



## **BEDIENUNGSANLEITUNG DMX RDM**

(C) SOUNDLIGHT 1996-2017 \* ALLE RECHTE VORBEHALTEN \* KEIN TEIL DIESER ANLEITUNG DARF OHNE SCHRIFTLICHE ZUSTIMMUNG DES HERAUSGEBERS IN IRGENDEINER FORM REPRODUZIERT, VERVIELFÄLTIGT ODER KOMMERZIELL GENUTZT WERDEN. \* WIR HALTEN ALLE ANGABEN DIESER ANLEITUNG FÜR VOLLSTÄNDIG UND ZUVERLÄSSIG. FÜR IRRTÜMER UND DRUCKFEHLER KÖNNEN WIR JEDOCH KEINE GEWÄHR ÜBERNEHMEN. VOR INBETRIEBNAHME HAT DER ANWENDER DIE ZWECKMÄSSIGKEIT DES GERÄTES FÜR SEINEN GEPLANTEN EINSATZ ZU PRÜFEN. SOUNDLIGHT SCHLIESST INSbesondere JEDE HAFTUNG FÜR SCHÄDEN -SOWOHL AM GERÄT ALS AUCH FOLGESCHÄDEN- AUS, DIE DURCH NICHTEIGNUNG, UNSACHGEMÄSEN AUFBAU, FALSche INBETRIEBNAHME UND ANWENDUNG SOWIE NICHTBEACHTUNG GELTENDER SICHERHEITSVORSchriften ENTSTEHEN.

## Vielen Dank, daß Sie sich für ein SOUNDLIGHT Gerät entschieden haben.

Fast alle SOUNDLIGHT 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 möchten. 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. SOUNDLIGHT-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](http://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

Rechner

PROGRAMMIERER

244

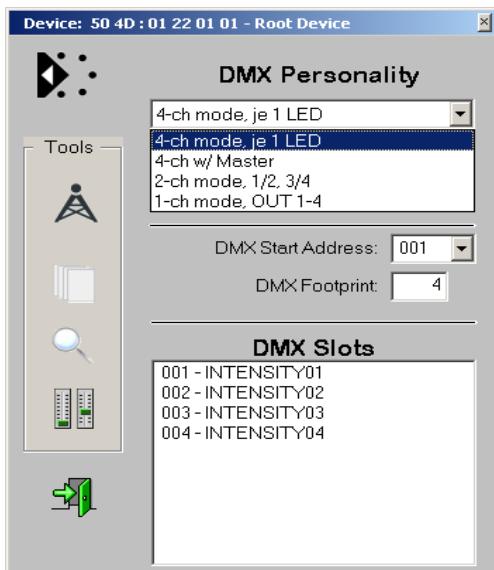
HEX F4  
DEC 244  
OCT 364  
BIN 1111 0100

Lsh Rsh Or Xor Not And Mod CE C = ÷ × − + ± 0 . =

Alle wichtigen Informationen haben wir auf unserer RDM Website zusammengetragen. Sie erreichen sie unter <http://www.rdm.soundlight.de>. Sie finden dort auch zahlreichen Anwendungsbeispiele, und auch die Umsetzung von RDM-Befehlen auf verschiedene Controller wird gezeigt.

## Datenformate

RDM-Befehle selbst verwenden hexadezimale Notation, d.h., alle Kommandos werden mit Hexadezimalzahlen (0...9,A,B,C,D,E,F) beschrieben. Es gibt jedoch auch Controller, die stattdessen dezimale Eingaben erwarten. Hier muß der Operator gegebenenfalls passend umcodieren; der Windows Taschenrechner (siehe links) kann auf Hexadezimale Eingabe umgestellt werden und ist dabei als Eingabehilfe recht nützlich. 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.*

## 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.

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. SOUNDLIGHT hat die Herstellerkennung „SL“, dargestellt durch die Hexadezimalwerte „53 4C“ (siehe: [http://tsp.estas.org/tsp/working\\_groups/CP/mfctrlIDs.php](http://tsp.estas.org/tsp/working_groups/CP/mfctrlIDs.php)). Danach vergeben wir die Gerätetypen und die Seriennummer.

Die UID „53 4C 36 03 12 34“ heisst also: dies ist ein Gerät von SOUNDLIGHT (53 4C), Gerätetyp 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 Gerätetyp, 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 können der Norm entnommen werden.

# PIDs und Befehle

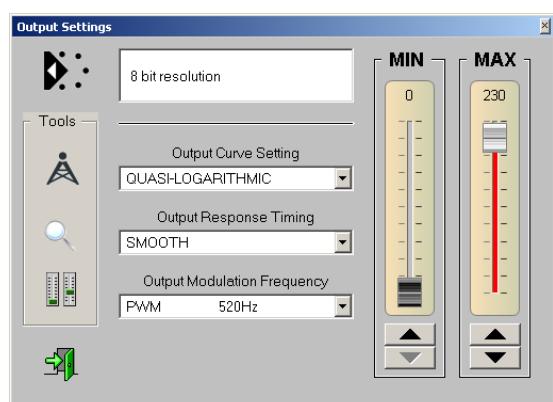
PIDs sind in vier Gruppen eingeteilt:

## 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 diesel-

be 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).

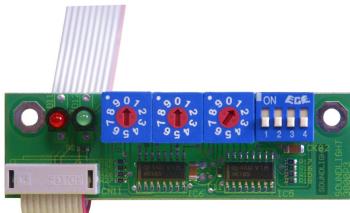
#### 4. PLASA / ESTA Kommandos

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 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 (dan kann, muss aber nicht angeschlossen bleiben, weil alle eingestellten Parameter in den Speicher des Grundgerätes übernommen werden).

Als Adressboards stehen verschiedene Versionen zur Verfügung:



Startaddressboard 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.



Startaddressboard 3006P:

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



#### TIPP:

Da Schalter stets auf einer festen Einstellung stehen, die sich von einer über DMX RDM getroffenen 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 dann. 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 oder 3006P setzen. Adressboards sind als optionales Zubehör separat erhältlich.

# DMX RDM

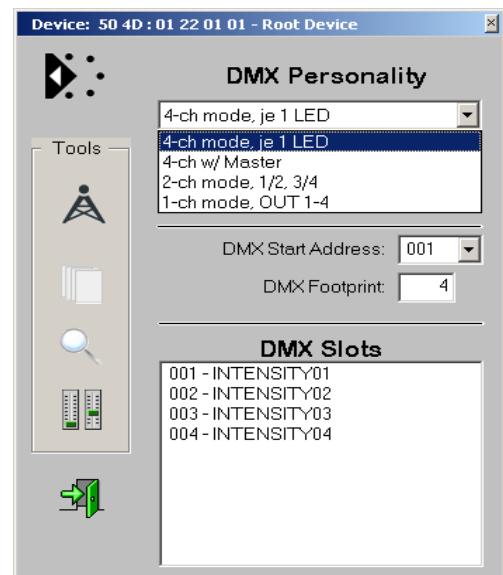
Die nachfolgenden Kommandos beziehen sich auf den Draft Standard ANSI E1-37 "Additional Commands for RDM". Dieser Standard wurde erst kürzlich ratifiziert. DMX-RDM Controller können also gegebenenfalls noch nicht über die zugehörigen Funktionsnamen verfügen. Über geeignete RDM Controller (z.B. Enttec 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.rdm.soundlight.de](http://www.rdm.soundlight.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 Controller 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 Controller automatisch die vom jeweiligen Responder benötigte Kanalzahl ab; für ein 4-Kanal Gerät ist damit die höchste einstellbare Startadresse 509.



## Spezielle Kommandos

Diese Kommandos umfassen sowohl Kommandos aus dem RDM Ergänzungsstandard als auch herstellerspezifische Kommandos, die in SOUNDLIGHT 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. 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 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.

<b>PID 8081: DMX HOLD INPUT 1</b>	Verhalten bei DMX Signalausfall DMX Eingang 1
<b>PID 8082: DMX HOLD INPUT 2</b>	Verhalten bei DMX Signalausfall DMX Eingang 2
<b>PID 8083: DMX HOLD INPUT 3</b>	Verhalten bei DMX Signalausfall DMX Eingang 3
<b>PID 8084: DMX HOLD INPUT 4</b>	Verhalten bei DMX Signalausfall DMX Eingang 4
<b>PID 80F1: DMX HOLD MODE</b>	Verhalten bei DMX Signalausfall DMX Eingang allgemein
<b>PID 80F2: MASTER HOLD MODE</b>	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)

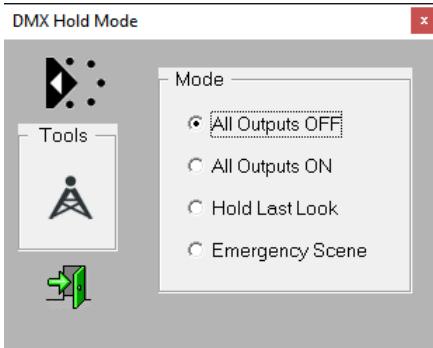
GET: Aufruf ohne Parameter  
Rückgabe: Parameter: 1 Byte (HOLD-Modus)

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 letzten gültigen Wert bei „Keep Last Look“
\$03	3	Eine voreingestellte Szene wird aufgerufen*

\*=nicht bei allen Modellen verfügbar



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

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

**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

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

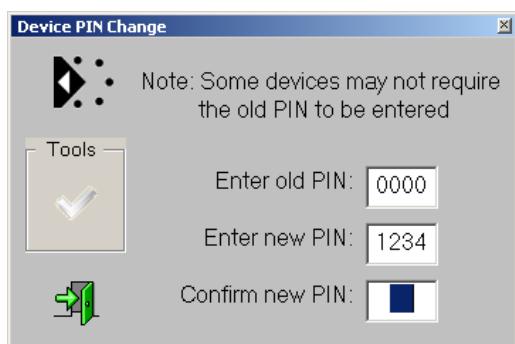
Mit Auslieferung ist das Gerät entriegelt und die Start-PIN ist 0000 (Hex 0000). Die Funktion erlaubt nur eine SET-Eingabe, keine Auslesung über GET. Die PID 0640 erlaubt die Konfiguration über User-Maske, die PID 8330 hingegen über Eingabe im Hexadezimal-Modus (bzw. Dezimalmodus, je nach Controller).

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!

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

### Verriegelungszustand

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:

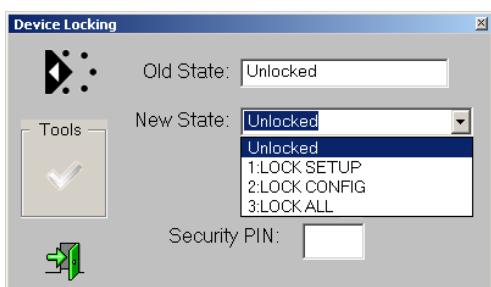
- nichts verriegelt: 0003
- Setup verriegelt: 0103
- Config verriegelt: 0203
- Alles verriegelt: 0303

**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 bei 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 das 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**  
**PID 8332: LOCK STATE DESCRIPTION**

### Ausgabe einer Beschreibung für die Verriegelung

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**  
**PID 8340: IDENTIFY MODE**

### Identify-Modus

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

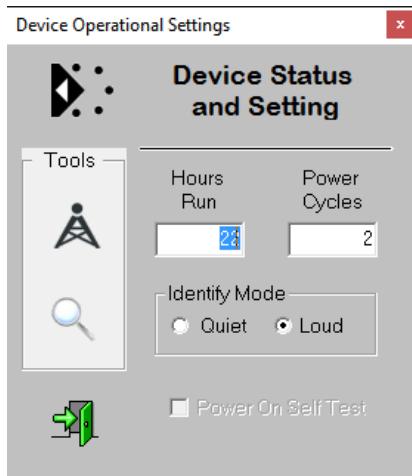
#### Identify-Modus:

Hex	Dezimal
00	0

Funktion  
Quiet Mode: Identify-Ausgabe nur auf den Anzeige-LEDs

FF	255
----	-----

Loud Mode: Identify über die Ausgänge



Identify Mode Einstellung beim GET/SET Controller

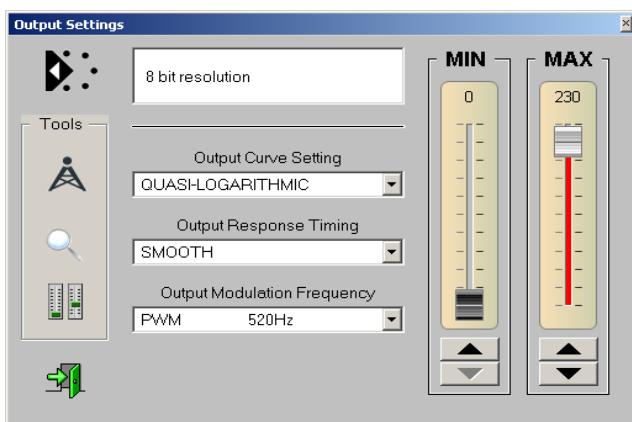
**PID 0341: MINIMUM LEVEL**  
**PID 0342: MAXIMUM LEVEL**

## 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 automatisch 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 das Ansteuersignal unter den Minimalwert fällt).

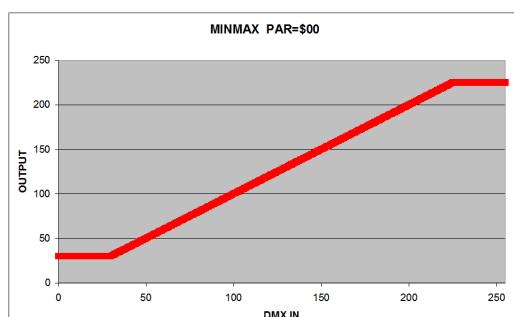
Die Pegelwerte werden stets als 16-Bit-Werte angegeben. Wenn der Pegelbereich des Gerätes kleiner als 16 Bit ist, werden nur die oberen n Bit berücksichtigt. Die nutzbare Auflösung muss in der PID DIMMER INFO deklariert werden!

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

Dabei gelten grundsätzlich folgende Einstellungen:

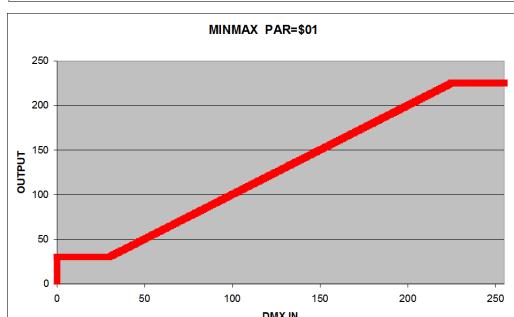
Der Ausgangspegel wird auf den eingestellten Minimal- und Maximalpegel begrenzt. Die Begrenzung tritt in Kraft, sobald der eingestellte Pegel durch den DMX Ansteuerwert erreicht und überschritten (bzw. unterschritten) wird. Danach ändert sich der Ausgangspegel nicht mehr.

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

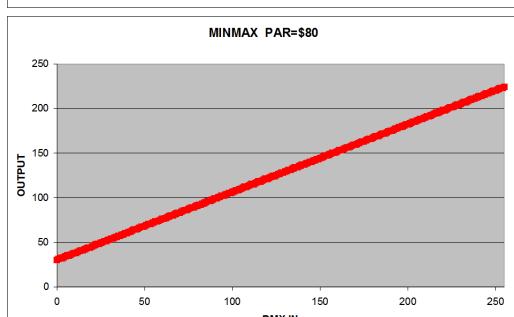


PARAMETER: 1 Byte

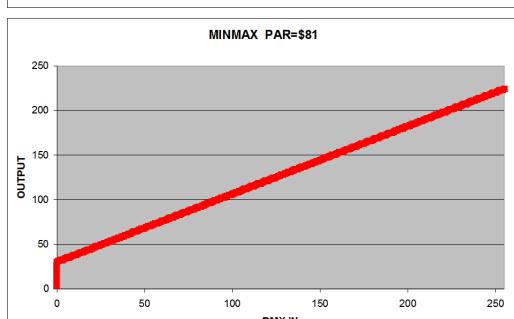
Parameter = \$00 (0 dez):  
Begrenzung als Standardfunktion



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 kontinuierlichen Fade vom MIN LEVEL bis zum MAX LEVEL ersetzt. Dadurch wird der Fade-Bereich weiter.

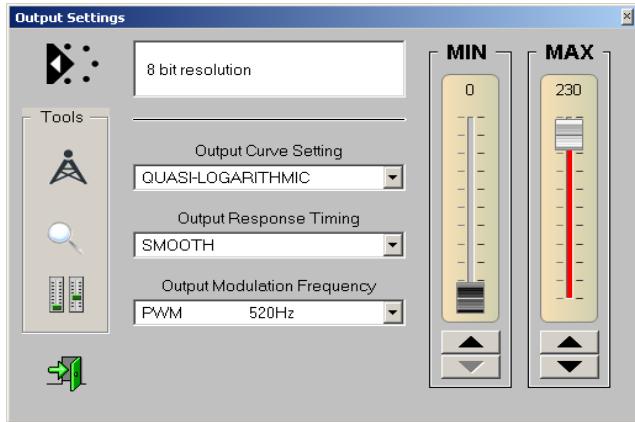


Parameter = \$81 (129 dez):  
Die Parameter können auch kombiniert werden.

## 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 Kennlinie 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

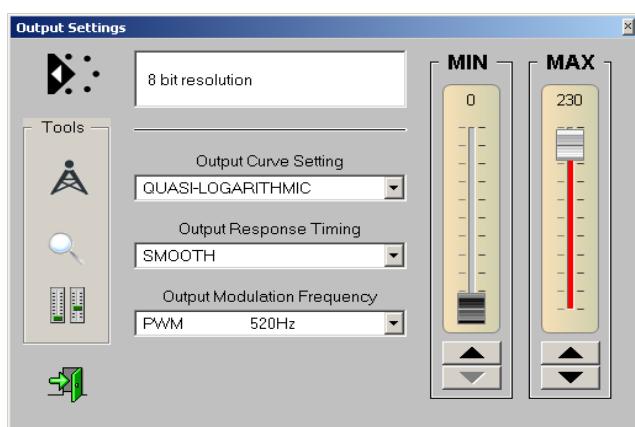
Die Eingabe der Curve-Nummer bei einer GET-Abfrage returniert die Beschreibung der Kurve. Ein SET-Kommando ist hier nicht möglich.

Das Ausgangsverhalten (Regelgeschwindigkeit) ist mit der Funktion OUTPUT RESPONSE TIME wählbar. Die Voreinstellung ist "SMOOTH" für optimal weiche Ausgabe. Bei 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



Langsam  
es Nach-  
ziehen des  
Ausgan-  
ges.

*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 Time kann man also die Anzahl der möglichen Einstellungen in Erfahrung bringen, und die namentliche Zuordnung durch die 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.

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 enable / 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 enabled worden sind, können sich die angezeigten Slot Labels bereits verändert haben. Sie werden nun nicht mehr aus den Voreinstellungen 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

Rückgabe:

Aufruf mit Parameter: <SlotLabel Nr:Word>

18 Bytes: <SlotLabel Nr:Word>

<SlotLabel Text: 16 Bytes>

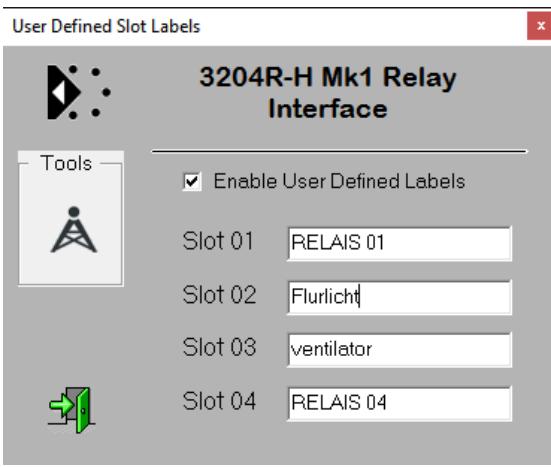
SET:

Rückgabe:

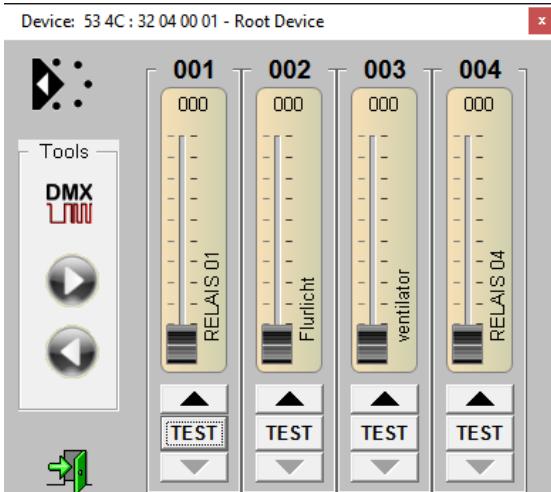
Aufruf mit Parameter: 18 Bytes

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

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 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). 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). 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). 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 FAHRBEREICHSGRENZE

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). 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 FAHRBEREICHSGRENZE

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). 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: POSITIONING DECEL ABBREMS-ZONE BEI POSITIONIERUNG RANGE

## PID C008: INIT SPEED

## INITIALISIERUNGS-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 (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 auszuprobiieren. 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 Aufruf ohne Parameter  
Rückgabe: 1 Byte: derzeitige Einstellung

SET: Aufruf mit Parameter: 1 Byte  
(ACCURACY\_BITS: Byte)  
Rückgabe: 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 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 Rückgabe:	Aufruf ohne Parameter n Bytes (n= Anzahl der Kanäle) Byte 1-n: derzeitige Einstellung		
GET Rückgabe:	Aufruf mit Parameter: 2 Bytes (Kanalnummer, 16 Bit) 1 Byte Byte 1: derzeitige Einstellung		
SET: Rückgabe:			Aufruf mit Parameter: 3 Bytes (Kanalnummer, 16 Bit und Einstellung 1 Byte (8 Bit) Status
Polarity:	00	FF	invertiert normal

## Relaisfunktionen beim Detektor-Betrieb

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

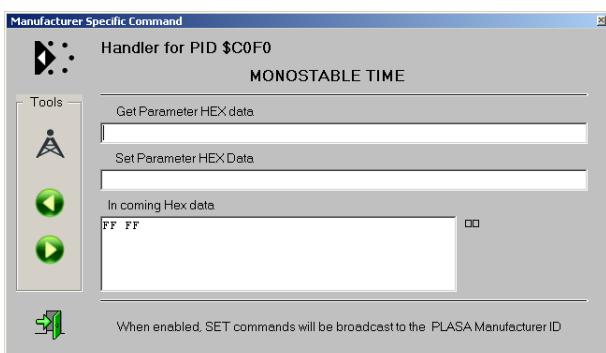
Kanalnummer = 0002: Kanal 2

....

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

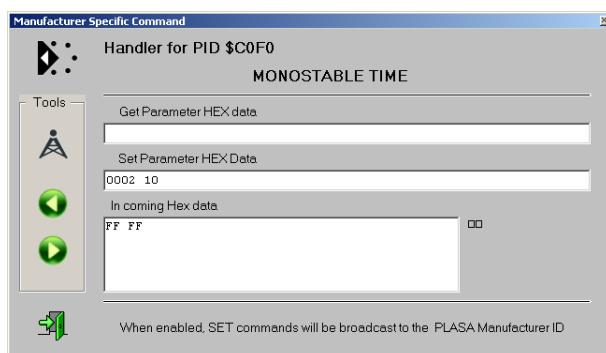
GET Aufruf mit Parameter: 2 Bytes (Kanalnummer, 16 Bit)

Rückgabe: 1 Byte : derzeitige Einstellung

SET Aufruf mit Parameter: 3 Bytes

(Kanalnummer, 16 Bit und Einstellung 1 Byte (8 Bit)

Rückgabe: Status

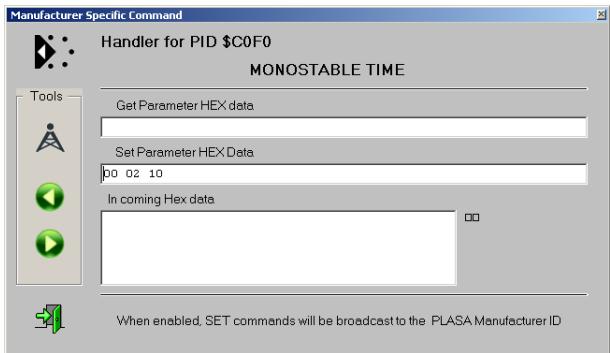


Um Relais 2 auf monostabilen Modus mit einer Impulszeit von 0,4 Sekunden (400ms) zu setzen, geben Sie ein:

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. Das 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

**PIDC0F1: EXCLUSIVE MODE****Relais schalten gegenseitig verriegelt**

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 Exklusivverknüpfung jeweils für die Relaispaare:

- Relais 1 und Relais 2

- Relais 3 und Relais 4

- Relais 5 und Relais 6

...etc...

**PIDC0F2: 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

**PIDC0F3: 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:            1 Byte: derzeitige Einstellung

SET:                 Aufruf mit Parameter: 1 Byte  
                      (SAFETY\_MODE: Byte)  
Rückgabe:            Status

**PIDC0F4: SAFETY DELAY****Auf SAFETY-Delay umschalten**

Dieser Modus schaltet eine zusätzliche Auslöseverzögerung zu. Das Safety Delay wird in 25ms Schritten spezifiziert.

Parameter: <CHANNEL: word><SAFETY\_DELAY:Byte>  
SAFETY\_DELAY = \$01...\$FE

GET Aufruf ohne Parameter  
Rückgabe: n Bytes: derzeitige Einstellung aller Ausgänge

SET Aufruf mit Parameter: 3 Byte  
<CHANNEL: word><SAFETY\_DELAY:Byte>  
Rückgabe: Status

**PID E001: LOWER TEMP TRIP**  
**PID E002: UPPER TEMP TRIP**  
**PID E003: ALARM TEMP TRIP**  
**PID E004: TEMP SAMPLE TIME**  
**PID E005: TEMP DROP LEVEL**

**UNTERER TEMPERATUR-REGELPUNKT**  
**OBERER TEMPERATUR-REGELPUNKT**  
**ALARMTEMPERATUR-SCHALTPUNKT**  
**TEMPERATUR-SAMPLEZEIT**  
**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.*

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 Ausgabepiegel 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.*

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 heruntergeregt 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)

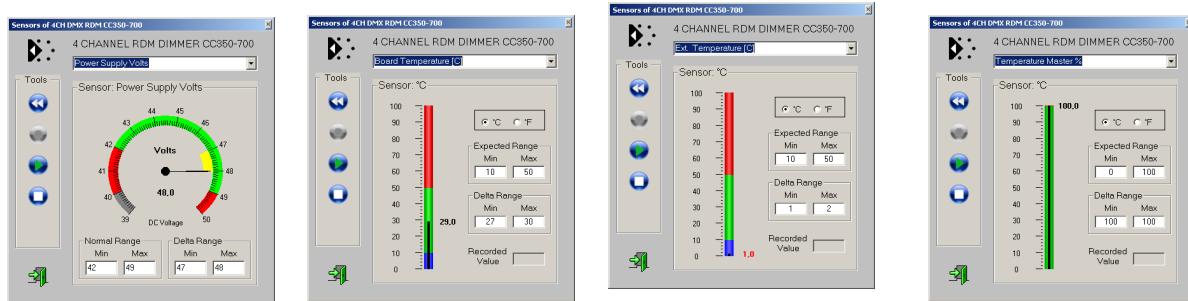
Die Zuordnung ist dem jeweiligen Gerätemanual zu entnehmen.

## PID 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 durch den Button „<<“ jederzeit zurückgesetzt werden. Die laufende 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.soundlight.de](http://www.rdm.soundlight.de).

# Anhang

PID	NAME	LOCK1	LOCK2
<b>E1-20 and E1-37 commands</b>			
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	-	-
<b>DMX Resolution</b>			
80E2	16BIT_MODE	-	X
<b>DMX Slot Count</b>			
80C1	SET_SLOTCOUNT	-	X
<b>DMX HOLD Mode</b>			
80F1	DMX_HOLD	X	-
80F2	MASTER_HOLD	X	-
80F3	MASTER_CHANNEL	-	X
80F4	MASTER_TABLE	-	X
80F5	MASTER_CHANNEL	-	X
<b>Slot Labeling</b>			
8121	SLOT_LABELS	-	X
<b>E1-20 and E1-37 commands</b>			
8341	MIN_MAX_MODE	-	-
<b>Sensor Configuration</b>			
8400	SENSOR_DEFINITION_DATA	X	X
8401	SENSOR_LIMITS	X	X
<b>Slot Configuration</b>			
8403	OUTPUT_CONFIGURATION	-	X
8433	DMX_FOOTPRINT	-	X
<b>TimerSetup</b>			
9002	TIMEBASE	-	X
9003	TRIGGER_LEVEL	-	X
9004	TRIGGER_COUNT	-	X
<b>MotorDriver</b>			
C001	STEP_WIDTH	-	X
C002	PWM_FACTOR	-	X
C003	LOWER_LIMIT	-	X
C004	UPPER_LIMIT	-	X
C005	ENDSWITCH_POLARITY	-	X
<b>Relay Properties</b>			
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

<b>Output Configuration</b>			
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
<b>Temperature Management</b>			
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
<b>Device Setup</b>			
FF01	FACTORY_SETUP	-	-
FF02	CALIBRATION	-	X
<b>SubDevices</b>			
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 Signalverarbeitung, 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 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 kann so einfachst um volle RDM Funktionalität erweitert werden. Einfacher gehts es wirklich nicht!

Mehr Infos auf:  
[www.soundlight.de/produkte/usbrdm-tri2](http://www.soundlight.de/produkte/usbrdm-tri2)

