

EYESY User Manual OS v3



EYESY OS v3.0 - March 2025

©2025 Critter & Guitari. All Rights Reserved. *本書は株式会社アンブレラカンパニーのオリジナル制作物です。無断での利用や公開、複製を固く禁じます。

EYESY は 2025 年に大規模なソフトウェアアップデートが行われています。本マニュアルは最新の EYESY OS バージョン 3 以降に対応しています。ソフトウェアのバージョンを確認するには、左上のオンスクリーンディスプレイ (OSD) ボタンを押してください。お使いの EYESY がバージョン 2.3 の場合、micro SD カードを書き換えることでバージョン 3 にアップグレードできます。手順についてはセクション 6.1 を参照してください。

同梱物一覧

開封した箱の中には、以下の 3 つのアイテムが含まれています：

- **EYESY 本体** - すぐに使用可能です！
- **USB-WiFi アダプター** - EYESY を WiFi に接続するためのものですが、動作には必須ではありません。

- **電源アダプター** – この電源アダプターは、米国仕様のプラグが付属していますが、100～240VAC、50/60Hz の入力電圧に対応しています。ご自宅のコンセント形状に適した変換アダプターが必要になる場合があります。

警告

1. 本ユーザーガイドおよび印刷された保証書に記載されたすべての警告と注意事項を遵守してください。
2. 本製品はユーザーによる修理ができません。損傷した場合はメーカーにお問い合わせください。
3. 本デバイスは、光過敏性発作を引き起こす可能性のある映像を生成することがあります。使用中に不快感、方向感覚の喪失、または身体的異常を感じた場合は、直ちに使用を中止し、医師に相談してください。

ビジュアルを作り出そう！

以下の手順に従って、サウンドをビジュアルに変換しましょう。

1. **EYESY の HDMI®またはコンポジットポートを、電源が入ったモニター、プロジェクター（ビーマー）、ビデオミキサー、ビデオキャプチャーデバイス、またはその他のディスプレイに接続します。**
2. **1/4 インチのモノラルまたはステレオオーディオケーブルを「Audio In」ポートに接続します。**
3. **電源アダプターをコンセントに接続し、そのプラグを EYESY 背面の左端にある 9VDC 電源ポートに接続します。**
4. **電源スイッチを「オン」にします。** EYESY の上部列のステータス LED が淡い緑色に点灯し、その後、起動中に青色で点滅します。接続されたディスプレイには、起動プロセスが表示されます。
5. **EYESY の上部列の「オンスクリーンディスプレイ (OSD) 」ボタンを押します。**（このボタンのアイコンには、画面の下部に文字が表示されているようなデザインがあります。）
 - 画面に現在のシステム情報のオーバーレイが表示されます。
 - もう一度ボタンを押すと、このオーバーレイをオフにすることができます。
6. **オーディオをビジュアルに変換するには、1/4 インチのオーディオケーブルを通じて、接続されたデバイスから音声を送信します。**
 - 左下の 2 つのボタンを使用して、パターン生成モードを変更します。
7. **終了する場合は、電源スイッチを「オフ」の位置にスライドさせます。**

- 。 ステータス LED が消灯するまで待ち、それから電源を取り外してください。

1. EYESY のコンセプト

この不思議なアイコンが並ぶボックスは何？！

EYESY の世界へようこそ！このビデオインストゥルメントは多様な用途に対応でき、使う人によって異なるスタイルで活用できます。まずは、EYESY とは何か、どこから来たのかを説明しましょう。その後、基本概念を確認し、共通の言語で話せるようにします。そして最後に、このマニュアルの使い方について説明し、あなたのビジュアル表現の世界へ踏み出せる準備を整えます。それでは、EYESY の起源について見ていきましょう…

1.1 EYESY はどこから来たのか？

私たち **Critter & Guitari**（こんにちは！）は、これまでにビデオシンセサイザーをいくつかリリースしてきました。そして、今でもそれらのデバイスを大切に思っています。その基本的な使い方はとてもシンプルでした。

1. **オーディオ入力を接続し、ライブのサウンドを EYESY に送る。**
2. **ビデオ出力をプロジェクターやディスプレイに接続する。**
3. **いくつかのノブを調整する。**
4. **音楽を演奏し続けると、ビジュアルが自動的に同期して再生される。**

EYESY もこの「セルフサービス」のアプローチを継承しつつ、さまざまなモードやオプションが追加されています。

EYESY は開封後すぐに使えますか？

はい！ EYESY を接続して電源を入れると（詳細は「ゼロ章」を参照）、すぐに工場出荷時のモードでビジュアルを生成し始めます。

EYESY には異なるビデオ生成方法がありますか？

はい！

EYESY のパターン生成モードは、前面パネル左下の **モードセレクターボタン**（目のアイコン付近）を押すことで、前後に切り替えることができます。プリロードされている工場出荷時のモードでは、多様なビジュアルエフェクトが用意されています。好みのビジュアルを選び、自分だけのモードセットを作成することも可能です（詳細は **第 4 章** で説明します）。

EYESY はビデオ楽器として演奏できますか？

はい！

EYESY のトップパネルにある **ノブとボタン** は、それぞれのビジュアルモードの内部設定を直接操作するためのものです。さらに、**MIDI コントローラー** を MIDI In ポートに接続するか、USB-MIDI デバイスを使用することで、制御の幅を広げることができます。

ノブの動きを記録して、小さなアニメーションを作成できますか？

はい！

内蔵の **ノブシーケンサー** を使用すると、ノブの動きを自動化し、ハンズフリーで操作することが可能になります。

EYESY を接続して、あとはベースを演奏するだけでいいですか？

はい！

オーディオが EYESY に入力されていれば、**いくつかのノブを調整するだけで、音に反応するビジュアルが得られます。**

EYESY を MIDI で制御できますか？

はい！

EYESY は **MIDI ノート** や **MIDI クロック** に反応します。

Ableton Live などの **DAW (デジタルオーディオワークステーション)** やパフォーマンスソフトウェアを使用して、EYESY に MIDI 信号を送ることができます。これにより、EYESY のパラメーターを自動制御したり、音楽と同期したビジュアルを作成したりできます。さらに **オンスクリーンメニュー** を使用すれば、**プログラムチェンジメッセージ** にシーンをマッピングすることも可能です。

EYESY が生成した映像をスクリーンショットとして保存できますか？

はい！

EYESY の **下部列のスクリーンショットボタン** (カメラのアイコン) を押すと、現在の映像を画像ファイルとして保存できます。さらに、**プログラムを改造できるユーザー** であれば、**直近にキャプチャしたスクリーンショットをビジュアル出力に再注入するモード** を作成することも可能です。これは、**フィードバックや再帰的エフェクト** を作り出すのに役立ちます。

EYESY に静止画像を入力素材として使用できますか？

はい！

各モードは、**モードのファイル構造内の静止画像** を使用できるように作成できます。

EYESY のモードを編集できますか？

はい！

EYESY のすべてのモードは、**Python** で書かれており、**Pygame** のグラフィックライブラリを使用しています。したがって、**プログラミングができる人ならモードを自由に編集できます**。さらに、新しいモードを **ゼロから作成することも可能** です。

EYESY の使い方に「正解」はありますか？

いいえ！

EYESY は **ビジュアルを作成するためのツール** です。

- ライブパフォーマンスで使いたい
- 出力映像をキャプチャして編集したい
- 誰かに EYESY を操作してもらおう
- 自分でシーンプリセットを曲ごとに切り替える
- テキストエディタを開いてカスタムモードをコーディングする
- 他のユーザーが作ったモードをダウンロードする

どの方法でも OK です！自分の目的に合った使い方を教えてください。

1.2 EYESY の基本概念

EYESY の背面に貼られた **ステッカー** などにも、EYESY のコンセプトを示すキーワードが記載されています。ここで「**モード**」と「**シーン**」の違いを明確にしておきましょう。

モードとは？

EYESY の **モード (Mode)** は、ビジュアルを生成する方法のことです。オーディオシンセサイザーで言えば「**パッチ**」に相当します。モードはフロントパネルのボタンで即座に切り替えることができ、**すべてのモードは microSD カードに保存されています。**

工場出荷時のモードには、主に **2つのタイプ** があります。

- **スコープモード (Scope Mode)** → **オーディオ信号を直接視覚化** するモード
 - **トリガーモード (Trigger Mode)** → **一定の閾値を超えたときにビジュアルを生成** するモード
-

シーンとは？

シーン (Scene) は、**プリセットのようなもの** です。各シーンには、

- **使用していたモード**
 - **5つのノブの設定値**
 - **Persist Toggle ボタンの ON/OFF 状態**
- が保存されます。

次の章では、これらの概念をどのように活用するかを詳しく説明します！

1.3 カラーパレット

EYESY が **スコープモード** と **トリガーモード** に分類されるのと同じように、**カラーの扱いにも一定のルール** があります。各モードは独自のビジュアル表現を持っていますが、すべてのモードで一貫したユーザー体験を提供することが目的です。

通常、

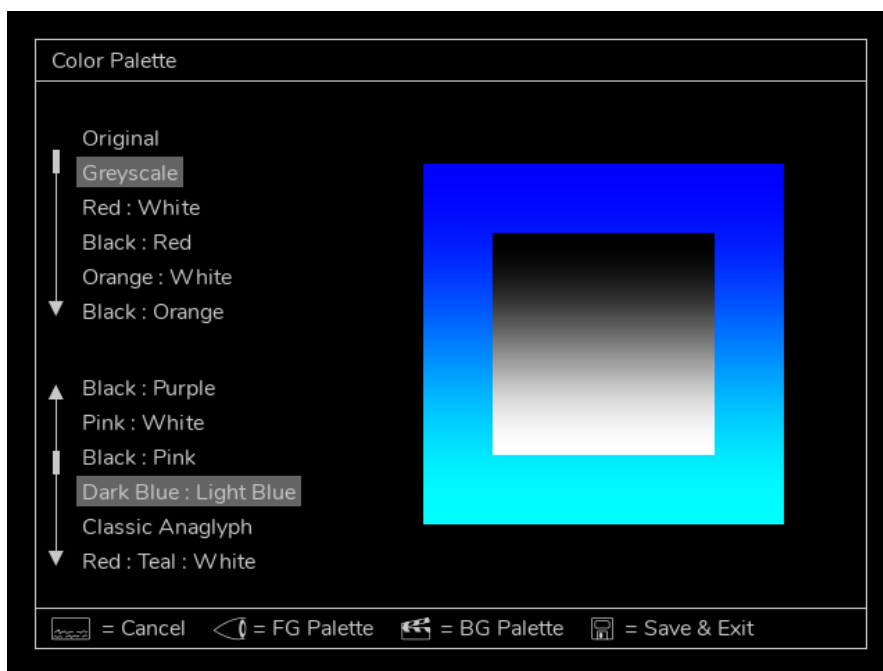
- **ノブ 4** は **前景色（フォアグラウンドカラー）** を制御し、
- **ノブ 5** は **背景色（バックグラウンドカラー）** を制御します。

OS v3 からの新機能：カラーパレットの選択

OS v3 から、**カラーパレットを選択**できるようになりました。これにより、すべてのモードで **同じカラーパレットを共有** できます。

- **白黒のモノクロ表現がいい？**
- **青のグラデーションを使いたい？**
- **カラフルな虹色を試したい？**

様々なパレットから選択し、**自分のビジュアル全体の雰囲気** を決定しましょう。



Example of the palette selection screen

カラーパレットの選択方法については、**次の章** で詳しく説明します。

1.4 ビデオ出力オプション

EYESY には **HDMI** と **コンポジット** の 2 つのビデオ出力オプション がありますが、**同時に使用できるのは 1 つだけ** です。

起動時の動作:

1. EYESY が起動すると、HDMI ディスプレイが接続されているかを確認します。
 - HDMI ディスプレイが検出された場合 → HDMI 出力が**有効** になります。
 - HDMI ディスプレイが検出されない場合 → コンポジット出力 (RCA 端子) が**有効** になります。

設定可能なビデオオプション:

- **HDMI の解像度は選択可能** (解像度の選択方法については **第 3 章** を参照してください) 。
- **コンポジット出力のフォーマット (NTSC または PAL)** は **設定メニュー** で変更できます。

この **自動切り替え機能** により、最新のデジタルディスプレイから **クラシックなアナログビデオ機器** まで、さまざまな環境での互換性が確保されています。

1.5 このマニュアルの使い方

本マニュアルの一部の章（例えばこの章！）は **すべてのユーザーにとって有益** ですが、EYESY の **使用目的** によって特に役立つ情報が異なります。

- **どのユーザーにとっても**、「クイックスタート（第0章）」と「基本概念（第1章）」の情報は役に立ちます。
 - **開封後すぐに使いたい方** は、「第2章」のハードウェアの操作方法を確認してください。この章では、各パネルのコントロールやポートの機能について詳しく説明しています。
 - **追加の設定や機能を知りたい方** は、「第3章」を参照してください。
 - **新しいモードを追加・削除したい、または独自のモードを作成したい方** は、「第4章」と「第5章」を参考にしてください。EYESY に適したコンテンツの準備方法が説明されています。
-

2. EYESY の基本操作

ここまでで、EYESY の接続と起動方法、基本的な使用方法、システム の概念について説明しました。 **では実際に操作して、ビジュアルを作成してみましょう！**

この章では、**ユーザー視点** で EYESY の操作を順を追って説明します。

ハードウェアのコントローラーを **手で操作** しながら、**背面パネルのポート** にも注目していきます。

各コントロールの **役割と関係性** を明確にしなが ら、**EYESY の機能と一般的な操作方法を理解** できるようにしていきます。

それでは、**ノブを回し、ボタンを押して、何が起ころのか見てみましょう！**

2.1 システムコントロール

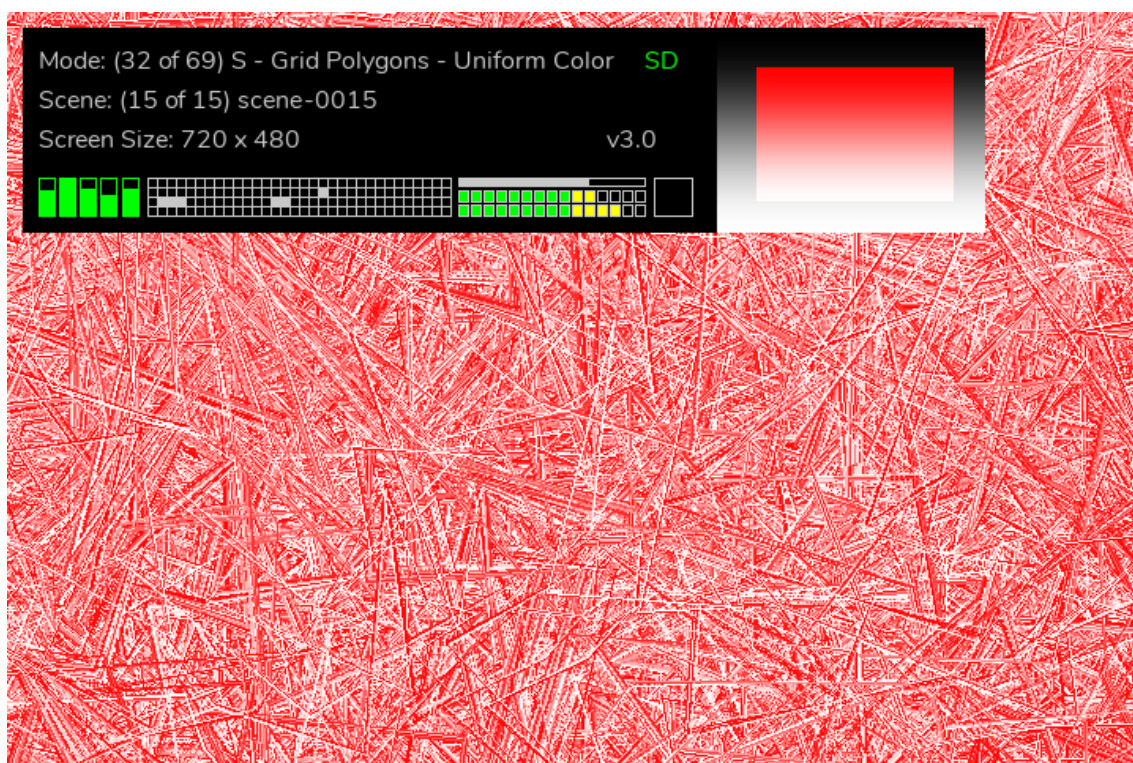
EYESY の上部列には **システムレベルのコントロール** が配置されています。

これらのコントロールを **左から順番に** 説明していきます。



オンスクリーンディスプレイ (OSD) ボタン は、現在のシステム情報を表示する **オーバーレイ** を呼び出す **トグルボタン** です。

以下の画像は、OSD の例です。



OSD（オンスクリーンディスプレイ）に表示される情報

画面上部に表示される情報:

- 現在のモード
- 緑色の「SD」または「USB」アイコン（モード、シーン、設定の保存場所を示す）
 - SD → 内部 SD カードを使用中
 - USB → 挿入された USB ドライブを使用中
- 直近に読み込まれたシーン（該当する場合）
- 画面サイズ
- 現在の OS バージョン

画面下部の表示内容（左から順に）:

- 各ノブの現在の位置
- 受信中の MIDI ノートのステータス

- オーディオゲイン設定とステレオ VU メーターのバーグラフ
 - トリガーが現在作動しているかのインジケーター
 - 入力オーディオ信号のレベルメーター
-

画面右側:

- 2つのグラデーションが表示されており、**現在使用中の前景色と背景色のカラーパレット**を示します。
- **OSD は現在のモードで発生したエラーメッセージも表示**します。これは、カスタムモードをプログラムする際に便利です。

OSD ボタンをもう一度押すと、OSD の表示がオフになります。

ステータス LED の動作

ステータス LED は、EYESY のハードウェアの状態や特定の動作を示すために、さまざまな色で点灯します。

- **暗い (消灯)** : EYESY が動作していません (電源がオフになっている、またはシャットダウン済み)。
 - **緑 / 水色の点滅**: EYESY が起動中。
 - **白**: EYESY が正常に動作中。
 - **マゼンタ**: ノブセンサーの録音待機状態。ノブを動かすと録音を開始。
 - **赤**: ノブセンサーが録音中。
 - **緑**: ノブセンサーが再生中。
-

シフトボタン (Shift Button)

シフトボタンを押すと、**追加機能** (オンスクリーンメニューを含む) にアクセスできます。

これらの機能の詳細は、**この章の後半と第 3 章** で説明します。

パーシストグルボタン (Persist Toggle Button)

パーシストグル (Persist Toggle) は、すべてのモードで利用できるユニークなエフェクトです。EYESY の描画処理に関する基本的な仕組みを理解すると、その動作がわかりやすくなります。

通常の描画処理:

- 各 EYESY モードには「**描画関数 (draw function)**」があります。
- これは、**各フレームごとに実行** されます。
- **通常は、描画関数が実行される直前に画面がクリア**されます。
 - これにより、背景色を制御するモードでは、画面全体が指定した色で塗りつぶされます。

パーシストグルの動作:

- **パーシストがオンの場合: 画面をクリアせず、新しいフレームが 前のフレームの上に描画** されます。
 - これにより、連続的な描画が重なり、新しいタイプのビジュアル表現が可能になります。
 - モードを変更するときに特に面白い効果が得られます。
 - **パーシストをもう一度押すと、通常の描画 (クリアあり) に戻ります。**
-

パーシストと背景色の関係

- パーシストをオンにすると、**背景色の変更が画面に反映されなくなります** (クリア処理が無効化されるため)。
- パーシストをオフに戻すと、**背景色の変更が再び適用** されます。

この動作を活用すると、**独自のビジュアルスタイル** を生み出すことができます！

2.2 モードパラメーターコントロール



中央列の5つのノブについて

EYESYの中央列には**5つのノブ**があります。これらのノブの**役割は現在のモードに依存**します。

ノブは、以下のような**モードのパラメーター**を制御できます。

- **要素のサイズ**
- **位置**
- **数**
- **色**
- **変化の速度** など

また、ノブの機能は**1つの明確なパラメーターのみを調整する場合**もあれば、**複数のパラメーターを組み合わせた「マクロ」機能**を持つこともあります。

各モードの詳細なノブ設定は、実際に使用することで把握できますが、**工場出荷時のモードは必ず5つのノブを持つ**という点を覚えておくと良いでしょう。

モードとシーンでのノブの動作の違い

第 1 章で「モード」と「シーン」の違いについて説明しましたが、ここでは **ノブの動作** に関してもう少し詳しく説明します。

モードを切り替えた場合のノブの動作

- **現在のノブの値は新しいモードにも引き継がれます。**
- 例えば、ノブ 1 を 3 時の位置、ノブ 3 を 10 時の位置にセットした場合、新しいモードに切り替えても **同じ値が適用されます。**

シーンを切り替えた場合のノブの動作

- **現在のノブの値は無視され、シーンに保存された値が適用されます。**
- つまり、シーンをロードすると、**シーンに保存されたノブの値が EYESY に適用** され、**現在のノブの物理的な位置とソフトウェアの値が一時的に切り離されます。**

シーンロード後のノブの動作について

- シーンは **EYESY の現在の状態を記録したスナップショット** であり、モードの種類やノブの設定、パーシストトグルの状態が含まれています。
- **ノブを回すと、新しい値が EYESY のビデオエンジンに即座に適用されます。**
- これは、**シーンがロードされた後でも同じです。**
- 例えば、ロードされたシーンのノブ設定と現在のノブの物理位置が大きく異なる場合、最初の調整で大きな変化が起こることがあります。
- しかし、一度ノブを回すと、以降の調整はスムーズな値の遷移になります。

モードやシーンの切り替え、シーンの保存方法について…

それについては、ちょうど次のセクションで説明します！

2.3 機能コントロール

EYESY の下部列には、ナビゲーションやアクションを操作するためのボタンが配置されています。これらのボタンはすべて **メープル製** で、左から順に見ていきましょう。



モードセレクトターボタン

EYESY の **左端（目のアイコン付近）** にある 2 つのボタンは、**モードセレクトターボタン** です。

- **左ボタン** を押すと **前のモード** に戻り、**右ボタン** を押すと **次のモード** に進みます。
- このボタンを長押しすると、モードを高速で切り替えることができます。
- 選択可能なモードは、EYESY の **microSD カード** または **USB ドライブの「Modes」フォルダ** に保存されています。

シーンセレクトターボタン

シーンセレクトターボタン（**カチンコのアイコン付近**）は、**保存されたシーンを切り替える** ためのボタンです。

- **左ボタン** を押すと **前のシーン** に戻り、**右ボタン** を押すと **次のシーン** に進みます。

- ボタンを長押しすると、シーンを高速で切り替えることができます。
-

シーン保存ボタン

フロッピーディスクのアイコン がついたボタンを押すと、現在の設定を **新しいシーンとして保存** できます。保存される内容は次の通りです。

- **現在のモード**
- **前景色と背景色のカラーパレット**
- **5つのノブの値**
- **パーシストグル (Persist Toggle) ボタンの状態**

また、シーン保存ボタンを約 2 秒間長押しすると、**現在のシーンを削除** できます（この操作は取り消せません）。

- **Shift + シーン保存ボタン** → **現在のシーンを上書き更新** できます。
 - 既存のシーンを微調整したい場合は、シーンを選択 → 設定を変更 → Shift + シーン保存ボタンで更新できます。
-

スクリーンショットボタン

カメラアイコン のボタンを押すと、**現在のビデオモードの出力を画像として保存** できます。

- スクリーンショットは EYESY の「Grabs」フォルダ に PNG 形式 (0.png、1.png…) で保存されます。
 - **オンスクリーンディスプレイ (OSD) の情報はスクリーンショットには含まれません。**
 - 保存した画像は **EYESY エディター (第 4 章で解説)** を使用してアクセスできます。
-

トリガーボタン

右端にある「！」マーク付きのボタン は **トリガーボタン** です。

- このボタンを押すと、外部トリガーと同じ動作が発生します（トリガーソースの設定については第〇章を参照）。
- トリガーモード（第 1 章で説明）では、特にこのボタンが便利です。
- オーディオが入力されていない場合、このボタンを押すと内部でサイン波が生成され、オーディオバッファが埋められます。

2.4 Shift + ボタンショートカット

Shift ボタン を押しながら別のボタンを押すと、以下の追加機能にアクセスできます。

- **Shift + モード前進 / 後退** → 現在のモードで使用する **前景色のカラーパレット** を選択。
 - OSD を有効にすると、現在選択されているパレットが表示されます。
 - ボタンを長押しすると、パレットの切り替え速度が速くなります。
- **Shift + シーン前進 / 後退** → **背景色のカラーパレット** を選択。
 - OSD で選択状態を確認可能。
- **Shift + 保存** → **現在のシーンを上書き保存**。
- **Shift + スクリーンショット / トリガー** → **ノブシーケンサーの制御**（次のセクションで解説）。
- **Shift + OSD ボタン** → **オンスクリーンメニューを開く**（詳細は次章で解説）。
- **Shift + ノブ 1** → **オーディオ入力ゲインを調整**。
 - OSD の VU メーター上にバーグラフとして表示。
 - 設定したゲイン値は次回の起動時にも保持されます。
 - **スコープモード（オシロスコープ表示）を適切に動作させるために重要です**。
 - トリガーモードで正しくトリガーが作動するように調整する際にも使用。

オーディオ入力ゲインの適切な設定例:

- **ギターやマイクを EYESY の Audio In に直接接続する場合: ゲインを高めに設定。**
 - **シンセサイザーやミキサー出力を接続する場合: ゲインを低めに設定。**
 - **独自の実験をする場合: 適切なゲインを探しながら調整!**
-

2.5 ノブシーケンサー

ノブの動きを記録する

ノブシーケンサーを使用すると、ノブの動きを記録・再生でき、自動化されたビジュアルの変化を実現できます。

- **録音開始:** Shift + トリガーボタンを押すと、**録音待機モード** (LED がマゼンタ) になります。
- **ノブを動かすと録音が始まる** (LED が赤に変わる)。
- **録音停止 & 再生開始:** Shift + トリガー または Shift + スクリーンショット。
- **シーケンス再生中:** ノブの動きが自動適用され、ハンズフリーでの操作が可能。
- **OSD での表示:**
 - 再生中のノブ → 緑
 - 録音中のノブ → 赤
 - 停止中のノブ → 白

シーケンスを保存する

- シーンを保存すると、再生中のノブシーケンスも一緒に**保存** されます。
- 保存したシーンをロードすると、シーケンスも自動的に**再生開始** されます。
- シーケンスを削除したい場合: Shift + スクリーンショットで **再生を停止** し、Shift + 保存で **更新**。

2.6 ファイル構造とストレージ

EYESY は、以下の 4 つの**主要フォルダ** によって構成されています。

- **Modes** → 使用可能なビジュアルモードを保存
- **Scenes** → 作成したシーンを保存 (モードやノブ設定含む)
- **Grabs** → スクリーンショットの保存先
- **System** → システム設定ファイル (MIDI 設定やビデオ設定など)

EYESY 起動時の動作:

- **USB ドライブが挿入されている場合:** EYESY は **USB 上の「Modes」フォルダ** からモードを読み込む。
- **USB がない場合:** 内部の **microSD カード** から動作。

この仕組みを活用すると、異なる **USB ドライブ** を使い分けて **設定を素早く変更** できます。

例えば、異なる **MIDI マッピング** や **ビデオ設定** を持つ **セットアップ** を、**USB ドライブごと** に **管理** できます。

2.7 EYESY の背面パネル（詳細） [📄](#)



EYESY のほとんどの端子は背面に配置されています。これまでの説明の中でもいくつかの端子に触れましたが、ここで改めて **各端子の役割と機能** を整理して説明します。

左から右へ、各端子の説明

① 電源スイッチ (Power Switch)

- **EYESY の電源をオン / オフするスイッチ** です。

- **電源をオフにする際の注意:**
 - スイッチを「オフ」にした後、EYESY が完全にシャットダウンするまで数秒待つ。
 - ステータス LED が暗くなる（消灯）したら、電源ケーブルを抜いて OK。
 - 電源ケーブルを早く抜くと、データが破損する可能性があります。
-

② 9VDC 電源端子

- EYESY 付属の電源アダプター専用の端子。
 - **仕様:**
 - 出力電圧: 9VDC
 - 電流: 1000mA (1A)
 - 極性: センタープラス (+)
 - **推奨:** 付属の電源アダプターを使用してください。
 - **不明な電源アダプターを使用しないでください。機器を破損する可能性があります。**
-

③ コンポジット映像出力端子 (Composite Out)

- RCA 端子を使用したアナログ映像出力 です。
 - **起動時の動作:**
 - EYESY が HDMI ディスプレイを検出しない場合、この端子から映像を出力。
 - NTSC / PAL フォーマットは設定メニューで切り替え可能 (第 3 章参照)。
-

④ HDMI 映像出力端子 (HDMI Out)

- EYESY の映像を HDMI ディスプレイやプロジェクターに出力 します。
 - **起動時の動作:**
 - EYESY が HDMI ディスプレイを検出すると、自動的に HDMI 出力が選択される。
 - **重要な注意:**
 - HDMI 端子は映像のみ出力し、オーディオ信号は送信されません。
 - HDMI とコンポジット出力は同時に使用できません (どちらか一方のみ)。
-

⑤ microSD カードスロット

- ラベルのないスロットですが、EYESY の内部 OS やモードが保存されている microSD カードが挿入されています。
 - PC で直接読み取ることはできません。
 - 通常、このカードを取り出したり操作したりする必要はありません。
 - OS の更新や工場出荷状態へのリセット（ファクトリーリセット）を行う場合のみ、カードをフラッシュすることが推奨されます（第 6 章参照）。
 - WiFi 経由でカードのデータにアクセスする方法については第 4 章で解説。
-

⑥ MIDI 入力端子（MIDI In）

- 外部 MIDI コントローラーを接続するための端子。
 - 仕様:
 - TRS-MIDI Type A ポート（Type B ケーブルは使用不可）。
 - EYESY の MIDI 機能についての詳細は第 3 章で説明。
-

⑦ オーディオ入力端子（Audio In）

- 外部オーディオソースを接続するための端子。
 - 仕様:
 - 1/4 インチ（6.35mm）ステレオ入力。
 - オーディオを入力すると、EYESY はその信号に基づいてビジュアルを生成。
 - オーディオ入力のレベル調整は「Shift + ノブ 1」で設定可能。
-

EYESY の左側パネル

最後に、USB ポートが EYESY の左側に配置されています。

USB 端子（USB ポート）

- USB-WiFi アダプターを接続し、WiFi 機能を有効化。

- USB-MIDI デバイス (クラスコンプライアント対応) を接続し、外部 MIDI コントロールが可能。

EYESY の接続のポイント

1. HDMI とコンポジット出力は同時に使えない。
2. オーディオ入力ゲイン (Shift + ノブ 1) は適切に調整することで、ビジュアル生成の精度が向上。
3. USB を使用すれば、WiFi 接続や MIDI コントローラーの利用が可能。
4. microSD カードは通常操作不要 (OS 更新時のみ使用) 。

次のセクションでは、EYESY のシステム設定について説明します！



EYESY の USB ポートを増やす必要がある場合、このポートに USB ハブを接続できます。接続するハブやその他のデバイスは、クラスコンプライアント (特別なソフトウェアドライバーのインストールが不要) である必要があります。これにより、EYESY との互換性が確保されます。

USB ポートをクラスコンプライアント MIDI コントローラーなどのデバイスの電源供給に使用する場合、EYESY は最大 500 ミリアンペア (0.5A) まで供給できます。より多くの電力が必要な場合は、セルフパワー型の USB ハブを使用してください。

3. オンスクリーンメニューと高度な設定

EYESY には **オンスクリーンメニュー** があり、**MIDI 設定**、**ビデオ出力調整**、**その他のシステム設定** などの高度な設定オプションにアクセスできます。このメニューを使用することで、デバイスの物理的なコントロールを超えた、より詳細なカスタマイズが可能になります。

メニューを開くには、**Shift + オンスクリーンディスプレイ (OSD) ボタン**を押してください。



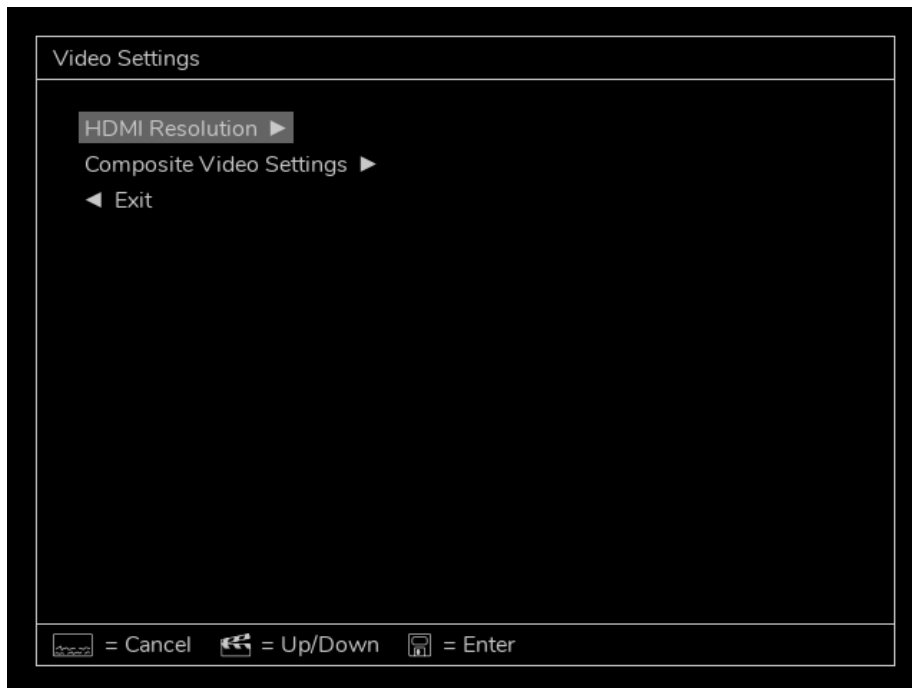
ナビゲーションはシンプルです。**シーン前進ボタン** と **シーン後退ボタン** を使用してメニュー内を上下に移動し、**保存 (Save) ボタン** を押してオプションを選択します。各メニュー画面の下部には、これらの操作の説明が表示されます。

メニューを終了したい場合は、**OSD ボタンを押す**といつでも**メニューを閉じることができます**。

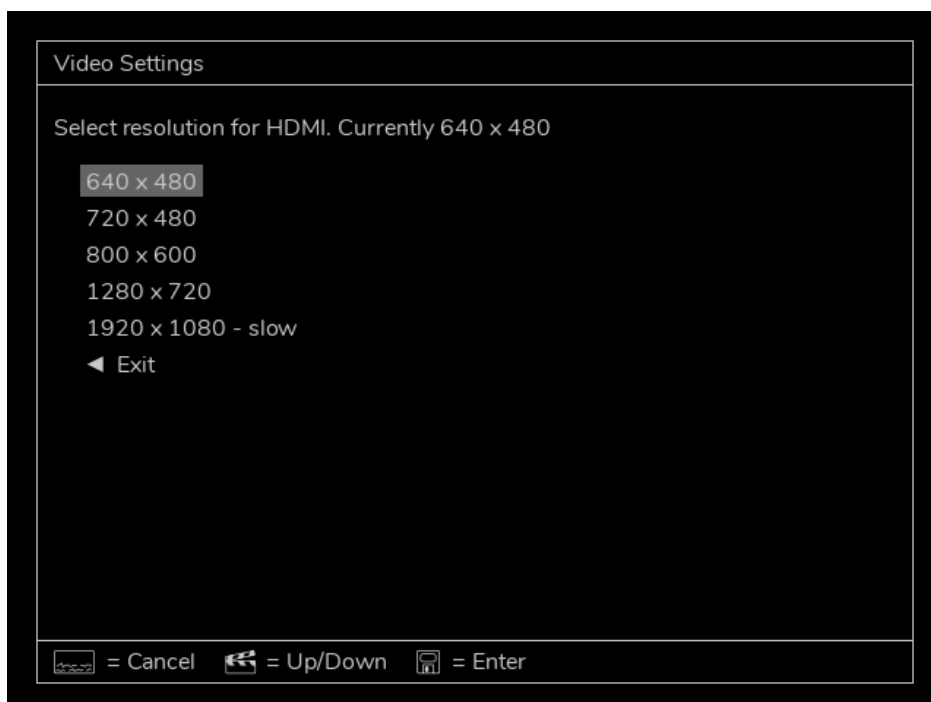
次のセクションでは、各メニューオプションについて詳しく説明します。

3.1 ビデオ設定

ビデオ設定メニューでは、**HDMI の画面解像度** と **コンポジットビデオのフォーマット** を調整できます。

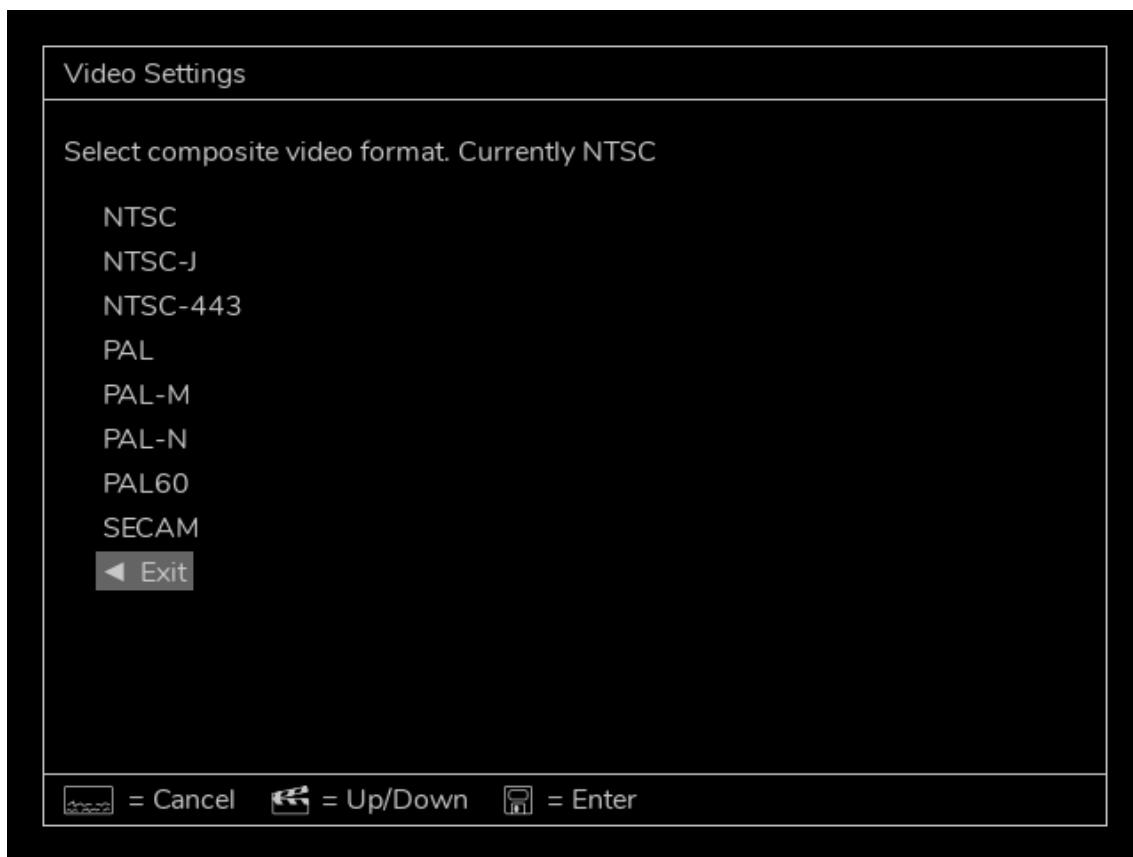


最初のメニューオプションでは、**HDMI の解像度** を選択できます。利用可能な解像度は以下の通りです。



新しい解像度を選択すると、変更を適用するために **EYESY ソフトウェアの再起動** を求められます。

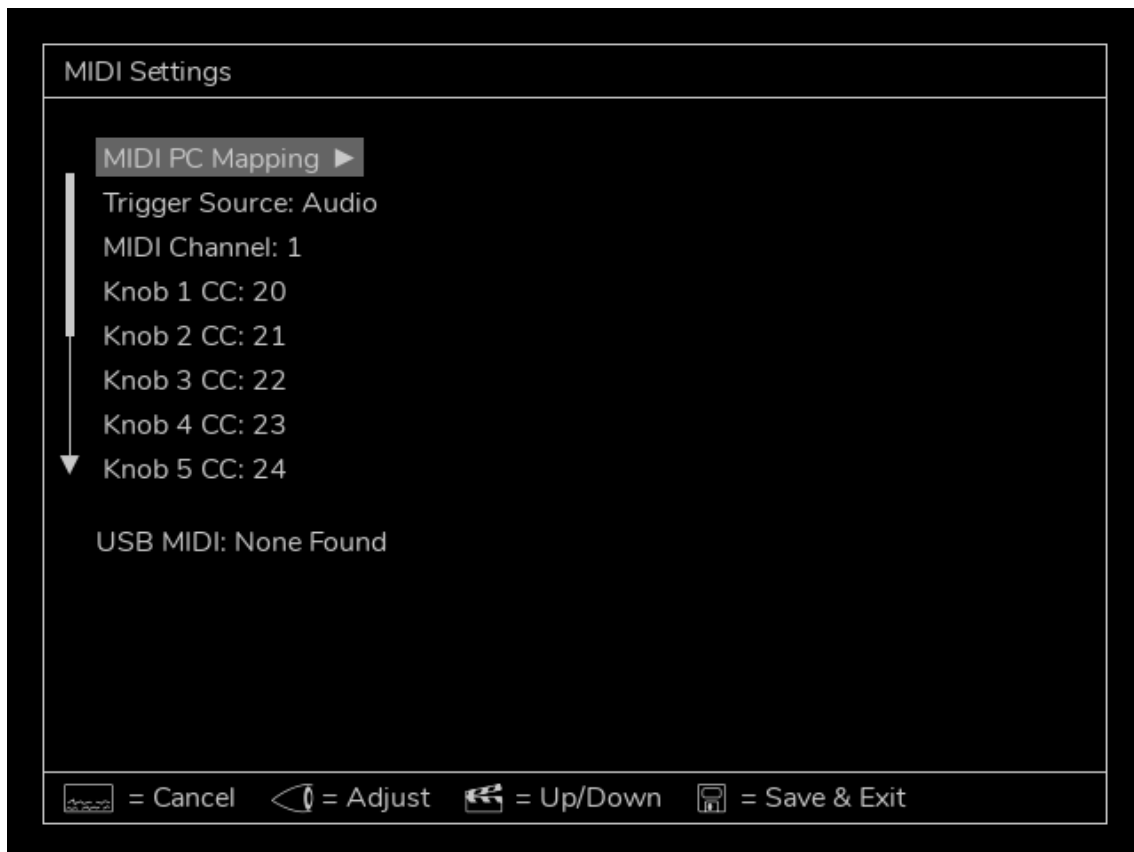
次のメニューオプションでは、コンポジット出力を使用する際のビデオフォーマット（NTSC または PAL）を選択 できます。



コンポジットビデオフォーマットの変更も、適用するには **再起動が必要** となります。

3.2 オーディオ & MIDI 設定

オーディオ & MIDI 設定メニューでは、**MIDI コントロールのマッピングをカスタマイズ** したり、**オーディオ関連の設定を調整** することができます。



3.2 オーディオ & MIDI 設定

モード前進 / 後退ボタン を使用して設定を調整できます。メニューの下部には、接続された **USB-MIDI デバイス** が検出された場合に **リスト表示** されます。

メニューの最初の項目は **MIDI プログラムチェンジシーンマッピング** であり、詳細は次のセクションで説明します。

調整可能な設定

- **トリガーソース (Trigger Source)** - トリガーを使用するモードのトリガーソースを設定
 - **Audio:** オーディオ入力が約 **-5dB** を超えたときにトリガー発生
 - **MIDI Note: MIDI ノートオン (キーを押したとき)** にトリガー発生 (キーを離してもトリガーは発生しない)
 - **MIDI Note または Audio:** いずれかの条件でトリガーを発生
 - **MIDI Clock:** MIDI クロック信号のビート分割に基づいてトリガー発生 (楽曲のテンポと同期可能)

注意:

MIDI クロックメッセージは **システム MIDI メッセージ** であり、EYESY の **MIDI チャンネル設定** に関係なく受信されます。

- **MIDI チャンネル (MIDI Channel)** - MIDI 受信チャンネルを 1~16 の範囲で設定
 - **MIDI コントロール設定 (MIDI CC マッピング)**
 - **Knob 1-5 CC Mapping:** 各ノブに MIDI CC 番号を割り当て
 - **Screen Background Clear CC: Persist Toggle ボタン** (トップ右) の動作と同じ
 - **Foreground Palette Selection CC:** 前景色のカラーパレットを選択
 - **Background Palette Selection CC:** 背景色のカラーパレットを選択
 - **Mode Select CC:** MIDI CC メッセージでモードを切り替える
 - **MIDI ノートによるモード選択 (MIDI Note Mode Selection)**
 - MIDI ノートを使用して異なるモードを選択可能
 - 例: MIDI キーボードを接続すると、各ノートが異なるモードをロード
 - デフォルトでは無効
-

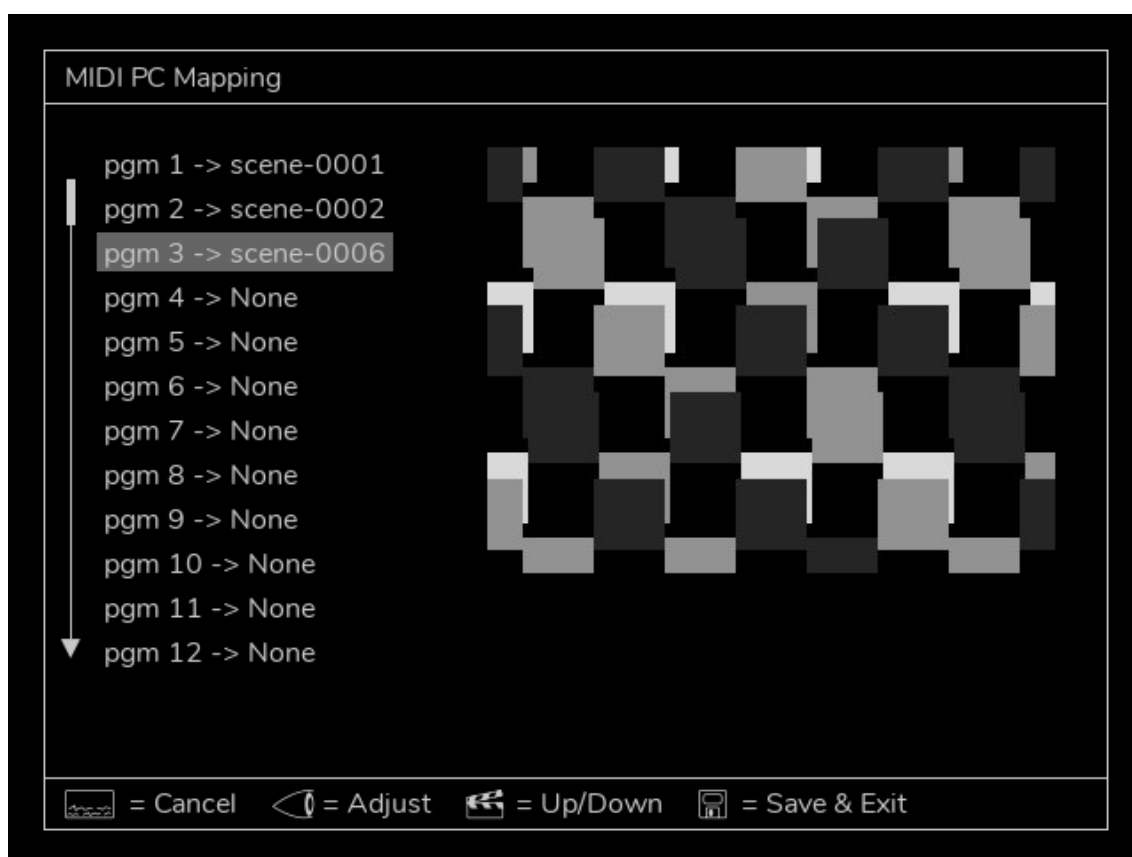
EYESY の MIDI 端子について

- **MIDI In ポートは TRS-MIDI Type A**
 - **Type A 3.5mm (1/8") TRS to 5 ピン DIN アダプターが必要** (古い 5 ピン DIN 機器を接続する場合)
 - **Type B は EYESY では使用不可** (見た目は同じなので注意)
- **TRS-MIDI デバイスとの接続**
 - **TRS-MIDI 出力を持つ機器は、3.5mm TRS ケーブル (Type A) で直接接続可能**
 - **Type A to Type B アダプターも利用可能** (デバイスが Type B の場合)

- **USB-MIDI デバイスの使用**
 - クラスコンプライアント（ドライバ不要）の USB-MIDI デバイスのみ互換性あり
 - EYESY は Linux ベースのシステム のため、Windows / macOS 専用の MIDI デバイスは動作しない可能性がある
 - 互換性が不明な場合は、TRS-MIDI ポートを使用するのが無難

3.3 MIDI プログラムチェンジシーンマッピング

このメニューでは、MIDI プログラムチェンジ（PC）メッセージを使用して特定のシーン呼び出すように設定できます。



各 **プログラムチェンジ番号（1～128）** をシーンに割り当てることができます。**シーン前進 / 後退ボタン** を使用してプログラム番号を選択し、対応するシーンを設定してください。右側には、選択したシーンの小さなプレビュー画像が表示され、確認できます。

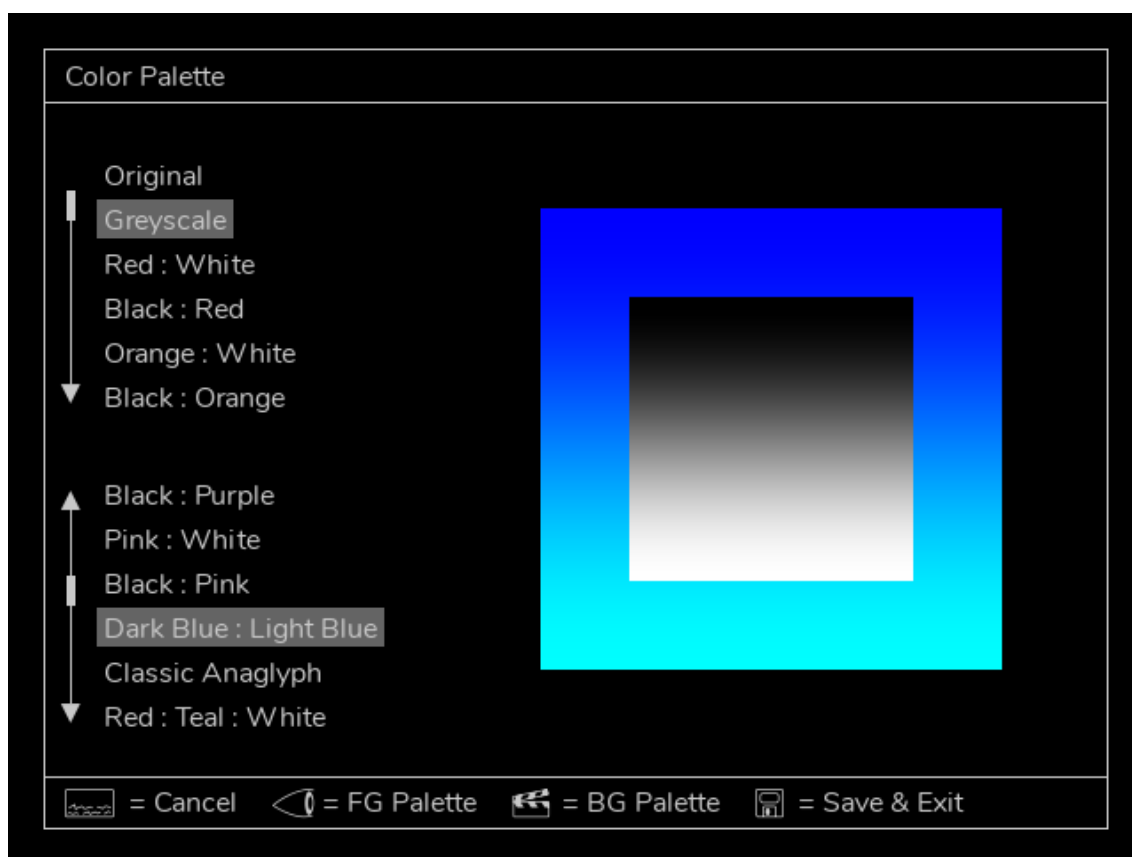
すべてのマッピングを設定したら、**保存 (Save) ボタン** を押して **システム設定に保存** します。これらの設定は再起動後も保持されます。

注意:

割り当てられたシーンが削除された場合、そのシーンに関連付けられたプログラムチェンジ番号の設定も自動的にリストから削除されます。

3.4 カラーパレット

カラーパレットメニューでは、**すべてのモードで使用するデフォルトの前景色と背景色のカラーパレットを選択** できます。



画面には **2つのメニュー** があります。

- **前景色のカラーパレットを選択 (Select Foreground Color Palette)** - モード内の要素を描画する際に使用するパレットを設定します。

- **背景色のカラーパレットを選択 (Select Background Color Palette)** – 背景色に使用するパレットを設定します。

モード前進 / 後退ボタン を使用して利用可能なパレットを切り替え、選択が完了したら **保存 (Save) ボタン** を押して設定を保存し、メニューを終了してください。

最初のリスト項目は、**EYESY (OS v2.3) のオリジナルカラーパレット** です。このオリジナルパレットには **組み込みのカラーバリエーション (ランダム要素)** が含まれています。選択画面の上下 (および OSD) で色が変わるのは **仕様通りの動作** です。

これらのカラーパレット設定は **システム設定として保存され、EYESY の起動時に適用** されます。ただし、シーンには独自のパレット設定が含まれているため、シーンをロードするとこのデフォルト設定は上書きされる可能性があります。このメニューのパレット設定は、**EYESY が最初に起動した際のデフォルトカラーとして機能し、シーンを読み込まれる前に適用** されます。

カラーパレットリストのカスタマイズ

EYESY の OS には、「**palettes.json**」という組み込みファイルがあり、デフォルトのカラーパレットを管理しています。

「**System**」フォルダに同じ名前とフォーマットのファイルを配置すると、このデフォルトのリストを上書きできます。

(詳細は **4.2 節** を参照)

標準の **palettes.json** ファイルは、EYESY のドキュメントメニューにあります。

- ダウンロードし、不要なパレットを削除したり、並べ替えたりできます。
- 完全なカラーパレットのフォーマット例:

```
{  
  "name": "Greyscale",  
  "a": [0.500, 0.500, 0.500],  
  "b": [0.500, 0.500, 0.500],  
  "c": [0.500, 0.500, 0.500],
```

```
"d": [0.500, 0.500, 0.500]
}
```

カスタムパレットの作成

独自のカラーパレットを作成するには、**Palette Picker ツール** を使用できます。

このツールは **EYESY ドキュメントメニュー** にあり、次のような画面になっています。

	Offset	Amplitude	Frequency	Phase
Red	0.5	0.47	0.51	0
Green	0.5	0.42	2.01	-0.5
Blue	0.5	-0.21	0.74	-0.12
Global	0	1	1	0

```
{"name": "My Palette", "a": [0.5, 0.5, 0.5], "b": [0.47, 0.42, -0.21], "c": [0.51, 2.01, 0.74], "d": [0.5, 0.5, 0.5]}
```

Copy JSON

Randomize

This tool was originally created by Karsten Schmidt. [Learn more on thi.ng.](#)

パレットピッカー (Palette Picker) は、パレットの 赤 (Red) 、緑 (Green) 、青 (Blue) の値を表す 3 つのコサイン波形グラフ で構成されています。

グラフの下には コサイン関数の出力 が表示され、これが EYESY の OSD やカラーパレットメニュー に反映されます。

パレットのカスタマイズ方法

出力の下には 波形を調整するためのコントロール があります。

各色のコサイン波には以下のパラメーターが設定できます。

- オフセット (Offset)
- 振幅 (Amplitude)
- 周波数 (Frequency)
- 位相 (Phase)

各列の最下部にある **グローバル設定** では、個々の色の波形全体をスケーリングできます。

水平スライダーを使って値を調整したり、スライダーの下のテキストボックスに数値を入力して範囲外の値を設定することも可能です。

カスタムパレットの保存方法

1. 新しいパレットを作成したら、「Copy JSON」ボタンを押して JSON コードをコピー。
 2. コピーした JSON を、自分の「palettes.json」ファイルの適切な場所に貼り付け。
 3. 最後のパレットでない場合は、閉じカッコ「}」の後に「, (カンマ)」を追加。
 4. 新しいパレットに分かりやすい名前を設定。
 5. 編集した「palettes.json」ファイルを「System」フォルダにアップロード。
 - 詳しくは 3.5 節 および 第 4 章 を参照。
 6. システムファイルを変更した場合は、EYESY のビデオエンジンを再起動。(詳細は 4.1 節)
-

エラーログの確認

もし **palettes.json** のフォーマットが正しくない場合、EYESY はエラーメッセージをログに記録し、デフォルトのパレットを使用します。

ログに次のようなエラーが表示された場合は、**JSON ファイルの記述ミス** がある可能性があります。

```
Error loading /sdcard/System/palettes.json: Expecting ',' delimiter: line 116 column 1 (char 2879), using default palettes.
```

JSON ファイルの書式が不安な場合:

- **DuckDuckGo の JSON バリデーター** などのツールを使用して、**フォーマットを確認・修正** できます。
 - 「**Indent with Four Spaces**」を選択し、「**Validate JSON**」をクリック。
 - **バリデーションを通過した JSON をコピーし、palettes.json に貼り付けて保存。**
 - **EYESY にアップロードする前に、必ず保存を確認。**
-

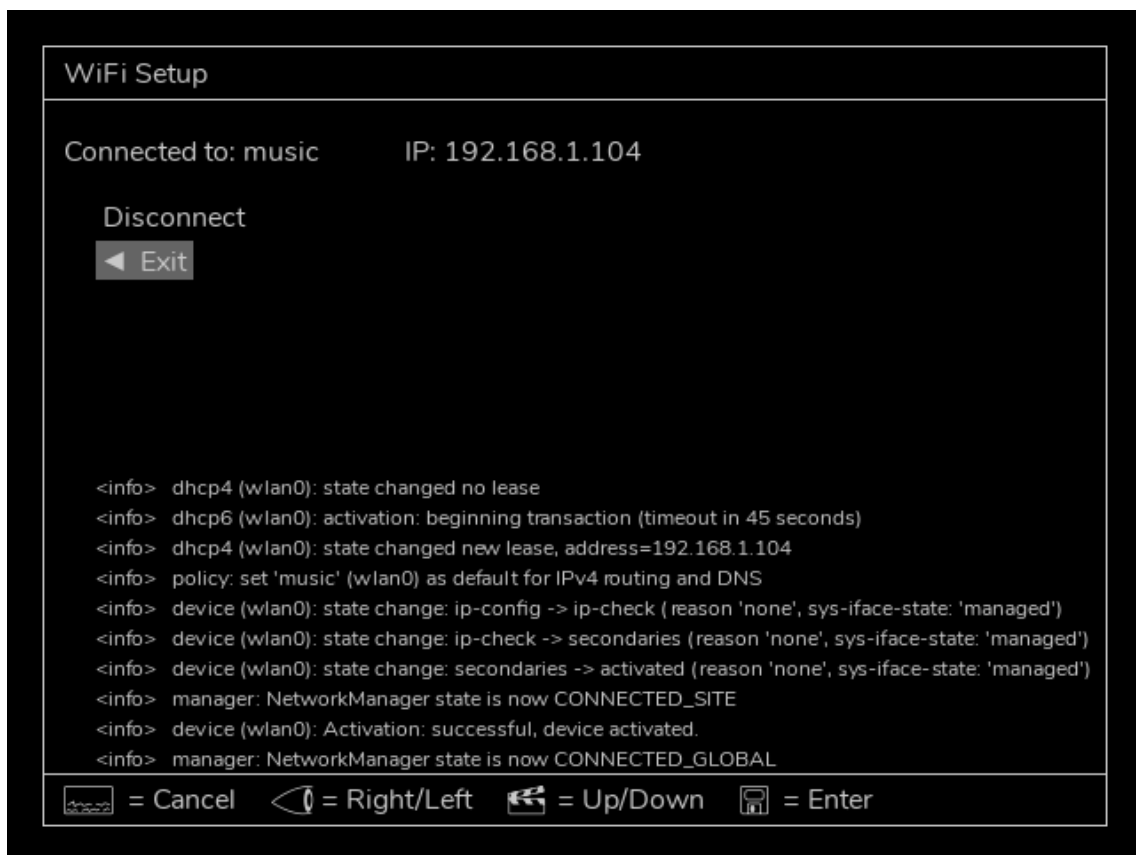
3.5 WiFi およびネットワーク設定

注意:

EYESY は信頼できるネットワーク（自宅の WiFi など）でのみ使用することを強く推奨します。

公共ネットワークや共有ネットワークに接続すると、他者にファイルや設定を操作される可能性があります。

WiFi 設定メニューでは、付属の **USB WiFi アダプター** を使用して、EYESY を WiFi ネットワークに接続できます。



EYESY がすでにネットワークに接続されている場合、**SSID (ネットワーク名)** と **IP アドレス** が表示されます。

IP アドレスは EYESY エディターへのアクセスに重要 です（詳細は次章を参照）。

接続されていない場合、利用可能な WiFi ネットワーク (SSID) のリストが表示されます。

ネットワークを選択すると、**オンスクリーンキーボード**を使用してパスワードの入力を求められます。

注意:

- 付属の USB-WiFi アダプターは **RT5370** または **RTL8192CU** チップセットを使用しています。
- **このアダプターは 2.4GHz 帯の WiFi ネットワークのみ接続可能です。**
- 他の WiFi アダプターの互換性については検証中のため、**動作確認済みの付属アダプターの使用を推奨します。**
- **他の WiFi アダプターについては、フォーラムで情報交換してください。**

画面の下半分にはネットワークのログメッセージが表示され、接続状況の更新を確認できます。

EYESY は **USB イーサネットアダプター** もサポートしています。

ネットワークに DHCP サーバーがある場合、アダプターを接続すると **自動的に接続が確立され、IP アドレスが表示されます。**

3.6 システム設定

システム設定メニューでは、バックアップ管理、WiFi 設定、EYESY ソフトウェアの再起動などの各種ユーティリティ機能を利用できます。



利用可能なオプションは以下の通りです。

- **Backup to USB Drive** – 現在の環境（Modes、Scenes、Grabs、System フォルダ）を、挿入された USB ドライブの **backups フォルダ** にコピーします。
 - EYESY が USB ドライブから動作している場合でも、常に内部 SD カードのデータがコピーされます。
- **Eject USB Drive** – USB ドライブを安全に取り外します。
 - バックアップ後や、内部 SD カードに戻して動作させたい場合に使用。
- **Forget Saved WiFi Data** – 保存された WiFi 接続設定をすべて削除します。
 - ネットワークをリセットまたは変更したい場合に便利。
- **Restart Video** – EYESY のソフトウェアを再起動（デバイス全体の再起動は不要）。

- EYESY が USB ドライブから動作している場合、USB を取り外し、ビデオエンジンを再起動すると、内部 SD カードからの動作に切り替わります。
- システム全体の再起動なしで、ストレージやネットワークの設定を切り替えられます。

これらのオプションを活用することで、ストレージ管理、ネットワークリセット、SD カードと USB 環境の切り替えをスムーズに行うことができます。

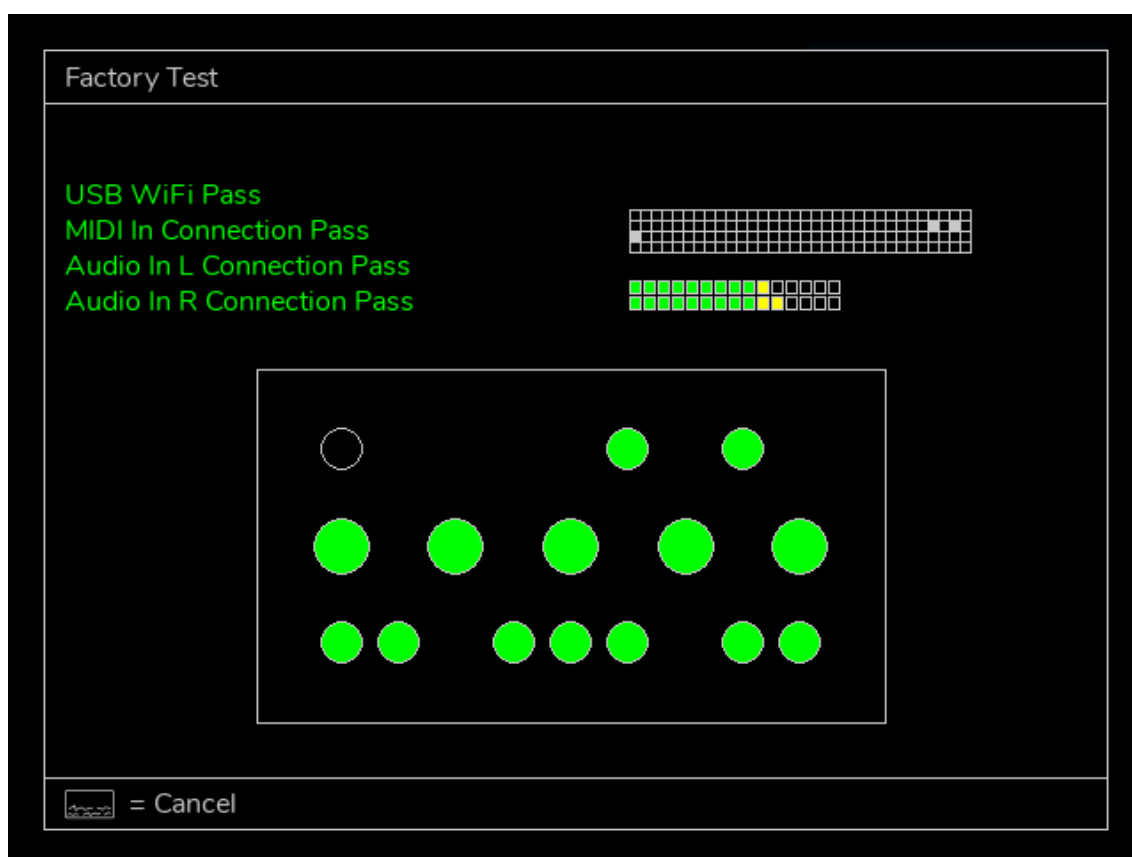
3.7 ハードウェアテスト

このメニュー項目は、ハードウェアの問題を診断するためのテスト機能です。

フォーラムやサポートメールで問い合わせをする際、このテストを実行するよう求められる場合があります。

テストページには、以下の情報が表示されます。

- EYESY のコントロールを示す簡単な線画
- 128 の MIDI ノートのグリッド
- ステレオ VU メーター
- 各機能がテストに合格したかどうかのテキスト情報



EYESY のコントロールテスト

- **ノブ (Knobs) :**
 - 各ノブ (ポテンシオメーター) を **最大左** まで回し、次に **最大右** まで回してください。
 - **この左から右の順番が重要です。** EYESY はこの順番で値を受け取るよう設計されています。
 - **適切な値範囲が出力されると、テストに合格し、ノブの表示が緑に変わります。**
 - **ボタン (Buttons) :**
 - 各ボタンを **3 回押すとテスト合格** となり、ボタンが緑に変わります。
 - **OSD ボタンを押すとテストを終了してしまうので注意してください!**
-

接続端子・ポートのテスト

- **USB WiFi アダプター / USB ポート:**
 - テストを開始する前に、**USB-WiFi アダプターを EYESY の USB ポートに接続し、WiFi ネットワークに接続してください。**
 - **アダプターがローカルネットワークに接続されていると、テストに合格となります。**
 - **詳細は「3.5 WiFi とネットワーク設定」を参照してください。**
- **MIDI 入力ポート (MIDI Input Port) :**
 - **MIDI 入力ポートを、シンセサイザーや DAW などの外部 MIDI 出力デバイスに接続してください。**
 - **外部デバイスと EYESY の MIDI チャンネルが一致していることを確認してください。**
 - **MIDI ノート 60 (C) 、62 (D) 、64 (E) を同時に押すと、テスト合格となります。**
 - **画面右上の 128 マスのグリッドに、受信した MIDI ノートが塗りつぶされて表示されます。**
 - **USB-MIDI デバイスが接続されている場合、そこから受信した MIDI ノートもグリッドに表示されます。**
- **オーディオ入力ポート (Audio Input Port) :**
 - **ステレオ 1/4 インチ (6.35mm) ケーブルを Audio In ポートに接続してください。**
 - **右チャンネルと左チャンネルの音量が十分に大きく、トリガーを作動させるとテスト合格になります。**
 - **画面の VU メーターが最初の黄色のインジケーターに到達すると、トリガーが作動し、テストに合格します。**

注意:

このテストは結果を保存しません。テストページを開くたびに、新しい状態で開始 されます。

3.8 ログ (Logs)

ログメニューでは、EYESY のシステムメッセージを表示します。

- デバッグの際の問題確認
- エラーチェック
- システムの動作状況の把握

などに役立ちます。



ログメッセージはリアルタイムで更新され、EYESY の動作、ネットワークの状態、ファイル操作、システムプロセスに関する情報が含まれる場合があります。予期しない動作が発生した場合、ログを確認する

ことで問題の特定に役立ちます。

より詳しくトラブルシューティングを行うために、ログはモードのエラー、スクリプトのクラッシュ、または欠落しているファイルに関する情報を提供することもあります。

4. EYESY エディター

EYESY をカスタマイズする

まず、ここまで読んでいただきありがとうございます！もしこの章に直接飛んできた場合は、第 2 章を読んでおくことをお勧めします。EYESY の基本的な操作について理解していることを前提に進めます。では、本題に入りましょう。

この章では、**EYESY 本体の操作を超えて、外部との連携方法に焦点を当てます。**

EYESY は工場出荷時のモードでも十分に動作しますが、新しいモードを追加したり、既存のモードをカスタマイズしたりすることで、さらに機能を拡張できます。ここでは、**Web ブラウザ上で動作する「EYESY Editor」** を使用して、その方法を説明します。

この章で扱う内容：

- **ファイル管理:** モードの追加・削除、画像のアップロード、スクリーンショットのダウンロードなど
- **モードのプログラミング:** 新しいモードの作成や、既存のモードのカスタマイズ

4.1 エディターの説明

エディターを使用するには、**EYESY がネットワークに接続されている必要があります。**

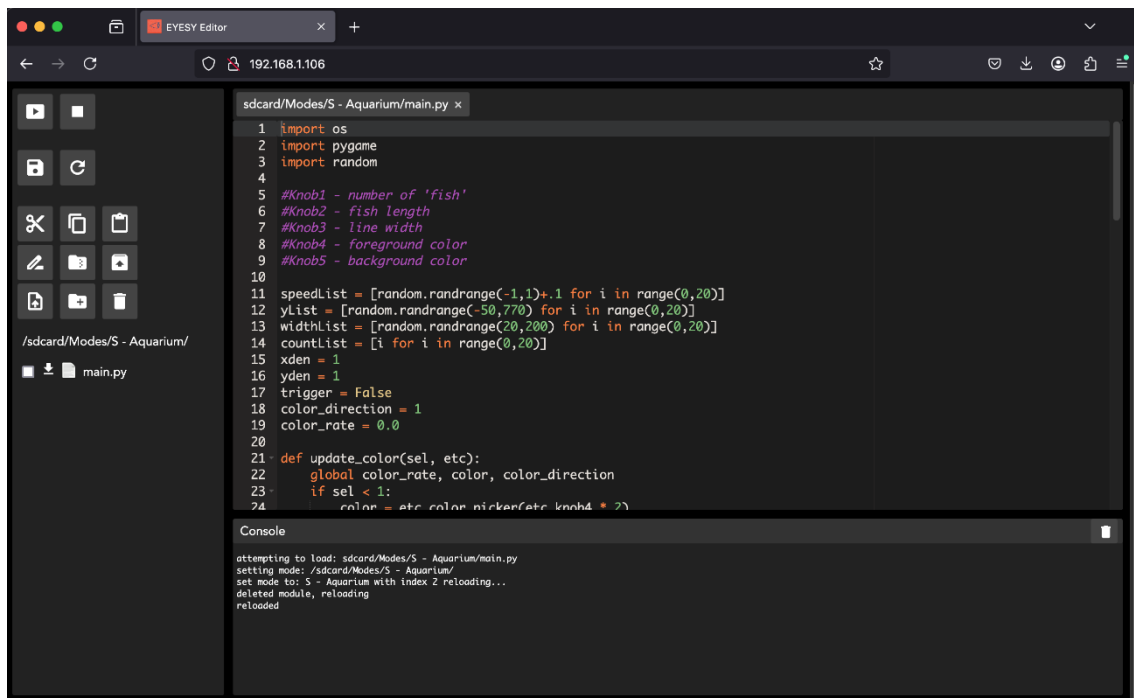
WiFi 接続の方法については、第 3 章を参照してください。

接続後、EYESY の IP アドレスを確認し、それを Web ブラウザのアドレスバーに入力します。

WiFi メニューに表示されている IP アドレスを **http://** の後に入力し、以下のような形式でアクセスしてください。

http://192.168.1.106

これで、EYESY エディターが表示されるはずです。



エディターは、EYESY をリアルタイムで確認および変更するための方法です。コンピューターの Finder やファイルエクスプローラーと同様に、ファイルやフォルダの移動、追加、削除、コピーができます。また、特定のモードのコードを表示、編集、リロードすることも可能です。プログラミングを支援するために、エディターにはコンソールがあり、モードのデバッグができます。

これらの機能を考えると、エディターが 3 つのセクションに分かれているのは当然のことです。

左側には、ビデオ出力とファイルブラウザおよび管理を制御するボタンがある狭いコントロールパネルがあります。ファイルブラウザをクリックして **sdcard/Modes/Python/T - Bits H/main.py** などの場所に移動すると、コードパネル（最も大きなパネル）に **main.py** のコードが表示されます。このコードは編集可能です。

コードの下には、EYESY のターミナル出力を表示する **コンソールパネル** があります。これはコードの動作を確認するために便利で、コード内の **print** コマンドを使用すると、変数の値などをここに表示できます。

次に、これらのセクションについて詳しく説明します。

コントロールパネル

- **Start Video:** このボタンを押すと、ビデオ「エンジン」を再起動します。起動時にはすでにビデオ出力が開始されています。コンソールには、ビデオエンジンの起動時のターミナル出力が表示されます。
- **Stop Video:** このボタンを押すと、ビデオ「エンジン」を停止し、ディスプレイが空白の画面になります。このボタンは、新しいモードを追加した場合や、モードに問題がある場合に便利です。
- **Save:** 現在編集モードのモードを保存します。
- **Reload Mode:** 現在編集モードのモードをリロードします。EYESY がこのモードをまだ実行していない場合、自動的に開始します。
- **Cut, Copy, Paste, Rename, Zip, Unzip, Delete, Upload, New Folder:** ファイルブラウザで使用する一般的なファイル管理コマンド。
 - 各アイテムの左側にある白いボックスをチェックしてから、コマンドを選択する必要があります。（Paste, New Folder, Upload には適用されません。）
 - フォルダを EYESY にコピーする場合や EYESY からコピーする場合、最初に ZIP 圧縮する必要があります。フォルダは ZIP 圧縮しないとアップロードやダウンロードができません。
- **File Browser:** ルートフォルダは 2 つあります: `/sdcard` および `/usbdrive`。
 - `/sdcard` で操作している場合、内部 SD カード上のファイルを編集しています。
 - **USB ドライブが接続されている場合、USB ドライブ上のファイルを編集することも可能です。**
 - `/sdcard` または `/usbdrive` をクリックして選択し、必要な場所を編集できます。
 - EYESY は USB または SD カードのどちらか一方からしか動作しません（オンスクリーンディスプレイで確認可能）。
 - **正しい場所で編集しているかを確認してください。**

コードパネル

- コードパネルはコードエディターであり、行番号と構文のハイライト/カラーリング機能が備わっています。
- 基本的なテキストエディターと同じように、コードの編集やプログラミングが可能です。
- 実際の編集やプログラミングでは、コードパネルをコントロールパネルのコマンドや、コンソールに表示される構文エラーやランタイムエラーと組み合わせて使用します。

4.2 microSD カード内のフォルダとファイル管理

セクション 2.6 で述べたように、EYESY は 4 つの主要なフォルダを使用して動作を管理します。これらのフォルダはデフォルトで microSD カード内にあります。

- **Modes** – 利用可能なすべてのビジュアルモードを保存します。
- **Scenes** – ユーザーが作成したシーンを保存し、モードの選択やノブの設定を含みます。
- **Grabs** – スクリーンショットボタンでキャプチャした画像を保存します。
- **System** – システム設定ファイルを保存します。

EYESY は、/sdcard または接続された /usbdrive のいずれかを、これらのディレクトリの保存場所として使用できます。

/sdcard または /usbdrive のディレクトリ構成例

Grabs/

0.png

1.png

2.png

Modes/

S - Circle Scope Connected/

main.py

T - Density Cloud/

main.py

System/

config.json

gradients.json

Scenes/

scene-0001

scene-0002

Grabs フォルダには、スクリーンショットボタンでキャプチャした画像が保存されます。このフォルダが空であっても問題ありません。

Modes フォルダには 2 つのモードが含まれています。したがって、EYESY で利用可能なモードは **S - Circle Scope Connected** と **T - Density Cloud** となります。

工場出荷時のモードの命名規則では、**モード名の先頭が「S」の場合はスコープモード、「T」の場合はトリガーモードを示します**（詳細は第 1 章を参照）。

Scenes フォルダには、個々のシーンがフォルダ単位で保存されます。シーンごとにフォルダが作成されるため、整理や管理がしやすくなっています。

シーンはアルファベット順にロードされるため、フォルダ名を変更することでロードの順序を変更することができます。

スクリーンショットボタンで撮影したスクリーンショットは Grabs に保存されます。このディレクトリが空であっても問題ありません。

Modes フォルダの中には 2 つのモードがあります。したがって、上記の例では、EYESY で利用可能な 2 つのモードは **S - Circle Scope Connected** と **T - Density Cloud** になります。さらに、工場出荷時のモードの命名規則として、先頭が **S -** の場合はスコープモード、**T -** の場合はトリガーモードを示します（第 1 章を参照）。

Scenes は Scenes ディレクトリ内の個別のフォルダとして保存されます。各シーンには独自のフォルダがあり、整理、名前変更、管理が容易になります。

シーンはアルファベット順にロードされるため、名前を変更することで順序を変更できます。

各シーンフォルダの中には以下のファイルが含まれます。

- **プレビュー用のシーンのサムネイル画像**
- **すべてのシーンパラメータ（モード、ノブの位置、設定）を保存する scene.json ファイル**
- **シーンにノブのシーケンスが保存されている場合、記録されたノブの動きを含むオプションの knob_sequence.json ファイル**

このフォルダベースのシステムにより、シーンをバックアップ、編集、共有しながら、関連データをすべて一緒に管理しやすくなっています。

最後に、/sdcard または /usbdrive にあるその他のフォルダやファイルは、EYESY の通常の動作では無視されることに注意してください。これにより、「ライブ」バージョンを編集しながら作業中のモードのバックアップフォルダを保持することができます。

4.3 モードの操作

モードを扱う際は、各フォルダ内のファイルを個別に管理するのではなく、フォルダ単位で作業するのが一般的です。例えば、以下のような場合があります。

- **外部ソースから新しいモードを EYESY に追加する場合**、アップロード前にフォルダ全体を ZIP 圧縮してください。アップロード後、チェックボックスを選択し、**Unzip** コマンドを使用します。
- **モードをコンピューターにバックアップする場合**、フォルダを ZIP 圧縮してコピーしてください。
- **モードの名前を変更する場合**、フォルダ名を変更してください（**main.py** はそのままにしておきます）。
- **モードを複製して新しいコピーを編集する場合**、モードフォルダをコピー＆ペーストし、異なる名前を付けます。
- **モードを削除する場合**、フォルダ全体を削除してください。

モードのコードを表示するには、**ファイルブラウザ** で編集したいファイル（例えば main.py）を選択します。編集後は、**Control パネル** の **Save** や **Reload** コマンド（場合によっては **Start Video** や **Stop Video**）を使用します。

コードエディターでスクリプトを変更したら、**Save** をクリックして変更を保存し、その後 **Reload** をクリックしてディスプレイ上のビジュアルを更新します。

構文エラーやランタイムエラーが発生した場合、**コンソール** や **OSD** を有効にしている場合は**ディスプレイ上**に表示されます。

モードに関する補足事項

- 上記のモードフォルダには、すべて **main.py** ファイルが含まれています。
 - **EYESY モードの最低限の要件を満たしている限り、これらの Python スクリプトは有効なモードとして認識され、デバイス起動時に読み込まれます。**
 - （モードのプログラミングや編集に関する詳細は **第 4 章** を参照してください。）
- **main.pyc ファイルが含まれている場合**、これは EYESY の Python コンパイラによって自動生成された中間ファイルです。
 - これらのファイルはスクリプトの読み込み速度を向上させますが、**必須ではありません。**

- モードに必要なその他のファイル（画像、追加の Python スクリプト、フォントファイルなど）は、モードフォルダ内に保存する必要があります。
 - モードフォルダにはサブフォルダを含めることが可能であり、サブフォルダ内にあるモードも認識されます。
-

次の章では、EYESY のプログラミングについて詳しく説明します！

5. EYESY のプログラミング

さて、ここまで来ました。夢や EYESY のモードが生まれる場所です。ファイル管理について学んだところで、プログラミングを始めましょう！

まず最初に、この章はあなたに向いていないかもしれません。正直に言うと、EYESY はプログラミングをしなくても十分に機能的で表現力豊かなデバイスです。もし EYESY の使用に慣れており、オーディオやビジュアルの直感的な操作を楽しみたいのであれば、それで問題ありません。最も役立つ方法に集中し、自由を楽しんでください。そして、もし再びここに戻りたくなったら、いつでもお待ちしております。

この章では、**USB-WiFi アダプター**を使用して、**コンピューター上で EYESY エディターを開けること**を前提としています。もし設定がまだであれば、**前章（第 4 章）** を参照してください。

この章では、EYESY の動作と、そのフレームワーク内で利用できる方法に焦点を当てますが、**プログラミングの基礎そのものは教えません**。しかし、心配はいりません。今では、プログラミングの基本や特定の言語（Python など）を学ぶためのリソースが豊富にあります。インターネット検索、図書館の書籍、地域のプログラミング教室などを活用すれば、学習の手助けとなる情報が見つかるでしょう。

もしすでにプログラミングを楽しんでいる、またはこれから挑戦しようとしているなら、または映像表現のアイデアを実現したいと考えているなら、さっそく始めましょう。エンジンのフードを開ける準備をしてください。

この章では、まず EYESY モードの概念と基本的な要件について説明します。各モードに必要なものは多くありませんが、**EYESY の動作原理を理解する** ために、何が必要なかを明確にしていきます。その後、**EYESY のアプリケーションプログラムインターフェース (API)** の概要を説明します。必要最低限の機能を超えて、どのような関数やプロパティを使用できるのかを紹介します。

さあ、これが私たちの進む道です。すでに EYESY の操作方法は理解しましたね。それでは、エンジンオイルを交換して、ドライブに出かけましょう…！

5.1 モードスクリプトの動作

EYESY がどのように動作するのかを理解すると、その仕組みがよく分かるようになります。ここでは、使用されているプログラミング言語や EYESY のシステム動作について簡単に説明した後、モードの基本要件について解説します。

Python、Pygame、そして EYESY

以前にも説明したように、EYESY のモードは **Python** で記述されており、描画処理には **Pygame** のグラフィックライブラリを使用しています。つまり、EYESY のプログラミング言語は **Python** であり、ビジュアル生成を担当するのが **Pygame** というわけです。

Python には豊富な標準ライブラリがあり、それらを EYESY のモード内で使用することができます。利用可能なモジュールの一覧は **Python の公式ドキュメント** に掲載されています。EYESY OS v3 では **Python 3.11** を使用しています。工場出荷時のモードを調べると、glob、imp、math、random、time などのモジュールが使われていることが分かります。これらのライブラリの詳細や Python 自体の使い方について学びたい場合は、公式ドキュメントを参照するとよいでしょう。

Pygame は、もともとビデオゲーム開発のために作られた Python のモジュール群です。EYESY はこの Pygame のグラフィックライブラリを活用してビジュアルを生成します。利用可能なモジュールの一覧は、**Pygame の公式ドキュメント** の「Reference」セクションに掲載されています。一部の工場出荷時のモードでは、pygame.freetype や pygame.gfxdraw などの追加ライブラリも使用されていますが、他のグラフィック関連のモジュールも利用可能です。

Pygame は、描画結果を **サーフェス (surface)** というオブジェクトに出力します。このサーフェスが、Python スクリプトの出力を Pygame に接続する仕組みになっています。この詳細については、次のセクションで説明します。

EYESY のモード読み込み動作

EYESY が起動すると、microSD カード内の **Modes フォルダ** にある **すべての有効なモード** を識別し、それらを同時に読み込みます。この並列処理のおかげで、**モードの切り替えが瞬時に行え、スムーズな操作が可能** になります。

しかし、この並列動作には **考慮すべき点** がいくつかあります。

- **メモリを大量に消費するモードは常に実行されている。**
setup() 内で大量の画像を読み込むようなモードを使用すると、大量の RAM を消費する可能性があります。このメモリは、別のモードに切り替えても解放されません。そのため、**EYESY 全体の動作が遅くなる場合は、メモリを過剰に使用しているモードが原因** である可能性があります。

対策として、テスト時には不要なモードを Modes フォルダから移動し、少数のモードのみをロードするようにしてください。

- **サポートファイルは最小限に抑える。**

画像を読み込むモードの例を考えると、**モードフォルダ内の画像サイズを 10MB 以下に抑えることを推奨**します。実際の使用状況によって適切なサイズは変わりますが、一般的な目安として考えてください。

また、カメラやスマートフォンで撮影した写真をそのまま使用する場合は、EYESY の **HDMI デフォルト出力解像度 (1280×720) 以下に縮小**すると、パフォーマンスの向上が期待できません。

すべてのケースに当てはまるわけではありませんが、基本的な指針として考えてください。

モードの最低要件

各モードが正常に読み込まれ、グラフィック出力を行うためには、以下の 4 つの基本要件を満たしている必要があります。

1. 「pygame」モジュールを読み込む。

- Python の他のモジュールと同様に、import 文を使用して Pygame を読み込む必要があります。
- **すべての EYESY モードスクリプトは以下のコードで始まるべきです。**
- `import pygame`

2. 「setup」関数を持つ。

- `setup()` 関数は、モードが読み込まれたときに自動的に呼び出されます。
- これは、**ビデオレンダリングの前に 1 回だけ実行する処理を記述するのに適した場所**です。

3. 「draw」関数を持つ。

- `draw()` 関数は、**各フレームごとに 1 回実行され、画面を描画する処理を担当します。**
- ここでは、以下のような処理が一般的に行われます。
 - **前のフレームからの制御パラメータの変更を反映**
 - **オーディオ情報を更新して再描画**
 - **トリガーイベントやその他のメッセージを処理**
- **この関数はフレームごとに繰り返し実行されるため、不要な処理を避け、軽量のコードにすることが重要です。**

4. 「screen」サーフェスに描画する。

- EYESY は **screen** という Pygame の予約済みサーフェスを作成します。
- これは HDMI ポート経由で出力されるため、描画する際は **screen** に対して描画処理を行う必要があります。

シンプルなモードの例

これらの要件を考慮した、最も基本的な EYESY モードの例を示します。

```
import pygame

def setup(screen, eyesy):
    pass

def draw(screen, eyesy):
    size = 640
    position = (510, 500)
    color = (255, 0, 0)
    pygame.draw.circle(screen, color, position, size, 0)
```

この main.py の動作を解説

1. **pygame モジュールのインポート**
 - import pygame により、Pygame の機能を使用可能にします。
2. **setup() 関数の定義**
 - setup(screen, eyesy) 関数を定義します。
 - この例では特に処理を記述していませんが、必要に応じて初期設定を行う場所です。
3. **draw() 関数の定義**
 - draw(screen, eyesy) 関数は、毎フレーム呼び出され、画面に描画を行います。
 - ローカル変数を定義し、pygame.draw.circle() を使用して **赤い円を描画** します。
 - screen を第一引数に指定することで、EYESY の HDMI 出力へ描画を送ります。

このモードの動作結果

- 画面中央付近（座標 510,500）に半径 640px の赤い円が描画される。
- この円は毎フレーム描かれ続け、更新される。

このシンプルなコードが、EYESY の工場出荷時のモードを調査したり、自分でモードを作成するための基本的な枠組みになります。

最後に 1 つだけアドバイス:

- 単純な円の描画だけではなく、時々異なる画像やアニメーションを追加すると良いでしょう。

5.2 EYESY の API

EYESY のモードの基本構造と要件について説明したところで、今度は EYESY で利用できる **API (アプリケーションプログラムインターフェース)** について見ていきます。

EYESY には **eyesy オブジェクト** という特定の変数群があり、すべてのモードからアクセス可能です。

eyesy オブジェクトの変数一覧

- **eyesy.audio_in**
 - EYESY のオーディオ入力チャンネルが記録した直近 100 個の音量レベルのリスト。
 - 左右の入力チャンネルは統合され、**モノラルの信号** として処理される。
 - 100 個のオーディオ値は、**16 ビットの符号付き整数** (最小 -32,768 から最大 +32,767) で格納される。
 - トリガーボタンを押すと、このリストに正弦波 (サイン波) が追加され、EYESY にオーディオ入力をシミュレートできる。
- **eyesy.trig**
 - トリガーイベントが発生しているかを示すブール値 (True/False)。
 - オーディオをトリガーソースとして選択している場合、音量が固定されたスレッシュホルド (最大値の約 80%) を超えた際に True となる。
 - トリガーボタンを押すと、自動的に `eyesy.trig = True` になる。
- **eyesy.xres**
 - 現在の出力解像度の横方向 (幅) の値 (float)。
- **eyesy.yres**
 - 現在の出力解像度の縦方向 (高さ) の値 (float)。
- **eyesy.knob1 ~ eyesy.knob5**
 - 現在の 1~5 番目のノブの値を示す float (0.0 ~ 1.0)。
 - MIDI コントロールチェンジ (CC) メッセージを受信すると、その値がノブの値として上書きされる。

- 対応する MIDI CC メッセージ:
 - Knob 1 → MIDI CC #21
 - Knob 2 → MIDI CC #22
 - Knob 3 → MIDI CC #23
 - Knob 4 → MIDI CC #24
 - Knob 5 → MIDI CC #25
 - ノブを手動で回すと、MIDI で上書きされた値はリセットされ、再び物理ノブの値が適用される。
 - eyesy.midi_notes
 - MIDI の 128 音階（ノートピッチ）のリスト。
 - 各値は、そのノートが現在オン（押されている）かオフ（押されていない）かを示すブール値（True/False）。
 - 例：
 - MIDI ノート 60（C4 = ミドル C）が押されているときに特定の処理を実行するコード。
 - `if eyesy.midi_notes[60]:`
 - `yourFunctionHere()`
-

このように、EYESY の API を利用することで、オーディオ信号の取得、MIDI ノートの検出、ノブの動作制御など、柔軟なモードを作成できます。

eyesy オブジェクトの追加変数と関数

追加の変数

- eyesy.midi_note_new
 - 新しい MIDI ノートオン メッセージが、前回の draw() 関数の実行以降に受信されたかを示すブール値（True/False）。
 - 少なくとも 1 つの新しいノートオンメッセージが受信されると True になる。
- eyesy.mode
 - 現在のモードの名前を示す文字列。
 - 例えば "T - Density Cloud" など、モード名がそのまま格納される。
- eyesy.mode_root
 - 現在のモードが保存されているフォルダのパスを示す文字列。
 - 例："/sdcard/Modes/Python/CurrentModeFolder"
 - 画像、フォント、その他のリソースをモードのフォルダから読み込む際に便利。

- `setup()` 関数内で使用するのが適切。
-

EYESY の重要な関数

- `eyesy.color_picker_bg()`
 - 背景色を設定する関数。
 - 通常、"`eyesy.color_picker_bg(eyesy.knob5)`" のように指定し、ノブ 5 を背景色のコントローラーとして使用する。
 - **0~1 のノブ値を RGB 値に変換し、背景色として適用。**
 - 他のノブ (`knob1~knob4`) を指定して制御することも可能。
- `eyesy.color_picker()`
 - 指定したノブの値を色 (RGB) に変換する関数。
 - 通常、"`eyesy.color_picker(eyesy.knob4)`" のように指定し、ノブ 4 を使用する。
 - 呼び出すと、**(赤、緑、青) の 3 つの整数値のタプルを返す。**
 - 工場出荷時のモードでは、以下のようにローカル変数 `color` にセットされることが多い。
 - `color = eyesy.color_picker(eyesy.knob4)`
- `eyesy.color_picker_lfo()`
 - `eyesy.color_picker()` に **LFO (低周波オシレーター)** を加え、色を自動で変化させる関数。
 - ノブの左半分の回転 (最大左~中央) は **固定色の選択**、右半分の回転 (中央~最大右) は **LFO の速度調整** に割り当てられる。
 - 通常、"`eyesy.color_picker_lfo(eyesy.knob4)`" のように指定し、ノブ 4 を使用するが、他のノブを使うことも可能。
 - LFO が有効になると、色がゆるやかに変化し、急激な色のジャンプを防ぐ。
 - 呼び出すと、(R, G, B) の整数タプルを返す。
 - 工場出荷時のモードでは、以下のようにローカル変数 `color` にセットされることが多い。
 - `color = eyesy.color_picker_lfo(eyesy.knob4)`
 - **オプション引数として最大 LFO 速度を指定可能。**
 - 指定がない場合、デフォルトの LFO 速度は **0.1** に設定される。
 - 例：最大 LFO 速度を 1.1 に設定する場合
 - `color = eyesy.color_picker_lfo(eyesy.knob4, 1.1)`

このように、EYESY の API を活用することで、背景色の変更、ノブに基づいた色の変更、LFO を使った色のアニメーションなど、さまざまなビジュアル表現が可能になります。

```
color = eyesy.color_picker_lfo(eyesy.knob4, 1.1)
```

実際の LFO の速度は、画面に描画される要素の数やサイズなどによって、モードごとに異なる場合があります。

3 つの `eyesy.color_picker...` 関数の目的は、特定のノブを使用して要素の色を簡単に設定できるようにすることです。

モードをプログラムする際、カラー ピッカーと背景色選択に同じノブを使用することを検討してください。

これにより、同じマッピングを使用してモードを切り替えた際に、前景色と背景色の一貫性が保たれ、スムーズなトランジションが可能になります。

これは、EYESY のモードエディター/プログラマーとして利用可能な API コンポーネントを示しています。

ハードウェア コントローラーに基づいた変数の値は、シーンを変更すると一時的に上書きされる場合があります。ことに注意してください（詳細は第 2 章を参照）。

また、`draw()` 関数は各ビデオ フレームごとに呼び出されるため、`draw()` 内で変数を適切に参照することで、モードの出力を動的に更新できます。

最後に、私たちはモードの作り方を指示するわけではありませんし、何が良いアイデアなのかを決めることもありません。

スコープモードとトリガーモードという分類さえも、完全に適用できるわけではありません。

ここからは、あなた自身のアイデア次第です。

Python の知識を活かし、自分なりのアイデアを試し、新しいモードを作ってみてください。

最初から完璧にできなくても、試行錯誤の中で思わぬ発見があるかもしれません。

アートとは、作っていく過程で生まれるものなのです。

6. 付録¶

6.1 SD カードディスクイメージの書き込み¶

microSD カードには、EYESY のオペレーティングシステムも保存されています。

microSD カードに新しいディスクイメージを書き込むことで、EYESY を工場出荷時の状態にリセットできます。これには、OS 用のパーティションとモード保存用のパーティションが含まれています。この作業は、最新の EYESY OS に更新する場合や、microSD カードの問題を修正する際に役立ちます。

この操作を行うと、microSD カードのデータが完全に消去されるため、必要なデータを事前にバックアップしてください。モードやスクリーンショット、その他のファイルをダウンロードする方法については第 4 章を参照してください。また、古い OS を保持したい場合は、新しい microSD カードを使用することもできます。EYESY に付属していたものと同じ仕様のカードを使用する場合、推奨スベックは以下の通りです。

SanDisk Edge 8GB microSDHC C4 UHS-I

新しい SD カードに書き込む手順：

1. microSD カード用のディスクイメージをコンピュータにダウンロード。
 - 最新の OS リリース：EYESY-3.0（8GB 以上の microSD カードが必要）
2. フラッシュャープログラムをコンピュータにダウンロード。
 - [Balena Etcher](<https://www.balena.io/etcher/>)
3. EYESY の電源を切る。
4. 本体背面のスリット（MIDI In ポートと HDMI ポートの間）を探す。
5. ピンまたはクリップを使って microSD カードを押し込み、取り出す（バネで軽く飛び出す）。
6. microSD カードをコンピュータに挿入（アダプターやカードリーダーが必要な場合あり）。
7. Etcher プログラムを使い、解凍した OS を microSD カードに書き込む。
 - Etcher の処理が完了すると、コンピュータが「このディスクは読み取れません」と表示する場合がありますが、これは正常です。「取り出す」をクリックして進めてください。
8. microSD カードをコンピュータから取り外し、EYESY に再挿入する。
 - SD カードが基板のソケットに正しく挿入されていることを確認する（本体内部に落とし込まないよう注意）。
 - SD カードがグラつく場合は、ソケットに正しく入っていない可能性がある。

- ピンやクリップを使い、カチッと音がするまで押し込む。

9. EYESY を再起動する。

6.2 EYESY の映像をコンピュータに取り込む¶

素晴らしいビジュアルを作成できるようになったら、それを動画として記録したり、ライブ配信したり、VJ ソフトウェアにルーティングしたりしたくなるかもしれません。これらの操作を行うには、EYESY の映像出力をキャプチャする必要があります。しかし、ほとんどのコンピュータにはビデオ入力端子がないため、この作業には「ビデオキャプチャカード」が必要になります。

キャプチャカードを選択する前に、以下のポイントを考慮してください。

- ****ビデオ入力:**** 一部のキャプチャカードは HDMI のみを受信します。他のカードは HDMI とコンポジットビデオの両方を受信できるものもあります（ただし、一度に使用できるのはどちらか一方）。
- ****ソフトウェアの互換性:**** 映像信号が他のプログラムで使用できるかどうかは、キャプチャデバイスによります。例えば、Elgato のキャプチャカードは Elgato 専用ソフトウェアでのみ録画可能ですが、Blackmagic のキャプチャカードはサードパーティ製ソフトウェアでも使用できます。これにより、EYESY の映像を Adobe Premiere や QuickTime などで録画したり、OBS でストリーミングしたり、VJ ソフトウェアで他の映像と組み合わせることができます。

もう 1 つ考慮すべき点として、多くのキャプチャカードは音声入力と映像入力の両方を備えています。EYESY にはオーディオ出力がありません。そのため、キャプチャカードで音声も録音する場合は、別途オーディオラインを接続する必要があります。

一般的なビデオキャプチャの手順

1. キャプチャカードをコンピュータに接続する。
2. EYESY の Video Out をキャプチャカードの入力に接続する（この時点では電源を入れない）。
3. ****（オプション）**** 音声をキャプチャする場合は、オーディオ出力をキャプチャカードに接続する。
4. キャプチャカードに対応したビデオソフトウェアを開く。
5. ソフトウェアの「キャプチャ設定」画面で、キャプチャカードからの映像・音声を受信するように設定する。
6. EYESY の電源を入れる。
7. 映像ソフトウェアに EYESY の映像出力が表示されていることを確認する。
8. 動画を録画したり、ストリーミングしたり、VJ ソフトウェアで他の映像とミックスする。