

14.3 Kommunikations-Protokoll



Achtung!

Dieses Kommunikations-Protokoll gilt ab der DMR-Version 00.37.

Wenn Sie unsere Bedienungssoftware für den PC nicht nutzen möchten, können Sie den Prüfschrank trotzdem mit Ihrem PC steuern. Hierfür benötigen Sie jedoch *Programmierkenntnisse*.

Das DMR des Prüfschranks ist mit einer RS232C-Schnittstelle ausgestattet. Für die Datenübertragung vom DMR zum PC gelten folgende Eigenschaften:

- 9600 / 19200 Baud
- 1 Startbit
- 8 Datenbit
- 1 Stopbit
- keine Parität
- kein Handshaking

Wie funktioniert die Kommunikation zwischen PC und DMR des Prüfschranks?

Sie müssen ein Steuerungsprogramm schreiben.

- ✦ Verwenden Sie eine gebräuchliche Programmiersprache (z.B. TurboPascal oder C++)



Hinweis!

Für TurboPascal und C++ können wir Ihnen die entsprechenden Treiber liefern.



Achtung!

Die Kommunikation zwischen PC und dem DMR des Prüfschranks funktioniert nur, wenn Sie den Programmteil „Prüfsumme“ mit Ihrem Steuerungsprogramm verknüpfen.

- ✦ Verknüpfen Sie den Programmteil „Prüfsumme“ mit Ihrem Steuerungsprogramm (↔ Seite 106).

Funktionsweise des Programms

Das Programm sendet einen String an das DMR. Durch diesen Sende-String fragt das Programm Daten ab. Das DMR antwortet auf diesen String, indem er einen Antwort-String an den PC zurücksendet.

... →

Ein String enthält eine Folge von ASCII-Zeichen (z.B. T45H/3&). Jeder String *beginnt* mit dem Code für „start of text (=STX)“ und *endet* mit dem Code für „end of text (=ETX)“. Für diese Codes werden ASCII-Zeichen verwendet:

{STX} = ASCII-Code 02

{ETX} = ASCII-Code 03

- Setzen Sie bei Ihrem Programm den ASCII-Code 02 *vor* einen String und den ASCII-Code 03 *hinter* einen String.

Die Antwort-Strings enthalten zwei weitere Zeichen: ACK (acknowledged) *oder* NAK (not acknowledged). Acknowledged bedeutet, daß das DMR den Sender-String vom PC erkannt und akzeptiert hat. Not acknowledged bedeutet, daß der String *nicht* erkannt und demzufolge auch nicht akzeptiert wurde; im zweiten Fall muß das Kommando wiederholt werden. Die ASCII-Codes lauten:

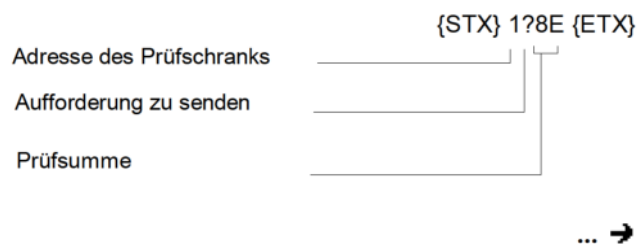
{ACK} = ASCII-Code 06

{NCK} = ASCII-Code 15

Beispiel für...

...einen Sende-String und entsprechenden Antwort-String:

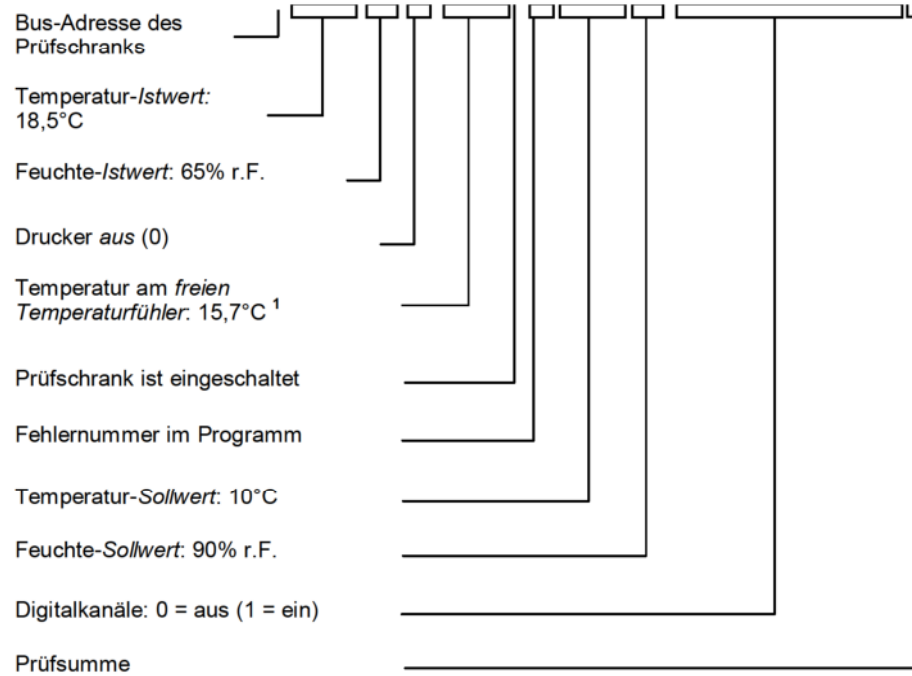
Sende-String (PC → DMR)



Antwort-String (DMR → PC)

Der folgende String enthält Informationen über Ist- und Sollwert von Temperatur und Feuchte sowie weitere Einstellwerte. Die Werte beziehen sich auf den momentanen Zeitpunkt:

```
{STX} 1T018.5F65POT015.7#11T010.0F90R10000000000000014 {ETX}
```



¹ Wenn Sie keinen freien Temperaturfühler haben, steht an dieser Stelle: T-99.9

Was können Sie noch mit Ihrem PC steuern?

Mit dem PC können Sie ...

- Sollwerte für Temperatur und Feuchte einstellen
- Temperatur von mehreren freien Fühlern abfragen
- Prüfprogramm starten
- Anzahl der Wiederholungen des Prüfprogramms bestimmen
- Prüfprogramm stoppen

Damit Sie diese Funktionen ausführen können, müssen Sie einen Sende-String an das DMR senden. In Tabelle 14.2 sind die entsprechenden Strings aufgeführt:

Tabelle 14.2:

Sende-Strings

z: 1...9 (Adresse des Prüfschranks)

CC: Platzhalter für Prüfsumme

T025.0: Temperatur-Sollwert = 25,0°C

F35: Feuchte-Sollwert = 35% r.F.

R100...0: Digitalkanäle 1 - 16; 1=ein, 0=aus

Funktion	Sende-String
1 Sollwerte einstellen	zT025.0F35R1000000000000000000CC
2 Temperatur von freien Fühlern abfragen	z:Get:P_Var:xx:CC xx: 83, 84 oder 85 ¹
3 Prüfprogramm starten	z:Set:AutoStart:x:CC x = 1...100 (Nummer des Prüfprogramms)
4 Wiederholungen des Prüfprogramms bestimmen	z:Set:AutoLoop:x:CC x = 1...9999 (Anzahl der Wiederholungen)
5 Prüfprogramm stoppen	z:Set:AutoStop:CC

**Achtung!**

Jeder String beginnt mit einem „start of text (=STX)“ und endet mit einem „end of text (ETX)“; STX, ETX, ACK und NAK werden als ASCII-Codes dargestellt (⇒ Seite 101).

Antwort-Strings auf Sende-Strings 1,3,4 und 5:

Wenn das DMR den Sende-String vom PC erkennt, sendet das DMR den Antwort-String:

¹ xx=83 entspricht dem freien Temperaturfühler, dessen Wert Sie auch mit dem Sende-String {STX} 1?8E {ETX} abfragen können.

{STX} z{ACK}CC {ETX}

Wenn das DMR den Sende-String nicht erkennt, sendet das DMR:

{STX} z{NAK}CC {ETX}

(z = Adresse des Prüfschranks; CC = *Platzhalter* für Prüfsumme)

... →

Antwort-String auf Sende-String 2:

Beispiel

```
{STX} z:Get:P_Var:xx: 20.4:CC {ETX}
```



Hinweis!

Die Prüfsumme ist eine Art Quersumme über die ASCII-Werte eines Strings, einschließlich des-ASCII-Werts für STX (→ Seite 106). Der ASCII-Wert für den „ETX“ und die Prüfsumme werden nicht addiert. Die Prüfsumme wird in Großbuchstaben angegeben, z.B. 8E.

Bedingungen an das Steuerungsprogramm

- Alle Sende-Strings müssen die Prüfsumme enthalten.
- Bevor das DMR einen Antwort-String sendet, synchronisiert das DMR den empfangenen Sende-String auf die Zeichen STX und ETX.
- Zusätzlich kontrolliert das DMR die Prüfsumme des Sende-Strings, um Übertragungsfehler zu vermeiden.
- Wenn eine Einstellung gültig ist (z.B. Sollwert ist richtig), wird der Sende-String akzeptiert. Der Antwort-String lautet:

```
{STX} z{ACK}CC {ETX}.
```

Wenn eine Einstellung ungültig ist (z.B. Sollwert ist falsch), sendet das DMR:

```
{STX} z{NAK}CC {ETX}
```

- Wenn der gleiche Sende-String mehrmals nicht erkannt wird, erscheint am PC eine Fehlermeldung.



Achtung!

Das DMR verarbeitet Daten langsamer als ein PC. Deshalb dürfen nicht mehrere Strings in kurzen Zeitabständen zum DMR gesendet werden, weil sonst die internen Abläufe im DMR (Steuern und Regeln) gestört werden können.

- Senden Sie nicht mehr als einen String pro fünf Sekunden.

Ausnahme: Wenn ein Sende-String nicht akzeptiert wird und wiederholt zum DMR gesendet wird, dann können mehr als ein Kommando pro fünf Sekunden gesendet werden.

Programmteil „Prüfsumme“

Bei der Prüfsumme handelt es sich um das 256-Komplement des Modulo-256-Divisionsrestes über die ASCII-Werte aller Zeichen im String; die Werte für ETX und die Prüfsumme selbst werden nicht addiert.

Beispiel

Programmiersprache: Turbo Pascal 5.0

```
FUNCTION PRUEFSUMME
(P RUEFSTR: STRING): STRING;

VAR L, I      : INTEGER;
    B, J, K   : BYTE;
    CH       : STRING[1];
BEGIN
  L := LENGTH(P RUEFSTR);
  B := 0;
  FOR I:= 1 TO L DO BEGIN
    CH := COPY(P RUEFSTR, I, 1);
    J := ORD(CH[1]);
    B := B - J;
  END;
  J := B DIV 16;
  IF J<10
  THEN J:= J+48
  ELSE J:= J+55;
  K := B MOD 16;
  IF K<10
  THEN K:= K + 48
  ELSE K:= K + 55;
  PRUEFSUMME:= CHR(J) + CHR(K);
END;
```



Hinweis!

Damit Sie kontrollieren können, ob die Prüfsumme richtig berechnet wird, sollten Sie den Sende-String des Beispiels auf Seite 101 durchführen. Für die Prüfsumme müssen Sie "14" erhalten (⇔ Antwort-String auf Seite 102).