Introduction

Yosco is a protocol bridge that receives OSC messages and translates them into a format that can be understood by Yamaha’s CL, QL and TF series consoles. Yosco runs on the same machine as the OSC sender, which can be any arbitrary application, such as QLab. It is not a controller in itself, it is a translator.

UI Indicators:

Send: Blinks when Yosco successfully dispatches a command to the network. Note that this does not indicate whether the message has reached a destination, nor whether it has been successfully interpreted, i.e. your Yamaha device may receive the message, but fail to properly read it.

Receive: Blinks when Yosco successfully receives an OSC message. Note that this does not indicate whether the message is valid or can be properly interpreted and translated by the Yosco application. Invalid messages and arguments generally open an error dialog.

UI Controls:

Receive Port: This is the network port that Yosco will receive OSC messages on. This is the port to which your OSC application should send messages.

Console IP: The IP address of the console you want to control. Currently, only one console can be controlled at once.

Console Model: Allows you to select the model of console you would like to control. While the OSC API is identical for each model, this is not the case for the format to which they are being converted. Further, Yosco will place constraints on what you can target, e.g. channel numbers, and will raise an error if you target a channel outside the console’s range, e.g. calling channel 64 on a QL1.

Set Up

Example using QLab

In QLab >>Settings>>Network create a network patch with the destination “localhost” and any port number; the 35000-36000 range is recommended.

Create a Network Cue, and select the Network Patch you just created.

Enter a valid OSC string, e.g. /set/mixer/input/5/fader/level -20

In Yosco, in the Receive Port parameter, enter the port number you designated in the QLab Network patch.

Enter the IP Address of the console you would like to control.

Select the model you want to control

You’re now ready to control your Yamaha console with OSC

API

/set/mixer/input/[n]/fader/level

/set/mixer/input/[n]/fader/on

/set/mixer/input/[n]/panmode

/set/mixer/input/[n]/pan (MIXER:Current/InCh/ToSt/Pan)

/set/mixer/input/[n]/role (MIXER:Current/InCh/Role)

/set/mixer/input/[n]/gain (MIXER:Current/InCh/Port/HA/Gain)

/set/mixer/input/[n]/dca/[d]/assign

/set/mixer/input/[n]/label/name

/set/mixer/input/[n]/label/color

/set/mixer/input/[n]/to\_mix/[m]/fader/on (get rid of fader?)

/set/mixer/input/[n]/to\_mix/[m]/fader/level (get rid of fader?)

/set/mixer/input/[n]/to\_mix/[m]/pan

(make these strings without arguments?

And instead of pre-post, just commands in and of themselves)

/set/mixer/input/[n]/to\_mix/[m]/pre

/set/mixer/input/[n]/to\_mix/[m]/post

/set/mixer/input/[n]/to\_mtx/[mtx]/level

/set/mixer/input/[n]/to\_mtx/[mtx]/on

DCA

/set/dca/[n]/fader/level [MIXER:Current/DCA/Fader/Level]

/set/dca/[n]/fader/on [MIXER:Current/DCA/Fader/On]

Bridge Control API

/bridge/network/port

/bridge/network/ip

/bridge/console

/bridge/on

/bridge/off