

# Technische Dokumentation (DE)

Dieses Handbuch beinhaltet technische Zusatzinformationen zur Erstellung von Projekten und Schnittstellen. Zielgruppe sind technische Administratoren welche die Projekte und System vor Ort einrichten und an zusätzliche Programme anbinden.

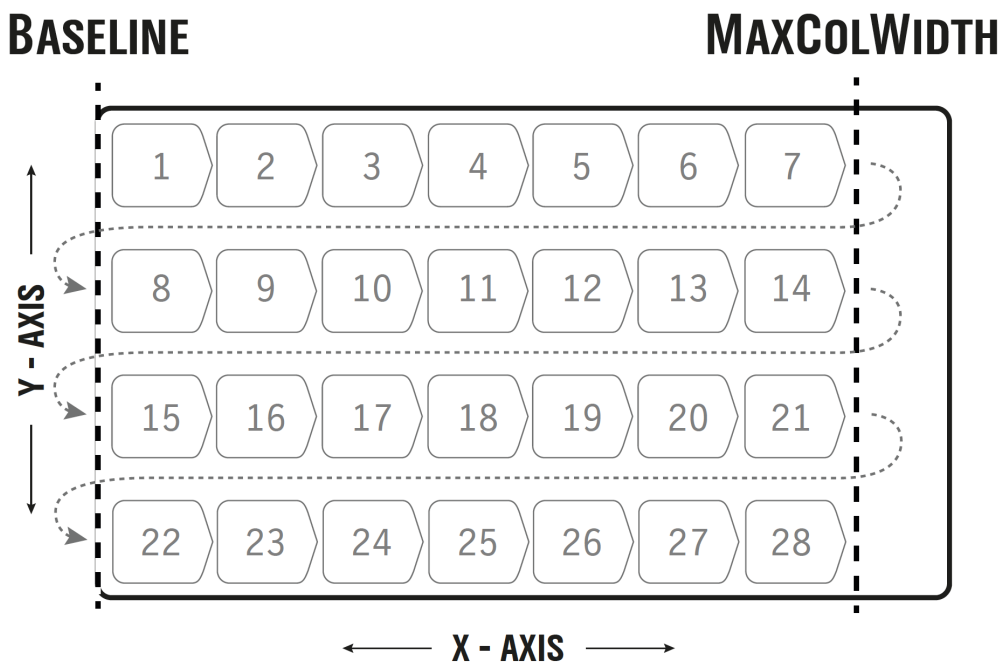
- XML Projektdatei
  - Individuelles XML Mapping
- Schnittstellen
  - REST-API
  - DMX512 ArtNET
  - MIDI
- LiveText
  - Allgemeine Informationen
  - Einrichtung eines Tags
  - Ansteuerung des Textes
  - KINEXON Connector

# XML Projektdatei

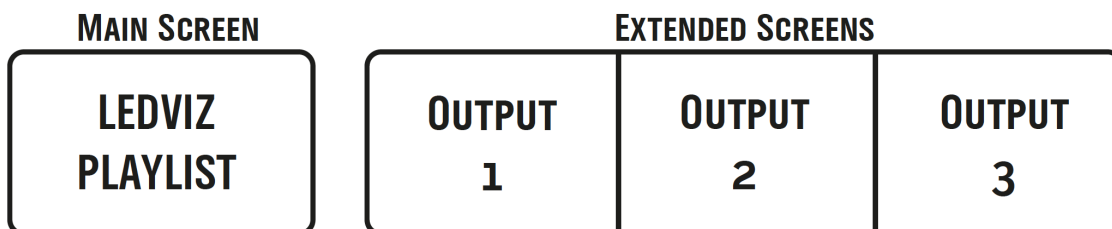
Am Anfang einer jeden Konfiguration muss ein Mapping erstellt werden, also eine Konfiguration die dem System sagt, welchen Banner er wo darstellten muss. Für die Basics reicht der Dialog beim Erstellen eines neuen Projektes in der Regel aus. Es gibt aber Konfigurationen die z.B. über mehrere Bildschirme gehen. Hier muss man manuell in die Konfiguration eingreifen. Die folgenden Seiten geben einen Überblick über die Verfügbaren TAGS in der XML Datei und ihre Auswirkungen

# Individuelles XML Mapping

Die Position von LED-Panels wird normalerweise automatisch berechnet. Auf die linke Seite ist die Grundlinie. Nach dem Erstellen einer Datei ist es immer Null ( $x = 0$ ). Auf der rechten Seite wird es auf den errechneten rechten Rand begrenzt. Dieser Wert ist sichtbar in jedem XML unter dem Tag `<MAXCOLWIDTH>`



Um eine Konfiguration mit mehreren Bildschirmen zu handhaben, muss das Ausgabefenster auf einem oder mehreren erweiterte Bildschirm wird so angepasst, dass er alle Bildschirme abdeckt. Da sich die Ausgabe immer über ein Fenster erstreckt, müssen hierzu die erweiterten Bildschirme nebeneinander in einer Reihenfolge von links nach rechts angeordnet werden. Diese Einstellungen können Sie in den Anzeigeeinstellungen von Windows vorgenommen werden.



Als Hardwarelösung kann hier eine Matrox TripleHead2Go oder DualHead2Go zum Einsatz kommen. Durch die Extender von Matrox wird die Auflösung an einem DVI / HDMI oder DSP Anschluss erweitert und anschließend wieder in einzelne Bildschirm Signale aufgeteilt. Alternativ, wenn die Grafikkarte mehrere Ausgänge hat, kann das Ausgabefenster von LEDVIZ auch in der

Höhe oder Breite geändert werden, so das es die komplette Ausgabe ausfüllt.

## Ausgabebereiche neu Positionieren

Jede Anzeigefläche ist mit der linken Seite an der vorherigen Anzeigefläche ausgerichtet. Bei der Berechnung der Position wird der Y-Wert, also die vertikale Positionierung, solange beibehalten bis ein Zeilenumbruch erfolgen muss. Mit dem Zeilenumbruch wird die Anzeige um seine Höhe nach unten verschoben und wieder neu links mit  $X = 0$  angesetzt. Es gibt grundsätzlich 3 Werte um eine Anzeigefläche direkt an eine andere Position zu bewegen:

<code>&lt;POSX&gt; ? &lt;/POSX&gt;</code>	Weißt der Anzeigefläche einen neuen horizontalen X Wert zu
<code>&lt;POSY&gt; ? &lt;/POSY&gt;</code>	Weißt der Anzeigefläche einen neuen vertikalen Y Wert zu
<code>&lt;BASEX&gt; ? &lt;/BASEX&gt;</code>	Weißt der Berechnungsroutine einen neuen horizontalen X Wert für die linke Seite im Fall eines Zeilenumbruchs zu

Bitte beachten Sie, dass die XML Tags immer Großgeschrieben werden. Wenn die Werte einer Anzeigefläche geändert wurde, sind diese dann die Ausgangsbasis für alle weiteren Berechnungen, d.h. das folgende Element wird dort wieder rechts angesetzt usw. Die rechte Seite für den Zeilenumbruch wird dabei immer über folgende Formel berechnet:

$$\text{Rechte Seite} = \text{<BASEX>} + \text{<MAXCOLWIDTH>}$$

Das bedeutet, wenn Sie den `<BASEX>` Wert ändern, ändert sich automatisch auch immer die Position der rechten Seite, die den Zeilenumbruch erzeugt. Sie können den `<BASEX>` Wert so oft ändern wie Sie möchten, es gibt hier keine Begrenzungen.

## Erzeugen eines Umbruchs in einer Anzeigefläche

Manchmal ist es notwendig, die Ausgabe einer Anzeigefläche zu unterbrechen und an einer anderen Stelle fortzusetzen. Einen Umbruch innerhalb einer Anzeigefläche kann durch folgenden XML Tag erzwungen werden:

<code>&lt;BREAK&gt; ? &lt;/BREAK&gt;</code>	Definiert den Umbruch innerhalb einer Anzeigefläche nach definierter Anzahl an Pixeln
---	---

Nur der Umbruch selbst reicht an der Stelle aber noch nicht aus. Im gleichen Schritt müssen nun auch neue Koordinaten zum Fortsetzen der Ausgabe angegeben werden:

<code>&lt;BREAKX&gt; ? &lt;/BREAKX&gt;</code>	Weißt einen neuen horizontalen X Wert für das fortsetzen der Anzeigenfläche zu
<code>&lt;BREAKY&gt; ? &lt;/BREAKY&gt;</code>	Weißt einen neuen vertikalen Y Wert für das Fortsetzen der Anzeigenfläche zu

Ebenfalls kann hier auch direkt gleich die Zuweisung eines neuen `<BASEX>` Wertes vorgenommen werden, dann ab der umzubrechenden Stelle zu verwenden ist:

<code>&lt;BREAKBASEX&gt; ? &lt;/BREAKBASEX&gt;</code>	Zuweisung eines neuen <code>&lt;BASEX&gt;</code> Wertes durch den Umbruch
---	---

Auch unterstützt werden bis zu 2 weitere Umbrüche:

<code>&lt;BREAK2&gt; ? &lt;/BREAK2&gt;</code>	Definiert einen zweiten Umbruch innerhalb einer Anzeigenfläche nach definierten Anzahl an Pixeln
<code>&lt;BREAK2X&gt; ? &lt;/BREAK2X&gt;</code>	Weißt einen neuen horizontalen X Wert für das fortsetzen der Anzeigenfläche zu
<code>&lt;BREAK2Y&gt; ? &lt;/BREAK2Y&gt;</code>	Weißt einen neuen vertikalen Y Wert für das Fortsetzen der Anzeigenfläche zu

bzw.

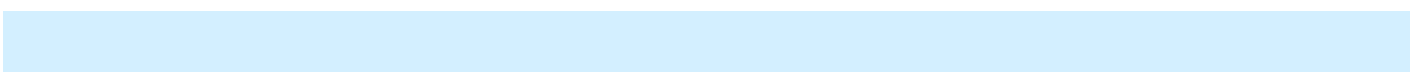
<code>&lt;BREAK3&gt; ? &lt;/BREAK3&gt;</code>	Definiert einen dritten Umbruch innerhalb einer Anzeigenfläche nach definierten Anzahl an Pixeln
<code>&lt;BREAK3X&gt; ? &lt;/BREAK3X&gt;</code>	Weißt einen neuen horizontalen X Wert für das fortsetzen der Anzeigenfläche zu
<code>&lt;BREAK3Y&gt; ? &lt;/BREAK3Y&gt;</code>	Weißt einen neuen vertikalen Y Wert für das Fortsetzen der Anzeigenfläche zu

## Erzeugen eines vorzeitigen Zeilenumbruchs

Für den Fall das die Ausgabe nicht am Standardwert des Zeilenumbruches geändert werden soll, sonder z.B. früher oder auch später, kann einer Anzeigenfläche ein individueller Rechter Rand für den Umbruch mitgeteilt werden.

<code>&lt;MAXX&gt; ? &lt;/MAXX&gt;</code>	Zuweisung eines neuen Wertes für einen Zeilenumbruch
---	--

Dieser `<MAXX>` Wert gilt aber ausschließlich für das jeweilige Element das diesen Wert beinhaltet. Alle anderen Anzeigenflächen richten sie weiterhin nach den Standardwerten aus em XML



Wenn früher in der Zeile umgebrochen werden soll, und die anderen Werte wie <BASEX> sich nicht ändern, kann damit oft auch der einfache <BREAK> ersetzt werden. Wenn möglich, sollte dem <MAXX> Wert immer der Vorzug vor einem <BREAK> gegeben werden!

## Sonderfälle

Da <POSX> oder <POSY> Werte mit einem 0 Wert, nicht angegeben werden müssen, gibt es einen Fall der zu einem logischen Problem bei der Verarbeitung der XML führt. Es ist kein direkter Rücksprung auf die Koordinaten  $X = 0$  und  $Y = 0$  möglich. In diesem Fall kann der Workaround angewendet werden und der <BASEX> Wert auf -1 und der <POSX> auf +1 gestellt werden. Mit der darauf folgenden Anzeigenfläche, können über die Definition des genauen <POSX> Wert diese dann wieder richtig gestellt werden. Es muss immer mindestens ein Wert von <POSX> oder <POSY> oder <BASEX> größer oder kleiner 0 angeben sein damit diese 3 Werte verarbeitet von LEDVIZ verarbeitet werden.

## XML Beispiele

```
<SEGMENT1 id="1" >
  □<POSX>1920</POSX>
  □<POSY>0</POSY>
  □<BASEX>1920</BASEX>
  □<NAME>Panel 2</NAME>
  □<WIDTH>1080</WIDTH>
  □<HEIGHT>60</HEIGHT>
</SEGMENT1>
```

oder

```
<SEGMENT4 id="4" >
  □<BREAK>512</BREAK>
  □<BREAK>1920</BREAK>
  □<BREAKY>0</BREAKY>
  □<BREAKBASEX>1920</BREAKBASEX>
  □<NAME>Panel 5</NAME>
  □<WIDTH>1080</WIDTH>
  □<HEIGHT>60</HEIGHT>
</SEGMENT4>
```



# Schnittstellen

Die folgenden Seiten beinhalten eine Beschreibung über der verschiedenen Schnittstelle für die Steuerung von LEDVIZ von außen. Über diese Schnittstellen können externe Trigger, z.B. über vMix oder auch jede andere Anwendung an LEDVIZ übergeben werden. Die Ausgabe der LED Bande kann dadurch im Verbund mit anderen Programmen und Anzeigen (z.B. Videowand) gesteuert werden.

# REST-API

Im folgenden wird näher auf die Funktionalität und Befehle der Rest-API eingegangen.

## Anwendungsbereich

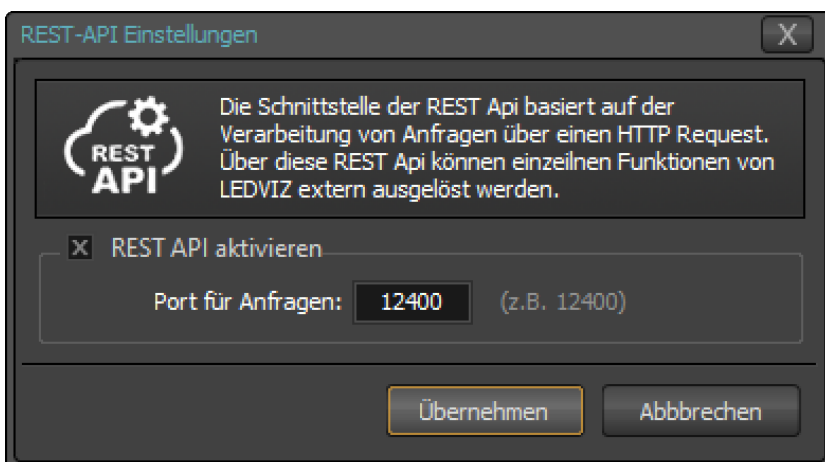
Die Rest-API stellt einen einfachen Zugriff für externe Anwendungen zur Verfügung. Der Zugriff erfolgt dabei über das HTTP Protokoll. Befehle werden dabei wie der Aufruf einer Webseite an LEDVIZ gesendet. Die aufgerufenen Seite werden dabei als Befehle interpretiert. Anwendungen wie vMix sind in der Lage solche HTTP Request zu senden.

Wird z.B. ein bestimmtes Video gespielt, kann dies mit einem solchen Aufruf verknüpft werden, um gleichzeitig den richtigen Content auf der Videowand und der LED Bande anzuzeigen. Dies kann zum Beispiel bei der Vorstellung einer Mannschaft vor einem Spiel genutzt werden.

## Einstellungen

In Bezug auf die Rest-API gibt es nicht viel einzustellen. Den Dialog dazu erreichen Sie oben über die Menüleiste im Hauptprogramm unter dem Punkt "Steuerung".

Grundsätzlich stehen 2 Parameter zur Verfügung:



Über die Option "Rest-API aktivieren" wird der Port zur externen Ansteuerung geöffnet. Im Gegensatz zu den anderen Schnittstellen, ist diese Schnittstelle immer aktiv. Zusätzlich haben sie hier die Möglichkeit den gewünschten Port anzugeben.

Als Standard ist hier der Port 12400 eingestellt.

Der externe Trigger muss hierbei über den ausgewählten Port und das HTTP Protokoll an LEDVIZ weitergeleitet werden.

Secured Socket Layer Anfragen über HTTPS werden von LEDVIZ aufgrund des fehlenden Zertifikates nicht unterstützt

Folgende Befehle stehen über die Rest-API zur Verfügung:

## Befehle

### Sequenzen abspielen

```
http://localhost:12400/play  
http://localhost:12400/play/5  
http://localhost:12400/play/5?beispiel.xml
```

Mit dem ersten Befehl wird nur der Play-Button getriggert, während der zweite Befehl direkt die Sequenz 5 im aktuellen Projekt ansteuert. Der letzte Befehl erweitert den Trigger um ein Projekt.

Über den Play Befehl kannst auch eine Sequenz mit DMX oder MIDI ID triggern.

**Wichtig:** Die Suche nach dieser ID erfolgt immer in allen Projekten!

```
http://localhost:12400/play/dmx/25  
http://localhost:12400/play/midi/ABC
```

Im Gegensatz dazu steht das Abspielen über vordefinierte Tasten. Genauso wie im Programm direkt, können auch die Short-cuts, also F1 bis F20 und 0 bis 9, angesprochen werden:

```
http://localhost:12400/key/F1
```

### Abspielen pausieren

Folgender Befehl triggert den Pause-Button in der Anwendung um den Abspielvorgang anzuhalten:

```
http://localhost:12400/pause
```

### Abspielen stoppen

Folgender Befehl triggert den Stop-Button in der Anwendung um den Abspielvorgang komplett anzuhalten:

```
http://localhost:12400/stop
```

## Abspielen fortsetzen

Folgender Befehl triggert den Pause-Button in der Anwendung um den Abspielvorgang fortzusetzen:

```
http://localhost:12400/continue
```

## Shortcut-Sequenz beenden und zurück zur ursprünglichen Playlist

Folgender Befehl beendet das Abspielen einer Sequenz, die über einen Short-Cut aufgerufen wurde und kehrt zurück zur ursprünglichen Position in der Playlist. Nach dem Rücksprung wird die ursprüngliche Sequenz neu gestartet und von Anfang an gespielt.

```
http://localhost:12400/resume
```

Für den Fall das es sich bei der Sequenz um einen Overlay handelt, so können diese über folgenden Befehl zurückgesetzt werden

```
http://localhost:12400/overlay/reset
```

## Zur nächsten Sequenz springen

Folgender Befehl triggert den Next-Button in der Anwendung um die nächste verfügbare Sequenz in der Liste abzuspielen:

```
http://localhost:12400/next
```

## Zur vorherigen Sequenz springen

Folgender Befehl triggert den Back-Button in der Anwendung um die vorherige verfügbare Sequenz in der Liste abzuspielen:

```
http://localhost:12400/prev
```

## Ausgabe auf Schwarz blenden (Fade-To-Black)

Folgender Befehl triggert den FTB-Button in der Anwendung um Ausgabe auf Schwarz auszublenden:

```
http://localhost:12400/ftb
```

## Gesamte Helligkeit und Helligkeitszonen regeln

Folgender Befehl triggert den Schieberegler um die globale Helligkeit zu steuern:

```
http://localhost:12400/brightness/???
```

Anstelle der **???** muss hier ein Wert zwischen 0 und 100 angegeben werden. Der Wert repräsentiert den prozentualen Transparenzwert eines Schwarzen Layers, der On-Top der Ausgabe gezeichnet wird.

Als Zusatz wird zur Unterstützung der StreamDeck TurnWheels, neben den exakten Werten, auch die Werte "up" and "down" bereitgestellt, die jeweils die Helligkeit um +/- 5% ändern

```
http://localhost:12400/brightness/up  
http://localhost:12400/brightness/down
```

Um Helligkeitszonen anzusprechen ist der entsprechende Wert der Zone hinzuzufügen, z.B. "/brightness1/" bis "/brightness8/". Die Variablen sind analog zur gesamten Helligkeit anzuwenden.

## Sequenzen sperren / entsperren

Um Sequenzen zu sperren oder freizugeben bitte folgenden Befehl verwenden

```
http://localhost:12400/sequence/lock/10
```

10 ist dabei die Sequenz und der Befehl würde für das aktuelle Preset angewendet werden. Wenn der Befehl in einem bestimmten Preset ausgeführt werden soll, dann den gewünschten Preset mit : anhängen, also z.B.:

```
http://localhost:12400/sequence/lock/10:3
```

Dieser Befehl sperrt die 10te Sequenz im 3ten Preset. Analog dazu lautet der Befehl zum entsperren ".../unlock/..."

```
http://localhost:12400/sequence/unlock/10
```

```
http://localhost:12400/sequence/unlock/10:3
```

## Presets der Sequenzen auswählen

Zur Auswahl der Sequenzen die abgespielt werden sollen, besteht die Möglichkeit direkt eine Auswahl am rechten Bildschirmrand von LEDVIZ zu treffen. Diese Auswahl wird in einem Preset zusammengefaßt und gesteuert.

Über die API können die vorhandenen Presets 1 - 9 direkt ausgewählt werden.

```
http://localhost:12400/preset/<preset>
```

Der Tag <preset> kann hierbei eine Nummer von 1 bis 9 enthalten. Unabhängig davon ob die Presets umbenannt wurden verbleibt die Angabe des <preset> immer bei 1 - 9. Eine Auswahl über den Namen wird nicht unterstützt.

## LiveText ansteuern

Um einen Tag Wert für die LiveText Darstellung an LEDVIZ zu übergeben, verwenden sie folgenden Befehl

```
http://localhost:12400/tags/<tag-name>?<tag-value>
```

## CuePoints setzen und aufrufen

In einer PlayList können über die REST-API Sprungmarken gesetzt und auch wieder aufgerufen werden. Das ermöglicht die gezielte Ansteuerung einzelner Sequenzen z.B. als Rücksprung aus Sequenzen mit ShortCuts

CuePoints werden mit der Option /set/ gesetzt. Der Name kann dabei ein beliebiger Alphanumerischer String sein.

```
http://localhost:12400/cue/set/<cue-point-name>
```

Um zu einem gespeicherten CuePoint zurück zu kehren wird der CuePoint mit seinem Namen direkt aufgerufen

```
http://localhost:12400/cue/<cue-point-name>
```

Eine zusätzliche Angabe von XML Dateien als Parameter ist bei CuePoints nicht nötig. Der CuePoint enthält bereits den Verweis auf die entsprechende XML Datei.

CuePoints werden nicht gespeichert und stehen nur während der Laufzeit zur Verfügung in der sie auch erstellt wurden. Nach einem Neustart ist die Liste der CuePoints leer. Sollen feste Sprungmarken vergeben werden, so kann das über "midi" oder "dmx" Kennungen realisiert werden.

## Steuerungen aktivieren / deaktivieren

Mit den Steuerungsmodulen können neben der der RestAPI auch andere verschiedene Protokolle für die Steuerung von LEDVIZ genutzt werden. Steuerungen können über die API wie folgt aktiviert oder deaktiviert werden

```
http://localhost:12400/control/hippo/true  
http://localhost:12400/control/hippo/false
```

```
http://localhost:12400/control/dmx/true  
http://localhost:12400/control/dmx/false
```

```
http://localhost:12400/control/midi/true  
http://localhost:12400/control/midi/false
```

## Sonderfunktionen ausführen

Über die Sonderfunktionen einer Sequenz kann ein bestimmtes Verhalten beim Abspielen von Animationen durchgeführt werden. Eine dieser Funktion ist z.B. stop / continue beim Abspielen von Videos.

Sollte eine Sequenz an einer Stop-Marke stehen kann sie über folgenden REST-API Befehl fortgesetzt werden

```
http://localhost:12400/gfx/continue
```

## Sprungmarken aufrufen

Über die Sonderfunktionen gesetzten Sprungmarken könne auch gezielt aufgerufen werden. Das gilt sowohl für Sprungmarken vor und hinter dem aktuellen Abspielpunkt. Der Aufruf erfolgt über folgenden Befehl:

```
http://localhost:12400/gfx/jump/<stop>
```

Der Wert von Stop kann hier die Werte 1 - 4 haben.

## Ändern von Content

Geladene Videos können über die REST-API von extern über folgenden Link geändert werden:

```
http://localhost:12400/file/dmx/1/<filename>  
http://localhost:12400/file/midi/ABC/<filename>
```

# DMX512 ArtNET

LEDVIZ besitzt ebenfalls eine Unterstützung für das DMX512 ArtNET Protokoll, das vor allem im Zusammenspiel mit Lichttechnik verwendet wird. Hier kann LEDVIZ z.B. über eine ARKAOS System remote und synchron zu anderen Elementen gesteuert werden.

Aufgrund der extremen hohen Frequenz mit der DMX512 Befehle gesendet werden, eignet sich das DMX512 Protokoll nicht zur Übertragung über ein reguläres Ethernet. Es sollte immer ein separates Netzwerk aufgebaut werden, z.B. über einen zusätzlichen USB - ETHERNET Adapter, damit der Zugriff auf das Netzwerk als solches nicht gestört wird.

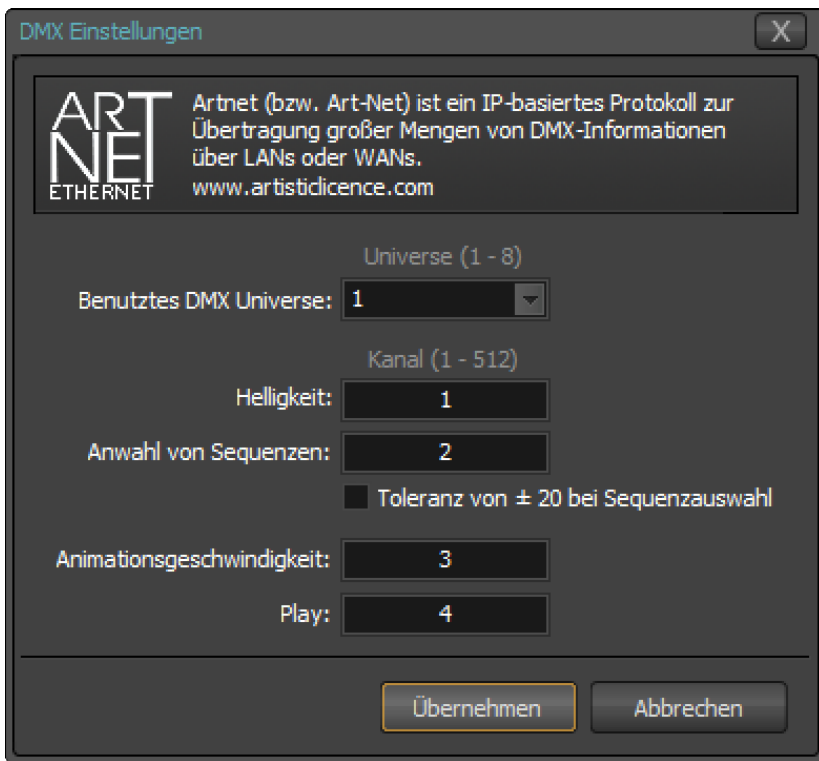
## Anwendungsbereich

Über das DMX Protokoll werden in der Praxis vor allem Lichteinheiten im showprogramm gesteuert. Über die gesendeten Werte können Helligkeit, Farbwerte und X/Y/Z for Spots gesteuert werden.

Im Zusammenspiel mit der LED Band können dann gezielt Abläufe eines Showprogrammes gesteuert werden.

## Einstellungen

Bevor die DMX Steuerung genutzt werden kann müssen einige Parameter eingestellt werden. Den Dialog für die Einstellungen erreichen Sie über die Menüleiste oben unter dem Punkt Steuerung.



Grundsätzlich besteht das Protokoll aus einer reinen Abfolge von Werten. Jede Liste wird durch eine ID, dem Universe gekennzeichnet. Dem Universe folgen 512 Werte, die jeweils von 0 bis 255 gehen können. Die genutzten Kanäle für LEDVIZ sind immer aufeinanderfolgend. Es muss deshalb nur der erste Kanal angegeben werden. Dabei werden folgende Werte unterstützt:

<b>Kanal + 0</b>	Helligkeit
<b>Kanal + 1</b>	Sequenz ID
<b>Kanal + 2</b>	Geschwindigkeit der Animation
<b>Kanal + 3</b>	Trigger für den Play Button (Wert 255 = Play)

Hinweis: Der Play-Befehl wird nur getriggert, wenn zuvor ein Wert empfangen wurde, der kleiner 255 ist!

Die Überwachung von eingehenden DMX Befehlen ist für den Zeitraum aktiv, für die Abfrage aktiv ist. Über das Menü "Steuerung" kann diese Überwachung aktiviert werden. Ein aktive DMX Überwachung wird durch einen blauen Balken oberhalb der Playliste deutlich gekennzeichnet.

**aktiv - DMX512 - Automatik aktiv - DMX512 - Auto**

Zusätzlich werden oben rechts durch das aufblinken des DMX512 Symbols eingehenden Befehle angezeigt. Bleibt das Symbol grau, bedeutet die das LEDVIZ keine DMX Befehle erkennt.

Sollte es Probleme mit dem Empfang vom DMX Befehlen geben, bitte überprüfen Sie ob der UDP Port **6454** in der Firewall geöffnet und die Daten auf dem PC empfangen werden

können.

Schnittstellen

# MIDI

Neben dem DMX und Rest-API unterstützt LEDVIZ auch das MIDI Protokoll. Über diese Protokoll können sehr einfach externe Bedieneinheiten angeschlossen und somit in LEDVIZ z.B. die Helligkeit über Schieberegler oder Sequenzen direkt auf Knopfdruck angesteuert werden.

# LiveText

Mit der LiveText Schnittstelle können dynamische Elemente wie Spielstände, Zeiten oder Wettquoten in Videos eingebettet und live dargestellt werden.

# Allgemeine Informationen

Mit dieser Funktion haben Sie die Möglichkeit einen oder mehrere Texte von außen auf der Bande live darzustellen und zu ändern.

Beispiele aus der Praxis sind unter anderem:

- Spielstände
- Uhrzeiten
- Wettquoten
- statistische Spielauswertungen
- dynamischer Content wie z.B. Twitter

Die Darstellung wird über einen zusätzlichen Layer über der jeweiligen Werbung realisiert. Das funktioniert sowohl über statischen Grafiken, also auch über Animationen.

LEDVIZ selbst führt hier intern eine Liste an Tags und zugehörigen Texten. Immer wenn der Text geändert wird, wird dieser an die Ausgabe weitergegeben. Die Ausgabe erzeugt eine transparente Textur und legt diese als Layer über die jeweiligen Grafik.

Die Liste an Tags wird global geführt und ist immer auf eine Grafik oder Animation bezogen. Egal wo in der Ausgabe diese Grafik eingesetzt wird, wird der Layer erstellt und dargestellt. Für eine Grafik oder Animation können gleichzeitig mehrere Tags erzeugt werden. Die Menge der Tags für die LiveText Layer ist nur durch die Performance des jeweiligen Rechners begrenzt. In der Praxis wurden hier bereits Sequenzen mit mehr als 25 LiveText Ausgaben in der Fussball-Bundesliga, gesteuert durch einen Twitter/Instagram-Crawler, realisiert.

Dadurch das sich die Darstellung auf Text beschränkt, wird die Darstellung von Bildern oder Emojis zur Zeit noch nicht unterstützt.

# Einrichtung eines Tags

Aktuell gibt es noch keine Einrichtungsmöglichkeit über LEDVIZ selbst, so dass die Tags für den LiveText Layer zur Zeit noch manuell im XML erfasst werden müssen.

Der Definitionsblock muss sich durch <TAGS> geklammert auf der obersten Ebene, genauso wie z.B. die Helligkeitszonen oder Segmente befinden. Hier ein Beispiel:

```
<TAGS>
  <TAG>
    <TAG-FILE>BNN_LIVE_Premium_15min.avi</TAG-FILE>
    <TAG-NAME>#TAG1</TAG-NAME>
    <TAG-VALUE>2: 0</TAG-VALUE>
    <TAG-DEFAULT>WILLKOMMEN</TAG-DEFAULT>
    <TAG-FONT>Tahoma</TAG-FONT>
    <TAG-SIZE>50</TAG-SIZE>
    <TAG-COLOR-RED>255</TAG-COLOR-RED>
    <TAG-COLOR-GREEN>255</TAG-COLOR-GREEN>
    <TAG-COLOR-BLUE>255</TAG-COLOR-BLUE>
    <TAG-BOLD>1</TAG-BOLD>
    <TAG-ITALIC>0</TAG-ITALIC>
    <TAG-CAPITALIZE>0</TAG-CAPITALIZE>
    <TAG-COUNT>4</TAG-COUNT>
    <TAG-Y>10</TAG-Y>
    <TAG-X>1024</TAG-X>
    <TAG-X2>2048</TAG-X2>
    <TAG-X3>4096</TAG-X3>
    <TAG-X4>8192</TAG-X4>
    <TAG-LINE-WIDTH>2550</TAG-LINE-WIDTH>
    <TAG-TIME-IN>2</TAG-TIME-IN>
    <TAG-TIME-OUT>20</TAG-TIME-OUT>
    <TAG-ALIGN>center</TAG-ALIGN>
  </TAG>
  ... (evtl. weitere Tags) ...
</TAGS>
```

Die einzelnen Untertags sind relativ selbsterklärend, werden aber zur Sicherheit hier nochmal aufgeführt:

<b>&lt;TAG-FILE&gt;</b>	(string) z.B. "werbung.avi"	Der Name der Datei für die der Text-Layer erstellt werden soll
<b>&lt;TAG-NAME&gt;</b>	(string) z.B. "TAG1"	Der Name des Tags mit dem der Tag von außen her angesprochen wird. Dieser Name muss eindeutig sein und kann nur für diesen Tag verwendet werden.
<b>&lt;TAG-VALUE&gt;</b>	(string) z.B. "2:0"	Dieser Wert enthält den aktuellen gespeicherten Wert und muss nicht angegeben werden, es sei denn Sie möchten den LiveText Wert vorbelegen
<b>&lt;TAG-DEFAULT&gt;</b>	(string) z.B. "WILLKOMMEN"	Der Wert dieses Tags wird immer dann verwendet, wenn ein leerer Wert mitgeteilt wurde, also im Prinzip der Wert <TAG-VALUE> nicht belegt ist
<b>&lt;TAG-FONT&gt;</b>	(string) z.B. "Tahoma"	Der Windowsname der zu verwendenden Schriftart. Es muss hierbei der Name der Schriftart verwendet werden. Den genauen Namen können Sie den Installierten Schriftarten der Windowsinstallation entnehmen. Der Dateiname der Schriftart spielt hier keine Rolle, das Tag bezieht sich auf den Namen unter dem die Schriftart in Windows registriert ist
<b>&lt;TAG-SIZE&gt;</b>	(integer) z.B. 50	Der Wert enthält die zu verwendende Schriftgröße. Zur besseren Lesbarkeit sollten hier Werte von >50 verwendet werden
<b>&lt;TAG-COLOR-RED&gt;</b>	(integer) 0 bis 255	Der rote Farbwert für den zu verwendenden Text.
<b>&lt;TAG-COLOR-GREEN&gt;</b>	(integer) 0 bis 255	Der grüne Farbanteil
<b>&lt;TAG-COLOR-BLUE&gt;</b>	(integer) 0 bis 255	Der blaue Farbanteil
<b>&lt;TAG-BOLD&gt;</b>	(bool) z.B. 0 oder 1	Bestimmt ob der Text fett oder normal gezeichnet wird.
<b>&lt;TAG-ITALIC&gt;</b>	(bool) z.B. 0 oder 1	Bestimmt ob der Text kursiv gezeichnet wird.

<b>&lt;TAG-CAPITALIZE&gt;</b>	(bool) z.B. 0 oder 1	Bestimmt ob der gesamt Text in Großbuchstaben umgewandelt werden soll.
<b>&lt;TAG-COUNT&gt;</b>	(integer) 0 bis 4	gibt die Anzahl der zu berücksichtigen <TAG-X?> Werte an.
<b>&lt;TAG-X&gt;</b> <b>&lt;TAG-X2&gt;</b> <b>&lt;TAG-X3&gt;</b> <b>&lt;TAG-X4&gt;</b>	(integer)	Dieser Wert bestimmt die horizontale Position in Pixeln innerhalb der Grafik. Der Wert 0 befindetets sich hierbei links.
<b>&lt;TAG-Y&gt;</b>	(integer)	Dieser Wert bestimmt die vertikale Position in Pixeln innerhalb der Grafik. Der Wert 0 befindetets sich hierbei oben. Bitte beachten Sie hierbei das die Schriftarten in der Regel etwas Freiraum oberhalb der Buchstaben besitzen und der Text nicht direkt pixelgenau an der Position beginnt.
<b>&lt;TAG-LINE-WIDTH&gt;</b>	(integer)	Dieser Wert gibt die maximale Breite des Textes an. Ist dieser Wert größer 0, dann wird der Text bei Erreichen des Wertes automatisch umgebrochen. Die Schrifgröße wird dann entsprechend angepaßt und verkleinert.
<b>&lt;TAG-TIME-IN&gt;</b>	(decimal) z.B. 15.4	Dieser Wert bestimmt den Zeitpunkt, wann der Text während des Abspielens erscheinen soll. Der Wert repräsentiert die Zeit in Sekunden, wobei hier Dezimale Werte mit einem Punkt als Dezimalzeichen zulässig sind.
<b>&lt;TAG-TIME-OUT&gt;</b>	(decimal) z.B. 15.4	Dieser Wert bestimmt den Zeitpunkt, wann der Text während des Abspielens ausgeblendet werden soll.
<b>&lt;TAG-ALIGN&gt;</b>	(string) z.B. left, right, center	Der Text kann an den durch <TAG-X> Koordinaten vorgegebenen Wert ausgerichtet werden. Zulässige Wert sind left, right und center
<b>&lt;TAG-TYPE&gt;</b>	(string) z.B. text, progress, image	Der Standardwert ist Text, allerdings kann hier auch definiert werden, das es sich um ein Bild handelt. Der Wert in <TAG-VALUE> und <TAG-DEFAULT> wird dann als Verweis auf ein Bild interpretiert.
<b>&lt;TAG-WIDTH&gt;</b> <b>&lt;TAG-HEIGHT&gt;</b>	(integer)	Größer der Ausgabe des referenzierten Bildes im Fall des <TAG-TYPE> "image"

<b>&lt;TAG-MIDI&gt;</b>	<code>(string),(string)</code>	Sollte eine Sequenz über einen Midi-Key für einen Aufruf verfügen, kann dieser Tags angewiesen werden, zusätzlich zum Dateinamen nur bei dieser Midi-Sequenzen angezeigt zu werden. Mehrere Midi-Keys können getrennt durch Komma eingegeben werden
-------------------------	--------------------------------	---

Bitte beachten das der Renderer Zeichen, wie Return oder doppelte Leerzeichen vor der Darstellung entfernt. Um die Grafik zu positionieren empfiehlt es sich zuerst mit einem Wert von 0 für <TAG-TIME-IN> und einem hohen Wert, z.B. 999 für <TAG-TIME-OUT> zu beginnen und erst später die Abstimmung der Ein und Ausblendzeiten vorzunehmen.

# Ansteuerung des Textes

## Ansteuerung über UDP Schnittstelle

LEDVIZ hält hierfür permanent einen UDP Port offen. Alle Anfragen der Art beginnen mit einem Hashtag gefolgt von dem Tag-Namen. Die Trennung des Textes erfolgt durch eine Pipe |. Ein Beispiel eines Textes könnte also wie folgt aussehen:

```
#TAG1| Hello world! |
```

Wird ein Tag-Element für einen LiveText empfangen wird dieses direkt an die Ausgabe weitergegeben. Das hat zur Folge das der Layer auch sofort aktualisiert wird. Der Text kann also während seiner Laufzeit jederzeit und mehrfach, während er ausgespielt wird, aktualisiert werden.

Die UDP Daten werden auf Port **28077** im UTF8 Format erwartet.

Aktuell wird noch an einer Eingabemaske unter Nutzung der bestehenden REST API gearbeitet. Neben dem Call über die RestAPI ist geplant eine Webseite mit verfügbaren Tags und Eingabe direkt als HTML Seite zur Verfügung zu stellen.

## Ansteuerung über den LEDVIZ

### Operational HUB

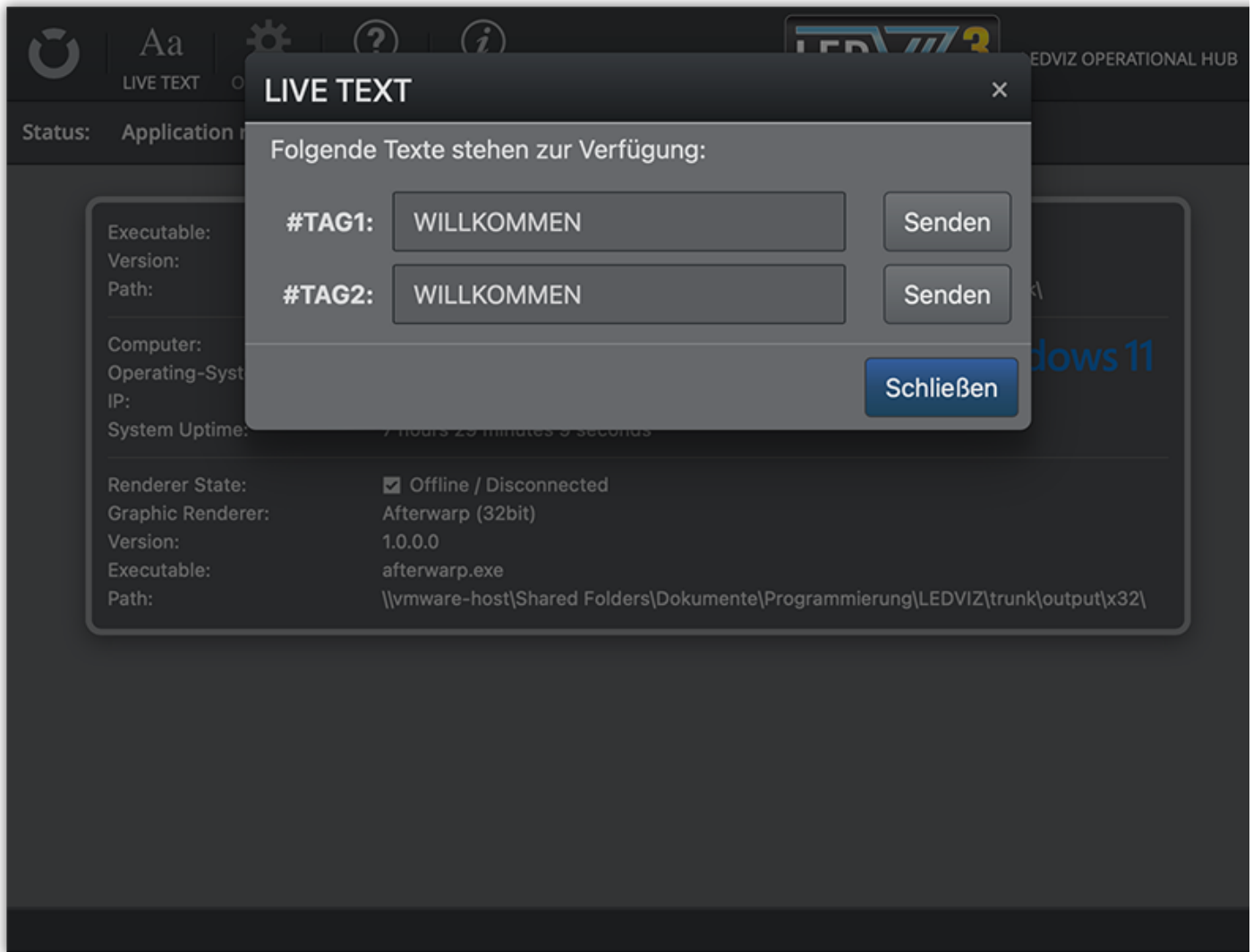
Der LEDVIZ Operational Hub ist eine Webseite die Funktionen von LEDVIZ als Webseite bereitstellt. Zugriff auf den Hub erfolgt über den Aufruf <http://localhost:19399>. Anstelle von localhost kann auch der spezifische Rechnername oder die IP angegeben werden. Somit ist eine Remote Eingabe der Texte über dritte Systeme jederzeit möglich.

Nach dem Aufruf des LEDVIZ Operational Hub im Browser sollte folgende Seite angezeigt werden:

The screenshot displays the LEDVIZ 3 Operational Hub interface. At the top, there is a navigation bar with icons for 'LIVE TEXT', 'OPTIONS', 'HELP', and 'ABOUT', along with the 'LEDVIZ 3' logo and the text 'LEDVIZ OPERATIONAL HUB'. Below the navigation bar, the status is shown as 'Application running and connected ...'. The main content area features a rounded rectangle containing system and application information:

Executable:	LEDVIZ.exe
Version:	3.2.0.0
Path:	\\vmware-host\Shared Folders\Dokumente\Programmierung\LEDVIZ\trunk\
Computer:	DESKTOP-1QPV5DT
Operating-System:	Windows 11
IP:	192.168.67.142
System Uptime:	7 hours 7 minutes 19 seconds
Renderer State:	<input checked="" type="checkbox"/> Offline / Disconnected
Graphic Renderer:	Afterwarp (32bit)
Version:	1.0.0.0
Executable:	afterwarp.exe
Path:	\\vmware-host\Shared Folders\Dokumente\Programmierung\LEDVIZ\trunk\output\x32\

Hier kann über den Menüpunkt oben Links ein dialog für die Eingabe von Texten geöffnet werden. Es werden nur die Tags gelistet, die auch im aktuellen Projekt enthalten sind. Eingaben können hier direkt vorgenommen und gesendet werden.



## Sonstige Tools

Zum Testen der Schnittstelle kann ein kleines Tool verwendet werden, das ursprünglich für die Bereitstellung von Wettquoten für ein Trainingsspiel des 1.FC Köln gegen den Watford FC erstellt wurde: <https://www.ledviz.de/download/LEDVIZlive.zip>. Das Tool verwendet die Tags:

`#QUOTE- WIN- FCK` | `#QUOTE- WIN- WTF` | `#QUOTE- TOR- FCK` | `#QUOTE- TOR- WTF`

# KINEXON Connector

Die Daten für die LiveText Schnittstelle können aus verschiedenen Quellen kommen. Unter anderem wurde bereits eine Schnittstelle zu Kinexon realisiert. Kinexon sammelt über Tracker verschiedenste statistische Daten z.B. in der Handball-Bundesliga.

Der Connector dient hier als Vermittler zwischen Kinexon und LEDVIZ. Die Daten werden in Realtime von der Kinexon Schnittstelle abgefragt und als Tags im LiveText Format an LEDVIZ weitergegeben. Hierbei werden die Daten aufbereitet und in Textform umformatiert. Ausgangspunkt sind die abgerufenen Daten im JSON Format der Kinexon REST-API.

