



# TechNotes

Revision 12/14/2016

## Create Custom Effects (For Effect 1)

### Overview:

An Effect is an automatic sequencing of one or more static colors which provides the illusions of motion or an animated light show. Within each ILC-x00 controller, there may be one or more **Effects** available, but within each unit there is a least one customizable Effect, known as **Effect(1)**, which can be customized by an installer or dealer.

This document explains how to create a custom effect for **Effect(1)**, based on 1 to 23 user or installer custom-selected colors. This is typically used for clients who may wish to play a holiday sequence of Green and Red, for instance or Red, White, and Blue, which would repeat indefinitely or until an OFF, STOP or other color command is issued. A more sophisticated example employed by hospitality venues (hotels for instance) that might want to customize their lobby colors or ballroom colors on the fly to match those colors of visiting sports teams or incoming convention groups. In this case where these pre-programmed colors would by necessity need to be changed for a wide variety of clients, typically a third-party automation system is used (i) to act as the storage medium for the pre-programmed colors, (ii) to download at run time (when the Effect was desired to be run) these colors into an ILC-x00 controller and (ii) to issue the **EFFECT(1)** command to call up the custom color set.

There are three distinct methods by which **EFFECT(1)** can be programmed. These are

Type	Programming Tool	Method
1	E-Node Pilot or Web Pilot	-Installer enters specific RGB values for <b>PRESET(1)</b> to <b>PRESET(n)</b> within the particular ILC-x00 device -Installer enters special <b>JumpToCommand</b> (RGB value of white--240.240.240) into the first unused Preset location (n+1)
2	Automation GUI (within Control4, Crestron, Elan, TI for instance)	-User or Installer selects first target color through sliders and then stores that color into <b>PRESET(1)</b> using GUI buttons within the particular ILC-x00 device -User or Installer selects subsequent target color through sliders and then stores than color into <b>PRESET(2)</b> using GUI buttons -User or Installer continue process until up to 23 colors are programmed into <b>PRESET(n)</b> locations, again using GUI buttons -User or Installer enters special <b>JumpToCommand</b> (RGB value of white--240.240.240) into <b>PRESET(n+1)</b>
3	Automation System Built-in macros (within RTI, Creston for instance)	Macro includes the following steps: -Color is stored in <b>PRESET(1)</b> location in the particular ILC-x00 device -Subsequent colors are stored in sequential <b>PRESET(n)</b> locations in the particular ILC-x00 device - <b>JumpToCommand</b> (RGB value of white--240.240.240) is stored in <b>PRESET(n+1)</b> location in the particular ILC-x00 device

**-EFFECT,1** command is issued to trigger macro

Over the next few pages are several examples that will show each of the various Types specified above (see color code under each Example matching the above table).

**Example 1**

A Red, White and Blue color set is desired to be programmed as **EFFECT(1)**. It is desired that the period of time that each color is illuminated (**SEQRATE**) should be 5 seconds and the fade time between the colors should be 1 second (**DISSOLVE3**). We will describe this process both in terms of **Type 1** programming as well as **Type 2** programming.

**Type 1 Programming (through e-Node Pilot or Web-Pilot)**

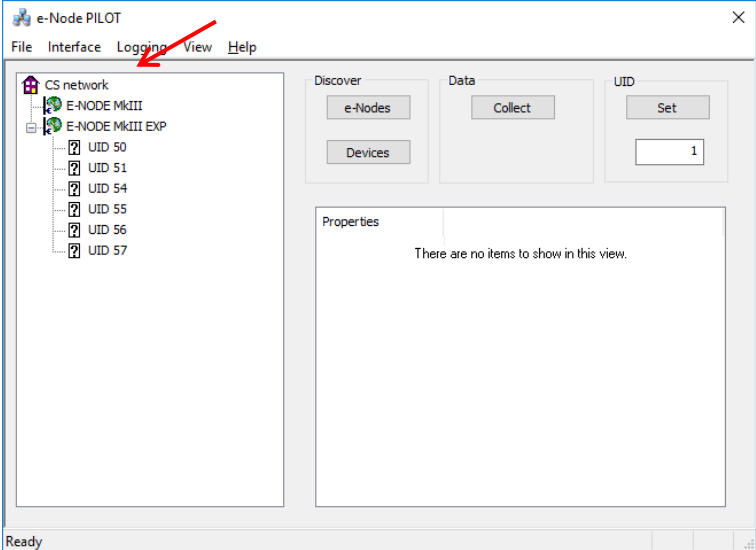
This section explains how to create a custom effect based on customer or installer-selected colors using the e-Node or Web-Pilot Application.

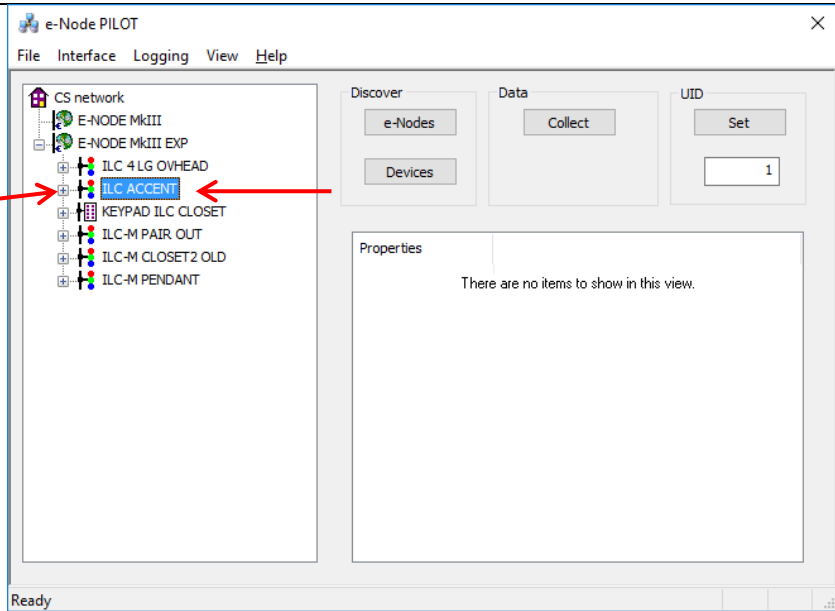
**Required Tools:**

You will need the following:

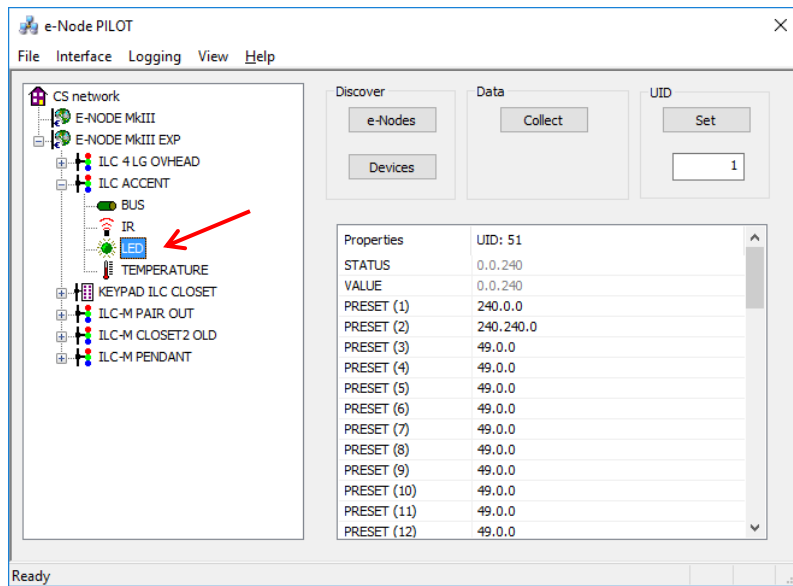
- 1.Windows e-Node Pilot application or Web-Pilot application
- 2.Previously set up e-Node and one or more ILC-x00 controllers addressed with UIDs and Zone/Group/node address (see [Quick Start Guides](#) for more information).

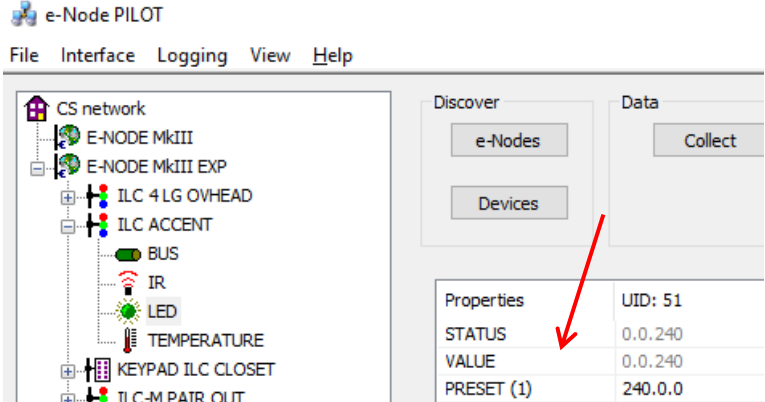
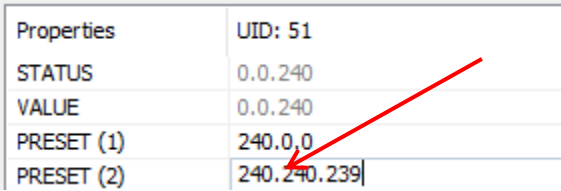
**Storing Colors and JumpToCommand into PRESET locations**

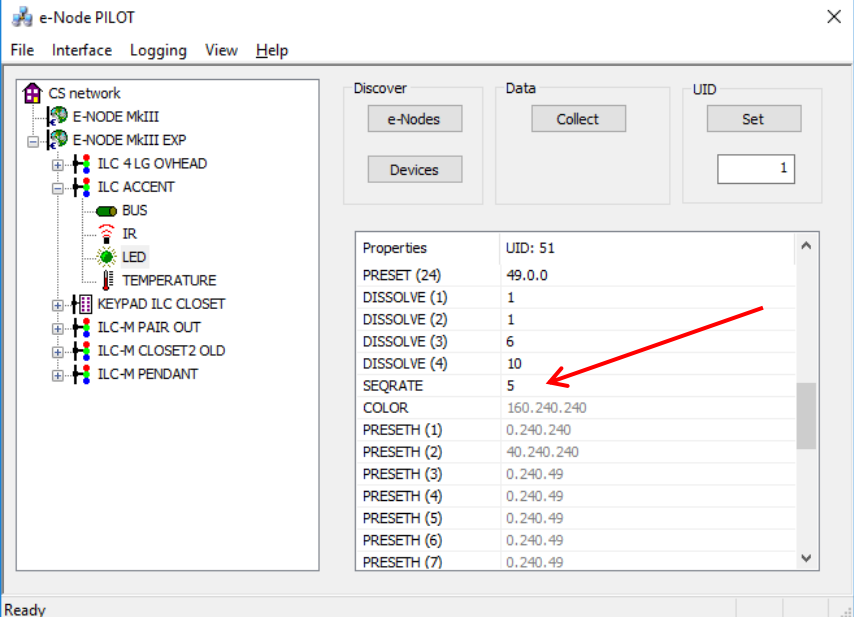
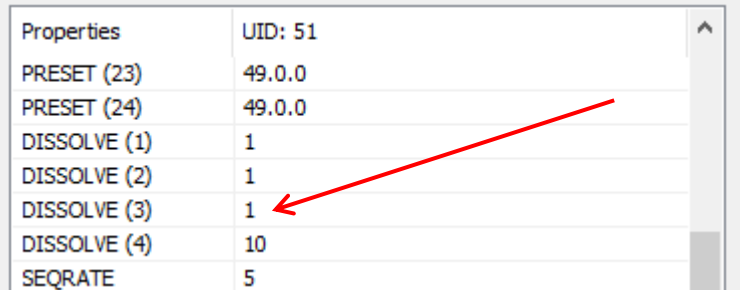
Step	Tasks	Detail
P-1a	Launch application utility	<p>-Open e-Node Pilot (or Web-Pilot) application. Select the <b>View Map</b> tab.</p>  <p>-Select the device into which you wish to store the lighting preset. In this case we will select ILC Accent</p>



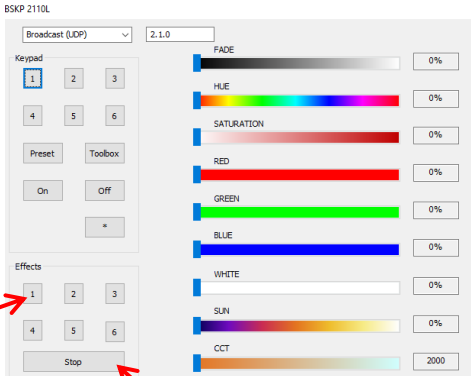
-Expand the + mark in front of the targeted ILC-x00 device and select the LED tab



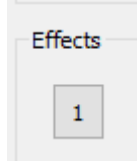
P-1b	Assign Red to <b>PRESET 1</b>	<p>Next to the <b>PRESET (1)</b> entry input the color red in RGB mode (240.0.0)</p> 
P-1c	Assign White to <b>PRESET (2)</b>	<p>Next to the <b>PRESET (2)</b> entry input the color white in RGB mode (240.240.240), <b>but wait a minute</b>. White is the one color (240.240.240) that just happens to conflict with our special <b>JumpToCommand</b> used to trick our software in the next step. <i>Therefore, adjust the desired White by just one digit to simulate a white but not to confuse the system.</i></p> <p>-In this case, enter 240.240.<b>239</b>.</p> <p>-Hit <b>Enter</b> to save.</p> 
P-1d	Assign Blue to <b>PRESET 3</b>	<p>-Next to the <b>PRESET (3)</b> entry input the color Blue in RGB mode (0.0.240).</p> <p>-Hit <b>Enter</b> to save.</p>
P-1e	Assign <b>JumpToCommand</b> to <b>PRESET 4</b>	<p>Now that all of your sequence colors have been saved into <b>PRESET(1)</b> , <b>PRESET(2)</b> and <b>PRESET(3)</b>, we are now ready to enter the special <b>JumpToCommand</b> into the first PRESET after the active PRESETS.</p> <p>-Next to the <b>PRESET (4)</b> entry input the special <b>JumpToCommand</b> of 240.240.240.</p> <p>-Hit <b>Enter</b> to save.</p>
P-1f	Set the <b>SEQRATE</b> to 5	<p>The <b>SEQRATE</b> command specified the time (after any dissolve) that the preset color is maintained before transitioning to the next color in sequence.</p> <p>-Scroll down under LED to <b>SEQRATE</b> and set to 5</p>

		
P-1g	Set <b>DISSOLVE (3)</b> to 1 (second)	<p>The <b>DISSOLVE (3)</b> command is the period of time for transition from one state to another for <b>EFFECT (1)</b>.</p> 

**Test EFFECT (1) using Virtual Terminal and Finish Up GUI Programming**

P-2a	Launch e-Node Pilot Virtual Terminal	<p>Select View/Virtual Terminal/Lighting to expose the Virtual Lighting Terminal</p> 
P-2b	Select <b>EFFECT (1)</b> to execute Effect	Any connected RGB LEDs will run through the sequence, Red, White, Blue, Red, White, Blue...
P-2c	Hit the <b>Stop</b> button to stop sequence	Press Stop to stop sequence (or any other command)

### Create a GUI Button for Effect(1) to activate this Red,White,Blue Effect

P-3a	Create a GUI Button for <b>EFFECT (1)</b>	<p>Program you automation system or lighting system with a button entitled Effect (1) or similar.</p> <div data-bbox="938 338 1073 499" style="text-align: center;">  </div> <p>Connect this button to our command for <b>Effect (1)</b>. If you had a device with an address of 2.1.1 here would be the command  <b>#2.1.1.LED=EFFECT,1;&lt;cr&gt;</b></p>
------	---	---

### Type 2 Programming (through Automation or Lighting Panel)

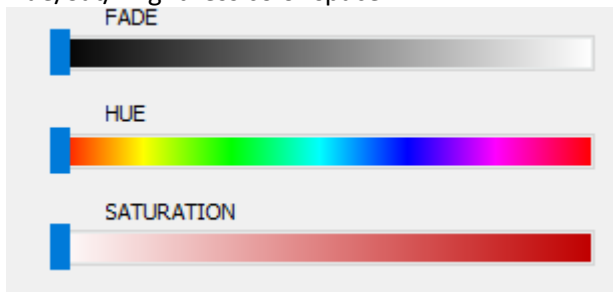
This section explains how to create a custom effect based on customer or installer-selected colors using a third-party Automation or Lighting System. We will describe this process in terms of **Type 2** programming.

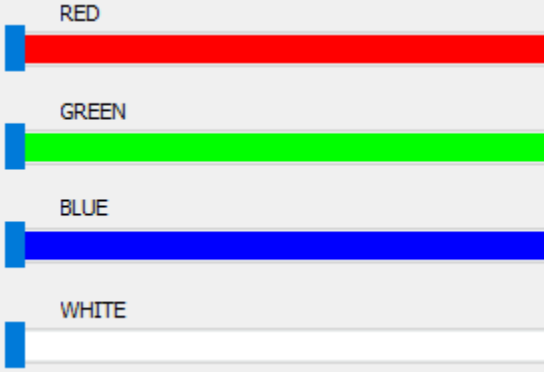

#### Required Tools:

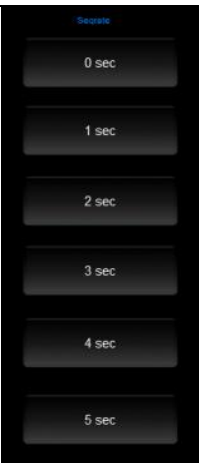

You will need the following:

1. Compatible Automation or Lighting Panel.
2. Previously set ILC-x00 controllers addressed with UIDs and Zone/Group/node address (see [Quick Start Guides](#) for more information).

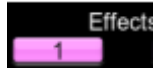
#### Create tasks in your Automation/Lighting System

Step	Tasks	Detail
A-1a	Create three sliders for <b>Hue/Saturation</b> and <b>Brightness</b> (or alternatively R,G,B, or RGBW sliders as appropriate)	<p>Here is a sample of what could be created here to select colors in the preferred Hue/Sat/Brightness color space</p> <div data-bbox="540 1304 1149 1593" style="text-align: center;">  </div> <p>If you operating in the RGBW color space, go ahead and create four sliders in this case.</p>

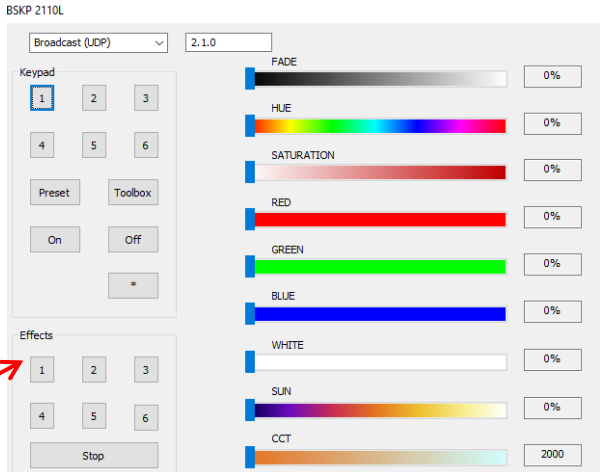
												
A-1b	<p>Create the number of <b>STORE(n)</b> buttons as the number of colors that will need to be stored for your color effect, PLUS 1.</p>	<p>In this case, you will need 3+1 STORE buttons to Store your three colors (plus the 4<sup>th</sup> button to store the special <b>JumpToCommand</b>).</p> <p><b>Note:</b> In this example just for aesthetics we have created 6 buttons.</p>  <p>-Program these buttons to send out the following commands (use your own Zone/Group/Address as appropriate).</p> <table border="1" data-bbox="544 993 1393 1178"> <thead> <tr> <th>Button</th> <th>Command</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>#2.1.1.LED=STORE,1;&lt;cr&gt;</td> </tr> <tr> <td>2</td> <td>#2.1.1.LED=STORE,2;&lt;cr&gt;</td> </tr> <tr> <td>3</td> <td>#2.1.1.LED=STORE,3;&lt;cr&gt;</td> </tr> <tr> <td>4</td> <td>#2.1.1.LED=STORE,4;&lt;cr&gt;</td> </tr> </tbody> </table>	Button	Command	1	#2.1.1.LED=STORE,1;<cr>	2	#2.1.1.LED=STORE,2;<cr>	3	#2.1.1.LED=STORE,3;<cr>	4	#2.1.1.LED=STORE,4;<cr>
Button	Command											
1	#2.1.1.LED=STORE,1;<cr>											
2	#2.1.1.LED=STORE,2;<cr>											
3	#2.1.1.LED=STORE,3;<cr>											
4	#2.1.1.LED=STORE,4;<cr>											
A-1c	<p>Create a One or more <b>SEQRATE</b> Buttons to select variable numbers (seconds) to transmit as the <b>SEQRATE</b></p>	<p>The <b>SEQRATE</b> command specifies the time (after any dissolve) that the preset color is maintained before transitioning to the next color in sequence.</p> <p>Various automation systems have motifs for keypad or dropdowns, etc. to pick number to concatenate to a command. Or for more simplistic cases, just create a few buttons and hard coat them with a specific number of seconds.</p>										

		 <p>In our case, we may want to allow a user to change the <b>SEQRATE</b> from 1 to x. In our example, we specified a <b>SEQRATE</b> of 5 (seconds), so at minimum create a button that can transmit our <b>SEQRATE</b> command and set it to 5</p> <p>Here would be the comman for a device with Z/G/N address of 2.1.1</p> <p><b>#2.1.1.LED.SEQRATE=5;&lt;cr&gt;</b></p>
A-1d	<p>Create one or more <b>DISSOLVE (3)</b> Buttons to select variable numbers (seconds) to transmit as the <b>DISSOLVE (3)</b> command</p>	<p>The <b>DISSOLVE (3)</b> command is the period of time for transition from one state to another for <b>Effect (1)</b>.</p> <p>Various automation systems have motifs for keypad or dropdowns, etc. to pick number to concatate to a command. Or for more simplistic cases, just create a few buttons and hard coat them with a specific nuber of seconds.</p>  <p>In our case, we may want to allow a user to change the <b>DISSOLVE (3)</b> from 1 to x. In our example, we specified a <b>DISSOLVE (3)</b> of 1 (second), so at minimum create a button that can transmit our <b>DISSOLVE (3)</b> command and set it to 1.</p>



		<p>Here would be the command for a device with Z/G/N address of 2.1.1</p> <p><b>#2.1.1.LED.DISSOLVE.3=1;&lt;cr&gt;</b></p>
1f	<p>Create an <b>EFFECT(1)</b> button to execute the customizable <b>Effect</b> feature</p>	<p>Here is a generic button labeled as the <b>EFFECT(1)</b> button.</p>  <p>Here would be the command for a device with Z/G/N address of 2.1.1</p> <p><b>#2.1.1.LED=EFFECT,1;&lt;cr&gt;</b></p>

**Test EFFECT(1) using Virtual Terminal and Finish Up GUI Programming**

2a	<p>Launch e-Node Pilot Virtual Terminal</p>	<p>Select View/Virtual Terminal/Lighting to expose the Virtual Lighting Terminal</p> 
2b	<p>Select <b>EFFECT (1)</b> to execute Effect</p>	<p>Any connected RGB LEDs will run through the sequence, Red, White, Blue, Red, White, Blue...</p>
2c	<p>Hit the <b>Stop</b> button to stop sequence</p>	<p>Press Stop to stop sequence (or any other command)</p>

## Example 2

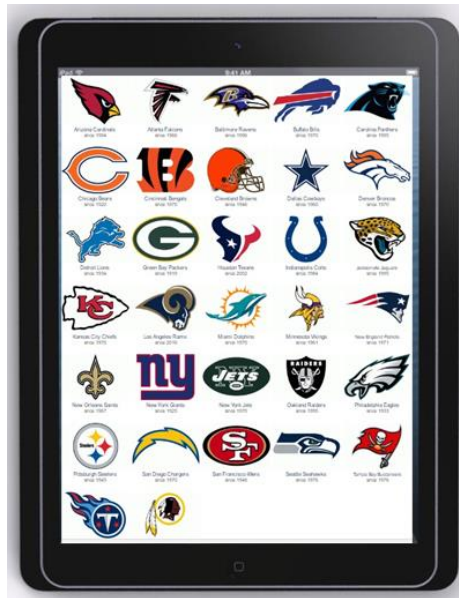
Assume you have an installation in a major market upscale hotel that caters to NFL teams visiting for football games. The host hotel wants to welcome the incoming team by proudly displaying the team's colors throughout its lobby. As an option, we demonstrate how a pair of adjustment values for Dissolve Rate (period between the color being on) and Sequence Rate (period that the color remains on) can be set as well.

Unfortunately, within any ILC-x00 controller, there are only 24 storage registers available so in this example we will use the power of the automation system to store those colors prior to runtime and then rely on that automation system to properly (i) download those color entries including a **JumpToCommand** and then (ii) transmit the **EFFECT(1)** command any time a particular logo button is selected (the magic of macros). In effect, the automation system will be used to store colors, transmit those colors, and issue an **EFFECT(1)** command for any logos selected.

Because of the sophisticated nature of this operation, only Type 3 programming (with Automation System) is possible.

### The GUI:

With an automation system, you could program 32 icons on you systems' GUI (don't be upset that we may not have selected your favorite team for our example).



## Type 3 Programming (through Automation or Lighting Panel)

This section explains how to create a custom effect based on customer or installer-selected colors using a third-party Automation or Lighting System. We will describe this process in terms of Type 3 programming only.

### Required Tools:

You will need the following:

1. Compatible Automation or Lighting Panel.

2. Previously set ILC-x00 controllers addressed with UIDs and Zone/Group/Node address (see [Quick Start Guides](#) for more information).

**Background:**

If you assume that you want to support logos for all NFL teams, and their respective colors, over 60 different colors would need to be programmed into an ILC-x00 device (but our controller only has 24 registers).

The easiest way to accomplish this task with any of our supported automation platforms (that support macros and the entry of RAW or ASCII commands) using our driver and ***harness the power of your automation system:***

- Store the first color into **PRESET(1)** on our controller (dynamically saved until the next color is downloaded into that location and therefore overwritten)
- Store the second color into **PRESET(2)** on our controller (dynamically saved until the next color is downloaded into that location and therefore overwritten)
- Store any remaining color(s) of the sequence into our controller into successive **PRESET(n)** locations until all colors are “saved”
- Store a final **JumpToCommand** (white 240.240.240) on our controller (dynamically saved until the next color is downloaded into that location and therefore overwritten)
- Send an execute **EFFECT,1** command which will then play those just written colors in a color sequence.
- Send optional SEQRATE and DISSOLVE(3) commands to impact the timing of the sequence

**Case 1 (where there is not a white color in the teams logos-like the Buffalo Bills)**



Under that logo would be a macro comprised of the following general steps:

Step	Actual programmed sequence for Buffalo Bills (see below for RGB colors for team)	Notes
1	<b>#2.1.1.LED.PRESET.1=RGB,0.51.141;&lt;cr&gt;</b>	This sets their Nautical Blue into <b>PRESET(1)</b> in the RGB color space
2	<b>#2.1.1.LED.PRESET.2=RGB,198.12.48;&lt;cr&gt;</b>	This sets their Red into <b>PRESET(2)</b> in the RGB color space
3	<b>#2.1.1.LED.PRESET.3=RGB,240.240.240;&lt;cr&gt;</b>	This is a special <b>JumpToCommand</b> which is an alias for the color White ( <b>240.240.240</b> ).  <b>Note:</b> Basically when we first encounter a <b>240.240.240</b> in any stored PRESET, <b>we don't display it in</b> the color sequence but we use it as a marker to go back to <b>PRESET(1)</b> and start over again on the motion sequence.
<b>Optional entries to adjust the Sequence Rate and Dissolve Rate for EFFECT,1</b>		
4	<b>#2.1.1.LED.DISSOLVE.3=n;&lt;cr&gt;</b>	Enter this command with a number for “n” indicating the time of transition (in seconds) between each Preset color turning on

5	#2.1.1.LED.SEQRATE=m;<cr>	Enter this command with a number for "m" indicating the time (in seconds) each PRESET color is ON before transitioning to the next color in the sequence.
<b>Final entry to execute Effect, 1</b>		
	#2.1.1.LED=EFFECT,1;<cr>	This final command executes Effect(1) and "plays" those colors stored above including any optional dissolve rates and sequence rates.

**Case 2 (where there is a white color in the team’s logos-like the Dallas Cowboys)**













Under that logo would be a macro comprised of the following general steps:

Step	Actual programmed sequence for Buffalo Bills (see below for RGB colors for team)	Notes
1	#2.1.1.LED.PRESET.1=13.37.76;<cr>	This sets their Navy into <b>PRESET(1)</b> in the RGB color space
3	#2.1.1.LED.PRESET.2=198.12.48;<cr>	This sets their Silver into <b>PRESET(2)</b> in the RGB color space
5	#2.1.1.LED.PRESET.3=240.240.239;<cr>	This sets their White offset by just digit (so it is not seen as a white by our system) into <b>PRESET(3)</b> in the RGB color space
6	#2.1.1.LED.PRESET.4=240.240.240;<cr>	This is a special jump to command which is really a white, but when our Effect sequence sees a white it doesn't display it but just jumps back to Step 1 ( <b>Preset1</b> ). This step inserts the <b>JumpToCommand</b> into <b>PRESET(4)</b>
<b>Optional entries to adjust the Sequence Rate and Dissolve Rate for EFFECT, 1</b>		
8	#2.1.1.LED.DISSOLVE.3=n;<cr>	Enter this command with a number for "n" indicating the time of transition between each Preset color turning on
9	#2.1.1.LED.SEQRATE=m;<cr>	Enter this command with a number for "m" indicating the time each Preset color is ON before transitioning to the next color in the sequence.
<b>Final entry to execute Effect, 1</b>		
7	#2.1.1.LED=EFFECT,1;<cr>	This executes the <b>Effect(1)</b> command which then looks for entries in <b>PRESET(1)</b> to <b>PRESET(n)</b> where a 240.240.240 is <b>first</b> found and then disregards that 240.240.240 and simply performs a GOTO back to PRESET(1)

### Appendix Colors for some NFL Teams

	Arizona Cardinals	 RGB 155,39,67	 RGB 0,0,0	 RGB 240,240,240
	Atlanta Falcons	 RGB 189,13,24	 RGB 0,0,0	
	Baltimore Ravens	 RGB 40,3,83	 RGB 0,0,0	 RGB 208,178,64
	Buffalo Bills	 RGB 0,51,141	 RGB 198,12,48	
	Carolina Panthers	 RGB 0,136,206	 RGB 0,0,0	 RGB 165,172,175
	Chicago Bears	 RGB 3,32,47	 RGB 221,72,20	
	Cincinnati Bengals	 RGB 251,79,20	 RGB 0,0,0	
	Cleveland Browns	 RGB 254,60,0	 81,47,45	
	Dallas Cowboys	 RGB 13,37,76	 136,144,154	 RGB 240,240,240
	Kansas City Chiefs	 RGB 178,0,50	 RGB 242,200,0	

	Los Angeles Raiders	 RGB 196,200,203	 RGB 0,0,0	
	New York Giants	 RGB 25,47,107	 RGB 202,0,26	 RGB 162,170,173
	Tennessee Titans	 RGB 100,143,204	 RGB, 13,37,76	

Src: <http://teamcolorcodes.com/tennessee-titans-color-codes/>  
<https://www.thepaperframer.com/TeamColors.php?type=nfl>