

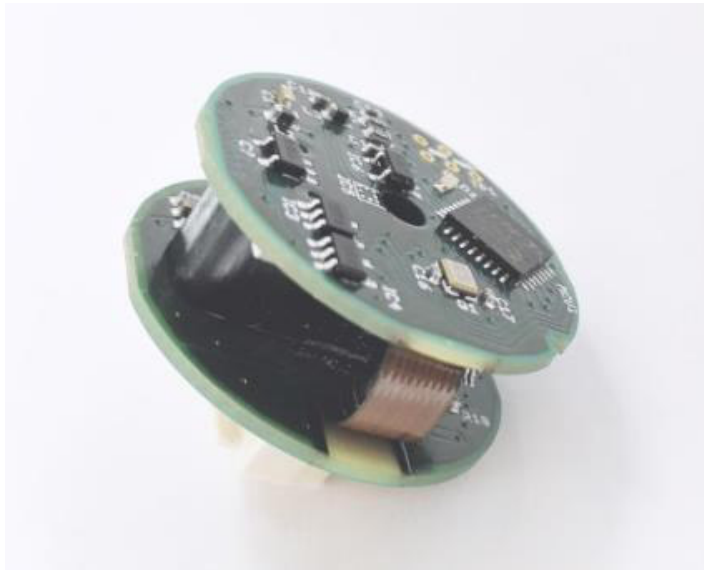
# FD-OEM-02

## OEM-Module with UART-Interface

---

QUICKSTART MANUAL

O2



# FD-OEM-02

## OEM-Module with UART-Interface

---

**Document Version 1.05**

**The device is manufactured by**

**PyroScience GmbH**

Kackertstr. 11  
52072 Aachen  
Germany

Phone +49 (0)241 5183 2210

Fax +49 (0)241 5183 2299

Email [info@pyroscience.com](mailto:info@pyroscience.com)

Web [www.pyroscience.com](http://www.pyroscience.com)

Registered: Aachen HRB 17329, Germany

# TABLE OF CONTENT

<b>Introduction .....</b>	<b>5</b>
<b>Overview .....</b>	<b>7</b>
2.1 Technical Drawing and Electrical Interface .....	7
2.2 Status LED .....	8
2.3 USB interface cable .....	8
<b>Option 1: operating the module with pyro developer tool .....</b>	<b>9</b>
3.1 Installing the software Pyro Developer Tool .....	9
3.2 Using the software Pyro Developer Tool .....	10
<b>Option 2: Simplified Custom Integration .....</b>	<b>11</b>
4.1 Configuring the Module using PyroScience Software .....	11
4.2 Configuration of the Serial Interface.....	11
4.3 Communication Protocol .....	12
4.3.1 General Definitions.....	12
4.3.2 MEA – Trigger Measurement.....	12
4.3.3 CHI – Calibrate oxygen Sensor at ambient air .....	15
4.3.4 CLO – Calibrate oxygen Sensor at 0% (anoxic).....	15
4.3.5 SVS – Save Configuration Permanently in Flash Memory.....	16
4.3.6 #VERS – Get Device Information .....	16
4.3.7 #IDNR – Get Unique ID Number.....	17
4.3.8 #LOGO – Flash Status LED.....	18
4.3.9 #PDWN – Power Down Sensor Circuits .....	18
4.3.10 #PWUP – Power Up Sensor Circuits.....	18
4.3.11 #STOP – Enter Deep Sleep Mode.....	18
4.3.12 #RSET – Reset Device.....	18
4.3.13 #RDUM – Read User Memory .....	19
4.3.14 #WRUM – Write User Memory .....	19
4.3.15 #ERRO – Response if Error Occurred.....	20
4.4 Available Implementations of Communication Protocol .....	21
<b>Option 3: Advanced Custom Integration.....</b>	<b>22</b>
<b>Specifications.....</b>	<b>23</b>

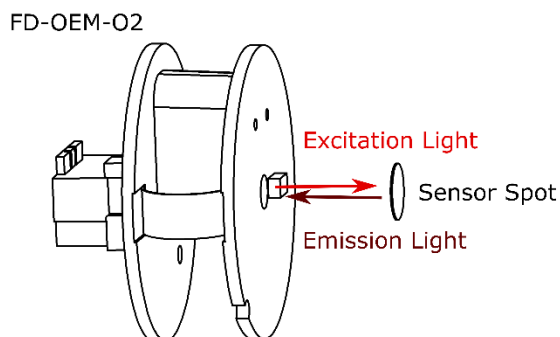
Safety Guidelines ..... 25

# INTRODUCTION

The **FD-OEM-O2** is an optical oxygen sensor module for OEM applications based on luminescence quenching of a sensor dye that is immobilized on a support foil ("sensor spot"). It can be used for gas measurements as well as for dissolved oxygen (DO). The sensor spot is excited with red light, and the properties of the emitted luminescence are measured in the near infrared. The presence of molecular oxygen quenches the luminescence, changing its intensity and lifetime fully reversibly. A distance between the opto-electronics and the sensor spot of up to 20 mm is possible. This gap can be filled with air or any transparent material that has no fluorescence in the NIR (glass, polycarbonate, PMMA, PET, etc.). The measurement principle is very robust. It shows virtually no interferences to other gases, has a very low drift, and the sensor is fully solid-state. The **FD-OEM-O2** features built-in temperature compensation and a digital interface that provides oxygen measurements in engineering units. No additional signal conditioning is necessary. Data is transferred as ASCII text via UART. An USB adapter cable and Windows® software for evaluation is available.

## Features

- High-accuracy measurement in gas or in water (DO)
- Low drift
- Long lifetime
- Fast response
- Digital output of all major oxygen units
- Temperature compensation
- Low power consumption
- Small dimensions
- Lead free



To control the **FD-OEM-02**, there are several options depending on the users' level of experience with optical sensors:

**Option 1:** For evaluation purposes, the module can be operated with the software **Pyro Developer Tool**. It offers simple settings and calibration procedures, as well as basic logging features. Furthermore, additional advanced settings offer full control on all features of the module. This software requires an encoded USB-interface cable (item no. **FD-OEM-USB**) for connecting the module to a Windows (see chapter 0).

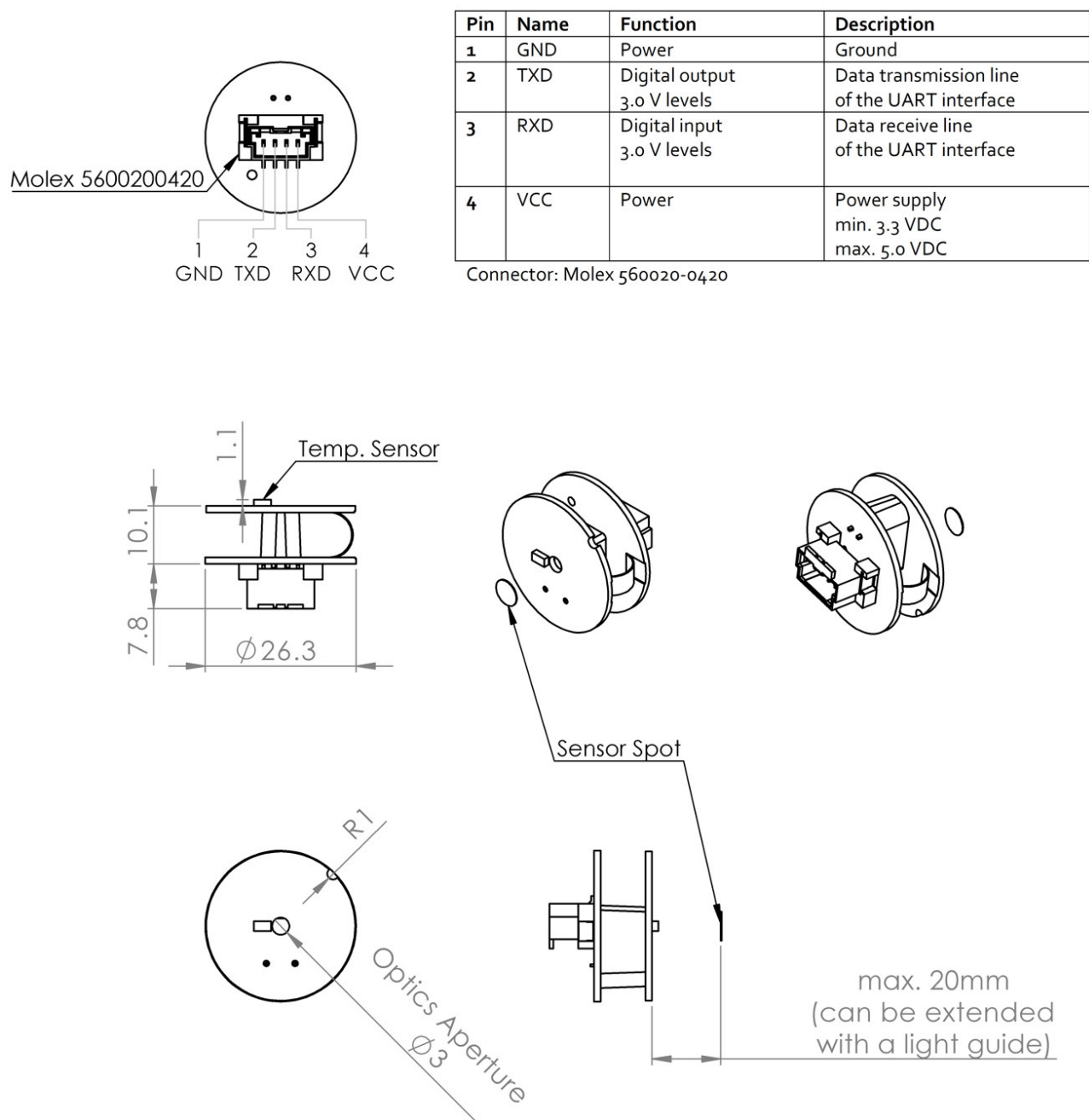
**Option 2:** A simplified custom integration of the module can be realized by adjusting the settings and performing sensor calibrations using the PyroScience software **Pyro Developer Tool** (requires the encoded USB interface cable **FD-OEM-USB**). After closing the software, the configuration is automatically saved within the internal flash memory of the module. The module can then be integrated into a specific setup, and your custom software can perform measurements using a proprietary USB/UART communication protocol (see chapter 4).

**Option 3:** For advanced custom integration the full USB/UART communication protocol is available on request, allowing custom software full control on all settings, calibration and measurement features of the module (see chapter 5).

# OVERVIEW

## 2.1 Technical Drawing and Electrical Interface

The following illustration gives an overview about the electrical interface and the dimensions of the device:



## 2.2 Status LED

The status LED is located on the PCB side where the optics aperture is located. The behavior of the status LED is given in Table 1.

*Table 1: Status LED*

<b>Power-Up</b>	The power supply is switched on.	A correct startup of the module is indicated by 4 flashes within 1-2 seconds.
<b>Active</b>	The module is either in idle mode waiting for a new command, or it is executing a command.	The LED flashes periodically with 1s interval.
<b>Deep sleep</b>	While the power supply is still enabled, the module can be put into deep sleep mode by the #STOP command.	The LED is switched off.
<b>#LOGO-command</b>	The #LOGO-command is sent to the module.	The LED flashes 4 times within 1-2 seconds.

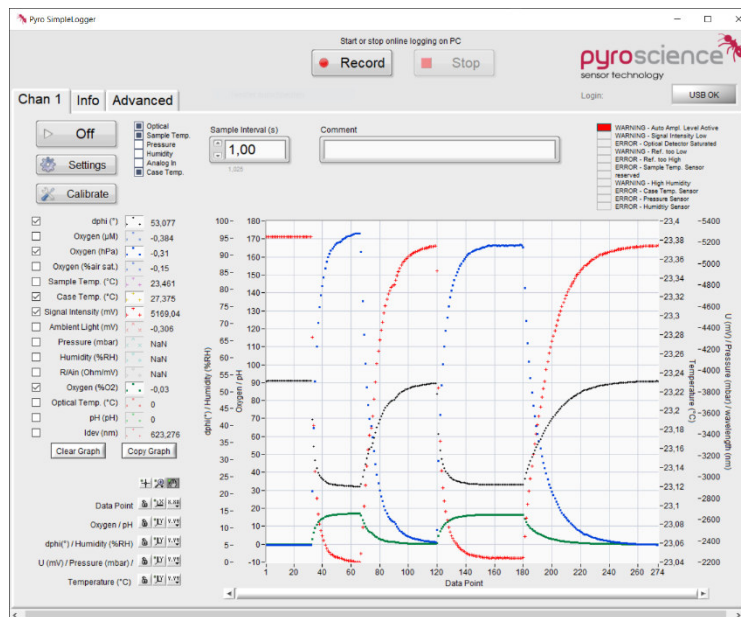
## 2.3 USB interface cable

For the operation of **FD-OEM-02** with a Windows PC, a coded USB interface cable (item no. **FD-OEM-USB**) is available from PyroScience. It includes a license for the software Pyro Developer Tool. Especially for initial testing purposes this software can speed up OEM-developments significantly. Additionally, the USB interface cable **FD-OEM-USB** provides a virtual COM-port. Custom software can use this virtual COM-port for communicating directly with the module based on the communication protocol.



## OPTION 1: OPERATING THE MODULE WITH PYRO DEVELOPER TOOL

For evaluation purposes the module can be operated with the software **Pyro Developer Tool**. It offers simple settings and calibration procedures, as well as basic logging features. Furthermore, additional advanced settings offer full control on all features of the module. This software requires the encoded USB interface cable **FD-OEM-USB** for connecting the module to a Windows PC.



### 3.1 Installing the software Pyro Developer Tool

**System requirements:** PC with Windows 7/8/10 and min. 1000 MB free disk space.

Do not connect the USB-interface cable to your PC before the **Pyro Developer Tool** has been installed. The software will install automatically the appropriate USB-drivers.

#### Installation steps:

- Download the **Pyro Developer Tool** from the downloads tab on [www.pyroscience.com](http://www.pyroscience.com)
- unzip and start the installer and follow the instructions
- connect the interface plug of the USB interface cable the connector X1 of the **FD-OEM-02**
- connect the USB plug to an USB port of the PC. The status LED of the **FD-OEM-02** should flash shortly indicating the correct startup of the module.
- Start the **Pyro Developer Tool** software.

## 3.2 Using the software Pyro Developer Tool

Please refer to the **Pyro Developer Tool manual** for general operation instructions for the software (available on our website).

Please refer to the **oxygen sensor manual** for general information on handling and calibration of the oxygen sensors (available on our website).

## OPTION 2: SIMPLIFIED CUSTOM INTEGRATION

A **simplified custom integration** of the module can be realized by adjusting the settings and performing sensor calibrations using the PyroScience software **Pyro Developer Tool** (requiring the encoded USB interface cable **FD-OEM-USB**). After closing the software, the configuration is automatically saved within the internal flash memory of the module. The module can then be integrated into a specific setup, and your custom software can perform measurements using a proprietary USB/UART communication protocol.

### 4.1 Configuring the Module using PyroScience Software

Please install the Pyro Developer Tool (Chapter 0). Adjust the settings and perform the required calibrations of the sensor.

After the module has been configured, close the PyroScience software. The configuration is automatically saved within the internal flash memory. This means that the adjusted settings and the last sensor calibration are persistent even after a power cycle of the module. Now the module can be integrated into a customer specific setup via its UART interface (or via the USB interface cable with its virtual COM port).

### 4.2 Configuration of the Serial Interface

**FD-OEM-02** is operated via a serial interface, which is realized as a UART interface at 3.0 V levels (max. 3.3 V) consisting of a receive and a transmit line. The configuration of the UART-interface is as follows:

**19200 baud, 8 data bit, 1 stop bit, no parity, no handshake**

Such an UART interface is very common for microcontrollers or microcontroller boards (e.g. Arduino or Raspberry Pi). The module can be directly connected to such UART interfaces without any further interface electronics.

**Note:** The serial interface of this module is **not** an RS232 interface. However, the UART interface can be made compatible to RS232 by integrating an appropriate "level shifter electronics".

## 4.3 Communication Protocol

### 4.3.1 General Definitions

A command always starts with a specific command header (e.g. MEA, #VERS, #LOGO) optionally followed by several input parameters. Input parameters are given as human readable decimal numbers, separated by spaces from each other. Each command must be terminated by a carriage return. If the command could be successfully interpreted by the module, the response is sent back to the master after completion of the requested task. The first part of response consists always of a copy of the original command, optionally appended with output parameters, and again terminated by a carriage return. After a response has been received by the master, the module is immediately ready for receiving the next command. If the internal processing of the received command causes any error within the module, the response will be the error header #ERRO followed by a space and an error code (see below).

Syntax Definitions	
MEA #VERS #LOGO	Examples for a command header
C S R	Examples for place holder for signed integer values transmitted as human readable ASCII strings of decimal numbers. The absolute maximum range of all values transmitted in the communication protocol is from -2147483648 to +2147483647 (signed 32bit integer), if not otherwise indicated.
␣	Space (ASCII code 0x20)
↵	Carriage return (ASCII code 0x0D)

### 4.3.2 MEA – Trigger Measurement

This command triggers a measurement and returns the results.

Command: **MEA\_C\_S↵**

Response: **MEA\_C\_S\_R0\_R1...R17↵**

#### Input Parameters:

C Optical channel number. Set C=1.

S If in doubt, then set S to 47!

This parameter defines the enabled sensor types, given as decimal representation of the following bit field:

Bit 0 (add 1): optical channel
Bit 1 (add 2): sample temperature (typ. the external Pt100-sensor)
Bit 2 (add 4): ambient air pressure

Bit 3 (add 8): relative humidity within the module
Bit 4 (add 16): reserved
Bit 5 (add 32): case temperature (temperature within the module)

Example:  $S = 1 + 2 + 4 + 8 + 32 = 47$  means, that the command will trigger the following measurements: optical channel (oxygen), sample temperature, case temperature, ambient air pressure, and relative humidity within the module housing.

### Output Parameters:

**R<sub>0</sub>** Returns errors and/or warnings of the last measurement as a decimal representation of the following bit field. The user has to distinguish between warnings and errors. A warning indicates, that the measurement results are in principle still valid, but their precision and/or accuracy might be deteriorated. An error means, that the respective measurement result is not at all valid.

Bit 0 (add 1): WARNING - automatic amplification level active
Bit 1 (add 2): WARNING - sensor signal intensity low
Bit 2 (add 4): ERROR - optical detector saturated
Bit 3 (add 8): WARNING - reference signal intensity too low
Bit 4 (add 16): ERROR - reference signal too high
Bit 5 (add 32): ERROR - failure of sample temperature sensor (e.g. Pt100)
Bit 6 (add 64): reserved
Bit 7 (add 128): WARNING high humidity (>90%RH) within the module
Bit 8 (add 256): ERROR - failure of case temperature sensor
Bit 9 (add 512): ERROR - failure of pressure sensor
Bit 10 (add 1024): ERROR - failure of humidity sensor

Example:  $R_0 = 34 = 2 + 32$  means, that there is a warning about low signal intensity of the optical sensor, and that the external temperature sensor (Pt100) had a failure.

If  $R_0 = 0$  then no error or warning appeared.

**R<sub>1</sub>...R<sub>17</sub>** The results of the measurement given as 17 values. The most important result values are highlighted.

	Name	Unit	Description
R1	<i>dphi</i>	m°	Phase shift of optical measurement (raw data)
R2	umolar	0.001 μmol/L	Oxygen level in units of μmol/L (valid only in liquids)
R3	mbar	0.001 mbar	Oxygen level in units of mbar (valid in gases and in liquids)
R4	airSat	0.001 %air sat.	Oxygen level in units of % air saturation (valid only in liquids)

R5	<i>tempSample</i>	0.001 °C	Sample temperature (typ. external Pt100 sensor)
R6	<i>tempCase</i>	0.001 °C	Case temperature (internal T-sensor within module)
R7	<i>signalIntensity</i>	0.001 mV	Signal intensity of the optical measurement
R8	<i>ambientLight</i>	0.001 mV	Ambient light entering the sensor
R9	<i>pressure</i>	0.001 mbar	Ambient air pressure
R10	<i>humidity</i>	0.001 %RH	Relative humidity within the module housing
R11	<i>resistorTemp</i>	0.001 Ohm	Resistance of the temperature sensor (raw data)
R12	<i>percentO2</i>	0.001 %O2	Oxygen level in units of %O2 (valid only in gases)
R13- R17	<i>-reserved-</i>		

This command is the essential command for triggering measurements. If the input parameter *S* is requesting several sensor types to be measured, the optical oxygen measurement ("optical channel") is always performed as the last measurement. This ensures that for enabled automatic temperature compensation the sample temperature measurement (typ. external Pt100) is done before it is then used for compensating the oxygen measurement.

**IMPORTANT:** If automatic temperature compensation is enabled for the optical sensor, it is mandatory to enable Bit1 of the input parameter *S*!

The output parameters *umolar*, *mbar*, *airSat*, *percentO2*, and *tempSample* give the results of the oxygen measurement and of the temperature measurement (typ. external Pt100).

The output parameter *signalIntensity* is a measure of the signal quality ("signal intensity") of the connected optical sensor. As a rule of thumb, typical values will be in the range of 20-500 mV. Low signal intensities (<50 mV) might lead to noisy oxygen measurements. A low signal intensity might be an indicator that the sensor is not configured optimally and/or that the sensor is "worn out"/depleted and has to be replaced. Please note, that the signal intensity is also dependent on the actually measured oxygen value. Low oxygen values have a high signal intensity, high oxygen values have a lower signal intensity.

The output parameter *ambientLight* is a measure how much ambient infrared light is entering the oxygen sensor. In principle, such ambient light is not influencing the oxygen measurement. However, excess ambient light might lead to a saturation of the optical detector (indicated by an enabled ERROR Bit2 in  $R_0$ ), which will lead to an invalid oxygen measurement. As a rule of thumb, the sum of *signalIntensity* and *ambientLight* should be kept below ca. 2000 mV (the optical detector saturates around 2500 mV).

**Example Communication:****Command:** MEA\_1\_3↵**Response:** MEA\_1\_3\_0\_30120\_270013\_210211\_98007\_20135\_0\_87016\_11788\_0\_0\_123022\_20980\_0\_0\_0\_0\_0↵

This example command triggers the measurement of the sample temperature (typ. external Pt100) and of the optical oxygen sensor. The highlighted output parameters of the shown example response are interpreted as follows:

$R_0 = 0 \rightarrow$  No error or warning occurred; the measurement is valid!

umolar = 270.013  $\mu\text{mol/L}$

mbar = 210.211 mbar

airSat = 98.007 % air sat.

tempSample = 20.135  $^{\circ}\text{C}$

signalIntensity = 87.016 mV

ambientLight = 11.788 mV

percentO2 = 20.980 %O2

**4.3.3 CHI - Calibrate oxygen Sensor at ambient air**

This command is used for calibrating the upper calibration point of the oxygen sensor at ambient air.

**Command:** CHI\_C\_T\_P\_H↵**Response:** CHI\_C\_T\_P\_H↵**Input Parameters:**

- C** Optical channel number. Set C=1
- T** Temperature of the calibration standard in units of  $10^{-3} ^{\circ}\text{C}$  (e.g. 20000 means  $20^{\circ}\text{C}$ )
- P** Ambient air pressure in units of  $10^{-3}$  mbar (e.g. 1013000 means 1013 mbar)
- H** Relative humidity of the ambient air in units of  $10^{-3}$  %RH (e.g. 50000 means 50%RH) Set H=100000 (=100%RH) for calibrations in air saturated water.

This command performs 16 repeated optical measurements, and uses the average for the calibration. The total duration for this procedure varies between ca. 3s and ca. 6s depending on the configuration of the module. In order to keep the calibration permanently even after a power cycle, the command SVS must be executed afterwards.

**4.3.4 CLO - Calibrate oxygen Sensor at 0% (anoxic)**

This command is used for calibrating the lower calibration point of the oxygen sensor at 0% O2.

**Command:** `CLO_C_T↵`

**Response:** `CLO_C_T↵`

#### Input Parameters:

- C Optical channel number. Set C=1
- T Temperature of the calibration standard in units of  $10^{-3} \text{ }^{\circ}\text{C}$  (e.g. 20000 means  $20^{\circ}\text{C}$ )

This command performs 16 repeated optical measurements, and uses the average for the calibration. The total duration for this procedure varies between ca. 3s and ca. 6s depending on the configuration of the module. In order to keep the calibration permanently even after a power cycle, the command SVS must be executed afterwards.

### 4.3.5 SVS – Save Configuration Permanently in Flash Memory

This command is used for storing the current configuration in the flash memory:

**Command:** `SVS_C↵`

**Response:** `SVS_C↵`

#### Input Parameters:

- C Optical channel number. Set C=1

Saves the actual settings and calibration as the new default values into the internal flash memory. These default values are automatically loaded after a power cycle.

#### Example Communication:

**Command:** `SVS_1↵`

**Response:** `SVS_1↵`

### 4.3.6 #VERS – Get Device Information

This command returns general information about the device.

**Command:** `#VERS↵`

**Response:** `#VERS_D_N_R_S_B_F↵`

#### Output Parameters:

- D Device ID, identifies the specific device type. For the FD-OEM-02 the device ID is always 8.
- N Number of optical channels. For the FD-OEM-02 this value is 1.
- R Firmware version, e.g. R=403 designates firmware version 4.03
- S Bit field about available sensor types and supported optical analytes as follows:



Bit 0-7: Available Sensor Types	Bit 8-15: Supported Optical Analytes
Bit 0: optical channel(s)	Bit 8: oxygen
Bit 1: sample temperature (typ. Pt100)	Bit 9: optical temperature
Bit 2: pressure	Bit 10: pH
Bit 3: humidity	Bit 11: CO2
Bit 4: analog in	Bit 12: reserved
Bit 5: case temperature	Bit 13: reserved
Bit 6: reserved	Bit 14: reserved
Bit 7: reserved	Bit 15: reserved

Example:  $S = 1 + 2 + 4 + 8 + 32 + 256 = 303$  means, that the device provides an optical channel as well as sample and case temperature, pressure, and humidity sensors, and the optical channel supports the analytes oxygen.

**B** Firmware build number starting at 1 for each firmware version (reflects minor firmware revisions which normally do not require a software or firmware update for the user)

**F** Bit field about available features as follows:

Bit 0: analog out 1	Bit 5: battery
Bit 1: analog out 2	Bit 6: stand-alone logging
Bit 2: analog out 3	Bit 7: sequence commands
Bit 3: analog out 4	Bit 8: user memory
Bit 4: user interface (display, buttons)	Bit 9-31: reserved

Example:  $F = 1 + 2 + 4 + 8 + 256 = 271$  means that 4 analog outputs are supported and the module possesses a user memory. Note, the optional analog outputs require additional hardware (more information on request).

### Example Communication:

Command: `#VERS_1↵`

Response: `#VERS_1_4_403_1071_2_271↵`

### 4.3.7 #IDNR – Get Unique ID Number

This command returns the unique identification number of the respective device.

Command: `#IDNR↵`

Response: `#IDNR_N↵`

#### Output Parameters:

**N** Unique ID number. Note, this parameter is given as an unsigned 64 bit integer!

Returns the unique identification number of the device (does NOT correspond to the serial number of the device).

### Example Communication:

Command: `#IDNR↵`

Response: `#IDNR_2296536137892833272↵`

### 4.3.8 #LOGO – Flash Status LED

This command lets the status LED flash for 4 times within ca. 1 s.

Command: **#LOGO**↵

Response: **#LOGO**↵

This command can be used to check proper communication with the device. Or it might be helpful in setups with more than one device, in order to identify which COM port is connected to which device.

### 4.3.9 #PDWN – Power Down Sensor Circuits

This command switches off the power supply of the sensor circuits.

Command: **#PDWN**↵

Response: **#PDWN**↵

This command can be used for some power saving during idle operation periods. Note, that the sensor circuits are automatically powered up again, if the module receives any command (e.g. MEA) requiring a sensor measurement. This is also the case if a broadcast measurement takes place.

### 4.3.10 #PWUP – Power Up Sensor Circuits

This command switches on the power supply of the sensor circuits.

Command: **#PWUP**↵

Response: **#PWUP**↵

The wake-up duration is up to 250 ms.

### 4.3.11 #STOP – Enter Deep Sleep Mode

This command puts the device into a deep sleep mode with very low power consumption

Command: **#STOP**↵

Response: **#STOP**↵

During deep sleep mode the device has very low power consumption. No standard communication via USB/UART is possible. The deep sleep mode can be exit by sending a <CR> (0x0D) to the device. The device will respond then also with a single <CR> indicating that it is ready to receive new commands. The wake-up duration can be up to 250 ms.

### 4.3.12 #RSET – Reset Device

This command triggers a reset of the device.

Command: **#RSET**↵

Response: **#RSET**↵

Triggers a reset of the device, as if the device experienced a power cycle.

### 4.3.13 #RDUM – Read User Memory

This command reads values from the user memory registers.

Command: **#RDUM\_R\_N**↵

Response: **#RDUM\_R\_N\_Y<sub>0</sub>...Y<sub>N</sub>**↵

#### Input Parameters:

**R** Address of first register to be read from the user memory (0...63)

**N** Total number of registers to be read (1...64)

#### Output Parameters:

**Y<sub>0</sub>...Y<sub>N</sub>** Content of the requested user memory registers (signed 32bit integers).

The device offers a user memory of altogether 64 signed 32bit integer numbers (range - 2147483648 to 2147483647) which is located in the flash memory and is therefore retained even after power cycles. This read command returns  $N$  ( $N=1...64$ ) consecutive values  $Y_1 \dots Y_N$  from the user memory starting at the user memory address  $R$  ( $R=0...63$ ). Note, that  $N+R$  must be  $\leq 64$ . The content of the user memory has no influence on the module itself. It can be used for any user specific purpose.

#### Example Communication:

Command: **#RDUM\_12\_4**↵

Response: **#RDUM\_12\_4\_-40323\_23421071\_0\_-555**↵

This example shows a command which requests the values of 4 consecutive beginning with the user memory address 12.

### 4.3.14 #WRUM – Write User Memory

This command writes values into the user memory registers.

Command: **#WRUM\_R\_N\_Y<sub>0</sub>...Y<sub>N</sub>**↵

Response: **#WRUM\_R\_N\_Y<sub>0</sub>...Y<sub>N</sub>**↵

#### Input Parameters:

**R** Address of first user memory register to be written (0...63)

**N** Total number of registers to be written (1...64)

**Y<sub>0</sub>...Y<sub>N</sub>** Values to be written to the user memory registers (signed 32bit integers).

This command writes  $N$  ( $N=1...64$ ) values  $Y_1 \dots Y_N$  consecutively starting at the user memory address  $R$  ( $R=0...63$ ). Note, that  $N+R$  must be  $\leq 64$ . This command must be used economically, because the flash memory is designed for typ. max. 20000 flash cycles. Each time this command is executed, it will trigger a flash cycle.

#### Example Communication:

Command: **#WRUM\_0\_2\_-16\_777**↵

Response: **#WRUM\_0\_2\_-16\_777**↵

This example shows a command which writes the value -16 into the memory address 0, and the value 777 into the memory address 1.

#### 4.3.15 #ERRO - Response if Error Occurred

If an error occurred, the device will give the following response:

Command: *any command*

Response: **#ERRO\_C** ←

This error response is mostly given, if the master did not send the command with the correct communication syntax. The output parameter C represents the general PyroScience error types as given by the following table.

Note: Warnings and errors directly related to the sensor measurements (e.g. a broken Pt100 temperature sensor, or a "worn out" optical oxygen sensor) will not result in such an #ERRO response. Instead, such warning and errors are given in the output parameter R<sub>0</sub> of the MEA command (see above).

C	Error Type	Description
-1	<i>General</i>	A non-specific error occurred.
-2	<i>Channel</i>	The requested optical channel does not exist.
-11	<i>Memory Access</i>	Memory access violation either caused by a not existing requested register, or by an out of range address of the requested value.
-12	<i>Memory Lock</i>	The requested memory is locked (system register) and a write access was requested.
-13	<i>Memory Flash</i>	An error occurred while saving the registers permanently. The SVS request should be repeated to ensure a correct permanent memory.
-14	<i>Memory Erase</i>	An error occurred while erasing the permanent memory region for the registers. The SVS request should be repeated.
-15	<i>Memory Inconsistent</i>	The registers in RAM are inconsistent with the permanently stored registers after processing SVS. The SVS request should be repeated.
-21	<i>UART Parse</i>	An error occurred while parsing the command string. The last command should be repeated.
-22	<i>UART Rx</i>	The command string was not received correctly (e.g. device was not ready, last request was not terminated correctly). Repeat the last command.
-23	<i>UART Header</i>	The command header could not be interpreted correctly (must contain only characters from A-Z). Repeat the last command.

-24	<i>UART Overflow</i>	The command string could not be processed fast enough to prevent an overflow of the internal receiving buffer
-25	<i>UART Baudrate</i>	The requested baudrate is not supported. No baudrate change took place.
-26	<i>UART Request</i>	The command header does not match any of the supported commands.
-27	<i>UART Start Rx</i>	The device was waiting for incoming data; however, the next event was not triggered by receiving a command.
-28	<i>UART Range</i>	One or more parameters of the command are out of range.
-30	<i>I2C Transfer</i>	There was an error transferring data on the I2C bus.
-40	<i>Temp Ext</i>	The communication with the sample temperature sensor was not successful.
-41	<i>Periphery No Power</i>	The power supply of the device periphery (sensors, SD card) is not switched on.

## 4.4 Available Implementations of Communication Protocol

We offer libraries for controlling **FD-OEM-02** using LabView programming language. The libraries and corresponding documentation are free for download from our website.

## OPTION 3: ADVANCED CUSTOM INTEGRATION

For advanced custom integration the full USB/UART communication protocol is available on request, allowing custom software full control on all settings, calibration and measurement features of the module.

## SPECIFICATIONS

### Analytical Performance (with OXSP5 Sensor Spot)

Oxygen unit	hPa	%O <sub>2</sub> *	%air sat.*	mg/L
Medium	gas/water	gas	water	water
Typ. measuring range	0-500	0-50	0-250	0-23
Max. measuring range	0-2000	0-200	0-500	0-50
Detection limit	0.1	0.01	0.05	0.005
Resolution at 10 hPa	0.1	0.01	0.05	0.005
Resolution at 200 hPa	0.5	0.05	0.25	0.025
Accuracy at 10 hPa (10°C - 40°C)	±0.2	±0.02	±0.1	±0.01
Accuracy at 200 hPa (10°C - 40°C)	±2	±0.2	±1	±0.1
Accuracy at 10 hPa (-10°C - 60°C)	±1	±0.1	±0.5	±0.05
Accuracy at 200 hPa (-10°C - 60°C)	±10	±1	±5	±0.5
Response time (t <sub>90</sub> )	< 15 sec (water), <5 sec (gas)**			
Drift @ 25°C	<1% O <sub>2</sub> / year at 20% O <sub>2</sub>			
Minimum lifetime	>50,000,000 measurements			
Storage life	>5 years in darkness at 20°C			
Warm-up time	3 min (reduced accuracy during warm-up)			

\* at 1013 mbar ambient gas pressure

\*\* faster response times on request

### Environment

Temperature range during operation	-10 to 60°C
Temperature range during storage	-40°C to 60°C
Humidity	Non-condensing
Maximum absolute pressure	20 bar

### Interface

Supply Voltage	3.3 - 5.0 V DC
Standby Current	typ. 8 mA at 25°C
Current in "Power Down Mode"	typ. 2.5 mA at 25°C
Current in "Deep Sleep Mode"	typ. 25 µA at 25°C
Energy Consumption per Measurement	ca. 1-2 mAs at 25°C
Communication Interface	3.0 V UART (max. 3.3 V)
Connector	Molex 560020-0430

Mechanical

Dimensions	Ø 26,3 mm x 19 mm
Weight	5 g
Distance Optics - Sensor Spot	max. 20 mm, may be extended with light guide
Conformity to RoHS directive	RoHS compliant, lead free



## SAFETY GUIDELINES

Before using the device and its sensors, read carefully the instructions and user manuals.

In case of problems or damage, disconnect the instrument and mark it to prevent any further use. Consult PyroScience for advice. There are no serviceable parts inside the device. Please note that opening the housing will void the warranty.

The device is not watertight. The device should be kept under dry and clean conditions, avoiding moisture, dust, corrosive conditions and excessive heating of the instrument (e.g. direct sun light).

Calibration and application of the sensors is on the user's authority, as well as data acquisition, treatment and publication.

The sensors and the device are not intended for medical, aerospace, or military purposes or any safety-critical applications.

The sensors should be used in the laboratory by qualified personnel only, following the user instructions and the safety guidelines of the manual, as well as the appropriate laws and guidelines for safety in the laboratory.

Keep the sensors and the device out of reach of children.

## CONTACT

**PyroScience GmbH**

Kackertstr. 11  
52072 Aachen  
Deutschland

Tel.: +49 (0)241 5183 2210

Fax: +49 (0)241 5183 2299

[info@pyroscience.com](mailto:info@pyroscience.com)

[www.pyroscience.com](http://www.pyroscience.com)