visual engineering
LIGHTWARE

Shure P300 DSP Driver

for

LARA

TABLE OF CONTENTS

1. Introduction

This document was created to help integrators understand and configure the Shure P300 DSP driver in LARA.

The system's main components are:

Lightware UCX/MMX2 Universal Matrix Switcher,

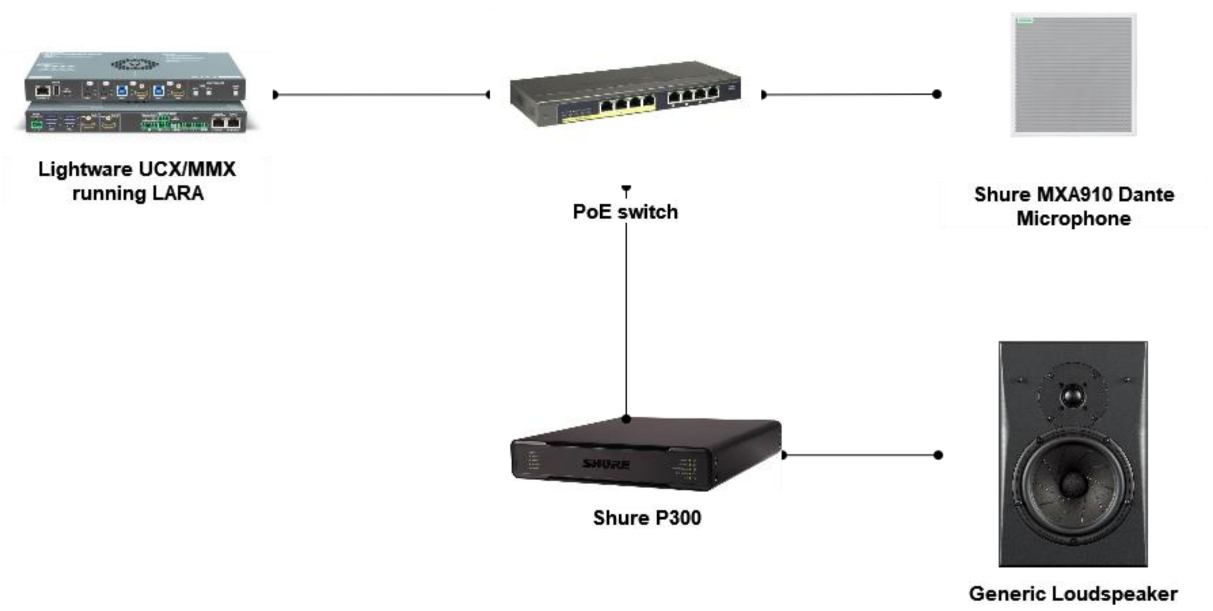
Shure P300 DSP

The room automation is made by Lightware Advanced Room Automation (LARA). It is a modular system which includes drivers for the equipment applied in the system. In this case, a P300 DSP driver, a logic module, for connecting the drivers, and a userpanel modul, for debugging, and display purposes.

2. Solution overview

Shure P300 DSP driver provides the ability for integrators to easily control their DSP installed in a conference room application, where Lightware's UCX/MMX2 universal matrix switcher is handling room control functions.

In a typical installation, a PoE gigabit switch connects the UCX/MMX2 with the audio DSP, and the ceiling microphone. The DSP drives the loudspeaker in the room.



The main feature of this solution is to provide DSP control functions for integrators, to be able to use the same user interface for controlling video, shades, lights, as well as audio. Our solution provides the following control functions:

- getAll: queries all parameters of the DSP
- getHookstatus: queries, if the DSP USB connection is in use, by a conference application, ONHOOK means the device is not in use, OFFHOOK means, that the DSP is in use as a USB audio device
- getUSBstatus: queries the connection status of the USB port on the DSP
- getErrorevent: queries the last recorded error log from the DSP

- getIPaddress: queries the IP address of the device
- getFWversion: queries device firmware version
- getPresetName: queries the name of the preset, that is given by the PresetID in the query message e.g.: getPresetName(2); response: Preset name: {name of preset number 2}
- getPresetID: queries the active preset’s ID
- automxrMuteToggle: toggles the mute state of the automixer output (Ch21) (The automixer mute toggle status is represented by the microphone LED)
- recallPreset: recalls the preset given by the preset ID (1-10)
- setMute: sets Mute status ON/OFF for selected channel
- toggleMute: toggles Mute status of selected channel
- decLevel: decreases the level/gain of selected channel, decrement amount can be specified per command, in dB value
- incLevel: increases the level/gain of selected channel, increment amount can be specified per command, in dB value
- getLevel: queries the level of the selected channel, given in the getLevel command, returns level in dB
- setLevel: sets the level of the selected channel, given in the setLevel command, available values: -110dB - +30 dB

3. Installation and Configuration

3.1. Installation

Minimum equipment requirement for the solution:

- Lightware UCX / MMX2 Universal Matrix Switcher
- Shure P300 DSP
- Ethernet network switch
- Cables

It is necessary to set up your Shure DSP before implementing any LARA control solution. For unboxing, initial setup of the DSP, please refer to the product page, and the user manual:

Product landing page:

https://www.shure.com/en-US/products/mixers/p300?variant=P300-IMX

User manual:

https://pubs.shure.com/guide/P300/en-US?_gl=1*j77scu*_gcl_au*OTgxMjI5OTU2LjE2ODc3ODI5ODU.*_ga*MjQyNDExNDcwLjE2Nzg5NTE0OTg.*_ga_DB3CR9SF0C*MTY4OTE2NzI3Ny44LjEuMTY4OTE2NzI3OC41OS4wLjA.

From Shure Getting started manual:

3.2. Getting Started

To control the P300, use Shure Designer software to adjust settings and route audio between Shure devices. After completing this basic setup process, you should be able to:

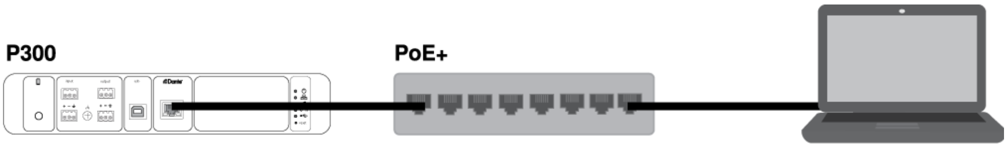
- Control the P300 in Designer
- Apply digital signal processing (DSP)
- Route audio to and from devices

Before you get started, you'll need:

- Cat5e (or better) Ethernet cable
- Network switch that provides Power over Ethernet Plus (PoE+)
- Shure Designer software installed on a computer. Download at www.shure.com.

3.3. Step 1: Connect to a Network and Discover in Designer

1. Connect the P300 to a PoE+ port on the network switch using Cat5e (or better) cable.
2. Connect your computer running Designer to the network switch.
3. Open Designer, and check that you’re connected to the correct network in Settings.
4. Click Online devices. A list of online devices appears.
5. To identify devices, click the product icon to flash the lights on a device.



3.4. Step 2: Route Audio and Apply DSP

The easiest way to route audio and apply DSP is with Designer's Optimize workflow. Optimize automatically routes audio signals, applies DSP settings, turns on mute synchronization, and enables LED logic control for connected devices.

For this example, we’ll connect an MXA910 ceiling array microphone. The process is the same for any networked Shure microphone in Designer.

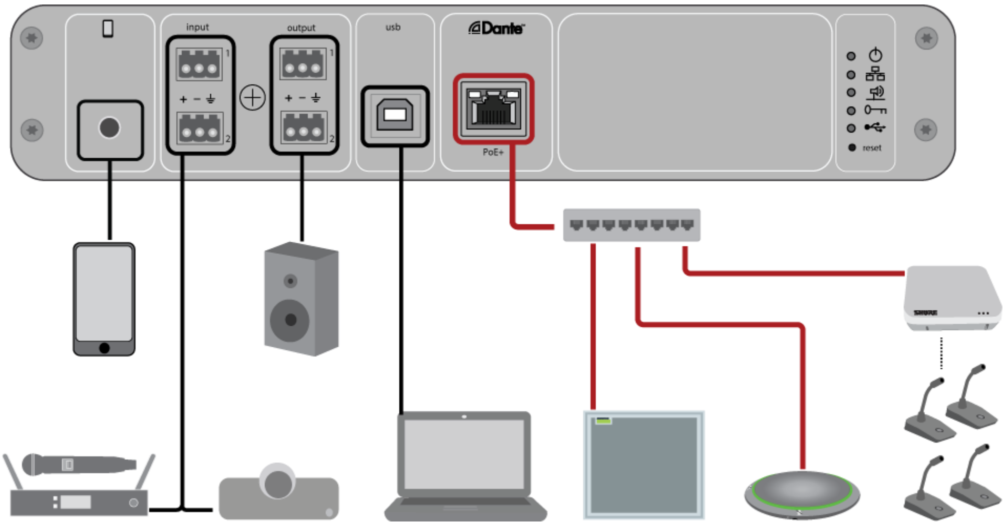
1. Go to `My projects` > `New` to create a new project.
2. Select `New` > `Room (live)` to create a new room. Any online devices appear in the list. Drag and drop the P300 and MXA910 to add them to your room.
3. Select Optimize.
4. Check the audio routes and settings to make sure they fit your needs. You might need to:
    - Delete unnecessary routes.
    - Fine-tune DSP blocks as needed.
    - Test and adjust microphone coverage.

You can also route audio manually in Designer outside of the Optimize workflow, or use Dante Controller.

3.5. Step 3: Adjust DSP and Route Signals

The final steps vary depending on what other devices you connect to the P300. Regardless of what devices you connect, the final steps are to check DSP settings and route signals in the matrix mixer.

1. Connect any other analog, USB, or mobile audio devices to the P300.



2. Check the signal routes in the Matrix mixer tab. Designer establishes many common routes automatically.
3. Adjust levels as needed in the Inputs and Outputs tabs.
4. Listen to your signal and adjust DSP settings as needed. Optimize applies DSP automatically, so you may not need to make any adjustments.
5. Check and adjust the acoustic echo cancellation (AEC) reference source in `Schematic` > `AEC`. Optimize sets one automatically, but you may need to adjust it depending on your setup.

   Use the channel that carries audio to loudspeakers as the AEC reference. If your room has an analog loudspeaker system or built-in display speakers, Analog -- To Speaker is the most common source. If you're using Dante loudspeakers, one of Dante outputs is the reference.

After you finish the initial setup of the device, now you can start your LARA configuration.

Important Notice!

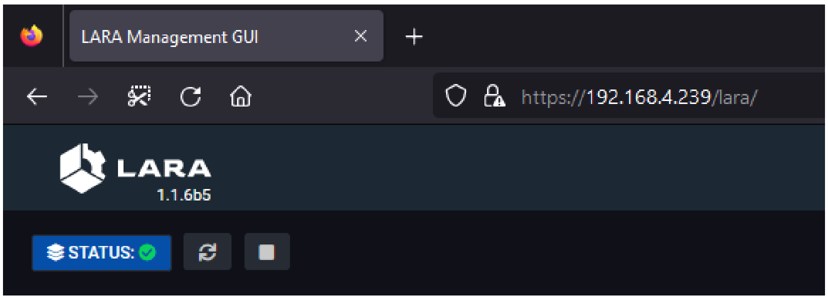
Before going to the next step, please visit https://lightware.com/lara for further information on how to enable, and start LARA service inside your UCX/MMX2.

3.6. Configuration

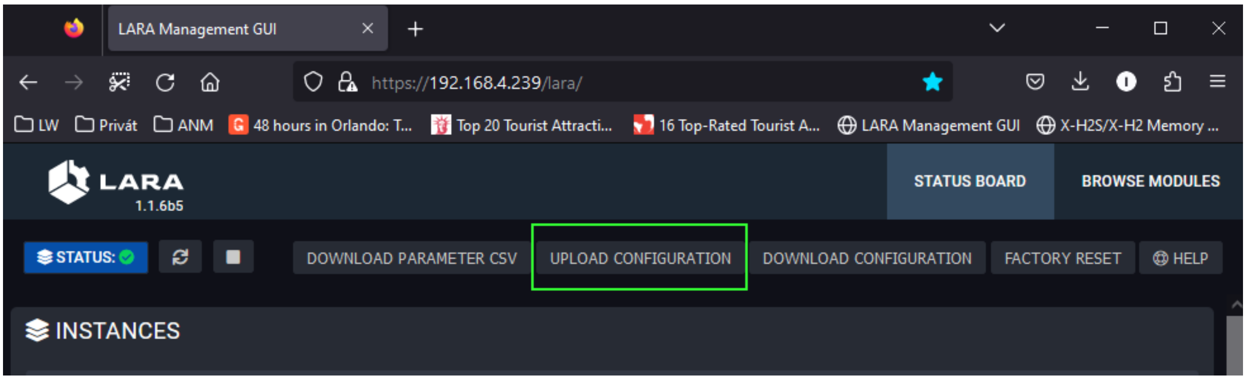
This section assumes that you are familiar with your Shure P300, and Lightware Taurus UCX, MMX2 devices, and you know their IP address and you have enabled LARA functionality. Further, you have your computer connected to the same IP subnet, where your devices located.

3.6.1. Starting LARA, Uploading Configuration

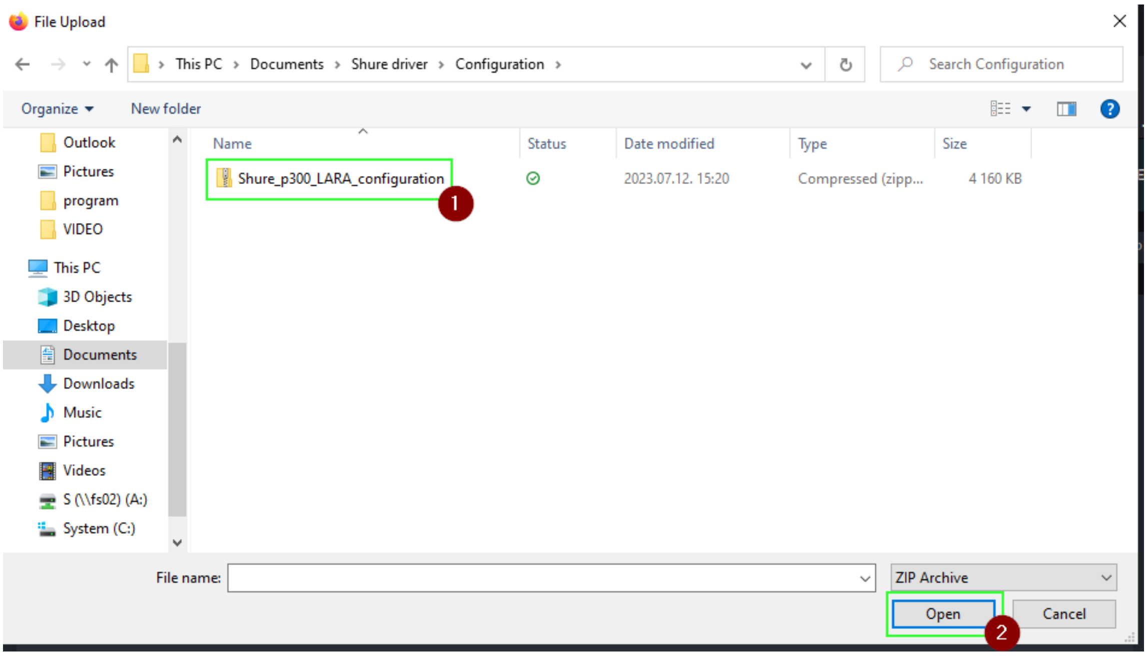
Start a new browser window and start LARA.



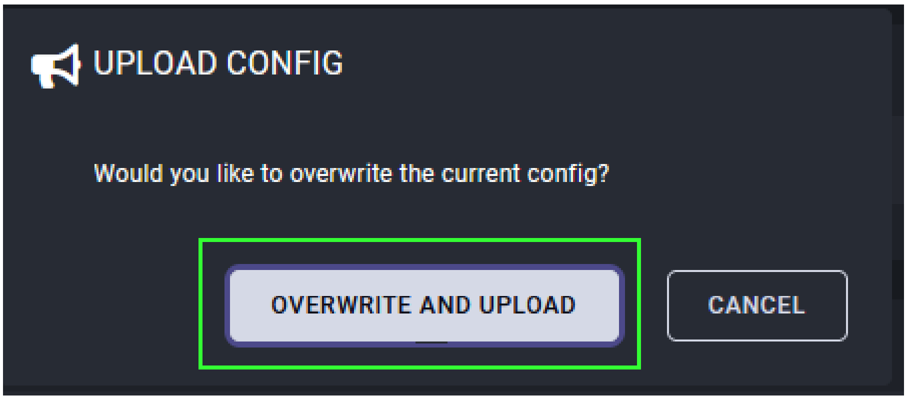
Choose: UPLOAD CONFIGURATION



Browse and open your configuration file, received from Lightware.



When prompted, choose to overwrite and upload

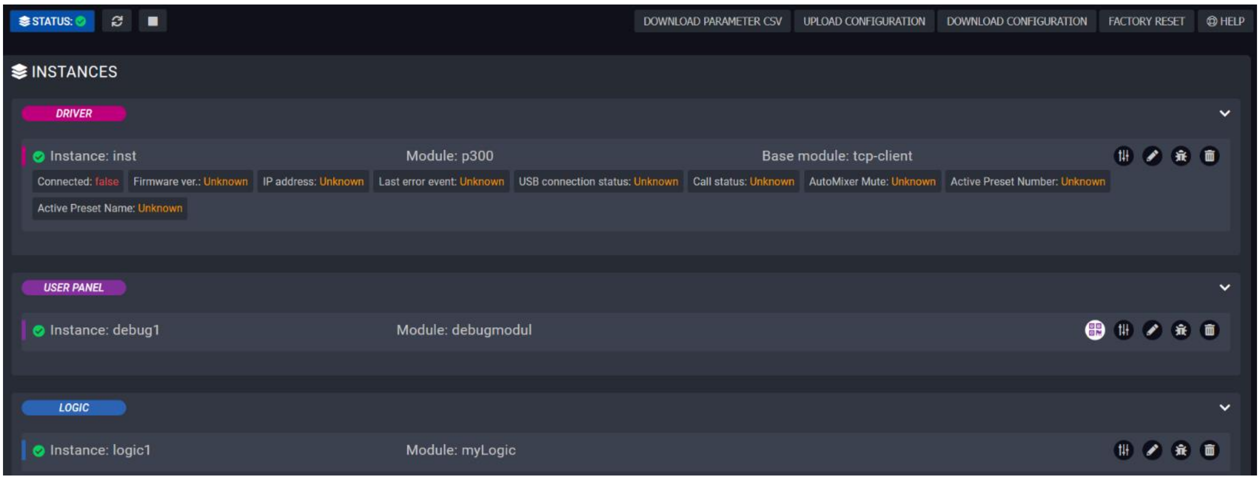


UPLOAD CONFIG

Would you like to overwrite the current config?

3.6.2. Status bar information

After uploading and starting the configuration the status board displays the actual state of the configuration:



Instance: inst Module: p300 Base module: tcp-client

Connected: false | Firmware ver.: Unknown | IP address: Unknown | Last error event: Unknown | USB connection status: Unknown | Call status: Unknown | AutoMixer Mute: Unknown | Active Preset Number: Unknown | Active Preset Name: Unknown

Instance: debug1 Module: debugmodul

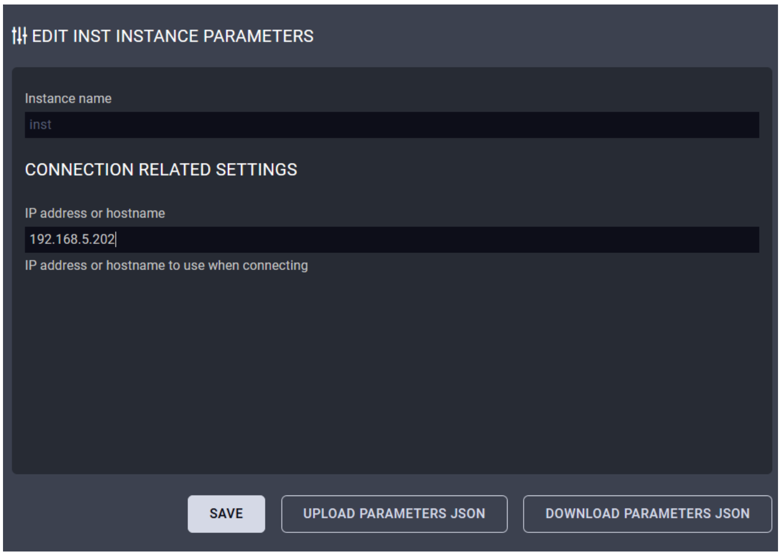
Instance: logic1 Module: myLogic

In case of the P300 module, you must specify the IP address of the DSP, to be able to communicate with the device: (please note, that 192.168.5.202 is just used in my case, the IP address should be your individual DSP’s IP address!) Setting the IP address in detail can be found below in paragraph 3.6.3.

The parameter setting can be done with opening each driver’s parameter section with this icon:

You do not need to setup anything in the logic and userpanel modules.

3.6.3. Setting parameters of “inst” instance:



EDIT INST INSTANCE PARAMETERS

Instance name: inst

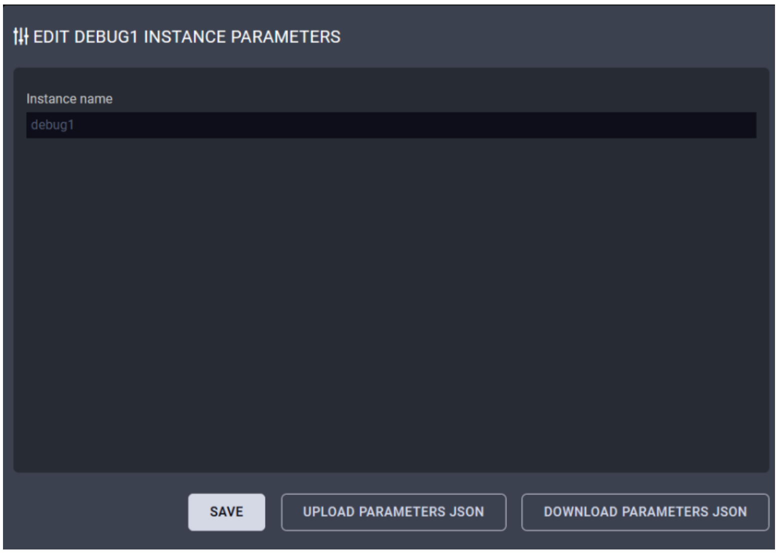
CONNECTION RELATED SETTINGS

IP address or hostname: 192.168.5.202

IP address or hostname to use when connecting

Enter the IP address of Shure P300 DSP. The name field is filled when creating the instance. In case of pre-loaded configuration, it cannot be changed.

3.6.4. Setting parameters of “debug1” instance:

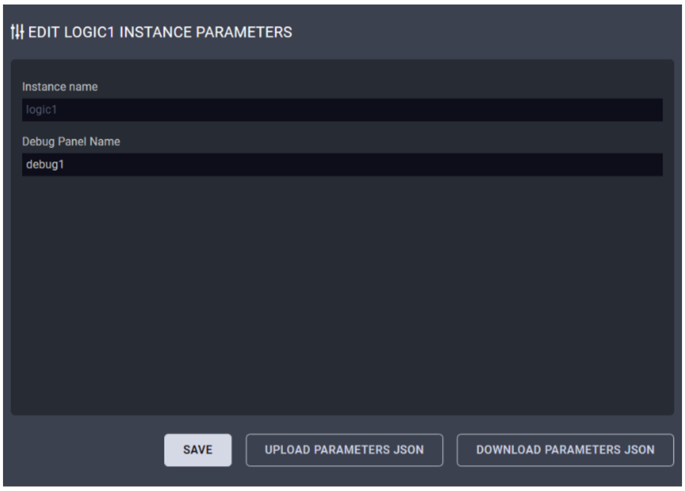


EDIT DEBUG1 INSTANCE PARAMETERS

Instance name: debug1

Only the name field is available, and only when creating the instance. No further setup needed.

3.6.5. Setting parameters of “logic1” instance:



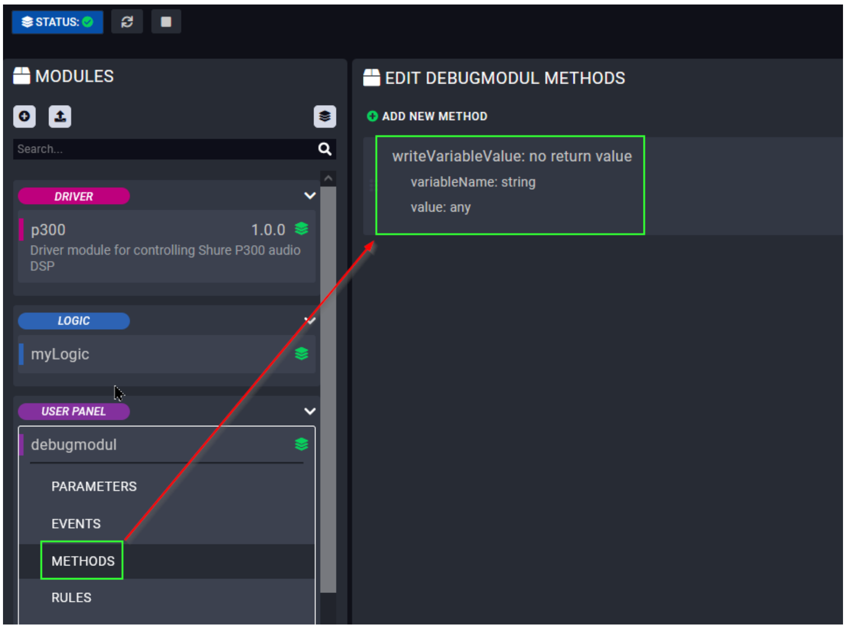
EDIT LOGIC1 INSTANCE PARAMETERS

Instance name: logic1

Debug Panel Name: debug1

Name field is filled when creating the instance from scratch. No need to modify, and it is not available either for modification.

The debug panel name field is important to fill, and the name of your debug panel instance must be given here. This name is used in the logic module, to update the parameters in the debug panel, with the built in “writeVariableValue” method inside the debug panel.

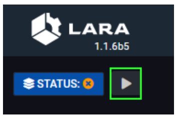


EDIT DEBUGMODUL METHODS

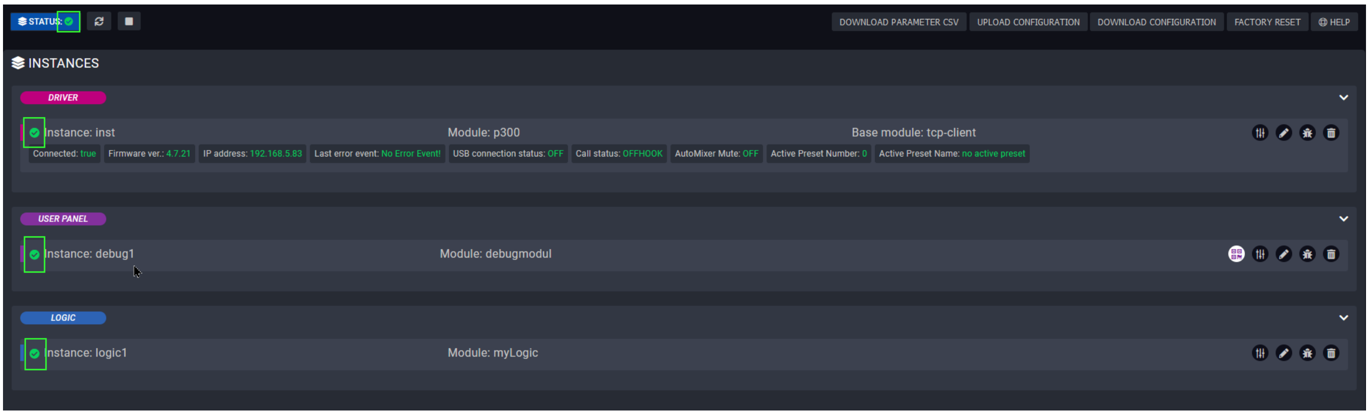
writeVariableValue: no return value
variableName: string
value: any

3.6.6. Finalizing the parameter settings:

After setting all the necessary parameters, please press start:



If the configuration runs, and the IP address settings are properly done all the status signs should be green on the left side of each instance’s status section:



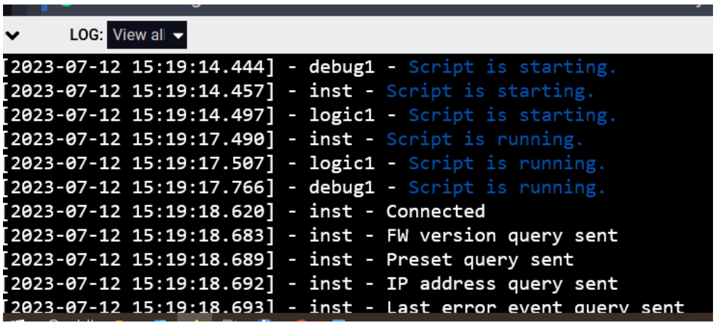
Instance: inst Module: p300 Base module: tcp-client

Connected: true | Firmware ver.: 4.7.21 | IP address: 192.168.5.83 | Last error event: No Error Event! | USB connection status: OFF | Call status: OFFHOOK | AutoMixer Mute: OFF | Active Preset Number: 0 | Active Preset Name: no active preset

Instance: debug1 Module: debugmodul

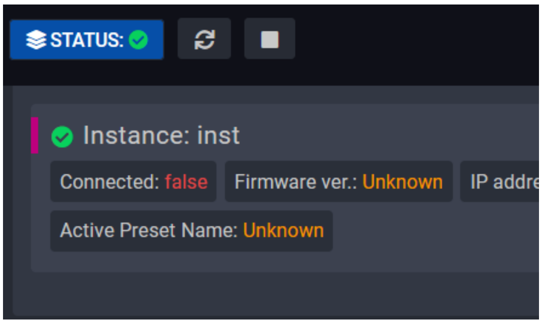
Instance: logic1 Module: myLogic

If everything is running, and there is not any error, you should see the following in the console log view:



```
[2023-07-12 15:19:14.444] - debug1 - Script is starting.
[2023-07-12 15:19:14.457] - inst - Script is starting.
[2023-07-12 15:19:14.497] - logic1 - Script is starting.
[2023-07-12 15:19:17.490] - inst - Script is running.
[2023-07-12 15:19:17.507] - logic1 - Script is running.
[2023-07-12 15:19:17.766] - debug1 - Script is running.
[2023-07-12 15:19:18.620] - inst - Connected
[2023-07-12 15:19:18.683] - inst - FW version query sent
[2023-07-12 15:19:18.689] - inst - Preset query sent
[2023-07-12 15:19:18.692] - inst - IP address query sent
[2023-07-12 15:19:18.693] - inst - Last error event query sent
```

If there is something written in red, there is a problem, try checking the IP address, and connections to the hardware, and try again. If the configuration is OK, but the DSP is not connected to LARA, you should see the following in the status board:



Instance: inst

Connected: false | Firmware ver.: Unknown | IP addre

Active Preset Name: Unknown

4. Operation

4.1. Using methods in the driver to control the DSP, or get information

Methods described in chapter 2 are available to query data from the DSP, or to set specific parameters. Typical use case, when your GUI designer created a nice touch control interface, where you can adjust levels, turn mutes on or off, or recall presets for different use cases of the room. These buttons and sliders must be connected to the DSP somehow, and here comes LARA, with the DSP driver. Selecting a mute button ID on your GUI, and telling LARA to engage "setMute" method as a reaction for the button press enables you to control mute parameter inside the DSP. The same applies for the other methods as well.

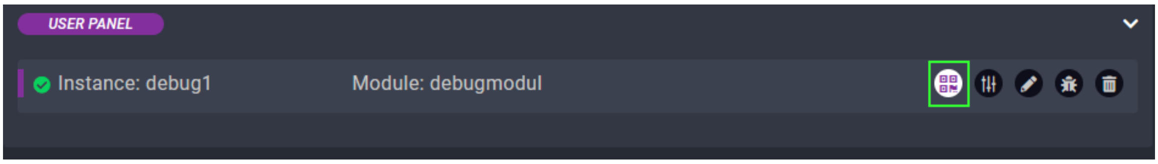
In case of "get" type methods, you can query different parameters from the DSP, and display it on your GUI, or trigger something with the data received by LARA, as a response for your query.

Designing a GUI for your application is out of Lightware's scope, but after you have your GUI, LARA and the methods enables you to make the connection between your GUI and the DSP.

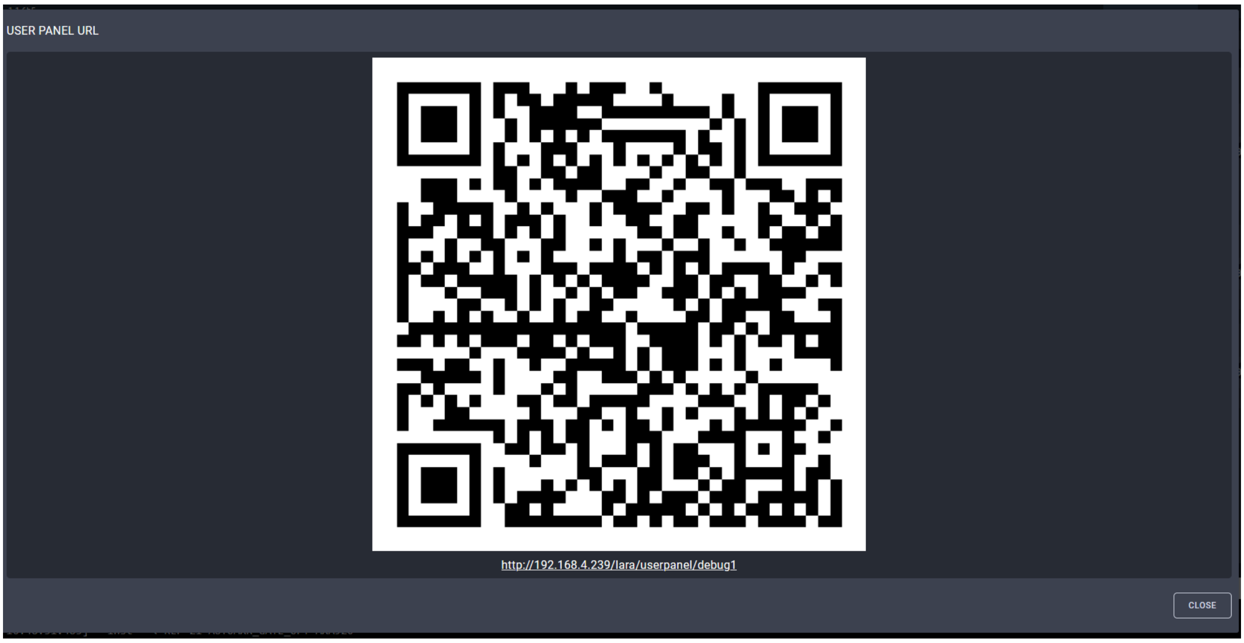
From your pre-designed UI, according to "HTML button press", or any HTML status change event, you can launch a desired method, to change parameters in the DSP.

4.2. Using the "debugmodule" user panel

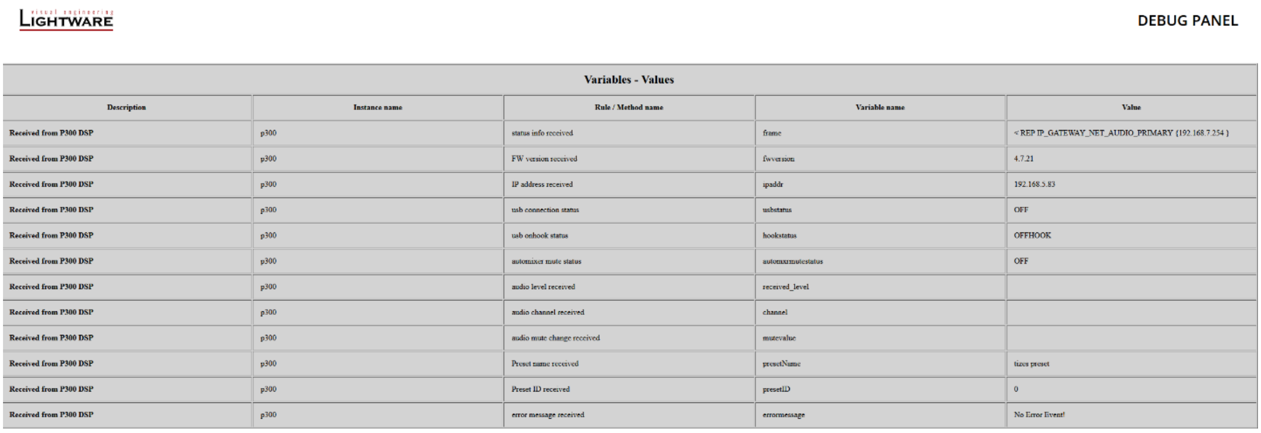
You can access the "debug panel" by clicking on the QR code icon in the debug1 instance field:



Clicking the icon opens the QR code, accompanied with the local network URL:



Clicking the URL opens the debugpanel in your default browser:



DEBUG PANEL

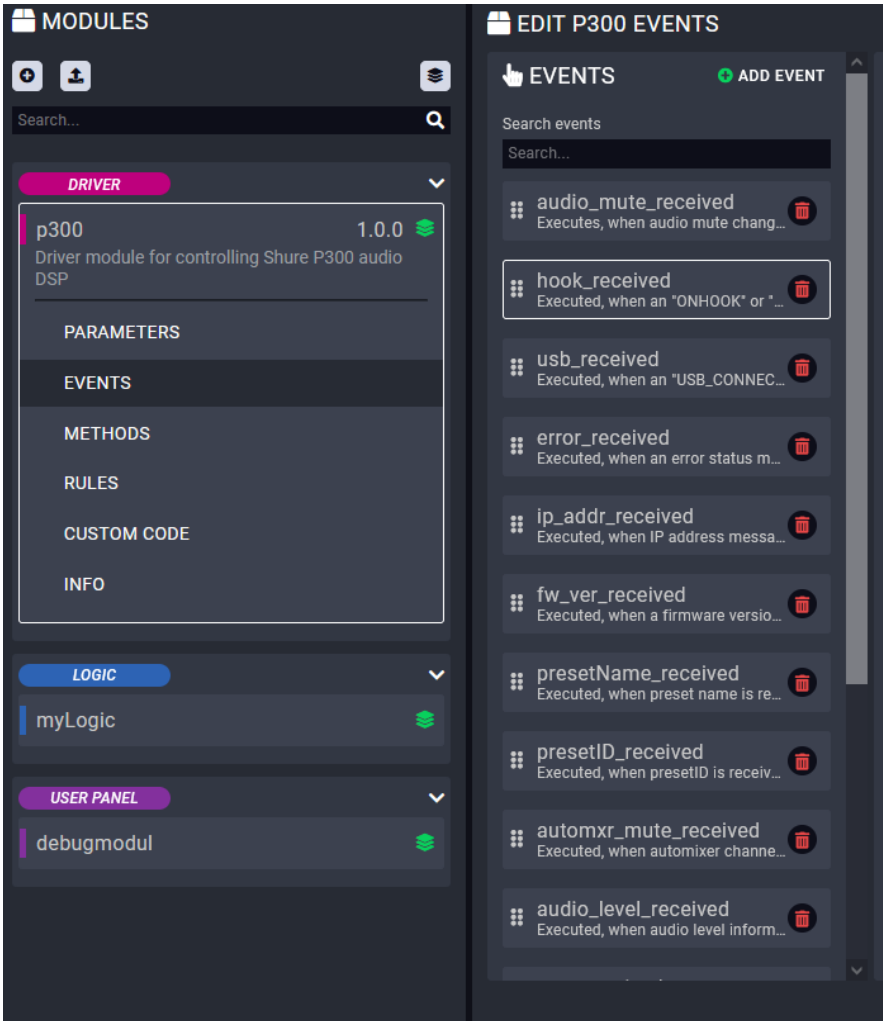
Variables - Values				
Description	Instance name	Rule / Method name	Variable name	Value
Received from P300 DSP	p300	status info received	frame	< REP IP_GATEWAY_NET_AUDIO_PRIMARY {192.168.7.254 }
Received from P300 DSP	p300	FW version received	fwversion	4.7.21
Received from P300 DSP	p300	IP address received	ipaddr	192.168.5.83
Received from P300 DSP	p300	usb connection status	usbstatus	OFF
Received from P300 DSP	p300	usb onhook status	hookstatus	OFFHOOK
Received from P300 DSP	p300	automixer mute status	automxrmutestatus	OFF
Received from P300 DSP	p300	audio level received	received_level	
Received from P300 DSP	p300	audio channel received	channel	
Received from P300 DSP	p300	audio mute change received	mutevalue	
Received from P300 DSP	p300	Preset name received	presetName	tizes preset
Received from P300 DSP	p300	Preset ID received	presetID	0
Received from P300 DSP	p300	error message received	errormessage	No Error Event!

Scanning the QR code also works, but only if your scanning device (phone/tablet) is connected to the same LAN.

This debug panel is a tool for integrators, to follow what is happening inside the DSP, see incoming commands, and variable values.

4.3. Using the Events

In the P300 module, you can find events created for your kind use.



audio_mute_received
Executes, when audio mute chang...

hook_received
Executed, when an "ONHOOK" or "...

usb_received
Executed, when an "USB_CONNEC...

error_received
Executed, when an error status m...

ip_addr_received
Executed, when IP address messa...

fw_ver_received
Executed, when a firmware versio...

presetName_received
Executed, when preset name is re...

presetID_received
Executed, when presetID is receiv...

automxr_mute_received
Executed, when automixer channe...

audio_level_received
Executed, when audio level inform...

These events are created, to be able to use the algorithm behind the module. Any time the DSP sends a report to LARA, the received frame is processed, and the corresponding event launches. This way, using the events, you can define objects on your UI, to illuminate, or change state, like a preset display, or a volume slider.

4.4. Logic module

You can find several pre-defined rules in the logic module. These rules are communicating with the debug panel, sending information to the page, for display. Thus, these rules should not be deleted, or modified.

5. Appendix

5.1. Document history

Rev.	Release date	Changes	Editor
1.0.0	July 17,2023	Initial version	István Kozma
1.1.0	January 22, 2023	LARA 1.2.0 compatibility	István Kozma