

NetLinx Module Interface Documentation

for

ATLONA

AT-H2H-88M
8x8 HDMI Matrix Switcher,

AT-H2H-44M
4x4 HDMI Matrix Switcher

Module developed for Atlona by Front Side Solutions.

www.frontsidesolutions.com

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-H2H88M, 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          = 0:3:0          // IP, Atlona HDMI Matrix
                OR
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
CHAR sSetIpAddr[15] = '192.168.50.122' // IP address if using IP control
INTEGER nSetIpPort = 23             // IP Port for IP control (Default is Telnet Port 23)

DEFINE_START

DEFINE_MODULE 'Atlona AT-H2H88M, Rev 1.2' 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 “,”.

IP Commands:

SEND_COMMAND vdvAtlona,""IPCONFIGGET=1',255"

Where 1 = arbitrary value. This will send the IP info query command to the switcher. The information can be parsed out of the virtual device ‘string’ event or as in the sample file, via the predefined variables:

sIpAddress
sIpSubnet_
sIpGateway
sIpCtlPort

SEND_COMMAND vdvAtlona," IPUSERINFOGET=1',255"

Where 1 = arbitrary value. This will send the User login info query command to the switcher.

SEND_COMMAND vdvAtlona," IPPORTSET=1',255"

Where 1 = Any value from 1-65535

SEND_COMMAND vdvAtlona,""IPADDRSET=192.168.1.16:255.255.255.0:23',\$FF"

Where 192.168.1.16 = New IP Address; 255.255.255.0 = Subnet mask; 23 = IP port number (i.e. Telnet) and \$FF = command terminator.

NOTE: IP parameters should be separated by a colon “:”.

SEND_COMMAND vdvAtlona," IPLOGIN=ON',255" or
 SEND_COMMAND vdvAtlona," IPLOGIN=OFF',255"
 Where ON = Broadcast mode on; OFF = Broadcast mode off

NOTE: This command is redundant as turning on or off virtual device channel 228 will achieve the same result. If IP login is enabled, only 1 IP connection is allowed, if login is disabled, up to 4 simultaneous connections are allowed. (This may change with firmware updates but as of V1.0.10 this is the case)

SEND_COMMAND vdvAtlona," IPUSERIDSET =1',255"
 Where 1 = Any value between 1 and 3. All other values will default to user ID 1

NOTE: This command sets the login ID that will be used for IP connections; it is only valid if the login feature is active.

SEND_COMMAND vdvAtlona," IPADDUSER=1;Username:Password',255"
 Where 1 = User ID number (Maximum value is 3); Username = User name you wish to register;
 Password = Password you wish to assign to this user.

NOTE: You are allowed to add up to 3 different users, if a password needs to be changed, the user would have to be deleted and then re-added with the updated password. To delete a user, see IPDELUSER below. Username and Password are limited to 20 alpha-numeric characters each.

SEND_COMMAND vdvAtlona," IPDELUSER=1',255"
 Where 1 = User ID number (Maximum value is 3)

NOTE: The method to add or delete a user is actually name based, and therefore the user ID number is relative to what is stored in a username variable array at the index number of the user ID used. If the user name in the variable does not match what is on the processor, a query for username and password information will need to be sent. See 'IPUSERINFOGET' above.

SEND_COMMAND vdvAtlona," IPBCAST=ON',255" or
 SEND_COMMAND vdvAtlona," IPBCAST=OFF',255"
 Where ON = Broadcast mode on; OFF = Broadcast mode off

NOTE: This command is redundant as turning on or off virtual device channel 229 will achieve the same result. When set to on, the responses from the switcher get sent to all serial and IP connections that are currently being made to the switcher. When set to off, you will only see switcher responses from your particular serial or IP connection.

SEND_COMMAND vdvAtlona," IPDHCPSET=ON',255" or
 SEND_COMMAND vdvAtlona," IPDHCPSET=OFF',255"
 Where ON = DHCP mode on; OFF = DHCP mode off

NOTE: This command is redundant as turning on or off virtual device channel 231 will achieve the same result.

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.
201-208 (8x8) 201-204 (4x4)	ON: Sets the Audio Return Channel on OFF: Sets the Audio Return Channel off
227	ON: Turns on Front Panel Lock OFF: Turns off Front Panel Lock
228	ON: Turns on IP Log in function OFF: Turns off IP log in function
230	ON: Turns on IR receiver on the front panel OFF: Turns off IR receiver on the front panel
229	ON: Turns on Broadcast mode OFF: Turns off Broadcast mode
231	ON: Turns on DHCP mode OFF: Turns off DHCP mode
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

Default switcher IP Port: 23 (Telnet). IP port can be changed to any value from 1-65535.

AMX processor used for testing: NI-3100

AMX processor firmware used for testing: v3.50.340

Protocol Notes

The four status query commands are for Power status, I/O status, DHCP and ARC status. However, upon initialization of the module i.e. Device online event, four additional queries are made; Login status, Broadcast mode status, IP info status, and User list info, These last 4 only get sent on initialization, but some of these queries can also be sent with send commands and/or using the PASSTHRU command.

Many of the IP commands will cause the IP socket to close. It is recommended that only experienced user attempt to use IP command functions while using an IP connection, as loss of switcher control/connectivity can occur. It is preferred that IP setup be done via a serial connection first and then use the IP connection once everything has been set. The module is written to allow IP functions to be used via the serial port, so using the module for IP setup is possible.

Product used for testing: AT-H2H44M

Product firmware used for testing: V1.0.10

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
02.28.2013	DJL	V1.1 Initial Release
03.11.2013	DJL	V1.2 Finalized IP functions for firmware v1.0.10