

Wheatstone Corporation Technical Documentation

Using Tallies On The SideBoard

A Guide To Activating OnAir Lights Or Other Tallies
According To SideBoard Channel On/Off Status



600 Industrial Drive
New Bern, NC 28562
252.638.7000
www.wheatstone.com

Rev 1.0 – June 1, 2015
Dick Webb

Table of Contents

1. Tally Basics.....	2
1.1 Lighting A Light.....	2
1.2 Coding The Logic.....	3
1.3 Wiring The Logic Port.....	3
2. The Details.....	3
2.1 Setting Up Associated Logic.....	3
2.2 Activating Logic On Utility Mixer Channels.....	5
2.3 Make The Necessary Crosspoint Connection.....	6

1. Tally Basics

It is often necessary to light some kind of indicator to indicate that a SideBoard Fader has been turned on. This situation is similar to lighting a tally when a normal control surface channel is turned on, for which an explanatory Wheatstone document already exists, but is dissimilar enough that it is worth explaining the entire process here.

Essentially the job will consist of setting up the necessary hardware to enable the external light to be lit, and then coding the logic to make that happen when the SideBoard fader is on.

We will first provide some additional details explaining what's involved, and then will show an example of how this can be done in a practical system situation..

1.1 Lighting A Light

Generally we will need to have an external light that can be appropriately interfaced with the Wheatstone WheatNet-IP system.

The first thing to consider is that the tally light must be something we can operate from a low voltage DC source, using low current. The activating output of the WheatNet-IP system will be a hardware logic port on a Blade in the system. This output port will provide a closure that can safely handle DC current on the order of 50 mA or less, and with a source DC voltage of 30 volts or less.

If your light requires a higher current, a higher DC voltage, or any amount of AC voltage, you will need to include an intermediate relay of some kind that can handle the voltage and current needs of your light on the output (contact) side, yet be activated by a DC voltage no higher than 30V with a current draw of no more than 50 mA on the input (coil) side.

Further, if such a relay is employed, you will need to include a snubber diode in the external wiring to prevent damage from the EMF generated when the relay coil is suddenly deenergized. It is up to the end user to choose an appropriate relay, if needed, and provide the proper wiring to activate it safely.

Beyond the requirements of the light and its powering method, the WheatNet-IP Blade logic port will provide a simple closure that will power the light, or the intervening relay if one is required, according to the status of system logic, as discussed next.

1.2 Coding The Logic

The logic required to ultimately activate the output logic port is not something that needs to be done in the SideBoard itself, or any scripting device in the WheatNet-IP system. It is all handled in the Wheatstone Navigator program.

You will need to know what logic port you will be using. Each hardware Blade in the system has 12 logic ports divided between two RJ-45 connectors on the back of the Blade. You will specifically need to know the Blade ID of the Blade and the port number (1 – 12) that you will be using.

1.3 Wiring The Logic Port

Please refer to the section of the *WheatNet-IP Audio Over IP Network Technical Manual* for details on connecting to the logic port.

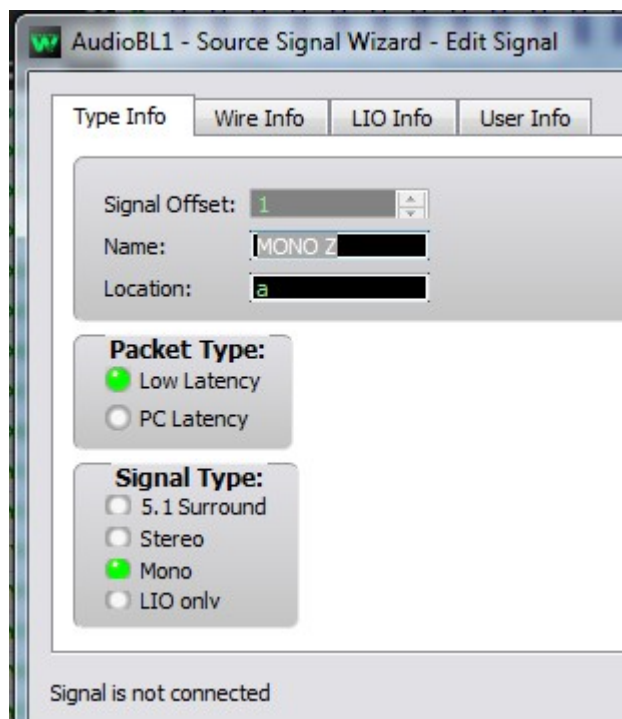
2. The Details

So here's where we dive in and look at the details of the task.

2.1 Setting Up Associated Logic

The logic required for this task uses what we call *Associated Logic*, which simply means we have programmed logic by associating it with a signal.

The signal, in this case, is the source that will be connected to the SideBoard fader, as indicated by an



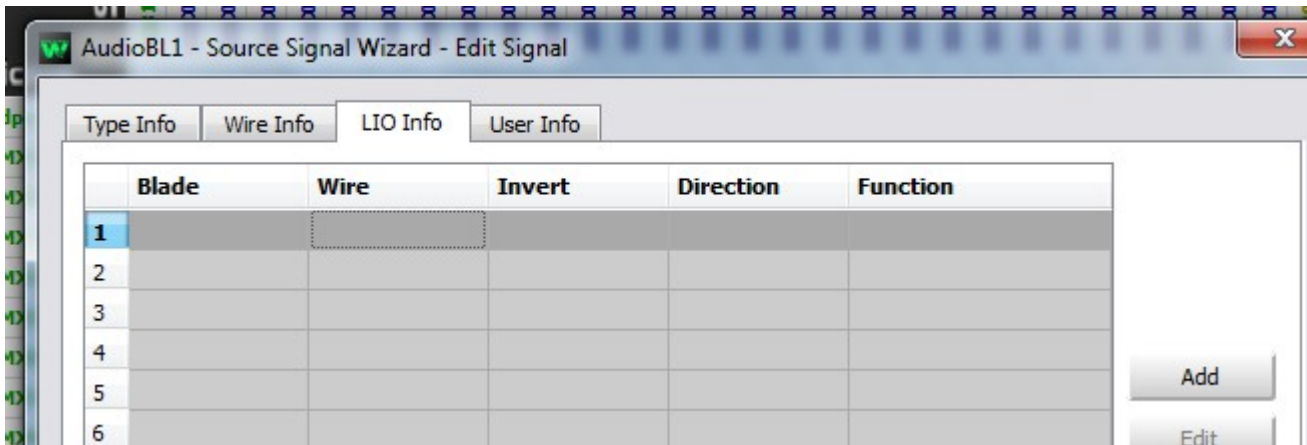
appropriate crosspoint indication on the **System Crosspoint** tab in Navigator. What may not be apparent is that the destination representing the SideBoard fader is, in truth, an input to the *Utility Mixer* that the SideBoard has been programmed to control.

It is important to realize that we edit or modify the source signal logic characteristics to do the programming, and we cannot edit or modify a signal while it has a crosspoint connection, so the source signal we are going to deal with has to have any crosspoint connection to a destination removed while the signal is being edited.

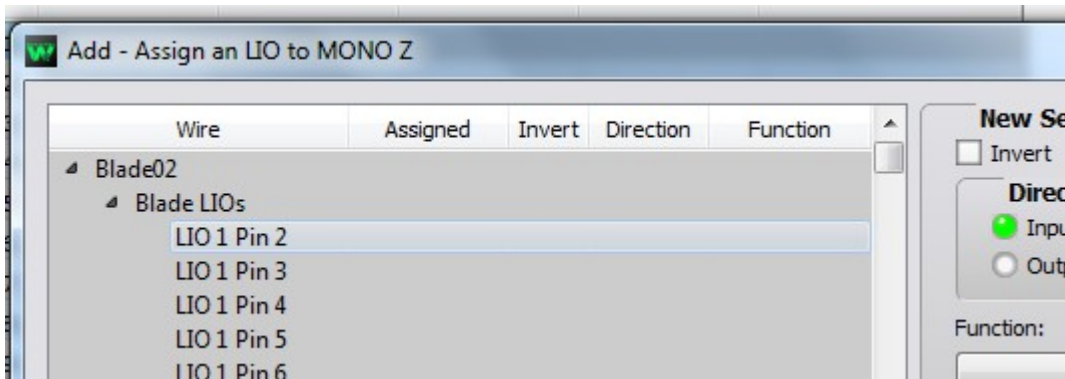
Find the desired source signal in the System Crosspoint matrix and right-click on the sources name. This will pop up a menu. On the popup, select *Modify Signal...* to bring up the . . . *Edit Signal* dialog.

Please note, if the message at the bottom of this window says “Signal is connected” you will need to close the dialog and break any crosspoints from that source to destinations, then bring up the dialog again.

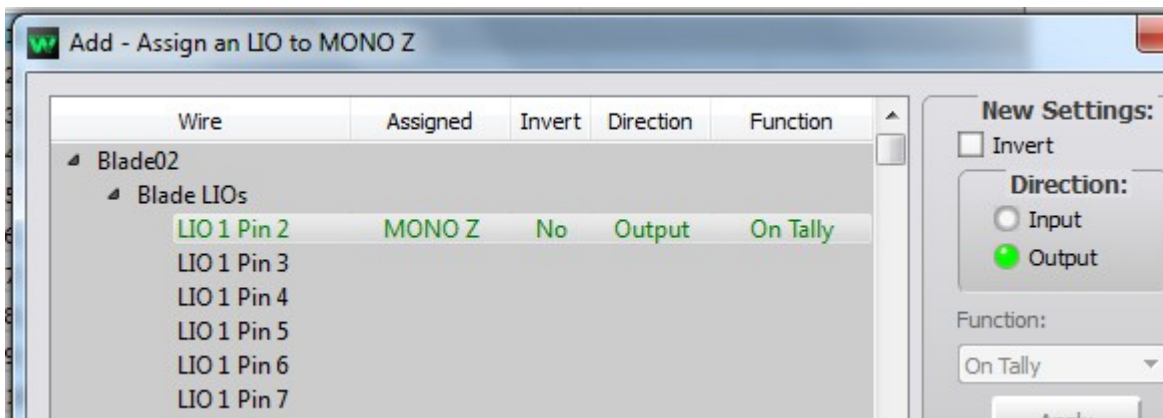
On the edit dialog, select the *LIO Info* tab.



Click **Add** to bring up the dialog to add the associated logic to the source:



Find the Blade that hosts the hardware logic port you will use, and expand the *Blade LIOs* section under that Blade to see the available hardware ports. Click on the row representing the hardware port you will be using to highlight it. For the logic to turn an external light on as a result of a fader channel being on, you will select a *Direction* of *Output* and a *Function* of *On Tally*. Click **Apply**.



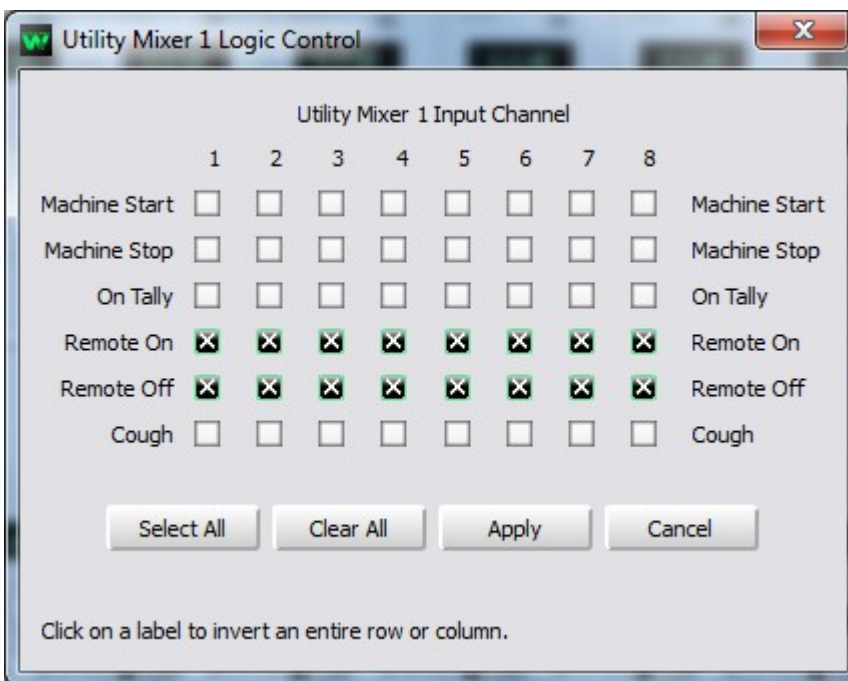
Click **Close**. Click **Finish**.

2.2 Activating Logic On Utility Mixer Channels



As indicated above, we are using the *On Tally* logic of the Utility Mixer channel to activate our external logic port. It is important to realize, however, that Utility Mixer fader logic is not all enabled by default. To see what we mean, select the Blade that hosts the Utility Mixer you are using in Navigator and click on the *Utility Mixer* tab. Look in the lower right corner of the tab and note the button labeled *LIO Ctrl* under the *MIXER 1* button.

Click the *LIO Ctrl* button to bring up the *Utility Mixer x Logic Control* (where *x* is 1 or 2, depending on which of the two Utility Mixers you are working with – our example uses Utility Mixer 1). If your



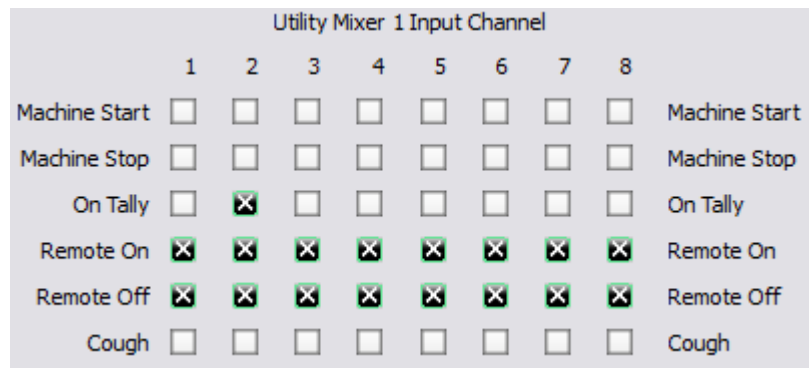
Utility Mixer logic has not yet been changed from default, it will look like the picture at left.

Note that, by default, the logic functions *Remote On* and *Remote Off* are enabled for all eight channels, while the remaining logic functions are disabled.

In order to use the *On Tally* function, you must enable it here by clicking the *On Tally* box for the desired channel. Optionally, it's OK to put checks in the *On Tally* check boxes for additional channels, up to and including all eight. In particular you would check all eight if you either

weren't sure which channel you were going to use, or wanted the freedom to have the logic work no matter what channel you have the source connected to. Naturally, you can go back later and change this programming if desired. Don't forget to click *Apply* once you have made your desired selection.

Here we have only checked the box for channel 2:



2.3 Make The Necessary Crosspoint Connection

The only thing left to do is to make the crosspoint connection between the audio source and the Utility Mixer channel corresponding to the SideBoard fader. As making crosspoints is a task you should be familiar with by now, we can bypass the addition of details here, other than to say that, since this may be your first time using Utility Mixer channels, Utility Mixer input channels are found as Destinations in the system, and exist in concert with the Blade housing the Utility Mixer in use. Default names are similar to BL02UMX1 through BL02UMX8 for the channels on MIXER 1, and BL02UMY1 through BL02UMY8 for the channels on MIXER 2. The prefix “BL02” in this case indicates the Utility Mixer is hosted on Blade 2 in the system.

Like other signals in the system, you can always change the signal names from their default.

