

# INTRODUCTION

## AMX<sup>®</sup> Driver for LiveCore<sup>™</sup> serie

V3.01

*Developed by*



**ANALOG WAY<sup>®</sup>**  
*Pioneer in Analog, Leader in Digital*

# ANALOG WAY LIVECORE™

## INTRODUCTION

### AMX NETLINX

Date: **3 janvier 2017**  
Version du driver : **V3.01**  
Compatible avec : **Firmware LiveCore™ v04.00.x ou supérieur**

## INTRODUCTION

Ce document présente l'interface du driver de communication entre un système AMX NetLinx et un processeur LiveCore™ (communication TCP).

Le package contient :

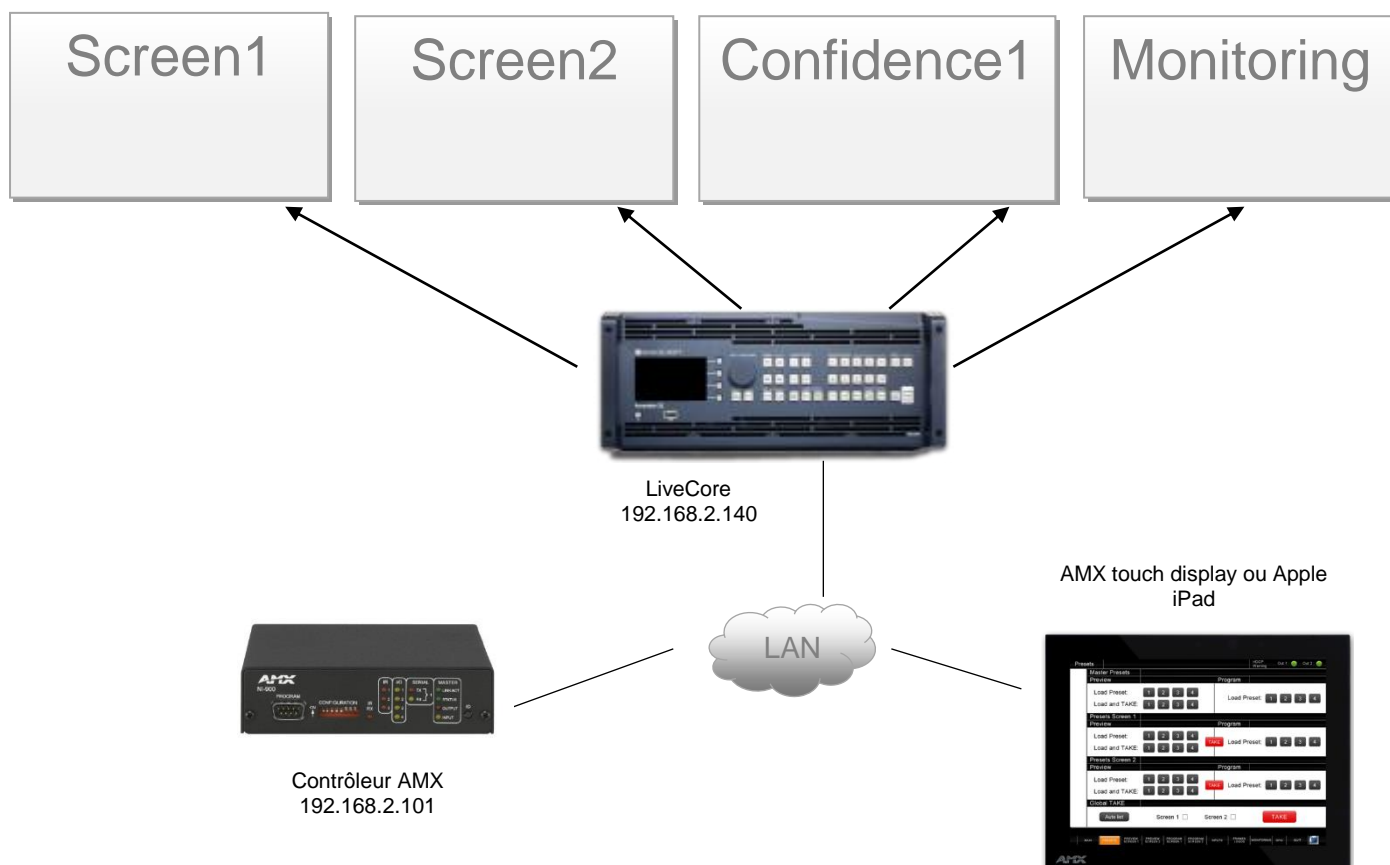
- Les fichiers de librairie \*.tko
- Les fichiers de définition des paramètres \*.axi
- Un exemple de projet Netlinx Studio (LiveCore.apw)
- Un projet exemple TPDesign pour iPad (résolution 1024x768)
- Un projet exemple TPDesign pour panel tactile 7" AMX MST-701 (résolution 1024x600)
- La documentation des modules

## FONCTIONS PRINCIPALES

- Compatible avec les contrôleurs intégrés NetLinx® proposés par AMX
- Compatible avec tous les processeurs AV de la gamme LiveCore™
- Rappel de Presets ou de Master Presets avec transitions Seamless
- Affichage des sources et des logos avec leurs propriétés et leur miniature dynamique associée
- Configuration des entrées du processeur (plug actif, protection HDCP, gel de l'image ...)
- Contrôle du contenu des PIPs et du fond natif
- TAKE : Transition du contenu du Preview vers le Program
- Propriétés des sorties et des écrans contrôlés (état de la protection HDCP, libellés...)
- Configuration de l'affichage de la sortie Monitoring et rappel de presets
- Configuration de l'affichage des écrans Confidence et rappel de presets
- Contrôle du processeur (arrêt, reboot, mise en veille, réveil à distance)
- Contrôle du GPI et des GPO...

## PROJET EXEMPLE

L'exemple fourni dans le package a été conçu pour contrôler un processeur LiveCore™ avec deux écrans (Program), un écran Confidence et une sortie Monitoring. Avant d'exécuter cet exemple, veuillez vérifier que votre configuration (réglages LiveCore, adresses IP, ...) correspond au schéma ci-dessous :



## UTILISATION DU PROJET EXEMPLE

La première étape consiste à lancer le programme AMX NetLinX Studio afin de se connecter à l'automate AMX (cf. menu **Settings/Master Communication Settings**). Pour plus d'informations à ce sujet, lire la documentation AMX correspondante.

L'étape suivante consiste à charger le projet exemple LiveCore.apw (situé dans le répertoire **Driver** du package) puis d'éditer le fichier LiveCore\_User\_Definitions.axi afin de modifier l'adresse IP et l'adresse MAC du processeur LiveCore™ à piloter. Après recompilation du projet exemple, il est alors nécessaire de transférer le programme compilé vers l'automate AMX (cf. menu **Tools/File Transfer**). Si le transfert s'est déroulé normalement, l'automate se réinitialise puis exécute le programme exemple automatiquement.

La dernière étape consiste à charger le projet exemple TPDesign. Deux projets sont disponibles, l'un pour iPad, l'autre pour une tablette tactile AMX 16/9 (les deux projets sont situés dans le répertoire **Panels** du package). Une fois le projet chargé, il est nécessaire de modifier les ressources dynamiques utilisées en cliquant sur le menu **Panel\Resource Manager** puis sur l'onglet **Dynamic Images** et en modifiant l'adresse IP de toutes les URL de la liste (captures live des entrées sur le LiveCore™) avec celle du switcher LiveCore™. Il est alors nécessaire de sauvegarder le projet puis de transférer le projet vers la tablette tactile (menu **Transfer/Send to Panel**). **Attention**, pour transférer le projet vers un iPad, il est nécessaire d'utiliser le programme AMX **TPTransfer** (et non TPDesign).



## IMPLEMENTATION

Pour utiliser le driver LiveCore™ dans un programme AMX, le programmeur doit réaliser les tâches suivantes :

- Inclure obligatoirement les fichiers LiveCore\_User\_Definitions.axi et LiveCore\_Definitions.axi dans le projet. Ces 2 fichiers doivent apparaître avant la déclaration des modules du driver eux-mêmes.
- Modifier le fichier LiveCore\_User\_Definitions.axi afin de l'adapter au programme final et configurer notamment l'adresse MAC, l'adresse IP et le port de la machine LiveCore™ ainsi que les numéros des différents périphériques (panel tactile, machine LiveCore™, automate AMX...).

Pour chaque module du driver utilisé dans le programme il est nécessaire d'attribuer la valeur 1 à la variable correspondante LiveCore\_Module\_Usage (où *Module* correspond au nom du module). Si ce n'est pas le cas, la valeur de cette variable doit rester à 0.

D'une manière générale, pour ne pas surcharger inutilement le processeur, il est vivement conseillé de ne pas 'charger' de modules qui ne seront jamais utilisés par le programme principal et donc de laisser la valeur des variables correspondantes à 0 dans le fichier LiveCore\_User\_Definitions.axi.

- NE PAS MODIFIER le fichier LiveCore\_Definitions.axi !
- Inclure dans le projet du programme principal tous les modules du driver nécessaires (cf. programme exemple disponible avec ce package) :

- Le module LiveCore\_Proc\_Com est **obligatoire** (driver de communication)
- Le module LiveCore\_General est **obligatoire** (module principal)
- Le module LiveCore\_Inputs est optionnel. Nécessaire seulement si le programme doit pouvoir contrôler les entrées et les plugs du processeur LiveCore™
- Le module LiveCore\_Frame\_Logo est optionnel. Nécessaire seulement si le programme doit pouvoir récupérer les propriétés des frames et des logos (disponibilité, dimensions...)
- Le module LiveCore\_Filtering est optionnel. Nécessaire seulement si le programme doit pouvoir modifier les propriétés de filtrage lors du rappel d'un Preset ou d'un Master Preset
- Le module LiveCore\_Master\_Presets est optionnel. Nécessaire si le programme doit offrir la possibilité de rappeler des Master Presets depuis la mémoire du processeur
- Le module "LiveCore\_Screen" est optionnel. Nécessaire pour accéder aux informations d'un écran LiveCore™. Ce module doit être déclaré autant de fois que d'écrans configurés et pilotés sur la machine LiveCore™. Par exemple, si la configuration comporte 2 écrans, le programme devra déclarer deux instances du modules LiveCore\_Screen (cf exemple disponible avec ce package)
- Le module "LiveCore\_Screen\_Presets" est optionnel. Nécessaire si le programme doit offrir la possibilité de rappeler des Presets depuis la mémoire vers le Program ou vers le Preview d'un écran donné. Ce module doit être déclaré autant

de fois que d'écrans configurés et pilotés sur la machine LiveCore™. Par exemple, si la configuration comporte 2 écrans, le programme devra déclarer deux instances du modules LiveCore\_Screen\_Presets (cf. exemple disponible dans le package du driver)

- Le module "LiveCore\_Monitoring" est optionnel. Nécessaire si le programme doit offrir la possibilité de gérer les sorties monitoring. Ce module doit être déclaré autant de fois que de processeurs LiveCore™ contrôlés. Par exemple, si la configuration comporte un processeur principal et un module LiveCore™ Expander, le programme devra déclarer deux instances du modules LiveCore\_Monitoring (cf. exemple disponible dans le package du driver)
- Le module "LiveCore\_GPIO" est optionnel. Nécessaire si le programme doit offrir la possibilité de gérer les entrées/sorties GPIO. Ce module doit être déclaré une seule fois, même dans le cas du contrôle des entrées/sorties GPIO d'un processeur LiveCore™ secondaire (module Expander ou processeur couplé).
- Le module "LiveCore\_Confidence" est optionnel. Nécessaire si le programme doit offrir la possibilité de gérer un ou plusieurs écrans Confidence.

## VARIABLES

Certaines variables sont directement utilisables dans le programme principal pour le retour d'informations.

**X correspond à l'index de l'écran (1 à 8)**

**Y correspond à l'index du layer (1 à 24)**

|  |               |  |
|--|---------------|--|
| Screen <b>X</b> _Main_Background_Source            | Integer       | Numéro de la source de fond (0 à 41) sur le Program de l'écran X (voir table ci-dessous)         |
| Screen <b>X</b> _Main_Layers_Source[ <b>Y</b> ]    | Integer array | Numéro de la source (0 à 41) pour le layer Y sur le Program de l'écran X (voir table ci-dessous) |
| Screen <b>X</b> _Preview_Layers_Source[ <b>Y</b> ] | Integer array | Numéro de la source (0 à 41) pour le layer Y sur le Preview de l'écran X (voir table ci-dessous) |

## Sources

|    |  |
|----|--|
| 0  | Aucune   |
| 1  | Entrée 1 de la carte IN 1 du processeur principal  |
| 2  | Entrée 2 de la carte IN 1 du processeur principal  |
| 3  | Entrée 3 de la carte IN 1 du processeur principal  |
| 4  | Entrée 4 de la carte IN 1 du processeur principal  |
| 5  | Entrée 1 de la carte IN 2 du processeur principal  |
| 6  | Entrée 2 de la carte IN 2 du processeur principal  |
| 7  | Entrée 3 de la carte IN 2 du processeur principal  |
| 8  | Entrée 4 de la carte IN 2 du processeur principal  |
| 9  | Entrée 1 de la carte IN 3 du processeur principal  |
| 10 | Entrée 2 de la carte IN 3 du processeur principal  |
| 11 | Entrée 3 de la carte IN 3 du processeur principal  |
| 12 | Entrée 4 de la carte IN 3 du processeur principal  |
| 13 | Entrée 1 de la carte IN 1 du processeur secondaire |
| 14 | Entrée 2 de la carte IN 1 du processeur secondaire |
| 15 | Entrée 3 de la carte IN 1 du processeur secondaire |
| 16 | Entrée 4 de la carte IN 1 du processeur secondaire |
| 17 | Entrée 1 de la carte IN 2 du processeur secondaire |
| 18 | Entrée 2 de la carte IN 2 du processeur secondaire |
| 19 | Entrée 3 de la carte IN 2 du processeur secondaire |
| 20 | Entrée 4 de la carte IN 2 du processeur secondaire |
| 21 | Entrée 1 de la carte IN 3 du processeur secondaire |
| 22 | Entrée 2 de la carte IN 3 du processeur secondaire |
| 23 | Entrée 3 de la carte IN 3 du processeur secondaire |
| 24 | Entrée 4 de la carte IN 3 du processeur secondaire |
| 25 | Frame 1 du processeur principal                    |
| 26 | Frame 2 du processeur principal                    |
| 27 | Frame 3 du processeur principal                    |
| 28 | Frame 4 du processeur principal                    |
| 29 | Frame 1 du processeur secondaire                   |
| 30 | Frame 2 du processeur secondaire                   |
| 31 | Frame 3 du processeur secondaire                   |
| 32 | Frame 4 du processeur secondaire                   |
| 33 | Logo 1 du processeur principal                     |
| 34 | Logo 2 du processeur principal                     |
| 35 | Logo 3 du processeur principal                     |
| 36 | Logo 4 du processeur principal                     |
| 37 | Logo 1 du processeur secondaire                    |
| 38 | Logo 2 du processeur secondaire                    |
| 39 | Logo 3 du processeur secondaire                    |
| 40 | Logo 4 du processeur secondaire                    |
| 41 | Couleur unie (ou noir)                             |

Connect with us on

