

### Telegramm-Aufbau

Die gesamte Kommunikation zwischen Teilnehmern erfolgt über Telegramme, die einen recht einfachen Aufbau haben. Ein solches Telegramm besteht aus mindestens 9 und maximal 23 Bytes. Die meisten Aktionen auf dem Bus, wie z.B. Schalten, Dimmen, etc., lassen sich mit dem kleinsten Telegramm realisieren. Der prinzipielle Aufbau sieht wie folgt aus:

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8...23	Byte 9..24
Kontrollbyte	Quelladresse		Zieladresse		DRL	Nutzdaten		Checksum

### Kontrollbyte

Das Kontrollbyte beinhaltet das Wiederholungsbit und die Priorität des Telegramms:

1	0	R	1	p1	p0	0	0
7	6	5	4	3	2	1	0

Es gibt 4 verschiedene Prioritäts-Levels.  
Systemfunktionen haben die höchste Priorität.

p1	p0	Bezeichnung
0	0	Systemfunktion
0	1	Alarmfunktion
1	0	hohe Priorität
1	1	normale Priorität

Wird ein Telegramm das erste mal gesendet, so ist das Wiederholungsbit = 1. Wird ein Telegramm wiederholt, z.B. weil ein Empfänger es beim ersten Mal nicht verarbeiten konnte, ist dieses Bit = 0. Somit können die Teilnehmer, die das Telegramm bereits beim ersten Mal korrekt empfangen haben, es bei der Wiederholung ignorieren.

### Quelladresse

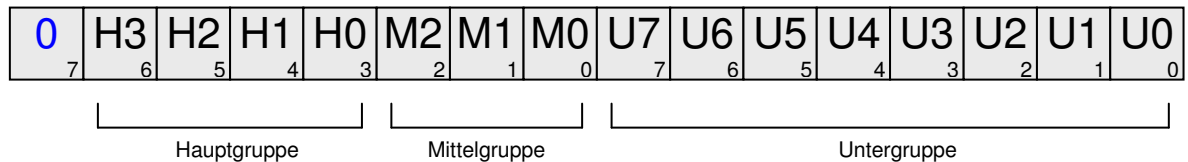
Die Quelladresse besteht aus zwei Byte, wobei erst das MSB dann das LSB übertragen wird.

B3	B2	B1	B0	L3	L2	L1	L0	T7	T6	T5	T4	T3	T2	T1	T0
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Bereich				Linie				Teilnehmer							

Die Quelladresse ist immer eine physikalische Adresse eines Gerätes. Die Schreibweise ist dezimal <Bereich>.<Linie>.<Teilnehmer>, also z.B. 1.1.34

### Zieladresse

Die Zieladresse kann entweder eine physikalische Adresse oder eine Gruppenadresse sein. Das wird im DRL-Byte festgelegt (s.u.) Im Falle einer physikalischen Adresse ist die Bit-Aufteilung so wie unter Quelladresse beschrieben. Handelt es sich um eine Gruppenadresse sieht die Aufteilung in der Regel wie folgt aus:



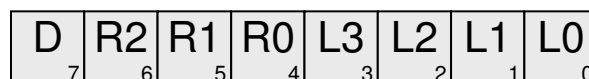
Dies ist die Einteilung in 3 Ebenen, die Schreibweise der Gruppenadresse ist z.B. 1/3/43

Es gibt auch die (selten genutzte) Möglichkeit die Gruppenadresse in nur 2 Ebenen aufzuteilen. Dann gibt es nur die 4 Bit breite Hauptgruppe und die aus 11 Bit bestehende Untergruppe.

Das erste Bit ist immer 0 für die Standard-Adressierung. Es gibt auch Sonderadressierungen, die interessieren uns aber erstmal überhaupt nicht.

### DRL-Byte

DRL kommt von Destination-adress-flag, Routing-counter, Length und das sind bereits die drei Bestandteile dieses Steuerbytes:



Länge der auf das DRL-Byte folgenden Nutzdaten.  
Der Wertebereich geht von 2 bis 17, d.h. bei  
Länge=0 folgen 2 Bytes, bei Länge=15 folgen 17.

Der Routing-Zähler gibt an wie oft Telegramme über Linien- und Bereichskoppler weitergereicht werden. Der Standardwert ist 6 und wird bei jedem Weiterreichen um eins reduziert. Bei 0 wird das Telegramm gar nicht weitergeleitet, bei 7 hingegen beliebig oft.

Das Destination-Adress-Flag gibt an ob die Zieladresse eine physikalische Adresse (0) oder eine Gruppenadresse (1) ist.

### Nutzdaten

Die Nutzdaten bestehen aus mindestens zwei, maximal 17 Bytes. Dabei haben die ersten beiden Bytes eine besondere Bedeutung, denn zum Einen ist in ihnen der auszuführende Befehl kodiert, zum Anderen kann man viele Aufgaben schon mit diesen 2-Byte erledigen.

Es gibt, grob gesagt, zwei mögliche Varianten für die Nutzdaten:

- Der so genannte EIS (**E**IB **I**nterworking **S**tandard) ist ein Standard zur Kommunikation von Geräten unterschiedlicher Hersteller. Der Befehlssatz ist ziemlich mächtig und erlaubt so ziemlich alles an Informationen auf Gruppenadress-Ebene zu übertragen. Es gibt 15 verschiedene EIS Formate für die folgenden Funktionen:

- EIS 1 Schalten
  - EIS 2 Dimmen
  - EIS 3 Uhrzeit
  - EIS 4 Datum
  - EIS 5 Wert, Zahl mit Nachkommastellen
  - EIS 6 Relativwert, 0-100%
  - EIS 7 Antriebssteuerung
  - EIS 8 Zwangssteuerung
  - EIS 9 Zahl mit Nachkommastellen nach IEEE
  - EIS 10 16-Bit Wert
  - EIS 11 32-Bit Wert
  - EIS 12 Zugangskontrolle
  - EIS 13 ASCII Zeichen
  - EIS 14 8-Bit Wert
  - EIS 15 Zeichenkette
- Die zweite mögliche Nutzung sind Befehle zur Kommunikation mit nur einem Teilnehmer. Dies sind alle möglichen Befehle zur Programmierung, Parametrierung, zum Lesen und Schreiben des EEPROM eines Teilnehmers, etc.

Wie genau die Inhalte der Telegramme zu interpretieren sind, gibt's in den nächsten Kapiteln. Hier soll zunächst nur mal generell der Aufbau eines Telegramms erläutert werden.

### **Prüfbyte**

Im Anschluss an die Nutzdaten wird immer das Prüfbyte gesendet. Es handelt sich dabei um die invertierte, bitweise EXOR-Verknüpfung aus allen vorher gesendeten Bytes des Telegramms.

Auf der Empfängerseite kann man sich die Prüfung ziemlich einfach machen. Es reicht dabei, alle empfangenen Bytes inklusive des Prüfbytes ohne Übertrag zu addieren. Nur wenn am Ende 0xFF herauskommt ist das empfangene Telegramm OK.

### **Beispiel**

Das folgende Beispiel ist ein EIS 1 Telegramm vom Teilnehmer mit der physikalischen Adresse 1.1.1, der z.B. eine Lampe an einem Aktor-Ausgang mit der Gruppenadresse 1/2/5 einschaltet:

<b>BC</b>	<b>1011 1100</b>	Kontrollbyte
<b>11</b>	<b>0001 0001</b>	Quelladresse 1.1.1
<b>01</b>	<b>0000 0001</b>	
<b>0A</b>	<b>0000 1010</b>	Zieladresse 1/2/5
<b>05</b>	<b>0000 0101</b>	
<b>00</b>	<b>0000 0000</b>	EIS 1
<b>81</b>	<b>1000 0001</b>	einschalten (zum ausschalten stünde hier 0x80)
<b>DD</b>	<b>1101 1101</b>	Prüfsumme