Anhang E Schnittstellenprotokoll Steuerung - PC

(Änderungen vorbehalten)

E.1 Aufbau des Protokolls

Schnittstelle:	RS 232
Baudrate:	19'200 Baud
Format:	8 Bit, ODD - Parity (ungerade, das Parity-Bit ergänzt die Summe der '1'en zu einer ungeraden Zahl)
Datenflusskontrolle:	keine
Framing:	**'STX' 'Daten' 'CHK' 'ETX'**
	STX = 0x02
	ETX = 0x03
	CHK = XOR-Verknüpfung aller Daten (ohne STX, ETX und CHK) Das höchste Bit (Bit 7 resp. MSB) der Daten und der CHK ist immer 1. Beispiel: ASC '1' =DEZ 49 ODER DEZ 128 = DEZ 177 resp. HEX 0x31 ODER HEX 0x80 = HEX 0xB1.
	ADR = 0x81 - 0xA0 (Adresse 01 - 32); wird über Software eingestellt. Default = 0x81 (Adresse 01).

Außer bei 'ETX' und 'STX' ist das höchste Bit (MSB) immer 1

E.2 Befehle und Antworten

E.2.1 Uhrzeit stellen

PC an CPU:

'STX' 'ADR' 't' date time 'CHK' 'ETX'

't'	ASCII-Code 0x74 OR 0x80 = **0xF4**	
date	TTMMJJ	je Byte in ASCII OR 0x80 (6 Bytes)
time	HHMMSS	je Byte in ASCII OR 0x80 (6 Bytes)

Beispiel: ADR = 1, date = 241196, time = 145535

String = 0x02 0x81 0xF4 0xB2 0xB4 0xB1 0xB1 0xB9 0xB6
0xB1 0xB4 0xB5 0xB5 0xB3 0xB5 0xFF 0x03 (17 Bytes)

CPU an PC:

'STX' 'ADR' 't' date time 'CHK' 'ETX' (eingestellter Wert)

't' ASCII-Code 0x74 OR 0x80 = **0xF4**

E.2.2 Uhrzeit lesen

PC an CPU:

'STX' 'ADR' 'T' 'CHK' 'ETX'

'T' ASCII-Code 0x54 OR 0x80 = **0xD4**

CPU an PC:

'STX' 'ADR' 'T' date time 'CHK' 'ETX' (gelesener Wert)

'T' ASCII-Code 0x54 OR 0x80 = **0xD4**

E.2.3 Analog-Werte stellen

PC an CPU:

‘STX’ ‘ADR’ ‘a’ KanalNr_Wert ‘CHK’ ‘ETX’

‘a’	ASCII-Code 0x61 OR 0x80 = **0xE1**
KanalNr	ein Byte in ASCII OR 0x80
_	Leerzeichen = 0x20 OR 0x80 = 0xA0
Wert	Format XXX.X (bei negativen Werten -XX.X) je Byte in ASCII OR 0x80

Hinweis: Bei mehreren Kanälen ist jeder Kanal einzeln zu stellen

Beispiel: ADR = 1, KanalNr = 0 (Temperaturkanal), Wert = -14.5 °C
String = 0x02 0x81 0xE1 0xB0 0xA0 0xAD 0xB1 0xB4 0xAE 0xB5 0xC3 0x03 (12 Bytes)

CPU an PC:

‘STX’ ‘ADR’ ‘a’ ‘CHK’ ‘ETX’

‘a’	ASCII-Code 0x61 OR 0x80 = **0xE1**

E.2.4 Analog-Werte lesen

PC an CPU:

‘STX’ ‘ADR’ ‘A’ KanalNr ‘CHK’ ‘ETX’

‘A’	ASCII-Code 0x41 OR 0x80 = **0xC1**
KanalNr	ein Byte in ASCII OR 0x80

Beispiel: ADR = 1, KanalNr = 0 (Temperaturkanal)
String = 0x02 0x81 0xC1 0xB0 0xF0 0x03 (6 Bytes)

CPU an PC:

‘STX’ ‘ADR’ ‘A’ KanalNr_IstWert_SollWert ‘CHK’ ‘ETX’ (gelesener Wert)

‘A’	ASCII-Code 0x41 OR 0x80 = **0xC1**
KanalNr	ein Byte in ASCII OR 0x80
_	Leerzeichen = 0x20 OR 0x80 = 0xA0
IstWert	Format XXX.X (-XX.X bei negativen Werten) je Byte in ASCII OR 0x80
SollWert	Format XXX.X (-XX.X bei negativen Werten) je Byte in ASCII OR 0x80

Beispiel: ADR = 1, KanalNr = 0 (Temperaturkanal), IstWert = -14.5 °C, SollWert = -13.8 °C
String = 0x02 0x81 0xC1 0xB0 0xA0 0xAD 0xB1 0xB4 0xAE 0xB5 0xA0 0xAD 0xB1 0xB3 0xAE 0xB8 0xFA 0x03 (18 Bytes)

Hinweis: Bei mehreren Kanälen ist jeder Kanal einzeln zu lesen

E.2.5 Definierte Änderungsgeschwindigkeiten (Gradienten) vorgeben

Allgemein: Der in den unten beschriebenen Befehlen vorgebbare Wert hat die Einheit K/min. Mit dieser Änderungsgeschwindigkeit wird der eingestellte Sollwert angefahren.

Achtung: Der so eingestellte Gradient bleibt in der Steuerung erhalten bis ein anderer Wert folgt. Soll nach einer Rampe ein Sollwertsprung mit maximaler Änderung gefahren werden, so sind die Gradienten auf den Maximalwert von 999.9 K/min zu setzen.

Zur Realisierung einer Sollwertrampe wird also zunächst die Änderungsgeschwindigkeit vorgegeben und anschließend der gewünschte Endwert als Sollwert eingestellt. Ist der Endwert erreicht, fährt die Anlage diesen Sollwert konstant weiter bis eine neue Vorgabe kommt.

Für die ITC-Steuerung wird mit dem Befehl zum Lesen der Analogwerte immer der aktuelle Sollwert innerhalb der Rampe zurückgegeben. Um den Eingestellten Endwert der Rampe abzufragen ist das Kommando ‚E' (siehe unten) zu benutzen.

Anmerkung: Das Format für den Wert des Gradienten kann auch in einer anderen Form übergeben werden. Sind z.B. genauere Werte für den Gradienten erforderlich, so können auch zwei Nachkommastellen angegeben werden.

Beispiel: 00.05 ➔ 0.05 K/min
23.45 ➔ 23.45 K/min

E.2.6 Gradient Steigen (z.B. Heizen) stellen

PC an CPU:

'STX' 'ADR' 'u' KanalNr_Wert 'CHK' 'ETX'

'u'	ASCII-Code 0x75 OR 0x80 = **0x F5**
KanalNr	ein Byte in ASCII OR 0x80
_	Leerzeichen = 0x20 OR 0x80 = 0xA0
Wert	Format XXX.X in K/min je Byte in ASCII OR 0x80

Hinweis: Bei mehreren Kanälen ist jeder Kanal einzeln zu stellen.
Der Wert für den Gradient ist immer Positiv.

CPU an PC:

'STX' 'ADR' 'u' 'CHK' 'ETX' (gelesener Wert)

'u'	ASCII-Code 0x75 OR 0x80 = **0x F5**

E.2.7 Gradient Sinken (z.B. Kühlen) stellen

PC an CPU:

'STX' 'ADR' 'd' KanalNr_Wert 'CHK' 'ETX'

'd'	ASCII-Code 0x64 OR 0x80 = **0x E4**
KanalNr	ein Byte in ASCII OR 0x80
_	Leerzeichen = 0x20 OR 0x80 = 0xA0
Wert	Format XXX.X in K/min je Byte in ASCII OR 0x80

Hinweis: Bei mehreren Kanälen ist jeder Kanal einzeln zu stellen.
Der Wert für den Gradient ist immer Positiv.

CPU an PC:

'STX' 'ADR' 'd' 'CHK' 'ETX' (gelesener Wert)

'd'	ASCII-Code 0x64 OR 0x80 = **0x E4**

E.2.8 Eingestellte Gradienten auslesen

PC an CPU:

‘STX’ ‘ADR’ ‘U’ KanalNr ‘CHK’ ‘ETX’

‘U’	ASCII-Code 0x55 OR 0x80 = **0xD5**
KanalNr	ein Byte in ASCII OR 0x80

CPU an PC:

‘STX’ ‘ADR’ ‘U’ KanalNr_Gradientsteigen_Gradientsinken ‘CHK’ ‘ETX’ (gelesener Wert)

‘U’	ASCII-Code 0x55 OR 0x80 = **0xD5**
KanalNr	ein Byte in ASCII OR 0x80
_	Leerzeichen = 0x20 OR 0x80 = 0xA0
Gradientsteigen	Format XXX.X in K/min je Byte in ASCII OR 0x80
Gradientsinken	Format XXX.X in K/min je Byte in ASCII OR 0x80

Hinweis: Bei mehreren Kanälen ist jeder Kanal einzeln zu lesen.
Der Wert für den Gradient ist immer Positiv.

E.2.9 Eingestellte Endwert für die Rampe auslesen

PC an CPU:

‘STX’ ‘ADR’ ‘E’ KanalNr ‘CHK’ ‘ETX’

‘E’	ASCII-Code 0x45 OR 0x80 = **0xC5**
KanalNr	ein Byte in ASCII OR 0x80

CPU an PC:

‘STX’ ‘ADR’ ‘E’ KanalNr_Endwert ‘CHK’ ‘ETX’ (gelesener Wert)

‘E’	ASCII-Code 0x45 OR 0x80 = **0xC5**
KanalNr	ein Byte in ASCII OR 0x80
_	Leerzeichen = 0x20 OR 0x80 = 0xA0
Endwert	Format XXX.X in K/min je Byte in ASCII OR 0x80

Hinweis: Bei mehreren Kanälen ist jeder Kanal einzeln zu lesen.

E.2.10 Status lesen

PC an CPU:

‘STX’ ‘ADR’ ‘S’ ‘CHK’ ‘ETX’

‘S’ ASCII-Code 0x53 OR 0x80 = **0xD3**

Beispiel: ADR = 1

String = 0x02 0x81 0xD3 0xD2 0x03 (5 Bytes)

CPU an PC:

‘STX’ ‘ADR’ ‘S’ Info1 Info2 ..... Info9 ‘CHK’ ‘ETX’ (gelesener Wert)

‘S’ ASCII-Code 0x53 OR 0x80 = **0xD3**

Info1 bis Info9 ‘0’ = „AUS“; ‘1’ = „Ein“

je 1 Byte in ASCII OR 0x80 (0xB0 oder 0xB1)

Info1 = Start/Stop
Info2 = Sammelstörung
Info3 = Merker/Softkeys
Info4 = Merker/Softkeys
Info5 = Merker/Softkeys
Info6 = Merker/Softkeys
Info7 = Merker/Softkeys
Info8 = Merker/Softkeys
Info9 = Fehlernummer

Beispiel: ADR = 1, Info1 = 1, Info2 = 0, Info3 = 1, Info4 = 1, Info5 = 0, Info6 = 0, Info7 = 0, Info8 = 0, Info9 = 0

String = 0x02 0x81 0xD3 0xB1 0xB0 0xB1 0xB1 0xB0 0xB0 0xB0 0xB0 0xB0 0xE3 0x03 (14 Bytes)

E.2.11 Digitale Werte stellen

PC an CPU:

‘STX’ ‘ADR’ ‘s’ Index_Wert ‘CHK’ ‘ETX’

‘s’ ASCII-Code 0x73 OR 0x80 = **0xF3**

Index Nummer der Info gemäss Status lesen (Datensatz ‘S’) in ASCII-Code OR 0x80.
Z.B. entspricht der Index 2 der Sammelstörung.

_ Leerzeichen = 0x20 OR 0x80 = 0xA0

Wert ‘1’ oder ‘0’ entsprechend Ein oder Aus

Beispiele: <u>Anlage Ein- / Ausschalten:</u> ADR = 1, Anlage Ein = 1 (Index = 1)

String = 0x02 0x81 0xF3 0xB1 0xA0 0xB1 0xD2 0x03 (8 Bytes)

<u>Fehler quittieren:</u> ADR = 1, Sammelstörung quittieren = 0 (Index = 2)

String = 0x02 0x81 0xF3 0xB2 0xA0 0xB0 0xD0 0x03 (8 Bytes)

CPU an PC:

‘STX’ ‘ADR’ ‘s’ Index ‘CHK’ ‘ETX’

‘s’ ASCII-Code 0x73 OR 0x80 = **0xF3**

E.2.12 Programmstatus lesen

PC an CPU:

‘STX’ ‘ADR’ ‘P’ ‘CHK’ ‘ETX’

‘P’ ASCII-Code 0x50 OR 0x80 = **0xD0**

Beispiel: ADR = 1

String = 0x02 0x81 0xD0 0xD1 0x03 (5 Bytes)

CPU an PC:

‘STX’ ‘ADR’ ‘P’ XXX ‘CHK’ ‘ETX’ (gelesener Wert)

‘P’ ASCII-Code 0x50 OR 0x80 = **0xD0**

XXX Aktuelle Programmnummer (3 Ascii-Zeichen, 001-099)
000 = es läuft kein Programm

Beispiel: ADR = 1, Programm 1 läuft (30Hex oder 80Hex, 30Hex oder 80Hex, 31Hex oder 80Hex)

String = 0x02 0x81 0xD0 0xB0 0xB0 0xB1 0xE0 0x03 (8 Bytes)

E.2.13 Programme starten/stoppen

PC an CPU:

‘STX’ ‘ADR’ ‘p’ XXX ‘CHK’ ‘ETX’

‘p’ ASCII-Code 0x70 OR 0x80 = **0xF0**

XXX Nummer des zu startenden Programms (001 - 099).
000 = Programm stoppen

Beispiele: ADR = 1, Programm 1 starten

String = 0x02 0x81 0xF0 0xB0 0xB0 0xB1 0xC0 0x03 (8 Bytes)

ADR = 1 Programm stoppen

String = 0x02 0x81 0xF0 0xB0 0xB0 0xB0 0xC1 0x03 (8 Bytes)

CPU an PC:

‘STX’ ‘ADR’ ‘p’ XXX ‘CHK’ ‘ETX’ (gelesener Wert)

‘p’ ASCII-Code 0x70 OR 0x80 = **0xF0**

Beispiel: ADR = 1, Programm 1 starten

String = 0x02 0x81 0xF0 0xB0 0xB0 0xB1 0xC0 0x03 (8 Bytes)

E.2.14 Fehlertext auslesen

PC an CPU:

‘STX’ ‘ADR’ ‘F’ ‘CHK’ ‘ETX’

‘F’ ASCII-Code 0x46 OR 0x80 = **0xC6**

Beispiel: ADR = 1

String = 0x02 0x81 0xC6 0xC7 0x03 (5 Bytes)

CPU an PC:

‘STX’ ‘ADR’ ‘F’ TEXT ‘CHK’ ‘ETX’ (gelesener Wert)

‘F’ ASCII-Code 0x46 OR 0x80 = **0xC6**

TEXT = in Steuerung hinterlegter Fehlertext. Länge immer 32 ASCII Zeichen falls kein Fehler vorliegt wird TEXT mit 32 x ‘ ‘ (Blank) zurückgeschickt.

Die Gesamtlänge des Datensatzes beträgt immer 37 Zeichen.
Die Bildung der Checksumme erfolgt in gleicher Weise wie bei den anderen Datensätzen.

E.2.15 weitere Digitalkanäle auslesen (ITC)

PC an CPU:

‘STX’ ‘ADR’ ‘O’ ‘CHK’ ‘ETX’

‘O’ ASCII-Code 0x4F OR 0X80 = **0xCF**

Beispiel : ADR = 1

String = 0x02 0x81 0xCF 0xCE 0x03 (5 Bytes)

CPU an PC:

‘STX’ ‘ADR’ ‘O’ Dig0 Dig1 Dig2 Mx0…Mxn Sk0…Skm ‘CHK’ ‘ETX’ (gelesener Wert)

‘O’ ASCII-Code 0x4F OR 0X80 = **0xCF**

Die Länge der Antwort der CPU hängt von der Konfiguration der Anlage ab. Es werden alle vorhandenen digitalen Kanäle - Merker und Softkeys - zurückgegeben. Ist die Funktion aktiv, wird eine ‘1’, ansonsten eine ‘0’ im Datensatz eingetragen.

Beispiel für die Zuordnung für die Kommission 053003:

Allgemeine Kanäle:

Dig0	=	Start Freigabe
Dig1	=	Fehler steht an
Dig2	=	Unbenutzt

Merkerkanäle für Status-Anzeige:

Mx0	=	Temperatur EIN
Mx1	=	Feuchte EIN
Mx2	=	Taupunkt > 7 °C
Mx3	=	Taupunkt < 7 °C
Mx4	=	Türverriegelung

Setzbare Kanäle (Softkeys):

Sk0	=	Tiefentfeuchtung
Sk1	=	Druckluft
Sk2	=	Zusatzentfeuchtung
Sk3	=	Regelung auf beweglichen PT100
Sk4	=	Digitaler Ausgang 1
Sk5	=	Digitaler Ausgang 2

Beispiel: ADR = 1

PC an CPU:

String = 0x02 0x81 0xCF 0xCE 0x03 (5 Bytes)

CPU an PC:

String = 0x02 0x81 0xCF 0xB0 0xB1 0xB0 0xB0 0xB0 0xB1 0xB0 0xB0 0xB0 0xB0 0xB0 0xB0 0xB0 0xB0 0xB0 0xCE 0x03 (19 Bytes)

E.2.16 weitere Digitalkanäle auslesen (Cadimac)

PC an CPU:

‘STX’ ‘ADR’ ‘O’ ‘CHK’ ‘ETX’

‘O’ ASCII-Code 0x4F OR 0X80 = **0xCF**

Beispiel : ADR = 1

String = 0x02 0x81 0xCF 0xCE 0x03 (5 Bytes)

CPU an PC:

‘STX’ ‘ADR’ ‘O’ Dig0 Dig1 Dig2 Dig3…Dign ‘CHK’ ‘ETX’ (gelesener Wert)

‘O’ ASCII-Code 0x4F OR 0X80 = **0xCF**

Die Länge der Antwort der CPU hängt von der Konfiguration der Anlage ab. Es werden alle vorhandenen digitalen Kanäle zurückgegeben. Die Zuordnung ist aus dem Index des Serviceausdrucks ersichtlich.

Dig0	=	Unbenutzt
Dig1	=	Unbenutzt
Dig2	=	Unbenutzt
Dig3	=	Digitalkanal mit Index 1 (3)
Dig4	=	Digitalkanal mit Index 2 (4)
Dig5	=	Digitalkanal mit Index 3 (5)
Dig6	=	Digitalkanal mit Index 4 (6)
Dig7	=	Digitalkanal mit Index 5 (7)
Dig8	=	Digitalkanal mit Index 6 (8)
Dig9	=	Digitalkanal mit Index 7 (9)
Dig10	=	Digitalkanal mit Index 8 (10)
Dig11	=	Digitalkanal mit Index 9 (11)
Dig12	=	Digitalkanal mit Index 10 (12)
Dig13	=	Digitalkanal mit Index 11 (13)
Dig14	=	Digitalkanal mit Index 12 (14)

Beispiel: ADR = 1

String = 0x02 0x81 0xCF 0xB0 0xB1 0xB0 0xB1 0xB1 0xB0 0xB1 0xB1 0xB1 0xB0 0xB0 0xB1 0xB1 0xB0 0xB1 0xFE 0x03 (20 Bytes)

E.2.17 weitere Digitalkanäle stellen (ITC)

PC an CPU:

‘STX’ ‘ADR’ ‘o’ Index10er Index1er_Wert ‘CHK’ ‘ETX’

‘o’	ASCII-Code 0x6F OR 0x80 = **0xEF**
Index10er	
Index1er	Nummer des Kanals entsprechend der Liste beim Befehl ‘O’. Es wird ab 0 gezählt. Jeweils in ASCII-Code OR 0x80.
_	Leerzeichen = 0x20 OR 0x80 = 0xA0
Wert	‘1’ oder ‘0’ entsprechend Ein oder Aus

Beispiel: Index 9
Index10er = ‘0’ = 0x30 OR 0x80 = 0xB0
Index1er = ‘9’ = 0x39 OR 0x80 = 0xB9
Der Index muss größer gleich 3 + Anzahl der Merkerkanäle sein. Für die Kommission 053003 würde mit dem Index 9 der Kanal Druckluft eingestellt.

Beispiel: Index 9 einschalten: ADR = 1, Druckluft = 1
String = 0x02 0x81 0xEF 0xB0 0xB9 0xA0 0xB1 0xF6 0x03 (9 Bytes)

CPU an PC:

‘STX’ ‘ADR’ ‘o’ Index10er Index1er ‘CHK’ ‘ETX’

‘o’	ASCII-Code 0x6F OR 0x80 = **0xEF**

Beispiel: Index 9 einschalten: ADR = 1, Druckluft = 1
String = 0x02 0x81 0xEF 0xB0 0xB9 0xE7 0x03 (7 Bytes)

E.2.18 weitere Digitalkanäle stellen (Cadimac)

PC an CPU:

‘STX’ ‘ADR’ ‘o’ Index10er Index1er_Wert ‘CHK’ ‘ETX’

‘o’	ASCII-Code 0x6F OR 0x80 = **0xEF**
Index10er	
Index1er	Nummer des Index gemäß Serviceausdruck zweistellig jeweils in ASCII-Code OR 0x80.
_	Leerzeichen = 0x20 OR 0x80 = 0xA0
Wert	‘1’ oder ‘0’ entsprechend Ein oder Aus

Beispiel: Index 9
Index10er = ‘0’ = 0x30 OR 0x80 = 0xB0
Index1er = ‘9’ = 0x39 OR 0x80 = 0xB9
Der Index muss größer gleich 3 sein, sein maximaler Wert hängt von der Kammerkonfiguration ab.

Beispiel: Index 7 einschalten: ADR = 1
String = 0x02 0x81 0xEF 0xB0 0xB7 0xA0 0xB1 0xF8 0x03 (9 Bytes)

CPU an PC:

‘STX’ ‘ADR’ ‘o’ Index10er Index1er ‘CHK’ ‘ETX’

‘o’	ASCII-Code 0x6F OR 0x80 = **0xEF**

E.2.19 Status Tastatursperre lesen

PC an CPU:

‘STX’ ‘ADR’ ‘L’ ‘CHK’ ‘ETX’

‘L’ ASCII-Code 0x4C OR 0x80 = **0xCC**

Beispiel: ADR = 1

String = 0x02 0x81 0xCC 0xCD 0x03 (5 Bytes)

CPU an PC:

‘STX’ ‘ADR’ ‘L’ status ‘CHK’ ‘ETX’ (gelesener Wert)

‘L’ ASCII-Code 0x4C OR 0x80 = **0xCC**

status = ‘0’: Bedienteil freigegeben
= ‘1’: Bedienteil gesperrt Stufe 1
= ‘2’: Bedienteil gesperrt Stufe 2

Beispiel: Bedienteil freigegeben

String = 0x02 0x81 0xCC 0xB0 0xFD 0x03 (6 Bytes)

E.2.20 Tastatur sperren

PC an CPU:

‘STX’ ‘ADR’ ‘l’ Stufe ‘CHK’ ‘ETX’

‘l’ ASCII-Code 0x6C OR 0x80 = **0xEC**

Stufe ‘0’: Tastatur freigeben
‘1’: Tastatur Sperren Ebene 1
‘2’: Tastatur Sperren Ebene 2

Beispiel: ADR = 1; Tastatur sperren Ebene 2

String = 0x02 0x81 0xEC 0xB2 0xDF 0x03 (6 Bytes)

CPU an PC:

‘STX’ ‘ADR’ ‘l’ Stufe ‘CHK’ ‘ETX’

‘l’ ASCII-Code 0x6C OR 0x80 = **0xEC**