

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



GENERAL INFORMATION

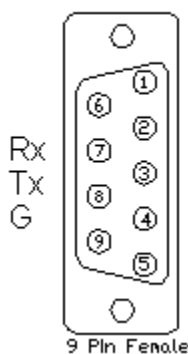
SIMPLWINDOWS NAME:	Biamp AudiaFlex + Nexia On-Off Control v7.0
CATEGORY:	Mixer
VERSION:	7.0
SUMMARY:	This module controls any mute point in the BiAmp AudiaFlex or Nexia.
GENERAL NOTES:	<p>This module will control any mute point in the Biamp AudiaFlex and Nexia.</p> <p>This module MUST be used in conjunction with the Biamp AudiaFlex + Nexia Command Processor v7.0.umc module. This module processes 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>The To_Processor output of this module must be connected to the From_Modules input on the BiAmp AudiaFlex + Nexia Command Processor v7.0 module. The Instance_ID_to_Processor must be connected to one of the Module_*_Instance_ID inputs on the BiAmp AudiaFlex + Nexia Command Processor v7.0 module. The From_Processor input on this module must be connected from the corresponding To_Module_* output on the BiAmp AudiaFlex + Nexia Command Processor v7.0 module.</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 file is compiled. Mute_Device_Instance is the "Logic Block's" ID that is automatically assigned when the .dap 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. A .dap file (Crestron Test v7.0.dap) was created by Crestron for testing purposes and MUST be used for proper operation of the v7.0 Demo Pro2 program.</p> <p>NOTE: THIS MODULE WAS DEVELOPED AND TESTED WITH THE BIAMP AUDIAFLEX. THE INCLUDED .DAP FILE WAS PROVIDED BY BIAMP, AND IS FOR THE AUDIAFLEX ONLY. ACCORDING TO BIAMP, THESE MODULES WILL WORK FOR THE NEXIA. A CONFIGURATION FILE WILL NEED TO BE CREATED FOR THE NEXIA (NOT PROVIDED), AND WILL BE REQUIRED FOR OPERATION OF THE UNIT. FOR MORE INFORMATION ABOUT CONFIGURATION FILES AND HOW TO CREATE THEM PLEASE CONTACT BIAMP.</p> <p>When the Initialize input on the BiAmp AudiaFlex + Nexia Command Processor v7.0 is pulsed, the BiAmp AudiaFlex + Nexia Command Processor v7.0 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 Instance_ID_to_Processor output. The Instance_ID_to_Processor output of a control module must be connected to one of the Module_*_Instance_ID inputs. The corresponding To_Module_* output must be</p>

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

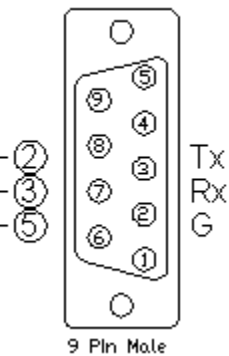


	connected to the From_Processor input of the same control module.
CRESTRON HARDWARE REQUIRED:	ST-COM, C2-COM
SETUP OF CRESTRON HARDWARE:	RS232 Baud: 38400 Parity: N Data Bits: 8 Stop Bits: 1
VENDOR FIRMWARE:	4.560

Rear View of Connector



Rear View of Connector



CONTROL:

Volume_Mute_On/Off/Toggle	D	Pulse to turn the mute on and off.
Poll_Mute	D	Pulse to poll for the current value.
From_Processor	S	Serial data signal to be routed from one of the To_Module_* outputs on the BiAmp AudiaFlex + Nexia Command Processor v7.0 module.

FEEDBACK:

Volume_Mute_is_On/Off	D	High to indicate the current mute state.
To_Processor	S	Serial data signal to be sent to the BiAmp AudiaFlex + Nexia Command Processor v7.0.
Instance_ID_to_Processor	S	Serial signal to be routed to one of the Module_*_Instance_ID inputs on the BiAmp AudiaFlex + Nexia Command Processor v7.0 module.

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

**PARAMETERS:**

Mute_Device_Type	ASCII	Select the proper device type from the drop down list.
Mute_Device_ID	ASCII	Device address automatically assigned after the Biamp .dap file is compiled
Mute_Device_Instance	ASCII	Logic Block ID assigned after the Biamp .dap file is compiled
Mute_Index_1	ASCII	Mute index to be controlled. This is the input, channel or output number being controlled.
Mute_Index_2	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.

TESTING:

OPS USED FOR TESTING:	3.155.1240
SIMPL WINDOWS USED FOR TESTING:	2.10.32
DEVICE DB USED FOR TESTING:	20.02.009.00
CRESTRON DB USED FOR TESTING:	20.00.05.00
SAMPLE PROGRAM:	BiAmp AudiaFlex + Nexia v7.0 Demo Pro2

REVISION HISTORY:

V3 – 2-Series Only, corrected dialer timing, text display, speed of dialing and over all operation (firmware)

V4 – Changed timing of dialer strings sent when off hook

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.

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.

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.