RIOT (BE2)

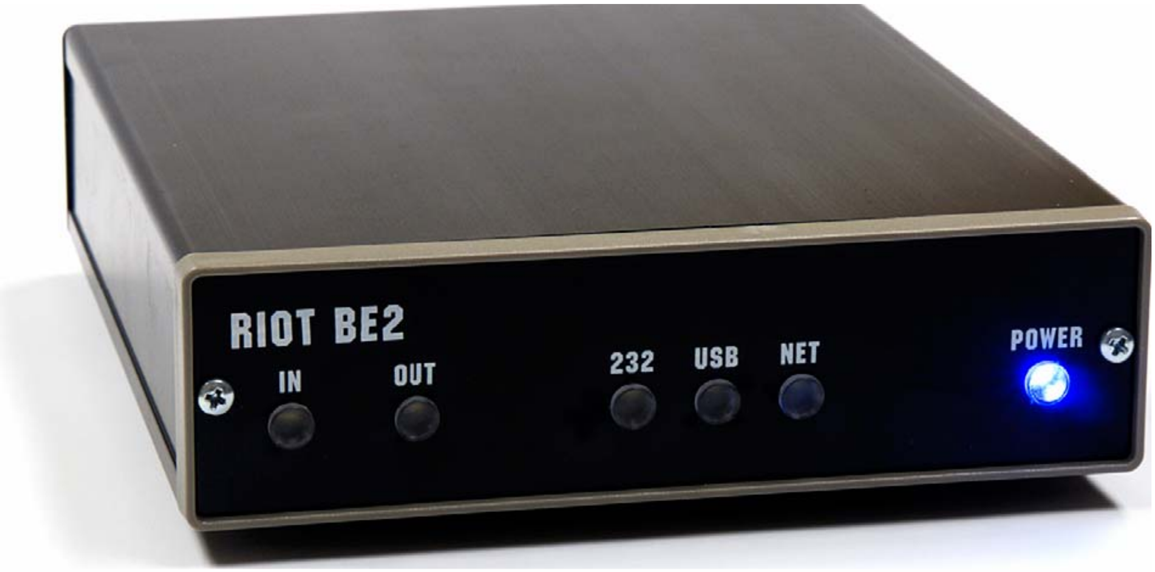
March 30th, 2007

GENERAL INFORMATION:

RIOT (Remote Input/Output Terminal) is an external remote control device, controllable via TCP/IP or USB. It runs an embedded Unix-like NetBSD based operating system.

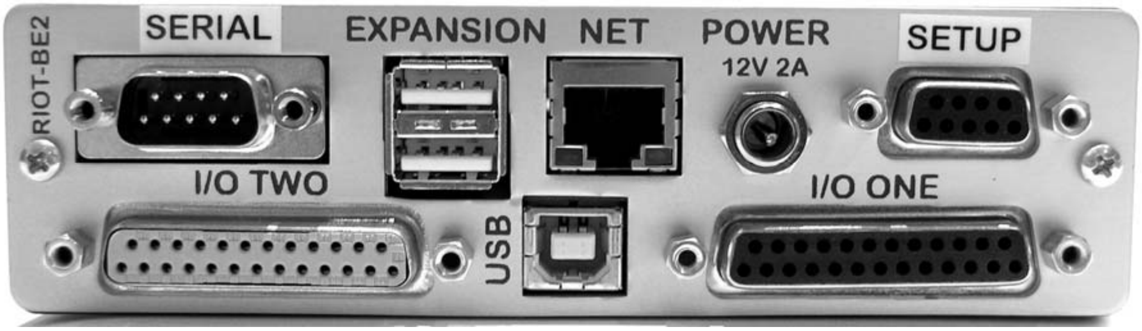
RIOT is the equivalent of two standard BEI cards, allowing 30 inputs and 16 outputs total. The pinouts on the device are the same as the standard BEI card. There are two parallel 25-pin ports on the back of the device for connecting to the device.

FRONT PANEL DISPLAY:



The front of each RIOT box has the following items:
- Blue Power Indicator
- Three green communications indicators, one each for Ethernet, USB and RS-232
- Two red I/O activity indicators, one each for input and output

BACK PANEL DISPLAY:



The back of each RIOT box has the following components:

- 10/100 Ethernet connector used for remote control of the RIOT box via TCP/IP commands in AudioVAULT applications.
- Two BEI Standard I/O 25-pin female connectors. Each has 15 inputs, 8 outputs. Each connector can supply up to 750mA of 5 or 12 volts (software selectable through web interface). Each output can sink up to 250mA at 24V. Input voltage ranges from 3 to 24 volts.
- Two RS-232 9 pin connectors, pinned out as "PC". One is intended to drive other devices such as Broadcast Tools switchers. The other is primarily intended for device configuration.
- A type B USB client connector - can be hooked to a host PC using this connector. The RIOT will look like a serial port to the host.
- Two type A USB host connections - These are for I/O expansion. These are not general purpose USB host ports.
- A locking power connector - This is a standard barrel power connector with a screw-locking ring

SETUP:

Initial setup of each RIOT box is done through a serial connection using HyperTerminal, or some other terminal application. A serial cable is built into the part number for RIOT and is supplied with each box. The serial cable (straight through 9-pin male to 9-pin female) should be connected to the port on the box marked "SETUP".

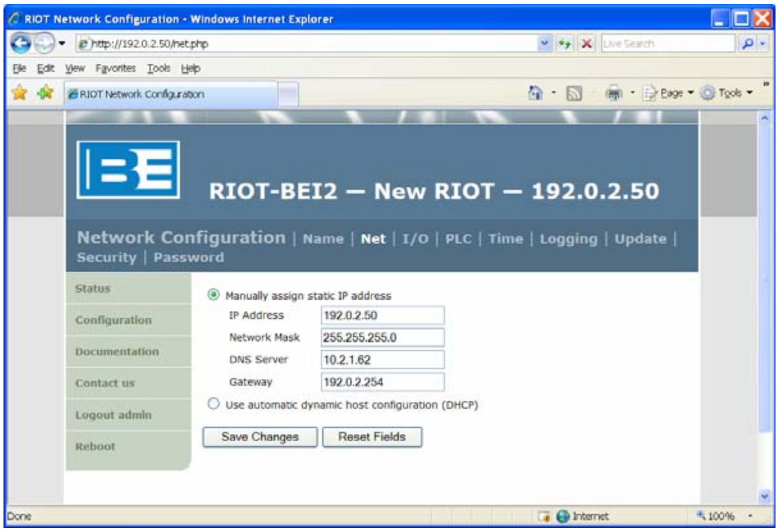
When connecting the serial cable, the connection should be set to
115200, 8, none, 1, none. If the connection is present when the RIOT box is powered on or rebooted, the user should see a status dump of information, and a prompt for logon when fully booted. If the connection is not present, the user can still login, but will not see the prompt or any status display until after the login is initiated.

RIOT needs to be rebooted in between configuration changes. For example, if the IP address is changed, RIOT needs to be restarted.

Standard login to any RIOT box is through user name "Admin", password "admin".

The command to set the IP address after logging in as Admin is "set ip <ipaddress>". After any configuration change, RIOT must be rebooted, so the command to type following the IP address change is "reboot". The user should then try to ping the RIOT over a typical network connection to see if the device is now available.

Following the setting of the IP address through the terminal program, the user should login to the RIOT via the web interface, and set the IP address, gateway, subnet mask and DNS server. To login to the web interface, from a browser, type http://<ipaddress>. On the home/status page, select the configuration option. This will prompt the user for a login. The default login is user "admin" with password of "admin". Once inside the configuration page, select the "Net" option. This will display a page for the user to type in the necessary network settings. After the changes are made, the user should click on the "Save Changes" button. Following this change, the RIOT box will once again need to be rebooted. This can be done by selecting the "Reboot" link on any of the RIOTs web pages.



RIOT Network Configuration - Windows Internet Explorer

http://192.0.2.50/net.php

RIOT-BEI2 — New RIOT — 192.0.2.50

Network Configuration | Name | Net | I/O | PLC | Time | Logging | Update | Security | Password

● Manually assign static IP address

IP Address 192.0.2.50

Network Mask 255.255.255.0

DNS Server 10.2.1.62

Gateway 192.0.2.254

○ Use automatic dynamic host configuration (DHCP)

CONTROLLING THE RIOT:

When using a RIOT box, AudioVAULT apps are typically configured to perform remote control via TCP. Getting data to and from the RIOT from apps like Audiovau.exe and AVAir.exe uses TCP Remote control and a series of strings. The RIOT box itself takes the parallel closures and converts them to strings, or takes the strings from the apps (over the TCP connection), and converts them to parallel closures. If the USB port on the RIOT is being used, then COMx remote control is used instead of TCP.

RIOT boxes can be controlled from multiple locations, since this is really just communication over IP. The RIOT box appears on the network, typically using a static IP address. Therefore, any machine running Vault software on the network could have access to the RIOT. A sample application where this would be used could be the case where several PCs/control rooms want to control the same switcher, but the switcher only has one serial port. In this case, we would hook the RIOT up to the switcher with a serial cable, and then let the control room PCs remote control the RIOT, basically sending pass-through commands to it.

Communication through RIOT can happen over up to 5 ports, or groups. Each group has settings for all ins and outs. Most often, these will not be changed and Group1 will be the only one used. Each group is accessed on a different port and can have different strings that it sends or receives based on the control. Standard use of the RIOT will have no need to reconfigure these settings.

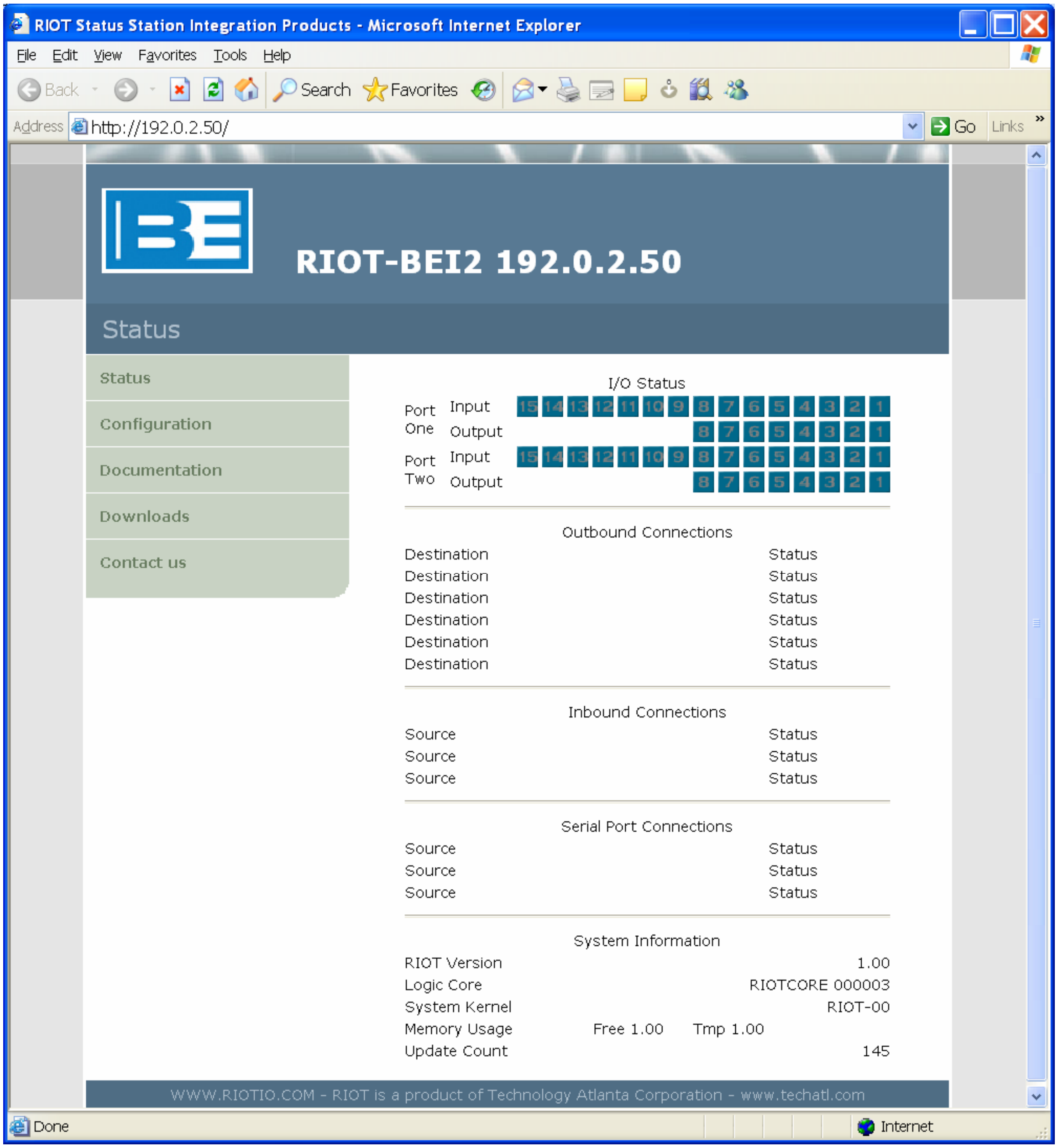
The groups are available on the following ports:

4100:	Group1
4101:	Group2
4102:	Group3
4103:	Group4
4104:	Group5

Sending data to port 4200 causes RIOT to pass those commands through to its serial port. This is useful when trying to control switchers where the only connection to the switcher is via a serial connection. The BCTools 16x16 is one of these for example.

RIOT can send its parallel closures in either a latched or pulsed logic, latched being the default.

A status page exists inside the embedded Web Server's interface. This shows a status 'LED' for each of the inputs and outputs. When data is sent to one of these 'pins', RIOT will show a 'lit' state on its web display. The user can access this interface through http://<ipaddress>.



RIOT-BEI2 192.0.2.50

Status

I/O Status

Port One	Input	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	Output								8	7	6	5	4	3	2	1
Port Two	Input	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
	Output								8	7	6	5	4	3	2	1

Outbound Connections

Destination	Status
Destination	Status
Destination	Status
Destination	Status
Destination	Status
Destination	Status

Inbound Connections

Source	Status
Source	Status
Source	Status

Serial Port Connections

Source	Status
Source	Status
Source	Status

System Information

RIOT Version		1.00
Logic Core		RIOTCORE 000003
System Kernel		RIOT-00
Memory Usage	Free 1.00	Tmp 1.00
Update Count		145

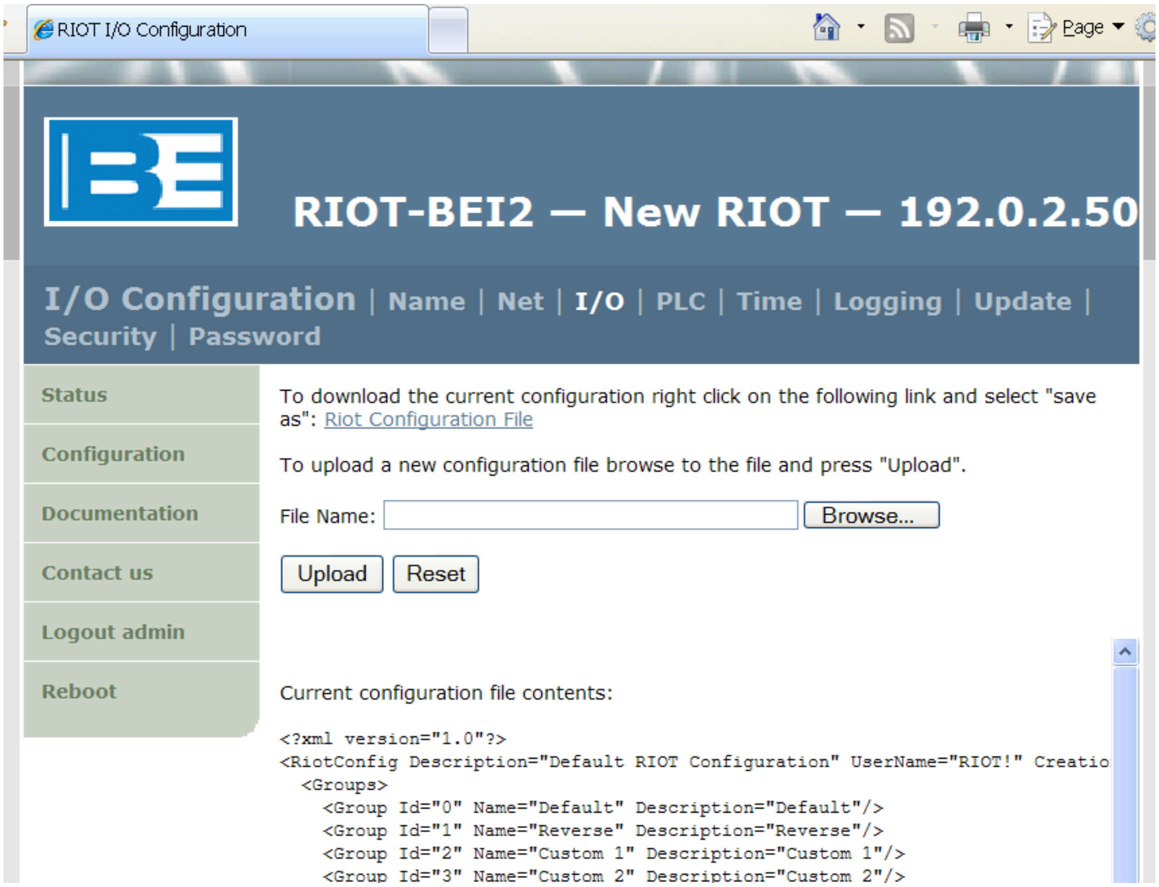
WWW.RIOTIO.COM - RIOT is a product of Technology Atlanta Corporation - www.techatl.com

PULSED REMOTE CONTROL:

RIOT has the ability to send its parallel closures in a pulsed mode. This means that RIOT will send the closure and depending on definition, turn that closure on or off based on configurable amounts of time. This feature is mostly used to interface with PR&E/Harris consoles.

To configure the RIOT box to send pulsed logic, the configuration XML file must be modified and uploaded to the RIOT. A default Pulsed Logic XML file can be requested from our Customer Service Team.

The current XML file in use can be viewed through the RIOT web interface. It can also be retrieved, modified, and then uploaded with changes. As with any change, the RIOT must be rebooted for the changes to take effect. The retrieval and uploading of the XML file happens through the Configuration | I/O page.


RIOT I/O Configuration
Page
RIOT-BEI2 — New RIOT — 192.0.2.50
I/O Configuration | Name | Net | I/O | PLC | Time | Logging | Update | Security | Password
Status
Configuration
Documentation
Contact us
Logout admin
Reboot
Browse...
Upload
Reset

To download the current configuration right click on the following link and select "save as": Riot Configuration File

To upload a new configuration file browse to the file and press "Upload".

File Name:

Current configuration file contents:

```
<?xml version="1.0"?>
<RiotConfig Description="Default RIOT Configuration" UserName="RIOT!" Creatio
  <Groups>
    <Group Id="0" Name="Default" Description="Default"/>
    <Group Id="1" Name="Reverse" Description="Reverse"/>
    <Group Id="2" Name="Custom 1" Description="Custom 1"/>
    <Group Id="3" Name="Custom 2" Description="Custom 2"/>
```

In the XML file, the user will see the following section:

```
  <PulseConfigurations>
          Some stuff
  </PulseConfigurations>
```

This section is where we define the pulse lengths for each of the output pins. We will use the following parameters on each pulse definition line: MaxOn, MaxOff, MinOn, MinOff. MaxOn defines the maximum amount of time to leave the closure on. MaxOff defines the maximum amount of time to leave a closure off. MinOn defines the minimum amount of time to leave the closure on. MinOff defines the minimum amount of time to leave the closure off.

A sample Pulse definition looks like this:

```
<Pulse Id="32" MaxOn="250" MinOn="250" MaxOff="-1" MinOff="250" Comment="PRnE Pulse"/>
<Pulse Id="33" MaxOn="250" MinOn="250" MaxOff="-1" MinOff="250" Comment="PRnE Pulse"/>
```

The first line defines the pulse configuration for Pin32, which happens to be Output 1.

```
<Pin Id="32" Name="OUT0.1" SendTrue="OUT0.1+" SendFalse="OUT0.1-" Inverted="0" Enabled="1"/>
```

Pin32 will be on for a maximum of 250 milliseconds. A value of -1 in the MaxOff parameter states to use a value of infinity, in essence allowing the pulse to be turned off and left off.

MinOff and MinOn are set to 250 in this example to specify the behavior if for some reason AudioVAULT sends a very short pulse – it guarantees the pulse is on long enough to be seen no matter what. The MinOff setting in the example makes sure the pulse actually goes away if AudioVAULT triggers it twice in short succession.

Normally two pulse configurations would be used for each channel on the PR&E consoles. A pulse on one pin will turn the console module on, and a pulse on a second pin will turn the console module off.

Sample INI file settings:

First of all, a section like the following will be required to establish a TCP socket to the RIOT:

```
[RemoteControl.RIOT1]
DeviceType=SOCKET
SocketType=Caller
ServiceName=4100
HostName=192.168.1.100
```

Samples for Latched Remote Control:
Some sample INI file settings for using the strings to make things happen.

The following line will cause a Start button to be 'pushed' when 'pin 1' is closed and will light a lamp when Deck 1 is running, and turn that lamp off when the deck is not running. It uses the first parallel port on the RIOT box:

```
IDD_Start1=RIOT1:+OUT0.1-OUT0.1+,RIOT1:IN0.1+;RIOT1:-OUT0.1-;
```

Note that in this line, we send the following string when the deck is started: OUT0.1-OUT0.1+. Coding an OFF-ON sequence like this guarantees a closure no matter what state the output was in previously.

To use the second parallel port on the RIOT box, but the same pins as in the above example:

```
IDD_Start1=RIOT1:+OUT1.1-OUT1.1+,RIOT1:IN1.1+;RIOT1:-OUT1.1-;
```

The following line uses both the down and the up click to advance the cursor in a list box on an AVRPS screen:

```
IDD_PreviousLine1=,RIOT1:IN0.2+,RIOT1:IN0.2-;
IDD_NextLine1=,RIOT1:IN0.3+,RIOT1:IN0.3-;
```

NOTEIn version 9.10, there is an issue where the remote control disconnects when the all remote controls are unloaded (like is commonly done when performing a LoadRemoteControl). In an effort to circumvent this, the following like may be used in the most common section (like AVRPS, AVNet, AVSat, etc.) to keep the connection active all the time IDD_ExpandTab=RIOT1:+Never

Samples for Pulsed Remote Control:

Some sample INI file settings for using the strings to make things happen.

The following line will cause a Start button to be 'pushed' when 'pin 1' is closed. When connected to a PR&E console, a pulse output on pin 1 can cause console module 1 to be turned on when the deck is running and will turn that module off with a pulse on pin 2 when the deck stops running. It uses the first parallel port on the RIOT box:

```
IDD_Start1=RIOT1:+OUT0.1-OUT0.1+,RIOT1:IN0.1+;RIOT1:-OUT0.2-OUT0.2+;
```

Note that in this line, we send the following string when the deck is started (leading edge): OUT0.1-OUT0.1+ and the following when the deck stops (trailing edge) OUT0.2-OUT0.2+. Coding an OFF-ON sequence like this guarantees a pulse no matter what state the output was in previously

Serial Pass Through Remote Control (BCTools Switcher Samples):

When talking to a BCTools switcher, the syntax for the stings to send to the switcher are executed in burst mode. This is specified by a "*" on the front of each string. The beginning of each string contains a "*" followed by the unit number (0-7), then the string. A detailed list of the BCTools modes and command set can be found in the BCTools manual.

The following are some sample INI file settings for controlling the BCTools 16x16 switcher:

First, a remote control section must be defined using port 4200 on the IPAddress at which the RIOT is located.

```
[RemoteControl.BCT]
DeviceType=SOCKET
SocketType=Caller
ServiceName=4200
HostName=192.168.1.100
```

Then the standard string to map an in to an out (perform a take) on the switcher, is

```
IDD_Indicator1=BCT:+*00101;
```

where BCT is the remote control device, and *00101 is the string for the RIOT to pass through to the switcher. In the string *00101, * specifies burst mode, 0 is the unit number for the switcher (not the RIOT), 01 is the input and 01 the output.

Another example, takes input 1 and routes it to output 8:

```
IDD_Indicator1=BCT:+*00108;
```

To take the same input (1) to 2 different outputs (15 and 16)

```
IDD_Indicator1=BCT:+*00115;BCT:+*00116;
```

To mute an output, use the following string:

```
IDD_Indicator1=BCT:+*0M01;
```

To mute all outputs at the same time, use this string:

```
IDD_Indicator1=BCT:+*0MA;
```

BCTools documentation lists the following as the command to set all outputs at the same time *uB,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a, where * specifies burst mode, u specifies the BCT unit number, and the first ‘a’ is output 1, the second ‘a’ is output 2, etc. Each ‘a’ should be the value hex 60 + the input number or F for off (can look this up in an ascii table). For example, to set output 1 to input 4, hex 60 + 4= 64 = ‘d’, in place of the first ‘a’ in the above string a ‘d’ would be used. Use the following letters to obtain the corresponding input number:

Input 1: a
Input 2: b
Input 3: c
Input 4: d
Input 5: e
Input 6: f
Input 7: g
Input 8: h
Input 9: i
Input 10: j
Input 11: k
Input 12: l
Input 13: m
Input 14: n
Input 15: o
Input 16: p

Note that in an IDD_ command a ‘,’ (comma) is considered a separator for the output, down, up click syntax, so in this case, our output string to send to the RIOT would have \x2C in place of each comma shown. The following example, maps input 1 to output 1, input 2 to output 2, input 3 to output 3, etc.

```
IDD_Indicator1=BCT:+*0B\x2Ca\x2Cb\x2Cc\x2Cd\x2Ce\x2Cf\x2Cg\x2Ch\x2Ci\x2Cj\x2Ck\x2Cl\x2Cm\x2Cn\x2Co\x2Cp;
```