ANALOG WAY MIDRA

INTRODUCTION

AMX NETLINX

Date: **16 Octobre 2017**
Version du driver: **V1.11**
Testé avec: **Midra Firmware v02.00.15**

INTRODUCTION

Ce document présente l'interface du driver de communication entre un système AMX NetLinx et une machine MIDRA (communication TCP).

Le package contient :

- Les fichiers de librairie *.tko
- Les fichiers de définition des paramètres *.axi
- Un exemple de projet Netlinx Studio (Midra.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

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 Midra.apw (situé dans le répertoire **Driver** du package) puis d'éditer le fichier Midra_User_Definitions.axi afin de modifier l'adresse IP du switcher de la gamme Midra 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). Il est alors nécessaire 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 Midra dans un programme AMX, le programmeur doit réaliser les tâches suivantes :

- Inclure obligatoirement les fichiers Midra_User_Definitions.axi et Midra_ Definitions.axi dans le projet. Ces 2 fichiers doivent apparaitre avant la déclaration des modules du driver eux-mêmes.
- Modifier le fichier Midra_User_Definitions.axi afin de l'adapter au programme final puis configurer notamment l'adresse IP et le port de la machine Livecore ainsi que les numéros des différents périphériques (panel tactile, machine Midra, automate AMX…).
  Pour chaque module du driver utilisé dans le programme il est nécessaire d'attribuer la valeur 1 à la variable correspondante Midra_*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 Midra_User_Definitions.axi.
- NE PAS MODIFIER le fichier Midra_Definitions.axi !
- Inclure dans le projet du programme principal tous les modules du driver nécessaires (cf. programme exemple disponible avec ce package) :

  - Le module Midra_Proc_Com est **obligatoire** (driver de communication)
  - Le module Midra_General est **obligatoire** (module principal)
  - Le module Midra_Inputs est optionnel. Nécessaire seulement si le programme doit offrir la possibilité de lire les informations d'une entrée ou de changer de connecteur sur une entrée
  - Le module Midra_Audio est optionnel. Nécessaire seulement si le programme doit pouvoir contrôler les entrées et sorties Audio et régler le volume correspondant.
  - Le module Midra_Frame_Logo est optionnel. Nécessaire seulement si le programme doit pouvoir récupérer les propriétés des frames ou des logos (disponibilité, dimensions…)
  - Le module "Midra_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 Midra. Par exemple, si la configuration comporte 2 écrans, le programme devra déclarer deux instances du modules Midra_Screen (cf exemple disponible avec ce package)

- Le module “Midra_Screen_Presets” est optionnel. Nécessaire si le programme doit offrir la possibilité de rappeler des Presets sur le Main ou sur le Preview d’un écran donné. Ce module doit être déclaré autant de fois que d’écrans configurés et pilotés sur la machine Midra. Par exemple, si la configuration comporte 2 écrans, le programme devra déclarer deux instances du modules Midra_Screen_Presets (cf. exemple disponible dans le package du driver)

Remarque : Le driver détecte automatiquement le type de machine Midra ainsi que ses possibilités techniques. Cela signifie que certaines fonctionnalités du driver pourront ne pas être disponibles en fonction de configuration matérielle choisie.

VARIABLES UTILISATEURS

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

X correspond au numéro de l’écran (1 à 2)
Y correspond au numéro du layer (1 à 8)
Z correspond au numéro de l’entrée vidéo (1 à 10)

Screen**X**_Main_Layers_Source[**Y**]	Integer array	Numéro de la source (0 à 10) pour le layer Y de l’écran X sur le Main (voir table ci-dessous)
Screen**X**_Preview_Layers_Source[**Y**]	Integer array	Numéro de la source (0 à 10) pour le layer Y de l’écran X sur le Preview (voir table ci-dessous)
Active_Video_Mode	Integer	Mode actif 1 : Mixer 2 : Matrix 3 : Quadravision
In_Active_Plug_FB[**Z**]	Integer array	Plug actif pour l’entrée Z 0 : analogique (HD15) 1 : DVI 2 : SDI 3 : HDMI 4 : HDBaseT

Sources (en fonction du type de layer)

0	Aucune
1	Entrée ou Frame 1
2	Entrée ou Frame 2
3	Entrée ou Frame 3
4	Entrée ou Frame 4
5	Entrée ou Frame 5
6	Entrée ou Frame 6
7	Entrée ou Frame 7
8	Entrée ou Frame 8
9	Entrée ou Frame 9
10	Entrée ou Frame 10
11	Couleur ou noir (remplissage du PIP)

Analog Way Worldwide
Europe, Middle East & Africa: +33 (0)1 81 89 08 60
The Americas: +1 212 269 1902
Asia Pacific: +65 6292 5800
www.analogway.com