



Duet Module Help File

Bose Videobar VB-S

TABLE OF CONTENTS

Introduction3

Overview4

Implementation5

Port Mapping.....6

Command Events10

Command Feedback.....12

Important Notes.....14

Programming Notes15

Revision History

Date	Initials	Comments
12-21-2022	AH	v1.0.0 Initial Release

Introduction

This is a reference manual to describe the interface provided between an AMX NetLinx system and a Bose VB-S Videobar conferencing system. The AMX module connects with the device over Ethernet only and utilizes protocol provided through the web interface of the device.

This module was written with Eclipse Mars.2 version 4.8.5, and NetLinx Studio version 4 build 4.4.1626. The panel program was built using TPDesign5 version 1.5.0, build 111.

This module is compatible with NX series controllers only, minimum firmware version is 1.6.175.

The module must be configured with the IP address and port of the device it is to connect to, and the page size of the event log (if used) on the touchpanel. This must be done manually via a telnet session or can be automated from the UI module. These properties must be set using the 'PROPERTY-' command. The 'REINIT' command is used to notify the module that new properties are now available and to start using them. When these commands are used in the appropriate order, the module will attempt to connect to the specified IP address periodically until a connection is made or a new IP address is submitted using the 'PROPERTY-' and 'REINIT' commands. Please see the [Programming Notes](#) section for additional information.

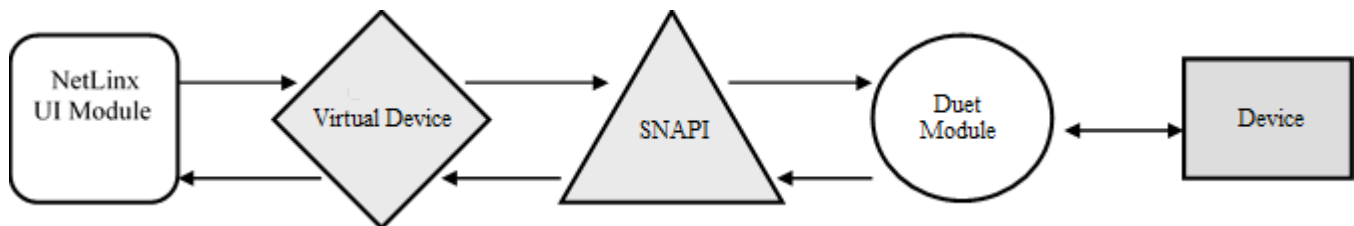
The module communicates over HTTPS with default port 443 and requires password authentication for some administrator-level commands. The default password is "Bose123!". Ensure that the password specified as module parameters match the administrator login for the camera.

Overview

The module translates between the standard interface described below and the device protocol. It parses the buffer for responses from the device, sends strings to control the device, and receives commands from the UI module or telnet sessions.

A User Interface (UI) module is also provided. This module uses the standard interface described below and parses the command responses for feedback.

The following diagram gives a graphical view of the interface between the interface code and the Duet module.



A sample UI module and a touch panel file are provided in the module package. These are not intended to cover every possible application scenario but can be expanded as needed by a dealer to meet the requirements of a particular installation.

Implementation

To interface to the device module, the programmer must perform the following steps:

1. Define the device ID for the physical device that will be controlled.
2. Define the main virtual device ID that the device COMM module will use to communicate with the main program and User Interface. Duet virtual devices use device numbers 41000 - 42000.
3. If a touch panel interface is desired, a touch panel file (Bose VB-S Demo v1_0.TP5) and module (Bose VB-S Demo v1_0 - UI.asx) have been created for testing.
4. The Duet BoseVbs_dr1_0_0.jar module must be included in the program with a DEFINE_MODULE command. This command starts execution of the module and passes in the following key information: the device ID of the device to be controlled, and the virtual device ID for communicating to the main program.

An example of how to do this is shown below.

```

DEFINE_DEVICE

// ----- PHYSICAL DEVICES

dvDevice = 0:3:0;
dvTp      = 10001:1:0;

// ----- VIRTUAL DEVICES

vdvDevice = 41001:1:0;

(*****
(*          CONSTANT DEFINITIONS GO BELOW          *)
*****)
DEFINE_CONSTANT

LOG_PAGE_SIZE = 10;

(*****
(*          STARTUP CODE GOES BELOW                  *)
*****)
DEFINE_START

// ---- COMM MODULE

DEFINE_MODULE 'BoseVbs_dr1_0_0' modBoseComm(vdvDevice, dvDevice);

// ---- UI MODULE

DEFINE_MODULE 'Bose VB-S Demo v1_0 - UI' modBoseUI(dvTp, vdvDevice);

(*****
(*          THE EVENTS GO BELOW                      *)
*****)
DEFINE_EVENT

DATA_EVENT[vdvDevice]
{
  ONLINE:
  {
    wait 50
    {
      SEND_COMMAND vdvDevice,'DEBUG-4';
      SEND_COMMAND vdvDevice,"PROPERTY-IP_Address,192.168.1.1";
      SEND_COMMAND vdvDevice,"PROPERTY-Port,443";
    }
  }
}

```

```

SEND_COMMAND vdvDevice,"PROPERTY-Password,Bose123!";
SEND_COMMAND vdvDevice,"PROPERTY-LOG_PAGE_SIZE,", ITOA(LOG_PAGE_SIZE);
SEND_COMMAND vdvDevice,'REINIT';
    }
}
}

```

Port Mapping

The module requires a single port for SNAPI channel, level, and command mapping. The module uses port 1. The port is defined as part of the device definition:

```
// ----- VIRTUAL DEVICES
```

```
vdvDevice      = 41001:1:0;
```

The virtual device identifier is 41001, the port is 1, and the system is 0 – indicating the module will use the system identifier defined within the processor configuration.

Channel Events

The UI module controls the device via channel events for features indicated by on/off or enabled/disabled state. The channels supported by the module are listed below. These channels are associated with the virtual device(s) and are independent of the channels associated with the touch panel device. Not all channels will be available on all virtual ports.

Note: An ‘*’ indicates an extension to the standard SNAPI API

Channel	Description
24	RAMP: Loudspeaker volume increased while channel is active.
25	RAMP: Loudspeaker volume decreased while channel is active.
26	PULSE: Toggle the loudspeaker volume mute state.
132	RAMP: Camera tilts up while channel is active.
133	RAMP: Camera tilts down while channel is active.
134	RAMP: Camera pans left while channel is active.
135	RAMP: Camera pans right while channel is active.
158	RAMP: Camera zooms out while channel is active.
159	RAMP: Camera zooms in while channel is active.
251	ON: Indicates the module is communicating with the device. OFF: Indicates the module is not communicating with the device.
252	ON: Indicates the module is initialized with all device state information. OFF: Indicates the module is not initialized.
*300	ON: Indicates the device ready state is on. OFF: Indicates the device ready state is off.
*310	PULSE: Toggle the system Bluetooth enable state.
*311	PULSE: Enable the system Bluetooth.
*312	PULSE: Disable the system Bluetooth.
*313	ON: Indicates the system Bluetooth is enabled. OFF: Indicates the system Bluetooth is disabled.
*315	PULSE: Toggle the Bluetooth Button enable state.
*316	PULSE: Enable the Bluetooth Button state.

*317	PULSE: Disable the Bluetooth Button state.
*318	ON: Indicates the Bluetooth Button is enabled. OFF: Indicates the Bluetooth Button is disabled.
*325	ON: Indicates the system WIFI is enabled. OFF: Indicates the system WIFI is disabled.
*326	ON: Indicates the system WIFI DHCP is enabled. OFF: Indicates the system WIFI DHCP is disabled.
*330	ON: Indicates the USB connection status is enabled. OFF: Indicates the USB connection status is disabled.
*331	ON: Indicates the USB call status is enabled. OFF: Indicates the USB call status is disabled.
*400	PULSE: Toggle the Microphone Mute enable state.
*401	PULSE: Enable the Microphone Mute state.
*402	PULSE: Disable the Microphone Mute state.
*403	ON: Indicates the Microphone Mute state is enabled. OFF: Indicates the Microphone Mute state is disabled.
*500	PULSE: Toggle the Auto Framing enable state.
*501	PULSE: Enable the Auto Framing state.
*502	PULSE: Disable the Auto Framing state.
*503	ON: Indicates the Auto Framing state is enabled. OFF: Indicates the Auto Framing state is disabled.
*600	ON: Indicates the Bluetooth and BLE state is enabled. OFF: Indicates the Bluetooth and BLE state is disabled.
*605	PULSE: Toggle the Bluetooth Pairing state.
*606	PULSE: Enable the Bluetooth Pairing state.
*607	PULSE: Disable the Bluetooth Pairing state.
*608	ON: Indicates the Bluetooth Pairing state is enabled. OFF: Indicates the Bluetooth Pairing state is disabled.

Table 1 - Virtual Device Channel Events

Level Events

The UI module controls the device via level events sent to the module for features indicated by a range value. The commands supported by the module are listed below.

Note: An ‘*’ indicates an extension to the standard API.

Level	Description
1	Level indicates current loudspeaker volume level as a scaled percentage range from 0 to 255.
15	Level indicates current zoom position as a scaled percentage range from 0 to 255.
27	Level indicates current pan position as a scaled percentage range from 0 to 255.
28	Level indicates current tilt position as a scaled percentage range from 0 to 255.

Table 2 – Level Events

Command Events

The UI module controls the device via command events sent to the module for features other than on/off or range elements. The commands supported by the module are listed below.

Note: An ‘*’ indicates an extension to the standard API.

Command	Description
DEBUG-<value>	<p>Set the state of debugging messages in the UI module and the Module. Use the first virtual port when sending this command.</p> <p>Note: See Programming Notes section.</p> <p><value> : 1 = set only error messages on 2 = set error and warning messages on 3 = set error, warning & debug messages on 4 = set all messages on</p> <p>Example: DEBUG-1</p>
?DEBUG	Returns the current debug level, format is DEBUG-<level>.
?FWVERSION	Returns the firmware version reported by the device, format is VERSION-<version>.
?VERSION	Returns the version of the module, format is VERSION-<version>.
DISCONNECT	Terminates the active connection to the device, closes the network port, and clears module state.
PASSTHRU	Sends a message directly to the device, format is PASSTHRU-<message>.
?CAMERAPRESET	Returns the last recalled preset.
?CAMERAPRESETCOUNT	Returns the number of available presets. Maximum number of presets is 3.
?CAMERAPRESETPROPERTIES	Returns list of available presets, format is CAMERAPRESETPROPERTIES-<preset number>,<preset name>.
?CAMERAPRESETPROPERTY	Returns the preset property of the requested preset, format is CAMERAPRESETPROPERTY-<preset number>,<preset name>.
CAMERAPRESETSAVE	Saves the camera pan, tilt, and zoom position to the specified preset number, format is CAMERAPRESETSAVE-<preset>.

CAMERAPRESET	Recalls the specified preset, format is CAMERAPRESET-<preset>.
*?BLUETOOTHPAIED	Returns the name of the Bluetooth paired device.
*?BUILDING	Returns the device building name.
*?CAMERASTATE	Returns the state of the camera. Possible values include: "active", "inactive", "upgrading".
*?FLOOR	Returns the device floor name.
*?MODEL	Returns the device model.
*?NAME	Returns the device name.
*REBOOT	Reboots the device, the module will automatically reconnect when the device returns online. Format is REBOOT.
*?ROOM	Returns the device room name.
*?SERIALNUMBER	Returns the serial number of the device.
*?WIFISTATE	Returns the state of the device WIFI interface. Possible values include: "idle", "failure", "association", "configuration", "ready", "disconnect", "online".
*?WIFIADDRESS	Returns the static IP address of the WIFI interface used when DHCP is disabled.
*?WIFIMAC	Returns the MAC Address of the WIFI interface.

Table 3 – Send Command Definitions

Command Feedback

The COMM module provides feedback to the User Interface module for all command events.

Command	Description
?BLUETOOTHPAIREDD	Response indicates the name of the Bluetooth paired hardware. Example: BLUETOOTHPAIREDD-Headset
?CAMERASTATE	Response indicates the current state of the camera. Example: CAMERASTATE-inactive
?CAMERAPRESET	Request returns the current camera preset. Example: CAMERAPRESET-1
?CAMERAPRESETCOUNT	Response indicates the number of saved presets. Example: CAMERAPRESETCOUNT-3
?CAMERAPRESETPROPERTIES	Response indicates the full list of saved presets. Example: CAMERAPRESETPROPERTY-"1,Home Preset" CAMERAPRESETPROPERTY-"2,First Preset" CAMERAPRESETPROPERTY-"3,Second Preset"
?CAMERAPRESETPROPERTY	Response indicates the preset properties for the specified preset. Example: CAMERAPRESETPROPERTY-1,Home Preset
?DEBUG	Request returns the state of debugging messages in the UI module and the Module. This is reported on the first virtual port. <value> : 1 = set only error messages on 2 = set error and warning messages on 3 = set error, warning and debug messages on 4 = set all messages on DEBUG-1

?FLOOR	<p>Response indicates the name of the floor reported by the device.</p> <p>Example: BUILDING-2nd Floor</p>
?MODEL	<p>Response indicates the model name of the device</p> <p>Example: MODEL-VB-S</p>
?NAME	<p>Response indicates the name of the device.</p> <p>Example: NAME-Videobar1-0522</p>
?SERIALNUMBER	<p>Response indicates the serial number reported by the device.</p> <p>Example: SERIALNUMBER-081413W10920522AE</p>
?WIFIADDRESS	<p>Response indicates the static IP Address of the WIFI interface.</p> <p>Example: WIFIADDRESS-0.0.0.0</p>
?WIFIMAC	<p>Response indicates the MAC Address of the WIFI interface.</p> <p>Example: WIFIMAC-4C:87:5D:9C:EA:B6</p>
?WIFISTATE	<p>Response indicates the network state of the WIFI interface.</p> <p>Example: WIFISTATE-ready</p>

Table 2 - Command Feedback Definitions

Important Notes

- Please contact manufacturer's customer service regarding support for the Bose VB-S.
- The hardware used to test this module include an NX-3200 series controller with firmware version 1.6.175, and an MXT-700i touchpanel with firmware version 2.104.28.
- Properties can be modified by calling the SEND_COMMAND with the property name and new value. See programming notes below.

Property Name	Default Value
IP Address	192.168.1.100
Port	443
Password	Bose123!
LOG_PAGE_SIZE	10

- The module will automatically initialize when the program starts up (or manually when the REINIT command is sent). Initialization consists of querying the device for all state information required for proper module operation.
- Response times can vary based on network connection and activity within the device. Some commands take longer to process than others. Note that the preset commands take up to 5 seconds to process and return a response.

Programming Notes

To establish a communication connection between the module and the device, the module must be informed of the IP address of the device it is to connect to, and the page size of the event log (if used) on the touchpanel. Use the PROPERTY- command to set this information and then use the REINIT command to force it to take effect. Here is an example of how these commands are used. Note that your virtual device (41001:1:0) and the IP address, credentials and log size may differ from this example. Substitute the appropriate values where necessary.

```
SEND_COMMAND vdvDevice,'DEBUG-4';
SEND_COMMAND vdvDevice,"PROPERTY-IP_Address,',          IP_ADDRESS";
SEND_COMMAND vdvDevice,"PROPERTY-Port,',          PORT";
SEND_COMMAND vdvDevice,"PROPERTY-LOG_PAGE_SIZE,',  ITOA(LOG_PAGE_SIZE)";
SEND_COMMAND vdvDevice,'REINIT';
```

These values have been defined as constant variables in the main module and should be changed to match your environment. The program takes care of passing the variables to the module.

```
(*****
(*      CONSTANT DEFINITIONS GO BELOW      *)
(*****)
DEFINE_CONSTANT

////////////////////////////////////
// DEVICE CONNECTION PROPERTIES
////////////////////////////////////
CHAR IP_ADDRESS[]  = '192.168.1.100';
CHAR PORT[]       = '443';

LOG_PAGE_SIZE     = 10;
```