

# Transaktionsprotokoll auf der Seriellen Schnittstelle voco Call-Button

Detlef Reil, ADE, 31. Juli 2006

## Zielsetzung

Verschiedenste Geräte sollen über die Serielle Schnittstelle des **voco** angeschlossen werden können. Die Art der Funktion und des zu verwendenden Protokolls soll automatisch erkannt werden (keine Konfiguration notwendig). Die Kommunikation soll auch zu den bestehenden Protokollen (Scanner mit Enquiry-Protokoll, FBConfig, Coligo, usw.) kompatibel sein.

## A. Schnittstelle, Konfiguration

Das beschriebene Transaktionsprotokoll basiert auf asynchroner Datenübertragung nach RS232 mit 9600 Baud (8,n,1) am Sub-D9-Steckverbinder des **voco**. Er muss für den Kompatibilitätsmodus mit folgenden Einstellungen konfiguriert sein:

- 9600 Baud
- Barcode-Scanner-Enquiry

## B. Signale

**voco** kann über die eingebaute Batterie oder auch extern versorgt werden. Bei Anschluss einer externen Stromversorgung über den Sub-D9-Steckverbinder MUSS die Batterie ausgebaut werden, denn es gibt keine Sperrdiode in der Batterieversorgung, damit die Batterie eine optimale Betriebszeit erreicht. Die Stromversorgung wird über die Pins 9 und 5 angeschlossen.

Belegung (**voco** als DCE):

- 1: reserviert
- 2: RxD (Daten vom **voco**)
- 3: TxD (Daten zum **voco**)
- 4: reserviert
- 5: GND
- 6: DSR (Knopf ist an)
- 7: RTS (Wecksignal)
- 8: reserviert
- 9: + (5V bis 6,4V externe Versorgung) NUR BEI AUSGEBAUTER BATTERIE!

## **C. Transaktionsprotokolle**

Folgende Transaktionsprotokolle werden unterstützt:

- 1) Scanner
- 2a) Protokoll, ungesichert
- 2b) Protokoll, gesichert

In allen Kommunikationsarten aktiviert das angeschlossene Gerät das RTS-Signal und weckt damit **voco** auf. Dieser sendet nach einigen 100ms kontinuierlich ENQ-Zeichen (ASCII: 0x05), bis das angeschlossene Gerät die Kommunikation startet. Nach dem Senden des ersten Telegramms nimmt es RTS sofort zurück, ansonsten wird durch erneutes Senden von ENQ ein neuer Vorgang gestartet (außer bei 1. Scanner).

### **1) Scanner-Kommunikation**

- Angeschlossenes Gerät sendet nach Empfang des ENQ-Zeichens ohne weitere Steuerzeichen einen String; min. 3, max. 32 druckbare Zeichen (0x20 bis 0x7E)
- Der gültige String wird zum Host gesendet, sobald ein CR (0x0D) folgt oder das RTS Signal deaktiviert wird. Wenn bis zur Rücknahme von RTS kein gültiger String empfangen wurde, wird nichts versendet und **voco** schläft wieder ein
- Es erfolgt keine Rückmeldung zum angeschlossenen Gerät

### **2) Universalprotokoll**

a) ungesichertes Protokoll

- Angeschlossenes Gerät sendet nach Empfang des ENQ-Zeichens zunächst ein DC1-Steuerzeichen (0x11)
- nun wie unter 1) beschrieben einen gültigen String
- Nach erfolgter Hostkommunikation sendet **voco** eine Rückmeldung zum angeschlossenen Gerät:
  - o Wenn alles OK („grün“):
    - <0x01>
    - Wenn vorhanden, String aus „Rueckmeldung“ in HTTP-Response, max. 20 Zeichen
    - CR <0x0D>

## **voco** – Transaktionsprotokoll der Seriellen Schnittstelle

- Oder bei Netzwerkfehler („rot“):
  - <0x00>
  - <0x01> - <0xFF> **voco** Fehlernummer wie im Display, siehe Dokumentation
  - CR <0x0D>
  
- Oder bei „Lampenstatus“ in HTTP-Response vom Host ungleich 0 („rot“):
  - <0x02>
  - <0x01> - <0xFF> Lampenstatus
  - Wenn vorhanden, String aus „Rueckmeldung“ in HTTP-Response, max. 20 Zeichen
  - CR <0x0D>

### b) gesichertes Protokoll

- Angeschlossenes Gerät sendet nach Empfang des ENQ-Zeichens:
  - DC1 (0x11)
  - STX (0x02)
  - String (0 bis 32 Zeichen zwischen 0x20 und 0xFF)
  - ETX (0x03)
  - Prüfbyte (s.u.)
  
- **voco** antwortet:
  - ACK (0x06) wenn Prüfbyte OK; Host-Kommunikation findet statt
  - NAK (0x15) wenn Prüfbyte nicht OK; angeschlossenes Gerät muss wiederholen: <STX> String <ETX> Prüfbyte
  
- Nach erfolgter Hostkommunikation sendet **voco** eine Rückmeldung zum angeschlossenen Gerät:
  - Wenn alles OK („grün“):
    - <0x01>
  
  - Oder bei Netzwerkfehler („rot“):
    - <0x00>
    - <0x01> - <0xFF> **voco** Fehlernummer wie im Display, siehe Dokumentation
  
  - Oder bei „Lampenstatus“ in HTTP-Response vom Host ungleich 0 („rot“):
    - <0x02>
    - <0x01> - <0xFF> Lampenstatus

## **voco** – Transaktionsprotokoll der Seriellen Schnittstelle

- Zusätzlich (außer bei Netzwerkfehler) Text aus „Rueckmeldung“ in HTTP-Response, wenn vorhanden:
  - STX <0x02>
  - String aus Rückmeldung, max. 20 Zeichen
  - ETX <0x03>
  - Prüfbyte (s.u.)

Angeschlossenes Gerät muss bestätigen:

- ACK (0x06) wenn verstanden (Prüfbyte OK)
- NAK (0x15) wenn nicht verstanden, **voco** wiederholt: <STX> String <ETX>

Wenn nach 3 Sekunden weder ACK noch NAK kommt, wird die Übertragung als korrekt angesehen und **voco** schläft ein, bzw. startet einen neuen Vorgang, wenn RTS noch aktiv ist

### **Berechnung des Prüfbytes: Alle Zeichen von STX bis ETX per EXOR akkumulieren:**

```
bool IsPruefByteOK(const unsigned char* buffer) {
    unsigned char pruefbyte = 0;
    for (int i=0; buffer[i]!=0x03; i++)
        pruefbyte ^= buffer[i];
    return pruefbyte==buffer[i];
}
```