last updated: 25-10-20

dmxrdm
ENGINEERING • DEVELOPMENT

BEDIENUNGSANLEITUNG
DMX RDM

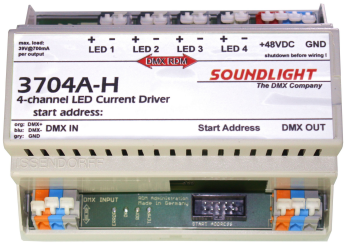
DMXRDM *Engineering / Deveelopment* Bennigser Str. 1 D-30974 Wennigsen

Vielen Dank, daß Sie sich für DMXRDM entschieden haben.

Fast alle **DMXRDM** Geräte unterstützen schon DMX RDM, und auch einige Mitbewerber haben schon ein paar zaghafte Versuche gemacht, ein paar RDM Funktionen in ihren Geräten zu implementieren. RDM ist im Kommen!

Wer einmal mit RDM gearbeitet hat, wird es bald nicht mehr missen wollen. Allerdings ist dazu auch ein guter DMX RDM Controller notwenig. Mehr dazu später!

Anwendungen



Erstmal zu den Vorzügen von DMX RDM: die Kommunikation erfolgt bidirektional, d.h., das angeschlossene Gerät empfängt nicht nur Informationen vom Controller, sondern kann auch welche zurückgeben. So kann man Abfragen machen und Einstellungen per ***Dialog*** vornehmen: man unterhält sich! Durch die Rückmeldung haben Sie auch jederzeit die Gewissheit, dass Ihre Einstellungen korrekt übernommen wurden.

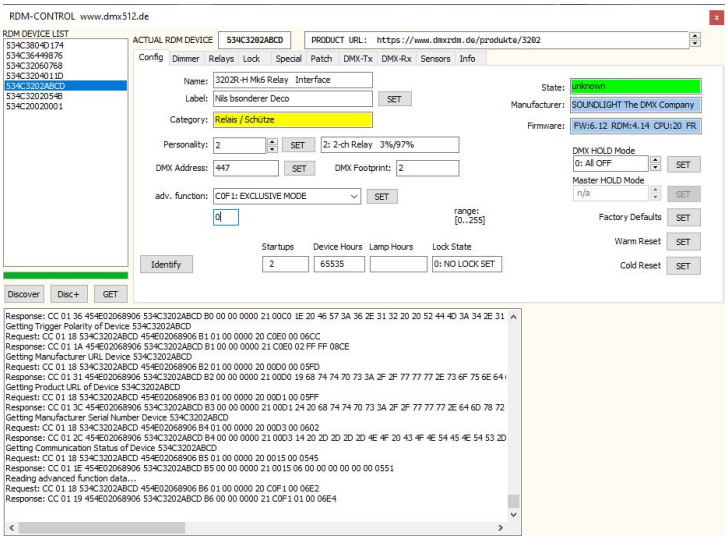
Es gibt derzeit bereits mehrere RDM Standards. Zum Grundstandard ANSI E1.20 kommen mehrere Ergänzungsdokumente der Reihe E1.37, die viele zusätzliche standardisierte RDM Kommandos beschreiben. Viele kennen diese Normen noch gar nicht, sie sind aber „das Salz in der Suppe“ und erweitern die Möglichkeiten beträchtlich. **DMXRDM**-Module verwenden zahlreiche Kommandos aus den E1.37 Ergänzungsstandards. Alle DMX RDM Normen sind als ANSI Standards verfügbar und können über das American National Standards Institute (ANSI, www.ansi.org) bezogen werden.

Hinzu kommen zahlreiche herstellerspezifische Kommandos, mit denen spezielle Eigenschaften der Decoder verwaltet werden können. Herstellerspezifische Kommandos sind in der Grundnorm E1.20 aufgeführt; diese beschreibt aber lediglich „einfache“ herstellerspezifische Kommandos. Das sind solche, die nur einen Parameter verwalten können (SMSC: Simple Manufacturer Specific Commands). Wir verwenden auch CMSC: Complex Manufacturer Specific Commands, nämlich solche, die auch mehrere Parameter (oder texte o.ä.) verwalten können. Alle von uns verwendeten Kommandos sind vollständig dokumentiert und auf unserer RDM Website gelistet.

RDM Website

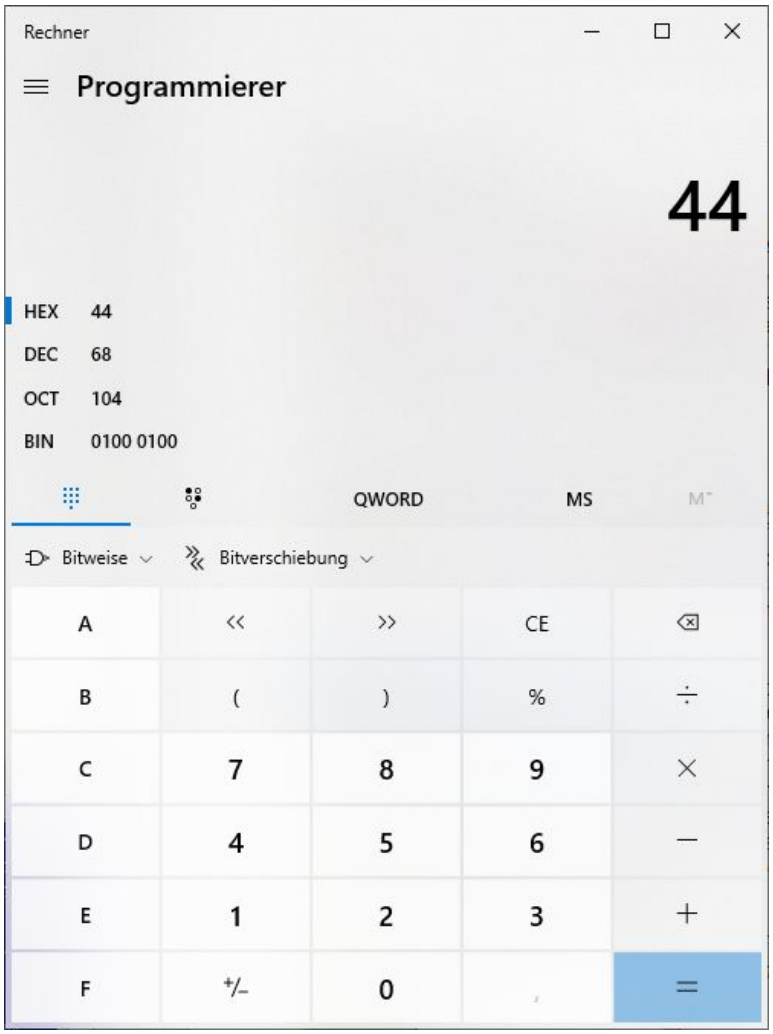
Alle wichtigen Informationen haben wir auf unserer DMX Website zusammengetragen. Sie erreichen sie im Internet unter: ***https://www.dmx512.de***. Dort finden Sie auch zahlreiche Anwendungsbeispiele, und auch die Umsetzung von RDM-Befehlen auf verschiedenen Contollern wird gezeigt. Zudem laden Sie dort den kostenlosen RDM Controller **RDMCONTROL,** der eine Vielzahl wichtiger RDM-Funktionen bietet und optimal zur Konfiguration unserer Geräte ausgelegt ist. Mehr dazu unter:

www.dmx512.de/rdmcontrol

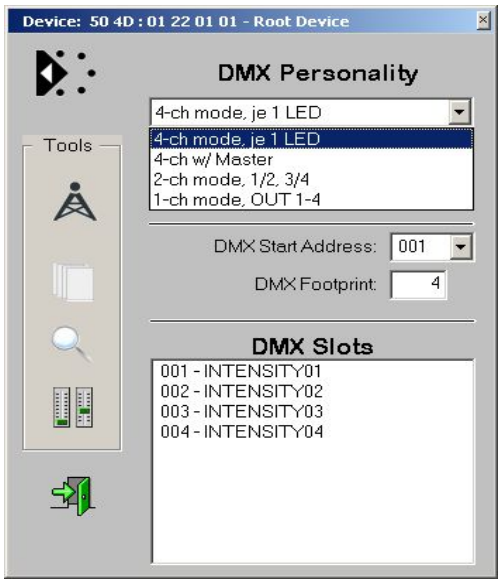


Datenformate

RDM-Befehle selbst verwenden hexadezimale Notation, d.h., alle Kommandos werden mit Hexadezimalzahlen (0...9,A,B,C,D,E,F) beschrieben. Jedes Byte besteht aus zwei hexadezimalen Ziffern, z.B. C2, 8B, 44 oder 01, und damit man diese eindeutig von der dezimalen Schreibweise unterscheiden kann, werden Hexadezimalzahlen durch ein vorangestelltes „$“ oder ein angehängtes „h“ bzw. „hex“ gekennzeichnet. So ist also „44“ eindeutig eine Dezimalzahl (vierundvierzig), während „$44“ oder „44h“ oder „44*hex*“ eine Hexadezimalzahl ausweist, die -siehe rechts- der dezimalen „68“ (4*16+4) enstspricht.



Es gibt jedoch auch Controller, die stattdessen dezimale Eingaben erwarten. Hier muß der Operator gegebenenfalls passend umcodieren; der Windows Taschenrechner (siehe rechts) kann auf Hexadezimale Eingabe umgestellt werden und ist dabei als Eingabehilfe recht nützlich.
Bitte beachten Sie auch die zu verwendende Formatierung: keine Trennung einzelner Bytes, Trennung durch Spaces, durch Kommata oder durch Semikolon sind möglich. Das hängt allein vom jeweilig verwendeten Controller ab.



Anwenderfreundliche RDM Controller, wie der von uns verwendete GET/SET Controller, stellen für Standardkommandos wenn immer möglich spezielle Anwender-Eingabemasken bereit und abstrahieren so von der Kommandosyntax und den Kommandodaten selbst. Da das aber noch keineswegs selbstverständlich ist, geben wir stets das Kommandoformat, den Datenbereich und die entsprechenden Datenwerte an.

Der GET/SET Controller verfügt über zahlreiche Eingabemasken für anwenderfreundliche Bedienung. So kann z.B. die Startadresse direkt dezimal eingeben werden.

GET und SET

DMX RDM verwendet grundsätzlich drei verschiedene Kommandoklassen:

1. **DISCOVERY** zum automatischen Auffinden und Erkennen angeschlossener Geräte

2. **GET** zum Abfragen von Parametern und Einstellungen

3. **SET** zum Setzen von Parametern und Einstellungen

Daher kommt dann auch die Bezeichnung *GET/SET* für unseren Controller. Eigentlich logisch, oder? Daß der natürlich aber auch eine Discovery macht, ist klar.

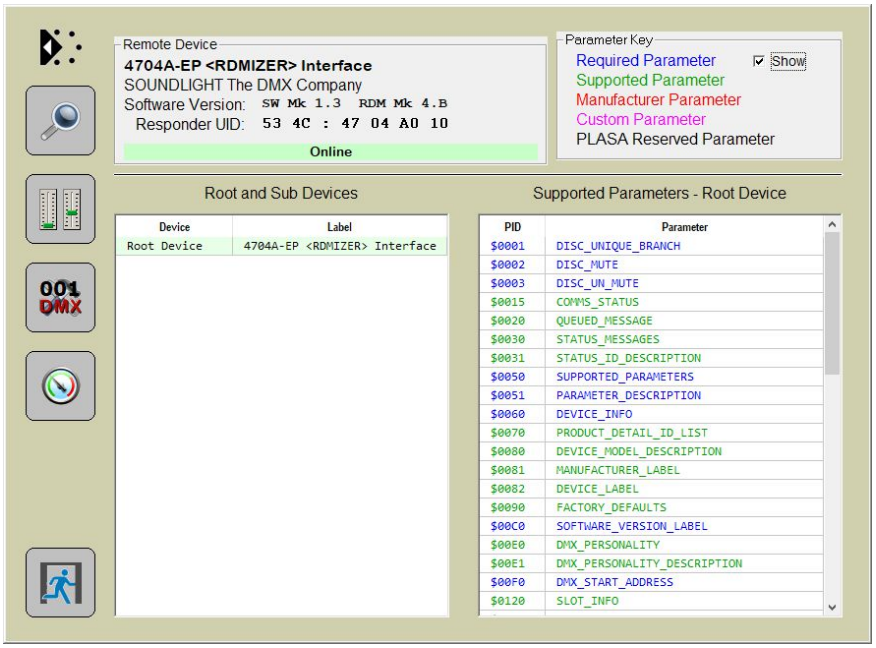
Smarter Controller?

Ausser einer hohen Merkfähigkeit muß ein RDM Controller eigentlich nicht viel können. Er kennt keine Daten, keine Einstellungen. Alle Informationen bezieht er jederzeit von den angeschlossenen Geräten: die wissen, wie sie heißen, was sie können, und wie sie eingestellt sind. Alle Daten sind in den angeschlossenen „Respondern“, wie diese auf englisch auch genannt werden, gespeichert. Und die teilen sie auf Anfrage natürlich jederzeit gerne mit.

Und so geht’s:

1. DISCOVERY

Jeder Responder hat eine (weltweit) eigene Kennung, auch UID (Unique Identifier) genannt. Der Controller sendet ein Kommando, das alle angeschlossenen Responder in einen Antwortmodus versetzt und prüft dann alle UIDs daraufhin ab, ob sich unter dieser Adresse ein Teilnehmer meldet. Ist das der Fall, wird der Teilnehmer für das weitere Antwortverfahren stummgeschaltet und vom Controller in die Anwesenheitsliste eingetragen.



Die UID des SOUNDLIGHT RDMIZER 4704A-EP: 53 4C 47 04 A0 10 wird beim Auslesen über den GET/SET Controller eindeutig dargestellt.

UIDs bestehen aus 6 Hexadezimalzahlen. Die ersten beiden Stellen geben den Hersteller an; die Herstellerkennung wird einheitlich vom PLASA/ESTA Normenausschuss festgelegt. Die Liste ist dort einsehbar. **DMXRDM** hat die Herstellerkennung „SL“, dargestellt durch die Hexadezimalwerte „53 4C“ (siehe: *http://tsp.esta.org/tsp/working_groups/CP/mfctrIDs.php*). Danach vergeben wir die Gerätekennung und die Seriennummer.
Die UID „53 4C 36 03 12 34“ heisst also: dies ist ein Gerät von **SOUNDLIGHT** (53 4C), Gerätetype ist „36 03“ und die Seriennummer ist „12 34“.

2. GET-Kommandos

Jeder Responder wird nun vom Controller gezielt unter seiner ermittelten UID angesprochen und per GET-Kommandos nach seinen Eigenschaften abgefragt (z.B. der Startadresse). Dafür wird für jeden Befehl eine spezielle Befehlsnummer, die sogenannte PID, benutzt. Der Befehl zum Abfragen (bzw. Setzen) der Startadresse hat beispielsweise die Befehlsnummer (PID) 00F0. Für den Controller ist nur die PID wichtig, der Anwender hat damit später nichts zu tun- für ihn heisst der Befehl natürlich „Startadresse“. Die wichtigsten Erstabfragen sind die Gerätetype, der Gerätename, die DMX Startadresse und die eingestellte DMX Personality (Betriebsart).

3. SET-Kommandos

Möchte der Controller andere Werte setzen als im angeschlossenen Responder eingestellt, so kann er dies durch ein SET-Kommando tun. Für das SET-Kommando wird die gleiche PID (Befehlsnummer) verwendet, diesmal jedoch mit einer SET-Kennung versehen. Zusätzlich werden die einzustellenden Parameter übergeben. Die erforderlichen Parameter (Typ, Reihenfolge und Anzahl) können für die in der Norm ANSI E1.20 (siehe: http://www.ansi.org) festgelegten Kommandos dem Normblatt, und für die herstellerspezifischen Kommandos der dem jeweiligen Kommando (SMSC) zugeordneten Parameterliste, die im betreffenden Interface abgefragt werden kann, entnommen werden.

PIDs und Befehle

PIDs sind in vier Gruppen eingeteilt:

1. Standard-Befehle, die jedes RDM-Gerät können muss

Das sind alle PIDs, die Funktionen umfassen, die für die Funktionalität des RDM-Verfahrens erforderlich sind. Es ist also der Mindestbefehlssatz, über den ein Responder verfügen muss, um überhaupt zu funktionieren. Dazu gehören alle Befehle zum Durchführen der Discovery, aber auch die sogenannte DEVICE INFO, die Liste der SUPPORTED PARAMETERS, und die STARTADRESSE. Responder, die kaum mehr können als das, sind also „unterste Schublade“ - und davon gibt es genug!

2. Standard-Befehle

Das sind alle PIDs, die in den aktuellen RDM Normen (siehe oben) beschrieben sind. Die jeweils erforderliche Befehlssyntax und die Liste der ausgetauschten Parameter sind standardmässig definiert, und intelligente Controller können daher für jeden dieser Befehle eine angepasste User-Maske bereithalten. Das vereinfacht die Eingabe für den Anwender beträchtlich (siehe GET/SET Controller!) Es gibt aber auch zahlreiche Controller, die nur die Eingabe der jeweiligen Parameterliste erwarten (und diese vom Anwender zusammengestellt werden muss). Daher geben wir bei unseren Beschreibungen stets auch die Formate, die Anzahl und die Wertebereiche der jeweiligen Parameter an.
Der PID-Bereich der Standardbefehle reicht von 0001 bis 7FFF.

3. Manufacturer Specific Commands

PIDs im Bereich von 8000 bis FFDF sind für herstellerspezifische Kommandos reserviert. Herstellerspezifische Kommandos verwalten Zusatzfunktionen, die in den Standards nicht definiert sind. Alle Produkte eines Herstellers müssen unter der gleichen PID jeweils dieselbe Funktion aufrufen; das bedeutet aber auch: unter derselben PID kann das Modell einen anderen Herstellers eine andere Funktion aufweisen! Das Verarbeiten einer MSC erfordert also stets auch die Prüfung der zugehörigen Hersteller-Kennung (UID).

3.1. Simple Manufacturer Specific Commands (SMSC)

Das sind Befehle, die stets nur eine feste Anzahl von Bytes bearbeiten - einfachster Weise eben nur ein Byte. Die Anzahl und der erlaubte Wertebereich sind on den Parameterdaten, die für das jeweilige Kommando abgefragt werden können, enthalten.

3.2. Complex Manufacturer Specific Commands (CMSC)

Wir verwenden zahlreiche PIDs, die als CMSC einzuordnen sind. Diese Kommandos erlauben sowohl für Eingabe als auch für die Ausgabe eine variable Parameterdatenlänge. Die Korrektheit eines Kommandos wird dabei einfach über die im RDM Befehl angegebene Datensatzlänge und die jeweilige Befehls-Checksum verifiziert. Somit ergibt sich die variable Parameterdatenlänge automatisch. Im RDM Standard E1.20 sind nur SMSC spezifiziert, CMSC wurden vom Standards-Komitee bisher noch nicht abschließend definiert (auf der „to do“ Liste)

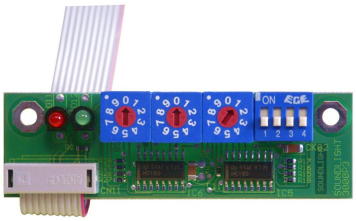
4. PLASA / ESTA Kommandos

Der PID-Bereich von FFE0 bis FFFF ist für das Standards-Komitee reserviert und dient der Entwicklung neuer Standard-Funktionen (z.B. Prototyping). In den Verkauf gehende Geräte dürfen jedoch keine PIDs aus diesem Wertebereich enthalten.

RDM Programmierung

Die Einstellung der Funktionsparameter kann bei unseren Decodern wahlweise über DMX RDM oder aber über ein anschließbares Adressboard erfolgen (das kann, muss aber nicht angeschlossen bleiben, weil alle eingestellten Parameter in den Speicher des Grundgerätes übernommen werden).

Als Adressboards stehen verschiedene Versionen zur Verfügung:



Startadressboard 3000P:
Die Einstellung der Startadresse erfolgt über Drehcodierschalter, die Einstellung der Personality und des HOLD-Modus über DIP-Schalter. Der Zustand des Decoders wird über Anzeige-LED signalisiert.

Startadressboards 3006P / 3008P:
Alle Einstellungen erfolgen über einen Dreh-Encoder und werden auf einem LC-Display ausgegeben. Das Adressboard kann vorhandene Einstellungen aus dem Decoder auslesen.

Startadressboards 3006P-SD:
Alle Einstellungen erfolgen über einen Dreh-Encoder und werden auf einem LC-Display ausgegeben. Das Adressboard kann SubDevices konfigurieren (mehrere unterschiedliche Startadressen in einem Responder) und vorhandene Einstellungen aus dem Decoder auslesen. Nur für Decoder, die dies Adressboard unterstützen!

TIPP:
Da Schalter stets auf einer festen Einstellung stehen, die sich von einer über DMX RDM getroffenen Adress-Einstellung unterscheiden kann, werden die mechanischen Schalter deaktiviert, sobald die Programmierung z.B. einer anderen Startadresse oder Personality über RDM erfolgt. Das wird durch die Signalisierung auf dem Decoder angezeigt; die gelbe RDM-LED leuchtet ab dann konstant. Man kann die Schalter einfach wieder aktivieren, indem man die Hunderterstelle kurzzeitig auf „9“ stellt - also eine fiktive Adresse zwischen 900 und 999 wählt. Die Schalter werden dann wieder freigegeben.

WICHTIGER HINWEIS: Alle Einstellungen der Startadresse und der Geräteeigenschaften ("DMX Personalities") können stets über DMX RDM erfolgen. Alternativ läßt sich die Startadresse, die Personality und der HOLD-Modus mit einem Adressboard 3000P, 3003P, 3005P, 3006P oder 3008P setzen. Adressboards sind als optionales Zubehör separat erhältlich.

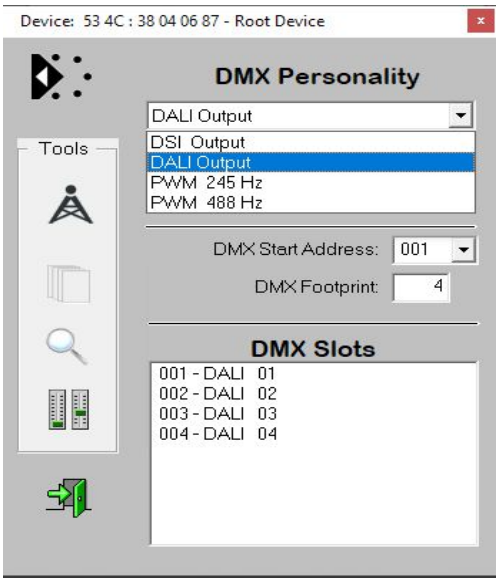


Bild: Personality- und Startadress-Menü beim RDM Controller „JESE GET/SET“

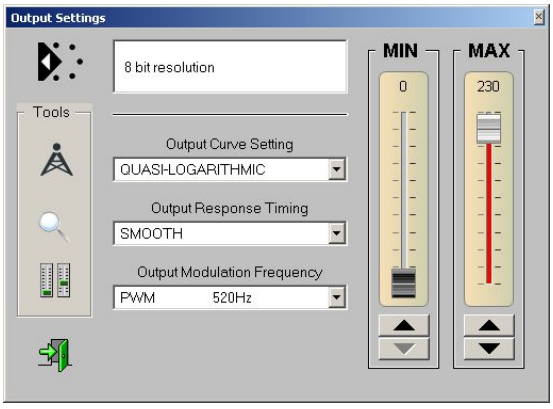


Bild: Sonderfunktionen lassen sich per RDM komfortabel über spezielle Menüs bedienen. Hier die Ausgabe-Konfiguration beim RDM Controller „JESE GET/SET“

RDM Controller

Lassen Sis ich nicht verwirren: viele Geräte, die als „RDM Controller“ angeboten werden, sind tatsächlich „RDM Responder“, also Geräte, die RDM Befehle empfangen und umsetzen können (wie z.B. LED Dimmer). Daher zunächst mal die korrekte Begriffsbestimmung:

RDM CONTROLLER: Ein Gerät, das DMX RDM Befehle ausgibt und RDM Antworten von angeschlossenen Teilnehmern empfangen kann. Also typischerweise ein Lichtsteuerpult, ein Laptop oder PC, oder ein Handgerät zum Konfigurieren der angeschlossenen RDM Teilnehmer.

RDM RESPONDER: Ein DMX Empfänger, der auch RDM Kommandos empfangen und umsetzen kann. Jeder RDM Responder hat eine weltweit einzigartige Kennung (UID, **U**nique **Id**entifier), über die er identifiziert, angesprochen und verwaltet werden kann.

Zur Konfiguration von RDM Respondern empfehlen wir Ihnen folgende RDM Ciontroller:

JESE GET/SET CONTROLLER **www.jese.co.uk**

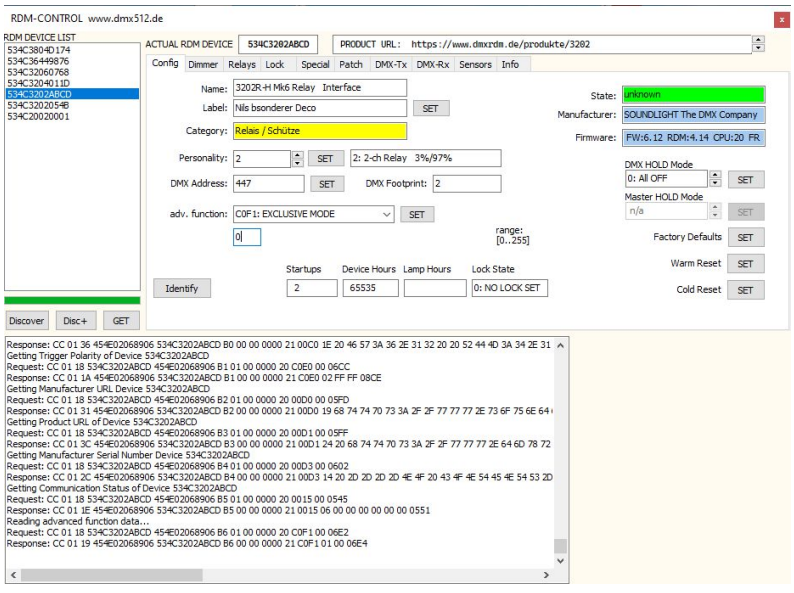
Kommt als Bundle einer Windows Software mit einem USB Interface USBRDM-TRI, das zugleich auch einen DMX Dateneingang zur Verfügung stellt. Dadurch ist es möglich, RDM Telegramme auf eine bestehnde DMX Verbindung „aufzumodulieren“ und somit auch nicht RDM-fähige Anlagen mit RDM zu erweitern.
Die Software wird fortlaufend aktualisiert und kann über ein Abo stets auf neuestem Stand gehalten werden.
Zahlreiche Beispiele in diesem Manual sind mit der GET/SET Software erstellt.
GET/SET nutzt hexadezimale oder Texteingaben und stellt universelle und viele spezielle Eingabemasken zur Verfügung, die das Arbeiten leicht und übersichtlich machen.

City Theatrical DMXcat **www.citytheatrical.com**

Kommt als Interface, das per Blutooth an ein Mobiltelefon gekoppelt wird. Die benötigte App steht kostenlos im AppStore zur Verfügung.
DMXcat nutzt dezimale Notation und stellt Sondermasken für einige Standardeingaben bereit.

DMX512 RDMCONTROL **www.dmx512.de**

Die kostenlose Windows - Freeware kann auf PC oder Laptop genutzt werden und benötigt als Ausgabeinterface ein beliebiges RDM-kompatibles USB Interface, das der OPEN RDM oder ENTTEC Spezifikation antspricht und auf dem FTDI Chip basiert (Industriestandard). Viele Hersteller (z.B. Enttec, Philips Lighting, Lumenpulse etc.) bieten solche Interfaces an.



Ein preisgünstiges Interface kann über DMXRDM (www.dmxrdm.de) bezogen werden.

RDMCONTROL nutzt dezimale Notation und bietet zahlreiche Sondermasken für Spezialkommandos, einen Log der RDM Kommunikation, einen grafischen Kurveneditor u.v.m.

DMX RDM

Die nachfolgenden Kommandos beziehen sich auf den Draft Standard ANSI E1.37 "Additional Commands for RDM". Dieser Standard wurde erst kürzlich ratifiziert. DMX-RDM Controller können also gegebenenfalls noch nicht über die zugehörigen Funktionsnamen verfügen. Über geeignete RDM Controller (z.B. Enttec RDM Controller, GET/SET RDM Controller) lassen sich die zugehörigen PIDs über ihre Funktionsnummer aber bereits aufrufen.

*Eine Beschreibung der Funktionen findet sich auf **www.DMXRDM.de**.*

Startadresse / Personality

Die Grundeinstellungen umfassen die Auswahl der DMX Personality sowie die Einstellung der Startadresse. Dafür sollten alle Controller entsprechende Masken vorhalten. Beim GET/Set Cotroller rufen beide Funktionen dieselbe Maske auf, und die Einstellungen können jeweils aus einer Drop-Down-Liste ausgewählt werden.

Die Einstellung der Startadresse ist auf den für DMX gültigen Wertebereich 001-512 begrenzt. Für den oberen Wert rechnet der JESE Controller automatisch die vom jeweiligen Responder benötigte Kanalzahl ab; für ein 4-Kanal Gerät ist damit die höchste einstellbare Startadresse 509.

Das kann bei anderen Controllern anders gehandhabt werden; grundsätzlich wäre auch eine Einstellung oberhalb dieser Adresse möglich. Dann würden aber nicht mehr alle Ausgänge mit Daten versorgt und somit disfunktional.

Spezielle Kommandos

Diese Kommandos umfassen sowohl Kommandos aus dem RDM Ergänzungsstandard als auch herstellerspezifische Kommandos, die in **DMXRDM** Produkten verwendet werden. Dies Manual erhebt keinen Anspruch auf Vollständigkeit, und gibt nur die grundlegenden Informationen zur Verwendung der vorgestellten PIDs wieder.

Für eigene Implementationen legen Sie bitte die relevanten RDM Standards ANSI E1.20 und ANSI E1.37 zugrunde.

Die PIDs sind, soweit möglich, numerisch geordnet. Bisweilen sind für einen Befehl zwei PIDs angegeben: das gilt insbesondere für neue Kommandos aus dem E1.37 Standard, für den bisher sogar mehrere Teile vorliegen (E1.37-1, E1.37-2, E1.37.5, E1.37-7). Da viele Controller die offiziellen Befehle aus dieser Norm noch nicht kennen, würden sie einen Aufruf mit der festgelegten E1.37 Standard-PID als „unbekannt“ abweisen. Beim Aufruf unter einer PID aus dem herstellerspezifischen Bereich wird aber meistens eine Standard-Eingabemaske angeboten. Da Sie die Funktion über beide PIDs erreichen können, ist es mit diesem Trick dann dennoch möglich, das Kommando zu verarbeiten. Nicht alle Responder unterstützen aber diese Doppel-PIDs.

PID 8081: DMX HOLD INPUT 1	**Verhalten bei DMX Signalausfall DMX Eingang 1**
PID 8082: DMX HOLD INPUT 2	**Verhalten bei DMX Signalausfall DMX Eingang 2**
PID 8083: DMX HOLD INPUT 3	**Verhalten bei DMX Signalausfall DMX Eingang 3**
PID 8084: DMX HOLD INPUT 4	**Verhalten bei DMX Signalausfall DMX Eingang 4**
PID 8085: DMX HOLD INPUT 5	**Verhalten bei DMX Signalausfall DMX Eingang 5**
PID 8086: DMX HOLD INPUT 6	**Verhalten bei DMX Signalausfall DMX Eingang 6**
PID 80F1: DMX HOLD MODE	**Verhalten bei DMX Signalausfall DMX Eingang allgemein**
PID 80F2: MASTER HOLD MODE	**Verhalten bei DMX Signalausfall Master-Eingang**

Der DMX HOLD MODE bestimmt das Verhalten des Gerätes bei Verlust des Steuersignals (DMX Signalausfall). Dabei sind grundsätzlich drei Möglichkeiten gegeben, die durch die angegebenen Parameter gesetzt werden können:

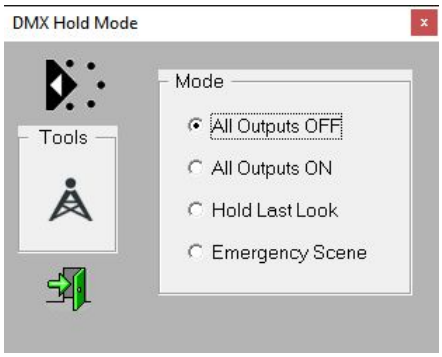
PARAMETER: 1 Byte (HOLD MODUS)

GET:	Aufruf ohne Parameter
Rückgabe:	Parameter: 1 Byte (HOLD-Modus)
SET:	Aufruf mit Parameter: 1 Byte (HOLD-Modus)
Rückgabe:	Status

Parameter:

Hex	Dez	Funktion
$00	0	Alle Ausgänge fahren auf 0% (aus)
$01	1	Alle Ausgänge fahren auf 100% (ein)
$02	2	Alle Ausgänge behalten den letztgültigen Wert bei „Keep Last Look“
$03	3	Eine voreingestellte Szene wird aufgerufen*

**=nicht bei allen Modellen verfügbar*



Die Einstellung des HOLD-Modus erfolgt beim Startadressboard 3000P durch die DIP-Schalter 1 und 2, bei Startadressboard 3003P durch die virtuellen DIP-Schalter S1 und S2 und bei den LCD-Adressboards sehr einfach durch ein Menü.

*Zur Einstellung des HOLD-Modus steht beim GET/SET Controller für **DMXRDM** Geräte eine komfortable Eingabemaske zur Verfügung (siehe nebenstehend).*

PID 80FA: HOLD LEVEL Ausgabewert bei DMX Datenausfall

Bei Datenausfall können Ausgänge wahlweise ausgeschaltet (Ausgabewert = 0) oder eingeschaltet werden (Ausgabewert = 255). Wenn ein HOLD LEVEL unterstützt wird, dann kann der Ausgabewet mit diesem Kommando auf jeden beliebigen Wert von 0...255 eingestellt werden.

PARAMETER: 1 Byte (HOLD LEVEL)

GET: Aufruf ohne Parameter
Rückgabe: Parameter: 1 Byte (HOLD LEVEL)

SET: Aufruf mit Parameter: 1 Byte (HOLD LEVEL)
Rückgabe: Status

Parameter HOLD LEVEL

Hex	Dez	Funktion
$00	0	Alle Ausgänge fahren auf 0% (aus)
...bis	...	
$FF	255	Alle Ausgänge fahren auf 100% (ein)

HINWEIS:
Falls Schaltschwellen oder Ausgabebegrenzungen (MIN LEVEL, MAX LEVEL) gesetzt sind, beachten Sie bitte, dass diese vorrangig sind!.

Ein HOLD LEVEL kann auch mit dem Standard Kommando PID 0141: DMX FAIL MODE gesetzt werden

PID 80E3: MODE_16BIT_RECEIVE 16Bit DMX Datenempfang

DMX512 ist generell ein 8 Bit Datenformat, Die Übertragung von 16 Bit Daten wird als aufeinanderfolgende Übertragung zweier Bytes angenommen, wobei diese aus HighByte und LowByte (oder LowByte, HighByte) bestehen.

Der Empfänger muss dem Rechnung tragen und dafür sorgen, dass diese beiden Bytes als ein gemeinsamer Datenblock behandelt und weiterverarbeitet werden.

Damit das sichergestellt ist, muss das erste Byte zwingend auf einer **UNGERADEN** Adresse (z.B. 1, 3 oder 7), das folgende Byte zwingend auf einer **GERADEN** Adresse (dann z.B. 2, 4 oder 8) übertragen werden. Nach dem Empfang werden beide Bytes nach dem Empfang des jweils kompletten Paares zur Auswertung gegeben.

HINWEISE:
Unvollständig empfangene Pakete werden verworfen.

Auf einer falschen Adresse startende 16Bit Pakete generieren Flicker, da dadurch Daten von zwei verschiedenen Slots zusammengewürfelt werden.

Im 16Bit Empfangsmodus werden auch 8Bit Daten empfangen und ausgewertet. Die Anzahl der empfangenen Daten muss jedoch gerade sein, oder sie wird auf die letzte gerade Anzahl gekürzt.

PARAMETER: 1 Byte (MODE_16BIT_RECEIVE)

GET:	Aufruf ohne Parameter
Rückgabe:	Parameter: 1 Byte (MODE_16BIT_RECEIVE)
SET:	Aufruf mit Parameter: 1 Byte (MODE_16BIT_RECEIVE)
Rückgabe:	Status

Parameter MODE_16BIT_RECEIVE

Hex	Dez	Funktion
$00	0	aus: Standard 8 Bit empfang aktiviert
$FF	255	ein: 16Bit empfang aktiviert

PID 0141: DMX FAIL MODE
PID 8301: DMX FAIL MODE

Verhalten bei DMX Signalausfall

Der DMX FAIL MODE ist eine erweiterte Eingabemöglichkeit für das Verhalten des Gerätes bei Signalausfall. Dabei werden mehrere Parameter gesetzt - nicht alle Parameter, die dieser Befehl vorsieht, werden vom Gerät benötigt. Insofern ist dies Kommando recht „oversized“ und nur in seltenen Fällen verfügbar. Die Abbildung der Einstellungen des HOLD-MODUS (PID 80F1) auf die Einstellungen des DMX FAIL MODE ist wie folgt:

PARAMETER: 7 Bytes, davon:
Byte 1,2 = Szenennummer (16 Bit)
Byte 3,4 = Verzögerung nach Signalausfall in 1/10 Sekunden (16 Bit)
Byte 5,6 = Haltezeit in 1/10 Sekunden (16 Bit)
Byte 7 = Pegel (HOLD LEVEL)

GET: Aufruf ohne Parameter
Rückgabe: Parameter: 7 Byte (Fail-Modus)

SET: Aufruf mit Parameter: 7 Byte (Fail-Modus)
Rückgabe: Status

Die Zuordnung ist wie folgt:

HOLD MODE	Funktion	FAIL MODE
00	alles aus	00 00 00 00 FF FF 00
01	alles ein	00 00 00 00 FF FF xx
02	"last look"	00 00 FF FF FF FF 00

Während die Einstellung des Signalausfallverhaltens über die DIP-Schalter und über den HOLD MODE lediglich das Setzen eines EIN/AUS-Wertes ermöglicht, kann über den FAILMODE Befehl jedoch auch der Ausfallpegel definiert werden. Dazu wird der Parameter "xx" mit dem gewünschten Ausfallwert belegt, hier also "00" (0) für "AUS", "FF" (255) für "EIN".

Alle Werte sind als Hexadezimalwerte einzugeben!

PID 0640: LOCK PIN
PID 8330: LOCK PIN

Eingabe einer PIN zur Verriegelung

Mit Auslieferung ist das Gerät entriegelt und die Start-PIN ist 0000 (Hex 0000). Die Funktion erlaubt nur eine SET-Eingabe, keine Auslesung über GET. Die PID 0640 erlaubt die Konfiguration über User-Maske, die PID 8330 hingegen über Eingabe im Hexadezimal-Modus (bzw. Dezimalmodus, je nach Controller).

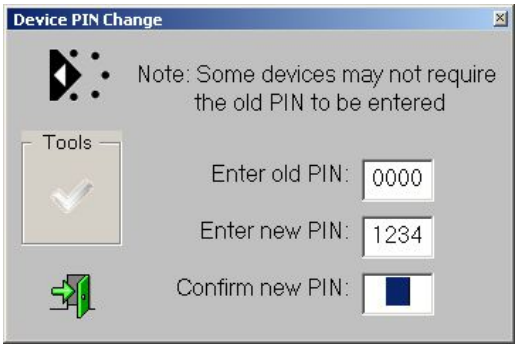
Um eine neue PIN einzugeben, geben Sie die neue PIN, gefolgt von der alten PIN, ein.
Beispiel: neue Pin 0220, alte Pin 1836: Eingabe 02201836.
PINS sind im Bereich von 0000(dez) bis 9999(dez) erlaubt, bei Auslieferung ist die Start-Pin 0000(dez) gesetzt.

WICHTIG: Sofern der Controller (das ist z.B. für den Enttec Controller der Fall!) eine hexadezimale Eingabe erwartet, müssen die Werte im Hex-Format eingegeben werden (Sie können sie z.B. mit dem Windows Calculator im Programmer's Mode einfach umrechnen, indem Sie das Zahlensystem von Dez auf Hex umschalten). Die Eingabe wäre dann 00DC072C. Wird die Eingabe akzeptiert, gilt ab sofort die neue PIN.

PARAMETER: 4 Byte
Byte 1,2: neue PIN (16 Bit)
Byte 3,4: alte PIN (16 Bit))

GET: kein GET-Aufruf möglich;
PIN kann nicht abgefragt werden!

SET: Aufruf mit Parameter: 4 Byte (PIN)
Rückgabe: Status



Eingabe einer PIN beim GET/SET Controller

WICHTIG: Merken Sie sich eine neu vergebene PIN gut! Ohne Kenntnis dieser PIN ist sonst ein weiterer Zugriff nicht mehr möglich. Eine Löschung unbekannter PINs ist nur im Werk möglich!

Die mehrfache Eingabe einer falschen PIN führt zu einer Eingabeverriegelung, die nur durch einen RESET wieder zurückgesetzt werden kann.

PID 0641: LOCK STATE **Verriegelungszustand**
PID 8331: LOCK STATE

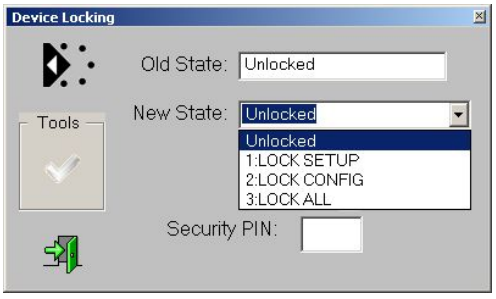
Der augenblickliche Modus kann abgefragt oder neu eingegeben werden.

00= keine Verriegelung
01= Verriegelt

Abfrage ergibt:
<aktuelle Verriegelung> <mögliche Verriegelungen>.
Da die Anzahl der möglichen Verriegelungen stets drei ist, erhalten Sie folgende Ausgabe für:
- nichts verriegelt: 0003
- Setup verriegelt: 0103
- Config verriegelt: 0203
- Alles verriegelt: 0303

Setzen erfordert die Eingabe der aktuelle PIN zur Ausführung des Kommandos: <PIN> <gewünschte Verriegelung>.
Beispiel: aktuelle PIN = 2345(dez), Verriegelung für Setup setzen: Kommando ist: 234501. Bitte beachten Sie auch hier die Verwendung des richtigen Zahlensystems!
Wenn Ihr Controller dei Eingabe von HEX-Zahlen erwartet, müßte das Kommando in diesem Falle lauten: 092901, da 2345(dez)=0929(hex).

Der LOCK STATE bestimmt die Verfügbarkeit verschiedener Einstellungen (siehe Tabelle im Anhang). Eine Änderung ist hier dann nicht mehr möglich, bei Zugriff meldet des Gerät „Write protected“ (schreibgeschützt).



Auswahl des LOCK-Modus beim GET/SET Controller

Eine Änderung dieser Parameter ist nicht mehr möglich, wenn die Verriegelung unter Verwendung einer PIN aktiviert wurde. Eine Deaktivierung ist dann nur noch unter Anwendung dieser PIN möglich; daher ist diese sicher aufzubewahren!

PID 0642: LOCK STATE DESCRIPTION **Ausgabe einer Beschreibung für die Verriegelung**
PID 8332: LOCK STATE DESCRIPTION

Die Eingabe der LOCK STATE-Nummer bei einer GET-Abfrage retourniert die Beschreibung des Status. Ein SET-Kommando ist hier nicht möglich.

PID 1040: IDENTIFY MODE **Identify-Modus**
PID 8340: IDENTIFY MODE

Jedes Gerät muß einen IDENTIFY Befehl verarbeiten können. Mit dem IDENTIFY Befehl muss ein Responder auf sich aufmerksam machen, die Art, wie er das macht, ist indes nicht vorgegeben. Ein Moving-Head wird zumeist mit dem Kopf wackeln, ein LED-Treiber durch EIN- und AUS-Schalten auf sich aufmerksam machen.

Da es äusserst störend sein könnte, wenn man versucht, ein Gerät während der Show zu identifizieren, wurde der IDENTIFY MODE hinzugefügt. Er erlaubt die Auswahl eines „quiet“ Modus, also einer unauffälligen Identifizierung. Üblicherweise werden dann nur Indikator-Anzeigen am Gerät zur Signalisierung verwendet.

Geräte wie z.B. Relaismodule, bei denen eine Identifizierung durch Schalten der Ausgangskontakte auch ausserhalb der Show möglicherweise unvorteilhaft wäre, werden von uns bereits auf den QUIET MODE vorkonfiguriert ausgeliefert. Sie identifizieren sich dann durch Blinken ihrer Status-LEDs.

PARAMETER: 1 Byte (Identify Modus)

GET: Aufruf ohne Parameter
Rückgabe: Parameter: 1 Byte (Identify-Modus)

SET: Aufruf mit Parameter: 1 Byte (Identify-Modus)
Rückgabe: Status

Identify-Modus:

Hex	Dezimal	Funktion
00	0	Quiet Mode: Identify-Ausgabe nur auf den Anzeige-LEDs
FF	255	Loud Mode: Identify über die Ausgänge

Identify Mode Einstellung beim GET/SET Controller

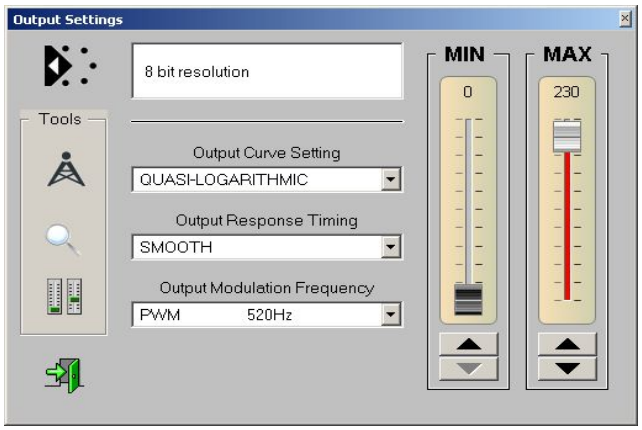
PID 0341: MINIMUM LEVEL
PID 0342: MAXIMUM LEVEL

Pegel-Begrenzung

Der einstellbare Minimal- und Maximalpegel kann durch die Funktionen MINIMUM LEVEL und MAXIMUM LEVEL bestimmt werden. Ziehen Sie den jeweiligen Fader einfach auf den gewünschten Wert. Ist gleichzeitig eine Ansterung vorhanden (oder der HOLD-Modus gesetzt), dann kann die Auswirkung auch optisch geprüft werden.

Die Einstellung erfolgt im DMX Wertebereich 0...255 ($00...$FF). Bitte beachten Sie bei einer Eingabe über numerische Maske auch hier das für den jeweiligen RDM Controller zu verwendende Eingabeformat.

Bei JESE GET/SET Controller kann die Eingabe bequem über Fader erfolgen. Die Übernahme erfolgt automatsich nach einigen Sekunden oder bei Verlassen der Maske.



Einstellen der Pegelbegrenzung beim GET/SET Controller ist sehr einfach.

Beim Aufrufen des Kommandos mit numerischer Eingabemaske ergibt sich:

GET:	Aufruf ohne Parameter	
Rückgabe:	5 Bytes, davon:	
	Byte 1,2:	Minimum-Level aufwärts, 16 Bit
	Byte 3,4:	Minimum-level abwärts, 16 Bit
	Byte 5:	„On“ Below Minimum

SET:	Aufruf mit 5 Bytes Parameter (siehe oben)
Rückgabe:	Status

„On Below Minimum“ ist entweder 0 (Ausgang ist Null wenn das Ansteuersignal unter den Minimalwert fällt) oder 1 (Ausgang bleibt auf Minimum Level wenn as Ansteuersignal unter den Minimalwert fällt).

Die Pegelwerte werden stets als 16-Bit-Werte angegeben. Wenn der Pegelbereich des Gerätes kleiner als 16 Bit ist, werden nur die oberen n Bit berücksichtigt. Die nutzbare Auflösung muss in der PID DIMMER INFO deklariert werden!

PID 8341: MINMAXMODE MINIMUM-/MAXIMUM-MODUS

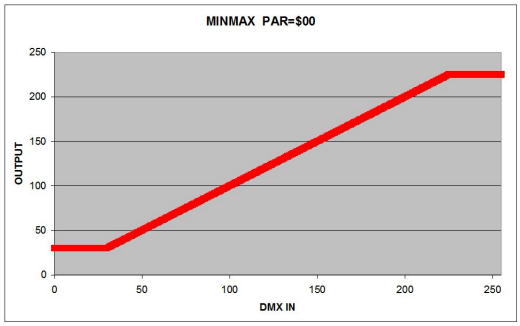
Mit dieser Funktion kann das Verhalten der Pegelbegrenzung bei MINIMUM LEVEL und MAXIMUM LEVEL definiert werden.

Dabei gelten grundsätzlich folgende Einstellungen:

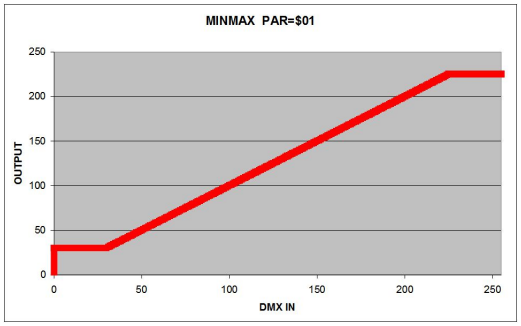
Der Ausgangspegel wird auf den eingestellten Minimal- und Maximalpegel begrenzt. Die Begrenzung tritt in Kraft, sobald der eingestellte Pegel durch den DMX Ansteuerwert erreicht und überschritten (bzw. unterschritten) wird. Danach ändert sich der Ausgangspegel nicht mehr.

Mit dem Befehl MINMAXMODE kann man statt einer Begrenzung auch a eine Skalierung des Ausganges einrichten, und entscheiden, ob derZustand „aus“ (Pegel Null) durchgereicht werden soll oder nicht.

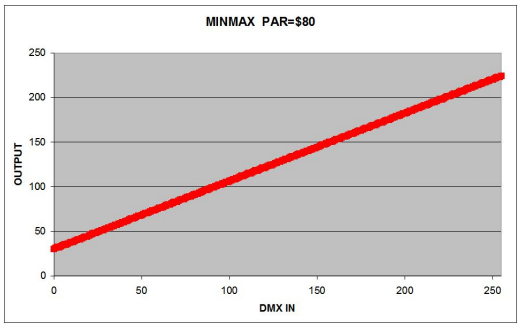
PARAMETER: 1 Byte



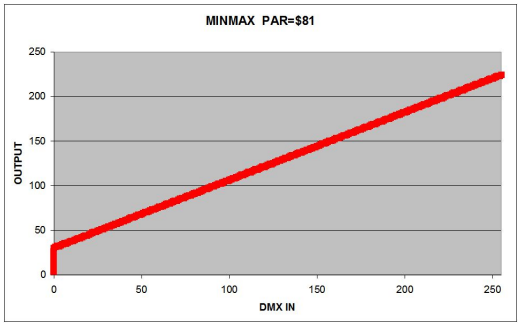
Parameter = $00 (0 dez):
Begrenzung als Standardfunktion



Parameter = $01 (1 dez):
Nullpegel geht durch: Bei Eingangspegel Null wird auch Ausgangspegel Null ausgegeben. Darüber springt der Ausgang sofort auf MIN LEVEL.



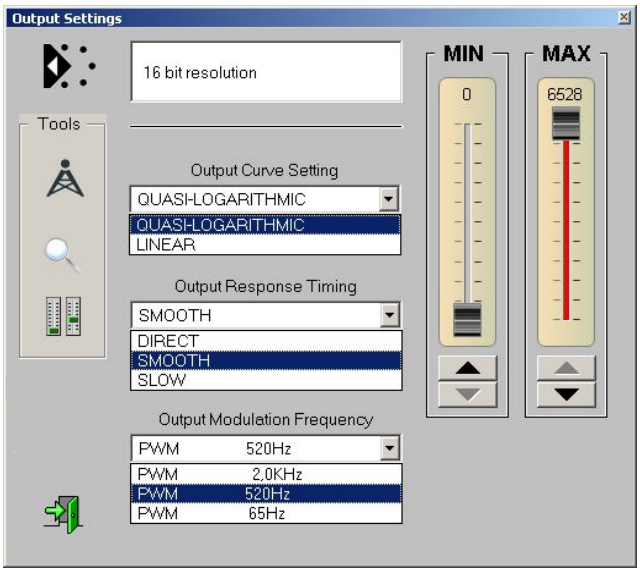
Parameter = $80 (128 dez):
Die Begrenzung wird außer Kraft gesetzt und durch einen kontiniuerlichen Fade vom MIN LEVEL bis zum MAX LEVEL ersetzt. Dadurch wird der Fade-Bereich weiter.



Parameter = $81 (129 dez):
Die Parameter können auch kombiniert werden.

PID 0343: CURVE — AUSGANGSKURVE

Mit der PID 0343 kann die Ausgangskennline bestimmt werden.



Eingabe der Output Curve beim GET/SET Controller. Wählen Sie einfach aus der aufklappbaren Drop-Down Liste aus.

Die voreingestellte Kennline ist hier "QUASI-LOGARITHMISCH" (Kennlinie 1). Das trifft z.B. Für die meisten PWM Decoder zu. In diesem Fall ist die alternative Kennlinie "LINEAR" (Kennlinie 2). Oft steht als weitere Kennlinie (Kennlinie 3) dann noch eine selbst definierbare User Curve zur Verfügung.

Bei Abfrage über ein GET Kommando wird die derzeitige Einstellung ausgegeben:

GET	Aufruf ohne Parameter
Rückgabe:	2 Bytes: Byte 1: derzeitige Einstellung Byte 2: Anzahl der möglichen Einstellungen
SET:	1 Parameter (1 Byte): neue Einstellung

Output Curve Einstellungen werden ab 01 aufsteigend gezählt; im oben angegebenen Fall wären also als Parameter die Werte 01, 02 und 03 erlaubt. Die Anzahl der möglichen Einstellungen würde als 03 zurückgemeldet. Durch Abfrage der augenblicklichen Output Curve kann man also die Anzahl der möglichen Einstellungen in Erfahrung bringen, und die namentliche Zuordnung durch die PID OUTPUT_CURVE_DESCRIPTION auslesen.

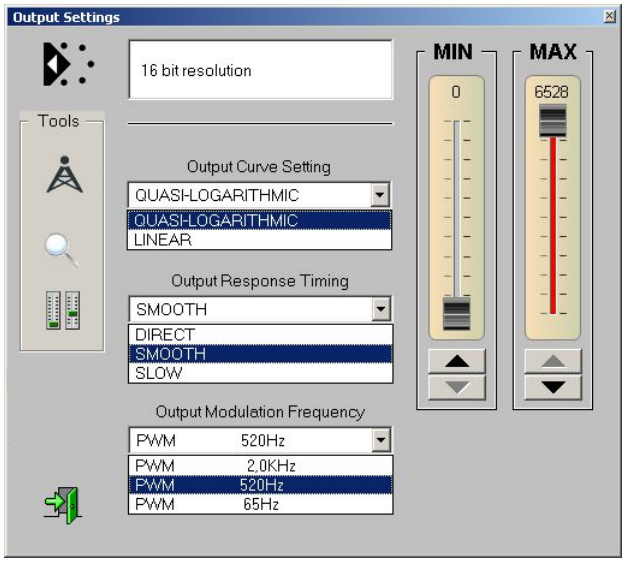
PID 0344: OUTPUT CURVE DESCRIPTION — Ausgabe einer Beschreibung für die Kurve

Die Eingabe der Curve-Nummer bei einer GET-Abfrage retourniert die Beschreibung der Kurve. Ein SET-Kommando ist hier nicht möglich.

PID 0345: OUTPUT RESPONSE AUSGANGSVERHALTEN

Das Ausgangsverhalten (Regelgeschwindigkeit) ist mit der Funktion OUTPUT RESPONSE TIME wählbar. Die Voreinstellung ist "SMOOTH" für optimal weiche Ausgabe.
Beui unseren Decodern stehen zumeist folgende Einstellmöglichkeiten zur Verfügung:

1: DIRECT	Sofortige Ausgabe des empfangenen DMX-Datenwertes. Kann bei zu niedriger DMX Refreshrate stufig erscheinen.
2:SMOOTH	Optimale Mittelung der einkommenden Daten
3: SLOW	Langsames Nachziehen des Ausganges.



Eingabe der Output Response beim GET/SET Controller. Wählen Sie einfach aus der aufklappbaren Drop-Down Liste aus.

Bei Abfrage über ein GET Kommando wird die derzeitige Einstellung ausgegeben:

GET	Aufruf ohne Parameter
Rückgabe:	2 Bytes: Byte 1: derzeitige Einstellung Byte 2: Anzahl der möglichen Einstellungen
SET:	1 Parameter (1 Byte): Neue Einstellung
Rückgabe:	Status

Output Response Einstellungen werden ab 01 aufsteigend gezählt; im oben angegebenen Fall wären also als Parameter die Werte 01, 02 und 03 erlaubt. Die Anzahl der möglichen Einstellungen würde als 03 zurückgemeldet. Durch Abfrage der augenblicklichen Response Einstellung kann man also auch die Anzahl der möglichen Einstellungen in Erfahrung bringen, und die namentliche Zuordnung sodann durch die jeweilige PID OUTPUT_RESPONSE_TIME_DESCRIPTION auslesen.

PID 0346: RESPONSE TIME DESCRIPTION	Ausgabe einer Beschreibung für die Glättung

Die Eingabe der ResponseTime-Nummer bei einer GET-Abfrage retourniert die Beschreibung des Status. Ein SET-Kommando ist hier nicht möglich.

PID 80E2: 16 BIT MODE **16 Bit Auflösung aktivieren**
PID 80E3: 16BIT RECEIVE MODE **16Bit DMX Empfang aktivieren**

DMX Daten sind grundsätzlich stets 8Bit Daten (Hexadezimal 00...FF, dezimal 0...255). Um 16Bit daten darstellen bzw. übertragen zu können, benutzt man zweit aufeinanderfolgende Bytes, die dann als ein 16Bit Wort interpretiert werden.

Dabei muss sichergestellt werden, dass diese beiden Datenbytes stets zusammen und aus dem leichen Übertragungspaket verarbeitet werden, da sonst ungültige Werte enstetehen können.

Das bedingt eine besondere Behandlung im DMX Empfänger. Diese kann je nach Erfordernis für den datenempfang oder die Verarbeitung eingeschaltet werden.

Default ist stets der 8-Bit Modus. .

PARAMETER: 1 Byte (16BIT MODE)

GET:	Aufruf ohne Parameter
Rückgabe:	Parameter: 1 Byte (16Bit-Modus)
SET:	Aufruf mit Parameter: 1 Byte (16Bit-Modus)
Rückgabe:	Status
Parameter:	$00 (0 dez) 16Bit Modus AUS
	$FF (255 dez) 16Bit Modus EIN

Hinweis:
Im 16Bit Empfangsmodus können nur geradzahlige Kaalzahlen empfangen werden (also 100 oder 400 oder 512 DMX Kanäle), nicht aber 255. In diesem Fall würden nur 254 ausgewertet werden, da das letzte Datenpaar unvollständig ist.

Jedes 16Bit Datenpaar beginnt mit einer ungeraden Adresse (also 1, 3, 5, 7, 9, 11 etc.) und benutzt die jeweils nächstfolgende Adresse als 16Bit Ergänzung. Ob die Reihenfolge LowByte-HighByte oder HighByte-LowByte ist, spielt für den Empfang keine Rolle. Das ist jedoch für die Datenauswertung wichtig und wird dann über die Auswahl der entsprechenden Personality eingestellt.

PID 8121: SLOT LABELS **SLOT LABEL SCHREIBEN**
PID 0121: SLOT LABELS*

Der Befehl SLOT LABELS existiert im RDM Standard E1.20, ist jedoch auf Lese-Zugriff beschränkt, d.h., Labels können nur ausgelesen, nicht aber durch den Anwender verändert werden. Das macht für viele Geräte der professionellen Lichttechnik, wie z.B. Moving Lites, durchaus Sinn, ist aber für solche Funktionsblöcke, wie wir sie anbieten -z.B- Relaismodule- eine schwerwiegende Einschränkung. Hier ist es durchaus notwendig, die einzelnen Slots mit anwenderspezifischen Texten zu belegen: beispielsweise Relais 1 mit „Motor“, Relais 2 mit „Ganglicht“, Relais 3 mit „Ventilator“ u.s.w.

Entgegen dem Standard dürfen Sie bei unseren Modulen also die PID 0121 auch schreiben; die Syntax und die Parameterliste ist dabei genauso aufgebaut wie beim genormten Lesezugriff. Da die meisten Editoren einen Schreibzugriff unter der Standard-PID zurückweisen werden, ist die Funktion auch als herstellerspezifisches Kommando unter der PID 8121 erreichbar. Hier ist ein Schreibzugriff in jedem Falle möglich. Das Kommando ist aufgrund seiner Struktur aber ein komplexes Kommando (CMSC, Complex Manufacturer Specific Command), das mehrere Parameter verwalten muss. Das wird nicht bei allen Controllern akzeptiert (positive Ausnahmen sind z.B. Der ENTTEC RDM Controller, GET/SET RDM Controller u.a.)

Wichtig: Slot Label sind auf 16 Zeichen begrenzt!

Wichtig: Die Funktion muss enabled werden, bevor in die Label-Register geschrieben werden kann.

User Slot Label enablen / einschalten:

Parameter:	<SLOT_LABEL_ON:Byte> $00: SlotLabel ausgeschaltet $FF: SlotLabel eingeschaltet
GET	Aufruf ohne Parameter
Rückgabe:	1 Byte: derzeitige Einstellung
SET:	Aufruf mit Parameter: 1 Byte (SLOT_LABEL_ON: Byte)
Rückgabe:	Status

Nachdem die Slot Labels anabled worden sind,können sich die angezeigten Slot Labels bereits verändert haben. Sie werden nun nicht mehr aus den Voreisntellungen ausgelesen, sondern aus dem Anwenderspeicher. Bei RGB-Decodern wechselt daher die Fader-Bennung von „Kanal 1, Kanal 2, Kanal 3“ usw. Gegebenenfalls auf „Rot, Grün, Blau“. Diese Label können Sie als Anwender nun überschreiben, z.B. Auf „Feuerrot, Grasgrün, Himmelblau“.

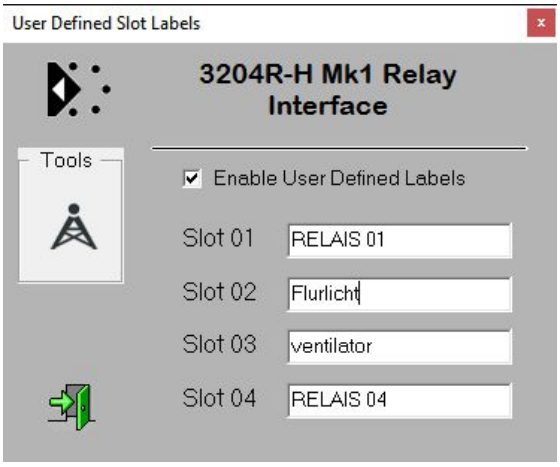
User Slot Label schreiben:

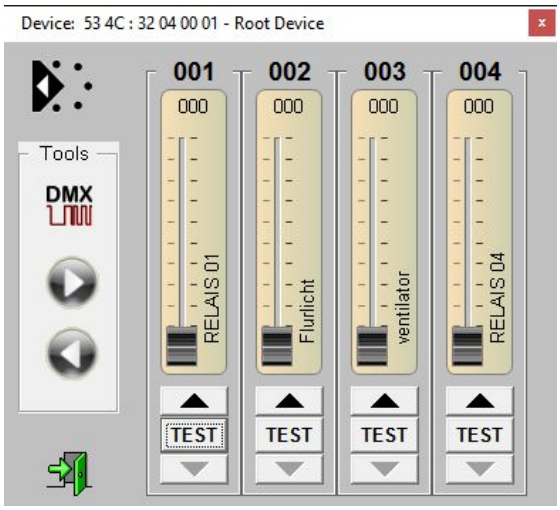
Parameter: <SlotLabel Nr:Word><SlotLabel Text: 16 Bytes>

Wobei:
SlotLabelNr: $0000 = Label 1
$0001 = Label 2
$0002 = Label 3
etc
SlotLabelText: ASCII Bytes

GET Aufruf mit Parameter: <SlotLabel Nr:Word>
Rückgabe: 18 Bytes: <SlotLabel Nr:Word>
<SlotLabel Text: 16 Bytes>

SET: Aufruf mit Parameter: 18 Bytes
<SlotLabel Nr:Word><SlotLabel Text: 16 Bytes>
Rückgabe: Status





Beim GET/SET Controller ist die User Slot Label Eingabe dank einer vorgefertigten Maske recht einfach: Funktion aktivieren und dann einfach die gewünschten Bezeichnungen eingeben.

Auf dem Faderpanel ist sofort die Auswirkung erkennbar. Die Texte werden im Decoder gespeichert, nicht im RDM Controller. Sie sind daher jederzeit und überall wieder auslesbar, auch durch andere Controller!

PID 8438 INPUT POLARITY EINGANGSPOLARITÄT ÄNDERN

This function sets the polarity of the switch sensor inputs.

Calls: GET <param = none> (no parameter needed)
Return: <param=Polarity [1 Byte]>

SET <param=Polarity [1Byte]>
Return: <param=none> (no parameter returned)

Polarity = $FF all normal polarity (standard mode)
Polarity = $00 all inverted polarity

From firmware version 1.1 onward, single switches can be inverted individually. Where available, use personality 3 „Test Mode“ to check for functionality and proper setting of the switch inputs. Switches should be set to show „L“ when disengaged (operational mode) and „H“ when engaged (error mode).
Changing the polarity of the center switch will result in changing the center point trigger flank.

RIGHT SWITCH: Bit 0 (normal: add value 1, inverted: add value 0)
CENTER SWITCH: Bit 1 (normal: add value 2, inverted: add value 0)
LEFT SWITCH: Bit 2 (normal: add value 4, inverted: add value 0)

Normal Mode: Bit set
Inverted Mode: Bit not set

Example: to invert the right and the left end sensor switch input, add 1+4 = 5.
Thus Polarity = $05

**= Laut RDM Standard ist PID 0121 nur auslesbar, aber nicht schreibbar. Unsere Geräte akzeptieren aber auch einen Schreibzugriff mit dem Standard-Befehlsformat. Für Controller, die daher einen Schreibzugriff unter PID 0121 ausschliessen, ist jederzeit ein identischer Zugriff unter PID 8121 möglich.*

PID A001 DALI DECADES DALI AUSGABEBEREICH

Der Ausgabebereich wird im DALI Standard mit Dekaden beschrieben (eine Dekade ist der Faktor 10). Der typische Ausgabebereich umfasst zwei Dekaden (1%...100%). Da Empfänger zur Verfügung stehen, die eine höhere Auflsöung ermöglichen, können hier 3 Dekaden (0,1%...100%) und für Empfänger mit geringerer Auflösung auch eine Dekade (10%...100%) spezifiziert werden.

Der DMX Wertebereich (0...255) - eine Faderlänge - wird somit auf eine Dekade, zwei oder 3 Dekaden aufgeteilt. Da die DALI Ausgabekennlinie logarithmisch verläuft, wird für jede Dekade der gleiche Faderweg belegt - bei zwei Dekaden also der halbe Faderweg für 1...10% und die zweite Hälfte für 10...100%. Bei drei Dekaden wird der Faderweg gedrittelt, das erste Drittel für 0,1...1%, das zweite Drittel für 1...10% und das letzte Drittel für 10...100%.

Mit dem Kommando A001 DALI DECADES lassen sich somit 1,2 oder 3 Dekaden zuweisen.

Mit DALI DECADES kann die Auflösung der DALI Intensitätsausgabe vorgegeben werden.

Parameter:	<DALI DECADES: Byte> DALI DECADES = $01...$03
GET	Aufruf ohne Parameter
Rückgabe:	1 Byte: derzeitige Einstellung
SET:	Aufruf mit Parameter: 1 Byte (DALI DECADES: Byte)
Rückgabe:	Status

PID A002 DALI REPEAT DALI PAKETWIEDERHOLUNG

Während DMX Daten laufend gesendet (wiederholt) werden, werden DALI Intensitätsdaten nur dann gesendet, wenn eine Änderung erfolgt ist. Vorteil: weniger Verkehr auf dem Datenbus (das ist bei einem „langsamen“ Bus wie dem DALI Bus vorteilhaft), nachteilig ist aber, dass der Empfänger nicht mehr synchron ist, wenn ihn ein ein telegramm nicht erreicht hat (weil er zwischendurch -aus welchen Gründen auch immer- offline war). Daher läßt sich ein ständiger Repeat einschalten, so-dass -wie bei DMX- ständig aktuelle Werte zur Verfügung stehen.

Mit dem Kommando A002 DALI REPEAT wird kontinuierliche Datenausgabe eingeschaltet..

Mit DALI REPEAT wird kontinuierliche Datenausgabe eingeschaltet..

Parameter DALI REPEAT:

Hex	Dez	Funktion
$00	0	aus: Daten nur bei Änderung senden
$FF	255	ein: Daten ständig wiederholen

GET Aufruf ohne Parameter
Rückgabe: 1 Byte: derzeitige Einstellung

SET: Aufruf mit Parameter: 1 Byte
(DALI REPEAT: Byte)
Rückgabe: Status

C000: SPEED SCALING STEPPER MOTOR GESCHWINDIGKEIT

Mit dem Speed Scaling kann ein Skalierungs-Faktor für die Geschwindigkeitseinstellung vorgegeben werden.

Parameter:	<SPEED_SCALING:Byte> STEP WIDTH = $01...$FF
GET	Aufruf ohne Parameter
Rückgabe:	1 Byte: derzeitige Einstellung
SET:	Aufruf mit Parameter: 1 Byte (SPEED_SCALING: Byte)
Rückgabe:	Status

HINWEIS: Die Funktion SPEED SCALING kann schreibgeschützt sein (LOCK STATE = 02 oder 03). In diesem Falle erfolgt keine Eingabebestätigung, sondern die Ausgabe "WRITE PROTECTED". Zur Freigabe ändern Sie den LOCK STATE unter Zuhilfenahme der gültigen PIN auf "00".

PID C001: STEP WIDTH — STEPPER MOTOR SCHRITTWEITE

Als Default wird ein DMX Schritt einem Stepper-Motor-Schritt zugeordnet. Pro DMX Step lassen sich 1 bis 99 Motorsteps zuweisen, um die Schrittweite zu vergrössern. Dazu ist die RDM-Funktion STEP WIDTH aufzurufen und der gewünschte Parameter (z.B. 2) einzugeben.

Parameter:	<STEP WIDTH:Byte> STEP WIDTH = $01...$FF
GET	Aufruf ohne Parameter
Rückgabe:	1 Byte: derzeitige Einstellung
SET:	Aufruf mit Parameter: 1 Byte (STEP WIDTH: Byte)
Rückgabe:	Status

HINWEIS: Die Funktion STEP WIDTH kann schreibgeschützt sein (LOCK STATE = 02 oder 03). In diesem Falle erfolgt keine Eingabebestätigung, sondern die Ausgabe "WRITE PROTECTED". Zur Freigabe ändern Sie den LOCK STATE unter Zuhilfenahme der gültigen PIN auf "00".

PID C002: PWM FACTOR STEPPER MOTOR PWM-FAKTOR

Wenn sich der Schrittmotor nicht dreht, führen die Motorwicklungen erhöhte Ströme, da die Gegeninduktion mangels Bewegung nicht auftritt. Das könnte zu thermischen Schäden im Motor führen. Schaltet man in diesem Fall jedoch den Motorstrom vollständig ab, hat der Motor keine Haltekraft mehr. Für den Fall, dass der Motor auf ein Untersetzungsgetriebe (Planetengetriebe, Schneckenantrieb etc.) arbeitet, kann der Strom auf Null gesetzt werden, da Selbsthaltung gewährleistet ist.
In allen anderen Fällen kann ein Haltestrom (0% [$00] bis 99% [$FF]) gesetzt werden.

Die Einstellung ist über die RDM-Funktion PWM FACTOR aufzurufen.

Die Funktion erlaubt die Eingabe von 0% (0 dez, 00hex) bis 100% (255 dez, FFhex).

Als Voreinstellung bei Auslieferung ist ein Wert von 60% (153dez, 99hex) gesetzt.

Parameter:	<PWM FACTOR:Byte> PWM FACTOR = $00...$FF
GET	Aufruf ohne Parameter
Rückgabe:	1 Byte: derzeitige Einstellung
SET:	Aufruf mit Parameter: 1 Byte (PWM_FACTOR: Byte)
Rückgabe:	Status

HINWEIS: Die Funktion PWM FACTOR kann schreibgeschützt sein (LOCK STATE = 02 oder 03). In diesem Falle erfolgt keine Eingabebestätigung, sondern die Ausgabe "WRITE PROTECTED". Zur Freigabe ändern Sie den LOCK STATE unter Zuhilfenahme der gültigen PIN auf "00".

HINWEIS: Alle Werte (sowohl für Schrittweite als auch für Haltestrom) müssen als Hexadezimalwerte eingegeben werden. Verwenden Sie den Windows Calculator, um bequem zwischen Dezimal- und Hexadezimalsystem zu konvertieren.

PID C003: LOWER LIMIT — UNTERE FAHRBEREICHS-GRENZE

Mit dieser Funktion kann "der untere Anschlag" beim Positionierbetrieb festgelegt werden. Die Funktion ist nur bei Positionierbetrieb aktiv.

Parameter:	<LIMIT_LO:Byte> LIMIT_LO = $00...$FF
GET	Aufruf ohne Parameter
Rückgabe:	1 Byte: derzeitige Einstellung
SET:	Aufruf mit Parameter: 1 Byte (LIMIT_LO: Byte)
Rückgabe:	Status

Limit_Lo = $00...$FE	übernimmt den *eingegebenen Wert* als Limit
Limit_Lo = $FF	übernimmt die *aktuelle Position* als Limit

HINWEIS: Die Funktion LOWER LIMIT kann schreibgeschützt sein (LOCK STATE = 02 oder 03). In diesem Falle erfolgt keine Eingabebestätigung, sondern die Ausgabe "WRITE PROTECTED". Zur Freigabe ändern Sie den LOCK STATE unter Zuhilfenahme der gültigen PIN auf "00".

PID C004: UPPER LIMIT — OBERE FAHRBEREICHS-GRENZE

Mit dieser Funktion kann "der obere Anschlag" beim Positionierbetrieb festgelegt werden. Die Funktion ist nur bei Positionierbetrieb aktiv.

Parameter:	<LIMIT_HI:Byte> LIMIT_HI = $00...$FF
GET	Aufruf ohne Parameter
Rückgabe:	1 Byte: derzeitige Einstellung
SET:	Aufruf mit Parameter: 1 Byte (LIMIT_HI: Byte)
Rückgabe:	Status

Limit_Hi = $00...$FE	übernimmt den *eingegebenen Wert* als Limit
Limit_Hi = $FF	übernimmt die *aktuelle Position* als Limit

HINWEIS: Die Funktion UPPER LIMIT kann schreibgeschützt sein (LOCK STATE = 02 oder 03). In diesem Falle erfolgt keine Eingabebestätigung, sondern die Ausgabe "WRITE PROTECTED". Zur Freigabe ändern Sie den LOCK STATE unter Zuhilfenahme der gültigen PIN auf "00".

PID C005: END SWITCH POLARITY — POLUNG ENDSCHALTER

Mit der Funktion C005 kann die Polarität der Endschalter invertiert werden, d.h., statt eines Öffner-Kontaktes wird dann ein Schliesser-Kontakt ausgewertet (oder umgekehrt). Die Funktion erlaubt die Anpassung des Decoders an die baulichen Gegebenheiten der Installation.

Parameter:	<POLARITY:Byte>	
	POLARITY = $FF:	normal
	POLARITY = $00:	invertiert
GET	Aufruf ohne Parameter	
Rückgabe:	1 Byte: derzeitige Einstellung	
SET:	Aufruf mit Parameter: 1 Byte (POLARITY: Byte)	
Rückgabe:	Status	

PID C006: POS DECEL RANGE — POSITIONIERUNGS-BREMSZONE

Mit dieser Funktion wird der Abbremsbereich festgelegt. Innerhalb des Abbremsbereiches wird die Geschwindigkeit linear bis zur vorgegebenen Minialgeschwindigekit reduziert. Der Abbremsbereich kann auf 250, 500, 1000 oder 2000 Steps festgelegt werden.

Aufrufe: GET <param = none> (kein Eingabeparameter erforderlich)
Rückgabe: <param=Decel_Range [Byte]>

SET <param=Decel_Range [Byte]>
Rückgabe: <param=none> (kein Rückgabeparameter)

Decel_Range = $01...$04
übernimmt den eingegebenen Wert als Bereichsangabe

$01 =	255 Steps
$02 =	512 Steps
$03=	1024 Steps
$04 =	2048 Steps

Decel_Range = $81...$85
übernimmt den eingegebenen Wert als Bereichsangabe

$81 =	255 Steps
$82 =	127 Steps
$83=	63 Steps
$84 =	31 Steps
$85 =	15 Steps

FUNKTION C007 POS DEAD ZONE TOTZONE

Mit dieser Funktion wird der die Totzone um den Nullpunkt (in Steps) festgelegt.

Aufrufe: GET <param = none>
(kein Eingabeparameter erforderlich)
Rückgabe: <param=DeadZone [Byte]>

SET <param=DeadZone [Byte]>
Rückgabe: <param=none>
(kein Rückgabeparameter)

DeadZone = $00...$FF
übernimmt den eingegebenen Wert als Bereichsangabe

HINWEIS:
Setzen Sie eine Pos_Dead_Zone, also eine Positionierungs-Totzone, dann sollten Sie keine Positionierungs-Accuracy aktivieren (diese beiden Einstellungen überlagern sich). Setzen Sie dann PositioningAccuracy = $00

PID C008: INIT SPEED INITALISIERUNGS-GESCHWINDIGKEIT

Mit dieser Funktion kann die Geschwindigkeit bei der Initialisierung (Suchen des Nullpunktes) festgelegt werden. Die initialization Speed ist eine Festgeschwindigkeit und beim Betrieb nicht veränderbar. Beachten Sie jedoch, dass die Initialisierung dennoch nur dann fährt, wenn dies durch den CONTROL Regler (meist DMX Kanal 1) jeweils freigegeben ist.

Parameter:	<INIT_SPEED:Byte> INIT_SPEED = $00...$FF
GET	Aufruf ohne Parameter
Rückgabe:	1 Byte: derzeitige Einstellung
SET:	Aufruf mit Parameter: 1 Byte (INIT_SPEED: Byte)
Rückgabe:	Status

Init_Speed = $00...$FF übernimmt den eingegebenen Wert als Geschwindigkeit

TIPP:
Es empfiehlt sich, das Gerät im Endlos-Modus (Personality 2) zu betreiben und die "richtige" Geschwindigkeit auszuprobieren. Notieren Sie die Einstellung und übernehmen Sie diese für die Programmierung.

ACHTUNG: Wenn mechanische Endpunkte berücksichtigt werden müssen, darauf achten, dass diese bei der Testfahrt nicht überfahren werden!

PID C009: ACCURACY BITS POSITIONIERUNGSGENAUIGKEIT (BITS)

Mit dieser Funktion kann "die Genauigkeit beim Einparken" beim Positionierbetrieb festgelegt werden.

Die Funktion ist nur bei Positionierbetrieb aktiv. Dabei wird aus der Positionierungsposition (Zähler) per Bitmaske ein Wertebereich ausgeblendet, für den gilt: "Ziel erreicht".
Als Vorstellung ist der Parameter 3 gesetzt (8 Steps)

Parameter:	<ACCURACY_BITS:Byte> ACCURACY_BITS = $00...$07
GET Rückgabe:	Aufruf ohne Parameter 1 Byte: derzeitige Einstellung
SET: Rückgabe:	Aufruf mit Parameter: 1 Byte (ACCURACY_BITS: Byte) Status

Accuracy_Bits = $00...$07	übernimmt den eingegebenen Wert als Bitzahl
	$00 volle Auflösung $01 Totzone 2 Steps $02 Totzone 4 Steps $03 Totzone 8 Steps $04 Totzone 16 Steps ....u.s.w. bis $07 Totzone 128 Steps

PID C00A: AUTO_INIT ENABLED AUTOMATISCHE INITIALISIERUNG

Mit dieser Funktion wird festgelegt, ob im automatischen Positioniermodus eine automatische Initialisierung gefahren wird. Die Funktion wirkt nur in Personality 1.

Aufrufe: GET <param = none>
(kein Eingabeparameter erforderlich)
Rückgabe: <param=Auto_Init [Byte]>

SET <param=Auto_Init [Byte]>
Rückgabe: <param=none>
(kein Rückgabeparameter)

Auto_Init = $00 Initalierung aus
Auto_Init = $FF Initialisierung ein

Hinweis: *Bitte beachten Sie, dass für eine automatische Initialisierung z.B. ein Nullpunktsensor vorhanden sein muss. Wird kein Nullpunkt gefunden, weil z.B. Bei Erst-Inbetriebnahme die aktuelle Position vollkommen „Out of Range“ war, dann wird die Initialisierung nach einigen Versuchen mit einer Error-Meldung abgebrochen. Ein Benutzer-Eingriff ist dann erforderlich. Das Gerät muss dann gegebenenfalls durch ein Reset-Kommando neu gestartet werden.*

SICHERHEITS-HINWEIS:
Nach einem automatischen Abbruch einer sicherheitsrelevanten Funktion darf eine weitere Neuinitialisierung der Anlage aus Sicherheitsgründen nur unter der Aufsicht eines Operators, der jederzeit eingreifen kann, erfolgen!

Dieser Vorgang kann sich wiederholen, bis der nutzbare Arbeitsbereich erreicht worden ist.

PID C010 MID POINT OFFSET

This function sets the position offset for the center point sensor. Usually, a inductive sensor is being used, which is detected over a certain span when the nozzle is moving.

The zero point detector is flank triggered only when the nozzle is moving upward. To compensate for the detection range, a MID POINT OFFSET value can be defined using this function. The data entry range is 000...255 (00*hex*...FF*hex*), with the default set to center position 128 (80*hex*) at the factory. Normally, this will work out fine for all standard applications.

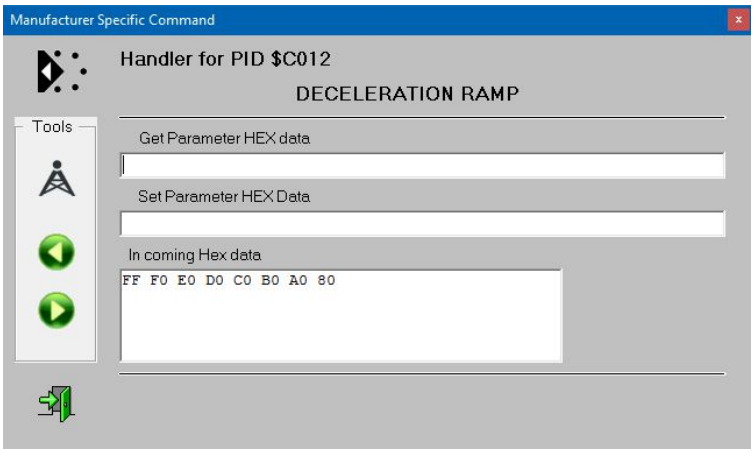
Calls: GET <param = none> (no parameter needed)
Return: <param=MidPointOffset [1 Byte]>

SET <param=MidPointOffset [1 Byte]>
Return: <param=none> (no parameter returned)

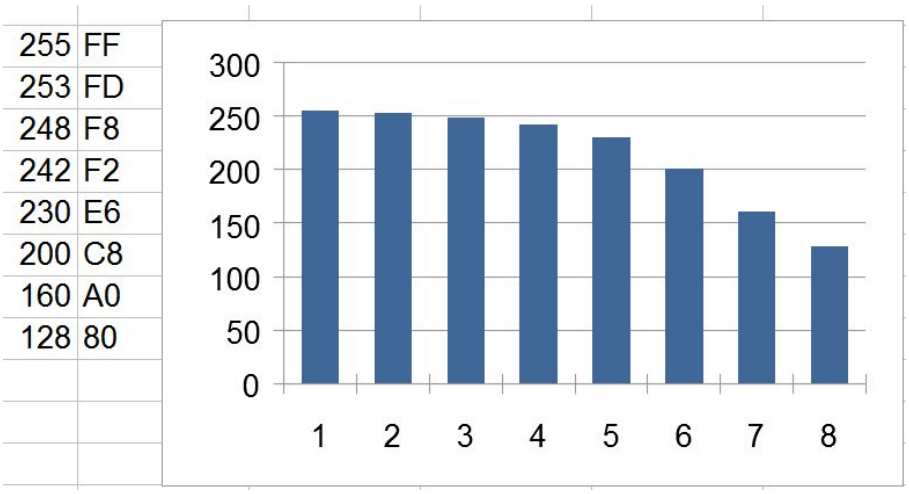
FUNCTION C011 ACCELERATION RAMP
FUNCTION C012 DECELERATION RAMP

This function sets the ramp for acceleration or deceleration of the stepper motor. 8 values must be given to define the acceleration curve, with full speed = $FF and zero speed = $00. See example for more details.

Calls: GET <param = none> (no parameter needed)
Return: <param=Ramp [8 Bytes]>



GET Mask



Ramp design

SET <param=Ramp [8Bytes]>
Return: <param=none> (no parameter returned)

Ramp = $00...$FF the value will be taken as speed factor

PID C013 MOVING RANGE

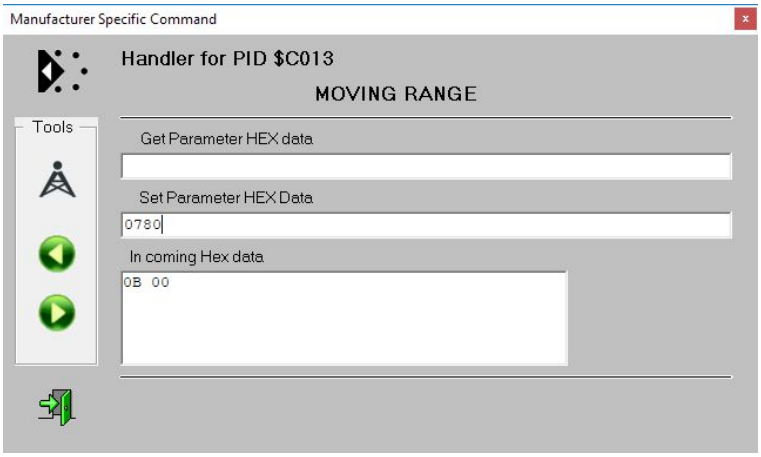
This function sets the total moving range of the nozzle. The standard resolution (when setting the stepper motor driver as stated in the addendum) is 32 steps per degree. This will result in the settings as per table 1:

Angle [°]	Steps	Hexadecimal
30	960	03C0
45	1440	05A0
60	1920	0780
90	2880	0B40
120	3840	0F00
150	4800	12C0
180	5760	1680

Table 1 Angle Step setting

Calls: GET <param = none> (no parameter needed)
Return: <param=StepSetting [2 Bytes]>

SET <param=StepSetting [2Bytes]>
Return: <param=none> (no parameter returned)



Changing from >90° to 60° (+/-30°) moving range

Adjusting the moving range will automatically scale the position faders to cover the full range.

PID C0C0: INTERNAL PATCHING Zuweisung der DMX Datenquelle für das Relais

Bei der monostabilen Funktion wird normalerweise Relais 1 durch DMX Kanal1, Relais 2 durch DMX Kanal 2 ausgelöst. Sie können die Quelle für die Auslösung durch diese Funktion neu zuordnen. Das ist nützlich, wenn z.B. ein Relais durch die aufsteigende, das andere durch die abfallende Flanke getriggert werden soll, aber beide auf dasselbe Eingangssignal reagieren sollen.

Parameter:	<Kanalnummer:word> <Source:byte>
GET	Aufruf ohne Parameter
Rückgabe:	n Bytes (n= Anzahl der Kanäle) Byte 1-n: derzeitige Einstellung
GET	Aufruf mit Parameter: 2 Bytes (Kanalnummer [16 Bit])
Rückgabe:	1 Byte Byte 1: derzeitige Einstellung
SET:	Aufruf mit Parameter: 3 Bytes (Kanalnummer, 16 Bit und Einstellung 1 Byte [8 Bit]
Rückgabe:	Status

Source:	01	Kanal 1
	02	Kanal 2
	....	
	06	Kanal 6

PID C0E0: DMX DATA POLARITY Flanke für die Auslösung des monostabilen Relais

Die monostabile Funktion wird normalerweise durch die aufsteigende Flanke (Übergang 0->100%) ausgelöst. Sie können die Polung durch diese Funktion umkehren, sodass der monostabile Impuls durch eine abfallende Flanke ausgelöst wird. Die Eingabe ist wie in der Monostabil-Funktion vorzunehmen.

Parameter:	<Kanalnummer:word> <Polarity:byte>
GET	Aufruf ohne Parameter
Rückgabe:	n Bytes (n= Anzahl der Kanäle) Byte 1-n: derzeitige Einstellung
GET	Aufruf mit Parameter: 2 Bytes (Kanalnummer, 16 Bit)
Rückgabe:	1 Byte Byte 1: derzeitige Einstellung
SET:	Aufruf mit Parameter: 3 Bytes (Kanalnummer, 16 Bit und Einstellung 1 Byte (8 Bit)
Rückgabe:	Status

Polarity:	00	invertiert
	FF	normal

Relaisfunktionen beim Detektor-Betrieb

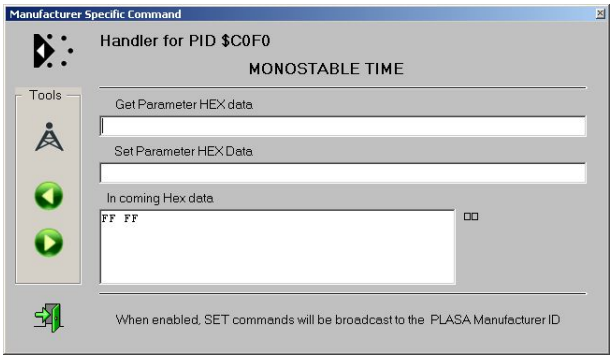
Hinweis:
Die PIDs C0C0, C0E0 und C0F0 stehen beim Detektor-Betrieb zwar zur Verfügung, sind hier aber wirkungslos, da dann hier keine Eingaben gemacht werden können.
Da im Detektor-Betrieb keine DMX-Slots vergeben sind, stehen diese weder als Eingabequelle noch als Eingabeziel zur Verfügung.

PID C0F0: MONOSTABLE TIME Setzen der monostabilen Impulsdauer

Die Ausgangsrelais z.B. der 3202R-H arbeiten im Grundmodus bistabil, d.h., sie bleiben in der angesteuerten Position. Die Funktion läßt sich auf monostabilen Betrieb (Impulskontakt) umschalten, wobei die Impulsdauer einstellbar ist. Dazu dient die Funktion C0F0.

Parameter: <Kanalnummer> <Monotime>
wobei: Kanalnummer = 0001: Kanal 1
Kanalnummer = 0002: Kanal 2
....
Kanalnummer = 0006: Kanal 6
Kanalnummer = FFFF: alle

Monotime: 01...3F 25ms...1,5s in 25ms steps
41...7F 0,25s...15,75s in 250ms steps
FF: bistabiler Modus



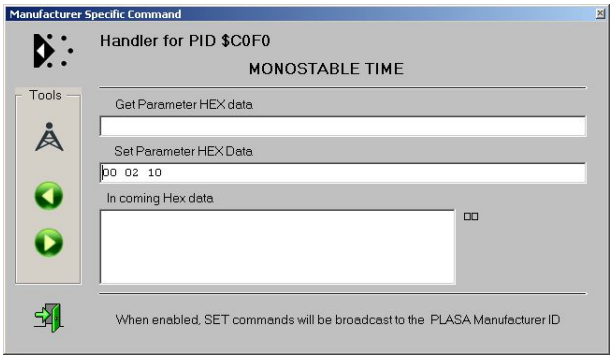
Eingabe mit dem GET/SET Controller. Es werden so viele Parameter ausgegeben, wie Relais (Kanäle) vorhanden sind.

Drücken Sie GET, um die aktuelle Belegung auszulesen.
Im obigen Beispiel wird für beide Kanäle der bistabile Modus angezeigt.

Parameter: <Kanalnummer:word> <Monostable Time:byte>

GET Aufruf ohne Parameter
Rückgabe: n Bytes (n= Anzahl der Kanäle)
Byte 1...n: derzeitige Einstellung
GET Aufruf mit Parameter: 2 Bytes (Kanalnummer, 16 Bit)
Rückgabe: 1 Byte : derzeitige Einstellung

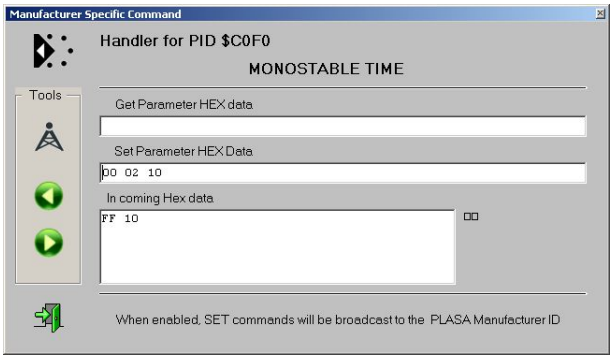
SET: Aufruf mit Parameter: 3 Bytes
(Kanalnummer, 16 Bit und Einstellung 1 Byte (8 Bit)
Rückgabe: Status



Um Relais 2 auf monostabilen Modus mit einer Impulszeit von 0,4 Sekunden (400ms) zu setzen, geben Sie ein:

0002 10 (hexadezimal)
weil: 10hex = 16dez, 16*25ms = 400ms

Drücken Sie den SET Knopf, um die Werte zu übernehmen.



Prüfung:
Bei der Kontrolle über GET wird die neue Zuweisung ausgegeben. Relais 2 arbeitet nun monostabil mit einer Impulszeit von 0,4 sec.

Das Relais bleibt solange angezogen, bis die eingestellte monostabile Impulsdauer abgelaufen ist. Dabei ist es gleichgültig, ob der Fader zur Ansteuerung aufgezogen ist (und aufgezogen bleibt) oder vorher zugezogen wird. Die Impulsdauer bleibt also in jedem Fall konstant.

Impulsdauer abkürzen

Sie können die Impulsdauer abkürzen, indem Sie den Fader zur Ansteuerung vor Ablauf der Impulsdauer wieder zuziehen. Dieses Verhalten kann für jedes Relais einzeln eingestellt werden.
Um diese Einstellung zu aktivieren, erhöhen Sie die eingestellte monostabile Zeit um 80(hex) bzw. 128 (dez). Die Einstellungen lauten dann:

Monotime:	*81...BF*	*25ms...1,5s in 25ms steps*
	C1...FE	*0,25s...15,5s in 250ms steps*
	FF:	*bistabiler Modus*

PID C0F1: EXCLUSIVE MODE — Relais schalten gegenseitig verriegelnd

Mit dem EXCLUSIVE MODE wird eine gegenseitige Verriegelung der Ausgangsrelais aktiviert. Die Zuordnung ist dann wie folgt:

CH1	CH2	RELAIS1	RELAIS 2
OFF	OFF	OFF	OFF
ON	OFF	ON	OFF
OFF	ON	OFF	ON
ON	ON	OFF	OFF

Parameter: <MODE:Byte>
MODE = $FF (255): Exclusive ON
MODE= $00 (0): Exclusive OFF

GET Aufruf ohne Parameter
Rückgabe: 1 Byte: derzeitige Einstellung

SET: Aufruf mit Parameter: 1 Byte
(MODE: Byte)
Rückgabe: Status

Bei Modulen oder Karten mit mehreren Relais gilt die Exklusiv-Verknüpfung jeweils für die Relaispaare:
- Relais 1 und Relais 2
- Relais 3 und Relais 4
- Relais 5 und Relais 6

...etc...

PID C0F2: FAST MODE — Hohe Schaltgeschwindigkeit setzen

Dieser Modus schaltet die Glitch-Unterdrückung ab und verhindert somit die Eingangsdaten-Mittelung; die Relais werden sofort nach Eintreffen des Datenpaketes ohne weitere Prüfung geschaltet. Dadurch wird die höchstmögliche Schaltgeschwindigkeit erreicht.

Parameter:	<MODE:Byte>	
	MODE = $FF (255):	Fast Mode ON
	MODE= $00 (0):	Fast Mode OFF

GET	Aufruf ohne Parameter
Rückgabe:	1 Byte: derzeitige Einstellung
SET:	Aufruf mit Parameter: 1 Byte (MODE: Byte)
Rückgabe:	Status

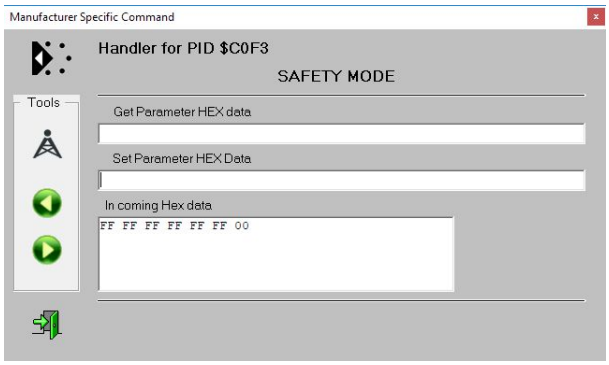
PID C0F3: SAFETY MODE — Auf SAFETY-Modus umschalten

Dieser Modus schaltet Relais auf Safety-Modus um und macht damit die Auslösung von einem Sicherheitskanal abhängig. Eine Auslösung erfolgt dann nur, wenn der Sicherheitskanal auf einen entsprechenden Sicherheitswert gesetzt wird.

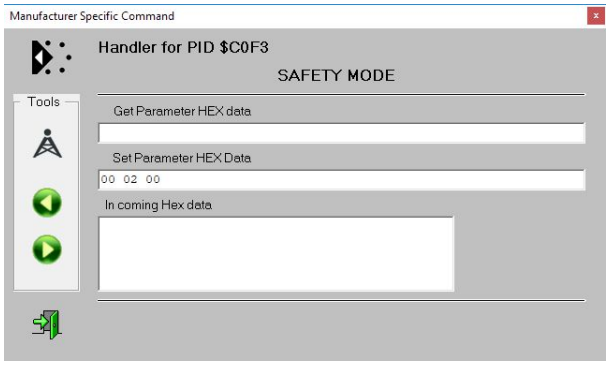
Parameter: <SAFETY_MODE:Byte>
MODE = $FF (255): Safety_Mode ON
MODE= $00 (0): Safety_Mode OFF

GET Aufruf ohne Parameter
Rückgabe: n Bytes (n= Anzahl der Kanäle)
Byte 1...n: derzeitige Einstellung
GET Aufruf mit Parameter: 2 Bytes (Kanalnummer, 16 Bit)
Rückgabe: 1 Byte : derzeitige Einstellung

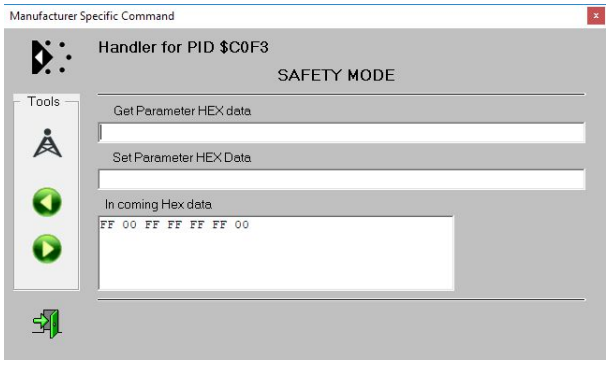
SET: Aufruf mit Parameter: 3 Bytes
(Kanalnummer, 16 Bit und Einstellung 1 Byte (8 Bit)
Rückgabe: Status



GET Abfrage ohne Parameter, alle Relais 1-6 auf Sicherheitskanal gesetzt



SET: für Relais 2 wird die Kopplung an den Sicherheitskanal aufgehoben



GET Abfrage ohne Parameter, alle Relais außer Relais Nr. 2 und Relais Nr. 6 sind auf Sicherheitskanal gesetzt

PID C0F4: SECURITY DELAY — Delay time for sensor evaluation

The sensor delay time can be activated to delay sensor related action. Thus it will be possible to trigger a relay and to disable the output again as soon as the sensor is triggered. This allows to create applications like automatic filling stations (relay activated and de-activated as soon as sensor reports filling level) or flame detectors (process started but halted if sensor not in range after delay time).

Delay time is defined in increments of 25ms. Thus the total time can be adjusted from 0...6,3 seconds.

Parameter: <Slot number> <DELAY> [Byte]
where: Slot number = $0001: Slot no. 1
Slot number = $0002: Slot no. 2
Slot number = $FFFF: all

DELAY = $00 (000) ... $FF (255)
(timecount * 25ms)

Example: The safety supervision for relay 2 shall be engaged after 2 seconds:

SET PID C0F4: 00 02 50

Calculation: 2 seconds is 80x 25ms, 80(dec) is 50 hex ($50). Most RDM controllers require values to be entered in hex format.

PID C0F5: DELAY TIME — DELAY-ZEIT EINSTELLEN

Dieser Modus schaltet eine Signalverzögerung zu. Das Signal Delay wird in 25ms Schritten spezifiziert.

Parameter:	<SIGNAL_DELAY:Byte> SIGNAL_DELAY = $01...$FE
GET	Aufruf ohne Parameter
Rückgabe:	n Bytes (n= Anzahl der Kanäle) Byte 1...n: derzeitige Einstellung
GET	Aufruf mit Parameter: 2 Bytes (Kanalnummer, 16 Bit)
Rückgabe:	1 Byte : derzeitige Einstellung
SET:	Aufruf mit Parameter: 3 Bytes (Kanalnummer, 16 Bit und Einstellung 1 Byte (8 Bit)
Rückgabe:	Status

PID C0F6: SAFETY WINDOW FENSTERBREITE FÜR AUSLÖSUNG FESTLEGEN

Soll die Auslösung für Sicherheitszwecke auf ein bestimmtes Pegelfenster festgelegt werden, dann kommt die Funktion SAFETY WINDOW zur Anwendung.

Parameter: <SAFETY_WINDOW:Byte,Byte>
1. Byte: unterer Pegelwert ($00...$FF)
2. Byte: oberer Pegelwert ($00...$FF)

GET Aufruf ohne Parameter
Rückgabe: 2 Bytes (unterer, oberer Pegelwert)

SET: Aufruf mit Parameter: 2 Bytes
unterer Pegel (1 Byte), oberer Pegel (1 Byte)
Rückgabe: Status

PID DC01: OUTPUT CURRENT AUSGANGSSTROM FESTLEGEN

Mit diesem Befehl wird der Ausgangsstrom bei LED-Stromtreibern festgelegt. Ob die Einstellung kontinuierlich oder in Stufen möglich ist, wird durch die jeweilige Hardware bestimmt. Bei stufiger Einstellung wird jeweils die Stufe gesetzt, die gewählt (oder überschritten) worden ist.

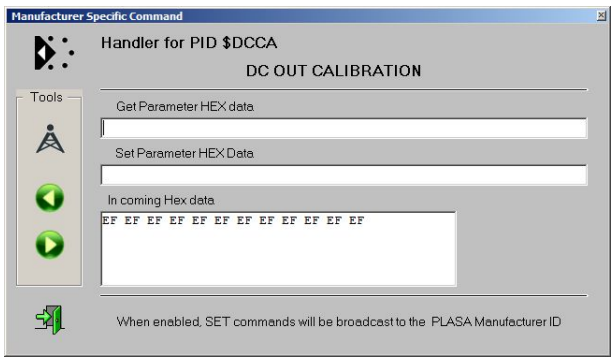
Parameter:	<OUTCURRENT: word> word: Strom in mA
GET	Aufruf ohne Parameter
Rückgabe:	2 Bytes (word)
SET:	Aufruf mit Parameter: 2 Bytes (word) OUTCURRENT: word
Rückgabe:	Status

Typische Werte für Stromeinstellungen:

350 mA:	$ 015E
500 mA:	$ 01F4
600 mA:	$ 0258
700 mA:	$ 02BC

PID DCCA: OUTPUT CALIBRATION — AUSGANGS-KALIBRIERUNG

This menu allows to set n individual calibration values for outputs 1...n. Data entry uses hexadecimal number format.



(shown: JESE GET/SET controller, setting for 12-channel DMX Demux 3012C-EP)

GET entry values:

none	displays list of calibration factors
00 xx	displays calibration factor for slot xx (xx=01...nn)

SET entry values:

00 xx yy	Sets calibration factor yy for slot xx (yy=00...FF; xx=01...nn)
FF FF yy	Sets same calibration factor yy for all outputs

PID E001: LOWER TEMP TRIP	**UNTERER TEMPERATUR-REGELPUNKT**
PID E002: UPPER TEMP TRIP	**OBERER TEMPERATUR-REGELPUNKT**
PID E003: ALARM TEMP TRIP	**ALARMTEMPERATUR-SCHALTPUNKT**
PID E004: TEMP SAMPLE TIME	**TEMPERATUR-SAMPLEZEIT**
PID E005: TEMP DROP LEVEL	**TEMPERATUR-ABFALLPEGEL**

Diese Funktionen dienen der Einstellung des Temperatur-Managements. Sie sollten vom Anwender nicht verstellt werden. Detailliertere Hinweise entnehmen Sie bitte dem separat erhältlichen Temperaturmanagement-Manual.

PID E001: LOWER TEMP TRIP UNTERER TEMPERATUR-REGELPUNKT

Dieser Einstellpunkt gibt die untere Temperaturschwelle an. Mit Unterfahren der eingestellten Schwelle wird die Master-Begrenzung wieder langsam zurückgeregelt. Als Default ist hier ein Temperaturwert von 60C eingestellt. Das bedeutet also: sobald die Temperatur 60C wieder unterschreitet, werden die Ausgänge wieder aufgeregelt.

Die Einstellung kann mit einem RDM-Editor ausgelesen und neu gesetzt werden.

Parameter:	<LOWER TEMP:Byte> LOWER TEMP = $00...$FF, $00=OFF
GET:	Aufruf ohne Parameter
Rückgabe:	1 Byte: derzeitige Einstellung
SET:	Aufruf mit Parameter: 1 Byte (LOWER TEMP: Byte)
Rückgabe:	Status

Der Einstellwert muß in 0,5Grad Schritten angegeben werden, d.h., zur Einstellung von 60C muß der Datenwert 120 (78hex) eingegeben werden. Die Eingabe von 0 (00hex) schaltet die Auswertung des Parameters ab. Bitte beachten Sie, daß viele RDM-Editoren zur Eingabe hexadezimale Werte erwarten- die Umrechnung kann mit einem geeigneten Rechner (z.B. Windows Calculator) erfolgen.

PID E002: UPPER TEMP TRIP OBERER TEMPERATUR-REGELPUNKT

Dieser Einstellpunkt gibt die obere Temperaturschwelle an. Mit Überfahren der eingestellten Schwelle wird die Master-Begrenzung langsam eingeregelt. Als Default ist hier ein Temperaturwert von 70C eingestellt. Das bedeutet also: sobald die Temperatur 70C überschreitet, werden die Ausgänge zurückgeregelt.

Das Maß der Zurückregelung bestimmt sich aus der Höhe der Überschreitung des Schwellwertes und beträgt etwa 10% pro Grad Überschreitung. Der minimale Ausgabepegel ist 20%. Die Einstellung kann mit einem RDM-Editor ausgelesen und neu gesetzt werden.

Parameter:	<UPPER TEMP:Byte> UPPER TEMP = $00...$FF, $00=OFF
GET:	Aufruf ohne Parameter
Rückgabe:	1 Byte: derzeitige Einstellung
SET:	Aufruf mit Parameter: 1 Byte (UPPER TEMP: Byte)
Rückgabe:	Status

Der Einstellwert muß in 0,5Grad Schritten angegeben werden, d.h., zur Einstellung von 70C muß der Datenwert 140 (8Chex) eingegeben werden. Die Eingabe von 0 (00hex) schaltet die Auswertung des Parameters ab. Bitte beachten Sie, daß viele RDM-Editoren zur Eingabe hexadezimale Werte erwarten- die Umrechnung kann mit einem geeigneten Rechner (z.B. Windows Calculator) erfolgen.

PID E003: ALARM TEMP TRIP ALARMTEMPERATUR-SCHALTPUNKT

Dieser Einstellpunkt gibt die absolute obere Temperaturschwelle an. Mit Überfahren der eingestellten Schwelle wird die eine Abschaltung ausgelöst, die die Ausgänge auf Minimalpegel (ca. 20%) begrenzt. Als Default ist hier ein Temperaturwert von 80C eingestellt. Mit Auslösen der Abschaltung wird die Signalisierung auf Alarm-Signalisierung (Temperatur-Sensor-Fehler: rot/grün blinkt gleichzeitig schnell) umgeschaltet.

Die Einstellung kann mit einem RDM-Editor ausgelesen und neu gesetzt werden.

Parameter:	<ALARM TEMP:Byte> ALARM TEMP = $00...$FF, $00=OFF
GET:	Aufruf ohne Parameter
Rückgabe:	1 Byte: derzeitige Einstellung
SET:	Aufruf mit Parameter: 1 Byte (ALARM TEMP: Byte)
Rückgabe:	Status

Der Einstellwert muß in 0,5Grad Schritten angegeben werden, d.h., zur Einstellung von 80C muß der Datenwert 160 (A0hex) eingegeben werden. Die Eingabe von 0 (00hex) schaltet die Auswertung des Parameters ab. Bitte beachten Sie, daß viele RDM-Editoren zur Eingabe hexadezimale Werte erwarten- die Umrechnung kann mit einem geeigneten Rechner (z.B. Windows Calculator) erfolgen.

PID E004: TEMP SAMPLE TIME TEMPERATUR-SAMPLEZEIT

Einstellung des Temperatur-Abtastintervalls. Das Abtastintervall wird in Minuten spezifiziert. Die Default-Einstellung ist 12 Minuten.

Parameter:	<SAMPLE TIME:Byte> SAMPLE TIME = $00...$FF, $00=OFF
GET:	Aufruf ohne Parameter
Rückgabe:	1 Byte: derzeitige Einstellung
SET:	Aufruf mit Parameter: 1 Byte (SAMPLE TIME: Byte)
Rückgabe:	Status

PID E005: TEMP DROP LEVEL TEMPERATUR-ABFALLPEGEL

Der Drop Level bezeichnet die Intensität, bis auf die bei Übertemperatur heruntergeregelt wird. Die Default-Voreinstellung ist 50% bzw. $80 (128 dez), (bezogen auf den DMX Pegel, nicht auf die Ausgabe-Intensität!). Der Pegelwert kann im Bereich von 000-255 eingegeben werden.

Parameter:	<TEMP DROP:Byte> TEMP DROP = $00...$FF
GET:	Aufruf ohne Parameter
Rückgabe:	1 Byte: derzeitige Einstellung
SET:	Aufruf mit Parameter: 1 Byte (TEMP DROP: Byte)
Rückgabe:	Status

PID E00E: TEMPSENSE ENABLE SENSOR-FREISCHALTUNG

Aktivierung der einzelnen Temperatur-Sensoren. Pro Eingang wird ein Bit gesetzt; 0= aus 1=ein. Es werden zwei Bytes verwendet.
Die TEMP SENSE ENABLE Funktion soll benutzt werden, um beispielsweise einen defekten Sensor oder nicht korrekt messenden von der Erfassung ausschließen zu können.

Default: 255 255 (1111 1111 1111 1111)

Die Zuordnung ist dem jeweiligen Gerätemanual zu entnehmen

PID FF08: RDM CONFIG ACCESS Konfiguration über RDM erlauben

Diese Einstellung definiert das Zeitintervall, während dessen Konfigurationseinstellungen über DMX RDM vorgenommen werden können. Sonst ist das Gerät verriegelt und Einstellungen der Betriebsparameter sind dann nicht möglich.

Die Einstellung kann mit einem RDM-Editor ausgelesen und neu gesetzt werden. Diese PID ist nicht von sonstigen Sperrungen (z.B. LOCK MODE) betroffen. Andere PIDs können jedoch auch weiterhin über LOCK MODE verriegelt sein; diese Verriegelung wird über den vorliegenden Befehl ENABLE CONFIGURATION nicht beeinflusst.

Parameter:	<Configuration Time (Minutes):Byte> CONFIG TIME = $01...$F0, $00=OFF, $FA=Always (immer EIN)
GET:	Aufruf ohne Parameter
Rückgabe:	1 Byte: derzeitige Einstellung bzw. Restzeit
SET:	Aufruf mit Parameter: 1 Byte (CONFIG TIME: Byte)
Rückgabe:	Status

Der Einstellwert muß in Minuten angegeben werden, d.h., zur Einstellung von 20 Minuten muß der Datenwert 20 (14*hex*) eingegeben werden. Der Einstellbereich ist 1 Minute (01*hex*) bis 240 Minuten (F0*hex*).

Die Eingabe von 0 (00*hex*) schaltet die Konfiguration über RDM ab (das eingestellte Zeitintervall wird, wenn noch nicht abgelaufen, somit vorzeitig beendet).

Die Eingabe von 250 (FA*hex*) schaltet die Zeitsteuerung ab (immer enabled).

Die Zeitsteuerung ist re-triggerbar, d.h., auch wenn die aktuelle Zeit noch nicht abgelaufen ist, kann ein neues Intervall jederzeit neu gestartet werden (Verlängerung).

Bitte beachten Sie, daß viele RDM-Editoren zur Eingabe hexadezimale Werte erwarten- die Umrechnung kann mit einem geeigneten Rechner (z.B. Windows Calculator) erfolgen.

HINWEIS: Ist die Verriegelung aktiv (entweder weil die CONFIG TIME abgelaufen ist oder manuell auf „00“ gesetzt wurde), dann wird bei ***jedem*** SET-Zugriff die Meldung „WRITE PROTECTED“ ausgegeben. Der Schreibzugriff läßt sich ***nur*** durch diese PID - und nicht durch andere Schreibzugriff-Verriegelungen wie z.B. LOCK MODE - aufheben.

PID FF0E: SUBDEVICE ADDRESS Adressen der SubDevices

Diese PID erlaubt das Auslesen bzw. die Einstellung der Sub-Device-Adressen.

Parameter: <SubDevice-Nummer:Word>

GET: Aufruf ohne Parameter
Rückgabe: n SubDevice Adressen: n • Word

GET: Aufruf mit Parameter: <SubDevice-Nummer: Word>

Rückgabe: SubDevice Adresse: Word

SET: Aufruf mit Parameter: SubdeviceNummer: Word, SubDevice-Adresse: Word
Rückgabe: Status

Hinweise zu Parametern:
-SubDevice-Nummer: 1...n (0001...00nn hex) oder FFFFhex (alle)
-SubDevice Adresse: 1 bis 512 (0001....0200hex)

n= Anzahl der verfügbaren SubDevices

PID FF0F: SUBDEVICE ENABLE Auf SubDevice Modus umschalten

Dieser Modus schaltet vom Root-Modus in den SubDevice-Modus (und zurück).
Je nach verwendetem RDM Controller kann es erforderlich sein, das Gerät nach der Umschaltung neu auszulesen.

Parameter:	<MODE:Byte>
	MODE = $FF (255): SubDevice Mode ON
	MODE= $00 (0): SubDevice Mode OFF
GET	Aufruf ohne Parameter
Rückgabe:	1 Byte: derzeitige Einstellung
SET:	Aufruf mit Parameter: 1 Byte (MODE: Byte)
Rückgabe:	Status

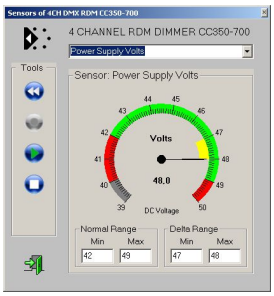
.

PID FF7F: RDM TESTAUSGABE

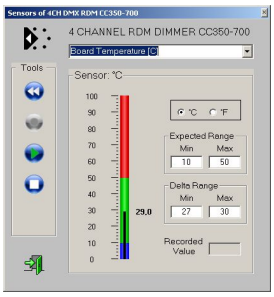
Diese Funktionen dienen der werksseitigen Einstellung.
Sie sind nicht für Anwender vorgesehen und werden daher hier nicht dokumentiert.

Sensoren

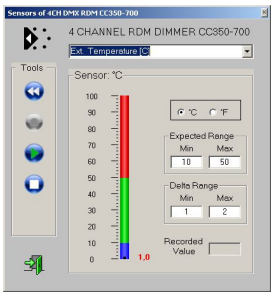
Viele Geräte verfügen über einen oder mehrere Sensoren, die über DMX RDM abgefragt werden können. Hier einige Beispiele:



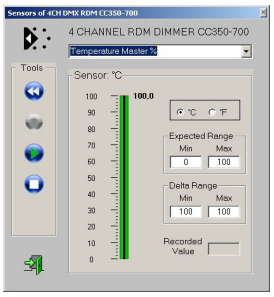
Sensor 1
Spannungsversorgung



Sensor 2
Elektronik-Temperatur



Sensor 3
LED-Temperatur



Sensor 4
Temperatur-Master %

Sensoren werden beim GET/SET Controller automatisch mit einer der Sensortype angepaßten Maske ausgegeben.

Manche Sensoren erfassen den Schwankungsbereich mit einer separaten Anzeige; die Erfassung kann beim JESE GET/SET Controller durch den Button „<<“ jederzeit zurückgesetzt werden. Die laufende periodische Sensorabfrage wird durch den Button „>“ gestartet.

Die Maske für die LED-Temperatur kann durch die Temperaturmanagement-Einstellungen angepaßt werden. Sie wird bei der Discovery automatisch geupdated.

Weitere RDM Info

Weitere Informationen zu DMX RDM, RDM-Einstellungen, den speziellen RDM-Befehlen u.s.w. finden Sie auf unserer Website ***www.rdm.DMXRDM.de***.

Anhang

PID	NAME	LOCK1	LOCK2
	E1.20 and E1.37 commands		
0015	COMMS STATUS	-	X
0082	DEVICE_LABEL	-	-
0090	FACTORY_DEFAULTS	-	-
00E0	PERSONALITY	X	-
00F0	START_ADDRESS	-	-
0121	SLOT_LABELS	-	-
0140	BLOCK_ADDRESS	-	-
0141	FAIL_MODE	X	-
0201	SENSOR_RESET	-	-
0341	MIN_LEVEL	-	X
0342	MAX_LEVEL	-	X
0343	CURVE	-	X
0345	OUTPUT_RESPONSE	-	X
0347	MODULATION_FREQUENCY	-	X
0603	REAL_TIME_CLOCK	-	-
0640	PIN	X	X
0641	SET_LOCK_STATE	-	-
1001	RESET	-	-
1010	IDENTIFY	-	-
1040	IDENTIFY_MODE	-	-
	DMX Resolution		
80E3	16BIT_RECEIVE MODE	-	X
	DMX Slot Count		
80C1	SET_SLOTCOUNT	-	X
	DMX HOLD Mode		
80F1	DMX_HOLD	X	-
80F2	MASTER_HOLD	X	-
80F3	MASTER_CHANNEL	-	X
80F4	MASTER_TABLE	-	X
80F5	MASTER_CHANNEL	-	X
	Slot Labeling		
8121	SLOT_LABELS	-	X
	E1.20 and E1.37 commands		
8341	MIN_MAX_MODE	-	-
	Sensor Configuration		
8400	SENSOR_DEFINITION_DATA	X	X
8401	SENSOR_LIMITS	X	X
	Slot Configuration		
8403	OUTPUT_CONFIGURATION	-	X
8433	DMX_FOOTPRINT	-	X
	TimerSetup		
9002	TIMEBASE	-	X
9003	TRIGGER_LEVEL	-	X
9004	TRIGGER_COUNT	-	X
	MotorDriver		
C001	STEP_WIDTH	-	X
C002	PWM_FACTOR	-	X
C003	LOWER_LIMIT	-	X
C004	UPPER_LIMIT	-	X
C005	ENDSWITCH_POLARITY	-	X
	Relay Properties		
C0C0	PATCHING	-	X
C0E0	POLARITY	-	X
C0F0	MONOSTABLE_TIME	-	X
C0F1	EXCLUSIVE_MODE	-	X
C0F3	SAFETY_MODE	-	X
C0F4	SAFETY_DELAY	-	X
C0F5	DELAY	-	X

	Output Configuration		
DC01	OUTPUT_CURRENT	X	X
DC0E	DC_OFFSET	X	X
DC0F	OUTPUT_OFFSET	X	X
DC10	DC_SUPPRESS	X	X
DC1F	16BIT_OFFSET	X	X
DCCA	DC_CALIBRATION	-	X
DCCD	USER_CURVE	-	X
DCCE	BENDEC_CURVE	-	X
	Temperature Management		
E001	LOWER_TRIP_POINT	-	X
E002	UPPER_TRIP_POINT	-	X
E003	ALARM_TRIP_POINT	-	X
E004	SAMPLE_TIME	-	X
E005	TEMP_DROPLEVEL	-	X
E00E	TEMP_ENABLE	-	X
	Device Setup		
FF01	FACTORY_SETUP	-	-
FF02	CALIBRATION	-	X
	SubDevices		
FF0E	SUBDEVICE_STARTADDRESS	-	X
FF0F	SUBDEVICE_ENABLE	X	X

GET/SET CONTROLLER

Wir bieten Ihnen mit dem USBRDM-TRI ein Interface an, das komplett mit einer RDM-Controller-Applikation kommt und bestens geeignet ist, um alle unsere DMX RDM Interfaces (siehe: RDM Interfaces) zu verwalten. Mit dem Interface erhalten Sie eine Installations-CD mit Gerätetreibern und die RDM Controllersoftware "GET/SET" für Windows 7,8, und Windows 10. Das Interface und die Software kann auf mehreren Computern installiert werden. Es wird nur dann aktiviert, wenn es angesteckt ist.

Die Besonderheit des USBRDM-TRI ist die interne Signalverbeitung, die die RDM-Kommunikation entlastet und beschleunigt. Laufende Updates der Controller-Firmware und der Applikations-Software sind enthalten; neue Versionen können jederzeit einfach von der Hersteller-Website gedownloaded werden. Der benötigte Update-Link ist in der Controller-Software enthalten.

Das USB-RDM TRI wird mit der GET/SET MV Controller Software gebundlet ausgeliefert. Da das Interface so wohl DMX empfangen als auch DMX senden kann, kann es einfach in eine DMX Leitung *eingeschleift* werden. Jede vorhandene DMX Steuerung -z.B. Ihr vorhandenes Lichtsteuerpult- kann so einfachst um volle RDM Funktionalität erweitert werden. Einfacher geht es nun wirklich nicht!

Mehr Infos auf:
www.DMXRDM.de/produkte/usbrdm-tri2