



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Alexander Maßmann

Integration und Inbetriebnahme eines
DMX-Controllers als eigenständige
Lichtsteuerung eines
Flugzeugkabinen-Teststands

Alexander Maßmann

Integration und Inbetriebnahme eines
DMX-Controllers als eigenständige Lichtsteuerung
eines Flugzeugkabinen-Teststands

Bachelorarbeit eingereicht im Rahmen der Bachelorarbeitprüfung
im Studiengang Mechatronik
an der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr.-Ing. Mark Wiegmann
Zweitgutachter : Prof. Dr.-Ing. Thomas Kletschkowski

Abgegeben am 29. Januar 2018

Alexander Maßmann

Thema der Bachelorarbeit

Integration und Inbetriebnahme eines DMX-Controllers als eigenständige Lichtsteuerung eines Flugzeugkabinen-Teststands

Stichworte

DMX512, Butler XT, e:cue, LED Steuerung, DMX-Controller, Traxon Technologies

Kurzzusammenfassung

Diese Arbeit befasst sich mit der Integration und Inbetriebnahme eines DMX-Controllers als Lichtsteuerung eines Flugzeugkabinen-Teststands. Dazu gehört die Analyse des DMX512 Bussystems, die Aufarbeitung der Programmierung und Inbetriebnahme des Butler XT an einer RGB LED Matrix, die Analyse der vorhandenen Schnittstellen und Eingabemöglichkeiten, sowie die Integration des Systems in ein Airbus-Kabinen-Modell. Außerdem wird eine mögliche Implementierung eines Touchscreens für die Bedienung des Controllers untersucht.

Alexander Maßmann

Title of the paper

Commissioning and integration of an DMX-Controller as a standalone lighting control system for an aircraft cabin mockup

Keywords

DMX512, Butler XT, e:cue, LED control, DMX-Controller, Traxon Technologies

Abstract

This thesis deals with the integration and commissioning of an DMX-controller as lighting control for an aircraft cabin mockup. That includes the analysis of the DMX512 bus system, the processing of programming and commissioning of the Butler XT on an RGB LED matrix, the analysis of the existing interfaces and input options as well as the integration of the system into an Airbus cabin mockup. Additional to that the possibility of the implementation of a touchscreen for operating the controller will be developed.

Inhaltsverzeichnis

Tabellenverzeichnis	6
Abbildungsverzeichnis	7
1 Einleitung	9
1.1 Themenbeschreibung	9
1.2 Ziel der Arbeit	10
1.3 Struktur der Arbeit	10
2 Grundlagen	11
2.1 Serielle Bussysteme	11
2.1.1 Unterschied Seriell/Parallel	11
2.1.2 Synchrone/asynchrone Datenübertragung	13
2.1.3 Leitungsanzahl (single-ended/differential)	14
2.2 DMX-512 Datenbus	14
2.2.1 Entstehung	15
2.2.2 Hardware	15
2.2.2.1 Sende-und Empfangsschaltung	15
2.2.2.2 Leitungsverstärker	17
2.2.2.3 Datenkabel	18
2.2.3 Datenprotokoll DMX-512	19
2.2.4 Zusammenhang DMX-512 und RS-485	21
2.2.5 Steuerung von RGB-Lichtquellen	22
3 Butler XT DMX-Controller	23
3.1 allgemeine technische Daten	23
3.2 Schnittstellen	24
3.3 Stromversorgung	25
3.4 RDM (Remote Device Management)	27
3.5 e:bus	28
4 Inbetriebnahme	30
4.1 Hardwareaufbau	30

4.2	Programmierung	32
4.2.1	Einrichten des Butler XT	32
4.2.1.1	Allgemeingültige Einrichtung	33
4.2.1.2	Showspezifische Einrichtung	34
4.2.2	Patchfile	36
4.2.3	Show-Programmierung	38
4.2.3.1	Allgemeines	39
4.2.3.2	Freie Programmierung	42
4.2.3.3	Programmierung mit Wizards	44
4.2.3.4	LiveFX Effektprogrammierung	48
4.3	Einbinden eines Touch-Bedienfeldes	50
4.3.1	RS232-Bedienfeld	50
4.3.2	e:bus-Bedienfeld	53
5	Integration	55
5.1	Hardwareaufbau	56
5.2	Programmierung	57
5.2.1	Erstellung eines Hardwareclusters	57
5.2.1.1	Allgemeingültige Einstellungen	57
5.2.1.2	Showspezifische Einstellungen	58
5.2.2	Patchfile	60
5.2.3	Show-Programmierung	61
5.2.3.1	Unterschiede bei der Showprogrammierung	61
5.2.3.2	Praktisches Beispiel LiveFX	62
5.2.4	Probleme und Hinweise	65
5.3	Bewertung der Integration	66
6	Zusammenfassung	68
	Literaturverzeichnis	69

Tabellenverzeichnis

2.1	Elemente des Datenprotokolls (vgl. Elation (2008) S.17)	20
-----	---	----

Abbildungsverzeichnis

2.1	Prinzip einer seriellen Datenübertragung	12
2.2	Prinzip einer parallelen Datenübertragung	13
2.3	DMX-512 Sendeschaltung (DIN (2000) S.3)	16
2.4	DMX-512 Empfangsschaltung (DIN (2000) S.3)	17
2.5	DMX-512 Leitungsverstärker (DIN (2000) S.3)	17
2.6	Datenkabel für DMX-512 Bus (Elation (2008), S.16)	18
2.7	Kontaktseitige Ansicht eines 5-Pol XLR-Steckers (Elation (2008), S.15)	19
2.8	DMX 512 Datenprotokoll	20
2.9	DMX-Bus Daisy-hCain	21
3.1	Butler XT Anschlussfeld (e:cue control GmbH (2011b) S.7)	24
3.2	Darstellung einer Power over Ethernet Phantomspeisung (Belden-Electronics-GmbH (2011), S.4)	25
3.3	Darstellung einer Power over Ethernet Spare-Pair-Speisung (Belden-Electronics-GmbH (2011), S.4)	26
3.4	RJ45-Stecker und DMX Pinbelegung	26
3.5	e:bus: Bustopologie	28
3.6	e:bus: freie Topologie	28
4.1	Traxon power/data injector box	30
4.2	Taster-Board für die digitalen Eingänge des Butler XT	31
4.3	Aufbau für die Inbetriebnahme des Butler XT an einer 8x8 RGB-LED-Matrix	32
4.4	Auswahl des e:net	33
4.5	Verfügbare Geräte im e:net	34
4.6	Butler XT im Device Manager	35
4.7	Showspezifisches Konfigurationsmenü für den Butler XT	35
4.8	Preview eines simulierten Patchfiles mit einer RGB-LED-Matrix	37
4.9	Patching zweier LED-Matrizen	38
4.10	Cue Konfiguration	40
4.11	Cuelist Konfiguration	41
4.12	Übertragung von Shows auf den Butler XT	42
4.13	Programmer mit Cuelist und Preview-Fenster	43
4.14	Chaser Wizard	45

4.15 Bitmap-Wizard	46
4.16 Mit dem Bitmap-Wizard gerendertes Bild	46
4.17 Scroller-Wizard im LiveFX-Tool	47
4.18 Erstellen von LiveFX Effekten	48
4.19 Hinzufügen eines RS232-Geräts	52
4.20 Erstellen eines RS232-Triggers im online-Modus	53
4.21 Butler XT mit eingebundenen e:bus Glass-Touchscreen	54
4.22 Konfiguration des Glass-Touchscreens	54
5.1 Airbus Single Aisle Kabinen-Modell	55
5.2 Schema des Hardwareaufbaus ohne Power/Data-Injectors	56
5.3 Device Setup des Butler XT	58
5.4 Butler-Cluster im Device Manager	59
5.5 Patchfile der LED-Matrizen des Kabinenmodells	60
5.6 Exportieren der Show auf den Butler-Cluster	62
5.7 Sonnenaufgang im LiveFX-Tool	63
5.8 Sonnenuntergang im LiveFX-Tool	64
5.9 Nacht im LiveFX-Tool	64
5.10 Tagesszenario in der Cuelist	65

1 Einleitung

Im Flugzeugbau gibt es laufend Änderungen und Neuentwicklungen, vor allem im Bereich der Flugzeugkabine. Dort werden regelmäßig neue Kundenwünsche umgesetzt oder Ideen entwickelt, was Kunden sich zukünftig wünschen könnten. Ein wichtiger Aspekt ist dabei die Kabinenbeleuchtung. Wo vor einigen Jahren die Dimmbarkeit der Kabinenbeleuchtung die einzige Innovation war, werden heute deutlich mehr Möglichkeiten wie vollfarbige Lichtszenarien beispielsweise zur Darstellung von Tageszeiten verlangt.

Diese Arbeit befasst sich genau mit diesen Möglichkeiten und deren Umsetzung. Im Folgenden wird zunächst das Thema genauer beschrieben und anschließend das Ziel und die Struktur der Arbeit erläutert.

1.1 Themenbeschreibung

Wie bereits erwähnt, beschäftigt sich diese Arbeit mit der Lichtsteuerung in einem Flugzeugkabinen-Mockup. Hierbei handelt es sich um das Kabinenmodell eines Airbus A320, welches zu Studienzwecken mit kommerziellen Systemen ausgerüstet wird, die in der Luftfahrt nicht üblich sind, wie zum Beispiel Beleuchtungssysteme aus der Veranstaltungstechnik. Dadurch können auch fachübergreifend wissenschaftliche Analysen, beispielsweise in der Medientechnik, an diesem Modell durchgeführt werden. Für die Steuerung wird das auf dem DMX Datenbus-Standard basierende Lichtsteuergerät „Butler XT“ der Firma e:cue in Betrieb genommen. Dieser wird für die Ansteuerung von LED Matrizen außerhalb und innerhalb des Kabinenmodells genutzt. Die Firma e:cue ist ein Teil der Firma Traxon Technologies, welcher sich auf die Entwicklung von DMX-Controllern und der dazugehörigen Software konzentriert. Traxon Technologies selbst stellt hauptsächlich Beleuchtungselemente und anderes Zubehör für DMX-Controller her. Seit 2011 gehört Traxon Technologies zur Osram Licht AG, einem der weltweit führenden Hersteller für Beleuchtungslösungen.

Ein wichtiger Aspekt dieser Ausarbeitung ist die theoretische Aufarbeitung des DMX-512 Bussystems und der Vergleich dessen mit dem RS-485 Bussystem. Zudem müssen alle Schnittstellen des Butler XT für die erfolgreiche Inbetriebnahme identifiziert und analysiert

werden.

Der Butler XT wird anschließend so programmiert, dass er verschiedene Lichtszenarien auf den LED Matrizen als alleinstehendes Gerät ohne zusätzlichen Computeranschluss abspielen kann. Zusätzlich werden die Möglichkeiten der Implementierung eines Touchscreens zur Steuerung des Butler XT untersucht, um eine noch komfortablere Bedienung zu ermöglichen. An dieser Stelle kommt die Analyse der Schnittstellen des Controllers zum Tragen.

1.2 Ziel der Arbeit

Das Ziel dieser Arbeit liegt in erster Linie darin, die Inbetriebnahme des Butler XT zusammen mit einer LED Matrix so aufzuarbeiten, dass der Prozess vom Anschluss der Komponenten über die Programmierung des Controllers, bis hin zum Starten der programmierten Beleuchtungsszenarien, für Leser der Arbeit verständlich und reproduzierbar ist.

Außerdem soll der Grundlagenteil dieser Ausarbeitung ein nachvollziehbares Bild davon liefern, wie die Datenübertragung vom Controller zur LED Matrix funktioniert und aus welchem Grund dafür der DMX-512 Datenbus ausgewählt wurde. Zudem sollen die Funktionen und Möglichkeiten, welche der Butler XT bietet, übersichtlich dargestellt sein. Dabei müssen sowohl positive, als auch negative Aspekte des Systems hinsichtlich verschiedener Eigenschaften wie Funktionalität, Benutzerfreundlichkeit und Funktionsumfang herausgearbeitet werden.

1.3 Struktur der Arbeit

Zunächst werden die theoretischen Grundlagen serieller Bussysteme und speziell des DMX-512 Bussystems aufgearbeitet. Anschließend folgt die Einführung in den Butler XT, wobei Funktionen und Schnittstellen erklärt werden. Aufbauend auf die Grundlagen wird die Inbetriebnahme des Systems mit einem Butler XT und einer LED Matrix anschaulich erklärt. Anschließend werden die Möglichkeiten zur Implementierung eines Touchscreens herausgearbeitet. Zum Schluss wird die Integration und Inbetriebnahme des Systems in dem Airbus Kabinenmodell erläutert.

2 Grundlagen

Dieses Kapitel behandelt die wichtigsten Grundlagen serieller Datenübertragung sowie die Übertragungsstandards der Bussysteme, die dem verwendeten System zu Grunde liegen. Dabei handelt es sich zum einen um den DMX-512 Standard für den gleichnamigen Datenbus und zum anderen um den proprietären, von der Firma e:cue entwickelten, e:bus.

2.1 Serielle Bussysteme

Es gibt verschiedenste serielle Bussysteme, welche jedoch alle gewisse Parallelen aufweisen. In diesem Kapitel geht es darum, diese Parallelen aufzuzeigen und die wichtigsten Eigenschaften serieller Datenübertragung herauszuarbeiten.

Serielle Bussysteme finden sich in verschiedensten Bereichen wieder. Dabei handelt es sich beispielsweise um Computernetze, Feldbussysteme, Gebäude- und Fertigungsautomation oder das Internet (vgl. Samson-AG (2000), S.5).

2.1.1 Unterschied Seriell/Parallel

Bei der seriellen Form der Datenübertragung werden die Daten in Form von Bitstreams bitseriell übertragen. Das bedeutet, dass Datenpakete Bit für Bit über eine Datenleitung oder ein Leitungspaar übertragen werden. Mit Hilfe einer festgelegten Codierung, welche auch Protokoll genannt wird, kann der Bitstream am Empfänger rekonstruiert und interpretiert werden. Erreicht wird dies beispielsweise durch Start- und Stopbits, die den Rahmen des Datenwortes bilden, durch Adressierungsbits für die Zuweisung der Datenworte zu bestimmten Empfängern sowie durch Bitfolgen, welche die Länge der zu erwartenden Nutzdaten angeben und schließlich durch die Nutzdaten selbst. Die Nutzdaten können beliebig aufgeteilt werden. Es können zum Beispiel 8 Bit (1 Byte) Nutzdaten zur Darstellung von Dimmerwerten für eine Helligkeitssteuerung mit einer Helligkeitsauflösung von 8 Bit (0-255) übertragen werden. Wie die Nutzdaten am Ende jedoch tatsächlich interpretiert werden, hängt letztlich vom jeweiligen Empfänger ab. Wichtig ist nur, dass die Übertragungsparameter für Sender und Empfänger identisch sind. Im EIA-RS-232 Standard beispielsweise gehören dazu die

Baudrate, sowie die Anzahl der Stopbits, Paritätsbits und Datenbits. Bei der Baudrate handelt es sich um die Anzahl der Schritte pro Sekunde. Ein Baud gibt dementsprechend die Schrittweite an. Bei der RS232-Schnittstelle beträgt diese beispielsweise 1 Bit. Damit wäre in diesem Fall die Baudrate gleich der Bitrate. Es gibt jedoch auch serielle Datenübertragungen bei der die Baudrate nicht der Bitrate entspricht und zum Beispiel durch bestimmte Modulationsverfahren mehrere Bits pro Schritt übertragen werden können. Abbildung 2.1 zeigt das Prinzip einer solchen bitseriellen Datenübertragung, bei der die einzelnen Bits des Datenpakets seriell über eine Datenleitung übertragen werden und um ein Start- bzw. Stopbit ergänzt werden.

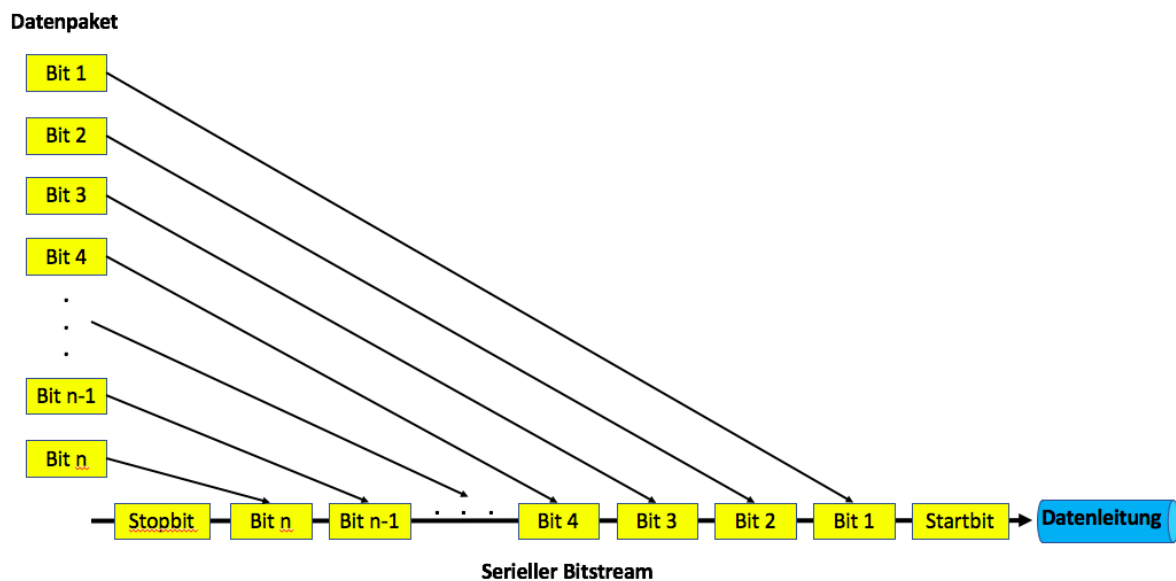


Abbildung 2.1: Prinzip einer seriellen Datenübertragung

An dieser Stelle zeigt sich auch der Unterschied zu parallelen Bussystemen. Parallele Bussysteme sind in der Lage, Daten paketweise über mehrere Datenleitungen zu übertragen. Somit kann die Schrittweite für die Datenübertragung durch die Anzahl der Datenleitungen beeinflusst werden. Da die einzelnen Bits dieser Datenpakete gleichzeitig am Empfänger ankommen müssen, muss jedoch sehr stark auf Laufzeitunterschiede durch verschiedene Leitungslängen geachtet werden, was diese Art der Datenübertragung gerade bei größeren Leitungslängen deutlich schwieriger in der Umsetzung macht. Außerdem steigen durch die hohe Leitungsanzahl Kosten und Gewicht des Systems sehr schnell bei größeren Leitungslängen. Abbildung 2.2 verdeutlicht das Prinzip der parallelen Datenübertragung, bei der jedes Bit des Datenpakets über eine eigene Datenleitung gesendet wird.

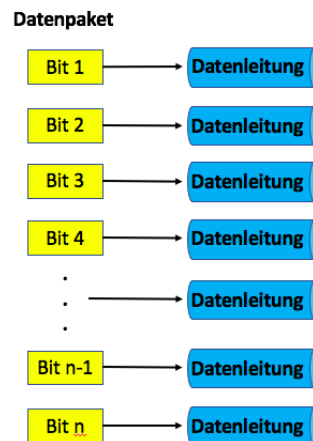


Abbildung 2.2: Prinzip einer parallelen Datenübertragung

2.1.2 Synchrone/asynchrone Datenübertragung

Unter den verschiedenen seriellen Bussystemen gibt es jedoch noch einige grundlegende Unterschiede. Dazu gehört der Unterschied zwischen synchroner und asynchroner serieller Datenübertragung.

Bei der seriellen synchronen Datenübertragung werden mehrere Zeichen als Gruppe zusammengefasst und übertragen. Anstatt eines Startbits wird hier ein Synchronisationsbyte als Startsignal für die Übertragung gesendet. Eine Kennung wird der Zeichengruppe nicht angehängt. Zudem wird neben den Daten ein zusätzlicher Takt gesendet, der eine explizite Synchronisation von Sender und Empfänger gewährleistet. Diese Form der Datenübertragung erfordert auf beiden Seiten Timing-Bausteine und ein hohes Maß an Präzision des Equipments (vgl. Elation (2008), S.11). Diese Anforderungen lassen folglich auch die Kosten des Systems steigen.

Im Fall der asynchronen Datenübertragung werden die Datenpakete mit einem Start- und einem Stopbit versehen und gesendet. Dabei wird auf eine explizite Synchronisierung durch einen extra Takt verzichtet. Die Synchronisierung erfolgt in diesem Fall implizit durch das Übertragungsprotokoll. Der Empfänger orientiert sich an den Start- und Stopbits, welche den Rahmen des Datenworts festlegen und der Takt wird über die eingestellte Baudrate vorgegeben. Dieses Verfahren erleichtert die Umsetzung, verringert die Komplexität der Hardware und senkt somit auch die Kosten des Systems erheblich.

2.1.3 Leitungsanzahl (single-ended/differential)

Es gibt zwei Möglichkeiten, die serielle Übertragung zu ermöglichen. Die Daten können auf einer einzigen Datenleitung (single-ended) oder über zwei Leitungen (differential) übertragen werden.

Wenn die Datenübertragung über eine einzige Datenleitung realisiert wird, entsteht eine sogenannte „single-ended“-Übertragung. Diese Form eignet sich nicht für sehr schnelle Übertragungen bei großen Leitungslängen. Durch die fehlende gegenpolige Datenleitung wird diese Art der Datenübertragung auch „unbalanced“ genannt. Sie ist dadurch sehr anfällig für Störungen und Interferenzen, da diese schwer zu identifizieren und kompensieren sind.

Die differentielle Datenübertragung ist die verbreitetere und leistungsstärkere Art der seriellen Datenübertragung. Dabei werden zwei Leitungen für die Übertragung verwendet, wodurch Störungen von außen in beiden Leitungen gleichermaßen auftreten. Dadurch, dass beide Datenleitungen eine gegensätzliche Polarität aufweisen, können die auftretenden Interferenzen, sofern es sich um Gleichtaktstörungen handelt, durch Differenzbildung beider Signale eliminiert und große Leitungslängen beziehungsweise schnelle Datenübertragungen ermöglicht werden. Außerdem werden die beiden Datenleitungen für die differentielle Datenübertragung meist verdrillt als sogenanntes „Twisted Pair“ genutzt. Durch das Verdrillen der Leitungen wird eine noch robustere Datenübertragung gewährleistet, da sich so Störungen durch elektromagnetische Störfelder größtenteils direkt aufheben.

2.2 DMX-512 Datenbus

Dieses Kapitel behandelt den DMX-512 Datenbus, welcher für die Lichtsteuerung des DMX-Controllers in dieser Arbeit verwendet wird. DMX steht in diesem Fall für Digital Multiplex und die Zahl 512 gibt die Anzahl der Kontrollkanäle des Datenbusses an. Der DMX-512 Standard wurde in erster Linie für die Veranstaltungstechnik entwickelt. Er soll es ermöglichen, Systeme verschiedener Hersteller effektiv miteinander zu verbinden (vgl. *plasastandards* (2011b), S.1).

Hier wird zunächst die Entstehung des Bussystems erläutert. Anschließend wird Bezug auf die benötigte Hardware und auf das Datenprotokoll genommen. Schließlich werden Gemeinsamkeiten und Unterschiede zum RS-485 Datenbus, welcher der Vorgänger des DMX-512 Bussystems ist, herausgearbeitet.

2.2.1 Entstehung

1986: Der DMX-512 Datenbus wurde von der USITT¹ entwickelt. Er wurde auf der Basis des EIA/TIA²-485 Standards aufgesetzt (vgl. Elation (2008), S.5).

1998: Die ESTA³ übernimmt in ihrem „Technical Standards Program“ die Wartung des DMX-512 Protokolls.

2004: Sechs Jahre später hat dann das ANSI⁴ das DMX-512 Protokoll übernommen und um zwei Funktionen erweitert:

- Remote Device Management (RDM)
- Architecture for Control Networks (ACN)

2.2.2 Hardware

Für die Übertragung von Signalen auf dem DMX-Bus gibt es bestimmte genormte Hardwarevorschläge. Diese finden sich unter anderem in der DIN 56930-2 (vgl. DIN (2000)). Nachfolgend wird auf diese DIN Bezug genommen.

2.2.2.1 Sende-und Empfangsschaltung

Wie in den Abbildungen 2.3 und 2.4 zu sehen ist, werden als Sende- und Empfangsbau- steine RS-485 Bausteine verwendet. Das ist dadurch zu erklären, dass für den DMX Bus der physical Layer (physikalische Schicht) und somit die Signalpegel des EIA-485 Standards verwendet werden.

¹United States Institute of Theatre Technology

²Electronics Industry Alliance / Telecommunications Industry Alliance

³Entertainment Services and Technology Association

⁴American National Standards Institute

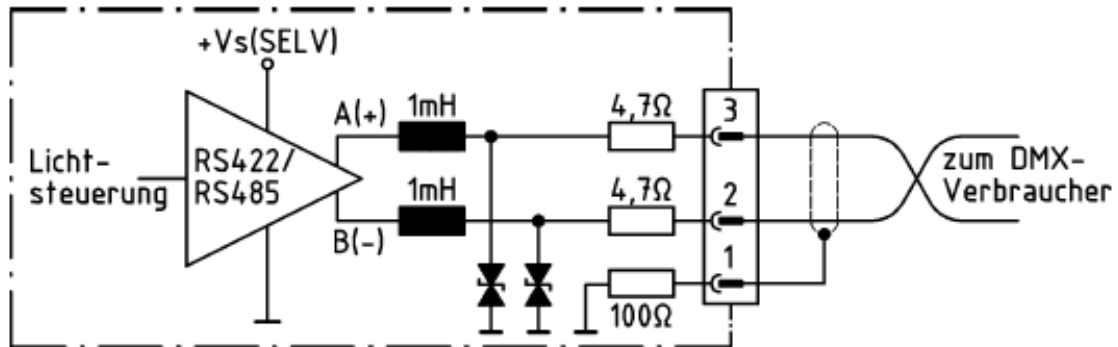


Abbildung 2.3: DMX-512 Sendeschaltung (DIN (2000) S.3)

Für die Schaltungen werden Reihenwiderstände und Transientenschaltungen als Überspannungsschutz verwendet (Siehe Abb. 2.3 und 2.4). Der Ausgang der Sendeschaltung wird mit Induktivitäten versehen, um der kapazitiven Wirkung langer Datenleitungen entgegenzuwirken (Siehe Abb. 2.3). Dabei wird der kapazitiven Blindleistung entgegengewirkt, die bei großen Leitungslängen entstehen kann und den Bus zu stark belasten würde. Induktivitäten werden an dieser Stelle genutzt, da die induktive und die kapazitive Blindleistung zueinander um 180 Grad phasenverschoben sind. Durch diese Eigenschaft kompensieren sie sich gegenseitig. Die Größe der Induktivitäten wird jedoch so gewählt, dass sie den Bus bei kurzen Leitungslängen durch fehlende kapazitive Blindleistung nicht zu sehr durch induktive Blindleistung belasten. Diese Maßnahme kann jedoch lediglich eingesetzt werden, weil für den DMX-512 Standard eine feste Übertragungsfrequenz von 250kHz gilt.

Sende- und Empfangsschaltung sind sich weitestgehend ähnlich. Die Empfangsschaltung hat lediglich keine Induktivitäten, da diese nur einmal auf der Datenleitung benötigt werden. Die Empfangsschaltung enthält außerdem noch zusätzliche $22k\Omega$ -Widerstände um den Eingängen Bezugspotentiale zu geben (Siehe Abb. 2.4). Dadurch wird im Leerlauf immer eine logische „1“ am Empfänger gewährleistet und undefinierte Zustände oder Schwingungsverhalten der Operationsverstärker vermieden. Weitere Informationen zu den Signalpegeln sind in Unterkapitel 2.2.3 zu finden.

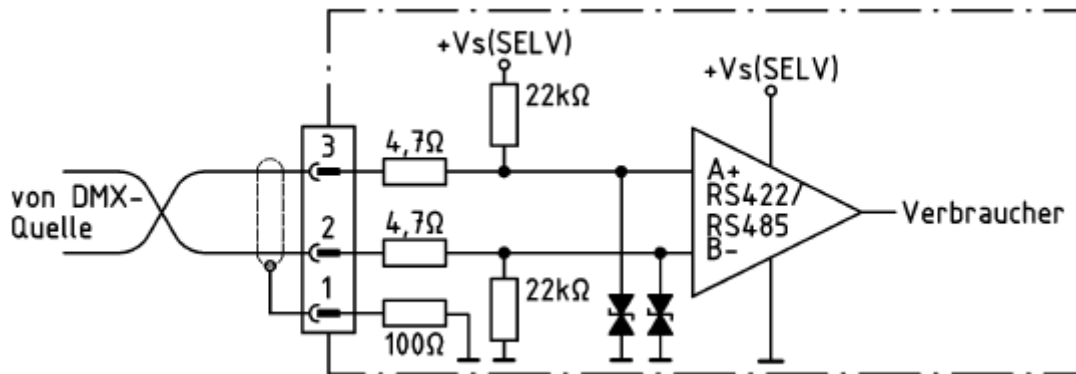


Abbildung 2.4: DMX-512 Empfangsschaltung (DIN (2000) S.3)

2.2.2.2 Leitungsverstärker

Leitungsverstärker werden bei sehr langen Übertragungswegen benötigt. Sie sind eine Kombination aus Sende- und Empfangsschaltung. Dabei wird wie in Abbildung 2.5 einem Sender direkt ein Empfänger vorgeschaltet, wodurch das Signal empfangen, seine Signalfanken rekonstruiert und es mit der vollen Sendeleistung wieder auf den Bus gesendet wird. Dadurch, dass diesem Prozess keine digitale Signalverarbeitung zu Grunde liegt, funktioniert die Signalaufbereitung latenzfrei.

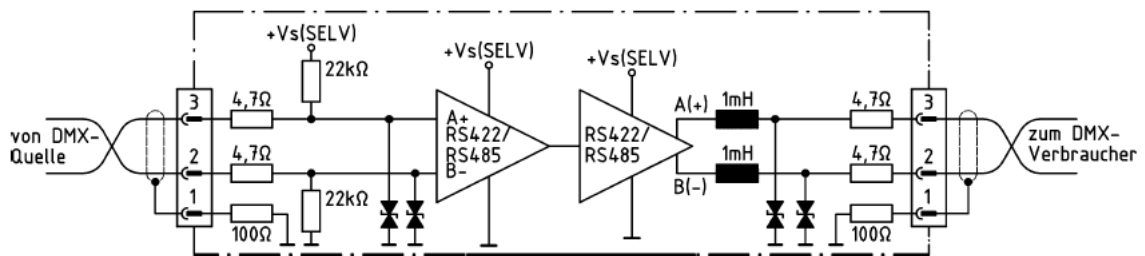


Abbildung 2.5: DMX-512 Leitungsverstärker (DIN (2000) S.3)

2.2.2.3 Datenkabel

Als Datenkabel kommen geschirmte Kabel mit verdrehten Datenleitungspaaren und einer Drain-Leitung für die einfache Erdung der Schirmung in Frage (Siehe Abb. 2.6). An dieser Stelle werden auch Ethernetleitungen (mindestens CAT5) mit RJ45-Steckern verwendet.

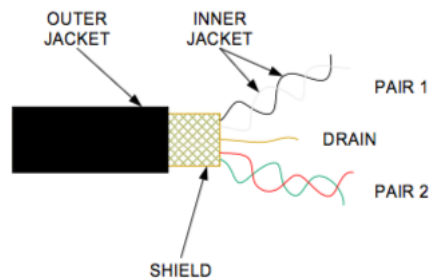


Abbildung 2.6: Datenkabel für DMX-512 Bus (Elation (2008), S.16)

Die verwendeten Leitungen müssen bestimmte Eigenschaften aufweisen. Zum einen benötigen sie die oben genannte Schirmung, zum anderen müssen Sie bestimmte Grenzwerte der Leitungskapazitäten einhalten. Dabei muss die Leiter zu Leiterkapazität unter einem Wert von 65pF pro Meter und die Kapazität von Leiter zur Schirmung unter einem Wert von 115pF pro Meter bleiben (vgl. plasastandards (2011b), S.4).

Da der DMX-512 Bus die differentielle Datenübertragung nutzt, muss es für die Datenübertragung immer zwei Datenleitungen geben (Siehe Unterkapitel 2.1.3). Sollte lediglich eine einseitige Datenübertragung gewünscht sein, reicht ein Datenleitungspaar. Moderne DMX-512 Geräte unterstützen teilweise auch bidirektionale Verbindungen, bei denen auf einem zweiten Leitungspaar beispielsweise Statusdaten der angeschlossenen Geräte an den DMX-Controller zurückgemeldet werden können.

Den Leitungsabschluss bildet beim DMX-512 Bussystem immer ein 120Ω -Widerstand mit einer Belastbarkeit von mindestens 0,25W (vgl. DIN (2000), S.3). Dieser wird am Leitungsende mit der positiven und negativen Datenleitung verbunden.

Als Stecker werden für den DMX-512 Bus entweder wie oben genannt RJ45-Stecker oder XLR-Rundstecker verwendet. Die Pinbelegung der RJ45-Stecker ist je nach Hersteller des verwendeten Systems unterschiedlich. Für den XLR-Stecker in Abbildung 2.7 gilt eine feste Pinbelegung:

- Pin 1:** Masse (Schirmung)
- Pin 2:** DATA1 -
- Pin 3:** DATA1 +
- Pin 4:** DATA2 -
- Pin 5:** DATA2 +

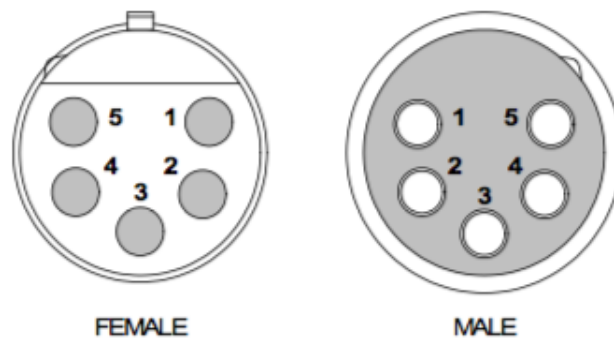


Abbildung 2.7: Kontaktseitige Ansicht eines 5-Pol XLR-Steckers (Elation (2008), S.15)

2.2.3 Datenprotokoll DMX-512

Im Folgenden wird das Datenprotokoll des Bussystems behandelt. Dabei werden sowohl die Spannungspegel, als auch der Datenwortaufbau näher betrachtet.

Die Spannungspegel des DMX-512 Bussystems basieren auf dem EIA-485 Standard für das RS-485 Bussystem. Dabei werden differentielle Spannungspegel von $\pm 1,5V$ bis $\pm 5V$ genutzt. Es gibt immer eine invertierte und eine nicht invertierte Signalleitung, welche immer denselben Spannungspegel nur mit entgegengesetztem Vorzeichen führen. Der Differenzspannungspegel wird nun aus der Differenz beider Spannungspegel gebildet. Der Zustand einer logischen 1 „MARK“ entspricht einer Differenzspannung von $-1,5V$ bis $-5V$ und der Zustand einer logischen 0 „SPACE“ entspricht einer Differenzspannung von $+1,5V$ bis $+5V$ (vgl. Schnell und Wiedemann (2006), S.60). Durch die Nutzung des Differenzspannungspegels können Gleichtaktstörungen effektiv herausgefiltert werden, da sie in beiden Datenleitungen gleichermaßen auftreten und nach der Differenzbildung wegfallen. Diese Eigenschaft erhöht die Datensicherheit des DMX-512 Bussystems erheblich.

Abbildung 2.8 und Tabelle 2.1 zeigen das Datenprotokoll des DMX-512 Standards. Das Datenprotokoll beinhaltet zum einen den übergeordneten Rahmen für das DMX-Datenpaket. Dieser besteht zunächst aus der „Break“, welche einen Reset und den Anfang eines neuen

Datenpakets darstellt. Danach folgt der „Mark After Break“, welcher dem Empfänger signalisiert, mit dem Lesen des Datenpakets zu beginnen. Dabei liest der Empfänger als erstes den Start Code, welcher die Art der Nutzdaten angibt. Bei den herkömmlichen Anwendungen ist der Start Code immer 0 für 8-Bit Datenwerte. Nach dem Start Code folgen die Nutzdaten. Zwischen den Nutzdatenwerten gibt es eine optionale Wartezeit (Mark Time Between Frames). Die Nutzdaten selber bestehen immer aus einem Startbit, dem 8-Bit Datenwert und zwei Stopbits. Dieses Format ist, anders als beim RS-232- oder RS-485-Standard, nicht veränderbar. Am Ende des DMX-Datenpakets gibt es noch eine optionale Wartezeit (Mark Time Between Packets), bevor das nächste Datenpaket anfängt. Die Daten werden mit einer festen Datenrate von 250kBit pro Sekunde übertragen. Die einzelnen Elemente des Datenprotokolls werden in Tabelle 2.1 näher erläutert.



Abbildung 2.8: DMX 512 Datenprotokoll

Tabelle 2.1: Elemente des Datenprotokolls (vgl. Elation (2008) S.17)

Element	Beschreibung	Pegel	Bits	Dauer
BREAK	Reset für ein neues DMX Paket	LO (0)	22b-250Kb	$88\mu s - 1s$
Mark After Break (MAB)	Signalisiert dem Empfänger Daten zu lesen	HI(1)	2b-250Kb	$8\mu s - 1s$
Start Code (SC)	0 für standard 8-Bit Dimmerwerte (alternative Startcodes möglich)	MIX	11b	$44\mu s$
Mark Time Between Frames (MTBF)	Zeit zwischen zwei Rahmen für Daten (Dimmerwerte) eines Datenpakets	HI(1)	0b-250Kb	$0 - 1s$
DATA	8-Bit DMX Datenwert (Dimmerwert) inklusive einem Start- und zwei Stopbits	Start:LO(0), Stop:HI(1), Daten:MIX	11b	$44\mu s$
Mark Time Between Packets (MTBP)	Zeit zwischen zwei Datenpaketen	HI(1)	0b-250Kb	$0-1s$

Wie hier zu erkennen ist, besitzt das DMX-Protokoll keine Fehlererkennungsmethodik wie beispielsweise Paritätsbits, eine Prüfsumme oder einen CRC-Wert. Daher eignet sich dieses

Bussystem nicht für kritische Anwendungen, bei denen Übertragungsfehler fatale Folgen haben könnten.

Wenn alle 512 Kanäle des DMX-Protokolls genutzt werden, hat das Datenwort eine minimale Größe von 5667 Bit. Da die Daten auf dem DMX-Bus mit einer Datenrate von 250KBaud bei einer Symbolgröße von 1 Baud = 1 Bit/s übertragen werden, können wie in Rechnung 2.1 44 solcher Datenpakete pro Sekunde auf den DMX-Bus gesendet werden.

$$n = \frac{250000s^{-1}}{5667} \approx 44,115s^{-1} \quad (2.1)$$

2.2.4 Zusammenhang DMX-512 und RS-485

Wie im Unterkapitel zuvor bereits angedeutet, gibt es gewisse Parallelen zwischen dem DMX-512 Bus und dem RS-485 Bus. Diese Parallelen ergeben sich daraus, dass der DMX-512 Übertragungsstandard auf dem EIA-485 Standard aufbaut. Die Gemeinsamkeiten beschränken sich jedoch auf Aspekte des physical Layers wie die Spannungspegel und damit auf die für die Signalübertragung genutzten Bauteile, sowie auf die Bustopologie.

Beide Bussysteme nutzen eine sogenannte Linientopologie, bei der die Busteilnehmer in einer „Daisy-Chain“ miteinander verbunden sind. Das bedeutet, dass die Datenleitungen immer von einem Teilnehmer zum nächsten geführt werden und der letzte Teilnehmer mit einem Widerstand den Busabschluss bildet. Dieser Busaufbau wird in Abbildung 2.9 dargestellt.

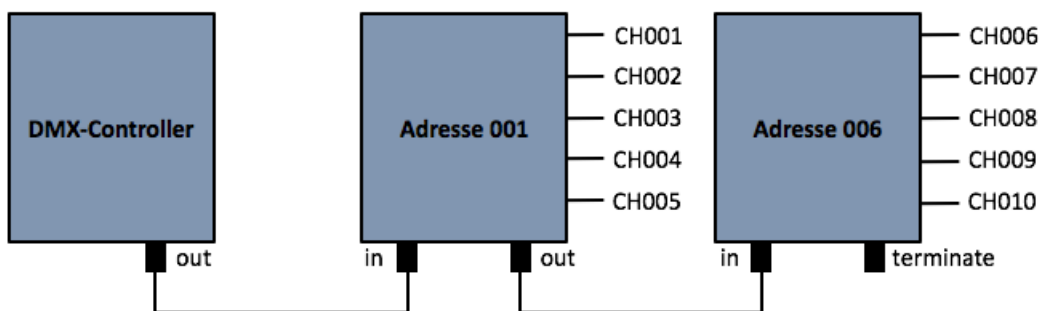


Abbildung 2.9: DMX-Bus Daisy-Chain

Den entscheidenden Unterschied zwischen beiden Bussystemen bildet letztlich der Datenaufbau und insbesondere die Adressierung der anzusteuern Geräte. Für den RS-485 Bus ist eine Adressierung des jeweiligen Empfängers in jedem Datenwort vorgesehen. Somit werden die Daten jedes Empfängers auf dem Datenbus einzeln mit der jeweiligen

Adresse gesendet. Der DMX-Bus arbeitet an dieser Stelle anders. Den Busteilnehmern, (z.B. Dimmern) werden vorab von dem DMX-Controller Adressbereiche zugeteilt. Die Adressbereiche werden in den Geräten gespeichert (vgl. Elation (2008), S.10). Dieser Vorgang wird auch „Patching“ genannt. Wie das Patching letztendlich funktioniert, hängt stark vom Hersteller des verwendeten Systems ab. Bei einigen Systemen werden die Adressen oder Adressbereiche der DMX-Empfänger direkt am Empfänger selbst über Dip-Schalter oder andere Hardwarelösungen eingestellt. Andere Hersteller nutzen Softwarelösungen für das Patching der Empfänger über die DMX-Datenleitungen. Diese Softwarelösungen sind aber in der Regel als geistiges Eigentum der Hersteller nicht näher spezifiziert. Die Daten werden, wie in Abbildung 2.8 zu sehen ist, bei Channel 1 angefangen der Reihe nach in einem Datenpaket übertragen. Da die Empfänger die für sie relevanten Channel (Adressen) kennen, ignorieren sie alle Datenwerte, die für sie nicht relevant sind. Dadurch wird eine spezielle Adressierung im Datenwort überflüssig.

Ein weiterer Unterschied zwischen beiden Bussystemen ist die fehlende Fehlererkennung beim DMX-512 Standard. Der RS-485 Bus unterstützt hingegen Fehlererkennungsverfahren wie ein Paritätsbit oder eine Prüfsumme.

2.2.5 Steuerung von RGB-Lichtquellen

Häufig wird der DMX-Bus zur Steuerung von RGB-Lichtquellen eingesetzt. RGB steht für „Red“, „Green“ und „Blue“. Diese Lichtquellen mischen die drei Farben mit einstellbaren Intensitäten, wodurch sie in der Lage sind, alle möglichen Farben darzustellen. Bei der Steuerung von RGB-Lichtquellen über den DMX-Bus gibt es eine Besonderheit zu beachten. RGB-Lichtquellen können nicht wie normale Dimmer über einen DMX-Kanal gesteuert werden. Um alle Farbmischungen darstellen zu können, müssen die drei Farben einzeln gedimmt werden können. Das bedeutet, dass jede RGB-Lichtquelle drei DMX-Kanäle für die Farbmischung benötigt:

- 1 - Intensität von RED
- 2 - Intensität von GREEN
- 3 - Intensität von BLUE

Konkret bedeutet das, dass pro DMX-512 Bus maximal $512/3 \approx 170$ RGB-Lichtquellen unabhängig gesteuert werden können. Ein DMX Bus mit 512 Kanälen wird auch DMX-Universum genannt. Oft besitzen DMX-Controller mehr als nur ein DMX-Universum. Diese DMX-Universen sind immer elektrisch voneinander getrennt.

3 Butler XT DMX-Controller

Das Folgende Kapitel befasst sich mit den Eigenschaften, Fähigkeiten und Besonderheiten des verwendeten DMX-512 Controllers „Butler XT“. Außerdem werden in diesem Kapitel die Schnittstellen und unterstützten Bussysteme des Controllers erläutert. Das Kapitel stützt sich hauptsächlich auf das Datenblatt und die Bedienungsanleitung des Butler XT (vgl. e:cue control GmbH (2011a) und e:cue control GmbH (2011b)).

3.1 allgemeine technische Daten

Der Butler XT bietet zwei voneinander unabhängige DMX Universen. Damit besitzt er insgesamt 1024 DMX Kanäle und ist mit anderen Butler XT und Butler Classic kaskadierbar auf bis zu 65536 DMX-Kanäle. Außerdem ist der Butler XT fähig, vollständig im Standalone-Modus ohne angeschlossenen PC zu funktionieren. Damit einher geht die Möglichkeit, Shows und Szenarien auf einer micro-SD Karte zu speichern und diese auf dem Butler XT abzuspielen. Die Shows können entweder direkt mit Hilfe eines SD-Kartenlesegeräts auf der SD-Karte gespeichert werden oder von einem PC mit der dazugehörigen Software (Lighting Application Suite) auf den Butler XT übertragen werden.

Dazu kommen eine interne Echtzeituhr sowie eine astronomische Uhr. Die Echtzeituhr ermöglicht es dem Butler XT auf bestimmte voreingestellte Uhrzeiten zu reagieren, die astronomische Uhr hingegen erlaubt bei korrekter Angabe der geografischen Position des Butler XT Reaktionen auf den Stand der Sonne in dem jeweiligen Gebiet. Für die Benutzereingabe besitzt der Butler XT zwei frei konfigurierbare Taster sowie acht konfigurierbare digitale Eingänge. Alle erwähnten Anschlüsse sind in Abbildung 3.1 zu sehen. Zur Anzeige des aktuell abgespielten Lichtszenarios und des Systemstatus besitzt der Butler XT eine 7-Segment Anzeige, sowie fünf Status-LEDs.

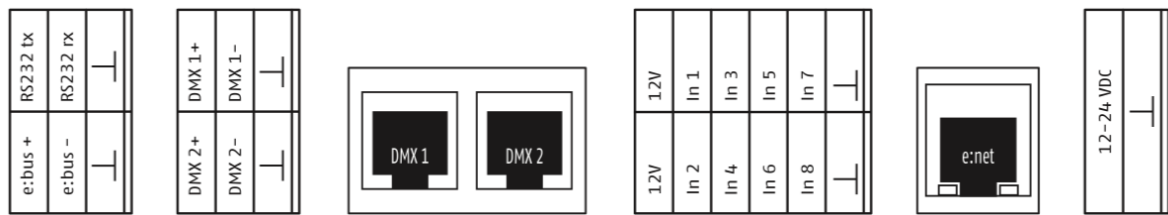


Abbildung 3.1: Butler XT Anschlussfeld (e:cue control GmbH (2011b) S.7)

3.2 Schnittstellen

In Abbildung 3.1 sind außerdem Anschlüsse diverser Schnittstellen sichtbar. Der Butler XT besitzt außerdem eine Infrarot-Schnittstelle für eine Traxon Infrarotfernbedienung. Die Kommunikation mit dem PC im sogenannten Live-Modus oder auch Online-Modus funktioniert über eine 100MBit e:net Schnittstelle, welche über einen RJ45-Stecker erreichbar ist. Das e:net ist nichts anderes als ein lokales Netzwerk, in welchem über einen Switch alle Butler DMX-Controller und der PC, mit dem diese programmiert werden sollen, miteinander verbunden werden. Im Online-Modus wird das e:net von der Programmiersoftware für das e:cue System gehostet, welche in Unterkapitel 4.2 näher betrachtet wird. Im Standalone-Modus können die Butler bei korrekter Einstellung selbstständig im e:net kommunizieren, was bei der Integration des Systems in das Flugzeug-Mockup von Vorteil ist.

Die zuvor erwähnten DMX512 Kanäle sind ebenfalls über RJ45-Stecker sowie über Anschlussklemmen erreichbar. Die DMX-Schnittstellen des Butler XT unterstützen gleichzeitig das RDM-Protokoll (Remote Device Management), worauf im späteren Verlauf noch einmal genauer eingegangen wird. Zusätzlich enthält der Butler XT eine RS232-Schnittstelle, welche über Anschlussklemmen erreichbar ist. Über diese RS232-Schnittstelle können Ein- und Ausgabegeräte anderer Hersteller angeschlossen werden. Für Eingabegeräte von Traxon Technologies bietet der Butler XT mit dem proprietären e:bus ein eigenes Bussystem. Wie das RDM-Protokoll wird auch dieses Bussystem im späteren Verlauf nochmals genauer erklärt.

3.3 Stromversorgung

Die Stromversorgung des Butler XT kann auf verschiedenen Wegen realisiert werden. Der empfohlene Weg ist die Versorgung über ein externes Netzteil mit einer Gleichspannung von 24V und einem Ausgangsstrom von mindestens 1,3A.

Eine Alternative dazu ist Power over Ethernet (PoE). Dort gibt es zwei verschiedene Verfahren zur Einspeisung der Versorgungsspannung. Das ist zum einen die „Phantomspeisung“ und zum anderen die „Spare-Pair-Speisung“ (vgl. Belden-Electronics-GmbH (2011), S.4).

Bei der in Abbildung 3.2 gezeigten Phantomspeisung wird die Versorgungsspannung in die Datenleitungen auf Pin 1/2 (+) und Pin 3/6 (-) eingekoppelt. Diese Methode bietet den Vorteil, dass sie auch bei vier adriger Verkabelung funktioniert.

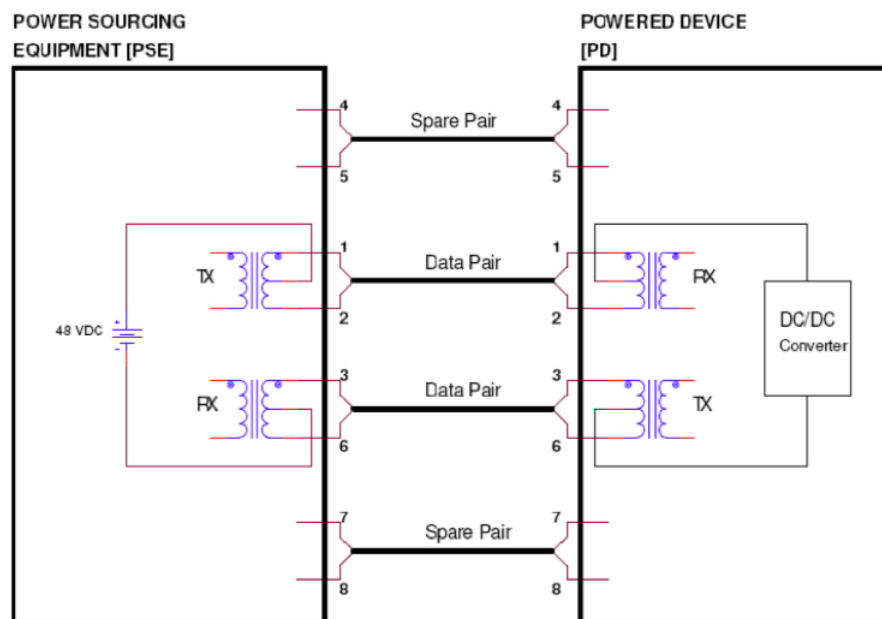


Abbildung 3.2: Darstellung einer Power over Ethernet Phantomspeisung (Belden-Electronics-GmbH (2011), S.4)

Die Spare-Pair-Speisung nutzt hingegen, wie in Abbildung 3.3 zu sehen ist, die beiden freien Aderpaare 4+5 (+) und 7+8 (-) für die Einkopplung der Spannungsversorgung. Diese Methode ist allerdings nur bei acht adriger Verkabelung nutzbar.

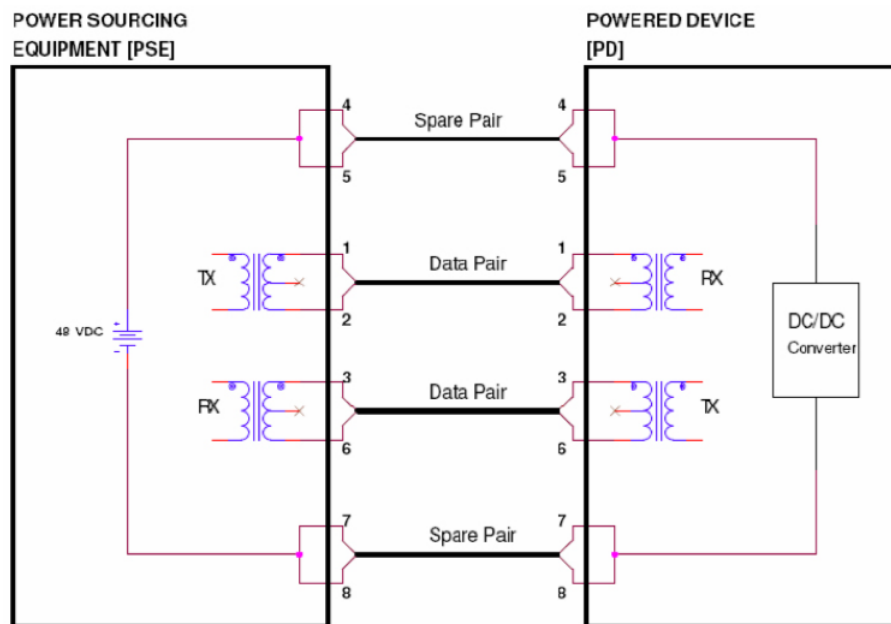


Abbildung 3.3: Darstellung einer Power over Ethernet Spare-Pair-Speisung (Belden-Electronics-GmbH (2011), S.4)

Die Pinreihenfolge auf dem RJ45-Stecker ist in Abbildung 3.4 sichtbar. Die Tabelle mit der Pinbelegung ist jedoch nur für DMX und nicht für Ethernet gültig.

Die dritte und letzte Möglichkeit der Versorgung ist Power via DMX. Dafür muss der Schalter hinter den DMX-Anschlüssen nach rechts geschoben werden. Die Spannungsversorgung funktioniert dann über Pin 5+6 des RJ45-Steckers (Siehe Abbildung 3.4). Diese Art der Spannungsversorgung hat jedoch zur Folge, dass der e:bus abgeschaltet wird.

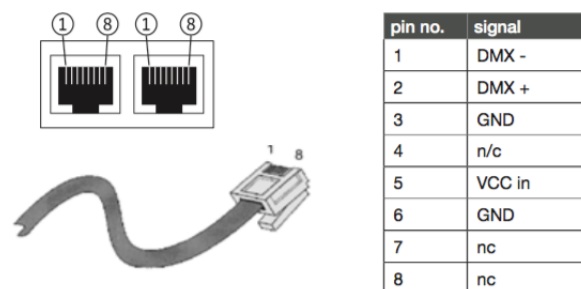


Abbildung 3.4: RJ45-Stecker und DMX Pinbelegung

3.4 RDM (Remote Device Management)

Das Remote Device Management ist eine Erweiterung des DMX-512 Bussystems. Es erlaubt eine bidirektionale Verbindung zwischen DMX-Controller und RDM-kompatiblen Ausgabegeräten. Bei dieser bidirektionalen Verbindung können beispielsweise Fehlermeldungen oder Statusmeldungen wie zum Beispiel Batteriestatus oder Leuchtmittelbetriebsstunden an den DMX-Controller zurückgemeldet werden. Diese Eigenschaft macht das Gesamtsystem sicherer und besser überwachbar. So können Probleme wie ein Leuchtmittelausfall behoben werden, bevor sie entstehen.

Es gibt zwei verschiedene Möglichkeiten von RDM-Übertragungen. Zum einen ist es möglich, diese Übertragungen über jene DMX-Datenleitungen zu realisieren, über die auch die Steuerung des normalen Sendebetriebs läuft. Dabei stellt der DMX-Controller den normalen Sendebetrieb für eine definierte Zeitspanne ein und sendet eine Anfrage an das jeweilige RDM-fähige Gerät. Jedes dieser Geräte hat einen sogenannten UID (unique identifier). Der DMX-Controller fragt immer nur einen UID zurzeit ab, damit nicht mehrere Geräte gleichzeitig auf den Bus senden. Das Senden von RDM-Nachrichten funktioniert über ein speziell angepasstes DMX-Protokoll (vgl. [plasastandards \(2011a\)](#), S.17 und folgende) und nutzt dabei einen speziellen Startcode (0xCC), damit kein anderes Gerät außer dem DMX-Controller fälschlicherweise diese Daten liest. Der Vorteil bei der Mitnutzung der zwei Datenleitungen für die Standardübertragung ist die Tatsache, dass nur ein Aderpaar für die gesamte Übertragung benötigt wird. Der Nachteil ist jedoch, dass der DMX-Controller für jede RDM-Anfrage seinen normalen Sendebetrieb einstellen muss.

Die zweite Möglichkeit der RDM-Übertragung ist ein zweites Datenleitungspaar allein für die Nutzung des RDM. Die Übertragung funktioniert in diesem Fall nicht anders als zuvor, der DMX-Controller muss nun jedoch nicht mehr seinen normalen Sendebetrieb für die RDM-Anfragen einstellen. Vorteile ergeben sich in diesem Fall vor allem bei Anwendungen mit sehr hoher Busauslastung, bei denen dem DMX-Controller für einen reibungslosen Betrieb keine Zeit für Unterbrechungen des Sendebetriebs bleibt.

Da das Remote Device Management nicht direkt relevant für diese Arbeit ist, wird der Umfang der theoretischen Ausarbeitung an dieser Stelle beschränkt. Weitere Informationen zum Thema RDM sind in der ANSI E1.20 (vgl. [plasastandards \(2011a\)](#)) zu finden.

3.5 e:bus

Der e:bus ist eine Eigenentwicklung von Traxon Technologies Europe. Es handelt sich hierbei um ein serielles Bussystem, welches dazu dient, User terminals von Traxon Technologies mit DMX-Controllern von e:cue in einer funktionalen Umgebung zu verbinden. Dabei kann ein Master wie zum Beispiel der Butler XT mit bis zu acht User Terminals verbunden werden, über welche der Butler XT dann gesteuert werden kann (vgl. Traxon-Technologies-Europe (2011), S.4).

Der e:bus unterstützt power over e:bus und funktioniert bei Leitungslängen von bis zu 400 Metern, insofern die Linientopologie, auch Bustopologie genannt, genutzt wird (Siehe Abb. 3.5). Lediglich beim Anschluss von Glass-Touchscreens sollte eine Leitungslänge von 100 Metern nicht überschritten werden.

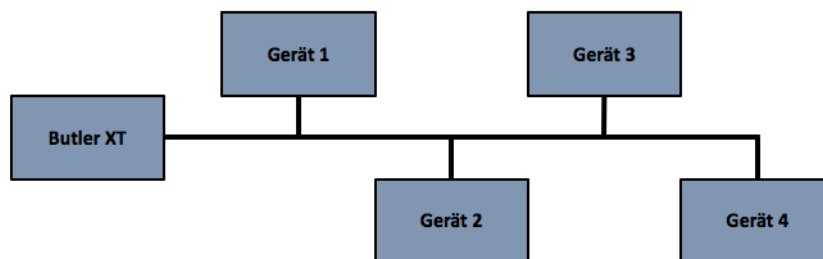


Abbildung 3.5: e:bus: Bustopologie

Der e:bus besitzt einige Besonderheiten. Dazu zählt die Topologiefreiheit. Das bedeutet, dass dieses Bussystem nicht an eine bestimmte Topologie gebunden ist, sondern auch freie Topologien, bei denen Baum- und Sternschaltungen kombiniert werden, unterstützt (Siehe Abb. 3.6).

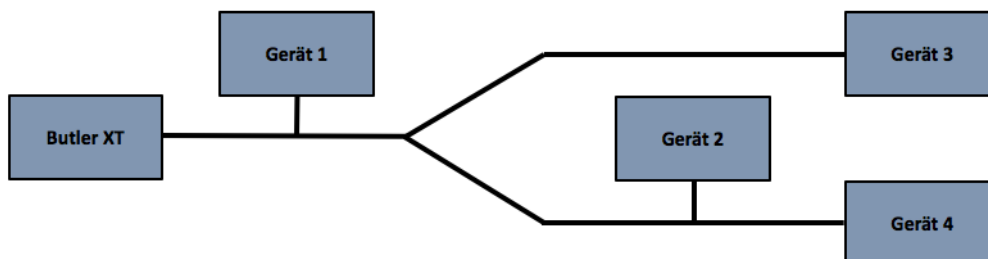


Abbildung 3.6: e:bus: freie Topologie

Außerdem ist der e:bus vollkommen unempfindlich gegen die Verpolung der Datenleitungen. Verpolungen werden automatisch erkannt und das Gerät dementsprechend automatisch umkonfiguriert. Auch defekte Geräte haben keinen negativen Einfluss auf das Gesamtsystem. Werden neue Geräte an den e:bus angeschlossen oder bestehende Geräte getauscht, werden diese bei einem kurzen Bus-Update automatisch registriert und adressiert.

Einige Dinge gibt es jedoch zu beachten. Offene Leitungsenden und Kurzschlüsse sind auch beim e:bus zu vermeiden. Es kann immer nur maximal einen Master in einem e:bus Netzwerk geben und außerdem muss beachtet werden, dass neue Geräte bei einem Gerätetausch auch erst nach dem Bus Update erkannt werden.

4 Inbetriebnahme

Das folgende Kapitel befasst sich mit der beispielhaften Inbetriebnahme des Butler XT an einer bzw. an zwei 8x8 RGB-LED-Boards. Dafür wird als erstes der Hardwareaufbau erläutert. Anschließend werden alle relevanten Aspekte der Programmierung des Butler XT und der Lichtszenarien behandelt. Zum Schluss wird ein Konzept zur Integration und Inbetriebnahme eines Touch-Bedienfeldes in den Aufbau erarbeitet.

4.1 Hardwareaufbau

Dieses Unterkapitel erläutert den Hardwareaufbau für die Inbetriebnahme des Butler XT an einer RGB-LED-Matrix. Dafür wird zunächst Bezug auf bestimmte Komponenten des Systems genommen.

Damit die DMX-Controller und Netzteile für die LEDs im Hardwareaufbau räumlich voneinander getrennt werden können, gibt es sogenannte „Power/Data-Injectors“. Wie in Abbildung 4.1 zu sehen ist, haben diese einen Eingang für das DMX-Signal über eine RJ45-Schnittstelle und einen Power-Eingang zum Anschluss des Netzteils über ein „Traxon Smart Interconnection Cable“. Der Power/Data-Injector führt Stromversorgung und DMX-Signal auf ein Interconnection Cable zusammen, sodass es jeweils eine Leitung für Spannungsversorgung und Masse, sowie jeweils eine Leitung für DMX+ und DMX- gibt. An dieses Kabel kann dann die LED-Matrix angeschlossen werden.

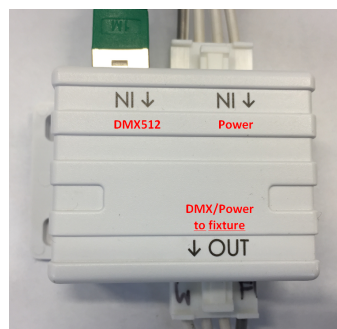


Abbildung 4.1: Traxon power/data injector box

Da das System auch im Standalone-Modus betrieben werden soll und die zwei konfigurierbaren Taster des Butler XT für die Bedienung nicht ausreichend sind, wurde zusätzlich ein Taster-Board für die konfigurierbaren Eingänge hergestellt (Siehe Abb. 4.2). Dafür wurden sechs Platinentaster verwendet, welche eine Spannung von 12 Volt auf die digitalen Eingänge durchschalten. Die Taster müssen dafür nicht entprellt oder über Pull-Down-Widerstände mit Masse verbunden werden. Diese Maßnahmen wurden bereits im Butler XT für die digitalen Eingänge umgesetzt.

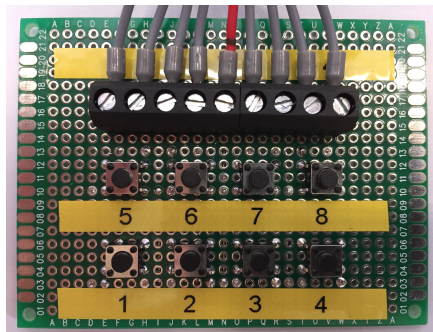


Abbildung 4.2: Taster-Board für die digitalen Eingänge des Butler XT

In Abbildung 4.3 ist nun das Gesamtsystem, also der Aufbau mit Butler XT und einer RGB-LED-Matrix, zu sehen. Der Butler XT wird dabei von einem 24 Volt Gleichspannungsnetzteil versorgt. Über ein Ethernet-Kabel ist er an einen Switch angeschlossen, über welchen er mit dem e:net verbunden ist. Auf diese Weise kann der Butler XT programmiert und im Online-Modus auch gesteuert werden (Siehe Unterkapitel 4.2). An die digitalen Eingänge ist das zuvor vorgestellte Schalterboard angeschlossen. Die RGB-LED-Matrix ist über das Interconnection Cable mit dem Power/Data-Injector verbunden, welcher wiederum mit dem DMX1-Ausgang des Butler XT und dem Injector-Netzteil verbunden ist. Um den Butler XT in den Standalone-Modus zu bringen, muss er lediglich vom e:net getrennt werden. Alternativ kann auch der Programmer (Lighting Application Suite) vom e:net getrennt werden. Wenn nun eine zweite RGB-LED-Matrix angeschlossen werden soll, kann dies nach dem Daisy-Chain-Prinzip erfolgen (Siehe Abb. 2.9). Dafür muss lediglich der Ausgang der ersten LED-Matrix über eine Interconnection Cable mit dem Eingang der zweiten LED-Matrix verbunden werden. Weiterhin muss diese Änderung im Aufbau noch im sogenannten Patchfile berücksichtigt werden (Siehe Unterkapitel 4.2.2).

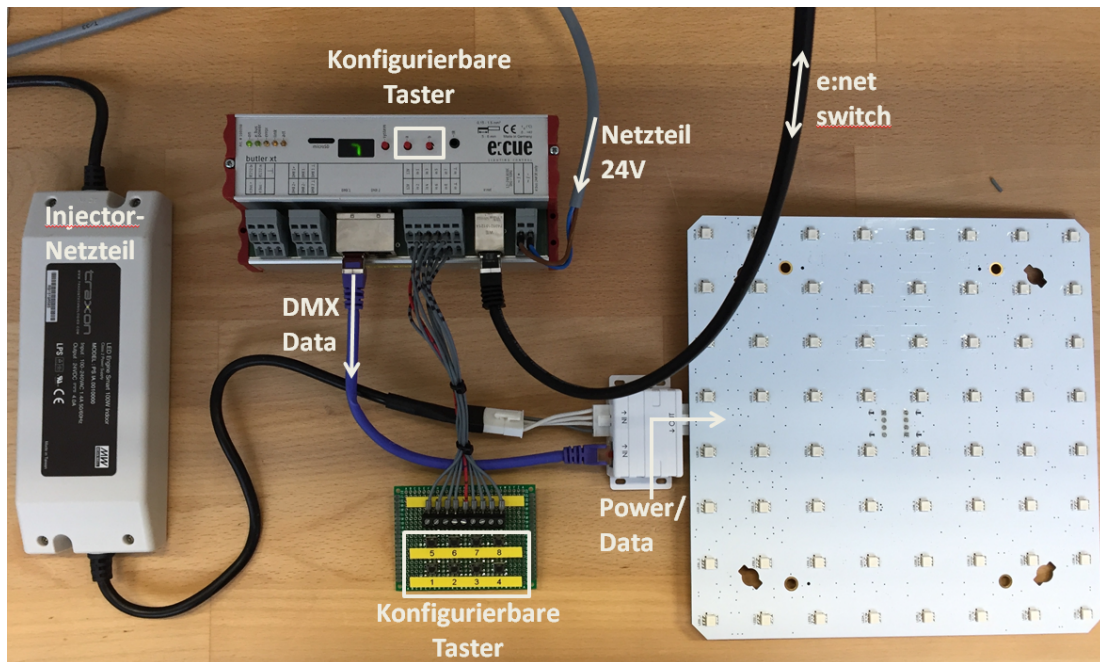


Abbildung 4.3: Aufbau für die Inbetriebnahme des Butler XT an einer 8x8 RGB-LED-Matrix

4.2 Programmierung

Nachfolgend wird die softwareseitige Inbetriebnahme erklärt. Dafür wird die „Lighting Application Suite“ (kurz LAS) der Firma e:cue verwendet. Dieses Softwarepaket beinhaltet diverse Programme, von denen hier lediglich der „Programmer“ und der „Patchelor“ verwendet werden. Außerdem muss der Treiber für die Identifikation des USB-Dongles zur Freischaltung der Software installiert werden. Für diese Arbeit wurde die LAS Version 6.1 verwendet.

Als erstes wird die Einrichtung des Butler XT sowohl für den Online-Modus, als auch für den Standalone-Modus erläutert. Daraufhin wird die Erstellung eines Patchfiles, welches als Ebenbild des realen Hardwareaufbaus dient und schließlich die eigentliche Show-Programmierung behandelt.

4.2.1 Einrichten des Butler XT

Zur Einrichtung des Butler XT sind zunächst ein paar allgemeingültige vorbereitende Schritte notwendig. Anschließend kann die showspezifische Einrichtung des Butler XT erfolgen.

4.2.1.1 Allgemeingültige Einrichtung

Nach dem Anschließen des Butler XT an den Switch, mit welchem auch der PC verbunden ist, muss im Programm die richtige Netzwerk für das e:net ausgewählt werden. Dafür muss im Reiter „Extras“ der Punkt „Application Options“ ausgewählt werden. Alternativ lässt sich dieser auch mit der Tastenkombination Shift+F6 öffnen. Anschließend muss, wie in Abbildung 4.4 markiert ist, der richtige LAN-Controller ausgewählt werden, an dem der Switch für das e:net angeschlossen ist.

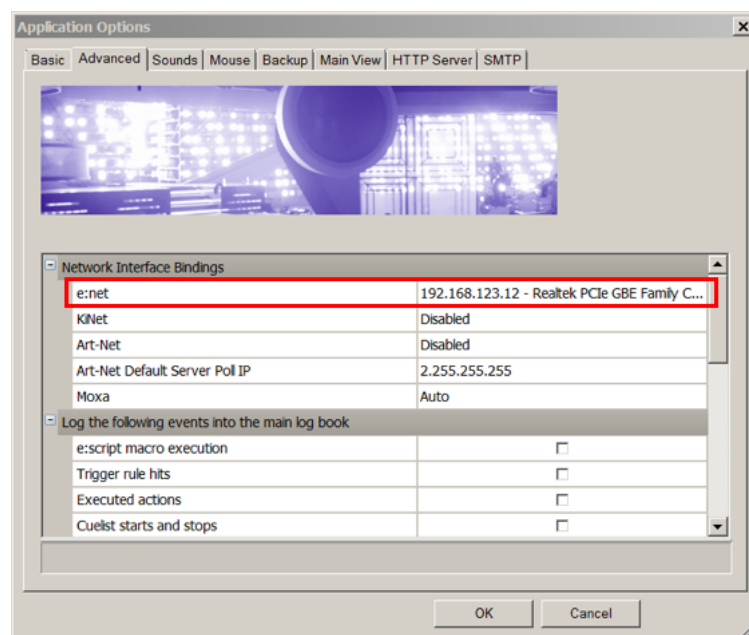


Abbildung 4.4: Auswahl des e:net

Nach der Auswahl des LAN-Controllers folgt die eigentliche Einrichtung des Butler XT. Im oberen linken blauen Fenster des Programms muss nun der Reiter „Network“ ausgewählt werden. Dort sollte nun wie in Abbildung 4.5 der Butler XT sowie alle anderen Butler im Netzwerk angezeigt werden.

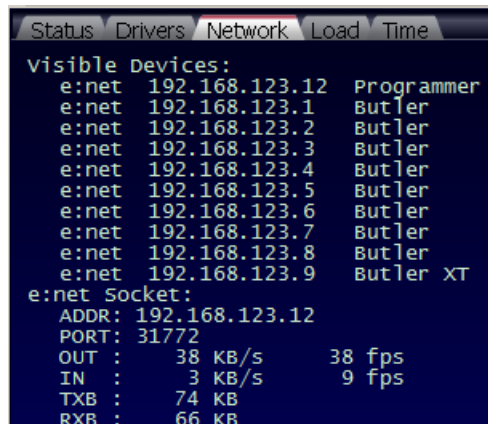


Abbildung 4.5: Verfügbare Geräte im e:net

Mit einem Doppelklick auf den Butler XT in der Liste wird der, in Abbildung 5.3 gezeigte, „Device Setup“ geöffnet, welcher die showübergreifenden Einstellungsmöglichkeiten des Butler XT beinhaltet. Für die Inbetriebnahme des Butler XT allein muss hier nicht viel geändert werden. Als erstes muss die IP-Adresse von dem Standard-Wert „192.168.123.1“ zu einem einzigartigen Wert geändert werden. Diese Änderung muss zwingend nach jedem Anschließen eines neuen Butlers im e:net durchgeführt werden, da jeder Butler anfangs diese IP-Adresse besitzt und Doppelbelegungen zu Fehlern führen. Zusätzlich kann hier bei Bedarf noch der Name des Butler XT geändert werden oder mit „Enable IR“ die Steuerung über die Traxon Infrarotfernbedienung aktiviert werden. Für die Nutzung von Tageszeit-getriggerten Aktionen im Standalone-Modus muss noch die interne Uhr des Butler XT eingestellt werden. Unter „Base Time/Date“ wird die Zeit ohne die Sommerzeitverschiebung eingestellt. mit „DST Start/End“ werden Anfang und Ende der Sommerzeit festgelegt.

4.2.1.2 Showspezifische Einrichtung

Wenn diese grundlegenden Einstellungen getätigt wurden, ist der Butler XT fertig in das e:net integriert. Als nächstes muss er in die Show-Datei eingebunden werden. Dafür muss in der Menüleiste der „Device Manager“ ausgewählt werden. Alle Traxon und e:cue Geräte können automatisch vom Programmer gesucht werden. Dafür muss der in Abbildung 4.6 markierte Knopf gedrückt werden. Der Device Manager sucht dann automatisch alle ihm bekannten e:net und e:bus Geräte. Alle Geräte anderer Hersteller, welche beispielsweise über RS232 angeschlossen werden, können über den Knopf „Add Device“ rechts daneben hinzugefügt werden. Die Änderungen, welche im Device Manager durchgeführt werden, sind lediglich für die zu diesem Zeitpunkt geöffnete Show-Datei gültig. Alle Änderungen im Device Setup (Siehe Abb. 5.3) sind auch darüber hinaus gültig.

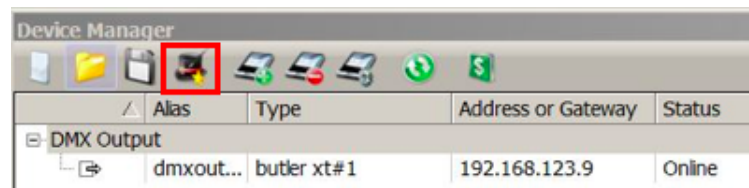


Abbildung 4.6: Butler XT im Device Manager

Ist der Butler XT hinzugefügt und online, können die Taster (Siehe Abb. 4.3) und speziell für den Standalone-Modus auch Trigger konfiguriert werden. Dafür ist zunächst ein Doppelklick auf den Butler XT im Device Manager notwendig. In dem sich öffnenden Menü (Siehe Abb. 4.7) können nun unabhängig voneinander die Taster und digitalen Eingänge sowohl für den Online-Modus, als auch den Standalone-Modus mit Funktionen belegt werden. Das hat den Grund, dass im Online-Modus deutlich mehr Funktionen für die Taster und Eingänge verfügbar sind und alle Aktionen wie Start, Stop, Pause usw. im Online-Modus immer cuelist-spezifisch gewählt werden müssen, da im Online-Modus auch mehrere Cuelists gleichzeitig abgespielt werden können. Cuelists stellen die einzelnen Lichtszenarien dar. Eine genaue Definition für Cuelists ist in Unterkapitel 4.2.3 zu finden. Um den jeweiligen Taster oder digitalen Eingang für die Konfiguration auszuwählen, müssen diese in den Abbildungen im Konfigurationsmenü per Mausklick angewählt werden (Siehe Abb. 4.7). In diesem Konfigurationsmenü können ebenfalls die DMX-Universen für den Butler XT zugewiesen werden. Dieses Thema erlangt jedoch erst im Unterkapitel 5.2.1 Bedeutung.

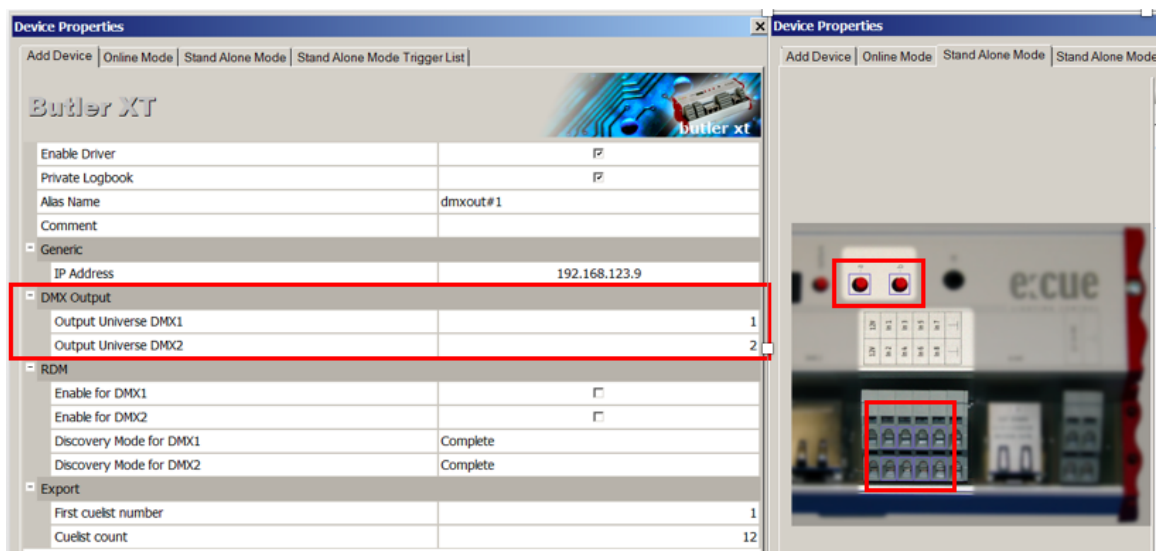


Abbildung 4.7: Showspezifisches Konfigurationsmenü für den Butler XT

Bei den Standalone-Triggern handelt es sich um Aktionen, die zu bestimmten Zeitpunkten (Triggern) ausgelöst werden, wie zu bestimmten Tageszeiten oder am Ende von Cuelists. Sollen beispielsweise die vorhandenen Cuelists im Standalone-Modus automatisch hintereinander abgespielt werden, kann diese Regel hier in Form von Triggern festgelegt werden. Damit ist eine gewisse Automatisierung der Lichtshows im Standalone-Modus realisierbar. Im Online-Modus gibt es hierfür ein gesondertes Trigger-Menü.

Die Lighting Application Suite bietet neben den Triggern zu bestimmten Uhrzeiten auch Trigger für den sogenannten „Sunrise Timer“. Dieser Timer wird dafür genutzt, Aktionen zu Zeitpunkten zu triggern, die von dem Stand der Sonne in der jeweiligen Region abhängen. Um diese Funktion nutzen zu können muss unter „Extras/Show Properties“ im Reiter „Sunrise Timer“ die Region eingestellt werden. Dafür können entweder direkt Längen- und Breitengrade verwendet oder eine Stadt aus der Liste ausgewählt werden. Wenn die richtige Region eingestellt wurde, können folgende Sonnenzeiten als Trigger verwendet werden (vgl. e:cue control GmbH (2015), S.168):

- Twilight AM - Beginn der Morgendämmerung
- Sunrise - Die Sonne wird am Horizont sichtbar
- Noon - Die Sonne hat den höchsten Punkt erreicht
- Sunset - Die Sonne ist hinter dem Horizont verschwunden
- Twilight PM - Es ist komplett dunkel geworden

Wenn alle Konfigurationen abgeschlossen sind, muss im Device Manager noch ein Quick Update durchgeführt werden, damit alle Änderungen auf dem Butler aktiv werden. An dieser Stelle kann der Butler vom Programmierer getrennt werden um in den Standalone-Modus gelangen, vorausgesetzt es wurde schon eine Lichtshow auf den Butler übertragen (Siehe Unterkapitel 4.2.3).

4.2.2 Patchfile

Das Patchfile ist ein elementarer Bestandteil der Lichtshow. Es spiegelt den realen Hardwareaufbau wieder, um die Lichtshow auf diesen anpassen zu können. Ohne das Patchfile erkennt der Programmierer nicht, mit welchen Lichtelementen er arbeiten kann und wie diese räumlich angeordnet sind.

Außerdem besitzt der Programmierer ein Preview-Fenster, in dem der Hardwareaufbau nach Einbinden des Patchfiles simuliert dargestellt wird. Dieses Fenster ist in der Menüleiste des

Programmers im Reiter „View“ zu finden. Mit diesem Preview-Fenster ist es möglich, Lichtshows auch zu simulieren, wenn die reelle Hardware noch gar nicht angeschlossen ist (vgl. Abb. 4.8).

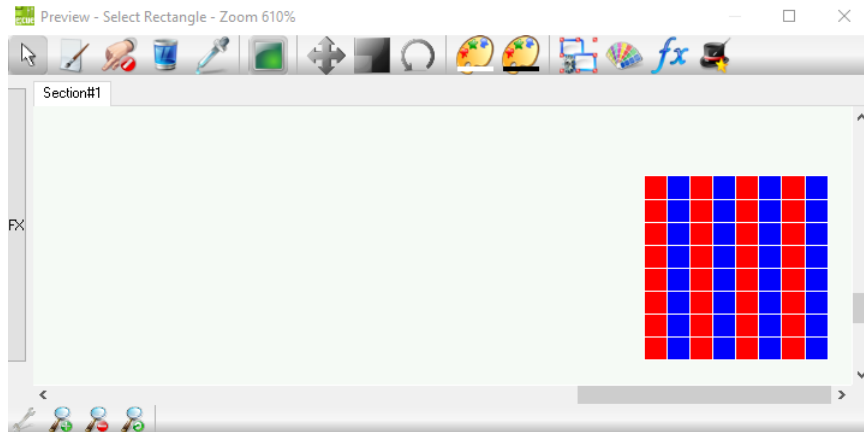


Abbildung 4.8: Preview eines simulierten Patchfiles mit einer RGB-LED-Matrix

Erstellt werden Patchfiles im sogenannten „Patchelor“. Dabei gibt es zwei Phasen des „Patching“: die Build- und die Patch-Phase. In der Build-Phase werden lediglich die Lichtelemente platziert und die geometrischen Rahmenbedingungen festgelegt. In der Patch-Phase werden die Lichtelemente wie in Abbildung 4.9 miteinander verbunden. Diese Verbindung repräsentiert später die DMX-Adressierung der Lichtelemente, wie es dem Unterkapitel 2.2.4 zu entnehmen ist.

Wie Abbildung 4.9 zeigt, ist die Erstellung des Patchfiles für die Inbetriebnahme des Butler XT an ein oder zwei LED-Matrizen sehr simpel und überschaubar. Bei komplexeren Systemen mit mehr Lichtelementen und mehreren DMX-Universen wird dieser Prozess jedoch etwas komplizierter, wie das Unterkapitel 5.2.2 zeigen wird.

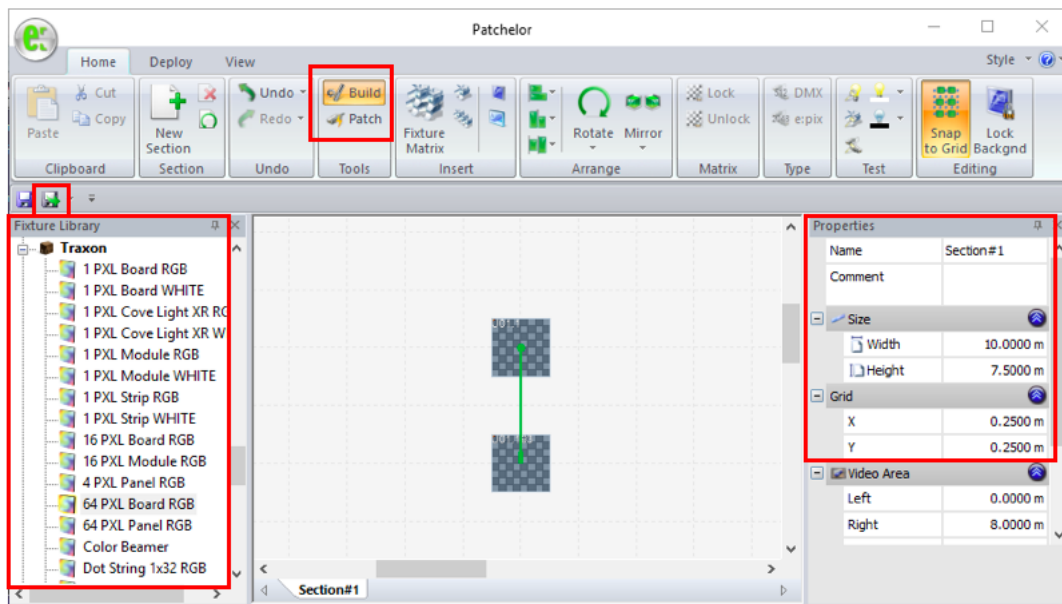


Abbildung 4.9: Patching zweier LED-Matrizen

Die Lichtelemente können über das Bibliotheksfenster links im Patchelor per drag and drop in den Aktionsbereich des Patchelors gezogen werden. Dafür muss der „Build“-Knopf in der Menüleiste des Patchelors gedrückt sein. Für diese Inbetriebnahme wird aus der Bibliothek das „64 Pixel RGB-Board“ von Traxon Technologies ausgewählt. Im Fenster auf der rechten Seite im Patchelor können die geometrischen Ausmaße des Bereichs festgelegt werden. Wenn alle Elemente platziert sind, können nach Drücken der „Patch“-Taste in der Menüleiste die Elemente miteinander verbunden (adressiert) werden. Ein Anklicken der Elemente öffnet dann ein Fenster, in welchem die Adressbereiche und Screen-IDs für diese angezeigt werden. Diese Wichtigkeit dieser Optionen wird in Unterkapitel 5.2.2 deutlich.

Wenn das Patchfile fertiggestellt ist, muss es lediglich noch als „patx“ Datei über den in Abbildung 4.9 markierten „Exportieren“-Knopf in der Menüleiste exportiert und später im Programmierer über „File/Import/Patch“ importiert werden.

4.2.3 Show-Programmierung

Die Lighting Application Suite bietet viele verschiedene Möglichkeiten der Show-Programmierung, wovon hier die wichtigsten, für die für die Nutzung in dieser Arbeit relevanten Möglichkeiten, erklärt werden.

Bei der Show-Programmierung muss in einigen Punkten noch unterschieden werden, ob die Show für den Online- oder Standalone-Modus erstellt wird. An den jeweiligen Stellen in diesem Kapitel wird darauf hingewiesen.

4.2.3.1 Allgemeines

Zunächst bedarf es einiger allgemeiner Informationen zu „Cuelists“, dem Aufbau von Lichtshows und ihrer Übertragung auf den Butler XT.

Lichtshows bestehen aus den oben erwähnten „Cuelists“, welche wiederum aus einzelnen „Cues“ bestehen. Ein Cue beschreibt meist nur einen einzelnen Zustand der Lichtelemente mit zeitlichen Merkmalen wie Dauer des Zustands oder seine Ein- bzw. Ausblendzeit. Außerdem ist es möglich, einem Cue für den Gebrauch im Online-Modus verschiedene Aktionen wie das Starten oder Stoppen einer anderen Cuelist zuzuordnen. In Abbildung 4.10 ist das Konfigurationsfenster zu sehen, welches durch einen Doppelklick auf den jeweiligen Cue geöffnet wird. In diesem Fenster findet sich außerdem noch das Feld „Control“. Dort gibt es vier verschiedene Auswahlmöglichkeiten:

- Manual - Cue kann nur manuell beendet werden
- Wait - Cue endet nach einer vorgegebenen Zeit
- Timecode - Cue endet bei einem bestimmten Timecode (Timecode kann in Cue Actions oder Trigger Actions gestartet oder gestoppt werden)
- Cuelist End - Cue wird beim Enden einer vorgegebenen Cuelist beendet

Für den Standalone-Modus sind lediglich die Möglichkeiten „Wait“ und „Manual“ relevant.

Ein Cue kann beispielsweise bewirken, dass jede zweite RGB-LED für 5 Sekunden auf voller Helligkeit leuchtet und am Anfang beziehungsweise am Ende die Helligkeit innerhalb von einer Sekunde hoch- bzw. heruntergefahren wird. In Spezialfällen kann ein Cue sogar einen komplexen, dynamischen Lichteffect beinhalten, wie es in Unterkapitel 4.2.3.4 beschrieben wird.

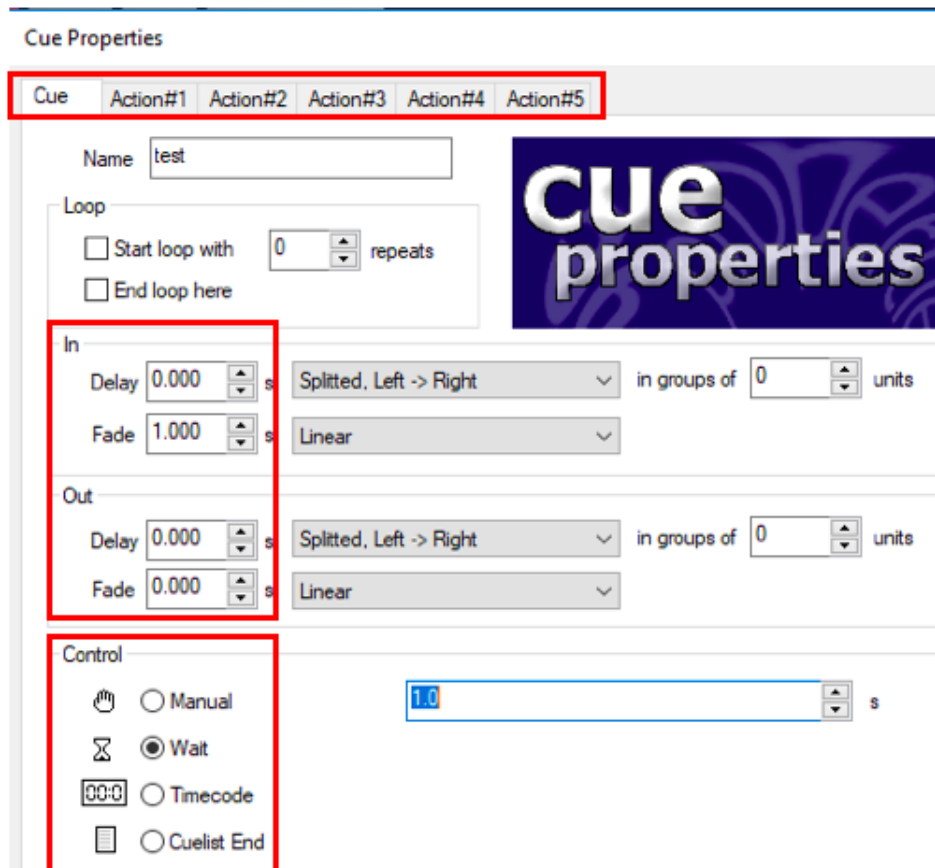


Abbildung 4.10: Cue Konfiguration

Die Cues werden letztendlich in einer Cuelist aneinandergereiht. Die gesamte Licht-Show besteht dann aus einer oder mehreren Cuelists, welche mit bestimmten Triggern automatisch in der gewünschten Reihenfolge abgespielt werden können. Das Erreichen des Triggermenüs erfolgt für den Standalone-Modus wie in Unterkapitel 4.2.1.2 beschrieben. Das Triggermenü des Online-Modus ist über „View/Trigger“ in der Menüleiste erreichbar. Beim Starten des Programmers werden im unteren Bereich des Programmers automatisch 12 Cuelists angelegt, welche sich durch einen Doppelklick öffnen lassen. Sollten mehr Cuelists benötigt werden, lassen sich diese über „Cuelist/Cuelist Directory“ in der Menüleiste hinzufügen.

Die Einstellungen der Cuelists sind durch einen Rechtsklick auf die jeweilige Cuelist verfügbar. Abbildung 4.11 zeigt zwei Reiter dieses Einstellungsfensters. Im Reiter „Main“ ist vor allem die Option „Auto release on end“ interessant. Sie bewirkt, dass die Cuelist am Ende automatisch freigegeben wird und somit über den Reiter „Release Action“ eine Aktion für

den Online-Modus zu diesem Zeitpunkt festgelegt werden kann. Aktionen beim Release einer Cuelist im Standalone-Modus müssen über das Standalone-Trigger Menü in der Butler XT Konfiguration eingestellt werden. Im Reiter „Extras“ können Einstellungen für den „Chaser Mode“ vorgenommen werden, welcher jedoch erst in den nachfolgenden Unterkapiteln thematisiert wird. Ansonsten lässt sich hier ein Zeitfenster für das Abspielen der Cuelist festlegen.

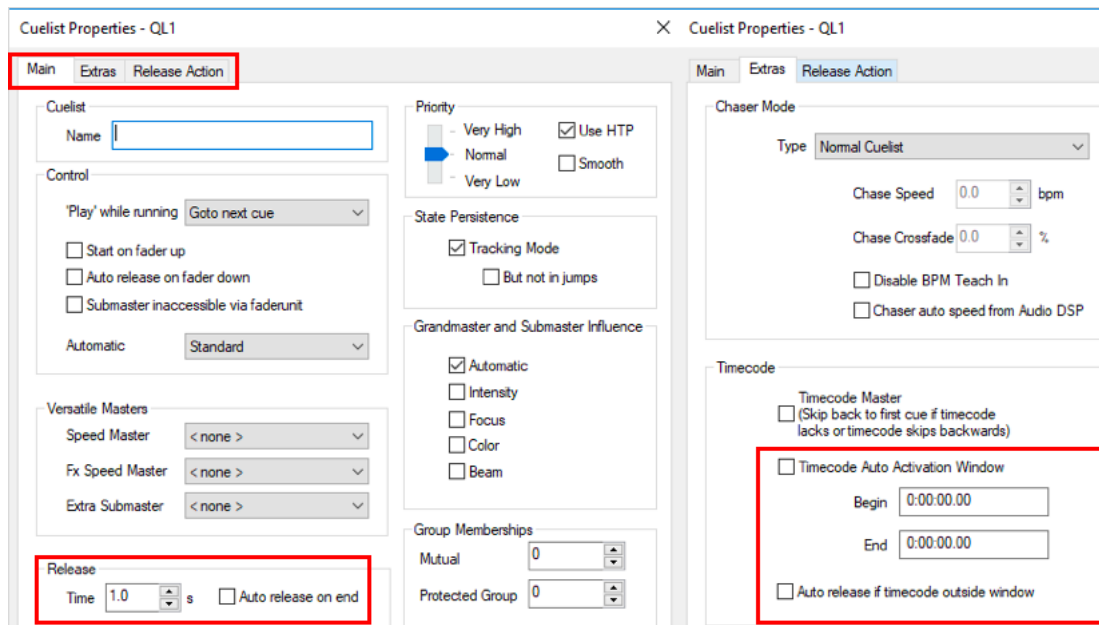


Abbildung 4.11: Cuelist Konfiguration

Für die Übertragung der Cuelists auf den Butler gibt es in der oberen Menüleiste des Programmers einen „Export to Device“-Knopf. Dieser öffnet das in Abbildung 4.12 gezeigte Fenster, wo die Anzahl der zu übertragenden Cuelists und die Zielgeräte ausgewählt werden können. Es können bis zu 99 Cuelists auf einen Butler übertragen werden, welche alle auf dessen microSD-Karte gespeichert werden.

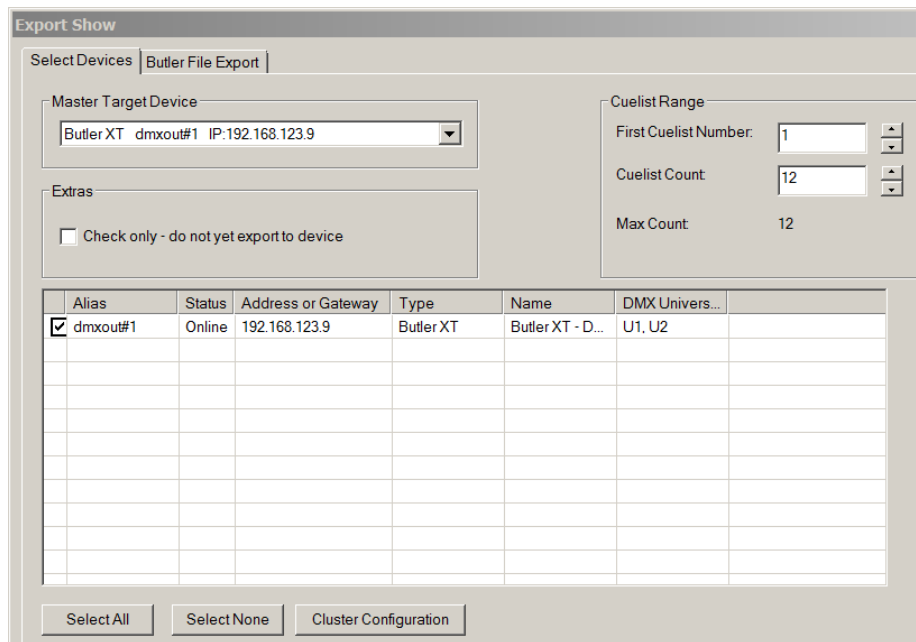


Abbildung 4.12: Übertragung von Shows auf den Butler XT

Wichtig!: Es sollte noch einmal gesondert darauf hingewiesen werden, dass der Online-Modus und der Standalone-Modus voneinander unabhängige Trigger-Menüs besitzen. Dazu gehört außerdem die Information, dass diverse Action-Optionen wie „Release Actions“ (Siehe Abb. 4.11) bei Cuelists oder Aktionen bei einzelnen Cues (Siehe Abb. 4.10) für den Standalone-Modus immer im „Standalone Trigger“ Menü in den Einstellungen des Butler XT realisiert werden müssen. Alle Aktionen, die in den Aktionsmenüs der Cuelists und Cues eingestellt werden, sind für den Standalone-Modus nicht gültig und können im schlimmsten Fall zu Fehlern bei der Übertragung der Cuelists auf einen Butler führen.

4.2.3.2 Freie Programmierung

Die freie Programmierung ist nur dann sinnvoll, wenn entweder sehr einfache Beleuchtungsszenarien mit wenig Cues oder sehr spezielle Szenarien mit Effekten oder Zeichen benötigt werden, welche mit der Programmierung durch Wizards und LiveFX Effekte nicht sinnvoll realisierbar sind.

Die freie Programmierung eines Beleuchtungsszenarios ähnelt der Erstellung eines Stop-Motion-Films. Es wird zuerst wie in Abbildung 4.13 die zu beschreibende Cuelist geöffnet. Dann können entweder in der Unit-Liste oder im Preview-Fenster die gewünschten LEDs

(Units) ausgewählt werden. Anschließend werden die Farben für die jeweiligen Units entweder wieder über die RGB-Fader oder über die Farbpalette im Preview Fenster festgelegt. Daraufhin muss der Cue aufgenommen werden. Das geschieht mit dem „Record Cue“-Knopf in der Cuelist. Als letztes können nun noch die Einstellungen für den Cue getätigt werden, wie es in Unterkapitel 4.2.3.1 erklärt wird. Auf diese Weise können nun Beleuchtungsszenarien Bild für Bild erstellt werden. Wenn eine Schleife über bestimmte Cues gelegt werden soll, müssen diese markiert und über die rechte Maustaste der Punkt „Create Loop“ ausgewählt werden.

Wichtig!: Vor jedem Cue Record müssen alle Units (auch diejenigen, die im aktuellen Frame lediglich ausgeschaltet sind) angewählt werden, da ansonsten Units aus dem vorherigen Frame unerwünschterweise ihren Zustand im aktuellen Frame beibehalten, da sie sich nicht von alleine abschalten.

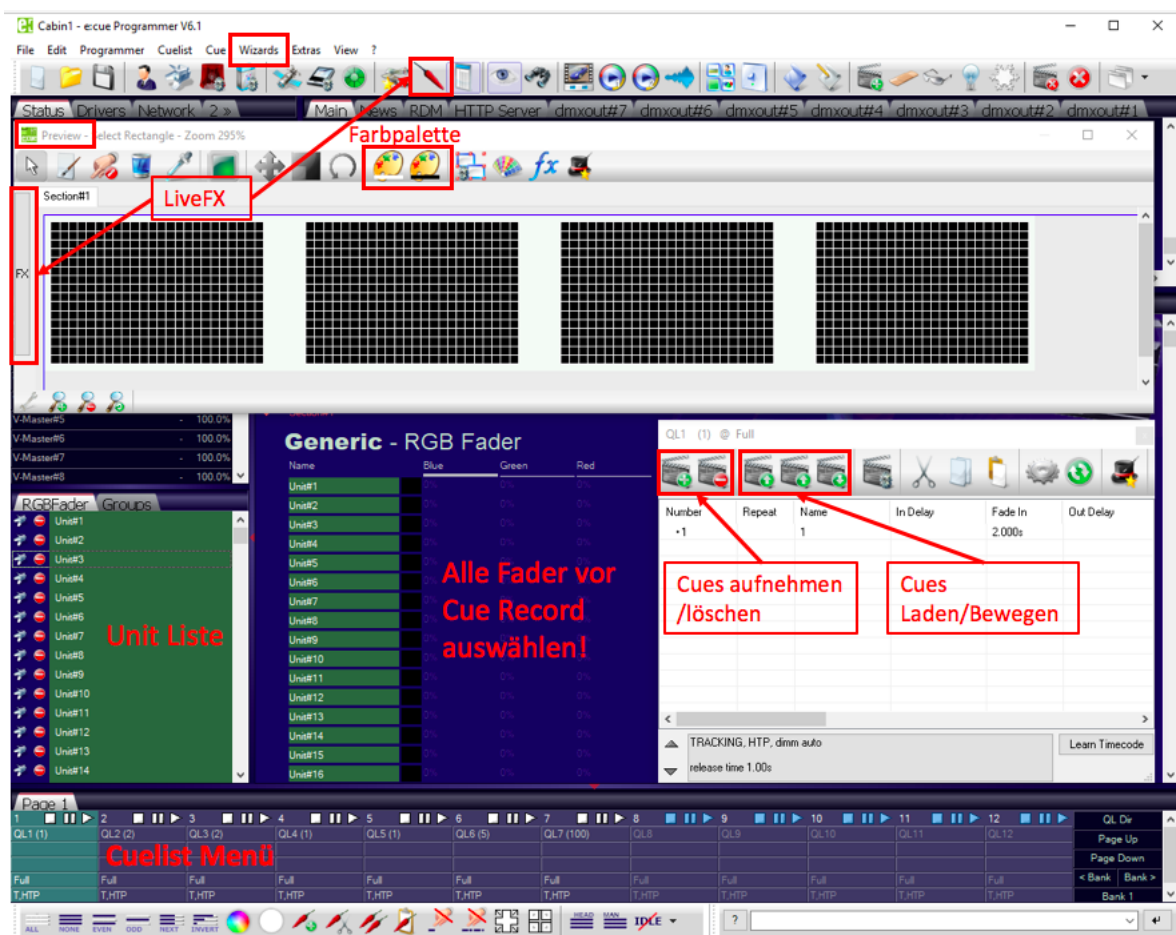


Abbildung 4.13: Programmierer mit Cuelist und Preview-Fenster

Die per Hand erstellte Cuelist kann anschließend auch als sogenannter „Chaser“ (Strahl) konfiguriert werden. Dazu muss diese in den Cuelist-Einstellungen im Reiter „Extras“ in den Chaser-Modus gebracht werden (Siehe Unterkapitel 4.2.3.1). Diese Einstellung bewirkt, dass die Cuelist Bild für Bild mit einer einstellbaren Geschwindigkeit (bpm - Beats per Minute) und einem einstellbaren Fading abgespielt wird. Diese Art des Abspielens der Cuelist lässt sich selbstverständlich auch manuell über Fade-In-Zeiten und Wait-Zeiten in den Cue-Einstellungen realisieren. Der große Vorteil des Chaser-Modus ist jedoch, dass dort im Online-Modus das Fading (Abklingzeit der LED-Helligkeit) und die Geschwindigkeit durch Gedrückthalten der linken Maustaste auf der gewünschten Einstellung in der Cuelist im unteren Cuelist-Menü (Siehe Abbildung 4.13) verändert werden kann.

4.2.3.3 Programmierung mit Wizards

Wizards sind Hilfsprogramme innerhalb des Programmers, welche die Erstellung von bestimmten Beleuchtungsszenarien und Effekten vereinfachen sollen. Unter den Wizards gibt es drei, die für diese Arbeit relevant sind. Diese drei Wizards werden in diesem Unterkapitel in Funktion und Anwendung erklärt.

Der erste dieser Wizards ist der Chaser-Wizard. Der Chaser-Modus wurde im vorherigen Kapitel bereits erläutert. Der Wizard ist nun ein Tool, welches die Programmierung eines solchen Chasers automatisiert, sodass er nicht Frame für Frame manuell erstellt werden muss. Der Chaser-Wizard benötigt lediglich den Start-Frame und erstellt alle folgenden Frames von selbst.

Der einfachste Weg, einen Chaser zu erstellen, ist über das Preview-Fenster. Dort werden zunächst alle Units ausgewählt, die den Chaser darstellen sollen. Diesen Units muss dann eine Farbe zugewiesen werden. Anschließend müssen alle Units und die dazugehörigen Fader ausgewählt werden, die zu dem Bereich gehören sollen, in dem sich der Chaser bewegt. Zuletzt wird der Chaser-Wizard über „Wizards/Chaser Wizard“ in der oberen Menüleiste (Siehe Abb. 4.13) ausgewählt. In dem sich öffnenden Fenster (Siehe Abb. 4.14) wird im Feld „Steps“ die Anzahl der Schritte für den Chaser bestimmt, welche letztendlich die Anzahl der Cues (Frames) darstellt. Das Feld „Offset“ besagt, wie viele Units (also jeweils drei DMX-Adressen) mit jedem Step übersprungen werden. Wenn beispielsweise in einer 8x8 LED-Matrix ein Streifen von acht LEDs leuchten soll und mit jedem Step nicht nur einen Pixel sondern eine komplette Zeile oder Spalte weitergehen soll, muss ein Offset von acht Pixeln eingestellt werden.

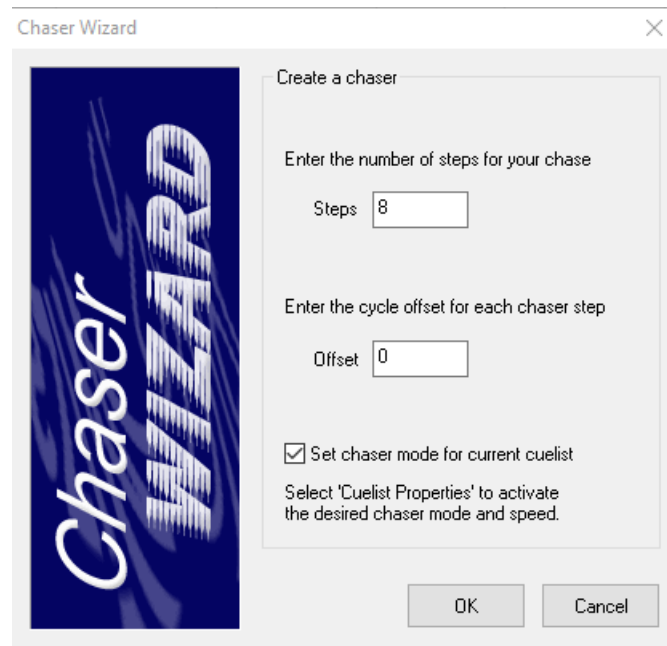


Abbildung 4.14: Chaser Wizard

Bei der Nutzung von mehreren LED-Matrizen ist der Chaser-Wizard jedoch nicht empfehlenswert, da dieser immer die Kanäle jedes DMX-Universums für sich durchläuft. Das macht die Arbeit mit dem Chaser-Wizard bei mehreren LED-Matrizen sehr kompliziert. Da Chaser auch in LiveFX realisierbar sind, sollte in jenem Fall auf dieses Tool zurückgegriffen werden. Sollte dieser Wizard dennoch bei mehreren LED-Matrizen verwendet werden, ist darauf zu achten, im Patchfile die richtige Reihenfolge der Screen-IDs einzuhalten (Siehe Unterkapitel 5.2.2).

Der Bitmap-Wizard ist ein Tool zum Rendern von Bildern auf eine LED-Matrix. Zusätzlich können den Bildern noch Bewegungseffekte hinzugefügt werden. Dafür muss im Feld „Filename“ als erstes das gewünschte Bild ausgewählt werden (Siehe Abb. 4.15). Das Bild wird dann als Vorschau angezeigt. Darunter muss das Feld „color channels“ ausgewählt sein, um Farben darstellen zu können.

Im Reiter „Animation“ können nun die Bewegungseffekte festgelegt werden. Über „Rotation“ können Anfangs- und Endwinkel festgelegt werden, um eine Drehbewegung des Bildes zu erreichen. Die Felder für „Zoom“ bewirken ein Rein- bzw. Rauszoomen des Bildes im Laufe der Cuelist. Der Offset dient nur zum Verschieben des Bildes auf der LED-Matrix.

Mit „Outside“ wird die Hintergrundfarbe festgelegt, welche insbesondere beim Bewegen des Bildes mit den oben genannten Funktionen sichtbar wird. Das Verhältnis des „Render XX cues“-Wertes zu dem „Duration“-Wert gibt an, wie flüssig Bewegungen dargestellt werden. Dieses Verhältnis stellt den fps-Wert (Frames per Second) dar.

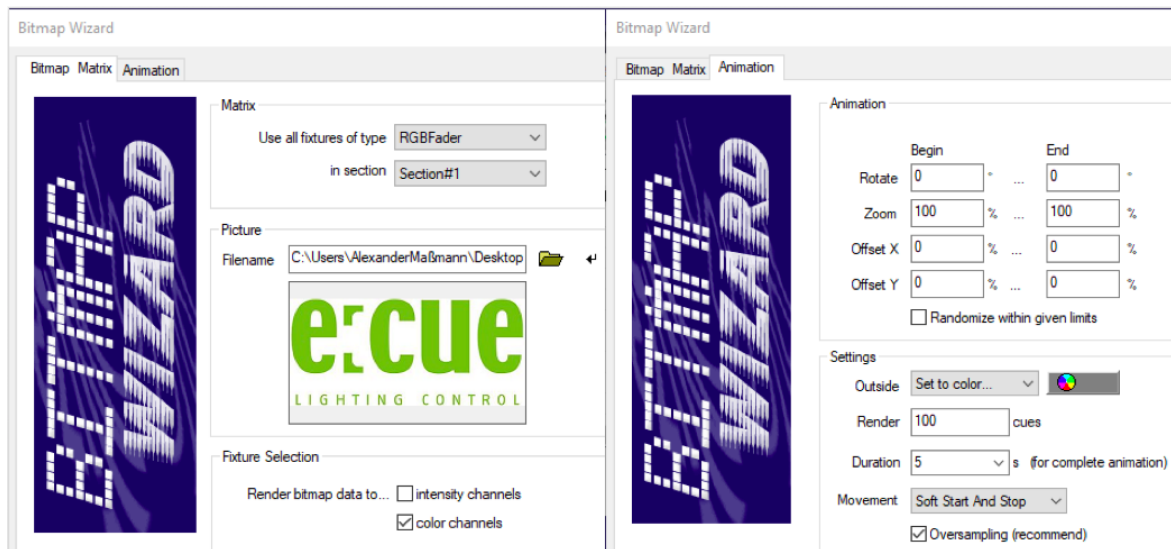


Abbildung 4.15: Bitmap-Wizard

Wie gut sich ein Bild auf die LED-Matrix anpassen lässt, hängt davon ab, wie detailreich das Bild ist und wie viele Pixel in der Matrix zur Verfügung stehen. Abbildung 4.16 zeigt, dass der große „e:cue“-Schriftzug sich verhältnismäßig gut darstellen lässt, der deutlich kleinere Schriftzug darunter bei der gegebenen Pixelanzahl jedoch nicht lesbar ist.

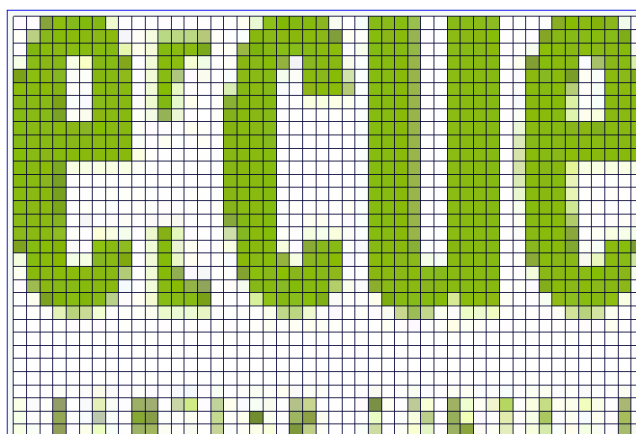


Abbildung 4.16: Mit dem Bitmap-Wizard gerendertes Bild

Der letzte relevante Wizard ist der Scroller-Wizard. Er dient zur automatischen Erzeugung von Schriftzügen und Lauftexten auf LED-Matrizen. Seit der Lighting Application Suite 6 gehört diese Funktion zwar zu den LiveFX-Effekten, ist aber immernoch im Reiter „Wizards“ in der obersten Menüleiste zu finden. So ein Scroller-Effekt kann jedoch auch direkt im LiveFX-Menü hinzugefügt werden.

Das LiveFX-Fenster mit dem Scroll-Wizard-Effekt ist in Abbildung 4.17 zu sehen. Dort wird zuerst der darzustellende Text eingegeben. Im Reiter „Font Properties“ werden Farb- und Schrifteneinstellungen vorgenommen und es kann dort sogar ein Hintergrundbild eingefügt werden. In den „Position Properties“ ist es vor allem wichtig, die Matrix-Größe in Pixeln anzugeben, damit der Text richtig angezeigt wird. Zuletzt wird in den „Movement Properties“ die Scroll-Richtung und die Scroll-Geschwindigkeit angegeben.

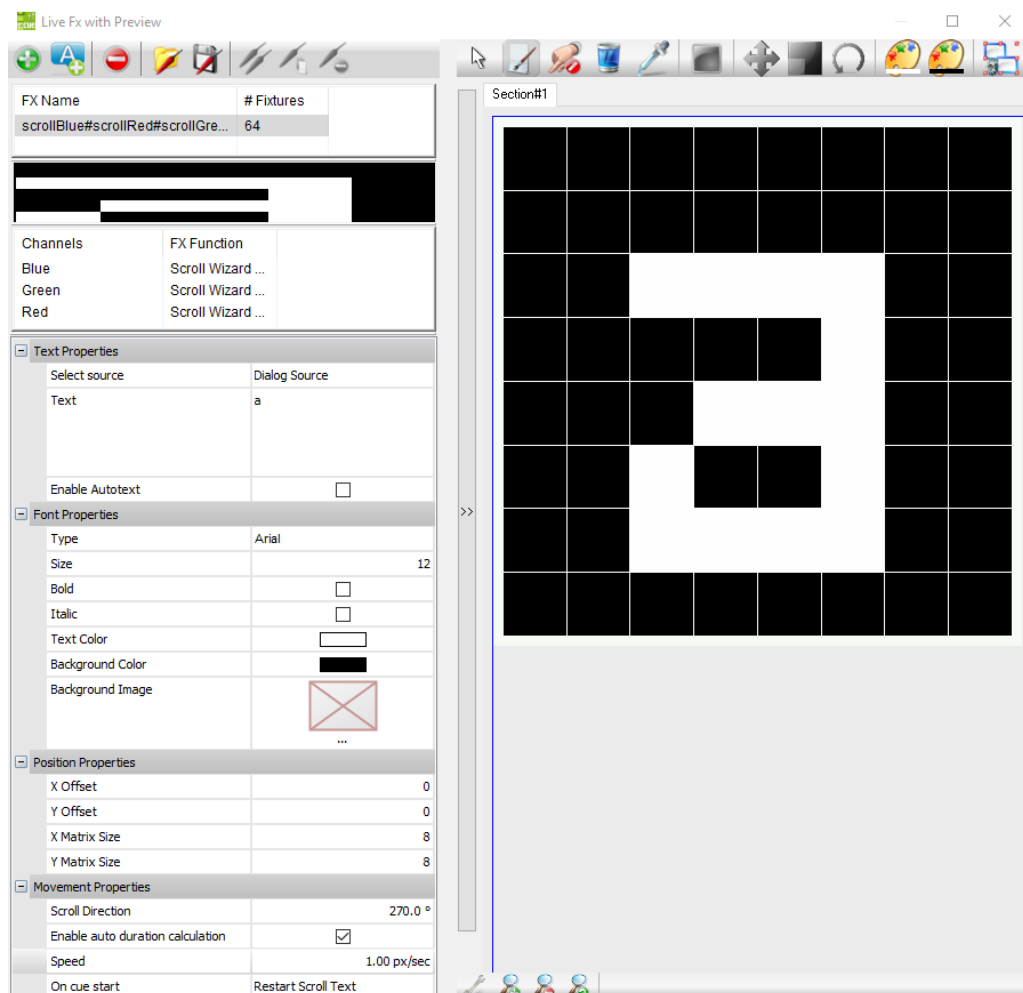


Abbildung 4.17: Scroller-Wizard im LiveFX-Tool

4.2.3.4 LiveFX Effektprogrammierung

Das wohl umfangreichste und intelligenteste Tool zur Effekt-Programmierung in der Lighting Application Suite ist das LiveFX-Tool, nicht zuletzt, da es Effekte nicht DMX-kanalabhängig, sondern auf die räumlichen Koordinaten des Patchfiles berechnet. In dieser Arbeit wird das grundlegenden Wissen zur Nutzung dieses Tools vermittelt. Trotzdem braucht vor allem dieses Tool Übung, um es effektiv nutzen zu können.

Abbildung 4.18 zeigt das LiveFX-Menü, welches seit LAS Version 6.0 fest mit dem Preview-Fenster verbunden ist. Für die Programmierung müssen wieder die betroffenen Units und ihre Fader ausgewählt werden. Anschließend wird ihnen ein LiveFX-Effekt hinzugefügt. Nun werden im Feld „Channels“ die drei verschiedenen Farbkanäle angezeigt, denen jeweils unabhängig voneinander ein LiveFX-Effekt zugeteilt werden kann.

Es gibt eine breite Auswahl an LiveFX-Effekten, von Sinus-, Cosinus-, Dreieck- oder Rechteckfunktionen über verschiedene audiosensitive Funktionen (der Butler XT besitzt einen internen digitalen Audio-Signalprozessor), bis hin zu einer Stroboskopfunktion.

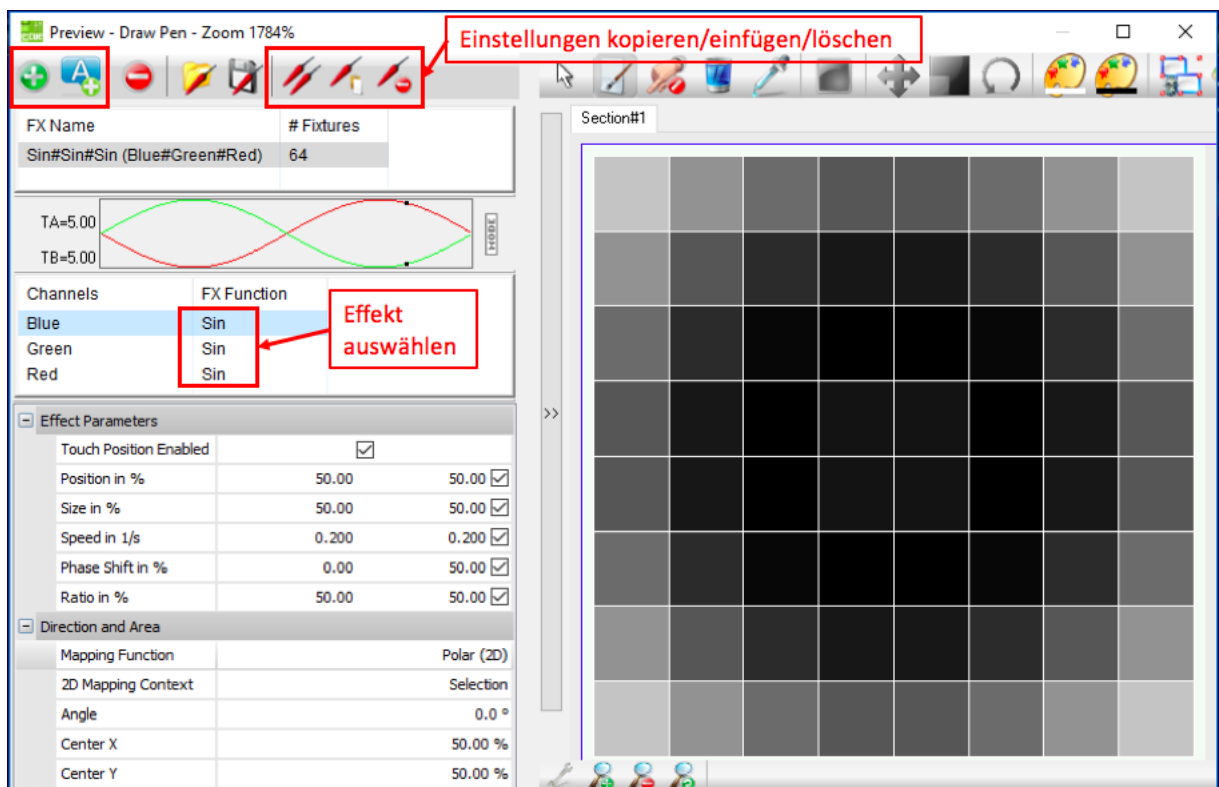


Abbildung 4.18: Erstellen von LiveFX Effekten

Wie in Abbildung 4.18 zu sehen ist, gibt es verschiedene Effektparameter, um den jeweiligen Effekt zu individualisieren. Das Feld „Position“ erscheint erst, wenn „Touch Position Enabled“ gewählt wird. Damit wird die Effektkurve auf ihrer Y-Achse verschoben. Außerdem können Amplitudenhöhe (Size), Geschwindigkeit (Speed), Phasenverschiebung (Phase Shift) und das Verhältnis von positiver zu negativer Halbwelle (Ratio) eingestellt werden. Jede dieser Einstellungen gibt es doppelt, da immer zwei Wellen pro Effekt gleichzeitig aktiv sind, welche jedoch individuell eingestellt werden können. Dadurch kann den Effekten Dynamik verliehen werden, da anhand der Nutzung zweier Wellen mit unterschiedlicher Geschwindigkeit oder Phasenverschiebung räumlich Verstärkungen und Auslöschungen erzeugt werden können.

Der Bereich „Direction and Area“ bietet ebenfalls einige wichtige Funktionen. Das „Center“ gibt den Effekt-Mittelpunkt an, welcher von der „Mapping Function“ genutzt wird. Das Mapping gibt an, ob der Effekt in der LED-Matrix beispielsweise von Links nach Rechts (Linear), von Außen nach Innen (Polar), im Kreis um den Mittelpunkt (Radar) oder spiralförmig um den Mittelpunkt (Spiral) läuft. Außerdem sind zufällige (Randomized) Mappings möglich. Der „Angle“ gibt an, in welchem Winkel der Effekt um den Mittelpunkt gekippt wird.

Wenn der Effekt fertig programmiert wurde, kann er einfach wie auch bei der freien Programmierung über „Record Cue“ in einer Cuelist gespeichert werden. Der gravierende Unterschied zu den Cues, die bei der freien Programmierung aufgenommen werden, ist hier jedoch, dass der Cue nicht nur einen einzelnen Frame enthält, sondern einen komplexen Lichteffekt.

Bei Lichteffekten, die mit dem LiveFX-Tool erstellt werden, gibt es außerdem noch einen bedeutenden Unterschied bei der Nutzung des Online- und Standalone-Modus zu beachten. Bei der Aufnahme eines Cues aus einem LiveFX-Effekt wird, sofern es beim Aufnehmen des Cues nicht anders ausgewählt wurde, die Einstellung „Control“ auf „Manual“ gestellt. Diese Einstellung funktioniert für den Online-Modus sehr gut, da der Effekt dann automatisch, bis er gestoppt wird, immer wieder durchläuft und die Übergänge zwischen den Durchläufen fließend berechnet werden. Wird der Effekt nun aber im Standalone-Modus genutzt, spielen die Butler mit dieser Einstellung lediglich einen Frame des Effekts ab und somit hält der Effekt nach kurzer Zeit selbstständig an. Die einzige Möglichkeit, LiveFX-Effekte im Standalone-Modus vernünftig abzuspielen, ist den dazugehörigen Cue mit einer „Wait“-Zeit zu versehen, welche angibt, wie lange der Effekt abgespielt werden soll. Der Programmierer berechnet dann die Frames des Cues für diesen Zeitraum, welche anschließend auf den Butler übertragen werden können.

4.3 Einbinden eines Touch-Bedienfeldes

Im Folgenden werden Konzepte vorgestellt, die eine Integration eines Touch-Bedienfeldes als Steuerung für den Butler XT in das Gesamtsystem beschreiben.

Zunächst muss geklärt werden, welche Bedienfelder sich in das System integrieren lassen. Der Butler XT hat zwei Schnittstellen für den Anschluss von externen Bedienelementen. Zum einen besitzt er eine RS232-Schnittstelle und zum anderen die im Kapitel 3.5 beschriebene e:bus-Schnittstelle. Die Einrichtung von Bedienfeldern beider Schnittstellen wird im Folgenden erläutert.

4.3.1 RS232-Bedienfeld

Prinzipiell können Bedienfelder jedes Herstellers, die eine RS232-Schnittstelle besitzen und programmierbar sind, genutzt werden. Bei der Nutzung eines RS232 Touchscreens erfolgt der Anschluss und die Einrichtung jedoch für Online- und Standalone-Modus verschieden. Dazu kommt, dass im Standalone-Modus nur eine begrenzte Anzahl an Befehlen über RS232 möglich sind, welche nicht das gesamte Spektrum der Standalone-Aktionen des Butler XT abdecken.

Die Nutzung des Bedienfeldes im Standalone-Modus ist unkompliziert. Hierfür muss das Bedienfeld zunächst mit den in Abbildung 3.1 gezeigten RS232-Anschlüssen verbunden werden. Dafür kann beispielsweise das serielle Kabel genutzt werden, welches als optionales Zubehör für den Butler XT verfügbar ist. Wenn der Anschluss fertig ist, erfolgt die Einrichtung lediglich auf dem Touch-Bedienfeld. Dort müssen bestimmte Parameter eingestellt werden, die zumindest im Standalone-Modus nicht variabel sind. Nachfolgend werden diese Parameter aufgelistet (vgl. e:cue control GmbH (2011b), S.11):

- Datenrate: 9600 Baud
- 8 Datenbits
- keine Parität
- 1 Stopbit

Mit diesen Parametern können dann folgende Befehle über das Bedienfeld an den Butler XT gesendet werden:

- Play Cuelist: PCxx<CR> xx = Nr. der Cuelist
- Pause/Resume: PP<CR>
- Next: NX<CR>
- Previous: PC<CR>
- Intensity: INxxx<CR> xxx = Helligkeit in Prozent

Wie diesen Befehlen bereits entnommen werden kann, ist keine Auswahl des DMX-Ausgangs für die jeweiligen Aktionen möglich. Alle Aktionen, die über RS232 an den Butler gesendet werden, gelten immer für beide DMX-Ausgänge.

Für die Nutzung des Online-Modus ist die Einrichtung des RS232-Bedienfeldes etwas umständlicher. Dafür sind im Online-Modus wesentlich mehr Aktionen verfügbar, die über das Bedienfeld getriggert werden können. Der erste große Unterschied zwischen Online- und Standalone-Modus ist, dass das Bedienfeld für den Online-Modus nicht an den Butler XT, sondern an den PC mit der Lighting Application Suite angeschlossen werden muss. Das setzt in erster Linie voraus, dass der PC eine serielle RS232-Schnittstelle besitzt. Die Lighting Application Suite ermöglicht einen direkten Zugriff auf diese Schnittstelle, falls sie vorhanden ist. Für den Anschluss kann ein RS232 Standard-Kabel mit einem 9-Pin Stecker verwendet werden. Genutzt werden von der LAS jedoch nur die RxD-, TxD- und Ground-Pins (vgl. e:cue control GmbH (2015), S.218). Nach dem Anschluss des Bedienfeldes muss sichergestellt werden, dass der jeweilige COM-Port (Port für serielle Datenübertragung) nicht von einem anderen Programm oder einer Windows-Funktion genutzt wird.

Wenn diese vorbereitenden Schritte abgeschlossen sind, muss das Bedienfeld über den Device Manager manuell hinzugefügt werden. Dafür muss über die Funktion „Add Device“ ein „Serial Device“ hinzugefügt werden. Abbildung 4.19 zeigt die Einstellmöglichkeiten beim Hinzufügen des seriellen Interfaces. Hier muss der serielle Port ausgewählt werden und die Einstellungen „Bit Rate“, „Data Bits“, „Parity“ und „Stop Bits“ auf die jeweiligen Einstellungen in dem verwendeten RS-232 Gerät angepasst werden. Wenn diese Einstellungen nicht genau übereinstimmen, kommt keine funktionierende Übertragung zustande.

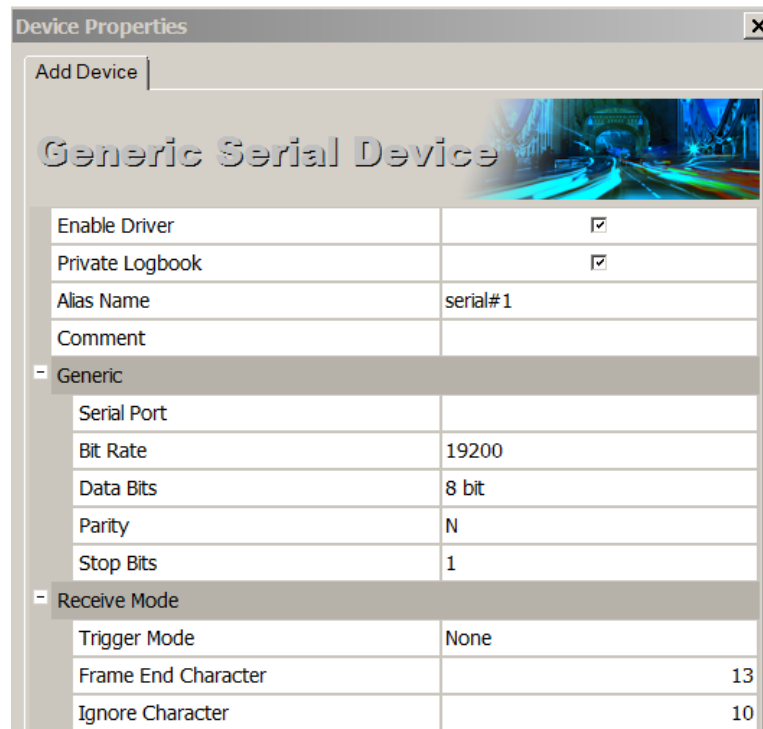


Abbildung 4.19: Hinzufügen eines RS232-Geräts

Um das System im Online-Modus über ein RS232-Touchscreen steuern zu können, müssen die in Abbildung 4.20 gezeigten „Serial Port“-Trigger im Triggermenü verwendet werden. Dort muss der verwendete Serial Port ausgewählt und mit „Value A“ und „Value B“ der zu erwartende Befehl vorgegeben werden. Wenn über den Touchscreen beispielsweise der Befehl „KL“ gesendet wird, muss für „Value A“ der Wert 75 (ASCII für „K“) und für „Value B“ der Wert 76 (ASCII für „L“) ausgewählt werden. Am Ende des Befehls muss, sofern das Touch-Bedienfeld dies nicht automatisch tut, immer ein „Carriage Return“ (ASCII Code 13) mitgesendet werden, da dieser das Ende des Befehls darstellt.

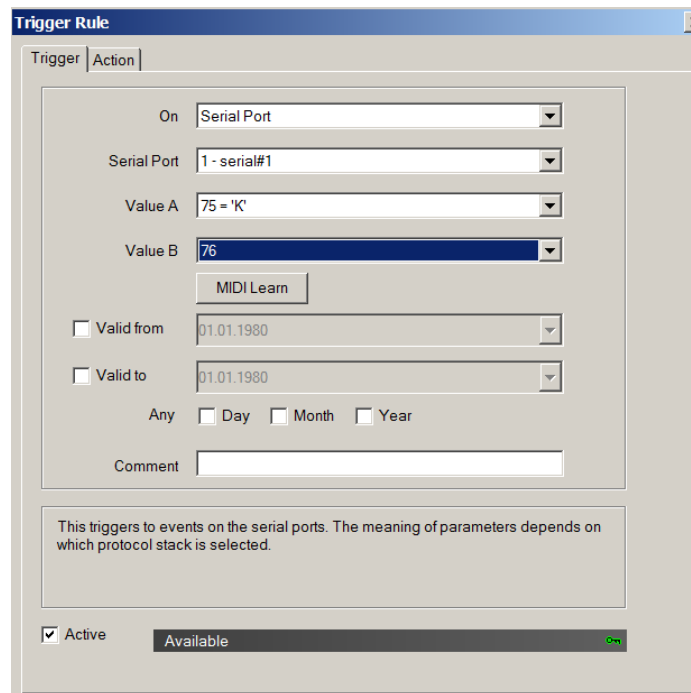


Abbildung 4.20: Erstellen eines RS232-Triggers im online-Modus

Es können außerdem auch Daten über RS-232 an das Bedienfeld gesendet werden. Diese Option gibt es bei den Cue Actions in den Cue Einstellungen oder bei Trigger Actions im Online-Modus. Dabei kann ausgewählt werden, ob Strings oder binäre Daten gesendet werden sollen. Ein Beispiel der Formatierung wird in der jeweiligen Aktion immer angezeigt.

4.3.2 e:bus-Bedienfeld

Wenn der Wunsch besteht, im Standalone-Modus alle Aktionen des Butler XT über das Bedienfeld ausführen zu können, muss das Bedienfeld den proprietären e:bus unterstützen. Dafür kommen Bedienfelder der Glass-Touch-Serie der Firma Traxon in Frage. Sie unterstützen den e:bus und somit auch den Standalone-Modus des Butler XT. Es gibt verschiedene Versionen dieses Glass-Touchscreens. Die Einrichtung funktioniert jedoch bei allen gleich. Wie bereits erwähnt, werden die Touchscreens per e:bus verpolungssicher angeschlossen. Die Einrichtung erfolgt dann im Programmer der Lighting Application Suite. Nachfolgend wird diese Einrichtung beschrieben.

Die Einbindung der e:bus-Bedienfelder funktioniert ähnlich wie die Einbindung des Butler XT über den Device Manager im Programmer. Dort muss wieder der in Abbildung 4.6 markier-

te Knopf für die automatische Suche von Geräten gewählt werden. In dem sich öffnenden Fenster muss nun der Haken für die Suche nach e:bus-Geräten gesetzt und die Suche gestartet werden. Nachdem der Butler XT und das Touch-Bedienfeld erkannt wurden, werden sie wie in Abbildung 4.21 im Device Manager angezeigt. Das Bedienfeld ist nun bereit für die Konfiguration.

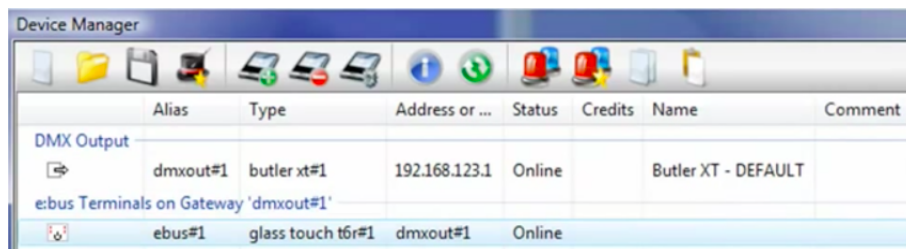


Abbildung 4.21: Butler XT mit eingebundenen e:bus Glass-Touchscreen

Die Konfiguration der Tasten des Glass-Touchscreens funktioniert ähnlich wie die Konfiguration der Taster und digitalen Eingänge des Butler XT. Mit einem Doppelklick auf den Touchscreen im Device Manager gelangt man in das in Abbildung 4.22 gezeigte Konfigurationsfenster. Dort können in den oberen Reitern der Online-Modus und der Standalone-Modus ausgewählt werden. Auch hier werden die einzelnen Tasten wieder wie bei der Konfiguration des Butler XT durch Anklicken in der Abbildung des Touchscreens für die Programmierung angewählt.

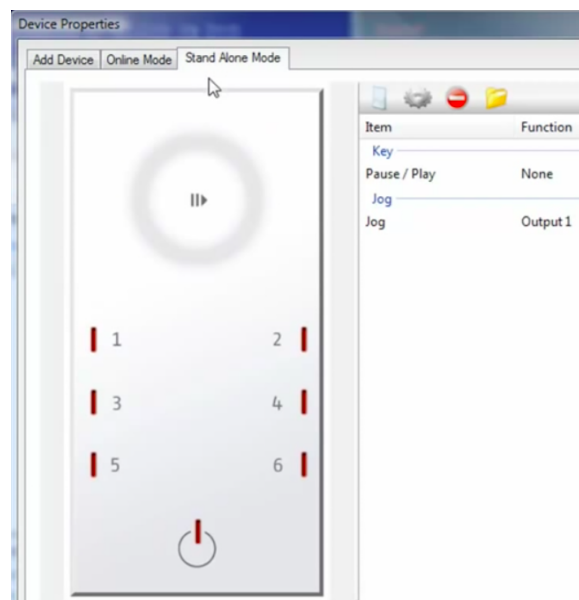


Abbildung 4.22: Konfiguration des Glass-Touchscreens

5 Integration

Das KKS-Labor (Labor für Kabine und Kabinensysteme) im HCAT (Hamburg Centre of Aviation Training) besitzt ein Airbus Single Aisle Kabinen-Modell (Siehe Abb. 5.1). Im Vorfeld dieser Arbeit wurde die Kabine bereits mit Lichtelementen ausgestattet, worauf im nachfolgenden Unterkapitel eingegangen wird. Der Butler XT wird so in das Kabinen-Modell integriert, dass allein über ihn die Beleuchtung in der Flugzeugkabine gesteuert werden kann. Alle nötigen Schritte bis dahin werden nachfolgend dokumentiert.



Abbildung 5.1: Airbus Single Aisle Kabinen-Modell

5.1 Hardwareaufbau

Zunächst muss der vorhandene Hardwareaufbau in der Kabine beschrieben werden. In die Decke der Flugzeugkabine sind bereits 24 der 8x8 RGB-LED-Matrizen eingebracht. Diese sind über die in Kapitel 4.1 beschriebenen Power/Data-Injectors mit insgesamt sechs Butler Classic verbunden. In Abbildung 5.2 ist schematisch der Hardwareaufbau dargestellt. Aus Übersichtsgründen wurden in dieser Abbildung die Power/Data-Injectors und die Netzteile ausgelassen. Der reale Hardwareaufbau besitzt pro DMX-Universum (zwei LED-Boards) einen Power/Data-Injector und jeweils ein 150W Netzteil für sechs LED-Boards. Über spezielle Y-Kabel, die Power und Data nach jedem zweiten LED-Board voneinander trennen, wird erreicht, dass die Versorgungsspannung durch alle sechs LED-Boards geleitet werden kann, ohne ein Aufeinandertreffen verschiedener DMX-Universen zu riskieren.

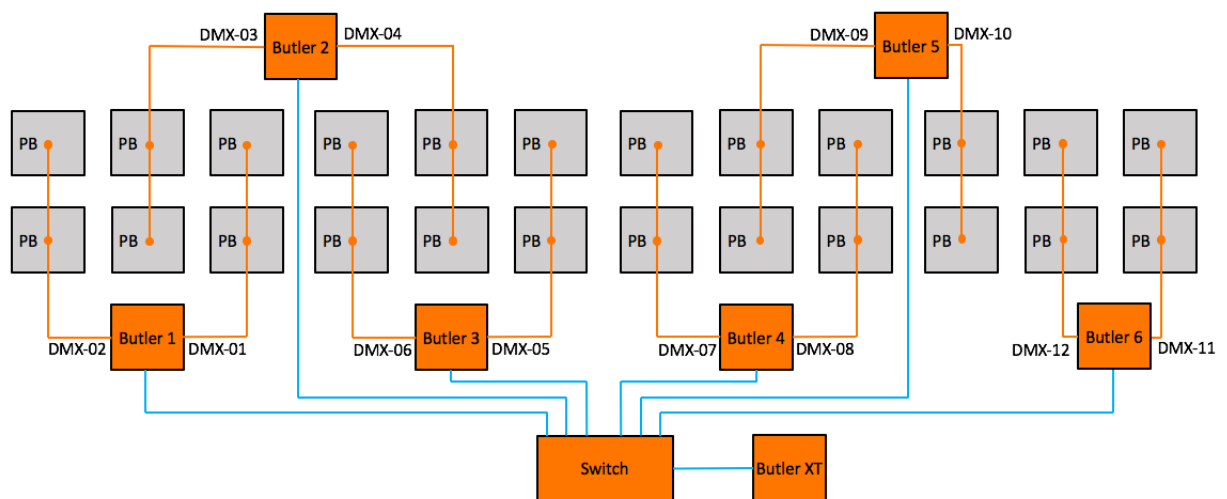


Abbildung 5.2: Schema des Hardwareaufbaus ohne Power/Data-Injectors

In Abbildung 5.2 ist außerdem zu sehen, dass der Butler XT für die Steuerung des Gesamtsystems dauerhaft mit dem e:net Switch verbunden sein muss. Er wird nun ausschließlich durch das Trennen vom Programmierer und nicht mehr durch Trennen vom Switch in den standalone-Modus gebracht.

5.2 Programmierung

Wie in Kapitel 4.2 werden auch für dieses Kapitel die Schritte zur Programmierung des Systems in der Lighting Application Suite erläutert. Zum einen werden hier neue Aspekte, wie die Erstellung eines Hardwareclusters, behandelt und zum anderen Unterschiede zu der Inbetriebnahme des Butler XT allein an einer LED-Matrix aufgezeigt. Zudem werden abschließend noch Probleme beziehungsweise Besonderheiten bei der Arbeit mit dem Hardwarecluster herausgearbeitet und die Integration damit qualitativ bewertet.

5.2.1 Erstellung eines Hardwareclusters

Die Clustererstellung dient dazu, den Butler XT und die Butler Classic in einem Master-Slave-Netzwerk zu verbinden, in welchem der Butler XT die Rolle des Masters und die Butler Classic die Rollen der Slaves übernehmen. Ein Butler XT kann in Verbindung mit Butler Classic nur die Rolle des Masters übernehmen. Als Slave kann der Butler XT nur in Verbindung mit anderen Butler XT fungieren. Außerdem ist zu vermerken, dass der Butler XT die Clusterverbindung mit den Butler Classic im Standalone-Modus nicht in voller Gänze unterstützt. Eine optimale Master-Slave-Kompatibilität unter verschiedenen Butler-Geräten wurde erst mit dem Butler XT2 und dem Butler S2 implementiert. Wenn eine vollständige Funktionskompatibilität im Standalone-Modus erwünscht ist, müssten in diesem Fall als Slaves Butler XT anstelle von Butler Classic eingesetzt werden. Weitere Informationen zu den auftretenden Problemen werden in Unterkapitel 5.2.4 ersichtlich. Dieses Unterkapitel erklärt, welchen Schritte zur Einrichtung der Cluster-Verbindung notwendig sind.

5.2.1.1 Allgemeingültige Einstellungen

Die Clustererstellung beginnt mit dem „Device Setup“ der Butler. Dieser ist, wie in Unterkapitel 4.2.1.1 bereits erklärt wird, über das Netzwerk-Fenster im Programmierer erreichbar. Der einzige Unterschied in diesem Kapitel ist jedoch, dass hier nicht nur der Butler XT, sondern auch die Butler Classic im Netzwerk konfiguriert werden müssen.

Wenn die Butler noch auf Werkseinstellungen laufen, muss einer nach dem anderen an das e:net angeschlossen werden und über den Device Setup jedes mal eine einzigartige IP-Adresse zugewiesen bekommen. Abbildung 5.3 zeigt den Device Setup des Butler XT mit seinen Einstellmöglichkeiten. Neben der einzigartigen IP-Adresse sollte jeder Butler zur Wahrung der Übersicht auch einen einzigartigen Namen erhalten. Wenn diese Einstellungen abgeschlossen sind, müssen die für den Cluster wichtigen Einstellungen getätigt werden.

Device Setup	
Butler XT	
Device Basics	
Device Name	Butler XT - DEFAULT
IP Address	192.168.123.9
Subnet Mask	255.255.255.0
Gateway Address	0.0.0.0
MAC Address	00-16-c1-f1-00-48
Base Time / Date	2100.01.04 - 00:49:17
DST Offset	1
DST Start	04.16 - 03:00
DST End	09.20 - 02:00
Enable IR	enabled
Version	
Hardware Build Version	5
Software Build Version	1.5.386
Show Settings	
Show File	default.bsf
Group ID	99
Enable Sync	<input checked="" type="checkbox"/>
Dmx Settings	
Sync Delay	2
DMX Break Distance	33
Merge Univ 2 in Univ 1	<input type="checkbox"/>

Abbildung 5.3: Device Setup des Butler XT

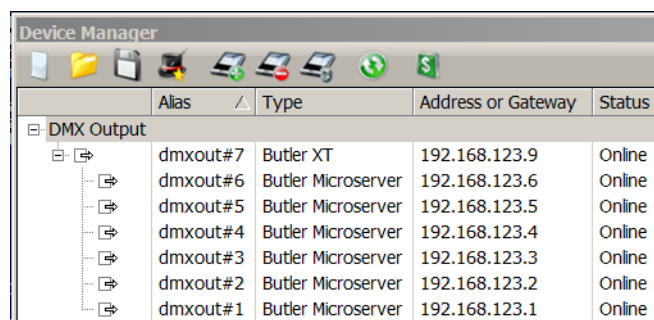
Zunächst müssen alle Butler des Clusters im Device Setup die selbe „Group ID“ zugewiesen bekommen. Des weiteren muss beim Butler XT die Einstellung „Enable Sync“ aktiviert werden. Bei den Butler Classic ist es noch wichtig, dass bei allen die Einstellung „Timecode Master“ deaktiviert ist, damit keiner der Butler Classic versucht als Master zu fungieren, da dies in Verbindung mit einem Butler XT zu Konflikten führt.

5.2.1.2 Showspezifische Einstellungen

In der jeweiligen Show müssen wieder wie in Unterkapitel 4.2.1.2 einige Einstellungen im Device Manager getätigt werden. Zunächst wird auch hier der Wizard zur automatischen Suche von Geräten im Netzwerk verwendet. Wenn alle allgemeingültigen Einstellungen richtig erledigt wurden, sollten nun alle angeschlossenen Butler im Device Manager angezeigt

werden. Als nächstes müssen alle Butler Classic, die zum Cluster gehören, wie in Abbildung 5.4 per Drag and Drop in den Butler XT gezogen werden, worauf sie dem Butler XT untergeordnet werden.

Zuletzt müssen jedem Butler noch DMX-Universen zugeordnet werden. Dazu müssen für jeden Butler im Device Manager in den jeweiligen Einstellungen die DMX-Universen den DMX-Ausgängen zugeschrieben werden (Siehe Abb. 4.7). Dem Butler XT müssen zwar standardmäßig auch DMX-Universen zugewiesen werden, diese werden jedoch in diesem Aufbau nicht verwendet, da der Butler XT hier nur als Steuerung für das Gesamtsystem dient. Abbildung 5.2 zeigt, wie die Universen in der Hardware verteilt sind, wenn der erste Butler Classic die ersten beiden Universen erhält, der zweite die darauffolgenden Universen erhält und so weiter. Natürlich können die Universen auch anders auf die Butler verteilt werden. Zum Beispiel so, dass die Universen in der Hardware aus Abbildung 5.2 von links nach rechts in der richtigen Reihenfolge aufsteigend verteilt sind. Diese Entscheidungen müssen dann lediglich im Patchfile berücksichtigt werden.



	Alias	Type	Address or Gateway	Status
DMX Output				
└─┬─	dmxout#7	Butler XT	192.168.123.9	Online
└─┬─	dmxout#6	Butler Microserver	192.168.123.6	Online
└─┬─	dmxout#5	Butler Microserver	192.168.123.5	Online
└─┬─	dmxout#4	Butler Microserver	192.168.123.4	Online
└─┬─	dmxout#3	Butler Microserver	192.168.123.3	Online
└─┬─	dmxout#2	Butler Microserver	192.168.123.2	Online
└─┬─	dmxout#1	Butler Microserver	192.168.123.1	Online

Abbildung 5.4: Butler-Cluster im Device Manager

Die Clustererstellung ist damit abgeschlossen. Die Einrichtung der Knöpfe und digitalen Eingänge des Butler XT funktioniert weiterhin wie im Unterkapitel 4.2.1.2. Für den Standalone-Modus kann bei der Auswahl des DMX-Ausgangs für die verschiedenen Aktionen einfach „Both Outputs“ gewählt werden. Diese Einstellung gilt dann für alle Butler im Cluster. Bei einigen Aktionen im Standalone-Modus gibt es bei der Nutzung von Butler Classic in Verbindung mit einem Butler XT Probleme, weshalb auf diese Aktionen verzichtet werden sollte. Näheres dazu wird in Unterkapitel 5.2.4 erläutert.

5.2.2 Patchfile

Das Patchfile wird für die Integration wie in Unterkapitel 4.2.2 beschrieben erstellt. Der einzige Unterschied ist in diesem Fall ein fixer, nicht ohne weiteres veränderbarer Hardwareaufbau, welcher zudem deutlich komplexer ist, als bei der Inbetriebnahme des Butler XT an einem LED-Board.

Zunächst müssen einige Details bei der Erstellung des Patchfiles bedacht werden. Das Patchfile muss exakt die Realität widerspiegeln. Demnach muss das jeweils erste RGB-LED-Board im jeweiligen DMX-Universum, welches direkt an den Power/Data-Injector angeschlossen ist, auch den Adressbereich von 1-192 erhalten. Das zweite LED-Board, welches an das erste LED-Board angeschlossen ist, erhält dementsprechend den Adressbereich 193-384. Die Adressbereiche ergeben sich aus der Anzahl der RGB-LEDs auf den LED-Boards (64 LEDs multipliziert mit 3 Kanälen pro RGB-LED). Es ist außerdem nicht möglich, die DMX-Adressen frei auf den LED-Boards bestimmten LEDs zuzuordnen. Die Zuordnung geschieht automatisch und für jedes Board gleich. Somit müssen auch Verdrehungen der LED-Boards im Patchfile berücksichtigt werden. Wenn der Hardwareaufbau also nicht genau einsehbar ist, gibt es keine andere Option, als den Aufbau einem Soll-Ist-Vergleich durch Ansteuern der LED-Boards durch den Programmierer über das Preview-Fenster zu unterziehen. Dazu kommt, dass jedes LED-Board einen festen Screen-ID-Bereich zugeordnet bekommt, in welchem jede LED ihre eigene Screen ID erhält. Beim Aufbau eines Systems mit mehreren LED-Boards sollte hier auf die richtige Anordnung der Boards geachtet werden, da sich einige Tools, wie zum Beispiel der Chaser-Wizard, an diesen Screen-IDs orientieren.

Das in Abbildung 5.5 dargestellte Patchfile zeigt die Installation der LED-RGB-Boards in der Decke der Flugzeugkabine. Immer zwei der LED-Boards sind in einem DMX-Universum zusammengefasst, welche, wie bereits erklärt, auf die verschiedenen Butler Classic verteilt sind (vgl. Unterkapitel 5.2.1).

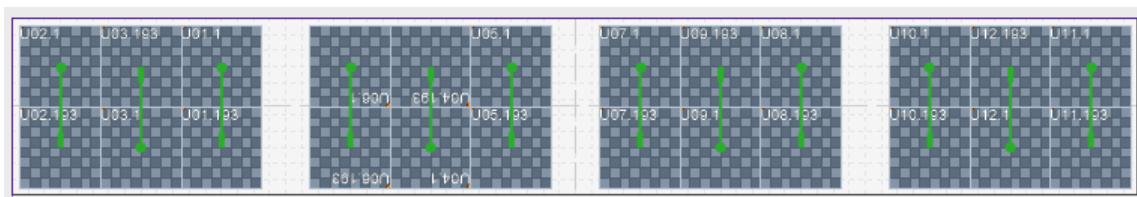


Abbildung 5.5: Patchfile der LED-Matrizen des Kabinenmodells

Wie an den Adressierungsbeschriftungen in Bild 5.5 erkennbar, ist die Adressierungsreihenfolge hier bedingt durch den realen Aufbau teilweise unterschiedlich. Außerdem sind einige

der LED-Boards um 180 Grad gedreht, was ebenfalls auf den realen Aufbau zurückzuführen ist.

5.2.3 Show-Programmierung

Dieses Unterkapitel befasst sich zum einen mit den Unterschieden der Programmierung des Butler-Clusters zur Programmierung des Butler XT aus Unterkapitel 4.2.3 und zum anderen mit einem praktischen Beispiel für die Programmierung von LiveFX-Effekten im Butler-Cluster.

5.2.3.1 Unterschiede bei der Showprogrammierung

Für den Programmierer der Lighting Application Suite macht es bei der Erstellung von Lichtshows prinzipiell keinen Unterschied, ob ein komplexes System mit mehreren DMX-Universen oder ein einfaches System mit nur einem DMX-Universum genutzt wird. Solange für das System ein fehlerfreies Patchfile existiert, ist die Programmierung in beiden Fällen außer beim Chaser-Wizard (Siehe Unterkapitel 4.2.3.3) identisch. Somit kann an dieser Stelle das Kapitel 4.2.3 für die Programmierung des Lichtsystems in der Flugzeugkabine als Leitfaden genutzt werden.

Es gibt jedoch einen minimalen Unterschied bei der Übertragung der Cuelists auf die Butler. Wie in Abbildung 5.6 zu sehen ist, werden dort in der Auflistung in diesem Fall alle Butler angezeigt, die für die aktuelle Show im Device Manager eingebunden wurden. Weiter oben im Feld „Master Target Device“ muss der Butler angegeben werden, der den Master im System darstellen soll. In der Auflistung werden außerdem die jeweiligen DMX-Universen angezeigt, die den Butlern zugeteilt wurden. Beim Exportieren der Show werden den einzelnen Butlern die Teile der Lichtshow überschrieben, die zu ihren jeweiligen DMX-Universen gehören. Somit werden die Daten an dieser Stelle für den Standalone-Modus auf die Butler aufgeteilt.

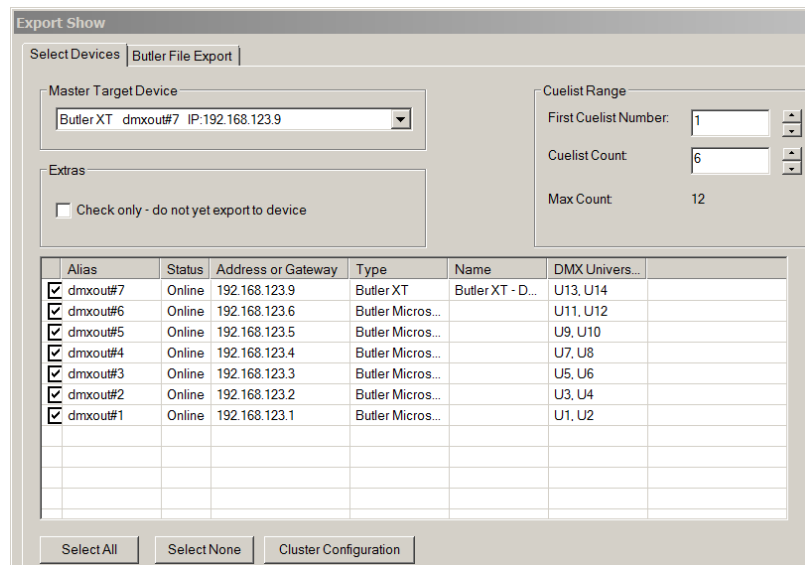


Abbildung 5.6: Exportieren der Show auf den Butler-Cluster

Wenn die CueLists auf die Butler übertragen wurden und die Taster sowie die digitalen Eingänge des Butler XT konfiguriert wurden, kann der Programmierer vom e:net getrennt werden. Das System ist dann in der Lage, die CueLists unter der Steuerung des Butler XT, eigenständig abzuspielen.

5.2.3.2 Praktisches Beispiel LiveFX

In diesem Unterkapitel wird ein praktisches Beispiel für eine Effektprogrammierung mit dem LiveFX-Tool erklärt. Dabei handelt es sich um die Simulation eines Tagesablaufs in der Flugzeugkabine mit Sonnenaufgangs-, Sonnenuntergangs-, Tages- und Nachtphase. Die ganze Simulation spielt sich in einem Zeitraum von ca. 3 Minuten ab. Dieses Szenario wurde speziell für die Nutzung im Standalone-Modus erstellt, wobei im weiteren Verlauf noch eine Besonderheit aufgezeigt werden.

Zunächst müssen die einzelnen Phasen anhand des LiveFX-Tools ausgedrückt werden. Dazu muss zuerst das Aussehen der einzelnen Phasen feststehen. In diesem Beispiel wird ein Sonnenaufgang wie folgt dargestellt: Er startet mit rotem Licht geringer Helligkeit, welches langsam an Helligkeit gewinnt und zunächst in einen orangenen, daraufhin in einen gelblichen und schließlich in einen weißen Farbton übergeht, welcher den höchsten Stand der Sonne darstellt. Der Sonnenuntergang sieht dementsprechend aus, nur dass er die Farbspektren rückwärts durchläuft. Die Nacht wird als blaues Licht dargestellt, welches

langsam an Helligkeit gewinnt und zum Morgen hin wieder dunkler wird.

Abbildung 5.7 zeigt die Darstellung des Sonnenaufgangs. Diese Darstellung wurde über Halbbrampen realisiert. Das bedeutet, dass je nach eingestellter „Ratio“ ein Teil des Effektes eine Rampe und ein anderer Teil einen konstanten Wert darstellt. Über die „Size“ wird die maximale Intensität der einzelnen Farbfader eingestellt. In diesem Fall liegen die Maximalwerte bei 100% Rot, 90% Grün und 80% Blau, was einen angenehmen, dem Tageslicht ähnlichen, Weißton darstellt. Über die „Ratio“ werden die Rampen so eingestellt, dass zunächst der rote Fader, gefolgt von dem Grünen und schließlich dem Blauen Fader an Helligkeit gewinnt. Auf diese Weise wird der gewünschte Farbverlauf erzielt. Die Tagesphase wird durch den Zeitraum dargestellt, in dem alle Halbbrampen ihren maximalwert erreicht haben und diesen halten. Dass die Phasenverschiebung in hier auf 35% eingestellt wird, hat folgenden Hintergrund: Wenn der Butler XT im Cluster mit den Butler Classic genutzt wird, werden die Startpunkte der LiveFX-Effekte im Standalone-Modus scheinbar willkürlich gesetzt. Der Effekt beginnt demnach nicht zwingend am Anfang, sondern an einem beliebigen anderen Punkt, der durch Ausprobieren ermittelt und dann über die Phasenverschiebung korrigiert werden muss. Mit den Effekten für den Sonnenuntergang und die Nacht verhält es sich ähnlich.

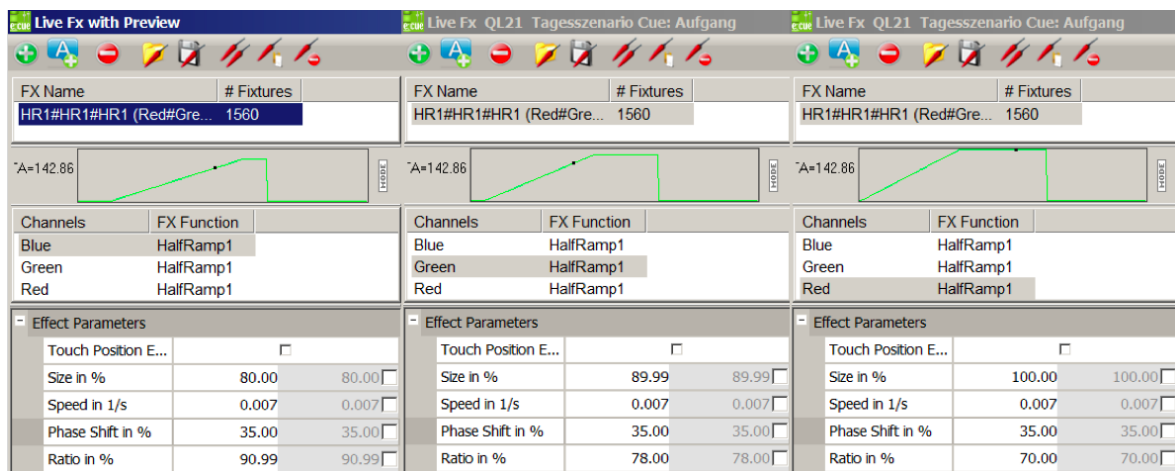


Abbildung 5.7: Sonnenaufgang im LiveFX-Tool

Der in Abbildung 5.8 dargestellte Sonnenuntergang wurde wie der Sonnenaufgang mit Halbbrampenfunktionen dargestellt. Diesmal wurden lediglich fallende anstatt steigende Rampen verwendet. dementsprechend beginnt dieses Szenario mit einer Weile Tageslicht und geht dann in die Dämmerung über, wobei als erstes der Blauanteil, dann der Grünanteil und schließlich der Rotanteil des Lichtes gedimmt wird. Auch hier muss der Startpunkt des Szenarios über die Phasenverschiebung angepasst werden.

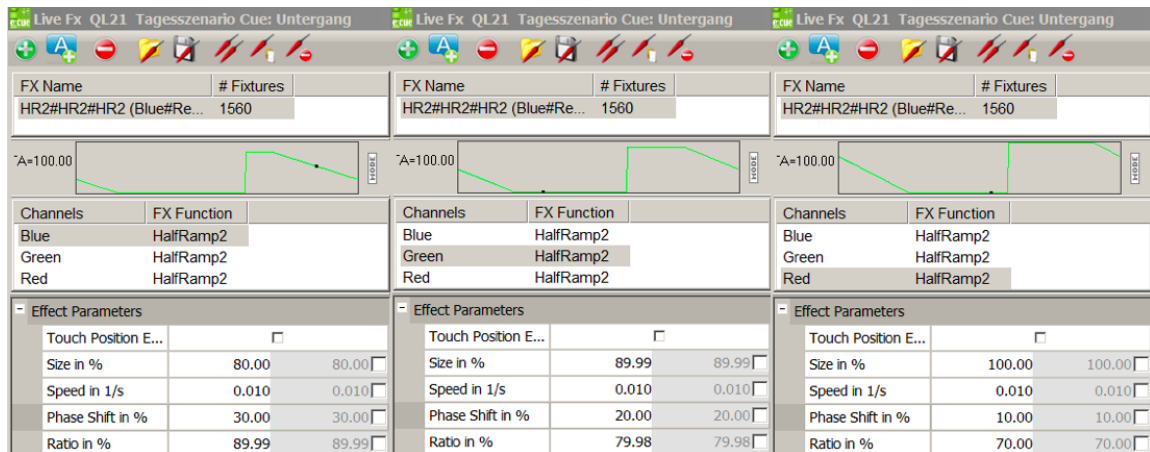


Abbildung 5.8: Sonnenuntergang im LiveFX-Tool

Der letzte Teil des Szenarios ist die Nachtphase, für die, wie in Abbildung 5.9 zu sehen ist, lediglich der blaue Fader verwendet wird. Dabei wird in Form einer Sinushalbwelle das blaue Licht bis auf einen Wert von 40% hoch- und schließlich wieder heruntergefahren. Daraufhin startet erneut das Sonnenaufgangsszenario.

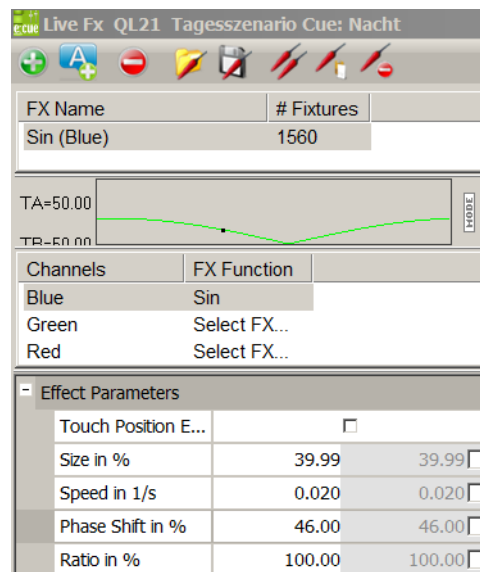
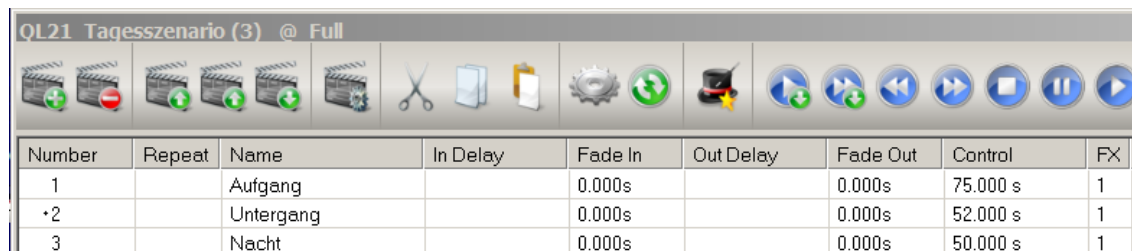


Abbildung 5.9: Nacht im LiveFX-Tool

In Abbildung 5.10 sind die drei LiveFX-Effekte in der Cuelist zu sehen. Sie werden einzeln über den „Cue Record“ aufgezeichnet und mit wait-Zeiten versehen. Die wait-Zeiten werden so gesetzt, dass sie, in Anlehnung an den im jeweiligen LiveFX-Effekt eingestellten „Speed“,

den gewünschten Effektbereich abbilden. Die in Abbildung 5.8 eingestellte Effektgeschwindigkeit von $0,01\frac{1}{5}$ bewirkt beispielsweise eine Effektdauer von 100 Sekunden. Der relevante Effektbereich, ohne jenen Bereich in welchem der Effekt den Intensitätswert 0% hat, macht etwas über die Hälfte der Effektdauer aus. Daher wird dem Sonnenuntergang eine Dauer von 52 Sekunden zugewiesen. Mit den anderen Effekten wird ähnlich verfahren.



Number	Repeat	Name	In Delay	Fade In	Out Delay	Fade Out	Control	FX
1		Aufgang		0.000s		0.000s	75.000 s	1
•2		Untergang		0.000s		0.000s	52.000 s	1
3		Nacht		0.000s		0.000s	50.000 s	1

Abbildung 5.10: Tagesszenario in der Cuelist

Wenn all diese Schritte wie angegeben durchgeführt werden, kann das Szenario im Standalone-Modus in dem Flugzeugkabinen-Mockup abgespielt werden.

5.2.4 Probleme und Hinweise

Die Arbeit mit dem Butler-Cluster in der Flugzeugkabine zeigt einige Besonderheiten, die sich als problematisch erweisen oder auf die zumindest hingewiesen werden sollte. Dabei handelt es ausschließlich um Probleme, die sich auf die Verwendung eines Clusters aus Butler XT und Butler Classic beziehen.

Zum einen sollte noch einmal erwähnt werden, dass Standalone-Trigger und Standalone-Aktionen für Taster und digitale Eingänge im Cluster nur für den Butler XT eingestellt werden können. Die resultierenden Aktionen sind jedoch für den gesamten Cluster gültig. Einige Aktionen funktionieren jedoch in einem Cluster aus Butler XT und Butler Classic nicht, da der Butler XT nur für eine Clusterbildung mit anderen Butler XT vorgesehen ist. Ein Cluster mit Butler Classic funktioniert daher nur mit Einschränkungen. Folgende Aktionen funktionieren im Cluster:

- Play Cuelist
- Pause/Resume (Pausiert im Cluster nur nach einem kompletten Durchlauf der Cuelist)
- Next
- Previous
- Restart

Von allen anderen Aktionen wie beispielsweise „Play/Stop“ oder „On/Off“ im Standalone-Modus sollte in einem Cluster aus Butler XT und Butler Classic Abstand genommen werden, da diese aufgrund der eingeschränkten Kompatibilität des Butler XT zu den Butler Classic unerwartetes Verhalten zeigen.

Der nächste Hinweis gilt der Vielfalt der Programmversionen der Lighting Application Suite. Für diese Arbeit wird die LAS Version 6.1 verwendet. Im Vorfeld wurde jedoch auch die ältere Version 5.5 getestet, wobei ein gravierender Nachteil der älteren Version aufgefallen ist. Wenn ein Cluster aus Butler XT und Butler Classic gebildet werden soll, muss die LAS Version 6.0 oder neuer verwendet werden. Ältere Versionen der LAS unterstützen solche Cluster nicht. Das ist auch der Grund dafür, wieso das Kapitel „Clustering“ erst ab Version 6.0 im LAS Software-Handbuch zu finden ist.

Weiterhin ist darauf hinzuweisen, dass im Cluster mit den Butler Classic gewisse Anomalien auftreten, die einige Funktionen bei der Show-Programmierung erschweren. Zum einen muss bei der Programmierung statischer Lichtszenarien (eine bestimmte statische Beleuchtung soll aktiv sein, bis sie manuell abgeschaltet wird) für den Standalone-Modus auf Fade-In Effekte zum langsamen Hochfahren der Beleuchtungsstärke verzichtet werden, da diese im Cluster immer zur Abschaltung des Szenarios nach dem Hochfahren der Beleuchtungsstärke führen. Des Weiteren muss darauf geachtet werden, dass die Startpunkte in LiveFX-Effekten im Standalone-Cluster meist nicht die erwarteten Startpunkte sind. So kann es sein, dass der Effekt anstatt am Anfang irgendwo in der Effektmitte beginnt. Wenn der Startpunkt für die Anwendung relevant ist muss dort mit der Phasenverschiebung der Startpunkt angepasst werden (Siehe Unterkapitel 5.2.3.2).

5.3 Bewertung der Integration

Die Integration des Butler XT in das Gesamtsystem im Online-Modus ist ohne Frage frei von Einschränkungen oder Problemen möglich, da die Steuerung in diesem Fall letztlich über den Programmier-PC läuft. Tastendrücke oder Befehle von e:bus Touchscreens werden über den Butler XT lediglich an den Programmier weitergeleitet, welcher dann die Steuerung des

Hardwareclusters übernimmt. Viel interessanter ist die geforderte Standalone-Integration des Butler XT als alleinige Steuerung des Hardwareclusters. Was außer Frage steht, ist, dass der Standalone-Modus insgesamt weniger Funktionen bietet als der Online-Modus, da an dieser Stelle die Rechenleistung des PCs fehlt. Dieser Aspekt hängt jedoch nicht mit der Integration des Butler XT in das System zusammen und fließt daher nicht in die Bewertung ein.

Ein positiver Aspekt der Integration ist vor allem die intuitive Clusterbildung per Drag and Drop im Device Manager. Ein Butler-Cluster ist somit schnell erstellt und funktioniert sofort nach der Zuweisung der DMX-Universen. Ein weiterer Vorteil ist die Funktionalität der komplexen Effekte des LiveFX-Tools über den Butler XT. Lediglich die Zeit, wie lange diese Effekte laufen sollen, muss vor dem Übertragen auf den Butler vorgegeben werden.

Nachteile bei der Integration ergeben sich hauptsächlich daraus, dass der Butler XT ursprünglich nicht dafür vorgesehen war, Hardwarecluster mit Butler Classic zu bilden. Dadurch werden, wie in Unterkapitel 5.2.4 beschrieben wird, nur bestimmte Aktionen im Standalone-Modus unterstützt. Einige Aktionen haben entweder keinen oder nicht den erwarteten Effekt auf das System. Ein weiterer Nachteil ist die fehlende räumliche Berechnung des Chaser-Wizards, wodurch dieser für die Nutzung von LED-Boards in mehreren DMX-Universen quasi unbrauchbar wird. Chaser müssen dadurch mühsam von Hand programmiert werden. Auch die in Unterkapitel 5.2.4 erwähnten Anomalien bei der Ausführung gewisser Beleuchtungsszenarien im Standalone-Modus erschweren die Arbeit mit dem Butler-Cluster.

Alles in allem war die Integration des Butler XT als alleinstehende Steuerung der Beleuchtung des Flugzeugkabinen-Mockups erfolgreich und sie eignet sich gut für sehr einfache Beleuchtungsszenarien wie beispielsweise statische Szenarien oder einzelne LiveFX-Effekte in der Flugzeugkabine. Die Voraussetzung für die erfolgreiche Nutzung des Systems im Standalone-Modus ist jedoch die Beachtung der Hinweise zu den Funktionen, welche im Cluster nicht unterstützt werden, um Verwirrungen bei der Verwendung zu vermeiden. Wenn die Anwendung jedoch ein komplexeres Maß an Funktionen verlangt, kann auf den Online-Modus nicht verzichtet werden. Der große Vorteil, den diese Integration bietet, ist der Verzicht auf den PC bei der Beleuchtungssteuerung. Der Butler XT ist im Verhältnis zu einem herkömmlichen PC sehr klein und lässt sich räumlich gut in das Kabinen-Mockup integrieren. Mit den Tastern kann beim Einschalten des Systems sofort und einfach die Kabinenbeleuchtung in Betrieb genommen werden. Die Möglichkeit der Verwendung eines Touchscreens kann die Bedienung sogar noch weiter erleichtern.

6 Zusammenfassung

Neben einer theoretischen Einführung in die Welt der seriellen Bussysteme, wobei speziell auf den DMX512-Standard sowie auf den proprietären e:bus eingegangen wird, werden in dieser Arbeit alle wichtigen Informationen zu dem verwendeten DMX-Controller „Butler XT“ aufgezeigt. Das Ergebnis dieser Arbeit ist in erster Linie ein auf das Wesentliche konzentrierter Einblick in ein sehr komplexes Programm für die Steuerung komplexer Beleuchtungslösungen. Somit ermöglicht diese Arbeit eine einfache und schnelle Inbetriebnahme des vorgestellten DMX-Controllers und zeigt auf, welche Möglichkeiten und welche Einschränkungen dieses System im Standalone-Modus ohne die Nutzung eines zusätzlichen PCs bietet. Es wird vor allem deutlich, dass der DMX-Controller für einfache Beleuchtungsszenarien nicht auf die Steuerung durch den PC angewiesen ist.

Außerdem werden praktisch umsetzbare Konzepte vorgestellt, die Steuerung des Beleuchtungssystems über Touch-Bedienfelder mit zwei verschiedenen Bussystemen einzurichten. Damit wird eine eventuelle zukünftige Nachrüstung des Systems mit einem solchen Bedienfeld kein Problem darstellen.

Darüber hinaus wird die Integration des Butler XT in das bestehende Beleuchtungssystem des Flugzeugkabinen-Mockups von der Hardwareeinrichtung, über die Erstellung eines Patchfiles, welches den genauen Hardwareaufbau widerspiegelt, bis hin zur Show-Programmierung erklärt. Diese Integration wurde im Rahmen dieser Arbeit darüber hinaus erfolgreich praktisch umgesetzt und alle Probleme und wichtigen Hinweise dokumentiert. Dazu wurde eine Lichtshow erstellt, welche zahlreiche Beleuchtungsszenarien enthält, welche ebenfalls als Teil der praktischen Umsetzung zählen.

Die Arbeit an diesem Thema verbindet das Erarbeiten theoretischer Fachkenntnis und umfangreichen praktischen Inhalten in einer Projektarbeit, welche alle Aspekte von der Planung über die Recherche, die strukturierte Umsetzung bis hin zur Dokumentation enthält. Sie lehrt, Zusammenhänge zwischen der Theorie und der Praxis sinnvoll zu knüpfen und zu verstehen, um diese zielführend einsetzen zu können.

Literaturverzeichnis

- [Belden-Electronics-GmbH 2011] BELDEN-ELECTRONICS-GMBH: *Power over Ethernet IEEE 802.3af*. 2011. – URL <http://belden.picturepark.com/Website/Download.aspx?DownloadToken=a33781c0-fabc-4a1e-a393-3ddd24e83335&Purpose=AssetManager&mime-type=application/pdf>. – Abruf: 2018-01-23
- [DIN 2000] DIN: *DIN 56930 Buehnenlichtstellsysteme Teil 2 - Steuersignale*. März 2000. – URL <https://www.din.de/en/wdc-beuth:din21:10645679>
- [Elation 2008] ELATION: *DMX 101: A DMX 512 HANDBOOK*. Mai 2008. – URL <https://www.manualslib.com/products/Elation-Dmx-101-2823015.html>. – Abruf: 2017-11-16
- [e:cue control GmbH 2011a] GMBH e:cue control: *Butler XT Datenblatt*, Mai 2011. – URL http://www.ecue.de/fileadmin/downloads/new2010/Product_Documentation/BUTLER_XT/DE_Butler_XT_Spec_v1p5.pdf. – Abruf: 2017-11-30
- [e:cue control GmbH 2011b] GMBH e:cue control: *Butler XT Setup Manual*, Mai 2011. – URL http://www.ecue.de/fileadmin/downloads/new2010/Product_Documentation/BUTLER_XT/DE_Butler_XT_Setup_v1p6.pdf. – Abruf: 2017-11-30
- [e:cue control GmbH 2015] GMBH e:cue control: *Lighting Application Suite System Manual*, März 2015. – URL http://www.ecue.de/fileadmin/downloads/new2010/Product_Documentation/LIGHTING_APPLICATION_SUITE/EN_LAS_7p0_System_v1p1_web.pdf. – Abruf: 2018-01-17
- [plasastandards 2011a] PLASASTANDARDS: *ANSI E1.20 - 2010 Entertainment Technology RDM Remote Device Management Over DMX512 Networks*. Januar 2011. – URL https://getdlight.com/media/kunena/attachments/42/ANSI_E1-20_2010.pdf. – Abruf: 2018-01-17

- [plasastandards 2011b] PLASASTANDARDS: *ANSI E1.27-1 - 2006 (R2011)Entertainment Technology Standard for Portable Control Cables for Use with ANSI E1.11 (DMX512-A) and USITT DMX512/1990 Products*. Juni 2011. – URL
http://modernperiodresources.weebly.com/uploads/4/3/5/6/43560237/ansi_e1-27-1_2006r2011.pdf. – Abruf: 2017-12-11
- [Samson-AG 2000] SAMSON-AG: *Serielle Dateneübertragung*. März 2000. – URL
<https://www.samson.de/document/1153de.pdf>. – Abruf: 2017-11-15
- [Schnell und Wiedemann 2006] SCHNELL, Gerhard ; WIEDEMANN, Bernhard:
Bussysteme in der Automatisierungs- und Prozesstechnik. Bd. 6. Friedr. Vieweg & Sohn Verlag, 2006. – URL <https://link.springer.com/content/pdf/10.1007%2F978-3-8348-9108-2.pdf>. – ISBN 3-8348-0045-7
- [Traxon-Technologies-Europe 2011] TRAXON-TECHNOLOGIES-EUROPE: *The e:bus system*, Juni 2011. – URL
http://www.ecue.de/fileadmin/downloads/new2010/Product_Documentation/EBUS_INFORMATION/EN_ebus_Manual_v2p2.pdf. – Abruf: 2017-12-04

Versicherung über die Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Hamburg, 29. Januar 2018

Ort, Datum

Unterschrift