HCA Technical Note: Generic Interfaces and Generic Server

Generic interface support allows you to integrate serial and network devices into your automation solution that HCA doesn't directly support. In this technical note examples are given for using a Lutron RadioRA controller with HCA.

Note: This technical note is not intended to replace the HCA User Guide Generic Interfaces appendix. It is a deeper examination of using HCA support to work with the Lutron interface. It is assumed that you have read the Generic Interfaces User Guide appendix first to familiarize yourself with HCA IP Interface support in general.

Integrating Home Control Assistant with Lutron RadioRA 2

The Lutron RadioRA 2 system is primarily used to control lighting in a home via the use of occupancy sensors, light switches and keypads, and motorized shades. The system is RF-based, and always includes one main controller, plus up to four repeaters. The main controller can cover an area of approximately 2500 sq. ft., as can each repeater. The repeaters connect to the main controller via either a wireless or wired connection.

The RadioRA 2 main controller can be integrated with other systems via either a built-in RS-232 port, or an Ethernet port. The Lutron software also loads the lighting system design via the controller's Ethernet port. Third party programs, such as HCA, can issue commands to, and receive status reports from, the Lutron system using the Ethernet port after establishing a Telnet session with the main controller.

While the Lutron system by itself provides control of natural and artificial lighting, including lighting scenes controlled by an astronomical clock, it doesn't have the capability to do the sophisticated conditional programmatic control that HCA does.
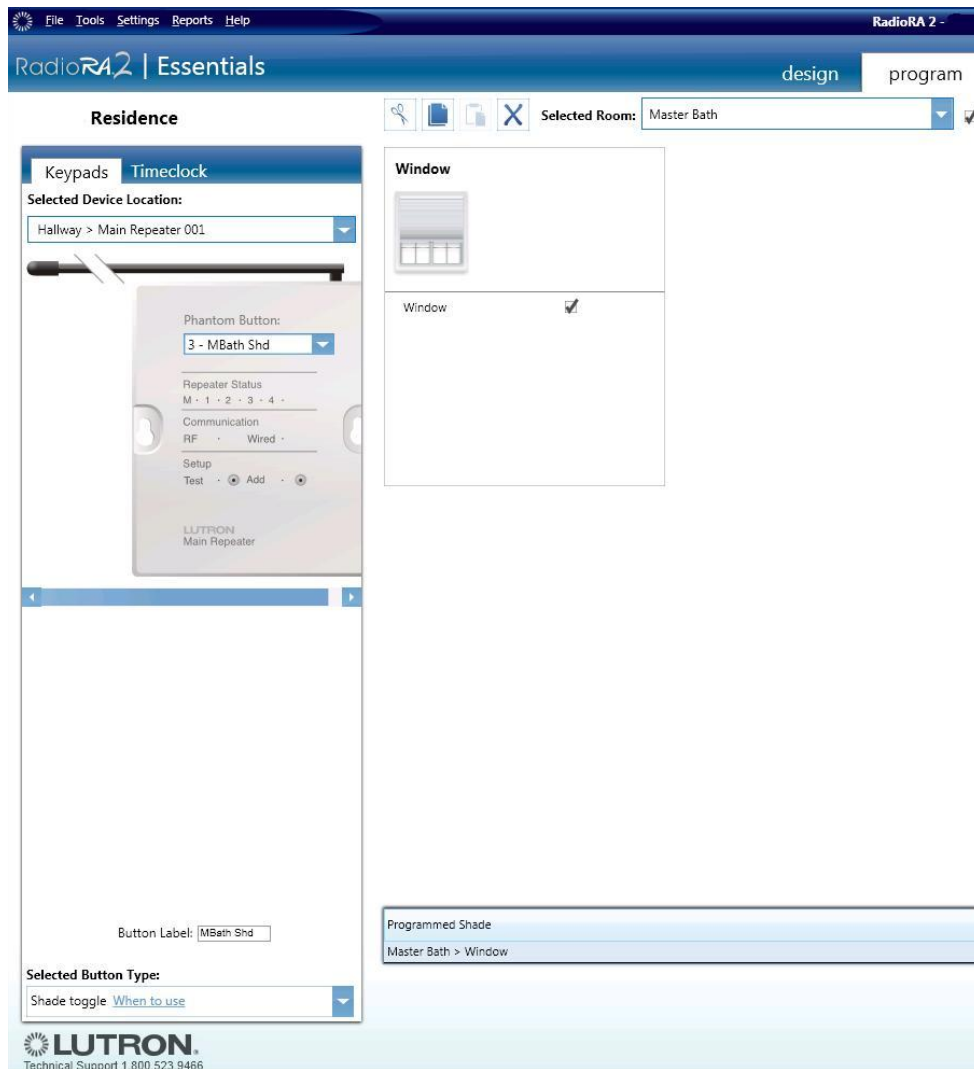
NOTE: While RadioRA 2 hardware can be readily purchased from a variety of sources online, in order to obtain the programming software that can load designs onto the controller, Lutron requires the completion of an online-training course. In order to be able to purchase Lutron cellular motorized shades, Lutron also requires an additional training course to be completed before you can gain access to their shade configuration tool (SCT). Both of these training requirements also require you to make certain one-time minimum purchases of Lutron RadioRA equipment.

Alternately, you could have a Lutron dealer install whatever Lutron devices you are interested in controlling from HCA. **Just be certain upfront that the dealer is willing to provide you with the integration report, described later, that the Lutron software generates.** This is so you know what Lutron devices use which ID#s. Also, the dealer would need to program buttons, such as the phantom buttons (described below) on the Lutron controller, to perform the functions you wanted so these functions would be accessible from HCA.

One of the most useful things a RadioRA 2 system can add to HCA is the ability to control motorized window shades—the focus of this Technical Note. Lutron makes easy-to-install radio-controlled cellular and roller shades that operate for several years on a set of batteries, meaning they can be installed with no new wiring. These shades are also available in a version using wired power.

Control of these shades can be accomplished using phantom buttons on the Lutron main controller, as well as direct commands to the shade devices. The phantom buttons can be programmed just like buttons on a physical Lutron keypad to trigger certain events, such as causing a window shade to open or close. There are 100 such programmable buttons on the main controller when using the base level Lutron software.

The Lutron software is used to create these phantom buttons and then program them into the main controller. In this screen image, a phantom button was created to control a shade in the Master Bath.

Lutron Integration Report

The Lutron software has the capability to generate a spreadsheet-like document called an "integration report". This report lists all Device IDs in the system, as well as the ID numbers for each button and LED on the device. This is a necessary report for integrating the Lutron system with HCA, as the above-mentioned IDs are used in the Lutron commands and messages.

Lutron Commands

The Lutron system has a simple command protocol for devices, such as multi-button controllers. These messages can be thought of as essentially Set (#DEVICE), Get (?DEVICE), and Report (~DEVICE). Lutron lighting devices (dimmers and shades), use OUTPUT commands (#OUTPUT-go to output level, ?OUTPUT-what is output level, and ~OUTPUT-report of output level). (There are also commands for Groups, Monitoring, and Timeclock. This document only describes the use of DEVICE and OUTPUT commands.)

The complete RadioRA 2 protocol is documented here:

http://www.lutron.com/TechnicalDocumentLibrary/040249.pdf

For example, if a phantom button on the main controller (DEVICE ID 1) has been programmed as a "shade toggle", this button can be "pressed" by HCA with the following command:

#DEVICE,1,3,3<CR><LF>

The message decodes as:

> **#DEVICE** - This means it is a command to the device. Commands are prefixed with "#"
>
> **1** - Integration ID, in this case, the main Lutron repeater with 100 "phantom" buttons. This ID is in the Lutron software's integration report.
>
> **3** – (Phantom) Button 3 (phantom buttons range from 1 to 100).
>
> **3** - Button action, 3 = Press

When this command is sent the Lutron controller responds back with a report that the command was received. The response is the command that was sent, with the "#" replaced with "~":

GNET> ~DEVICE,1,3,3<CR><LF>

(Note: the above response is also prefixed with "GNET>" by the controller. Sending a command to the controller can turn off this prefix. However, if the Lutron controller is re-booted, it will start sending the prefix again, so it is best to just leave it on always, and ignore it in messages.)

If the button "pressed" is programmed as a "shade toggle" for a shade with integration ID 20, and the shade went to full open position, the response received would be:

~OUTPUT,20,1,100.00<CR><LF>

This message decodes as:

> **~OUTPUT** - This means that this is an output report from a device. Responses are prefixed with "~".
>
> **20** - Integration ID. This ID is in the Lutron software's integration report.   While the main controller with the phantom buttons is ID=1, when a button is pressed on it that toggles a shade, the status report via the ~OUTPUT command contains the ID not of the main controller, but the ID for the shading device (in this case 20). The only status received from a main controller is the state of the phantom LED associated with a phantom button. For example, when a shade toggle is pressed, the LED is on during the time the shade is in motion.
>
> **1** - Field that follows is an output level.
>
> **100.00** = 100.0%. Means a light is at full brightness, or in the case of a shade, is fully open.

Note that a shade toggle can be linked to control multiple shades so that they will move in tandem. In that case, each shade will report its state with an ~OUTPUT report.

The output report is also received by HCA if an actual button - not one of the phantom buttons that HCA "pushes" but a button on a real Lutron keypad in the home - is pressed that causes the shade to open or close.

In addition, buttons have an LED that can indicate when the shade is moving, and phantom buttons have phantom LEDs. On the Lutron controller, phantom button IDs 1-100 have corresponding phantom LEDs 101-200. In the above #DEVICE example, when the button is first pressed, the Lutron controller also sends:

~DEVICE,1,103,9,1<CR><LF>

This message decodes as:

> **1** - Integration ID.
>
> **103** - Phantom LED ID for phantom button 3.
>
> **9** - LED state.
>
> **1** - LED ON.

Configuring HCA for the Lutron interface

To configure HCA to use the Lutron interface open the hardware configuration dialog and select as the type "Generic IP Interface".
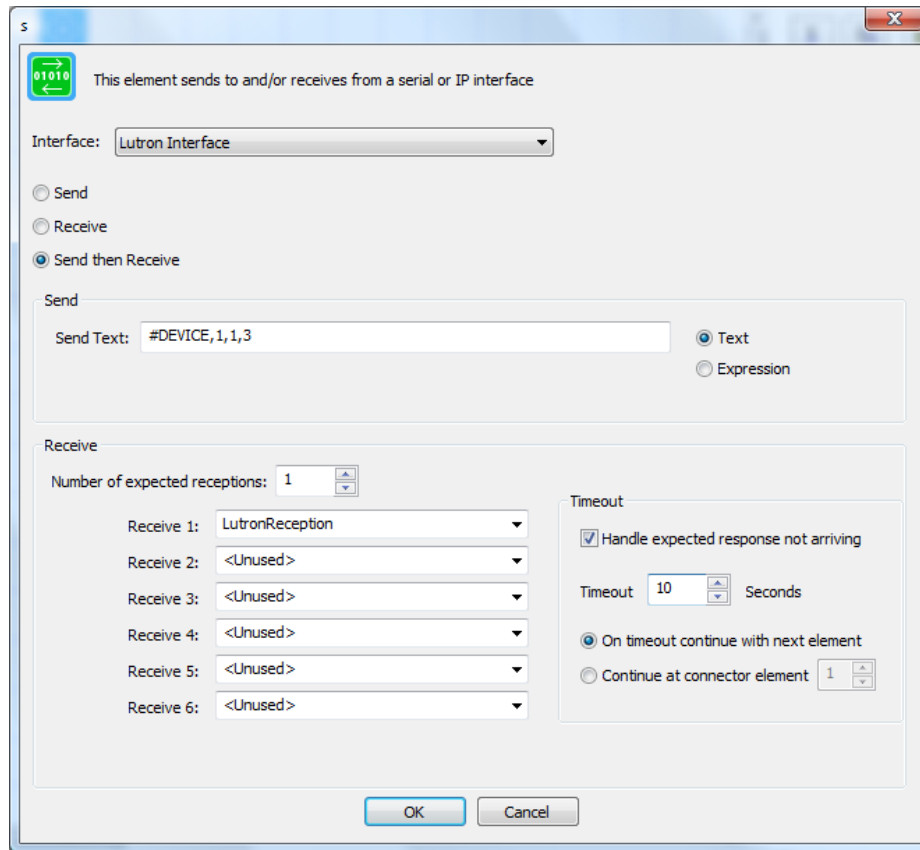
Press the *Configure* button to change the interface settings:

When using the Lutron interface, you must first, using the Lutron software, configure it to have an IP address on your internal network and enter the Telnet username and password (username = Lutron, password = integration) in the HCA interface configuration dialog. HCA takes care of the Telnet login upon connection.

Sending commands to the Lutron Interface

To send commands to the Lutron interface, the visual programming "Port I/O" element is used. To send the example command from above, these are the settings for that element:

The Lutron interface is selected by name – that name comes from the interface configuration. The action is to send a command and expect a response – "Send then Receive". Running the program containing this element will "push" the phantom button to raise or lower the shade.

The response to the command from the interface when received is saved in the LutronReception flag.

Auto Disconnect

The Lutron interface like other network addressed interfaces allows only a limited number of connections at a time. This can be a problem if HCA opens the connection and leaves it open forever. Doing that may prevent other users of that interface from connecting to it.

If the *Disconnect* option in the interface configuration dialog is enabled, then HCA connects to the interface when it needs to use it, and then leaves it open for the specified number of minutes and if not used again during that time then it disconnects. The next time it needs to use the interface then it reconnects.

If you use this option then you will not receive *Asynchronous* messages from the interface! Suppose you connect to the Lutron interface and send a command. HCA sends the message and receives a reply saying that the operation was completed. That is a *synchronous* message protocol - you send something

and it replies. The message from the interface (its reply) is always preceded by a message sent to it (your request).

Now suppose you have a physical keypad that controls the shade or light and someone pushes a button on that keypad. The keypad sends to HCA a message to report what occurred. This would be an *asynchronous* message. You didn't ask the Interface to do anything; it just sends you a message. The message you received (the report) isn't preceded by a message you sent to it. That is why this is an asynchronous message.

If HCA is not connected to the interface when the interface has something to report it will not get received since HCA doesn't have an open connection to it. If you want to receive these kinds of asynchronous messages – reports for actions that HCA didn't initiate – then you can't use the disconnect option.

Example of Controlling Lutron Shades from a UPB Keypad

Here's an example of integrating Lutron Sivoia QS motorized cellular shades with an HCA system that also incorporates UPB devices, specifically a UPB keypad. The button on the UPB keypad can be used like a button on a Lutron keypad to toggle the shade, with the LED on the UPB keypad tracking the status LED on a Lutron keypad button that was programmed as a "shade toggle". A shade toggle works as follows: if the shade is closed, pressing the button will start the shade moving up and the LED will turn on. If pressed again, the shade would stop, and the LED would go out. Pressing the button again would cause the shade to begin moving in the opposite direction, and the LED would be lit while the shade is in motion. Groups of shades, as well as a singles shade, can be controlled by a toggle. The association between a phantom button on the main controller and a group of shades is set using the Lutron RadioRA software.

The UPB keypad button would be programmed to send a UPB link command, which could be used to trigger an HCA program to send the appropriate Lutron command to the Lutron main controller. Here's an example of setting up a trigger using a UPB link command to do this:
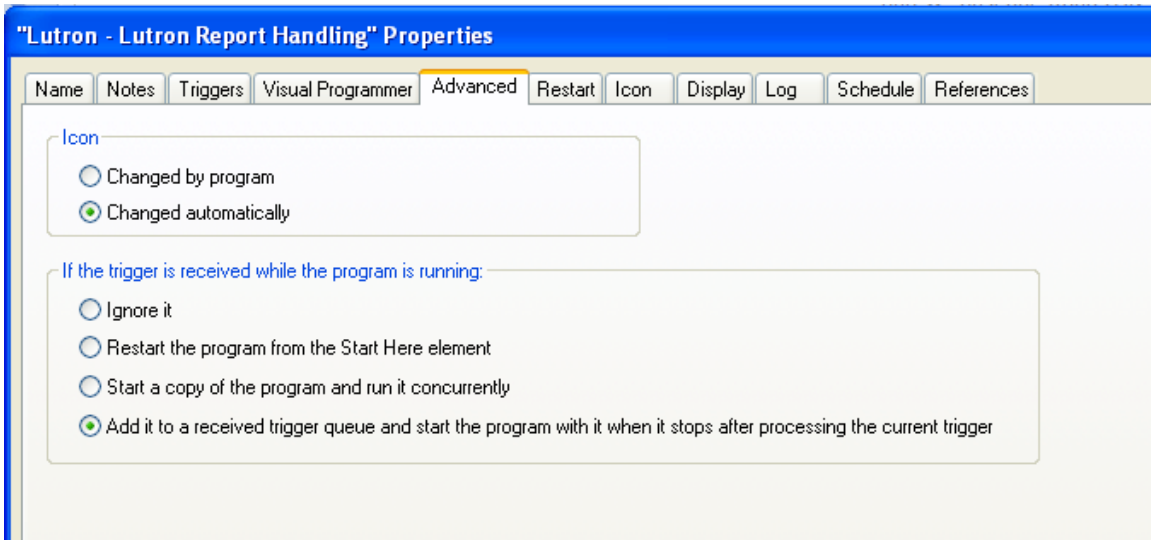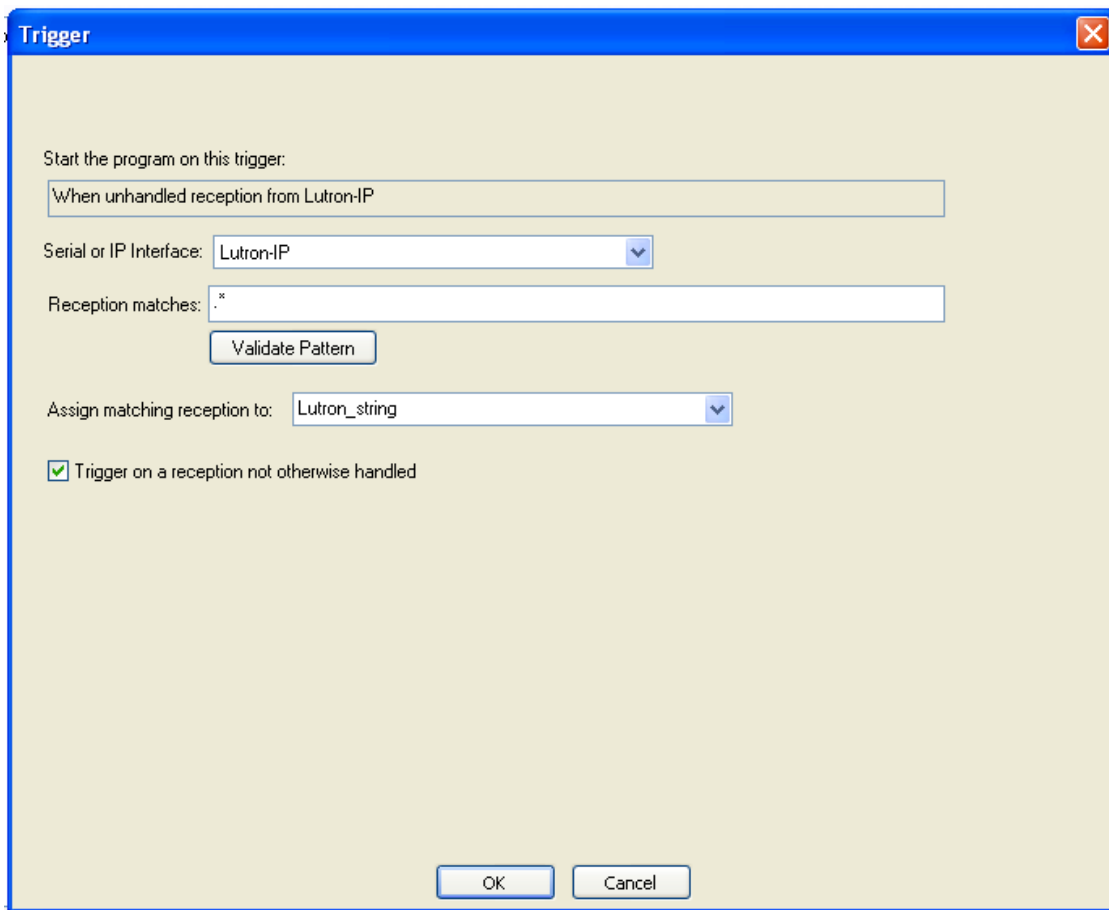
The Lutron controller would send back status messages for the Lutron button LED, which would be used to control the LED on the UPB keypad, as well as a message indicating the shade's position: open (100%), closed (0%), or somewhere in-between (e.g. 25%), which would occur if the button was pressed a second time before the shade(s) reached the fully open or closed position. The position of the shade, e.g. 100%=open, could be reflected in the title of the HCA program icon that controls the shade: e.g. "Kitchen Shades-Open" and the icon could also be changed to one showing a fully open window shade, using the appropriate HCA VP elements.

These Lutron status reports—for example window shade position—can trigger an HCA program, as previously mentioned. Because these messages are sent asynchronously and can occur at any time, the easiest way to deal with them all is via a single HCA program set up as a Lutron message handler. This handler can analyze the beginning text in the string, and determine the type of message. It can then parse the rest of the string to determine the message parameters. Because all flags (variables) in HCA are global, it is important that a received message in a flag cannot be overwritten by a subsequent message until the earlier message has been fully processed. HCA v12 has added the capability to queue messages that would trigger a program, so that they can be processed sequentially, and none will be lost.

Here's how this would be setup for the program handling Lutron controller messages.

Because all messages from the Lutron controller are going to be handled by one program, setting up the trigger is simple, as shown in this screen shot:

In the body of the triggered program you can use the HCA string functions in a *Compute* element to decode the message into its component parts.

Here are more example Lutron commands to control a group of shades:

To send three shades to fully open, these three commands could be sent sequentially to the controller:

#OUTPUT,23,1,100<CR><LF>

#OUTPUT,24,1,100<CR><LF>

#OUTPUT,25,1,100<CR><LF>

These decode as:

> **#OUTPUT** = Output command
>
> **23 or 24 or 25** = Device (Shades are 23, 24, and 25 in the Lutron integration report)
>
> **1** = "Set"
>
> **100** = 100% "brightness" or fully open.

To send shade the three shades to fully closed, these three commands would be sent to the controller:

#OUTPUT,23,1,0<CR><LF>

#OUTPUT,24,1,0<CR><LF>

#OUTPUT,25,1,0<CR><LF>

These decode as:

> **#OUTPUT** = Output command
>
> **23 or 24 or 25** = Device (Shades are 23, 24, and 25 in the Lutron integration report)
>
> **1** = "Set"
>
> **0** = 0% "brightness" or fully closed

The shades would report their status with ~OUTPUT messages.

Note that all three shades in the above example could be controlled by one command sent to a button set up as a toggle. However, you may not want to do this. Consider this example: You want the shades closed at sunset. During the day, someone closes the shades using a keypad. If you just send a shade toggle, the shades will open again. While you could track the shades state, and not send the toggle command if they are fully closed, sending explicit #OUTPUT commands to each shade will always result in the shade ending up in the desired state.

A shade could also be queried for its state by this command:

?OUTPUT,24,1<CR><LF>

Which would return a string as follows:

~OUTPUT,24,1,n<CR><LF>

Where n is the percent the shade is open, from 0% to 100%.