



BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

B A C H E L O R A R B E I T

Konzeption und Realisierung
eines universellen
DMX-Protokollwandlers

vorgelegt von
Andy Streubel

Veranstaltungstechnik und -management
Matrikelnummer: [REDACTED]

Drontheimer Straße 9
13359 Berlin
andy.streubel@beuth-hochschule.de
Tel.: [REDACTED]

Betreuer:
Prof. Boris Balin

Abgabedatum: 11. Juli 2013



Die Inhalte dieser Arbeit stehen, soweit nicht anders vermerkt, unter der Creative Commons Lizenz CC-BY-SA 3.0

Zusammenfassung

Die Steuerung von professioneller Lichttechnik muss immer größeren und universellen Anforderungen und Ansprüchen gerecht werden. DMX512-A kann diese Anforderungen schon seit einiger Zeit nicht mehr allein erfüllen. Mit zunehmender Komplexität der Systeme wurde es notwendig, dieses seit Jahren bewährte Protokoll, durch andere Netzwerktechnologien zu erweitern. Vertreter der Hersteller aus der Beleuchtungsindustrie, Branchenverbände und Norm-Organisationen konnten sich jedoch lange Zeit nicht auf einheitliche weiterführende Standards in diesem Bereich einigen. Dadurch entstand eine Vielzahl verschiedener Protokolle namhafter Hersteller von Beleuchtungstechnik, welche jedoch zueinander inkompatibel sind. Die Behäbigkeit der für eine Normenfindung verantwortlichen Institutionen führte auch dazu, dass schon längst erforschte Neuerungen, bspw. RDM, lange Zeit benötigen, um Relevanz im Alltag der professionellen Lichtsteuerung zu entwickeln. Des Weiteren stellt die gleichzeitige Entwicklung von zahlreichen proprietären Protokollen einen Rückschritt gegenüber einem einheitlichen und somit herstellerübergreifenden Standard dar. Dies erzeugt auch eine sehr große Diskrepanz zwischen der aktuellen Entwicklungsstufe und der tatsächlich im Einsatz befindlichen Technik. Um der somit entstanden Vielfalt und Heterogenität der Systeme gerecht werden zu können, ist es notwendig, geeignete Hard- und Software zu entwickeln die es Systemintegratoren, Betreibern von Veranstaltungsstätten, Hardwareherstellern und Endanwendern ermöglicht, die Vielzahl an Protokollen zu wandeln um eine Integration in bestehende und zukünftige Systeme zu ermöglichen. Diese Bachelorarbeit befasst sich mit der Analyse aktuell am Markt erhältlicher Protokollwandlungssysteme sowie der Konzeption und Konstruktion eines universellen DMX-Protokollwandlers auf Basis der Open Lighting Architecture und der Open-Source Hardwareplattform Raspberry Pi. Sie zeigt damit auch einen möglichen Lösungsansatz, wie aktuelle und zukünftige Entwicklungen im Bereich der Lichttechnik durch den Einsatz von Open-Source Hard- und Software schnellen Einzug in den Alltag der Beleuchtungsindustrie erhalten können.

Inhaltsverzeichnis

Abstract	I
Abbildungsverzeichnis	IV
Tabellenverzeichnis	IV
Acronym	V
1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung	2
2 Das DMX-Protokoll und seine Entwicklung	3
2.1 Das Übertragungsverfahren	3
2.2 DMX512-Datenprotokoll	4
2.3 Grenzen des DMX512-Standards	6
2.4 RDM - Remote Device Management	9
2.4.1 Funktionsweise RDM	9
3 Der Ethernet-Standard in der modernen Lichtsteuerung	13
3.1 Die Ethernet-Technologie	13
3.1.1 Netzwerktopologien	13
3.1.2 Ethernet-Datenkomponenten	15
3.1.3 Unicast, Multicast und Broadcast	16
3.1.4 Das ISO/OSI-Referenzmodell	17
3.1.5 Bestandteile eines Ethernet-Netzwerks	20
3.2 Ethernetbasierte Lichtsteuerprotokolle	23
3.2.1 ArtNet	23
3.2.2 Architecture for Control Networks - ACN und sACN	25
3.2.3 Weitere Steuerprotokolle	27
4 Marktübersicht Ethernetfähige Lichttechnik	28
4.1 Lichtsteuerpulse	28
4.2 Scheinwerfertechnik	29
4.3 Ethernet-Nodes	29
5 Aufbau des Ethernet-DMX-Wandlers	33
5.1 Hardware- und Funktionsübersicht	33
5.2 Embedded Hardware Plattform	35
5.3 Anschluss- und Erweiterungsboard	36
5.4 DMX Out- und Input	38
5.4.1 Anforderungen an das USB-DMX-Interface	39
5.4.2 Prinzipieller Aufbau eines USB-DMX-Interfaces am Beispiel des DE-Interfaces	40
5.5 Stromversorgung	42
5.6 Gehäuse	43
5.7 Kosten	45
6 Übersicht der verwendeten Software	48
6.1 Betriebssystem und Softwareplattform	48
6.2 Softwareverwaltung- und Veröffentlichung	49
6.3 Peripherietreiber und Softwareerweiterungen	49

7 OLA - Open Lighting Architecture	50
7.1 Aufbau der Open Lighting Architecture	51
7.2 Funktionsübersicht der Open Lighting Architecture	51
8 Funktionstests des DMX-Protokollwandlers	55
8.1 Test der maximalen Netzwerkgeschwindigkeit	55
8.2 Prozessorlast in Abhängigkeit der übertragenen Signale	59
8.3 Produkttests für einen kommerziellen Vertrieb	61
9 Fazit und Ausblick	62
9.1 Ergebnis der Arbeit	62
9.2 Erkenntnisse und zukünftige Verbesserungen	63
9.3 Ausblick	63
Literatur	73

Abbildungsverzeichnis

1	DMX-Protokoll zeitlicher Ablauf	4
2	vereinfachtes Schema des Aufbaus eines DMX-Ethernet-Netzwerks	8
3	Schemata Netzwerktopologien	13
4	vereinfachter Aufbau eines Ethernet-Frames	16
5	schematische Darstellung Unicast, Broadcast und Multicast	17
6	OSI-Schichtenmodell	18
7	Aufbau eines ArtDMX-Pakets	24
8	Übersichtstabelle Ethernet-DMX Nodes Teil 1	30
9	Übersichtstabelle Ethernet-DMX Nodes Teil 2	31
10	Frontansicht DMX-Protokollwandler - Nahaufnahme Display	33
11	Schema Funktionsübersicht	34
12	Schema Aufbau Raspberry Pi Model B	35
13	Boardlayout Anschluss- und Erweiterungsplatine	38
14	Schema Signalpfad USB-DMX-Interface	40
15	Auflistung des Stromverbrauchs der verwendeten Komponenten	42
16	Anschlüsse und Bedienelemente der Stromversorgung	43
17	Renderings 3D-Gehäusekonzept	43
18	Explosionsansicht Gehäusebauteile	44
19	Fotografie des fertiggestellten Gehäuseprototypen	45
20	Auflistung der Kosten aller verwendeten Bauteile	46
21	Screenshot des Webinterfaces der Open Lighting Architecture	53
22	schematische Darstellung zweier priorisierter Merge-Vorgänge	54
23	schematischer Aufbau einer Messung mit iperf	55
24	Messergebnise iperf - maximaler Datenfluss Ethernet	56
25	Messergebnise iperf - maximaler Datenfluss WLAN	57
26	Messergebnise iperf - Jitter und Datenverlust	58
27	CPU-Last "olad" in Abhängigkeit der gewandelten Signale	60
28	detailliertes OSI-Schichtenmodell	64
29	Schaltplan Anschluss- und Erweiterungsplatine	65
30	Fotografie Gehäuseprototyp - Gesamtansicht vorn	66
31	Fotografie Gehäuseprototyp - Gesamtansicht hinten	67
32	Fotografie Gehäuseprototyp - Detailansicht DMX-Ports	68
33	Verdrahtungsplan der Erweiterungsmodule	69
34	Schaltplan Digital Enlightenment USB-DMX-Interface	70
35	technische Zeichnung Gehäuseentwurf Seite 1	71
36	technische Zeichnung Gehäuseentwurf Seite 2	72

Anm.: Soweit nicht explizit anders vermerkt, wurden alle Abbildungen selbst hergestellt.

Tabellenverzeichnis

1	Pinbelegung XLR 5-Pol	4
2	Timingparameter - DMX512	5
3	Aufbau eines RDM-Pakets	10
4	Übersicht proprietärer Lichtsteuerprotokolle	27
5	Übersicht der verfügbaren Anschlüsse und deren Funktion	34
6	Belegung GPIO-Header (nach BCM-Schema)	37

Acronym

DMX	Digital Multiplex
RDM	Remote Device Management
USITT	United States Institute for Theatre Technology
ANSI	American National Standards Institute
TCP	Transmission Control Protocol
IP	Internet Protocol
UDP	Universal Datagram Protocol
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
IEEE	Institute of Electrical and Electronics Engineers
MAC	Media Access Control
OSI	Open Systems Interconnection
NIC	Network Interface Card
PoE	Power over Ethernet
WOL	Wake on Lan
GPIO	General purpose Input/Output
OLA	Open Lighting Architecture
UART	Universal Asynchronous Receiver Transmitter
IC	Integrated Circuit
ACN	Architecture for Control Networks
OSC	Open Sound Control

1 Einleitung

1.1 Motivation

Seit über zwanzig Jahren ist DMX512 der Standard für das Steuern professioneller Lichttechnik im Veranstaltungsbereich. Die Technikanforderungen der Veranstaltungsindustrie haben sich im Lauf der Jahre rasant entwickelt und DMX512 kann diesen Anforderungen nicht mehr allein gerecht werden. Jüngst zeichnete sich die Nachfolgegeneration der Ethernet-basierten Lichtsteuerstandards am Markt ab. Aufgrund des vergleichsweise kleinen Nischenmarktes, teilt sich die Branche größtenteils unter wenigen großen Herstellern auf. Diese müssen einen Großteil der Entwicklungskosten in Standards, die den aktuellen Anforderungen gerecht werden, tragen. Da jeder Hersteller einen möglichst großen Teil des Marktes für sich beanspruchen will, ist ein gemeinschaftlicher und herstellerübergreifender Ansatz nur limitiert erkennbar. Für den Endkunden resultiert dies in langen Zeiträumen der Normenfindung und einer Vielzahl von herstellerspezifischen Protokollen. Bereits 2004 schrieb Dan Antonuk, Mitarbeiter der ETC-Entwicklungsabteilung und Projektmanager der ACN-Arbeitsgruppe, dazu in einem Artikel der Fachzeitschrift PROTOCOL:

Our industry has DMX512 and about twenty proprietary Ethernet protocols.
Some of these protocols don't work well, none of them work together. [Ant04]

Dieser proprietäre Ansatz schafft Alleinstellungsmerkmale für die Systeme der Hersteller und dient so zur Abgrenzung zu anderen Marktteilnehmern. Er resultiert jedoch auch in mehr Komplexität und weniger Kompatibilität.

Um der großen Komplexität der Systeme Herr werden zu können, ist es jedoch notwendig, möglichst offene und gut dokumentierte Techniken bereitzustellen. Da eine herstellerübergreifende Kompatibilität im Bereich der Lichttechnik unabdingbar ist, wurde vor wenigen Jahren durch die PLASA mit sACN ein erfolgsversprechendes Protokoll definiert. Es bleibt jedoch abzuwarten, welcher Standard sich letztendlich am Markt durchsetzen wird.

Eine weitere zu beobachtende Entwicklung ist, dass die Bereiche Lichttechnik, Videotechnik und Netzwerktechnik zunehmend verschmelzen. Dadurch steigen auch die Kenntnisanforderungen an die Betreiber der Hardware. Um zu vermeiden, dass in naher Zukunft das Wissen über die Funktionsweise der Geräte nur noch bei den Herstellern und ihren Vertriebspartnern liegt, ist im Bezug auf Wartung und Wissensvermittlung ein offenes System die nachhaltigere Lösung für Betreiber und Endanwender.

1.2 Zielsetzung

Diese Arbeit befasst sich mit der Konzeption und Realisierung eines offenen und erweiterbaren Systems zur Protokollwandlung und DMX512-Ausgabe. Die verwendete Hardwareplattform ist ein Raspberry Pi Model B. Ziel ist dabei die Fertigstellung eines betriebsbereiten Prototypen zum Wandeln zahlreicher Lichtsteuer-Protokolle über Ethernet, Wireless-LAN und die konventionelle DMX512-Schnittstelle. Der Prototyp soll in seiner Konstruktion den Anforderungen der Veranstaltungstechnik angepasst sein, also standardisierte und bewährte Steckverbinderysteme und eine solide Gehäusekonstruktion aufweisen. Des Weiteren soll dieser modular aufgebaut sein, um Erweiterbarkeit und einfache Wartung zu gewährleisten. Dies wird durch die Herstellung einer Add-On-Platine bewerkstelligt, welche den Raspberry Pi um analoge Eingänge, PWM-Ausgänge und Steckverbinder für weitere Peripherie ergänzt.

Die Funktionen der Software sollen auf der für den Raspberry Pi portierten Version der Open Lighting Architecture aufbauen. Die angestrebten Funktionserweiterungen sind die Anzeige von Statusinformationen auf dem LCD-Display und das Integrieren der Ausgabe von PWM-Signalen durch die Programmiersprache Python.

Weiterhin ist die Vermittlung aller notwendiger Grundlagen, die zum Verstehen und Betreiben von zukünftigen Lichtsteuernetzwerken benötigt werden, sowie eine Übersicht, der aktuell am Markt befindlichen Ethernet-fähigen Lichttechnik, Inhalt dieser Arbeit.

2 Das DMX-Protokoll und seine Entwicklung

DMX ist ein Acronym für die Worte “Digital Multiplex” und beschreibt ein digitales Übertragungsverfahren zwischen Lichtsteueranlagen und Beleuchtungsequipment im professionellen Umfeld der Veranstaltungstechnik.

Die Erstdefinition dieses Standards erfolgte 1986 durch das United States Institute for Theatre Technology (USITT). Durch eine marginale Revision im Jahr 1990 wurde DMX512/1990 weitergehend spezifiziert und definierte das Übertragungsprotokoll, den Übertragungsweg, die Topologie und Steckverbinderarten¹. Das auf dieser Norm basierende deutsche Pendant ist die DIN 56930 “Bühnentechnik - Bühnenlichtstellsysteme”, auf welche ich mich in der weiteren Protokollbeschreibung beziehe. Die aktuelle Evolutionsstufe findet sich im, von der Plasa herausgegebenen, ANSI-Standard E1.11-2008, USITT DMX512-A. Die grundlegenden Übertragungsparameter wurden, abgesehen von einigen marginalen Anpassungen des Signaltimings, nicht geändert, sondern nur um Zusatzfunktionen erweitert.

2.1 Das Übertragungsverfahren

DMX ermöglicht das Übertragen von bis zu 512 Steuerkanälen bei je 8 Bit Auflösung (256 Stufen)². Jedem Gerät innerhalb des DMX-Bus³ wird eine Adresse im Adressbereich 1-512 zugewiesen. Da das DMX-Protokoll auf dem Übertragungsstandard EIA-RS-485 basiert, ist es möglich bis zu 32 Empfänger bei einer maximalen Länge von ca. 1200 m zu verbinden. Empfohlen wird eine linienartige Bustopologie bei der das Signal an den einzelnen Geräten durchgeschleift wird. Für verzweigte Topologien und bei Kabelwegen über 1200 m sollten Leistungsverstärker verwendet werden.⁴

Zwingend vorgeschrieben sind fünfpolige XLR-Steckverbinder. Da für die meisten Geräte eine dreipolige Verbindung ausreichend ist, die zwei optionalen Pins definieren einen zusätzlichen und meist nicht verwendeten zweiten Datenkanal, werden in der Praxis jedoch aus Kostengründen sehr häufig auch dreipolige XLR-Steckverbinder, wie von Mikrofonkabeln üblich, verwendet. Dies ist jedoch nicht ratsam, da es so zu Verwechslungen mit Audiokabeln kommen kann, welche einen vom DMX-Standard abweichenden Wellenwiderstand besitzen können. Verwendete DMX-Kabel sollten den Standards EIA-RS-422/EIA-RS-485 entsprechen und einen Wellenwiderstand von ca. 110Ω aufweisen. Kabetypen nach CAT-5 sind auch geeignet⁵.

Die Sendegeräte verfügen über Buchsensteckverbindungen und die Empfangsgeräte über Stift-Steckverbindungen.

¹Vgl. [SOU99]

²Vgl. [DIN00] S. 2

³Erklärung “Bus“ siehe Kap. 3.1.1

⁴Vgl. [DIN00] S. 3

⁵Vgl. [DIN00] S. 6

Pin	Funktion
1	Signal GND (Schirm)
2	DMX 1- / negatives Potenzial
3	DMX 1+ / positives Potenzial
4	DMX 2- / negatives Potenzial
5	DMX 2+ / positives Potenzial

Tabelle 1: Pinbelegung XLR 5-Pol
Quelle: [DIN00]

Eine galvanische Trennung der Geräte ist nicht vorgeschrieben, Vorkehrungen für den Schutz vor zu hohen Eingangsspegeln, EIA-RS-485 spezifiziert Gleichtaktspannungen von -7V bis +12V, sind jedoch bei komplexeren Anlagen empfehlenswert⁶.

Als Leitungsabschluss ist ein $120\Omega/0,25W$ Widerstand vorgeschrieben. Dieser wird zwischen den Signalleitungen vor dem Eingang des letzten Empfängers eingesetzt. Der Abschlusswiderstand am Busende beseitigt auftretende Reflektionen, welche aufgrund entgegengesetzter Phasenlage zum Nutzsignal Auslöschungen verursachen.

2.2 DMX512-Datenprotokoll

Abweichend vom EIA-RS-485-Standard implementiert DMX512 ein eigenes Datenprotokoll.

Dieses ist ein asynchrones, d.h. nicht taktgebundenes, serielles Signal. Die Übertragung erfolgt sequentiell, die bis zu 512 Steuerwerte werden also nacheinander übertragen⁷.

Es wird eine Datenrate von 250kBit/s verwendet. Die einzelnen Bestandteile des Protokolls sind in Tabelle 2 und deren Timing in Abbildung 1 ersichtlich.

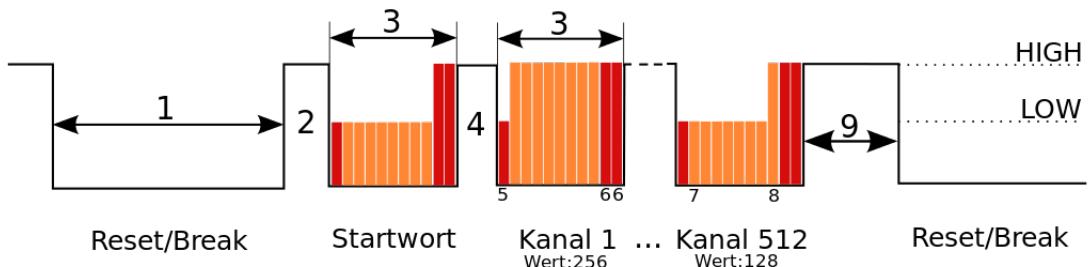


Abbildung 1: DMX-Protokoll zeitlicher Ablauf

Am Anfang jedes zu übertragenden DMX-Rahmens wird ein Break-Signal (LOW-Signal) mit einer Länge von $88\mu s$ gesendet.

Darauf folgt für mind. $8\mu s$ bis max 1s ein HIGH-Pegel, der sog. MAB (Mark after Break), welcher das Ende des Breaks signalisiert.

⁶Vgl. [DIN00] S. 3

⁷Vgl. [DIN00] S. 3

Nachfolgend wird das erste Datenbyte, das Startwort, gesendet. Die Datenbytes beginnen jeweils mit einem Startbit (LOW-Pegel), gefolgt von einem Byte (8 Bit) DMX-Daten und 2 Stopbits (HIGH-Pegel). Die Zeitdauer für ein Bit beträgt $4\mu s$ bei einer maximalen Toleranz von $+/- 0,08\mu s$ ⁸. Dabei wird zuerst das minderwertigste Bit⁹ und zuletzt das hochrangigste Bit¹⁰ übertragen.

Für das Versenden von DMX-Werten werden alle Datenbits des Startworts mit einem LOW-Pegel, also einer Null, übertragen. Dieses Nullsignal ist das Standard-Startwort bei der DMX-Übertragung. Ergänzend dazu sind nach dem Plasa-Standard ANSI E1.20-2010 Remote Device Management Over DMX512 Networks¹¹ weitere Startworte für die DMX-Erweiterung Remote Device Management (RDM) definiert.

Zwischen jedem Datenwort kann sich zusätzlich ein weiterer Mark befinden. Dies wird in der Praxis jedoch meist nicht angewandt, da Beginn und Ende des Datenwortes bereits ausreichend durch die Start- und Stopbits definiert sind und dies eine Verringerung der zeitlichen Auflösung zur Folge hätte.

Nr.	Signalname	Min	Typisch	Max
1	Reset/Break	$88\mu s$	$88\mu s$	-
2	Mark zwischen Reset und Startwort	$8\mu s$	-	1
3	Startwort-Zeit	$43,12\mu s$	$44\mu s$	$44,88\mu s$ ^a
4	Mark zwischen Datenworten	0s	0s	1s
5	Startbit (LOW)	$3,92\mu s$	$4\mu s$	$4,08\mu s$
6	Stopbit	$3,92\mu s$	$4\mu s$	$4,08\mu s$
7	Least Significant Bit (LSB)	$3,92\mu s$	$4\mu s$	$4,08\mu s$
8	Most Significant Bit (MSB)	$3,92\mu s$	$4\mu s$	$4,08\mu s$
9	Mark zwischen Datenpaketen	0s	0s	1s

Tabelle 2: Timingparameter - DMX512

Vgl. [DIN00] S. 5

^aIn der DIN 56930-2 Bühnenlichtstellsysteme sind hier fälschlicherweise $44,48\mu s$ angegeben

Auf das Startwort folgt eine sequentielle Übertragung von bis zu 512 DMX-Kanälen. Dies bedeutet, dass alle Kanäle nacheinander in aufsteigender Reihenfolge übertragen werden. Will man einen Wert auf Kanal 512 übertragen ist es somit notwendig das komplette DMX-Universum zu versenden, selbst wenn alle anderen Kanäle einen Nullwert beinhalten.

Ein Vorteil bei dieser Art der Übertragung ist, dass keine Übertragungskapazität für das Adressieren von Kanälen verschwendet wird, weil die Adressierung sich aus der Reihenfolge der Signale ableiten lässt. Sendet man dagegen beispielsweise nur Werte auf Kanal 1-12, kann das DMX-Paket bereits nach 12 Kanälen beendet werden.

⁸Vgl. [DIN00] S. 4

⁹auch LSB, Least Significant Bit

¹⁰auch MSB, Most Significant Bit

¹¹zum freien Download erhältlich unter http://tsp.plasa.org/tsp/documents/published_docs.php
Stand: 30.05.2013

Dies ermöglicht, zumindest theoretisch, eine höhere zeitliche Auflösung gegenüber der vollständigen Übertragung von 512 Kanälen. In der Praxis ist dies jedoch abhängig von der Implementierung des DMX-Standards in den einzelnen Geräten. Zu diesem Punkt finden sich auch widersprüchliche Angaben in den technischen Standards. In der DIN 56930-2 wird keine Mindestzykluszeit, also keine Mindestlänge eines DMX-Datenpaketes spezifiziert. Die Grundlage für diese DIN-Norm bildet das Dokument der USITT¹²: DMX512/1990. Dieses definiert jedoch eine maximale Wiederholrate mit folgendem Wortlaut:

The Period between the falling edge at the start of any break shall be not less than $1196\mu s$ from the falling edge at the start of the next break. (USITT 1990: 3)

Frei übersetzt bedeutet dies, dass der Abstand zwischen dem Beginn von zwei Break-Signalen mindestens $1196\mu s$ betragen soll.

Nach dem Versenden des letzten Kanals, kann ein weiterer Mark übertragen werden. Dieser stellt eine Pause im DMX-Datenstrom dar. Jene Sendeunterbrechung muss für einen Zeitraum von bis zu einer Sekunde ein HIGH-Signal ausgeben. Jedes Gerät innerhalb des DMX-Bus muss für diesen Zeitraum seinen aktuellen Zustand halten können und bei fehlender Unterbrechung das nachfolgende Datenpaket erkennen können.

Aus den eben erklärten Timingparametern lässt sich das Zeitverhalten ableiten. Da ein Bit eine Länge von $4\mu s$ besitzt und ein Datenwort aus insgesamt 11 Bits besteht, ergibt sich eine Rahmenzeit von $44\mu s$. Die Rahmenzeit multipliziert mit der maximalen Kanalanzahl und aufsummiert mit den zusätzlichen Signalanteilen¹³ ergibt eine minimale Zyklusdauer von 22,67ms (siehe Formel 1).

Aus der Zyklusdauer ergibt sich eine maximale Wiederholrate von ca. 44Hz für das Senden eines kompletten DMX-Universums.

$$\begin{aligned}
 T_{min} &= \text{Rahmenzeit} * 512 \text{ Kanäle} + (\text{Breaksignal} + \text{Mark} + \text{Startwort}) \\
 T_{min} &= 44\mu s * 512 + (88\mu s + 8\mu s + 44\mu s) \quad (1) \\
 T_{min} &= 22,67ms
 \end{aligned}$$

2.3 Grenzen des DMX512-Standards

Die Erstdefinition des DMX-Standards, welche bis heute nur geringe Änderungen erfahren hat, fand bereits im Jahr 1986 statt. In der Welt der Informationstechnik sind 27 Jahre Entwicklungsunterschied enorm. Ein gutes Beispiel für die Geschwindigkeit der Entwicklung im IT-Bereich stellt das Moore'sche Gesetz dar. Dieses besagt, dass die Integrationsdichte, die Dichte von Transistoren auf einem Computerchip, exponentiell

¹²United States Institute for Theatre Technology

¹³Breaksignal, Startwort und Marks

wächst. Daraus resultiert, dass die Rechenleistung eines Computers ca. alle 18 Monate verdoppelt wird¹⁴. Die Entwicklung von Scheinwerfertechnik ist zwar momentan nur in relativ geringem Maße von Prozessorleistung abhängig, bei den vielen Funktionen von aktuellen Multifunktionsscheinwerfern zeigt sich jedoch bereits, dass die Übertragungskapazität von DMX512 seine Grenzen erreicht, oder bereits überschritten hat. In naher Zukunft wird die Komplexität und Anzahl integrierter Schaltkreise zunehmen, um auch komplexere Netzwerke als DMX512 nutzen zu können.

Das größte Problem liegt in der Skalierbarkeit des Standards. Die einzige Möglichkeit zur Erweiterung der Kanäle, wenn man mehr als 512 Kanäle benötigt, ist die Eröffnung von zusätzlichen DMX-Universen. Mehr Universen resultieren jedoch in einem erheblichen Mehraufwand in der Bereitstellung der Infrastruktur (zusätzliche Verteilungstechnik, mehr Verkabelungsaufwand, etc.). Die Kapazitäten dieser Erweiterungsmöglichkeit stoßen jedoch bei heutigen Produktionen an die wirtschaftlichen und logistischen Grenzen. Moderne Movinglights bieten bspw. zahlreiche Funktionen: die Bewegung diverser Goboräder, Farbwechsler, Pan und Tilt sowie ein variabler Beam benötigen eine Vielzahl an Kanälen. Die Pan- und Tilt-Steuerung benötigt zudem meist jeweils zwei Kanäle um eine höhere Bewegungsauflösung von 16 Bit zu ermöglichen. Als Beispiel sei der Vari-lite VL3015LT genannt, der als einzelner Scheinwerfer 36 DMX-Kanäle benötigt¹⁵. Wie stark das Protokoll bereits in die Jahre gekommen ist, zeichnet sich vor allem an der Schnittstelle zwischen Licht und Video ab. Beim Ansteuern von Licht-Matrizen über DMX wird dies besonders deutlich. Bei einer Größe von 20x20 Pixeln einer einfarbigen Matrix ergeben sich bereits 400 Kanäle, erweitert auf den RGB-Farbraum, sogar 1200 Kanäle.

Ein weiterer nicht mehr zeitgemäßer Teil des Standards ist die unidirektionale Verbindung zu den Beleuchtungsgeräten. Vorgesehen ist ein klassischer Master-Slave-Betrieb. Ein Beleuchtungspult (Master) sendet Steuersignale an die im System befindlichen Dimmer und Scheinwerfer, welche wiederum die empfangenen Signale verarbeiten. Es sind jedoch keine Rückmeldungen der Geräte möglich. Die Beleuchtungsindustrie versucht diesem Umstand seit einigen Jahren beizupflichten und hat sich 2011 auf die DMX-Erweiterung RDM im Standard ANSI E1.20¹⁶ geeinigt. Bestehende DMX-Geräte sollten im Betrieb nicht durch RDM-Anfragen gestört werden. Das hat den Nachteil, dass die Rückmeldungen nicht dauerhaft erfolgen können, sondern nur auf Anfrage versandt werden¹⁷. Dieser Kompromiss war jedoch nötig, um die bestehende Technik weiter nutzen zu können.

Das Protokoll beinhaltet auch keine Fehlererkennung und -korrektur, welche in anderen Netzwerkprotokollen, ein Beispiel ist das seit Jahrzehnten bestehende TCP/IP, längst Standard ist.

¹⁴Vgl. [Hä11] S. 167

¹⁵Vgl. http://www.vari-lite.com/clientuploads/directory/downloads/VL3015LT_User_0.pdf Stand: 30.05.2013

¹⁶vollständiger Name: ANSI E1.20 - 2010 Remote Device Management over DMX512 Networks

¹⁷genaueres im Kapitel 2.4 auf Seite 9

Alle aufgezählten Probleme lassen sich mit Ethernet-basierten Lösungen umgehen. Diese können in die bestehende Netzwerkinfrastruktur integriert werden, um bestehende DMX-Systeme zu erweitern oder langfristig evtl. zu ersetzen.

Diese Umstellung bietet derzeitig aber auch noch große Nachteile. Es gibt eine Vielzahl an Übertragungsstandards. Dadurch ist es dem Endanwender überlassen, sich für ein System, welches am Besten auf seine Bedürfnisse zugeschnitten ist und die größtmögliche Zukunft besitzt, zu entscheiden. Die zunehmende Komplexität dieser Systeme ist ebenso problematisch, wie die durch den Umbau, bzw. die Erweiterung entstehenden Kosten.

Eine weitere große Herausforderung bei der Integration moderner Netzwerkinfrastruktur in konventionelle DMX512-Systeme ist, dass diese momentan das bestehende Netz nicht ersetzen, sondern nur erweitern. In der Praxis werden zusätzliche Nodes¹⁸ in ein Ethernet-Netzwerk eingefügt um von diesem Ort aus, die DMX512-Geräte anzusprechen. Eine Ende-zu-Ende-Verbindung auf Ethernet-Basis ist aktuell mit nur sehr wenigen, hochpreisigen Geräten möglich (siehe Kapitel 4.2 auf Seite 29).

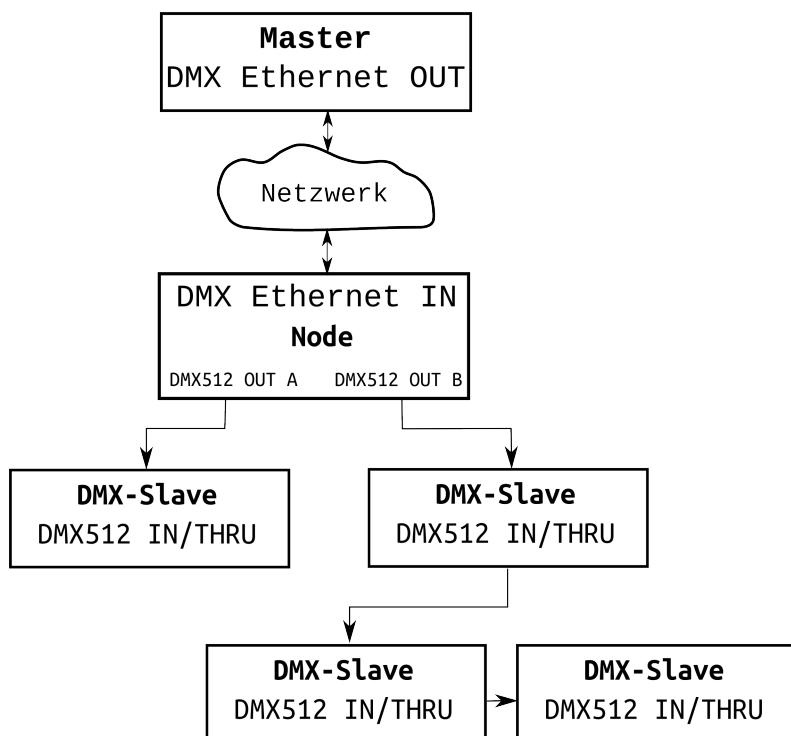


Abbildung 2: vereinfachtes Schema des Aufbaus eines DMX-Ethernet-Netzwerks

Dies liegt, abgesehen von der höheren Komplexität der Empfängertechnik, auch an den weniger robusten RJ45-Steckverbindern. Diese sind mechanisch deutlich empfindlicher als XLR-Stecksysteme¹⁹. Die steiferen Cat-Kabeltypen stellen noch einen Nachteil dar, da diese sich schwieriger verlegen lassen und somit für den Bühnenbetrieb ungeeignet sind. Flexible Cat-Kabel sind zwar bereits am Markt erhältlich, jedoch im preislichen

¹⁸Netzwerknoten und Protokollwandler

¹⁹eine Ausnahme stellen angepasste Steckverbinderysteme, bspw. Ethercon, der Firma Neutrik dar

Vergleich noch deutlich teurer als konventionelle DMX-Kabel.

In absehbarer Zeit, werden diese kombinierten Lichtsteuer-Systeme am Markt vorherrschend sei. Darum ist es nötig, Lösungen zu finden, welche die maximale Leistungsfähigkeit dieser hybriden Lösungen ausnutzen.

2.4 RDM - Remote Device Management

Eine wichtige Evolution in der Entwicklung von DMX512-A stellt die Erweiterung RDM dar. Diese ermöglicht es, Rückmeldungen der Geräte im DMX-Bus zu erhalten. Obwohl auf EIA-485 aufbauend, wurde der Rückkanal bei DMX512-A standardmäßig nicht implementiert, sondern erst durch diese zusätzliche Formulierung nachgereicht. Der aktuelle Standard zu RDM wird definiert durch ANSI E1.20 - 2010²⁰.

Ein Beispielszenario für die Nutzung von RDM ist die Fernkonfiguration von DMX-Geräten. Am Lichtsteuerpult mit RDM-Unterstützung können alle RDM-fähigen Geräte im Netzwerk ermittelt und anschließend Einstellungsparameter, bspw. der DMX-Kanal, überprüft und verändert werden.

In dem physischen Aufbau der DMX-Systeme ändert sich für den Anwender nichts. Die Topologie und Verkabelung der Geräte innerhalb des DMX-Bus ist bei RDM identisch. Die einzigen physischen Änderungen der Hardware umfassen die Treiberschaltungen, da diese bei RDM auf einer Leitung Senden und Empfangen müssen. Dies resultiert in einer Änderung der verwendeten Sende- und Empfangselektronik. Das ist auch der Grund, warum man die RDM-Funktionen nicht nachträglich in einem Gerät implementieren kann, sondern Neuentwicklungen der Hardware benötigt. Will man neue RDM-fähige Geräte in ein bestehendes System integrieren, wird es nötig, alle Geräte zur Signalverteilung durch RDM-fähige Hardware zu ersetzen. Controller (das Lichtsteuerpult), Booster, Splitter und das zu steuernde Gerät müssen RDM-Funktionalität implementiert haben.

2.4.1 Funktionsweise RDM

Anders als bei DMX, findet die Übertragung der Daten bei RDM paketweise statt. Eine Übersicht des Paketaufbaus findet sich in Tabelle 3. Zum Versenden der RDM-Pakete wird der DMX-Datenstrom kurzzeitig unterbrochen, um das RDM-Paket in die bestehende Übertragung zu integrieren²¹.

Die RDM-Kommunikation lässt sich in drei Arten klassifizieren. Ein Discovery Signal inklusive Rückantwort, eine Unicast-Kommunikation zu bereits bekannten Geräten und eine Broadcast-Kommunikation zur Kontrolle aller RDM-Geräte, oder aller Geräte eines Herstellers gleichzeitig. Jedes übermittelte RDM-Signal enthält einen vom DMX512-Standard abweichenden Startcode mit dem Hexadezimalwert 0xCC und einen darauf

²⁰voller Titel ANSI E1.20 - 2010 RDM Remote Device Management Over DMX512 Networks

²¹Vgl. [Ric09] S. 9

folgenden Sub-Startcode²² 0x01. So wird sichergestellt, dass die konventionellen Geräte die RDM-Nachrichten nicht verarbeiten.

Es kann jedoch bei schlechten Implementierungen der Startcode-Erkennung zu einem Flackern während des Betriebs kommen. Bei standardkonformen Geräten darf und sollte dies jedoch nicht geschehen.

Slot-Nr.	Bedeutung
0	alternativer Startcode (0xCC)
1	Sub-Startcode (0x01)
2	Paketlänge in Bytes
3-8	Zieladresse (UID)
9-14	Quelladresse (UID)
15	Transaktionsnummer
16	Port-ID / Typ der Antwort
17	Anzahl der wartenden Nachrichten
18-19	Sub-Device
20	Befehlsart
21-22	Parameter-ID (PID)
23	Länge der Nutzdaten in Bytes
24-(n-2)	Nutzdaten
(n-1)-n	Prüfsumme

Tabelle 3: Aufbau eines RDM-Pakets

Vgl: [Hoe08]

RDM-Discovery

Zu Beginn müssen einmalig alle RDM-Geräte im DMX-Netzwerk erkannt werden. Dies geschieht zunächst mit einem Broadcast-UNMUTE-Befehl und einer Discovery-Nachricht, die an alle Geräte versandt wird²³. Hier wird jedoch die größte Herausforderung des Standards deutlich. Die Rückmeldungen erfolgen nicht auf einem separaten Kanal, obwohl bei dem 5-poligen Steckverbinder auch eine Rückleitung möglich gewesen wäre. Sie werden im Semi-Duplex im Wechsel mit den Sendesignalen übertragen. Diese Entscheidung ist wohl darauf zurückzuführen, dass mehrere Hersteller bereits andere Verwendungen für Pin 4/5 des XLR-Anschlusses genutzt haben und die aktuellen DMX-Normen einen zweiten Datenkanal auf diesen beiden Leitungen spezifizieren.

Senden nun alle Geräte direkt nach Erhalt der Discovery-Nachricht ihre Antwort, kommt es unweigerlich zu Kollisionen und Überlagerungen der Signale. Also muss der Bereich der antwortenden Geräte eingegrenzt werden. Um dies zu bewerkstelligen, werden die Anfragen nun gezielt gestellt, dass heißt der Adressraum wird immer weiter eingeschränkt, bis die Rückantwort eindeutig verifizierbar ist.

Dies geschieht aus informationstechnischer Sicht mit dem binären Suchbaum²⁴.

²²der Sub-Startcode soll in zukünftigen Versionen des Standards zur Angabe abweichender Paketstrukturen dienen

²³Vgl. [New12]

²⁴häufig abgekürzt als BST (binary search tree)

Dabei werden der Reihe nach alle möglichen Gerätekennungen abgearbeitet. Andere Algorithmen sind möglich, die binäre-Suchbaum-Methode wird jedoch empfohlen²⁵. Nach Erkennung eines Geräts wird die eindeutige Gerätekennung (UID²⁶) gespeichert. Es erfolgt zusätzlich eine Stummschaltung (MUTE) des gespeicherten Geräts, so dass es nicht mehr auf Discovery-Nachrichten antwortet. Dieser Vorgang wird so lang wiederholt, bis keine Geräte mehr auf eine Discovery-Anfrage reagieren²⁷.

RDM-Unicast und RDM-Broadcast

Sind nun alle Geräte mit ihrer UID abgespeichert, können gezielt Daten an sie versendet und von ihnen empfangen werden. Diese Unicast-Verbindung²⁸ erfolgt mittels SET- und GET-Kommandos. Ein SET-Kommando schickt Daten an die Geräte, das GET-Kommando ruft Informationen von den Geräten ab. Jedes Gerät quittiert den Empfang eines Steuerbefehls mit einer Rückantwort²⁹.

Die Art der gesendeten und empfangenen Daten bestimmt die PID (Parameter-ID). Jedes RDM-fähige Gerät muss eine vordefinierte Mindestanzahl abrufbarer Parameter unterstützen, so dass einige grundlegende Funktionen von jedem RDM-Gerät sicher unterstützt werden. Dazu zählen bspw. der Abruf von Geräteinformationen, Betriebsstunden und Sensordaten sowie die Einstellung der DMX-Adresse. Es ist jedoch auch möglich, herstellerspezifische Kommandos zu implementieren. Das Einstellen der Parameter über SET-Befehle ist zusätzlich über Broadcast-Nachrichten möglich. In diesem Fall lautet die UID der Zieladresse 0xFFFFFFFFFFFF³⁰. Auf diese Weise ist es möglich, mit nur einem RDM-Paket eine Veränderung an allen RDM-Geräten vorzunehmen. Um Kollisionen zu vermeiden, sind Rückantworten der RDM-Geräte für Broadcast-Nachrichten nicht gestattet.

Problemstellungen des RDM-Standards

Trotz der vielversprechenden Neuerungen bietet RDM im Betrieb auch gewisse Nachteile, welche größtenteils auf die Kompatibilität mit konventionellen DMX512-Geräten zurückzuführen sind.

Aus den Kompatibilitäts-Anforderungen resultiert bspw. die Problematik der Semi-Duplex-Übertragung. RDM-Geräte können, um Kollisionen zu vermeiden, nur auf Anfragen des Controllers antworten. Dringende Warnmeldungen eines Scheinwerfers, innerhalb eines großen Systems, können somit nur durch regelmäßige Abfragen erkannt werden. Um den DMX-Datenstrom nicht zu stark einzuschränken, können RDM-Anfragen nur sehr begrenzt gestellt werden. Die Erkennung einer dringenden Fehlermeldung geschieht so im Praxisfall vielleicht zu spät.

²⁵Vgl. [ANS10] S. 30

²⁶48 Bit Unique Identifier, besteht aus 16 Bit Hersteller-ID und 32 Bit Geräte-ID

²⁷Vgl. [ANS10] S. 31

²⁸näheres zu Unicast- und Broadcastverbindungen in 3.1.3 auf S. 16

²⁹Vgl. http://www.soundlight.de/techtips/dmx512/dmx_rdm.htm Stand: 24.05.2013

³⁰Vgl. http://www.soundlight.de/techtips/dmx512/dmx_rdm.htm Stand: 24.05.2013

Da RDM nur auf dem bestehenden DMX-Datenstrom aufbaut, resultiert jede RDM-Anfrage in einer Bandbreitenverringerung, der ohnehin schmalen Datenrate von DMX512. In Fachkreisen wird, bezüglich der RDM-Geschwindigkeit, oftmals argumentiert, dass RDM immer noch schneller sei, als erneut “einen Techniker auf die Leiter zu schicken”³¹. Da RDM vorrangig für die Konfiguration und Statusabfrage benötigt wird, ist die Datenrate in der Praxis nur ein geringes Problem.

Ein weiterer Kritikpunkt am RDM-Standard ist, dass die komplette DMX-Infrastruktur an die bidirektionale Übertragung angepasst werden muss. Dies betrifft Controller, Booster und Splitter, was einen nicht zu verachtenden Kostenaufwand bei der Umstellung eines Systems hervorrufen kann.

Die zeitliche Überschneidung der Veröffentlichung des DMX-RDM-Standards und das Aufkommen von Ethernet-Lichtsteuerprotokollen, resultierte in einem hohen zusätzlichen Entwicklungsaufwand für die Hersteller der Lichttechnik-Branche. Diese Kombination aus Ethernet-Protokollen und RDM stellt, speziell in Bezug auf herstellerübergreifende Kompatibilität, hohe technische Anforderungen. Gleichzeitig bietet es jedoch vielversprechende Möglichkeiten für die Zukunft der Lichttechnik.

³¹Vgl. [Kel10] S. 169

3 Der Ethernet-Standard in der modernen Lichtsteuerung

3.1 Die Ethernet-Technologie

Ethernet ist einer der grundlegendsten Netzwerkstandards in der Computertechnik. Er spezifiziert die unter der Arbeitsgruppe IEEE 802.3 definierten Software- und Hardwrebestandteile eines Netzwerksystems. Da diese Technologie anfangs für lokale Netze gedacht war, hat sich im Sprachgebrauch auch die Begrifflichkeit der LAN-Technik nahezu synonym etabliert. Ethernet ermöglicht den Datenaustausch zwischen Geräten im Netzwerk in Form von Ethernet-Rahmen. Dabei sind Geschwindigkeiten von 10MBit/s, 100MBit/s (Fast-Ethernet), 1000MBit/s (Gigabit-Ethernet) sowie bis zu 100GBit/s standardisiert. Zu Beginn bestand das Ethernet-Netzwerk aus einem koaxialem Bus, auf den die Netzwerkteilnehmer gemeinsam zugegriffen haben. Der Anschluss der Teilnehmer erfolgte damals über zusätzliche Transceiver oder T-Stücke. Jeweils am Leitungsende musste eine Terminierung der Leitung erfolgen, um Signalreflexionen zu vermeiden. Heute bestehen die Leitungen nahezu ausschließlich aus Twisted-Pair- und Glasfaserleitungen. Eine einzelne gemeinsam genutzte Busleitung ist nicht mehr üblich, heutige Netze nutzen meist sternförmige Verkabelungen.

3.1.1 Netzwerktopologien

Netzwerkstrukturen sind nach klar definierten Regeln aufgebaut. Es ist genau definiert wie, und über welche Zwischenstellen, Netzwerkteilnehmer miteinander verbunden sind. Im Folgenden werden die für die lichttechnischen Netzwerke relevanten Topologien vorgestellt.

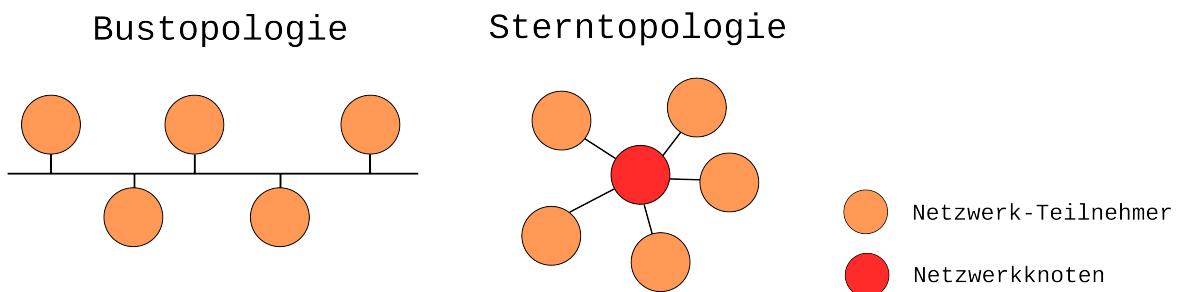


Abbildung 3: Schemata Netzwerktopologien

Die für unseren Anwendungsbereich entscheidenden Strukturen sind die Bus- und die Sterntopologie. Des Weiteren wird noch die Abwandlung der Sterntopologie, die Baumtopologie betrachtet.

Bustopologie

Seit Jahren in der Anwendung für Lichtsteuernetzwerke ist die, vom DMX512-Standard her bekannte, Bustopologie. Sie ermöglicht das Nutzen eines Mediums für viele Teilnehmer innerhalb des Busses. Eine gemeinsam genutzte Leitung hat den Vorteil, dass die Kabelwege zwischen den einzelnen Teilnehmern sehr kurz sind. Die Kabellänge ist stets nur so lang, wie die Entfernung zum Bus.

Der Nachteil eines gemeinsam genutzten Mediums für das Versenden von Signalen ist, dass jeweils nur ein Gerät Senden kann. Die restlichen Teilnehmer, können alle das ausgesandte Signal erkennen, sind jedoch selbst nur dann in der Lage, Signale zu versenden, wenn kein anderes Gerät sendet. So vermeidet man Signalüberlagerungen, die in unlesbaren Signalen resultieren. Diese können auch entstehen, indem die Busleitungen nicht abgeschlossen, also terminiert, werden. Dies ist nötig, um Signalreflexionen an den Kabelenden zu vermeiden, welche ebenso überlagerte Signale hervorrufen.

Ein weiterer Kritikpunkt wird im Fehlerfall deutlich. Ist die Busleitung an einer Stelle unterbrochen, können keine Signale an alle Teilnehmer hinter der Bruchstelle versandt werden. Zusätzlich fehlt nun die Terminierung am Kabelende, so dass Signalreflexionen auftreten.³²

Sterntopologie

Die in Ethernet-Netzwerken heutzutage übliche Topologie ist der Stern. Der Aufbau besteht aus einem zentralen Knoten, der der Signalverteilung dient, und den daran angeschlossenen Netzwerkeinheiten (siehe Abb. 3). Der zentrale Netzwerkknoten kann ein Hub, Switch oder Router sein³³. Bei Ausfall einer einzelnen Station kann das Netzwerk weiterhin problemlos funktionieren. Fällt jedoch der zentrale Knotenpunkt aus, kann keine Kommunikation zwischen den einzelnen Stationen mehr aufgebaut werden. Da es oft nötig ist, die Netzwerke auf größere Bereiche zu erweitern, kaskadiert man mehrere Netzwerknoten. Diese oft hierarchisch strukturierten Sterntopologien werden auch als Baumstruktur bezeichnet.

Die Baumstruktur stellt jedoch in der Praxis eine der größten Herausforderungen im Betrieb von Ethernet-gesteuerten Lichtanlagen dar. In größeren Netzwerken mit Ethernet-Anschlüssen in mehreren Etagen eines Gebäudes erfolgt die Verteilung zum Einen vertikal von Etage zu Etage, zum Anderen horizontal vom zentralen Netzwerknoten einer Etage (Etagenverteiler) zu den einzelnen Netzwerkeinheiten. Die Verbindungen vom Etagenverteiler zum Endgerät dürfen maximal eine Länge von 100 m pro Twisted-Pair-Kabel (bzw. 200m bei hochwertigen Kabeln, ab Kategorie 5)³⁴ aufweisen. Sollen nun alle Geräte eines lichttechnischen Netzwerks über Ethernet gesteuert werden, wird es nötig, zu jedem einzelnen Gerät eine Verbindung zum Etagenverteiler zu verlegen. Dadurch entstehen hohe zusätzliche Infrastrukturkosten und ein personeller Mehraufwand

³²Vgl. [Zei11] S. 13

³³zur näheren Erläuterung von Hub, Switch und Router siehe Kapitel 3.1.5 S. 20

³⁴Vgl. [TW12] S. 337

bei der Installation. Der große Vorteil liegt jedoch in der dadurch gewonnenen Betriebssicherheit. Fällt eine Leitung zu einem Gerät aus, sind alle anderen Netzwerkeinnehmer davon nicht betroffen und können weiter störungsfrei betrieben werden.

Für den Praxiseinsatz resultiert daraus, dass Lichtsteuernetzwerke, welche komplett aus einem Ethernet-Netzwerk bestehen, nur schwer und mit einigen Nachteilen realisierbar sind. Der aktuell verwendete Ansatz ist eine Mischung aus DMX-Bus- und Ethernet-Stern-Topologien. Dadurch ist es möglich die Vorteile der einzelnen Systeme zu kombinieren. Die Balance zwischen Ethernet- und DMX-Bestandteilen eines Steuersystems ist stark abhängig von den Anforderungen des Anwenders.

3.1.2 Ethernet-Datenkomponenten

Die Daten in einem Ethernet-Netzwerk werden von Gerät zu Gerät über sog. Ethernet-Frames versendet. Diese haben eine Mindestlänge von 64 Byte und eine Maximallänge von 1518 Bytes. Die Mindestlänge ist begründet durch die Art der implementierten Fehlererkennung CSMA/CD³⁵.

Um Kollisionen zu vermeiden prüft bei CSMA/CD jeder Sender vor dem Sendevorgang zunächst, ob die Leitung bereits belegt ist. Falls ja, wartet dieser bis die Leitung wieder frei ist. Andernfalls beginnt er das Senden. Kommt es während der Übertragung zu keiner Kollision, wird die Übertragung erfolgreich beendet³⁶. Tritt jedoch während des Übertragungsvorgangs eine Kollision auf, beenden beide Sender die Übertragung und warten eine per Zufallsprinzip definierte Zeitspanne, bis sie erneut den Sendevorgang starten. Durch eine Kollision während der Übertragung entsteht eine Überlagerung der Signale. Diese resultiert in einem erhöhten Signalpegel, auch als Kollisionspegel bezeichnet, der zum Abbruch der Datenübertragung führt. Um jedoch überhaupt eine Kollision feststellen zu können, ist es notwendig, abhängig von der Leitungslänge eine Mindestlänge der Datenframes zu definieren. Diese Mindestlänge ist abhängig von der Signalausbreitungsgeschwindigkeit und der Übertragungsrate. Obwohl CSMA/CD weiterhin in der Ethernet-Definierung Bestand hat, findet es in der Praxis kaum Anwendung, da es durch den Einsatz von Vollduplexverbindungen und Switches zu keinen Kollisionen auf der Leitung kommen kann. Heutige Übertragungsfehlererkennungen arbeiten meist auf höheren Protokollebenen des OSI-Modells (genaueres siehe Kapitel 3.1.4, Seite 17).

Um alle Teilnehmer für das Senden und Empfangen von Paketen eindeutig identifizieren zu können, besitzt jedes Gerät im Verbund eine eindeutige Kennung, die aus 48-Bit bestehende MAC-Adresse³⁷. Die MAC-Adressen von Sender und Empfänger werden gemeinsam mit einer Präambel und dem sog. SFD³⁸ im Ethernet-Header übertragen. Präambel und SFD dienen zur Synchronisation der Bit-Abstände zwischen verschiedenen Geräten. Das erste Bit des ersten Bytes einer MAC-Adresse gibt an, ob es sich um

³⁵Carrier Sense Multiple Access/Collision Detection

³⁶Vgl. [Zei11] S. 16

³⁷Media-Access-Control-Adresse

³⁸Start Frame Delimiter

eine Einzel- oder Gruppenadresse handelt³⁹⁴⁰. Wichtig ist dies für die Übertragung von Lichtsteuersignalen, da so kenntlich gemacht wird, ob die gesendeten Daten im Broadcast, Multicast ($I/G = 1$) oder Unicast übermittelt werden (genaueres siehe Kapitel 3.1.3 auf Seite 16).

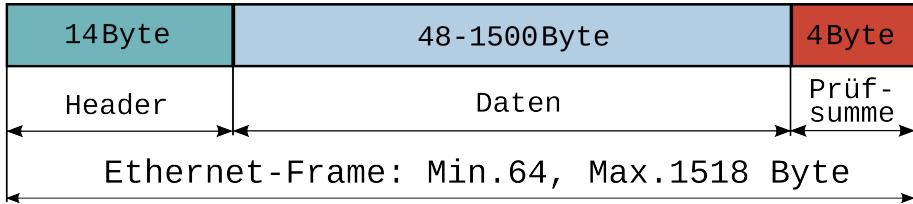


Abbildung 4: vereinfachter Aufbau eines Ethernet-Frames

Den Abschluss jedes Ethernet-Frames bildet das FCS-Feld⁴¹. Dieses enthält eine 32-Bit CRC-Prüfsumme⁴². Dies ist die aus den Daten im Ethernet-Frame generierte Prüfsumme, welche beim Erstellen vom Sender an das Ende des Frames angehängt wird. Nach dem Empfang des Frames an seinem Zielort, erstellt der Empfänger erneut eine Prüfsumme aus den empfangenen Daten und vergleicht diese mit der empfangenen CRC-Prüfsumme. Stimmt diese nicht überein verwirft der Empfänger die Daten und fragt diese erneut an.

3.1.3 Unicast, Multicast und Broadcast

Wenn Geräte in einem Ethernet-Netzwerk miteinander kommunizieren, nutzen sie entweder Unicast, Broadcast oder Multicast. Das Standard-Übertragungsszenario, also das Senden von einem Paket zwischen zwei Teilnehmern, wird als Unicast bezeichnet⁴³. Dies ist sinnvoll für alle Übertragungen, welche nur an einen bestimmten Empfänger gerichtet werden sollen⁴⁴. Unicast hat jedoch den Nachteil, dass es nicht performant für die Übertragung großer Datenmengen an mehrere Teilnehmer skaliert. Jeder Frame muss für jeden Empfänger einzeln versendet werden, was zum unnötigen Versenden vieler identischer Ethernet-Frames und somit zum Verschwenden von Bandbreite führt.

Um effektiver große Mengen an Frames an viele Teilnehmer versenden zu können gibt, es zusätzlich noch die Möglichkeit Pakete im Broadcast zu versenden. Bei einem Broadcast wird ein Paket an alle Netzwerkteilnehmer versendet⁴⁵. Dadurch wird es nötig, dass jeder Empfänger die Zieladresse des Pakets überprüfen muss, um zu erkennen, ob es für ihn bestimmt ist. Broadcast-Übertragungen funktionieren nur in einem ganz speziellen Subnet⁴⁶, meist dem lokalen Subnetz, und nicht über dessen Grenzen hinaus.

³⁹daher auch die Bezeichnung I/G des ersten Bits für Individual/Group

⁴⁰Vgl. <https://de.wikipedia.org/wiki/MAC-Adresse> Stand: 30.06.2013

⁴¹Frame Check Sequence

⁴²cyclic redundancy check - zyklische Redundanzprüfung

⁴³Vgl. [TW12] S. 39

⁴⁴eine Beispielanwendung dafür ist die Dateiübertragung oder die Internetnutzung

⁴⁵Vgl. [TW12] S. 40

⁴⁶auch als Broadcast-Domäne bezeichnet

Diese Art der Anfragen sind perfekt, um einen bestimmten Teilnehmer innerhalb des Netzwerks zu ermitteln und mit diesem anschließend eine Unicast- oder Multicastverbindung aufzubauen.

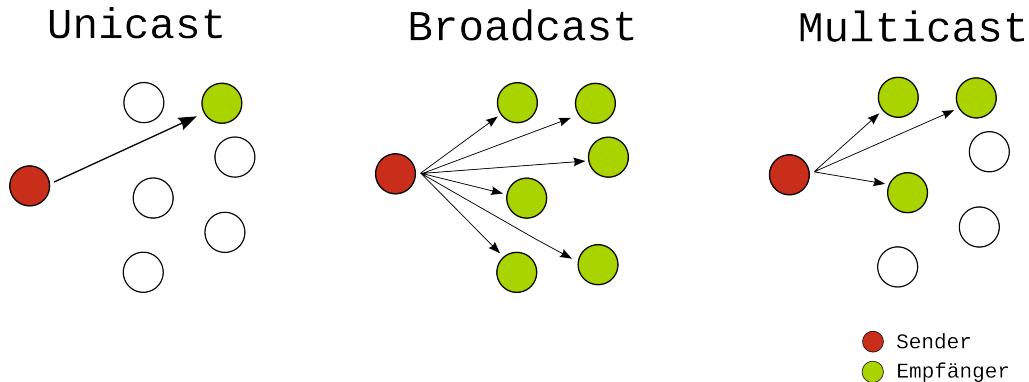


Abbildung 5: schematische Darstellung Unicast, Broadcast und Multicast

Multicast ist eine Mischung aus Broadcast und Unicast. Es ermöglicht das gezielte Senden an vorher definierte Empfänger, wie bei Unicast üblich, jedoch auch das Senden an mehrere Empfänger gleichzeitig, wie bei Broadcast üblich, ohne dabei an alle Teilnehmer senden zu müssen⁴⁷. Multicast-Verbindungen sind zwar effizienter als Broadcast-Verbindungen, da die Pakete nur an eine Teilmenge der Netzwerkteilnehmer versandt werden müssen, setzen jedoch auch einen zusätzlichen Konfigurationsaufwand voraus. Der Versand von Paketen erfolgt bei Multicast an Gruppenadressen, dies macht es notwendig die Geräte vor dem Versand einzelnen Gruppen zuzuordnen und diese zu verwalten.

Die am besten geeignete Übertragungsart ist von der Art der Anwendung abhängig. Sollen bspw. Lichtsteuerdaten zwischen zwei Punkten, einem Steuerpult und einem Ethernet-DMX-Node, versendet werden, ist Unicast die beste Variante. Für die Erstkonfiguration von vielen Ethernet-fähigen Scheinwerfern ist ein Broadcast die empfehlenswerte Variante, um die einzelnen Gerätadressen im Netzwerk zu ermitteln. Sollen Steuerdaten direkt vom Lichtsteuerpult an mehrere Ethernet-kompatible Movinglights geschickt werden, ist Multicast die effizienteste Methode.

3.1.4 Das ISO/OSI-Referenzmodell

Um zu verstehen, wie die Kommunikation zwischen Teilnehmern eines Netzwerks funktioniert, bedient man sich in der IT-Technik mehrerer abstrakter Referenzmodelle, welche die einzelnen Funktionen während der Kommunikation beschreiben. Das bekannteste, von der ISO⁴⁸ entwickelte Modell ist das OSI-Referenzmodell⁴⁹. Es umfasst sieben Schichten, welche jeweils einen abgegrenzten Aufgabenbereich beschreiben. Dieses Mo-

⁴⁷Vgl. [TW12] S. 40

⁴⁸International Standards Organization

⁴⁹OSI - Open Systems Interconnection

dell spezifiziert selbst keine Dienste und Protokolle, sondern legt lediglich die Aufgaben innerhalb der einzelnen Schichten fest⁵⁰. Eine schematische Darstellung des OSI-Modells findet sich in Abb. 6⁵¹.

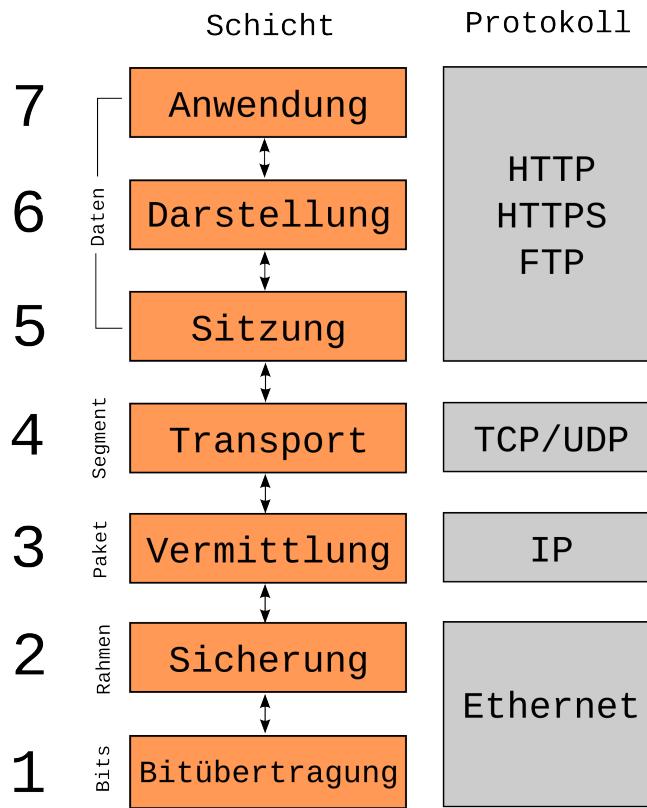


Abbildung 6: OSI-Schichtenmodell

Layer 1 - Die Bitübertragungsschicht (Physical Layer)

Die Bitübertragungsschicht beschreibt die Art der Signale, also bspw. wie werden die Logikpegel repräsentiert, und das Medium, auf welchem sie sich ausbreiten. Es werden hier also die physischen und elektrischen Eigenschaften, Kabeltypen, Steckverbinder, Spannungspegel usw. definiert⁵².

Layer 2 - Die Sicherungsschicht (Data Link Layer)

Während Layer 1 einen „rohen“ Strom an Bits generiert, besteht die Aufgabe in Layer 2 darin, diese Bits in Frames zusammenzufassen. Eine konkrete Bitfolge bestimmt Anfang und Ende eines Frames und signalisiert somit Start und Ende eines Datenrahmens. Diese Rahmen werden nacheinander, also sequentiell, übertragen. Nach einer erfolgreichen Übertragung zum Empfänger quittiert dieser den Empfang mit einem Bestätigungsrah-

⁵⁰Vgl. [TW12] S. 66

⁵¹ein detaillierteres Schema des OSI-Schichtenmodells findet sich im Anhang in Abb. 28

⁵²Vgl. [TW12] S. 68

men⁵³ ⁵⁴.

Auf der Sicherungsschicht, sowie auf den meisten höheren Layern, wird auch festgelegt, wie man verhindert, dass ein schneller Sender einen langsamen Empfänger mit Daten überflutet. Hierzu ist es nötig, dem Sender mitzuteilen, wann der Empfänger für weitere Daten bereit ist.

Weiterer Bestandteil dieses Layers ist die MAC-Teilschicht. Diese regelt die Nutzung des gemeinsamen Kanals, um Kollisionen zu vermeiden. Anhand der MAC-Adressen wird auch die Verteilung der Datenpakete im Netzwerk über Switches geregelt (genaueres siehe Kapitel 3.1.5 Seite 20).

Layer 3 - Die Vermittlungsschicht (Network Layer)

Die Hauptaufgabe der Vermittlungsschicht ist das Routen von Paketen, also die Festlegung, über welchen Weg die Pakete vom Ursprungs- zum Bestimmungsort gelangen. Es stellt somit die Verknüpfung zwischen unterschiedlichen Netzwerken her. Einer der elementarsten Standards dieser Modellebene ist das IP-Protokoll. Infolgedessen findet innerhalb der Vermittlungsschicht auch die Umsetzung von IP-Adressen in MAC-Adressen statt⁵⁵.

Da bei einem Broadcast meist nur das aktuelle Subnetz verwendet wird, treten dabei ⁵⁶keine Routing-Problematiken auf. Die OSI-Schicht 3 spielt bei diesen Netzen nur eine nebensächliche Rolle.

Layer 4 - Die Transportschicht (Transport Layer)

Die Transportschicht ist für die zuverlässige Zustellung der Pakete verantwortlich. Auf welche Weise diese Zustellungskontrolle funktioniert, ist vom verwendeten Protokoll abhängig⁵⁷. Typische Protokolle der Transportschicht sind TCP und UDP. TCP ist ein verbindungsorientiertes und UDP ein verbindungsloses Protokoll. Das bedeutet, dass bei TCP eine direkte Verbindung zwischen den beiden Teilnehmern, Sender und Empfänger, hergestellt wird und der Empfänger jedes empfangene Paket bestätigt. Bei UDP dagegen verzichtet man auf diese Empfangsbestätigungen und erreicht so höhere Datenraten bei der Übertragung, jedoch auf Kosten der geringeren Zuverlässigkeit.

Layer 5 - Die Sitzungsschicht (Session Layer)

Auf- und Abbau von Sitzungen zwischen Netzwerkteilnehmern sind grundlegende Aufgaben der Sitzungsschicht. Sie stellt innerhalb dieser Sitzungen verschiedene Dienste zur Verfügung, um einen geordneten Dialog der Teilnehmer zu ermöglichen. Layer 5 regelt also, dass nachdem Sitzungen geöffnet wurden, die Geräte Informationen austauschen

⁵³acknowledgement Frame

⁵⁴Vgl. [TW12] S. 68

⁵⁵Vgl. [Zei11] S. 30

⁵⁶Vgl. [TW12] S. 39

⁵⁷Vgl. [TW12] S. 69

können und nach erfolgreicher Übertragung die Sitzung beendet wird⁵⁸.

Layer 6 - Die Darstellungsschicht (Presentation Layer)

Unterschiedliche Endsysteme stellen Daten innerhalb des Systems auf verschiedene Weise dar, die Darstellungsschicht formatiert diese Daten in ein einheitliches Format und ermöglicht somit eine standardisierte Kommunikation. In ihr werden bspw. Zeichen encodiert und Daten komprimiert, sowie ver- und entschlüsselt⁵⁹.

Layer 7 - Die Anwendungsschicht (Application layer)

Die Anwendungsschicht ist die direkte Schnittstelle zwischen Nutzer und Computer. In ihr finden die typischen Protokolle für die Übertragung von Daten, Druckaufträgen und den Austausch von Emails ihre Anwendung⁶⁰.

3.1.5 Bestandteile eines Ethernet-Netzwerks

Für den Aufbau einer Ethernet-Netzwerkinfrastruktur sind Geräte zur Verbindung der Netzwerkteilnehmer und der Verteilung der Ethernet-Frames nötig. Die wichtigsten Bestandteile für den Aufbau eines Ethernet-Netzwerks werden in diesem Kapitel dargestellt.

Netzwerkmodule

Jedes Gerät innerhalb eines Netzwerks muss, um mit anderen Netzwerkteilnehmern kommunizieren zu können, über ein Netzwerkmodul⁶¹ verfügen. Dieses auch als NIC bezeichnete Erweiterungsmodul stellt die Verbindung zwischen dem Gerät und dem Übertragungsmedium (meist Twisted-Pair-Kabel oder Funkwellen) her und ermöglicht dadurch die Kommunikation im Netzwerk. Das Netzwerkmodul bestimmt die unterstützten Übertragungsgeschwindigkeiten, und ob etwaige Zusatzimplementierungen, bspw. Wake On Lan (WOL) oder Power over Ethernet (PoE) unterstützt werden. Jedes Netzwerkmodul verfügt, wie in Kap. 3.1.2 beschrieben, über eine eindeutige Kennung, die MAC-Adresse. Hauptaufgabe des NIC ist es, Daten für den Transport auf dem Übertragungsmedium vorzubereiten und zu steuern, wann und wie gesendet und empfangen werden kann.

Hub

Innerhalb eines Netzwerkverbunds, benötigt es Hardware, um die Signale im Netzwerk zu verteilen. Die einfachste Art der Verteilung gelingt mittels eines Hubs. Dieser verbindet die Teilnehmer elektrisch an allen angeschlossenen Ports auf der Bitübertragungsschicht⁶² (siehe Kap. 3.1.4).

⁵⁸Vgl. [Zei11] S. 30

⁵⁹Vgl. [Zei11] S. 30

⁶⁰Vgl. [TW12] S. 71

⁶¹in PC's oft auch als Netzwerkkarte bezeichnet

⁶²Vgl. [TW12] S. 395

Jedes ankommende Signal wird somit immer an alle angeschlossenen Geräte des Hubs versendet. Zusätzlich erfüllt der Hub noch die Funktion eines Repeaters, um die Reichweite innerhalb eines Netzwerks zu erhöhen.

Switch

Ein Switch, obwohl äußerlich sehr ähnlich, unterscheidet sich grundlegend von einem Hub. Der Switch leitet die ankommenden Frames nicht an alle angeschlossenen Ports weiter, sondern nur an den Port, an welchen das Zielgerät angeschlossen ist. Dies vermeidet Kollisionen und verhindert unnötigen Datenverkehr⁶³.

Ein Switch arbeitet im Gegensatz zu einem Hub ab OSI-Schicht 2. Man unterscheidet zwischen managed und unmanaged Switches. Unmanaged Switches sind für den Betrieb in kleineren Netzen, managed Switches für komplexere Anforderungen geeignet. Managed Switches bieten zusätzliche Steuer- und Überwachungsfunktionen, haben meist eine höhere Portanzahl und die Möglichkeit einer Fernkonfiguration.

Die einfachste Art eines Switches arbeitet ausschließlich auf OSI-Schicht 2. Er verteilt die Ethernet-Frames nach ihren Ziel-MAC-Adressen. Dazu legt sich jeder Switch eine SAT (Source Address Table) an, in denen, die am Switch befindlichen MAC-Adressen zusammen mit dem Port an welchem sie angeschlossen sind, gespeichert werden. Anschließend wird jede ankommende Ziel-MAC-Adresse im Speicher abgerufen und je nach SAT-Eintrag weitergeleitet⁶⁴. Managed Switches arbeiten meist auch auf höheren Protokollebenen⁶⁵ und bieten somit die Möglichkeit, Pakete ähnlich wie ein Router auch nach IP-Adressen weiterzuleiten⁶⁶.

Ein Switch ist generell, wie ein Hub, kaskadierungsfähig und kann somit hintereinander geschaltet werden. Der große Vorteil des Switches liegt in der Geschwindigkeit. Ein Switch hat zwar, da Pakete erst den Ports zugewiesen werden müssen, eine höhere Latenzzeit⁶⁷ als ein Hub. Durch die dadurch drastisch verringerten Kollisionen nimmt die Gesamtlatenz jedoch stark ab. Verringert sich bei einem Hub mit jeder aktiven Verbindung die gesamte verfügbare Übertragungskapazität, ist dagegen bei einem Switch stets die volle Datenrate nutzbar.

Die Kaskadierung von Switches könnte außerdem eine hohe Relevanz für den Aufbau von zukünftigen Lichtsteuernetzen haben. Sollte in Zukunft die Steuerung der Lichttechnik ausschließlich über Ethernet stattfinden, resultiert die herkömmliche Verkabelung eines Ethernet-Netzwerks in einigen Problemstellungen. Die konventionelle Verkabelung findet, wie in Kapitel 3.1.1 beschrieben, zentral von einem Etagenverteiler (strukturierter Verkabelung) statt. Dies resultiert in ausfallsicheren Systemen, jedoch auch in einer deutlich größeren Gesamtkabellänge gegenüber eines DMX-Bus-Systems. Eine Mög-

⁶³Vgl. [TW12] S. 395

⁶⁴Vgl. [Zei11] S. 21

⁶⁵ab OSI-Schicht 3

⁶⁶Vgl. [Zei11] S. 22

⁶⁷diese liegt bei 100BaseTX pro Switch bei ca. 5-20µs Vgl. [https://de.wikipedia.org/wiki/Switch_\(Netzwerktechnik\)](https://de.wikipedia.org/wiki/Switch_(Netzwerktechnik)) Stand: 30.05.2013

keit dies zu umgehen wäre der Einbau eines 3-Port Switches in jedes zu steuernde Gerät innerhalb des Netzwerks, so dass die Steuersignale weiterhin von Gerät zu Gerät weitergegeben werden könnten. Kritische Punkte sind hierbei der Kostenaufwand der zusätzlichen Ethernet-Hardware und der Wegfall der Robustheitsvorteile einer strukturierten Verkabelung. In absehbarer Zeit werden Lichtsteuersysteme jedoch aus Mischformen von DMX512 und Ethernet-Netzen bestehen.

Wireless Access Points

Drahtlose Netzwerkübertragungen haben eine große Bedeutung innerhalb moderner Netzwerkinfrastrukturen. Das Verbindungsstück zwischen dem kabelgebundenen Ethernet-Netzwerk und einer drahtlosen Übertragung⁶⁸ bilden Wireless Access Points. Funkübertragungen sind nach dem Standard IEEE-802.11 spezifiziert. Diese sind entweder als einzelne Module erhältlich oder bereits in einem Router integriert. Jene Art der Datenübertragung ist für die Steuerung von Lichttechnik besonders interessant, da kein Verkabelungsaufwand nötig und die Steuerung örtlich flexibel ist. Die zu steuernden Geräte benötigen nur ein eigenes Funkmodul, um mit Steuerinformationen versorgt zu werden. Der Grund, warum der Einsatz von Wireless-Technik, heutzutage wenig Praxisrelevanz hat, ist die Zuverlässigkeit dieser Systeme. Im Gegensatz zu Kabelnetzen, arbeiten drahtlose Systeme unzuverlässiger und sind rauschbelastet. Dies ist auf Interferenzen mit anderen Gerätearten, welche im selben Frequenzband funken, zurückzuführen.

Die Zuverlässigkeit ist jedoch der wichtigste Punkt innerhalb eines Steuersystems. Darauf ist die Tatsache zurückzuführen, dass Wireless-DMX-Systeme, in der Praxis nur ein Schattendasein, neben der konventionellen Steuerung, für Spezialanwendungen führen. Ein interessanter Anwendungsbereich ist die mobile Steuerung für kleinere oder weniger aufwendige Veranstaltungen, bspw. Empfänge, Galaveranstaltungen und dergleichen. Wireless-LAN ermöglicht die Steuerung via mobiler Endgeräte, also Tablets und Smartphones, über spezielle Zusatzprogramme. So ist es möglich, bei ausreichender WIFI-Infrastruktur, von jedem Ort der Veranstaltungsstätte Steuersignale an die Scheinwerfer zu versenden, um bspw. das Farbschema einer Veranstaltung anzupassen.

⁶⁸diese wird auch als Wireless Lan (WLAN) bezeichnet

3.2 Ethernetbasierte Lichtsteuerprotokolle

Die evolutionäre Weiterentwicklung von Steuertechnik, zeichnet sich auch im Bereich der Lichtsteuerprotokolle ab. Da der DMX512-Standard allein, nicht mehr allen Anforderungen gerecht werden konnte, entwickelten sich weitere Steuerprotokolle auf Basis der Ethernet-Technologie.

Bei der Definierung eines herstellerübergreifenden Standards wurden dabei jedoch in Bezug auf Kompatibilität die gleichen Fehler wie bei der damaligen Umstellung von analoger auf digitaler Steuerung wiederholt.

Ein Artikel in der April-Ausgabe 2013 der deutschen Fachzeitschrift “Production Partner“ beschreibt dies so:

So ist es auch nicht verwunderlich, warum einige Hersteller an ihren Netzwerken festhalten, um Alleinstellungsmerkmale für ihre Systeme anbieten zu können. Vom Prinzip her wiederholt sich diese Einstellung und Vorgehensweise bei der Umsetzung von Architecture for Control Networks (ACN) wie damals bei der Einführung von DMX, als jeder Hersteller sein eigenes Protokoll nur im Detail anders als DMX zusammensetzte und dieses dann PMX, VMX, AD62,5, SMX oder Lee Coltran nannte. (Herbert Bernstädt, Production Partner 4/2013)

Dieser proprietäre Ansatz führte dazu, dass Hersteller ihre eigenen Protokolle definierten, welche jedoch nicht untereinander kompatibel sind.

In den letzten Jahren zeichnen sich jedoch, wie damals bei DMX, herstellerübergreifende Standards ab, welche im Bereich der Lichttechnik aufgrund der Vielzahl an Geräten besondere Relevanz haben. Die folgenden Abschnitte geben einen Überblick über die momentan wichtigsten Standards im Bereich der Ethernet-Lichtsteuerung.

3.2.1 ArtNet

Der erste frei zugängliche Standard, war das von dem britischen Unternehmen “Artistic License Ltd.“ entwickelte ArtNet Protokoll. Die technischen Spezifikationen wurden dabei öffentlich zur Verfügung gestellt um den Herstellern eine Implementierung des Protokolls zu ermöglichen. Dies ist ein notwendiger Schritt für eine Kompatibilität der Systeme verschiedener Hersteller. Alle Entscheidungen über die Entwicklung werden jedoch von Artistic License getroffen. Es ist daher dennoch als proprietärer, also herstellergebundener Standard einzuführen.

Die erste Version von ArtNet wurde bereits 1998 entwickelt und Anfang der 2000er Jahre veröffentlicht⁶⁹. Die Kommunikation fand dabei noch ausschließlich über Broadcast-Nachrichten statt, was, in Kombination mit den damals verwendeten 10BaseT-Netzen, die maximale Anzahl an Kanälen stark einschränkte⁷⁰.

⁶⁹Vgl. http://www.artisticlicencintegration.com/?page_id=931 Stand: 30.06.2013

⁷⁰Vgl. http://www.artisticlicencintegration.com/?page_id=977 Stand: 30.06.2013

Aufgrund des wachsenden Bedarfs an DMX-Universen wurde es nötig das Protokoll weiterzuentwickeln und auch Unicast-Kommunikation zu unterstützen. Die 2006 veröffentlichte Version 2 des ArtNet-Standards ermöglichte neben Broadcasting erstmalig auch das Unicasting im Ethernet. Durch diese Anpassung ermöglicht ArtNet 2 das Adressieren von bis zu 256 Universen. Um eine erneute Erhöhung der maximal möglichen Kanalzahl zu erreichen, wurde 2011, die momentan aktuelle Revision 3 veröffentlicht. Jene hebt das Adressierungs-Limit auf 32768 Universen⁷¹. Dies ist jedoch ein theoretischer Wert. Der tatsächlich erreichbare Wert ist von der verwendeten Übertragungsvariante sowie der Ethernet-Infrastruktur abhängig.

Funktionsweise ArtNet

Das ArtNet Protokoll unterstützt die Übertragung von DMX512- und RDM-Signalen über das Versenden von UDP-Paketen. Die Übertragung erfolgt, wie von DMX512 bekannt, in Datenpaketen mit jeweils 512 Kanälen. Ein ArtNet-Netzwerk besteht aus dem Zusammenschluss von Controllern (bspw. ein Steuerpult, generiert und sendet ArtNet-Pakete) und Nodes (bspw. ein Ethernet-DMX Wandler, empfängt ArtNet-Pakete).

Der Controller verwendet zum Erkennen aller im Netzwerk befindlichen, ArtNet-fähigen Nodes, Medienserver und Controller einen sog. "ArtPoll"⁷². Dies ist ein im Standard definiertes Broadcast-Paket, welches von allen ArtNet-Teilnehmern nach dem Empfang mit einem "ArtPollReply", ebenfalls ein Broadcast, beantwortet wird. Durch diese Antwort, wird dem Controller die Art und Funktion jedes Geräts mitgeteilt. Der ArtPollReply wird zusätzlich bei jedem Start eines ArtNet-Geräts automatisch versendet um die "Ankunft" des Geräts im Netzwerk zu melden⁷³.

Nachdem alle Geräte im Netzwerk erkannt sind, kann die Übermittlung der Daten erfolgen. Diese werden über sog. "ArtDmx"-Pakete im Uni- oder Broadcast übertragen, wobei Unicast die bevorzugte Variante ist⁷⁴. Der Aufbau eines solchen Pakets ist in Abb. 7 ersichtlich.

Byte-Versatz	0	1	2	3
0	'A'	'r'	't'	'.'
4	'N'	'e'	't'	0
8	Opcode ArtDMX (0x5000)		Protokoll-Version (14)	
12	Sequence	Physical	Universum	
16	Länge (2 bis 512, gerade)		Daten	Daten
20	Daten ...			

Abbildung 7: Aufbau eines ArtDMX-Pakets
Quelle: <https://en.wikipedia.org/wiki/Art-Net>

⁷¹Vgl. http://www.artisticlicencelntegration.com/?page_id=977 Stand: 30.06.2013

⁷²Vgl. [Art12] S. 5

⁷³Vgl. [Art12] S. 8

⁷⁴Vgl. [Art12] S. 20

Der orange markierte Bereich kennzeichnet unveränderliche Werte des ArtDMX-Pakets. Veränderliche Paketbestandteile sind gelb und grün hinterlegt.

Jedes Paket startet mit einer ID, welche von einem Nullbyte terminiert wird. Der darauf folgende OpCode gibt die Art des ArtNet-Paketes an (hier: ArtDmx). Den Abschluss des unveränderlichen Teils des Pakets, bildet die Nummer der Protokollversion (hier: 14). Das “Sequence“-Feld wird genutzt, um sicherzustellen das ArtNet-Pakete in der richtigen Reihenfolge empfangen werden. Dazu wird dieses Feld beim Versand von Paketen zwischen den Werten 0x01 und 0xff inkrementiert, was dem Empfänger eine Restrukturierung der Pakete ermöglicht⁷⁵. Der mit “Physical“ gekennzeichnete Bereich, ist ein Informationsfeld zur Darstellung des physischen DMX512 Input-Ports. Darauf folgt die im ArtNet 3 Protokoll 15 Byte lange Port-Adresse⁷⁶. Das nachfolgende Feld enthält eine gerade Ganzzahl, welche die Länge des DMX512-Arrays angibt. Die folgenden Bytes enthalten die Werte des übertragenen DMX512-Universums⁷⁷.

ArtNet unterstützt neben der einfachen Übertragung von DMX512-Werten zahlreiche Zusatzfunktionen. Besonders relevante Zusatzfunktionen sind bspw. die Übertragung von RDM-Nachrichten, die Time-Code Synchronisation, das Merging zweier Eingangssignale und das Einspielen von Firmware-Updates.

Generell gilt es zu beachten, dass ein Node nach der Protokolldefinition nicht mehr als vier Ports (Ein- oder Ausgänge) aufweisen darf. In der Beschreibung des ArtPollReply wird festgelegt, dass der Wert der Portzahl maximal vier betragen darf⁷⁸. Das in Kapitel 7.2 beschriebene ArtNet Plugin der Open Lighting Architecture hält sich strikt an diese Definitionen und ermöglicht somit nur vier Ein- und Ausgangsports. Will man in einem Produkt abweichend vom Standard mehr Ports definieren, ist es möglich einem Gerät mehrere virtuelle IP-Adressen zu vergeben und somit mehrere ArtPollReplies zu versenden⁷⁹.

Aufgrund der weitreichenden Funktionalitäten des ArtNet-Protokolls hat sich dieses bereits als “Quasi-Standard“ in der Beleuchtungsindustrie etabliert und wird von zahlreichen Herstellern unterstützt.

3.2.2 Architecture for Control Networks - ACN und sACN

Der zweite wichtige Standard in Bezug auf herstellerübergreifende Lichtsteuerung ist das auf der Architecture for Control Networks (ACN) aufbauende sACN Protokoll. ACN ist ein von der PLASA veröffentlichter ANSI-Standard, welcher 17 Netzwerkprotokolle für den Bereich der Veranstaltungstechnik definiert⁸⁰. Besondere Bedeutung hat dieser, da er von einer internationalen Normungsorganisation, der PLASA, herausgegeben wird.

⁷⁵Vgl. [Art12] S. 21

⁷⁶in Abb. 7 ist ein ArtNet2 Paket abgebildet, dieses hat hier nur eine Länge von 8 Bit

⁷⁷Vgl. [Art12] S. 21

⁷⁸Vgl. [Art12] S. 10

⁷⁹Vgl. <http://www.artisticlicence.com/support/index.php?/Knowledgebase/Article/View/3/0/art-net-nodes-with-more-than-four-ports> Stand: 30.06.2013

⁸⁰Vgl. http://tsp.plasa.org/tsp/documents/published_docs.php Stand: 30.06.2013

Die Bezeichnung des ACN-Standards ist ANSI E1.17 - 2010. Jene ACN-Protokoll-Suite ist, genau wie ArtNet, ebenfalls auf UDP basierend. Sie hat jedoch den großen Unterschied, nicht nur den Bereich der Lichttechnik abzudecken, sondern ist deutlich abstrakter um bspw. auch die Steuerung in Audionetzwerken zu ermöglichen.

Ein speziell auf die Übertragung von DMX512-Signalen über die ACN-Protokolle angepasster Standard ist sACN (ANSI E1.31⁸¹). Dieser nutzt nur einen kleinen Teilbereich der im ACN-Standard definierten Protokolle. Die Funktionsweise von sACN wird im folgenden Abschnitt des Kapitels näher erläutert.

Funktionsweise sACN

Die generelle Funktionsweise von sACN basiert, ähnlich wie bei ArtNet, darauf die konventionellen DMX512-Frames in eine weitere Paketebene eines Ethernet-Frames zu "verpacken". Diese Pakete werden dann nacheinander wiederholt übertragen. Die für sACN benutzte Paketebene ist gemäß des ACN-Standards aufgebaut⁸². Die für sACN relevanten Protokolle aus der ACN-Protokollsuite sind vor allem das "Device Management Protocol" (DMP), das "Session Data Transport Protocol" (SDT) sowie das "Root Layer Protocol" (RLP).

Ein grundlegender Unterschied von sACN zu ArtNet ist dagegen die standardmäßige Verwendung von Multicast. Neben Multicast sieht sACN des Weiteren die Verwendung von Unicast-Verbindungen vor⁸³.

Jedes übertrage sACN-Paket besteht aus drei Teilebenen. Dem "Root Layer", dem "E1.31 Framing Layer" und dem "DMP-Layer"⁸⁴. Nachfolgend werden die relevanten Inhalte der jeweiligen Paketebenen dargestellt⁸⁵. Die unterste Paketebene bildet das RLP. Diese definiert u.a. die gesendeten Daten als ACN-Paket und gibt über eine UID⁸⁶ Auskunft über den Sender⁸⁷.

Der darauf folgende Bereich des Pakets ist der E1.31 Framing Layer. Er enthält, neben anderen Informationen, den vom Benutzer vergebenen Quellennamen, eine mit 16 Bit adressierbare Universums-Nummerierung, eine, wie von ArtNet bekannte, Sequenznummerierung zur Erkennung der korrekten Reihenfolge der Pakete und Prioritätsinformationen⁸⁸. Diese können einen Wert zwischen 0-200 annehmen, welcher als Grundlage zum Merging verschiedener Quellen dient⁸⁹. Dies ist ein Vorteil gegenüber dem ArtNet-Standard, der diese Funktionalität nicht bietet.

Den Abschluss eines sACN-Paketes bildet der DMP-Layer. Dieser beinhaltet die Anzahl

⁸¹volle Bezeichnung: ANSI E1.31 - 2009 - Lightweight streaming protocol for transport of DMX512 using ACN

⁸²Vgl. [ANS09] S. 1

⁸³Vgl. [ANS09] S. 1

⁸⁴Vgl. [ANS09] S. 4

⁸⁵für eine Komplettbeschreibung der Paketform siehe [ANS09]

⁸⁶Unique Identifier

⁸⁷Vgl. [ANS09] S. 5

⁸⁸Vgl. [ANS09] S. 6

⁸⁹für Details zum sACN Merge innerhalb der Open Lighting Architecture siehe Kap. 7.2

der verwendeten DMX-Kanäle sowie die eigentlichen DMX512-Daten⁹⁰. Der Standard sieht dabei nicht das Senden von RDM-Signalen, also alternativen Start-Codes vor, sondern definiert, dass der DMX512 Start-Code benutzt werden soll. Ein Senden von RDM Daten über sACN ist somit nicht möglich. Eine Erweiterung die dieser Problematik Abhilfe schafft wird im zukünftigen Standard ANSI E1.33 definiert. Dieser befindet sich aktuell jedoch noch in der Revisionierung und ist noch nicht veröffentlicht.

Dies ist aktuell noch ein deutlicher Vorteil in der Nutzung von ArtNet in Installationen mit RDM-fähigen Geräten. Zudem zeigt dies erneut, welche lange Zeiträume zur Definierung von Standards in der Beleuchtungsindustrie benötigt werden.

Die Entscheidung zwischen ArtNet und sACN hängt demnach ebenfalls davon ab, welche Funktionen für den Endanwender hohe Relevanz besitzen.

3.2.3 Weitere Steuerprotokolle

Neben dem herstellerübergreifenden Ansatz bei der Definierung von Protokollen, ist, wie in Kap. 3.2 bereits angedeutet, eine Verbreitung von proprietären Standards am Markt erkennbar.

Eine Teilübersicht einiger Lichtsteuerprotokolle und der dazugehörigen Hersteller ist in Tabelle 4 ersichtlich.

Hersteller	Protokoll
MA Lighting	MA-Net/MA-Net2
ETC	ETCNet1/ETCNet2/ETCNet3
Enttec	ESP Net
Pathway Connectivity	Pathport
Strand Lighting	ShowNet
SAND Network Systems	SandNet
Color Kinetics	KiNet

Tabelle 4: Übersicht proprietärer Lichtsteuerprotokolle

Zur Erweiterung der Steuermöglichkeiten einer Lichtstallanlage kommt zudem das OSC-Protokoll zum Einsatz. Dies ist ein nachrichtenbasiertes Kommunikationsprotokoll, welches ursprünglich zur Echtzeitverarbeitung von Sound über Netzwerke konzipiert wurde⁹¹. Zusätzliche Anwendungen, wie bspw. TouchOSC⁹², ermöglichen das einfache Erstellen von grafischen Multitouch-Oberflächen, welche dann zur Steuerung einer Lichtanlage über Tablet und Mobiltelefon benutzt werden können.

⁹⁰Vgl. [ANS09] S. 10

⁹¹Vgl. https://de.wikipedia.org/wiki/Open_Sound_Control Stand: 01.07.2013

⁹²siehe <http://hexler.net/software/touchosc>

4 Marktübersicht Ethernethfähige Lichttechnik

Da die Ethernet-Technik, bereits seit mehreren Jahren Einzug in die lichttechnischen Anwendungen der professionellen Veranstaltungstechnik gehalten hat, ergibt sich hier ein breites Spektrum der verfügbaren Produkte. Der lange Zeitraum der Normenfindung eines branchenübergreifenden Standards (sACN) hat jedoch zu vielen proprietären Systemkomponenten geführt, welche, je nach Marktsegment, teilweise nicht mit anderen Herstellern kompatibel sind.

Im Folgenden werden die Marktbereiche Lichtsteuerpulte, Scheinwerfertechnik und Ethernet-Nodes analysiert. Besonderes Augenmerk liegt bei dieser Analyse auf dem gebotenen Funktionsumfang, dem Preis und der herstellerübergreifenden Kompatibilität und somit der Unterstützung für verschiedene Ethernet-Protokolle, der Produkte.

4.1 Lichtsteuerpulte

Bei Lichtsteuerpulten zeigt sich, besonders im gehobenen Marktsegment, ein sehr positives Bild in Bezug auf Herstellerübergreifende Kompatibilität. Die Pulte aller namhafter Hersteller (MA Lighting, ETC, High End Systems, etc.) unterstützen eine Vielzahl an Protokollstandards. Dies sind neben den herstellereigenen Protokollen wie ETC Net und MA-Net, die herstellerübergreifenden Standards ArtNet und sACN.

Die Abgrenzungsmerkmale der proprietären Protokolle liegen oftmals in speziell aufeinander abgestimmten Produktgruppen der Hersteller, welche einige Zusatzfunktionen, bspw. die Fernkonfiguration oder ein zentrales Update der Geräte, nur über das "hauseigene" Protokoll unterstützen. Ein Beispiel dafür sind die bekannte grandMA 2 von MA Lighting in Kombination mit einem MA Lighting 4/8 Port Node (genaueres zum Gerät siehe Abb. 9). Für die Konfiguration der DMX-Ports des Nodes wird eine grandMA 2 Konsole oder ein GrandMA on PC-System benötigt, was die einfache Integrierung dieses Geräts in ein System eines anderen Herstellers erschwert⁹³.

Dank breiter Unterstützung für verschiedene Protokolle, bleibt die Wahl beim Endanwender, ob er die geschlossenen Systeme eines Herstellers nutzen oder Geräte unterschiedlicher Anbieter kombinieren will. Dies ermöglicht maßgeschneiderte Lösungen, welche auf die Bedürfnisse der Kunden angepasst werden können.

Einige der Funktionen werden in modernen Lichtsteuernetzwerken in die Netzwerk-Knoten ausgelagert. Dies sind vorrangig Berechnungen bei der Wandlung und Ausgabe der Steuersignale, jedoch auch komplexere Aufgaben, wie das Show-Backup während eines Fehlerfalls.

Der Ethernet-Node stellt damit neben dem Lichtstelltppult einen elementaren Bestandteil des modernen Lichtsteuernetzes dar.

⁹³Vgl. http://www.lightpower-files.de/Assets/MA_4_Port_Node_Quickstart_Manual_2013_Version_2_de_en.pdf Stand: 23.06.2013

4.2 Scheinwerfertechnik

Aufgrund der in Kap. 3.1.5 beschrieben Problematik der Sternverkabelungen in Lichtsteuernetzwerken, hat sich der Einsatz von Ethernet-Technik in den Scheinwerfern nur in sehr geringem Maße etabliert. Ob ein Einsatz eines Netzwerkanschlusses und den damit einhergehenden Funktionen sinnvoll ist, hängt vorrangig von der Komplexität des Geräts ab.

Generell ist eine Ethernet-Schnittstelle mit steigender Komplexität des Geräts und der damit einhergehenden steigenden Anzahl an DMX-Kanälen zweckmäßig. Die Verschmelzung von Licht und Video beschleunigt diese Wandlung zunehmends.

Ein Beispiel dafür ist der DigitalSpot 3500 DT des Unternehmens ROBE lighting. Dieses Gerät ist eine Mischung aus einem multifunktionellen Movinghead und einem Videoprojektor. Es benötigt max. 136 DMX-Kanäle und bietet auch die Möglichkeit des direkten Einbindens in ein Ethernet-System durch das ArtNet Protokoll. Über diese Schnittstelle wird zusätzlich ein Webinterface zur Konfiguration der Medieninhalte und anderer Einstellungen bereit gestellt⁹⁴. Die Produktreihe dieses Herstellers bietet nahezu durchgehend direkte Unterstützung für Ethernet- und sogar Wireless-Steuersysteme⁹⁵.

Diese breite Unterstützung stellt jedoch eher die Ausnahme als die Regel dar. Der Großteil der Hersteller stattet seine neuen Geräte zwar mit RDM- jedoch nicht mit direkten Ethernet-Kompatibilitäten aus. Dies ist jedoch nicht nur dem erhöhten technischen Aufwand, sondern viel mehr den aktuellen Topologien der Lichtsteuer-Netzwerke geschuldet. In absehbarer Zeit, wird hier weiterhin der Einsatz von klassischen DMX512-Netzen, vermutlich mit zunehmender RDM-Funktionalität, in Kombination mit Ethernet-Systemen dominieren. Diese hybriden Systeme mit Ethernet-Teilstrecken und Wandler-Nodes werden den Markt vermutlich noch lange Zeit dominieren.

4.3 Ethernet-Nodes

Auch bei Ethernet-Nodes, also den Netzwerkknoten, welche für die Wandlung von Ethernetprotokollen auf DMX512 verwendet werden, ergibt sich bereits eine große Marktvielfalt. Darum wird sich innerhalb dieser Analyse auf Produkte mit ähnlichem Funktionsumfang, wie des in dieser Arbeit vorgestellten Protokollwandlers, beschränkt.

Der häufigste Anwendungsfall eines Ethernet-Nodes, ist der Empfang bzw. die Konvertierung von Ethernet-Lichtsteuerprotokollen und der Wandel jener auf konventionelle DMX512 Signale. Daraus resultieren als wichtigste Kriterien bei der Auswahl von Komponenten, die Anzahl der Aus- und Eingänge sowie die Anzahl der unterstützten Ethernet-Protokolle. Weitere relevante Produktmerkmale sind bspw. implementierte Zusatzfunktionen, wie die des Mergings, digitaler/analoger Aus- und Eingänge, RDM-Unterstützung, das Abspielen gespeicherter Steuerdaten und die Art der Konfiguration

⁹⁴Vgl. <http://www.robe.cz/products/article/digitalspot-3500-dt/#specifications> Stand: 21.06.2013

⁹⁵Vgl. http://www.robe.cz/fileadmin/robe/downloads/catalogues/ROBE_Product_Guide_april_2013.pdf Stand: 21.06.2013

des Geräts.

Eine Übersicht, fünf aktuell am Markt erhältlicher Systeme, befindet sich in den Abbildungen 8 und 9. Dort zeigt sich, dass je nach Gerät die Unterstützung von Protokollen sehr unterschiedlich ist.

Die von ETC vertriebene transtechnik E-Gate Reihe bietet bspw. keine Unterstützung des ArtNet-Protokolls⁹⁶ ⁹⁷, was von allen anderen untersuchten Geräten unterstützt wird. ArtNet hat sich abgesehen von diesen Ausnahmen als quasi-Branchenstandard etabliert. Ein besonders positiv zu wertender Fakt ist, dass abgesehen von der CueCore Serie von Visual productions alle betrachteten Produkte bereits den herstellerübergreifenden Standard sACN unterstützen. Diese Entwicklung ist vor allem in Bezug auf Kompatibilität zukünftiger Lichtsteuersysteme notwendig und erstrebenswert.

	transtechnik E-Gate pro	ELC dmXLAN Node 3TM	Enttec Datagate MK2
Abmessungen	19Zoll, 2HE, Bautiefe 300mm	140x87x52 mm (BxTxH)	19Zoll, 1HE
Anzahl und Art der Ein- und Ausgänge	6x DMX / 2x IN, 4x Out 1x RJ45	3x DMX frei konfigurierbar 2x EtherCon	8x DMX, frei konfigurierbar 1x RJ45
digitale und analoge Aus -und Eingänge	12x Analog In 0-10V 6x Digital IN	-	-
unterstützte Netzwerkprotokolle	AVAB/UDP (IN/OUT), AVAB/IPX (IN/OUT), sACN (nur IN)	sACN, ArtNet	ArtNet 1&2, ESP, kiNet, sACN, OSC
RDM-Unterstützung	-	Ja	Ja
Speichern und Abspielen von Signalen	Nein, nur halten des Wertes bei Signalausfall	Nein, nur halten des Wertes bei Signalausfall	Ja
Merging-Implementat ionen	HTP	HTP, LTP, Prioritäten	HTP, LTP, Prioritäten
Zusatzfunktionen	Proxy für Dimmerrückmeldung	Konfiguration über lokales Display oder ELC dmXLAN Software	Konfiguration über Webinterface
	Bedienung über E-Gatekeeper Software	zusätzliche Befestigungen für Truss-Montage	1x RS232 DMX-Parameter einstellbar
Anmerkungen	Input: 2x DMX, 4x Ethernet (max. 4 gleichzeitig); Output: max. 4x DMX 2x Ethernet	integrierter Switch verbindet EtherCon Anschlüsse	ungeöhnlicher Lizenzkauf um Softwarebestandteile freizuschalten
Preis (UVP) exkl. MwSt.	3.350,-- €	829,-- €	4271,-- € inkl. aller Lizenzen

Abbildung 8: Übersichtstabelle Ethernet-DMX Nodes Teil 1

Eine besonders umfangreiche Protokollunterstützung bietet der Enttec Datagate MK2 (siehe Abb. 8). Diese muss jedoch teilweise erst über zusätzliche Softwarelizenzen freigeschaltet werden. Diese restriktive Verkaufsstrategie führt zu einer vergleichsweise günstigen Einstiegsvariante des Geräts von 1425,- €, jedoch auch zu teilweise sehr teuren Zusatzmodulen. Zum Auslesen von Sensordaten über die DMX-Erweiterung RDM, muss

⁹⁶Vgl. http://www.etcconnect.com/docs/docs_downloads/manuals/EGate_pro_R300_d.pdf
Stand: 21.06.2013

⁹⁷dies gilt auch für den direkt von ETC produzierten Net3 Four Port Gateway

bspw. eine Zusatzlizenz für 899,- € erworben werden⁹⁸. Der Preis des Geräts summiert sich somit auf bis zu 4271,- € (siehe Abb. 8).

Die Unterstützung von digitalen und analogen Ein- und Ausgängen ist stark abhängig von der angesprochenen Zielgruppe der Geräte, da diese Funktionen vor allem im Bereich der Festinstallationen, interactive Media, Architekturbeleuchtung und Kunstillustrationen relevant sind. Geräte, welche für den reinen Show- und Theaterbetrieb konstruiert wurden, bspw. der MA-Lighting 4Port Node oder der ELC dmXLAN Node 3TM, bieten meist keine Unterstützung für derlei Funktionen.

Weitreichende Funktionalitäten für kreative Anwendungen und Festinstallationen bietet der "CueCore" von Visual Productions. Dieser ist modular erweiterbar durch das Zusatzgerät IoCore⁹⁹ und bietet dadurch acht zusätzliche Inputs/Outputs (wahlweise analog oder digital¹⁰⁰). Das Zusatzmodul IoCore besitzt eine UVP exkl. MwSt. von 599,- € .

	MA Lighting 4Port Node	Visual productions CueCore	DMX-Protokollwandler
Abmessungen	9Zoll, 1HE, Bautiefe 184mm	170x100x28 mm (BxTxH)	200x180x94 mm (BxTxH)
Anzahl und Art der Ein- und Ausgänge	4x DMX frei konfigurierbar 1x EtherCon	3x DMX / 2x OUT, 1x IN 1x EtherCon	4x DMX / 2x OUT, 2x IN 3x EtherCon
digitale und analoge Aus -und Eingänge	-	4x Digital IN	8x Analog IN 16x PWM OUT
unterstützte Netzwerkprotokolle	MA-Net2, sACN, Art-Net 3	ArtNet, OSC, MIDI	ArtNet 1&2&3, sACN, ESP, Shownet, Pathport, kiNet, OSC
RDM-Unterstützung	Ja	Ja	nur durch Anschluss zusätzlicher Hardware
Speichern und Abspielen von Signalen	-	Ja, sehr umfangreicher Funktionsumfang	Ja
Merging-Implementationen	HTP, LTP, LoTP (nur für MA-Net2)	HTP, LTP	HTP, LTP, Prioritäten
Zusatzfunktionen	Konfiguration über grandMA2 oder grandMA onPC	Konfiguration über Webinterface und vManager-Software	modulare Erweiterung durch USB-Geräte
	integrierte PSU	Time-code Unterstützung	integr. Wifi-Accesspoint Konfiguration über Webinterface, oder angeschlossenens HDMI Display/USB-Tastatur
Anmerkungen	zusätzliche Befestigung für Rack- und Rigging-befestigung erhältlich	erweiterbar um zusätzliche I/Os durch IoCore	Open Source Hardware- und Softwarebestandteile
Preis (UVP) exkl. MwSt.	ca. 2500,-- €	949,-- €	210,-- € reine Bauteilkosten

Abbildung 9: Übersichtstabelle Ethernet-DMX Nodes Teil 2

In Bezug auf den unterstützten Funktionsumfang ist erkennbar, dass die Open Lighting Architecture (Basis des hier vorgestellten Protokollwandlers, siehe Kap. 7.2), in nahezu allen Bereichen mit den am Markt befindlichen Systemen konkurrieren kann,

⁹⁸Vgl. http://www.enttec.com/?main_menu=Products&pn=70044 Stand: 21.06.2013

⁹⁹Vgl. <http://www.visualproductions.nl/products/cuecore.html> Stand: 21.06.2013

¹⁰⁰nur analoger Input möglich, kein Output eines Analogsignals

bzw. ihnen teilweise überlegen ist. Dies zeigt sich vor allem in den Bereichen der unterstützten Protokolle und Merging Funktionen. Weitere Zusatzfunktionen, bspw. die Recording-Funktion und die Konfiguration der digitalen/analogen I/Os, sind jedoch nicht so komfortabel zu bedienen, sondern benötigen teilweise einige Kenntnisse mit der Bedienung der Kommandozeile, was die Bedienung durch den Endanwender deutlich erschwert. Hier könnten zukünftige Entwicklungen der Software weitere Funktionen in das Webinterface integrieren, was die Bedienung erheblich erleichtern würde.

5 Aufbau des Ethernet-DMX-Wandlers

Der aufwendigste Bestandteil während der Entwicklung dieses Protokollwandlers, ist die Kombination von Hard- und Software. Die Fragestellungen dabei lauten: Wo liegen die definierten Schnittstellen, mit denen Computer und Peripherie verbunden werden? Welche Module werden zur Erweiterung benötigt? Wie werden diese über die Software angesteuert und wie werden die Schnittstellen physisch für den Anwender zur Verfügung gestellt?

In den folgenden Kapiteln wird die Funktionsweise der, für den Protokollandler relevanten, Module näher betrachtet und analysiert.

5.1 Hardware- und Funktionsübersicht

Die verwendete Hardware des DMX-Protokollwandlers besteht aus einem embedded Computer, zwei USB-DMX Interfaces mit jeweils einem DMX512 Aus- und Eingang, einem Router zur Erweiterung der Netzwerkfunktionen um WLAN und drei herausgeführte EtherCon-Anschlüsse, einem 16 kanaligem PWM-Ausgang, acht analogen Eingängen und einem 16x2 Zeilen LCD (das Display ist in Abb. 10 ersichtlich).



Abbildung 10: Frontansicht DMX-Protokollandler - Nahaufnahme Display

Diese Module ermöglichen das gleichzeitige Nutzen von zwei DMX-Inputs zum Wandeln auf zahlreiche Ethernet-Protokolle¹⁰¹ sowie die Ausgabe von DMX512-Signalen auf den zwei DMX-Outputs.

¹⁰¹eine detaillierte Funktionsbeschreibung der Software findet sich in Kapitel 7

Durch die analogen Eingänge ist der Anschluss von diversen Sensoren¹⁰², deren Ausgabe variierende Spannungspegel sind, und den Signalen analoger Lichtsteuerungen möglich¹⁰³.

Anzahl	Anschluss	Funktion
2	5pol. XLR Female	DMX Out
2	5pol. XLR Male	DMX In
3	EtherCON	100MBit Ethernet
1	HDMI	Anschluss eines externen Bildschirms
1	USB 2.0 Port	vollwertiger USB-2.0 Anschluss
1	8-pol. Buchsenleiste 2.54mm	achtkanaliger analoger Input
1	16-pol. Buchsenleiste 2.54mm	16-kanaliger PWM Ausgang
1	PowerCON	Anschluss 230V AC

Tabelle 5: Übersicht der verfügbaren Anschlüsse und deren Funktion

Der PWM-IC stellt insgesamt 16 Kanäle zum direkten Ansteuern von LED-Modulen, Servomotoren und anderer, per PWM steuerbarer, Elektronik bereit. Die Funktionserweiterung der digitalen Aus- und Eingänge ist ausgerichtet auf die Anwendung in Festinstallationen, bspw. Museen oder Ausstellungen, oder in Kunstinstitutionen bzw. dem Einsatz interaktiver Medien.

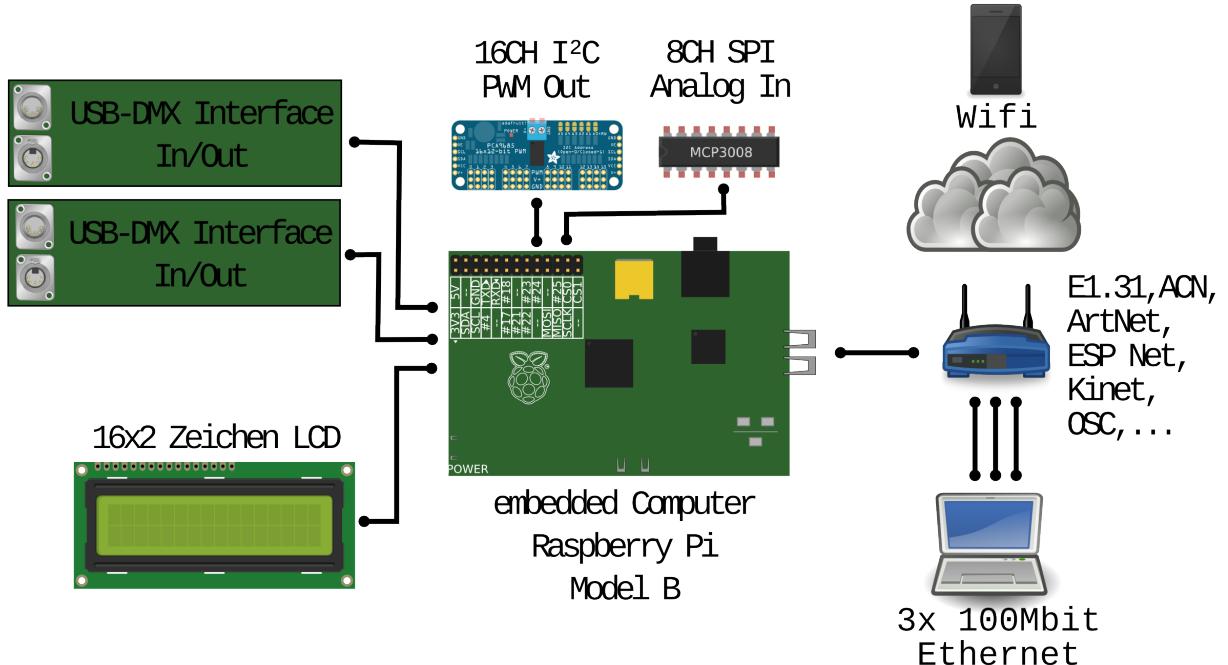


Abbildung 11: Schema Funktionsübersicht^a

^aGrafik basierend auf Symbolen von <http://www.opensecurityarchitecture.org/cms/library/icon-library> und <https://github.com/adafruit/Fritzing-Library> Stand: 04.06.2013

¹⁰²dies sind bspw. Helligkeitssensoren, Temperatursensoren, etc.

¹⁰³da diese meist mit einem 0-10V Pegel arbeiten ist dafür ein zusätzlicher Spannungsteiler nötig

Die gesamte Peripherie befindet sich in einem robusten Gehäuse. Intern werden die einzelnen Module auf einer zusätzlichen Platine mit dem eingebetteten Computer verbunden.

Alle Schnittstellen sind auf langlebigen und branchenüblichen Steckverbinderystemen herausgeführt. Die vorhandenen Anschlüsse und deren Funktion sind in Tabelle 5 ersichtlich.

Eine schematische Übersicht der verwendeten Hardware ist in Abbildung 11 ersichtlich. Der Aufbau und die Funktionsweise der relevanten Module werden in den folgenden Kapiteln näher erläutert.

5.2 Embedded Hardware Plattform

Das Kernstück dieses Protokollwandlers bildet ein embedded Computer. Diese Kleinstcomputer bieten auf engem Raum, meist einer einzelnen Platine, zahlreiche Schnittstellen und Prozessoren. Die wichtigsten Kriterien für die Auswahl eines geeigneten, eingebetteten Computers für dieses Projekt waren die Leistungsfähigkeit der Prozessoren, die verfügbaren Schnittstellen, die Größe der Entwicklergemeinde¹⁰⁴ und die Anschaffungskosten.

Eine grundlegende Veränderung im Markt dieser Entwicklungsplattformen geschah 2012 durch die Markteinführung des Raspberry Pi. Ziel dieses Projekts ist die Konstruktion eines leistungsfähigen und kostengünstigen embedded PC für Bildungszwecke. Hinter der Konzeption des Raspberry Pi steht die Raspberry Pi Foundation, eine in Großbritannien eingetragene Wohltätigkeitsorganisation. Aktuell (Stand: 04.06.2013) sind bereits mehr als eine Million Raspberry Pi's verkauft.

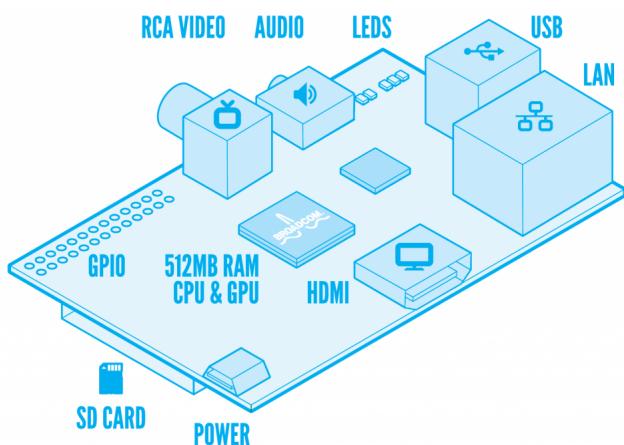


Abbildung 12: Schema Raspberry Pi Model B (Quelle: Paul Beech, Quelle: <http://www.raspberrypi.org/wp-content/uploads/2011/07/RaspiModelB-1024x902.png> Stand: 04.06.2013)

Der Raspberry Pi wird in zwei Varianten, Model A und B, angeboten. Gemein haben

¹⁰⁴ hiermit ist die Anzahl an aktiven Entwicklern, welche mit diesem System arbeiten gemeint

beide Varianten einen mit 700MHz getakteten ARM11-Prozessor¹⁰⁵, eine leistungsfähige Videocore 4 GPU, welche in der Lage ist 1080p30 H.264-kodierte Videos mit einer Bitrate von 40MBit/s abzuspielen, einen HDMI und S-Video Ausgang und einen GPIO¹⁰⁶-Header mit Unterstützung für SPI, I2C und UART. Model B ist mit 35 US\$ ca. 10 US\$ teurer als Model A, bietet jedoch auch zwei anstatt nur einem USB Anschluss, 512MB statt 256MB RAM und eine, auf dem USB 2.0 Anschluss aufbauende, Ethernet-Schnittstelle. Die Größe des kompletten Boards beträgt 85,6mm x 56mm x 21mm. In der Abbildung 12 ist der schematische Aufbau und die Platzierung der Anschlüsse zu erkennen.

Seit dem Verkaufsstart des Raspberry Pi, sind bereits eine Vielzahl vielversprechender Entwicklungsboards erschienen. Eine interessante Alternative ist das BeagleBone Black, welches sich in etwa im gleichen Preissegment befindet und ähnliche Leistungsdaten aufweist.

Für dieses Projekt ist, meiner Meinung nach, momentan jedoch der Raspberry Pi die beste Entscheidung, da er explizit von der verwendeten Softwareplattform OLA unterstützt wird und er bereits eine sehr große Nutzerbasis erreicht hat. Somit ist sichergestellt, dass die Unterstützung für diese Hardwareplattform langfristig vorhanden ist und es gute Dokumentationen für die einzelnen Funktionen der Hardware gibt.

Die Ethernet-Schnittstelle des Raspberry Pi ist auf dem USB-2.0 Controller aufbauend. Dadurch teilt sich die USB-Peripherie die maximale Bandbreite des USB-Bus mit den angeschlossenen Ethernet-Geräten. Für die Anwendung des in dieser Arbeit vorgestellten Protokollwandlers hat dies jedoch keinen Einfluss, da die zwei angeschlossenen DMX-USB-Interfaces mit dem USB-1.1 Standard kommunizieren und somit jeweils nur eine maximale Datenrate von 12Mbit/s aufweisen können. Der Bus bietet somit genügend Bandbreite für eine Fast-Ethernet Schnittstelle mit 100Mbit/s.

Das verwendete Betriebssystem ist Raspbian. Dies ist ein, für den Raspberry Pi optimiertes, Debian Derivat. Ein vorinstalliertes Image, welches die für OLA-benötigten Pakete installiert hat, sowie alle Quelldateien werden durch die Entwickler der Open Lighting Architecture bereitgestellt¹⁰⁷. Eine SD-Karte wird mit den Daten des Betriebssystems beschrieben und dient als Festspeicher des Raspberry Pi. Dies ermöglicht einen einfachen Wechsel des Betriebssystems, was die Entwicklungsarbeit erheblich erleichtert.

5.3 Anschluss- und Erweiterungsboard

Ein Entwicklungsboard wie der Raspberry Pi ermöglicht es, ihn mit einer Vielzahl an Peripherie zu verbinden. Im Falle des Ethernet-DMX-Protokollwandlers werden diverse Zusatzmodule, bspw. zum Einlesen von analogen Signale und zur Ausgabe von PWM-

¹⁰⁵dies ist eine Prozessorarchitektur mit geringem Energiebedarf, meist eingesetzt für eingebettete Systeme und mobile Computer

¹⁰⁶GPIO steht für General purpose Input/Output und ermöglicht das anschließen zusätzlicher Peripherie

¹⁰⁷vorkonfigurierte Images verfügbar unter: <http://nomis52.net/data/raspi/> Stand: 12.06.2013

Signalen, benötigt. Um diesen Anschluss zusätzlicher Peripherie zu erleichtern, verwendet man Erweiterungsplatinen, sog. Daughterboards. Diese stellen zusätzliche Funktionen bereit und erleichtern den Anschluss der Zusatzmodule.

Es gibt bereits eine Vielzahl an Add-On Boards für den Raspberry Pi. Diese haben jedoch den Nachteil, dass sie teilweise teurer und größer sind, als der Raspberry Pi¹⁰⁸. Dies liegt meist in der Tatsache begründet, dass sie weitreichende spezialisierte Zusatzfunktionen ermöglichen. Das direkte Steuern von Motoren, D/A- und A/D-Wandler und die Verbindung mit einem zusätzlichen Mikrocontroller sind nur einige dieser Erweiterungen.

GPIO-Pin	Funktion
GPIO 2	SDA I2C PWM-Modul
GPIO 3	SCL I2C PWM-Modul
GPIO 4	CS MCP3008
GPIO 9	MISO MCP3008
GPIO 10	MOSI MCP3008
GPIO 11	SCLK MCP3008
GPIO 14	Shutdown Taster
GPIO 17	DB5 LCD
GPIO 22	DB7 LCD
GPIO 23	DB4 LCD
GPIO 24	Enable LCD
GPIO 25	RS LCD
GPIO 27	DB6 LCD

Tabelle 6: Belegung GPIO-Header (nach BCM-Schema)

Da viele dieser Funktionen für den Protokollwandler nicht benötigt werden und somit die Kosten und der Platzbedarf unnötig ansteigen würden, wurde im Rahmen dieser Arbeit eine, den Anforderungen angepasste, Erweiterungsplatine entworfen. Diese ergänzt den Raspberry Pi um folgende Module: einen über SPI¹⁰⁹ angeschlossenen, Digital/Analog-Wandler; einen, mit einem Pull-Down Widerstand versehenen Anschluss für einen Taster; separierte Stiftleisten-Anschlüsse für das verwendete I2C¹¹⁰ PWM-Modul, den Analogwandler und das 16x2 Zeichen HD44780-LCD. Ein Potentiometer ermöglicht zudem die Kontrasteinstellungen des Displays anzupassen.

Die zusätzlichen Anschlussleisten (siehe Abb. 13) gewährleisten einen einfachen Anschluss und Wechsel der Peripherie und verringern die Wahrscheinlichkeit einer falschen Verkabelung. Dies ist besonders hilfreich, da sonst jede Verbindung zum GPIO-Header des Raspberry Pi mit einzelnen Verbindungskabeln auf die Stiftleisten geführt werden muss. In Tabelle 6 sind die Funktionen jedes verwendeten GPIO Pins aufgelistet.

Der mit der Software EAGLE erstellte Schaltplan des Erweiterungsboards befindet sich in der im Anhang befindlichen Abbildung 29.

¹⁰⁸ein Erweiterungsboard auf welches diese Punkte zutreffen ist bspw. das Gertboard

¹⁰⁹Serial Peripheral Interface, ist ein synchrones, serielles Bus-System

¹¹⁰Inter-Integrated Circuit, ist ein serieller Datenbus

Das Boardlayout der Platine ist in der Abbildung 13 ersichtlich.

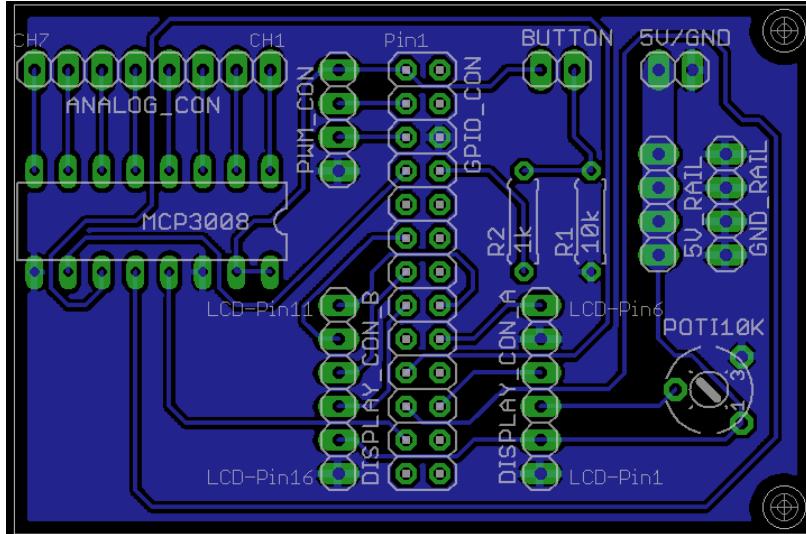


Abbildung 13: Boardlayout Anschluss- und Erweiterungsplatine

Die Größe des Add-On-Boards beträgt 60mm x 40mm. Da das Layout der Platine einseitig, also auf nur einer Seite mit Leiterbahnen versehen ist, ist die Fertigung auf einfache Art möglich. Somit kann auf eine industrielle Fertigung verzichtet werden. Das verwendete Verfahren zur Herstellung war die Fotopositiv-Methode, bei der eine fotosensitive Schicht auf der Platine aufgebracht ist, welche mit einer übergelegten Folienvorlage belichtet und anschließend entwickelt und geätzt wird.

5.4 DMX Out- und Input

Für die Ausgabe eines DMX-Signals reichen die vorhandenen Anschlussmöglichkeiten des Raspberry Pi nicht aus. Aus diesem Grund ist es notwendig, zusätzliche Hardware mit den Schnittstellen zu verbinden.

Da der Raspberry Pi über ein UART-Modul verfügt, war die erste Überlegung, dieses für den Versand von DMX-Frames zu benutzen. Ein UART ist die Grundlage für das Versenden und Empfangen von digitalen, seriellen Signalen innerhalb einer seriellen Schnittstelle. In Kombination mit einem geeigneten Bustreiber-IC sowie einer Schutzschaltung wäre die Ausgabe von DMX-Signalen möglich.

Das UART-Modul ist auf dem GPIO-Header auf Pin 14 und 15 herausgeführt. Standardmäßig wird dieses für die Debugging-Ausgabe und optional zum Einloggen während des Bootvorgangs benutzt. Bei der Kommunikation mit angeschlossener Peripherie wäre dies hinderlich und muss somit deaktiviert werden. Dies ist durch das Ändern einer Konfigurationsdatei möglich¹¹¹.

Ein großes Problem für das Versenden von DMX-Frames ist jedoch der Break am Anfang eines DMX-Signals. Wie in Kapitel 2.2 beschrieben, besteht ein DMX-Datenbyte jeweils

¹¹¹Vgl.: [rpi13]

aus einem Startbit, acht Datenbits und zwei Stopbits. Zu Beginn jedes DMX-Frames wird jedoch ein Break, welcher nur aus logischen Nullpegeln besteht, gesendet. Aus den vorliegenden Quellen war jedoch nicht ersichtlich, wie man solch ein Break-Signal über die UART-Schnittstelle generiert. Des Weiteren ist die maximale Geschwindigkeitseinstellung für die UART-Schnittstelle im Standardkernel¹¹² 115200Baud. Somit wäre eine Modifizierung des Kernels notwendig um die, für DMX benötigte, Geschwindigkeit von 250kBaud zu ermöglichen.

Daraus resultierte, dass im Rahmen dieser Arbeit eine Verwendung des UART für die Ausgabe von DMX nicht in Frage kommt. Als Alternative für die Kommunikation mit DMX-Geräten bietet sich, aufgrund der großen Marktverbreitung von USB-DMX-Interfaces, die USB-Schnittstelle an.

5.4.1 Anforderungen an das USB-DMX-Interface

Für die stabile und sichere Ausgabe von DMX-Kanälen, müssen spezielle Anforderungen bei der Auswahl des Interfaces beachtet werden.

Eine, zwar nicht von der DMX-DIN-Norm spezifizierte, dennoch sinnvolle Vorgabe, ist die galvanische Trennung des DMX-Universums vom Controller. Somit wird eine Beschädigung der Steuerelektronik vermieden. Der Ausgang des Interfaces sollte durch einen Mikrocontroller intern gepuffert werden, um die zeitkritischen Sendevorgänge nicht dem Scheduling¹¹³ eines Multiprozesssystems zu überlassen und somit eine störungsfreie Übertragung zu gewährleisten. Einfache DMX-Interfaces verzichten teilweise auf diese Pufferung¹¹⁴. Findet keine Zwischenpufferung statt, ist die DMX-Ausgabe alleinig abhängig von den ausgesandten Signalen des Computers, an welchen das Interface angeschlossen ist. Da bei einem nicht-Echtzeitbetriebssystem die Rechenleistung, auf mehrere Prozesse aufgeteilt wird, kann dies zu falsch oder schlecht gesendeten DMX-Frames führen.

Weitere für dieses Projekt relevante Anforderungen waren die Möglichkeiten DMX Signale neben dem Versand auch empfangen zu können und die Kompatibilität mit Linux-Betriebssystemen, im Speziellen mit der Open Lighting Architecture sicherzustellen. Ein geringer Kostenaufwand, kleine Baumaße und gute Dokumentationen waren ebenfalls gewichtig. Nach diesen Kriterien wurde das Digital Enlightenment DMX-USB Interface ausgewählt. Dieses zeichnet sich besonders durch frei zugängliche Information zur Herstellung und Inbetriebnahme aus, was eine zukünftige Anpassbarkeit und Modifikation an die eigenen Anforderungen gewährleistet. Obwohl die Dokumentationen zur Software und Technik frei verfügbar sind, wird eine Variante, das FX5 DMX, kommerziell vertrieben. Dies resultiert für den hier verwendeten Anwendungszweck, in einem kommerziell erprobten und dennoch offenen System.

¹¹²ein Kernel, ist der zentrale Bestandteil eines Betriebssystems und regelt die grundlegenden Softwarefunktionen

¹¹³ein Scheduler regelt, in welcher zeitlichen Reihenfolge, Prozessen Ressourcen zugeteilt werden

¹¹⁴ein Beispiel für ein solches Interface ist das ENTTEC Open-DMX-USB

5.4.2 Prinzipieller Aufbau eines USB-DMX-Interfaces am Beispiel des DE-Interfaces

Der elementarste Bestandteil innerhalb des Protokollwandlers ist das USB-DMX Interface. Es ermöglicht den Empfang von DMX512-Signalen und somit die Grundvoraussetzung für eine spätere Wandlung auf Ethernet-Signale sowie die Ausgabe von gewandelten Ethernet-Paketen in Form von DMX512-Frames. Aufgrund der Wichtigkeit für die Funktionsweise des gesamten Protokollwandlers, wird das verwendete USB-DMX-Interface nun genauer beschrieben¹¹⁵.

Mit den in Kapitel 5.4.1 beschriebenen Anforderungen besteht ein USB-DMX-Interface mindestens aus einem USB-Controller, einem Mikrocontroller und dem EIA-485 Bustreiber mit vorhandener Optoisolierung (siehe Abb. 14).

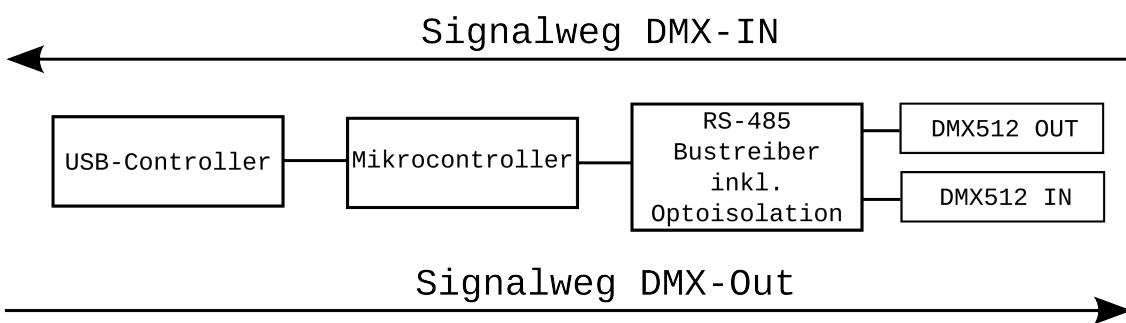


Abbildung 14: Schema Signalpfad USB-DMX-Interface

Ein USB-Controller wandelt die Signale der USB Schnittstelle des Computers in eine, für den Mikrocontroller verwertbare Form. Das Digital-Enlightenment USB-Interface¹¹⁶, verwendet den IC USBN9604-28M der Firma National Semiconductor, welcher die parallele 8Bit Ausgabe ermöglicht. Weiterhin unterstützt dieser USB-Controller die DMA-Technik¹¹⁷, welche es erlaubt die transferierten Daten direkt, also ohne "Umwege" über den Mikrocontroller, im Speicher abzulegen (siehe Abb. 34). Dies erhöht die Ausführungsgeschwindigkeit des Programms, da die Weiterleitung der transferierten Daten in den Speicher durch den Mikrocontroller viele Takte schritte in Anspruch nimmt.

Der in diesem Interface verwendete Mikrocontroller ist der ATmega 8515 aus der Atmel AVR-Serie. Dieser holt sich die über die USB-Schnittstelle gesendeten Daten, über ein Schieberegister (Latch) aus dem Speichermodul (hier ein SRAM¹¹⁸). Das Schieberegister wird benötigt, um den addressierbaren Bereich des 8-Bit Mikrocontroller zu erweitern¹¹⁹. Ein 8Bit Mikrocontroller kann während eines Zyklus genau ein Byte an Daten verarbeiten. Das hier verwendete SRAM-Modul, ermöglicht das Speichern von 8192 Speicherblöcken.

¹¹⁵eine detaillierte Funktionsübersicht ist im angehangen Schaltplan auf S. 70 in Abb. 33 ersichtlich

¹¹⁶Dokumentation und Aufbauanleitungen ersichtlich unter www.digital-enlightenment.de/usb-dmx.htm Stand: 12.06.2013

¹¹⁷Direct Memory Access

¹¹⁸Static random access memory

¹¹⁹Vgl. [Dre13]

Der Zugriff auf alle Speicherblöcke ist durch eine Adressierung mit 8Bit (Vgl. 8Bit = 256 Dezimal) nicht möglich. Aufgrund dessen ist es notwendig, die Daten zunächst in ein Schieberegister zu schreiben um seinen addressierbaren Bereich auf 13Bit (entspricht 8192 Speicherblöcken) zu vergrößern.

Die via USB empfangenen Daten werden nun im Mikrocontroller verarbeitet und die resultierenden Signale an den Bustreiber versendet. Im hier verwendeten Bustreiber-IC, dem 75176AP, findet die Pegelwandlung auf die nach EIA-485 spezifizierten, differenziellen Signalpegel statt. Der Signalfluss ist zwischen der Ausgabe des Mikrocontrollers und dem Wandeln im Bustreiber durch einen Optokoppler des Typs 6N137 galvanisch getrennt.

Der Empfang und Versand von RDM-Nachrichten ist mit dieser Schaltung nicht möglich, da wie in Kapitel 2.4 beschrieben, jeder DMX512-Kanal eine kombinierte Send- und Empfangsschaltung inkl. Erkennung der Signalrichtung benötigt. Die RDM-Funktionalität kann jedoch durch Anschluss eines RDM-fähigen USB-DMX Interfaces an den Protokollwandler erreicht werden.

Das Digital Enlightenment Interface hat zusätzlich den Vorteil, dass sich die Timingparameter des DMX-Signals konfigurieren lassen. Dies ermöglicht das Integrieren von DMX-Technik, die nicht zu 100% dem DMX512-Standard entspricht und bspw. ein vergrößertes Break-Signal benötigt.

5.5 Stromversorgung

Die Stromversorgung der gesamten Peripherie findet über ein, im Gehäuse integriertes, Schaltnetzteil statt. Dieses wandelt 230V Wechselspannung auf 5V Gleichspannung. Der maximal mögliche Gleichstrom beträgt 5A. Die Zuführung der Wechselspannung zum Gehäuse erfolgt über einen PowerCON-Anschluss der Firma Neutrik.

Die Verteilung der Gleichspannung vom Netzteil zu den einzelnen 5V-Bauteilen erfolgt über die in Kapitel 5.3 beschriebene Erweiterungsplatine. Über 2,54mm-Buchsenleisten wird über Steckverbinder die 5V-Spannung für den Raspberry Pi, den USB-Hub, das LCD-Display sowie die PWM-Ausgänge bereitgestellt. Für das PWM-Modul ist eine zusätzliche 5V-Spannungsquelle erforderlich, da an den herausgeführten Buchsenleisten für jeden PWM-Kanal ein separater 5V-Anschluss herausgeführt ist¹²⁰.

230V AC Bauteile			
Nr.	Bauteil	Anzahl	Stromverbrauch in mA
1	15 % Verlustleistung Schaltnetzteil 5A 5V	1	750
		Teilstrom in mA	750
		Teilleistung in W	3,75

5V DC Bauteile			
2	Raspberry Pi Model B	1	750
3	DE USB-DMX Interface	2	375
4	LCD-Display	1	50
5	15% Verlustleistung DC/DC- Boost-Modul	1	160
6	PWM-Modul	1	2000
		Teilstrom in mA	3335
		Teilleistung in W	16,675

9V DC Bauteile			
7	TP-Link TL-WR841ND	1	600
		Teilstrom in mA	600
		Teilleistung in W	5,4
Leistung in W			25,825

Abbildung 15: Auflistung des Stromverbrauchs der verwendeten Komponenten

Das Ein- und Ausschalten des Geräts ist über einen Schalter neben der PowerCON-Buchse möglich. Da ein plötzliches Trennen der Stromversorgung des Raspberry Pi, jedoch zum Korrumpern einzelner Speicherblöcke der SD-Karte führen kann, ist der Raspberry Pi über einen zwischengeschalteten Akku mit der Spannungsquelle verbunden. Im Fall eines Stromausfalls, bleibt der Raspberry Pi eingeschalten und kann durch einen zusätzlichen Taster an der rechten Gehäuseseite heruntergefahren werden (siehe Abb. 16).

¹²⁰dies ist beim Anschluss zusätzlicher Peripherie (bspw. Servomotoren) von Vorteil, da keine externe Spannungsquelle benötigt wird

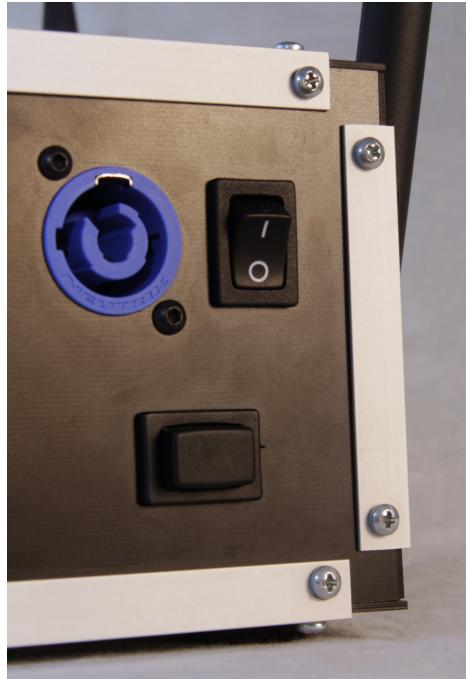


Abbildung 16: Anschlüsse und Bedienelemente der Stromversorgung

Der verwendete Router, ein TP-Link TL-WR841ND, hat eine Betriebsspannung von 9V DC. Um die Integration eines zweiten Netzteils mit einer Spannung von 9V bzw. eines deutlich kostspieligeren kombinierten 5V/9V Netzteils zu vermeiden, wird die 5V Spannung durch einen Gleichspannungswandler auf 9V erhöht.

Die kombinierte, maximale Leistungsaufnahme bei gleichzeitiger Benutzung aller Komponenten beträgt, wie in Abb. 15 ersichtlich, ca. 25,8W.

5.6 Gehäuse

Als überraschend große Herausforderung stellte sich die Suche nach einem geeigneten Gehäuse dar. Zu Beginn des Projekts wurden 3D-Modelle der Bauteile zusammengefügt um die Gehäusedimensionen, sowie die Positionen der Anschlüsse definieren zu können. Die daraus entstandenen Renderings, in Abb. 17 ersichtlich, dienten als Vorlage für den weiteren Entwicklungsprozess.

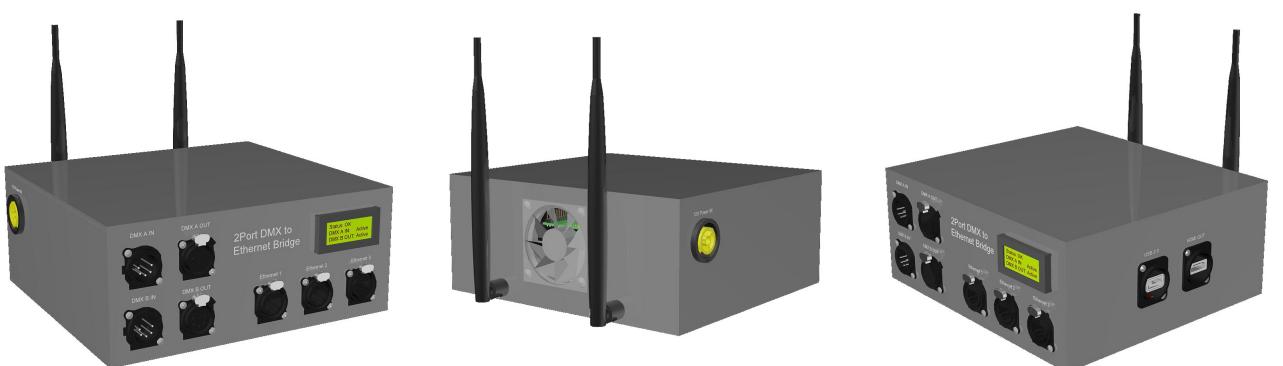


Abbildung 17: Renderings 3D-Gehäusekonzept

Mit den aus dieser Zeichnung bekannten Abmaßen, ca. 200 x 180 x 100mm, wurde versucht ein passendes Standardgehäuse auf den Onlineplattformen großer Elektronikversandhäuser zu finden, welche eine Vielzahl an Gehäusen anbieten. Diese Suche erzielte jedoch nicht die gewollten Ergebnisse. Die Gehäuseabmaße waren zu stark von den Gewünschten abweichend. Somit wurde es nötig ein Gehäuse zu konstruieren und in den geforderten Maßen zu fertigen. Dieses, 200x180x94mm große, Gehäuse besteht aus vier Seitenteilen, einer Deck- und einer Bodenplatte sowie zwei Einlegeböden, zum Positionieren der Peripherie. Die Frontplatte bietet Aussparungen für zwei fünfpolige XLR DMX-Ein- und Ausgänge, drei Ethercon-Buchsen, zur Netzwerkanbindung und einem 16x2 Zeichen LCD.

Die rückseitige Platte enthält Aussparungen, für die Antennen des WLAN-Routers. Bei der Erstellung des ersten Gehäusekonzepts wurde noch davon ausgegangen, dass eine aktive Kühlung durch einen Lüfter nötig ist. Die Berechnung der maximalen Leistungsaufnahme (in Abb. 15 auf S. 42 ersichtlich) zeigte jedoch, dass die maximal mögliche Abwärme keinen zusätzlichen Lüfter rechtfertigt.

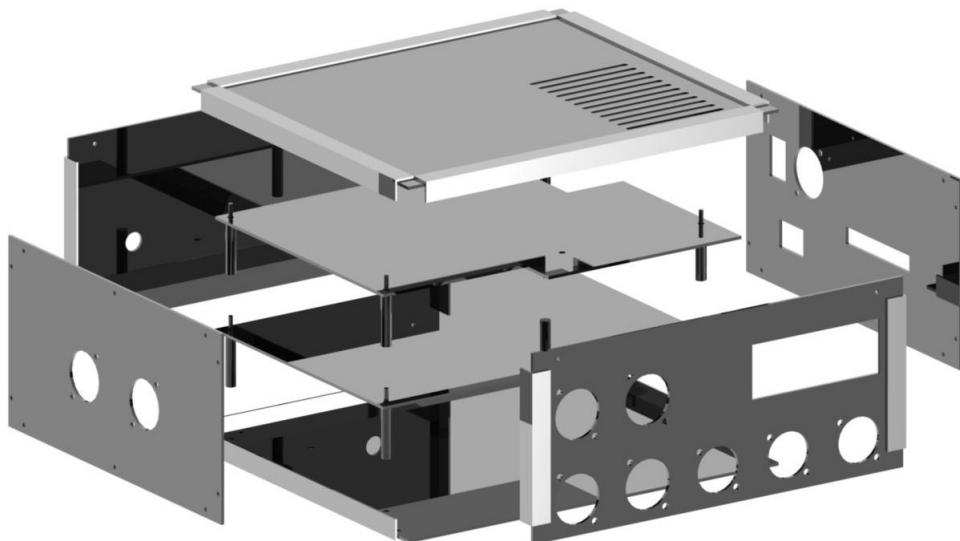


Abbildung 18: Explosionsansicht Gehäusebauteile

Die linke Seitenplatte enthält Bohrungen für einen USB- und einen HDMI-Anschluss. Auf der rechten Gehäuseseite sind Ausschnitte für die PWM-Ausgänge, die analogen Eingänge, eine PowerCon-Buchse sowie ein On/Off-Schalter und ein Taster zum Herunterfahren vorhanden. Auf der Bodenplatte, befinden sich Bohrungen, zur Befestigung der Einlegeböden¹²¹. Alle Gehäuseteile wurden aus schwarzem Glasfaserverstärkten Kunststoff (GFK) gefertigt und mittels 10x10x1mm Aluminium-Profilen verbunden. Die Herstellung der Bauteile erfolgte auf einer CNC-Fräse des Typs "Advance 02 CNC Fräsmaschine Johannes Kerkhoff Technik", welche eine Bearbeitungsfläche von 600mm x 400mm bietet, so können alle Bauteile aus einer Platte gefräst werden.

¹²¹die technischen Zeichnungen des Gehäuses sind in Abb. 35 und 36 ersichtlich

Das verwendete Werkzeug war ein spiralverzahnter 2mm Vollhartmetall-Fräser. Da jede Platte über separate Schraubverbindungen zum Gehäuse verfügt, ist ein Austausch eines defekten Gehäusebauteils einfach möglich. Die Wartung und der Austausch defekter Module im Innern des Geräts, ist dadurch ebenfalls erleichtert. Ebenfalls wichtig für gute Wartbarkeit, ist die Art der Bauteilbefestigungen.



Abbildung 19: Fotografie des fertiggestellten Gehäuseprototypen

Jegliche Peripherie wird mittels Schraubverbindungen auf den Einlegeplatten befestigt. Die Einlegeplatten sind über Abstandshalter miteinander verbunden, so dass diese etagenweise abgenommen werden können, um defekte Module auszutauschen.

5.7 Kosten

Eines der wichtigsten Kriterien während der Konstruktion eines Produkts, sind die dabei entstehenden Kosten. Es gilt eine ständige Abwägung zu treffen zwischen Funktionsvielfalt und maximalen Kostenrahmen. Die implementierten Funktionen des DMX-Protokollwandlers wurden bereits in Kapitel 5.1 besprochen. Die angestrebte Gesamtsumme der Bauteilkosten betrug 200 € exkl. MwSt. .

Wie in Abb. 20 ersichtlich wurde diese Zielsetzung knapp überschritten. Dies ist insbesondere darauf zurückzuführen, dass die in der Veranstaltungstechnik standardisierten und bewährten Steckverbinder system einen erheblichen Kostenaufwand verursachen. Als Beispiel sei der Vergleich zwischen den Anschaffungskosten des Routers, 22,85 €, und den drei EtherCON Steckverbinder, 25,05 €, genannt. Der Preis für die Steckverbinder übersteigt hier die Kosten der elektronischen Komponenten.

Embedded Computer und Zusatzmodule				
Nr.	Artikelbeschreibung	Kosten/Stk.	Anzahl	Artikelkosten
1	Bauteile Digital Enlightenment USB DMX Interface (ohne Platinenherstellung)	25,45 €	2	50,90 €
2	Raspberry Pi Model B	31,95 €	1	31,95 €
3	SD-Karte 4GB Class 10	7,00 €	1	7,00 €
4	I²C-PWM Interface – Adafruit	12,00 €	1	12,00 €
5	2x16 Zeichen LCD-Modul	3,99 €	1	3,99 €
6	WLAN Router / TP-Link TL-WR841ND	22,85 €	1	22,85 €
7	powered USB Hub	5,00 €	1	5,00 €
8	Akku zur unterbrechungsfreien Stromvers.	5,95 €	1	5,95 €
9	Bauteilkosten Erweiterungsplatine; inkl. MCP3008	6,00 €	1	6,00 €
Teilsumme 1:				145,64 €

Gehäuse und Steckverbinder				
Nr.	Artikelbeschreibung	Kosten/Stk.	Anzahl	Artikelkosten
10	Rohmaterial (GFK) für Gehäuse	20,00 €	1	20,00 €
11	DMX 5-Pol Einbaubuchse / NC5FD-L-B-1	9,97 €	2	19,94 €
12	DMX 5-Pol Einbaustecker / NC5MD-LX-BAG	6,00 €	2	12,00 €
13	EtherCON Einbaubuchse / NE8FDV-YK-B	8,35 €	3	25,05 €
14	USB 2.0 Einbaubuchse / NAUSB-W-B	4,52 €	1	4,52 €
15	HDMI-Out Einbaubuchse / NAHDMI-W-B	10,69 €	1	10,69 €
16	PowerCON Buchse / NAC3MPA-1	2,78 €	1	2,78 €
17	Verbindungskabel und Kleinteile	10,00 €	1	10,00 €
Teilsumme 2:				104,97 €
Gesamtsumme inkl. MwSt.:				250,61 €
Gesamtsumme exkl. MwSt.:				210,60 €

Abbildung 20: Auflistung der Kosten aller verwendeten Bauteile

Das Verhältnis der Kosten zwischen aktiven Komponenten (Elektronik-Bauteile) und inaktiven Komponenten (Steckverbinder, Gehäuse, Verbindungskabel, etc.) beträgt ca. 58% zu 42%. Diese unerwartete Kostenverteilung ist meiner Einschätzung nach darauf zurückzuführen, dass die Produktion der elektronischen Komponenten in deutlich größeren Stückzahlen und für einen viel größeren Absatzmarkt erfolgt, was den Kostenfaktor eines Produkts deutlich verringert.

Ein am Markt erhältliches Gerät mit ähnlicher Funktionsvielfalt wie der in dieser Arbeit beschriebene DMX-Ethernet-Wandler, bspw. die in Kapitel 4 beschriebene E-Gate-Serie der Firma transtechnik¹²², kostet in der Basisversion E-Gate plus 2350€. Dies bedeutet einen Preisunterschied von ca. Faktor zehn.

Hierbei gilt es jedoch zu beachten, dass bei einem kommerziellen Produkt, weitreichende Zusatzkosten im Herstellungs- und Vertriebsprozess anfallen. Dies sind Gewährleistungs- und Serviceleistungen, Zertifizierungen, Entwicklungs- und Forschungsarbeit, Fertigungskosten sowie Vertriebs- und Marketingkosten.

Um einen relevanten Vergleich zwischen am Markt bestehenden und dem hier vorgestellten Protokollwandler anführen zu können, wäre eine detaillierte Abschätzung der

¹²²vertrieben durch die Electronic Theatre Controls GmbH - ETC

Herstellungskosten nötig, was nicht Teil dieser Arbeit sein soll.

Der Preisunterschied der Produkte erscheint dennoch enorm. Eine Einordnung der Preisgestaltungen bleibt jedoch sehr komplex, da die Branche der Veranstaltungstechnik aufgrund ihres kleinen Marktsegments und den damit einhergehenden geringen Stückzahlen nur schwer vergleichbar mit den vorgeschlagenen Margen in der Wirtschafts-Fachliteratur ist.

6 Übersicht der verwendeten Software

Um die Funktionen der Hardware zu ermöglichen, werden die einzelnen Komponenten durch Software angesteuert. Eine der größten Anforderungen an die Hardware, der freie Zugang zu Funktion und Aufbau der Module, galt bezüglich der Software. Diese sollte quelloffen und somit transparent und anpassbar sein.

6.1 Betriebssystem und Softwareplattform

Das grundlegendste Element der Software innerhalb eines Computers ist das Betriebssystem. Das für den Raspberry Pi verwendete Betriebssystem ist das Debian Derivat "Raspbian". Dies ist eine Linux-Distribution, welche speziell auf die Hardware, vor allem auf den ARM11-Prozessor, des Raspberry Pi angepasst wurde¹²³.

Die von mir verwendete Distribution¹²⁴ wird durch die Entwickler der Open Lighting Architecture bereitgestellt und enthält alle Abhängigkeit, welche zum kompilieren der OLA-Software benötigt werden. Als Kompilieren bezeichnet man den Vorgang der Übersetzung von Programmcode in maschinenlesbaren Code, also Binärdateien.

Die Open Lighting Architecture (OLA) bildet die Grundlage für die Funktionsweise des Protokollwandlers. Sie verfügt bereits über zahlreiche Implementierungen verschiedener Lichtsteuerprotokolle und bietet Unterstützung für eine Vielzahl an Lichtsteuerungs-Peripherie. Die einzelnen Treiber und Protokolle sind als Plugins integriert und können somit einfach durch das Hinzufügen weiterer Plugins ergänzt werden. Zudem werden diverse Programmierschnittstellen (sog. APIs¹²⁵), bspw. in den Programmiersprachen C++, Python und Java, angeboten¹²⁶. Eine detailliertere Beschreibung der Funktionsweise der Open Lighting Architecture ist in Kapitel 7 ersichtlich.

Die von mir verwendete Programmiersprache zur Funktionserweiterung der Software ist Python. Diese ist eine interpretierte Programmiersprache, welche unterschiedliche Programmierparadigmen, bspw. Objektorientierung, Aspektorientierung und die funktionale Programmierung, unterstützt. Besondere Bedeutung liegt bei der Entwicklung auf einfacher Programmlesbarkeit und somit guter Verständlichkeit des Quellcodes¹²⁷. Python ist aktuell die fünftbeliebteste Programmiersprache auf der bekannten Software-Hosting-Plattform Github¹²⁸.

¹²³Quelle: <http://www.raspbian.org/RaspbianFAQ> Stand: 17.06.2013

¹²⁴zum Download verfügbar unter <http://nomis52.net/data/raspi/raspbian-git-21042013.zip> Stand: 17.06.2013

¹²⁵application programming interface

¹²⁶Vgl. http://www.opendmx.net/index.php/OLA_developer_info Stand: 17.06.2013

¹²⁷Vgl. <http://www.python.org/about/> Stand: 17.06.2013

¹²⁸Quelle: <https://github.com/languages/Python> Stand: 17.06.2013

6.2 Softwareeverwaltung- und Veröffentlichung

Die Verwaltung und Veröffentlichung von Software sowie deren Dokumentation ist über eine Vielzahl an kollaborativen Online-Tools möglich.

Alle hier verwendeten Projekte benutzen zur Verwaltung und Versionierung der Software das Versionsverwaltungs-Tool “git“. Dies ist eine freie Software, welche ursprünglich für die Quellcode-Verwaltung des Linux-Kernels entwickelt wurde¹²⁹. Sie ermöglicht das verteilte Arbeiten an einem gemeinsamen Projekt (Repository), indem sich jeder Bearbeiter eine lokale Version des zu bearbeitenden Repositorys herunterlädt (ein sog. pull) und anschließend seine Änderung wieder in das Hauptprojekt integriert (ein sog. push). Des Weiteren ist es möglich, eine unabhängige Kopie des Hauptprojekts anzulegen, um Änderungen, Anpassungen und Weiterentwicklungen vorzunehmen. Diese Abspaltung des Hauptprojekts wird als “fork“ bezeichnet.

Für die Speicherung und Versionierung der Repositorys werden meist Online-Speicher sog. Git-Hosts verwendet. OLA verwendet als Online-Speicher Google Code¹³⁰. Alle im Rahmen dieser Bachelorarbeit verwendeten Repositorys und deren Entwicklungen sowie Dokumentationen zur Hardware finden sich auf dem dafür eingerichteten Github-Repository¹³¹. Eine Übersichtsseite des Projekts wurde unter der URL: <http://astreubel.github.io/Bachelorthesis/> bereitgestellt.

6.3 Peripherietreiber und Softwareerweiterungen

Der wichtigste Software-Teil dieses Protokollwandlers ist, aufbauend auf der Open Lighting Architecture, das Plugin und somit der Treiber zum Ansteuern des USB-DMX Interface. Zu Beginn der Bearbeitungszeit (April 2013) dieser Bachelorarbeit stand kein Plugin für das Digital-Enlightenment Interface für die Open Lighting Architecture zur Verfügung. Eine Alternative war, vor allem in Bezug auf Preis und Funktionsumfang, nur mit Abstrichen verfügbar.

Erfreulicherweise wurde durch einen Fork des OLA-Hauptprojekts bereits an einer Implementierung gearbeitet, welche seit Mitte Juni verfügbar ist. Diese ist jedoch noch nicht im zuvor erwähnten OLA-Repository vorhanden, sondern nur unter dem Fork-Repository erreichbar¹³² und muss somit selbst kompiliert werden¹³³. Das Plugin implementiert den ursprünglich für Windows bereitgestellten Treiber¹³⁴ mit Hilfe der Software-Bibliothek “hidapi“¹³⁵ unter Linux-Systemen. Die Ansteuerung des LC-Displays erfolgt

¹²⁹Vgl. <https://de.wikipedia.org/wiki/Git> Stand: 17.06.2013

¹³⁰<https://code.google.com/p/open-lighting/> Stand: 17.06.2013

¹³¹verfügbar unter der URL: <https://github.com/Astreubel/Bachelorthesis>

¹³²Fork erreichbar unter <https://github.com/mlba-team/open-lighting> Stand: 17.06.2013

¹³³ausführliche Dokumentation zum Kompilationsvorgang, verfügbar unter http://www.opendmx.net/index.php/OLA_on_Linux Stand: 17.06.2013

¹³⁴der Treiber wurde unabhängig vom OLA-Plugin entwickelt und ist unter der URL <https://github.com/mlba-team/usbdmx> abrufbar

¹³⁵diese Bibliothek ist eine der benötigten Abhängigkeiten beim Kompilieren des OLA-Forks, erreichbar unter <https://github.com/mlba-team/hidapi> Stand: 17.06.2013

auf der Grundlage einer Software-Bibliothek von Adafruit¹³⁶ und der OLA Python-API. Für die Kommunikation mit dem Display wird die parallele Schnittstelle des HD44780-LCD-Moduls im 4-Bit-Modus verwendet. Die implementierten Funktionen des LCD-Skripts sind die Rückmeldung während des Bootvorgangs und das Anzeigen der aktuellen IP-Adresse. Nach abgeschlossenem Systemstart werden, wenn vorhanden, die momentan konfigurierten Universen angezeigt. Ist kein Universum definiert, wird dies über das LCD ausgegeben. Erfolgt ein Herunterfahren über den in Kapitel 5.5 beschriebenen Taster, wird ebenfalls eine Rückmeldung ausgegeben, sobald das Herunterfahren gestartet wird.

Um das Programm automatisch bei jedem Bootvorgang starten zu lassen, ist es nötig, einen Systemaufruf zu platzieren, der ein “init.d“-Skript startet. Dabei handelt es sich um Shellskripte in Unix-basierten Betriebssystemen, welche Systemprozesse bei jedem Bootvorgang ausführen¹³⁷.

Zur Ausgabe der PWM-Signale wird ein Python Skript, welches ebenfalls auf den Raspberry Pi-Bibliotheken von Adafruit aufbaut, verwendet. Der Treiber implementiert die PWM-Ausgabe über die I2C-Schnittstelle des Raspberry Pi. Ein auf Github veröffentlichtes Projekt verbindet diese Bibliothek mit der Open Lighting Architecture. Dieser als “lightingPi“ benannte Fork der Adafruit-I2C-Bibliothek ermöglicht die Kommunikation zwischen OLA und dem PWM-Modul. Über eine Konfigurationsdatei kann eingestellt werden, mit welchem DMX-Universum die PWM-Ausgänge gekoppelt (Standard ist Universum 1) und welche DMX-Kanäle benutzt werden sollen. Das PWM-Modul belegt standardmäßig die ersten 16 Kanäle des konfigurierten Universums. Das Python-Skript wird nicht beim Systemstart, sondern manuell gestartet, um eine ungewollte Ausgabe von Signalen auf den PWM-Kanälen zu verhindern. Das Starten ist entweder über einen angeschlossenen Bildschirm und eine Tastatur oder über eine SSH-Verbindung¹³⁸ möglich. Eine Einbindung in das OLA-Webinterface wäre in einer zukünftigen Weiterentwicklung erstrebenswert. Die analogen Eingänge sind momentan nur in der Hardware vorbereitet, jedoch noch nicht in der Software implementiert.

7 OLA - Open Lighting Architecture

Die Grundlage aller Lichtsteuerungs- und Netzwerkfunktionen dieses Protokollwandlers bildet die Open Lighting Architecture. OLA, ursprünglich unter dem Namen LLA gestartet, ist ein Open-Source Framework des Open Lighting Projects mit dem Ziel die Einführung neuer Lichtsteuerprotokolle durch qualitativ hochwertige und zuverlässige Software zu beschleunigen¹³⁹. Sie ermöglicht das Konvertieren von zahlreichen Ethernet-

¹³⁶verfügbar unter https://github.com/adafruit/Adafruit-Raspberry-Pi-Python-Code/tree/master/Adafruit_CharLCD Stand: 17.06.2013

¹³⁷Quelle: http://openbook.galileocomputing.de/unix_guru/node103.html Stand: 17.06.2013

¹³⁸das SSH(Secure Shell)-Protokoll ermöglicht die Fernwartung eines Geräts über eine Netzwerkverbindung

¹³⁹Vgl. http://www.opendmx.net/index.php/Open_Lighting_Project Stand: 19.06.2013

DMX Protokollen und die Ein- und Ausgabe von DMX512-Signalen inkl. RDM durch eine große Vielfalt an USB-DMX Interfaces¹⁴⁰. Die Kernfunktion von OLA besteht aus der Verknüpfung von Hardware (bspw. DMX-USB Interfaces) und den Ethernet-Steuerprotokollen. Dies ermöglicht das Routen und Wandeln von Lichtsteuersignalen verschiedenster Anforderungen in einem Netzwerk. Im Folgenden werden die unterstützten Funktionen und Geräte der Software näher dargestellt.

7.1 Aufbau der Open Lighting Architecture

Das OLA-Framework besteht im Wesentlichen aus vier Komponenten. Dem OLA-Daemon (Bez.: olad), als Daemon wird unter unixoiden Betriebssystemen ein im Hintergrund laufendes Programm bezeichnet, welches unterschiedliche Dienste bereitstellt¹⁴¹, der OLA-Client-Bibliothek (Bez.: libola für C++; OlaClient.py für Python), diversen Plugins (Bez.: libolaX, wobei X für den Namen des Plugins steht) und zahlreichen Beispieldatenwenden der Plugins und Client-Programme¹⁴².

Der OLA-Deamon stellt einen TCP-Server bereit, mit dem jeder Client kommuniziert. Die Client-Anwendung kann dabei virtuell, bspw. eine Implementierung eines Ethernet-Protokolls, oder real, bspw. ein DMX512-Interface sein¹⁴³. Jeder Client stellt dabei mindestens einen Port zur Verfügung, der 512 DMX-Kanäle repräsentiert. Diese Kanäle können je nach Konfiguration Ein- oder Ausgänge sein.

Ein besonderer Client ist der “OladHTTPServer“. Dieser trägt die Informationen des Daemons in einem Webinterface auf Port 9090 zusammen und ermöglicht zudem die Konfiguration und Verwaltung von Plugins. Ein Plugin kann aus mehreren Ports bestehen, also über Ein- und Ausgänge verfügen. Ein Beispiel dafür ist das ArtNet-Plugin, welches einen Input- und einen Output-Port bereitstellt. Bei jedem Start des OLA-Daemons wird überprüft, welche Plugins über angeschlossene Geräte verfügen, also ob bspw. ein DMX512-Interface angeschlossen ist.

7.2 Funktionsübersicht der Open Lighting Architecture

Der Funktionsumfang der Software wird durch die implementierten Plugins bestimmt. Die relevantesten Plugins für den in dieser Arbeit beschrieben Protokollwandler sind das ArtNet-, das sACN- und das Enlightenment-Plugin. Letzteres implementiert die Funktionalität des Digital-Enlightenment DMX-USB Interfaces in dem OLA-Framework. Weitere Plugins bieten Unterstützung für die Protokolle ESP-Net von Enttec, Shownet von Strand, Pathport von Pathway Connectivity, SandNet von Sandsys und KiNet von

¹⁴⁰die unterstützten Protokolle sowie USB-Interfaces sind in Kapitel 7.2 ersichtlich

¹⁴¹Vgl. <https://de.wikipedia.org/wiki/Daemon> Stand: 19.06.2013

¹⁴²Vgl. http://www.opendmx.net/index.php/Using_OLA Stand: 19.06.2013

¹⁴³Vgl. http://www.opendmx.net/index.php/OLA_developer_info Stand: 19.06.2013

Color Kinetics¹⁴⁴. Des Weiteren ist die Integration von OSC¹⁴⁵-Signalen über ein separates Plugin möglich.

Aktuell werden bereits über 20 DMX-Interfaces durch die Open Lighting Architecture unterstützt. Darunter auch zahlreiche RDM-fähige Geräte, bspw. das DMX-USB-PRO der Firma Enttec¹⁴⁶.

Für diese RDM-fähigen Interfaces ist eine weitere Funktion, der "RDM Responder Test" verfügbar. Jener ermöglicht es Herstellern von RDM-fähiger Hardware, ihre Implementation des E1.20-Standards (RDM) anhand zahlreicher Tests zu überprüfen¹⁴⁷. Diese Tests können wahlweise über die Kommandozeile oder das Webinterface durchgeführt werden. Es werden insgesamt 235 Testfälle, darunter bspw. Grenzfälle des in Kapitel 2.4.1 beschriebenen RDM-Discoverys, abgedeckt. Über das unter Port 9099¹⁴⁸ erreichbare Webinterface kann der Test vor der Durchführung konfiguriert und die Ergebnisse anschließend gespeichert werden. Nach Abschluss des Tests ist es auch möglich, die über ein Gerät gesammelten Ergebnisse auf die Internetseite www.rdm.openlighting.org hochzuladen. Dadurch ist bereits eine Datenbank von über 300 RDM-Geräten und ihren RDM-Implementationen abrufbar¹⁴⁹.

Analog zu den RDM-Tests ist bereits eine Vorabversion eines E1.33-Testtools verfügbar¹⁵⁰. E1.33, auch als RDMNet bezeichnet, ist ein Protokoll zum Senden von RDM-Signalen über ein Ethernet-Netzwerk. Die endgültige Fassung des Standards ist noch nicht herausgegeben, eine Veröffentlichung ist jedoch in absehbarer Zeit zu erwarten.

OLA stellt auch ein SPI-Plugin zur Verfügung. Dieses ermöglicht das direkte Ansteuern von LED-Pixeln, sofern die Hardware eine SPI-Schnittstelle zur Verfügung stellt. Die dabei unterstützten LED-Pixel-Treiber sind "WS2801" und "LPD8806"¹⁵¹. Die SPI-Schnittstelle ist bei dem hier vorgestellten Protokollwandler jedoch nicht am Gehäuse herausgeführt und somit nur über den GPIO-Pinheader des Raspberry Pi verfügbar.

Eine besonders wichtige Funktionalität von Protokollwandlern ist das Priorisieren mehrerer Lichtsteuerungs-Signalquellen. In konventionellen DMX512-Kreisen hat sich dabei die Unterscheidung zwischen HTP (Highest Takes Precedence) und LTP (Latest Takes Precedence) etabliert. Im HTP-Modus wird jeweils der höhere Wert, im LTP-Modus der zuletzt gesendete Wert von mind. zwei Signalquellen übertragen.

Moderne Lichtsteuernetzwerke unterstützen zusätzlich jedoch auch die separate Priorisierung von Quellen über Prioritätswerte. Dies ist deutlich universeller und bietet weitreichendere Möglichkeiten des "Mergings".

Das von OLA unterstützte Streaming-ACN-Protokoll (E1.31) stellt bspw. Funktionen zur Priorisierung von DMX-Universen bereit. Bei der Konfiguration von Universen in

¹⁴⁴Quelle: <http://www.opendmx.net/index.php/OLA> Stand: 19.06.2013

¹⁴⁵Open Sound Control

¹⁴⁶Quelle: <http://www.opendmx.net/index.php/OLA> Stand: 19.06.2013

¹⁴⁷Vgl. http://www.opendmx.net/index.php/OLA_RDM_Responder_Testing Stand: 19.06.2013

¹⁴⁸die vollständige URL ist von der IP-Adresse des Protokollwandlers abhängig

¹⁴⁹Vgl. [New13]

¹⁵⁰abrufbar unter http://www.opendmx.net/index.php/E1.33_SLP_SA_Tests Stand: 19.06.2013

¹⁵¹Vgl. http://www.opendmx.net/index.php/OLA_LED_Pixels Stand: 19.06.2013

OLAs Webinterface (siehe Abb. 21) wird generell jeder Signalquelle ein Wert (“source priority“) zur Priorisierung von 0 bis 200 vergeben.

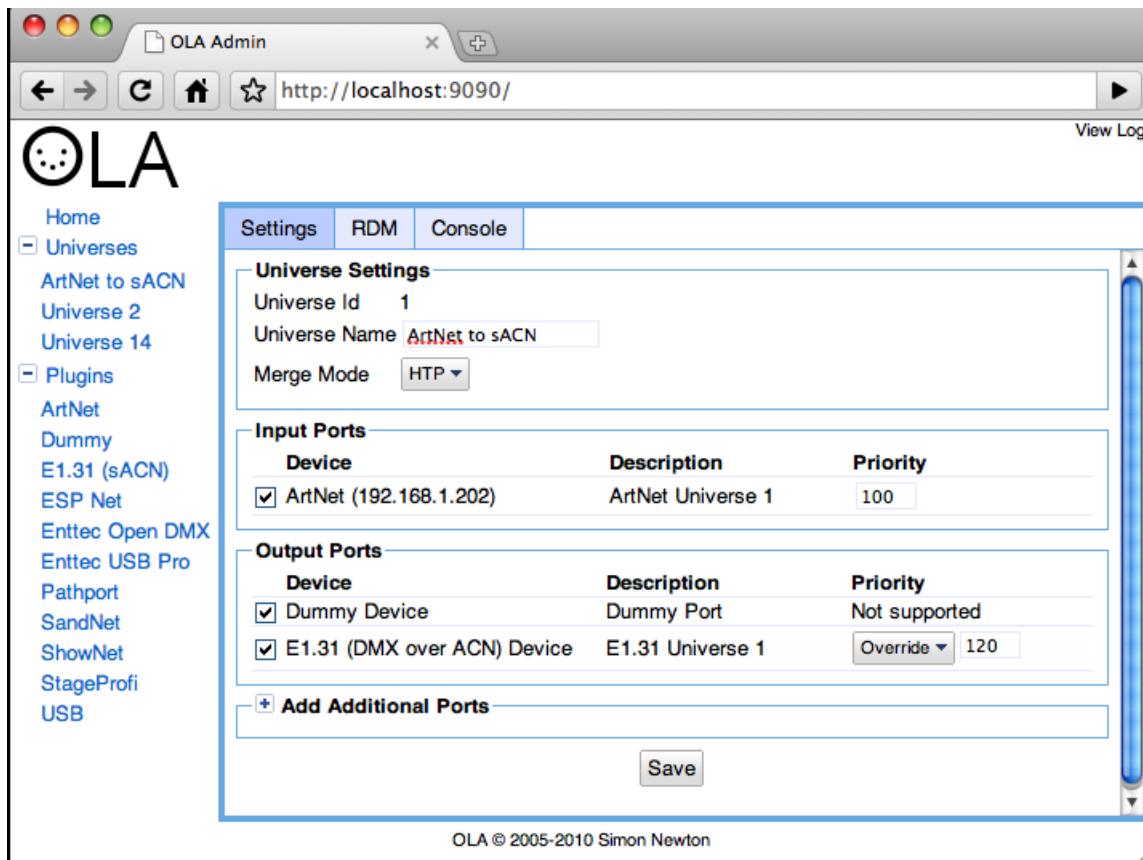


Abbildung 21: Screenshot des Webinterfaces der Open Lighting Architecture
 Quelle: Simon Newton, http://www.opendmx.net/index.php/File:Llad_home.png
 Stand: 20.06.2013

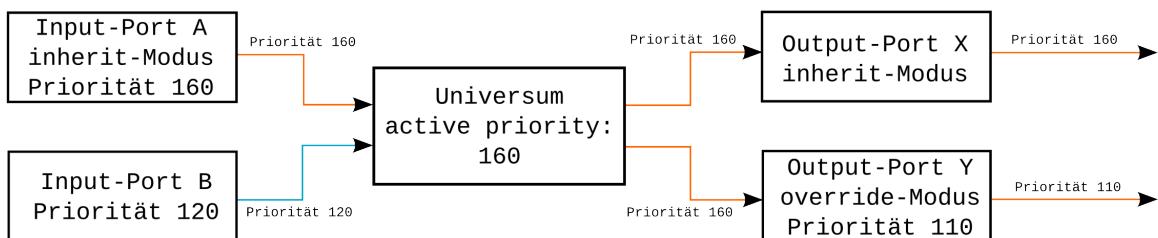
Dabei steht 0 für die niedrigste, 200 für die höchste und 100 für die Standard-Priorisierung. Es ist auch möglich einen sog. “Inherit“-Modus zu benutzen, bei dem die Signalquelle ihre Priorität der Priorität des ankommenden Signals (bspw. eines sACN-Datenstroms) anpasst.

Der “override“-Modus dagegen überschreibt die Priorität des ankommenden Signals mit dem in der Open Lighting Architecture spezifizierten Wert. Die “Override“- und “Inherit“-Modi können, wie in Abb. 22 zu erkennen, ebenso auf die Ausgangsports angewandt werden.

Inwieweit diese Funktionen unterstützt werden, ist jedoch vom verwendeten Plugin abhängig, da nur wenige Steuerprotokolle den “Inherit“-Modus unterstützen. Jedes Universum besitzt zudem einen “active priority“-Wert. Dieser bestimmt, welche Signalquelle an den Ausgabeports ausgegeben wird. Die Informationen jeder Signalquelle mit einer Priorität unter dem aktuellen “active priority“-Wert werden verworfen¹⁵² (siehe Abb. 22). Empfängt jedoch der Input-Port mit der höchsten Priorität keine Signale mehr, verringert sich die “active priority“ auf den nächsthöheren Prioritätswert eines Eingangssignals.

¹⁵²Vgl. http://www.opendmx.net/index.php/OLA_Merging_Algorithms Stand: 19.06.2013

Zustand 1 - Input-Port A mit aktivem Signal



Zustand 2 - Input-Port A ohne aktivem Signal

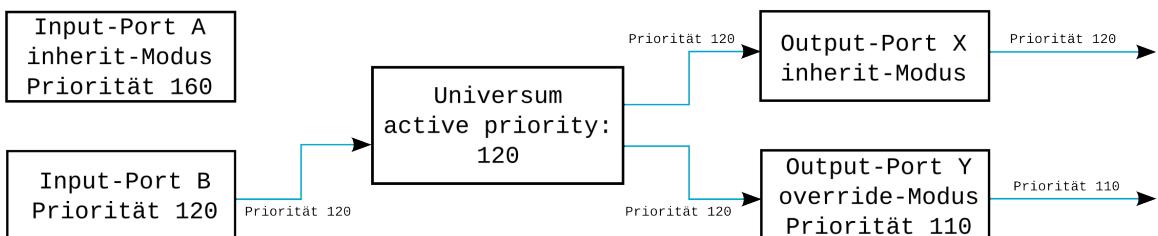


Abbildung 22: schematische Darstellung zweier priorisierte Merge-Vorgänge

Weisen zwei Quellen die gleiche Priorität auf, findet ein Merging auf Basis von LTP oder HTP statt.

Diese Merging-Varianten bieten zahlreiche Kombinationsmöglichkeiten für spezifische Anwenderfälle und selbstverständlich auch die Möglichkeit zur Kopplung eines Haupt- und Backup-Steuerpults.

Eine weitere Entwicklung des Open Lighting Projekts ist der “OLA DMX Trigger“. Dieses Kommandozeilenprogramm ermöglicht es, andere Programme durch den Empfang von DMX-Daten auszuführen¹⁵³. Da der verwendete embedded Computer, der Raspberry Pi, über einen HDMI-Ausgang verfügt, ist es bspw. möglich, Audio- und Videodateien ferngesteuert durch ein Lichtsteuerpult abzuspielen. Dazu wird in einer Konfigurationsdatei ein DMX-Kanal, ein DMX-Wertebereich sowie der Programmaufruf definiert. Verändert sich nun der spezifizierte DMX-Kanal auf einen festgelegten Wert, wird der Programmaufruf gestartet. Zusätzliche Einstellmöglichkeiten spezifizieren, ob ein Aufruf nur bei einem DMX-Werteanstieg oder beim Verringern des DMX-Wertes stattfinden soll¹⁵⁴. Vor dem Starten des “OLA DMX Triggers“ muss ein DMX-Universum mit einem DMX512-Input definiert werden. Anschließend kann das Programm von der Kommandozeile aus gestartet und mittels Überabeparameter beim Starten das verwendete DMX-Universum definiert werden.

¹⁵³Vgl. http://www.opendmx.net/index.php/OLA_DMX_Trigger Stand: 19.06.2013

¹⁵⁴Vgl. http://www.opendmx.net/index.php/OLA_DMX_Trigger Stand: 20.06.2013

8 Funktionstests des DMX-Protokollwandlers

Um die Qualität der implementierten Funktionen beurteilen zu können, wurden einige Tests an der verwendeten Kombination von Hard- und Software ausgeführt. Da die größte Beschränkung der Leistungsfähigkeit des Protokollwandlers in der Prozessorleistung und der maximal möglichen Netzwerkgeschwindigkeit des Raspberry Pi liegt, wurde der Schwerpunkt bei den Tests auf diese Komponente gelegt.

8.1 Test der maximalen Netzwerkgeschwindigkeit

Um die maximale Netzwerkgeschwindigkeit der verwendeten Komponenten und die Qualität der Verbindung zu ermitteln, wurden einige Tests mit dem Open-Source Kommandozeilenprogramm “iperf“ durchgeführt. Dieses ermöglicht das Erzeugen von UDP- und TCP-Datenströmen sowie das anschließende Messen der maximal möglichen Bandbreite. Die grundlegende Funktionsweise des Programms basiert auf einer Server-Client-Struktur. Messungen erfolgen demnach immer zwischen zwei Teilnehmern, dem Client und dem Server. Auf den beiden Hosts läuft jeweils eine Instanz des Programms. Der iperf-Client verbindet sich mit dem iperf-Server¹⁵⁵. Die gemessene Bandbreite entspricht dem Datendurchsatz der Verbindung zwischen den beiden Hosts (siehe Abb. 23).

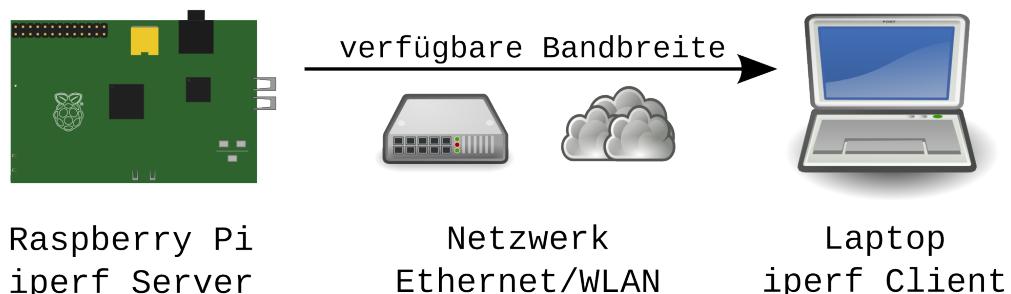


Abbildung 23: schématischer Aufbau einer Messung mit iperf^a

^aGrafik basierend auf Symbolen von <http://www.opensecurityarchitecture.org/cms/library/icon-library>

Beim Aufruf des Programms können über diverse Einstellungsparameter unterschiedliche Tests ausgeführt werden. Dies ermöglicht unter anderem das Versenden von UDP oder TCP Paketen mit einstellbarer Paketgröße und die Auswahl zwischen uni- und bidirektionalem Netzwerkverkehr.

Bei dem in Abb. 24 ersichtlichem Test, wurde die maximale Bandbreite der Ethernet-Verbindung vom Client zum Server mittels des TCP-Protokolls gemessen. Dazu wurde ein zweiter Computer mit dem EtherCon-Anschluss an der Vorderseite des Geräts über ein 7,5 m langes Cat 5e Kabel verbunden. Der dazu verwendete Laptop ist deutlich leistungsfähiger als der im Protokollwandler verbaute embedded Computer, sodass davon

¹⁵⁵Vgl. <http://wiki.ubuntuusers.de/iperf> Stand: 20.06.2013

ausgegangen werden kann, dass Einbrüche der Netzwerk-Performance auf den Raspberry Pi zurückzuführen sind.

Alle in den folgenden Tabellen dargestellten Werte bilden jeweils den Mittelwert von zehn aufeinander folgenden Messungen mit einer Länge von jeweils 10s. Die Pfeile in den Tabellen beschreiben die Richtung des Signalflusses im Netzwerk.

Bandbreite Ethernet				
Messungs-Nr.	Inhalt	Protokoll	Art der Verbindung	gemittelte Bandbreite in Mbit
1	unidirektionale Bandbreite	TCP	Raspberry Pi (Client) → Laptop (Server)	78,37
2	unidirektionale Bandbreite	TCP	Raspberry Pi (Server) ← Laptop (Client)	94,5
3	bidirektionale Bandbreite	TCP	Raspberry Pi (Server) ↔ Laptop (Client)	90,97 (C → S) 43 (C ← S)

Abbildung 24: Messergebnisse iperf - maximaler Datenfluss Ethernet

Bei der Messung des maximalen Datenflusses durch iperf, wird der Strom der Pakete vom Client zum Server gemessen. In den Messungen 1, 2 und 3 ist ersichtlich, dass die Bandbreite des Raspberry Pi im Client-Modus deutlich geringer gegenüber dem Server-Modus ist. Eine zusätzliche Messung der Prozessorlast während des Tests zeigte, dass das Senden der Daten signifikant rechenintensiver ist als das Empfangen. Die Prozessorlast stieg während des Empfangs auf bis zu 55% an, beim Senden wurde dagegen die maximale Auslastung des Systems erreicht. Die dritte Messung wurde mit bidirektionalem Datenverkehr durchgeführt. Das bedeutet, dass die Daten von den Teilnehmern gleichzeitig gesendet und empfangen werden müssen. Dabei verringerten sich die Maximalwerte zwar deutlich, die erreichte Geschwindigkeit, war dennoch akzeptabel.

Die erzielten Werte der maximalen Bandbreite sind jedoch sehr positiv zu werten. Eine Übertragungsrate von bis zu 94,5 Mbit genügt, abgesehen von wenigen Ausnahmefällen, heutigen Ansprüchen der Lichtsteuerung¹⁵⁶.

Neben den Messungen der Ethernet-Strecke, wurde zusätzlich auch die WLAN-Verbindung zum eingebauten Router überprüft. Diese Messungen wurden in zwei Szenarien unterteilt. In Szenario 1 befand sich der mit dem WLAN des Routers verbundene Laptop in ca. 5m Entfernung ohne störende Elemente zwischen den Geräten. In Szenario 2 wurde die Entfernung auf ca. 10m erhöht. Zusätzlich befand sich eine ca. 30cm starke Gipskarton-Wand zwischen den beiden Netzwerkteilnehmern. Die aufgebaute Verbindung des Laptops mit dem Router wurde in den Netzwerkeinstellungen mit 150 Mbit/s angegeben. Eine erste Messung (Messungs-Nr.4) unter „Idealbedingungen“ (Szenario 1) einer WLAN-Verbindung zeigte bereits einen deutlichen Unterschied, gegenüber der Kabelverbindung. Die verfügbare Bandbreite entsprach nur noch einem Bruchteil der verkabelten Variante. Dies bestätigte sich auch in der Messung Nr. 5 und 6 (Szenario

¹⁵⁶für die Netzwerkbelaistung während einer Lichtsteuersignal-Übertragung siehe Kap. 8.2

Bandbreite WLAN – Szenario 1

Messungs-Nr.	Inhalt	Protokoll	Art der Verbindung	gemittelte Bandbreite in Mbit
4	unidirektionale Bandbreite	TCP	Raspberry Pi (Client) → Laptop (Server)	7,32
5	unidirektionale Bandbreite	TCP	Raspberry Pi (Server) ← Laptop (Client)	13,5
6	bidirektionale Bandbreite	TCP	Raspberry Pi (Server) ↔ Laptop (Client)	4,8 (C → S) 2,44 (C ← S)

Bandbreite WLAN – Szenario 2

Messungs-Nr.	Inhalt	Protokoll	Art der Verbindung	gemittelte Bandbreite in Mbit
7	unidirektionale Bandbreite	TCP	Raspberry Pi (Client) → Laptop (Server)	2,55
8	unidirektionale Bandbreite	TCP	Raspberry Pi (Server) ← Laptop (Client)	5,57
9	bidirektionale Bandbreite	TCP	Raspberry Pi (Server) ↔ Laptop (Client)	2,29 (C → S) 1,24 (C ← S)

Abbildung 25: Messergebnisse iperf - maximaler Datenfluss WLAN

2), bei denen sich der maximale Durchsatz weiter verringerte.

Die bidirektionale Übertragung führte im WLAN-Netz zu einer größeren Verringerung der maximalen Bandbreiten, als im Ethernet-Netzwerk. Dies ist darauf zurückzuführen, dass sich die Teilnehmer im WLAN ein gemeinsames Übertragungsmedium (die Luft) teilen müssen, während bei der Kabelübertragung separierte Hin- und Rückleitungen zur Verfügung stehen.

Bei den beiden Testreihen zeigte sich, dass der verbaute WLAN-Router nur geringen Ansprüchen gerecht werden kann. Dies war jedoch auch aufgrund des günstigen Preis (22,95,-€) zu erwarten, welcher das Hauptauswahlkriterium darstellte.

Aus den Ergebnissen der maximal möglichen Bandbreite zeigt sich, dass der WLAN-Router zwar für einen Zugriff auf das Webinterface zur Konfiguration des Protokollwandlers sowie für die Übertragung von nur wenigen Universen ausreichend ist, der Einsatz für komplexere Lichtsteuer-Aufgaben jedoch vermieden werden sollte.

Die niedrigen Testwerte könnten jedoch auch auf einen Produktfehler des verwendeten Geräts zurückzuführen sein. Dies konnte aufgrund des Mangels eines Referenzgeräts nicht ausgeschlossen werden.

Die reine Bandbreite gibt jedoch noch keinen endgültigen Aufschluss darüber, wie hoch die Qualität der Anwendungen im Netz ist. Weitere Qualitätsmerkmale zur Netzwerkbewertung sind die Verlustrate, also die Quote der verlorenen Pakete während der Übertragung und der sog. „Jitter“. Dieser gibt die Schwankung bei der Übertragungsverzögerung oder den Paketankunftszeiten an¹⁵⁷. Ein erhöhter Jitter ermöglicht das Verringern der Verlustmenge von Paketen, indem diese beim Empfänger zwischengespeichert werden.

¹⁵⁷Vgl. [TW12] S. 466

Jitter ist ein besonders wichtiges Bewertungsmerkmal für die Anwendung der Lichtsteuerung, da dieser Verzögerungen im Netzwerk deutlich erhöhen kann¹⁵⁸.

Zur Messung von Paketverlusten und Jitter wird das UDP Protokoll verwendet. Der grundlegende Unterschied zwischen UDP- und TCP-Verbindungen ist, wie in Kapitel 3.1.4 angedeutet, dass bei der Übertragung von TCP-Paketen im Netzwerk, überprüft wird, ob die Pakete erfolgreich beim Empfänger angekommen sind, während bei UDP-Übertragungen ein Versand ohne anschließende Kontrolle erfolgt. Das Fehlen der Ankunfts-Überprüfungen, ermöglicht eine “schnellere“ Übertragung, als bei TCP und somit eine höhere Bandbreite, da Neuübertragungen von Paketen entfallen. In Abbildung 26 sind die Ergebnisse der UDP-Tests ersichtlich.

UDP-Tests zur Ermittlung von Jitter und Paketverlust

Messungs-Nr.	Inhalt	Protokoll	Art der Verbindung	Mittelwerte		
				Bandbreite in Mbit	Jitter in ms	Verlustrate in %
10	Jitter und verlorene Pakete – Ethernet	UDP	Raspberry Pi (Server) → Laptop (Client)	95,73	0,16	0
11	Jitter und verlorene Pakete – WLAN Szenario 1	UDP		16,6	2,27	0
12	Jitter und verlorene Pakete – WLAN Szenario 2	UDP		6,04	14,94	0

Abbildung 26: Messergebnise iperf - Jitter und Datenverlust

Alle Tests wurden mit dem Raspberry Pi als Server und dem Laptop als Client durchgeführt.

Wie erwartet, erhöhte sich die Bandbreite durch das UDP Protokoll geringfügig. Dies ist vermutlich auch einer der Gründe, warum die aktuellen lichttechnischen Ethernet-Standards, ArtNet und sACN, auf UDP basieren.

Die Verlustrate, also die Anzahl der verlorenen Pakete, betrug bei den Messungen 10, 11 und 12 durchgängig 0%.

Die Mittelwerte des Jitters, machen deutlich, warum eine Verwendung von herkömmlichen Wireless-Equipment aus dem Bereich der Netzwerktechnik in der Anwendung der Lichtsteuerung nur wenig Anwendung findet. Obwohl im Ethernet-Betrieb der Mittelwert des Jitters bei sehr guten 0,16ms lag, stieg dieser im WLAN Testszenario 2 bereits auf 14,94ms an. Dies würde eine deutliche Verzögerung der Laufzeit innerhalb der Signalkette zur Folge haben. Der Einsatz besserer Wireless-Hardware, könnte dieser Problematik vermutlich positiv entgegenwirken.

Bei zeitkritischen und qualitativ anspruchsvollen Anwendungen sollte daher stets auf die Ethernet-Technologie oder auf besser geeignete Wireless Systeme zurückgegriffen werden.

¹⁵⁸Vgl. [TW12] S. 467

8.2 Prozessorlast in Abhangigkeit der ubertragenen Signale

Da der Test des maximalen Netzwerkdurchsatzes sehr synthetisch ist, wird ein zweiter, fur die Praxis relevanterer Test angewandt. Dabei wird die Prozessorlast des Raspberry Pi in Abhangigkeit der gewandelten Steuersignale gemessen.

Es wird der gleiche Aufbau wie im zuvor beschrieben Test verwendet. Auf dem Laptop lauft eine Instanz von OLA, welche als Sender, auf dem Raspberry Pi eine davon unabhangige zweite Instanz, welche als Empfanger agiert. Vor dem Beginn der Messungen wurden am Sender vier Universen mit ArtNet-Outputs und beim Empfanger vier Universen als ArtNet-Inputs und gleichzeitig als sACN-Outputs definiert. In der OLA-Instanz des Raspberry Pi findet also ein Empfang und eine Wandlung der ankommenden ArtNet-Datenstrome statt. Die Ubertragung der ArtNet-Signale erfolgt im Unicast-Betrieb.

Allein das definieren eines Ausgangs in der Open Lighting Architecture reicht jedoch nicht aus, um tatsachlich Signale zu versenden. Dazu ist es notwendig, ein zusatzliches Programm zur Signalerzeugung heranzuziehen. Das fur diesen Test verwendete Programm ist ein, auf der Python-API aufbauendes, Skript¹⁵⁹, welches alle 512 DMX-Kanale durchgehend im Wertebereich von 0-255, mit der minimalen Periodendauer von 22,67ms (entspricht einer Wiederholrate von ca. 44Hz) fadet.

Fur die Messung der CPU-Last wird das in nahezu jedem Linux-Betriebssystem vorhandene Tool “top” verwendet¹⁶⁰. Dieses fuhrt 30 aufeinander folgende Messungen im Intervall von 2 Sekunden durch. Anschlieend werden die Ergebnisse nach dem Prozess “olad¹⁶¹“ gefiltert, um festzustellen, wie gro die Ressourcen-Belastung durch die Open-Lighting Architecture ist.

Die erste Messung erfolgte im Idle, also im Ruhezustand ohne gesendete Daten. Vor dem Start der zweiten Messung wurde zunachst das Python-Skript zum Senden der ArtNet-Signale fur Universum 1 gestartet. Dieses Vorgehen erfolgte analog bei den darauf folgenden drei Messungen. Die Ergebnisse dieser Messungen sind in Abb. 27 ersichtlich.

Aus den Ergebnissen der Tests ist erkennbar, dass die CPU-Last im Ruhezustand ca. 1% betragt und fur jedes von ArtNet zu sACN gewandelte Universum um ca. 5% zunimmt. Die Mittelwerte der finf Messungen sind 1,2%, 5,6%, 11,35%, 14,9% und 17,4%. Der leichte Abfall der Prozessorlast-Zunahme ist durch Mess-Abweichungen zu erklaren. Bei weiteren Messungen waren teilweise Abweichungen bei der 4. und 5. Messung von ca. 2% zu erkennen.

Eine Uberprufung der Netzwerklast, wahrend des Tests zeigte, dass fur jedes hinzugefigte Universum ca. 250kBit/s an zusatzlichem Netzwerktraffic entstanden. Dies entspricht sehr genau der maximalen Ubertragungsrate des DMX512-Standards. Dieser Wert deckt

¹⁵⁹das verwendete Skript ist ebenfalls unter dem Github-Repository unter <https://github.com/Astreubel/Bachelorthesis> verfugbar

¹⁶⁰Vgl. <http://wiki.ubuntuusers.de/top> Stand: 20.06.2013

¹⁶¹Anm. Hauptprozess der Open Lighting Architecture

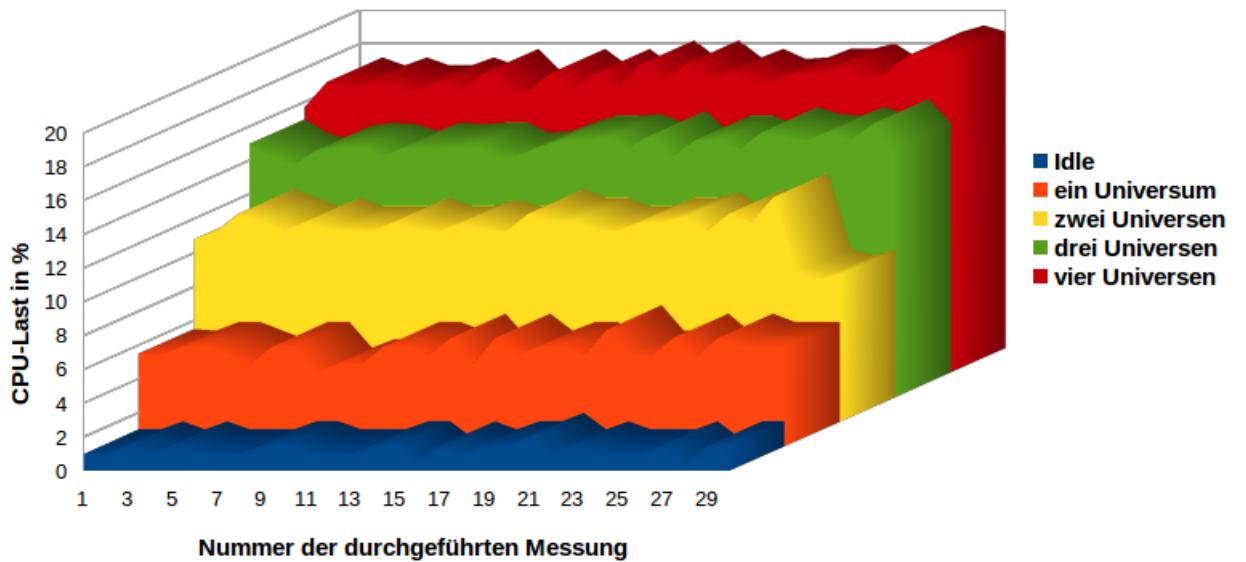


Abbildung 27: CPU-Last “olad“ in Abhängigkeit der gewandelten Signale

sich auch mit den angegebenen Übertragungskapazitäten im ArtNet-Standard. Der Standard gibt an, dass die maximale Anzahl zu übertragender Universen bei einer 100BaseT-Infrastruktur, bei ca. 400 liegt¹⁶². Dies entspricht den eben erwähnten 250kBit/s pro Universum. Eine Weiterführung des Tests bis zur maximalen Belastungsgrenze ist aufgrund der ArtNet-Implementation der Open Lighting Architecture nicht möglich, da diese sich strikt an den ArtNet Protokollstandard hält, welcher eine maximale Anzahl von vier Universen für einen Empfänger-Node definiert¹⁶³.

Zusammenfassend lässt sich jedoch zu den durchgeführten Tests sagen, dass sowohl die verfügbare Netzwerk-Bandbreite, als auch die Prozessorleistung heutigen und zukünftigen Ansprüchen an einen Ethernet-Node genügen.

¹⁶²Vgl. [Art12] S. 1

¹⁶³Vgl. [Art12] S. 23

8.3 Produkttests für einen kommerziellen Vertrieb

Wird der kommerzielle Vertrieb eines Produktes angestrebt, müssen zahlreiche Tests durchgeführt werden um Sicherheit und Funktionalität gewährleisten zu können. Die im Rahmen eines solchen Vertriebs notwendigen Tests unterscheiden sich jedoch deutlich, von denen in dieser Arbeit umsetzbaren Untersuchungen.

Für den hier vorgestellten Signalwandler würde dies in erster Linie Funktionstests der Hard- und Software sowie elektrotechnische Zertifizierungen umfassen.

Die Tests zur Überprüfung der beschriebenen Funktionen sind notwendig, um die Qualität der eigenen Produkte bewerten zu können. Sie stellen somit einen elementaren Bestandteil der späteren Kundenzufriedenheit dar. Neben den Standardfunktionen müssen vor allem Sonder- und Spezialfälle abgebildet und überprüft werden. Dies würde bspw. das Senden und Empfangen von nicht standardkonformen Paketen und Signalen umfassen.

Die weiteren Funktionstests des hier vorgestellten Protokollwandlers müssten somit neben einer Überprüfung der Einzelfunktionen der Open Lighting Architecture auch Randbereiche, wie bspw. die Fehlbedienung durch den Endnutzer und unsachgemäße Handhabungen abdecken.

Ergänzend müsste sichergestellt werden, dass das Gerät den Regeln der Technik und somit den geltenden Anforderungen deutscher Normen und den EU-Richtlinien gerecht wird. Die Erfüllung dieser Richtlinien ist Voraussetzung für den Erhalt einer CE-Kennzeichnung¹⁶⁴, welche angibt, ob ein Produkt allen gesetzlichen Anforderungen gerecht wird. Die zwei relevantesten EU-Richtlinien für elektronische Geräte sind 2004/108/EG (Richtlinie über die elektromagnetische Verträglichkeit) und 2006/95/EG (Niederspannungsrichtlinie). Jeder Hersteller ist verpflichtet, alle notwendigen Produktrichtlinien auf sein Gerät anzuwenden¹⁶⁵ und eine somit eine Einhaltung der Sicherheitsstandards zu gewährleisten.

¹⁶⁴Vgl. <https://de.wikipedia.org/wiki/CE-Kennzeichnung> Stand: 28.06.2013

¹⁶⁵Vgl. <http://www.din.de/cmd?level=tpl-unterrubrik&menuid=47421&cmsareaid=47421&menurubricid=47429&cmsrubid=47429&menusubrubid=47435&cmssubrubid=47435> 28.Juni 2013

9 Fazit und Ausblick

9.1 Ergebnis der Arbeit

Diese Arbeit hatte die Konzeption sowie die Realisierung eines universellen DMX512-Ethernet Protokollwandlers zum Ziel. Dabei galt es sich mit den folgenden Problemstellungen auseinanderzusetzen:

- Auswahl der Einzelmodule und Definierung der Soft- und Hardwareschnittstellen des Gesamtsystems
- Konstruktion eines veranstaltungstauglichen Endprodukts mit branchenüblichen Steckverbindersystemen
- Auseinandersetzung mit der modernen Netzwerktechnik und den relevanten lichttechnischen Standards (ANSI E1.11, E1.17, E1.20, E1.31 und DIN 56930-2)
- Zusammenfassung aller notwendigen Grundlagen zum Verstehen des Systems und moderner Lichtsteuersysteme im Allgemeinen
- Konzeption und Herstellung eines, auf die Anforderungen des Protokollwandlers angepassten, Erweiterungsboards für den Raspberry Pi
- Auseinandersetzung mit der Programmiersprache Python zur Nutzung der Python-API der Open Lighting Architecture
- Entwicklung der Software zur Ausgabe von Statusinformation der Open Lighting Architecture auf einem 16x2 Zeichen LCD-Display
- Analyse der bereits am Markt erhältlichen Systeme
- Fertigung und Zusammenbau des Systems
- Überprüfung der Leistungsfähigkeit des konzipierten Systems in dafür entwickelten Tests

Die Aufgabenstellung ist, wie an der Auflistung zu erkennen sehr umfangreich und tangiert Bereiche der Veranstaltungstechnik, Elektrotechnik und der Informatik.

Im Ergebnis ist ein lauffähiges Produkt entstanden, welches abgesehen von dem noch nicht implementierten analogem Input, alle Funktionen der vorher definierten Zielsetzung erfüllt. Es bietet neben einigen Nachteilen, bspw. der teilweise wenig endanwendерfreundlichen Bedienung, sogar einige relevante Vorteile gegenüber Konkurrenzprodukten. Dazu zählen die breite Protokollunterstützung, der günstige Preis, die modulare Bauweise (einfache Anpassung an die tatsächlichen Bedürfnisse des Kunden), die WLAN-Funktionalität sowie die freie Verfügbarkeit aller technischen Dokumentationen und der verwendeten Software.

9.2 Erkenntnisse und zukünftige Verbesserungen

Da ein solches technisches System aus vielen Teilgebieten besteht, ergibt sich ein sehr komplexes Projekt. Durch das gleichzeitige Abarbeiten vieler ineinander greifender Problemstellungen, fällt es schwer für eine Einzelperson alle Bereiche abzudecken. Speziell der Software-Bereich hätte, für die Implementation der zu Beginn angestrebten Funktionalitäten, bspw. der Verknüpfung des analogen Inputs mit OLA, noch deutlich mehr Zeit in Anspruch genommen. Die Komplexität einzelner Fragestellungen, speziell die Konzeption des Gehäuses, wurde teilweise unterschätzt.

Verbesserungen des Systems können in vielen Bereichen erfolgen. Zukünftige Weiterentwicklungen sollen die Implementierung des analogen Eingangs, die Entwicklung RDM-fähiger DMX-Ports und somit evtl. auch den Wechsel auf eine andere embedded Hardware Plattform zum Ziel haben. Eine interessante Hardwareplattform für zukünftige Entwicklungen wäre der Beaglebone Black. Eine weitere interessante Option ist die Erweiterung des Raspberry Pi durch die Arduino Plattform. Diese zwei Systeme können bspw. über die UART-Schnittstelle miteinander verbunden werden, um die Anschluss- und Ausgabemöglichkeiten des Gesamtsystems zu erweitern. Die Installation eines beseren Routers und die Installation des offenen Betriebssystems OpenWrt zur Erhöhung der Performance und des Funktionsumfangs ist ebenfalls erstrebenswert.

Zur Bewertung der Stabilität des Gesamtsystems sind des weiteren zusätzliche Tests erforderlich.

9.3 Ausblick

Trotz der Komplexität dieses Projekts, sind die in dieser Arbeit gesammelten Erkenntnisse aufeinander aufbauend und leicht verständlich. Sie bieten somit Personen ohne jegliche Programmiererfahrung einen guten Anlaufpunkt, in der Anwendung von Software zur Umsetzung eigener Ideen und Vorstellungen. Eine Erweiterung des Funktionsumfangs ist auf einfache Weise durch die anfängertaugliche Programmiersprache Python möglich.

Auch nach Abschluss dieser Arbeit, soll aktiv weiter an diesem Projekt gearbeitet werden. Die Projektfortschritte werden im dafür angelegten Github-Repository, unter der Url: <http://astreubel.github.io/Bachelorthesis/> frei zugänglich verwaltet.

Die zukünftige Entwicklung der professionellen Lichtsteuerung ist bisher nur zu erahnen. Eine weitere Beobachtung, speziell in Bezug auf die Verbindung der konventionellen DMX-Systeme mit der modernen Ethernettechnik bietet noch großen Spielraum für innovative Entwicklungen.

Appendix

Kommunikation im OSI-7-Schicht-Modell (Open Systems Interconnection Reference Model)

PC 1 in Netzwerk A (z.B. Client)



Ablauf: PC 1 sendet eine Anfrage (Request) an PC 2, indem diese zunächst vor der eigentlichen Übertragung durch Hinzufügen der Schichtenheader-/trailer formatiert wird. PC 2 empfängt den Request von PC 1 und nimmt die Schichtenheader-/trailer wieder aus der Nachricht, bis nur noch die Anwendungsdaten (innerste Bits) vorhanden sind und verarbeitet diese in der Endanwendung. Die Antwort (Response) läuft analog zur Übertragung der Anfrage, bloß in umgekehrter Richtung ab.

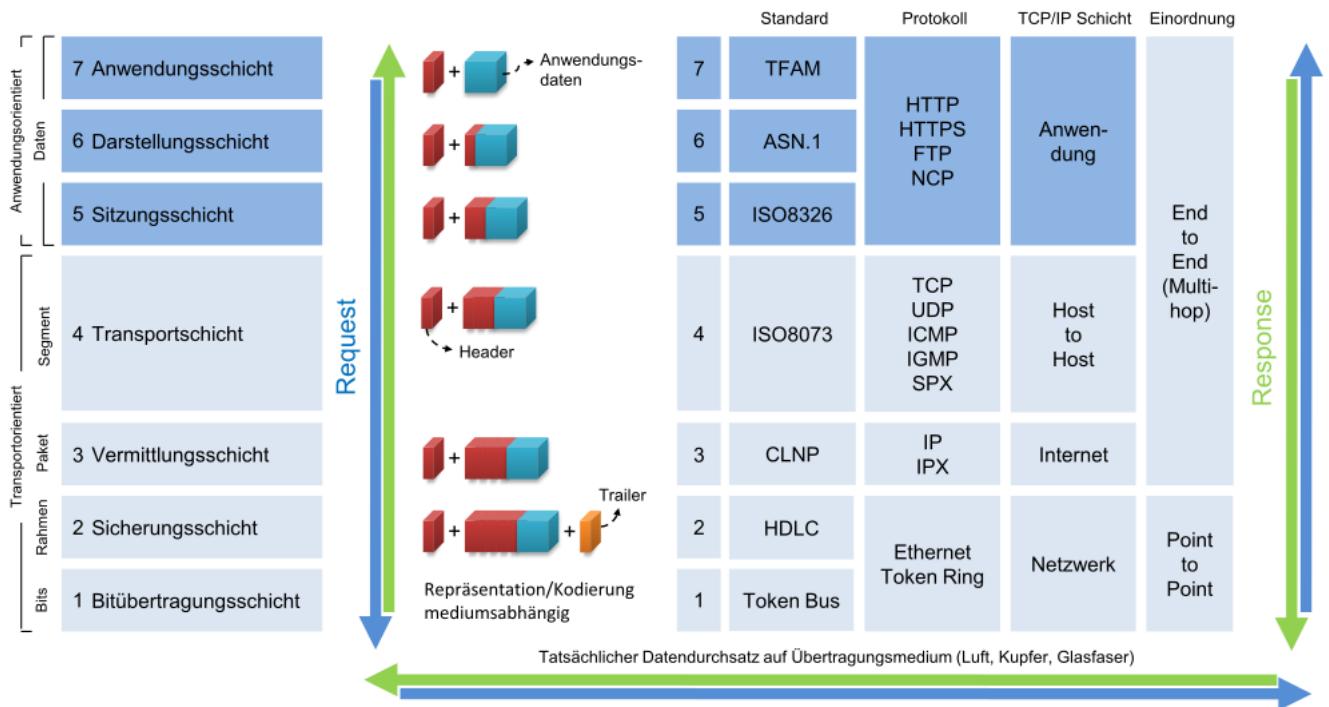


Abbildung 28: detailliertes OSI-Schichtenmodell

Quelle: <https://de.wikipedia.org/wiki/OSI – Modell> Stand: 08.06.2013

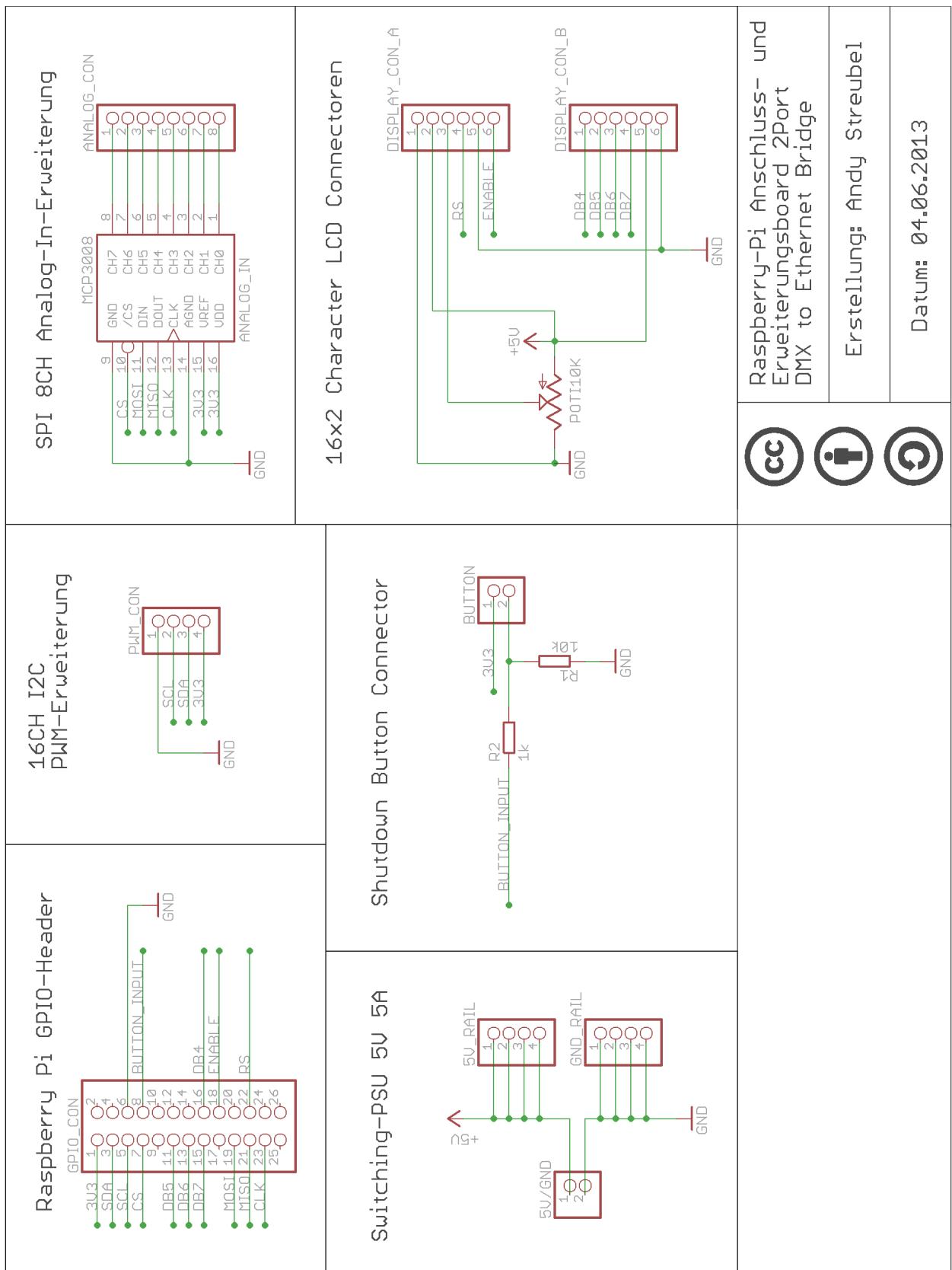


Abbildung 29: Schaltplan Anschluss- und Erweiterungsplatine



Abbildung 30: Fotografie Gehäuseprototyp - Gesamtansicht vorn



Abbildung 31: Fotografie Gehäuseprototyp - Gesamtansicht hinten

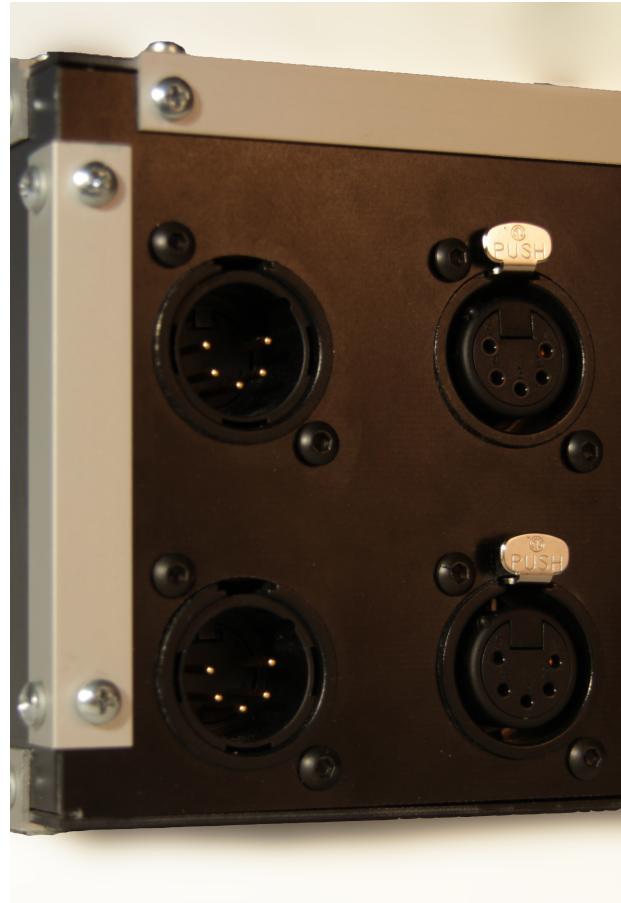


Abbildung 32: Fotografie Gehäuseprototyp - Detailansicht DMX-Ports

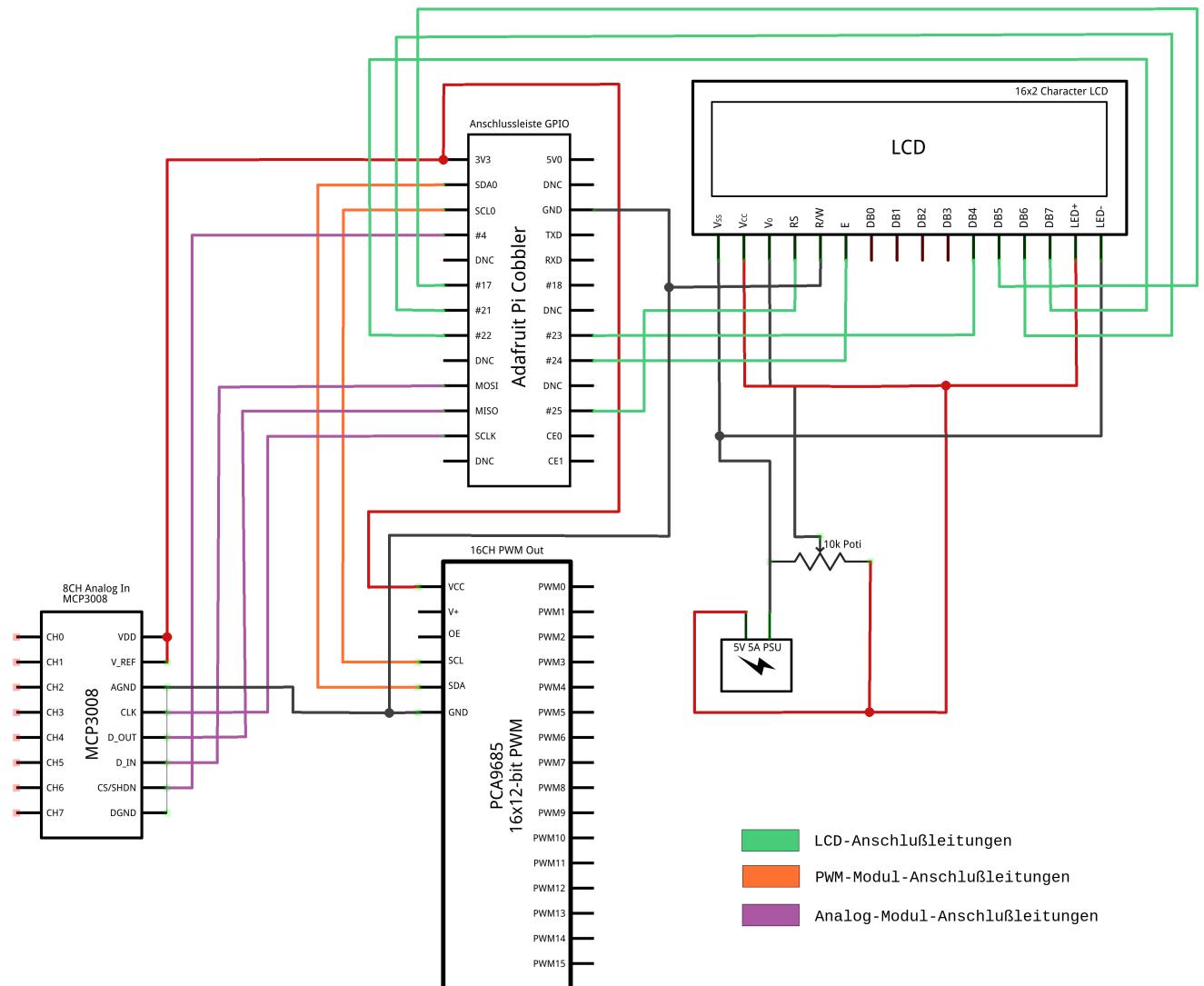


Abbildung 33: Verdrahtungsplan der Erweiterungsmodule

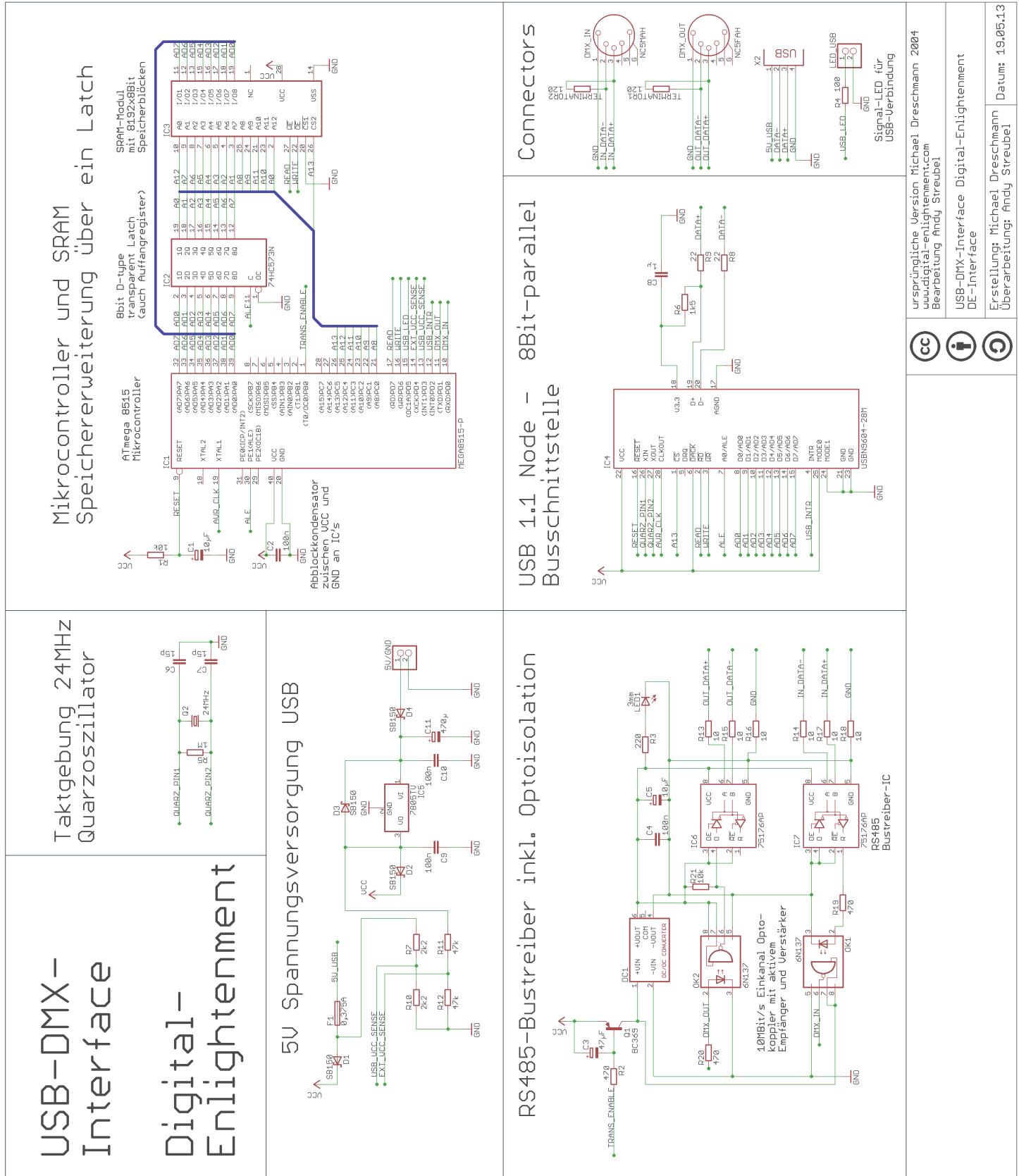
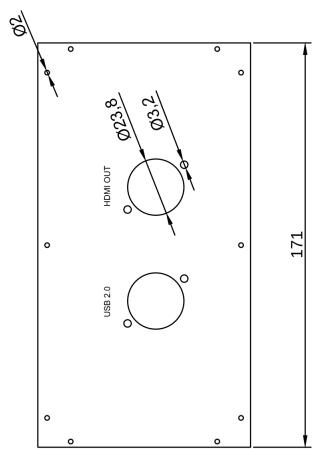


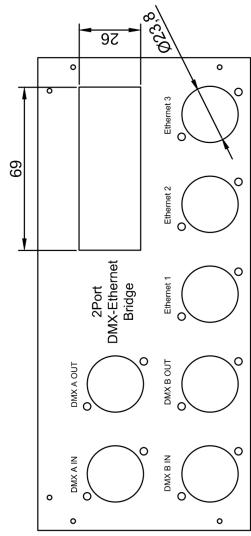
Abbildung 34: Schaltplan Digital Enlightenment USB-DMX-Interface

Seitenpteile

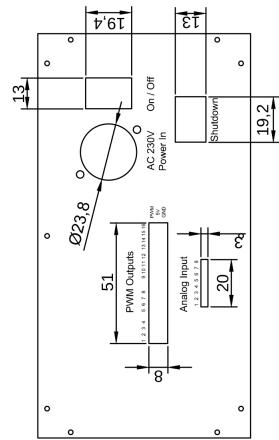
Links



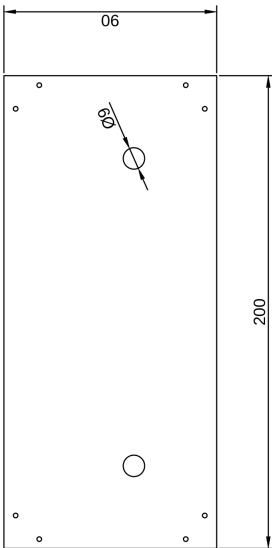
Vorderseite



Rechts



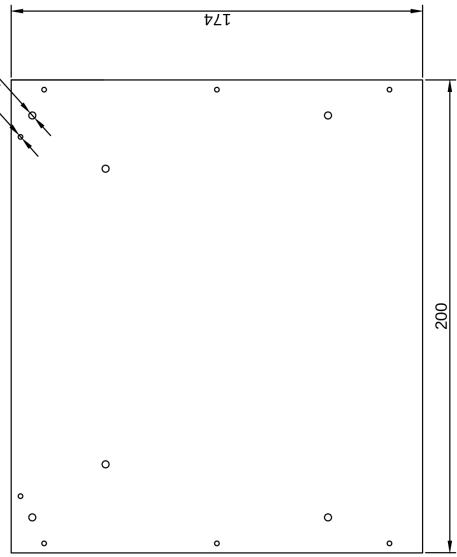
Rückseite



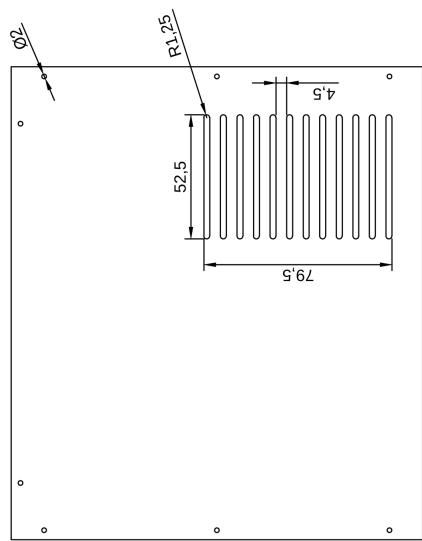
Verantwortliche Abt. Fachber. VII	Technische Referenz	Dokumentenstatus Fertigungszeichnung	Dokumentenstatus Bereit für Fertigung
BEETHOVENSCHE FACHSCHULE FÜR TECHNIK BERLIN	Erstellt durch Andy Streubel	Titel, zusätzlicher Titel Gehäuse DMX- Ethernet-Wandler	Maßstab 1:2
University of Applied Sciences Matrikel Nr. 769137	Genehmigt von	And/Ausgabedatum 27.05.2013	Sur. Blatt De/1/2

Abbildung 35: technische Zeichnung Gehäuseentwurf Seite 1

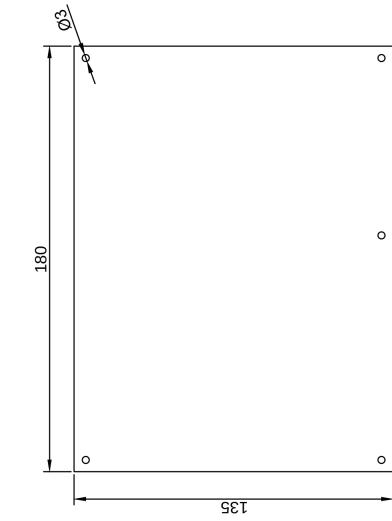
Bodenplatte



Deckel



Einlegeplatten



Verantwortliche Abt. Fachber. VIII	Technische Referenz ERSTELLTE DURCH BEUTH-HOCHSCHULE FÜR TECHNIK BERLIN Andy Streubel	Dokumentenart Fertigungszeichnung Titel, zusätzlicher Titel Gehäuse DMX- Ethernet-Wandler	Dokumentenstatus Bereit für Fertigung
	Genehmigt von Matrikel-Nr. 769137		Maßstab 1:2 And Ausgabedatum Ispr. Blatt 27.05.2013 De 2/2

Abbildung 36: technische Zeichnung Gehäuseentwurf Seite 2

Literatur

- [ANS08] ESTA, *ANSI E1.11-2008 Asynchronous Serial Digital Data Transmission Standard for Controlling Lighting Equipment and Accessories*. 2008
- [ANS09] ESTA *ANSI E1.31-2009 Lightweight streaming protocol for transport of DMX512 using ACN*. 2009
- [ANS10] PLASA *ANSI E1.20-2010 Remote Device Management Over DMX512 Networks*. 2010
- [Ant04] ANTONUK, Dan: *New Protocols Up and Running*. Version: 2004. <http://www.rdmprotocol.org/files/Protocol%20Winter%202004%20New%20Protocols.pdf>, Abruf: 28.05.2013
- [Art12] *Artistic License - Art-Net 3 Ethernet Communication Standard*. 2012
- [But06] BUTLER, Javid: *DMX versus Ethernet infrastructures for lighting control networks*. Version: 2006. <http://www.rdmprotocol.org/files/Protocol%20Spring%202006%20DMX%20vs%20Ethernet.pdf>, Abruf: 28.05.2013
- [DIN00] *DIN 56930-2 Bühnentechnik-Bühnenlichtstellsysteme Steuersignale*. 2000
- [Dre13] DRESCHMANN, Michael: *Digital Enlightenment USB-DMX Interface*. Version: 2013. <http://www.digital-enlightenment.de/usbdmx.htm>, Abruf: 16.06.2013
- [Hoe08] HOELSCHER, Hendrik: *AN017:Remote Device Management [Responder]*. Version: 2008. <http://www.hoelscher-hi.de/hendrik/light/ressources/an017.pdf>, Abruf: 28.05.2013
- [Hä11] HÄBERLEIN, Tobias: *Eine praktische Einführung in die Informatik mit BASH und Python*. München : Oldenbourg Wissenschaftsverlag GmbH, 2011. – ISBN 978–3–486–70423–5
- [Kel10] KELLER, Max: *Faszination Licht - Licht auf der Bühne*. 4. aktualisierte Auflage. Prestel Verlag, 2010. – ISBN 978–3791343723
- [New12] NEWTON, Simon: *RDM discovery: How does it work?* Version: 2012. http://www.lightingandsoundamerica.com/mailing/PLASAProtocol/PSpring12_RDMCornerCases.pdf, Abruf: 28.05.2013
- [New13] NEWTON, Simon: *RDM responder testing with the Raspberry Pi*. Version: 2013. http://www.lightingandsoundamerica.com/mailing/PLASAProtocol/PWinter13_RDMRaspberryPi.pdf, Abruf: 19.06.2013

- [Ric09] RICK, Matthias: *Konzeption und Implementierung eines Firmware-Update-Systems in einem Embedded System über das RDM-Protokoll*, HTWK Leipzig, Diplomarbeit, August 2009. <http://www.imn.htwk-leipzig.de/~mrick/Bachelorarbeit%20-%20mrick.pdf>
- [rpi13] *RPi Serial Connection.* Version: 2013. http://elinux.org/RPi_Serial_Connection, Abruf: 04.06.2013
- [SOU99] SOUNDLIGHT: *DMX-512A - ein neues Protokoll?* Version: 1999. <http://www.soundlight.de/techtips/dmx512/dmx2000a.html>, Abruf: 27.05.2013
- [TW12] TANENBAUM, Andrew S. ; WETHERALL, David J.: *Computernetzwerke*. 5. aktualisierte Auflage. München : Pearson Education Deutschland GmbH, 2012. – ISBN 978–3–86894–137–1
- [USI90] *USITT DMX512/1990 - Digital Data Transmission Standard for Dimmers and Controllers.* 1990
- [Zei11] ZEINER, Ulrich: *Grundlagen der Netzwerktechnik*. expert Verlag, 2011. – ISBN 978–3–8169–3057–0