How to use the Mira Connect Room Control API with KeyDigital KD-WP8-2

Introduction

Mira Connect is an award-winning AV control system that makes it easy to build professional AV control systems. Mira Connect supports thousands of devices, making it easy to use in many different applications. Mira Connect also includes built-in equipment emulators, making it possible to create and experience, a full system before purchasing any AV equipment.

Mira Connect is configured through Mira Portal, the secure cloud management platform for Mira Connect that enables remote management for AV/IT teams.

In this guide we'll show how to use Mira Connect software, Mira Portal, and a KeyDigital KD-WP8-2 8-button keypad as the user interface to control a room using Mira Connect's room control API.

This guide shows you how to configure the keypad for volume and mute control, source selection, and a room power-off feature, as shown below. KeyDigital keypads include multiple sets of informative button labels and clear key caps that cover the button and labels to help users understand what each button does.

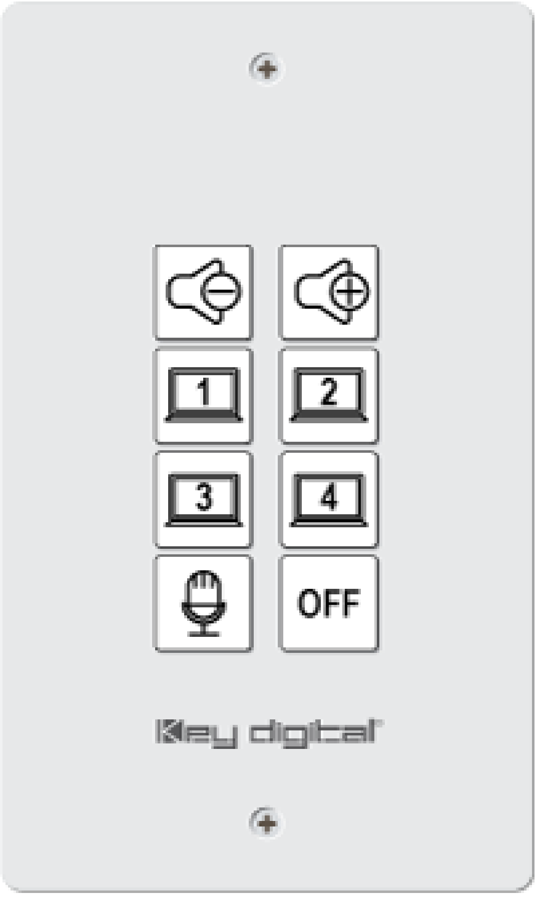


Figure 1: The button layout for the KeyDigital keypad design.

This guide is organized into four main sections:

- Getting Started with Mira Portal and Mira Connect software shows how to create a Mira Portal account; how to create an organization, site, and room in Mira Portal; and how to add the equipment for this example to the room in Mira Portal.
- Controlling Mira Connect shows how to create an API Key and how to get the display ID and source IDs required to switch video sources to a display.
- Configuring the KeyDigital keypad shows how to define the buttons and how to define the events for the KeyDigital keypad. This section will provide all the details you need for configuring the keypad for this example.
- Using the KeyDigital keypad shows how to confirm operation, test, and troubleshoot the keypad's configuration.

Other Things You Can Do While this article details one specific keypad and one application, the information presented here can be used to help you create an unlimited number of additional applications, create different keypad configurations with different functions assigned to buttons, or use different brands of keypads with Mira Connect's room control API.

Multiple keypads can also be used with Mira Connect software to provide multiple points of control in the room or to control different features of the room from different keypads. There is no end to the possibilities for using the Mira Connect Room Control API once you get started.

Now, let's get started!

Getting Started with Mira Connect

The Mira Connect AV control system is configured using Mira Portal, a secure cloud-based configuration and management platform. Mira Connect software includes an application programming interface (API) for controlling a room by sending simple and easy-to-understand commands to the Mira Connect software while Mira Connect does the work of controlling the underlying equipment.

Mira Connect Room Control API

Mira Connect's room control API is an application programming interface that allows control of Mira Connect from other applications. Mira Connect is **the first control system that allows itself to be easily controlled**, providing tremendous power and flexibility to programmers to add AV control into their applications or to use other user interface options for interacting with the room.

The room control API allows applications, or other devices, to send commands to Mira Connect over a network connection using **web sockets** or **TCP connections**. Mira Connect receives the commands, controls the underlying AV devices and reports status back to the application.

The benefits of using Mira Connect's Room Control API include:

- Enables new applications and user interfaces while lowering the cost for adding room control to any AV application.
- Reduces setup time, complexity, and troubleshooting for creating new applications because it is not necessary to know the API syntax of every device you want to control. The Mira Connect room control API works the same regardless of the underlying equipment used in the room since Mira Connect handles all the underlying device-specific details.
- Allows you to extend the capabilities of simple user interfaces because Mira Connect can control equipment over TCP, UDP, HTTP, HTTPS, SSH, and perform advanced features that aren't easy (or even possible) to configure on simple user interface devices, including sending Wake-on-LAN packets to displays, scheduling room power-off, or simply

synchronizing mute states across multiple devices.

- Users have more flexibility to create custom user interfaces to meet the needs of their customers. Need a low-cost way to make it easy for users to switch sources while at a podium? Simply add a keypad.
- In addition to any custom user interface, **users can always access Mira Connect's professionally-designed native interface through the browser of their PC or mobile device**.

Here's What We'll Be Setting Up In this guide, we'll show how to use the KeyDigital KD-WP8-2 keypad's eight buttons as the user interface for controlling volume up and down, four video source selections, microphone mute, and to power off the room, as shown in the following figure.

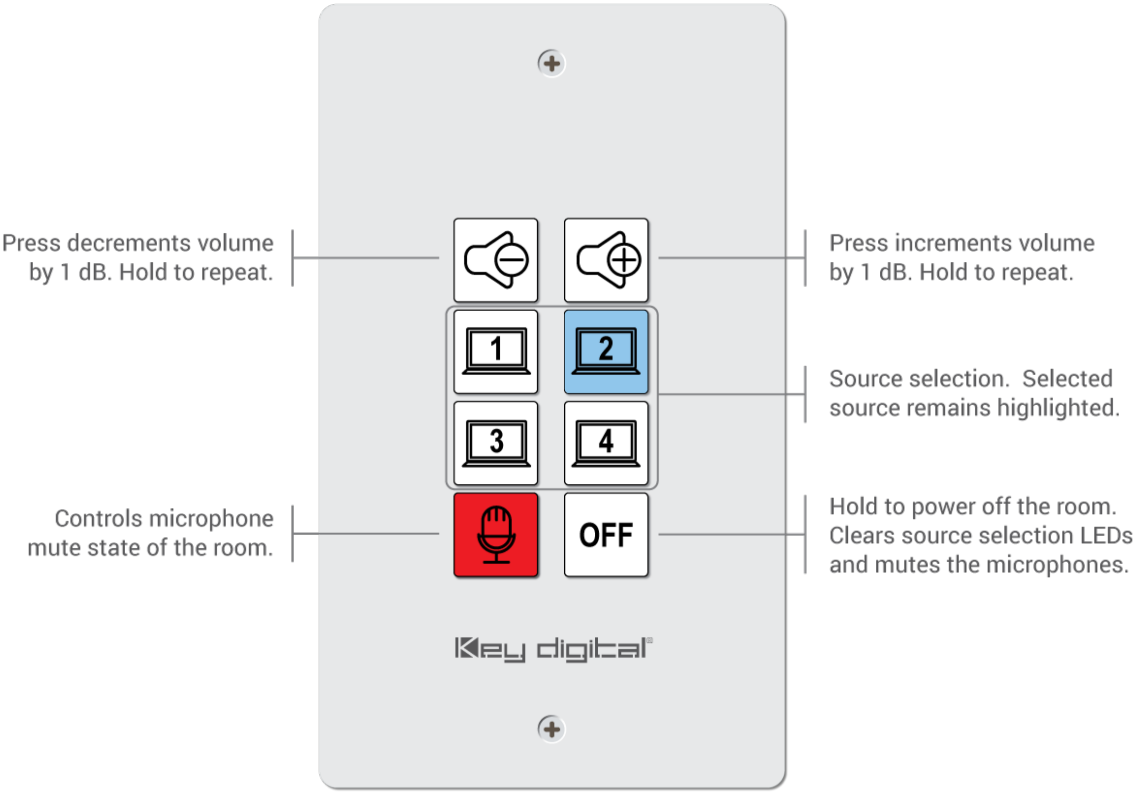

Press decrements volume by 1 dB. Hold to repeat.
Press increments volume by 1 dB. Hold to repeat.
1
2
3
4
Source selection. Selected source remains highlighted.
Controls microphone mute state of the room.
OFF
Hold to power off the room. Clears source selection LEDs and mutes the microphones.
Key digital

Figure 2: The proposed button layout for the KeyDigital keypad design.

The eight buttons will be configured with the following functionality:

Button	Behavior	Description
1	Volume Decrement	A single press will lower the volume by 1dB. Holding the button will continue to decrease the volume. The button will illuminate red when pressed.
2	Volume Increment	A single press will increase the volume by 1dB. Holding the button will continue to increase the volume. The button will illuminate blue when pressed.

Button	Behavior	Description
3	Select source 1	When the button is pressed, source 1 is selected on the display. The button will illuminate blue when pressed Only one source button will be illuminated at any time.
4	Select source 2	When the button is pressed, source 2 is selected on the display. The button will illuminate blue when pressed Only one source button will be illuminated at any time.
5	Select source 3	When the button is pressed, source 3 is selected on the display. The button will illuminate blue when pressed Only one source button will be illuminated at any time.
6	Select source 4	When the button is pressed, source 4 is selected on the display. The button will illuminate blue when pressed Only one source button will be illuminated at any time.
7	Microphone mute	The button will mute all the microphones on the first press and illuminate red. On the second press, all the microphones will be unmuted and the LED will be turned off.
8	Room power off	The button must be held for 2 seconds before the room will be powered off. The button will illuminate red initially and after two seconds will turn blue. After two seconds all LEDs will be turned off.

If your application requires different functionality, you can follow these steps and further customize the keypad configuration and room control commands used.

We'll use the following steps to build our system with Mira Connect software as the control engine and a KeyDigital keypad as the user interface:

Step 1. Create a Mira Portal account or login to Mira Portal ***Step 2.*** Add an Organization, Site, and Room in Mira Portal ***Step 3.*** Install Mira Connect software on a local computer ***Step 4.*** Pair the Mira Connect software to the room ***Step 5.*** Add the equipment to the room ***Step 6****.* Create an API Key and enable TCP control for Mira Connect ***Step 7.*** Define the buttons on the KeyDigital keypad ***Step 8.*** Configure the button events on the KeyDigital keypad ***Step 9.*** Test the keypad configuration ***Step 10****.* If necessary, troubleshoot the keypad configuration

You'll be able to create a configured keypad in less than 30 minutes the first time you do it. You will be more efficient on subsequent applications of the Mira Connect software and KeyDigital keypads.

Next, let's create a Mira Portal account or log in to an existing account.

Step 1 — Create a Mira Portal Account

If you don't already have a Mira Portal account, create a no-cost Mira Portal account. Otherwise, log in to your account. A valid email address is required to create an account.

Once a Mira Portal account has been created, follow the tutorial for creating an organization, site, and room. There are also training videos for how to create a Mira Portal account and how to add equipment to the room.

> NOTE: As we go through this example, **you can follow along using simulated equipment** and confirm the operation with our Mira Connect **Remote Control user interface** in your browser.

While there is no cost to creating a Mira Portal account and for using Mira Connect room control API with simulated equipment, to use Mira Connect software with *actual* equipment, a room license is required and may be purchased through Mira Portal.

See our licensing information for more information and options for controlling real equipment.

Step 2 — Add an Organization, Site, and Room in Mira Portal

In Mira Portal, create an organization, and then add one or more sites to the organization. The tutorial that starts automatically will show you how to do this.

For each site, you can add one or more rooms. In this example, we'll create an organization called **Example Organization**, a site called **My Site**, and a room called **Example Room**.

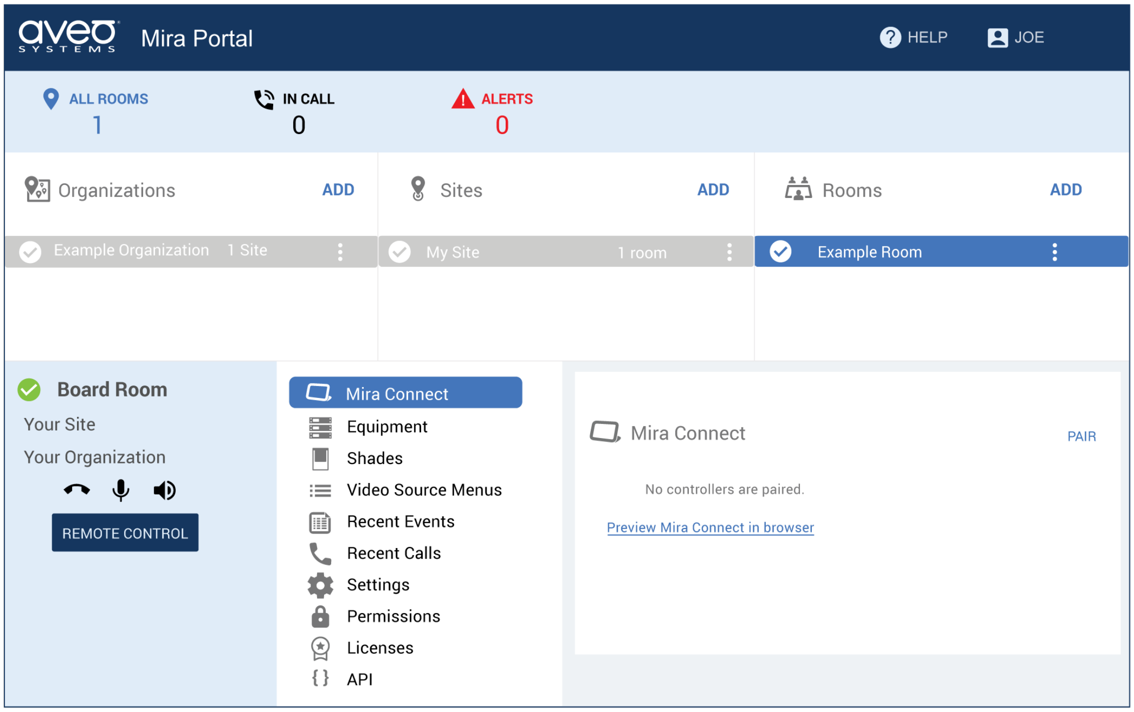


Figure 3: Use Mira Portal to create the organization, site, and room.

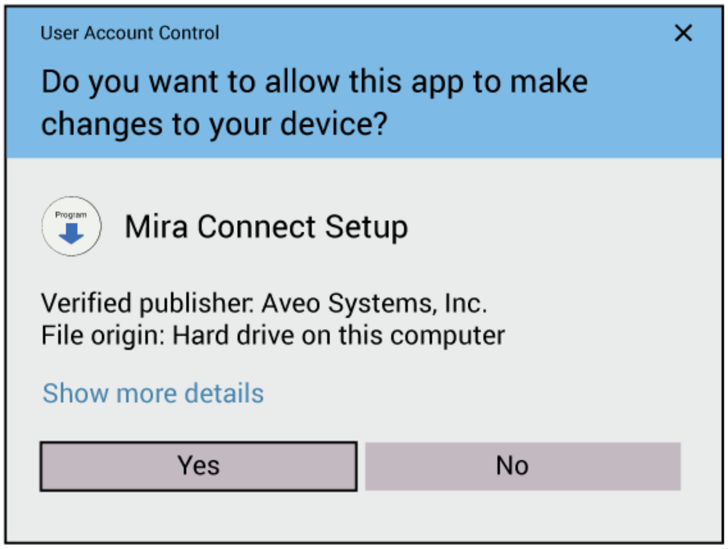
Once we have created the room, we can install Mira Connect software, pair Mira Connect software to the room, and then add the equipment to control, as described in the following sections.

For more information on the equipment supported by Mira Connect and step-by-step instructions for how to add specific equipment to Mira Portal, see our integration guides.

Step 3 — Install Mira Connect Software

Mira Connect software is a software-only version of the Mira Connect AV control system that can be installed on any Windows® 10 or newer computer. Mira Connect software can be downloaded from AveoSystems.com.

Once downloaded, follow the instructions in the Mira Connect Software quick install guide to install the Mira Connect software.



User Account Control

Do you want to allow this app to make changes to your device?

Mira Connect Setup

Verified publisher: Aveo Systems, Inc.
File origin: Hard drive on this computer

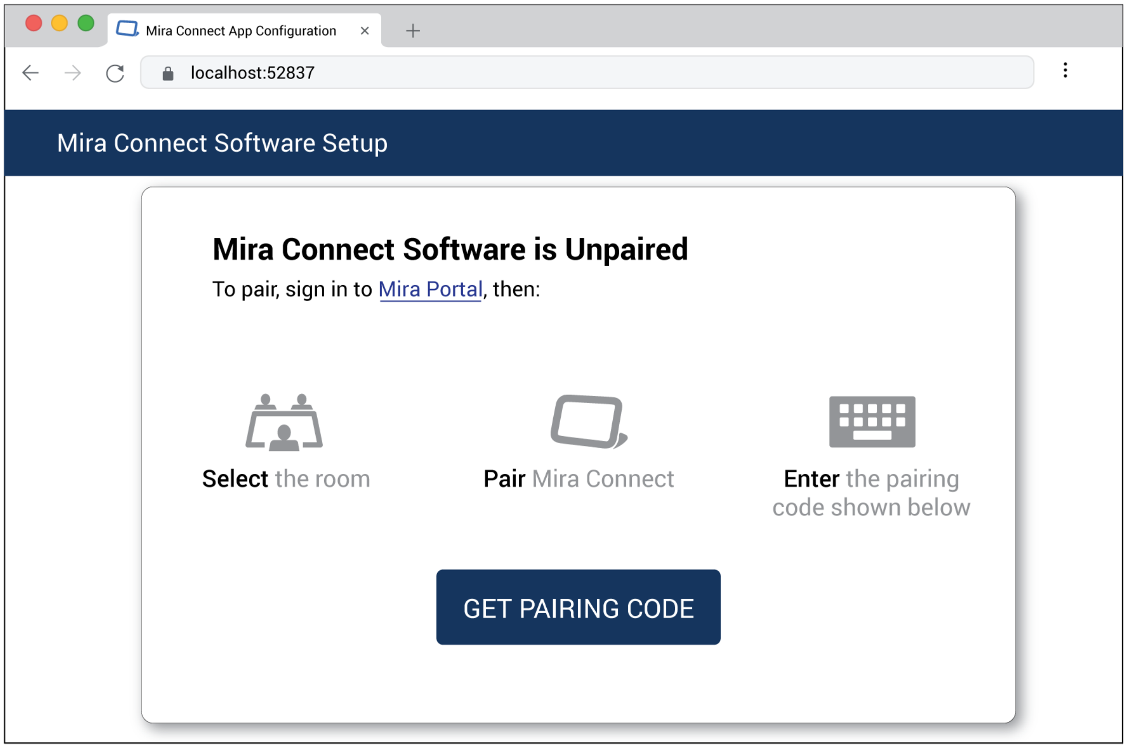
Figure 4: Install Mira Connect software on a local PC.

Mira Connect software installs as a *service* and runs in the background, automatically restarting if the computer is rebooted. Mira Connect software also periodically checks for updates and will update itself automatically when there are new releases.

> NOTE: Mira Connect software requires WAN access in order to be configured through Mira Portal. For more information on how to add WAN to your network, see our IT networks for AV professionals.

Step 4 — Pair Mira Connect to the Room

Once Mira Connect software has been installed, browse into the Mira Connect software at *localhost: 52837* and click **Get Pairing Code** to get a pairing code that will let us associate Mira Connect with the desired room in Mira Portal.



Mira Connect Software Setup

Mira Connect Software is Unpaired

To pair, sign in to Mira Portal, then:

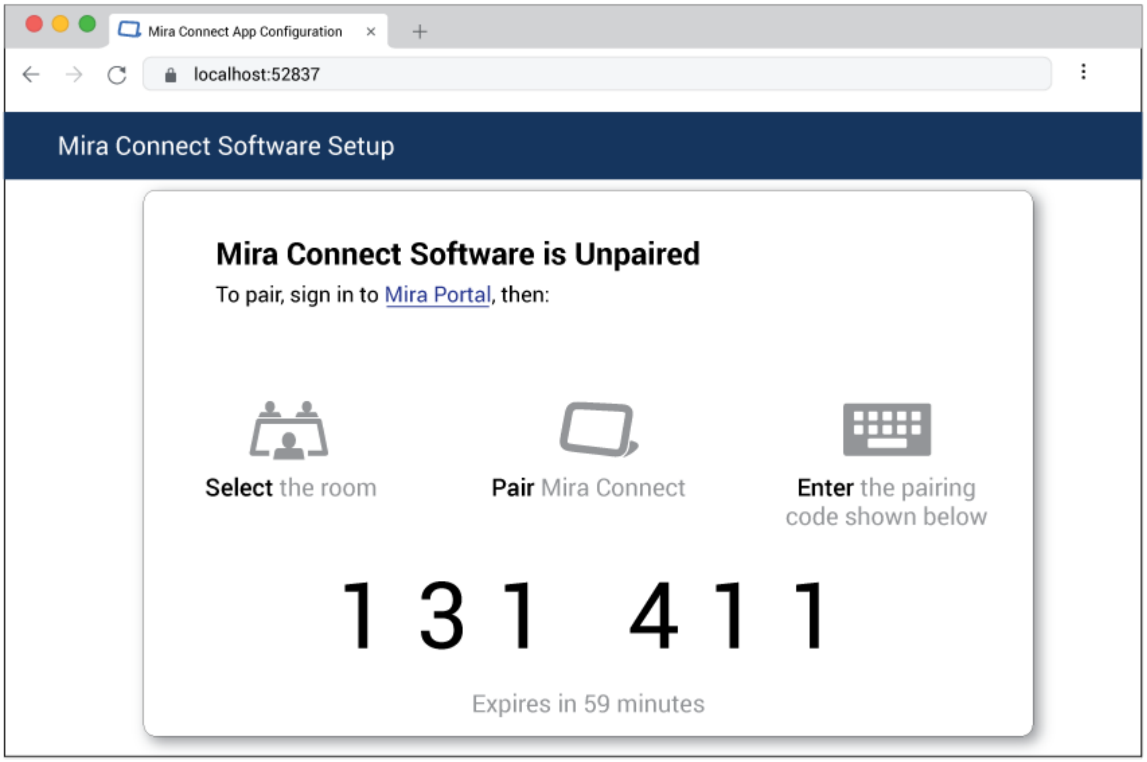
Select the room

Pair Mira Connect

Enter the pairing code shown below

Figure 5: Browse into Mira Connect to get a pairing code.

Mira Connect software will connect to Mira Portal and get a unique pairing code. Note the pairing code will expire after an hour, after which a new pairing code can be requested.



Mira Connect Software is Unpaired

To pair, sign in to Mira Portal, then:

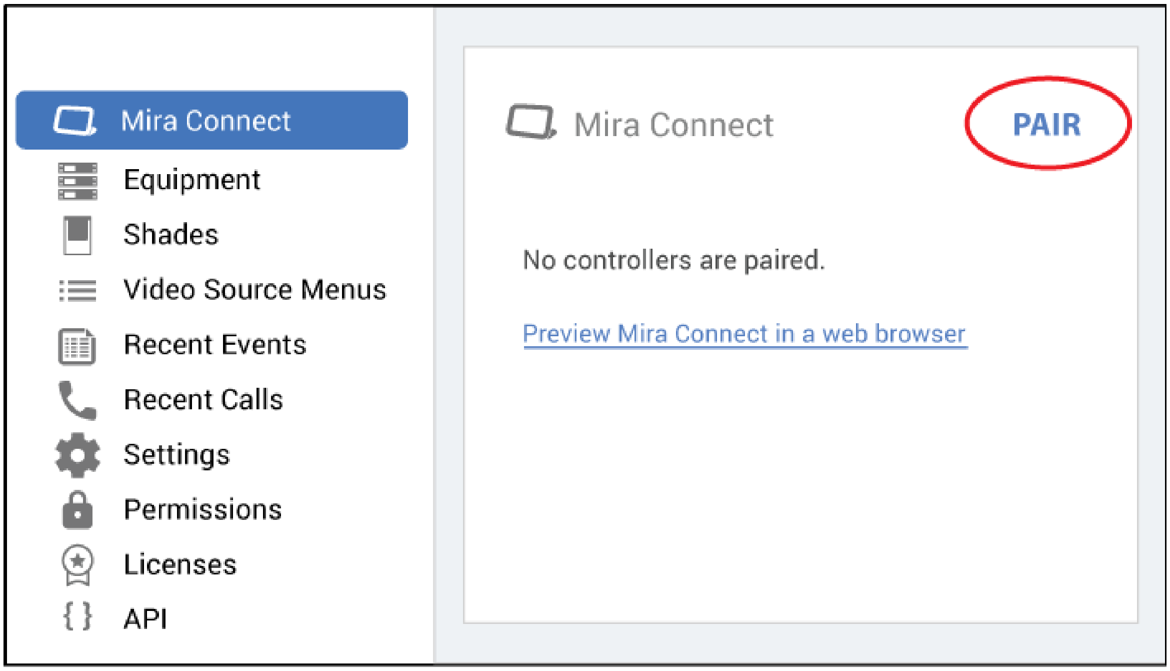
1. **Select** the room
2. **Pair** Mira Connect
3. **Enter** the pairing code shown below

1 3 1 4 1 1

Expires in 59 minutes

Figure 6: A pairing code will be used with Mira Portal to associate the Mira Connect with the room.

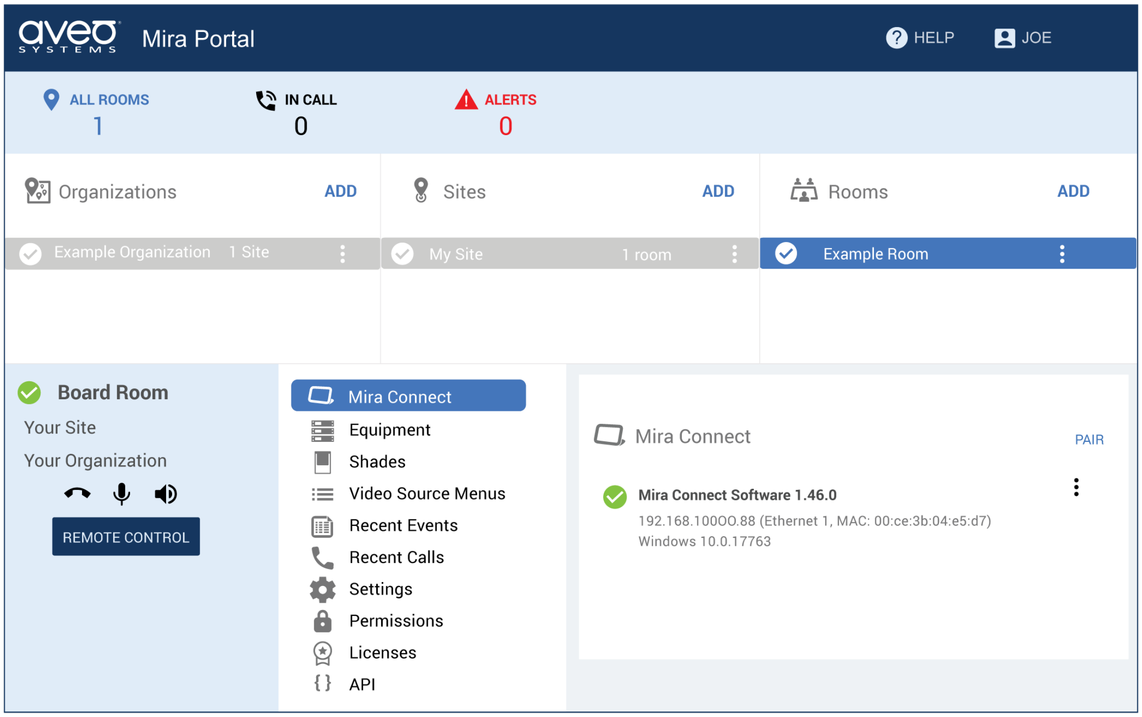
Enter the pairing code into Mira Portal by clicking the PAIR button on the Mira Connect tab.



No controllers are paired.

Figure 7: Click the PAIR button to enter the pairing code in Mira Portal.

Once paired to the room, Mira Portal will show the updated status, as shown in the following figure.



Mira Connect Software 1.46.0
192.168.10000.88 (Ethernet 1, MAC: 00:ce:3b:04:e5:d7)
Windows 10.0.17763

Figure 8: Once paired to the room, Mira Portal shows information about the Mira Connect software.

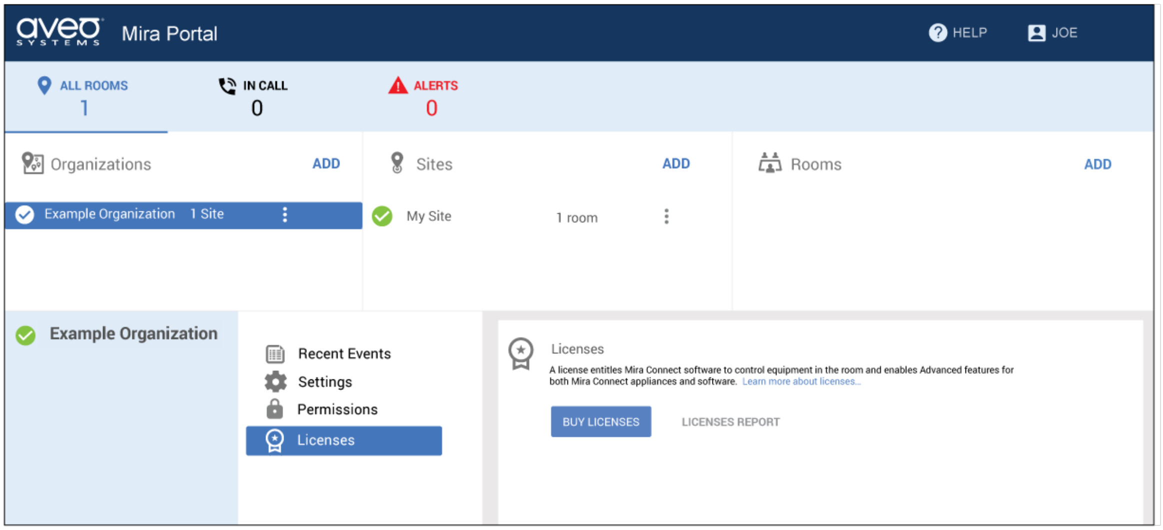
Step 5 — Add Equipment to the Room

Continuing with this example, we'll use the following equipment as part of our room:

- A Yamaha ADECIA DSP for mute and volume control
- An AVPro video switcher for video switching multiple sources to a display
- A Sharp display for displaying video sources in the room

Mira Connect makes it easy to use devices from a wide variety of manufacturers, allowing you to use the equipment that you are familiar with or to meet cost and delivery requirements.

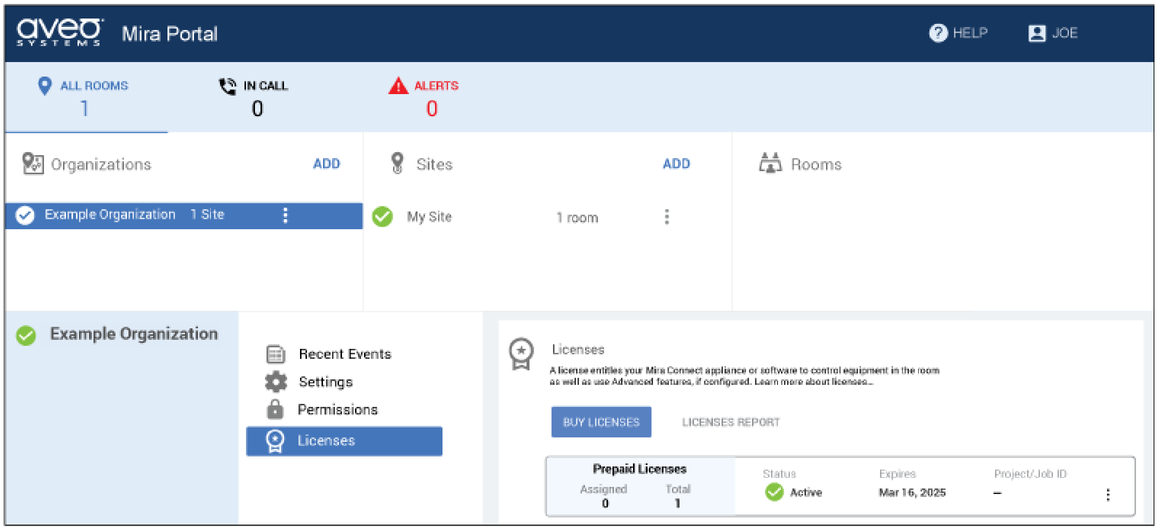
Mira Connect software requires an advanced room license to control actual equipment in the room. An advanced room license may be purchased on Mira Portal for an organization and then assigned to the room. If you don't have a valid advanced license assigned to the room, you must choose to simulate the equipment to continue.



A license entitles Mira Connect software to control equipment in the room and enables Advanced features for both Mira Connect appliances and software. Learn more about licenses...

Figure 9: Select the organization name and then the licensing menu to buy a license.

Once a room license has been purchased, the Licenses menu will show the new license information.



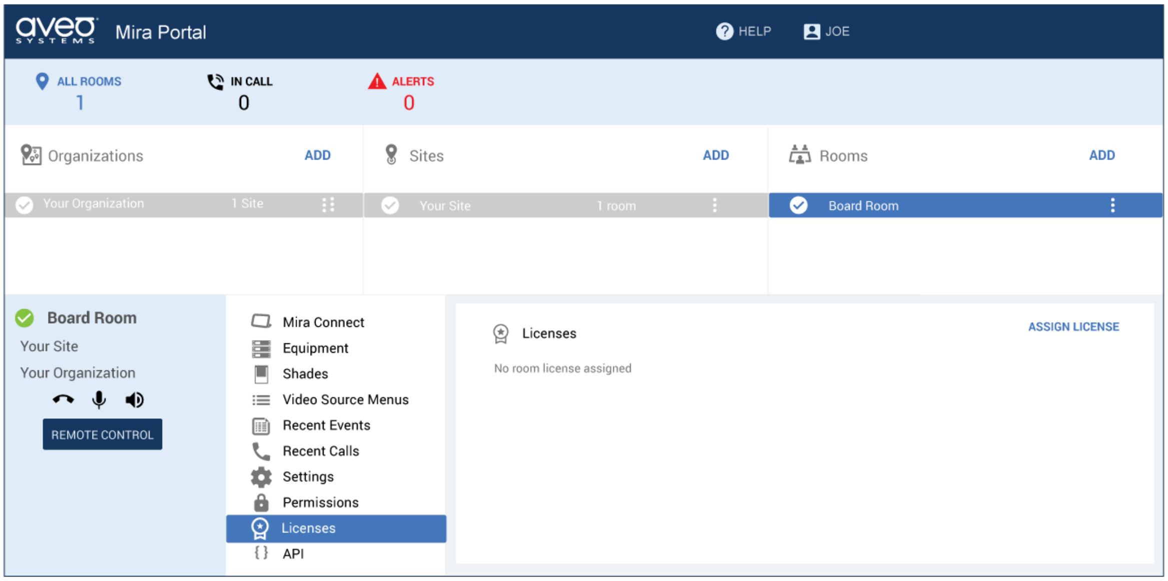
Licenses

A license entitles your Mira Connect appliance or software to control equipment in the room as well as use Advanced features, if configured. Learn more about licenses...

Prepaid Licenses		Status	Expires	Project/Job ID
Assigned	Total			
0	1	✓ Active	Mar 16, 2025	–

Figure 10: After purchasing the license, the license status is updated.

Navigate to the room and assign the license by clicking **ASSIGN LICENSE** and select the recently purchased license.

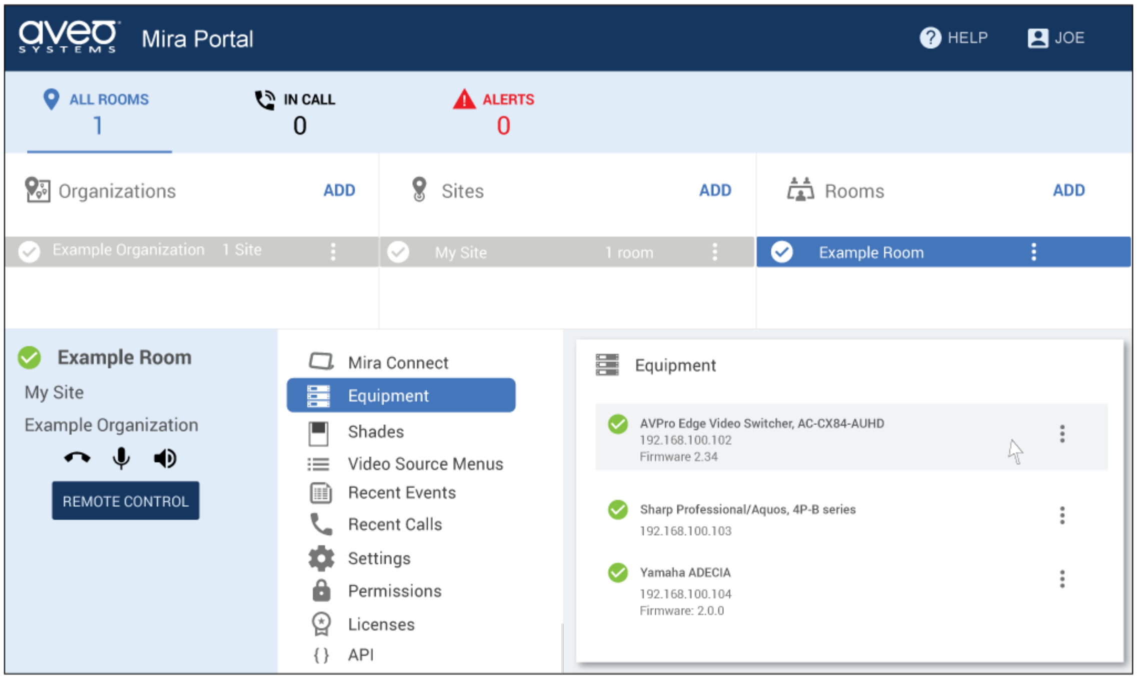


No room license assigned

Figure 11: Once assigned to the room, the license will appear in the room's licenses menu.

> NOTE: While this application focuses on audio control and video source selection, it would be just as easy to control cameras (for example, recalling camera presets), launching speed dials to audio or video conferencing systems (including launching Zoom meetings), controlling lights and/or shades, selecting channel favorites on a TV Tuner, starting/stopping recording or streaming, and more.

With the Mira Connect Room Control API, regardless of the equipment you want to control in the room, the Mira Connect API commands *are always the same*. This makes it easy to use your favorite equipment without having to worry about the specific API commands needed to control that equipment. Mira Connect takes care of that for you.



AVPro Edge Video Switcher, AC-CX84-AUHD
192.168.100.102
Firmware 2.34

Sharp Professional/Aquos, 4P-B series
192.168.100.103

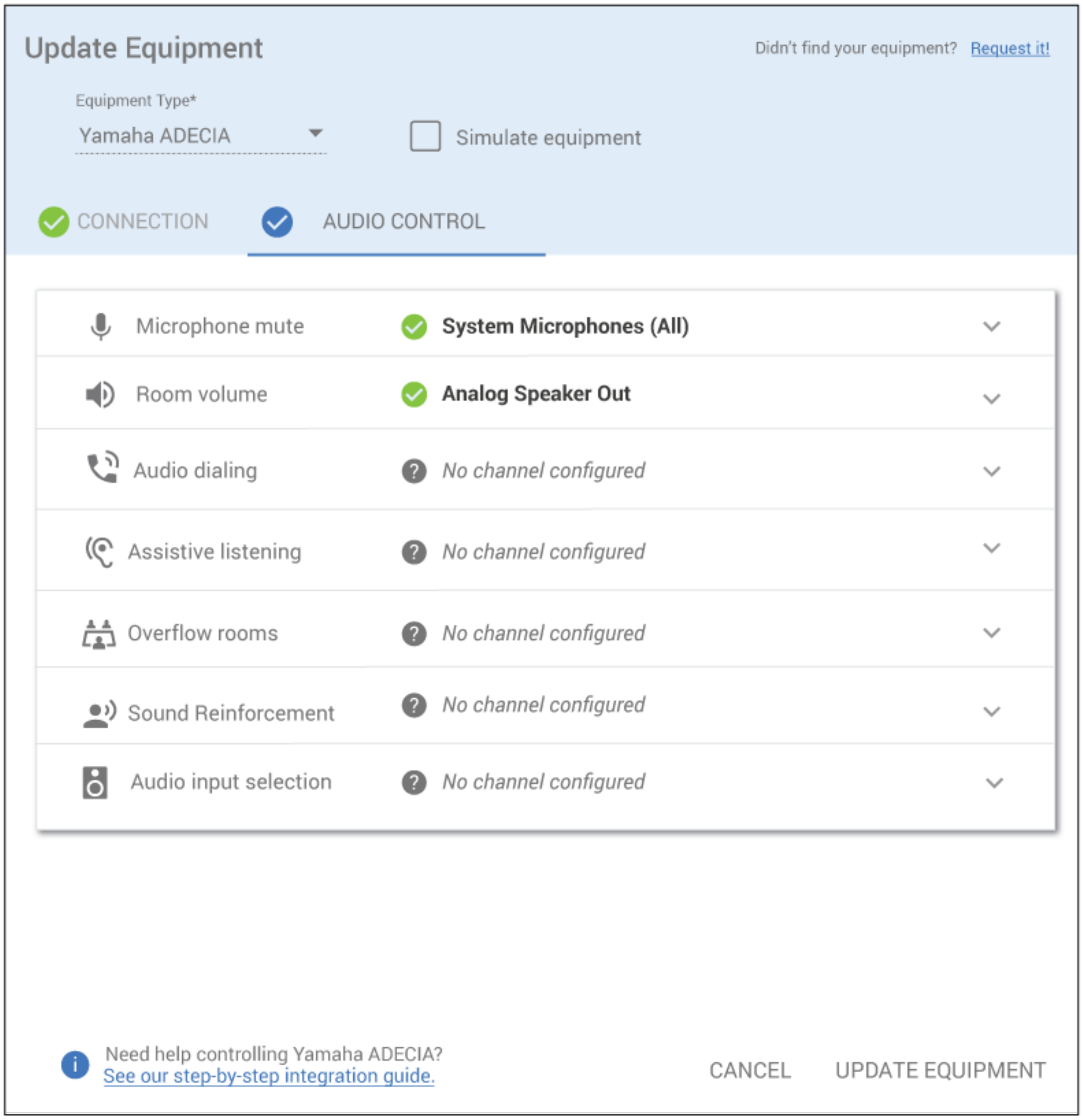
Yamaha ADECIA
192.168.100.104
Firmware: 2.0.0

Figure 12: Using Mira Portal, select and add the equipment to the room and then configure the equipment options.

The Yamaha DSP will be configured with a microphone mute and volume control, as shown in the following figure. In this example, we're using an array microphone that can be mounted in the ceiling and an analog output signal that will feed the loudspeakers in the room.

> NOTE: For each family of products supported, Mira Portal provides a link to an integration guide with step-by-step instructions for how to configure the settings of that particular equipment in Mira Portal.

As mentioned earlier, if you don't have the equipment or a room license yet, you can choose **Simulate Equipment** to build the user interface and control system around Mira Connect's built-in equipment emulators.



Update Equipment

Didn't find your equipment? Request it!

Equipment Type*: Yamaha ADECIA

□ Simulate equipment

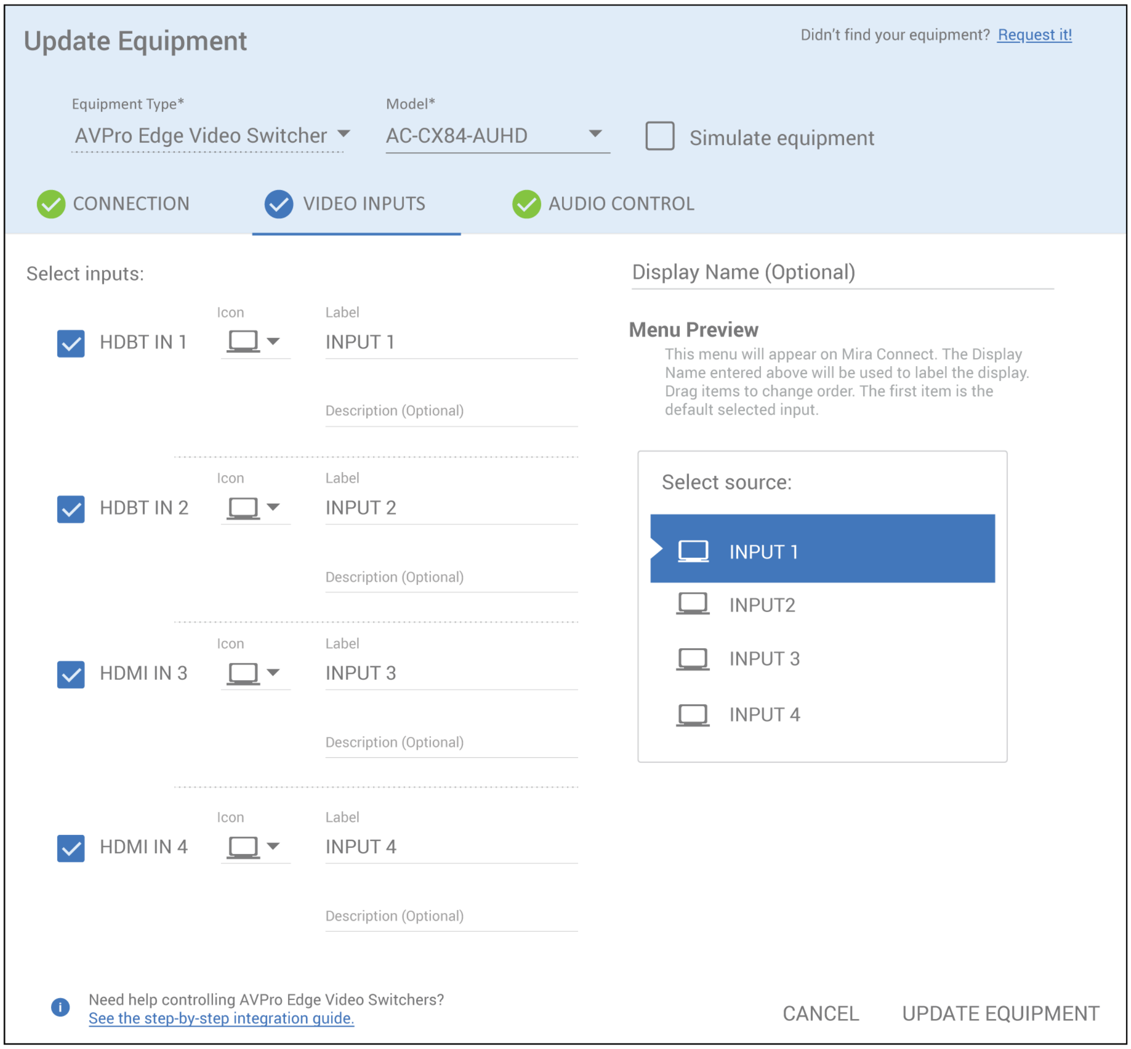
✓ CONNECTION ✓ AUDIO CONTROL

Microphone mute	✓	**System Microphones (All)**
Room volume	✓	**Analog Speaker Out**
Audio dialing	?	*No channel configured*
Assistive listening	?	*No channel configured*
Overflow rooms	?	*No channel configured*
Sound Reinforcement	?	*No channel configured*
Audio input selection	?	*No channel configured*

Need help controlling Yamaha ADECIA? See our step-by-step integration guide.

Figure 13: The Yamaha ADECIA audio system configured with a microphone and room volume control.

The AVPro video switcher is configured with four inputs that we'll label as **INPUT 1**, **INPUT 2**, **INPUT 3**, and **INPUT 4**.



Update Equipment

Didn't find your equipment? Request it!

Equipment Type*: AVPro Edge Video Switcher

Model*: AC-CX84-AUHD

□ Simulate equipment

✓ CONNECTION ✓ VIDEO INPUTS ✓ AUDIO CONTROL

Select inputs:

	Icon	Label
☑ HDBT IN 1	🖵	INPUT 1 Description (Optional)
☑ HDBT IN 2	🖵	INPUT 2 Description (Optional)
☑ HDMI IN 3	🖵	INPUT 3 Description (Optional)
☑ HDMI IN 4	🖵	INPUT 4 Description (Optional)

Display Name (Optional)

Menu Preview

This menu will appear on Mira Connect. The Display Name entered above will be used to label the display. Drag items to change order. The first item is the default selected input.

Select source:

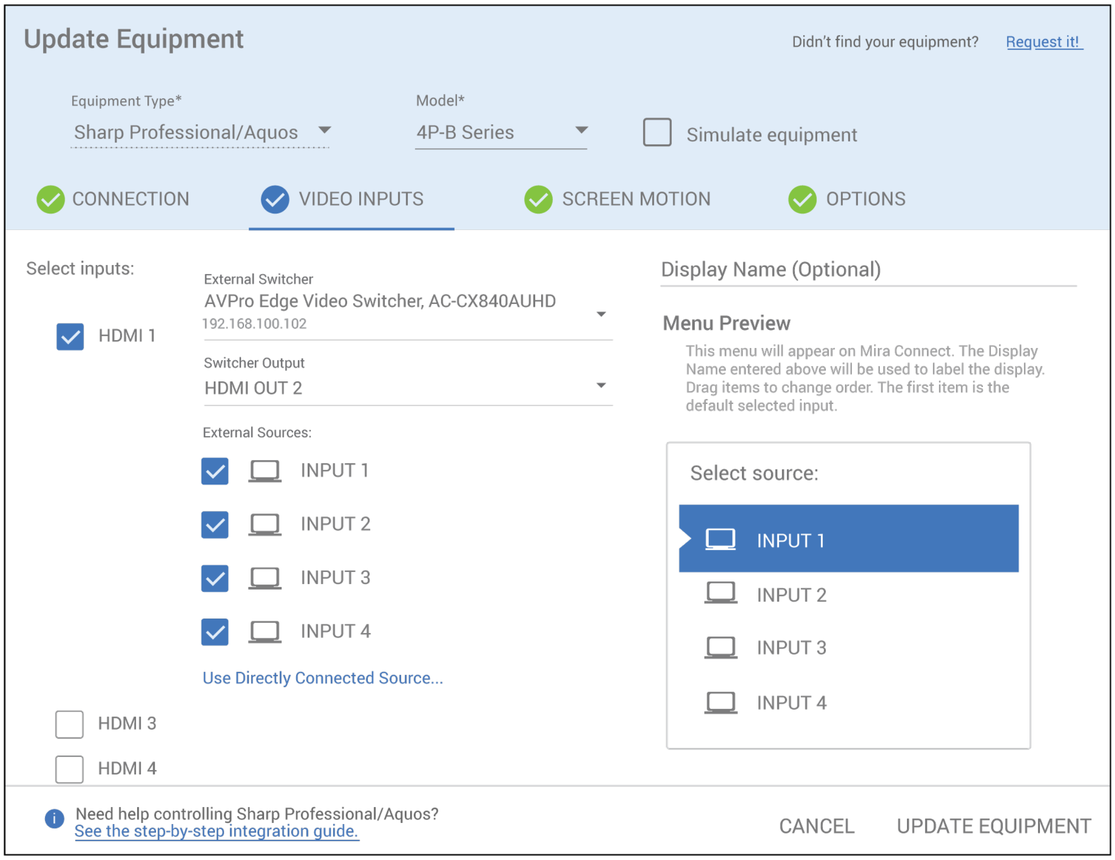
- INPUT 1
- INPUT2
- INPUT 3
- INPUT 4

Need help controlling AVPro Edge Video Switchers? See the step-by-step integration guide.

Figure 14: The AVPro video switcher has 4 inputs enabled.

And finally, the Sharp display will be configured to use the HDMI 2 output from the video switcher connected to HDMI 1 input of the display.

> NOTE: We used the 'Use sources connected to…' option on the display menu to access the output from the video switcher. See our training videos for more information about adding equipment and using video switchers with displays.



Update Equipment

Didn't find your equipment? Request it!

Equipment Type*: Sharp Professional/Aquos

Model*: 4P-B Series

□ Simulate equipment

✓ CONNECTION ✓ VIDEO INPUTS ✓ SCREEN MOTION ✓ OPTIONS

Select inputs:

☑ HDMI 1

External Switcher: AVPro Edge Video Switcher, AC-CX840AUHD
192.168.100.102

Switcher Output: HDMI OUT 2

External Sources:

☑ INPUT 1

☑ INPUT 2

☑ INPUT 3

☑ INPUT 4

Use Directly Connected Source...

□ HDMI 3

□ HDMI 4

Display Name (Optional)

Menu Preview

This menu will appear on Mira Connect. The Display Name entered above will be used to label the display. Drag items to change order. The first item is the default selected input.

Select source:

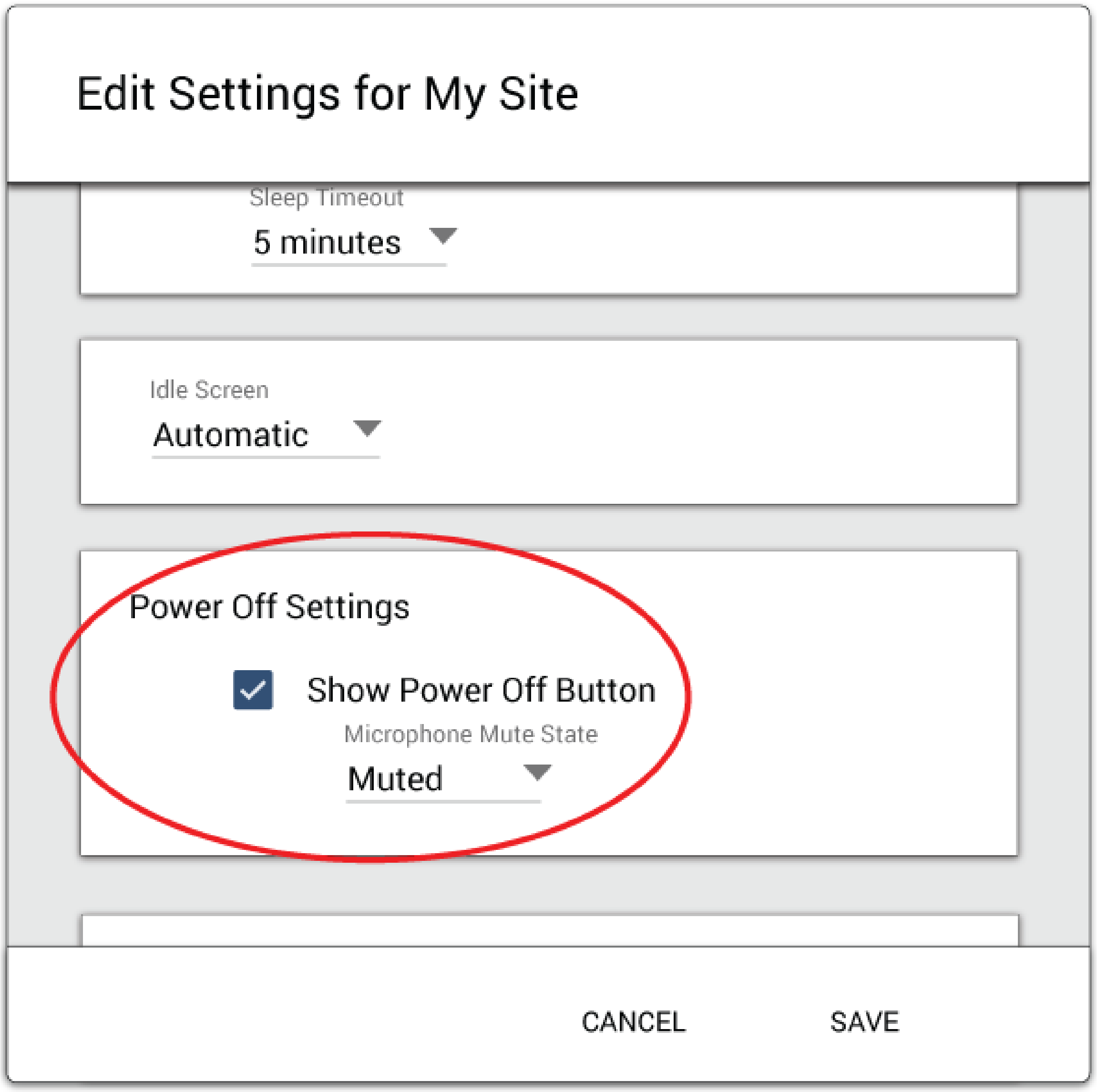
- INPUT 1
- INPUT 2
- INPUT 3
- INPUT 4

Need help controlling Sharp Professional/Aquos?
See the step-by-step integration guide.

CANCEL UPDATE EQUIPMENT

Figure 15: Sharp display with input HDMI 1 enabled with the four inputs from the video switcher available as sources.

Finally, once the equipment has been configured, we'll edit the site settings for the 'My Site' site and enable the 'Show Power Off Setting' feature, as shown below. While you don't need this to use the `set roomOff` API command, we enable it so that we can see the 'Room power off' function in the remote control UI.



Edit Settings for My Site

Sleep Timeout: 5 minutes

Idle Screen: Automatic

Power Off Settings

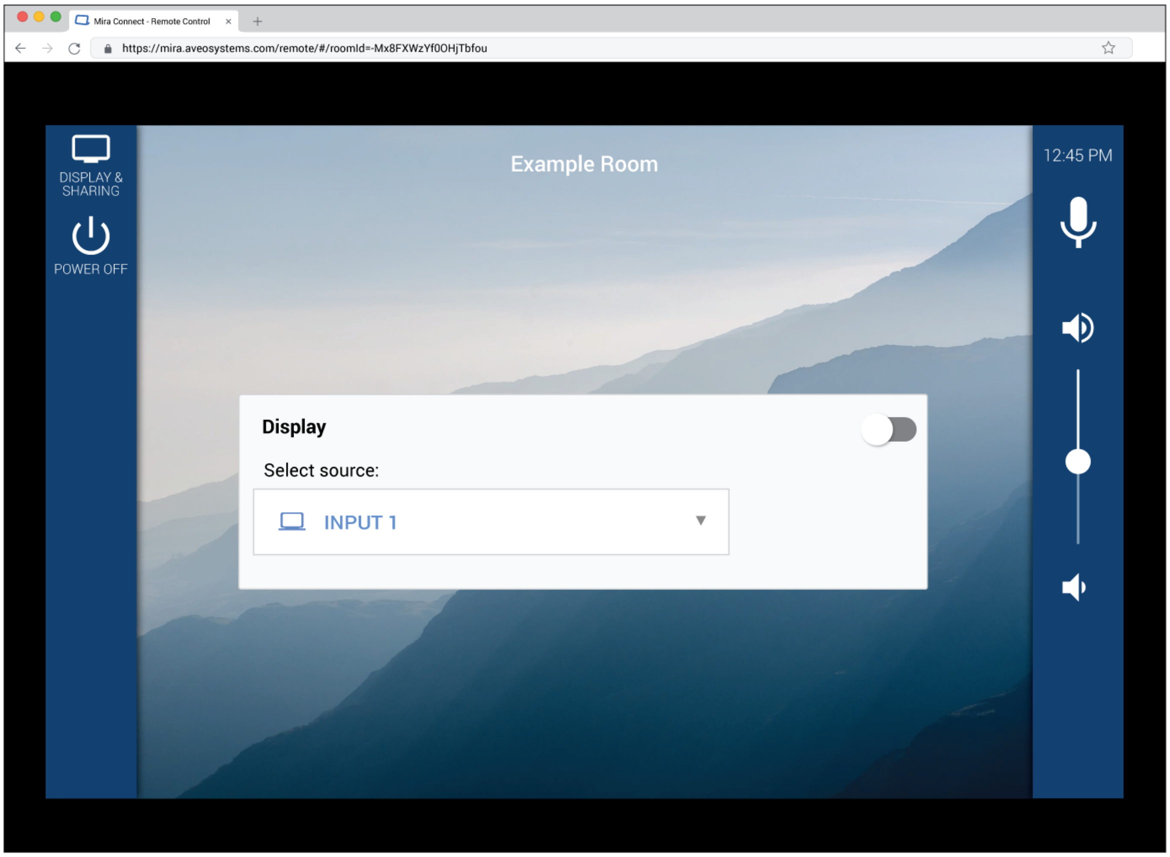
☑ Show Power Off Button

Microphone Mute State: Muted

Figure 16: Edit the site settings to enable the 'Room power off' button on the Mira Connect remote control user interface.

Once equipment has been added to the room, Mira Connect's user interface is *created automatically* and available via your browser by clicking the 'Remote Control' button.

The Remote Control feature makes it easy for AV and IT teams to manage rooms remotely from any location since they get access to the full user interface in the room and can remotely control the system.


Mira Connect - Remote Control
https://mira.aveosystems.com/remote/#/roomId=-Mx8FXWzYf0OHjTbfou
DISPLAY & SHARING
POWER OFF
Example Room
12:45 PM
Display

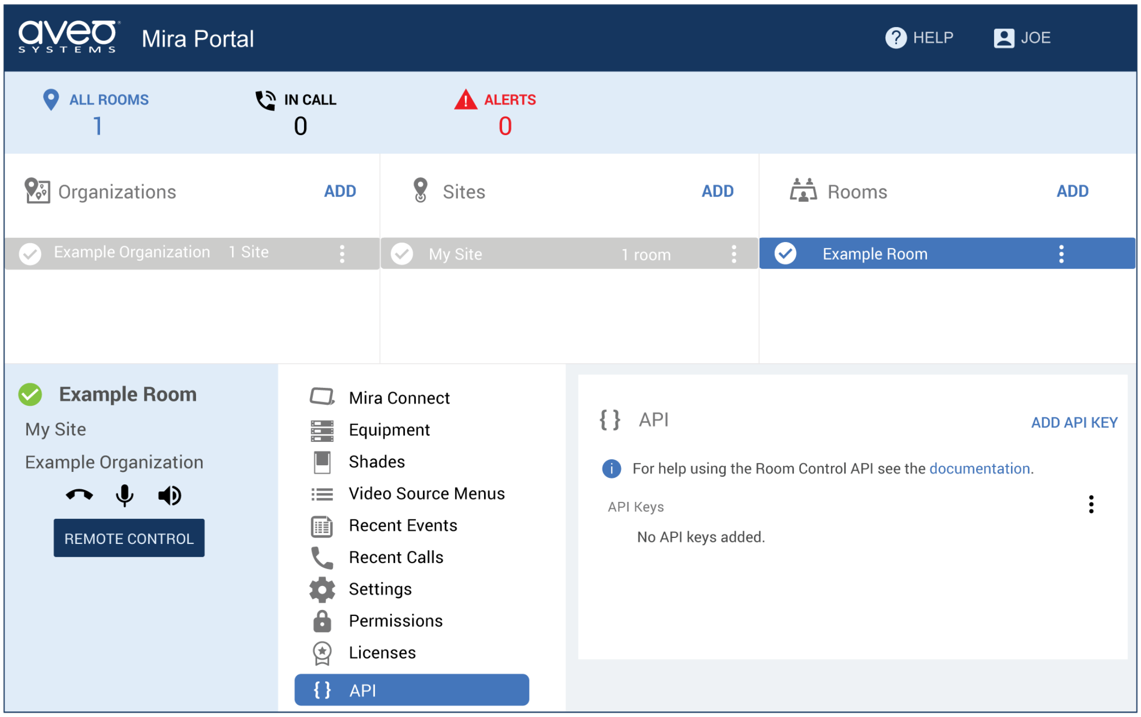
Select source: INPUT 1

Figure 17: Mira Connect's 'Remote Control' feature shows the UI in your browser and allows you to control the room from anywhere.

Once we have completed all the steps to the keypad integration, we'll show you how you can control the system using the 'Remote Control' feature and how you'll also be able to see the affects of button presses made on the keypad.

Step 6 — Create a Mira Connect API Key

Once the room is paired to Mira Connect software, navigate to the API menu and click **ADD API KEY**. If you are going to be using different API keys for different applications, you can give each API key a different, appropriately-named label.



For help using the Room Control API see the documentation.

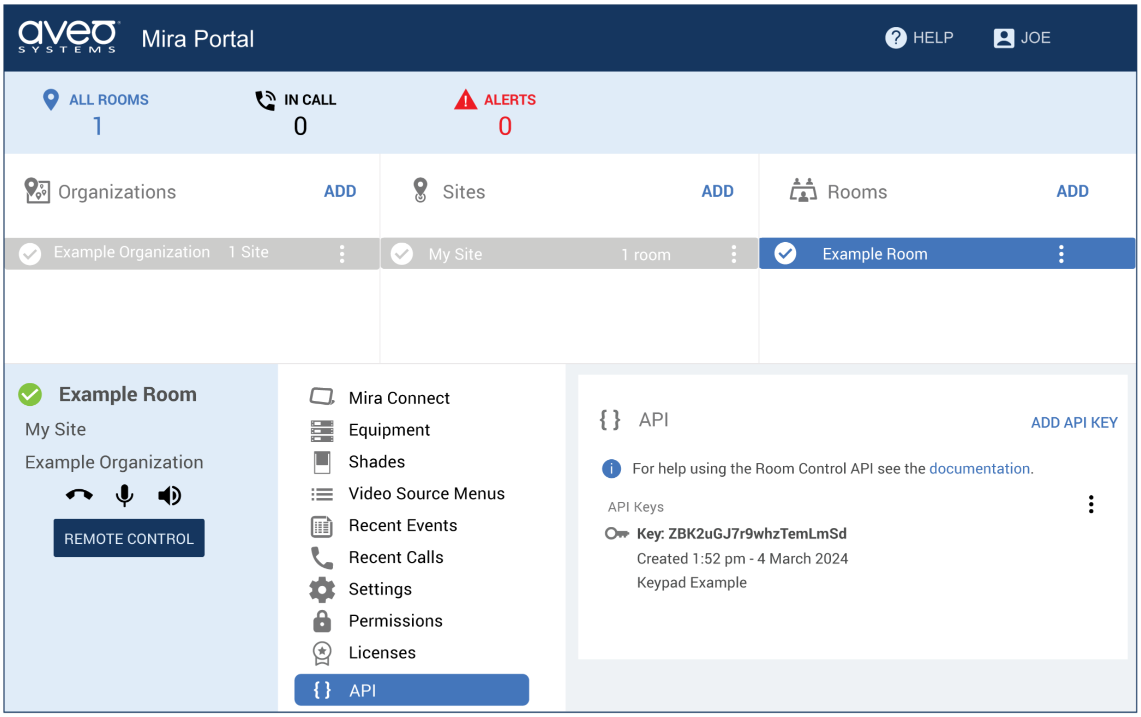
No API keys added.

Figure 18: Select the API menu to create an API key.

The API key value will be used in the commands that we send to the Mira Connect software to control the room. In this example, Mira Portal created an API key of **ZBK2uGJ7r9whzTemLmSd** . Click the 3 dots to the right of the API key to copy the key to your clipboard.

> NOTE: The API key is *case sensitive* and is *unique to your application and room*. If you want to control Mira Connect from different applications, you can create different API keys, one for each application, or you can use the same API key across different applications.

If the API key is removed, all applications using that particular API Key will no longer be able to control Mira Connect as the commands will be refused by Mira Connect. Creating separate API keys for each application is a good practice and makes it easier for you to control access in the future.



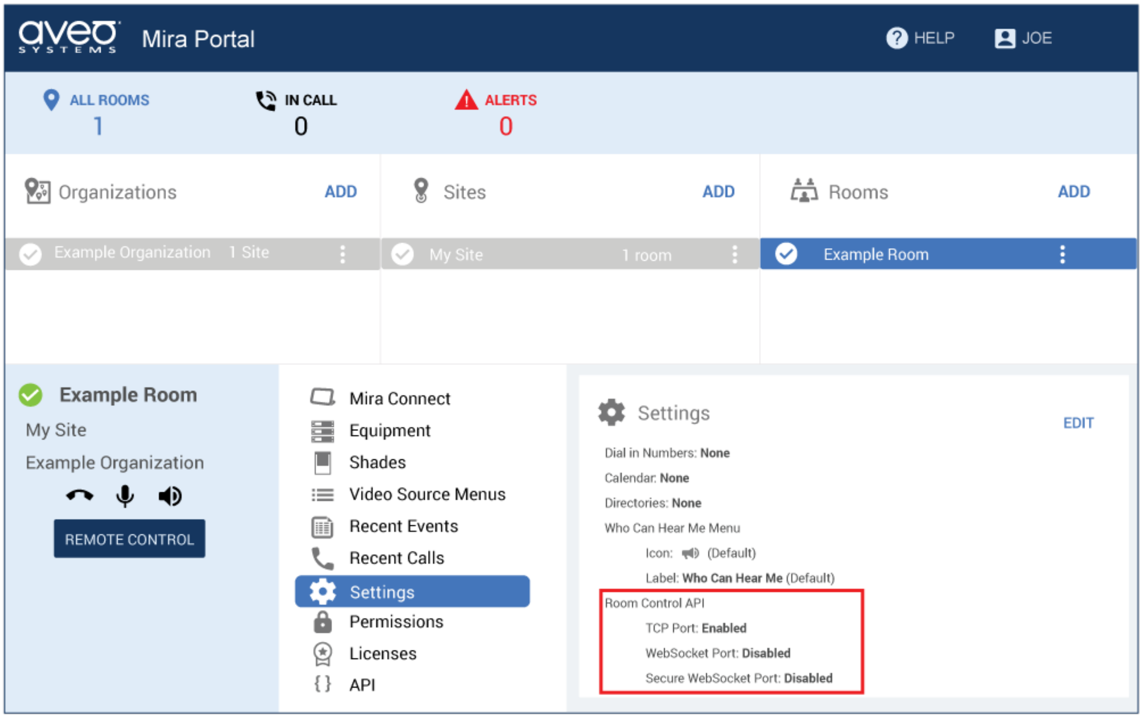
For help using the Room Control API see the documentation.

API Keys

Key: ZBK2uGJ7r9whzTemLmSd
Created 1:52 pm - 4 March 2024
Keypad Example

Figure 19: Once created, copy the API key to your clipboard for use with the room control API commands.

Before using the API key, we must first **ensure that TCP is enabled** for the room in Mira Portal. Enabling TCP allows TCP communications to be created over port 52838 to the Mira Connect software.



Settings

Dial in Numbers: None
Calendar: None
Directories: None
Who Can Hear Me Menu
Icon: (Default)
Label: Who Can Hear Me (Default)
Room Control API
TCP Port: Enabled
WebSocket Port: Disabled
Secure WebSocket Port: Disabled

Figure 20: Enable TCP communication on Mira Connect software so Mira Connect will listen for commands on port 52838.

In addition, **if the PC running Mira Connect software is using a firewall**, follow our step-by-step instructions to create an inbound firewall rule on the PC that will allow incoming connections over port 52838. By default, the PC may be blocking inbound communication to port 52838 from external devices until an inbound firewall rule has been created.

Mira Connect is now ready to be controlled.

Controlling Mira Connect

Once we have an API key, we can start to use the room control API. For more information about the room control API, see the API overview here and the complete command reference guide.

Continuing with our keypad example, we're going to use the following room control commands to control mute, volume, and source selection with Mira Connect:

Command	Description
set roomMute true	Mutes the room
set roomMute false	Unmutes the room
set roomVolumeInc -1	Decrements the room's volume by 1dB
set roomVolumeInc 1	Increments the room's volume by 1dB
set displaySource *displayId value*	Routes the specified source on a display
set roomOff	Powers off the room and resets the audio levels and mute

Command	Description
subscribe roomCaps filters: displays.*.sources	Gets the unique display ID and source IDs for this room to use with the displaySource command

There are many more API commands that can be used for other applications, including: recalling camera presets, triggering screen motion, adjusting lighting and shades, initiating recording, dialing speed dials, and much more. Anything Mira Connect can do can also be done using the Mira Connect room control API.

> NOTE: Mira Connect's room control API command syntax is case sensitive. White spaces are used to separate commands and arguments. More than one whitespace character can be used.
>
> Commands also support keyword arguments which have a key: value format.

Find the Display ID and Source ID Values

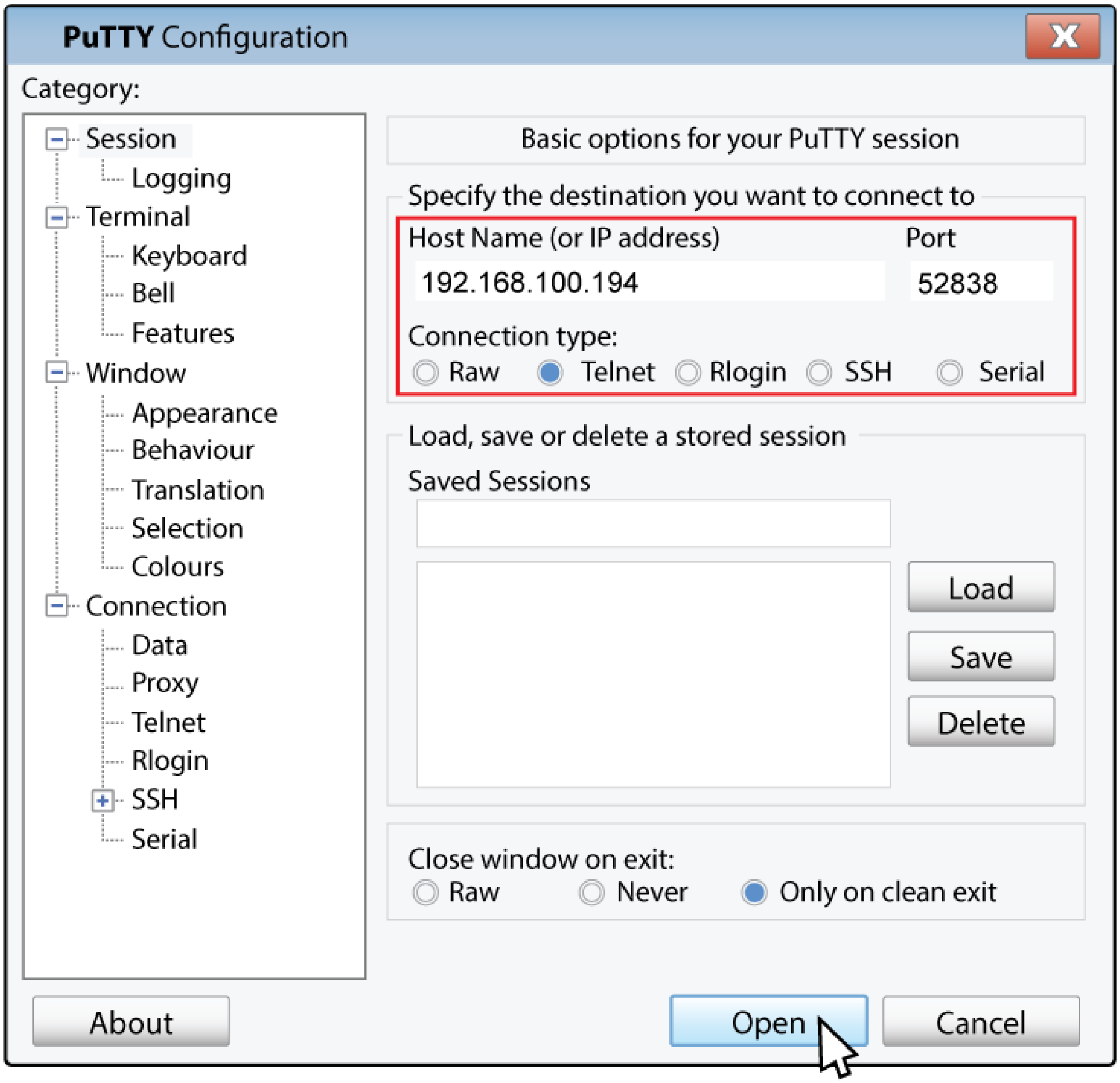
Each display and all sources defined for a display have unique IDs that can be used to reference those devices and sources with API commands.

To get the display ID and source ID values for the room, we'll send the `subscribe roomCaps` command. This command will report back all the capabilities of the room, including the unique display and source IDs. If a display is deleted and a new display is added in Mira Portal, then the display ID for the new display will differ from the original display ID. Similarly, if sources are removed and added back again, the new sources will have different source IDs.

To send the `subscribe roomCaps` command, we'll create a telnet session to the IP address of the computer running the Mira Connect software. If your computer does not support the telnet command, then we can use a telnet program, such as PuTTY, to create a telnet connection to the Mira Connect software at the computer's IP address and port 52838.

In this example, the Mira Connect software is running on a PC that has an IP address of 192.168.100.194.

After downloading and installing PuTTY, run the application and click **Open** to create the connection.



Specify the destination you want to connect to

Host Name (or IP address): 192.168.100.194

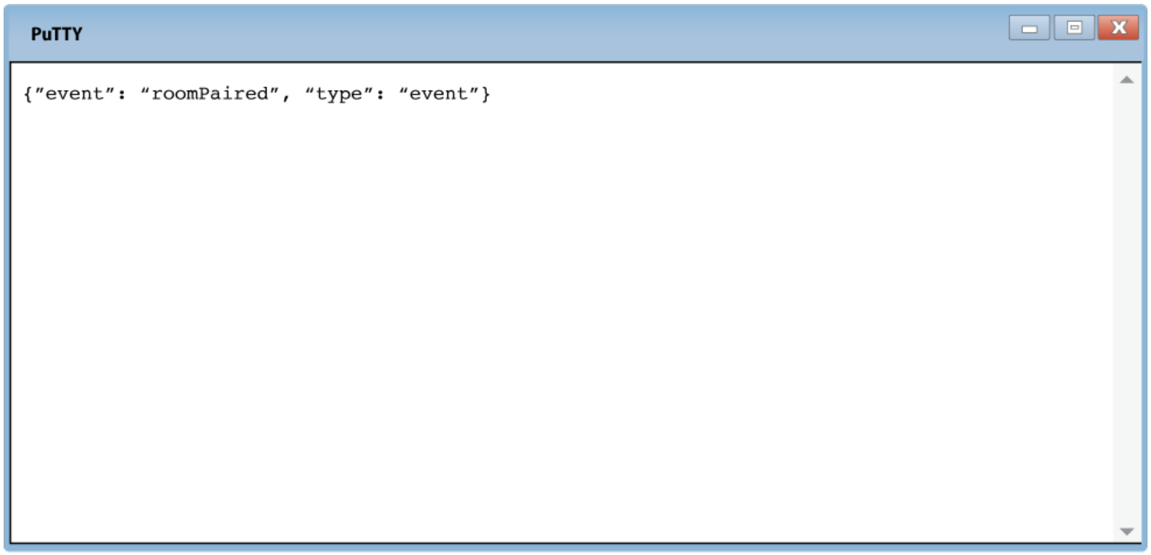
Port: 52838

Connection type: ○ Raw ● Telnet ○ Rlogin ○ SSH ○ Serial

Close window on exit: ○ Raw ○ Never ● Only on clean exit

Figure 21: Create a Telnet session to Mira Connect using port 52838.

Once the connection is established, Mira Connect will automatically report back that the room is paired.



```
{"event": "roomPaired", "type": "event"}
```

Figure 22: Use a terminal program like PuTTY to create a socket to Mira Connect software.

If the connection is refused, confirm the IP address and port number and ensure TCP is enabled in the room settings in Mira Portal as described previously. You may also need to create an inbound firewall rule following our step-by-step instructions.

Once the room is paired to Mira Connect, we'll send the `subscribe roomCaps` command to get information about the capabilities of the room. The subscribe command requires a filters argument that specifies which information we want to receive from the subscription. In this case, we'll filter the information to just the display source information for all displays in the room using the filter **displays.*.sources**. To get all the information, use the filter * . More information about subscriptions and filters is available in the Mira Connect Room Control API documentation.

While the room control API supports authenticating using the API key once per socket connection or with each command. In our application we'll send the API key with each command since it doesn't make sense to authenticate because each command sent by the KeyDigital keypad will create a new socket connection. To send the API key with each command we'll add the apiKey: keyword to each command.

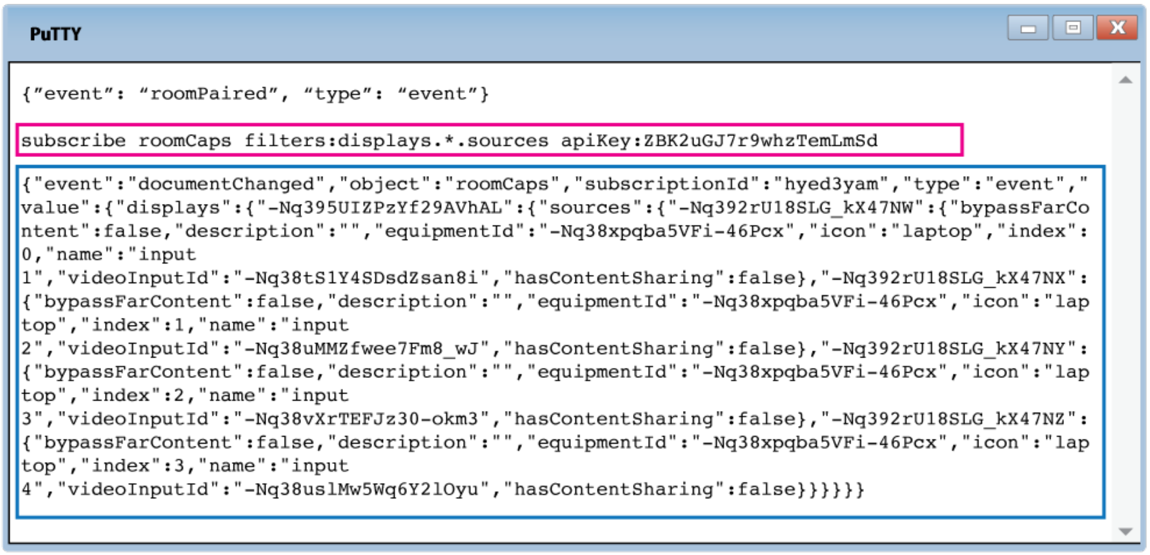
> NOTE: Since a new TCP socket is created for each command from the KeyDigital keypad, the apiKey: keyword **must be** included with each command. Remember to use **your specific API key** in the command — not the value used in this example.

The command we'll send to find the room capabilities in this example is:

```
subscribe roomCaps filters:displays.*.sources apiKey:ZBK2uGJ7r9whzTemLmSd
```

Enter this command string and **your specific API key** into the terminal window and press the Return/Enter key.

Sending the subscribe command (shown below with the pink outline) will cause a response (shown below in the blue outline) to be returned by Mira Connect software with the current room capabilities.



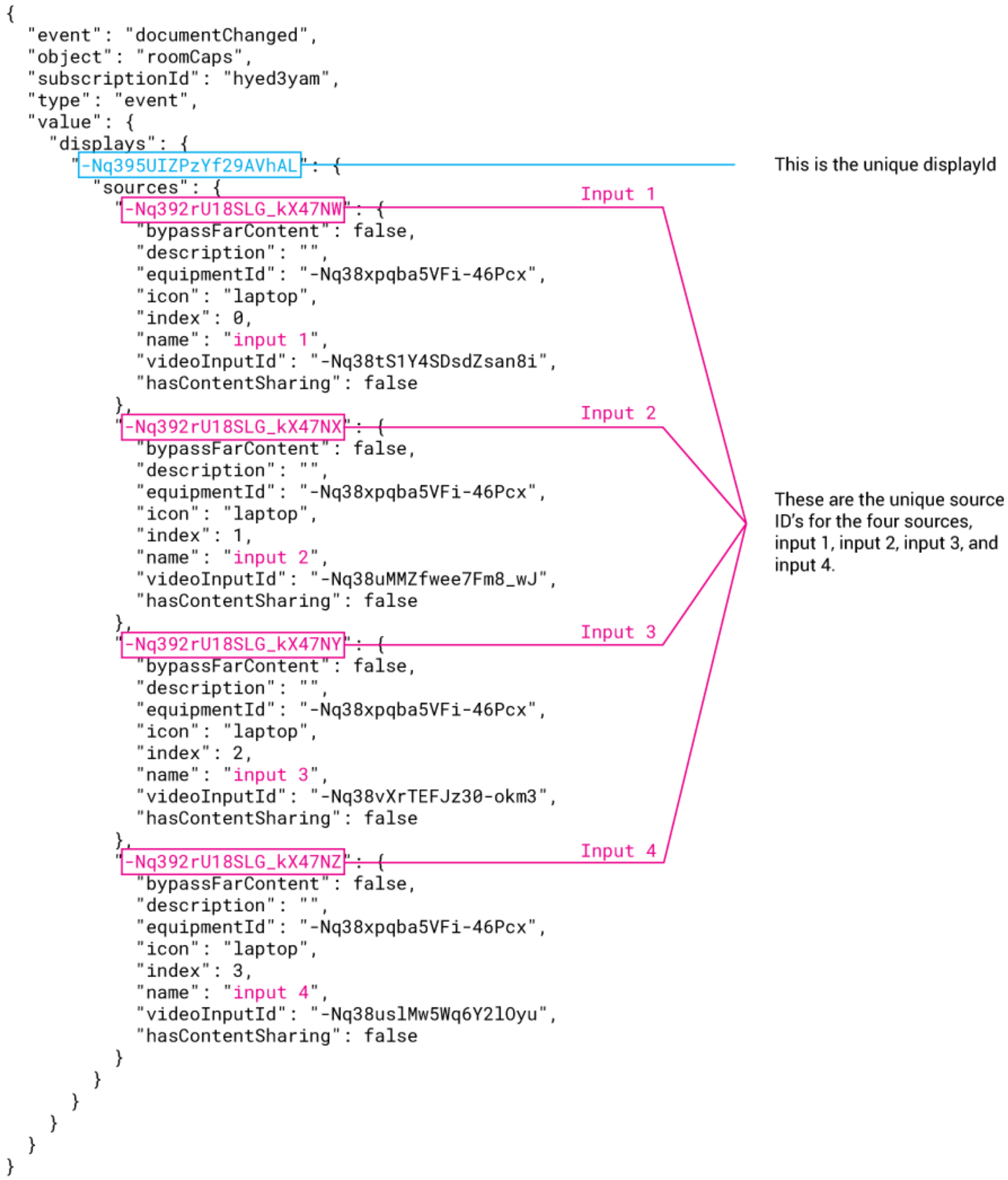
PuTTY

```
{"event": "roomPaired", "type": "event"}

subscribe roomCaps filters:displays.*.sources apiKey:ZBK2uGJ7r9whzTemLmSd

{"event":"documentChanged","object":"roomCaps","subscriptionId":"hyed3yam","type":"event","
value":{"displays":{"-Nq395UIZPzYf29AVhAL":{"sources":{"-Nq392rU18SLG_kX47NW":{"bypassFarCo
ntent":false,"description":"","equipmentId":"-Nq38xpqba5VFi-46Pcx","icon":"laptop","index":
0,"name":"input
1","videoInputId":"-Nq38tS1Y4SDsdZsan8i","hasContentSharing":false},"-Nq392rU18SLG_kX47NX":
{"bypassFarContent":false,"description":"","equipmentId":"-Nq38xpqba5VFi-46Pcx","icon":"lap
top","index":1,"name":"input
2","videoInputId":"-Nq38uMMZfwee7Fm8_wJ","hasContentSharing":false},"-Nq392rU18SLG_kX47NY":
{"bypassFarContent":false,"description":"","equipmentId":"-Nq38xpqba5VFi-46Pcx","icon":"lap
top","index":2,"name":"input
3","videoInputId":"-Nq38vXrTEFJz30-okm3","hasContentSharing":false},"-Nq392rU18SLG_kX47NZ":
{"bypassFarContent":false,"description":"","equipmentId":"-Nq38xpqba5VFi-46Pcx","icon":"lap
top","index":3,"name":"input
4","videoInputId":"-Nq38uslMw5Wq6Y2lOyu","hasContentSharing":false}}}}}}
```

Figure 23: Use a terminal program like PuTTY to send Mira Connect room control API commands to the Mira Connect software.

The received roomCaps room capabilities information includes the source information for all displays in the room. If sources are deleted or added, or displays are added or removed in Mira Portal, there will be additional responses that are automatically generated that include the latest information about the room's capabilities.

We can copy the response and format the JSON response with a JSON formatter to make the response more readable, and we can also annotate the response as shown below.



```
{
  "event": "documentChanged",
  "object": "roomCaps",
  "subscriptionId": "hyed3yam",
  "type": "event",
  "value": {
    "displays": {
      "-Nq395UIZPzYf29AVhAL": {
        "sources": {
          "-Nq392rU18SLG_kX47NW": {
            "bypassFarContent": false,
            "description": "",
            "equipmentId": "-Nq38xpqba5VFi-46Pcx",
            "icon": "laptop",
            "index": 0,
            "name": "input 1",
            "videoInputId": "-Nq38tS1Y4SDsdZsan8i",
            "hasContentSharing": false
          },
          "-Nq392rU18SLG_kX47NX": {
            "bypassFarContent": false,
            "description": "",
            "equipmentId": "-Nq38xpqba5VFi-46Pcx",
            "icon": "laptop",
            "index": 1,
            "name": "input 2",
            "videoInputId": "-Nq38uMMZfwee7Fm8_wJ",
            "hasContentSharing": false
          },
          "-Nq392rU18SLG_kX47NY": {
            "bypassFarContent": false,
            "description": "",
            "equipmentId": "-Nq38xpqba5VFi-46Pcx",
            "icon": "laptop",
            "index": 2,
            "name": "input 3",
            "videoInputId": "-Nq38vXrTEFJz30-okm3",
            "hasContentSharing": false
          },
          "-Nq392rU18SLG_kX47NZ": {
            "bypassFarContent": false,
            "description": "",
            "equipmentId": "-Nq38xpqba5VFi-46Pcx",
            "icon": "laptop",
            "index": 3,
            "name": "input 4",
            "videoInputId": "-Nq38uslMw5Wq6Y2lOyu",
            "hasContentSharing": false
          }
        }
      }
    }
  }
}
```

This is the unique displayId
Input 1
Input 2
Input 3
Input 4
These are the unique source ID's for the four sources, input 1, input 2, input 3, and input 4.

Figure 24: The room capabilities as returned by the 'subscribe roomCaps' command.

As highlighted in the annotated response above,

- The unique display ID for the display in the room is: **-Nq395UIZPzYf29AVhAL**
- The unique source IDs that will be used for selecting those sources are:
  - Input 1: **-Nq392rU18SLG_kX47NW**

- Input 2: **-Nq392rU18SLG_kX47NX**
- Input 3: **-Nq392rU18SLG_kX47NY**
- Input 4: **-Nq392rU18SLG_kX47NZ**

We'll use the display ID and source IDs in the commands as described in the following sections.

> NOTE: The display IDs and source IDs are *case sensitive* and start with the '-' character. If displays or sources are added or removed, new IDs will be automatically reported via the roomCaps subscription. If you are not tracking changes to the roomCaps, then you can also send the `subscribe roomCaps` command again to get the latest values.

Now that we have the display ID and source IDs, we are ready to configure the KeyDigital keypad.

Configuring the KeyDigital Keypad

Creating a configuration for the KeyDigital keypad requires two main steps:

1. Defining the buttons, labels, and the types of button actions
2. Creating the events associated with the button actions

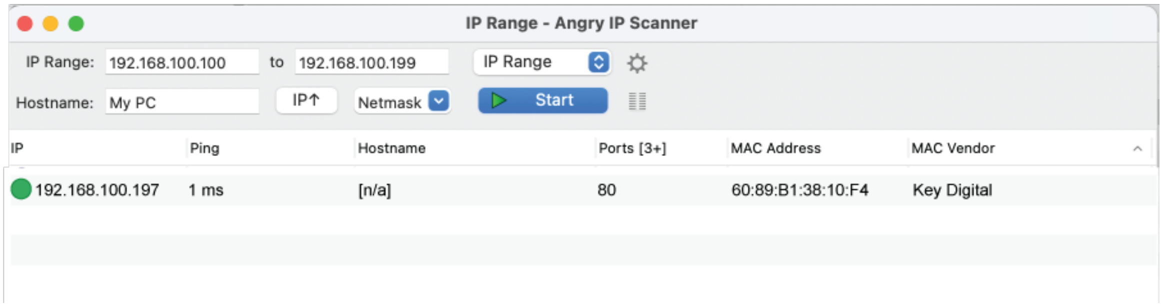
KeyDigital keypads support TCP socket connections from the keypad, allowing the keypad to send commands to Mira Connect. The keypad does not process any feedback from Mira Connect.

Step 7 — Define the Buttons on the KeyDigital Keypad

Connect the KeyDigital keypad to a PoE ethernet connection to power up the keypad. Open your browser and enter the keypad's IP address.

> NOTE: The keypad has a default IP address of: **192.168.1.239** with default password of **12345678**. For more information about configuring the keypad, refer to KeyDigital's Setup Manual.

If you have configured the IP address for the keypad already, but don't remember the IP address, use a tool like Angry IP Scanner to discover the IP addresses of equipment on the local network. In this example, we used Angry IP Scanner to discover the keypad at 192.168.100.197 on our local network.



IP Range - Angry IP Scanner

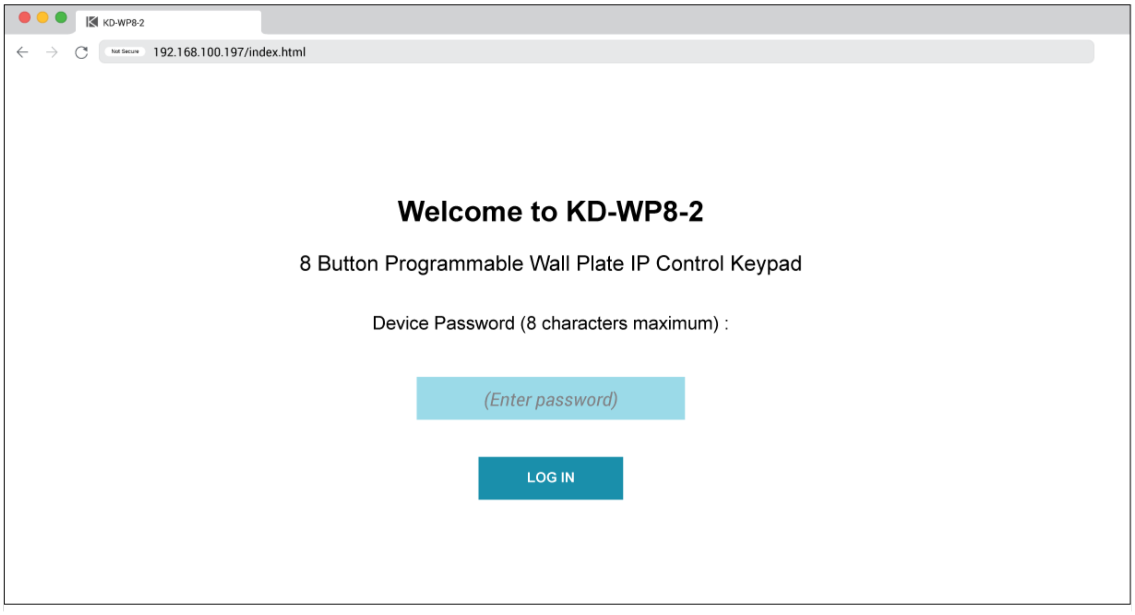
IP Range: 192.168.100.100 to 192.168.100.199 IP Range

Hostname: My PC

IP	Ping	Hostname	Ports [3+]	MAC Address	MAC Vendor
192.168.100.197	1 ms	[n/a]	80	60:89:B1:38:10:F4	Key Digital

Figure 25: Angry IP scanner can be used to find the IP address of the keypad.

Once you have found the IP address of the keypad, browse to the keypad's IP address (192.168.100.197 in our example), enter the default password of **12345678**, and click **LOG IN**.



KD-WP8-2

Not Secure 192.168.100.197/index.html

Welcome to KD-WP8-2

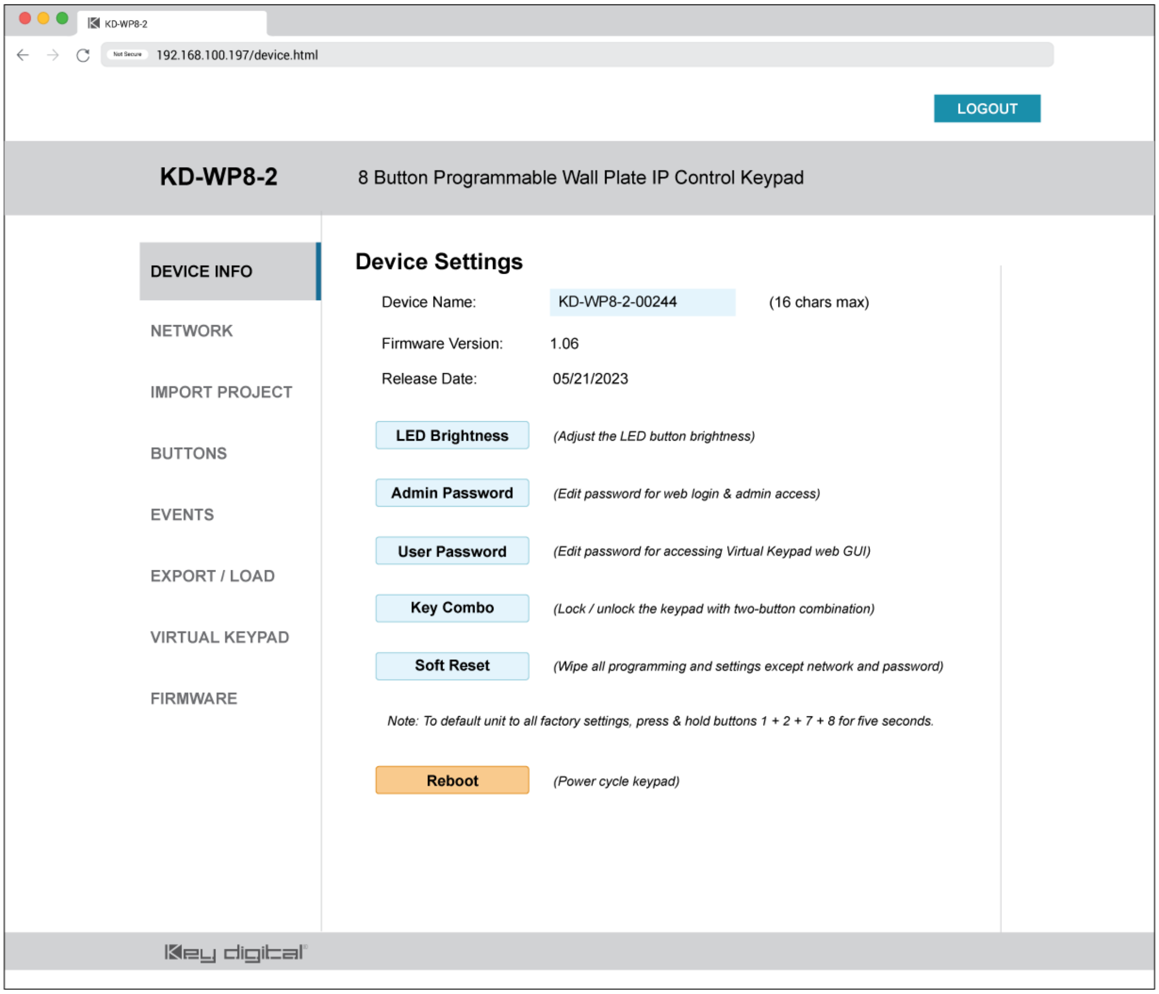
8 Button Programmable Wall Plate IP Control Keypad

Device Password (8 characters maximum) :

(Enter password)

Figure 26: Browse into the keypad to start the configuration.

Once logged in, the device info page will be shown. Here you can configure additional features of the keypad, including LED brightness and passwords.



KD-WP8-2 8 Button Programmable Wall Plate IP Control Keypad

Device Settings

Device Name: KD-WP8-2-00244 (16 chars max)

Firmware Version: 1.06

Release Date: 05/21/2023

LED Brightness *(Adjust the LED button brightness)*

Admin Password *(Edit password for web login & admin access)*

User Password *(Edit password for accessing Virtual Keypad web GUI)*

Key Combo *(Lock / unlock the keypad with two-button combination)*

Soft Reset *(Wipe all programming and settings except network and password)*

Note: To default unit to all factory settings, press & hold buttons 1 + 2 + 7 + 8 for five seconds.

Reboot *(Power cycle keypad)*

Figure 27: Access the KeyDigital keypad configuration menu.

For this example, we'll continue to the **BUTTONS** menu to start configuring the buttons for our application.

Button 1: Volume Down

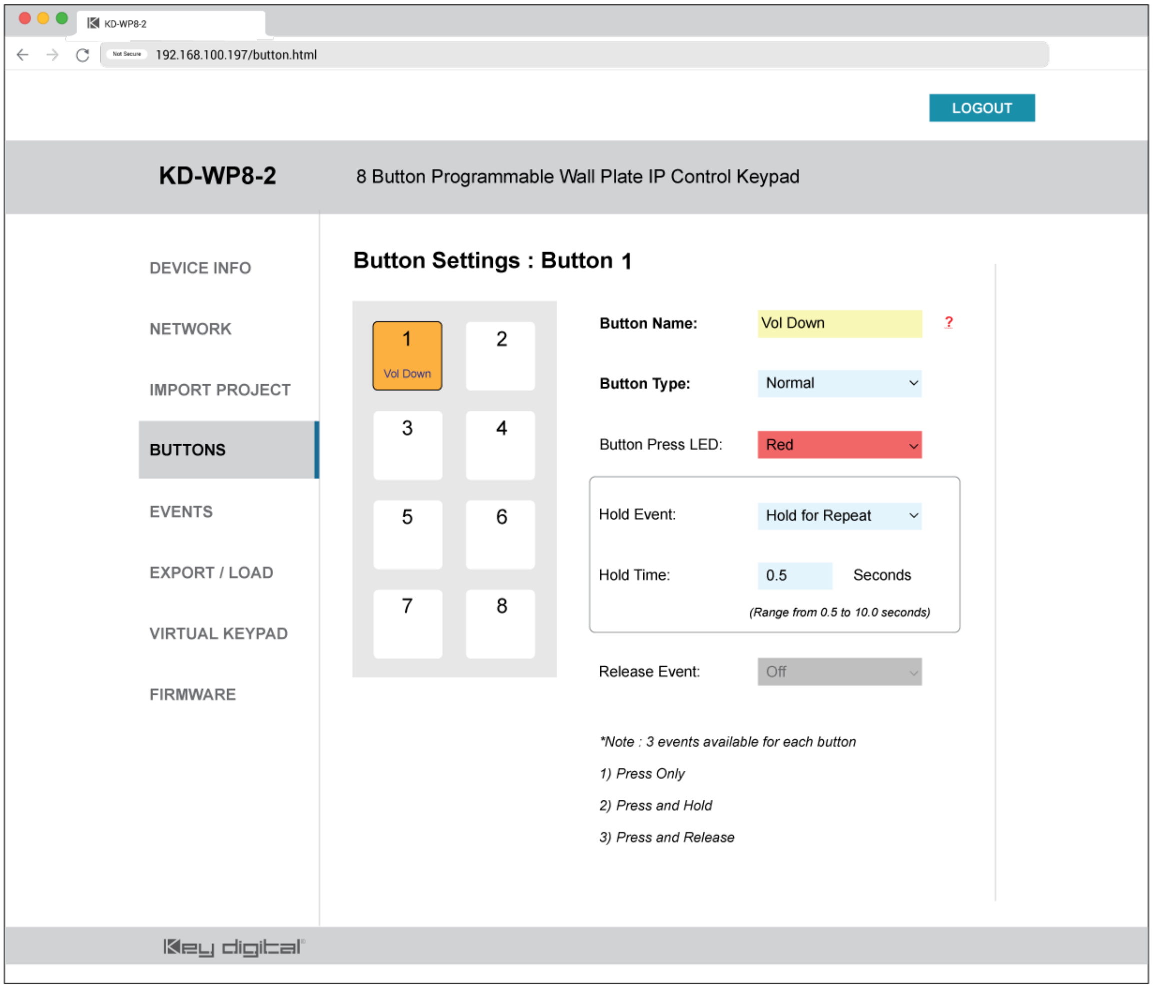
We'll start with Button 1 and configure that as *Volume Down*.

We'll add a 'Hold for Repeat' function for repeating the command after being held for 0.5 seconds. This will make it easy to adjust the volume down by either pressing and holding the button or by repeatedly pressing the button.

While this button is pressed, our design will have it turn on the Red LED.

As shown in the following figure, we'll configure the button as follows:

- Set the button name to **Vol Down**,
- Select button type as **Normal**,
- Configure the Button Press LED to be **Red**, and
- Configure the hold for repeat with a **0.5** second hold time



KD-WP8-2 8 Button Programmable Wall Plate IP Control Keypad

Button Settings : Button 1

Button Name: Vol Down

Button Type: Normal

Button Press LED: Red

Hold Event: Hold for Repeat

Hold Time: 0.5 Seconds

(Range from 0.5 to 10.0 seconds)

Release Event: Off

*Note : 3 events available for each button

1) Press Only

2) Press and Hold

3) Press and Release

Figure 28: Configure Button 1 of the keypad.

Button 2: Volume Up

Button 2 will be configured as *Volume Up*.

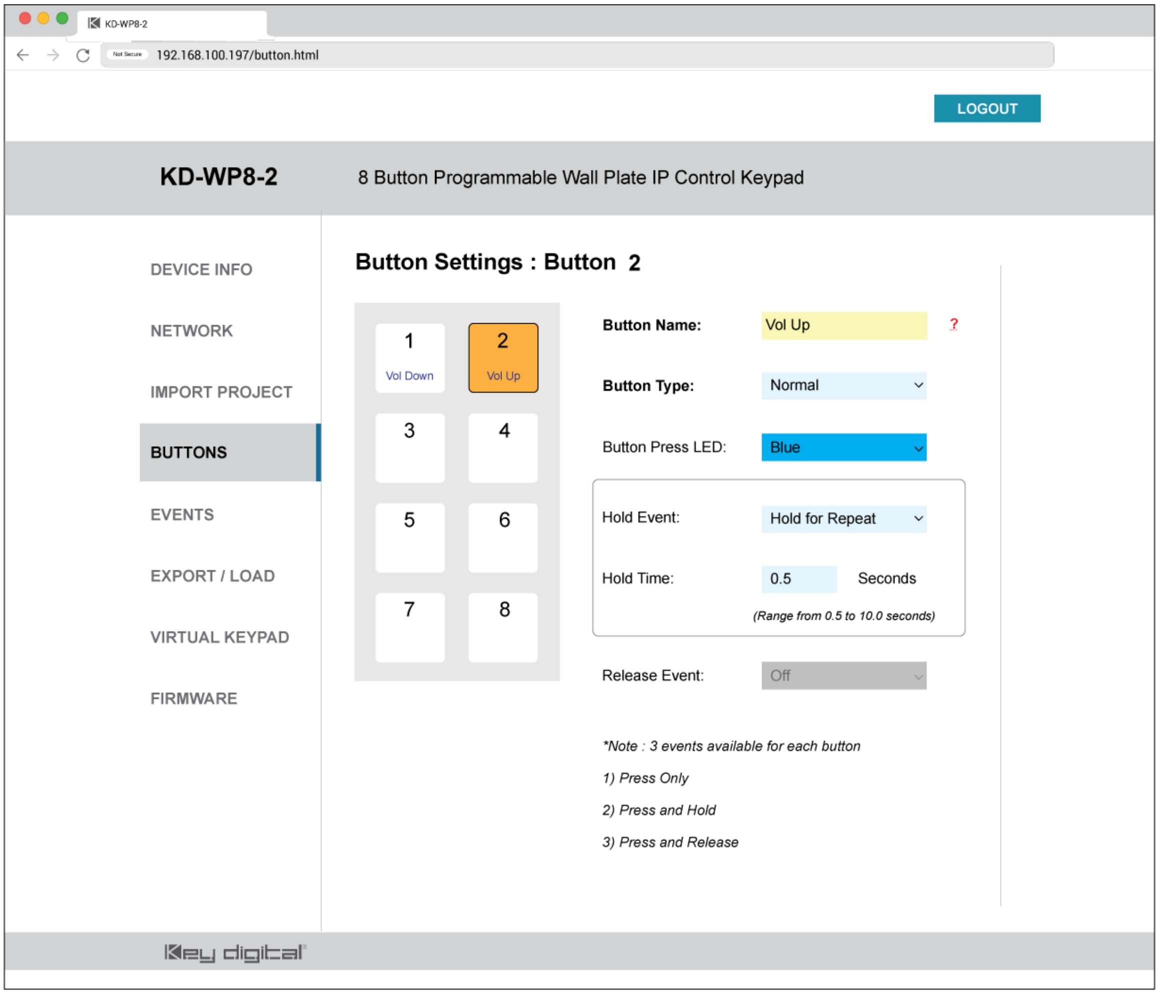
Similar to the Volume Down button, we'll add a 'Hold for Repeat' function for repeating the command after being held for 0.5 seconds.

While this button is pressed, it will light up blue.

As shown in the following figure, we'll configure the button as follows:

- Set the button name to **Vol Up**,
- Select button type as **Normal**,
- Configure the Button Press LED to be **Blue**, and
- Configure the hold for repeat with a **0.5** second hold time

.



KD-WP8-2 — 8 Button Programmable Wall Plate IP Control Keypad

Button Settings : Button 2

Button Name: Vol Up

Button Type: Normal

Button Press LED: Blue

Hold Event: Hold for Repeat

Hold Time: 0.5 Seconds

(Range from 0.5 to 10.0 seconds)

Release Event: Off

**Note : 3 events available for each button*

1) Press Only

2) Press and Hold

3) Press and Release

Figure 29: Configure Button 2 of the keypad.

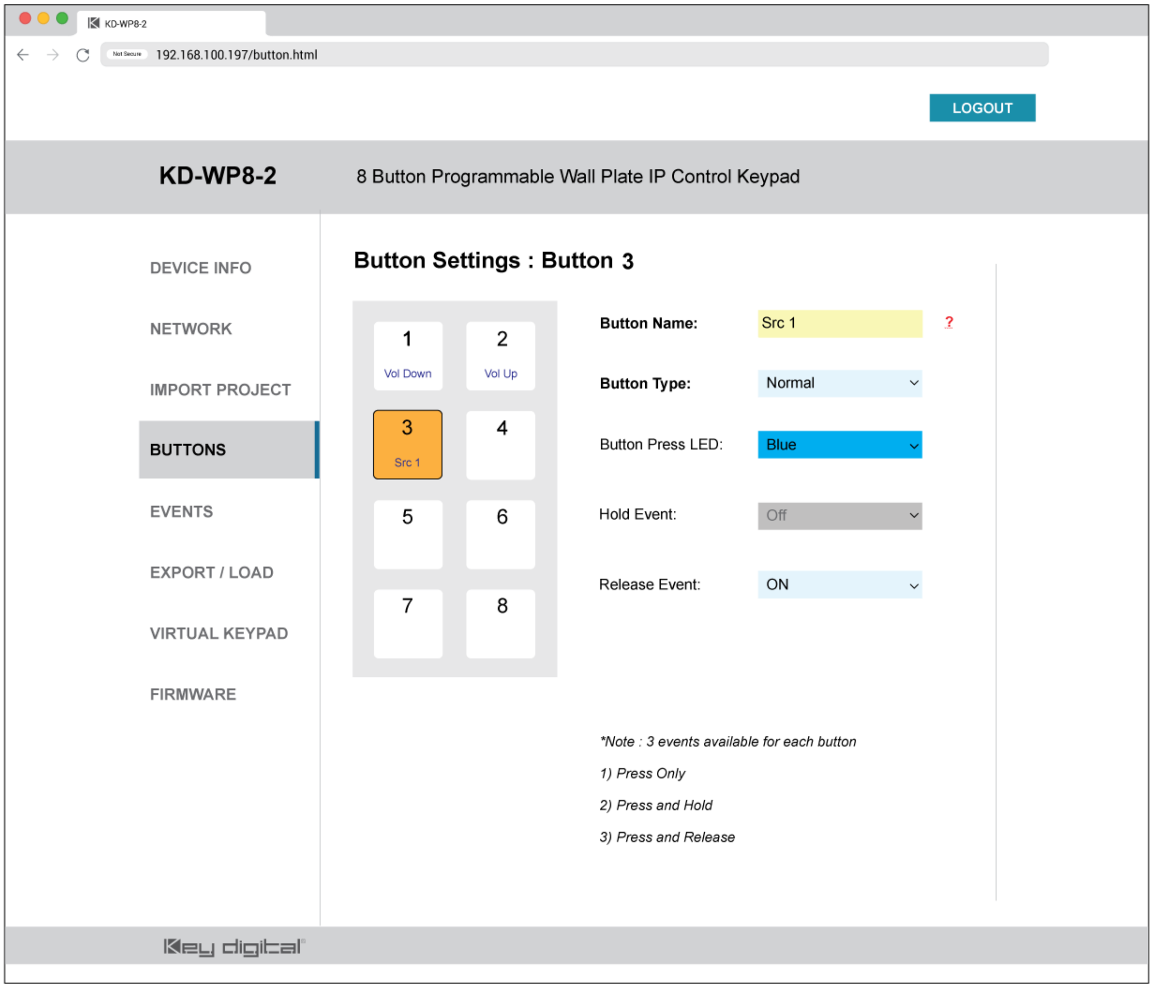
Button 3: Video Input 1

Button 3 will be configured to route Input 1 from the video switcher to the display. While there is no hold event for this button, we'll add support for a release event that will only leave one of the source selection buttons illuminated after the source has been selected. We'll configure these events in the event section.

While button 3 is pressed, it will light up blue. After releasing the button, the button will stay illuminated to indicate this source is selected.

As shown in the following figure, we'll configure the button as follows:

- Set the button name to **Src 1**,
- Select button type as **Normal**,
- Configure the Button Press LED to be **Blue**, and
- Set the release event to **ON**.



KD-WP8-2 8 Button Programmable Wall Plate IP Control Keypad

Button Settings : Button 3

Button Name: Src 1

Button Type: Normal

Button Press LED: Blue

Hold Event: Off

Release Event: ON

*Note : 3 events available for each button

1) Press Only

2) Press and Hold

3) Press and Release

Figure 30: Configure Button 3 of the keypad.

Buttons 4, 5, and 6 will be configured similarly as described below.

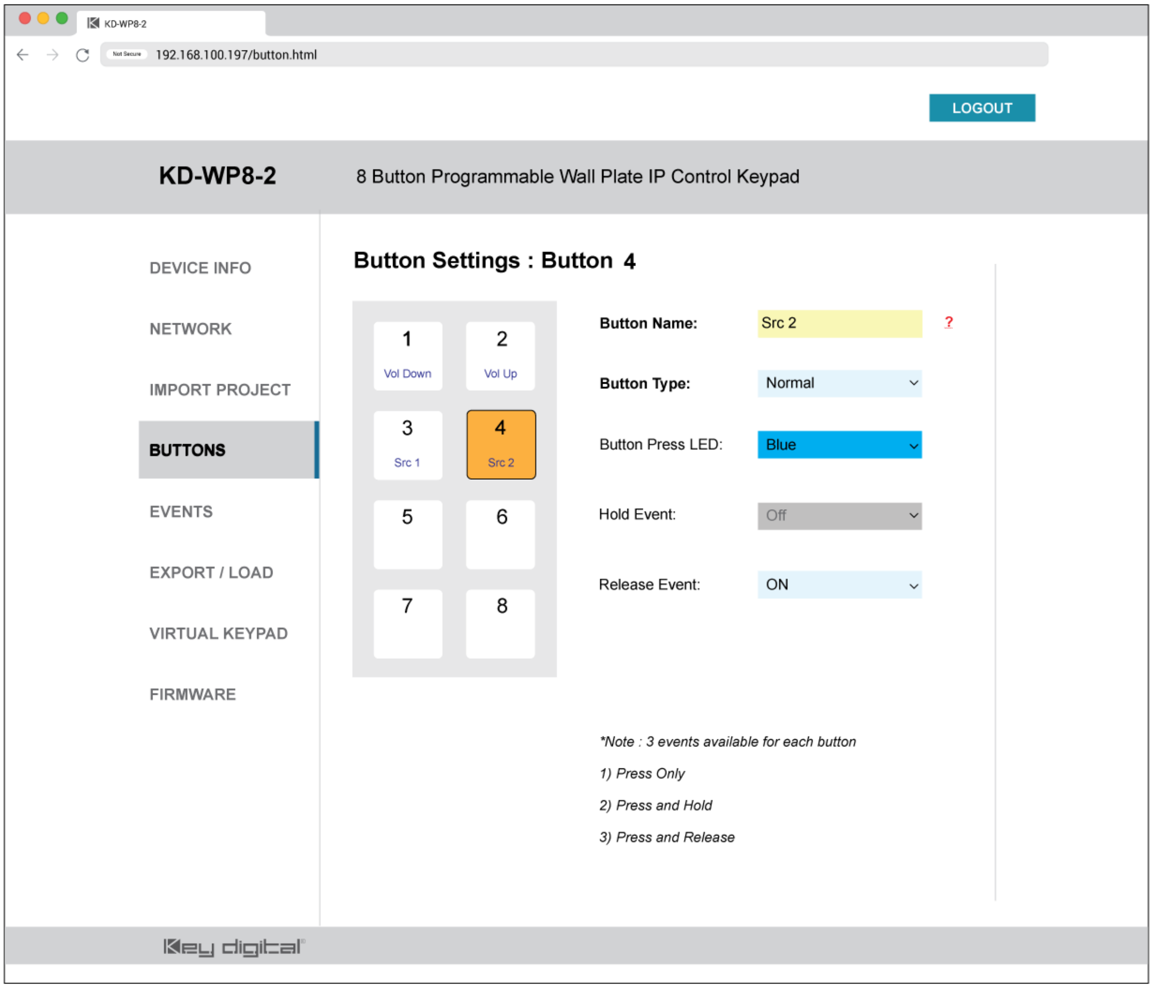
Button 4: Video Input 2

Button 4 will be configured to route Input 2 on the video switcher to the display. While there is no hold event for this button, we'll add support for a release event that will only leave one of the source selection buttons illuminated after the source has been selected. We'll configure the events in the event section.

When this button is pressed, it will light up blue.

As shown in the following figure, we'll configure the button as follows:

- Set the button name to **Src 2**,
- Select button type as **Normal**,
- Configure the Button Press LED to be **Blue**, and
- Set the release event to **ON**



KD-WP8-2 8 Button Programmable Wall Plate IP Control Keypad

Button Settings : Button 4

Button Name: Src 2

Button Type: Normal

Button Press LED: Blue

Hold Event: Off

Release Event: ON

*Note : 3 events available for each button

1) Press Only

2) Press and Hold

3) Press and Release

Figure 31: Configure Button 4 of the keypad.

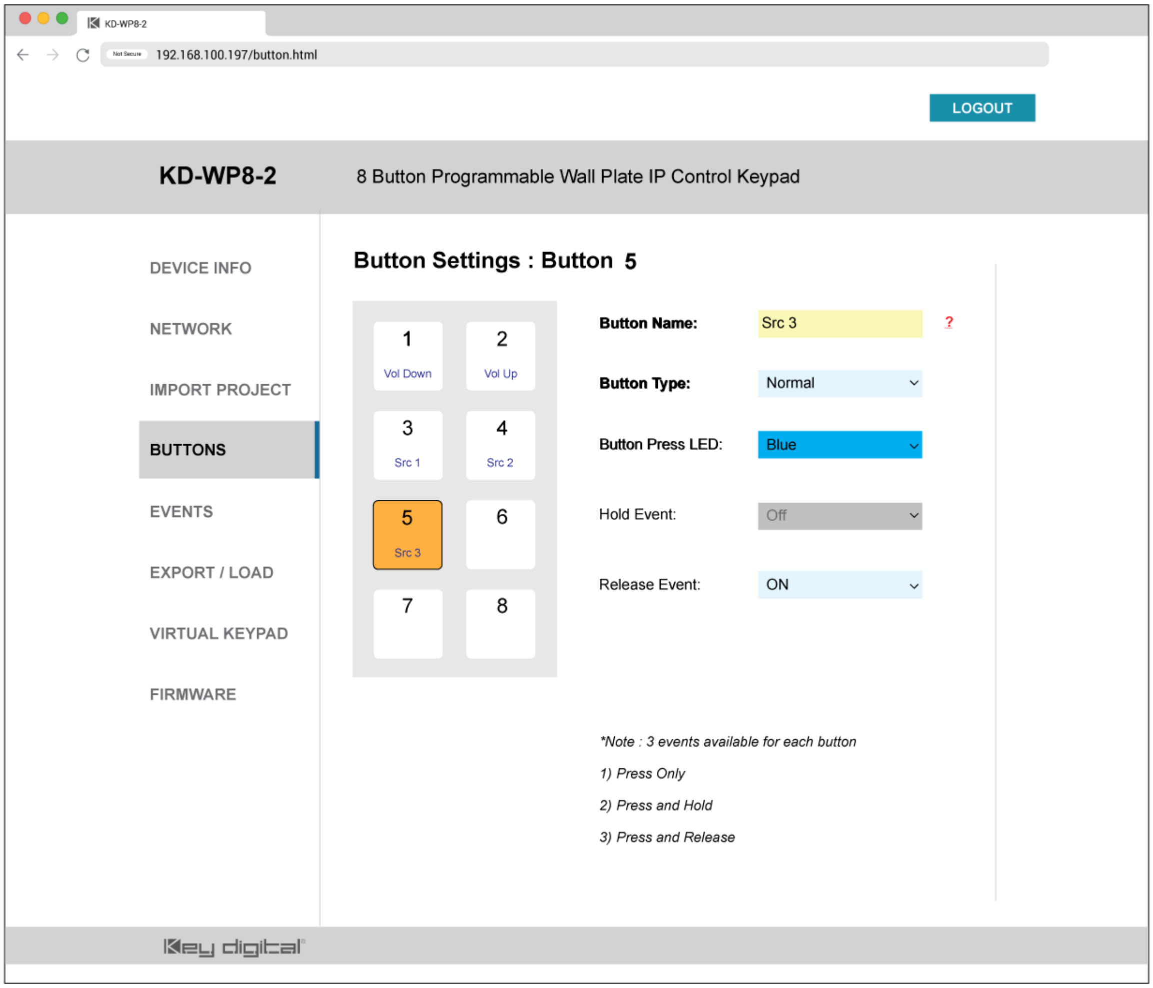
Button 5: Video Input 3

Button 5 will be configured to route Input 3 on the video switcher to the display. While there is no hold event for this button, we'll add support for a release event that will only leave one of the source selection buttons illuminated after the source has been selected. We'll configure the events in the event section.

When this button is pressed, it will light up blue.

As shown in the following figure, we'll configure the button as follows:

- Set the button name to **Src 3**,
- Select button type as **Normal**,
- Configure the Button Press LED to be **Blue**, and
- Set the release event to **ON**



KD-WP8-2 8 Button Programmable Wall Plate IP Control Keypad

Button Settings : Button 5

Button Name: Src 3

Button Type: Normal

Button Press LED: Blue

Hold Event: Off

Release Event: ON

*Note : 3 events available for each button

1) Press Only

2) Press and Hold

3) Press and Release

Figure 32: Configure Button 5 of the keypad.

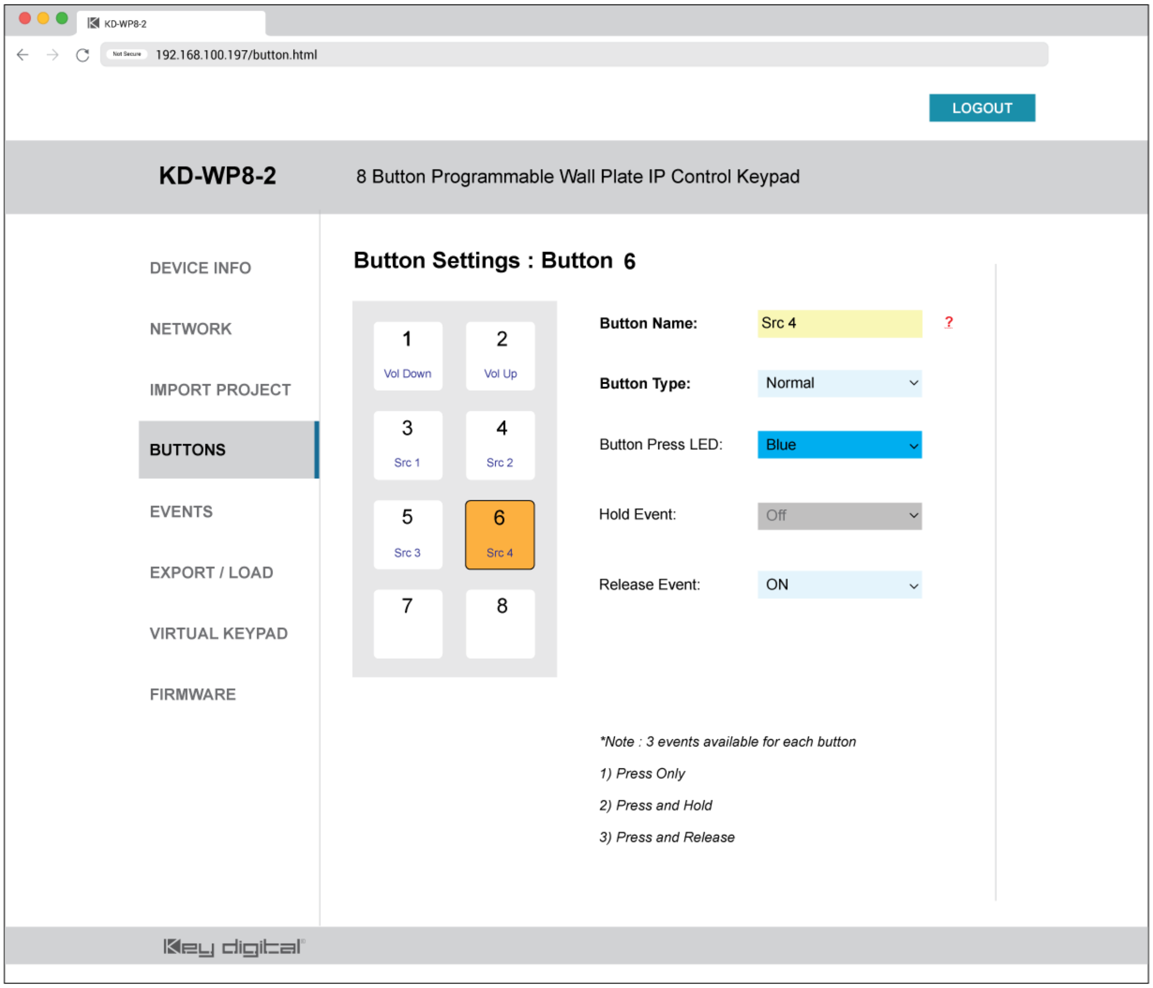
Button 6: Video Input 4

Button 6 will be configured to route Input 4 on the video switcher to the display. While there is no hold event for this button, we'll add support for a release event that will only leave one of the source selection buttons illuminated after the source has been selected. We'll configure the events in the event section.

When this button is pressed, it will light up blue.

As shown in the following figure, we'll configure the button as follows:

- Set the button name to **Src 4**,
- Select button type as **Normal**,
- Configure the Button Press LED to be **Blue**, and
- Set the release event to **ON**



KD-WP8-2 8 Button Programmable Wall Plate IP Control Keypad

Button Settings : Button 6

Button Name: Src 4

Button Type: Normal

Button Press LED: Blue

Hold Event: Off

Release Event: ON

*Note : 3 events available for each button

1) Press Only

2) Press and Hold

3) Press and Release

Figure 33: Configure Button 6 of the keypad.

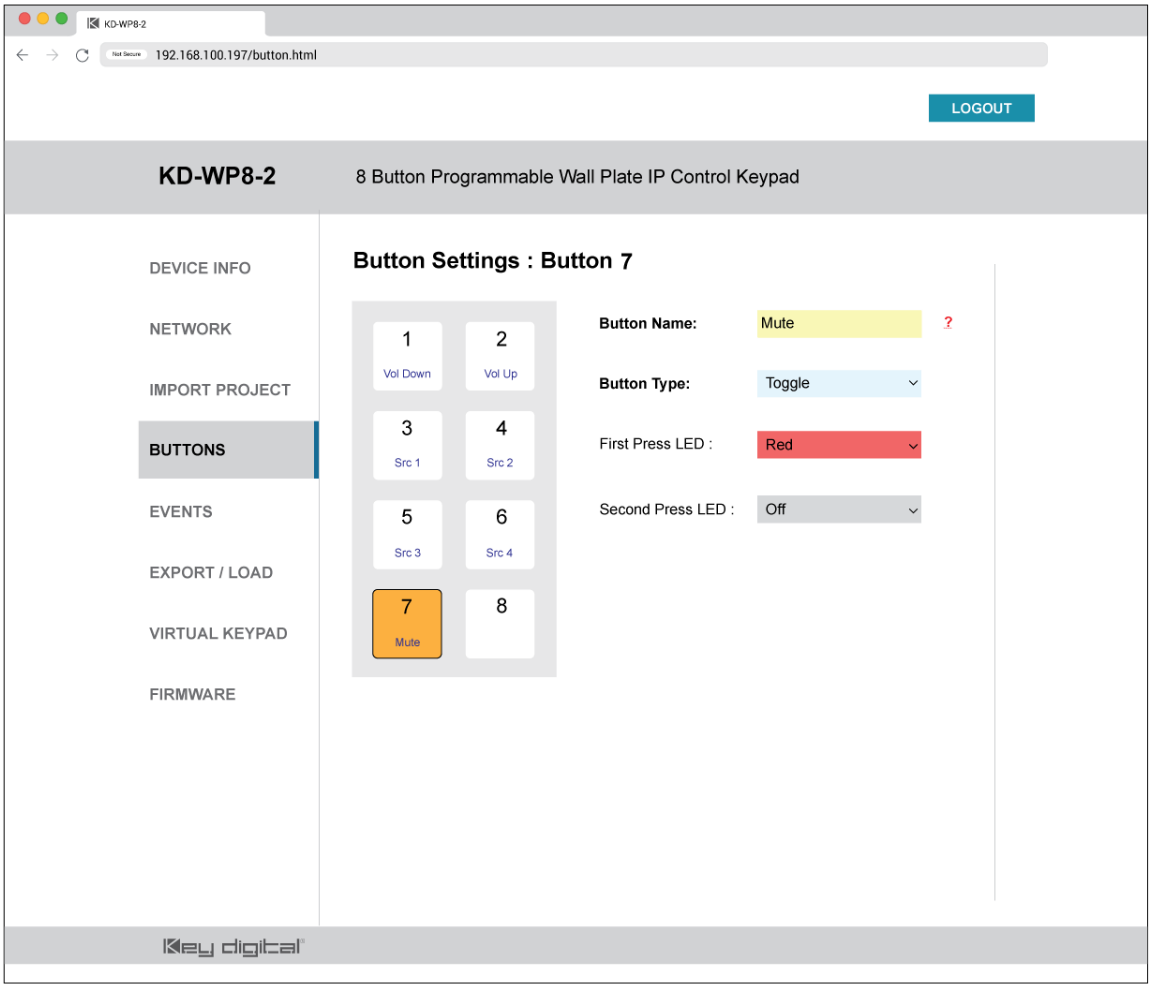
Button 7: Mute

Button 7 will be configured to mute the room's microphones on the first press and to unmute the microphones on the second press.

As shown in the following figure, we'll configure the button as follows:

- Set the button name to **Mute**,
- Select button type as **Toggle**,
- Configure the first button press LED to be **Red** to indicate the system is muted.
- Configure the second button press LED to be **Off** to indicate the system is unmuted.

We'll configure the Toggle functions in the events section.



Button Settings : Button 7

Button Name: Mute

Button Type: Toggle

First Press LED : Red

Second Press LED : Off

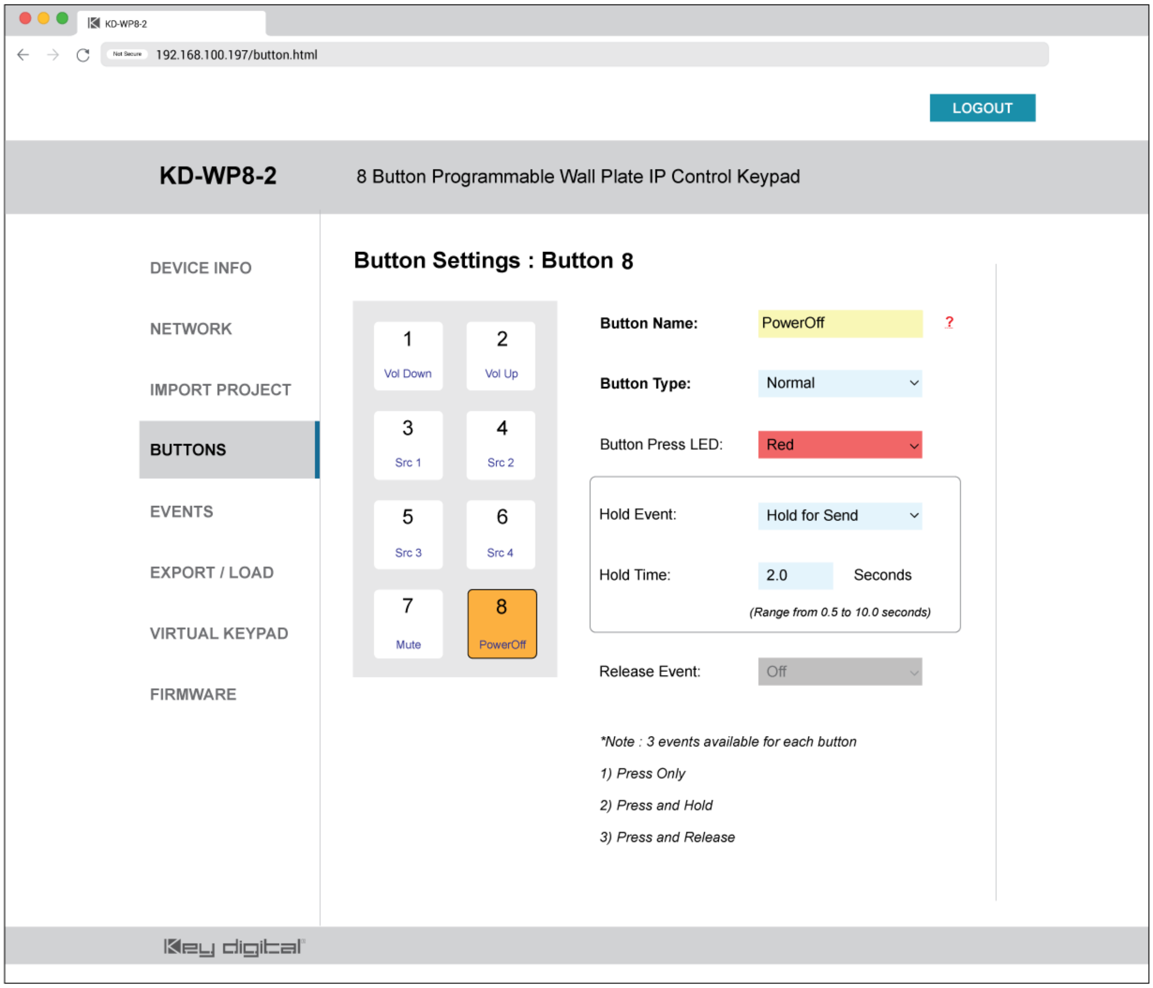
Figure 34: Configure Button 7 of the keypad.

Button 8: Power Off

Button 8 will be configured to power off the room, but only after the button has been held for 2 seconds.

As shown in the following figure, we'll configure the button as follows:

- Set the button name to **PowerOff**,
- Select button type as **Normal**,
- Configure the button press LED to be **Red**.
- Configure a hold event as **Hold for Send**
- Set the hold time to **2** seconds for our example.



KD-WP8-2 8 Button Programmable Wall Plate IP Control Keypad

Button Settings : Button 8

Button Name: PowerOff

Button Type: Normal

Button Press LED: Red

Hold Event: Hold for Send

Hold Time: 2.0 Seconds

(Range from 0.5 to 10.0 seconds)

Release Event: Off

**Note : 3 events available for each button*

1) Press Only

2) Press and Hold

3) Press and Release

Figure 35: Configure Button 8 of the keypad.

Step 8 — Configuring Button Events

Once the buttons have been defined, the next step is to create the events that will send room control API commands to Mira Connect software.

Navigate to the **EVENTS** tab, and we'll configure the events for the buttons, as described in the following sections.

Events for Button 1: Volume Down

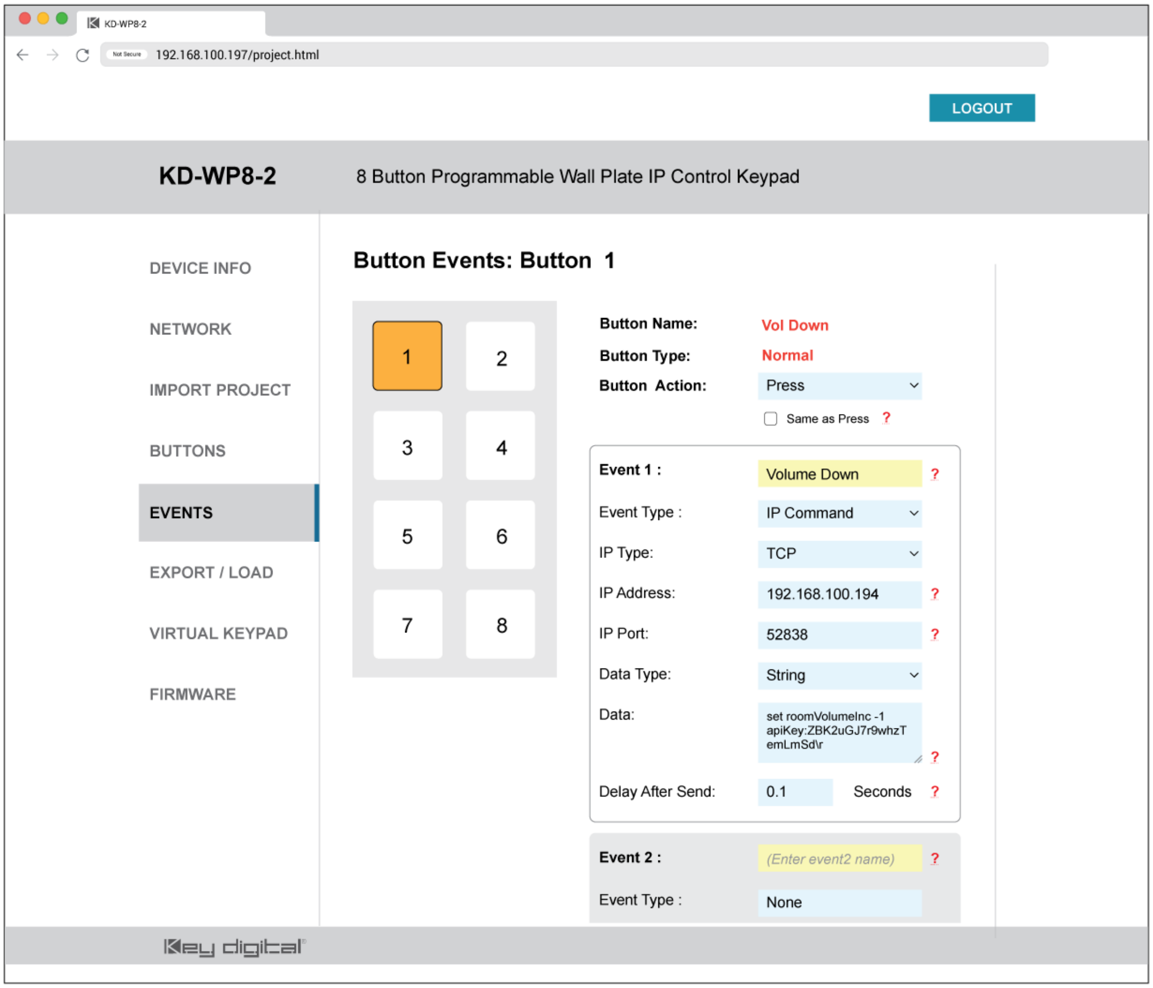
Button 1 will be configured to decrement the volume of the room.

As shown in the following figure, we'll configure the events as follows:

- For Event 1:
  - Label the event as **Volume Down**.
  - Select the event type to **IP Command**.
  - Set the IP Type to **TCP**.

- Set the IP address to the IP address of the Mira Connect PC: **192.168.100.194** in this example.
- Set the IP port to **52838**.
- Set the Data Type to **String**.
- Set the data to be the volumeInc command: **set roomVolumeInc -1 apiKey: ZBK2uGJ7r9whzTemLmSd\r**
  - ⋆ While this example sets the increment of -1 dB to 'decrement' the volume, any valid value can be used.
  - ⋆ This command explicitly includes the \r carriage return command terminator which has a value of 0x0D.
  - ⋆ This command uses the apiKey specific to this room.
- Set the Delay After Send value to **0.1** seconds.

- The remaining events are left empty.



KD-WP8-2 8 Button Programmable Wall Plate IP Control Keypad

Button Events: Button 1

Button Name: Vol Down
Button Type: Normal
Button Action: Press
□ Same as Press ?

Event 1 : Volume Down ?
Event Type : IP Command
IP Type: TCP
IP Address: 192.168.100.194 ?
IP Port: 52838 ?
Data Type: String
Data: set roomVolumeInc -1 apiKey:ZBK2uGJ7r9whzTemLmSd\r ?
Delay After Send: 0.1 Seconds ?

Event 2 : (Enter event2 name) ?
Event Type : None

Figure 36: Configure the events for Button 1 of the keypad.

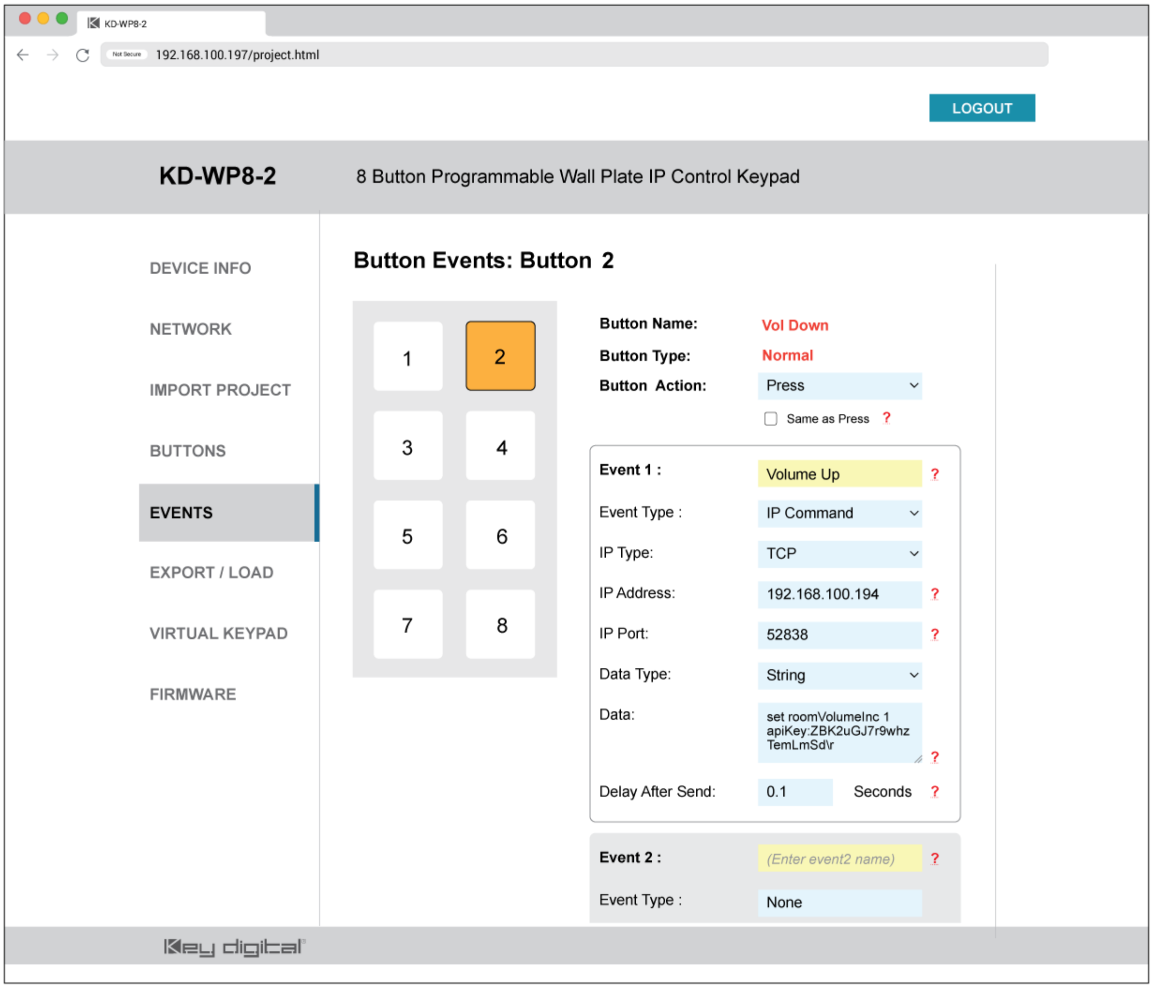
Events for Button 2: Volume Up

Button 2 will be configured to increment the volume of the room.

As shown in the following figure, we'll configure the events as follows:

- For Event 1:

  - Label the event as **Volume Up**.
  - Select the event type to **IP Command**.
  - Set the IP Type to **TCP**.
  - Set the IP address to the IP address of the Mira Connect PC: **192.168.100.194** in this example.
  - Set the IP port to **52838**.
  - Set the Data Type to **String**.
  - Set the data to be the volumeInc command: **set roomVolumeInc 1 apiKey: ZBK2uGJ7r9whzTemLmSd\r**
    - ⋆ While this example sets the increment to 1 dB to ‘increment’ the volume, any valid value can be used.
    - ⋆ This command explicitly includes the \r carriage return command terminator which has a value of 0x0D.
    - ⋆ This command uses the apiKey specific to this room.
  - Set the Delay After Send value to **0.1** seconds.
- The remaining events are left empty.



Button Events: Button 2

Button Name: Vol Down
Button Type: Normal
Button Action: Press
□ Same as Press ?

Event 1 : Volume Up ?
Event Type : IP Command
IP Type: TCP
IP Address: 192.168.100.194 ?
IP Port: 52838 ?
Data Type: String
Data: set roomVolumeInc 1 apiKey:ZBK2uGJ7r9whzTemLmSd\r ?
Delay After Send: 0.1 Seconds ?

Event 2 : (Enter event2 name) ?
Event Type : None

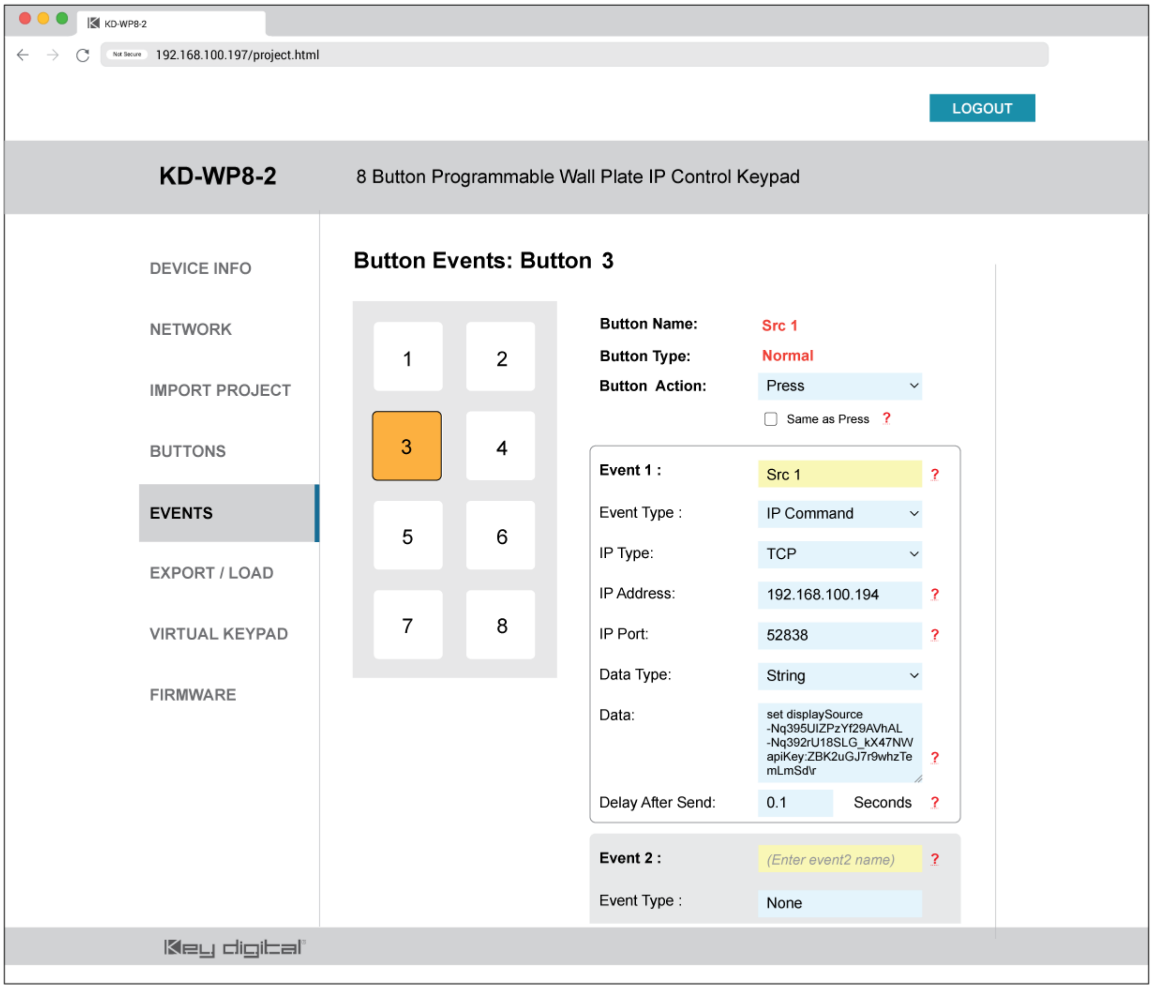
Figure 37: Configure the events for Button 2 of the keypad.

Events for Button 3: Video Input 1

Button 3 will be configured to route Input 1 of the video switcher to the display.

As shown in the following figure, we'll configure the events as follows:

- For Event 1:
  - Set the button Action to **Press**.
  - Label the event as **Src 1**.
  - Select the event type to **IP Command**.
  - Set the IP Type to **TCP**.
  - Set the IP address to the IP address of the Mira Connect PC: **192.168.100.194** in this example.
  - Set the IP port to **52838**.
  - Set the Data Type to **String**.
  - Set the data to be the volumeInc command:
    set displaySource -Nq395UIZPzYf29AVhAL -Nq392rU18SLG_kX47NW apiKey: ZBK2uGJ7r9whzTemLmSd\r
    - This example uses the display ID for the Sharp display and uses the source ID for the input 1 source.
    - This command explicitly includes the \r carriage return command terminator which has a value of 0x0D.
    - This command uses the apiKey specific to this room.
  - Set the Delay After Send value to **0.1** seconds.
- The remaining events are left empty.



KD-WP8-2 8 Button Programmable Wall Plate IP Control Keypad

Button Events: Button 3

Button Name: Src 1
Button Type: Normal
Button Action: Press
□ Same as Press

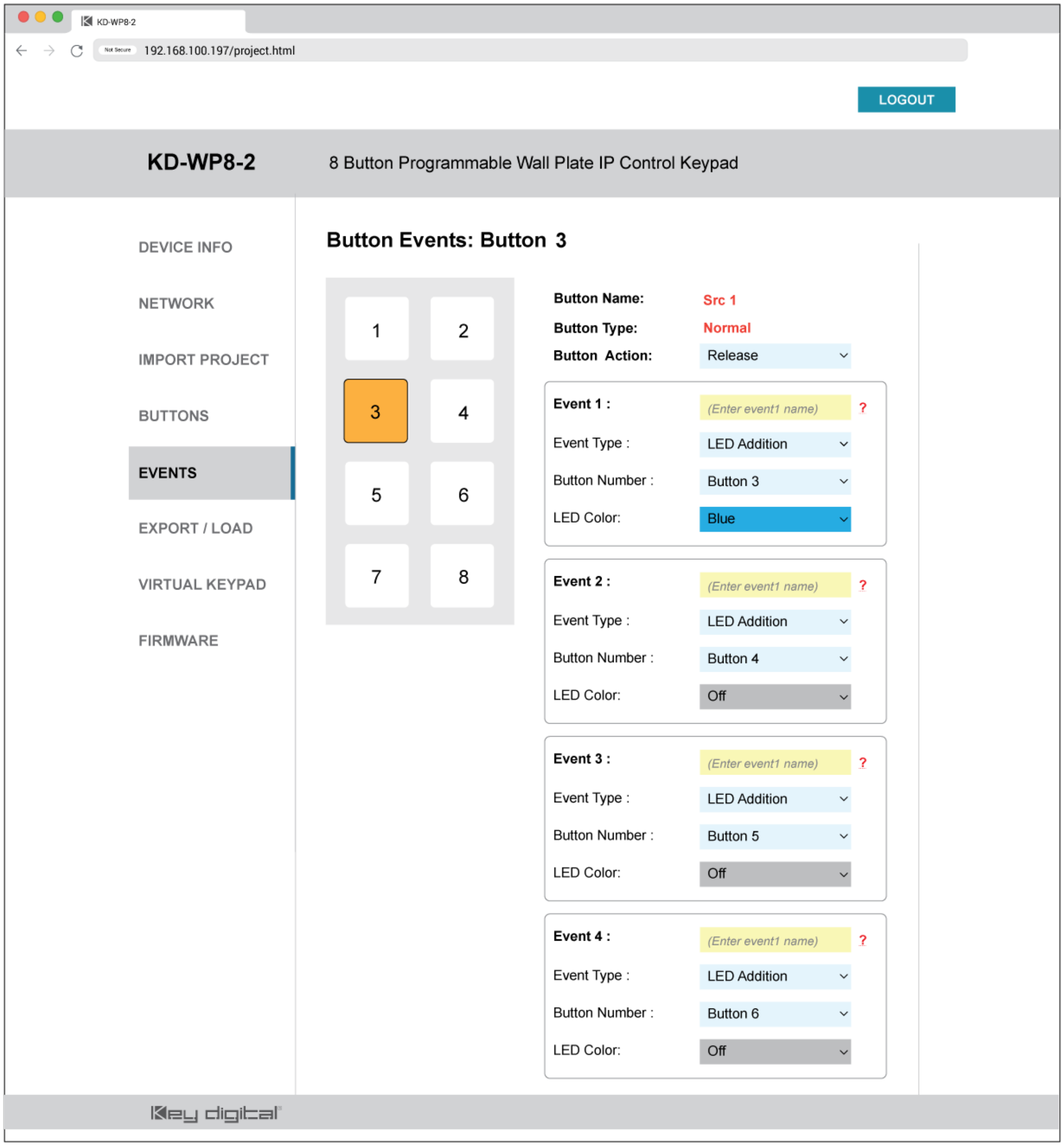
Event 1 : Src 1
Event Type : IP Command
IP Type: TCP
IP Address: 192.168.100.194
IP Port: 52838
Data Type: String
Data: set displaySource -Nq395UIZPzYf29AVhAL -Nq392rU18SLG_kX47NW apiKey:ZBK2uGJ7r9whzTemLmSd\r
Delay After Send: 0.1 Seconds

Event 2 : (Enter event2 name)
Event Type : None

Figure 38: Configure the events for Button 3 of the keypad.

In addition, as shown in the following figure, configure a **Release** event to illuminate Button 3 and to turn off the LEDs on the other source selection Buttons 4, 5, and 6 as follows:

- Set the button Action to **Release**.
- For Event 1:
  - Set the Event Type to: **LED Addition**.
  - Set the Button Number to **Button 3**.
  - Set the LED Color to **Blue**.
- For Event 2:
  - Set the Event Type to: **LED Addition**.
  - Set the Button Number to **Button 4**.
  - Set the LED Color to **Off**.
- For Event 3:
  - Set the Event Type to: **LED Addition**.
  - Set the Button Number to **Button 5**.
  - Set the LED Color to **Off**.
- For Event 4:
  - Set the Event Type to: **LED Addition**.
  - Set the Button Number to **Button 6**.
  - Set the LED Color to **Off**.



KD-WP8-2 8 Button Programmable Wall Plate IP Control Keypad

Button Events: Button 3

Button Name: Src 1
Button Type: Normal
Button Action: Release

Event 1 : (Enter event1 name) ?
Event Type : LED Addition
Button Number : Button 3
LED Color: Blue

Event 2 : (Enter event1 name) ?
Event Type : LED Addition
Button Number : Button 4
LED Color: Off

Event 3 : (Enter event1 name) ?
Event Type : LED Addition
Button Number : Button 5
LED Color: Off

Event 4 : (Enter event1 name) ?
Event Type : LED Addition
Button Number : Button 6
LED Color: Off

Figure 39: Configure the release events for Button 3 of the keypad.

Events for Button 4: Video Input 2

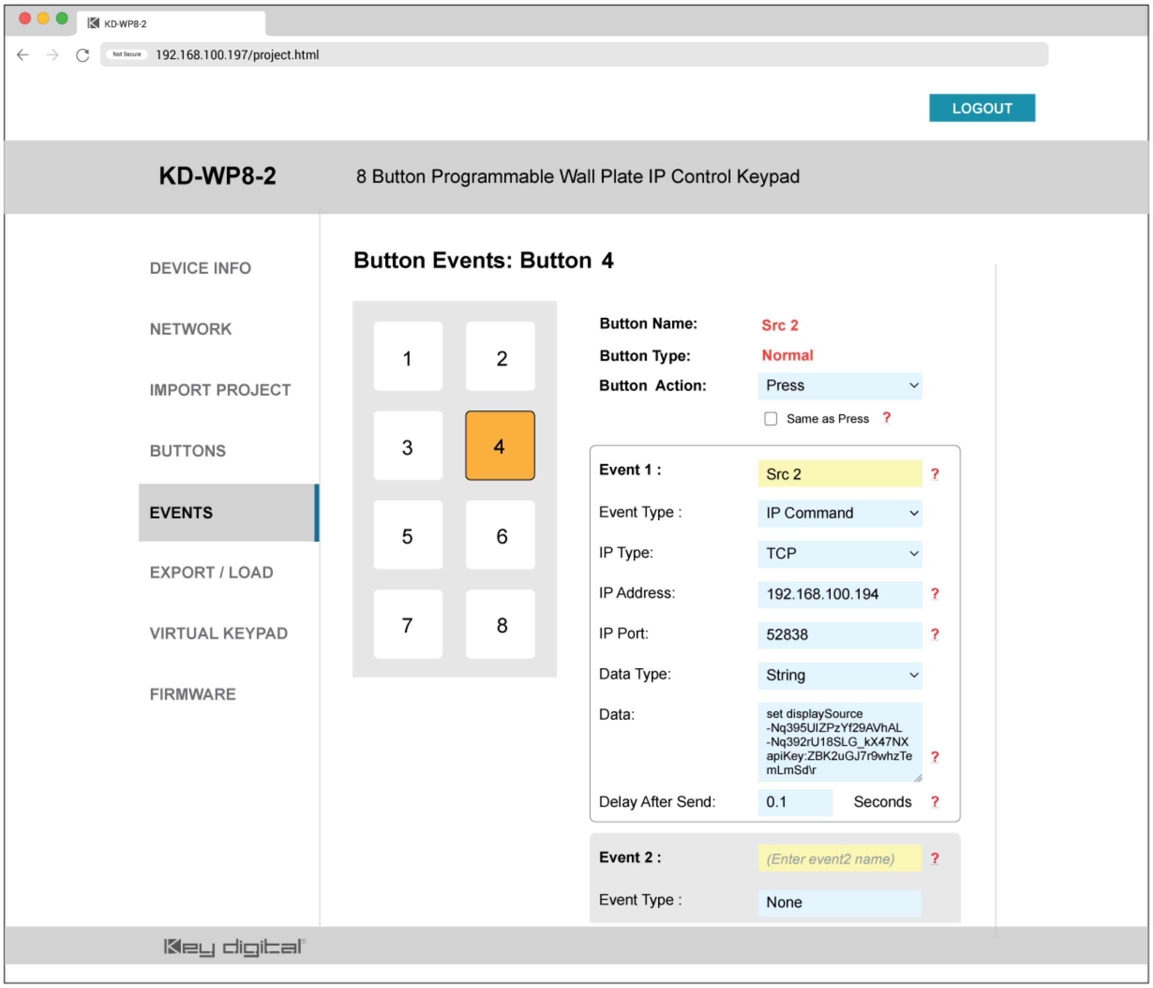
Button 4 will be configured to route Input 2 of the video switcher to the display.

As shown in the following figure, we'll configure the events as follows:

- For Event 1:
  - Set the button Action to **Press**.
  - Label the event as **Src 2**.

- Select the event type to **IP Command**.
- Set the IP Type to **TCP**.
- Set the IP address to the IP address of the Mira Connect PC: **192.168.100.194** in this example.
- Set the IP port to **52838**.
- Set the Data Type to **String**.
- Set the data to be the volumeInc command:
  set displaySource -Nq395UIZPzYf29AVhAL -Nq392rU18SLG_kX47NX apiKey: ZBK2uGJ7r9whzTemLmSd\r
  - ⋆ This example uses the display ID for the Sharp display and uses the source ID for the Input 2 source.
  - ⋆ This command explicitly includes the \r carriage return command terminator which has a value of 0x0D.
  - ⋆ This command uses the apiKey specific to this room.
- Set the Delay After Send value to **0.1** seconds.

- The remaining events are left empty.



192.168.100.197/project.html

KD-WP8-2 8 Button Programmable Wall Plate IP Control Keypad

Button Events: Button 4

Button Name: Src 2
Button Type: Normal
Button Action: Press
□ Same as Press ?

Event 1 : Src 2 ?
Event Type : IP Command
IP Type: TCP
IP Address: 192.168.100.194 ?
IP Port: 52838 ?
Data Type: String
Data: set displaySource -Nq395UIZPzYf29AVhAL -Nq392rU18SLG_kX47NX apiKey:ZBK2uGJ7r9whzTemLmSd\r ?
Delay After Send: 0.1 Seconds ?

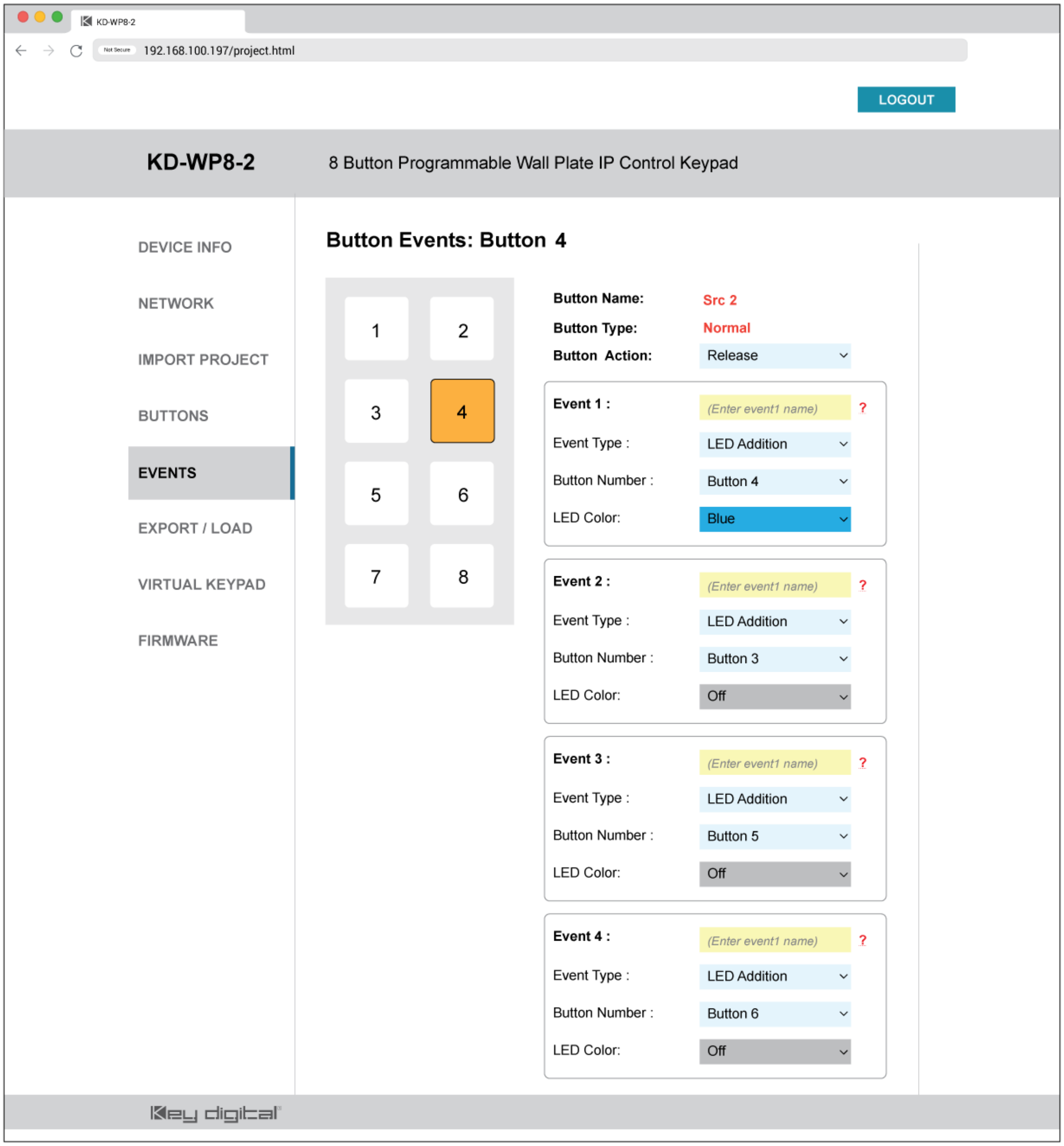
Event 2 : (Enter event2 name) ?
Event Type : None

Figure 40: Configure the events for Button 4 of the keypad.

In addition, as shown in the following figure, configure a **Release** event to illuminate Button 4 and to turn off the LEDs on buttons 3, 5, and 6 as follows:

- Set the button Action to **Release**.

- For Event 1:
  - Set the Event Type to: **LED Addition**.
  - Set the Button Number to **Button 4**.
  - Set the LED Color to **Blue**.
- For Event 2:
  - Set the Event Type to: **LED Addition**.
  - Set the Button Number to **Button 3**.
  - Set the LED Color to **Off**.
- For Event 3:
  - Set the Event Type to: **LED Addition**.
  - Set the Button Number to **Button 5**.
  - Set the LED Color to **Off**.
- For Event 4:
  - Set the Event Type to: **LED Addition**.
  - Set the Button Number to **Button 6**.
  - Set the LED Color to **Off**.



KD-WP8-2 8 Button Programmable Wall Plate IP Control Keypad

Button Events: Button 4

Button Name: Src 2
Button Type: Normal
Button Action: Release

Event 1 : (Enter event1 name)
Event Type : LED Addition
Button Number : Button 4
LED Color: Blue

Event 2 : (Enter event1 name)
Event Type : LED Addition
Button Number : Button 3
LED Color: Off

Event 3 : (Enter event1 name)
Event Type : LED Addition
Button Number : Button 5
LED Color: Off

Event 4 : (Enter event1 name)
Event Type : LED Addition
Button Number : Button 6
LED Color: Off

Figure 41: Configure the release events for Button 4 of the keypad.

Events for Button 5: Video Input 3

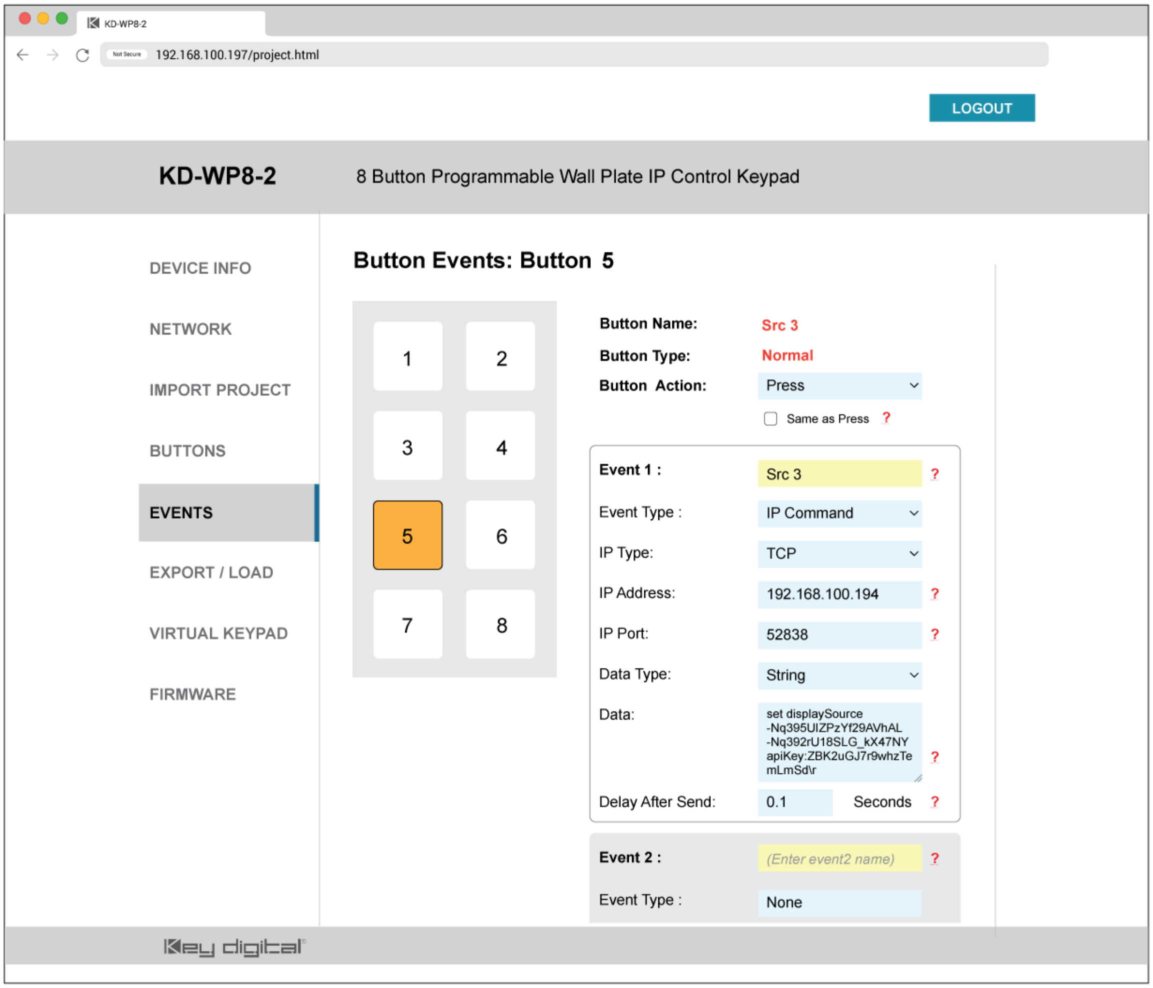
Button 5 will be configured to route Input 3 of the video switcher to the display.

As shown in the following figure, we'll configure the events as follows:

- For Event 1:
  - Set the button Action to **Press**.
  - Label the event as **Src 3**.

- Select the event type to **IP Command**.
- Set the IP Type to **TCP**.
- Set the IP address to the IP address of the Mira Connect PC: **192.168.100.194** in this example.
- Set the IP port to **52838**.
- Set the Data Type to **String**.
- Set the data to be the volumeInc command:
  set displaySource -Nq395UIZPzYf29AVhAL -Nq392rU18SLG_kX47NY apiKey: ZBK2uGJ7r9whzTemLmSd\r
  - This example uses the display ID for the Sharp display and uses the source ID for the input 3 source.
  - This command explicitly includes the \r carriage return command terminator which has a value of 0x0D.
  - This command uses the apiKey specific to this room.
- Set the Delay After Send value to **0.1** seconds.

- The remaining Press events are left empty.



KD-WP8-2 8 Button Programmable Wall Plate IP Control Keypad

Button Events: Button 5

Button Name: Src 3
Button Type: Normal
Button Action: Press
□ Same as Press ?

Event 1 : Src 3 ?
Event Type : IP Command
IP Type: TCP
IP Address: 192.168.100.194 ?
IP Port: 52838 ?
Data Type: String
Data: set displaySource -Nq395UIZPzYf29AVhAL -Nq392rU18SLG_kX47NY apiKey:ZBK2uGJ7r9whzTe mLmSd\r ?
Delay After Send: 0.1 Seconds ?

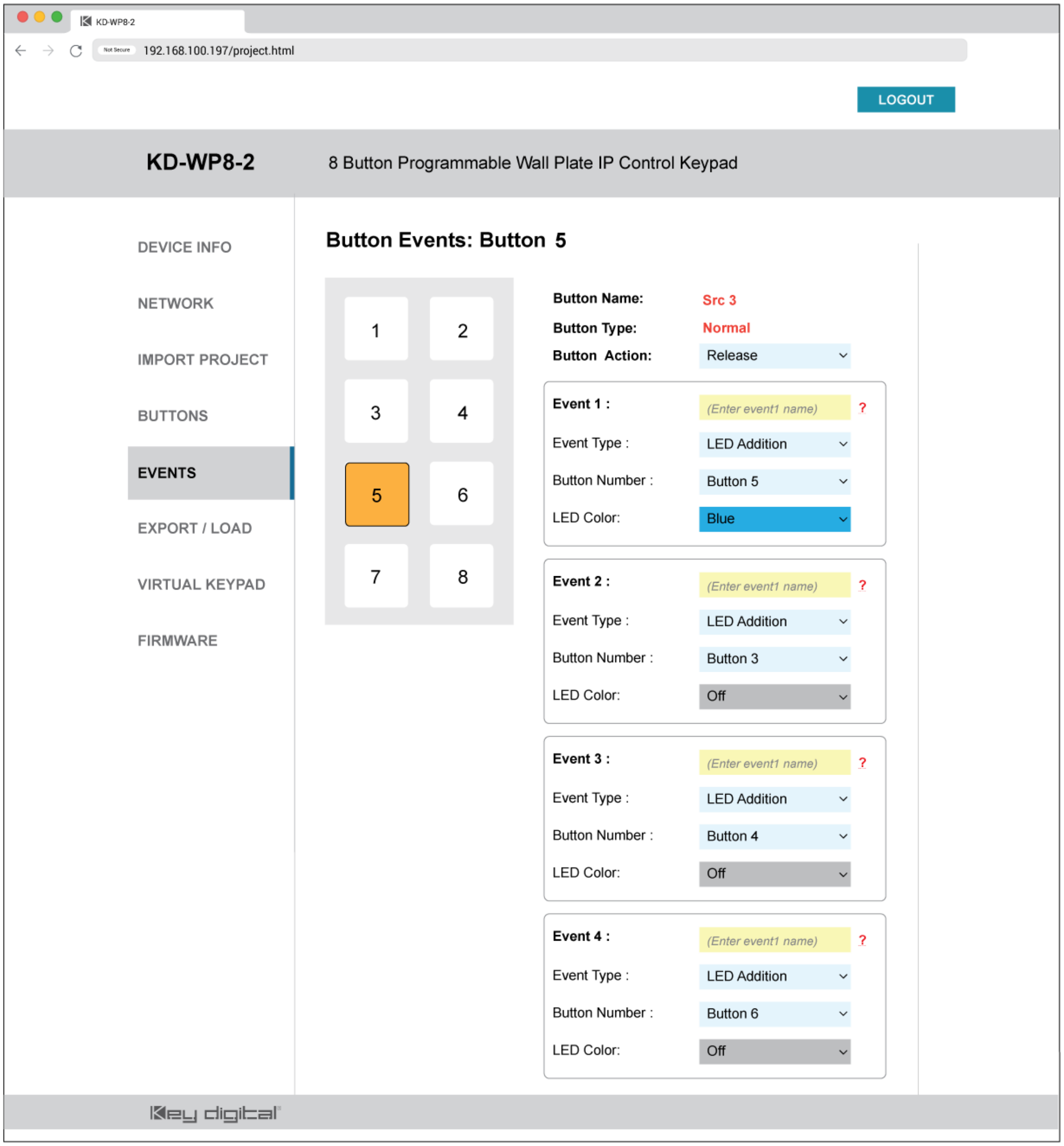
Event 2 : (Enter event2 name) ?
Event Type : None

Figure 42: Configure the events for Button 5 of the keypad.

In addition, as shown in the following figure, configure a **Release** event to illuminate Button 5 and to turn off the LEDs on buttons 3, 4, and 6 as follows:

- Set the button Action to **Release**.

- For Event 1:
  - Set the Event Type to: **LED Addition**.
  - Set the Button Number to **Button 5**.
  - Set the LED Color to **Blue**.
- For Event 2:
  - Set the Event Type to: **LED Addition**.
  - Set the Button Number to **Button 3**.
  - Set the LED Color to **Off**.
- For Event 3:
  - Set the Event Type to: **LED Addition**.
  - Set the Button Number to **Button 4**.
  - Set the LED Color to **Off**.
- For Event 4:
  - Set the Event Type to: **LED Addition**.
  - Set the Button Number to **Button 6**.
  - Set the LED Color to **Off**.



KD-WP8-2 8 Button Programmable Wall Plate IP Control Keypad

Button Events: Button 5

Button Name: Src 3
Button Type: Normal
Button Action: Release

Event 1 : (Enter event1 name) ?
Event Type : LED Addition
Button Number : Button 5
LED Color: Blue

Event 2 : (Enter event1 name) ?
Event Type : LED Addition
Button Number : Button 3
LED Color: Off

Event 3 : (Enter event1 name) ?
Event Type : LED Addition
Button Number : Button 4
LED Color: Off

Event 4 : (Enter event1 name) ?
Event Type : LED Addition
Button Number : Button 6
LED Color: Off

Figure 43: Configure the release events for Button 5 of the keypad.

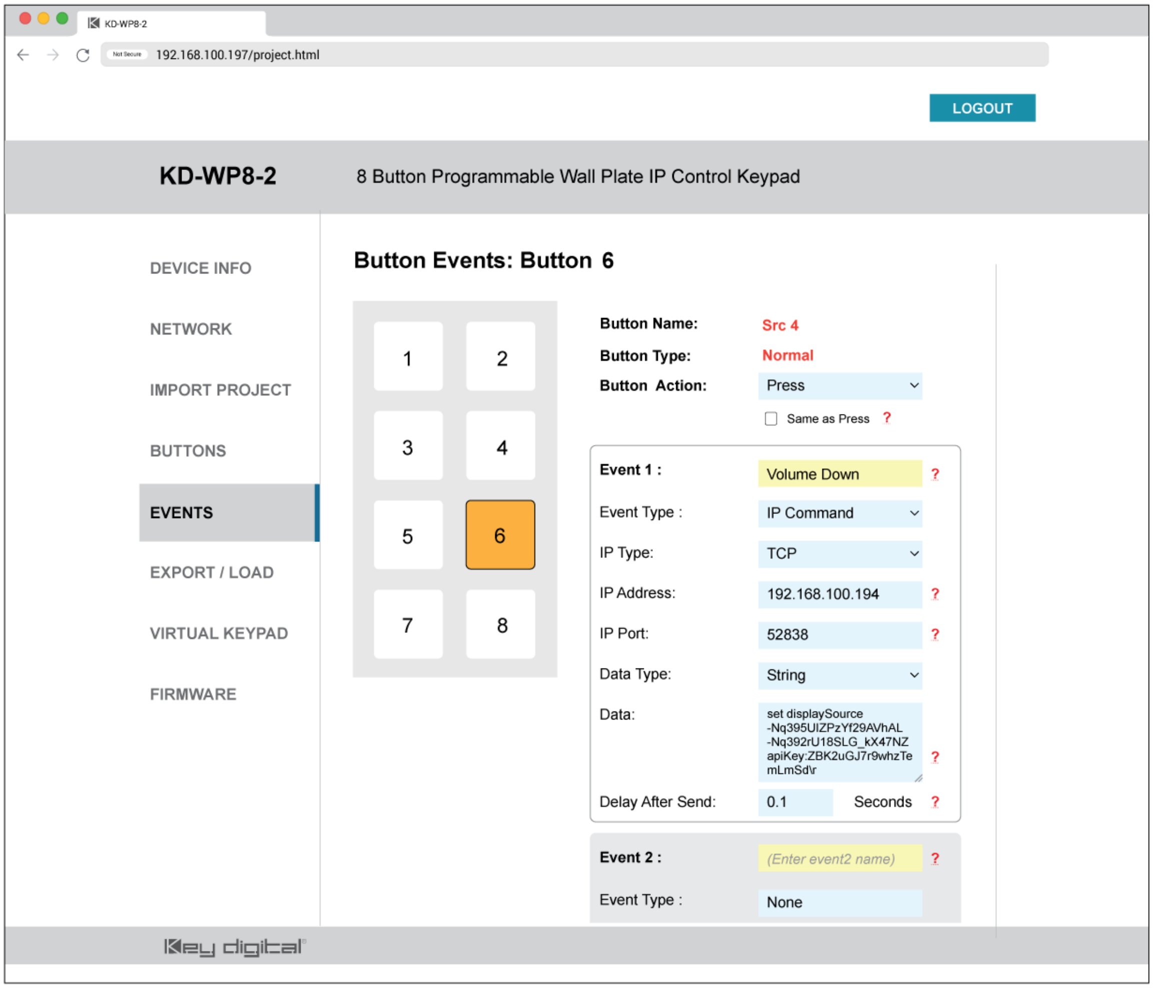
Events for Button 6: Video Input 4

Button 6 will be configured to route Input 4 of the video switcher to the display.

As shown in the following figure, we'll configure the events as follows:

- For Event 1:
  - Set the button Action to **Press**.
  - Label the event as **Src 4**.

  - Select the event type to **IP Command**.
  - Set the IP Type to **TCP**.
  - Set the IP address to the IP address of the Mira Connect PC: **192.168.100.194** in this example.
  - Set the IP port to **52838**.
  - Set the Data Type to **String**.
  - Set the data to be the displaySource command:
    set displaySource -Nq395UIZPzYf29AVhAL -Nq392rU18SLG_kX47NZ apiKey: ZBK2uGJ7r9whzTemLmSd\r
    - ⋆ This example uses the display ID for the Sharp display and uses the source ID for the input 4 source.
    - ⋆ This command explicitly includes the \r carriage return command terminator which has a value of 0x0D.
    - ⋆ This command uses the apiKey specific to this room.
  - Set the Delay After Send value to **0.1** seconds.
- The remaining Press events are left empty.



Button Events: Button 6

Button Name: Src 4
Button Type: Normal
Button Action: Press
□ Same as Press ?

Event 1 : Volume Down ?
Event Type : IP Command
IP Type: TCP
IP Address: 192.168.100.194 ?
IP Port: 52838 ?
Data Type: String
Data: set displaySource -Nq395UIZPzYf29AVhAL -Nq392rU18SLG_kX47NZ apiKey:ZBK2uGJ7r9whzTemLmSd\r ?
Delay After Send: 0.1 Seconds ?

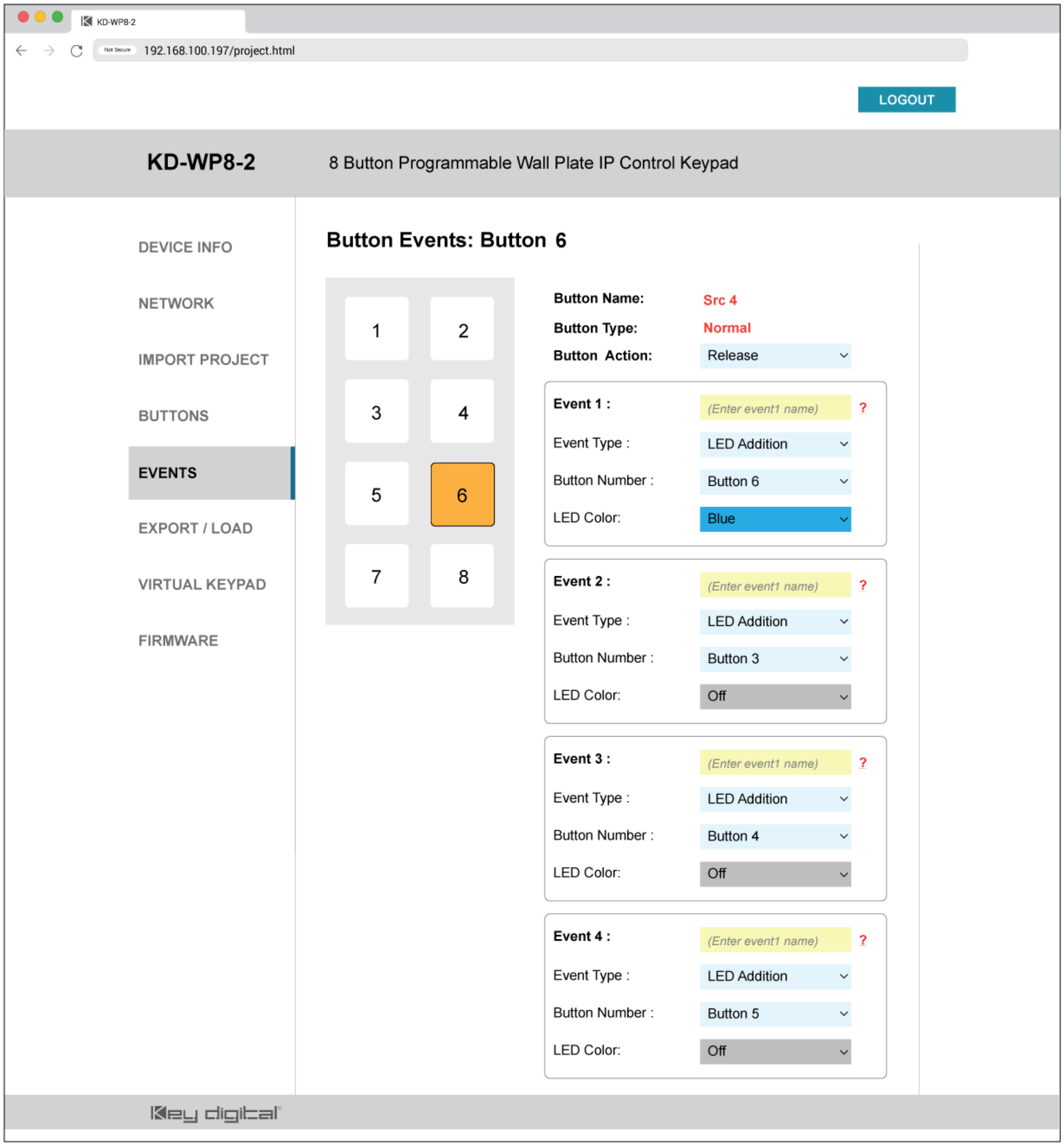
Event 2 : (Enter event2 name) ?
Event Type : None

Figure 44: Configure the events for Button 6 of the keypad.

In addition, as shown in the following figure, configure a **Release** event to illuminate Button 6 and to turn off the LEDs on buttons 3, 4, and 5 as follows:

- Set the button Action to **Release**.

- For Event 1:
  - Set the Event Type to: **LED Addition**.
  - Set the Button Number to **Button 6**.
  - Set the LED Color to **Blue**.
- For Event 2:
  - Set the Event Type to: **LED Addition**.
  - Set the Button Number to **Button 3**.
  - Set the LED Color to **Off**.
- For Event 3:
  - Set the Event Type to: **LED Addition**.
  - Set the Button Number to **Button 4**.
  - Set the LED Color to **Off**.
- For Event 4:
  - Set the Event Type to: **LED Addition**.
  - Set the Button Number to **Button 5**.
  - Set the LED Color to **Off**.configure



KD-WP8-2 8 Button Programmable Wall Plate IP Control Keypad

Button Events: Button 6

Button Name: Src 4
Button Type: Normal
Button Action: Release

Event 1 : (Enter event1 name) ?
Event Type : LED Addition
Button Number : Button 6
LED Color: Blue

Event 2 : (Enter event1 name) ?
Event Type : LED Addition
Button Number : Button 3
LED Color: Off

Event 3 : (Enter event1 name) ?
Event Type : LED Addition
Button Number : Button 4
LED Color: Off

Event 4 : (Enter event1 name) ?
Event Type : LED Addition
Button Number : Button 5
LED Color: Off

Figure 45: Configure the release events for Button 6 of the keypad.

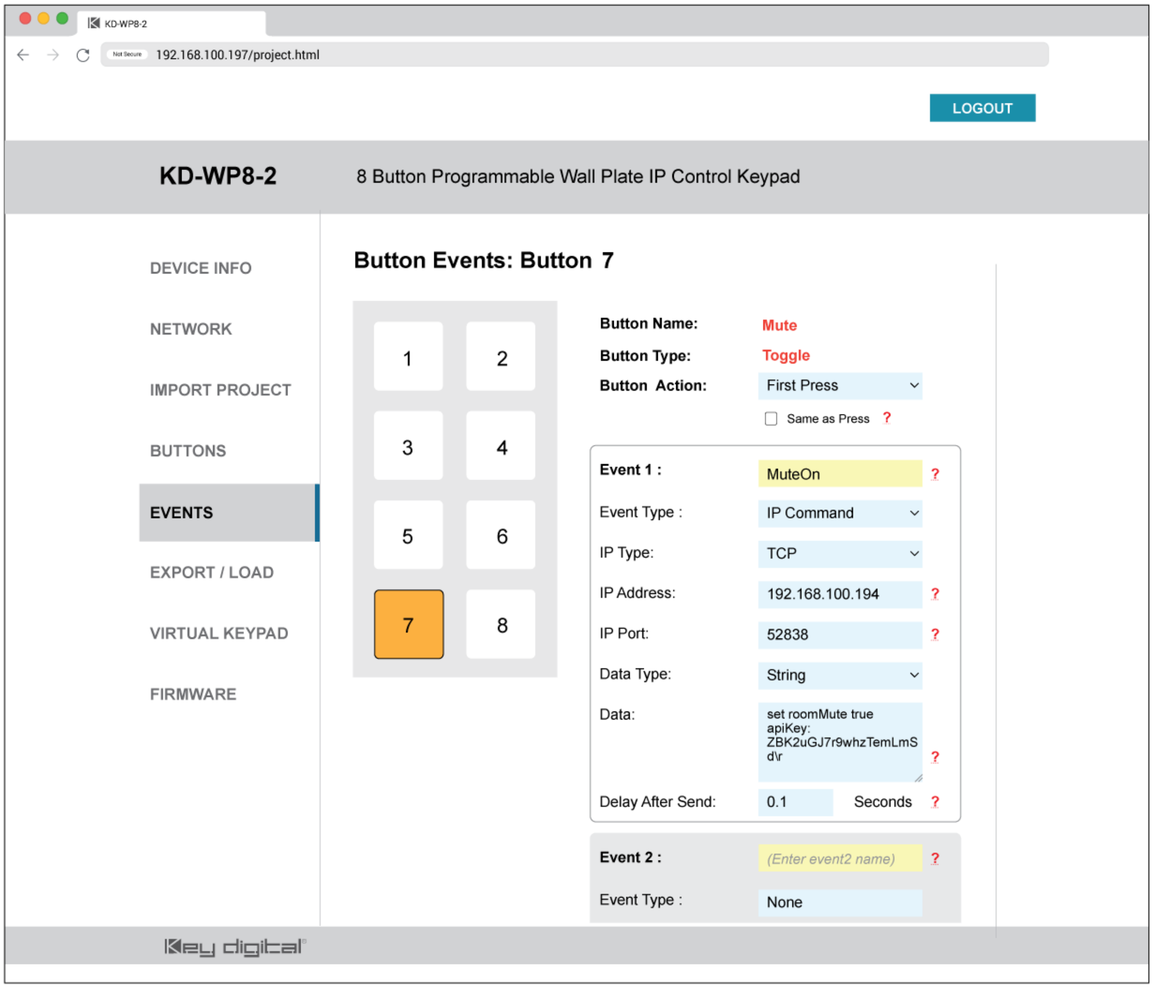
Events for Button 7: Mute

Button 7 will be configured to mute the room microphones on the first press and to unmute the microphones on the second press.

As shown in the following figure, we'll configure the events as follows:

- Set the button Action to **First Press**.
- Label the event as **MuteOn**.
- Select the event type to **IP Command**.

- Set the IP Type to **TCP**.
- Set the IP address to the IP address of the Mira Connect PC: **192.168.100.194** in this example.
- Set the IP port to **52838**.
- Set the Data Type to **String**.
- Set the data to be the mute command: **set roomMute true apiKey: ZBK2uGJ7r9whzTemLmSd\r**
  - This command explicitly includes the \r carriage return command terminator which has a value of 0x0D.
  - This command uses the apiKey specific to this room.
- Set the Delay After Send value to **0.1** seconds.



Button Events: Button 7

Button Name: Mute
Button Type: Toggle
Button Action: First Press
□ Same as Press ?

Event 1 : MuteOn ?
Event Type : IP Command
IP Type: TCP
IP Address: 192.168.100.194 ?
IP Port: 52838 ?
Data Type: String
Data: set roomMute true apiKey: ZBK2uGJ7r9whzTemLmSd\r ?
Delay After Send: 0.1 Seconds ?

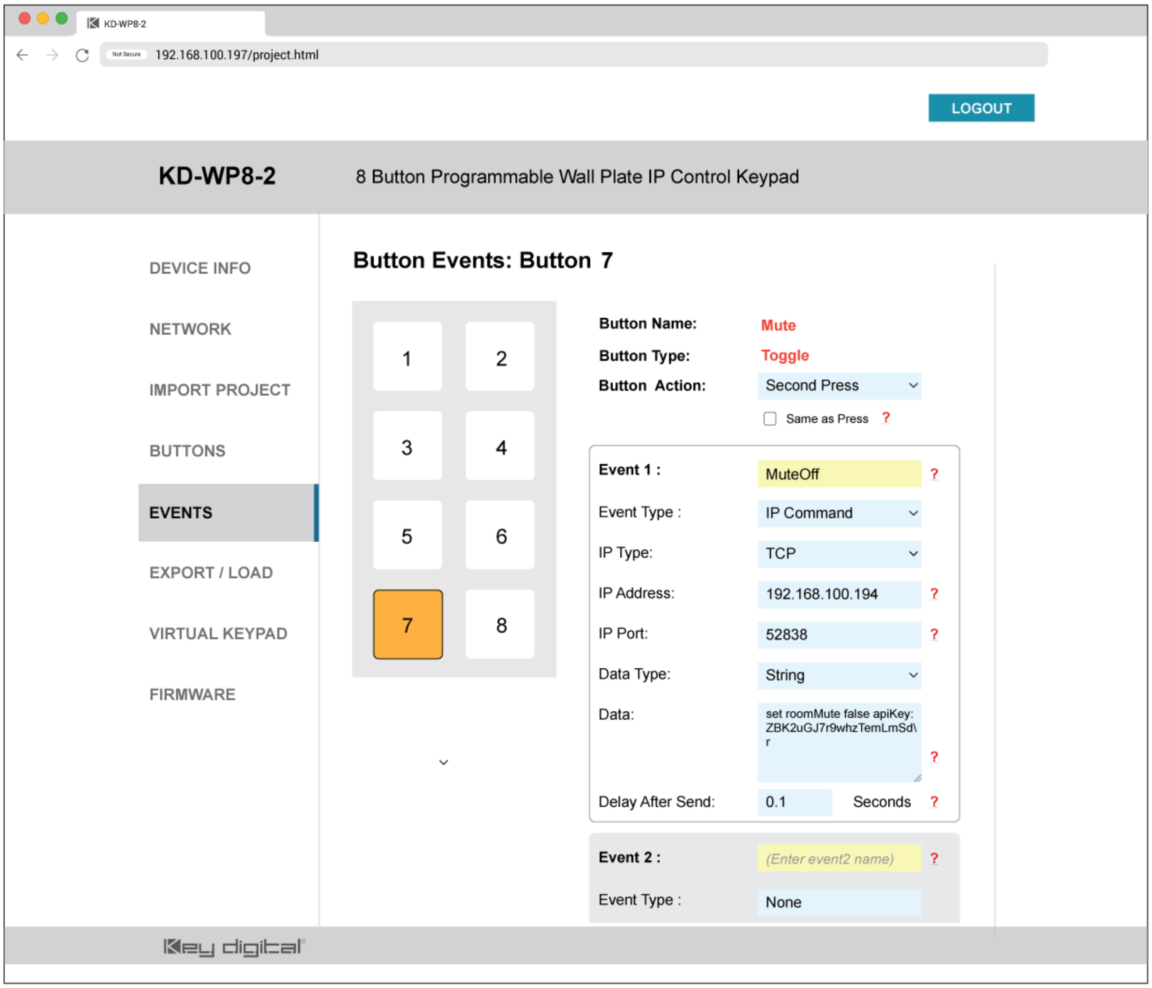
Event 2 : (Enter event2 name) ?
Event Type : None

Figure 46: Configure the events for Button 7 of the keypad.

In addition, configure a Second Press as follows:

- Set the button Action to **Second Press**.
- Label the event as **MuteOff**.
- Select the event type to **IP Command**.
- Set the IP Type to **TCP**.
- Set the IP address to the IP address of the Mira Connect PC: **192.168.100.194** in this example.

- Set the IP port to **52838**.
- Set the Data Type to **String**.
- Set the data to be the mute command: **set roomMute false apiKey: ZBK2uGJ7r9whzTemLmSd\r**
    - This command explicitly includes the \r carriage return command terminator which has a value of 0x0D.
    - This command uses the apiKey specific to this room.
- Set the Delay After Send value to **0.1** seconds.



KD-WP8-2 8 Button Programmable Wall Plate IP Control Keypad

Button Events: Button 7

Button Name: Mute
Button Type: Toggle
Button Action: Second Press
□ Same as Press ?

Event 1 : MuteOff ?
Event Type : IP Command
IP Type: TCP
IP Address: 192.168.100.194 ?
IP Port: 52838 ?
Data Type: String
Data: set roomMute false apiKey: ZBK2uGJ7r9whzTemLmSd\r ?
Delay After Send: 0.1 Seconds ?

Event 2 : (Enter event2 name) ?
Event Type : None

Figure 47: Configure the second press events for Button 7 of the keypad.

Events for Button 8: Power Off

Button 8 will be configured to power off the room which will power off the display and reset the volume in the room to the middle of the range.

In this example, we'll also turn the button red when pressing it, and after the room power off command has executed, we'll turn off the LEDs on the video source selections, and after 0.5 seconds, we'll set this button's LED color to blue on to indicate the room power off function has executed.

Finally, we'll send the microphone mute command and set button 7 LED color to red to indicate the room is muted.

As shown in the following figure, we'll configure the events as follows:

- Set the button Action to **Hold for Send**.
- For Event 1:
  - Label the event as **PowerOff**.
  - Select the event type to **IP Command**.
  - Set the IP Type to **TCP**.
  - Set the IP address to the IP address of the Mira Connect PC: **192.168.100.194** in this example.
  - Set the IP port to **52838**.
  - Set the Data Type to **String**.
  - Set the data to be the roomOff command: **set roomOff apiKey: ZBK2uGJ7r9whzTemLmSd\r**
    - This command explicitly includes the \r carriage return command terminator which has a value of 0x0D.
    - This command uses the apiKey specific to this room.
  - Set the Delay After Send value to **0.1** seconds.
- For Event 2:
  - Set the Event Type to **LED Addition**.
  - Set the button number to **Button 3**.
  - Set the LED Color to **Off**.
- For Event 3:
  - Set the Event Type to **LED Addition**.
  - Set the button number to **Button 4**.
  - Set the LED Color to **Off**.
- For Event 4:
  - Set the Event Type to **LED Addition**.
  - Set the button number to **Button 5**.
  - Set the LED Color to **Off**.
- For Event 5:
  - Set the Event Type to **LED Addition**.
  - Set the button number to **Button 6**.
  - Set the LED Color to **Off**.
- For Event 6:
  - Set the Event Type to **LED Addition**.
  - Set the button number to **Button 8**.
  - Set the LED Color to **Blue.**
- For Event 7:
  - Set the Event Type to **Delay Addition**.
  - Set the Delay Time to **0.5** seconds.
- For Event 8:
  - Set the Event Type to **LED Addition**.
  - Set the button number to **Button 8**.

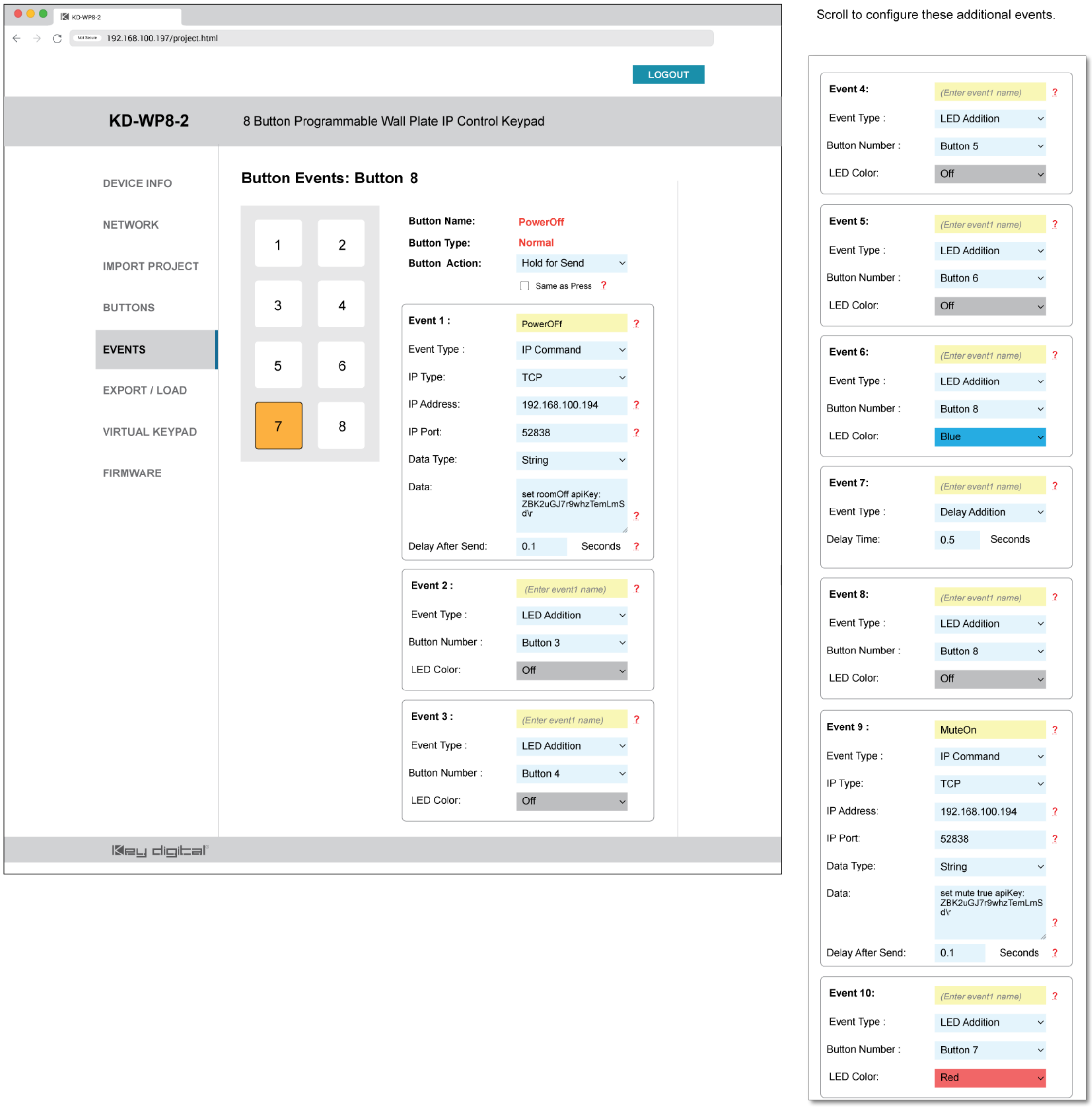
  - Set the LED Color to **Off.**

- For Event 9:

  - Label the event as **MuteOn**.
  - Select the event type to **IP Command**.
  - Set the IP Type to **TCP**.
  - Set the IP address to the IP address of the Mira Connect PC: **192.168.100.194** in this example.
  - Set the IP port to **52838**.
  - Set the Data Type to **String**.
  - Set the data to be the roomOff command: **set roomMute true apiKey: ZBK2uGJ7r9whzTemLmSd\r**
    - This command explicitly includes the \r carriage return command terminator which has a value of 0x0D.
    - This command uses the apiKey specific to this room.
  - Set the Delay After Send value to **0.1** seconds.

- For Event 10:

  - Set the Event Type to **LED Addition**.
  - Set the button number to **Button 7**.
  - Set the LED Color to **Red** to indicate the system is muted.



Scroll to configure these additional events.

KD-WP8-2 — 8 Button Programmable Wall Plate IP Control Keypad

Button Events: Button 8

Button Name: PowerOff
Button Type: Normal
Button Action: Hold for Send
□ Same as Press

Event 1 : PowerOFf
Event Type : IP Command
IP Type: TCP
IP Address: 192.168.100.194
IP Port: 52838
Data Type: String
Data: set roomOff apiKey: ZBK2uGJ7r9whzTemLmSd\r
Delay After Send: 0.1 Seconds

Event 2 : (Enter event1 name)
Event Type : LED Addition
Button Number : Button 3
LED Color: Off

Event 3 : (Enter event1 name)
Event Type : LED Addition
Button Number : Button 4
LED Color: Off

Event 4: (Enter event1 name)
Event Type : LED Addition
Button Number : Button 5
LED Color: Off

Event 5: (Enter event1 name)
Event Type : LED Addition
Button Number : Button 6
LED Color: Off

Event 6: (Enter event1 name)
Event Type : LED Addition
Button Number : Button 8
LED Color: Blue

Event 7: (Enter event1 name)
Event Type : Delay Addition
Delay Time: 0.5 Seconds

Event 8: (Enter event1 name)
Event Type : LED Addition
Button Number : Button 8
LED Color: Off

Event 9 : MuteOn
Event Type : IP Command
IP Type: TCP
IP Address: 192.168.100.194
IP Port: 52838
Data Type: String
Data: set mute true apiKey: ZBK2uGJ7r9whzTemLmSd\r
Delay After Send: 0.1 Seconds

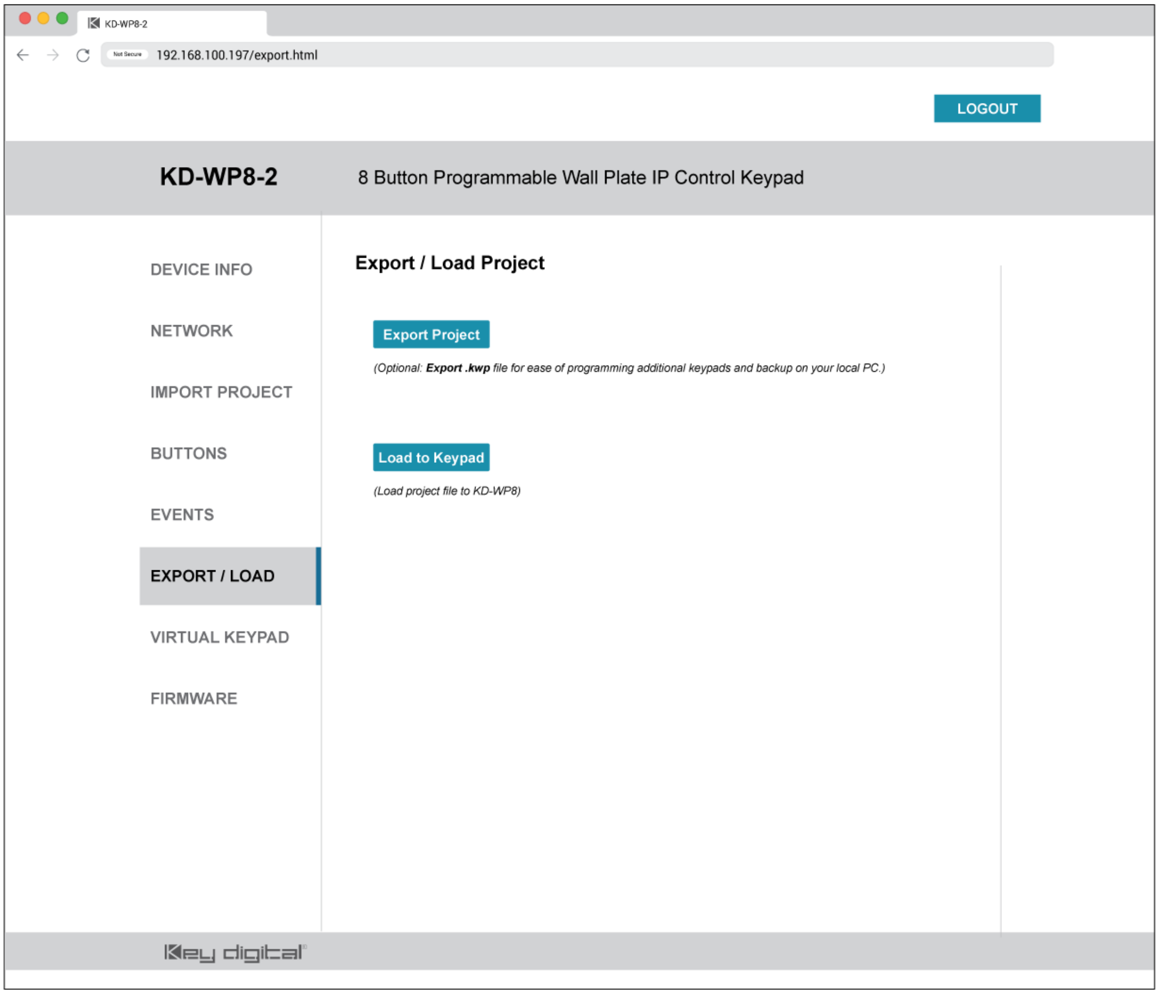
Event 10: (Enter event1 name)
Event Type : LED Addition
Button Number : Button 7
LED Color: Red

Figure 48: Configure the events for Button 8 of the keypad.

Using the KeyDigital Keypad

Once the buttons have been configured, the keypad is almost ready to be used.

First, we must load the configuration into the keypad. To load the configuration, navigate to the **EXPORT / LOAD** option and click the **Load to Keypad** button to load the configuration into the keypad.

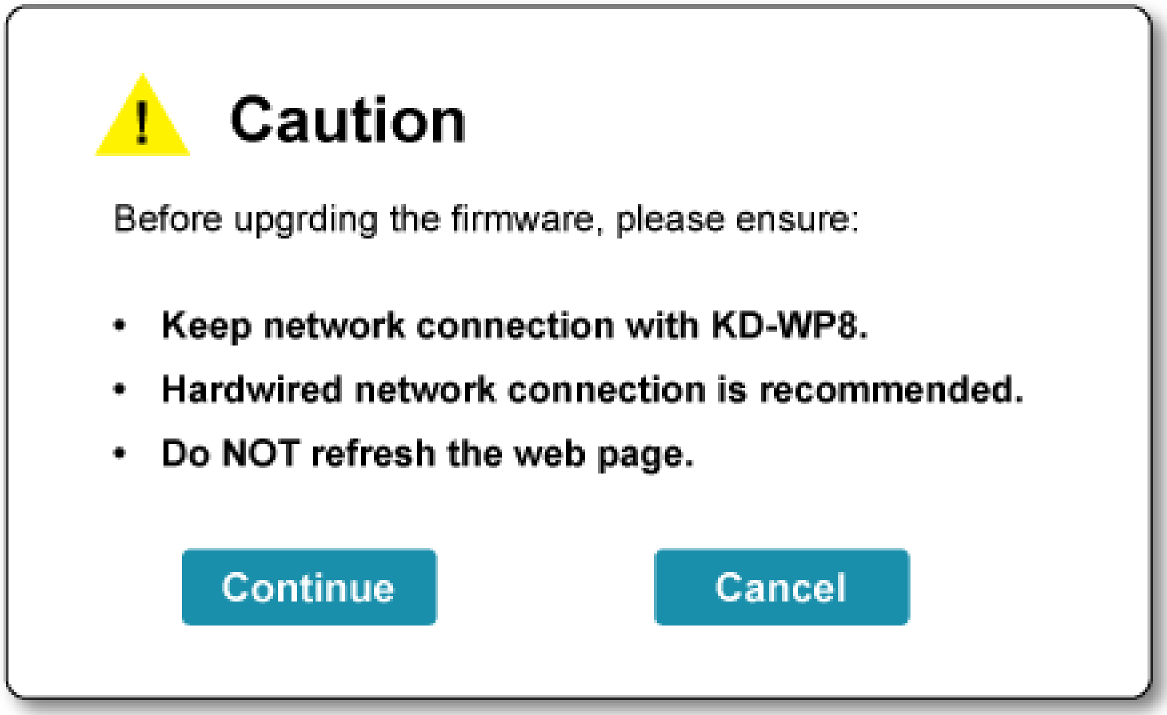


*(Optional: **Export .kwp** file for ease of programming additional keypads and backup on your local PC.)*

(Load project file to KD-WP8)

Figure 49: Load the settings to the keypad from the EXPORT/LOAD page.

When presented with a confirmation regarding firmware updates after clicking **Load To Keypad**, click **Continue**. While this seems like an odd message, click **Continue** to load the configuration file into the keypad.



Caution

Before upgrding the firmware, please ensure:

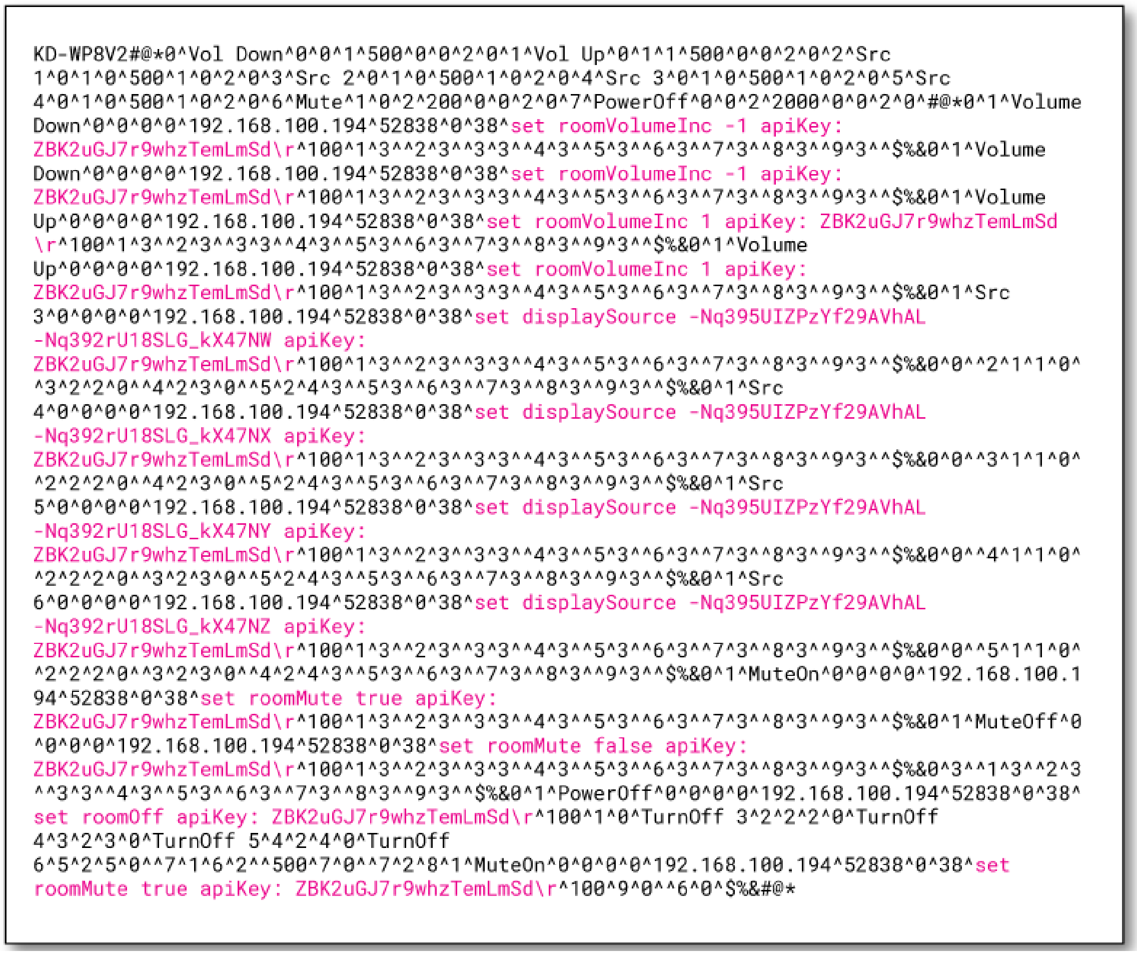
- Keep network connection with KD-WP8.
- Hardwired network connection is recommended.
- Do NOT refresh the web page.

Figure 50: Click ‘continue’ to confirm loading the settings to the keypad.

From the EXPORT / LOAD page you can also export the settings for the keypad to a .kwp file for backing up the keypad or for deploying to additional keypads.

In this example, the exported KWP file is shown below and available for download from here. You can see the file includes the volume, mute, display source, and room power-off commands and events.

> NOTE: Advanced users can modify this file by changing the display ID, source IDs, and Mira Connect IP addresses when using the keypad with additional other rooms. Each room will have different display and source IDs and a unique API Key.



```
KD-WP8V2#@*0^Vol Down^0^0^1^500^0^0^2^0^1^Vol Up^0^1^1^500^0^0^2^0^2^Src
1^0^1^0^500^1^0^2^0^3^Src 2^0^1^0^500^1^0^2^0^4^Src 3^0^1^0^500^1^0^2^0^5^Src
4^0^1^0^500^1^0^2^0^6^Mute^1^0^2^200^0^0^2^0^7^PowerOff^0^0^2^2000^0^0^2^0^#@*0^1^Volume
Down^0^0^0^0^192.168.100.194^52838^0^38^set roomVolumeInc -1 apiKey:
ZBK2uGJ7r9whzTemLmSd\r^100^1^3^^2^3^^3^3^^4^3^^5^3^^6^3^^7^3^^8^3^^9^3^^$%&0^1^Volume
Down^0^0^0^0^192.168.100.194^52838^0^38^set roomVolumeInc -1 apiKey:
ZBK2uGJ7r9whzTemLmSd\r^100^1^3^^2^3^^3^3^^4^3^^5^3^^6^3^^7^3^^8^3^^9^3^^$%&0^1^Volume
Up^0^0^0^0^192.168.100.194^52838^0^38^set roomVolumeInc 1 apiKey: ZBK2uGJ7r9whzTemLmSd
\r^100^1^3^^2^3^^3^3^^4^3^^5^3^^6^3^^7^3^^8^3^^9^3^^$%&0^1^Volume
Up^0^0^0^0^192.168.100.194^52838^0^38^set roomVolumeInc 1 apiKey:
ZBK2uGJ7r9whzTemLmSd\r^100^1^3^^2^3^^3^3^^4^3^^5^3^^6^3^^7^3^^8^3^^9^3^^$%&0^1^Src
3^0^0^0^0^192.168.100.194^52838^0^38^set displaySource -Nq395UIZPzYf29AVhAL
-Nq392rU18SLG_kX47NW apiKey:
ZBK2uGJ7r9whzTemLmSd\r^100^1^3^^2^3^^3^3^^4^3^^5^3^^6^3^^7^3^^8^3^^9^3^^$%&0^0^^2^1^1^0^
^3^2^2^0^^4^2^3^0^^5^2^4^3^^5^3^^6^3^^7^3^^8^3^^9^3^^$%&0^1^Src
4^0^0^0^0^192.168.100.194^52838^0^38^set displaySource -Nq395UIZPzYf29AVhAL
-Nq392rU18SLG_kX47NX apiKey:
ZBK2uGJ7r9whzTemLmSd\r^100^1^3^^2^3^^3^3^^4^3^^5^3^^6^3^^7^3^^8^3^^9^3^^$%&0^0^^3^1^1^0^
^2^2^2^0^^4^2^3^0^^5^2^4^3^^5^3^^6^3^^7^3^^8^3^^9^3^^$%&0^1^Src
5^0^0^0^0^192.168.100.194^52838^0^38^set displaySource -Nq395UIZPzYf29AVhAL
-Nq392rU18SLG_kX47NY apiKey:
ZBK2uGJ7r9whzTemLmSd\r^100^1^3^^2^3^^3^3^^4^3^^5^3^^6^3^^7^3^^8^3^^9^3^^$%&0^0^^4^1^1^0^
^2^2^2^0^^3^2^3^0^^5^2^4^3^^5^3^^6^3^^7^3^^8^3^^9^3^^$%&0^1^Src
6^0^0^0^0^192.168.100.194^52838^0^38^set displaySource -Nq395UIZPzYf29AVhAL
-Nq392rU18SLG_kX47NZ apiKey:
ZBK2uGJ7r9whzTemLmSd\r^100^1^3^^2^3^^3^3^^4^3^^5^3^^6^3^^7^3^^8^3^^9^3^^$%&0^0^^5^1^1^0^
^2^2^2^0^^3^2^3^0^^4^2^4^3^^5^3^^6^3^^7^3^^8^3^^9^3^^$%&0^1^MuteOn^0^0^0^0^192.168.100.1
94^52838^0^38^set roomMute true apiKey:
ZBK2uGJ7r9whzTemLmSd\r^100^1^3^^2^3^^3^3^^4^3^^5^3^^6^3^^7^3^^8^3^^9^3^^$%&0^1^MuteOff^0
^0^0^0^192.168.100.194^52838^0^38^set roomMute false apiKey:
ZBK2uGJ7r9whzTemLmSd\r^100^1^3^^2^3^^3^3^^4^3^^5^3^^6^3^^7^3^^8^3^^9^3^^$%&0^3^^1^3^^2^3
^^3^3^^4^3^^5^3^^6^3^^7^3^^8^3^^9^3^^$%&0^1^PowerOff^0^0^0^0^192.168.100.194^52838^0^38^
set roomOff apiKey: ZBK2uGJ7r9whzTemLmSd\r^100^1^0^TurnOff 3^2^2^2^0^TurnOff
4^3^2^3^0^TurnOff 5^4^2^4^0^TurnOff
6^5^2^5^0^^7^1^6^2^^500^7^0^^7^2^8^1^MuteOn^0^0^0^0^192.168.100.194^52838^0^38^set
roomMute true apiKey: ZBK2uGJ7r9whzTemLmSd\r^100^9^0^^6^0^$%&#@*
```

Figure 51: The KeyDigital exported settings file with the API commands highlighted.

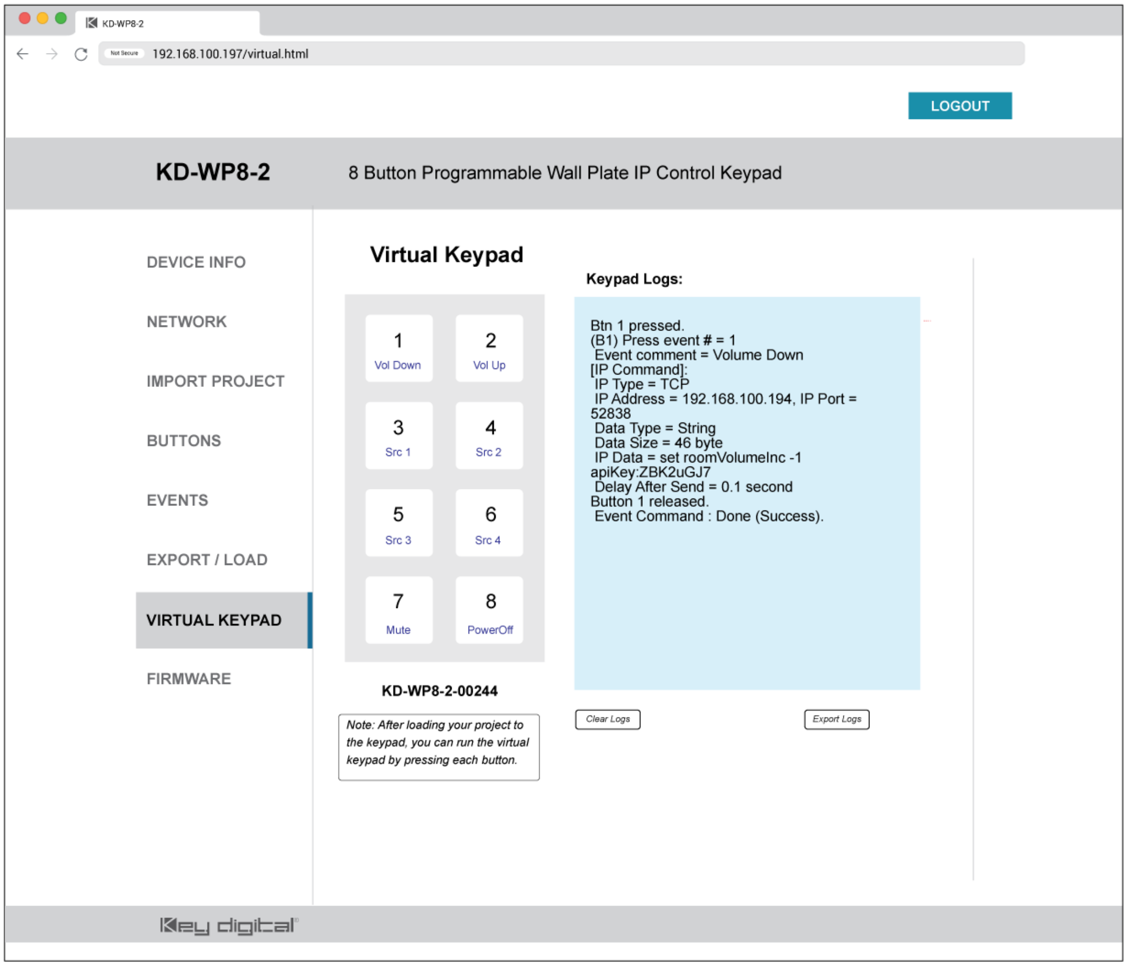
Once the button definitions and events have been loaded, the keypad is ready to be tested and used.

Step 9 — Testing the Keypad

To confirm operation, navigate to the **VIRTUAL KEYPAD** tab and press Button 1: Vol Down on either the actual keypad or on the virtual keypad shown on the **VIRTUAL KEYPAD** page. You should see the button turn red while pressed, and there should be a log entry, as shown in the following figure.

If the button is configured correctly, you will see that a TCP connection was created to Mira Connect at 192.168.100.194 and that the `set roomVolumeInc` command was executed successfully.

For each command sent, the keypad creates a TCP connection to Mira Connect software at port 52838, sends the command, and then closes the network connection.



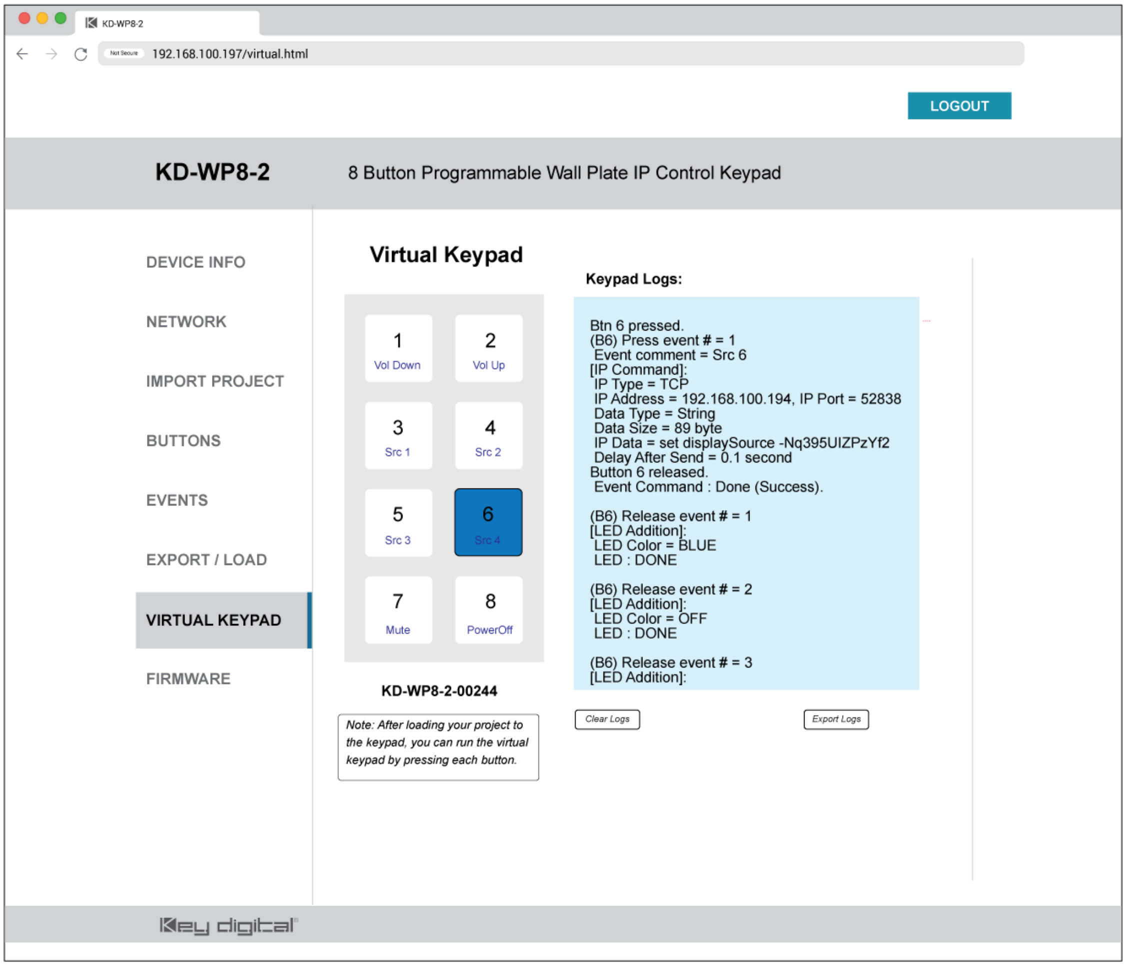
Note: After loading your project to the keypad, you can run the virtual keypad by pressing each button.

Keypad Logs:

```
Btn 1 pressed.
(B1) Press event # = 1
 Event comment = Volume Down
[IP Command]:
 IP Type = TCP
 IP Address = 192.168.100.194, IP Port =
52838
 Data Type = String
 Data Size = 46 byte
 IP Data = set roomVolumeInc -1
apiKey:ZBK2uGJ7
 Delay After Send = 0.1 second
Button 1 released.
 Event Command : Done (Success).
```

Figure 52: Use the Virtual Keypad to see the command log and to virtually press buttons on the keypad.

Next, press Button 6 on the keypad to select input 4 on the display. The log will show the events associated with the button press and the `set displaySource` command. As configured, Button 6 changes the video source and illuminate blue once pressed. If the display is powered off when a source is selected, Mira Connect will automatically power on the display and then choose the desired source.



```
Btn 6 pressed.
(B6) Press event # = 1
 Event comment = Src 6
[IP Command]:
 IP Type = TCP
 IP Address = 192.168.100.194, IP Port = 52838
 Data Type = String
 Data Size = 89 byte
 IP Data = set displaySource -Nq395UIZPzYf2
 Delay After Send = 0.1 second
Button 6 released.
 Event Command : Done (Success).

(B6) Release event # = 1
[LED Addition]:
 LED Color = BLUE
 LED : DONE

(B6) Release event # = 2
[LED Addition]:
 LED Color = OFF
 LED : DONE

(B6) Release event # = 3
[LED Addition]:
```

Note: After loading your project to the keypad, you can run the virtual keypad by pressing each button.

Figure 53: Testing Button 6 on the keypad.

> NOTE: You can also confirm the keys are functional by using the Mira Connect Remote Control feature in your browser. As commands are sent to select a source, you will see that the display has been powered on and that the source has been set to Input 4.

Continue testing each key on the keypad and notice that the Mira Connect remote control UI updates in real time as volume adjustments are made, the microphones are muted or unmuted, source selections are changed, and finally, when the keypad's room power off button is pressed and held for 2 seconds.

The KeyDigital Virtual Keypad page allows you to press buttons on the KeyDigital web interface and see the results of button presses on the physical keypad — as well as on Mira Connect's remote control user interface.

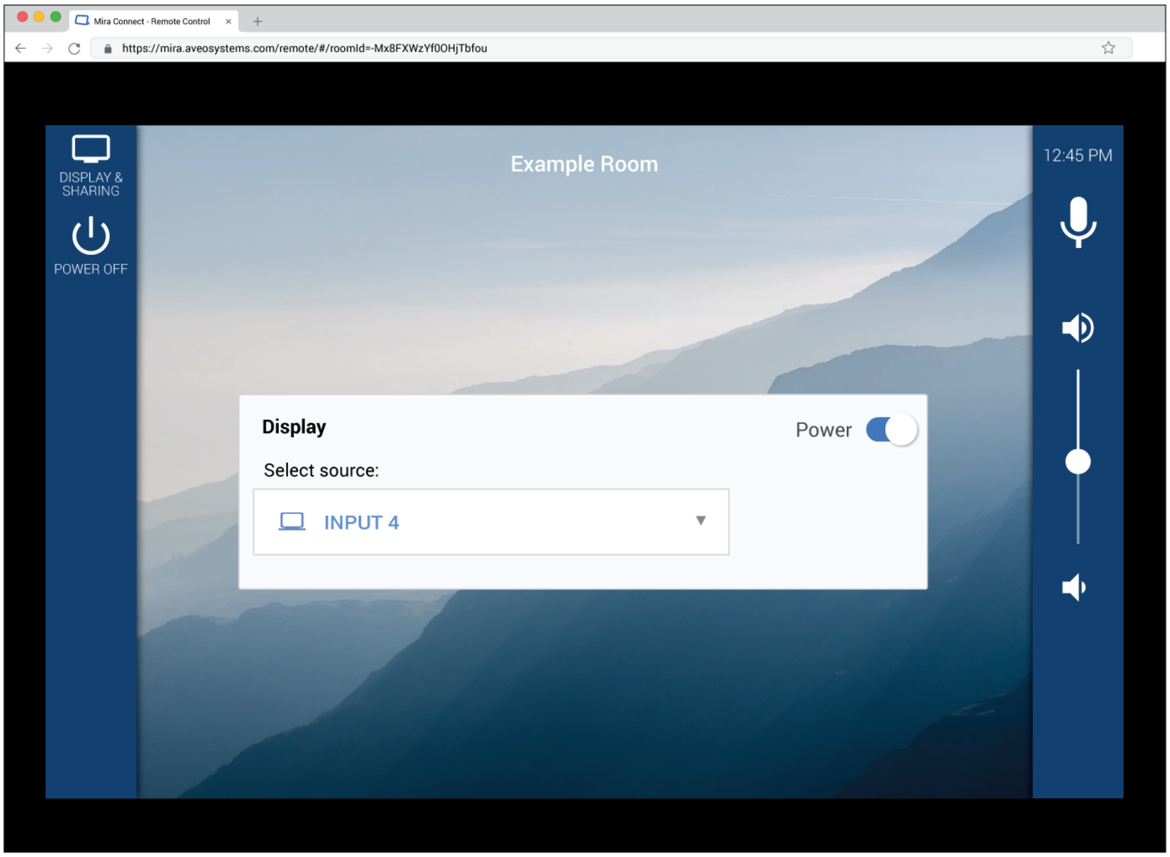

Mira Connect - Remote Control
https://mira.aveosystems.com/remote/#/roomId=-Mx8FXWzYf0OHjTbfou
DISPLAY & SHARING
POWER OFF
Example Room
12:45 PM
Display
Power
Select source:
INPUT 4

Figure 54: Mira Connect's Remote Control UI updates in real time as button presses are made on the keypad, allowing the room to be fully controlled from anywhere.

Step 10 — Troubleshooting the Keypad Configuration

If button presses are not affecting the room, look closely at the following items:

- Ensure the keypad configuration has been loaded into the keypad as described in the **Using the KeyDigital keypad** section. After configuring the keypad, It's easy to forget to *load the settings into the keypad*. Trust me on this one.
- If you are configuring multiple keypads, ensure you are working with the one you think you are working with by confirming the IP address of the keypad.
- Ensure you are using a valid API key with each command string.
  - Get the API key from the room's API menu in Mira Portal, as described in the **Create a Mira Connect API Key** section.
  - *Do not use the API key used in this document since that is specific to the room used to create this example and will not work with your room.*
- Ensure the event type for commands is set to **IP Command** and the IP Type is set to **TCP**.
- Ensure the IP address of the Mira Connect software used in each of the command strings is set correctly.
- Ensure the port for each command is set to 52838 and that TCP is enabled on the room in Mira Portal, as described in the **Create a Mira Connect API Key** section.

- Ensure the Data Type is set to **String**, as described in the **Configuring Button Events** section.
- Ensure each command includes the command terminator ”.

If these settings are correct, but there is still no communication with Mira Connect software, follow our step-by-step instructions for creating an inbound firewall rule on the PC that will allow incoming connections over port 52838. By default, the PC may be blocking incoming communication to port 52838 until an inbound firewall rule is created.

If mute and volume commands are functioning but video source selections are not working:

- Ensure you have the correct display ID for the display from the `subscribe roomCaps` command. The display ID is unique to your equipment and your room. If you have a different room or delete the current and add a new display, the display ID will change.
  - *Do not use the display ID and source IDs from this example as they are specific to the room used to create this example.*
- Ensure you are using the correct source IDs for the sources from the `subscribe roomCaps` command. The source IDs are unique to this display. If you delete and add new sources in Mira Portal, the source IDs will change.

If button LEDs are not lighting as expected, confirm the LED selections and events in the Button and Event steps.

Conclusion

Mira Connect is an AV control system that makes it easy to build professional AV control systems for a wide variety of AV applications. To learn more about Mira Connect, see our website AveoSystems.com.

In this guide, we showed how to integrate a KeyDigital 8-button keypad with Mira Connect software, assigning different functions to each button to create a specific user interface. We showed how to use different features of the KeyDigital keypad, including Press and Toggle features, Hold functions, and more.

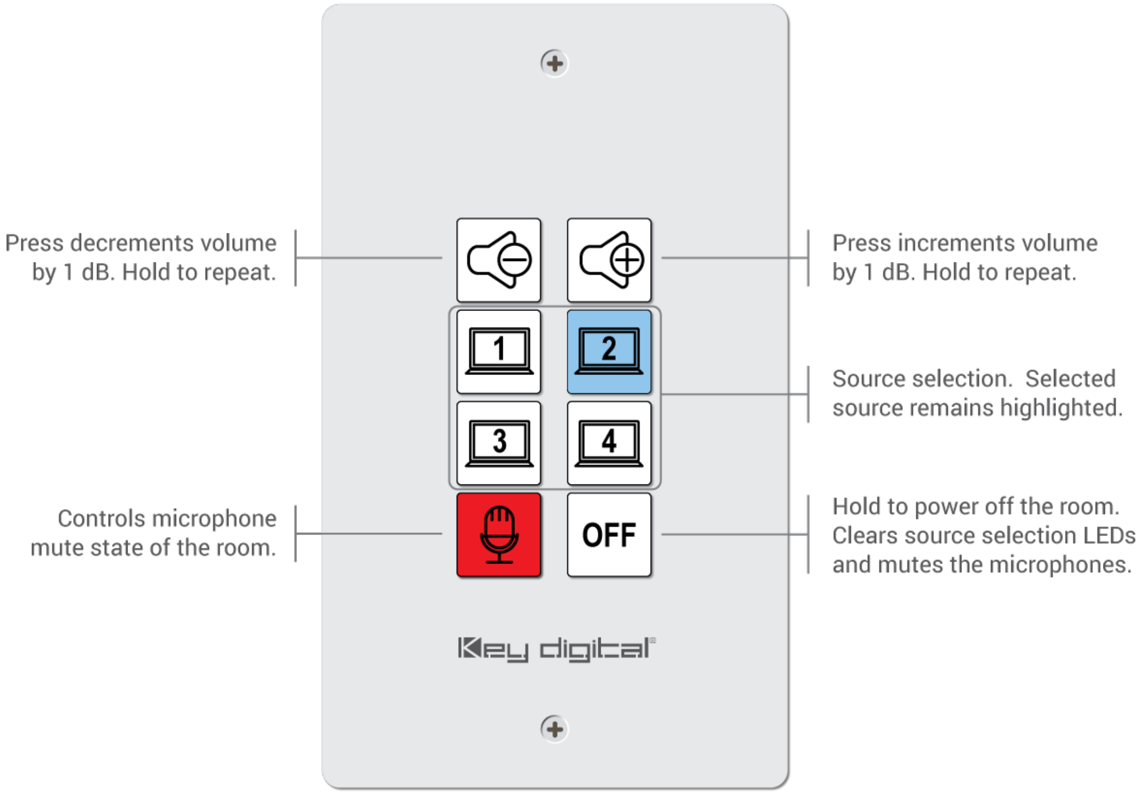

Press decrements volume by 1 dB. Hold to repeat.
Press increments volume by 1 dB. Hold to repeat.
1
2
3
4
Source selection. Selected source remains highlighted.
Controls microphone mute state of the room.
OFF
Hold to power off the room. Clears source selection LEDs and mutes the microphones.
Key digital

Figure 55: The button layout for the KeyDigital keypad design.

You can use this example as the basis for your own applications and customize it to meet your needs.

Mira Connect is the only control system that has an API for controlling a room with simple-to-use API commands. Any feature from Mira Connect's user interface can be controlled via the room control API.

Mira Connect's room control API provides tremendous power to programmers who want to embed AV control with their applications without having to spend time deciphering and debugging different APIs from each manufacturers' equipment in the system.

As shown in the following figure, Mira Connect software controls the underlying devices using their individual control syntax, allowing an application to send consistent and simple commands to control the room regardless of the underlying equipment. Mira Connect also supports advanced features, including connecting to equipment using SSH, HTTPS, HTTP, UDP, and using Wake-on-LAN for powering on displays. Regardless of how Mira Connect connects to and controls equipment, your application only has to configure a TCP connection to Mira Connect, allowing your application to easily control devices that it never could before.

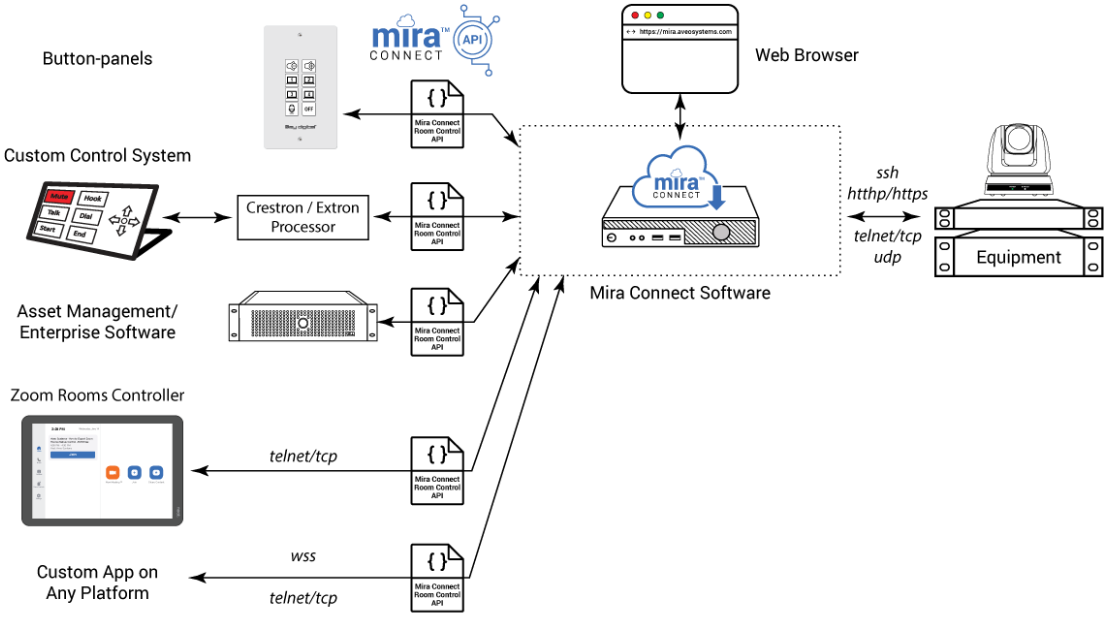


Figure 56: Mira Connect room control API enables new applications of Mira Connect.

Using the room control API and similar steps, you can configure this and other network keypads, along with other third-party devices to control anything you'd like from Mira Connect. Using the power of Mira Connect software, you can easily control anything that Mira Connect touch panels can control and create your own user interface.

Want to learn more?

Contact us if you have an application for Mira Connect and would like to get more information.

To learn how to use the Mira Connect Room Control API, see our detailed Room Control API guide.