

**Partner: BSS**  
**Model: BLU Series**  
**Device Type: Digital Signal Processor**



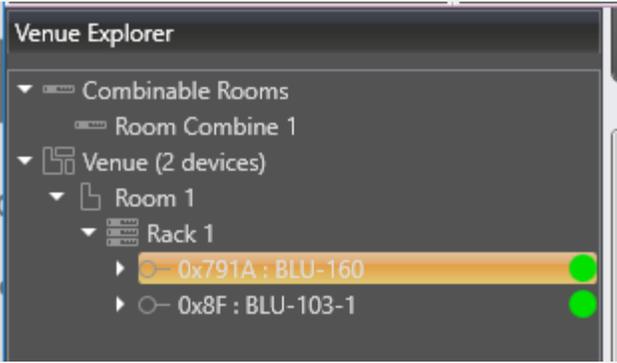
**GENERAL INFORMATION**

<b>SIMPLWINDOWS NAME:</b>	BSS BLU Command Processor IP v1.4
<b>CATEGORY:</b>	DSP
<b>VERSION:</b>	v1.4
<b>SUMMARY:</b>	This module controls all TCP/IP communication with the BSS BLU device.
<b>GENERAL NOTES:</b>	<p>This module is the core IP communication module for a suite of modules. The suite of modules utilizes the SIMPL# technology and will only work on the 3-Series and 4-Series Controllers.</p> <p>Up to 10 instances of this module can be used in a single program slot. The module has a parameter that allows you to choose one of the 10 instance IDs. <b>Each instance ID can only be used once.</b> Each command processor module will be dedicated to a single BSS Audio Node.</p> <p>The other modules in this suite are control modules. The control modules are responsible for providing the actual control interface in SIMPL. With the SIMPL# technology, the Control modules no longer need to be physically "connected" to the command processor. They register themselves automatically behind the scenes. Each of the control modules also have a command processor ID parameter that you assign to the instance of the command processor to which they report to. You can virtually have an unlimited number of control modules report to a single instance of a command processor.</p> <p>The command processor module has two feedback signals, "Is Communicating" and "Is Initialized" that <b>both must be high in order for the module to be used.</b> The command processor goes through an initialization processes where as it tests and subscribes to all the control points that have been defined by the included control modules. When this process is complete and successful, the module will assert the "Is Initialized" signal high. If there is a problem during the initialization process this signal will not be asserted, and the command processor will keep on attempting to get initialized. The main culprit preventing the command processor from getting initialized will be settings on the control modules. When the command processor has issue with communicating with a control point, it will log the failure in the controller's error log. The log entries should give you a hint at which control module(s) are giving you the issue. However if three sequential control points have settings issue, it will force the command processor to perform a strike out, and drop the connection to the unit, then re-establish. The strike out will also be in the error log if that occurs, along with information on the three failed attempts that caused the strike out. In very large systems, the best way to determine what control modules that are causing an issue, would be to perform divide and concur type of troubleshooting tactics. Commenting out all the control modules for a given command processor will allow the command processor to initialize. Adding and removing them back a group at a time and re-testing will allow you to find the problematic control modules.</p>
<b>CRESTRON HARDWARE REQUIRED:</b>	3-Series & 4-Series processors <b>only</b>
<b>SETUP OF CRESTRON HARDWARE:</b>	N/A
<b>VENDOR FIRMWARE:</b>	This module was tested using BSS BLU Firmware Version: 86.04.2

**Partner: BSS**  
**Model: BLU Series**  
**Device Type: Digital Signal Processor**



**PARAMETERS:**

<p><b>CommandProcessorID</b></p>	<p>Instance ID for the module.</p> <p>Up to 10 instances of this module can be used in a single program slot. The module has a parameter that allows you to choose one of the 10 instance IDs. <b>Each instance ID can only be used once.</b> Each command processor module will be dedicated to a single BSS Audio Node.</p>
<p><b>NodeAddress</b></p>	<p>This parameter is a hexadecimal value that needs to match the BSS Node to which this module is communicating with. You can find this ID by looking in the BSS Audio Architect software with the DSP program file open. In the venue explorer will be list of nodes.</p> <p><i>In the example below "791ah" or "8fh" would be valid node addresses.</i></p> 
<p><b>IPAddress</b></p>	<p>IP Address or valid DNS host name of the BSS DSP.</p>
<p><b>IPPort</b></p>	<p>IP Port used to communicate with the BSS DSP. 1023d is the default value.</p>

**Partner: BSS**  
**Model: BLU Series**  
**Device Type: Digital Signal Processor**

**CONTROL:**

<b>Connect</b>	D	Pulsing this signal will start the module communicating with the BSS BLU DSP. Alternatively you can assert this signal to high, and when the module is ready, will automatically connect.
<b>Disconnect</b>	D	Pulsing the signal will stop the module from communicating with the BSS BLU DSP.
<b>Initialize</b>	D	Pulsing the signal will start the initialization process once the module is communicating with the BSS. You can alternatively assert the signal to high prior to communicating, once communication has been successful, will automatically start the initialization process.
<b>Debug</b>	D	Asserting this signal high will allow debugging trace statements to be sent to debugger for troubleshooting help.

**Partner: BSS**  
**Model: BLU Series**  
**Device Type: Digital Signal Processor**

**FEEDBACK:**

<b>Is_Communicating</b>	D	This signal indicates if the successfully receiving properly formatted messages from the BSS.
<b>Is_Initialized</b>	D	This signal indicates that the control module is ready to be controlled, and that all the registered control modules have successfully received their state and their feedback is accurate.

**Partner: BSS**  
**Model: BLU Series**  
**Device Type: Digital Signal Processor**

**TESTING:**

<b>OPS USED FOR TESTING:</b>	CP3 v1.8001.5061.26823 CP4 v2.8000.00017.01
<b>SIMPL WINDOWS USED FOR TESTING:</b>	4.2000.00
<b>DEVICE DB USED FOR TESTING:</b>	200.240.001.00
<b>CRES DB USED FOR TESTING:</b>	216.00.001.00
<b>SYMBOL LIBRARY USED FOR TESTING:</b>	1179
<b>SAMPLE PROGRAM:</b>	BSS BLU v1.4 IP Demo.smw or BSS BLU v1.4 RS232 Demo.smw
<b>REVISION HISTORY:</b>	v1.0 – Initial Release v1.2 – Improved initial configuration between Simpl+ and Simpl# v1.3 – Updated to use latest CCI.Library / migrated to use PriorityCommUtil which has a passing time for thread sync v1.4 – Fix index issue with preset recall in library. – Updated level control demo to show use of SetValue.