

Se[real] Lighting Killer マニュアル

これは照明制御システム「Se[real] Lighting Killer」のマニュアルです。

このシステムはプリセットを保存して呼び出す従来の方法に加え、照明制御信号をリアルタイムに生成・編集し入出力することが可能なシステムです。照明制御信号の生成には内包する“VAE for Lighting”というシステムなどを使用します。

また、照明信号の入出力を拡張したことにより、他の演出との同期をより柔軟に行うことができるようになりました。

パフォーマンスの例は[YouTube](#)にありますので、ぜひご覧ください。

マニュアルを用いてシステムを利用する上で必要なファイル等は[こちら](#)にまとまっています。(テキスト内でも都度紹介します。)

目次

- 概要
 - Se[real] Lighting Killerとは
- 初期設定(PC設定含む)
 - IPアドレスの設定
 - モニタリングの方法
- 基本的な操作
 - 用意した灯体の種類
 - 3つの操作画面について
 - 色
 - ムービング
 - パラメータの種類
- プリセットの設定
- “VAE for Lighting”の使い方
 - データセットの作成
 - 学習・モデルの保存
 - パターンの生成と編集
- 外部入出力の方法
 - OSCでの入出力
 - midiコントローラの接続と入力
- その他
 - 照明機材の接続方法
 - Maxを使ったカスタム設定

概要

Se[real] Lighting Killerとは

「Se[real] Lighting Killer」(以下、SLK)は、照明制御信号をリアルタイムに生成・編集し入出力することが可能なシステムです。

本システムは、従来のプリセットを呼び出す方法に加え、リアルタイムに照明制御のパターンを生成・編集し出力することが可能です。

制作したシステムは大きく3つに分かれます。

- 照明信号を入出力するメインシステム
- VAE用のデータセットを作成するシステム
- データセットで学習する用のシステム([VAE Rhythm Generator for Ableton Live](#)を参考に制作しています。)

システムの入出力を拡張することにより、一般的な照明制御信号のDMXやArt-Netのみでなく、音や映像などの他の演出との連携が容易に行える信号を出力することが可能です。

動作確認環境

macbook pro (15-inch, 2016)

macOS Mojave(10.14.6), Catalina (10.15.7)など

OMEN (windows 10)

PCのIPアドレス：169.254.206.150

インターフェース：Open Dmx Ethernet

IPアドレス：169.254.206.200

今回のマニュアルはmacで操作することを想定して制作しています。

(windowsでも使用可能です。適宜読み替えてください。)

また、照明信号のインターフェースにはENTTECのOpen DMX Ethernet(以下ODE)を使用しています。

マニュアル内にある画像にはすべてリンクがついていますので、詳しく知りたい場合は参照してください。

初期設定(PC設定含む)

IPアドレスの設定

照明の信号を照明機材(今回はODE)にネットワーク(LAN)経由で信号を送る場合、照明機材と制御信号出力PCを同じネットワークに設定する必要があります。(シミュレーションのみで使用する場合は、有線LANに接続する必要はありません。使用しているPCのIPアドレス(あるいは127.0.0.1)を取得して、シミュレーションソフトの設定に使用します。)



これは、IPアドレスを正しく設定することで可能です。

例えばODEが”169.254.206.200”だった場合、PCはサブネットを”255.255.0.0”、ipを”169.254.206.150”などに設定します。

(今回は、この設定にします。)

PCとODEが同じネットワークに接続されている場合は、これに沿っていなくても問題ありません。

以降の説明は、適宜読み替えてください。

以下、設定方法についてです。

まず、照明機材とPCを有線LANで接続します。

PCのipアドレスの設定は”システム環境設定/ネットワーク”から、接続しているLANを選択し、

”IPv4の設定” - 手入力

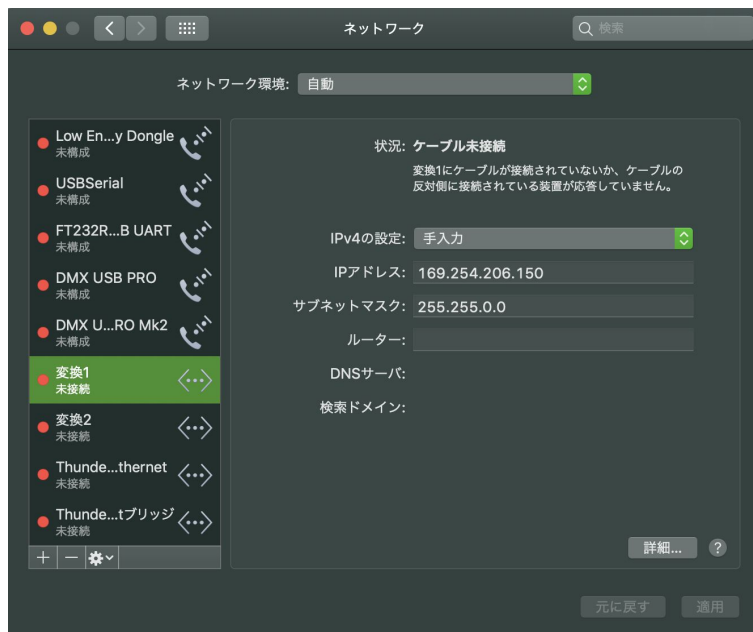
IPアドレス : 169.254.206.150

サブネットマスク : 255.255.0.0

に設定します。

(画像内左の”変換1”の部分が赤いですが、接続が成功している場合は緑あるいは黄色になります。)

IPv4の設定の設定は、”DHCPサーバの使用(アドレスは手入力)”でも基本的には問題ありません。



ODE側は、専用のアプリケーションからIPを設定する必要があります。
詳しくは[こちらの記事](#)をご覧ください。
これで、ODEのIPを”169.254.206.200”に設定します。

モニタリングの方法

モニタリングをするには一般的な照明シミュレーションソフトが必要になります。
こちらは任意のもので構いませんが、今回は”Q Light Controller Plus”(QLC+)と”PrefLight”というソフトウェアでモニタリングする方法を紹介します。
“Q Light Controller Plus”は照明信号を”Se[real] Lighting Killer”から”PrefLight”に送るために使用します。
“PrefLight”は入力した照明信号によって照明器具の動作を模擬表示します。
どちらも無料の範囲内で十分に使用可能です。

Q Light Controller Plus

Q Light Controller Plus(QLC+)は、照明制御信号を入出力することができるフリーのPC用ソフトウェアです。

今回は、照明制御信号をPC内でルーティングするために使用します。

(ルーティングとは、照明信号を出力するソフトと、信号を受け取るソフトを繋げることを言います。実際、今回はSLKを立ち上げるときのみQLC+を起動すれば十分です。

一度全てのソフトを接続できればQLC+は必要ありません。)

インストール

Q Light Controller Plus(以下QLC+)は[こちらのページ](#)からmacOS版をダウンロードします。
ダウンロードしたdmgファイルを実行し、インストールします。

設定

(マニュアルは[こちら](#))

設定を完了したデータを[こちら](#)に用意しています。
このファイルをQLC+で開けば、以下の設定は飛ばして構いません。

最初に起動した際に”Fixture Definition Editor”か”Q Light Controller Plus”が聞かれます。
今回は”Q Light Controller Plus”の方を選択します。

アプリ画面が開くと、下方にある”入力/出力設定”タブのみ設定します。(今回は、他の機能は使いません。)

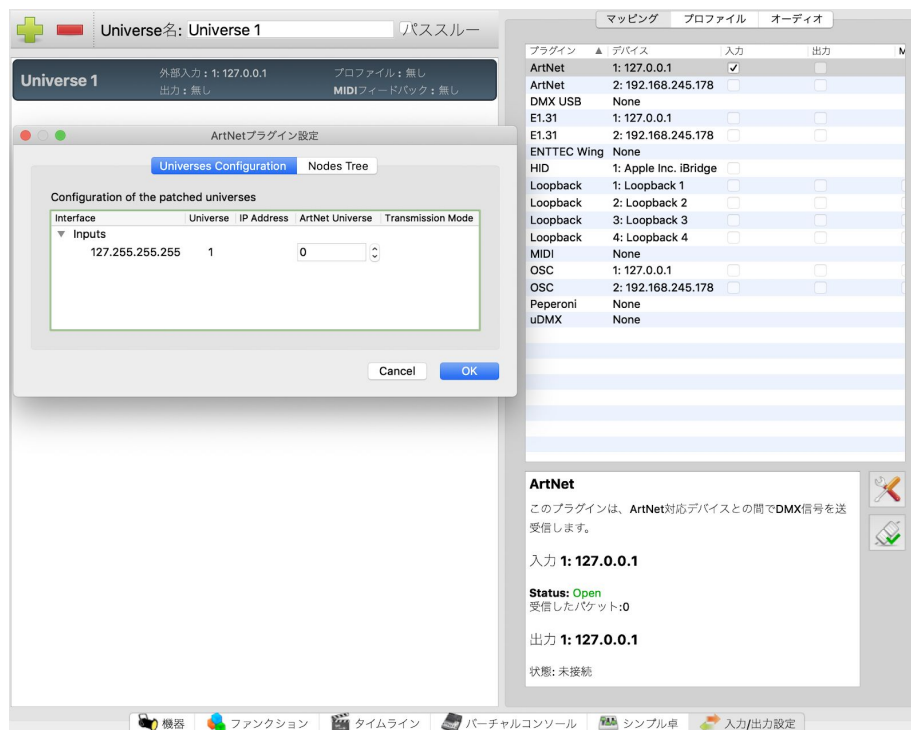
Universe2~4を左上の赤マイナスで削除し、Universe1の設定を変更して使用します。

Universe1を選択し、右側欄の”入力(input) - ArtNet 1: 127.0.0.1”にチェックマークをつけます。

他のチェックマークは外します。

また、”Art-Net1”をダブルクリックすると、ポップアップメニューが出てきます。

このメニューは”ArtNetプラグイン設定”です。この中の”ArtNet Universe”が”0”になっていることを確認します。



QLC+の設定は以上です。

QLC+を閉じるときは、ファイルを保存すると次回から設定が楽です。

PrefLight

PrefLightは有料ソフトですが、無料でも一部制限付きで使用可能です。

詳細はダウンロードページから確認できます。

インストール

PrefLightは[こちらのページ](#)からmacOS版をダウンロードします。
ダウンロードしたpkgファイルを実行し、インストールします。

設定

無料版では、立ち上げてしばらく待つと”続ける”というボタンが押せるようになります。
(2回ほど押してください。)
また、このアプリを起動する際は、”Se[real] Lighting Killer”とQLC+を落としてから、起動してください。

ウィンドウタブの”DMX 接続...”を開きます。
出てきたウィンドウで、1を”ArtNet Sub Net : 0 Universe : 0”と設定します。
すると、その行の右端に”動作中”と表示されます。(されない場合は、QLC+や”Se[real] Lighting Killer”のメインシステムなどを閉じて、選択肢を”ArtNet”から”なし”に一度戻してから、再度設定し直してください。)

その後、照明の灯体等の設定を行います。
こちらの[ファイルをダウンロード](#)してください。
(ファイルをダウンロードしない場合は、[こちらの画像](#)を参考にして設定してください。)
こちらの設定は、もちろん実際に使用する照明の設定に合わせてください。
今回は、このファイルの設定を前提として、説明します。

灯体や物体の配置などの詳しいやり方は、[PrefLightのマニュアル](#)をご覧ください。

基本的な操作

ここからは、照明制御システムの”Se[real] Lighting Killer”の使い方の説明をします。

システムの開き方

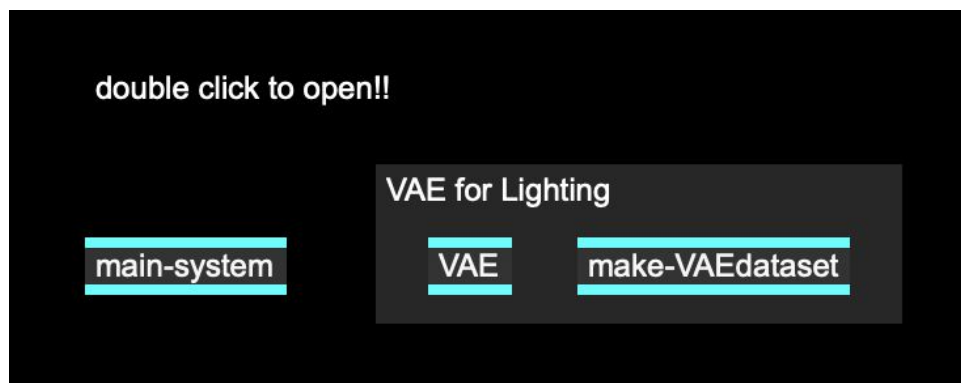
zipファイルを展開して、アプリケーションを開いてください。

自動的に小さなシステムの画面が開きます。

開かない場合は”File/Open...”から、一緒に添付した”wSL.mxf”を開いてください。

(セキュリティソフトをご確認ください。)

アプリケーションを立ち上げたら、以下のような画面がでます。



この画面の中の水色で囲まれた3つの四角をダブルクリックすると、それぞれの制御画面が出現します。

用意した灯体の種類

今回は

LED * 8(2種類)

LEDムービング * 2

ムービングスポット * 2

スポットLED * 1

ストロボ * 2(同じチャンネル)

のように照明を用意し仕込むことを想定します。

(詳しい配置等については、シミュレーションソフトを参考にしてください。)

3つの操作画面について

このシステムは大きく3つの画面で構成されています。

1. 照明信号の入出力をするメインのシステム。(main-system)
2. VAEに学習をさせるデータを作成・保存するシステム。(make-VAEdataset)

3. VAEで照明のパターンを学習させ、制御パターンを生成させるシステム。(VAE)

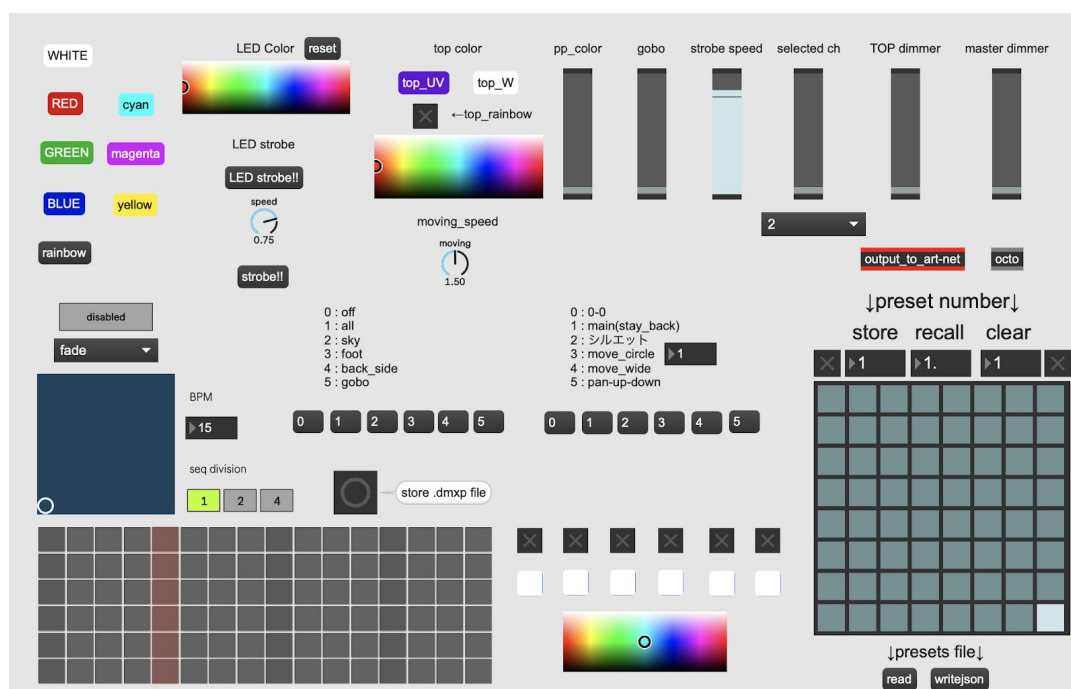
三つ目のシステムは、[VAE Rhythm Generator for Ableton Live](#) (Nao Tokui, 2019)を参考にしています。

また、2,3のシステムを合わせて”VAE for Lighting”と呼ぶことにします。

この”VAE for Lighting”は、次章で詳しく説明します。

メインシステム

このシステムは、照明の色やムービングの動きの制御を行うことができます。



シミュレーションソフトや実際の照明機材に出力する際には、出力先のIPアドレスを指定する必要があります。

中段右側の”output_to_artnet-node”をダブルクリックすると、新しい画面が現れます。



赤枠の”unicast IP”で、照明信号の出力先を設定ができます。

シミュレーションソフトの場合は、自分のPCのIP(システム環境設定で取得したもの、あるいは”127.0.0.1”)を指定します。

照明機材に出力する場合は、ArtNetコンバータのIP(今回は169.254.206.200)を指定します。

また、この画面では

- 信号出力のフレームレートの設定
- blackout
- DMXアドレスに対するマニュアル出力

を行うことができます。

この画面を閉じて問題ありませんが、メインシステムの画面を閉じると、自動で”blackout”が有効になり、全てのDMXアドレスに対して”0”が出力されます。

信号を出力しても反応しない際は、こちらの設定の状態(xがついていないとoff)を確認してください。

これで、照明信号の出力が可能です。

“master dimmer”(右上)のフェーダーを一番上まで上げてみると、照明が点きます。

また、ステージ中央上に6つの灯体が六角形の各頂点上に配置しています。

これは画面右側の”octo”をダブルクリックして、制御できます。

octo master dimmerのフェーダーをあげると、場所がわかると思います。

色

LEDタイプの灯体は、カラーピッカーを使用して、色指定することが可能です。

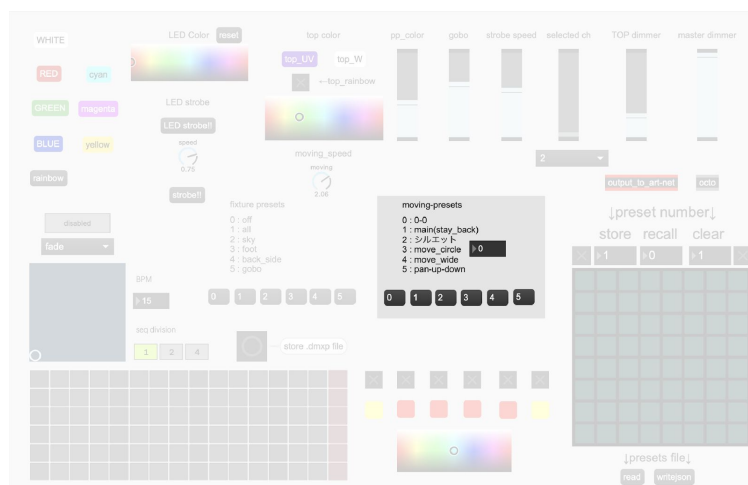
カラーピッカーを使用することで、滑らかにカラーフェードさせながら、指定することができます。

ムービングスポットは、光源の色が固定でカラーフィルターによって光の色を変化させているため、滑らかにカラーフェードすることができません。

ムービング

ムービングの動きは3種類用意しています。

また、固定の位置を狙っているものを2種類用意しています。



0. これは、ムービングライトの初期位置です。
1. すべてのムービングライトが中央の立ち位置を向きます。
2. 1の状態から、スポットライトの向きが正面に移動します。
3. 後ろの壁(ホリゾン)に当たらない程度に動きます。
4. 後ろの壁も無視して動きます。
5. 中央の立ち位置付近を向いて、上下に移動します。

決めの位置を指定した場合は、Maxを使用する必要があります。
この方法はその他の”Maxを使ったカスタム設定”をご覧ください。

パラメータの種類

メインのシステムにはたくさんのパラメータが存在します。
どのようなパラメータが存在し、どのように機能するのか説明します。

- master dimmer

top以外の灯体の光量を調整します。

- TOP dimmer

演者の真上にある灯体(トップ)の光の強さを調整します。

- selected ch

選択した灯体の光の強さを個別に調整します。

- strobe speed

ストロボのパルスの間隔を調整します。

- gobo

ムービングスポットのゴボ(柄)の種類を変更します。

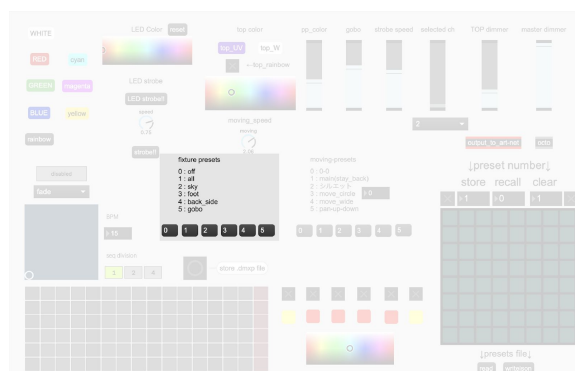
- `pp_color`

ムービングスポットの色を調整します。

- top color

トップの色を調整します。

- 灯体のプリセット



下に並んでいるボタンを押すと、各設定の灯体が点灯します。
例えば、“all”の場合はすべての灯体がonになります。
“sky”は両左右上の灯体、footは足元の灯体です。
実際に点けて確認してください。

- シーケンス



上のボタンを押して“enable”にすると、シーケンスがスタートします。
シーケンスの速度は右側の“BPM”の数字で調整可能です。
また、シーケンスのBPMを変更せずに“seq division”を変更すると、その数字の倍速でシーケンスが進みます。

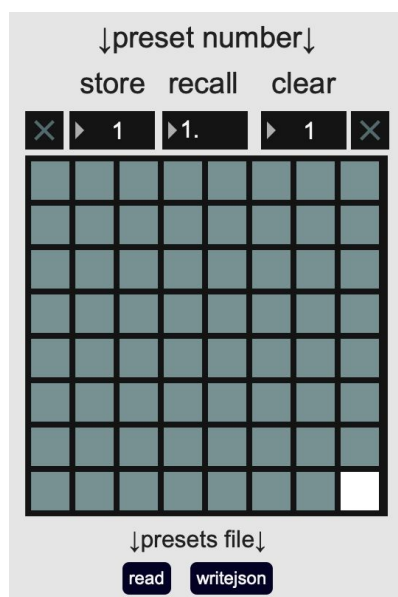
紺色の四角枠内は、VAEでの照明制御パターン生成に使用します。
何も設定していない場合は、何も反応しません。
詳しくは次章で説明します。

プリセットの設定

プリセットを保存し、すぐに呼び出せるようにする方法を紹介します。

やり方は、[こちらの動画](#)を参考にしてください。

プリセットはメインシステム内右側のマトリックスで保存・呼び出しできます。



保存するには、保存したい照明の状態をメインシステムで作り、storeの左側の×をつけて(on)、保存する番号を指定します。

呼び出すには、recallで番号を指定するか、下のマスで該当する番号の場所を押します。

保存したプリセットは、clearの部分に数字を指定すると消すことができます。

こちらも、右側の×をつけて(on)から、番号を指定してください。

作成したプリセットのファイルは、最下部にある”presets file”で保存・呼び出しができます。

保存はjson形式のファイルになります。

presetファイルを更新した場合、システムを落とす際に自動的に保存するか聞かれます。

テストファイルとして[test-presets.json](#)を添付します。

読み込んでマスを押してみてください。

こちらは、縦の列に同じ色を並べていて、上にあるものは激しいシーン、下にあるものは緩やかなシーンとなっています。

“VAE for Lighting”の使い方

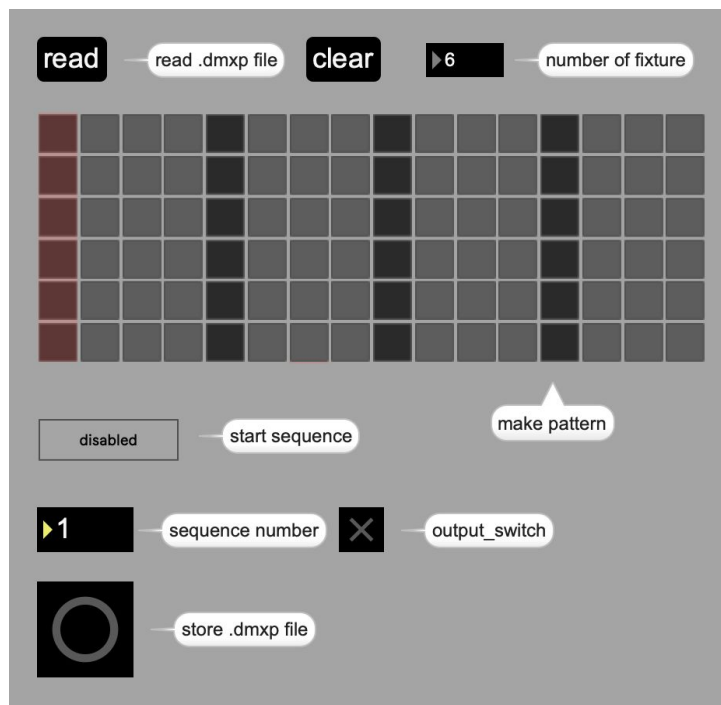
VAE for Lightingは、基本的にmain-systemと切り離して使用します。
裏で動いていますが、基本的には別で使用します。
こちらは、主にセッティングや準備の際に使用します。

また、このシステム内の”学習・保存”については、”[M4L.RhythmVAE: VAE Rhythm Generator for Ableton Live](#)” (Nao Tokui, 2020)を参考に、照明用に改変して制作しています。

VAEの詳細などは、こちらのプログラムを参考にしてください。

データセットの作成

データセットの作成には”make_VAEdataset”を起動します。



このシステムでは、

- データセット用のデータの作成
- 作成したデータの保存
- 作成したデータの呼び出し・編集

が可能です。

以下の説明の映像は[こちら](#)です。

ボタンの説明

- read

保存しているパターンデータ(.dmxp)のファイルを読み込みます。

- clear

マトリックスをすべて消します。

- number of fixture

on-off制御したい灯体の数を選択します。

- start sequence

シーケンスを開始・停止します。

- sequence number

シーケンスの位置を指定します。

- output switch

指定したシーケンスの状態を出力するためのスイッチです。
プレビューしたい場合は、この×をつけて(on)にしてください。

- store .dmxp file

作成したパターンを".dmxp"ファイルとして保存するときに押してください。
ファイル名には".dmxp"という拡張子を各自で指定する必要があります。

データの作成・保存

データを作成するにはマスをタップします。

(copy&pasteはできません。)

このマスは、縦がそれぞれの照明灯体、横がシーケンス(時間変化)を表します。

赤で表示されている列が、現在の状態をあらわします。

マスをタップすると、白く表示されます、この白い状態のときは灯体がfullで点灯している状態です。

作成したら、一番下の"store .dmxp file"をタップします。

ファイル名の最後に".dmxp"という拡張子を必ず指定してください。

(学習させるときに、認識されなくなります。)

データの呼び出し・編集

一度保存したデータは、"read"ボタンを押して呼び出すことができます。

呼び出したデータを編集し、再度保存することが可能です。

編集して保存したい場合は、"store .dmxp file"をタップしてください。

学習・モデルの保存

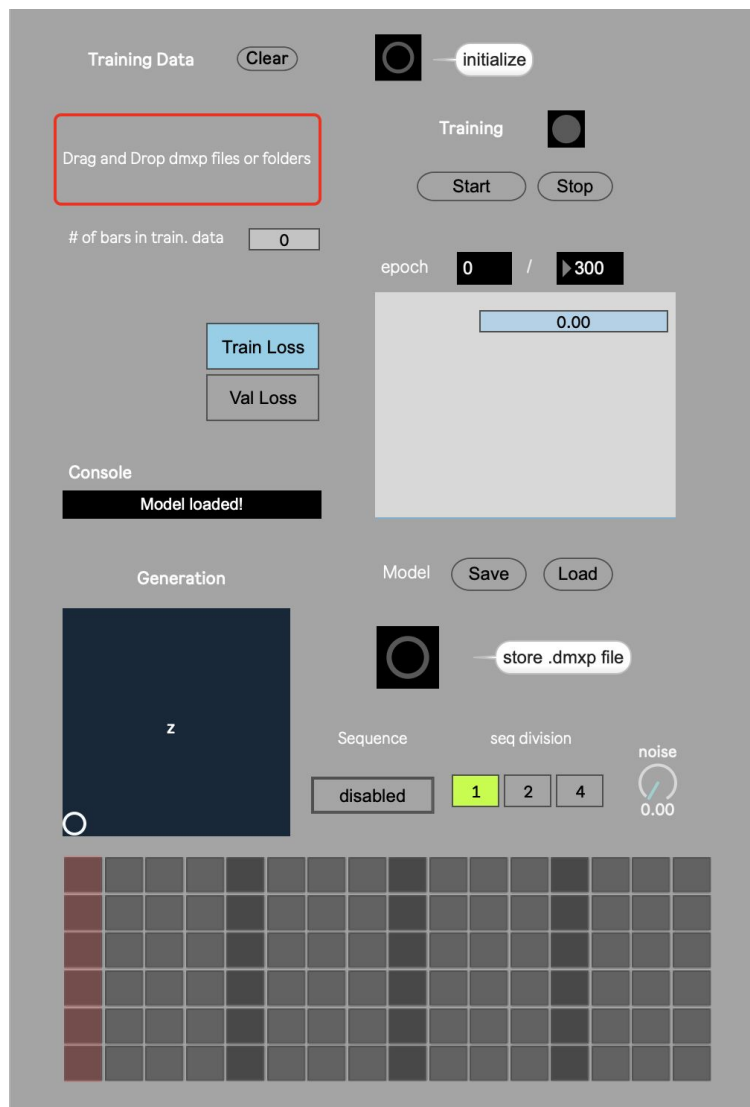
作成したデータセットを使い、VAEのモデルを学習させる方法を説明します。

また、学習後のモデルの保存も紹介します。

保存したモデルのサンプルは[こちら](#)です。

データセットのサンプルは[こちら](#)です。(灯体の数は6台で作成したデータが9個あります。)

作成したデータセットを用いて、学習させるには“VAE”というシステムを使います。
このシステムは、“[M4L.RhythmVAE: VAE Rhythm Generator for Ableton Live](#)” (Nao Tokui, 2020)を参考に制作しています。



このシステムでは

- データの学習(モデルの制作)
 - モデルの保存・呼び出し
 - パターンの生成・編集・保存
- を行うことができます。

このシステムを立ち上げたら、一番最初に最上部の“initialize”ボタンをタップしてください。

また、何かしらうまくいかないときにもこのボタンを押してください。(連絡をください。)

データの学習

まず、学習させたい".dmxp"ファイルを用意します。
このファイルを上部の赤枠内にフォルダごとDrag&Dropし、読み込ませます。
あとは、その右側にある"Start"ボタンを押すだけです。
赤枠右上の"Clear"を押すと、読み込ませたトレーニングデータが消えます。

.dmxpのデータは5~12の範囲だと、うまく学習しやすいです。
epoch数(学習を回す回数)はデフォルトで300回にしています。データが多い場合は回数を増やすと良い結果が得られやすいですが、その分時間がかかります。

モデルの保存・呼び出し

学習が終わった後のモデルは中段の"Save"ボタンを押すと保存できます。
また、一度保存したモデルはその横の"Load"ボタンで呼び出すことができます。

つまり、システムを使用するたびに学習させる必要はありません。

モデルを保存するとjsonファイルとbinファイルの2ファイルが生成されます。
呼び出しの際には、jsonファイルを指定することになりますが、同じフォルダ内にweights.binファイルがなければ、正常に動作しないので、ご注意ください。
(モデルの保存の際に両方のファイルが生成されますので、両方消さないようにしてください。)

パターンの生成・編集

パターンを学習させる、あるいはモデルを呼び出したら、左下の紺色の枠("z map")内がタップできるようになります。
ここをタップすると、その度にモデルがパターンを生成します。
このパターンは、データの作成と同様に、編集が可能です。
また、"store .dmxp file"をタップすると、生成したパターンを保存することができます。

VAE for Lightingの使い方の説明は以上です。

外部入出力の方法

OSCでの入出力

外部アプリやPCから信号の入出力を行って照明制御を行う方法を説明します。

入力について

各信号出力ソフトの設定で、IPアドレスやポートを設定してください。
IPアドレスは、最初に設定したこのPCのIPアドレスで、portは”13581”にします。

このポートに対して以下に列挙するアドレスを付与した形で、信号を指定してください。

- /XX OO

の”/XX”部分がアドレス、”OO”部分が指定する値の種類・範囲です。

- /pp_color 0-55 (int)

[pocket pro](#)という灯体の色情報(カラーマクロ)

0 - 7 white

8 - 14 red

15 - 21 umber

22 - 28 yellow

29 - 35 green

36 - 42 UV

43 - 49 light blue

50 - 55 pink

- /gobo 0-62 (int)

同じく [pocket pro](#) という灯体のゴボ

0-7がゴボ(柄)なし

詳しくは灯体マニュアルを参照

- /strobe_t 0,1 (toggle)

ストロボをオフにするときは0、オンにするときは1にする

- /st_speed 0-1 (float)

ストロボスピード

0 - 0.7の間は、かなり遅い

- /BPM 1-300 (int)

チェイスパターンのスピード

ゆっくり回したいときは、15 - 30くらいが適当

300はほとんどストロボのようになる(この場合はchange_typeをcut(0)にする)

- /sel_ch_id 0-6(int)

それぞれの灯体をひとつずつ調整したい時に、使用する

下ふたつの信号を送る時に、同時に番号を送る

- /selected_ch 0-1 (float)

こちらは選択した灯体のディマーの調整

- /spot_dimmer 0-1 (float)

真ん中の真上にあるスポット照明のディマーの調整

- /master_dimmer 0-1 (float)

全ての灯体のマスターディマー

- /moving_sp 0-1 (float)

ムービングライトの動くスピードを調整する

- /LED_strobe_t 0,1 (toggle)

LEDでストロボを使いたい時に1、オフにするときに0を送る

- /LED_strobe_speed 0-1 (float)

LEDのストロボのスピードを調整する

- /mov_pat_pre 0-5(int)

ムービングの動くパターンを選択する

0 - デフォルト(静止)

1 - 中央(静止)

2 - 客側向き(静止)

3 - 円状に動く

4 - 3より大きく動く

5 - 上下に動く

これ以外に欲しいムービングの位置がある場合はMaxで作成可能です。

- /dim_pat_pre 0-5(int)

どの灯体をオンにするか、のプリセットです。

例えば、足元のムービングだけ付けたい、という時に3を送る、という形です。

0 - 全てオフ

1 - 全てオン

2 - 吊ってある両サイドのLEDのみ

3 - 足元のムービングのみ

4 - 吊ってあるLEDの下手のみ

5 - pocket proのみ

- /seq_t 0,1 (toggle)

チェイスパターンを走らせたいときは1、チェイスを止める時は0を送る

- /change_type 0,1 (toggle)

チェイスを「カットイン/カットオフさせる(0)」か、「フェードイン/フェードアウト(1)」

- /vae_x -300 - 300 (int)

- /vae_y -300 - 300 (int)

チェイスパターンを生成するためのもの

値を変えると、チェイスパターンが変化する

- /top_w 1,0 (toggle)

topのwhiteを100%にします。

dimmerをあげないと、光は出ません。

- /top_rainbow 1,0 (toggle)

topの色がrainbowに移り変わります。

dimmerをあげないと、光は出ません。

- /top_color_hue 0-1 (float)

topの色情報です。

0.0から1.0までを(赤黄緑青紫赤)の順番でフェードする。

目安： 赤 - 0.0 / 1.0

黄 - 0.16 緑 - 0.33

シアン - 0.5 青 - 0.66

紫 - 0.76 ピンク - 0.85

- /color_w 1

top以外の色を全て白にします。

- /color_r 1

top以外の色を全て赤にします。

- /color_g 1

top以外の色を全て緑にします。

- /color_b 1

top以外の色を全て青にします。

- /color_c 1

top以外の色を全てシアンにします。

- /color_m 1

top以外の色を全てマゼンタにします。

- /color_y 1

top以外の色を全てイエローにします。

- /color_rainbow 1,0 (toggle)

全体の灯体で虹色に変化する照明にするときは1、虹色の变化を止めるときは0を送る

- /led_color_hue 0-1 (float)

top, pocket pro以外のLEDの色情報です。

0.0から1.0までを(赤黄緑青紫赤)の順番でフェードする。

目安： 赤 - 0.0 / 1.0

黄 - 0.16 緑 - 0.33

シアン - 0.5 青 - 0.66

紫 - 0.76 ピンク - 0.85

- /led_color_light 0-1 (float)

/led_color_hueで指定した色相に対しての彩度を指定できます。

0.5が一番鮮やかで、1の時は白、0の時は黒になって色が消えます。

出力について

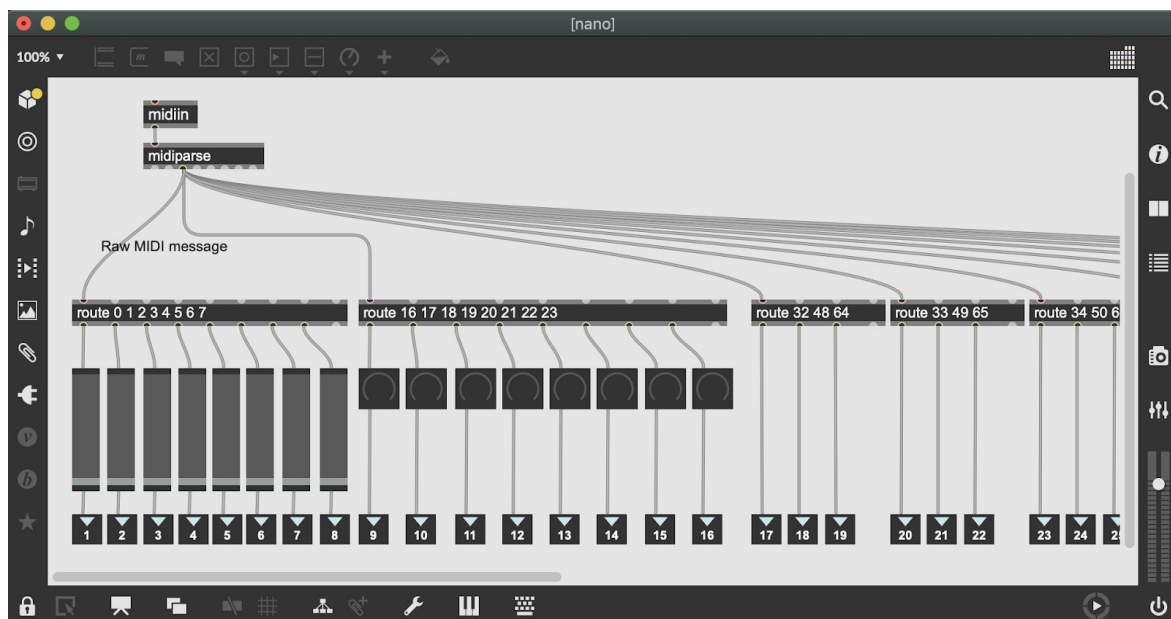
出力を行うには、Maxが必要になります。

送りたいパラメータを取得し、“udpsend”オブジェクトで出力してください。
好きなパラメータを出力できます。

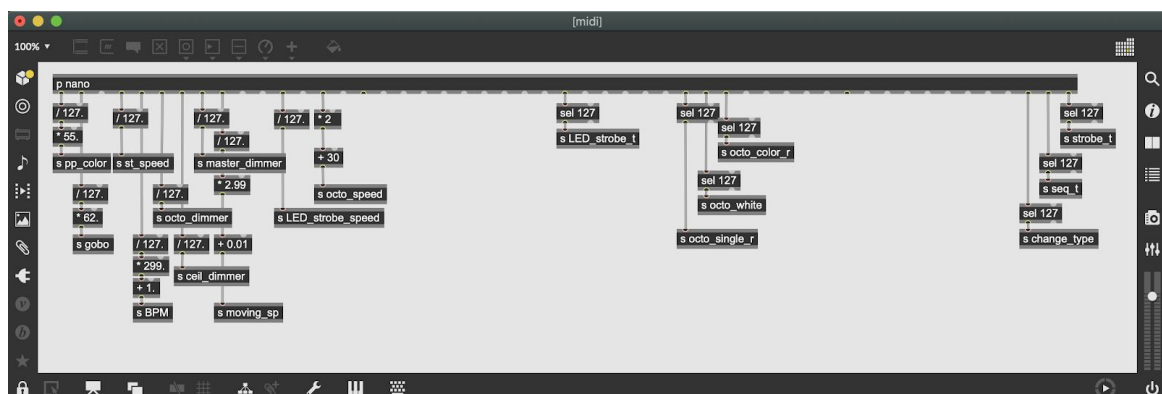
midiコントローラの接続と入力

midiコントローラからの入力を行う場合は、Maxが必要になります。
midiコントローラの接続を行い、それぞれの値を”OSCでの入出力”欄に記入したパラメータに指定すると、制御できます。

例えば任意のmidiコントローラからの入力を



このようにrouteオブジェクトでフィルタリングし



のようにそれぞれのパラメータへ値を送ると制御することができます。
使いやすいmidiコントローラを接続して、お試しください。
また、sendオブジェクトにて”preset_wSL”に数字を送ると、送った番号のプリセットを呼び出します。

その他

照明機材の接続方法

DMX(DMX512)について

国際的に統一され使用されている調光信号です。

ディマー(光量)以外にもムービングライトやLEDなどの制御信号としても使用されています。DMXは制御信号を120Ωの5ピン(3ピン)ケーブル1本にまとめることができます。

これにより、独自規格ではなくメーカー問わずに制御できるようになりました。

名前の通り1つのケーブル(1系統)で512チャンネルの信号を伝送できます。

DMXについては、[Jean-Marc Pelletier氏のサイト](#)に詳しく掲載されています。

ネットワーク化

最近では、これらのDMX信号をイーサネットに変換し、イントラネット、インターネットを介して情報を伝える調光ネットワークシステム(Art-Net, sACNなど)の構築が進んでいます。今回のシステムでもArt-Netを使用しています。

これにより、調光制御信号以外に調光卓、調光盤、昇降装置、ホール内の管理サーバ、ネットワークPC等の間で、双方向による情報のやりとりを行うことが可能になりました。

調光操作卓

操作できる照明設備は「HTP制御機能」と「LTP制御機能」に大きく分類されます。

- HTP制御機能

一般照明の調光操作を行う機能

複数の制御系からレベルをセットしようとした場合、その中で最も高いレベルによってコントロールされる調光制御方式です。

- LTP制御機能

ムービングライトやLED照明器具などの操作が行える機能です。

チャンネルのレベルが最後に指定されたレベルを維持する方式で、ムービングライトやカラーチェンジャーを使用する際、変化を指定しない時に、その都度0レベルのポジションにもどらないようにします。

通常の制御卓ではHTP制御を採用しているため、プリセットを呼び出した後の修正・変更が困難です。

照明の仕込み

照明機材はDMXケーブルをデジチェーン接続(数珠つなぎ)で結線します。

DMXケーブルで繋ぐのは、同じ系統(ユニバース)の機材同士です。

Maxを使ったカスタム設定

基本的な操作については本システムで完結しますが、マニュアル内にも出てきた通りいくつかの設定にはCycling '74の”Max”が必要になります。
照明信号への変換や、それぞれの処理の実装については[Qiita](#)などに記載していますので、ご覧ください。