Version / Release

erstellt von / created by	Datum / date	Name / name	Unterschrift / signature*
R&D		Christoph Saller	
Freigegeben von / approved by	**Datum / date**	**Name / name**	**Unterschrift / signature***
R&D		Jacek Smeda	
Product Manager		Christian Huber	
QMB		Claudia Schröder	
Head of R&D		Achim Stangelmayer	

* Signatures are not necessary for any drafts

Verteiler / Distribution List

Nr. / No.	Beschreibung / description	Empfänger / recipient
1	Original with signature	R&D Electronics

Anlage / Attachment List

Nr. / No.	Beschreibung / description	Version / version

Dokumenthistorie / Revision History

Revision / release	Datum / date	Dateiname / file name	Grund für Revision / reason for revision	Geändert von / changed by	Freige-geben von / approved by
dv1	25.9.2013	140911_chsa_PCP-PG2-01_dv4.doc	First Release	ChSa	JaSm
dv2	27.2.2014	140911_chsa_PCP-PG2-01_dv4.doc	PSt7/PSt8 added. New commands „clun“ and „sckv“	ChSa	ChHu
dv3	4.6.2014	140911_chsa_PCP-PG2-01_dv4.doc	Connector configuration added. Maximum write cycles corrected to 10.000	ChSa	ChHu

dv4	11.9.2014	140911_chsa_PCP-PG2-01_dv4.doc	New commands: mmwr, mmer walp; write cycle count added to report.	ChSa	ChHu
dv5	24.10.2014	141024_chsa_PCP-PG2-01_dv5.doc	Range of commands scks and sckv corrected	ChSa	ChHu
Dv6	24.10.2014	141028_chsa_PCP-PG2-01_dv6.doc	Firmware Version updated due to minor bug fixed	ChSa	ChHu

Contents

1 Purpose

This document describes communication protocol compatible with PG2-O2 devices and firmware version PGT1.0.0.8. The user should contact PreSens GmbH support directly in case of questions about this communication protocol.

2 General Information

The interface of the PG2-O2 allows for data exchange between a Host unit and OEM DO modules. After a proper initialization the host can freely exchange data with PG2, sending the communication commands and receiving information data strings. Therefore the host must always take care about baud rate, timing and amount of data send.
The PG2-O2 interface consists of RS232 full-duplex driver working at baud rate 19200bps. The protocol is 8 data bits long with no parity, no handshakes and 1 stop bit. The interface uses standard 7-Bit-ASCII-Code compatible with DIN 66003 norms for commands syntax.

3 Communication protocol

3.1 Principles

The data communication between the host and PG2-O2 unit is based on point-to-point. The master and slave role changes according to the operation mode programmed. In general the host is a master; it starts the communication chain and receives required data from slave (PG2-O2). The one exception is when the PG2-O2 is set to continuous mode (see Operating Modes for details). Here, the data string is being sent continuously to the host, which must act as a slave.
There are two data exchange ways: host to PG2 and PG2 to host. Since the input buffer of a single PG2 limited to 32 characters (including special chars), the host must take care not to send too many data at a time. A certain time delay must be used to avoid the data string to be lost and ignored by PG2 (see Timing specification). To ensure a proper data exchange , you may activate acknowledge and not-Acknowledge Messages. All interface commands consist of four letter message code and optional four or seven digit value code terminated with carriage return (CR) code.

NOTE: All codes are case sensitive.

Message codes are unique and described in detail in the following section. Additionally the query command can be send from the host in order to receive parameter value. The query command consists of the massage code plus question mark and CR.

There are two modes for communication

Format String in Mode 1, suitable for H2M (Human-to-Machine) communication in a command line interface (e.g. hyperterminal)

Data		Delimiter
Command Code	**Command dependent data**	**CR**

Format String in Mode 2, suitable for M2M (Machine-to-Machine)-communication:

Data		Checksum	Delimiter
Command Code	**Command depended data**	**2bytes**	**CR**

In Mode 2, after every command an ACK or (if bad checksum appeared, e.g. device was busy and could not obtain one char) NAK is transmitted before the actual answer to the command is sent. Example (checksum appears as [CS]):

Host: oxyu0001[CS]<<CR>
Target01: ACK[CS]<LF><CR>

Same example as above, but in mode 1:
Host: oxyu0001<CR>
(no answer transmitted)

NOTE: Unless otherwise stated, all examples in this document are in mode1

3.1.1 Command execution time, timeout and repeating

Commands executions time is not unique. Each command activates alternative procedures and requires different execution time. This time can vary from few milliseconds to several tents of milliseconds. Execution of any command can also affect sampling rate. For applications where precise data flow synchronization is required, a use of appropriate operation mode should be considered.

3.1.2 Special characters

The following special characters are used in the interface syntax:

CR – end of command line from host or parameter query from PG2
LF CR – end of data string from PG2
SPC – space (separator)
? – data query character
; : . – value splitting characters

3.2 System initialization

After power-up PG2 needs up to ~4 seconds to initialize the self-test and measurement routines. In this time PG2 ignores all data send to it. Once the internal initialization is ready and self-test passed, PG2 goes into last programmed operation mode. As default the operation mode is set to 1. If any other mode is required, the host must send the right command to PG2 at this time point (see section about operating modes). After power-up sequence all parameters are recalled from internal Flash memory, so the first data string is a valid measurement data (upon calibration parameters are valid). PG2For some examples for initializing the device, see Chapter 6.

3.3 Command codes

There are three commands formats: short, long and data query. The short format is used when the massage contains no value information, and the long one is for sending to the PG2 the parameter value.

Type	Format	Description
Short	*xxxx<CR>*	xxxx is the four letter message code; see commands list Table 2. List of the short type commands .
Long	*xxxxnnnn<CR>*	*xxxx* is the four letter message code; *nnnn* is a four digit value of parameter as integer in the range -999..9999 dependent on parameter; see commands list Table 2. List of the short type commands .

Sensor constants	*xxxxnnnnnnn<CR>*	*As above, but with three extra digit, so values between -999999 and 9999999 are possible, see commands list Table 5*
Data query	*xxxx?<CR>*	*xxxx* is the four letter message code; If it is one of a long type command codes, PG2 will return the actual value of the requested parameter.

Table 1. Command types

Example of a short command to get status report: *repo<CR>*
Example of a long command to set sampling rate to 10 second: *samp0100<CR>*
Example of a data query to get actual sampling rate: *samp?<CR>* ; Response: *100<LF><CR>*

Command Code	Action
repo	Returns status report
data	Data request when in *mode 1*
tmpa	Sets temperature compensation with interne NTC 22k sensor active.
calz	Stores currently measured values (phase and temperature) as low pO_2 calibration
calh	Stores currently measured values (phase and temperature) as high pO_2 calibration
post	Returns Self-test message

Table 2. List of the short type commands

Command Code	Parameter range	Decimal points	Action	Default	Memory Area
avrg	1..9	0	Signal dynamic filter length.	1	1
malp	500..2000	0	Default pressure of measurement [hPa]	1013	1
walp	500.2000	0	Pressure of measurement [hPa]. *NOTE*: Value is NOT saved to Flash. So after reboot, you need to send this value again.	/	/
calp	500..2000	0	Pressure of calibration [hPa]	1013	1
calt	0 or 1	0	Calibration type for gases 1 – dry 0 – humid, in solution	0	1
malt	0 or 1	0	Measurement type for gases 1 – dry 0 – humid, in solution	0	1
clhp	0..90	2	2nd calibration point phase angle [°]	sensor dependent	1
clht	0..50	2	2nd calibration point temperature [°C]	20	1
clof	0..999	0	Fraction of the programmable oxygen concentration [%O_2] for 2nd calibration point.	sensor dependent	1
cloi	0..999	0	Integer of the programmable oxygen concentration [%O_2] for 2nd calibration point.	sensor dependent	1
clzp	0..90	2	1st calibration point phase angle [°]	sensor dependent	1
clzt	0..50	2	1st calibration point temperature [°C]	20	1

Command Code	Parameter range	Decimal points	Action	Default	Memory Area
freq	1	9999	Modulation frequency in Hz divided by 10 *NOTE*: This command shall only be used be experienced users.	N.A.	1
gain	1..6	0	Gain of the Photo detector *NOTE*: This command shall only be used be experienced users.	2	1
idno	0..32	0	Sets device ID number	1	1
mode	0..3	0	Device operation mode. See Operation Mode for details.	1	/
muxm	0 or 1	0	Not supported by PG2	0	1
oxyu	0..6	0	Oxygen unit for data output string: 0 = % air saturation 1 = % O_2 2 = hPa 3 = Torr 4 = mg/l, ppm 5 = µmol/l 6 = ppm gas	0	1
clun	0..6	0	Oxygen unit for the O2-2nd value (this value is set via commands “cloi” and “clof” 0 = % air saturation 1 = % O_2 2 = hPa 3 = Torr 4 = mg/l, ppm 5 = µmol/l 6 = ppm gas See basic functions for details.	Sensor dependent	1
pcco	0..2	0	Pulse counter function: 0 – halted 1 – active 2 – reset (bit 9 of error flag)	0	1
pcof	1..9000	0	Pulse counter overflow limit: 1 = 1.000 9000 = 9.000.000	0	1
phof	-500..500	2	Phase shift offset correction of currently selected sensor *NOTE*: For support and more information please contact PreSens GmbH.	N.A.	1
rdef	password	0	Recalls all parameters to factory default value *NOTE*: Command has no query option	N.A.	1,2
sacu	0..64	0	Changes the current of the signal LED	20	1
racu	0..64	0	Changes the current of the reference LED	7	1
samp	2..9599	1	Sampling rate; Format: *mssd* *(m - minutes, s – seconds, d – first decimal place of the seconds).*	1.5	1
sens	0..6	0	Oxygen sensor type. See basic functions for details.	0	1

Command Code	Parameter range	Decimal points	Action	Default	Memory Area
tmpc	0..60	2	Sets compensation temperature value for oxygen calculation [°C]. Compensation with interne NTC 22k disabled. *NOTE*: Value is NOT saved to Flash. So after reboot, you need to send this value again.	20	/
tclp	0..60	1	1st calibration point of the interne NTC temperature sensor [°C]	N.A.	1
tchp	30.01..75	1	2st calibration point of the interne NTC temperature sensor [°C]	N.A.	1
wdtc	0 or 1	0	System watchdog configuration. See basic functions for details.	1	1
mmwr	0 or 1	0	Activates/Deactivates writing to Flash. See basic functions for details. *NOTE*: Value is NOT saved to Flash. So after reboot, you need to send this value again.	1	/
mmer	1,2 or 3	0	Receive the memory status for specific memory area. *NOTE*: This command will send data although it is whether a query nor a short type command	N.A.	/

Table 3. List of the long type commands

Command Code	Parameter range	Decimal points	Action	Default (PSt3/ PSt6/PSt9/PSt7/PSt8)	Memory Area
scfo	-99.9999 … 999.9999	4	Changes the Sensor constant f1 of the current selected sensor	(0.8080 / 0.8690/ 0.7860 / 0.865 / 0.849)	1
scpo	-9.99999 … 99.99999	5	Changes the Sensor constant dPhi1 of the current selected sensor	(-0.08030 / -0.03796 / -0.02400 / -0-03045 /-0.04691)	1
scpt	-9.99999 … 99.99999	5	Changes the Sensor constant dPhi2 of the current selected sensor	(-0.00000/ -0.00000/ -0.00000/ 0.0/ -0.00057)	1
scks	0.000000 … 9.999999	6	Changes the Sensor constant dKsv1 of the current selected sensor	(0.000433 / 0.008770 / 0.000000 / 0.000365/ 0.000903)	1

Command Code	Parameter range	Decimal points	Action	Default (PSt3/ PSt6/PSt9/PSt7/PSt8)	Memory Area
sckv	0.000000 … 9.999999	6	Changes the Sensor constant dKsv2 of the current selected sensor	(0.000000 / 0.000000 / 0.000000 / 0.00000/ 0.00000)	2
scmm	-9999.99 … 99999.99	2	Changes the Sensor constant m of the current selected sensor	(29.87 / 28.76 / 15.80/ 13.95 / 19.98)	1

Table 4 List of Sensor Constant Commands

Command Code	Action
code?	Returns device code version Example output: `FW Version:  PGT1.0.0.8`
pcav?	Returns the current pulse counter value Example output: `0000001000`
srno?	Returns device serial number Example output: `SAAK0004000080`

Table 5. List of the data-query-only commands

3.4 Measurement data string

Output data string is presented below.

N	N0	;	A	N1	;	P	N2	;	T	N3	;	O	N4	;	E	N8	;	LF CR

List of abbreviations:

CODE DESCRITPTION

N - code for begin of device address N0

N0 - byte value of device address, no decimal places

A - code for begin of amplitude value N1

N1 - long value of amplitude, no decimal places

P - code for begin of phase value N2

N2 - integer value of phase, two decimal places

T - code for begin of temperature value N3

N3 - integer value of temperature, two decimal places

O - code for begin of oxygen value N4

N4 - integer value of oxygen, decimal places 2 (standard) or 4 (only for Oxygen Unit mg/L and ppm gas)

E - code for error value N5

N5 - integer value of error code, no decimal places
Bit 0 – Reference channel overflow
Bit 1 – Reference CLR Status

Bit 2 – Reference DRDY State
Bit 3 – Signal channel overflow
Bit 4 – Signal CLR Status
Bit 5 – Signal DRDY State
Bit 6 – No sensor calculation / Amplitude too low
Bit 7 – Pulse Counter overflow
Bit 8 – Reference Amplitude out of range
Bit 9 – Signal Photo Detector Overflow
Bit 10 – Reference Photo Detector Overflow
Bit 11 – Memory Write Error detected
Bit 12 – reserved
Bit 13 – PME Interrupt error
Bit 14 – PME Interval out of range
Bit 15 – Input voltage out of range
Bit 16 - CRC Error in Memory Sector #1
Bit 17 - CRC Error in Memory Sector #2
Bit 18 - CRC Error in Memory Sector #3
Bit 19 etc – reserved for future use

LF CR - End of frame

NOTE: Measurement string has fixed size for simple data extraction purpose, the only different is shown in example 2, where the oxygen unit is changed to mg/L.

Data string example 1 (nit: %a.s.): N03; A0012941;P2507;T2150;O010120; E00000000;<LF><CR>

Output data interpretation:
N03; *data string from device number 3*
A0012941; *12941 signal amplitude*
P2507; *25,07 degree (signal phase shift)*
T2150; *21.50 °C (compensation temperature)*
O010210; *102.10 oxygen concentration (by two decimal places)*
E00000000; *No error*

Data string example 2 (unit: mg/L): N03; A0012941;P2507;T2150;O00109061; E00000000;<LF><CR>

Output data interpretation:
N03; *data string from device number 3*
A0012941; *12941 signal amplitude*
P2507; *25.07 degree (signal phase shift)*
T2150; *21.50 °C (compensation temperature)*
O00109061; *10.9061 mg/L (by four decimal places)*
E00000000; *No error*

3.5 Limitation & Notes

1. The initializing procedure is started after switching the power on and lasts up to ~4 sec. During this time no data is accepted by PG2.
2. The Host has to take care of the data flow. There is no input buffer to store multiple commands.
3. The hardware-dependent command lines finished with <CR> may not be sent faster than each 250 ms
4. Oxygen concentration value can be negative if calibration constants are not valid or sensor drift occurred.
5. Device saves most settings to flash storage. Please note that flash has a maximum of **10.000 write cycles.** Therefore, **change settings only when necessary. See chapter 7 for further details.**

4 Function and parameters in detail

4.1 System status report

Command : repo <CR>
Displays system parameters stored in the Flash memory in form of a report.

Example of status report:

```
IDENTIFICATION:
Serialnumber: SAAK0004000090
FW Version:  PGT1.0.0.8

PARAMETERS:
RefAmpl:  138472 @ racu   7
SigAmpl:      25 @ sacu  51
pACT in mbar: 1000
Averaging:                 1
Gain:                      2

MEMORY STATUS:
Used RAM #1:                        512
Write Cycles #1:                     57
Used RAM #2:                         52
Write Cycles #2:                      4
Used RAM #3:                         18
Write Cycles #3:                     36

SYSTEM SETTINGS:
Continuous Transmission: 0
Multiplexed Bus Mode:    0
M2M Mode:                0
Watchdog active:         0
Modulation Frequency in  Hz:  4500
Pulse counts:                          0
Pulse counter overflow limit:   9000000
Pulse counter status:        0
Selftest:                          0

Oxygen unit:            %a.s.
Measurement mode:       Humid
Sample Frequency:       0015
T Offset:           0.00000000
T Slope:   1.000

CALIBRATION:
Sensor Type:     PSt3
dPhi1:            -0.08030000
dPhi2:             0.00000000
m:                29.87000000
dKSV1:             0.00043300
dKSV2:             0.00000200
f1:                0.80800000
Cal0:             59.00000000
T0:               20.00000000
A0:                     -100
Cal2nd:           27.00000000
T2nd:             20.00000000
A2nd:                   -100
O2nd in %a.s.  :  100.00000000
pATM in mbar:   970.00000000
Calibration mode: Humid
Reset condition: none
```

4.2 Operation Mode

Command : modexxxx<CR>
where: xxxx = ”0000” ... “0002”

Defines the operation mode:

Mode 0 = continuous mode, this is the basic operation mode when the device sends measurement data with the rate defined by *sampling rate* command.

Mode 1 = request mode H2M, in this mode the device sends measurement data only on request with command *data<CR>*.

Mode 2 = request mode M2M, in this mode the device sends measurement data only on request with command *data[CS]<CR>*. The device will send a ACK or NAK message

Mode 3 = request mode M2M with Done message, in this mode the device sends measurement data only on request with command *data[CS]<CR>*. The device will send a ACK or NAK message and an additional DONE message when the device has completed processing the command

NOTE: This command may be used without checksum calculation (checksum bytes omitted). This ensures proper communication initialization

4.3 Data request

Command : data<CR>

Command “data” used to request data. In mode 1 command activates PG2 to perform a single measurement and send a single data string. The time delay for the PG2 to answer with data string can vary from 200ms up to 300ms. It depends on the AVRG function configuration (averaging filter) as well as of the measurement conditions.

4.4 Selftestmessage

Command: post<CR>

Retuns the bit-coded selftest code. Decode with the following bit code:

Bit	Error
0	Reference Amplitude out of range
1	PME Interrupt error
2	PME Interval out of range
3	Input voltage out of range
4	CRC Error in Memory Sector #1
5	CRC Error in Memory Sector #2
6	CRC Error in Memory Sector #3
7, etc	Reserved for future use

Example Answer:

Selftest: 1

Meaning: Reference Amplitude out of range.

NOTE: For support and more information please contact PreSens GmbH.

4.5 Recall of default settings and parameters

Command : rdef**xxxx**<CR>

where: ***xxxx*** = *default password*

Password protected, allows recalling manufacturer settings. Most parameters will be restored and overwritten. Serial Number and Phase Offsets will be kept.

NOTE: For support and more information please contact PreSens GmbH.

4.6 Sampling rate

Command : samp**xxxx**<CR>

where: ***xxxx*** = parameter value

Command defines the measurement sampling rate with the following format:

x	x	x	x
Minutes *(Value: 0 .. 9)*	Seconds *(Value: 0 .. 59)*		decimal place of seconds *(Value: 2 .. 9)*

Example:

samp1031 will set the sample rate to 1 minute and 3.1 seconds.

4.7 Temperature Compensation

Command : tmpc**xxxx**<CR>, tmpa<CR>

where: ***xxxx*** = *"0000"* ... *"7000"* (corresponds to 0.00°C .. 70.00 °C)

Command defines the compensation temperature for oxygen measurement. By use of the command *tmpcxxxx<CR>*, a host can set a constant temperature value. Temperature compensation with internal NTC sensor is then deactivated. To activate internal NTC (temperature sensor) a *tmpa<CR>* command must be used.

4.8 Signal LED Current

Command : sacu**xxxx**<CR>

where: ***xxxx*** = *"0000"* ... *"0064"* (100 corresponds to app. : to be defined)

Command defines the peak current of the Signal LED. Increasing or decreasing of this parameter changes sensor amplitude. It influences also signal stability, sensor lifetime and drift.

NOTE: Any change of this value can influence calibration validity. For support and more information please contact PreSens GmbH.

4.9 Reference LED Current

Command : racu**xxxx**<CR>

where: ***xxxx*** = *"0000"* ... *"0064"* (100 corresponds to app. : to be defined)

Command defines the peak current of the reference LED.

NOTE: Any change of this value can influence calibration validity. For support and more information please contact PreSens GmbH.

4.10 Gain

Command : gain**xxxx**<CR>

where: ***xxxx*** = *"0001"* ... *"0006"*

Command defines the gain of the PG2 Device (max gain value = 1, min gain value = 6, default = 2)

NOTE: Any change of this value can influence calibration validity. For support and more information please contact PreSens GmbH.

4.11 Measurement Frequency

Command : freq**xxxx**<CR>

where: ***xxxx*** = *"0001" ... "9999"*

Command defines the measurements frequency in Hz divided by 10. Value will be overwritten when using the "sens" command.
Example: freq0450 will change the measurement frequency to 4500Hz.

NOTE: Any change of this value is strictly prohibited. For support and more information please contact PreSens GmbH.

4.12 Sensor type

Command : sens**xxxx**<CR>

where: ***xxxx*** = *"0001" ... "0009"*

Used to define the optical sensor type. Only respective sensor type can be used by defined device type. Please always refer to the sensor's data sheet. Wrong sensor settings make the measurement value incorrect.

Parameter value	Sensor type
1	PSt1 (not supported on PG2)
2	PSt3
3	PSt6
4	PSt7
5	PSt8
6	PSt9

Table 6. Sensor types

NOTE: The sensor type adjustment is limited. For support and more information please contact PreSens GmbH.

4.13 Sensor Specific Constants

Commands: scfo**nnnnnnn**<CR>; scpo**nnnnnnn**<CR>; scpt**nnnnnnn**<CR>; scks**nnnnnnn**<CR>; sckv**nnnnnnn**<CR>; scmm**nnnnnnn**<CR>

Used to set sensor specific calibration constants (see Table 5 for details)

*NOTE: These commands will change the constants of the current selected sensor. To make sure you are editing the correct sensor, use command sens**xxxx** before changing these values!*

NOTE: Wrong sensor constants settings make the measurement value incorrect! For support and more information please contact PreSens GmbH.

The query-option for these commands is different than the querys in the short type commands. For better readability and error checking the output will be as in the following example (please note that all spaces do appear in the output):

Host:	scfo?
Target:	f1: 0.71600000
Host:	scmm?
Target:	m: 6.47000000
Host:	scpo?
Target:	dPhi1: -0.01254000
Host:	scpt?
Target:	dPhi2: -0.00011000
Host:	scks?
Target:	dKSV1: 0.00047300

4.14 Calibration type

The command: caltxxxx<CR>

Where: ***xxxx*** = *"0000" or "0001"*

Sets calibration type as follows: 0 -> calibration with humid gases or liquids, 1 -> calibration with dry gases.

NOTE: It is important to set this parameter prior to the calibration process. Wrong calibration type selection can lead to erroneous oxygen calculation. For support and more information please contact PreSens GmbH.

4.15 Calibration commands

calz<CR> stores the current phase and temperature values for a 1st calibration point.

calh<*CR*> stores the current phase and temperature values for a 2nd calibration point.

The command ***clzp****xxxx*<*CR*> stores the four given digits as the phase value for 1st calibration point. Example: *clzp5623*<*CR*> stores 56.23 as phase for 1st calibration point. All digits must be sent.

The command ***clzt****xxxx*<*CR*> stores the given digits as the temperature value for 1st calibration point. Example: *clzt2150*<*CR*> stores 21.5 as temperature for 1st calibration point c0. All digits must be sent.

The command ***clhp****xxxx*<*CR*> stores the four given digits as the phase value for 2nd calibration point. Example: *clhp2845*<*CR*> stores 28.45 as phase for 2nd calibration point. All digits must be sent.

The command ***clht****xxxx*<*CR*> stores the given digits as the temperature value for 2nd calibration point. Example: *clht1995*<*CR*> stores 19.95 as temperature for 2nd calibration point. All digits must be sent.

The command ***cloi****xxxx*<*CR*> stores the given digits as the integer part of the oxygen concentration in [%O_2] (for sensor type 6 (PSt9 ppm gas is used) for the 2nd calibration point. The command ***clof****xxxx*<*CR*> stores the given digits as the fraction part of the oxygen concentration in [%O_2] for the 2nd calibration point. The PSt9 sensor makes no use of the clof function.

Example 1: *cloi0020*<*CR*> stores 20,95%O_2 as value. The fraction part must be sent separately as *clof0950*<*CR*>. All digits must be sent.

Example 2: To set 99ppm gas as a calibration point for PSt9 sensor the following command must be sent: *cloi0099*<*cr*>

Example 3: To set 9,55%O_2 as a calibration point the following commands must be send: *cloi0009*<*cr*> and *clof0550*<*cr*>.

4.16 Pulse Counter

The commands: pccoxxxx<CR>; pcav?<CR>; pcof xxxx<CR>;

Used to count measurements - sensor readings. If sensor amplitude is over 1000 units, the pulse counter is increased by five by each 100th sensor read out (maximal counter accuracy +/-100 pulses). Counter resolution is 32bit; maximal mathematical counter value is therefore 2^15, practical limit is set to 9.000.000.
If the overflow limit is reached, bit 7 in measurement error register is set until pulse counter is reset.

pccoxxxx**:**

Parameter value	Sensor type
0	Halted - Pulse Counter will not be increased
1	Active
2	Reset – Pulse counter is reset and will remain in previous mode (halted or active)

pcav?
Query only, returns current pulse count

pcofxxxx**:**
Sets the overflow limit

4.17 System Watchdog

The command : wdtcxxxx<CR>
where: ***xxxx*** = *"0000" or "0001"*

Activates (0001)/deactivates (0000) watchdog function. When watchdog is ON (active) the PG2 will reset to its initial state if any firmware deadlock occurs. Watchdog response time is set to app. 2,5sec.

NOTE: When using the watchdog function, the power consumption of the device is increased. For higher resolution temperature measurement and low power consumption, keep the watchdog deactivated. Care is taken that no firmware deadlock appears, but for high safety demand you may activate the watchdog.

4.18 Device ID

Command: idno**xxxx**
Where: ***xxxx*** = *"0001" … "0032"*

Sets the device ID. You may use the device ID for easy identification when using multiple devices on your system.

4.19 Oxygen Unit

Command: oxyu**xxxx**
Where: ***xxxx*** = *"0000" … "0006"*

Parameter value	Sensor type
0	**%a.s.**
1	**%O2**
2	**hPa**
3	**Torr**
4	**mg/L**

5	**µmol/L**
6	**ppm gas**

Sets the oxygen unit used in mode 0 and on data-request. Please not that not every unit may be used with every sensor in any measurement mode. The oxygen unit is kept on false input. When selecting a PSt9 Sensor, the oxygen unit will automatically set to ppm gas. For details on which units are possible in various combinations of sensor and measurement mode, see Tables 8 and 9. When selecting a sensor which does not support the currently selected unit, the unit will automatically be reset to %a.s.

Table 7.Possible Oxygen Units in Measurement Mode "Dry" (x = possible):

	PSt3/PSt7	PSt6/PSt8	PSt9
%a.s.	x	x	
%O2	x	x	
hPa	x	x	
Torr	x	x	
mg/L			
µmol/L			
ppm gas		x	x

Table 8.Possible Oxygen Units in Measurement Mode "Humid" (x = possible):

	PSt3/PSt7	PSt6/PSt8	PSt9
%a.s.	x	x	
%O2	x	x	
hPa	x	x	
Torr	x	x	
mg/L	x	x	
µmol/L	x	x	
ppm gas			x

4.20 O2-2nd Oxygen Unit

Command: clun**xxxx**
Where: ***xxxx*** = *"0000" … "0006"*

This command sets the oxygen unit of the calibration value O2-2nd. Please note, that setting the correct calibration mode (dry/humid. See command "calt") has to be set first. Not all units are possible for every calibration mode. Limitations are equal to Tables 7 and 8.

4.21 Activate/Deactivate Saving to Flash Memory

Command: mmwr**xxxx**
Where: xxxx= "0000" (deactivated) or "0001" (activated)

Due to the limitation of 10.000 write cycles (see chapter 3.5) you may deactivate the automatic writing to the flash memory. When deactivated, all future changes will be applied but not saved. Hence after the next reboot all changes will be **lost**.

This function should be used in applications where continuous write commands are sent, e.g. measuring in regular changing oxygen units.

NOTE: After reactivation all previous changes will be saved to the flash memory on the next write command.

4.22 Calibration of Temperature Measurement

The PG2 is delivered in a calibrated state. However, an experienced user may increase the precision in his specific temperature range with a 2-point temperature calibration. Here you find the calibration process in detail:

Low Point (has to be set first!:

- Set up an environment with a temperature in your desired temperature range (between 0° und 60°C). You need to be able to sustain the temperature. A precision reference temperature measurement device is necessary.
- Put the PG2 in your environment. Wait some minutes until the PG2 temperature measurement is steady. You may encounter a difference between your measured reference value and the PG2 measurement.
- Write the command “tclpXXXX” where XXXX = your reference temperature.

High Point:

- **Increase** the temperature by some degrees (between 30.01° und 75°C. Again, wait some minutes until the PG2 temperature measurement is steady. You may again encounter a difference between your measured reference value and the PG2 measurement.
- Write the command “tchpXXXX” where XXXX = your reference temperature.

4.23 Get Memory Status

Command : mmer**xxxx**<CR>

Where: XXXX = “0000”, “0001” or “0002”

Command “mmer” is used to request the memory status of a specific memory area.

Example:
M0001;E0000000;C0007001;

List of abbreviations:

CODE		DESCRITPTION
M	-	Memory area (here: 1)
E	-	Number of occurred errors
C	-	Number of write cycles (here: 7001)

NOTE: Memory status string has fixed size for simple data extraction purpose

5 Operation Modes

The single measurement takes app 0,2..0.3sec In this period the device performs several operation (reference, measurement, calculation, compensation, error handling) before sending the data to serial port. During this time the device will accept but *not* execute any of externally sent commands. The command execution takes place after measurement’s data sending and before next measurement cycle. It is important to say, that the device has only 32-character input buffer. The string longer then the buffer size will be ignored. Once correct command received, the device must recognize it and execute required action e.g.: store calibration values, change measurement parameter etc. Thus the command execution process delays the next measurement.

5.1 Mode 0 (Continuous)

This is the continuous operation mode. The PG2 sends output data string continuously onto RS232 port with the rate defined by *sampling rate* command. Default sending rate is 1 sec. No more action is needed by the host but receiving (reading out) the serial port buffer. After power-on the device recalls the last set parameters his mode is useful for monitoring and on-line measurements, wherever continuous and fast data flows are needed.

5.2 Mode 1 (H2M Communication)

This is the basic operation mode. In this mode PG2 sends measurement data only on request command from a host.

NOTE: There is a certain delay between the data request command and data string. For details see technical specification.

5.3 Mode 2 (M2M Communication)

This is the basic operation mode for communication between a third party software and the PG2.
NOTE: There is a certain delay between the data request command and data string. For details see technical specification.

NOTE: If device is in Mode 1, but third party software tries to communicate with mode 2 communication frame, no ACK is sent and checksum will not be checked, but the command will still be interpreted. This ensures proper communication initialization

5.4 Mode 3 (M2M Communication with DONE Message)

This mode is similar to Mode 2 but sends additionally a “DONE[CS]” message when the device has finished processing the command and can again accept a new command.

Examples:

Host: post[CS]<CR>
Target: ACK[CS]<LF><CR>
Target: DONE[CS]<LF><CR>

Host: data[CS]<CR>
Target: ACK[CS]<LF><CR>
Target: N01;A0000479;P8414;T2000;O000000;E00000000;[CS]<LF><CR>
Target: DONE[CS]<LF><CR>

6 Establishing of a Connection

Due to the multitude of different operation modes, some examples are given here

6.1 Connection after Power Up, Command line interface

Device is automatically set to mode 1,
Initialization routine:

Host: post<CR>
Target: *Selftest: 0<LF><CR>*

No special arrangements have to be made

6.2 Hot Plug In, Command line interface

6.2.1 Precondition: Device is in mode 2

Host: post<CR>
Target: NAK*[CS]<LF><CR> (because there is no checksum in the host's message)*
Host: mode0001<CR>
Host: post<CR>
Target: Selftest: 0<LF><CR>

6.2.2 Precondition: Device is in mode 1

Host: post<CR>
Target: Selftest: 0<LF><CR>

6.3 Connection after Power Up, third party software (M2M)

Device is automatically set to mode 1. Initialization routine:

Host: mode0002CS<CR> (no answer, because device was in Mode 1)
Host: postCS<CR>
Target: ACK[CS]<LF><CR>
Target: Selftest: 0 [CS]<LF><CR>

6.4 Hot Plug In, third party software (M2M)

6.4.1 Precondition: Device is in mode 1,

Initialization routine:

Host: mode0002[CS]<CR> (no answer, because device was in Mode 1)
Host: post[CS]<CR>
Target: ACK[CS]<LF><CR>
Target: Selftest: 0[CS]<LF><CR>

6.4.2 Precondition: Device is in mode 2.

Initialization routine:

Host: mode0002[CS]<CR>
Target: ACK[CS]<LF><CR>
Host: post[CS]<CR>
Target: ACK[CS]<LF><CR>
Target: Selftest: 0 [CS]<LF><CR>

6.5 Hot Plug In, third party software with Done message (M2M)

6.5.1 Precondition: Device is in mode 1,

Initialization routine:

Host: mode0002[CS]<CR> (no ACK, because device was in Mode 1 while analyzing package)
Target: DONE[CS]<LF><CR> (device is now in mode 3)
Host: post[CS]<CR>
Target: ACK[CS]<LF><CR>
Target: Selftest: 0[CS]<LF><CR>

Target: DONE[CS]<LF><CR>

6.5.2 Precondition: Device is in mode 2.

Initialization routine:

Host: mode0002[CS]<CR>
Target: ACK[CS]<LF><CR>
Target: DONE[CS]<LF><CR>

Host: post[CS]<CR>
Target: ACK[CS]<LF><CR>
Target: Selftest: 0 [CS]<LF><CR>
Target: DONE[CS]<LF><CR>

7 Memory Management

Due to the limitation of 10.000 write cycles (see chapter 3.5) various precautions are implemented:

- The commands for temperature and pressure compensation (tmpc and walp) are commonly used very frequently and are therefore never saved to flash. Nevertheless the default pressure value may be changed with the “malp” command.
- Most settings are saved to memory area #1 (see Table 3. List of the long type commands and Table 4 List of Sensor Constant Commands). The user should always keep an eye on the write cycles of this area (see command “repo”).
- When making extensive use of the numerous commands, the command “mmwr” (see chapter 4.21) should be used to deactivate writing to the flash memory generally.
- Every memory area is protected with 16bit CRC. The CRC is checked on startup. If the saved and the calculated CRC do not match, a selftest error is set (see chapter 4.4). If a selftest error is detected, a proper measurement and/or oxygen calculation cannot be guaranteed. All “Parameter-Values”, ““System Settings” and “Calibration”-Values (see command “repo” in chapter 4.1) need to be set to correct values.
- Although only 10,000 write cycles are guaranteed you may write up to 64,000 times to each area. Beyond 64,000 no further writes are allowed and all changes will be lost after reboot.
- If a memory error was detected on a write command, the error code received in the “data”-package will contain the error bit 12 (see 3.4 for error definition)
- The pulse counter (see chapter 4.16) will perform a flash write every 100th measurement when activated which will of course increment the write cycle count.

8 Connector's Configuration

Table 9. Connector's Configuration

Pin Number	Color	Description
1	Yellow	TxD
2	Green	RxD
3	White	GND
4	Brown	+5Vdc (+/-5%)

The supply voltage of 5V may not be exceeded as it will damage the device.