

# Bedienungsanleitung Software SPECTRO2-Scope V1.1

(PC-Software für Microsoft® Windows® 7, Vista, XP, Me, 2000)

## für Sensoren der SPECTRO-2 Serie

Die vorliegende Bedienungsanleitung dient zur Installation und Inbetriebnahme der PC-Software für Sensoren der SPECTRO-2 Serie. Zur Unterstützung der Inbetriebnahme der Sensoren werden in dieser Bedienungsanleitung die einzelnen Funktionselemente der graphischen Windows® Benutzeroberfläche erklärt.

Sensoren der SPECTRO-2 Serie sind zweikanalig aufgebaut. D.h. sie erfassen das Analogsignal von zwei Empfängern und werten diese aus. Ebenso stehen zwei unabhängig voneinander einstellbare Sendequellen zur Verfügung. Als Sender können dabei verschiedene Lichtquellen wie z.B. Weißlicht, UV-Licht oder IR-Licht dienen. Der Empfänger ist dem Sender entsprechend angepasst.

Das ausgewertete Signal wird über einen Spannungsausgang oder über einen Stromausgang ausgegeben.

Mit Hilfe der Software können verschiedene Auswertemodi für das Analogsignal gewählt werden.

Über 2 digitale Ausgänge wird der Zustand des Ausgangssignals abhängig vom Auswertemodus ausgegeben.

Ein digitaler Eingang ermöglicht ein externes „Teachen“ des Sensors.

Ein weiterer Eingang ermöglicht ein „Einfrieren“ des analogen Ausgangssignals bei einer positiven Eingangsfanke.

Die Signalerfassung mit dem SPECTRO-2 Sensor 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 so genannte „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“.

Ein Mikrokontroller sorgt für eine 12-Bit Analog/Digital-Wandlung des Analogsignals, dadurch kann das Signal aufgezeichnet und ausgewertet werden. Zusätzlich bietet der SPECTRO-2 Sensor verschiedene Optionen zur intelligenten Signalaufarbeitung wie z.B. Verschmutzungskompensation.

Es können wahlweise über RS232 oder Ethernet (mit Hilfe eines Ethernet-Adapters) Parameter und Messwerte zwischen PC und Sensor ausgetauscht werden. Sämtliche Parameter können über die Schnittstelle im nichtflüchtigen EEPROM des Sensors abgelegt werden.

Die PC-Software erleichtert die Parametrisierung, die Diagnose und das Einjustieren des Sensorsystems (Oszilloskop-Funktion). Ferner verfügt die Software über die Funktion eines Datenrecorders, mit dessen Hilfe Daten automatisch aufgezeichnet werden und auf der Festplatte im PC gespeichert werden.

SPECTRO-2 Sensoren sind über einen Bereich von 0°C bis 80°C temperaturkompensiert.

Sollte ein Firmware-Update erforderlich sein, kann dieses sehr einfach über RS232 auch im eingebauten Zustand des Sensorsystems durchgeführt werden.

Nach erfolgter Parametrisierung arbeitet der Sensor im STAND-ALONE Betrieb ohne PC weiter.

## 0 Inhalt

	Seite
1 Installation der SPECTRO2-Scope Software .....	3
2 Bedienung der SPECTRO2-Scope Software .....	4
2.1 Registerkarte (Reiter oder Tab) CONNECT (Verbindungsaufbau) .....	5
2.2 Registerkarte PARA, Taste SEND, GET, GO, STOP (Parametrierung und Datenaustausch) .....	7
2.3 Registerkarte TEACH (Einlernen und Verarbeitung des Analogsignals) .....	11
2.4 Graphische Anzeigeelemente .....	17
2.5 Registerkarte CONVERSION .....	18
2.6 Registerkarte RECORDER (Datenaufzeichnung) .....	19
2.7 Registerkarte SCOPE .....	21
2.8 Offsetkalibrierung .....	22
3 Bedienung der TEMPCOMP-Scope Software .....	23
4 Anschlussbelegung .....	24
5 RS232 Schnittstellenprotokoll .....	25

**Shortcuts:**

SEND	F9
GET	F10
GO	F11
STOP	F12

# 1 Installation der SPECTRO2-Scope Software

Für eine erfolgreiche Installation der SPECTRO2-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

Installieren Sie nun die Software wie im Folgenden beschrieben:

Die SPECTRO2-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:

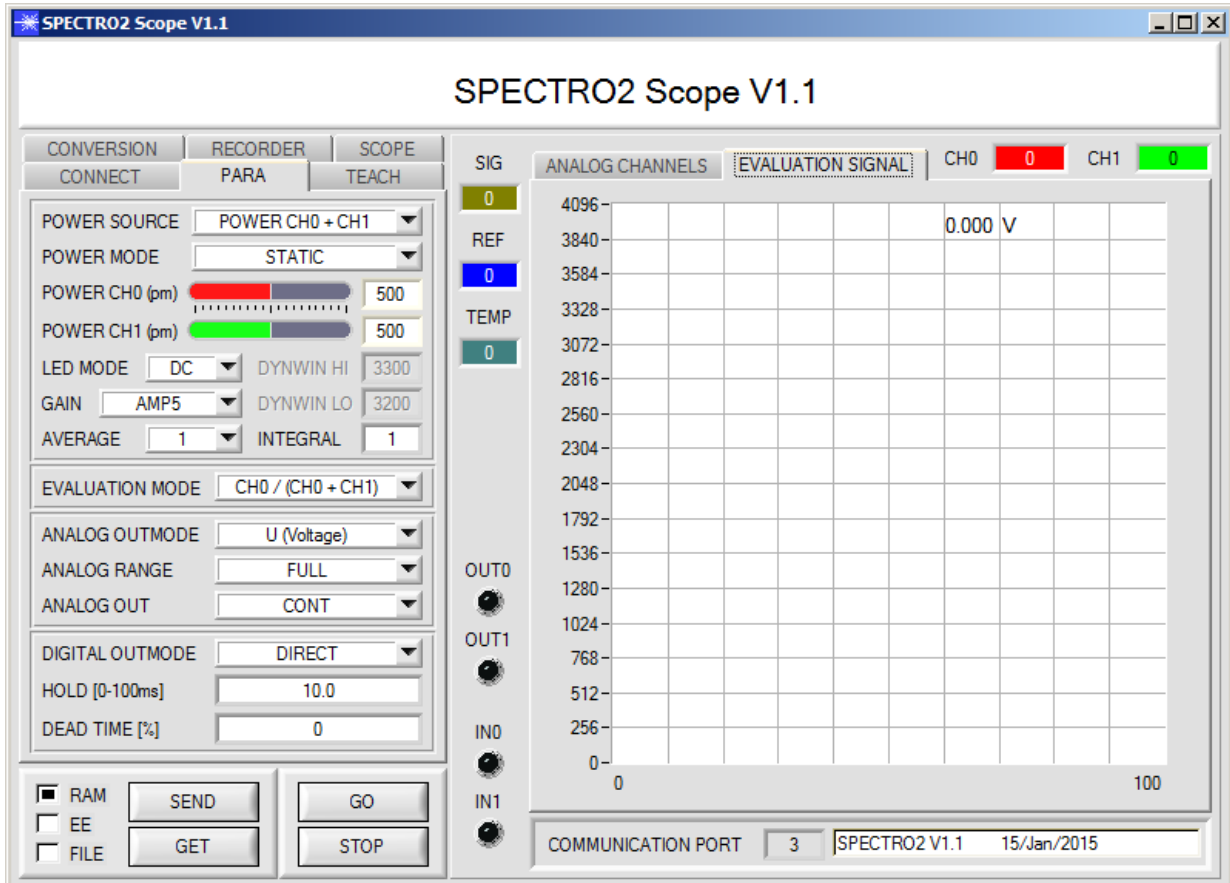
1. Sie können die Software direkt von der Installations-CD-ROM installieren. Auf der CD-ROM befindet sich der Ordner SOFTWARE. In diesem Ordner ist eine SETUP Anwendung. Zum Installieren der Software müssen Sie diese Setup-Anwendung starten.
2. Das Installationsprogramm meldet sich mit einem Dialogfeld 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 Warenzeichen der Microsoft Corp.  
VGA™ ist ein Warenzeichen der International Business Machines Corp.

## 2 Bedienung der SPECTRO2-Scope Software

Bitte lesen Sie diesen Abschnitt zuerst durch, bevor Sie die Einjustierung und Parametrisierung des Sensorsystems vornehmen.

Nach dem Aufruf der SPECTRO2-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.

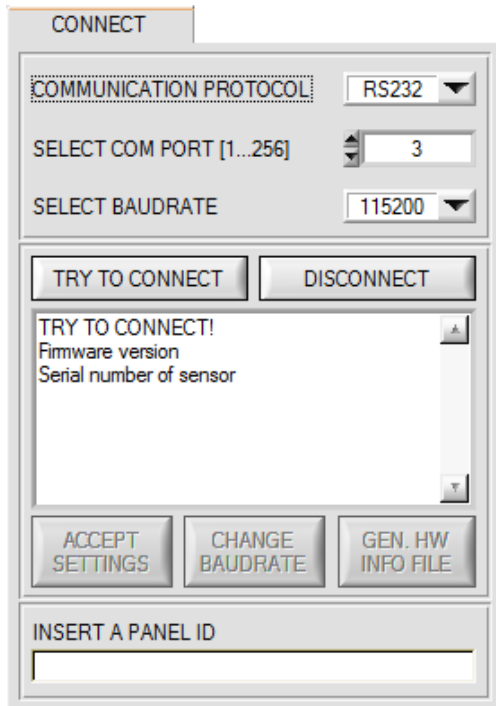
Ist ein Sensor angeschlossen und es kommt trotzdem zu keinem Verbindungsaufbau, dann stimmen entweder die SCOPE Version (Programm auf PC) und die Firmware Version (Programm im Sensor) nicht überein oder man muss die Schnittstelle zum Sensor richtig konfigurieren.

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



### 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 Anzeigedisplay 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>
---	--

COMMUNICATION PROTOCOL: RS232  
 SELECT COM PORT [1...256]: 3  
 SELECT BAUDRATE: 115200

TRY TO CONNECT    DISCONNECT

TRY TO CONNECT!  
 Firmware version  
 Serial number of sensor

ACCEPT SETTINGS    CHANGE BAUDRATE    GEN. HW INFO FILE

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.

COMMUNICATION PROTOCOL: RS232  
 SELECT COM PORT [1...256]: 3  
 SELECT BAUDRATE: 57600

TRY TO CONNECT    DISCONNECT

TRY TO CONNECT!  
 Firmware version  
 Serial number of sensor

ACCEPT SETTINGS    CHANGE BAUDRATE    GEN. HW INFO FILE

Unter **SELECT BAUDRATE** kann jetzt eine neue Baudrate ausgewählt werden. Durch Drücken von **CHANGE BAUDRATE** wird die neue Baudrate zum Sensor übertragen.

COMMUNICATION PROTOCOL: RS232  
 SELECT COM PORT [1...256]: 3  
 SELECT BAUDRATE: 57600

TRY TO CONNECT    DISCONNECT

TRY TO CONNECT!  
 Firmware version  
 Serial number of sensor

ATTENTION!!! CONFIRM NEW BAUDRATE WITH EEPROM AND SEND!

ACCEPT SETTINGS    CHANGE BAUDRATE    GEN. HW INFO FILE

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.

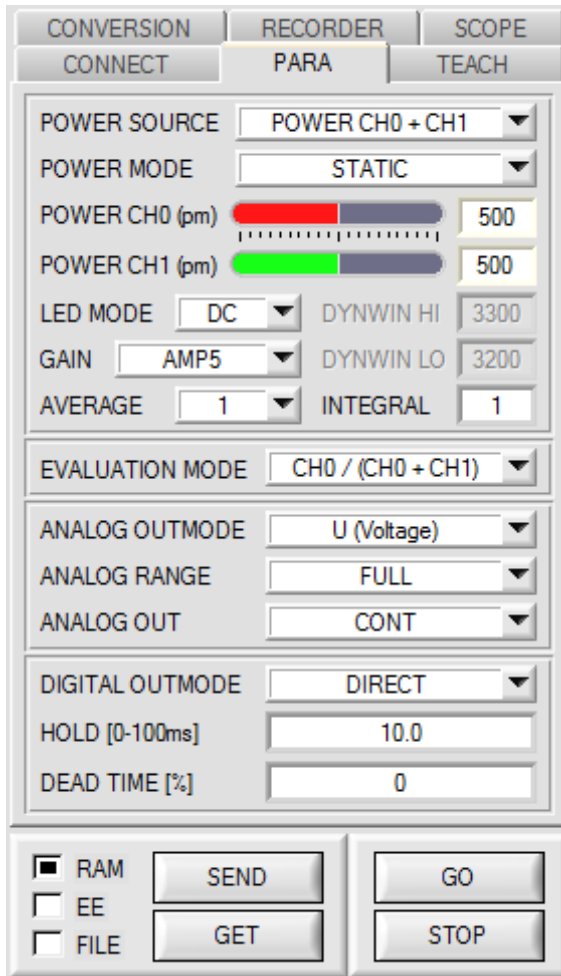
Durch Drücken von **ACCEPT SETTINGS** werden die aktuellen Schnittstellen-Einstellungen gespeichert und nach einem Neustart der Software automatisch eingestellt.

HARDWARE INI FILE HAS BEEN SAVED TO:  
 d:\Mist\HardwareInitialFile.ini

ACCEPT SETTINGS    CHANGE BAUDRATE    GEN. HW INFO FILE

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 PARA, Taste SEND, GET, GO, STOP



### PARA:

Durch Drücken von **PARA** 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:

Das **RAM** ist ein **flüchtiger** Speicher im Mikrokontroller des Sensors, d.h. nach Ausschalten der Spannung am Sensor gehen diese Parameter wieder verloren.

**Der Sensor arbeitet grundsätzlich mit den Parametern, die sich im RAM befinden.**

Ist der Auswahlknopf **RAM** gewählt, dann werden die aktuellen Parameter nach Drücken von **SEND** in den **RAM** Speicher des Sensors geschrieben bzw. nach Drücken von **GET** aus dessen **RAM** Speicher gelesen.

### EEPROM:

Das **EEPROM** ist ein **nichtflüchtiger** Speicher im Mikrokontroller des Sensors. Nach Ausschalten der Spannung am Sensor gehen die Parameter im **EEPROM** nicht verloren. Nach Wiedereinschalten der Spannung werden die Parameter aus dem **EEPROM** ins **RAM** geladen. Bildlich gesehen liegt das **EEPROM** also eine Stufe tiefer als das **RAM**. Der Datenaustausch zwischen **PC** und **EEPROM** erfolgt automatisch über das **RAM**. D.h. Parameter die ins **EEPROM** geschrieben werden, werden automatisch auch ins **RAM** geschrieben und Daten die aus dem **EEPROM** gelesen werden, werden automatisch auch ins **RAM** gelesen.

Ist der Auswahlknopf **EEPROM** gewählt, dann werden die aktuellen Parameter nach Drücken von **SEND** in den Speicher des nichtflüchtigen **EEPROMS** im Sensor geschrieben oder durch Drücken von **GET** aus dessen **EEPROM** gelesen.

Man sollte beim Parametrieren des Sensors mit dem **RAM** arbeiten. Wenn man eine geeignete Parametrierung für die jeweilige Anwendung gefunden hat, dann muss diese im **EEPROM** des Sensors hinterlegt werden, damit die Parameter bei einem Neustart des Sensors aus dem **EEPROM** ins **RAM** geladen werden können.

### 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.

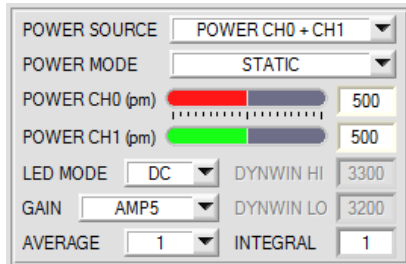
**TIPP!** Eine für eine bestimmte Anwendung gefundene Parametereinstellung sollte in jedem Fall in einem File auf dem PC hinterlegt werden.

### GO [F11]:

Anklicken dieser Taste startet den Datentransfer vom Sensor zum PC über die serielle RS232 Schnittstelle.

### STOP [F12]:

Anklicken dieser Taste beendet den Datentransfer vom Sensor zum PC über die serielle RS232 Schnittstelle.



**POWER SOURCE:**

In diesem Funktionsfeld wird eingegeben über wie viele Lichtquellen der Sensor verfügt.

**POWER CH0 + CH1** bedeutet, dass sowohl der **CH0** als auch **CH1** über eine eigene Sendequelle verfügen die bei **LED MODE = DC** dauernd an sind und im **AC** Betrieb gleichzeitig ein oder ausgeschaltet werden.

**POWER CH0** bedeutet, dass nur eine Lichtquelle am **CH0** Anschluss vorhanden ist. Dito **POWER CH1**.

Bei **ALTERNATING 1** und **ALTERNATING 2** werden die beiden Sendequellen für CH0 und CH1 nacheinander ein und ausgeschaltet.

CH0 und CH1 werden dabei wie folgt bestimmt:

**ALTERNATING 1: CH0(1) =**Sendequelle **CH0(1)** an (CH1(0) aus) – Sendequelle **CH0** und **CH1** aus

**ALTERNATING 2: CH0(1) =**Sendequelle **CH1(0)** an (CH0(1) aus) – Sendequelle **CH0** und **CH1** aus

**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. Außerdem ist ein externes Teachen mit **DYN** nicht möglich.

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

Zusätzlich kann für **GAIN AMP 1 2 3 4**, **AMP 5 6 7 8**, **AMP 1 3 5 7** und **AMP 2 4 6 8** eingestellt werden.

Dies ermöglicht es, mit den Eingänge IN0 und IN1 verschiedene Verstärkungsfaktoren am Empfänger einzustellen. IN0 und IN1 stehen dadurch für keine weiteren Optionen mehr zur Verfügung.

Abhängig von **GAIN** und dem Zustand der Eingänge IN0 und IN1 ergeben sich folgende Verstärkungsfaktoren.

IN1	IN0	AMP 1 2 3 4	AMP 5 6 7 8	AMP 1 3 5 7	AMP 2 4 6 8
0	0	1	5	1	2
0	1	2	6	3	4
1	0	3	7	5	6
1	1	4	8	7	8

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.



EVALUATION MODE CH0 / (CH0 + CH1) ▼

**EVALUATION MODE:**

Mit diesem Parameter wählt man aus, wie die beiden Kanäle **CH0** und **CH1** ausgewertet werden.

**CH0:** Als **EVALUATION SIGNAL** wird Kanal **CH0** direkt verwendet.

**CH1:** Als **EVALUATION SIGNAL** wird Kanal **CH1** direkt verwendet.

**CH0-CH1:** Das **EVALUATION SIGNAL** berechnet sich aus **CH0-CH1**.

**(CH0+CH1)/2:** Das **EVALUATION SIGNAL** berechnet sich aus **(CH0+CH1)/2**.

**CH0/(CH0+CH1):** Das **EVALUATION SIGNAL** berechnet sich aus **(CH0\*4095)/(CH0+CH1)**.

ANALOG OUTMODE U (Voltage) ▼  
 ANALOG RANGE FULL ▼  
 ANALOG OUT CONT ▼

Das **EVALUATION SIGNAL** wird zur Ausgabe eines Analogsignals herangezogen.

**Es** wird mit einer Auflösung von 12 Bit erfasst und kann daher Werte zwischen 0 und 4095 annehmen.

**ANALOG OUTMODE:**

In diesem Funktionsfeld kann eingestellt werden, wie der Analogausgang vom Sensor bedient wird.

**OFF:**

Es wird keine Analogsignal ausgegeben.

**U (Voltage):**

**EVALUATION SIGNAL** wird auf dem Analogausgang als Spannung von 0 bis 10V ausgegeben.

**I (Current):**

**EVALUATION SIGNAL** wird auf dem Analogausgang als Strom von 4 bis 20mA ausgegeben.

**ANALOG RANGE:**

In diesem Funktionsfeld kann eingestellt werden, wie und in welchem Bereich der Sensor das **EVALUATION SIGNAL** analog ausgibt.

**FULL:**

**EVALUATION SIGNAL** wird in seinem vollen Wertebereich von 0-4095 als Analogsignal ausgegeben.

**MIN-MAX while IN0:**

Solange der Eingang IN0 auf HI liegt, wird im Sensor ein maximaler und minimaler **EVALUATION SIGNAL** Wert ermittelt.

Das Analogsignal wird nach Abfall von IN0 innerhalb dieses **MIN-MAX** Bereiches voll (0-10V oder 4-20mA) ausgegeben.

**CONV TABLE:**

Dem **EVALUATION SIGNAL** kann in einer Tabelle ein bestimmter Konvertierungswert zugewiesen werden. Bei **CONV TABLE** wird dieser Wert analog ausgegeben. (Siehe Registerkarte CONVERSION).

**ANALOG OUT:**

In diesem Funktionsfeld kann eingestellt werden, wann der Sensor das Analogsignal ausgibt.

**CONT:** Das Analogsignal wird kontinuierlich ausgegeben.

**RISING EDGE of IN1:** Das Analogsignal wird nur bei einer positiven Flanke an IN1 ausgegeben.

**DIGITAL OUTMODE:**

Mit dieser Funktionstastengruppe wird die Ansteuerung der Digitalausgänge **OUT0** und **OUT1** bei **Verlassen** der Toleranzschwelle ausgewählt. Der Zustand von **OUT0** und **OUT1** ist wie folgt abhängig von **OUTMODE** und **THRESHOLD MODE**:

	<b>THRESHOLD MODE LOW und HI</b>	<b>THRESHOLD MODE WIN</b>
<b>OUTMODE OFF</b>	OUT0=0 VDC OUT1=0 VDC	OUT0=0 VDC OUT1=0 VDC
<b>OUTMODE DIRECT</b>	OUT0=0 VDC OUT1=0 VDC	OUT0=0 VDC OUT1=0 VDC wenn <b>SIG</b> kleiner Toleranzfenster OUT1=0+24 VDC wenn <b>SIG</b> größer Toleranzfenster
<b>OUTMODE INVERSE</b>	OUT0=+24 VDC OUT1=0 VDC	OUT0=+24 VDC OUT1=0 VDC wenn <b>SIG</b> größer Toleranzfenster OUT1=0+24 VDC wenn <b>SIG</b> kleiner Toleranzfenster

Der jeweilige Zustand der Ausgänge wird auf der Benutzeroberfläche über die LED **OUT0** und **OUT1** visualisiert.

**HOLD:** Der Sensor arbeitet mit minimalen Scanzeiten in der Größenordnung von kleiner 10µs. Durch Eingabe in die Edit Box **HOLD** wird eine Pulsverlängerung am Digitalausgang des Sensors bis zu 100 ms gewährleistet, dadurch kann eine SPS kurze Schaltzustandsänderungen sicher erkennen.



**DEAD TIME [%]:**

Mit diesem Funktionsfeld kann eine dynamische Totzeit aktiviert werden.

Am besten kann man die Totzeit anhand eines Beispiels erklären.

Der Sensor liefert aufgrund der Applikation ein regelmäßiges Schaltsignal von 100ms. Diese 100ms werden als 100% interpretiert. Ist eine **DEAD TIME** von 20% eingestellt, dann ist der Sensor nach jedem Schaltsignal 20% „tot“. Er ignoriert jede Störung die innerhalb von 20ms nach dem letztem Schalten auftritt und schaltet demnach nicht. Dadurch können z.B. unsaubere Kanten einwandfrei erkannt werden, ohne das der Sensor Mehrfachimpulse ausgibt.

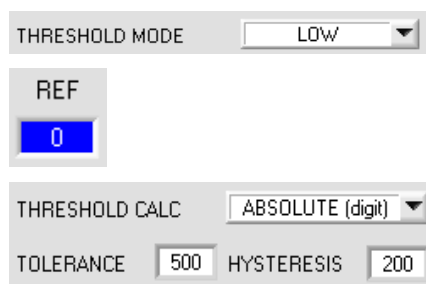
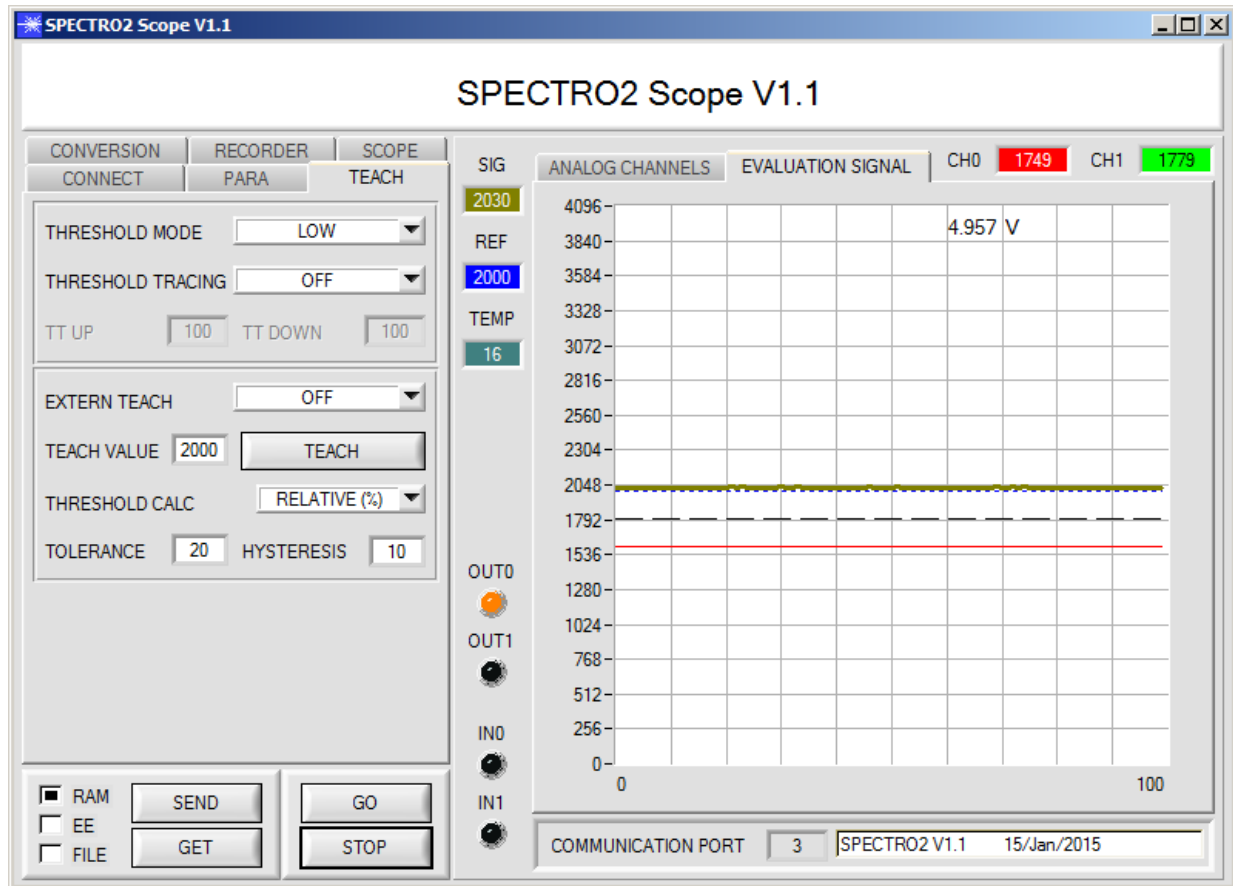
Ändert sich die Geschwindigkeit von 100ms auf 200ms, dann werden aus den 20ms aufgrund der prozentualen Berechnung 40ms. Deshalb spricht man von dynamischer Totzeit.

Eine Totzeit ist dann ratsam, wenn Mehrfachimpulse nicht mit einer geeigneten **HYSTERESIS** unterdrückt werden können.

**DEAD TIME [%] = 0** deaktiviert die Totzeit.

## 2.3 Registerkarte TEACH

Nach Drücken von **GO** beginnt eine Datenübertragung vom Sensor zum PC.  
 Das Analogsignal **SIG** wird in einem Graph und in einem Zahlendisplay zur Anzeige gebracht.  
 Die einfachste Methode, ein Analogsignal zu „teachen“ funktioniert über den **TEACH** Button.  
 Nach Anklicken von **TEACH** wird der aktuelle **SIG** Wert als **TEACH VALUE** übernommen.  
 Drückt man jetzt **SEND**, dann wird der **TEACH VALUE** zur neuen Referenz **REF**.  
 Ausgehend von **REF** werden dann die Schaltschwelle und die Hystereseschwelle gesetzt.



### THRESHOLD MODE:

In diesem Funktionsfeld kann eine der möglichen Positionen der Schaltschwelle und Hystereseschwelle in Bezug zum Referenzwert **REF** ausgewählt werden.

Der Referenzwert wird entweder über Software oder den externen Eingang **IN0** gelernt. Außerdem kann eine Schwellennachführung über **THRESHOLD TRACING** aktiviert werden. Dabei wird **REF** zyklisch neu ermittelt.

Ausgehend von **REF** und mit Hilfe von **TOLERANCE** und **HYSTERESIS** werden die Schaltschwelle und die Hystereseschwelle bei **THRESHOLD MODE LOW, HI** und **WIN** unterschiedlich berechnet.

Ist **THRESHOLD CALC = ABSOLUTE(digit)** gewählt, dann werden die Schwellen absolut in Digit zu **REF** berechnet. D.h. **TOLERANCE** und **HYSTERESIS** werden direkt auf **REF** aufsummiert oder davon abgezogen.

Ist **THRESHOLD CALC = RELATIVE(%)** gewählt, dann werden die Schwellen relativ zu **REF=100%** berechnet. D.h. **TOLERANCE** und **HYSTERESIS** werden relativ zu **REF** aufsummiert oder davon abgezogen.

Die verschiedenen Linien in den folgenden Grafikfenstern haben folgende Bedeutung

- Grüne Linie = aktueller Messwert SIG
- Blau gepunktete Linie = Referenzwert REF
- Rote Linie = Schaltschwelle
- Schwarz gestrichelte Linie = Hystereseschwelle

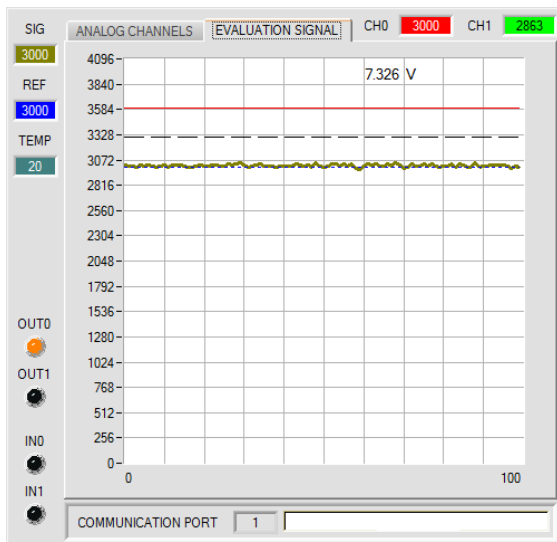


**THRESHOLD MODE = LOW:**

Schaltschwelle =  $REF - TOLERANCE$   
 Hystereseschwelle =  $REF - HYSTERESIS$

Falls der aktuelle Messwert **SIG** die Schaltschwelle unterschreitet, wird der digitale Ausgang **OUT0** auf Fehler gesetzt.

Wenn der aktuelle Messwert wieder größer wird als die Hystereseschwelle, dann wird der Fehlerausgang wieder zurückgenommen.

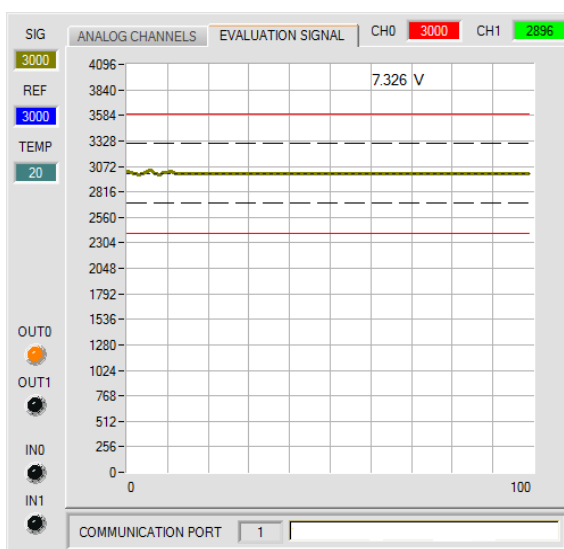


**THRESHOLD MODE = HI:**

Schaltschwelle =  $REF + TOLERANCE$   
 Hystereseschwelle =  $REF + HYSTERESIS$

Falls der aktuelle Messwert **SIG** die Schaltschwelle überschreitet, wird der digitale Ausgang **OUT0** auf Fehler gesetzt.

Wenn der aktuelle Messwert wieder kleiner wird als die Hystereseschwelle, dann wird der Fehlerausgang wieder zurückgenommen.



**THRESHOLD MODE = WIN:**

Obere Schaltschwelle =  $REF + TOLERANCE$   
 Obere Hystereseschwelle =  $REF + HYSTERESIS$

Untere Schaltschwelle =  $REF - TOLERANCE$   
 Untere Hystereseschwelle =  $REF - HYSTERESIS$

Die Schaltschwellen bilden ein symmetrisches Toleranzband um den aktuellen Referenzwert. Falls der aktuelle Messwert **SIG** dieses Toleranzband verlässt, wird der digitale Ausgang **OUT0** auf Fehler gesetzt.

Der Fehlerausgang wird wieder zurückgenommen, wenn der aktuelle Messwert wieder unter bzw. über die Hystereseschwelle kommt.

Der Ausgang **OUT1** wird abhängig von **OUTMODE** gesetzt bzw. rückgesetzt, wenn das Signal das Toleranzfenster nach oben bzw. nach unten verlässt.

THRESHOLD TRACING

TT UP  TT DOWN

---

THRESHOLD CALC

TOLERANCE  HYSTERESIS

**THRESHOLD TRACING:**

Mit Hilfe dieses Funktionsfeldes kann eine automatische Schwellennachführung aktiviert werden. Eigentlich wird der Referenzwert **REF** einem sich ändernden **SIG** Wert zyklisch nachgeführt. Ausgehend von **REF** werden dann die Schaltschwelle und die Hystereseschwelle gesetzt. Dies wird im Allgemeinen als Schwellennachführung bezeichnet.

**OFF:** Die automatische Schwellennachführung ist inaktiv.

**ON TOL und ON CONT:** Die automatische Schwellennachführung ist aktiv. Der aktuelle Referenzwert **REF** wird zyklisch nachgeführt falls sich der aktuelle **SIG** Wert z.B. aufgrund einer zunehmenden Verschmutzung verringert. Bei **ON TOL** erfolgt die Nachführung, wenn das aktuelle Signal innerhalb der Toleranz liegt. Bei **ON CONT** wird immer nachgeführt.

**TT UP und TT DOWN:** In diesem Funktionsfeld kann eine Zeitkonstante für die Geschwindigkeit der Schwellennachführung eingestellt werden.

**TT UP (THRESHOLD TRACING UP):** Falls der aktuelle **SIG** Wert zunimmt, wird **REF** mit der eingestellten Verzögerung um ein Digit erhöht.

**TT DOWN (THRESHOLD TRACING DOWN):** Falls der aktuelle **SIG** Wert abnimmt, wird **REF** mit der eingestellten Verzögerung um ein Digit erniedrigt.

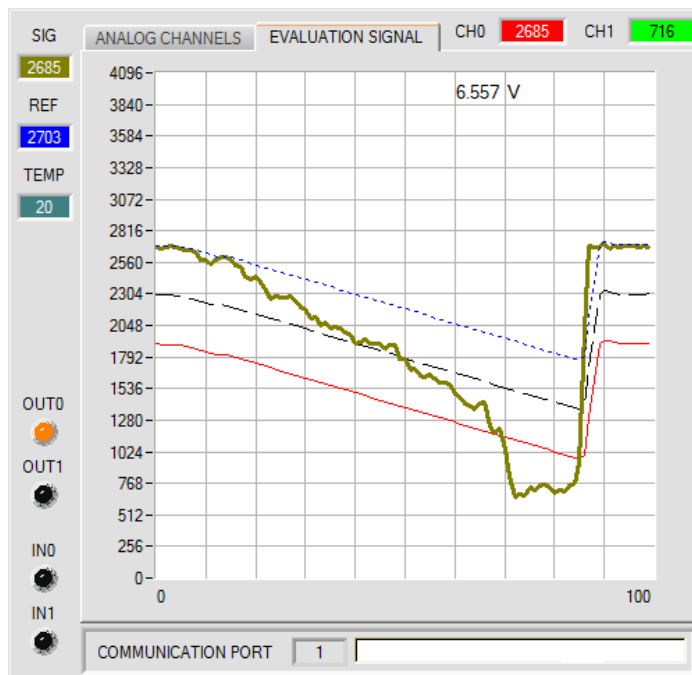
Für **TT UP** und **TT DOWN** kann ein Wert zwischen 0 und 60000 gewählt werden.

Ein Schritt bedeutet 100 Mikrosekunden Verzögerung. Die Nachführung kann jedoch nicht schneller sein als die durch **AVERAGE** erreichbare Scanfrequenz.

Wert 0: minimale Zeitverzögerung, schnellste Nachführung.

Wert 60000: maximale Zeitverzögerung, langsamste Nachführung.

Die Berechnung der Schaltschwellen hängt von **THRESHOLD MODE** ab (siehe **THRESHOLD MODE**).



Im Beispiel wurde für **TT DOWN** ein Wert von 50 gewählt. Das bedeutet, dass die Schwellennachführung nach unten relativ langsam durchgeführt wird. In der Grafik kann man das an der blauen Referenzlinie deutlich sehen.

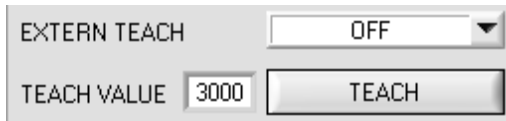
Der Messwert **SIG** fällt relativ schnell ab, der Referenzwert **REF** folgt im Vergleich dazu viel langsamer. Dies ist der typische Fall, wenn eine zunehmende Verschmutzung des Sensors kompensiert werden muss.

Die Schwellennachführung wird bei **ON TOL** gestoppt, wenn der aktuelle Messwert unter die Schaltschwelle fällt. Wenn **SIG** wieder größer wird als die Hystereseschwelle, dann wird die Schwellennachführung wieder aktiv. Im Beispiel wird die Schwelle nach oben schnell nachgeführt, da für **TT UP** ein kleiner Wert gewählt wurde.



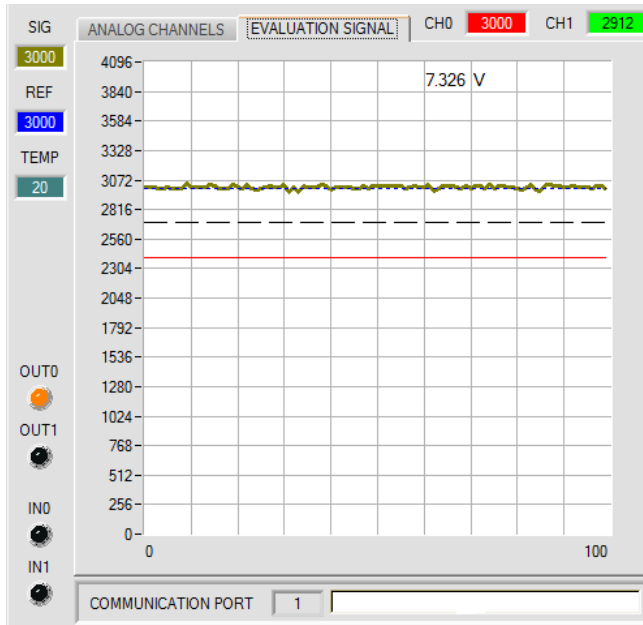
**EXTERN TEACH:**

Mit Hilfe dieses Funktionsfeldes wird ausgewählt, wie man den Sensor „teachen“ kann.



**EXTERN TEACH = OFF:**

Die einfachste Methode, ein Analogsignal zu „teachen“ funktioniert über den **TEACH** Button.

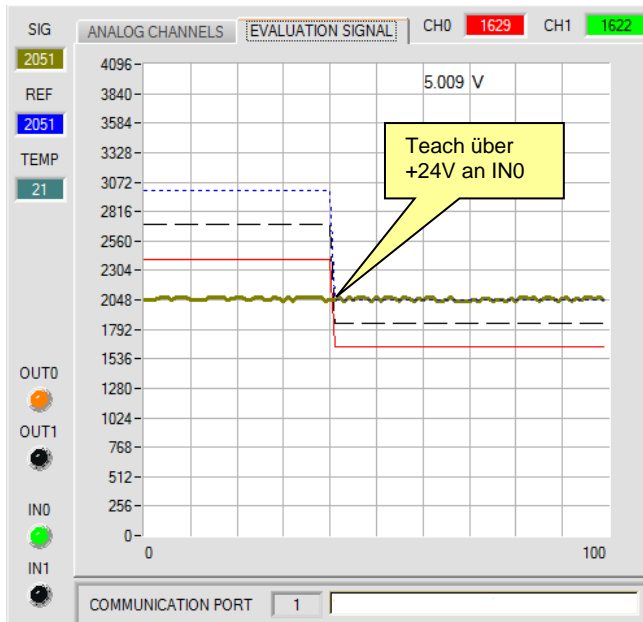


Nach Anklicken von **TEACH** wird der aktuelle **SIG** Wert als **TEACH VALUE** übernommen.

Drückt man jetzt **SEND**, dann wird der **TEACH VALUE** zur neuen Referenz **REF**.

Ausgehend von **REF** werden dann die Schaltschwelle und die Hystereseschwelle gesetzt.

Der **TEACH** Button ist nicht aktiv, wenn entweder **EXTERN TEACH** oder **THRESHOLD TRACING** ungleich **OFF** sind.

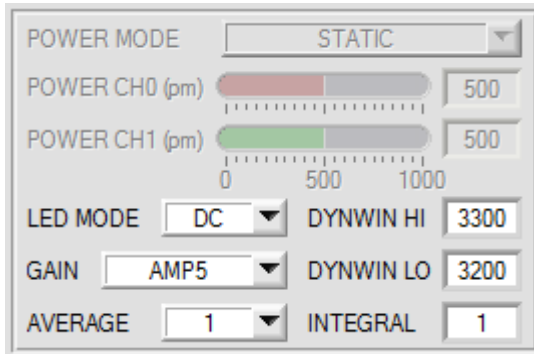


**EXTERN TEACH = DIRECT:**

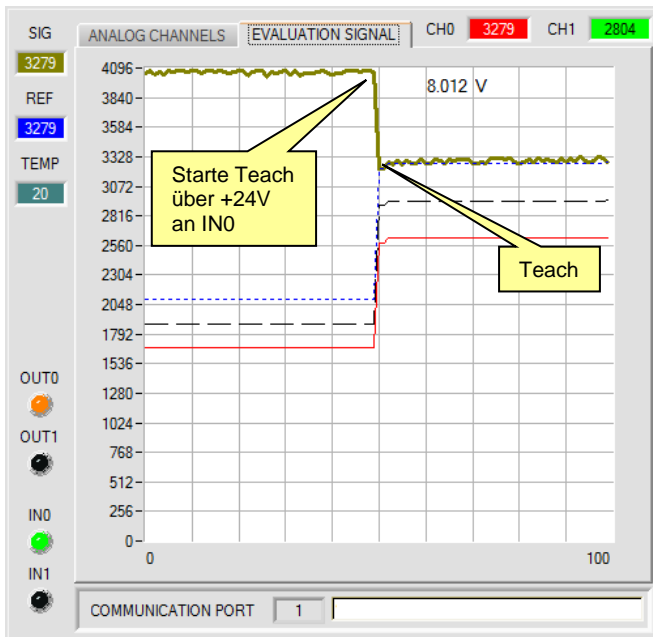
Durch eine positive Flanke am Eingang **IN0** wird der aktuelle **SIG** Wert zur neuen Referenz **REF**.

Ausgehend von **REF** werden dann die Schaltschwelle und die Hystereseschwelle gesetzt.

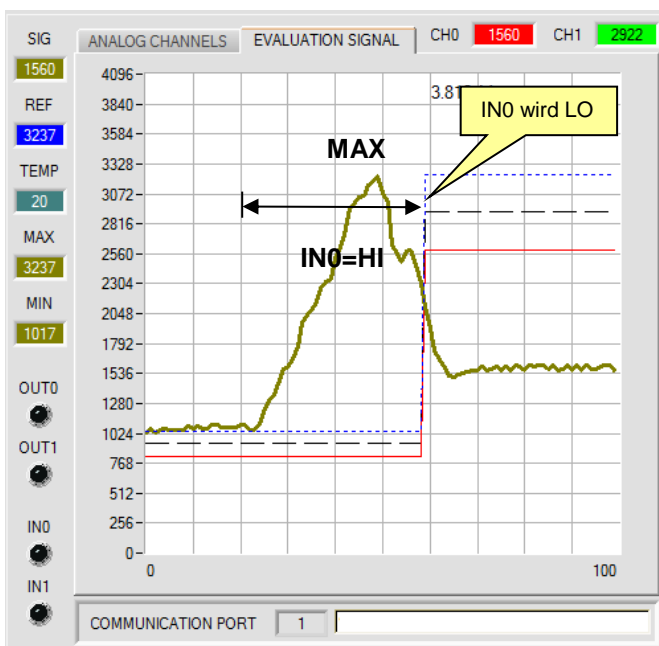
Die neue Referenz **REF** wird nur im **RAM** und nicht im **EEPROM** des Sensors hinterlegt oder **THRESHOLD TRACING** ungleich **OFF** sind.



**EXTERN TEACH = DYN:**  
**POWER MODE** wird automatisch auf **STATIC** eingestellt und ist bei **EXTERN TEACH = DYN** genauso wie **POWER** inaktiv.



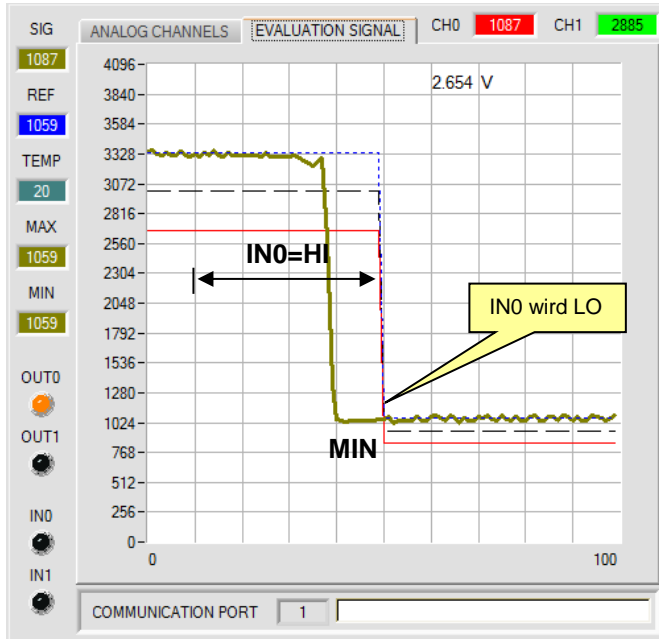
**EXTERN TEACH = DYN:**  
**POWER MODE** wird automatisch auf **STATIC** eingestellt und ist bei **EXTERN TEACH = DYN** genauso wie **POWER** inaktiv.



**EXTERN TEACH = MAX:**  
 Während der Eingang **INO=HI (+24V)** ist, wird der maximale **SIG** Wert ermittelt, der nach Abfall von **INO** zur neuen Referenz **REF** wird.

Ausgehend von **REF** werden dann die Schaltschwelle und die Hystereseschwelle gesetzt.

Die neue Referenz **REF** wird nur im **RAM** und nicht im **EEPROM** des Sensors hinterlegt.

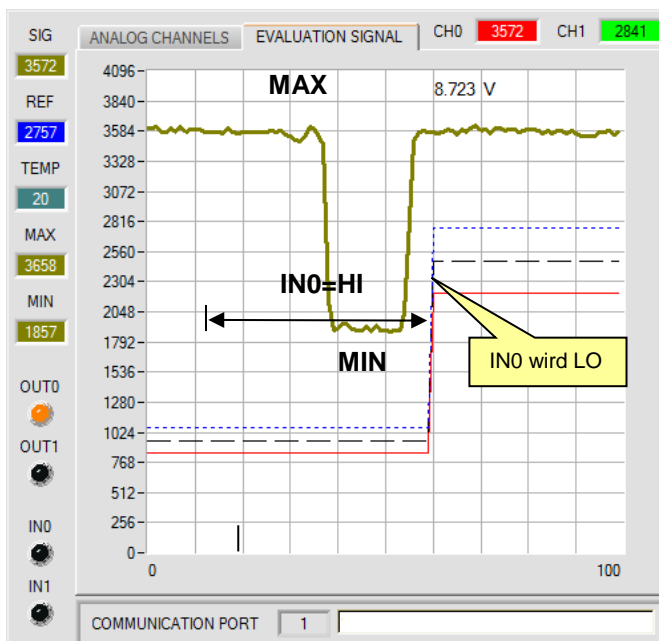


**EXTERN TEACH = MIN:**

Während der Eingang IN0=HI (+24V) ist, wird der minimale **SIG** Wert ermittelt, der nach Abfall von IN0 zur neuen Referenz **REF** wird.

Ausgehend von **REF** werden dann die Schaltschwelle und die Hystereseschwelle gesetzt.

Die neue Referenz **REF** wird nur im **RAM** und nicht im **EEPROM** des Sensors hinterlegt.



**EXTERN TEACH = (MAX+MIN)/2:**

Während der Eingang IN0=HI (+24V) ist, wird der maximale und minimale **SIG** Wert ermittelt. Nach Abfall von IN0 wird die neuen Referenz **REF** genau zwischen **MAX** und **MIN** gelegt.

Ausgehend von **REF** werden dann die Schaltschwelle und die Hystereseschwelle gesetzt.

Die neue Referenz **REF** wird nur im **RAM** und nicht im **EEPROM** des Sensors hinterlegt.



## 2.4 Graphische Anzeigeelemente

Zur Visualisierung aller zum Parametrieren wichtigen Daten stehen verschiedene Displays sowie ein graphisches Fenster zur Verfügung. Die Bedeutung der einzelnen Displays und des Graphen wird nachfolgend erklärt.



### CH0 und CH1:

In diesen Displays wird das Analogsignal der einzelnen Kanäle angezeigt. Im Graphen unter der Registerkarte **ANALOG CHANNELS** werden die Werte als Linie angezeigt.



### SIG:

In diesem Display wird der aus **CH0 und CH1** ermittelte Messwert zur Anzeige gebracht. Im Graphen unter dem Reiter **EVALUATION SIGNAL** wird **SIG** als Linie visualisiert.



### REF:

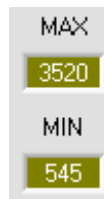
Hier wird der aktuelle Referenzwert angezeigt. Dieser Wert ist Basis zur Berechnung der Schaltschwelle und Hystereseschwelle.

Ist **THRESHOLD TRACING** und **EXERN TEACH** gleich **OFF** eingestellt, dann ist der **TEACH VALUE** der Referenzwert.

Ist **THRESHOLD TRACING** gleich **ON** eingestellt, dann ist die automatische Schwellennachführung aktiv. Der aktuelle Referenzwert **REF** wird zyklisch nachgeführt falls sich der aktuelle **SIG** Wert z.B. aufgrund einer zunehmenden Verschmutzung verringert.

Außerdem kann abhängig von **EXERN TEACH** die Referenz **REF** über den Eingang IN0 auf verschiedene Methoden eingelernt werden.

Der Referenzwert **REF** ist im Graphen als blaue Linie dargestellt.



### MAX und MIN:

Ist bei **EXTERN TEACH** oder **ANALOG RANGE** die Suche eines minimalen und oder maximalen **SIG** Wertes erforderlich, dann wird dieser Wert bzw. diese Werte in diesen Displays angezeigt.

Die beiden Displays werden nur angezeigt, wenn sie gebraucht werden.



### TEMP:

In diesem Display wird die im Sensorgehäuse herrschende Temperatur angezeigt.

Die Anzeige entspricht **NICHT** Grad Celsius oder Fahrenheit.



### OUT0 und OUT1:

Die LED's visualisieren den physikalischen Zustand der Ausgänge OUT0 und OUT1.

Ist die LED schwarz, dann liegt der Ausgang auf 0V.

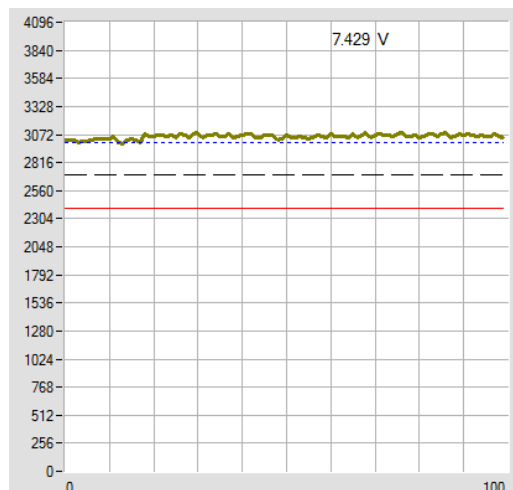
Ist die LED orange, dann liegt der Ausgang auf +24V

### IN0 und IN1:

Die LED's visualisieren den physikalischen Zustand der Eingänge IN0 und IN1.

Ist die LED schwarz, dann liegt der Eingang auf 0V.

Ist die LED grün, dann liegt der Eingang auf +24V



### GRAPH:

Im graphischen Anzeigefenster werden die Schaltschwelle (rote Linie), die Hystereseschwelle (schwarz gestrichelte Linie), die Referenz **REF** (blau gestrichelte Linie) sowie das aktuelle Analogsignal **SIG** abhängig von den eingestellten Parametern zur Anzeige gebracht.

Im nebenstehenden Graphen waren die Parametereinstellungen wie folgt:

**THRESHOLD MODE = LOW, THRESHOLD TRACING = OFF**

**EXTERN TEACH = OFF**

**TEACH VALUE = 3000 (→ REF=3000)**

**THRESHOLD CALC = RELATIVE (%)**

**TOLERANCE = 20**

**HYSTERESIS = 10**

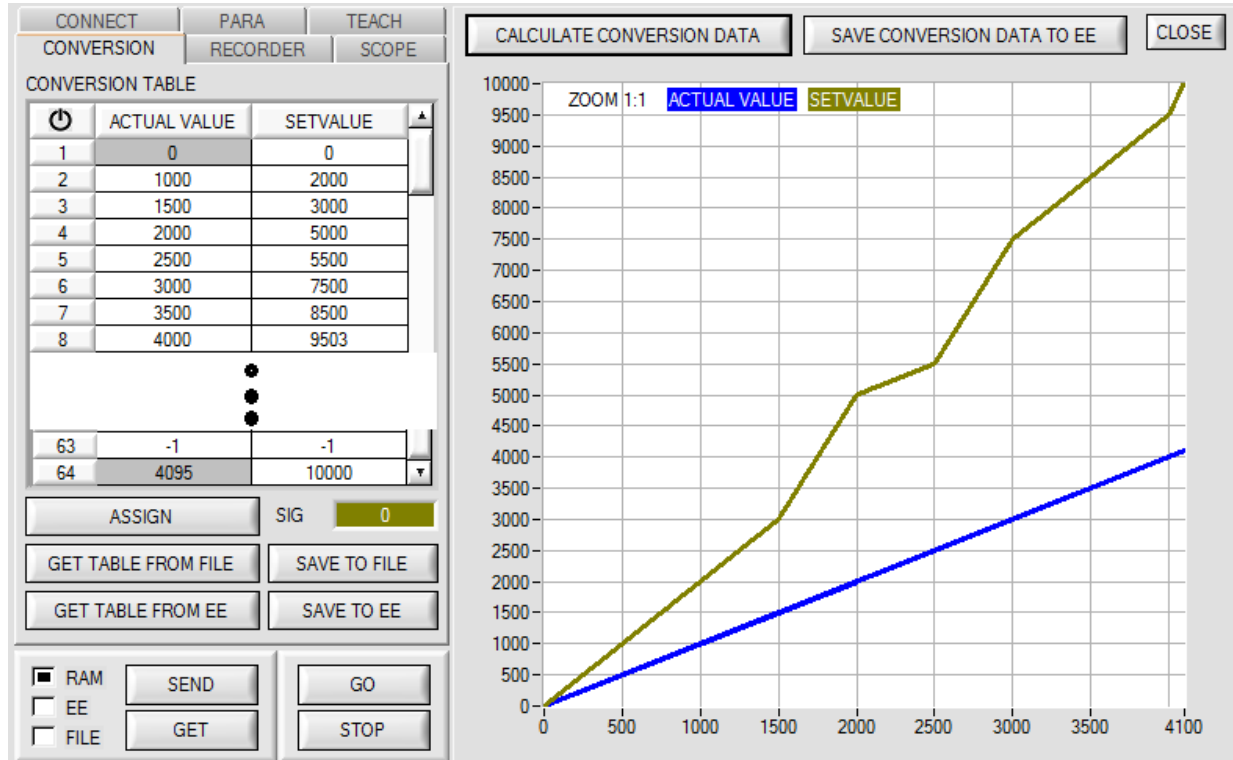
Daraus ergibt sich eine Schaltschwelle von **TEACH VALUE** minus

**TOLERANCE** (rote Linie), sowie eine Hystereseschwelle von

**TEACH VALUE** minus **HYSTERESIS** (schwarz gestrichelte Linie).

## 2.5 Registerkarte CONVERSION

In der Registerkarte **CONVERSION** kann man **SIG** einen bestimmten Konvertierungswert zuordnen. Ist **ANALOG RANGE = CONV TABLE** eingestellt, dann wird anstelle von **SIG** der dazugehörige Konvertierungswert ausgegeben.



Durch Drücken von **GO** wird der aktuelle **SIG** Werte im Display **SIG** angezeigt.

Selektiert man nur mit der Maus eine Zelle in der Spalte **ACTUAL VALUE** und drückt **ASSIGN**, dann wird dieser Wert in die Tabelle übernommen. In der Spalte **SETVALUE** gibt man nun den Wert an, der analog ausgegeben werden soll.

Insgesamt kann man 64 Konvertierungswerte eingeben.

Die erste und die letzte Zeile müssen ausgefüllt sein (siehe **CONVERSION TABLE** Zeile 1 und 64).

Nicht benutzte Zeilen müssen mit -1 „disabled“ werden (siehe **CONVERSION TABLE** Zeile 63).

In der Spalte **ACTUAL VALUE** können Werte zwischen 0 und 4095 eingegeben werden.

In der Spalte **SETVALUE** geht der Wertebereich von 0 bis 10000.

10000 wurde deshalb gewählt, weil der analoge Spannungsausgang von 0 bis 10 V (10000mV) geht.

Möchte man z.B. bei einem **SIG** Wert von 2500 eine analoge Ausgangsspannung von 5,5V, gibt man in der entsprechenden Zelle in der Spalte **SETVALUE** den Wert 5500 ein (siehe **CONVERSION TABLE** Zeile 5).

Durch Drücken von **CALCULATE CONVERSION DATA** kann man sich den Verlauf der Konvertierungstabelle ansehen. Die blaue Linie zeigt dabei den Istwert und die grüne den Sollwert.

Man kann in den Graphen hinein zoomen, indem man die Steuerungstaste gedrückt hält und mit der linken Maustaste ein Fenster aufzieht. Durch Drücken von **ZOOM 1:1** wird der Graph wieder wie gewohnt angezeigt.

Mit **SAVE CONVERSION DATA TO EE** sendet man die Konvertierungsdaten in EEPROM des Sensors.

Außerdem hat man die Möglichkeit mit **GET TABLE FROM FILE** und **SAVE TO FILE** die Konvertierungstabelle in eine File zu speichern, bzw. von dort zu lesen.

Mit **GET TABLE FROM EE** und **SAVE TO EE** wird die Konvertierungstabelle im EEPROM des Sensors gespeichert, bzw. von dort geladen.

Nach Drücken auf das Symbol  wird die Tabelle zurückgesetzt.

## 2.6 Registerkarte RECORDER

Die SPECTRO2-Scope Software beinhaltet einen Datenrekorder, welcher es erlaubt **CH0**, **CH1**, **SIG** und **TEMP** aufzuzeichnen. Das aufgezeichnete File wird auf der Festplatte des PC abgespeichert und kann anschließend mit einem Tabellenkalkulationsprogramm ausgewertet werden.

Das erzeugte File hat vier Spalten und so viele Zeilen, wie Datenframes aufgezeichnet worden sind. Eine Zeile ist wie folgt aufgebaut: **Datum, Uhrzeit, CH0, CH1, SIG, TEMP**.

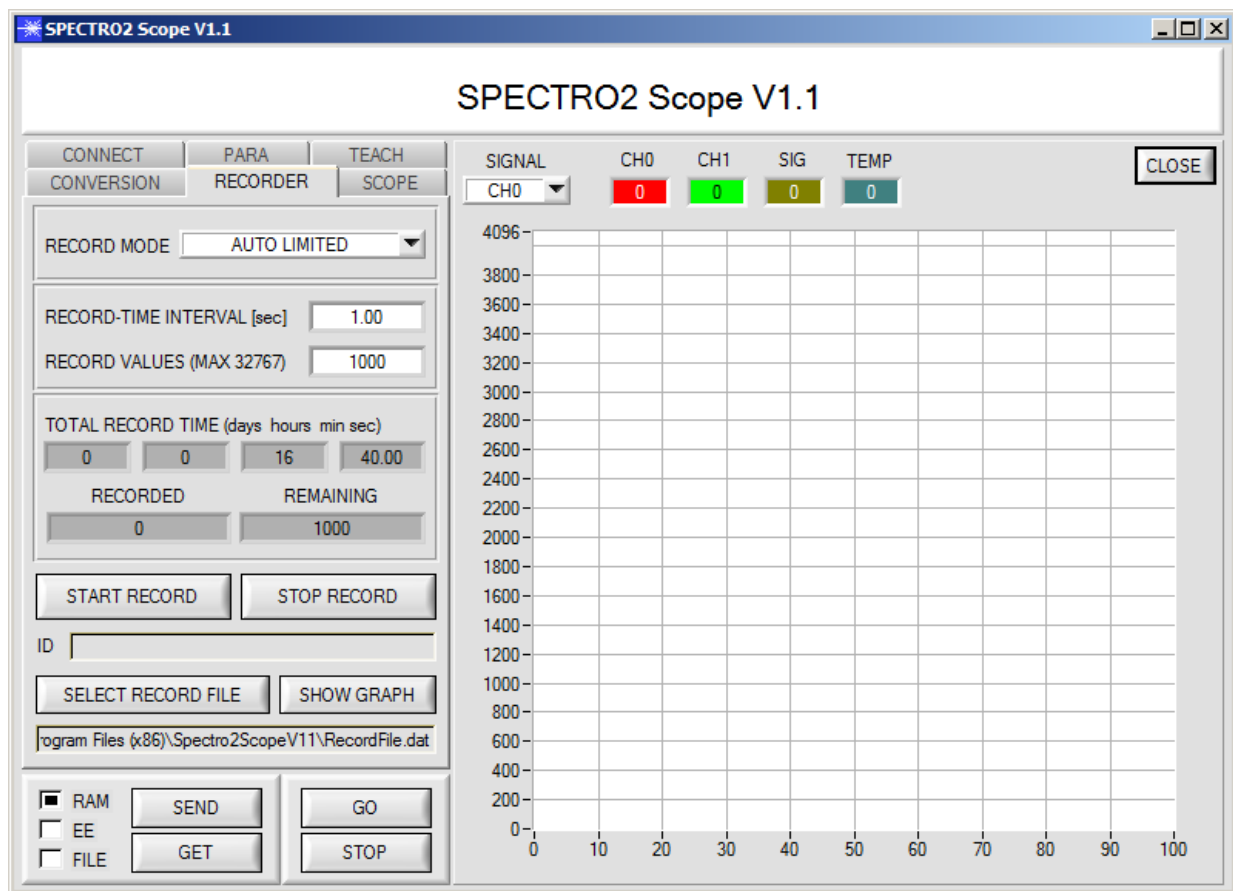
Führen Sie folgende Schritte durch, um Datenframes mit dem Recorder aufzuzeichnen:

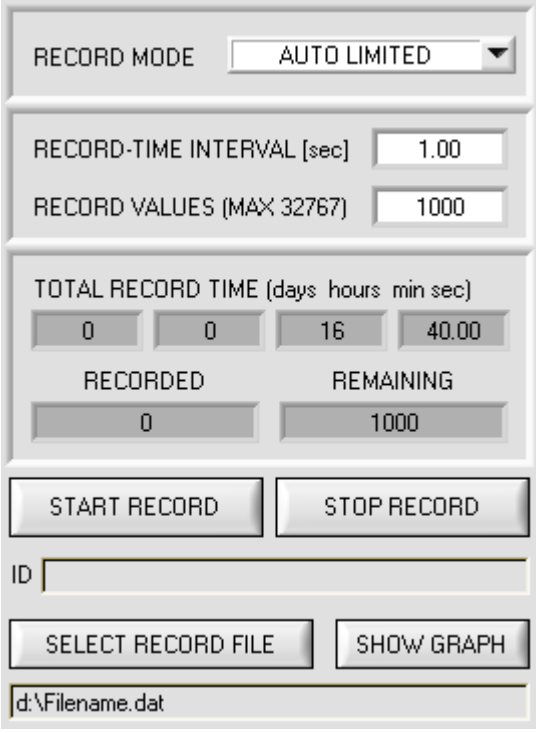
### 1. Schritt:

Nach Drücken von **RECORDER** 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.





### 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 File-Namen 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:

**Sobald man mit SELECT RECORD FILE ein File ausgewählt hat, wird dieses File neu erzeugt. Anschließend wird ein File Header über die Bedeutung der einzelnen Spalten im File hinterlegt.**

**Werden jetzt Daten aufgezeichnet, dann werden diese an das ausgewählte File angehängt, auch wenn die Datenaufzeichnung gestoppt wird und anschließend wieder aufgenommen wird.**

## 2.7 Registerkarte SCOPE

In der Registerkarte SCOPE wurde ein Oszilloskop nachgebildet.

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.

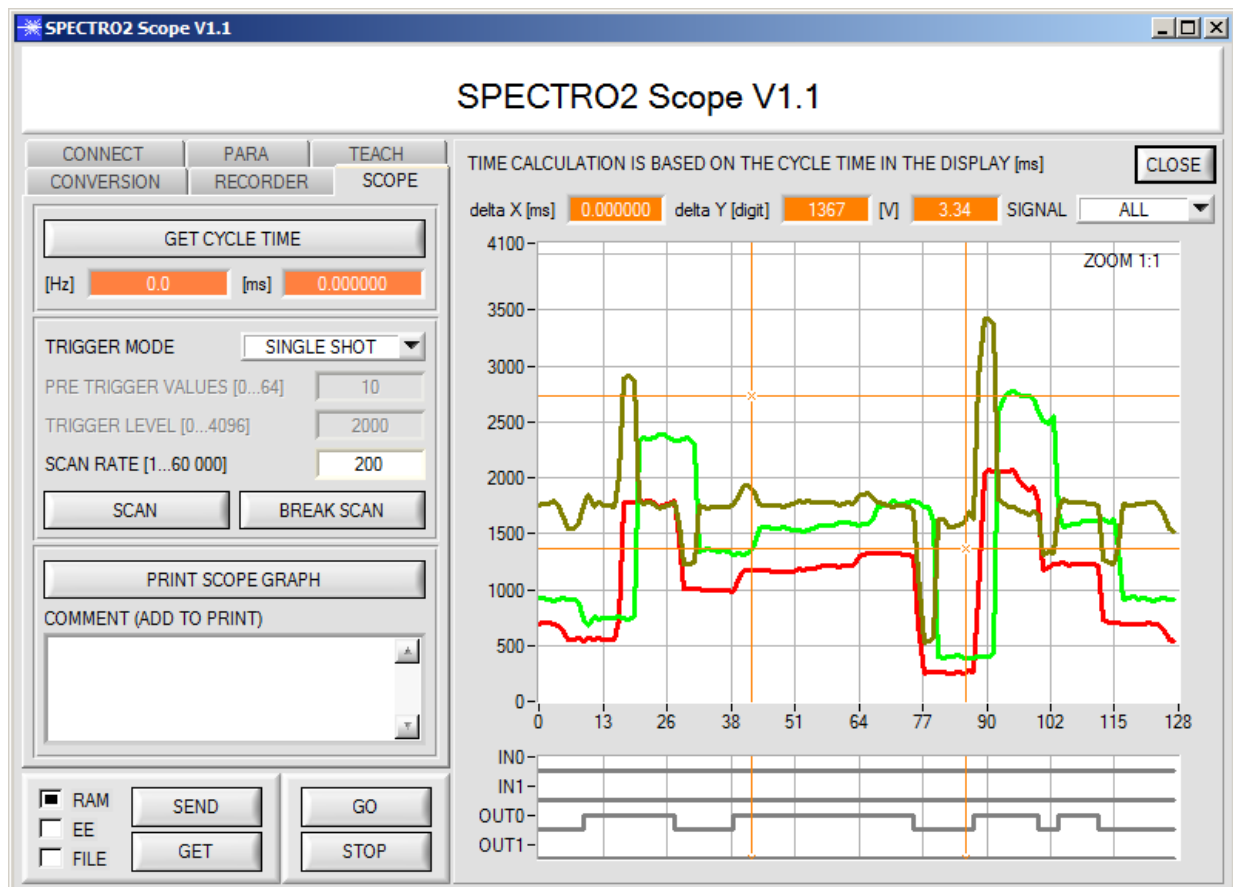
Im **TRIGGER MODE = SINGLE SHOT** wird nach Drücken von **SCAN** ein Datenframe aufgezeichnet und im Graphen zur Anzeige gebracht.

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

Im **TRIGGER MODE = INTERN OUT0** startet die Aufzeichnung selbständig, sobald OUT0 aus seiner Toleranz fällt.

Über **TRIGGER 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 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]**, **deltaY[digit]** und **deltaY[V]** aktualisiert.

**deltaX[ms]** zeigt die Zeit zwischen den Cursor in X-Richtung.

**deltaY[digit]** bzw. **[V]** zeigt die Differenz der beiden Cursor in Y-Richtung in Digit bzw. in Volt.

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

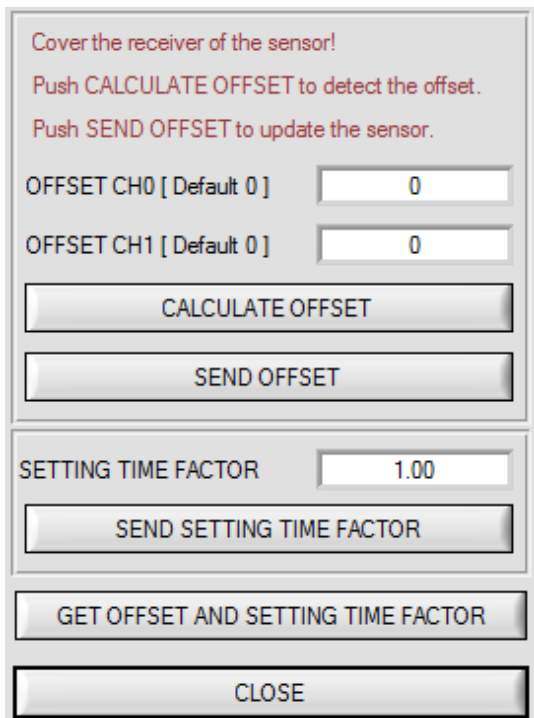
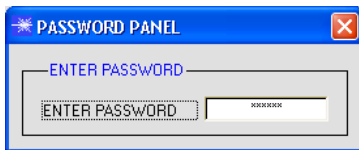
Im unteren Graphen werden die Zustände der beiden Ausgänge **OUT0** und **OUT1** sowie der Eingänge **IN0** und **IN1** dargestellt.

## 2.8 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.



z.B. hier:  
 Doppelklick mit der rechten Maustaste.



Um zur Offsetkalibrierung zu gelangen, muss in der Registerkarte PARA ein Doppelklick mit der rechten Maustaste exakt auf den Rahmen unterhalb INTEGRAL durchgeführt werden.

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

Jetzt muss 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**. Der Offsetwert sollte auf alle Fälle deutlich unter 100 liegen. Er wird mit **SEND OFFSET** im EEPROM des Sensors hinterlegt.

Über **GET OFFSET AND SETTING TIME FACTOR** kann man kontrollieren, welcher Wert als Offsetwert abgespeichert ist. Über **SEND OFFSET** kann man gegebenenfalls selber einen Offsetwert abspeichern. (! nicht empfohlen).

**SETTING TIME FACTOR** ist ein Wert, der bei der Fertigung des Sensors fest vergeben wird. Er beträgt bei Sensoren mit einer Infrarot Lichtquelle 2,5. Bei allen anderen sollte dieser Wert 1,0 sein. Eine Änderung dieses Wertes ist zwar möglich wird aber in diesem Manual aus Sicherheitsgründen nicht beschrieben. Sollte eine Änderung dennoch erforderlich sein, muss man sich mit seinem Lieferanten in Verbindung setzen.

### 3 Bedienung der TEMPCOMP-Scope Software

Sollte bei einem Firmware-Update 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.


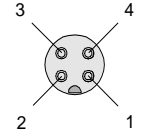


1. Schritt: Laden Sie jetzt über **GET EQUATION** oder **LOAD DATA FILE** das Temperaturkompensations-File, welches Sie von Ihrem Lieferanten erhalten haben.
2. Schritt: Drücken Sie **CALCULATE CURVES**, um die Daten im Graphen anzuzeigen.
3. Schritt: Wählen Sie die sensorinterne Betriebstemperatur (nicht in °C), welche der Sensor bei einer Umgebungstemperatur von 20° hat, falls dies 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.

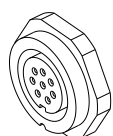
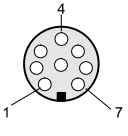
Anmerkung! Wenn Sie das Temperaturkompensations-File nicht gleich zur Hand haben, dann starten Sie einfach die TEMPCOMP-Scope Software. Bauen Sie eine Verbindung auf, soweit noch nicht vorhanden, und drücken Sie einfach **SEND-CF**. Der Sensor funktioniert jetzt wie gehabt, ist jedoch nicht temperaturkompensiert.

## 4 Anschlussbelegung von SPECTRO-2 Sensoren

### Anschluss SPECTRO-2 an PC:

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

### Anschluss SPECTRO-2 an SPS:

<b>8-pol. Buchse (Typ Binder 712)</b> <b>SPECTRO-2/SPS</b>		 
Pin-Nr.:	Litzenfarbe: (cab-las8/SPS-fem)	Belegung:
1	weiß	0V (GND)
2	braun	+24V (± 10 %)
3	grün	IN0 (Digital 0: 0 ... 1V, Digital 1: +Ub – 10%)
4	gelb	IN1 (Digital 0: 0 ... 1V, Digital 1: +Ub – 10%)
5	grau	OUT0 (Digital 0: 0 ... 1V, Digital 1: +Ub – 10%)
6	rosa	OUT1 (Digital 0: 0 ... 1V, Digital 1: +Ub – 10%)
7	blau	ANALOG (0...10V bzw. 4...20mA)
8	rot	N.C.



## 5 RS232 Schnittstellenprotokoll

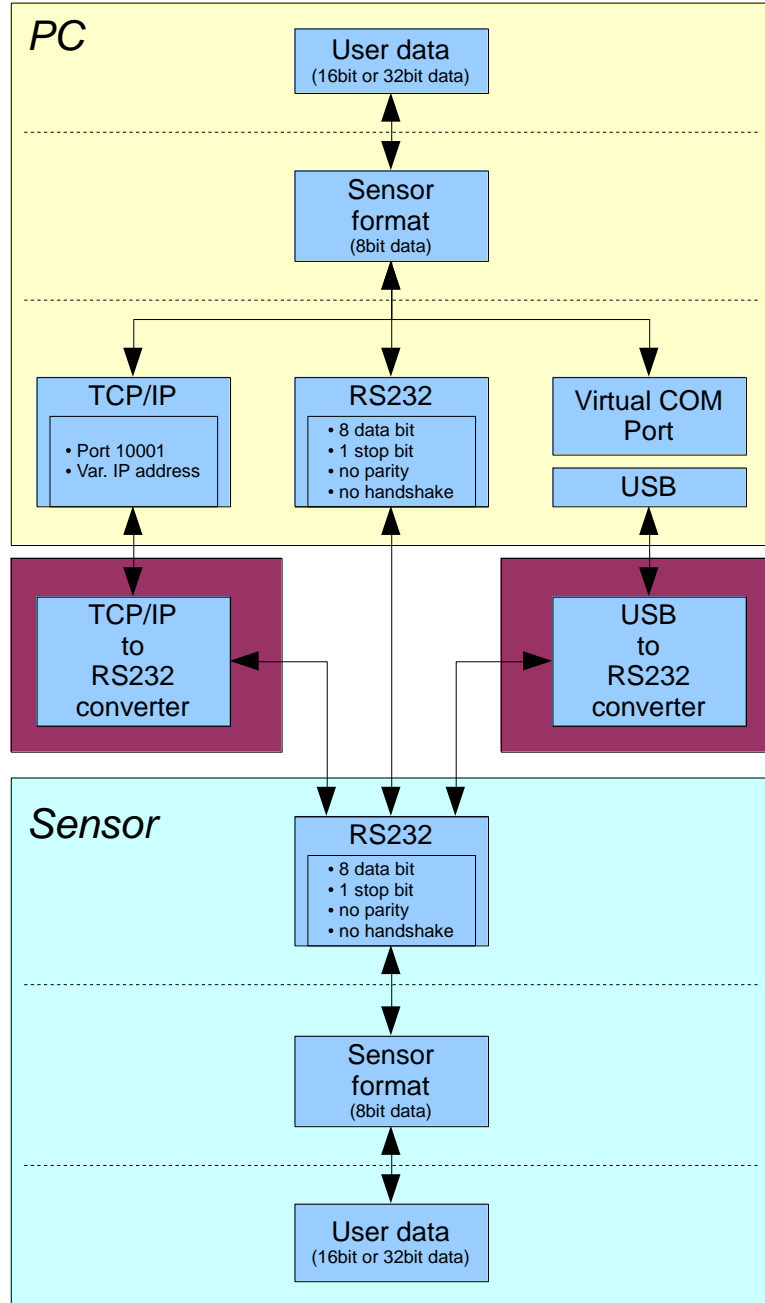
Sensoren der SPECTRO-2 Serie arbeiten mit folgenden **Parametern**, die in der angegebenen Reihenfolge zum Sensor übertragen werden bzw. davon ausgelesen werden.  
 Info! 2 Bytes (8bit) sind ein Word (16bit).

TABLE PARAMETER		
Parameter	Type	Meaning
<b>Para1:</b> POWER SOURCE	Word	Power source CH0+CH1, ALTERNATING1, ALTERNATING2, CH0, CH1, OFF coded to (0, 1, 2, 3, 4, 5)
<b>Para2:</b> POWER CH0	Word	Transmitter intensity channel 0 (0...1000) Attention intensity in thousandth!
<b>Para3:</b> POWER CH1	Word	Transmitter intensity channel 1 (0...1000) Attention intensity in thousandth!
<b>Para4:</b> POWER MODE	Word	Transmitter mode: STATIC, DYNAMIC coded to (0, 1)
<b>Para5:</b> DYNWIN LO	Word	Low limit for dynamic window when POWER MODE=dynamic (0...4095)
<b>Para6:</b> DYNWIN HI	Word	High limit for dynamic window when POWER MODE=dynamic (0...4095)
<b>Para7:</b> LED MODE	Word	Control for the internal light source DC, AC coded to (0,1)
<b>Para8:</b> GAIN	Word	Amplification of the integrated receiver AMP1, AMP2, AMP3, AMP4, AMP5, AMP6, AMP7, AMP8, AMP1234, AMP5678, AMP1357, AMP2468 coded to (1, 2, 3, 4, 5, 6, 7, 8,9,10,11,12)
<b>Para9:</b> AVERAGE	Word	Signal averaging 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384 or 32768
<b>Para10:</b> INTEGRAL	Word	Signal integration (1...250)
<b>Para11:</b> ANALOG OUTMODE	Word	Function of the analog output: OFF, U, I, U+I coded to (0,1,2,3)
<b>Para12:</b> ANALOG RANGE	Word	Function of the analog range: FULL, MIN-MAX, when IN0, CONV TABLE coded to (0,1,2)
<b>Para13:</b> ANALOG OUT	Word	Function of the analog out: CONT, RISING EDGE of IN1 coded to (0,1)
<b>Para14:</b> DIGITAL OUTMODE	Word	Function of the digital output: OFF, DIRECT, INVERSE coded to (0,1,2)
<b>Para15:</b> HOLD	Word	Hold time for minimum pulse length coded to (0...100 [ms]) send value*10
<b>Para16:</b> THRESHOLD MODE	Word	Select Threshold Mode LOW, HI, WIN coded to (0,1,2)
<b>Para17:</b> THRESHOLD TRACING	Word	Select Threshold Tracing OFF, ON TOL, ON CONT coded to (0,1,2)
<b>Para18:</b> TT UP	Word	Time delay for Threshold Tracing Up (0...60000)
<b>Para19:</b> TT DOWN	Word	Time delay for Threshold Tracing Down (0...60000)
<b>Para20:</b> THRESHOLD CALC	Word	Threshold Calculation ABSOLUTE (digit), RELATIVE (%) coded to (0,1)
<b>Para21:</b> TEACH VALUE	Word	Teach Value (Reference) for threshold calculation (0...4095)
<b>Para22:</b> TOLERANCE	Word	Tolerance Value for threshold calculation (0...4095)
<b>Para23:</b> HYSTERESIS	Word	Hysteresis Value for threshold calculation (0...4095)
<b>Para24:</b> EXTERN TEACH	Word	External teach mode: OFF, DIRECT, DYN, MAX, MIN, (MAX-MIN)/2+MIN coded to (0,1,2,3,4,5)
<b>Para25:</b> EVALUATION MODE	Word	Evaluation Mode CH0, CH1, CH0-CH1, (CH0+CH1)/2, CH0/(CH0+CH1) coded to (0,1,2,3,4)
<b>Para26:</b> DEAD TIME	Word	Dead Time in [%] coded to (0...100)

Die vom Sensor erfassten und aufbereiteten Daten werden auf Anfrage in folgender Reihenfolge vom Sensor gesendet.

TABLE DATA VALUE		
DATA VALUE	Type	Meaning
<b>DatVal1:</b> CH0	Word	Analogue raw signal of the receiver channel 0
<b>DatVal2:</b> CH1	Word	Analogue raw signal of the receiver channel 1
<b>DatVal3:</b> TEMP	Word	Sensor internal temperature (not in °C or F)
<b>DatVal4:</b> REF	Word	Reference value for threshold calculation
<b>DatVal5:</b> SIG	Word	Evaluation Signal
<b>DatVal6:</b> MIN	Word	Minimum value of SIG during IN0 was HI
<b>DatVal7:</b> MAX	Word	Maximum value of SIG during IN0 was HI
<b>DatVal8:</b> DIGITAL IN	Word	Bit 0 is 0 when IN0 is LO Bit 0 is 1 when IN0 is HI Bit 1 is 0 when IN1 is LO Bit 1 is 1 when IN1 is HI
<b>DatVal9:</b> DIGITAL OUT	Word	Bit 0 is 0 when signal is out of tolerance Bit 0 is 1 when signal is in tolerance Bit 1 is 0 when THRESHOLD MODE = LO or HI Bit 1 is 0 when THRESHOLD MODE = WIN and signal is under the window. Bit 1 is 1 when THRESHOLD MODE = WIN and signal is above window.
<b>DatVal10:</b> ANALOG OUT	Word	Analogue output value

Der Datenaustausch zwischen der Software, die auf dem PC läuft und dem Sensor erfolgt digital seriell. Dazu besitzt die Kontrolleinheit eine EIA-232 kompatible Schnittstelle, die mit den (festen) Parametern „8 Daten bit, 1 Stopp bit, kein Paritätsbit, kein Handshake“ betrieben wird. Die Baudrate kann aus fünf Optionen gewählt werden: 9600baud, 19200baud, 38400baud, 57600baud und 115200baud. Wahlweise kann die PC Software auch über TCP/IP bzw. USB kommunizieren. In diesen Fällen sind transparente Schnittstellenwandler zu benutzen, die die Anbindung an die RS232 Schnittstelle ermöglichen.



Für alle physikalischen Varianten der Verbindung zwischen PC Software und Kontrollelektronik gleichermaßen wird ein proprietäres Protokollformat verwendet, das die gewünschten Daten organisiert und bündelt. Die Nutzdaten sind abhängig von ihrer Art und Funktion 16- oder 32bit Variablen und repräsentieren Ganzzahl oder Gleitkommawerte. Die Bestandteile des Protokollformats sind 8bit weite vorzeichenlose Worte („bytes“). Dadurch müssen Nutzdaten zum Teil auf mehrere Bytes verteilt werden.

Die Kontrolleinheit verhält sich immer passiv (es sei denn ein anderes Verhalten wurde gezielt aktiviert). Ein Datenaustausch wird daher immer von der PC Software initiiert. Der PC sendet einem dem Protokollformat entsprechendes Datenpaket („Frame“) wahlweise mit oder ohne angehängte Daten, worauf die Kontrolleinheit mit einem der Anforderung entsprechenden Frame antwortet.

Das Protokollformat besteht aus zwei Komponenten:

Einem Kopfteil („Header“) und einem optionalen Anhang („Data“).

Die Struktur des Headers ist stets gleich.

Das erste Byte ist ein Synchronisationsbyte und ist immer 85<sub>dez</sub> (55<sub>hex</sub>).

Das zweite Byte ist das sog. Befehlsbyte (order), das bestimmt welche Aktion durchgeführt werden soll (Daten senden, Daten speichern, usw.).

Als drittes und viertes Byte folgt ein 16bit Wert (Argument). Das Argument wird abhängig vom Befehl mit einem entsprechenden Wert belegt.

Das fünfte und sechste Byte bilden wieder einen 16bit Wert. Er gibt die Anzahl der angehängten Datenbytes an.

Ohne angehängte Daten sind beide 0<sub>dez</sub> bzw. 00<sub>hex</sub>, die maximale Anzahl an Bytes sind 512.

Das siebte Byte wird mit der CRC8 Checksumme über alle Datenbytes (Datenbyte 1 bis incl. Datenbyte n) belegt.

Das achte Byte ist die CRC8 Checksumme über den Header und wird über die Bytes 1 bis incl. 7 gebildet.

Die Gesamtlänge des Headers ist immer 8 Bytes. Der gesamte Frame kann zwischen 8 und 520 Bytes umfassen.

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

Folgende **Befehle** können zum Sensor abgesetzt werden.

Number	ORDER (header byte no. 2)	Example
0	Sensor answers with order=0 if a communication error occurs. ARG=1: Invalide 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
105	Get cycle time from sensor	order=105
190	Write new baud rate to the sensor	order=190

### CRC8 Checksumme

Zur Verifizierung der Datenintegrität wird der sog. „Cyclic Redundancy Check“ oder CRC verwendet. Mit Hilfe dieses Algorithmus können einzelne Bitfehler, fehlende Bytes und fehlerhafte Frames erkannt werden. Dazu wird über die zu testenden Daten (-bytes) ein Wert – die sog. Checksumme – berechnet und mit dem Datenpaket übertragen. Die Berechnung folgt dabei einer genau vorgegebenen Methode basierend auf einem Generatorpolynom. Die Länge der Checksumme ist 8bit (= 1 byte). Das Generatorpolynom entspricht:

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

Um die Daten nach dem Empfang zu verifizieren wird die CRC Berechnung erneut durchgeführt. Stimmen übertragener und neu errechneter CRC Wert überein, sind die Daten fehlerfrei.

Um die Checksumme zu berechnen kann folgender Pseudocode verwendet werden:

```

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.

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

DATA FRAME PC → Sensor

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

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

DATA FRAME Sensor → PC

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

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

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

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

DATA FRAME PC → Sensor

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

DATA FRAME Sensor → PC

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

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

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

DATA FRAME PC → Sensor

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

DATA FRAME Sensor → PC

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

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

DATA FRAME PC → Sensor

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

DATA FRAME Sensor → PC

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

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

DATA FRAME PC → Sensor

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

DATA FRAME Sensor → PC

ARG determines the serial number of the sensor

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

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

DATA FRAME PC → Sensor

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

DATA FRAME Sensor → PC

ARG determines the serial number of the sensor

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

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

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

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

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

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

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



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

DATA FRAME PC → Sensor

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

DATA FRAME Sensor → PC

Byte1 Header	Byte2 Header	Byte3 Header	Byte4 Header	Byte5 Header	Byte6 Header	Byte7 Header	Byte8 Header	Byte9 Data	Byte10 Data	Byte11 Data	Byte12 Data
0x55	<order>	<ARG> (lo byte)	<ARG> (hi byte)	<LEN> (lo byte)	<LEN> (hi byte)	CRC8 (Data)	CRC8 (Header)	Para1 (lo byte)	Para1 (hi byte)	Para2 (lo byte)	Para2 (hi byte)
85 (dec)	8	0	0	14	0	325	154	76	11	1	0
ARG=0				LEN=14				DatVal1 = 2892		DatVal2 = 0	

Byte13 Data	Byte14 Data	Byte15 Data	Byte16 Data	Byte17 Data	Byte18 Data	Byte19 Data	Byte20 Data	Byte21 Data	Byte22 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)
184	11	17	0	0	0	0	0	0	0
DatVal3 = 3000		DatVal4 = 17		DatVal5 = 0		DatVal6 = 0		DatVal7 = 0	

**Example order=105:** Get cycle time from sensor

DATA FRAME PC → Sensor

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

DATA FRAME Sensor → PC

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

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

$$\text{Cycle Time [Hz]} = \text{CYCLE COUNT} / (\text{COUNTER TIME} * 0,0001)$$

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

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

DATA FRAME PC → Sensor

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

New baud rate is determined by argument.

ARG=0: baud rate = 9600

ARG=1: baud rate = 19200

ARG=2: baud rate = 38400

ARG=3: baud rate = 57600

ARG=4: baud rate = 115200

DATA FRAME Sensor → PC

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