CONVERGING SYSTEMS

# Integration Note

| | |
|---|---|
| Automation/Lighting Panel Manufacturer: | Lutron Electronics Co. Inc. |
| **Platforms:** | **HomeWorks QS** |
| Versions: | All Lutron setup software (Designer) V. 6.2.0 or later |
| Specific Profile/Driver Version: | No driver required from Lutron. e-Node version 2.9 or later. Pilot software 4.4 build 2 or later.<br>Note: newer functionality may require newer versions of e-Node firmware and/or Pilot software |
| Website location for profiles/drivers | No drivers required other than built-in functionality to e-Node device |
| Document Revision Date: | August 6, 2014 |

## OVERVIEW AND SUPPORTED FEATURES

The Lutron lighting systems, RadioRA2 and HomeworksQS support the Converging Systems' family of motor and LED lighting control products using the Converging Systems e-Node device. This integration note is customized for the **HomeWorks QS** platform. If you desire to develop compatibility with RadioRa2 systems, please see the separate Integration Note referencing the RadioRA2 system.

Integration of the Converging Systems' platforms are enabled from Lutron keypads, and Android and iOS devices both locally and remotely.

### CURRENT PROFILES SUPPORT THE FOLLOWING FEATURES

The following commands can be supported by installer entered commands into the Lutron setup page within the e-Node Ethernet device. Those commands which currently cannot be supported are grayed out.

**LED Lighting Commands**

| General Commands | HSB Color Space Command | RGB Color Space Command | Correlated Color Temperature (CCT) Command |
|---|---|---|---|
| ON | HUE_UP | RED,R | CCT_UP |

|  |  |  |  |
|---|---|---|---|
| OFF | HUE_DOWN | GREEN,G | CCT_DOWN |
| EFFECT,# | HUE,H | BLUE,B |  |
| STORE,# | SAT_UP | VALUE=R.G.B |  |
| RECALL,# | SAT_DOWN | PRESET.X=XXX.XXX.XXX |  |
| DISSOLVE.X=XX | SAT_S |  |  |
| SEQRATE=XX | FADE_UP |  |  |
|  | FADE_DOWN |  |  |
|  | SET,L |  |  |
|  | STOP |  |  |
|  | COLOR-H.S.L |  |  |
|  | PRESETH.X=XXX.XXX.XXX |  |  |
| **Special Commands for Lutron Home Control+ Application** |  |  |  |
|  | HUE<br>**Note:** slider position provides the trailing data |  | CCT<br>**Note**: slider position provides the trailing data |
|  | SET<br>**Note**: slider position provides the trailing data |  |  |
|  | FADE<br>**Note**: slider position provides the trailing data |  |  |
| **Bi-Directional Commands** |  |  |  |
|  | COLOR=? | VALUE=? |  |
|  | PRESETH.X=? | PRESET.X=? |  |

## Motor Commands

| **General Commands** |  |  |
|---|---|---|
| UP |  |  |
| DOWN |  |  |
| STOP |  |  |
| RETRACT |  |  |
| STORE,# |  |  |
| RECALL,# |  |  |
| PRESET.X=XX.XX |  |  |
|  |  |  |
| **Bi-Directional Commands** |  |  |
| STATUS=? |  |  |
| POSITION=? |  |  |

## CURRENT PROFILES DO NOT SUPPORT THE FOLLOWING FEATURES

Other than any features that are grayed out below, the following features of Converging Systems' motor and lighting control devices are currently unsupported.

| | |
|---|---|
| | |

Any feature not specifically notes as supported should be assumed to be unsupported

## WIRING DIAGRAM (for HomeWorks QS)

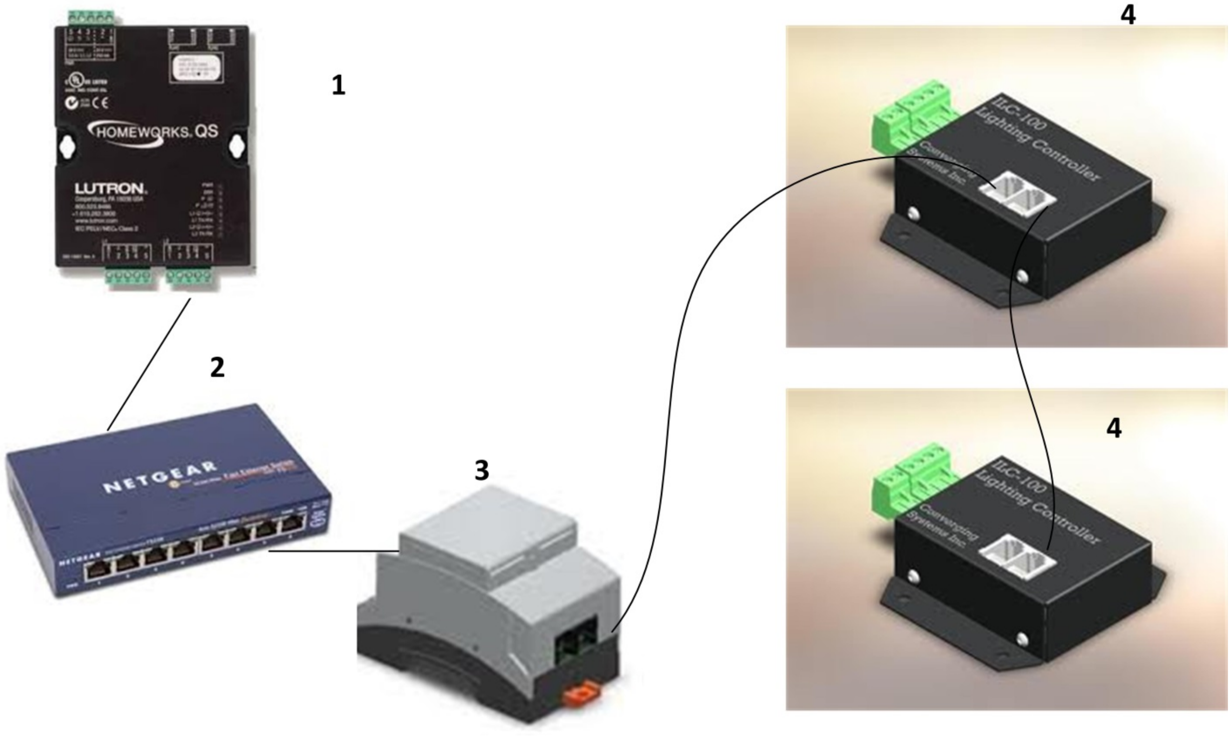


Figure 1

Wiring/Configuration Notes:

1. Maximum length of CS-Bus cabling from e-Node to the last ILC-100 using CAT5e or better cabling (and obeying the 1-1 pin-out requirements for the RJ-25-RJ25 cable) = 4000 feet
2. Maximum number of ILC-100 controllers and Converging Systems' keypads (if provided) that can exist on a single network connected to a single e-Node device = 254
3. Maximum number of e-Nodes that can exist on a Lutron RadioRA2 or HomeworksQS system = 254

## BILL OF MATERIALS (for HomeWorks QS)

| # | Device | Manufacturer | Part Number | Protocol | Connector Type | Notes |
|---|---|---|---|---|---|---|
| 1 | HomeWorks QS processor | Lutron | HQP6-2-120 HomeWorks QS Processor | Ethernet | various | |
| 2 | Network Switch | Various | Various | Ethernet | RJ-45 | |
| 3 | e-Node | Converging Systems | e-Node | Ethernet | RJ-45 (for Ethernet)<br>RJ-25 for local bus | |
| 4 | Lighting Controller (or Motor Controller) | Converging Systems | ILC-100 or IMC-100 or (Stewart BRIC) | CS-Bus protocol | RJ-25 for CS-Bus communication | Must terminate beginning and end of bus with 120 ohm resister on pins 3/4 |

# Theory of Operation

The Converging System network controller and one or more connected CS-Bus compatible devices (LED or projection screen motors) is made operational with the targeted Lutron system by following a few simple steps. No drivers or changes to Lutron equipment in general are required to establish communication with Converging Systems equipment, although you may wish to fine tune the button logic in your Lutron project to generate the type of output commands which will most effectively control the Converging Systems equipment.

The following steps summarize the steps that will be required to complete this process. Details for each Step can be found immediately after this section.

-**Step 1**. Complete your Lutron programming with Designer software. Establish a **Telnet Username** with **Telnet Password** that you can allocate to enable the Converging Systems network connection.

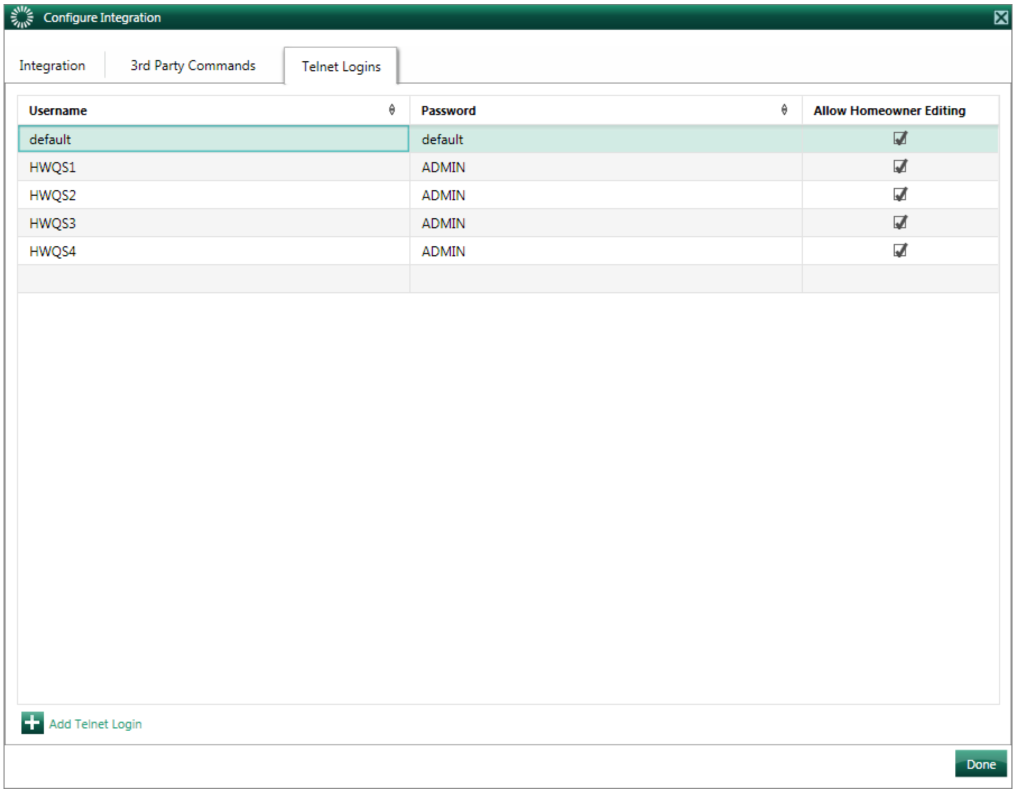
-**Step 2**. Customize any actual Lutron hardware devices or Lutron phantom keypads (usually for sliders on the Lutron app) with specific types of buttons optimized to achieve the results desired. (Lutron button logic varies depending upon how these buttons are programmed initially by the installer using the Lutron Designer software.)

-**Step 3**. Input a minimum amount of communication information in order to have the Converging Systems' e-Node processor establish a Telnet connection to the HomeWorks QS processor (**IP address** of Lutron processor, dedicated **Telnet Username** and **Telnet Password**).

-**Step 4**. Establish links between targeted Lutron button pushes on Lutron hardware devices (or buttons or sliders on the Lutron app) and the desired behavior of the Converging Systems' lighting or motor controller(s).

## STEP 1 (Telnet info and Lutron processor IP address)

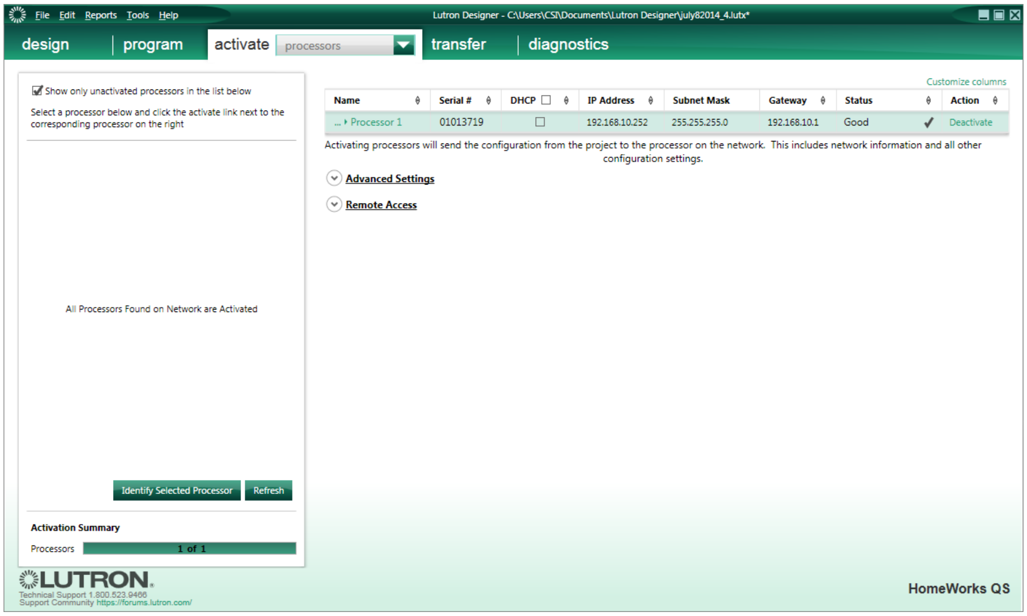
Program into your Lutron processor a ***<u>dedicated</u>*** **Telnet Username** and **Telnet Password** for a Telnet channel that can be dedicated to the Converging Systems' interface. Telnet channels cannot be shared, so if you wish to have the Lutron Home Control + app and the Converging System application running, it is necessary to establish two separate Telnet channels for these two operations to occur.  You can enter this Telnet information within the Lutron software **Tools/Configure Integration/Telnet Logins** tab as seen below in Figure 3.



| Username | Password | Allow Homeowner Editing |
|---|---|---|
| default | default | ☑ |
| HWQS1 | ADMIN | ☑ |
| HWQS2 | ADMIN | ☑ |
| HWQS3 | ADMIN | ☑ |
| HWQS4 | ADMIN | ☑ |

Figure 2

Also, take note of the IP address of your Lutron processor which is available from your **Activate Processor Tab**


File Edit Reports Tools Help
Lutron Designer - C:\Users\CSI\Documents\Lutron Designer\july82014_4.lutx*
design
program
activate
processors
transfer
diagnostics
Identify Selected Processor
Refresh
Customize columns
LUTRON
Technical Support 1.800.523.9466
Support Community https://forums.lutron.com/
HomeWorks QS

☑ Show only unactivated processors in the list below

Select a processor below and click the activate link next to the corresponding processor on the right

All Processors Found on Network are Activated

Activation Summary

Processors 1 of 1

| Name | Serial # | DHCP □ | IP Address | Subnet Mask | Gateway | Status | Action |
|---|---|---|---|---|---|---|---|
| ... ▸ Processor 1 | 01013719 | □ | 192.168.10.252 | 255.255.255.0 | 192.168.10.1 | Good ✓ | Deactivate |

Activating processors will send the configuration from the project to the processor on the network. This includes network information and all other configuration settings.

Advanced Settings

Remote Access

Figure 3

Now enter the dedicated **Telnet Username** and **Telnet Password** below along with the **IP address** of your Lutron processor into the attached spreadsheet in *Appendix 3* at the end of this Integration Note for future reference:

## STEP 2 (Lutron button-type programming)

Within HomeWorks QS, buttons can be created to behave in several discrete manners. Those relevant to our setup instructions are specified below. These button operations are summarized on the table below and described in further detail after the table. It is important to understand the discrete operations for how buttons behave, for the Converging Systems connected devices can only be programmed to respond to those output commands generated by Lutron. If those buttons do not generate the correct Lutron output codes, Converging Systems' products cannot properly respond. PERIOD.

| **Button Type** | **Operation** | **Lutron system software output** | | | | **Cases where this type of button is desirable** |
|---|---|---|---|---|---|---|
| | | **Press On** | **Release** | **Double Tap** | **Hold** | |
| Type 1A | Single | **"3"** | N/A | N/A | N/A | ALL OFF, RECALL (n), MOTOR STOP, |

| | | | | | | |
|---|---|---|---|---|---|---|
| | Action (“SA) | | | | | MOTOR UP (to end of travel),<br>MOTOR DOWN (to end of travel) |
| Type 1B | Single **with Hold (“SAH”)** | N/A | “**4**” | N/A | “**5**” | USER 1 (Recall and Store with Hold) |
| Type 1C | Single with Double Press option (“SADP”) | “**3**” | | “**6**” | | LED ON/OFF (On with a single press, OFF with a double press) |
| Type 2 | Dual Action (“DA”) | “**3**” | “**4**” | N/A | N/A | SAT+/-, HUE+/-, CCT+/-, MOTOR JOG UP, MOTOR JOG DOWN |
| Type 3 | Toggle (“TG”) | “**3**” | N/A | N/A | N/A | **Note**: this Toggle command is not particularly useful for most CS-Bus operations, because for most implementations the Lutron system sends the same command once on button press, and then again upon button release (the LED on the Lutron button goes ON after the first button press and then goes OFF after the second button press to indicate the light has been turned off). |
| Type 4 | RAISE/ LOWER (“RAISE”) (“LOWER”) | “**3**” | “**4**” | N/A | N/A | FADE UP, or FADE DOWN<br>(or MOTOR UP, or MOTOR DOWN) |

**Type 1A- Single Action (“SA”)**. This is standard and most common operation that can be used for discrete operations such as ALL OFF, ALL ON, MOTOR STOP  In this case the Lutron system generates a “**3**” from its processor, and no other output strings are generated.

**Type 1B- Single Action with Hold (“SAH”)**. This is a useful derivative case from the standard Single Action operation.  This is most useful where you desire a single button to both select a previously stored Color or Motor position, but with the additional functionality, that an extended hold will transmit a separate software string that can be utilized by CS-BUS to invoke a STORE command.  In this case the Lutron system generates a “**4**” from its processor upon a Button Press/Release, a “**5**” upon a Hold.

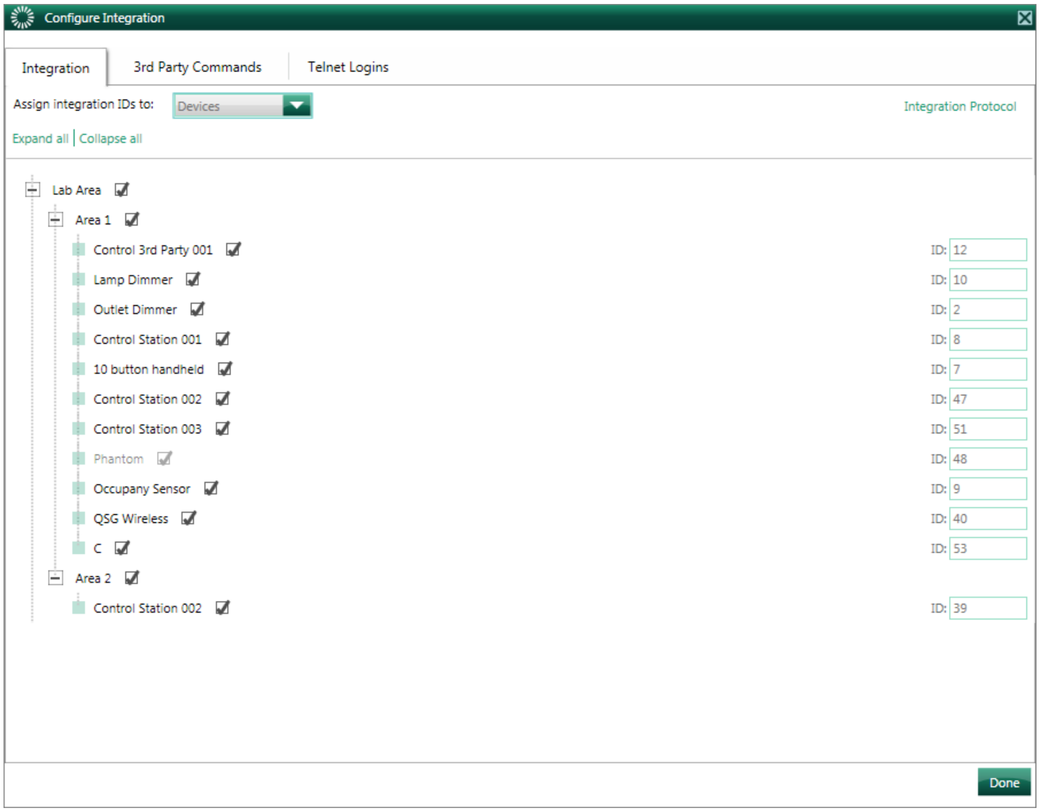
**Type 1C- Single Action with Double Press Option (“SADP”)**. This is a useful derivative case from the standard Single Action operation.  This is most useful where you desire a single button to turn ON lighting elements with a single press, as well as turn OFF lighting elements with a double press. In this case the Lutron system generates a “**3**” from its processor upon a single Button Press, a “**6**” upon a double Button Press.

**Type 2-Dual Action (“DA”)**. This is a useful selection for selections such as Hue UP, Hue DOWN, Saturation UP, Saturation DOWN, Color Temperature UP, and Color Temperature DOWN. This is also useful for MOTOR JOG operations where you actually wish to hold the button until you want the motor to stop, at which point you would release the button. In this case the Lutron system generates a “**3**” from its processor upon a Button Press, a “**4**” upon a quick Button Release.

**Type 3-Toggle (“TG”)**. This button type is presented for completeness but may not serve any particular function for most CS-Bus operations. This button logic is primarily targeted for Lutron keypads and touchscreen keypads which toggles ON the on-board LED or touchscreen indicator upon the first button push, and then turns OFF that same indicator on the subsequent button push.  In this case the Lutron system generates a “**3**” from its processor upon a Button Press, and then generates a “3” upon the second Button Press as well. CS-Bus devices are looking for unique subsequent software strings for each new activity only (e.g. a “**4**” following a “**3**” for instance, or a “**5**” following a “**3**”) and therefore receiving the same software string (e.g. a “**3**” following a “**3**”) for a second button push operation is not applicable here.

**Type 4- Special FADE (“RAISE” “LOWER”)**. Within HomeWorks QS, these are the only two buttons that have a different operation. These buttons are reserved for FADE type operations. When these buttons are pressed, the Lutron system generates a “**3**” but when these buttons are released, there is the special case of a “**4**” being generated. This logic is particular good only for our FADE UP and FADE DOWN LED commands which would provide a STOP command when the button is released. Alternatively, these buttons could also be used similarly to Type 2 button for Motor operation.

**Required Action (Potentially)**. Now with this information in mind, update any of your Lutron Designer programming for each button programmed that you wish to trigger a Converging Systems’ event in a unique way, if required. Also, take note under **Tools/ Configure Integration/Integration** seen in Figure 4 can be found the **Integration IDs** that will be used later in this Integration Note to link Lutron button pushes with Converging Systems’ invoked CS-Bus commands.



Configure Integration

Integration | 3rd Party Commands | Telnet Logins

Assign integration IDs to: Devices

Integration Protocol

Expand all | Collapse all

- Lab Area ☑
  - Area 1 ☑
    - Control 3rd Party 001 ☑ ID: 12
    - Lamp Dimmer ☑ ID: 10
    - Outlet Dimmer ☑ ID: 2
    - Control Station 001 ☑ ID: 8
    - 10 button handheld ☑ ID: 7
    - Control Station 002 ☑ ID: 47
    - Control Station 003 ☑ ID: 51
    - Phantom ☑ ID: 48
    - Occupany Sensor ☑ ID: 9
    - QSG Wireless ☑ ID: 40
    - C ☑ ID: 53
  - Area 2 ☑
    - Control Station 002 ☑ ID: 39

Figure 4

## STEP 3 (e-Node communication setup for Lutron/e-Node)

You will next need to enter several communication parameters within the e-Node Pilot application as part of the e-Node. These parameters are summarized in the below table.

| Step | Setting | Choices Available within the Pilot application |
|---|---|---|
| 1 | IP Address (of e-Node) | It is recommended that the e-Node be set to static addressing in this environment.<br>In order to change the factory default for the e-Node which is Dynamic addressing to Static addressing, follow these steps within e-node Pilot<br>-change **Static_IP** address to desired setting<br>-change **Gateway_ADD** to desired setting<br>-finally, change **DHCP** to **DISABLED** and hit **RESTART** |
| 2 | Telnet Operation | -Go to the **Lutron** tab and **Enable Telnet**<br>-Go to the **Telnet** tab and type in the address of the Lutron processor within the **Adapter_IP** entry. Within same page, make sure the **LOGIN** is set to **ENABLE (alias for Telnet client Enabled)** |

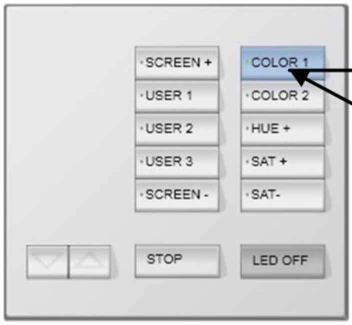
|  |  |  |
|---|---|---|
| 3 | Additional Lutron Processor Communication Information | -Go to the **Lutron** tab and enter the particular type of Lutron processor to which you would like to connect. For this application note, select **HOMEWORKS** for HOMEWORKS QS<br>-Under the **CONNECTION** tab, select ENABLE (for Telnet client)<br>-On the same page for the **LOGIN** entry, select the **Telnet username** that you secured above and hit the ENTER key to save.<br>-On the same page for the **PASSWORD** entry, select the **Telnet password** that you secured above and hit the ENTER key to save. |
| 4 | Restart Enode to Save Settings | -**Hit the RESTART button which reboots the e-Node and saves these new settings to the new defaults.** |

## STEP 4 (CSI CS bus commands linkage to Lutron button pushes)

The final step is to enter within the spreadsheet in *Appendix 3* at the end of this ***Integration Note*** data relating to the targeted (i) Lutron button push, (ii) with its particular Integration ID, (ii) with its particular type of button push (“3”, “4”, “5”,”6,” etc.) , and (iv) the desired Converging Systems CS-Bus command that will be triggered onto the CS-Bus when a the specific Lutron button push is encountered by the e-Node. Finally after this is done, a subset of that data can be entered into the e-Node Pilot application. **Although this seems complex, it is very quick to do and usually takes just a few minutes to complete an entire project.**

**Flowchart**. The following flowchart shows that if a Lutron identified button (that is to say, a button with a known Device ID, a known button number, and a known mode of operation output string, such as a “3”, a “4”, a “5” or a “6”) is activated, the software logic within the e-Node is able to translate that button push into a compatible CS-Bus command that can be directed to any CS-Bus compatible controller on that CS-Bus. In the below example, the noted button push generates a Telnet string of **~DEVICE,5,1,3** which is then translated by the e-Node to a CS-BUS command (entered by the programmer) as a **#2.7.1.LED=RECALL, 1** (in our example). This command causes the targeted Lutron button push to cause the Converging Systems lighting controller to turn on the factory programmed RED or Recall 1 color setting.

## Theory of Operation- From Lutron Button Push to Converging Systems Controller Operation

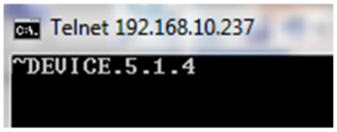


Device ID: 5
Button: 1
Operation: 4

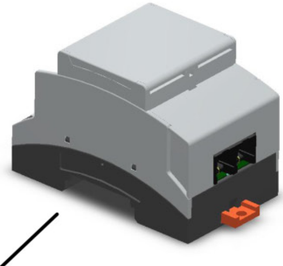
HomeWorks QS keypad

HomeWorks QS Processsor

Lutron command on bus

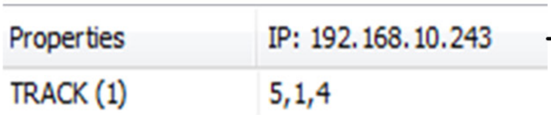


```
~DEVICE.5.1.4
```



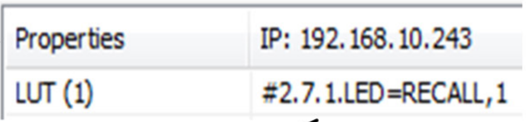
IP Address of e-Node: 192.168.10,243

E-Node Pilot entry for "Track (1)" data corresponding to particular Lutron button push of:



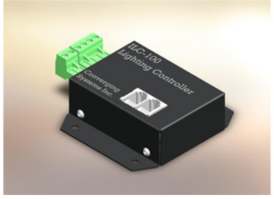
| Properties | IP: 192.168.10.243 |
|---|---|
| TRACK (1) | 5,1,4 |

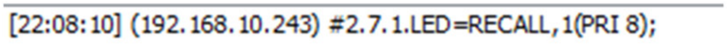
E-Node Transmits this Look Up Table (LUT-1) command referencing Track(1)



| Properties | IP: 192.168.10.243 |
|---|---|
| LUT (1) | #2.7.1.LED=RECALL,1 |

E-Node Transmits this command to CS-Bus compatible device





```
[22:08:10] (192.168.10.243) #2.7.1.LED=RECALL,1(PRI 8);
```

Figure 5

## e-Node "Lutron Tab" Programming.

The above diagram details the logical process that occurs from the point in time that a Lutron keypad button is pressed until a linked Converging Systems command is issued. From this information, you can determine exactly want operations you wish to cross-connect between Lutron and Converging Systems. You will enter that data in the form shown in the example below into the e-Node Pilot application under the **LUTRON** tab (again using the spreadsheet provided in *Appendix 3* for easy of data entry) and then from that spreadsheet, **you can now can easily transcribe that data directly into the e-Node Pilot application within the LUTRON tab**. Pay particular attention to the use of **PERIODS** (rather than commas) for Converging Systems' **Zone/Group/Node** addresses, and the use of **COMMAS for the TRACK command** and make sure for each entry where there is a **TRACK** entered, the associated **LUT** is populated.

**Note**: There is limited error-handling within the e-Node PILOT application, so must check your work carefully, and if all the data entries are valid, immediately after you populate the fields, your e-Node will be able to make the proper translations and your system should be operational.

## Programming Note on Lutron Button LED Logic

HomeWorks QS offers a wealth of programming options to the dealer, particular with respect to the operation and logic of on-button Lutron LED indicators. Although it is beyond the scope of this Integration Note to provide adequate background on how the Lutron LED logic operates, we have provided specific field-tested suggestions for how the specific buttons incorporated within our example might be programmed. We have chosen to utilize two variables (one for lighting called the ***ENODE***, and one for motor control called ***SCREEN***). Within each of those custom variables (which we have created) we have added a number of discrete states (documented under "Variables") that can be assigned to each button action to enable the LED Logic to perform as expected.  Linking these variables as documented below to each button, assures the on-button LEDs will (i) turn on when  presets are selected, (ii) turn OFF when an OFF is selected, and (ii) momentary operate in the case of increment or decrement of a lighting level or motor jog. You may find this example helpful but you may also have your own tricks that you may wish to integrate separately. See the columns marked "LED Logic" and "Variable" for detailed settings here.

### Note on Testing/Troubleshooting

We highly recommend launching the e-Node Pilot application and selecting the VIEW TRAFFIC window to make sure the proper commands that have been programmed can be seen on the CS-BUS. We can almost guarantee that if there is no appropriate CS-BUS command appearing in the VIEW TRAFFIC window, that the TRACK and LUT entries were not properly entered. You can also launch your computer's TELNET application, to verify the expected Lutron command strings are appearing on the IP bus. If those commands are NOT appearing on the IP bus, then there is no way the Converging Systems' e-Node can do its work.

### Programming Example-Case A

Following is a simple example all of the button programming on the specified 10-button HomeWorks QS keypad (HQR-T10RL). This example assumes that all buttons have been programmed using an appropriate **Type 1A (SA)**, **Type 1B (SAH),** **Type 1C (SADP)**, **Type 2 (DA),** **Type 3 (TG)** or **Type 4 (FADE)** button type for each button to be programmed. Buttons within the table are color-coated as per the above button type (see pages 6-7 for more information on button types).

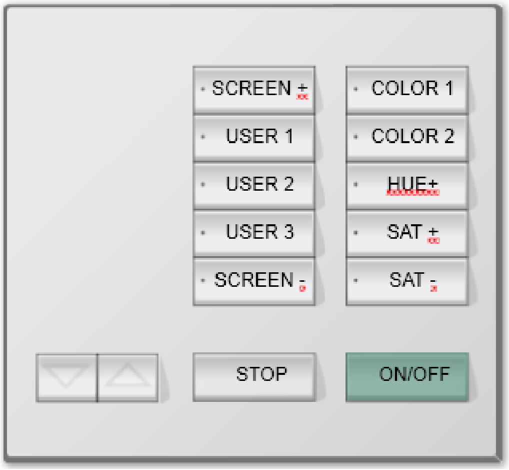


Figure 6

| Marked Lutron Button* | Button Type | LED Logic | Variables | Desired Action | Lutron output string | CS Bus resultant command |
|---|---|---|---|---|---|---|
| Color 1 | Type 1A-SA | Scene | $y^x$ Enode | -On button push, causes controller to go to | TRACK(**1**)=7,1,3 | LUT(**1**)#2.1.1.LED=RECALL,1 |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
|  |  |  | (State 1) | Preset 1 (Recall, 1). |  |  |
| **Color 2** | **Type 1A-SA** | Scene | $y^x$ Enode<br>(State 1) | -On button push, causes controller to go to Preset 2 (Recall, 2). | TRACK(**2**)=7,2,3 | LUT(**2**)#2.1.1.LED=RECALL,2 |
| **HUE UP** | **Type 2-DA** | **Room**<br>**Note: logic is set to Room** | $y^x$ Enode (un-affected)<br>$y^x$ Enode (un-affected) | -On button push, causes controller to go start incrementing the HUE variable.<br>-With a button release, the incrementing operation will immediately cease. | TRACK(**3**)=7,3,3<br>TRACK(**4**)=7,3,4 | LUT(**3**)#2.1.1.LED=HUE_UP<br>LUT(**4**)#2.1.1.LED=STOP |
| **SAT UP** | **Type 2-DA** | **Room**<br>**Note: logic is set to Room** | $y^x$ Enode (un-affected)<br>$y^x$ Enode (un-affected) | -On button push, causes controller to go start incrementing the SAT variable.<br>-With a button release, the incrementing operation will immediately cease. | TRACK(**5**)=7,4,3<br>TRACK(**6**)=7,4,4 | LUT(**5**)#2.1.1.LED=SAT_UP<br>LUT(**6**)#2.1.1.LED=STOP |
| **SAT DOWN** | **Type 2-DA** | **Room**<br>**Note: logic is set to Room** | $y^x$ Enode (un-affected)<br>$y^x$ Enode (un-affected) | -On button push, causes controller to go start incrementing the SAT variable.<br>-With a button release, the incrementing operation will immediately cease. | TRACK(**7**)=7,5,3<br>TRACK(**8**)=7,5,4 | LUT(**7**)#2.1.1.LED=SAT_DOWN<br>LUT(**8**)#2.1.1.LED=STOP |
| **LED ON/OFF** | **Type 1-SADP** | Scene | $y^x$ Enode (on)<br>-Enode (off) | -On button push, causes controller to turn any already OFF LEDS to turn previous ON state.<br>-On double press, causes controller to turn any already ON LEDS to turn OFF. | TRACK(**9**)=7,16,3<br>TRACK(**10**)=7,16,6 | LUT(**9**)#2.1.1.LED=ON<br>LUT(**10**)#2.1.1.LED=OFF |
| **SCREEN UP** | **Type 1-SA** | Scene | $y^x$ Screen (UP) | -On button push, causes connected projection screen to MOVE UP. **Note**: With Type 3 programming, a button release will not issue a STOP command | TRACK(**11**)=7,6,3 | LUT(**11**)#1.1.1.MOTOR=UP |
| **USER 1** | **Type 1-SAH** | Scene | $y^x$ Enode (on) | -On button push, causes LEDS to go to a USER 1 setting (in this case RECALL location #10) | TRACK(**12**)=7,7,4<br>TRACK(**13**)=7,7,5 | LUT(**12**)#2.1.1.LED=RECALL, 10<br>LUT(**13**)#2.1.1.LED=STORE,10 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | $y^x$ Enode (User 1) | -If the button is HELD for a preprogrammed amount of time, a command is sent to the CS-Bus device to STORE the current color setting into its specified memory location (i.e. memory location #10) | | |
| USER 2 | Type 1-SAH | Scene | $y^x$ Enode (on)<br>$y^x$ Enode (User 2) | -On button push, causes LEDS to go to a USER 1 setting (in this case RECALL location #10)<br>-If the button is HELD for a preprogrammed amount of time, a command is sent to the CS-Bus device to STORE the current color setting into its specified memory location (i.e. memory location #11) | TRACK(**14**)=7,8,4<br>TRACK(**15**)=7,8,5 | LUT(**14**)#2.1.1.LED=RECALL,11<br>LUT(**15**)#2.1.1.LED=STORE,11 |
| USER 3 | Type 1-SAH | Scene | $y^x$ Enode (on)<br>$y^x$ Enode (User 3) | -On button push, causes LEDS to go to a USER 1 setting (in this case RECALL location #10)<br>-If the button is HELD for a preprogrammed amount of time, a command is sent to the CS-Bus device to STORE the current color setting into its specified memory location (i.e. memory location #12) | TRACK(**16**)=7,9,4<br>TRACK(**17**)=7,9,5 | LUT(**16**)#2.1.1.LED=RECALL,12<br>LUT(**17**)#2.1.1.LED=STORE,12 |
| SCREEN DOWN | Type 1-SA | Scene | $y^x$ Screen (DOWN) | -On button push, causes connected projection screen to MOVE DOWN. **Note**: With Type 3 programming, a button release will not issue a STOP command | TRACK(**18**)=7,10,3 | LUT(**18**)#1.1.1.MOTOR=DOWN |
| STOP | Type 1-SA | Scene | $y^x$ Screen (STOP) | -On button push, causes connected projection screen to STOP, if it is currently moving. | TRACK(**19**)=7,17,3 | LUT(**19**)#1.1.1.MOTOR=STOP |
| FADE DOWN | Lower | | N/A | -On button push, FADES LEDS DOWN<br>-On button release, | TRACK(**20**)=7,24,3 | LUT(**20**)#2.1.1.LED=FADE_DOWN |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
|  |  |  |  | STOPS the fade process | TRACK(**21**)=7,24, 4 | LUT(**21**)#2.1.1.LED=STOP |
| **FADE UP** | **Raise** |  | N/A | -On button push, FADES LEDS UP<br>-On button release, STOPS the fade process | TRACK(**22**)=7,25, 3<br>TRACK(**23**)=7,25, 4 | LUT(**22**)#2.1.1.LED=FADE_UP<br>LUT(**23**)#2.1.1.LED=STOP |

**Programming Note**: There is no requirement for the order in which you add commands, with the only caveat being that the TRACK(n) and LUT(n) index numbers for the same operation must relate to each other. Specifically, TRACK(**1**) is associated with LUT(**1**), TRACK(**2**) is associated with LUT(**2**), TRACK(**3**) is associated with LUT(**3**), etc.. There are a total of 256 unique **TRACKS**, and therefore 256 associated **LUTS** that can be utilized by programmer. You can have duplicate entries from one TRACK(n) entry to another TRACK(n) entry to enable two different CS-BUS commands to be invoked from the same LUTRON button push, which is effectively an easy way to program a macro in this environment, conceptually.

## Appendix 1 Home Control+ Slider Application Notes

This section details Level 2 more sophisticated programming techniques. Several important topics will be covered:

-Lutron LED Button Logic handling

-Slider utilization within the Home Control+ application.

-Phantom Keypad Programming (for Home Control+ application)

-Additional Phantom Keypad Example

### A. Lutron LED Button Logical Lutron Button LED indicators

Within the examples detailed within this Integration Note are contained hints that can be utilized to make sure the Lutron LED logic performs as desired.  See Page 12 of this document (“**Programming Note on Lutron Button LED Logic”**) for additional information here.

Also, within the examples throughout this document can be found customized settings and button selection hints that can be used to program with Lutron Designer to insure the best LED keypad logic. Pay particular attention to the columns titled “LED Logic” and “Variable” for customized settings that can be utilized to enable logical Lutron LED keypad logic. Also, thoroughly review the button type discussion on pages 6-9, for information on picking the correct button type vis-à-vis its LED button control logic. Although our examples have been demonstrated to work, you may desire to try your own tricks as well.

### B. Standard Keypad Control—Standard Buttons with Operational Lutron Button LED indicators

Below can be seen a screen that reveals standard and actual hardware Lutron keypad previously programmed. Some of these buttons refer to the control of Converging System’s LED lighting control products, while other buttons refer to the control of Converging Systems’ motor (projection screen) control products.

**Programming Steps-Case A**. Actual programming steps to support Figure 7 below are detailed on pages 13-16 above. No changes are required to generate this specific screen within the Home Control + application. Your specific implementation may vary.

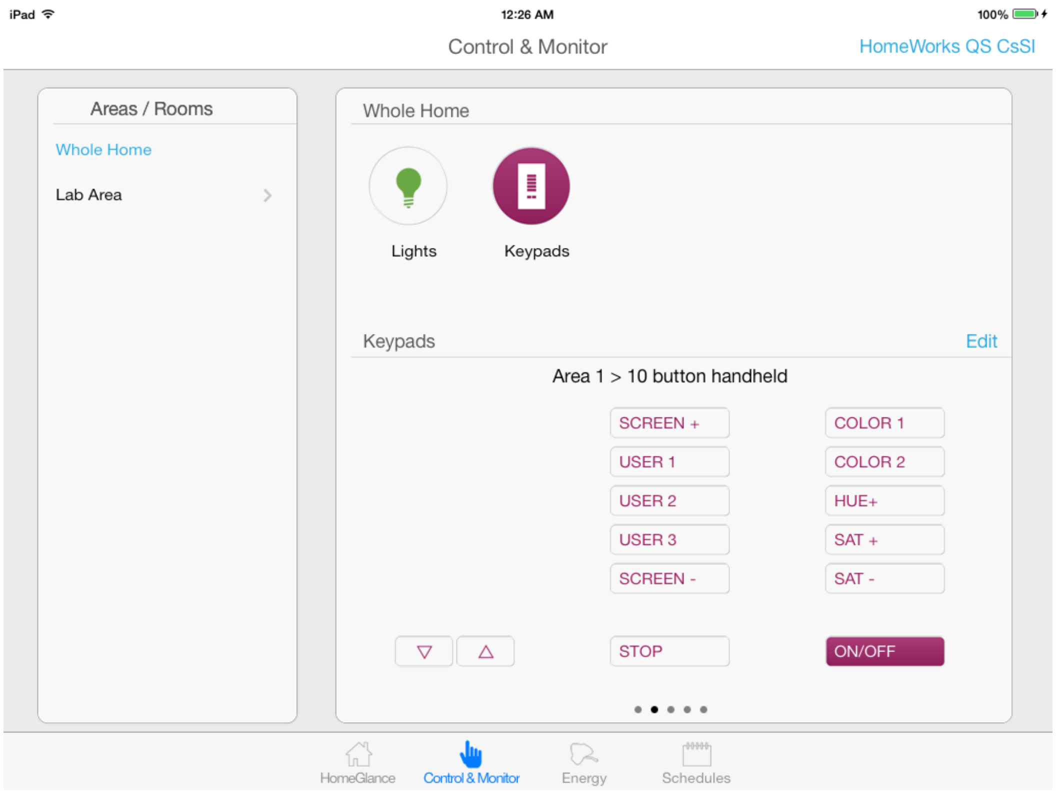


Figure 7

## C. Slider Control (Full color Hue, Saturation, Brightness as well as Correlated Color Temperature Control--CCT)

Below can be seen two screens that reveal (i) standard **Hue**, **Saturation**, and **Brightness** control as well (ii) **Correlated Color Temperature** (CCT) control. All of these controls are derived from the creation of set of Lutron phantom keypads with the Lutron Designer application but which utilize a derivative of the standard CS-Bus command set to interrelate with sliders only.

**Programming Steps-Case B**. Actual programming sets for the e-Node Pilot application are provided after the screens.

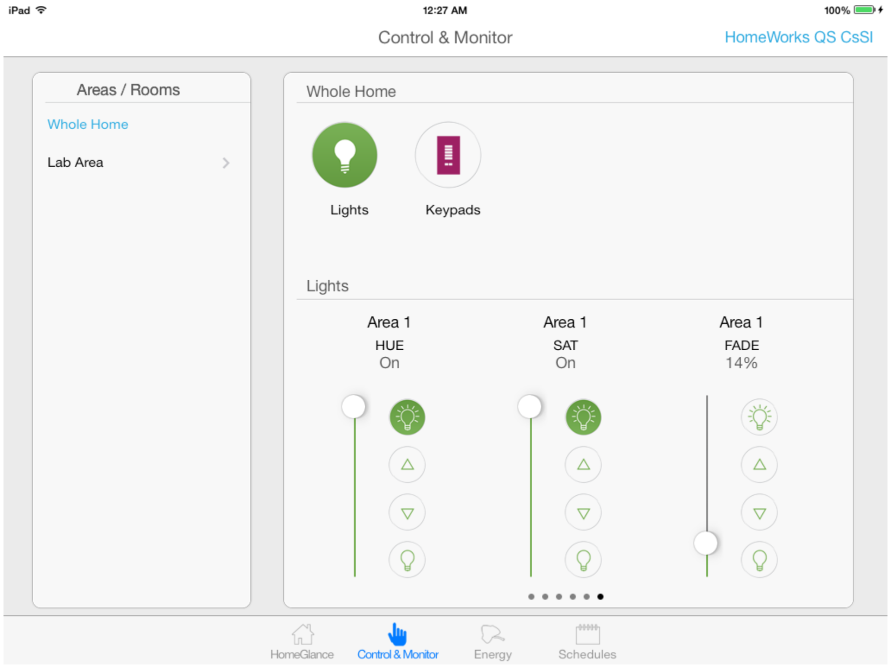


Figure 8

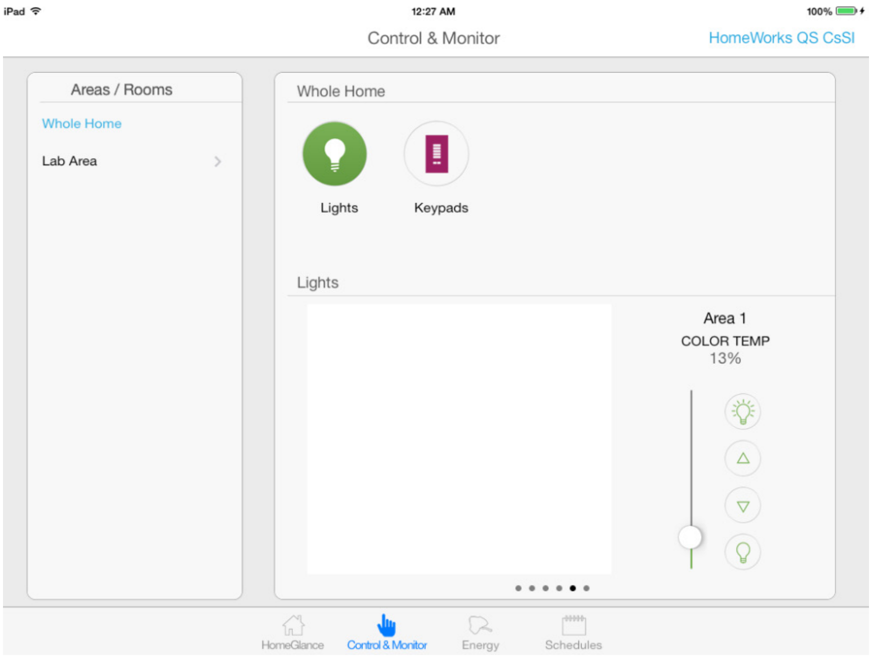


Figure 9

The above Slider can be easily integrated into the above project with the insertion of the following steps:

| Marked Lutron Button | Desired Action | Lutron output string entry | CS Bus resultant command |
|---|---|---|---|
| **Hue Slider** | -On movement of slider from 0% to 100%, Hue commands are transmitted to CS-Bus system.<br>**Note:** HUE of 0 or 100% equates to RED, while a HUE of 80 equates of GREEN, and a HUE of 160 equates to BLUE | TRACK(**50**)=23,1<br>**Note**: there is no trailing comma and third number in this slider case | LUT(**50**)#2.1.1.LED=HUE<br>**Note**: there is no trailing characters after the HUE command in this slider case |
| **Sat Slider** | -On movement of slider from 0% to 100%, SAT commands are transmitted to CS-Bus system.<br>**Note:** SAT of 0 is fully saturated (very white) while a SAT of 100 preserves the HUE of the original selected color | TRACK(**51**)=25,1<br>**Note**: there is no trailing comma and third number in this slider case | LUT(**51**)#2.1.1.LED=SAT<br>**Note**: there is no trailing characters after the SAT command in this slider case |
| **Brightness Slider** | -On movement of slider from 0% to 100%, Brightness (FADE) commands are transmitted to CS-Bus system.<br>**Note:** FADE of 0 is fully OFF (dark) while a FADE of 100 is fully ON | TRACK(**52**)=28,1<br>**Note**: there is no trailing comma and third number in this slider case | LUT(**52**)#2.1.1.LED=SET<br>**Note**: there is no trailing characters after the SET command in this slider case |
| **CCT Slider (Color Temperature)** | -On movement of slider from 0% to 100%, Correlated Color Temperature (CCT) commands are transmitted to CS-Bus system.<br>**Note:** CCT of 0% equates to a CCT of 1800K while CCT of 100% equates to a CCT of 7000K | TRACK(**53**)=21,1<br>**Note**: there is no trailing comma and third number in this slider case | LUT(**53**)#2.1.1.LED=CCT<br>**Note**: there is no trailing characters after the CCT command in this slider case |

## D. Additional Phantom Keypad Example

Below can be seen a screen that reveals **Correlated Color Temperature** (CCT) control. In this case, there is no real keypad, but only a representation of this keypad in the Home Control + application. All of these controls are derived from the creation of set of Lutron phantom keypads within the Lutron Designer application and with only simple “connection” within the e_Node Pilot application to make the virtual keypad actually perform.

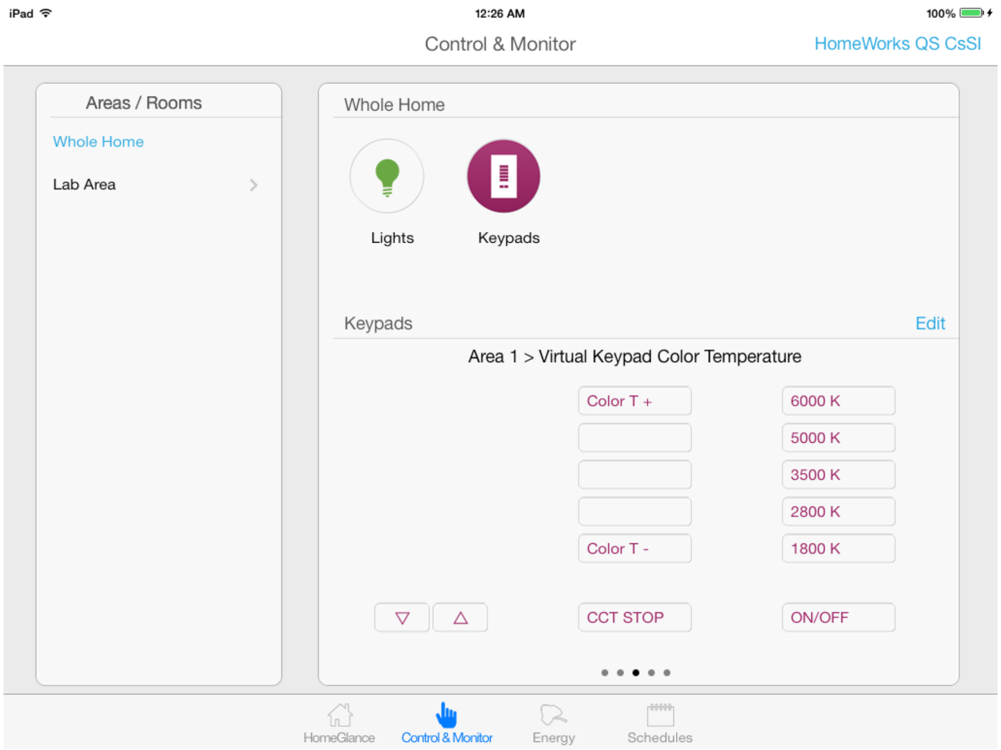


Figure 10

**Programming Steps-Case C**. Actual programming steps to support the virtual keypad shown in Figure 10 above are presented within this section.  Simply add in this case a virtual keypad into your Designer software and make a **Link Assignment** and label it appropriately. Determine the **Integration ID** for this virtual keypad and use that information below. Your specific implementation may vary.

| Marked Lutron Button* | Button Type | LED Logic | Variables | Desired Action | Lutron output string | CS Bus resultant command |
|---|---|---|---|---|---|---|
| 6000K | Type 1A-SA | Scene | $y^x$ Enode<br>(State 6000K) | -On button push, causes controller to go to CCT of 6000K. | TRACK(**60**)=57,1,3 | LUT(**60**)#2.1.1.LED=CCT,6000 |
| 5000K | Type 1A-SA | Scene | $y^x$ Enode<br>(State | -On button push, causes controller to go to CCT of 5000K | TRACK(**61**)=57,2,3 | LUT(**61**)#2.1.1.LED=CCT,5000 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | 5000K) | 2). | | |
| **3500K** | **Type 1A-SA** | Scene | $y^x$ Enode<br>(State 3500K) | -On button push, causes controller to go to CCT of 3500K | TRACK(**62**)=57,3,3 | LUT(**62**)#2.1.1.LED=CCT,3500 |
| **2800K** | **Type 1A-SA** | Scene | $y^x$ Enode<br>(State 2800K) | -On button push, causes controller to go to CCT of 2800K. | TRACK(**63**)=57,4,3 | LUT(**63**)#2.1.1.LED=CCT,2800 |
| **1800K** | **Type 1A-SA** | Scene | $y^x$ Enode<br>(State 1800K) | -On button push, causes controller to go to CCT of 1800K. | TRACK(**64**)=57,5,3 | LUT(**64**)#2.1.1.LED=CCT,1800 |
| **LED ON/OFF** | **Type 1-SADP** | Scene | $y^x$ Enode (on)<br>-Enode (off) | -On button push, causes controller to turn any already OFF LEDS to turn previous ON state.<br>-On double press, causes controller to turn any already ON LEDS to turn OFF. | TRACK(**65**)=57,16,3<br>TRACK(**66**)=57,16,6 | LUT(**65**)#2.1.1.LED=ON<br>LUT(**66**)#2.1.1.LED=OFF |
| **Color T +** | **Type 2-DA** | **Room**<br>**Note: logic is set to Room** | $y^x$ Enode (un-affected) | -On button push, causes controller to go start incrementing the HUE variable. | TRACK(**67**)=57,6,3 | LUT(**67**)#2.1.1.LED=CCT_UP |
| **Color T -** | **Type 2-DA** | **Room**<br>**Note: logic is set to Room** | $y^x$ Enode (un-affected) | -On button push, causes controller to go start incrementing the SAT variable. | TRACK(**68**)=57,10,3 | LUT(**68**)#2.1.1.LED=CCT_DOWN |
| **CCT STOP** | **Type 1-SA** | Scene | $y^x$ Screen (STOP) | -On button push, causes connected projection screen to STOP, if it is currently moving. | TRACK(**69**)=57,17,3 | LUT(**69**)#2.1.1.LED=STOP |
| **FADE DOWN** | **Lower** | | N/A | -On button push, FADES LEDS DOWN<br>-On button release, | TRACK(**20**)=57,24,3<br>TRACK(**21**)=57,24,4 | LUT(**20**)#2.1.1.LED=FADE_DOWN<br>LUT(**21**)#2.1.1.LED=STOP |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | STOPS the fade process | | |
| **FADE UP** | **Raise** | | N/A | -On button push, FADES LEDS UP<br>-On button release, STOPS the fade process | TRACK(**22**)=7,25,3<br>TRACK(**23**)=7,25,4 | LUT(**22**)#2.1.1.LED=FADE_UP<br>LUT(**23**)#2.1.1.LED=STOP |

## Appendix 2 Common Mistakes

# Common Mistakes

1. Forgetting to set turn on the Telnet Login under the **TELNET** page (to ENABLE). The Lutron processor does require a valid username and password, If **TELNET** is set to no login, the e-Node and the Lutron processors will fail to communicate.

2. Forgetting to input accurate Zone/Group/Nodes addresses within the LUTRON tab. The factory defaults will work with some systems but certainly, if your particular lighting or motor controllers do not function, check these addresses.

3. Forgetting to press the **RESTART** button within e-Node Pilot application after changes on the **NETWORK**, **TELNET**, or **LUTRON** pages are made.

4. Forgetting to match a valid LUT for each related TRACK within e-Node Pilot or the web application.

5. Forgetting to properly use COMMAS within the TRACK section or failure to properly use PERIODS within the LUT section of the e-Node Pilot application.

**Here is an example that works (commas with TRACK and periods with LUT):**

| | |
|---|---|
| TRACK(1) 5,1,4 | LUT(1) #2.1.1.LED=FADE_UP |

**Here is an example THAT WILL NOT WORK**

| | |
|---|---|
| TRACK(1) 5.1.4 | LUT(1) #2,1,1,LED=FADE_UP |

## Appendix 3 Spreadsheet

| | |
|---|---|
| **Telnet Username** | |
| **Telnet Password** | |
| **IP address of the Lutron primary processor** | ____ ____ ____, ____ ____ ____, ____ ____ ____, ____ ____ ____ |

(note **RED BOLD** **Column Entries** below are required for Programming)

| **Lutron button targeted for connection to Converging Systems LED or Motor operation** | | | | | | | **Desired outcome when Lutron button is pushed** | **Command that needs to be entered into e-Node Pilot application[1]** |
|---|---|---|---|---|---|---|---|---|
| Index | | Button Alias[1] | **Integration ID** | **Button Number** | **Button Logic** | | Descriptive Summary[2] | **Actual programming string[3]** |
| | | Recall 1 | (e.g.) 5• | (e.g) 1• | (e.g.) 3• | | (e.g.)Color goes to Recall 1 | (e.g.) #2.1.1.LED=RECALL,1 |
| (1) | | | | | | | | |
| (2) | | | | | | | | |
| (3) | | | | | | | | |
| (4) | | | | | | | | |
| (5) | | | | | | | | |
| (6) | | | | | | | | |
| (7) | | | | | | | | |
| (8) | | | | | | | | |
| (9) | | | | | | | | |
| (10) | | | | | | | | |
| (11) | | | | | | | | |
| (12) | | | | | | | | |
| (13) | | | | | | | | |
| (14) | | | | | | | | |
| (15) | | | | | | | | |
| (16) | | | | | | | | |
| (17) | | | | | | | | |
| (18) | | | | | | | | |
| (19) | | | | | | | | |
| (20) | | | | | | | | |
| (21) | | | | | | | | |
| (22) | | | | | | | | |
| (23) | | | | | | | | |
| (24) | | | | | | | | |

(Up to 256 indexes are available, please make additional copies of this table as needed.)

[1] Note: CS-Bus commands that can be utilized are described in a separate document entitled “Third-Party CS-Bus Device Driver Toolkit-Programmers Guide (DDK) which can be downloaded from http://www.convergingsystems.com/inres_programmingdesignkit.htm

[2] These entries are not required for programming but are only provided to assist in the programmer’s ease of project documentation.

[1] These programming strings assume a pre-programmed CS-Bus device with a Zone address of 2, a Group address of 1, and Node address of 1. The factory default for lighting controllers is Z.G.N= 2.1.0 while the defaults for motor controllers is Z.G.N=1.1.0. The e-Node Pilot application is required to change the factory default address to a unique address. The device address shown above as #2.1.1 can be any address from 1-254 per field. This address would need to have been programmed for the specific device being controller using the e-Node Pilot application also available from Converging Systems at http://www.convergingsystems.com/customerportal/1000/downloads.htm#anch4