

User Manual



KioscTouch2

Customizable wall-mount touchscreen

Revision History

Revision	Date	Author(s)	Description
1	06.07.2026	ME	Initial version

Contents

1	Safety Precautions	4
2	Introduction	5
3	Installation	9
4	Network Configuration	11
5	Kiosk App	13
6	Web Interface	32
7	PurpleCloud	40
8	KioskEditor	49
9	File Locations	65
10	vManager	68
	Appendices	72
A	Elements	73
B	API	96
C	Certifications	98

© 2026 Visual Productions BV. All rights reserved.

No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Due to the dynamic nature of product design, the information contained in this document is subject to change without notice. Revisions of this information or new editions may be issued to incorporate such changes.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

Art-Net™ Designed by and Copyright Artistic Licence.

The Visual Productions logo and the term "CueCore" are registered trademarks of Visual Productions BV.

Chapter 1

Safety Precautions

This chapter contains important safety information for the installation, operation, and maintenance of the KioskTouch2. Read these precautions carefully before using the device to ensure safe operation and to prevent damage to equipment or injury to personnel.

Before installing the KioskTouch2, read this manual completely and ensure that all installation personnel are familiar with the product and applicable safety requirements. Verify that the installation environment is suitable for the device and within the specified operating conditions. Disconnect power before making any wiring connections. Do not install or operate the product if it appears damaged. Ensure adequate ventilation around the unit and prevent exposure to moisture, excessive heat, or mechanical stress.

1.1 Indoor Use Only

The KioskTouch2 is intended for indoor use only. Do not install or operate the device in environments exposed to rain, moisture, condensation, excessive dust, or direct sunlight. Operating the product outside its specified environmental conditions may result in malfunction, damage, or reduced product lifetime.

1.2 No User-Serviceable Parts



The KioskTouch2 contains no user-serviceable parts. Do not attempt to open, disassemble, or repair the device. Opening the enclosure may expose sensitive electronic components and may void the warranty. Repairs must only be carried out by qualified service personnel or the manufacturer.

1.3 Product Modifications



Do not modify the KioskTouch2 or its accessories. Unauthorized modifications may impair product performance, compromise safety, invalidate regulatory compliance certifications, and void the warranty.

Chapter 2

Introduction

Thank you for choosing KioscTouch2.

KioscTouch2 is a compact, wall-mounted touchscreen controller designed for intuitive control of lighting, architectural, and AV installations. With its modern design, responsive touch interface, and seamless integration with the Visual Productions ecosystem, KioscTouch2 provides a simple and reliable user interface for both permanent and temporary installations.

Whether used in hospitality venues, museums, retail environments, conference facilities, or entertainment spaces, KioscTouch2 allows users to trigger scenes, control playback, adjust levels, and interact with connected systems through a fully customizable graphical interface.



Figure 2.1: KioscTouch2

Designed for Integration

KioscTouch2 is designed to work seamlessly with Visual Productions controllers and software platforms. Configuration is performed remotely, allowing system designers and integrators to create custom user interfaces tailored to the requirements of each installation.

The device communicates over Ethernet and supports Power over Ethernet (PoE), enabling both power and data connectivity through a single network cable for simple and efficient installation.

At the time of writing this manual the KioscTouch2 firmware is at version 1.20.0.

2.1 Compliance

This device is in compliance with the following regulations:

- CE
- UKCA

See the appendix for the associated certificates.

2.2 Key Features

- High-quality capacitive touchscreen
- Customizable user interface layouts
- Power over Ethernet (PoE) support
- Ethernet-based communication using OSC, UDP and TCP
- Compact wall-mounted design
- Cloud Connectivity
- Integration with Visual Productions control systems
- Compatible with third party control systems
- Suitable for architectural, entertainment, and commercial installations

2.3 Specifications

The KioscTouch2 has the following specifications.

- PoE Class 0
- 8 Inch full colour display
- Capacitive multitouch
- Resolution of 1280 x 800 pixels
- Wall mountable

- 460 gram (1 lbs)
- Operating temperature -20°C to +50°C (-4°F to 122°F)
- Compliance EN55103-1 EN55103-2

2.4 What's in the box?

The KioskTouch2 packaging contains the following items (see figure 2.2):

- KioskTouch2
- Wall-mount bracket
- 2m network cable
- Quick Start card



Figure 2.2: KioskTouch2 box contents

2.5 Unpacking

Carefully unpack the KioskTouch2 and inspect the contents for any signs of damage that may have occurred during transportation. Verify that all supplied accessories and documentation are present. If any damage is found, do not install or operate the device and contact your supplier immediately. Retain the original packaging materials whenever possible, as they provide the safest means for transporting or storing the product in the future.

2.6 Intended Disposal

At the end of its service life, this product must not be disposed of with household waste. The KioscTouch2 contains electronic components that should be collected and recycled in accordance with local regulations for Waste Electrical and Electronic Equipment (WEEE). Please dispose of the product through an authorized recycling facility or return it to an appropriate electronic waste collection point. Proper disposal helps conserve natural resources and prevents potential negative effects on the environment.

2.7 Further help

If you have further questions after reading this manual, please consult the online forum: <https://forum.visualproductions.nl>

Chapter 3

Installation

This chapter discusses how to physically set up the KioskTouch2.

3.1 Wall Mounting

Each KioskTouch2 comes with a metal backplate, which attaches to standard gang boxes (European and American). If no gang box is available then the KioskTouch2 can also be surface mounted. The KioskTouch2 itself is then held securely in place using strong magnets.

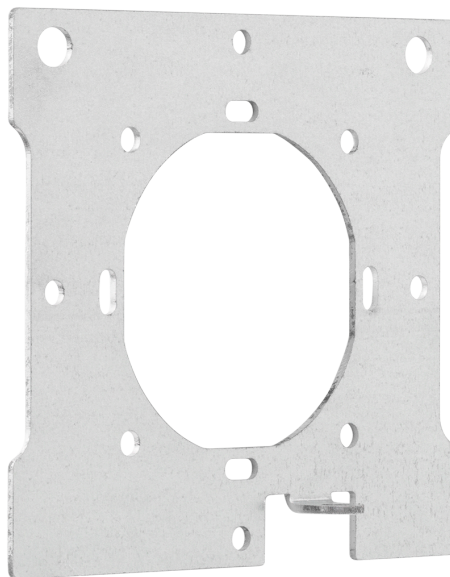


Figure 3.1: Wall mounting

A screw is provided to lock the KioskTouch2 to the backplate.

3.2 Power

Connect the unit to a PoE (Power over Ethernet) switch or injector using a standard Ethernet cable. The unit will power on automatically.

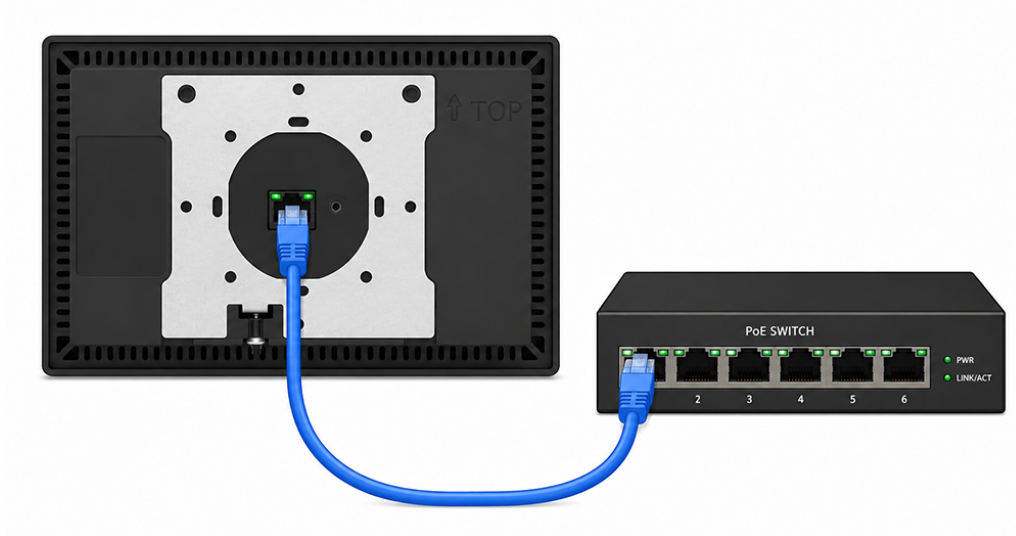


Figure 3.2: PoE powered

Power and data are provided over the Ethernet cable. It has a power consumption rating of PoE Class I.

Chapter 4

Network Configuration

This chapter explains the network configuration options available on the KioskTouch2. It describes IP addressing, DHCP operation, and the methods available for configuring or restoring network settings.

4.1 IP Address

The KioskTouch2 supports both static IP addressing and automatic IP addressing. By default, the KioskTouch2 is set to 'DHCP' in which it will be automatically assigned an IP address by the DHCP server in the network.

If the KioskTouch2 is set to DHCP mode and no DHCP server is present in the Network, it will automatically fall back to a random IP address in the range 169.254.xx.xx after a few seconds.

Static IP addresses are useful, for instance when there is a direct peer-to-peer connection between a KioskTouch2 and a computer. It is also useful in permanent installations where the IP address of the KioskTouch2 is known by other equipment and therefore should not change. When using DHCP there is often the risk of automatically being given a new IP address in the event that the IP lease time runs out. When using static IP addresses, make sure that all equipment on the network has a unique IP addresses within the same subnet.

The KioskTouch2's LED helps to determine which kind of IP address is set. The LED will indicate red when using DHCP and it will indicate white in the case of a static IP address.

There are three ways to change the IP address setting of the KioskTouch2.

- **vManager** can be used to detect a KioskTouch2 on the network. Once found, the vManager software (figure chapter 10) allows for changing the IP address, subnet mask and DHCP settings.
- If the IP address is already known then browsing to this address using the computer's browser will show the KioskTouch2's **web interface**. The *Settings* page of the web interface enables changing the same network related settings.

4.2 Network Connector LEDs

The KioskTouch2 has status LEDs integrated in the RJ45 network connector. These LEDs provide a quick indication of the Ethernet link, network activity and PoE power status.

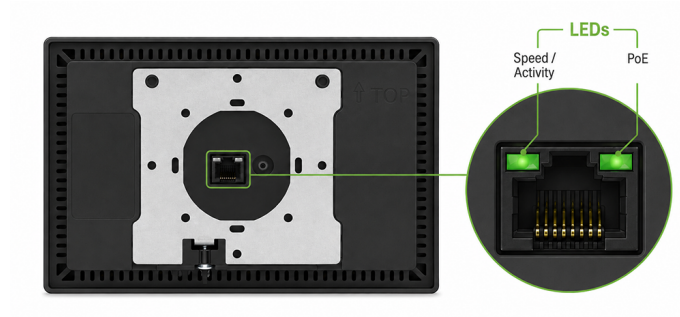


Figure 4.1: Network connector LEDs

The LEDs are located at the top-left and top-right of the RJ45 connector.

LED	State	Meaning
Left	Green on	1000 Mbit/s link
Left	Green blinking	1000 Mbit/s activity
Right	Green on	PoE power is present
Right	Off	No PoE power detected

Table 4.1: Network connector LEDs

Chapter 5

Kiosc App

The Kiosc app will present the customised dashboard to the user, typically running on a touch screen device. The Kiosc app itself has no editing capability in order to prevent the user from making accidental changes.

5.1 Modes

The Kiosc app operates in two modes, *Remote* and *File*.

5.1.1 Remote

The *Remote* mode serves two purposes. It allows you to quickly connect to a Visual Productions device and control it directly. Secondly, in this mode, Kiosc can connect to a KioscEditor and receive live updates as the user works on the design. This way it provides a real-time preview on the actual target device. Please note that due to Microsoft Store security measurements, it is not possible to make a remote connection to KioscEditor if Kiosc and KioscEditor are running on the same PC. To transfer a show-file, it can be manually copied using the file-browser. More information can be found in chapter 9.

Although quick to set up, a *Remote* connection to a CueCore™ or similar device does not enable the user to make elaborate designs. The editing capabilities inside the CueCore™ are basic. In order to create more demanding user-interface designs it is recommended to use the *File* mode.

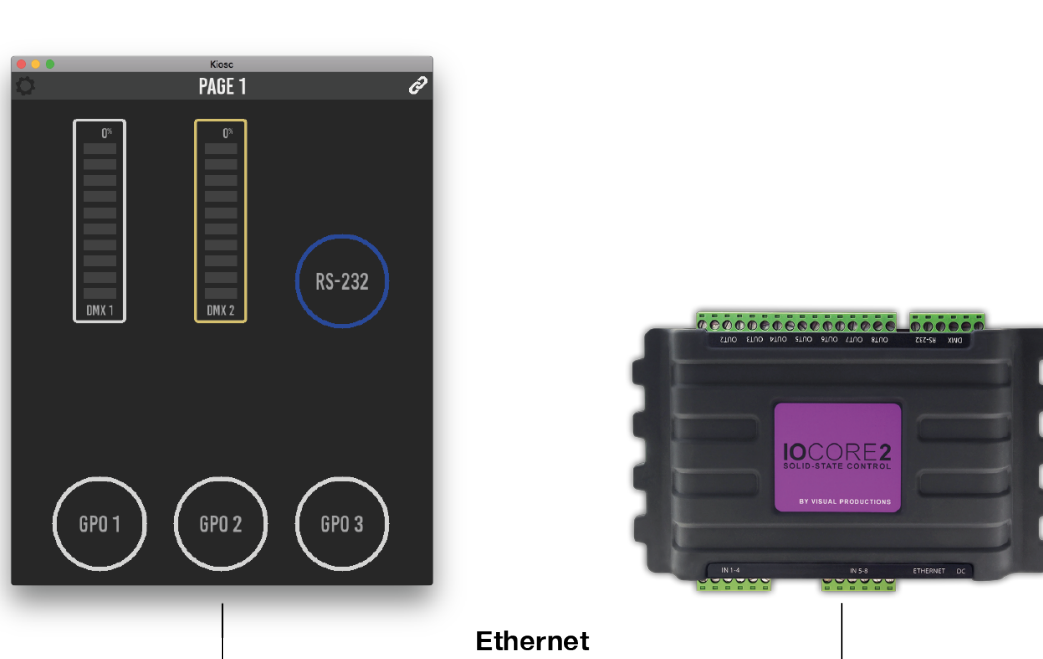
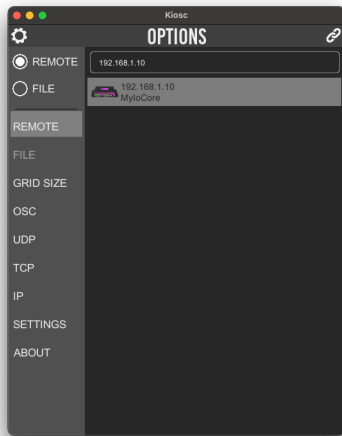


Figure 5.1: Kiosc connected to an ioCore2

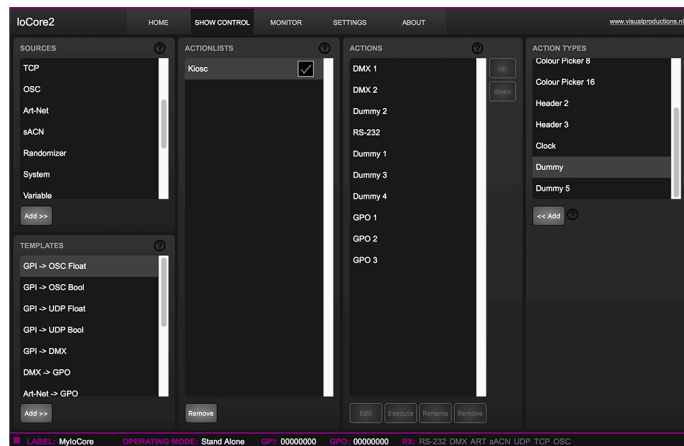
Kiosc can communicate with only one external device when operating in *Remote* mode. It is using a propriety protocol that is only supported by the Visual Productions range of devices. This protocol is uni-directional and thus provides no means for giving feedback to the Kiosc app. No OSC or UDP is used in this mode.

The *Remote* mode makes Kiosc compatible with VisualTouch, the previous-generation touch screen software from Visual Productions.

Figure 5.1 shows an example of Kiosc being connected to an ioCore2. By using the *Remote* mode, a simple dashboard can be created in a few mouse clicks. In this example Kiosc is used for setting DMX values, transmitting a RS-232 message and toggling GPO relays.



(a) Selecting the loCore2 remote server



(b) Actions inside the loCore2

Figure 5.2: Connecting Kiosc to an loCore2

Figure 5.2 shows how to select the loCore2 in the remote options. Once connected, the dashboard is created by adding a 'Kiosc' actionlist inside the loCore's show control page. Buttons, sliders and other controls can be added to this actionlist. The order of the actions determines the position of the controls in the Kiosc screen. Empty positions can be created adding *Dummy* actions. If there are more controls to fit in the screen, then extra pages will be created automatically. How many controls do fit in the screen is dictated by the *Grid Size* setting.

Figure 5.2 displays the actions created for this example. Programming actions is discussed in more detail in the loCore2's user manual.

5.1.2 File

In *File* mode Kiosc displays a layout designed with the KioscEditor. The file generated by the Editor needs to be stored locally on the device running Kiosc.

The Editor enables you to make custom control screens and to personalise them with logos and colours.

In *File* mode it is possible to control multiple external devices on the network. By using OSC, UDP or TCP these devices can also send data back to the Kiosc app, providing it with feedback that could update the status of GUI controls.

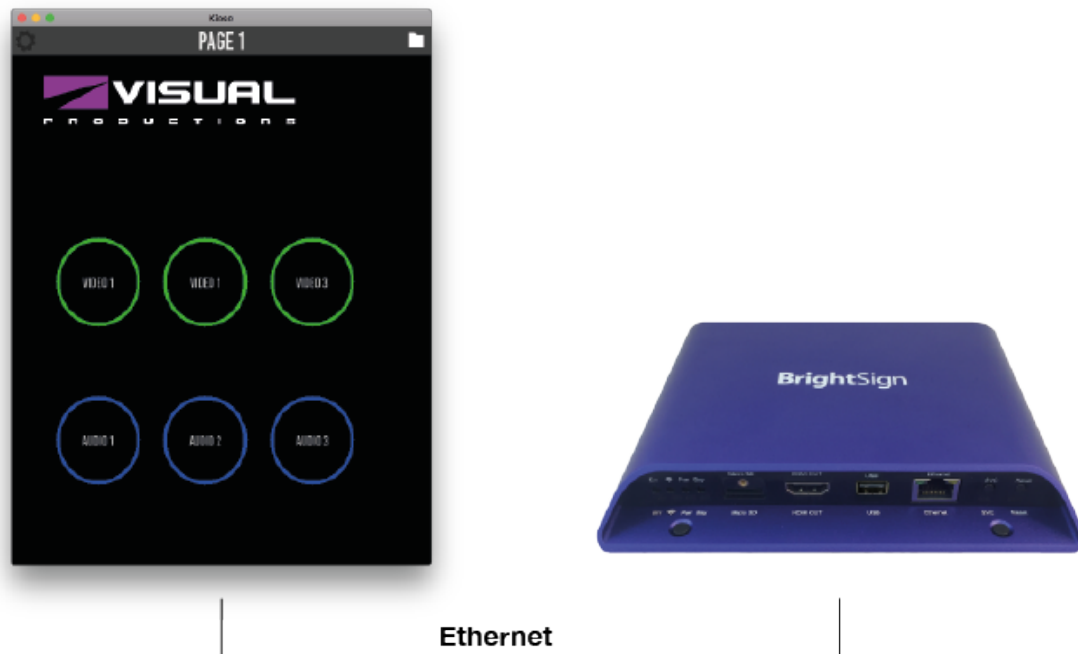


Figure 5.3: Kiosk connected to a media player

Figure 5.3 shows an example of Kiosk controlling a Brightsign, a 3rd party media player. Kiosk is using UDP messages to trigger the playback of different media files inside the Brightsign.

5.2 Protocols

Kiosk can communicate with external equipment via three protocols, OSC, UDP and TCP. All protocols run over the Ethernet network. This chapter describes these protocols.

5.2.1 OSC

Open Sound Control (OSC) is a protocol for communicating between software and various multi-media type devices. OSC uses the network to send and receive messages.

5.2.2 UDP

User Datagram Protocol (UDP) is a simple protocol for sending messages across the network. It is supported by various media devices like video projectors and Show Controllers. It does not incorporate error checking, making it an easy and fast protocol.

5.2.3 TCP

Transmission Control Protocol (TCP) is a connection-oriented protocol for sending messages across the network. It is supported by various media devices like video projectors and Show Controllers. It creates a reliable connection link between the sender and the receiver of the message and incorporates error checking. This makes it slower than UDP.

5.2.4 OSC vs UDP

There are many similarities between OSC and UDP. In fact, OSC uses UDP packets under the hood.

There is one difference between these protocols that is significant when choosing the right protocol for the job. OSC has a fixed syntax and UDP does not. Although UDP's technical characteristics are specified in the protocol, the syntax is not. This means that two UDP capable equipment from two different vendors will be able to send each other technically correct UDP messages. What they write into the messages is left to the discretion of the developers. As it turns out, different developers come up with different syntaxes. Even though the two devices say 'UDP' on the box, they might not be able to speak to each other.

This problem can be illustrated by an example. Let's assume a device from vendor X wants to send a message to a device from vendor Y. The message should set a parameter inside device Y to 50%. Device X could send something like:

- `parameter=50%` assuming a range from 0% to 100%
- `parameter=0.5` assuming a range from 0.0 to 1.0
- `parameter=128` assuming a range from 0 to 255
- `parameter=0x80` assuming a single hexadecimal byte

There are even more variations possible. The point being that whatever syntax vendor X has chosen, the chance that vendor Y choose the same is not that great.

OSC solves this problem by incorporating rules on syntax in its protocol specification. Each OSC compatible device uses exactly the same syntax. This is a strong reason to prefer OSC over UDP when possible. Of course, there are many devices out there that do not support OSC, leaving UDP an important asset inside your protocol toolbox.

5.2.5 UDP vs TCP

UDP and TCP are using the same message syntax. Thanks to the error checking TCP is more reliable than UDP and allows the sender to know if the message has really been received by the recipient. This is not possible with UDP, but that makes it faster. As a communication link must be established between the sender and the recipient, it makes broadcast unavailable for TCP compared to OSC and UDP where you can easily broadcast messages.

5.3 Options

The options screen presents a few settings for customising the Kiosc behaviour.

5.3.1 Modes

The modes selector allows you to choose between *Remote* and *File* mode. Modes are discussed on page 13.

5.3.2 Remote

This menu (see figure 5.4) is only enabled when in *Remote* mode. It allows you to select the remote server. E.g. a CueCore™ or KioskEditor. Please note that due to Microsoft Store security measurements, it is not possible to make a remote connection to KioskEditor if Kiosk and KioskEditor are running on the same PC.

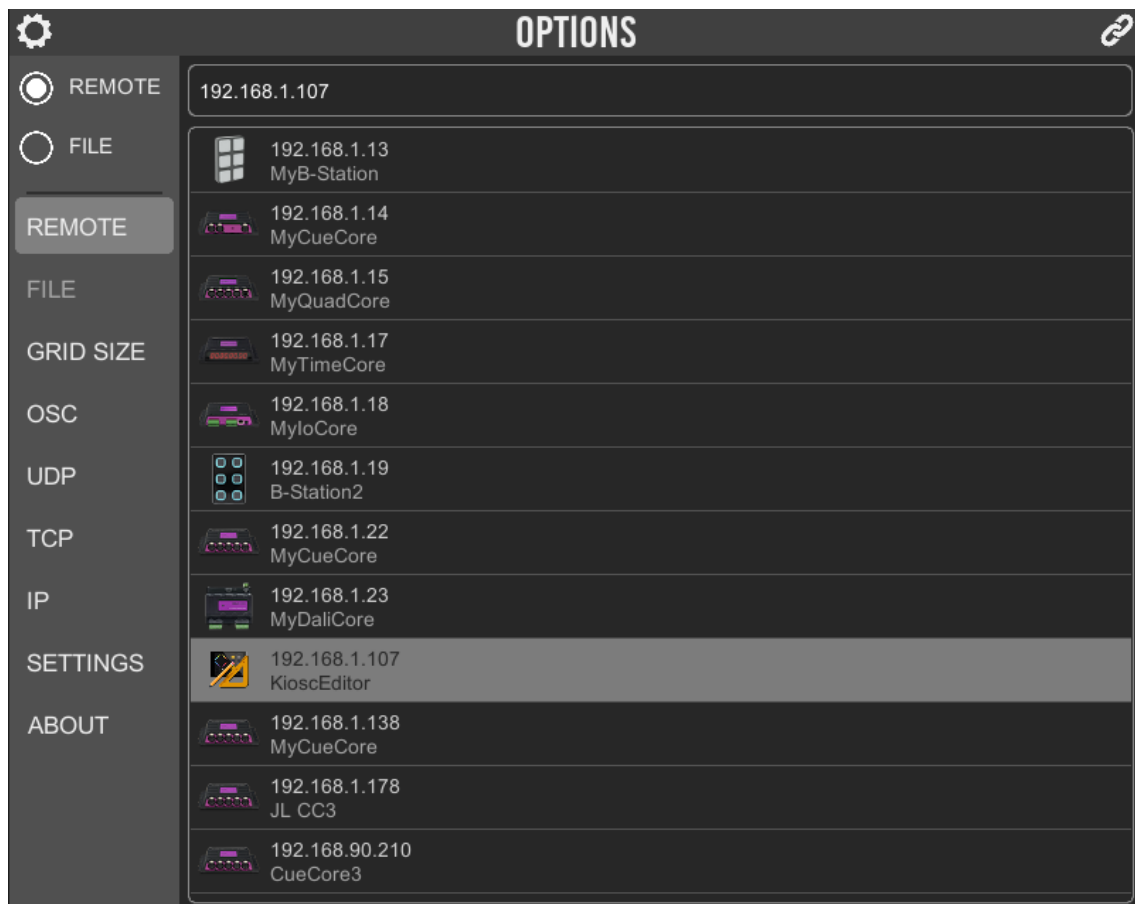


Figure 5.4: Remote options

5.3.3 File

This menu is only enabled when in *File* mode. It presents a list of the layout files. It also has buttons for saving, deleting and renaming files.

The files can be locked and unlocked by long pressing them. When locked, they can not be modified.

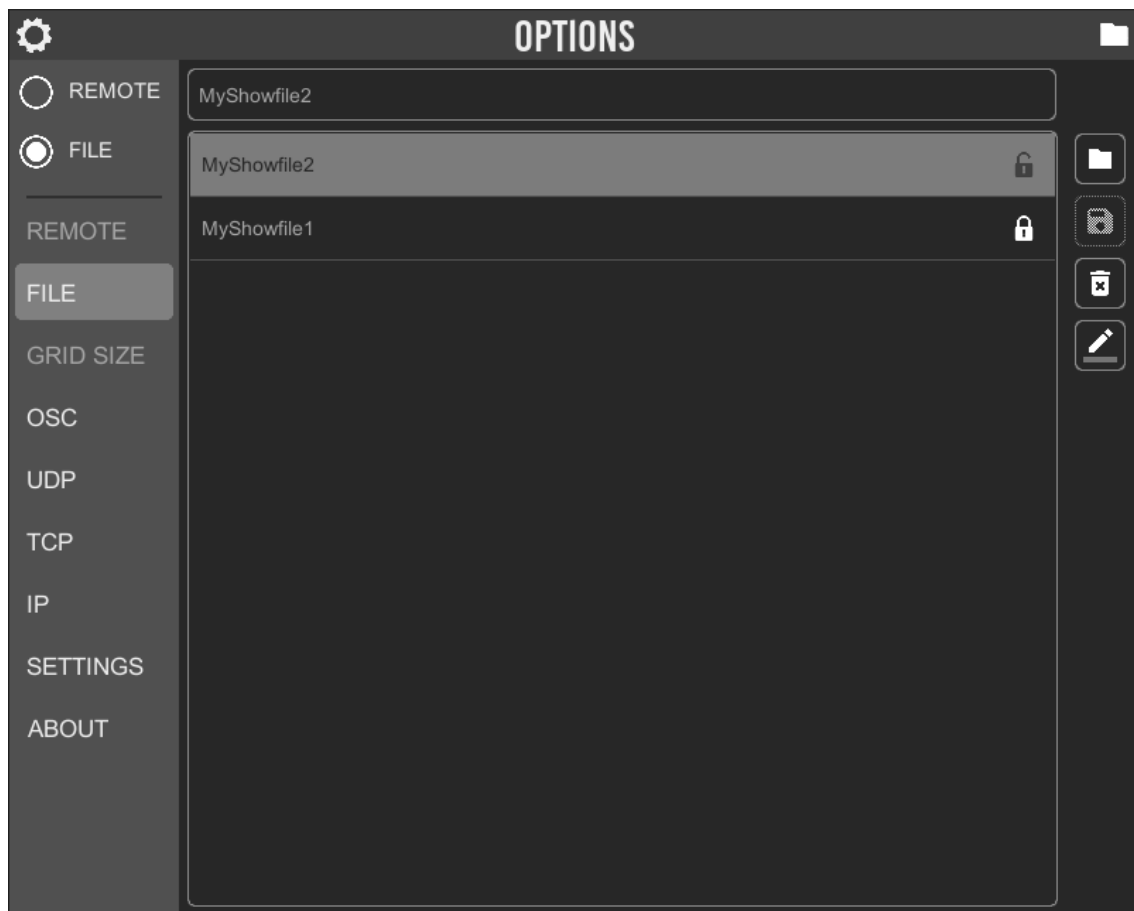


Figure 5.5: File options

The *Folder* button opens a file browser at the path where the files are stored.

5.3.4 Grid Size

In *Remote* mode (except when connected to *KioscEditor*) the size of the buttons and other controls are determined by the *Grid Size* setting. A smaller *Grid Size* results in larger controls. A larger *Grid Size* will allow more controls to fit on one page.

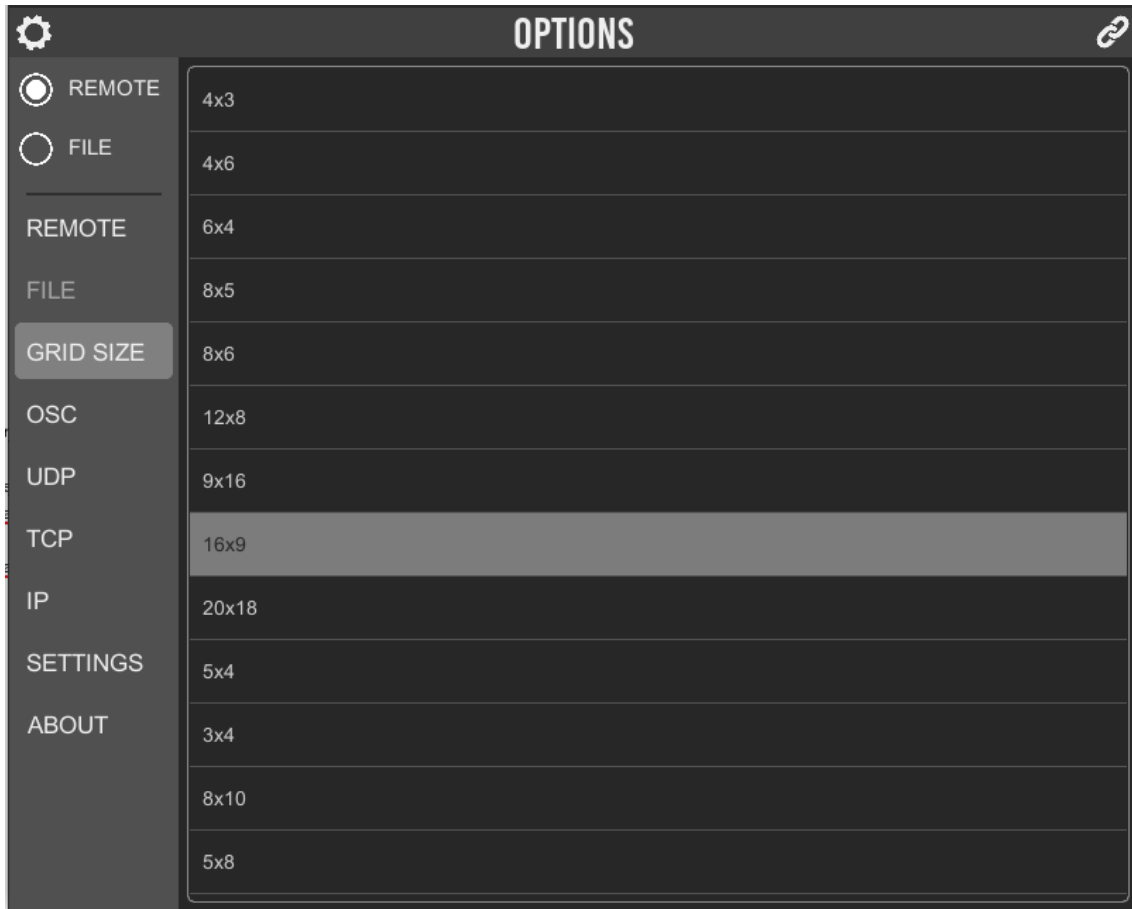


Figure 5.6: Grid Size options

The *Grid Size* menu is only available in *Remote* mode.

5.3.5 OSC

The settings in this screen (see figure 5.7) determine to which IP addresses the OSC messages are sent to. Multiple IP addresses can be added to the list. With each IP address there also needs to be specified a port number. Add the port number in the following fashion: `XXX.XXX.XXX.XXX:PORT`. Where the XXXs denote the IP address and PORT is a number between 1 and 65535.

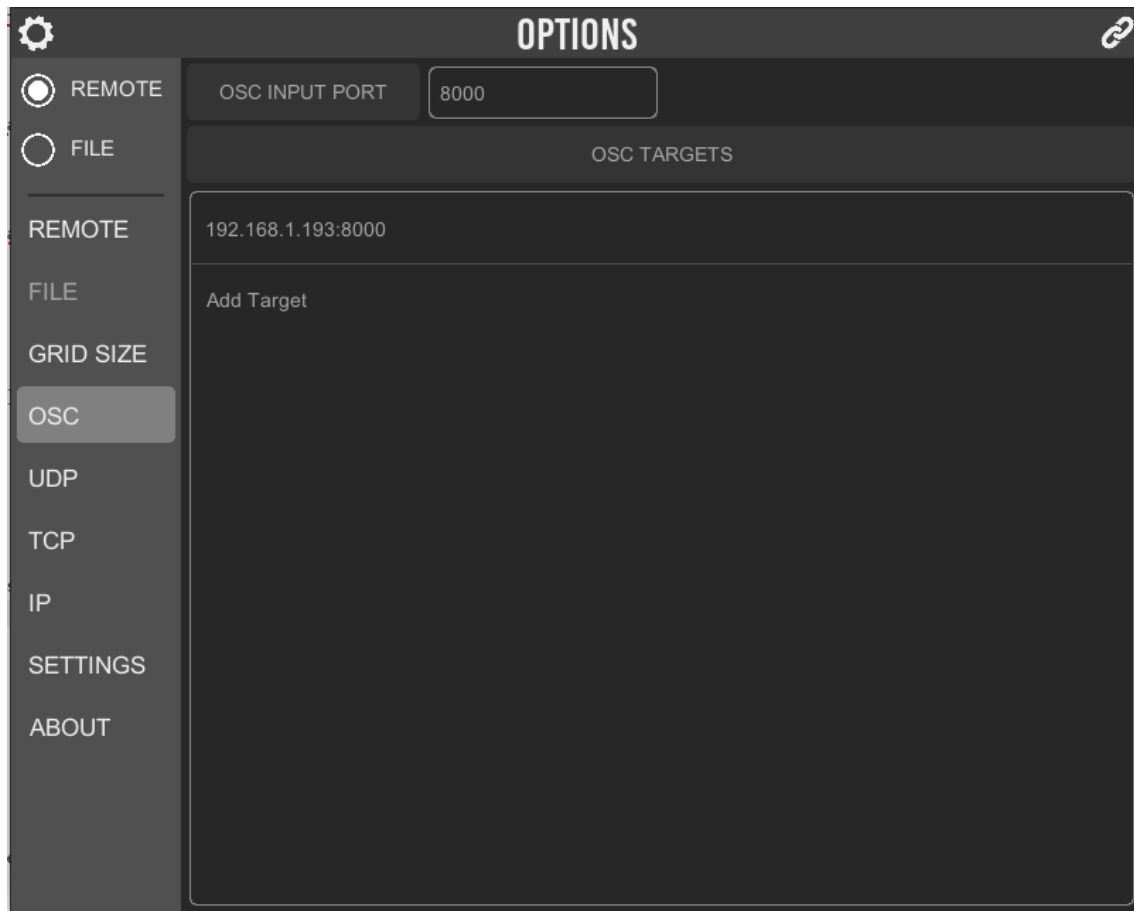


Figure 5.7: OSC options

The incoming port field specifies the port number on which Kiosc will receive OSC messages. External equipment that will send feedback information to Kiosc needs to know this incoming port number. By default this port number is set to 8000.

5.3.6 UDP

The settings in this screen (see figure 5.8) determine to which IP addresses the UDP messages are sent to. Multiple IP addresses can be added to the list. With each IP address there also needs to be specified a port number. Add the port number in the following fashion: `XXX.XXX.XXX.XXX:PORT`. Where the XXXs denote the IP address and PORT is a number between 1 and 65535.

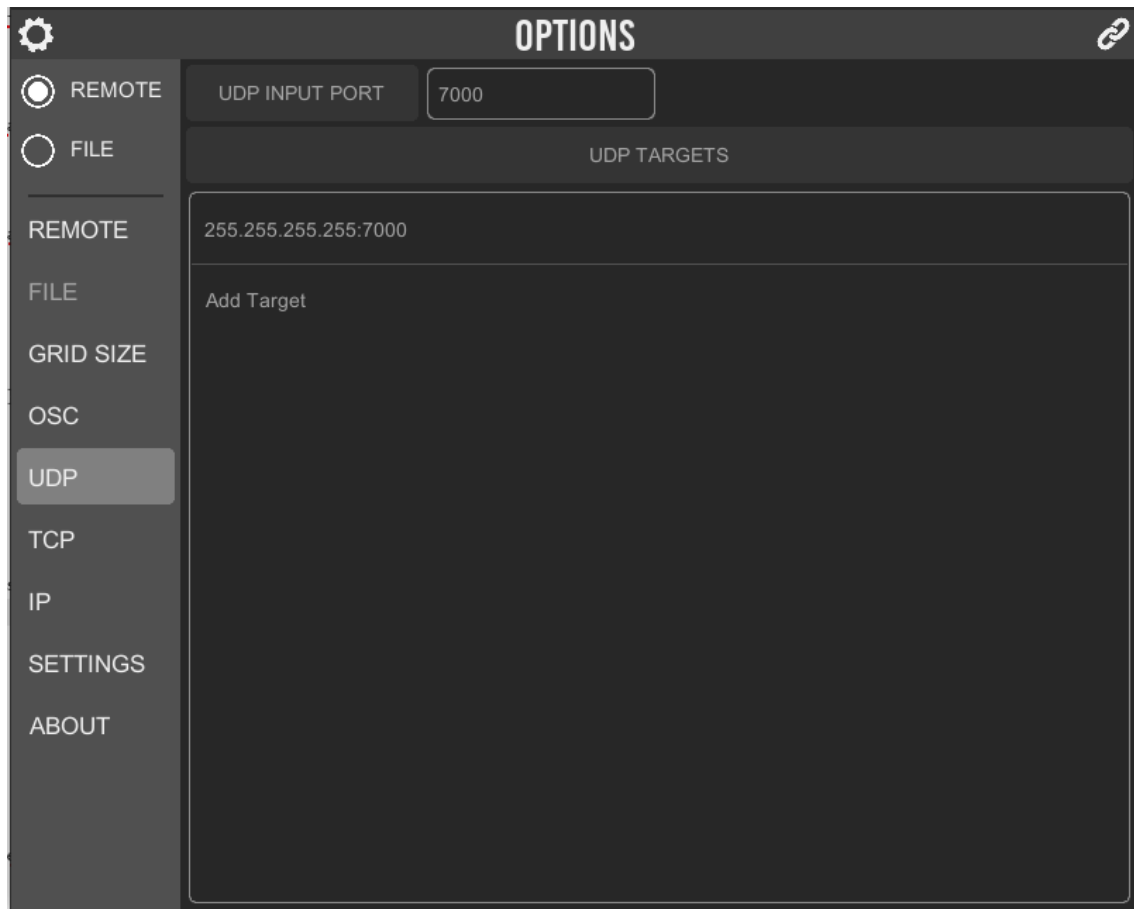


Figure 5.8: UDP options

The incoming port field specifies the port number on which Kiosk will receive UDP messages. External equipment that will send feedback information to Kiosk needs to know this incoming port number. By default this port number is set to 7000.

5.3.7 TCP

The settings in this screen (see figure 5.9) determine to which IP addresses the TCP messages are sent to. Multiple IP addresses can be added to the list. With each IP address there also needs to be specified a port number. Add the port number in the following fashion: `XXX.XXX.XXX.XXX:PORT`. Where the XXXs denote the IP address and PORT is a number between 1 and 65535.

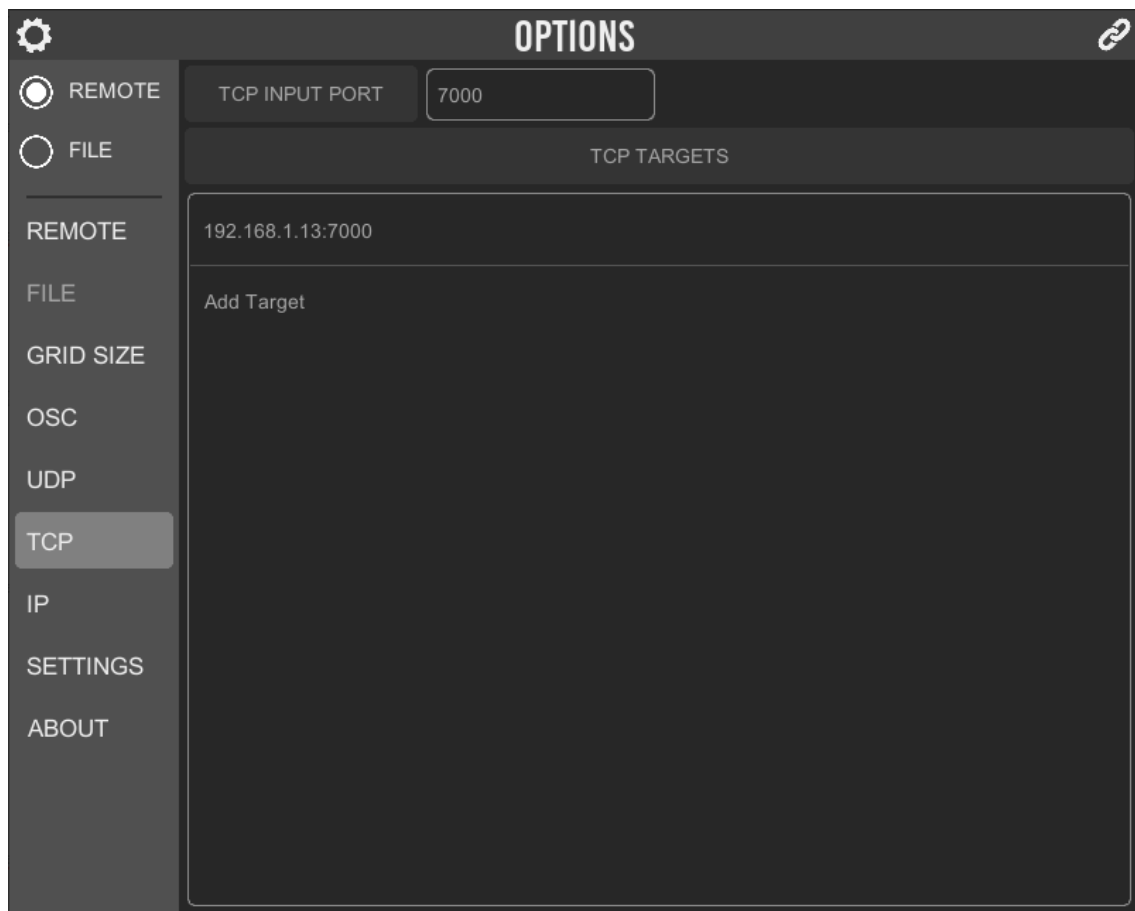


Figure 5.9: TCP options

The incoming port field specifies the port number on which Kiosk will receive TCP messages. External equipment that will send feedback information to Kiosk needs to know this incoming port number. By default this port number is set to 7000.

5.3.8 Settings

Language

You can easily change the language used by the Kiosk choosing one of the languages available in the list. See figure 5.10.

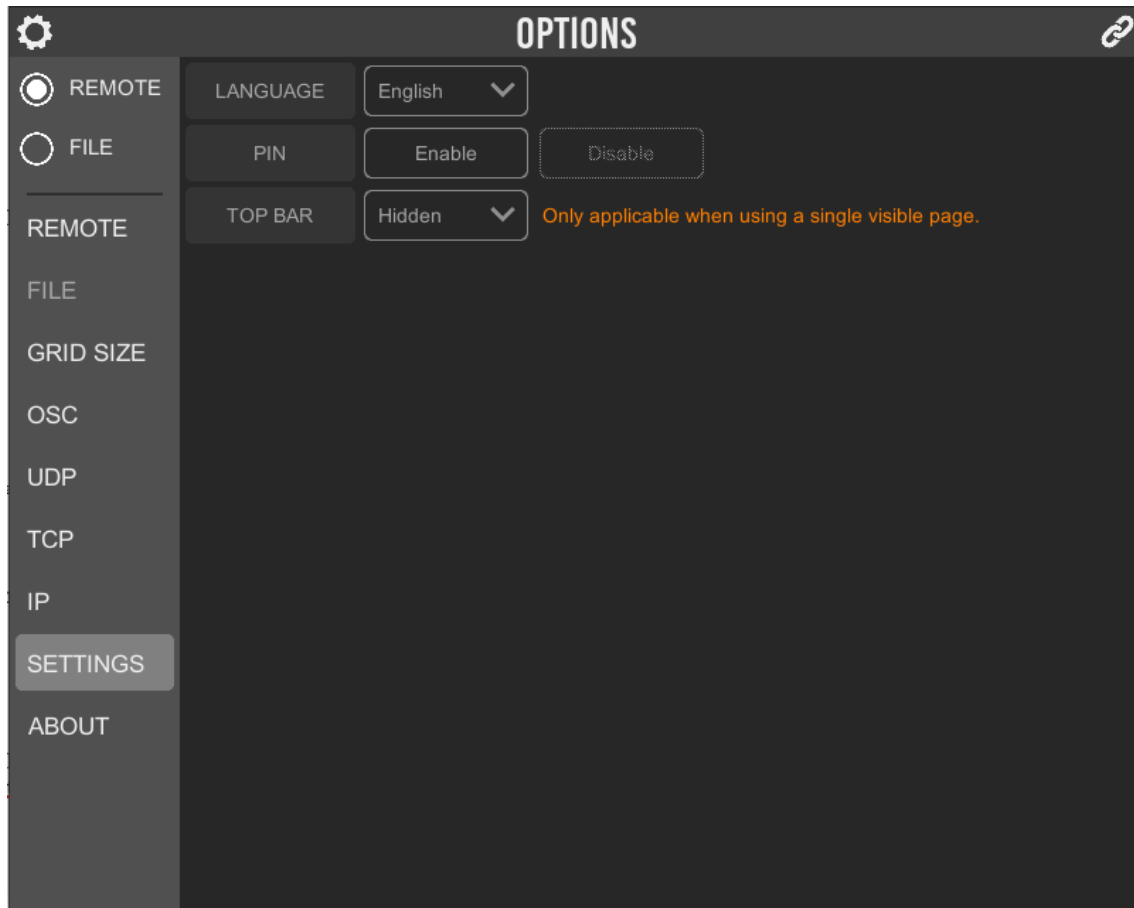


Figure 5.10: Changing settings

PIN

Although the Kiosk app has no editing capabilities (all editing is done in the KioskEditor), it does have several settings that, if erroneously changed, can stop the system from working. To restrict access to the settings a PIN code can be engaged. See figure 5.10.

Top Bar

The page selection top bar can be set as visible or invisible in the layout. Note that it is only possible to hide it when you have only one visible page in your layout. When the top bar is hidden, the *gear* icon for accessing the settings will still be available on the top right corner. See figure 5.10.

The *Settings* page of the KioskTouch2 displays some extra information compared to the Kiosk App, as shown below.

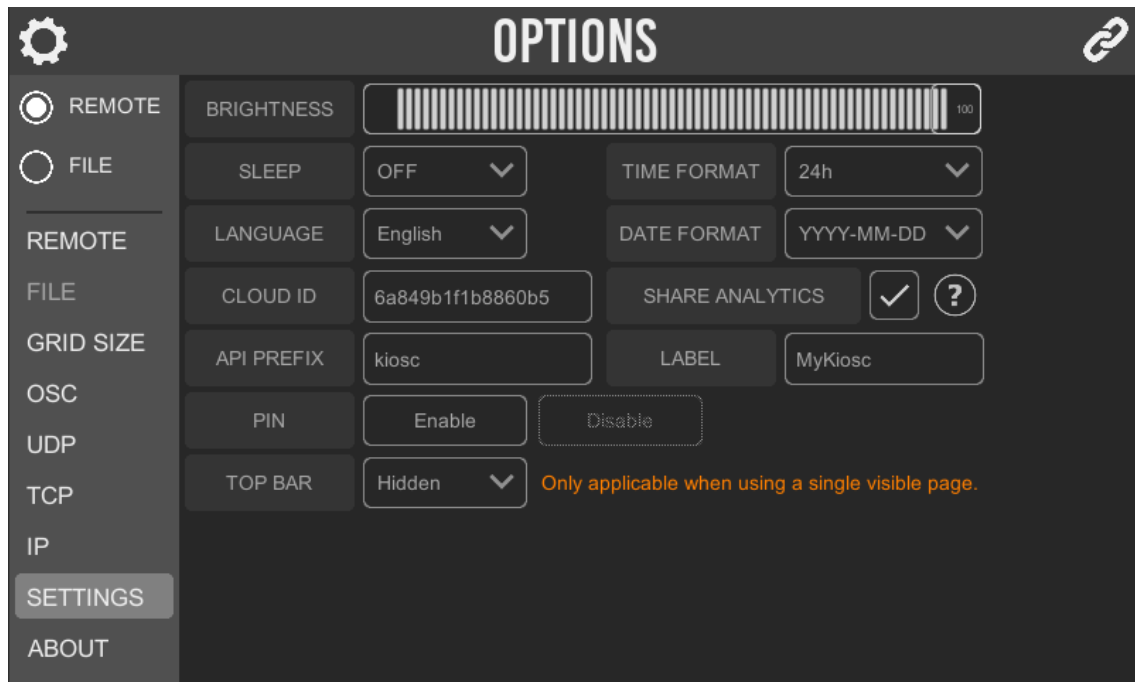


Figure 5.11: KioskTouch2 settings

Brightness

Set the screen's brightness of the KioskTouch. See figure 5.10.

Sleep

Set the time before the KioskTouch's screen goes into sleep mode. In sleep mode, the screen turns dark to avoid useless brightness and save energy. Just touch once the screen to wake up the KioskTouch. 5.10.

Cloud ID

The KioskTouch2, can be remotely supervised using the PurpleCloud you will find on the website <http://www.visualproductions.nl/>. Once registered in the PurpleCloud, a unique ID that will be issued allows to link Visual Productions devices to the cloud platform. See figure 5.10.

API Prefix

KioskTouch, include API commands to directly control some functions via OSC, UDP or TCP. See API commands list in the appendix `refsec:api`. Those commands start with a prefix that is set to *kiosc* by default. When using multiple devices from Visual Productions it can be useful to assign unique labels to these prefix, especially when using broadcasted messages. Read more about feedback loops in paragraph B.3.1. See figure 5.10.

Time Format

Set the format in which KioskTouch2 is going to represent time. There are two options between 24h or AM/PM.

Date Format

Set the format in which KioskTouch is going to represent date. There are four different options.

Share Analytics

If enabled, the unit is going to share data to connect to Visual Production's cloud platform: PurpleCloud.

Label

Displays and lets the user change the unit's name.

5.3.9 IP

The IP page differs between the Kiosk App and the KioskTouch, as shown below.

Kiosk App

On the Kiosk App the IP page shows a list of local IP addresses used by the device, as shown in figure 5.12. This information can be useful when troubleshooting network issues. The network settings are handled by the Operating System of the device running the Kiosk App.

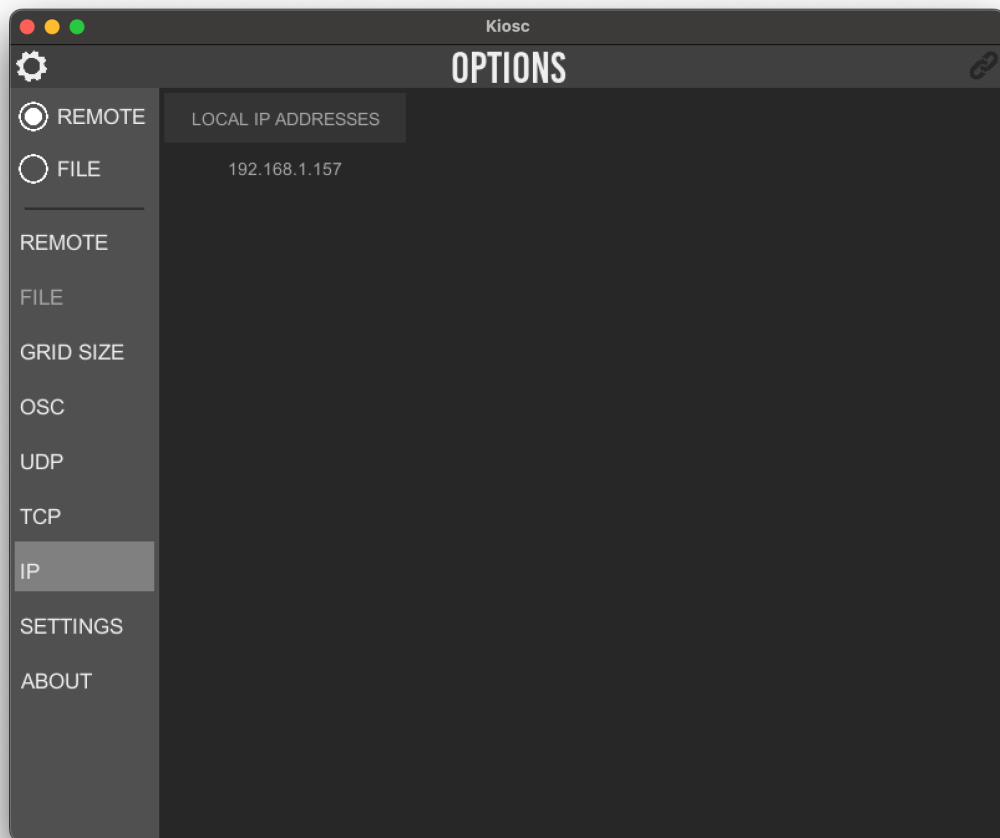


Figure 5.12: IP page on the Kiosc App

KioscTouch2

The IP page on the KioscTouch2 is used for setting the desired network settings, see figure 5.14. A static IP can be set, or the DHCP option can be checked for automatic configuration.

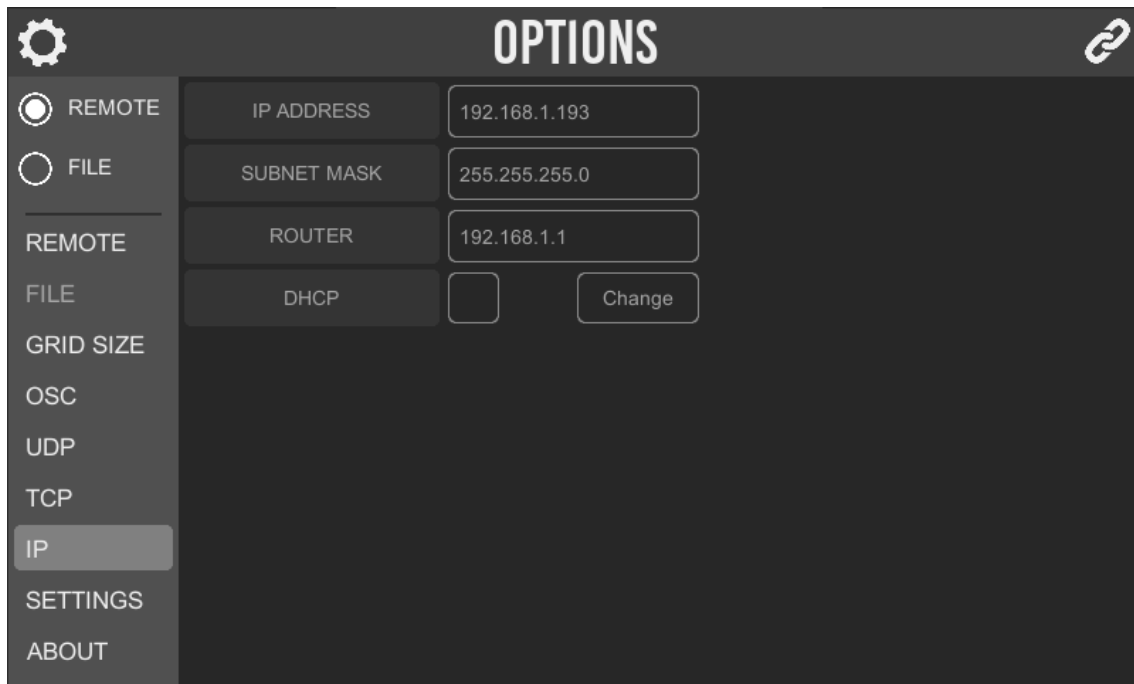


Figure 5.13: IP page on the KioscTouch

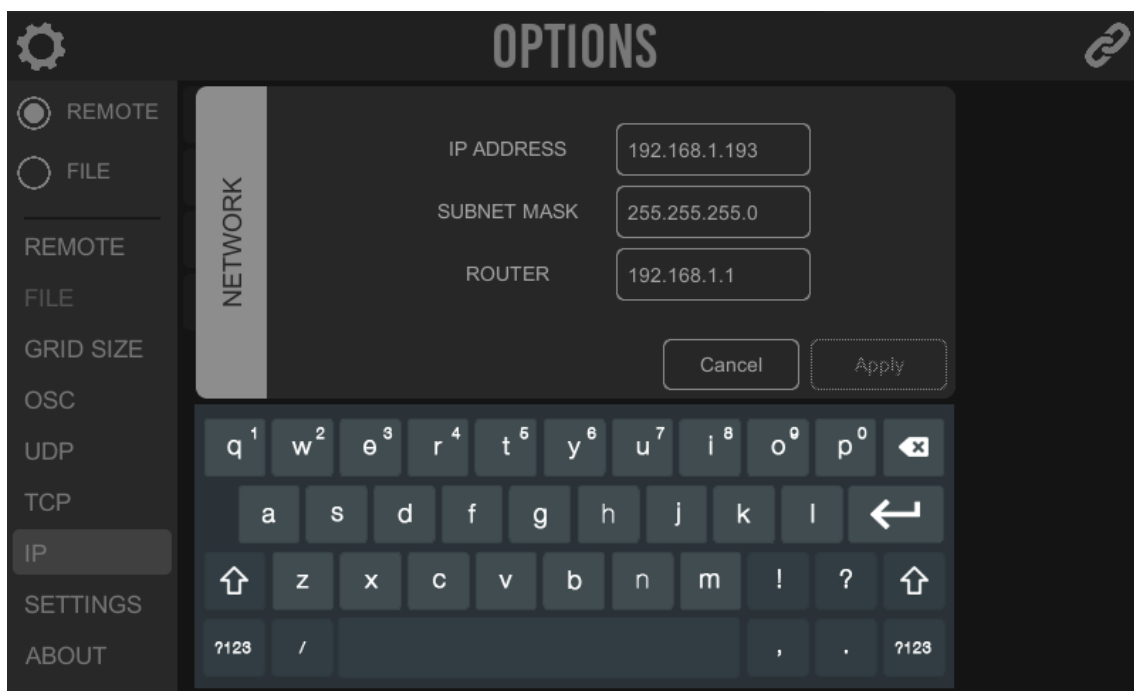


Figure 5.14: IP dialog on the KioscTouch2

5.3.10 About

Apart from the ubiquitous copyright notice, the about screen in figure 5.16 indicates the current software version.

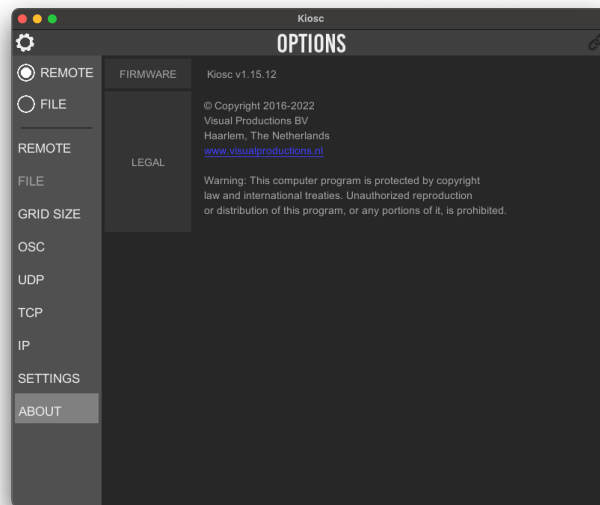


Figure 5.15: KIOSC app About screen

The *About* page of KioscTouch shows some complementary information about your device compare to the Kiosc app version. You will be able to find:

- Serial number
- PCB version
- Label (you can change it via vManager or in Settings page)
- Uptime since last power cycle.

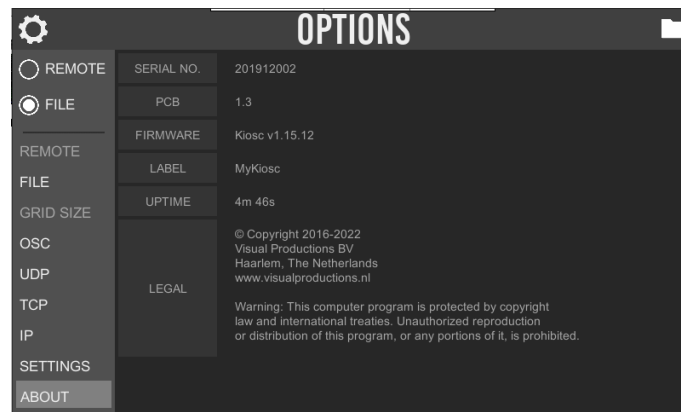


Figure 5.16: KioscTouch2 About screen

5.4 Installation

The Kiosc app is available on a wide range of operating systems, both mobile and desktop.



Figure 5.17: Kiosc on macOS, Windows and Ubuntu

The software is distributed through app-stores; there are no stand-alone installation packages (e.g. EXE, APK, DMG or DEB files) available. One of the major advantages of installing an app via an app store is to receive future software updates automatically.

- **iOS**

Kiosc can be downloaded from the Apple iOS app-store at:

<https://itunes.apple.com/us/app/kiosc/id1131605366>

- **Android**

Kiosc can be found on the Google Play store at

<https://play.google.com/store/apps/details?id=nl.visualproductions.kiosc>

Android 6.0 or higher is required.

- **Windows**

Kiosc can be downloaded from the Microsoft store at:

<https://www.microsoft.com/store/apps/9nblggh4s74f>

Windows 11 is required.

- **macOS**

Visit the Apple macOS app store at:

<https://apps.apple.com/us/app/kiosc/id1131605497>

macOS 11+ is recommended.

- **Ubuntu**

You can acquire the Kiosk app from the snap store:

<https://snapcraft.io/kiosk>

Alternatively, it can be installed by using the command-line:

```
snap find kiosk  
snap install kiosk
```

To update the apps later on via the command-line type:

```
snap refresh kiosk
```

Ubuntu 22.04 LTS is recommended. The app is only available for the amd64 architecture.

Chapter 6

Web Interface

The KioscTouch2 features an internal web interface used for configuration, monitoring, and diagnostics. A modern web browser is required during setup. After configuration, the KioscTouch2 can operate without an active web connection.

6.1 Home

The *Home* page provides a read-only overview of the KioscTouch2 's current operating status

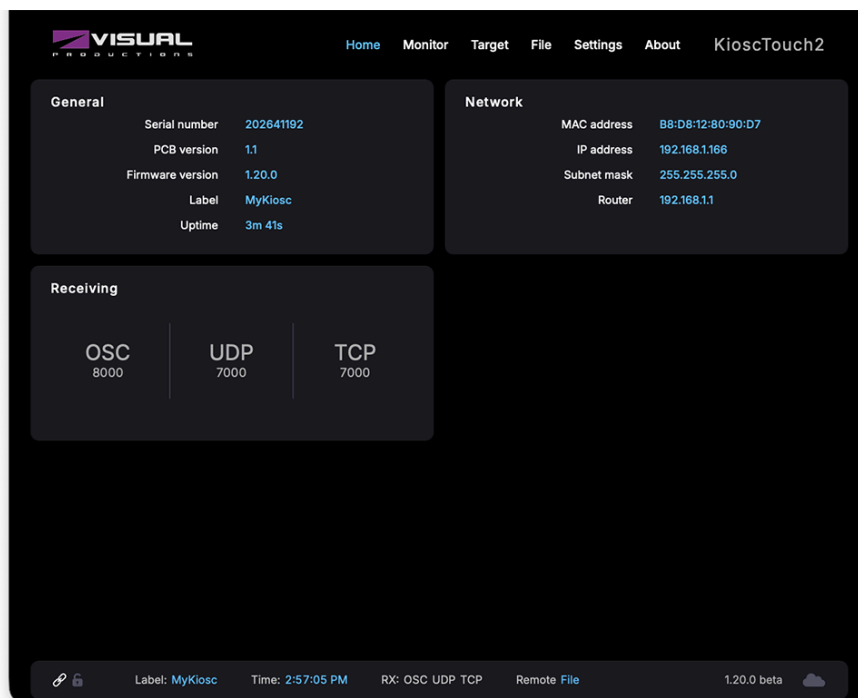


Figure 6.1: Home Page

The *Receiving* section shows which communication protocols are currently active. When data is received on a protocol, its indicator is highlighted. The same protocol activity indicators are also

visible in the Status Bar. See Figure 6.1.

6.2 Monitor

The *Monitor* page displays the most recent network messages transmitted and received by the KioskTouch2. This information can be particularly useful during installation, commissioning, and troubleshooting.

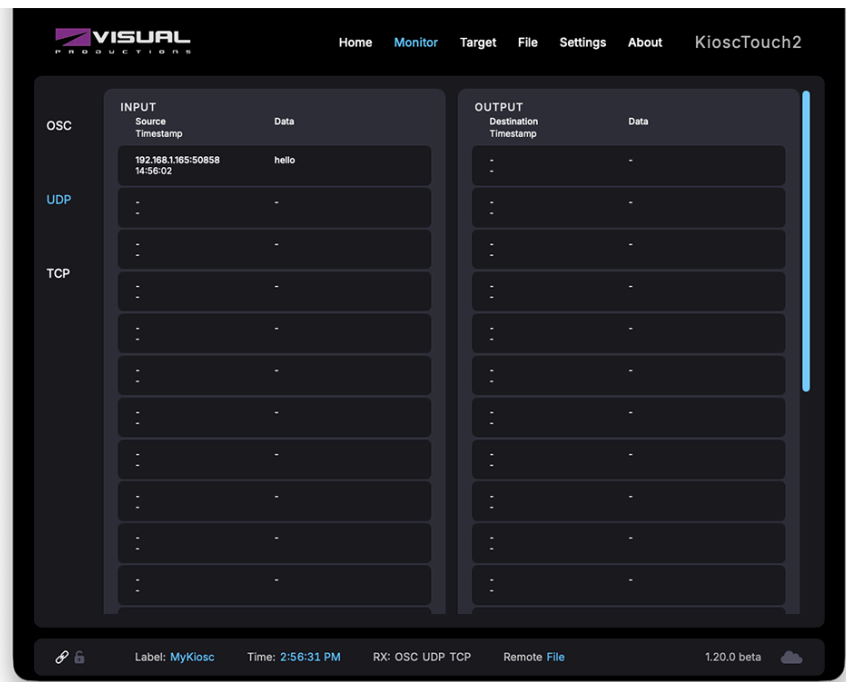


Figure 6.2: Monitor Page

All timestamps are shown relative to the device’s uptime. Messages highlighted in blue indicate valid API messages. For more information about the API protocol, see Appendix B on page 96.

6.3 Target

The *Target* page is used to configure the network destinations for outgoing messages generated by the KioskTouch2. Separate target lists are maintained for OSC, UDP, and TCP communication.

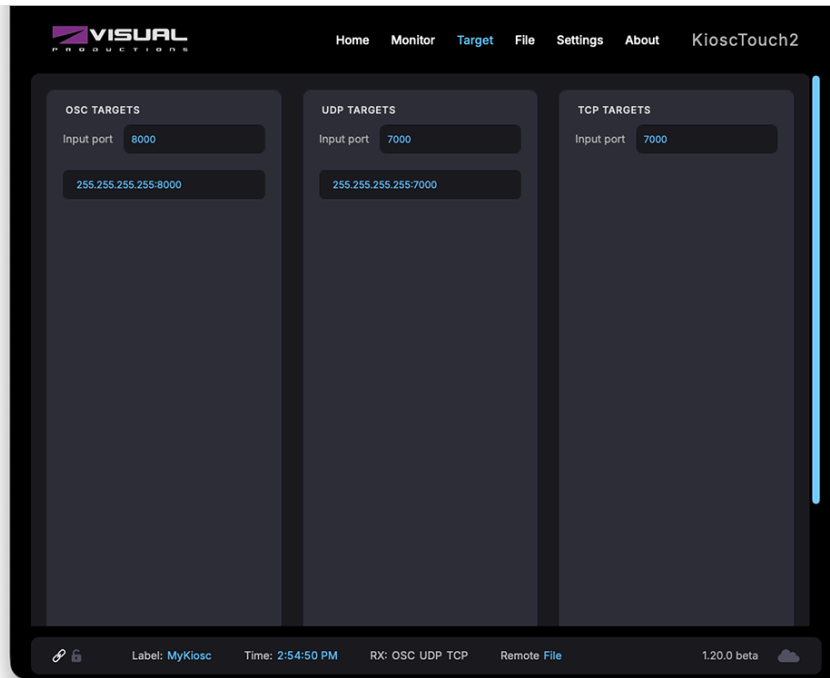


Figure 6.3: Target Page

Each protocol section contains an input port and a list of configured targets. A target is defined by an IP address and port number. Messages transmitted by the KioskTouch2 are sent to all targets configured for the corresponding protocol.

The input port specifies the local port on which the KioskTouch2 listens for incoming messages of that protocol. The configured targets below determine where outgoing messages are sent.

Targets can be added, edited, or removed as required by the installation. Broadcast addresses may also be used when communication with multiple devices on the network is required.

6.4 File

The *File* page is used to manage layout files stored on the KioskTouch2. Layout files contain the user interface configuration and associated settings that determine the behavior and appearance of the device.

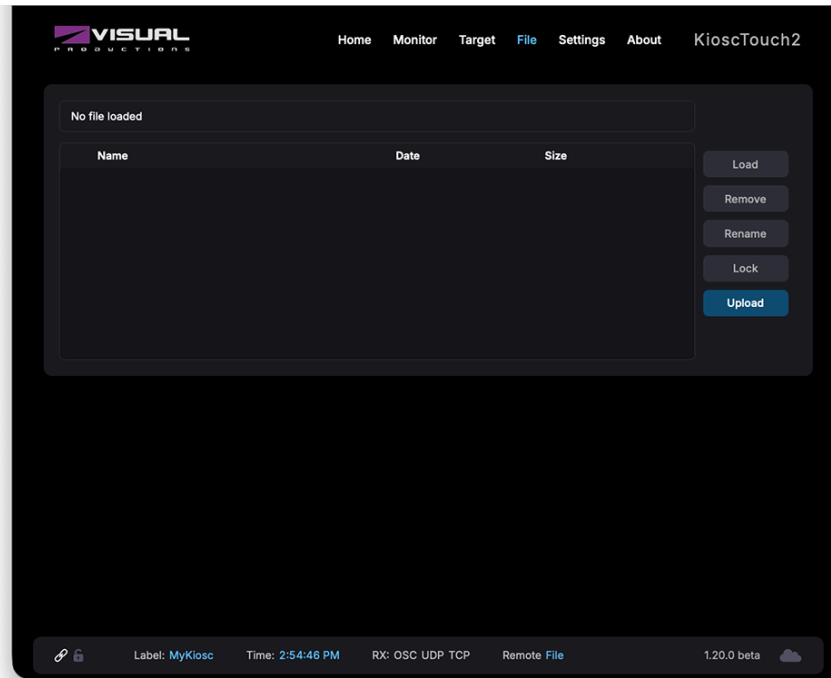


Figure 6.4: File Page

The currently loaded project is displayed at the top of the page. Below, a list of available project files is shown together with their modification date and file size.

The following actions are available:

Load Loads the selected layout file and makes it active.

Remove Permanently deletes the selected layout file from the device.

Rename Changes the name of the selected layout file.

Lock Protects the selected layout file from accidental modification or deletion.

Upload Uploads a new layout file to the device.

Only one project file can be active at a time. Loading a different file immediately replaces the current configuration with the selected project.

6.5 Settings

The *Settings* page provides access to the device's general configuration options. Settings are grouped into several categories, including General, Network, Options, API, Display, and Cloud.

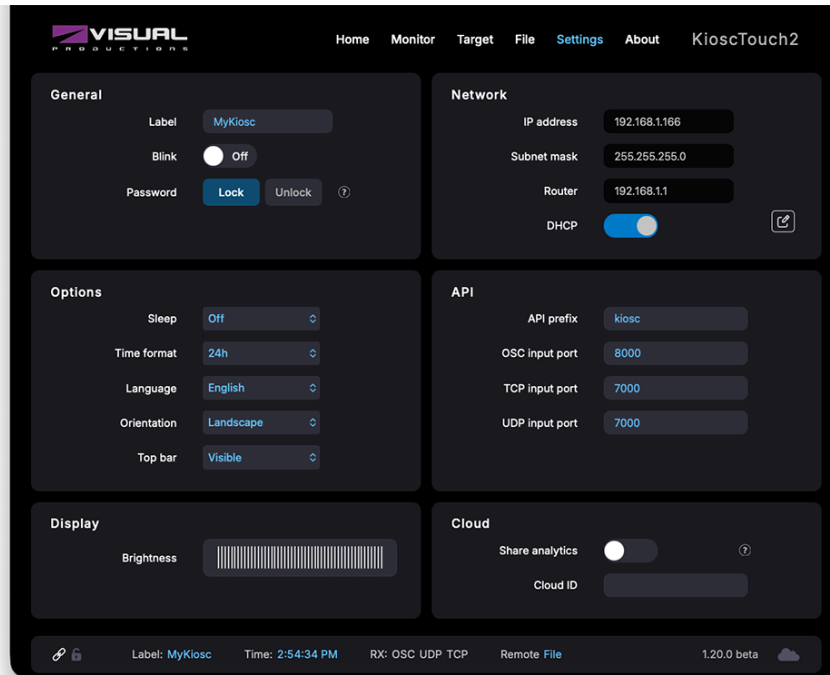


Figure 6.5: Settings Page

General

The *General* section contains device-specific settings:

Label Defines the name of the device. The label is displayed throughout the user interface and can help identify the device on the network.

Blink Causes the device to visually identify itself, making it easier to locate in an installation with multiple units.

Password Locks or unlocks access to configuration settings.

Network

The *Network* section contains the Ethernet configuration of the device.

IP address The device's network address.

Subnet mask The subnet mask used by the network.

Router The default gateway address.

DHCP Enables automatic network configuration from a DHCP server.

When DHCP is enabled, the network settings are assigned automatically by the network infrastructure.

Options

The *Options* section contains user interface and operating preferences.

Sleep Configures the display sleep behavior.

Time format Selects either 12-hour or 24-hour time display.

Language Selects the user interface language.

Orientation Sets the screen orientation.

Top bar Controls the visibility of the status bar.

API

The *API* section contains settings related to network communication.

API prefix Defines the prefix used for API message paths.

OSC input port The port used for incoming OSC messages.

TCP input port The port used for incoming TCP messages.

UDP input port The port used for incoming UDP messages.

Display

The *Display* section allows adjustment of the screen brightness to suit the installation environment.

Cloud

The *Cloud* section contains settings related to cloud connectivity and analytics.

Share analytics Enables the anonymous sharing of device analytics.

Cloud ID Displays the unique identifier used for cloud services.

Refer to chapter 7 on page 40 for more information on the textitPurpleCloud.

6.6 About

The *About* page provides information about the KioscTouch2, including hardware and software version details, support resources, and legal information.

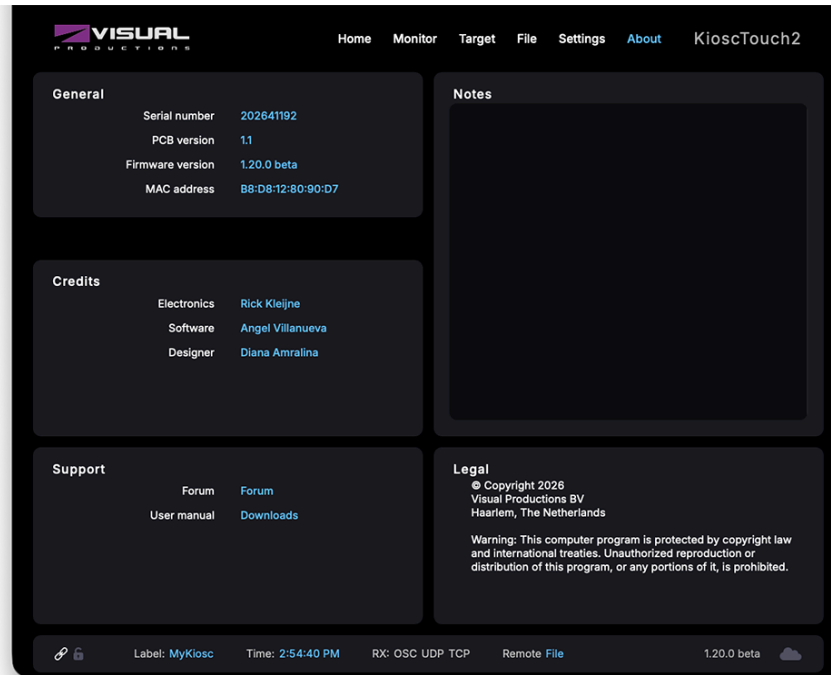


Figure 6.6: About Page

General

The *General* section displays identification and version information for the device.

Serial number The unique serial number assigned to the device.

PCB version The hardware revision of the printed circuit board.

Firmware version The currently installed firmware version.

MAC address The Ethernet MAC address of the device.

This information may be requested when contacting technical support.

Notes

The *Notes* section provides a space for installation-specific notes and documentation. It can be used to record information such as project details, network settings, maintenance records, or other relevant comments.

Credits

The *Credits* section acknowledges the individuals involved in the design and development of the product.

Support

The *Support* section provides quick access to online resources, including the Visual Productions forum and the latest user documentation.

Legal

The *Legal* section contains copyright and licensing information relating to the firmware and software running on the device.

Chapter 7

PurpleCloud

PurpleCloud is the cloud platform developed by Visual Productions for remote access, monitoring, and management of supported devices.

When connected to the internet and linked to a PurpleCloud account, the KioskTouch2 can be securely accessed from anywhere. This enables installers, system integrators, and support personnel to monitor device status, perform maintenance tasks, and manage installations without requiring direct access to the local network.

PurpleCloud provides the following benefits:

- Remote access to devices from any location.
- Centralized management of multiple installations.
- Simplified troubleshooting and support.
- Device status monitoring and diagnostics.
- Secure cloud-based connectivity without complex network configuration.

7.1 Register

The *PurpleCloud* is accessible via <https://www.visualproductions.nl/cloud>.

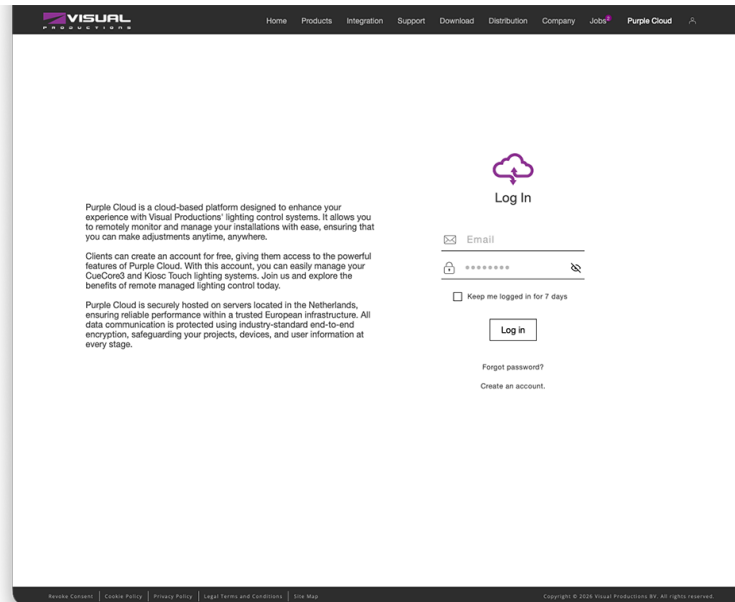


Figure 7.1: Login to PurpleCloud

Creating a PurpleCloud account is free of charge and provides access to remote device management, project organization, cloud backups, and other cloud-based services.

PurpleCloud is available in multiple subscription tiers.

Freemium

The *Freemium* plan is available at no cost and is suitable for small installations and evaluation purposes. It includes support for:

- Up to 25 devices
- Up to 10 projects
- Up to 10 backup files
- Up to 10 users

Premium

The *Premium* plan is intended for larger installations and organizations that require additional capacity and advanced features. Premium accounts provide increased limits for devices, projects, backups, and users, as well as access to RDM fixture details.

For information about current pricing and subscription options, please contact Visual Productions sales.

Custom

For large-scale deployments requiring capacities beyond the Premium plan, custom subscriptions are available. Please contact Visual Productions sales to discuss your requirements.

7.2 Account

The *Account* page contains user, organization, subscription, and cloud identification settings associated with your PurpleCloud account. It also provides tools for managing team members and configuring support access.

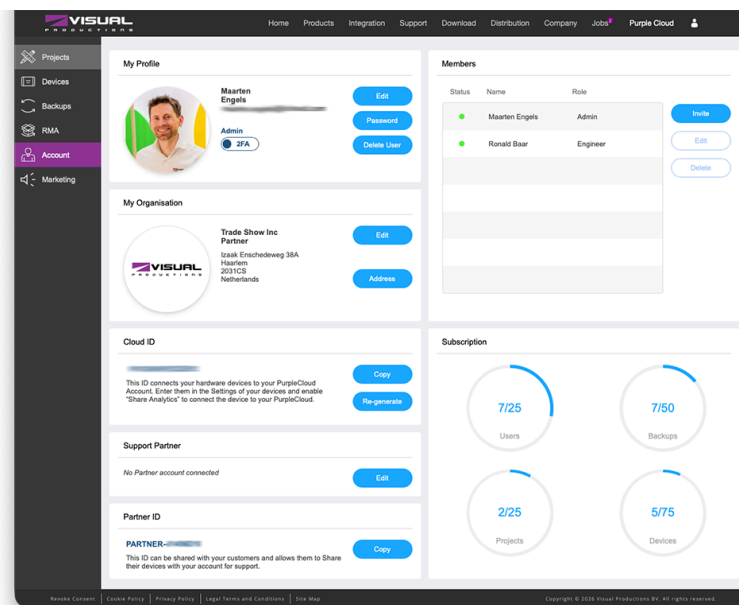


Figure 7.2: Account Page

My Profile

The *My Profile* section displays information about the currently logged-in user, including name, email address, and account role.

The available actions allow you to:

Edit Update personal account information.

Password Change the account password.

2FA Enable or disable two-factor authentication.

Delete User Permanently remove the user account.

Members

The *Members* section lists all users who have access to the organization account.

For each member, the following information is displayed:

Status Indicates whether the user is currently active.

Name The user's name.

Role The assigned access level within the organization.

There are three different Role's available:

User Users can view information and manage RMAs.

Engineer Engineers can browse devices and edit projects.

Admin Administrators can invite new members and manage existing user accounts and permissions.

My Organization

The *My Organization* section contains information about the organization associated with the PurpleCloud account, including company name and address details.

This information is used throughout PurpleCloud and can be updated as required.

Cloud ID

Each PurpleCloud account is assigned a unique *Cloud ID*. This identifier is used to link compatible Visual Productions devices to the account.

To connect a device:

1. Enable cloud connectivity on the device.
2. Enter or associate the displayed Cloud ID.
3. The device will appear in the PurpleCloud account once the connection has been established.

The Cloud ID can be copied for easy distribution or regenerated when required.

Support Partner

The *Support Partner* section allows an authorized installer, distributor, or service provider to be associated with the account.

When a support partner is configured, they may be granted access to assist with configuration, maintenance, and troubleshooting tasks.

Partner ID

The *Partner ID* can be shared with customers to allow them to grant support access to your organization. This simplifies remote assistance and device management across multiple installations.

Subscription

The *Subscription* section provides an overview of account usage and available resources. Depending on the subscription plan, limits may apply to:

- Users
- Projects
- Devices
- Backups

Usage indicators show the current consumption relative to the available subscription capacity.

7.3 Projects

The *Projects* page is the main workspace within PurpleCloud. Projects are used to organize devices according to installations, venues, customers, or events. Each project can contain one or more devices and serves as a central location for monitoring and managing an installation.

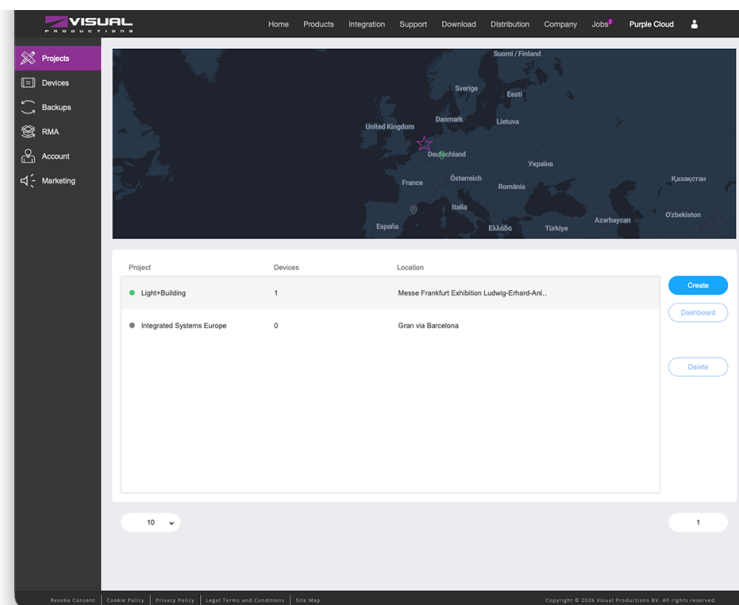


Figure 7.3: Projects Page

A map at the top of the page provides a geographical overview of all projects associated with the account. Projects with a known location are displayed on the map, making it easy to locate installations around the world.

Below the map, a table lists all available projects together with the following information:

Project The name of the project.

Devices The number of devices assigned to the project.

Location The physical location associated with the project.

Selecting a project enables the available project actions:

Create Creates a new project.

Dashboard Opens the dashboard for the selected project.

Delete Permanently removes the selected project.

Projects provide a convenient way to group devices and keep large installations organized. For example, a project may represent a building, venue, exhibition stand, hotel, retail store, or any other installation containing one or more Visual Productions devices.

7.4 Devices

The *Devices* page provides an overview of all devices connected to your PurpleCloud account. From this page, devices can be monitored, managed, backed up, and accessed remotely.

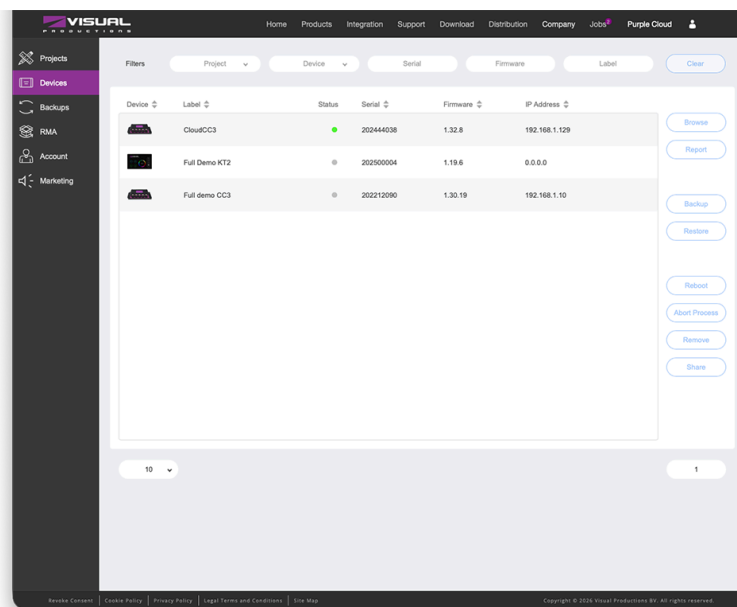


Figure 7.4: Devices Page

The device list displays key information for each registered device:

Device The type of Visual Productions device.

Label The user-defined device name.

Status Indicates whether the device is currently online or offline.

Serial The device serial number.

Firmware The installed firmware version.

IP Address The current network address of the device.

Filtering

The filter controls at the top of the page can be used to locate devices based on project, device type, serial number, firmware version, or label. The *Clear* button removes all active filters.

Device Management

After selecting a device, several management functions become available:

Browse Opens the device dashboard and provides access to its status and configuration.

Report Generates a diagnostic report for the selected device.

Backup Creates a backup of the device configuration and stores it in PurpleCloud.

Restore Restores a previously created backup to the device.

Reboot Restarts the device remotely.

Abort Process Cancels an active operation, such as a firmware update or restore procedure.

Remove Removes the device from the PurpleCloud account.

Share Grants access to the device to another PurpleCloud user or organization.

Online Status

The status indicator provides a quick overview of device availability. Online devices can be accessed and managed remotely, while offline devices remain visible in the inventory but cannot be reached until they reconnect to PurpleCloud.

The Devices page serves as the central location for managing all connected Visual Productions hardware within an account.

7.5 Backups

The *Backups* page provides centralized storage and management of device backups within PurpleCloud. Backups can be used to archive configurations, restore devices to a known state, or transfer settings between devices.

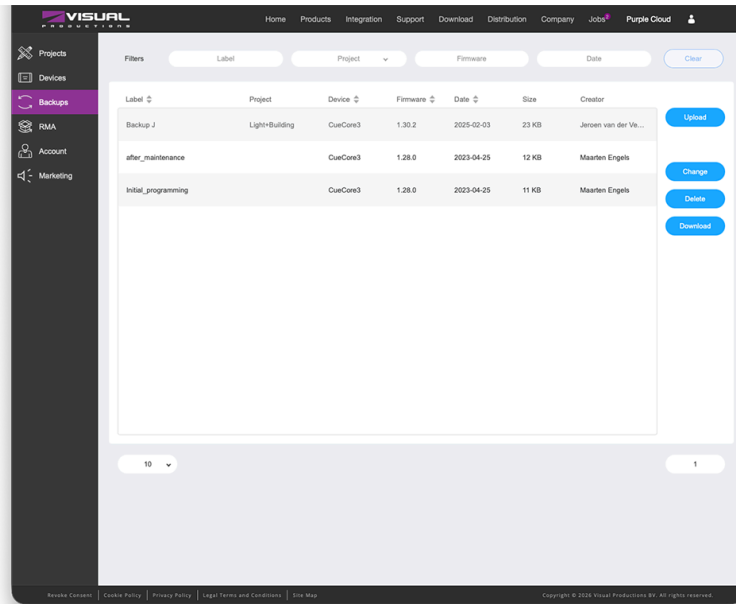


Figure 7.5: Backups Page

The backup list displays all backups available within the account. For each backup, the following information is shown:

Label The name assigned to the backup.

Project The project associated with the backup.

Device The device type from which the backup was created.

Firmware The firmware version active when the backup was created.

Date The creation date of the backup.

Size The file size of the backup.

Creator The user who created the backup.

Filtering

The filter controls at the top of the page can be used to quickly locate backups by label, project, firmware version, or date. The *Clear* button removes all active filters.

Backup Management

The following actions are available for the selected backup:

Upload Uploads an existing backup file to PurpleCloud.

Change Modifies the properties of the selected backup.

Delete Permanently removes the selected backup.

Download Downloads the selected backup to a local computer.

Restoring Configurations

Backups can be downloaded and restored to compatible devices when required. This allows installers and operators to quickly recover from configuration changes, hardware replacement, or accidental data loss.

Maintaining regular backups is recommended, especially before performing firmware updates or making significant changes to a device configuration.

Chapter 8

KioscEditor

The KioscEditor is a software tool for designing Kiosk layouts.

At the time of writing this manual the KioscEditor software is at version 1.18.0.

8.1 Installation

The Editor is available on Microsoft Windows, Apple macOS and Ubuntu Linux. The software is distributed through app-stores; there are no stand-alone installation packages (e.g. EXE, APK, DMG or DEB files) available. One of the major advantages of installing an app via an app store is to receive future software updates automatically.

- **Windows**

KioscEditor can be downloaded from the Microsoft store at:

<https://www.microsoft.com/store/apps/9n4ph6cs6tth>.

Windows 11 is required.

- **macOS**

Visit the Apple macOS app store at:

<https://apps.apple.com/us/app/kiosc-editor/id1257068566>

macOS 26 or higher is recommended.

- **Ubuntu**

You can acquire KioscEditor from the Snapcraft store:
<https://snapcraft.io/kiosceditor>

Alternatively, it can be installed by using the command-line:

```
snap find kiosceditor  
snap install kiosceditor
```

To update the apps later on via the command-line type:

```
snap refresh kiosceditor
```

Ubuntu 20.04 LTS is recommended. The app is only available for the amd64 architecture.

8.2 The Editor

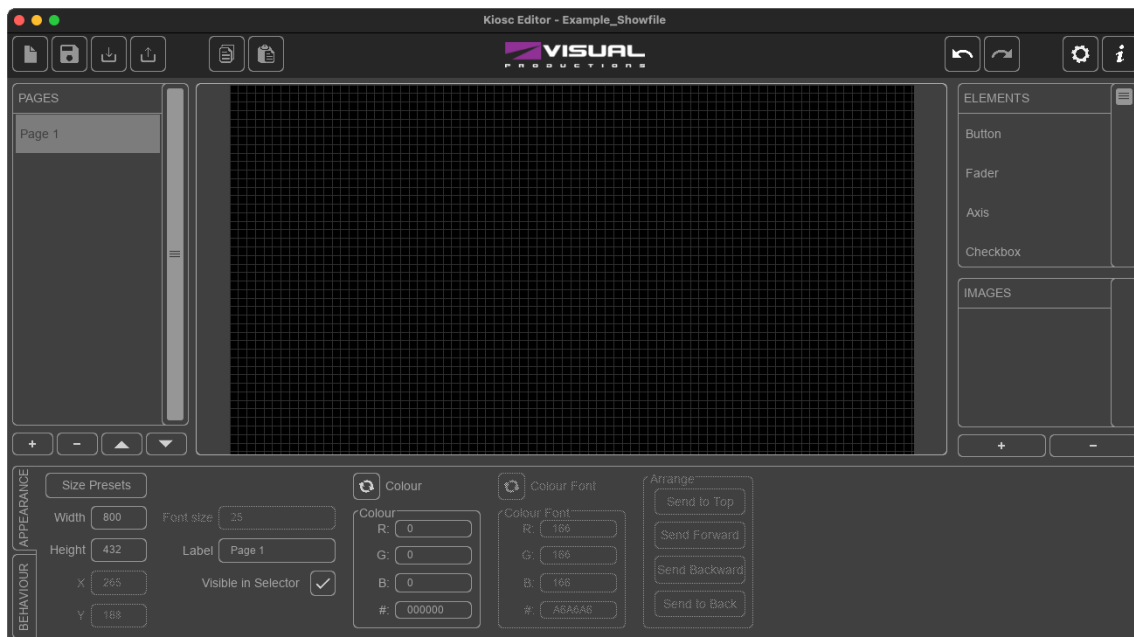


Figure 8.1: KioscEditor user-interface

Figure 8.1 shows the Editor's user-interface. Each section is discussed in detail below.

8.2.1 Menu Bar

At the top of the Editor screen there is a menu bar. Reading from left to right, it contains buttons for New File, Save, Save As and Open. There are clipboard buttons for Copy and Paste. Next buttons are Undo and Redo. Finally, there are buttons to open the Options and About dialogs.



Figure 8.2: Menu bar

8.2.2 Pages

Your design needs at least one page and can include up to 16 pages. Add more pages by pressing the + button in figure 8.3. The *Up* and *Down* buttons allow you to change the order of the pages.

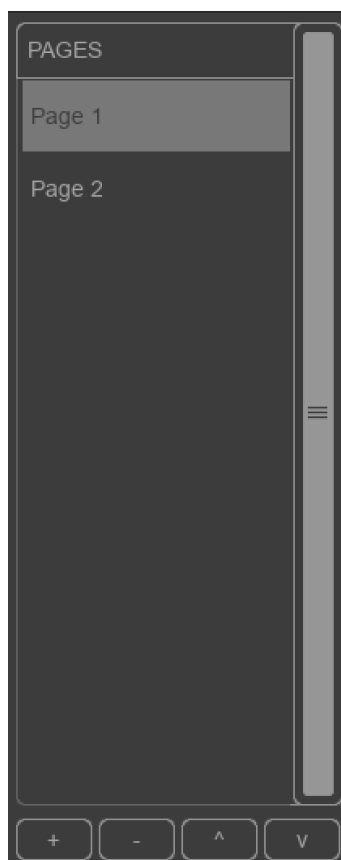


Figure 8.3: Pages

You can rename the page by selecting it and then changing the *Label*.

8.2.3 Elements

Controls like buttons, faders, encoders, and others are called *Elements* (see figure 8.4). A layout is basically created by dragging several *Elements* into the *Pages*.

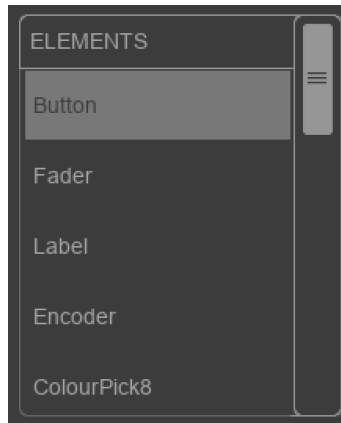


Figure 8.4: Elements

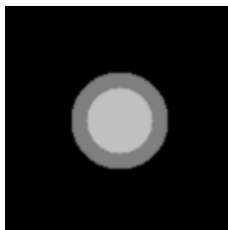
More information on each element can be found in appendix A. The available Elements include:

Button



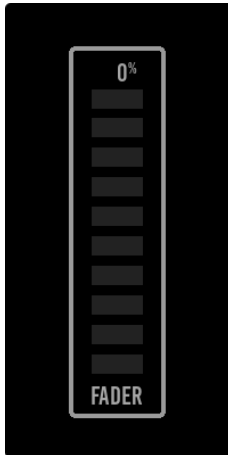
The button can be used to send a bool value (ON/OFF) or integer value (0 or 1). When set to Control, the Button sends ON or 1 when pressed, and OFF or 0 when released. Using Set the Button can send any integer value when pressed.

Indicator



The indicator can be used as a virtual LED. It does not send anything, but using feedback the inner circle can light up and change colour.

Fader



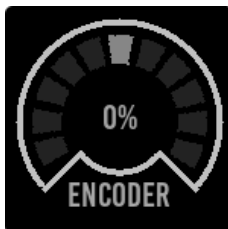
The fader can send a range of values. The range can be set using the Min and Max values. The type of value can be set to a float or an integer. A fader can simply be resized to get a horizontal fader. Displayed value can be changed between numeric value and percentage.

Label



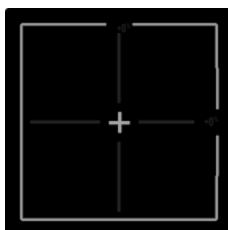
The label element is for creating a clear, easy to use layout. It has no function when pressed.

Encoder



The Encoder sends a float or integer value. Minimum, maximum and default value can be set.

Axis



The Axis element is for controlling XY values, like a moving head's pan and tilt.

Colourpicker 8



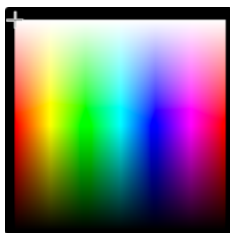
The Colourpicker 8 has eight colours to choose from. These are sent as OSC colour value, or via UDP/TCP using Hex #RRGGBB.

Colourpicker 16



The Colourpicker 16 has sixteen colours to choose from. These are sent as OSC colour value, or via UDP/TCP using Hex #RRGGBB.

ColourSpectrum



The Colourspectrum has many colours to choose from. These are sent as OSC colour value, or via UDP/TCP using Hex #RRGGBB.

Wheelcolour



The Wheelcolour has many colours to choose from. These are sent as OSC colour value, or via UDP/TCP using Hex #RRGGBB.

WheelRGBW



The WheelRGBW has many colours to choose from, which can be mixed with the fader that controls the white value. The colour is sent as OSC colour value (abusing the alpha channel for white) or via UDP/TCP using Hex #RRGGBBWW.

WheelColTemp



The WheelColTemp is ideal to mix between warm and cold white. It sends a float value between 0.00 and 1.00

Text



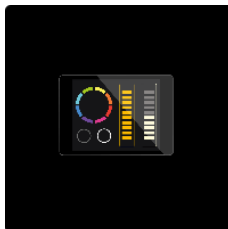
The text element is a simple text box. You can type text here and send it as a OSC or UDP/TCP message. It can also receive and display text.

Checkbox



A checkbox element can be used as a button. It will send ON or 1 when checked, and OFF or 0 when unchecked.

Image



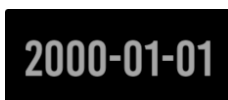
The image element is perfect for adding your own logo to your Kiosk layout. It behaves similar to the button element, so it can be used to create custom buttons.

Clock



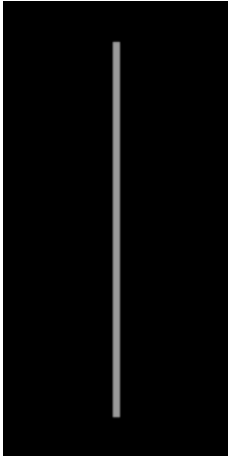
The clock element displays the current time of the system. It has no other functions when pressed.

Date



The date element displays the current date of the system. It has no other functions when pressed.

Line



The line element is used purely for design, it does not send any commands but is ideal as a separator between controls. It can be turned into an horizontal line by changing the size ratio. It can change colour using feedback.

8.2.4 Images

A *Page* can have a background image. Also, *Image Element* can display an image. To use images in your design, you first have to import them into the images list (see figure 8.5). Add images by pressing the + button. Images are read from a specific folder. The import dialog allows you to browse this folder by clicking on the *Folder* button.

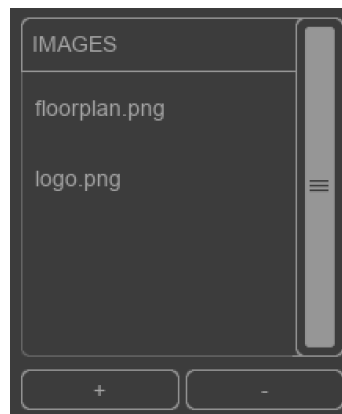


Figure 8.5: Images

To set a background image for a *Page*, select the *Page* and click on one of the images from the list. You can clear the background image clicking on an empty position in the list or clicking on the same image again.

Setting an *Image Element* is done in a similar way. Add the *Image Element*, make sure it is selected, then click on one of the images from the list.

8.2.5 Appearance Properties

This section allows you to change the (default) appearance of the page or selected *Elements*.

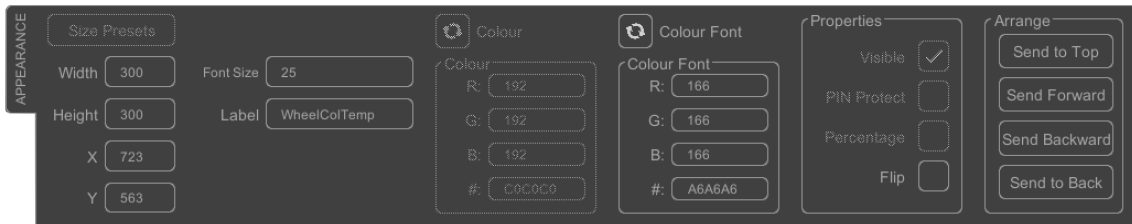


Figure 8.6: Appearance properties

8.3 Screen Size

Before starting to design your dashboard it is recommended to set the screen size for your layout. To change to screen size of your layout, make sure no Elements are selected, then set the desired width and height in the fields as shown in 8.7.



Figure 8.7: Setting the Screen Size

For your convenience, there is a popup dialog with size presets for common mobile devices. Press the *Size Presets* button to open this dialog.

Size Presets

When no element is selected, the size affects the size of the canvas. Presets can be chosen by pressing the *Size Presets* button, and selecting the target device.

From this windows you will also be able to select if you want your layout to be landscape or portrait, and also if your layout will include the top bar or not as this will impact the actual size of it.

If your layout already include some elements and you change the size of it, then a popup message will ask you if you want to automatically scale the elements.

Size and position

When an element is selected, *Size Presets* is disabled. The size of the element is decided by setting a desired *Width* and *Height*. The placing of the element can be done either via drag&drop, or by setting the *X* and *Y* values.

Fontsize and Label

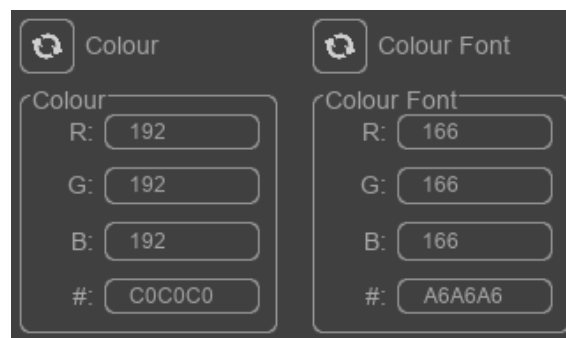
This decides the label and the font size for the selected element. When a page is selected, the label of the page can be set here.

Visible in selector

Currently only applies to pages and buttons. When selected, the page or button will not be shown on the device running Kiosk. Useful for hiding work-in-progress pages or special buttons.

Colours

The colour of the *Element* can be changed by setting the RGB fields in figure 8.8. It also features RGB fields for changing the *Element*'s font colour. A different colourpicker for selecting the colour is possible by pressing the button next to *RGB Fields*.



Colour	Colour Font
R: 192	R: 166
G: 192	G: 166
B: 192	B: 166
#: C0C0C0	#: A6A6A6

Figure 8.8: Colour properties

Arrange

When *Elements* overlap, you can use the *Arrange* buttons to control which *Element* lies at the top (see figure 8.9).



Arrange

Send to Top

Send Forward

Send Backward

Send to Back

Figure 8.9: Arrange buttons

Properties

Some elements can have specific properties that can be set in the properties section.

Visible Page and buttons have a "Visible" property. Not visible pages can be displayed via the *automation* section.

Pin Protect The access to some pages can be locked via a requested password. To use it, the *Pin Protect* option must be activated and a *Pin* must be set in the *Kiosc app* or *KioscTouch*.

Percentage Select this option if you want your fader to display a percentage value instead of the numeric value.

Flip For the *Colour temperature wheel* you can select it if you want the warm temperature to be on the left or on the right of the wheel.

For the *Fader* you can select it if you want the range to be flipped.

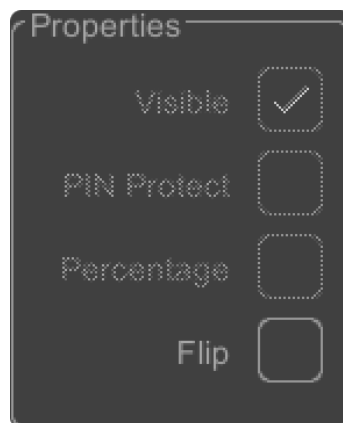


Figure 8.10: Properties options

8.3.1 Behaviour Properties

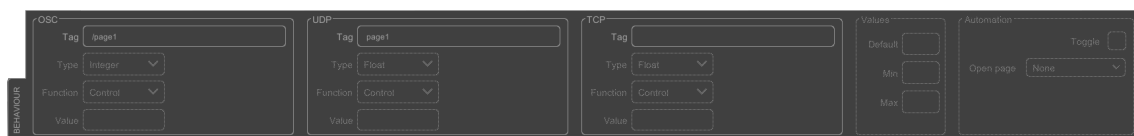


Figure 8.11: Behaviour properties

OSC/UDP/TCP

When the user clicks on a button in Kiosc, or controls any other element, it typically sends an OSC, UDP or TCP message to an external device. Which message is send out is determined by the *OSC*, *UDP* or *TCP* settings (see figure 8.13).

An OSC tag should always start with a '/' prefix (if you forget it typing your tag *KioscEditor* will automatically add it). For some *Elements* it is possible to specify the data type that will be sent in the OSC, UDP or TCP message. For instance, for a button *Bool* (true/false) or *Unsigned* (1/0) can be specified. The Function and Value fields can further specify the value that will be sent. More information on each specific element can be found in appendix A.

The figure shows three side-by-side panels for configuring OSC, UDP, and TCP messages. Each panel contains the following fields:

- OSC Panel:** Tag: ; Type: Integer (dropdown); Function: Control (dropdown); Value: .
- UDP Panel:** Tag: ; Type: Float (dropdown); Function: Control (dropdown); Value: .
- TCP Panel:** Tag: ; Type: Float (dropdown); Function: Control (dropdown); Value: .

Figure 8.12: Tags

UDP

You can use the UDP tag to send hex values. To do that just start the UDP tag by "\0x". You can indicate the end of the TCP message by adding "\n" at the end of your message or just "\" if you want to add a normal text part to your tag.

TCP

The TCP tag gets some extra features.

You can use the TCP tag to send hex values. To do that just start the TCP tag by "\0x". You can indicate the end of the TCP message by adding "\n" at the end of your message or just "\" if you want to add a normal text part to your tag. Make sure to NOT place any spaces between your hex values. This will make the line be read as text instead.

The valid special characters are: "\n" (line feed), "\r" (carriage return), "\t" (tab), "\0" (number 0) and "\\" (backslash).

The figure shows the TCP configuration panel with the following fields:

- Tag:**
- Type:** Bool (dropdown)
- Function:** Control (dropdown)
- Value:**
- Individual Target:** ☐ (checkbox)
- Kiosc List:**

Figure 8.13: TCP tags

The TCP tag can also be set to be sent to a unique IP address. Just check the *individual target* check box to activate it and set the target IP and port in the beside input.

Note that when the *individual target* is activated the TCP tag will not be sent to the IP addresses set in the TCP target list of KIOSC.

TCP

Tag: \\tag1

Type: Bool

Function: Control

Value: 192.168.1.23:7000

Individual Target: ☒

Figure 8.14: TCP individual target

Values

Some *Elements* (e.g. faders) allow to specify the minimum, maximum and default value.

Values

Default: 0

Min: 0

Max: 1

Figure 8.15: Values

Automation

Toggle When a button is selected, it is optional to convert it into a toggling button by selecting *Toggle Button*. When function is set to control, the button will send its on-value when toggled on, and off-value when toggled off. When using the set-function, it will send the same value in both cases.

Open Page In *Kiosc* you can use a button or an image to switch between layout pages. A page doesn't have to be set as *Visible* to be selectable and displayed via a button. This is really handy when you want some pages to be only available from a specific page.

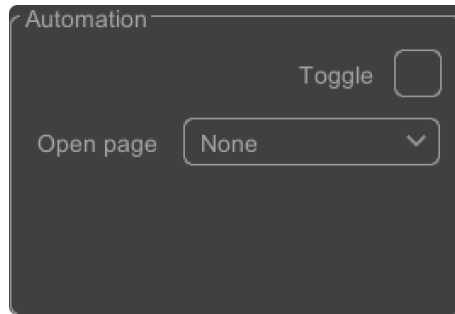


Figure 8.16: Automation Properties

8.3.2 Layout

This is where you can view and edit the currently selected page. After dragging in an element from the Elements menu on the right, it can be resized by selecting the element, and dragging the small square located at the corner of the element. It can be removed by dragging it back to the Elements menu or pressing Delete on your keyboard.

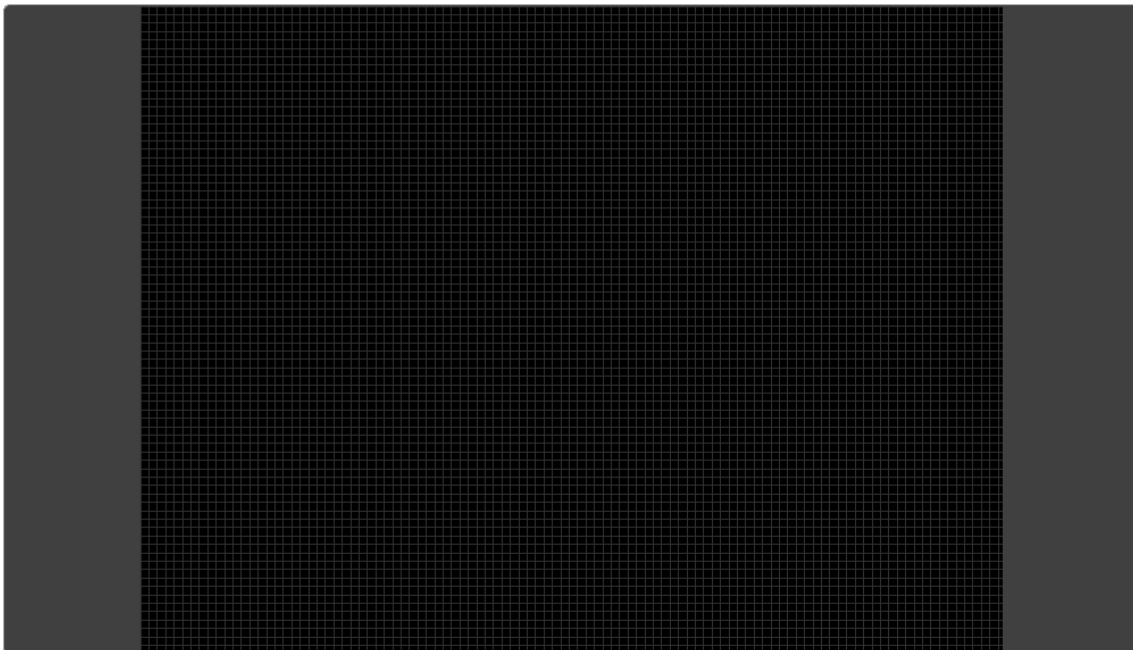


Figure 8.17: Layout

8.3.3 Options

The Editor only has a few options, see figure 8.18.

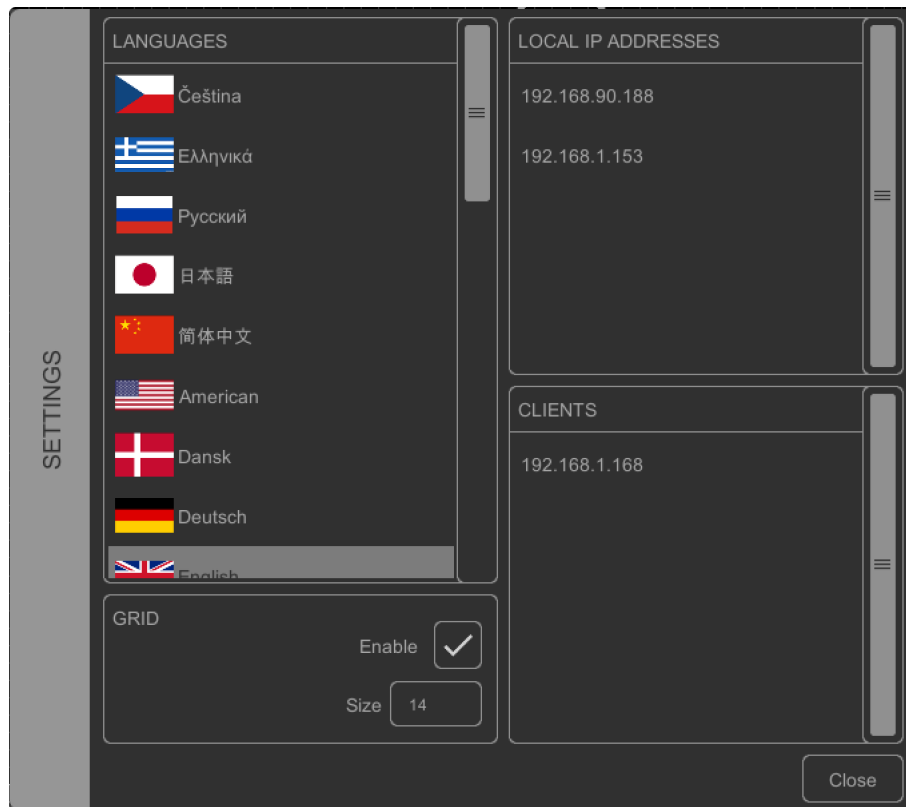


Figure 8.18: Options

Language

There are various languages available.

Grid Enable

Enable the *Grid* to have an aid in aligning Elements that will automatically snap to the closest grid connection. The size of the grid can be specified in the *size* field.

Local IP Addresses

A list with local IP addresses used by the computer running the KioskEditor. This information can be useful when troubleshooting network issues.

Clients

A list of all Kiosk clients currently connected to this Editor.

8.4 Feedback

Feedback in the Kiosk app is created by having the external equipment sending messages back to Kiosk. The feedback message should match the *Tag* of the *Element*. More information on which element supports which kind of feedback can be found in appendix A.

Chapter 9

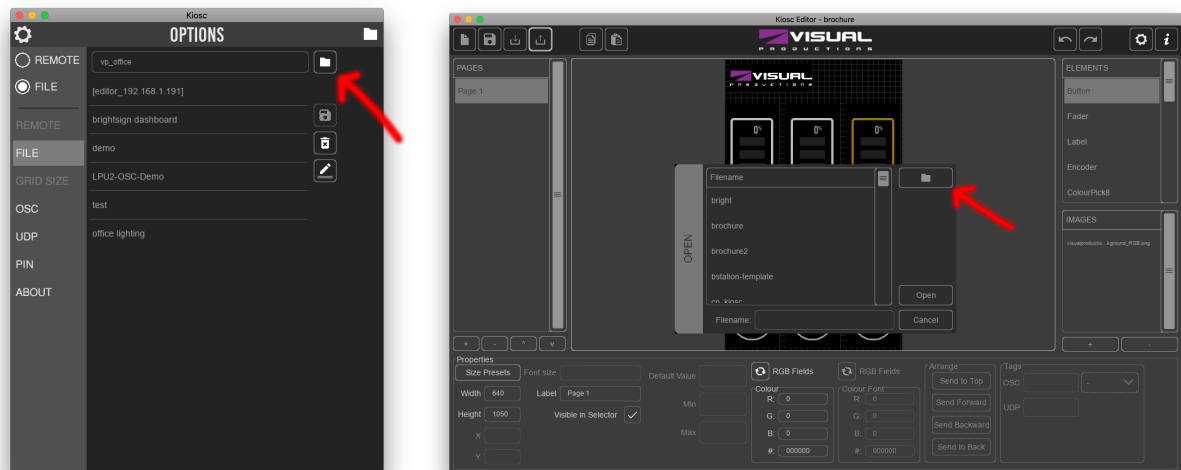
File Locations

When Kiosk is running in *Remote* mode there are no files involved, however, when running in *File* mode Kiosk must open a file that has been created by the Editor.

Apps distributed by app stores are not allowed to access files outside this designated location. Even apps made by the same developer like Kiosk and KioskEditor are not allowed to reach each other files.

Therefore, it is important to know where Kiosk and KioskEditor are storing their files, as you will need to transfer the Editor's file to the Kiosk app's location.

The designated file location differs per operating system and is likely to be a long and obscure path. For this reason, Kiosk and KioskEditor provide you with a shortcut to the correct file location. A *Folder* button can be found in the file related dialogs in both softwares. Clicking this button will open a file browser at the appropriate folder.



(a) Kiosk

(b) KioskEditor

Figure 9.1: Folder shortcut button

The layout file created by the Editor will have the `.kiosc` file extension.

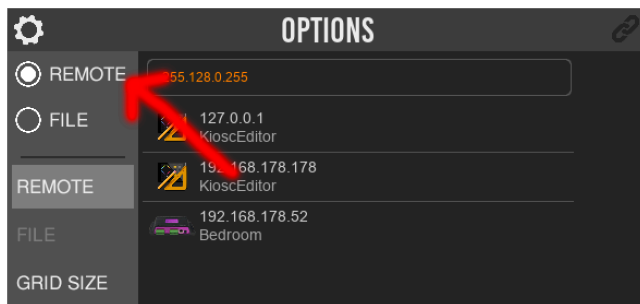
Transferring files from the KioskEditor running on a desktop OS to the Kiosk app running on a mobile OS can be done with a tool for copying files to the mobile device. E.g. Apple iTunes can

copy files to an iOS device.

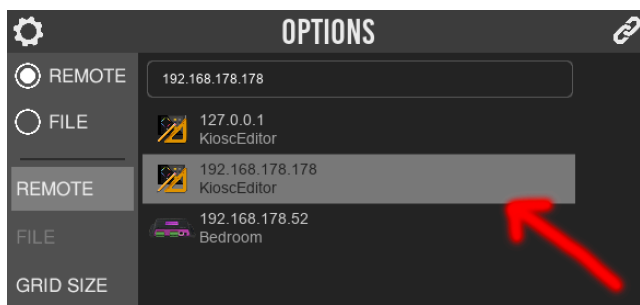
To make copying files to Kiosk on a mobile device more convenient, Kiosk is capable to connect to a KioskEditor under *Remote* mode and subsequently save it as a local file in *File* mode. Please note that due to Microsoft Store security measurements, it is not possible to make a remote connection between Kiosk and KioskEditor if Kiosk and KioskEditor are running on the same PC.

To store the showfile locally on the device running Kiosk, follow these steps:

1. Activate *Remote* mode.

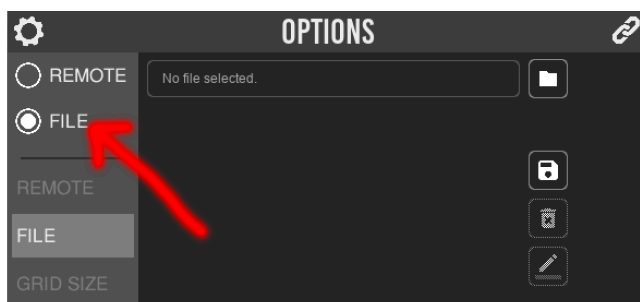


2. Connect to the Editor.

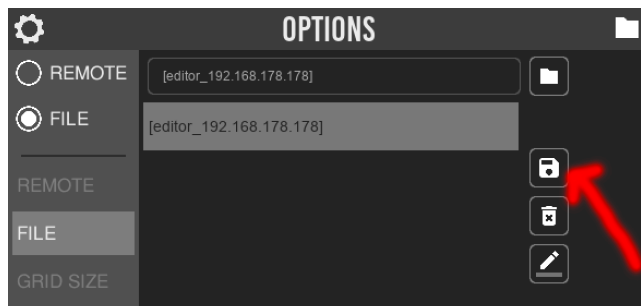


The layout is now loaded into Kiosk's memory.

3. Activate *File* mode.



4. Save the file by pressing the Save button.



Chapter 10

vManager

A free-of-charge software tool called vManager has been developed to manage the devices. vManager allows you to:

- Label your device with a distinguishable name
- Configure the IP address, subnet mask, router and DHCP
- Backup and restore the device's data and settings
- Perform firmware upgrades
- Identify a specific device (in a multi device set-up) by blinking its LED
- Revert to factory defaults

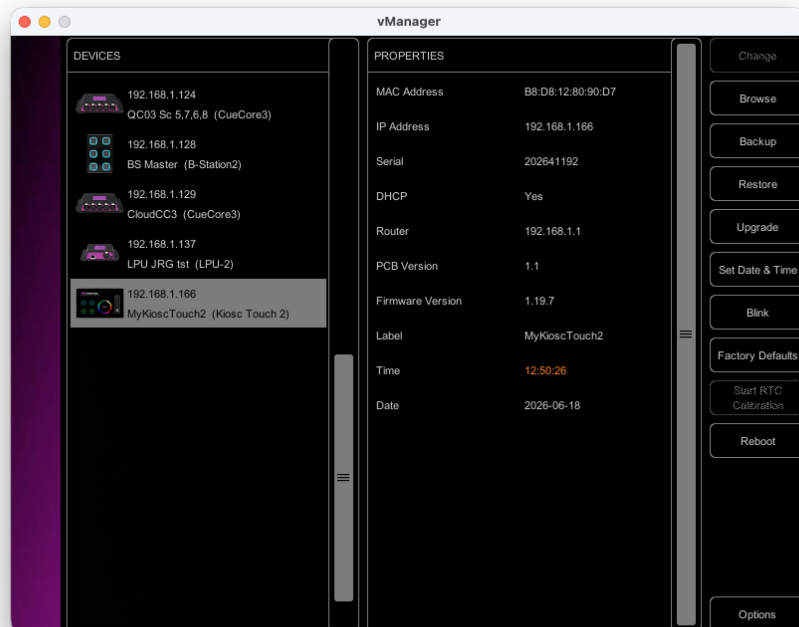


Figure 10.1: vManager

The following section explains the buttons in the vManager, as seen in figure 10.1.

10.1 Backup

Backups of all the programming data inside the KioscTouch2 can be made. This backup file is saved on the computer's hard-disk and can be easily transferred via e-mail or USB stick. The data of the backup can be restored via the *Restore* button.

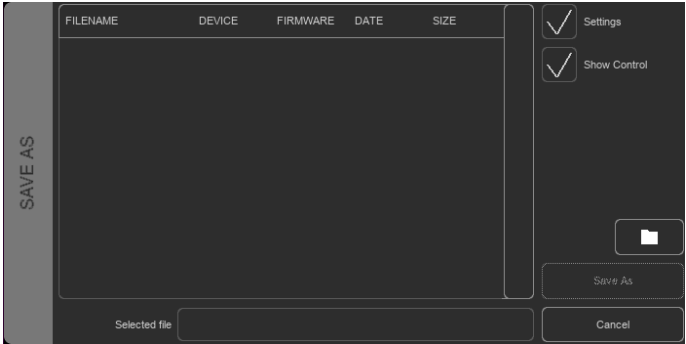


Figure 10.2: Creating a backup for the KioscTouch2

Apps distributed by app stores are not always allowed to access files outside their designated location. It is important to know where vManager is storing its files, in case you wish to transfer a backup file to a memory stick or dropbox.

This file location differs per operating system and can be a long and obscure path. For this reason, vManager provides you with a shortcut to the correct file location. A *Folder* button can be found in the file related dialogs. Clicking this button will open a file browser at the appropriate folder.

10.2 Upgrade Firmware

vManager will show the firmware version of a selected device in orange when a newer *Release* version is available. To upgrade the firmware, make sure the device is selected and press the *Upgrade* button. The dialogue allows for selecting from the list of available firmware versions. Firmware versions in the list will show whether they are stable *release* versions or *beta* with new and or experimental functionality. *Beta* versions are not recommended to be used in production environments.



Figure 10.3: Firmware upgrade



Important: Make sure the power to the device is not interrupted during the upgrade process.

10.3 Blink

The device's status LED can be set to *Blink* fast for about 2 seconds, to identify the specific unit amongst multiple devices. This blinking is enabled by double-clicking on a device in the Devices list or by selecting a device and then clicking the Blink button. Alternatively, browse to the Settings page of the device, and choose Blink in the top left. Now the device will blink as long as the slider is set to 'on'.

10.4 Factory Defaults

All the user data are stored in the on-board memory. They will be completely erased and all settings reverted to their defaults by pressing the *Factory Defaults* button. This action does not affect the device's IP settings.

10.5 Reboot

The *Reboot* button allows you to remotely restart the device. This is useful for testing the unit's behaviour after a power-cycle.

10.6 Installing vManager

The vManager app is available on the most common range of operating systems.

The software is distributed through app-stores to make receiving future updates automatic and very easy.

Note: vManager is no longer provided for mobile devices such as tablets or mobile phones.

10.6.1 Windows

Visit the Microsoft store at <https://www.microsoft.com/en-us/p/vmanager/9nblggh4s758>.

Windows 11 is required.

10.6.2 macOS

Visit the Apple macOS app store at <https://apps.apple.com/us/app/vmanager/id1074004019>.

macOS 26.5 is recommended.

10.6.3 Ubuntu

You can acquire the vManager from Snapcraft at <https://snapcraft.io/vmanager>.

Alternatively, it can be installed by using the command-line:

```
snap find vmanager  
snap install vmanager
```

To update the apps later on via the command-line type:

```
snap refresh vmanager
```

Ubuntu 26.04 LTS is recommended. The software is only available for the amd64 architecture.

Appendices

Appendix A

Elements

This chapter will give a more detailed explanation of each element's ability to send and receive various protocols.

A.1 Page

The Page is not a real element but can be used to send messages when displayed. It can also be displayed using feedback messages. The following sections will give a detailed explanation about the send and receive capabilities of a Page.

A.1.1 Send

OSC

Type	Function	Displayed
-	Active	true

UDP and TCP

Type	Function	Displayed	Example
-	Active	true	my-page=true

A.1.2 Feedback

Feedback can be sent to a page using its tag to directly display a specific page.

OSC

Type	Value	Controls
Bool	On	→ Display

UDP and TCP

Type	Value	Controls	Example
Bool	On	→ Display	page1=on

A.2 Button

The button element can be used to send various messages. It can be controlled using feedback messages. The following sections will give a detailed explanation about the send and receive capabilities of the Button element.

A.2.1 Send

OSC

Type	Function	Property	Value Down	Value Up
Bool	Control	-	On	Off
Bool	Control	Toggle	On	Off
Integer	Control	-	1	0
Integer	Control	Toggle	1	0
Integer	Set	-	<Number>	-
Integer	Set	Toggle	<Number>	<Number>

UDP and TCP

Type	Function	Property	Value Down	Value Up	Example
None	-	-	Tag only	-	my-button
None	-	Toggle	Tag only	Tag only	my-button
Bool	Control	-	On	Off	my-button=On
Bool	Control	Toggle	On	Off	my-button=On
Integer	Control	-	1	0	my-button=1
Integer	Control	Toggle	1	0	my-button=1
Integer	Set	-	<Number>	-	my-button=23
Integer	Set	Toggle	<Number>	<Number>	my-button=23

A.2.2 Feedback

Feedback can be sent to a button using its tag. The type of message determines which attribute of the button it changes. The following attributes of a button can be set using feedback:

- **Visual state**

The visual state of a button can be used as feedback to the user, indicating for example if a light or cue is active. It is much like having an LED light inside a physical button. When enabled, either via a feedback message or by pressing a button with the special property "Toggle", the inside of the button will be filled.

- **Label**

The label is a string of text displayed inside the button. The default value is set in the showfile, but it can be temporarily changed using a feedback message. Reloading the showfile will return the button to its default label.

- **Colour**

The default colour of the button is set in the showfile, but it can be temporarily changed using a feedback message. Reloading the showfile will return the button to its default colour.

OSC

Type	Value	Controls	
Bool	On/Off	→	State
Integer	1/0	→	State
String	<text>	→	Label
Colour	<colour>	→	Colour

UDP and TCP

Type	Value	Controls		Example
Bool	On/Off	→	State	my-button=on
Integer	1/0	→	State	my-button=1
String	<text>	→	Label	my-button=text
Colour	<HEX RGB colour>	→	Colour	my-button=#FF00FF

A.3 CheckBox

The check box can be used to sent various messages. It can be controlled using feedback messages. The following sections will give a detailed explanation about the send and receive capabilities of a check box.

A.3.1 Send

OSC

Type	Function	Value Checked	Value Unchecked
Bool	Control	On	Off
Integer	Control	1	0

UDP and TCP

Type	Function	Value Checked	Value Unchecked	Example
Bool	Control	On	Off	my-check=On
Integer	Control	1	0	my-check=1

A.3.2 Feedback

Feedback can be sent to a check box using its tag. The type of message determines which attribute of the check box it changes. The following attributes of a check box can be set using feedback:

- **State**

The state of a check box can be used as visual feedback to the user, indicating for example if a light or cue is active. When enabled, either via a feedback message or by pressing check box area, the inside of the box will be filled.

- **Label**

The label is a string of text displayed close to the check box. The default value is set in the showfile, but it can be temporarily changed using a feedback message. Reloading the showfile will return the check box to its default label.

- **Colour**

The default colour of the check box is set in the showfile, but it can be temporarily changed using a feedback message. Reloading the showfile will return the check box to its default colour.

OSC

Type	Value	Controls
Bool	On/Off	→ State
Integer	1/0	→ State
String	text	→ Label
Colour	<colour>	→ Colour

UDP and TCP

Type	Value	Controls	Example
Bool	On/Off	→ State	my-check=on
Integer	1/0	→ State	my-check=1
String	text	→ Label	my-check=awesome
Colour	<HEX RGB colour>	→ Colour	my-check=#FF00FF

A.4 Fader

The fader element can be used to sent various control values. It can be controlled using feedback messages. The following sections will give a detailed explanation about the send and receive capabilities of the fader element.

A.4.1 Send

OSC

Type	Function	Value	Range
Float	Control	<decimal number>	Min-Max
Integer	Control	<whole number>	Min-Max

UDP and TCP

Type	Function	Value	Range	Example
Float	Control	<decimal number>	Min-Max	my-fader=0.50
Integer	Control	<whole number>	Min-Max	my-fader=127

A.4.2 Feedback

Feedback can be sent to a fader using its tag. The type of message determines which attribute of the fader it changes. The following attributes of a fader can be set using feedback:

- **Value**

The current value of a fader can be set by a feedback message, for example to indicate the current intensity of a playback. The fader responds to feedback matching its type, and it takes into account the min and max range set in the showfile.

- **Label**

The label is a string of text displayed inside the fader. The default value is set in the showfile, but it can be temporarily changed using a feedback message. Reloading the showfile will return the fader to its default label.

- **Colour**

The default colour of the fader is set in the showfile, but it can be temporarily changed using a feedback message. Reloading the showfile will return the fader to its default colour.

OSC

Type	Value	Controls
Float	Min/Max	→ Value
Integer	Min/Max	→ Value
String	<text>	→ Label
Colour	<colour>	→ Colour

UDP and TCP

Type	Value	Controls	Example
Float	Min/Max	→ Value	my-fader=0.75
Integer	Min/Max	→ Value	my-fader=190
String	<text>	→ Label	my-fader=text
Colour	<HEX RGB colour>	→ Colour	my-fader=#FF00FF

A.5 Encoder

The encoder element can be used to sent various control values. It can be controlled using feedback messages. The following sections will give a detailed explanation about the send and receive capabilities of the encoder element.

A.5.1 Send

OSC

Type	Function	Value	Range
Float	Control	<decimal number>	Min-Max
Integer	Control	<whole number>	Min-Max

UDP and TCP

Type	Function	Value	Range	Example
Float	Control	<decimal number>	Min-Max	my-encoder=0.50
Integer	Control	<whole number>	Min-Max	my-encoder=127

A.5.2 Feedback

Feedback can be sent to an encoder using its tag. The type of message determines which attribute of the fader it changes. The following attributes of an encoder can be set using feedback:

- **Value**

The current value of an encoder can be set by a feedback message, for example to indicate the current intensity of a playback. The encoder responds to feedback matching its type, and it takes into account the min and max range set in the showfile.

- **Label**

The label is a string of text displayed inside the encoder. The default value is set in the showfile, but it can be temporarily changed using a feedback message. Reloading the showfile will return the encoder to its default label.

- **Colour**

The default colour of the encoder is set in the showfile, but it can be temporarily changed using a feedback message. Reloading the showfile will return the encoder to its default colour.

OSC

Type	Value	Controls
Float	Min/Max	→ Value
Integer	Min/Max	→ Value
String	<text>	→ Label
Colour	<colour>	→ Colour

UDP and TCP

Type	Value	Controls	Example
Float	Min/Max	→ Value	my-encoder=0.75
Integer	Min/Max	→ Value	my-encoder=190
String	<text>	→ Label	my-encoder=text
Colour	<HEX RGB colour>	→ Colour	my-encoder=#FF00FF

A.6 Axis

The axis element can be used to sent position values in a 2 float number between -1.00 and 1.00 separated by a coma (,) format. It can be controlled using feedback messages. The following sections will give a detailed explanation about the send and receive capabilities of the axis element.

A.6.1 Send

OSC

Type	Function	Value	Range
Float	Control	<decimal number, decimal number>	(-1.00,-1.00) - (1.00,1.00)

UDP and TCP

Type	Function	Value	Range	Example
Float	Control	<dec. num., dec. num.>	(-1.00,-1.00) - (1.00,1.00)	axis=0.50,-0.46

A.6.2 Feedback

Feedback can be sent to an axis using its tag. The type of message determines which attribute of the fader it changes. The following attributes of an encoder can be set using feedback:

- **Value**
The current value of an axis can be set by a feedback message, for example to indicate the current position of a moving head. The axis responds to feedback matching its type.
- **Colour**
The default colour of the axis is set in the showfile, but it can be temporarily changed using a feedback message. Reloading the showfile will return the axis to its default colour.

OSC

Type	Value	Controls
Float	0.00,0.00	→ Value
Colour	<colour>	→ Colour

UDP and TCP

Type	Value	Controls		Example
Float	0.00,0.00	→	Value	my-axis=0.75,-0,27
Colour	<HEX RGB colour>	→	Colour	my-axis=#FF00FF

A.7 Text

The Text element can be used to send text messages. It can be controlled using feedback messages. The following sections will give a detailed explanation about the send and receive capabilities of the Text element.

A.7.1 Send

OSC

Type	Function	Value
String	set	<text>

UDP and TCP

Type	Function	Value	Example
String	Set	<text>	my-text=awesome

A.7.2 Feedback

Feedback can be sent to a text element using its tag. The type of message determines which attribute of the text element it changes. The following attributes of a text element can be set using feedback:

- **Text**

The displayed text of a Text element can be set by a feedback message, for example to indicate the name of the current light scene running.

- **Colour**

The default colour of the Text element is set in the showfile, but it can be temporarily changed using a feedback message. Reloading the showfile will return the text element to its default colour.

OSC

Type	Value	Controls
String	<text>	→ text
Colour	<colour>	→ Colour

UDP and TCP

Type	Value	Controls		Example
String	<text>	→	Label	my-text=awesome
Colour	<HEX RGB colour>	→	Colour	my-text=#FF00FF

A.8 Indicator

The indicator element can be seen as a virtual LED. It can only be used for visual feedback, it does not send any message. The indicator can be changed using feedback messages. The following sections will give a detailed explanation about the receive capabilities of the indicator element.

A.8.1 Feedback

Feedback can be sent to an indicator using its tag. The type of message determines which attribute of the indicator it changes. The following attributes of an indicator can be set using feedback:

- **State**

The state of an indicator can be used as visual feedback to the user, indicating for example if a light or cue is active. It is much like having an LED light. By default the indicator state is off and the colour of the indicator is dimmed. When enabled via a feedback message the colour inside of the indicator will be brightened.

- **Colour**

The default colour of the indicator is set in the showfile, but it can be temporarily changed using a feedback message. Reloading the showfile will return the indicator to its default colour.

OSC

Type	Value	Controls
Bool	On/Off	→ State
Integer	0 or lower	→ State (Off)
Integer	1 or higher	→ State (On)
Float	0 or lower	→ State (Off)
Float	0.001 or higher	→ State (On)
Colour	<colour>	→ Colour

UDP and TCP

Type	Value	Controls	Example
Bool	On/Off	→ State	my-indicator=on
Integer	0 or lower	→ State (Off)	my-indicator=-10
Integer	1 or higher	→ State (On)	my-indicator=1
Float	0 or lower	→ State (Off)	my-indicator=0
Float	0.001 or higher	→ State (On)	my-indicator=0.5
Colour	<HEX RGB colour>	→ Colour	my-indicator=#FF00FF

A.9 Image

The image element can be used to send various messages. It can be controlled using feedback messages. The following sections will give a detailed explanation about the send and receive capabilities of the image element.

A.9.1 Send

OSC

Type	Function	Property	Value Down	Value Up
Bool	Control	-	On	Off
Bool	Control	Toggle	On	Off
Integer	Control	-	1	0
Integer	Control	Toggle	1	0
Integer	Set	-	<Number>	-
Integer	Set	Toggle	<Number>	<Number>

UDP and TCP

Type	Function	Property	Value Down	Value Up	Example
None	-	-	Tag only	-	my-image
None	-	Toggle	Tag only	Tag only	my-image
Bool	Control	-	On	Off	my-image=On
Bool	Control	Toggle	On	Off	my-image=On
Integer	Control	-	1	0	my-image=1
Integer	Control	Toggle	1	0	my-image=1
Integer	Set	-	<Number>	-	my-image=23
Integer	Set	Toggle	<Number>	<Number>	my-image=23

A.9.2 Feedback

Feedback can be sent to an image using its tag. The type of message determines which attribute of the button it changes. The following attributes of an image can be set using feedback:

- **Alpha**

The default alpha of the image is set to 100%, but it can be temporarily changed using a feedback message. Reloading the showfile will return the image to its default alpha.

OSC

Type	Value	Controls
Bool	On/Off	→ Alpha
Float	1.00/0.00	→ Alpha

UDP and TCP

Type	Value	Controls	Example
Bool	On/Off	→ Alpha	my-image=on
Float	1.00/0.00	→ Alpha	my-image=0.5

A.10 Label

The label element can only be used for feedback, it does not send any message. The label can be changed using feedback messages. The following sections will give a detailed explanation about the receive capabilities of the label element.

A.10.1 Feedback

Feedback can be sent to a label using its tag. The type of message determines which attribute of the label it changes. The following attributes of a label can be set using feedback:

- **Label**

The label is a string of text. The default value is set in the showfile, but it can be temporarily changed using a feedback message. Reloading the showfile will return the default text to the label element.

- **Colour**

The default font colour of the label is set in the showfile, but it can be temporarily changed using a feedback message. Reloading the showfile will return the label to its default colour.

OSC

Type	Value	Controls
String	<text>	→ Label
Colour	<colour>	→ Colour

UDP and TCP

Type	Value	Controls	Example
String	<text>	→ Label	my-label=awesome
Colour	<HEX RGB colour>	→ Colour	my-label=#800080

A.11 ColorSpectrum

The spectrum elements can send color value on an hex format. The following sections will give a detailed explanation about the receive capabilities of the Color Spectrum element.

A.11.1 Send

OSC

Type	Function	Value	Range
Colour	Control	<HEX RGB colour>	#000000-#FFFFFF

UDP and TCP

Type	Function	Value	Range	Example
String	Control	<HEX RGB colour>	#000000-#FFFFFF	my-color=#800080

A.11.2 Feedback

Feedback can be sent to a ColorSpectrum using its tag.

- **Colour**

The color value can be set via feedback if the ColorSpectrum element has such value. Check available colors at A.16.

OSC

Type	Value	Controls
Colour	Control	<HEX RGBA colour> #00000000-#FFFFFFFF

UDP and TCP

Type	Value	Controls	Example
Colour	<HEX RGB colour>	→ Colour	my-spectrum=#800080

A.12 Color Pickers: ColorPicker8, ColorPicker16, WheelColor

The color picker elements can send color value on an hex format. The label can be changed using feedback messages. The following sections will give a detailed explanation about the receive capabilities of the color picker elements.

A.12.1 Send

OSC

Type	Function	Value	Range
Colour	Control	<HEX RGBA colour>	#000000FF-#FFFFFFF

UDP and TCP

Type	Function	Value	Range	Example
String	Control	<HEX RGB colour>	#000000-#FFFFFF	my-color=#800080

A.12.2 Feedback

Feedback can be sent to a color picker using its tag.

- **Label**

The label is a string of text. The default value is set in the showfile, but it can be temporarily changed using a feedback message. Reloading the showfile will return the default text to the color picker element.

- **Colour**

The color value can be set via feedback if the Color Picker element has such value. Check available colors at A.16.

OSC

Type	Value	Controls
String	<text>	→ Label
Colour	Control	<HEX RGBA colour> #00000000-#FFFFFFF

UDP and TCP

Type	Value	Controls	Example
String	<text>	→ Label	my-colorpicker=awesome
Colour	<HEX RGB colour>	→ Colour	my-colorpicker=#800080

A.13 WheelRGBW

The WheelRGBW elements can send color value on an hex format. The label can be changed using feedback messages. The following sections will give a detailed explanation about the receive

capabilities of the WheelRGBW element.

A.13.1 Send

OSC

Type	Function	Value	Range
Colour	Control	<HEX RGBA colour>	#00000000-#FFFFFFF

UDP and TCP

Type	Function	Value	Range	Example
Colour	<HEX RGB colour>	→	Colour	my-wheel=#800080

A.13.2 Feedback

Feedback can be sent to a WheelRGBW element using its tag.

- **Label**

The label is a string of text. The default value is set in the showfile, but it can be temporarily changed using a feedback message. Reloading the showfile will return the default text to the color picker element.

- **Colour**

The color value can be set via feedback if the WheelRGBW element has such value. Check available colors at A.16.

OSC

Type	Value	Controls
String	<text>	→ Label
Colour	Control	<HEX RGBA colour> #00000000-#FFFFFFF

UDP and TCP

Type	Value	Controls	Example
String	<text>	→ Label	my-wheelrgbw=awesome

A.14 WheelColTemp

The WheelColTemp elements can send color temperature ratio between warm white and cold white on an float format. It can be controlled using feedback messages. The following sections will give a detailed explanation about the send and receive capabilities of the WheelColTemp element.

A.14.1 Send

OSC

Type	Function	Value	Range
Float	Control	<decimal number>	0.00-1.00

UDP and TCP

Type	Function	Value	Range	Example
Float	Control	<decimal number>	0.0-1.0	my-temp=0,82

A.14.2 Feedback

Feedback can be sent to a WheelRGBW element using its tag. The label of the element can be set using feedback:

- **Label**

The label is a string of text. The default value is set in the showfile, but it can be temporarily changed using a feedback message. Reloading the showfile will return the default text to the color picker element.

OSC

Type	Value	Controls
String	<text>	→ Label

UDP and TCP

Type	Value	Controls	Example
String	<text>	→ Label	my-wheelrgbw=awesome

A.15 Date, Clock and Line

The Date, Clock and Line elements can only be used for feedback, it does not send any message. The color can be changed using feedback messages. The following sections will give a detailed explanation about the receive capabilities of these elements.

A.15.1 Feedback

Feedback can be sent to a Date, Clock or Line element using its tag.

- **Colour**

The default font colour of a Date, Clock or Line element is set in the showfile, but it can be temporarily changed using a feedback message. Reloading the showfile will return the element to its default colour.

OSC

Type	Value	Controls
Colour	<colour>	→ Colour

UDP and TCP

Type	Value	Controls	Example
Colour	<HEX RGB colour>	→ Colour	my-date=#800080

A.16 Color elements values

A.16.1 ColorSpectrum

Any color with RGB channels is available:

- #XXXXXX (with X from 0 to F).

A.16.2 Color Pickers

ColorPicker8

The following colors are available:

 #FF0000	 #FF00FF	 #7F007F	 #0000FF
 #00FFFF	 #00FF00	 #FFFF00	 #FF7F00

ColorPicker16

The following colors are available:

 #FF0000	 #7F0000	 #FF007F	 #FF00FF
 #7F007F	 #00007F	 #0000FF	 #007FFF
 #00FFFF	 #00FF00	 #007F00	 #007F7F
 #7F7F00	 #FFFFFF	 #FFFF00	 #FF7F00

WheelColor

Any color with one of the channels at 0 is available:

- #00XXXX (with X from 0 to F).
- #XX00XX (with X from 0 to F).
- #XXXX00 (with X from 0 to F).

A.16.3 WheelRGBW

Any color with RGBW channels is available:

- #XXXXXXXX (with X from 0 to F).

Appendix B

API

This chapter will give the list of all the API commands you can send to Kiosc and what will be its feedback. The API commands are only compatible with *KioscTouch2*.

The API prefix is configurable and can be viewed or modified on the *Settings* page. All examples assume the default API prefix *kiosc*.

For OSC communication, all commands must be preceded by a leading forward slash (/). For UDP and TCP communication, forward slashes (/) are replaced by hyphens (-).

B.1 OSC

Open Sound Control (OSC) is the preferred protocol for integrating with the KioscTouch2, as its strongly typed message structure provides greater consistency, reliability and interoperability than plain text-based protocols.

URI	Parameter	Description
/prefix/hello	-	The unit will reply with the same Hello message
/prefix/goodbye	-	The unit will erase the IP from the feedback list
/prefix/blink	-	Momentarily flashes the KioscTouch2's header
/prefix/page/select	<integer>	Switch to page index of the layout

B.2 UDP and TCP

For TCP communication, the KioscTouch2 determines the end of a message by detecting a carriage return and line feed sequence (`\r\n`).

Depending on the operating system, programming language, or software used, these characters may be represented or encoded differently.

Some TCP clients automatically append the required end-of-line characters to transmitted mes-

sages, while others require them to be included explicitly. Refer to the documentation of your TCP client or control system to determine how to append the `\r\n` sequence to outgoing messages.

UDP messages do not require message termination characters, as each UDP packet is received as a complete message.

URI	Parameter	Description
prefix-hello	-	The unit will reply with the same Hello message
prefix-goodbye	-	The unit will erase the IP from the feedback list
prefix-blink	-	Momentarily flashes the KioscTouch2's header
prefix-page-select=	<integer>	Switch to page index of the layout

B.3 Feedback

The KioscTouch2 is able to send feedback to external equipment using its API, so called 'clients'. The KioscTouch2 keeps a memory of the last eight OSC clients, the last eight UDP clients and last eight TCP clients.

The API's feedback message will only be sent to the client list of the protocol used to send the command.

For example, if you send an API command via OSC, only the clients registered in the OSC clients list will receive the feedback message. Below is a table listing the messages the KioscTouch2 will send back to its clients.

OSC	UDP/TCP
/prefix/hello	prefix-hello

The `hello` command is ideal for polling the device; it allows you to verify that the KioscTouch2 is online at the IP address and port that you expect.

A power-cycle will clear the internal client lists. Send `/kiosc/goodbye` or `kiosc-goodbye` to explicitly be removed from the client list.

B.3.1 Preventing a feedback loop

Feedback is automatically sent to a device which uses the OSC, UDP or TCP API. If the external device is also a Visual Productions unit then the feedback message could be interpreted by the external unit a new command. This can result in another feedback message being generated. An endless stream of feedback messages can stall the units involved.

This feedback loop can be prevented by assign a unique label the device's API prefix. This setting is discussed on page 25.

Appendix C

Certifications

This appendix contains the declarations of conformity and regulatory certifications applicable to the KioscTouch2.



EU Declaration of Conformity

We, **Visual Productions BV**, as the manufacturer, hereby declare under our sole responsibility that the following device:

Product Name: KioscTouch2
Product Type: Touch Screen

complies with the requirements of the following directives and standards:

Applicable Directives:

- 2014/30/EU – Electromagnetic Compatibility (EMC)
- 2011/65/EU (as amended by 2015/863) – Restriction of Hazardous Substances (RoHS)

Applied Harmonised Standards:

- EN 61000-6-1:2019 – Electromagnetic compatibility (EMC) – Part 6-1: Generic standards – Immunity for residential, commercial, and light-industrial environments
- EN 63000:2018 – Technical documentation for the assessment of electrical and electronic products concerning the restriction of hazardous substances

This declaration is issued under the sole responsibility of the manufacturer, confirming that the object of the declaration complies with the relevant Union harmonisation legislation.

Authorised Representative:

Full name and identification of the person responsible for product quality and compliance with standards on behalf of the manufacturer:

Date: March 28, 2025
Place: Haarlem, The Netherlands



Ing. Maarten Engels
Managing Director
Visual Productions BV

Index

- 2FA, 42
- About, 28
- About page, 37
- Admin, 43
- Android, 30
- API, 96
- API prefix, 25, 97
- Appearance, Properties, 56
- Arrange, 58
- Axis element, 53, 82
- Backing up your data, 69
- Backup, 68
- Backups, 46
- Behaviour, Properties, 59
- Blink, 70
- Box contents, 7
- Bracket, 9
- Button element, 52, 74
- CE, 6, 100
- Certifications, 98
- CheckBox element, 77
- Checkbox element, 55
- Clients, 63
- Clock element, 55, 94
- Cloud, 40
- Cloud ID, 43
- ColorPicker16 element, 90
- ColorPicker8 element, 90
- ColorSpectrum element, 89
- Colourpicker elements, 54
- Colours, 58
- ColourSpectrum element, 54
- Copyright, 3
- Date element, 55, 94
- Device Label, 68
- Device Management, 68
- Device Name, 68
- Devices, 45
- DHCP, 11
- Disposal, 8
- Downloads, 29, 49
- Dummy, 15
- Editor, 49
- Elements, 51, 73
- Encoder element, 53, 80
- Engineer, 43
- Factory Defaults, 70
- Fader element, 53, 78
- Feedback, 14, 15, 64, 73, 80, 97
- File, 15, 18
- File Locations, 65
- File page, 34
- Firmware, 68
- Firmware upgrade, 69
- Flip, 59
- Forum, 8
- Function, 59
- Grid Enable, 63
- Grid Size, 15, 19
- Hex, 60
- Home page, 32
- Image element, 55, 87
- Images, 56
- Incoming port, 21–23
- Indicator element, 52, 86
- Individual, 60
- Installing vManager, 70
- iOS, 30
- IP, 26
- IP Address, 11, 63, 68
- iTunes, 65
- Key Features, 6
- Kiosc, 13
- Label, 68
- Label element, 53, 89
- Language, 63
- LED, 11
- Line element, 56, 94
- Local IP Addresses, 63

macOS, 30, 49, 70
Management, 68
Members, 42
Menu bar, 50
Modes, 13, 18
Monitor page, 33

Name, 68
Network Address, 11
Network Configuration, 68

Open page, 61
Options, 17, 62
OSC, 16, 20, 59, 96

Page element, 74
Pages, 51
Partner ID, 43
Percentage, 59
Pin Protect, 59
PoE, 6
Prefix, 25, 97
Projects, 44
Properties, 59
Protocols, 16
PurpleCloud, 40

Reboot, 70
Remote, 13, 18
Restore a Backup, 68

Safety precautions, 4
Screen Size, 57
Settings, 24
Settings page, 35
Size Presets, 57
Snap, 31, 50, 71
Static IP, 11
Status LED, 11
Subnet mask, 11
Subscription, 44
Support Partner, 43

Tags, 59, 60
Target, 60
Target page, 33
TCP, 17, 22, 59, 60
Text element, 55, 84
Toggle, 61
Type, 59

Ubuntu, 31, 50, 71
UDP, 16, 21, 59
UDP and TCP, 96

UKCA, 6
Unpacking, 7
Upgrade, 68
User, 43

Value, 59
Values, 61
Version, 28
Visible, 59
Visible in selector, 58
VisualTouch, 14
vManager, 11, 68

Web Interface, 32
Wheel elements, 54
WheelColor element, 90
WheelColTemp element, 93
WheelRGBW element, 91
Windows, 30, 49, 70