P-3X – Interface protocol

1. Interface configuration

USB 2.0 compliant full-speed, Virtual COM-Port

9600 baud, 8 data bits, no parity, 1 stop bit

2. Operating mode

The P-3X can be alternatively configured to allow different modes of operation:

service	HOST transmits					Transmitter replies				
	1	2	3	4	5	1	2	3	4	5
Set operating mode	S 0x53	O 0x4F	mode	CS	CR 0x0D	s 0x73	o 0x6F	mode	CS	CR 0x0D

Mode:		
	0xFF	refer to 2.1. Polling mode
	0xFE	refer to 2.2. Cyclic mode (pressure value in digit)
	0xFD	refer to 2.3. Cyclic mode (pressure and temperature value in digit)
	0xFC	refer to 2.4. Cyclic mode (pressure value in physical unit)
	0xFB	refer to 2.5. Cyclic mode (pressure and temperature value in physical unit)

Switching from one operating mode to another can be done in every mode of operation. These changes are not permanently saved.
Permanent changes of the operating mode can only be done via EasyCom software.

2.1 Polling mode

Transmitter sends on request
(see 3 Additional services)

2.2 Cyclic mode (pressure value in digit)

The transmitter sends the current pressure value in time intervals set by the user.

Pressure value in digit					
1	2	3	4	5	6
k 0x6B	H-Byte	L-Byte	0x00	CS	CR 0x0D

The resolution of “pressure value in digit” is 10000 to 50000.

Converting digits to physical unit (pressure):

$$p = \frac{[(hb \cdot 256 + lb) - 10000] \cdot (ZP - FS)}{50000} + FS$$

2.3 Cyclic mode (pressure and temperature value in digit)

The transmitter sends the current pressure and temperature value in time intervals set by the user. (10 pressure values followed by 1 temperature value)

Pressure value					
1	2	3	4	5	6
k 0x6B	H-Byte	L-Byte	0x00	CS	CR 0x0D

Converting digits to physical unit see 2.2 Cyclic mode (pressure value in digit)

Temperature value					
1	2	3	4	5	6
T 0x54	H-Byte	L-Byte	0x00	CS	CR 0x0D

Converting digits to physical unit (temperature): $T[°C] = \frac{LoByte}{2}$

If $HiByte = 1$, then $T[°C] = -\frac{LoByte}{2}$

If $HiByte = 0$, then $T[°C] = +\frac{LoByte}{2}$

e.g.: H-Byte = 0x01 negative value

L-Byte =0x13 absolute value = $\frac{0x13}{2} = \frac{19_{dec}}{2} = 9{,}5$

➔Temperature = - 9,5 °C

2.4 Cyclic mode (pressure value in physical unit)

The transmitter sends the current pressure value in time intervals set by the user.

Transmitter sends							
1	2	3	4	5	6	7	8
P 0x50	Byte0	Byte1	Byte2	Byte3	unit	CS	CR

Pressure value format acc. IEEE754 Floating-Point Arithmetic (see 7. Number format))

2.5 Cyclic mode (pressure and temperature value in physical unit)

The transmitter sends the current pressure and temperature value in time intervals set by the user.
(10 pressure values followed by 1 temperature value)

Pressure value (physical unit)							
1	2	3	4	5	6	7	8
P 0x50	Byte0	Byte1	Byte2	Byte3	unit	CS	CR

Pressure value format acc. IEEE754 Floating-Point Arithmetic (see 7. Number format))

Temperature value (physical unit)							
1	2	3	4	5	6	7	8
T 0x54	H-Byte	L-Byte	0x00	CS	CR 0x0D		

Converting digits to physical unit (temperature) see 2.3 Cyclic mode (pressure and temperature value in digit)

3. Additional services

3.1 Read “Zero Point” (ZP) / “Full Scale” (FS)

service	Host sends					Transmitter replies							
	1	2	3	4	5	1	2	3	4	5	6	7	8
Read ZP	M 0x4D	A 0x41	0x00	CS 0x72	CR 0x0D	0x03	Byte0	Byte1	Byte2	Byte3	unit	CS	CR
Read FS	M 0x4D	E 0x45	0x00	CS 0x6E	CR 0x0D	0x04	Byte0	Byte1	Byte2	Byte3	unit	CS	CR

Byte0 to Byte3 format acc. IEEE754 Floating-Point Arithmetic (see 7. Number format))

Encoding of unit see chapter 6. Physical units

3.2 Read process values

service	Host sends					Transmitter replies							
	1	2	3	4	5	1	2	3	4	5	6	7	8
Pressure (digit)	P 0x50	K 0x4B	0x00	CS 0x65	CR 0x0D	k 0x6B	H-Byte	L-Byte	0x00	CS	CR 0x0D		
Pressure (physical unit)	P 0x50	Z 0x5A	0x00	CS 0x56	CR 0x0D	P 0x50	Byte0	Byte1	Byte2	Byte3	unit	CS	CR
Temperature	T 0x54	W 0x57	0x00	CS 0x55	CR 0x0D	T 0x54	H-Byte	L-Byte	0x00	CS	CR 0x0D		

Pressure value format acc. IEEE754 Floating-Point Arithmetic (see 7. Number format))

Encoding of unit see chapter 6. Physical units

3.3 Read serial number

service	Host sends					Transmitter replies							
	1	2	3	4	5	1	2	3	4	5	6	7	8
Seriennummer lesen	K 0x4B	N 0x4E	0x00	CS 0x67	CR 0x0D	K 0x4B	Byte0	Byte1	Byte2	Byte3	CS	CR	

Byte0 to Byte3 format acc. IEEE754 Floating-Point Arithmetic (see 7. Number format))

3.4 Configure transfer rate

Valid range: 10 … 65525 milliseconds

service	Host sends					Transmitter replies				
	1	2	3	4	5	1	2	3	4	5
Set transfer rate	I 0x49	H-Byte	L-Byte	CS	CR 0x0D	i 0x69	H-Byte	L-Byte	CS	CR 0x0D

4. Reset (factory settings)

Only possible via EasyCom software

5. Checksum calculation

example code (Delphi):

```
//---------------------------------------------------------
// Calculation of checksum from string

function createChecksum (Value: String): Char;
var
  i,
  tmp   : integer;
begin
  tmp := 0;

  for i := 1 to length (Value) do
    tmp := tmp + ord(Value[i]);

  tmp := LoByte(tmp);
  tmp := (tmp xor $FF);  // two´s complement
  inc(tmp);

  createChecksum := chr(tmp);
end;
```

e.g.: checksum of ”Read Zero Point”

Host sends

M 0x4D	A 0x41	0x00	CS 0x72	CR 0x0D

Calculation:

0x4D + 0x41 + 0x00 = 0x008E
Low-Byte of 0x008E = 0x8E
Two´s complement = 0x72

⇨ Checksum = 0x72

6. Physical units

unit code		unit
gauge	**absolute**	
$FE	$FF	bar
$1E	$1F	Psi
$AE	$AF	MPa
$BE	$BF	kg / cm²

7. Number format

7.1 UNSIGNED32

	Byte0	Byte1	Byte2	Byte3
UNSIGNED32	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$

Data = $b_{31} \cdot 2^{31} + b_{30} \cdot 2^{30} + \ldots + b_1 \cdot 2^1 + b_0 \cdot 2^0$

7.2 Float-Format acc. IEEE754

	Byte0	Byte1	Byte2	Byte3
Float32	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$

Byte0 to Byte3 format acc. IEEE754 Floating-Point Arithmetic

FLOAT32(*b*) = $(-1)^S \cdot 2^{E-127} \cdot (1+F)$

$$S = b_{31}$$

$$E = b_{30} \cdot 2^7 + \ldots + b_{23} \cdot 2^0$$

$$F = 2^{-23} \cdot (b_{22} \cdot 2^{22} + \ldots + b_1 \cdot 2^1 + b_0 \cdot 2^0)$$

8. Abbreviations

CS	Checksum
CR	carriage return (0x0D)
hb	High-Byte
lb	Low-Byte
ZP	zero point
FS	full scale
MSB	most significant bit
LSB	least significant bit
0x	hexadecimal