

P-3X – Schnittstellenprotokoll für den Anwender

1. Schnittstellenkonfiguration

USB 2.0 compliant full-speed, Virtual COM-Port

9600 Baud, 8 Datenbits, keine Parität, 1 Stoppbit

2. Betriebsmodus

Der P-3X lässt sich wahlweise in unterschiedliche Betriebsarten konfigurieren.

Dienst	Anfrage, PC sendet					Antwort, PC empfängt				
	1	2	3	4	5	1	2	3	4	5
Setze Ausgabemodus	S 0x53	O 0x4F	Modus	CS	CR 0x0D	s 0x73	o 0x6F	Modus	CS	CR 0x0D

Modus: 0xFF Polling-Betrieb nach 2.1
0xFE zyklische Druckwertausgabe (in Digit) nach 2.2
0xFD zyklische Druck- und Temperatúrausgabe (in Digit) nach 2.3
0xFC zyklische Druckausgabe (in physikalischer Einheit) nach 2.4
0xFB zyklische Druck- und Temperatúrausgabe (in physikalischer Einheit) nach 2.5

Die Umstellung von einem Betriebsmodus in einen anderen ist mittels Softwarebefehl im laufenden Betrieb jederzeit möglich. Diese geht jedoch mit dem Abschalten der Versorgung verloren. Eine bleibende Änderung des Betriebsmodus ist nur mittels EasyCom-Software möglich.

2.1 Polling-Betrieb

Der Transmitter sendet nur nach einer Anfrage
(siehe Kapitel 3 Schnittstellendienste)

2.2 zyklische Druckwertausgabe (in Digit)

Der Transmitter sendet selbständig im Abstand eines vom Anwender einstellbaren Zeitintervalls den aktuellen Druck

Druckwert in Digit					
1	2	3	4	5	6
k 0x6B	H-Byte	L-Byte	0x00	CS	CR 0x0D

Das Drucksignal in Digit wird in der Auflösung von 50.000 Digits codiert. Im MBA liefert der Transmitter 10.000 Digits, im MBE 60.000 Digits.
Der reale Druckwert errechnet sich hieraus mit:

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

2.3 zyklische Druck- und Temperatursausgabe (in Digit)

Der Transmitter sendet selbständig im Abstand eines vom Anwender einstellbaren Zeitintervalls jeweils 10-mal den aktuellen Druck und dann 1-mal die aktuelle Temperatur.

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

Interpretation des Drucksignals siehe Kapitel 2.2 zyklische Druckwertausgabe (in Digit)

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

Der Temperaturwert errechnet sich aus: $T[^\circ\text{C}] = \frac{LoByte}{2}$, wobei das H-Byte das Vorzeichen enthält:

0x01: Temperaturwert negativ

0x00: Temperaturwert positiv

Beispiel: H-Byte = 0x01

L-Byte = 0x13

Wert ist negativ

Betrag ist $\frac{0x13}{2} = \frac{19_{dez}}{2} = 9,5$

→ Temperatur = -9,5 °C

2.4 zyklische Druckausgabe (in physikalischer Einheit)

Der Transmitter sendet selbständig im Abstand eines vom Anwender einstellbaren Zeitintervalls den aktuellen Druck in der physikalischen Einheit

Antwort, PC empfängt							
1	2	3	4	5	6	7	8
P 0x50	Byte0	Byte1	Byte2	Byte3	Einh	CS	CR

Der Druck in physikalischer Einheit ist im Float-Format nach IEEE754 kodiert (siehe Kap. 7 Zahlenformate)

2.5 zyklische Druck- und Temperatursausgabe (in physikalischer Einheit)

Der Transmitter sendet selbständig im Abstand eines vom Anwender einstellbaren Zeitintervalls jeweils 10 mal den aktuellen Druck und dann 1 mal die aktuelle Temperatur.

Druck in physikalischer Einheit							
1	2	3	4	5	6	7	8
P 0x50	Byte0	Byte1	Byte2	Byte3	Einh	CS	CR

Der Druck in physikalischer Einheit ist im Float-Format nach IEEE754 kodiert (siehe Kap. 7 Zahlenformate)

Temperaturwert							
1	2	3	4	5	6	7	8
T 0x54	H-Byte	L-Byte	0x00	CS	CR 0x0D		

Interpretation des Temperatursignals siehe Kapitel 2.3 zyklische Druck- und Temperatursausgabe (in Digit)

3. Schnittstellendienste

3.1 Messbereichsanfang (MBA) / Messbereichsende (MBE) des Transmitters auslesen

Dienst	Anfrage, PC sendet					Antwort, PC empfängt							
	1	2	3	4	5	1	2	3	4	5	6	7	8
MBA lesen	M 0x4D	A 0x41	0x00	CS 0x72	CR 0x0D	0x03	Byte0	Byte1	Byte2	Byte3	Einheit	CS	CR
MBE lesen	M 0x4D	E 0x45	0x00	CS 0x6E	CR 0x0D	0x04	Byte0	Byte1	Byte2	Byte3	Einheit	CS	CR

Byte0 bis Byte3 sind im Float-Format nach IEEE754 kodiert (siehe Kap. 7 Zahlenformate)

Kodierung der Einheit siehe Kap. 6 Einheiten

3.2 Messwerte auslesen

Dienst	Anfrage, PC sendet					Antwort, PC empfängt							
	1	2	3	4	5	1	2	3	4	5	6	7	8
Druck in Digit	P 0x50	K 0x4B	0x00	CS 0x65	CR 0x0D	k 0x6B	H-Byte	L-Byte	0x00	CS	CR 0x0D		
Druck in physikal. Einheit	P 0x50	Z 0x5A	0x00	CS 0x56	CR 0x0D	P 0x50	Byte0	Byte1	Byte2	Byte3	Einh	CS	CR
Temperaturwert	T 0x54	W 0x57	0x00	CS 0x55	CR 0x0D	T 0x54	H-Byte	L-Byte	0x00	CS	CR 0x0D		

Der Druck in physikalischer Einheit ist im Float-Format nach IEEE754 kodiert (siehe Kap. 7 Zahlenformate)

Kodierung der Einheit siehe Kap. 6 Einheiten

3.3 Seriennummer auslesen

Dienst	Anfrage, PC sendet					Antwort, PC empfängt							
	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 bis Byte3 sind im Format UNSIGNED32 kodiert (siehe Kap. 7 Zahlenformate)

3.4 Ausgabeintervall für zyklischen Betriebsmodus

Als Ausgabeintervall lässt sich ein Wert zwischen 10 und 65535 Millisekunden programmieren.

Dienst	Anfrage, PC sendet					Antwort, PC empfängt				
	1	2	3	4	5	1	2	3	4	5
Setze Ausgabeintervall	I 0x49	H-Byte	L-Byte	CS	CR 0x0D	i 0x69	H-Byte	L-Byte	CS	CR 0x0D

4. Reaktivierung der Werkseinstellung

Ist ausschließlich mit der EasyCom-Software möglich.

5. Erzeugung der Checksumme

Die Checksumme wird generiert, indem zunächst die vorangehenden Bytes des Sendestrings addiert werden. Von dem Resultat wird alleine das Low-Byte weiterverarbeitet. Dessen Zweierkomplement bildet schließlich die Checksumme.

Nachfolgend ein Delphi-Quellcodebeispiel zur Checksummengenerierung:

```
//-----  
// Erzeugt eine Prüfsumme aus dem übergebenen String  
//  
// Wert: der String aus dem die CS zu berechnen ist  
// Return: Checksumme  
  
Function createChecksum (Wert: String): Char;  
var  
    i,  
    tmp    : integer;  
begin  
    tmp := 0;  
  
    for i := 1 to length (szWert) do  
        tmp := tmp + ord(szWert[i]);  
  
    tmp := LoByte(tmp);  
    tmp := (tmp xor $FF);  
    inc(tmp);  
  
    createChecksum := CHR(tmp);  
end;
```

Beispiel zur Checksummengenerierung anhand des Befehls „Messbereichsanfang (MBA) des Transmitters auslesen“

Wie bereits in Kapitel 3.1 dargestellt sendet der Host zum Auslesen des MBA:

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

Hierzu errechnet sich die Checksumme CS wie folgt:

- Addition vorangehender Bytes: $0x4D + 0x41 + 0x00 = 0x008E$
- hiervon das Low-Byte: $0x8E$
- hiervon das Zweierkomplement: $0x72$

Damit ist für diesen Sendebefehl die Checksumme 0x72.

6. Einheiten

Einheit		Einheit
rel	abs	
\$FE	\$FF	bar
\$1E	\$1F	Psi
\$AE	\$AF	MPa
\$BE	\$BF	kg / cm ²

7. Zahlenformate

7.1 UNSIGNED32

	Byte0	Byte1	Byte2	Byte3
UNSIGNED32	b ₇ ..b ₀	b ₁₅ ..b ₈	b ₂₃ ..b ₁₆	b ₃₁ ..b ₂₄

$$\text{Daten} = b_{31} \cdot 2^{31} + b_{30} \cdot 2^{30} + \dots + b_1 \cdot 2^1 + b_0 \cdot 2^0$$

7.2 Float-Format nach IEEE754

	Byte0	Byte1	Byte2	Byte3
Float32	b ₇ ..b ₀	b ₁₅ ..b ₈	b ₂₃ ..b ₁₆	b ₃₁ ..b ₂₄

Die Kodierung erfolgt nach dem Standard IEEE 754-1985 (single precision floating-point)

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

Wobei:

$$S = b_{31}$$

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

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

8. Abkürzungen

CS	Checksumme, erzeugt aus dem Zweierkomplement vom Low-Byte der Summe der vorangehenden Bytes des Sendestrings
CR	carriage return (0x0D) als Befehlsende-Kennung
hb	High-Byte
lb	Low-Byte
MSB	most significant bit
LSB	least significant bit
0x	hexadezimal