

NetLinx Module Interface Documentation

for

ATLONA

AT-PRO2HD1616M
16x16 HDMI Matrix Switcher,

AT-PRO2HD88M
8x8 HDMI Matrix Switcher,

AT-PRO2HD44M
4x4 HDMI Matrix Switcher

Implementation

To interface to the Atlona HDMI Matrix Switcher module:

1. Define the device ID for the Matrix Switcher that will be controlled. This is the “physical” control port i.e. RS-232 port
2. Define the virtual device ID that the Matrix Switcher’s control module will use to communicate with the main program and User Interface. Typical device ID’s start 33001.

3. Tie the module using the DEFINE_MODULE command.

Example: DEFINE_MODULE 'Atlona AT-PRO2HD1616M, Rev 1.0' nMdlComm (dvAtlona, vdvAtlona, sSwtName)

4. Create a “friendly” name for use when debugging*

Example: CHAR sSwtName[30] = 'Main Switcher'

*This is an optional value. Use null string if not using a friendly name for the switcher.

An example of how to do this is shown below.

```

DEFINE_DEVICE
dvAtlona           = 5001:1:0           // RS232, Atlona HDMI Matrix
vdvAtlona          = 33001:1:0          // Virtual Device for Atlona HDMI Matrix module
dvAvPanel          = 10001:1:0          // Main control panel

DEFINE_VARIABLE

CHAR sSwtName[30] = 'Main Switcher'      // Friendly name for use with Diagnostics

DEFINE_START

DEFINE_MODULE 'Atlona AT-PRO2HD1616M, Rev 1.0' nMdlComm ( dvAtlona, vdvAtlona, sSwtName, sFileName )

```

Command Control

The module controls the Matrix Switcher via command events (NetLinx command SEND_COMMAND) sent to the virtual device. The commands supported by the COMM module are listed below. All commands use a terminator character of \$FF (hex).

Basic 1 to 1 route:

SEND_COMMAND vdvAtlona,"" SWITCH_I_O=1,7',\$FF"

Where **1** = Input 1; **7** = Output 7 and \$FF = command terminator.

NOTE: Input and output should be separated by a comma “,”.

Multi Output route:

SEND_COMMAND vdvAtlona,"" MULTI_SWT_CMD=1,2,4,6,8,16',\$FF"

Where **1** = Input 1; **2,4,6,8,16** = Outputs 2,4,6,8, 16 and \$FF = command terminator.

NOTE: All outputs should be separated by a comma “,”.

RS-232 Command to controlled device connected to an Atlona AT-PRO2HDREC:

SEND_COMMAND vdvAtlona,"RS-232_CMD=PORT1,ka 1 ff',\$0D,\$FF"

Where **PORT1** = Output on the Atlona switcher that has the AT-PRO2HDREC connected and ‘**ka 1 ff**’,**\$0D** = Device command and \$FF = command terminator.

NOTE: Command to the device should be formatted as if it were being sent out a standard RS-232 port. If the device needs a command terminator i.e. Carriage Return “\$0D”, this must be placed in front of the SEND_COMMAND terminator \$FF

Channels

The control module uses certain channels for both control and feedback and some are used for diagnostics. Those that are used for feedback or debugging only will be indicated in the table below.

Channel	Description
27	PULSE: To Power On the switcher. ON: Switcher ON state feedback
1-8	PULSE: To trigger preset #n
11-18	PULSE: To save current switcher I/O as a preset.
28	PULSE: To Power Off the switcher. ON: Switcher OFF state feedback
29	PULSE: This will reset to the switcher I/O to the 1 to 1 mode. i.e. i1 to o1, i2 to o2, i3 to o3, etc.
101 - 116 (16x16) 101 - 108 (8x8) 101 - 104 (4x4)	ON: Channel flag used to allow bar graph control of output.
249	ON: Turns on polling for status OFF: Turns off polling for status
250*	ON: Allows module error messages to be viewed via Virtual device. (Debug only)
251*	ON: Allows viewing of commands transmitted by the module to the switcher. (Debug only)
252*	ON: Allows viewing of commands received by the module from the switcher. (Debug only)
255	ON: Switcher is on (Feedback only) OFF: Switcher is off (Feedback only)

Table 1 - Virtual Device Channels

*These channels can also be used on the “physical” device address for use in debugging via a Terminal/Telnet session.

Levels

The levels supported by the module are listed below. These levels are associated with the virtual device(s) and are independent of the levels associated with the touch panel device. These values can be used for feedback or control. If used for control see channel list above.

Level	Description
1-16	Output Level value to indicate input (0..16)
	Example. Level value of 5 = input 5 for the respective output level number.

Table 1 - Virtual Device Level Events

Command Feedback

The module provides feedback to the main NetLinx code via device levels or via “String” Data Events. When receiving data from the switcher, the commands are preceded with a header that will describe the incoming data type.

Output Level feedback:

Each output has a level associated with it to indicate what input is currently routed to it. The level values are 1-16 which represents the available inputs.

String Feedback:

‘DIAG=’ Commands preceded by this header are meant for debugging any module errors and can be turned off via the debug channel on the virtual device.

‘TX=’ Commands preceded by this header are meant for debugging the transmitted commands to the switcher and can be turned off via the transmit channel on the virtual device.

‘RX=’ Commands preceded by this header are meant for debugging the received commands from the switcher and can be turned off via the transmit channel on the virtual device.

Device Notes

The switcher can be setup for constant polling of I/O and power status, by turning on channel 249 on the virtual device, the system will automatically start polling at a preset interval. (Default is set to 2 second intervals.)

All commands are sent via a command queue that keeps a “pace” for the switcher so that several commands can be sent without overflowing the switcher’s device buffer.

Default switcher baud rate: 115200,n,8,1

AMX processor used for testing: NI-3100

AMX processor firmware used for testing: v3.50.340

Protocol Notes

The two status query commands are for Power status and I/O status. The latter is for ALL input outputs.

Product used for testing: AT-PRO2HD1616M

Product firmware used for testing: v1.1.19

Additional Device Commands Through The Module

The module allows additional device commands to be sent. This is done via the ‘PASSTHRU=’ command, which allows protocol strings to be passed through the module. The device-specific protocol must be known in order to use this feature.

SEND_COMMAND vdvAtlona, "PASSTHRU=ALL#', \$0D"

NOTE: The command should be terminated with the carriage return “\$0D” terminator and not the SEND_COMMAND terminator “\$FF”.

Revision History

Date	Initials	Comments
12.02.2012	DJL	V1.0 Initial Release
12.28.2012	DJL	V1.1 Added preset save/recall and switcher reset function
03.12.2013	DJL	V1.2 Resolved issues with panel diagnostics feature