AMX modules for
tvONE CORIOmaster Video Processor

Integration Guide

1 – Summary

This document will assist AMX programmers and installers with the integration of this module into their AMX program.

The module was designed to control a tvONE CORIOmaster and CORIOmaster mini video processor.

2 – Resources and Assumptions

2.1 – Supported Systems

The module has been designed for use with an AMX control system with Ethernet capability. A Demo project is provided. This project is not intended for end users, but is provided so that all features of the module can be demonstrated and exercised.

2.2 – Software and Firmware

This module was developed using the following firmware and software versions. Ensure you are using the same version or newer.

AMX NetLinx Studio : 3.2.0.418

tvONE CORIOMAX V1.300B34.P4

2.3 – Assumptions

It is assumed that you already have a good understanding of AMX Programming and integration. Knowledge of TCP/IP networking would also be beneficial.

It is assumed that the tvONE device is installed and functioning correctly. The tvONE device must be connected to the same LAN as the AMX processor for TCP/IP control.

3 – AMX Modules

3.1 – Module Format

The tvONE project module has been provided in a zip package called "tvONE_CORIOmaster_AMX_[version].zip". Copy the "tko" file to your project folder and add it in to your project. A demo file has been provided for demonstration / evaluation of the module and is not intended for direct integration into your project.

3.2 – Module Features

The module supports the functions listed in the table below. Please consult your tvONE device documentation to check which functions are supported by your device, and for the maximum preset value allowed.

Function	**Notes**
SET WINDOW	Changes a window's source
LOAD PRESET [Preset Number]	Load a preset where [Preset Number] = 1 to 56.

3.3 – Using the module in your program

Copy the "tvONE_CORIOmaster.tko" file from the zip file to your AMX project folder and add it to your project. Next, add parameter definitions and a define_module line to your master source file, similar to the following:

```
define_device
  dvTvONE_CORIOmaster = 0:4:0
  vdvTvONE_CORIOmaster  = 33001:1:1
  dvTP  = 10001:1:1
define_variables
  char
  TvONE_CORIOmaster_IP[15] = '172.16.105.3',
  TvONE_CORIOmaster_USERNAME[100] = 'admin',
  TvONE_CORIOmaster_PASSWORD[100] = 'adminpw';
  integer
  TvONE_CORIOmaster_PORT = 10001;
define_module 'tvONE_CORIOmaster' TVONE_CORIOMASTER(
  vdvTvONE_CORIOmaster,
  dvTvONE_CORIOmaster,
  TvONE_CORIOmaster_IP,
  TvONE_CORIOmaster_PORT,
  TvONE_CORIOmaster_USERNAME,
  TvONE_CORIOmaster_PASSWORD)
```

The module parameters are as follows:

Parameters	**Details**
`vdvTvONE_CORIOmaster`	The virtual device used for communication between the program and the switcher module. Define a device in the virtual device range.
`dvTvONE_CORIOmaster`	The physical device for the switcher. Define a device in the IP device range.
`TvONE_CORIOmaster_IP`	The IP address of the CORIOmaster.
`TvONE_CORIOmaster_PORT`	The TCP/IP port of the CORIOmaster (default port = 10001).
`TvONE_CORIOmaster_USERNAME`	The username required to access the CORIOmaster (default = "admin").
`TvONE_CORIOmaster_PASSWORD`	The password required to access the CORIOmaster (default = "adminpw").

3.4 – Testing the modules using the supplied test harness

The supplied demo test harness may be used to test the module. The test harness includes a UI module which controls the CORIOmatrix with commands sent via the virtual device. The commands are sent to the module using **send_command**. For instance:

send_command vdvTvONE_CORIOmaster, 'SET-WINDOW_1=s2i1';

...will switch window 1 to slot 2, input 1 (where **s2i1** = slot 2, input 1).

send_command vdvTvONE_CORIOmaster, 'LOAD-PRESET_4';

...will load preset number 4.

All commands supported by the module are listed below.

Command	**Details**
`SET-WINDOW_<window>=s<slotNum>i<input>;`	Changes the input displayed in a particular window. **Window** = the window number **slotNum** = the slot number occupied by the input card **input** = the input number to be displayed
`LOAD-PRESET_<number>;`	Loads a preset defined by number. The preset number must be within the range supported by the device (e.g. 1 – 50).
`GET-VERSION_?;`	Requests the module version
`GET-STATUS_?;`	Requests the current status of the module

The test harness supplied allows the user to change window sources, select presets and also requests status.

4 – Feedback from the module

The following table contains feedback received by the module:

Command	**Details**
`VERSION <value>;`	Module version
`STATUS <number>=<explanation>;`	The current status of the device. **Number** = 1 (Good and waiting for commands) **Number** = 2 (Processing commands. Any commands sent will be queued) **Number** = 3 (There's a problem. Most likely a bad command or connection)
`PERSETNAME <number>=<name>;`	**Number** = present number **Name** = preset name

5 – Troubleshooting and tips

5.1 – Communication failure

Check the IP address of the CORIOmaster is correctly entered in the module.

If the commands sent to the CORIOmaster do not seem to be working, check that you can ping the device from a pc.

The CORIOmaster does not support multi-users and therefore can only support one connection at a time. If the module is failing to connect to the device, check no other systems are currently connected.

After resetting the device or a communication failure, it may take 15 – 30 seconds for the connection to be re-established.

If the source requested is not selected, check that the configuration of the CORIOmaster supports the selected input number.

If the preset requested is not selected, check that the preset exists on the CORIOmaster by manually selecting it.