

8 DMX 512 - RDM

Die in Abschnitt 4 vorgestellte Hardware kann durch den *GPIO18* zwischen Lese- und Schreibfunktionalität umgeschaltet werden. Diese Funktion ermöglicht es theoretisch, die *Remote Device Management* (RDM) Erweiterung des DMX Protokolls zu nutzen, womit sich der folgende Abschnitt näher befasst.

RDM wird im Live-Bereich der Veranstaltungstechnik sehr selten bis nicht verwendet und grundsätzlich während der Show deaktiviert. Da dieses Feature nur von hochpreisigen Endgeräten angeboten wird, ist die Wahrscheinlichkeit groß, dass sich ebenfalls Geräte ohne RDM im Strang befinden.

Durch den Zugriff mehrerer Sender auf den gleichen Bus muss ein passendes Buszugsverfahren gewählt werden, bei dem die rückmeldenden Pakete den Datenfluss vom Mischpult an die Endgeräte nicht stört. Der Reset zur Einleitung jeder Übertragungssequenz setzt mithilfe des *Carrier Sense Multiple Access / Collision Avoidance* (CSMA/CA) Zugriffsverfahrens eine Priorität, sodass alle Endgeräte ihre Übertragung im Sinne der Kollisionsvermeidung abbrechen und den Bus freigeben.

Dies setzt zusätzlich voraus, dass jedes Endgerät in der Lage ist, einen vom Lichtpult gesendeten Reset zu detektieren. Alte oder nicht standardkonform programmierte Geräte können hier möglicherweise aufgrund falscher Interpretationen ein Fehlverhalten aufweisen.

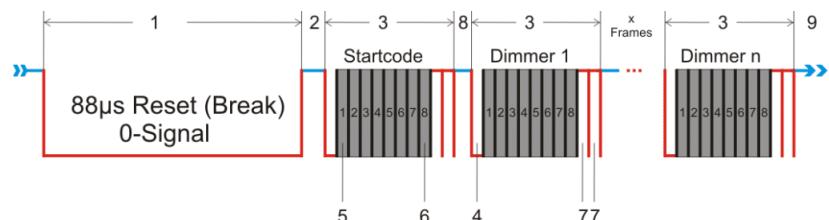


Abbildung 1: Timing bei DMX

Quelle: <https://wiki.production-partner.de/licht/lichtsteuerung-mit-dmx-512/>

8.1 Aufbau der RDM Pakete

Die Kommunikation im RDM-Fall ist vergleichbar mit I^2C . Das Lichtpult fungiert als Master und fragt gezielt eine Information an einer Lampe an, welche daraufhin antwortet. Zur Adressierung wird eine 48 Byte UID verwendet, welche stark an die MAC-Adresse bei Ethernet erinnert. Die ersten 16 Byte geben Aufschluss über den Hersteller, die restlichen 32 Byte können individuell vergeben werden.

Die UID FF.FF.FF.FF.FF.FF wird als Broadcast verwendet.

Um eine eindeutige Unterscheidung zwischen DMX- und RDM-Paketen zu gewährleisten wird als abweichender Startcode 0xCC anstelle 0x00 verwendet. Bei einer korrekten

Implementierung des DMX Protokolls sollte das empfangene Paket verworfen werden, wenn keine RDM Funktionalität im Endgerät verfügbar ist.

An den Startcode wird der sogenannte Substartcode Byte 0x01 angehängt. Dieses dient als Platzhalter für zukünftige Protokollerweiterungen.

Befehle und Antworten setzen sich aus drei Byte zusammen. Das erste Byte ist die Befehlsklasse, siehe Tabelle 1 die anderen beiden die Befehls-ID, siehe ausschnittsweise Tabelle 2.

Klasse	Funktion
0x10	Discovery Befehl
0x11	Discovery Antwort
0x20	Leseanforderung
0x21	Antwort mit angefordertem Wert
0x30	Schreibbefehl
0x31	Antwort auf Schreibbefehl

Tabelle 1: Befehlsklassen bei RDM

Quelle: https://www.soundlight.de/techtips/dmx512/dmx_rdm.htm

Befehls-ID	Befehl
0x0001	Discovery
0x0002	Mute für Discovery
0x0003	Unmute für Discovery
0x0050	Liste unterstützter Parameter
0x0051	Parameterbeschreibung
0x0060	Geräteinformationen
0x0081	Herstellername
0x00C0	Softwareversion
0x0400	Betriebsstunden
0x0401	Lampenstunden
0x0402	Lampenzündungen
0x0403	Lampenstatus

Tabelle 2: BefehlsID bei RDM

Quelle: https://www.soundlight.de/techtips/dmx512/dmx_rdm.htm

Durch die Mute- bzw. Unmute Funktion wird der Buszugriff zusätzlich unterstützt. Erhält man auf einen Discovery-Broadcast eine Antwort, kann aus dieser die UID des Endgerätes entnommen werden. Wird im Anschluss gezielt an dieses Gerät ein Mute-Befehl gesendet, antwortet es nicht auf künftige Discovery-Broadcasts.

Da alle Hersteller unterschiedliche Befehle unterstützen oder teilweise umfunktionieren, ist es erschwert, einen einheitlichen Monitor oder Interpreter zu entwickeln. Der

individuelle Funktionsumfang muss lampenspezifisch der Anleitung entnommen werden. Der Lampenhersteller Robe beschreibt im Handbuch des BMFL Blade die BefehlsIDs *ROBE_DMX_INPUT* oder *ROBE_WIRELESS_UNLINK* und gibt ebenfalls in dieser Liste durch eine Fußnote an, andere Optionen für gewisse BefehlsIDs zu verwenden. Quelle: https://www.robelighting.de/res/downloads/user_manuals/User_manual_Robin_BMFL_Bla...

8.2 Praxistest

Die Veranstaltungssicherheit und -technik hat dankenswerterweise den Audimax der Technischen Universität Kaiserslautern zur Verfügung gestellt, um das RDM Protokoll in der Praxis testen zu können. Hierzu wurde die bestehende Verkabelung aufgelöst, da sich zwischen Regiekabine und Beleuchterzug ein nicht-RDMfähiger DMX-Splitter befindet.

Im Beleuchterzug wurden zwei RDM kompatible Robe BMFL Spot und vier Eurolite ML56 LED Par ohne RDM Funktion verbunden. Nach dem Aktivieren der RDM Funktion am grandMA 2 Steuerpult konnte hier direkt ein Fehlverhalten der ML56 beobachtet werden, welches sich ähnlich wie ein Busfehler zeigte. Der Versuchsaufbau wurde daraufhin so verändert, dass lediglich die beiden BMFL an die grandMA angeschlossen worden sind. Eine erste Spekulation ist die Missinterpretation des geänderten Startcodes von 0xCC für RDM anstelle des standardmäßigen 0x00.

Im veränderten Aufbau wurden beide BMFL nach einiger Zeit korrekt detektiert. Auch wenn die grandMA einen limitierten Umfang der eigentlichen Möglichkeiten unterstützt konnten einige Sensorwerte gelesen und händisch interpretiert werden. Alternativ wäre mit dem Robe RDM Communicator eine einfache herstellerbundene Möglichkeit gegeben, fertig interpretierte Werte zu erhalten.

Der Testaufbau hat gezeigt, dass es in diesem Fall nicht möglich ist, beliebige Lampen in einer einzigen DMX-RDM-Kette zu betreiben. Würde man im Audimax die RDM-fähigkeit der BMFL nutzen wollen, müsste ein weiterer DMX-Strang in die Beleuchterzüge eingebracht werden. Dieser Aufwand ist im Vergleich zu einer Ethernetverkabelung so verschwindend gering, dass es wirtschaftlich nicht zu vertreten wäre, DMX statt Ethernet zu verwenden. Ethernet bietet im Vergleich zudem weitere Vorteile und Möglichkeiten wie beispielsweise eine Überwachung der Lampen über *Simple Network Monitoring Protocol* (SNMP) oder das Verbinden weiterer Geräte über trunk Ports. Da eine Ansteuerung der Lampen aus der bestehenden Peripherie über sACN oder ArtNet problemlos möglich ist, entfällt die Notwendigkeit der klassischen DMX Verkabelung.

9 Hardware- und Softwareanpassung

Der in der ursprünglichen Projektarbeit beschriebene DMX-Hat für den Raspberry Pi wurde modifiziert und ist verfügbar unter https://git.agg-hosting.de/Erwin/DMX-Hat_RasPi

Einige Hardwareanpassungen wie eine zusätzliche galvanische Trennung des DMX Ausgangs, eine Erhöhung der Busspannung auf 5V und eine Montagemöglichkeit für einen 20mm Lüfter wurden einbezogen.

Die aktuelle Platine ist als Projekt für KiCad 6.0 verfügbar.

Softwareseitig wurde ein Lesen für eingehende Pakete implementiert, eine Auswertung wurde aufgrund der sehr Herstellerspezifischen Interpretationsweisen von RDM nicht durchgeführt.

10 Fazit

Insbesondere der Praxistest hat gezeigt, dass das Remote Device Management zwar viele Funktionen bietet, welche das Leben erleichtern, bei inhomogenen Systemen aber oft zu Fehlern führt. Gepaart mit einer Wünsch-Dir-Was Mentalität der Hersteller wurde hier jegliche Innovation bereits im Keim erstickt.

Der erste Versuchsaufbau im Praxistest konnte musste bei einer überschaubaren Menge an Endgeräten abgebrochen werden, was bei einer Skalierung in große Systeme nicht unbedingt Besserung vermuten lässt. Auch wenn die Kommunikation im zweiten Versuch erfolgreich war, bleibt eine gewisse Skepsis gegenüber dem Protokoll. Im Live-Betrieb wäre eine Lampenüberwachung zwar wünschenswert, unterliegt aber der Störungsresistenz des Gesamtsystems.

Im Ausblick auf die nächsten Jahre verschwinden hoffentlich immer mehr Endgeräte, die aktuell nicht in der Lage sind, RDM von DMX Paketen zu unterscheiden. Mit ausbleibenden Störungen auf dem Bus könnte RDM allmählich Fuß fassen. Ein vollständiger Ersatz durch Ethernet ist jedoch nicht zu vernachlässigen.