DALICORE
MANUAL

Revision History

Revision	Date	Author(s)	Description
1	01.05.2021	ME	Initial version
2	28.05.2021	ME	Updated screen shots
3	21.06.2021	ME	Updated screen shots

Contents

Declaration of Conformity

$C\epsilon$

We, manufacturer Visual Productions BV, herby declare under sole responsibility, that the following device:

DaliCore

Conforms to the following EC Directives, including all amendments:
EMC Directive 2004/108/EG

And the following harmonized standards have been applied:
NEN-EN-IEC 61000-6-1:2007
NEN-EN-IEC 61000-6-3:2007

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

Date:
March 5th, 2020

Place:
Haarlem, The Netherlands

ing. Maarten Engels
Managing Director
Visual Productions BV

Chapter 1

Introduction

Thank you for choosing the DaliCore. The DaliCore is a solid state controller designed for controlling DALI and DMX lighting systems.

The DaliCore is an application lighting controller that supports DALI and - pending a software upgrade - also supports DALI-2. An ideal solution for a stand-alone DALI installation as the DaliCore takes care of the commissioning and daily control of the DALI fixtures.

An internal web-server provides the web-interface through which you can program the DaliCore. A modern browser is required to access this web-interface during set-up. A browser or computer is not required for stand-alone use after the initial set-up.

At the time of writing this manual the DaliCore firmware was at version 1.00.

Figure 1.1: DaliCore

This manual discusses installing and programming the unit. Chapter 2 provides background information on the communication protocols used in the DaliCore. Chapters 3 and 4 cover how to set up the unit and configure the network connection.

The DaliCore needs to know which fixtures are connected to both the DALI and DMX port. Entering this data is called patching and this is discussed in chapters 5 and 6.

Chapter 7 discussed directly controlling fixtures and programming lighting scenes. Responding to (external-)triggers, automation and protocol conversion is covered in chapters 8 and 9.

Incoming and outgoing data can be monitored, as covered in chapter 10. The DaliCore's device settings are listed in chapter 11.

Finally, the DaliCore is compatible with complimentary software tools vManager and Kiosc. These tools are discussed in chapters 12 and 13 respectively.

1.1   Features

The feature set of the DaliCore includes:

- DALI
- DT-6 and DT-8
- DMX-512
- RDM
- Ethernet based, supporting protocols like UDP, OSC, Art-Net and sACN
- Web-based user-interface for programming
- 4 GPI ports
- PoE (Power Over Ethernet) Class I
- Compatible with DaliCore, CueCore2, QuadCore and IoCore2
- Bundled with vManager software
- Compatible with Kiosc software

1.2   Limitations

The DaliCore is a powerful device with many possibilities, however, there are some limitations as shown in the following table.

DALI Fixtures	64
DMX Fixtures	32
Actionlists	8
Actions per list	48
Actions system-wide	64
Tasks per Action	8
Tasks system-wide	128
Variables	10
Timers	4

The DMX playback implemented in the DaliCore supports 16 static scenes; one can be active at a time.

1.3 What's in the box?

The DaliCore packaging contains the following items (see figure 1.2):

- DaliCore
- Info card
- UTP network cable

Figure 1.2: DaliCore box contents

1.4 Saving data to memory

This manual will describe how to configure the DaliCore and program scenes, action, etc. The unit's web-interface is used for editing these kinds of elements. When changes are made, these changes are directly stored in the RAM memory of the DaliCore and the programming will directly influence the behaviour of the unit. RAM memory is, however, volatile and its content will be lost through

a power cycle. For this reason the DaliCore will copy any changes in the RAM memory to its onboard flash memory. Flash memory retains its data even when not powered. The DaliCore will load all its data back from the flash memory upon startup.

This memory copy process is conducted automatically by the DaliCore and should not be of any concern of the user. One point of consideration is, however, that after making a change the unit should be given time to perform the copy to flash. As a rule of thumb, do not disconnect the power from the device within 30 seconds from making a programming change.

1.5   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

Protocols

The DaliCore supports various protocols. This chapter describes these protocols and to which extent they are implemented in the DaliCore.

2.1 DALI

Digital Addressable Lighting Interface (DALI) is a protocol for controlling lighting systems. DALI has been developed to replace 0-10V controlled lighting. The protocol is owned by DiiA (Digital Illumination Interface Alliance) and defined in the IEC 62386 standard.

A DALI network consists of at least one controller and a bus power supply. Between one and maximum 64 lighting fixtures can be connected to the network. The communication is bi-directional via an asynchronous, half-duplex, serial protocol over a two-wire bus at a transfer rate of 1200 bit/s. The wiring of the network can be made in a bus or star topology, or a combination of the two.

The controller can address fixtures individually, per group or all together (broadcast). Each fixture has an address between 0 and 63. These addresses are assigned during the commissioning of the system. The DaliCore can perform these commissioning steps.

Figure 2.1: Mean Well DLP-04R

The bus power supply (e.g. figure 2.1) puts 16V DC on the bus wires.

2.2 DMX-512

DMX-512 is the standard communication protocol for stage lighting. Its official name is E1.11-2008 USITT DMX512-A. Nowadays the reach of the DMX protocol has extended beyond entertainment lighting and is also used for architectural lighting.

Originally one DMX network contained 512 channels which is called a 'universe'. With the growing size and complexity of lighting systems it is now very common for a system to compose of multiple universes, each conveying 512 channels.

It is advised to use a shielded twisted pair cable for DMX cabling. The cable should be terminated with an 120 Ohm resistor.

DMX-512 is a very successful protocol with, however, a few limitations. The maximum number of attached devices is limited to 32 and they all have to be connected in bus-topology having one cable running via each device. Furthermore, a DMX-512 cable should not be longer than 300 meters.

Figure 2.2: Visual Productions' RdmSplitter

The DIN Rail RdmSplitter from Visual Productions (See figure 2.2) helps tackle

those inconvenient limitations. The Splitter takes a DMX signal and sends it out again on its 6 DMX output ports for scaling group topology. Each output port is capable of driving 32 more devices. The Splitter can also function as a signal booster as each port supports another 300 meter long connection.

2.3 RDM

The Remote Device Management (RDM) protocol - officially called ANSI E1.20 - is build on top of the DMX-512 protocol. RDM enhances DMX with bi-directional communication allowing the fixtures to be detected, addressed and polled for status information.

Although standard DMX cables can be used, the equipment requires specific electronic considerations for being RDM capable.

The DaliCore can discover RDM fixtures, set starting addresses and choose modes.

2.4 GPI

The DaliCore features four General Purpose Input (GPI) ports that can be connected to external equipment, switches and sensors. State changes on these GPI ports can be used to trigger programmed events inside the DaliCore.

Each GPI port can be switched between *Digital* and *Analog*. Set to *Digital* the port works as a contact-closure. In *Analog* mode the port is a 0-10V level input.

2.5 Art-Net

The Art-Net protocol primarily transfers DMX-512 data over Ethernet. The high bandwidth of an Ethernet connection allows Art-Net to transfer up to 256 universes.

The data sent out for Art-Net does put a certain load on the network, therefore it is recommended to disable Art-Net when not in use.

The DaliCore supports sending and receiving of one universe of Art-Net data.

2.6 sACN

The streaming Architecture of Control Networks (sACN) protocol uses a method of transporting DMX-512 information over TCP/IP networks. The protocol is specified in the ANSI E1.31-2009 standard.

The sACN protocol supports multi-cast in order to take efficient use of the network's bandwidth.

The DaliCore supports sending and receiving of one sACN universe.

2.7 TCP

The Transmission Control Protocol (TCP) is a core protocol of the Internet Protocol Suite. It is used for its reliable, ordered and error checked delivery of a stream of bytes between applications and hosts over IP networks. It is considered 'reliable' because the protocol itself checks to see if everything that was transmitted was delivered at the receiving end. TCP allows for the re-transmission of lost packets, thereby making sure that all data transmitted is received.

The DaliCore supports receiving TCP messages.

2.8 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, therefor it is faster than TCP but less reliable.

There are two ways how to have the DaliCore respond to incoming UDP messages. The API (see page 84) makes typical DaliCore functions available through UDP. Furthermore, custom messages can be programmed in the Show Control page (see page 33). This is also the place where to program outgoing UDP messages.

2.9 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, it can contain various information.

There are apps available for creating custom-made user interfaces on iOS (iPod, iPhone, iPad) and Android. These tools allow to program fool-proof user-interfaces for controlling the device. E.g. Kiosc from Visual Productions.

There are two ways how to have the DaliCore respond to incoming OSC messages. Firstly, the API (see page 82) makes typical DaliCore functions available through OSC. Secondly, custom messages can be programmed in the Show Control page (see page 33).

2.10 DHCP

The Dynamic Host Configuration Protocol (DHCP) is a standardised network protocol used on Internet Protocol (IP) networks for dynamically distributing network configuration parameters, such as IP addresses.

The DaliCore is a DHCP client.

Chapter 3

Installation

This chapter discusses how to set up the DaliCore.

3.1   Mounting

The DaliCore is DIN rail mounted. The unit takes up 6 DIN modules.

3.2   Power

The DaliCore requires a DC power supply between 9 and 24 Volt with a minimum of 500mA. The DC connector is two-pin screw terminal.

The DaliCore is also Power-over-Ethernet (PoE) enabled. It requires PoE Class I.

Please note that the DaliCore is shipped without a DC power-supply.

3.3   DALI

A DALI system requires a bus power-supply on the DALI bus. Figure 3.1 illustrates a bus power-supply connected to the DALI network. Please note a separate power-supply is needed to power the DaliCore.

The DaliCore can detect if a bus power-supply is missing. In that case the message as seen in figure 3.2 is shown on the *DALI Patch* and *Playback* page.
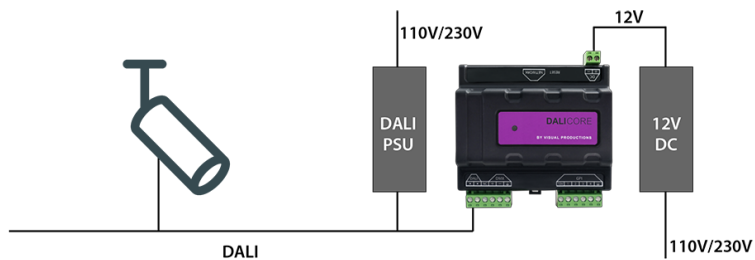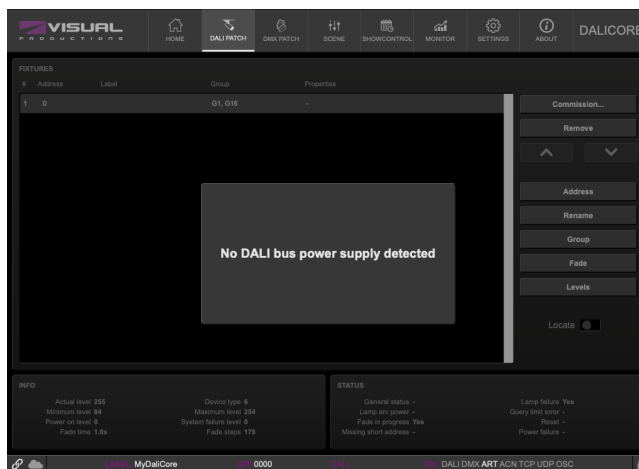
15

Figure 3.1: DALI connections

Figure 3.2: Bus power-supply missing

3.4 DMX

The DMX cable is connected to the DaliCore via a three-pin screw terminal (see figure 3.3).

The port can be configured either as DMX input or DMX output. This setting is discussed on page 54.

3.5 GPI

The DaliCore features four General Purpose Inputs (GPI) ports that can be connected to external equipment, switches and sensors. State changes on these GPI ports can be used to trigger programmed events inside the DaliCore.

Each GPI port can be switched between 'digital' and 'analog'. In the digital mode the signal is held up by an internal pull-up resistor and results in a logic '0'. The external equipment is intended to short the port's pin to the provided ground pin. This short will create a logic '1'. All four ports share one common ground pin.
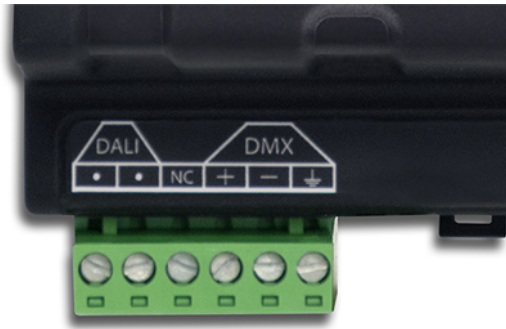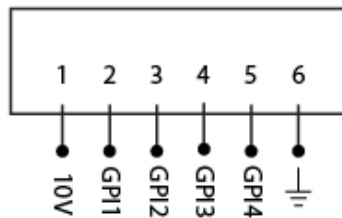
Figure 3.3: DMX connection

Figure 3.4: GPI Pinout

When set to analog the external equipment is supposed to supply a voltage between 0V and 10V to the port's pin. For convenience, a 10V supply is available on one of the pins of the GPI connector. Please refer to figure 3.4 for the pinout of the GPI connector. Be careful not to supply more than 10V to the GPI port as that might cause permanent damage.

Figure 3.5 shows examples of how to wire a contact-closure to a GPI port set to digital. And it shows an example of potentiometer connected to a GPI port set to analog.

Please refer to page 57 for more information on configuring the GPI ports. Programming events based on GPI activity is done in the Show Control page, which is discussed on page 33.

The green connector used for the GPI port has part number CTB9200/6A and replacements can be ordered from Farnell/Newark (order code 3881880).
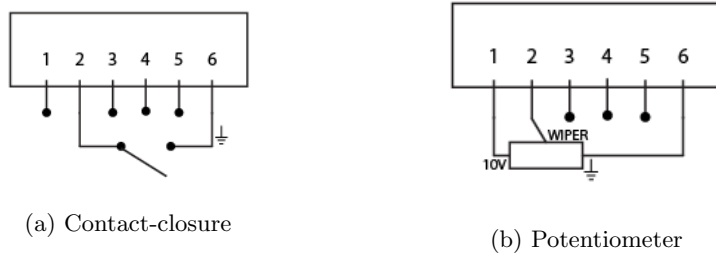
(a) Contact-closure

(b) Potentiometer

Figure 3.5: GPI Wiring examples

Figure 3.6: GPI Connector

Chapter 4

Network

The DaliCore is a network capable device. A network connection between a computer and the unit is required to configure and program the DaliCore, however, once the device is programmed then it is not necessary anymore for the DaliCore to be connected to an Ethernet network.

There are multiple arrangements possible for connecting the computer and the DaliCore. They can be connected peer-to-peer, via a network switch or via Wi-Fi. Figure 4.1 illustrates these different arrangements.
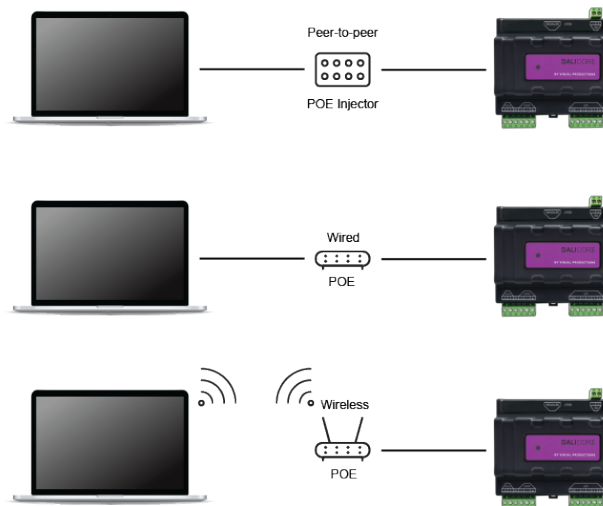

Figure 4.1: Network arrangements

The Ethernet port on the DaliCore is auto-sensing; it does not matter whether a cross or straight network-cable is being used. Although the Ethernet port is classified as 100 Mbps, buffer limits may apply for specific tasks as API messages.

4.1 IP Address

The DaliCore supports both static IP addresses and automatic IP addresses. By default, the DaliCore is set DHCP in which it will be automatically 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 DaliCore and a computer. It is also useful in permanent installations where the IP address of the DaliCore 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.

The DaliCore'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 DaliCore.

Figure 4.2: Reset button

- **vManager** can be used to detect a DaliCore on the network. Once found, the vManager software (figure chapter 12) 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 DaliCore's **web-interface**. The Settings page on this web-interface enables changing the IP address, subnet mask and DHCP settings.

- By briefly pressing the **reset button** on the device it toggles between static and automatic IP addresses. By pressing and holding the reset button (see figure 4.2) on the device for 3 seconds, it will reconfigure the unit to the factory default IP address and subnet mask. No other settings will be changed. The default IP address is 192.168.1.10 with the subnet mask set to 255.255.255.0.

4.2   Web-interface

The DaliCore features an inbuilt web-server. This web-interface can be accessed via a standard browser. It is recommended to use any of the following browsers:

- Microsoft Edge
- Google Chrome (v83 or higher)
- Apple Safari (v14 or higher)
- Mozilla Firefox (v54 or higher)

The web-interface enables you to configure and program the DaliCore. When browsing to the unit the home page (figure 4.3) will appear first. The home page is read-only; it provides information but does not allow for changing any setting. The other pages present many settings that can be edited. These pages will be discussed in the subsequent chapters.
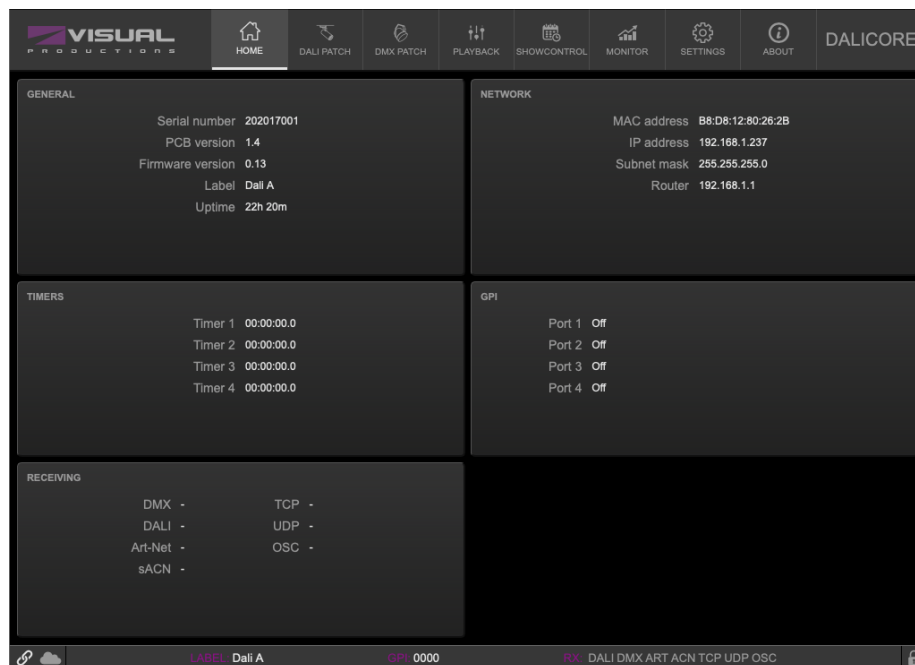

Figure 4.3: Home page

4.2.1   Uptime

This field indicates how long the unit has been alive since its last reboot.

4.2.2   Last Server Poll

Indicates the last time the time & date was fetched from a NTP time server.

4.2.3   Master IP

When the unit is not in Stand Alone mode, then this field displays the IP number of system that is mastering the DaliCore. Refer to chapter **??** for more information on operating modes.

4.3   Access via Internet

The DaliCore can be accessed through the Internet. There are two ways to achieve this: Port Forwarding and VPN.

- **Port Forwarding** Is relatively easy to setup in the router. Each router is different so it is advised to consult the router's documentation (sometimes it is revered to as NAT or Port-Redirecting). Please note that port forwarding is not secure, since anybody could access the DaliCore this way.

- Accessing via a **Virtual Private Network** (VPN) tunnel requires more setup efforts, also the router needs to support the VPN feature. Once set up, this is a very secure way to communicate with the DaliCore. A VPN is a network technology that creates a secure network connection over a public network such as the Internet or a private network owned by a service provider. Large corporations, educational institutions, and government agencies use VPN technology to enable remote users to securely connect to a private network. For further information about VPN please refer to `http://whatismyipaddress.com/vpn`.

Chapter 5

DALI Patch

The *DALI Patch* page is the starting point when using the DaliCore to control DALI fixtures. This page is used for the commissioning of the DALI fixtures.
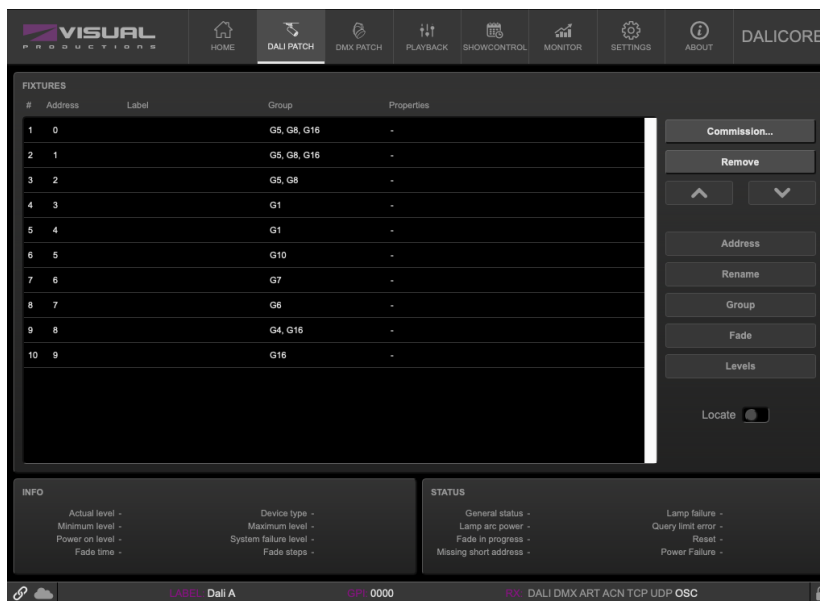

Figure 5.1: DALI Patch Page

5.1 Commission...

DALI fixtures that still have their factory default settings do not have a *short address* assigned to them. A *short address* is a value between 0 and 63. Each fixture should have a unique short address. The process of giving these fixtures an address is called commissioning.

Use the *Commissioning...* button to bring up the commissioning dialog; it offers the following choices:

- **Find** fixtures that have been commissioned before. This function only *reads* the DALI network, it does not alter any settings in fixtures.

- Only **New** fixtures that do not have a short address assigned yet will be found. Each will be given a free address.

- First, the current patch will be cleared. Then **All** connected fixtures will be given a new address regardless whether they already had an address or not.

- **Re-commission selected** fixtures only. These fixtures will be given new addresses. If multiple fixtures share the same *short address* then the communication will be corrupted. This option helps you restore this situation.

5.2   Address

DALI fixtures require a *short address* in the range of [0,63]. Use the *Address dialog* to give each fixture a unique address.
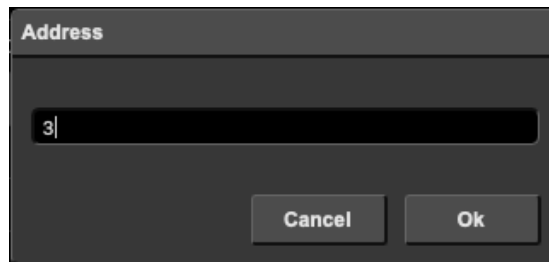

Figure 5.2: Address dialog

5.3   Rename

The *Rename dialog* allows you to assign a label to the fixture. This label is stored in the DaliCore.

5.4   Group

The *Group dialog* (see figure 5.4) enables you to specify in which groups a particular fixture is present. Once present in a group, this fixture will listen to all commands send to this group ID.

A fixture can be represented in multiple groups. The group information is stored inside the DALI luminaire.
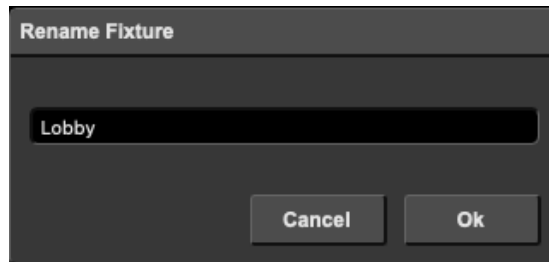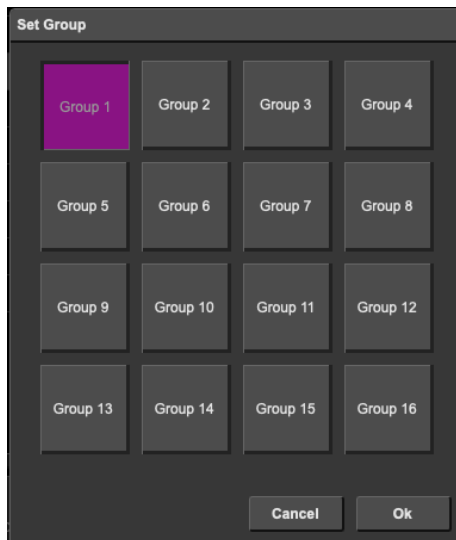
Figure 5.3: Rename dialog

Figure 5.4: Group dialog

5.5 Fade

The *Fade Time* is the time in seconds it will take the fixture to reach a newly instructed lighting level.

The *Fade Rate* defines the size of the *delta* when using *up* and *down* commands.

5.6 Levels

The *Actual Level* is the light level the fixture is currently emitting.

The *Minimum Level* is the lowest brightness the fixture will go to; commands to set the fixture at a lower level than the minimum are ignored.

The *Maximum Level* is the highest lighting level the fixture will go to; the fixture will also ignore commands trying to set the lighting level higher than this maximum setting.
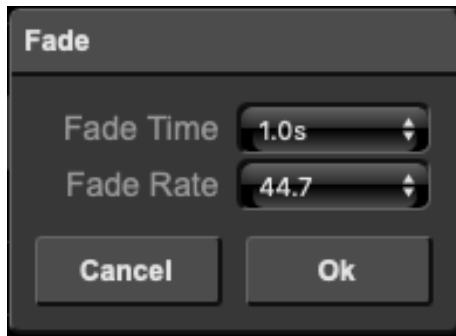
Figure 5.5: Fade dialog

At power-on the fixture will automatically set the intensity at the *Power on level*.

The *Failure Level* is the lighting level the fixture will go to when no DALI communication is detected.
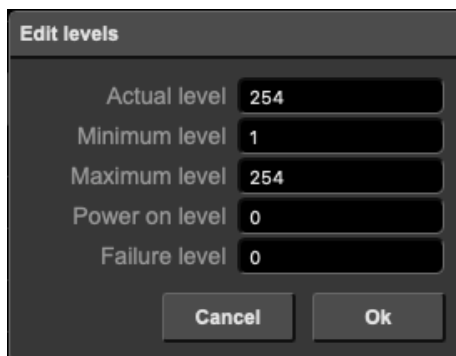
Figure 5.6: Level dialog

5.7   Locate

When the *Locate* checkbox is enabled, then all fixtures will be doused except the selected fixtures; they will be fully on. This function will help identify selected fixtures.

5.8   Info

The *Info* section displays the settings currently stored in the DALI fixture.

Figure 5.7: Info section

5.9 Status

The *Status* section displays possible faults reported by the DALI fixture.

Figure 5.8: Status section

Chapter 6

DMX Patch

The *DMX Patch* page is the starting point when using the DaliCore to control
DMX fixtures. This page is used for identifying which fixtures will be controlled.
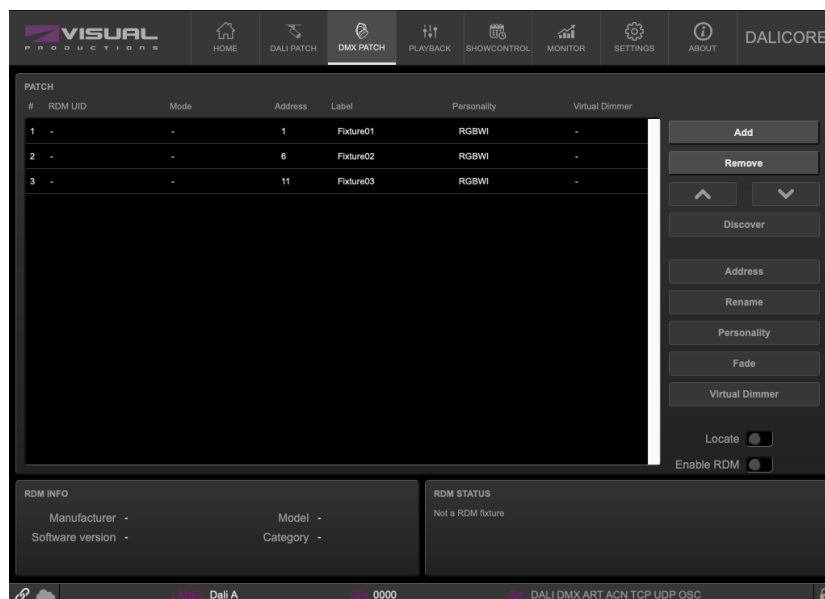Up to 32 fixtures can be entered.

Figure 6.1: DMX Patch Page

6.1 Address

DMX starting addresses range between 1 and 512. The DaliCore automati-
cally sets the DMX address when adding fixtures to the patch, by finding the
first available DMX channel in your configuration. You are free to change the
address.

If you have multiple fixtures selected, the DaliCore will set the entered address for the first selected fixture and will automatically have the next selected fixtures follow the previous ones. For example, if you select four RGB fixtures in the patch and set the address to 101, this will result in addresses 101, 104, 107 and 110 for the selected fixtures.

6.2 Personality

Each fixture requires a *personality*; a profile that matches the fixture's DMX traits. The personality can be composed out of the following parameters:

I	Intensity
R	Red
G	Green
B	Blue
C	Cold White
W	Warm White
A	Amber
Z	Zoom
F	Focus
S	Special

When a parameter is 16-bit it requires two DMX channels. In this case, the *fine* channel is denoted by the parameter character in small capitals. E.g. a 16-bit RGB fixture, occupying 6 DMX channels, would have the following personality `RrGgBb`.

A personality string can be entered directly in the patch table or by using the *Personality* dialog (figure 6.2).

6.3 Fade

Pressing the *Fade* button will bring up a dialog (figure 6.3) where a fade time can be set for the selected fixtures. This fade time will be applied to all scene changes.

6.4 Virtual Dimmer

Personalities with RGB but without Intensity will automatically be given a virtual dimmer. This means that the DaliCore will act as if the fixture has an
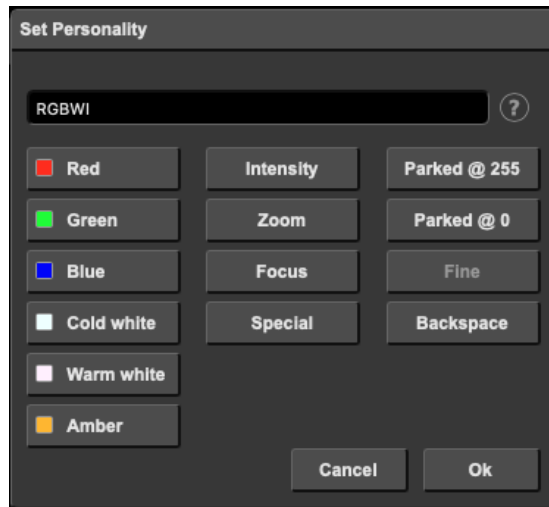
Figure 6.2: Personality dialog
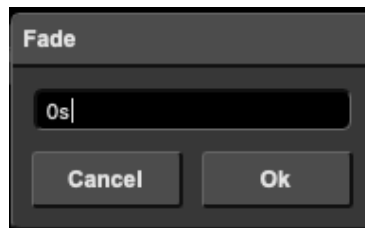
Figure 6.3: Fade dialog

intensity channel and allows for RGB and intensity to be programmed separately, however, it will modulate the intensity on the RGBCWA values.

6.5 Discover

Use the *Discover* button to search for RDM capable fixtures. Fixtures found via a RDM discovery will show their RDM UID in the patch table.

Before using RDM, RDM needs to be enabled. This can be done on the *Patch* or *Settings* page. By default RDM is disabled.

6.6 Locate

When the *Locate* checkbox is enabled, then all fixtures will be doused except the selected fixtures; they will be fully on. This function will help identify selected fixtures.

Chapter 7

Scene

The *Scene* page enables you to store and recall lighting scenes for both the DALI and DMX luminaires.

For the DALI, the scene and group data is stored inside the fixtures. Where as for DMX this data is stored in the DaliCore.
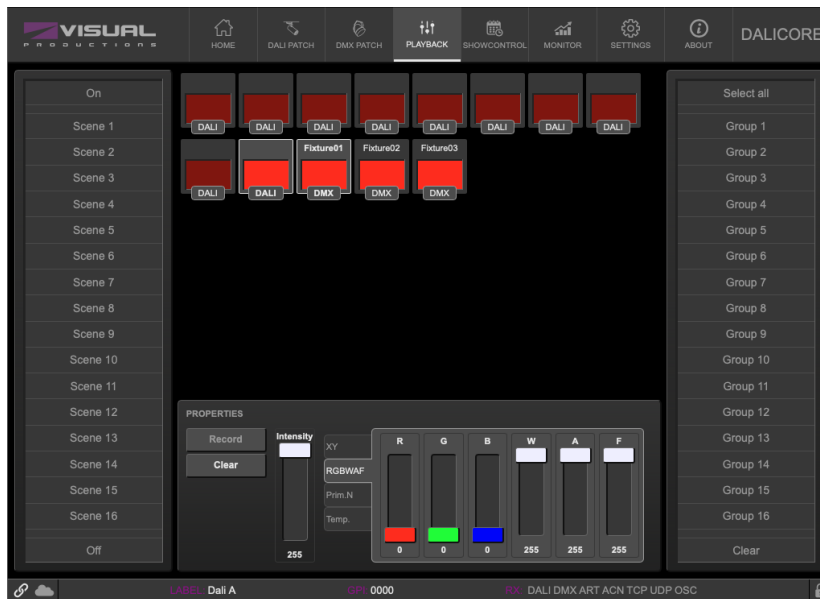
Figure 7.1: Scene Page

7.1   Scenes

There are 16 scenes. Each scene holds a certain lighting level for each fixture. It is also possible set a *don't care* value in a scene. In this case, when the scene is

called the luminaire does not change its levels. In DALI the value 255 reserved for *don't care*.

To program a scene, select some fixtures, set their levels and press the *Record* button and finally one of the scene buttons.

A scene is simply recalled by pressing its button (make sure the *Record* button is not active).

7.2   Groups

Luminaires can the organized into groups. There are 16 groups, each can be empty or hold one or luminaires.

To program a group, select some fixtures, press the *Record* button and finally one of the group buttons.

A group is simply recalled by pressing its button (make sure the *Record* button is not active).

Chapter 8

Show Control

The DaliCore can interact with the outside world; the contact-closures can be triggered or it can send and receive messages and values through various protocols. It is possible to automate the DaliCore by having it respond automatically to incoming signals. An example of this would be to activate a scene upon receiving a certain UDP message via the network.

Another option is integrating the DaliCore with other systems, by using the various protocols it can receive to trigger its functionality. The *Show Control* page (See figure 8.1) enables this kind of programming to be made.
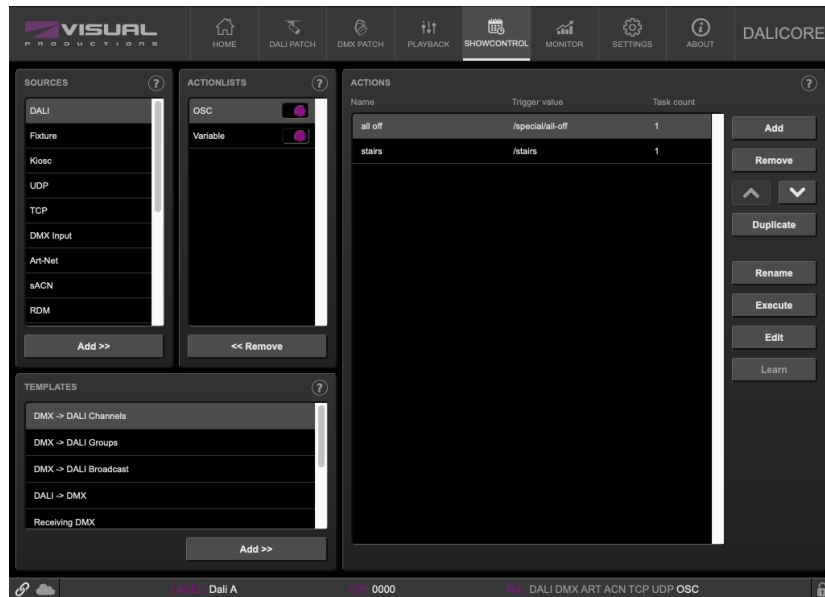

Figure 8.1: Show Control page

The *Show Control* page presents a system of *actions*. A signal that the DaliCore should to respond to or perhaps convert into some other signal, needs to be

expressed in an action. Before programming actions please consider the show control structure in figure 8.2.
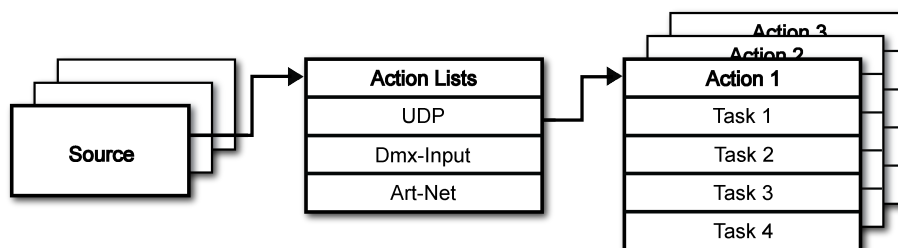

Figure 8.2: Show Control structure

The DaliCore is capable of listening to various protocols. These available protocols are listed in *Sources*, however, the DaliCore can only actively listen to 8 protocols at once. The active protocols are listed in *Actionlists*. Each actionlist can contain actions. Within a protocol/source each individual signal requires its own action. For example, when listening to channel 1 and 2 on the incoming DMX, the DMX actionlist needs two actions; one for each channel.

Inside the action we define the trigger and tasks. The trigger specifies for which signal to filter. In the above DMX example the trigger would be set to 'channel 1' and 'channel 2' respectively. The tasks determine what the DaliCore will do when this action is triggered. Several tasks can be placed in the action. There are tasks available for a wide range of DaliCore features and external protocols. Task types are detailed in Appendix B on page 74.

Please consult the API appendix on page 82 before implementing incoming OSC or UDP messages; the API already exposes typical functionality through OSC and UDP and therefor it might not be necessary to implement custom messages.

8.1   Sources and Actionlists

The *Sources* listing presents all protocols that the DaliCore is capable of receiving. It also includes internal features that can create events that can be used for triggering actions, such as the GPI ports. These sources are available, however, they will only be actively listened to once added to the actionlist table.

DALI	Received DALI messages.
Scene	Events generated by the scenes
Fixture	Events generated by the fixtures
Kiosc	Triggers from Kiosc. For each Action various controls can be chosen such as buttons and sliders, colour picker etc. The order of the actions will control the arrangement in Kiosc.
GPI	GPI port activity
OSC	OSC network messages
UDP	UDP network messages
TCP	TCP network messages
DMX Input	DMX received on the DMX port (switch port to *Input* in the *Settings* page)
Art-Net	Art-Net DMX data
sACN	sACN DMX data
RDM	RDM discovery result
System	Events such as 'Start up'
Timer	There are 4 internal timers in the DaliCore. An event can be raised when a timer starts or ends. Timers are set and activated by the Timer tasks.
Variable	There are 10 Variables in the DaliCore. The Variable source can trigger when the value of a variable is changed or refreshed. Variables are changed or refreshed using the Variable task.
Randomizer	The Randomizer source can trigger on a number randomly created by the Randomizer task.
Actionlist	Triggers whenever an Actionlist is enabled or disabled.
User List 1-4	These actionlists have no trigger value. They can only be triggered by the Action task. They are useful for advanced programming.

Actionlists can be temporarily suspended by disabling their checkbox in the *Show Control* page. There is also a task available to automate changing the state of this checkbox.

8.2   Actions

Actions are executed when a certain signal is received. This signal is defined by the trigger. A trigger is always relative to the actionlist the action belongs to. For example, when the trigger-type is set to 'Channel' then it refers to a single

DMX channel if the action is placed inside a 'DMX Input' list and it means a single Art-Net channel if the action resides in an Art-Net actionlist.

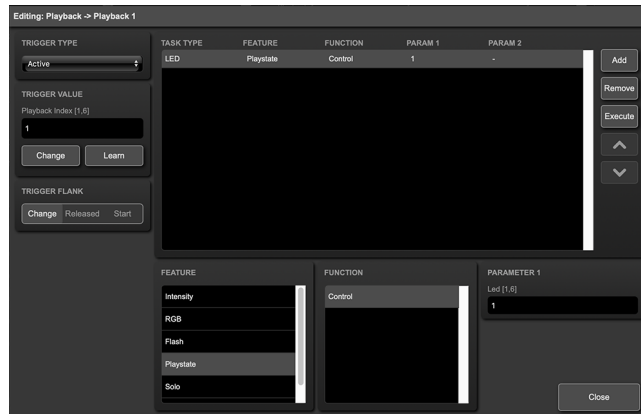Figure 8.3 shows the screen when editing an action.

Figure 8.3: Edit Action dialog

A trigger is determined by the trigger-type, trigger-value and trigger-flank fields. Although these fields are not applicable for all actionlists and are therefor sometimes omitted in the web GUI. The trigger-type field specifies what kind of signal the action will be triggered by.

For example, when making an action in the Art-Net list there is the choice between 'Channel' and 'Receiving' trigger-types. The trigger-value specifies the actual signal value. In the Art-Net example the trigger-value could be set to 'channel 1' or 'channel 2'.

In some actionlists actions do also need to specify the trigger-flank. The flank further specifies the value that the signal should have before triggering the action. For example, when an action is triggered from the Kiosc list, the flank will determine whether to trigger only when the button goes down or only when it goes up. Appendix A provides an overview of the available trigger-types.

An actionlist can have up to 48 actions, system-wide there is a maximum of 64 actions.

8.3   Tasks

Tasks are added to an action in order to specify what to do when it gets executed. Up to 8 tasks can be included in an action, systemwide there is a maximum of 128 tasks. The tasks are executed in the order of the list. There is a wide selection of tasks available to choose from, they include altering any of the internal software features like playbacks and recorder but also sending out messages through any of the supported protocols. The tasks are organised in categories. Once a task is chosen from these categories each task allows for further choice between several

Features and *Functions*. Tasks contain up to two parameters that might be required for its execution.

If the event that triggers the action passes a parameter along then this parameter can be used in a task. The *Set* function makes a task use a fixed value, however, when using the *Control* function then the trigger's parameter is used. This is very useful for conversions between protocols.

For example when converting 0-10V to DMX the GPI action specifies the port (e.g. #1) and flank (e.g. OnChange) on which it will trigger. The actual 0-10V level sampled on the GPI port will be passed along and fed into the action. Then when a task (e.g. DMX) uses the function *Control* this 0-10V level will be used for setting the DMX value.

A task can be tested by selecting it and pressing the 'execute' button in the *Edit Action* dialog. The complete action can also be tested; go to the *Show Control* page, select the action and press the *Execute* button.When either of these *Execute* buttons are used, the source of the *Control* value for tasks will be the *Execute* button. The result will depend on the chosen task and feature, but will most likely be 100%, 1.0, or 255 when pressed and 0%, 0.0, or 0 when depressed.

Appendix B provides a detailed overview of the available tasks, features, functions and parameters.

8.4   Templates

The *Show Control* page presents a list of templates. A template is a set of actionlists, actions and task. These templates configure the DaliCore to perform typical functions; for example convert Art-Net to DMX or control the 16 DALI scenes through OSC. The templates thus save time; otherwise actions should have been set up manually. They can also function as a guide to soften the learning curve on actions; a lot can be learned from adding a template and then exploring the actions and tasks it created.

Please note that some templates require settings to be changed in the settings page; for example the 'Receiving Art-Net' template needs the DMX outlets to be set to outputs in order to achieve an Art-Net to DMX conversion.

Appendix C gives an overview of the available templates.
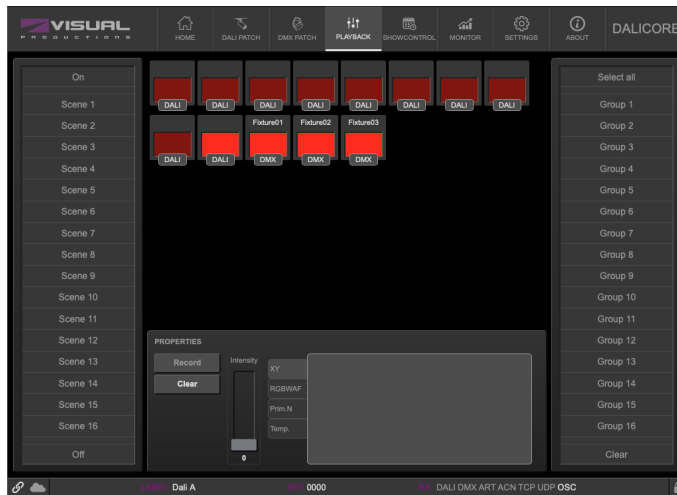
8.5   Variables

Variables are part of the show control system in the DaliCore. There are 10 variables and each can hold a value in the range of [0,255]. These values can be manipulated by tasks and can be used for advanced action programming. Variables can be added as sources in order to have actions triggered when a variable changes value.

You can see the status of the variables in the monitor page, as discussed on page 51.

Please note that the variable values are not stored between power cycles.

To illustrate the use of variables, please see the following example in which a variable is used to keep track of the last activated scene between power-cycles. (By default, the DaliCore does not remember which scene was active after a power-cycle.) For simplicity, we assume that an external system is selecting cues inside the DaliCore by sending UDP messages.

- Program 4 scenes on the *Scene* page.

- Add the *Variable* source and insert one action. If the variable changes then jump to the appropriate scene by adding a *Scene* task to this action.
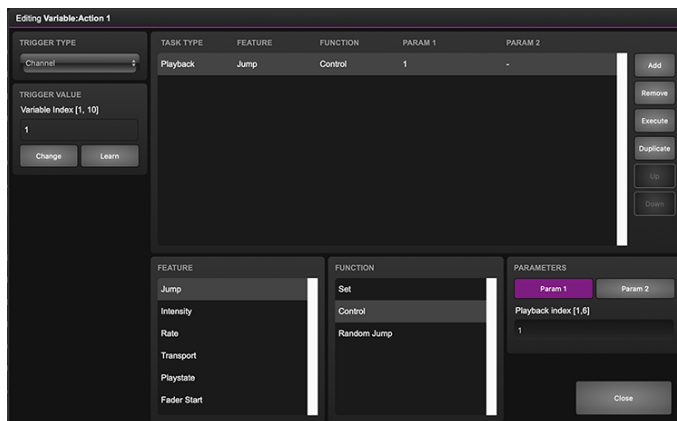
- Have the external system use the UDP API to set the variable values in order to select a cue. The appropriate API message is `core-va-1-set=<integer>` where <integer> is the scene number.
- Add the *System* source and insert one action. At start-up, trigger a variable 'change' in order to jump to the previously selected scene.

8.6   Timers

The show control system of the DaliCore features four internal timers. By using tasks, the timers can be set to certain durations and they can be started. Once started the timers will countdown to zero. When the timer r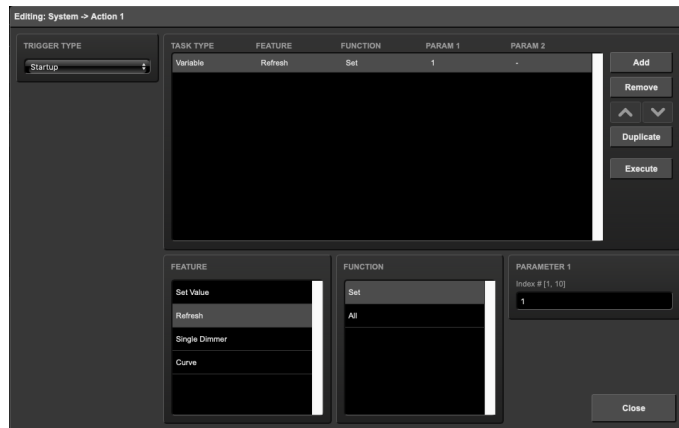eaches zero it will generate an event that can be captured by using the *Timer* actionlist. Please note that the timer values are not stored between power cycles. You can see the status of the four timers in the monitor page, under *Timers*.

The next example will show a timer being used to regularly send a UDP message to an external system in order to notify that the DaliCore is still 'alive'.

- Setup and start the timer at power-on. This is done by creating an action in the *System* actionlist.
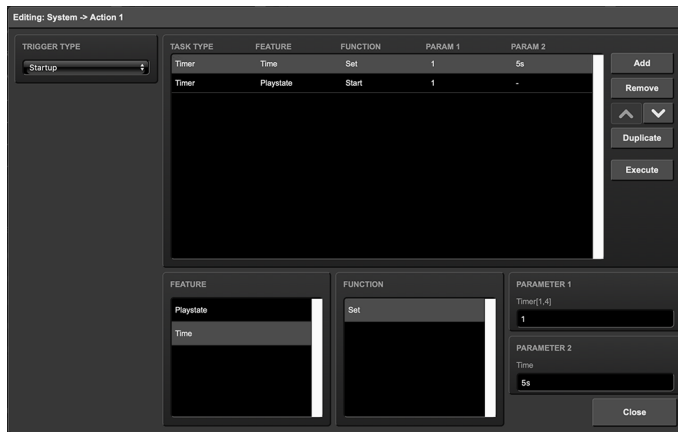
- If the timer expires then send the UDP message and restart the timer. This is done by creating an action in the *Timer* actionlist.
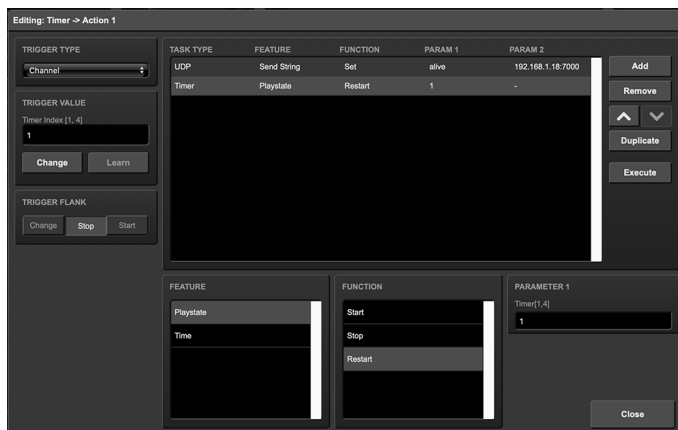
8.7 Randomizer

The *Randomizer* is an internal software feature that can generate a (pseudo-)random number. This is useful for having an event trigger a random lighting scene in a themed environment. The randomizer is activated by the Randomizer task. The result of the randomizer's calculation can be obtained by catching the event in the Randomizer actionlist.

The following example shows how to use a Kiosc button to trigger a random scene.

- Inside the Kiosc actionlist is a button-action. This triggers the task Randomizer, which is set to a range of between 1 and 6. (Parameters 1 and 2 of the task Randomizer)
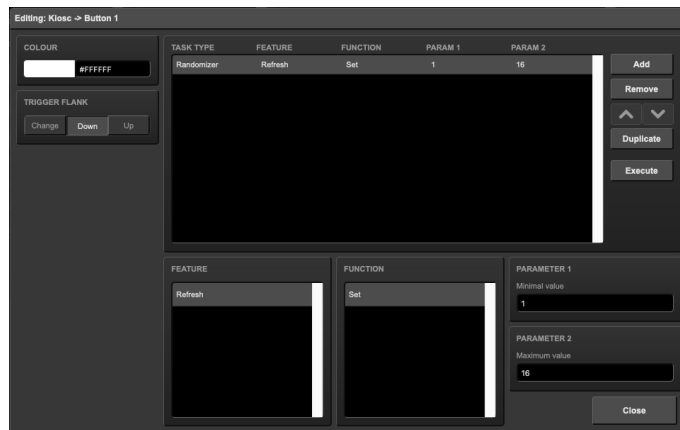
Kiosc will just simply show one button.
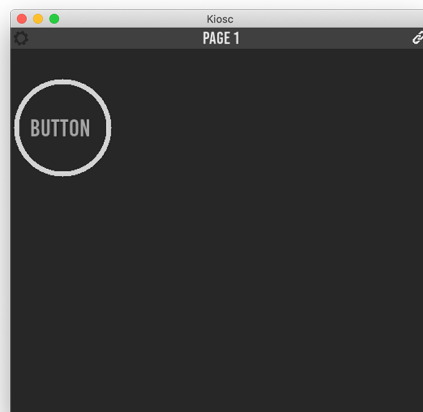
- Next in the actionlist Randomizer there is an action triggered by the Randomizer. The task *Scene*, controlled by the result of the Randomizer, jumps to a scene.
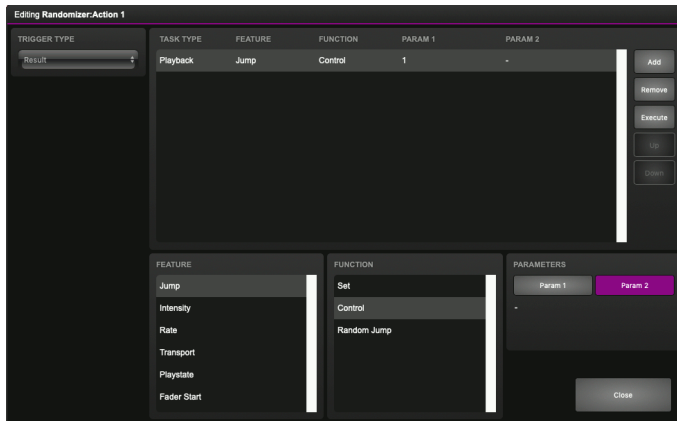
- When the Kiosc button is pressed the Randomizer picks a number between one and six, as defined in the Randomizer task. The *Scene* task receives this number and triggers the corresponding scene.
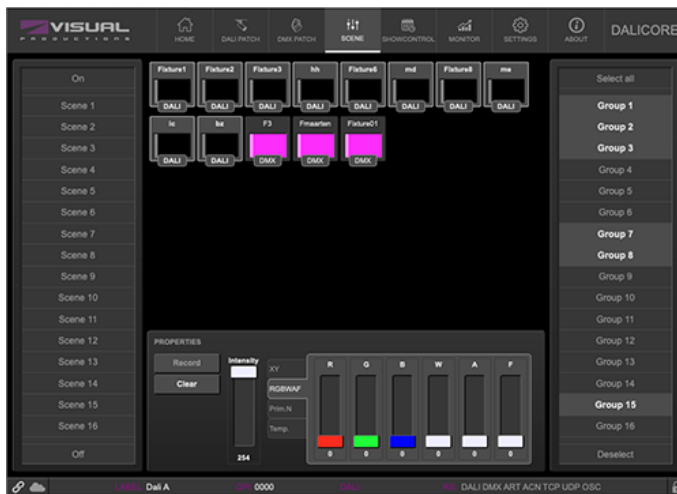
8.8 User Lists

Normally a source is connected to a communication port, protocol or software feature. Actionlists will be triggered as soon as an event happens in a source. There is an exception; the User Lists are not connected to any source and therefor will not be triggered by any communication or other event. The purpose of User Lists is to have extra actionlists containing actions which can be triggered by explicitly linking to it. See page 79 for details of the *Link* feature in the *Action* task.

The following example demonstrates the use of a User List. Imagine a scenario in which the GPI is used to change between three lighting scenes. To prevent unauthorised use a key-contact-closure is connected to GPI pin 4; the first three GPI should only respond when pin 4 is a closed contact by using the key.

- Program three scenes
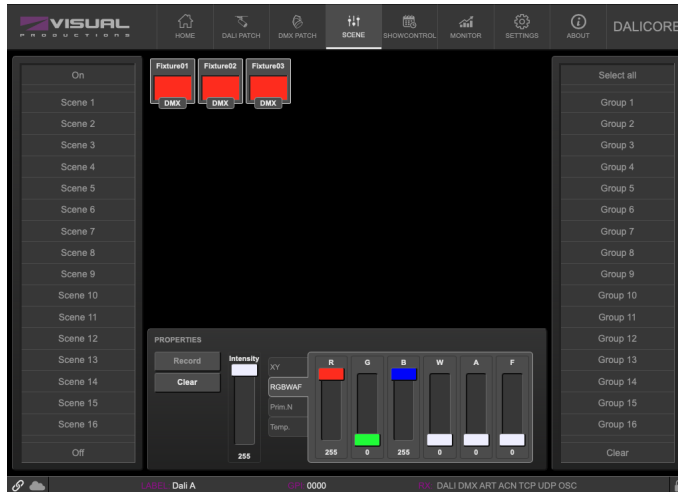
• Create an User List with three actions.
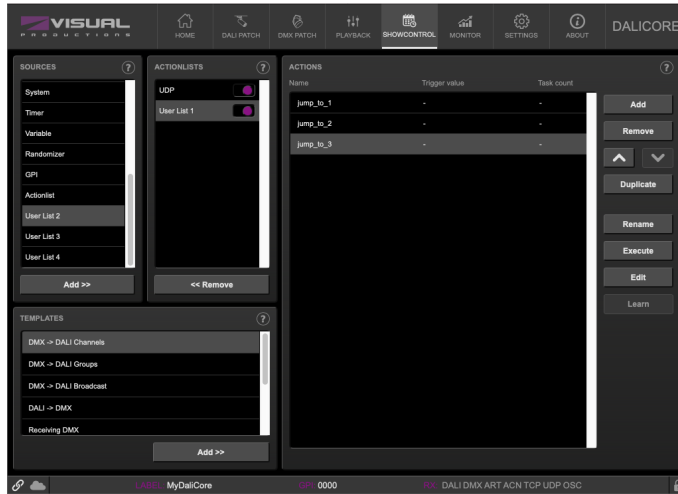
Set each action to jump to one of the three scenes.
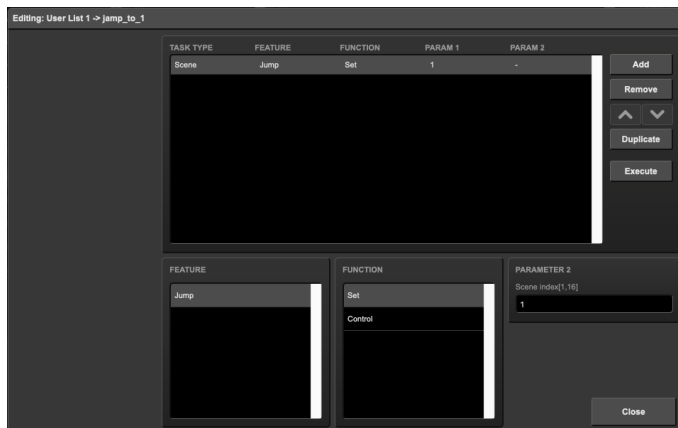
- Add an GPI actionlist and link the first three GPI ports to the action in the User List.
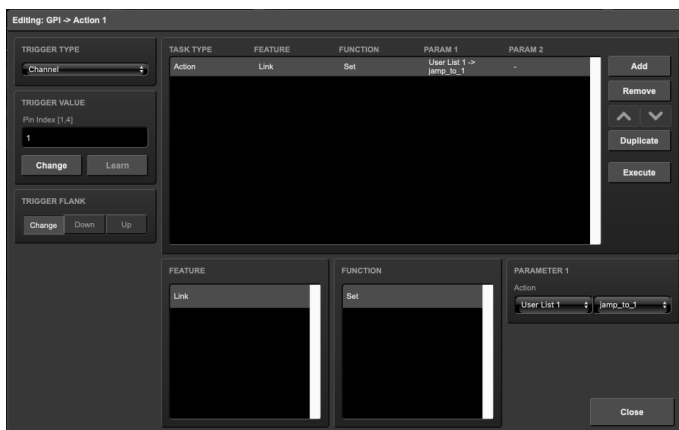
- Use the fourth GPI port enable or disable the User List.
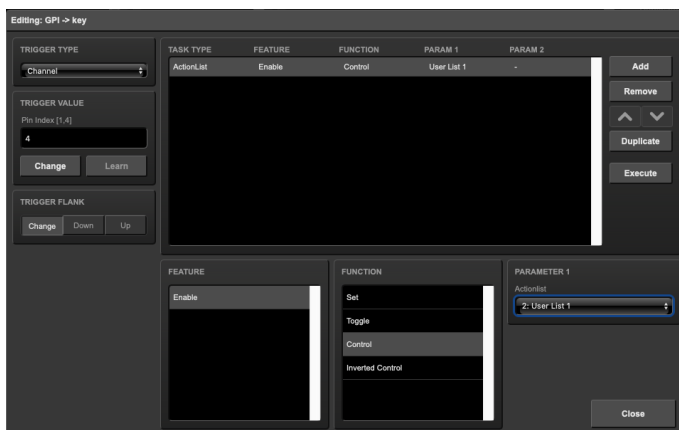
For brevity the state of the GPI port 4 is ignored at start-up. Usually additional actions are created to *Refresh* this port on start-up.

Chapter 9

Protocol Conversion

The DaliCore supports a wide range of communication protocols. The protocols vary from physically connected data busses like DALI and DMX, to Ethernet-based data-protocols such as UDP and OSC. It even extends to lighting protocols based on Ethernet, e.g. Art-Net and sACN.

The DaliCore is capable of converting one protocol into another. This chapter provides an insight on which conversions are possible and how to set them up.

9.1   Converting DMX and DALI

A typical function of the DaliCore is to convert DMX to DALI. Likewise, the unit can also convert DALI communication into DMX.

When converting from DMX to DALI it advised to consider which DALI commands are needed. The incoming DMX channels can be translated into *individual*, *group* or *broadcast* messages. The template section in the *Show Control* page offers ready-made templates for all three different messages. A user is free to make combinations of these messages. Please note that the DMX port should be set a an *Input* on the *Settings* page.

Incoming DALI messages can also be converted to DMX levels. Currently, only individual DALI messages are received. Each of these 64 messages can be translated to a DMX channel. A template is also offered for this translation. The DMX port should be configured as an *Output*.

9.2   Converting Control Protocols

This category of conversions comprise the protocols typically used for triggering or transporting one piece of information. The following table shows these protocols and what kind of information they are able to carry.

Control Protocols	Information
UDP	-
TCP	-
OSC	percentage [0%,100%], number, string, colour, On/Off
DMX	number [0,255]
Art-Net	number [0,255]
sACN	number [0,255]

Although DMX, Art-Net and sACN are dedicated lighting protocols and naturally fit in the next category, their individual channels lend themselves well for conveying control messages.

Setting up a conversion is done in the *Show Control* page. First add the incoming protocol from the *Sources* table into the *Actionlist* table. Then add an action to this new actionlist. Inside this action the trigger-flank field (if visible) should be set to *Change*; as this action should be triggered every time the incoming signal changes. Furthermore, a *task* need to be added, the task-type determines which protocol is the output of our conversion. It is important that the *Function* in this task is set to *Control*. This will make sure that the output is not a fixed value, rather it will output the information received from the incoming signal.

Figure 9.1 shows a conversion between DMX and OSC. This example assumes the DMX port is set to *Input* on the *Settings* page.

9.3   Converting DMX Universes

This category includes all protocols that carry a DMX Universe (a block of 512 DMX channels). These protocols are DMX, Art-Net, sACN and KiNet. The DaliCore is capable of receiving a complete DMX universe from one protocol and sending it out on a different protocol. Furthermore, it is able to merge DMX universes from multiple sources into one output protocol. All this is done with a minimal amount of configuration in the DaliCore. The following table lists examples of the conversions that can be made.

Example DMX Universe Conversions
DMX ->Art-Net
Art-Net ->DMX
DMX ->sACN
sACN ->DMX
Art-Net ->sACN

It is also possible to create combinations of the examples above. For instance
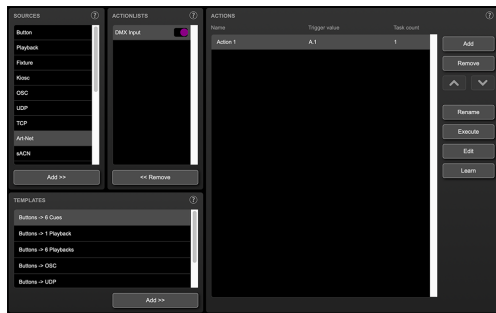
Step 1
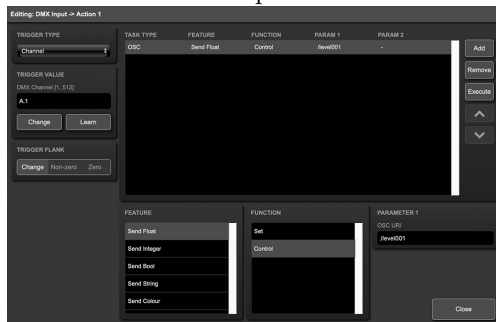
Step 2

Figure 9.1: Conversion from DMX to OSC

you could set up a conversion from DMX to both Art-Net and sACN. Or merge incoming Art-Net and sACN together into the DMX output. Also, at any point it is possible to merge the incoming DMX data with the data generated by the internal scenes.

To set up the conversion go to the *Show Control* page and choose the incoming protocol from the *Sources* table and add it to the *Actionlists* table. Then add an action for the DMX Universe you wish to convert. The trigger-type in the actions should be set to *UniverseA* to make the DaliCore process the 512 channels as a whole rather then process individual channels. The action should contain a DMX-task with the *Feature* set to *Universe*; all DMX Universe data is first being copied into the DaliCore's internal DMX buffer. From this buffer it can be copied to the DMX, Art-Net and/or sACN outlet using the function *Control HTP*. The different protocols can be enabled or disabled on the settings page. Figure 9.2 provides a schematic for this data flow.

The additional *Clear* function is not related to the data merging process; it is just a function to clear the whole universe to zero.

Please note that the *Templates* table provides pre-programmed configurations for the most popular conversions.

A very typical conversion that can illustrate as an example is to convert Art-Net universe 0.0 to the DMX output. Figure 9.3 shows the actionlist, figure 9.4 shows the contents of the action and figure 9.5 shows the required configuration
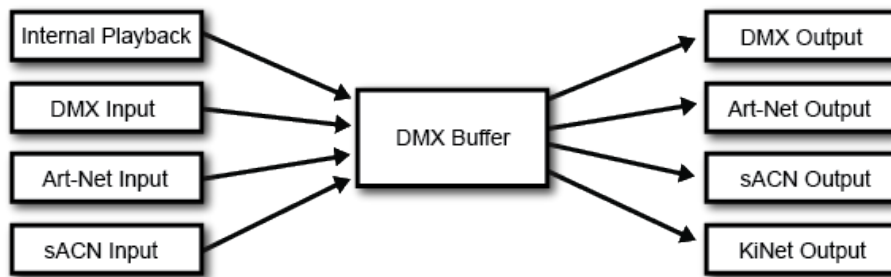
Figure 9.2: DMX merging data flow

of the Settings page.
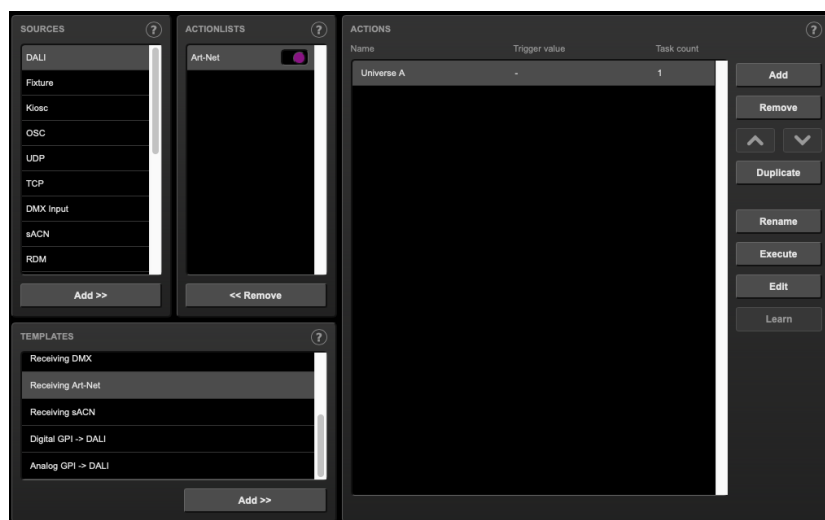
Figure 9.3: Converting Art-Net to DMX step 1
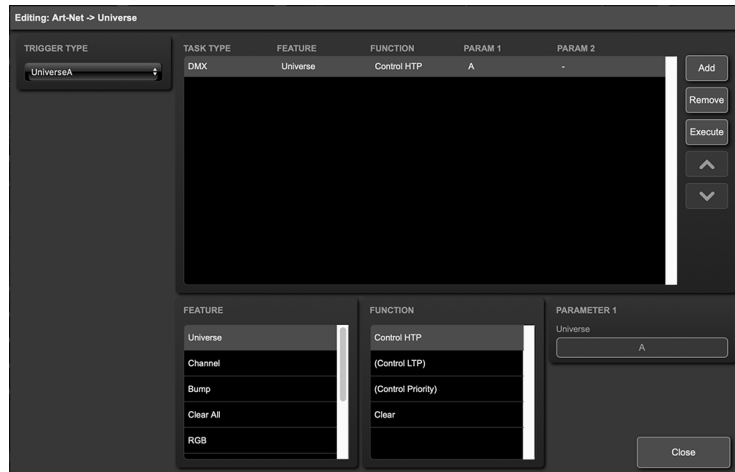
Figure 9.4: Converting Art-Net to DMX step 2
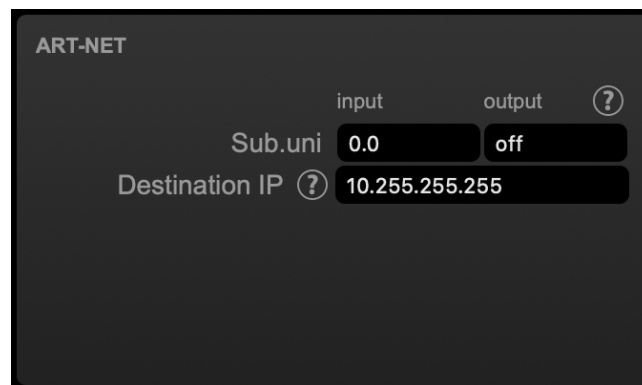
Figure 9.5: Converting Art-Net to DMX step 3

Chapter 10

Monitors

This page allows the user to inspect the incoming and outgoing data, both
DMX-type data (See figure 10.1) as well as control messages (See figure 10.2).
Monitoring incoming and outgoing data can help the user troubleshoot during
programming.
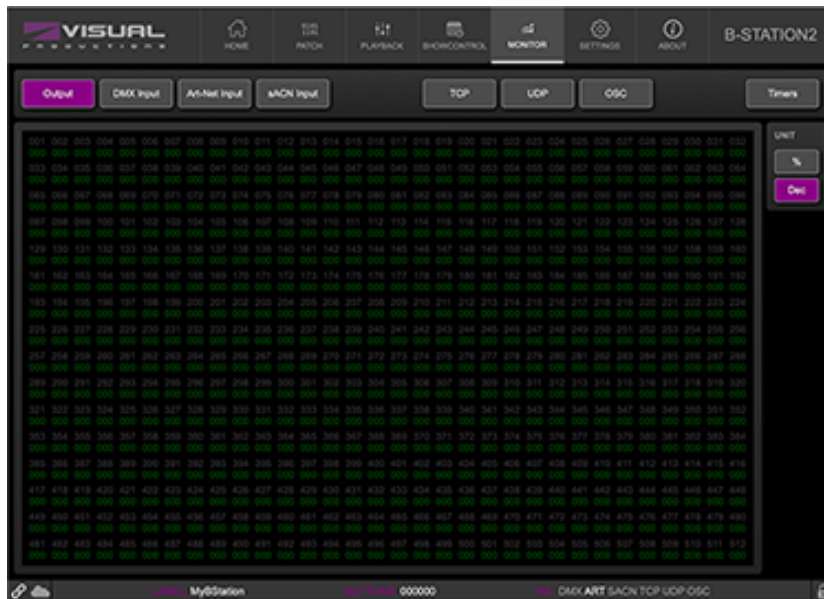
Figure 10.1: DMX Monitor page

On the monitor page the following information is available:

Output	The combined Playback, ShowControl, DMX, Art-Net and sACN output.
DMX Input	The dmx signal physically inserted in the DMX port
Art-Net Input	The signal received over Art-Net
sACN Input	The signal received over sACN
TCP	The messages received over TCP
UDP	The messages sent or received over UDP
OSC	The messages sent or received over OSC
Timers	The internal timers and variables

For the DMX, Art-Net and sACN inputs and outputs the preferred unit for displaying the information can be chosen on the right.

Figure 10.2: OSC Monitor page

Chapter 11

Settings

The DaliCore's settings are organised into sections, see the *Settings* page figure 11.1. This chapter will discuss each section.

Figure 11.1: Settings page

11.1   General

You can change the DaliCore's label. This label can be used to distinguish the unit in a set-up with multiple devices.

By enabling the *Blink* checkbox the device's LED will blink to help to identify it amongst multiple devices.

The API commands discussed in appendix D start with a prefix that is set to *core* by default. When using multiple devices from Visual Productions it can be
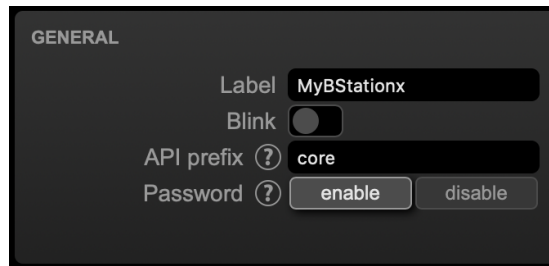
Figure 11.2: General Settings

useful to assign unique labels to these prefix, especially when using broadcasted messages. Read more about feedback loops in paragraph D.4.1.

Unauthorised users can be prevented from making changes to the DaliCore by enabling the *Password* protection. Once enabled, the password can be disabled via the web-interface (using the *Disable* button) and the reset button (see figure 4.2). Long-press the reset button to disable the password protection; this will also revert the unit's static IP back to the default factory settings.

11.2 IP

The IP fields are for setting up the IP address and subnet mask of the DaliCore. The *Router* field is only required when Port Forwarding is used. You can also enable or disable the DHCP feature (For more information see chapter 4 at page 19).
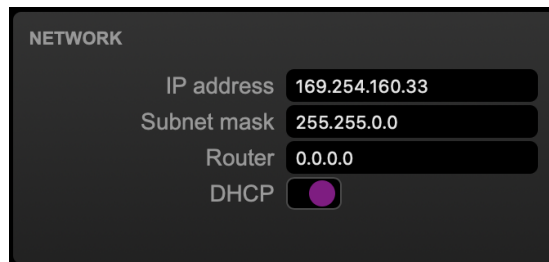
Figure 11.3: IP Settings

11.3 DMX

The *DMX* settings specify whether the DMX port is an *Input* or *output.*
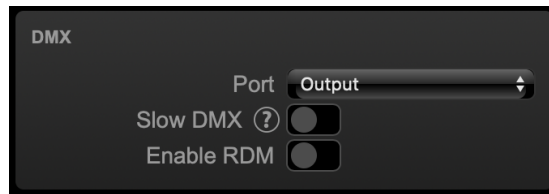
Figure 11.4: DMX settings

When the *Slow DMX* checkbox is enabled, the DaliCore will slowdown the rate at which it sends out DMX. This is done to facilitate DMX fixtures that have difficulties keeping up with the optimal DMX transmission rate.

RDM can be disabled by using the *Enable RDM* checkbox. This checkbox is repeated on the *Patch* page.

11.4 Art-Net

The *Art-Net* feature in the DaliCore supports 1 universe out and 1 universes in.

These universes can be mapped to any of the 256 available universes in the Art-Net protocol. The universe is entered in the 'subnet.universe' format, i.e. the lowest universe number is written as '0.0' and the highest universe number is denoted as '15.15'. The outgoing Art-Net transmission can be disabled by entering 'off' in the output fields.
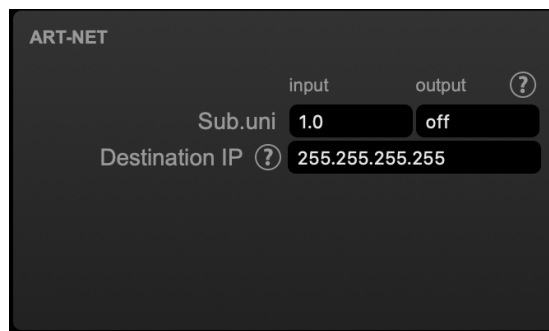
Figure 11.5: Art-Net settings

The *Destination IP* determines where the outgoing Art-Net data will be sent to. Typically, this field contains a broadcast address like 2.255.255.255 which will send the Art-Net data to the 2.x.x.x IP range. Another typical Art-Net broadcast address is 10.255.255.255. When using broadcast address 255.255.255.255 then all the devices on the network will receive the Art-Net data.

It is also possible to fill in a unicast address like 192.168.1.11; in this case the Art-Net data will be send to one IP address only. This keeps the rest of the network clean of any Art-Net network messages.

11.5 sACN

The DaliCore supports 1 incoming *sACN* universe and 1 outgoing universe. The universe field should hold a number in the range of [1,63999]. Outgoing sACN transmission can be disabled by entering *off* into the sACN output field.

Figure 11.6: sACN settings

Set priority field to control the priority level of the output universe. The priority can be set from 0 (lowest) to 200 (highest).

11.6 OSC

External equipment sending OSC messages to the DaliCore need to be aware of the number specified in the 'Port' field. This is the port the DaliCore listens to for incoming messages.
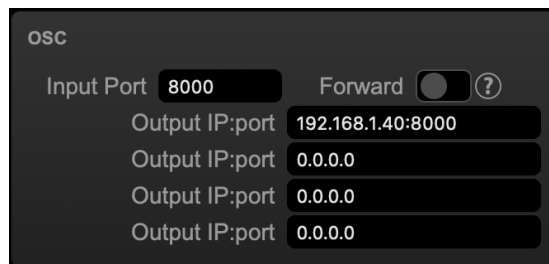
Figure 11.7: OSC Settings

The DaliCore will send its outgoing OSC messages to the IP addresses specified in the 'Out IP' fields. Up to four IPs can be specified here. Use the 'ip-address:port' format in these fields, e.g. "192.168.1.11:9000". If a field should not be used then it can be filled with IP 0.0.0.0:0. It is possible to enter a broadcast IP address like 192.168.1.255 in order to reach more than four recipients.

Enabling the *Forward* checkbox will have the DaliCore copy every incoming OSC message and send it the addresses specified in the 'Out IP' fields.

11.7   TCP/IP

Defines the listening ports for TCP and UDP messages. External systems intending to send TCP or UDP message to the DaliCore need to know the unit's IP address and this port number. By default both ports are set to 7000.
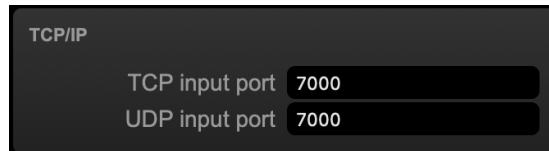
Figure 11.8: TCP/IP settings

11.8   Cloud

By enabling the *Share Analytics* options, you will help Visual Productions improve its products. When enabled, the DaliCore will send low-bandwidth diagnostics and usage data to Visual Productions. This information remains anonymous.

The *Cloud ID* is not implemented yet, and is for future functionality.
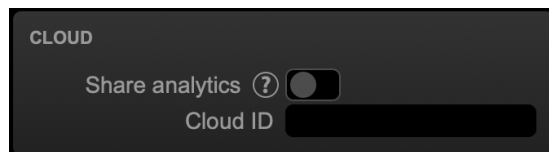
Figure 11.9: Cloud settings

11.9   GPI

The GPI ports can be configured either as digital (contact-closure) or as analog (0-10V) input.

In case a port is set as analog and the signal supplied falls within the 0-10V range (e.g. 1-5V) then it is recommended to calibrate to port in order to map the minimal voltage to a logical 0% and map the maximum voltage to 100%.

To perform a calibration first enable the Calibrate button, then bring the incoming signal to the minimum level, bring the signal to the maximum level and then close the calibration process by clicking the Calibrate button again.

Do not apply more than 10V to the GPI port, as this could cause permanent damage.

Each port has a label field that can contain a user-description.
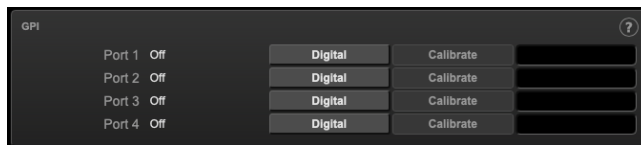
Figure 11.10: GPI Settings

Chapter 12

vManager

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

- Setup the IP address, subnet mask, router and DHCP

- Backup and restore the device's internal data and settings

- Perform firmware upgrades

- Identify a specific device (in a multi device set-up) by blinking its LED

- Revert to factory defaults
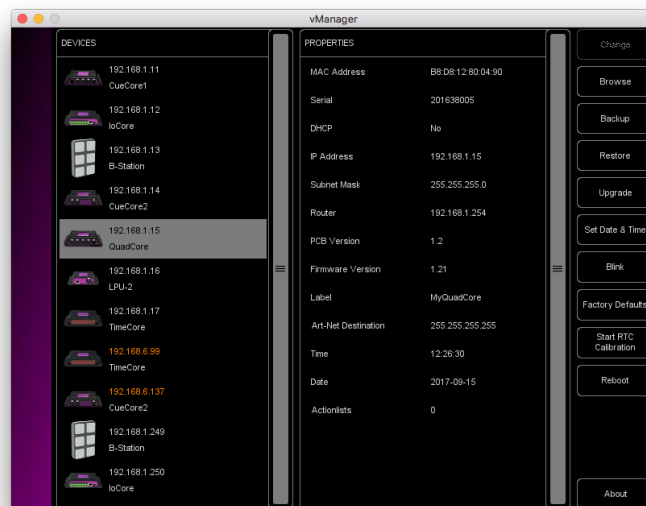
Figure 12.1: vManager

The following section explain the buttons in the vManager, as seen in figure 12.1.

12.1   Backup

Backups of all the programming data inside the device can be made. This backup file (an XML) 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.
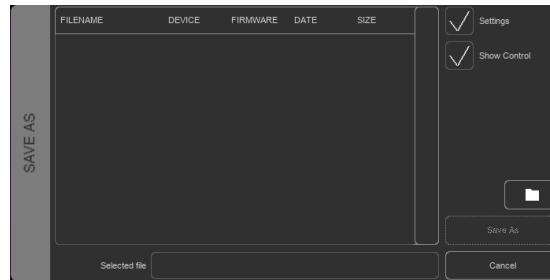
Figure 12.2: Creating a backup

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

The designated file location differs per operating system and is likely to 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 appropriate folder.

12.2   Upgrade Firmware

To upgrade the firmware, first select the device and press the *Upgrade* Firmware button. The dialogue allows for selecting from the list of firmware versions available.
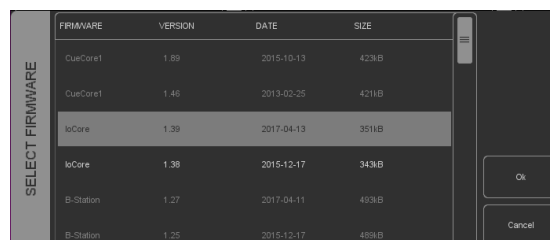
Figure 12.3: Firmware upgrade

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

12.3   Set Date & Time

The computer's date and time can be quickly copied to the unit by selecting a device and clicking the *Set Date & Time* button. Not all Visual Productions devices feature an internal real-time clock. The DaliCore does not have such a RTC.

12.4   Blink

The device's LED can be set to *Blink* fast for identifying the particular unit amongst multiple devices. The blinking is enabled by double-clicking on a device in the Devices list or by selecting a device and then clicking the Blink button.

12.5   Factory Defaults

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

12.6   Reboot

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

12.7   Installing vManager

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

The softwares is distributed through app-stores to take advantage of receiving future software updates automatically.

12.7.1   iOS

vManager can be downloaded from the Apple iOS app-store at `https://itunes.apple.com/us/app/vman/id1133961541`.

12.7.2 Android

vManager can be found on the Google Play store at `https://play.google.com/store/apps/details?id=org.visualproductions.manager`.

Android 5.0 or higher is required.

12.7.3 Windows

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

Windows 10 is required.

12.7.4 macOS

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

macOS 11.3 is recommended.

12.7.5 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 20.04 LTS is recommended. The software is only available for the amd64 architecture.

Chapter 13

Kiosc

Kiosc is an application for creating custom touch screen user-interfaces for the range of lighting controllers from Visual Productions. Kiosc is designed to have no editing capability, making it a fool-proof interface that can safely be presented to non-technical operators.
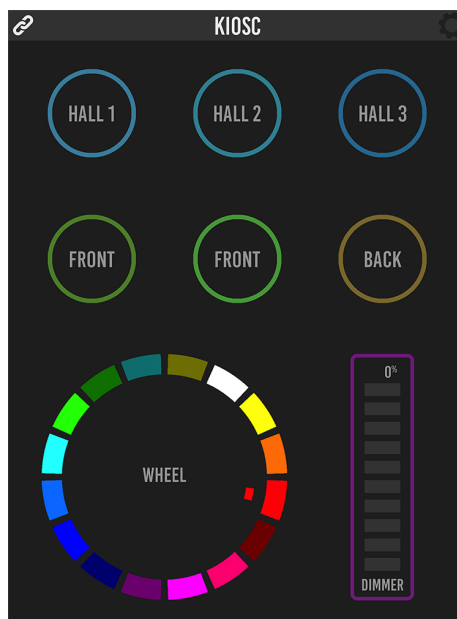
Figure 13.1: Kiosc

Kiosc is the ideal way to remote control our solid-state lighting controllers like CueluxPro, CueCore1, CueCore2, QuadCore, IoCore1, IoCore2, LPU-2, DaliCore, B-Station1 and the DaliCore. Kiosc enables you to choose scenes or presets, set intensity levels or choose RGB colours.

You can also use it to control third-party AV equipment. Kiosc speaks UDP and OSC.

Kiosc is available as software app and as a physical product. The hardware version of Kiosc is a wall-mount 7" touch screen with Kiosc pre-installed. It is powered by PoE and requires only a RJ-45 connection.

Figure 13.2: Kiosc

Please read the Kiosc manual, available from `http://www.visualproductions.nl/downloads` for more details.

Appendices

Appendix A

Trigger Types

The following tables list the different types of triggers that can be used in the DaliCore. The different types are accompanied with values and flanks.

A.1 Scene

Trigger Type	Trigger Value	Flank	Description
-	Scene #	Change	

A.2 DALI

Trigger Type	Trigger Value	Flank	Description
Intensity	Fixture	Change	
Intensity	Fixture	Down	
Intensity	Fixture	Up	
Scene	Fixture	Change	
Scene	Fixture	Down	
Scene	Fixture	Up	
Lamp Failure	Fixture	Change	
Lamp Failure	Fixture	Down	
Lamp Failure	Fixture	Up	

A.3 GPI

Trigger Type	Trigger Value	Flank	Description
Channel	Port number	Change	Port state changes
Channel	Port number	Down	Port is closed
Channel	Port number	Up	Port is opened
Channel 0-19%	Port number	Change	Analog level enters or leaves the range
Channel 0-19%	Port number	Down	Analog level enters the range
Channel 0-19%	Port number	Up	Analog level leaves the range
Channel 20-39%	Port number	Change	Analog level enters or leaves the range
Channel 20-39%	Port number	Down	Analog level enters the range
Channel 20-39%	Port number	Up	Analog level leaves the range
Channel 40-59%	Port number	Change	Analog level enters or leaves the range
Channel 40-59%	Port number	Down	Analog level enters the range
Channel 40-59%	Port number	Up	Analog level leaves the range
Channel 60-79%	Port number	Change	Analog level enters or leaves the range
Channel 60-79%	Port number	Down	Analog level enters the range
Channel 60-79%	Port number	Up	Analog level leaves the range
Channel 80-100%	Port number	Change	Analog level enters or leaves the range
Channel 80-100%	Port number	Down	Analog level enters the range
Channel 80-100%	Port number	Up	Analog level leaves the range
Binary	Combination value	-	A combination of ports being closed
Short press	Port number	-	Short closure on port
Long press	Port number	-	Long closure on port

Use the Binary trigger type to catch port combinations when they are set as
digital. Specify the port combination by adding the values that correspond to
the ports (see table below). This value is entered as the trigger value.

Port	Value
1	1
2	2
3	4
4	8

For example, to trigger on both port 1 and 2 being closed fill in trigger value 3
(1+2). To trigger on port 3 and 4, fill in value 12 (4+8).

A.4 DMX Input

Trigger Type	Trigger Value	Flank	Description
Channel	DMX Address	Change	Channel changes
Channel	DMX Address	Non-zero	Channel becomes non-zero
Channel	DMX Address	Zero	Channel becomes zero
UniverseA	-	-	A DMX level change in the first universe
UniverseB	-	-	A DMX level change in the second universe
Receiving	-	Change	Start receiving or loose DMX signal
Receiving	-	Stop	Lost DMX signal
Receiving	-	Start	Start receiving DMX signal

A.5 UDP

Trigger Type	Trigger Value	Flank	Description
Message	String	-	Receive a message that matches the trigger-value
Receiving	-	-	Receive any message

The user can define his own string as the trigger value of a message. Please note that this string has a maximum length of 31 characters.

It is possible to pass a parameter along with a message. In order to do this use the syntax `trigger=value`. For example when the trigger type is set to 'message' and the trigger value is set to `intensity` then the transmitting equipment can pass an intensity level by sending `intensity=255`, where 255 can be any number in the range [0,255].

A.6 TCP

Trigger Type	Trigger Value	Flank	Description
Message	String	-	Receive a message that matches the trigger-value
Receiving	-	-	Receive any message

The user can define his own string as the trigger value of a message. Please note that this string has a maximum length of 31 characters.

A.7   OSC

Trigger Type	Trigger Value	Flank	Description
Message	URI	Change	Receive a message that matches the URI
Message	URI	Down	Receive a message that matches the URI and the value non-zero
Message	URI	Up	Receive a message that matches the URI and the value is zero
Receiving	-	-	Receive any message

The user can define his own URI as the trigger value of a message, however, the OSC specification dictate this string must start with a '/' sign. Please note that this string has a maximum length of 31 characters, including the '/'.

A.8   Art-Net

Trigger Type	Trigger Value	Flank	Description
Channel	DMX Address	Change	Channel changes
Channel	DMX Address	Non-zero	Channel becomes non-zero
Channel	DMX Address	Zero	Channel becomes zero
UniverseA	-	-	A DMX level change in the first universe
UniverseB	-	-	A DMX level change in the second universe
Receiving	-	Change	Start receiving or loose Art-Net signal
Receiving	-	Stop	Lost Art-Net signal
Receiving	-	Start	Start receiving Art-Net signal

A.9 sACN

Trigger Type	Trigger Value	Flank	Description
Channel	DMX Address	Change	Channel changes
Channel	DMX Address	Non-zero	Channel becomes non-zero
Channel	DMX Address	Zero	Channel becomes zero
UniverseA	-	-	A DMX level change in the first universe
UniverseB	-	-	A DMX level change in the second universe
Receiving	-	Change	Start receiving or loose sACN signal
Receiving	-	Stop	Lost sACN signal
Receiving	-	Start	Start receiving sACN signal

A.10 Kiosc

Trigger Type	Trigger Value	Flank	Description
-	-	Change	Button/Fader goes up or down
-	-	Down	Button is pressed
-	-	Up	Button is released

When editing the Kiosc actionlist it will be possible to add different kind of actions such as Button, Fader and Colour Picker. These elements will be displayed in the Kiosc software.

A.11 Randomizer

Trigger Type	Trigger Value	Flank	Description
Result	-	-	The Randomizer made a new value
Specific Value	Number in the range of [0,255]	-	The Randomizer made a value that matches

A.12   System

Trigger Type	Trigger Value	Flank	Description
Startup	-	-	The DaliCore has been powered up
Network Connection	-	Change	Network connection established or lost
Network Connection	-	Stop	Network connection lost
Network Connection	-	Start	Network connection established

Use *Startup* if something needs to happen as soon as the DaliCore is powered on. Please note that ethernet network might not be online yet.

Use the *Network Connection* trigger if something needs to happen as soon as the ethernet network becomes online after a power-cycle.

A.13 Variable

Trigger Type	Trigger Value	Flank	Description
Channel	Variable Index	-	The specified variable changes
Variable 1	Number [0,255]	Change	Variable 1 becomes = or # to the specified number
Variable 1	Number [0,255]	Equal	Variable 1 = number
Variable 1	Number [0,255]	Stop Equal	Variable 1 stops to be = to number
Variable 2	Number [0,255]	Change	Variable 2 becomes = or # to the specified number
Variable 2	Number [0,255]	Equal	Variable 2 = number
Variable 2	Number [0,255]	Stop Equal	Variable 1 stops to be = to number
Variable 3	Number [0,255]	Change	Variable 3 becomes = or # to the specified number
Variable 3	Number [0,255]	Equal	Variable 3 = number
Variable 3	Number [0,255]	Stop Equal	Variable 1 stops to be = to number
...			
Variable 9	Number [0,255]	Change	Variable 9 becomes = or # to the specified number
Variable 9	Number [0,255]	Equal	Variable 9 = number
Variable 9	Number [0,255]	Stop Equal	Variable 1 stops to be equal to number
Variable 10	Number [0,255]	Change	Variable 10 becomes = or # to the specified number
Variable 10	Number [0,255]	Equal	Variable 10 = number
Variable 10	Number [0,255]	Stop Equal	Variable 1 stops to be = to number

A.14 Timer

Trigger Type	Trigger Value	Flank	Description
Channel	Timer Index	Change	The timer starts or stops
Channel	Timer Index	Stop	The timer stops
Channel	Timer Index	Start	The timer starts
Time	Timer Index	-	A Stream of the current time of that timer

The timer will automatically stop when it reaches 00:00.0. A action needs to be programmed in order to restart the timer and make it loop.

A.15 Actionlist

Trigger Type	Trigger Value	Flank	Description
-	Actionlist Index	Change	The actionlist enable checkbox changes
-	Actionlist Index	Down	The actionlist is enabled
-	Actionlist Index	Up	The actionlist is disabled

A.16 User List (1-4)

User lists have no triggers. Actions inside user lists can only be activated by other actions through 'Action' task with the 'Link' feature.

Appendix B

Task Types

Tasks allow you to automate the functionality in the DaliCore. All this functionality is categorized in task-types. This appendix provides a listing of the various task-types. The tables present an overview of all available features and functions per task-type.

B.1 Scene

Call the DaliCore's scenes for DALI and DMX fixtures.

Feature	Function	Parameter 1	Parameter 2
Jump	Set	Fixture [1,64]	Scene [1,32]
Jump	Control	Fixture [1,64]	-

B.2 DALI

Send DALI commands.

Feature	Function	Parameter 1	Parameter 2
Intensity	Set	Fixture [1,64]	Level [0,254]
Intensity	Control	Fixture [1,64]	-
Jump	Set	Fixture [1,64]	Scene [1,16]
Jump	Control	Fixture [1,64]	-

This task will send messages directly to the DALI network. When using the *Jump* feature to call a scene it will bypass the DMX fixtures (they are controlled from the DaliCore's logic).

B.3   Fixture

Manipulate the fixtures directly.

Feature	Function	Parameter 1	Parameter 2
Intensity	Set	Fixture [1,32]	Percentage [0%,100%]
Intensity	Control	Fixture [1,32]	-
Intensity	Decrement	Fixture [1,32]	Delta [1,255]
Intensity	Increment	Fixture [1,32]	Delta [1,255]
Colour	Set	Fixture [1,32]	Colour
Colour	Control	Fixture [1,32]	-
Cold	Set	Fixture [1,32]	Percentage [0%,100%]
Cold	Control	Fixture [1,32]	-
Warm	Set	Fixture [1,32]	Percentage [0%,100%]
Warm	Control	Fixture [1,32]	-
Amber	Set	Fixture [1,32]	Percentage [0%,100%]
Amber	Control	Fixture [1,32]	-
UV	Set	Fixture [1,32]	Percentage [0%,100%]
UV	Control	Fixture [1,32]	-
Shutter	Set	Fixture [1,32]	Percentage [0%,100%]
Shutter	Control	Fixture [1,32]	-
Zoom	Set	Fixture [1,32]	Percentage [0%,100%]
Zoom	Control	Fixture [1,32]	-
Focus	Set	Fixture [1,32]	Percentage [0%,100%]
Focus	Control	Fixture [1,32]	-

B.4   UDP

Send a UDP message via the network. Specify the recipient in Parameter 2. For example "192.168.1.11:7000".

Feature	Function	Parameter 1	Parameter 2
Send Float	Set	floating point number	IP address & port
Send Float	Control	-	IP address & port
Send Unsigned	Set	positive number	IP address & port
Send Unsigned	Control	-	IP address & port
Send Bool	Set	true or false	IP address & port
Send Bool	Control	-	IP address & port
Send String	Set	text string	IP address & port
Send String	Control	-	IP address & port
Wake On Lan	Set	MAC Address	IP address & port

Please note that string in parameter 1 has a maximum length of 31 characters.

When using the Wake On Lan feature parameter 1 should contain the MAC Address of system's NIC (Network Interface Controller) you wish to wake up. The recommended value for parameter 2 is `255.255.255.255:7`. This broadcasts the message to the whole network at port 7 which is most commonly used for Wake On Lan.

B.5   OSC

Send an OSC message via the network. The OSC recipients are specified in the Settings page.

Feature	Function	Parameter 1	Parameter 2
Send Float	Set	URI	floating point number
Send Float	Control	URI	-
Send Unsigned	Set	URI	positive number
Send Unsigned	Control	URI	-
Send Bool	Set	URI	true or false
Send Bool	Control	URI	-
Send String	Set	URI	String of characters
Send String	Control	URI	-
Send Colour	Set	URI	RGB colour
Send Colour	Control	URI	-

Please note that string in parameter 1 has a maximum length of 31 characters, including the compulsory leading '/' sign.

B.6   DMX

Manipulate the DMX levels. These are the levels that can also be sent out via Art-Net or sACN.

Feature	Function	Parameter 1	Parameter 2
Universe	Control HTP	Universe #	-
Universe	Clear	Universe #	-
Channel	Set	DMX Channel	DMX Value
Channel	Toggle	DMX Channel	-
Channel	Control	DMX Channel	-
Channel	Inverted Control	DMX Channel	-
Channel	Decrement	DMX Channel	-
Channel	Increment	DMX Channel	-
Bump	Set	DMX Channel	DMX Value
Bump	Control	DMX Channel	-
Clear	All	-	-
RGB	Set	DMX Address	RGB Colour Value
RGB	Control	DMX Address	-
RGBW	Control	DMX Address	-
XY	Control	DMX Address	-
XxYy	Control	DMX Address	-
Ii	Set	DMX Address	Intensity Value
Ii	Control	DMX Address	-
Block	Control	DMX address and footprint	Fixture count
Block RGBW	Control	DMX address and footprint	Fixture count
Block CW	Control	DMX address and footprint	Fixture count
Block CW	Inverted Control	DMX address and footprint	Fixture count

The *Bump* feature will momentarily set the value of a DMX channel and then take it back to 0, effectively creating a pulse.

XY will take an incoming position and translate it to two DMX channels. *XxYy* will translate it to four DMX channels; a 16-bit pan and a 16-bit tilt level.

The *Ii* feature can take an incoming value and translate it to two-channel DMX 16-bit value.

Block can set a number of DMX channels to the same value. Parameter 1 will denote the starting channel and parameter 2 determines how many channels will be set. The footprint field in parameter 1 offers the possibility to have 'space' between the channels.

The *Block RGBW* feature is similar to the normal *Block*. The difference is that it takes RGBW colour which will be converted to four DMX channels. Therefor, the footprint should at least be set to 4.

The *Block CW* feature is similar to the normal *Block*. The difference is that it takes a level value which will be converted to two DMX channels. Therefor, the footprint should at least be set to 2.

B.7 Variable

Manipulate one of the 10 variables.

Feature	Function	Parameter 1	Parameter 2
Set Value	Set	Variable #	Number in the range of [0,255]
Set Value	Toggle	Variable #	Number in the range of [0,255]
Set Value	Control	Variable #	-
Set Value	Inverted Control	Variable #	-
Set Value	Decrement	Variable #	-
Set Value	Increment	Variable #	-
Set Value	Stop Continuous	Variable #	-
Set Value	Continuous Decrement	Variable #	Delta
Set Value	Continuous Increment	Variable #	Delta
Set Value	Control Scaled	Variable #	-
Set Value	Control Offset	Variable #	-
Refresh	Set	Variable #	-
Single Dimmer	Set	Variable #	Delta
Curve	Control	Variable #	Curve
Curve	Inverted Control	Variable #	Curve

Variables are further explained on page 37.

The Single Dimmer feature is used to increase or decrease a level by using only one switch. When controlling this task through a GPI action , then closing the GPI will increase or decrease the level. Opening the GPI port will freeze on the current level. This feature is useful for controlling an intensity will just one button.

B.8 System

Miscellaneous tasks.

Feature	Function	Parameter 1	Parameter 2
Blink	Set	On or Off	-
Blink	Toggle	-	-
Blink	Control	-	-

The Blink feature controls the LED on the unit as seen in figure 4.2.

B.9 Action

Use the Link feature to have one action trigger another action.

Feature	Function	Parameter 1	Parameter 2
Link	Set	Action	-

B.10 Actionlist

Manipulate an actionlist.

Feature	Function	Parameter 1	Parameter 2
Enable	Set	Actionlist	On or Off
Enable	Toggle	Actionlist	-
Enable	Control	Actionlist	-
Enable	Inverted Control	Actionlist	-

B.11 Randomizer

Trigger the Randomizer to generate a new random number.

Feature	Function	Parameter 1	Parameter 2
Refresh	Set	Minimum value	Maximum value

The Randomizer functionality is discussed on page 40.

B.12   Timer

Manipulate on of the four internal timers.

Feature	Function	Parameter 1	Parameter 2
Playstate	Start	Timer #	-
Playstate	Stop	Timer #	-
Playstate	Restart	Timer #	-
Time	Set	Timer #	Time

Appendix C

Templates

This appendix discusses the templates provided in the Show Control page.

Template	Description
DMX ->DALI Channels	DMX channels 1 to 32 will be converted to DALI fixture levels.
DMX ->DALI Groups	DMX channels 1 to 16 will be converted to DALI groups levels.
DMX ->DALI Broadcast	DMX channel 1 will be converted to the DALI broadcast level.
DALI ->DMX	DAlI fixture intensities 1 to 32 will be converted to DMX channels.
GPI ->Scene	Each GPI port will trigger scene 1, 2,, 3, or 4. Set the GPI ports to digital on the Settings page.
Art-Net ->DMX	Converts incoming Art-Net to DMX. Specify the Art-Net subnet and universe on the Settings page.

Appendix D

API

The DaliCore is pre-programmed to make its internal functionality available via OSC, TCP, UDP and HTTP. There is a simple API implemented for each protocol. Notwithstanding these API's, it is possible to create your own OSC, TCP and UDP implementation in the Show Control page.

The API is originally designed for external equipment to control the DaliCore, however, it is also capable to send information back. This feedback mechanism is discussed at the end of this chapter, on page 86.

D.1   OSC

The scenes can be called by using the following command. The number '1' can be replaced by any number in the range of [1,16].

URI	Parameter	Description
/core/scene/1	-	Activate scene #1

The following table uses actionlist #1 as an example. The number '1' can be replaced by any number in the range of [1,8]. The table also uses action #2 as an example. The number '1' can be replaced by any number in the range of [1,48].

URI	Parameter	Description
/core/al/1/2/execute	bool/float/integer	Execute action #2 inside actionlist #1
/core/al/1/enable	bool	Set the 'enable' checkbox for actionlist #1

The following table uses actionlist #1 as an example. The number '1' can be replaced by any number in the range of [1,8]. The table also uses action #2 as an example. The number '1' can be replaced by any number in the range of [1,48].

URI	Parameter	Description
/core/al/1/2/execute	bool/float/integer	Execute action #2 inside actionlist #1
/core/al/1/enable	bool	Set the 'enable' checkbox for actionlist #1

The following table uses timer #1 as an example. The number '1' can be replaced by any number in the range of [1,4].

URI	Parameter	Description
/core/tm/1/start	-	Start timer #1
/core/tm/1/stop	-	Stop timer #1
/core/tm/1/restart	-	Restart timer #1
/core/tm/1/pause	-	Pause timer #1
/core/tm/1/set	time-string	Set timer #1 at the time-string

The following table uses variable #1 as an example. The number '1' can be replaced by any number in the range of [1,8].

URI	Parameter	Description
/core/va/1/set	integer	Set the value of variable #1
/core/va/1/refresh	-	Refresh variable #1; a trigger will be generated as if the variable changed value
/core/va/refresh	-	Refresh all variables; triggers will be generated

The following table shows how to active miscellaneous functions.

URI	Parameter	Description
/core/dmx/1	integer	Set the value of a DMX channel
/core/blink	-	Momentarily flashes the DaliCore's LED
/core/hello	-	The unit will reply with the same Hello message

D.2   TCP & UDP

TCP (Transmission Control Protocol) is a protocol for sending messages across an Ethernet network. TCP provides reliable, ordered and error-checked delivery of messages between programs running on computers connected to a local area network, intranet or the public Internet.

UDP (User Datagram Protocol) is a simple protocol for sending message across the network. It does not provide any error checking. Although UDP is a bit faster than TCP, it is less secure.

Typically either TCP or UDP is supported by various media devices like video projectors and show controllers.

The functionality within the DaliCore can be controlled by using the following ASCII strings (human readable text) messages:

Scenes can be called by using the following command. The number '1' can be replaced by any number in the range of [1,16].

String	Description
core-scene-1	Activate scene #1

The following table uses actionlist #1 as an example. The number '1' can be replaced by any number in the range of [1,8]. The table also uses action #2 as an example. The number '1' can be replaced by any number in the range of [1,48].

String	Description
core-al-1-2-execute=\<arg\>	Execute action #2 inside actionlist #1
core-al-1-enable=\<bool\>	Set the 'enable' checkbox for actionlist #1

The following table uses timer #1 as an example. The number '1' can be replaced by any number in the range of [1,4].

String	Description
core-tm-1-start	Start timer #1
core-tm-1-stop	Stop timer #1
core-tm-1-restart	Restart timer #1
core-tm-1-pause	Pause timer #1
core-tm-1-set=\<text\>	Set timer #1 at the time-string

The following table uses variable #1 as an example. The number '1' can be replaced by any number in the range of [1,8].

String	Description
core-va-1-set=<integer>	Set the value of variable #1
core-va-1-refresh	Refresh variable #1; a trigger will be generated as if the variable changed value
core-va-refresh	Refresh all variables; triggers will be generated

The following table shows how to activate miscellaneous functions.

String	Description
core-dmx-1=<integer>	Set the value of a DMX channel
core-blink	Momentarily flashes the DaliCore's LED
core-hello	The unit will reply with the same Hello message

D.3 HTTP

HTTP (Hyper Text Transfer Protocol) is the standard protocol to access web pages. It can also be used to control the DaliCore, using the URLs listed below.

Scenes can be called by using the following command. The number '1' can be replaced by any number in the range of [1,16].

Description	URL	Parameter Range	Example
Activate scene	/ajax/scene/1	-	http://192.168.1.10/ajax/scene/1

The following table uses actionlist #1 as an example. The number '01' can be replaced by any number in the range of [01,08].

Description	URL	Parameter Range	Example
Execute action	/ajax/alXX/2/exe	-	http://192.168.1.10/ajax/al01/2/exe=true
Enable actionlist	/ajax/alXX/ena	true/false	http://192.168.1.10/ajax/al01/enable=false

Description	URL	Parameter Range	Example
Start timer #1	/ajax/tm/1/start	-	http://192.168.1.10/ajax/tm/1/start
Stop timer #1	/ajax/tm/1/stop	-	http://192.168.1.10/ajax/tm/1/stop
Restart timer #1	/ajax/tm/1/restart	-	http://192.168.1.10/ajax/tm/1/restart
Pause timer #1	/ajax/tm/1/pause	-	http://192.168.1.10/ajax/tm/1/pause
Set time	/ajax/tm/1/set	<hh:mm:ss>	http://192.168.1.10/ajax/tm/1/set=<01:02:03>

Description	URL	Parameter Range	Example
Blink LED	/ajax/bli	-	http://192.168.1.10/ajax/blink

You can send your HTTP GET requests to port 80.

D.4 Feedback

The DaliCore is able to send feedback to external equipment using its API, so called 'clients'. The DaliCore keeps a memory of the last four OSC clients and last four UDP clients. The clients will automatically receive updates on several playback related state changes.

Below is a table listing the messages the DaliCore will send back to its clients.

OSC	UDP
/core/scene	core-scene
/core/al/1/enable	core-al-1-enable
/core/hello	core-hello

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

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

Consider programming custom action in the show control when additional feedback functionality is required.

D.4.1 Preventing a feedback loop

Feedback is automatically send to a device which uses the OSC or UDP 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 53.

Index