



HÖHERE TECHNISCHE BUNDESLEHRANSTALT Wien 3, Rennweg  
IT & Mechatronik

HTL Rennweg :: Rennweg 89b  
A-1030 Wien :: Tel +43 1 24215-10 :: Fax DW 18

# Diplomarbeit

## **Mediatrix** **Ausgeschriebener Titel der Diplomarbeit**

ausgeführt an der  
Höheren Abteilung für Informationstechnologie/Medientechnik  
der Höheren Technischen Lehranstalt Wien 3 Rennweg

im Schuljahr 2017/2018

durch

**Nußbaumer Dominik**  
**Scharwitzl Clemens**  
**Steiner Florian**

unter der Anleitung von

Fink Andreas  
Stimpfl Franz

Wien, 21. Februar 2018



# Kurzfassung

Darum geht es.



# Abstract

Thats why – the translated text „Kurzfassung“ (this should be a translation).



# Ehrenwörtliche Erklärung

Ich erkläre an Eides statt, dass ich die individuelle Themenstellung selbstständig und ohne fremde Hilfe verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Wien, am 21. Februar 2018

---

Mitarbeiter Eins

---

Mitarbeiter Zwei

---

Mitarbeiter Drei





# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>xi</b>
----------------------------	-----------

<b>Abbildungsverzeichnis</b>	<b>xiii</b>
------------------------------	-------------

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Problemstellung . . . . .	1
1.2	Ziel der Arbeit . . . . .	1
1.3	Abgrenzung und Voraussetzungen . . . . .	1
1.4	Aufbau . . . . .	1
<b>2</b>	<b>Hardware</b>	<b>3</b>
2.1	Gehäuse . . . . .	3
2.2	Raspberry Pi . . . . .	3
2.3	Ein- und Ausschlatversögerung . . . . .	3
2.4	Lautsprecherschutzschaltung . . . . .	3
2.5	Infrarotsender . . . . .	3
2.6	DMX Interface . . . . .	3
2.7	Verkabelung . . . . .	3
2.8	Gehäusebelüftung . . . . .	3
2.9	Stromversorgung . . . . .	3
2.10	Anschlüsse für den Anwender . . . . .	3
<b>3</b>	<b>Betriebssystem</b>	<b>5</b>
3.1	Raspbian . . . . .	5
3.2	Sicherheit . . . . .	5
3.3	Ola . . . . .	5
3.4	WiringPi . . . . .	5
<b>4</b>	<b>Backend</b>	<b>7</b>
4.1	PHP Extension . . . . .	7
4.2	Websocket . . . . .	7
4.3	LDAP . . . . .	7
4.4	DMX . . . . .	7
4.4.1	Funktionsweise des Protokolls . . . . .	7
4.4.2	Anwendung in diesem Projekt . . . . .	8
4.5	Infrarot . . . . .	9
4.6	Mischpult . . . . .	9
4.7	SQLite . . . . .	9

<b>5</b>	<b>Frontend</b>	<b>11</b>
5.1	CSS . . . . .	11
5.1.1	Flexbox . . . . .	11
5.2	jQuery . . . . .	14
5.3	Design . . . . .	14
<b>6</b>	<b>Zusätzliches</b>	<b>15</b>
6.1	Bedienungsanleitung . . . . .	15
6.2	Beleuchtungskonzept Konferenzsaal . . . . .	15
<b>A</b>	<b>Anhang 1</b>	<b>17</b>
	<b>Literaturverzeichnis</b>	<b>19</b>

# Tabellenverzeichnis



# Abbildungsverzeichnis

4.1 Darstellung des DMX-Signals . . . . .	8
5.1 alt text . . . . .	13



# **1 Einleitung**

## **1.1 Problemstellung**

## **1.2 Ziel der Arbeit**

## **1.3 Abgrenzung und Voraussetzungen**

## **1.4 Aufbau**





## **2 Hardware**

### **2.1 Gehäuse**

### **2.2 Raspberry Pi**

### **2.3 Ein- und Ausschlatversögerung**

### **2.4 Lautsprecherschutzschaltung**

### **2.5 Infrarotsender**

### **2.6 DMX Interface**

### **2.7 Verkabelung**

### **2.8 Gehäusebelüftung**

### **2.9 Stromversorgung**

### **2.10 Anschlüsse für den Anwender**



## **3 Betriebssystem**

### **3.1 Raspbian**

### **3.2 Sicherheit**

### **3.3 Ola**

### **3.4 WiringPi**



## 4 Backend

### 4.1 PHP Extension

### 4.2 Websocket

### 4.3 LDAP

### 4.4 DMX

Das **DMX** (Data Multiplexed) Protokoll wurde erstmalig durch das USITT (United States Institute for Theatre Technology) definiert. Es beschreibt die Steuerung von bis zu 512 Dimmern über eine serielle Verbindung. Das Protokoll findet hauptsächlich in der Theater- und Bühnenbeleuchtungstechnik Anwendung. Hierbei werden die Scheinwerfer über ein Busnetzwerk mit einem Wertebereich von 8-bit gesteuert. Es gilt als State-of-the-art und ist durch die DIN 56930-2 Norm definiert.[2]

#### 4.4.1 Funktionsweise des Protokolls

Die Informationen werden im DMX Protokoll digital übertragen, wobei hier zwischen einer positiven und negativen Spannung von ungefähr 2,5 Volt unterschieden wird. Ein Einsen entspricht einer positiven Spannung für 4  $\mu$ s und ein Nuller einer negativen Spannung für die selbe Zeitspanne. DMX verwendet eine 8-bit Datenlänge. Der Wertebereich eines Kanals liegt also zwischen Null und 255. Die jeweiligen Werte für jeden Kanal werden nacheinander gesendet. Am Anfang jedes Signals wird eine Reset-Sequenz gefolgt von einem Startbyte gesendet.[1] Siehe Bild 4.1.

Der Vorteil dieses Protokolls ist, dass alle Empfänger nur an ein Kabel angeschlossen werden müssen und die meist schon vorhandene XLR-Verkabelung genutzt werden kann.[2]

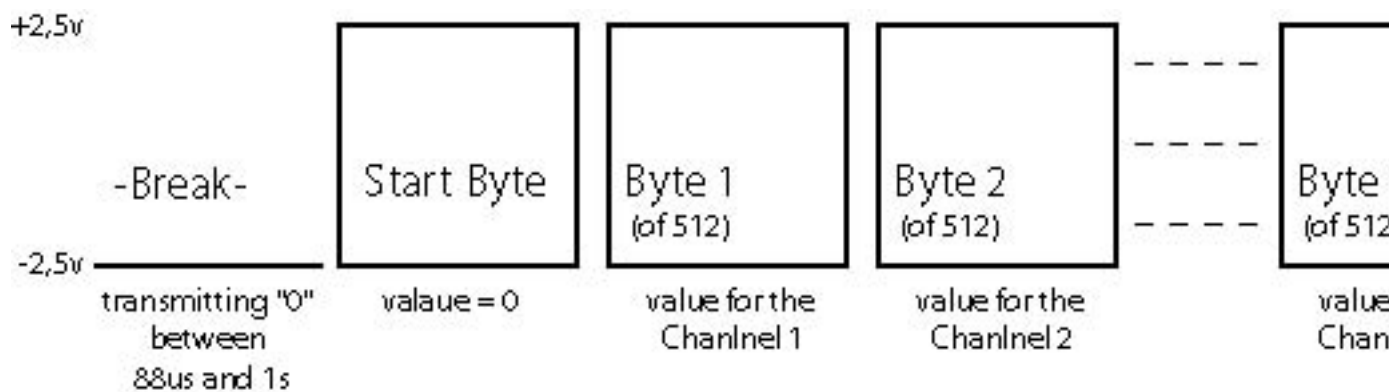


Abbildung 4.1: Darstellung des DMX-Signals

Ein Nachteil ist, dass bei vielen genutzten Kanälen das Signal und somit auch die Refreshrate sehr gering wird. Das bedeutet, dass in der Praxis DMX mit weniger Geräten betrieben werden sollte.[2]

#### 4.4.2 Anwendung in diesem Projekt

In diesem Projekt wurde DMX als Schnittstelle zur Beleuchtung integriert. Dadurch bleibt das System flexibel, so gibt der Administrator bei der Installation nur an welcher Kanal, welchem Scheinwerfer zuzuordnen ist. Weiters wird dadurch auch garantiert, dass das System mit allen DMX-Standards konformen Scheinwerfern funktioniert. Ein weiterer Vorteil, der sich durch die Integration des Protokolls ergibt, ist, dass nicht nur Dimmer, sondern auch RGB- und RGBW-Scheinwerfer gesteuert werden können. Die Art des Scheinwerfers wird an der Anzahl der durch den Administrator angegebenen Kanäle pro Scheinwerfer bestimmt. So ist es auch möglich, dass verschiedene Typen gemischt werden können.

Wenn vom User der Befehl zur Änderung einer Einstellung an einem der Scheinwerfer gesendet wird, wird im Hintergrund das zu diesem Scheinwerfer passende Scheinwerfer-Objekt aufgerufen, welches über die Php-Extension (siehe 4.1) und Ola (siehe ??) den entsprechenden Wert des DMX-Kanals ändert.

---

## **4.5 Infrarot**

## **4.6 Mischpult**

## **4.7 SQLite**





# 5 Frontend

## 5.1 CSS

### 5.1.1 Flexbox

Flexbox, offiziell CSS Flexible Box Layout Module, ist eine neue Art und ein neues Konzept um eindimensionale Layouts auf Webseiten umzusetzen. Die herkömmliche Art Objekte auf einer Webseite zu positionieren ist, fixe Positionen und Maße zu vergeben.

Doch bei Flexbox werden bestimmte Regeln festgelegt, diese machen das Verhalten der Webseite vorhersagbar bei einer Veränderung der Bildschirmgröße. Anschließend ist es dem Browser überlassen, die Breite, Höhe, Position und Anordnung zu wählen.

#### 5.1.1.1 Das Konzept

Die Grundidee ist es, dem Flex-Container die Möglichkeit zu geben, die Maße der Elemente so zu verändern, dass der Platz auf unterschiedlichen Bildschirmauflösungen bestmöglich ausgenutzt ist. Um das zu erzielen lässt das Elternelement die Kindelemente je nach Bedarf wachsen oder schrumpfen.

#### 5.1.1.2 technische Spezifikation

Innerhalb eines `<div>` Tags können die einzelnen Elemente ihre Größe “flexibel” verändern. Sie wachsen, um freien Platz zu verwenden oder schrumpfen, um innerhalb des Elternobjekts zu bleiben und einen Overflow zu vermeiden. Der große Vorteil des Flexbox Layouts ist die Richtungsunabhängigkeit. Dadurch ist es sehr flexibel, was Orientierungsänderungen bei mobilen Geräten oder Auflösungsänderungen auf Desktop Geräten betrifft.

### 5.1.1.3 Erklärung anhand eines realen Beispiels

Auf dem Dashboard soll eine seitliche Navigation angezeigt werden, die auf mobilen Geräten an den unteren Rand des Bildschirms wandert, siehe Abbildung 1.

Mithilfe von Flexbox ist dieses Verhalten einfach zu erzielen. Ich erstelle ein Elternelement mit folgenden Eigenschaften:

```
.parent{  
  display: flex;  
  overflow: hidden;  
}
```

Die Kindelemente dieser Flexbox werden auf der horizontalen Hauptachse ausgerichtet. Der Overflow auf der X- und Y-Achse wird ausgeblendet. Die Navigation auf der Seite ist in folgendem Code-Block beschrieben.

Dieses Element ist durch `order:1` das erste Element in der Flexbox. Der Overflow auf der Y-Achse ist versteckt, um die Leiste zu fixieren. Weiters werden die Elemente innerhalb vertikal und horizontal zentriert und sind entlang der Y-Achse positioniert.

```
.side-nav{  
  display: flex;  
  order: 1;  
  justify-content: center;  
  align-items: center;  
  flex-direction: column;  
}
```

Das Inhaltselement hat `order:2` damit es neben dem ersten auf der X-Achse positioniert wird. Ebenso ist der Overflow auf der Y-Achse versteckt.

```
.content{  
  overflow-y: hidden;  
  display: flex;  
  justify-content: center;  
  flex-direction: column;  
  order: 2;  
}
```

Damit die Navigation auf mobilen Geräten am unteren Rand positioniert ist, benötigen wir eine Media Query. Mithilfe dieser können CSS-Stile anhand von verschiedenen Eigenschaften wie z.B. Bildschirmauflösung oder Seitenverhältnis manipuliert werden. Im untenstehenden Code-Block wird dies veranschaulicht. Indem wir die Hauptachse



des Flexbox Elternelements auf die Y-Achse ändern, werden die beiden Kindelemente nun vertikal verteilt. Damit nun auch die Navigation unter dem Inhalt positioniert ist ändern wir die order auf 2. Weiters müssen die Höhe und Breite angepasst werden.

```
@media (max-width: 576px){  
  .parent{  
    flex-direction: column; //changed  
  }  
  
  .side-nav{  
    order: 2;           //changed  
    width: 100vw;       //changed  
    height: 66px;       //changed  
  }  
}
```

## 5.2 jQuery

## 5.3 Design

## **6 Zusätzliches**

### **6.1 Bedienungsanleitung**

### **6.2 Beleuchtungskonzept Konferenzsaal**



# A Anhang 1

was auch immer: technische Dokumentationen etc.

Zusätzlich sollte es geben:

- Abkürzungsverzeichnis
- Quellenverzeichnis (hier: Bibtex im Stil plaindin)





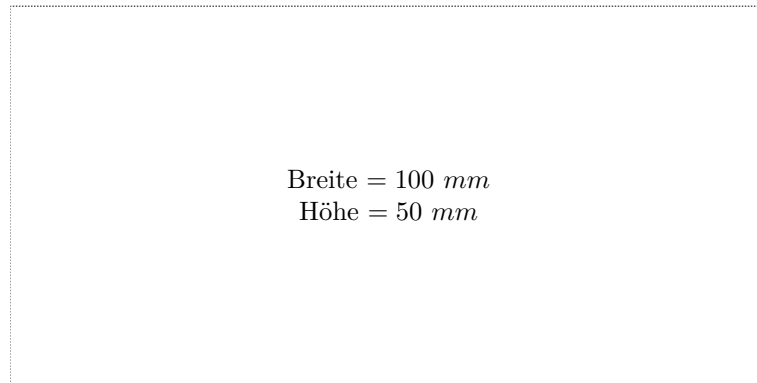
# Literaturverzeichnis

[1] *DMX german*. <http://www.theater-technisch-lab.nl/dmxdu.htm>

[2] *DMX512/1990*. <http://www.soundlight.de/techtips/dmx512/dmx512.htm>



— Druckgröße kontrollieren! —



— Diese Seite nach dem Druck entfernen! —

Diese  
Seite  
nach dem  
Druck  
entfer-  
nen!