

# Bedienungsanleitung Software SPECTRO3-SLA-Scope V1.0

(PC Software für Microsoft® Windows® XP, VISTA, 7)

## für Farbsensoren der SPECTRO-3-...-SLA Serie mit interner Temperaturkompensation und Weißlichtabgleich

Die vorliegende Bedienungsanleitung dient zur Installation der PC-Software für den SPECTRO-3-...-SLA Farbsensor. Zur Unterstützung der Inbetriebnahme des Farbsensors werden in dieser Bedienungsanleitung die einzelnen Funktionselemente der graphischen Windows® Benutzeroberfläche erklärt.

Die Signalerfassung mit dem SPECTRO-3-...-SLA ist sehr flexibel. Der Sensor kann z.B. im Wechsellicht Modus (AC Mode) betrieben werden. Hier ist der Sensor unabhängig gegen Fremdlicht. Auch ein Gleichlichtbetrieb (DC Mode) kann eingestellt werden. Hier ist der Sensor extrem schnell. Eine OFF Funktion schaltet die integrierte Lichtquelle am Sensor aus und wechselt in den DC-Betrieb, dann kann der Sensor sogenannte „Selbstleuchter“ erkennen. Die stufenlose Einstellmöglichkeit der integrierten Lichtquelle sowie eine selektierbare Verstärkung des Empfängersignals und eine INTEGRAL Funktion ermöglichen eine Einstellung des Sensors auf nahezu jede Oberfläche oder jeden „Selbstleuchter“.

Ist die integrierte Beleuchtung des SPECTRO-3-...-SLA Farbsensors aktiviert, detektiert der Sensor die am Messobjekt diffus zurückreflektierte Strahlung. Als Lichtquelle wird am SPECTRO-3-...-SLA Farbsensor eine Weißlicht-LED mit einstellbarer Sendeleistung eingesetzt. Als Empfänger wird ein integrierter 3-fach-Empfänger für den Rot-, Grün- und Blau-Anteil des vom Messobjekt zurückreflektierten Lichtes, oder des vom „Selbstleuchter“ emittierten Lichtes, verwendet.

Der Sensor hat 3 Analogausgänge. Wahlweise werden entweder die Rot-, Grün-, Blau-Anteile oder die errechneten Farbkoordinaten (X, Y, INT oder s, i, M) von 0...+10V oder 4...20mA ausgegeben.

Über die RS232-Schnittstelle können Parameter und Messwerte zwischen PC und dem SPECTRO-3-...-SLA Farbsensor ausgetauscht werden. Sämtliche Parameter zur Farberkennung können über die serielle Schnittstelle RS232 im nichtflüchtigen EEPROM des SPECTRO-3-...-SLA Farbsensors gespeichert werden. Nach erfolgter Parametrisierung arbeitet der Farbsensor im STAND-ALONE Betrieb mit den aktuellen Parametern ohne PC weiter.

Die Sensoren der SPECTRO-3-...-SLA Serie können kalibriert werden (Weißlichtabgleich). Der Abgleich kann dabei auf eine beliebige weiße Oberfläche erfolgen. Alternativ dazu ist eine ColorChecker™ Tabelle erhältlich. Diese verfügt über 24 Farbfelder nach der CIE-NORM. Der Weißlichtabgleich bzw. die Kalibrierung kann auf eines der weißen Felder erfolgen.

Bei den SPECTRO-3-UV-SLA (oder SPECTRO-3-FIO-UV-SLA) Farbsensoren wird als Lichtquelle eine UV-LED (375 nm) mit einstellbarer Sendeleistung zur Anregung der lumineszierenden Markierung eingesetzt. Diese UV-Sensoren können auf nahezu jeden, im langwelligen UV-Bereich (365 nm bzw. 375 nm) anregbaren, lumineszierenden Farbstoff optimal eingestellt werden. Die Sensoren der SPECTRO-3-...-UV-SLA Serie können ebenfalls kalibriert werden. Analog zum Weißlichtabgleich bei den Farbsensoren könnte der Abgleich des SPECTRO-3-UV-SLA (oder SPECTRO-3-FIO-UV-SLA) Farbsensors auf eine beliebige lumineszierende Farbmarkierung erfolgen.

## 0. Inhalt

	Seite
1. Installation der SPECTRO3-SLA-Scope Software .....	3
2. Bedienung der SPECTRO3-SLA-Scope Software .....	4
2.1 Registerkarte (Reiter oder Tab) CONNECT (Verbindungsaufbau).....	5
2.2 Registerkarte PARA1, Taste SEND, GET, GO, STOP (Parametrierung, Datenaustausch) .....	7
2.3 Registerkarte REC (Datenaufzeichnung) .....	11
2.4 Registerkarte CALIB.....	13
2.4.1 Weißlichtabgleich .....	13
2.4.2 Offsetkalibrierung .....	16
2.5 Registerkarte SCOPE .....	17
2.6 Graphische Anzeigeelemente .....	18
3. Bedienung der TEMPCOMP-Scope Software .....	19
4. Anschlussbelegung der SPECTRO-3-...-SLA Farbsensoren.....	20
5. RS232 Schnittstellenprotokoll (communication protocol).....	21

**Shortcuts:**

SEND	F9
GET	F10
GO	F11
STOP	F12

## 1. Installation der SPECTRO3-SLA-Scope Software

Für eine erfolgreiche Installation der SPECTRO3-SLA-Scope Software müssen folgende Voraussetzungen erfüllt sein:

- Microsoft® Windows® XP, VISTA, 7
- IBM PC AT oder kompatibler
- VGA-Grafik
- Microsoft®-kompatible Maus
- CD-ROM-Laufwerk
- Serielle RS232-Schnittstelle am PC oder USB Slot
- Kabel **cab-las4/PC** für die RS232-Schnittstelle oder **cab-las4/USB** für USB Slot

Die SPECTRO3-SLA-Scope Software kann nur unter Windows installiert werden. Deshalb müssen Sie zunächst Windows starten, falls es noch nicht aktiv ist.

Installieren Sie nun die Software wie im Folgenden beschrieben:

1. Sie können die Software direkt von der Installations-CD-ROM installieren. Auf der CD-ROM befindet sich der Ordner SOFTWARE. In diesem Ordner ist eine SETUP Anwendung. Zum Installieren der Software müssen Sie diese Setup-Anwendung starten.
2. Das Installationsprogramm meldet sich mit einem Dialogfeld und schlägt vor, die Software im Verzeichnis C:\DATEINAME" auf der Festplatte einzurichten. Akzeptieren Sie den Vorschlag mit **OK** oder **[ENTER]** oder ändern Sie die Pfad-Vorgaben nach Ihren Wünschen.
3. Während der Installation wird eine neue Programm-Gruppe für die Software im Windows Programm-Manager erzeugt. Außerdem wird in der erzeugten Programmgruppe ein Icon für den Start der Software automatisch generiert. Falls die Installation erfolgreich durchgeführt werden konnte, meldet sich das Installationsprogramm mit einer Dialogbox "Setup OK".
4. Nach erfolgreicher Installation kann die Software durch Doppelklick auf das Icon mit der linken Maustaste gestartet werden.

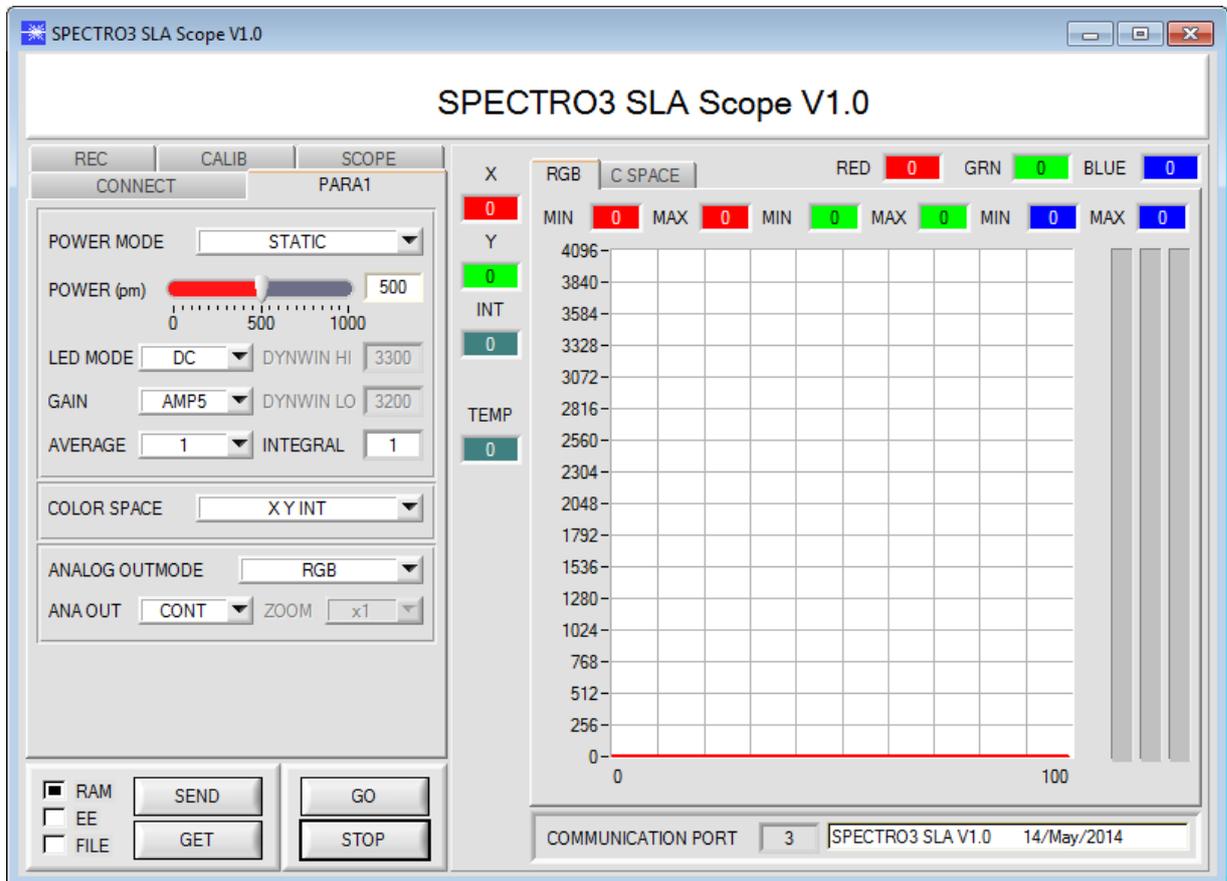
Windows® ist ein eingetragenes Warenzeichen der Microsoft Corp.

VGA™ ist ein Warenzeichen der International Business Machines Corp.

## 2. Bedienung der SPECTRO3-SLA-Scope Software

Bitte lesen Sie diesen Abschnitt zuerst durch, bevor Sie die Einjustierung und Parametrisierung des SPECTRO3-...-SLA Farbsensors vornehmen.

Nach dem Aufruf der SPECTRO3-SLA-Scope Software erscheint folgendes Fenster auf der Windows Oberfläche:



Das Fenster wird in seiner Größe und Position wieder dort platziert, wo es sich beim letzten Verlassen der Software befand. Durch einen Doppelklick mit der rechten Maustaste z.B. unterhalb des Minimierungssymbols wird das Fenster in seiner Originalgröße mittig zentriert.

Kommt es nicht automatisch zu einem Verbindungsaufbau z.B. wenn kein Sensor angeschlossen ist, dann kann die Software im OFFLINE Modus betrieben werden. Im Offline Modus ist lediglich ein Parameterraustausch mit einer Datei auf einem Speichermedium möglich. Dies ist für Analysezwecke von Parameterfiles oft hilfreich.

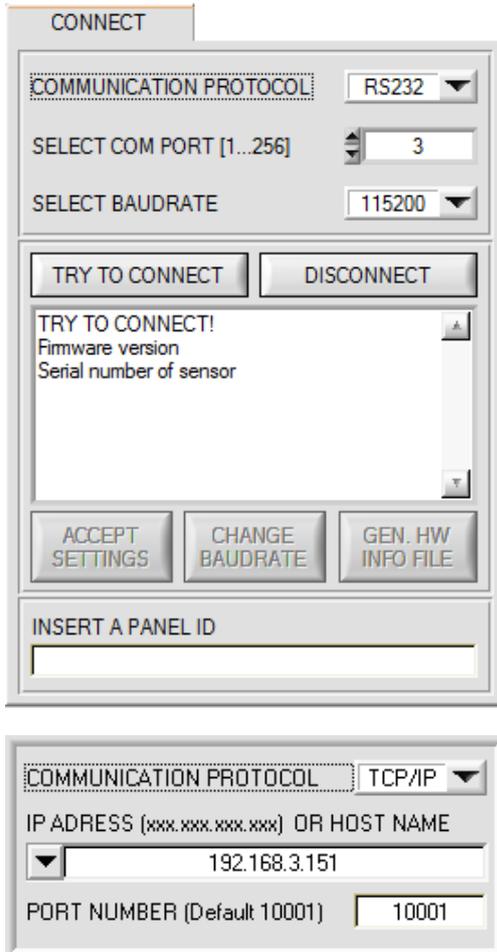
Ist ein Sensor angeschlossen und es kommt trotzdem zu keinem Verbindungsaufbau, dann stimmen entweder die SCOPE Version (Programm auf PC) und die Firmware Version (Programm im Sensor) nicht überein oder man muss die Schnittstelle zum Sensor richtig konfigurieren.

Sollte das Problem eine unterschiedliche Scope und Firmware Version sein, dann muss man sich die zur Firmware passende Scope Version vom Lieferanten besorgen.

Das Konfigurieren der Schnittstelle wird in der Registerkarte CONNECT erklärt.

**Eine Kurz-Hilfe wird durch Drücken der rechten Maustaste auf ein einzelnes Element angezeigt.**

## 2.1 Registerkarte CONNECT



The image shows two screenshots of the 'CONNECT' dialog box. The top screenshot shows the 'COMMUNICATION PROTOCOL' set to 'RS232', 'SELECT COM PORT' set to '3', and 'SELECT BAUDRATE' set to '115200'. It includes buttons for 'TRY TO CONNECT', 'DISCONNECT', 'ACCEPT SETTINGS', 'CHANGE BAUDRATE', and 'GEN. HW INFO FILE'. The bottom screenshot shows the 'COMMUNICATION PROTOCOL' set to 'TCP/IP', 'IP ADDRESS (xxx.xxx.xxx.xxx) OR HOST NAME' set to '192.168.3.151', and 'PORT NUMBER (Default 10001)' set to '10001'.

### CONNECT:

Durch Drücken von **CONNECT** öffnet sich eine Ansicht, in der man die Schnittstelle wählen und konfigurieren kann.

In dem Funktionsfeld **COMMUNICATION PROTOCOL** kann entweder ein **RS232** oder ein **TCP/IP** Protokoll ausgewählt werden.

Wählt man **RS232**, kann man mit **SELECT COM PORT** einen Port von 1 bis 256 auswählen, je nachdem an welchem der Sensor angeschlossen ist. Der Sensor arbeitet mit einer eingestellten Baudrate, die über **CHANGE BAUDRATE** verändert werden kann (siehe unten). Sowohl der Sensor als auch die Benutzeroberfläche müssen mit der gleichen Baudrate arbeiten.

Über **SELECT BAUDRATE** stellt man auf der Benutzeroberfläche die Baudrate ein. Sollte die Software nach dem Starten nicht automatisch eine Verbindung aufbauen kann mit **SELECT BAUDRATE** die richtige Baudrate gefunden werden.

Wenn man mit einem Adapter arbeitet, dann kann man die **COM PORT** Nummer über den Hardwaremanager in der Systemsteuerung ermitteln.

Zur Kommunikation des Sensors über ein lokales Netzwerk wird ein RS232 zu Ethernet Adapter benötigt. Dieser ermöglicht es eine Verbindung zum Sensor über das **TCP/IP** Protokoll herzustellen.

Die von uns erhältlichen Netzwerk Adapter basieren auf dem Lantronix XPort Modul. Um die Adapter zu parametrisieren (Vergabe von IP-Adresse, Einstellung der Baudrate,...), kann man die von Lantronix im Internet kostenlos bereitgestellte Software („DeviceInstaller“) unter <http://www.lantronix.com/> downloaden. DeviceInstaller basiert auf dem „.NET“ Framework von Microsoft. Eine ausführliche Anleitung zur Bedienung der Software „DeviceInstaller“ kann ebenso von Lantronix bezogen werden.

Um eine Verbindung zum Adapter herzustellen, muss dessen

IP-Adresse oder HOST Name in das Eingabefeld **IP ADRESS (xxx.xxx.xxx.xxx) OR HOST NAME** eingetragen werden. Im DROP DOWN Menü (Pfeil nach unten) sind die letzten 10 verwendeten IP Adressen aufgelistet und können durch Anklicken direkt übernommen werden. Die DROP DOWN Liste bleibt auch nach Beenden der Software erhalten.

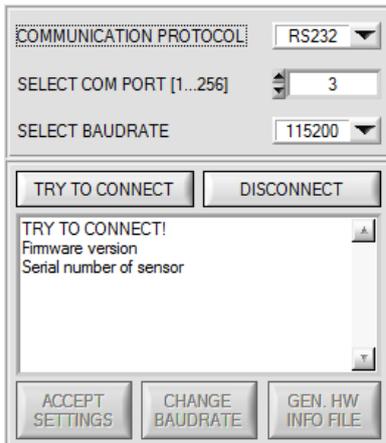
Die **PORT NUMBER** für die auf dem XPort basierenden Netzwerkadapter ist auf 10001 festgelegt und muss belassen werden.

Nach Drücken von **TRY TO CONNECT** versucht die Software eine Verbindung mit den eingestellten Parametern aufzubauen. Der Status der Kommunikation wird im Anzeigedisplays angezeigt. Meldet sich der Sensor mit seiner FIRMWARE ID, kann man mit **ACCEPT SETTINGS** die eingestellte Verbindungsart beibehalten. Die Software schaltet automatisch auf den Registerkarte **PARA** um. Erhält man ein **TIMEOUT**, konnte die Software keine Verbindung zum Sensor herstellen. In diesem Fall sollte zunächst geprüft werden, ob das Schnittstellenkabel richtig angebracht wurde, ob der Sensor an Spannung liegt und ob die eingestellten Parameter richtig gewählt wurden.

Wurde eine Verbindung mit **ACCEPT SETTINGS** bestätigt, dann startet die Software beim nächsten Aufruf automatisch mit dieser Einstellung.

Mit **DISCONNECT** trennt man die Verbindung vom Sensor zum PC. Die Software schaltet in den OFFLINE Modus in dem nur ein Parameteraustausch mit einer Datei auf einem Speichermedium möglich ist.

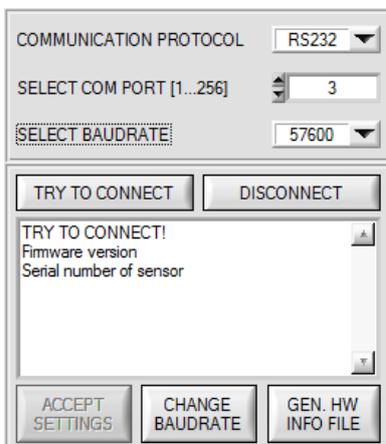
 <b>Achtung !</b>	<p><b>Beachte:</b> Grundvoraussetzung für die Messwertübertragung vom PC zum Sensor ist die stabile Funktion der Schnittstelle.</p> <p>Aufgrund der begrenzten Datenübertragungsrate über die serielle RS232-Schnittstelle können nur langsame Veränderungen der Rohsignale am Sensor-Frontend im graphischen Ausgabefenster des PC mitverfolgt werden.</p> <p>Zur Einhaltung der maximalen Schaltfrequenz am Sensor muss zudem der Datenaustausch mit dem PC beendet werden (STOP-Taste drücken).</p>
---	--



Die Baudrate zur Datenübertragung über die RS232 Schnittstelle kann mit **SELECT BAUDRATE** und **CHANGE BAUDRATE** eingestellt werden.

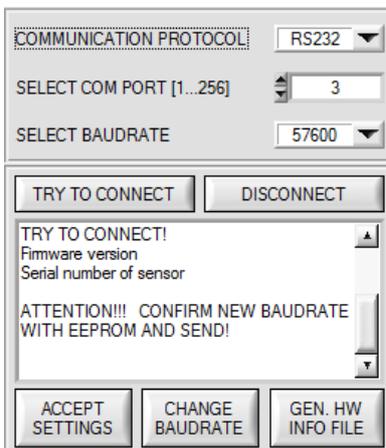
Zum Ändern muss zuerst über **TRY TO CONNECT** eine Verbindung aufgebaut werden.

Erst jetzt ist der Button **CHANGE BAUDRATE** aktiv.



Unter **SELECT BAUDRATE** kann jetzt eine neue Baudrate ausgewählt werden.

Durch Drücken von **CHANGE BAUDRATE** wird die neue Baudrate zum Sensor übertragen.



Nachdem die neue Baudrate erfolgreich übertragen worden ist arbeitet der Sensor mit der neuen Baudrate. Außerdem erscheint im Anzeigefenster eine Aufforderung **EEPROM** zu selektieren und anschließend **SEND** zu drücken. Erst nach Drücken von **EEPROM** und **SEND** wird bei einem Hardware-Reset mit der neuen Baudrate gestartet.

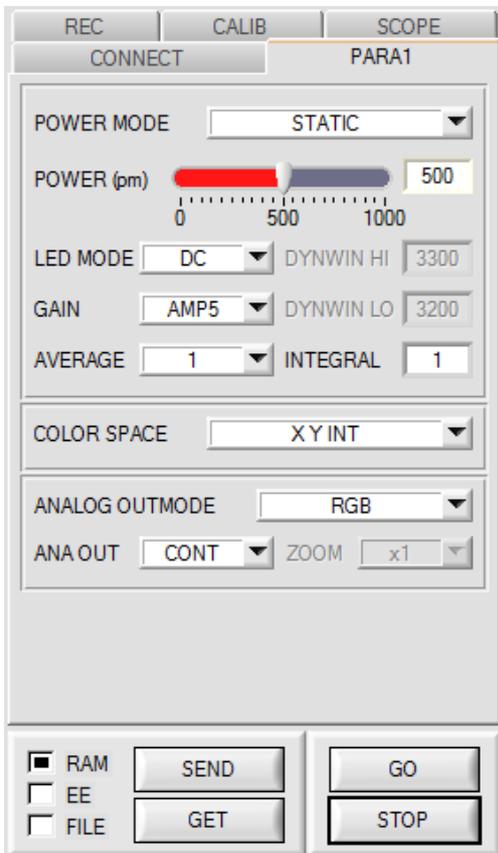
Durch Drücken von **ACCEPT SETTINGS** werden die aktuellen Schnittstellen-Einstellungen gespeichert und nach einem Neustart der Software automatisch eingestellt.



Über den Button **GEN. HW INFO FILE** wird ein File erzeugt, in dem alle wichtigen Sensordaten verschlüsselt hinterlegt werden.

Dieses File kann zu Diagnosezwecke an den Hersteller gesendet werden.

## 2.2 Registerkarte PARA1, Taste SEND, GET, GO, STOP



### PARA1:

Durch Drücken von **PARA1** öffnet sich eine Ansicht, in der man die Sensorparameter einstellen kann.

**Beachte: Eine Änderung der Funktionsgruppen Parameter wird erst nach Betätigung der SEND-Taste im MEM-Funktionsfeld am Sensor wirksam!**

### SEND [F9]:

Durch Anklicken der Taste **SEND** (bzw. per Shortcut Keytaste F9) werden alle aktuell eingestellten Parameter zwischen PC und dem Sensor übertragen. Das Ziel der jeweiligen Parameterübertragung wird durch den selektierten Auswahlknopf (**RAM**, **EEPROM** oder **FILE**) festgelegt.

### GET [F10]:

Durch Anklicken der Taste **GET** (bzw. per Shortcut Keytaste F10) können die aktuellen Einstellwerte vom Sensor abgefragt werden. Die Quelle des Datenaustausches wird über den selektierten Auswahlknopf (**RAM**, **EEPROM** oder **FILE**) festgelegt.

### RAM:

Die aktuellen Parameter werden nach Drücken von **SEND** in den **RAM** Speicher des Sensors geschrieben bzw. nach Drücken von **GET** aus dessen **RAM** Speicher gelesen, d.h. nach Ausschalten der Spannung am Sensor gehen diese Parameter wieder verloren.

### EEPROM:

Die aktuellen Parameter werden nach Drücken von **SEND** in den Speicher des nichtflüchtigen **EEPROMS** im Sensor geschrieben oder durch Drücken von **GET** aus dessen **EEPROM** gelesen, d.h. nach Ausschalten der Spannung am Sensor bleiben die im internen **EEPROM** abgelegten Parameter erhalten.

### FILE:

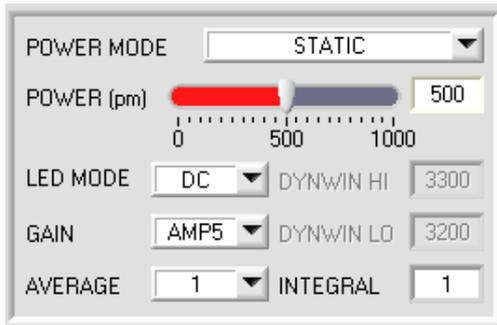
Die aktuellen Parameter können nach Drücken von **SEND** in ein auswählbares File auf der Festplatte geschrieben werden bzw. durch Drücken von **GET** davon gelesen werden. Nach Drücken von **SEND** oder **GET** öffnet sich eine Dialogbox, in der man das gewünschte File selektieren kann.

### GO [F11]:

Nach Anklicken dieser Taste wird der Datentransfer vom Sensor zum PC über die serielle RS232 Schnittstelle gestartet. Unter **SOURCE** wählt man aus, welche Signale in den Displays und Graphen zur Anzeige gebracht werden.

### STOP [F12]:

Nach Anklicken dieser Taste wird der Datentransfer vom Sensor zum PC über die serielle RS232 Schnittstelle beendet.



#### POWER MODE:

In diesem Funktionsfeld kann die Betriebsart der Leistungsnachregelung an der Sendeeinheit eingestellt werden.

#### STATIC:

Die Senderleistung wird entsprechend dem am Schieberegler **POWER [pm]** eingestellten Wert konstant gehalten (empfohlene Betriebsart). **POWER** kann mit Hilfe des Schiebereglers oder durch Eingabe in die Edit-Box eingestellt werden. Der Wert 1000 bedeutet volle Intensität an der Sendereinheit, beim Wert 0 wird die kleinste Intensität am Sender eingestellt.

#### DYNAMIC:

Die LED-Sendeleistung wird automatisch anhand der vom Gegenstand diffus zurückreflektierten Strahlungsmenge dynamisch geregelt. Der Regelkreis versucht anhand der an den Empfängern gemessenen Intensitäten die Sendeleistung automatisch so einzustellen, dass der Dynamikbereich, welcher mit **DYN WIN LO** und **DYN WIN HI** festgelegt wird, möglichst nicht verlassen wird.

#### LED MODE:

Hier kann eingestellt werden, wie die integrierte Lichtquelle des Sensors angesteuert wird.

**DC:** In diesem Mode ist der Sensor extrem schnell. Leider ist der Sensor im **DC** Mode leicht Fremdlicht empfindlich. Leuchtet jedoch die Fremdlichtquelle nicht direkt in den Empfänger des Sensors, dann wird das Signal nur sehr geringfügig beeinflusst.

**AC:** Hier ist der Sensor unabhängig gegenüber Fremdlicht. Dies wird dadurch erreicht, dass die integrierte Lichtquelle „moduliert“ wird. D.h. das Licht wird ein und ausgeschaltet. Im ausgeschalteten Zustand wird einfach der Fremdanteil im Signal ermittelt und vom eingeschalteten Zustand abgezogen.

**OFF:** Die interne Lichtquelle des Sensors wird ausgeschaltet. Jetzt kann der Sensor für sogenannte „Selbstleuchter“ verwendet werden. Selbstleuchter sind Lichtquellen, die aktiv Licht emittieren (LEDs, Lampen, etc.). Im **OFF** Mode können weder der **POWER MODE** noch **POWER** verstellt werden.

#### GAIN:

Hier wird die Verstärkung des Empfängers eingestellt. Es können 8 verschiedene Verstärkungsstufen eingestellt werden (AMP1 bis AMP8). **GAIN** sollte so eingestellt werden, dass der Sensor bei einem mittleren **POWER** Wert in seinem Dynamikbereich (Rot, Grün, Blau zwischen 2750 und 3750) arbeitet.

Im **AC** Mode wirkt sich **GAIN** direkt auf die Scanfrequenz aus. Die momentane Scanfrequenz wird in der Registerkarte **SCOPE** angezeigt.

#### AVERAGE:

In diesem Funktionsfeld wird die Anzahl der Abtastwerte (Messwerte) eingestellt, über die das am Empfänger gemessene Rohsignal gemittelt wird. Ein größerer **AVERAGE** Vorgabewert reduziert das Rauschen der Rohsignale der Empfangseinheit, gleichzeitig verringert sich die maximal erreichbare Schaltfrequenz des Sensors.

#### INTEGRAL:

In diesem Funktionsfeld wird die Anzahl der Abtastwerte (Messwerte) eingestellt, über die das am Empfänger gemessene Rohsignal aufsummiert wird. Durch diese Integralfunktion lassen sich auch extrem schwache Signale sicher erkennen. Ein größerer **INTEGRAL** Vorgabewert erhöht das Rauschen der Rohsignale der Empfangseinheit, gleichzeitig verringert sich die maximal erreichbare Schaltfrequenz des Sensors.

#### INFO:

Der **POWER** Schieberegler ist nur im **POWER MODE = STATIC** wirksam.

**DYN WIN LO** und **DYN WIN HI** sind nur im **POWER MODE = DYNAMIC** wirksam.

COLOR SPACE

XY INT
▼

**COLOR SPACE:**

Hier stellt man den zu berechnenden Farbraum ein.  
Zur Verfügung stehen **X Y INT** und **s i M**

Berechnung der Koordinaten:

**X** X-Wert der Lernfarbe (im Farbdreieck  
**s** Zahlenwert an der x-Achse: ROT-Farbanteil)

$$X = \frac{R}{R + G + B} * 4095$$

**s** wird angelehnt an die L\*a\*b\*  
Farbauswertemethode errechnet.

$$s = 5000 * \left( \left( \frac{R}{4096} \right)^{1/3} - \left( \frac{G}{4096} \right)^{1/3} \right) + 5000$$

**Y** Y-Wert der Lernfarbe (im Farbdreieck  
**i** Zahlenwert an der y-Achse: GRÜN-Farbanteil)

$$Y = \frac{G}{R + G + B} * 4095$$

**i** wird angelehnt an die L\*a\*b\*  
Farbauswertemethode errechnet.

$$i = 2000 * \left( \left( \frac{G}{4096} \right)^{1/3} - \left( \frac{B}{4096} \right)^{1/3} \right) + 2000$$

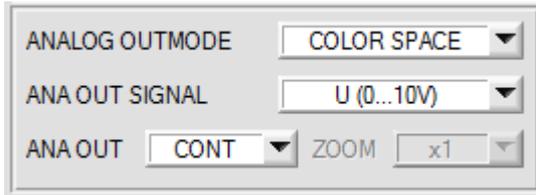
**INT** Intensitätswert der jeweiligen Farbe.

**M**

$$INT = \frac{R + G + B}{3}$$

**M** wird angelehnt an die L\*a\*b\*  
Farbauswertemethode errechnet.

$$M = 1160 * \left( \frac{G}{4096} \right)^{1/3}$$



**ANALOG OUTMODE:**

In diesem Funktionsfeld kann eingestellt werden, welche Signale der Sensor an seinen Analogausgängen ausgibt.

**OFF:** Es wird keine Analogsignal ausgegeben.

**R G B:**

Die Signale **Rot, Grün, Blau** werden mit einer Auflösung von 12 Bit erfasst.

Die Signale können daher Werte zwischen 0 und 4095 annehmen.

Diese Werte werden auf den entsprechenden Analogausgängen ausgegeben.

Mit **ANA OUT SIGNAL** wählt man ob die Analogsignale als Spannung von 0...+10V oder als Strom von 4...20mA ausgegeben werden.

**RGB MM:**

Solange der Eingang IN0 auf HI liegt, wird im Sensor ein maximaler und minimaler Wert für den roten, grünen und blauen Kanal ermittelt.

Das Analogsignal für **R G B** wird innerhalb der jeweiligen **MIN-MAX** Bereiche voll (0...+10V bzw. 4...20mA) ausgegeben.

Die gefundenen **MIN-MAX** Werte für die Kanäle **R G B** werden im Graphfenster unter dem Reiter **RGB** angezeigt.

**COLOR SPACE:**

**X Y INT:**

Die Farbkoordinaten **X, Y, INT** werden normiert auf 12 Bit berechnet.

Die Signale können daher Werte zwischen 0 und 4095 annehmen.

Diese Werte werden auf den entsprechenden Analogausgängen ausgegeben.

**s i M:**

Die Farbkoordinaten **s, i, M** werden angelehnt an die L\*a\*b\* Farbauswertemethode errechnet.

Aus dieser Berechnung ergeben sich folgende Wertebereiche:

**s:** 0 bis 10000    **i:** 0 bis 4000    **M:** 0 bis 1160

Die Werte für **s, i, M** werden auf 12 Bit normiert und auf den entsprechenden Analogausgängen ausgegeben.

**CS REF:**

Wenn der Eingang IN0 auf HI geht, werden die aktuellen Farbraumwerte **X, Y, INT** oder **s, i, M** als 5V Referenzwerte gespeichert. Wenn der jeweilige aktuelle Farbraumwert dem Referenzwert entspricht, wird an dem entsprechenden Analogausgang 5V ausgegeben. Weicht der aktuelle Wert vom Referenzwert ab, dann wird diese Abweichung um den Faktor **ZOOM** verstärkt am entsprechendem Analogausgang ausgehend von 5V ausgegeben

**Beispiel: ZOOM=x4, s(Referenz) = 6000, s(aktuell = 6100) → Abweichung = 100 (1 Digit entspricht ca. 2,44mV)**

Ausgegeben wird:  $5V + (100 \text{Digit} * 4(\text{Zoom}) * 2,44\text{mV}) = \text{ca. } 5,976V$ .

Die Referenzwerte für die **s, i, M** werden im Graphfenster unter dem Reiter **INTENSITY** angezeigt.

Da der Sensor nur über einen Hardware Eingang (IN0) verfügt, ist diese Eingabemöglichkeit nicht immer gegeben. Ist z.B. **ANA OUT = RGB MM** gewählt, dann wird der Eingang zur **MIN-MAX** Suche gebraucht und kann somit nicht für eine gesteuerte Ausgabe des Analogsignals verwendet werden.

Funktionsfelder, die nicht zur Verfügung stehen, werden entweder ausgeblendet oder inaktiv geschaltet.



**INO:**

Der Zustand von IN0 wird auf der Oberfläche durch die LED **INO** angezeigt.

Ist die LED schwarz, dann ist IN0=LO=0V.

Ist die LED grün, dann ist IN0=HI=+24V.

Die LED wird nur eingeblendet, wenn der Eingang IN0 verwendet wird.

## 2.3 Registerkarte REC

Die SPECTRO3-SLA-Scope Software beinhaltet einen Datenrekorder, welcher es erlaubt die vom Sensor erfassten und berechneten Daten abzuspeichern. Das aufgezeichnete File wird auf der Festplatte des PC abgespeichert und kann anschließend mit einem Tabellenkalkulationsprogramm ausgewertet werden.

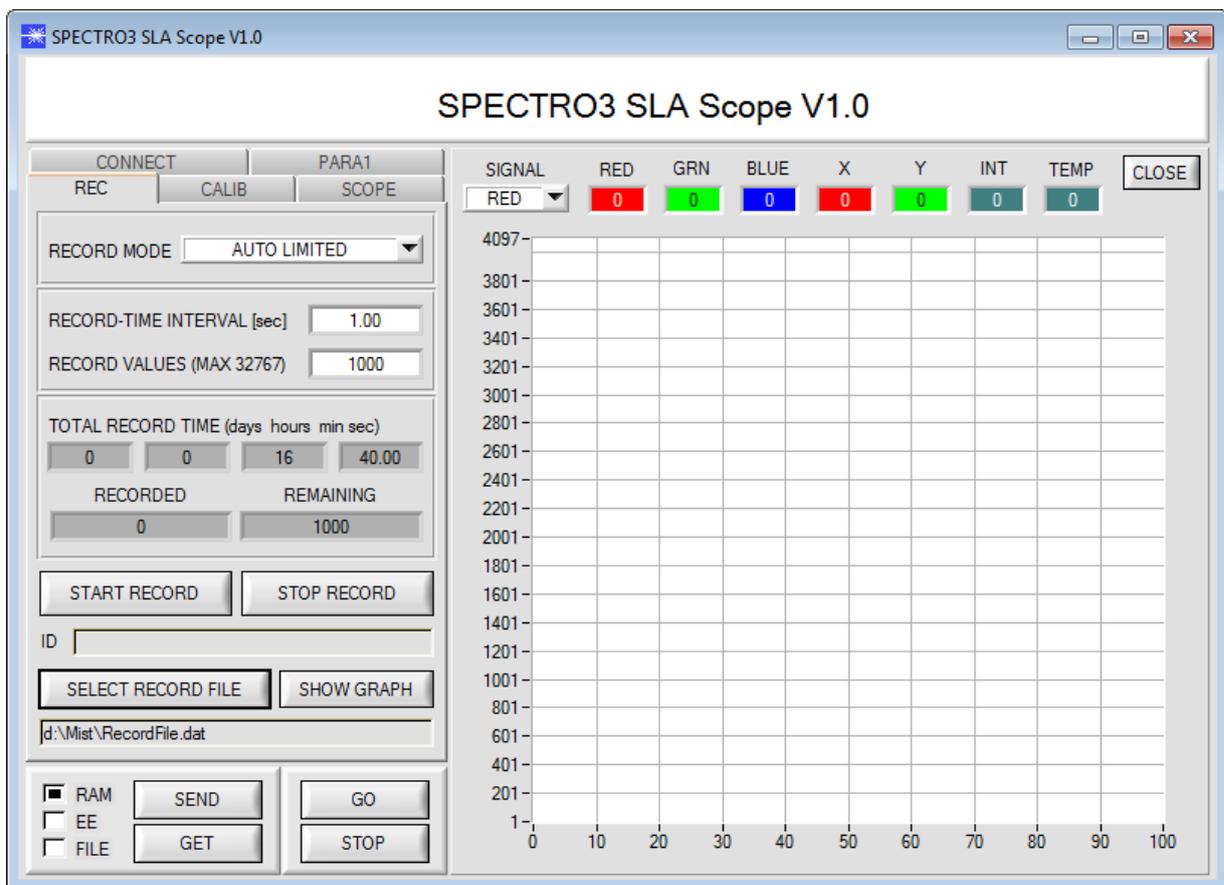
Führen Sie folgende Schritte durch, um Datenframes mit dem Recorder aufzuzeichnen:

### 1. Schritt:

Nach Drücken von **REC** öffnet sich folgendes Fenster:

Nach Drücken von **SHOW GRAPH** erscheint ein Panel, welches dem Benutzer erlaubt die verschiedenen Signale zu monitoren.

Über das DROP DOWN Menü **SIGNAL** kann zwischen den einzelnen Signalen hin und her geschaltet werden.



RECORD MODE <span style="float: right;">AUTO LIMITED ▼</span>	
RECORD-TIME INTERVAL [sec]	1.00
RECORD VALUES (MAX 32767)	1000
TOTAL RECORD TIME (days hours min sec)	
0	0
16	40.00
RECORDED	REMAINING
0	1000
<input type="button" value="START RECORD"/> <input type="button" value="STOP RECORD"/>	
ID <input style="width: 100%;" type="text"/>	
<input type="button" value="SELECT RECORD FILE"/> <input type="button" value="SHOW GRAPH"/>	
d:\Mist\RecordFile.dat	

**2. Schritt:**

Zur automatischen Aufzeichnung von mehreren Datenframes wählen Sie **AUTO LIMITED** unter **RECORD MODE** aus.

Geben Sie unter **RECORD-TIME INTERVAL [sec]** ein Zeitintervall für die Aufzeichnung ein, im Beispiel wurde 1 gewählt, d.h. jede Sekunde wird ein neuer Frame vom Sensor angefordert.

Geben Sie nun bei **RECORD VALUES [MAX 32767]** ein, wie viele Werte Sie maximal aufzeichnen wollen. Anmerkung: Die Aufzeichnung kann auch vorher durch **STOP RECORD** gestoppt werden, ohne dass die bisher aufgezeichneten Daten verloren gehen.

Bei **TOTAL RECORD TIME** wird in Tagen, Stunden, Minuten und Sekunden angezeigt, wie lange die Aufzeichnung dauert, wenn alle Daten aufgezeichnet werden.

**3. Schritt:**

Selektieren Sie über **SELECT RECORD FILE** ein File, in welches der Datenframe abgespeichert werden soll.

Sollten Sie einen bereits existierenden Filenamen auswählen, werden Sie gefragt, ob Sie das bestehende File überschreiben wollen oder nicht.

**4. Schritt:**

Durch Drücken von **START RECORD** starten Sie die automatische Aufzeichnung der Daten.

Der Recorder beginnt mit der Aufzeichnung. Dabei wird der Button **START RECORD** rot eingefärbt als Zeichen für eine aktive Aufzeichnung.

Die jeweiligen Datenframes werden in den Anzeigefenstern zur Ansicht gebracht.

Zusätzlich können Sie in den beiden Anzeigefenstern **RECORDED** und **REMAINING** kontrollieren, wie viele Datenframes schon aufgezeichnet wurden und wie viele noch aufzuzeichnen sind.

**Beachte:**

**Während der Aufzeichnung sind die beiden Eingabefelder RECORD-TIME INTERVAL und VALUES TO BE RECORDED inaktiv.**

**5. Schritt:**

Nachdem so viele Datenframes wie unter **RECORD VALUES [MAX 32767]** eingestellt aufgezeichnet worden sind bzw. durch Drücken von **STOP AUTO RECORD** erscheint ein Pop-up-Fenster, welches das Speichern des Files bestätigt.

Wenn Sie eine unbegrenzte Anzahl von Daten aufzeichnen wollen, wählen Sie unter **RECORD MODE** die Funktion **AUTO UNLIMITED**. Selektieren Sie ein gewünschtes Aufzeichnungsintervall und drücken Sie **START RECORD**.

Wenn Sie Daten „von Hand“ aufzeichnen wollen, wählen Sie unter **RECORD MODE** die Funktion **MANUAL RECORDING** aus.

Über **GO** beginnen Sie Daten vom Sensor einzulesen. Diese Daten werden in dem Anzeigefenster visualisiert. Durch Drücken von **CAPTURE DATA FRAME** wird ein Datenframe in das unter **SELECT RECORD FILE** ausgewählte File abgespeichert. In **RECORDED** wird die Summe der bereits aufgezeichneten Frames angezeigt.

**Hinweis:**

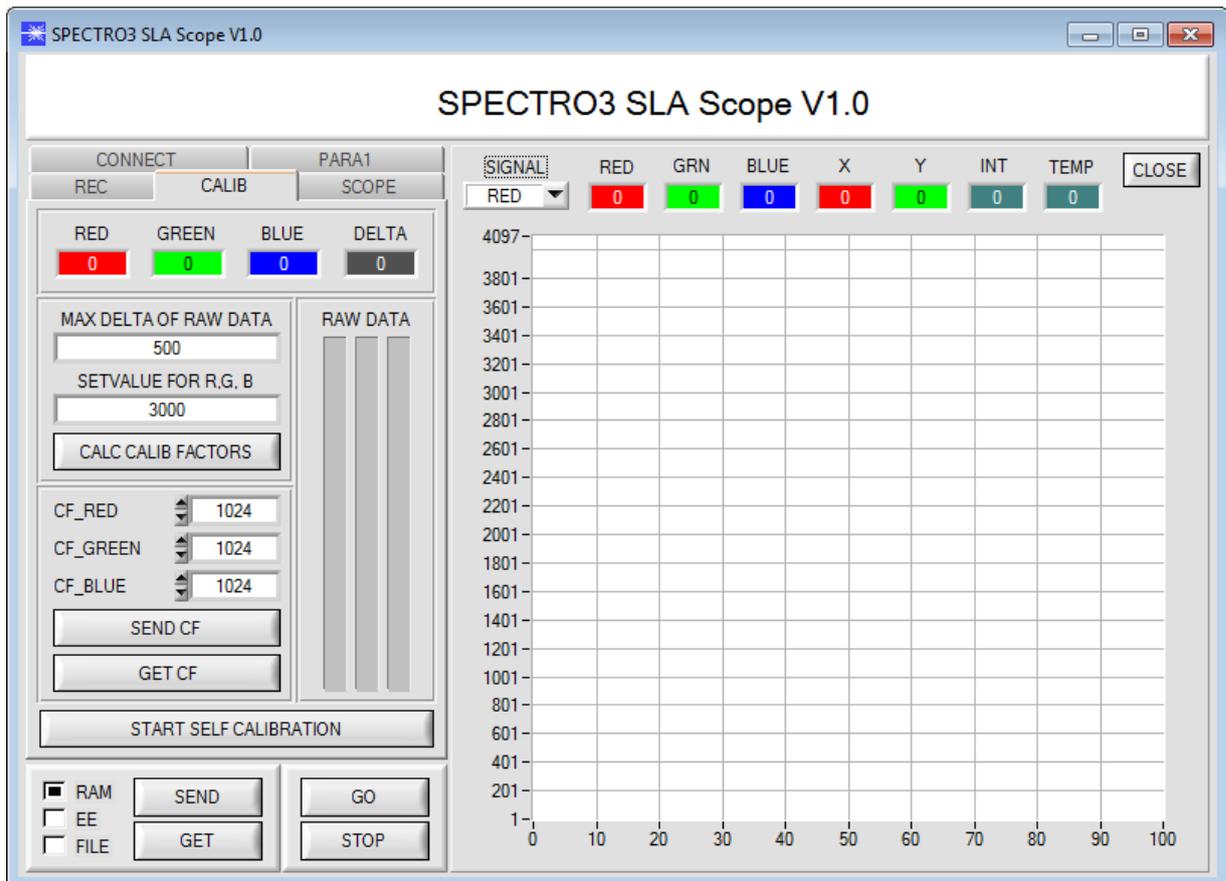
**Nach Drücken von START RECORD wird das File, welches unter SELECT RECORD FILE ausgewählt ist, gelöscht. Bei RECORD FRAME MANUALLY wird das File sofern es noch nicht besteht erzeugt. Sollte das File schon bestehen, werden die Daten an das bestehende File angehängt.**

## 2.4 Registerkarte CALIB

### 2.4.1 Weißlichtabgleich

Mit den Sensoren der SPECTRO-3-...-SLA Serie kann ein Weißlichtabgleich durchgeführt werden. Der Abgleich kann dabei auf eine beliebige weiße Oberfläche erfolgen. Alternativ dazu ist eine ColorChecker™ Tabelle erhältlich. Diese verfügt über 24 Farbfelder nach der CIE-NORM. Der Weißlichtabgleich bzw. die Kalibrierung kann auf eines der weißen Felder erfolgen.

Nach Drücken von **CALIB** öffnet sich folgendes Fenster:



Rechenbeispiel zur Bestimmung der Kalibrierungsfaktoren:

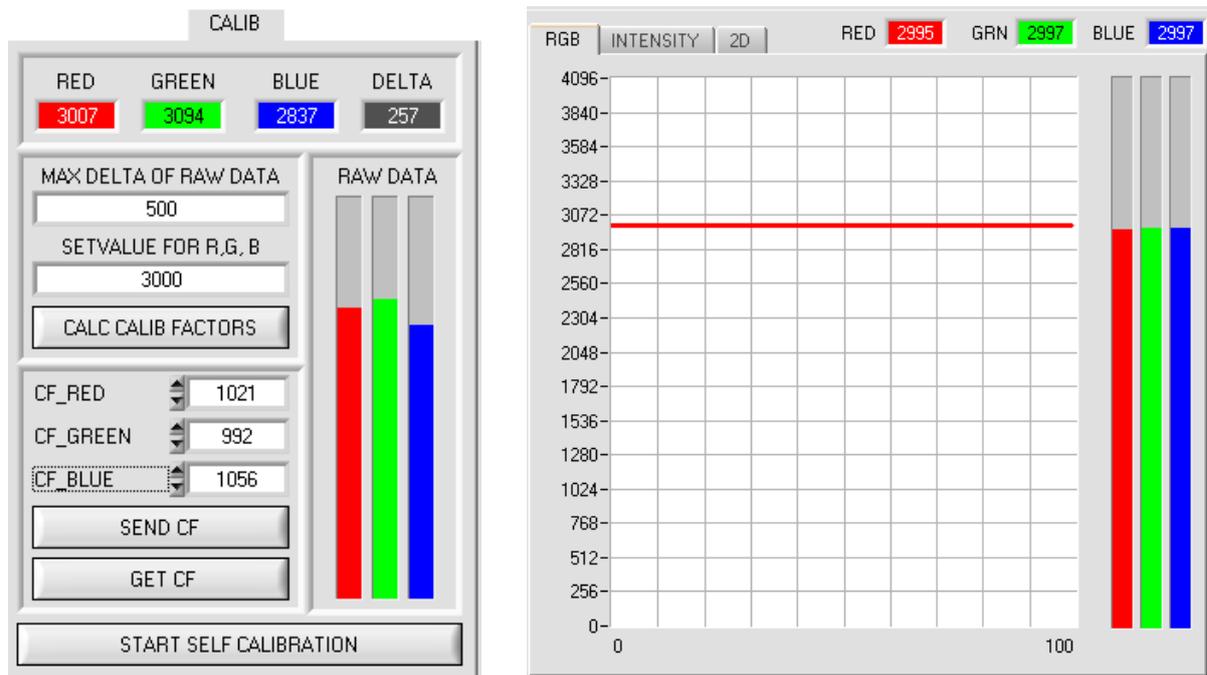
Wie Sie am Beispiel der unteren Graphik sehen, wurde ein POWER-Wert eingestellt, bei dem sich die drei Balken der Rohsignale **RAW DATA** im Dynamikbereich befinden. Jeder der drei Balken liegt bei ca. 3000 Digits. Bestimmen Sie nun einen Sollwert von 3000 (siehe **SETVALUE FOR R,G,B**) für die drei Balken. Nachdem durch Drücken von **CALCULATE CALIBRATION FACTORS** die Kalibrierung gestartet wurde, berechnet die Software automatisch die Kalibrierungsfaktoren für Kanal RED, Kanal GREEN und Kanal BLUE. Die Kalibrierungsfaktoren werden als Ganzzahl auf den Wert 1024 normiert.

Formel:

$$CF\_RED = (SETVALUE / RAW DATA RED) * 1024 = (3000 / 3007) * 1024 = 1021$$

$$CF\_GREEN = (SETVALUE / RAW DATA GREEN) * 1024 = (3000 / 3097) * 1024 = 992$$

$$CF\_BLUE = (SETVALUE / RAW DATA BLUE) * 1024 = (3000 / 2908) * 1024 = 1056$$



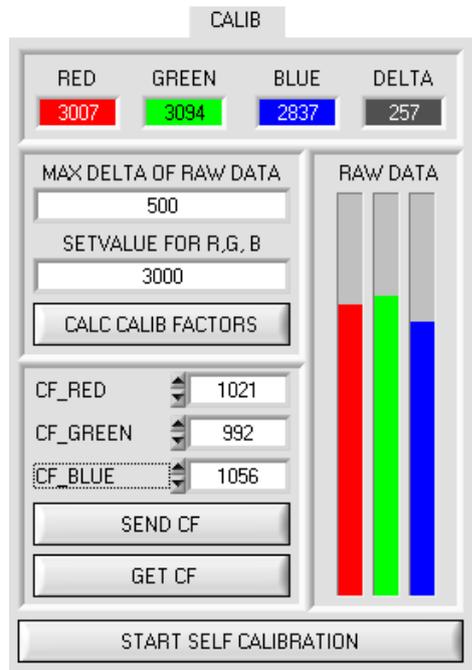
Nachdem die Kalibrierungsfaktoren von der Software auf der Benutzeroberfläche berechnet worden sind, werden sie automatisch in dem nichtflüchtigen Speicher **EEPROM** des Sensors abgelegt. Die Kalibrierung ist somit beendet und es kann im Hauptpanel weitergearbeitet werden.

Detektiert der Sensor ein Rohsignal, so beaufschlagt er dieses Rohsignal mit dem im **EEPROM** abgespeicherten Kalibrierungsfaktor. D.h., im Hauptpanel kommen nur die kalibrierten Daten für die Kanäle **RED**, **GREEN** und **BLUE** zur Anzeige. Die Auswertung von Seiten des Mikrokontrollers erfolgt auch ausschließlich mit den kalibrierten Daten.

Im Folgenden werden die einzelnen Schritte zur Kalibrierung der Sensoren beschrieben.

**INFO:** Die einzelnen Pop-up-Fenster sind als Hilfe gedacht, um Sie durch die Kalibrierung zu führen.

**ACHTUNG:** Grundvoraussetzung für eine erfolgreiche Kalibrierung ist, dass das Sensorfrontend auf eine weiße Oberfläche kalibriert wird.



**1. Schritt:**

Zuerst muss ein geeigneter **POWER** Wert gefunden werden, so dass sich die Rohdaten **RAW DATA** für **RED**, **GREEN** und **BLUE** im Dynamikbereich befinden (oberes Drittel der Balkenanzeige).

**2. Schritt:**

Nachdem Sie einen passenden **POWER**-Wert eingestellt haben, bestimmen Sie einen **SETVALUE FOR R,G,B**. Die Software berechnet nun die Kalibrierungsfaktoren so, dass über die Rohdaten dieser **SETVALUE** erreicht wird (siehe Rechenbeispiel oben).

**3. Schritt:**

Bestimmen Sie ein **MAX DELTA OF RAW DATA** (Software schlägt 500 vor).

Die Kalibrierung wird nur zugelassen, wenn das aktuelle **DELTA** der **RAW DATA** kleiner ist als **MAXI DELTA OF RAW DATA**.

**DELTA** ist das Maximum von **RED**, **GREEN** und **BLUE** minus dem Minimum von **RED**, **GREEN** und **BLUE**. Dies ist erforderlich, um sicher zu gehen, dass die Funktionalität des Sensors gegeben ist und die Kalibrierung auf eine weiße Oberfläche erfolgt.

**4. Schritt:**

Starten Sie die Kalibrierung durch Drücken von **CAL CALIB FACTORS**. Der Button beginnt Rot zu blinken, gleichzeitig werden über die Schnittstelle 100 Rohdaten aufgezeichnet, von denen der jeweilige Mittelwert von **RED**, **GREEN** und **BLUE** gebildet wird. Anhand dieser Mittelwerte und des **SETVALUES FOR R,G,B** werden die einzelnen Kalibrierungsfaktoren gebildet und in die entsprechenden Edit-Boxen eingetragen. Die Kalibrierungssoftware speichert die berechneten Kalibrierungsfaktoren automatisch in das EEPROM des Sensors. Anschließend wechselt die Software in den **GO**-Modus und bringt die **RAW DATA** sowie im Hauptpanel die kalibrierten Daten zur Anzeige. Beachten Sie, dass die Werte für **RED**, **GREEN** und **BLUE** im Hauptpanel ungefähr dem Wert vom **SETVALUE** entsprechen.

Sie können die Kalibrierungsfaktoren **CF\_RED**, **CF\_GREEN**, **CF\_BLUE** auch von Hand über die entsprechenden Eingabefelder ändern. Beachten Sie, dass Sie mit **SEND CF** diese Faktoren im EEPROM ablegen. Über **GET CF** werden die aktuell im EEPROM abgespeicherten Kalibrierungsfaktoren abgelesen.

Führt das Drücken von **CALC CALIB FACTORS** nicht zum Erfolg, dann folgen Sie den Hinweisen in den Pop-up-Fenstern.

Eine Kalibrierung war erst dann erfolgreich, wenn folgendes Pop-up-Fenster erscheint:

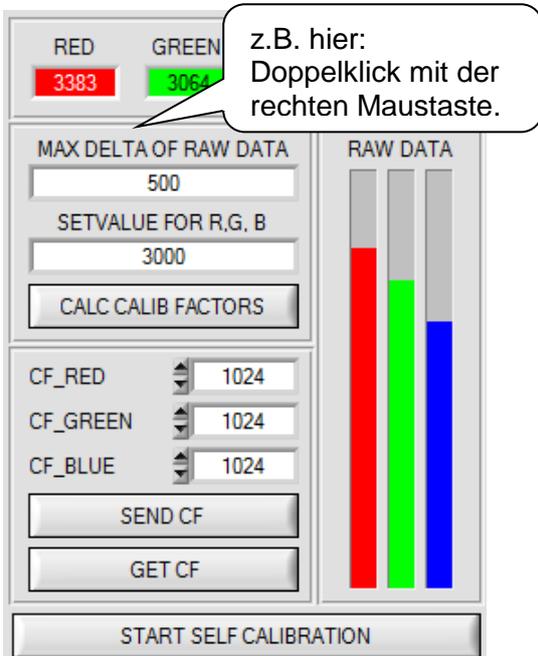


Durch Drücken von **START SELF CALIBRATION** kann man den Sensor die Kalibrierungsfaktoren selbst berechnen lassen. Man kann hier keinen **SETVALUE** und kein **MAX DELTA** vorgeben.

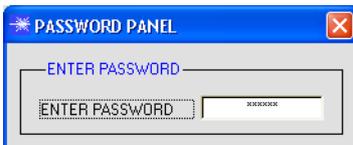
Nachdem der Sensor die Kalibrierfaktoren berechnet hat, zeigt er sie auf der Oberfläche an. Außerdem zeigt er den **SETVALUE**, den er zur Berechnung herangezogen hat, sowie **MAX DELTA**, das sich bei der Berechnung ergeben hat, in den entsprechenden Eingabefeldern an. Zur Bestätigung der errechneten Kalibrierungsfaktoren muss **SEND CF** gedrückt werden.

## 2.4.2 Offsetkalibrierung

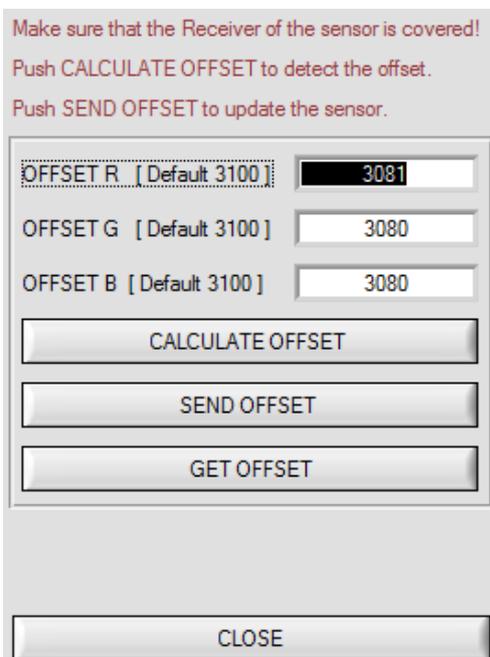
Damit beim Verwenden der Integralfunktion (Parameter **INTEGRAL**) nicht der elektronische Offset verstärkt wird, kann dieser durch eine Offsetkalibrierung bzw. Nullpunktkalibrierung, eliminiert werden. Die dazu notwendige Registerkarte ist durch ein Passwort geschützt, damit nicht versehentlich etwas verstellt werden kann.



Um zur Offsetkalibrierung zu gelangen, muss in der Registerkarte **CALIB** ein Doppelklick mit der rechten Maustaste exakt auf den Rahmen gemacht werden.



Anschließend wird nach dem Passwort verlangt. Das Passwort lautet: mellon



Jetzt müssen den Anweisungen in der Registerkarte gefolgt werden.

### ACHTUNG!

Bei der Offsetkalibrierung ist es sehr wichtig, dass der Empfänger absolut kein Fremdlicht sieht.

Bedecken Sie dazu den Empfänger des Sensors z.B. mit einem schwarzen, Licht undurchlässigem Tuch.

**Dies ist absolut notwendig für einen einwandfreien Offsetabgleich.**

Drücken Sie jetzt **CALCULATE OFFSET**. Die Offsetwerte für Rot, Grün und Blau sollten bei ca. 3080 plus minus 40 liegen.

Erst jetzt können die Offsetwerte dem Sensor durch Drücken von **SEND OFFSET** gesendet werden.

Über **GET OFFSET** kann man kontrollieren, ob die Daten gesendet wurden.

## 2.5 Registerkarte SCOPE

In der Registerkarte SCOPE wurde ein Oszilloskop nachgebildet.

Angezeigt werden jetzt wahlweise über **TRIG MODE** die Signale **R G B, X Y INT** bzw. **s i M**, das ausgegebene Analogsignal (abhängig von der Einstellung in **ANA OUT**) sowie der Zustand der digitalen Ausgänge.

Durch Drücken von **GET CYCLE TIME** erhält man die aktuelle Sensor Scanfrequenz in **[Hz]** und **[ms]**. Die aktuelle Scanfrequenz muss ermittelt werden, damit **deltaX[ms]** richtig ermittelt werden kann. Man muss dem Sensor zur Ermittlung der richtigen Scanfrequenz 8 Sekunden Zeit geben, bevor man **GET CYCLE TIME** betätigt.

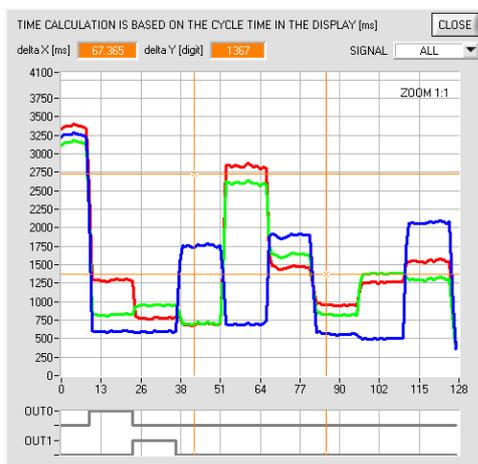
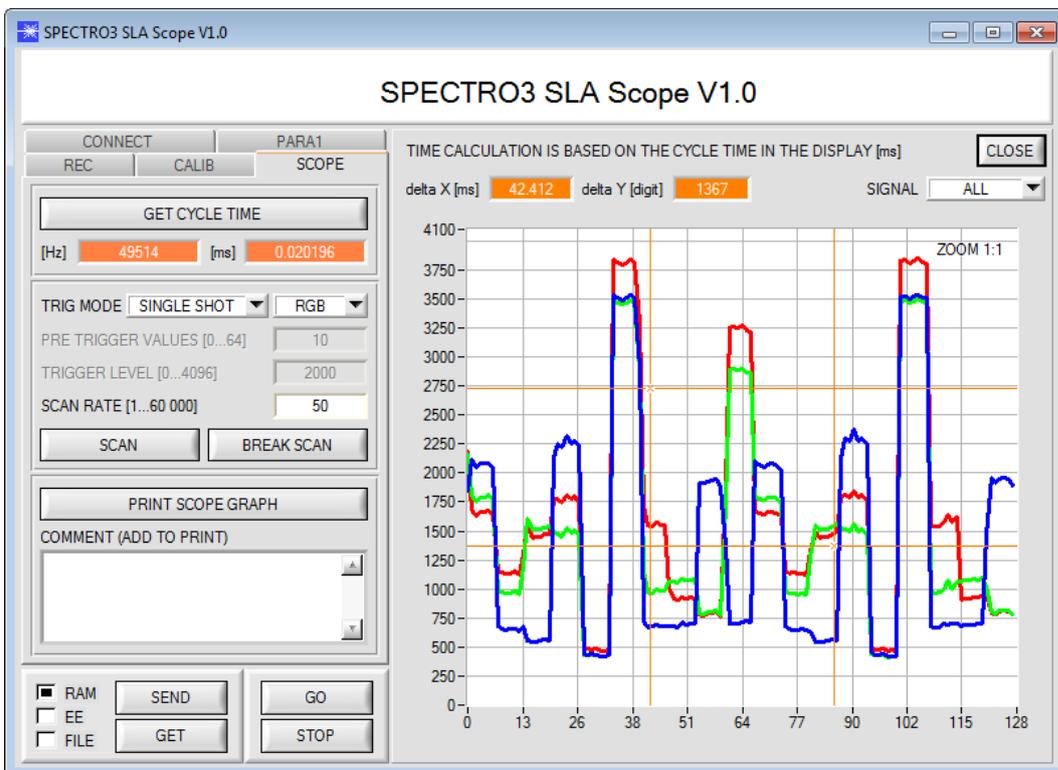
Im **TRIG MODE = SINGLE SHOT** wird nach Drücken von **SCAN** ein Datenframe aufgezeichnet und im Graphen zur Anzeige gebracht.

Im **TRIG MODE = FALLING EDGE** und **RIISING EDGE** kann man eine getriggerte Aufzeichnung mit Drücken von **SCAN** starten. Dabei hat man die Möglichkeit über **TRIGGER LEVEL** einen Trigger-Start festzulegen.

Getriggert wird entweder auf **BLAU, INT** oder **M**, je nachdem welches Signal aufgezeichnet werden soll. Dies ist im Graphen die blaue Linie.

Im **TRIG MODE= INTERN C-No.0** startet die Aufzeichnung selbständig, sobald C-No. 0 erkannt wird. Über **TRIG MODE= EXTERN IN0** startet man die Aufzeichnung von extern über den Eingang IN0.

Über **SCAN-RATE** kann man die Aufzeichnung verzögern oder beschleunigen. Dies entspricht, dem bei einem Oszilloskop bekannten TIMEBASE. Über **PRE TRIGGER VALUES** legt man fest, wie viele Werte vorm eigentlichen Trigger-Start noch angezeigt werden sollen.



Durch Halten der Steuerungstaste Strg (CTRL) und Aufziehen eines Fensters mit der Maus im Graphen kann in den Graphen gezoomt werden.

Mit **ZOOM 1:1** wird die Zoomfunktion wieder aufgehoben.

Die beiden orangen Cursor können mit der Maus verschoben werden. Dabei werden die Displays **deltaX[ms]** und **deltaY[digit]** aktualisiert.

**deltaX[ms]** zeigt die Zeit zwischen den Cursor in X-Richtung.  
**deltaY[digit]** zeigt die Differenz der beiden Cursor in Y-Richtung in Digit.

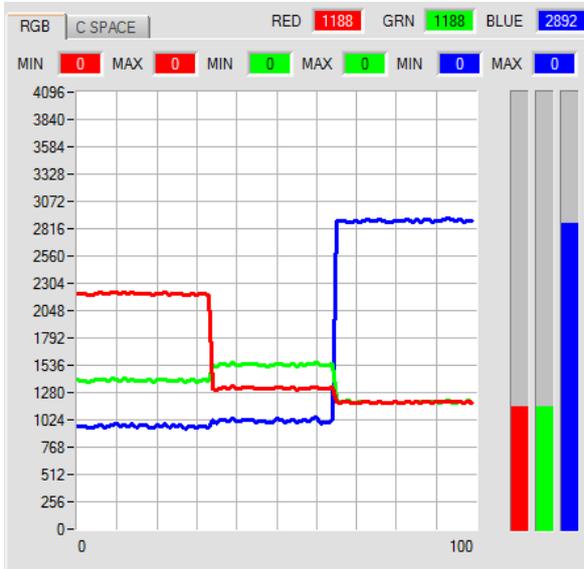
Unter **SIGNAL** kann man einzelne Kurven darstellen.

Mit **PRINT SCOPE GRAPH** wird der aktuelle Bildschirm, zusammen mit dem Text der im Textfeld **COMMENT** steht, ausgedruckt.

## 2.6 Graphische Anzeigeelemente

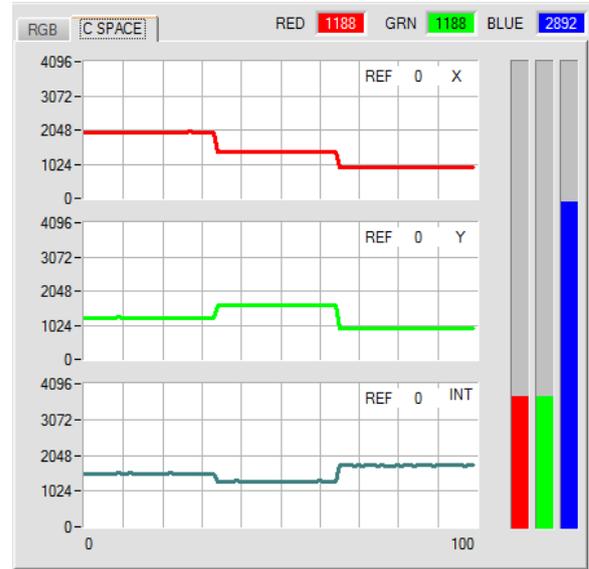
### Registerkarte RGB:

Aktuelle Rohsignale des 3-fach Empfängers (Rot, Grün, Blau) werden angezeigt.



### Registerkarte INTENSITY:

Aktuell ermittelte Intensität von X, Y, INT bzw. s, i, M wird angezeigt.



TEMP

30

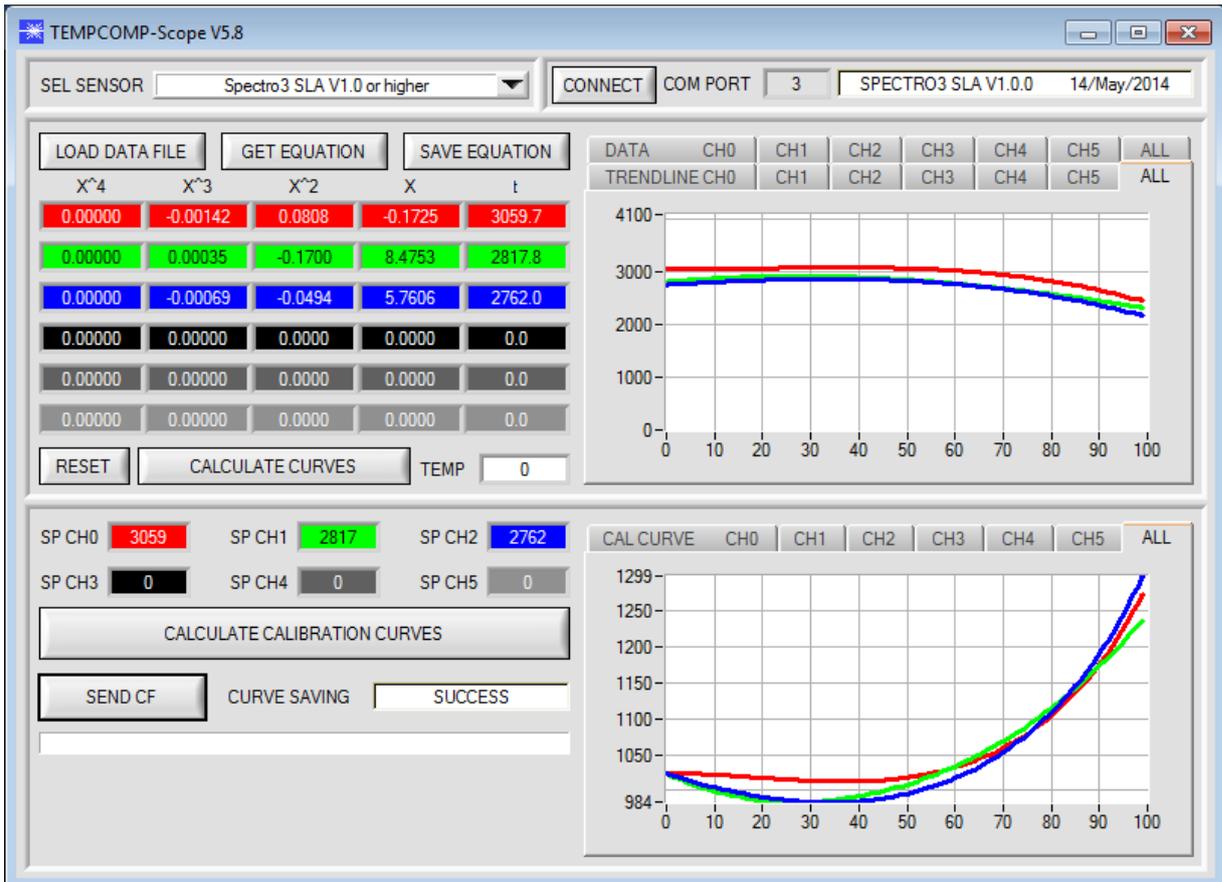
### TEMP:

In diesem Display wird die im Sensorgehäuse herrschende Temperatur angezeigt. Die Anzeige entspricht **NICHT** Grad Celsius oder Fahrenheit.

### 3. Bedienung der TEMPCOMP-Scope Software

Sollte bei einem Firmwareupdate etwas schief gehen, so dass die im EEPROM gespeicherten Temperaturkennlinien verloren gegangen sind, dann ist es notwendig, diese Kennlinien wieder zu erstellen. Dazu benötigen Sie ein File mit den entsprechenden Daten. Dieses File erhalten Sie von Ihrem Lieferanten.

Zur Temperaturkompensation starten Sie bitte die entsprechende, auf der CD mitgelieferte Software **TEMPCOMP-Scope**. Bitte stellen Sie sicher, dass Sie mit dem Sensor verbunden sind. Eventuell müssen Sie die Verbindung über **CONNECT** auswählen. Stellen Sie unter **SELECT SENSOR** den richtigen Sensor ein, sofern dies nicht automatisch erfolgt.

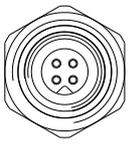


1. Schritt: Laden Sie jetzt über **GET EQUATION** oder **LOAD DATA FILE** das Temperaturkompensations-File, welches Sie von Ihrem Lieferanten erhalten haben.
2. Schritt: Drücken Sie **CALCULATE CURVES**, um die Daten im Graphen anzuzeigen.
3. Schritt: Wählen Sie die sensorinterne Betriebstemperatur (nicht in °C), welche der Sensor bei einer Umgebungstemperatur von 20° hat, fall diese nicht schon automatisch geschehen ist. Der Wert müsste in der File-Bezeichnung enthalten sein.
4. Schritt: Drücken Sie **CALCULATE CALIBRATION CURVES**, um die Ausgleichsgeraden zu berechnen.
5. Schritt: Mit Drücken von **SEND CF** werden die Ausgleichsgeraden im **EEPROM** des Sensors abgelegt.
6. Schritt: Eine erfolgreiche Temperaturkompensation sehen Sie, wenn der Status **SUCCESS** angezeigt wird.

Anmerkung! Wenn Sie das Temperaturkompensations-File nicht gleich zur Hand haben, dann starten Sie einfach die TEMPCOMP-Scope Software. Bauen Sie eine Verbindung auf, soweit noch nicht vorhanden, und drücken Sie einfach **SEND-CF**. Der Sensor funktioniert jetzt wie gehabt, ist jedoch nicht temperaturkompensiert.

## 4. Anschlussbelegung der SPECTRO-3-...-SLA Farbsensoren

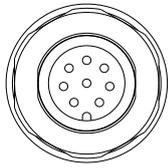
### Anschluss SPECTRO-3-...-SLA an PC:

<b>4-pol. M5 Buchse (Typ Binder 707)</b> <b>SPECTRO-3-...-SLA/PC-RS232</b>		
Pin-Nr.:		Belegung:
1		+24VDC (+Ub)
2		0V (GND)
3		Rx0
4		Tx0

**Anschlusskabel zur Wahl:**

cab-las4/PC-...  
cab-las4/USB-...  
SI-RS232/Ethernet-4-...

### Anschluss SPECTRO-3-...-SLA an SPS:

<b>8-pol. M12 Stecker (Typ Binder 763/713)</b> <b>SPECTRO-3-...-SLA/SPS</b>		
Pin-Nr.:	Farbe:	Belegung:
1	Weiß	0V (GND)
2	Braun	+24VDC (± 10%)
3	Grün	IN0
4	Gelb	Not connected
5	Grau	Not connected
6	Rosa	ANA OUT (Analog <b>R</b> oder <b>X</b> oder <b>s</b> : 0...+10V bzw. 4...20mA)
7	Blau	ANA OUT (Analog <b>G</b> oder <b>Y</b> oder <b>i</b> : 0...+10V bzw. 4...20mA)
8	Rot	ANA OUT (Analog <b>B</b> oder <b>INT</b> oder <b>M</b> : 0...+10V bzw. 4...20mA)

**Anschlusskabel:**

cab-M12/8-...-shd (geschirmt)

## 5. RS232 Schnittstellenprotokoll

The sensors of the SPECTRO-3-...-SLA 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> DYN WIN LO	Word	Low limit for dynamic window when POWER MODE=dynamic (0...4095)
<b>Para5:</b> DYN WIN HI	Word	High limit for dynamic window when POWER MODE=dynamic (0...4095)
<b>Para6:</b> LED MODE	Word	Control for the internal light source DC, AC, OFF coded to (0,1,2)
<b>Para7:</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>Para8:</b> INTEGRAL	Word	Signal integration (1...250)
<b>Para9:</b> COLOR SPACE	Word	Color Space X Y INT or s i M coded to (0,1)
<b>Para10:</b> ANALOG OUTMODE	Word	Function of the analogue outmode: OFF, RGB, RGB MM, COLOR SPACE, CS REF coded to (0,1,2,3,4)
<b>Para11:</b> ANA OUT SIGNAL	Word	Analogue output signal: U(0...10V), I(4...20mA) coded to (0,1)
<b>Para12:</b> ANA OUT	Word	Function of analogue out: CONT, IN0 L-->H coded to (0,1)
<b>Para13:</b> ANA ZOOM	Word	Zoom factor at ANALOG OUTMODE = CS REF: x1, x2, x4, x8, x16, x32, x64, x128 Coded to (0,1,2,3,4,5,6,7)

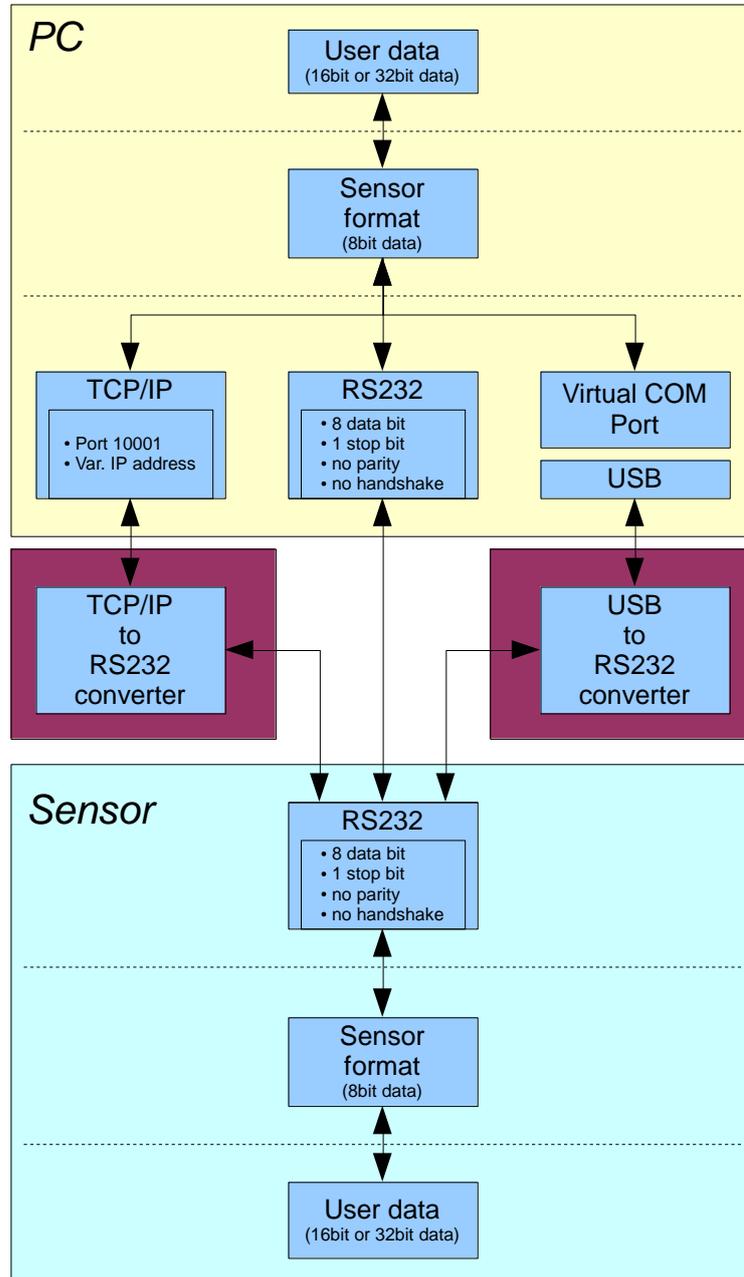
Upon request, the data acquired and processed by the sensor are sent by the sensor in the following sequence.

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> IN0	Word	IN0 is 1 when input IN0 is HI
<b>DatVal8:</b> TEMP	Word	Temperature in the sensor ( <b>not</b> in °C or °F)
<b>DatVal9:</b> RAW RED	Word	None Calibrated and none temperature compensated signal from channel red
<b>DatVal10:</b> RAW GREEN	Word	None Calibrated and none temperature compensated signal from channel green
<b>DatVal11:</b> RAW BLUE	Word	None Calibrated and none temperature compensated signal from channel blue
<b>DatVal12:</b> MIN RED	Word	Calibrated and temperature compensated minimum signal from channel red
<b>DatVal13:</b> MIN GREEN	Word	Calibrated and temperature compensated minimum signal from channel green
<b>DatVal14:</b> MIN BLUE	Word	Calibrated and temperature compensated minimum signal from channel blue
<b>DatVal15:</b> MAX RED	Word	Calibrated and temperature compensated maximum signal from channel red
<b>DatVal16:</b> MAX GREEN	Word	Calibrated and temperature compensated maximum signal from channel green
<b>DatVal17:</b> MAX BLUE	Word	Calibrated and temperature compensated maximum signal from channel blue
<b>DatVal18:</b> REF CSX	Word	Reference Value when ANALOG OUTMODE = CS REF
<b>DatVal19:</b> REF CSY	Word	Reference Value when ANALOG OUTMODE = CS REF
<b>DatVal20:</b> REF CSI	Word	Reference Value when ANALOG OUTMODE = CS REF

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 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Data	Byte9 Data	...	Byte n+6 Data	Byte n+7 Data
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	Data1 (lo byte)	Data1 (hi byte)	...	Data n/2 (lo byte)	Data n/2 (hi byte)

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

Number	ORDER (header byte no. 2)	Example
0	Sensor answers with order=0 if a communication error occurs. ARG=1: Invalid order number was sent to the sensor ARG=2: General communication error (wrong baudrate, overflow, ...)	
1	Write parameter to the RAM of the sensor	order=1
2	Read parameter from the RAM of the sensor	order=2
3	Load parameter and actual Baudrate from RAM to EEPROM of the sensor	order=3
4	Load parameter from EEPROM to RAM of the sensor	order=4
5	Read CONNECTION OK and serial number from sensor	order=5
6	Free	
7	Read Firmware String and serial number from sensor	order=7
8	Read data values from sensor	order=8
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.

DATA FRAME PC → Sensor

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Data	Byte9 Data	Byte10 Data	Byte11 Data
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	Para1 (lo byte)	Para1 (hi byte)	Para2 (lo byte)	Para2 (hi byte)
85 (dec)	1	0	0	26	0	95	139	244	1	0	0
		ARG=0		LEN=26				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	128	12	228	12	0	0	5	0	1	0
Para3=1		Para4=3200		Para5=3300		Para6=0		Para7=2		Para8=1	

Byte24 Data	Byte25 Data	Byte26 Data	Byte27 Data	Byte28 Data	Byte29 Data	Byte30 Data	Byte31 Data	Byte30 Data	Byte31 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)
0	0	1	0	0	0	0	0	0	0
Para9=0		Para10=1		Para11=0		Para12=0		Para 13=0	

DATA FRAME Sensor → PC

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

DATA FRAME PC → Sensor

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	
85 (dec)	2	0	0	0	0	170	185	
ARG=0			LEN=0					

DATA FRAME Sensor → PC

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Data	Byte9 Data	Byte10 Data	Byte11 Data	
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	Para1 (lo byte)	Para1 (hi byte)	Para2 (lo byte)	Para2 (hi byte)	
85 (dec)	2	0	0	26	0	95	210	244	1	0	0	
ARG=0			LEN=26				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	128	12	228	12	0	0	5	0	1	0
Para3=1		Para4=3200		Para5=3300		Para6=0		Para7=2		Para8=1	

Byte24 Data	Byte25 Data	Byte26 Data	Byte27 Data	Byte28 Data	Byte29 Data	Byte30 Data	Byte31 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)
0	0	1	0	0	0	0	0
Para9=0		Para10=1		Para11=0		Para12=0	

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

DATA FRAME PC → Sensor

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	
85 (dec)	3	0	0	0	0	170	142	
ARG=0			LEN=0					

• DATA FRAME Sensor → PC

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	
85 (dec)	3	0	0	0	0	170	142	
ARG=0			LEN=0					

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

DATA FRAME PC → Sensor

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	4	0	0	0	0	170	11
ARG=0			LEN=0				

DATA FRAME Sensor → PC

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	4	0	0	0	0	170	11
ARG=0			LEN=0				

**Example order=5:** Read CONNECTION OK from sensor.

DATA FRAME PC → Sensor

ARG determines the serial number of the sensor

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	5	0	0	0	0	170	60
ARG=0			LEN=0				

DATA FRAME Sensor → PC

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	5	170	0	0	0	170	178
ARG=170			LEN=0				

**Example order=7:** Read Firmware String from sensor

DATA FRAME PC → Sensor

ARG determines the serial number of the sensor

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	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> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	ASCII	ASCII	ASCII	ASCII
85 (dec)	7	0	0	72	0	183	38	F	I	R	M
ARG=0				LEN=72							

Byte12 Data	Byte13 Data	Byte14 Data	Byte15 Data	Byte16 Data	Byte17 Data	Byte18 Data	Byte19 Data	Byte20 Data	Byte21 Data	Byte22 Data	Byte23 Data
ASCII											
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											
											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											
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											

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											

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

**Example order=8:** Read data values from sensor.

DATA FRAME PC → Sensor

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	8	0	0	0	0	170	118
ARG=0				LEN=0			

DATA FRAME Sensor → PC

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)	DatVal1 (lo byte)	DatVal1 (hi byte)	DatVal2 (lo byte)	DatVal2 (hi byte)
85 (dec)	8	0	0	40	0	55	43	54	10	151	6
ARG=0				LEN=40				DatVal1=2614		DatVal2 =1687	

Byte12 Data	Byte13 Data	Byte14 Data	Byte15 Data	Byte16 Data	Byte17 Data	Byte18 Data	Byte19 Data	Byte20 Data	Byte21 Data	Byte22 Data	Byte23 Data
DatVal3 (lo byte)	DatVal3 (hi byte)	DatVal4 (lo byte)	DatVal4 (hi byte)	DatVal5 (lo byte)	DatVal5 (hi byte)	DatVal6 (lo byte)	DatVal6 (hi byte)	DatVal7 (lo byte)	DatVal7 (hi byte)	DatVal8 (lo byte)	DatVal8 (hi byte)
153	4	162	7	237	4	34	7	0	0	32	0
DatVal3=1177		DatVal4=1954		DatVal5=1261		DatVal6=1826		DatVal7=0		DatVal8=32	

Byte24 Data	Byte25 Data	Byte26 Data	Byte27 Data	Byte28 Data	Byte29 Data	Byte30 Data	Byte31 Data	Byte32 Data	Byte33 Data	Byte34 Data	Byte35 Data
DatVal 9 (lo byte)	DatVal 9 (hi byte)	DatVal 10 (lo byte)	DatVal 10 (hi byte)	DatVal 11 (lo byte)	DatVal 11 (hi byte)	DatVal 12 (lo byte)	DatVal 12 (hi byte)	DatVal 13 (lo byte)	DatVal 13 (hi byte)	DatVal 14 (lo byte)	DatVal 14 (hi byte)
54	10	151	6	153	4	0	0	0	70	0	0
DatVal9=2614		DatVal10=1687		DatVal11=1177		DatVal12=0		DatVal13=0		DatVal14=0	

Byte36 Data	Byte37 Data	Byte38 Data	Byte39 Data	Byte40 Data	Byte41 Data	Byte42 Data	Byte43 Data	Byte44 Data	Byte45 Data	Byte46 Data	Byte47 Data
DatVal 15 (lo byte)	DatVal 15 (hi byte)	DatVal 16 (lo byte)	DatVal 16 (hi byte)	DatVal 17 (lo byte)	DatVal 17 (hi byte)	DatVal 18 (lo byte)	DatVal 18 (hi byte)	DatVal 19 (lo byte)	DatVal 19 (hi byte)	DatVal 20 (lo byte)	DatVal 20 (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=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> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (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> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	CF RED (lo byte)	CF RED (hi byte)	CF GREEN (lo byte)	CF GREEN (hi byte)
85 (dec)	103	0	0	10	0	212	28	228	3	223	3
ARG=0				LEN=28				CF_RED = 996		CF_GREEN = 991	

Byte12 Data	Byte13 Data	Byte14 Data	Byte15 Data	Byte16 Data	Byte17 Data
CF BLUE (lo byte)	CF BLUE (hi byte)	SET VALUE (lo byte)	SET VALUE (hi byte)	MAX DELTA (lo byte)	MAX DELTA (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 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	105	0	0	0	0	170	130
ARG=0				LEN=0			

DATA FRAME Sensor → PC

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Data	Byte9 Data	Byte10 Data	Byte11 Data
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	lo word lo byte	lo word hi byte	hi word lo byte	hi word hi byte
85 (dec)	105	0	0	8	0	206	163	40	28	2	0
ARG=0				LEN=8				CYCLE COUNT = 138280			

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

$$\text{Cycle Time [Hz]} = \text{CYCLE COUNT} / (\text{COUNTER TIME} * 0,01)$$

$$\text{Cycle Time [ms]} = (\text{COUNTER TIME} * 0,01) / \text{CYCLE COUNT}$$

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

DATA FRAME PC → Sensor

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	190	1	0	0	0	170	14
		ARG=1		LEN=0			

New baud rate is determined by argument.

ARG=0: baud rate = 9600

ARG=1: baud rate = 19200

ARG=2: baud rate = 38400

ARG=3: baud rate = 57600

ARG=4: baud rate = 115200

DATA FRAME Sensor → PC

Byte0 Header	Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)
85 (dec)	190	0	0	0	0	170	195
		ARG=0		LEN=0			