

## DigiShow LINK

DigiShow is a lightweight cross-media control software designed for live performances and interactive art installations. It provides an intuitive console interface that enables signal control and mapping among various audio, lighting, robotic, and interactive devices.

|  |   |
|--|---|
| <b>Digital Artists</b>                 | Create cross-media interactive art installations                                |
| <b>Stage &amp; Immersive Designers</b> | Program cues to synchronize lighting, mechanical devices, and music             |
| <b>Musicians</b>                       | Enhance live performances with DIY electronic instruments or automated lighting |
| <b>Engineers &amp; Makers</b>          | Develop smart interactive projects like home automations                        |

## Install and Run

### macOS

Copy app "DigiShow" to your Applications folder and run it.

If you see the error message says **DigiShow app is damaged and can't be opened**, please also need to run this command in the terminal before starting the app for the first time.

```
xattr -cr /Applications/DigiShow.app
```

### Windows

Copy folder "DigiShow LINK" to your disk and run "DigiShow.exe" in the folder.

If you see the error message says **The code execution cannot proceed because MSVCP140.dll was not found**, please also need to run **Extra\vc\_redist.x64.exe** to install Visual C++ Redistributable to your windows system.

It's also recommended to install **K-Lite Codec Pack** and **loopMIDI** to your Windows, the installer files can be found in Extra folder.

## Key Features

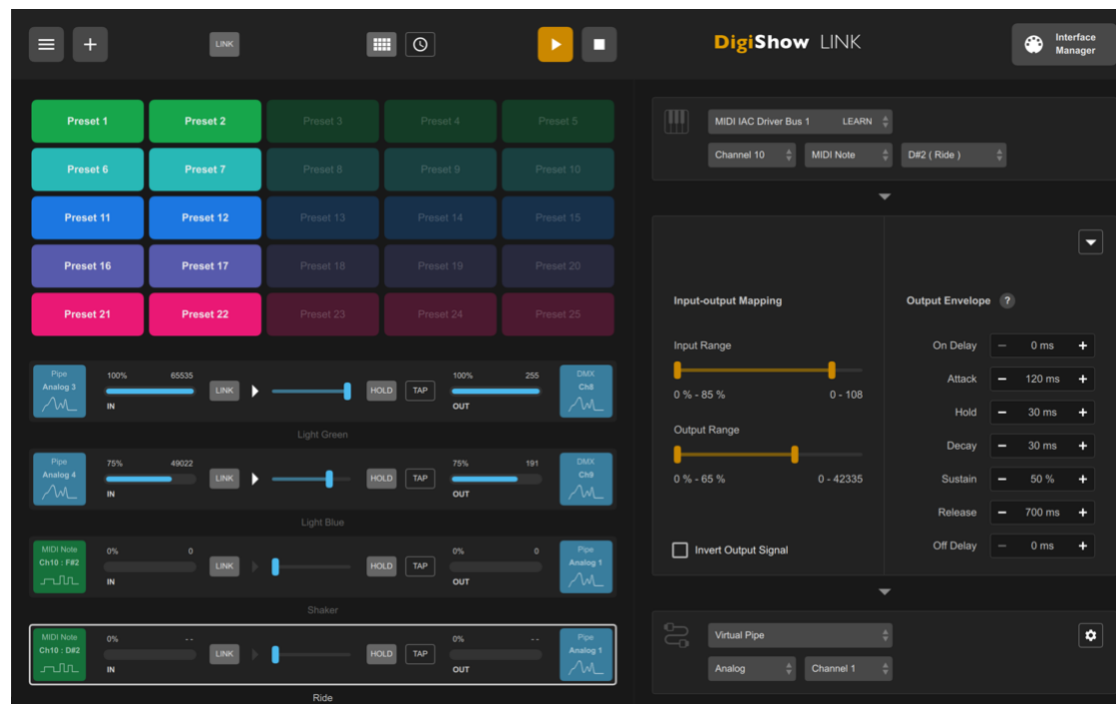
- **Multi-Protocol Support**

Supports MIDI, DMX, OSC, Modbus, Arduino, Hue, and other protocols for coordinated control of audio, lighting, screens, robotics and more.

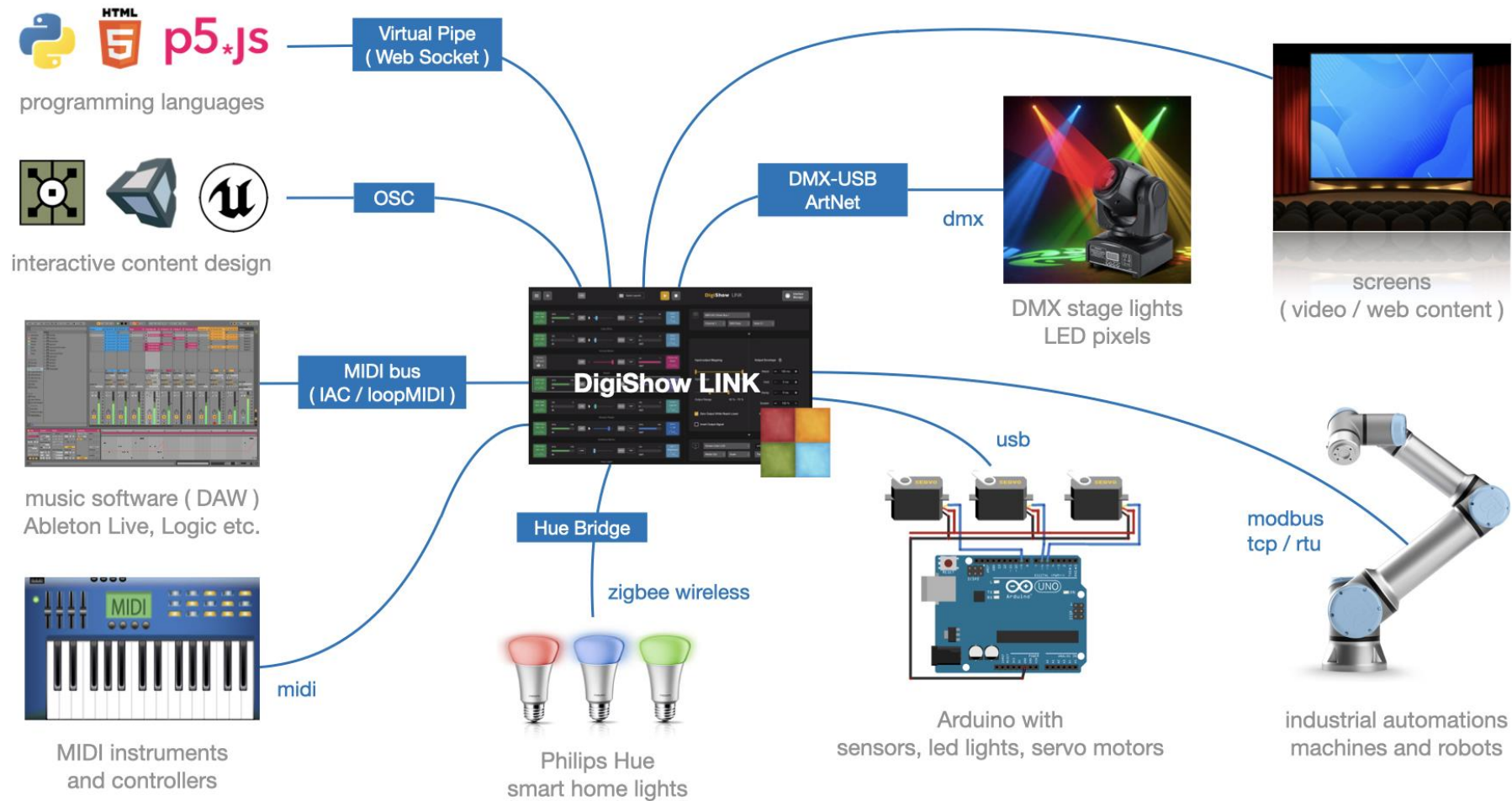
- **Signal Mapping**

Converts MIDI notes, OSC control signals, etc., into lighting, servo motor, and media playback commands. Also transforms sensor inputs into MIDI/OSC signals for music software (e.g., Ableton Live, Logic Pro) and real-time visual creative tools (e.g., TouchDesigner, Unreal Engine).

- **Interactive Control** Ideal for DJ performances, stage or space lighting synchronization, experimental music, and interactive installations to enhance live engagement and visual effects.



*DigiShow main window:  
Preset launcher, signal link table and  
signal mapping settings.*



For a typical 'digital' show, requires some particular digital things working together, along with DigiShow LINK.

## Technologies and Functions

- **No-Coding Arduino**

Provides a ready-to-use sketch drives mapping sensors and actuators connected to Arduino as inputs and outputs used directly in the DigiShow.

- **Cue Player**

Features a graphical interface for designing output curves for specific scenarios, which can be attached to presets for synchronized playback.

- **Pixel Mapping**

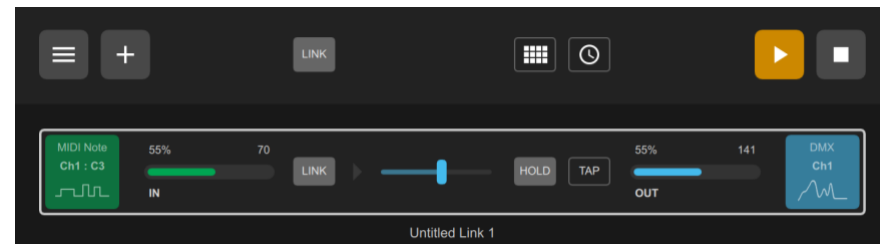
Dynamically maps video or image pixels to lighting arrays for visual effects.

- **Scripting Support**

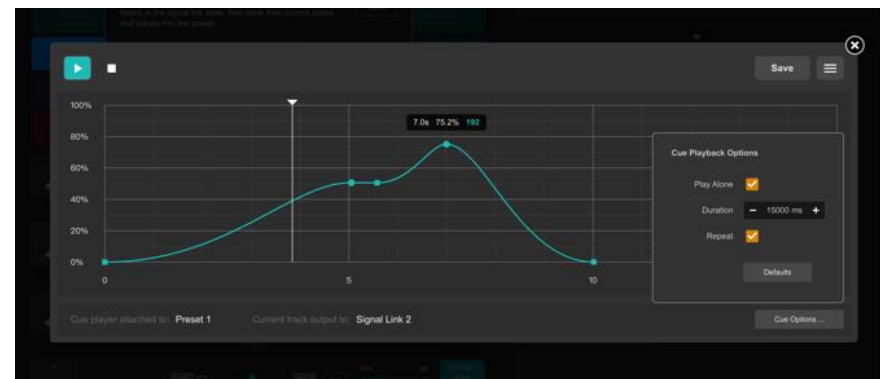
Allows JavaScript/QML expressions and scripts for advanced interactive logic.

- **Network Protocols**

Supports OSC and pipes over WebSocket for external extensions, as well as Art-Net for large-scale lighting control systems.



*A signal bar linking a MIDI note to a DMX lighting channel.*



*Timeline editor for Cue Player: Design the output signal playback curve in the timeline.*



*Preset launch buttons in the DigiShow app and the mobile.*

## DigiShow for Beginners

If you are using DigiShow for the first time, after successfully installing the software on your computer, we can try the following operations to learn and understand DigiShow's features.


### 1. Configure Interfaces

Set up control interfaces (e.g., MIDI, DMX, Arduino) for audio, lighting, and interactive devices in the Interface Manager.

- Click the "Interface Manager" button in the top-right corner of the main window, select "Screen", click "+" to add a screen presentation interface, and set it as "Preview Window" type. Close the "Interface Manager" dialog when done.
- In the following experiment, we won't need to connect physical lighting fixtures. Instead, we'll use a screen window to simulate an RGB full-color dimming light.



### 2. Add Signal Links

Add signal bars in the Signal Link Table and assign specific outputs to them, where adjust output values in real time using faders.

- Click the "+" button in the top-left corner of the main window three times to create three blank signal bars in the Signal Link Table. Select the first signal bar, and in the bottom-right of the main window, set "Output Destination" to "Screen (Preview Window)" + "Backlight" + "Red", then click the "Apply" button.
- Continue selecting the second and third signal bars in the Signal Link Table and set their "Output Destination" to "Screen (Preview Window)" + "Backlight". Change the color options to "Green" and "Blue" respectively, then click the "Apply" button.
- Click  the play button at the top of the main window to start our first DigiShow project. You'll see a window with a black background appear on the screen. Move it to the side for observation.
- Drag the three faders on the signal bars to change the color and brightness of the preview window. The three faders correspond to the red, green, and blue color channels respectively.

### 3. Save Scenes

Save output values in the grouped signal bars as Presets for specific scenarios. Store timeline-based signal output curves in the Cue Player.

- Click  the grid button at the top of the main window, and a Preset Launcher panel with many buttons will appear. Right-click the "Preset 1" button and select "Create Preset" from the popup context menu.
- At this point, you'll notice a "+CUE" button  appears below the fader in each signal bar. Click this button to display the timeline editing window of the Cue Player. Click on the lines in the chart to add polyline control points (right-click for Bezier curves), and drag the control points to design an output variation curve for the current signal bar. When finished, click the "Save" button and close the timeline editing window.
- Complete the output curve design for the red, green, and blue signal bars using this method, then click the "Save Preset" button. After this, clicking the "Preset 1" button will start the Cue Player to execute the RGB dynamic lighting program we just designed.
- After completing "Preset 1" design, we can continue designing "Preset 2", "Preset 3"... using the method described above. Each preset contains a different set of signal output curves, corresponding to different dynamic lighting programs. If you want the launched program content to play in a loop, click "Cue Options" in the bottom-right corner of the timeline editing window and check "Repeat".

### 4. Switch Scenes (Manual)

Activate the saved Presets and Cue Players by tapping the buttons in the Preset Launcher or a remote web page in another smart phone.

- You can click buttons in the Preset Launcher at any time to start various pre-designed scenes.
- During performance, you might want to launch presets through your phone or tablet. Right-click any preset button, select "Remote Control" from the popup context menu, and click the "Start" button. Take out your phone and scan the QR code on the screen. Make sure your phone is using the same local network WiFi as your computer, and a page showing the same preset buttons will appear on your phone. Pressing buttons on your phone will have the same effect as on your computer.

### 5. Trigger Scenes (Automatic)

Alternatively, connects scene switching to specific sensor triggers, by selecting the sensors as inputs and Presets as outputs in the signal bars.

- We usually use Arduino to connect various sensors, but in this experiment for beginners, we won't connect any external hardware. Instead, we'll use keyboard keys to simulate external signal triggers. Please click the "+" button in the top-left corner of the main window to create another blank signal bar in the Signal Link Table. In the top-right corner of the main window, select "Input Source" as "Hot Key" + "(none)" + "(none)" + "A", then click the "Apply" button. This way, the signal bar can receive input signals when users press the A key on the keyboard.
- Then, in the bottom-right corner of the main window, select "Output Destination" as "Preset Launcher" + "Preset 1", and click the "Apply" button. This completes a signal mapping binding on the signal bar. Pressing key "A" will trigger the launch of "Preset 1".
- Once the signal mapping between key "A" and "Preset 1" is established, you can also create more key-to-preset mappings following the same procedure. This way, pressing specific keys will trigger specific presets to launch various scenes.

### **Complete**

Thus, a live performance lighting console or an interactive art installation is ready.

After completing this experiment, you can continue to explore more signal types and conversion parameters during signal mapping. For example, you can try setting the signal bar's output end to "Audio Player" and adding a synchronized sound to your scene.

For more learning, please read DigiShow Tutorials

<https://github.com/robinz-labs/digishow/blob/master/guides/tutorials.md>

## Release Downloads

Please visit <https://github.com/robinz-labs/digishow/releases/latest> to download the latest releases:

- |   |                                |
|---|--------------------------------|
| - DigiShow LINK for windows (64bit Intel)             | digishow_win_x.x.x_x64.zip     |
| - DigiShow LINK for macOS (64bit Intel)               | digishow_mac_x.x.x_x64.zip     |
| - DigiShow LINK for macOS (64bit Apple M series)      | digishow_mac_x.x.x_arm64.zip   |
| - DigiShow LINK for Linux (Raspberry Pi or 64bit ARM) | digishow_linux_x.x.x_arm64.zip |

Go to the page, where choose the file in Assets list to download.

## Extra Downloads and Resources

### Arduino

DigiShow RIOCI library

Required to enable DigiShow LINK app to control sensors and actuators connected on your Arduino.

Find and install DigiShow RIOCI in the library manager of Arduino IDE, or download it from github.

<https://github.com/robinz-labs/rioc-arduino/releases>

### MIDI

Virtual MIDI bus drivers (IAC / loopMIDI)

In order to communicate with MIDI messages between DigiShow LINK and other software, users just need to setup a virtual MIDI bus in the operation system.

IAC for Mac

<https://help.ableton.com/hc/en-us/articles/209774225-How-to-setup-a-virtual-MIDI-bus>

loopMIDI for Windows

<http://www.tobias-erichsen.de/software/loopmidi.html>



|               |  |
|---------------|--|
| <b>DMX</b>    | <p>ENTTEC DMX USB Pro driver (FTDI VCP)</p> <p>Required to enable DigiShow LINK to control DMX lightings through an ENTTEC adapter.</p> <p><a href="https://www.ftdichip.com/Drivers/VCP.htm">https://www.ftdichip.com/Drivers/VCP.htm</a></p>   |
| <b>Screen</b> | <p>K-Lite Codec Pack (for windows)</p> <p>Required to enable DigiShow LINK to play MP4, MOV video files on your Windows computer.</p> <p><a href="https://www.codecguide.com/download_kl.htm">https://www.codecguide.com/download_kl.htm</a></p> |

## Developer Resources

DigiShow is open-source.

If you would like to rebuild this software using the source code we contributed, please visit <https://github.com/robinz-labs/digishow> .

Building executables from source using the qmake tool or the QtCreator application requires Qt 5.12 or 5.15 LTS.

Additional library dependencies are already included in the repository:

- RtMidi 4.0.0 <http://www.music.mcgill.ca/~gary/rtmidi/>
- TinyOSC library <https://github.com/mhroth/tinyosc>
- Ableton Link library <https://ableton.github.io/link/>
- global hotkey library <https://github.com/Skycoder42/QHotkey>

The source code can be compiled for target platforms compatible with:

- macOS 10.13 or higher
- windows 10 or windows 11 ( 64-bit version )
- linux ( see the websites of Qt and other dependent libraries for compatibility details )