are several millimeters wider than the material.
8. Position material below the roller (6) and guide it through the print unit.

Attention! Guide material through the print unit below the label sensor (9).

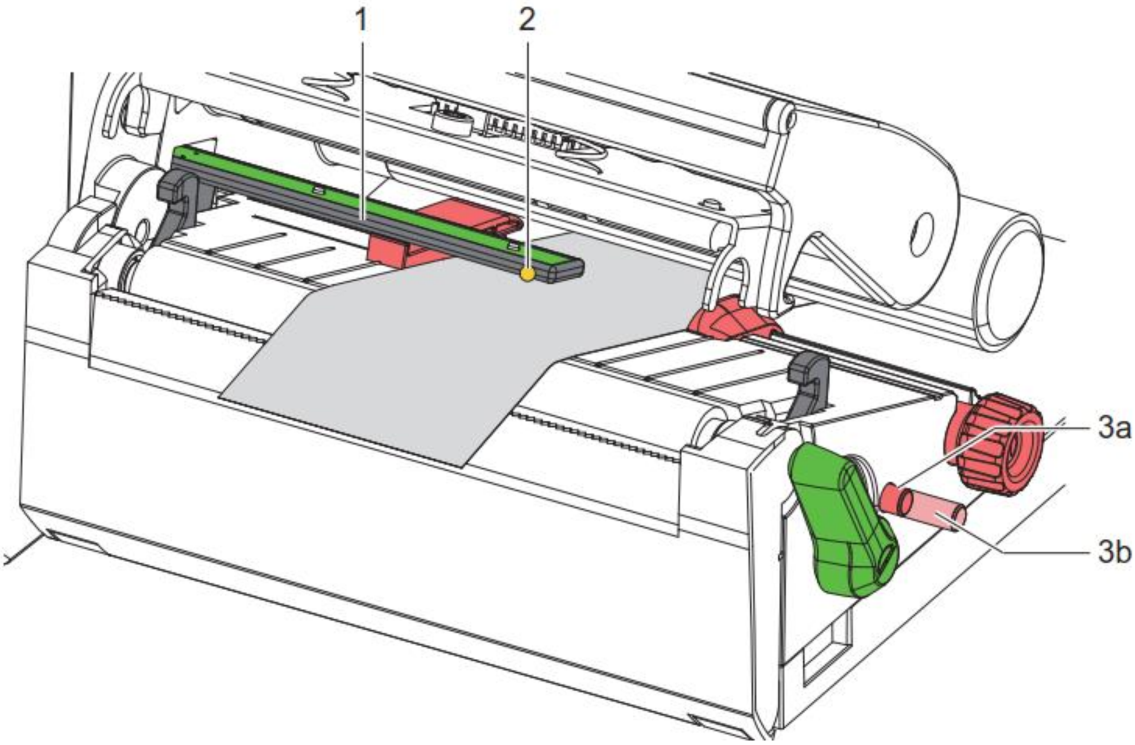
9. Move guides (5) closely to the edges of the material without clamping the material.
10. Adjust label sensor (see next page)
11. Press printhead retainer down and turn lever (8) clockwise to lock the printhead.

Adjusting the label sensor

Note!

When the printer is delivered the label sensor is positioned in the middle of the paper feed.
Thus, the label sensor must only be adjusted if materials with the following characteristics are used:
• Material with reflex or cut-out marks, which are not in the middle.
• Multi-strip material with an even number of strips.
• Material with irregularly shaped labels.

If the labels recommended by us are used, an adjustment is not necessary.



The sensor position (3) is marked with a yellow LED in the sensor retainer.

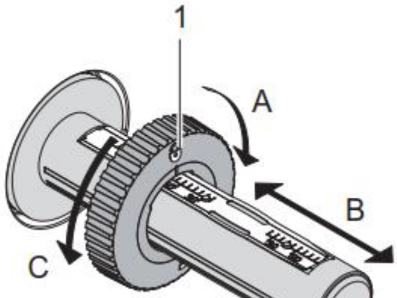
1. Open the printhead.

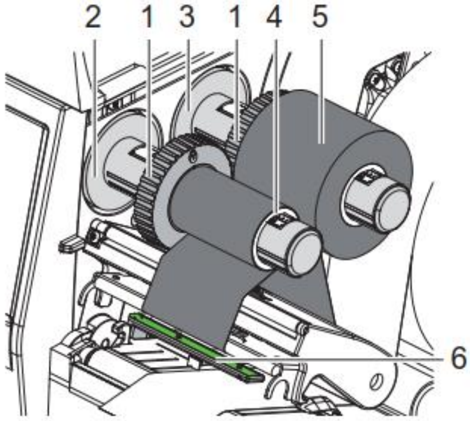
2. Bring knob (3) by pressing it into the setting position (3b).

3. Position the sensor holder (1) by turning the knob (3) so that the sensor (2) can detect the front edge of the label in the direction of paper flow or the reflex or cut-out mark.

4. Bring knob (3) by pressing it into the basic position (3a).

6.1.2 Loading transfer ribbon





1. Clean the printhead before loading the transfer ribbon (see Operator's Manual EOS2, ▷ 6.3 on page 18).

2. Open the printhead.

3. Set guide (1) on the ribbon supply hub to the correct transfer ribbon width (Figure left):
   • Hold the hub and unlock the guide (1) by turning it in direction A..
   • Slide the guide (1) in direction B and adjust the guide to the ribbon width using the scale.
   • Hold the hub and lock the guide (1) by turning it in direction C.

4. Slide transfer ribbon roll (5) onto the ribbon supply hub (3) until it reaches the guide (1) so that the color coating of the ribbon faces downward when being unwound.

Note!
To rewind the ribbon a ribbon core (4) is needed that must be at least equal in width to the supply ribbon. When changing the transfer ribbon use the empty supply ribbon core for winding the next ribbon.

5. Adjust the position of the guide on the take-up hub to the width of the ribbon core (4) and push ribbon core on the take-up hub (2).

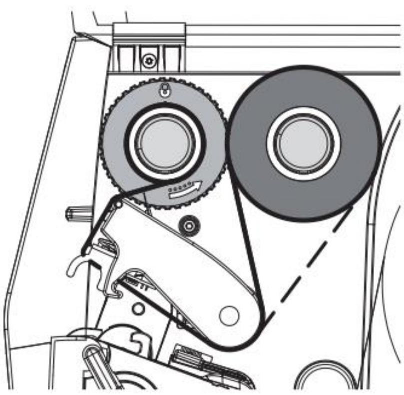
6. Guide the transfer ribbon though the printing unit as shown in Figure below.

ATTENTION!
Guide transfer ribbon over the label sensor (6).

7. Secure starting end of transfer ribbon to the transfer ribbon core (4) with adhesive tape. Regard counterclockwise rotation direction of the transfer ribbon take-up hub here.

8. Turn transfer ribbon take-up hub (2) counterclockwise to smooth out the feed path of the transfer ribbon.

9. Close the printhead.



6.2 Installing the printer driver for Windows via USB cable

The following describes how to install the driver for the EOS2 label printer under Windows via the USB interface. See also PDF file „Configuration manual cab Label Printers“ under the following link:
https://www.cab.de/en/marking/label-printer/eos25/#dokumentation

1. Make sure that the label printer is switched off.

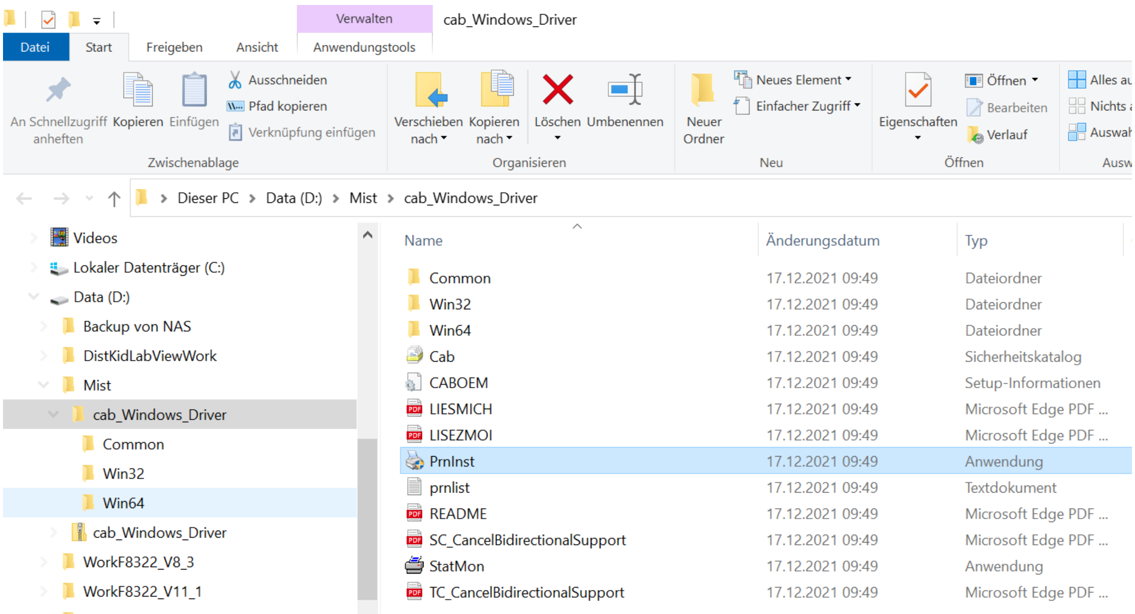
2. Exit all running programs

3. Connect the printer to the power supply and connect the A-B cable to your computer.
   ATTENTION!
   Do not yet switch on the printer!

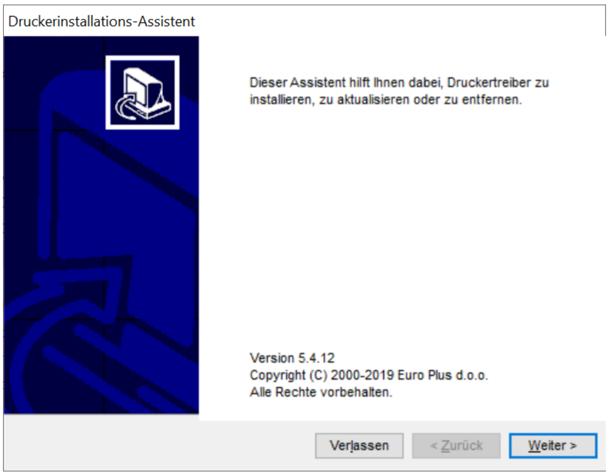
4. First download the printer driver "cab Windows Driver" to the hard disk of your PC.
   You can find the driver on the cab homepage under the following link:
   https://www.cab.de/en/marking/label-printer/eos25/#software
   → Driver → cab Windows Driver → Save file
   The driver is then either in the download folder or in the path you may have selected.
   Unzip the "cab Windows Driver.zip" file.

5. Now turn on the printer. The Windows installation wizard is started automatically.

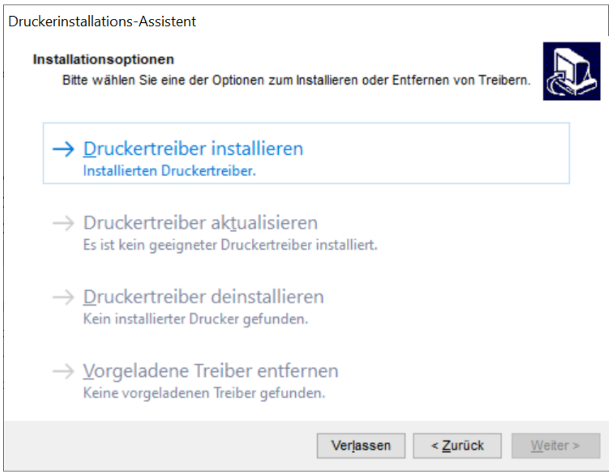
6. Start the application „PrnInst“ and follow the instructions on the screen. After successful installation, an icon for the label printer appears in the Windows system folder "Printers".



Confirm with "Next".

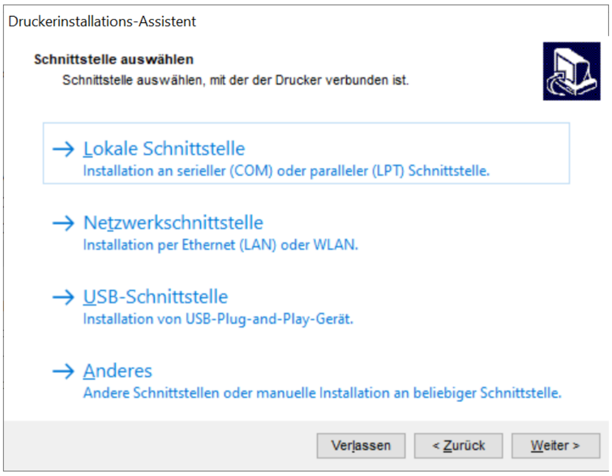


Select → Install printer driver.

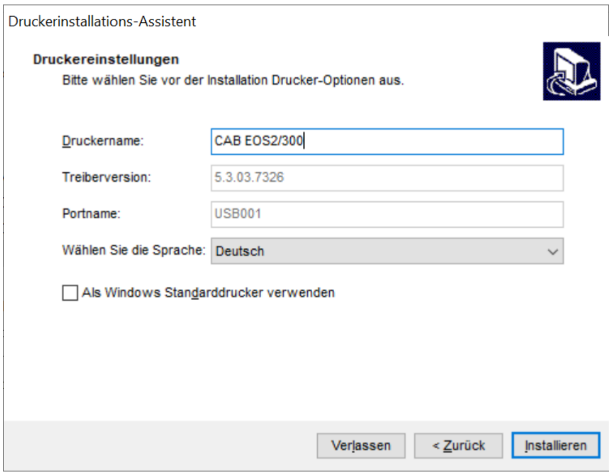


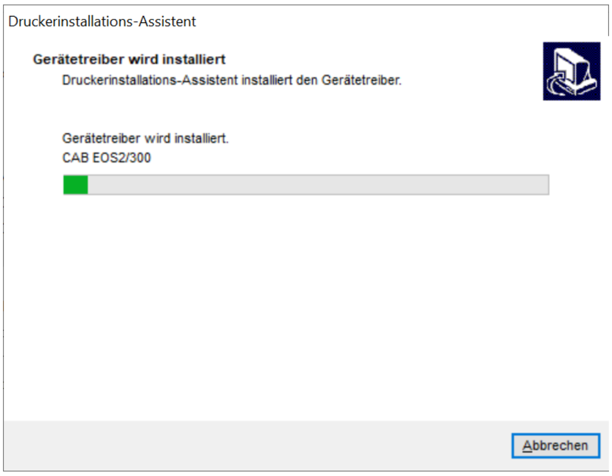
Select → USB interface.

The USB interface may have to be briefly disconnected and then reconnected.

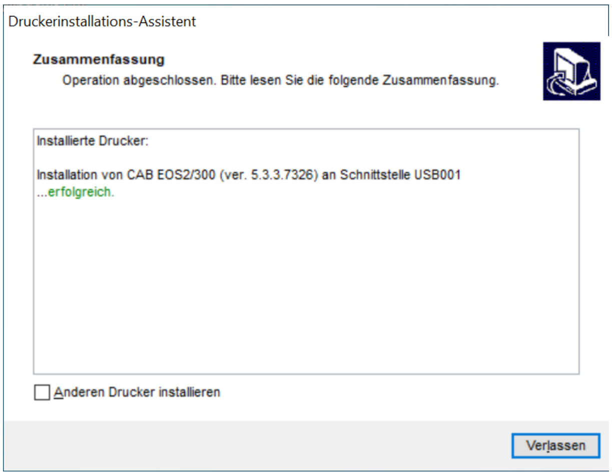


If the printer was recognized, you can still make printer settings here.
However, this is normally not necessary.
Press → Install.



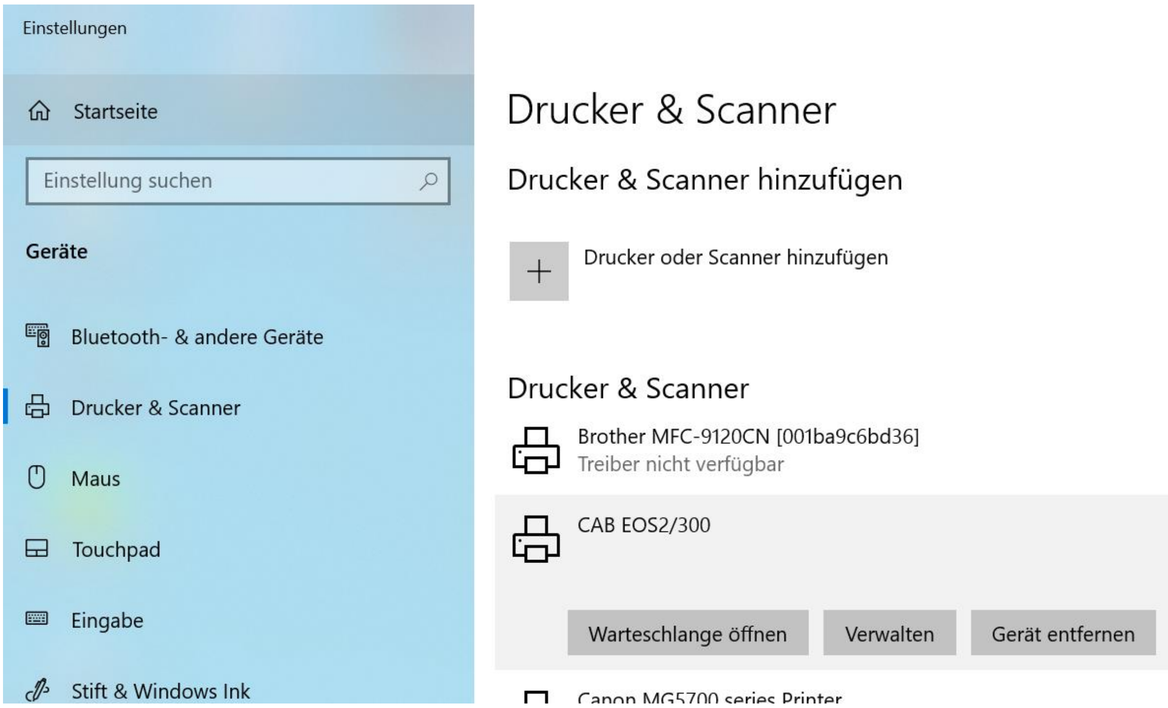


If the installation was successful, the following window appears.
Now press → Exit.

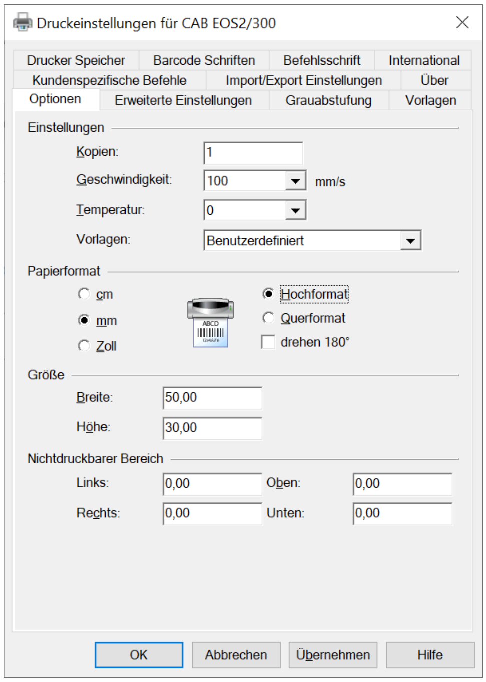


6.3 Setting the printer properties

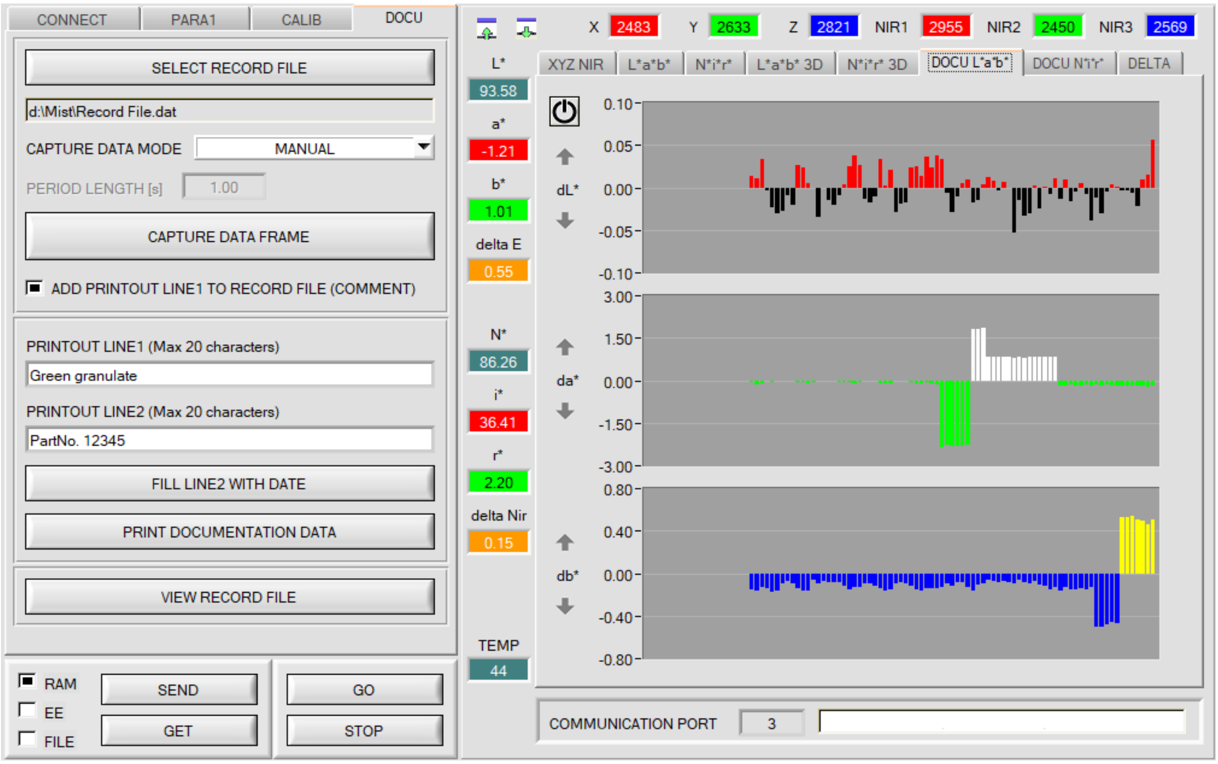
After the printer has been installed, it is displayed in the Windows system folder "Printers".



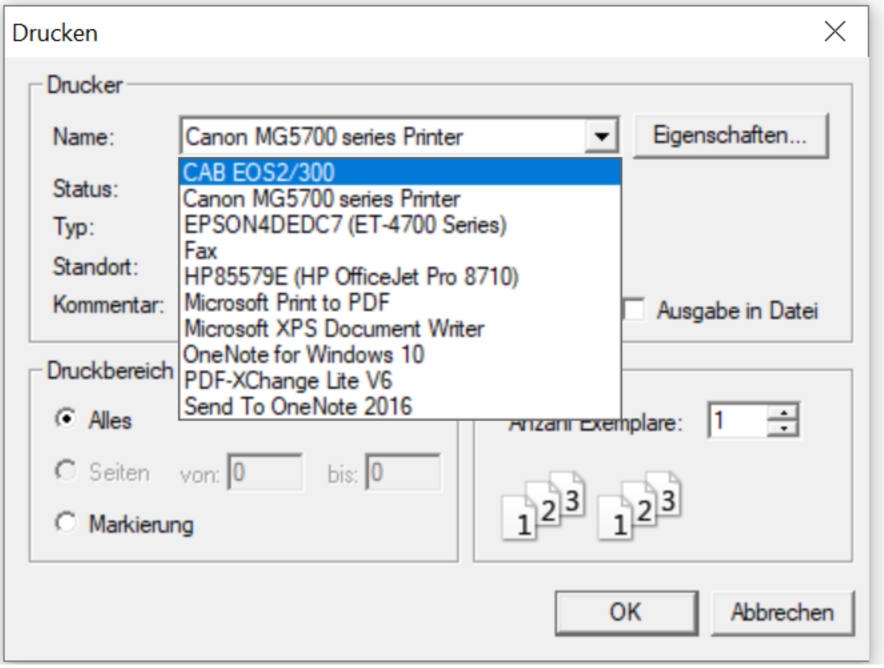
Press → Manage and then → Printer settings.
For → Size, set 50.00 for Width and 30.00 for Height.
This is the label size.
Confirm with → Apply and → OK.



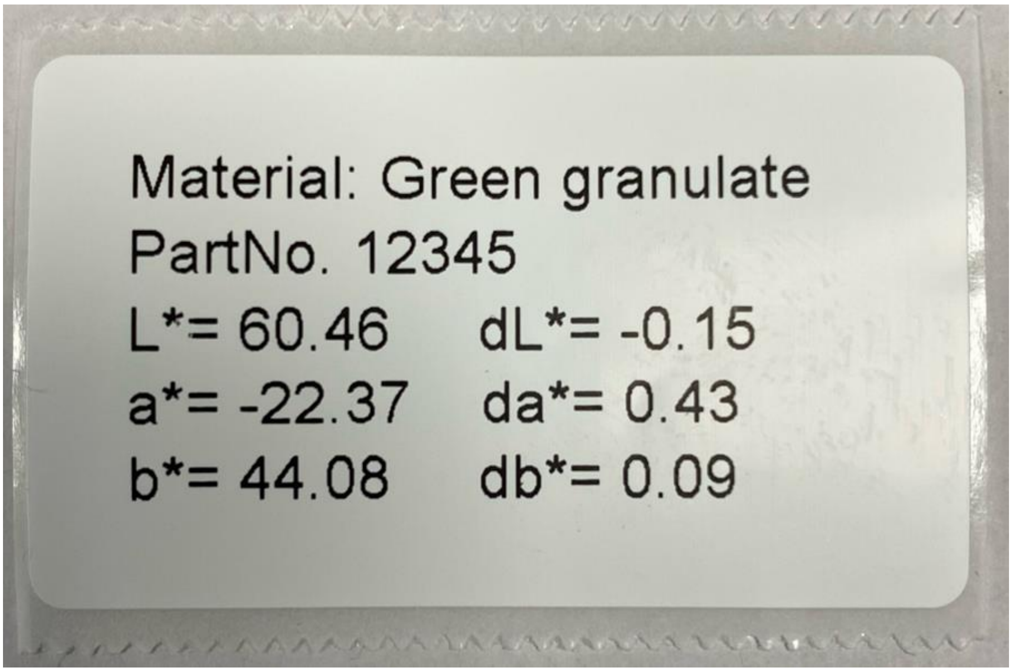
Now start the **SPECTRO-T-3-DOCAL-Scope** and establish the connection to the sensor.
In the **DOCU** tab, some data records can now be recorded.
The **PRINTOUT LINE** fields can also be written.



Then press → **PRINT DOCUMENTATION DATA**.
Select the label printer **"CAB EOS2/300"** and confirm with **OK**.



A successful printout looks like this, for example:
Two labels are printed out. One for L*a*b* and one for N*i*r*.



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® 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 ‘Install’ 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.

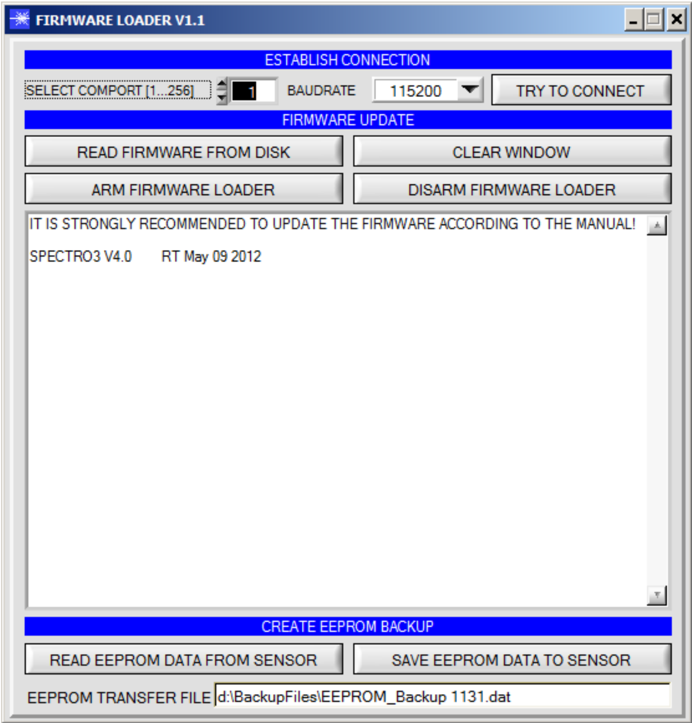
Windows® is a trademark of the Microsoft Corp.
VGA™ is a trademark of the International Business Machines Corp.

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:

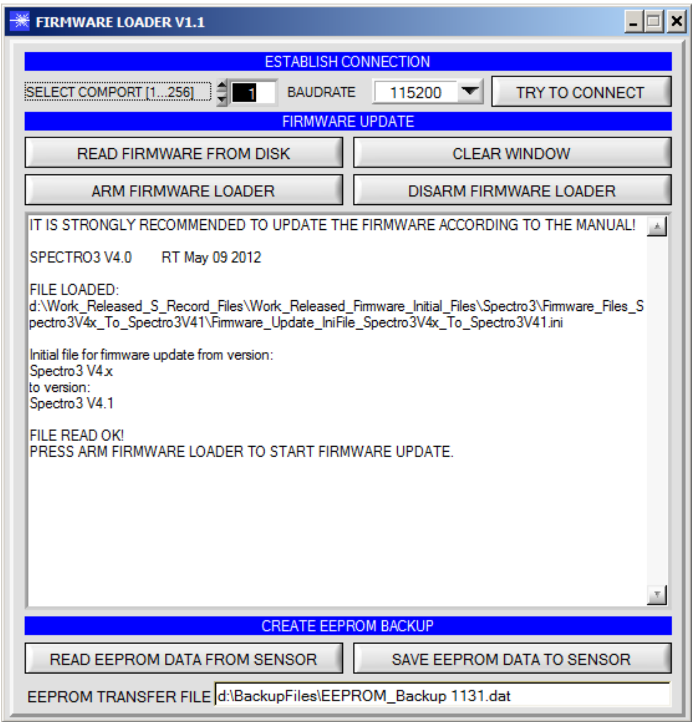
When the Firmware Loader software is started, this window opens on the Windows® user interface.



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.

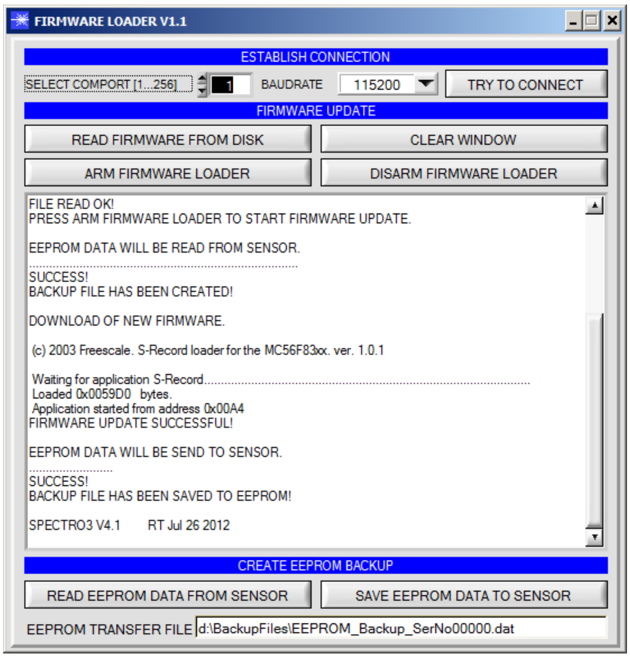
When you press the **ARM FIRMWARE LOADER** button the firmware update will be performed automatically.

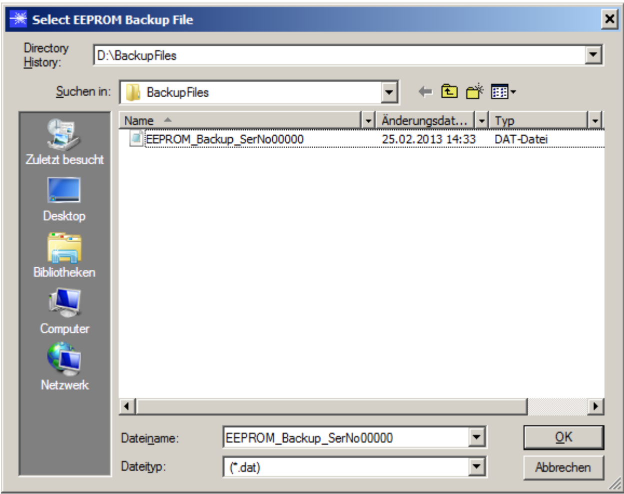
In the course of the update process you will only be prompted to enter a name for the EEPROM backup file. If the firmware update should run perfectly until the EEPRM data are read out, but should then go wrong for any reason whatsoever, the EEPROM backup file can always be written back with **SAVE EEPROM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

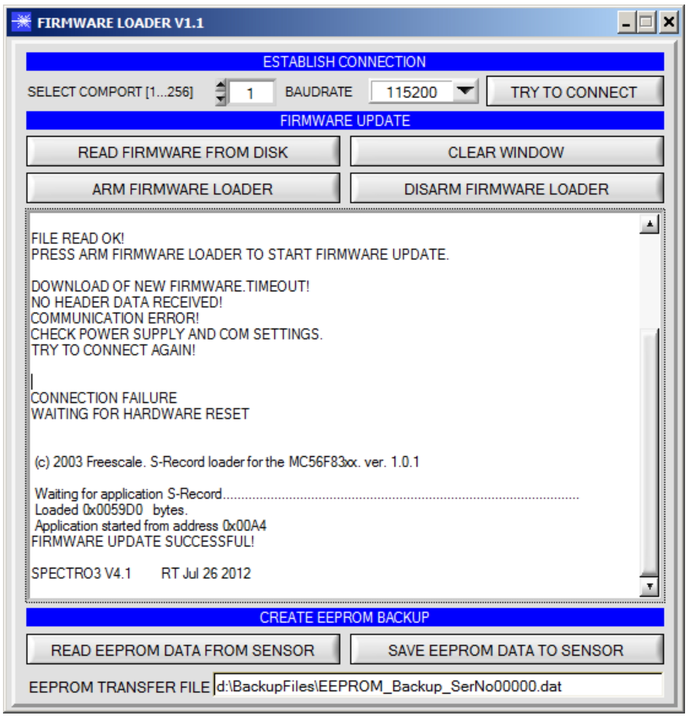




If, contrary to expectations, there should be any trouble with the update of the program memory, it will still be possible to perform an update, even though it may look like the sensor was "killed".
Please make sure that you have selected the correct **COM PORT** and the correct **BAUDRATE**.
You will 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.

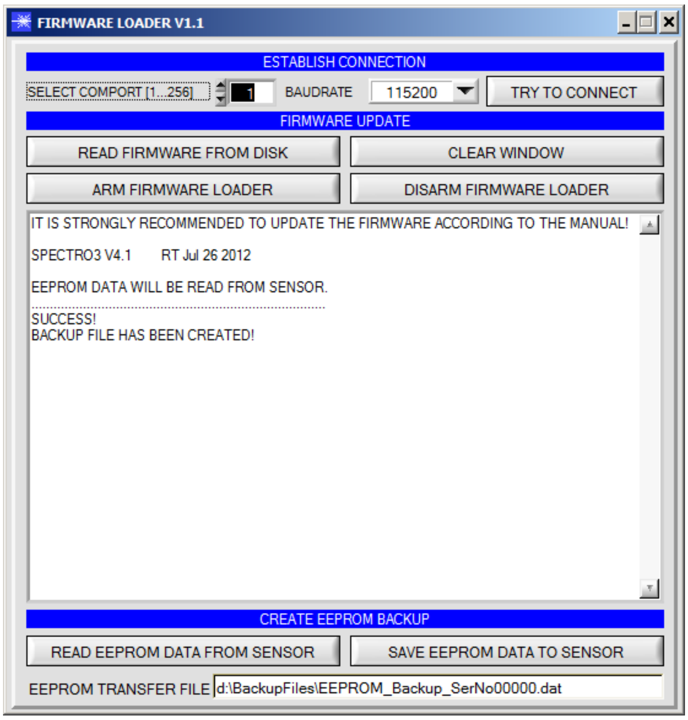


INFO! In case that the sensor was "killed", the sensor will work with a BAUDRATE of 115200.

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.