

---

# **ClubDMX Doku**

***Release 0.9***

**Gunther Seiser**

**06.11.2021**



<b>1</b>	<b>README</b>	<b>3</b>
1.1	Online testen . . . . .	3
1.2	Support . . . . .	3
<b>2</b>	<b>Features</b>	<b>5</b>
2.1	Web-Interface . . . . .	5
2.2	Open Lighting Architecture . . . . .	5
2.3	Szenen . . . . .	5
2.4	Daten in CSV-Dateien . . . . .	6
2.5	Angepasst an Benutzer . . . . .	6
2.6	MIDI . . . . .	6
2.7	OSC Input . . . . .	6
<b>3</b>	<b>Erste Schritte</b>	<b>7</b>
<b>4</b>	<b>Einrichtung</b>	<b>9</b>
4.1	ClubDMX und OLA . . . . .	9
4.2	Verzeichnisse . . . . .	10
4.3	Raum einrichten . . . . .	10
4.4	Config . . . . .	11
4.5	Stage einrichten . . . . .	15
4.6	Fader einrichten . . . . .	16
4.7	Buttons einrichten . . . . .	18
4.8	Exec-Seite einrichten . . . . .	20
4.9	Einrichten von zeitabhängiger Steuerung mit Crontab (in Linux) . . . . .	21
<b>5</b>	<b>ClubDMX benutzen</b>	<b>23</b>
5.1	Stage . . . . .	23
5.2	Exec . . . . .	27
5.3	Fader . . . . .	27
5.4	Button . . . . .	28
5.5	Einrichtung . . . . .	28
5.6	Info . . . . .	29
5.7	Benutzer-Datenbank . . . . .	29
5.8	OSC Input . . . . .	30
<b>6</b>	<b>Benutzer</b>	<b>31</b>

6.1	Basic . . . . .	31
6.2	Standard . . . . .	31
6.3	Admin . . . . .	31
6.4	Funktionen . . . . .	32
<b>7</b>	<b>Patch</b>	<b>33</b>
7.1	Neuen Patch erstellen . . . . .	33
<b>8</b>	<b>Grundlagen</b>	<b>35</b>
8.1	DMX . . . . .	35
8.2	Hardware . . . . .	35
8.3	Räume und Config . . . . .	36
8.4	Bearbeitungsmodus . . . . .	36
8.5	HTP und LTP . . . . .	36
8.6	Topcue . . . . .	37
<b>9</b>	<b>Raspberry einrichten</b>	<b>39</b>
9.1	Raspberry Buster neu installieren . . . . .	39
9.2	OLA installieren . . . . .	40
9.3	OLA für ClubDMX vorbereiten . . . . .	41
9.4	ClubDMX installieren . . . . .	42
9.5	Troubles schießen . . . . .	43
<b>10</b>	<b>Entwicklungsschritte</b>	<b>47</b>
10.1	Erste Idee . . . . .	47
10.2	Die Entwicklungsschritte . . . . .	48
<b>11</b>	<b>Impressum</b>	<b>51</b>

Version 0.9



ClubDMX ist eine Lichtsteuerung für kleine bis mittelgroße Anwendungsbereiche: Bars, Probenräume, Ausstellungen, Wohnbereiche. Das Ziel der Entwicklung ist eine auch für Laien einfach zu bedienende Software.

- Läuft auf einem Raspberry PI
- Bedienung über Web-Interface, Midi und OSC.
- Verbindung zu den DMX-Geräten über diverse Hardware (USB-Dongles, Artnet, ...)
- Geräte-Einzelsteuerung und Lichtszenen
- Portable Datenstruktur, alle Daten in CSV-Tabellen
- Angepasst an den Benutzer über Login

## 1.1 Online testen

Das Web-Interface läuft zum Testen online. Hier im Testbetrieb kann die Funktionalität getestet werden. Eine Hardware-Steuerung ist damit natürlich nicht möglich. Es sind verschiedene **Räume** vorbereitet: Eine Bar, ein Probenraum, ein Wohnzimmer und andere Beispiele. Hier gehts zu [ClubDMX online](#) .

## 1.2 Support

Schick mir eine Mail: [gunther.seiser.63@gmail.com](mailto:gunther.seiser.63@gmail.com)





### 2.1 Web-Interface

ClubDMX läuft auf einem Raspberry PI (oder auf einem anderen Computer) und wird durch eine Webseite bedient. Auf dem Raspberry läuft also die Software von ClubDMX und ein Webserver. ClubDMX wird von einem im selben (lokalen) Netzwerk befindlichen Gerät bedient. Das kann ein Tablet, Smartphone oder ein Rechner sein. Es wird dazu keine App benötigt, der Browser genügt.

Die Webseite ist im Responsive Design gestaltet und passt sich an die Größe des jeweiligen Bildschirms an.

### 2.2 Open Lighting Architecture

Die DMX Verbindung erfolgt über [OLA](#). Damit stehen eine Reihe von Hardware-Optionen zur Verfügung, wie zum Beispiel Enttec DMX USB Pro, DMXking. OLA ermöglicht auch diverse Ethernet Protokolle wie ArtNet, sACN und andere. Eine Übersicht über die Möglichkeiten zur DMX-Ausgabe findest du [hier](#).

### 2.3 Szenen

Bei der Entwicklung stand als erstes und grundlegendes Ziel: **Lichtszenen (=Cues)** einfach zu erzeugen und zu bedienen. Genau diese Eigenschaft ist es, die vielen kostengünstigen Lichtsteuerungen aus meiner Sicht fehlt.

**Wie sind also hier in ClubDMX die Möglichkeiten für Lichtszenen?**

- Cues mit Fader.
- Cues mit Button.
- Cue mit Top-Priorität zur Bedienung der einzelnen Geräte und Erzeugung von weiteren Cues. Siehe [Topcue](#)

## 2.4 Daten in CSV-Dateien

Die Programmdateien werden in CSV-Dateien gespeichert. Das sind Textdateien, die mit jedem Editor (Notepad, Excel, Libre Office, ...) bearbeitet werden können. Natürlich ist auch eine Bearbeitung in ClubDMX möglich. Es stehen Zeilen- und Zellen-Bearbeitungsoptionen zur Verfügung.

## 2.5 Angepasst an Benutzer

Indem man sich auf der Webseite anmeldet, werden die benutzerspezifischen Möglichkeiten und Daten aktiv. Ohne Anmeldung ist keine Bedienung der DMX-Geräte möglich.

## 2.6 MIDI

Außer der Webseite kann ClubDMX auch über MIDI bedient werden. Vorzugsweise über ein Korg NanoKontrol-2. Die Buttons und Fader des NanoKontrol2 können mit den Buttons und Fadern verbunden werden.

## 2.7 OSC Input

Über [OSC](#) kann eine Verbindung von einer externen Software zu ClubDMX hergestellt werden. Damit können alle Geräte mit allen Attributen, die Szenen-Fader und Szenen-Buttons bedient werden. Getestet mit [Isadora](#).

---

## Erste Schritte

---

Das ist eine kurze Zusammenfassung der Arbeitsschritte, um ein neu installiertes ClubDMX zum Leben zu erwecken.

- Anmelden
- **Leeren Raum oder bestehenden Raum laden** Einrichtung -> Raum -> neuen Raum anlegen
- Patch nach vorhandener Hardware erstellen:

Patch *\_neu* speichern als <neuer Patchname>

neue Zeile(n) erstellt einen oder mehrere neue Heads. Im Formular die entsprechenden Daten eingeben.

Für weitere Hinweise siehe *Neuen Patch erstellen*

- Config:

Config *\_neu* speichern als <neuer Configname>

Falls ClubDMX und OLA am selben Rechner laufen, dann ist die OLA-Adresse die Default-Adresse 127.0.0.1. Falls nicht, dann in Einrichtung -> Config -> OLA-Adresse die IP-Adresse eintragen. Entsprechend der vorhandenen Hardware die Anzahl der Universen eintragen.

- OLA:

Wenn die OLA IP-Adresse korrekt ist, dann kann über Einrichtung -> Config -> OLA einrichten zur OLA Administration gewechselt werden.

TODO: Seite zur OLA Einrichtung...

Die Verbindung zur Hardware kann in ola.html -> Universes -> <uname> -> DMX Console getestet werden.

- Stage:

Stage *\_neu* speichern als <neuer Stagenamen>

Import Patch legt für jeden in der Stage noch nicht vorhandenen Head ein Symbol an, mit Headnummer, Name und Kommentar aus Patch.

- Heads verwenden, um Cues, Fader und Buttons zu erzeugen.

In Stage -> Single können die Geräte einzeln getestet werden.

In Info -> DMX Output kann das errechnete DMX-Signal angesehen werden.

ClubDMX kommt mit sehr sparsamen Hardware-Komponenten aus. Als Basis-Hardware kann ein Raspberry PI eingesetzt werden, das ist aber nicht zwingend notwendig. ClubDMX läuft auf jeder Hardware, wo Python-3 mit einigen Erweiterungen (die zusätzlich zu Python installiert werden) einsatzbereit ist.

### 4.1 ClubDMX und OLA

Open Lighting Architecture (OLA) ist die Schnittstelle zur Hardware, daher muss auch OLA installiert werden. Zur Einrichtung von OLA findest du [hier weitere Informationen](#). Zur Installation von OLA auf Mac kann ich leider nichts beitragen, ich habe OLA auf dem Raspberry und auf (Debian) Linux installiert und verwendet.

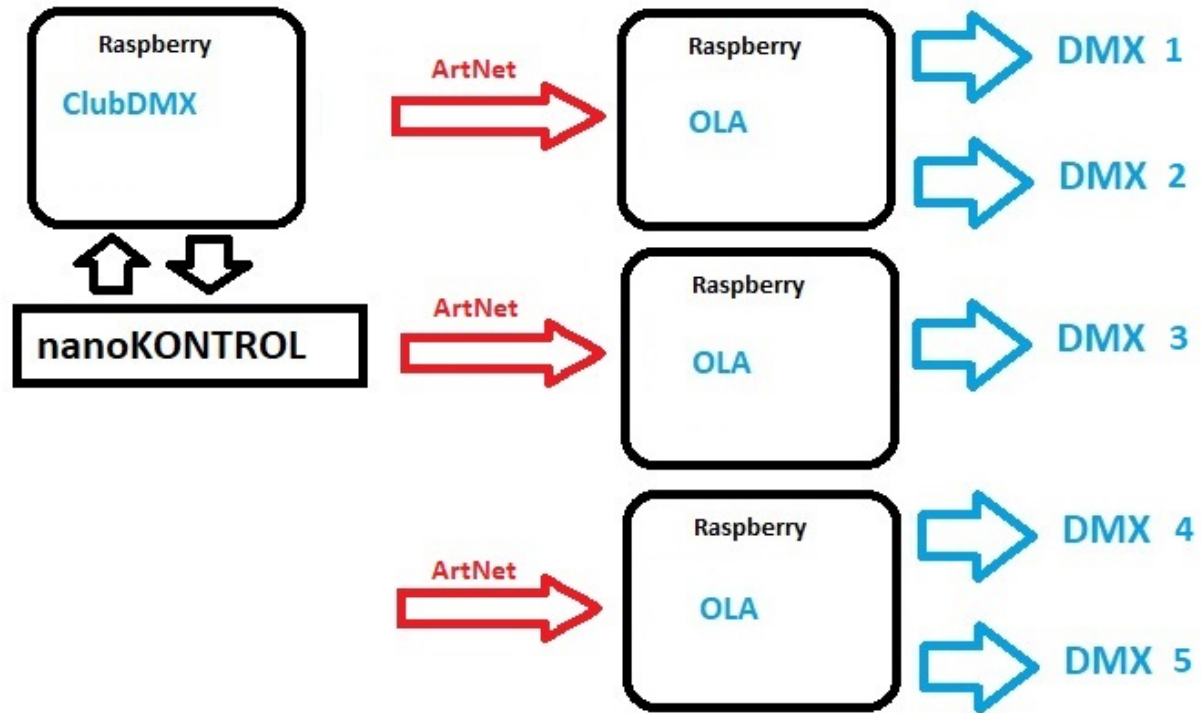
Der große Vorteil von OLA besteht darin, dass damit eine ganze Palette von Ethernet Protokollen und USB/Serial/Netzwerk Geräten zur Anbindung zur Verfügung stehen.

Ein einfacher Hardware-Aufbau mit einem Raspberry PI sieht so aus:



ClubDMX und OLA lassen sich sehr einfach auf mehrere Universen erweitern. Falls ClubDMX in einem größeren Aufbau gedacht ist, können auch mehrere Raspberry PIs eingesetzt werden. Oder es können an einen Raspberry PI mehrere USB DMX Adapter angesteckt und für verschiedene Universen konfiguriert werden.

Ein größerer Hardware-Aufbau mit mehreren Raspberry PIs könnte so aussehen:



## 4.2 Verzeichnisse

Die Verzeichnisstruktur besteht aus zwei Ordnern, einem Code-Ordner und einem Raum-Ordner. Für diese Ordner gibt es Vorgabe-Namen, nämlich `clubdmx_code` und `clubdmx_rooms`.

Über die *Environment*-Variablen `CLUBDMX_CODEPATH` und `CLUBDMX_ROOMPATH` können auch andere Verzeichnisse festgelegt werden.

## 4.3 Raum einrichten

Die Einrichtung eines Raumes erzeugt im Raum-Verzeichnis (`clubdmx_rooms`) einen neuen Ordner. Hier werden alle projektspezifischen Daten gespeichert: Patch, Cues, Fader, Buttons, Stage.

Es gibt einige Möglichkeiten zur Erzeugung eines neuen Raums.

- Einen neuen Raum anlegen,
- einen bestehenden Raum unter einem anderen Namen sichern,
- einen auf USB-Stick gespeicherten Raum laden (restore),
- einen gezippten Raum hochladen.

Raum Config Bedienelemente Midi Datenbank Admin

## Raum Management

Eine [Übersicht zur Raum-Organisation](#) findest du hier.

### Aktueller Raum

Du bist im Raum **jazzitbar**.

[Raum wechseln](#)

Nach dem Wechsel des Raums muss noch eine Konfiguration geladen werden.

Aktuelle Config: **jazzitbar**

[Config öffnen](#)

### Raum Aktionen

[Raum umbenennen](#)

[Raum sichern als...](#)

[neuen Raum anlegen](#)

[Raum löschen](#)

[Backup auf USB](#)

[Restore von USB](#)

[neuen Raum auf USB anlegen](#)

[Raum herunterladen](#)

[Raum hochladen](#)

## 4.4 Config

In einer Config sind die im Einsatz befindlichen Hardware-Komponenten, das Fader- und Button-Layout, das Stage-Layout und weitere Einstellungen gespeichert. In einem Raum können auch mehrere Configs vorhanden sein, zum Beispiel um unterschiedliche Theaterprojekte zu speichern. Der Zusammenhang von Raum und Config wird in [Räume und Config](#) noch näher erläutert.

Die Config ist in verschiedene Bereiche gegliedert. Diese Bereiche sind in Tabs zusammengefasst. Der erste Bereich definiert die Hardware-nahen Einstellungen: Auswahl eines *Patch*, OLA und OSC.

[Raum](#) [Config](#) [Bedienelemente](#) [Midi](#) [Datenbank](#) [Admin](#)

## Konfiguration

Eine [Übersicht zur Konfiguration](#) findest du hier.

Aktuelle Config: **jazzitbar** [speichern](#)

Patch	<input type="text" value="jazzitbar"/>
Hardware:	
OLA-Adresse	<input type="text" value="192.168.0.11"/>
Universes	<input type="text" value="2"/>
OSC	
OSC Input ein/aus	<input type="text" value="ein"/>
OSC Input Port	<input type="text" value="8800"/>

### Config Aktionen

- [Config öffnen](#)
- [Config umbenennen](#)
- [Config sichern als...](#)

### Patch

- [Patch bearbeiten](#)

### DMX

Die DMX-Ausgabe erfolgt über [Open Lighting Architecture \(OLA\)](#).

- [OLA einrichten](#)

Der nächste Config-Bereich behandelt die Bedienelemente. Hier werden die Tabellen ausgewählt, die *Stage*, *Fader* und *Buttons* definieren.

Das Aussehen der Webseite kann hier durch eine Auswahl eines Themas verändert werden.



[Raum](#) [Config](#) **Bedienelemente** [Midi](#) [Datenbank](#) [Admin](#)

## Bedienelemente

Eine [Übersicht zu Bedienelementen](#) findest du hier.

Aktuelle Config: **jazzitbar**

[speichern](#)

### Verwendete Dateien:

Stage  ▼

Executer-Buttons  
oben  ▼

Executer-Buttons  
unten  ▼

Zusatz-Buttons  ▼

Executer-Fader  ▼

Zusatz-Fader  ▼

### Start-Optionen:

Levels speichern  ▼

Starte mit  
Startcue  ▼

wähle Startcue  ▼

### Bedienelemente Aktionen

[Fader-Tabellen bearbeiten](#)

[Button-Tabellen bearbeiten](#)

### Themen

[Standard](#)

[Darkly](#)

[Flatly](#)

[Cyborg](#)

[Spacelab](#)

[Superhero](#)

Der nächste Bereich beinhaltet die Midi-Einstellungen. Das ist zwar auch ein Hardware-nahes Thema, bekommt aber aus Gründen der Übersichtlichkeit einen eigenen Tab.

[Raum](#) [Config](#) [Bedienelemente](#) [Midi](#) [Datenbank](#) [Admin](#)

## Midi-Konfiguration

Eine [Übersicht der getesteten Midi-Geräte](#) findest du hier.

Aktuelle Config: **jazzitbar**

[speichern](#)

Midi ein/aus	<input type="text" value="ein"/>
Midi Gerät 1	<input type="text" value="nanoKONTROL2"/>
Midi Gerät 2	<input type="text" value="kein Midi-Input"/>
Midi Gerät 3	<input type="text" value="kein Midi-Input"/>
Midi Gerät 4	<input type="text" value="kein Midi-Input"/>
Midi Output	<input type="text" value="nanoKONTROL2"/>

Der nächste Bereich in der Config gibt einen Überblick über die verwendeten Datenbank-Tabellen. In ClubDMX sind ja alle Daten in CSV-Tabellen gespeichert. Diese Tabellen sind im Raum-Ordner gespeichert. Hier können sie angesehen und editiert werden. Dazu stehen Zeilen- und Zellen-Funktionen zur Verfügung.

Mit Copy und Paste können einzelne oder mehrere Zeilen von einer Tabelle in eine andere übertragen werden, Zeilen können gelöscht oder neue Zeilen können über Eingabeformular angelegt werden.

Beim Editieren von einzelnen Zellen wird geprüft, ob der eingegebene Wert in die Zelle „passt“, das heißt, ob die Regeln für diese Zelle erfüllt sind.

Als weitere Hilfsmittel finden sich auf diesem Tab Funktionen zur Bereinigung der Datenbank: unbenutzt Cues löschen, alle Änderungen sichern oder alle Änderungen speichern.

[Raum](#) [Config](#) [Bedienelemente](#) [Midi](#) **Datenbank** [Admin](#)

## Datenbank

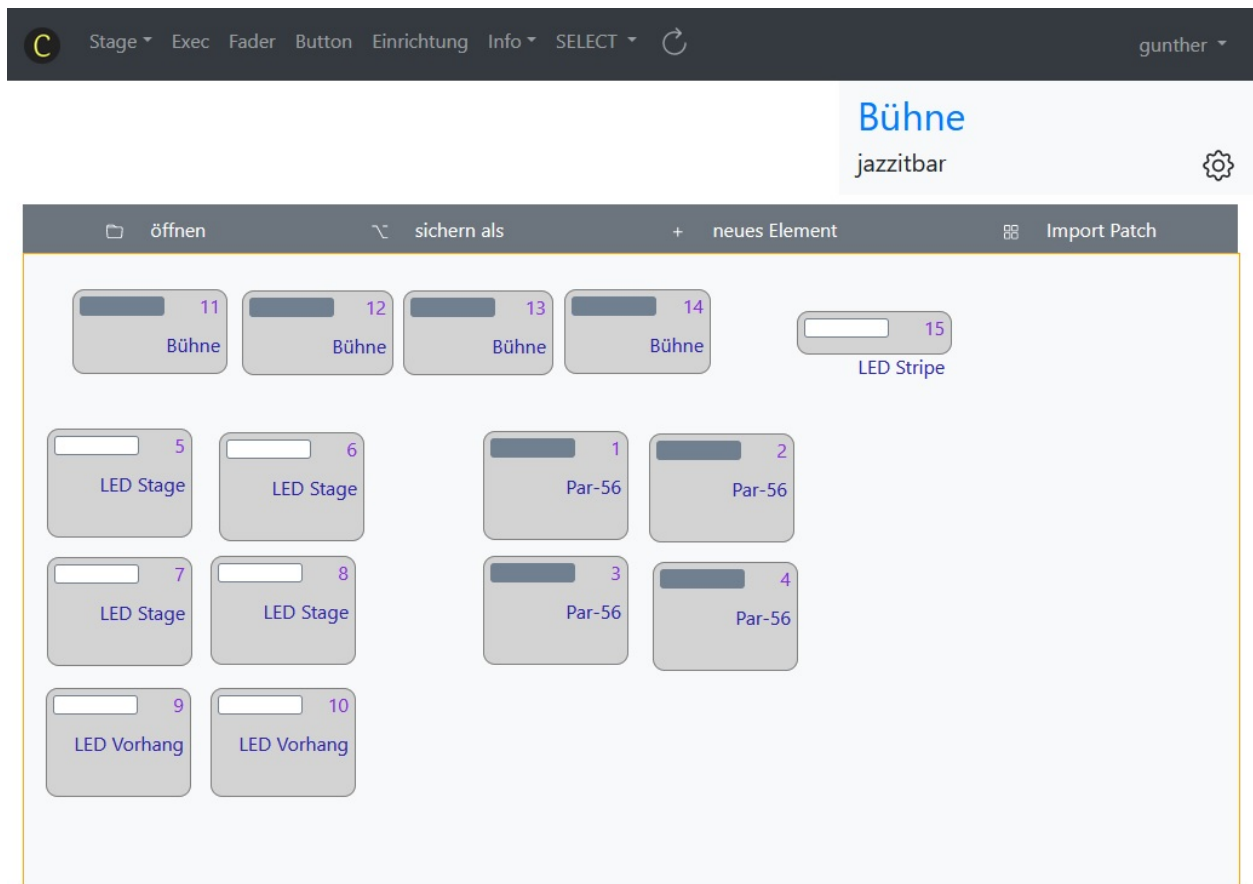
Eine [Übersicht zur Datenbank](#) findest du hier.

Basis-Tabellen	Erweitert
<a href="#">Config</a>	<a href="#">Head</a>
<a href="#">Patch</a>	<a href="#">Layout</a>
<a href="#">Cue</a>	<a href="#">Unbenutzte Cues löschen</a>
<a href="#">Fader</a>	<a href="#">Alle Änderungen sichern</a>
<a href="#">Button</a>	<a href="#">Alle Änderungen verwerfen</a>
<a href="#">Stage</a>	

Für den User *Admin* gibt es einen weiteren Tab, auf dem einige Tools zu finden sind: Backup und Restore der User-Datenbank, einen Link zur OLA-einrichtung und diverse Debug-Tools.


## 4.5 Stage einrichten

Sobald ein [Patch](#) angelegt ist, kann dieser in die Stage importiert werden. Die Heads werden in einem Standard-Raster eingefügt. Im Modus *SELECT* können die Heads in Position und Größe verändert werden.









## 4.6 Fader einrichten

Die Fader befinden sich in zwei Tabellen. Eine Tabelle beinhaltet die Fader, die auf der Exec-Seite sichtbar sind. Eine zweite Tabelle beinhaltet die Fader, die nur auf der Fader-Seite zu finden sind. Das hat den Zweck, für den Basic-Benutzer nur die für ihn relevanten Fader auf seiner Start-Seite platzieren zu können.





Stage ▾ Exec **Fader** Button Einrichtung Info ▾ SELECT ▾ ↺
gunther ▾

## Fader




	<input type="text"/>	Bar	Liste	>Topcue
	<input type="text"/>	Session	Liste	>Topcue
	<input type="text"/>	LED Intensity	Liste	>Topcue
	<input type="text"/>	Wand	Liste	>Topcue
	<input type="text"/>	Tische	Liste	>Topcue


  

	<input type="text"/>	LED rot	Liste	>Topcue
	<input type="text"/>	LED grün	Liste	>Topcue
	<input type="text"/>	LED blau	Liste	>Topcue

Durch Drücken auf das Zahnrad-Symbol gelangst du zur Einrichtungs-Seite für die Fader. Hier kannst du mit den üblichen Tabellen-Funktionen (copy/paste, Feld editieren, sichern als, neue Zeile) die Fader einrichten. Die beiden oben genannten Tabellen (Exec-Fader, Zusatz-Fader) stehen hier zur Bearbeitung. Mit cut/paste können die Fader von einer Tabelle in die andere verschoben werden.






Stage ▾ Exec Fader Button Einrichtung Info ▾ SELECT ▾ ↺
gunther ▾

## Fader-Info

jazzitbar 


Fader im Executer: jazzitbar


Zusätzliche Fader: extra-Fader

 öffnen  sichern als           + neue Zeile  upload           ✓ speichern  verwerfen           einfügen           leeren					
Text	Filename		Midiinput	Midifader	Level
Bar	Spots	Liste >Topcue	1	1	0
Session	session	Liste >Topcue	1	2	0
LED Intensity	LED intensity.csv	Liste >Topcue	1	3	0
Wand	Wand.csv	Liste >Topcue	1	7	0
Tische	Tische.csv	Liste >Topcue	1	8	0

## 4.7 Buttons einrichten

Die Buttons befinden sich in drei Tabellen: Exec-Buttons oben, Exec-Buttons unten und zusätzliche Buttons. Je nachdem, in welcher Tabelle die Buttons platziert werden, wird die Position der Buttons festgelegt.


Stage ▾ Exec Fader **Button** Einrichtung Info ▾ SELECT ▾ ↻
gunther ▾

Buttons 

Barbetrieb

Liste >Topcue

Bassbrause

1 Liste >Topcue

Homebase

1 Liste >Topcue

Afterhour

1 Liste >Topcue

Session Pause

1 Liste >Topcue

Session Live

1 Liste >Topcue

Panic

Liste >Topcue

Über das Zahnrad-Symbol kann auch hier zur Einrichtungs-Seite gewechselt werden. Auch hier kann über die üblichen Tabellen-Funktionen die Anordnung, die Beschriftung und die Funktion der Buttons eingerichtet werden.

Mit Copy/Paste können Fader zu Buttons gemacht werden und Buttons zu Fadern. Einfach Zeile(n) ausschneiden und in die gewünschte Tabelle einfügen.

Stage ▾ Exec Fader Button Einrichtung Info ▾ SELECT ▾ ↻

gunther ▾

Button-Info

Jazz IT bar

↩

Buttons im Executer oben: \_neu

Buttons im Executer unten: Jazz IT bar

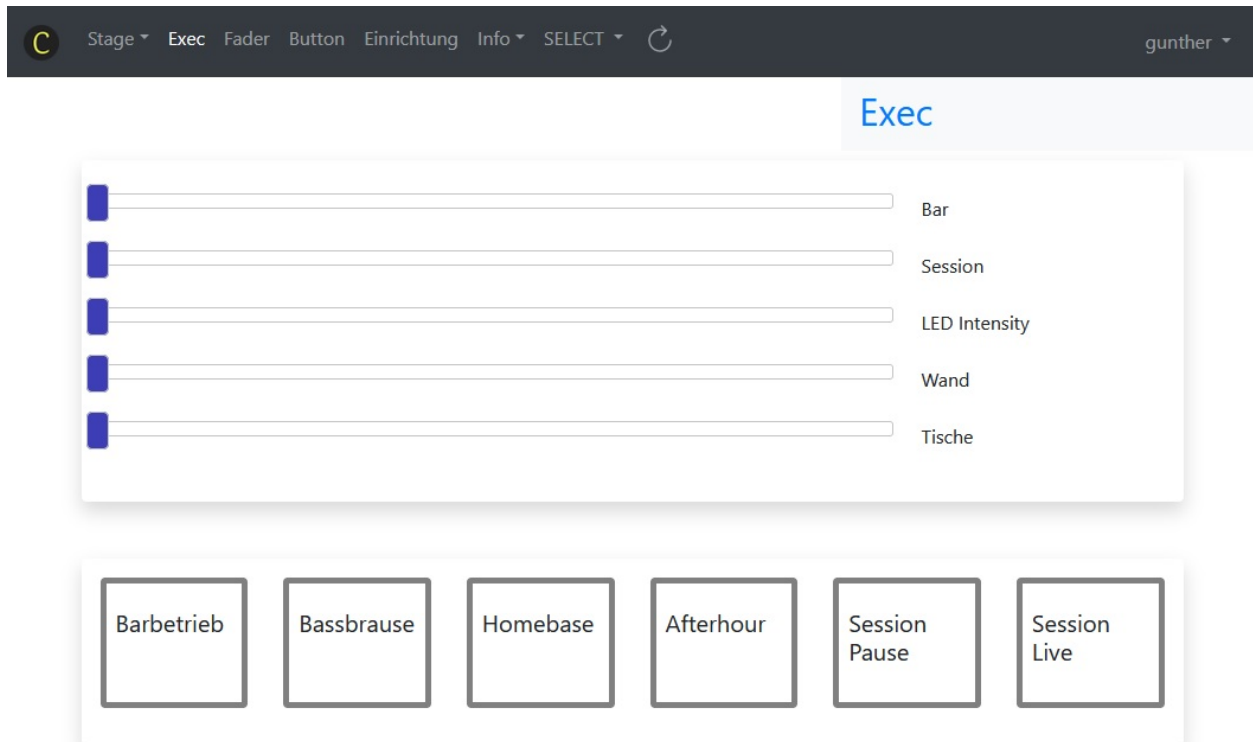
Zusätzliche Buttons: extra-Buttons

öffnen	sichern als	+ neue Zeile	upload	speichern	verwerfen	einfügen	leeren	
Text	Filename	Type	Group	Fadein	Fadeout	Midiinput	Midibutton	Level
Barbetrieb	Barbetrieb_normal.csv <div>Liste &gt;Topcue</div>	Schalter	0	0.00	3.00	1	25	0
Bassbrause	homebase_bassbrause.csv <div>Liste &gt;Topcue</div>	Auswahl	1	3.00	3.00	1	26	0
Homebase	homebase_blue.csv <div>Liste &gt;Topcue</div>	Auswahl	1	3.00	3.00	1	27	0
Afterhour	Afterhour.csv <div>Liste &gt;Topcue</div>	Auswahl	1	3.00	3.00	1	28	0
Session Pause	session_button_pause.csv <div>Liste &gt;Topcue</div>	Auswahl	1	5.10	9.20	1	29	0
Session Live	session_button_full.csv <div>Liste &gt;Topcue</div>	Auswahl	1	5	5	1	30	0

## 4.8 Exec-Seite einrichten

Die Einrichtung der Fader und Buttons bestimmt das Erscheinungsbild der Exec-Seite. Es gibt daher keine eigene Einrichtungs-Seite für die Exec-Seite, siehe also *Fader* und *Buttons* .





Weitere Informationen zur Einrichtung findest du im Kapitel *ClubDMX benutzen*, da gibt es Details zu den einzelnen Tabs.

## 4.9 Einrichten von zeitabhängiger Steuerung mit Crontab (in Linux)

Zur Unterstützung von Fixinstallationen mit Tages- oder Wochen-abhängigen Anforderungen gibt es die Möglichkeit, über den Linux-Befehl `crontab` Lichtszenen zu steuern. Hier ist exemplarisch skizziert, wie das funktioniert.

Über OSC können externe Programme auf ClubDMX zugreifen. Diese Eigenschaft nützt das Kommandozeilen-Tool `sendosc.sh` (im Programm-Verzeichnis zu finden).

Beispiel für einen Crontab-Eintrag, der jede Minute aufgerufen wird und den dritten Button aus der oberen Exec-Zeile betätigt:

```
* * * * * /home/pi/clubdmx_code/sendosc.sh --address=/exebutton1/3
```

Die unterstützten OSC-Befehle finden sich auf der ClubDMX Webseite und auch hier in der Doku im Kapitel *ClubDMX benutzen*.


Mit dem Linux-Befehl `crontab -e` wird die Crontab editiert, mit dem Befehl `crontab -l` wird sie angezeigt und mit `crontab -r` wird sie gelöscht.



---

## ClubDMX benutzen

---

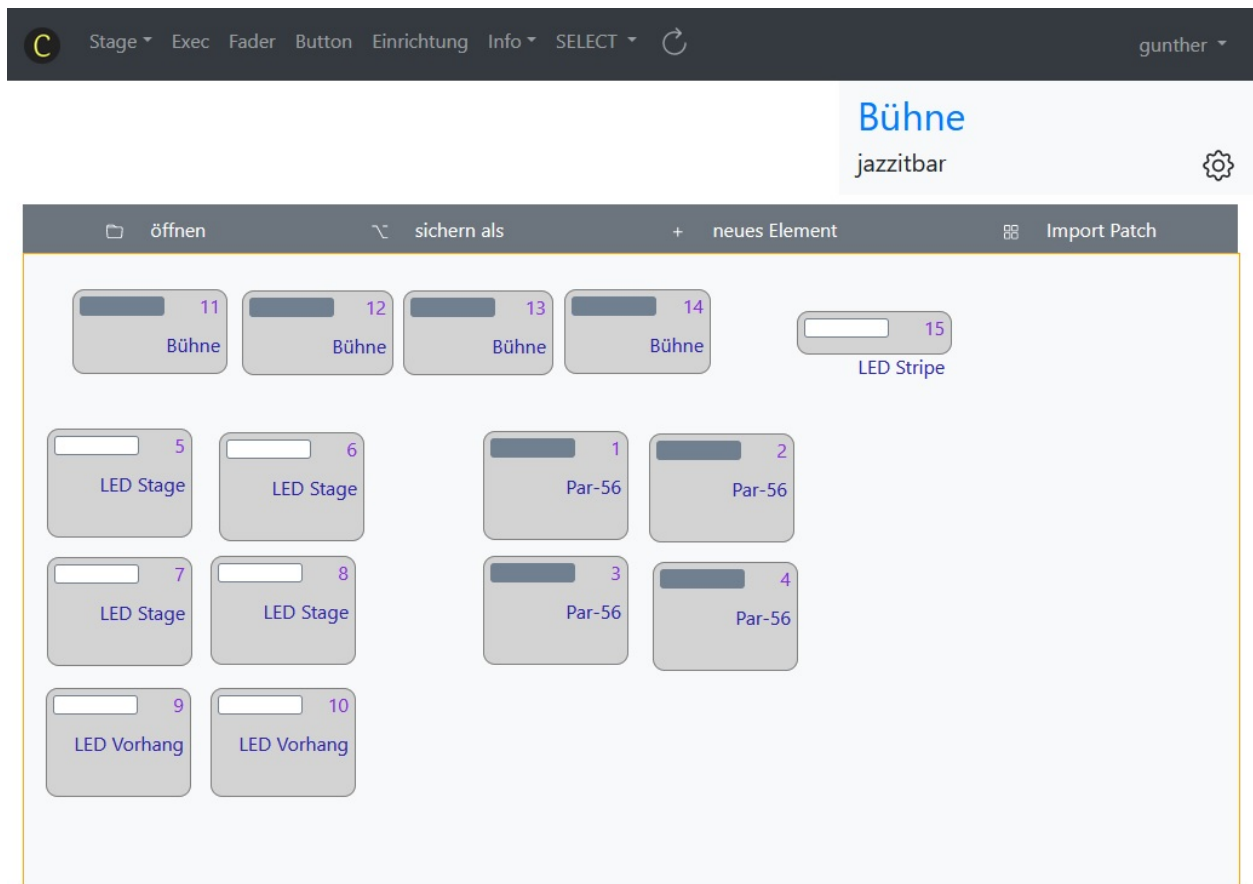
Über die Startseite (erreichbar über das Symbol ) können die wichtigsten Module aufgerufen werden, über die Navigationsleiste können alle Module aufgerufen werden.

Hier ist eine Übersicht über die Navigation:

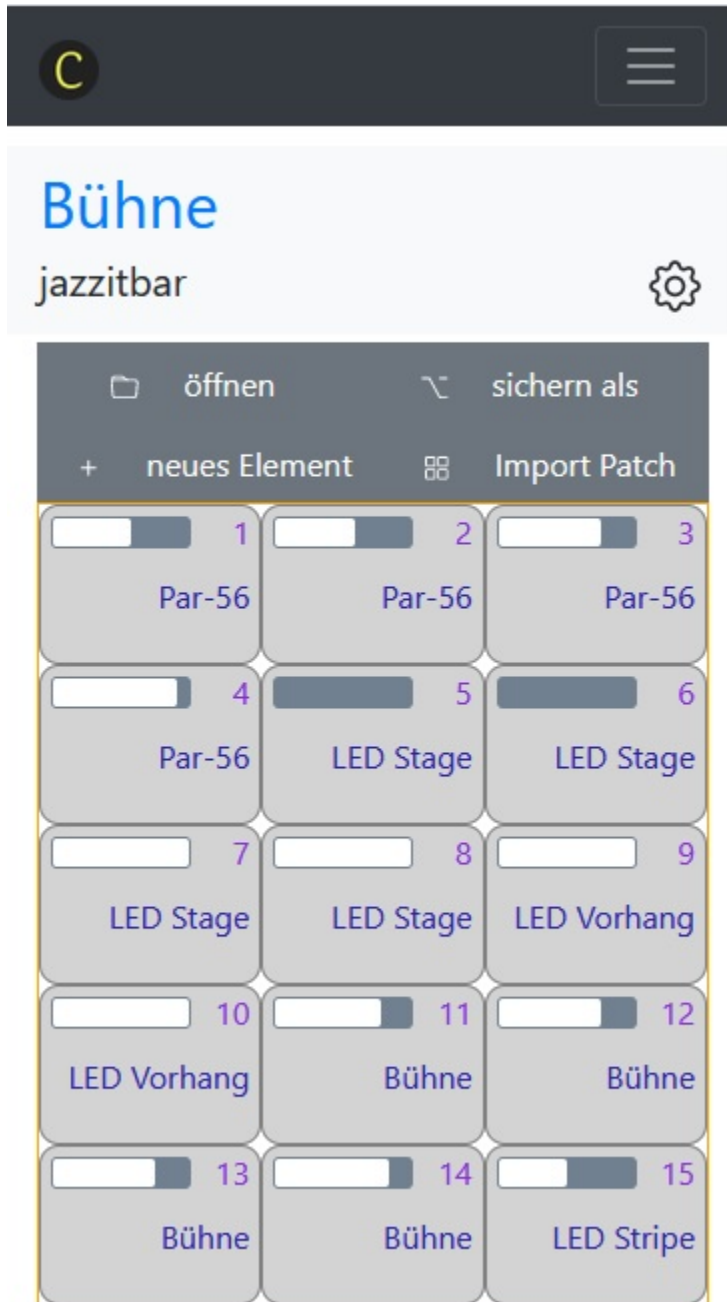
### 5.1 Stage

- Stage
- Stage kompakt
- Single

Die erste Wahl in der Navigation ist die **Stage**. Dieses Modul besteht aus mehreren Webseiten. Die erste Wahl ist die Stage mit beweglichen und selektierbaren Elementen. Als Idee dahinter ist, die **Geräte (= Heads)** wie in einem Grundriss eines Raumplanes zu positionieren. Damit können die Geräte auf dem Bildschirm so angeordnet werden, wie sie sich im reellen Raum befinden.



Da bei Smartphones und anderen kleinen Anzeigen der Platz oft nicht ausreicht, wird eine zweite Stage-Ansicht zur Verfügung gestellt: **Stage kompakt**. Hier werden die Elemente neben- und untereinander so platziert, dass der Bildschirm optimal genutzt wird (Responsive Design).



In der **Single** Seite kann mittels Navigationspfeilen <<, <, >, >> jedes Gerät einzeln in den *Topcue* geladen werden. Für das ausgewählte Gerät werden im Arbeitsbereich Fader angezeigt, die die einzelnen *Attribute* bedienbar machen.

C

☰

**Topcue:**   leeren   anzeigen   speichern ▾

Intensity

Red

Green

Blue

Single:

15

«

<

>

»

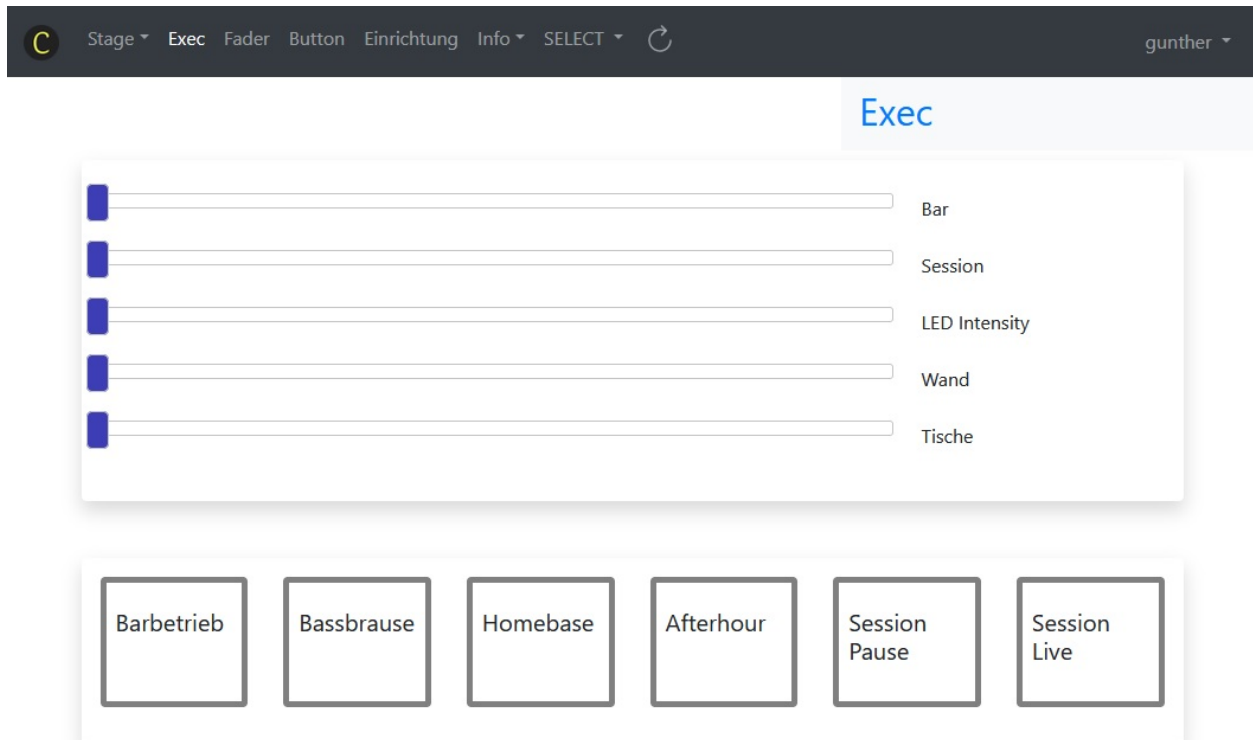
Head Details:

HeadType	Addr	Name	Gel	Comment
LEDrgb_v.csv	1-500	LED Stripe	0	

## 5.2 Exec

Die Exec-Seite enthält Buttons und Fader. Sie kann so konfiguriert werden, dass sie einen Button-Bereich, einen Fader-Bereich und einen weiteren Button-Bereich anzeigt. Jeder dieser Bereiche kann auch leer sein.

Das Ziel der Exec-Seite ist ein schneller und simpler Zugang zu Szenen. Auf die Editiermöglichkeiten wurde daher verzichtet. Zum Einrichten und Editieren der Exec-Seite wechselt man auf die Fader- oder Button-Seite. Genauerer dazu findest du unter *Fader einrichten* und *Buttons einrichten*.



## 5.3 Fader

Die Fader-Seite war der Startpunkt für die Arbeiten zu ClubDMX, diese Seite war das erste Ziel: Lichtszenen (Cues) über einen Webseiten-Fader zu bedienen.

Die dazu nötigen Voraussetzungen waren dann doch eher umfangreich, aber die für die Anwender wichtige Klarheit ist hier nach wie vor gegeben: Ein Fader für jeden Cue, ein Text zur Identifizierung, ein Button zur Ansicht, welche Geräte mit diesem Fader bedient werden und ein Button zum Editieren des Cues.

Neue Fader hinzufügen oder entfernen, die Reihenfolge der Fader ändern: Das alles ist möglich, und zwar in der Fadertabelle. Siehe *Fader einrichten*

In der Fadertabelle ist es auch möglich, einen MIDI-Regler einem Fader zuzuordnen.

## 5.4 Button

Ganz gleich wie die Fader-Seite ist die Button-Seite für die einfache Bedienung konzipiert: Hier können Szenen (Cues) über die Webseite ein- oder ausgeschaltet werden.

Buttons unterscheiden sich von Fadern dadurch, dass bei Buttons nur die Endpunkte 0 und 100% (Aus oder Ein) vom Benutzer gewählt werden können.

Der Übergang von Aus zu Ein beziehungsweise von Ein zu Aus kann mit Fade-Zeiten verknüpft werden. Dann bestimmt die Zeit in Sekunden einen linearen Übergang von Ursprungs- zum Zielzustand.

Es gibt drei verschiedene Arten von Buttons:

- Schalter
- Taster
- Auswahlshalter

Ein Schalter verändert mit jedem Drücken seinen Status, von *aus* nach *ein* und wieder zurück. Ein Taster ist nur dann *ein*, wenn er gedrückt und gehalten wird. Das funktioniert nur auf einem Midi-Keyboards, nicht auf der Webseite.

Ein Auswahlshalter hängt mit anderen Auswahlshaltern zusammen: Mehrere Auswahlshalter bilden eine Gruppe (mit einer Gruppen-Nummer). Wird ein Schalter aus dieser Gruppe *ein*-geschaltet, dann wird damit ein momentan aktiver Schalter aus dieser Gruppe *aus*-geschaltet. Es kann also immer nur ein Schalter aus einer Gruppe *ein* sein.

Es können beliebig viele Buttons eingerichtet werden. Auswahlshalter müssen nicht nebeneinander platziert werden, sie werden durch die Gruppen-Nummer als zusammengehörig definiert.

Auf der Button-Seite sind die für den Benutzer essentiellen Informationen enthalten: Text zur Identifizierung, Status (Aus: grauer Rand, Ein: roter Rand), weiters Buttons zur Ansicht der Cue-Informationen und zum Editieren.

In der Buttontabelle werden alle Infos zu den Buttons verwaltet. Hier können Buttons hinzugefügt, entfernt und in der Reihenfolge verändert werden. Weiters werden hier die Fade-Zeiten und die Zuordnung zu MIDI-Tasten eingetragen.

## 5.5 Einrichtung

Die Einrichtung von ClubDMX ist schon recht vielseitig, meine Hoffnung ist, dass sie trotzdem noch übersichtlich ist, um sich schnell zurecht zu finden.

- Raum
- Config
- Bedienelemente
- Midi
- Datenbank
- Admin

Die Einrichtung ist in verschiedene Bereiche gegliedert, die sich auf der Webseite in Tabs wiederfinden.

Im **Raum**-Tab werden die ersten und elementaren Einrichtungen getätigt: Einen neuen Raum anlegen, den Raum wechseln, den aktiven Raum umbenennen. Im Raum-Tab befinden sich auch die Optionen zum Backup und Restore.

Die nächste Ebene der Einrichtung ist die **Config**, deren Hardware-Komponenten sich im Config-Tab befinden. Das Kernstück der Konfiguration ist der **Patch**. Die zweite wesentliche Hardware-Komponente ist OLA, das die physische Zuordnung über DMX zu den Geräten herstellt.



Falls für einen Raum mehrere Configs angelegt wurden, kann hier eine davon ausgewählt werden. Die aktuelle Config kann auch unter einem anderen Namen gespeichert werden.

Zur Erläuterung der Begriffe Raum und Config siehe [Räume und Config](#).

Nach dem Öffnen einer Config werden der [Patch](#) ausgewählt, die OLA-Adresse und die Anzahl der Universen eingestellt.

Im **Config**-Tab wird auch OSC Input ein- oder ausgeschaltet.

Im **Bedienelemente**-Tab sind die Tabellen auswählbar, die die zur Nutzung relevanten Seiten definieren: Stage, Fader und Buttons. Damit können für einen Anwendungsbereich die nötige Lokalisierung der Geräte auf der Stage und die passenden Fader und Buttons bereitgestellt werden.

Für den Fall, dass ClubDMX als Steuerung für eine Fix-Installation verwendet wird, die nach einem Ausschalten mit den letzten Einstellungen wieder hochgefahren werden soll, gibt es den Schalter *Levels speichern*.

Verschiedene Stil-Themen für die Webseite können im Bedienelemente-Tab ausgewählt werden.

Im **Midi**-Tab können maximal vier Midi-Geräte als Input definiert werden. Diese Midi-Geräte können Fadern und Buttons zugeordnet werden. Damit können Cues auch ohne Website bedient werden.

Der **Datenbank**-Tab ermöglicht einen Zugriff auf die Tabellen, die hier bearbeitet werden können. Je nach gewähltem [Bearbeitungsmodus](#) können hier Zell- oder Zeilenbearbeitungen gemacht werden.

Umfassende Erläuterungen zum Thema Einrichtung finden sich im Kapitel [Einrichtung](#).

## 5.6 Info

Information zur Software und den aktuellen DMX-Output

- DMX-Output: zeigt alle DMX-Werte, die nicht Null sind. Das ist zur Kontrolle gedacht, da man hier sieht, welche Berechnungen ClubDMX für die einzelnen Fader, Buttons und den Topcue vorgenommen hat.
- Info: Kurz-Informationen
- Doku: Umfangreiche Information zu verschiedenen Themen: Erste Schritte, Einrichtung, die einzelnen Navigations-Punkte (Module) und ein Bereich zu Grundlagen der Lichtsteuerung.

## 5.7 Benutzer-Datenbank

Die Benützung von ClubDMX ist nur mit einem Login möglich. Es gibt verschiedene Rollen der Berechtigung, zur Zeit sind das Basic, Standard und Admin.

Die genaue Kategorisierung, welche Aktionen mit welcher Rolle durchgeführt werden dürfen, ist noch in Arbeit. Den aktuellen Stand zu den Benutzer-Rollen siehe [Benutzer](#).

## 5.8 OSC Input

Dieses Modul ermöglicht die Verbindung mit externen Programmen wie zum Beispiel *Isadora* . Über **OSC** können folgende Aktionen in ClubDMX ausgeführt werden:

- `/head <attribut> <headnr> <level>` ermöglicht de Zugriff auf sämtliche Geräte und deren Attribute.

`<attribut>` bezeichnet den Attribut-Namen, so wie er in der Head-Definition angelegt ist.

`<headnr>` ist eine Ganzzahl entsprechend der Headnummer im Patch.

`<level>` ist eine Gleitzahl im Bereich zwischen 0 und 1.

- `/clear` leert den *Topcue* .
- `/fader/<nummer> <level>` ändert den Level eines Faders.
- `/exefader/<nummer> <level>` ändert den Level eines Faders aus der Exec-Seite.

`<nummer>` ist eine Ganzzahl entsprechend der Fadertabelle.

`<level>` ist eine Gleitzahl im Bereich zwischen 0 und 1.

- `/button/<nummer>` betätigt einen Button.
  - `/exebutton1/<nummer>` betätigt einen Button aus der oberen Reihe der Exec-Seite.
  - `/exebutton2/<nummer>` betätigt einen Button aus der unteren Reihe der Exec-Seite.
- `<nummer>` ist eine Ganzzahl entsprechend der Buttontabelle.

ClubDMX ist mit einer Benutzerstruktur ausgestattet. Je nach Rolle des Benutzers sind Funktionen und Seiteninhalte verfügbar.

Es gibt drei Rollen: Basic, Standard und Admin.

### 6.1 Basic

Benutzer mit dieser Rolle können Stage, Fader und Buttons bedienen. Damit kann ein Basic-Benutzer alle voreingestellten Szenen aufrufen und auf alle Geräte zugreifen. Eine Veränderung von Szenen ist nicht möglich.

### 6.2 Standard

Benutzer mit Standard-Berechtigung haben vollen Zugriff auf die Programmierung, sie können Veränderungen nach ihren Vorstellungen vornehmen. Alle Änderungen werden so gespeichert, dass sie auch rückgängig gemacht werden können. Das Rückgängig-Machen kann vom Standard-Benutzer oder vom Admin gemacht werden.

Ein Standard-Benutzer kann auch neue Benutzer anlegen, denen er die Rolle Standard oder Basic zuordnen kann.

### 6.3 Admin

Gegenüber dem Standard-Benutzer hat der Admin Rechte in der Form, dass er einen aktuellen Programmstatus in Teilen oder zur Gänze permanent speichern kann. Permanent gespeicherte Daten können dann nicht mehr rückgängig gemacht werden.

Der Admin kann auch weitere Benutzer mit Admin-Rechten anlegen, Benutzer löschen und die Passwörter von Benutzern zurücksetzen.

## 6.4 Funktionen

Die verschiedenen Seiten zur Benutzerverwaltung sind entsprechend der Rolle des jeweiligen Benutzers verfügbar. Die komplette Benutzerverwaltung ist nur Admins zugänglich, ein Standard-Benutzer sieht die Liste der Benutzer und kann neue Benutzer anlegen. Ein Basic-Benutzer kann sich nur an- und abmelden und das Passwort ändern.

Jedes physisch vorhandene Gerät wird im Patch eingetragen. Damit wird es für ClubDMX verfügbar.

Aufruf über die Navigation: Einrichtung -> Config -> Patch bearbeiten.

Zu den einzelnen Spalten:

- HeadNr: Jedes Gerät wird mit einer Zahl identifiziert. Es können auch mehrere Geräte dieselbe HeadNr haben. Sie werden dann von ClubDMX gleich behandelt.
- HeadType: Jedem Gerät wird eine Head-Datei zugeordnet. Das ist die eindeutige Beschreibung der einzelnen DMX-Kanäle für das zu patchende Gerät.
- Addr: Die DMX-Startadresse. Setzt sich aus zwei Komponenten zusammen, die mit Bindestrich getrennt sind: *<Universum>-<Startadresse>*. Von ClubDMX wird beim Erstellen von neuen Zeilen geprüft, ob die Adressen korrekt im Bereich von 1-512 liegen. Nicht geprüft wird, ob eventuell Überschneidungen mit anderen Heads auftreten.
- Name: Das ist ein Textfeld, das für den Benutzer zur Identifizierung des Hardware-Gerätes dient. Beim Import des Patch in die Stage wird der Name zur Identifikation des Stage-Elementes verwendet.
- Gel (Farbe): Das ist ein Textfeld, das für die Farbnummer eines konventionellen Scheinwerfers gedacht ist. Dient zur Dokumentation eines Projektes und ist zur Zeit nicht weiter ausgeführt.
- Comment: Wie schon der Name sagt, für Kommentar und Anmerkung

## 7.1 Neuen Patch erstellen

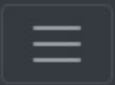

In jeder Datenbank-Kategorie gibt es eine Tabelle mit Namen *\_neu*, die als Prototyp fungiert. So auch in der Kategorie Patch. Sichere diesen Prototyp unter einem neuen Namen und beginne nun mit der Erstellung der Zeilen.

Mit dem Button *Neue Zeile(n)* wird ein Formular zum Erzeugen eines neuen Eintrags geöffnet. Fülle das Formular zur Erstellung eines neuen Eintrags aus.

Mit dem Formular können mehrere Heads vom selben Head-Typ in einem Schritt erzeugt werden. Je nach Head-Typ werden die Adressen für die einzelnen Heads berechnet, mit dem Abstand, der sich aus der Head-Definition errechnet.

Dabei wird auch der maximale DMX-Wert von 512 berücksichtigt. Unter Umständen können daher weniger als die gewünschten Heads erzeugt werden, wenn ein oder mehrere Attribute einen DMX-Wert über 512 hätten.


Im Bearbeitungsmodus SELECT kannst du nun sehr schnell weitere Zeilen mit Copy/Paste erzeugen, die du im Modus EDIT entsprechend anpasst.



## Neue Zeilen in Patch

Patch: jazzitbar

---

Anzahl neuer Heads	<input type="text" value="1"/>
Head-Nummer	<input type="text" value="16"/>
Head-Typ	<input type="text" value="_neu"/> 
Adresse	<input type="text" value="uni-address"/>
Name	<input type="text"/>
Farbe	<input type="text"/>
Kommentar	<input type="text"/>

---

erzeugen

schließen

### 8.1 DMX

Über das DMX-Protokoll können alle möglichen unterschiedlichen Geräte bedient werden, zum Beispiel Dimmer, LED-Scheinwerfer, Nebelmaschinen etc. Diese Geräte haben ein oder mehrere Attribute, wie Intensität, Farben, Position oder Geschwindigkeit. Ein Gerät hat eine Startadresse, das ist eine Zahl zwischen 1 und 512. So ein Zahlenbereich von 1-512 wird als Universum bezeichnet, was natürlich verglichen mit dem realen Universum reichlich übertrieben ist. Daher ist es mit dieser Lichttechniker-Definition von Universum möglich, dass sich in einem Raum mehrere Universen befinden.

### 8.2 Hardware

Geräte-Eigenschaften werden mit Attributen beschrieben. Diese Beschreibung nennt man Head-Definition. In ClubDMX werden Head-Definitionen wie alle anderen Daten in CSV-Tabellen gespeichert.

Ein Gerät im Universum könnte zum Beispiel ein LED-Scheinwerfer mit den Farben rot, grün und blau sein. Diese Farben sind die Attribute, die unabhängig voneinander in der Intensität verändert werden können. In einer simplen Lichtsteuerung erhält jede Farbe einen Regler zugewiesen, mit dem dann die Intensität eingestellt wird. Wenn ich jetzt aus den drei Farben einen schönen Farbton gemischt habe, dann wäre es wünschenswert, wenn ich diesen Farbton nun in der Intensität verändern könnte. Das ist mit drei Reglern etwas schwierig, da ich ja die drei Regler im richtigen Verhältnis zueinander auf oder ab schieben soll.

Und hier kommt ein Konzept der Lichttechnik ins Spiel, das sich in den letzten Jahren verbreitet hat, nämlich der *virtuelle Dimmer*. Ein virtueller Dimmer ist ein Faktor (zwischen 0 und 1), mit dem jeder Farbwert des LED-Scheinwerfers multipliziert wird. In der Lichttechnik ist der virtuelle Dimmer nun ebenso ein Regler, der zusätzlich zu den Farben des LED-Scheinwerfers zur Verfügung steht.

## 8.3 Räume und Config

In ClubDMX werden in einem *Raum* alle Daten der Datenbank zusammengefasst, die für die Steuerung notwendig sind. Das sind unter anderem Konfigurationen, Patches, Head-Daten, Cues, Stages, Fadertabellen und Buttontabellen. Durch diese Struktur ist es einfach, alle zu einem Projekt gehörenden Daten zu sichern und auch, ClubDMX in verschiedenen Anwendungsbereichen zu betreiben.

Die Aufgabe der Config ist es, für einen Raum die Anforderungen an eine spezielle Aufgabe anzupassen, zum Beispiel für eine Aufführung oder eine Personengruppe, die einen Veranstaltungsort nützt.

Bespiele dazu wären, wenn man für einen Club verschiedene Configs für Standard- und Session-Bespielung oder für einen Probenraum verschiedene Configs für die jeweiligen Nutzer anlegt.

## 8.4 Bearbeitungsmodus

Wie aus anderen Tabellenkalkulationen bekannt, werden zwei unterschiedliche Editiermethoden benötigt: Zellenweises Editieren und Zeilenoperationen.

In ClubDMX ist es genauso, daher wird der Begriff des Bearbeitungsmodus eingeführt. Das Editieren von Text in Zellen geschieht im Modus *EDIT*, Zeilenoperationen wie Ausschneiden, Kopieren und Einfügen geschieht im Modus *SELECT*.

Auf der **Stage**-Seite hat der Modus *SELECT* eine weitere Aufgabe: In diesem Modus werden Symbole selektiert, die dann in der Größe verändert und verschoben werden können. Für die selektierten *Heads* werden Attribut-Slider angezeigt, mit denen die Attribute verändert werden können.

## 8.5 HTP und LTP

Zu Zeiten, als in der Theaterbeleuchtung außer Dimmern bestenfalls noch ansteuerbare Schalter zu finden waren, gab es eine klare Regelung: Wenn ein Dimmer von einem Regler in einem bestimmten Wert angefordert wurde und von einem anderen Regler in einem anderen Wert, so galt der höhere Wert.

In Zeiten von Moving Lights, Farbwechslern und Equipment, das andere Parameter als Intensität kennt, ist diese einfache Regel nicht mehr ausreichend.

Ein einfaches Beispiel: Wird die Position eines Moving Light mit zwei Werten für die x- und die y-Achse beschrieben, so definiert zum Beispiel Szene 1 die Position als 50% für x und 50% für y. Wenn nun in einer nachfolgenden Szene 10% für x und 90% für y angefordert wird, dann eräbe sich nach der „höchsten-Wert-Regel“ als Kombination der beiden Szenen ein x-Wert von 50% und ein y-Wert von 90%. Das entspricht dann keiner der beiden Szenen und ist als Ergebnis völlig unbrauchbar.

Daher braucht es für solche Attribute wie Position eine andere Regel. Diese ist ebenso simpel wie die Regel „der höchste Wert gilt“, die Regel lautet: Der letzte Wert gilt.

In der englischsprachigen Welt gibt es natürlich auch Überlegungen diesbezüglich, und als Resultat davon haben sich bei uns auch die englischen Abkürzungen für diese Regeln zur Beschreibung durchgesetzt: HTP und LTP.

HTP bedeutet **highest takes precedence**, LTP steht für **latest takes precedence**.

Als Zusammenfassung dieser beiden Regeln kann für die zeitgemäße Lichtsteuerung gesagt werden, dass die Intensität nach der HTP-Regel und alle anderen Attribute nach der LTP-Regel gesteuert werden.

Die Ausnahme von dieser Aussage bilden die RGB-Farben eines LED-Scheinwerfers, bei dem die Farben rot, grün und blau entweder als eigenständige Intensitäten und somit gemäß HTP-Regel oder als kombinierter Farbwert und



damit gemäß LTP-Regel behandelt werden. Wie RGB-Werte gesteuert werden, hängt von den Präferenzen des Lichttechnikers ab und die beste Methode ist von der jeweiligen Situation abhängig.

Siehe: [http://www.thedmxwiki.com/dmx\\_basics/ltp\\_and\\_http](http://www.thedmxwiki.com/dmx_basics/ltp_and_http)

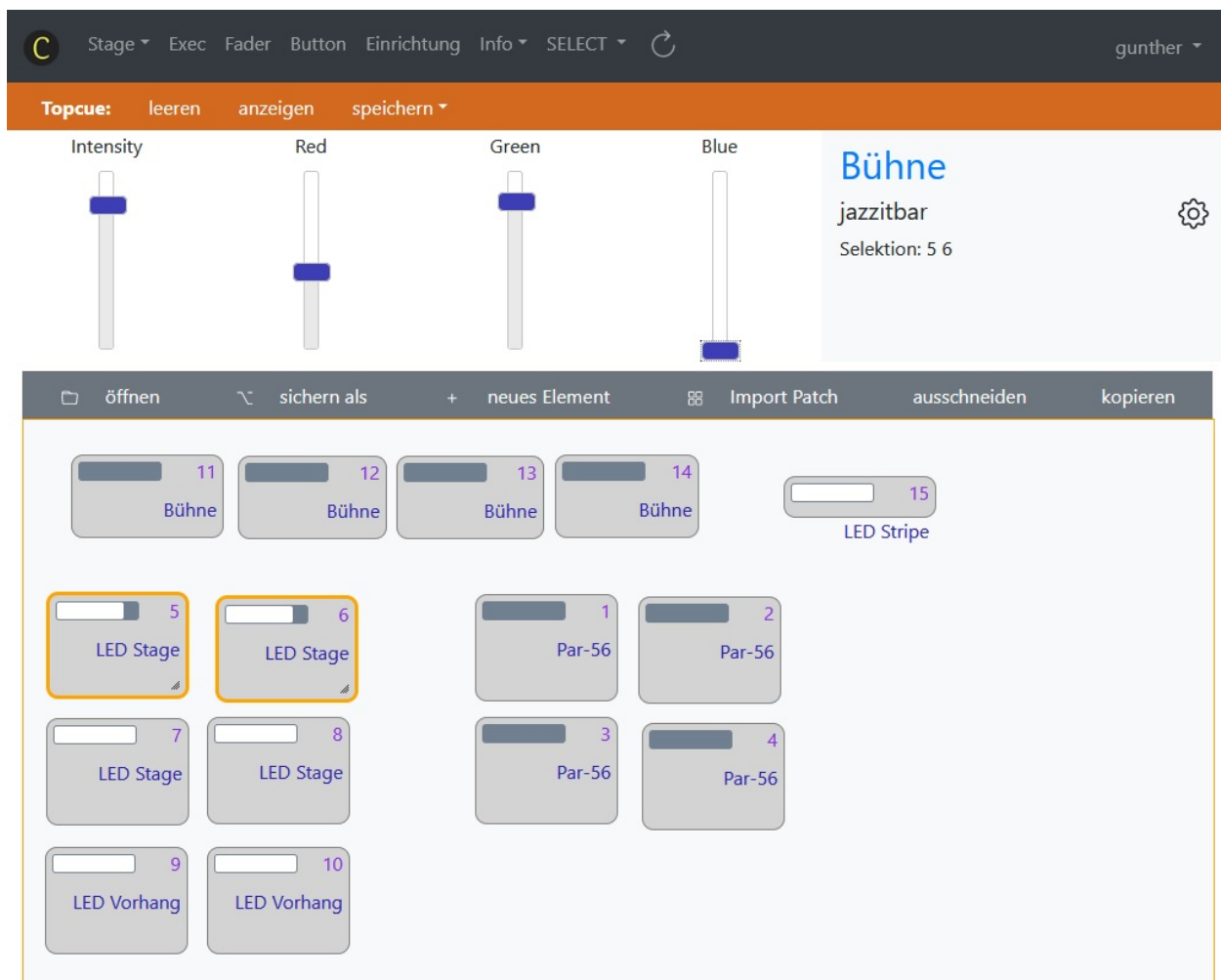
## 8.6 Topcue

Hinter dem Begriff **Topcue** steht eine simple Idee: Das ist der Speicher der Werte, die neu programmiert oder verändert wurden. Diese Werte sollen aktuell gültig sein und es soll auch eine Speicherung in einem neuen Cue oder eine Änderung eines vorhandenen Cues ermöglichen.

Der Topcue ist eine Lichtszene (**Cue**), die aber nicht gemäß HTP-Regel behandelt wird. Für diese Szene gilt, dass jeder hier enthaltene Wert Vorrang (= Top-Priorität) hat. Somit kann ein aus der Summe der aktiven Szenen errechneter Wert außer Kraft und durch einen anderen Wert ersetzt werden.

Es gibt verschiedene Möglichkeiten, damit Werte in den Topcue gelangen:

- Attribute werden im Single-Modus verändert.
- In der Stage werden Geräte ausgewählt und es werden Attribute verändert.
- Ein Cue wird mit dem *Topcue-Button* in den Topcue aufgenommen.

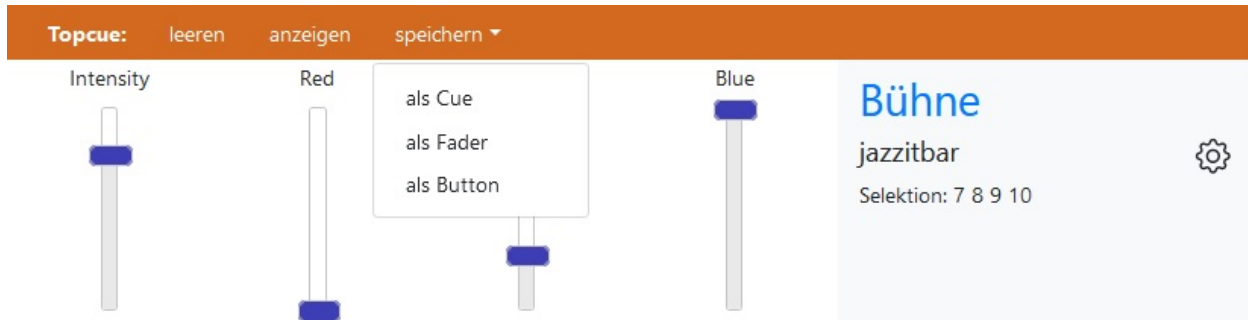


Sind Werte im Topcue enthalten, dann erscheint unterhalb der Standard-Navigationsleiste eine zweite Navigationsleiste mit den Optionen **leeren**, **anzeigen** und **speichern**.

Mit *leeren* wird der Topcue gelöscht und es gelten wieder die errechneten Werte für alle Geräte.

Mit *anzeigen* wird der Inhalt des Topcue in einer Liste dargestellt.

Der Button *speichern* eröffnet weitere Möglichkeiten: Die Werte des Topcue können als neuer Cue, als Fader oder als Button abgespeichert werden.



---

## Raspberry einrichten

---

Der Raspberry PI ist eine für ClubDMX getestete Hardware, hier ist eine Anleitung zur Neu-Installation von Raspbian Buster als Betriebssystem, Installation von OLA und Installation von ClubDMX.

### 9.1 Raspberry Buster neu installieren

- Programm Imager von <https://www.raspberrypi.org/downloads/>
- Image erzeugen
- Raspi mit Bildschirm und Tastatur starten, anschließend Guide ausführen
- Raspi-Config:

```
sudo raspi-config
2 Network Options
# -> N1 Hostname: Pi-Name eintragen (optional)

3 Boot Options
# -> B2 Wait for Network on Boot
# (sonst ist OLA nicht mit allen Plugins ausgestattet)

4 Localisation Options
# -> Ländercode + Utf-8, auch Ländercode für Wlan (wichtig)

5 Interfacing Options
# -> P2 SSH enable (wichtig)

5 Interfacing Options
# -> P3 VNC enable (optional)
```

## 9.2 OLA installieren

Für die Installation von OLA gibt es zwei Möglichkeiten, eine einfache, die das nicht tagesaktuelle Repository installiert, und eine langwierigere, die das aktuelle OLA von GIT installiert. Die zweite Version war für mich notwendig in einem Fall, wo ich als DMX-Adapter den *Eurolite DMX512 Pro MK2* verwenden wollte, der war im Repository noch nicht verfügbar. Hier sind beide Varianten beschrieben, nur eine davon muss installiert werden.

### Einfache OLA Installation

```
sudo nano /etc/apt/sources.list
```

hier eintragen:

```
#ola:
deb http://apt.openlighting.org/raspbian wheezy main
```

anschließend Reboot. Nach dem Neustart:

```
sudo apt-get install ola
```

Damit ist OLA installiert und startet automatisch bei jedem Neustart.

### OLA von GIT installieren

Wenn die *einfache OLA-Installation* gewählt wurde, kann der folgende Abschnitt übersprungen werden. Es geht dann weiter im Abschnitt *OLA für ClubDMX vorbereiten*.

Hier sind meine Notizen, die ich während der Installation gemacht habe. Diese sind auch in [Google Groups](#) nachzulesen.

```
sudo apt-get install git
git clone https://github.com/OpenLightingProject/ola.git ola
cd ola

# (Error: could not resolve host..., therefore:)
git config --global --unset http.proxy
git config --global --unset http.proxy

sudo apt-get install autoconf libtool bison flex uuid-dev libcppunit-dev
sudo apt-get install libmicrohttpd-dev protobuf-compiler libprotobuf-dev python-
↳protobuf
sudo apt-get install libftdi-dev liblo-dev libavahi-client-dev libprotoc-dev
sudo apt-get install libusb-1.0.0-dev libcurl5-dev pkg-config liblo
autoreconf -i
./configure
# optionally: ./configure --enable-rdm-tests --enable-python-libs
make
make check
sudo make install
sudo ldconfig
```

Damit ist die neueste Version von OLA installiert. Nun muss noch Autostart konfiguriert werden:

```
sudo nano /etc/rc.local
```

hier eintragen vor der letzten Zeile (*exit 0*):

```
su pi -c "olad -f"
```

### Eurolite usb-dmx mk2

Dieser Abschnitt muss nur dann ausgeführt werden, wenn der Eurolite DMX-Adapter verwendet werden soll. Diese Anleitung funktioniert mit der neuesten Version von OLA, nicht aber mit der einfachen OLA-Installation. Ansonsten weiter im Abschnitt *OLA für ClubDMX vorbereiten*.

```
sudo nano /etc/modprobe.d/eurolite-dmx.conf
```

hier eintragen:

```
blacklist cdc_acm
```

Eurolite Adapter anstecken und Vendor-ID und Product-ID prüfen:

```
lsusb
sudo nano /etc/udev/rules.d/02-eurolite-dmx.rules:
# (this is one line:)
SUBSYSTEM=="usb|usb_device", ACTION=="add", ATTRS{idVendor}=="0403",
  ATTRS{idProduct}=="6001", GROUP="plugdev" MODE="660"
```

Nun müssen noch einige conf-Dateien von OLA angepasst werden. Diese können sich an unterschiedlichen Orten befinden, im Raspberry PI mit der beschriebenen Installation aber wahrscheinlich in `/home/pi/.ola/`. Zur Sicherheit am Besten im Browser die OLA-Admin Seite `127.0.0.1:9090` öffnen und im Abschnitt *Plugins* eine Seite öffnen, dann findet sich hier die *Config Location*.

In den drei Dateien `/home/pi/.ola/ola-opendmx.conf` , `/home/pi/.ola/ola-usbserial.conf` und `/home/pi/.ola/ola-stageprofi.conf` jeweils die Zeile `enabled = true` auf `enabled = false` ändern.

In der Datei `ola-usbdmx.conf` die Zeile `enable_eurolite_mk2 = true` einfügen.

Damit ist nach einem Neustart der Eurolite Adapter verfügbar.

## 9.3 OLA für ClubDMX vorbereiten

Weiter im Browser, auf Seite `127.0.0.1:9090`.

ClubDMX kommuniziert mit OLA über OSC, daher muss in OLA in jedem Universum ein OSC-Input eingetragen werden.

Ein oder mehrere Universen anlegen:

```
Universes -> Add Universe
Universe ID: 1
Universe Name: Un1
Checkbox anhaken bei erster Zeile OSC Device Input /dmx/universe/%d
Checkbox für gewünschten Output anhaken.
```

Für weitere Universen wiederholen.

## 9.4 ClubDMX installieren

Ich entwickle ClubDMX auf einem Windows-Rechner, daher kann es sein, dass Shell-Skripte erst ins Unix-Format umgewandelt werden müssen. Dazu gibt es ein Hilfsprogramm namens `dos2unix`. Siehe auch [hier](#)

```
sudo apt-get install dos2unix
```

Und nun zur eigentlichen Installation von ClubDMX. Die aktuelle Version ist in meinem [Google Drive](#) zu finden.

### Standard-Installation

In der Standard-Installation werden ein Code-Verzeichnis und ein Raum-Verzeichnis im `/home` Verzeichnis des Users `pi` angelegt. Diese Verzeichnisse können auch an andere orte verschoben werden. Falls das gewünscht wird, dann bitte weiter unten in den Anmerkungen nachlesen.

Die ZIP-Datei wird mit Filezilla, WinSCP oder von einem USB-Stick ins `/home` Verzeichnis kopiert, anschließend mit dem Befehl

```
unzip clubdmx_code.zip
```

entpackt. Damit wird das code-Verzeichnis `clubdmx_code` angelegt und mit den aktuellen Code-Daten befüllt. Das Raum-Verzeichnis wird mit dem Befehl

```
mkdir clubdmx_rooms
```

angelegt.

Beispiele für Räume können von meiner Webseite [guntherseiser.pythonanywhere.com](http://guntherseiser.pythonanywhere.com) heruntergeladen und in das Raumverzeichnis übertragen werden.

### Alias anlegen

Diese Zeile am Ende von `/home/pi/.bashrc` anfügen:

```
alias clubdmx='/home/pi/clubdmx_code/app_start.sh'
```

### Python Module

Alle nötigen Module installieren:

```
cd clubdmx_code
./python_setup.sh
```

### Secret Key

User-Datenbank und Cookies funktionieren nur, wenn für die Webseite ein *SECRET KEY* angelegt wird. Nur um das einmal klarzustellen: Ich gehe davon aus, dass du mit Cookies einverstanden bist, wenn du mein Programm verwendest. Die Cookies dienen nur zur Funktion der Webseite und werden nicht für irgendwelche anderen Zwecke verwendet.

```
cd clubdmx_code
nano .env
# hier eintragen:
SECRET_KEY = b'84nrf97vzih47vzkd98747'
# nicht genau diesen String, sondern eine zufällige Zeichenkette,
# beginnend mit b' und abgeschlossen mit '
```

### ClubDMX starten

Die letzten Schritte sind schnell erledigt:

```
cd /home/pi/clubdmx_code
dos2unix *.sh
chmod +x *.sh
./app_start.sh start
```

### Autostart einrichten

Damit ist die ClubDMX installiert. Nun muss noch Autostart konfiguriert werden. Hier ist eine Unterscheidung zu treffen, je nachdem der Raspberry in der Konsole (wird in der Regel für ClubDMX die richtige Wahl sein) oder mit grafischer Benutzeroberfläche startet (auch dafür wird es Gründe geben).

#### Konsolen-Start:

```
sudo nano /etc/rc.local
```

hier eintragen vor der letzten Zeile (*exit 0*):

```
# clubdmx app:
su pi -c "/home/pi/clubdmx_code/app_start.sh start" &
```

#### Desktop-Start

Wenn der Pi im Desktop-Modus laufen soll, dann muss ClubDMX anders gestartet werden. Daher den Start von ClubDMX nicht in `/etc/rc.local` machen, sondern so:

```
sudo nano /etc/xdg/lxsession/LXDE-pi/autostart
# hier eintragen vor der letzten Zeile (= @screensaver...):
@/home/pi/clubdmx_code/app_start.sh start
```

### Anmerkungen

Die Verzeichnisse für Code und Räume können beliebig positioniert werden. ClubDMX findet die Verzeichnisse über Environment-Variablen.

```
CLUBDMX_CODEPATH
# (default, falls nicht gesetzt: /home/pi/clubdmx_code)
CLUBDMX_ROOMPATH
# (default, falls nicht gesetzt: /home/pi/clubdmx_rooms)
```

Anmerkung: Am Raspberry werden Environment-Variablen in der Datei `/etc/environment` eingetragen, zum Beispiel

```
export CLUBDMX_ROOMPATH="/home/pi/Documents/rooms"
```

## 9.5 Troubles schießen

Die Installation ist - beginnend bei einer leeren SD-Karte - nun doch ein wenig lang. Daher wird vielleicht der eine oder andere Fehler auftauchen. In diesem Abschnitt der Doku werde ich versuchen, Hilfestellung zur Fehlersuche zu liefern.

Meine Methode zur Fehlersuche ist, einzelne Abschnitte zu testen, um nach Möglichkeit einen Fehler in einem bestimmten Bereich zu lokalisieren. Die erste große Unterteilung ist zwischen Hardware und Software, die nächste Trennung ist zwischen OLA und ClubDMX.

### Hardware

- Netzwerk prüfen

Wird der Raspberry mit Bildschirm und Tastatur verwendet oder über das Netzwerk (WinSCP, Filezilla, ssh)?

IP Adresse des PI. Am Terminal oder der Tastatur eingeben:

```
ip address
```

Eine gültige IP Adresse kann so ermittelt werden. Diese ist wichtig, um die OLA-Seite und die Seite von ClubDMX zu finden. Die IP Adresse des Raspberry PI wird im folgenden immer mit <PI-IP> bezeichnet.

### OLA prüfen

Ich gehe davon aus, dass du am Raspberry mit Bildschirm und Tastatur sitzt. Wenn nicht, dann bitte zuerst das Netzwerk prüfen.

- Ist die OLA Webseite erreichbar? Im Browser die Seite <PI-IP>:9090 aufrufen.

Auf der OLA Webseite prüfen:

- Ist OLA richtig konfiguriert? Ist das DMX-Universum / sind die DMX-Universen richtig angelegt?
- Ist ein OSC-Input in jedem Universum angelegt?
- Sind die Outputs angelegt?

Um mithilfe der DMX Console Werte senden zu können, darf ClubDMX nicht laufen. Sonst schicken sowohl DMX Console als auch ClubDMX Werte, das geht nicht. Sollte es also mit der DMX Console möglich sein, DMX-Werte zu senden, dann kann man davon ausgehen, dass ClubDMX nicht gestartet ist.

- ClubDMX stoppen. Wenn der Alias *clubdmx* wie oben beschrieben angelegt ist, dann kann im Terminal mit dem Befehl `clubdmx stop` ClubDMX angehalten werden. Wenn nicht, dann zuerst ins `clubdmx_code` Verzeichnis wechseln:

```
cd /home/pi/clubdmx_code
./app_start.sh stop
```

Nun kann mit Sicherheit festgestellt werden, ob OLA mit der Hardware kommunizieren kann. In der DMX Console müssen nun alle DMX Kanäle bedienbar sein und die entsprechenden Geräte darauf reagieren.

### ClubDMX prüfen

Wenn im vorigen Schritt ClubDMX gestoppt wurde, dann jetzt wieder starten mit dem Befehl `clubdmx start`. Im Terminal werden nun nur wenige Meldungen ausgegeben, die Hinweise auf das korrekte Funktionieren geben.

- Ist die Webseite erreichbar?

Im Browser die Seite <PI-IP>:5000 aufrufen.

Ist die Seite erreichbar? Erscheint im Browser eine Fehlermeldung?

- Mehr Hinweise am Terminal erhalten

Im Normalfall wird ClubDMX als Daemon ausgeführt, das bedeutet, dass nur der Start im Terminal festgestellt werden kann. Ab nun wird das Programm im Hintergrund ausgeführt, Fehlermeldungen kommen also nicht aufs Terminal.

Gunicorn im Terminal und nicht als Daemon starten:

```
clubdmx stop
export GUNICORNSTART="/home/pi/.local/bin/gunicorn"
clubdmx start
```

Nun sieht man die Meldungen von ClubDMX auf dem Terminal. Das kann sicher hilfreich bei der Fehlersuche sein.



Falls Gunicorn in einem anderen als dem oben genannten Verzeichnis installiert wurde, dann muss zuerst der Pfad zu Gunicorn ermittelt werden.

- Gunicorn finden

Gunicorn ist ein Python Package, das als Webserver für ClubDMX in Verwendung ist. Packages können an unterschiedlichen Stellen installiert werden, was uns als User grundsätzlich nicht kümmern muss. Allerdings sieht es etwas anders aus, wenn ein Start-Skript einen exakten Pfad benötigt, weil zum Zeitpunkt des Startens noch kein Suchpfad definiert ist.

Genau das ist beim Autostart von ClubDMX der Fall. Daher ist es nötig, dem Skript `app_start.sh` mitzuteilen, wo sich Gunicorn befindet. In der Regel wird sich Gunicorn im Verzeichnis `/home/pi/.local/bin` befinden, aber das kann auch anders sein.

Mit dem Befehl

```
find / -name "gunicorn*" 2>/dev/null
```

erhält man die Auskunft über den Speicherort von Gunicorn. Für den Fall, dass ein anderer als der oben angegebene Pfad ermittelt wird, dann muss die Environment-Variable `GUNICORNSTART` gesetzt werden (siehe *Anmerkungen* weiter oben).

```
sudo nano /etc/environment
# hier eintragen, Beispiel:
export GUNICORNSTART="/usr/bin/gunicorn3"
```



---

## Entwicklungsschritte

---

Die Idee zur Programmierung einer Lichtsteuerung trage ich schon lange Zeit mit mir herum. Meine erste Software zur Lichtsteuerung stammt aus 1990-2000, sie habe ich für meine Projekte als Lichttechniker auch gleich im Einstz erproben können.

Hier gab es eine Software in C, die für eine Einzel-Anfertigung eines Lichtpultes entwickelt wurde. Der Entwickler der Hardware Günter Humpel lebt leider nicht mehr, ich möchte ihm hier an dieser Stelle meine große Wertschätzung für seine Genialität ausdrücken.

Die Idee zu einer neuen Software schlummert seit dem Umstieg auf kommerzielle Lichtpulte, diverse Versuche zur Realisierung verliefen ohne Erfolg.

Als wesentlichen Zwischenschritt zur Entwicklung einer selbständigen Software kann das Projekt **olaremote** bezeichnet werden, für das ich erstmals Python und [Open Lighting Architecture](#) zur Fernsteuerung von Beleuchtung eingesetzt habe.

Der entscheidende Impuls zu **ClubDMX** kam von Paco Gonzalez-Rivero beim Gespräch während einer Weihnachtsfeier. Paco überzeugte mich, dass zeitgemäße Software ein Web-Interface zur Benutzerkontrolle haben muss. Damit ist die Unabhängigkeit von Betriebssystemen und eine breite Palette an Möglichkeiten für die Benutzer gegeben, ohne auf spezifische, für ein Betriebssystem zugeschnittene Einzellösungen gebunden zu sein. So entstand die Idee einer simplen Lichtsteuerung mit Web-Interface bei einem Smalltalk auf einer Weihnachtsfeier.

### 10.1 Erste Idee

Die erste Idee war, Software-Fader für Lichtszenen zu programmieren. Das ist die Grundidee: Lichtszenen in Kombination mit einer Browser-Steuerung.

Dazu braucht es einiges an Vorbereitung.

- Wie werden die Daten gespeichert, Datenbank?

Ich habe mich für die Verwendung von CSV-Tabellen entschieden, um möglichst unabhängig von Betriebssystem-Vorgaben zu sein. Hier war das Thema zu lösen, wie gegebenenfalls Änderungen rückgängig gemacht werden können.

- Klassische Lichttechnik arbeitet nach dem Prinzip der Kanal-Steuerung. Zeitgemäße Lichttechnik hat mehr zu bieten, hier gibt es unterschiedliche Möglichkeiten für unterschiedliche Geräte. Das bedeutet einen erhöhten Level beim Einstieg in die Programmierung, es braucht einen Patch, eine Unterscheidung zwischen HTP- und LTP-Attributen und einige andere Konzepte mehr.

Ich wähle den spannenderen Weg und entscheide mich für die zeitgemäße Umsetzung der Steuerung.

- Welche Programmiersprache(n), welche Tools?

Hier sind einige Stichworte für meine Entscheidungen: Visual Studio Code als Programmieroberfläche. Python und Flask für die Grundlagen und die HTML-Seiten. Jinja2 für die HTML-Templates. Javascript und JQuery. Bootstrap für das Design.

## 10.2 Die Entwicklungsschritte

- Version 0.1

Der Grundstein ist gelegt: Es gibt eine Patch als Kernstück. Hier wird die Hardware in die Software eingebunden. Und es gibt eine fixe Anzahl von 10 Lichtstimmungen, die über Fader gesteuert werden können. Alle Daten zu Patch, Heads (= Geräte) und Cues (= Lichtstimmungen) werden in CSV-Dateien geschrieben, da dies ein Datenformat ist, das auf jedem Betriebssystem bearbeitbar ist.

Das Konzept der virtuellen Dimmer existiert bereits jetzt: Für LED Geräte gibt es eine Trennung zwischen Intensität und Farbe.

- Version 0.2

Die Anpassung an verschiedene Bedürfnisse wird wesentliches Thema. Das heißt:

- eine erste Variante von Configurationsdatei entsteht.
- Die Anzeige der CSV-Tabellen und deren Editierbarkeit werden überarbeitet.
- Die Web-Funktionen befinden sich allesamt in einer immer länger werdenden Datei.

- Version 0.3

Die Datenbankstruktur wird einheitlich und damit universeller. Jede Kategorie an Daten erhält ein eigenes Verzeichnis. Die Anzahl der Cues ist konfigurierbar und nicht mehr auf 10 beschränkt. Damit ist der Grundstein für eine modulare Struktur gelegt, allerdings wird die Datei mit den Web-Funktionen zu einem Monster, weil mir die Infos zur Strukturierung fehlen. In dieser Programmstruktur ist das Limit erreicht.

Messages: Die Rückmeldungen der Software werden überarbeitet und einheitlicher.

- Version 0.4

Sehr wichtige Neuerungen:

- Die Stage  
Damit ist eine erste Geräte Einzelsteuerung geschaffen. Auf der Stage-Fläche können Symbole, die für Geräte (Heads) oder Text stehen, platziert werden.
- Neue Programmstruktur: Die Webseiten werden in Module aufgeteilt.  
Für die Benutzer zwar nicht relevant, aber mit dieser Struktur werden viele Überschneidungen und Nebeneffekte vermieden.
- Neues Datenkonzept: Alle Daten werden in einem **Projekt** zusammengefasst. Diese Zusammenfassung ermöglicht das Wechseln in verschiedene Einheiten von Datenbanken.

- Version 0.5

Aus dem **Projekt** wird der **Raum**. Dieser Begriff ist verständlicher und beschreibt besser, worauf die Datenstruktur abzielt. In einem Raum können mehrere **Konfigurationen** abgespeichert werden, die sich die Datenbanken teilen, wie Cues, Stage, Patch etc.

Der **Raum** wird mit Environment-Vaiable vor Start der App bestimmt. Ein Wechsel des Raums ist noch nicht implementiert.

Für die Tabellen gibt es erweiterte Editierfunktionen: Zelleneditor und Zeilenselektion. Mittels Selektion können Zeilen mit copy/paste ausgeschnitten und vervielfältigt werden.

- Version 0.6

Die in dieser Version wichtigste Neuerung ist das Login, damit sind die Webseiten und die Steuerung vor unbefugtem Zugriff geschützt.

Eine weitere Neuerung ist die Midi-Steuerung. Die Cues können mit einem Midi-Faderboard (Korg nanoKontrol 1 oder 2) geregelt werden.

Neben der Stage-Seite gibt es nun auch eine Stage-Kompakt Seite, die für die Verwendung von kleinen Touch-Bildschirmen konzipiert ist. Da Selektion und Verschieben nicht verträglich mit Seite-Verschieben ist, braucht es diese Möglichkeit.

Neu hinzu kommen auch Buttons: Mit Buttons können Cues zeitabhängig ein- und ausgefadet werden.

- Version 0.7

Ab nun ist eine erweiterte Raum-Bedienung möglich: Das Wechseln des Raums, Backup und Restore auf USB-Stick. Diese USB-Funktionen arbeiten plattformübergreifend auf Windows und Linux.

Das Editieren der Konfiguration wird wesentlich erleichtert: Es geht nun in einem Formular, wo für jedes Feld die entsprechenden Optionen ausgewählt werden können.

Eine neue Stage-Ansicht kommt hinzu: Der Single-Modus. Hier können die Attribute aller Geräte einfach bedient werden, die geänderten Attribute sind im *Topcue* und können so abgespeichert werden.

- Version 0.8

Hier kommen wieder einige neue Funktionen dazu.

- Es gibt eine Demo-Version auf <<https://guntherseiser.pythonanywhere.com>>
- Änderungen in Fadern, Buttons und der Stage werden auf allen offenen Browser-Fenstern aktualisiert.
- Es gibt eine neue Seite, die den DMX-Output anzeigt und regelmäßig aktualisiert
- Es gibt eine eigene Seite für die Dokumentation.
- Die Benutzer-Rollen werden detailliert unterschieden. Somit haben die Benutzer entsprechend ihrer Rolle unterschiedliche Möglichkeiten zur Interaktion und bekommen auch unterschiedliche Webseiten zur Ansicht. Siehe *Benutzer*.
- Die Seite zum Benutzer-Management wird überarbeitet.
- Das Speichern des Topcue wird vereinfacht. Nun kann der Topcue als Fader oder Button abgespeichert werden.
- Die Raum-Sicherung wird erweitert um upload/download.

- Version 0.9

Diverse Vereinfachungen beim Benutzen, zum Beispiel Auswahlfelder und Inhalts-Überprüfung beim Editieren der Felder in der Datenbank.

Neue Funktionen:

- Buttons mit verschiedenen Aufgaben: Schalter, Taster, Auswahlschalter
- OSC Input
- In den Seiten Stage, Fader und Button: Einrichtungs-Symbol für Sprung zur jeweiligen Setup-Tabelle.
- Raum-Management: umbenennen, unter anderem Namen sichern, löschen.
- Die Exec-Seite beinhaltet Buttons und Fader. Ist die Startseite für den Basic-Benutzer.
- Fader und Buttons müssen nicht zwingend auf der Exec-Seite zu sehen sein, sie können auch auf zusätzlichen Seiten platziert werden.
- Start-Cue: Es kann ein Cue konfiguriert werden, der beim Start von ClubDMX ausgeführt wird.
- Patch-Formular zum Erzeugen mehrerer Heads in einem Schritt. siehe [Neuen Patch erstellen](#).
- Kommandozeilen-Skripte `sendosc.sh` und `sendosc.bat` zum Senden von OSC-Befehlen, siehe [Einrichten von zeitabhängiger Steuerung mit Crontab \(in Linux\)](#).
- Stage: Geräte-Fader werden in einem Fenster angezeigt, Selektion kann gemeinsam verschoben werden.
- Stage: In der Config (Bedienelemente) wird eine Default-Stage angegeben. Die angezeigte Stage kann davon abweichen: Die angezeigte Stage wird in einer Session-Variablen gespeichert.

# KAPITEL 11

---

## Impressum

---

Gunther Seiser Mascagnigasse 27 5020 Salzburg

Telefon: +43 699 8188 1989

[gunther.seiser.63@gmail.com](mailto:gunther.seiser.63@gmail.com)

Erstellt am 06.11.2021