

# TPM2 Protokoll Spezifikationen

## V1.0 - 2013

### 1. Verwendung

Das TPM2-Protokoll wurde konzipiert um Daten (von Lichtinstallationen jeglicher Art) zwischen zwei Geräten auszutauschen. Grundlegender Gedanke war dabei ein möglichst einfaches Protokoll zu entwerfen, welches in seiner Struktur dennoch hinreichend robust ist und viele Möglichkeiten zur Erweiterung bietet.

### 2. Übertragungsarten

TPM2-Telegramme können sowohl seriell (via UART) als auch paketbasiert (via Ethernet) übertragen werden. Diese beiden Übertragungsformen sollen im Folgenden mit den Zusätzen .Serial sowie .Net gekennzeichnet werden.

Bei serieller Datenübertragung werden keine Anforderungen an Baudrate, Parität oder Anzahl der Stop-Bits gestellt. Im Falle der Übertragung via Ethernet wird das TPM2.Net Telegramm in Form eines UDP-Datenpaketes übertragen. Der Port für die UDP-Kommunikation ist [65506](#).

### 3. Paketaufbau

Der Aufbau von TPM2-Paketen unterscheidet sich je nachdem welche Form der Datenübertragung zu Grunde liegt und ist nachfolgend schematisch dargestellt.

Position	TPM2.Serial	TPM2.Net
0	Block Start Byte	Block Start Byte
1	Block Type	Block Type
2	Frame Size (High Byte)	Frame Size (High Byte)
3	Frame Size (Low Byte)	Frame Size (Low Byte)
4	Data	<b>Packet Number</b>
5	...	<b>Total Packet Count</b>
	...	Data
4+Frame Size	Data	...
4+Frame Size +1	Block End Byte	...
4+Frame Size +2		Data
4+Frame Size +2		Block End Byte

## Block Start Byte

Das erste Byte eines TPM2-Paketes ist immer das sog. „Block Start Byte“. Es signalisiert dem Empfänger den Beginn eines neuen Paketes.

Block Start Byte = 0xC9 für TPM2.Serial  
Block Start Byte = 0x9C für TPM2.Net

## Block Typ

Das zweite Byte eines TPM2 Paketes definiert den Typ des Paketes. Folgende Paket-Typen sind definiert:

Daten Paket	Block Typ	=	0xDA
Kommando Paket	Block Typ	=	0xC0
Bestätigung	Block Typ	=	0xAC
Antwort mit Daten	Block Typ	=	0xAD

Die Bedeutung der einzelnen Paket-Typen sowie die Form der Daten für die jeweiligen Paket-Typen sollen später im Detail erläutert werden.

## Frame Size

Das dritte und das vierte Byte eines TPM2-Paketes definieren gemeinsam die Länge der folgenden Nutzdaten. Es wird zuerst das High-Byte und dann das Low-Byte übertragen. Die Angabe der Nutzdatengröße dient in erster Linie der Synchronisierung der Datenübertragung.

## Paket Nummer / Total Packet Count

Die Paketnummer ist nur in [TPM2.Net](#) Paketen vorhanden und dient der gesplitteten Datenübertragung um maximale Paketgrößen von UDP / Ethernet bei großen Datenmengen nicht zu überschreiten. Mit der Paketnummer und der Gesamtanzahl an Paketen (Total Packet Count) kann ein Empfänger gesplittete Daten unabhängig von der Reihenfolge ihres Eintreffens folgerichtig rekonstruieren. Paketnummer und Total Packet Count **MÜSSEN** immer Teil eines TPM2.Net Paketes sein, auch wenn keine gesplittete Datenübertragung stattfindet. In letzterem Fall kann die Paketnummer z.B. durch 0x00 festgesetzt werden.

## Daten

Es können maximal [65536 Bytes](#) an Nutzdaten übertragen werden. Die Anzahl übertragender Bytes muss dem unter „Frame Size“ angegebenen Wert entsprechen. Die genaue Form der Daten ist vom „Paket-Typ“ des Datenpakets abhängig und soll später im Detail erläutert werden.

## Block Ende Byte

Das letzte Byte eines TPM2-Paketes ist immer **0x36** und dient ausschließlich der Synchronisierung der Datenübertragung.

## 4. Paket-Typen und Nutzdatenformate

### Daten-Paket (0xDA)

Für ein Datenpaket sind keine speziellen Anforderungen an das Format der Nutzdaten spezifiziert. Die Nutzdaten dienen in diesem Fall der reinen Datenübertragung.

### Kommando-Paket (0xC0)

Das erste Byte der Nutzdaten eines Kommandopakets enthält immer das sog. „**Kommando-Kontroll-Byte**“. Dieses enthält in seinen einzelnen Bits folgende Informationen über den Befehl:

MSB	Richtung des Befehls (0=Lesebefehl, 1=Schreibbefehl)
MSB-1	Befehl erwartet Antwort? (0=Nein, 1=Ja)
MSB-2	Reserviert
MSB-3	Reserviert
MSB-4	Reserviert
MSB-5	Reserviert
MSB-6	Reserviert
LSB	Reserviert

Das zweite Byte der Nutzdaten eines Kommandopakets enthält den eigentlichen Befehl = „**Befehlstyp**“.

Folgende Befehle sind in der aktuellen Version der TPM2-Protokoll-Spezifikation (**Stand 03/2013**) festgelegt und werden in künftigen Versionen erweitert / ergänzt.

0x00	Konfigurationsdaten = Parameterübermittlung
0x01	Befehl zum Speichern der aktuellen Konfigurationsdaten
0x02	Befehl zum Laden einer best. Konfiguration aus dem Speicher
0x03	Initialisierung des Schreibens auf SD-Karte
0x04	Befehl zum Starten des Abspielens von SD-Karte
0x05	Befehl zum Stoppen des Abspielens von SD-Karte
0x06	Einen Frame von/auf SD-Karte lesen/schreiben
0x0A	Setzen / Lesen der Master-Helligkeit
0x0B	Setzen / Lesen der Einstellungen der globalen Gamma-Korrektur
0x0C	Setzen / Lesen der Master-Geschwindigkeit
0x0D	Setzen / Lesen der aktuellen Programm-Nummer

0x0E	Setzen / Lesen der Start-Farbe
0x0F	Setzen / Lesen des globalen Time-out-Wertes
0x10	Setzen / Lesen der Anzahl angeschlossener Pixel
0x11	Setzen / Lesen der Anzahl an Wiederholungen
0x12	Setzen / Lesen der Start-Adresse
0x20	„Ping“-Befehl -> erwartet Antwort-Paket vom Type „0xAC“ (s.o.)
0xFF	Benutzerdefinierter Befehl (s.u.)

Nach dem Befehlstyp folgen eventuell vorhandene Parameter des jeweiligen Befehls.

Der Befehl **0xFF** steht als Hinweis für einen folgenden frei belegbaren, benutzerdefinierten Befehl der keiner Standardisierung unterliegt. Im Anschluss an **0xFF** sollte der genaue Nutzer-Befehl mit evtl. anschließenden Parametern übermittelt werden.

Wenn im „**Kommando-Kontroll-Byte**“ des Befehls das Bit (MSB-1) gesetzt ist so muss der Empfänger mit einem Paket vom Typ „0xAC“ bzw. „0xAD“ (s.o.) antworten. Welcher der beiden Antwort-Typen Einsatz findet richtet sich danach ob der Befehl Daten als Antwort erwartet oder nicht (bestimmt durch das MSB des „**Kommando-Kontroll-Byte**“).

### **Bestätigung und Antwort mit Daten (0xAC, 0xAD)**

Auf Befehle die eine Antwort erwarten, sowohl mit Daten als auch ohne Daten, wird mit einem Paket von Typ 0xAC bzw. 0xAD geantwortet (s.o.).

Das erste Byte der Nutzdaten eines solchen Paketes enthält immer das sog. „**Antwort-Byte**“ welches folgende Werte annehmen kann:

0x00	Befehl wurde erkannt und Parameter empfangen
0x01	Befehl wurde erkannt, Daten/Parameter waren fehlerhaft
0x02	Befehl unbekannt
0x03	Protokoll-Fehler
0x04	Daten zu groß

Nach dem „Antwort-Byte“ folgen (falls angefordert) Parameter / Nutzdaten.

## 5. Schlussbemerkungen

Das TPM2-Protokoll wurde von Nutzern des Internetforums [www.ledstyles.de](http://www.ledstyles.de) entworfen. Es kann frei benutzt / modifiziert werden. Geräte und / oder Software die das TPM2-Protokoll **offiziell unterstützen** sind jedoch angehalten sich an die Vorgaben der vorliegenden Spezifikationen zu halten.

---