

B-STATION2 MANUAL



Revision History

| Revision | Date | Author(s) | Description |
|----------|------------|-----------|--|
| 1 | 01.10.2020 | BZ, ME | Initial version |
| 2 | 22.10.2020 | ME | Improved images |
| 3 | 19.02.2021 | ME | Added Timer and Userlist examples. Corrections in API. Updated screen shots. |

Contents

| | | |
|-----------|----------------------------|-----------|
| 1 | Introduction | 7 |
| 2 | Protocols | 12 |
| 3 | Quickstart | 16 |
| 4 | Setting up | 21 |
| 5 | Network | 23 |
| 6 | Patch | 27 |
| 7 | Playbacks | 30 |
| 8 | Show Control | 37 |
| 9 | Protocol Conversion | 48 |
| 10 | Monitors | 53 |
| 11 | Settings | 56 |
| 12 | vManager | 63 |
| 13 | Kiosc | 67 |
| | Appendices | 69 |
| A | Trigger Types | 70 |
| B | Task Types | 77 |
| C | Templates | 85 |
| D | API | 86 |

©2021 Visual Productions BV. All rights reserved.

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

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

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

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



Declaration of Conformity

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

BStation2

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



ЕВРАЗИЙСКИЙ ЭКОНОМИЧЕСКИЙ СОЮЗ

ДЕКЛАРАЦИЯ О СООТВЕТСТВИИ

Заявитель Общество с дополнительной ответственностью "Арт Рамос Студио", зарегистрировано в Едином государственном регистре юридических лиц и индивидуальных предпринимателей за № 101125337, место нахождения (адрес юридического лица): Республика Беларусь, Минская обл., 220116, г. Минск, пр-т газеты "Правда", д. 29, пом. 8, номер телефона: +375 17 2442299, адрес электронной почты: sales@ars-by.com в лице заместителя директора Стрельцова Антона Валерьевича (Доверенность № 1 от 03.01.2017)

заявляет, что световые контроллеры «Visual Productions» модели CueCore2, QuadCore, IOCore2, TimeCore, LPU-1, LPU-2, B-Station2

изготовитель: Visual Productions BV (Королевство Нидерландов),

место нахождения (адрес юридического лица): Izaak Enschedeweg 38A NL-2031 CR Haarlem (Королевство Нидерландов)

изготавливается по технической документации изготовителя

код ТН ВЭД ЕАЭС: 8537 10 910 9

серийный выпуск

соответствует требованиям технического регламента Таможенного союза ТР ТС 020/2011 «Электромагнитная совместимость технических средств».

Декларация о соответствии принята на основании протокола испытаний № 102-19-1927 от 30.09.2019, выданного испытательной лабораторией открытого акционерного общества "Испытания и сертификация бытовой и промышленной продукции "БЕЛЛИС", аттестат аккредитации №ВУ/112 1.001.;

Схема декларирования соответствия – Id.

Дополнительная информация: Условия хранения: температура хранения от + 5°C до + 40°C; влажность 60 % при + 20 °C при отсутствии в воздухе кислотных, щелочных и иных агрессивных примесей с соблюдением правил пожарной безопасности. Срок службы 5 лет. Только для профессионального использования.

Примененные стандарты:

ГОСТ EN 55103-1-2013 Совместимость технических средств электромагнитная. Электромагнитные помехи от профессиональной аудио-, видео-, аудиовизуальной аппаратуры и аппаратуры управления световыми приборами для зрелищных мероприятий. Нормы и методы измерений

ГОСТ 30805.22-2013 (CISPR 22:2006) Совместимость технических средств электромагнитная.

Оборудование информационных технологий. Радиопомехи индустриальные. Нормы и методы измерений

ГОСТ 30805.14.1-2013 (CISPR 14-1:2005) Совместимость технических средств электромагнитная. Бытовые приборы, электрические инструменты и аналогичные устройства. Радиопомехи индустриальные. Нормы и методы измерений

ГОСТ 30804.3.2-2013 (IEC 61000-3-2:2009) Совместимость технических средств электромагнитная. Эмиссия гармонических составляющих тока техническими средствами с потребляемым током не более 16 А (в одной фазе). Нормы и методы испытаний

ГОСТ 30804.3.3-2013 (IEC 61000-3-3:2008) Совместимость технических средств электромагнитная. Ограничение изменений напряжения, колебаний напряжения и фликера в низковольтных системах электроснабжения общего назначения. Технические средства с потребляемым током не более 16 А (в одной фазе), подключаемые к электрической сети при несоблюдении определенных условий подключения. Нормы и методы испытаний

Общество с дополнительной ответственностью "Арт Рамос Студио" является уполномоченным лицом Visual Productions BV (Королевство Нидерландов) на основании договора от 01.10.2019 № б/н.

Декларация о соответствии действительна с даты регистрации по 29.10.2024 включительно.


(подпись) 

Стрельцов Антон Валерьевич

(Ф. И. О. заявителя)

Регистрационный номер декларации о соответствии:

ЕАЭС № ВУ/112 11.01. ТР020 000 07206

Дата регистрации декларации о соответствии: 31.10.2019

Chapter 1

Introduction

Thank you for choosing the B-Station2. The B-Station2 is a solid state controller designed for two purposes.

Firstly, for remote-controlling a lighting system like CueCore2, LPU-2 or DaliCore. The buttons can be freely assigned to any function in the lighting controller, e.g. select a cue or start/stop a show. Because the B-Station2 uses communication protocols that are open (in the public domain) such as UDP and OSC, it can also be used for controlling third-party equipment like show-controllers, lighting consoles, media servers or audio devices.

Furthermore, the B-Station2 is designed as a stand-alone lighting control system for small lighting setups like LED or architectural projects. It can control up to 32 individual DMX fixtures and supports 6 playbacks with each 64 cues. It even features a dynamic FX-generator.

An internal web-server provides the web-interface through which you can program the B-Station2. A modern browser is required to access this web-interface during set-up. A browser or computer is not required for standalone use after the initial set-up.

At the time of writing this manual the B-Station2 firmware was at version 1.05.

This manual discusses setting up and programming the unit. Chapter 2 provides background information on the communication protocols used the B-Station2. Chapters 4 and 5 cover how to set up the unit and configure the network connection.

When the B-Station2 is just used for controlling external equipment then the chapters 6 and 7 can be ignored. Setting up the remote control functionality is done in the 8 chapter discussed on page 37.

Setting up the B-Station2 for stand-alone lighting control will involve the following steps:

- Patching fixtures (discussed in chapter 6)
- Creating cues (discussed in chapter 7)



Figure 1.1: B-Station2

- Assigning buttons to trigger the cues (discussed in chapter 8)

1.1 Features

The feature set of the B-Station2 includes:

- 6 user-assignable push-buttons
- RGB coloured LEDs
- Ethernet based, supporting protocols like UDP, OSC, Art-Net and sACN
- DMX-512
- RDM
- Web-based user-interface for programming
- Standalone control for 6 zones, 32 fixtures and 6x64 cues
- Standalone FX-generator
- PoE (Power Over Ethernet) Class I
- Compatible with CueCore2, QuadCore and IoCore2
- Bundled with vManager software
- Compatible with Kiosc software

1.2 Comparison

The following table visualises the difference between the B-Station1 and the B-Station2.

| | B-Station2 | B-Station1 |
|---------|--------------|--------------|
| Buttons | 6 | 6 |
| DMX | 1 universe | 1 universe |
| RDM | yes | - |
| LED | RGB | White |
| Art-Net | input+output | input+output |
| sACN | input+output | - |
| TCP | input | - |
| UDP | input+output | input+output |
| OSC | input+output | input+output |
| PoE | class I | class I |
| DHCP | yes | - |

1.3 Limitations

The B-Station2 is a powerful device with many possibilities, however, there are some limitations as shown in the following table.

| | |
|---------------------|-----|
| Fixtures | 32 |
| Playbacks | 6 |
| Cues per Playback | 64 |
| Action lists | 8 |
| Actions per list | 48 |
| Actions system-wide | 64 |
| Tasks per Action | 8 |
| Tasks system-wide | 128 |
| Variables | 10 |
| Timers | 4 |

1.4 What's in the box?

The B-Station2 packaging contains the following items (see figure 1.2):

- B-Station2
- Info card
- UTP network cable
- 3 sheets with adhesive icons



Figure 1.2: B-Station2 box contents

1.5 Saving data to memory

This manual will describe how to configure the B-Station2 and program playbacks, 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 B-Station2 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 B-Station2 will copy any changes in the RAM memory to its onboard flash memory. Flash memory retains its data even when not powered. The B-Station2 will load all its data back from the flash memory upon startup.

This memory copy process is conducted automatically by the B-Station2 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.

1.6 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 B-Station2 supports various protocols. This chapter describes these protocols and to which extent they are implemented in the B-Station2.

2.1 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 2.1: Visual Productions' RdmSplitter

The DIN Rail RdmSplitter from Visual Productions (See figure 2.1) 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.

The B-Station2 has one DMX port and is therefore able to control 512 channels. The port can be configured either as input or output. Figure 2.2 shows the pinout of the connector.

Figure 2.2: DMX Pinout

2.2 RDM

The Remote Device Management (RDM) protocol - officially called ANSI E1.20 - is built on top of the DMX-512 protocol. RDM enhanced 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 B-Station2 can discover RDM fixtures, set starting addresses and choose modes.

2.3 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 B-Station2 supports sending and receiving of one universe of Art-Net data.

2.4 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 B-Station2 supports sending and receiving of one sACN universe.

2.5 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 retransmission of lost packets, thereby making sure that all data transmitted is received.

The B-Station2 supports reception of TCP message.

2.6 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, therefore it is faster than TCP but less reliable.

There are two ways how to have the B-Station2 respond to incoming UDP messages. The API (see page 88) makes typical B-Station2 functions available through UDP. Furthermore, custom messages can be programmed in the Show Control page (see page 37). This is also the place where to program outgoing UDP messages.

2.7 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. Kiosk from Visual Productions.

There are two ways how to have the B-Station2 respond to incoming OSC messages. Firstly, the API (see page 86) makes typical B-Station2 functions available through OSC. Secondly, custom messages can be programmed in the Show Control page (see page 37).

2.8 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 B-Station2 is a DHCP client.

Chapter 3

Quickstart

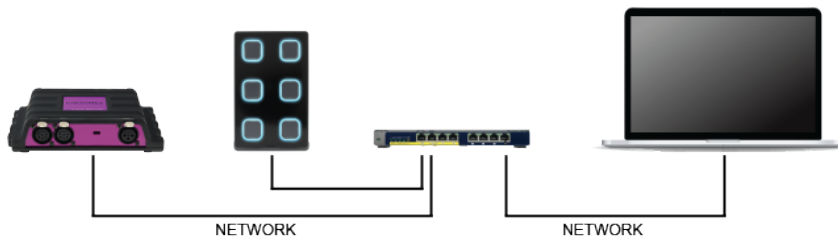
This chapter provides step by step tutorials on how to program your B-Station2 for some typical tasks:

- Remote control the playbacks in a CueCore2
- Stand-alone control of a RGB fixture

3.1 Remote control the playbacks in CueCore2

This tutorial shows how to use the B-Station2 to remote the CueCore2. Follow the next few steps in order to link the B-Station's buttons to the CueCore2's playbacks.

1. Setup power and network
Connect the B-Station2, CueCore2 and the computer to a PoE capable network switch.



2. Connect to the network

Configure the equipment to have the following static IP addresses:

| | IP | Subnet |
|------------|--------------|-----------------------------------|
| Computer | 192.168.1.9 | 255.255.255.0 |
| B-Station2 | 192.168.1.10 | 255.255.255.0 (Default static IP) |
| CueCore 2 | 192.168.1.11 | 255.255.255.0 |

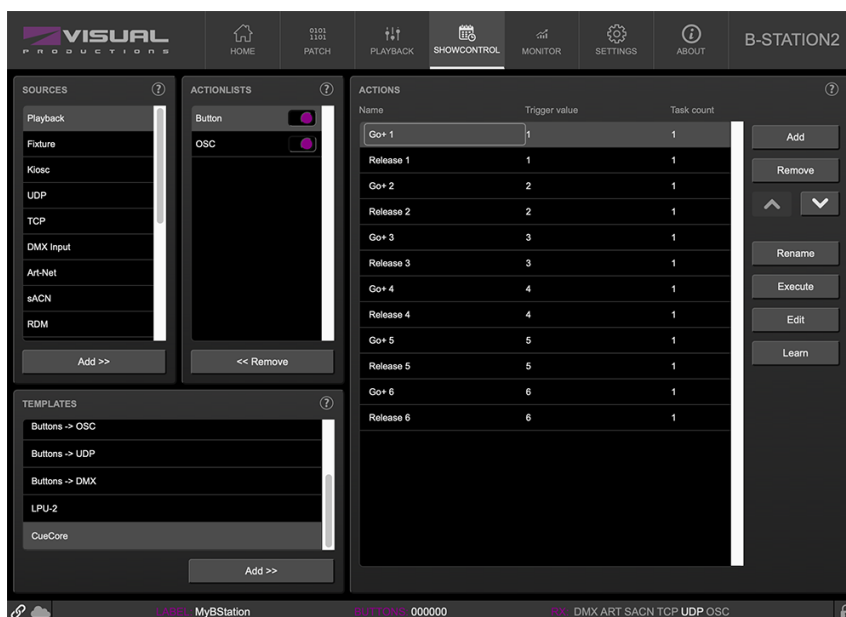
The B-Station2 can be set to static IP 192.168.1.10 by long-pressing the reset button on the device's electronic board.

3. Prepare the CueCore2

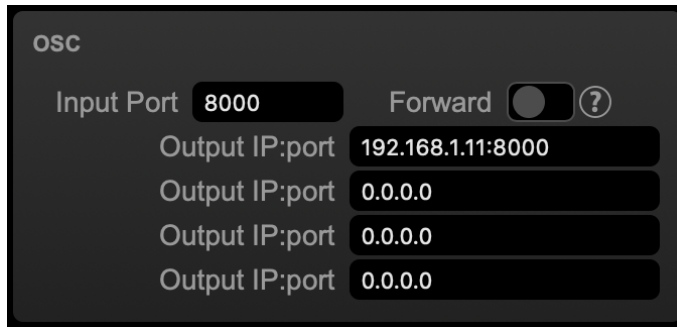
Make sure the CueCore2 has six cues inside playback 1. Each cues should refer to a track with a certain lighting scene. Also, its OSC *In port* should be set to 8000. Please refer to the CueCore2's manual for more information.

4. Add template

Open a browser on the computer and browse to the web-interface of the B-Station2 by typing '192.168.1.10' in the browser's address field. Go to the *Show Control* page and select the 'CueCore' template. Press the 'Add >>' button in the lower left corner of the screen.



5. Set OSC IP address
Go to the B-Station2 *Settings* page and fill in '192.168.1.11:8000' in the *Out IP 1* field in the OSC settings.

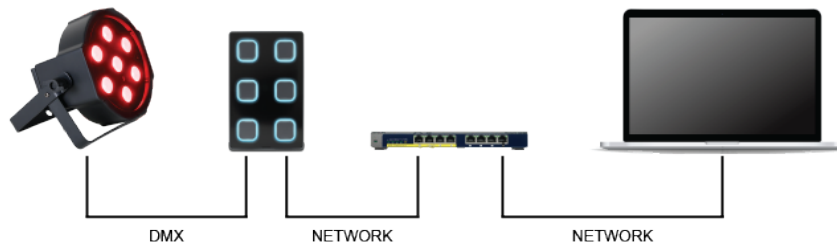


Pressing the buttons on the B-Station2 should now activate cues within the CueCore2.

3.2 Stand-alone control of a RGB fixture

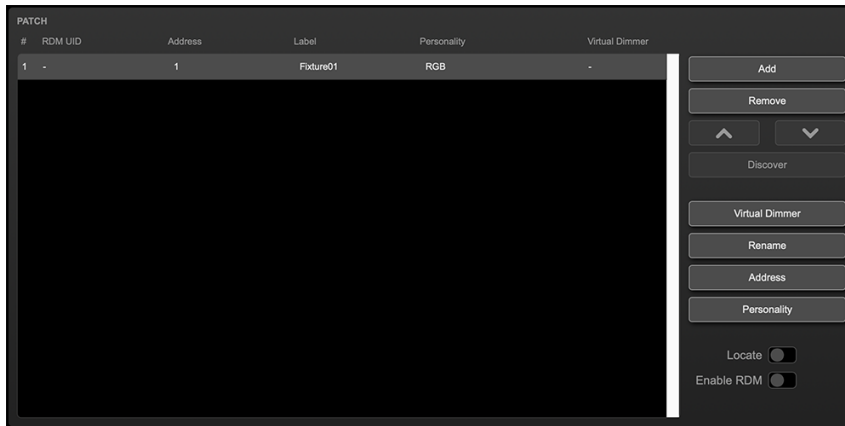
This example will patch one RGB fixture, create a playback with three cues and assign buttons to those cues. Please take the following steps:

1. Setup power and network
Connect the B-Station2 and the computer to a PoE capable network switch. Connect the fixture to the B-Station2's DMX connector.



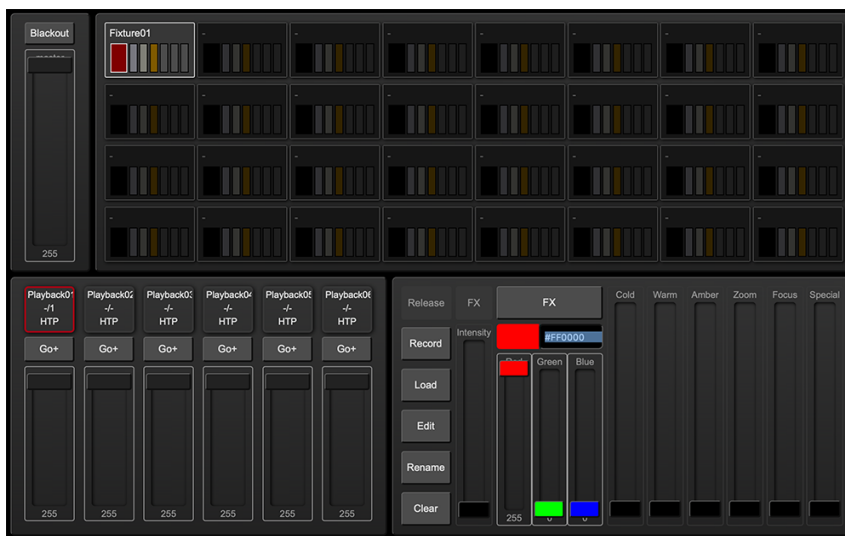
2. Patch

Use the browser to go to the B-Station2's *Patch* page. Select the first fixture slot and assign a RGB personality from the table on the right.



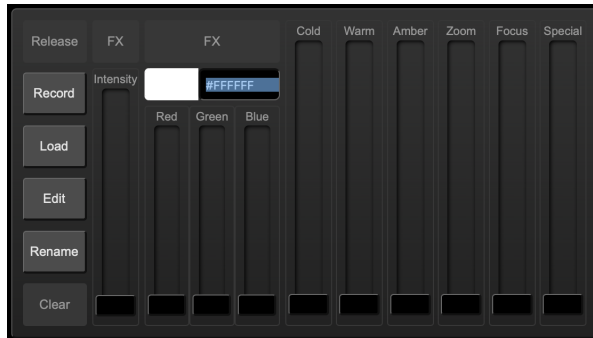
3. Program the playback

Go to the *Playback* page and select the first fixture. Bring up the Red slider in the *Programmer* section. Press the *Record* button, then click on Playback1's *Go+* button. You have now stored the first cue. Repeat the steps to record two more cues in Playback1.



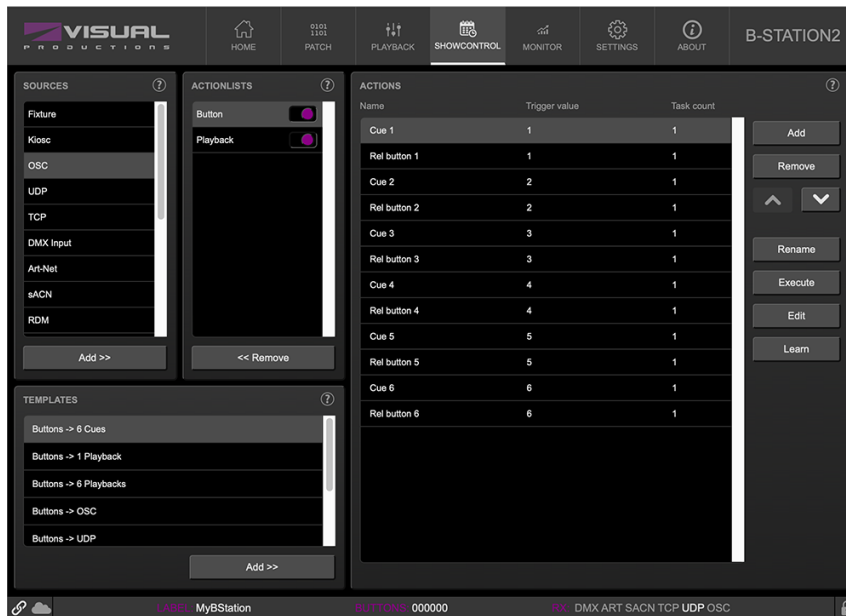
4. Clear the Programmer

Clear the *Programmer* by pressing the *Clear* button twice.



5. Assign buttons

Go to the *Show control* page. Add the 'Buttons ->6 Cues' template.



Pressing the buttons 1 till 3 will now activate cues 1 till 3. The button's LEDs will indicate which cue is active.

Chapter 4

Setting up

This chapter discusses how to set up the B-Station2.

4.1 Mounting

The B-Station2 is wall-mount, it is compatible with European and American gang boxes. If no gang box is available then the B-Station2 can also be surface mounted. The inside of the B-Station2's enclosure suggests where you can create a hole for the cable to enter.

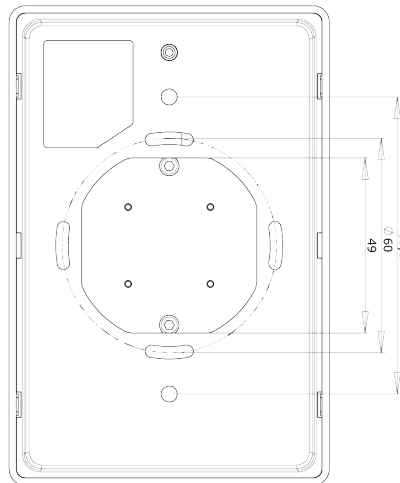


Figure 4.1: Mounting dimensions

4.2 Power

The B-Station2 is Power-over-Ethernet (PoE) powered, it requires PoE Class I.

4.3 Labels

The B-Station2 is shipped with various adhesive labels that you can apply to the buttons. If you wish to create your own labels then please use the following dimensions.

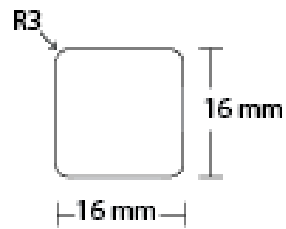


Figure 4.2: Label dimensions

Chapter 5

Network

The B-Station2 is a network capable device. A network connection between a computer and the unit is required to configure and program the B-Station2, however, once the device is programmed then it is not necessary anymore for the B-Station2 to be connected to an Ethernet network. A PoE switch or a PoE injector can simply be used to power the B-Station2 without any further need of network connection.

There are multiple arrangements possible for connecting the computer and the B-Station2. They can be connected peer-to-peer using a PoE injector, via a PoE network switch or using a combination of these and a Wi-Fi access point. Figure 5.1 illustrates these different arrangements.

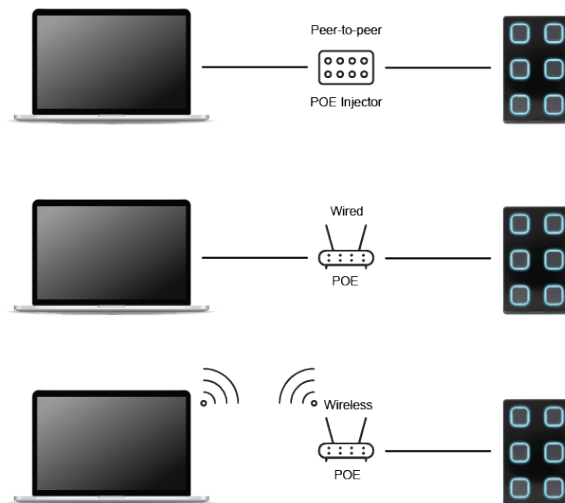


Figure 5.1: Network arrangements

The Ethernet port on the B-Station2 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.

5.1 IP Address

The B-Station2 supports both static IP addresses and automatic IP addresses. By default, the B-Station2 is set to 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 B-Station2 and a computer. It is also useful in permanent installations where the IP address of the B-Station2 is known by other equipment and therefore 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.

There are three ways to change the IP address setting of the B-Station2.

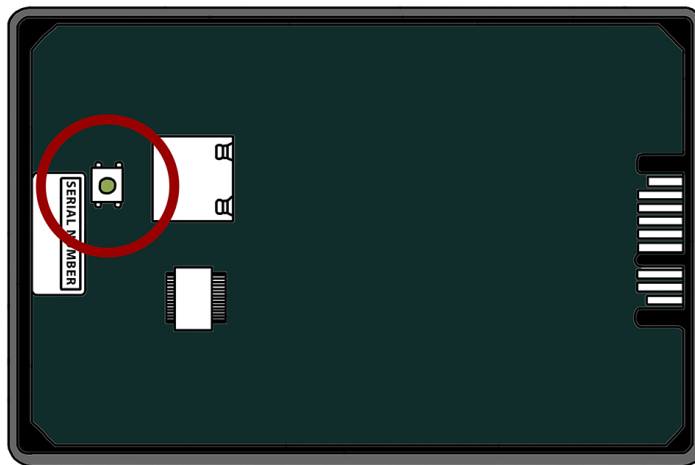


Figure 5.2: Reset button

- **vManager** can be used to detect a B-Station2 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 B-Station2'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 back of the device it toggles between static and automatic IP addresses. When switching modes, the B-Station2's leds will briefly flash in red for DHCP, or in white for static. By pressing and holding the reset button (see figure 5.2) 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.

5.2 Web-interface

The B-Station2 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 (v13 or higher)
- Mozilla Firefox (v54 or higher)

The web-interface enables you to configure and program the B-Station2. When browsing to the unit the home page (figure 5.3) 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.

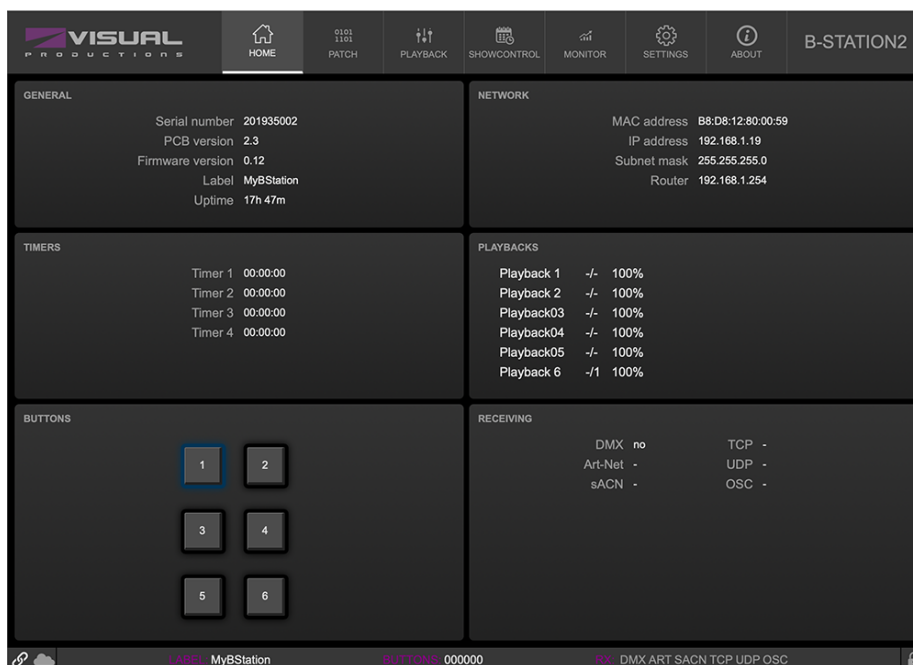


Figure 5.3: Home page

5.2.1 Uptime

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

5.3 Access via Internet

The B-Station2 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 referred to as NAT or Port-Redirecting). Please note that port forwarding is not secure, since anybody could access the B-Station2 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 B-Station2. 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 6

Patch

The *Patch* page is the starting point when using the B-Station2 as a stand-alone DMX controller. This page is used for identifying which fixtures will be controlled. Up to 32 fixtures can be entered.

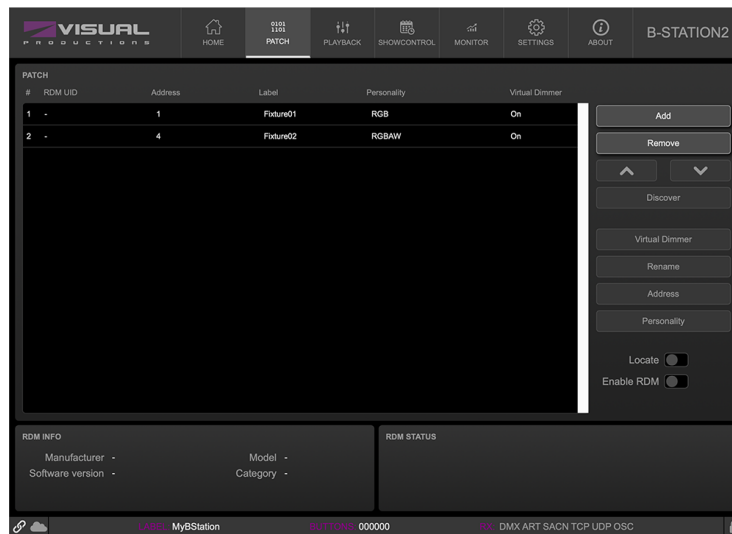


Figure 6.1: Patch Page

6.1 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.

6.2 Address

DMX starting addresses range between 1 and 512. the B-Station2 automatically 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 had multiple fixtures selected, the B-Station2 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.

6.3 Virtual Dimmer

Personalities with RGB but without Intensity will automatically be given a virtual dimmer. This means that the B-Station2 will act as if the fixture has an intensity channel and allows for RGB and intensity to be programmed separately, however, it will modulate the intensity on the RGB values.

6.4 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.

6.5 Locate

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

Chapter 7

Playbacks

When using the B-Station2 as a stand-alone lighting controller then the lighting scenes and effects are programmed in the *Playback* page. It is assumed that a patch has already been created on the *Patch* page prior to programming playbacks.



Figure 7.1: Playback page

There are six playbacks. Each playback can have up to 64 cues. A cue holds a lighting scene; this can be a static level for a fixture's parameter (such as Intensity or Warm White) or it can be a dynamic effect from the FX Engine.

7.1 Cues

A cue is flexible on how many fixtures it targets; ranging from a single fixture to all. Likewise, it can control a single parameter or all of them.

7.2 Programmer

When recording cues, first create a lighting scene which will be then stored in the cue. The Programmer (figure 7.2) is the place where the scene is created. First select a fixture, the programming window will update to show the capabilities of those fixtures and enables you to set the parameters, e.g. set a dimmer level or set a colour. You can now also deselect the fixture and select new ones; the values will be kept in the Programmer's memory.

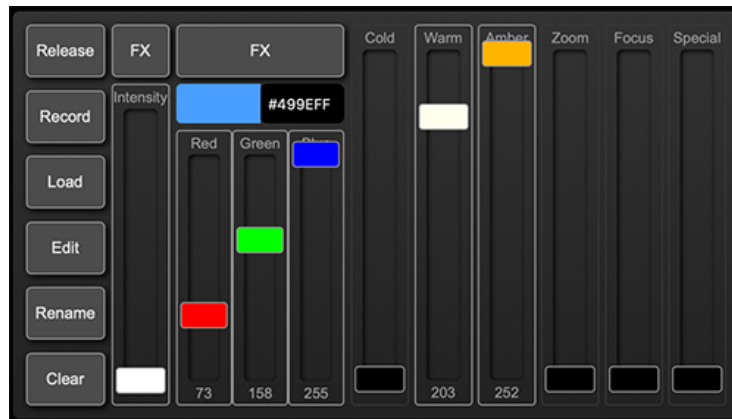


Figure 7.2: Programmer

Values that reside in the programming windows' memory have priority. For example you might have a cuelist running that sets a certain dimmer to 50% but if the programming window contains a dimmer level of 10% for that particular fixture, then this dimmer will be set 10%. Therefore it is important to clear the programmer once you're finished recording cues.

The programmer has a few associated buttons:

7.2.1 Clear

Press the *Clear* button once for clearing all values for the currently selected fixtures and press it twice for clearing all values, even for fixtures currently not selected.

7.2.2 Release

To stop a running playback you need to 'release' it. There are a few ways how to do this:

- Long-press the playback's 'Go+' button.
- Enable the *Release* button and then press a playback's 'Go+' button.
- Long-press the *Release* button; this will stop all playbacks.
- Program an action to release a playback
- Send a release command via the API

7.2.3 Record

To copy the programmer's content into a cue, press the *Record* button and then click on one of the playback's Go buttons.

If the playback was released then automatically a new cue will be added to this playback. The new cue will contain the programmer's information.

If the playback was currently playing a cue then the *Record* button will trigger a pop-up dialog (figure 7.3) offering options on how to copy the programmer's information into the playback.

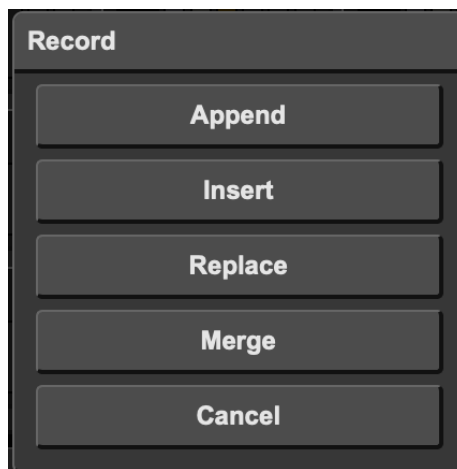


Figure 7.3: Record dialog

The record dialog offers the following choices:

| | |
|---------|--|
| Append | A new cue will be added at the end of the playback. |
| Insert | A new cue will be inserted at the current position. |
| Replace | The selected cue will have its value overwritten. |
| Merge | The selected cue will have its value merged with the values from the programmer. |

7.2.4 Load

It is also possible to transfer the content from a cue back to the programmer. This is useful for altering an existing cue. Make sure a playback is playing the desired cue. Then press the *Load* button and then press the playback's *Go+* button. In case the playback has multiple cues then a pop-up dialog (figure 7.4) will allow you to choose which cue to load.

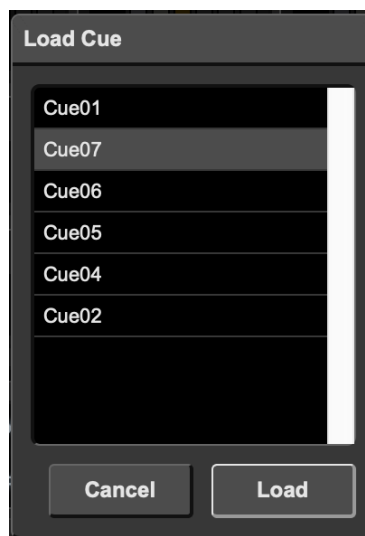


Figure 7.4: Load dialog

The programmer will now have a copy of the cue's information. Also the fixtures present in that cue will be automatically selected. After making your changes you can write back the scene by using the Record button.

7.2.5 Edit

The content of the playback can be exposed by enabling the *Edit* button and the clicking on a playback's Go button. This will open the Playback Editor which is discussed on page 34.

7.2.6 Rename

To rename a playback or fixture, first enable the Rename button and then click on a playback or fixture.

7.3 FX Engine

Your playback consists of cues and cues are static lighting 'looks'. To animate your lighting you usually create multiple cues and have the playback chase between them. However, you can use FX to quickly create a dynamic animation in your lighting. For example a smooth rainbow colour chase on your RGB fixture would require you to program many steps. With FX you simply used the FX dialog (see figure 7.5) to select the *Rainbow* effect for your colour attribute, set a few parameter levels and it's done.

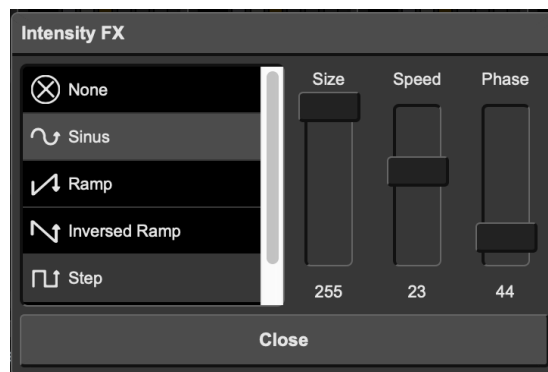


Figure 7.5: FX dialog

There are also FX for RGB colour mixing e.g. a rainbow effect. FX also exists for the intensity attribute. You can use intensity effects for quickly creating some chases by using the *Phase* parameter.

7.4 Playback Editor

The Playback Editor dialog allows you to change the properties of the playback and its cues.

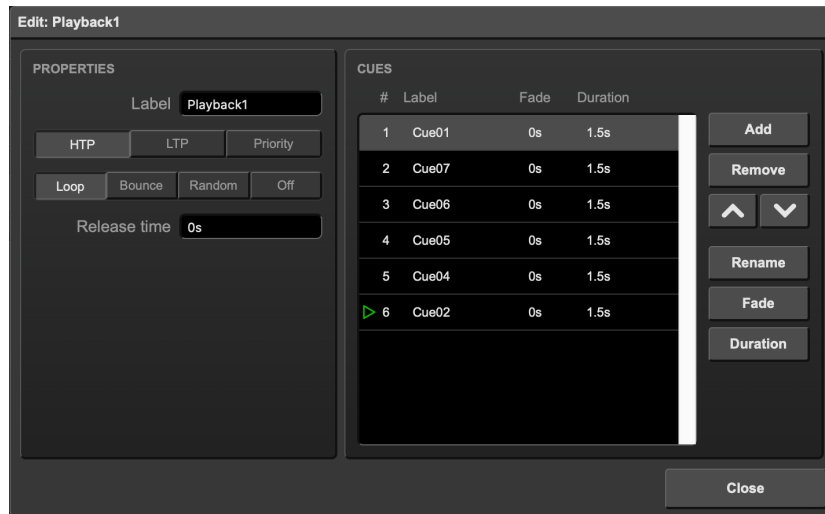


Figure 7.6: Playback Editor

7.4.1 Precedence

All active Playbacks produce DMX values. These values will be merged together and sent to the DMX output. The precedence setting determines how this merging is done. Each playback can be set to either HTP (Highest Takes Precedence), LTP (Latest Takes Precedence) or Priority.

HTP is the most common choice in precedence. With HTP the output of all playbacks is compared to each other; for each DMX channel the level is set to the highest value found in that particular channel amongst all playbacks. The table below shows an example of HTP merging.

| | Playback 1 | Playback 2 | Playback 3 | Merged Output |
|-----------|------------|------------|------------|---------------|
| Channel 1 | 0% | 0% | 25% | 25% |
| Channel 2 | 100% | 0% | 25% | 100% |
| Channel 3 | 0% | 0% | 0% | 0% |
| Channel 4 | 0% | 100% | 25% | 100% |

In the LTP approach only one playback is active amongst all LTP playbacks. The output of that active playback is included in the merge with all HTP playbacks. All other LTP playbacks are ignored. Which LTP playback is active is determined by which playback is started latest, or which received a Go+ command latest. Please consider figure 7.7.

If there is a playback active with its precedence set to Priority then all other playbacks are ignored. When there are multiple Priority playback then those will be merged together according to the HTP principle.

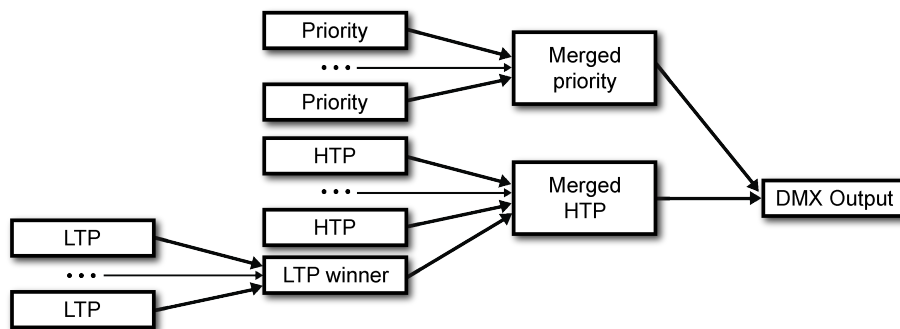


Figure 7.7: Precedence

7.4.2 Release Time

The release time defines the behaviour when the playback releases. If the time is set to 0s then the playback will shut off immediately. Otherwise the values will gradually fade out over the duration set by the release time field.

7.4.3 Repeat

This property determines what the playback does when it finishes the last cue.

| | |
|--------|---|
| Loop | Will start over from the beginning. |
| Bounce | Will make it traverse back to the beginning, and it will keep going back and forth. |
| Random | The order of the cues will be random. |
| Off | The Playback will automatically release when reaching the end of the cues. |

7.4.4 Fade

The cue will fade from the current levels to its programmed levels. The time it takes to cross-fade is specified by *Fade*. When the fade is set to 0 then there will be no cross-fade; the values will change instantly.

7.4.5 Duration

Determines how long the cue will be active before traversing to the next cue. This is the time between the completion of the cross-fade into this cue and the start of the cross-fade to the next cue.

The second option for the duration field is to input *halt*. In this case the cue will continue to run indefinitely; it requires a Go+, Go- or Jump command to traverse to the next cue.

Chapter 8

Show Control

The B-Station2 can interact with the outside world; the buttons on the device can be pressed or it can send and receive messages and values through various protocols. It is possible to automate the B-Station2 by having it respond automatically to incoming signals. An example of this would be to start a playback upon pressing a button on the B-Station2. Another option is integrating the B-Station2 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.

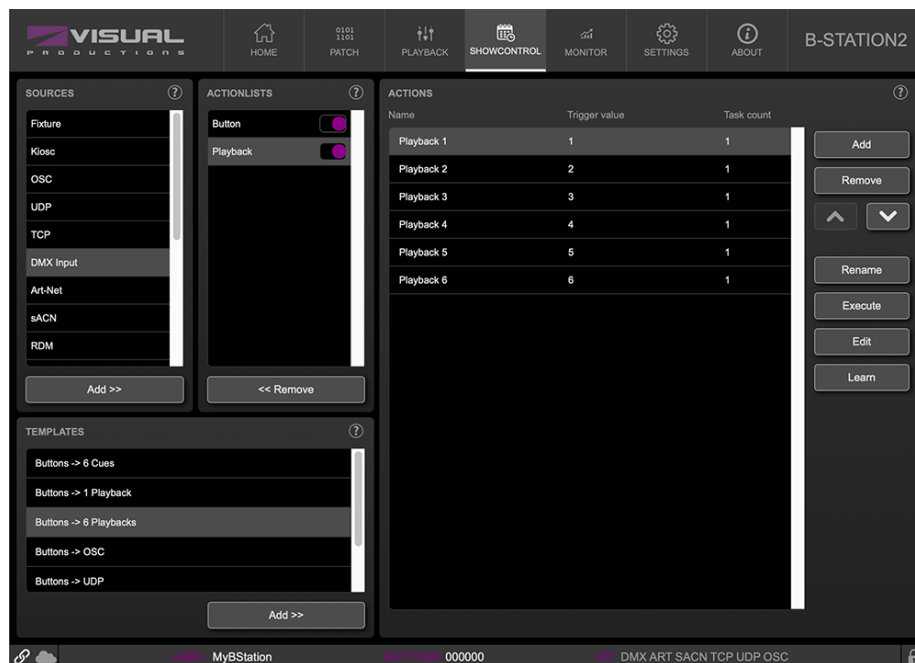


Figure 8.1: Show Control page

The *Show Control* page presents a system of 'actions'. A signal that the B-

Station2 needs 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.

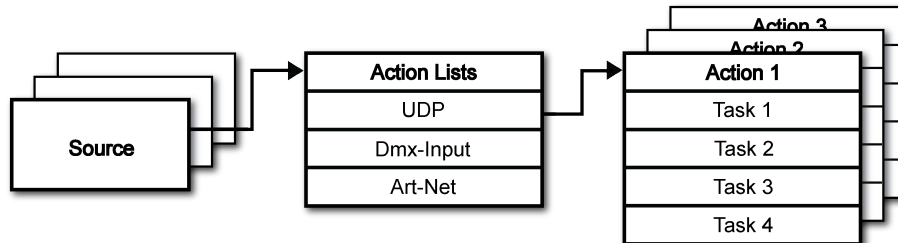


Figure 8.2: Show Control structure

The B-Station2 is capable of listening to various protocols. These available protocols are listed in *Sources*, however, the B-Station2 can only actively listen to 8 protocols at once. The active protocols are listed in *Actionlists*. Each action list 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 action list 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 B-Station2 will do when this action is triggered. Several tasks can be placed in the action. There are tasks available for a wide range of B-Station2 features and external protocols. Task types are detailed in Appendix B on page 77.

Please consult the API appendix on page 86 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 Action Lists

The *Sources* listing presents all protocols that the B-Station2 is capable of receiving. It also includes internal features that can create events that can be used for triggering actions, such as the B-Station2's buttons. These sources are available, however, they will only be actively listened to once added to the action-list table.

| | |
|---------------|---|
| Button | Triggers when pressing a button the the B-Station2. |
| Playback | Events generated by the playbacks |
| 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. |
| OSC | OSC network message |
| 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 |
| System | Events such as 'Start up' |
| Timer | There are 4 internal timers in the B-Station2. 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 B-Station2. 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 action-lists have no trigger value. They can only be triggered by the Action task. They are useful for advanced programming. |

Action-lists 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 action-list the action belongs to. For example, when the trigger type is set to 'Channel' and the actionlist is *DMX Input* it will only respond to the selected DMX channel inserted via the physical DMX connections on the device. If the trigger-type is set to *Channel* and the actionlist is *Art-Net* it will only respond to the selected channel inserted via Art-Net.

Figure 8.3 shows the screen when editing an action.

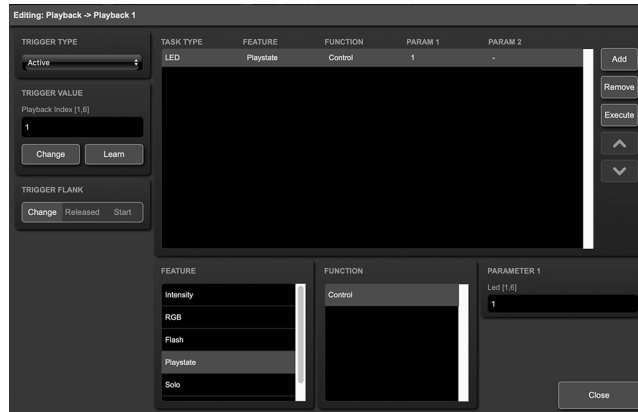


Figure 8.3: Editing an action

A trigger is determined by the trigger-type, trigger-value and trigger-flank fields. Although these fields are not applicable for all action-lists and are therefore 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 Button actionlist there is the choice between various trigger types. By selecting 'Short press' for trigger type and '1' for Trigger value, the action will only be executed when the first button is shortly pressed. If 'Long press' is selected for trigger type and the trigger value is set to '2', the action will only execute when the second button of the B-Station is pressed for longer than one second.

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 Kiosk actionlist, 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 action-list 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 the action 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 fixtures 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 OSC to DMX the OSC action specifies the URI (e.g. /dmx100) and flank (e.g. OnChange) on which it will trigger. The actual OSC (float) data received in the message will be passed along and fed into the action. Then when a task (e.g. DMX) uses the function *Control* this OSC float-level will be used for setting the DMX value.

A task can be tested by selecting it and pressing the *Execute* button in the action-edit 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 one or more action-lists, actions and tasks. These templates configure the B-Station2 to perform typical functions; for example convert Art-Net to DMX or control the 6 playbacks through the buttons. 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 Show Control; 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 direction to be set to output 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 B-Station2. 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.

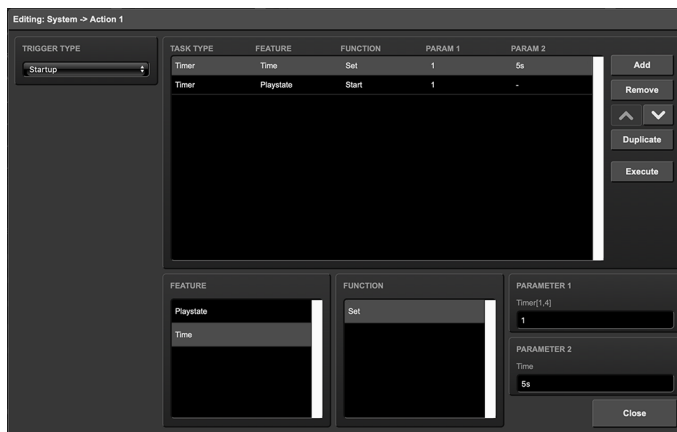
Please note that the variable values are not stored between power cycles. You can see the status of the ten variables in the monitor page, under *Timers*.

8.6 Timers

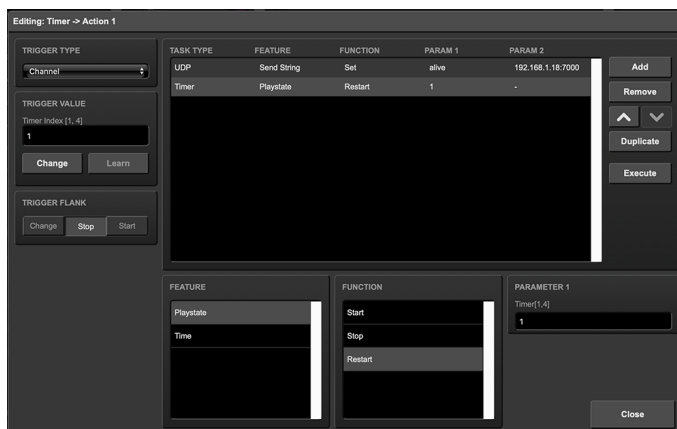
The show control system of the B-Station2 features four internal timers. By using tasks, the timers can be set to certain durations and they can be started. Once started the timers will countdown to zero. When the timer reaches zero it will generate an event that can be captured by using the *Timer* action list. 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 B-Station2 is still 'alive'.

- Setup and start the timer at power-on. This is done by creating an action in the *System* actionlist.



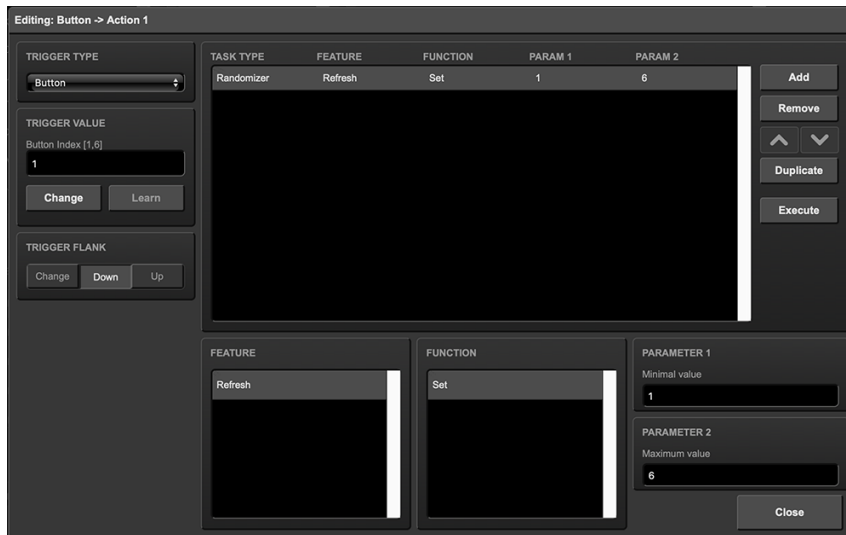
- If the timer expires then send the UDP message and restart the timer. This is done by creating an action in the *Timer* actionlist.



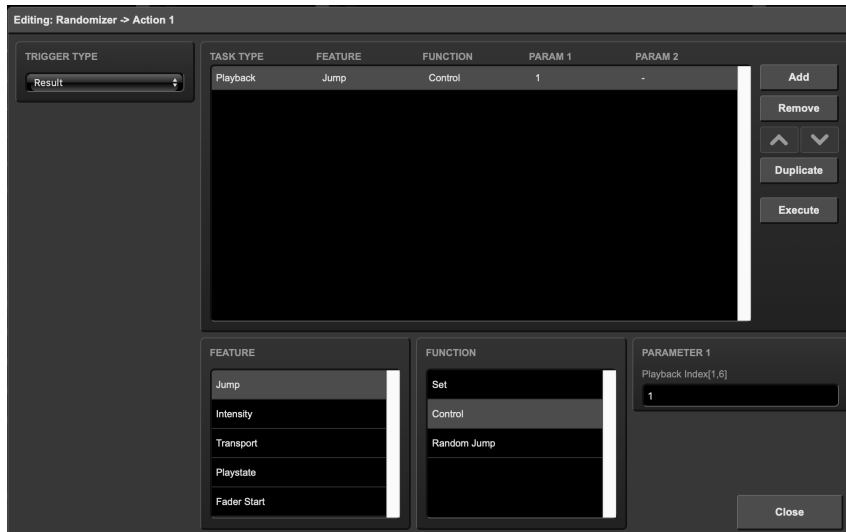
8.7 Randomizer

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 button to trigger a random cue.

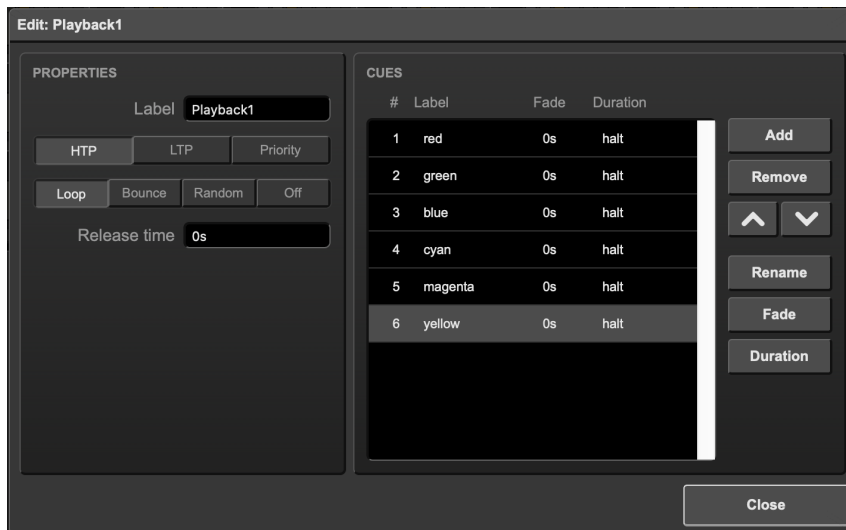
- Inside the *Button* actionlist is an action with Button 1 as a trigger. This triggers the task Randomizer, which is set to a range of between 1 and 10. (Parameters 1 and 2 of the task Randomizer)



- Next in the actionlist *Randomizer* there is an action triggered by the Randomizer's result. The task *Playback*, controlled by the result of the Randomizer, jumps to a cue in Playback 1. (The playback is chosen in Parameter 1 of the task Playback).



- When the button is pressed the randomizer picks a number between one and 10, as defined in the *Randomizer* task. The *Playback* task receives this number and triggers the corresponding cue.



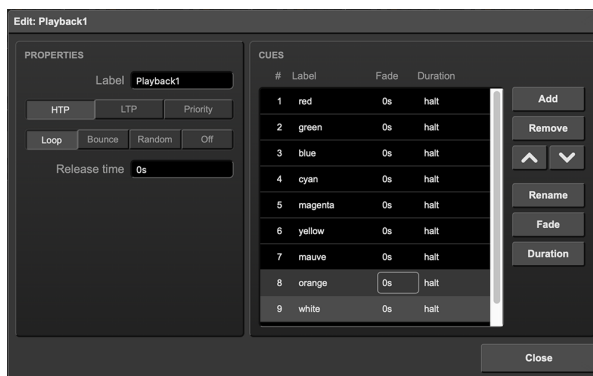
8.8 User Lists

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 there-

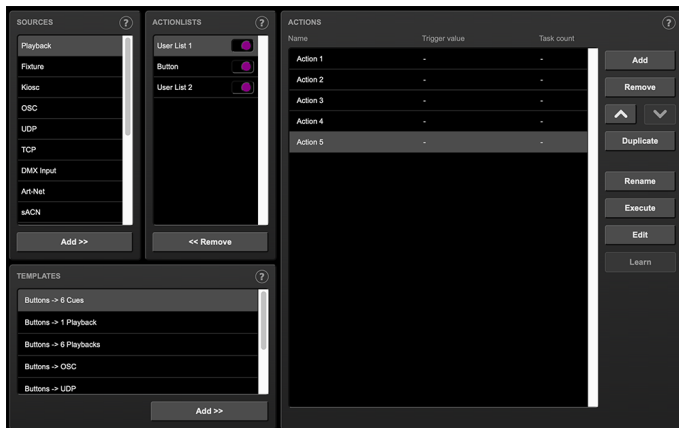
for will not be triggered by any communication or other event. The purpose of User Lists is to have extra actionlists containing actions which can be triggered by explicitly linking to it. See page 84 for details of the *Link* feature in the *Action* task.

The following example demonstrates the use of an User List. Imagine a scenario in which the buttons are used to change between ten lighting scenes. Because the B-Station2 has only six buttons, one button will be used to switch to the second 'page'.

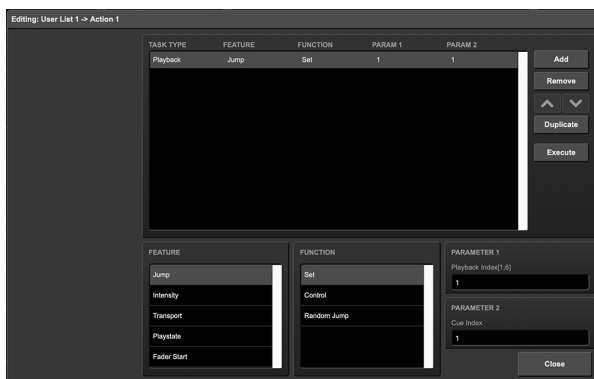
- Create a playback with 10 cues



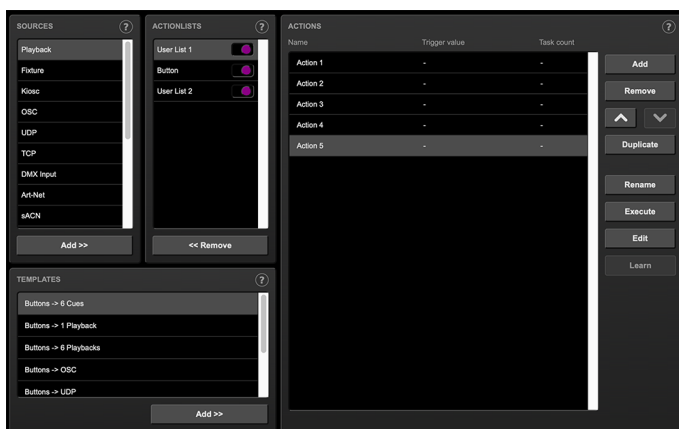
- Create two User Lists with five actions. One User List for each 'page'.



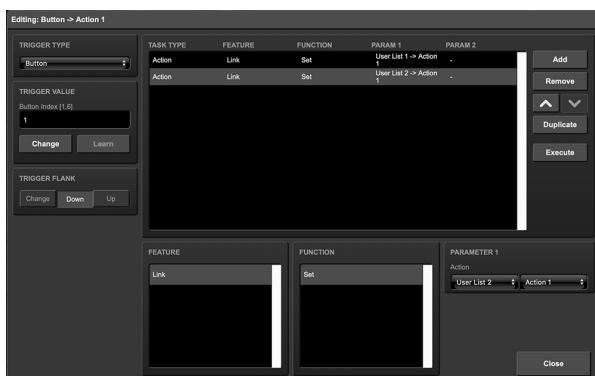
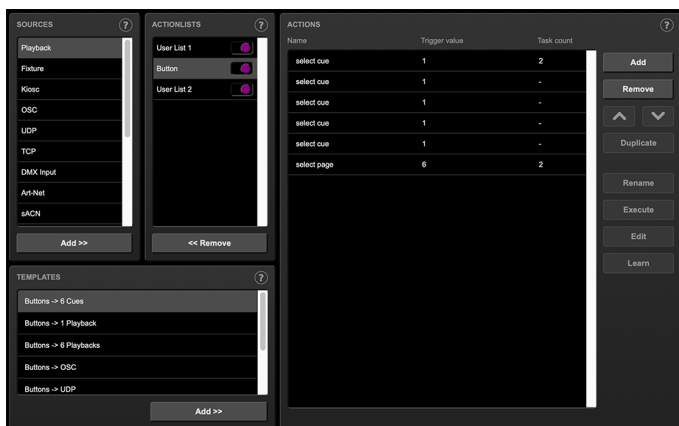
Set the actions in the first User List to jump to cues 1,2,3,4 and 5.



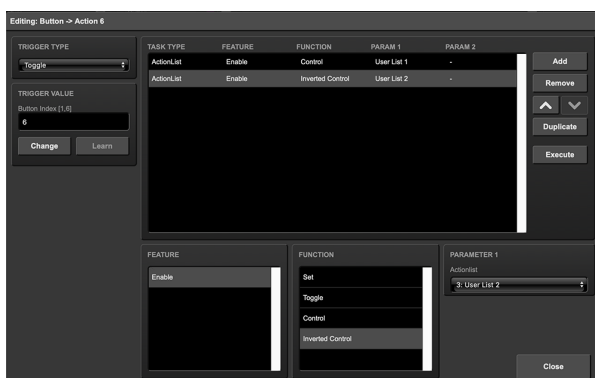
Set the actions in the second User List to jump to cues 6,7,8,9 and 10.



- Add an Button actionlist and link the first five buttons to the action in both User Lists.



- Use the sixth button to enable or disable the two User Lists.



For brevity the start-up state is ignored. Usually additional actions are created to enable/disable the User Lists on start-up.

Chapter 9

Protocol Conversion

The B-Station2 supports various Ethernet-based protocols. Although some protocols are predominantly used for triggering the internal playbacks (such as UDP and OSC) the B-Station2 is capable of converting one protocol into another. This chapter provides an insight on which conversions are possible and how to set them up.

All possible conversions can be organised into two categories: Converting Control Protocols and Converting DMX Universe Protocols.

9.1 Converting Control Protocols

The first 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.

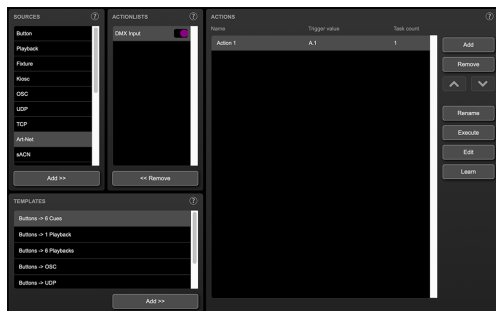
| Control Protocols | Information |
|-------------------|--|
| UDP | - |
| TCP | - |
| OSC | percentage [0%,100%], number, string, colour, On/Off |
| DMX | number [0,255] |
| Art-Net | number [0,255] |
| sACN | number [0,255] |

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.

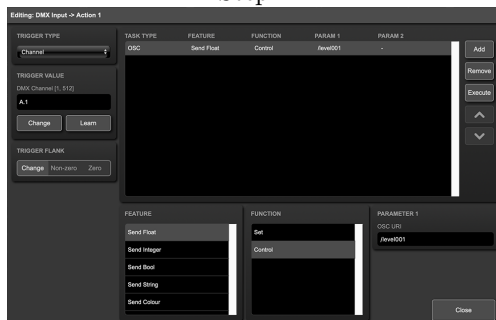
Setting up a conversion is done in the *Show Control* page. First add the incoming protocol from the 'Sources' table into the *Actionlist* table. Then add an action to this new actionlist. Inside this action the trigger-flank field (if visible) should

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.



Step 1



Step 2

Figure 9.1: Conversion from DMX to OSC

9.2 Converting DMX Universe Protocols

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 B-Station2 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 B-Station2. 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 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 playbacks.

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 B-Station2 process the 512 channels as a whole rather than 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 B-Station2'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.

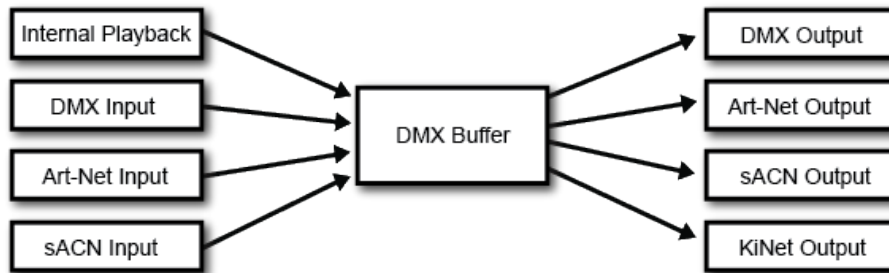


Figure 9.2: DMX merging data flow

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 of the Settings page.

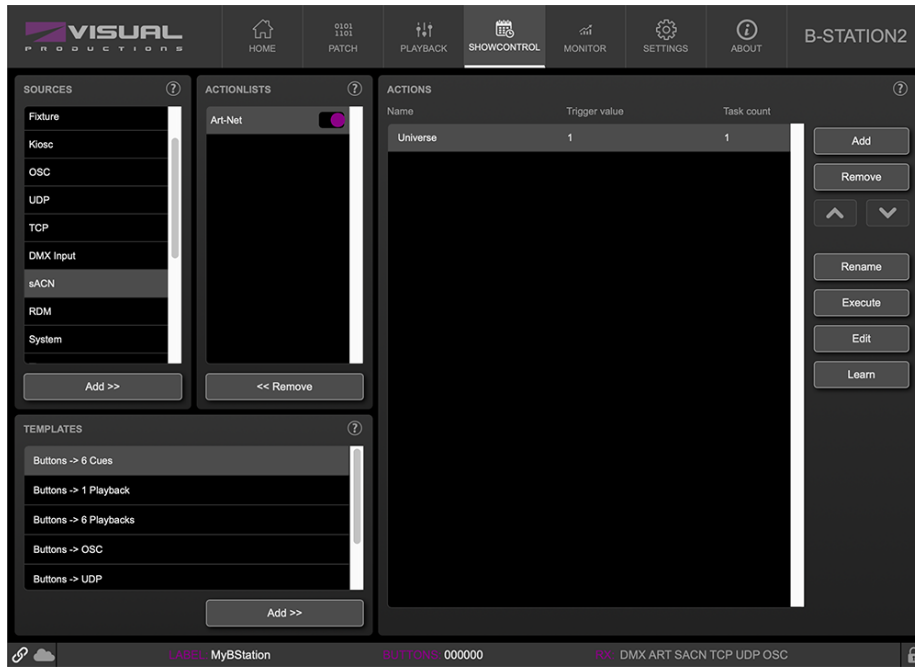


Figure 9.3: Converting Art-Net to DMX step 1

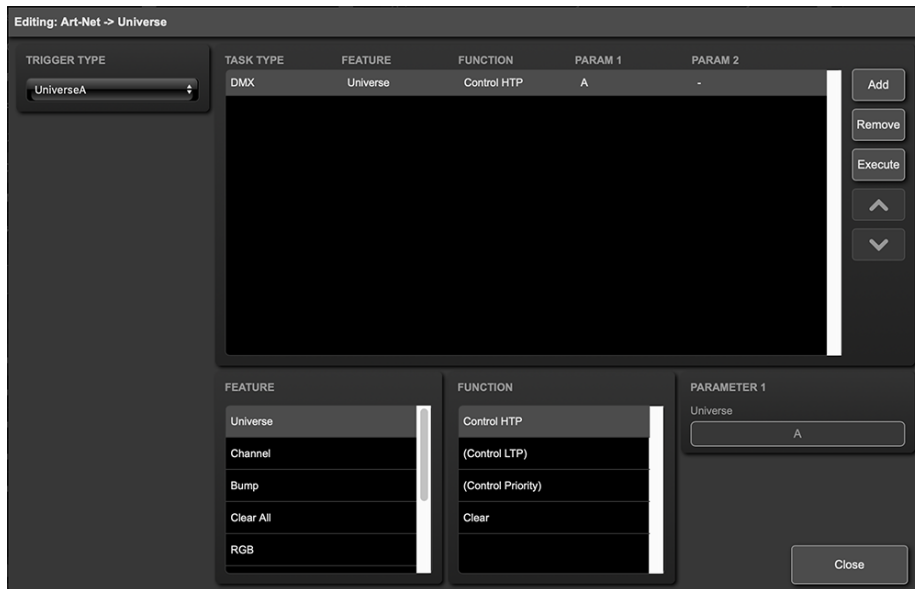


Figure 9.4: Converting Art-Net to DMX step 2

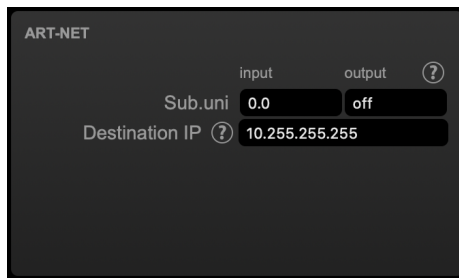


Figure 9.5: Converting Art-Net to DMX step 3

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.

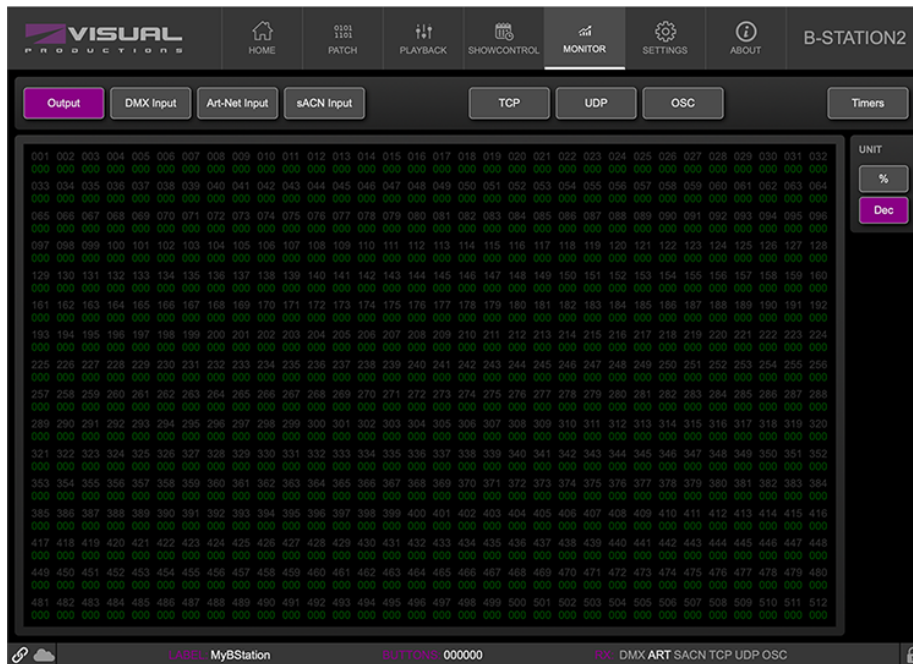


Figure 10.1: DMX Monitor page

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 send or received over UDP |
| OSC | The messages send 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.

The screenshot shows the OSC Monitor interface with a dark theme. At the top, there are tabs for 'Output', 'DMX Input', 'Art-Net Input', 'sACN Input', 'TCP', 'UDP', 'OSC', and 'Timers'. The 'OSC' tab is selected. The main area is divided into two columns: 'INPUT' and 'OUTPUT'. Each column has a table with three columns: 'IP', 'URI', and 'Argument'.

| INPUT | | | OUTPUT | | |
|-------------------|---------|----------|-------------------|----------|----------|
| IP | URI | Argument | IP | URI | Argument |
| 192.168.1.14:8000 | /heart2 | beat2 | 192.168.1.40:8000 | /button4 | Off |
| 192.168.1.14:8000 | /test | hello | 192.168.1.40:8000 | /button2 | Off |
| 192.168.1.14:8000 | /heart2 | beat2 | 192.168.1.40:8000 | /button2 | On |
| 192.168.1.14:8000 | /test | hello | 192.168.1.40:8000 | /button4 | On |
| 192.168.1.14:8000 | /heart2 | beat2 | 192.168.1.40:8000 | /button3 | Off |
| - | - | - | 192.168.1.40:8000 | /button3 | On |
| - | - | - | - | - | - |
| - | - | - | - | - | - |
| - | - | - | - | - | - |
| - | - | - | - | - | - |

Figure 10.2: OSC Monitor page

The Monitor page also presents an overview of timer and variable values. See figure 10.3

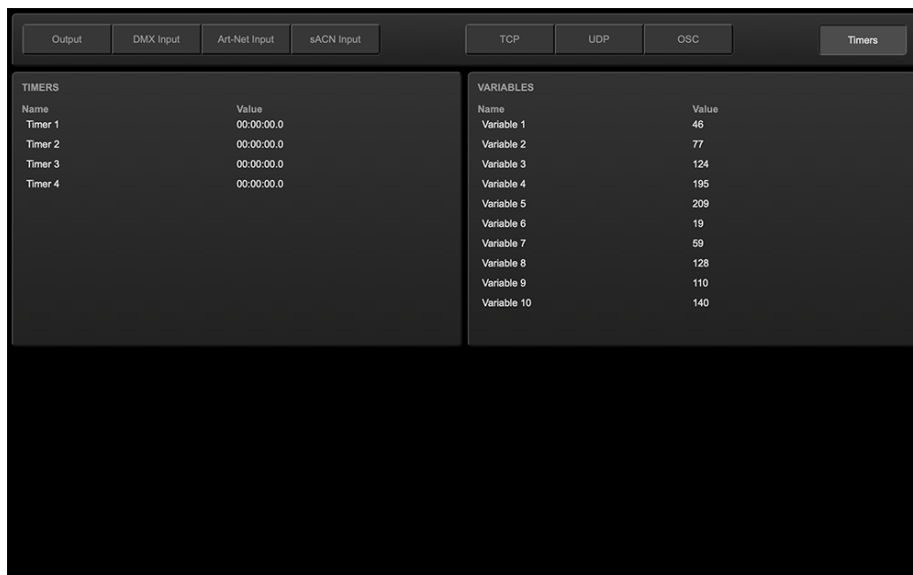


Figure 10.3: Timer & Variable Monitor

Chapter 11

Settings

The B-Station2's settings are organised into sections, see the *Settings* page figure 11.1. This chapter will discuss each section.

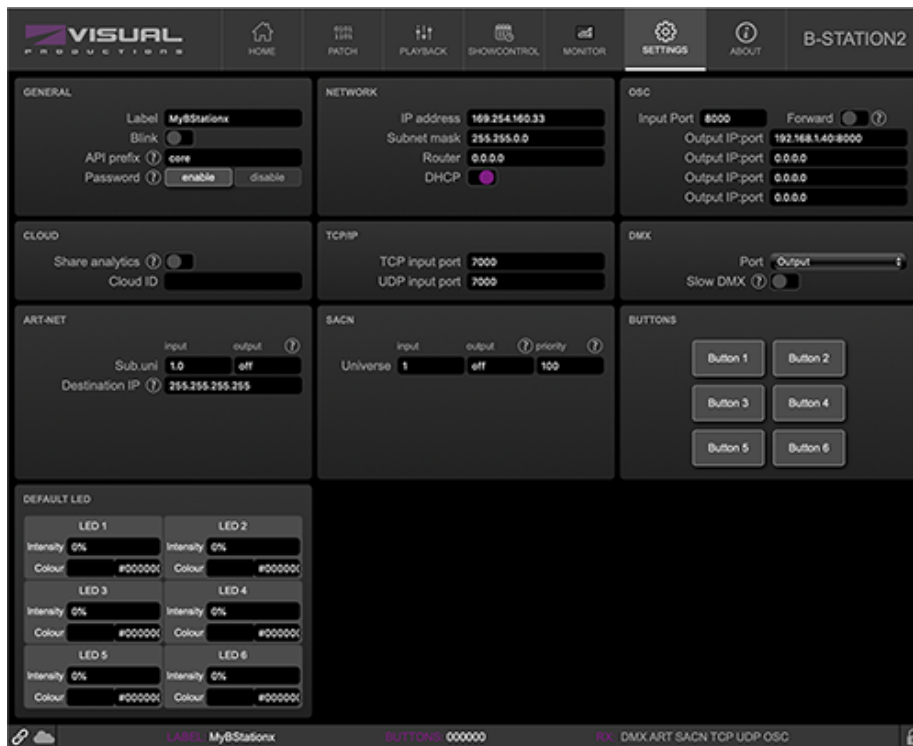


Figure 11.1: Settings page

11.1 General

You can change the B-Station2'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.

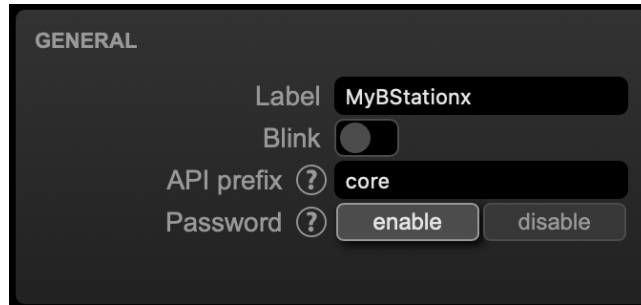


Figure 11.2: General Settings

Unauthorised users can be prevented by making changes to the B-Station2 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 5.2). 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 B-Station2. 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 5 at page 23).

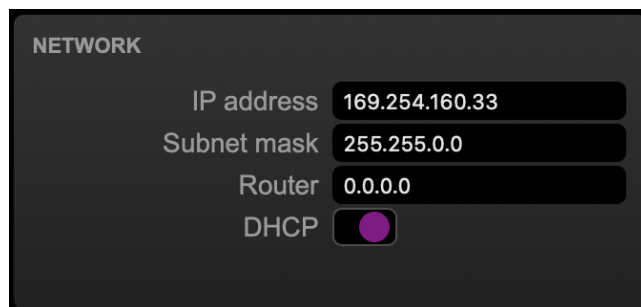


Figure 11.3: IP Settings

11.3 Buttons

Here the six buttons of the B-Station2 are represented in the browser interface. They can be pressed for testing purposes.

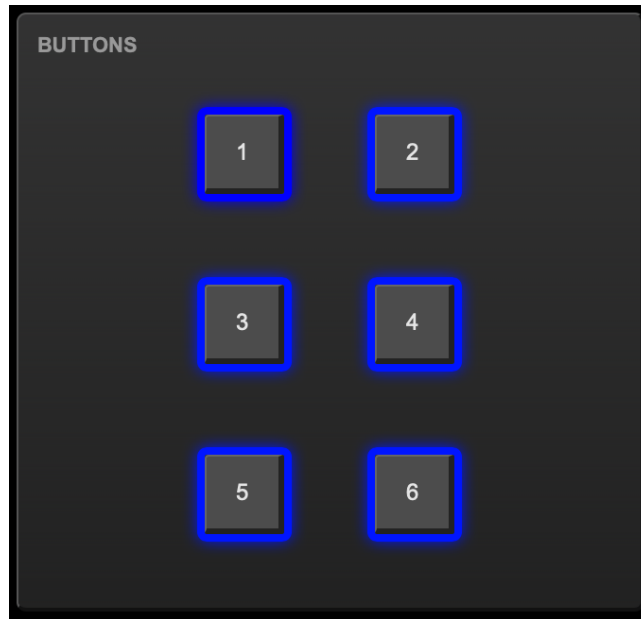


Figure 11.4: Buttons

11.4 DMX

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

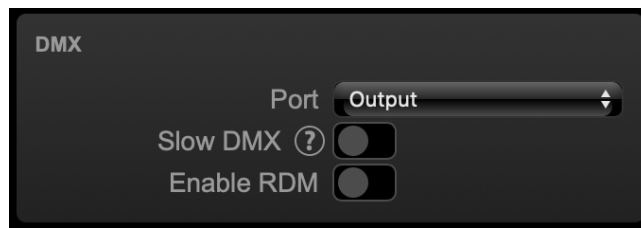


Figure 11.5: DMX settings

When the *Slow DMX* checkbox is enabled, the B-Station2 will slowdown the rate at which it sends out DMX from its ports. 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.

11.5 Art-Net

The *Art-Net* feature in the B-Station2 supports 1 universe out and 1 universes in.

This universe 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.

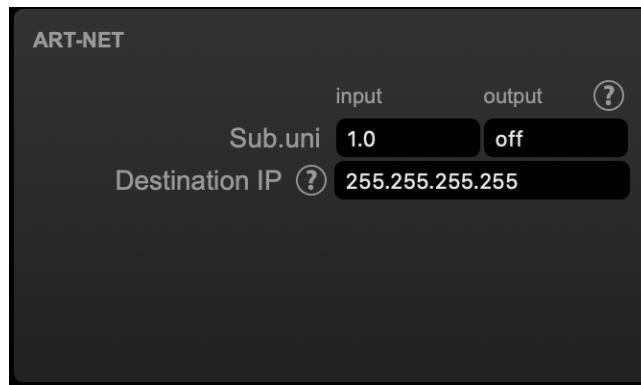


Figure 11.6: Art-Net settings

The *Destination IP* determines where the outgoing Art-Net data will be send 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.

11.6 sACN

The B-Station2 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.

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

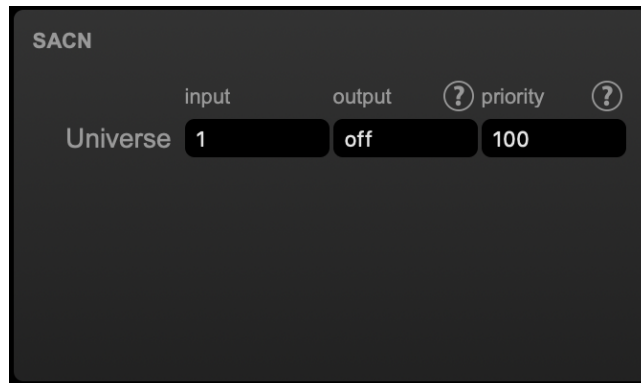


Figure 11.7: sACN settings

11.7 OSC

External equipment sending OSC messages to the B-Station2 need to be aware of the number specified in the 'Port' field. This is the port the B-Station2 listens to for incoming messages.

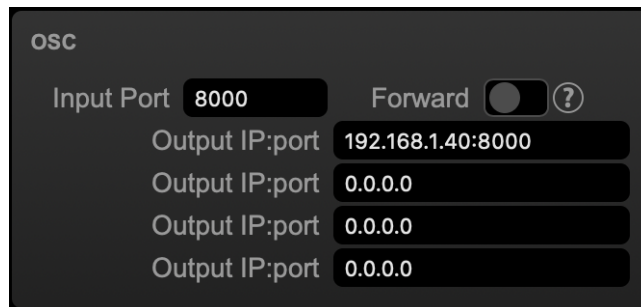


Figure 11.8: OSC Settings

The B-Station2 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 'ipaddress:port' format in these fields, e.g. "192.168.1.11:9000". If a field should not be used that 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 B-Station2 copy every incoming OSC message and send it the addresses specified in the 'Out IP' fields.

11.8 TCP/IP

Defines the listening ports for TCP and UDP messages. External system intending to send TCP or UDP message to the B-Station2 should need to know

the unit's IP address and this port number. By default both ports are set to 7000.

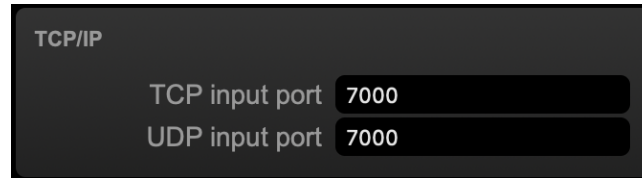


Figure 11.9: TCP/IP settings

11.9 Cloud

By enabling the *Share Analytics* options, you will help Visual Productions improve its products. When enabled, the B-Station2 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.

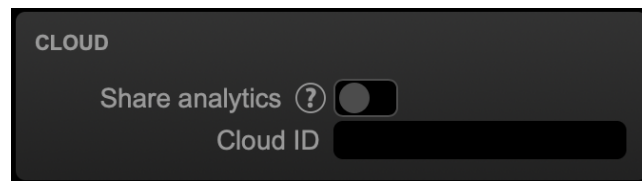


Figure 11.10: Cloud settings

11.10 Default LED Settings

The *Default LED* settings can be set here. These settings will apply every time the device is powered up. Further usage of the LEDs is programmed in the Show Control page using the LED task. For more information, see chapter 8.

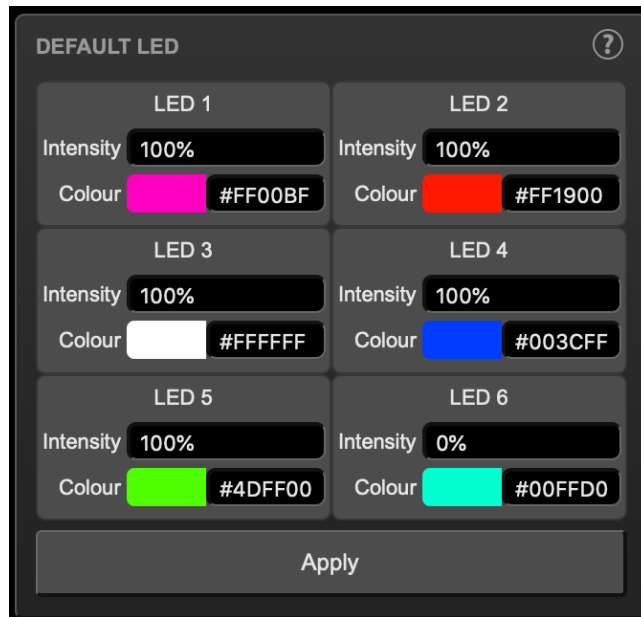


Figure 11.11: Default LED settings

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

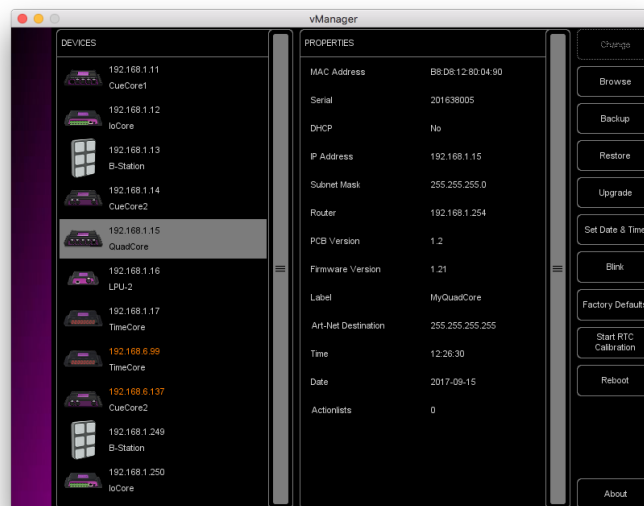


Figure 12.1: vManager

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.

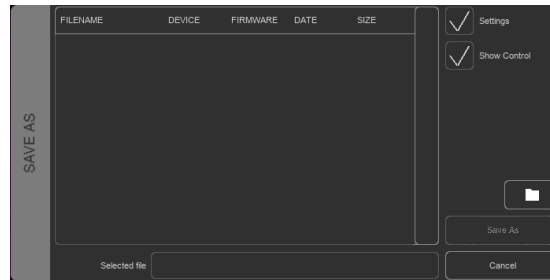


Figure 12.2: Creating a backup

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.

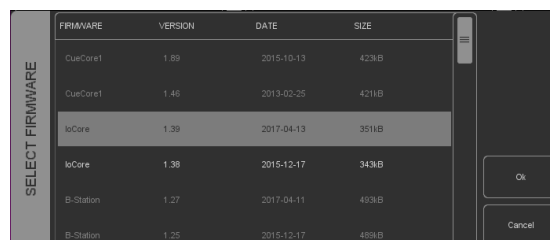


Figure 12.3: Firmware upgrade

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.

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 on the memory. They will be completely erased and all settings will be 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 software 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 10.15 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 18.04 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.

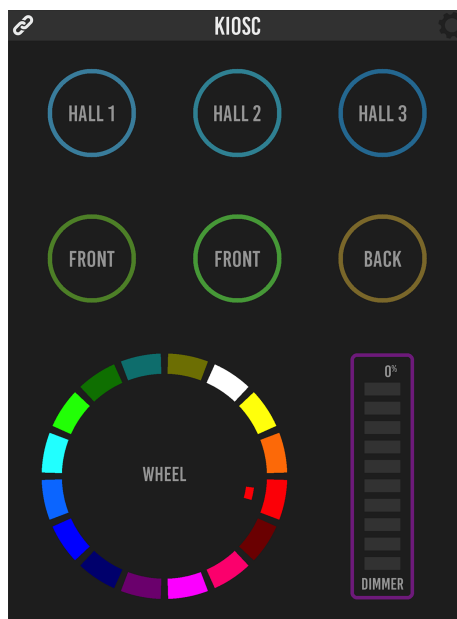


Figure 13.1: Kiosc

Kiosc is the ideal way to remote control our solid-state lighting controllers like CueluxPro, CueCore1, CueCore2, QuadCore, IoCore1, IoCore2, LPU-2, Dal-iCore, B-Station1 and the B-Station2. 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 B-Station2. The different types are accompanied with values and flanks.

A.1 Button

| Trigger Type | Trigger Value | Flank | Description |
|--------------|---------------|--------|----------------------------|
| Button | Button Index | Change | Button pressed or released |
| Button | Button Index | Down | Button pressed |
| Button | Button Index | Up | Button released |

A.2 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.3 Playback

| Trigger Type | Trigger Value | Flank | Description |
|--------------|----------------|----------|---------------------------------------|
| Active | Playback Index | Change | Playback starts or stops |
| Active | Playback Index | Released | Playback stops |
| Active | Playback Index | Start | Playback starts |
| Release | Playback Index | Change | Playback starts or finishes releasing |
| Release | Playback Index | Released | Playback finished releasing |
| Release | Playback Index | Release | Playback starts releasing |
| Released | Playback Index | Change | Playback starts or stops |
| Released | Playback Index | Playing | Playback starts playing |
| Released | Playback Index | Released | Playback finished releasing |
| Playing | Playback Index | Change | Playback starts or stops |
| Playing | Playback Index | Release | Playback starts releasing |
| Playing | Playback Index | Playing | Playback starts playing |
| Running | Playback Index | Change | Playback starts or pauses |
| Running | Playback Index | Paused | Playback pauses |
| Running | Playback Index | Playing | Playback starts playing |
| Intensity | Playback Index | Change | Playback intensity changes |
| Intensity | Playback Index | Non-zero | Playback intensity becomes >0% |
| Intensity | Playback Index | Zero | Playback intensity becomes 0% |
| End | Playback Index | - | Playback stops playing |
| CueChange | Cue Index | Change | Cue activated or deactivated |
| CueChange | Cue Index | Inactive | Cue becomes inactive |
| CueChange | Cue Index | Active | Cue becomes active |
| CueIndex | Playback Index | - | Active cue index changed |
| CueLabel | Playback Index | - | Label of the current Cue |
| TrackBegin | Playback Index | - | Track starts |

A.4 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.

A.5 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.6 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.7 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.8 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.9 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.10 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.11 System

| Trigger Type | Trigger Value | Flank | Description |
|--------------------|---------------|--------|---|
| Startup | - | - | The B-Station2 has been power up |
| Network Connection | - | Change | Network connection established or lost |
| Network Connection | - | Stop | Network connection lost |
| Network Connection | - | Start | Network connection established |
| ReleasedByMaster | - | Change | Master (e.g. CueluxPro) released or obtained connection |
| ReleasedByMaster | - | Stop | Master released connection |
| ReleasedByMaster | - | Start | Master obtained connection |

A.12 Variable

| Trigger Type | Trigger Value | Flank | Description |
|--------------|----------------|------------|--|
| Channel | Variable Index | - | The specified variable changes |
| Variable 1 | Number [0,255] | Change | Variable 1 becomes equal or not equal 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 equal to number |
| Variable 2 | Number [0,255] | Change | Variable 2 becomes equal or not equal 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 equal to number |
| Variable 3 | Number [0,255] | Change | Variable 3 becomes equal or not equal 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 equal to number |
| ... | | | |
| Variable 9 | Number [0,255] | Change | Variable 9 becomes equal or not equal 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 equal or not equal 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 equal to number |

A.13 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 |

A.14 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 |

Appendix B

Task Types

Tasks allow you to automate the functionality in the B-Station2. 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 Button

Force the Button actions to be triggered.

| Feature | Function | Parameter 1 | Parameter 2 |
|---------|----------|-------------|-------------|
| Refresh | Set | - | - |

B.2 Playback

Manipulate one of the six playbacks.

| Feature | Function | Parameter 1 | Parameter 2 |
|-------------|------------------|----------------|----------------------|
| Intensity | Set | Playback Index | percentage [0%,100%] |
| Intensity | Control | Playback Index | - |
| Transport | Pause | Playback Index | - |
| Transport | Release | Playback Index | - |
| Transport | Go+ | Playback Index | - |
| Transport | Go- | Playback Index | - |
| Transport | Jump | Playback Index | Cue number[1,64] |
| Transport | Solo | Playback Index | - |
| Transport | Random Solo | Playback Index | - |
| Play State | Toggle | Playback Index | - |
| Play State | Control | Playback Index | - |
| Play State | Inverted Control | Playback Index | - |
| Fader Start | Toggle | Playback Index | - |
| Fader Start | Control | Playback Index | - |
| Fader Start | Inverted Control | Playback Index | - |

B.3 Playback Master

Manipulate the master settings on the Playback page.

| Feature | Function | Parameter 1 | Parameter 2 |
|--------------|----------|-------------|----------------------|
| Grand Master | Set | - | percentage [0%,100%] |
| Grand Master | Control | - | - |
| Black-out | Set | - | true or false |
| Black-out | Control | - | - |
| Release | All | - | - |

B.4 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.5 LED

Control the button's LEDs.

| Feature | Function | Parameter 1 | Parameter 2 |
|-----------|--------------|-------------|----------------------|
| Intensity | Set | LED [1,6] | Percentage [0%,100%] |
| Intensity | Set All | - | Percentage [0%,100%] |
| Intensity | Toggle | LED [1,6] | Percentage [0%,100%] |
| Intensity | Control | LED [1,6] | - |
| Intensity | Decrement | LED [1,6] | Delta [1,255] |
| Intensity | Increment | LED [1,6] | Delta [1,255] |
| Intensity | Fade in 0.3s | LED [1,6] | Percentage [0%,100%] |
| Intensity | Fade in 1s | LED [1,6] | Percentage [0%,100%] |
| Intensity | Fade in 3s | LED [1,6] | Percentage [0%,100%] |
| Intensity | Solo | LED [1,6] | Percentage [0%,100%] |
| Colour | Set | LED [1,6] | #RRGGBB |
| Colour | Set All | - | #RRGGBB |
| Colour | Toggle | LED [1,6] | #RRGGBB |
| Colour | Control | LED [1,6] | - |
| Colour | Fade in 0.3s | LED [1,6] | #RRGGBB |
| Colour | Fade in 1s | LED [1,6] | #RRGGBB |
| Colour | Fade in 3s | LED [1,6] | #RRGGBB |
| Colour | Solo | LED [1,6] | #RRGGBB |
| Pulse | Set | LED [1,6] | Percentage [0%,100%] |
| Pulse | Control | LED [1,6] | - |
| Pulse | Solo | LED [1,6] | Percentage [0%,100%] |
| Blink | Set | LED [1,6] | Percentage [0%,100%] |
| Blink | Control | LED [1,6] | - |
| Blink | Solo | LED [1,6] | Percentage [0%,100%] |

B.6 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.7 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.8 DMX

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

| Feature | Function | Parameter 1 | Parameter 2 |
|--------------|------------------|-------------------|------------------|
| Universe | Control HTP | - | - |
| Universe | Control LTP | - | - |
| Universe | Control Priority | - | - |
| Universe | Clear | - | - |
| Set Channel | Set | DMX Channel | DMX Value |
| Set Channel | Toggle | DMX Channel | - |
| Set Channel | Control | DMX Channel | - |
| Set Channel | Inverted Control | DMX Channel | - |
| Set Channel | Decrement | DMX Channel | - |
| Set Channel | Increment | DMX Channel | - |
| Bump Channel | Set | DMX Channel | DMX Value |
| Bump Channel | Control | DMX Channel | - |
| Clear All | Set | - | - |
| RGB | Set | DMX Address | RGB Colour Value |
| RGB | Control | DMX Address | - |
| XY | Control | DMX Address | - |
| XxYy | Control | DMX Address | - |
| Ii | Set | DMX Address | Intensity Value |
| Ii | Control | DMX Address | - |
| Block | Control | DMX Start Address | DMX End Address |

B.9 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 41.

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.10 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 5.2.

B.11 Action

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

| Feature | Function | Parameter 1 | Parameter 2 |
|---------|----------|-------------|-------------|
| Link | Set | Action | - |

B.12 Action-list

Manipulate an action-list.

| Feature | Function | Parameter 1 | Parameter 2 |
|---------|------------------|-------------|-------------|
| Enable | Set | Action-list | On or Off |
| Enable | Toggle | Action-list | - |
| Enable | Control | Action-list | - |
| Enable | Inverted Control | Action-list | - |

B.13 Randomiser

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 43.

B.14 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 |
|-----------------------|--|
| Buttons ->6 Playbacks | Each button controls one playback. The LEDs indicate which playbacks are active. |
| Buttons ->6 Cues | Each button will jump one of the first six cues inside playback 1. The LEDs will indicate which cue is active. Playback 1 should contain at least six cues. |
| Buttons ->OSC | Each button will send an OSC message. |
| Buttons ->UDP | Each button will broadcast a UDP message on down and up movement. |
| Buttons ->DMX | Each button will control one of the first six DMX channels. |
| Buttons ->LPU-2 | The buttons will control the first six playbacks in the LPU-2 device. The LPU-2's feedback is used to control the button's LEDs. Specify the LPU-2's IP address in the OSC table on the Settings page. |
| Buttons ->CueCore2 | Control the playbacks on a CueCore2/QuadCore through the OSC fixed mapping. |
| Art-Net ->DMX | Converts incoming Art-Net to DMX. Specify the Art-Net subnet and universe on the Settings page. |
| OSC ->LEDs | Creates actions for controlling the button's LEDs through OSC. |

Appendix D

API

The B-Station2 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 B-Station2, however, it is also capable to send information back. This feedback mechanism is discussed at the end of this chapter, on page 91.

D.1 OSC

The following table uses playback #1 as an example. The number '1' can be replaced by any number in the range of [1,6].

| URI | Parameter | Description |
|----------------------|-----------|---|
| /core/pb/1/go+* | - | Jump to the next cue in playback #1 |
| /core/pb/1/go- | - | Jump to the previous cue in playback #1 |
| /core/pb/1/jump | integer | Jump to a specific cue in playback #1 |
| /core/pb/1/release | - | Release the playback |
| /core/pb/1/intensity | float | Set the playback's intensity |
| /core/pb/release | - | Release all playbacks |
| /core/pb/intensity | float | Set the master intensity |
| /core/pb/blackout | bool | Set the blackout |
| /core/pb/solo | integer | Start a solo playback |

* This command does not work when the parameter is off or a 0 value.

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 action list #1 |
| /core/al/1/enable | bool | Set the 'enable' checkbox for action list #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/bu/1 | bool | Set the button state |
| /core/ld/1 | float | Set the LED intensity |
| /core/ld/1 | colour | Set the LED colour |
| /core/dmx/1 | integer | Set the value of a DMX channel |
| /core/blink | - | Momentarily flashes the B-Station2'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 B-Station2 can controlled by using the following ASCII strings (human readable text) messages:

The following table uses playback #1 as an example. The number '1' can be replaced by any number in the range of [1,6].

| String | Description |
|-----------------------------|---|
| core-pb-1-go+ | Jump to the next cue in playback #1 |
| core-pb-1-go- | Jump to the previous cue in playback #1 |
| core-pb-1-jump=<integer> | Jump to a specific cue in playback #1 |
| core-pb-1-release | Release the playback |
| core-pb-1-intensity=<float> | Set the playback's intensity |
| core-pb-release | Release all playbacks |
| core-pb-intensity=<float> | Set the master intensity |
| core-pb-blackout=<bool> | Set the blackout |
| core-pb-solo=<integer> | Start a solo playback |

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 action list #1 |
| core-al-1-enable=<bool> | Set the 'enable' checkbox for action list #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-bu-1=<bool> | Set the button state |
| core-ld-1=<float> | Set the LED intensity |
| core-ld-1=<colour> | Set the LED colour |
| core-dmx-1=<integer> | Set the value of a DMX channel |
| core-blink | Momentarily flashes the B-Station2's LED |
| core-hello | The unit will reply with the same Hello message |

An example for setting LED #3 to red is `core-ld-3=#ff0000`. To set all LEDs to blue send `core-ld=#0000ff`.

D.3 HTTP

HTTP (Hyper Text Transfer Protocol) is the standard protocol to access web pages. It can also be used to control the B-Station2, using the URLs listed below.

The following table uses playback #1 as an example. The number '01' can be replaced by any number in the range of [01,06].

| Description | URL | Parameter Range | Example |
|------------------------|----------------|-----------------|--|
| Playback Go Forward | /ajax/pbXX/go+ | - | http://192.168.1.10/ajax/pb01/go+ |
| Playback Go Back | /ajax/pbXX/go- | - | http://192.168.1.10/ajax/pb01/go- |
| Playback Jump | /ajax/pbXX/jmp | [1, 32] | http://192.168.1.10/ajax/pb01/jump=9 |
| Release playback | /ajax/pbXX/rel | - | http://192.168.1.10/ajax/pb01/rel |
| Set playback intensity | /ajax/pbXX/int | [0.0, 1.0] | http://192.168.1.10/ajax/pb01/int=0.55 |
| Release all playbacks | /ajax/pb/rel | - | http://192.168.1.10/ajax/rel |
| Set master intensity | /ajax/pb/int | [0.0, 1.0] | http://192.168.1.10/ajax/pb/int=0.55 |
| Set blackout | /ajax/pb/bo | true/false | http://192.168.1.10/ajax/pb/bo=true |
| Start a solo playback | /ajax/pb/sol | [1, 6] | http://192.168.1.10/ajax/pb/sol=3 |

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 |
|-----------------------|------------|-----------------|---------------------------------------|
| Set the button state | /ajax/bu/1 | true/false | http://192.168.1.10/ajax/bu/1=true |
| Set the LED intensity | /ajax/ld/1 | [0.0, 1.0] | http://192.168.1.10/ajax/ld/1=0.7 |
| Set the LED colour | /ajax/ld/1 | #RRGGBB | http://192.168.1.10/ajax/ld/1=#ff0000 |
| Blink LED | /ajax/bli | - | http://192.168.1.10/ajax/blink |

Omit the index in order to set all LEDs at once. For example <http://192.168.1.10/ajax/ld=0.5>.

You can send your HTTP GET requests to port 80.

D.4 Feedback

The B-Station2 is able to send feedback to external equipment using its API, so called 'clients'. The B-Station2 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 B-Station2 will send back to its clients.

| OSC | UDP |
|----------------------|---------------------|
| /core/bu/1 | core-bu-1 |
| /core/pb/1/intensity | core-pb-1-intensity |
| /core/pb/1/rate | core-pb-1-rate |
| /core/pb/intensity | core-pb-intensity |
| /core/pb/1/active | core-pb-1-active |
| /core/pb/1/cue | core-pb-1-cue |
| /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 B-Station2 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.

Index

- vManager, 63
- 16-bit, 28
- Action, 37, 84
- Action list, 38
- Actionlist, 76
- Address, 28
- Android, 65
- Append, 32
- Art-Net, 13, 59, 73
- Auto-sensing, 23
- Backup, 64
- Blink, 57, 65
- Bounce, 36
- Box contents, 10
- Clear, 31
- Cloud, 61
- Cloud ID, 61
- Control, 41
- Conversion, 48
- Cues, 31
- Date, 65
- Destination IP, 59
- DHCP, 15, 24, 57
- DIN Rail, 13
- Discover, 29
- DMX, 12, 58
- DMX Input, 70
- Duration, 36
- Edit, 33
- Effect, 34
- Execute, 41
- Factory Defaults, 65
- Fade, 36
- Feature, 77
- Feedback, 91
- Fine, 28
- Firmware, 64
- Flank, 40, 70
- Forum, 11
- Forward, 60
- Function, 77
- FX Engine, 34
- General, 57
- Halt, 36
- HTP, 35
- HTTP, 90
- Insert, 32
- iOS, 65
- IP, 57
- Jump, 36
- KiNet, 49
- Kiosc, 67, 73
- Label, 57
- LED, 61
- Limitations, 9
- Load, 33
- Locate, 29
- Loop, 36
- LTP, 35
- macOS, 66
- Memory, 11
- Merge, 32
- Network Interface Controller, 81
- Off, 36
- OSC, 14, 60, 72, 86
- Password, 57
- Personality, 27
- Playback Editor, 34
- PoE, 22

Port, 61
Port Forwarding, 26, 57
Precedence, 35
Priority, 35
Programmer, 31
Protocol Conversion, 48

Random, 36
Randomizer, 43, 74, 84
RDM, 13, 29, 59
RdmSplitter, 13
Reboot, 65
Record, 32
Release, 32
Release Time, 36
Rename, 33
Repeat, 36
Replace, 32
Reset button, 25
Router, 57

sACN, 14, 59, 73
Share Analytics, 61
Show Control, 37
Single Dimmer, 83
Slow DMX, 58
Snap, 66
Source, 38
Subnet mask, 24
System, 74, 83

Task, 38
Task type, 77
TCP, 14, 60, 72, 88
TCP/IP, 60
Templates, 41, 85
Time, 65
Timer, 76, 84
Timers, 42
Trigger, 38
Tutorial, 16

Ubuntu, 66
UDP, 14, 60, 72, 88
Uptime, 26
URI, 72
User List, 44

Variable, 75
Variables, 41
Virtual Dimmer, 28

VPN, 26
Wake On Lan, 81
Windows, 66