Manual
Software SPECTRO1-Scope V2.0

(PC software for Microsoft® Windows® 7, Vista, XP, 2000, Me)

for sensors of SPECTRO-1 Series

This manual describes the installation of the PC software for the sensors of SPECTRO-1 series. As a support for commissioning of the color sensor this manual explains the functional elements of the Windows® user interface.

The sensors of the SPECTRO-1 series are of single-channel design, i.e. they acquire the analog signal that comes from a receiver and evaluate this signal. Various light sources such as white light, UV light, IR light, or a laser can be used as a transmitter. The receiver is correspondingly matched to the transmitter. The acquired analog signal is provided through a voltage output and a current output.
The software can be used to select various evaluation modes for the analog signal. The status of the analog signal is provided through 2 digital outputs in accordance with the selected evaluation mode.
A digital input allows external "teaching" of the sensor.
An additional input allows the "freezing" of the analog output signal upon a positive input edge.

The SPECTRO-1 sensor allows highly flexible signal acquisition. The sensor, for example, can be operated in alternating-light mode (AC mode), which means the sensor is not influenced by external light, or in constant-light mode (DC mode), which provides outstanding high-speed sensor operation. An OFF function deactivates the sensor's integrated light source and changes to DC mode, which allows the sensor to detect so-called "self-luminous objects". With the stepless adjustment of the integrated light source, the selectable gain of the receiver signal, and an INTEGRAL function the sensor can be adjusted to almost any surface or any "self-luminous object".

A micro-controller performs 12-bit analog/digital conversion of the analog signal, which allows recording and evaluation of the signal. Furthermore the SPECTRO-1 sensor offers various options for intelligent signal processing such as e.g. dirt accumulation compensation.

Parameters and measurement values can be exchanged between PC and sensor either through RS232 or Ethernet (using an Ethernet adaptor, e.g. SI-RS232/Ethernet-4-...). Through the interface all the parameters can be stored in the non-volatile EEPROM of the sensor.

The PC software facilitates the parameterisation, diagnostics, and adjustment of the sensor system (oscilloscope function). The software furthermore provides a data recorder function that automatically records data and stores them on the hard disk of the PC.

SPECTRO-1 sensors are temperature-compensated over a range of 0°C to 80°C.

Possible firmware updates can be easily performed through the RS232 interface, even with the sensor system in installed condition.

When parameterisation is finished, the color sensor continues to operate with the current parameters in STAND-ALONE mode without a PC.

Sensor Instruments GmbH
Schlinding 11 - 94169 Thurmansbang
Tel.: +49 8544 / 9719-0 - Fax: +49 8544 /9719-13
E-Mail: info@sensorinstruments.de
http://www.sensorinstruments.de

0 Contents

Shortcuts:

SEND	F9
GET	F10
GO	F11
STOP	F12

1 Installation of the SPECTRO1-Scope software

Hardware requirements for successful installation of the SPECTRO1-Scope software:

- IBM PC AT or compatible
- VGA graphics
- Microsoft® Windows® 7, Vista, XP, Me, 2000
- Serial RS232 interface at the PC
- Microsoft-compatible mouse
- Cable for the RS232 interface (cab-las4/PC or cab-las4/USB)
- CD-ROM drive
- 20 MByte of free hard disk space

The SPECTRO1-Scope software can only be installed under Windows. Windows must therefore be started first if it is not yet running.

Please install the software as described below:

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

2. The installation program displays a dialog and suggests to install the software in the C:\”FILENAME” directory on the hard disk. You may accept this suggestion with OK or [ENTER], or you may change the path as desired. Installation is then performed automatically.

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

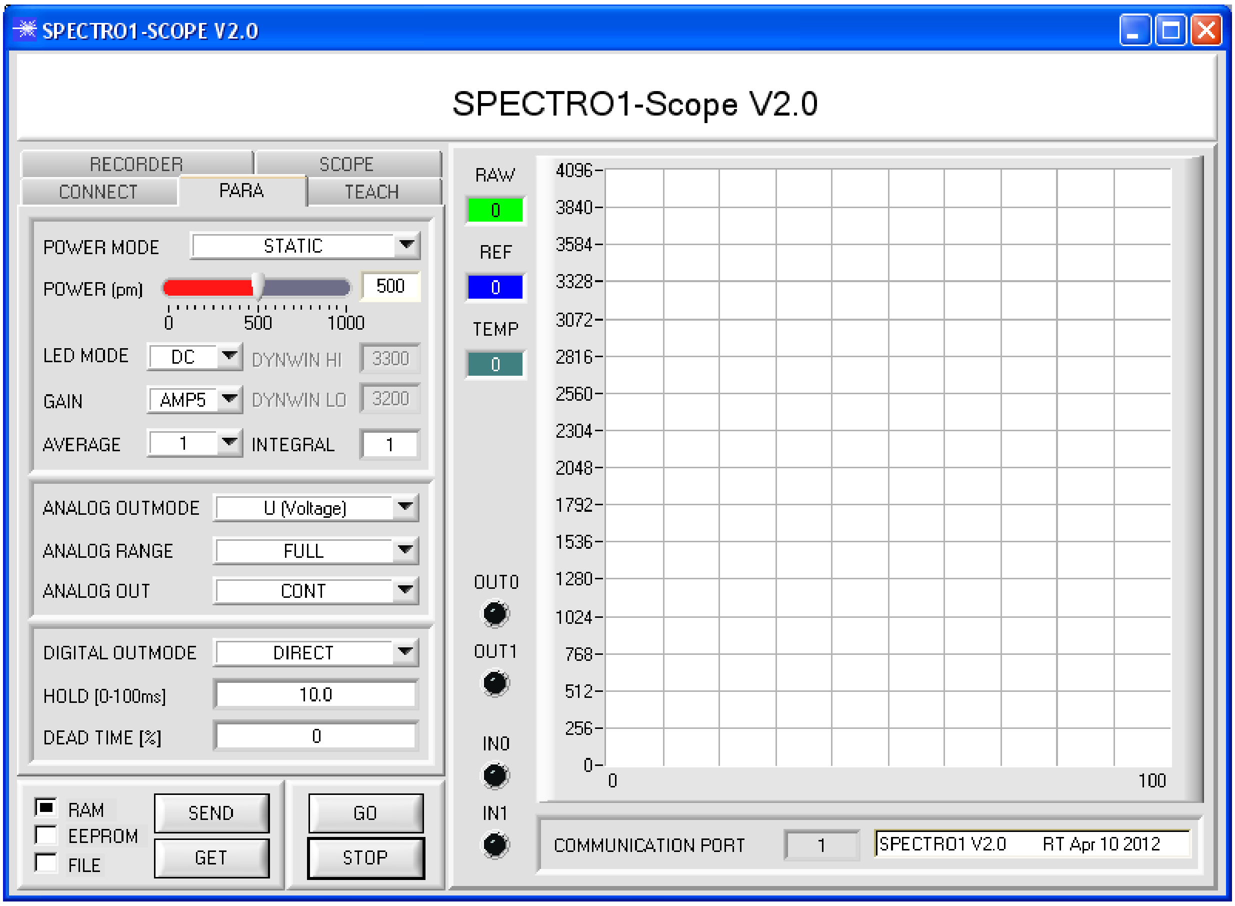
4. After successful installation the software can be started with a left mouse button double-click on the icon.

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

2 Operation of the SPECTRO1-Scope software

Please read this chapter first before you start to adjust and parameterise the SPECTRO-1 sensor.

When the SPECTRO1-Scope software is started, the following window appears on the Windows interface:

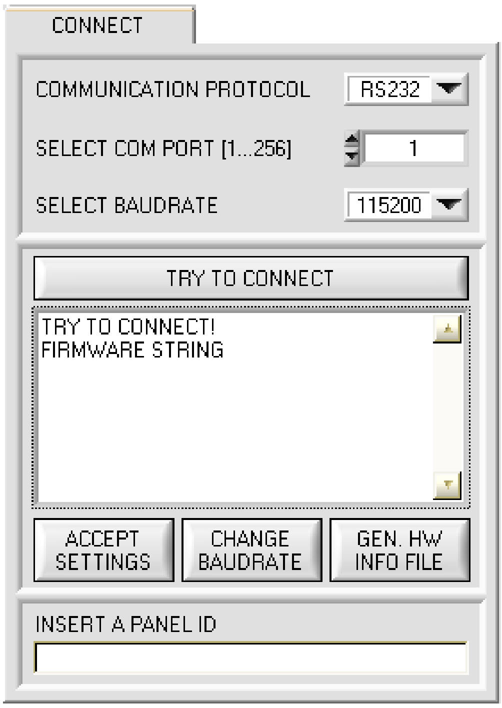


POWER MODE STATIC
POWER (pm) 500
LED MODE DC
DYNWIN HI 3300
GAIN AMP5
DYNWIN LO 3200
AVERAGE 1
INTEGRAL 1
ANALOG OUTMODE U (Voltage)
ANALOG RANGE FULL
ANALOG OUT CONT
DIGITAL OUTMODE DIRECT
HOLD [0-100ms] 10.0
DEAD TIME [%] 0
RAW 0
REF 0
TEMP 0
■ RAM
☐ EEPROM
☐ FILE
COMMUNICATION PORT 1
SPECTRO1 V2.0 RT Apr 10 2012

The window size and position will be the same as when the software was last closed. A double-click with the right mouse button e.g. under the minimise symbol places the window centrally in its original size.

Pressing the right mouse button on an individual element will call up a short help text.

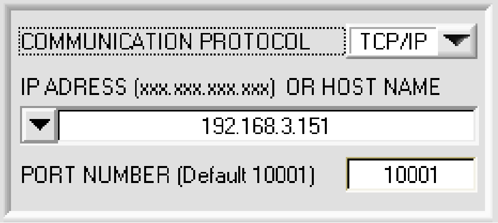
2.1 Tab CONNECT



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

TRY TO CONNECT!
FIRMWARE STRING

INSERT A PANEL ID



COMMUNICATION PROTOCOL: TCP/IP
IP ADRESS (xxx.xxx.xxx.xxx) OR HOST NAME: 192.168.3.151
PORT NUMBER (Default 10001): 10001

CONNECT:
Pressing the **CONNECT** tab opens a window for selecting and configuring the interface.

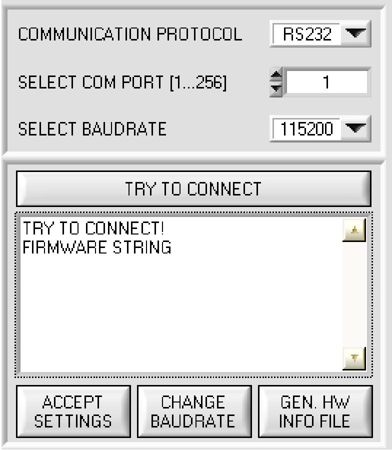
The **COMMUNICATION PROTOCOL** function field is used for selecting either an **RS232** or a **TCP/IP** protocol.
If **RS232** is selected, a port from 1 to 256 can be selected with **SELECT COM PORT**, depending on which port the sensor is connected to.
The sensor operates with a set baudrate that can be modified with **CHANGE BAUDRATE** (see below). The sensor and the user interface both must operate with the same baudrate. At the user interface the baudrate is set with **SELECT BAUDRATE**. If after starting the software should not automatically establish a connection, the correct baudrate can be found with **SELECT BAUDRATE**.
If an adaptor is used, the **COM PORT** number can be determined by way of the hardware manager in the system control panel.
If the sensor should communicate through a local area network, an RS232 to Ethernet adaptor will be needed. This adapter makes it possible to establish a connection to the sensor with the **TCP/IP** protocol.
The network adaptors that are available from us are based on the Lantronix XPort module. For parameterising these adapters (assigning of an IP address, setting of the Baud rate of 19200) please download the "DeviceInstaller" software that is provided free of charge by Lantronix at http://www.lantronix.com/. DeviceInstaller is based on Microsoft's ".NET" framework. Detailed operating instructions for the "DeviceInstaller" software also are available from Lantronix.

In order to establish a connection to the adaptor, its IP address or HOST name must be entered in the field **IP ADDRESS (xxx.xxx.xxx.xxx) OR HOST NAME**. The DROP DOWN menu (down arrow) shows the last 10 IP addresses that were used. An address from this list can be directly selected by clicking on the respective item. The DROP DOWN list is saved and is thus always available when the software is closed.
The **PORT NUMBER** for the XPort-based network adaptors is 10001. This port number must not be changed.
When you press the **TRY TO CONNECT** button, the software tries to establish a connection with the set parameters. The communication status is shown in the display field. If the sensor answers with its FIRMWARE ID, the set connection type can be accepted by pressing **ACCEPT SETTINGS**. You will then be returned to the **PARA** tab. If you get a **TIMEOUT** message, the software could not establish a connection to the sensor. In this case please check if the interface cable is correctly connected, if the sensor is supplied with power, and if the set parameters are correct.
If a connection has been accepted by pressing **ACCEPT SETTINGS**, the software starts automatically with these settings when called the next time.

Please note: **The stable function of the interface is a basic prerequisite for measured value transfer from the PC to the sensor.**

Due to the limited data transfer rate through the serial RS232 interface only slow changes of the raw signals at the sensor front end can be observed in the graphic output window of the PC.
For maintaining maximum switching frequency at the sensor data communication with the PC must be stopped (press the STOP button).

ATTENTION !

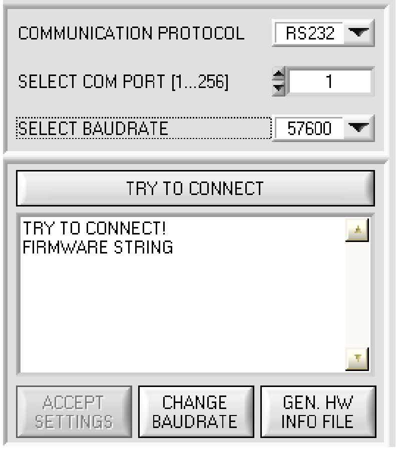


COMMUNICATION PROTOCOL RS232
SELECT COM PORT [1...256] 1
SELECT BAUDRATE 115200

```
TRY TO CONNECT!
FIRMWARE STRING
```

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

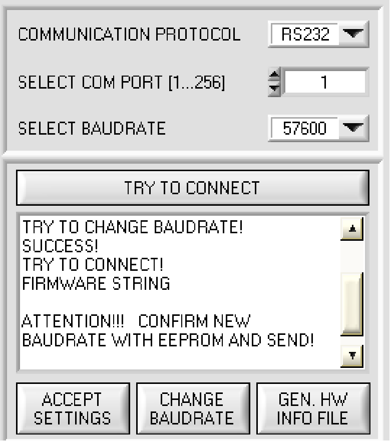
If the baudrate should be changed, a connection must first be established by clicking on **TRY TO CONNECT**.
The **CHANGE BAUDRATE** button will then be active.



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

```
TRY TO CONNECT!
FIRMWARE STRING
```

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



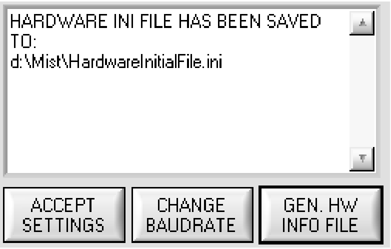
COMMUNICATION PROTOCOL RS232
SELECT COM PORT [1...256] 1
SELECT BAUDRATE 57600

```
TRY TO CHANGE BAUDRATE!
SUCCESS!
TRY TO CONNECT!
FIRMWARE STRING

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

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

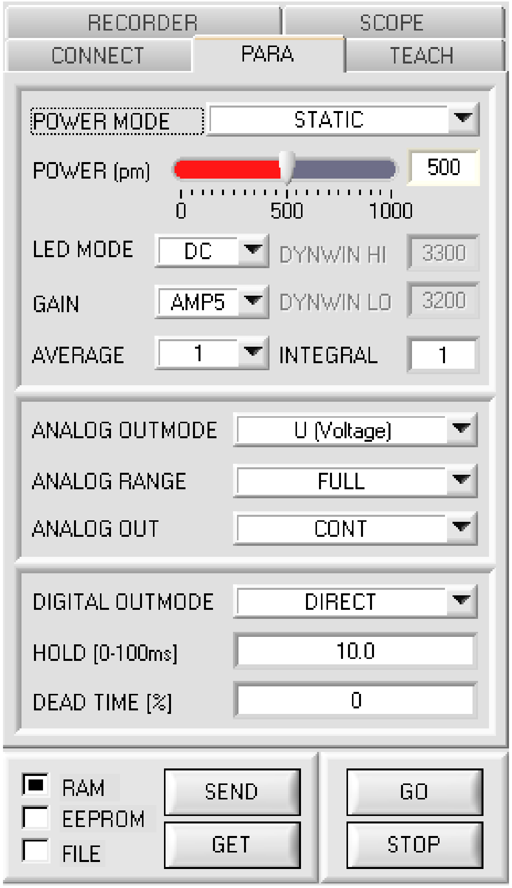
A click on **ACCEPT SETTINGS** saves the current interface settings, which will then be automatically set when the software is restarted.



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

A click on the **GEN. HW INFO FILE** generates a file in which all the important sensor data are stored in encrypted form.
This file can be sent to the manufacturer for diagnostic purposes.

2.2 Tab PARA, button SEND, GET, GO, STOP



POWER MODE: STATIC
POWER (pm): 500
LED MODE: DC
DYNWIN HI: 3300
GAIN: AMP5
DYNWIN LO: 3200
AVERAGE: 1
INTEGRAL: 1
ANALOG OUTMODE: U (Voltage)
ANALOG RANGE: FULL
ANALOG OUT: CONT
DIGITAL OUTMODE: DIRECT
HOLD [0-100ms]: 10.0
DEAD TIME [%]: 0

☑ RAM
☐ EEPROM
☐ FILE

PARA:
Pressing the **PARA1** tab opens a window for setting the sensor parameters.

ATTENTION!
A change of the parameter function groups only becomes effective at the sensor after actuation of the SEND button in the MEM function field!

SEND [F9]:
When the **SEND** button is clicked (or shortcut key button F9 is pressed), all the currently set parameters are transferred between PC and sensor. The target of the respective parameter transfer is determined by the selected button (**RAM, EEPROM,** or **FILE**).

GET [F10]:
The currently set values can be interrogated from the sensor by clicking on the **GET** button (or with shortcut key button F10). The source of data exchange is determined by the selected button (**RAM**, **EEPROM**, or **FILE**).

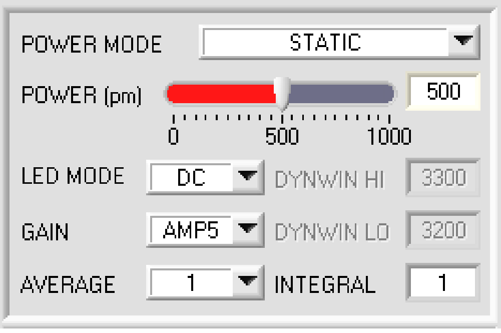
RAM:
The **RAM** is a **volatile** memory in the sensor's micro-controller, i.e. when the power at the sensor is turned off, these parameters will be lost again.
The sensor always operates with the parameters in its RAM.
If the **RAM** option is selected, a click on **SEND** writes the current parameters to the sensor's **RAM** memory, and a click on **GET** reads the parameters from the sensor's **RAM** memory.

EEPROM:
The **EEPROM** is a **non-volatile** memory in the sensor's micro-controller. When the power at the sensor is turned off the parameters in the **EEPROM** will not be lost. When power is turned on again, the parameters are loaded from the **EEPROM** to the **RAM** memory. Figuratively speaking the **EEPROM** thus is a level lower than the **RAM**. Data exchange between **PC** and **EEPROM** automatically is performed through the **RAM** memory, which means that parameters that are written to the **EEPROM** automatically are also written to the **RAM,** and data that are read from the **EEPROM** automatically are also read to the **RAM**.
If the **EEPROM** option is selected, a click on **SEND** writes the current parameters to the sensor's non-volatile **EEPROM** memory, and a click on **GET** reads the parameters from the sensor's **EEPROM**.
The **RAM** memory should always be used for parameterising the sensor. When suitable parameters have been found for the respective application, these parameters must be written to the sensor's **EEPROM** so that after restarting the sensor these parameters can be loaded from the **EEPROM** into the **RAM** memory.

FILE:
After pressing **SEND**, the current parameters can be written to a selectable file on the hard disk. With **GET** parameters can be read from such a file. When the **SEND** or **GET** button is pressed, a dialog box opens for selecting the desired file.
TIP! Once suitable parameters have been found for a certain application, these should always be saved in a file on the PC.

GO [F11]:
A click on this button starts data transfer from the sensor to the PC through the serial RS232 interface.

STOP [F12]:
A click on this button stops data transfer from the sensor to the PC through the serial RS232 interface.



POWER MODE: STATIC
POWER (pm): 500
LED MODE: DC
DYNWIN HI: 3300
GAIN: AMP5
DYNWIN LO: 3200
AVERAGE: 1
INTEGRAL: 1

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

STATIC:
The transmitter power is constantly kept at the value set with the **POWER [pm]** slider (recommended operation mode). The **POWER** can be set with the slider or by entering a value in the edit-box. A value of 1000 means full intensity at the transmitter unit, a value of 0 sets the lowest intensity at the transmitter.

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

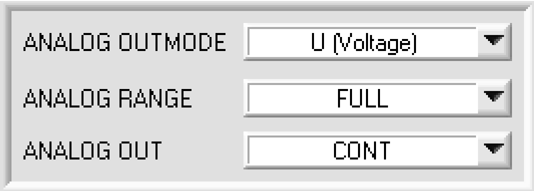
LED MODE:
This item serves for setting the control mode for the integrated light source of the sensor.
DC: In this mode the sensor operates extremely fast. Unfortunately the sensor is somewhat sensitive to extraneous light in DC mode, but if the extraneous light source does not directly shine into the sensor's receiver, the signal only is influenced to a very small extent.
AC: In this mode the sensor is insensitive to extraneous light, which is achieved by "modulating" the integrated light source, i.e. by turning the light on and off. The extraneous content in the signal is determined in off status and is simply subtracted from the on status.
OFF: The sensor's internal light source is turned off in DC mode by POWER [pm] = 0, the sensor can be used for so-called "self-luminous objects". Self-luminous objects are light sources that actively emit light (LEDs, lamps, etc.). In **OFF** mode the **POWER MODE** and **POWER** cannot be adjusted, and external teaching with **DYN** is not possible.

GAIN:
This item is used for setting the gain of the receiver in 8 different gain stages (AMP1 to AMP8). **GAIN** should be set such that with a medium **POWER** value the sensor operates in its dynamic range (red, green, blue between 2750 and 3750).
In **AC** mode, **GAIN** directly influences the scan frequency. The current scan frequency is displayed in the **SCOPE** tab.

AVERAGE:
This function field is used for adjusting the number of scanning values (measurement values) over which the raw signal measured at the receiver is averaged. A higher **AVERAGE** default value reduces noise of the raw signals at the receiver unit and there will be a decrease of the maximal available switching frequency of the sensor

INTEGRAL:
This function field is used to set the number of scan values (measurement values) over which the raw signal measured at the receiver is summed up. This integral function allows the reliable detection even of extremely weak signals. A higher **INTEGRAL** value increases the noise of the raw signals of the receiver unit, and simultaneously decreases the maximum achievable switching frequency of the sensor.

INFO:
The **POWER** slider is only effective in the **POWER MODE = STATIC**.
DYN WIN LO and **DYN WIN HI** are only effective in **POWER MODE = DYNAMIC.**



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

The **RAW** signal is used for outputting an analog signal.
The **RAW** signal is acquired with a resolution of 12 bit and therefore may have values between 0 and 4095

ANALOG OUTMODE:
This function field is used to determine the analog outputs that the sensor uses.

OFF:
No analog signal is output.

U (Voltage):
At the respective analog output **RAW** is provided as a voltage of 0 - 10V.

I (Current):
At the respective analog output **RAW** is provided as a current of 4 - 20mA.

U + I:
At the respective analog outputs **RAW** is provided both as a voltage and as a current.

ANALOG RANGE:
This function field is used to determine how and in which range the sensor provides the **RAW** signal at its analog outputs.

FULL:
RAW is output in its full value range of 0-4095 as an analog signal.

MIN-MAX while IN0:
As long as input IN0 is HI, a maximum and minimum **RAW** value are determined in the sensor.
After IN0 drops, the analog signal is fully output (0-10 V and/or 4-20mA) within this **MIN-MAX** range.

ANALOG OUT:
This function field is used to determine when the sensor outputs the analog signal. .

CONT: The analog signal is output continuously.

RISING EDGE of IN1: The analog signal only is output when there is a positive edge at IN1.

DIGITAL OUTMODE:
This function field is used to determine the operating mode of digital outputs **OUT0** and **OUT1** when the tolerance threshold is **exceeded**. Depending on **OUTMODE** and **THRESHOLD MODE** the status of **OUT0** and **OUT1** is as follows:

	THRESHOLD MODE LOW and **HI**	**THRESHOLD MODE WIN**

OUTMODE OFF	OUT0=0 VDC
	OUT1=0 VDC
		OUT0=0 VDC
		OUT1=0 VDC

OUTMODE DIRECT	OUT0=0 VDC
	OUT1=0 VDC
		OUT0=0 VDC
		OUT1=0 VDC if RAW is lower than the tol. window
		OUT1=0+24 VDC if RAW is higher than the tol. window

OUTMODE INVERSE	OUT0=+24 VDC
	OUT1=0 VDC
		OUT0=+24 VDC
		OUT1=0 VDC if RAW is higher than the tol. window
		OUT1=0+24 VDC if RAW is lower than the tol. window

On the user interface the respective status of outputs **OUT0** and **OUT1** is visualised by way of the **OUT0** and **OUT1** LEDs.

HOLD: The sensor operates with minimum scanning times in the magnitude of less than 10µs. Entering a suitable value in the **HOLD** edit box sets a pulse lengthening of up to 100 ms at the sensor's digital output. This allows a PLC to reliably detect short changes of switching states.

DEAD TIME [%]:
This function field can be used to activate a dynamic dead time.

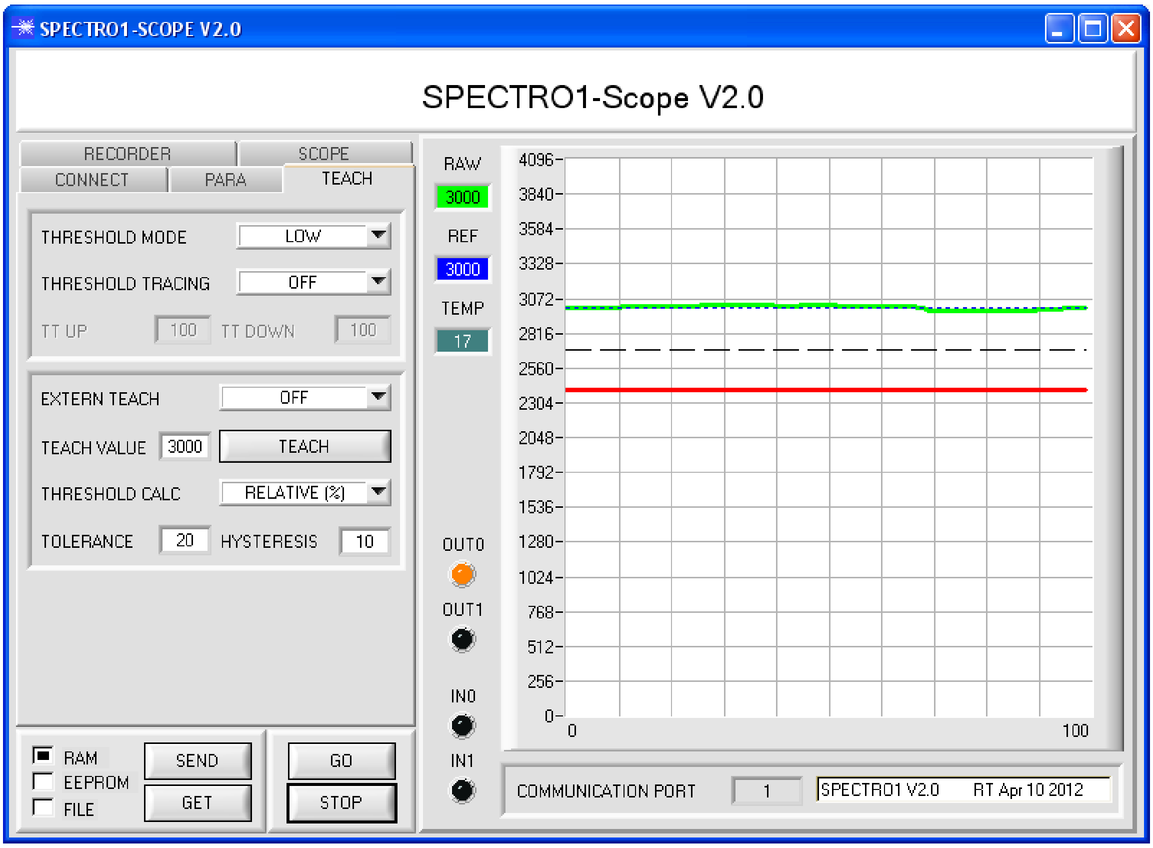
The dead time function can best be explained by way of an example.
Because of the application the sensor provides a regular switching signal of 100ms. These 100ms are interpreted as 100%. If a **DEAD TIME** of 20% is set, the sensor is "dead" for 20% after every switching signal. It ignores any fault that occurs within 20ms after the last switching process, and consequently also does not switch. For example this allows the correct detection of unclear edges without the sensor providing multiple pulses.
If the speed changes from 100ms to 200ms, 20ms dead time will become 40ms due to percentage calculation, which is why this is referred to as a dynamic dead time.

It is advisable to use a dead time, if multiple pulses cannot be suppressed with a suitable **HYSTERESIS**.

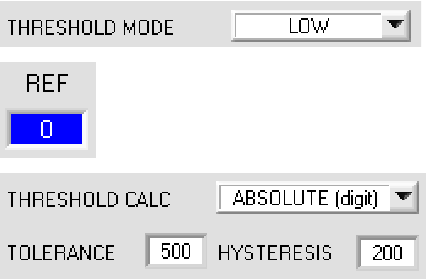
DEAD TIME [%] = 0 deactivates the dead time.

2.3 Tab TEACH

When the **GO** button is pressed, data transfer from the sensor to the PC is started.
The analog signal **RAW** is shown in a graph and in a numerical display.
Using the **TEACH** button is the easiest method for "teaching" an analog signal.
With a click on the **TEACH** button the current **RAW** value is accepted as the **TEACH VALUE**.
With a click on **SEND** the **TEACH VALUE** becomes the new reference value **REF**.
The switching threshold and the hysteresis threshold are then set based on **REF**.



THRESHOLD MODE LOW
THRESHOLD TRACING OFF
TT UP 100 TT DOWN 100
EXTERN TEACH OFF
TEACH VALUE 3000
THRESHOLD CALC RELATIVE (%)
TOLERANCE 20 HYSTERESIS 10
RAW 3000
REF 3000
TEMP 17
COMMUNICATION PORT 1 SPECTRO1 V2.0 RT Apr 10 2012



THRESHOLD MODE LOW
REF 0
THRESHOLD CALC ABSOLUTE (digit)
TOLERANCE 500 HYSTERESIS 200

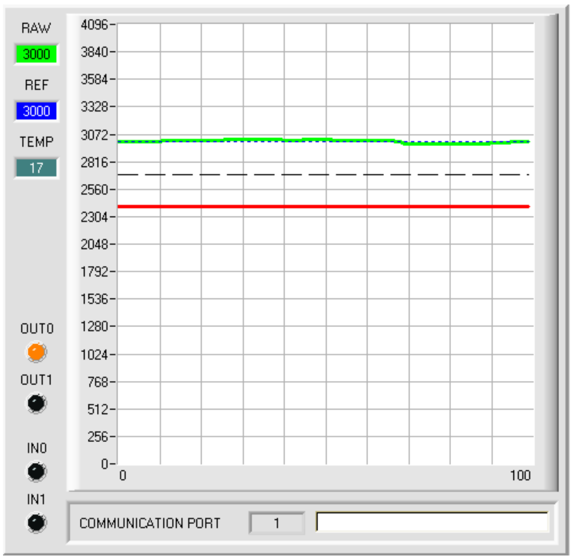
THRESHOLD MODE:
In this function field one of the possible positions of the switching threshold and hysteresis threshold with respect to the reference value **REF** can be selected.
The reference value is learned either by means of the software or through the external input IN0. Furthermore, automatic threshold tracing can be activated with **THRESHOLD TRACING**, which means that **REF** is cyclically re-determined.

Based on **REF** and with the help of **TOLERANCE** and **HYSTERESIS** the switching threshold and the hysteresis threshold are calculated differently with **THRESHOLD MODE LOW, HI** and **WIN**.

If **THRESHOLD CALC = ABSOLUTE(digit)** is selected, the thresholds are calculated absolutely in digits to **REF**, i.e. **TOLERANCE** and **HYSTERESIS** are directly added to or subtracted from **REF**.

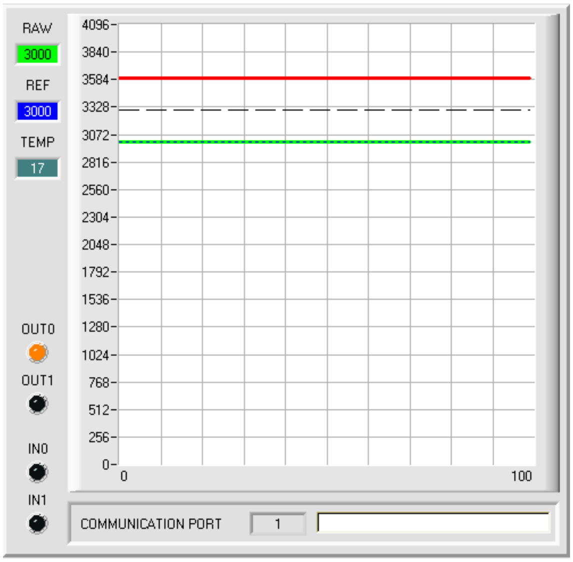
If **THRESHOLD CALC = RELATIVE(%)** is selected, the thresholds are calculated relatively to **REF**=100%, i.e. **TOLERANCE** and **HYSTERESIS** are relatively added to or subtracted from **REF**.

The different lines in the following graphic windows have the following meaning:
Green line = current measurement value RAW
Blue dotted line = reference value REF
Red line = switching threshold
Black dashed line = hysteresis threshold



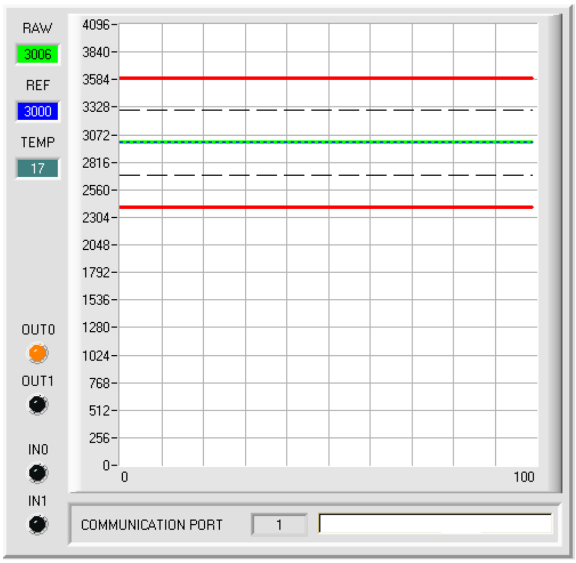
THRESHOLD MODE = LOW:
Switching threshold = **REF – TOLERANCE**
Hysteresis threshold = **REF** - **HYSTERESIS**

When the current measurement value **RAW** falls below the switching threshold, the digital output **OUT0** is set to error. When the current measurement value rises above the hysteresis threshold again, the error output is reset again.



THRESHOLD MODE = HI:
Switching threshold = **REF + TOLERANCE**
Hysteresis threshold = **REF + HYSTERESIS**

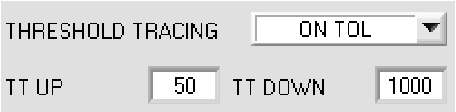
When the current measurement value **RAW** rises above the switching threshold, the digital output **OUT0** is set to error.
When the current measurement value falls below the hysteresis threshold again, the error output is reset again.



THRESHOLD MODE = WIN:
Upper switching threshold = **REF** + **TOLERANCE**
Upper hysteresis threshold = **REF** + **HYSTERESIS**

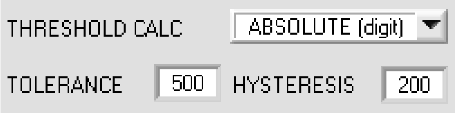
Lower switching threshold = **REF** - **TOLERANCE**
Lower hysteresis threshold = **REF** - **HYSTERESIS**

The switching thresholds form a symmetrical tolerance band around the current reference value. When the current measurement value **RAW** leaves this tolerance band, the digital output **OUT0** is set to error.
The error output is reset again when the current measurement value lies below/above the hysteresis threshold again.
Depending on **OUTMODE**, the output **OUT1** is set or reset when the signal leaves the tolerance window upwards or downwards.



THRESHOLD TRACING ON TOL

TT UP 50 TT DOWN 1000



THRESHOLD CALC ABSOLUTE (digit)

TOLERANCE 500 HYSTERESIS 200

THRESHOLD TRACING:
Automatic threshold tracing can be activated in this function field.
Actually the reference value **REF** cyclically follows a changing **RAW** value.
Switching threshold and hysteresis threshold then are set based on **REF**.
Generally this is referred to as threshold tracing.

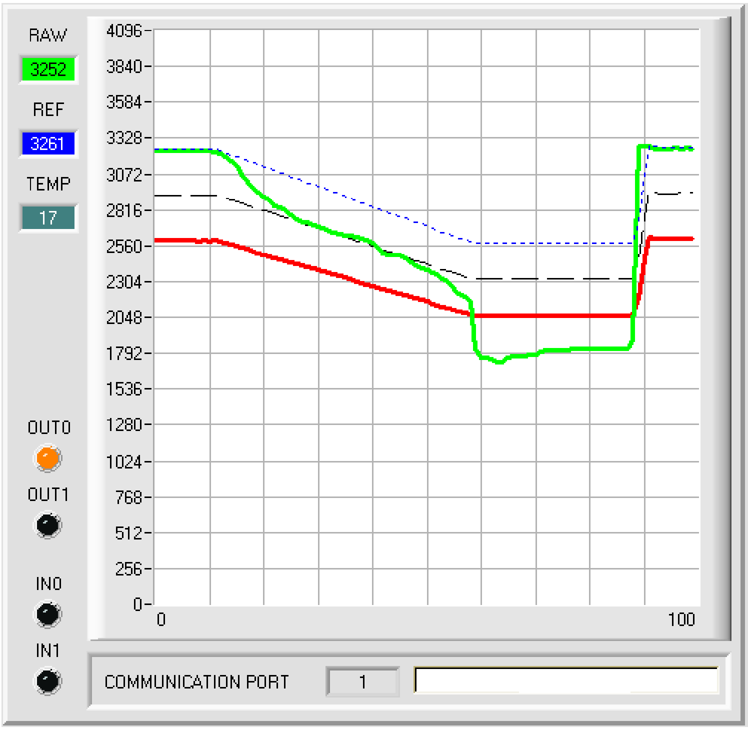
OFF: Automatic threshold tracing is deactivated.

ON TOL and **ON CONT:** Automatic threshold tracing is activated. The current reference value **REF** is cyclically adapted if the current **RAW** value decreases, e.g. due to increasing dirt accumulation.
With **ON TOL** tracing is performed when the current signal lies within the tolerance.
With **ON CONT** tracing is performed continuously.

TT UP and **TT DOWN**: In this function field a time constant for the speed of automatic threshold tracing can be set.
TT UP (THRESHOLD TRACING UP): When the current **RAW** value rises, **REF** is increased by one digit with the set delay.
TT DOWN (THRESHOLD TRACING DOWN): When the current **RAW** value falls, **REF** is decreased by one digit with the set delay.

A value between 0 and 60000 can be selected for **TT UP** and **TT DOWN**.
One increment means a delay of 100 microseconds. However, tracing cannot be faster than the scan frequency that can be achieved with **AVERAGE**.
Value 0:   Minimum time delay, fastest tracing.
Value 60000: Maximum time delay, slowest tracing.

The calculation of the switching thresholds depends on **THRESHOLD MODE** (see **THRESHOLD MODE**).



RAW 3252

REF 3261

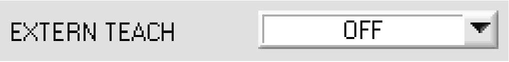
TEMP 17

COMMUNICATION PORT 1

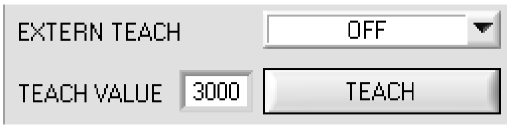
In this example a value of 50 was specified for **TT DOWN**. This means that downward threshold tracing is performed at a relatively slow speed.
In the chart this can be clearly seen in the blue reference line.

The measurement value **RAW** (green line) falls relatively fast, compared to this the reference value **REF** follows much slower. This is a typical case that occurs when increasing dirtying of the sensor must be compensated.

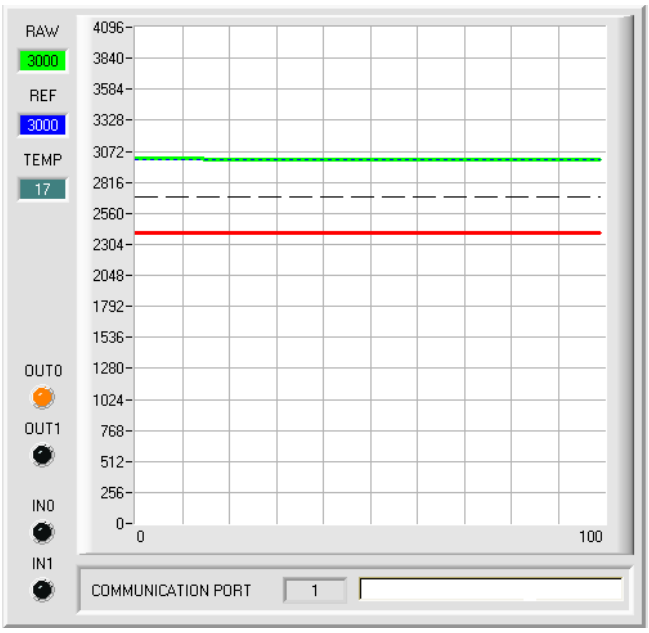
With **ON TOL**, threshold tracing is stopped when the current measurement value falls below the switching threshold. When **RAW** rises above the hysteresis threshold again, threshold tracing becomes active again. In this example upward threshold tracing is fast because a small value was specified for **TT UP**.


EXTERN TEACH: OFF

EXTERN TEACH:
This function field is used to specify how sensor "teaching" should be performed.


EXTERN TEACH: OFF
TEACH VALUE: 3000

EXTERN TEACH = OFF:
Using the **TEACH** button is the easiest method for "teaching" an analog signal.

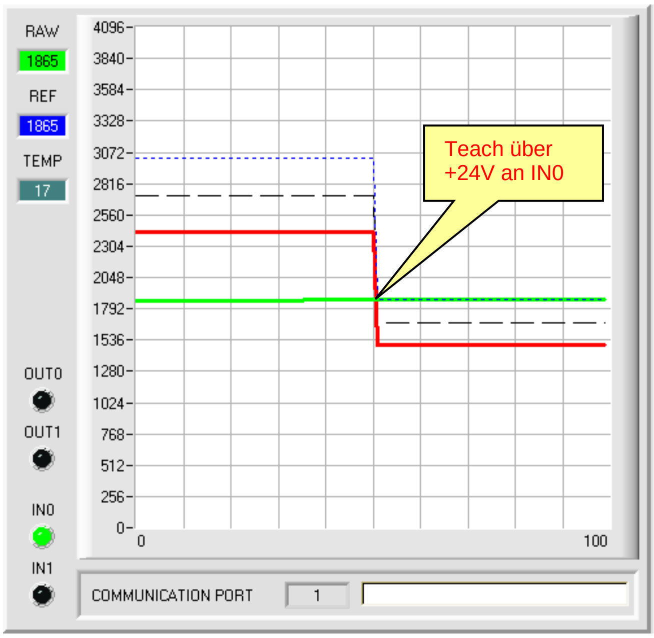

RAW: 3000
REF: 3000
TEMP: 17
COMMUNICATION PORT: 1

With a click on the **TEACH** button the current **RAW** value is accepted as the **TEACH VALUE**.

With a click on **SEND** the **TEACH VALUE** becomes the new reference value **REF**.

The switching threshold and the hysteresis threshold are then set based on **REF**.

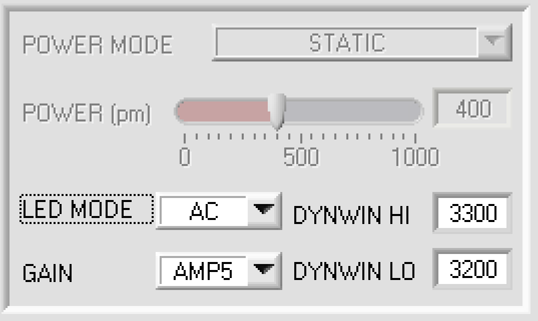
The **TEACH** button is not active if either **EXTERN TEACH** or **THRESHOLD TRACING** is not **OFF**.


RAW: 1865
REF: 1865
TEMP: 17
COMMUNICATION PORT: 1

EXTERN TEACH = DIRECT:
When there is a positive edge at input IN0, the current **RAW** value becomes the new reference **REF**.

The switching threshold and the hysteresis threshold are then set based on **REF**.

The new reference **REF** only is saved in the sensor's **RAM** and not in its **EEPROM**.



POWER MODE STATIC

POWER (pm) 400

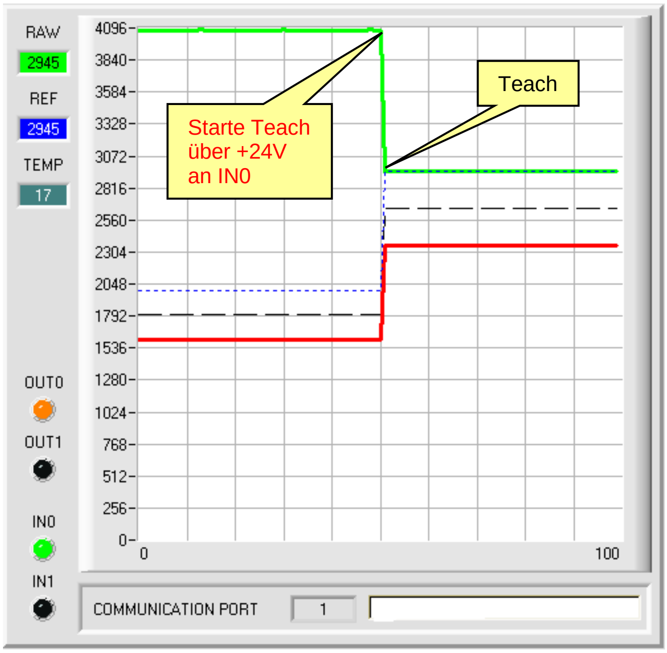
LED MODE AC

GAIN AMP5

DYNWIN HI 3300

DYNWIN LO 3200

EXTERN TEACH = DYN:
POWER MODE automatically is set to **STATIC** and like **POWER** is not active with **EXTERN TEACH = DYN**.



RAW 2945

REF 2945

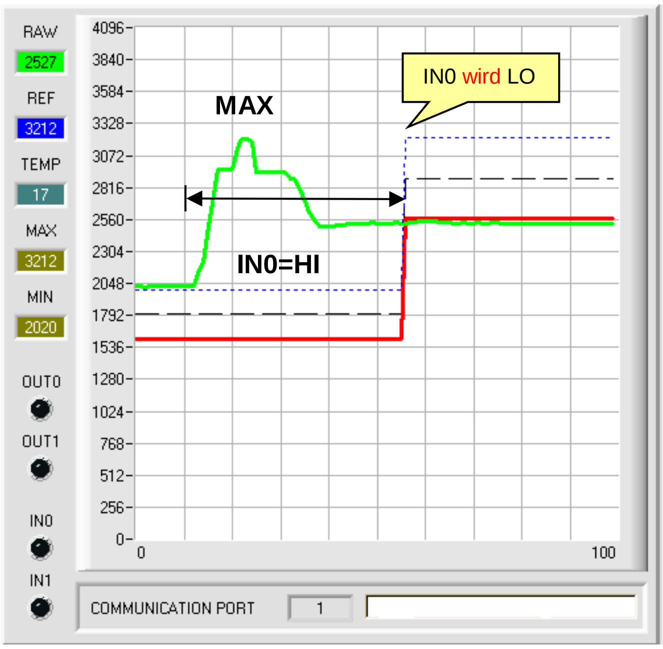
TEMP 17

COMMUNICATION PORT 1

After a positive signal (+24V) at input IN0 the transmitter power is adjusted such that the sensor is in its dynamic range, which is set by **DYN WIN LO** and **DYN WIN HI**.
The current **RAW** value becomes the new reference **REF**.
The switching threshold and the hysteresis threshold are then set based on **REF**.

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

The new reference **REF** only is saved in the sensor's **RAM** and not in its **EEPROM**.



RAW 2527

REF 3212

TEMP 17

MAX 3212

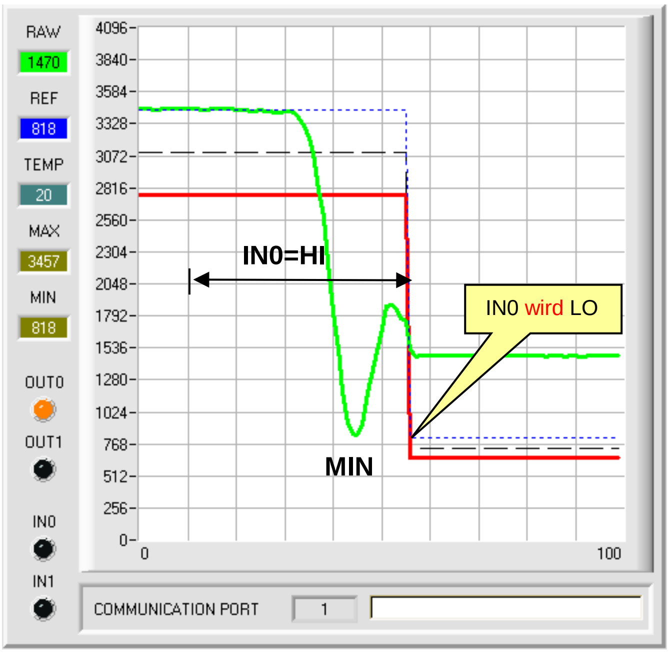
MIN 2020

COMMUNICATION PORT 1

EXTERN TEACH = MAX:
While input IN0=HI (+24V), the maximum **RAW** value is determined and becomes the new reference **REF** when IN0 changes to low.

The switching threshold and the hysteresis threshold are then set based on **REF**.

The new reference **REF** only is saved in the sensor's **RAM** and not in its **EEPROM**.



RAW 1470

REF 818

TEMP 20

MAX 3457

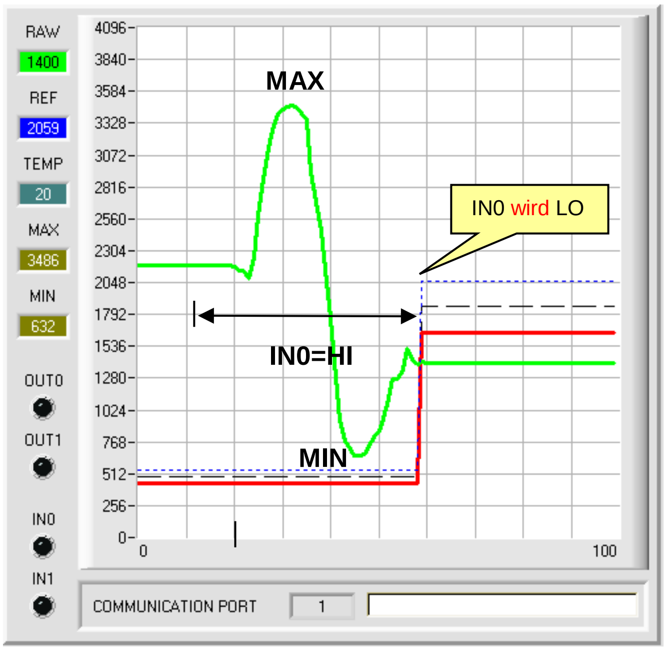
MIN 818

COMMUNICATION PORT 1

EXTERN TEACH = MIN:
While input IN0=HI (+24V), the minimum **RAW** value is determined and becomes the new reference **REF** when IN0 changes to low.

The switching threshold and the hysteresis threshold are then set based on **REF**.

The new reference **REF** only is saved in the sensor's **RAM** and not in its **EEPROM**.



RAW 1400

REF 2059

TEMP 20

MAX 3486

MIN 632

COMMUNICATION PORT 1

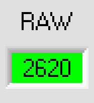
EXTERN TEACH = (MAX+MIN)/2:
While input IN0=HI (+24V), the maximum and minimum **RAW** values are determined.
When IN0 changes to low, the new reference **REF** is set exactly between MAX and MIN.

The switching threshold and the hysteresis threshold are then set based on **REF**.

The new reference **REF** only is saved in the sensor's **RAM** and not in its **EEPROM**.

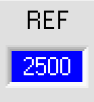
2.4 Graphic display elements

The software provides various display elements and a graphic window for the visualisation of all the data that are important for parameterisation. The individual display elements and the graph are explained in the chapter below.


RAW
2620

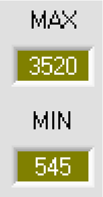
RAW:
This display shows the current measurement value.

In the graph **RAW** is visualised as a green line.

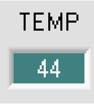

REF
2500

REF:
This display shows the current reference value. This value is the basis for the calculation of switching threshold and hysteresis threshold.
If **THRESHOLD TRACING** and **EXERN TEACH** are set to **OFF**, the **TEACH VALUE** is the reference value.
If **THRESHOLD TRACING** is set to **ON**, automatic threshold tracing is activated. The current reference value **REF** is cyclically adapted to compensate a lowering of the current **RAW** value, e.g. due to increasing dirt accumulation.
Depending on the setting of **EXERN TEACH** the reference **REF** can also be taught through input IN0 with various methods.

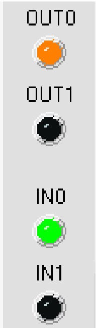
In the graph the reference value **REF** is represented as a blue line.


MAX
3520
MIN
545

MAX and MIN:
If with **EXTERN TEACH** or **ANALOG RANGE** a search for a minimum and/or maximum **RAW** value is necessary, these values are shown in these displays.
The two displays only are active when they are needed.

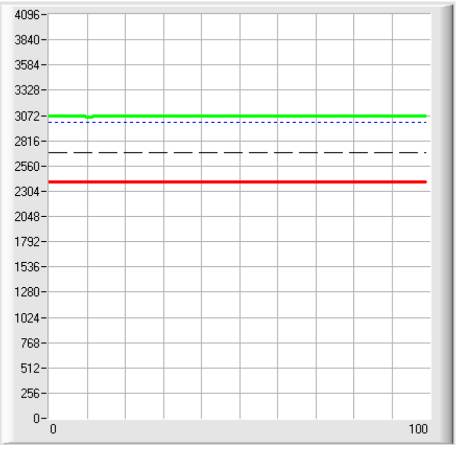

TEMP
44

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



OUT0 und OUT1:
These LEDs visualise the physical status of outputs OUT0 and OUT1.
When the LED is black, the output value is 0V.
When the LED is orange, the output value is +24V.

IN0 and IN1:
These LEDs visualise the physical status of inputs IN0 and IN1.
When the LED is black, the input value is 0V.
When the LED is green, the input value is +24V.



GRAPH:
The graphic display window shows the switching threshold (red line), the hysteresis threshold (black dashed line), the reference value **REF** (blue dotted line), and the current analog signal **RAW** (green line) depending on the set parameters.
In the graph of this example the parameter settings are as follows:
THRESHOLD MODE = LOW, THRESHOLD TRACING = OFF
EXTERN TEACH = OFF
TEACH VALUE = 3000 (→ REF=3000)
THRESHOLD CALC = RELATIVE (%)
TOLERANCE = 20
HYSTERESIS = 10
This results in a switching threshold of **TEACH VALUE** minus **TOLERANCE** (red line), and in a hysteresis threshold of **TEACH VALUE** minus **HYSTERESIS** (black dashed line).

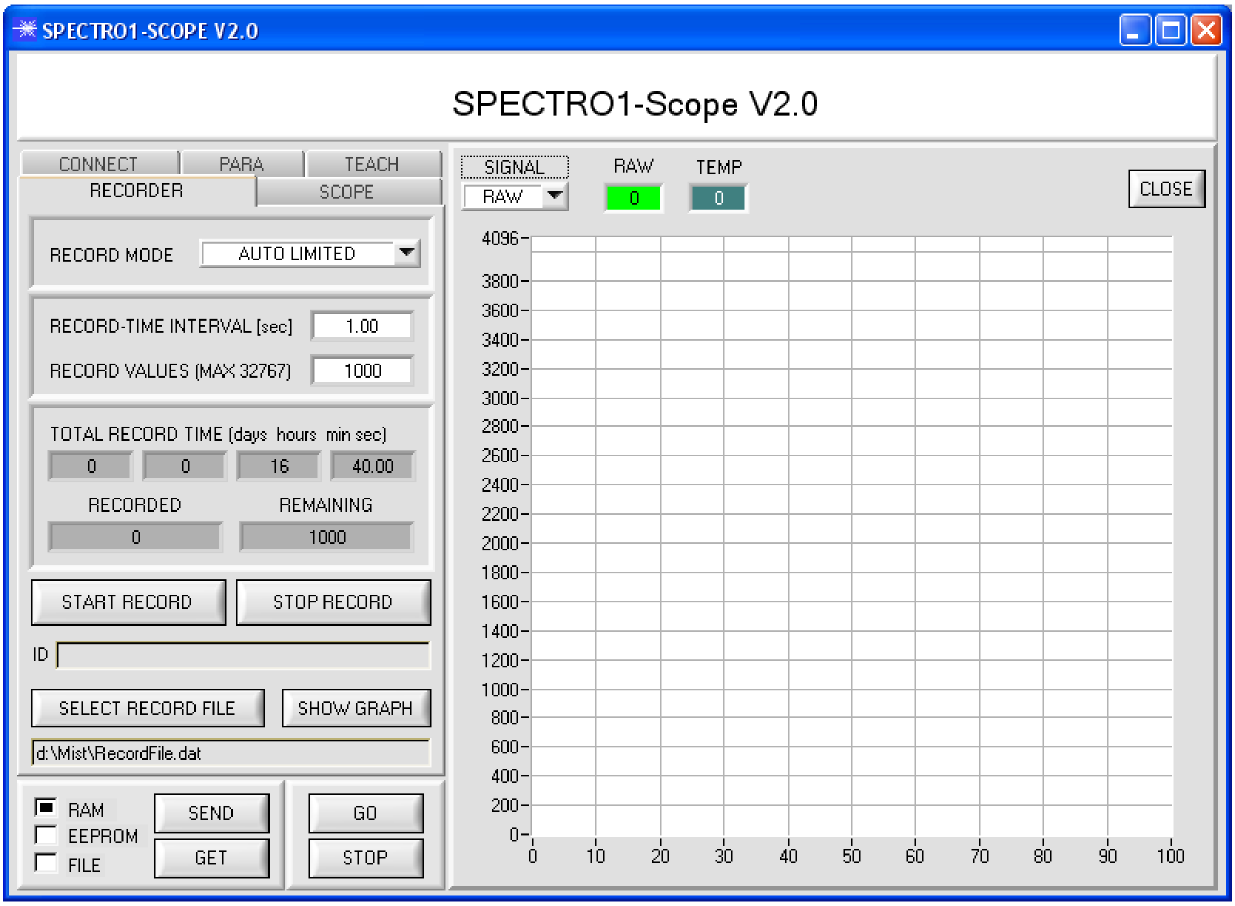
2.5 Tab RECORDER

The SPECTRO1-Scope software features a data recorder that allows the saving of **RAW** and **TEMP**. The recorded file is saved to the hard disk of the PC and can then be evaluated with a spreadsheet program.

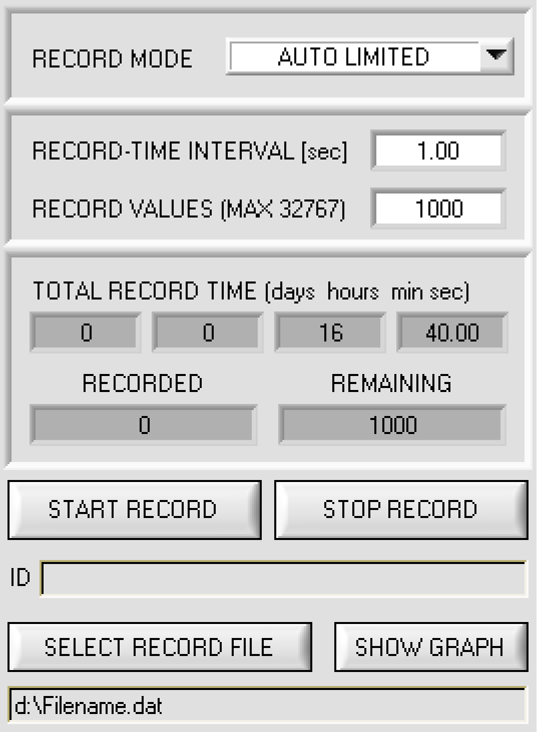
The file that is created has four columns and as many rows as data frames were recorded.
A row is structured as follows: **Date**, **time**, **RAW**, **TEMP**.

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

Step 1:
When the **RECORDER** button is pressed, the following window will be displayed:
When the **SHOW GRAPH** button is pressed, a panel will be displayed that allows the user to monitor the different signals.
The individual signals can be activated from the **SIGNAL** drop-down menu.



SIGNAL: RAW
RAW: 0
TEMP: 0
RECORD MODE: AUTO LIMITED
RECORD-TIME INTERVAL [sec]: 1.00
RECORD VALUES (MAX 32767): 1000
TOTAL RECORD TIME (days hours min sec): 0 0 16 40.00
RECORDED: 0
REMAINING: 1000
ID:
d:\Mist\RecordFile.dat
● RAW
☐ EEPROM
☐ FILE



RECORD MODE: AUTO LIMITED

RECORD-TIME INTERVAL [sec]: 1.00

RECORD VALUES (MAX 32767): 1000

TOTAL RECORD TIME (days hours min sec): 0 0 16 40.00

RECORDED: 0

REMAINING: 1000

ID:

d:\Filename.dat

Step 2:
If you want to automatically record several data frames, please select **AUTO LIMITED** under **RECORD MORE**.
Enter a time interval for recording under **RECORD-TIME INTERVAL [sec]**, in this example: 1, i.e. a new value is called from the sensor every second).
Then enter the maximum number of values you wish to record in the **RECORD VALUES [MAX 32767]** field.
Please note: Recording can also be stopped earlier by clicking **STOP RECORD**, the data recorded so far will not be lost.

The **TOTAL RECORD TIME** field indicates how long recording will take (in days, hours, minutes, and seconds) if all data are recorded.

Step 3:
By pressing the button **SELECT RECORD FILE** a file can be selected in which the data frame will be stored.
If you select an already existing file name, you will be asked whether you want to overwrite the existing file or not.

Step 4:
Pressing the START RECORD button starts automatic data recording.

The recorder starts to record data, and the button **START RECORD** is red to indicate that recording is active.
The respective data frames are shown in the display windows.

In the two display fields **RECORDED** and **REMAINING** you can check how many data frames have been recorded, and how many frames remain to be recorded.

Please note:
During recording the two input fields RECORD-TIME INTERVAL and VALUES TO BE RECORDED are inactive.

Step 5:
When as many data frames as set under **RECORD VALUES [MAX 32767]** have been recorded, or when the **STOP AUTO RECORD** button is pressed, a pop-up window will appear which confirms that the file is stored.

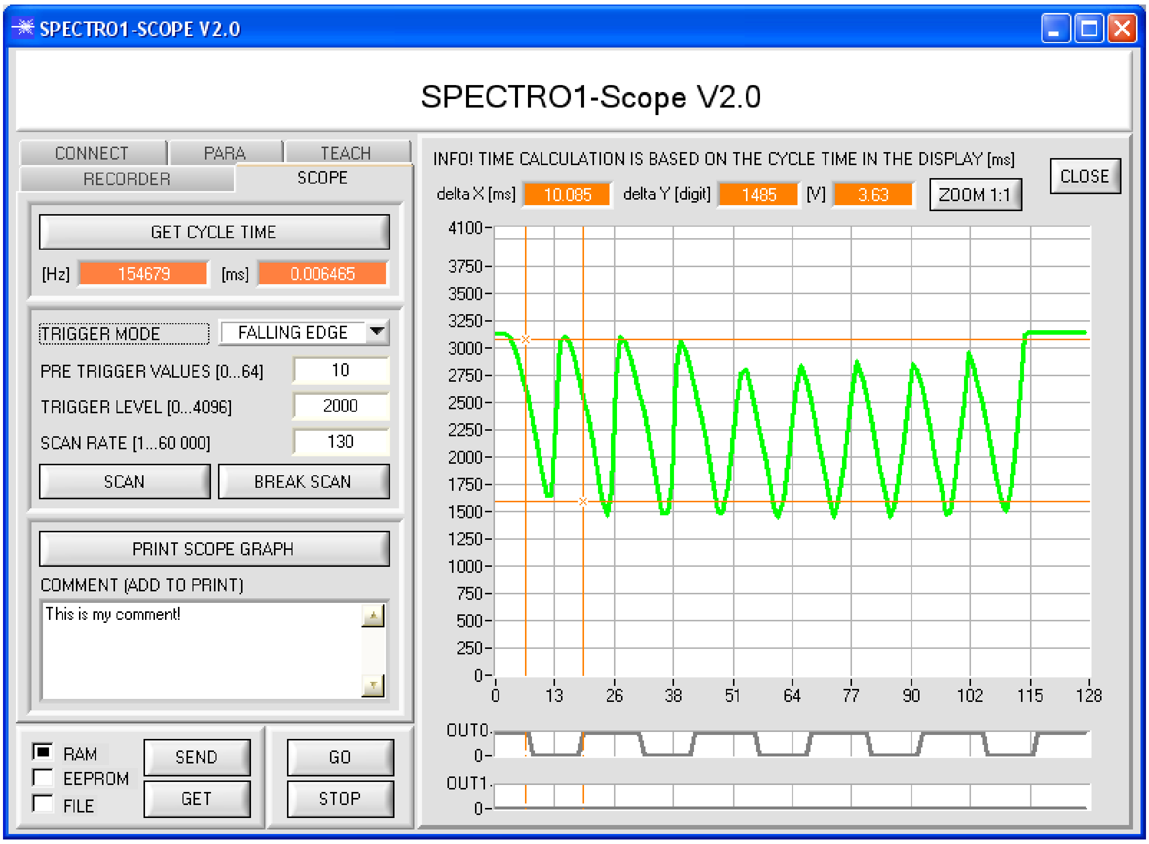
If you want to record an unlimited number of data, select the **AUTO UNLIMITED** function under **RECORD MORE**.
Then select the desired recording interval and press **START RECORD**.

If you want to record data "manually", select the **MANUAL RECORDING** function under **RECORD MORE**.
You can start reading data from the sensor by pressing the **GO** button. These data are visualised in the display window. Pressing the **CAPTURE DATA FRAME** button saves a data frame in the file that was selected under **SELECT RECORD FILE**. The **RECORDED** field shows the sum of the frames already recorded.

Please note:
When you select a file with SELECT RECORD FILE this file is created as a new file. Then a file header with the meaning of the individual columns is written to the file.
When data are then recorded, these data are appended to the selected file, even when data recording is stopped and then resumed again.

2.6 Tab SCOPE

The SCOPE tab visualises an oscilloscope.
A click on **GET CYCLE TIME** displays the current sensor scan frequency in **[Hz]** and **[ms]**. The current scan frequency must be determined for the correct calculation of **deltaX[ms]**. Please give the sensor 8 seconds to determine the correct scan frequency before you click on **GET CYCLE TIME**.
In **TRIGGER MODE = SINGLE SHOT** a click on **SCAN** records a data frame and displays it in the graph.
In **TRIGGER MODE = FALLING EDGE** and **RISING EDGE** a click on **SCAN** starts triggered recording. A trigger start can be defined with **TRIGGER LEVEL**.
SCAN-RATE can be used to delay or accelerate recording. This corresponds with the TIMEBASE function known in oscilloscopes. **PRE TRIGGER VALUES** can be used to define how many values should still be displayed before the actual trigger start.



[Hz] 154679 [ms] 0.006465

TRIGGER MODE: FALLING EDGE

PRE TRIGGER VALUES [0...64]: 10

TRIGGER LEVEL [0...4096]: 2000

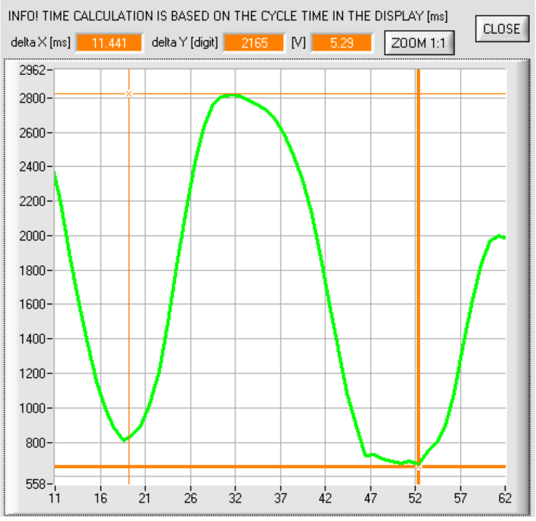
SCAN RATE [1...60 000]: 130

COMMENT (ADD TO PRINT): This is my comment!

☒ RAM
☐ EEPROM
☐ FILE

INFO! TIME CALCULATION IS BASED ON THE CYCLE TIME IN THE DISPLAY [ms]

delta X [ms] 10.085 delta Y [digit] 1485 [V] 3.63



INFO! TIME CALCULATION IS BASED ON THE CYCLE TIME IN THE DISPLAY [ms]

delta X [ms] 11.441 delta Y [digit] 2165 [V] 5.29

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

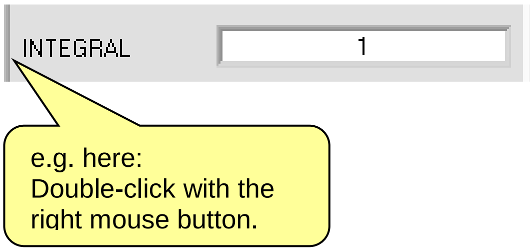
The two orange cursors can be moved with the mouse. The displays **deltaX[ms]**, **deltaY[digit]**, and **deltaY[V]** will be updated.
deltaX[ms] shows the time between the cursors in X direction.
deltaY[digit] or **deltaY[V]** shows the difference between the two cursors in Y direction in digits or in Volt.

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

The two bottom graphs (picture above) represent the states of the two outputs **OUT0** and **OUT1**.

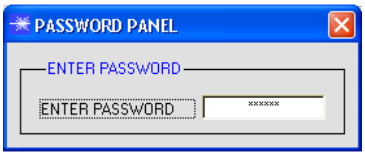
2.7 Offset calibration

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

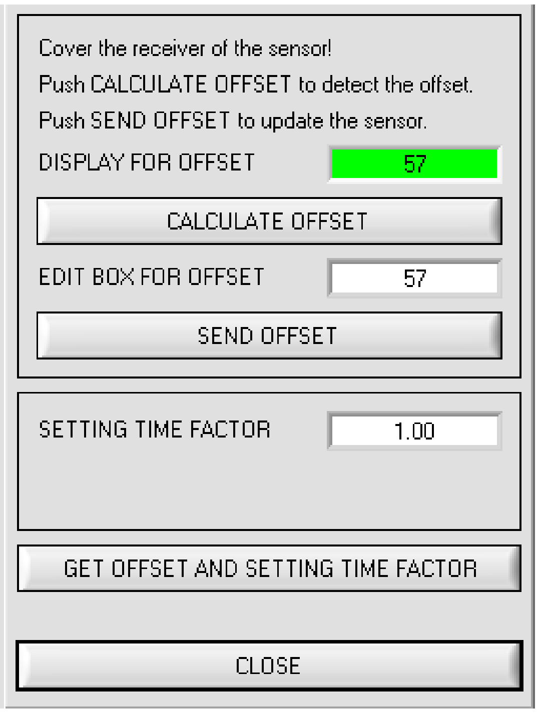

INTEGRAL 1

e.g. here:
Double-click with the
right mouse button.

Offset calibration can be accessed by double-clicking with the right mouse button left of **INTEGRAL** in the **PARA** tab.



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


Cover the receiver of the sensor!
Push CALCULATE OFFSET to detect the offset.
Push SEND OFFSET to update the sensor.

DISPLAY FOR OFFSET 57

EDIT BOX FOR OFFSET 57

SETTING TIME FACTOR 1.00

Then follow the instructions provided in the tab.

ATTENTION!
With offset calibration it is of utmost importance that the receiver does not see any extraneous light.
Therefore cover the receiver of the sensor, e.g. with a black cloth that is impermeable to light.

This is absolutely necessary for proper and perfect offset calibration.

Now press **CALCULATE OFFSET**. The offset value always should be clearly less than 100.
The value automatically is written to the EEPROM of the sensor.

GET OFFSET AND SETTING TIME FACTOR can be used to check the value that is saved as the offset value.
If necessary, **SEND OFFSET** can be used to save an offset value manually. (! not recommended !).

SETTING TIME FACTOR is a value that is defined when the sensor is manufactured.
For sensors with infrared light source this value is 2.5.
For all other sensors this value should be 1.0.
It is possible to change this value, but for safety reasons the procedure to change this value is not described in this manual.
If it should be necessary to change this value, please contact your supplier.

3 Short instructions for the operation of SPECTRO-1 sensors with the SPECTRO1-Scope V2.0 software

These instructions describe how to perform quick teaching of the sensors of SPECTRO-1 series with the **SPECTRO1-Scope V2.0** software interface.

Basically there are 3 methods of switching threshold monitoring. The desired method can be set with the **THRESHOLD MODE** software parameter.

Threshold mode THRESHOLD MODE LOW:
In this mode the **switching threshold** lies below the current reference value. The distance of the switching threshold from the reference value **REF** is defined by the specified **TOLERANCE** value. In this mode the **HYSTERESIS THRESHOLD** lies above the switching threshold. If automatic threshold tracing is activated **(THRESHOLD TRACING=ON)**, appropriate time constants must be selected for threshold tracing.

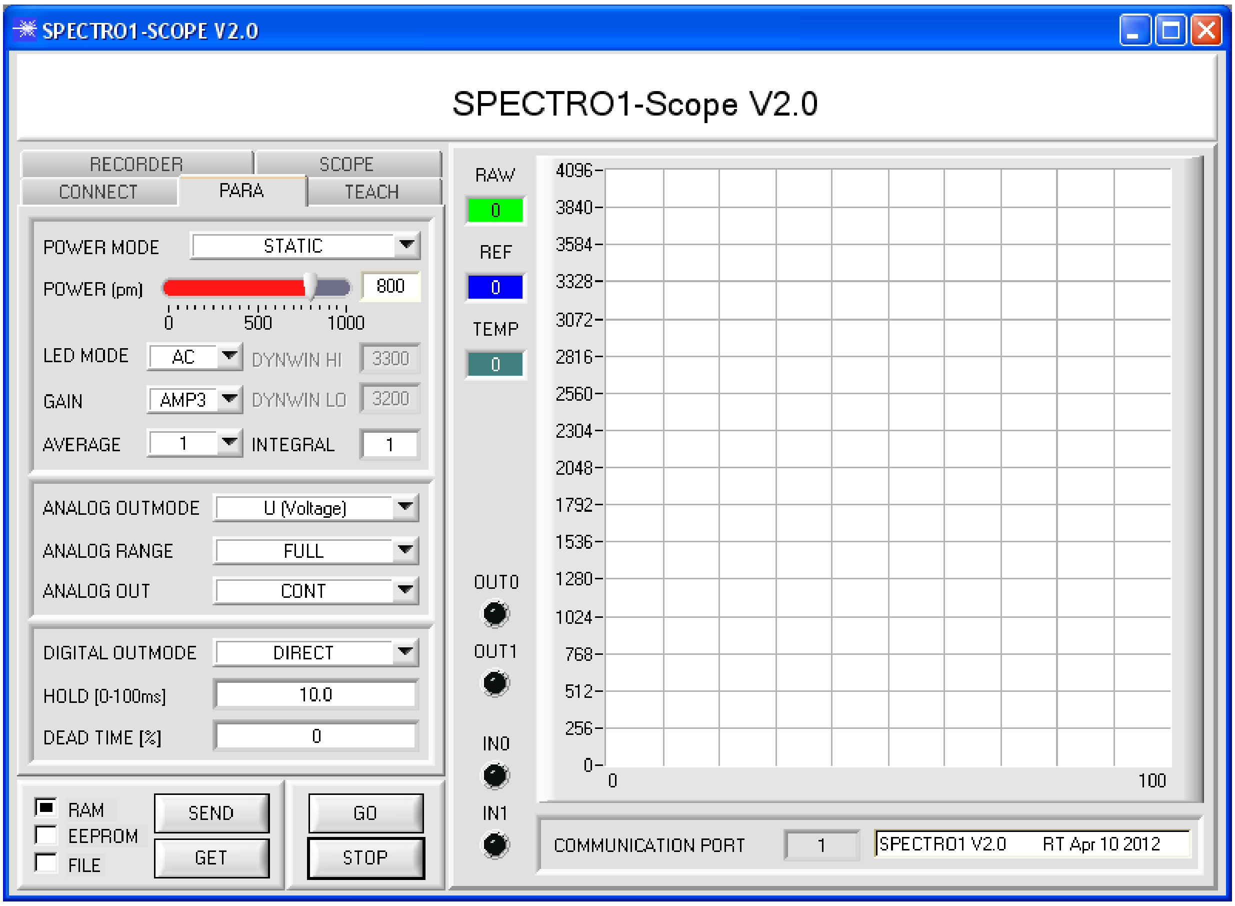
Threshold mode THRESHOLD MODE HI:
In this mode the **switching threshold** lies above the current reference value. The distance of the switching threshold from the reference value **REF** is defined by the specified **TOLERANCE** value. In this mode the **HYSTERESIS THRESHOLD** lies below the switching threshold. If automatic threshold tracing is activated **(THRESHOLD TRACING=ON)**, appropriate time constants must be selected for threshold tracing

Threshold mode THRESHOLD MODE WIN:
This mode operates with two **switching thresholds** that lie symmetrically around the current reference value **REF**. The distance of the switching thresholds from the reference value **REF** is defined by the specified **TOLERANCE** value. In this mode the two **HYSTERESIS THRESHOLDS** lie within the tolerance band. If automatic threshold tracing is activated **(THRESHOLD TRACING=ON)**, appropriate time constants must be selected for threshold tracing.

In the following example parameterisation of the sensor system is performed with these settings: **POWER MODE = STATIC, LED MODE = AC, OUTMODE = DIRECT, THRESHOLD MODE = LOW, THRESHOLD TRACING = OFF, EXTERN TEACH = OFF** and **THRESHOLD CALC = RELATIVE(%)**.

Step 1:
Prior to the use of the software aids (graphic display of sensor signals) the sensor must be manually adjusted as accurately as possible. For this procedure please refer to the data sheet of the respective sensor type.
Make sure that the sensor is properly connected and supplied with power.

Step 2:
Start the SPECTRO1-Scope V2.0 software. Please check whether the status line at the right bottom displays the „SPECTRO1 V2.0 RT:KW*xx/xx* “ message.
Info: Moving the mouse cursor to a control element and clicking with the right mouse button displays a short info on the respective individual control element


SPECTRO1-SCOPE V2.0

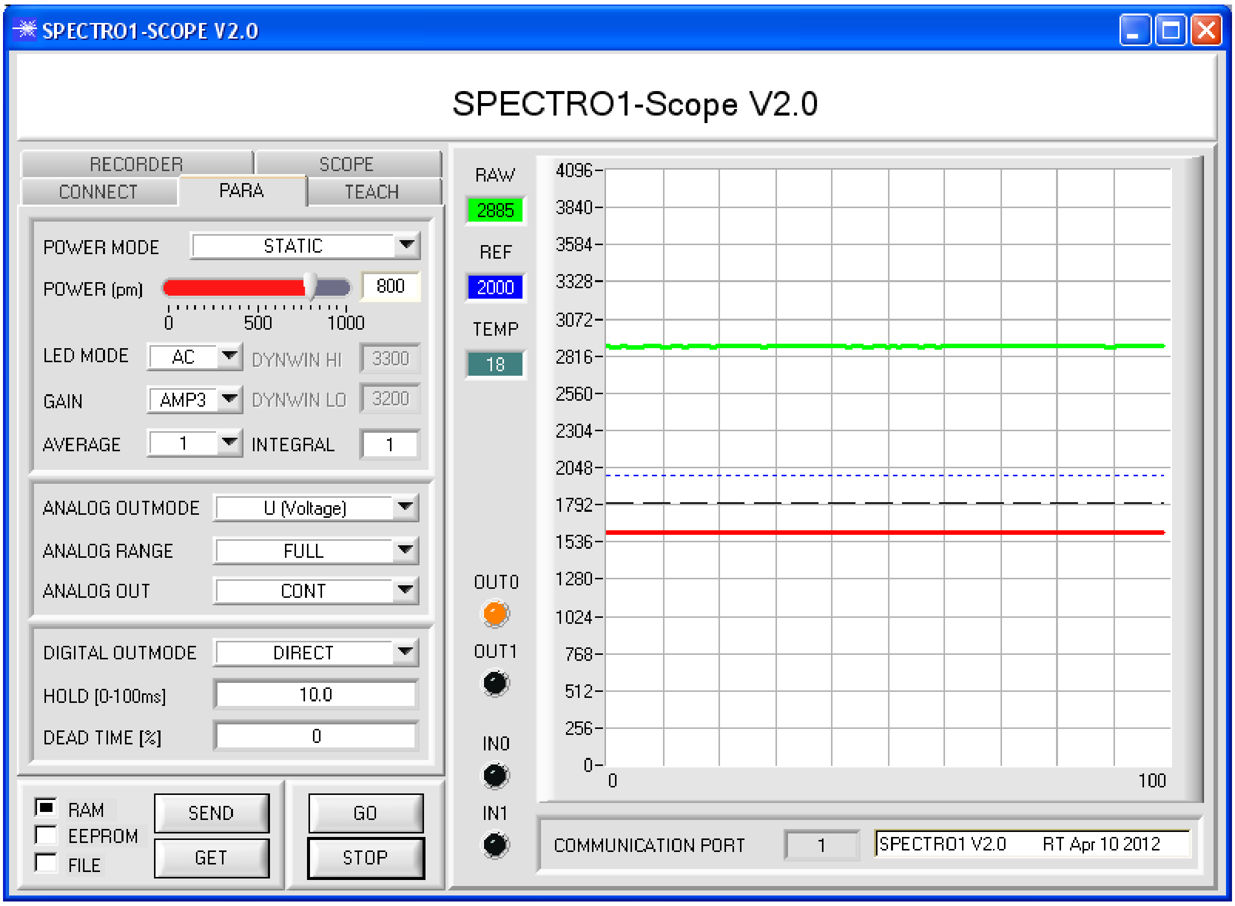
SPECTRO1-Scope V2.0

POWER MODE: STATIC
POWER (pm): 800
LED MODE: AC
DYNWIN HI: 3300
GAIN: AMP3
DYNWIN LO: 3200
AVERAGE: 1
INTEGRAL: 1
ANALOG OUTMODE: U (Voltage)
ANALOG RANGE: FULL
ANALOG OUT: CONT
DIGITAL OUTMODE: DIRECT
HOLD [0-100ms]: 10.0
DEAD TIME [%]: 0
RAW: 0
REF: 0
TEMP: 0
☑ RAM
☐ EEPROM
☐ FILE
COMMUNICATION PORT: 1
SPECTRO1 V2.0 RT Apr 10 2012

Step 3:
Please make sure that for the time being **RAM** and not **EEPROM** is selected for the data exchange with the sensor (RAM is a volatile memory in the sensor, i.e. the data will be lost when power is turned off. EEPROM is a non-volatile memory in the sensor, i.e. the data will not be lost when power is turned off.)
Click on the **TEACH** tab to display the teach parameters.
Now press the **GO** button. Data exchange between sensor and PC will then start. The software shows the measurement value **RAW** in a numeric display element and as a green line in the graph. Adjust the **POWER** value such that **RAW** lies in the upper third of its dynamic range.
Ideally **POWER** and **GAIN** should be set such that **POWER** lies in the range of 500 - 900.

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

Tip! There is a trick for finding a suitable **POWER** value in no time at all. Set **POWER MODE = DYNAMIC**. The sensor then tries to find a suitable **POWER** value such that **RAW** lies between **DYN WIN LO** and **DYN WIN HI**. Check this by pressing **GO**. Press **STOP** when **RAW** has "levelled out". Then press **GET**. The **POWER** value that was found now is shown in the **POWER** function field. Now set **POWER MODE = STATIC**, and press **SEND**.



POWER MODE: STATIC
POWER (pm): 800
LED MODE: AC
DYNWIN HI: 3300
GAIN: AMP3
DYNWIN LO: 3200
AVERAGE: 1
INTEGRAL: 1
ANALOG OUTMODE: U (Voltage)
ANALOG RANGE: FULL
ANALOG OUT: CONT
DIGITAL OUTMODE: DIRECT
HOLD [0-100ms]: 10.0
DEAD TIME [%]: 0
RAW: 2885
REF: 2000
TEMP: 18
COMMUNICATION PORT: 1
SPECTRO1 V2.0 RT Apr 10 2012

Step 4:
Press the **GO** button.

Now press the **TEACH** button. The current **RAW** value is written to the **TEACH VALUE** edit box.
With a click on **SEND** the **TEACH VALUE** is updated as reference value **REF** in the sensor.

Info: If **THRESHOLD TRACING** or **EXTERN TEACH** is not **OFF**, the electronic control unit calculates the reference value **REF** automatically or adopts this value from "teaching" after a corresponding event at IN0. In this case **TEACH VALUE** and **TEACH** are not active on the software interface.

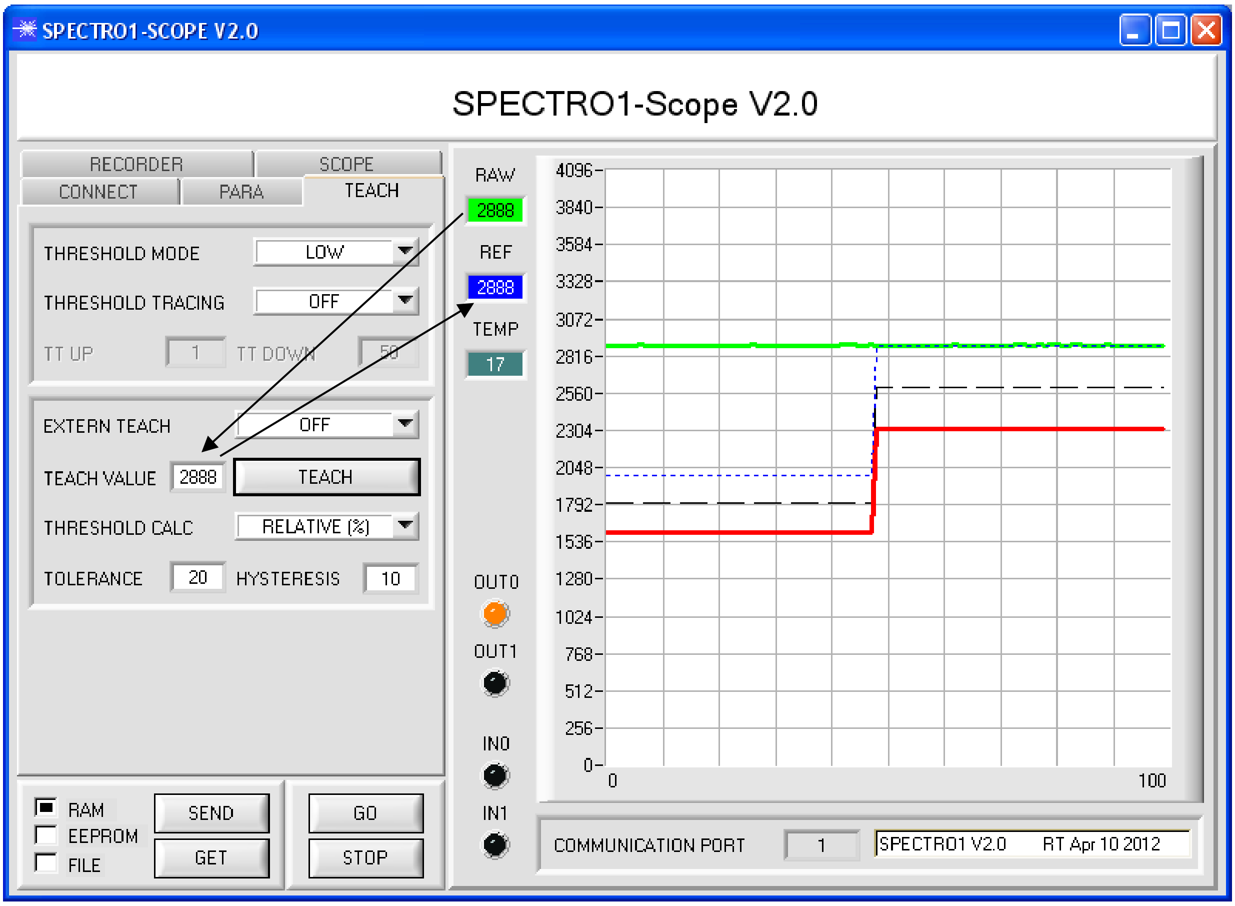
The reference value **REF** (blue line) is used for the calculation of the switching threshold (red line) and hysteresis threshold (black line).
Switching threshold = **REF - (TOLERANCE * REF) / 100** = red line
Hysteresis threshold = **REF- (HYSTERESIS * REF) / 100** = black line

The **TOLERANCE** value must be chosen such that when a faulty object is measured the switching threshold is reliably overshot (or undershot).

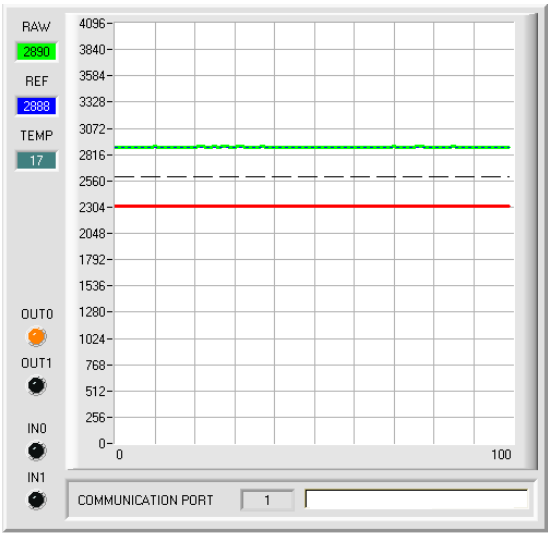
Accept parts should lie in the range between the reference value **REF** and the switching threshold. The value of **TOLERANCE** is determined by a for example production-related variance of accept parts.

The hysteresis threshold prevents a switching of the digital output in case of short-time undershooting of the switching threshold.

Press the **GO** button once again.

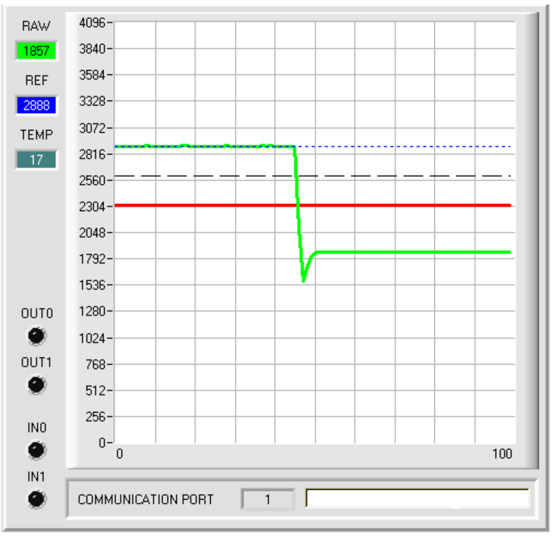


THRESHOLD MODE: LOW
THRESHOLD TRACING: OFF
TT UP: 1
TT DOWN: 50
EXTERN TEACH: OFF
TEACH VALUE: 2888
THRESHOLD CALC: RELATIVE (%)
TOLERANCE: 20
HYSTERESIS: 10
☒ RAM
☐ EEPROM
☐ FILE
RAW: 2888
REF: 2888
TEMP: 17
COMMUNICATION PORT: 1
SPECTRO1 V2.0 RT Apr 10 2012



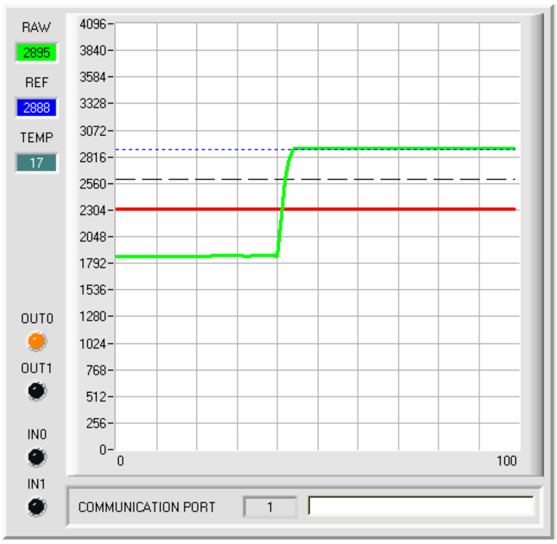
RAW 2890
REF 2888
TEMP 17
COMMUNICATION PORT 1

As long as the current measurement value **RAW** (green line) lies above the switching threshold (red line), the output **OUT0** has +24V (signal is OK). This is visualised by the **OUT0** LED.



RAW 1857
REF 2888
TEMP 17
COMMUNICATION PORT 1

When the current measurement value **RAW** (green line) falls below the switching threshold (red line), the output **OUT0** changes to 0V (error). This is visualised by the **OUT0** LED.



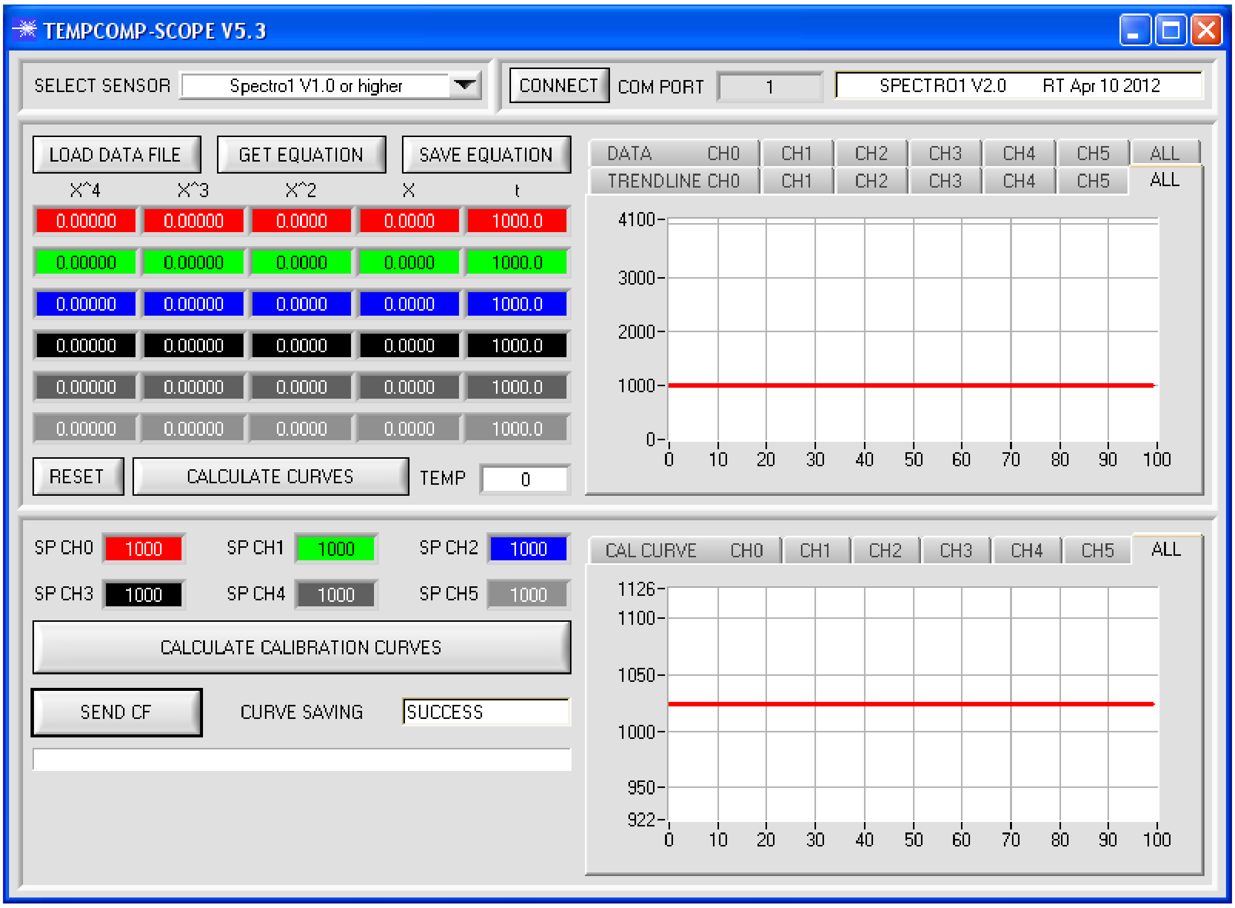
RAW 2895
REF 2888
TEMP 17
COMMUNICATION PORT 1

When the current measurement value **RAW** (green line) rises above the hysteresis threshold (black line) again, the output **OUT0** also changes to +24V again (signal is OK). This is visualised by the **OUT0** LED.

Step 5:
When you have finished parameterisation, please select **EEPROM** and press **SEND** to save the data to the non-volatile memory of the sensor.

4 Operation of the TEMPCOMP-Scope software

If a firmware update should go wrong and the temperature characteristics that are stored in the EEPROM should be lost, these characteristics must be created anew. For this purpose you will need a file with the corresponding data. This file can be obtained from your supplier.
To perform temperature compensation please start the corresponding **TEMPCOMP-Scope software** that is included on the supplied CD. Please make sure that you have a functioning sensor connection. It may be necessary to select the connection with **CONNECT**. Set the correct sensor under **SELECT SENSOR**, if this is not done automatically.



SELECT SENSOR: Spectro1 V1.0 or higher
COM PORT: 1
SPECTRO1 V2.0 RT Apr 10 2012

X^4	X^3	X^2	X	t
0.00000	0.00000	0.0000	0.0000	1000.0
0.00000	0.00000	0.0000	0.0000	1000.0
0.00000	0.00000	0.0000	0.0000	1000.0
0.00000	0.00000	0.0000	0.0000	1000.0
0.00000	0.00000	0.0000	0.0000	1000.0
0.00000	0.00000	0.0000	0.0000	1000.0

TEMP: 0

SP CH0: 1000
SP CH1: 1000
SP CH2: 1000
SP CH3: 1000
SP CH4: 1000
SP CH5: 1000

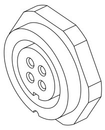
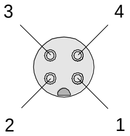
CURVE SAVING: SUCCESS

Step 1:	Load the temperature compensation file that you have received from your supplier with **GET EQUATION** or **LOAD DATA FILE**.
Step 2:	Press **CALCULATE CURVES** to display the data in the graph.
Step 3:	Select the sensor-internal operating temperature (not in °C) that the sensor has at an ambient temperature of 20°, if this has not already been done automatically. The value should be included in the file designation.
Step 4:	Press **CALCULATE CALIBRATION CURVES** to calculate the mean straight line.
Step 5:	Pressing the **SEND CF** button saves the mean straight lines in the **EEPROM** of the sensor.
Step 6:	Temperature compensation is successfully finished if the **SUCCESS** status message is then displayed.

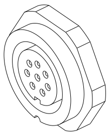
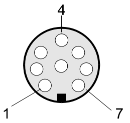
Comment! If you do not immediately have the temperature compensation file at hand, simply start the **TempComp-Scope software**. Establish a connection, if it is not already established, and simply press **SEND-CF**. The sensor then functions as before, but it is not temperature-compensated.

5 Connector assignment of the SPECTRO-1 sensors

Connection of SPECTRO-1 to PC:

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

Connection of SPECTRO-1 to PLC:

8-pole fem. connector (type Binder 712) ***SPECTRO-1/PLC***		 
Pin No.:	**Color of wire: (cab-las8/SPS-fem)**	**Assignment:**
1	white	0V (GND)
2	brown	+24V (± 10 %)
3	green	IN0 (Digital 0: 0 … 1V, Digital 1: +Ub – 10%)
4	yellow	IN1 (Digital 0: 0 … 1V, Digital 1: +Ub – 10%)
5	grey	OUT0 (Digital 0: 0 … 1V, Digital 1: +Ub – 10%)
6	pink	OUT1 (Digital 0: 0 … 1V, Digital 1: +Ub – 10%)
7	blue	IN0 (Digital 0: 0 … 1V, Digital 1: +Ub – 10%)
8	red	NC

6 RS232 communication protocol

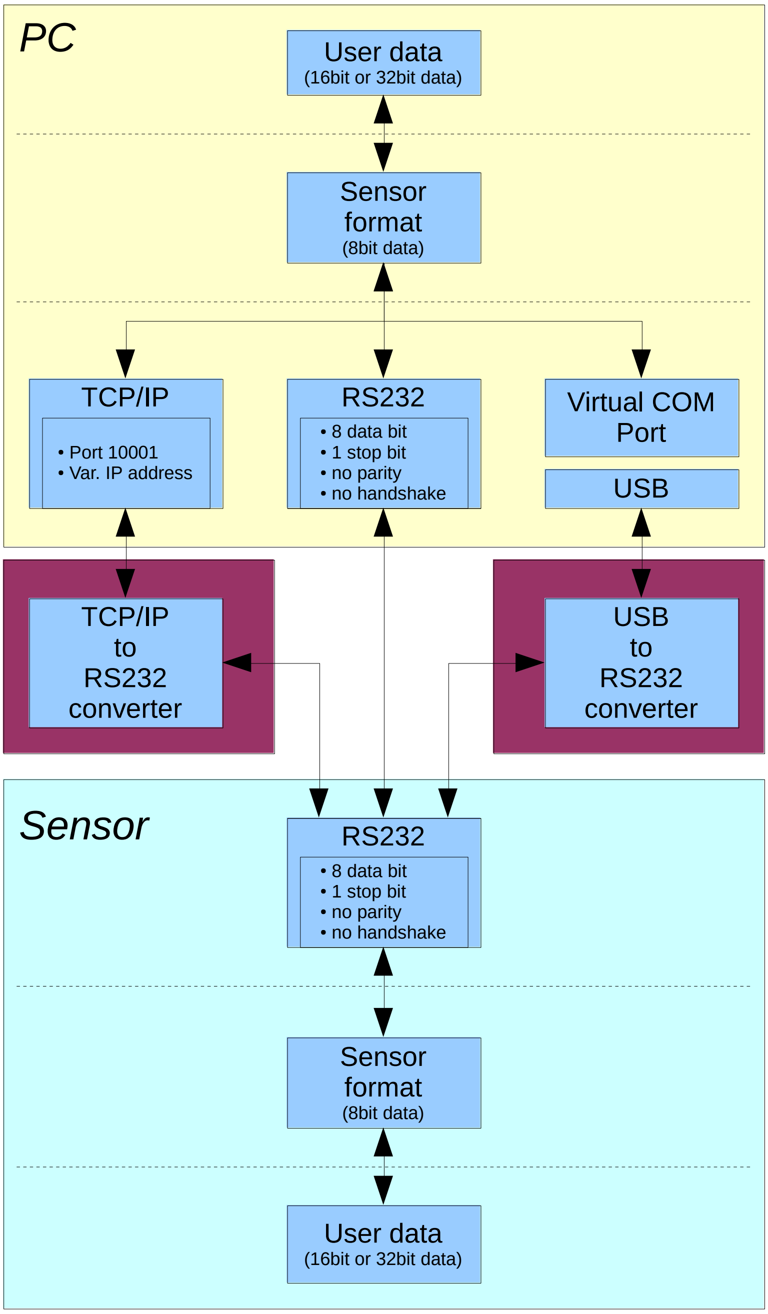
The sensors of the SPECTRO-1 series operate with the following **parameters** that are sent to the sensor or read from the sensor in the stated sequence.
Info! 2 bytes (8bit) are one word (16bit).

Parameter	Type	Meaning
Para1: POWER	Word	Transmitter intensity (0 ... 1000) Attention intensity in thousandth!
Para2: POWER MODE	Word	Transmitter mode: STATIC, DYNAMIC coded to (0, 1)
Para3: DYNWIN LO	Word	Low limit for dynamic window when POWER MODE=dynamic (0…4095)
Para4: DYNWIN HI	Word	High limit for dynamic window when POWER MODE=dynamic (0…4095)
Para5: LED MODE	Word	Control for the internal light source DC, AC, OFF coded to (0,1,2)
Para6: 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)
Para7: AVERAGE	Word	Signal averaging 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384 or 32768
Para8: INTEGRAL	Word	Signal integration (1…250)
Para9: ANALOG OUTMODE	Word	Function of the analog output: OFF, U, I, U+I coded to (0,1,2,3)
Para10: ANALOG RANGE	Word	Function of the analog range: FULL, MIN-MAX when IN0 coded to (0,1)
Para11: ANALOG OUT	Word	Function of the analog out: CONT, RISING EDGE of IN1 coded to (0,1)
Para12: DIGITAL OUTMODE	Word	Function of the digital output: OFF, DIRECT, INVERSE coded to (0,1,2)
Para13: HOLD	Word	Hold time for minimum pulse length coded to (0…1000) [ms]
Para14: THRESHOLD MODE	Word	Select Threshold Mode LOW, HI, WIN coded to (0,1,2)
Para15: THRESHOLD TRACING	Word	Select Threshold Tracing OFF, ON TOL, ON CONT coded to (0,1,2)
Para16: TT UP	Word	Time delay for Threshold Tracing Up (0…60000)
Para17: TT DOWN	Word	Time delay for Threshold Tracing Down (0…60000)
Para18: THRESHOLD CALC	Word	Threshold Calculation ABSOLUTE (digit), RELATIVE (%) coded to (0,1)
Para19: TEACH VALUE	Word	Teach Value (Reference) for threshold calculation (0…4095)
Para20: TOLERANCE	Word	Tolerance Value for threshold calculation (0…4095)
Para21: HYSTERESIS	Word	Hysteresis Value for threshold calculation (0…4095)
Para22: EXTERN TEACH	Word	External teach mode: OFF, DIRECT, DYN, MAX, MIN, (MAX-MIN)/2+MIN coded to (0,1,2,3,4,5)
Para23: 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.

DATA VALUE	Type	Meaning
DatVal1: RAW	Word	Analogue raw signal of the receiver
DatVal2: 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.
DatVal3: REF	Word	Reference value for threshold calculation
DatVal4: TEMP	Word	Sensor internal temperature (not in °C or F)
DatVal5: 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
DatVal6: MIN	Word	Minimum value of raw signal during IN0 was HI
DatVal7: MAX	Word	Maximum value of raw signal during IN0 was HI

Digital serial communication is used for the exchange of data between the software running on the PC and the sensor.
For this purpose the control unit features an EIA-232 compatible interface that operates with the (fixed) parameters "**8 data bits, 1 stop bit, no parity bit, no handshake"**.
Five values are available for the baudrate: 9600baud, 19200baud, 38400baud, 57600baud and 115200baud. As an option the PC software also can communicate through TCP/IP or USB. In these cases transparent interface converters must be used that allow a connection to the RS232 interface.



A proprietary protocol format that organises and bundles the desired data is used for all physical connection variants between PC software and control unit. Depending on their type and function the actual data are 16- or 32-bit variables and represent integer or floating-point values. The protocol format consists of 8-bit wide unsigned words ("bytes"). The actual data therefore sometimes must be distributed to several bytes.

The control unit always behaves passively (except if another behaviour has been specifically activated). Data exchange therefore always is initiated by the PC software. The PC sends a data package ("frame") corresponding to the protocol format, either with or without appended data, to which the control unit responds with a frame that matches the request.

The protocol format consists of two components:
A "header" and an optional appendant ("data").
The header always has the same structure.
The first byte is a synchronisation byte and always is 85_{dez} (55_{hex}).
The second byte is the so-called order byte. This byte determines the action that should be performed (send data, save data, etc.).
A 16-bit value (argument) follows as the third and fourth byte. Depending on the order, the argument is assigned a corresponding value.
The fifth and sixth byte again form a 16-bit value. This value states the number of appended data bytes. Without appended data both these bytes are 0_{dez} or 00_{hex}, the maximum number of bytes is 512.
The seventh byte contains the CRC8 checksum of all data bytes (data byte 1 up to and incl. data byte n).
The eight byte is the CRC8 checksum for the header and is formed from bytes 1 up to and incl. 7.
The header always has a total length of 8 bytes. The complete frame may contain between 8 and 520 bytes.

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)

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: 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 from sensor	order=5
6	Free	
7	Read Firmware String 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

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:

$$X^8+X^5+X^4+X^1$$

To verify the data after they have been received, CRC calculation is performed once again. If the sent and the newly calculated CRC values are identical, the data are without error.
The following pseudo code can be used for checksum calculation:

```
calcCRC8 (data[ ], table[ ])
Input:   data[ ], n data of unsigned 8bit
         table[ ], 256 table entries of unsigned 8bit
Output: crc8, unsigned 8bit

crc8  := $AA_{hex}$
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

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)	1	0	0	46	0	232	122	32	3	0	0
		ARG=0		LEN=46				Para1=800		Para2=0	

Byte13 Data	Byte14 Data	Byte15 Data	Byte16 Data	Byte17 Data	Byte18 Data	Byte19 Data	Byte20 Data	Byte21 Data	Byte22 Data	Byte23 Data	Byte24 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)
128	12	228	12	1	0	3	0	1	0	1	0
Para3=3200		Para4=3300		Para5=1		Para6=3		Para7=1		Para8=1	

Byte25 Data	Byte26 Data	Byte27 Data	Byte28 Data	Byte29 Data	Byte30 Data	Byte31 Data	Byte32 Data	Byte33 Data	Byte34 Data	Byte35 Data	Byte36 Data
Para9 (lo byte)	Para9 (hi byte)	Para10 (lo byte)	Para10 (hi byte)	Para11 (lo byte)	Para11 (hi byte)	Para12 (lo byte)	Para12 (hi byte)	Para13 (lo byte)	Para13 (hi byte)	Para14 (lo byte)	Para14 (hi byte)
1	0	0	0	0	0	1	0	100	0	0	0
Para9=1		Para10=0		Para11=0		Para12=1		Para13=100		Para14=0	

Byte37 Data	Byte38 Data	Byte39 Data	Byte40 Data	Byte41 Data	Byte42 Data	Byte43 Data	Byte44 Data	Byte45 Data	Byte46 Data	Byte47 Data	Byte48 Data
Para15 (lo byte)	Para15 (hi byte)	Para16 (lo byte)	Para16 (hi byte)	Para17 (lo byte)	Para17 (hi byte)	Para18 (lo byte)	Para18 (hi byte)	Para19 (lo byte)	Para19 (hi byte)	Para20 (lo byte)	Para20 (hi byte)
0	0	100	0	100	0	1	0	184	11	20	0
Para15=0		Para16=100		Para17=100		Para18=1		Para19=3000		Para20=20	

Byte49 Data	Byte50 Data	Byte51 Data	Byte52 Data	Byte53 Data	Byte54 Data
Para21 (lo byte)	Para21 (hi byte)	Para22 (lo byte)	Para22 (hi byte)	Para23 (lo byte)	Para23 (hi byte)
10	0	0	0	0	0
Para21=10		Para22=0		Para23=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)	1	0	0	0	0	170	224
		ARG=0		LEN=0			

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

Example order=2: Read parameter from the 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)	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	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)	2	0	0	46	0	232	35	32	3	0	0
		ARG=0		LEN=46				Para1=800		Para2=0	

Byte13 Data	Byte14 Data	Byte15 Data	Byte16 Data	Byte17 Data	Byte18 Data	Byte19 Data	Byte20 Data	Byte21 Data	Byte22 Data	Byte23 Data	Byte24 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)
128	12	228	12	1	0	3	0	1	0	1	0
Para3=3200		Para4=3300		Para5=1		Para6=3		Para7=1		Para8=1	

Byte25 Data	Byte26 Data	Byte27 Data	Byte28 Data	Byte29 Data	Byte30 Data	Byte31 Data	Byte32 Data	Byte33 Data	Byte34 Data	Byte35 Data	Byte36 Data
Para9 (lo byte)	Para9 (hi byte)	Para10 (lo byte)	Para10 (hi byte)	Para11 (lo byte)	Para11 (hi byte)	Para12 (lo byte)	Para12 (hi byte)	Para13 (lo byte)	Para13 (hi byte)	Para14 (lo byte)	Para14 (hi byte)
1	0	0	0	0	0	1	0	100	0	0	0
Para9=1		Para10=0		Para11=0		Para12=1		Para13=100		Para14=0	

Byte37 Data	Byte38 Data	Byte39 Data	Byte40 Data	Byte41 Data	Byte42 Data	Byte43 Data	Byte44 Data	Byte45 Data	Byte46 Data	Byte47 Data	Byte48 Data
Para15 (lo byte)	Para15 (hi byte)	Para16 (hi byte)	Para16 (hi byte)	Para17 (lo byte)	Para17 (hi byte)	Para18 (lo byte)	Para18 (hi byte)	Para19 (lo byte)	Para19 (hi byte)	Para20 (lo byte)	Para20 (hi byte)
0	0	100	0	100	0	1	0	184	11	20	0
Para15=0		Para16=100		Para17=100		Para18=1		Para19=3000		Para20=20	

Byte49 Data	Byte50 Data	Byte51 Data	Byte52 Data	Byte53 Data	Byte54 Data
Para21 (lo byte)	Para21 (hi byte)	Para22 (lo byte)	Para22 (hi byte)	Para23 (lo byte)	Para23 (hi byte)
10	0	0	0	0	0
Para21=10		Para22=0		Para23=0	

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

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

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			

Cycle Time [Hz] = CYCLE COUNT / (COUNTER TIME * 0,0001)

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