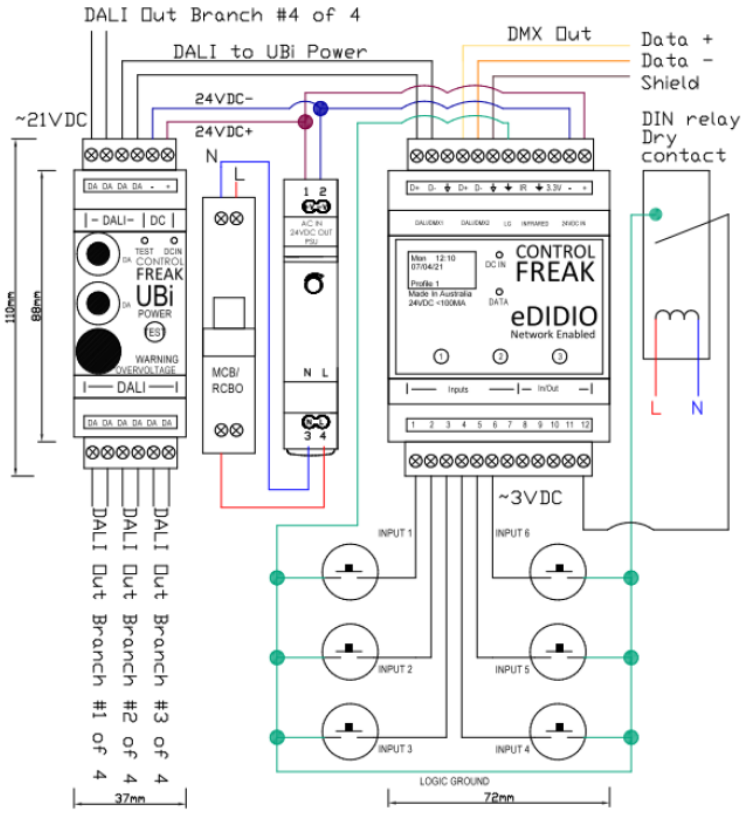
# CBUS NAC Control Freak® eDIDIO Library

Firmware Version – 1.1.0
Date – 2/9/24

**Introduction**
This document outlines the CBUS NAC Control Freak® eDIDIO Lua library. This guide shows how to use the library to control DALI and DMX interfaces via TCP/IP.

**Hardware Considerations**
The eDIDIO controller must be connected to a 24V power supply. It must be on the same network as the CBUS NAC. Appropriate network settings must be configured to allow a TCP connection between the NAC device and the eDIDIO controller.

If the eDIDIO controller has DALI, it must have a suitable DALI PSU to control the lines. The DALI line(s) should be addressed and grouped as necessary. We suggest the Control Freak UBi DALI PSU. An example 1D1X is shown below.




The eDIDIO can be ordered in a multiple of configurations. 1D1X refers to DALI | DMX. 2D is DALI | DALI.
A 4-line unit is also available

# 5500NAC Configuration

- Log into the 5500NAC Configuration. This can be done by entering the units IP address into a browser.
- Click on Configurator
- Navigate to Scripting -> Tools and press Restore Scripts
- Open the eDIDIO Scripting Library provided.

# Lua Functions

The library includes several functions for DALI and DMX commands, as well as enumerations for ease of use.

## Enumeration

- DALI_0 to DALI_63
- DALI_G0 to DALI_G15
- DALI_BROADCAST
- DALI Type 8
    - SET_TEMP_X_COORD
    - SET_TEMP_Y_COORD
    - ACTIVATE
    - SET_TEMP_COLOUR_TEMP
    - COLOUR_TEMP_COOLER
    - COLOUR_TEMO_WARMER
- DALI Fade
    - DALI_NO_FADE
    - DALI_0_7s_FADE
    - DALI_1_0s_FADE
    - DALI_1_4s_FADE
    - DALI_2_0s_FADE
    - DALI_2_8s_FADE
    - DALI_4_0s_FADE
    - DALI_5_7s_FADE
    - DALI_8_0s_FADE
    - DALI_11_3s_FADE
    - DALI_16_0s_FADE
    - DALI_22_6s_FADE
    - DALI_32_0s_FADE
    - DALI_45_3s_FADE
    - DALI_64_0s_FADE
    - DALI_90_5s_FADE

## TriggerType

DALI_ARC = 0 -- For controlling DALI Arc Levels (0 to 254) and 255 for MASK
DALI_COMMAND = 1
DMX_CHANNELS_SPLIT_LOW = 2 -- NOTE: Expects the channel number (not zero-based)
DMX_CHANNELS_SPLIT_HIGH = 3 -- NOTE: Expects the channel number (not zero-based)
DMX_MULTICAST_CHANNELS_SPLIT_LOW = 4 -- NOTE: Expects the channel INDEX to start from, as it takes into account the start address set from Spektra
DMX_MULTICAST_CHANNELS_SPLIT_HIGH = 5 -- NOTE: Expects the channel INDEX to start from, as it takes into account the start address set from Spektra
DMX_BROADCAST = 6 -- Affects all DMX lights as per the Spektra Settings (number of lights and channels per light)
DIDIO = 7 -- DEPRECATED
FADE_UP_WITH_MIN = 8 -- DALI Fade Up Command - Query level and set Minimum if Off
LIST_START = 9 -- Start a List action once
LIST_START_CONTINUOUS = 10 -- Start a List action with repeat
LIST_STOP = 11 -- Stop a List
SPEKTRA_START_SEQ = 12 -- Start a Spektra Sequence
SPEKTRA_STOP_SEQ = 13 -- Stop a playing Spektra Sequence
SPEKTRA_THEME = 14 -- Apply a Spektra Theme
SPEKTRA_STATIC = 15 -- DEPRECATED
SPEKTRA_SCHEDULE = 16 -- Start the scheduled Spektra item
LINK_START = 17 -- Enables the UDP Link State - If Configured
LINK_STOP = 18 -- Temporarily disables the UDP Link State
DISABLE_BURN = 19 -- Disable Burn-In
ENABLE_BURN = 20 -- Enable Burn-In
ON_OFF_TOG = 21 -- Turn a Group/Addres On/Off based on query level. If DALI_GROUP_ALL, toggle based on flag
MIN_MAX_TOG = 22 -- On/Off Toggle replaced by Min/Max
ENABLE_INPUT = 23 -- Enable Input - If latching, Input will trigger immediatly
DISABLE_INPUT = 24 -- Disable Input
ENABLE_TOG_INPUT = 25 -- Toggle Enable/Disable Input
OUTPUT_TOG = 26 -- Toggle Output State between High (~22Vdc) and Low (0Vdc)
OUTPUT_HIGH = 27 -- Set Output HIGH
OUTPUT_LOW = 28 -- Set Output LOW
OUTPUT_TRIG = 29 -- Set Output to trigger momentarily based on configuration
PROFILE_CHANGE = 30 -- Change Profile - This action will reset sensor state
FADE_LONG_PRESS = 31 -- Long Press Fade based on Toggle Flag
SYNCRO = 32 -- Command sets clock to 11:59PM. Used for hardware time update by external Timeclock
PRESET_CODE = 33 -- Preset Code - See Configurator Description
CUSTOM_CODE = 34 -- Project Specific Custom Code - Talk to Creative Lighting for support
SPEKTRA_SLEEP = 35 -- Pause Spektra sequence
SPEKTRA_RESUME = 36 -- Resume Spektra sequence
DEVICE_RESET = 37 -- Admin Command for Hardware Reset
DEVICE_SAVE = 38 -- Admin Command for manual device memory save
USER_LEVEL_STORE_NEW = 39 -- Store Current Level to Variable
USER_LEVEL_SET_DEFAULT = 40 -- Reset User Level Variable
USER_LEVEL_RECALL = 41 -- Recall User Level Variable
ROOM_JOIN = 43 -- DEPRECATED
ROOM_UNJOIN = 44 -- DEPRECATED
TYPE8_TC_WARMER = 45 -- DALI Type 8 Warmer Command. 1 Mirek increments
TYPE8_TC_COOLER = 46 -- DALI Type 8 Cooler Command. 1 Mirek increments
TYPE8_TC_ACTUAL = 47 -- DALI Type 8 Set Colour to Mirek value
LOGIC_OPERATION = 48 -- Not Implemented
ALARM_ENABLE = 49 -- Enable Alarm at Index
ALARM_DISABLE = 50 -- Disable Alarm at Index
DALI_CONTROL_SENSOR_OVERRIDE = 51 -- Puts the DALI Sensor in 'override mode', which means it will no longer control the lighting until occupancy has timed-out or control is manually resumed
DALI_CONTROL_SENSOR_TEMP_DISABLE = 52 -- Sets the occupancy timer to zero and puts the DALI Sensor in a temporary 'disable mode' (duration depends on Sensor configuration: 'Disable Period')
DALI_CONTROL_SENSOR_RESUME = 53 -- Takes the DALI Sensor out of 'override mode'
DALI_ARC_OVERRIDE = 54 -- For controlling DALI Arc Levels (0 to 254) and 255 for MASK - Sets associated group to override mode
DALI_COMMAND_OVERRIDE = 55 -- For sending DALI commands - Sets associated group to override mode
FADE_UP_WITH_MIN_OVERRIDE = 56 -- Non-native DALI command override (sets associated group to override mode)
ON_OFF_TOG_OVERRIDE = 57 -- Non-native DALI command override (sets associated group to override mode)
MIN_MAX_TOG_OVERRIDE = 58 -- Non-native DALI command override (sets associated group to override mode)
MAX_OFF_TOG = 59 -- Not Implemented

MAX_OFF_TOG_OVERRIDE = 60 -- Not Implemented
FADE_LONG_PRESS_OVERRIDE = 61 -- Non-native DALI command override (sets associated group to override mode)
USER_LEVEL_RECALL_OVERRIDE = 62 -- Non-native DALI command override (sets associated group to override mode)
DMX_ZONE_FADE_UP = 63 -- DMX Spektra Zone Fade UP
DMX_ZONE_FADE_DOWN = 64 -- DMX Spektra Zone Fade DOWN
LOGGING_LEVEL = 65 -- Enable Logging to EEPROM to be read by configurator
SPEKTRA_SHOW_CONTROL = 66 -- DEPRECATED
CIRCADIAN_TEMPERATURE = 67 -- Selects Colour Temperature based on clock
DALI_CONTROL_SENSOR_MUTE = 68 -- Mute Sensor at Index (or all with Index 255)
DALI_CONTROL_SENSOR_UNMUTE = 69 -- Unmute to Sensor at Index (or all with Index 255)
SPEKTRA_INTENSITY = 70 -- Allow you to specify the maximum Spektra Sequence or Theme output intensity (10 to 100)%
ENABLE_INPUT_NO_ACTION = 71 -- Allow you to enable an input (Latching), but not trigger the action.
SET_DALI_FADE_TIME = 72 -- Sets the DALI Fade Time
NO_COMMAND = 254 -- This TriggerType should always be at the bottom of the list. Add any new TriggerTypes above it (up to 253).

## Functions

| Function | Command |
|---|---|
| Sends 3 DALI Levels to an RGB fixture | sendDALIRGBMessage(line, address, red, green, blue) |
| Converts RGB to XY Coordinates for DALI DT8 | sendDALIRGBDT8Message(line, address, red, green, blue, brightness) |
| Converts Kelvin to Mirek and outputs DALI CCT DT8 | sendDALICCTDT8Message(line, address, kelvin, brightness) |
| Send a DALI Level (0-254) | flag = sendDALIArcLevel(line, address, level) |
| Set the DALI fixtures to a specific fade (See Enums) | flag = sendDALIFadeMessage(line, address, fadetime) |
| Sends a specific DT8 Command | flag = sendDT8Cmd(line, address, cmd, arg) |
| Get DALI Level (0 to 254) | flag, level = getDALILevel(line, address) |
| Sends a singular DMX Level. Fade = fadetime * 10ms. Repeat for block commands | sendDMXLevel(line, channel, level, fadetime, repeat) |
| Sends a DMX RGB command. Repeat for consecutive addresses | sendDMXRGB(line, channel, red, green, blue, fadetime, repeat) |
| Sends a DMX RGBW command. Repeat for consecutive addresses | flag = sendDMXRGBW(line, channel, red, green, blue, white, fadetime, repeat) |
| Send a trigger, specified by TriggerType | flag = sendTrigger(line, zone, TriggerType, target, value, query) |

## Parameters

- Line – This depends on the installed hardware. I.e. eDIDIO 1D 1X = DALI | DMX
    - Physical Line 1 = 0x01
    - Physical Line 2 = 0x02
    - Physical Line 3 = 0x04
    - Physical Line 4 = 0x08
    - Multiple Lines (Line 1 + Line 2) = 0x01 + 0x02 = 0x03
- Address – See Enum. DALI_0 to DALI_63 + DALI_G0 to DALI_G15 + DALI_BROADCAST
- Red, Green, Blue, White, Brightness
    - For DALI, values range from 0 to 254
    - For DMX, values range from 0 to 255
- Fadetime
    - For DALI, seen Enum, DALI_0_7s_FADE = 0.7s fade
    - For DMX, total fade time = fadetime * 10ms.
- Repeat
    - Value can be 0 to Max DMX.
        - For Standard DMX, value can be 512 – channel
        - For RGB, value can be 512/3 – channel
        - For RGBW, value can be 512/4 – channel
- Flag – Success or Failed

## Library Configuration

Library Configuration is through the eDIDIOConfig script. Enter the IP of the eDIDIO.

The IP can be found through the DIDIO Configuration software, or via the keypad on the unit.

```
eDIDIOS10_1 = {
    ip = "192.168.20.137"
}
eDIDIOS10_2 = {
    ip = "192.168.20.228"
}
```

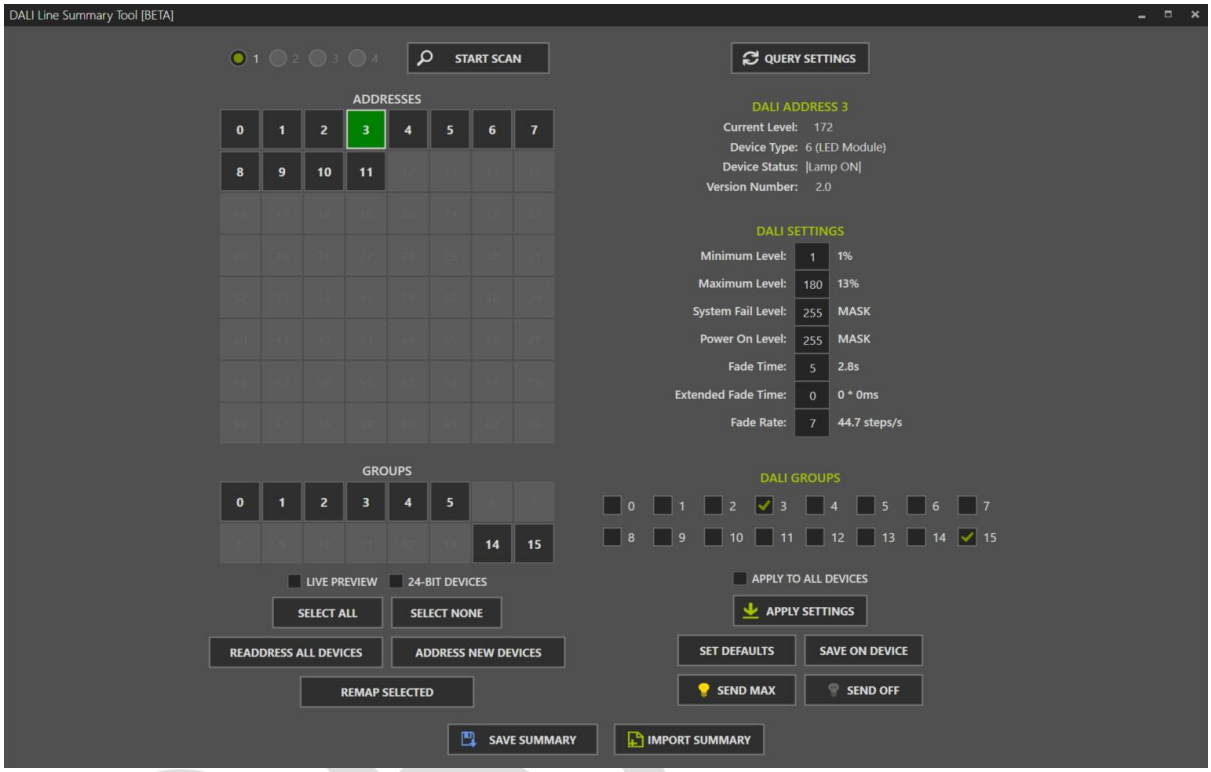
## Examples

```
require("user.eDIDIOCore")
require("user.eDIDIOConfig")

-- Example Usage

-- Get Level
value1 = GetLightingLevel(0)
if (value1 == 255) then
    value1 = 254    -- DALI max level
end

-- Create new eDIDIO instance
local eDIDIOS10_1 = EDIDIO.new(eDIDIOS10_1)

-- Send an Arc Level
local flag = eDIDIOS10_1:sendDALIArcLevel(LINE_1, DALI_0, value1)
-- Example Logging
log(flag)

-- Create a second eDIDIO instance
local eDIDIOS10_2 = EDIDIO.new(eDIDIOS10_2)
eDIDIOS10_2:sendDALIRGBDT8Message(LINE_1, DALI_0, 127, 100, 50, 100)
-- Send an RGBDT8 Message, {127, 100, 50}
eDIDIOS10_2:sendDALICCTDT8Message(LINE_1, DALI_0, 3000, 254)
-- Send an CCTDT8 Message. 3000K, 100% output
eDIDIOS10_2:sendDMXLevels(LINE_2, 10, {255, 127, 0}, 100, 5)
-- Send DMX Levels to Line 2, Address 10. Levels 255, 127, 0, fade over 1 second
eDIDIOS10_2:sendDMXRGBW(LINE_2, 1, 0, 10, 20, 30, 100, 2)
-- Send DMX Levels to an RGBW fixture, 1 second fade, repeat over 2 fixtures
eDIDIOS10_2:sendTrigger(LINE_1, 0, DALI_ARC_OVERRIDE, DALI_1, 200, 0)
-- Send DALI Arc level with sensor override to DALI_1, level 200, line 1
```

## DALI Installation

The DALI fixtures should be configured into controllable groups. This can be done using the Control Freak DIDIO Configurator software.

https://github.com/CreativeLightingAdmin/DIDIO-Configurator-Releases




## DALI Speed

DALI DT8 commands are slow. Please note that to change colour via DT8 can take around 11 DALI messages.