

Application Notes

TPN / OPTN System Configuration Guide



Table of Contents

1. INTRODUCTION 3

1.1. INTRODUCTION 4

1.1.1. The Purpose of the Document..... 4

1.1.2. What is the TPN / OPTN AV Network?..... 4

1.1.3. Lightware SDVoE-Based Devices 4

1.2. SUPPORTED MODELS 4

1.3. APPLICATION MATRIX 5

2. PREPARATION OF THE SYSTEM 6

2.1. STARTUP CHECKLIST 7

2.1.1. Port Connections..... 7

2.1.2. Switching on the Devices..... 7

2.1.3. Network Switch Configuration..... 7

2.2. TROUBLESHOOTING CHECKLIST.....7

2.2.1. TPX/OPTX and TPN/OPTN Devices on the Network..... 7

2.2.2. Ethernet Connection over the Endpoints 7

2.2.3. Only One SDVoE Server Presence on the AV Network..... 7

2.2.4. Dynamic IP Address (DHCP) is Recommended for the Endpoints 7

2.2.5. Gateway Address Validation..... 7

2.2.6. Icron Control over SDVoE Server 7

2.3. RESERVED PORTS AND SECURITY OPTIONS 8

2.4. BLUERIVER SERVERS ON THE NETWORK 9

3. LARA - SYSTEM CONFIGURATION GUIDE..... 10

3.1. FIRST STEPS 11

3.2. CONNECTING TO THE CTU 11

3.3. ACCESSING LARA.....11

3.4. CONFIGURATION OF LARA FOR TPN MATRIX.....12

3.4.1. Creating the Driver Module 12

3.4.2. Creating an Instance from the Driver Module..... 13

3.4.3. Creating the Userpanel Module 13

3.4.4. Creating an Instance from the Userpanel Module 14

3.4.5. Starting of the LARA Services 14

3.4.6. Configuration of the UI 15

3.5. CONFIGURATION OF LARA FOR ICRON USB 2.0 17

3.5.1. Creating the Driver Module 17

3.5.2. Creating an Instance from the Driver Module..... 17

3.5.3. Creating the Userpanel Module 18

3.5.4. Creating an Instance from the Userpanel Module 18

3.5.5. Starting of the LARA Services 19

3.5.6. Configuration of the UI 19

3.6. CUSTOMIZATION OF THE UI.....21

3.6.1. Changing the Company Logo 21

3.6.2. Changing the Color of the Background..... 22

3.6.3. Changing the Color of the Crosspoint Tiles..... 23

3.7. TROUBLESHOOTING.....24

3.7.1. Gateway Address Validation..... 24

3.7.2. Icron Control over SDVoE Server 24

4. BLUERIVER AV MANAGER - CONTROLLING OVER GUI.....25

4.1. INTRODUCTION 26

4.2. INSTALLATION.....26

4.2.1. Accessing the Software 27

4.3. HOME PAGE 27

4.4. MONITORING MENU 28

4.5. OPERATE MENU.....29

4.5.1. Video and Wall..... 29

4.5.2. Audio 30

4.5.3. USB..... 30

4.5.4. RS232..... 31

4.5.5. CEC..... 31

4.5.6. Presets 32

4.6. CONFIGURE MENU.....32

4.6.1. Devices..... 32

4.6.2. Wall..... 33

4.7. SETTINGS MENU.....35

4.8. CONNECTION VIA MOBILE DEVICE.....35

5. BLUERIVER API MANAGEMENT - CONTROLLING OVER COMMANDS 36

5.1. INSTRUCTIONS FOR THE TERMINAL APPLICATION USAGE 37

5.1.1. Terminal Application 37

5.1.2. Web Browser Plugins 37

5.2. BLUERIVER API MANAGEMENT STRUCTURE EXPLANATION 38

5.2.1. Command Types 38

5.2.2. Background Commands 38

5.2.3. Monitoring Command Execution..... 38

5.3. REST API HTTP COMMAND BASICS 39

5.4. AV INTERFACE CONTROL OVER API 39

5.4.1. Applying API Version..... 39

5.4.2. Starting an HDMI Audio/Video Stream..... 40

5.4.3. Stopping a Stream..... 40

5.4.4. Switching an HDMI Audio/Video Stream..... 41

5.4.5. Disconnecting an HDMI Stream 42

5.4.6. Setting the EDID..... 43

5.4.7. Scaler Mode Setting..... 44

5.4.8. Scaler Parameter Settings 45

5.4.9. Restoring the Factory Default Settings 46

5.4.10. Rebooting the Endpoint Device 47

5.4.11. Renaming the Device 48

5.5. ICRON USB INTERFACE CONTROL OVER API 49

5.5.1. Applying API Version..... 49

5.5.2. Pairing the Devices - SUI Mode 49

5.5.3. Unpairing One Device - SUI Mode..... 49

5.5.4. Unpairing All Devices - SUI Mode 50

5.5.5. Pairing the Devices - MSA Mode 50

5.5.6. Setting the USB HID Mode..... 52

Document Information

Document revision: v1

Release date: 05-06-2025

Editor: Tamas Forgacs

Contact Us

sales@lightware.com
+36 1 255 3800

support@lightware.com
+36 1 255 3810

Lightware Visual Engineering PLC.
Gizella 51-57, Budapest H-1143, Hungary
www.lightware.com

©2025 Lightware Visual Engineering. All rights reserved. All trademarks mentioned are the property of their respective owners. Specifications subject to change without notice.

1

Introduction

The first chapter is about the technical background of the functions and features that will be introduced in the following chapters.

- ▶ [INTRODUCTION](#)
- ▶ [SUPPORTED MODELS](#)
- ▶ [APPLICATION MATRIX](#)

1.1. Introduction

1.1.1. The Purpose of the Document

Lightware's SDVoE-based AV extenders supports several different methods configuring and controlling the AV network. The document is a step-by-step guide about these methods with useful know-hows and troubleshooting sections.

Three main methods are described in the document in the details:

- AV network configuration and controlling by **LARA** - see the details in the [LARA - System Configuration Guide](#) chapter.
- AV network configuration and controlling by **BlueRiver AV Manager** - see the details in the [BlueRiver AV Manager - Controlling over GUI](#) chapter.
- AV network configuration and controlling by **BlueRiver API** - see the details in the [BlueRiver API Management - Controlling over Commands](#) chapter.

1.1.2. What is the TPN / OPTN AV Network?

The TPN- and OPTN-based AV system is a virtual AV matrix with several transmitters, receivers, and a Control Unit (CTU) that controls the AV network.



1.1.3. Lightware SDVoE-Based Devices

TPN Control Unit

TPN-CTU-X50 is a Control Unit (CTU) for Lightware's TPN AV Over IP product line. With this control unit and an off-the-shelf 10G Ethernet switch installed as a crosspoint in the system, a virtual matrix can be created for TPN devices connected to the IP network as input and output endpoints.

TPN / OPTN series Endpoints

The HDMI-TPN series transmitter and receiver devices with SDVoE technology are Lightware's newest development allowing users to extend HDMI 2.0 signals up to 4K60 4:4:4 video resolution from a single source to multiple destinations through 10G Ethernet networks.



1.2. Supported Models

Controller Unit

- TPN-CTU-X50
- UCX-2x1-TPN-TX20 (-LCC)
- UCX-4x3-TPN-TX20 (-LCC)
- DCX-3x1-TPN-TX10

TPN (CATx-based) Endpoint Models

- All available Lightware TPN models and any 3rd-party SDVoE based extenders

OPTN (Fiber Optical-based) Endpoint Models

- All available Lightware OPTN models and any 3rd-party SDVoE based extenders

Icron-based USB 2.0 & KVM Extenders

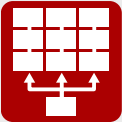
- All available Lightware USB20 series Icron extenders

1.3. Application Matrix

The TPN / OPTN AV matrix systems can be configured and controlled over three main methods:

- **LARA** - see the details in the [LARA - System Configuration Guide](#) chapter;
- **BlueRiver AV Manager** - see the details in the [BlueRiver AV Manager - Controlling over GUI](#) chapter;
- **BlueRiver API command protocol** - see the details in the [BlueRiver API Management - Controlling over Commands](#) chapter.

All three methods have different skills and the following table helps to find the best solution to the user’s requirements.

	 AV matrix management and control	 Video wall	 Icron USB 2.0 matrix management	 Serial message sending	 Room automation actions
 LARA	✓ Link to the section: Configuration of LARA for TPN Matrix	✗	✓ Link to the section: Configuration of LARA for Icron USB 2.0	✗	✓ Learn more in the user manual of LARA (PDF)
 BlueRiver AV Manager	✓ Link to the section: Operate Menu	✓ Link to the section: Operate Menu	✓ Link to the section: Operate Menu	✓ Link to the section: Operate Menu	✗
 BlueRiver API	✓ Link to the section: AV Interface Control over API	✗	✓ Link to the section: Icron USB Interface Control over API	✗	✗

2

Preparation of the System

The chapter makes a list about the preparation steps of the SDVoE-based AV network. The following sections can be found here:

- ▶ [STARTUP CHECKLIST](#)
- ▶ [TROUBLESHOOTING CHECKLIST](#)
- ▶ [RESERVED PORTS AND SECURITY OPTIONS](#)
- ▶ [BLUERIVER SERVERS ON THE NETWORK](#)

2.1. Startup Checklist

The following section provides a checklist related to the startup of the TPN / OPTN matrix system before the system configuration begins.

2.1.1. Port Connections

Before powering on the system, please check the **Connecting steps** section of the [TPN / OPTN user manual](#) to be sure all installation steps are met with the connection requirements.

2.1.2. Switching on the Devices

Power on the network switch, the CTU, all the transmitters and the receivers, the source, and the sink devices as well.

ATTENTION! Lightware recommends to power on the network switch at first. The bootup and configuration loading time of the network switch is longer than the Lightware extenders'. If the CTU and the endpoint devices boot up earlier than the switch, the AV network may be not to able to restore the last configuration settings.

2.1.3. Network Switch Configuration

One of the most important part of the TPN/OPTN system setup is the network switch configuration. Learn more details about this topic and find step-by-step descriptions covered various switch vendors in the [Installation and Network Setup Guide for TPN / OPTN Extenders Application Notes](#). The document can be downloaded on our website.

2.2. Troubleshooting Checklist

2.2.1. TPX/OPTX and TPN/OPTN Devices on the Network

If there are TPX/OPTX (point-to-point) devices and TPN/OPTN (point-multipoint) devices mixed as well in the AV system, it is highly recommended to be in **separated networks**.

2.2.2. Ethernet Connection over the Endpoints

The Ethernet connection ports of the endpoint devices is **highly NOT recommended to use connecting to other networks** but the TPN/OPTN network only.

2.2.3. Only One SDVoE Server Presence on the AV Network

The recommended setup of a TPN/OPTN AV network is **using only one SDVoE server**.

For example, the TPN-CTU-X50 control unit runs an SDVoE server and If a laptop is also connected to the same AV network which runs a BlueRiver AV Manager that includes another SDVoE server. This setup with two SDVoE servers may cause unexpected failure symphthons in the AV network.

2.2.4. Dynamic IP Address (DHCP) is Recommended for the Endpoints

The IP address setting of the endpoint devices is highly recommended to be **dynamic (DHCP)**. If it is static IP address or mixed DHCP and static, the endpoints may set to link local and the enpoint could be unreachable and can be fixed by restarting the devices only. See more details about the IP address settings in the [Installation and Network Setup Guide for TPN / OPTN Extenders Application Notes](#).

2.2.5. Gateway Address Validation

If the IP address setting of the device is dynamic (DHCP), check the gateway address validation. If the gateway address is 0.0.0.0 or left empty, the discovery process of the Icron USB 2.0 can be compromised.

The screenshot shows the 'Network' tab in a settings application. Under 'NETWORK SETTINGS', the 'Current gateway address' is set to '192.168.7.254'. Other settings include 'Current IP address: 192.168.4.45', 'Current subnet mask: 255.255.252.0', 'Host name: lightware-DB385678', and 'Obtain IP address automatically (DHCP, AutoIP):' checked. Static settings are also visible at the bottom.



Valid gateway address

The screenshot shows the 'Network' tab in a settings application. Under 'NETWORK SETTINGS', the 'Current gateway address' is set to '0.0.0.0'. Other settings include 'Current IP address: 169.254.11.134', 'Current subnet mask: 255.255.0.0', 'Host name: lightware-00007344', and 'Obtain IP address automatically (DHCP, AutoIP):' checked. Static settings are also visible at the bottom.



Invalid gateway address

2.2.6. Icron Control over SDVoE Server

Make sure that the controlling Icron through SDVoE control server setting is **disabled**.

The screenshot shows the 'BLUERIVER / SDVOE SETTINGS' section. It contains two settings: 'SDVoE Control Server:' with a checked checkbox and 'Enable' text, and 'Control ICron through SDVoE Control Server:' with an unchecked checkbox and 'Enable' text.

ATTENTION! Enabling this option might interfere with using the Icron LARA module and might prevent the ability to change SUI / MSA mode of the extenders.

2.3. Reserved Ports and Security Options

The following table summarizes the ports, protocols, features and the security options.

Port number	Protocol	Function	Affected device		Affected software	Port disable option	Encryption	Authentication	Other features
			CTU	Endpoints					
80	TCP	HTTP port (LW3 over WS, REST API, LARA user panels)	✓	✗	LDC, LDU2	✓	✗	✓	Firmware update, Welcome Screen image upload, Log files, User Scripts Serial messaging
443	TCP	HTTPS port (LW3 over WSS, REST API, LARA management GUI)	✓	✗	LDC, LDU2	✓	✓	✓	
6107	TCP	LW3 protocol	✓	✗	LDC	✓	✗	✓	
69	UDP	Firmware update TFTP	✓	✗	LDU2	✗	✗	✗	
49990	UDP		✓	✗		✗	✗	✗	
49995	UDP		✓	✗		✗	✗	✗	
10004	UDP		✗	✓		✗	✗	✗	
8001, 8002	TCP	Serial over IP (RS-232)	✓	✗	-	✓	✗	✗	
6108	TCP	Lightware Multicast Device Management (LMDM)	✓	✓	-	✗	✗	✗	
6970	TCP	TCI API protocol	✓	✗	BlueRiver Control Server	✗	✗	✗	
8080	TCP	REST API and websocket	✓	✗		✗	✗	✗	
230.76.87.82:37422	UDP	Lightware Multicast Device Protocol (LMDP)	✗	✓	-	✗	✗	✗	
10001, 10002, 10003, 10004	UDP	Serial / Infrared messages between the endpoints and CTU	✓	✓	-	✗	✗	✗	
6137, 6971	UDP	USB 2.0 discovery	✗	✓	-	✗	✗	✗	
6973	UDP	USB KVM connection between the AU2K series endpoints	✗	✓	-	✗	✗	✗	
6969	UDP	Communication between the API server and the endpoints	✓	✓	-	✗	✗	✗	
224.0.1.188:1900	UDP	SSDP messaging between the BlueRiver SDP and the endpoint devices	✗	✓	-	✗	✗	✗	
224.0.0.251: 5353	UDP	mDNS /Bonjour (Device Discovery)	✓	✓	LDC, LDU2	✗	✗	✗	
230.76.87.82: 37421	UDP	Remote IP	✓	✓	LDC, LDU2	✗	✗	✗	

INFO: The ports are necessary to be passed via a network switch/firewall for proper operation between the device and the softwares.

INFO: The **AU2K** endpoint models are built with two separated network boards with **two MAC addresses** and two IP addresses. In case of dynamic IP address (DHCP) setting, one device will reserve two different IP addresses.

2.4. BlueRiver Servers on the Network

The recent stage of development the CTU and the TPN / OPTN endpoint devices do not support the multi-server presence on the network. It means only one BlueRiver server is accepted in the Lightware's SDVoE AV system.

If a BlueRiver server is running on a PC, and it's connected to the AV network beside the TPN-CTU-X50, it causes two servers presence on the network.

Solution

The BlueRiver server shall be stopped in the PC by terminating it down in the Windows Task Manager.

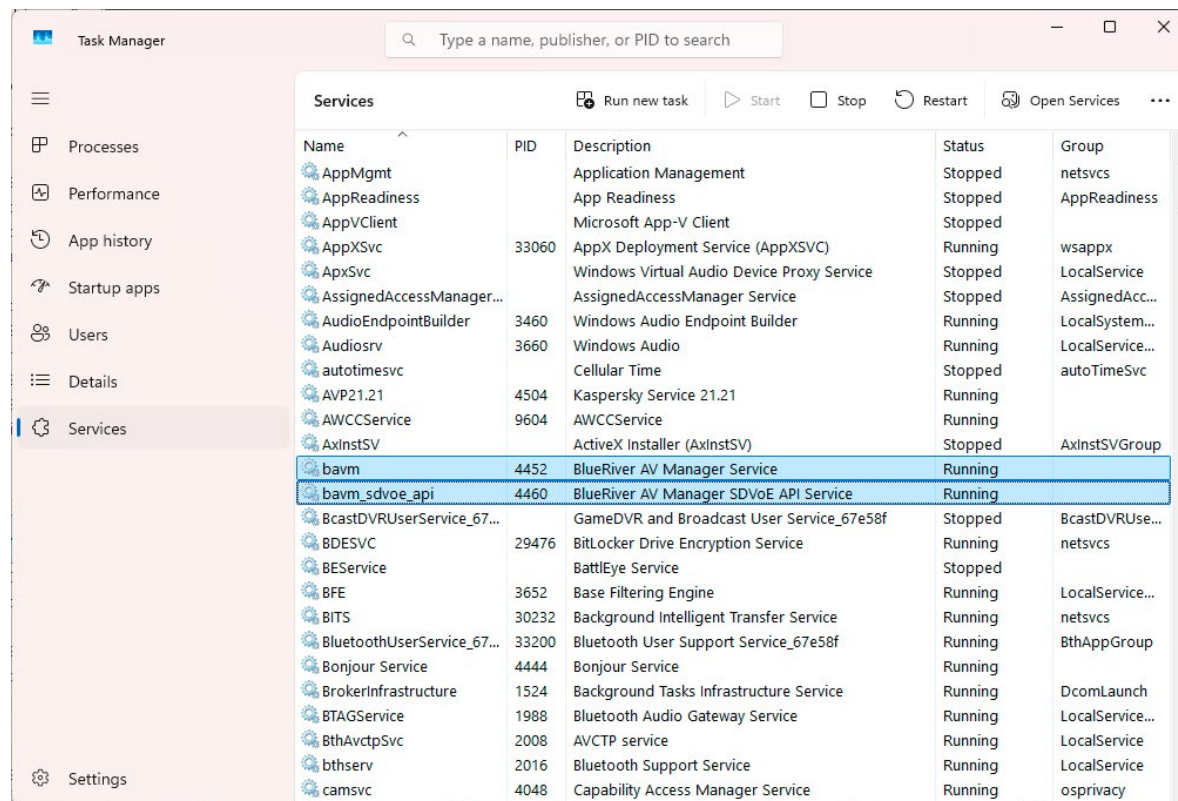
Step 1. Launch the Task Manager in the Windows operating system.

Step 2. Navigate to the Services menu.

Step 3. Search these two services:

- bavm - BlueRiver AV Manager Service
- bavm_sdvoe_api - BlueRiver AV Manager SDVoE API Service

Step 4. Select both lines and click on the ☐ **Stop** button to terminate them.



BlueRiver AV Manager services are running under Windows

3

LARA - System Configuration Guide

The chapter guides through the AV system configuration steps one by one from powering up to the matrix controlling via user interface. The following sections can be found in the chapter:

- ▶ [FIRST STEPS](#)
- ▶ [CONNECTING TO THE CTU](#)
- ▶ [ACCESSING LARA](#)
- ▶ [CONFIGURATION OF LARA FOR TPN MATRIX](#)
- ▶ [CONFIGURATION OF LARA FOR ICRON USB 2.0](#)
- ▶ [CUSTOMIZATION OF THE UI](#)
- ▶ [TROUBLESHOOTING](#)

3.1. First Steps

The following section describes the startup of the TPN / OPTN matrix system step-by-step from the beginning to the displayed video on the sink devices.

ATTENTION! Before powering on the system, please check the **Connecting Steps** section in the [TPN / OPTN user manual](#) to be sure all installation steps meet the connection requirements.

Switch on the Devices

Power on the network switch, the CTU, all the transmitters and the receivers, the source and the sink devices as well.

3.2. Connecting to the CTU

Establish the connection between the Control Unit (CTU) and the controller device. There are three ways to connect to the CTU:

- **LAN** (Ethernet 1, 2 and 3 ports; or via the network switch)
 - The factory default network setting is **dynamic IP address (DHCP)**. See the details of how the device can be discovered on the network or change the IP address in the **Establishing the Connection** section in the [TPN / OPTN user manual](#).
- **Serial communication** (RS-232 1 and 2 ports)
 - The factory default setting of the RS-232 port is **9600 BAUD 8N1**.

ATTENTION! User Ethernet is also transmitted over the SFP+ interface, so be sure not to make a network loop.

INFO: In the case of LAN connection: the CTU can be accessed via the Ethernet ports of the endpoint devices as well.

3.3. Accessing LARA

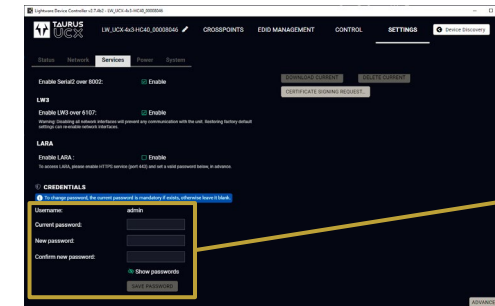
Short Steps

- Step 1.** Setting the password in the CTU.
- Step 2.** Enabling port #443 (factory default state is enabled).
- Step 3.** Enabling LARA in the device.
- Step 4.** Opening LARA in a browser.

Detailed steps

Step 1 – Setting the Credentials in the CTU

- Start the **LDC software** and connect to the device or open the **web LDC** in a browser by typing the IP address of the device into the address/URL bar.
- Navigate to the **Services** tab.
- **Set a password** for the user 'admin' (if not set previously).



CREDENTIALS

To change password, the current password is mandatory if exists, otherwise leave it blank.

Username:

admin

Current password:

New password:

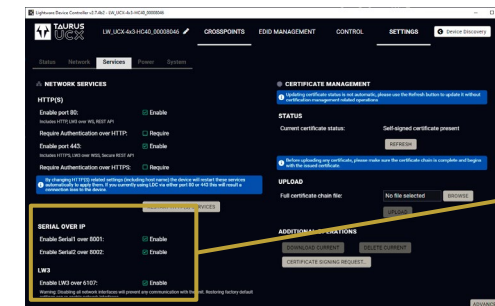
Confirm new password:

Show passwords

SAVE PASSWORD

Step 2 – Enabling Port #443

- Navigate to the **Services** tab in LDC or in web LDC.
- See the Network services section and mark the **Enable port 443** setting (factory default state is **enabled**).



HTTP(S)

Enable port 80:

Enable

Includes HTTP, LW3 over WS, REST API

Require Authentication over HTTP:

Require

Enable port 443:

Enable

Includes HTTPS, LW3 over WSS, Secure REST API

Require Authentication over HTTPS:

Require

By changing HTTP(S) related settings (including host name) the device will restart these services automatically to apply them. If you currently using LDC via either port 80 or 443 this will result a connection loss to the device.

RESTART HTTP(S) SERVICES

Step 3 – Enabling LARA in the Device

- Navigate to the **Services** tab in LDC or web LDC.
- See the Network services section and mark the **Enable LARA** (factory default state is **disabled**).



Step 4 – Opening LARA in a Browser

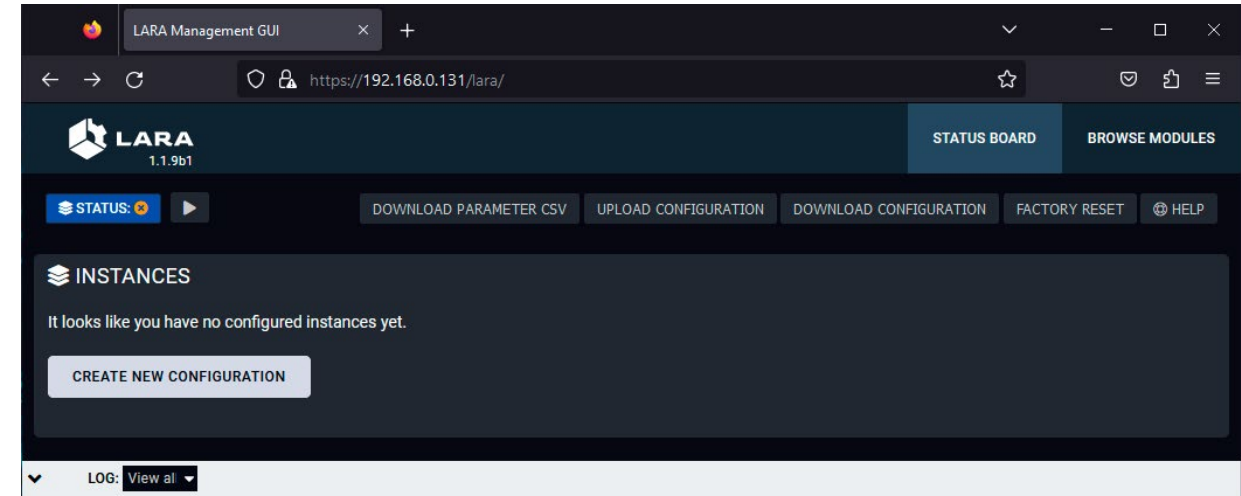
- Open a **web browser** and type the following in the address line: (make sure to type **https**)

`https://<IP_address>/lara`

TIPS AND TRICKS: If you select the **System** tab, you can open LARA with the **OPEN LARA** button.

After that you have to enter the user **admin** and the set **password**.

INFO: When opening LARA for the first time or after factory reset, you must accept the **disclaimer statement**.



The Status Board of LARA – in Factory Default State

3.4. Configuration of LARA for TPN Matrix

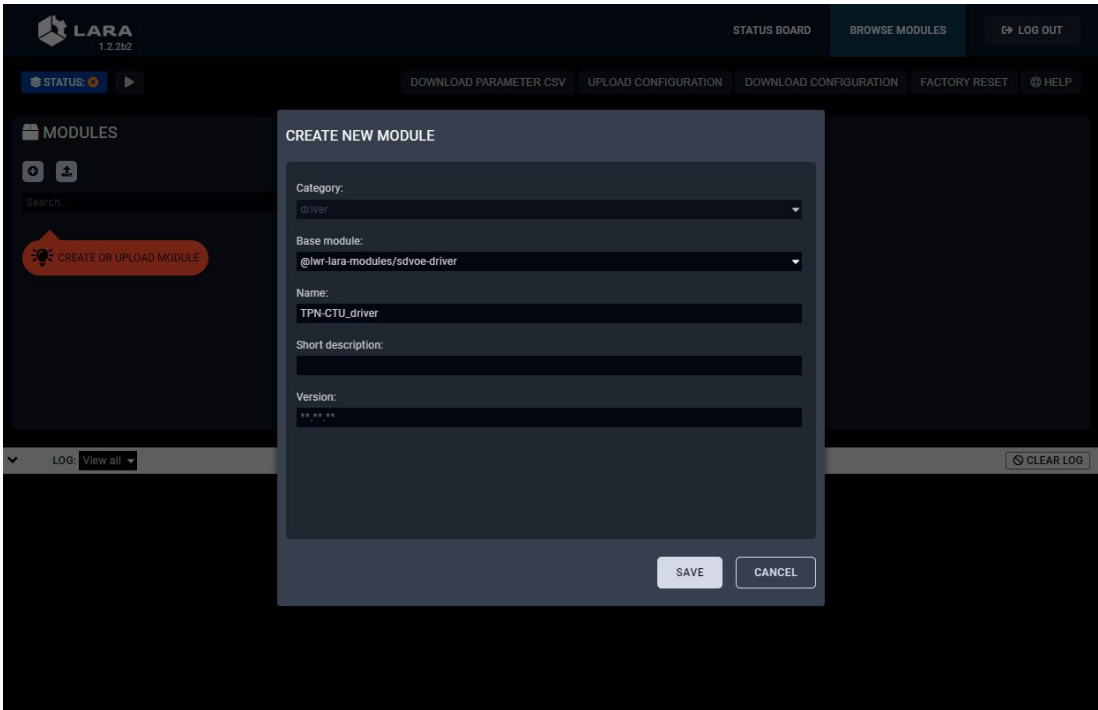
Configuration Steps in a Nutshell

- Step 1. Creating the **Driver** module.
- Step 2. Creating an **Instance** from the driver module.
- Step 3. Creating the **Userpanel** module.
- Step 4. Creating an **Instance** from the userpanel module.
- Step 5. Starting of the LARA services.
- Step 6. Configuration of the UI.

3.4.1. Creating the Driver Module

At first, LARA needs a driver module to handle the SDVoE extenders.

- Step 1. Select the **Create New Configuration** button.
- Step 2. Click on the **+** button to create a new module.
- Step 3. Fill out the parameters of the new module as follows:
 - Category: **driver**
 - Base module: **@lwr-lara-modules/sdvoe-driver**
 - Name: a custom name.



Step 4. Click on the **Save** button to complete the driver module.

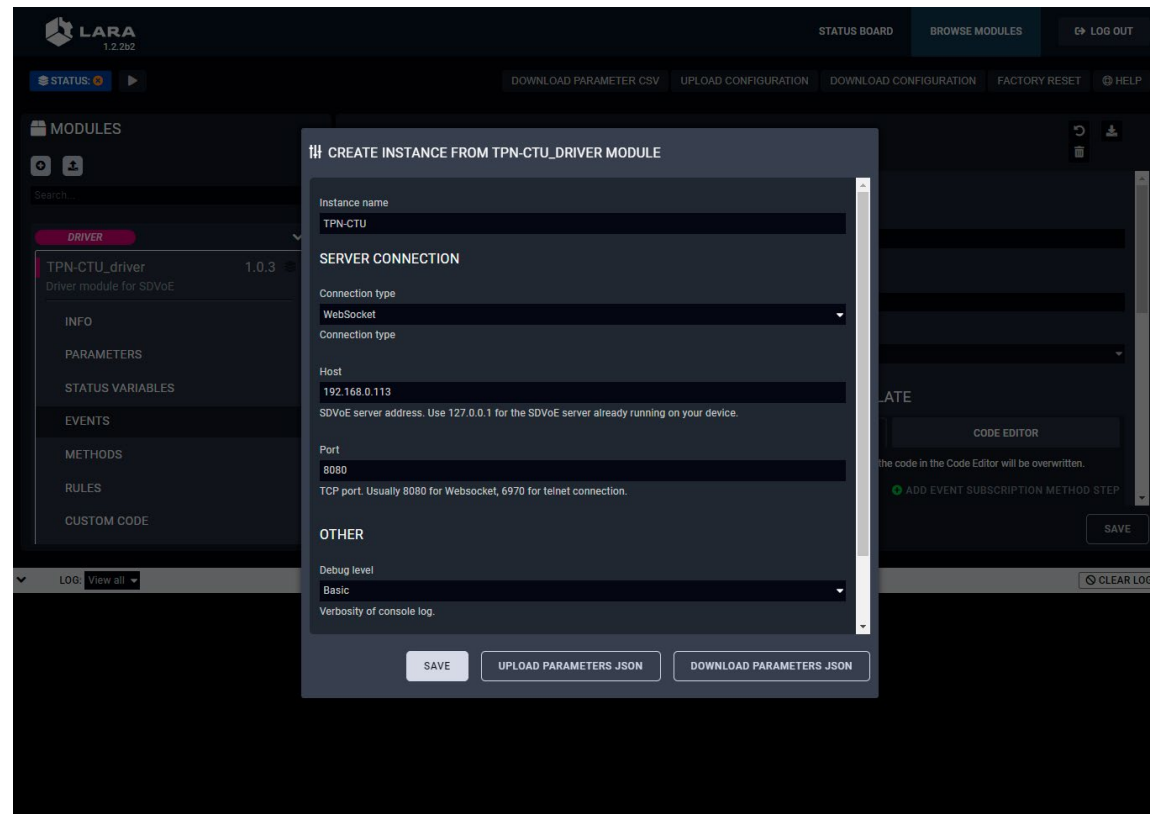
3.4.2. Creating an Instance from the Driver Module

An instance is needed to be created from the driver module as follows:

Step 1. Click on the button to create an instance.

Step 2. Fill out the parameters of the new instance as follows:

- Instance name: a custom name (**note down the given name**, it must be used later)
- Connection type: **WebSocket**
- Host: **the IP address of the TPN-CTU-X50**
- Port: **8080**



Step 3. Click on the **Save** button to complete the instance.

3.4.3. Creating the Userpanel Module

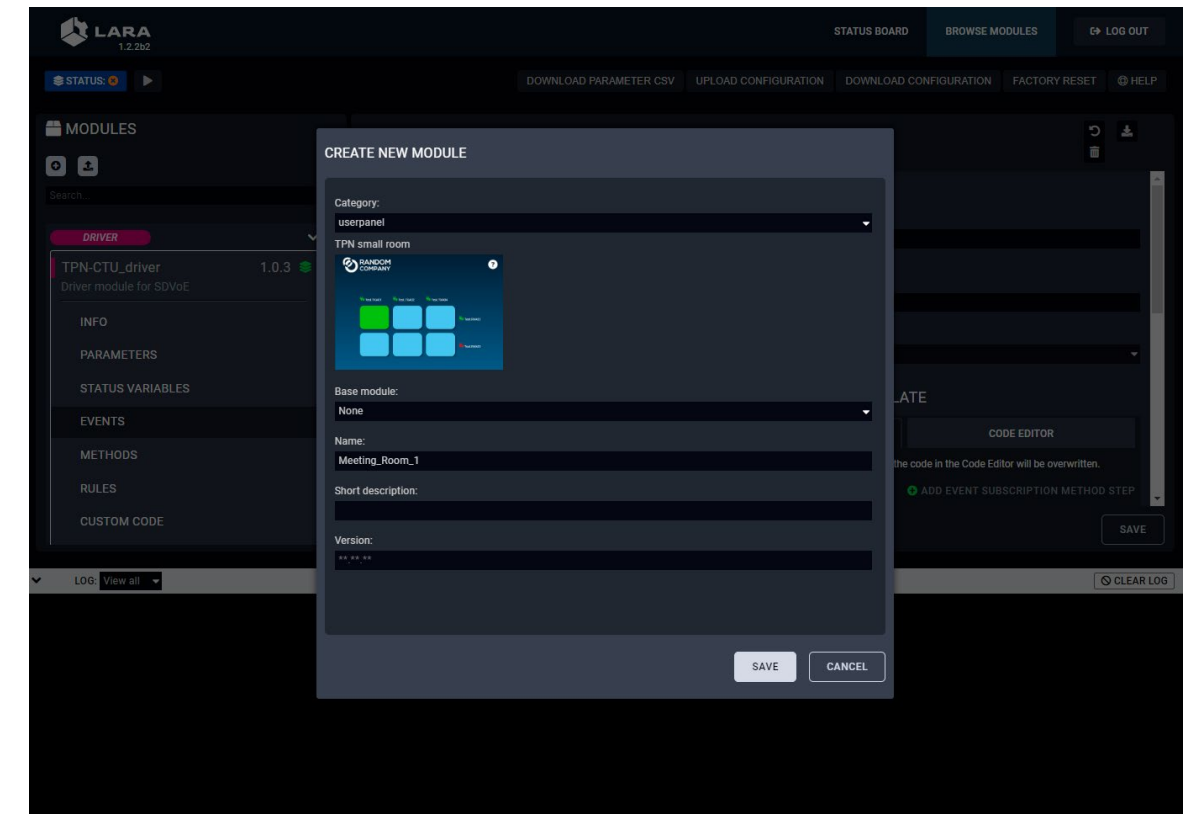
Users need a UI where the crosspoint settings can be managed.

Step 1. Navigate to the **Browse Modules** tab.

Step 2. Click on the button to create a new module.

Step 3. Fill out the parameters of the new module as follows:

- Category: **userpanel**
- Template: **TPN small room**
- Name: a custom name



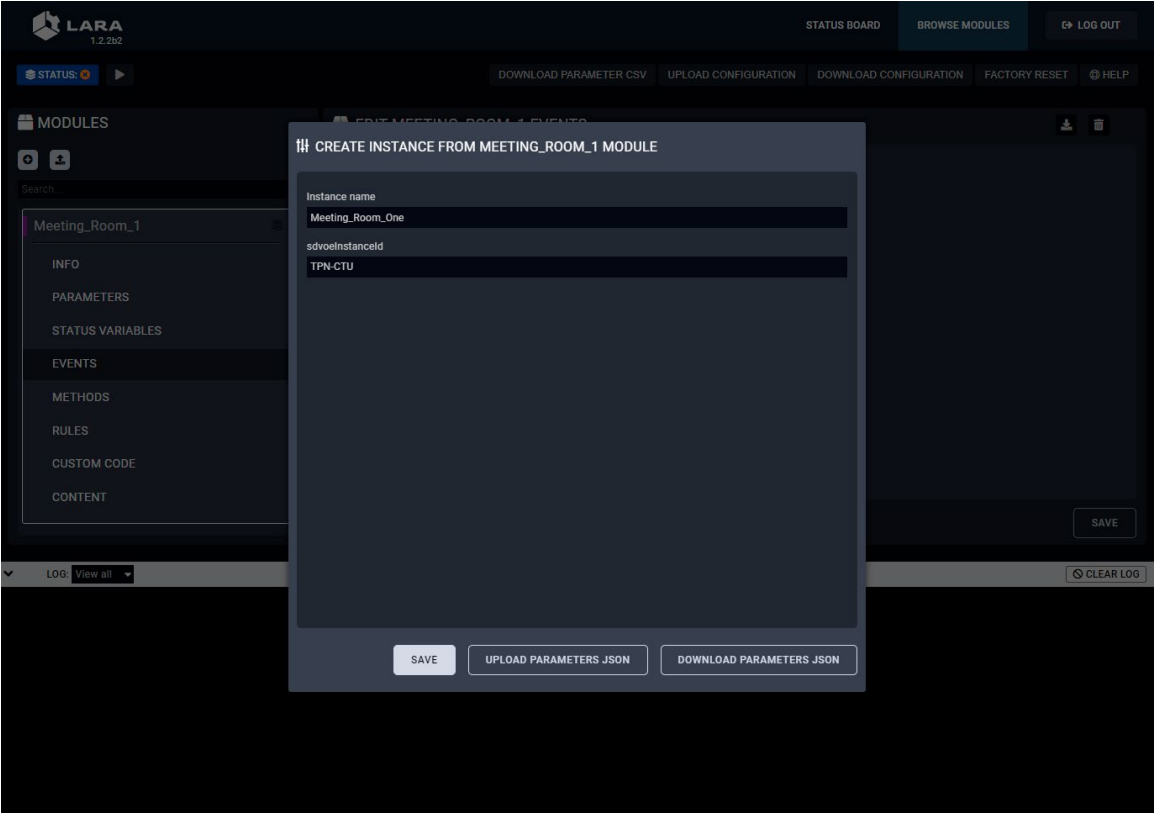
Step 4. Click on the **Save** button to complete the driver module.

TIPS AND TRICKS: The built-in templates can be customized freely and users can add new buttons and features to the touchpanel as well. See more personalization samples in the [Customization of the UI](#) section.

3.4.4. Creating an Instance from the Userpanel Module

An instance is needed to be created from the userpanel module as follows:

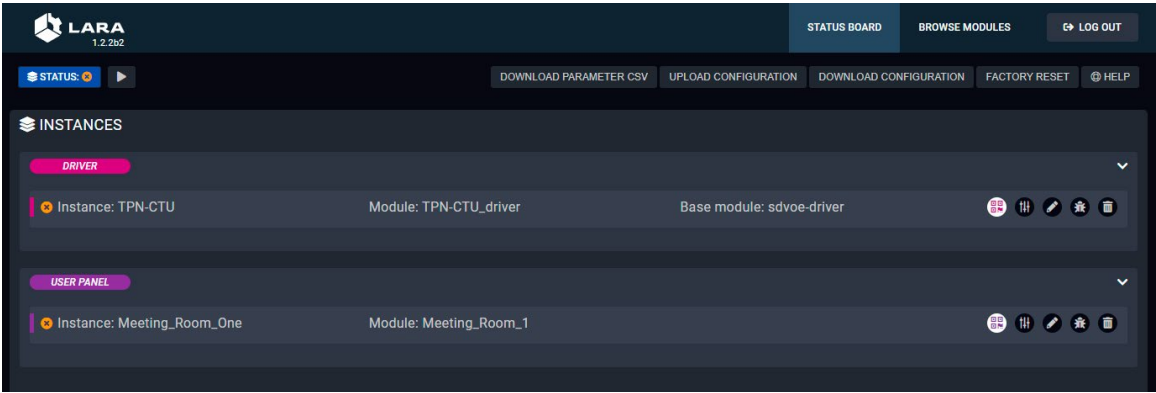
- Step 1. Click on the  button to create an instance.
- Step 2. Fill out the parameters of the new instance as follows:
 - Instance name: a custom name
 - sdvoelInstanceId: the name of the instance created from the driver



Step 3. Click on the **Save** button to complete the instance.

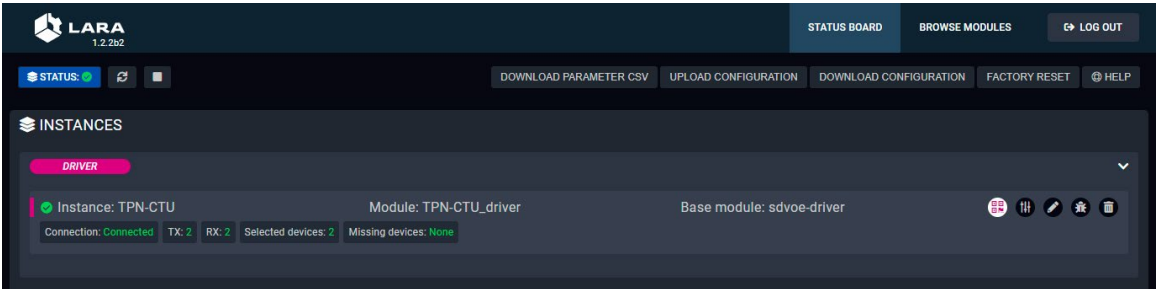
3.4.5. Starting of the LARA Services

LARA services need to be started by clicking on the  button.



Services are not running in LARA

After the services start to run, the status board displays the details of the services.



Services are running in LARA

The services can be stopped by clicking on the  button and they can be restarted by clicking on the  button.

Information on the Driver Status Board

Field	Value	Description
Connection	Connected	TPN-CTU-X50 is online and managed by the LARA services.
	Disconnected	TPN-CTU-X50 is not connected to the LARA services.
TX	<number>	Number of the connected TPN/OPTN series transmitters.
RX	<number>	Number of the connected TPN/OPTN series receivers.
Selected devices	<number>	Number of selected TPN/OPTN series extenders in the Icron UI panel.
Missing devices	<number>	Number of selected TPN/OPTN series extenders in the Icron UI panel that are disconnected from the network.

3.4.6. Configuration of the UI

Accessing of the TPN GUI

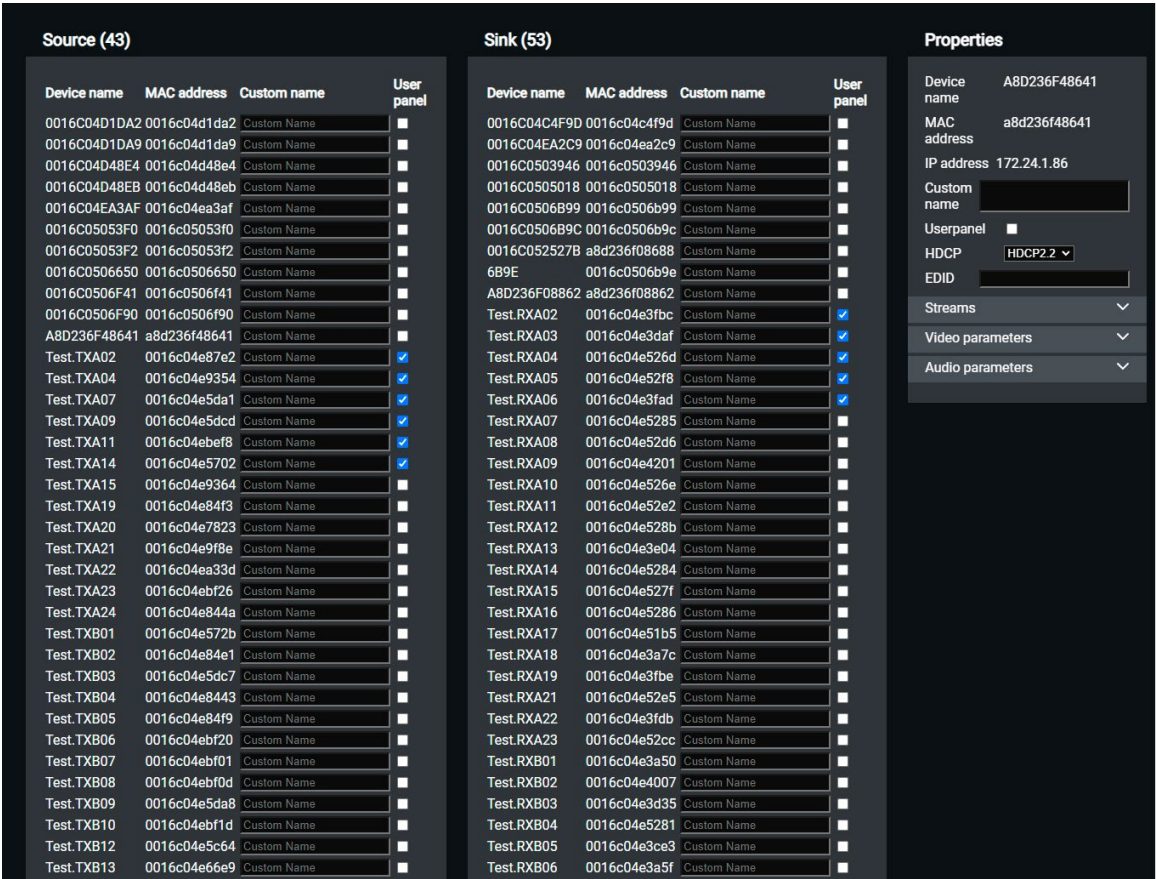
Click on the  icon in the status board of the TPN driver.



QR code and the direct link to the UI of the driver

Configuration of the Endpoint Devices with TPN GUI

Open the link that guides to the TPN GUI where the transmitters (Source) and the receivers (Sink) can be managed and the crosspoint table can be built.



TPN GUI window

Description

The default **device name** of the endpoints is the ID of the device. It can be personalized in the **Custom name** field. Where the **User panel checkbox** is **ticked in**, the stream of the endpoint will appear on the Userpanel as a **crosspoint possibility**.

Select an endpoint device to check the parameters in the **Properties** section. The panel displays information about the transmitted audio/video stream and in case of scaling receivers (-SR models), the parameters of the scaler can be set. `#scaler #hdcp #edid #audio #mac #endpointmanagement`

INFO: Learn more about the scaler modes of the -SR series endpoint devices in the [TPN/OPTN user manual](#).

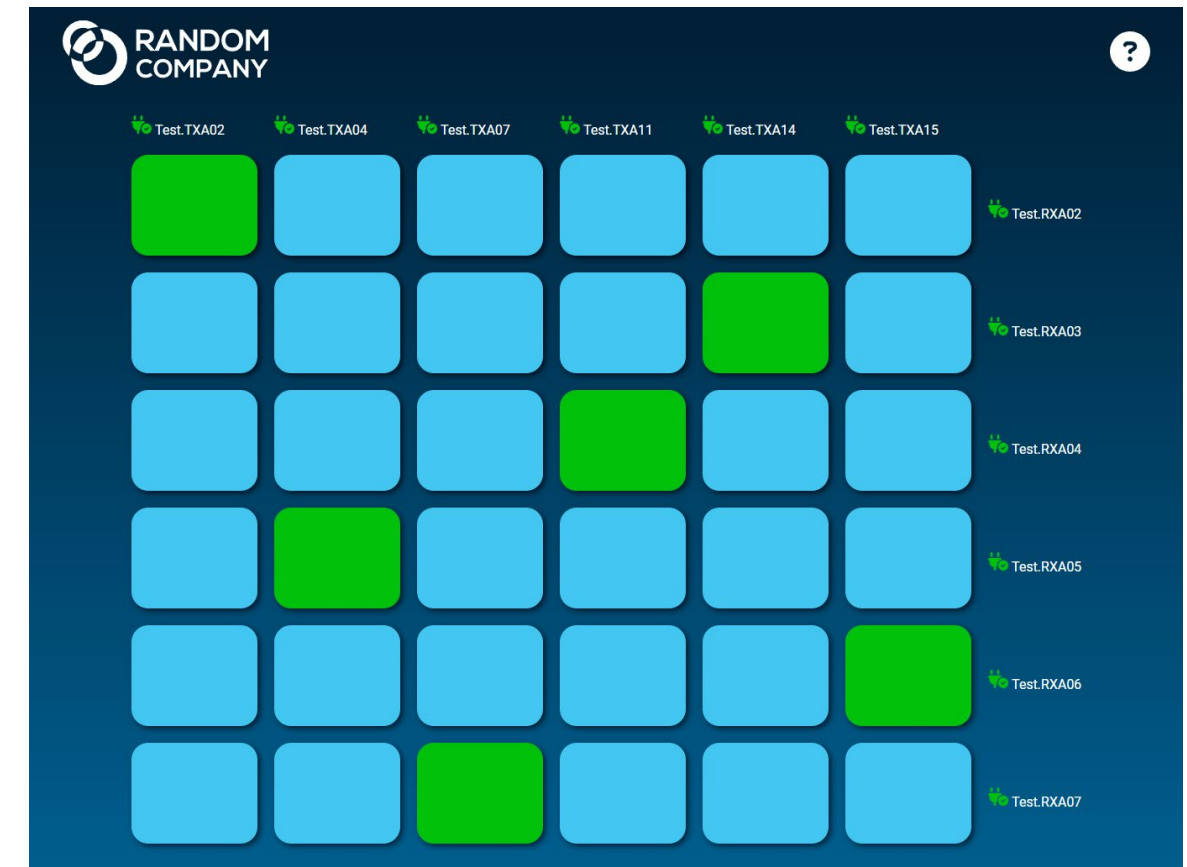
Crosspoint View on the TPN Userpanel

Click on the  icon in the status board of the TPN userpanel:



QR code and the direct link to the userpanel

Open the link that guides to the user panel where the crosspoint settings can be managed.



Crosspoint UI of a sample 6x6 TPN matrix system

#switch #crosspoint

3.5. Configuration of LARA for Icron USB 2.0

DIFFERENCE! The Icron USB 2.0 module of the LARA can be applied to the **U2K series** TPN and OPTN endpoint models and **USB20 series** extenders **only**.

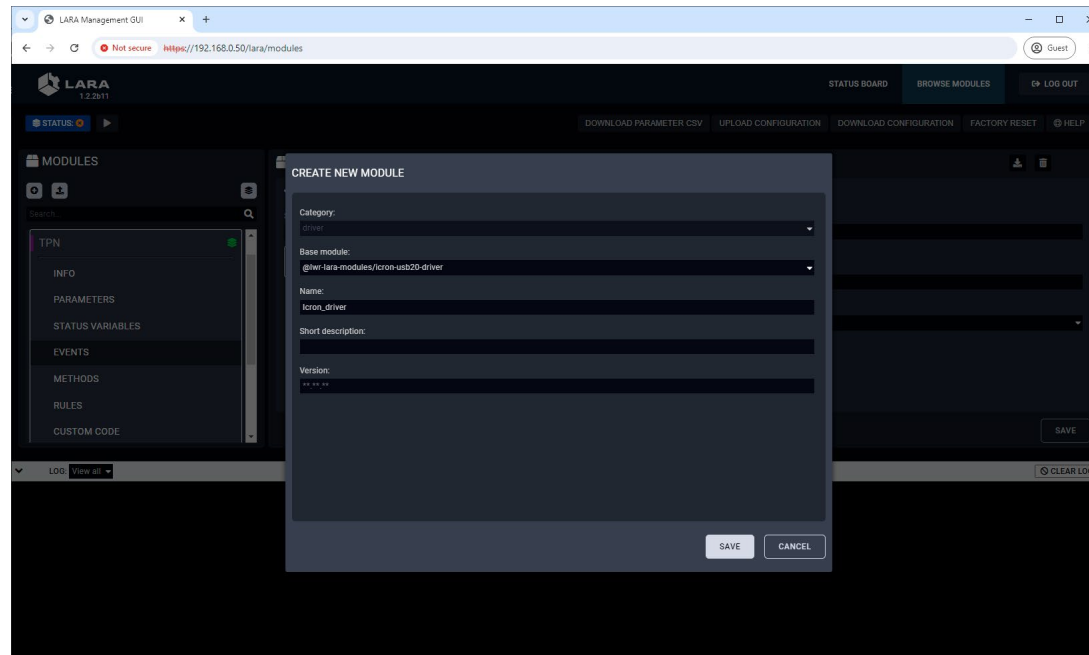
Configuration Steps in a Nutshell

- Step 1.** Creating the **Driver** module.
- Step 2.** Creating an **Instance** from the driver module.
- Step 3.** Creating the **Userpanel** module.
- Step 4.** Creating an **Instance** from the userpanel module.
- Step 5.** Starting of the LARA services.
- Step 6.** Configuration of the UI.

3.5.1. Creating the Driver Module

At first, LARA needs a driver module to handle the SDVoE extenders.

- Step 1.** Select the **Create New Configuration** button.
- Step 2.** Click on the **+** button to create a new module.
- Step 3.** Fill out the parameters of the new module as follows:
 - Category: **driver**
 - Base module: **@lwr-lara-modules/icron-usb20-driver**
 - Name: a custom name.



- Step 4.** Click on the **Save** button to complete the driver module.

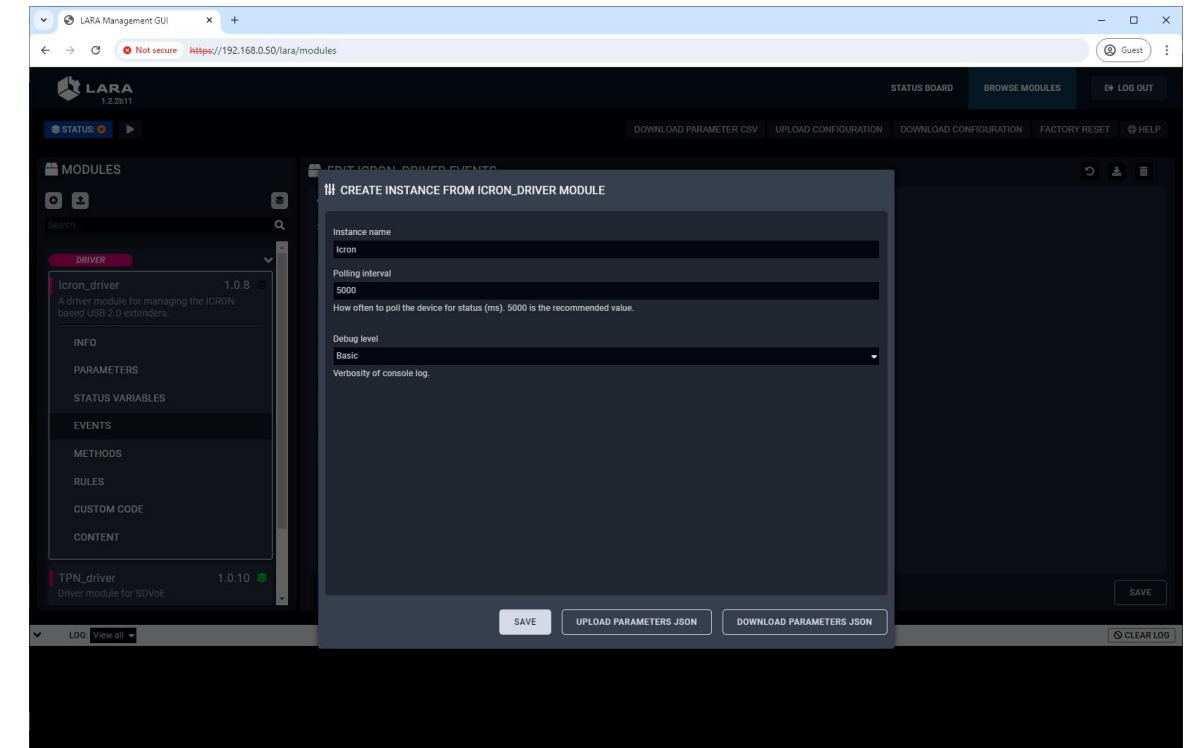
3.5.2. Creating an Instance from the Driver Module

An instance is needed to be created from the driver module as follows:

Step 1. Click on the button to create an instance.

Step 2. Fill out the parameters of the new instance as follows:

- Instance name: a custom name (**note down the given name**, it must be used later)



- Step 3.** Click on the **Save** button to complete the instance.

3.5.3. Creating the Userpanel Module

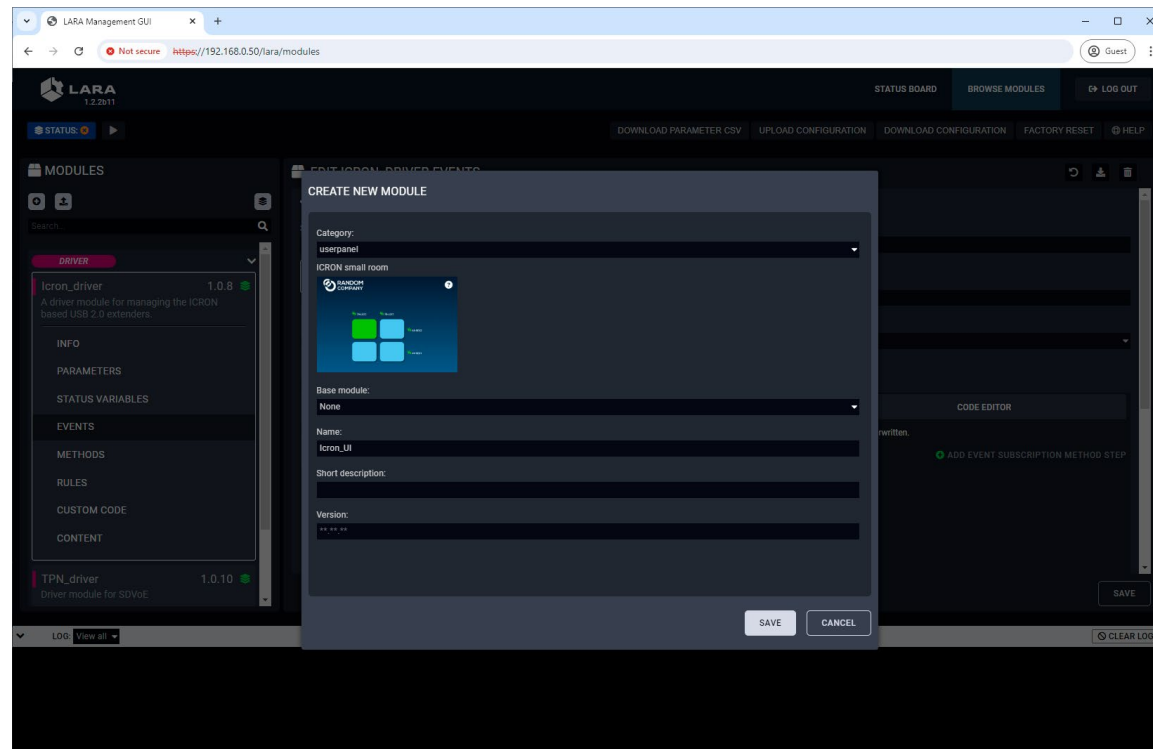
Users need a UI where the crosspoint settings can be managed.

Step 1. Navigate to the **Browse Modules** tab.

Step 2. Click on the **+** button to create a new module.

Step 3. Fill out the parameters of the new module as follows:

- Category: **userpanel**
- Template: **lcron small room**
- Name: a custom name



Step 4. Click on the **Save** button to complete the driver module.

TIPS AND TRICKS: The built-in templates can be customized freely and users can add new buttons and features to the touchpanel as well. See more personalization samples in the [Customization of the UI](#) section.

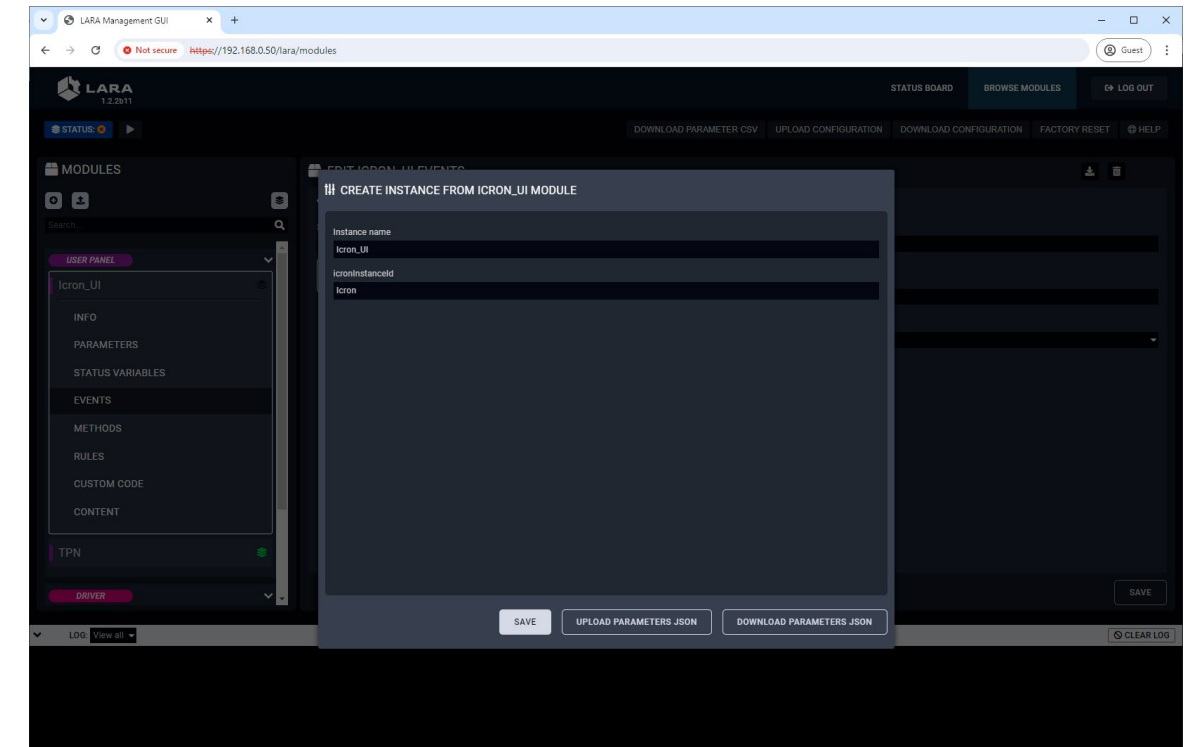
3.5.4. Creating an Instance from the Userpanel Module

An instance is needed to be created from the userpanel module as follows:

Step 1. Click on the **≡** button to create an instance.

Step 2. Fill out the parameters of the new instance as follows:

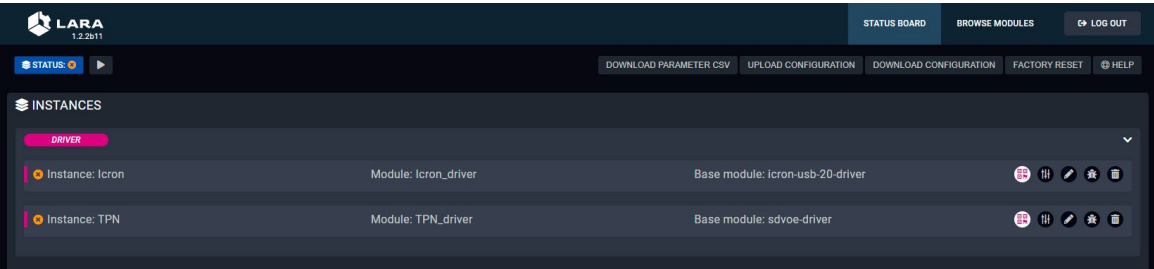
- Instance name: a custom name
- sdvoelInstanceId: **the name of the instace created from the driver**



Step 3. Click on the **Save** button to complete the instance.

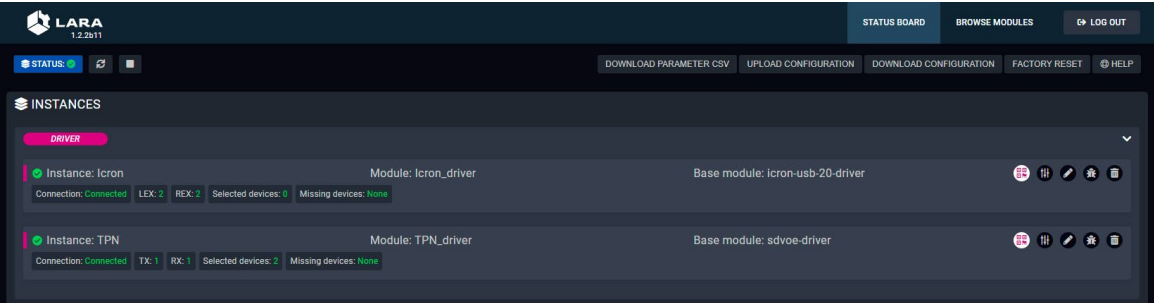
3.5.5. Starting of the LARA Services

LARA services need to be started by clicking on the  button.



Services are not running in LARA

After the services start to run, the status board displays the details of the services.



Services are running in LARA

The services can be stopped by clicking on the  button and they can be restarted by clicking on the  button.

3.5.6. Configuration of the UI

Accessing of the Icron GUI

Click on the  icon in the status board of the Icron USB 2.0 driver.



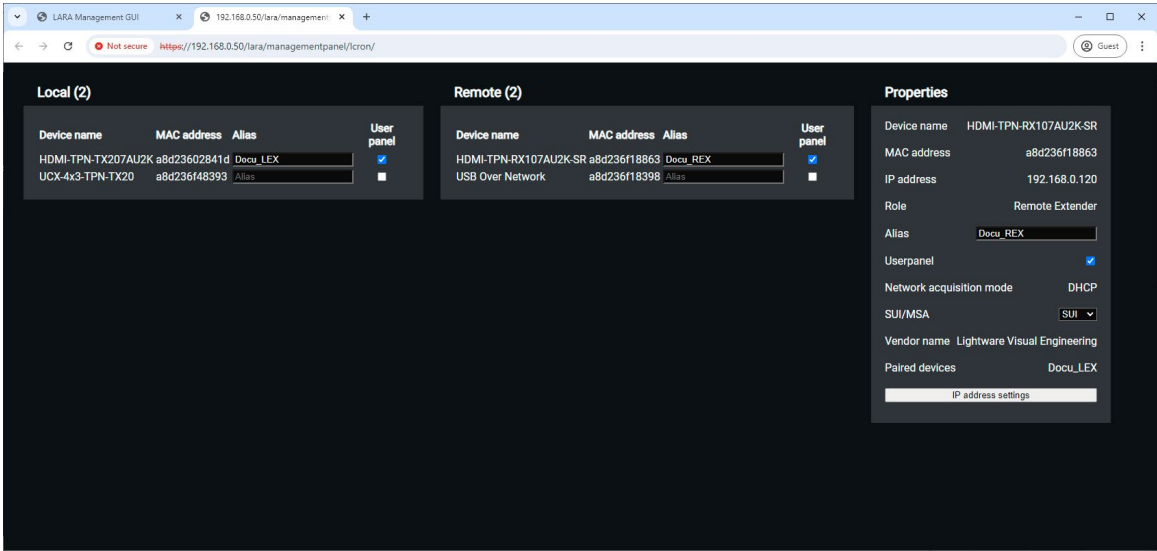
QR code and the direct link to the UI of the driver

Information on the Driver Status Board

Field	Value	Description
Connection	Connected	TPN-CTU-X50 is online and managed by the LARA services.
	Disconnected	TPN-CTU-X50 is not connected to the LARA services.
LEX	<number>	Number of the connected AU2K series transmitters.
REX	<number>	Number of the connected AU2K series receivers.
Selected devices	<number>	Number of selected AU2K series extenders in the Icron UI panel.
Missing devices	<number>	Number of selected AU2K series extenders in the Icron UI panel that are disconnected from the network.

Configuration of the Endpoint Devices with Icron GUI

Open the link that guides to the Icron GUI where the transmitters (Local - LEX) and the receivers (Remote - REX) can be managed and the crosspoint table can be built.



TPN GUI window

Description

The default **device name** of the endpoints is the ID of the device. It can be personalized in the **Custom name** field. Where the **User panel checkbox is ticked in**, the stream of the endpoint will appear on the Userpanel as a **crosspoint possibility**.

Select an endpoint device to check the parameters in the **Properties** section. The panel displays information about the IP address and MAC address of the Icron module in the AU2K series endpoint device. Alias can be defined for easier recognition in the crosspoint table.

SUI / MSA setting:

- **SUI** - Simultaneous Users Interaction - this mode is recommended in case of point-multipoint connections (multiple REXes are connected to a single LEX).
- **MSA** - Mass Storage Acceleration

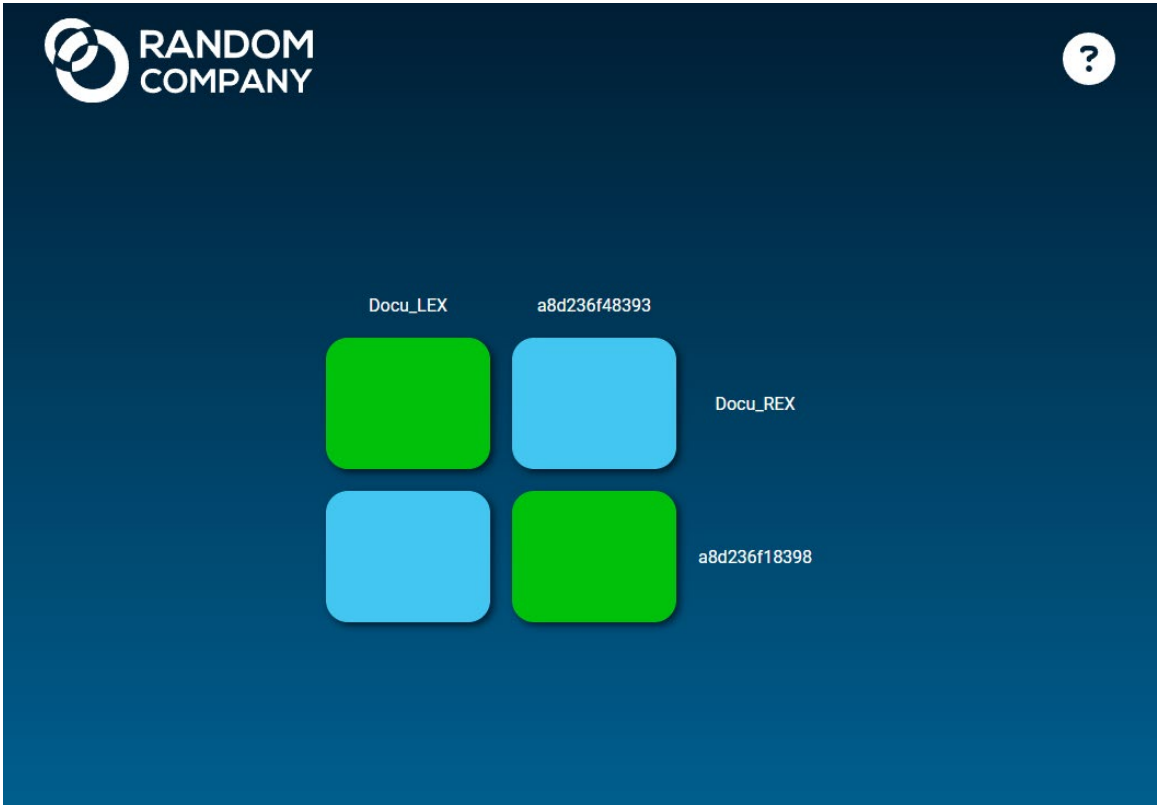
Crosspoint View on the Icron Userpanel

Click on the  icon in the status board of the Icron userpanel:



QR code and the direct link to the userpanel

Open the link that guides to the user panel where the crosspoint settings can be managed.



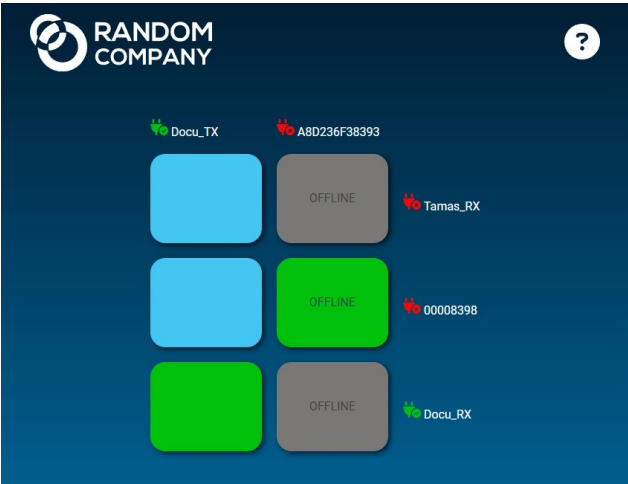
Crosspoint UI of a sample 2x2 Icron USB 2.0 matrix system

3.6. Customization of the UI

This section shows how to personalize the touchscreen UI templates through some simple examples. The TPN small room will be used in the following samples.

3.6.1. Changing the Company Logo

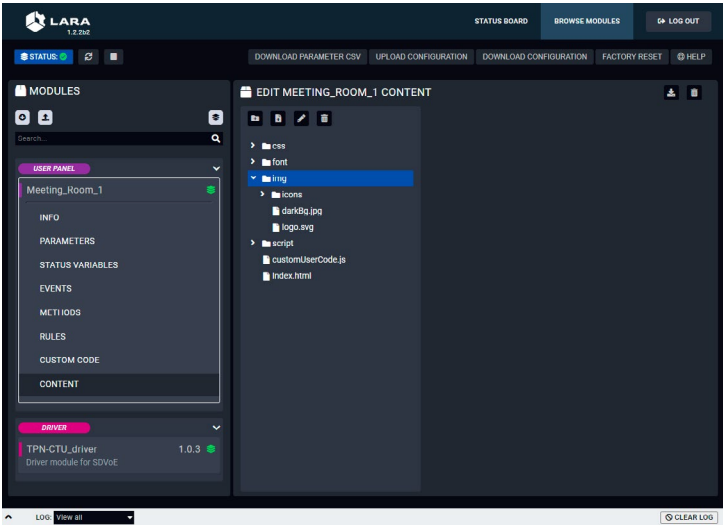
The template contains a logo which is displayed on the touchscreen UI by default. The logo can be changed easily in few simple steps.



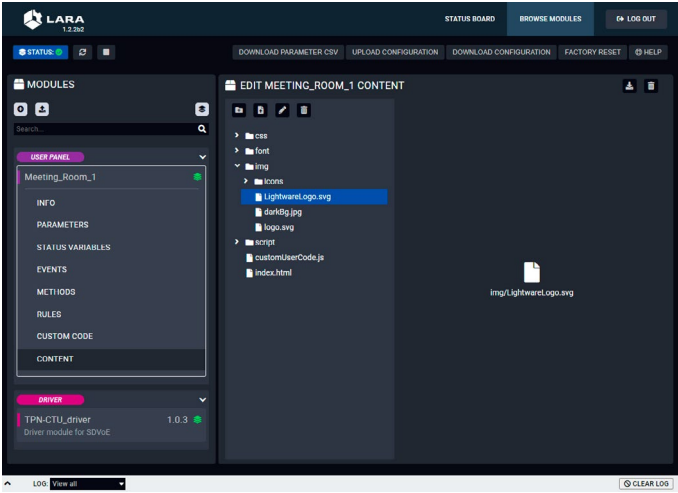
TPN small room template with the default company logo

Uploading the New Logo

Step 1. Edit the user panel module by clicking on the  icon. Go to the Content section and select the  img folder.



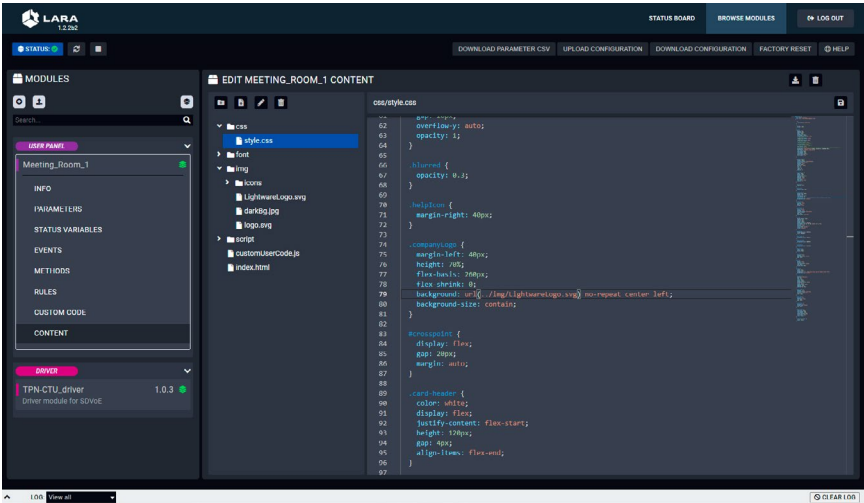
Step 2. Upload the new company logo to the **img** folder. Click on the **Upload file** icon. Browse the new logo file and upload it. Recommended file extension is **.svg**. In our example it is the LightwareLogo.svg.



Editing the CSS File

Step 3. Select the **style.css** and find the row in the code where the logo.svg is linked. In our example it is in row 79. Edit the link of the logo file to the new one like in our example:

Original code: `background: url (./img/logo.svg) no-repeat center left;`
Edited code: `background: url (./img/LightwareLogo.svg) no-repeat center left;`



Step 4. Save the style.css by clicking on the **Save file** icon.

Refreshing the Touchscreen UI

Step 5. Go the touchscreen panel UI and refresh it in the web browser. The new company logo appears immediately.



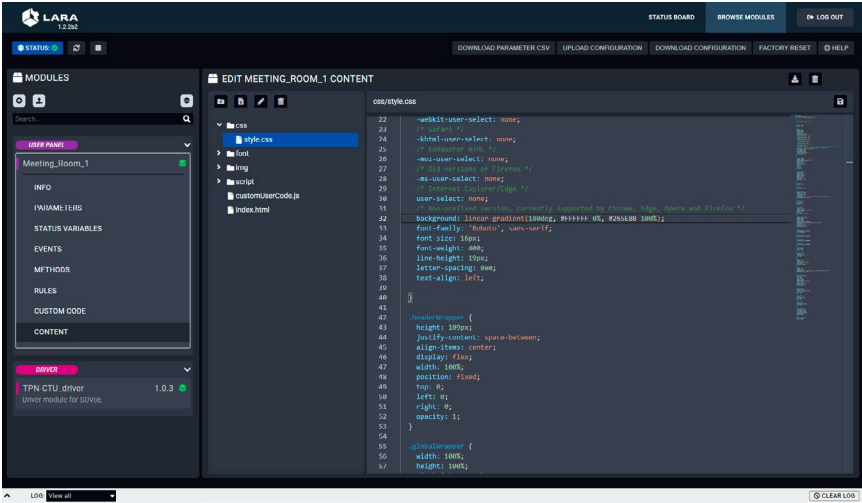
Touchscreen UI with the new company logo

3.6.2. Changing the Color of the Background

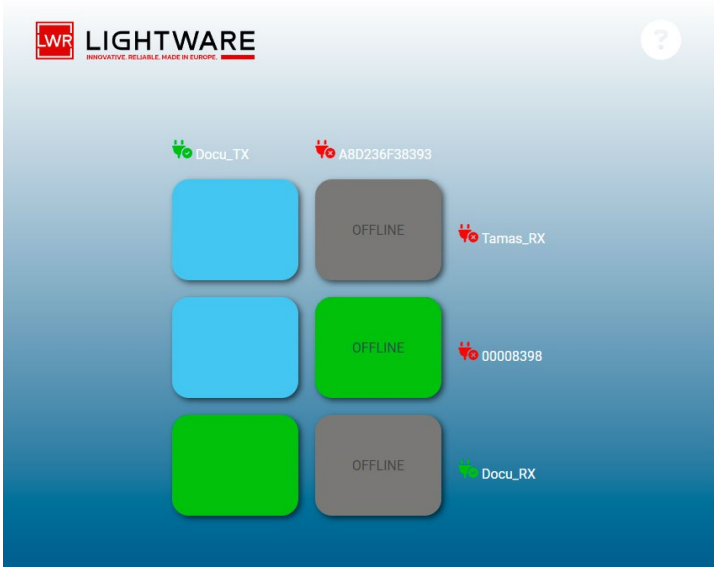
The next few steps show how to change the color of the background.

Editing of the CSS File

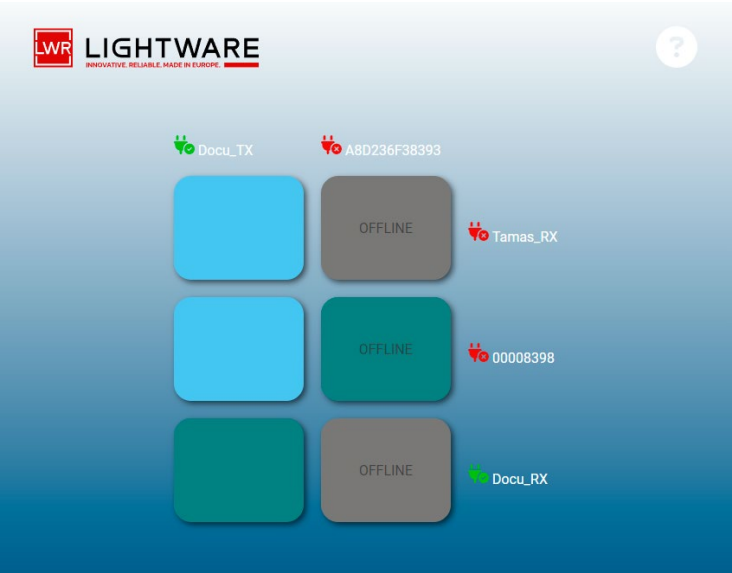
Step 1. Select the **style.css** and find the row in the code where the linear gradient background colors of the background are defined. In our example it is in row 32.



- Step 2.** Type the wished RGB code in hex after the # character or select it in the color picker pop-up window.
- Step 3.** Save the style.css by clicking on the **Save file** icon.
- Step 4.** Go the touchscreen panel UI and refresh it in the web browser. The new background color appears immediately.



Touchscreen UI with the re-designed background



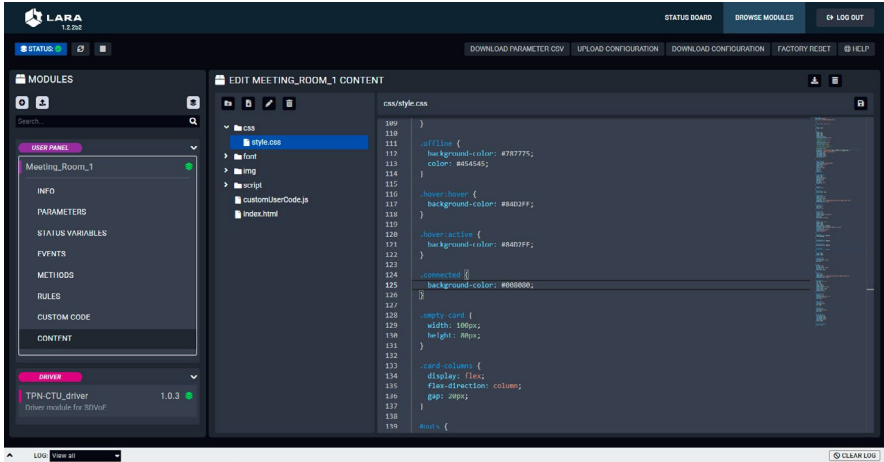
Touchscreen UI with the re-designed tiles

3.6.3. Changing the Color of the Crosspoint Tiles

The next few steps show how to change the color of the crosspoint cards in active state.

Editing of the CSS File

- Step 1.** Select the **style.css** and find the row in the code where the active state of the crosspoint card is defined. In our example it is in row 125.



3.7. Troubleshooting

3.7.1. Gateway Address Validation

If the IP address setting of the device is dynamic (DHCP), check the gateway address validation. If the gateway address is 0.0.0.0, the discovery process of the Icron USB 2.0 can be compromised.

TX Status

RX Status

Network

Services

⚙️ NETWORK SETTINGS

Current IP address:192.168.4.45

Current subnet mask:255.255.252.0

Current gateway address:192.168.7.254

Host name:lightware-DB385678

Obtain IP address automatically (DHCP, AutoIP):☒

Static IP address:192.168.0.100

Static subnet mask:255.255.255.0

Static gateway address:192.168.0.1

APPLY CHANGES

CANCEL



Valid gateway address

Status

Network

Services

System

⚙️ NETWORK SETTINGS

Current IP address:169.254.11.134

Current subnet mask:255.255.0.0

Current gateway address:0.0.0.0

Host name:lightware-00007344

Obtain IP address automatically (DHCP, AutoIP):☒

Static IP address:192.168.1.199

Static subnet mask:255.255.255.0

Static gateway address:192.168.1.5

APPLY CHANGES

CANCEL



Invalid gateway address

3.7.2. Icron Control over SDVoE Server

Make sure that the controlling Icron through SDVoE control server setting is disabled.

Status

Network

Services

System

🌐 BLUERIVER / SDVOE SETTINGS

SDVoE Control Server:☒ Enable

Control ICRON through SDVoE Control Server:☐ Enable

Blueriver / SDVoE settings in LDC

ATTENTION! Enabling this option might interfere with using the Icron LARA module and might prevent the ability to change SUI / MSA mode of the extenders.

4

BlueRiver AV Manager - Controlling over GUI

The chapter shows the BlueRiver AV Manager software from Semtech, which provides easy access and control features to the TPN or OPTN matrix over a user-friendly GUI.

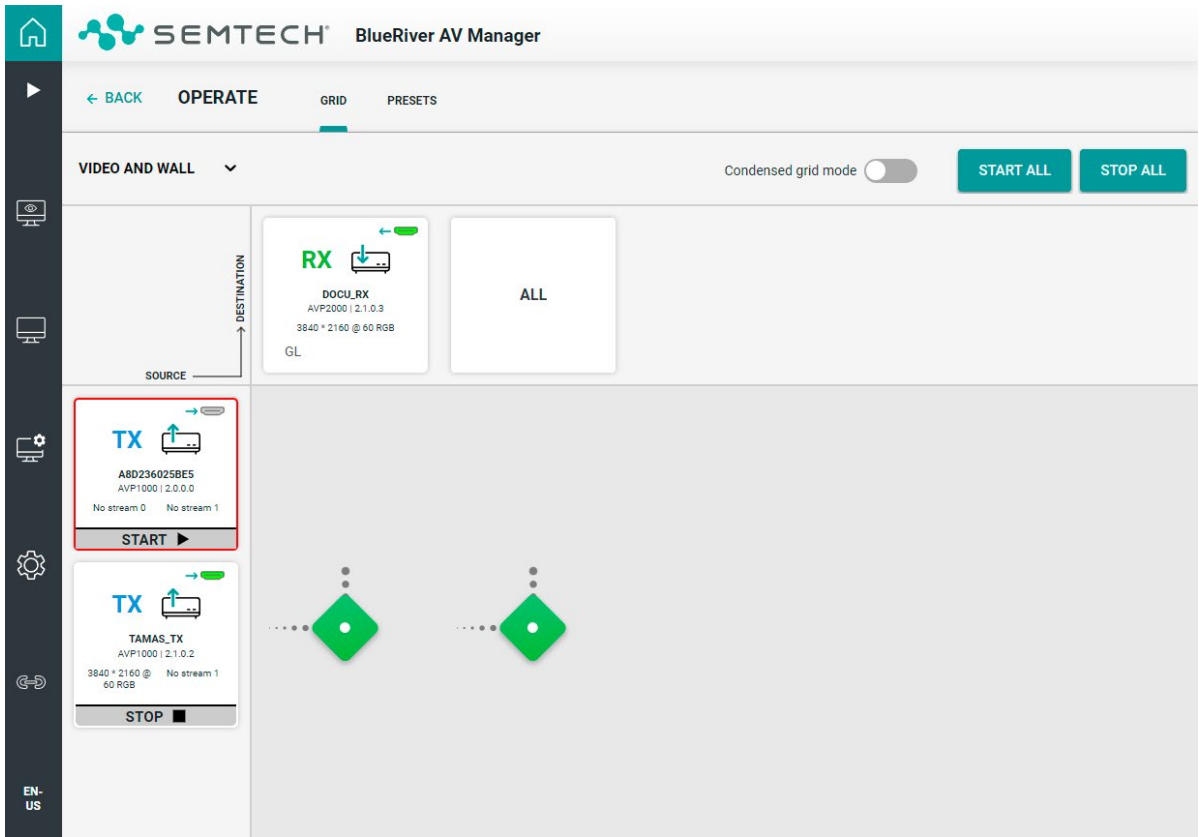
- ▶ [INTRODUCTION](#)
- ▶ [INSTALLATION](#)
- ▶ [HOME PAGE](#)
- ▶ [MONITORING MENU](#)
- ▶ [OPERATE MENU](#)
- ▶ [CONFIGURE MENU](#)
- ▶ [SETTINGS MENU](#)
- ▶ [CONNECTION VIA MOBILE DEVICE](#)

4.1. Introduction

ATTENTION! In case of using BlueRiver AV Manager, user needs a computer which runs the SDVoE server and the software itself. In this case the TPN-CTU-X50 controller unit presence on the network is not recommended because of the running more than one SDVoE servers.

BlueRiver AV Manager controls and manages AV distribution systems and endpoints that utilize Semtech's BlueRiver technology platform, including all devices that are compatible with the Software Defined Video over Ethernet (SDVoE™) standard. The easy-to-deploy, highly intuitive software provides manufacturers, resellers and users of AV equipment featuring BlueRiver a free and easily customizable baseline AV control software that can be repackaged to deliver a wide variety of highly targeted user interface (UI) experiences. In addition to standard video and audio, the software can be used to establish and control Video Walls layouts. Basic USB HID and USB 2.0 routing and switching is also supported (mouse, keyboard, webcam, etc.). *

* Source: <https://info.semtech.com/blueriver-av-manager>



Crosspoint view in the BlueRiver AV Manager software

Functions

The following functions and features are available in the software:

- Endpoint configuration
- Device monitoring
- Crosspoint setting
 - Video and videowall
 - Audio
 - USB
 - RS-232
 - Infrared
- Video wall configuration

The official **user guide** can be downloaded from Semtech's website (PDF):

<https://info.semtech.com/blueriver-av-manager>

INFO: Downloading of the software and the user guide requires a **registration** on the Semtech's website.

4.2. Installation

The software can be downloaded from Semtech's website (<https://info.semtech.com/blueriver-av-manager>) for Windows and Linux operating systems.

INFO: After the installation the Windows and the Linux application have the same look and functionality.

The following ports are reserved by the software:

Protocol	Port Number	Description
TCP	5920	BlueRiver AV Manager Service
TCP	5936	BlueRiver AV Manager SDVoE API Service
TCP	20020	TCI API
TCP	5148	BlueRiver AV Manager

WARNING! If the BlueRiver AV Manager running on a PC and the Lightware's controller device (TPN-CTU-X50, etc) are on the same network, the presence of two (or more) BlueRiver Server might cause malfunctions on the SDVoE AV system. Learn more about it and how to prevent it in the [BlueRiver Servers on the Network](#) section.

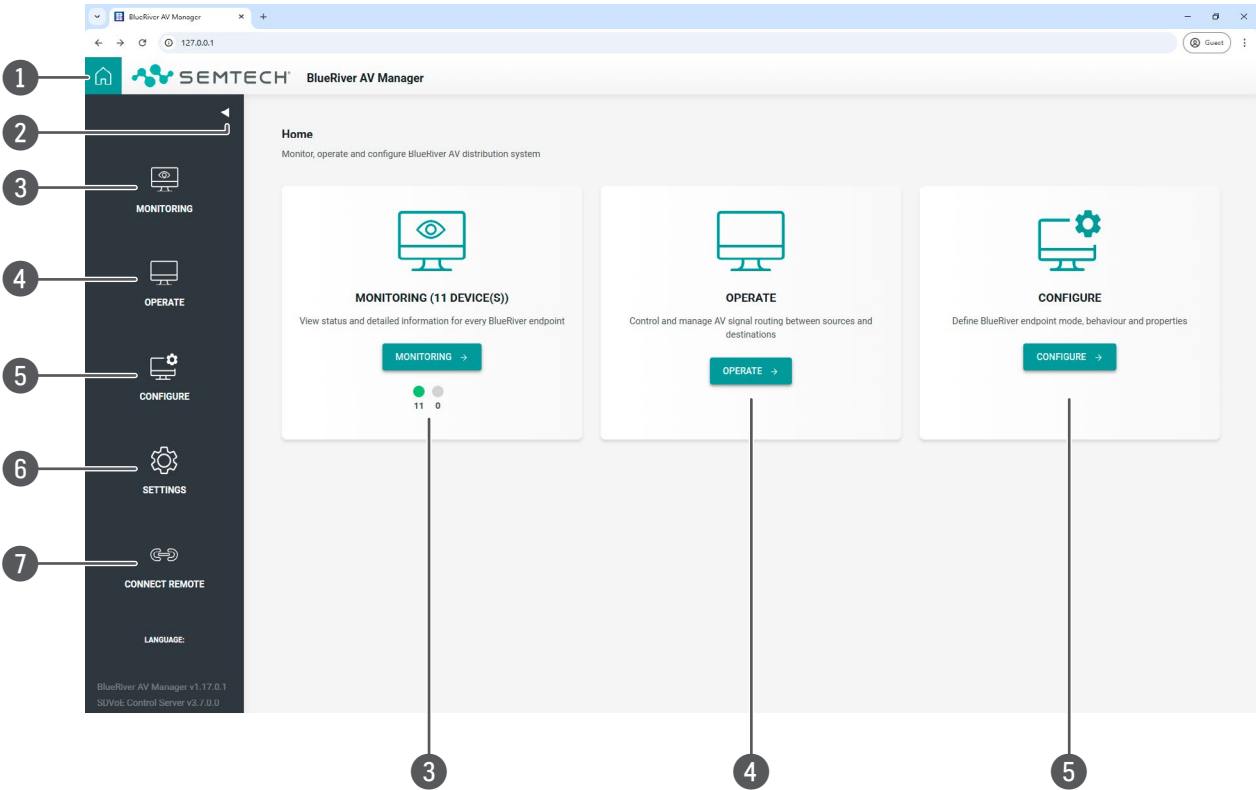
ATTENTION! The running BlueRiver services on the listed ports can obstruct the SDVoE device discovery functions of the **Lightware Device Updater V2 (LDU2)**. Read more details about it in the [Reserved Ports and Security Options](#) section.

4.2.1. Accessing the Software

After installation, the application GUI is accessed through a browser by simply entering the IP address of the computer where the application is installed in the address bar.

If the browser is being run from the same computer as the BlueRiver AV Manager is installed on, user can enter "localhost" or "http://127.0.0.1" instead of the computers IP address.

4.3. Home Page



Home page of BlueRiver AV Manager

After the application is opened, the home page launched firstly. The three main menu can be opened here in the front window: the [Monitoring Menu](#), the [Operate Menu](#) and the [Configure Menu](#), and further menus can be selected on the left side of the page.

Legend

- 1

Home page

The landing page of the BlueRiver AV Manager.
- 2

Navigation menu tool

The collapsible Navigation panel can be used to switch between the tools.
- 3

Monitoring menu

When opening either the Monitoring or the Configure tool, all known endpoints are displayed as device cards. The Configure tool also includes pages to configure Video Wall, Multiview, and firmware update. See more details in the [Monitoring Menu](#) section.

The number with the green dot means the available online endpoint devices; the number with the grey dot means the offline devices.
- 4

Operate menu

Control and manage AV and control signal routing between sources and destinations. The functions what are available in the tool:
 - **Video signal and video wall crosspoint settings** - see more details in the [Video and Wall](#) section.
 - **Audio crosspoint settings** - see more details in the [Audio](#) section.
 - **USB (KVM and Icron USB 2.0 as well) extension settings** - see more details in the [USB](#) section.
 - **RS-232 signal routing** - see more details in the [RS232](#) section.
 - **Infrared signal routing** - see more details in the [Infrared](#) section.
 - **CEC signal routing** - see more details in the [CEC](#) section.
- 5

Configure menu

The Configure menu contains the following tools:
 - **Devices** - Configuring the endpoint devices on the network. See more details in the [Devices](#) section.
 - **Wall** - Video wall configurator. See more details in the [Wall](#) section.
 - **Multiview** - Multiview configurator. See more details in the [Multiview](#) section.
 - **FW update** - Firmware updating of the endpoint devices. See more details in the [FW Update](#) section.
- 6

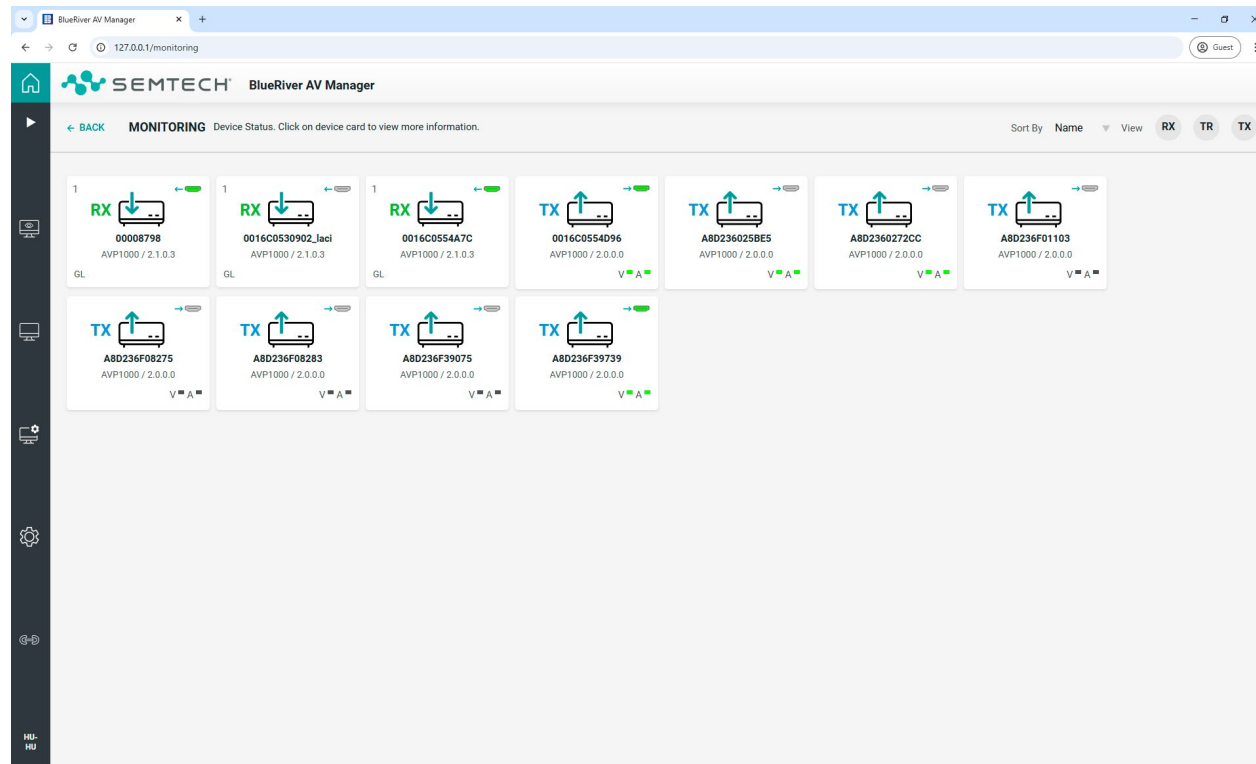
Settings

Application settings are accessed directly through the navigation panel. See more details in the [Settings Menu](#) section.
- 7

Connect remote

Selecting the menu provides accessing the software via a mobile device. See more details in the [Connection via Mobile Device](#) section.

4.4. Monitoring Menu



Device cards in the Monitoring menu

The available devices on the network are listed in the Monitoring menu one-by-one on **Device cards**. There are three submenu on each cards:

- **Config** - General and video/audio/control signal-related settings of the extender.
- **Details** - General and status information about the extenders.
- **Diagnostics** - Network-related diagnostic tool.

When a device card is opened from the Monitoring menu, the default submenu is the **Details**.

The following information can be found on the Device cards:

- **Device name:** it is managed through the Configuration tool.
- **Type:** There are two possible TPN / OPTN device types. TX (transmitter) that has an HDMI Input. RX (receiver) that has an HDMI output.
- **Chipset / Firmware version:** Lists the BlueRiver chipset used by the device as well as current firmware version.
- **Icon:** A configurable pictogram to emphasize the type of HDMI device that is connected to the endpoint.
- **HDMI connector:** Illustrates the HDMI connection status. Green indicates that cable is connected while grey indicates that there is no cable connected. The associated arrow indicates if the connector is an input or an output.
- **Color:** Online devices are white. Devices running on fallback golden firmware are yellow, while a greyed-out device indicates that the device is offline (powered off or disconnected from the network).
- **Output mode:** Applies only to RX devices, not applicable to TX devices. Devices that have an HDMI output can operate in the following output modes: **GL** (Genlock), **GL-S** (Genlock Scaling), **FS** (Fast Switch), **W-GL** (Wall-Genlock), **W-FS** (Wall-Fast Switch) and **MV** (Multiview). The output mode is configured using the Configuration tool.
- **Video stream:** Applies only to TX and TR devices, not applicable to RX devices. Up to two LEDs indicate video streaming status. The top LED represents the native video stream while the bottom LED indicates the scaled video stream. When the LED is green, the stream is started. When the LED is dark, the stream is stopped.

INFO: The BlueRiver AV Manager uses the native video stream except for Multiview mode, which utilizes the scaled stream.

- **Audio stream:** Up to three visible LEDs are present, which indicate the audio streaming status. The top LED represents the HDMI audio stream, the middle LED represents the Multichannel auxiliary audio stream, and the bottom LED represents the Stereo auxiliary audio stream. When the LED is green, the stream is started, and when dark, the stream is stopped. See more information about which LEDs are visible is dependent on the audio settings in the [Settings Menu](#) settings.
- **Video subscriptions:** Applies to RX and TR devices and indicates the number of active video subscriptions.

INFO: The device must be operating in Multiview mode for this number to be higher than 1.

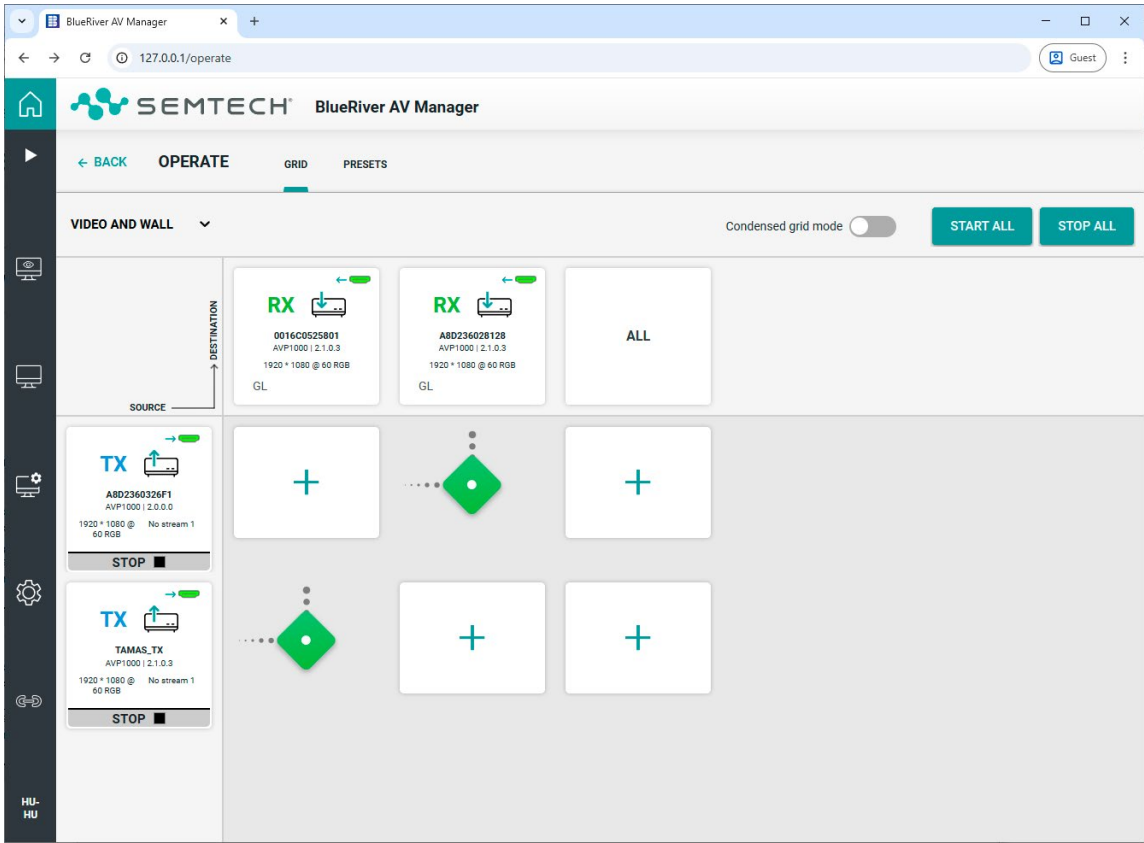
4.5. Operate Menu

The Operate tool is used to manage and control signal routing between sources and destinations.

4.5.1. Video and Wall

The Operate tool is used to manage and control signal routing between sources and destinations. This includes routing video between HDMI sources, such as PCs or video players, and HDMI displays such as monitors, TVs, and projectors.

Standard video routing is done using the **Video and Wall** page within the Operate tool.



Video and Wall submenu in the Operate menu

Endpoints connected to sources or displays are shown as device cards and form a routing grid, where devices connected to sources appear on the grid vertical axis, while endpoints connected to displays are shown on top. Active walls appear on the horizontal axis as individual displays and are used the same way as other video destinations. An **ALL** card is also shown on the horizontal axis, allowing video routing to all the destination devices.

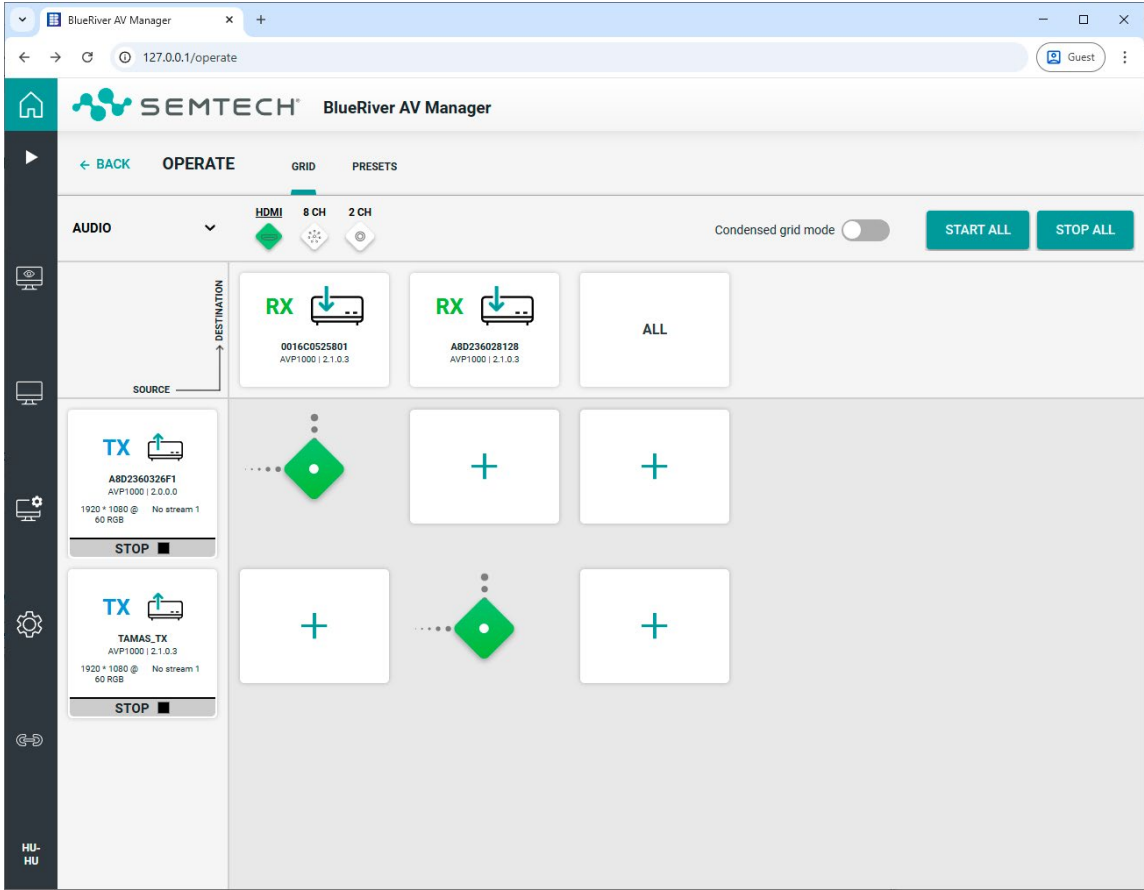
- To route video from a source to display, simply press the intersection point between the two.
- To remove the route, just press the intersection point again.
- To route the same signal to multiple displays, press corresponding intersection points along the horizontal axis.
- To route the same signal to all displays, press the intersection point between the corresponding source device and the “ALL” card.
- To switch the display to a different source, select the new intersection point along the vertical axis.

Sources and destinations are represented as device cards. The cards display various information and are also interacted with by clicking on them.

- **Source:** Represents the device connected to the source. Pressing it opens the device details.
- **Source resolution:** Resolution of connected video source and video streaming on the network.
- **Start / Stop:** Press to start or stop the video stream.
- **Border:** Red means no video source, yellow means video source stable but not streaming, no border means video source stable and streaming.
- **Display:** Represents the device connected to display. Pressing it opens the device details.
- **Output mode:** Can be one of the following, GL for Genlock mode, FS for Fast Switch mode and GL-S for Genlock scaling mode.
- **Output resolution:** Video resolution going to display.
- **Wall:** Represents a group of devices operating together in wall mode.
- **Wall Mode:** Can be one of the following, W-FS for Fast Switch Wall and W-GL for Genlock Wall. Pressing it prompts a message to unload the wall.

4.5.2. Audio

Up to three different audio streams can be controlled and routed using the Audio page within the Operate tool. In addition to the HDMI audio stream which is the audio that originates from an HDMI input source, BlueRiver/SDVoE endpoints can also have stereo and/or multichannel (up to 8-channels) auxiliary audio IOs. Audio signals are routed and controlled using the same kind of matrix grid as for video, where audio sources are displayed on left and audio sinks are arranged on the top.

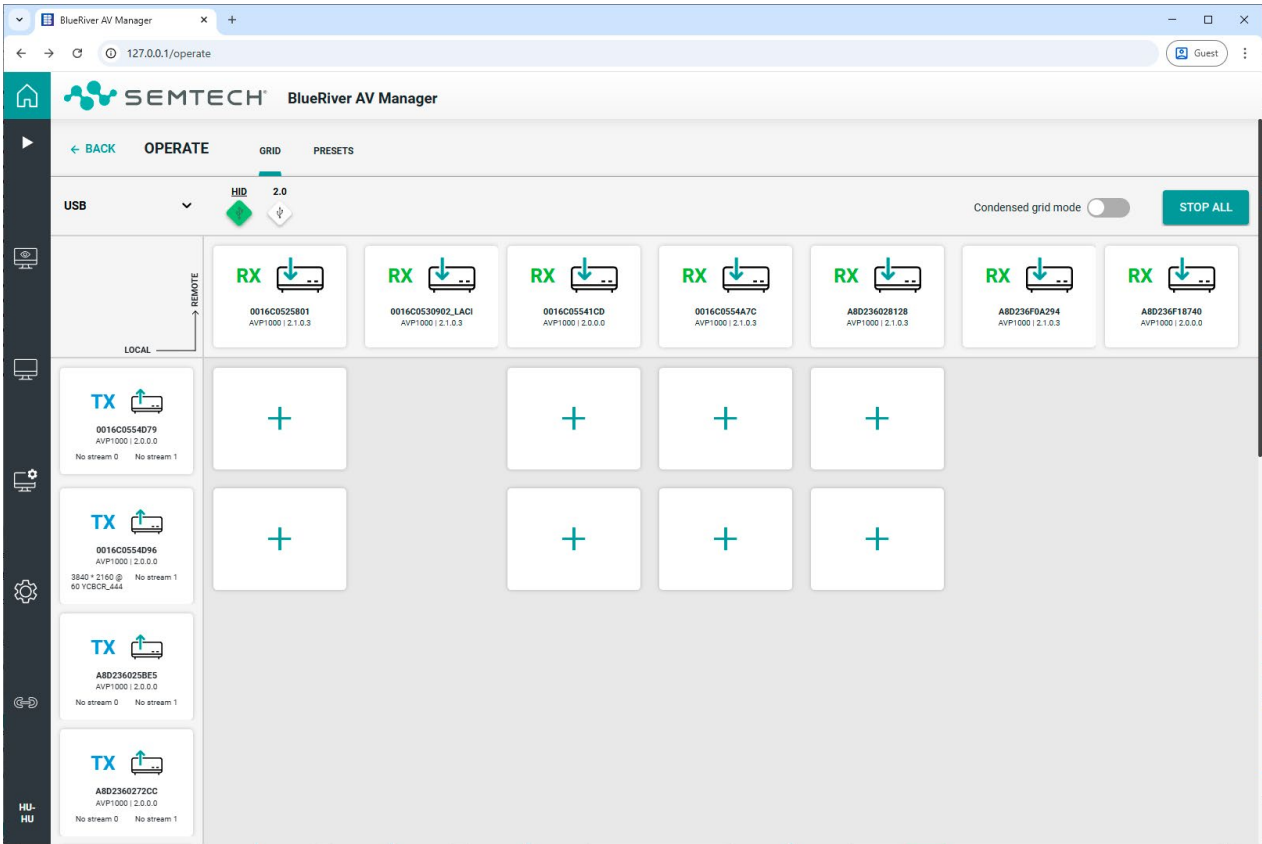


Audio submenu in the Operate menu

4.5.3. USB

DIFFERENCE! The USB submenu is related to the -U2K series TPN and OPTN endpoint models and USB20 series extenders only.

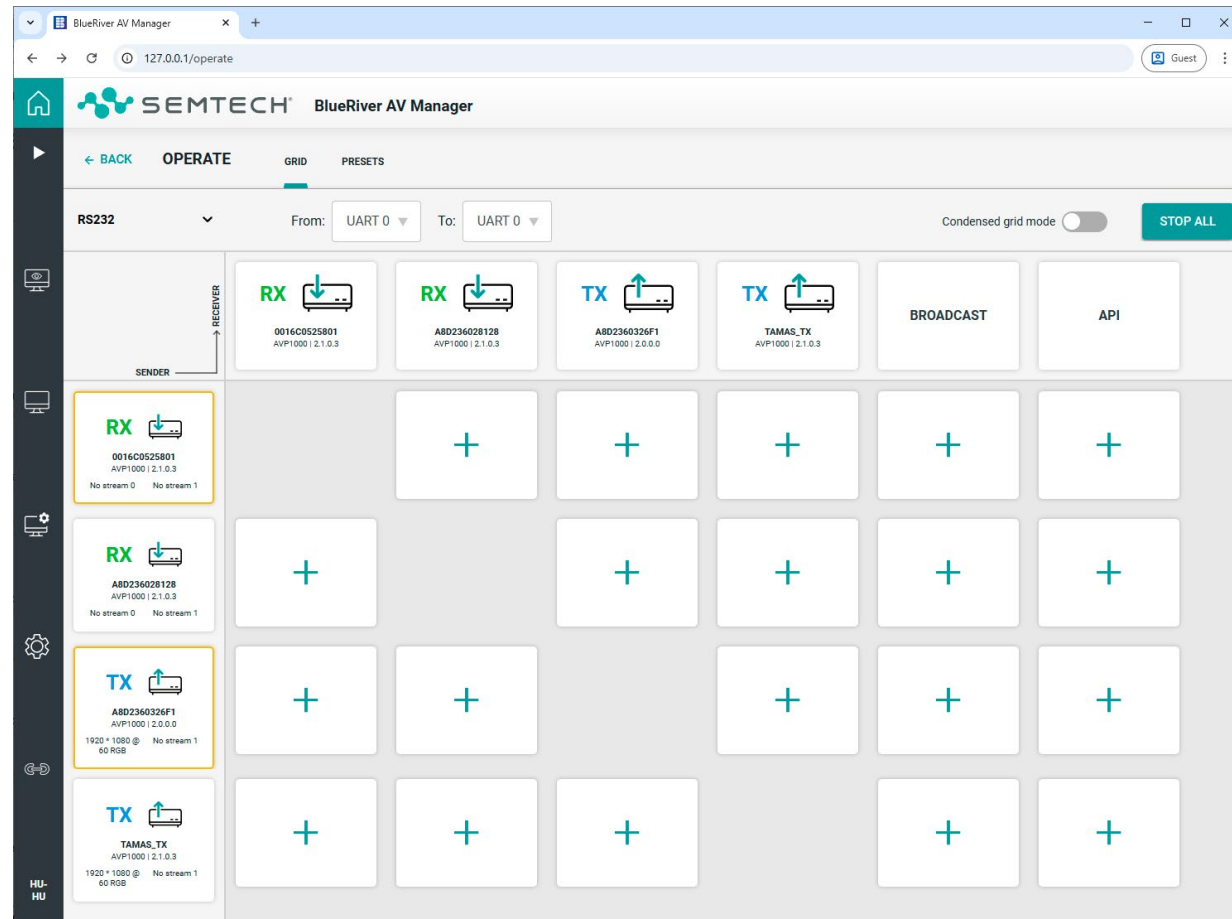
Two types of USB interfaces are supported: HID interface for keyboards, mice and other human interface devices and USB 2.0 (Icron) interface that in addition to mice and keyboards, also supports other types of USB devices such as data storage, audio, and webcams.



USB submenu - HID tab in the Operate menu

4.5.4. RS232

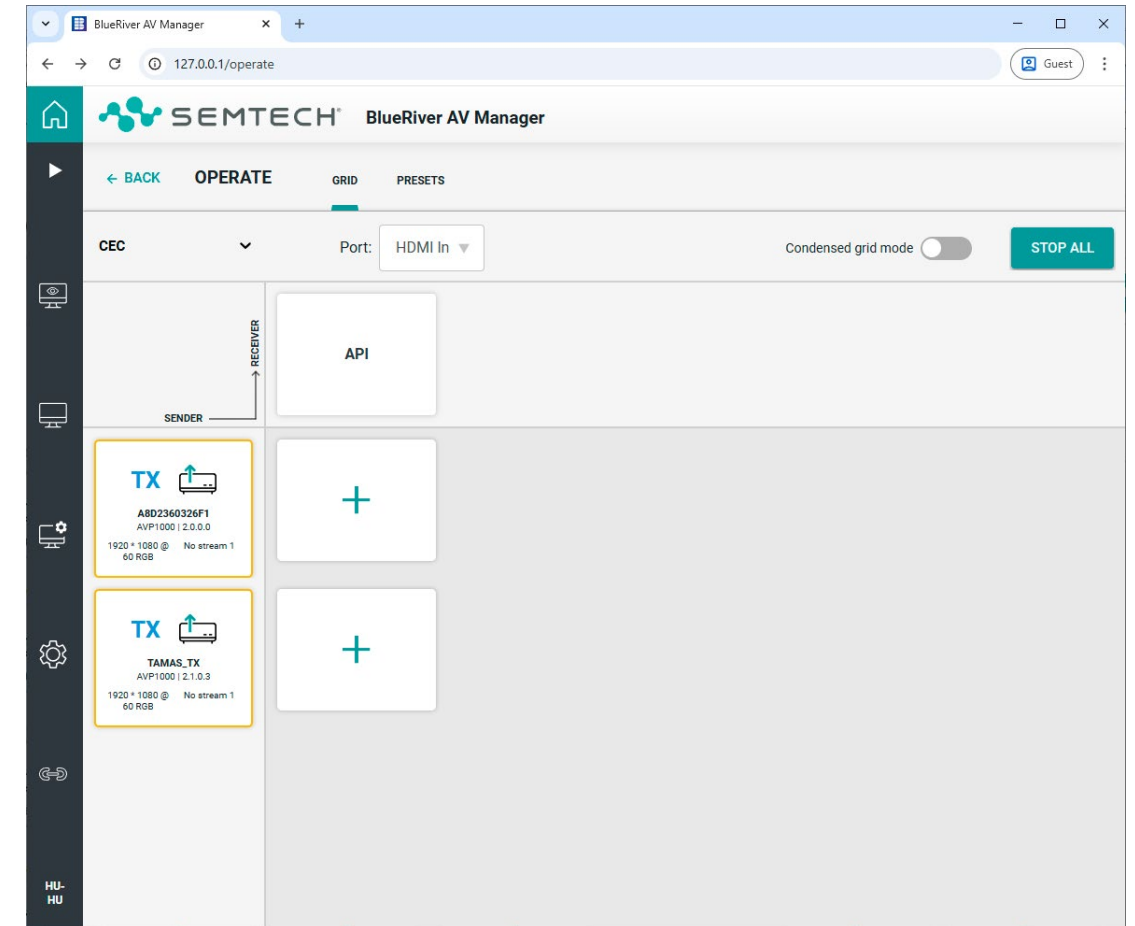
RS-232 signal routing can be applied in the RS232 submenu. All Lightware endpoint devices are built with RS-232 ports.



RS232 submenu in the Operate menu

4.5.5. CEC

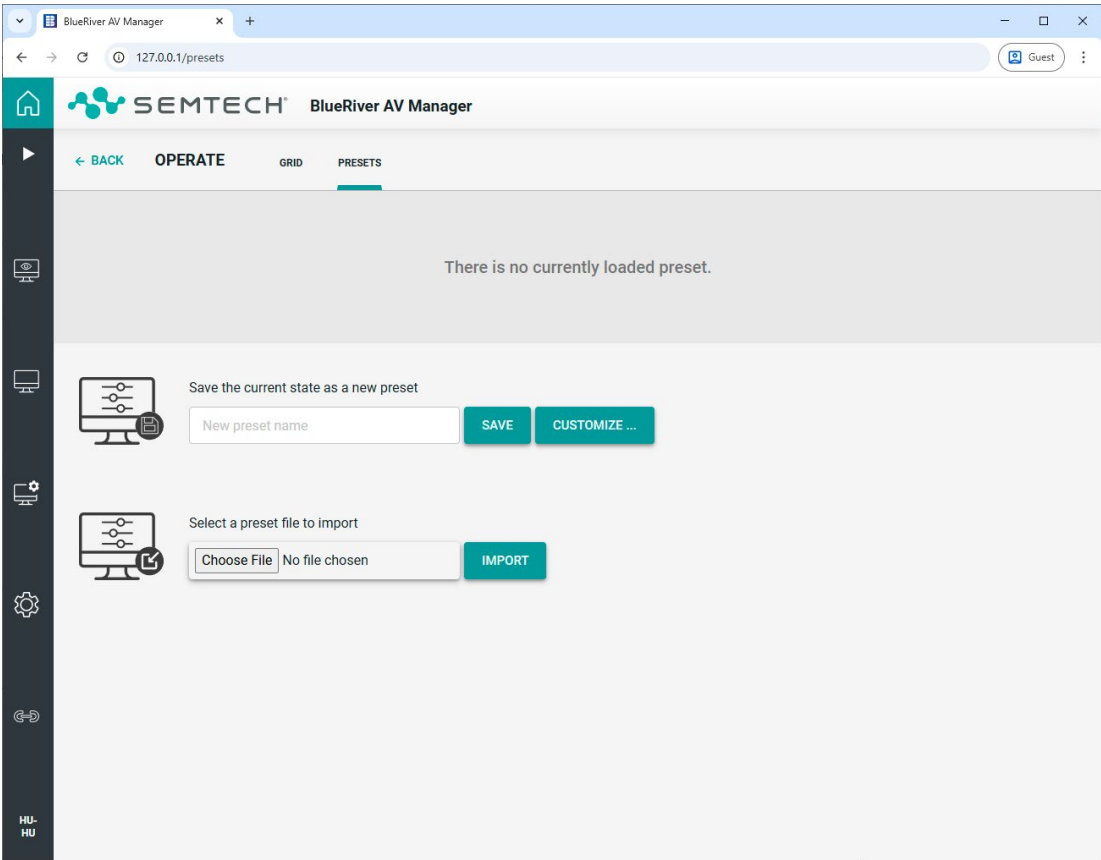
For CEC, the only available receiver tile is named “API” and allows CEC routing from a device to the API (routing CEC between devices without passing through the API is not supported by SDVoE devices).



CEC submenu in the Operate menu

4.5.6. Presets

System state can be saved as a ‘Preset’ that can be recalled later. To access the **Preset** page, select Presets within the Operate tool. Here presets can be saved, applied, renamed, or deleted. It is also possible to customize the presets to save either all of the video, audio, USB-HID and USB 2.0 configurations or only some of them, and for all or only selected devices. These presets can be exported to and imported from a file.

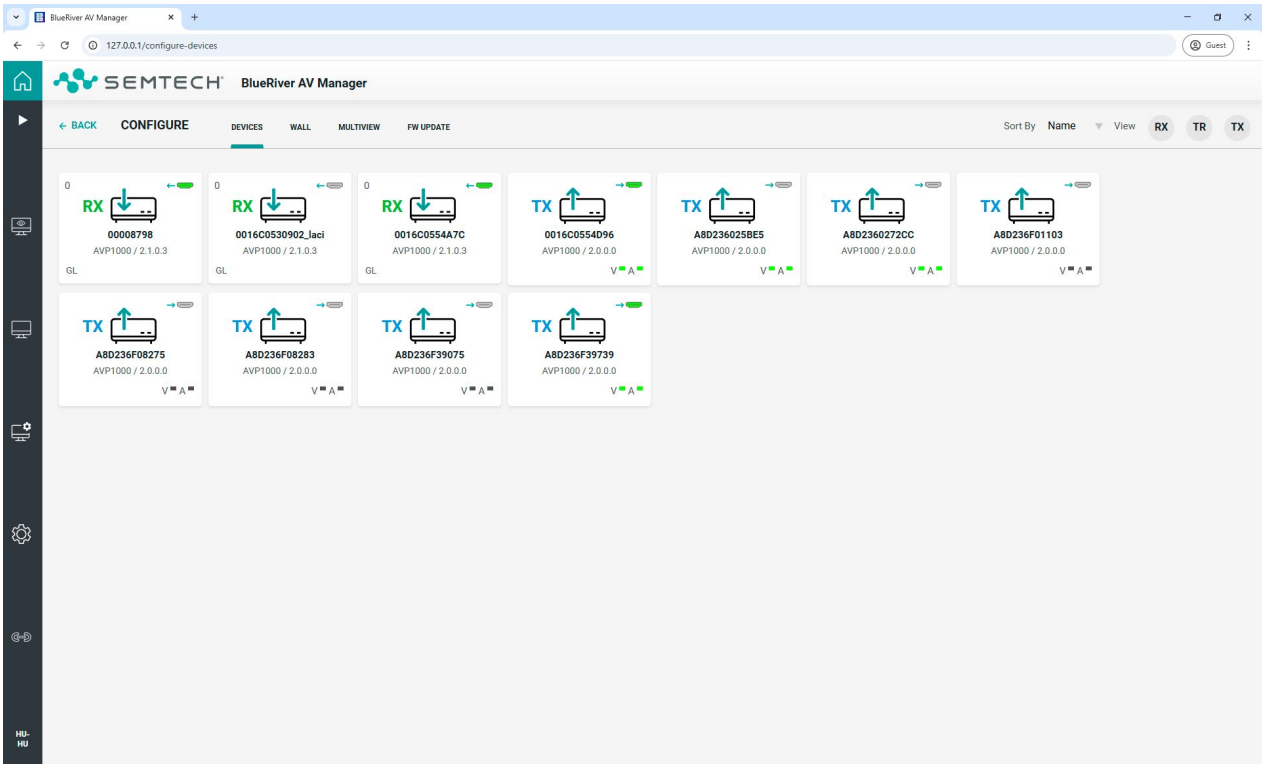


Presets submenu in the Operate menu

To customize, click on the **CUSTOMIZE** button and select the appropriate categories and devices, as shown in screenshot below. The chosen customization can also be saved as default for future presets if desired.

4.6. Configure Menu

4.6.1. Devices



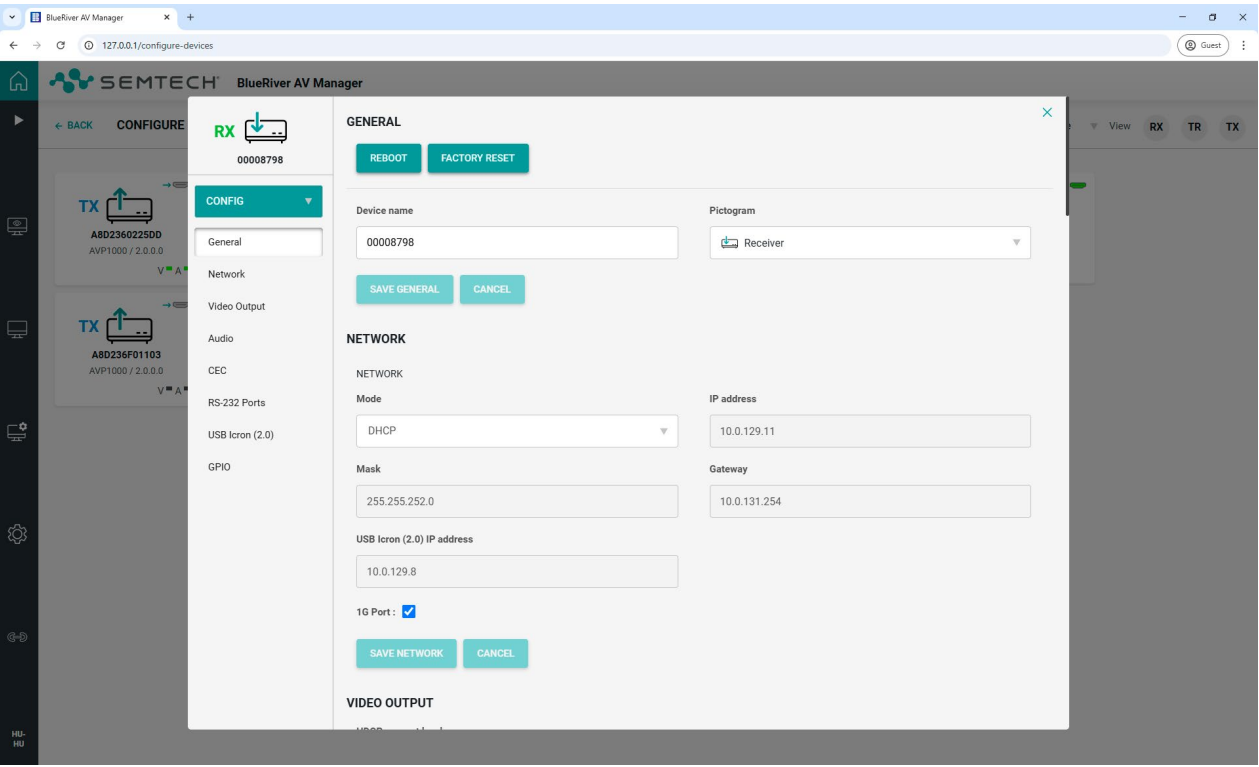
Devices tab in the Configure menu

The available devices on the network are listed in the Configure menu one-by-one on **Device cards**. There are three submenu on each cards:

- **Config** - General and video/audio/control signal-related settings of the extender.
- **Details** - General and status information about the extenders.
- **Diagnostic** - Network-related diagnostic tool.

When a device card is opened from the Configure menu, the default submenu is the **Config**.

Config Submenu



Config submenu on a Device card

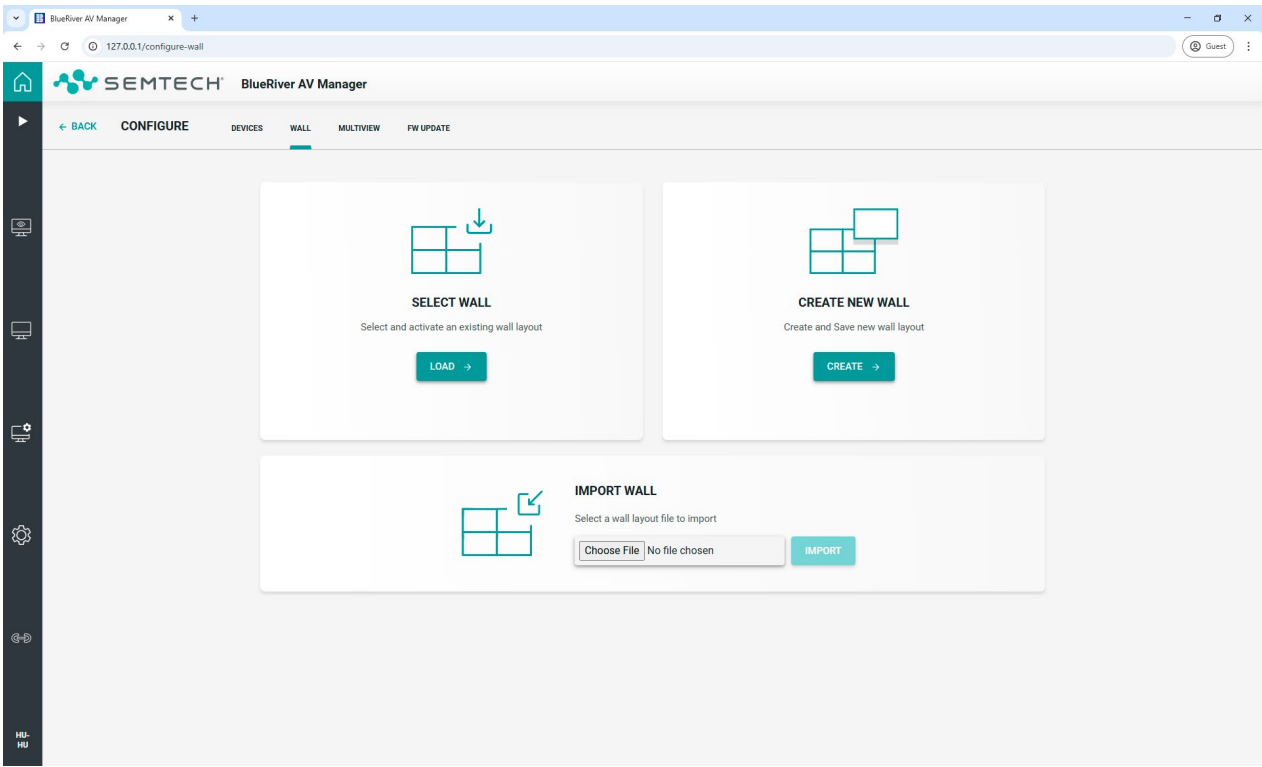
Configuration settings can be applied in the endpoint devices as follow:

- **General actions**; e.g. reboot and recalling factory default settings;
- **Network related settings**; e.g. dynamic IP address (DHCP) setting;
- **Video input / output settings**; e.g. HDCP settings, display mode (scaler settings), video compression level, EDID uploading;
- **Audio settings**; e.g. HDMI output source;
- **CEC** (Consumer Electronics Control) settings;
- **RS-232 ports settings**; e.g. Baud rate, data bits, stop bits, parity;
- **USB Icron (2.0) settings**; e.g. local or remote role;
- **USB HID settings**; e.g. local or remote role;
- **GPIO settings**; e.g. mode and value by ports.

4.6.2. Wall

DIFFERENCE! Lightware highly recommends **using scaling receivers** (HDMI-TPN-SR / HDMI-OPTN-SR series) on the video wall side.

Built into the Configuration tool are dedicated Video Wall editor. This editor is used to create and load Wall layouts or presets. These layouts can be exported to and imported from a file.

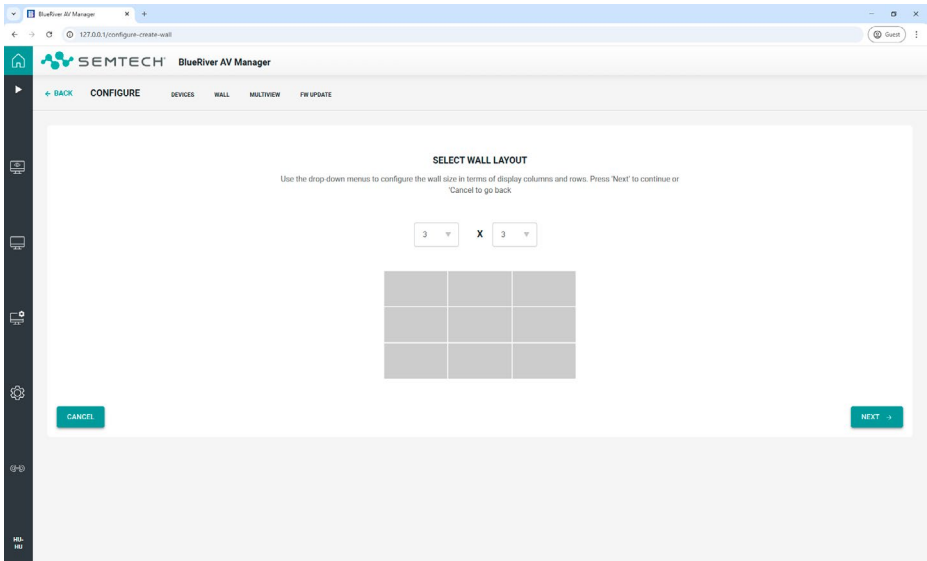


Video Wall tab in the Configure menu

Creating a Wall

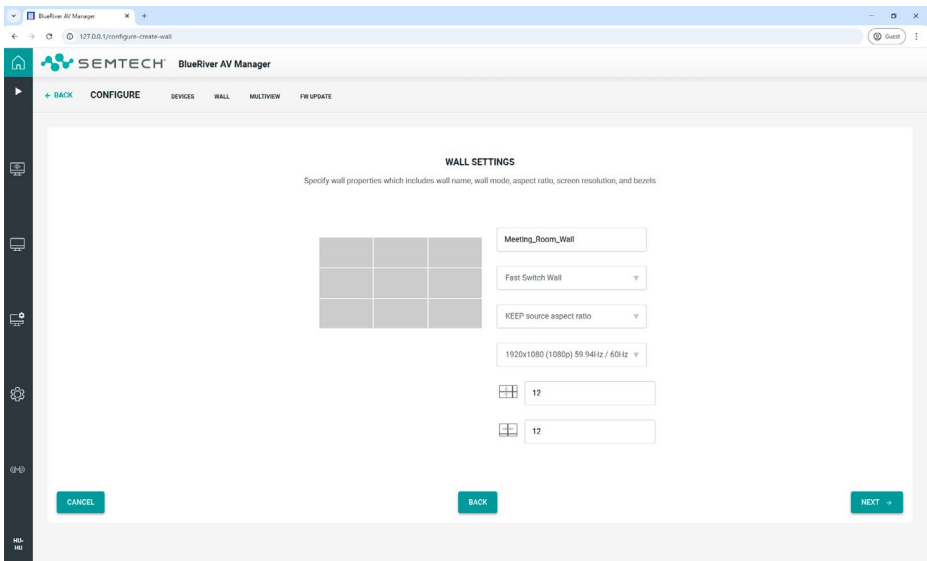
Creating a new wall layout is a three-step process:

Step 1. Set number of monitor rows and columns.



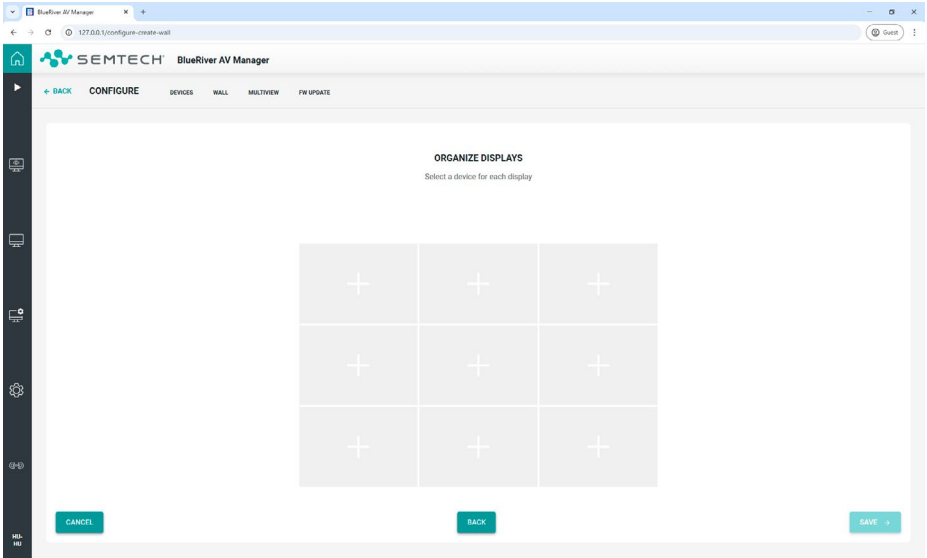
Video Wall tab - Layout selection

Step 2. Set wall parameters including layout name, wall output mode, wall aspect ratio mode, display resolution and display bezels.



Video Wall tab - Wall parameters

Step 3. Assign an endpoint to each wall display.

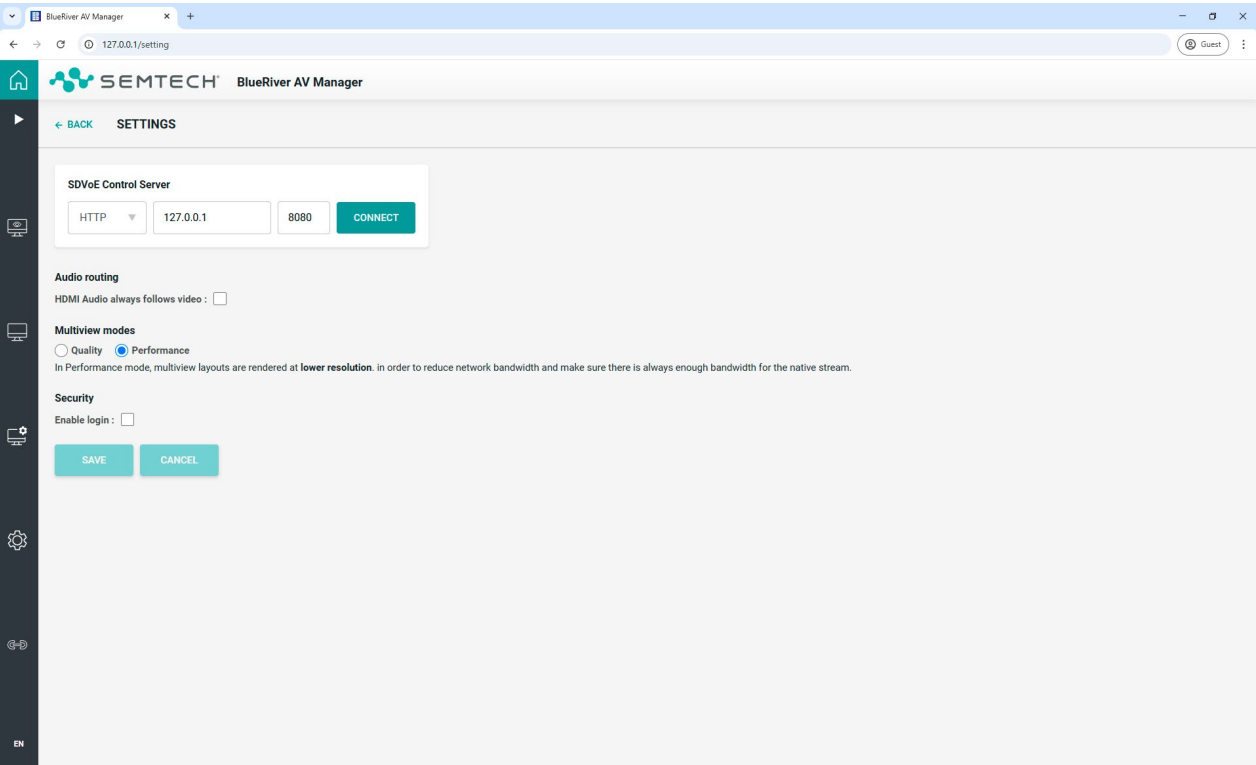


Video Wall tab - Endpoint assignment

To activate Video Wall layout, it need to be loaded. To load a layout, press the **Load** button, select a layout, and then press **Confirm**. It is possible to edit a layout before applying. To do that, press the **Edit** button. Finally, old layouts that are no longer in use can be deleted.

4.7. Settings Menu

Application settings are accessed directly through the navigation panel.



Settings menu

SDVoE Control Server: When BlueRiver AV Manager is started, it connects to the local BlueRiver AV Manager API or to the SDVoE API configured in the BlueRiver AV Manager configuration.conf file if this config is present. Once BlueRiver AV Manager is running, it is possible to change this for BlueRiver AV Manager to connect to a different API via either HTTP or HTTPS, by entering the desired API IP address and port in the global settings SDVoE Control Server text boxes and clicking on the CONNECT button.

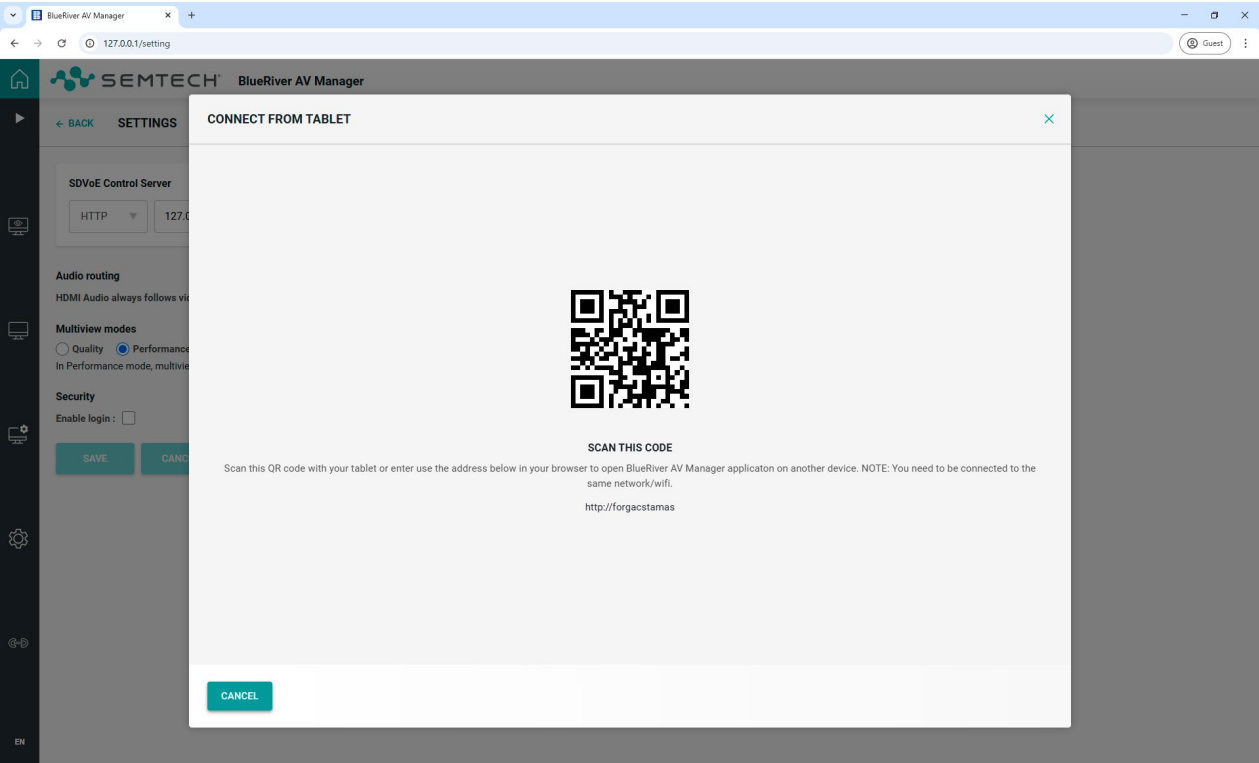
Audio Routing: By default, when using the Operate tool to route video, the associated audio has to be routed separately. Enabling this option allows any audio received from the video source to be automatically routed to the same destinations as the video.

Multiview modes: Multiview relies on additional scaled video streams that can interfere with the native source video stream (not enough bandwidth to carry both). When in Performance mode, Multiview layouts are rendered in lower resolution to reduce the bandwidth usage to accommodate both the native and scaled streams used for Multiview. On the other hand, Quality mode renders Multiview layouts at full resolution while automatically stopping the native video stream.

Security: By default, user login is disabled. Enable this option to activate username and password login.

4.8. Connection via Mobile Device

The BlueRiver AV Manager supports mobile device usage as well. Scan the QR code located in the Connection via Mobile Device menu to open the website of the application.



Connection via Mobile Device menu

5

BlueRiver API Management - Controlling over Commands

This section refers to the HDMI video and HDMI audio interfaces of the endpoint devices and describes several related control commands with examples. Two methods are described for the same task in each section: TCI API and REST API.

- ▶ [INSTRUCTIONS FOR THE TERMINAL APPLICATION USAGE](#)
- ▶ [BLUERIVER API MANAGEMENT STRUCTURE EXPLANATION](#)
- ▶ [REST API HTTP COMMAND BASICS](#)
- ▶ [AV INTERFACE CONTROL OVER API](#)
- ▶ [ICRON USB INTERFACE CONTROL OVER API](#)

5.1. Instructions for the Terminal Application Usage

5.1.1. Terminal Application

The SDVoE (BlueRiver) protocol commands can be applied to the TPN and OPTN series endpoint devices using a terminal application. You need to install one of them to your control device, for example **YAT** (Yet Another Terminal) or **Putty**.

Establishing Connection

Follow the steps for establishing connection to the device:

Step 1. Connect the endpoint to a LAN over Ethernet.

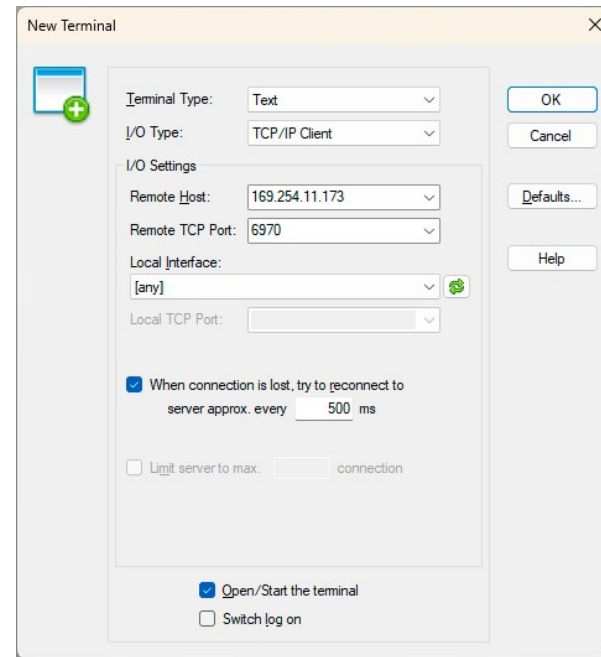
Step 2. Open the terminal application (e.g. YAT).

Step 3. Set the New Terminal window as the follows:

- Terminal type: **Text**
- I/O Type: **TCP/IP Client**
- Remote Host: the **IP address** of the endpoint
- Remote TCP Port: **6970**

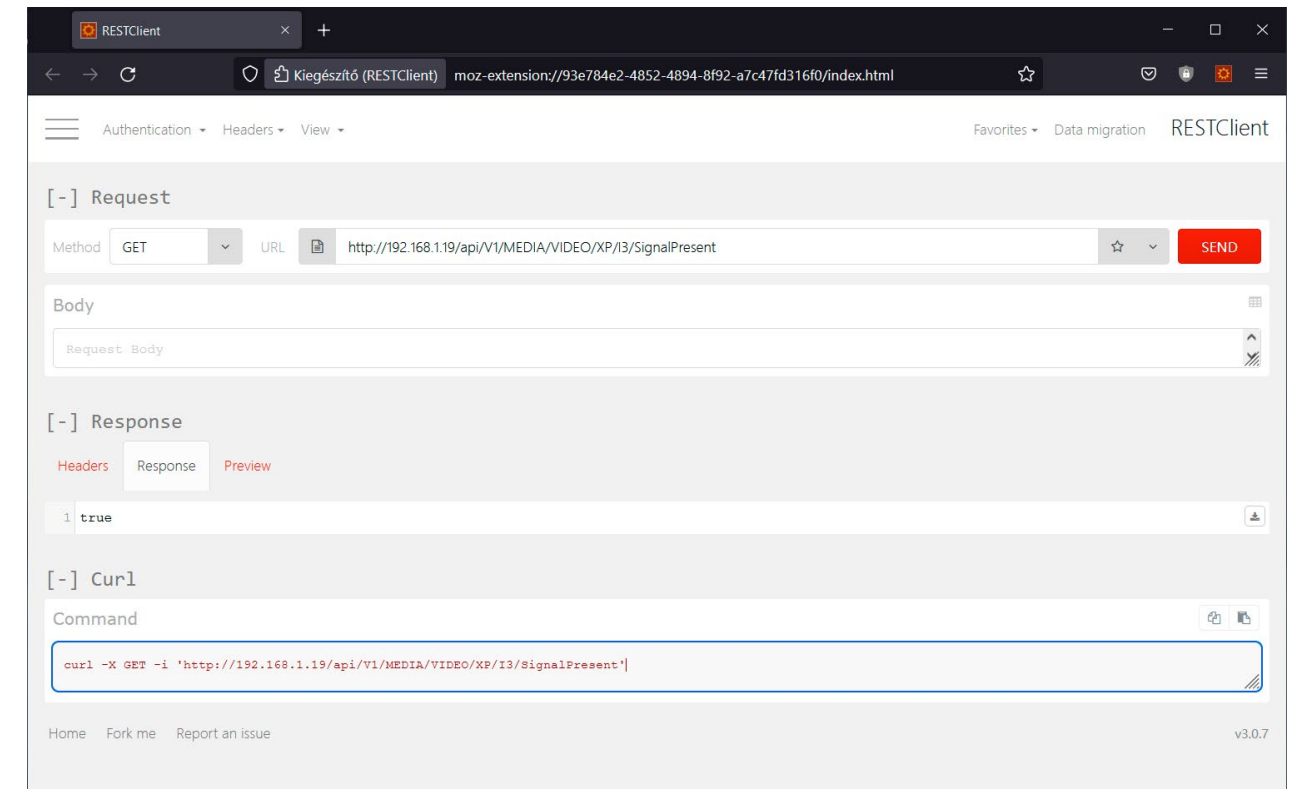
Step 4. Select the **OK** button and open the connection.

Once the terminal window is opened, you can enter the BlueRiver protocol commands, which are listed in the following sections.



5.1.2. Web Browser Plugins

REST API interface can be easily accessed via a web browser's plugin, see the example below:



RESTClient plugin for Mozilla Firefox

5.2. BlueRiver API Management Structure Explanation

This section makes an explanation how the BlueRiver API management structure is working.

5.2.1. Command Types

The BlueRiver Control Server (API) supports two types of commands:

Immediate: This type of command always immediately returns with either a *success* or *error* status.

Background: These commands complete a task in the background. The response of these commands is retrieved using a **request command**. A unique request ID is assigned to each command of this type when it is issued and using the **request** command user can poll the status of the request.

5.2.2. Background Commands

Implementing background commands prevents stalling client software execution. They prevent the API from waiting for a device to complete a command's execution. Instead, commands return with a status of "PROCESSING" along with a unique request ID in the **request_id** member of a response.

The request command is then used with the request ID to retrieve the command's return.

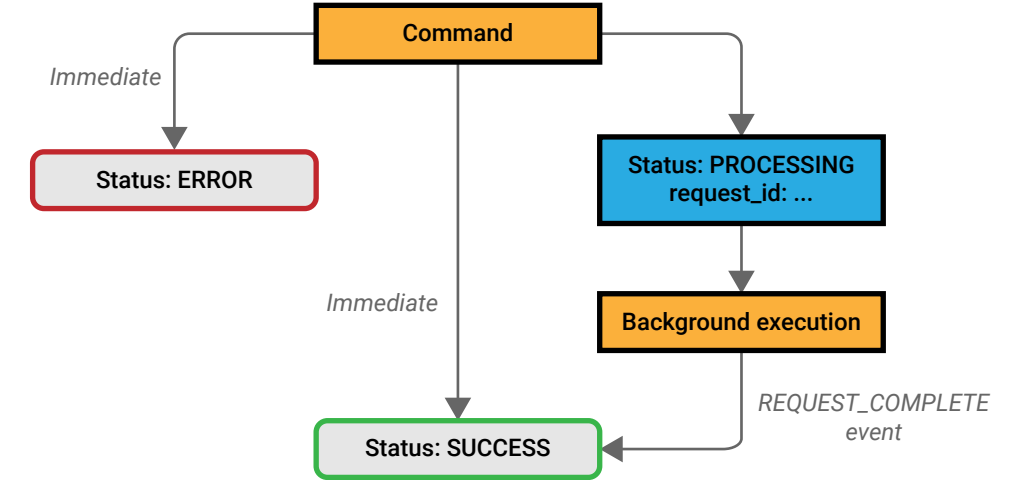
In the REST API, a PROCESSING response has the HTTP status **201 Created**. The Location HTTP header of the response provides a link to the appropriate request resource.

The BlueRiver Control Server (API) buffers command returns and notification events.

The lifecycle of a background command is as follows:

- Valid background commands return with the status member set to PROCESSING, along with a unique request ID specified in the JSON response.
- In case of an error, such as "illegal argument", the command immediately returns an ERROR status.
- After a command has been issued, its status is retrieved using the request ID through the request command:
 - If the command is still being handled, the request command returns a PROCESSING status.
 - Once the command has completed, the request command returns a SUCCESS status, as well as the return value (i.e. returned data) of the command.
- Once a command completes, the event REQUEST_COMPLETE is triggered.
 - Once a REQUEST_COMPLETE event is triggered for a given request ID, the request command is guaranteed to return with a SUCCESS status when the request ID is called.

Summary of Command Lifecycle



Summary of command lifecycle

5.2.3. Monitoring Command Execution

Command Syntax - REST API

➔ GET `http://<ip>:<port>/api/request/<request_ID>`

Example - REST API

The following example illustrates confirming the successful execution of a stop command sent to a specific transmitter device.

```

➔ request-line: POST http://169.254.11.134:8080/api/device/RX1
➔ body:      {
➔           "op": "stop",
➔           "stream_type": "HDMI",
➔           "stream_index": "0",
➔           "subscription_index": 0
➔           }
➔ HTTP/1.1 201 Created
➔ Location: /api/request/85521
  
```

```

← status-line: {
←               "status": "PROCESSING",
←               "request_id": 85521,
←               "result": null,
←               "error": null
←             }
  
```

Response example of the BlueRiver Control Server (API) for the “REQUEST_COMPLETE” event raised when stop command has completed:

```
→ request-line: GET-http://169.254.11.134:8080/api/event?after=85528
→ ...
→ HTTP/1.1 200 OK
→ body:      {
→           "status" : "SUCCESS",
→           "request_id" : null,
→           "result" : {
→               "events" : [
→                   {
→                       "device_id" : null,
→                       "event_id" : 85529,
→                       "event_type" : "REQUEST_COMPLETE",
→                       "timestamp" : 1462902262,
→                       "request_id" : 85521
→                   }
→               ]
→           },
→           "error" : null
→       }

→ request-line: GET-http://169.254.11.134:8080/api/request/85528
→ ...
→ HTTP/1.1 200 OK
→ body:      {
→           "status" : "SUCCESS",
→           "request_id" : 85521,
→           "result" : {
→               "devices" : [
→                   {
→                       "device_id" : "001ec0f03668",
→                       ...
→                   }
→               ],
→               "error" : []
→           },
→           "error" : null},
→       }
```

5.3. Rest API HTTP Command Basics

The following table summarizes the REST API based HTTP commands and their functions.

Command	Description
GET	The GET method requests a representation of the specified resource. Requests using GET should only retrieve data and should not contain a request content.
POST	The POST method submits an entity to the specified resource, often causing a change in state or side effects on the server.
PUT	The PUT method replaces all current representations of the target resource with the request content.
DELETE	The DELETE method deletes the specified resource.

INFO: There are commands that immediately take effect and there are commands that need time to be executed. In case of the latter one, the status PROCESSING is returned. The final status of the execution can be queried based on the request_id.

5.4. AV Interface Control over API

This section refers to the HDMI video and HDMI audio interfaces of the endpoint devices and describes several related control commands with examples. Two methods are described for the same task in each section:

- TCI API - it can be applied in a terminal application, e.g. YAT
- REST API - it can be applied in a web browser with plugins, e.g. Mozilla with RESTClient plugin

5.4.1. Applying API Version

API version is required to be specified in **each TCP session** (not before each command sending) by sending the following command.

Command and Response

```
→ require-blueriver_api<api_version>
← {"status":"<status>","request_id":"<request_ID>","result":"null","error":"null"}
```

Example

```
→ require blueriver_api3.8.0.1
← {"status":"SUCCESS","request_id":"1501","result":"null","error":"null"}
```

5.4.2. Starting an HDMI Audio/Video Stream

The *start* command is used to initiate/start one or more stream(s) on one or more devices.

Command and Response - TCI API

```
→ start<transmitter>:<AV_interface>:0
← {"status": "<status>", "request_id": "<request_ID>", "result": "<result>", "error": "<error>"}
```

Parameters

Parameter	Parameter description	Values	Parameter value
<transmitter>	The name or MAC address of the transmitter		Device name / MAC address
		ALL_TX	All transmitter devices
<AV_interface>	Applied AV interface	HDMI	HDMI video stream
		HDMI_AUDIO	HDMI audio stream

Example - TCI API

```
→ start TX1:HDMI:0
← {"status": "PROCESSING", "request_id": "709", "result": "null", "error": "null"}
```

Command and Response - REST API

```
➔ request-line: POST-http://<ip>:<port>/api/device/<transmitter>
➔ body: {
➔     "op": "start",
➔     "stream_type": "<AV_interface>",
➔     "stream_index": "0",
➔ }
← status-line: {
←     "status": "<status>",
←     "request_id": <request_ID>,
←     "result": <result>,
←     "error": <error>
← }
```

Parameters

Parameter	Parameter description	Values	Parameter value
<ip>	The IP address of the transmitter		IP address
<port>	Port number	8080	Port number
<transmitter>	The name or MAC address of the transmitter		Device name / MAC address
		ALL_TX	All transmitter devices
<AV_interface>	Applied AV interface	HDMI	HDMI video stream
		HDMI_AUDIO	HDMI audio stream

Example - REST API

```
➔ request-line: POST-http://169.254.11.134:8080/api/device/TX1
➔ body: {
➔     "op": "start",
➔     "stream_type": "HDMI",
➔     "stream_index": 0,
➔ }
← status-line: {
←     "status": "PROCESSING",
←     "request_id": 709,
←     "result": null,
←     "error": null
← }
```

5.4.3. Stopping a Stream

The *stop* command is used to exterminate one or more stream(s) on one or more devices.

Command and Response - TCI API

```
→ stop<device>:<interface>:0 [free]
← {"status": "<status>", "request_id": "<request_ID>", "result": "<result>", "error": "<error>"}
```

Stops one or more stream(s) on one or more device(s) and optionally free the multicast address associated with each stopped stream so it can be re-used for another stream.

Parameters

Parameter	Parameter description	Values	Parameter value
<device>	The name or MAC address of the transmitter or receiver		Device name / MAC address
		ALL	All devices
		ALL_TX	All transmitter devices
		ALL_RX	All receiver devices
<interface>	Applied AV interface	HDMI	HDMI video stream (for TX only)
		HDMI_AUDIO	HDMI audio stream (for TX only)
		CEC	CEC stream
		INFRARED	Infrared signal
		RS232	RS-232 signal
		USB_HID	USB HID signal
free	If the free keyword is present (text command) or free argument is true (REST API) then the multicast IP address associated with each target stream is freed. For streams with a type of either RS232 or INFRARED where no multicast IP address is associated with the stream, the free keyword is allowed but ignored.		

Example - TCI API

→ stop TX1:HDMI:0 free

← {"status":"PROCESSING","request_id":"968","result":"null","error":"null"}

Command and Response - REST API

```
→ request-line: POST:http://<ip>:<port>/api/device/<transmitter>
→ body: {
→       "op" : "stop",
→       "stream_type" : "<interface>",
→       "stream_index" : "0",
→       "free" : "<logical_value>",
→     }
← status-line: {
←       "status" : "<status>",
←       "request_id" : <request_ID>,
←       "result" : <result>,
←       "error" : <error>
←     }
```

Parameters

Parameter	Parameter description	Values	Parameter value
<ip>	The IP address of the device		IP address
<port>	Port number	8080	Port number
<device>	The name or MAC address of the transmitter or receiver		Device name / MAC address
		ALL	All devices
		ALL_TX	All transmitter devices
		ALL_RX	All receiver devices
<interface>	Applied AV interface	HDMI	HDMI video stream (for TX only)
		HDMI_AUDIO	HDMI audio stream (for TX only)
		CEC	CEC stream
		INFRARED	Infrared signal
		RS232	RS-232 signal
		USB_HID	USB HID signal
free	If the free keyword is present (text command) or free argument is true (REST API) then the multicast IP address associated with each target stream is freed.	true	The multicast IP address associated with each target stream is freed.
		false	The multicast IP address associated with each target stream is not freed.

Example - REST API

```
→ request-line: POST:http://169.254.11.134:8080/api/device/TX1
→ body: {
→       "op" : "stop",
→       "stream_type" : "HDMI",
→       "stream_index" : 0,
→       "free" : true,
→     }
← status-line: {
←       "status" : "PROCESSING",
←       "request_id" : 968,
←       "result" : null,
←       "error" : null
←     }
```

5.4.4. Switching an HDMI Audio/Video Stream

ATTENTION! Applying the *start* command is required before switching a stream. See more details in the [Starting an HDMI Audio/Video Stream](#) section.

Command and Response - TCI API

→ join-<transmitter>:<AV_interface>:0-<receiver>:<AV_interface>:0

← {"status":"<status>","request_id":"<request_ID>","result":"<result>","error":"<error>"}

Parameters

Parameter	Parameter description	Values	Parameter value
<transmitter>	The name or MAC address of the transmitter		Device name / MAC address
<receiver>	The name or MAC address of the receiver		Device name / MAC address
		ALL_RX	All receiver devices
<AV_interface>	Applied AV interface	HDMI	HDMI video stream
		HDMI_AUDIO	HDMI audio stream

Example - TCI API

→ join TX1:HDMI:0 RX1:HDMI:0

← {"status":"PROCESSING","request_id":"1501","result":"null","error":"null"}

Command and Response - REST API

```
➔ request-line: POST-http://<ip>:<port>/api/device/<receiver>
➔ body:      {
➔           "op" : "join",
➔           "source_device" : "<transmitter>",
➔           "stream_type" : "<AV_interface>",
➔           "stream_index" : "0",
➔           "subscription_index" : 0
➔       }
⬅ status-line: {
⬅           "status" : "<status>",
⬅           "request_id" : <request_ID>,
⬅           "result" : <result>,
⬅           "error" : <error>
⬅       }
```

Parameters

Parameter	Parameter description	Values	Parameter value
<ip>	The IP address of the transmitter		IP address
<port>	Port number	8080	Port number
<transmitter>	The name or MAC address of the transmitter		Device name / MAC address
<receiver>	The name or MAC address of the receiver		Device name / MAC address
<AV_interface>	Applied AV interface	ALL_RX	All receiver devices
		HDMI	HDMI video stream
		HDMI_AUDIO	HDMI audio stream

Example - REST API

```
➔ request-line: POST-http://169.254.11.134:8080/api/device/RX1
➔ body:      {
➔           "op" : "join",
➔           "source_device" : "TX1",
➔           "stream_type" : "HDMI",
➔           "stream_index" : 0,
➔           "subscription_index" : 0
➔       }
⬅ status-line: {
⬅           "status" : "PROCESSING",
⬅           "request_id" : 1501,
⬅           "result" : null,
⬅           "error" : null
⬅       }
```

5.4.5. Disconnecting an HDMI Stream

ATTENTION! Applying the *start* command is required before disconnecting a stream. See more details in the [Starting an HDMI Audio/Video Stream](#) section.

Command and Response - TCI API

```
➔ leave-<receiver>:<AV_interface>:0
⬅ {"status":"<status>","request_id":"<request_ID>","result":"<result>","error":"<error>"}
```

Parameters

Parameter	Parameter description	Values	Parameter value
<receiver>	The name or MAC address of the receiver		Device name / MAC address
<AV_interface>	Applied AV interface	ALL_RX	All receiver devices
		HDMI	HDMI video stream
		HDMI_AUDIO	HDMI audio stream

Example - TCI API

```
➔ leave RX1:HDMI:0
⬅ {"status":"PROCESSING","request_id":"1129","result":"null","error":"null"}
```

Command and Response - REST API

```
➔ request-line: POST-http://<ip>:<port>/api/device/<receiver>
➔ body:      {
➔           "op" : "leave",
➔           "subscription_type" : "<AV_interface>",
➔           "subscription_index" : 0
➔       }
⬅ status-line: {
⬅           "status" : "<status>",
⬅           "request_id" : <request_ID>,
⬅           "result" : <result>,
⬅           "error" : <error>
⬅       }
```

Parameters

Parameter	Parameter description	Values	Parameter value
<ip>	The IP address of the transmitter		IP address
<port>	Port number	8080	Port number
<receiver>	The name or MAC address of the receiver		Device name / MAC address
		ALL_RX	All receiver devices
<AV_interface>	Applied AV interface	HDMI	HDMI video stream
		HDMI_AUDIO	HDMI audio stream

Example - REST API

```
➔ request-line: POST-http://169.254.11.134:8080/api/device/RX1
➔ body:      {
➔           "op" : "leave",
➔           "subscription_type" : "HDMI",
➔           "subscription_index" : 0
➔       }
⬅ status-line: {
⬅           "status" : "PROCESSING",
⬅           "request_id" : 1129,
⬅           "result" : null,
⬅           "error" : null
⬅       }
```

5.4.6. Setting the EDID

The command option is used to set the EDID of one or more transmitter device(s). Command applies **only to transmitter** devices.

Command and Response - TCI API

```
➔ set-<transmitter>-edid-<data_hex>
⬅ {"status":<status>,"request_id":<request_ID>,"result":<result>,"error":<error>}
```

Parameters

Parameter	Parameter description	Values	Parameter value
<transmitter>	The name or MAC address of the transmitter		Device name / MAC address
		ALL_TX	All transmitter devices
<data_hex>	EDID data argument in hexadecimal string		The data argument must be a hexadecimal string representing the binary data to set. Its length must be 512 hexadecimal characters (for example, data length must be 256 bytes).

Example - TCI API

```
➔ set TX1 edid 00ffffffff00410c2fc0c52d00000c140103802f1a782e3585a656489a241250
542f6f00714f8180818a9500950fb3000101d1c0023a801871382d40582c4500dc0c
1100001e000000ff0041553531303132303131373137000000fd00384c1e5315000a
2020202020000000fc005068696c69707320323231450a01b602031ef04b100501
020304061213141f230907078301000065030c001100023a801871382d40582c4500
dc0c1100001e8c0ad08a20e02d10103e9600dc0c11000018011d007251d01e206e28
5500dc0c1100001e8c0ad090204031200c405500dc0c11000018000000000000000 0000000000000000
000000000000000000ac
⬅ {"status":"PROCESSING","request_id":"5698","result":"null","error":"null"}
```

Command and Response - REST API

```
➔ request-line: POST-http://<ip>:<port>/api/device/<transmitter>
➔ body:      {
➔           "op" : "set:edid",
➔           "data_hex" : <data_hex>
➔       }
⬅ status-line: {
⬅           "status" : "<status>",
⬅           "request_id" : <request_ID>,
⬅           "result" : <result>,
⬅           "error" : <error>
⬅       }
```

Parameters

Parameter	Parameter description	Values	Parameter value
<ip>	The IP address of the transmitter		IP address
<port>	Port number	8080	Port number
<device>	The name or MAC address of the transmitter or receiver		Device name / MAC address
		ALL_TX	All transmitter devices
<data_hex>	EDID data argument in hexadecimal string		The data argument must be a hexadecimal string representing the binary data to set. Its length must be 512 hexadecimal characters (for example, data length must be 256 bytes).

Example - REST API

```
➔ request-line: POST-http://169.254.11.134:8080/api/device/TX1
➔ body:      {
➔           "op" : "set:edid",
➔           "data_hex" :
00fffffffff00410c2fc0c52d00000c140103802f1a782e3585a656489a241250
542f6f00714f8180818a9500950fb3000101d1c0023a801871382d40582c4500dc0c
1100001e000000ff0041553531303132303131373137000000fd00384c1e5315000a
202020202020000000fc005068696c69707320323231450a01b602031ef04b100501
020304061213141f230907078301000065030c001100023a801871382d40582c4500
dc0c1100001e8c0ad08a20e02d10103e9600dc0c11000018011d007251d01e206e28
5500dc0c1100001e8c0ad090204031200c405500dc0c110000180000000000000000
000000000000000000ac
➔           }
⬅ status-line: {
⬅           "status" : "PROCESSING",
⬅           "request_id" : 368,
⬅           "result" : null,
⬅           "error" : null
⬅       }
```

5.4.7. Scaler Mode Setting

DIFFERENCE! Only HDMI-TPN-RX107A-SR, HDMI-TPN-RX107AU2K-SR, HDMI-OPTN-RX100A-SR and HDMI-OPTN-RX100AU2K-SR receiver models have integrated scaler function.

INFO: Learn more about the scaler modes of the -SR series endpoint devices in the [TPN/OPTN user manual](#).

The command sets the scaler mode of the -SR series receiver.

Command and Response - TCI API

```
➔ set-<receiver>-video-<scaling_mode>
⬅ {"status":"<status>","request_id":"<request_ID>","result":"<result>","error":"<error>"}
```

Parameters

Parameter	Parameter description	Values	Parameter value
<receiver>	The name or MAC address of the receiver		Device name / MAC address
		ALL_RX	All receiver devices
<scaling_mode>	Scaling mode of the receiver	genlock	Genlock
		genlock_scaling	Genlock scaling
		fastswitch	Fast switch
		wall	Wall (Genlock Wall)
		wall_fs	Wall FS (Fastswitch Wall)

Example - TCI API

```
➔ set RX2 video genlock_scaling
⬅ {"status":"PROCESSING","request_id":"1661","result":"null","error":"null"}
```

Command and Response - REST API

```
➔ request-line: POST-http://<ip>:<port>/api/device/<receiver>
➔ body:      {
➔           "op" : "set:video",
➔           "display_mode" : "<scaling_mode>",
➔           }
⬅ status-line: {
⬅           "status" : "<status>",
⬅           "request_id" : <request_ID>,
⬅           "result" : <result>,
⬅           "error" : <error>
⬅       }
```

Parameters

Parameter	Parameter description	Values	Parameter value
<ip>	The IP address of the transmitter		IP address
<port>	Port number	8080	Port number
<receiver>	The name or MAC address of the receiver		Device name / MAC address
		ALL_RX	All receiver devices
<scaling_mode>	Scaling mode of the receiver	genlock	Genlock
		genlock_scaling	Genlock scaling
		fastswitch	Fast switch
		wall	Wall (Genlock Wall)
		wall_fs	Wall FS(Fastswitch Wall)

Example - REST API

```
➔ request-line: POST http://169.254.11.134:8080/api/device/RX2
➔ body: {
➔   "op" : "set:video",
➔   "display_mode" : "genlock_scaling",
➔ }
⬅ status-line: {
⬅   "status" : "PROCESSING",
⬅   "request_id" : 1661,
⬅   "result" : null,
⬅   "error" : null
⬅ }
```

5.4.8. Scaler Parameter Settings

DIFFERENCE! Only HDMI-TPN-RX107A-SR, HDMI-TPN-RX107AU2K-SR, HDMI-OPTN-RX100A-SR and HDMI-OPTN-RX100AU2K-SR receiver models have integrated scaler function.

INFO: Learn more about the scaler modes of the -SR series endpoint devices in the [TPN/OPTN user manual](#).

Command and Response - TCI API

```
➔ set <receiver> video <scaling_mode> size <horizontal_pixels> <vertical_pixels> fps <fps>
⬅ {"status": "<status>", "request_id": "<request_ID>", "result": "<result>", "error": "<error>"}
```

Parameters

Parameter	Parameter description	Values	Parameter value
<receiver>	The name or MAC address of the receiver		Device name / MAC address
		ALL_RX	All receiver devices
<scaling_mode>	Scaling mode of the receiver	genlock	Genlock
		genlock_scaling	Genlock scaling
		fastswitch	Fast switch
		wall	Wall (Genlock Wall)
		wall_fs	Wall FS(Fastswitch Wall)
<horizontal_pixels>	Horizontal pixels of the display		
<vertical_pixels>	Vertical pixels of the display		
<fps>	Refresh rate (frame per second)		

Example - TCI API

```
➔ set RX2 video fastswitch size 1920 1080 fps 60
⬅ {"status": "PROCESSING", "request_id": "68", "result": "null", "error": "null"}
```

Command and Response - REST API

```

➔ request-line: POST-http://<ip>:<port>/api/device/<receiver>
➔ body:      {
➔           "op" : "set:video",
➔           "display_mode" : "<scaling_mode>",
➔           "video" : {
➔               "width" : <horizontal_pixels>,
➔               "height" : <vertical_pixels>,
➔               "fps" : <fps>
➔           }
➔       }
⬅ status-line: {
⬅     "status" : "<status>",
⬅     "request_id" : <request_ID>,
⬅     "result" : <result>,
⬅     "error" : <error>
⬅ }

```

Parameters

Parameter	Parameter description	Values	Parameter value
<ip>	The IP address of the transmitter		IP address
<port>	Port number	8080	Port number
<receiver>	The name or MAC address of the receiver		Device name / MAC address
<scaling_mode>	Scaling mode of the receiver	ALL_RX	All receiver devices
		genlock	Genlock
		genlock_scaling	Genlock scaling
		fastswitch	Fast switch
		wall	Wall (Genlock Wall)
		wall_fs	Wall FS(Fastswitch Wall)
<horizontal_pixels>	Horizontal pixels of the display		
<vertical_pixels>	Vertical pixels of the display		
<fps>	Refresh rate (frame per second)		

Example - REST API

```

➔ request-line: POST-http://169.254.11.134:8080/api/device/RX2
➔ body:      {
➔           "op" : "set:video",
➔           "display_mode" : "fastswitch",
➔           "video" : {
➔               "width" : 1920,
➔               "height" : 1080,
➔               "fps" : 60
➔           }
➔       }
⬅ status-line: {
⬅     "status" : "PROCESSING",
⬅     "request_id" : 68,
⬅     "result" : null,
⬅     "error" : null
⬅ }

```

5.4.9. Restoring the Factory Default Settings

The command is used to restore the factory default settings of a single or group of devices.

Command and Response - TCI API

```

➔ factory:<device>
⬅ {"status": "<status>","request_id": "<request_ID>","result": "<result>","error": "<error>"}

```

Parameters

Parameter	Parameter description	Values	Parameter value
<device>	The name or MAC address of the transmitter or receiver		Device name / MAC address
		ALL	All devices
		ALL_TX	All transmitter devices
		ALL_RX	All receiver devices

Example - TCI API

```

➔ factory TX1
⬅ {"status": "PROCESSING","request_id": "155","result": "null","error": "null"}

```

Command and Response - REST API

```
➔ request-line: POST-http://<ip>:<port>/api/device/<device>
➔ body:      {
➔           "op" : "factory",
➔           }
⬅ status-line: {
⬅           "status" : "<status>",
⬅           "request_id" : <request_ID>,
⬅           "result" : <result>,
⬅           "error" : <error>
⬅           }
```

Parameters

Parameter	Parameter description	Values	Parameter value
<ip>	The IP address of the transmitter		IP address
<port>	Port number	8080	Port number
<device>	The name or MAC address of the transmitter or receiver		Device name / MAC address
		ALL	All devices
		ALL_TX	All transmitter devices
		ALL_RX	All receiver devices

Example - REST API

```
➔ request-line: POST-http://169.254.11.134:8080/api/device/TX1
➔ body:      {
➔           "op" : "factory",
➔           }
⬅ status-line: {
⬅           "status" : "PROCESSING",
⬅           "request_id" : 68,
⬅           "result" : null,
⬅           "error" : null
⬅           }
```

5.4.10. Rebooting the Endpoint Device

The command causes the target device(s) to restart. The target device(s) become unavailable for a short period of time while the restart is completed.

Command and Response - TCI API

```
➔ reboot-<device>
⬅ {"status":"<status>","request_id":"<request_ID>","result":"<result>","error":"<error>"}
```

Parameters

Parameter	Parameter description	Values	Parameter value
<device>	The name or MAC address of the transmitter or receiver		Device name / MAC address
		ALL	All devices
		ALL_TX	All transmitter devices
		ALL_RX	All receiver devices

Example - TCI API

```
➔ reboot ALL_TX
⬅ {"status":"PROCESSING","request_id":"368","result":"null","error":"null"}
```

Command and Response - REST API

```
➔ request-line: POST-http://<ip>:<port>/api/device/<device>
➔ body:      {
➔           "op" : "reboot"
➔           }
⬅ status-line: {
⬅           "status" : "<status>",
⬅           "request_id" : <request_ID>,
⬅           "result" : <result>,
⬅           "error" : <error>
⬅           }
```

Parameters

Parameter	Parameter description	Values	Parameter value
<ip>	The IP address of the transmitter		IP address
<port>	Port number	8080	Port number
<device>	The name or MAC address of the transmitter or receiver		Device name / MAC address
		ALL	All devices
		ALL_TX	All transmitter devices
		ALL_RX	All receiver devices

Example - REST API

```
➔ request-line: POST-http://169.254.11.134:8080/api/device/ALL_TX
➔ body:      {
➔           "op" : "reboot"
➔           }
⬅ status-line: {
⬅           "status" : "PROCESSING",
⬅           "request_id" : 368,
⬅           "result" : null,
⬅           "error" : null
⬅           }
```

5.4.11. Renaming the Device

The command sets the device name to a custom text.

Command and Response - TCI API

```
➔ set-<device>·property·configuration.device_name·<name>
⬅ {"status":"<status>","request_id":"<request_ID>","result":"<result>","error":"<error>"}
```

Parameters

Parameter	Parameter description	Values	Parameter value
<device>	The name or MAC address of the transmitter or receiver		Device name / MAC address
<name>	Custom device name		

Example - TCI API

```
➔ set 001ec0f04d9c property configuration.device_name TX1
⬅ {"status":"PROCESSING","request_id":"691","result":"null","error":"null"}
```

Command and Response - REST API

```
➔ request-line: POST-http://<ip>:<port>/api/device/<device>
➔ body:      {
➔           "op" : "set:property",
➔           "key" : "configuration.device_name",
➔           "value" : "<name>"
➔           }
⬅ status-line: {
⬅           "status" : "<status>",
⬅           "request_id" : <request_ID>,
⬅           "result" : <result>,
⬅           "error" : <error>
⬅           }
```

Parameters

Parameter	Parameter description	Values	Parameter value
<ip>	The IP address of the transmitter		IP address
<port>	Port number	8080	Port number
<device>	The name or MAC address of the transmitter or receiver		Device name / MAC address
<name>	Custom device name		

Example - REST API

```
➔ request-line: POST-http://169.254.11.134:8080/api/device/ 001ec0f04d9c
➔ body:      {
➔           "op" : "set:property",
➔           "key" : "configuration.device_name",
➔           "value" : "TX1"
➔           }
⬅ status-line: {
⬅           "status" : "PROCESSING",
⬅           "request_id" : 691,
⬅           "result" : null,
⬅           "error" : null
⬅           }
```

5.5. Icron USB Interface Control over API

DIFFERENCE! The Icron USB 2.0 module of the LARA can be applied to the **U2K series** TPN and OPTN endpoint models and **USB20 series** extenders **only**. *#kvm #usbkvm #icron*

ATTENTION! The Icron interface within the TPN/OPTN devices can only be controlled through the SDVoE protocol if Icron control is enabled in the SDVoE settings.

This section refers to the Icron USB 2.0 interfaces of the endpoint devices and describes several related control commands with examples. Two methods are described for the same task in each section:

- **TCI API** - it can be applied in a terminal application, e.g. YAT
- **REST API** - it can be applied in a web browser with plugins, e.g. Mozilla with RESTClient plugin

5.5.1. Applying API Version

API version is required to be specified in **each TCP session** (not before each command sending) by sending the following command.

Command and Response

```
→ require blueriver_api<api_version>
← {"status":"<status>","request_id":"<request_ID>","result":"null","error":"null"}
```

Example

```
→ require blueriver_api3.8.0.1
← {"status":"SUCCESS","request_id":"1501","result":"null","error":"null"}
```

5.5.2. Pairing the Devices - SUI Mode

DIFFERENCE! The following commands belong to the setting of **SUI mode** for LEX and REX devices only. If the devices are in MSA mode, see the [Pairing the Devices - MSA Mode](#) section for pairing.

DEFINITION: SUI mode: Simultaneous Users Interaction. This mode allows a single USB tier to be presented to the connected device.

The command makes pairing a LEX (transmitter) to a REX (receiver). The command is required calling **twice**, transmitter to receiver and receiver to transmitter as well.

Command and Response - TCI API

```
→ icron<transmitter>·pair·<receiver>\r
← {"status":"<status>","request_id":"<request_ID>","result":"<result>","error":"<error>"}

→ icron<receiver>·pair·<transmitter>\r
← {"status":"<status>","request_id":"<request_ID>","result":"<result>","error":"<error>"}

Parameters
```

Parameter	Parameter description	Values	Parameter value
<transmitter>	The name or MAC address of the transmitter		Device name / MAC address
<receiver>	The name or MAC address of the receiver		Device name / MAC address

Example - TCI API

```
→ icron TX1 pair RX1\r
← {"status":"PROCESSING","request_id":"2336","result":"null","error":"null"}

→ icron RX1 pair TX1\r
← {"status":"PROCESSING","request_id":"2337","result":"null","error":"null"}
```

5.5.3. Unpairing One Device - SUI Mode

DIFFERENCE! The following commands belong to the setting of **SUI mode** for LEX and REX devices only. If the devices are in MSA mode, see the [Pairing the Devices - MSA Mode](#) section for pairing.

DEFINITION: SUI mode: Simultaneous Users Interaction. This mode allows a single USB tier to be presented to the connected device.

The command makes disconnecting a LEX (transmitter) to a REX (receiver). The command is required calling **twice**, transmitter to receiver and receiver to transmitter as well.

Command and Response - TCI API

```
→ icron<transmitter>·unpair·<receiver>\r
← {"status":"<status>","request_id":"<request_ID>","result":"<result>","error":"<error>"}

→ icron<receiver>·unpair·<transmitter>\r
← {"status":"<status>","request_id":"<request_ID>","result":"<result>","error":"<error>"}

Parameters
```

Parameters

Parameter	Parameter description	Values	Parameter value
<transmitter>	The name or MAC address of the transmitter		Device name / MAC address
<receiver>	The name or MAC address of the receiver		Device name / MAC address

Example - TCI API

→ icron TX1 unpair RX1\r

← {"status":"PROCESSING","request_id":"2336","result":"null","error":"null"}

→ icron RX1 unpair TX1\r

← {"status":"PROCESSING","request_id":"2337","result":"null","error":"null"}

5.5.4. Unpairing All Devices - SUI Mode

DIFFERENCE! The following commands belong to the setting of **SUI mode** for LEX and REX devices only. If the devices are in MSA mode, see the [Pairing the Devices - MSA Mode](#) section for pairing.

DEFINITION: SUI mode: Simultaneous Users Interaction. This mode allows a single USB tier to be presented to the connected device.

The command removes all Icron pairings from the selected device.

Command and Response - TCI API

→ icron-<device>-unpair\r

← {"status":"<status>","request_id":"<request_ID>","result":"<result>","error":"<error>"}

Parameters

Parameter	Parameter description	Values	Parameter value
<device>	The name or MAC address of the transmitter or receiver		Device name / MAC address

Example - TCI API

→ icron RX1 unpair\r

← {"status":"PROCESSING","request_id":"2336","result":"null","error":"null"}

5.5.5. Pairing the Devices - MSA Mode

DIFFERENCE! The following commands belong to the setting of **MSA mode** for LEX and REX devices only. If the devices are in SUI mode, see the [Pairing the Devices - SUI Mode](#) section for pairing.

DEFINITION: MSA mode: Mass Storage Acceleration. This mode allows the connected device to see the USB devices connected to the extender as multiple, separate USB tiers.

The command makes pairing a LEX (transmitter) to one or multiple REXs (receiver(s)). The command is required calling **twice**, transmitter to receiver and receiver to transmitter as well.

Command and Response - TCI API

→ switch-<transmitter>:USB_ICRON:0-<receiver>:USB_ICRON:0

← {"status":"<status>","request_id":"<request_ID>","result":"<result>","error":"<error>"}

→ switch-<receiver>:USB_ICRON:0-<transmitter>:USB_ICRON:0

← {"status":"<status>","request_id":"<request_ID>","result":"<result>","error":"<error>"}

Parameters

Parameter	Parameter description	Values	Parameter value
<transmitter>	The name or MAC address of the transmitter		Device name / MAC address
<receiver>	The name or MAC address of the receiver		Device name / MAC address
		ALL_RX	All REX (receiver) devices

Example - TCI API

→ switch TX1:USB_ICRON:0 RX1:USB_ICRON:0

← {"status":"PROCESSING","request_id":"2336","result":"null","error":"null"}

→ switch RX1:USB_ICRON:0 TX1:USB_ICRON:0

← {"status":"PROCESSING","request_id":"2337","result":"null","error":"null"}

Command and Response - REST API

```
➔ request-line: POST·http://<ip>:<port>/api/device/<transmitter>
➔ body:      {
➔           "op" : "switch:device",
➔           "stream_type" : "USB_ICRON",
➔           "stream_index" : "0",
➔           "dest_device" : "<receiver>",
➔           "subscription_index" : 0
➔       }
← status-line: {
←           "status" : "<status>",
←           "request_id" : <request_ID>,
←           "result" : <result>,
←           "error" : <error>
←       }

➔ request-line: POST·http://<ip>:<port>/api/device/<receiver>
➔ body:      {
➔           "op" : "switch:device",
➔           "stream_type" : "USB_ICRON",
➔           "stream_index" : "0",
➔           "dest_device" : "<transmitter>",
➔           "subscription_index" : 0
➔       }
← status-line: {
←           "status" : "<status>",
←           "request_id" : <request_ID>,
←           "result" : <result>,
←           "error" : <error>
←       }
```

Parameters

Parameter	Parameter description	Values	Parameter value
<ip>	The IP address of the transmitter		IP address
<port>	Port number	8080	Port number
<transmitter>	The name or MAC address of the transmitter		Device name / MAC address
<receiver>	The name or MAC address of the receiver		Device name / MAC address
		ALL_RX	All REX (receiver) devices

Example - REST API

```
➔ request-line: POST·http://169.254.11.134:8080/api/device/TX1
➔ body:      {
➔           "op" : "switch:device",
➔           "stream_type" : "USB_ICRON",
➔           "stream_index" : "0",
➔           "dest_device" : "RX1",
➔           "subscription_index" : 0
➔       }
← status-line: {
←           "status" : "PROCESSING",
←           "request_id" : 2336,
←           "result" : null,
←           "error" : null
←       }

➔ request-line: POST·http://169.254.11.134:8080/api/device/RX1
➔ body:      {
➔           "op" : "switch:device",
➔           "stream_type" : "USB_ICRON",
➔           "stream_index" : "0",
➔           "dest_device" : "TX1",
➔           "subscription_index" : 0
➔       }
← status-line: {
←           "status" : "PROCESSING",
←           "request_id" : 2337,
←           "result" : null,
←           "error" : null
←       }
```

5.5.6. Setting the USB HID Mode

The command is used to configure a device(s) to serve as USB HID extender(s) and stop any active USB HID streams.

Command and Response - TCI API

```
→ set-<transmitter>·usb·role·<usb_hid_mode>
← {"status":"<status>","request_id":"<request_ID>","result":"<result>","error":"<error>"}
```

Parameters

Parameter	Parameter description	Values	Parameter value
<transmitter>	The name or MAC address of the receiver		Device name / MAC address
		ALL_TX	All LEX (transmitter) devices
<usb_hid_mode>	USB HID mode of the LEX	DISABLED	USB HID support is disabled
		OTG	The device automatically detects the type of the connected USB HID device and set its role to LOCAL or REMOTE accordingly.
		LOCAL	The device acts as a USB device that connects to a USB host (e.g. a PC).
		REMOTE	The device acts as a USB host to which USB HID devices are connected (keyboard, mouse, etc.)

Example - TCI API

```
→ set TX1 usb role local
← {"status":"PROCESSING","request_id":"1059","result":"null","error":"null"}
```

Command and Response - REST API

```
➔ request-line: POST·http://<ip>:<port>/api/device/<transmitter>
➔ body: {
➔     "op" : "set:usb",
➔     "role" : <usb_hid_mode>
➔ }
← status-line: {
←     "status" : "<status>",
←     "request_id" : <request_ID>,
←     "result" : <result>,
←     "error" : <error>
← }
```

Parameters

Parameter	Parameter description	Values	Parameter value
<ip>	The IP address of the transmitter		IP address
<port>	Port number	8080	Port number
<transmitter>	The name or MAC address of the receiver		Device name / MAC address
		ALL_TX	All LEX (transmitter) devices
<usb_hid_mode>	USB HID mode of the LEX	DISABLED	USB HID support is disabled
		OTG	The device automatically detects the type of the connected USB HID device and sets its role to LOCAL or REMOTE accordingly.
		LOCAL	The device acts as a USB device that connects to a USB host (e.g. a PC).
		REMOTE	The device acts as a USB host to which USB HID devices are connected (keyboard, mouse, etc.)

Example - REST API

```
➔ request-line: POST·http://169.254.11.134:8080/api/device/TX1
➔ body: {
➔     "op" : "set:usb",
➔     "role" : LOCAL
➔ }
← status-line: {
←     "status" : "PROCESSING",
←     "request_id" : 1059,
←     "result" : null,
←     "error" : null
← }
```