Biamp TesiraFORTÉ DSP driver for LARA

v1.0.1

TABLE OF CONTENTS

1. Introduction

The scope of this document is to help integrators understand and configure the Biamp Tesira driver for LARA. The driver supports Tesira control over Telnet (TCP/IP).

Firmware and software versions used and tested during the development:

- Biamp Tesira Designer Software v4.5.0.23124
- Biamp Tesira Firmware v4.5
- Lightware UCX firmware v2.9.0b6
- LARA version v1.2.0b36

The downloadable package contains:

- Tesira driver module: Biamp_TesiraFORTE_driver_module_v1.0.1.zip
- Sample configuration: Sample_config_Tesira_control_v1.0.1.zip
- Documentation: LARA_Biamp_TesiraFORTE_DSP_driver_module_v1.0.1.pdf
- TesiraFORTE X 400 config file: sample_config_FORTEX400.tmf

1.1. References

Lightware LARA Users Manual

Lightware UCX series Users Manual

Biamp Tesira Text Protocol 3.0

Biamp Tesira Text Protocol Cornerstone

Biamp Tesira Command String Calculator

2. Solution Overview

Biamp Tesira DSP driver module for LARA provides the ability to control selected parameters of the Tesira DSP installed beside a Lightware Universal Matrix Switcher (either UCX or MMX2).

In order to have a working solution the UCX/MMX2 has to be connected to the Tesira DSP through a local network.

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. Functions provided by the driver module are introduced in the following chapters.

2.1. Module release notes

V1.0.1 – LARA 1.2 support added. Please note that the module is not backward compatible with earlier versions of LARA. It is recommended to use the version v1.0.0 module with LARA 1.1.x.

V1.0.0 – initial version

2.2. Events defined in the driver

Event	Description
TTP connected	The Tesira DSP is ready for receiving and executing commands
Subscribed parameter changed	Fires, when any subscribed parameter is changed
Parameter query response received	Fires, when a response is received for a get command. Response for getSerial and getFirmware does not activate the event.

There are other events in the driver, but they originate from the base module, so those are not listed here.

2.3. Methods Defined in the Driver

Methods are used to realize functions in the DSP, e.g. setting values, querying parameters, etc. The methods can be called from other modules e.g. Logic module, or User panel module. The implemented methods can be found below:

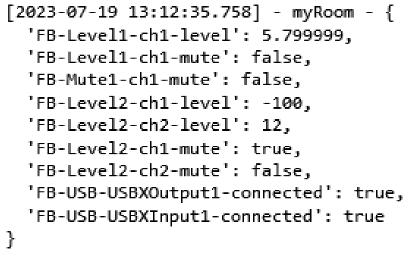
- recallPreset
- subscribeLevel
- subscribeMute
- subscribeUSBStatus
- setLevel
- incLevel
- decLevel
- setMute
- toggleMute
- getLevel
- getMute
- getSerial
- getFirmware
- unsubscribeLevel
- unsubscribeMute
- unsubscribeUSBStatus
- sendFrame

2.4. Subscribe Type Methods

These methods are used to subscribe to parameter changes of different processing blocks. When a subscribed parameter is changed, the "Subscribed parameter changed" event getting triggered and the unique name (custom label for feedback in Tesira) and the new value of the parameter is being stored/updated in a json object called dspData. The unique name is generated automatically, for details of the unique name, see the particular method's description.

With the event, the actual state of all the subscribed parameters are handed over with the dspData json object.

Below you can see the content of the json object logged to the console. For more detailes please check the sample configuration.



```
[2023-07-19 13:12:35.758] - myRoom - {
  'FB-Level1-ch1-level': 5.799999,
  'FB-Level1-ch1-mute': false,
  'FB-Mute1-ch1-mute': false,
  'FB-Level2-ch1-level': -100,
  'FB-Level2-ch2-level': 12,
  'FB-Level2-ch1-mute': true,
  'FB-Level2-ch2-mute': false,
  'FB-USB-USBXOutput1-connected': true,
  'FB-USB-USBXInput1-connected': true
}
```

2.4.1. subscribeUSBStatus

Subscribes to changes of the connection status of the given USB input/output block.

Parameter	Description
instanceTag	Instance tag of the USB input/output block
refreshRate	Time in milliseconds, describes how often the DSP will send updates

The update sent by the DSP is processed by the driver module and it stores the latest values in a json object. The property of each item is the unique name, what is calculated automatically according to the followings:

unique name: **FB-USB-**x**-connected**

where FB-USB- is fixed

x is the instance tag

-connected is fixed.

2.4.2. subscribeMute

Subscribe to changes of a mute button change.

Parameter	Description
instanceTag	Instance tag of the level/mute block
channel	Channel number
refreshRate	Time in milliseconds, describes how often the DSP will send updates

The unique name is calculated automatically according to the followings:

unique name: **FB-**x**-ch**y**-mute**

where FB- is fixed

x is the instance tag

-ch is fixed

y is the number of the channel inside the processing block

-mute is fixed.

2.4.3. subscribeLevel

Subscribe to changes of a level parameter.

Parameter	Description
instanceTag	Instance tag of the level block
channel	Channel number
refreshRate	Time in milliseconds, describes how often the DSP will send updates

The unique name is calculated automatically according to the followings:

unique name: **FB-**x**-ch**y**-level**

where FB- is fixed

x is the instance tag

y is the number of the channel inside the processing block

-level is fixed

2.5. Get Type Methods

These methods are used to query the actual value of a parameter. When the DSP sends its response to a get command, the "Parameter query response received" event getting triggered, and the unique name of the parameter and it's value are stored in variables (qrVariable, qrValue). Those variables are handed over as event parameters.

The unique name is generated automatically, for details of the unique name, see the particular method's description.

2.5.1. getMute

Queries the actual state of a mute button.

Parameter	Description
instanceTag	Instance tag of the level/mute block
channel	Channel number

The unique name is calculated automatically according to the followings:

unique name: **QR-**x**-ch**y**-mute**

where QR- is fixed

x is the instance tag

-ch is fixed

y is the number of the channel inside the processing block

-mute is fixed.

The value is a Boolean type, can be true or false.

2.5.2. getLevel

Queries the actual value of a level parameter.

Parameter	Description
instanceTag	Instance tag of the level block
channel	Channel number

The unique name is calculated automatically according to the followings:

unique name: **QR-**x**-ch**y**-level**

where QR- is fixed

x is the instance tag

-ch is fixed

y is the number of the channel inside the processing block

-level is fixed.

The value is a number in dB

2.5.3. getFirmware

Queries the firmware version of the connected Tesira DSP device. The driver automatically queries the firmware version at startup, and shows it on the Status Board. The method does not have any parameters.

2.5.4. getSerial

Queries the serial number of the connected Tesira DSP device. The driver automatically queries the serial number at startup, and shows it on the Status Board. The method does not have any parameters.

2.6. Set Type Methods

These methods are used to set parameters to an absolute value.

2.6.1. setLevel

Sets the level to an absolute value.

Parameter	Description
instanceTag	Instance tag of the level block
channel	Channel number
value	The required value of the level parameter in dB

2.6.2. setMute

Sets a mute button to on or off.

Parameter	Description
instanceTag	Instance tag of the level/mute block
channel	Channel number
value	The required state of the mute button (true for mute on, false for mute off)

2.7. Other Methods

2.7.1. recallPreset

Used for recalling presets set up in Tesira software by the preset’s ID number.

Parameter	Description
presetNumber	ID number of the preset, you want to recall

2.7.2. incLevel

Increases the value of a level parameter by the specified value.

Parameter	Description
instanceTag	Instance tag of the level block
channel	Channel number
value	Value in dB by which you want to increase the original value

2.7.3. decLevel

Decreases the value of a level parameter by the specified value.

Parameter	Description
instanceTag	Instance tag of the level block
channel	Channel number
value	Value in dB by which you want to decrease the original value

2.7.4. toggleMute

Toggles between the on and off state of a mute button.

Parameter	Description
instanceTag	Instance tag of the level/mute block
channel	Channel number

2.7.5. unsubscribeLevel

Stops the Tesira DSP device to send updates regarding the specified level parameter.

Parameter	Description
instanceTag	Instance tag of the level block
channel	Channel number

2.7.6. unsubscribeMute

Stops the Tesira DSP device to send updates regarding the specified mute parameter.

Parameter	Description
instanceTag	Instance tag of the level/mute block
channel	Channel number

2.7.7. unsubscribeUSBStatus

Stops the Tesira DSP device to send updates regarding the USB connection status.

Parameter	Description
instanceTag	Instance tag of the USB input or output block

2.7.8. sendFrame

Sends a custom command to the Tesira DSP device. With the help of this method it is possible to send commands to the DSP, which are not yet implemented in the driver.

Parameter	Description
message	Command string to send to the DSP

3. Installation and Configuration of the Driver

3.1. Prerequisites

In order to use the Tesira driver for LARA, you must have a configured Biamp DSP. To do that please refer to the user manual of the Biamp device/software.

There are settings that has to be set at Biamp side in order to let the control system work:

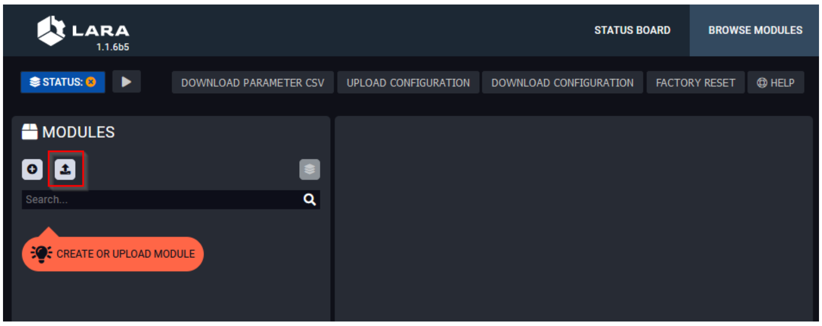
- In case of Telnet control, Telnet must be enabled in Tesira-Network Settings
- Tesira device must be unprotected
- Instance tags must not contain spaces for proper operation

3.2. Installation

This section assumes that you are familiar with your Biamp Tesira DSP, 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.

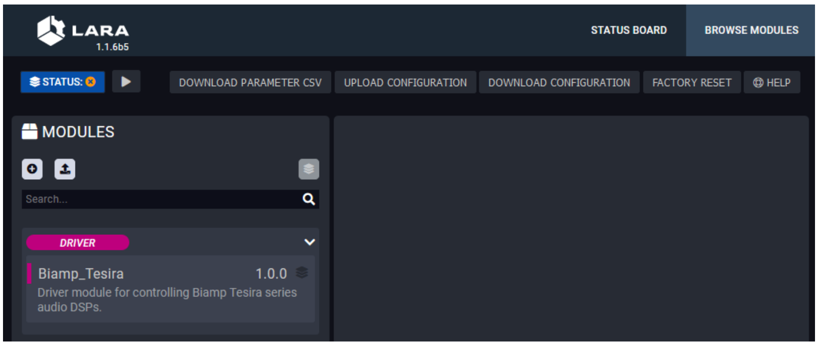
3.2.1. Uploading the Tesira driver module

You can upload the Tesira DSP driver module to a new/existing LARA configuration by clicking on the Upload module icon shown below.



Browse and open the module as a ZIP archive. You have to enter a name for the module.

When uploaded successfully it has to appear on the Browse Modules tab as it is shown below.



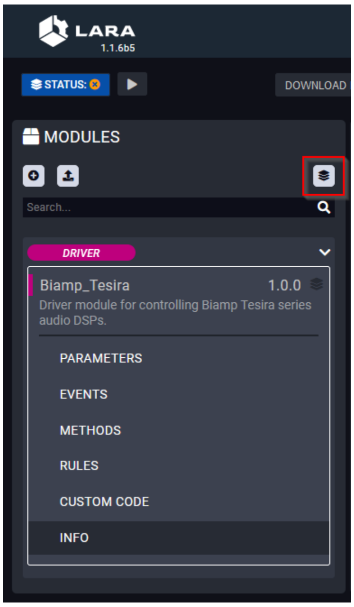
Biamp_Tesira 1.0.0

Driver module for controlling Biamp Tesira series audio DSPs.

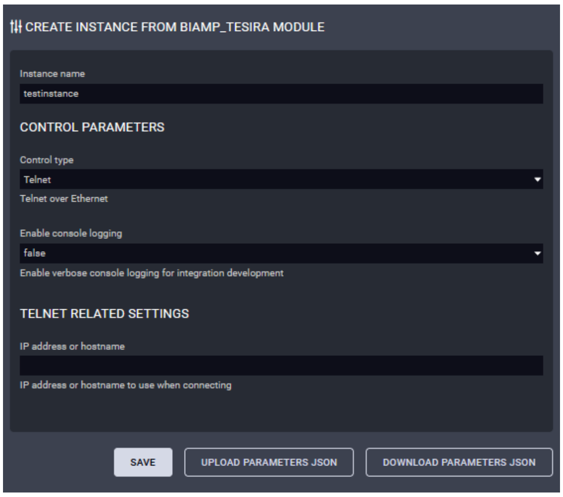
3.2.2. Create an instance for a particular Tesira device to control

In order to be able to communicate with a device, an instance has to be created from the module, what is called “Biamp_Tesira” in the example above.

To create an instance, you have to select the module and click on the ”Create new instance from the selected module” shown below.



Instances have to have a unique name, let's call it "testinstance". Configuration can be done at the time of creation, right from the window shown below, but those settings can be modified later as well. Please note that instance names cannot be modified after creation!


CREATE INSTANCE FROM BIAMP_TESIRA MODULE

Instance name: testinstance

CONTROL PARAMETERS

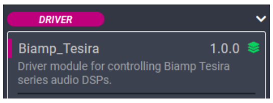
Control type: Telnet
Telnet over Ethernet

Enable console logging: false
Enable verbose console logging for integration development

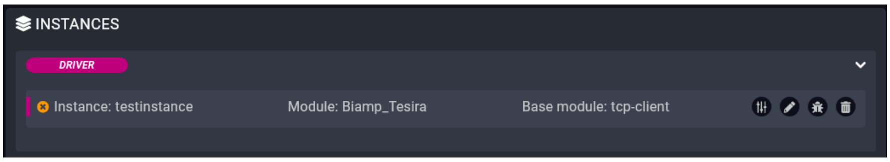
TELNET RELATED SETTINGS

IP address or hostname:
IP address or hostname to use when connecting

When the instance has been created successfully, it is indicated by the green instance indicator next to the module's name, on the Browse Modules tab.


DRIVER
Biamp_Tesira
1.0.0
Driver module for controlling Biamp Tesira series audio DSPs.

When an instance is created it appears on the Status Board as well, as it is shown below.

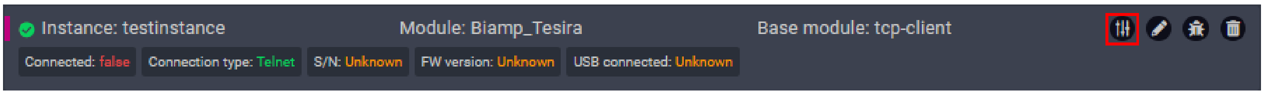


Instance: testinstance | Module: Biamp_Tesira | Base module: tcp-client

3.2.3. Configuring the instance

The instance has to be configured by its parameters. Most of the parameters can be selected using drop-down lists.

Parameters can be edited by clicking the “Edit instance parameters” button as shown below:



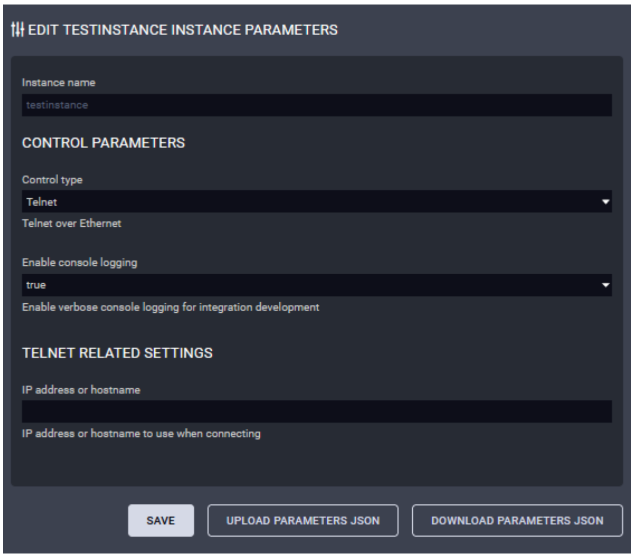
Instance: testinstance Module: Biamp_Tesira Base module: tcp-client

Connected: false | Connection type: Telnet | S/N: Unknown | FW version: Unknown | USB connected: Unknown

Editable parameters and its possible values can be find in the table below.

Parameter	Possible values	Description
CONTROL PARAMETERS		
Control type	Telnet	To control the DSP by Telnet over Ethernet
Enable console logging	false	Messages regarding the operation will not be logged to the console.
	true	Messages regarding the operation will be logged to the console.
TELNET RELATED SETTINGS		
IP address or hostname	x.x.x.x	IP address or hostname to use when connecting

The Edit instance parameters window is shown below.



EDIT TESTINSTANCE INSTANCE PARAMETERS

Instance name: testinstance

CONTROL PARAMETERS

Control type: Telnet
Telnet over Ethernet

Enable console logging: true
Enable verbose console logging for integration development

TELNET RELATED SETTINGS

IP address or hostname:
IP address or hostname to use when connecting

4. Sample Configuration

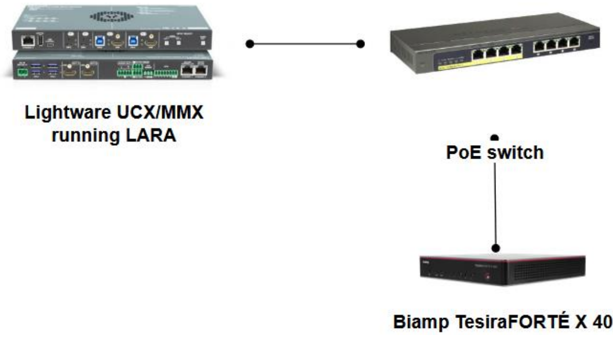
The downloaded package contains the driver module itself, and a sample configuration package. The sample configuration is provided to let you study how the driver module interacts with other modules in a configuration.

The sample configuration package contains the followings:

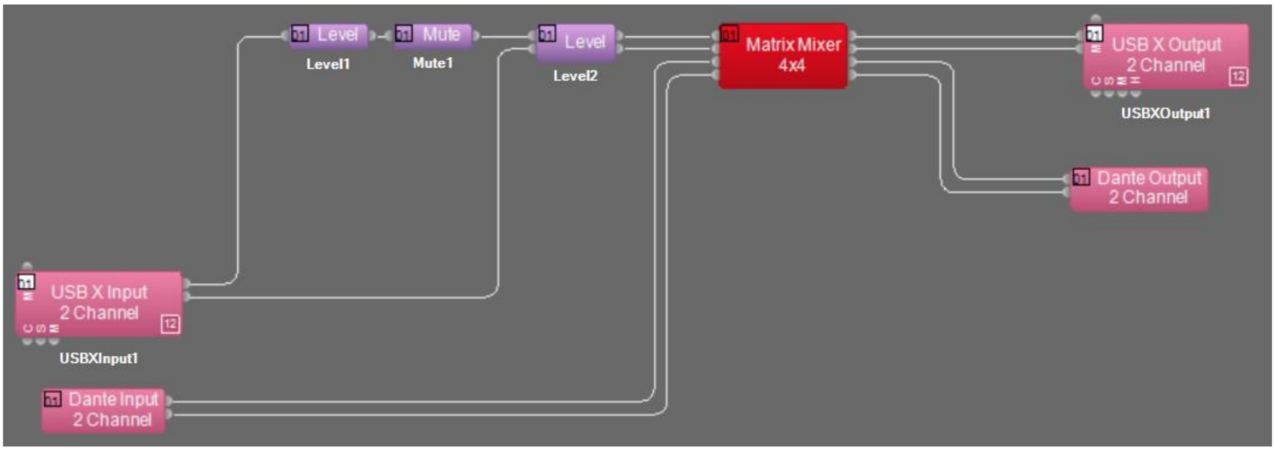
- Tesira configuration file (prepared for TesiraFORTÉ X 400)
- LARA configuration file

What you need to test the sample configuration:

- Lightware Taurus UCX/MMX2 series universal matrix switcher
- Biamp TesiraFORTÉ X 400



Instead of the X 400 you can use other TesiraFORTÉ DSP units, but you might have to recreate the Tesira configuration file for yourself. The screenshot below shows the design of the sample configuration, with the instance tags of the blocks used in the LARA configuration.



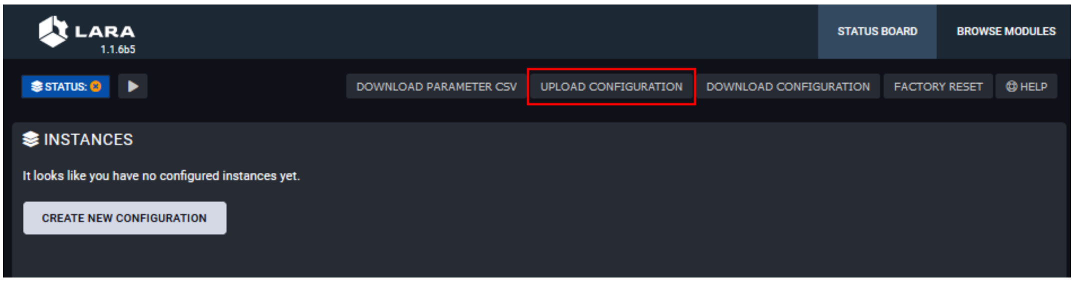
4.1. Using the Sample Configuration

4.1.1. Upload the sample configuration to the Tesira DSP

For the method, please refer to the Biamp documentation.

4.1.2. Upload the sample LARA configuration to the Lightware UCX/MMX2 device

Click upload configuration on your main LARA window. Browse and select the sample configuration file.



INSTANCES

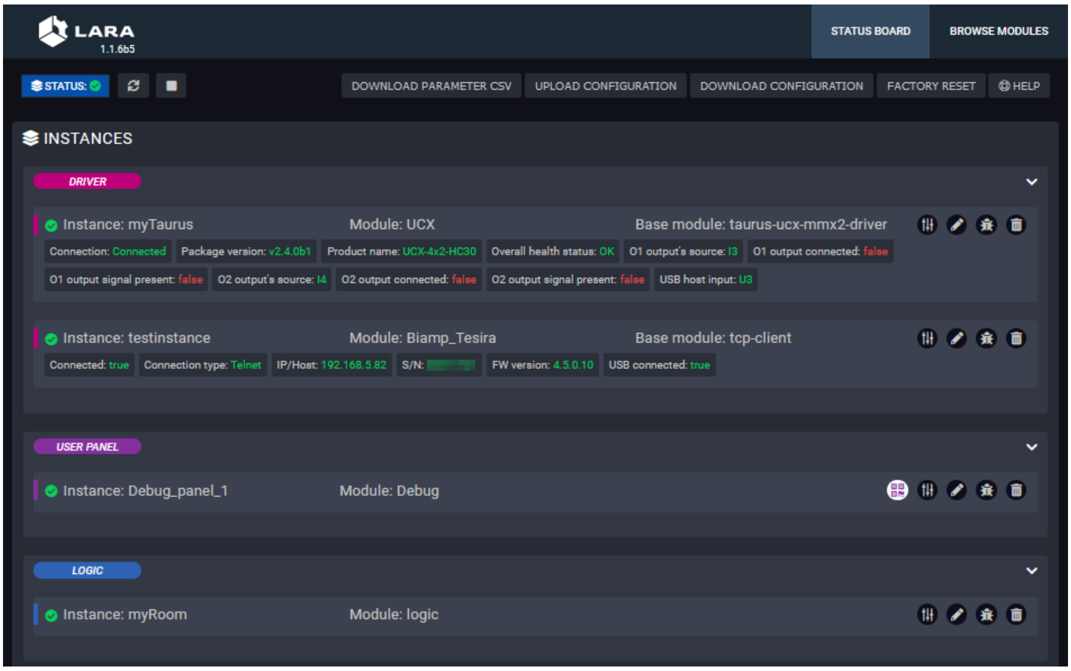
It looks like you have no configured instances yet.

CREATE NEW CONFIGURATION

4.1.3. Modify the IP address settings of testinstance to match your actual Tesira device

Refer to section 3.2.3.

After you’ve finished the configuration, you should see something like this:



INSTANCES

DRIVER

Instance: myTaurus Module: UCX Base module: taurus-ucx-mmx2-driver

Connection: Connected | Package version: v2.4.0b1 | Product name: UCX-4x2-HC30 | Overall health status: OK | O1 output's source: I3 | O1 output connected: false | O1 output signal present: false | O2 output's source: I4 | O2 output connected: false | O2 output signal present: false | USB host input: U3

Instance: testinstance Module: Biamp_Tesira Base module: tcp-client

Connected: true | Connection type: Telnet | IP/Host: 192.168.5.82 | S/N: [redacted] | FW version: 4.5.0.10 | USB connected: true

USER PANEL

Instance: Debug_panel_1 Module: Debug

LOGIC

Instance: myRoom Module: logic

4.1.4. Operation of the configuration

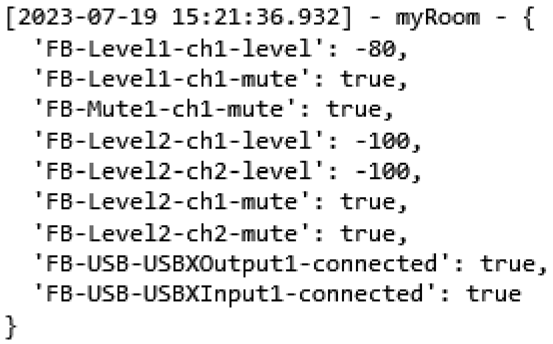
At startup the configuration subscribes to the following parameters:

Instance tag	Attribute	Channel nr.	Automatically generated unique name for feedback
Level1	level	1	FB-Level1-ch1-level
	mute	1	FB-Level1-ch1-mute
Mute1	mute	1	FB-Mute1-ch1-mute
Level2	level	1	FB-Level2-ch1-level
		2	FB-Level2-ch2-level
	mute	1	FB-Level2-ch1-mute
		2	FB-Level2-ch2-mute
USBXOutput1	connected	N/A	FB-USB-USBXOutput1-connected
USBXInput1	connected	N/A	FB-USB-USBXInput1-connected

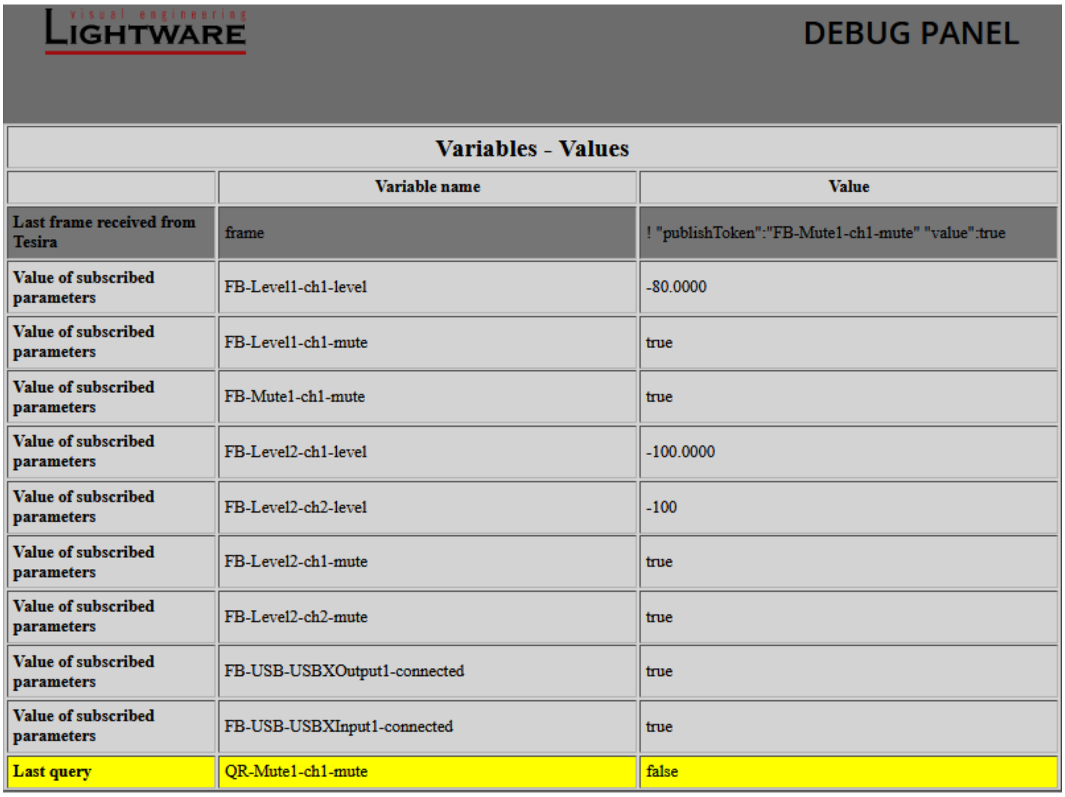
When there are changes to the parameters above in Tesira, the new value is stored in the json object named dspData. The property will be the automatically generated unique name and the value is the new value of the particular parameter.

In the Logic module there is a rule (Subscribed parameter changed) what is triggered when any subscribed parameter changed in Tesira. The rule logs the content of dspData to the console and simultaneously write the actual values to the debug panel.

The pictures below are showing the console, and the user panel output.



```
[2023-07-19 15:21:36.932] - myRoom - {
  'FB-Level1-ch1-level': -80,
  'FB-Level1-ch1-mute': true,
  'FB-Mute1-ch1-mute': true,
  'FB-Level2-ch1-level': -100,
  'FB-Level2-ch2-level': -100,
  'FB-Level2-ch1-mute': true,
  'FB-Level2-ch2-mute': true,
  'FB-USB-USBXOutput1-connected': true,
  'FB-USB-USBXInput1-connected': true
}
```



DEBUG PANEL

Variables - Values

	Variable name	Value
Last frame received from Tesira	frame	! "publishToken":"FB-Mute1-ch1-mute" "value":true
Value of subscribed parameters	FB-Level1-ch1-level	-80.0000
Value of subscribed parameters	FB-Level1-ch1-mute	true
Value of subscribed parameters	FB-Mute1-ch1-mute	true
Value of subscribed parameters	FB-Level2-ch1-level	-100.0000
Value of subscribed parameters	FB-Level2-ch2-level	-100
Value of subscribed parameters	FB-Level2-ch1-mute	true
Value of subscribed parameters	FB-Level2-ch2-mute	true
Value of subscribed parameters	FB-USB-USBXOutput1-connected	true
Value of subscribed parameters	FB-USB-USBXInput1-connected	true
Last query	QR-Mute1-ch1-mute	false

5. Appendix

5.1. Further Information

Document Revision History

Rev.	Release date	Changes	Editor
1.0.0	21.07.2023	Initial version	Péter Krischneider
1.0.1	23.01.2024	Added release notes	Péter Krischneider