KIOSC MANUAL

WWW.VISUALPRODUCTIONS.NL

Revision History

Revision	**Date**	**Author(s)**	**Description**
1	07.06.2018	ME	Initial version.
2	14.06.2019	BZ	Added Kiosc Touch chapter, added info on elements, expanded the index section
3	01.02.2021	BZ	Added Elements appendix

Contents

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.

Chapter 1

Introduction

Thank you for choosing Kiosc. Kiosc (see figure 1.1) is a simple app for creating custom user-interfaces. These user-interfaces are typically used for controlling lighting and AV equipment. Kiosc communicates with the external equipment via OSC and UDP; protocols that run over the Ethernet network.

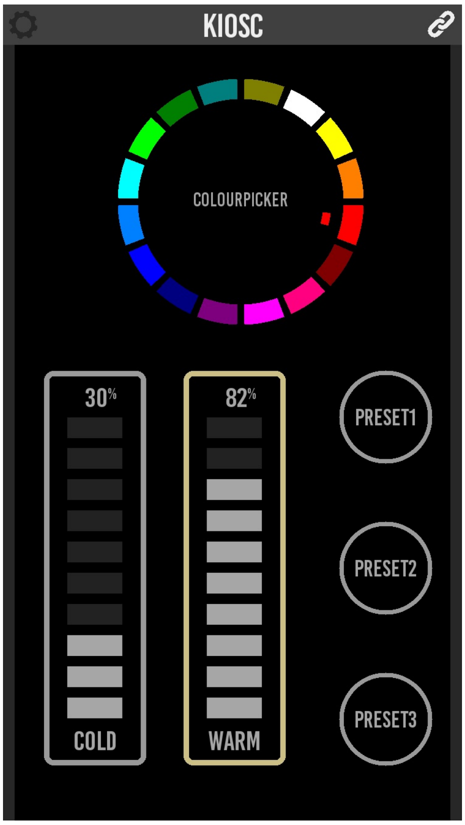


Figure 1.1: Kiosc

Kiosc is accompanied by the Kiosc Editor. This software tool enables you to design a simple control dashboard in a few minutes. While it also carries the features required to create complex multi-page interactive user-interfaces. Importantly, the Kiosc Editor allows you to personalise the dashboard; you are free to add logos, images and change colours.

The Kiosc app works well with the Visual Productions range of lighting controllers, in fact, it is compatible with any OSC and/or UDP capable 3rd party device or software.

Kiosc is available on many operating systems: iOS, Android, Windows, macOS & Ubuntu Linux. The Kiosc Editor is available on Windows, macOS & Ubuntu.

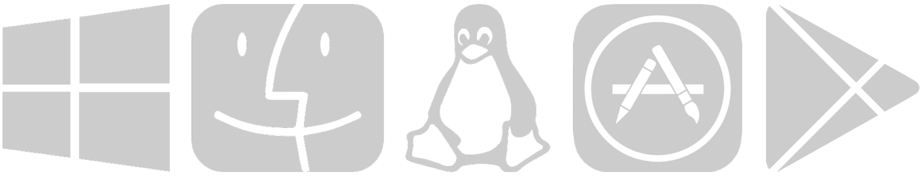


Figure 1.2: Platforms

This manual covers the functionality of the Kiosc app in chapter 2. The Kiosc Touch device is discussed in Chapter 3. Chapter 4 explains how to design custom layouts by using the Kiosc Editor. Finally, chapter 5 discusses the locations where the apps save and load the files on disk.

At the time of writing this manual the Kiosc Touch, App and Editor are at version 1.9.5.

1.1 Features

Kiosc has the following features.

- Generic control elements like buttons, sliders and check-boxes
- Lighting control elements like colour wheels and spectrums
- Feedback
- Multiple pages
- Available on 5 operating systems
- OSC and UDP
- PIN lock

1.2 Compatibility

Kiosc is compatible with the following devices.

- Visual Productions DaliCore
- Visual Productions CueCore2
- Visual Productions CueCore1
- Visual Productions QuadCore
- Visual Productions IoCore2
- Visual Productions IoCore1
- Visual Productions Cuety LPU-2
- Visual Productions B-Station2
- Visual Productions B-Station1
- Visual Productions TimeCore
- Any OSC or UDP compatible device

1.3 Further Help

If, after reading this manual, you have further questions then please consult the online forum at `http://forum.visualproductions.nl/` for more technical support.

Chapter 2

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.

2.1 Modes

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

2.1.1 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 Kiosc Editor and receive live updates as the user works on his 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 to make a remote connection to Kiosc Editor if Kiosc and Kiosc Editor are running on the same PC. To transfer a showfile, it can be manually copied using the filebrowser. More information can be found in chapter 5.

Although quick to set up, a *Remote* connection to a CueCore2 or similar device does not enable to user to make elaborate designs. The editing capabilities inside the CueCore2 are basic. In order to create more demanding user-interface designs it is recommended to use the *File* mode.

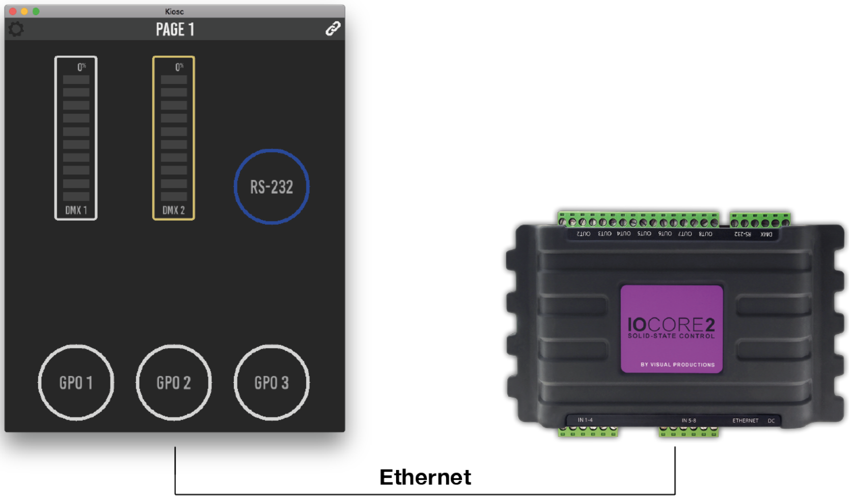
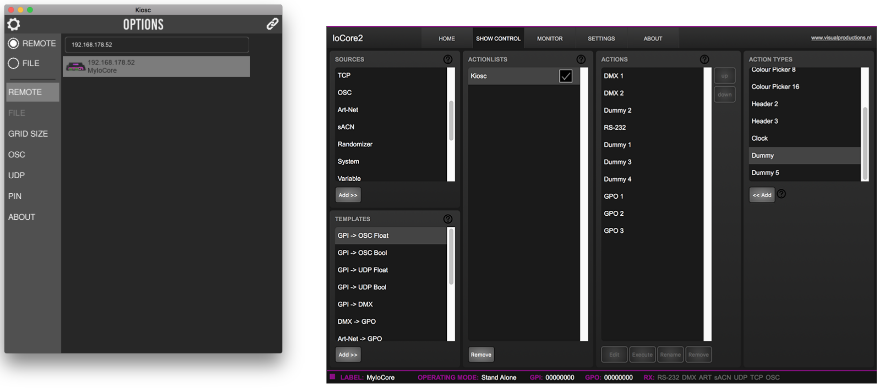


Figure 2.1: Kiosc connected to a IoCore2

Kiosc can only be connected to a single external device when using *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 2.1 shows an example of Kiosc being connected to a 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.



192.168.178.52

192.168.178.52
MyIoCore

LABEL: MyIoCore OPERATING MODE: Stand Alone GPI: 00000000 GPO: 00000000 RX: RS-232 DMX ART sACN UDP TCP OSC

(a) Selecting the IoCore2 remote server

(b) Actions inside the IoCore2

Figure 2.2: Connecting Kiosc to an IoCore2

Figure 2.2 shows how to select the IoCore2 in the remote options. Once connected, the dashboard is created by adding a 'Kiosc' actionlist inside the IoCore'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 then fits 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 2.2 displays the actions created for this example. Programming actions is discussed in more detail in the IoCore2's user manual.

2.1.2 File

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

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

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

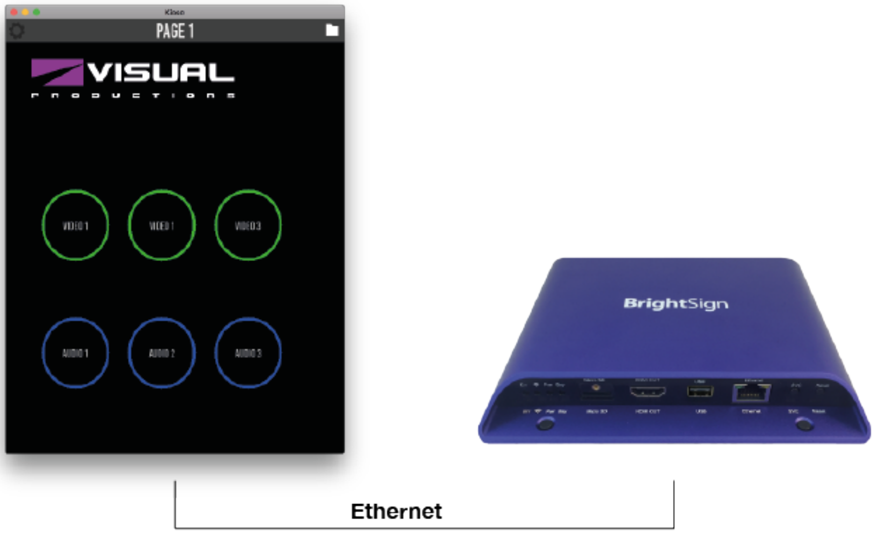


Figure 2.3: Kiosc connected to a media player

Figure 2.3 shows an example of Kiosc controlling a Brightsign, a 3rd party media player. Kiosc is using UDP messages to trigger the playback of different media files inside the Brightsign.

2.2 Protocols

Kiosc can communicate with external equipment via two protocols, OSC and UDP. Both protocols run over the Ethernet network. This chapter describes these protocols.

2.2.1 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.

2.2.2 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.

2.2.3 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 the UDP capable equipment from two different vendors will be able 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.

2.3 Options

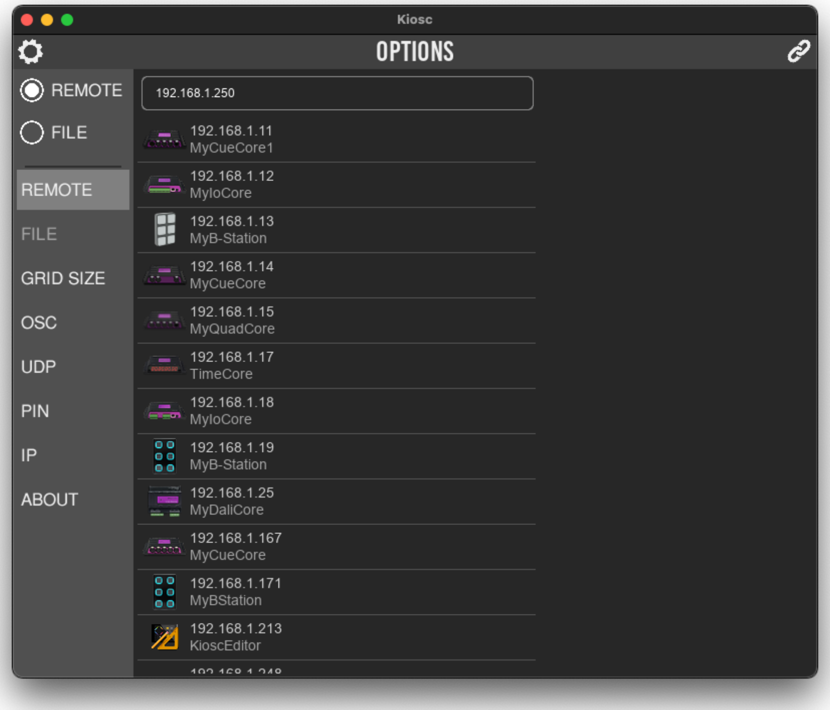
The options screen presents a few settings for customising the Kiosc behaviour.

2.3.1 Modes

The modes selector allows you to choose between *Remote* and *File* mode. Modes are discussed on page 8.

2.3.2 Remote

This menu (see figure 2.4) is only enabled when in *Remote* mode. It allows you to select the remote server. E.g. a CueCore or Kiosc Editor. Please note that due to Microsoft Store security measurements, it is not possible to to make a remote connection to Kiosc Editor if Kiosc and Kiosc Editor are running on the same PC.



192.168.1.250

192.168.1.11
MyCueCore1

192.168.1.12
MyIoCore

192.168.1.13
MyB-Station

192.168.1.14
MyCueCore

192.168.1.15
MyQuadCore

192.168.1.17
TimeCore

192.168.1.18
MyIoCore

192.168.1.19
MyB-Station

192.168.1.25
MyDaliCore

192.168.1.167
MyCueCore

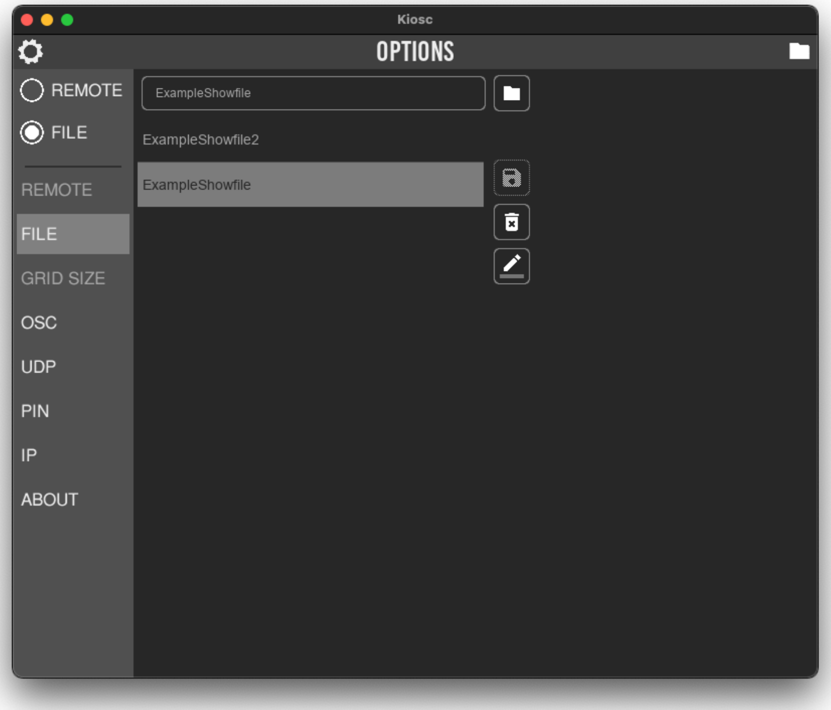
192.168.1.171
MyBStation

192.168.1.213
KioscEditor

Figure 2.4: Remote options

2.3.3 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.



ExampleShowfile

ExampleShowfile2

ExampleShowfile

Figure 2.5: File options

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

2.3.4 Grid Size

In *Remote* mode 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.

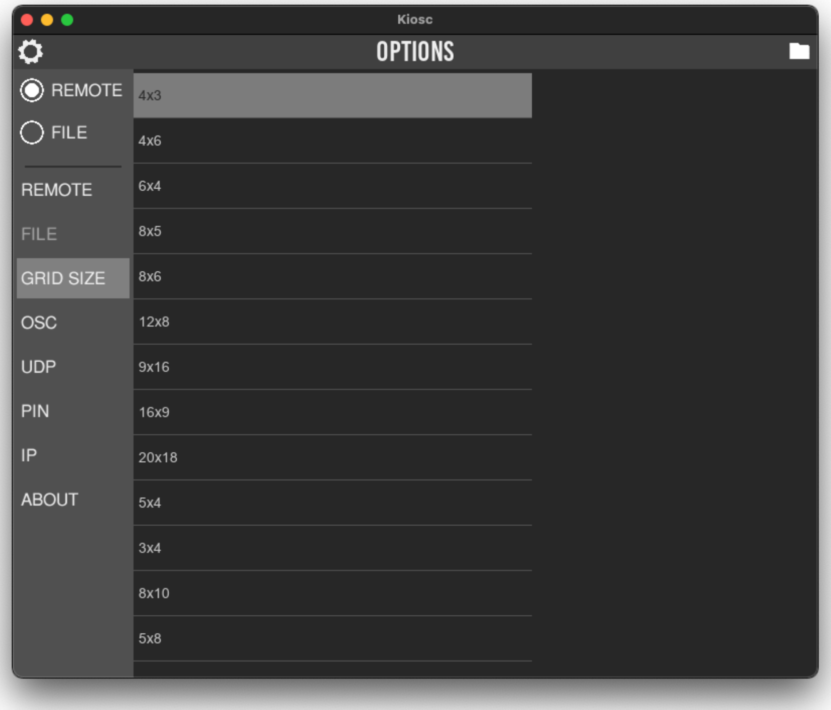
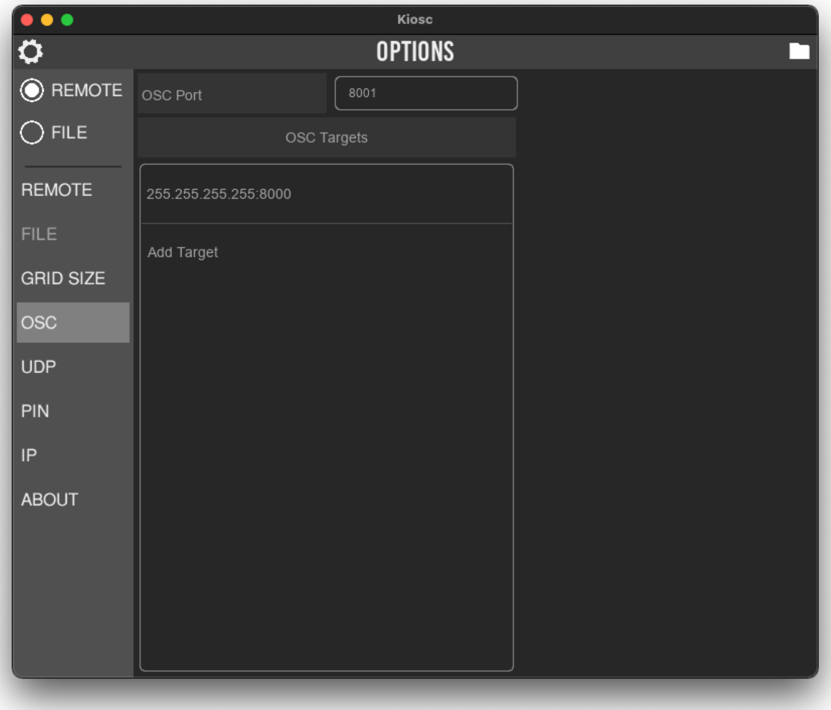


Figure 2.6: Grid Size options

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

2.3.5 OSC

The settings in this screen (see figure 2.7) 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 decimal number.



● REMOTE
○ FILE

OSC Port 8001

OSC Targets

255.255.255.255:8000

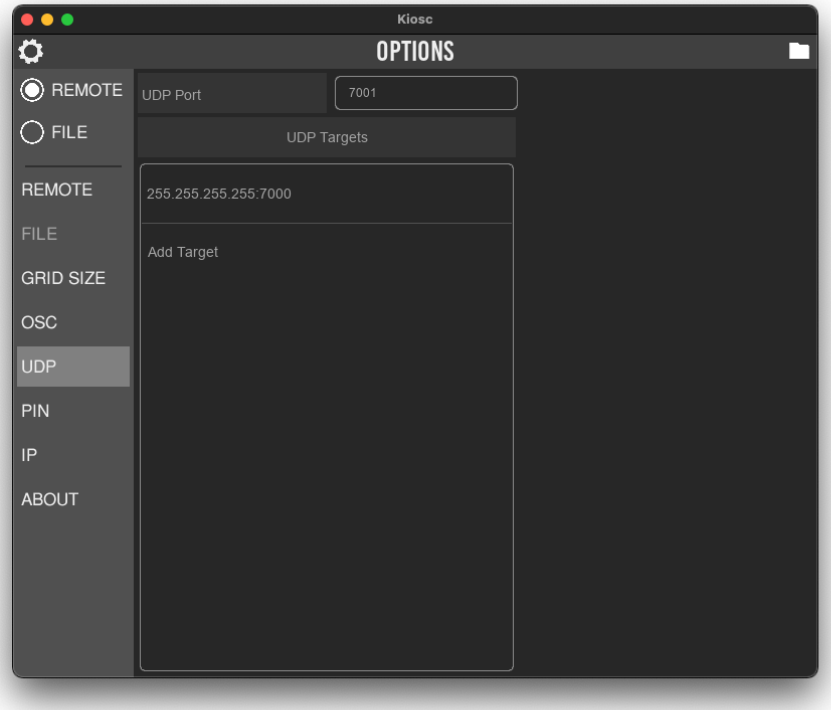
Add Target

Figure 2.7: 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 need to know this incoming port number. By default this port number is set to 8000.

2.3.6 UDP

The settings in this screen (see figure 2.8) 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 decimal number.



UDP Port 7001

UDP Targets

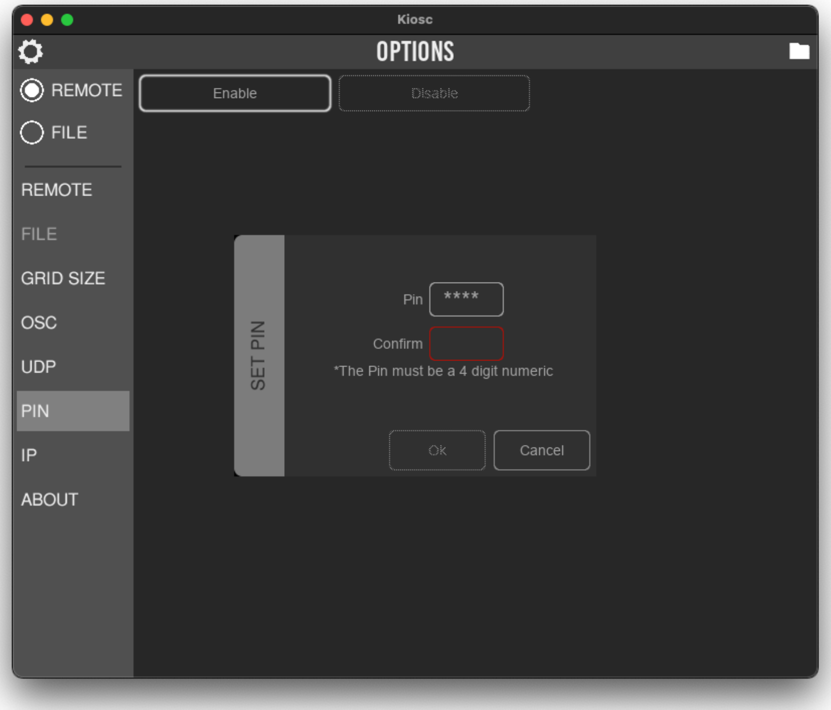
255.255.255.255:7000

Figure 2.8: UDP options

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

2.3.7 PIN

Although the Kiosc app has no editing capabilities (all editing is done in the Kiosc Editor), 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 2.9.



Pin ****

Confirm

*The Pin must be a 4 digit numeric

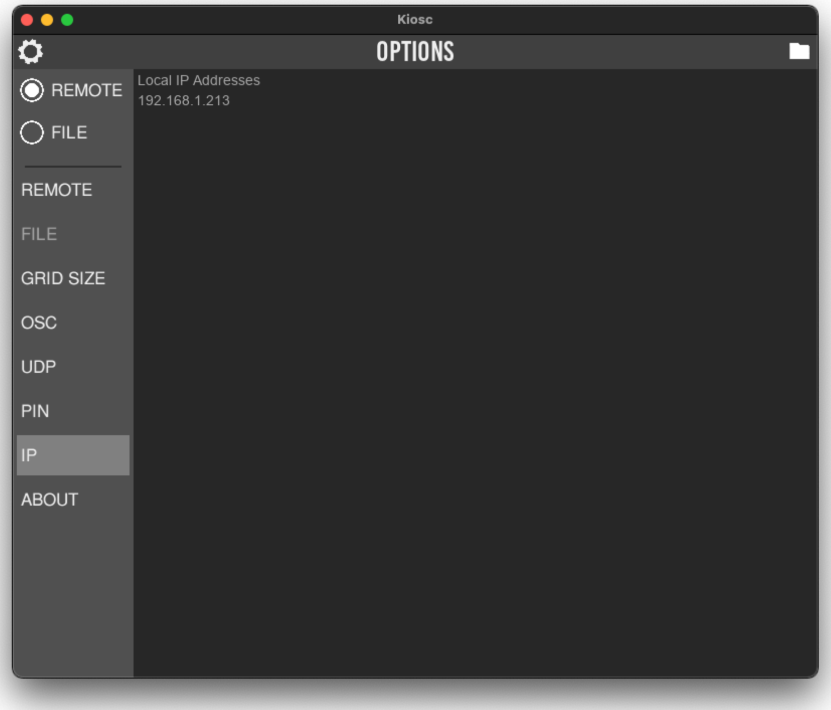
Figure 2.9: Enabling the PIN

2.3.8 IP

The IP page differs between the Kiosc App and the Kiosc Touch, as shown below.

Kiosc App

On the Kiosc App the IP page shows a list of local IP addresses used by the device, as shown in figure 2.10. This information can be useful when trouble shooting network issues. The network settings are handled by the Operating System of the device running the Kiosc App.



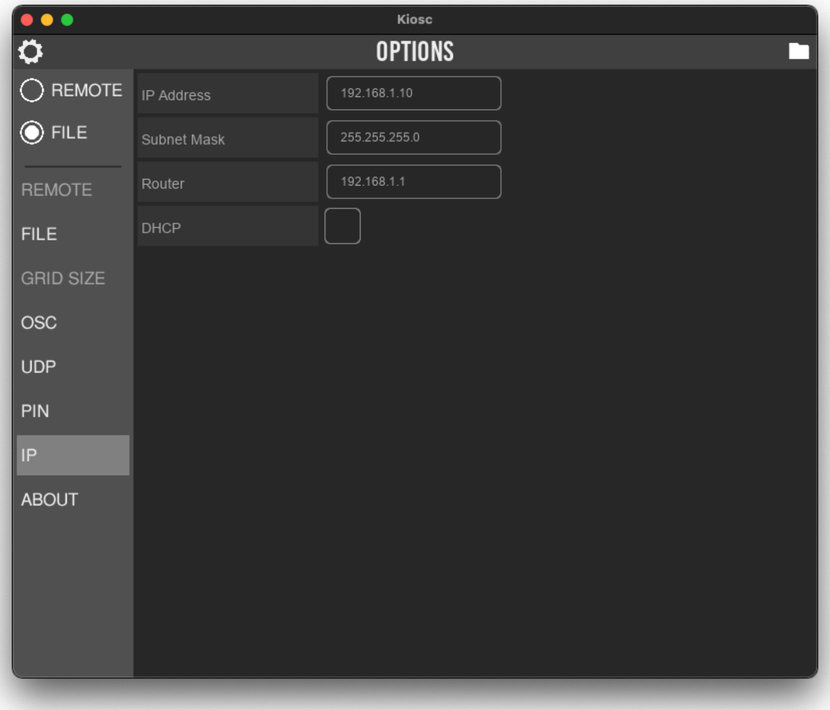
● REMOTE
○ FILE

Local IP Addresses
192.168.1.213

Figure 2.10: IP page on the Kiosc App

Kiosc Touch

The IP page on the Kiosc Touch is used for setting the desired network settings, see figure 2.11. A static IP can be set, or the DHCP option can be checked for automatic configuration.



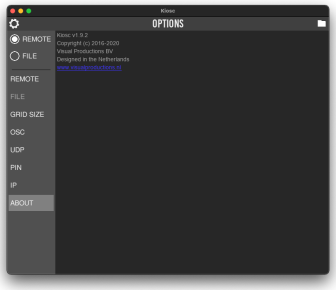
○ REMOTE
● FILE

IP Address	192.168.1.10
Subnet Mask	255.255.255.0
Router	192.168.1.1
DHCP	□

Figure 2.11: IP page on the Kiosc Touch

2.3.9 About

Apart from the ubiquitous copyright notice, the about screen in figure 2.12 indicates the current software version.



Kiosc v1.9.2
Copyright (c) 2016-2020
Visual Productions BV
Designed in the Netherlands
www.visualproductions.nl

Figure 2.12: The About screen

2.4 Installation

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

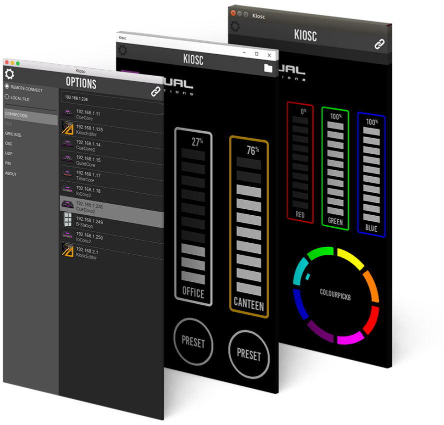


Figure 2.13: Kiosc on macOS, Windows and Ubuntu

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 10 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 Kiosc app from the snap store:
  `https://snapcraft.io/kiosc`

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

  ```
  snap find kiosc
  snap install kiosc
  ```

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

  ```
  snap refresh kiosc
  ```

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

Chapter 3

Kiosc Touch

The Kiosc Touch is a Touch screen solution created by Visual Productions. It is a standalone PoE-powered device running the Kiosc app.

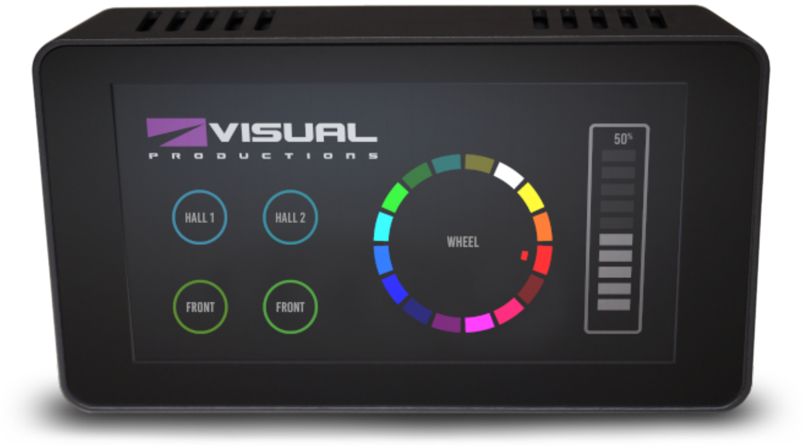


Figure 3.1: Kiosc Touch

3.1 Specifications

The Kiosc Touch has the following specifications.

- PoE (Power-over-Ethernet) Class III.
- 7 Inch full colour display
- Viewable screen size: 155mm x 86mm
- Capacitive multitouch

- Resolution of 800 x 480 pixels
- Wall mountable
- VESA 75
- 460 gram (1 lbs)
- Operating temperature -20°C to +50°C (-4°F to 122°F)
- Compliance EN55103-1 EN55103-2

3.2 Mounting

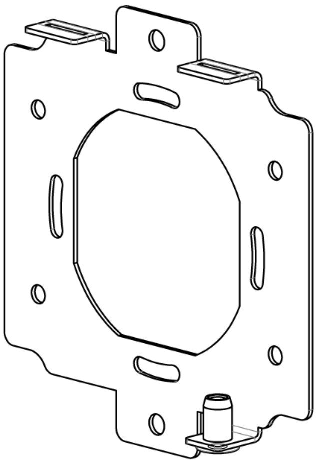


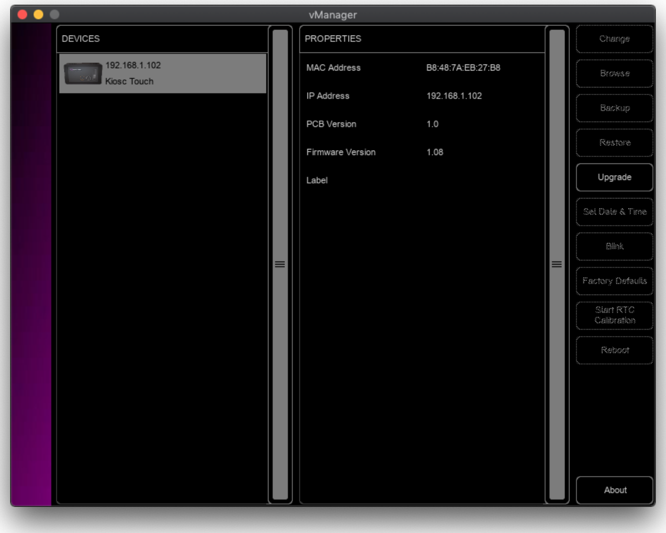
Figure 3.2: Kiosc Touch mounting bracket

The Kiosc Touch is wall-mount, it is compatible with European and American gang boxes. If no gang box is available then the Kiosc Touch can also be surface mounted. The mounting bracket of the Kiosc Touch suggests where you can create a hole for the cable to enter.

3.3 vManager

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

- Discovering any Kiosc on the Network
- Perform firmware upgrades



192.168.1.102
Kiosc Touch

MAC Address	B8:48:7A:EB:27:B8
IP Address	192.168.1.102
PCB Version	1.0
Firmware Version	1.08
Label	

Figure 3.3: vManager

3.4 Network

The Kiosc Touch is a network device. A network connection between between a computer running Kiosc Editor and the unit is required to program the Kiosc Touch. More information on this can be found in chapter 4.

There are multiple arrangements possible for connecting the computer and the Kiosc Touch. They can be connected peer-to-peer, via a network switch or via a Wi-Fi bridge device. Figure 3.4 illustrates these different arrangements.

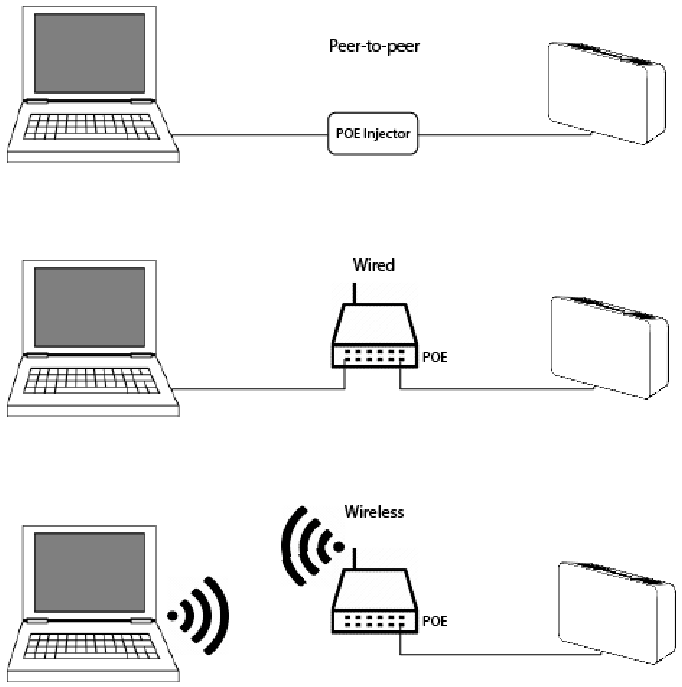


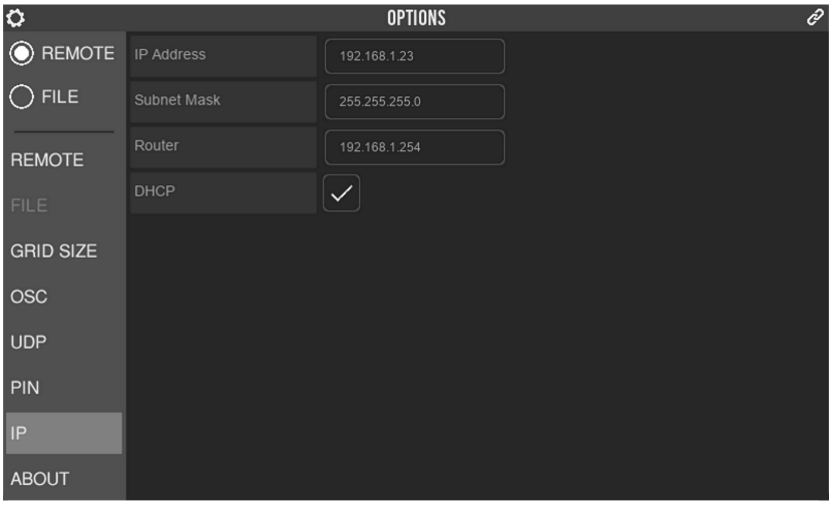
Figure 3.4: Network arrangements

The Ethernet port on the Kiosc Touch is auto-sensing; it does not matter whether a cross or straight network-cable is being used.

3.5 IP Address

The Kiosc Touch supports both static IP addresses and automatic IP addresses. By default, the Kiosc Touch is set to DHCP in which it will automatically be assigned an IP address by the DHCP server in the network. The 'DHCP server' is typically part of the router's functionality.

Static IP addresses are useful when there is no DHCP server in the network, for instance when there is a direct peer-to-peer connection between a Kiosc Touch and a computer. It is also useful in permanent installations where the IP address of the Kiosc Touch is known by other equipment and therefor should not change. When using DHCP there is always the risk of automatically being given a new IP address in the event that the DHCP server is replaced. When using static IP addresses make sure that all equipment on the network have unique IP addresses.



IP Address: 192.168.1.23

Subnet Mask: 255.255.255.0

Router: 192.168.1.254

DHCP: ☑

Figure 3.5: IP Settings

The IP address or DHCP mode of the Kiosc Touch can be set in the 'IP' tab in the settings menu, as shown in figure 3.5. The 'IP Address' field is also where the current IP is shown.
Note: This only applies to the Kiosc Touch, on the Kiosc App the network settings are configured using the operating system of the device running Kiosc.

Chapter 4

Kiosc Editor

The Kiosc editor is a software tool for designing Kiosc layouts.

4.1 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**

  Kiosc Editor can be downloaded from the Microsoft store at:
  `https://www.microsoft.com/store/apps/9n4ph6cs6tht`.

  Windows 10 is required.

- **macOS**

  Visit the Apple macOS app store at:
  `https://apps.apple.com/us/app/kiosc-editor/id1257068566`

  macOS 11+ is recommended.

- **Ubuntu**

  You can acquire Kiosc Editor from the Snapcraft store:
  `https://snapcraft.io/kiosceditor`

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

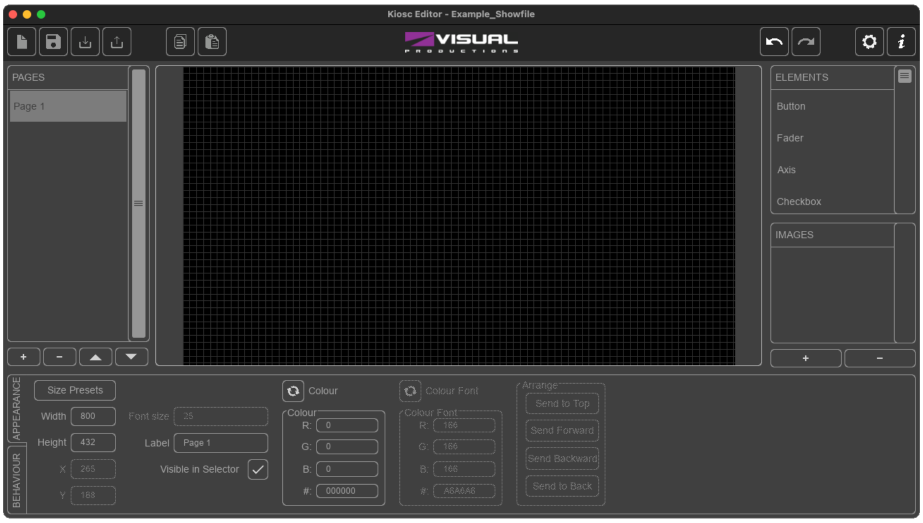
  ```
  snap find kiosceditor
  snap install kiosceditor
  ```

  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.

4.2 The Editor



Page 1

Width 800
Height 432
X 265
Y 188
Font size 25
Label Page 1
Visible in Selector ☑

Colour
R: 0
G: 0
B: 0
#: 000000

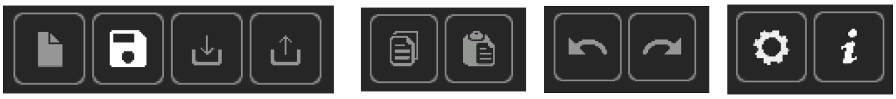
Colour Font
R: 166
G: 166
B: 166
#: A6A6A6

Figure 4.1: Kiosc Editor user-interface

Figure 4.1 shows the Editor's user-interface. Each section is discussed in detail below.

4.2.1 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.



(a) File (b) Clipboard (c) Undo (d) Options

Figure 4.2: Menu bar

4.2.2 Pages

Your design needs at least one page. Add more pages by pressing the + button in figure 4.3. The *Up* and *Down* buttons allow you to change the order of the pages.

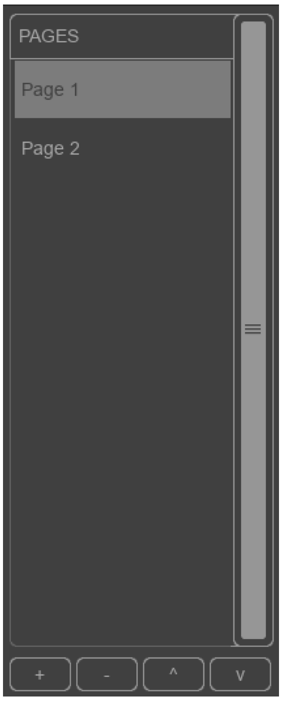


Figure 4.3: Pages

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

4.2.3 Elements

Controls like buttons, faders, encoders, etc. are called *Elements* (see figure 4.4). A layout is basically created by dragging several *Elements* into the *Pages*.

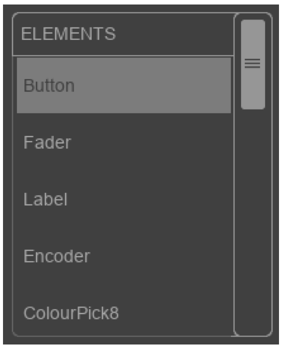


Figure 4.4: Elements

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

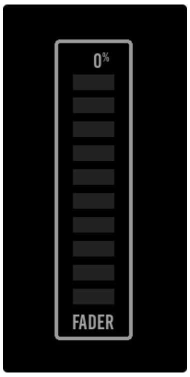
Button

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

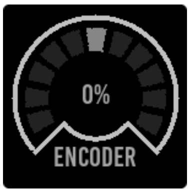


The fader can send a range of values. The range can be set using the Min and Max values. The type of value can be set to a float or an integer. A fader can simply be resized to get a horizontal fader.

Label

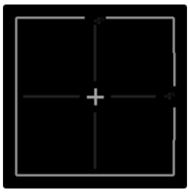
The label element is for creating a clear, easy to use layout. It has no function when pressed.

Encoder



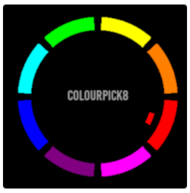
The Encoder sends a float value of -1.0 to 1.0.

Axis



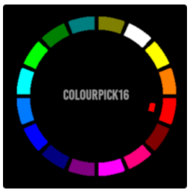
The Axis element is for controlling XY values, like a moving head's pan and tilt.

Colourpicker 8



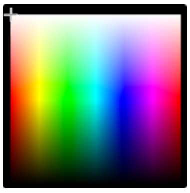
The Colourpicker 8 has eight colours to choose from. These are send as OSC colour value, or via UDP using Hex #RRGGBB.

Colourpicker 16



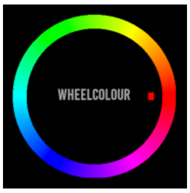
The Colourpicker 16 has sixteen colours to choose from. These are send as OSC colour value, or via UDP using Hex #RRGGBB.

ColourSpectrum



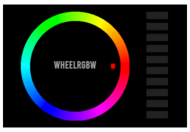
The Colourspectrum has many colours to choose from. These are send as OSC colour value, or via UDP using Hex #RRGGBB.

Wheelcolour



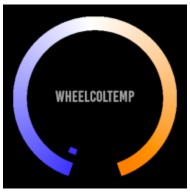
The Wheelcolour has many colours to choose from. These are send as OSC colour value, or via UDP using Hex #RRGGBB.

WheelRGBW



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

WheelColTemp

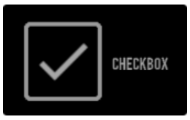


The WheelColTemp is ideal to mix between warm and cold white. It sends a float value between 0.00 and 1.00

Text

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

Checkbox

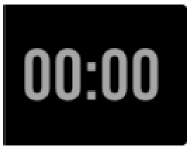


A checkbox element can be used as a button. It will send ON or 1 when checked, and OFF or 0 when unchecked.

Image

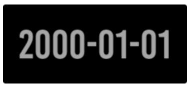
The image element is perfect for adding your own logo to your Kiosc layout. It behaves similar to the button element, so it can be used to create custom buttons.

Clock



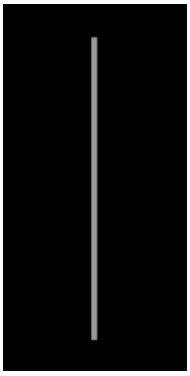
The clock element displays the current time of the system. It has no other functions when pressed.

Date



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

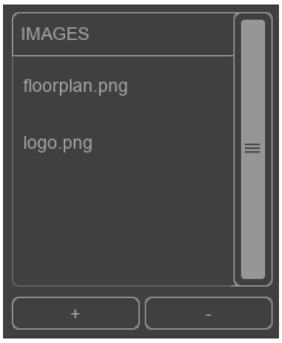
Line



The line element is used purely for design, it does not send any commands but is ideal as a separator between controls. It can change colour using feedback.

4.2.4 Images

A *Page* can have a background image. Also, there is an *Image Element* that can display an image. To use images in your design, you first have to import them into the images list (see figure 4.5). 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.



IMAGES

floorplan.png

logo.png

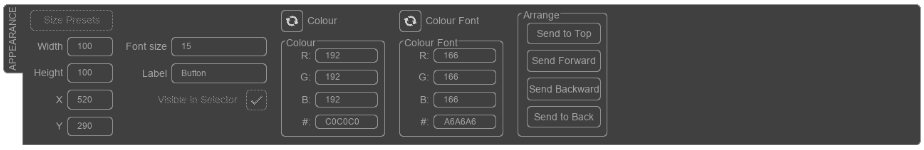
Figure 4.5: Images

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.

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.

4.2.5 Appearance Properties

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



Width 100
Height 100
X 520
Y 290
Font size 15
Label Button
Visible In Selector ☑

Colour
R: 192
G: 192
B: 192
#: C0C0C0

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

Figure 4.6: Appearance properties

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. 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

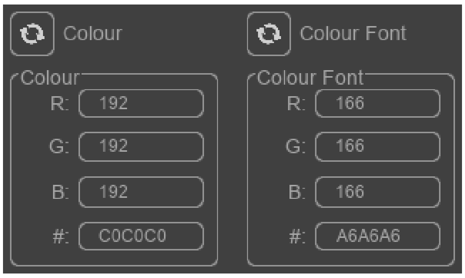
This decides the label and the font size for the selected element. When a page is selected, the label of the page can be set here.

Visible in selector

Currently only applies to pages. When selected, the page will not be shown on the device running Kiosc. Useful for hiding work-in-progress pages.

Colours

The colour of the *Element* can be changed by setting the RGB fields in figure 4.7. 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
R: 192
G: 192
B: 192
#: C0C0C0

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

Figure 4.7: Colour properties

Arrange

When *Elements* overlap, you can use the *Arrange* buttons to control which *Element* lies at the top (see figure 4.8).

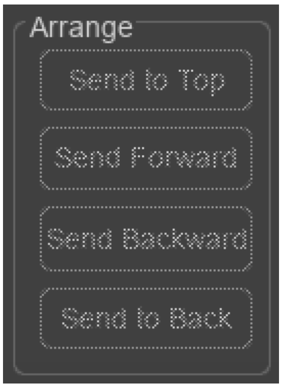
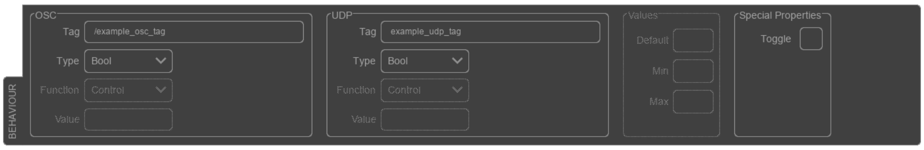


Figure 4.8: Arrange buttons

4.2.6 Behaviour Properties



OSC
Tag /example_osc_tag
Type Bool
Function Control
Value

UDP
Tag example_udp_tag
Type Bool
Function Control
Value

Values
Default
Min
Max

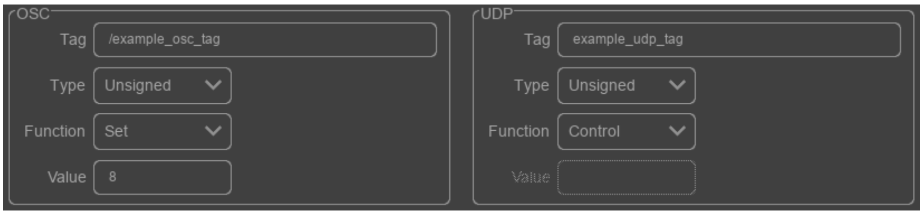
Special Properties
Toggle □

Figure 4.9: Behaviour properties

OSC/UDP

When the user clicks on a button in Kiosc, or controls any other element, it typically sends an OSC or UDP message to an external device. Which message is send out is determined by the *OSC* or *UDP* settings (see figure 4.10).

An OSC tag should always start with a '/' prefix. For some *Elements* it is possible the specify the data type that will be send in the OSC or UDP 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 send. More information on each specific element can be found in appendix A.



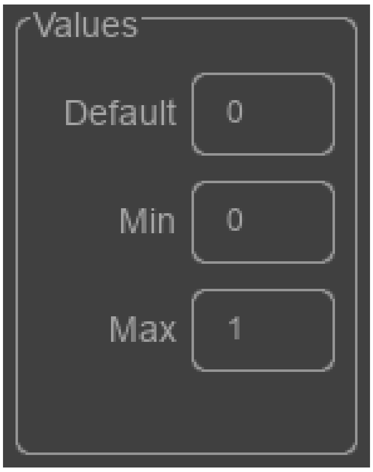
OSC
Tag /example_osc_tag
Type Unsigned
Function Set
Value 8

UDP
Tag example_udp_tag
Type Unsigned
Function Control
Value

Figure 4.10: Tags

Values

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



Values

Default 0

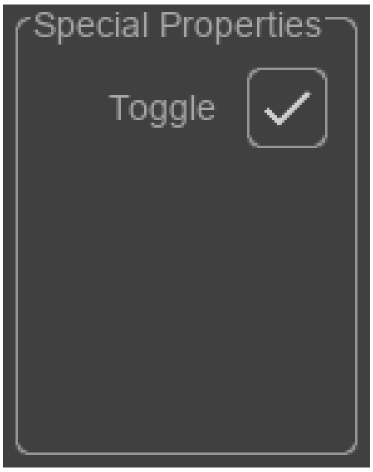
Min 0

Max 1

Figure 4.11: Values

Special Properties

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.



Special Properties

Toggle ☑

Figure 4.12: Special Properties

4.2.7 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.

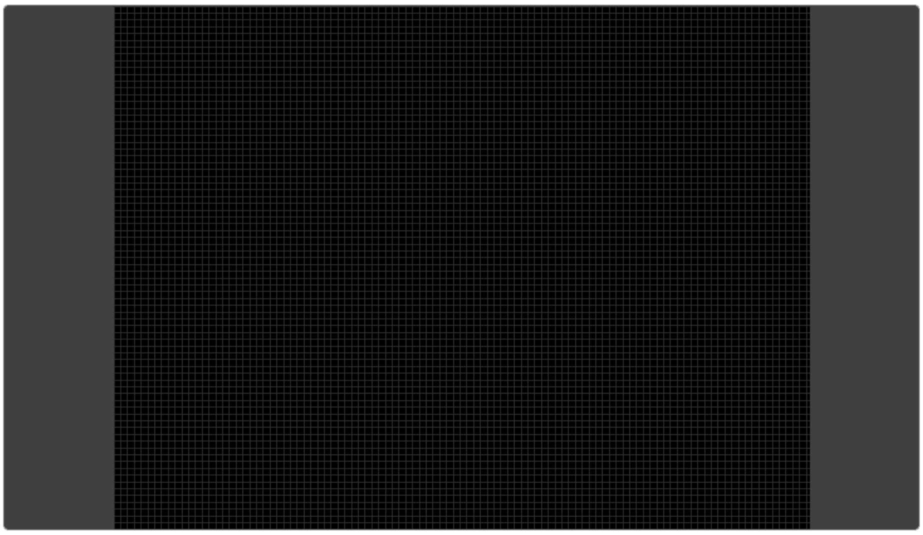
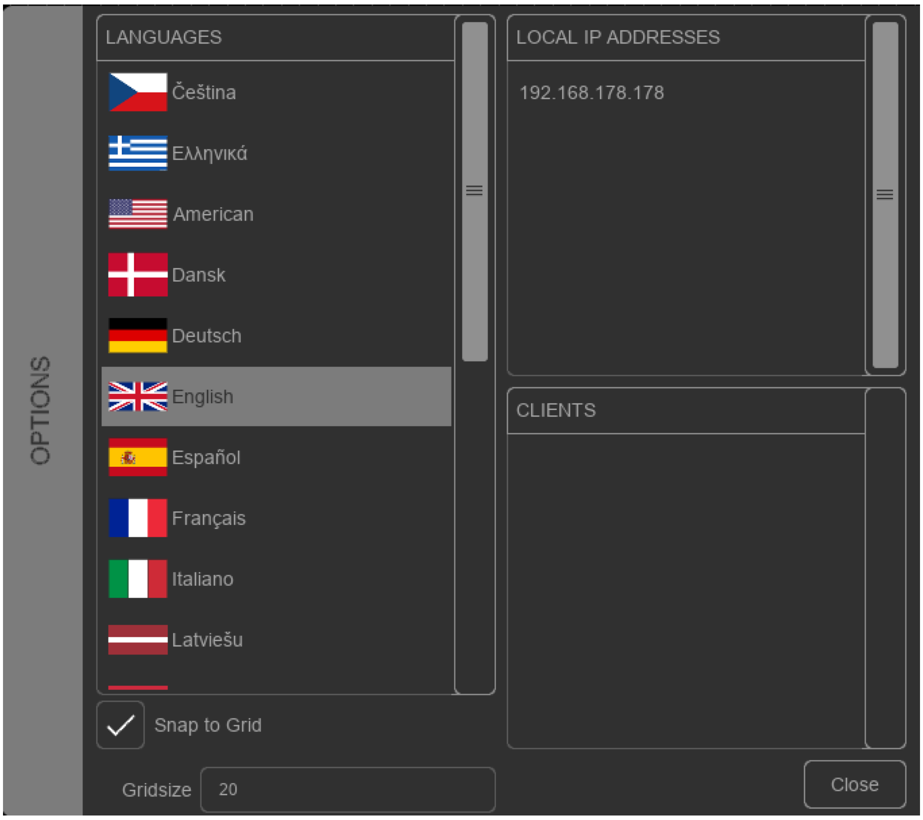


Figure 4.13: Layout

4.2.8 Options

The Editor only has a few options, see figure 4.14.



192.168.178.178

☑ Snap to Grid

Gridsize 20

Figure 4.14: Options

Language

There are various languages available.

Snap to Grid

Enable the *Snap to Grid* to have an aid in aligning Elements. The size of the grid can be specified in the *Gridsize* field.

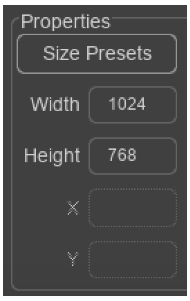
Local IP Addresses

A list with local IP addresses used by the computer running the Kiosc Editor. This information can be useful when trouble shooting network issues.

Clients

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

4.3 Screen Size



Properties

Width 1024

Height 768

X

Y

Figure 4.15: Setting the 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 4.15.

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

4.4 Feedback

Feedback in the Kiosc app is created by having the external equipment sending messages back to Kiosc. 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 5

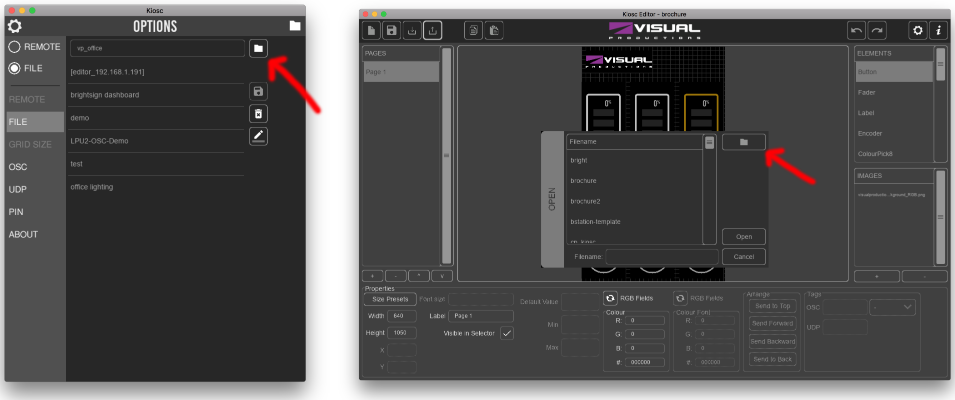
File Locations

When Kiosc is running in *Remote* mode then there are no files involved, however, when running in *File* mode then Kiosc 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 Kiosc and Kiosc Editor are not allowed to reach each other files.

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

The designated file location differs per operating system and is likely to be a long and obscure path. For this reason, Kiosc and the Editor 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) Kiosc

(b) Kiosc Editor

Figure 5.1: Folder shortcut button

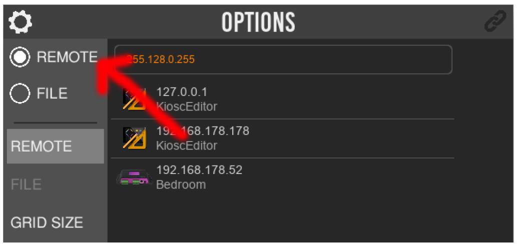
The layout file created by the Editor will have the `.kiosc` file extension.

Transferring files from the Kiosc Editor running on a desktop OS to the Kiosc 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.

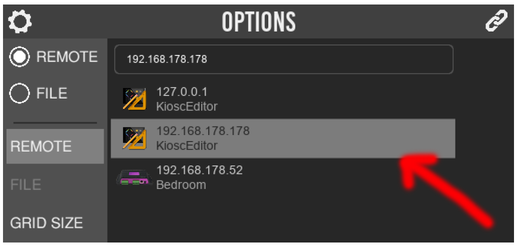
To make copying files to Kiosc on a mobile device more convenient, Kiosc is capable to connect to a Kiosc Editor under *Remote* mode and subsequently save it as a local file in *File* mode. Please note that due to Microsoft Store security measurements, it is not possible to to make a remote connection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

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

1. Activate *Remote* mode.

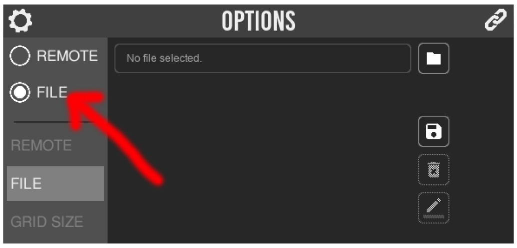


2. Connect to the Editor.

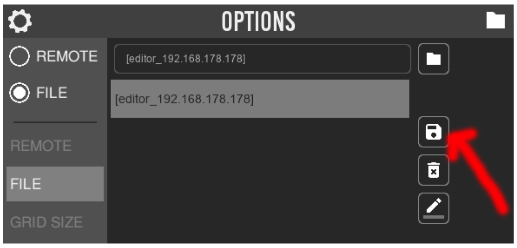


The layout is now loaded into Kiosc's memory.

3. Activate *File* mode.



4. Save the file by pressing the *Save* button.



[editor_192.168.178.178]

[editor_192.168.178.178]

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 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.1.1 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

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.1.2 Feedback

Feedback can be send 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:

- **State**
  The state of a button 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 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

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.2 Fader

The fader element can be used to send 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.2.1 Send

OSC

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

UDP

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.2.2 Feedback

Feedback can be send to a fader using its tag. The type of message determines which attribute of the fader it changes. The following attributes of a button 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

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

A.3 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.3.1 Feedback

Feedback can be send 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

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.4 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.4.1 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

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.4.2 Feedback

Feedback can be send 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

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

A.5 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.5.1 Feedback

Feedback can be send 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

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

Index