

# Manual

## Software SPECTRO3-MSM-ANA-Scope V3.1

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

**for color sensors of SPECTRO-3-MSM-ANA Series**  
**with internal temperature compensation and white light balancing**

This manual describes the installation of the PC software for the SPECTRO-3-MSM-ANA color sensor. As a support for commissioning of the color sensor this manual explains the functional elements of the Windows® user interface.

The SPECTRO-3-MSM-ANA color sensor provides highly flexible signal acquisition. For example, the sensor can be operated in alternating-light mode (AC mode), which makes the sensor insensitive to extraneous light. It also can be set to constant-light mode (DC mode), which makes the sensor extremely fast. With the stepless adjustment of the integrated light source as well as the selectable gain of the receiver signal and an INTEGRAL function the sensor can be set to almost any surface or any "self-luminous object".

When the integrated light source of the SPECTRO-3-MSM-ANA color sensor is activated, the sensor detects the radiation that is diffusely reflected from the object. As a light source the SPECTRO-3-MSM-ANA color sensor uses LEDs (in case of color sensor type -VISUV additional UV LEDs) with adjustable transmitter power. An integrated 3-fold receiver for the red, green, and blue content of the light that is reflected from the object, or the light that is emitted by a "self-luminous object", is used as a receiver.

The SPECTRO-3-MSM-ANA color sensor can be "taught" up to 3 colors. For each of these taught colors it is possible to set tolerances. Evaluation of the taught colors is performed either with the "FIRST HIT" or with the "BEST HIT" mode. With "FIRST HIT" the first hit in the teach table is output, and with "BEST HIT" the best hit in the teach table is output. Raw data are represented with 12 bit resolution.

Color detection either operates continuously or is started through an external PLC trigger signal. The respective detected color either is provided as a binary code at the 2 digital outputs or can be sent directly to the outputs, if only up to 2 colors are to be detected.

In addition to the 2 digital outputs there also are 3 analog outputs that either provide the red, green, and blue components or the calculated color coordinates in values of 0...+10V or 4...20mA.

Through input IN0 (green wire of the cab-las8/SPS cable) the color sensor can be taught up to 3 colors. For this purpose the corresponding parameter mode must be set with the software.

Parameters and measurement values can be exchanged between a PC and the SPECTRO-3-MSM-ANA color sensor through the serial RS232 interface. All the parameters for color detection also can be saved to the non-volatile EEPROM of the SPECTRO-3-MSM-ANA color sensor through this serial RS232 interface. When parameterisation is finished, the color 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 sensors of the SPECTRO-3-MSM-ANA series can be calibrated. Calibration can be performed to any surface with known color coordinates. A ColorChecker™ table is available as an alternative. This table has 24 color fields according to the CIE STANDARD that can be used for calibration.

Furthermore, input IN0 can be used for performing white light balancing.

## 0 Contents

|   | Page |
|---|------|
| 1. Installation of the SPECTRO3-MSM-ANA-Scope software .....                      | 3    |
| 2. Operation of the SPECTRO3-MSM-ANA-Scope software.....                          | 4    |
| 2.1 Tab CONNECT .....   | 5    |
| 2.2 Tab PARA1, button SEND, GET, GO, STOP (parameterization, data exchange) ..... | 7    |
| 2.3 Graphic display elements .....  | 17   |
| 2.4 Tab TEACH (teach table) .....   | 19   |
| 2.5 Tab GEN (color rows and color specifications) .....                           | 23   |
| 2.6 Tab REC (data recording) .....  | 24   |
| 2.7 Tab CALIB.....  | 26   |
| 2.7.1 Calibration .....   | 26   |
| 2.7.2 Offset calibration .....  | 29   |
| 2.8 Tab SCOPE .....   | 30   |
| 2.9 Tab XYZOFF .....  | 31   |
| 3. Operation of the TEMPCOMP-Scope software .....                                 | 32   |
| 4. External triggering of the SPECTRO-3-MSM-ANA color sensor .....                | 33   |
| 5. Connector assignment of the SPECTRO-3-MSM-ANA color sensor.....                | 34   |
| 6. RS232 communication protocol.....  | 35   |
| A. Firmware update via software Firmware Loader .....                             | 47   |

**Shortcuts:**

|      |     |
|------|-----|
| SEND | F9  |
| GET  | F10 |
| GO   | F11 |
| STOP | F12 |

## 1. Installation of the SPECTRO3-MSM-ANA-Scope software

The following requirements must be fulfilled for successful installation of the SPECTRO3-MSM-ANA-Scope 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*** for USB slot or ***cab-4/ETH***

Please install the software as described below:

1. The software can be installed directly from the installation CD-ROM. To install the software, start the SETUP program in the SOFTWARE folder of the CD-ROM.  
As an alternative the software also can be downloaded from our homepage.
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.

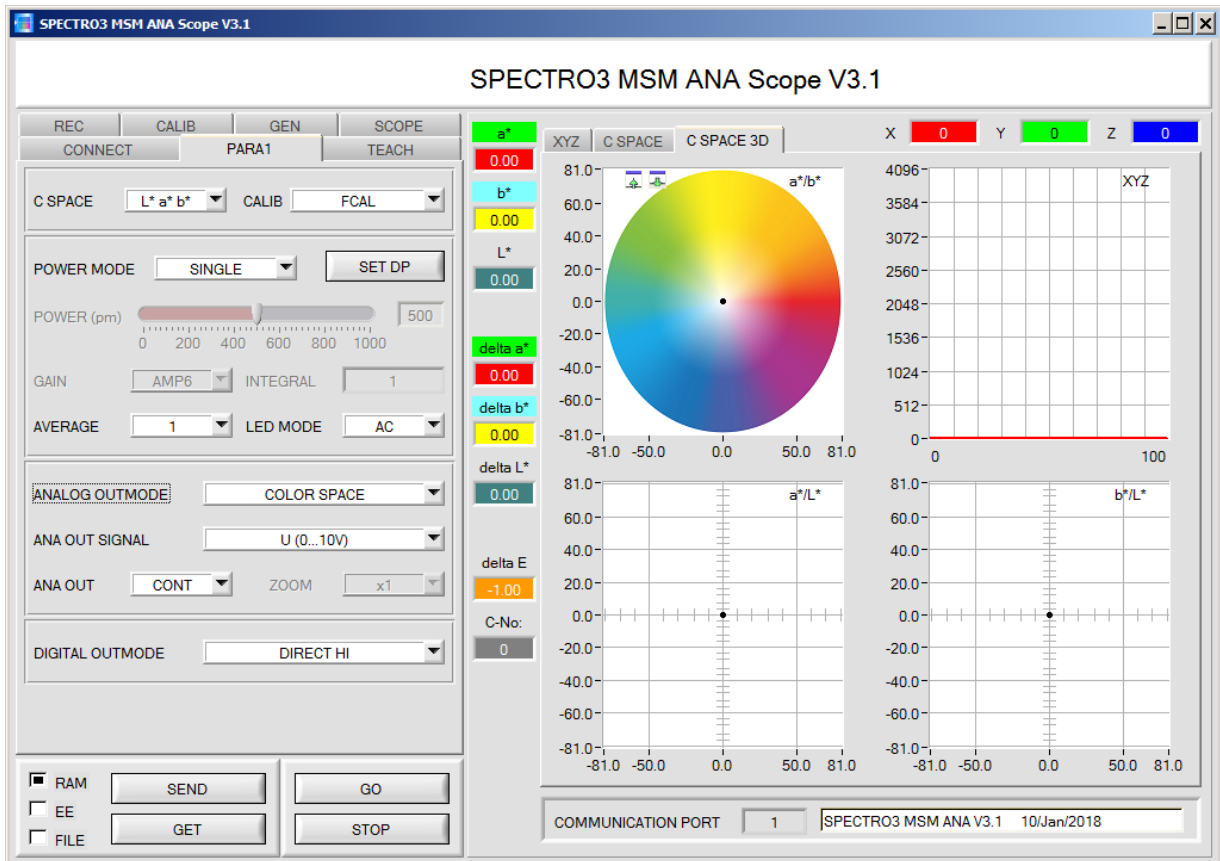
Windows® is a trademark of the Microsoft Corp.

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

## 2. Operation of the SPECTRO3-MSM-ANA-Scope software

Please read this chapter first before you start to adjust and parameterise the SPECTRO-3-MSM-ANA color sensor.

When the SPECTRO3-MSM-ANA-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.**

## 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 adaptor 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 adaptor (**cab-4/ETH**) is needed if the sensor should communicate through a local network. With this adaptor a connection to the sensor can be established using the **TCP/IP** protocol.

Parameterisation of the **cab-4/ETH** adaptor (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 adaptor, 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).

COMMUNICATION PROTOCOL: RS232

SELECT COM PORT [1...256]: 1

SELECT BAUDRATE: 115200

TRY TO CONNECT DISCONNECT

Try to connect!  
Firmware Version  
Sensor: Type  
Serial number: xxx

ACCEPT SETTINGS CHANGE BAUDRATE GEN. HW INFO FILE

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.

SELECT BAUDRATE: 19200

CHANGE BAUDRATE

Now a new baudrate can be selected under **SELECT BAUDRATE**.

A click on **CHANGE BAUDRATE** sends the new baudrate information to the sensor.

Firmware Version  
Sensor: Type  
Serial number: xxx

Try to change baudrate!  
Success!  
Try to connect!  
Firmware Version  
Sensor: Type  
Serial number: xxx

ATTENTION!!! Confirm new baudrate with EEPROM and SEND!

ACCEPT SETTINGS CHANGE BAUDRATE GEN. HW INFO FILE

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.

File has been saved!  
d:\MistHardwareInitialFile.ini

ACCEPT SETTINGS CHANGE BAUDRATE GEN. HW INFO FILE

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

### 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:

After a click on the **SEND** button the current parameters are written into the **RAM** memory of the sensor, or they are read from the **RAM** by clicking on the **GET** button, i.e. these parameters are lost when the voltage at the sensor is switched off.

### EEPROM:

After a click on the **SEND** button the current parameters are written into the non-volatile memory of the **EEPROM** in the sensor, or they are read from the **EEPROM** by clicking on the **GET** button, i.e. the parameters in the internal **EEPROM** are stored when the voltage at the sensor is switched off.

### 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.

### GO [F11]:

A click on this button (or with shortcut key button F11) starts data transfer from the sensor to the PC through the serial RS232 interface.

**SOURCE** is used to select which signals should be shown in the displays and graphs.

### STOP [F12]:

A click on this button (or with shortcut key button F12) stops data transfer from the sensor to the PC through the serial RS232 interface.



The screenshot shows a control panel with the following elements:

- POWER MODE:** A dropdown menu set to 'SINGLE' and a 'SET DP' button.
- POWER (pm):** A horizontal slider bar ranging from 0 to 1000. The slider is currently positioned at 500, which is also displayed in a text box to the right of the slider.
- GAIN:** A dropdown menu set to 'AMP6' and a text box containing the value '1'.
- AVERAGE:** A dropdown menu set to '1'.
- LED MODE:** A dropdown menu set to 'AC'.

**POWER MODE:** see SET DP

**POWER:**

The transmitter power is constantly kept at the value set with the **POWER [pm]** slider (recommended operation mode). 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.

**POWER** only is available with **CALIB=OFF** or **XYZ OFFSET**

**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 in 8 different gain stages (AMP1 to AMP8). **GAIN** should be set such that with a medium **POWER** value the sensor operates in its dynamic range (X,Y,Z between 2750 and 3750). In **AC** mode, **GAIN** directly influences the scan frequency. The current scan frequency is displayed in the **SCOPE** tab.

**GAIN** only is available with **CALIB=OFF** or **XYZ OFFSET**.

**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

**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.

**GAIN** only is available with **CALIB=OFF** or **XYZ OFFSET**.

**INFO:**

The **POWER** slider, **GAIN** and **INTEGRAL** only are effective with **CALIB = OFF** and **POWER MODE = SINGLE** or with **XYZ OFFSET**.

Otherwise the sensor operates with the values set in calibration.



## SET DP:

If an application involves both very bright and very dark surfaces, and the sensor is set such that it is not in saturation at the bright surfaces, the dark surfaces frequently provide only a very low signal. Little variations with a very low signal lead to a large change in color space calculation (**C-SPACE**). To minimise this effect, the sensor in **PMODE = DOUBLE** automatically switches back and forth between 2 **DOUBLE PARAMETER sets**. Set 2 must be parameterised such that the signal is maximally amplified. When the signal is too high (saturation), it automatically switches to set 1. To get correct color space coordinates with set 2, a correction factor must be used for the signal. The two **DOUBLE PARAMETER sets** must be set at the sensor in advance. When you click on **SET DP** (Set Double Parameter) a window will open at the right side, where you can set two sets. A Double Parameter set is defined by **POWER, GAIN and INTEGRAL**.

The screenshot shows the 'SET DP' window in the IPF ELECTRONIC software. The window is titled 'SET DP' and contains several sections for configuring the double parameter sets. On the left, there are tabs for REC, CALIB, GEN, SCOPE, CONNECT, PARA1, and TEACH. The main area is divided into several sections: 'DOUBLE PARAMETER (DP)' with a table for POW, GAIN, and INT; 'XYZ VALUES' with a table for BRIGHT DP1, DARK DP1, and DARK DP2; 'CORRECTION VALUES' with a table for X, Y, and Z; and a 'DOUBLE PARAMETER SET' display showing '0'. The left sidebar contains various settings like C SPACE, POWER MODE, POWER (pm), GAIN, AVERAGE, ANALOG OUTMODE, ANA OUT SIGNAL, ANA OUT, DIGITAL OUTMODE, and RAM/EE/FILE checkboxes. The bottom right shows a vertical bar with red, green, and blue segments.

## Automatic setting of DOUBLE parameter sets:

Pressing **START ASSISTANT** starts a menu guidance for automatically setting of the Double parameter sets. Just follow the instructions.

## Manual setting of DOUBLE parameter sets:

Set **CALIB=OFF** and **POWER MODE=SINGLE**.

Place the brightest surface before the sensor, and select suitable **POWER, GAIN and INTEGRAL** values so that the brightest channel is at approx. 3000 digits.

Then click on **ASSIGN 1** to write the power parameters to row 1 in the **PP** table.

The X Y Z values are entered in the **XYZ VALUES** table.

Now place the darkest surface before the sensor.

The dark surface should be such that the lowest channel provides a signal higher than 100 digits.

Click on **ASSIGN 2**, and the X Y Z values are written to the DARK PP1 row in the **XYZ VALUES** table.

Now set **POWER, GAIN and INTEGRAL** such that the highest value is at approx. 3000.

Click on **ASSIGN 3** to write the power parameters to row 2 in the **PP** table, and the X Y Z values to the **XYZ VALUES** table.

The correction values are calculated automatically and are written to the **CORRECTION VALUES** table.

## INFO!

The power parameter sets only become active when you click on **SEND**!

The tables only are used to display the values. They cannot be edited.

The **DOUBLE PARAMETER SET** display shows which DOUBLE PARAMETER set the sensor currently uses.

When the display shows 0, the sensor operates with the settings in the **PARA1** tab.



### C SPACE:

Selection of a color space (**COLOR SPACE**) for the calculation of the color stimulus specification.

### x y Y:

The chromaticity values (chromaticity coordinates) **x**, **y** are calculated from the tristimulus values X, Y, Z. **x** and **y** define the color, and **Y** is used as a brightness reference (lightness variable).

|                          |  |
|--------------------------|--|
| Lightness variable       | $Y = \frac{Y}{4096}$   |
| Chromaticity coordinates | $x = \frac{X}{X + Y + Z} \quad y = \frac{Y}{X + Y + Z}$          |
| Color difference         | $\Delta E^* = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta Y)^2}$ |

### 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 | $a^* = 500 \left[ \left( \frac{X}{X_n} \right)^{1/3} - \left( \frac{Y}{Y_n} \right)^{1/3} \right] \quad b^* = 200 \left[ \left( \frac{Y}{Y_n} \right)^{1/3} - \left( \frac{Z}{Z_n} \right)^{1/3} \right]$ |
| Color difference         | $\Delta E^* = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2}$  |

### L\*C\*h\*:

The L\*C\*h\* color system is based on the same diagram as the L\*a\*b\* system, but polar coordinates are calculated here. In the L\*C\*h\* color system it is not possible to determine teach vectors, which is why **DIGITAL OUTMODE = OFF** is set and disabled.

|                    |   |
|--------------------|---|
| Lightness variable | $L^* = 116 \left( \frac{Y}{Y_n} \right)^{1/3} - 16$ |
| Chroma             | $C^* = \sqrt{(a^*)^2 + (b^*)^2}$                    |
| Hue angle          | $h^* = \tan^{-1} \frac{b^*}{a^*}$                   |

### L\*u\*v\* and L\*u'v\*:

The L\*u\*v\* color space (also referred to as the CIELUV 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 | $u^* = 13L^*(u' - u_n')$ $v^* = 13L^*(v' - v_n')$<br>$u' = \frac{4X}{X + 15Y + 3Z} = \frac{4x}{3 - 2x + 12y}$ $v' = \frac{9Y}{X + 15Y + 3Z} = \frac{9y}{3 - 2x + 12y}$<br>$u_n' = \frac{4X_n}{X_n + 15Y_n + 3Z_n} = \frac{4x_n}{3 - 2x_n + 12y_n}$ $v_n' = \frac{9Y_n}{X_n + 15Y_n + 3Z_n} = \frac{9y_n}{3 - 2x_n + 12y_n}$ |
| Color difference         | $\Delta E^* = \sqrt{(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2}$  |

|                                    |  |
|------------------------------------|--|
| X Y Z                              | Tristimulus values XYZ of the specimen.  |
| Xn Yn Zn                           | Tristimulus values of a perfect reflecting diffuser.                                 |
| $\Delta L^* \Delta a^* \Delta b^*$ | Difference in L*, a*, and b* values between the specimen color and the target color. |
| $\Delta L^* \Delta u^* \Delta v^*$ | Difference in L*, u*, and v* values between the specimen color and the target color. |
| u' v'                              | Chromaticity coordinates from the CIE 1976 UCS diagram.                              |
| u'n v'n                            | Chromaticity coordinates of a perfect reflecting diffuser.                           |

CALIB

## CALIB:

The sensors are factory-calibrated.

**CALIB=OFF** can be selected if the sensor should operate in "uncalibrated" mode. This can be done if the sensor should operate fast. Both **FCAL** and **UCAL** will slow down the sensor.

If the **CALIB** parameter is set to **FCAL** (Factory **CAL**ibration), the sensor operates with this factory calibration.

As an alternative the user can himself calibrate the sensor to different surfaces (User **CAL**ibration).

If **CALIB=UCAL** is selected, the sensor operates with the calibration made by the user.

If you select **CALIB=UCAL WB** or **FCAL WB**, white light balancing can be performed through input IN0. For this purpose the reference surface used for calibration must be placed before the sensor, and input IN0 must then be activated.

The calculated white light calibration factors are displayed in the **CALIB FACTORS OF WHITE BALANCE** table in the **SENSOR DATA** tab.

To get the white light calibration factors for **CALIB=UCAL WB**, press **GET CF** in the **CALIB** tab.

To get the white light calibration factors for **CALIB=FCAL WB**, double-click with the right mouse button on the frame in which **GET CF** is contained, and enter GETCF as password.

With **CALIB = XYZ OFFSET** and **XYZ OFFSET IN0** you can determine an offset for X Y Z.

See tab **XYZOFF**.

ANALOG OUTMODE

ANA OUT SIGNAL

ANA OUT  ZOOM

## ANALOG OUTMODE:

This function field is used to determine the signals that the sensor provides at its analog outputs.

**OFF:** No analog signal is output.

## XYZ:

The tristimulus values **X**, **Y**, **Z** are acquired with 12 bit resolution.

The signals therefore may have values between 0 and 4095.

At the respective analog outputs these values are provided in a 0...+10V or 4...20mA range.

## COLOR SPACE:

The color coordinates of the selected **C SPACE** are scaled to 12 bit and are output as a voltage from 0 to 10 V at the respective analog outputs.

Depending on the color coordinate, the analog output range is divided as follows:

x y Y from 0 to +1 (0V=0, 10V=1)

L\* from 0 to +100 (0V=0, 10V=100)

a\* from -150 to +150 (0V=-150, 10V=150)

u\* from -788 to +724 (0V=-788, 10V=724)

u' from 0 to +4 (0V=0, 10V=4)

C\* from 0 to +538 (0V=0, 10V=538)

b\* from -150 to +150 (0V=-150, 10V=150)

v\* from -154 to +164 (0V=-154, 10V=164)

v' from 0 to +0,6 (0V=0, 10V=0,6)

h\* from 0 to 360 (0V=0, 10V=360)

## CS REF:

When input IN0 is set to HI, the current **C SPACE** color coordinates are saved as 5 V reference values.

This means that when the respective current **C SPACE** color value is equal to the reference value, 5 V will be output at the corresponding analog output. If the current value deviates from the reference value, this deviation is output at the corresponding analog output starting from 5 V and amplified by the **ZOOM** factor.

**Example:** **ZOOM** = x4, **a\*(reference)** = 60, **a\*(current)** = 70 → deviation = 10 (1 digit is approx. 2.44mV)

Output: 5 V+(10 digit \* 32 (zoom) \* 2.44 mV) = approx. 5.78 V.

The reference values are displayed in the graph window under the **C SPACE** tab.

Since the sensor only has one hardware input (IN0) this input possibility is not always available. If, for example, you select **ANA OUT = IN0 L→H**, the input already is used for a controlled output of the analog signal and is no longer available for **CS REF**.

Function fields that are not available will either be hidden or deactivated.

## ANA OUT SIGNAL:

Here you set whether the analog signal should be output as voltage (0 ... 10V) or current (4 ... 20mA).

SHAPE MODE

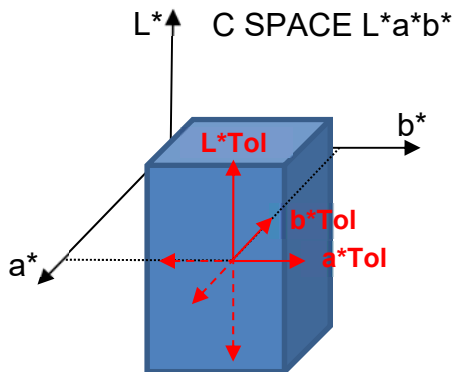
## SHAPE MODE:

This function field is used to specify whether the color space coordinates (e.g.  $L^*$   $a^*$   $b^*$ ) should be viewed as a block, cylinder, or sphere in space.

When viewed as a block, each color space coordinate has its own tolerance ( $a^*Tol$ ,  $b^*Tol$ ,  $L^*Tol$ ).

The color distance **delta E** is calculated from the color coordinates ( $a^*$   $b^*$ ).

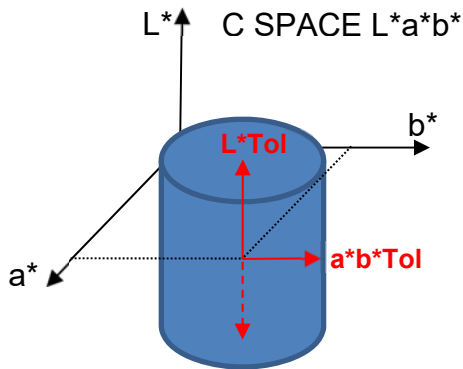
A color is recognised if the current color space coordinates lie in the color block.



When viewed as a cylinder, the color coordinates ( $a^*$   $b^*$ ) have their own tolerance ( $a^*b^*Tol$ ), and the brightness value ( $L^*$ ) has its own tolerance ( $L^*Tol$ ).

The color distance **delta E** is calculated from the color coordinates ( $a^*$   $b^*$ ).

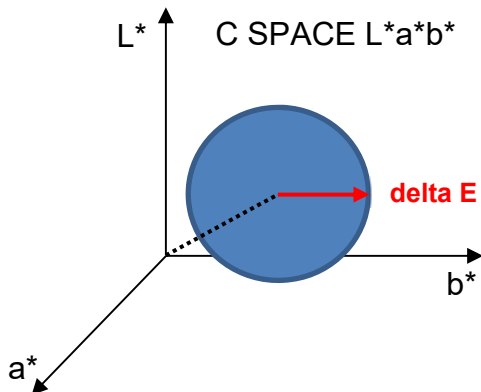
A color is recognised if the current color space coordinates lie in the color cylinder.



In sphere mode there only is one tolerance for the color space coordinates.

The color distance **delta E** is calculated from the color space coordinates ( $L^*$   $a^*$   $b^*$ ).

A color is recognised if the current color space coordinates lie in the color sphere.



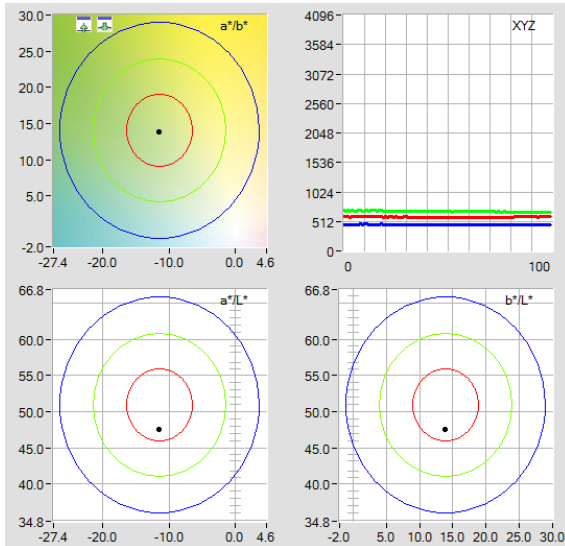
EVALUATION MODE

## EVALUATION MODE:

In this function field the evaluation mode can be set at the SPECTRO-3-MSM-ANA color sensor.

In the text below the **SHAPE MODE "SPHERE"** is used to explain the evaluation modes **FIRST HIT** and **BEST HIT**.

|   | a*     | b*    | L*    | deltaE |      |      |  |
|---|--------|-------|-------|--------|------|------|--|
| 0 | -11.35 | 13.99 | 50.85 | 5.00   | 0.00 | 0.00 |  |
| 1 | -11.35 | 13.99 | 50.85 | 10.00  | 0.00 | 0.00 |  |
| 2 | -11.35 | 13.99 | 50.85 | 15.00  | 0.00 | 0.00 |  |



## FIRST HIT:

The currently measured color values are compared with the default values in the **TEACH TABLE**, starting with teach-color 0. If in the row-by-row comparison the current color values correspond with the teach-parameters entered in the TEACH TABLE, this first "hit" in the TEACH TABLE is displayed as a color number (**C-No.**) and is output at the digital outputs (OUT0 and OUT1) according to the settings of the **OUTMODE** parameter (see **OUTMODE**).

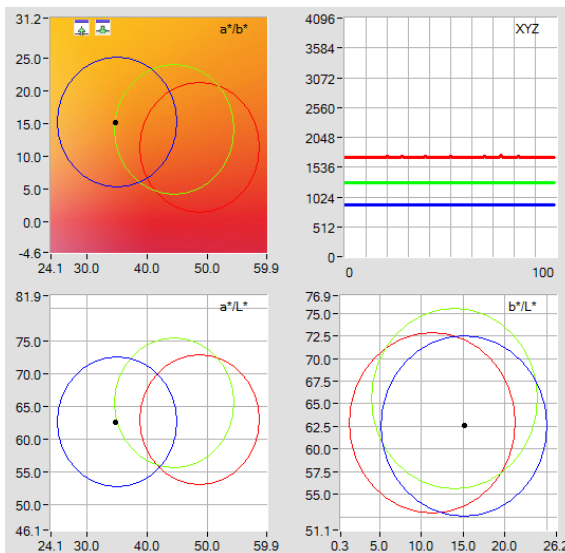
If the current color does not correspond with any of the teach-colors, the color code **C-No.** = 255 will be set ("error status").

**Tip!** This mode is used if only one color is taught, and this color must be checked to avoid "drifting away". With the rising tolerance windows this can be detected very well, and countermeasures can be initiated, if necessary.

**Tip!** Values can be entered in a cell of the table either by double-clicking on the respective cell, or by highlighting the respective cell and then pressing F2.

**Tip!** The teach colors are only activated when the **SEND** button is pressed!

|   | a*    | b*    | L*    | deltaE |      |      |  |
|---|-------|-------|-------|--------|------|------|--|
| 0 | 48.89 | 11.34 | 62.84 | 10.00  | 0.00 | 0.00 |  |
| 1 | 44.65 | 14.01 | 65.48 | 10.00  | 0.00 | 0.00 |  |
| 2 | 35.10 | 15.17 | 62.53 | 10.00  | 0.00 | 0.00 |  |



## BEST HIT:

The currently measured color values are compared with the default values in the **TEACH TABLE**, starting with teach-color 0. If in the row-by-row comparison the current color values correspond with several of the teach parameters entered in the color table, the teach parameter that has the shortest distance from the current color value will be a hit.

This "hit" in the TEACH TABLE is displayed as a color number (**C-No.**) and is output at the digital outputs (OUT0 and OUT1) according to the settings of the **OUTMODE** parameter (see **OUTMODE**).

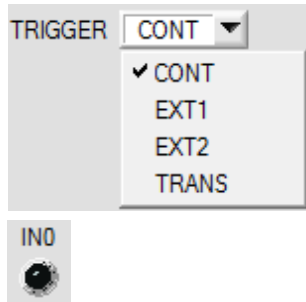
If the current color does not correspond with any of the teach-in colors, the color code **C-No.** = 255 will be set ("error status").

**Tip!** This mode is used if several colors must be separated from each other, and only certain surface fluctuations are allowed.

**Tip!** Since what you are looking for here with several "hits" is the shortest distance of the current color to the centres of the taught colors, the individual tolerance windows (circles) may overlap here. The sensor detects the "best hit".

**Tip!** Values can be entered in a cell of the table either by double-clicking on the respective cell, or by highlighting the respective cell and then pressing F2.

**Tip!** The teach colors are only activated when the **SEND** button is pressed!



## TRIGGER:

This function field serves for setting the trigger mode at the sensor. If **TRIGGER** is not **CONT**, the LED **IN0** shows a trigger event.

**CONT:** Continuous color detection (no trigger event required).

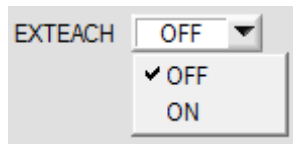
**EXT1:** Color detection is started through the external trigger input (IN0 pin3 green of cable cab-las8/SPS). A trigger event is recognized as long as +24V is present at the IN0 input (HIGH-active).

After the trigger input goes to LOW again, the state (C-No.) that was last detected will be held at the outputs.

**EXT2:** Same behaviour as in EXT1 mode, with the difference that an error state (color no. = 255) will be output after the trigger input goes to LOW again.

## TRANS:

Same behaviour as in **CONT**. However, in contrast to **CONT** in this mode the transmitter source will be switched off if no trigger event (IN0) is given.



## EXTEACH:

In all the evaluation modes teaching of a color can be performed externally through IN0.

**OFF:** The external TEACH feature is deactivated.

**ON:** The sensor can be taught up to 3 colors through IN0.

In **BEST HIT** evaluation mode teaching through IN0 can be performed to every individual row in the **TEACH TABLE**. With **FIRST HIT** the currently present color is taught to all the rows that are activated with **MAXCOL-No.**

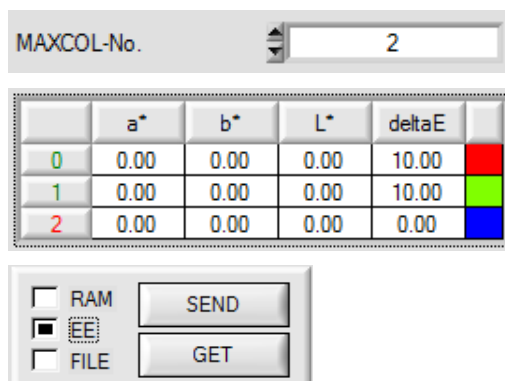
Select the function **EXTERN TEACH = ON**.

Set the power such that the sensor is not overdriven and that the signal also is not too low.

Select how many colors you want to teach externally.

Click on the **TEACH** tab to change to the **TEACH TABLE**.

Then enter the respective tolerances for the colors you want to teach.



In this example **MAXCOL-No.** = 2 was selected. This means that the sensor should find the color information that is saved in the first 2 rows of the **TEACH TABLE** by way of external teaching through IN0. Since the sensor cannot calculate its **deltaE** tolerances itself, these values must be entered once (in this example 10 in all fields) and together with **MAXCOL-No.** and **EXTEACH = ON** must be saved in the **EEPROM**.

Then select **EE** in the function field, and click on **SEND**.

From now on, the PC is no longer necessary, as long as you only wish to teach colors up to the **MAXCOL-No.** and do not want to change the tolerances.

## INFO:

Of course the taught colours can be viewed at any time with the PC.

Colors that are taught are stored in the **EEPROM** of the sensor, i.e. the information will not be lost when the system is turned off.

This function rather is intended for "after-teaching", because the sensor does not automatically adjust its **POWER** value.

## EXTEACH=ON:

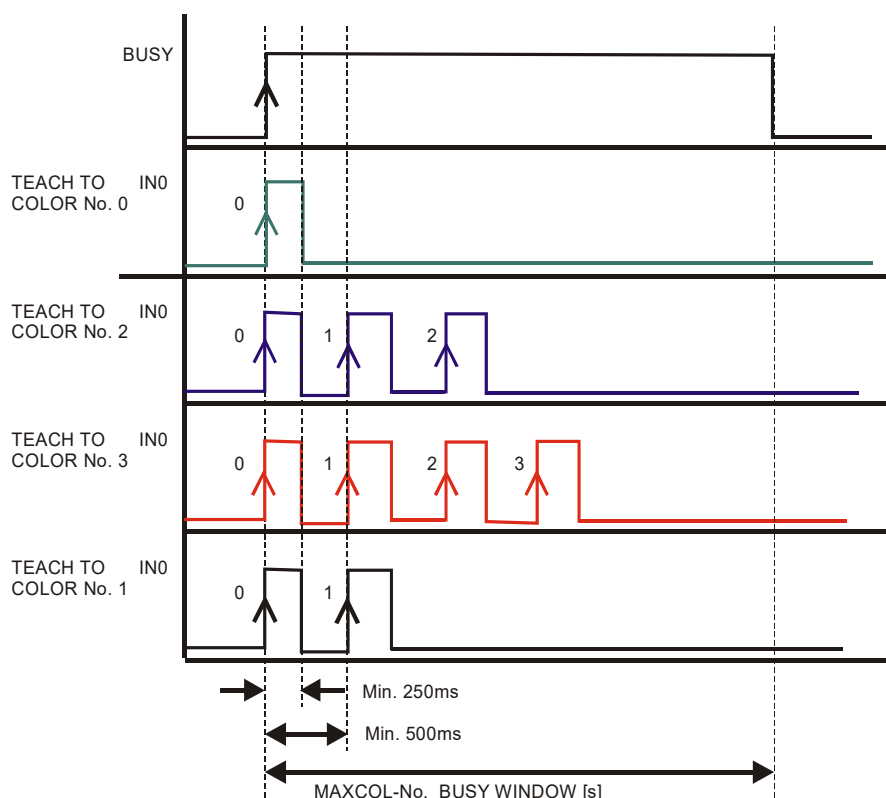
Before the external teach-process can be started, the color to be taught must be presented to the sensor. The external teach process is started with a positive edge at IN0 (green wire). The outputs OUT0 and OUT1 alternately change their state between 0V and 24V. From now on, the user has a certain number of seconds time to inform the sensor about the position at which the color information (**X Y INT** or **siM**) should be placed in the **TEACH TABLE**. The **BUSY** window is the time that the user has for informing the sensor about the teach rows. It depends on the value that is set in **MAXCOL-No.** (e.g. approx. 2 seconds when **MAXCOL-No.** = 2).

The first positive edge (start edge 0) selects position 0 in the **TEACH TABLE**. Every additional positive edge selects one position higher (see table below).

Example:

If you wish to save the current color to position 3 in the **TEACH TABLE**, the following steps have to be performed:

1. Start the external TEACH process with a positive edge (0) at IN0 → Position 0 is selected.
2. Another positive edge (1) selects position 1 in the TEACH TABLE.
3. Another positive edge (2) selects position 2 in the TEACH TABLE.
4. Another positive edge (3) selects position 3 in the TEACH TABLE.
5. The desired position is now selected.
6. When the **BUSY** window is over, the sensor starts the evaluation.
7. To teach another color → go to pos. 1





INTLIM

**INTLIM:**

An intensity limit can be set in this edit box. If the current intensity  $(X+Y+Z)/3$  arriving at the receiver unit falls below this limit, color evaluation will no longer be performed, and an error state will be

output.

**Attention: ERROR STATE if:  $(X+Y+Z)/3 < \text{INTLIM}$**

MAXCOL-No.

**MAXCOL-No.:**

This function field serves for setting the number of colors to be checked.

With **OUTMODE = BINARY** a maximum of 3 colors can be taught.

With **OUTMODE = DIRECT HI** or **DIRECT LO** a maximum of 2 colors can be taught.

The numerical value set here determines the currently possible scanning rate of the color sensor. The less the colors to be checked, the faster the operation of the sensor. The numerical value set here refers to the number of rows (starting with row 0) in the **TEACH TABLE** (→ 2.4 Tab TEACH TABLE).

DIGITAL OUTMODE

**DIGITAL OUTMODE:**

This group of buttons offers the method of how to control the 2 digital outputs.

**OFF:**

No digital signal is output. All the corresponding function fields are hidden.

**DIRECT:**

In this mode the maximum number of colors to be taught is 2.

If in this row-by-row comparison the current color values correspond with the teach-in parameters entered in the TEACH TABLE, this color in the TEACH TABLE is displayed as a color number (C-No.) and is sent **direct** to the digital outputs (OUT0 and OUT1).

**DIRECT HI:**

If **DIRECT HI** is activated, the respective digital output is set to HIGH. If the current color does not correspond with any of the teach-in colors, all digital outputs are set to LOW.

**DIRECT LO:**

If **DIRECT LO** is activated, the specially digital output is set to LOW, while the other ones are set to HIGH. If the current color does not correspond with any of the teach-in colors, all digital outputs are set to HIGH.

**BINARY HI:**

If in the row-by-row comparison the current color values correspond with the teach-parameters entered in the color table, this "hit" in the color table is displayed as a color number (C-No.) and is provided at the digital outputs (OUT0 and OUT1) as a **bit pattern**.

A maximum of 3 colors can be taught.

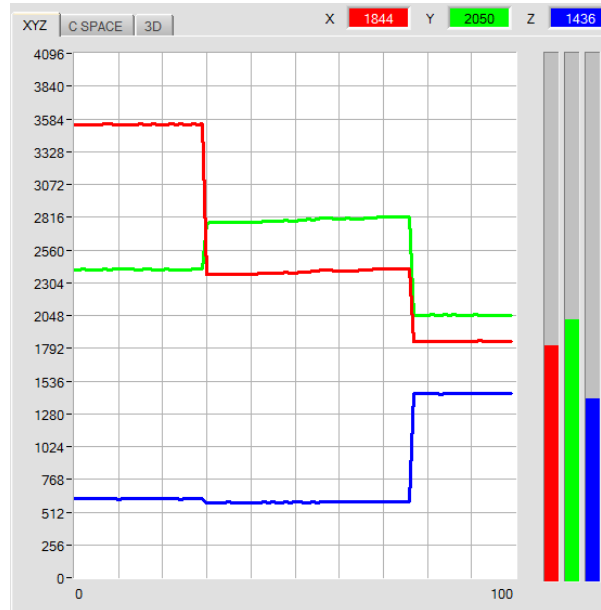
**BINARY LO:**

Same function as **BINARY HI**, but the **bit pattern** is output inversely.

## 2.3 Graphical display elements

### Tab **XYZ**:

Display of the current raw signals (tristimulus values) **X**, **Y**, **Z** of the 3-fold receiver.



### Tab **C SPACE**:

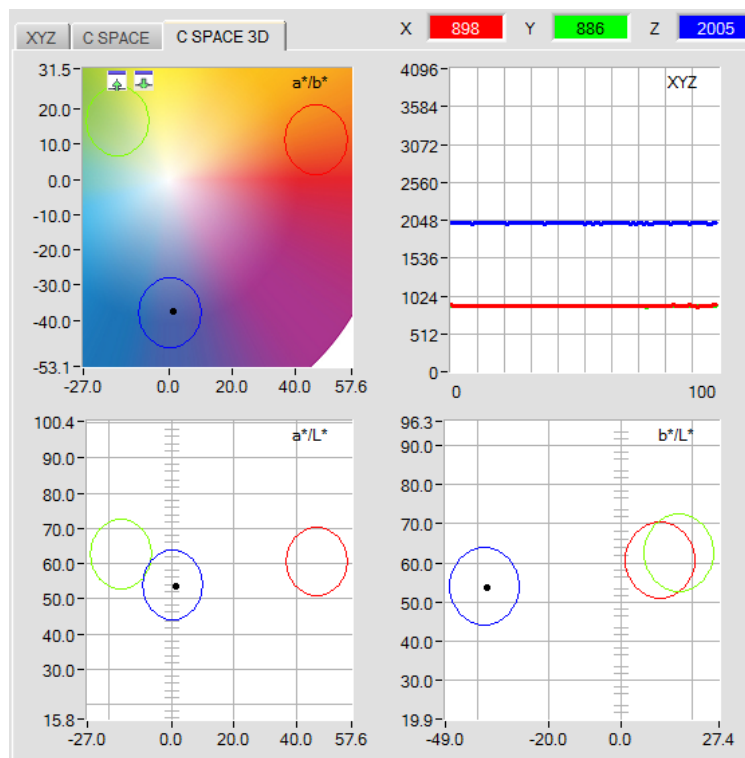
Line graph display of the color coordinates of the color space selected under **C SPACE**.



### Tab **C SPACE 3D**:

Three-side view of the taught color in space.

The taught color space coordinates with their tolerances and the current color position are displayed. A three-side view in space was chosen to provide a clearer representation.



A left double-click in one of the graphs causes all the circles to be displayed in grey, only the circle with the recognised color is displayed in color.

A single left click returns to the standard view.

A left double-click in the XYZ graph starts an automatic zoom function.

A single left click closes this zoom function again

The graph with the color chart in the background can be zoomed in and out.

|        |        |        |       |        |
|--------|--------|--------|-------|--------|
| a*     | u*     | u'     | C*    | x      |
| -12.98 | -17.22 | 0.3196 | 35.17 | 0.2645 |
| b*     | v*     | v'     | h*    | y      |
| -8.18  | -15.34 | 0.4736 | 14.43 | 0.3029 |
| L*     | L*     | L*     | L*    | Y      |
| 67.63  | 54.01  | 50.10  | 32.31 | 0.2007 |

These numerical value displays show the color coordinates and brightness values that are calculated from the tristimulus values **X, Y, Z**.  
Calculation is performed as described in **C SPACE**

|          |          |         |
|----------|----------|---------|
| delta a* | delta u* | delta x |
| 0.10     | 0.01     | 0.0004  |
| delta b* | delta v* | delta y |
| 0.08     | 0.15     | 0.0097  |
| delta L* | delta L* | delta Y |
| 0.05     | 0.06     | 0.0007  |

These numeric value displays show the deviations of the individual color coordinates and of the brightness value of the current surface from the "color hit" (C-No.) in the teach table.  
The values are calculated on the PC user interface and not in the sensor, and are only shown in these displays.

|         |
|---------|
| delta E |
| 0.12    |

## delta E:

This display shows the color distance from a color hit.

Calculation is performed as described in **C SPACE**.

In **FIRST HIT** evaluation mode **delta E** is equal to the distance from a color hit. If there was no color hit, **delta E** is calculated in relation to the last valid color in the color table (depending on **MAXCOL-No.**)

In **BEST HIT** evaluation mode **delta E** also is equal to the distance from a color hit. However, if there was no color hit, **delta E** has the value -0.00.

This also applies if the current intensity  $(X+Y+Z)/3$  falls below the **INTLIM** parameter.

|       |
|-------|
| C-No: |
| 0     |

## C-No.:

This numerical value output field shows the currently detected color number according to the entry in the **TEACH TABLE**. The currently detected color number is sent to the digital outputs OUT0 and OUT1 as a corresponding bit pattern.

A value of 255 means that none of the taught colors was detected.

A double-click on the display will open a larger display window.



## IN0:

This LED display visualises the status of input IN0.

If 0 V is applied at the input, the LED is black.

If 24 V is applied at the input, the LED is green.

This display only is visible if the input is used.

|      |    |
|------|----|
| TEMP | 27 |
|------|----|

## TEMP:

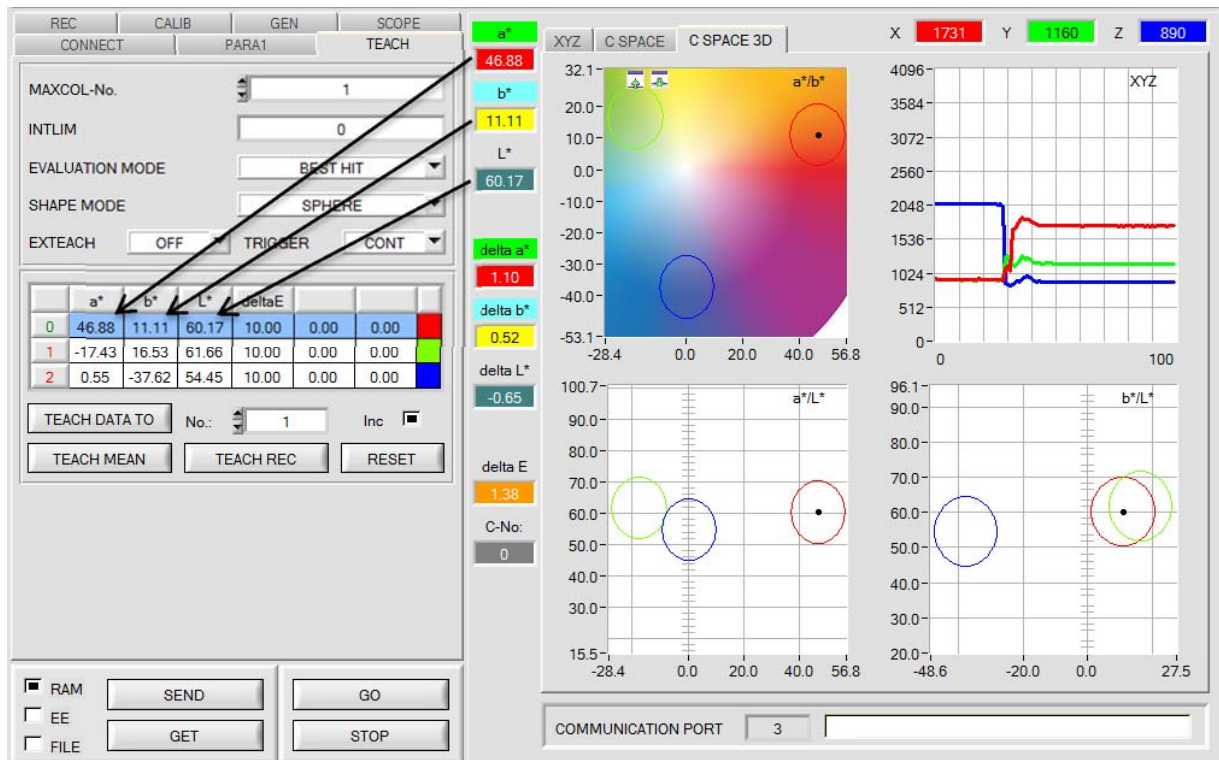
This display shows the temperature in the sensor housing.

The display does **NOT** show degrees Celsius or Fahrenheit.

|                     |   |
|---------------------|---|
| <b>Please note:</b> | <b>The above-mentioned output fields are only updated when data transfer between PC and sensor is active (GO button pressed).</b> |
|---------------------|---|

## 2.4 Tab TEACH

When the **GO** button is pressed, data transfer from the sensor to the PC is started. The tristimulus values **X**, **Y**, **Z** and the calculated color coordinates and brightness values are shown in the displays and in the graphic windows.



MAXCOL-No.

INTLIM

EVALUATION MODE

SHAPE MODE

EXTTEACH  TRIGGER

|   | a*     | b*     | L*    | deltaE |      |      |
|---|--------|--------|-------|--------|------|------|
| 0 | 46.88  | 11.11  | 60.17 | 10.00  | 0.00 | 0.00 |
| 1 | -17.43 | 16.53  | 61.66 | 10.00  | 0.00 | 0.00 |
| 2 | 0.55   | -37.62 | 54.45 | 10.00  | 0.00 | 0.00 |

TEACH DATA TO No.:  Inc ☐

TEACH MEAN TEACH REC RESET

## TEACH:

Click on the **TEACH** tab to open a window where you can teach colors to the **TEACH TABLE**.

**Please note: The teach colors must be sent to the sensor by clicking on SEND.**

The **TEACH TABLE** is organised in rows, i.e. the individual parameters for the teach colors are listed side by side in the respective row.

The sensor can work with up to 3 teach colors. The number of the respective teach color is shown in the left column of the table.

Only rows with a green number are used for evaluation by the sensor. The number of rows to be used is defined with **MAXCOL-No.**

After a click on **TEACH DATA TO** the currently displayed color coordinates and the brightness value are written to the row in the **TEACH TABLE** that is selected under **No.:**.

A color tolerance (**delta E**) also is defined. If necessary, the tolerances and the teach values can be changed by double-clicking on the respective field with the left mouse button (or by pressing F2) and then entering a numerical value with the PC keyboard.

When **Inc:** is activated, and the **TEACH DATA TO** button is pressed, the **No.:** input field is automatically incremented (increased) by 1, i.e. the next row in the **TEACH TABLE** is selected.

A click on the **RESET TABLE** button resets the **TEACH TABLE** (RESET value = 0).

|   | a*     | b*    | L*    | deltaE |      |      |
|---|--------|-------|-------|--------|------|------|
| 0 | 50.99  | 57.89 | 73.46 | 12.00  | 0.00 | 0.00 |
| 1 | -23.44 | 62.78 | 75.59 | 12.00  | 0.00 | 0.00 |
| 2 | -8.64  | 13.44 | 66.56 | 12.00  | 0.00 | 0.00 |

It is possible to edit several cells at the same time.

For this purpose the cells that should be changed must be selected.

Help

Set selection to

Increment selection with

Reset selection

Teach tolerance off

Teach tolerance on

Right-click with the mouse to open a pop-up menu.

Use **Set selection to** if you wish to fill several cells in the **TEACH TABLE** with a certain value.

Use **Increment selection with** if you wish to enter increasing tolerance values.

The upper left cell is the starting value.

Starting from this cell the following cells will be increased by the value that is entered in the pop-up menu.

Use **Reset selection** to set the selected cells to 0.

When **TEACH DATA TO** is used the software suggests tolerance values and writes these values to the **deltaE** column.

This function can be switched on and off with **Teach tolerance off** and **Teach tolerance on**.

A click on **Help** will open a pop-up window that provides explanations of the individual functions.

VALUE!

Insert a cell value!

OK

The following panel will be displayed after a click on **TEACH MEAN**.

REC CALIB GEN SCOPE  
CONNECT PARA1 TEACH

MAXCOL-No. 1  
INTLIM 0  
EVALUATION MODE BEST HIT  
SHAPE MODE SPHERE  
EXTTEACH OFF TRIGGER CONT

|   | a*    | b*    | L*    | deltaE |      |      |
|---|-------|-------|-------|--------|------|------|
| 0 | 46.78 | 11.03 | 60.08 | 10.00  | 0.00 | 0.00 |
| 1 | 0.00  | 0.00  | 0.00  | 0.00   | 0.00 | 0.00 |
| 2 | 0.00  | 0.00  | 0.00  | 0.00   | 0.00 | 0.00 |

TEACH DATA TO No.: 1 Inc. ☐  
TEACH MEAN TEACH REC RESET

RAM SEND GO  
EE GET STOP  
FILE

COUNTER 16  
TEACH TO TABLE

|    | a*    | b*    | L*    |
|----|-------|-------|-------|
| 1  | 46.57 | 11.00 | 60.09 |
| 2  | 47.30 | 10.99 | 60.00 |
| 3  | 47.36 | 10.99 | 60.03 |
| 4  | 46.42 | 11.17 | 60.11 |
| 5  | 46.75 | 11.06 | 60.05 |
| 6  | 45.90 | 11.15 | 60.20 |
| 7  | 46.26 | 11.04 | 60.11 |
| 8  | 46.97 | 10.84 | 60.05 |
| 9  | 47.73 | 10.72 | 59.98 |
| 10 | 47.58 | 10.94 | 59.98 |
| 11 | 47.73 | 10.94 | 59.98 |
| 12 | 45.90 | 11.28 | 60.20 |
| 13 | 46.88 | 11.13 | 60.09 |
| 14 | 46.88 | 11.00 | 60.09 |
| 15 | 45.99 | 11.11 | 60.17 |
| 16 | 46.33 | 11.07 | 60.13 |
| 17 | 0.00  | 0.00  | 0.00  |
| 18 | 0.00  | 0.00  | 0.00  |
| 19 | 0.00  | 0.00  | 0.00  |
| 20 | 0.00  | 0.00  | 0.00  |
| 21 | 0.00  | 0.00  | 0.00  |
| 22 | 0.00  | 0.00  | 0.00  |
| 23 | 0.00  | 0.00  | 0.00  |

MEAN 46.78 11.03 60.08

delta E of captured table data 1.00

CLOSE

After a click on **GO**, data are automatically obtained from the sensor and displayed.

A click on the **CAPTURE** button enters a parameter frame in the table.

The **COUNTER** display field shown how many frames have already been recorded.

The last frames that were entered in the table can be deleted again by pressing **UNDO**.

**RESET TABLE** sets the whole table back to 0.

After every activation of **CAPTURE**, **UNDO**, or **RESET TABLE**, the mean values for the individual parameters are calculated automatically and are displayed in the mean value table.

Furthermore, a maximum color tolerance **delta E** is formed with respect to the mean values.

When the **TEACH TO TABLE** button is pressed, the respective mean values are taught to the row in the **TEACH TABLE** that is selected under **No.:**.

A click on the **CLOSE** button will take you back to the main panel.

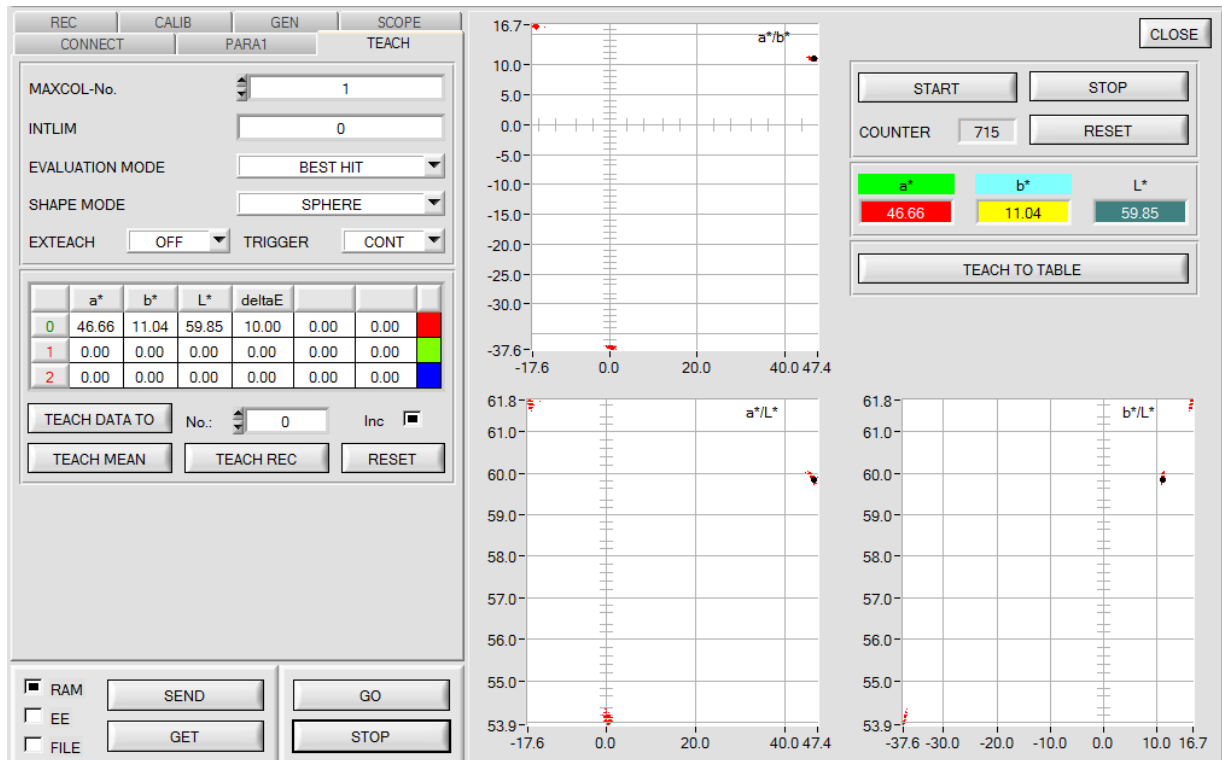
A click on the button **TEACH REC VAL** (teach recorded values) opens a panel which, after a click on the **START** button, starts to record data and display them in the three graphs. This function is useful if the material to be detected cannot be directly placed in front of the sensor, for example because it is transported on a conveyor belt that cannot be stopped or cannot be exactly stopped at the position required for teaching. After some time, area centres will become apparent that can be assumed to be the objects to be taught. The screenshot below shows 3 of these positions.

After a click on **STOP** the cursor can then be positioned on such an area centre in any of the graphs. The cursor will be automatically positioned in the other two graphs.

With **TEACH TO TABLE** the current cursor position can then be transferred to the **TEACH TABLE**. Teaching will be performed to the row that is set under **No.:**.

**RESET** can be used to reset the graphs and the counter.

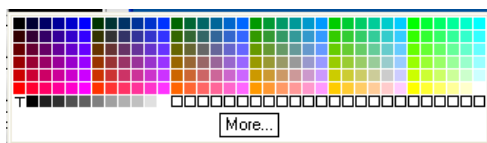
**CLOSE** takes you back to the main panel.





## 2.5 Tab GEN

| SPECIFICATION |         |
|---------------|---------|
| 0             | Color 0 |
| 1             | Color 1 |
| 2             | Color 2 |



### GEN:

A click on **GEN** opens a display of the current temperature value **TEMP** in the sensor housing.

The display is **NOT** in degrees Celsius or Fahrenheit.

In the GEN tab you also can manually select the row colors in which the individual tolerance circles are displayed, or let the system set these colors automatically based on the colors detected by the system.

Furthermore, designations for the individual rows can be entered in the **SPECIFICATION** table. These designations are saved on the PC's hard disk and can be loaded again, if necessary.

If **SET ROW COLORS** is set to **MANUALLY**, the row color that should be changed must be set under **SELECT ROW**.

A click on the colored area of **ROW COLOR** opens a color palette where the desired color can be selected.

After a click on the **SET** button, the color will be displayed in the 6th column and the selected row of the **TEACH TABLE**.

If **SET ROW COLORS** is set to **AUTOMATICALLY**, the system will automatically calculate the corresponding row color, display the color in a color display beside the graph and, after a click on **TEACH DATA TO**, automatically insert it in the corresponding row.

The **SAVE TO FILE** and **GET FROM FILE** functions allow you to save specified row color arrays and the **SPECIFICATION** table on the hard disk, or to load previously saved arrays.

**SAVE TO SENSOR** and **GET FROM SENSOR** writes these to the sensor or read these from the sensor.

A click on **RESET** resets the colors to a default value.

When the software is started again, the **ROW COLOR** table and the **SPECIFICATION** table that were active when the software was closed are automatically loaded again.

## 2.6 Tab REC

The SPECTRO3-MSM-ANA-Scope software includes a data recorder that makes it possible to save the data acquired and calculated by the sensor with a time stamp. The recorded file is saved to the hard disk of the PC and can then be evaluated with a spreadsheet program.

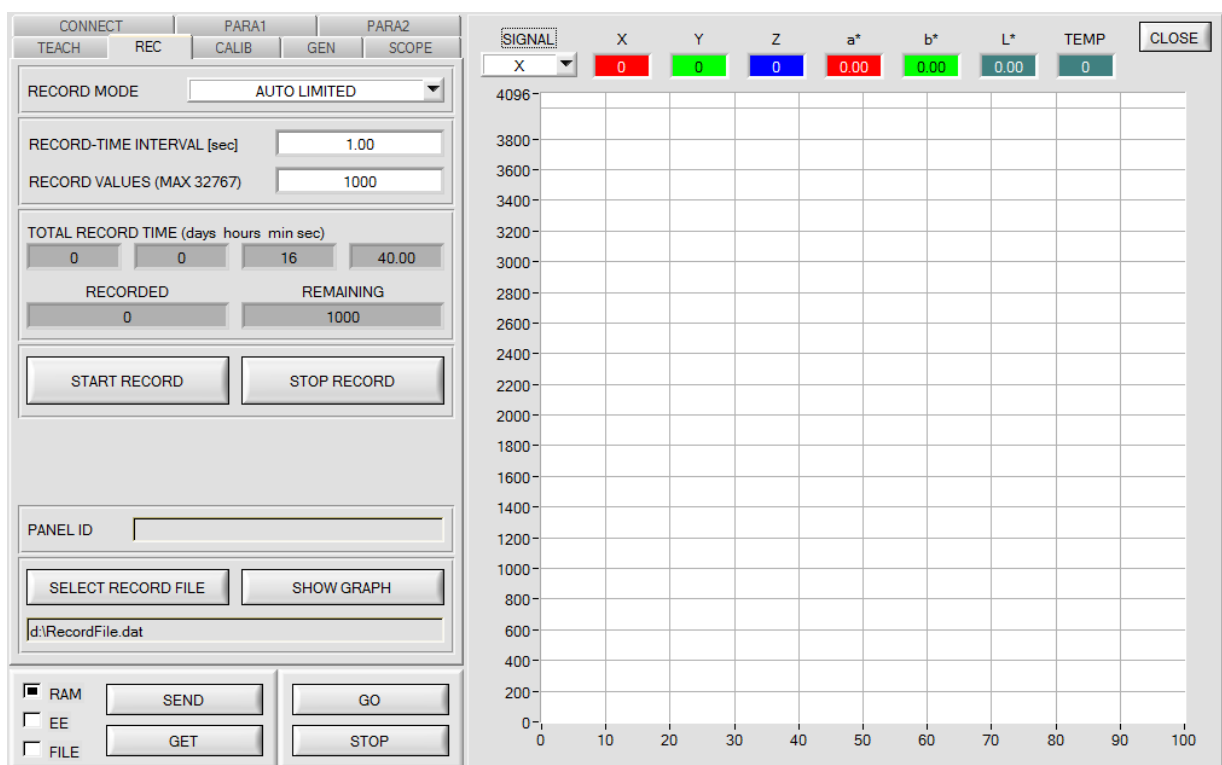
The following steps describe how data frames are recorded with the recorder:

### Step 1:

When the **REC** 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.



## Step 2:

If you want to automatically record several data frames, please select **AUTO LIMITED** under **RECORD MORE**. 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 MORE**. Then select the desired recording interval and press **START RECORD**.

If you want to record data "manually", select the **MANUAL RECORDING** function under **RECORD MORE**.

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 **AUTO TRIGGERED** is selected under **RECORD MORE**, and **TRIGGER = EXT1, EXT2, or TRANS** is selected, the sensor will automatically send a data frame after each drop of the trigger when **START RECORD** is pressed. This data frame is captured and recorded by the recorder.

Pressing the **STOP RECORD** button terminates the automatic sending function of the sensor again.

## Please note:

When you press **START RECORD**, the file that is selected under **SELECT RECORD FILE** will be deleted. With **RECORD FRAME MANUALLY**, the file will be created if it does not already exist. If the file already exists, the data are added to the existing file.

## 2.7 Tab CALIB

### 2.7.1 Calibration

The sensors are factory-calibrated. If the **CALIB** parameter is set to **FCAL** (Factory **CAL**ibration), the sensor operates with this factory calibration.

As an alternative the user can himself calibrate the sensor to different surfaces (**User CAL**ibration).

If **CALIB=UCAL** is selected, the sensor operates with the calibration made by the user.

Calibration is performed by using different surfaces with known color coordinates and brightness values. A ColorChecker™ table is available as an alternative. This table has 24 color fields according to the CIE STANDARD.

A click on **CALIB** will open the following window:

The screenshot shows a software window for calibration. It has a top menu bar with 'CONNECT', 'PARA1', and 'PARA2'. Below this is a sub-menu bar with 'TEACH', 'REC', 'CALIB', 'GEN', and 'SCOPE'. The 'CALIB' tab is active.

On the left side, there are three main sections:

- PARAMETER**: A text box with the instruction 'Make sure that you have adjusted the sensor with proper parameter values.'
- REFERENCE**: A section with a 'COLOR SPACE OF REF SETVALUE' dropdown set to 'L\* a\* b\*', an 'ASSIGN ACTUAL XYZ VALUE OF REF' button, and a 'SETVALUE TABLE' section with a 'C SPACE' dropdown set to 'L\* a\* b\*', a 'ROWS' input set to '1', and buttons for 'CALC CF', 'SEND CF', and 'GET CF'.
- ACTUAL XYZ VALUE TABLE**: A section with an 'ASSIGN TO ROW' button, a 'No.' input set to '1', an 'Inc' button, and the same 'CALC CF', 'SEND CF', and 'GET CF' buttons.

At the bottom left, there are checkboxes for 'RAM', 'EE', and 'FILE', and buttons for 'SEND', 'GET', 'GO', and 'STOP'.

The main area on the right is divided into three tabs: 'CALIB REF', 'CALIB DATA', and 'SENSOR DATA'. The 'CALIB REF' tab is active.

Under 'CALIB REF', there are two tables:

- REFERENCE SETVALUE**: A table with columns L\*, a\*, and b\*. All values are 0.0000.
- ACTUAL XYZ VALUE OF REF**: A table with columns X SI, Y SI, and Z SI. All values are 0.

Below these tables, there is a 'CALCULATED XYZ OF REF' table with columns X, Y, and Z. All values are 0.0000.

To the right of the 'CALCULATED XYZ OF REF' table, there is a table for 'Xn Yn Zn' with columns Xn 95,05, Yn 100, and Zn 108,9. All values are 4096.

At the bottom of the 'CALIB REF' tab, there is a text box with the instruction 'Make sure that you have adjusted the sensor with proper parameter values.' and a section for 'Recommended settings: PMODE=DOUBLE (be aware that you have set proper double parameter sets) LED MODE = AC'.

On the far right, there is a 'CLOSE' button and a vertical bar with three colored segments: red, green, and blue.

**REFERENCE**

COLOR SPACE OF REF SETVALUE L\* a\* b\*

ASSIGN ACTUAL XYZ VALUE OF REF

**REFERENCE SETVALUE**

| L*     | a*     | b*     |
|--------|--------|--------|
| 0.0000 | 0.0000 | 0.0000 |

**CALCULATED XYZ OF REF**

| X      | Y      | Z      |
|--------|--------|--------|
| 0.0000 | 0.0000 | 0.0000 |

**ACTUAL XYZ VALUE OF REF**

| X SI | Y SI | Z SI |
|------|------|------|
| 0    | 0    | 0    |

Xn Yn Zn

| Xn 95,05 | Yn 100 | Zn 108,9 |
|----------|--------|----------|
| 4096     | 4096   | 4096     |

Please perform the following steps in sequence:

- Place a white reference surface before the sensor.
- Click on **GO** to start data exchange between sensor and PC software.
- **POWER**, **GAIN**, and **INTEGRAL** must be set such that the highest signal of **RAW X, Y, Z** is approx. 3000 digits. It is important to select the correct color space **C SPACE OF REF SETVALUE**.
- The color coordinates and the brightness value of the reference surface must be known and must be entered in the **REFERENCE SETVALUE** table. The corresponding tristimulus values **X, Y, Z** are calculated automatically and are written to the **CALCULATED XYZ OF REF** table.
- With **ASSIGN ACTUAL XYZ VALUE OF REF** the tristimulus values **RAW X, Y, Z** acquired by the sensor are written to the **ACTUAL XYZ VALUE OF REF** table, and the tristimulus values **Xn, Yn, Zn** of a completely matt white body are calculated automatically.
- **Xn, Yn, Zn** are needed for calculating the color coordinates and the brightness value in almost all color spaces.

Use the file and disk symbol if you want to write the **REFERENCE** settings to a file or read them from a file. A click on the reset symbol will reset the tables.

**SETVALUE TABLE**

C SPACE L\* a\* b\* ROWS 24

**SETVALUE TABLE**

|    | L*      | a*       | b*       |
|----|---------|----------|----------|
| 1  | 38.0800 | 12.0900  | 14.3900  |
| 2  | 66.3800 | 13.2200  | 17.1400  |
| 3  | 51.0600 | 0.3800   | -22.0600 |
| 4  | 43.3000 | -16.5200 | 21.4600  |
| 5  | 56.3600 | 12.8400  | -25.2900 |
| 6  | 71.6000 | -30.7100 | 1.1700   |
| 7  | 61.7000 | 27.5400  | 58.2300  |
| 8  | 41.2200 | 17.9500  | -43.1600 |
| 9  | 51.5700 | 43.0000  | 14.7500  |
| 10 | 30.7700 | 25.7400  | -23.3800 |
| 11 | 72.5800 | -28.1700 | 58.6100  |
| 12 | 71.6000 | 12.4500  | 66.5800  |

Several different surfaces are needed for sensor calibration.




The number of surfaces must be entered in **ROWS**.

The color coordinates and brightness values must be known.

These must be entered in the **SETVALUE TABLE**.

The corresponding color space must be selected under **C SPACE**.

Use the file and disk symbol if you want to write the **SETVALUE** settings to a file or read them from a file. A click on the reset symbol will reset the tables.

**ACTUAL XYZ VALUE TABLE**   

**ASSIGN TO ROW** No.:  Inc ☐

Now the surfaces corresponding with the **SETVALUES** must be placed before the sensor.

With **ASSIGN TO ROW** the tristimulus values **RAW X, Y, Z** that are currently determined by the sensor are written to the row in the **ACTUAL XYZ VALUE TABLE** that is selected under **No.:**.

**ACTUAL XYZ VALUE TABLE**

|    | X    | Y    | Z    |
|----|------|------|------|
| 1  | 394  | 345  | 248  |
| 2  | 1290 | 1224 | 913  |
| 3  | 641  | 760  | 1173 |
| 4  | 389  | 454  | 237  |
| 5  | 853  | 911  | 1488 |
| 6  | 1166 | 1633 | 1492 |
| 7  | 1313 | 929  | 293  |
| 8  | 467  | 518  | 1338 |
| 9  | 901  | 607  | 524  |
| 10 | 280  | 248  | 495  |

Use the file and disk symbol if you want to write the **ACTUAL XYZ VALUE TABLE** to a file or read it from a file. A click on the reset symbol will reset the tables.

**CALC CF** **SEND CF** **GET CF**

With a click on the **CALC CF** button the corresponding calibration factors that are determined from the tables **SETVALUE TABLE** and **ACTUAL XYZ VALUE TABLE** are calculated.

**SEND CF** saves the **ACTUAL XYZ VALUE TABLE** and the calculated calibration factors in the sensor's EEPROM.

A click on **GET CF** will display the tables saved in the sensor and the reference parameters in the **SENSOR DATA** tab.

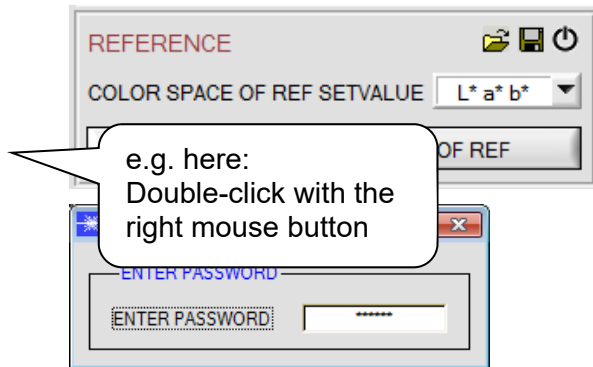
With **CALIB=UCAL WB** or **FCAL WB** white light balancing can be performed through input IN0. For this purpose the reference surface used for calibration must be placed before the sensor, and input IN0 must then be activated. The calculated white light calibration factors are displayed in the **CALIB FACTORS OF WHITE BALANCE** table in the **SENSOR DATA** tab.

To get the white light calibration factors for **CALIB=UCAL WB**, press **GET CF** in the **CALIB** tab.

To get the white light calibration factors for **CALIB=FCAL WB**, double-click with the right mouse button on the frame in which **GET CF** is contained, and enter GETCF as password.

## 2.7.2 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.

Make sure that the Receiver of the sensor is covered!  
Push **CALCULATE OFFSET** to detect the offset.  
Push **SEND OFFSET** to update the sensor.

|                           |      |
|---------------------------|------|
| OFFSET X [ Default 3100 ] | 3077 |
| OFFSET Y [ Default 3100 ] | 3078 |
| OFFSET Z [ Default 3100 ] | 3085 |

**CALCULATE OFFSET**

**SEND OFFSET**

**GET OFFSET**

**CLOSE**

### 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. with a black cloth that is impervious to light.

**This is absolutely necessary for proper offset calculation.**

Now press **CALCULATE OFFSET**. The offset values for red, green, and blue should be approximately 3080 plus/minus 40.

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.



## 2.8 Tab SCOPE

The SCOPE tab visualises an oscilloscope.

Depending on **TRIG MODE**, the display shows the **XYZ** signals, the color space, the output analog signal (depending on the **ANA OUT** settings), and the status of the digital outputs.

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]**. For determining the correct scan frequency please give the sensor a time of 8 seconds before you click on **GET CYCLE TIME**.

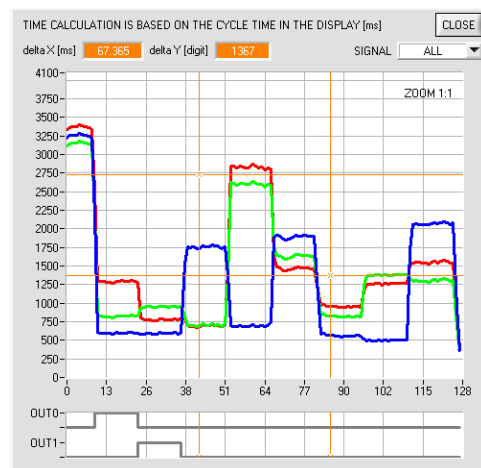
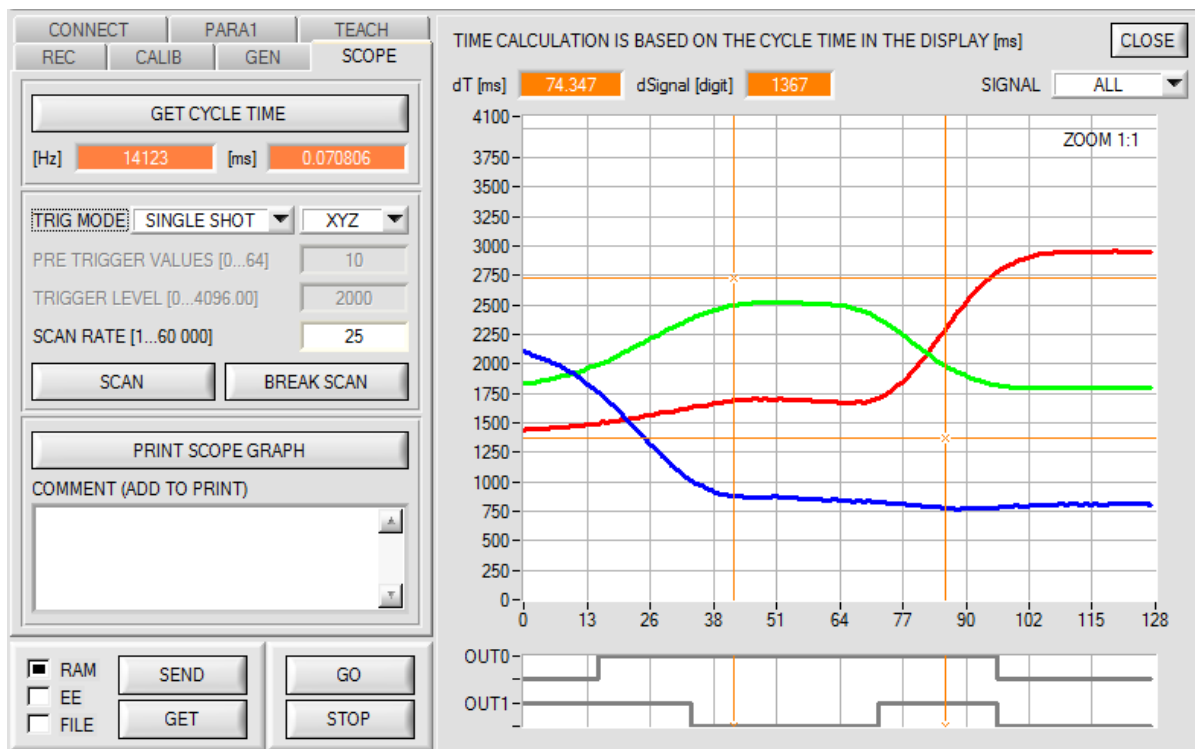
In **TRIG MODE = SINGLE SHOT** a data frame is recorded and displayed in the graph when you click on **SCAN**.

In **TRIG MODE = FALLING EDGE** and **RIISING EDGE** triggered recording can be started by clicking on **SCAN**. A trigger start can be defined with **TRIGGER LEVEL**.

Triggering is performed either to **Z** or to the brightness value of the color space, depending on the signal that should be recorded. In the graph this is the blue line.

In **TRIG MODE = INTERN C-No.0** recording starts automatically when C-No. 0 is detected. **TRIG MODE= EXTERN IN0** can be used to start recording externally through 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 two 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 digits or in Volt in Y direction.

**SIGNAL** can be used to display individual curves.

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

## 2.9 Tab XYZOFF

An offset for X Y Z can be defined with **CALIB = XYZ OFFSET** and **XYZ OFFSET IN0**.

For defining the offset, the surface that should be compensated must be placed under the sensor, and **GO** mode must be activated.

Then the sensor parameters must be set.

In the **XYZOFF** tab the raw signal is shown in the corresponding displays and bars.

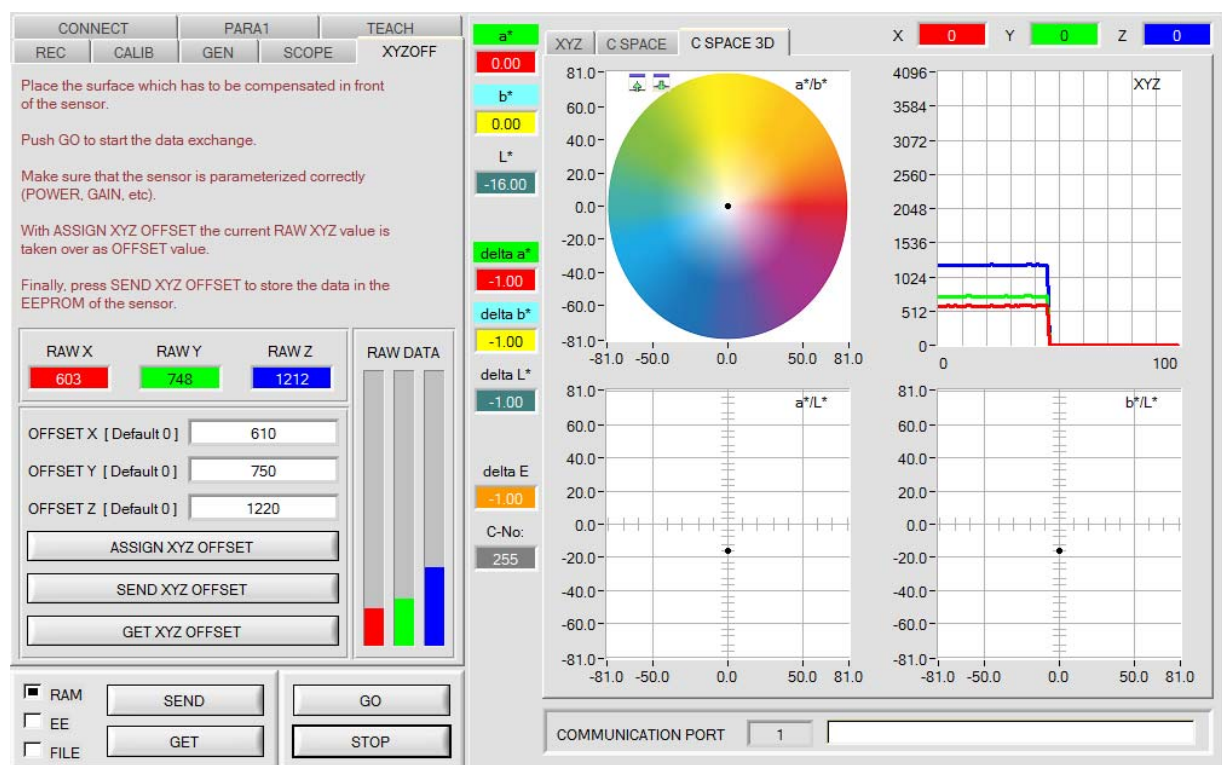
When you click on **ASSIGN XYZ OFFSET** the current raw signals are copied to the input fields.

With **SEND XYZ OFFSET** these values are sent to the sensor's EEPROM (non-volatile memory). With **GET XYZ OFFSET** the values can be loaded from the sensor.

The offset values are effective when they have been saved in the sensor and when **CALIB = XYZ OFFSET** has been selected.

If you select **CALIB = XYZ OFFSET IN0**, the offset values also can be updated through input IN0.

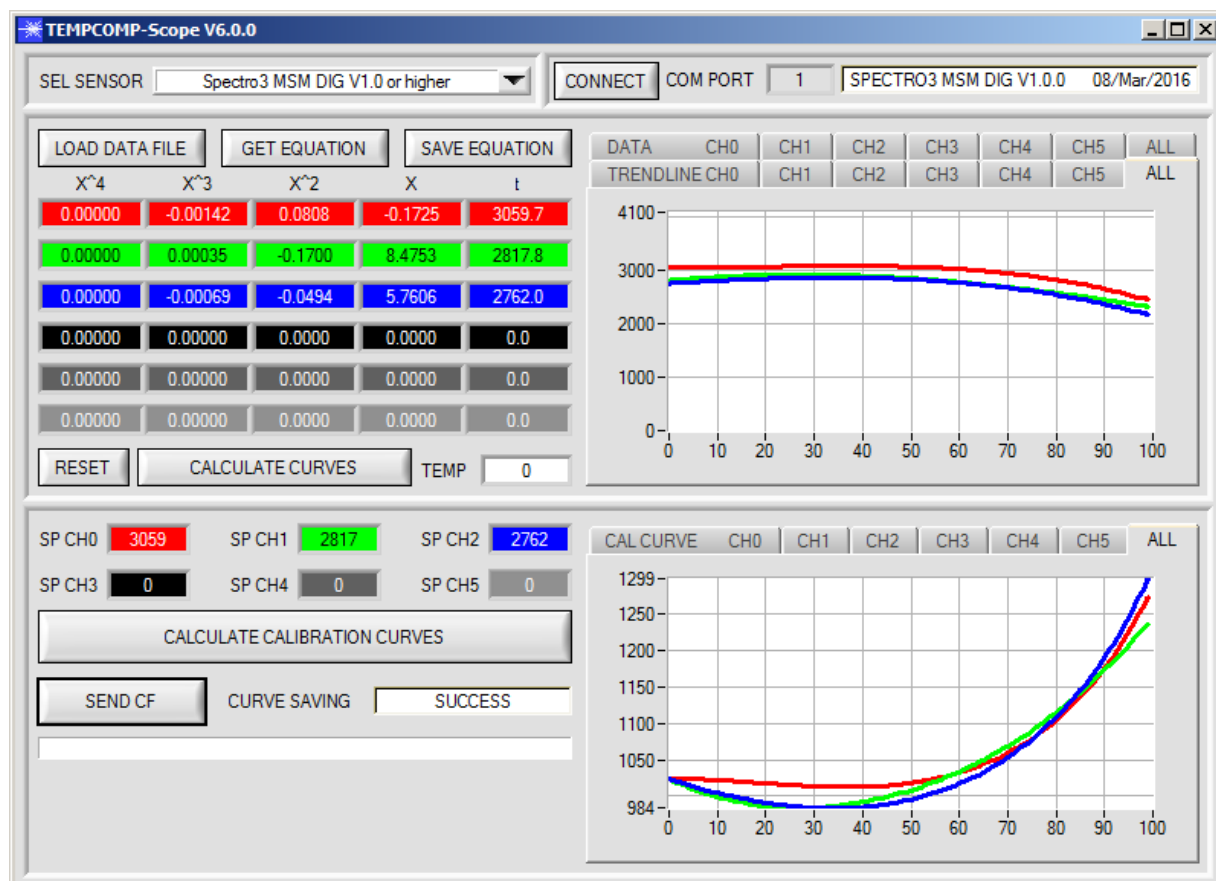
**Attention!** Through IN0 the offset values only are saved in the sensor's RAM (volatile memory).



### 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.

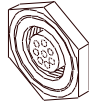
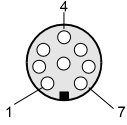
To perform temperature compensation please start the corresponding **TEMPCOMP-Scope software** that is included on the supplied CD. Please make sure that you have a functioning sensor connection. It may be necessary to select the connection with **CONNECT**. Set the correct sensor under **SELECT SENSOR**, if this is not done automatically.



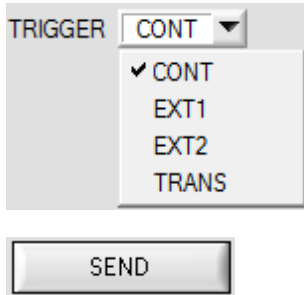
- Step 1: Load the temperature compensation file that you have received from your supplier with **GET EQUATION** or **LOAD DATA FILE**.
- Step 2: Press **CALCULATE CURVES** to display the data in the graph.
- Step 3: Select the sensor-internal operating temperature (not in °C) that the sensor has at 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. External triggering of the SPECTRO-3-MSM-ANA color sensor



External triggering is performed through pin no. 3 (grn) at the 8-pole socket of the SPECTRO-3-MSM-ANA/PLC connection.

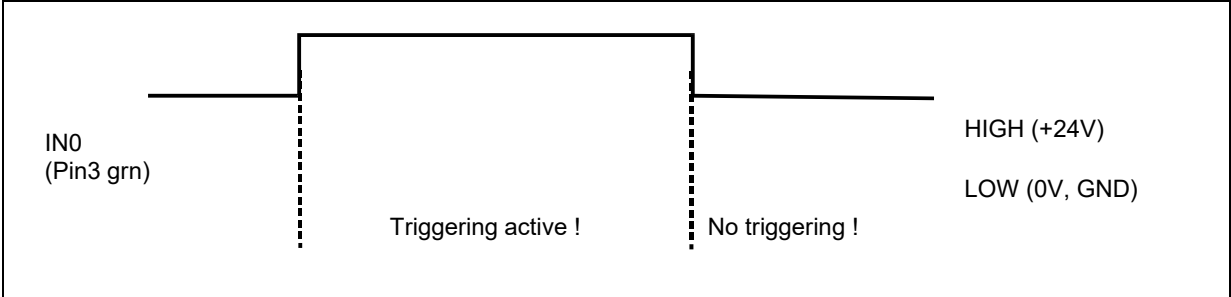
**TRIGGER:**

First the external trigger mode must be set at the color sensor. For this purpose option **EXT1**, **EXT2**, or **TRANS** must be selected in the **TRIGGER** selection field.

**PLEASE NOTE:**


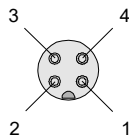
The new setting is only activated at the sensor after a click on the **SEND** button!

**Please note:** The trigger input (IN0 PIN3 green of cable cab-las8/SPS) is HIGH-active, i.e. a trigger event is detected as long as IN0 = HIGH (+24V).


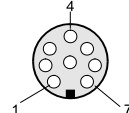


## 5. Connector assignment of the SPECTRO-3-MSM-ANA color sensor

Connection of SPECTRO-3-MSM-ANA to PC:

| <b>4-pole fem. M5 connector (type Binder 707)</b><br><b>SPECTRO-3-MSM-ANA/PC-RS232</b> |  |              |   |
|--|--|--------------|---|
| Pin-No.:   |  | Assignment:  |   |
| 1  |  | +24VDC (+Ub) |   |
| 2  |  | 0V (GND)     |   |
| 3  |  | Rx0          |   |
| 4  |  | Tx0          |   |

Connection of SPECTRO-3-MSM-ANA to PLC:

| <b>8-pole female connector (type Binder 712)</b><br><b>SPECTRO-3-MSM-ANA/PLC</b> |        |                             |   |
|--|--------|-----------------------------|---|
| Pin No.:   | Color: | Assignment:                 |   |
| 1  | White  | 0V (GND)                    |   |
| 2  | Brown  | +24VDC ( $\pm 10\%$ ) (+Ub) |   |
| 3  | Green  | IN0                         |   |
| 4  | Yellow | OUT0                        | Digital 0: typ. 0 ... 1V, Digital 1: typ. +Ub – 10%   |
| 5  | Grey   | OUT1                        | Digital 0: typ. 0 ... 1V, Digital 1: typ. +Ub – 10%   |
| 6  | Pink   | OUT2                        | Analog X, x, a*, u*, u' or C* (0...10V or 4...20mA)   |
| 7  | Blue   | OUT3                        | Analog Y, y, b*, v*, v' or h* (0...10V or 4...20mA)   |
| 8  | Red    | OUT4                        | Analog Z, Y or L* (0...10V or 4...20mA)   |

## 6. RS232 communication protocol

The sensors of the SPECTRO-3-MSM-ANA 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**

| Parameter                          | Type | Meaning   |
|------------------------------------|------|---|
| <b>Para1:</b> POWER                | word | Transmitter intensity (0 ... 1000) Attention intensity in thousandth!   |
| <b>Para2:</b> PMODE                | word | Power Mode: SINGLE, DOUBLE coded to (0,1)   |
| <b>Para3:</b> GAIN                 | word | Amplification of the integrated receiver AMP1, AMP2, AMP3, AMP4, AMP5, AMP6, AMP7, AMP8 coded to (1, 2, 3, 4, 5, 6, 7, 8)                           |
| <b>Para4:</b> INTEGRAL             | word | Signal integration (1...250)  |
| <b>Para5:</b> AVERAGE              | word | Signal averaging 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384 or 32768  |
| <b>Para6:</b> LED MODE             | word | Control for the internal light source DC, AC coded to (0,1)   |
| <b>Para7:</b> C SPACE              | word | Color Space: xyY, L*a*b*, L*u*v*, L*C*h*, L*u'v' coded to (0,1,2,3,4)   |
| <b>Para8:</b> CALIB                | word |   |
| <b>Para9:</b> DIGITAL OUTMODE      | word | Function of the digital output: OFF, DIRECT HI, DIRECT LO, BINARY HI, BINARY LO coded to (0,1,2,3,4)  |
| <b>Para10:</b> MAXCOL-No.          | word | Number of the colours (1,2,3)   |
| <b>Para11:</b> INTLIM              | word | Intensity limit (0 ... 4095)  |
| <b>Para12:</b> EVALUATION MODE     | word | Evaluation mode: FIRST HIT, BEST HIT coded to (0,1)   |
| <b>Para13:</b> SHAPE MODE          | word | Shape mode: Block, Cylinder, Sphere coded to (0,1,2)  |
| <b>Para14:</b> EXTEACH             | word | External teach mode: OFF, ON coded to (0,1)   |
| <b>Para15:</b> TRIGGER             | word | Trigger mode: CONT, EXT1, EXT2, TRANS coded to (0, 1, 2, 3)   |
| <b>Para16:</b> ANALOG OUTMODE      | word | Function of the analogue outmode: OFF, X Y Z, COLOR SPACE, CS REF coded to (0,1,2,3)  |
| <b>Para17:</b> ANA OUT SIGNAL      | Word | Function of the analog output: U, I coded to (0,1)  |
| <b>Para18:</b> ANA OUT             | word | Function of analogue out: CONT, IN0 L--->H coded to (0,1)   |
| <b>Para19:</b> ANA ZOOM            | word | Zoom factor at ANALOG OUTMODE = CS REF: x1, x2, x4, x8, x16, x32, x64, x128 Coded to (0,1,2,3,4,5,6,7)  |
| <b>Para20:</b> POWER DP1           | word | Transmitter intensity of Double Parameter Set 1 (0 ... 1000) Attention intensity in thousandth!   |
| <b>Para21:</b> GAIN DP1            | word | Amplification of the integrated receiver of Double Parameter Set 1 AMP1, AMP2, AMP3, AMP4, AMP5, AMP6, AMP7, AMP8 coded to (1, 2, 3, 4, 5, 6, 7, 8) |
| <b>Para22:</b> INTEGRAL DP1        | word | Signal integration of Double Parameter Set1 (1...250)   |
| <b>Para23:</b> POWER DP2           | word | Transmitter intensity of Double Parameter Set 2 (0 ... 1000) Attention intensity in thousandth!   |
| <b>Para24:</b> GAIN DP2            | word | Amplification of the integrated receiver of Double Parameter Set 2 AMP1, AMP2, AMP3, AMP4, AMP5, AMP6, AMP7, AMP8 coded to (1, 2, 3, 4, 5, 6, 7, 8) |
| <b>Para25:</b> INTEGRAL DP2        | word | Signal integration of Double Parameter Set2 (1...250)   |
| <b>Para26:</b> COR VAL X           | word | Correction Value for X multiplied by 128  |
| <b>Para27:</b> COR VAL Y           | word | Correction Value for Y multiplied by 128  |
| <b>Para28:</b> COR VAL Z           | word | Correction Value for Z multiplied by 128  |
| <b>Para29:</b> COR VAL X 3'rd root | word | 3'rd root of COR VAL X = $(\text{COR VAL X} * 128)^{(1/3)} * 1024$  |
| <b>Para30:</b> COR VAL Y 3'rd root | word | 3'rd root of COR VAL Y = $(\text{COR VAL Y} * 128)^{(1/3)} * 1024$  |
| <b>Para31:</b> COR VAL Z 3'rd root | word | 3'rd root of COR VAL Z = $(\text{COR VAL Z} * 128)^{(1/3)} * 1024$  |

One row in the Teach Table determines a **TEACH VECTOR**.

The sensors of the SPECTRO-3-MSM-ANA series operate with 3 **TEACH VECTORS** that are sent to the sensor or read from the sensor in one block in the stated sequence.

Info! 1 **bytes** = 8bit

1 **word** = 2 **byte**

1 **long** = 2 **word** = 4 **byte**

| TEACH VECTOR       |                            | Type | Meaning  |
|--------------------|----------------------------|------|--|
| <b>TeachVal1:</b>  | Teach Table Row 0 Column 0 | long | x, a*, u*, u' of row 0. Must be multiplied with 65536.   |
| <b>TeachVal2:</b>  | Teach Table Row 0 Column 1 | long | y, b*, v*, v' of row 0. Must be multiplied with 65536.   |
| <b>TeachVal3:</b>  | Teach Table Row 0 Column 2 | long | Y, L*, L*, L* of row 0. Must be multiplied with 65536.   |
| <b>TeachVal4:</b>  | Teach Table Row 0 Column 3 | long | Shape mode Block: xTol or a*Tol or u*Tol or u' Tol of row 0.<br>Shape mode Cylinder: xyTol or a*b*Tol or u*v*Tol or u'v' Tol of row 0.<br>Shape mode Sphere: CTO or deltaE of row 0.<br>Must be multiplied with 65536.<br>Must be sent as a dummy if not needed. |
| <b>TeachVal5:</b>  | Teach Table Row 0 Column 4 | long | Shape mode Block: yTol or b*Tol or v*Tol or v' Tol of row 0.<br>Shape mode Cylinder: YTol or L*Tol of row 0.<br>Shape mode Sphere: no use.<br>Must be multiplied with 65536.<br>Must be sent as a dummy if not needed.   |
| <b>TeachVal6:</b>  | Teach Table Row 0 Column 5 | long | Shape mode Block: YTol or L*Tol of row 0.<br>Shape mode Cylinder: no use.<br>Shape mode Sphere: no use.<br>Must be multiplied with 65536.<br>Must be sent as a dummy if not needed.  |
| <b>TeachVal7:</b>  | Free                       | word | Send 0 as dummy  |
| <b>TeachVal8:</b>  | Free                       | word | Send 0 as dummy  |
| <b>TeachVal9:</b>  | Free                       | word | Send 0 as dummy  |
| <b>TeachVal10:</b> | Free                       | word | Send 0 as dummy  |
| <b>TeachVal11:</b> | Teach Table Row 1 Column 0 | long | x, a*, u*, u' of row 1. Must be multiplied with 65536.   |
| <b>TeachVal12:</b> | Teach Table Row 1 Column 1 | long | y, b*, v*, v' of row 1. Must be multiplied with 65536.   |
| <b>TeachVal13:</b> | Teach Table Row 1 Column 2 | long | Y, L*, L*, L* of row 1. Must be multiplied with 65536.   |
| <b>TeachVal14:</b> | Teach Table Row 1 Column 3 | long | Shape mode Block: xTol or a*Tol or u*Tol or u' Tol of row 1.<br>Shape mode Cylinder: xyTol or a*b*Tol or u*v*Tol or u'v' Tol of row 1.<br>Shape mode Sphere: CTO or deltaE of row 1.<br>Must be multiplied with 65536.<br>Must be sent as a dummy if not needed. |
| <b>TeachVal15:</b> | Teach Table Row 1 Column 4 | long | Shape mode Block: yTol or b*Tol or v*Tol or v' Tol of row 1.<br>Shape mode Cylinder: YTol or L*Tol of row 1.<br>Shape mode Sphere: no use.<br>Must be multiplied with 65536.<br>Must be sent as a dummy if not needed.   |
| <b>TeachVal16:</b> | Teach Table Row 1 Column 5 | long | Shape mode Block: YTol or L*Tol of row 1.<br>Shape mode Cylinder: no use.<br>Shape mode Sphere: no use.<br>Must be multiplied with 65536.<br>Must be sent as a dummy if not needed.  |
| <b>TeachVal17:</b> | Free                       | word | Send 0 as dummy  |
| ...                | ...                        | ...  | ...  |
| <b>TeachVal30</b>  | Free                       | word | Send 0 as dummy  |



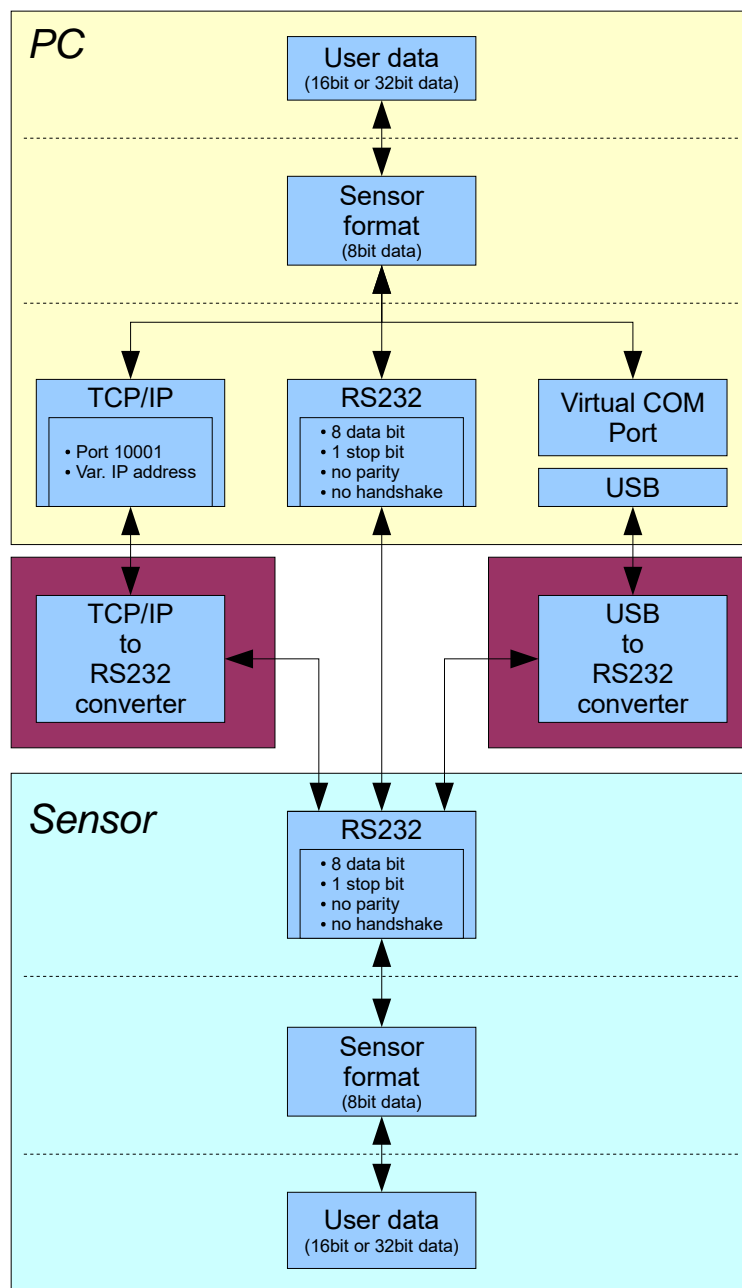
Upon request, the data acquired and processed by the sensor are sent by the sensor in the following sequence.  
 Info! 1 **bytes** = 8bit      1 **word** = 2 **byte**      1 **long** = 2 **word** = 4 **byte**

| DATA VALUE              | Type | Meaning  |
|-------------------------|------|--|
| <b>DatVal1:</b> CSX     | long | Color Space X chromaticity coordinate (x, a*, u*, C* or u'). Must be divided by 65536. |
| <b>DatVal2:</b> CSY     | long | Color Space Y chromaticity coordinate (y, b*, v*, h* or v'). Must be divided by 65536. |
| <b>DatVal3:</b> CSI     | long | Color Space Lightness variable (Y or L*). Must be divided by 65536.                    |
| <b>DatVal4:</b> REF CSX | long | Reference CSX when ANALOG OUTMODE = CS REF. Must be divided by 65536.                  |
| <b>DatVal5:</b> REF CSY | long | Reference CSY when ANALOG OUTMODE = CS REF. Must be divided by 65536.                  |
| <b>DatVal6:</b> REF CSI | long | Reference CSI when ANALOG OUTMODE = CS REF. Must be divided by 65536.                  |
| <b>DatVal7:</b> delta E | long | Color difference to a color hit. Must be divided by 65536.                             |
| <b>DatVal8:</b> X       | word | Calibrated and temperature compensated tristimulus value of channel X                  |
| <b>DatVal9:</b> Y       | word | Calibrated and temperature compensated tristimulus value of channel Y                  |
| <b>DatVal10:</b> Z      | word | Calibrated and temperature compensated tristimulus value of channel Z                  |
| <b>DatVal11:</b> RAW X  | word | None Calibrated tristimulus value of channel X   |
| <b>DatVal12:</b> RAW Y  | word | None Calibrated tristimulus value of channel Y   |
| <b>DatVal13:</b> RAW Z  | word | None Calibrated tristimulus value of channel Z   |
| <b>DatVal14:</b> C-No:  | word | Detected color   |
| <b>DatVal15:</b> DIG IN | word | DIG IN is 1 when input IN0 is HI   |
| <b>DatVal16:</b> TEMP   | word | Temperature in the sensor ( <b>not</b> in °C or °F)                                    |
| <b>DatVal17:</b> DP SET | word | Double Parameter Set   |

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<sub>dez</sub> (55<sub>hex</sub>).

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<sub>dez</sub> or 00<sub>hex</sub>, 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 eighth 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<br>Header | Byte1<br>Header | Byte2<br>Header    | Byte3<br>Header    | Byte4<br>Header    | Byte5<br>Header    | Byte6<br>Header | Byte7<br>Header  | Byte8<br>Data      | Byte9<br>Data      | ... | Byte n+6<br>Data      | Byte n+7<br>Data      |
|-----------------|-----------------|--------------------|--------------------|--------------------|--------------------|-----------------|------------------|--------------------|--------------------|-----|-----------------------|-----------------------|
| 0x55            | <order>         | <ARG><br>(lo byte) | <ARG><br>(hi byte) | <LEN><br>(lo byte) | <LEN><br>(hi byte) | CRC8<br>(Data)  | CRC8<br>(Header) | Data1<br>(lo byte) | Data1<br>(hi byte) | ... | Data n/2<br>(lo byte) | Data n/2<br>(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.<br>ARG=1: Invalid order number was sent to the sensor<br>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   |
| 108    | Read 3 data values from sensor   | order=108 |
| 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<sub>hex</sub>

**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.

SET ARG=2 if you want to send the Teach Values.

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

| Byte9<br>Data      | Byte10<br>Data     | Byte11<br>Data     | Byte12<br>Data     | Byte13<br>Data     | Byte14<br>Data     | Byte15<br>Data     | Byte16<br>Data     | Byte17<br>Data     | Byte18<br>Data     |
|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| Para1<br>(lo byte) | Para1<br>(hi byte) | Para2<br>(lo byte) | Para2<br>(hi byte) | Para3<br>(lo byte) | Para3<br>(hi byte) | Para4<br>(lo byte) | Para4<br>(hi byte) | Para5<br>(lo byte) | Para5<br>(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<br>Header | Byte2<br>Header | Byte3<br>Header    | Byte4<br>Header    | Byte5<br>Header    | Byte6<br>Header    | Byte7<br>Header | Byte8<br>Header  |
|-----------------|-----------------|--------------------|--------------------|--------------------|--------------------|-----------------|------------------|
| 0x55            | <order>         | <ARG><br>(lo byte) | <ARG><br>(hi byte) | <LEN><br>(lo byte) | <LEN><br>(hi byte) | CRC8<br>(Data)  | CRC8<br>(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.

SET ARG=2 if you want to read the Teach Values.

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

DATA FRAME Sensor → PC

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

| Byte9<br>Data      | Byte10<br>Data     | Byte11<br>Data     | Byte12<br>Data     | Byte13<br>Data     | Byte14<br>Data     | Byte15<br>Data     | Byte16<br>Data     | Byte17<br>Data     | Byte18<br>Data     |
|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| Para1<br>(lo byte) | Para1<br>(hi byte) | Para2<br>(lo byte) | Para2<br>(hi byte) | Para3<br>(lo byte) | Para3<br>(hi byte) | Para4<br>(lo byte) | Para4<br>(hi byte) | Para5<br>(lo byte) | Para5<br>(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><br>(lo byte) | <ARG><br>(hi byte) | <LEN><br>(lo byte) | <LEN><br>(hi byte) | CRC8<br>(Data) | CRC8<br>(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><br>(lo byte) | <ARG><br>(hi byte) | <LEN><br>(lo byte) | <LEN><br>(hi byte) | CRC8<br>(Data) | CRC8<br>(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><br>(lo byte) | <ARG><br>(hi byte) | <LEN><br>(lo byte) | <LEN><br>(hi byte) | CRC8<br>(Data) | CRC8<br>(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><br>(lo byte) | <ARG><br>(hi byte) | <LEN><br>(lo byte) | <LEN><br>(hi byte) | CRC8<br>(Data) | CRC8<br>(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><br>(lo byte) | <ARG><br>(hi byte) | <LEN><br>(lo byte) | <LEN><br>(hi byte) | CRC8<br>(Data) | CRC8<br>(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><br>(lo byte) | <ARG><br>(hi byte) | <LEN><br>(lo byte) | <LEN><br>(hi byte) | CRC8<br>(Data) | CRC8<br>(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><br>(lo byte) | <ARG><br>(hi byte) | <LEN><br>(lo byte) | <LEN><br>(hi byte) | CRC8<br>(Data) | CRC8<br>(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><br>(lo byte) | <ARG><br>(hi byte) | <LEN><br>(lo byte) | <LEN><br>(hi byte) | CRC8<br>(Data) | CRC8<br>(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><br>(lo byte) | <ARG><br>(hi byte) | <LEN><br>(lo byte) | <LEN><br>(hi byte) | CRC8<br>(Data) | CRC8<br>(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** (See Table DATA VALUE)

**Example order=108:** Read 3 data values from sensor.

DATA FRAME PC → Sensor

| Byte0<br>Header | Byte1<br>Header | Byte2<br>Header    | Byte3<br>Header    | Byte4<br>Header    | Byte5<br>Header    | Byte6<br>Header | Byte7<br>Header  |
|-----------------|-----------------|--------------------|--------------------|--------------------|--------------------|-----------------|------------------|
| 0x55            | <order>         | <ARG><br>(lo byte) | <ARG><br>(hi byte) | <LEN><br>(lo byte) | <LEN><br>(hi byte) | CRC8<br>(Data)  | CRC8<br>(Header) |
| 85 (dec)        | 108             | 0                  | 0                  | 0                  | 0                  | 170             | 105              |
| ARG=0           |                 |                    |                    | LEN=0              |                    |                 |                  |

DATA FRAME Sensor → PC

8 Header Bytes and 3 **DATA VALUES** (DatVal1, DatVal2, DatVal3. See Table DATA VALUE)

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

**Start** triggered sending of data frames

DATA FRAME PC → Sensor

<ARG> determines which data should be send when Trigger goes HI

ARG = 1 --> The HEADER and all DATA VALUES will be send

ARG = 2 --> The HEADER and only the color space coordinates CSX, CSY, CSZ will be send

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

DATA FRAME Sensor → PC

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

DATA FRAME Sensor → PC

| Byte0<br>Header | Byte1<br>Header | Byte2<br>Header    | Byte3<br>Header    | Byte4<br>Header    | Byte5<br>Header    | Byte6<br>Header | Byte7<br>Header  |
|-----------------|-----------------|--------------------|--------------------|--------------------|--------------------|-----------------|------------------|
| 0x55            | <order>         | <ARG><br>(lo byte) | <ARG><br>(hi byte) | <LEN><br>(lo byte) | <LEN><br>(hi byte) | CRC8<br>(Data)  | CRC8<br>(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><br>(lo byte) | <ARG><br>(hi byte) | <LEN><br>(lo byte) | <LEN><br>(hi byte) | CRC8<br>(Data) | CRC8<br>(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><br>(lo byte) | <ARG><br>(hi byte) | <LEN><br>(lo byte) | <LEN><br>(hi byte) | CRC8<br>(Data) | CRC8<br>(Header) | lo word<br>lo byte   | lo word<br>hi byte | hi word<br>lo byte | hi word<br>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<br>lo byte | lo word<br>hi byte | hi word<br>lo byte | hi word<br>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><br>(lo byte) | <ARG><br>(hi byte) | <LEN><br>(lo byte) | <LEN><br>(hi byte) | CRC8<br>(Data) | CRC8<br>(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><br>(lo byte) | <ARG><br>(hi byte) | <LEN><br>(lo byte) | <LEN><br>(hi byte) | CRC8<br>(Data) | CRC8<br>(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:

- IBM PC AT or compatible
- VGA graphics
- Microsoft® Windows® 7, 8, 10
- Serial RS232 interface at the PC
- Microsoft® compatible mouse
- Cable ***cab-las4/PC (cab-las5/PC)*** for the RS232 interface or ***cab-4/USB (cab-5/USB)*** for USB slot or ***cab-4/ETH (cab-5/ETH)***

Please install the software as described below:

1. The software can be installed directly from the installation CD-ROM. To install the software, start the SETUP program in the INSTALL folder of the CD-ROM.
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.**

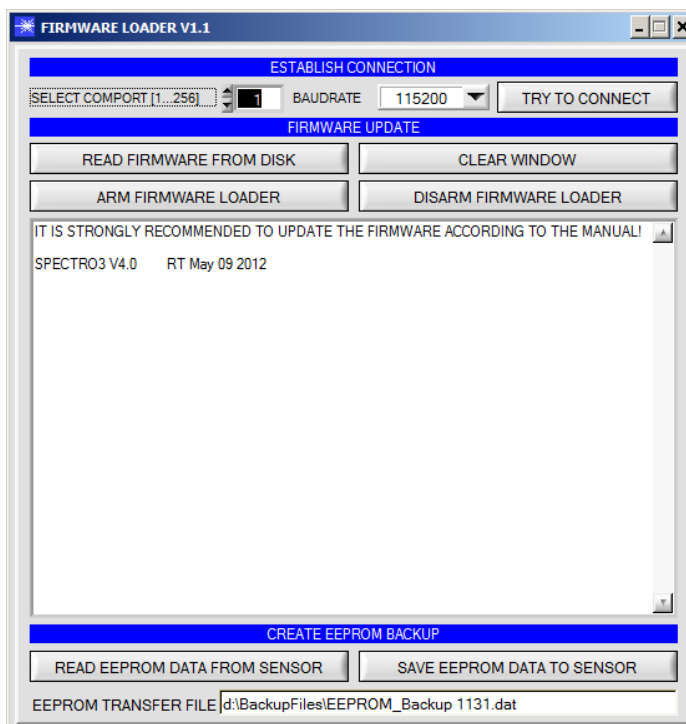
## Step 1:

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.

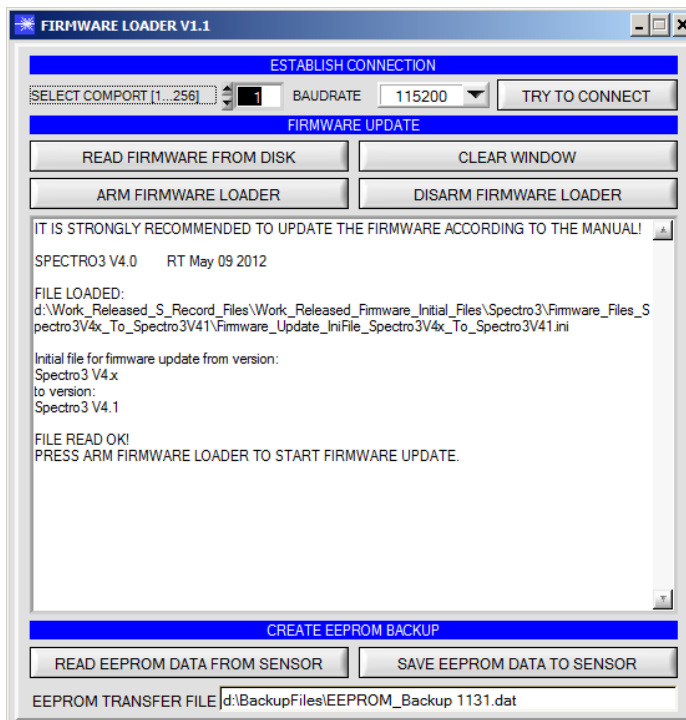
As described above, a plausibility check of the initialisation file will be performed first.

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.

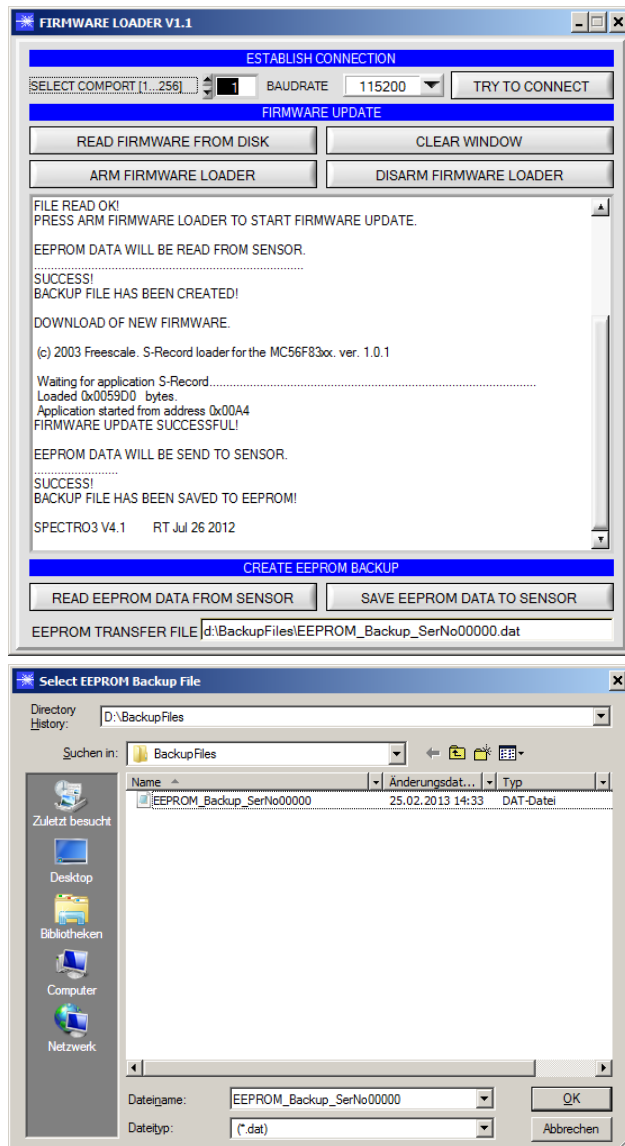
When you press the **ARM FIRMWARE LOADER** button the firmware update will be performed automatically.

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 EEPROM 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 choose 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.

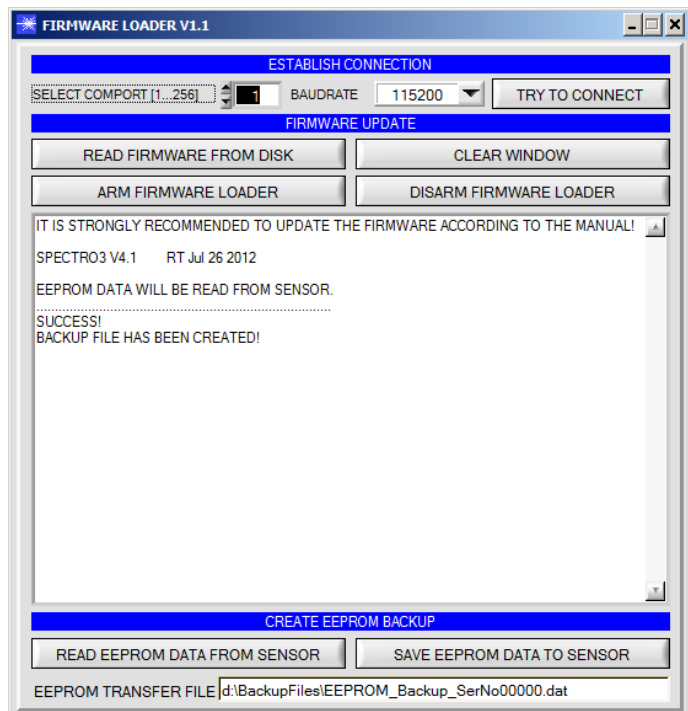
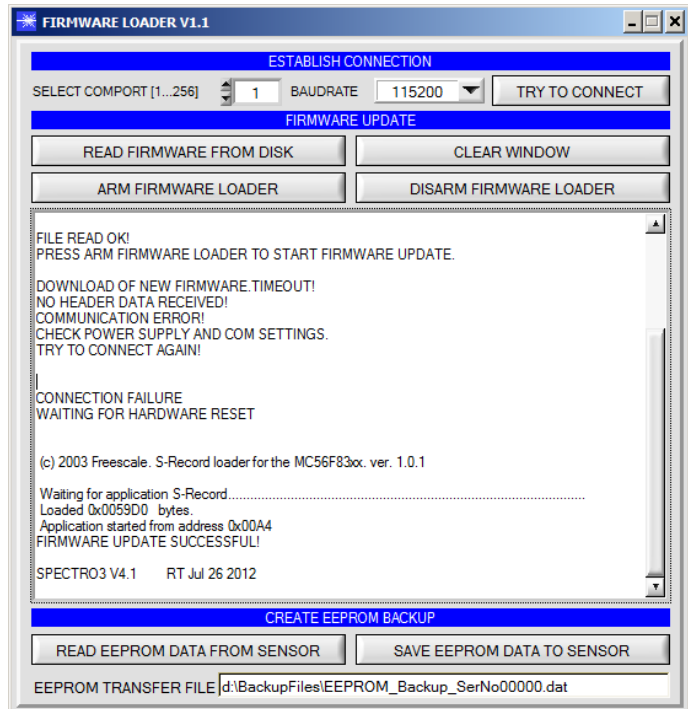
The file name for your **EEPROM backup file** should be chosen such that the names for several sensors cannot be mixed up. Using a file name that contains the sensor serial number might be advisable.

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.

