

**Partner: BiAmp**  
**Model: AudiaFlex & Nexia**  
**Device Type: DSP**



## GENERAL INFORMATION

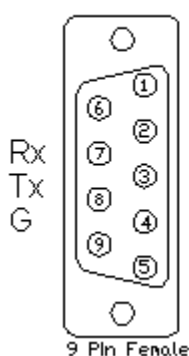
<b>SIMPLWINDOWS NAME:</b>	Biamp AudiaFlex + Nexia On-Off Control v7.4
<b>CATEGORY:</b>	Mixer
<b>VERSION:</b>	7.4
<b>SUMMARY:</b>	This module controls any mute point in the BiAmp AudiaFlex or Nexia.
<b>GENERAL NOTES:</b>	<p>This module will control any mute point in the Biamp AudiaFlex and Nexia.</p> <p>This module <b>MUST</b> be used in conjunction with the BiAmp AudiaFlex + Nexia Command Processor Serial v7.4 module or the BiAmp AudiaFlex + Nexia Command Processor IP v7.4. These modules process all transmitted and received serial strings and reformats device feedback so that this data can be sent to the proper module for final processing.</p> <p>When polling the BiAmp for current status, you should poll for only the information you really need at the time. The more data points you poll for at one time, the longer it will take to get an update for any one data point. It should not normally be necessary to poll for all data points all the time.</p> <p>This module has (5) five parameter fields, all of which must be set for proper module operation. All parameters are entered as ASCII characters. Mute_Device_Type is the control block type. This selected from a drop down list. Mute_Device_ID is the device's ID and is automatically assigned when the .dap or .nex file is compiled. Mute_Device_Instance is the "Logic Block's" ID that is automatically assigned when the .dap or .nex file is compiled. Mute_Index_1 is the first index number from the BiAmp software. This is typically the channel, input or output number to be controlled. Mute_Index_2 is the second index number from the BiAmp software. In a lot of cases this will be zero.</p> <p>This information is all contained in the Block properties field when developing the .dap file within the BiAmp AudiaFlex Windows software or the .nex file within the BiAmp Nexia software. A .dap file (Crestron Test w-VOIP.dap) was created for Crestron testing purposes and <b>MUST</b> be used for proper operation of the BiAmp AudiaFlex + Nexia Serial v7.4 Demo program. A .nex file (2 tc Room Combining.nex) was created for Crestron testing purposes and <b>MUST</b> be used for proper operation of the BiAmp AudiaFlex + Nexia IP v7.4 Demo program.</p> <p>When the Initialize input on the BiAmp AudiaFlex + Nexia Command Processor Serial v7.4 or on the BiAmp AudiaFlex + Nexia Command Processor IP v7.4 is pulsed, the BiAmp AudiaFlex + Nexia Command Processor Serial v7.4 or the BiAmp AudiaFlex + Nexia Command Processor IP v7.4 module will send out initialization strings to each of the To_Module_* outputs, asking for the connected module's command type, instance ID or Tag and indexes. The control module will transmit that information out its To_Processor output.</p> <p>NOTE: Digital inputs are buffered inside the module to prevent rapid button presses. Presses are only allowed once every 1.0 seconds.</p>
<b>CRESTRON HARDWARE REQUIRED:</b>	ST-COM, C2-COM, C2ENET-1/2
<b>SETUP OF CRESTRON HARDWARE:</b>	<p>RS232</p> <p>Baud: 38400</p> <p>Parity: N</p>

**Partner: BiAmp**  
**Model: AudiaFlex & Nexia**  
**Device Type: DSP**

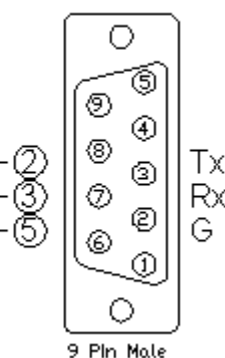


	Data Bits: 8 Stop Bits: 1 TCP/IP Port: 23 (Telnet)
<b>VENDOR FIRMWARE:</b>	4.560

Rear View of Connector



Rear View of Connector



## CONTROL:

<b>Volume_Mute_On/Off/Toggle</b>	D	Pulse to turn the mute on and off.
<b>Poll_Mute</b>	D	Pulse to poll for the current value.
<b>From_Processor</b>	S	Serial data signal to be routed from one of the To_Module_* outputs on the BiAmp AudiaFlex + Nexia Command Processor Serial v7.4 module or the BiAmp AudiaFlex + Nexia Command Processor IP v7.4 module.

## FEEDBACK:

<b>Volume_Mute_is_On/Off</b>	D	High to indicate the current mute state.
<b>To_Processor</b>	S	Serial data signal to be sent to the BiAmp AudiaFlex + Nexia Command Processor Serial v7.4 or the BiAmp AudiaFlex + Nexia Command Processor IP v7.4 module.

**Partner: BiAmp**  
**Model: AudiaFlex & Nexia**  
**Device Type: DSP**

**PARAMETERS:**

<b>Mute_Device_Type</b>	ASCII	Select the proper device type from the drop down list.
<b>Mute_Device_ID</b>	ASCII	Device address automatically assigned after the Biamp .dap file is compiled
<b>Mute_Device_Instance</b>	ASCII	Logic Block ID assigned after the Biamp .dap file is compiled
<b>Mute_Index_1</b>	ASCII	Mute index to be controlled. This is the input, channel or output number being controlled.
<b>Mute_Index_2</b>	ASCII	This used for cross point type devices. Typically, this would be zero. For cross points this is the output number for the cross point being controlled.

**Partner: BiAmp**  
**Model: AudiaFlex & Nexia**  
**Device Type: DSP**

**TESTING:**

<b>OPS USED FOR TESTING:</b>	PRO2: 4.003.0015 MC3: 1.002.0000
<b>SIMPL WINDOWS USED FOR TESTING:</b>	3.10.20
<b>DEVICE DB USED FOR TESTING:</b>	39.00.005.00
<b>CRESTRON DB USED FOR TESTING:</b>	29.01.004.00
<b>SAMPLE PROGRAM:</b>	BiAmp AudiaFlex + Nexia Serial v7.4 Demo

<b>REVISION HISTORY:</b>	<p>V3 – 2-Series Only, corrected dialer timing, text display, speed of dialing and over all operation (firmware)</p> <p>V4 – Changed timing of dialer strings sent when off hook</p> <p>V5 – Made changes for the new responses from the BiAmp. These new responses have the command details and status in them. This eliminates the need to poll for status when making changes. Added new commands. Added buffering for the responses to improve system response.</p> <p>V5.1-Changed the Command Processor module to handle the response for presets. Also eliminated the Command Processor sending any response if the unit ID is determined to be 0. Changed all of the modules to allow instance IDs up to 65534d. Changed all modules to look for the proper channel ID. Added MBMUTE command to the On-Off module.</p> <p>V7.0 – Changed all modules to allow the use on Instance ID Tags. Changed the volume control module to allow for the selection of the size of the volume change step. Changed the command processor module to handle all filtering of the feedback. Eliminated the unit buffer module. Also eliminated the need for using serial buffers.</p> <p>V7.1 – Fixed an issue in the Processor module that allowed feedback from the BiAmp to be sent to the wrong module. Also fixed an issue in the level control module with controlling the AEC Inputs.</p> <p>V7.2 – Created separate processor module for IP and Serial control. The IP processor module sets the telnet echo to off. Fixed an issue in the BiAmp AudiaFlex + Nexia Dialer module where a wait statement in Simpl+ was not programmed correctly. Fixed an issue with the processor module that allowed the processor to "lockup" if the queue pointers in the wrapped around. Changed the method that the processor module uses to collect the InstanceID information. The new method should be less confusing to program. Added a Simpl Windows gather to the processor modules to reduce the number of entries into the Simpl+. Added code to the control modules to prevent buttons presses more frequently than every 1 second.</p> <p>V7.4 – Adjusted a timing issue with the way that the modules initialize. Changed the Initialize input to allow a larger time window for pulsing the Initialize input after the 60 second lock out. Fixed an issue with the VoIP module when sending the DTMF tones. Tested the modules with the 3-Series processor. Added new compiler directives #IF-SERIES2 and #IF_SERIES3 in Simpl+ to clean up warnings when compiling Simpl+.</p>
--------------------------	--