

Appendix E Interface protocol CTS control ↔ PC

(Subject to modifications)

E.1 Construction of the protocol

Interface: RS 232
Baud rate: 19,200 bauds
Format: 8 bits, ODD parity (the parity bit completes the total of 1s to an odd figure)
Data flow control: none
Framing: **'STX' 'Date' 'CHK' 'ETX'**
STX = 0x02
ETX = 0x03
CHK = XOR connection of all data (without STX, ETX nor CHK)
The highest bit (bit 7 resp. MSB) of the data and of the CHK always is 1.
Example: ASC „1“ = DEC 49 or DEC 128 = DEC 177 resp.
HEX 0x31 or HEX 0x80 = HEX 0xB1.
ADR = 0x81 - 0xA0 (address 01 - 32); is set through the software.
Default = 0x81 (address 01).

But for „ETX“ and „STX“, the highest bit (MSB) is always 1.

E.2 Commands and responses

E.2.1 Set time

PC to CPU:

'STX' 'ADR' 't' date time 'CHK' 'ETX'
't' ASCII code 0x74 OR 0x80 = **0xF4**
date DDMMYY per byte in ASCII OR 0x80 (6 bytes)
time HHMMSS per byte in ASCII OR 0x80 (6 bytes)
Example: ADR = 1, date = 241196, time = 145535
String = 0x02 0x81 0xF4 0xB2 0xB4 0xB1 0xB1 0xB9 0xB6
0xB1 0xB4 0xB5 0xB5 0xB3 0xFF 0x03 (17 bytes)

CPU to PC:

'STX' 'ADR' 't' date time 'CHK' 'ETX' (set value)
't' ASCII code 0x74 OR 0x80 = **0xF4**

E.2.2 Read time

PC to CPU:

'STX' 'ADR' 'T' 'CHK' 'ETX'
'T' ASCII code 0x54 OR 0x80 = **0xD4**

CPU to PC:

'STX' 'ADR' 'T' date time 'CHK' 'ETX' (set value)
'T' ASCII code 0x54 OR 0x80 = **0xD4**

E.2.3 Set analog parameters

PC to CPU:

'STX' 'ADR' 'a' channelno_value 'CHK' 'ETX'
'a' ASCII code 0x61 OR 0x80 = **0xE1**
channelno. one byte in ASCII OR 0x80
_ blank = 0x20 OR 0x80 = 0xA0
value format XXX.X (for negative values -XX.X)
per byte in ASCII OR 0x80

Example: ADR = 1, channelno. = 0 (temperature channel), value = -14.5°C
String = 0x02 0x81 0xE1 0xB0 0xA0 0xAD 0xB1 0xB4 0xAE 0xB5 0xC3 0x03
(12 bytes)

Remark: If several channels are available, each one must be set separately

CPU to PC:

'STX' 'ADR' 'a' 'CHK' 'ETX'
'a' ASCII code 0x61 OR 0x80 = **0xE1**

E.2.4 Read analog parameters

PC to CPU:

'STX' 'ADR' 'A' channelno 'CHK' 'ETX'
'A' ASCII code 0x41 OR 0x80 = **0xC1**
channelno. one byte in ASCII OR 0x80

Example: ADR = 1, channel no. = 0 (temperature channel)
String = 0x02 0x81 0xC1 0xB0 0xF0 0x03 (6 bytes)

CPU to PC:

'STX' 'ADR' 'A' channelno_actualvalue_setvalue 'CHK' 'ETX' (read value)
'A' ASCII code 0x41 OR 0x80 = **0xC1**
channelno. one byte in ASCII OR 0x80
_ blank = 0x20 OR 0x80 = 0xA0
actualvalue format XXX.X (-XX.X for negative values)
per byte in ASCII OR 0x80
setvalue format XXX.X (-XX.X for negative values)
per byte in ASCII OR 0x80

Example: ADR = 1, channel no. = 0 (temperature channel), actualvalue = -14.5°C,
setvalue = -13.8°C
String = 0x02 0x81 0xC1 0xB0 0xA0 0xAD 0xB1 0xB4 0xAE 0xB5 0xA0
0xAD 0xB1 0xB3 0xAE 0xB8 0xFA 0x03 (18 bytes)

Remarks: If several channels are available, each one must be read separately.

E.2.5 Specifying the rate of change (gradients)

General: The value to be inputted in the commands described below has the unit of measurement K/min. The system will run with this rate of change using the set target value.

Attention: The gradient that is set will remain in effect in the controls until a new value is set. If a run is to be carried out with the maximum target value stepping range then the gradient is to be set to a maximum value of 999.9 K/min.

In order to carry out a ramp using a target value, input the change speed first and then input the desired final value as target value. When the final value is reached, the system will continue to run with this target value until a new value is inputted.

With the ITC controller, the actual target value in the context of the ramp will always be outputted when the command to read the analogue values is given. The command 'E' (see below) is to be used to query the set target value of the ramp.

Note: The value of the gradient can also be inputted in a different format. If, for example, more precise values are necessary for the gradient then values can be inputted with two decimal places.

Example: 00.05 → 0.05 K/min
23.45 → 23.45 K/min

E.2.6 Set gradient ramp up (e.g. heating)

PC to CPU:

```
'STX' 'ADR' 'u' channelno_value 'CHK' 'ETX'  
'u'          ASCII code 0x75 OR 0x80 = 0xF5  
channelno.  one byte in ASCII OR 0x80  
_          blank = 0x20 OR 0x80 = 0xA0  
value      format XXX.X in K/min  
           each byte in ASCII OR 0x80
```

Remarks: If several channels are available, each one must be read separately. The value for the gradient is always a positive one.

CPU to PC:

```
'STX' 'ADR' 'u' 'CHK' 'ETX' (read value)  
'u'          ASCII code 0x75 OR 0x80 = 0xF5
```

E.2.7 Set gradient ramp down (e.g. cooling)

PC to CPU:

```
'STX' 'ADR' 'd' channelno_value 'CHK' 'ETX'  
'd'          ASCII code 0x64 OR 0x80 = 0xE4  
channelno.  one byte in ASCII OR 0x80  
_          blank = 0x20 OR 0x80 = 0xA0  
value      format XXX.X in K/min  
           each byte in ASCII OR 0x80
```

Remarks: If several channels are available, each one must be read separately. The value for the gradient is always a positive one.

CPU to PC:

```
'STX' 'ADR' 'd' 'CHK' 'ETX' (read value)  
'd'          ASCII code 0x64 OR 0x80 = 0xE4
```

E.2.8 Read adjusted gradient

PC to CPU:

'STX' 'ADR' 'U' channelno 'CHK' 'ETX'
'U' ASCII code 0x55 OR 0x80 = **0xD5**
channelno. one byte in ASCII OR 0x80

CPU to PC:

'STX' 'ADR' 'U' channelno_gradientUp_gradientDown 'CHK' 'ETX' (read value)
'U' ASCII code 0x55 OR 0x80 = **0xD5**
channelno. one byte in ASCII OR 0x80
_ blank = 0x20 OR 0x80 = 0xA0
gradientUp format XXX.X in K/min
each byte in ASCII OR 0x80
gradientDown format XXX.X in K/min
each byte in ASCII OR 0x80

Remarks: If several channels are available, each one must be read separately.
The value for the gradient is always a positive one.

E.2.9 Read adjusted final value of the ramp

PC to CPU:

'STX' 'ADR' 'E' channelno 'CHK' 'ETX'
'E' ASCII code 0x45 OR 0x80 = **0xC5**
channelno. one byte in ASCII OR 0x80

CPU to PC:

'STX' 'ADR' 'E' channelno_finalvalue 'CHK' 'ETX' (read value)
'E' ASCII code 0x45 OR 0x80 = **0xC5**
channelno. one byte in ASCII OR 0x80
_ blank = 0x20 OR 0x80 = 0xA0
finalvalue format XXX.X in K/min
each byte in ASCII OR 0x80

Remarks: If several channels are available, each one must be read separately.

E.2.10 Read status

PC to CPU:

'STX' 'ADR' 'S' 'CHK' 'ETX'
'S' ASCII code 0x53 OR 0x80 = **0xD3**
Example: ADR = 1
String = 0x02 0x81 0xD3 0xD2 0x03 (5 bytes)

CPU to PC:

'STX' 'ADR' 'S' Info1 Info2 Info9 'CHK' 'ETX' (read value)
'S' ASCII code 0x53 OR 0x80 = **0xD3**
Info1 to info 9 '0' = „OFF“, '1' = „ON“
per byte in ASCII OR 0x80 (0xB0 or 0xB1)
Info1 = start/stop
Info2 = collective failure
Info3 = indicator/softkeys
Info4 = indicator/softkeys
Info5 = indicator/softkeys
Info6 = indicator/softkeys
Info7 = indicator/softkeys
Info8 = indicator/softkeys
Info9 = Error number

Example: 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 0xE3
0x03 (14 bytes)

E.2.11 Set digital parameters

PC to CPU:

'STX' 'ADR' 's' index_value 'CHK' 'ETX'
's' ASCII code 0x73 OR 0x80 = **0xF3**
index Read the info. number according to the status
(data record „S“ in ASCII code OR 0x80
i.e. the index 2 corresponds to the collective failure.
_ blank = 0x20 OR 0x80 = 0xA0
value „1“ or „0“ correspond to ON or OFF

Examples:

Switch the installation on/off: ADR = 1, installation ON = 1 (index = 1)
String = 0x02 0x81 0xF3 0xB1 0xA0 0xB1 0xD2 0x03 (8 bytes)

Acknowledge error: ADR = 1, acknowledge collective failure = 0 (index 2)
String = 0x02 0x81 0xF3 0xB2 0xA0 0xB0 0xD0 0x03 (8 bytes)

CPU to PC:

'STX' 'ADR' 's' index 'CHK' 'ETX'
's' ASCII code 0x73 OR 0x80 = **0xF3**

E.2.12 Read program status

PC to CPU:

'STX' 'ADR' 'P' 'CHK' 'ETX'
'P' ASCII code 0x50 OR 0x80 = **0xD0**

Example: ADR = 1
String = 0x02 0x81 0xD0 0xD1 0x03 (5 bytes)

CPU to PC:

'STX' 'ADR' 'P' XXX 'CHK' 'ETX' (read value)
'P' ASCII code 0x50 OR 0x80 = **0xD0**
XXX actual program number (3 ASCII characters, 001-099)
000 = no program currently running

Example: ADR = 1, program 1 runs (30hex or 80 hex, 30 hex or 80 hex,
31 hex or 80 hex)
String = 0x02 0x81 0xD0 0xB0 0xB0 0xB1 0xE0 0x03 (8 bytes)

E.2.13 Start/stop program

PC to CPU:

'STX' 'ADR' 'p' XXX 'CHK' 'ETX'
'p' ASCII code 0x70 OR 0x80 = **0xF0**
XXX number of the program to be started (001-099)
000 = stop the program

Examples:

ADR = 1, start program 1
String = 0x02 0x81 0xF0 0xB0 0xB0 0xB1 0xC0 0x03 (8 bytes)

ADR = 1, stop program
String = 0x02 0x81 0xF0 0xB0 0xB0 0xB0 0xC1 0x03 (8 bytes)

CPU to PC:

'STX' 'ADR' 'p' XXX 'CHK' 'ETX' (read value)
'p' ASCII code 0x70 OR 0x80 = **0xF0**

Example: ADR = 1, start program 1
String = 0x02 0x81 0xF0 0xB0 0xB0 0xB1 0xC0 0x03 (8 bytes)

E.2.14 Read error text

PC to CPU:

'STX' 'ADR' 'F' 'CHK' 'ETX'
'F' ASCII code 0x46 OR 0x80 = **0xC6**

Example: ADR = 1
String = 0x02 0x81 0xC6 0xC7 0x03 (5 bytes)

CPU to PC:

'STX' 'ADR' 'F' text 'CHK' 'ETX' (read value)
'F' ASCII code 0x46 OR 0x80 = **0xC6**
text text for the error memorized in the control unit.
Length always 32 ASCII characters.
If there is no error, TEXT is sent back with 32x" (blank).

The total length of the data record always is 37 characters.

The formation of the check sum occurs in the same way than for the other data records.

E.2.15 Read additional digital parameters (ITC-Controller)

PC to CPU:

'STX' 'ADR' 'O' 'CHK' 'ETX'
'O' ASCII code 0x4F OR 0x80 = **0xCF**

Example: ADR = 1
String = 0x02 0x81 0xCF 0xCE 0x03 (5 bytes)

CPU to PC:

'STX' 'ADR' 'O' dig0 dig1 dig2 Mx0...Mxn Sk0...Skm 'CHK' 'ETX' (read value)
'O' ASCII code 0x4F OR 0x80 = **0xCF**

The length of the replied string depends on the configuration of the chamber.
All configured digital channels, flags and softkeys are reported.
When function active a '1' otherwise a '0' is replied.

Example of a configuration:

General channels:

dig0 = Start Enable
dig1 = Error pending
dig2 = unused

Status bits (flags):

Mx0 = Temperature ON
Mx1 = Humidity ON
Mx2 = dew point > 7°C
Mx3 = dew point < 7°C
Mx4 = door closed

Settable channels (softkeys):

Sk0 = deep dehumidification
Sk1 = compressed air
Sk2 = additional dehumidification
Sk3 = regulating on mob. PT100
Sk4 = Digital Output 1
Sk5 = Digital Output 2

Example: ADR = 1

PC to CPU:

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

CPU to PC:

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

E.2.16 Read additional digital parameters (Cadimac-Controller)

PC to CPU:

'STX' 'ADR' 'O' 'CHK' 'ETX'
'O' ASCII code 0x4F OR 0x80 = **0xCF**

Example: ADR = 1
String = 0x02 0x81 0xCF 0xCE 0x03 (5 bytes)

CPU to PC:

'STX' 'ADR' 'O' dig0 dig1 dig2 dig3...dign 'CHK' 'ETX' (read value)
'O' ASCII code 0x4F OR 0x80 = **0xCF**

The length of the answer of the CPU depends on the configuration of the unit.
All available digital channels are returned. The coordination is evident out of the index of the service expression.

dig0 = unused
dig1 = unused
dig2 = unused
dig3 = digital channel with index 1 (3)
dig4 = digital channel with Index 2 (4)
dig5 = digital channel with Index 3 (5)
dig6 = digital channel with Index 4 (6)
dig7 = digital channel with Index 5 (7)
dig8 = digital channel with Index 6 (8)
dig9 = digital channel with Index 7 (9)
dig10 = digital channel with Index 8 (10)
dig11 = digital channel with Index 9 (11)
dig12 = digital channel with Index 10 (12)
dig13 = digital channel with Index 11 (13)
dig14 = digital channel with Index 12 (14)

Example: 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 Set additional digital parameters (ITC-Controller)

PC to CPU:

'STX' 'ADR' 'o' index10er index1er_value 'CHK' 'ETX'
'o' ASCII code 0x6F OR 0x80 = **0xEF**
index10er
index1er Number of channel according list at command 'O' counting from zero.
In ASCII-Code OR 0x80
_ blank = 0x20 OR 0x80 = 0xA0
value '1' or '0' according to ON or OFF

Example: index = 9
index10er = '0' = 0x30 OR 0x80 = 0xB0
index1er = '9' = 0x39 OR 0x80 = 0xB9
Index must be greater or equal 3 + number of flags. In the example above
index = 9 would match channel compressed air.

Example: index = 9 turn on: ADR = 1, compressed air = 1
String = 0x02 0x81 0xEF 0xB0 0xB9 0xA0 0xB1 0xF6 0x03 (9 bytes)

CPU to PC:

'STX' 'ADR' 'o' index10er index1er 'CHK' 'ETX'
'o' ASCII code 0x6F OR 0x80 = **0xEF**

Example: index = 9 turn on: ADR = 1, compressed air = 1
String = 0x02 0x81 0xEF 0xB0 0xB9 0xE7 0x03 (7 Bytes)

E.2.18 Set additional digital parameters (Cadimac-Controller)

PC to CPU:

'STX' 'ADR' 'o' index10er index1er_value 'CHK' 'ETX'
'o' ASCII code 0x6F OR 0x80 = **0xEF**
index10er
index1er A two-digit number of index - compare the service
expression - in ASCII-Code OR 0x80
_ blank = 0x20 OR 0x80 = 0xA0
value '1' or '0' according to ON or OFF

Example: index = 9
index10er = '0' = 0x30 OR 0x80 = 0xB0
index1er = '9' = 0x39 OR 0x80 = 0xB9
The index must be greater than 3, the maximum is conditioned
of the configuration of the chamber

Example: index = 7 turn on: ADR = 1
String = 0x02 0x81 0xEF 0xB0 0xB7 0xA0 0xB1 0xF8 0x03 (9 bytes)

CPU to PC:

'STX' 'ADR' 'o' index10er index1er 'CHK' 'ETX'
'o' ASCII code 0x6F OR 0x80 = **0xEF**

E.2.19 Read lock state of keyboard

PC to CPU:

'STX' 'ADR' 'L' status 'CHK' 'ETX'
'L' ASCII code 0x4C OR 0x80 = **0xCC**

Example: ADR = 1
String = 0x02 0x81 0xCC 0xCD 0x03 (5 Bytes)

CPU to PC:

'STX' 'ADR' 'L' status 'CHK' 'ETX'
'L' ASCII code 0x4C OR 0x80 = **0xCC**
status = '0': keyboard unlocked
= '1': keyboard locked level 1
= '2': keyboard locked level 2

Example: ADR = 1; unlock keyboard
String = 0x02 0x81 0xCC 0xB0 0xFD 0x03 (6 Bytes)

E.2.20 Lock/unlock keyboard

PC to CPU:

'STX' 'ADR' 'I' level 'CHK' 'ETX'
'I' ASCII code 0x6C OR 0x80 = **0xEC**
level = '0': unlock keyboard
= '1': lock keyboard level 1
= '2': lock keyboard level 2

Example: ADR = 1; Lock keyboard level 2
String = 0x02 0x81 0xEC 0xB2 0xDF 0x03 (6 Bytes)

CPU to PC:

'STX' 'ADR' 'I' level 'CHK' 'ETX'
'I' ASCII code 0x6C OR 0x80 = **0xEC**