



## *Button Helper*

### **1. Intro**

Programming multiple functions or behaviours into buttons in Control4 can be a bit tricky, and you may not achieve the performance you desire. Or sometimes can be accomplish with a combination of programming, but this is time-consuming and can lead to more errors, especially when the LED behaviours needs to follow specific conditions.

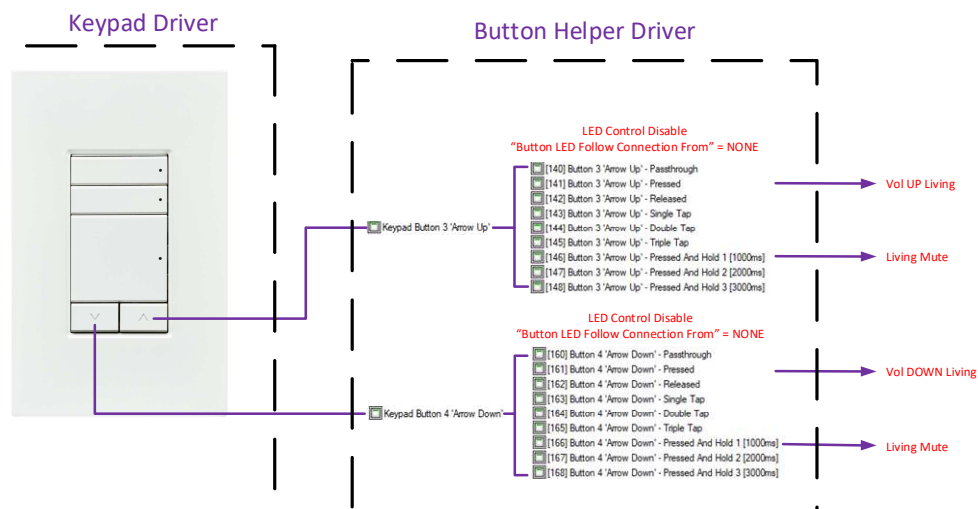
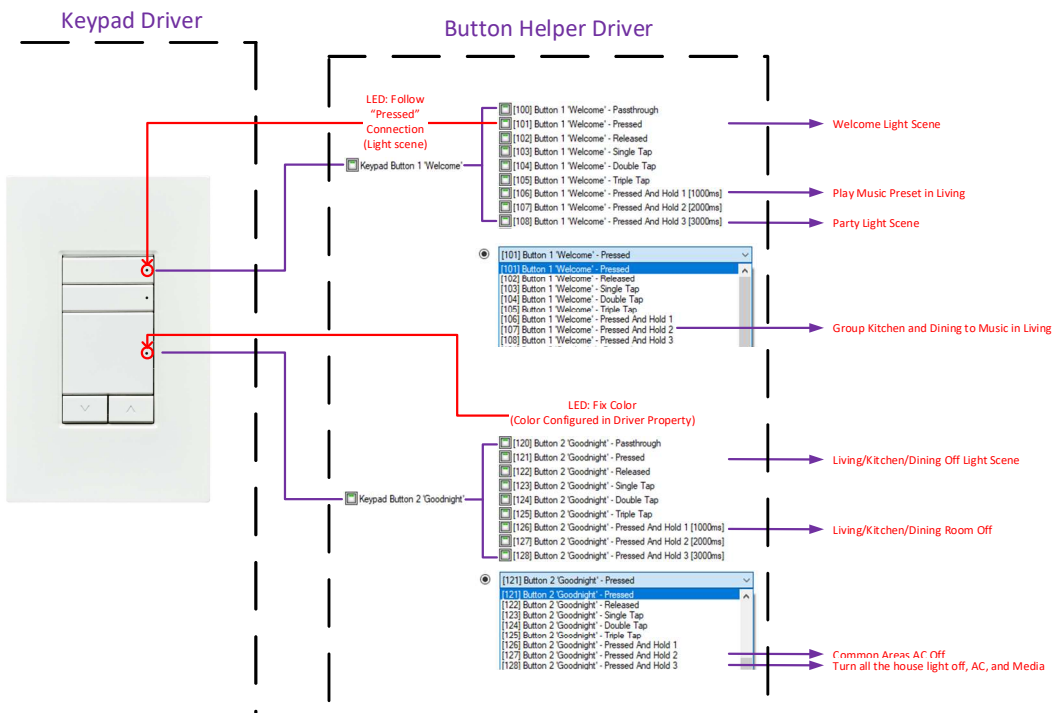
This driver was created to address this need. It allows you to enhance the functionality of a button (any type of device with a "button" connection in Composer Pro). Once you connect a button to the driver, you can distinguish between various button actions, including:

- Passthrough
- Pressed
- Released
- Single Tap
- Double Tap
- Triple Tap
- Press And Hold 1
- Press And Hold 2
- Press And Hold 3

Each action has a specific Composer Pro connection and event, making it easy to configure the button to have different responses based on user interaction.

Additionally, the button's LED can be configured to follow a specific connection, remain fixed with a certain configured color or not control the LED.

Here some examples:



The driver was tested on OS 3.3.3 and OS 3.4.0 but should work from OS 2.10+. Please any doubt, contact **Barbini.dev**.

## 2. Properties

- **Cloud Status:** Current status of the driver license.

- *Automatic Updates*: Enable driver automatics updates (Recommended!)
- *Driver*: Driver name.
- *Driver Version*: Current driver version.
- *Driver Information*: Last debug information.
- *Debug Mode*: If On, driver will print debug information on Lua output.
- *Button Quantity*: You can configure up to 10 buttons per instance of the driver. The project can have more than one instance of driver.
- *Multi Tap LED Wink*: On/Off – Enable or disable the LED Wink after a multi tap event.  
NOTE: This function is only applied on double and triple multi tap.
- *Multi Tap LED Wink Color*: RGB Color – Select the colour of the wink.
- *Pressed And Hold 1 Timer*: Time, in ms, that the button needs to be pressed to reach the first press and hold function.
- *Pressed And Hold 2-1 Time Difference*: Time, in ms, after the first press and hold function to reach the press and hold function 2.
- *Pressed And Hold 3-2 Time Difference*: Time, in ms, after the second press and hold function to reach the press and hold function 3.
- *Press And Hold LED Wink*: On/Off – Enable or disable the LED Wink after a press and hold event.
- *Press And Hold LED Color*: RGB Color – Select the colour of the wink.
- *Button Configuration*: Select the button you want to configure.
- *Button Name*: You can set the name of each button to easily identify it using connections filter.
- *Button Multi Tap Functions*: Yes/No – Enable or disable the button Multi Tap functions.
- *Button Press And Hold Functions*: Yes/No – Enable or disable the button Press And Hold functions.
- *Button LED Follow Connections From*: Select the connection where you want the LED to follow the connection from. NONE will ignore all the LED control. If select FIX\_COLOR, the next property appears.
- *Button LED Fix Color*: RGB Color – Select the color of the button LED.

### **3. Actions**

- N/A

#### 4. Commands

- *Set Press And Hold LED Wink*: On/Off – Enable or disable the LED Wink after a press and hold event.
- *Set Press And Hold LED Color*: RGB Color – Select the colour of the wink.
- *Set Multi Tap LED Wink*: On/Off – Enable or disable the LED Wink after a multi tap event. NOTE: This function is only applied on double and triple multi tap.
- *Set Multi Tap LED Wink Color*: RGB Color – Select the colour of the wink.
- *Button Enable/Disable Multi Tap*: Select the button you want to Enable/Disable Multi Tap functions.
- *Button Enable/Disable Multi Press And Hold*: Select the button you want to Enable/Disable Multi Press And Hold functions.
- *ALL Buttons Enable/Disable Multi Tap*: Enable/Disable Multi Tap functions to ALL buttons.
- *ALL Buttons Enable/Disable Multi Press And Hold*: Enable/Disable Multi Press And Hold functions to ALL buttons.

#### 5. Events

- ❖ *Button X – Pressed*: Triggered when the button “x” was pressed.
- ❖ *Button X – Released*: Triggered when the button “x” was released.
- ❖ *Button X – Single Tap*: Triggered when the button “x” was single tap\*.
- ❖ *Button X – Double Tap*: Triggered when the button “x” was double tap\*.
- ❖ *Button X – Triple Tap*: Triggered when the button “x” was triple tap\*.
- ❖ *Button X – Press And Hold 1*: Triggered when the button “x” was pressed and hold until 1<sup>st</sup> function\*\*.
- ❖ *Button X – Press And Hold 2*: Triggered when the button “x” was pressed and hold until 2<sup>nd</sup> function\*\*.
- ❖ *Button X – Press And Hold 3*: Triggered when the button “x” was pressed and hold until 3<sup>rd</sup> function\*\*.

**\*NOTE: This will be executed if the multi tap functions of the button are enabled.**

**\*\*NOTE: This will be executed if the multi press and hold functions of the button are enabled.**

## **6. Driver Variables**

❖ NA

## **7. Best Practices, warnings, and recommendations**

- Be careful not generating LOOPS in the driver (For example, the pressed connection toggle a Light, and the press and hold 1 toggle a scene that contains that light).
- If you use the Wink function, be cautious if you have a large ZIGBEE mesh or an already high ZIGBEE traffic. Test the wink performance to ensure everything works smoothly. If you encounter unexpected behavior, consider disabling the Wink function.
- Do not daisy chain Button Helper drivers.
- The driver will control the button LED if the “LED Behavior” of the button was configured as “Follow connection”. If not, select “NONE” on the Button Helper Driver “Button LED Follow Connections From” property.
- You can use several instances of the driver per project.

## **8. Warranty and disclaimer**

**Barbini.dev** provide drivers tested as much as possible. However, due to the highly variants of systems setups and the constant changing of the electronics systems, updates and/or modifications may be required to fixing bugs or improve the driver functionality.

Many of our drivers interact with systems APIs (like Control4 and/or any other 3<sup>rd</sup> party system). Therefore, if any API changes, the driver may stop functioning correctly. **Barbini.dev** cannot guarantee long-term functionality of any driver developed that uses APIs. **Barbini.dev** has the right to repair, provide updates, or discontinue a driver at any time. These repairments or updates could be free or required additional expenses. Despite the above, our efforts will always be to maintain the drivers if possible and free of charge of upgrades as much as possible.

**Developed By: Barbini.dev**

**Contact: [barbini.dev@highqualityautomation.com](mailto:barbini.dev@highqualityautomation.com)**



---

**VERSION 231024xx:** Initial released