



WIR RETTEN, WAS ANDERE
AUFGEBEN.

www.dmxface.at

DMXface

Object Communication manual

Für DMXface ab Firmware Version 5.60

| | |
|---|-----------|
| Änderungen zu Vorversionen und zur DMXface Basisversion | 2 |
| Allgemeines | 2 |
| Struktur der Befehle und Antworten | 2 |
| Antworten der Befehle Get 0x28 und Set 0x29 | 4 |
| Objekt Nummern für den Zugriff im DMXface | 5 |
| Datentypen | 7 |
| BEFEHL GET OBJECT DATA | 8 |
| Abfrage von Szenen, Szenen Namen und der Szenen Programmierung | 8 |
| Abfrage von Programm Namen oder der Programminhalte | 9 |
| Abfrage von DMX OUT Namen oder Kanal Werten | 10 |
| Abfrage von DMX IN Kanal Werten | 11 |
| Abfrage von TIMELINE Namen, Laufdaten und Programmierung | 12 |
| Abfrage von Sequenz Namen oder der Programmierung | 13 |
| Abfrage von Trigger Namen, Programmierung und Zeitschalt Settings | 14 |
| Abfrage von Outport Namen oder Zustand | 15 |
| Abfrage von Inport und Busport Namen oder Zustand analog oder digital | 16 |
| Abfrage der Timer Zustände | 17 |
| Abfrage von Buffern | 18 |
| Abfrage von LCD Page Namen oder der Programmierung | 20 |
| Abfrage von TEXT Pages (Legacy) | 21 |
| Die Zuordnung der Bezeichnungs-Texte zu den TEXT Pages bis Ver. 5.40 | 21 |
| BEFEHL SET OBJECT DATA | 24 |

Änderungen zu Vorversionen und zur DMXface Basisversion

Initiale Implementation mit DMXface V.5.60 und DMXface Console 5.60

Allgemeines

Die Object Communication ist ein eigener Teil des Kommunikationsprotokolls MAIN COMMUNICATION welches über die Befehle

#40 / 0x28 GET_OBJECT

und

#41 / 0x29 SET_OBJECT

abgewickelt wird.

Struktur der Befehle und Antworten

Ein Datenpaket hat im ersten Byte immer den Befehl welcher einen Wert zwischen 0x00 und 0x7F hat. Diese Befehle sind grundsätzlich im alten Kommunikationsprotokoll beschrieben und haben meist kein einheitliches Format der Sendung oder Antwort.

Aus Kompatibilitätsgründen werden die Befehle aber beibehalten oder nur herausgenommen, wenn diese absolut nicht mehr erforderlich sind.

Diese Beschreibung befasst sich ausschließlich mit den Kommando Bytes

Commandbyte (0x28) GET BEFEHL Objekte abfragen

Commandbyte (0x29) SET BEFEHL Objekte beschreiben / Aufrufen

Jedes Kommando wird mit einem 2ten Options bzw. Datentypenbyte unterstützt welches sich auf Unterfunktionen, oder unterschiedliche Datenformate bezieht.

Die Options-, bzw. Datentypenbytes sind in den [Datentypen](#) aufgelistet.

Jede Einheit in einem DMXface (DMX OUT Kanal, Szene, Timer, Eingang, Ausgang, ...) hat eine eindeutige Objekt Nummer, ebenso einige wichtige Funktionen wie Timeline-Stopp, Timeline-Start....

Diese Objekt Nummern sind in den [Objektnummern](#) aufgelistet

So ergeben sich einheitliche Kommando und Antwort Strukturen

Z.B. Abruf der Szene 5 (Objekt Nummer 0x004) wie folgt

| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
|-----------------------------|--------------------------------------|--------------------------|-------------------------|
| Kommando Byte SET Befehl | Options Byte Datentype Default | Objektnummer HighByte | Objektnummer LowByte |
| 0x29 | 0x00 | 0x00 | 0x04 |

Zu sendende Daten inkl. Header (LAN / USB)

| | | | | |
|----------|-----------|------------|------------------------|----------|
| STX Byte | Länge Low | Länge High | Daten 4 Bytes | ETX Byte |
| 0x01 | 0x04 | 0x00 | 0x29, 0x00, 0x00, 0x04 | 0x03 |

Z.B. Aufruf von Programm 25 (Objekt Nummer 0x1018)

| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
|----------------------|--------------------------------------|--------------------------|-------------------------|
| Kommando SET Byte | Options Byte Datentype Default | Objektnummer HighByte | Objektnummer LowByte |
| 0x29 | 0x00 | 0x10 | 0x18 |

Z.B. Auslesen des Namens der Szene 10 (Objekt Nummer 0x009)

| Byte 1 | Byte 2 | Byte 3 | Byte 4 |
|-----------------|---|--------------------------|-------------------------|
| Kommando GET | Options Byte Datentype Dtype_NAME | Objektnummer HighByte | Objektnummer LowByte |
| 0x28 | 0xA1 | 0x00 | 0x09 |

Sollen beim Auslesen von Daten gleich mehrere Objekte hintereinander gelesen werden, so kann zusätzlich eine 16 Bit Information über die Anzahl der Objekte angefügt werden. Dabei ist die maximale Anzahl der gleichzeitig lesbaren Objekte nur durch die Sende Empfangspuffer Größe limitiert (ca. 1KB)

Z.B. Auslesen der Szenen Namen von Szene 1-32 (Texte immer fix 32 Byte pro Text / Szene, Startobjekt Szene 1 = 0x0000)

| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 |
|----------------------|---|--------------------------|-------------------------|-------------------|------------------|
| Kommando GET Byte | Options Byte Datentype Dtype_NAME | Objektnummer HighByte | Objektnummer LowByte | Länge Highbyte | Länge Lowbyte |
| 0x28 | 0xA1 | 0x00 | 0x00 | 0x00 | 0x20 |

Antworten der Befehle Get 0x28 und Set 0x29

Bei einer Antwort wird das Command Byte um 0x80 erhöht und an **erster** Stelle zurückgesendet.

An zweiter Stelle kommt die Datentype (Dtype) der Antwort.

Position 3 und 4 sind die Objektnummer (16 Bit)

Danach folgen falls erforderlich die entsprechenden Daten.

Z.B. Befehl auslesen der DMX OUT Kanäle 1-512 (Startobjekt DMX1 = 0x2000)

| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | Byte 6 |
|----------------------|---|--------------------------|-------------------------|-------------------|------------------|
| Kommando GET Byte | Options Byte Datentype Dtype_Byte | Objektnummer HighByte | Objektnummer LowByte | Länge Highbyte | Länge Lowbyte |
| 0x28 | 0x01 | 0x20 | 0x00 | 0x02 | 0x00 |

Zu sendende Daten inkl. Header (LAN/ USB)

| | | | | |
|----------|-----------|------------|-------------------------------|----------|
| STX Byte | Länge Low | Länge High | Daten 6 Bytes | ETX Byte |
| 0x01 | 0x06 | 0x00 | 0x28 0x01 0x20 0x00 0x02 0x00 | 0x03 |

Die Antwort

| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | | Byte 516 |
|-------------------------------------|---|--------------------------|-------------------------|---------------|-----|-----------------|
| Return Kommando GET Byte (+0x80) | Options Byte Datentype Dtype_Byte | Objektnummer HighByte | Objektnummer LowByte | DMX 1 Byte | ... | DMX 512 Byte |
| 0xA8 | 0x01 | 0x20 | 0x00 | Wert DMX1 | | Wert DMX512 |

Empfangene Daten inkl. Header (LAN/ USB), gesamt 520 Bytes, Datenbereich 516 Bytes

| | | | | | |
|----------|-----------|------------|------------------------|---------------|------|
| STX Byte | Länge Low | Länge High | CMD Response 4 Bytes | DMX 512 Bytes | ETX |
| 0x01 | 0x04 | 0x02 | 0xA8, 0x01, 0x20, 0x00 | Werte | 0x03 |

Objekt Nummern für den Zugriff im DMXface

Jede Einheit des Controllers ist über eine 16 Bit Objektnummer erreichbar.

Statusabfragen und Steuerbefehle für die Umschaltung von Modi bleiben weiterhin lt. dem Grundprotokoll bestehen.

Objektnummern Bereiche (Startnummer), zum Einfügen in C# .NET

//SZENEN

```
public const UInt16 OBJNR_SCENE = 0x0000; //Szene1 = 0x0000 bis Szene 198 = 0x00C5
public const UInt16 OBJNR_ALLOFF = 0x0FFE;
public const UInt16 OBJNR_DONOTHING = 0x0FFF;
```

//PROGRAMME

```
public const UInt16 OBJNR_PROGRAM = 0x1000;
```

//DMXOUT

```
public const UInt16 OBJNR_DMXOUT = 0x2000;
public const UInt16 OBJNR_DMXOUT1 = 0x2000;
public const UInt16 OBJNR_DMXOUT2 = 0x2400;
public const UInt16 OBJNR_DMXOUT3 = 0x2800;
public const UInt16 OBJNR_DMXOUT4 = 0x2C00;
```

//DMXIN

```
public const UInt16 OBJNR_DMXIN = 0x3000;
public const UInt16 OBJNR_DMXIN1 = 0x3000;
public const UInt16 OBJNR_DMXIN2 = 0x3400;
public const UInt16 OBJNR_DMXIN3 = 0x3800;
public const UInt16 OBJNR_DMXIN4 = 0x3C00;
```

//TIMELINE

```
public const UInt16 OBJNR_TIMELINE = 0x4000;
public const UInt16 OBJNR_TIMELINE_STOP = 0x4FFE;
public const UInt16 OBJNR_TIMELINE_RESET = 0x4FFF;
```

//SEQUENZEN

```
public const UInt16 OBJNR_SEQUENCE = 0x5000;
```

//TRIGGER

```
public const UInt16 OBJNR_TRIGGER = 0x6000;
```

//OUTPUTS

```
public const UInt16 OBJNR_OUTPORT = 0x7000;
```

//GRUPPE PORTS INPORT und BUSPORT

```
public const UInt16 OBJNR_PORTS = 0x8000;
public const UInt16 OBJNR_INPORT = 0x8000;
public const UInt16 OBJNR_INPORT_AD = 0x8200;
public const UInt16 OBJNR_BUSPORT = 0x8800;
public const UInt16 OBJNR_BUSPORT_AD = 0x8A00;
```

```
//TIMER
```

```
public const UInt16 OBJNR_TIMER = 0x9000;
```

```
//GRUPPE BUFFERS CHARBUFFERS und ANDERE Zugänge
```

```
public const UInt16 OBJNR_BUFFERS = 0xA000;  
public const UInt16 OBJNR_CHARBUFFER = 0xA000;  
public const UInt16 OBJNR_RTCTIME = 0xAE00;  
public const UInt16 OBJNR_DMXMASTER = 0xAE10;  
public const UInt16 OBJNR_COM1BUFFER = 0xAF00;  
public const UInt16 OBJNR_COM2BUFFER = 0xAF10;  
public const UInt16 OBJNR_USBBUFFER = 0xAF60;  
public const UInt16 OBJNR_RS485BUFFER = 0xAF70;  
public const UInt16 OBJNR_LANBUFFER = 0xAF80;  
public const UInt16 OBJNR_IRBUFFER = 0xAFF0;
```

```
//LCD PAGES und Text Page Zugriff
```

```
public const UInt16 OBJNR_LCDPAGE = 0xB000;  
public const UInt16 OBJNR_TEXTPAGE= 0xB800;
```

```
//DATA TABLES
```

```
public const UInt16 OBJNR_DATATABLE = 0xC000;
```

```
//VALUE DIRECT für Programme dzt. nicht implementiert
```

```
public const UInt16 OBJNR_VALUE_BYTE = 0xF000;  
public const UInt16 OBJNR_VALUE_WORD = 0xF001;  
public const UInt16 OBJNR_VALUE_INT = 0xF002;  
public const UInt16 OBJNR_VALUE_CHAR = 0xF000;  
public const UInt16 OBJNR_VALUE_ARRAY = 0xF010;  
public const UInt16 OBJNR_VALUE_FLOAT = 0xF020;  
//alles andere ist immer ein ARRAY
```

```
//NICHTS
```

```
public const UInt16 OBJNR_NOTHING = 0xFFFF;
```

Datentypen

Jeder GET SET Befehl beinhaltet einen Datentypen Information, über die das Datenformat im Inhalt der Sendung definiert ist.

GET-Befehle können auch mit der Datentypen Option 0x00= Default abgesetzt werden.

Der Controller liefert in der Antwort immer die dafür vordefinierte Datentype, ansonsten den Dtype des GET/ SET Befehls.

```
public const Byte Dtype_DEFAULT = 0x00;
public const Byte Dtype_BYTE = 0x01;
public const Byte Dtype_WORD = 0x02;
public const Byte Dtype_INT = 0x03;
public const Byte Dtype_BYTEsigned = 0x04;
public const Byte Dtype_WORDsigned = 0x05;
public const Byte Dtype_INTsigned = 0x06;
public const Byte Dtype_ARRAY = 0x10;
public const Byte Dtype_FLOAT = 0x20;
public const Byte Dtype_TRIG_TIME = 0x30;
public const Byte Dtype_DATA_TEXTRETURN = 0x40;
public const Byte Dtype_LCDPAGE = 0x90;
public const Byte Dtype_PROGRAM = 0xA0;
public const Byte Dtype_NAME = 0xA1;
public const Byte Dtype_DALI = 0xA2;
public const Byte Dtype_MIDI = 0xA3;
public const Byte Dtype_PASSWORD = 0xFE;
public const Byte Dtype_NOT_SUPPORTED = 0xFF;
```

BEFEHL GET OBJECT DATA

GET COMMAND 0x28 / #40, zum Abfragen von einem oder mehreren Objekten

Format

[0x28 / #40], [Dtype / Option], [OBJNR High], [OBJNR Low], Option [Anzahl High], [Anzahl Low]

Abfrage von Szenen, Szenen Namen und der Szenen Programmierung

GET OBJECT DATA für OBJNR_SCENE:

Akzeptierte Datentypen:

| | |
|----------------------|----------------------------------|
| <i>Dtype_PROGRAM</i> | <i>für Szenen Programmierung</i> |
| <i>Dtype_NAME</i> | <i>für Szenen Namen</i> |

Akzeptierte Objekt Nummern:

OBJNR_SCENE = Szene 1

OBJNR_SCENE + (Maximale Anzahl der Szenen des DMXface -1) = letzte Szene

Akzeptierte Option Anzahl, Default = 1 wenn weggelassen:

Bei Namen bis max. 32 Stk.

Bei Szenenprogrammierung max. 1 Stk.

Return:

Dtype_NAME: pro abgefragten Namen fix. 32 Bytes, Terminierung des Strings mit 0x00 wenn kürzer

Dtype_PROGRAM: bis zu 1024 Bytes der Szenen Programmierung.

Beispiel:

Abfrage von 5 Szenen Namen ab Szene 100 (=OBJNR_SCENE +100 -1 = 0x63)

| | | | | |
|---------------|--------------------|---------------------|--|---------------------|
| Header | Länge Daten | Datenbereich | 6 Bytes | Terminierung |
| STX | LOW,HIGH | GET, | Dtype, OBJNR Word, Anzahl Word | |
| [0x01], | [0x06,0x00] | [0x28], | [0xA1], [0x00], [0x63], [0x00], [0x05] | [0x03] |

Antwort:

| | | | | |
|---------------|--------------------|---------------------|--|---------------------|
| Header | Länge Daten | Datenbereich | 164 Bytes | Terminierung |
| STX | LOW,HIGH | GET, | Dtype, OBJNR Word, Daten | |
| [0x01], | [0xA4,0x00] | [0xA8], | [0xA1], [0x00], [0x63], [5 x 32 Bytes] | [0x03] |

Abfrage von Programm Namen oder der Programminhalte

GET OBJECT DATA für OBJNR_PROGRAM:

Akzeptierte Datentypen:

| | |
|----------------------|---------------------------|
| <i>Dtype_PROGRAM</i> | <i>für Programm Daten</i> |
| <i>Dtype_NAME</i> | <i>für Programm Namen</i> |

Akzeptierte Objekt Nummern:

OBJNR_PROGRAM = Programm 1

OBJNR_PROGRAM + (Maximale Anzahl der Programme des DMXface -1) = letztes Programm

Akzeptierte Option Anzahl, Default = 1 wenn weggelassen:

Bei Namen bis max. 16 Stk.

Bei Programmierung max. 1 Stk.

Return:

Dtype_NAME: pro abgefragten Namen fix. 32 Bytes, Terminierung des Strings mit 0x00 wenn kürzer

Dtype_PROGRAM: 48 Bytes der Programmierung ggf. mehr bei höheren DMXface Versionen

Beispiel:

Abfrage von Programm 1

| | | | |
|---------------|--------------------|--|---------------------|
| Header | Länge Daten | Datenbereich 6 Bytes | Terminierung |
| STX | LOW,HIGH | GET, Dtype, OBJNR Word, Option Anzahl Word | |
| [0x01], | [0x06,0x00] | [0x28], [0xA0], [0x10], [0x00], [0x00], [0x01] | [0x03] |

Abfrage von DMX OUT Namen oder Kanal Werten

GET OBJECT DATA für OBJNR_DMXOUT:

Akzeptierte Datentypen:

| | |
|----------------------------------|--|
| <i>Dtype_DEFAULT, Dtype_BYTE</i> | <i>für Kanal Werte</i> |
| <i>Dtype_NAME</i> | <i>für DMX Out Kanal Namen</i> |
| <i>Dtype_DATA_TEXTRETURN</i> | <i>für Kanal Wert mit Text optional über Tabellenkonversion, wenn zugeordnet</i> |

Akzeptierte Objekt Nummern:

OBJNR_DMXOUT = DMX Kanal 1

OBJNR_DMXOUT+ (Maximale Anzahl der DMX Kanäle des DMXface -1) = letzter Kanal

Akzeptierte Option Anzahl, Default = 1 wenn weggelassen:

Bei Namen bis max. 32 Stk.

Bei Kanälen max. Anzahl der verfügbaren Kanäle im DMXface oder Buffer Limit

Bei *Dtype_DATA_TEXTRETURN* max. 1Stk.

Return:

Dtype_NAME: pro abgefragten Namen fix. 32 Bytes, Terminierung des Strings mit 0x00 wenn kürzer

Dtype_DEFAULT, Dtype_BYTE: Kanalwerte, 1 Byte pro Kanal

Dtype_DATA_TEXTRETURN: [Byte Kanalwert], [Byte ON / OFF], [String Text], [0x00 Terminierung]

Beispiel:

Abfrage von 100 DMX Kanälen ab DMX 1 (=OBJNR_DMXOUT +1 -1 = 0x2000)

| | | | | |
|---------------|--------------------|--|--------------------------|---------------------|
| Header | Länge Daten | Datenbereich | 6 Bytes | Terminierung |
| STX | LOW,HIGH | GET, Dtype, OBJNR | Word, Anzahl Word | |
| [0x01], | [0x06,0x00] | [0x28], [0x01], [0x20], [0x00], [0x00], [0x64] | | [0x03] |

Antwort:

| | | | | |
|---------------|--------------------|---|--------------------|---------------------|
| Header | Länge Daten | Datenbereich | 104 Bytes | Terminierung |
| STX | LOW,HIGH | GET, Dtype, OBJNR | Word, Daten | |
| [0x01], | [0x68,0x00] | [0xA8], [0x01], [0x20], [0x00], [100 Bytes] | | [0x03] |

Abfrage von DMX IN Kanal Werten

GET OBJECT DATA für OBJNR_DMXIN:

Akzeptierte Datentypen:

Dtype_DEFAULT, Dtype_BYTE für Kanal Werte

Akzeptierte Objekt Nummern:

OBJNR_DMXIN = DMX Eingangs Kanal 1

OBJNR_DMXIN+ (Maximale Anzahl der DMX IN Kanäle des DMXface -1) = letzter Kanal

Akzeptierte Option Anzahl, Default = 1 wenn weggelassen:

Bei Kanälen max. 512 Stk.

Return:

Dtype_DEFAULT, Dtype_BYTE: Kanalwerte, 1 Byte pro Kanal

Beispiel:

Abfrage von 512 DMX Input Kanälen ab DMX 1 (=OBJNR_DMXIN +1 -1 = 0x3000)

| | | | | |
|---------|-------------|--|-------------------|--------------|
| Header | Länge Daten | Datenbereich | 6 Bytes | Terminierung |
| STX | LOW,HIGH | GET, Dtype, OBJNR | Word, Anzahl Word | |
| [0x01], | [0x06,0x00] | [0x28], [0x01], [0x30], [0x00], [0x02], [0x00] | | [0x03] |

Antwort:

| | | | | |
|---------|-------------|--|-------------|--------------|
| Header | Länge Daten | Datenbereich | 104 Bytes | Terminierung |
| STX | LOW,HIGH | GET, Dtype, OBJNR | Word, Daten | |
| [0x01], | [0x04,0x02] | [0xA8], [0x01], [0x30], [0x00], [512Bytes] | | [0x03] |

Abfrage von TIMELINE Namen, Laufdaten und Programmierung

GET OBJECT DATA für OBJNR_TIMELINE:

Akzeptierte Datentypen:

Dtype_DEFAULT, Dtype_BYTE für Laufzeit Daten
 Dtype_NAME für Timeline Namen
 Dtype_PROGRAM für Timeline Programmierung

Akzeptierte Objekt Nummern:

OBJNR_TIMELINE = TIMELINE 1
 OBJNR_TIMELINE+ (Maximale Anzahl der TIMELINES des DMXface -1) = letzte TIMELINE

Akzeptierte Option Anzahl, Default = 1 wenn weggelassen:

Bei Namen bis max. 16 Stk.

Bei Programmierung und Laufzeitdaten 1 Stk.

Return:

Dtype_NAME: pro abgefragten Namen fix. 32 Bytes, Terminierung des Strings mit 0x00 wenn kürzer

Dtype_DEFAULT, Dtype_BYTE: Laufzeitdaten des Players (für alle Objektnummer gleich, es gibt nur einen Player)
 [Word Aktuelle Zeitposition Player x 50 msek, 0xFFFF = OFF]
 [Word Zeilen Position des Players]
 [Control Byte des Players 0= OFF / 1-n Timeline die gespielt wird]

Dtype_PROGRAM: Programmierung der Timeline 1KB Daten

Beispiel:

Abfrage des Player Status per OBJNR_TIMELINE = 0x4000

| | | | | |
|---------|-------------|--|--------------------------|--------------|
| Header | Länge Daten | Datenbereich | 6 Bytes | Terminierung |
| STX | LOW,HIGH | GET, Dtype, OBJNR | Word, option Anzahl Word | |
| [0x01], | [0x06,0x00] | [0x28], [0x01], [0x40], [0x00], [0x00], [0x01] | | [0x03] |

Antwort:

| | | | | |
|---------|-------------|--|-------------|--------------|
| Header | Länge Daten | Datenbereich | 9 Bytes | Terminierung |
| STX | LOW,HIGH | GET, Dtype, OBJNR | Word, Daten | |
| [0x01], | [0x09,0x00] | [0xA8], [0x01], [0x40], [0x00], [5Bytes] | | [0x03] |

Abfrage von Sequenz Namen oder der Programmierung

GET OBJECT DATA für OBJNR_SEQUENCE:

Akzeptierte Datentypen:

Dtype_PROGRAM für Szenen Programmierung
Dtype_NAME für Szenen Namen

Akzeptierte Objekt Nummern:

OBJNR_SEQUENCE = Sequenz 1

OBJNR_SEQUENCE + (Maximale Anzahl der Sequenzen des DMXface -1) = letzte Sequenz

Akzeptierte Option Anzahl, Default = 1 wenn weggelassen:

Bei Namen bis max. 16 Stk.

Bei Sequenz Programmierung max. 1 Stk.

Return:

Dtype_NAME: pro abgefragten Namen fix. 32 Bytes, Terminierung des Strings mit 0x00 wenn kürzer

Dtype_PROGRAM: bis zu 512 Bytes oder mehr der Sequenz Programmierung.

Beispiel:

Abfrage der Programmierung von Sequenz 1 (=OBJNR_SEQUENCE+1-1 = 0x5000)

| Header | Länge Daten | Datenbereich | Terminierung |
|---------|-------------|--|--------------|
| STX | LOW,HIGH | GET, Dtype, OBJNR Word, option Anzahl Word | |
| [0x01], | [0x06,0x00] | [0x28], [0xA0], [0x50], [0x00], [0x00], [0x01] | [0x03] |

Antwort:

| Header | Länge Daten | Datenbereich | Terminierung |
|---------|-------------|---|--------------|
| STX | LOW,HIGH | GET, Dtype, OBJNR Word, Daten | |
| [0x01], | [0x2,0x04] | [0xA8], [0xA0], [0x50], [0x00], [512 Bytes] | [0x03] |

Abfrage von Trigger Namen, Programmierung und Zeitschalt Settings

GET OBJECT DATA für OBJNR_TRIGGER:

Akzeptierte Datentypen:

| | |
|------------------------|--|
| <i>Dtype_NAME</i> | <i>für Timeline Namen</i> |
| <i>Dtype_PROGRAM</i> | <i>für Timeline Programmierung</i> |
| <i>Dtype_TRIG_TIME</i> | <i>für Zeitschalt Setting des Triggers</i> |

Akzeptierte Objekt Nummern:

OBJNR_TRIGGER = TRIGGER 1

OBJNR_TRIGGER + (Maximale Anzahl der TRIGGER des DMXface - 1) = letzter TRIGGER

Akzeptierte Option Anzahl, Default = 1 wenn weggelassen:

Bei Namen bis max. 16 Stk.

Bei Programmierung und Zeitschaltung 1 Stk.

Return:

| | |
|-------------------------|--|
| <i>Dtype_NAME:</i> | pro abgefragten Namen fix. 32 Bytes, Terminierung des Strings mit 0x00 wenn kürzer |
| <i>Dtype_TRIG_TIME:</i> | Return 8 Bytes mit den Zeitschaltinformationen des Triggers Zeitschaltfunktionen arbeiten mit Dezimalen Werten Binär hinterlegt. D.h. z.B. die Zeit 45 Sekunden wird als 0x45 hinterlegt. 00 = 0x00, 01 = 0x01, ..., 10 = 0x10 usw. |

[Byte ON / OFF]

[Byte Sekunden: Bit 7 = ON/OFF,]

[Byte Minuten: Bit 7 = ON/OFF, Bit 0-6 = Zeit dezimal als HEX]

[Byte Stunden: Bit 7 = ON/OFF, Bit 0-6 = Zeit dezimal als HEX]

[Byte Wochentag: Bit 7 = ON/OFF, Bit 0= Montag, Bit6 = Sonntag.]

[Byte Tag: Bit 7 = ON/OFF, Bit 0-6 = Tag dezimal als Hex]

[Byte Monat: Bit 7 = ON/OFF, Bit 0-6 = Monat dezimal als Hex]

[Byte Jahr: Bit 7 = ON/OFF, Bit 0-6 = Jahr dezimal als Hex]

Dtype_PROGRAM: Programmierung des Triggers

Beispiel:

Abfrage des Zeitschalt Settings Trigger 1

| | | | |
|---------|-------------|--|--------------|
| Header | Länge Daten | Datenbereich 6 Bytes | Terminierung |
| STX | LOW,HIGH | GET, Dtype, OBJNR Word, option Anzahl Word | |
| [0x01], | [0x06,0x00] | [0x28], [0x30], [0x60], [0x00], [0x00], [0x01] | [0x03] |

Antwort:

| | | | |
|---------|-------------|---|--------------|
| Header | Länge Daten | Datenbereich 12 Bytes | Terminierung |
| STX | LOW,HIGH | GET, Dtype, OBJNR Word, Daten | |
| [0x01], | [0x0C,0x00] | [0xA8], [0x30], [0x60], [0x00], [8 Bytes] | [0x03] |

Abfrage von Output Namen oder Zustand

GET OBJECT DATA für OBJNR_OUTPORT

Akzeptierte Datentypen:

Dtype_DEFAULT, Dtype_BYTE für Output Zustand
Dtype_NAME für Output Namen

Akzeptierte Objekt Nummern:

OBJNR_OUTPORT = OUTPORT 1

OBJNR_OUTPORT+ (Maximale Anzahl der OUTPUTS =24 -1) = letzter Output egal ob physisch vorhanden

Akzeptierte Option Anzahl, Default = 1 wenn weggelassen:

Bei Namen bis max. 16 Stk.

Bei Kanälen max. 24 Stk.

Return:

Dtype_NAME: pro abgefragten Namen fix. 32 Bytes, Terminierung des Strings mit 0x00 wenn kürzer

Dtype_DEFAULT, Dtype_BYTE: 1 Byte pro Output [0x00 = OFF] oder [0x01 = ON]

Beispiel:

Abfrage von 16 Outputs ab OUT 1 (=OBJNR_OUTPORT +1 -1 = 0x7000)

| | | | |
|---------|-------------|--|--------------|
| Header | Länge Daten | Datenbereich 6 Bytes | Terminierung |
| STX | LOW,HIGH | GET, Dtype, OBJNR Word, Anzahl Word | |
| [0x01], | [0x06,0x00] | [0x28], [0x01], [0x70], [0x00], [0x00], [0x10] | [0x03] |

Antwort:

| | | | |
|---------|-------------|--|--------------|
| Header | Länge Daten | Datenbereich 20 Bytes | Terminierung |
| STX | LOW,HIGH | GET, Dtype, OBJNR Word, Daten | |
| [0x01], | [0x14,0x00] | [0xA8], [0x01], [0x70], [0x00], [16 Bytes] | [0x03] |

Abfrage von Inport und Busport Namen oder Zustand analog oder digital

GET OBJECT DATA für OBJNR_PORTS

Akzeptierte Datentypen:

Dtype_DEFAULT, Dtype_BYTE für (Bus) Inport Zustand oder AD Wert
Dtype_NAME für (Bus) Inport Namen
Dtype_DATA_TEXTRETURN für (Bus) Inport Zustand inkl. Text Konvertierung (opt. Tabelle)

Akzeptierte Objekt Nummern:

Zugriff auf digitalen Zustand des Inport

OBJNR_INPORT = INPORT 1

OBJNR_INPORT + (Maximale Anzahl der INPORTS =24 -1) = letzter Inport egal ob physisch vorhanden.

Zugriff auf analogen und digitalen Zustand des Inport

OBJNR_INPORT_AD = INPORT 1

OBJNR_INPORT_AD+ (Maximale Anzahl der INPORTS =16 -1) = Max 16 Inports mit AD Funktion egal ob physisch vorhanden.

OBJNR_BUSPORT = BUSPORT 1

OBJNR_BUSPORT + (Maximale Anzahl der BUSports =32 -1) = letzter BUSport

Zugriff auf analogen und digitalen Zustand des Inport

OBJNR_BUSPORT_AD = BUSPORT 1

OBJNR_BUSPORT_AD + (Maximale Anzahl der BUSports =32 -1) = letzter BUSport

Akzeptierte Option Anzahl, Default = 1 wenn weggelassen:

Bei Namen bis max. 16 Stk.

Bei AD und Digitem Zustand max. 32 Stk.

Dtype_DATA_TEXTRETURN max. 1Stk.

Return:

Dtype_NAME

pro abgefragten Namen fix. 32 Bytes, Terminierung des Strings mit 0x00 wenn kürzer

Dtype_DEFAULT, Dtype_BYTE bei BUSPORT und INPORT Objekten

1 Byte pro Inport / Busport [0x00 = OFF] oder [0x01 = ON]

Dtype_DEFAULT, Dtype_BYTE bei BUSPORT_AD und INPORT_AD Objekten

Word AD Value + 1 Byte Zustand digital pro abgefragten Kanal

Dtype_DATA_TEXTRETURN

AD Value Word, ON OFF State Byte, Konvertierter Text, optional aus Tabelle, wenn Kanal zugewiesen, Terminierung 0x00

Beispiel:

Abfrage von 16 Inports ab IN1 (=OBJNR_INPORT +1 -1 = 0x8000)

| | | | | |
|---------|-------------|--|-------------------|--------------|
| Header | Länge Daten | Datenbereich | 6 Bytes | Terminierung |
| STX | LOW,HIGH | GET, Dtype, OBJNR | Word, Anzahl Word | |
| [0x01], | [0x06,0x00] | [0x28], [0x01], [0x80], [0x00], [0x00], [0x10] | | [0x03] |

Antwort:

| | | | | |
|---------|-------------|--|-------------|--------------|
| Header | Länge Daten | Datenbereich | 20 Bytes | Terminierung |
| STX | LOW,HIGH | GET, Dtype, OBJNR | Word, Daten | |
| [0x01], | [0x14,0x00] | [0xA8], [0x01], [0x80], [0x00], [16 Bytes] | | [0x03] |

Abfrage der Timer Zustände

GET OBJECT DATA für OBJNR_TIMER

Akzeptierte Datentypen:

Dtype_DEFAULT, Dtype_BYTE *Timer aktuelle Laufzeitdaten*
Dtype_NAME *Timer Name*

Akzeptierte Objekt Nummern:

OBJNR_TIMER = Timer 1

OBJNR_TIMER + (Maximale Anzahl der Timer =8 -1) = letzter Timer des DMXface

Akzeptierte Option Anzahl, Default = 1 wenn weggelassen:

Bei Namen und Laufzeitdaten bis max. Anzahl der verfügbaren Timer (dzt. Firmware 5.31 = 8)

Return:

Dtype_NAME

pro abgefragten Namen fix. 32 Bytes, Terminierung des Strings mit 0x00 wenn kürzer

Dtype_DEFAULT

Dtype_BYTE

Word, nächster CALL Object NR dzt. Leer

Word Timer Zeit x50msek (0xFFFF = nicht aktiv)

Byte, reserviert 0x00

Byte, nächste Szenen / Programm Nr. nach altem Standard

Byte, reserviert 0x00

Byte Timer Status Flags (0x01 Audio, 0x20 immediate, 0x40 Reset Trig, 0x80 Step next)

Beispiel:

Abfrage von 8 Timer Zuständen ab Timer 1 (=OBJNR_TIMER +1 -1 = 0x9000)

| | | | | |
|----------------|--------------------|---------------------|------------------------|-------------------------------|
| Header | Länge Daten | Datenbereich | 6 Bytes | Terminierung |
| STX | LOW,HIGH | GET, | Dtype, OBJNR | Word, Anzahl Word |
| [0x01], | [0x06,0x00] | [0x28], | [0x01], [0x90], | [0x00], [0x00], [0x08] |
| | | | | [0x03] |

Antwort:

| | | | | |
|----------------|--------------------|---------------------|------------------------|---------------------------|
| Header | Länge Daten | Datenbereich | 68 Bytes | Terminierung |
| STX | LOW,HIGH | GET, | Dtype, OBJNR | Word, Daten |
| [0x01], | [0x44,0x00] | [0xA8], | [0x01], [0x90], | [0x00], [64 Bytes] |
| | | | | [0x03] |

Abfrage von Buffern

GET OBJECT DATA für OBJNR_BUFFERS

Akzeptierte Datentypen:

Dtype_DEFAULT, Dtype_BYTE für Datenabfrage
Dtype_NAME Name (nur für OBJNR_CHARBUFFER)

Akzeptierte Objekt Nummern:

OBJNR_RTCTIME RTC Echtzeituhr
OBJNR_DMXMASTER DMX Master Wert
OBJNR_COM1BUFFER RX Datenbuffer Serial Port 1
OBJNR_COM2BUFFER RX Datenbuffer Serial Port 2
OBJNR_USBBUFFER RX Datenbuffer USB
OBJNR_RS485BUFFER RX Datenbuffer RS485
OBJNR_LANBUFFER RX Datenbuffer LAN (Alle Sockets)
OBJNR_IRBUFFER RX Datenbuffer Infrarot
OBJNR_CHARBUFFER CHAR BUFFER

Akzeptierte Option Anzahl immer 1, Default = 1 wenn weggelassen:

Return:

OBJNR_RTCTIME Byte Sekunde (Dezimal als Hexwert !)
 Byte Minute (Dezimal als Hexwert !)
 Byte Stunde (Dezimal als Hexwert !)
 Byte Wochentag (01 Montag .. 07 Sonntag)
 Byte Tag (Dezimal als Hexwert !)
 Byte Monat (Dezimal als Hexwert !)
 Byte Jahr (Dezimal als Hexwert !)
 Byte Sommerzeit Setting
 (Bit 0x01 Automatische Umstellung 1=ON / 0=OFF)
 (Bit 0x80 Aktuelle Zeit ist 0= Winter / 1 = Sommer)

OBJNR_CHARBUFFER je nach Firmwareversion min 21 Bytes
 String Terminiert mit 0x00

OBJNR_DMXMASTER 1 Byte DMX Master Einstellung

OBJNR_IRBUFFER Byte Status Capture
 0x80 = Fertiger Capture
 0x40 = Flankenerwartung
 0x20 = Sequence Send Mode
 0x10 = Sequence Capture Mode
 0x08 = Sequence Capture complete
 0x04 = Sequence Output suppress edge
 0x02 = First Edge
 0x01 not used
 8 Byte Code 1 Capture
 8 Byte Code 2 Capture

OBJNR_COM1BUFFER
OBJNR_COM2BUFFER
OBJNR_USBBUFFER
OBJNR_RS485BUFFER
OBJNR_LANBUFFER

Aktuelle im entsprechenden RX Buffer befindlichen Daten,
länge je nach Inhalt

Bei LAN nur ein RX Buffer gemeinsam für alle Sockets!

Abfrage von LCD Page Namen oder der Programmierung**GET OBJECT DATA für OBJNR_LCDPAGE:**

Akzeptierte Datentypen:

Dtype_PROGRAM

Abfrage der LCDpage mit 1024 Bytes (komplette Seite)

Dtype_LCDPAGE

Abfrage der LCDpage im aktuellen Standard mit ersten 512 Bytes

Dtype_NAME

Name der LCD-Page

Akzeptierte Objekt Nummern:

OBJNR_LCDPAGE = LCDPAGE 0

OBJNR_LCDPAGE + (Maximale Anzahl der PAGES des DMXface -1) = letzte PAGE

Akzeptierte Option Anzahl, Default = 1 wenn weggelassen:

Bei Dtype_NAME bis max. 16 Stk.

Bei Dtype_PROGRAM / Dtype_LCDPAGE max. 1 Stk.

Return:

Dtype_NAME: pro abgefragten Namen fix. 32 Bytes, Terminierung des Strings mit 0x00 wenn kürzer

Dtype_PROGRAM: 512 Bytes der LCDPAGE Programmierung

Dtype_LCDPAGE: 1024 Bytes der LCDPAGE Programmierung

Abfrage von TEXT Pages (Legacy)

GET OBJECT DATA für OBJNR_TEXTPAGE:

Direkter Zugriff auf 1KB Blöcke der Textspeicher, primär zum schnellen löschen des Speichers.

Zum normalen Abfragen und speichern, die Set / Dtype_Name Option mit den jeweiligen Objektnummern verwenden.

Akzeptierte Datentypen:

Dtype_DEFAULT

Dtype_BYTE

Dtype_ARRAY

Dtype_NAME

Alle Abfrage liefern einen 1KB Block aus den Texten der jeweiligen Page lt. Tabelle, Jeder Einzeltext belegt 32 Byte also 32 Texte pro KB / Text Page

Die Zuordnung der Bezeichnungs-Texte zu den TEXT Pages bis Ver. 5.40

| Text Page Objekt Nummer | Text Nummer (legacy) | Bezeichnungs-Text von |
|-------------------------|----------------------|-----------------------|
| 0xB800 | 0-31 | DMXOUT CH1- CH32 |
| 0xB801 | 32-63 | DMXOUT CH33-CH64 |
| 0xB802 | 64-95 | DMXOUT CH65-CH96 |
| 0xB803 | 96-127 | DMXOUT CH97-CH128 |
| 0xB804 | 128-159 | DMXOUT CH129-CH160 |
| 0xB805 | 160-191 | DMXOUT CH161-CH192 |
| 0xB806 | 192-223 | DMXOUT CH193-CH224 |
| 0xB807 | 224-255 | DMXOUT CH225-CH256 |
| 0xB808 | 256-287 | Szene 1 - 32 |
| 0xB809 | 288-319 | Szene 33 - 64 |
| 0xB80A | 320-351 | Szene 65 - 96 |
| 0xB80B | 352-383 | Szene 97 - 128 |
| 0xB80C | 384-415 | Szene 129 - 160 |
| 0xB80D | 416-447 | Szene 161 - 192 |
| 0xB80E | 448-479 | Szene 193 - 224 |
| 0xB80F | 480-511 | Szene 225 - 256 |
| 0xB810 | 512-543 | Trigger 1 -32 |
| 0xB811 | 544-575 | Trigger 33 -64 |
| 0xB812 | 576-607 | Trigger 65 - 96 |
| 0xB813 | 608-639 | Trigger 97 - 128 |
| 0xB814 | 640-671 | Inport 1 - 32 |
| 0xB815 | 672-703 | Inport 33 - 64 |
| 0xB816 | 704-735 | Inport 65 - 96 |
| 0xB817 | 736-767 | Inport 97 - 128 |
| 0xB818 | 768-799 | Output 1 - 32 |
| 0xB819 | 800-831 | Programm 1 - 32 |
| 0xB81A | 832-863 | Programm 33-64 |
| 0xB81B | 864-895 | Sequenz 1 - 32 |
| 0xB81C | 896-927 | Sequenz 33 - 64 |
| 0xB81D | 928-959 | My Control Pages |
| 0xB81E | 960-991 | Data Tables 9-16 |
| 0xB81F | 992-1023 | Data Tables 1-8 |
| 0xB820 | 1024-1055 | DMX 257- 288 |
| 0xB821 | 1056-1087 | DMX 289 – 320 |
| 0xB822 | 1088-1119 | DMX 321 - 352 |
| 0xB823 | 1120-1151 | DMX 353- 384 |
| 0xB824 | 1152-1183 | DMX 385 - 416 |
| 0xB825 | 1184-1215 | DMX 417 - 448 |
| 0xB826 | 1216-1247 | DMX 449 - 480 |
| 0xB827 | 1248-1279 | DMX 481 - 512 |
| 0xB828 | 1280-1311 | DMX 513 - 544 |
| 0xB829 | 1312-1343 | frei |
| 0xB82A | 1344-1375 | frei |
| 0xB82B | 1376-1407 | frei |
| 0xB82C | 1408-1439 | LCDPAGE 0-31 |

| | | |
|--------|-----------|--|
| 0xB82D | 1440-1471 | LCDPAGE 32-63 |
| 0xB82E | 1472-1503 | TIMELINE 1-16 und TIMER 1-16 |
| 0xB82F | 1504-1535 | frei |
| 0xB830 | 1536-1567 | frei |
| 0xB831 | 1568-1599 | frei |
| 0xB832 | 1600-1631 | frei |
| 0xB833 | 1632-1663 | frei |
| 0xB834 | 1664-1695 | frei |
| 0xB835 | 1696-1727 | frei |
| 0xB836 | 1728-1759 | |
| 0xB837 | 1760-1791 | Reserviert Applikation Programm Data, reserviert zum Ablegen von Infos aus DMXface-Console (letzte Änderung, Dateiname, ...) |
| | | |
| | | |

GET OBJECT DATA für OBJNR_DATATABLE:

Akzeptierte Datentypen:

Dtype_PROGRAM

für Datentabellen Programmierung (inkl. Name)

Akzeptierte Objekt Nummern:

OBJNR_DATATABLE = Tabelle 1

OBJNR_DATATABLE + (Maximale Anzahl der Tabellen des DMXface -1) = letzte Tabelle

Akzeptierte Option Anzahl, Default = 1 wenn weggelassen:

Bis max. 8 Stk.

Return:

Dtype_PROGRAM: 128 Bytes der Tabellen Programmierung x Anzahl

| | |
|-----------|---|
| Byte | Funktion 0x00 = OFF, 0x01 = Interpolation , 0x02 = Textmode |
| Byte [20] | Name der Tabelle |
| Byte | Anzahl der Kommastellen |
| Byte [8] | Einheiten Text |
| Byte [8] | Response bei Ausgabe unter kleinstem Tabellen Wert |
| Byte [8] | Response bei Ausgabe über höchstem Tabellen Wert |
| Byte [32] | Zugewiesene Objects |
| Byte [10] | Tabellen Punkt 1 |
| Byte [10] | Tabellen Punkt 2 |
| Byte [10] | Tabellen Punkt 3 |
| Byte [10] | Tabellen Punkt 4 |
| Byte [10] | Tabellen Punkt 5 |

BEFEHL SET OBJECT DATA

Das Set Command 0x29 / #41 ist ein universaler Befehl, der in Kombination mit dem Datentype Dtype, der Objektnummer und den zugehörigen Daten gesendet wird. Je nach Dtype und Objekttype stehen unterschiedliche Möglichkeiten zur Verfügung die Objekte zu steuern.

Format

[0x29 / #41], [Dtype / Option], [OBJNR High], [OBJNR Low], **Databyte0, Databyte1,**

SET OBJECT DATA für OBJNR_SCENE:

Akzeptierte Datentypen:

Dtype_BYTE

Szenen Abruf,

optional mit alternativer Helligkeit (DataByte0)

0=0% / 0xFF = 100%

optional mit alternativer Fadezeit (Word DataByte 1+2)

100mSek Schritte 0x0000 bis max. 0x1388 = 500Sekunden

Dtype_PROGRAM

Schreiben der Szenen Programmierung

Dtype_NAME

Schreiben des Szenen Namens

Beispiel Szenen Abruf Szene 10

| Protokoll | | | Daten | | | | | | | |
|-----------|------------|-------------|-------|----------------------|---------------|--------------|------|--|--|--|
| STX | LEN low | LEN high | SET | Dtype | OBJNR high | OBJNR low | ETX | | | |
| 0x01 | 0x04 | 0x00 | 0x29 | 0x01 oder 0x00 | 0x00 | 0x09 | 0x03 | | | |
| Return | | | | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0x01 | 0x00 | 0x09 | 0x03 | | | |

Beispiel Szenen Abruf Szene 20 mit Submaster Wert 50%

| Protokoll | | | Daten | | | | | | | |
|-----------|------------|-------------|-------|----------------------|---------------|--------------|---------------|------|--|--|
| STX | LEN low | LEN high | SET | Dtype | OBJNR high | OBJNR low | Sub Master | ETX | | |
| 0x01 | 0x04 | 0x00 | 0x29 | 0x01 oder 0x00 | 0x00 | 0x14 | 0x80 | 0x03 | | |
| Return | | | | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0x01 | 0x00 | 0x14 | 0x80 | 0x03 | | |

Beispiel Szenen Abruf Szene 30 mit Submaster Wert 80% und Fadezeit 100Sek.*

| Protokoll | | | Daten | | | | | | | | | |
|-----------|------------|-------------|-------|----------------------|---------------|--------------|---------------|---------------|--------------|------|--|--|
| STX | LEN low | LEN high | SET | Dtype | OBJNR high | OBJNR low | Sub Master | *FADE High | *Fade Low | ETX | | |
| 0x01 | 0x04 | 0x00 | 0x29 | 0x01 oder 0x00 | 0x00 | 0x1E | 0xCC | 0x03 | 0xE8 | 0x03 | | |
| Return | | | | | | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0x01 | 0x00 | 0x1E | 0xCC | 0x03 | 0xE8 | 0x03 | | |

*Die Fadezeit wird in 100msek Schritten angegeben die Maximale Zeit ist 500 Sekunden

SET OBJECT DATA für OBJNR_DMXOUT:

Akzeptierte Datentypen:

Dtype_BYTE, Dtype_DEFAULT
DMX Kanal setzen (Byte 0), optional mit alternativer Fadezeit x16 (Byte 1+2) = Fadezeit x 100msek.
Dtype_ARRAY
Mehrere DMX-Kanäle auf Wert setzen Byte 0-n, Start bei OBJNR
Dtype_DATA_TEXTRETURN
Einen DMX-Kanal setzen mit der Rückantwort des Kanalwertes als Byte + Text optional aus Tabellenkonversion
Dtype_NAME
Schreiben DMX Kanal Namens

 Beispiel *Dtype_BYTE / Dtype_DEFAULT*:

Setzen des DMXkanal 10 auf Wert 100 mit Fadezeit 1 Sek. (Fadezeit = Wert 16Bit x 100msek)

| Protokoll | LEN | LEN | Daten | | | | | | | |
|-----------|------|------|-------|----------------|------------|-----------|----------|-----------|----------|------|
| STX | low | high | SET | Dtype | OBJNR high | OBJNR low | DMX Wert | FADE high | FADE low | ETX |
| 0x01 | 0x07 | 0x00 | 0x29 | 0x01 oder 0x00 | 0x20 | 0x09 | 0x64 | 0x00 | 0x0A | 0x03 |
| Return | | | | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0x01 | 0x20 | 0x09 | 0x03 | | | |

Setzen der DMXkanäle ab 200 auf Werte 10,20,30,40,50

| Protokoll | LEN | LEN | Daten | | | | | | | | | |
|-----------|------|------|-------|-------|------------|-----------|-----------|-----------|-----------|-----------|-----------|------|
| STX | low | high | SET | Dtype | OBJNR high | OBJNR low | DMX Wert1 | DMX Wert2 | DMX Wert3 | DMX Wert4 | DMX Wert5 | ETX |
| 0x01 | 0x09 | 0x00 | 0x29 | 0x0E | 0x20 | 0xC7 | 0x0A | 0x14 | 0x1E | 0x28 | 0x32 | 0x03 |
| Return | | | | | | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0x0E | 0x20 | 0xC7 | 0x03 | | | | | |

Setzen des DMXkanal 500 auf Werte 255 mit Textreturn

| Protokoll | LEN | LEN | Daten | | | | | | | | |
|-----------|------|------|-------|-------|------------|-----------|----------|------------------|------|--|--|
| STX | low | high | SET | Dtype | OBJNR high | OBJNR low | DMX Wert | ETX | | | |
| 0x01 | 0x05 | 0x00 | 0x29 | 0x40 | 0x20 | 0xC7 | 0xFF | 0x03 | | | |
| Return | | | | | | | DMX Wert | Wert als String* | | | |
| 0x01 | 0x09 | 0x00 | 0xA9 | 0x40 | 0x21 | 0xF3 | 0xFF | „ 255“ | 0x03 | | |

*ist der DMX-Kanal einer Tabelle zugeordnet so wird der String aus der Tabelle generiert. Die Länge variiert je nach Ausgabe aus der Tabelle

Ändern des Namen des DMX Kanal 1 auf „ABDCE“

| Protokoll | LEN | LEN | Daten | | | | | | |
|-----------|------|------|-------|-------|------------|-----------|---------|-----------|------|
| STX | low | high | SET | Dtype | OBJNR high | OBJNR low | String | Terminert | ETX |
| 0x01 | 0x0A | 0x00 | 0x29 | 0xA1 | 0x20 | 0x00 | „ABCDE“ | 0x00 | 0x03 |
| Return | | | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0xA1 | 0x20 | 0x00 | 0x03 | | |

Die maximale Länge eines Namens ist 32Byte,

am Ende des Strings immer ein Terminerungsbyte 0x00 schicken.

Das Kommando ist sinnvoll, wenn einzelne DMX-Kanäle umbenannt werden sollen. Im System muss dazu immer ein kompletter 1KB Block neu geflasht werden.

Für die Änderungen mehrerer DMX Kanal Namens sollen in jedem Fall die Objekte für den direkten Zugriff auf Textpages verwendet werden um unnötige FLASH Vorgänge zu vermeiden. Mit diesen Objekten werden komplette 1KB Text Pages geändert.

SET OBJECT DATA für OBJNR_TIMELINE:

Akzeptierte Datentypen:

Dtype_PROGRAM

Dtype_NAME

Dtype_Default, Dtype_BYTE

Schreiben der Timeline Programmierung 1024Byte Länge

Ändern des Namens der Timeline

 Starten der Timeline am Player optional mit setzen der
Abspielzeitpunktes (Word x50msek)

Beispiel Timeline 1 starten

| Protokoll | LEN | LEN | Daten | | | | |
|-----------|------|------|-------|----------------------|-------|-------|------|
| STX | low | high | SET | Dtype | OBJNR | OBJNR | ETX |
| 0x01 | 0x04 | 0x00 | 0x29 | 0x00 oder 0x01 | 0x40 | 0x00 | 0x03 |
| Return | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0x01 | 0x40 | 0x00 | 0x03 |

Beispiel Timeline 3 starten am Abspielzeitpunkt 500,00 Sekunden (500 / 0.05sek = 10000 = 0x2710)

| Protokoll | LEN | LEN | Daten | | | | | | |
|-----------|------|------|-------|----------------------|-------|-------|------|------|------|
| STX | low | high | SET | Dtype | OBJNR | OBJNR | TIME | TIME | ETX |
| 0x01 | 0x06 | 0x00 | 0x29 | 0x00 oder 0x01 | 0x40 | 0x02 | 0x27 | 0x10 | 0x03 |
| Return | | | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0x01 | 0x40 | 0x02 | 0x03 | | |

Beispiel Timeline Player stoppen ohne Zeit zu verändern (OBJNR_TIMELINE_STOP)

Optional kann auch eine Zeitangabe mitgesendet werden um den Abspielzeitpunkt neu zu setzen

| Protokoll | LEN | LEN | Daten | | | | |
|-----------|------|------|-------|----------------------|-------|-------|------|
| STX | low | high | SET | Dtype | OBJNR | OBJNR | ETX |
| 0x01 | 0x06 | 0x00 | 0x29 | 0x00 oder 0x01 | 0x4F | 0xFE | 0x03 |
| Return | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0x01 | 0x4F | 0xFE | 0x03 |

Beispiel Timeline Player Stop + Reset, Zeit auf 0 setzen (OBJNR_TIMELINE_RESET)

| Protokoll | LEN | LEN | Daten | | | | |
|-----------|------|------|-------|----------------------|-------|-------|------|
| STX | low | high | SET | Dtype | OBJNR | OBJNR | ETX |
| 0x01 | 0x06 | 0x00 | 0x29 | 0x00 oder 0x01 | 0x4F | 0xFF | 0x03 |
| Return | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0x01 | 0x4F | 0xFF | 0x03 |

Ändern des Namens der Timeline 4

| Protokoll | LEN | LEN | Daten | | | | | | |
|-----------|------|------|-------|-------|-------|-------|-----------------|------|--|
| STX | low | high | SET | Dtype | OBJNR | OBJNR | String | ETX | |
| 0x01 | 0x24 | 0x00 | 0x29 | 0xA1 | 0x40 | 0x03 | Text 32 Byte | 0x03 | |
| Return | | | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0xA1 | 0x40 | 0x03 | 0x03 | | |

SET OBJECT DATA für OBJNR_TRIGGER:

Akzeptierte Datentypen:

Dtype_PROGRAM

Schreiben der Trigger Programmierung

Dtype_NAME

Ändern des Trigger Namen

Dtype_Default, Dtype_BYTE

Ausführen des Triggers

Dtype_TRIGGER_TIME

Zeitschalt Setting des Triggers überschreiben

Beispiel Trigger 1 abrufen / ausführen wie wenn eines der Trigger Events eingetreten wäre

| Protokoll | | | Daten | | | | | |
|-----------|------------|-------------|-------|----------------------|---------------|--------------|--|------|
| STX | LEN low | LEN high | SET | Dtype | OBJNR high | OBJNR low | | ETX |
| 0x01 | 0x04 | 0x00 | 0x29 | 0x00 oder 0x01 | 0x60 | 0x00 | | 0x03 |
| Return | | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0x01 | 0x60 | 0x00 | | 0x03 |

Beispiel Trigger 2 neu programmieren

| Protokoll | | | Daten | | | | | |
|-----------|------------|-------------|-------|-------|---------------|--------------|------------|------|
| STX | LEN low | LEN high | SET | Dtype | OBJNR high | OBJNR low | Daten | ETX |
| 0x01 | 0x36 | 0x00 | 0x29 | 0xA0 | 0x60 | 0x01 | 50 Byte | 0x03 |
| Return | | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0xA0 | 0x60 | 0x01 | | 0x03 |

Trigger 10 Zeitschaltsetting neu programmieren

| Protokoll | | | Daten | | | | | |
|-----------|------------|-------------|-------|-------|---------------|--------------|--------------|------|
| STX | LEN low | LEN high | SET | Dtype | OBJNR high | OBJNR low | Zeit Setting | ETX |
| 0x01 | 0x0C | 0x00 | 0x29 | 0x30 | 0x60 | 0x09 | 8 Bytes | 0x03 |
| Return | | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0x01 | 0x60 | 0x09 | | 0x03 |

Trigger Zeit Setting Bytes

| | |
|--------|--|
| Byte 0 | ON = 0x80 / OFF = 0x00 |
| Byte 1 | Zeit Sekunden, ON / OFF = Bit 7, Bit 0-6 = Sekunden dezimal als HEX eingesetzt |
| Byte 2 | Zeit Minuten, ON / OFF = Bit 7, Bit 0-6 = Minuten dezimal als HEX eingesetzt |
| Byte 3 | Zeit Stunden, ON / OFF = Bit 7, Bit 0-6 = Stunden dezimal als HEX eingesetzt |
| Byte 4 | Wochentage, ON / OFF = Bit 7, Bit 0-6 = Montag - Sonntag |
| Byte 5 | Zeit Tag, ON / OFF = Bit 7, Bit 0-6 = Tag dezimal als HEX eingesetzt |
| Byte 6 | Zeit Monat, ON / OFF = Bit 7, Bit 0-6 = Monat dezimal als HEX eingesetzt |
| Byte 7 | Zeit Jahr, ON / OFF = Bit 7, Bit 0-6 = Jahr dezimal als HEX eingesetzt |

Ändern des Namens des Trigger 4

| Protokoll | | | Daten | | | | | |
|-----------|------------|-------------|-------|-------|---------------|--------------|-----------------|------|
| STX | LEN low | LEN high | SET | Dtype | OBJNR high | OBJNR low | String | ETX |
| 0x01 | 0x24 | 0x00 | 0x29 | 0xA1 | 0x60 | 0x03 | Text 32 Byte | 0x03 |
| Return | | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0xA1 | 0x60 | 0x03 | | 0x03 |

SET OBJECT DATA für OBJNR_OUTPORT:

Akzeptierte Datentypen:

Dtype_Default, Dtype_BYTE, Dtype_ARRAY
Dtype_NAME
Ein oder mehrere Outport setzen
Ändern des OUTPORT Namen

Beispiel Outport 1 setzen

| Protokoll | | | Daten | | | | | |
|-----------|------------|-------------|-------|--------------------------------------|---------------|--------------|---|------|
| STX | LEN low | LEN high | SET | Dtype | OBJNR high | OBJNR low | Wert | ETX |
| 0x01 | 0x04 | 0x00 | 0x29 | 0x00 oder 0x01 oder 0x10 | 0x70 | 0x00 | 0x00 = OFF 0x01-0xFF = ON Auch mehrere Bytes hintereinander für die nachfolgenden Outports | 0x03 |
| Return | | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0x01 | 0x70 | 0x00 | 0x03 | |

Ändern des Outport 3 Namens

| Protokoll | | | Daten | | | | | |
|-----------|------------|-------------|-------|-------|---------------|--------------|----------------------|------|
| STX | LEN low | LEN high | SET | Dtype | OBJNR high | OBJNR low | String | ETX |
| 0x01 | 0x24 | 0x00 | 0x29 | 0xA1 | 0x70 | 0x02 | Text max. 32 Byte | 0x03 |
| Return | | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0xA1 | 0x70 | 0x02 | 0x03 | |

SET OBJECT DATA für OBJNR_LCDPAGE:

Akzeptierte Datentypen:

Dtype_PROGRAM

Schreiben der von LCDPAGE Daten mit 512 oder 1024Byte Länge auf die LCD-PAGE

Dtype_NAME

Ändern des Namens der LCD-PAGE

Akzeptierte Objekt Nummern:

OBJNR_LCDPAGE= LCD-PAGE 0 bis (OBJNR_LCDPAGE + Anzahl der verfügbaren Displayseiten -1)

Beispiel LCD Page 0 neu programmieren mit 512 Byte langer Page

| Protokoll | | | Daten | | | | | | | | | |
|-----------|---------|----------|-------|-------|------------|-----------|-----------|------|--|--|--|--|
| STX | LEN low | LEN high | SET | Dtype | OBJNR high | OBJNR low | Daten | ETX | | | | |
| 0x01 | 0x04 | 0x02 | 0x29 | 0xA0 | 0xB0 | 0x00 | 512 Bytes | 0x03 | | | | |
| Return | | | | | | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0xA0 | 0xB0 | 0x00 | | 0x03 | | | | |

SET OBJECT DATA für OBJNR_TEXTPAGE:

Akzeptierte Datentypen:

Dtype_DEFAULT

Dtype_BYTE

Dtype_ARRAY

Dtype_NAME

Alle Datentypen schreiben 1024 Bytes in die adressierte Text Page

Beispiel Szenen Namen 1-32 neu programmieren – Text PAGE 0xB808

| Protokoll | | | Daten | | | | | | | | | |
|-----------|---------|----------|-------|-------|------------|-----------|------------------|------|--|--|--|--|
| STX | LEN low | LEN high | SET | Dtype | OBJNR high | OBJNR low | Daten | ETX | | | | |
| 0x01 | 0x04 | 0x04 | 0x29 | 0x01 | 0xB8 | 0x08 | 1024 Bytes Texte | 0x03 | | | | |
| Return | | | | | | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0x01 | 0xB8 | 0x08 | | 0x03 | | | | |

SET OBJECT DATA für OBJNR_DATATABLE:

Akzeptierte Datentypen:

Dtype_PROGRAM
für Datentabellen Programmierung (inkl. Name)

Akzeptierte Objekt Nummern:

OBJNR_DATATABLE = Tabelle 1
OBJNR_DATATABLE + (Maximale Anzahl der Tabellen des DMXface -1) = letzte Tabelle

Schreiben der Tabelle 1

| | | | | | | | | | |
|-----------|------------|-------------|-------|-------|---------------|--------------|---------------------|------|--|
| Protokoll | | | Daten | | | | | | |
| STX | LEN low | LEN high | SET | Dtype | OBJNR high | OBJNR low | ARRAY | ETX | |
| 0x01 | 0x84 | 0x00 | 0x29 | 0xA0 | 0xC0 | 0x00 | 128Byte Struktur | 0x03 | |
| Return | | | | | | | | | |
| 0x01 | 0x04 | 0x00 | 0xA9 | 0xA0 | 0xC0 | 0x00 | 0x03 | | |

| | |
|--------------|--|
| | Struktur eines Konversionstabelle 128 Bytes |
| Byte 0 | Funktion 0x00 = OFF, 0x01 = Interpolation , 0x02 = Textmode |
| Byte 1-20 | Tabellen Name, unbenutzte Bytes = 0x00 |
| Byte 21 | Anzahl der Kommastellen |
| Byte 22-29 | Einheit Text, unbenutzte Bytes = 0x00 |
| Byte 30-37 | Ausgabe Text bei Ausgabe unter kleinstem Tabellenwert, unbenutzte Bytes = 0x00 |
| Byte 38-45 | Ausgabe Text bei Ausgabe über höchstem Tabellenwert, unbenutzte Bytes = 0x00 |
| Byte 46-77 | Zugewiesene Channels, 2 Byte pro Channel |
| Byte 78-87 | Tabelle Punkt 1 - 16 Bit Wert + 8 Byte Text, unbenutzte Bytes = 0x00 |
| Byte 88-97 | Tabelle Punkt 2 - 16 Bit Wert + 8 Byte Text, unbenutzte Bytes = 0x00 |
| Byte 98-107 | Tabelle Punkt 3 - 16 Bit Wert + 8 Byte Text, unbenutzte Bytes = 0x00 |
| Byte 108-117 | Tabelle Punkt 4 - 16 Bit Wert + 8 Byte Text, unbenutzte Bytes = 0x00 |
| Byte 118-127 | Tabelle Punkt 5 - 16 Bit Wert + 8 Byte Text, unbenutzte Bytes = 0x00 |