Serial Communication Interface

PCP-3016

Document revision history

Revision	Date	Description	Change by
05.01	25.01.05	First release – PBH1.21-3.016 compatible	DaAn
09/01	15.09.09	PHB1.21-v.3.016-Trace compatibility added	DaAn

PreSens
Precision Sensing GmbH
Josef-Engert str 11
D93053 Regensburg

Version : PCP-3016 rev.09/01
Author : D.Andrzejewski
Date: 15.09.09
e-mail: d.andrzejewski@presens.de

INDEX

1 General Information

The PCP-3016 interface allows for data exchange between a PC or another host unit and PHB1.21 based transmitter(s) (PBM – Phase Board Module). After a proper initialization the host can freely exchange the data with modules, sending the communication commands and receiving information data strings. There is no handshake function provided in the interface. Therefore the host must always take care about timing and amount of data send.

From hardware side the interface consist of RS232C driver working at baud rate 19200bps (38400bps for Multi-Channel Systems OXY4/OXY10). 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 norm for commands syntax. The interface allows for use of multiple devices at a single host port by using the integrated Smart-Bus feature (see section about Multi-Channel Systems).

2 Data exchange

2.1 Communication basis

The data communication between the host and PBM is based on master-slave base. The master and slave role changes according to the operation mode programmed. In general the host is the master, it starts the communication chain and receives required data from slave (PBM). The one exception is when the PBM 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 three data exchange ways: host to PBM, PBM to host and broadcast command from host to multiple PBMs. Since the input buffer of a single PBM is 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 PBM (see Timing specification). To ensure a proper data exchange an echo function can be use. When activated the function sends a copy of the string received onto the serial port, so the master can validate it.

All interface commands consist of four letter message code and optional four digit value code terminated with carriage return (CR) code. These 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. All codes are case sensitive.

2.2 System initialization

After power-up PBM needs app. 1 seconds (see tech. spec.) for internal procedures to initialize the measurement routines. In this time PBM ignores all data send to it. After the initialization and system check, PBM goes into last programmed operation mode.

As default the operation mode is set to 0 (continuous sending). If any other mode is required, the host must send the right command to PBM at this time point (see section about operating modes). After power-up sequence a last set mode as well as all key parameters are recalled from internal EEPROM memory, so the first data string is a valid measurement data (upon calibration parameters are valid).

For multi-channel systems the initialization time is same as for single PBM. The first measurement data string is here NOT valid. This is due to multi-channel system configuration time. To get the valid data the host must send the data query command to a selected channel. See Multi-Channel system section.

2.3 Special characters

The following special characters are used in the interface syntax:

CR	– end of command line from host or parameter query from PBM
LF CR	– end of data string from PBM
?	– data query character
@	– begin of echo message
;	– value splitting characters

2.4 Command code format

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

Message type	Format	Description
Short	*xxxx<CR>*	*xxxx* is the four letter message code; see commands list Table 1.
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.
Data query	*xxxx?<CR>*	*xxxx* is the four letter message code; If it is one of a long type command codes, PBM will return the actual value (as integer, mind the decimal points!) of the requested parameter.

Example of a short command to get status report: *repo<CR>*

Example of a long command to set signal current to 100: *scur0100<CR>*

Example of a data query to get actual signal current: *scur?<CR>*, and PBM's response: *100<LF><CR>*

Table 1. List of the short type commands

Command Code	Action	Status	Comment
calh	stores currently measured values as the second calibration point	enabled	stored in EEPROM
calz	stores currently measured values as the first calibration point	enabled	stored in EEPROM
data	data request for mode 1,3	enabled	
soff	Remote shut down	enabled	Not for OEM devices
tmpa	Temperature compensation active	enabled	Not for OEM devices
aoaX	Sets analogue channel 1 output parameter, where **X** is the parameter as follows: o – oxygen, p – phase shift, t – temperature, a - amplitude	enabled	Not for OEM devices
aobX	Sets analogue channel 2 output parameter, where **X** is the parameter as follows: o – oxygen, p – phase shift, t – temperature, a - amplitude	enabled	Not for OEM devices

Table 2. List of the long type commands

Command Code	Parameter range	Decimal points	Action	Stored in EEPROM
aplc	0 or 1	0	Automatic Pulse Length function	Yes
aotc	0 or 1	0	Analogue outputs activation/deactivation. Not for OEM devices.	Yes
avrg	0..9	0	Signal dynamic filter length. Only if APL function is deactivated	Yes
cald	1..31	0	Day of calibration	Yes
call	0..23	0	When in mode 2 calls data from particular instrument When in mode 4 wakes up particular instrument for configuration	No
calm	1..12	0	Month of calibration	Yes
calp	500..2000	0	Pressure of calibration [hPa]	Yes
caly	0..99	0	Year of calibration	Yes
clhp	0..90	2	2nd calibration point phase angle [°]	Yes
clht	0..50	1	2nd calibration point temperature [°C]	Yes
clof	0..99	0	Fraction of the programmable oxygen concentration [%a.s.] for 2nd calibration point. Used for trace sensors only.	Yes
cloi	0..400	0	Integer of the programmable oxygen concentration [%a.s.] for 2nd calibration point. Used for trace sensors only.	Yes
clzp	0..90	2	1st calibration point phase angle [°]	Yes
clzt	0..50	1	1st calibration point temperature [°C]	Yes
echo	0 or 1	0	RS 232 echo activation/deactivation	Yes
idno	0..23	0	Sets device channel number. *NOTE: Changing this parameter for Multi-Channel Systems can cause device malfunction.*	Yes
mode	0..4	0	Device operation mode See Operation Mode for details.	Yes
oxyu	0..5	0	Oxygen unit for data output string: 0 = % air saturation 1 = % oxygen saturation 2 = hPa (mbar) 3 = Torr 4 = mg/l, ppm 5 = ?mol/l	Yes
samp	0..120	0	Sampling rate [s]	Yes
scur	0..255	0	Changes the current of the signal LED	Yes

Command Code	Parameter range	Decimal points	Action	Stored in EEPROM
sens	0..7	0	Oxygen sensor type: 0 = no sensor 1 = B2 2 = PSt3 3 = TOS7 4 = PSt3 500% 5 = PSt6 6 = reserved 7 = reserved	Yes
tmpc	-10..60	1	Sets temperature value for oxygen calculation [°C]	Yes

2.5 Data string format

Depending on the operating mode the output data string looks as follows:

MODE 0,1,4 and 5

A	N1	;	P	N2	;	T	N3	;	O	N4	;	E	N5	;	LF CR

MODE 2,3

N	N0	;	A	N1	;	P	N2	;	T	N3	;	O	N4	;	E	N5	;	LF CR

N - code for begin of device address N0
A - code for begin of amplitude value N1
P - code for begin of phase value N2
T - code for begin of temperature value N3
O - code for begin of oxygen value N4
E - code for error value N5
LF CR - end of data string
N0 - byte value of device address, no decimal places
N1 - long value of amplitude, no decimal places
N2 - integer value of phase, two decimal places
N3 - integer value of temperature, one decimal place
N4 - integer value of oxygen, two decimal places
N5 - byte value of error code, no decimal places
Bit 0 - ADC 1 overflow
Bit 1 - ADC 2 overflow
Bit 2 - Amplitude too low
Bit 3 - reserved
Bit 4 - reserved
Bit 5 - No oxygen calculation
Bit 6 - reference LED amplitude < 50000
Bit 7 - not used

Data string example 1: A12941;P2507;T215;O10120;E0;<LF><CR>

Output data interpretation:
A12941; *12941 signal amplitude*
P2507; *25,07 degree (signal phase shift)*
T215; *21,5 °C (compensation temperature)*
O10210; *102,10 oxygen concentration*
E0; *No error*

Data string example 2: N3;A566;P-653;T58;O230;E12;<LF><CR>

Output data interpretation:
N3; *data string from channel number 3*
A566; *566 signal amplitude*
P-653; *-6,53 degree (signal phase/may appear if no sensor)*
T58; *5,8 °C (compensation temperature)*
O230; *2,30 oxygen concentration*
E12; *Error : no temp sensor & amplitude too low*

2.6 Echo format

Echo function is used to confirm proper data exchange between host and PBM. If the echo function is activated in mode 0 and 1, the echo string is send back to the host right after receive by PBM. It consist of @ character followed by the entire message string received. See examples below:

Echo of a valid command in mode 0,1: command sent: *repo<CR>*, echo string: *@repo<LF><CR>*

Echo of a invalid command in mode 0,1: command sent: *aaaa56<CR>*, echo string: *@aaaa56<LF><CR>*

2.7 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.

If echo function is activated in mode 0, PBM sends a copy of received string to the serial port within 500ms. If no echo string is available within 500ms, the timeout condition occurred. It means that PBM was busy and could not handle the command. The host must repeat sending the command in this case.

NOTE 1

The initializing procedure is started after switching the power on and lasts ~2 sec. During this time no data are accepted by PBM.

NOTE 2

There is a character input buffer of 32 characters to store multiple commands (including 'carriage-returns' !). It should be stated, that an overflow of this buffer causes commands to be lost and not be interpreted.

NOTE 3

Hardware-dependent single characters may not be sent faster than 2ms.

NOTE 4

The hardware-dependent command lines finished with <CR> may not be sent faster than each 250 ms.

3 Basic functions in detail

3.1 System status report

The command : repo <CR>

Displays system parameters stored in the EEPROM in form of a report.
Example of the version & status listing:

```
IDENTIFICATION
PHIboard number   : 20020167
PM number         : 00000000
Serial number     : PHB1.21-opta-006-0006
MUX channel       : ON - 01

PARAMETERS
Signal LED current: 150
Ref LED current   : 050
Ref LED amplitude : 29446
Frequency         : 005
Sending interval  : 0001
Averaging         : 2
Internal temp     : 20.0 C

SYSTEM SETTINGS
APL function      : ON
Temp compensation : OFF
Analog out        : OFF
RS232 echo        : OFF
Oxygen unit       : %a.s.

CALIBRATION
Sensor type       : 2
0%a.s.phase 1     : 40.02 at 020.0°C amp 020100
100.00%a.s.phase 2: 27.00 at 020.0°C amp 012200
Date   (ddmmyy)   : 020304
Pressure (mBar)   : 1013

FIRMWARE
Code 3.016 (IAP)  : 11/12/03, 09:10:31
Xilinx built      : 01/05/04 (MM/DD/YY)
```

3.2 RS 232 Echo

Command : echo**xxxx**<CR>

where: ***xxxx*** = *"0000"* for echo OFF or *"0001"* for echo ON

Activates/deactivates "echo" function. When echo is ON the PBM sends the copy (echo) of the command it received on the RS232. This can be used to ensure the proper communication.

3.3 Operation Mode

Command : mode***xxxx***<CR>

where: ***xxxx*** = *"0000"* ... *"0004"*

Defines the operation mode.

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

Mode 1 = sleep, this is the power saving mode. The device sends measurement data only on request by the host .

Mode 2 = MUX serial, this is the multi-channel mode, when several devices are connected to a single RS 232 port. The device sends measurement data only on request by call-by-name function.

Mode 3 = MUX parallel, this is simultaneous mode, enabled only for Multi-Channel Systems (OXY4/OXY10). The device sends measurement data only on request by data function. All channels are activated simultaneously and send data one-by-one sequentially.

Mode 4 = MUX configuration, this is the mode to configure multi-channel systems. Once set, the parameters of particular channel can be set after sending call-by-name function.

3.4 Data (mode 1, 3, 5 only)

Command : data<CR>

Used to request data from the PBM when in mode 1 or 3. The time delay for the PBM to answer with data string can vary from 200ms up to 1000ms. It depends on the APL function configuration as well as of the signal's amplitude and measurement conditions.

3.5 Call-by-name (mode 2 & 4 only)

Command : call**xxxx**<CR>

where: ***xxxx*** = *"0001"* ... *"0023"*

Used to request a data from the particular device of the name **xxxx (**set with *idno* function). In mode 2 activates PBM to perform a single measurement and send a single data string, when in mode 4 it awake PBM as in mode 0.

3.6 Sampling rate

Command : samp**xxxx**<CR>

where: ***xxxx*** = *"0001"* ... *"0120"*

Defines the measurement sampling rate (in sec). This is used in mode 0 only. The interval accuracy is ?0.01 sec.

3.7 Compensation temperature for oxygen measurement

Command : tmpc**xxxx**<CR>

where: ***xxxx*** = *"-100"* ... *"0600"* (corresponds to -10.0°C .. 60.0 °C)

Defines the compensation temperature for measurement. This value is the important for the correct oxygen calculation.

3.8 Signal LED current

Command : scur**xxxx**<CR>

where: ***xxxx*** = *"0000"* ... *"0255"* (255 corresponds to app. 60mA current)

Defines the peak-to-peak current through the led driver stage.
NOTE: Any change of this value can influence calibration validity. For support and more information please contact PreSens GmbH.

3.9 Sensor type

Command : sens**xxxx**<CR>

where: ***xxxx*** = *"0000"* ... *"0009"*

Used to define the oxygen sensor type or disables the oxygen calculation if set to 0. Only respective sensor type can be used by defined device type. The sensor type adjustment is limited by the device type, see below:
NOTE: For support and more information please contact PreSens GmbH directly.

3.10 System watchdog

The command : wdtcxxxx<CR>

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

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

3.11 Calibration

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: *clzt0200<CR>* stores 20,0 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: *clht0200<CR>* stores 20,0 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 in % air saturation for the 2nd calibration point. Example 1: *cloi0040<CR>* stores 40 % air sat. as value. All digits must be sent. The fraction part must be send separately. Example 2: To set 100,05% air saturation of oxygen as a calibration point the following commands must be send: *cloi0100<cr>* and *clof0005<cr>.*
The command ***clof****xxxx<CR>* stores the given digits as the fraction part of the oxygen % air saturation for the 2nd calibration point. Example 1: *clof0040<CR>* stores 0.40 % air sat. as value. All digits must be sent. Example 2: To set 9,55% air saturation of oxygen as a calibration point the following commands must be send: *cloi0009<cr>* and *clof0055<cr>.*

4 Parameters in details

The measurement routine is performed by the PBM according to internal procedures with parameters programmed by the user. Summary of the most important parameters are shown in table 3.

Table 3. Measurement parameters

Sensor type	Please always referee to the sensor's data sheet. Wrong sensor settings make the oxygen measurement incorrect.
LED current	The LED light intensity control - important for signal stability and sensor lifetime. High current (>200) = high signal from the sensor (better S/N ratio), but also faster degradation of the sensor chemistry (short lifetime, stronger drift). Low current (<100) = best sensor lifetime, low drift, but higher signal noise when an old sensor used or sensor's signal low. Best option for long-term measurements with high sampling rates.

APL function	**A**utomatic **P**ulse **L**ength – when ON the function changes the light pulse length (averaging time) according to the signal intensity, in order achieve optimal signal stability this function should be activated. *NOTE: when the sensor's signal is low the longer averaging time is needed to get noise-free readout. It is important in the fast measurement mode described in the next section.*
Dynamic Averaging	When APL function is disabled (OFF) the user can manually program the dynamic averaging filter with the value between 1 and 9. The measurement cycle with the shortest filter length takes app. 100ms. Increasing the filter by one prolongs that time by app. 85ms. The filter equal to 9 results in measurement cycle time of app. 800ms.
Temperature compensation	The PBM is not ready for automatic temperature compensation of the oxygen value. By use of the command *tmpcxxxx<CR>*, the host can program the information about compensation temperature.

5 Operation modes

5.1 Single measurement routine

The single measurement takes app 1sec. In this period the device performs several operation (reference, measurement, calculation, compensation, error handling) before sending the data to RS 232 port. During this time the device will accept but *not* execute any of externally send 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. Thus, not more then 32 characters (including special characters) can be sent to the device in the time gap of app. 1sec. The string longer then the buffer size will be ignored. Once a 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.

5.2 Mode 0 (continuous)

This is the basic operation mode. The PBM sends output data string continuously onto RS 232 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 and starts sending data. This mode is useful for monitoring and on-line measurements, wherever continuous and fast data flow is needed.

5.2.1 Fast measurement

In the *Mode 0* it is possible to set the sending rate parameter to zero. This is a special case when the measurement data will be sent as fast as possible to the RS 232 port. In *the fast option* sending time is affected by instrument settings and options. To achieve fastest sampling rate and a good performance, the device should be configured with following parameters/options:

1) Automatic Pulse Length (APL) function must be disabled
2) Dynamic averaging filter must be set to 1

NOTE: the fast option should be used only if sensor's signal amplitude is higher then 10000 units. This guaranties reliable and stable signal.

5.3 Mode 1 (power safe/sleep)

This is the power saving mode. This mode is useful for long time measurements when the fast scan rate is not of importance. In this mode PBM sends measurement data only on request command from a host. After sending the data string PBM automatically goes to low power consumption state.

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

5.4 Mode 2 (MUX serial)

The Mode 2 is the basic multi-channel mode. By mean of PreSens Smart Bus (PSB) several single PBM can be plugged to a single RS232 port. The operation of such a system is possible neither in mode 0 nor 1. To allow multiple accesses to a single serial port, mode 2,3 or 4 ought to be used. In general, the operation in mode 2 is similar to mode 1, but here each channel is "triggered" separately by calling-by-name. The fastest sequential scanning rate of the n-channel system can be derived from:

Scan rate [sec] ? n + 1,5

NOTE: When all PBM are plugged to one serial port, all can listen but only one can answer at the time. Therefore before calling next channel, the data string from the previous one must be fully received!

NOTE: In any of MUX modes it is possible to program all boards simultaneously with the same parameters. More about changing parameter of a particular device (channel) only see in mode 4.

To achieve data from a particular channel the request command must be send to the system. All devices listen to this request, but only one with the right name responds and sends data to the port. This is what we call the call-by-name function. All PBM must be programmed to mode 2 before operating as the multi-channel system.

NOTE: It is essential to program a device number (command idnoxxxx<CR>) prior to operating in mode 2.

5.5 Mode 3 (MUX parallel)

The Mode 3 is the optional multi-channel mode. This mode is applicable only for Multi-Channel Systems (OXY4/OXY10 series) and the OEM modules (requires additional wiring). In mode 3 all channels are activated simultaneously and send data one-by-one sequentially. Measurement action is activated by command *data<CR>*. Similar to mode 2 all channels listen to commands from RS232 port, so all can be reprogrammed simultaneously. This mode is useful wherever fast scanning of multiple channels is needed. Single channel can be reached in mode 4.

NOTE : All channels are triggered exactly at the same time. In the fast mode 3 (similar to fast mode 0) the scanning rate of 2 Hz for all 10 channels can be reached.

5.6 Mode 4 (MUX configuration)

This is the special mode used to configure multi-channel systems. When in mode 4 the system (a single device as well) ignores all commands but *mode* and *call*. The idea of mode 4 is to be able to program or "talk" to the one, particular channel only even if all the other are listening. Though others are listening, they do not interpret commands which are sent to the active, selected channel.

Configuration procedure always starts with mode 4 activation. Then the multi-channel PBM system is waiting for setting the active channel. It is done by use of call-by-name function. The channel called by call command is set to active state and automatically changes it's mode to 0. This means, it starts to send data continuously onto the serial port. Only the active channel reacts to the commands sent. This makes possible reprogramming/configuring single channel without

changing entire system settings. To go out from mode 4, it is enough to set previous system's mode 2 or 3.

NOTE: The mode 4 is not stored in EEPROM, thus after power-on the system will be initialized with last set mode 2 or 3.