Muxlab's AV over IP Driver for the AMX NetLinx Integrated Controller

Release Date	Initials	Comments
2017/02/01	FO	

Introduction

This document provides guidelines on how to control Muxlab's AV over IP products with an AMX NetLinx Integrated Controller over an Ethernet (TCP-IP) network. The driver enabling this control was written using NetlLinx Studio version 4.4.1626. (http://www.amx.com/products/NetLinxStudio.asp).

Figure 1 below illustrates a basic system setup incorporating the basic building blocks/components necessary to run the entire system. Note that the text in parenthesis in the below list identifies example model numbers of some of the required system components, but other models (or other brands in the case of the Ethernet Switch, such as Luxul, NetGear, Niveo, etc.) may be utilized.

- A Tablet (Apple or Android) to run the user application interface.
- An Ethernet Network Switch (for example: Cisco, SG300 28-Port Gigabit PoE Managed Switch).
- An AMX NetLinx Integrated Controller (for example: AMX NI-900).
- A MuxLab AV over IP Transmitter and Receiver (for example: MuxLab 500753-TX and 500754-RX). Note that each source in the system will need to be connected to a Transmitter and each destination display will need to be connected to a Receiver.
- A MuxLab ProDigital Network Controller (MNC or MNC811) to manage the AV over IP devices and provide an Application Program Interface (API) to the AMX Driver (for example: MuxLab Network Controller 500811).

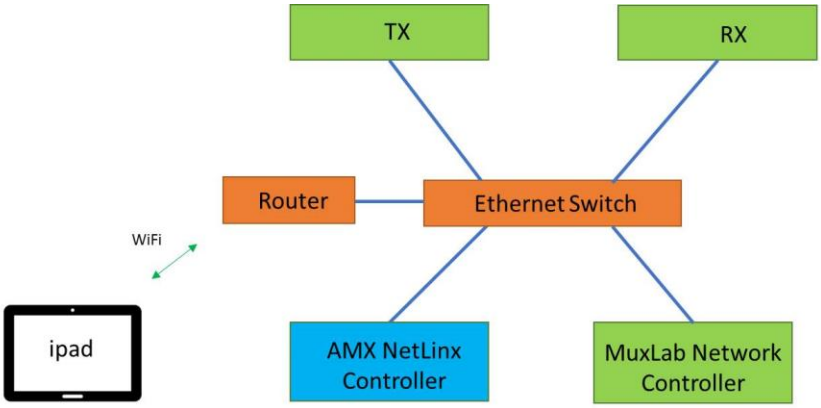

Figure 1: Basic System Setup

Implementation

To ensure communication between the AMX NetLinx Integrated Controller and the MuxLab Network Controller (MNC) , the programmer needs to define the following:

1. The devices involved in the system (Device: Port: System)
2. The ID of the interface (in our case an IPad)
3. The IP of the MNC (500811) device, that the program is communicating with.

An example follows

```
DEFINE_DEVICE

dvMuxLab = 0:3:0 // IP Port to connect to MuxLab controller
dvIPAD = 10001:1:0 // iPad Interface ; the ID is 10001

(***********************************************************)
(*    CONSTANT DEFINITIONS SHOWN BELOW    *)
(***********************************************************)
DEFINE_CONSTANT
long lMuxLabPort = 80 //MuxLab TCP Port number
char chMuxLabIP[] = '192.168.168.53' //example of IP Address of MuxLab controller
...
```

Notes for iPad and Android Tablet:

- The interface was designed with the TPDesign4 PC software, which can be found at: (http://www.amx.com/products/TPDesign.asp). This software running on a PC will be used to help create the user interface application .TP4 file for the Apple iPad, Android tablet or other device operating systems (OS).
- The.TP4 file created above must be transferred to the tablet (to be used to run the user interface application) using the TPTransfer windows software found at: (http://touchpanelcontrol.com/catalog/product/download). This software will transfer the TP4 file to an Apple iPad, Android Tablet, etc. running the TPControl application.
- For the Apple iPad, download the Apple TPControl application at: https://itunes.apple.com/ca/app/tpcontrol/id348715945?mt=8
- For the android tablet, download the Android TPControl application at : https://play.google.com/store/apps/details?id=com.touchpanelcontrol.tpcontrol&hl=en).
- Carefully follow the installation guidelines provided in the documentation within the TPTransfer software.

Note: for an AMX Touch Panel Device, please refer to the documentation for the AMX device, rather than the instructions above.

Figure 2 below shows an example (Example #1) user application interfaces running on a tablet, supporting the AMX Driver and MuxLab AV over IP Devices. This simple example can be used to manage virtual matrix switch configurations. Example code for this Example #1 is included with the driver files under the following file names:

Example #1 Code:

Muxlab_AMX_Driver_Ex1.axs
Muxlab_AMX_Driver_Ex1.src
Muxlab_AMX_Driver_Ex1.tkn
Muxlab_AMX_Driver_Ex1.tko

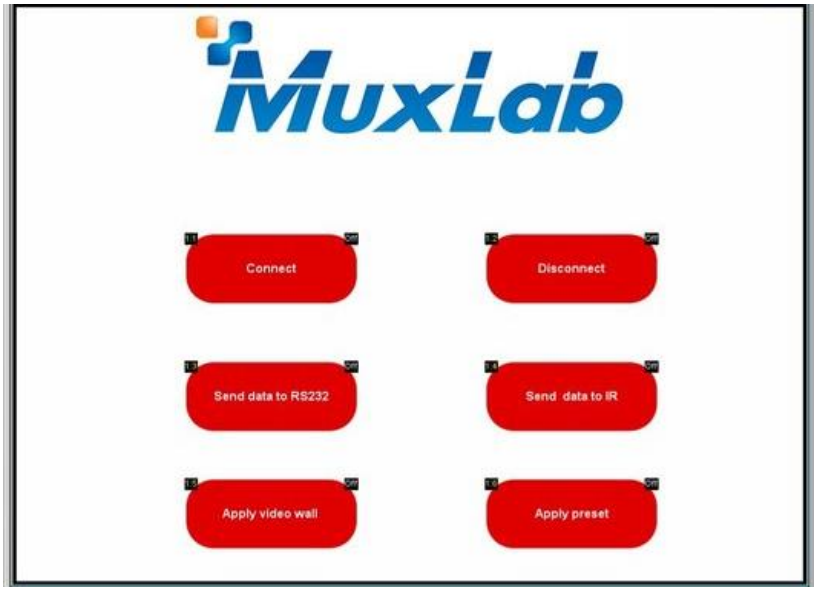

Figure 2: Tablet User Application Interface Example #1

Figure 3 below shows another example (Example #2) user application interfaces running on a tablet, supporting the AMX Driver and MuxLab AV over IP Devices. In this case the example can be used to manage a simple video wall arrangement. Example code for this Example #2 is included with the driver files under the following file names:

Example #2 Code:

Muxlab_AMX_Driver_Ex2.axs
Muxlab_AMX_Driver_Ex2.src
Muxlab_AMX_Driver_Ex2.tkn
Muxlab_AMX_Driver_Ex2.tko

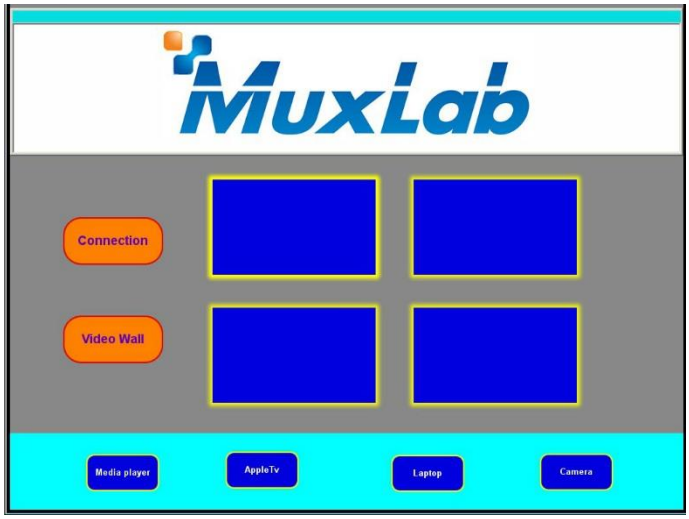

Figure 3: Tablet User Application Interface Example #2

In this example, by pressing the “Connection” button, the button will momentarily turn green to indicate that the interface is in “connection mode”, where the user can connect a given screen with a source. In this example, you have four sample sources (media player, Apple TV, Laptop and camera). Once in “connection mode”, pressing on a screen will momentarily turn the screen green to indicate that it is selected, see Figure 4. To complete the operation, you need to select a source, therefore pressing on one of the sources will complete the command and establish a connection between the selected display screen and source.

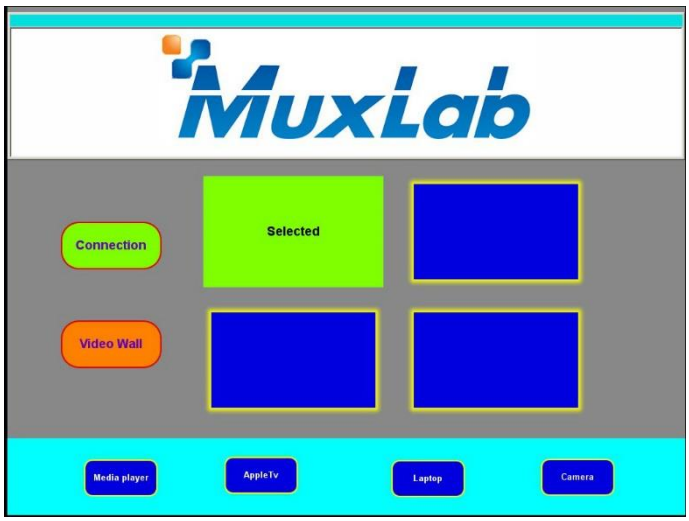

Figure 4: “Connection Mode” with the screen selected

When you want to apply a pre-existing video wall configuration, press on the “Video Wall” button (which will momentarily turn the button green) and the four screens in this example will also be green, indicating that they are all selected, as shown in Figure 5.

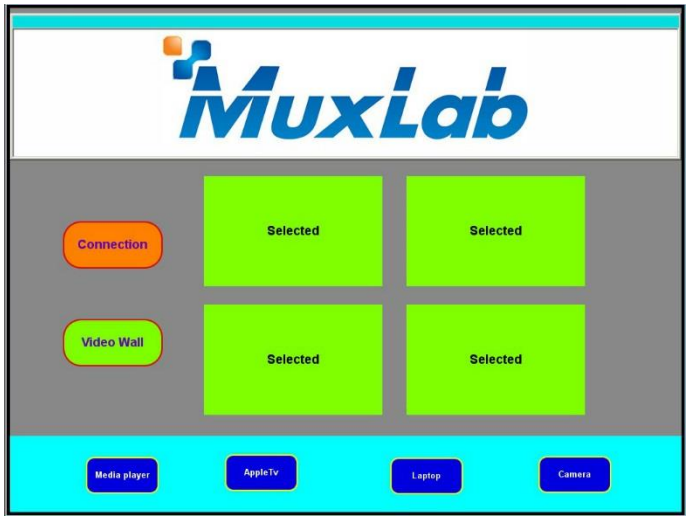

Figure 5: "Video Wall Mode" selected

Simply pressing on a given source, will connect that source to the Video Wall display screens.

Both the above examples (#1 & #2) are simple implementations of what can be achieved. More complex and flexible implementations can now be created by the developer making use of this driver, using the basic principles found within these examples. The commands supported by this Driver follow.

Commands to Send to the MNC to Control MuxLab AV over IP TX & RX Devices

The user controls the MNC via IP commands (JSON-formatted) sent by the AMX NetLinx Integrated Controller over the network. There are six core commands, as shown below. These commands are assuming a target device ID for the MNC of 1, and a user name and password of "admin", and they assume that at least one MuxLab AV over IP Transmitter and Receiver are present in the target system. If the Target ID for the MNC or if the user name or password differs from the above, then adjust the below commands accordingly.

1. Connect: Perform a connection between AV over IP devices.
   Syntax:
   {*"p_targetId":1,"p_cmd":"connection","p_userName":"admin","p_password":"admin","p_data": [{"portIn":"1","portOut":"1"}]}*

   Description: Perform a connect/disconnect between devices on portIn=1 and on portOut=1\

   Note: One can also use the MAC addresses to establish a connection. In this case the command will look like:
   {"p_targetId":1,"p_cmd":"connection","p_userName":"admin","p_password":"admin","p_data": [{"macRx":"<Rx mac address>","macTx":"<Tx mac address>"}]}

2. Disconnect: Perform a disconnect between AV over IP devices
   Syntax:
   {*"p_targetId":1,"p_cmd":"connection","p_userName":"admin","p_password":"admin","p_data": [{"portIn":"0","portOut":"1"}]}*

   Description: Perform a connect/disconnect between devices on portIn=1 and on portOut=1\

   Note: when using MAC addresses , set the TX MAC address to "00-00-00-00-00-00" :
   {"p_targetId":1,"p_cmd":"connection","p_userName":"admin","p_password":"admin","p_data": [{"macRx":"<Rx max address>","macTx":"00-00-00-00-00-00"}]}

3. Send data to the RS232 port: Send RS232 data (in hexadecimal format) to the AV over IP device RS232 port.
   Syntax:
   {"p_targetId":1,"p_cmd":"send_data_to_rs232","p_userName":"admin","p_password":"admin","p_data":[{"portIn":"1","hexdata":"<hex data>", "ack":0 , "feedback":0}]}

   Note:
   - portOut can also be used
   - only for Muxlab devices 500753/754/755/756/758/759
   - "hex data" maximum length = 390 characters (i.e.: 195 bytes)
   - Set "feedback" = 1, if you want to wait for feedback data to be returned from the RS-232 port
   - Set "ack" =1, if you want to wait for a command acknowledge
   - When using MAC address (Tx or Rx, 00-0B-78-00-77-C3 for example), the syntax will be:
   *{"p_targetId":1,"p_cmd":"send_data_to_rs232","p_userName":"admin","p_password":"admin","p_data": [{"mac":"*00-0B-78-00-77-C3*","hexdata":"AABBFF", "ack":0, "feedback":0}]}*

4. Send data to the IR port: Send IR data (in hexadecimal format) to the AV over IP device IR port
   Syntax:
   {"p_targetId":1,"p_cmd":"send_data_to_ir","p_userName":"admin","p_password":"admin","p_data":[{"portIn":"1" , "hexdata":"<hex data>"}]}'")

   Note:
   - portOut can also be used
   - only for Muxlab devices 500752/753/754 (Tx) and 500755/756 (Tx and Rx)
   - "hex data" maximum length = 390 characters (i.e.: 195 bytes)
   - When using MAC address (Tx or Rx), the syntax will be:
   *{"p_targetId":1,"p_cmd":"send_data_to_ir","p_userName":"admin","p_password":"admin","p_data": [{"mac":"*Tx/Rx mac address*>","hexdata":"<hex data>"}]}*

5. Apply a video wall configuration: Apply an existing (pre-established) video wall configuration
   Syntax:
   {"p_targetId":1,"p_cmd":"vwApplyConfig","p_userName":"admin","p_password":"admin","p_data":[{"vwConfigId":"<vwConfig id number>"}]}

   If desired, one can program the system to allow the change of the video wall source (on the fly) by using the following command in which the portId ("portIn") of the new source is specified:
   {"p_targetId":<systemID>,"p_cmd":"vwApplyConfig","p_userName":"<MNC User Name>","p_password":"<MNC password>","p_data":[{"vwConfigId":"<vwConfig id number>","portIn":"<input port number>"}]}

6. Apply preset : Apply an existing (pre-established) preset connection
   Syntax:
   {"p_targetId":1,"p_cmd":"select_preset","p_userName":"admin","p_password":"admin","p_data":[{"presetId":"<preset id>"}]}