VISUAL PRODUCTIONS

CUECORE2 MANUAL

Contents

Declaration of Conformity

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

CueCore2

Is in conformity with 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:
November 18th, 2016

Place:
Haarlem, The Netherlands

ing. Maarten Engels
Managing Director
Visual Productions BV

VISUAL PRODUCTIONS BV
IZAAK ENSCHEDEWEG 38A
NL-2031CR HAARLEM
THE NETHERLANDS
TEL +31 (0)23 551 20 30
WWW.VISUALPRODUCTIONS.NL
INFO@VISUALPRODUCTIONS.NL
ABN-AMRO BANK 53.22.22.261
BIC ABNANL2A
IBAN NL18ABNA0532222261
VAT NL851328477B01
COC 54497795

Preface

Thank you for choosing the CueCore2. The engineering team at Visual Productions is proud to present to you the latest development in solid-state lighting control.

Design Goals

After successfully developing and marketing the original CueCore1, the team at Visual Productions set out to design a new generation stand-alone lighting controller. We took our experience in developing solid-state controllers and combined it with the feedback received from CueCore1 users all over the world. We added our lessons learnt from supporting year's worth of projects and installs, all in order to design the best lighting controller for (semi-)permanent installations. During the design process we focused on a set of priorities that we valued the most:

Solid-State

The solid-state aspect was perhaps the most important trait that made the CueCore1 a default choice for many system designers. The CueCore2 continues this design without any moving part, without forced cooling, and its data safely stored in flash memory. The resulted reliability outperforms any PC based lighting system.

Multi-zone Replay Unit

One of the principle functions of the CueCore2 is to playback DMX shows. We took a lot of consideration into making this feature as powerful as possible with the given hardware platform. The playback mechanism we developed can control six different zones. Each zone will be controlled by a completely independent playback. This playback features many options that provides the freedom to the users to employ very smart programming. These options include intensity, rate, precedence, release-time, repeating, fading and inter-cue conditions.

Networking

Our world is one big connected network and the CueCore2 will blend in nicely. This Ethernet-based device is setup via DHCP or static address, hosts a modern web-interface for programming and is of course also powered by PoE.

Protocol Conversion

One of the strongest Unique Selling Points of the products designed at Visual Productions is the number of communication protocols supported. The CueCore2 further raises this bar. It contains protocols familiar to the CueCore1 (DMX, Art-Net, UDP, OSC) and introduces a fresh set of new protocols: sACN, KiNet, TCP and NTP. This vast collection of protocols can be used for recording, sending, triggering and converting.

Scalability

One CueCore2 can do a lot, multiple CueCore2 units can do even more. Through using CueluxPro it is possible to control up to 32 universes by connecting multiple CueCore2 units. For stand-alone scenarios we even developed a brand new Master/Slave protocol that, once set up with just a few mouse clicks, allows 25 CueCore2 to work together and keep 50universes synchronised at 40 frames per second.

We hope that you enjoy integrating the CueCore2 into your lighting designs.

Have fun!

The CueCore2 engineering team,
Michael Chiou
Jurrin de Klerk
Thijs Engels
Guido Visser
Maarten Engels

Chapter 1

Introduction

The CueCore2 is a DMX-512 lighting controller designed for (semi-)permanent installations. This document discusses setting up the device and programming its internal software functions.

Figure 1.1: CueCore2

1.1 Features

The feature set of the CueCore2 includes:

- 2 x DMX-512 optically isolated port (bi-directional)
- Art-Net, sACN, KiNet
- TCP, UDP & OSC
- 4x GPI
- MIDI, MSC &MMC
- SMPTE, MTC & Art-Net timecode

- Scheduling with Real-Time clock, weekdays and sunrise/sunset
- Desktop or DIN Rail mounted
- Kensington lock
- Locked power cable protection
- PoE (Power Over Ethernet) Class I
- Bundled with CueluxPro, vManager and VisualTouch software

1.2 What's in the box?

The CueCore2 packaging contains the following items (see figure 1.2):

- Info card
- CueCore2
- Power supply

Figure 1.2: CueCore2 box contents

1.3 Saving data to memory

This manual will describe how to configure the CueCore2 and program tracks, 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 CueCore2 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 CueCore2 will copy any changes in the RAM memory to its onboard flash memory. Flash memory retains its data even when not powered. The CueCore2 will load all its data back from the flash memory upon startup.

This memory copy process is conducted automatically by the CueCore2 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.

1.4 Comparison

The following table visualises the difference between the QuadCore, CueCore2 and CueCore1 . This overview might prove to be helpful to CueCore1 users considering choosing the model for their new designs.

	QuadCore	CueCore2	CueCore1
CPU	180MHz	180MHz	120MHz
Flash memory	32MB	32MB	8MB
DMX Outputs	4	2	2
DMX Inputs	switchable outputs	switchable outputs	1
MIDI	-	in+out	in+thru+out
GPI	-	4x digital/analog	4x digital
SMPTE	-	input	input
MTC	-	input+output	input
Art-Net	input+output	input+output	input+output
sACN	input+output	input+output	-
KiNet	output	output	-
POE	class I	class I	class I
DHCP	yes	yes	-
NTP	yes	yes	-
Real-time Clock	yes	yes	yes
CueluxPro Licence	yes	yes	yes

1.5 Further Help

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

Chapter 2

Protocols

The CueCore2 is fitted with several communication ports and supports various protocols. This chapter describes these protocols and to which extent they are implemented in the CueCore2

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.

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 CueCore2 has two ports and is therefor able control 1,048 channels. Each port can also be configured to become a DMX input allowing external DMX data to be recorded or to use an external DMX source to trigger events within the CueCore2.

Figure 2.1: Visual Productions' RdmSplitter

2.2 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.

Additional to transmitting DMX-512 data, Art-Net can also be used for transferring timecode information for equipment synchronisation.

Each CueCore2 supports sending and receiving of 2 Art-Net universes as well as Art-Net timecode.

2.3 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 CueCore2 supports sending and receiving of 2 sACN universes.

2.4 KiNet

KiNet is a proprietary protocol of Philips Color Kinetics to control their LED fixtures and power supplies. It is a lightweight Ethernet-based protocol that carries DMX-style data. Within the CueCore2 it can only be used to output data.

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 CueCore2 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, therefor it is faster than TCP but less reliable.

There are two ways how to have the CueCore2 respond to incoming UDP messages. The API (see page 91) makes typical CueCore2 functions available through UDP. Furthermore, custom messages can be programmed in the Show Control page (see page 50). This is also the place where to program outgoing UDP messages.

2.7 OSC

Open Sound Control[1] (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 MIDI and custom 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. TouchOSC from http://hexler.net/software/touchosc. There is a TouchOSC layout available from `http://www.visualproductions.nl/products/quadcore.html` that is configured to control the Playbacks of the CueCore2.

There are two ways how to have the CueCore2 respond to incoming OSC messages. Firstly, the API (see page 89) makes typical CueCore2 functions available through OSC. Secondly, custom messages can be programmed in the Show Control page (see page 50).

2.8 GPI

The CueCore2 features four General Purpose Inputs (GPI) ports that can be connected to external equipment, switches and sensors. State changes on these GPI ports can be used to trigger programmed events inside the CueCore2.

[1] www.opensoundcontrol.org

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

When set to analog the external equipment is supposed to supply a voltage between 0V and 10V to the port's pin. For convenience, a 10V supply is available on one of the pins of the GPI connector. Please refer to figure 2.2 for the pinout of the GPI connector. Be careful not to supply more than 10V to the GPI port as that might cause permanent damage.

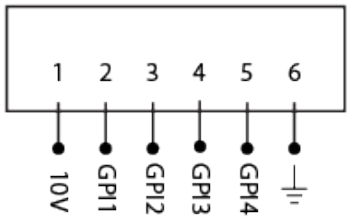


Figure 2.2: GPI Pinout

Please refer to page 69 for more information on configuring the GPI ports. Programming events based on GPI activity is done in the Show Control page, which is discussed on page 50.

2.9 MIDI

The MIDI protocol[2] is intended for inter-connecting musical devices such as synthesisers and sequencers. Furthermore, this protocol is also very suitable to send triggers from one device to another and is often used to synchronise audio, video and lighting equipment. There is also a large collection of MIDI control surfaces available; user-interface consoles with knobs, (motorised-)faders, rotary-encoders, etc.

The CueCore2 is fitted with a MIDI input and MIDI output port. It supports receiving and sending MIDI messages like NoteOn, NoteOff, ControlChange and ProgramChange.

2.9.1 MTC

MIDI Timecode (MTC) is the timecode signal which is embedded into MIDI. The CueCore2 supports receiving and transmitting MTC. It is not recommended

[2] www.midi.org

to combine the use of MTC with ordinary MIDI as MTC consumes the bandwidth of the MIDI connection.

2.9.2 MMC

MIDI Machine Control (MMC) is part of the MIDI protocol. It defines special messages for controlling audio equipment such as multi-track recorders. The CueCore2 supports the sending of MMC commands; please refer to page 76.

2.9.3 MSC

MIDI Show Control (MSC) is an extension of the MIDI protocol. It comprises of commands for synchronising show equipment like lighting, video and audio devices. The CueCore2 supports receiving MSC commands. This support is hard coded and does not require any Show Control programming.

2.10 SMPTE

SMPTE is timecode signal which can be used to synchronise audio, video, lighting and other show equipment. The CueCore2 supports receiving SMPTE that is transferred as an audio signal, also know as LTC timecode.

2.11 NTP

Network Time Protocol (NTP) is a networking protocol for clock synchronisation between computer systems over networks.

The real-time clock (RTC) in the CueCore2 can be synchronised to an external time server using the NTP protocol.

2.12 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 CueCore2 is a DHCP client.

Chapter 3

Quickstart

This chapter provides step by step tutorials on how to program your CueCore2 for some typical tasks:

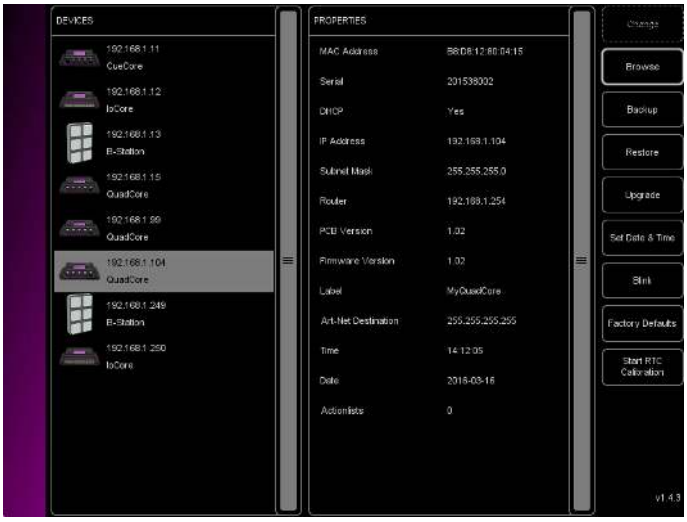
- Playback lighting scenes based on the scheduler
- Choose between different lighting scenes via incoming UDP messages
- Record a show from an external DMX console

3.1 Playback based on scheduler

This tutorial shows how to create a lighting scene and have it activated at a certain time of the day. The scene will be de-activated at another time. Follow the steps below:

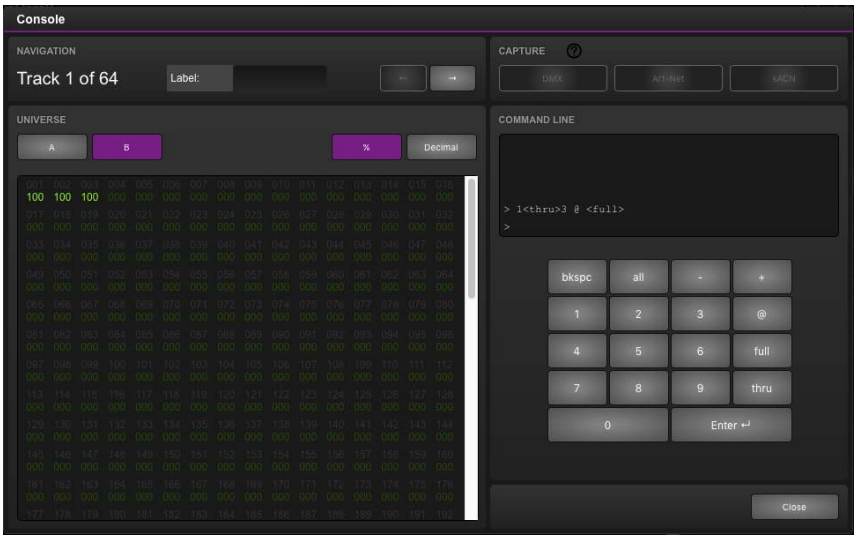
1. Connect to the network
   Connect the CueCore2 with an Ethernet cable to the router. It is required that the network is managed by a router that features a DHCP server. If the network router is not DHCP capable then read the network chapter on page 28 for alternative setups.

2. Install the vManager
   To access the web-interface of the CueCore2, the vManager tool is required. This tool can be downloaded from the Visual Productions website. Once the installation is complete, run the vManager to discover the IP address of the CueCore2.

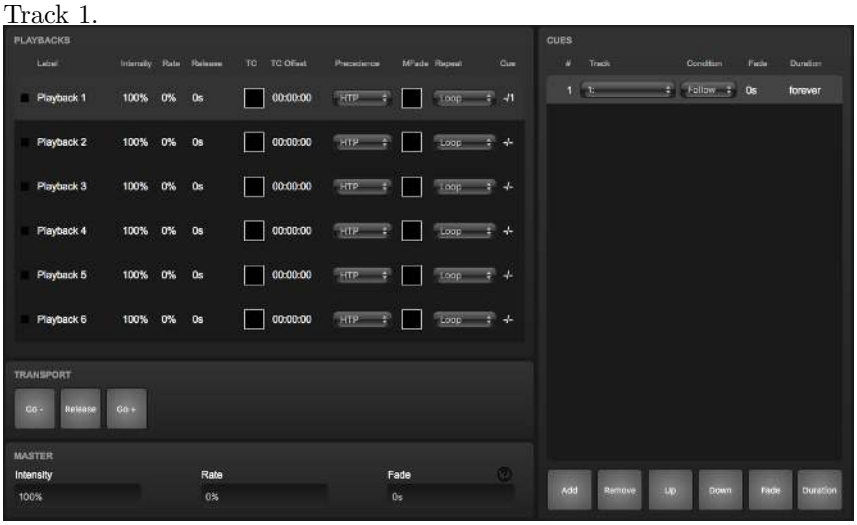


3. Open the web-interface
   Choose the CueCore2 from the device list and click on the *Browse* button to open the web-interface.

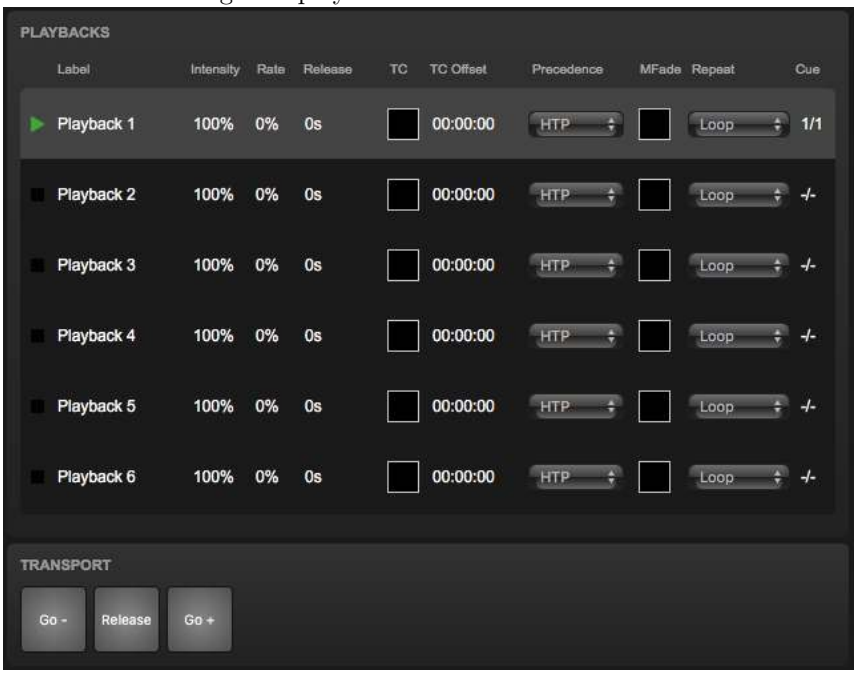
4. Create the scene
   Use the browser to go to the CueCore2's 'Track' page. Select a track from the table and press the 'Open Console' button. Create a scene by using the command-line syntax. E.g. `1<thru>3 @ <full>`



5. Create a cue
   Go to the Playback page and select Playback 1. Press the Add button to create a new cue. Once the Cue is added it will automatically refer to Track 1.



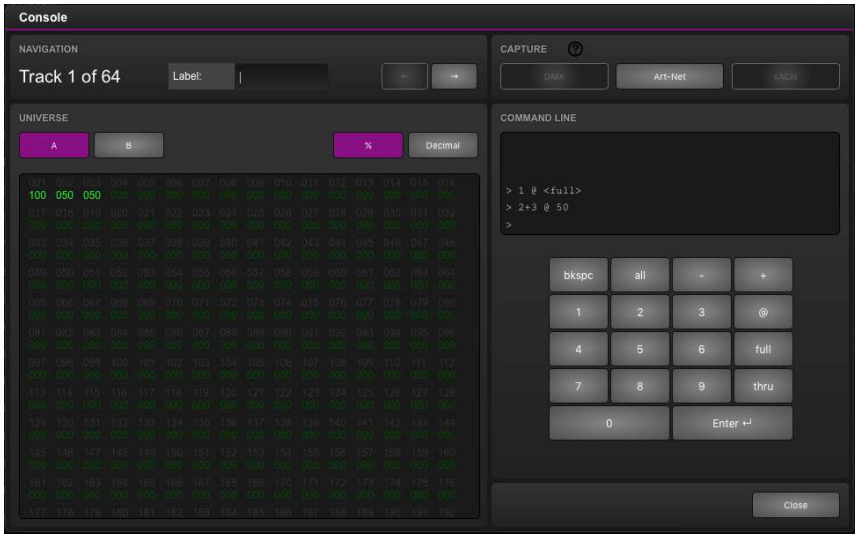
6. Start playback
   Press Go+ on the transport area to start the Playback. The playback now indicated the green 'play' icon.



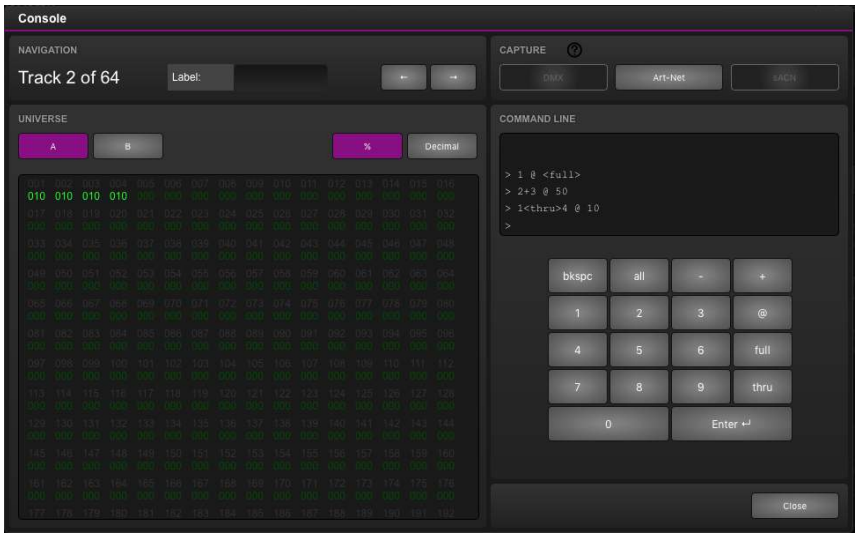
3.2 Choose scenes via UDP

This example will create two lighting scenes. They will be put into a single playback. This means only one scene will be active at a time. Furthermore, a cross-fade will be defined between the scenes and the scenes will be triggered by receiving simple UDP network messages. Please take the following steps:

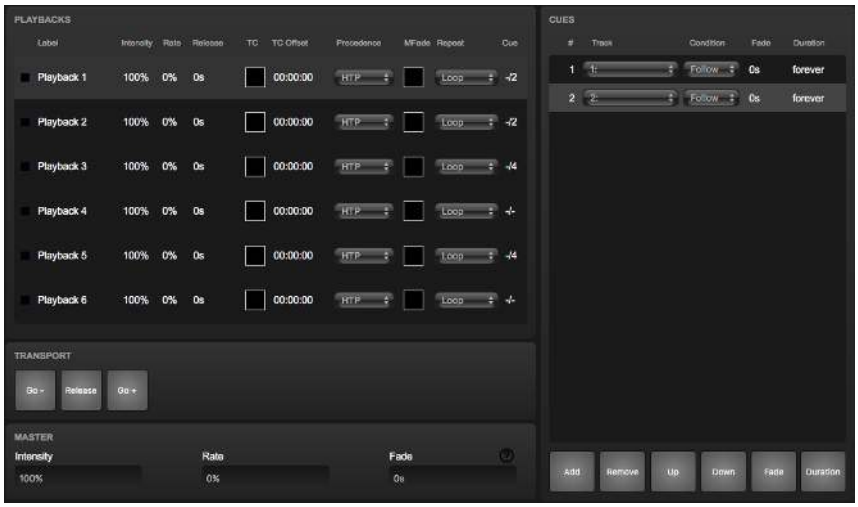
1. Create the first scene
   Use the browser to go to the CueCore2's 'Track' page. Select a track from the table and press the 'Open Console' button. Create a scene by using the command-line syntax. E.g. `1 @ <full>` or `2+3 @ 50 <enter>`



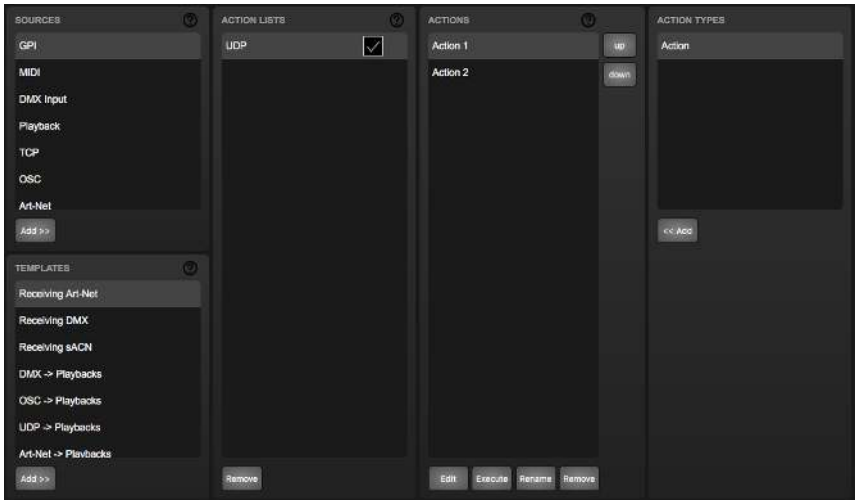
2. Create the second scene
   Press the 'right arrow' button to switch to the next track. Again make a scene by using some command-line syntax; e.g. `1 THRU 4 @ 10 ENTER`



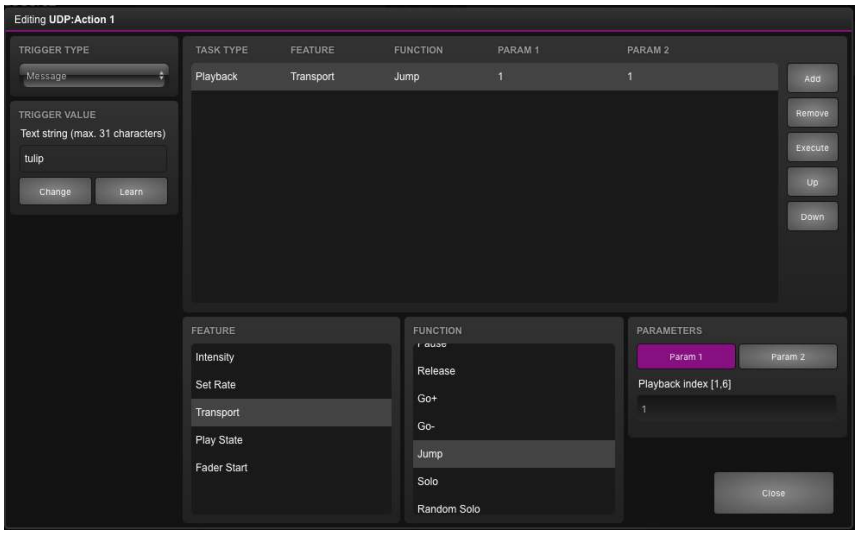
3. Program the playback
   Go to the 'Playback' page, select the first of the six playback and insert two cues by pressing the 'add' button. Set cue #1 to refer to your first track and cue #2 to refer to your second track.



4. Create an action list
   Go to the 'Show control' page. Select 'UDP' from the 'Sources' table. Copy UDP to the 'Action list' table by using the 'Add >>' button. Select the new UDP action list and insert two actions by pressing the '<< Add' button twice.

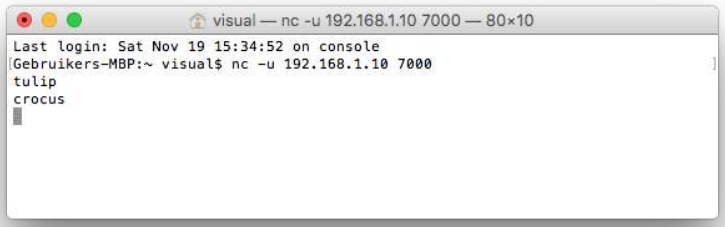


5. Create actions
   Select the first action and press 'Edit' to open the dialog. Change the trigger value to "tulip". Add one task by using the 'Add' button. Choose 'Playback' from the list of task types. Select the newly added task and set the 'feature' to 'Transport' and set the 'function' to 'Jump. Parameter 1 should be set to '1' (addressing the first playback) and parameter 2 should be set to '1' (jump to the first cue).

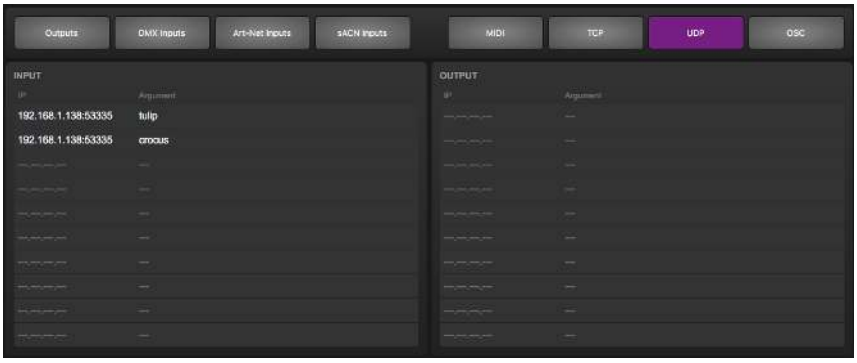


   Press the 'Close' button, select the second action and press 'Edit' again. Change this trigger value to "crocus". Add a task by pressing 'Add' and choose the 'Playback' task-type. Select the newly added task and set the 'feature' to 'Transport' and set the 'function' to 'Jump. Parameter 1 should be set to '1' (addressing the first playback) and parameter 2 should be set to '2' (jump to the second cue).

6. Test with netcat and monitor
   On your computer, use a simple command-line tool like netcat to send a UDP string to the CueCore2. On Mac OSX netcat is started with the command `nc -u 192.168.1.10 7000` (replace 192.168.1.10 with IP address of your CueCore2). From now on you can type `tulip` <enter> or `crocus` <enter> to send this messages to the CueCore2.



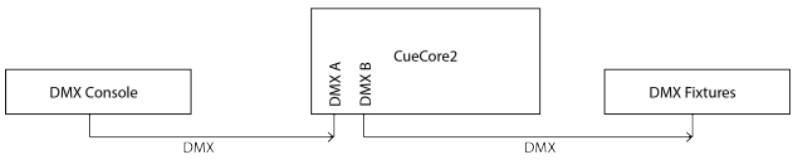
   Go to the 'Monitor' page in your browser and select 'UDP In' to verify your device is receiving the UDP messages correctly. On the 'Playback' page you should see playback #1 respond to the incoming UDP commands by activating either cue #1 or cue #2.



3.3 Record a show from an external DMX

The CueCore2 is capable of recording DMX data. This tutorial explains the required procedure.

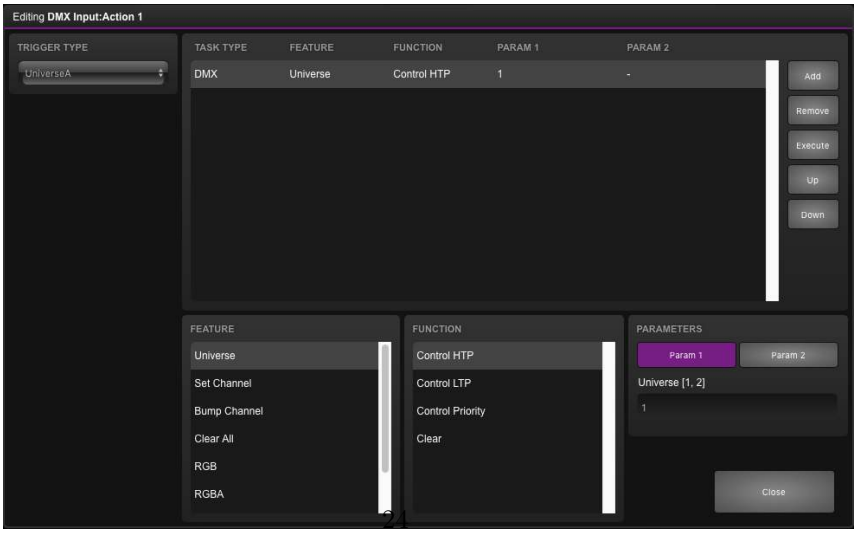
1. Connect the external console
   Connect the DMX output of the DMX console to *Port A* of the CueCore2. Connect the fixtures to *Port B*.



2. Configure port settings
   Go to the *Settings* page and set DMX *Port A* to *In*. Set *Port B* to *Universe A*, it will now transmit DMX channels 1-512.

3. Throughput the DMX
   The DMX received by the CueCore2 will not automatically be output to the fixtures, however, it is desirable to see the console's output on the actual fixtures. To achieve throughput of the DMX, go to the *Show Control* page. Create a *DMX Input* action list and insert one action.

Edit the action. Set the *Trigger Type* to *UniverseA*. Add a *DMX* task and set its feature to *Universe* and its function to *Control HTP*, the first parameter should be set to 1.



4. Configure the recording
   Go to the *Track* page. Select the first track and press the *Erase* button. Wait until the erase process is completed. Set *Mode* to *Manual*. Set *Source* to *DMX* and set *Sample rate* to 40 FPS.

5. Record
   Press the *Record* button at the begin of the console's show. Press the *Stop* button when the show is finished.

6. Test the result
   Make sure the console outputs only zero values. Then playback the track's content by enabling the *Track Preview* checkbox.

Chapter 4

Setting up

This chapter discusses how to set up the CueCore2.

4.1 Mounting

The device can be placed desktop or it can be DIN Rail mounted. The device is prepared for DIN Rail mounting by using the 'DIN rail holder TSH 35' from Bopla (Product no. 22035000).

Figure 4.1: Bopla DIN rail adapter

This adapter is - amongst others - available from:

- Farnell / Newark (order code 4189991)
- Conrad (order code 539775 - 89)
- Distrelec (order code 300060)

4.2 Kensington Lock

The device can be secured by using a Kensington style laptop lock.

Figure 4.2: Kensington lock

4.3 Power

The CueCore2 requires a DC power supply between 9 and 24 Volt with a minimum of 500mA. The 2,1 mm DC connector is center-positive. The CueCore2 is also Power-over-Ethernet (PoE) enabled. It requires PoE Class I.

Figure 4.3: DC polarity

Chapter 5

Network

The CueCore2 is a network capable device. A network connection between between a computer and the unit is required to configure and program the CueCore2, however, once the device is programmed then it is not necessary anymore for the CueCore2 to be connected to an Ethernet network.

There are multiple arrangements possible for connecting the computer and the CueCore2. They can be connected peer-to-peer, via a network switch or via Wi-Fi. Figure 5.1 illustrates these different arrangements.

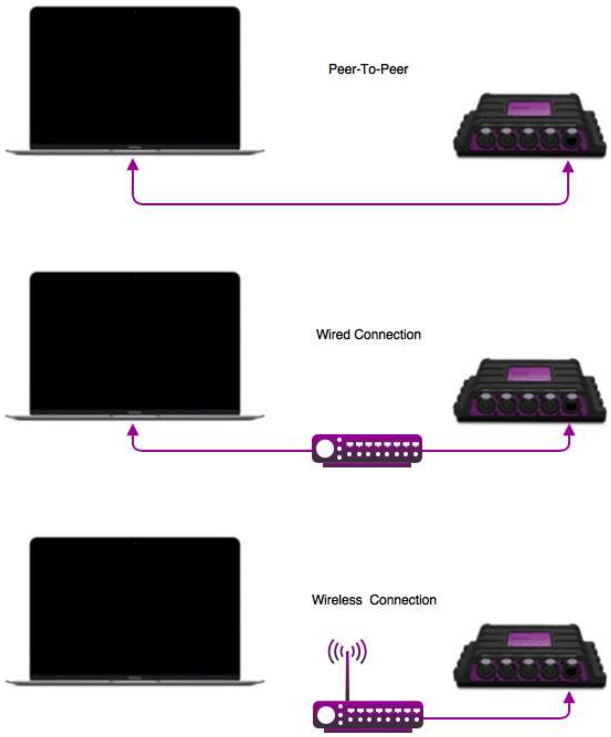


Figure 5.1: Network arrangements

The Ethernet port on the CueCore2 is auto-sensing; it does not matter whether a cross or straight network-cable is being used.

5.1 IP Address

The CueCore2 supports both static IP addresses and automatic IP addresses. By default, the CueCore2 is set DHCP in which it will be automatically assigned an IP address by the DHCP server in the network. The 'DHCP server' is typically part of the router's functionality.

Static IP addresses are useful when there is no DHCP server in the network, for instance when there is a direct peer-to-peer connection between a CueCore2 and a computer. It is also useful in permanent installations where the IP address of the CueCore2 is known by other equipment and therefor should not change. When using DHCP there is always the risk of automatically being given a new IP address in the event that the DHCP server is replaced. When using static IP addresses make sure that all equipment on the network have unique IP addresses.

The CueCore2's LED helps to determine which kind of IP address is set. The LED will indicate red when using DHCP and it will indicate white in the case of a static IP address.

There are three ways to change the IP address setting of the CueCore2.

- **vManager** can be used to detect a CueCore2 on the network. Once found, the vManager software (figure 5.3) 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 CueCore2's **web-interface**. The Settings page on this web-interface enables changing the IP address, subnet mask and DHCP settings.
- By briefly pressing the **reset button** on the device it toggles between static and automatic IP addresses. By pressing and holding the reset button (see figure 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 Access via Internet

The CueCore2 can be accessed through the Internet. There are two ways to achieve this: Port-Forwarding and VPN.

- **Port-Forwarding** Is relatively easy to setup in the router. Each router is different so it is advised to consult the router's documentation (sometimes it is revered to as NAT or Port-Redirecting). Please note that port

Figure 5.2: Reset button

forwarding is not secure, since anybody could access the CueCore2 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 CueCore2. 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`.

5.3 vManager Software Tool

A free-of-charge software tool called vManager has been developed to manage the devices. This tool is available on Microsoft Windows, macOS and Ubuntu Linux via the Visual Productions website. vManager allows for:

- Setup the IP address, subnet mask, router and DHCP
- Backup and restore the device's internal data and settings
- Perform firmware updates
- Set the real-time clock of the CueCore2 (The computer's date and time will be used)
- Identify a specific device (in a multi device set-up) by blinking its LED
- Revert to factory defaults

The following section explain the buttons in the vManger, as seen in figure 5.3.

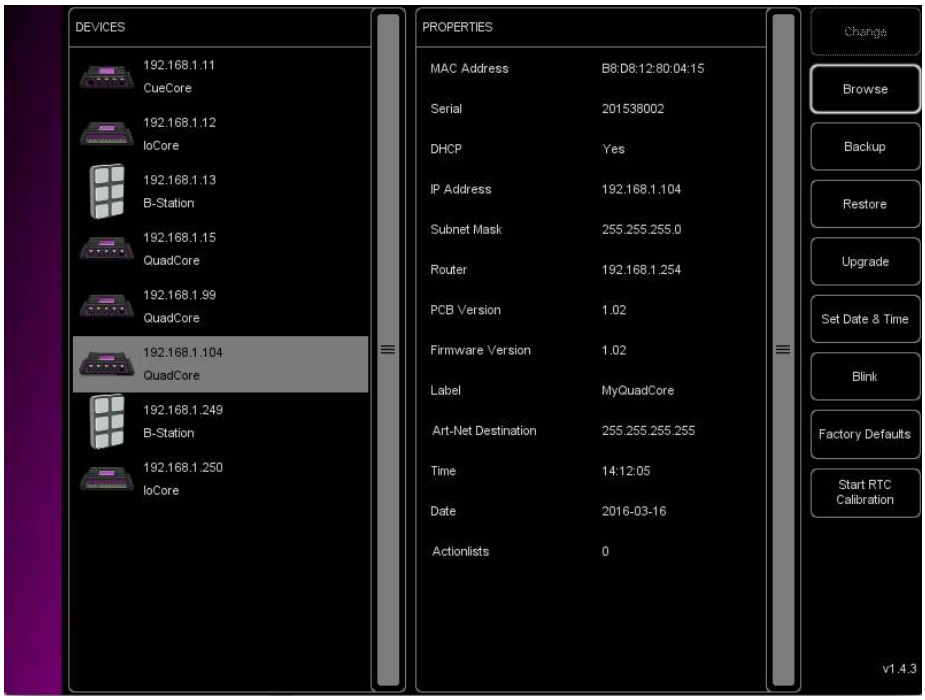


Figure 5.3: vManager

5.3.1 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.

The backup files created by vManager can be found at the following locations:

Microsoft Windows	\Users\[username]\Documents\Visual Productions\Common\Backups
MacOS	/Users/[username]/Visual Productions/Common/Backups
Ubuntu Linux	/home/[username]/Visual Productions/Common/Backups

5.3.2 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.

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

5.3.3 Set Date & Time

The computer's date and time can be quickly copied to the CueCore2 by selecting a device and clicking the Set Date & Time button.

5.3.4 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.

5.3.5 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 reverted to their defaults by pressing the Factory Defaults button. This action does not affect the device's IP settings.

Chapter 6

Operating Modes

A CueCore2 can operate in three modes, each mode resulting in a different behaviour of the device.

- Stand-alone
- Slave
- CueluxPro

By default the CueCore2 operates in the Stand-alone mode.

6.1 Stand-alone mode

In this mode the CueCore2 is an autonomous device for controlling lighting. Typically it is loaded with lighting content and programmed to respond to external triggers and/or internal scheduling. This is the default behaviour of a CueCore2; the stand-alone mode is active whenever the CueCore2 is not in the slave or CueluxPro mode.

6.2 Slave Mode

Some demanding lighting designs can require more than two universes of DMX. When multiple CueCore2 units are combined to create a large multi-universe system there is the need for synchronisation of those CueCore2 devices. The Slave mode facilitates this. See figure 6.1.

When in Slave mode the CueCore2 is taken over by a master-CueCore2 and is no longer responsible for its playbacks and scheduling; the master takes care of this. All the slave requires is to contain the lighting content in its tracks. The master-CueCore2 will control all its slaves to activate the same tracks and keep the playback of those tracks synchronised.

It is necessary to put all action-programming in the master-CueCore2. In fact, the playback information inside the slaves will be overwritten by the master. The master does this because it stores a copy of its playback-data in each slave to

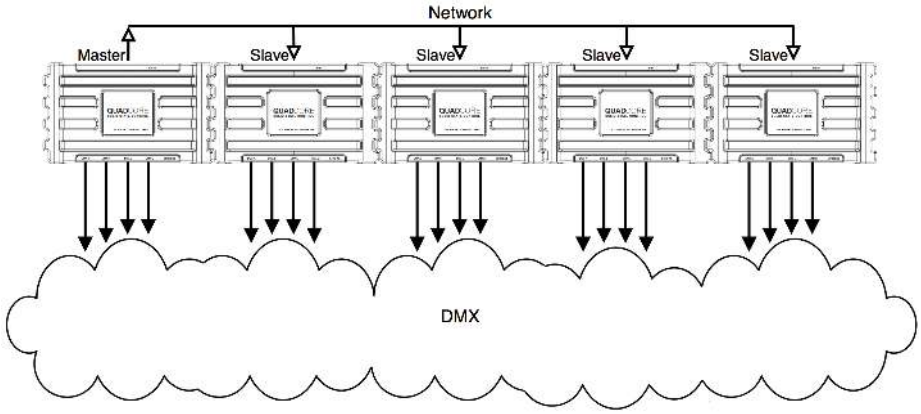


Figure 6.1: Master/Slave setup

enable the slave to continue autonomously in case the communication between master and slave is interrupted.

The logical place for the action lists and action for a master/slave system is also inside the master, however, it is allowed to place actions in a slave and they will get executed.

The Slave mode is enabled in the Settings page (See chapter **??**, page 65). Once enabled, the Slave mode is entered as soon as the master connects to the slave. The Slave mode reverts back to the Stand-alone mode when the master disconnects or when the slave disables Master/Slave in the Settings page.

6.3 CueluxPro Mode

CueluxPro (see figure 6.2) is a software-based lighting console that is bundled with the CueCore2. The purpose of the CueCore2 in this mode is to be an interface between CueluxPro and the DMX lighting fixtures. Therefore the CueCore2 will forward the data received from the CueluxPro software to its DMX outlets. During this mode all internal playback and scheduling within the CueCore2 is suspended. Figure 6.3 illustrates a typical CueluxPro/CueCore2 system.
The CueCore2 enters the CueluxPro mode as soon as it is patched to one or more universes within the CueluxPro software. This mode is exited by un-patching the CueCore2 or closing down the CueluxPro software.
Using the CueluxPro software in combination with the CueCore2 results in a lighting control system with a larger feature set than using the CueCore2 on its own in the stand-alone mode. CueluxPro features:

- Personality library with 3000+ fixtures
- FX Generator
- Matrix Pixel-mapping

Figure 6.2: CueluxPro

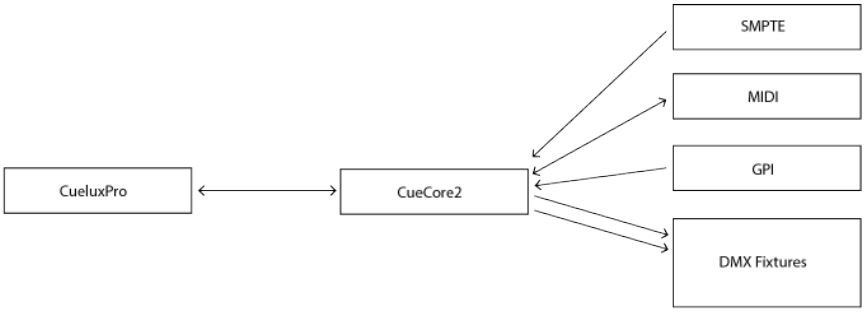


Figure 6.3: A typical CueluxPro system

- Groups
- Palettes
- Timeline editor

CueluxPro can also be used for generating the lighting content that can be uploaded to the CueCore2. After uploading, the CueCore2 can continue to be used stand-alone. For information on how to use CueluxPro please refer to the CueluxPro manual on the Visual Productions website. This manual provides instructions for connecting to CueluxPro and uploading content to the CueCore2.

Chapter 7

Tracks

A Track is a piece of lighting content that can be activated by a playbacks. Tracks can contain dynamic lighting effects; each track can be a 'DMX recording' with a certain duration. Of course a static scene can also be stored in a track.

There are three different ways to put the content inside the track. The 'Console' page allows the user to create and edit a static scene directly via the web-interface. This page also is capable of recording a static scene from an external DMX, Art-Net or sACN source. The Console page is discussed in detail on page 37.

The second way for storing content into the tracks is done via the 'Recorder' section; this section of the Tracks page contains control for recording dynamic DMX content from external DMX, Art-Net and sACN sources.

Furthermore, it is also possible to create the lighting content using the CueluxPro software and upload it to the CueCore2. This can be dynamic as well as static content. For more information on CueluxPro see chapter 6, page 33.

7.1 Number of Tracks

The CueCore2 has a fixed memory chip onboard. This memory chip is divided into a number of equally sized slots called 'Tracks'. Go to the Settings page to choose the amount of slots the memory chip is divided into. The CueCore2 offers a choice of 1, 2, 4, 8, 16, 32, 64 or 128 tracks. More tracks will result in a smaller memory size per track.

Once the number of tracks has been set, the content of the tracks must be erased. It is recommended to choose how many tracks will be used before filling them with content.

Warning: Changing the number of tracks will result in loosing the current content of the tracks.

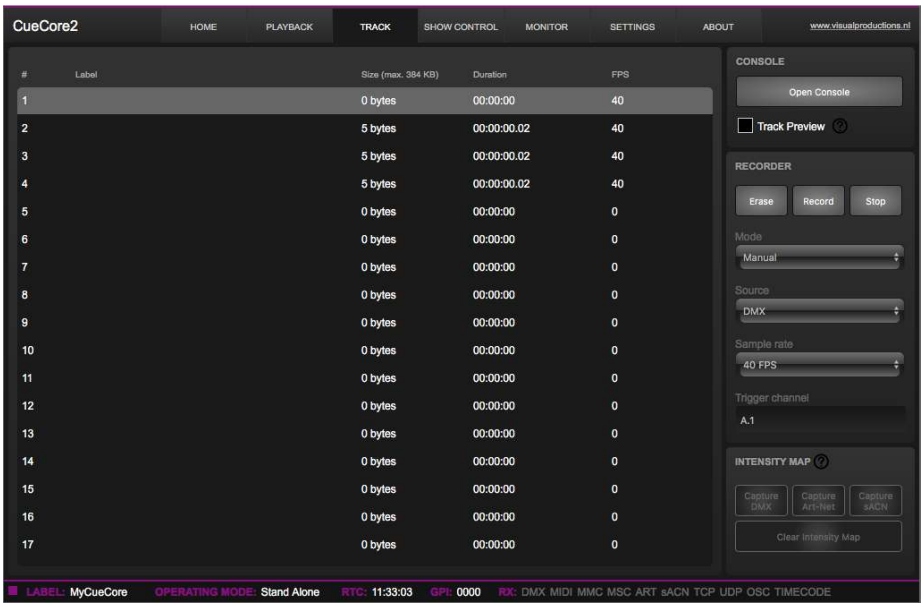


Figure 7.1: Tracks

7.2 Track Properties

The Track listing (See figure 7.1) displays the following track properties:

- Label: The name of the track; this field can be changed by double-clicking.
- Size: The number of bytes used by the data inside the track. The maximum size is indicated at the top of this column. This maximum depends on the 'number of tracks' selected in the Settings page.
- Duration: The length of the track displayed in hours:minutes:seconds.milliseconds.
- FPS: The sample rate of the track displayed in Frames Per Second (FPS). The sample rate has been chosen during the recording process and cannot be altered afterwards.

7.3 Console

The Console page (see figure 7.2) allows to edit a track directly through the web-interface, however, a track does need to be a static scene; it should only contain a single DMX frame. If the track already contains more than one DMX frames and thus it is a dynamic track, then it can be made static by erasing it. The track can be edited by selecting the track in the table and then pressing the 'Open Console' button. This will automatically enable the 'Track Preview' checkbox so the content that is being edited in the Console page is also outputted live.

The 'Track Preview' is a useful option to briefly test the content stored in a

track without having to configure a playback for it. Please note that any active playback will be released when the Track Preview is enabled.

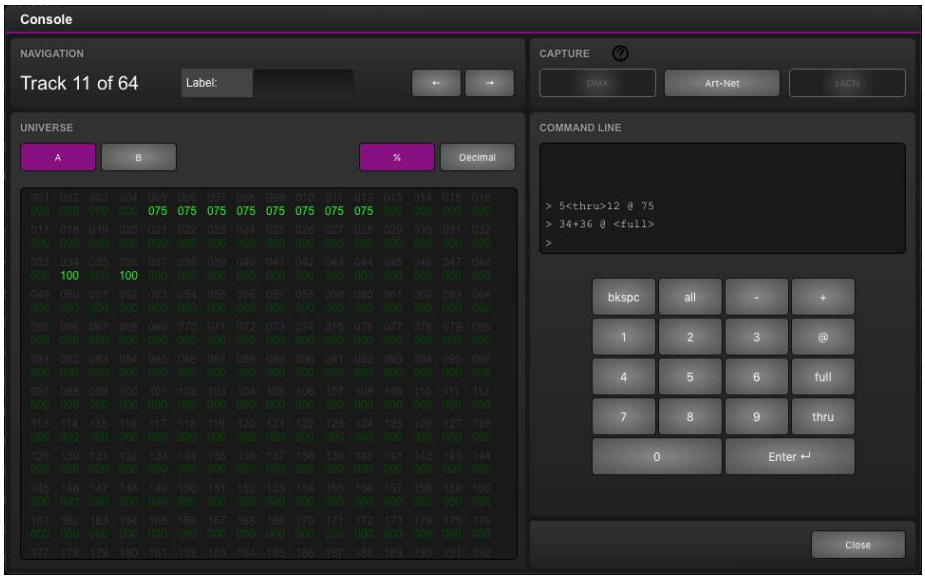


Figure 7.2: Console page

Inside the Console page the DMX values of the track can be changed by using the Command-line interface. The following table offers examples of the supported commands.

Command	Function
`1 @ 50 ENTER`	Sets channel 1 at 50%
`1 + 2 @ FULL`	Sets channel 1 and 2 at 100%
`1 THRU 3 @ 0`	Sets channels 1 through 3 at 0%
`1 THRU 3 + 5 @ 0 ENTER`	Sets channels 1, 2, 3, and 5 at 0%
`ALL @ 100 ENTER`	Sets all channels in the selected universe at 100%
`1 @ + 10 ENTER`	Increases channel 1 value with 10%
`ALL @ - 20 ENTER`	Decreases all channels in the selected universe by 20%

By default the Console page presents the DMX values in percentage (%). When the representation is switched to decimal (by using the 'Decimal' button) then the values in the table above would be interpreted as decimal values as well. E.g. `1 @ 50 ENTER` would set the channel at decimal value 50 which relates to 20%.

Instead of setting the values manually, the Console page also offers to record

the entire scene from an external DMX, Art-Net or sACN source. The buttons in the Capture section become available when the CueCore2 is receiving the signal of the corresponding protocol. I.e. that the 'DMX' button is disabled unless the unit is receiving actual DMX. Please be aware that - once enabled - pressing one of the capture buttons will overwrite the current channel levels in all universes.

7.4 Recorder

The Recorder section is used to capture dynamic content from an external source and store it inside a track. In order to be stored in flash memory, a track requires to be erased first. It is advised to manually erase the track before starting a record. This is done by selecting it in the table and then pressing the 'Erase' button. In case a non-erased track will be directly recorded then the CueCore2 will automatically first erase the track, however, this gives less control over the timing of the start of the recording, especially in the Manual mode.

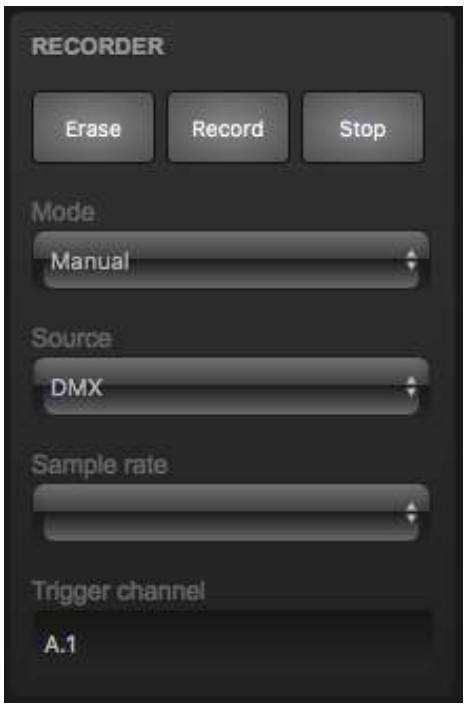


Figure 7.3: Recorder section

The icons in the track table visualise the different states of the recorder. The 'trash icon' indicates a track is being erased. The 'orange dot' signifies a track being ready to start recording, this corresponds to the Triggered or Timecode mode. A 'red dot' indicates a recording in progress.

7.4.1 Mode

The triggering modes define how the recorder is initiated. There are three different modes.

- The most simple mode is **Manual**. In this mode the user has to manually press the 'Record' button to start and press the 'Stop' button to end the recording. Although simple to operate, this mode does not give accurate control over the timing of the begin and end of the recording. Both human interaction and operation through a web-based user-interface will introduce some degree of lag.
- An automated way of starting and stopping the recording process is done in the **Triggered** mode. One of the data channels is allocated to control the start/stop. The channel address is denoted by the 'Trigger Channel' field. It is advised to include this channel in the show programming done on the external source; a typical lighting console allows accurate timing of DMX channels which gives fine control over when the recording starts and ends in relationship to the show content. When using the Triggered mode pressing the 'Record' button will prepare the track for recording; it will be erased when necessary and then stay idle in anticipation of the trigger channel going high to indicate 'start'. The recording is ended by setting the trigger channel to 0%.
- The **Timecode** mode allows for the recording process to be synchronised by incoming timecode. Pressing the 'Record' button will prepare the track for recording; it will be erased when necessary and then stay idle in anticipation of the timecode to start running, it stops when the timecode resets back to 00:00:00.00. Always record from frame 00:00:00.00. If the content is supposed to run at a different frame then use the playback's 'TC offset' property to achieve that.

A typical challenge with recording dynamic DMX data is to create a seamless loop. Often the manual mode will most likely not be sufficiently accurate to achieve a seamless loop. The triggered mode offers a way to remote control and make the recording seamless. Alternatively, the lighting content can be designed in CueluxPro instead of recording from an external source, as CueluxPro automatically takes care of making its content seamless.

7.4.2 Sources

The CueCore2 is capable of recording DMX data from an external source by using three different protocols:

- DMX
- Art-Net
- sACN

Please consider that the operation of these protocols depend on their properties the Settings page.

7.4.3 Sample Rate

The Sample Rate setting will determine how many samples of the data are taken per second and stored in memory. This setting variants are 5, 10, 30 and 40 Frames Per Second (FPS). 40 FPS gives maximum quality in terms of smooth dimming curves. 5 FPS is a low value but useful for slow DMX changes and consumes much less memory. The 40 FPS setting is recommended unless there is a reason to reduce the sample rate.

7.5 Track Capacity

The CueCore2 has 32MB memory, of which 24MB is reserved for the tracks. The device uses a compression algorithm to store the data and optimise the storage for best use. The duration of the recording that the track can hold depends on several parameters: number of tracks, dynamic lighting content and the number of DMX channels used. Therefor the maximum duration is hard to specify, however, some guidance can be provided:

In the most likely example the memory will hold 2m07s with a setup of 128 tracks. For a setup of 1 track the memory will hold up to 4h36m07s.

In the worst-case scenario in which 2,048 channels changing in every frame at 40 FPS the memory will hold 3s per track in a 128 tracks setup. In the setup of 1 track the memory will hold 12m08s.

If the limits of the capacity are reached then there are three different ways to help overcome this.

- Reduce the 'number of tracks' in the settings page. Note that the current track content is lost when changing this setting.
- Reduce the sample rate.
- Spread the content over multiple tracks. They can be linked together later on the Playback page (For more information go to chapter Playbacks, page 20). This way cross-fades can be generated by the CueCore2 instead of being recorded.

7.6 Intensity map

Typically, a DMX recorder stores the values of the channels without knowing its functions. When reducing the output level at the Playback all channels are reduced, also the ones that contain information other than intensity/dimmer levels. This has the undesired effect that RGB or Pan Tilt channels are also affected, whereas ideally only the intensity levels should be lowered. This is a challenge all DMX recorders have. The intensity map (figure 7.4) overcomes this issue by specifying to the CueCore2 which channels control intensity.

For example, when there is a track with the recording of a moving head that includes pan, tilt and intensity channels. Changing the Intensity of the track would have changed the values of pan and tilt as well. To change only the value of the intensity channel without affecting the pan and tilt, the intensity map

Figure 7.4: Intensity Map section

will specify which channel values can be altered and which can not.

To set up the Intensity Map follow steps below:

1. Connect an external lighting console via DMX, Art-Net or sACN.
2. Create a scene in which intensity channels are set to 100%. In case of 16-bit dimming, the coarse (or MSB) channels are set to 100% and the Fine (or LSB) channels are set to 50%. All other channels go to 0%.
3. Press the 'Capture' button.

The recording of this lighting scene is now saved in the Intensity Map.

The capture buttons remain disabled while the CueCore2 is not receiving the actual signal from the corresponding protocol. The 'Clear Intensity Map' button is only enabled when there is an intensity map present; a disabled 'Clear Intensity Map' button is an indication that there is no map stored in the memory.

Chapter 8

Playbacks

A playback is capable of activating the lighting content stored in the tracks. Tracks are merely storage for lighting scenes and effects; the playbacks actually plays them. The playbacks are located in the Playback page in the web-interface, see figure 8.1.

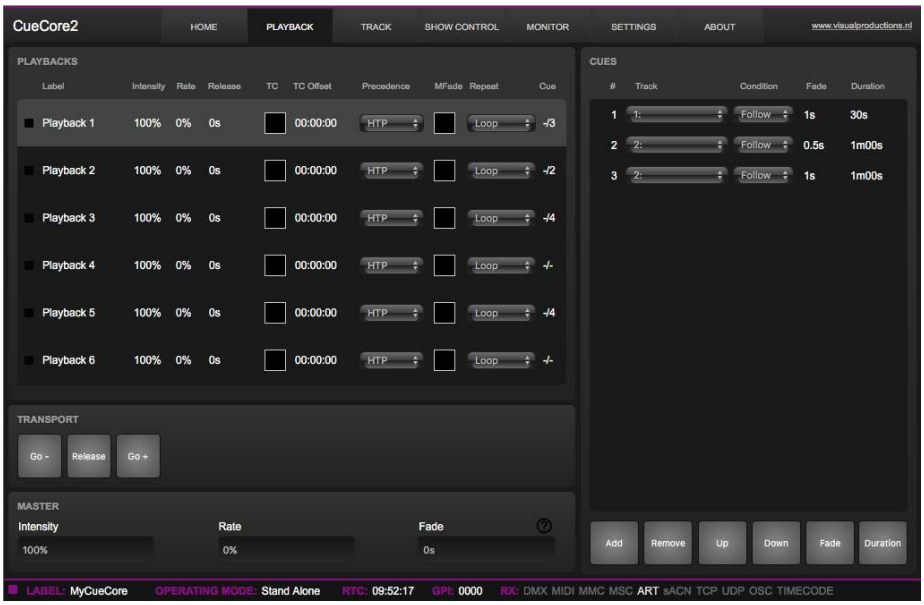


Figure 8.1: Playback page

There are 6 playbacks available. Each can contain up to 32 steps, called 'cues'. A cue will contain a reference to a track plus additional information such as fade-time and duration. Figure 8.2 illustrates the structure of a playback.

Playbacks can be run independently of each other; they can all start or stop at different times. It is possible to control the same DMX channels from multiple playbacks and have them merged together. Also, it is possible to have each playback control a different set of DMX channels; making each playback responsible

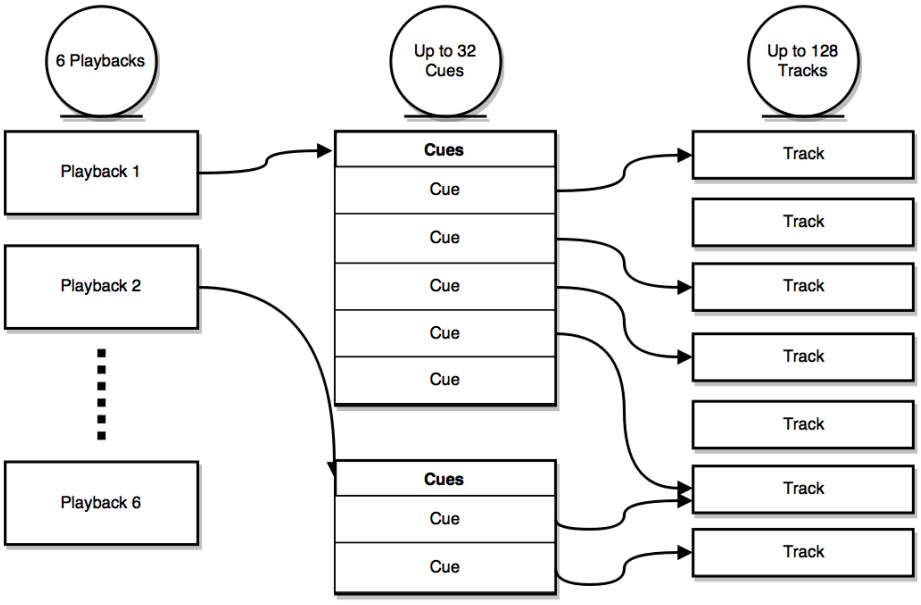


Figure 8.2: Playback structure

for a different zone. Figure 8.3 shows an example of controlling multiple zones in a hypothetical restaurant.

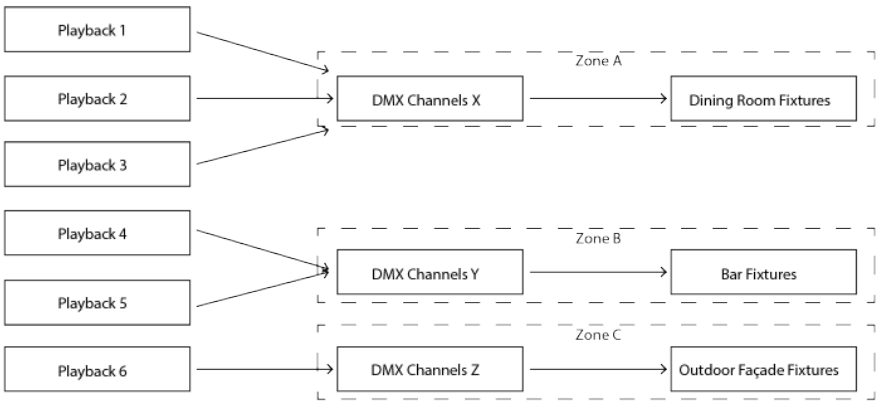


Figure 8.3: Playbacks controlling zones in a restaurant

8.1 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 8.4.

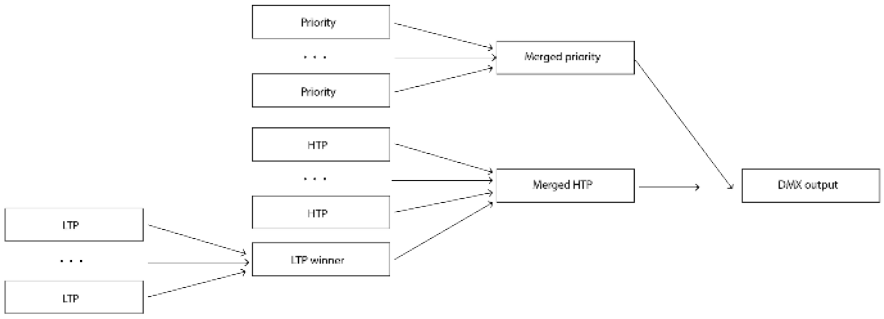


Figure 8.4: Playback precedence

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.

8.2 Playback Properties

Each playback provides a set of properties that can be used to customise the playback's behaviour. Some properties are changed by double-click.

Label	The name of the playback.
Intensity	The output level of the playback.
Rate	The speed of the playback. By default, it is set to 0%. It can go up to 100% (faster) and down to -100% (slower).
Release	When released the playback can fade out to zero. This release time defines how long this fade out will take. Setting it to 0s will result in an instant release.
TC	When enabled, the playback is synchronised to the current timecode (TC). By default, TC is disabled. Note that the Settings page provides a field for selecting the timecode protocol, e.g. 'internal' or 'Art-Net'.
TC Offset	Specifies at which timecode frame the playback starts.
Precedence	Determines how the output of the playbacks is merged together, as explained on page 44.
MFade	Normally the fade time between cues is determined by the 'fade' field in the cue properties. When Mfade is enabled then the playback will ignore the cue's fade times and use the master fade time for all its cues.
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.
Cue	Current/Total of Cues. Indicates which cue is currently active and indicates the total number of cues in the Playback.

The intensity and rate properties are not stored in the CueCore2's internal flash memory. It is expected that these properties can change often during the operation of the CueCore2 and could consequently wear out the flash memory.

A consequence of not storing these properties is that after a power cycle their levels will be reverted to the default values. If the intensity or rate requires to be permanently set to a value other than the default value then it is recommended to use the Show Control page and create an action in the 'System' action list. This action can have its trigger set to 'Startup' and contain tasks to set the playback's intensity and rate to the desired values.

8.3 Cue

A cue is a step inside a playback. A playback can contain up to 32 cues. A cue does not contain a lighting scene, rather, it refers to a track which do contain the lighting scenes. It is possible for multiple cues to refer to one track. The cue does contain information on how long the lighting scene should be played

and if it should be cross-faded from the previous cue.

Figure 8.5: Cues

Each cue provides the following properties:

Track	A reference to the track that will be played in this step.
Condition	Determines what the playback does after the cue is finished with the cross-fade and the duration. In case of 'halt' it will pause the playback, basically waiting for a Go+ command. In case of 'follow' the playback will automatically continue to the next cue; this condition is useful for creating chases.
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.
Duration	Determines how long the cue will be active. This is the time between the completion of the cross-fade into this cue and and the start of the cross-fade to the next cue. When the condition is set to halt then the duration has no effect; the cue will stay active until a Go+, Go-, Jump or Release command is given. The duration field accept not only 'time' input such as ".5" "30s" or "1m15", it also accepts 'number of cycles'; the playback can run the cue "1x" or "10x". This is particularly useful when the track referred to by the cue contains a (seamless-)looped effect. Please note that if the track contains a static scene; i.e. the track only holds a single DMX frame, then running it for a number of cycles will create a very short cue as a single DMX frame only takes 25ms. The third option for the duration field is to input "forever". In this case the cue will continue to run indefinitely; it requires a Go+ to traverse to the next cue. When set to "forever" the condition of the next cue becomes irrelevant.

The Playback page provides the following buttons to edit the cues:

- Add: Will add a new empty cue.
- Remove: Will remove the selected cue
- Up: Will move the selected cue up a position.
- Down: Will move a selected cue down a position.
- Fade: Will open a pop-up window where you can set the fade time.
- Duration: Will open a pop-up window where you can set the duration.

8.4 Transport

The transport section offers buttons to control the playbacks.

Go+	Jump to the next cue.
Go-	Jump to the previous cue.
Release	Deactivates the selected playback. Press and hold to release all playbacks.

8.5 Master

The master section provides features that are applied to all playbacks.

Intensity	The master intensity works like a theatrical 'grand master'; it dims the output of all playbacks taking their individual intensity setting into account.
Rate	The master rate will control the play speed of all playbacks; with taking their individual rate settings into account.
Fade	The master fade time overrides the fade time of all cues. This only applies to playbacks that have 'MFade' enabled.

Similar to some of the playback properties, the master properties are not stored in the internal flash memory. Please refer to the discussion on page 46.

Chapter 9

Show Control

The CueCore2 can interact with the outside world; it can receive messages and values through various protocols and it can send out many protocols. It is possible to automate the CueCore2 by having it respond automatically to incoming signals. An example of this would be to start a playback upon receiving a specific UDP network message. The Show Control page (See figure 9.1) enables this kind of programming to be made.

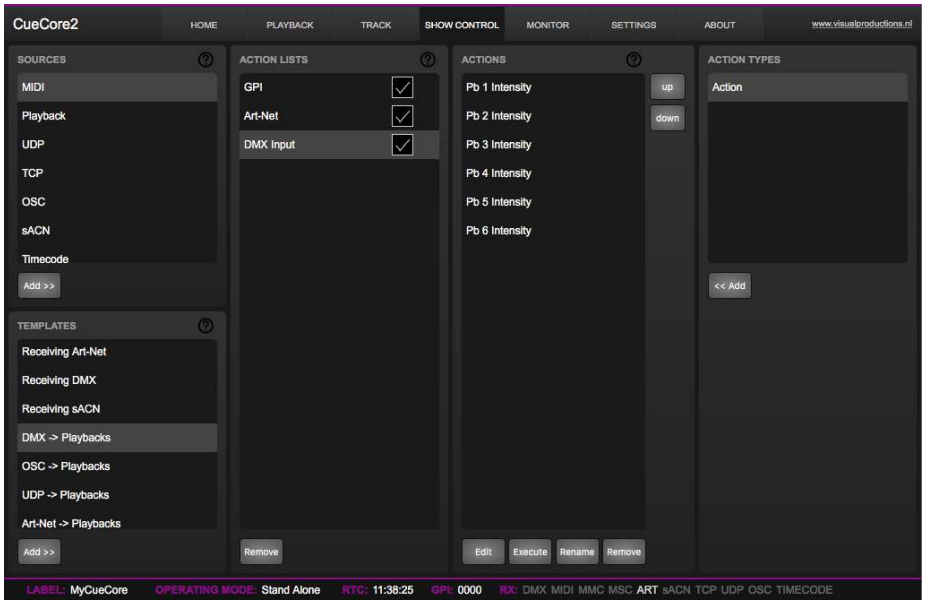


Figure 9.1: Show Control page

The Show Control page presents a system of 'actions'. A signal that the CueCore2 needs to respond to or perhaps convert into some other signal, needs to be expressed in an actions. With the exception of converting timecode protocols; this can be done in the Settings page (see page 58).Before programming actions please consider the Show Control structure in figure 9.2.

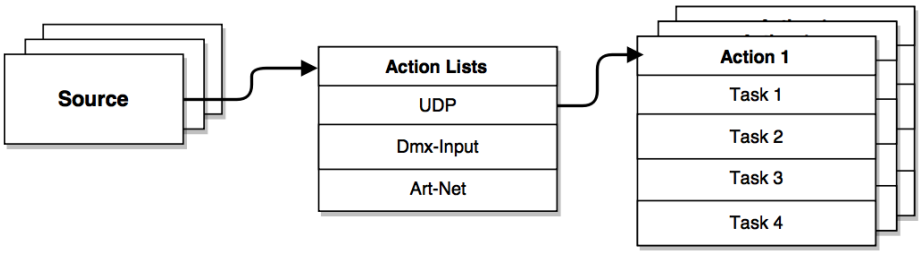


Figure 9.2: Show Control structure

The CueCore2 is capable of listening to various protocols. These available protocols are listed in Sources, however, the CueCore2 can only actively listen to 8 protocols at once. The active protocols are listed in 'Action Lists'. 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 CueCore2 will do when this action is triggered. Several tasks can be placed in the action. There are tasks available for a wide range of CueCore2 features and external protocols. Task types are detailed in Appendix A on page 72.

Please consult the API appendix on page 89 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.

9.1 Sources and Action Lists

The Sources listing presents all protocols that the CueCore2 is capable of receiving. It also includes internal features that can create events that can be used for triggering actions, such as the calendar-scheduler. These sources are available, however, they will only be actively listened to once moved to the action-list table.

UDP	UDP network messages
TCP	TCP network messages
OSC	OSC network message
DMX Input	DMX received on one or more of the DMX ports (switch port to input in the settings page)
Art-Net	Art-Net DMX data
sACN	sACN DMX data
Timecode	Timecode signal, specify the incoming timecode protocol on the Settings page.
Touch Screen	Triggers from VisualTouch. 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 VisualTouch.
Schedular	Triggers based on time, date, weekdays, sunrise & sunset
Playback	Events generated by the playbacks
Randomiser	The randomiser can generate a random number
System	Events such as 'Power on'
Variable	The Variable source works in combination with the variable task (For more information about the Variable task please refer to Task Types). The Variable task will set a value of which an enabled action-list type with Variable as Source will use as a trigger. The CueCore2 will keep the values of the 8 variables even after shut down so long as the RTC battery is not empty.
Timer	There are 4 internal timers in the CueCore2. An event will be raised when a timer expires. Timers are set and activated by the Timer tasks.
User List 1-4	These action-lists will never trigger an event, however, 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.

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

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 therefor sometimes omitted in the web GUI. The trigger-type field specifies what kind of signal the action will be triggered by. For example, when making an action in the Scheduler list there is the choice between 'DateAndTime' and 'WeekdayAndTime' trigger-types. The trigger-value specifies the actual signal value. In the schedular example the trigger-value could be set to "2016-03-24 11:00" or "Weekend 10:00" respectively.

In some action-lists 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 a Touch Screen list and it is linked to a button in the VisualTouch software, the flank will determine whether to trigger only when the button goes down or only when it goes up. Appendix B 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.

9.3 Tasks

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

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.

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

9.4 Templates

The Show Control page presents a list of templates. A template is a set of action-list, actions and task. These templates configure the CueCore2 to perform typical functions; for example convert Art-Net to DMX or control the 6 playbacks through OSC. The templates thus save time; otherwise actions should have been set up manually. They can also function as a guide to soften the learning curve on actions; a lot can be learned from adding a template and then exploring the actions and tasks it created. Please note that some templates require settings to changed in the settings page; for example the 'Receiving Art-

Net' template needs the DMX outlets to set to outputs in order to achieve an Art-Net to DMX conversion. Appendix C gives an overview of the available templates.

Chapter 10

Protocol Conversion

The CueCore2 is fitted with several communication ports and additionally supports various Ethernet-based protocols. Although some protocols are predominantly used for triggering the internal playbacks (such as GPI, UDP, OSC, MIDI, etc.) and some other protocols are mainly used for recording (such as DMX input, Art-Net and sACN) the CueCore2 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.

10.1 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
Digital GPI	On/Off
Analog GPI	percentage [0%,100%]
UDP	-
TCP	-
OSC	percentage [0%,100%], number, string, colour, On/Off
MIDI	number [0,127]
DMX	number [0,255]
Art-Net	number [0,255]
sACN	number [0,255]

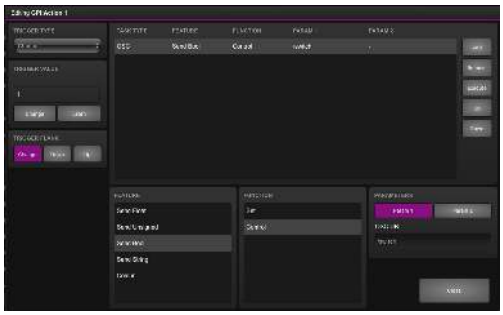
Although DMX, Art-Net and sACN are dedicated lighting protocols and naturally fit in the next category, their individual channels lend themselves well for

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 'Action list' table. Then add an action to this new action-list. 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.

Please consider two examples. Figure 10.1 shows a conversion between Digital GPI and OSC. This example assumes the GPI Port 1 is set to 'Digital' on the Settings page.

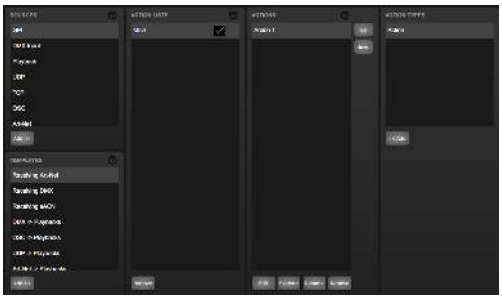
(a) Step 1



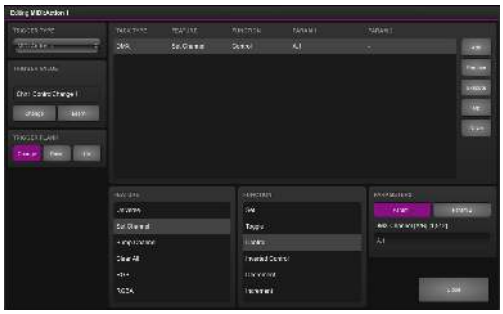
(b) Step 2

Figure 10.1: Conversion from GPI to OSC

Figure 10.2 shows a conversion between MIDI and DMX. This example assumes the DMX Port A is set to 'Output A' on the Settings page.



(a) Step 1



(b) Step 2

Figure 10.2: Conversion from MIDI to DMX

10.2 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 to some extend KiNet. The CueCore2 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, allowing the user to define the method of merging (HTP, LTP or Priority). All this is done with a minimal amount of configuration in the CueCore2. 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
DMX ->KiNet
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 'Action lists' table. Then add an action for each DMX Universe you wish to convert; e.g. when converting two DMX ports to Art-Net it requires two action to programmed. The trigger-type in the actions should be set to 'Universe' to make the CueCore2 process the 512 channels as a whole rather then process individual channels. Each action should contain a DMX-task with the 'feature' set to 'Universe'; all DMX Universe data is first being copied into the CueCore2's internal DMX buffer. From this buffer it can be copied to the DMX outlet or the other protocols such as Art-Net and sACN. Figure 10.3 provides a schematic for this data flow.

The 'function' determines how the DMX data is merged; it controls the prece-

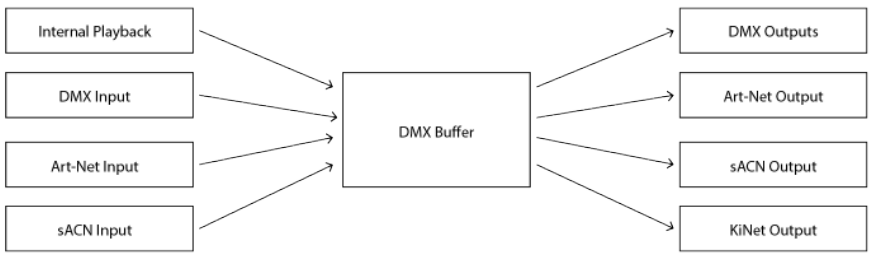


Figure 10.3: DMX merging data flow

dence. There is the following choice:

Function	
Control HTP	Highest Takes Precedence
Control LTP	Highest Takes Precedence
Control Priority	
Clear	

The HTP precedence is the default choice where all channels are compared and the highest levels are used for the merged output. Amongst all the playbacks set to LTP only one of them is included in the HTP merge; the LTP playback that has been activated latest. If there is task function set to Priority and this signal is actively received, then this data will be send directly to the output, temporarily suspending all HTP and LTP sources. When multiple conversions are set to Priority then those will be merged according to highest takes precedence. Figure 10.4 illustrates this mechanism.

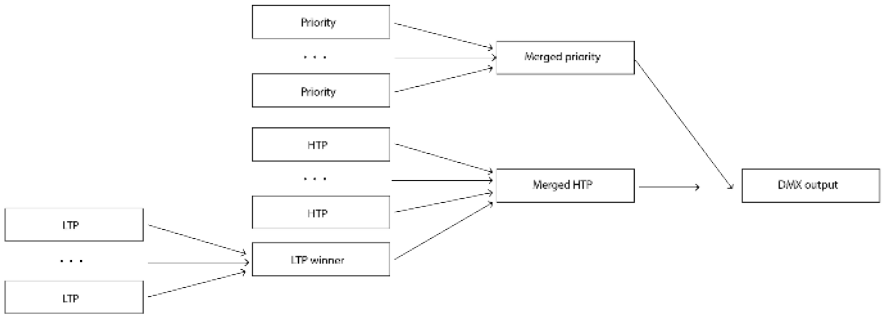


Figure 10.4: DMX merging precedence

The additional 'Clear' function is not related to the data merging precedence; 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 universes 0.0 and 0.1 to DMX output A and B respectively. Figure 10.5 shows action-list, figure 10.6 show the contents of 'Action 1' and figure 10.7 show the required configuration of the Settings page.

10.3 Converting Timecode Protocols

The CueCore2 supports several timecode protocols. The following table lists its capabilities:

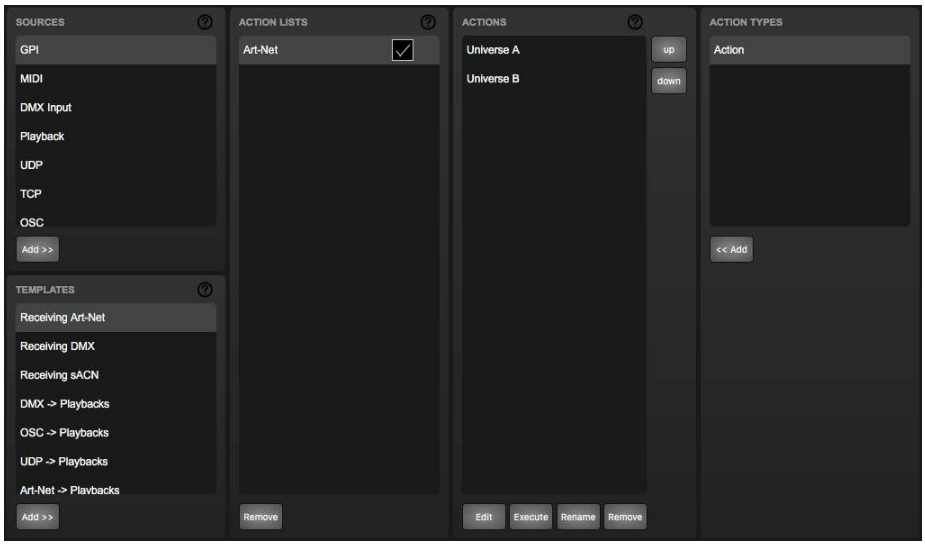


Figure 10.5: Converting Art-Net to DMX step 1

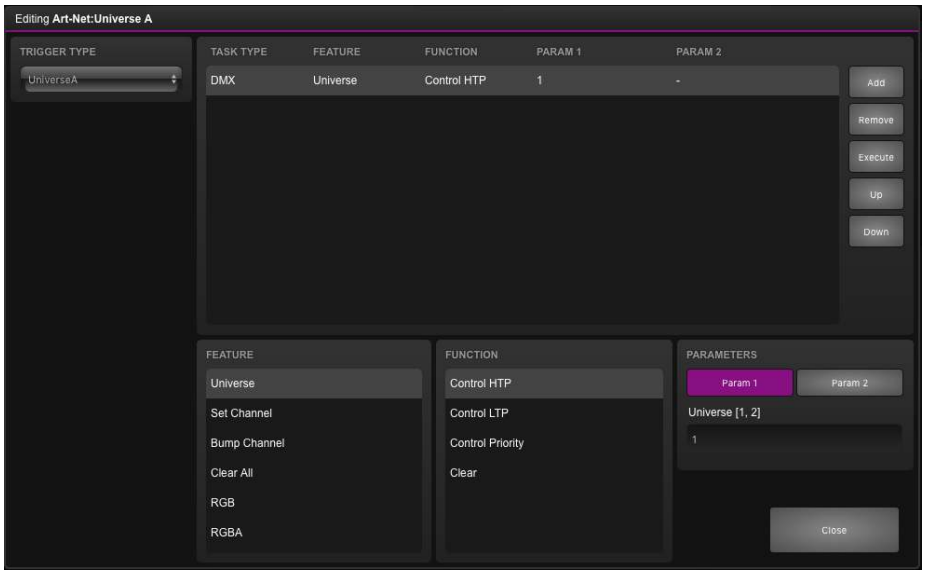


Figure 10.6: Converting Art-Net to DMX step 2

Figure 10.7: Converting Art-Net to DMX step 3

Timecode Protocol	Capability
SMPTE	Input
MTC	Input + Output
Art-Net Timecode	Input + Output
Internal Timecode	Output

These protocols are primarily used by the CueCore2 to synchronise its internal playbacks, however, the CueCore2 is also capable of converting an incoming timecode protocol into a different outgoing protocol. This conversion is made in the Settings page, please see figure 10.8.

Figure 10.8: Converting SMPTE timecode to MTC

To set up the conversion, choose an incoming protocol and one of the outgoing protocols.

Chapter 11

Monitors

This page allows the user to inspect the incoming and outgoing data, both DMX-type data (See figure 11.1) as well as control messages (See figure 11.2). Monitoring incoming and outgoing data can help the user troubleshoot during programming.

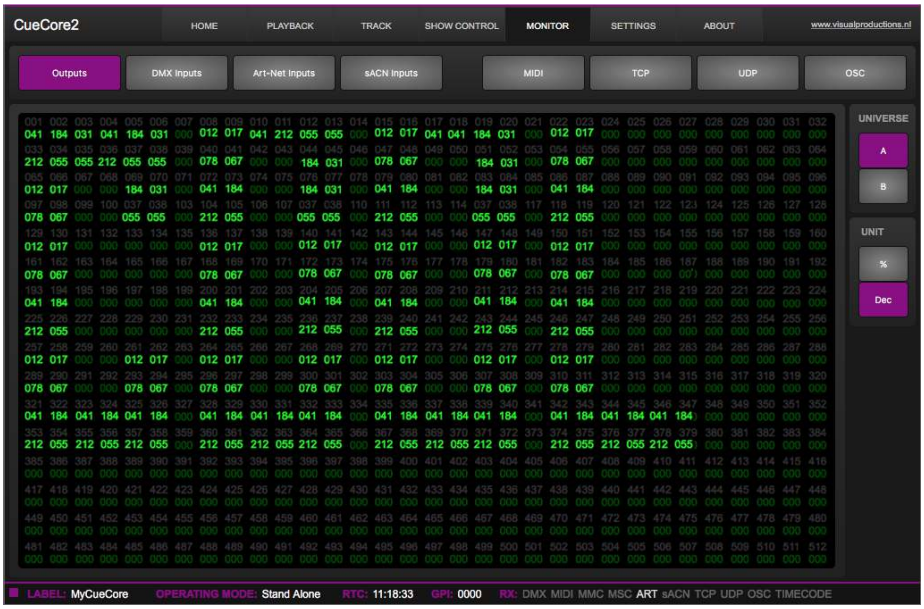


Figure 11.1: DMX Monitor page

In the Monitor page three different sources of input can be found (DMX, Art-Net and sACN), along with the control input and output sources (TCP, UDP and OSC). On the right side of the page there are the universes were the user can swap between the four of them or choose a preferable unit for displaying the requested information.

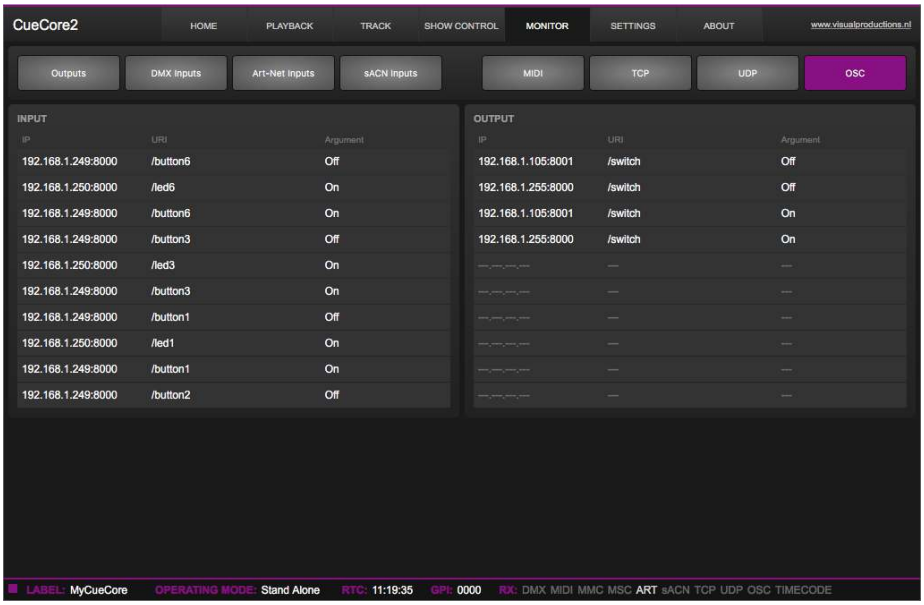


Figure 11.2: OSC Monitor page

Chapter 12

Settings

The CueCore2's settings are organised into sections, see the Settings page figure 12.1. This chapter will discuss each section.

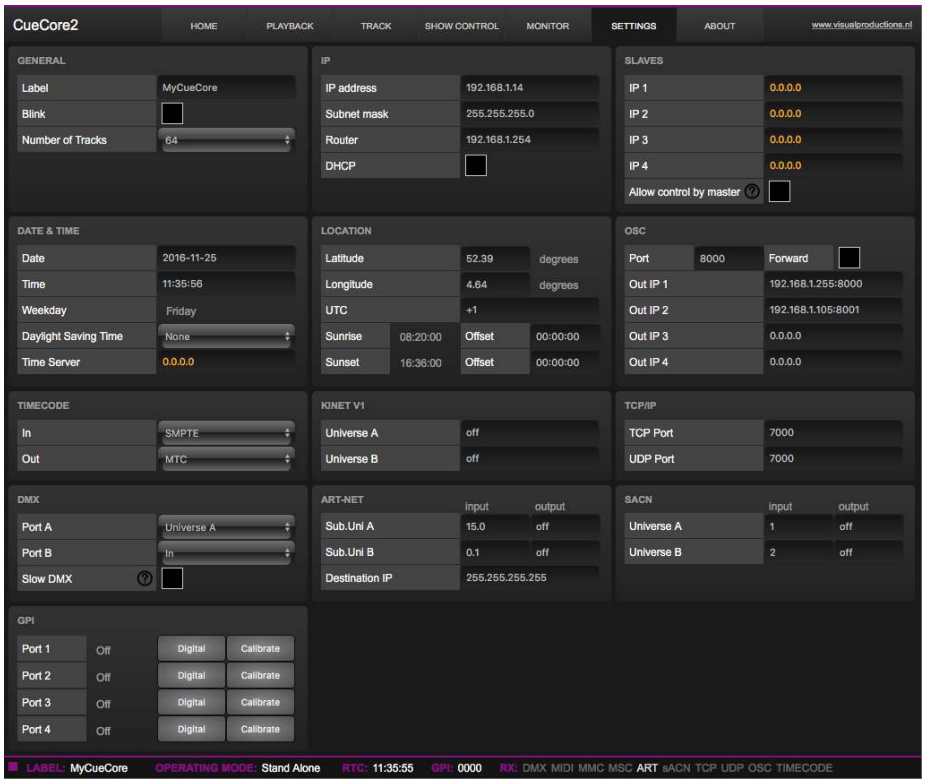


Figure 12.1: Settings page

12.1 General

You can change the CueCore2's label. This label can be used to distinguish the unit in a set-up with multiple devices. By enabling the 'Blink' checkbox the device's LED will blink to help to identify it amongst multiple devices. The Number of Tracks drop-down determines the organisation of the Track memory. This is discussed on page 41.

Figure 12.2: General Settings

Changing the number of tracks will result in loosing the current content of the tracks.

12.2 IP

The IP fields are for setting up the IP address and subnet mask of the CueCore2. 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 28).

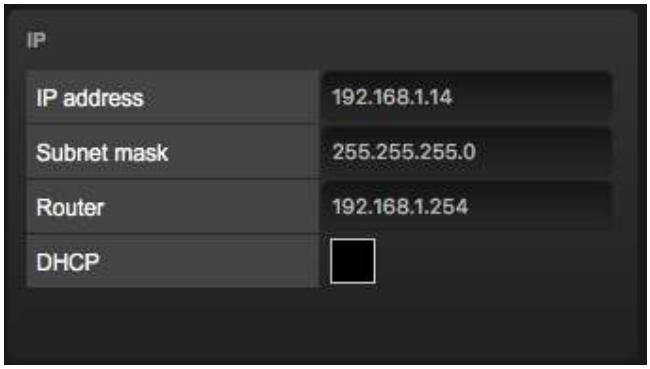


Figure 12.3: IP Settings

12.3 Slaves

This section enables the master-slave synchronisation.

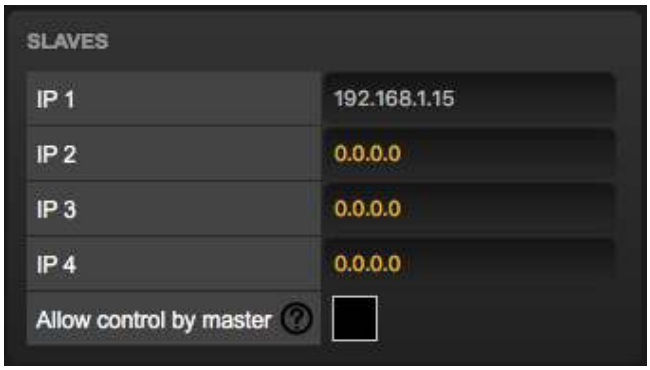


Figure 12.4: Slaves Settings

The master-CueCore2 should specify the IP addresses of its slaves. When the IP is indicated in white then the master-slave connection is established, otherwise the IP is indicated in orange. For creating a system with more than four slaves, a broadcast IP can be set. A typical broadcast IP address is 192.168.1.255, however, this depends on the subnet used.

The slave-CueCore2 units require the 'Allow control by master' checkbox to be enabled. Enabling 'Allow control by master' checkbox will cause playback data to be overwritten.

12.4 Date & Time

The date and time of the RTC can be set here. The clock has a back-up battery to keep the time during a power down. Daylight Saving Time (DST) is supported for the regions Europe and United States.

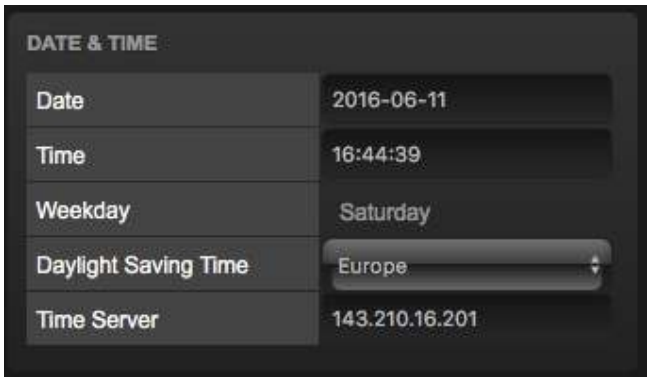


Figure 12.5: Date & Time Settings

The Time Server field allows a NTP server to specified. At start up, the CueCore2 will fetch the time and date from this server. Additionally, an action can be used to fetch the time. The DST and the Coordinated Universal Time (UTC) are taken into account when obtaining the time for the NTP server.

The following table lists suggested NTP servers.

Continent	Server
North America	64.90.182.55
South America	201.49.148.135
Europe	143.210.16.201
Africa	196.23.245.74
Asia	133.100.9.2
Australia	137.92.140.80

12.5 Location

The astronomical clock in the CueCore2 calculates the sunrise and sunset times based on day of the year, latitude, longitude and UTC. The latitude and longitude values define the position in the world and should be entered in degrees. The latitude value should be positive for North and negative for South, the longitude should positive for East and negative for West. The website http://www.findlatitudeandlongitude.com/ can help discover the latitude and longitude values for the current location. The time-zone and perhaps daylight saving time of the current location is expressed in the UTC value. UTC is - in this context - equivalent to Greenwich Mean Time (GMT). For example, Visual Productions' HQ is based in the city of Haarlem, the Netherlands. During the winter the UTC equals +1 and in the summer during day light saving time it is set to +2. So, the settings for the Visual Productions HQ are shown in Figure 12.6.

Figure 12.6: Location settings

The Offset fields allows to shift the sunrise and sunset triggers, both earlier and later. For example, to have a trigger half an hour before sunrise set the offset to -00:30.

12.6 OSC

External equipment sending OSC messages to the CueCore2 need to be aware of the number specified in the 'Port' field. This is the port the CueCore2 listens to for incoming messages.

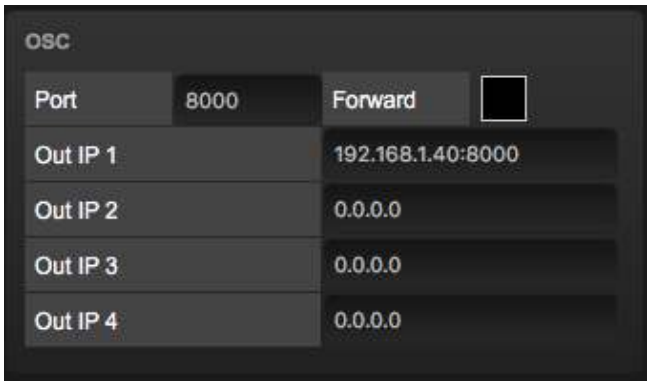


Figure 12.7: OSC Settings

The CueCore2 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 CueCore2 copy every incoming OSC message and send it the addresses specified in the 'Out IP' fields.

12.7 Timecode

The CueCore2 can receive SMPTE, MTC and Art-Net timecode. This section allows to choose one of these protocols as the timecode source. Alternatively, the CueCore2 also has 'internal' timecode; a timecode generated by the unit itself. Synchronisation of playbacks and actions depend on this choice.
The device is also capable of transmitting MTC and Art-Net timecode. This section allows you to enable this timecode output. Please refer to page 58 on more details about converting timecode protocols.

12.8 KiNet v1

The CueCore2 features transmission of DMX data via KiNet; it supports KiNet protocol version 1.

Figure 12.8: Timecode Settings

Figure 12.9: Kinet Settings

12.9 TCP/IP

Defines the listening ports for TCP and UDP messages. External system intending to send TCP or UDP message to the CueCore2 should need to know the unit's IP address and this port number. By default both ports are set to 7000.

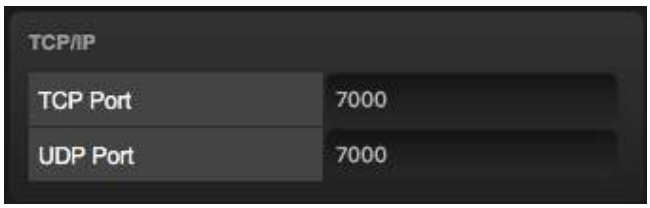


Figure 12.10: TCP/IP settings

12.10 DMX

The DMX settings specify wether a DMX port is input or output. Select universe to set to output.
When the 'Slow DMX' checkbox is enabled, the CueCore2 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.

12.11 Art-Net

The Art-Net feature in the CueCore2 supports 2 universes out or 2 universes in. These universes can be mapped to any of the 256 available universes in the

Figure 12.11: DMX settings

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.

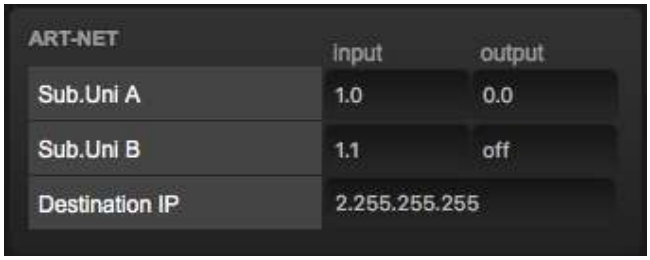


Figure 12.12: Art-Net settings

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.

12.12 sACN

The CueCore2 supports 2 incoming sACN universes and 2 outgoing universes. Each 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 fields.

12.13 GPI

Each GPI port can be configured either as digital or analog input. It is advised to calibrate the GPI port when it is set to analog. To calibrate, enable the Calibrate button, apply the minimum voltage to the GPI, apply the maximum

Figure 12.13: sACN settings

voltage and then disable the Calibrate button.

Be careful not to supply more than 10V to the GPI port as that might cause permanent damage.

Appendices

Appendix A

Task Types

Tasks allow you to automate the functionality in the CueCore2. 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.

A.1 Playback

Manipulate one of the six playbacks.

Feature	Function	Parameter 1	Parameter 2
Intensity	Set	Playback Index	percentage [0%,100%]
Intensity	Control	Playback Index	-
Set Rate	Set	Playback Index	percentage [-100%,100%]
Set Rate	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
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	-

A.2 Playback Master

Manipulate the master settings on the Playback page.

Feature	Function	Parameter 1	Parameter 2
Intensity	Set	-	percentage [0%,100%]
Intensity	Control	-	-
Set Rate	Set	-	percentage [-100%,100%]
Set Rate	Control	-	-
Fade time	Set	Time	-
Fade time	Control	-	-
Release	All	-	-

A.3 Track

Manipulate the settings on the Track page.

Feature	Function	Parameter 1	Parameter 2
Program	Stop	-	-
Program	Record	Track Index	-
Program	Erase	Track Index	-
Intensity Map	Clear	-	-
Intensity Map	Capture DMX	-	-
Intensity Map	Capture Art-Net	-	-
Intensity Map	Capture sACN	-	-

A.4 UDP

Send an 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

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

A.5 OSC

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

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

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

A.6 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 #	-
Universe	Control LTP	Universe #	-
Universe	Control Priority	Universe #	-
Universe	Clear	Universe #	-
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	-
RGBA	Control	DMX Address	-
XY	Control	DMX Address	-
XxYy	Control	DMX Address	-

A.7 MIDI

Send an MIDI message.

Feature	Function	Parameter 1	Parameter 2
Send	Set	MIDI Address	MIDI Value
Send	Control	MIDI Address	-

A.8 MMC

Send an MMC (MIDI Machine Control) message via the MIDI port.

Feature	Function	Parameter 1	Parameter 2
Send	Start	MIDI Channel	-
Send	Stop	MIDI Channel	-
Send	Restart	MIDI Channel	-
Send	Pause	MIDI Channel	-
Send	Record	MIDI Channel	-
Send	Deferred Play	MIDI Channel	-
Send	Record Exit	MIDI Channel	-
Send	Record Pause	MIDI Channel	-
Send	Eject	MIDI Channel	-
Send	Chase	MIDI Channel	-
Send	Fast Forward	MIDI Channel	-
Send	Rewind	MIDI Channel	-
Send	Goto	MIDI Channel	Time

A.9 GPI

Manipulate the GPI port.

Feature	Function	Parameter 1	Parameter 2
Sample Binary	Set	-	-
Refresh	Set	-	-

A.10 Time Server

Reach out to the Time Server specified in the Settings page.

Feature	Function	Parameter 1	Parameter 2
Refresh	Set	-	-

A.11 Variable

Manipulate one of the eight 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	Control Scaled	Variable #	-
Set Value	Control Offset	Variable #	-
Refresh	Set	Variable #	-
Single Dimmer	Set	Variable #	Delta

A.12 System

Miscellaneous tasks.

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

A.13 Action

Trigger another action.

Feature	Function	Parameter 1	Parameter 2
Link	Set	Action	-

A.14 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	-

A.15 Randomiser

Trigger the Randomizer to generate a new random number.

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

A.16 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

A.17 Timecode

Manipulate the internal timecode generator.

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

Appendix B

Trigger Types

The following tables list the different types of triggers that can be used in the CueCore2. The different types are accompanied with values and flanks.

B.1 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.

B.2 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.

B.3 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 '/'.

B.4 DMX Input

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

B.5 Art-Net

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

B.6 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

B.7 Timecode

Trigger Type	Trigger Value	Flank	Description
Time	Frame	-	Timecode frame
Receiving	-	Change	Start receiving or loose timecode signal
Receiving	-	Stop	Lost timecode signal
Receiving	-	Start	Start receiving timecode signal

B.8 Touch Screen

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

B.9 Scheduler

Trigger Type	Trigger Value	Flank	Description
WeekdayAndTime	-	-	Enable weekdays and specify a time (don't care 'X' can be used)
DateAndTime	-	-	Specify a specific date and time (don't care 'X' can be used)
Sunrise	-	-	When the sun rises in the morning
Sunset	-	-	When the sun goes down in the evening
DaylightST	-	Change	Daylight Saving Time period starts or ends
DaylightST	-	Stop	Daylight Saving Time period ends
DaylightST	-	Start	Daylight Saving Time period starts

B.10 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%

B.11 Randomizer

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

B.12 System

Trigger Type	Trigger Value	Flank	Description
Startup	-	-	The CueCore2 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

B.13 Variable

Trigger Type	Trigger Value	Flank	Description
Channel	Variable Index	-	The specified variable changes
Variable 1	Number in the range of [0,255]	Change	Variable 1 becomes equal or not equal to the specified number
Variable 1	Number in the range of [0,255]	Equal	Variable 1 becomes equal to the specified number
Variable 1	Number in the range of [0,255]	Unequal	Variable 1 becomes not equal to the specified number
Variable 2	Number in the range of [0,255]	Change	Variable 2 becomes equal or not equal to the specified number
Variable 2	Number in the range of [0,255]	Equal	Variable 2 becomes equal to the specified number
Variable 2	Number in the range of [0,255]	Unequal	Variable 2 becomes not equal to the specified number
Variable 3	Number in the range of [0,255]	Change	Variable 3 becomes equal or not equal to the specified number
Variable 3	Number in the range of [0,255]	Equal	Variable 3 becomes equal to the specified number
Variable 3	Number in the range of [0,255]	Unequal	Variable 3 becomes not equal to the specified number
Variable 4	Number in the range of [0,255]	Change	Variable 4 becomes equal or not equal to the specified number
Variable 4	Number in the range of [0,255]	Equal	Variable 4 becomes equal to the specified number
Variable 4	Number in the range of [0,255]	Unequal	Variable 4 becomes not equal to the specified number
Variable 5	Number in the range of [0,255]	Change	Variable 5 becomes equal or not equal to the specified number
Variable 5	Number in the range of [0,255]	Equal	Variable 5 becomes equal to the specified number
Variable 5	Number in the range of [0,255]	Unequal	Variable 5 becomes not equal to the specified number
Variable 6	Number in the range of [0,255]	Change	Variable 6 becomes equal or not equal to the specified number
Variable 6	Number in the range of [0,255]	Equal	Variable 6 becomes equal to the specified number
Variable 6	Number in the range of [0,255]	Unequal	Variable 6 becomes not equal to the specified number
Variable 7	Number in the range of [0,255]	Change	Variable 7 becomes equal or not equal to the specified number
Variable 7	Number in the range of [0,255]	Equal	Variable 7 becomes equal to the specified number
Variable 7	Number in the range of [0,255]	Unequal	Variable 7 becomes not equal to the specified number
Variable 8	Number in the range of [0,255]	Change	Variable 8 becomes equal or not equal to the specified number

B.14 Timer

Trigger Type	Trigger Value	Flank	Description
-	Timer Index	Change	The timer starts or stops
-	Timer Index	Stop	The timer stops
-	Timer Index	Start	The timer starts

Appendix C

Templates

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

Template	Description
Receiving DMX	Receiving DMX on all ports. DMX properties in the Settings page have to be configured accordingly.
Receiving Art-Net	Receiving DMX on all universes. Art-Net properties in the Settings page have to be configured accordingly.
Receiving sACN	Receiving sACN on all universes. sACN properties in the Settings page have to be configured accordingly.
DMX ->Playbacks	DMX port A (channel 1-6) will control the intensity of all six playbacks. When a channel >0% it will activate the playback, when set 0% it the playback will be released.
Art-Net ->Playbacks	Art-Net input universe A will control the intensity of all six playbacks. When a channel >0% it will activate the playback, when set 0% it the playback will be released.
VisualTouch->Playbacks	Creates a VisualTouch layout with buttons and sliders to operate the six playbacks.
DMX ->MIDI	Translates 8 channels from DMX port A into outgoing MIDI ControlChange messages in MIDI channel 1. DMX port A should be configured as an input in the Settings page.

Appendix D

API

The CueCore2 is pre-programmed to make its internal functionality available via OSC and UDP. There is a simple API implemented for each protocol. Notwithstanding these API's, it is possible to create your own OSC and UDP implementation in the Show Control page.

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/release	-	Release the playback
/core/pb/1/intensity	float	Set the playback's intensity
/core/pb/1/rate	float	Set the playback's intensity
/core/pb/release	-	Release all playbacks
/core/pb/intensity	float	Set the master intensity
/core/pb/rate	float	Set the master rate
/core/pb/fade	string	Set the master fade time

The following table uses track #1 as an example. The number '1' can be replaced by any number in the range of [1,128].

URI	Parameter	Description
/core/tr/select	integer	Select a track
/core/tr/erase	-	Erase the selected track
/core/tr/record	-	Start recording the selected track
/core/tr/stop	-	Stop recording
/core/tr/1/erase	-	Erase track #1
/core/tr/1/record	-	Start recording track #1

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

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

The following table shows how to manipulate the internal timecode.

URI	Parameter	Description
/core/tc/start	-	Start timecode
/core/tc/stop	-	Stop timecode
/core/tc/restart	-	Restart timecode
/core/tc/pause	-	Pause timecode
/core/tc/set	time-string	Set the timecode frame at the specified string. For example "23:59:59.24"

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/blink	-	Momentarily flashes the CueCore2's LED

D.2 UDP

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-release	Release the playback
core-pb-1-intensity=<float>	Set the playback's intensity
core-pb-1-rate=<float>	Set the playback's intensity
core-pb-release	Release all playbacks
core-pb-intensity=<float>	Set the master intensity
core-pb-rate= <float>	Set the master rate
core-pb-fade=<text>	Set the master fade time

The following table uses track #1 as an example. The number '1' can be replaced by any number in the range of [1,128].

String	Description
core-tr-select=<integer>	Select a track
core-tr-erase	Erase the selected track
core-tr-record	Start recording the selected track
core-tr-stop	Stop recording
core-tr-1-erase	Erase track #1
core-tr-1-record	Start recording track #1

The following table uses actionlist #1 as an example. The number '1' can

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

String	Description
core-al-1-1-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 shows how to manipulate the internal timecode.

String	Description
core-tc-start	Start timecode
core-tc-stop	Stop timecode
core-tc-restart	Restart timecode
core-tc-pause	Pause timecode
core-tc-set=<text>	Set the timecode frame at the specified string. For example "23:59:59.24"

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 active miscellaneous functions.

String	Description
core-blink	Momentarily flashes the CueCore2's LED

Index