

INTRODUCTION

AMX[®] Driver for LiveCore[™] serie

V3.00

Developed by



ANALOG WAY[®]
Pioneer in Analog, Leader in Digital

ANALOG WAY LIVECORE™

INTRODUCTION

AMX NETLINX

Date: **April 06, 2016**
Driver version: **V3.00**
Compatible with: **LiveCore™ Firmware v04.00.x or above**

INTRODUCTION

This document describes the driver interface provided between an AMX NetLinx system and a LiveCore™ series processor (IP protocol only).

The package provided includes:

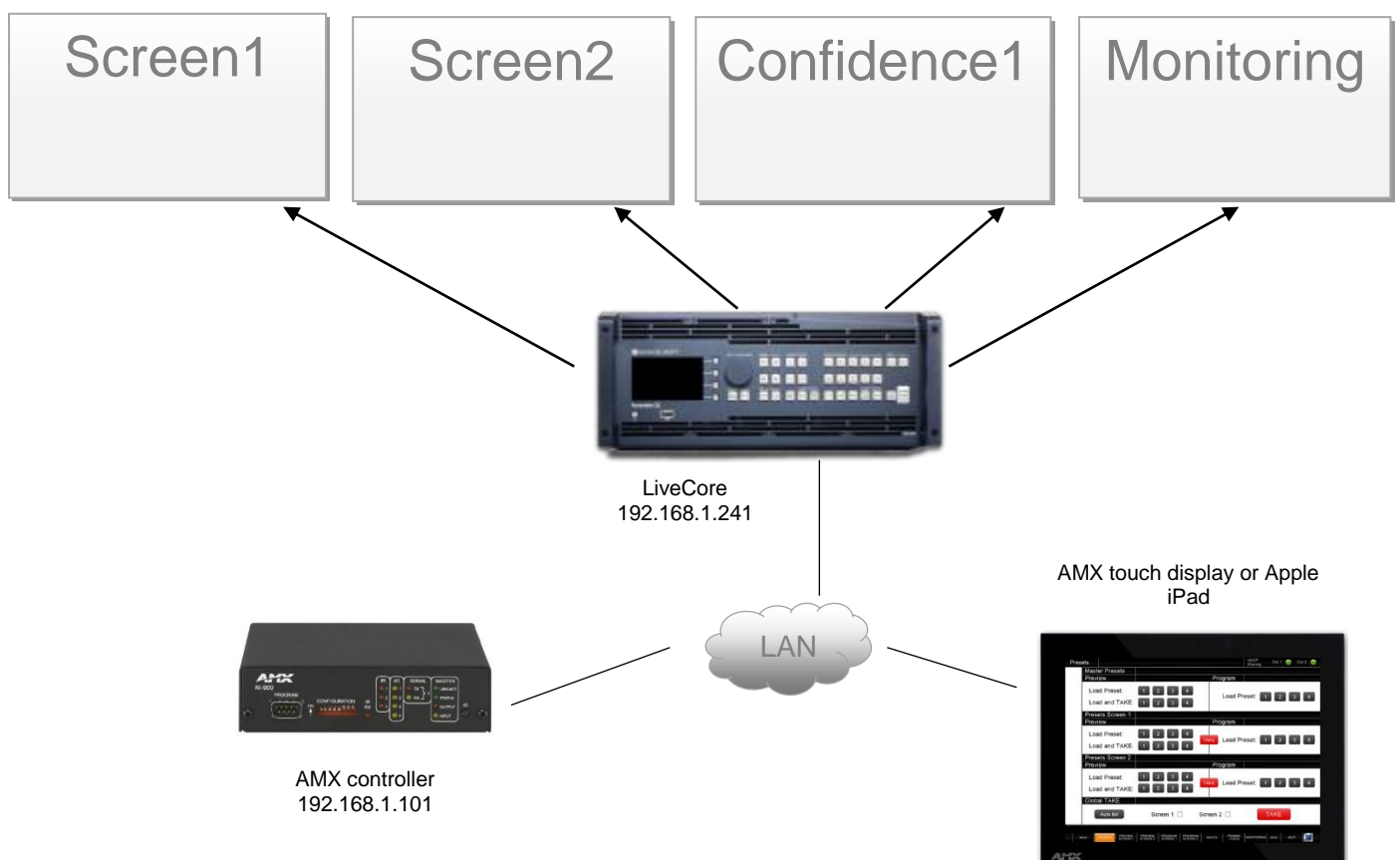
- *.tko library files
- *.axi file for parameters definition
- A Netlinx Studio project sample
- A TPDesign project sample for iPad and a TPDesign sample project for AMX MST-701 touch panel
- Modules documentation

KEY FEATURES

- Compatible with any AMX NetLinx® control systems
- Compatible with any LiveCore™ AV processor
- Load Presets or Master Presets with seamless transitions
- Display sources and logos with dynamic snapshots and properties
- Configure input settings (current plug, HDCP, freeze...)
- Change the native background or the source displayed in a layer
- Transitioning the Preview content to the Program
- Display output and screen properties (HDCP status, labels, layer count...)
- Configure Monitoring output(s) current layout and load Monitoring presets
- Configure Confidence screen(s) current layout and load Confidence presets
- Control device (reboot, shutdown, sleep or wake over LAN)
- Control General-Purpose Input/Outputs (GPIO) and Tally

EXAMPLE SETTINGS

The example provided within the package has been designed to control one LiveCore™ device with two screens (Program), one Confidence screen and one Monitoring output. Before running this sample, please make sure your configuration (as well as the IP addresses) matches the diagram below:



USING THE EXAMPLE

- 1 - Launch the AMX NetLinx Studio program then connect the AMX controller (see menu **Settings / Master Communication Settings** menu). For more information about this topic, read the corresponding AMX documentation.
- 2 - Load the LiveCore.apw project (located in the provided package **Driver** directory) then edit the file LiveCore_User_Definitions.axi to change the default settings. Compile the sample project then transfer it to the AMX controller (see menu **Tools / File Transfer**). If successful, the controller automatically reboots and runs the sample program.
- 3 - Load the TPDesign sample project. Two projects are available, one for iPad, one for an AMX 16/9 touch panel (both located in the provided package **Panels** directory). Once the selected file loaded, modify the dynamic resources used in this project: click the menu **Panel / Resource Manager** then click the **Dynamic Images** tab. Change the IP of all the URLs in the list to match the configured Livecore switcher IP address. Save the project and then transfer it to the panel (see menu **Transfer / Send to Panel**). **Important:** In order to transfer the project to an iPad device, it is mandatory to use the AMX TPTransfer program (and not TPDesign).



IMPLEMENTATION

In order to use this driver in a LiveCore™ AMX program, the programmer must perform the following tasks:

- Include both the LiveCore_User_Definitions.axi and LiveCore_Definitions.axi files in the application project. These two files must appear before the declaration of the driver modules themselves.
- Open the LiveCore_User_Definitions.axi file and configure the MAC address, the IP address and the port of the LiveCore™ processor as well as the different device numbers (touch panel, Livecore machine, AMX controller ...). For each module used in the program, you need to assign the value 1 to the corresponding variable LiveCore_Module_Usage (*Module* is the module name). If an optional module is not used, the value of this variable must remain at 0. In order to avoid overloading the processor, it is strongly advised not to 'load' any optional modules that will never be used by the main program.
- DO NOT CHANGE the file LiveCore_Definitions.axi.
- Include all necessary driver modules in the main program project (see sample program available with this package) :
 - The LiveCore_Proc_Com module is **required** (communication module)
 - The LiveCore_General module is **required** (core module)
 - The LiveCore_Inputs module is optional. Required to control inputs and plugs
 - The LiveCore_Frame_Logo module is optional. Required to retrieve frames and logos properties (availability, dimensions ...)
 - The LiveCore_Filtering module is optional. Required to change Preset filters properties before recalling any Preset or any Master Preset
 - The LiveCore_Master_Presets module is optional. Required to recall Master Presets from memory to the Program or to the Preview (multiple screens)
 - The LiveCore_Screen module is optional. Required to access Livecore screen information. This module must be implemented as many times as the number of screens configured and controlled on the LiveCore™ processor
 - The LiveCore_Screen_Presets module is optional. Required to recall Presets from memory to the Program or to the Preview of a given screen. This module must be implemented as many times as the number of screens configured and controlled on the LiveCore™ processor
 - The LiveCore_Monitoring module is optional. Required to control monitoring outputs. This module can be implemented once or twice (twice when controlling also the monitoring output of a secondary LiveCore™ processor like a LiveCore™ Expander)
 - The LiveCore_GPIO module is optional. Required to manage GPI and GPO on the LiveCore™ processor. This module must be implemented only once, even when controlling a secondary LiveCore™ processor like a LiveCore™ Expander or a linked LiveCore™ processor
 - The LiveCore_Confidence module is optional. Required to manage Confidence screens on the LiveCore™ processor

VARIABLES

Some global variables can be used in the main program to get feedback from the driver.

X is screen number (1 to 8)

Y is layer number (1 to 24)

Screen X _Main_Background_Source	Integer	Source index assigned to Screen X Program background See table below (0 to 41)
Screen X _Main_Layers_Source[Y]	Integer array	Source index assigned to the Layer Y - Screen X Program See table below (0 to 41)
Screen X _Preview_Layers_Source[Y]	Integer array	Source index assigned to the Layer Y - Screen X Preview See table below (0 to 41)

Sources

0	No input
1	Input 1 of In card 1 of Master Device
2	Input 2 of In card 1 of Master Device
3	Input 3 of In card 1 of Master Device
4	Input 4 of In card 1 of Master Device
5	Input 1 of In card 2 of Master Device
6	Input 2 of In card 2 of Master Device
7	Input 3 of In card 2 of Master Device
8	Input 4 of In card 2 of Master Device
9	Input 1 of In card 3 of Master Device
10	Input 2 of In card 3 of Master Device
11	Input 3 of In card 3 of Master Device
12	Input 4 of In card 3 of Master Device
13	Input 1 of In card 1 of Slave Device
14	Input 2 of In card 1 of Slave Device
15	Input 3 of In card 1 of Slave Device
16	Input 4 of In card 1 of Slave Device
17	Input 1 of In card 2 of Slave Device
18	Input 2 of In card 2 of Slave Device
19	Input 3 of In card 2 of Slave Device
20	Input 4 of In card 2 of Slave Device
21	Input 1 of In card 3 of Slave Device
22	Input 2 of In card 3 of Slave Device
23	Input 3 of In card 3 of Slave Device
24	Input 4 of In card 3 of Slave Device
25	Still picture 1 of master device
26	Still picture 2 of master device
27	Still picture 3 of master device
28	Still picture 4 of master device
29	Still picture 1 of slave device
30	Still picture 2 of slave device
31	Still picture 3 of slave device
32	Still picture 4 of slave device
33	Reduced still picture 1 of master device
34	Reduced still picture 2 of master device
35	Reduced still picture 3 of master device
36	Reduced still picture 4 of master device
37	Reduced still picture 1 of slave device
38	Reduced still picture 2 of slave device
39	Reduced still picture 3 of slave device
40	Reduced still picture 4 of slave device
41	Color (or Black) fill the PiP

Connect with us on

