

---

# **RS232-Schnittstellen- Beschreibung**

**für**

**ergo\_bike 2002**

**Version 1.4.4**

**Dezember 2003**

---



© Copyright 2003  
**daum electronic gmbh**  
Flugplatzstr. 100  
D-90768 Fürth

Alle Rechte, auch die der Übersetzung, behalten wir uns vor.  
Änderungen behalten wir uns vor. Die Firma **daum electronic gmbh** übernimmt für eventuelle Fehler und deren Konsequenzen keine Haftung.

**Daten vom Cockpit zum PC :**

- aktuelle Wattleistung in 5-Schritten
- aktuelle RPM-Drehzahl
- RPM-Kennlinien Zustand
- die Momentan-Geschwindigkeit
- die zurückgelegte Distanz
- die gefahrene Zeit
- die verbrauchten KJoule
- das eingestellte Programm
- die angegebene Person
- den aktuellen Pulswert
- den Pulszustand (Up-Ok-Down-Alarm)
- den Relax-Wert
- den Relax-Modi (Eineichen oder Messen)
- den aktuellen Gang
- die Cockpit-Version
- Temperatur (zukünftiges Feature)

**Daten vom PC zum Cockpit :**

- die Person (Limit-Daten, Cardio-Puls, Drehzahl und Programm)
- das gewünschte Programm
- die gewünschte Watt-Leistung (IL 1)
- den aktuellen Gang
- den möglichen Ziel-Puls (IP 2)
- die Soll-Geschwindigkeit (IR 3)

**Übertragungsparameter:**

9600,8,n,1 : 9600 Baud, 8 Bit/Byte, keine Parität, 1 Stopbit.

Die Daten werden in Form von einzelnen Datenpakete übertragen, die einen Mindestabstand (Mindestpause) von 50 ms aufweisen müssen. Das Cockpit antwortet meist innerhalb 50 ms und garantiert innerhalb 1 s, falls im Cockpit nicht ein Menü aufgerufen wurde.

Zum Anschluss an die serielle Schnittstelle eines Rechners muss das Update-Kabel verwendet werden.

Falls die Übertragung über die Onboard-Schnittstelle des Rechners unzuverlässig ist, sollte ein USB-RS232-Adapter wie UC-232A verwendet werden.

**Datenverkehr :**

Die nachfolgende Tabelle soll den anfallenden Datenstream sowohl für die Stellparameter als auch für die momentanen Cockpit-Daten aufzeigen. Eine Erklärung für die einzelne Byte-Struktur wird nachfolgend erklärt.

<i>PC</i>		<i>B</i>	<i>Cockpit</i>	<i>Bezeichnung</i>
<b>Befehl</b>	<b>Daten</b>		<b>Daten</b>	
0x10	Adr		0x10,adr,0x??	Check_Cockpit
0x11			0x11,adr	Get Adress
0x12	Adr		0x12,adr	Reset device
0x21	Adr		0x21, adr,0x00 (0x01)	Start_Prog
0x22	Adr		0x22, adr,0x00 (0x01)	Stop_Prog
0x23	Adr,0x??		0x23, adr,0x??,0x00 (0x01)	Set_Prog
0x24	Adr,0x??,.....,0x??		0x24, adr,0x??,....,0x??	Set_Person
0x25	Adr		0x25,adr, 0x00 (0x01)	Start_Relax
0x30	Adr	S	0x30, adr	Wakeup
0x31	Adr	S	0x31, adr	Sleep
0x32	Adr	S	0x32, adr	Get_Temperatur
0x40	Adr		0x40/0x41, adr,.....,0x??	Run_Daten
0x42	Adr		0x42, adr,0x00(0x01),.....,0x??	Relax_Daten
0x51	Adr,0x??		0x51, adr,0x??	Set_Watt
0x52	Adr,0x??		0x52, adr,0x??	Set_Puls
0x53	Adr,0x??		0x53, adr,0x??	Set_Gang
0x54	Adr,0x??		0x54, adr,0x??	Set_Speed
0x60	Adr,0x??		0x60,adr	Set_Language
0x61	Adr		0x61, adr,0x??	Get_Language
0x62	Adr,0x?? ,0x?? ,0x??		0x62,adr	Set_Timer
0x63	Adr		0x63, adr,0x??, 0x??, 0x??	Get_Timer
0x64	Adr,0x?? ,0x?? ,0x??		0x64,adr	Set_Date
0x65	Adr		0x65, adr,0x??, 0x??, 0x??	Get_Date
0x66	Adr		0x66, adr,0x??	Get_Prog
0x67	Adr		0x67, adr,0x??	Get_Person
0x68	Adr,0x00(0x01)		0x68,adr,0x00(0x01)	Set_Watt_Profile
0x70	Adr,0x??		0x70,adr	Del_Person
0x71	Adr,0x??		0x71,adr	Del_Iprog
0x73	Adr		0x73, adr,0x??,....., 0x??	Get_Version
0x74	Adr,block		0x74, adr,block,....., 0x??	Get_Person_Data
0x75	Adr,block		0x75, adr,block,....., 0x??	Get_Person_Data1
0xD3	0x??, 0x??		0xD3,adr	Ctrl_Ton

Bemerkungen (Spalte B):

S: Funktioniert korrekt und stabil auch nach Tests, ist aber bisher nur in Test-Versionen aktiviert.

Das Cockpit ist passiv, sendet also nur auf Anfrage.

Man beachte bitte, daß generell jeder Sendestring vom PC aus zumindest mit dem Befehl vom Cockpit quittiert wird. Bei manchen Sendestrings werden zusätzlich noch Informationsdaten an den Befehl angehängt.

Die übertragenen Werte sind Ganzzahlen im Little-Endian-Format, die generell vorzeichenlos sind und bei Rundung/Cast (im Cockpit) generell abgerundet werden.

Generell werden SI-Einheiten verwendet.

In den Erläuterungen zu den einzelnen Befehlen bedeutet das + bei den Bytes der Datenpakete nicht Addition sondern dass die beiden Bytes low und high Byte eines Wertes sind.

Das Cockpit kann vom PC nur im Programm 0 (freies Trainig) richtig gesteuert werden, weil sonst das Cockpit steuern würde.

Wichtig: Wenn der Mindest-Abstand zwischen den Datenpaketen nicht eingehalten wird oder die Befehle nicht richtig verwendet werden, können Daten oder Code im Cockpit überschrieben werden! Im Update-Modus werden andere Befehle verwendet.

Check Cockpit:

dieser Befehl ist lediglich zum Prüfen vorhanden, ob es ein ganz altes oder ein neues Ergo\_bike-Cockpit ist. Das beruht auf der Tatsache, daß das alte Cockpit diesen Befehl einfach ignoriert und nicht antwortet.

Sendestring-PC	0x10	*adr	
Länge	2 Bytes		
Empfangsstring-PC	0x10	*adr	*1
Länge	3 Bytes		

\*adr Cockpit-Adresse

\*1 aktueller Versionsstand

Get Adress:

dieser Befehl dient zum Abfragen der derzeitigen Cockpit-Adresse.

Sendestring-PC	0x11	
Länge	1 Bytes	
Empfangsstring-PC	0x11	*adr
Länge	2 Bytes	

\*adr Cockpit-Adresse

**Reset Device:**

dieser Befehl dient zum softwaremäßigen Reseten des Cockpitss und ist nicht zu vergleichen mit einem Hardware Reset !

Sendestring-PC	0x12	*adr	
Länge	2 Bytes		
Empfangsstring-PC	0x12	*adr	
Länge	2 Bytes		

\*adr Cockpit-Adresse  
\*1 aktueller Versionsstand

**Start Prog :**

dieser Befehl teilt dem Cockpit mit, daß der Anwender ein neues Programm fahren möchte und es somit seine Variablen in einen Grundzustand bringt. Der Anwender (PC-Benützer) bekommt auf Grund des 3-ten Bytes die Information, ob er die Pedale noch in Benutzung hat oder nicht. Erst wenn er nicht mehr tritt, werden die weiteren Sequenzen zum Starten eines neuen Programmes gesendet und die Variablen gelöscht.

Sendestring-PC	0x21	*adr	
Länge	2 Bytes		
Empfangsstring-PC	0x21	*adr	*1
Länge	3 Bytes		

\*adr Cockpit-Adresse  
\*1 0x00 entspricht dem Zustand „Nicht\_Treten“  
0x01 entspricht dem Zustand „Treten“

**Stop Prog :**

dieser Befehl teilt dem Cockpit mit, daß der Anwender seinen Programmablauf gestoppt hat. Dies kann er jedoch nur wenn die Tretinformation 0x00 ist. Das bedeutet, daß der Benutzer nicht mehr trampelt.

Sendestring-PC	0x22	*adr	
Länge	2 Bytes		
Empfangsstring-PC	0x22	*adr	*1
Länge	3 Bytes		

\*adr Cockpit-Adresse  
\*1 0x00 entspricht dem Zustand „Nicht\_Treten“  
0x01 entspricht dem Zustand „Treten“

**Set Prog :**

dieser Befehl stellt am Cockpit ein gewünschtes Programm ein. Dieses Programm kann nur eines der freigeschalteten sein. Außerdem gibt es nur Programme von 0..79.

Sendestring-PC	0x23	*adr	*1	
Länge	3 Bytes			
Empfangsstring-PC	0x23	*adr	*1	*2
Länge	4 Bytes			

- \*adr Cockpit-Adresse  
 \*1 entspricht dem gewünschten Programm !  
 \*2 0x00 entspricht dem Zustand „Nicht\_Treten“  
 0x01 entspricht dem Zustand „Treten“

**Set Person :**

hiermit wird die gewünschte Person eingestellt. Person kann sein der Gast = 0 oder die Personen 1..4. Andere können nicht eingestellt werden.

Sendestring-PC	0x24	*adr	*1	*2	*3	*4	*5	*6	*7	*8	*9	*0	*a	*b
Länge	15 Bytes													
			*c											
Empfangsstring-PC	0x24	*adr	*1	*2	*3	*4	*5	*6	*7	*8	*9	*0	*a	*b
Länge	16 Bytes													
			*c	*d										

- \*adr Cockpit-Adresse  
 \*1 entspricht der gewünschten Person !  
 \*2 entspricht dem Alter der Person, kann sein 0,10..99  
 \*3 entspricht dem Geschlecht 0=Male 1=Female, 2=Unbekannt  
 \*4 entspricht der Körpergröße, ist 100..220  
 \*5 entspricht dem Körpergewicht, ist 10..250  
 \*6 entspricht dem Körperfett, ist 0..110  
 \*7 entspricht der Leistungseinstufung (Coaching), ist 0..3  
 \*8 entspricht der Trainingshäufigkeit (Coaching), ist 3..7  
 \*9 entspricht dem gewünschten Limit-Watt, ist 0,5..80(160)  
 \*0 entspricht dem gewünschten Limit-Puls, ist 0,80..199  
 \*a ist die Limit-Zeitangabe, maximal 0,1..99  
 \*b spiegelt die Limit-Distanz wieder, ist 0..99  
 \*c ist die Limit-Kcalangabe, kann sein 0..199  
 \*d 0x00 entspricht dem Zustand „Nicht\_Treten“  
 0x01 entspricht dem Zustand „Treten“

**Start Relax :**

dieser Befehl startet den Relaxvorgang.

Sendestring-PC	0x25	*adr	
Länge	2 Bytes		
Empfangsstring-PC	0x25	*adr	*1
Länge	3 Bytes		

- \*adr Cockpit-Adresse  
 \*1 0x00 entspricht dem Zustand „Nicht\_Treten“  
 0x01 entspricht dem Zustand „Treten“

**Sleep :**

dieser Befehl versetzt das Cockpit in den Sleep-Modus.

Sendestring-PC	0x30	*adr	
Länge	2 Bytes		
Empfangsstring-PC	-		
Länge	0 Bytes		

\*adr Cockpit-Adresse

Vor dem Wechsel in den Sleep-Modus wird nichts gesendet.

Stand Nov. 2003: Bisher nicht benötigt u. nicht aktiviert.

**Wakeup :**

dieser Befehl weckt das Cockpit aus dem Sleep-Modus.

Sendestring-PC	0x31	*adr	
Länge	2 Bytes		
Empfangsstring-PC	0x31	*adr	*1
Länge	2 Bytes		

\*adr Cockpit-Adresse

Stand Nov. 2003: Bisher nicht benötigt u. nicht aktiviert.

**Get Temperatur :**

dieser Befehl fragt die cockpitinterne Temperatur ab.

Sendestring-PC	0x32	*adr		
Länge	2 Bytes			
Empfangsstring-PC	0x32	*adr	*1	*2
Länge	3 Bytes			

\*adr Cockpit-Adresse

\*1 high Byte Temperatur-Word

\*2 low Byte Temperatur-Word

Die Temperatur ist in °C gleich  $(\text{Temperatur-Word} - 2692)/10$ .

Auflösung:  $\pm 4^\circ\text{C}$  ohne Eichung, nach Eichung  $\pm 1^\circ\text{C}$  im Bereich 0 ... 40°C.

Stand Nov. 2003: Bisher nicht aktiviert.

**Run Daten :**

hiermit werden die aktuellen Tret-Daten aus dem Cockpit empfangen. Diese Tret-Daten haben eine gewisse Bedeutung und müssen dementsprechend vom PC aus verarbeitet und angezeigt werden. Sie werden im Zyklus von 1 Sekunde abgeholt. Falls vorher ein Programm bzw. eine Person eingestellt wurde, müssen diese beiden Informationen hinsichtlich der Tret-Daten gleich sein. Bei Ungleichheit sollte *ergo\_win* dies sofort dem Anwender mitteilen und seine Abfragen stoppen.

Sendestring-PC	0x40	*adr												
Länge	2 Bytes													
Empfangsstring-PC	0x40	*adr	*1	*2	*3	*4	*5	*6	*7	*8	*9	*0	*a	*b
Länge	19 Byte													
			*c	*d	*e	*f	*g							

\*adr Cockpit-Adresse

\*1 entspricht dem eingestellten Programm. Der Wert kann natürlich nur dem eines freigeschalteten Programmes entsprechen und darf nicht größer als 79 sein.

\*2 entspricht der eingestellten Person. 0=Gast 1-4=Person 1-4.

\*3 0x00 entspricht dem Zustand „Nicht\_Treten“

0x01 entspricht dem Zustand „Treten“

Stand Ende Nov. 2003: Ist bisher mit Offset 128.

\*4 Aktuelle Wattleistung geteilt durch 5 W. Gültige Werte können hierbei von 5..80 (= 25..400 Watt) sein, oder 10..160 (= 50..800 Watt). Dieser Bereich hängt vom Cockpittyp ab. Werte außerhalb dieses Wattbereiches sind ungültig.

\*5 Momentane RPM-Zahl. Dieser Wert kann sich von 0..199 Umdrehungen / Minute erstrecken.

\*6 Ist die Vorkommastelle der aktuellen Geschwindigkeit. Der Wert kann nicht größer als 99 werden.

\*7+\*8 Entspricht der bereits gefahrenen Distanz in 100 m (d. h. km mit einer Nachkomma-Stelle).

\*9+\*0 Entspricht dem vorhandenem Tret-Zeitraum in s.

\*a+\*b Entspricht den bisher verbrauchten 100 Joule (d. h. KJoule mit einer Nachkommastelle).

\*c Derzeitiger gemessener Puls, kann sich von 0..199 erstrecken.

\*d Aktueller Pulszustand. 0=kein Puls vorhanden oder nicht verarbeitet.

1=Puls zu gering

2=Puls ok

3=Puls zu hoch

4=aufgrund vom Pulszustand wird geblinkt

5=Piepsen kommt und Wattleistung wird reduziert

Größere Werte dürfen nicht vorkommen.

Stand Ende Nov. 2003: Ist bisher unzuverlässig (meistens 0), kann aber aus den anderen Daten im PC berechnet werden.

\*e Momentaner Gang. Diese Variable wird auch nur von Distanzgesteuerten Programmen verwertet. Es gibt nur die Gänge 1..28.

\*f+\*g Entspricht den bisher verbrauchten realen 100 Joule (d. h. Kjoule mit einer Nachkommastelle). Bei Person 0 ist dieser Wert immer 0,0.

**Relax Daten :**

es werden hier nur die aktuellen Relax-Informationen vom Cockpit im Takte von 1 Sekunde abgeholt. Dies geht solange bis das Cockpit resetet oder gestoppt wurde.

Sendestring-PC	0x42	*adr			
Länge	2 Byte				
Empfangsstring-PC	0x42	*adr	*1	*2	*3
Länge	5 Bytes				

- \*adr Cockpit-Adresse  
 \*1 aktueller Zustand des Relaxmodes  
 0x00 es wird auf einstecken des Relaxsensors gewartet  
 0x01-0x03 bedeutet der Relax wird eingemessen (199)  
 0x04 bedeutet es kommt ein aktueller gültiger Wert.  
 \*2 dies ist der Relax-Wert 0..255 !.  
 \*3 aktueller Pulswert

**Set Watt :**

stellt die gewünschte Watt-Leistung ein.

Sendestring-PC	0x51	*adr	*1
Länge	3 Bytes		
Empfangsstring-PC	0x51	*adr	*1
Länge	3 Bytes		

- \*adr Cockpit-Adresse  
 \*1 ist der gewünschte Watt-Wert geteilt durch 5 W, also Watt-Wert = 5 bei 25 W, 80 bei 400 W.

**Set Puls :**

stellt den Limit bzw. Ziel-Puls ein ( im Cardio-Programm den Cardio-Puls).

Sendestring-PC	0x52	*adr	*1
Länge	3 Bytes		
Empfangsstring-PC	0x52	*adr	*1
Länge	3 Bytes		

- \*adr Cockpit-Adresse  
 \*1 ist der gewünschte Puls-Wert von 80..199.

**Set Gang :**

hiermit wird ein möglicher Gang in einem Distanzprogramm eingestellt !.

Sendestring-PC	0x53	*adr	*1
Länge	3 Bytes		
Empfangsstring-PC	0x53	*adr	*1
Länge	3 Bytes		

\*adr Cockpit-Adresse

\*1 ist der gewünschte Gang-Wert, dieser geht von 1..28 .

Funktioniert nur in Distanzprogrammen (vom Cockpit), also beispielsweise nicht vom PC im Programm 0.

Stand Ende Nov. 2003: Es wird bei Distanzprogrammen (vom Cockpit) der vom PC gesetzte Gang zwar angezeigt, aber nicht zur Berechnung der Bremsleistung übernommen.

**Set Speed :**

mit diesem Befehl wird eine mögliche Geschwindigkeit eingestellt ( siehe IR3-Programm)..

Sendestring-PC	0x54	*adr	*1
Länge	3 Bytes		
Empfangsstring-PC	0x54	*adr	*1
Länge	3 Bytes		

\*adr Cockpit-Adresse

\*1 ist der gewünschte Geschwindigkeits-Wert von 5..99 !.

**Set Language :**

mit diesem Befehl wird die gewünschte Landessprache eingestellt.

Sendestring-PC	0x60	*adr	*1
Länge	3 Bytes		
Empfangsstring-PC	0x60	*adr	
Länge	2 Bytes		

\*adr Cockpit-Adresse

\*1 ist der gewünschte Sprache

0 Deutsch

1 Englisch

2 Französisch

3 Italienisch

**Get Language :**

mit diesem Befehl wird die eingestellte Landessprache abgefragt.

Sendestring-PC	0x61	*adr		
Länge	2 Bytes			
Empfangsstring-PC	0x61	*adr	*1	
Länge	3 Bytes			

\*adr Cockpit-Adresse  
 \*1 ist die eingestellte Sprache  
 0 Deutsch  
 1 Englisch  
 2 Französisch  
 3 Italienisch

**Set Timer :**

mit diesem Befehl wird die gewünschte Uhrzeit eingestellt.

Sendestring-PC	0x62	*adr	*1	*2	*3
Länge	5 Bytes				
Empfangsstring-PC	0x62	*adr			
Länge	2 Bytes				

\*adr Cockpit-Adresse  
 \*1 Sekunden, 0..59  
 \*2 Minuten, 0..59  
 \*3 Stunden, 0..23

**Get Timer :**

mit diesem Befehl wird die derzeitige Cockpit Uhrzeit abgefragt.

Sendestring-PC	0x63	*adr			
Länge	2 Bytes				
Empfangsstring-PC	0x63	*adr	*1	*2	*3
Länge	5 Bytes				

\*adr Cockpit-Adresse  
 \*1 Sekunden, 0..59  
 \*2 Minuten, 0..59  
 \*3 Stunden, 0..23

**Set Date :**

mit diesem Befehl wird das gewünschte Datum eingestellt.

Sendestring-PC	0x64	*adr	*1	*2	*3
Länge	5 Bytes				
Empfangsstring-PC	0x62	*adr			
Länge	2 Bytes				

\*adr Cockpit-Adresse  
 \*1 Tag 0..31(28..31) abhängig vom Monat und dem (Schalt-)Jahr  
 \*2 Monat 1..12  
 \*3 Jahr 1..99 entspricht 20xx

**Get Date :**

mit diesem Befehl wird das derzeitige Cockpit Datum abgefragt.

Sendestring-PC	0x65	*adr			
Länge	2 Bytes				
Empfangsstring-PC	0x65	*adr	*1	*2	*3
Länge	5 Bytes				

\*adr Cockpit-Adresse  
 \*1 Tag 0..31(28..31) abhängig vom Monat und dem (Schalt-)Jahr  
 \*2 Monat 1..12  
 \*3 Jahr 1..99 entspricht 20xx

**Get Programm :**

mit diesem Befehl wird die derzeitige Programmnummer abgefragt.

Sendestring-PC	0x66	*adr	
Länge	2 Bytes		
Empfangsstring-PC	0x67	*adr	*1
Länge	3 Bytes		

\*adr Cockpit-Adresse  
 \*1 Programmnummer 0..79

**Get Person :**

mit diesem Befehl wird die derzeitige Person abgefragt.

Sendestring-PC	0x67	*adr	
Länge	2 Bytes		
Empfangsstring-PC	0x66	*adr	*1
Länge	3 Bytes		

\*adr Cockpit-Adresse  
 \*1 Person, 0..4

**Set Watt profile :**

mit diesem Befehl wird das Niveau der Wattleistung bezüglich des eingesetzten Programm adaptiert!.

Sendestring-PC	0x68	*adr	*1
Länge	3 Bytes		
Empfangsstring-PC	0x68	*adr	*1
Länge	3 Bytes		

\*adr Cockpit-Adresse  
 \*1 0=inkrement 5 Watt  
 1=dekrement 5 Watt

**Del Person :**

mit diesem Befehl werden die internen Daten der entsprechenden Person gelöscht.

**ACHTUNG:**

Die Person(nummer) darf nur 1, 2, 3 oder 4 sein. Wird intern nicht gecheckt ob 0 ist !!!

Sendestring-PC	0x70	*adr	*1
Länge	3 Bytes		
Empfangsstring-PC	0x70	*adr	*1
Länge	3 Bytes		

\*adr Cockpit-Adresse  
 \*1 Person, 1..4

**Del IProg :**

mit diesem Befehl werden die internen Daten aller Individualprogramme der entsprechenden Person gelöscht.

**ACHTUNG:**

Die Person(nummer) darf nur 1, 2, 3 oder 4 sein. Wird intern nicht gecheckt ob 0 ist !!!

Sendestring-PC	0x71	*adr	*1
Länge	3 Bytes		
Empfangsstring-PC	0x71	*adr	*1
Länge	3 Bytes		

\*adr Cockpit-Adresse  
 \*1 Person, 1..4

**Get Version :**

mit diesem Befehl werden die Versionsdaten weitergegeben.

Sendestring-PC	0x73	*adr		
Länge	2 Bytes			
Empfangsstring-PC	0x73	*adr	*1-8	*9
Länge	11 Bytes			

\*adr Cockpit-Adresse  
 \*1-8 Seriennummer  
 \*9 eingestellter Cockpit Typ  
 10=Cardio  
 20=Fitness  
 30=Vita De Luxe  
 40=8008  
 50=8080  
 60=Therapie

**Get Person Data :**

mit diesem Befehl werden die Daten der derzeitigen Person abgefragt.

Sendestring-PC	0x74	*adr	*blk	
Länge	3 Bytes			
Empfangsstring-PC	0x74	*adr	*blk	*1-16
Länge	19 Bytes			

\*adr Cockpit-Adresse  
 \*blk 16 Byte Datenblock des Gesamt Datenumfanges einer Person ( siehe Anhang )  
 \*1-16 Daten

**Get Person Data1 :**

mit diesem Befehl werden die Daten1 der derzeitigen Person abgefragt.

Sendestring-PC	0x75	*adr	*blk	
Länge	3 Bytes			
Empfangsstring-PC	0x75	*adr	*blk	*1-16
Länge	19 Bytes			

\*adr Cockpit-Adresse  
 \*blk 16 Byte Datenblock des Gesamt Datenumfanges einer Person ( siehe Anhang )  
 \*1-16 Daten

**Ctrl Ton**

mit diesem Befehl wird der Lautsprecher mit einer bestimmten Frequenz gespeist!

Sendestring-PC	0xD3	*1-2		
Länge	3 Byte			
Empfangsstring-PC	0xD3			
Länge	2 Byte			

\*1-2 Frequenz  
 \*adr neue Cockpit-Adresse

## Beispiel: Starten eines Programmes

Es soll das Programm 4 mit der Person 2 gestartet werden. Weiterhin soll ein Alter von 33 und ein Limit-Puls von 150 vorhanden sein. Dabei kein Limit-Zeitraum, -Distanz und -Kcal eingestellt werden. Außerdem soll kein CD-Player spielen. Die Cockpitadresse ist 1.

Stand Ende Nov. 2003: Die Cockpit-Adr. ist egal und bisher 0.

Sendstring-PC	Sendstring-Cockpit
0x23,0x01,0x04	
	0x23,0x01,0x04,0x00
0x24,0x01,0x02,0x21,0x94,0x00, 0x00,0x00,0x00,0x00,0x00,0x00, 0x00	
	0x24,0x01,0x02,0x21,0x96,0x00, 0x00,0x00,0x00,0x00,0x00,0x00, 0x00
0x21,0x01	
	0x21,0x01,0x00
0x40,0x01	
	0x40,0x01,0x04,0x02,0x01,0x05, 0x41,0x17,0x22,0x00,0x7a,0x00, 0x03,0x00,0x45,0x01,0x01
..	..
..	..
0x23,0x01	
	0x23,0x01,0x01
0x23,0x01	
	0x23,0x01,0x00

**ANHANG**

Vormerkungen:

sizeof(char) = 8, sizeof(int) = 16, sizeof(long int) = 32, sizeof(float) = 32, little endian, ANSI-C, IEEE 754.

**Personendaten**

```

struct person
{
    unsigned char        status;
    unsigned char        programm;
    unsigned char        therapie;
    unsigned int         step_programm;
    unsigned char        cardio_puls;
    unsigned char        persoenl_drehzahl;
    unsigned char        limit[12];
    unsigned char        award;
    struct pfit_data     pfitness;
    unsigned char        hfmaxi;
    unsigned char        lmax;
    struct coaching      coach;
    unsigned char        lmaxtemp;
};

struct pfit_data
{
    float                last_av_speed;
    float                last_max_speed;
    unsigned int         last_zeit;
    float                last_km;
    unsigned char        last_rpm;
    unsigned char        last_av_watt;
    unsigned char        last_av_puls;
    unsigned char        last_max_puls;
    float                last_phys_kjoule;
    float                last_real_kjoule;
    unsigned int         bmi;
    unsigned char        fit_note;
    unsigned int         gesamt_zeit;
    float                gesamt_km;
    float                gesamt_phys_kjoule;
    float                gesamt_real_kjoule;
};

/* Benutzerspezifische Erweiterung */
struct coaching
{
    unsigned int  start_kw; /* Erste echte KW im coaching */
    unsigned int  real_KW; /* 1-52, letzte echte Trainingswoche (nicht ganz nach
DIN 1355) */
    unsigned char TA; /* 0-3, letzte Trainingsart */
    unsigned char TE; /* 1-7, letzte Trainingseinheit */
    unsigned char TE_perc; /* 0-100, Prozent der letzten T-Einheit */
    unsigned char endlos; /* TRUE, FALSE */
    unsigned long rand_db; /* 3-19, 29-38 Bitfeld für Zufallsdatenbank */
    unsigned int  week_db[5]; /* 32 Wochen 0=Woche ausgelassen */
    unsigned char prog_nr; /* aktuell gefahrene Programmnummer */
    unsigned char prog_time; /* aktuell gefahrene Zeit */
    signed char  prog_korr;
};

```

```
struct person1
{
    struct gew_pers    gewicht_data;
    struct fett_pers   fett_data;
};

struct gew_pers
{
    unsigned char      delta_week[60];
    unsigned int       start_week;
    unsigned char      delta_year[60];
    unsigned int       start_year;
    unsigned char      filler[4];
};

struct fett_pers
{
    unsigned char      delta_week[60];
    unsigned int       start_week;
    unsigned char      delta_year[60];
    unsigned int       start_year;
    unsigned char      filler[4];
};
```

#### Programm Daten

```
#define FREIES_TRAINING    0
#define FP_WATT1_PROG      19
#define FP_WATT2_PROG      28
#define FP_PULS_PROG       38
#define FP_KONA            39
#define FP_ROTH           40
#define FP_LANZA           41
#define FP_COOL1          42
#define FP_COOL3          44
#define FP_CONCONI1       45
#define FP_CONCONI2       46
#define FP_CARDIO         47
#define FP_RPM_KONST      48
#define FP_STEIGUNG       49
#define FP_DREHZAHL15     50
#define FP_DREHZAHL19     51
#define FP_TDF_FIRST      52
#define FP_TDF_MAX        72
#define FP_THERAPIE_LT    73
#define FP_IND_WATT       77
#define FP_IND_PULS      78
#define FP_IND_SPEED     79
```