

# Manual

## Software SPECTRO3-Scope V4.2

(PC software for Microsoft® Windows® XP, VISTA, 7)

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

This manual describes the installation of the PC software for the SPECTRO-3 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 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 and allows a scan-frequency of up to 35 kHz. An OFF function turns off the integrated light source at the sensor and changes to DC operation. The sensor then can detect so-called "self-luminous objects". In PULSE operation extremely dark surfaces can be reliably detected. 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 color sensor is activated, the sensor detects the radiation that is diffusely reflected from the object. As a light source the SPECTRO-3 color sensor uses a white-light LED 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 color sensor can be "taught" up to 31 colors. For each of these taught colors it is possible to set tolerances. In "X Y INT - 2D" or "s i M - 2D" mode these tolerances form a color cylinder in space. In "X Y INT - 3D" or "s i M - 3D" mode the tolerances form a color sphere in space. Color evaluation according to "s i M - 2D" is based on the lab calculation method. All modes can be used in combination with several operating modes such as "FIRST HIT" and "BEST HIT". Raw data are represented with 12 bit resolution.

As a special feature the sensor can be taught two completely independent parameter sets. Input IN0 can then be used to tell the sensor which parameter set it should work with.

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 5 digital outputs or can be sent directly to the outputs, if only up to 5 colors are to be detected. At the same time the detected color code is visualised by means of 5 LEDs at the housing of the SPECTRO-3. **[Please note: Visualisation by means of LEDs not available with SPECTRO-3-...-JR types.]**

With a TEACH button at the sensor housing the color sensor can be taught up to 31 colors. For this purpose the corresponding evaluation mode must be set with the software. The TEACH button is connected in parallel to the input IN0 (green wire at cable cab-las8/SPS). **[Please note: TEACH button not available with SPECTRO-3-...-JR types.]**

Parameters and measurement values can be exchanged between a PC and the SPECTRO-3 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 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.

The sensors of the SPECTRO-3 series can be calibrated (white-light balancing). Balancing can be performed to any white surface. A ColorChecker™ table with 24 color fields is available as an alternative. White-light balancing or calibration can be performed to one of the white fields.

As a light source the SPECTRO-3-...-UV color sensor uses a UV-LED (375 nm, or 365 nm in case of using an external UV illumination unit SPECTRO-ELS-UV) with adjustable transmitter power to excite the luminescent marking. These UV-sensors can be optimally adjusted to almost any luminescent colorant that can be excited in the long-wave UV range (375 nm or 365 nm). The sensors of the SPECTRO-3-...-UV series also can be calibrated. Analogous to white-light balancing with color sensors, balancing of the SPECTRO-3-...-UV could be performed to any luminescent color marking.

## 0 Contents

|  | Page |
|--|------|
| 1. Installation of the SPECTRO3-Scope software .....   | 3    |
| 2. Operation of the SPECTRO3-Scope software.....   | 4    |
| 2.1 Tab CONNECT .....  | 5    |
| 2.2 Tab PARA1, button SEND, GET, GO, STOP (parameterization, data exchange) .....                    | 7    |
| 2.3 Tab PARA2 (color groups and hold time setting) .....   | 19   |
| 2.4 Tab TEACH (Teach Table).....   | 20   |
| 2.5 Tab GEN (color rows and color specifications) .....  | 24   |
| 2.6 Tab RECORDER (data recording) .....  | 25   |
| 2.7 Tab CALIBRATE .....  | 27   |
| 2.7.1 White light balancing .....  | 27   |
| 2.7.2 Offset calibration .....   | 30   |
| 2.8 Tab SCOPE .....  | 31   |
| 2.9 Graphic display elements .....   | 32   |
| 3. Short instructions for the operation of color sensors with the SPECTRO3-Scope V4.2 software ..... | 34   |
| 4. Operation of the TEMPComp-Scope software .....  | 38   |
| 5. External triggering of the SPECTRO-3 color sensor .....   | 39   |
| 6. Function of the LED display <b>[not available with SPECTRO-3-...-JR types]</b> .....              | 40   |
| 7. Connector assignment of the SPECTRO-3 color sensor.....   | 42   |
| 8. RS232 communication protocol.....   | 43   |

### Shortcuts:

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

# 1 Installation of the SPECTRO3-Scope software

Hardware requirements for successful installation of the SPECTRO3-Scope software:

- IBM PC AT or compatible
- VGA graphics
- Microsoft® Windows® XP, VISTA, 7
- Serial RS232 interface at the PC
- Microsoft-compatible mouse
- Cable for the RS232 interface (cab-las4/PC or cab-las4/USB)
- CD-ROM drive
- 20 MByte of free hard disk space

The SPECTRO3-Scope software can only be installed under Windows. Windows must therefore be started first, if it is not yet running.

Please install the software as described below:

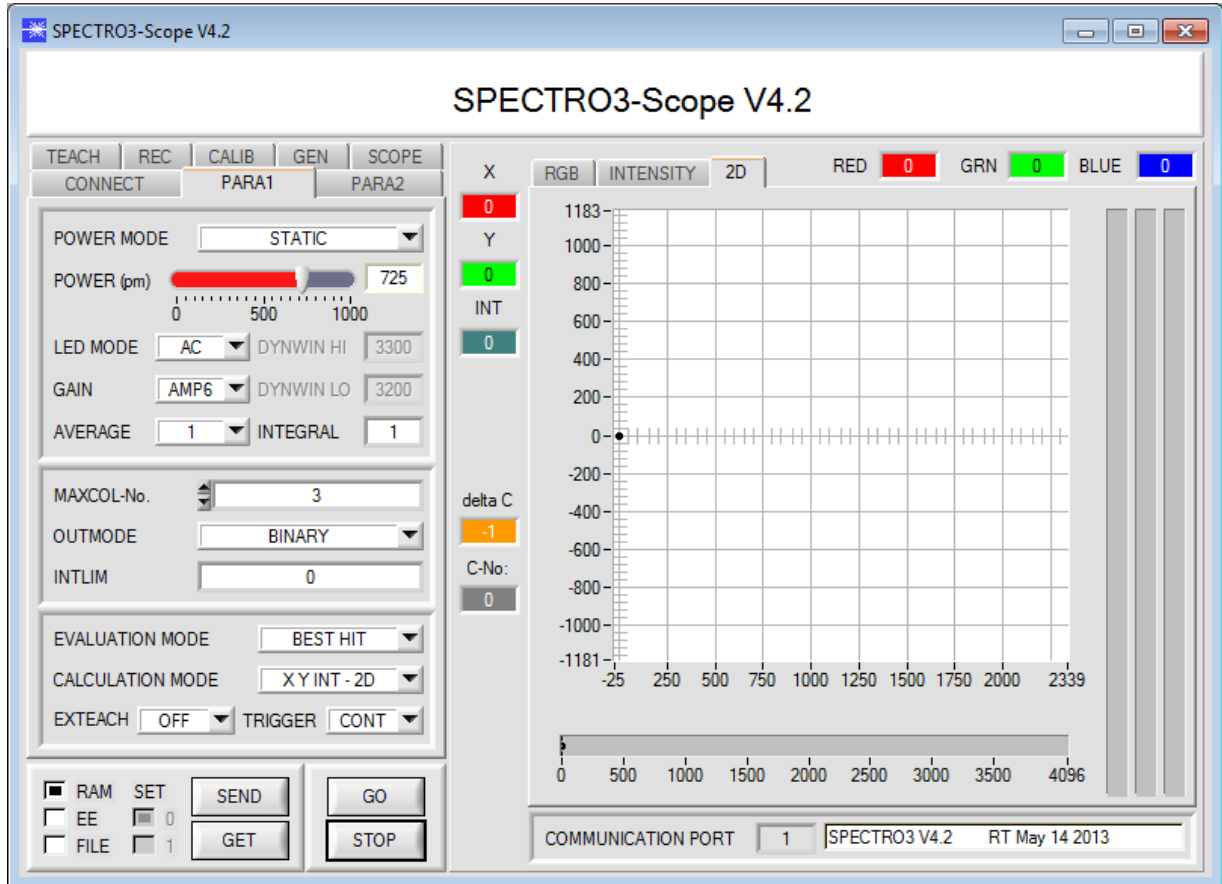
1. The software can be installed directly from the installation CD-ROM. To install the software, start the SETUP program in the SOFTWARE folder of the CD-ROM.
2. The installation program displays a dialog and suggests to install the software in the C:\"FILENAME" directory on the hard disk. You may accept 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-Scope software

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

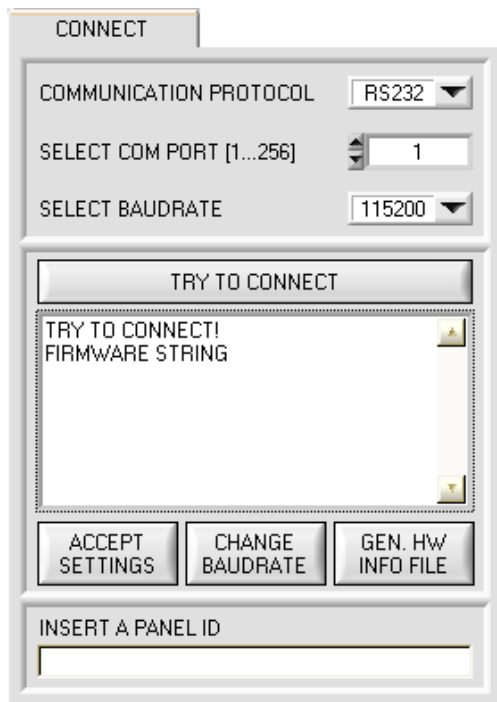
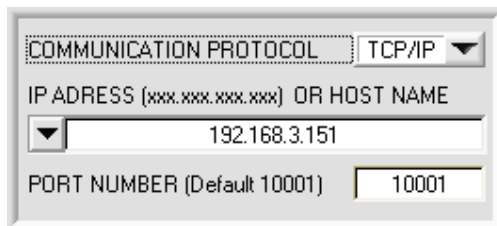
When the SPECTRO3-Scope software is started, the following window appears on the Windows interface:



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.

**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.

If the sensor should communicate through a local area network, an RS232 to Ethernet adaptor will be needed. This adapter makes it possible to establish a connection to the sensor with the **TCP/IP** protocol.


The network adaptors that are available from us are based on the Lantronix XPort module. For parameterising these adapters (assigning of an IP address, setting of the Baud rate of 19200) please download the "DeviceInstaller" software that is provided free of charge by Lantronix at <http://www.lantronix.com/>. DeviceInstaller is based on Microsoft's ".NET" framework. Detailed operating instructions for the "DeviceInstaller" software also are available from Lantronix.

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 XPort-based network adaptors is 10001. 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.

|  |  |
|--|--|
| <p><b>Please note:</b></p>  <p><b>ATTENTION !</b></p> | <p>The stable function of the interface is a basic prerequisite for measured value transfer from the PC to the sensor.</p> <p>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.</p> <p>For maintaining maximum switching frequency at the sensor data communication with the PC must be stopped (press the STOP button).</p> |
|--|--|

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

If the baudrate should be changed, a connection must first be established by clicking on **TRY TO CONNECT**.

The **CHANGE BAUDRATE** button will then be active.

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

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

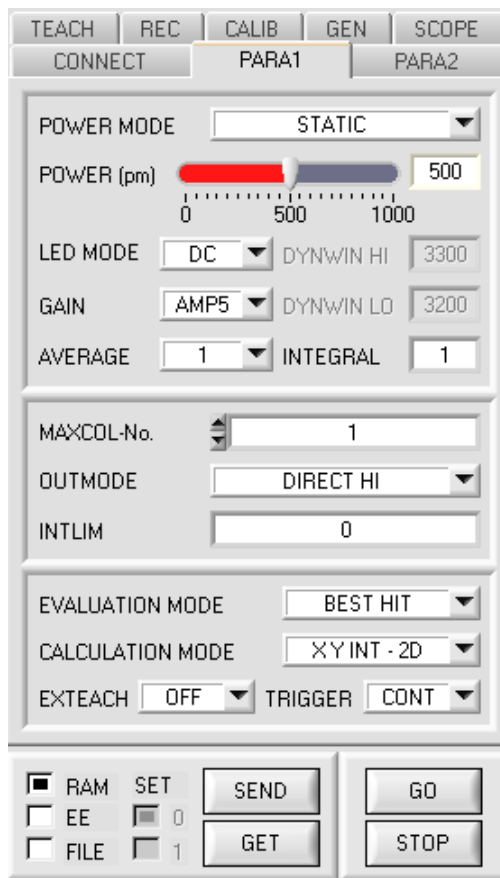
When the new baudrate information has been successfully sent, the sensor operates with the new baudrate. A window will pop up, prompting you to select **EEPROM** and then to press **SEND**. After a hardware reset the new baudrate only will be used when **EEPROM** and **SEND** have been pressed.

A click on **ACCEPT SETTINGS** saves the current interface settings, which will then be automatically set when the software is restarted.

A click on the **GEN. HW INFO FILE** generates a file in which all the important sensor data are stored in encrypted form.

This file can be sent to the manufacturer for diagnostic purposes.

## 2.2 Tab PARA1, button SEND, GET, GO, STOP



### 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.

### SET:

If you work with **TRIGGER=PARA**, two parameter sets can be saved for the sensor.

**SET** can then be used to select whether the current parameters on the user interface should be saved as parameter set 0 or parameter set 1 in the sensor.

Input IN0 is used to tell the sensor which parameter set it should work with.

**SET** only is active with **TRIGGER=PARA**.

EVALUATION MODE BEST HIT

#### EVALUATION MODE:

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

The **CALCULATION MODE** function field is active in evaluation modes **FIRST HIT**, **BEST HIT**, **MIN DIST** and **COL5**. Color information that is then correspondingly evaluated is calculated from the raw data red, green, and blue.

With evaluation mode **THD RGB** only switching thresholds are defined for the respective channels.

For evaluation the raw data red, green, and blue are considered individually here.

If the current signal is higher than the set switching threshold, the respective output is HI.

If the signal is lower, the output changes to LO.

The **TEACH TABLE** is correspondingly adapted depending on **EVALUATION MODE** and **CALCULATION MODE**.

In the example below the calculation method **CALCULATION MODE "X Y INT - 2D"** is used to explain the evaluation modes **FIRST HIT**, **BEST HIT**, **MIN DIST** and **COL5**.

EVALUATION MODE FIRST HIT

|   | X    | Y   | CTO | INT  | ITO |  |
|---|------|-----|-----|------|-----|--|
| 0 | 2354 | 903 | 100 | 1555 | 100 |  |
| 1 | 2354 | 903 | 200 | 1555 | 200 |  |
| 2 | 2354 | 903 | 300 | 1555 | 300 |  |
| 3 | 2354 | 903 | 400 | 1555 | 400 |  |
| 4 | 1    | 1   | 1   | 1    | 1   |  |
| 5 | 1    | 1   | 1   | 1    | 1   |  |

#### 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 ... OUT4) according to the settings of the **OUTMODE** parameter (see **OUTMODE**).

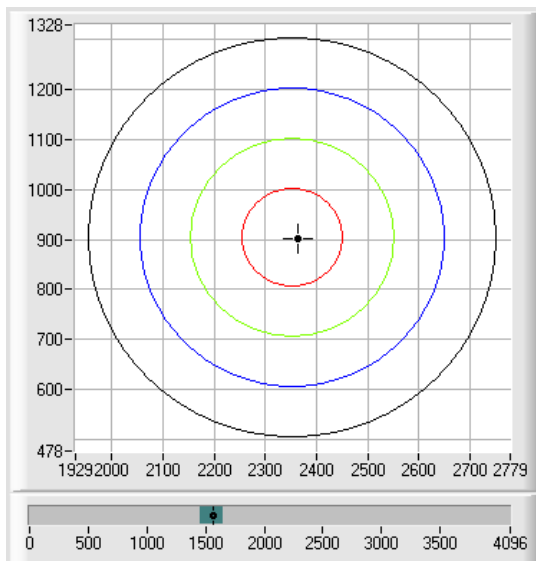
If the current color does not correspond with any of the teach-colors, the color code **C-No.** = 255 will be set ("error status").

**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!** If, for example, only the X/Y coordinates should be checked, and intensity **INT** is of no importance, a tolerance of 4000 can be selected for **ITO**. This inspection criterion is thus always fulfilled.

**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.

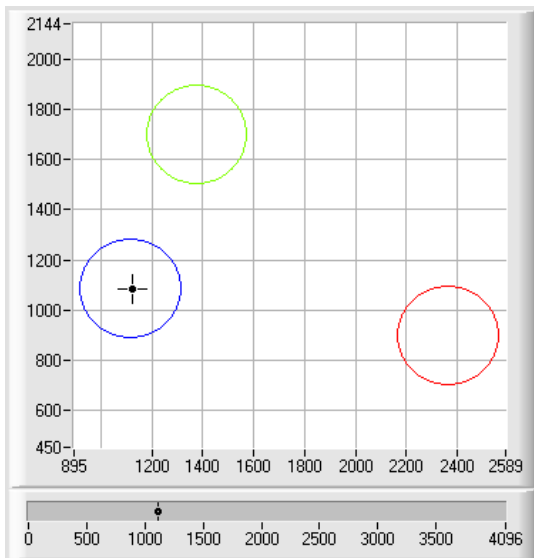
**Tip!** The teach colors are only activated when the **SEND** button is pressed!





EVALUATION MODE BEST HIT

|   | X    | Y    | CTO | INT  | ITO |       |
|---|------|------|-----|------|-----|-------|
| 0 | 2364 | 894  | 200 | 1580 | 200 | Red   |
| 1 | 1379 | 1700 | 200 | 1112 | 200 | Green |
| 2 | 1120 | 1084 | 200 | 1127 | 200 | Blue  |
| 3 | 1    | 1    | 1   | 1    | 1   | Black |



#### 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 x/y 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 ... OUT4) 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").

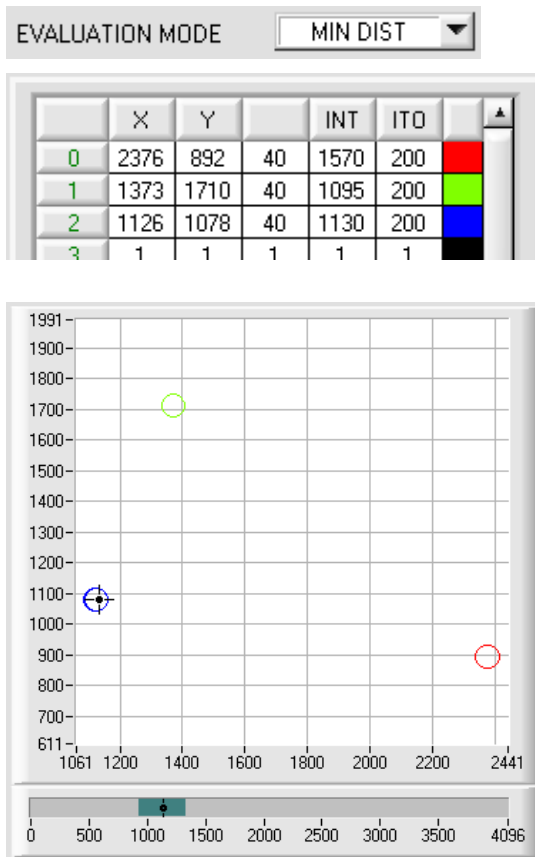
**Tip!** This mode is used if several colors must be separated from each other, and only certain surface fluctuations are allowed.

**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!** In **CALCULATION MODE** = "X Y INT - 3D" or "s i M - 3D" the shortest distance in a three-dimensional space is calculated.

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



#### MIN DIST:

The individual teach-in colors defined in the TEACH TABLE are present as points in the color triangle, defined by their (X,Y) value pairs. When this evaluation mode is set at the sensor, the evaluation algorithm, starting from the currently measured color value (X,Y), calculates the distance to the individual teach-in colors in the color triangle. The current color value (X,Y) is assigned to the teach-in color that is closest in the color triangle.

It is furthermore checked whether the intensity condition for this color also is true. If the intensity condition is not true, the second shortest distance will be checked, etc.

The color detected this way is output at the digital outputs (OUT0 ... OUT4) according to the settings of the **OUTMODE** parameter (see **OUTMODE**).

**C-No.** will only be set to 255, if the current intensity falls below the value set under **INTLIM** (see **INTLIM**).

**Comment!** The value of 40 is only entered here in order to be able to represent the coordinates of the individual teach colours in the graph. It has no meaning for evaluation.

**Tip!** This mode is used if you have taught several surfaces of separate colors, and a current color in any case must be assigned to one of the taught colors. This applies, for example, if you want to compensate a product variation, or if you have to solve sorting tasks.

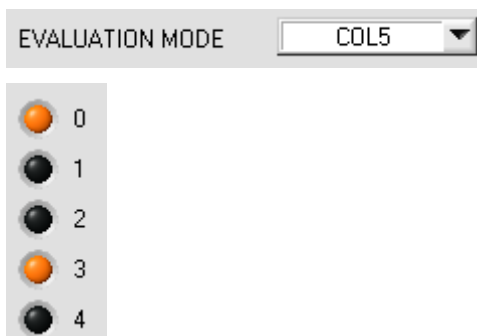
**Tip!** If, for example, only the X/Y coordinates should be

checked, and intensity **INT** is of no importance, a tolerance of 4000 can be selected for **ITO**. This inspection criterion is thus always fulfilled.

**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!** In **CALCULATION MODE = "X Y INT - 3D"** or **"s i m - 3D"** the shortest distance in a three-dimensional space is calculated.

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



#### COL5:

In this evaluation mode the rows 0 to 4 in the **TEACH TABLE** are evaluated.

Every match of current vector (row number) and teach vector is directly sent to the corresponding output.

#### Example.

If the result of evaluation is that both row 0 and row 3 are a hit, outputs OUT0 and OUT3 will be set to high (+24V).

**Tip!** If the result of evaluation is that both row 0 and row 3 are hits, outputs OUT0 and OUT3 will be set to high (+24V).

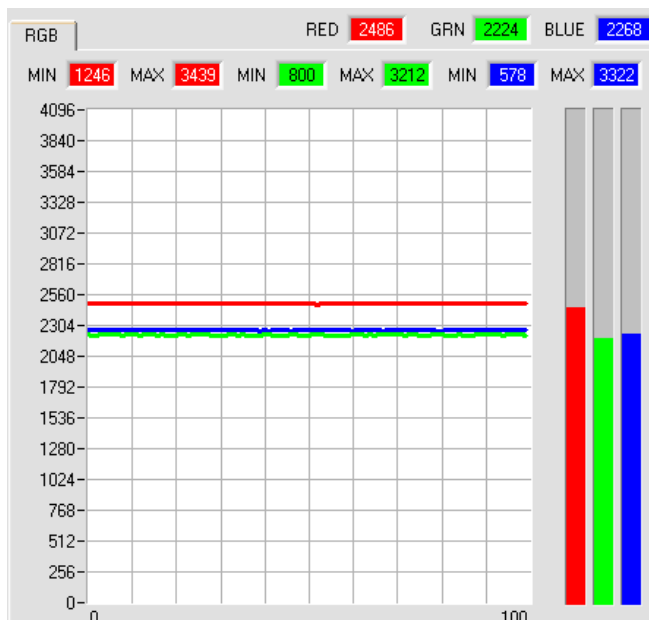
**Tip!** Values can be entered in a cell of the table either by double-clicking on the respective cell, or by highlighting the respective cell and then pressing F2.

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

**Attention!** In **COL5** evaluation mode it is not possible to form color groups!

MAXCOL-No.  0 1 2  
 OUTMODE   
 INTLIM   
 EVALUATION MODE   
 CALCULATION MODE   
 EXTEACH  TRIGGER

| TEACH |      |   |   |   |   |
|-------|------|---|---|---|---|
|       | THD  |   |   |   |   |
| 0     | 2314 | 1 | 1 | 1 | 1 |
| 1     | 1930 | 1 | 1 | 1 | 1 |
| 2     | 1131 | 1 | 1 | 1 | 1 |



EVALUATION MODE   
 CALCULATION MODE   
 EXTEACH  TRIGGER

### THD RGB:

In this evaluation mode a specific switching threshold can be set for red, green, and blue. If the respective channel is higher than this threshold, OUT0 = HI, otherwise it is LO.

**HOLD** is used as output pulse lengthening for error state 255.

If one of the three outputs changes its state, the **HOLD** time is started.

On the screen this is visualised by LEDs **0, 1** and **2**.

In the teach table the switching threshold for red is set in row 0 under the **THD** column.

Green is set in row 1, and blue in row 2.

If you click on **TEACH DATA TO**, the current red, green, and blue values are automatically written to the table.

If a teach mode is selected under **EXTEACH**, the switching thresholds for red, green, and blue can be learned through IN0.

While IN0=HI, a maximum and a minimum is determined for every channel.

The determined **MIN-MAX** values for the **RED GRN BLUE** channels are displayed in the **RGB** tab.

The respective switching threshold is calculated from  $THD = (Max + Min) / 2$  and thus lies exactly in the middle of the determined Max-Min values.

If you click on GET, the determined thresholds are displayed in the teach table.

With **EXTEACH=ON** the switching thresholds are stored in the sensor's EEPROM.

With **EXTEACH=STAT1** the switching thresholds are stored in the sensor's RAM.

With **EXTEACH=DYN1** the sensor first automatically adjusts itself, then starts the **MIN-MAX** search, and finally stores the switching thresholds in the RAM.

Since the **THD RGB** mode is no color evaluation in the proper sense, certain input elements and display elements are masked out (**C-No.**, **delta C**, ...) or grayed out (**TRIGGER**, **CALCULATION MODE**, **MAXCOL-No.**, ...).

CALCULATION MODE

XY INT - 2D

CALCULATION MODE:

#### "X Y INT - 2D":

The X/Y pairs of the individual red, green, and blue components, and the intensity are used for evaluation. For **X/Y** a color tolerance **CTO** can be set, and for the intensity an **INT** tolerance **ITO** can be set. With the individual tolerances the color is represented as a cylinder in space (see picture below). **CTO** defines the diameter of the cylinder, and **ITO** the height of the cylinder.

#### "s i M - 2D":

The s/i pairs of the individual red, green, and blue components, and M are calculated for evaluation. This calculation method follows the Lab calculation method. For **s/i** a color tolerance **siTO** can be set, and for the intensity an **M** tolerance **MTO** can be set. With the individual tolerances the color is represented as a cylinder in space (see picture below). **siTO** defines the diameter of the cylinder, and **MTO** the height of the cylinder.

#### "X Y INT - 3D":

For evaluation, **X**, **Y**, and **INT** are calculated from the individual red, green, and blue components. These three values define a point in three-dimensional space. The tolerance value that is entered defines a sphere in space with radius **TOL** (see picture below).

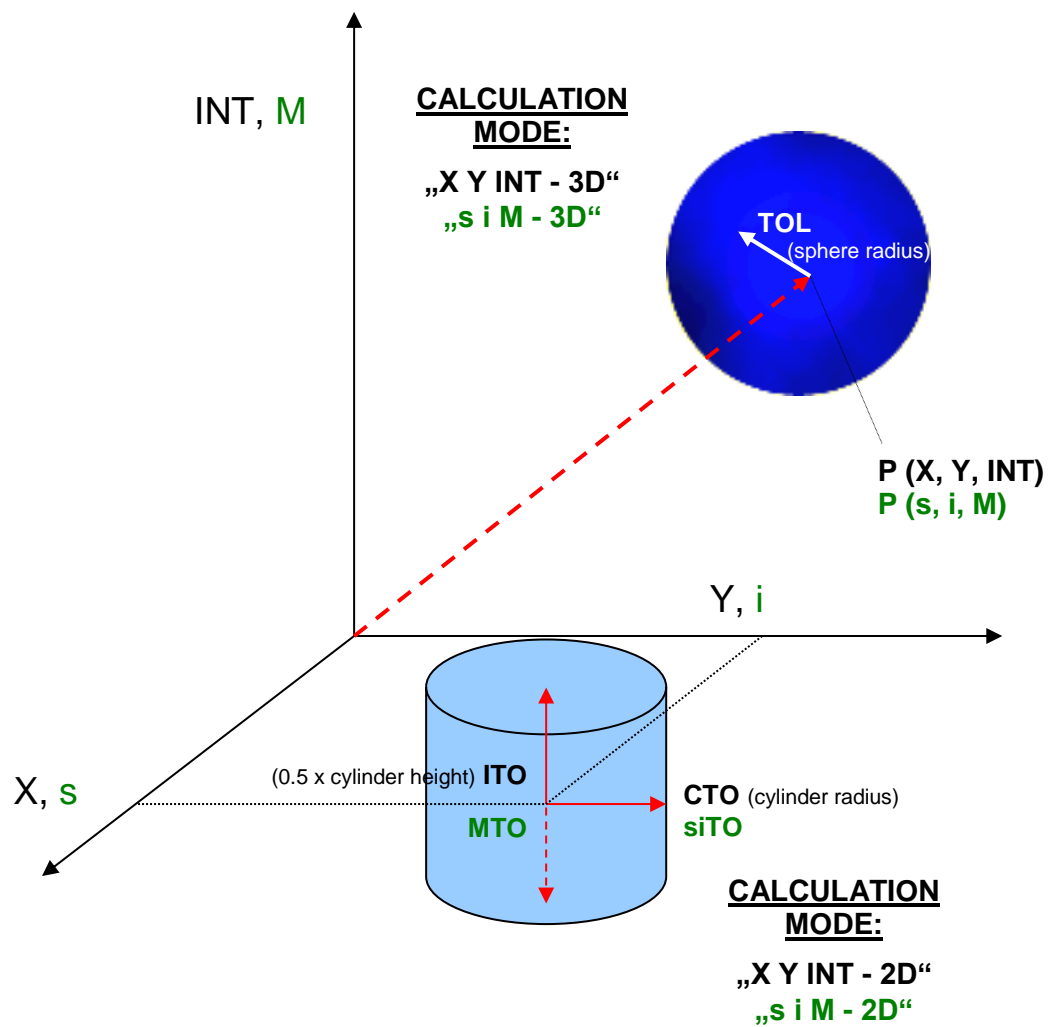
#### "s i M - 3D":

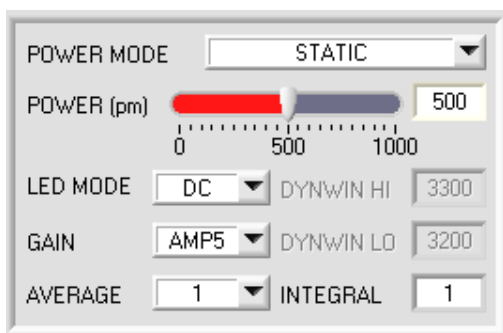
For evaluation, **s**, **i**, and **M** are calculated from the individual red, green, and blue components following the Lab calculation method. These three values define a point in three-dimensional space. The tolerance value that is entered defines a sphere in space with radius **TOL** (see picture below).

Calculation of coordinates:

|          |   |  |
|----------|---|--|
| X<br>s   | X-value of the teach color (in the color triangle the numerical value of the x-axis: RED color component)<br>$X = \frac{R}{R + G + B} * 4095$   | s is calculated according to the L*a*b* color evaluation method.<br>$s = 5000 * \left( \left( \frac{R}{4096} \right)^{1/3} - \left( \frac{G}{4096} \right)^{1/3} \right) + 5000$ |
| Y<br>i   | Y-value of the teach color (in the color triangle the numerical value of the y-axis: GREEN color component)<br>$Y = \frac{G}{R + G + B} * 4095$ | i is calculated according to the L*a*b* color evaluation method.<br>$i = 2000 * \left( \left( \frac{G}{4096} \right)^{1/3} - \left( \frac{B}{4096} \right)^{1/3} \right) + 2000$ |
| INT<br>M | Intensity value of the respective color.<br>$INT = \frac{R + G + B}{3}$   | M is calculated according to the L*a*b* color evaluation method.<br>$M = 1160 * \left( \frac{G}{4096} \right)^{1/3}$   |

|             |   |
|-------------|---|
| CTO<br>siTO | In <b>CALCULATION MODE "X Y INT - 2D"</b> or <b>"s i M - 2D"</b> , <b>CTO</b> or <b>siTO</b> is the color tolerance radius around the respective taught X/Y or s/i pair. <b>CTO</b> or <b>siTO</b> defines the radius of the color cylinder in space. In this thus-defined "tolerance circle" the current color is recognised as the teach color. A color is recognised if <b>delta C</b> is smaller than <b>CTO</b> or <b>siTO</b> and <b>INT</b> or <b>M</b> lies in the intensity window that is defined by <b>INT±ITO</b> or <b>M±MTO</b> .                     |
| ITO<br>MTO  | In <b>CALCULATION MODE "X Y INT - 2D"</b> or <b>"s i M - 2D"</b> , <b>ITO</b> or <b>MTO</b> is the intensity tolerance window around the respective taught intensity <b>INT</b> or <b>M</b> . <b>ITO</b> or <b>MTO</b> defines the height of the color cylinder in space. In this thus-defined "tolerance window" the current color is recognised as the teach color. A color is recognised if <b>delta C</b> is smaller than <b>CTO</b> or <b>siTO</b> and <b>INT</b> or <b>M</b> lies in the intensity window that is defined by <b>INT±ITO</b> or <b>M±MTO</b> . |
| TOL         | In <b>CALCULATION MODE "X Y INT - 3D"</b> or <b>"s i M - 3D"</b> , <b>TOL</b> is the tolerance radius around the respective taught point "X Y INT - 3D" or "s i M - 3D" in space. <b>TOL</b> defines the radius of the color sphere in space. In this sphere the current color is recognised as the teach color. A color is recognised if <b>delta C</b> is smaller than <b>TOL</b> .   |





#### POWER MODE:

In this function field the operating mode of automatic power correction at the transmitter unit (transmitter LED) can be set.

#### STATIC:

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.

#### DYNAMIC:

The LED transmitter power is dynamically controlled in accordance with the amount of radiation that is diffusely reflected from the object. By using the intensities measured at the receivers the automatic control circuit attempts to adjust the transmitter power in such a way that the dynamic range, which is determined by **DYN WIN LO** and **DYN WIN HI**, is not exceeded.

With **POWER MODE = DYNAMIC** the **POWER** value is used for intensity evaluation, and no longer the intensity that is calculated from the signals. One could also say that the transmitter intensity is evaluated, which corresponds with the **POWER** parameter. This is done because with **DYN WIN LO** and **HI** the sensor more or less is told to which intensity it should adjust itself. The power that flows into the sensor in order to achieve this intensity is much more meaningful than the intensity of the signal itself.

#### 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 and reaches a scan frequency of up to 35 kHz. 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.

**PULSE:** In PULSE mode the transmitter unit is pulsed, which for a very short time directs a high light intensity onto the object. Afterwards the transmitter unit must be turned off for a longer time, which compared to the **AC** mode reduces the scan frequency approximately by a factor of 5.

**OFF:** The sensor's internal light source is turned off in DC mode by **POWER [pm] = 0**, the sensor can be used for so-called "self-luminous objects". Self-luminous objects are light sources that actively emit light (LEDs, lamps, etc.). In **OFF** mode the **POWER MODE** and **POWER** cannot be adjusted, and external teaching with **DYN1** is not possible.

#### 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 (red, green, blue between 2750 and 3750).

In **AC** and **PULSE** mode, **GAIN** directly influences the scan frequency. The scan-frequency table on the software CD shows the different scan frequencies.

#### 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.

#### INFO:

The **POWER** slider is only effective in the **POWER MODE = STATIC**.

**DYN WIN LO** and **DYN WIN HI** are only effective in **POWER MODE = DYNAMIC**.



#### TRIGGER:

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

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

**SELF:** By selecting **SELF** the sensor can be operated in self-trigger mode.

The "free status" must be taught to row 0. With a split optical fibre in transmitted-light operation the free status, for example, is the uncovered status. In reflected-light operation the free status is the status when there is no part. Color detection is started when row 0 is no longer detected (self-trigger). After the trigger, i.e. when color 0 is detected again, the color from the taught colors will be output that was detected most frequently during triggering.

When the sensor leaves row 0, row 0 will be output in accordance with the output mode.

When it returns to row 0, an error will be output, or the color that was present for the longest time.

This means that after a self-trigger there always is a change of output states.

The same applies to **COLOR GROUP = ON**. When the sensor leaves group 0, group 0 will be output in accordance with the output mode. When it returns to group 0, an error will be output, or the group that was present for the longest time. This means that even if an individual color was present for the longest time it will not be output if, for example, two colors that belong to the same group together were present for a longer time. Furthermore, several colors can be combined in group 0 for the trigger condition, because the leaving of group 0 determines the trigger condition.

After dropping of the trigger an average over **X, Y, INT (or siM)** and **delta C** is output once. The average is determined from all the color hits, but not from row 0, and can be acquired with **RECORD MODE = AUTO TRIGGERED**.

**EXT1:** Color detection is started through the external trigger input (IN0 Pin3 grn at cable cab-las8/SPS) or by pressing the TEACH button. After triggering the color from the taught colors or groups in case of **COLOR GROUP = ON** will be output that was detected most frequently during triggering. Furthermore, an average over **X, Y, INT (or siM)** and **delta C** is output once. The average is determined from all the color hits and can be acquired with **RECORD MODE = AUTO TRIGGERED**.

**EXT2:** Color detection is started through the external trigger input (IN0 pin3 green of cable cab-las8/SPS) or by means of a click on the TEACH button. 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 (color no.) that was last detected will be held at the outputs.

**EXT3:** Same behaviour as in EXT2 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 or TEACH button) is given.

#### PARA:

If you work with **TRIGGER=PARA**, two parameter sets can be saved for the sensor.

**SET** can then be used to select whether the current parameters on the user interface should be saved as parameter set 0 or parameter set 1 in the sensor.

The external trigger input (IN0 pin3 grn of the cable cab-las8/SPS) or the TEACH button is used to tell the sensor which parameter set it should work with.



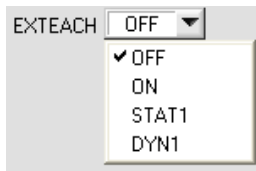
On the software interface the status of IN0 is displayed by the **TRIG** LED. If this LED is black, IN0=LO=0V, and the sensor operates with parameter set 0.

If this LED is green, IN0=HI=+24V, and the sensor operates with parameter set 1.

To watch the signal on the software interface, select 0 or 1 depending on **TRIG SET**, and then click on the **GET** button.

**SET** only is active with **TRIGGER=PARA**.





#### EXTEACH:

In all the evaluation modes teaching of a color can be performed externally through IN0 or by means of the button at the sensor housing.

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

**ON:** See below

**STAT1:** In static power mode, a color is taught to position 0 in the **TEACH TABLE**.

The **POWER MODE** is automatically set to **STATIC**. A fixed transmitter power must be set with the **POWER** slider. When the button at the sensor housing is pressed, or after a positive signal (+24V) at input IN0, the current color is taught to row 0.

The taught color is stored only in the **RAM** and not in the **EEPROM** of the sensor.

**DYN1:** In dynamic power mode, a color is taught to position 0 in the **TEACH TABLE**, and evaluation is then performed statically.

The **POWER MODE** is automatically set to **STATIC**. When the button at the sensor housing is pressed, or after a positive signal (+24V) at input IN0, the transmitter power is set such that the sensor is in the dynamic range, which is defined by **DYN WIN LO** and **DYN WIN HI**. The current color is then taught to position 0 in the **TEACH TABLE**.

The sensor then continues to operate statically with the established **POWER** value.

The taught color is stored only in the **RAM** and not in the **EEPROM** of the sensor.

**ATTENTION!** If **EVALUATION MODE = FIRST HIT**, the rows up to **MAXCOL-No.** are filled with the same teach values with **EXTEACH = ON**, **STAT1** and **DYN1**.

#### TEACH PROCESS WITH EXTEACH=ON:

Through IN0 or by way of the button at the sensor housing, the sensor can be taught up to 31 colors.

In evaluation mode **FIRST HIT** the currently present color is taught to all active rows depending on **MAXCOL-No.**

In the evaluation modes **BEST HIT**, **MIN DIST**, and **COL5** a single row in the **TEACH TABLE** can be selected with the teach button or through IN0.

The example shows the external teaching of 4 colors in **EVALUATION MODE = BEST HIT**.

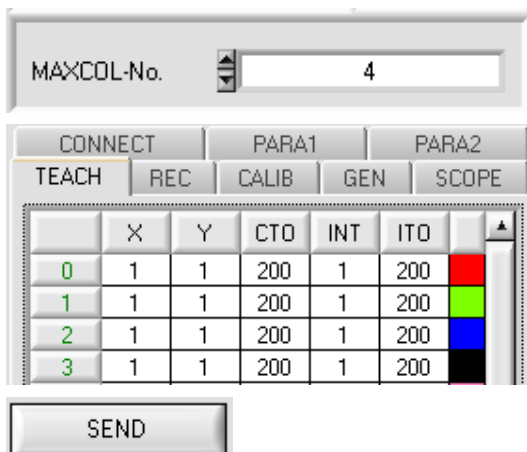
Select the **EXTERN TEACH = ON** function.

Adjust the power value such that the sensor is neither overloaded nor that a too low signal arrives at the sensor.

Select how many colors you wish to teach from extern.

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

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



In this example, **MAXCOL-No. = 4** was selected, i.e. the sensor should detect the color information that is stored in the first 4 rows of the **TEACH TABLE** by means of external teaching through IN0. Since the sensor cannot calculate the tolerances for color circle (**CTO**) and intensity (**ITO**) itself, these values must be entered once only (in this case 200 at all places) and must be stored in the **EEPROM** together with the **MAXCOL-No.** and with **EXTEACH = ON**.

Now select the setting **EEPROM** in the function field, and then 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 by way of **EXTEACH = ON** 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:

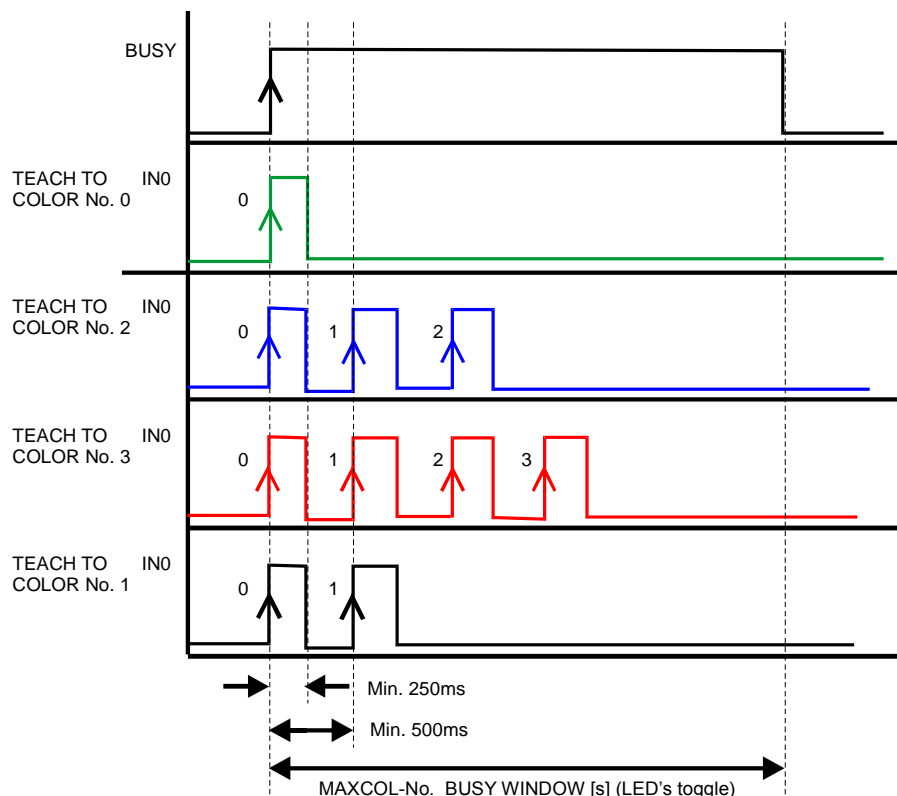
Before the external teach-process can be started, the color to be taught must be presented to the sensor. The external teach process is started with a positive edge at IN0 (green wire). When such a positive edge is applied, the output LEDs (OUT0 ... OUT4) start to blink. From now on, the user has a certain number of seconds time to inform the sensor about the position at which the color information (X,Y,INT) should be placed in the TEACH TABLE. The period of time, in which the user can teach the color information to the sensor depends on the value that is set in MAXCOL-No. (e.g. approx. 5 seconds when MAXCOL-No. = 5).

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 of the external TEACH process with a positive edge (0) at IN0 → Position 0 is selected, the LEDs start to blink.
2. An additional positive edge (1) selects position 1 in the TEACH TABLE. This position 1 is indicated by the LEDs. The LED with binary value 1 remains on HIGH level all the time, while the other 3 LEDs continue to blink.
3. An additional positive edge (2) selects position 2 in the TEACH TABLE. This position 2 is indicated by the LEDs. The LED with binary value 2 remains on HIGH level all the time, while the other 3 LEDs continue to blink.
4. An additional positive edge (3) selects position 3 in the TEACH TABLE. This position 3 is indicated by the LEDs. The LEDs with binary value 3 remain on HIGH level all the time, while the other 2 LEDs continue to blink.
5. The desired position is now selected.
6. When the BUSY window is over (the LEDs stop to blink), the sensor starts the evaluation.
7. To teach another color → go to pos. 1



INTLIM

**INTLIM:**

This edit box is used for setting an intensity limit. Color evaluation is stopped, if the current intensity **INT** arriving at the receiver unit falls below this limit, and ERROR STATE is output.

With **POWER MODE = DYNAMIC** the **POWER** value is used for intensity evaluation, and no longer the intensity that is calculated from the signals (see above). The **INT** or **M** display therefore does not show the "true" intensity, but the transmitter intensity that corresponds with the actual **POWER** parameter.

In the sensor, however, the "true" intensity is still calculated and used for the interrogation of **INTLIM**.

**Attention: ERROR STATE if:  $INT < INTLIM$**

MAXCOL-No.

**MAXCOL-No.:**

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

In the **BINARY** mode the maximum number of colors to be checked is 31. In the **DIRECT HI** or **DIRECT LO** mode the maximum number of colors to be checked is 5 (colors no. 0, 1, 2, 3, 4). 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).

OUTMODE

**OUTMODE:**

This group of buttons offers the method of how to control the 5 digital outputs.

**BINARY:**

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 to the digital outputs (OUT0 ... OUT4) as a **bit pattern**. The maximum number of colors to be taught is 31.

**DIRECT:**

In this mode the maximum number of colors to be taught is 5.

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 ... OUT4).

**DIRECT HI:**

If **DIRECT HI** is activated, the specially digital output is set to HI. If the current color does not correspond with any of the teach-in colors, all digital outputs are set to LOW (no LED is lighting).

**DIRECT LO:**

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

## 2.3 Tab PARA2

| COLOR GROUPS |       |            |
|--------------|-------|------------|
|              | GROUP |            |
| 0            | 0     | Red        |
| 1            | 0     | Green      |
| 2            | 0     | Blue       |
| 3            | 0     | Black      |
| 4            | 0     | Yellow     |
| 5            | 0     | Brown      |
| 6            | 0     | Purple     |
| 7            | 0     | Cyan       |
| 8            | 0     | Magenta    |
| 9            | 0     | Pink       |
| 10           | 0     | White      |
| 11           | 0     | Grey       |
| 12           | 0     | Dark Green |
| 13           | 0     | Dark Blue  |

| SELECT HOLD [ms] FOR EACH ROW |      |            |
|-------------------------------|------|------------|
|                               | HOLD |            |
| 255                           | 0    | Red        |
| 0                             | 10   | Green      |
| 1                             | 10   | Blue       |
| 2                             | 10   | Black      |
| 3                             | 10   | Yellow     |
| 4                             | 10   | Brown      |
| 5                             | 10   | Purple     |
| 6                             | 10   | Cyan       |
| 7                             | 10   | Magenta    |
| 8                             | 10   | Pink       |
| 9                             | 10   | White      |
| 10                            | 10   | Grey       |
| 11                            | 10   | Dark Green |
| 12                            | 10   | Dark Blue  |

C-No:  
3

GRP  
1

### PARA2:

A click on **PARA2** opens a view where colors in the **TEACH TABLE** can be assigned to certain groups, and for each color and the error status an explicit **HOLD** time can be set.

#### Please note:

**The sensor must be informed about the color groups and HOLD settings by pressing SEND.**

### COLOR GROUPS:

It is possible to form color groups in evaluation modes **FIRST HIT**, **BEST HIT**, and **MIN DIST**. This means that in a special table the individual rows are assigned to a group.

In this example, **COLOR GROUPS** has been set to **ON**, i.e. group evaluation is activated.

Rows 0 and 1 have been assigned to group 0.

Rows 2 and 3 have been assigned to group 1, and row 4 to group 2.

A **GRP** display is shown under the **C-No:** display.

If, as in this example, row 3 is detected in the evaluation, this row and the corresponding group will be visualised.

The group number will be output through outputs OUT0 to OUT4.

In evaluation mode **DIRECT HI** and **DIRECT LO** 31 different colors can be taught. However, a maximum of only 5 groups can be formed (group 0 to group 4).

A maximum of 31 groups (group 0 to group 30) can be formed in evaluation mode **BINARY**.

All the cell values can be set to 0 by pressing **RESET**.

A double-click on the **GRP** display opens a larger display window.

### HOLD in milliseconds [ms]:

The sensor operates with minimum scanning times in the magnitude of less than 100µs. This is why most of the PLCs that are connected to the digital outputs have difficulties with the safe detection of the resulting short switching state changes. Entering values in the table provides pulse lengthening at the digital outputs of the sensor system up to 100 ms. Each individual row can be assigned its own HOLD time.

When the **RESET** button is pressed, the complete table is set to the reset value indicated beside the **RESET** button.

### Example:

Print marks are transported at a very high speed. The print marks are positioned in the sensor's range only for a very short time (milliseconds or even shorter). A minimum pulse of 10 ms is required for the reliable detection of a mark.

The background is taught to row 0, and the print mark to row 1. This is desirable because it allows the use of the **BEST HIT** evaluation mode.

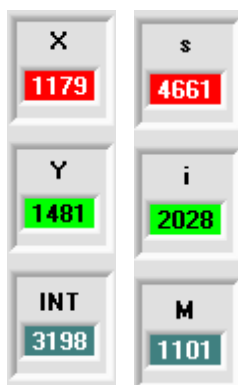
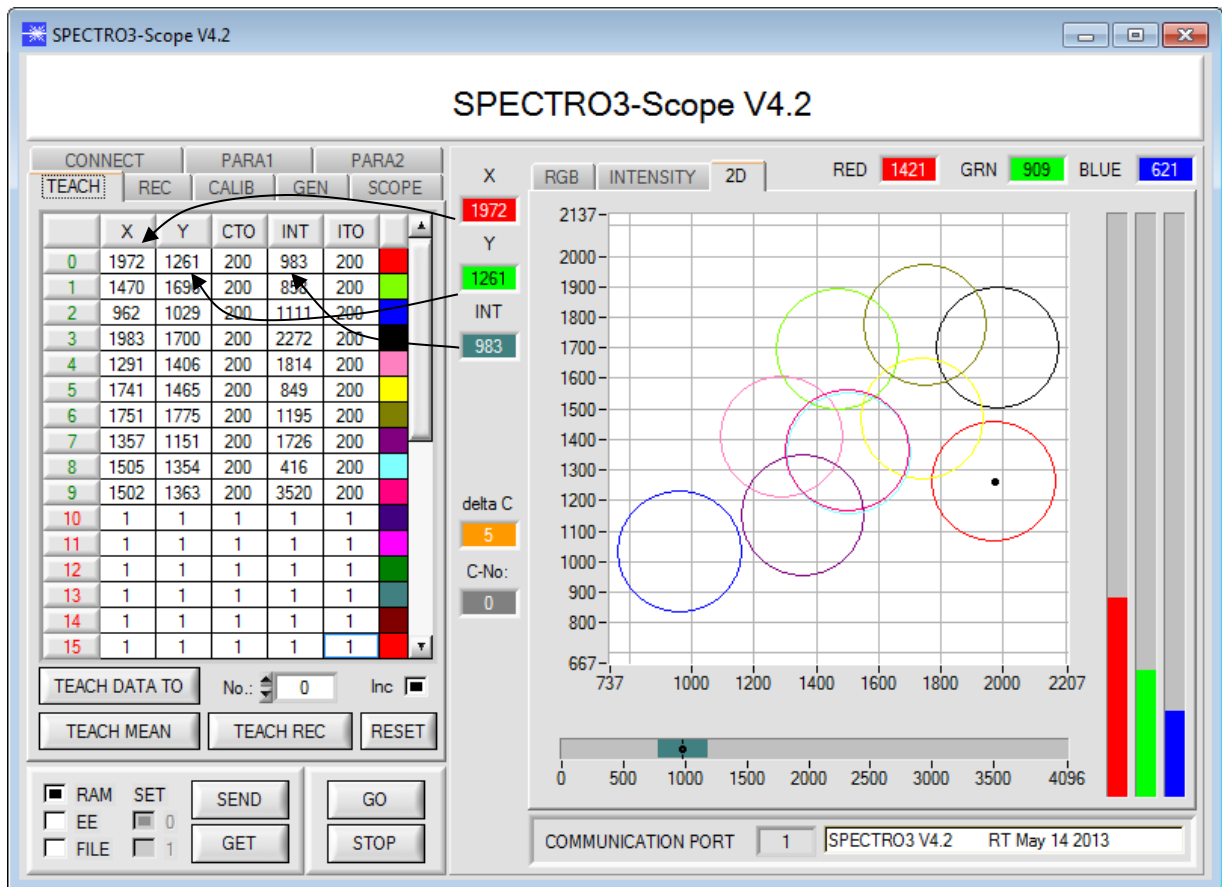
If row 0 and row 1 overlap seamlessly, the sensor operates perfectly. The output is immediately activated and remains active at least for the time specified in the corresponding row.

If row 0 and row 1 do not overlap, the sensor detects an error at the transition from row 0 to row 1 (print mark), which immediately is sent to the output and remains active there at least for the **HOLD** time. An error inevitably would occur here if **HOLD** for the error state (255) was chosen too high, because the sensor only outputs a status change again after **HOLD** has elapsed (this may be the print mark, but it does not have to be).

It is absolutely necessary here to select a **HOLD** time of 0 for error state 255.

## 2.4 Tab TEACH

When the **GO** button is pressed, data transfer from the sensor to the PC is started. The respective RED, GREEN, and BLUE contents are shown by way of the bars beside the graph. The calculated **X**, **Y**, **INT** or **s,i,M** values are visualised in the displays.



### **X or s:**

This numerical value output field displays the RED content (x-axis) of the scattered light currently arriving at the receiver.

### **Y or i:**

This numerical value output field displays the GREEN content (y-axis) of the scattered light currently arriving at the receiver.

### **INT or M:**

This numerical value output field displays the currently measured intensity (proportional to the average of the intensities at the triple receiver).



### **C-No.:**

This numerical value output field displays the currently detected color number in accordance with the entry in the TEACH TABLE. The currently detected color number is sent to the digital outputs OUT0 ... OUT4 as a corresponding bit pattern.

The value 255 means that none of the taught colors is recognised.

A double-click on the display opens a larger display window.

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

**TEACH**

|    | X    | Y    | CTO | INT  | ITO |             |
|----|------|------|-----|------|-----|-------------|
| 0  | 1972 | 1261 | 200 | 983  | 200 | Red         |
| 1  | 1470 | 1696 | 200 | 858  | 200 | Green       |
| 2  | 962  | 1029 | 200 | 1111 | 200 | Blue        |
| 3  | 1983 | 1700 | 200 | 2272 | 200 | Black       |
| 4  | 1291 | 1406 | 200 | 1814 | 200 | Pink        |
| 5  | 1741 | 1465 | 200 | 849  | 200 | Yellow      |
| 6  | 1751 | 1775 | 200 | 1195 | 200 | Olive       |
| 7  | 1357 | 1151 | 200 | 1726 | 200 | Purple      |
| 8  | 1505 | 1354 | 200 | 416  | 200 | Cyan        |
| 9  | 1502 | 1363 | 200 | 3520 | 200 | Magenta     |
| 10 | 1    | 1    | 1   | 1    | 1   | Dark Purple |
| 11 | 1    | 1    | 1   | 1    | 1   | Bright Pink |
| 12 | 1    | 1    | 1   | 1    | 1   | Dark Green  |
| 13 | 1    | 1    | 1   | 1    | 1   | Dark Red    |
| 14 | 1    | 1    | 1   | 1    | 1   | Dark Blue   |
| 15 | 1    | 1    | 1   | 1    | 1   | Dark Red    |

TEACH DATA TO No.: 0 Inc ☐

TEACH MEAN TEACH REC RESET

#### TEACH:

A click on the **TEACH** tab opens a view where colors can be taught to the **TEACH TABLE**.

**Please note: The sensor must be informed about the teach colors by pressing SEND.**

After a left mouse button double click (or a click on shortcut key button F2) on the respective field the default values can be changed by entering numerical values with the PC keyboard.

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

The sensor is able to check up to 31 teach-in colors. The number of the respective teach-in color is given in the left column of the table.

Only the green rows are used for evaluation in the sensor. The number of rows to be inspected is set with **MAXCOL-No.**

When **TEACH DATA TO** is pressed, the currently displayed data for **X, Y, INT or s, i, M** are transferred to the row in the **TEACH TABLE** that is selected under **No.:**

A Color Tolerance (**CTO**) and an Intensity Tolerance (**ITO**) also are set. If necessary, these tolerances, as well as the teach values, can be changed as described above.

**No.:** also is used to select which **INT** or **M** tolerance window is shown in the respective graphs for the intensity or for M.

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 = 1)

|   | X    | Y    | INT  | TOL |   |
|---|------|------|------|-----|---|
| 0 | 1996 | 1171 | 827  | 200 | 1 |
| 1 | 1495 | 1673 | 789  | 200 | 1 |
| 2 | 958  | 921  | 1027 | 200 | 1 |
| 3 |      |      |      |     |   |
| 4 |      |      |      |     |   |
| 5 |      |      |      |     |   |
| 6 |      |      |      |     |   |
| 7 |      |      |      |     |   |
| 8 |      |      |      |     |   |
| 9 |      |      |      |     |   |

Help  
Set selection to  
Increment selection with  
Reset selection  
Teach tolerance off  
Teach tolerance on

**VALUE!**

Insert a cell value!

200

OK

With Set selection to several cells in the **TEACH TABLE** can be filled with a value.

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

A right mouse click then opens a popup menu.

If you click on **Set selection to** there, a popup window will open where you can enter the value to which the cells should be changed.

|   | X    | Y    | INT  | TOL |   |
|---|------|------|------|-----|---|
| 0 | 1996 | 1171 | 827  | 200 | 1 |
| 1 | 1495 | 1673 | 789  | 250 | 1 |
| 2 | 958  | 921  | 1027 | 300 | 1 |

**VALUE!**

Insert a cell value!

OK

Help  
Set selection to  
Increment selection with  
Reset selection  
Teach tolerance off  
Teach tolerance on

Incrementing tolerance values can be entered by using **Increment selection with**.

The start value is the value of the topmost left cell.

Starting from this cell the following cells are increased each by the value that is entered in the popup window.

A click on **Reset selection** resets the selected cells to a value of 1.

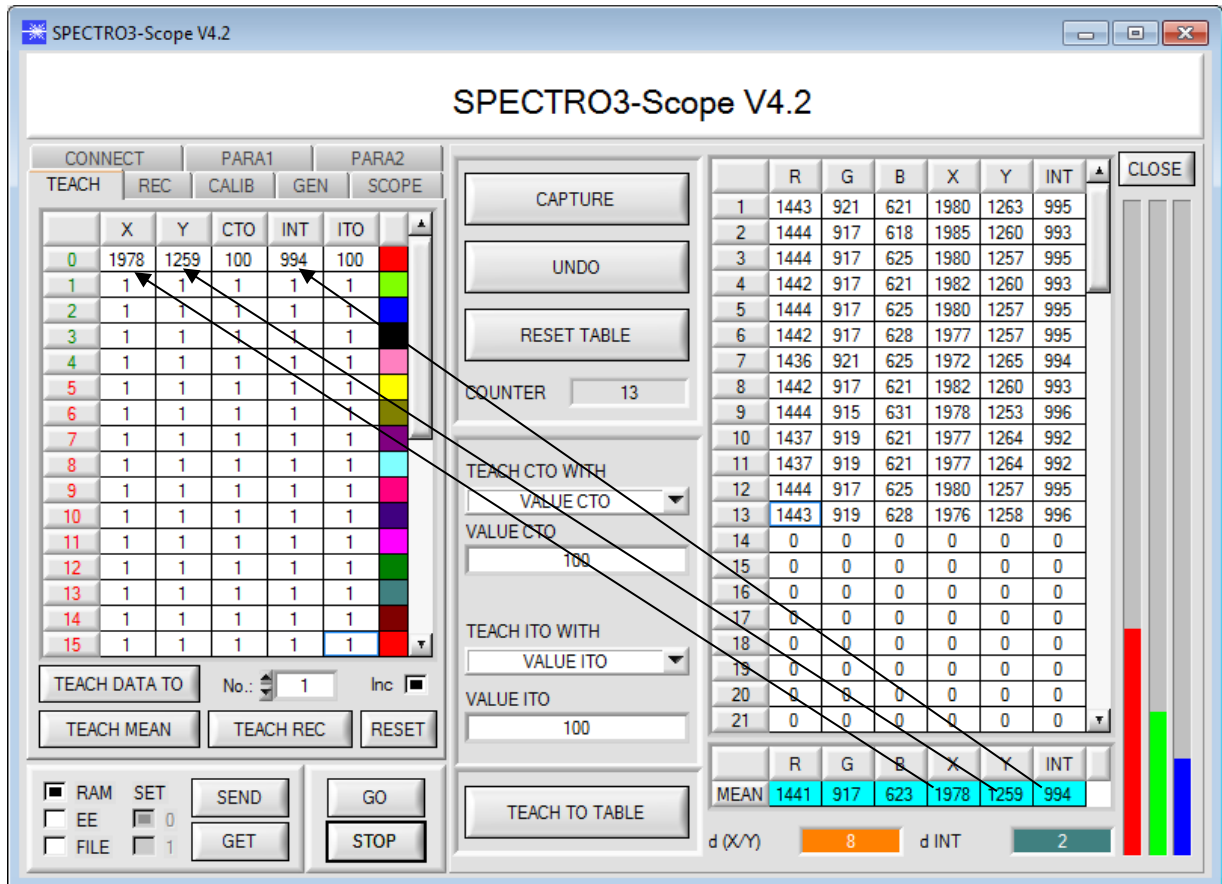
When you use **TEACH DATA TO** the software suggests tolerance values and writes these values to the respective cells.

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

A click on **Help** opens a popup window that provides explanations for the respective functions.

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

The Teach Panel can be used in every **EVALUATION** and **CALCULATION MODE**. The explanation herein uses the **EVALUATION MODE=BEST HIT** and the **CALCULATION MODE="X Y INT - 2D"**.



Data are automatically gathered 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 deviation **d(X/Y)** and a maximum intensity deviation **dINT** for the average values will be formed.

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.:** (→ 2.3 Tab TEACH TABLE).

Teaching of the circular tolerance and of the intensity tolerance can be set by way of **TEACH CTO WITH** and **TEACH ITO WITH**, respectively.

- If the setting is **VALUE CTO**, the value that is set under **VALUE CTO** will be taught (ditto intensity).
- If the setting is **d(X/Y)**, the value that is determined under **d(X/Y)** will be taught (ditto intensity).
- If the setting is **d(X/Y) + VALUE CTO**, the value that is determined under **d(X/Y)** plus **VALUE CTO** will be taught (ditto intensity).
- With **NO CHANGE**, the value that is set in the **TEACH TABLE** will remain unchanged.

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

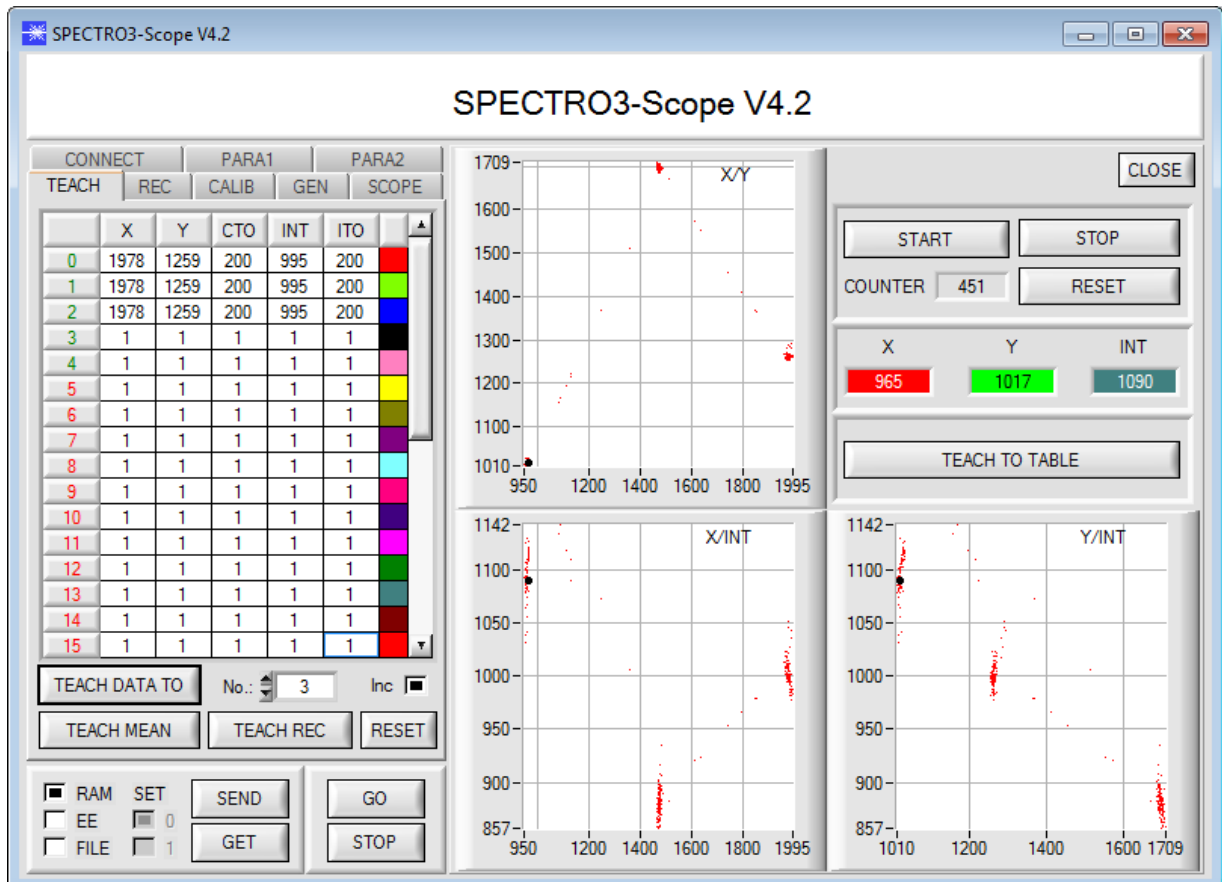
A click on the button **TEACH REC VAL** (teach recorded values) opens a panel on the right side which, after a click on the **START** button, starts to record data and display them in the three graphs. This function is useful if the material to be detected cannot be directly placed in front of the sensor, 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 4 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.:**.

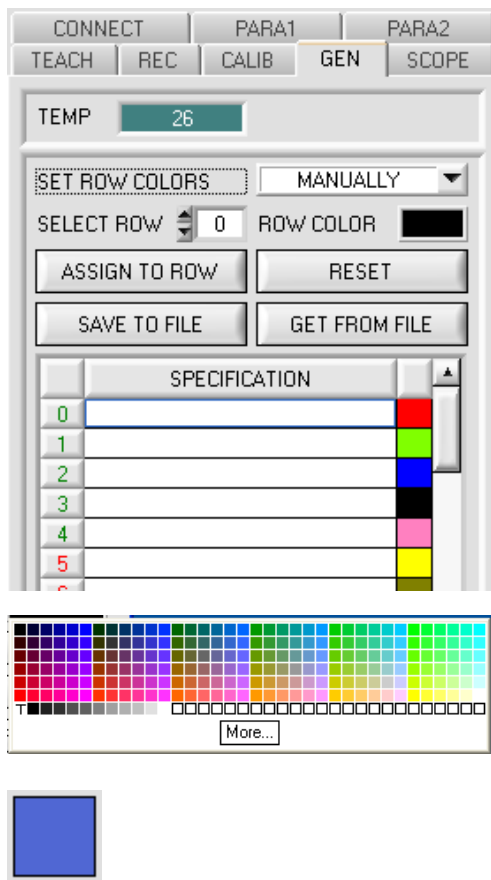
**RESET** can be used to reset the graphs and the counter.

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





## 2.5 Tab GEN



### 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 COLOR** is set to **MANUALLY**, the row color that should be changed must be set under **SELECT ROW**.

A click on the colored area of **SELECT 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 COLOR** 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.

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

The file that is created has 13 columns and as many rows as data frames were recorded.

A row is structured as follows:

**Date, time, RED, GREEN, BLUE, X, Y, INT, delta C, COLOR, GROUP, TRIGGER, TEMP**

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

**Please note:**

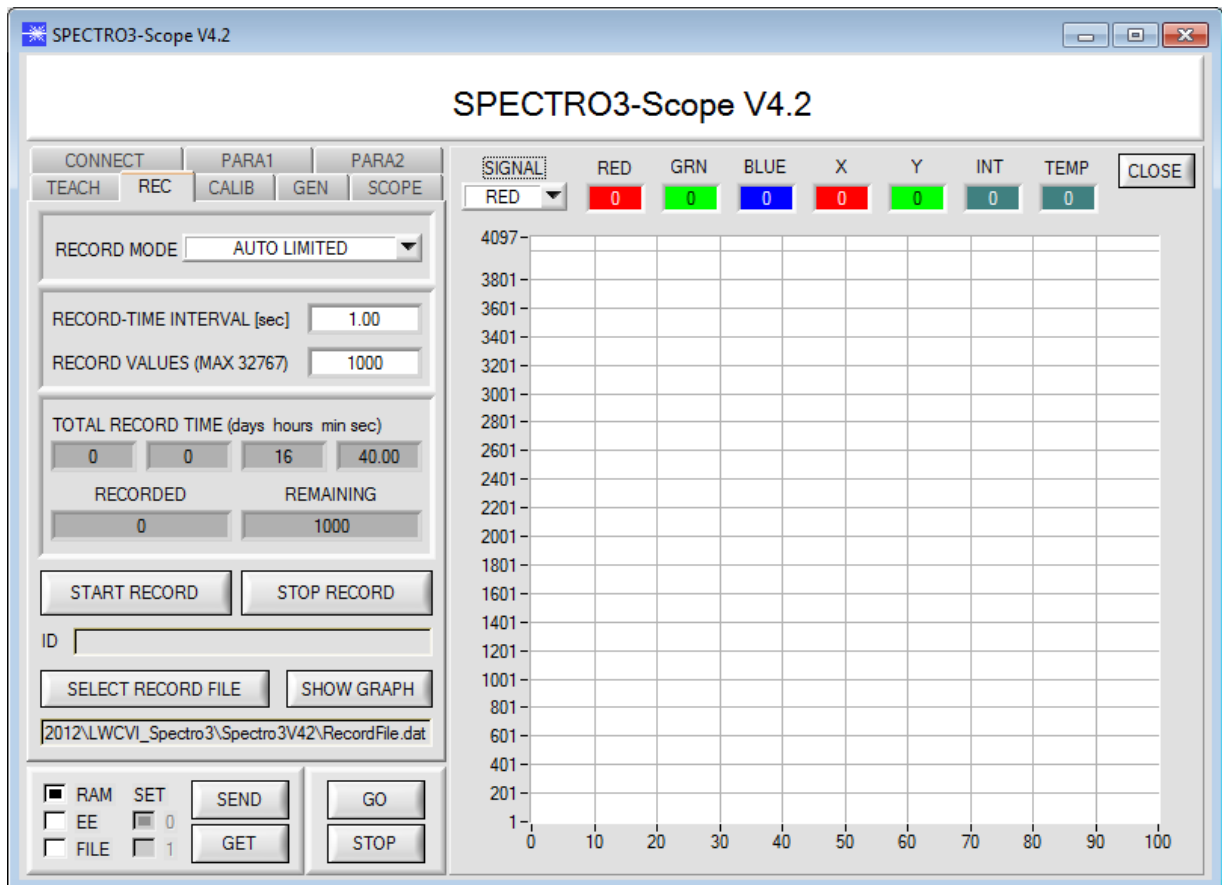
Recording depends on the **EVALUATION MODE** that is chosen. Several **EVALUATION MODES** do not need certain data, therefore these data will be set to 0, i.e. for these data the value 0 will be recorded.

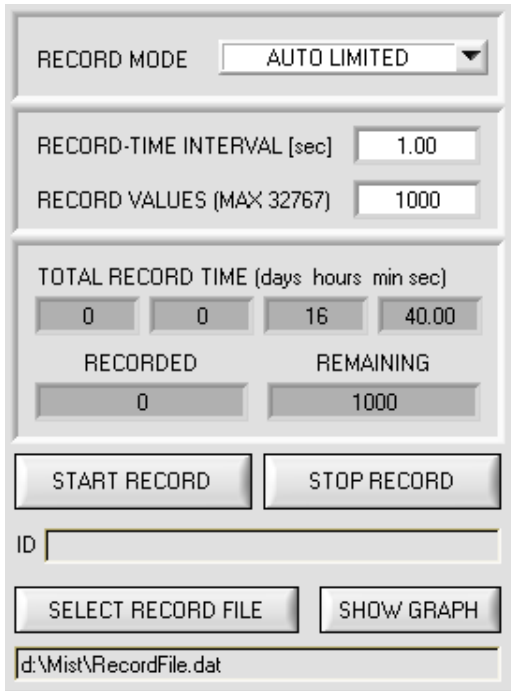
**Step 1:**

When the **RECORDER** button is pressed, the following window will be displayed:

When the **SHOW GRAPH** button is pressed, a panel will be displayed that allows the user to monitor the different signals.

The individual signals can be activated from the **SIGNAL** drop-down menu.





#### 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 = SELF, EXT1, EXT2, EXT3, TRANS** or **PARA** 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:

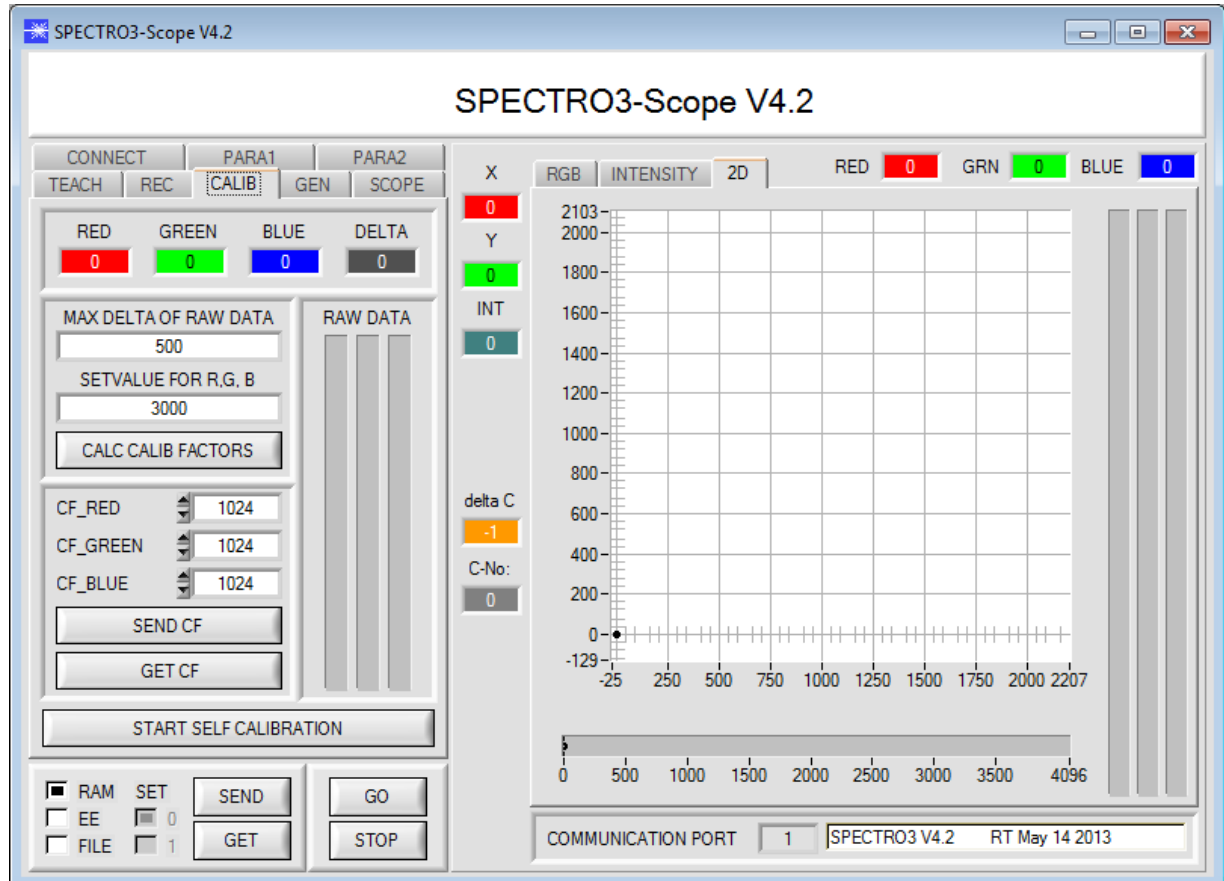
When you press **START RECORD**, the file that is selected under **SELECT RECORD FILE** will be deleted. With **RECORD FRAME MANUALLY**, the file will be created if it does not already exist. If the file already exists, the data are added to the existing file.

## 2.7 Tab CALIB

### 2.7.1 White light balancing

White light balancing can be performed with the sensors of the SPECTRO-3 series. Balancing can be performed to any white surface. A ColorChecker™ table with 24 color fields according to CIE standard is available as an alternative, and white light balancing or calibration can then be performed to one of the white fields

The following panel will be displayed after a click on **CALIBRATE**:



### Calculation example for determining the calibration factors

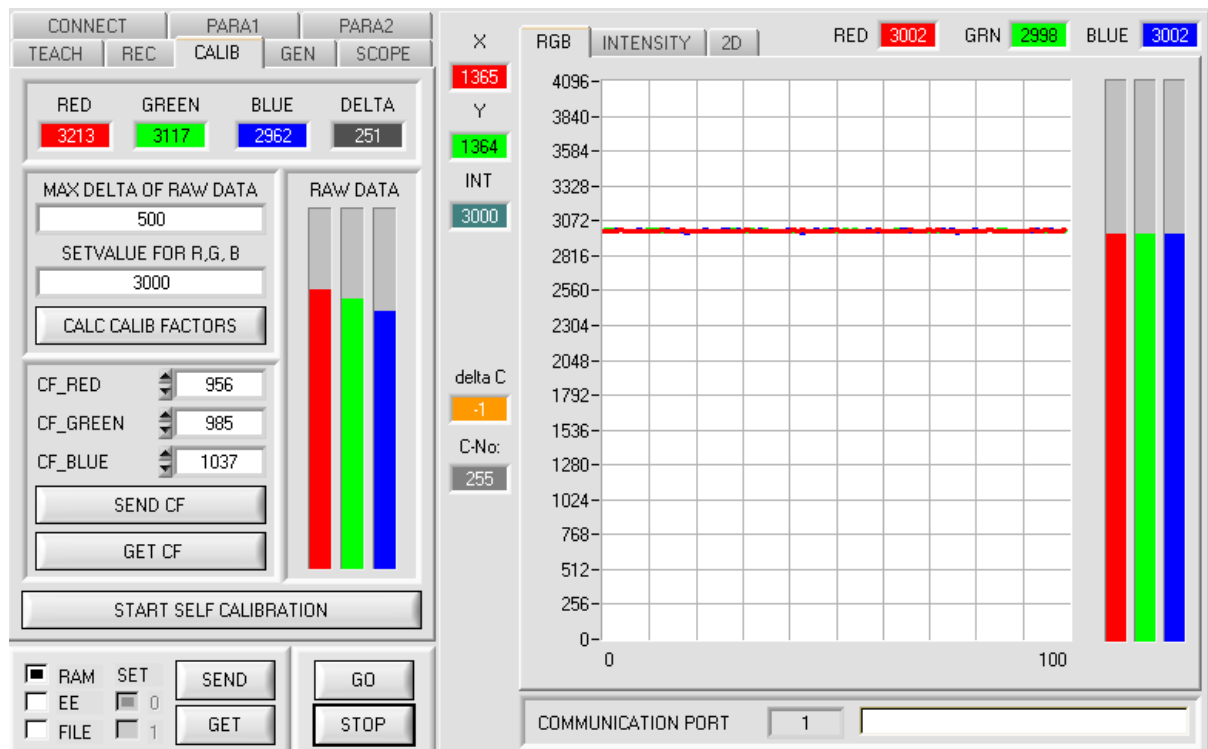
In the example in the picture below, a POWER value at which the three bars of the raw signals **RAW DATA** are in the dynamic range has been set. Each of the three bars is at approx. 3000 digits. It is thus appropriate to set a setpoint value of 3000 (see **SETVALUE FOR R,G,B**) for the three bars. When calibration is now started by pressing **CALCULATE CALIBRATION FACTORS**, the software automatically calculates the calibration factors for channel RED, channel GREEN, and channel BLUE. The calibration factors are normalized as integers to the value 1024.

Formula:

$$CF\_RED = (SETVALUE / RAW DATA RED) * 1024 = (3000 / 3213) * 1024 = 956$$

$$CF\_GREEN = (SETVALUE / RAW DATA GREEN) * 1024 = (3000 / 3117) * 1024 = 985$$

$$CF\_BLUE = (SETVALUE / RAW DATA BLUE) * 1024 = (3000 / 2962) * 1024 = 1037$$



When the calibration factors have been calculated by the software on the user interface, they are automatically saved to the non-volatile **EEPROM** memory of the sensor. Calibration is then finished, work can then be continued in the main panel.

When the sensor detects a raw signal, it applies the calibration factor saved in the **EEPROM** to this raw signal, i.e. in the main panel only the calibrated data for the RED, GREEN, and BLUE channels are displayed. Evaluation by the micro-controller also is exclusively done with the calibrated data.

In the following the individual steps for calibrating the sensors are described:

**INFO:** The individual pop-up windows are intended as a help to guide you through the calibration process.

**ATTENTION:** It is a prerequisite for successful calibration that the sensor front-end is calibrated to a white surface.

**Step 1:**

First of all a suitable **POWER** value must be found such that the **RAW DATA** for RED, GREEN and BLUE lie in the dynamic range (upper third of the bar display).

**Step 2:**

When you have set a suitable **POWER** value, determine a **SETVALUE FOR R,G,B**. The software now calculates the calibration factors in such a way that this **SETVALUE** is reached for the raw data (see calculation example above).

**Step 3:**

Determine a **MAX DELTA OF RAW DATA** (the software suggests 500).

Calibration is only permitted, if the current **DELTA** of the **RAW DATA** is smaller than the **MAX DELTA OF RAW DATA**.

**DELTA** is the maximum of **RED**, **GREEN**, and **BLUE**, minus the minimum of **RED**, **GREEN**, and **BLUE**. This is necessary in order to ensure that the sensor functions properly and calibration is performed on a white surface.

**Step 4:**

Start calibration by pressing **CALC CALIB FACTORS**.

The button starts to flash in red, and at the same time 100 raw data are recorded through the interface, of which the respective mean value of **RED**, **GREEN**, and **BLUE** is formed.

The individual calibration factors are formed from these mean values and from the **SETVALUES FOR R,G,B** and they are then entered in the corresponding edit-boxes.

The calibration software automatically saves the calculated calibration factors to the EEPROM of the sensor.

Then the software changes to the GO mode and displays the **RAW DATA** and the calibrated data in the main panel.

Please note that the values for **RED**, **GREEN**, and **BLUE** in the main panel approximately are equal to the value of **SETVALUE**.

You may also change the calibration factors **CF\_RED**, **CF\_GREEN**, **CF\_BLUE** manually by entering new values in the corresponding input fields. Please note that these factors are saved to the EEPROM by pressing **SEND CF**. **GET CF** reads the calibration factors that are currently saved in the EEPROM.

If pressing **CALC CALIB FACTORS** should not be successful, please follow the information provided in the pop-up windows.

Calibration only is completed successfully, if the following pop-up window is displayed:



A click on **START SELF CALIBRATION** causes the sensor to calculate the calibration factors itself. It is not possible to specify a **SETVALUE** and a **MAX DELTA** here.

When the sensor has calculated the calibration factors, it displays them on the software interface. In the corresponding input fields it also displays the **SETVALUE** that it used for calculation and the **MAX DELTA** value that resulted from calculation. The **SEND CF** button must be pressed to confirm the calculated calibration factors.

## 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 at any place between the individual elements in the **CALIBRATE** tab.

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

Now please follow the instructions provided in the tab.

### ATTENTION!

It is extremely important for offset calibration that the receiver is not exposed to any extraneous light. Please make sure that you cover the receiver of the sensor e.g. 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.

The display shows depending on the **TRIG MODE** settings either the signals **R G B**, or **X Y INT**, or **s i M** as well as the status of the digital outputs and of the digital input IN0.

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

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

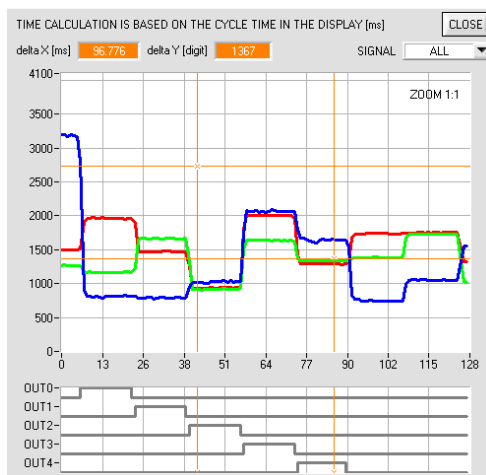
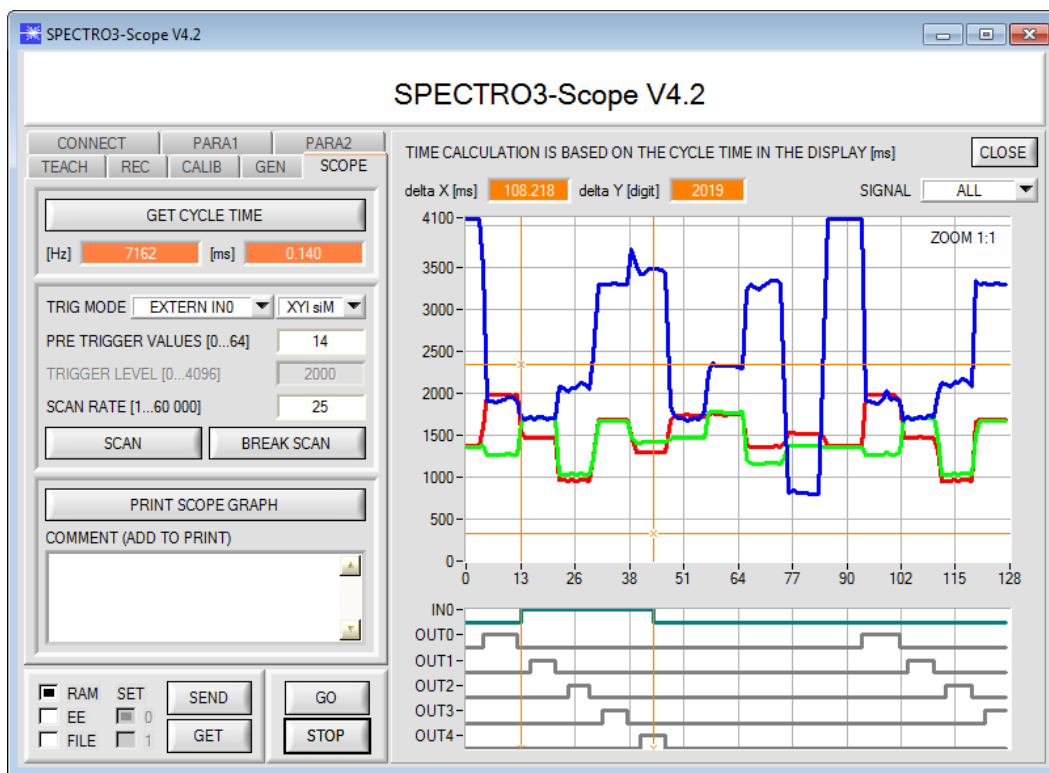
In **TRIG MODE = FALLING EDGE** and **RISEING EDGE** a click on **SCAN** starts triggered recording. A trigger start can be defined with **TRIGGER LEVEL**.

Triggering is done either to **BLUE**, **INT** or **M**, depending on which signal shall be recorded (**R G B**, or **X Y INT**, or **s i M**). In the graph this is the blue line.

In **TRIG MODE= INTERN C-No.0** recording starts stand-alone once C-No. 0 is detected.

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

**SCAN-RATE** can be used to delay or accelerate recording. This corresponds with the TIMEBASE function known in oscilloscopes. **PRE TRIGGER VALUES** can be used to define how many values should still be displayed before the actual trigger start.



The zoom function in the graph can be activated by holding the control key (CTRL) and drawing a window with the mouse. A click on **ZOOM 1:1** cancels the zoom function again.

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

**deltaX[ms]** shows the time between the cursors in X direction. **deltaY[digit]** or [V] show the difference between the two cursors in digits or in Volt in Y direction.

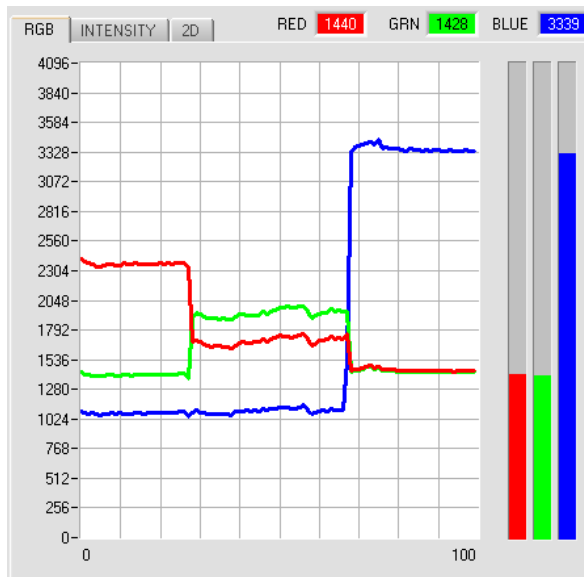
**SIGNAL** can be used to display individual curves.

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

## 2.9 Graphic display elements

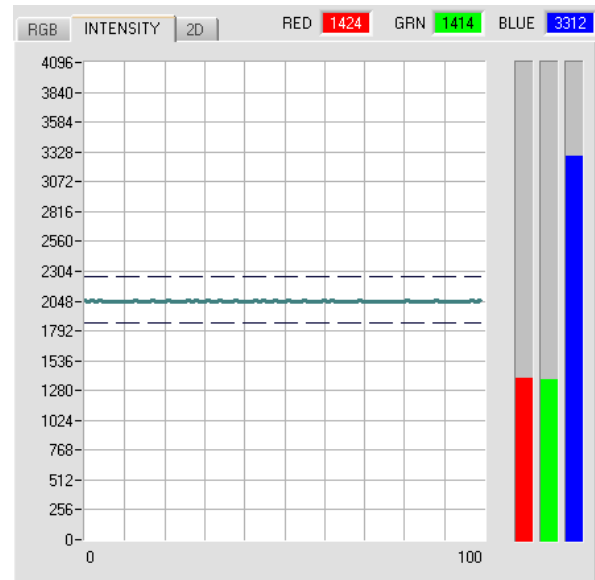
### RGB tab:

Display of the current raw signals of the 3-fold receiver (red, green, blue).



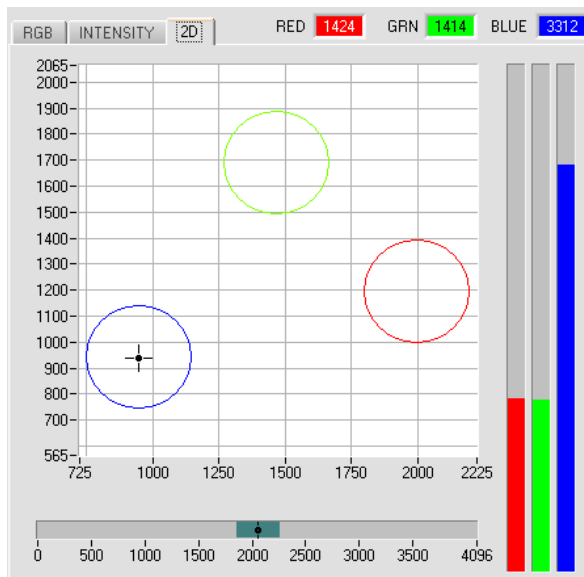
### INTENSITY tab:

Display of the currently determined intensity **INT** or **M**.



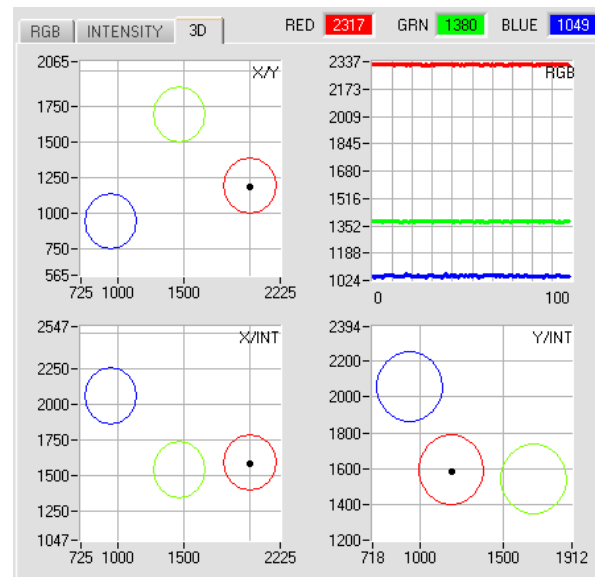
### 2D tab:

Display of **X/Y** or **s/i** pairs in a zoomed graph. The intensity **INT** or **M** with the tolerance window set under No.: is shown directly below this. 2D view of the color cylinder in space.



### 3D tab:

Display of the taught color spheres and of the current color position. A 3D view with the graphs **X/Y** (**s/i**), **X/INT** (**s/M**) and **Y/INT** (**i/M**) was chosen for the purpose of improved representation.





TEMP 27

**TEMP:**

This display shows the temperature prevailing in the sensor housing.  
 The display **DOES NOT** show degrees Centigrade or Fahrenheit.

delta C  
-1

**delta C:**

This display (delta color) shows the deviation from a color hit, **delta C** corresponds with  $\Delta E$  that is calculated in a color measurement.

In **FIRST HIT** evaluation mode **delta C** represents the distance from a color hit. When no color has been hit, **delta C** is calculated in relation to the last valid color in the color table (depending on **MAXCOL-No.**).

In **BEST HIT** and **MIN DIST** evaluation mode **delta C** also represents the distance from a color hit. If, however, no color has been recognised, **delta C** assumes the value of -1.

In **COL5** evaluation mode **delta C** is not calculated.

When the current intensity (**INT** or **M**) falls below the **INTLIM** parameter, **delta C=-1** will be displayed.

In **CALCULATION MODE "X Y INT - 2D"** and **"s i M - 2D"** **delta C** represents the distance from the cylinder axis and is calculated as follows:

$$\text{delta}C = \sqrt{X^2 + Y^2} \quad \text{or} \quad \text{delta}C = \sqrt{s^2 + i^2}$$

In **CALCULATION MODE "X Y INT - 3D"** and **"s i M - 3D"** **delta C** represents the distance from the centre of the sphere and is calculated as follows:

$$\text{delta}C = \sqrt{X^2 + Y^2 + INT^2} \quad \text{or} \quad \text{delta}C = \sqrt{s^2 + i^2 + M^2}$$

### 3. Short instructions for the operation of color sensors with the SPECTRO3-Scope V4.2 software

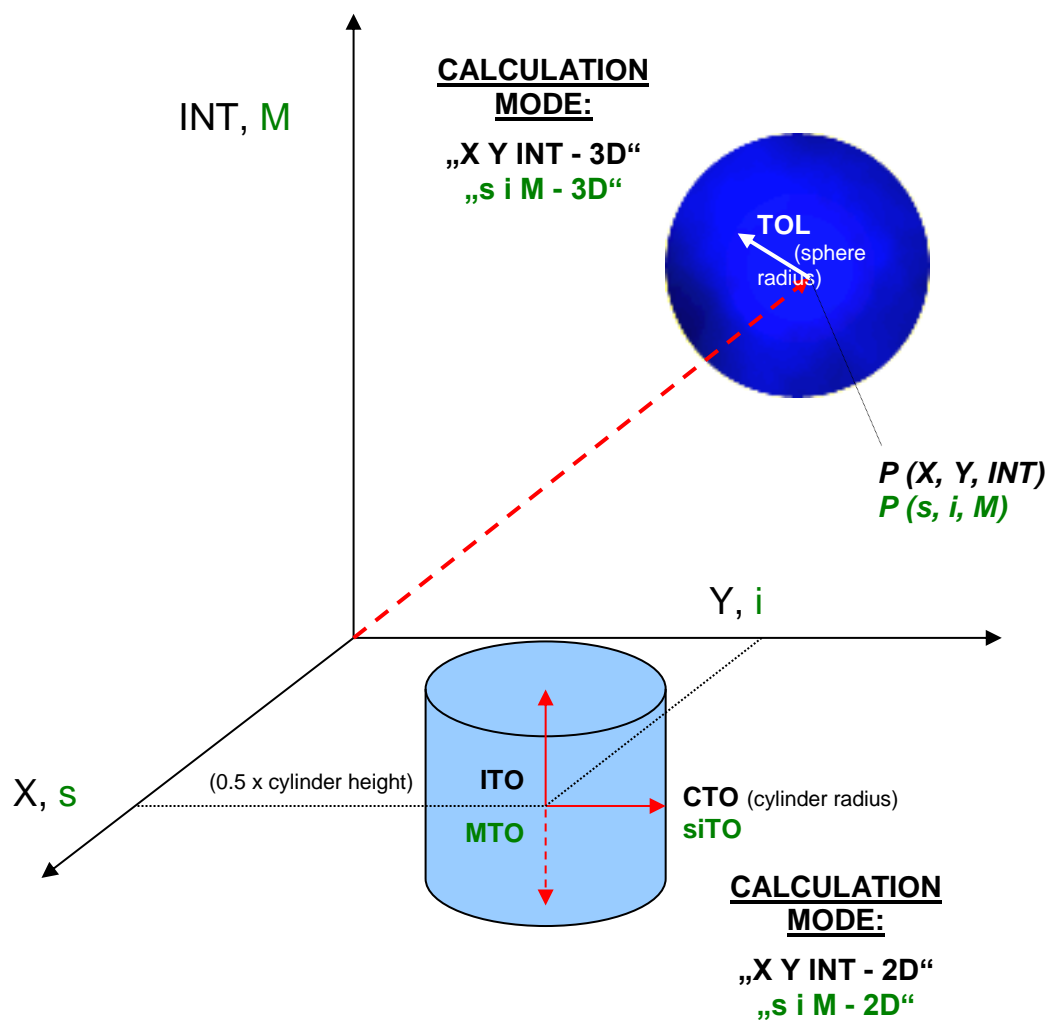
These instructions describe how to perform quick teaching of the color sensors of SPECTRO-3 series with the **SPECTRO3-Scope V4.2** software interface.

Basically there are 2 methods of teaching a color. These methods can be set with **CALCULATION MODE**.

The **CALCULATION MODE "X Y INT - 3D"** (or "**s i M - 3D**") uses a **color sphere** in space with radius **TOL**. Contrary to this, the **CALCULATION MODE "X Y INT - 2D"** (or "**s i M - 2D**") uses a **color cylinder** in space with radius **CTO** or **siTO** and with height **ITO** or **M**.

The teach process is the same for both methods.

Color evaluation according to "**s i M - 2D**" uses the Lab calculation method.



The text below describes a teach process with **EVALUATION MODE = BEST HIT** and **CALCULATION MODE = "X Y INT - 3D"**.

#### Step 1:

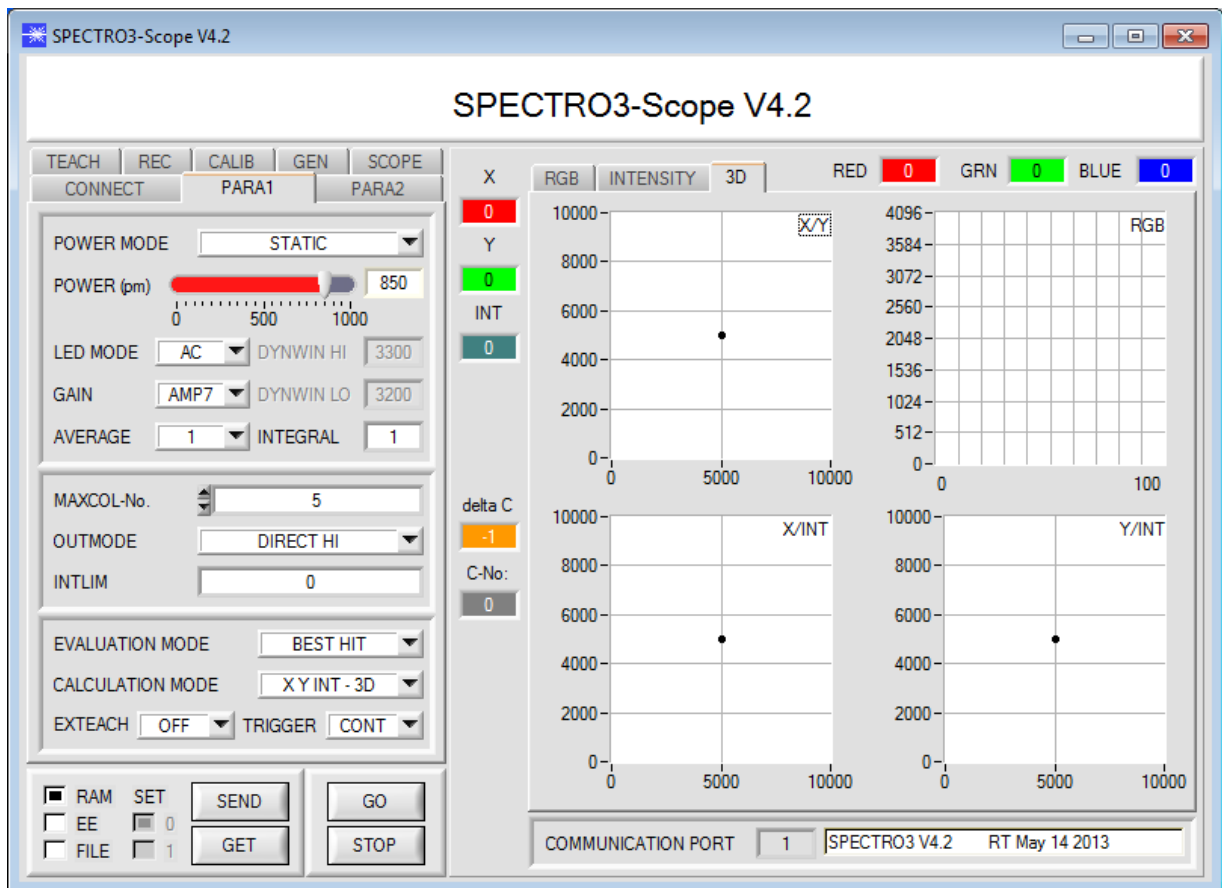
Prior to the use of the software aids (graphic display of sensor signals) the sensor must be manually adjusted to the respective target or background as accurately as possible. The reference distance of the sensor to the target is defined in the data sheet of the respective sensor.

Please make sure that the sensor is properly connected and supplied with power. Furthermore the sensor head must be directed onto the brightest surface (paper, leather, glass, etc.) of the color to be taught. This is absolutely necessary for being able to set a suitable **POWER** value for the sensor.

#### Step 2:

Start the SPECTRO3-Scope V4.2 software. Please check whether the status line at the right bottom displays the "SPECTRO3 V4.2 xx xxx xx xxxx " message.

**Info:** Moving the mouse cursor to a control element and clicking with the right mouse button displays a short info on the respective individual control element.



### Step 3:

Please make sure that for the time being **RAM** and not **EEPROM** is selected for the data exchange with the sensor (RAM is a volatile memory in the sensor, i.e. the data will be lost when power is turned off. EEPROM is a non-volatile memory in the sensor, i.e. the data will not be lost when power is turned off.)

All the other parameters should be set as shown in the picture below. Press the tab **PARA1** to show the parameters. (See the **FILE** function in the operating instructions for SPECTRO3-Scope.)

Now press the **GO** button. Data exchange between sensor and PC will then start. The contents for **RED**, **GREEN**, and **BLUE** that are diffusely reflected from the surface are represented as a graph and as displays on the software interface.

Set the **POWER** value and **GAIN** such that at least one of the three channels of the brightest color that should be taught lies in the upper third of its dynamic range, but none is in saturation. Ideally **POWER** and **GAIN** are set in such a way that **POWER** lies in the range of 300 to 900.

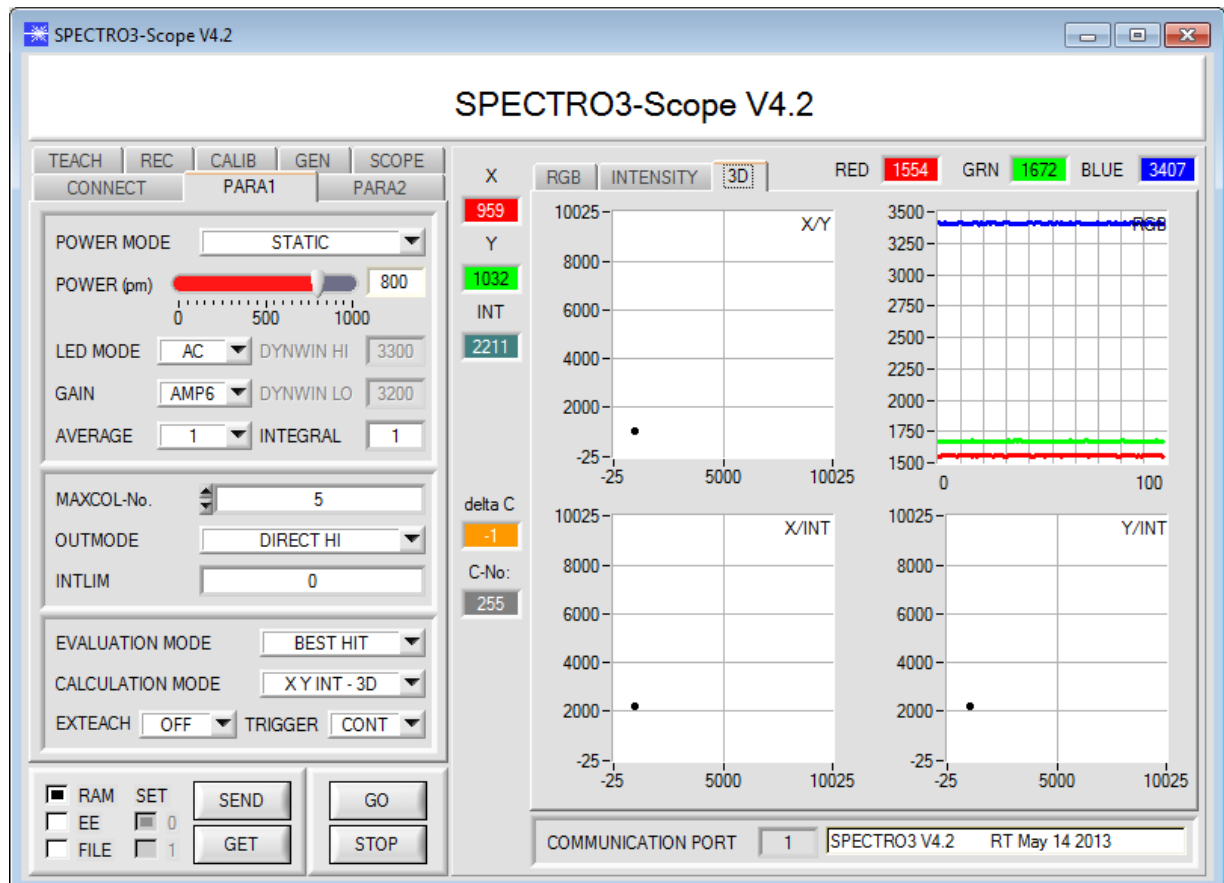
**ATTENTION:** The sensor must be informed when you have changed the **POWER** or **GAIN** value. Press the **SEND** button to send this information to the sensor. Check the bars by pressing **GO** again. Repeat this process until you have a suitable **POWER** value and **GAIN**.

Optimally the three bars should be at approximately the same level on white paper.

If the bars vary too much, you need to perform white light balancing.

(For white light balancing see the SPECTRO3-Scope operation instructions on the supplied CD.)

**Tip!** There is a trick for finding a suitable **POWER** value in no time at all. Show the sensor the brightest color that should be taught, and set **POWER MODE = DYNAMIC**. The sensor then tries to find a suitable **POWER** value. Check this by pressing **GO**. Press **STOP** when the channels have "levelled out". Then press **GET**. The **POWER** value that was found now is shown in the **POWER** function field. Now set **POWER MODE = STATIC**, and press **SEND**.



#### Step 4:

An **X** and a **Y** coordinate as well as an intensity **INT** are calculated from the red, green, and blue data.

$$X = (R \cdot 4095) / (R + G + B)$$

$$Y = (G \cdot 4095) / (R + G + B)$$

$$INT = (R + G + B) / 3$$

Press the **GO** button once again to read out the current data.

Press **STOP** to stop the data exchange.

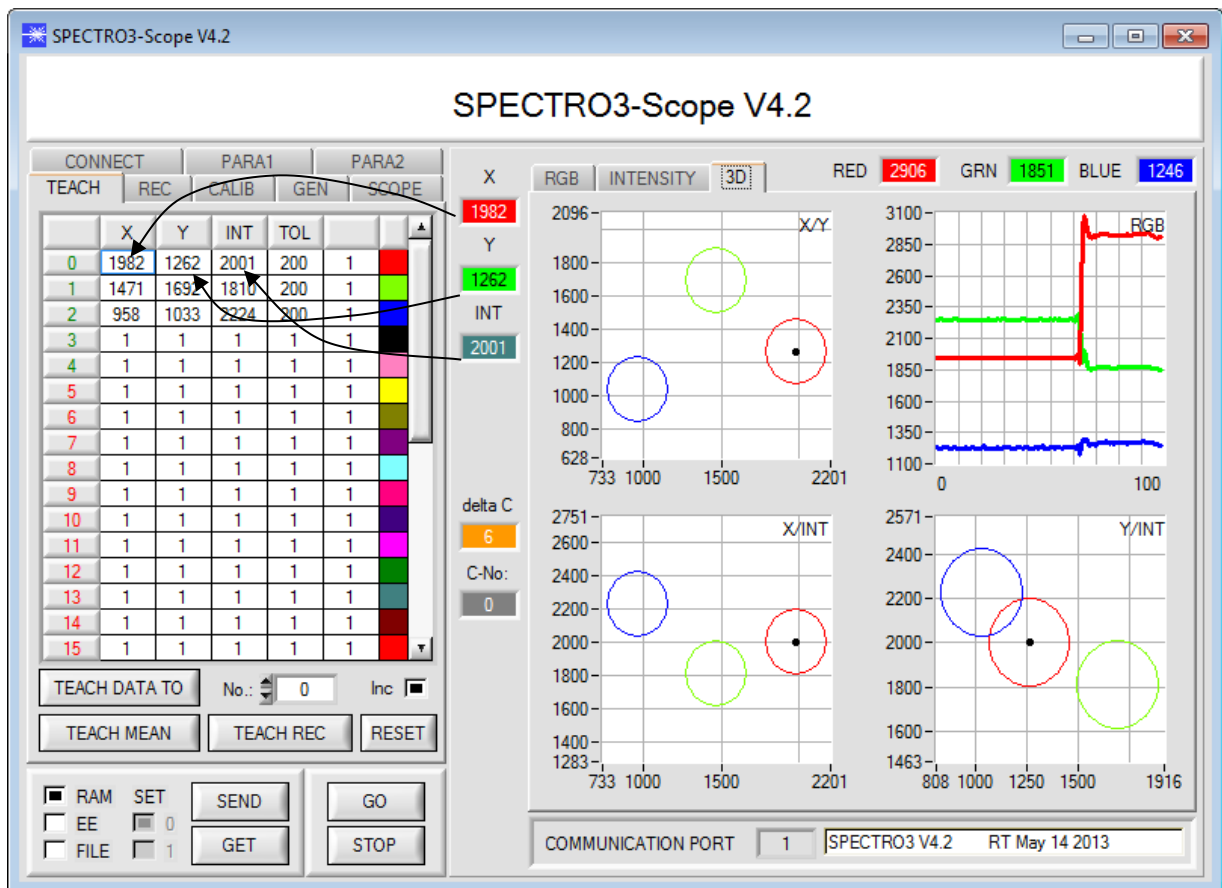
Now use the **TEACH TABLE** tab to switch to the **TEACH TABLE**.

Select a row under **No.:** to which the currently present color should be taught.

(**ATTENTION:** Only the rows marked with green numbers are evaluated by the sensor. If you want to teach more colors, use the **MAXCOL-No.** parameter to select how many colors you want to teach.)

Now press the **TEACH DATA TO** button. The calculated values for **X**, **Y**, and **INT** are transferred to the **TEACH TABLE** in the row that was selected under **No.:**

The graphic displays show a 3D view of a color sphere in space. The radius of this color sphere can be set under **TOL** (tolerance). To change the **TOL** value please double-click on the corresponding cell in the **TEACH TABLE** with the left mouse button.



#### Step 5:

For teaching additional colors please make sure that the sensor head is directed onto the respective color. Then repeat the procedure starting from step 4.

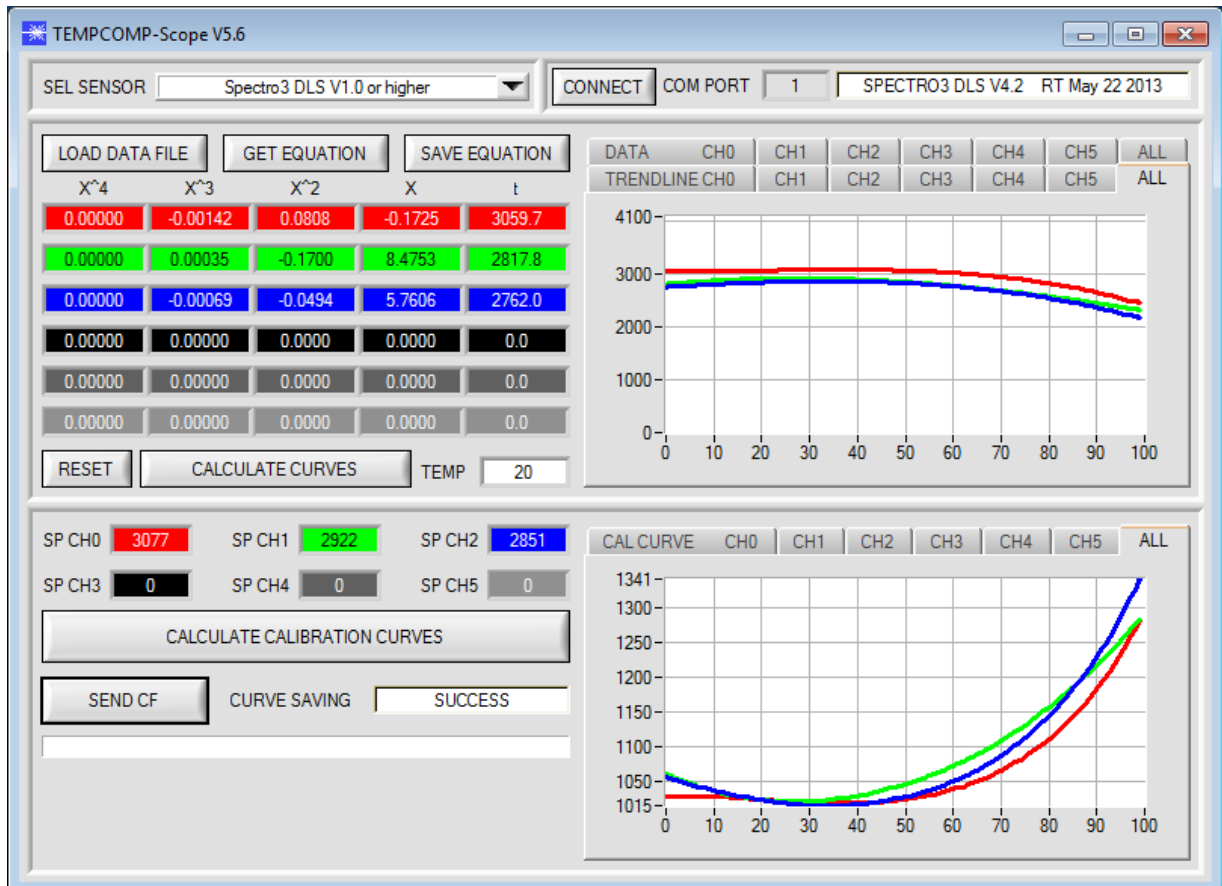
#### Step 6:

When you have taught all the channels, please select **EEPROM** and press **SEND** to save the data to the non-volatile memory of the sensor.

## 4. 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.

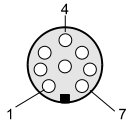
To perform temperature compensation please start the corresponding **TEMPComp-Scope software** that is included on the supplied CD. Please make sure that you have a functioning sensor connection. It may be necessary to select the connection with **CONNECT**. Set the correct sensor under **SELECT SENSOR**, 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.

## 5. External triggering of the SPECTRO-3 color sensor



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



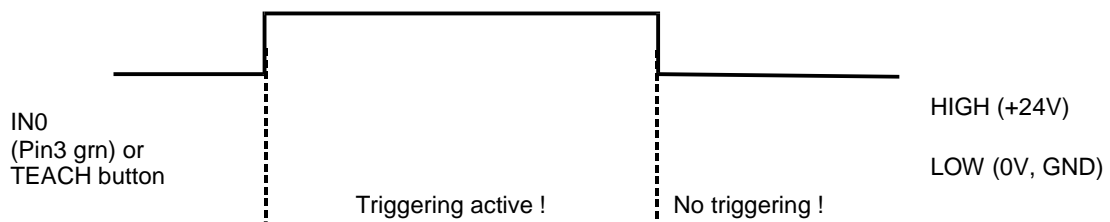
### TRIGGER:

First the external trigger mode must be set at the color sensor. For this purpose option **EXT1**, **EXT2**, **EXT3**, **TRANS**, or **PARA** must be selected in the **TRIGGER** selection field.

### PLEASE NOTE:

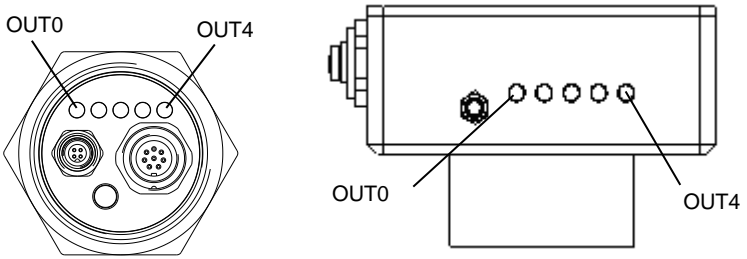
The new setting is only activated at the sensor after a click on the **SEND** button!

**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).



## 6. Function of the LED display [not available with SPECTRO-3-...-JR types]

LED display:



**BINARY**

The color code is visualised by way of 5 yellow LEDs at the housing of the SPECTRO-3 color sensor. At the same time in the binary mode (OUT BINARY) the color code indicated on the LED display is output as 5-bit binary information at the digital outputs OUT0 to OUT4 of the 8-pin SPECTRO-3/PLC socket.

The SPECTRO-3 color sensor is able to process a maximum of 31 colors (color code 0 ... 30) in accordance with the corresponding rows in the TEACH TABLE. An "error" respectively a "not detected color" is displayed by the lighting of all LED (OUT0 .. OUT4 digital outputs are set to HIGH-level).

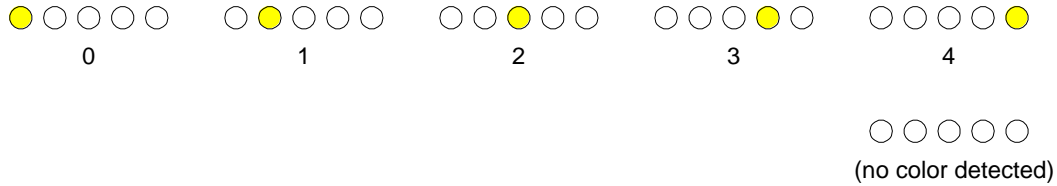
|           |           |           |                       |
|-----------|-----------|-----------|-----------------------|
| ○ ○ ○ ○ ○ | ● ○ ○ ○ ○ | ○ ● ○ ○ ○ | ● ● ○ ○ ○             |
| 0         | 1         | 2         | 3                     |
| ○ ○ ● ○ ○ | ● ○ ● ○ ○ | ○ ● ● ○ ○ | ● ● ● ○ ○             |
| 4         | 5         | 6         | 7                     |
| ○ ○ ○ ● ○ | ● ○ ○ ● ○ | ○ ● ○ ● ○ | ● ● ○ ● ○             |
| 8         | 9         | 10        | 11                    |
| ○ ○ ● ● ○ | ● ○ ● ● ○ | ○ ● ● ● ○ | ● ● ● ● ○             |
| 12        | 13        | 14        | 15                    |
| ○ ○ ○ ○ ● | ● ○ ○ ○ ● | ○ ● ○ ○ ● | ● ● ○ ○ ●             |
| 16        | 17        | 18        | 19                    |
| ○ ○ ● ○ ● | ● ○ ● ○ ● | ○ ● ● ○ ● | ● ● ● ○ ●             |
| 20        | 21        | 22        | 23                    |
| ○ ○ ○ ● ● | ● ○ ○ ● ● | ○ ● ○ ● ● | ● ● ○ ● ●             |
| 24        | 25        | 26        | 27                    |
| ○ ○ ● ● ● | ● ○ ● ● ● | ○ ● ● ● ● | ● ● ● ● ●             |
| 28        | 29        | 30        | Error or not detected |



### DIRECT HI:

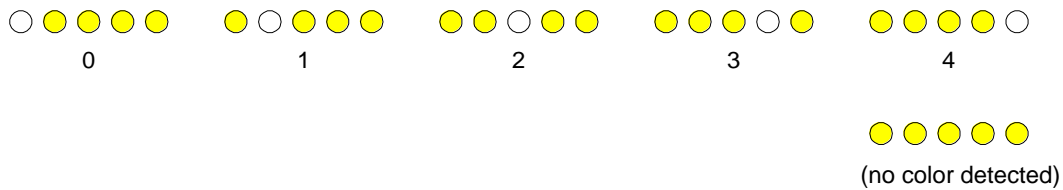
In the DIRECT mode (OUT DIRECT HI or OUT DIRECT LO) the maximum numbers of colors to be taught is 5 (color no. 0, 1, 2, 3, 4).

If DIRECT HI is activated, the specially digital output is set to HI, while the other 4 are set to LO. If the current color does not correspond with any of the teach-in colors, color C-No = 0 is set, i.e. all digital outputs are set to LOW (no LED is lighting).




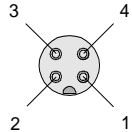
### DIRECT LO:

If DIRECT LO is activated, the specially digital output is set to LO, while the other 4 are set to HI. If the current color does not correspond with any of the teach-in colors, color C-No = 255 is set, i.e. all digital outputs are set to HIGH (all LED are lighting).


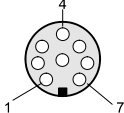


## 7. Connector assignment of the SPECTRO-3 color sensor

### Connection of SPECTRO-3 to PC:

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

### Connection of SPECTRO-3 to PLC:

| <b>8-pole female connector (type Binder 712)</b><br><b>SPECTRO-3/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 (Digital 0: typ. 0 ... 1V, Digital 1: typ. +Ub – 10%) |   |
| 7  | Blue   | OUT3 (Digital 0: typ. 0 ... 1V, Digital 1: typ. +Ub – 10%) |   |
| 8  | Red    | OUT4 (Digital 0: typ. 0 ... 1V, Digital 1: typ. +Ub – 10%) |   |

## 8. RS232 communication protocol

The sensors of the SPECTRO-3 series operate with the following **parameters** that are sent to the sensor or read from the sensor in the stated sequence.  
 Info! 2 bytes (8bit) are one word (16bit).

| Parameter                        | Type | Meaning   |
|----------------------------------|------|---|
| <b>Para1:</b> POWER              | Word | Transmitter intensity (0 ... 1000) Attention intensity in thousandth!   |
| <b>Para2:</b> POWER MODE         | Word | Transmitter mode: STATIC, DYNAMIC coded to (0, 1)   |
| <b>Para3:</b> AVERAGE            | Word | Signal averaging 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384 or 32768                            |
| <b>Para4:</b> EVALUATION MODE    | Word | Evaluation mode: FIRST HIT, BEST HIT, MIN DIST, COL5, THD RGB coded to (0,1,2,3,4)  |
| <b>Para5:</b> HOLD for C-No: 255 | Word | Hold time for failure condition (C-No: 255) coded to (0...100) [ms]   |
| <b>Para6:</b> INTLIM             | Word | Intensity limit (0 ... 4095)  |
| <b>Para7:</b> MAXCOL-No.         | Word | Number of the colours (1,2,3,...,31)  |
| <b>Para8:</b> OUTMODE            | Word | Function of the digital output: direct/HI, binary, direct/LO coded to (0,1,2)   |
| <b>Para9:</b> TRIGGER            | Word | Trigger mode: CONT, SELF, EXT1, EXT2, EXT3, TRANS, PARA coded to (0, 1, 2, 3, 4, 5, 6)                                    |
| <b>Para10:</b> EXTEACH           | Word | External teach mode: OFF, ON, STAT1, DYN1 coded to (0,1,2,3)  |
| <b>Para11:</b> CALCULATION MODE  | Word | Calculation mode: „X Y INT - 2D“, „s i M - 2D“, „X Y INT - 3D“, „s i M - 3D“ coded to (0,1,2,3)                           |
| <b>Para12:</b> DYN WIN LO        | Word | Low limit for dynamic window when POWER MODE=dynamic (0...4095)   |
| <b>Para13:</b> DYN WIN HI        | Word | High limit for dynamic window when POWER MODE=dynamic (0...4095)  |
| <b>Para14:</b> COLOR GROUPS      | Word | Color groups enable: OFF, ON coded to (0,1)   |
| <b>Para15:</b> LED MODE          | Word | Control for the internal light source DC, AC, PULSE, OFF coded to (0,1,2,3)   |
| <b>Para16:</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>Para17:</b> INTEGRAL          | Word | Signal integration (1...250)  |

One row in the Teach Table, Group Table and Hold Table determines a **TEACH VECTOR**.

The sensors of the SPECTRO-3 series operate with 31 **TEACH VECTORS** that are sent to the sensor or read from the sensor in one block in the stated sequence.

| TEACH VECTOR                                  | Type | Meaning                                 |
|---|------|---|
| <b>TeachVal1:</b> Teach Table Row 0 Column 0  | Word | X respectively s of row 0               |
| <b>TeachVal2:</b> Teach Table Row 0 Column 1  | Word | Y respectively i of row 0               |
| <b>TeachVal3:</b> Teach Table Row 0 Column 2  | Word | CTO or INT respectively M of row 0      |
| <b>TeachVal4:</b> Teach Table Row 0 Column 3  | Word | INT respectively M or TOL of row 0      |
| <b>TeachVal5:</b> Teach Table Row 0 Column 4  | Word | ITO or free of row 0                    |
| <b>TeachVal6:</b> Group Table Row 0           | Word | Group to which row 0 belongs if enabled |
| <b>TeachVal7:</b> Hold Table Row 0            | Word | Hold time for row 0                     |
| <b>TeachVal8:</b> Free                        | Word | Send 0 as dummy                         |
| <b>TeachVal9:</b> Teach Table Row 1 Column 0  | Word | X respectively s of row 1               |
| <b>TeachVal10:</b> Teach Table Row 1 Column 1 | Word | Y respectively i of row 1               |
| <b>TeachVal11:</b> Teach Table Row 1 Column 2 | Word | CTO or INT respectively M of row 1      |
| ...   | ...  | ...                                     |
| <b>TeachVal248:</b> Free                      | Word | Send 0 as dummy                         |

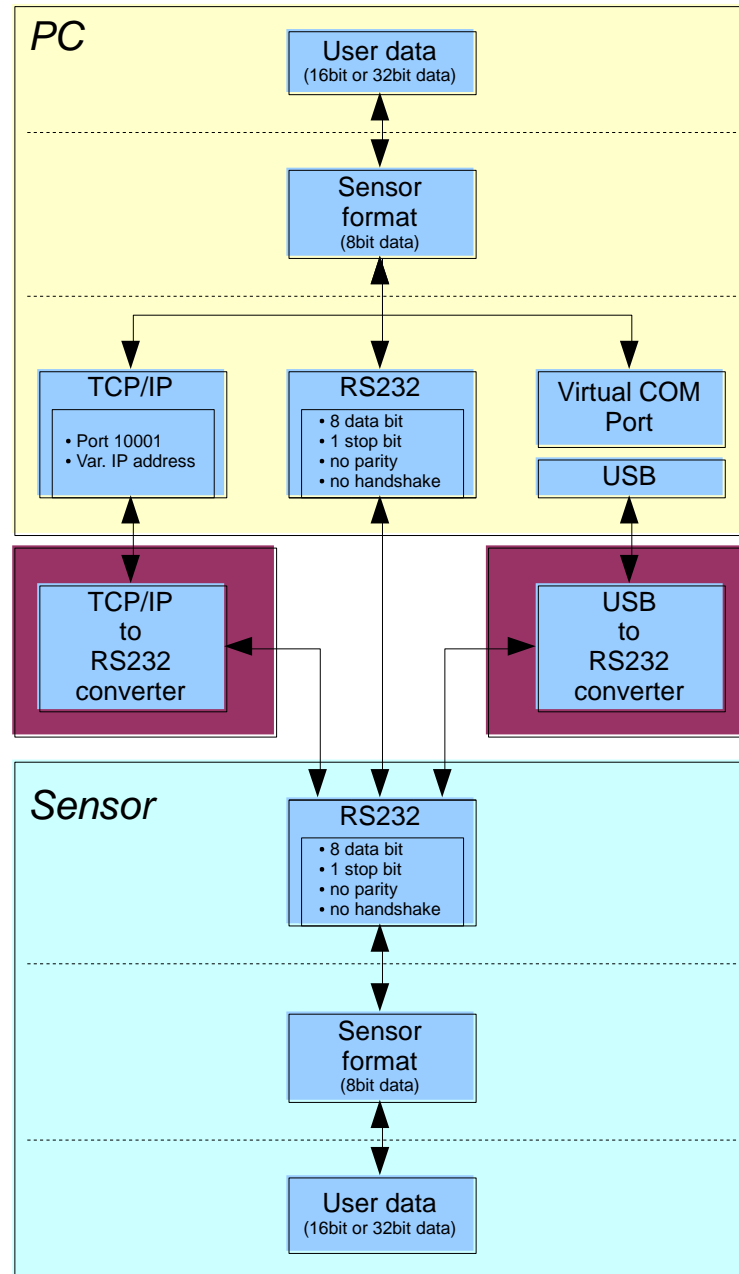
Upon request, the data acquired and processed by the sensor are sent by the sensor in the following sequence.

| DATA VALUE                  | Type | Meaning  |
|-----------------------------|------|--|
| <b>DatVal1:</b> RED         | Word | Calibrated and temperature compensated signal from channel red             |
| <b>DatVal2:</b> GREEN       | Word | Calibrated and temperature compensated signal from channel green           |
| <b>DatVal3:</b> BLUE        | Word | Calibrated and temperature compensated signal from channel blue            |
| <b>DatVal4:</b> X resp. s   | Word | Calculated X respectively s value  |
| <b>DatVal5:</b> Y resp. i   | Word | Calculated Y respectively i value  |
| <b>DatVal6:</b> INT resp. M | Word | Calculated INT respectively M value  |
| <b>DatVal7:</b> delta C     | Word | Distance to a color hit  |
| <b>DatVal8:</b> C-No:       | Word | Detected color   |
| <b>DatVal9:</b> GRP         | Word | Detected group   |
| <b>DatVal10:</b> TRIG       | Word | TRIG is 1 when a trigger condition is given                                |
| <b>DatVal11:</b> TEMP       | Word | Temperature in the sensor (not in °C or °F)                                |
| <b>DatVal12:</b> RAW RED    | Word | None Calibrated and none temperature compensated signal from channel red   |
| <b>DatVal13:</b> RAW GREEN  | Word | None Calibrated and none temperature compensated signal from channel green |
| <b>DatVal14:</b> RAW BLUE   | Word | None Calibrated and none temperature compensated signal from channel blue  |
| <b>DatVal15:</b> MIN RED    | Word | Calibrated and temperature compensated minimum signal from channel red     |
| <b>DatVal16:</b> MAX RED    | Word | Calibrated and temperature compensated maximum signal from channel red     |
| <b>DatVal17:</b> MIN GREEN  | Word | Calibrated and temperature compensated minimum signal from channel green   |
| <b>DatVal18:</b> MAX GREEN  | Word | Calibrated and temperature compensated maximum signal from channel green   |
| <b>DatVal19:</b> MIN BLUE   | Word | Calibrated and temperature compensated minimum signal from channel blue    |
| <b>DatVal20:</b> MAX BLUE   | Word | Calibrated and temperature compensated maximum signal from channel blue    |

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 from sensor   | order=5   |
| 6      | Free   |           |
| 7      | Read Firmware String from sensor   | order=7   |
| 8      | Read data values from sensor   | order=8   |
| 30     | Start and Stop triggered sending of data frames  | order=30  |
| 103    | Start white light correction and get calibration factors, setvalue and max delta of raw data.  | order=103 |
| 105    | Get cycle time from sensor   | order=105 |
| 190    | Write new baud rate to the sensor  | order=190 |

## CRC8 checksum

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

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

To verify the data after they have been received, CRC calculation is performed once again. If the sent and the newly calculated CRC values are identical, the data are without error.

The following pseudo code can be used for checksum calculation:

**calcCRC8** (data[ ], table[ ])

**Input:** data[ ], n data of unsigned 8bit

table[ ], 256 table entries of unsigned 8bit

**Output:** crc8, unsigned 8bit

```

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

```

**table[ ]**

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



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

<ARG> determines whether you want to save parameter or teach vectors:

ARG = 0 --> Parameter Set 0

ARG = 1 --> Parameter Set 1

ARG = 2 --> Teach Vector Set 0

ARG = 3 --> Teach Vector Set 1

DATA FRAME PC → Sensor for Parameter Set 0

| Byte0 Header | Byte1 Header | Byte2 Header    | Byte3 Header    | Byte4 Header    | Byte5 Header    | Byte6 Header | Byte7 Header  | Byte8 Data      | Byte9 Data      | Byte10 Data     | Byte11 Data     |
|--------------|--------------|-----------------|-----------------|-----------------|-----------------|--------------|---------------|-----------------|-----------------|-----------------|-----------------|
| 0x55         | <order>      | <ARG> (lo byte) | <ARG> (hi byte) | <LEN> (lo byte) | <LEN> (hi byte) | CRC8 (Data)  | CRC8 (Header) | Para1 (lo byte) | Para1 (hi byte) | Para2 (lo byte) | Para2 (hi byte) |
| 85 (dec)     | 1            | 0               | 0               | 34              | 0               | 162          | 249           | 244             | 1               | 0               | 0               |
| ARG=0        |              |                 |                 | LEN=34          |                 |              |               | Para1=500       |                 | Para2=0         |                 |

| Byte12 Data     | Byte13 Data     | Byte14 Data     | Byte15 Data     | Byte16 Data     | Byte17 Data     | Byte18 Data     | Byte19 Data     | Byte20 Data     | Byte21 Data     | Byte22 Data     | Byte23 Data     |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Para3 (lo byte) | Para3 (hi byte) | Para4 (lo byte) | Para4 (hi byte) | Para5 (lo byte) | Para5 (hi byte) | Para6 (lo byte) | Para6 (hi byte) | Para7 (lo byte) | Para7 (hi byte) | Para8 (lo byte) | Para8 (hi byte) |
| 1               | 0               | 1               | 0               | 10              | 0               | 0               | 0               | 5               | 0               | 0               | 0               |
| Para3=1         |                 | Para4=1         |                 | Para5=10        |                 | Para6=0         |                 | Para7=5         |                 | Para8=0         |                 |

| Byte24 Data     | Byte25 Data     | Byte26 Data      | Byte27 Data      | Byte28 Data      | Byte29 Data      | Byte30 Data      | Byte31 Data      | Byte32 Data      | Byte33 Data      | Byte34 Data      | Byte35 Data      |
|-----------------|-----------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|
| Para9 (lo byte) | Para9 (hi byte) | Para10 (lo byte) | Para10 (hi byte) | Para11 (lo byte) | Para11 (hi byte) | Para12 (lo byte) | Para12 (hi byte) | Para13 (lo byte) | Para13 (hi byte) | Para14 (lo byte) | Para14 (hi byte) |
| 0               | 0               | 0                | 0                | 2                | 0                | 128              | 12               | 228              | 12               | 0                | 0                |
| Para9=0         |                 | Para10=0         |                  | Para11=2         |                  | Para12=3200      |                  | Para13=3300      |                  | Para14=0         |                  |

| Byte36 Data      | Byte37 Data      | Byte38 Data      | Byte39 Data      | Byte40 Data      | Byte41 Data      |
|------------------|------------------|------------------|------------------|------------------|------------------|
| Para15 (lo byte) | Para15 (hi byte) | Para16 (lo byte) | Para16 (hi byte) | Para17 (lo byte) | Para17 (hi byte) |
| 1                | 0                | 8                | 0                | 1                | 0                |
| Para15=1         |                  | Para16=8         |                  | Para17=1         |                  |

DATA FRAME PC → Sensor for Teach Vector Set 0

| Byte0 Header | Byte1 Header | Byte2 Header    | Byte3 Header    | Byte4 Header    | Byte5 Header    | Byte6 Header | Byte7 Header  | Byte8 Data           | Byte9 Data           | Byte10 Data          | Byte11 Data          |
|--------------|--------------|-----------------|-----------------|-----------------|-----------------|--------------|---------------|----------------------|----------------------|----------------------|----------------------|
| 0x55         | <order>      | <ARG> (lo byte) | <ARG> (hi byte) | <LEN> (lo byte) | <LEN> (hi byte) | CRC8 (Data)  | CRC8 (Header) | Teach Val1 (lo byte) | Teach Val1 (hi byte) | Teach Val2 (lo byte) | Teach Val2 (hi byte) |
| 85 (dec)     | 1            | 2               | 0               | 240             | 1               | 28           | 197           | 1                    | 0                    | 1                    | 0                    |
| ARG=0        |              |                 |                 | LEN=496         |                 |              |               | TeachVal1=1          |                      | TeachVal2=1          |                      |

| Byte12 Data          | Byte13 Data          | Byte14 Data          | Byte15 Data          | Byte16 Data          | Byte17 Data          | Byte18 Data          | Byte19 Data          | Byte20 Data          | Byte21 Data          | Byte22 Data          | Byte23 Data          |
|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| Teach Val3 (lo byte) | Teach Val3 (hi byte) | Teach Val4 (lo byte) | Teach Val4 (hi byte) | Teach Val5 (lo byte) | Teach Val5 (hi byte) | Teach Val6 (lo byte) | Teach Val6 (hi byte) | Teach Val7 (lo byte) | Teach Val7 (hi byte) | Teach Val8 (lo byte) | Teach Val8 (hi byte) |
| 1                    | 0                    | 1                    | 0                    | 1                    | 0                    | 0                    | 0                    | 10                   | 0                    | 0                    | 0                    |
| TeachVal3=1          |                      | TeachVal4=1          |                      | TeachVal5=1          |                      | TeachVal6=0          |                      | TeachVal7=10         |                      | TeachVal8=0          |                      |

| Byte24 Data          | Byte25 Data          | Byte26 Data           | Byte27 Data           | Byte28 Data           | Byte29 Data           | Byte30 Data           | Byte31 Data           | Byte32 Data           | Byte33 Data           | ... | ... |
|----------------------|----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----|-----|
| Teach Val9 (lo byte) | Teach Val9 (hi byte) | Teach Val10 (lo byte) | Teach Val10 (hi byte) | Teach Val11 (lo byte) | Teach Val11 (hi byte) | Teach Val12 (lo byte) | Teach Val12 (hi byte) | Teach Val13 (lo byte) | Teach Val13 (hi byte) | ... | ... |
| 1                    | 0                    | 1                     | 0                     | 1                     | 0                     | 1                     | 0                     | 1                     | 0                     | ... | ... |
| TeachVal9=1          |                      | TeachVal10=1          |                       | TeachVal11=1          |                       | TeachVal12=1          |                       | TeachVal13=1          |                       | ... |     |

DATA FRAME Sensor → PC

| Byte0 Header | Byte1 Header | Byte2 Header    | Byte3 Header    | Byte4 Header    | Byte5 Header    | Byte6 Header | Byte7 Header  |
|--------------|--------------|-----------------|-----------------|-----------------|-----------------|--------------|---------------|
| 0x55         | <order>      | <ARG> (lo byte) | <ARG> (hi byte) | <LEN> (lo byte) | <LEN> (hi byte) | CRC8 (Data)  | CRC8 (Header) |
| 85 (dec)     | 1            | 0               | 0               | 0               | 0               | 170          | 224           |
| ARG=0        |              |                 |                 | LEN=0           |                 |              |               |

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

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

<ARG> determines whether you want to read parameter or teach vectors:

ARG = 0 --> Parameter Set 0

ARG = 1 --> Parameter Set 1

ARG = 2 --> Teach Vector Set 0

ARG = 3 --> Teach Vector Set 1

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)        | 2               | 0                  | 0                  | 0                  | 0                  | 170             | 185              |
| 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  | Byte8<br>Data      | Byte9<br>Data      | Byte10<br>Data     | Byte11<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) | Para1<br>(lo byte) | Para1<br>(hi byte) | Para2<br>(lo byte) | Para2<br>(hi byte) |
| 85 (dec)        | 2               | 0                  | 0                  | 34                 | 0                  | 162             | 160              | 244                | 1                  | 0                  | 0                  |
| ARG=0           |                 |                    |                    | LEN=34             |                    |                 |                  | Para1=500          |                    | Para2=0            |                    |

| Byte12<br>Data     | Byte13<br>Data     | Byte14<br>Data     | Byte15<br>Data     | Byte16<br>Data     | Byte17<br>Data     | Byte18<br>Data     | Byte19<br>Data     | Byte20<br>Data     | Byte21<br>Data     | Byte22<br>Data     | Byte23<br>Data     |
|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| Para3<br>(lo byte) | Para3<br>(hi byte) | Para4<br>(lo byte) | Para4<br>(hi byte) | Para5<br>(lo byte) | Para5<br>(hi byte) | Para6<br>(lo byte) | Para6<br>(hi byte) | Para7<br>(lo byte) | Para7<br>(hi byte) | Para8<br>(lo byte) | Para8<br>(hi byte) |
| 1                  | 0                  | 1                  | 0                  | 10                 | 0                  | 0                  | 0                  | 5                  | 0                  | 0                  | 0                  |
| Para3=1            |                    | Para4=1            |                    | Para5=10           |                    | Para6=0            |                    | Para7=5            |                    | Para8=0            |                    |

| Byte24<br>Data     | Byte25<br>Data     | Byte26<br>Data      | Byte27<br>Data      | Byte28<br>Data      | Byte29<br>Data      | Byte30<br>Data      | Byte31<br>Data      | Byte32<br>Data      | Byte33<br>Data      | Byte34<br>Data      | Byte35<br>Data      |
|--------------------|--------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| Para9<br>(lo byte) | Para9<br>(hi byte) | Para10<br>(lo byte) | Para10<br>(hi byte) | Para11<br>(lo byte) | Para11<br>(hi byte) | Para12<br>(lo byte) | Para12<br>(hi byte) | Para13<br>(lo byte) | Para13<br>(hi byte) | Para14<br>(lo byte) | Para14<br>(hi byte) |
| 0                  | 0                  | 0                   | 0                   | 2                   | 0                   | 128                 | 12                  | 228                 | 12                  | 0                   | 0                   |
| Para9=0            |                    | Para10=0            |                     | Para11=2            |                     | Para12=3200         |                     | Para13=3300         |                     | Para14=0            |                     |

| Byte36<br>Data      | Byte37<br>Data      | Byte38<br>Data      | Byte39<br>Data      | Byte40<br>Data      | Byte41<br>Data      |
|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| Para15<br>(lo byte) | Para15<br>(hi byte) | Para16<br>(lo byte) | Para16<br>(hi byte) | Para17<br>(lo byte) | Para17<br>(hi byte) |
| 1                   | 0                   | 8                   | 0                   | 1                   | 0                   |
| Para15=1            |                     | Para16=8            |                     | Para17=1            |                     |

**Example order=3:** Load parameter and actual Baudrate from RAM to EEPROM of the 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)        | 3               | 0                  | 0                  | 0                  | 0                  | 170             | 142              |
| 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)        | 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<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)        | 4               | 0                  | 0                  | 0                  | 0                  | 170             | 11               |
| 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)        | 4               | 0                  | 0                  | 0                  | 0                  | 170             | 11               |
| ARG=0           |                 |                    |                    | LEN=0              |                    |                 |                  |

**Example order=5:** Read CONNECTION OK 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)        | 5               | 0                  | 0                  | 0                  | 0                  | 170             | 60               |
| 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)        | 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

| 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 |
| ASCII       | ASCII       | ASCII       | ASCII       | ASCII       | ASCII       | ASCII       | ASCII       |
|             |             |             |             |             |             |             |             |

**Example order=8:** Read 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)        | 8               | 0                  | 0                  | 0                  | 0                  | 170             | 118              |
| 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  | Byte8<br>Data        | Byte9<br>Data        | Byte10<br>Data       | Byte11<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) | DatVal1<br>(lo byte) | DatVal1<br>(hi byte) | DatVal2<br>(lo byte) | DatVal2<br>(hi byte) |
| 85 (dec)        | 8               | 0                  | 0                  | 40                 | 0                  | 175             | 248              | 246                  | 11                   | 180                  | 7                    |
| ARG=0           |                 |                    |                    | LEN=40             |                    |                 |                  | DatVal=3062          |                      | DatVal=1972          |                      |

| Byte12<br>Data       | Byte13<br>Data       | Byte14<br>Data       | Byte15<br>Data       | Byte16<br>Data       | Byte17<br>Data       | Byte18<br>Data       | Byte19<br>Data       | Byte20<br>Data       | Byte21<br>Data       | Byte22<br>Data       | Byte23<br>Data       |
|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| DatVal3<br>(lo byte) | DatVal3<br>(hi byte) | DatVal4<br>(lo byte) | DatVal4<br>(hi byte) | DatVal5<br>(lo byte) | DatVal5<br>(hi byte) | DatVal6<br>(lo byte) | DatVal6<br>(hi byte) | DatVal7<br>(lo byte) | DatVal7<br>(hi byte) | DatVal8<br>(lo byte) | DatVal8<br>(hi byte) |
| 185                  | 5                    | 137                  | 7                    | 218                  | 4                    | 118                  | 8                    | 255                  | 255                  | 255                  | 0                    |
| DatVal3=1465         |                      | DatVal4=1929         |                      | DatVal5=1242         |                      | DatVal6=2166         |                      | DatVal7=65535        |                      | DatVal8=255          |                      |

| Byte24<br>Data       | Byte25<br>Data       | Byte26<br>Data        | Byte27<br>Data        | Byte28<br>Data        | Byte29<br>Data        | Byte30<br>Data        | Byte31<br>Data        | Byte32<br>Data        | Byte33<br>Data        | Byte34<br>Data        | Byte35<br>Data        |
|----------------------|----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| DatVal9<br>(lo byte) | DatVal9<br>(hi byte) | DatVal10<br>(lo byte) | DatVal10<br>(hi byte) | DatVal11<br>(lo byte) | DatVal11<br>(hi byte) | DatVal12<br>(lo byte) | DatVal12<br>(hi byte) | DatVal13<br>(lo byte) | DatVal13<br>(hi byte) | DatVal14<br>(lo byte) | DatVal14<br>(hi byte) |
| 255                  | 0                    | 0                     | 0                     | 23                    | 0                     | 246                   | 11                    | 180                   | 7                     | 185                   | 5                     |
| DatVal9=255          |                      | DatVal10=0            |                       | DatVal11=23           |                       | DatVal12=3062         |                       | DatVal13=1972         |                       | DatVal14=1465         |                       |

| Byte36<br>Data        | Byte37<br>Data        | Byte38<br>Data        | Byte39<br>Data        | Byte40<br>Data        | Byte41<br>Data        | Byte42<br>Data        | Byte43<br>Data        | Byte44<br>Data        | Byte45<br>Data        | Byte46<br>Data        | Byte47<br>Data        |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| DatVal15<br>(lo byte) | DatVal15<br>(hi byte) | DatVal16<br>(lo byte) | DatVal16<br>(hi byte) | DatVal17<br>(lo byte) | DatVal17<br>(hi byte) | DatVal18<br>(lo byte) | DatVal18<br>(hi byte) | DatVal19<br>(lo byte) | DatVal19<br>(hi byte) | DatVal20<br>(lo byte) | DatVal20<br>(hi byte) |
| 0                     | 0                     | 0                     | 0                     | 0                     | 0                     | 0                     | 0                     | 0                     | 0                     | 0                     | 0                     |
| DatVal15=0            |                       | DatVal16=0            |                       | DatVal17=0            |                       | DatVal18=0            |                       | DatVal19=0            |                       | DatVal20=0            |                       |

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

Start triggered sending of data frames

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)     | 30           | 1                  | 0                  | 0                  | 0                  | 170            | 82               |
| ARG=1        |              |                    |                    | LEN=0              |                    |                |                  |

DATA FRAME Sensor → PC

| Byte0 Header | Byte1 Header | Byte2 Header       | Byte3 Header       | Byte4 Header       | Byte5 Header       | Byte6 Header   | Byte7 Header     |
|--------------|--------------|--------------------|--------------------|--------------------|--------------------|----------------|------------------|
| 0x55         | <order>      | <ARG><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

| 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)     | 30           | 0                  | 0                  | 0                  | 0                  | 170            | 159              |
| ARG=0        |              |                    |                    | LEN=0              |                    |                |                  |

DATA FRAME Sensor → PC

| Byte0 Header | Byte1 Header | Byte2 Header       | Byte3 Header       | Byte4 Header       | Byte5 Header       | Byte6 Header   | Byte7 Header     |
|--------------|--------------|--------------------|--------------------|--------------------|--------------------|----------------|------------------|
| 0x55         | <order>      | <ARG><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=103:** Start white light correction and get calibration factors, setvalue and max delta of raw data.

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)     | 103          | 0                  | 0                  | 0                  | 0                  | 170            | 145              |
| 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) | CF<br>RED<br>(lo byte) | CF<br>RED<br>(hi byte) | CF<br>GREEN<br>(lo byte) | CF<br>GREEN<br>(hi byte) |
| 85 (dec)     | 103          | 0                  | 0                  | 10                 | 0                  | 212            | 28               | 228                    | 3                      | 223                      | 3                        |
| ARG=0        |              |                    |                    | LEN=10             |                    |                |                  | CF_RED = 996           |                        | CF_GREEN = 991           |                          |

| Byte12 Data             | Byte13 Data             | Byte14 Data               | Byte15 Data               | Byte16 Data               | Byte17 Data               |
|-------------------------|-------------------------|---------------------------|---------------------------|---------------------------|---------------------------|
| CF<br>BLUE<br>(lo byte) | CF<br>BLUE<br>(hi byte) | SET<br>VALUE<br>(lo byte) | SET<br>VALUE<br>(hi byte) | MAX<br>DELTA<br>(lo byte) | MAX<br>DELTA<br>(hi byte) |
| 65                      | 4                       | 134                       | 12                        | 43                        | 1                         |
| CF_BLUE = 1089          |                         | SETVALUE = 3206           |                           | MAX DELTA = 299           |                           |

**Example order=105:** Get cycle time 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)        | 105             | 0                  | 0                  | 0                  | 0                  | 170             | 130              |
| 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  | Byte8<br>Data        | Byte9<br>Data      | Byte10<br>Data     | Byte11<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) | 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<br>Data     | Byte13<br>Data     | Byte14<br>Data     | Byte15<br>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<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)        | 190             | 1                  | 0                  | 0                  | 0                  | 170             | 14               |
| ARG=1           |                 |                    |                    | LEN=0              |                    |                 |                  |

New baud rate is determined by argument.

ARG=0: baud rate = 9600

ARG=1: baud rate = 19200

ARG=2: baud rate = 38400

ARG=3: baud rate = 57600

ARG=4: baud rate = 115200

DATA FRAME Sensor → PC

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