



HDM-C6 Series HDMI-over-IP Video Systems

AMX Module User Guide

Version 1.03

Driver developed by



Introduction

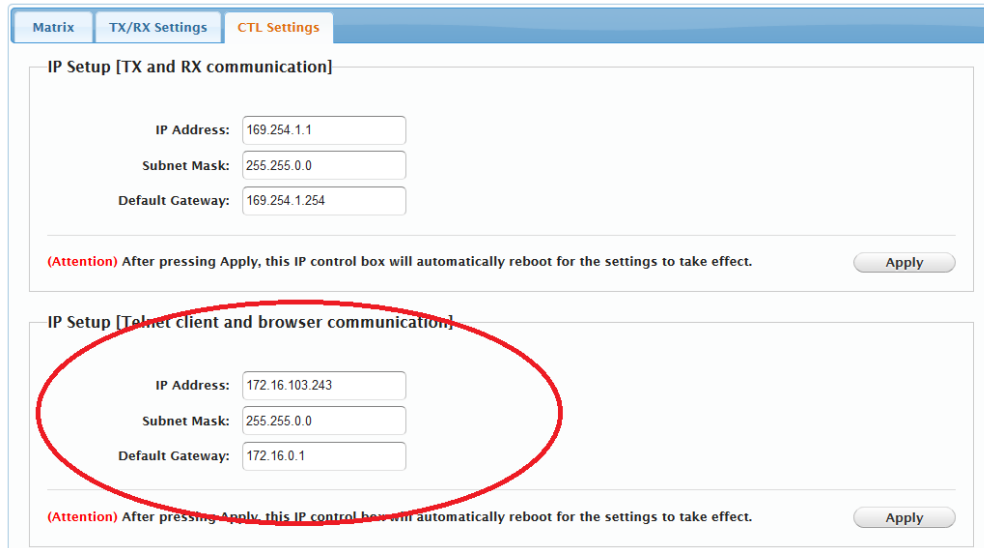
This driver has been designed to provide two-way control of Avenview HDM-C6 series HDMI-over-IP video systems, via TCP/IP. Three separate driver files are included with the packages, providing support for the original HDM-C6VWIP series products, as well as the newer HDM-C6MXIP and HDM-C6MVIP ranges.

Avenview Configuration

It is recommended that the Avenview system be installed, configured and tested by a suitably qualified engineer, according to Avenview documentation, prior to integration with this driver. Some additional, specific configuration is required to ensure correct operation of the driver:

The Avenview Control Interface Telnet Client must be configured with a static IP address in the same range as the AMX processor in order for the two to communicate:

1. Enter the IP address of the IP Control Box into the web browser of a computer connected to the same network, to display the Web Interface (the default IP address is "192.168.11.243" and default password is "admin").
2. Choose the **CTL Settings** tab.
3. Enter the static IP address information into the **IP Setup [Telnet client and browser communication]** section, and click **Apply**.



Matrix TX/RX Settings **CTL Settings**

IP Setup [TX and RX communication]

IP Address: 169.254.1.1

Subnet Mask: 255.255.0.0

Default Gateway: 169.254.1.254

(Attention) After pressing Apply, this IP control box will automatically reboot for the settings to take effect. Apply

IP Setup [Telnet client and browser communication]

IP Address: 172.16.103.243

Subnet Mask: 255.255.0.0

Default Gateway: 172.16.0.1

(Attention) After pressing Apply, this IP control box will automatically reboot for the settings to take effect. Apply

Figure 1: Avenview CTL Settings

It is additionally necessary to configure an **Alias** (name) for each transmitter (input) device and each receiver (output) device. Access the web interface as described above, this time choosing the **TX/RX Settings** tab. Note that the current device names are displayed in **Device Settings** at the top of the page. Select a device to display its current configuration:

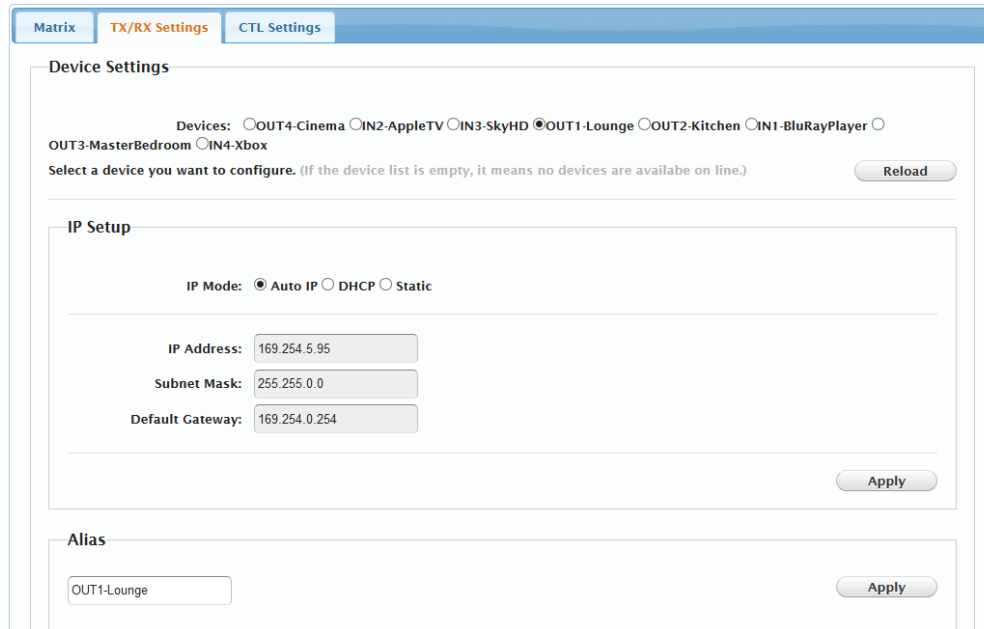


Figure 2: Avenview TX/RX Settings

You can edit the device name in the **Alias** field, clicking Apply when done. Note that the name must comply with the following conventions:

For Transmitter (input) devices: *IN[number]-[name]*
For Receiver (output) devices: *OUT[number]-[name]*

It is important that each name begins with "IN" or "OUT", which is then followed by the input or output number. You can then optionally add a hyphen (-) followed by an appropriate description for the device (note that no spaces are allowed). For example, in the screenshot above, the first input is named *IN1-BluRayPlayer*, equally valid is just *IN1*.

Module Installation & Configuration

Three driver files are included with the zip package, corresponding to the three different product ranges:

Avenview_C6MVIP.tko
Avenview_C6MXIP.tko
Avenview_C6VWIP.tko

Extract the appropriate driver to the folder where your project is stored and then add it into your project. Next, add a `define_module` line to your master source file, similar to the following:

```
define_module 'Avenview_C6VWIP' AVMatrix( vdvAVMatrixModule,  
dvAVMATRIX_SWITCH,  
AVMATRIX_SWITCH_IP,  
MX_VIRTUAL_SERIAL_OUTPUTS,  
MX_VIRTUAL_SERIAL_INPUTS,  
favourite );
```

Note: for a full, working example, please see the demonstration program that comes bundled with the module.

Configuring Module Parameters

The following parameters are defined in the module:

Parameter	Description
<code>vdvAVMatrixModule</code>	Virtual device for the matrix
<code>dvAVMATRIX_SWITCH</code>	Ethernet port
<code>AVMATRIX_SWITCH_IP</code>	The IP address on the LAN for the Avenview IP control box
<code>MX_VIRTUAL_SERIAL_OUTPUTS</code>	Virtual device array for the serial port of each receiver
<code>MX_VIRTUAL_SERIAL_INPUTS</code>	Virtual device array for the serial port of each transmitter
<code>favourite</code>	Persistent char array containing preset configurations of the system saved as "favourites". Persists after reboot.

Table 1: Module Parameters

Module Commands

For a complete example of the usage of this module, see the accompanying demonstration project included in the zip package. The module supports the following commands, sent using **send_command**:

Command	Description
SW [outputs]=[input];	<p>This command is used to switch outputs. For example:</p> <p>SW 1=4; Single Switch (output 1 to input 4) SW 1,2,4,7,8=4; Switch multiple outputs to same input</p> <p>So, to switch outputs 1,2,3,4,5 to input 2....</p> <p>SEND_COMMAND vdvAVMatrixModule,"'SW 1,2,3,4,5=2;'"</p>
FAV [selected outputs]=[favourite];	<p>Requests the switch to save the configuration of an output or list of outputs as a favourite (up to a maximum of 8 individual favourites can be stored). Note that video wall configurations cannot be saved. For example:</p> <p>FAV 4=1;</p> <p>...saves the current configuration of output 4 as favourite number 1.</p> <p>FAV 1,7,19=5;</p> <p>...saves the current configuration of outputs 1, 7 and 19 as favourite number 5.</p> <p>Note: attempting to save a configuration as a favourite that has already been defined will overwrite the existing information.</p>
RECALL [favourite];	<p>Recalls a favourite by number, which has been defined using the FAV command. For example:</p> <p>RECALL 5;</p> <p>...recalls the configuration stored as favourite number 5.</p>
VW [wall name]>[x]x[y]:[outputs]=[input];*	<p>This command is used to configure a single host video wall. The programmer must define a name for the wall (which other commands may refer to), the wall size, the outputs used to create the wall and the input used as the host. For example:</p> <p>VW wall1>2x2:1,2,3,4=1;</p>

	<p>...or...</p> <pre>VW wall1>2x2:1-4=1;</pre> <p>...or...</p> <pre>VW wall1>2x2:1,2-4=1;</pre> <p>...creates a video wall named "wall1", with a 2x2 configuration (i.e. 4 screens), using outputs 1, 2, 3 and 4, with input 1 as the source.</p>
<pre>SW_VW [wall name]=[input];*</pre>	<p>This command is used to switch a video wall to a given input. For example:</p> <pre>SW_VW wall1=4;</pre> <p>...switches the video wall named "wall1" to use input 4.</p>
<pre>BEZELGAP [wall name]>[TV size x]x[TV size y]:[screen size x]x[screen size y];*</pre>	<p>This command is used to define the size of the TV frame (video edge) to correct for large bezel screens in video wall mode. The programmer must specify the wall name (as defined in the video wall command), the overall TV size in mm, and the actual screen size in mm. For example:</p> <pre>BEZELGAP wall1>600x450:550x400;</pre> <p>...applies a bezel gap to the video wall named "wall1", for a TV with an overall size of 600x450mm, and an actual screen size of 550x400mm.</p>
<pre>PICPARAM [wall name]>[horizontal shift],[vertical shift]:[horizontal scale],[vertical scale]:[tearing delay],[outputs];*</pre>	<p>This command allows you to make adjustments to the screen appearance of specified outputs within a video wall. The programmer must specify the wall name (as defined in the video wall command), horizontal and vertical "shift" values (1 unit = 8 pixels), horizontal and vertical "scale" values (1 unit = 1 row or column), tearing delay (in microseconds) and the outputs to which the parameters are to be applied. For example:</p> <pre>PICPARAM wall1>0,10:1,1:12500:1,2-4;</pre> <p>...shifts the image 0 pixels horizontally and 80 pixels vertically, scales the image up by 1 row and 1 column, and adjusts the tearing delay by 12,500 microseconds or outputs 1 to 4.</p>
<pre>VW2 [wall name]>[x]x[y]:[outputs] =[input]:[outputs] =[input];*</pre>	<p>This command is used to configure a multi host video wall. The programmer must define a name for the wall (which other commands may refer to), the wall size, as well as the outputs and inputs used for each row of the video wall. For example:</p> <pre>VW2 wall2>2x3:1,2=1:3,4=2:5,6=3;</pre> <p>...creates a video wall named "wall2", with a 2x3 configuration (i.e. 6 screens), displaying input 1 via outputs 1 and 2 for the</p>

	top row, input 2 via outputs 3 and 4 for the middle row and input 3 via outputs 5 and 6 for the bottom row.
SW_VW2 [wall name]=[input], }+; *	<p>This command is used to switch a multi-host video wall to a given input. For example:</p> <pre>SW_VW2 wall1=4,1,0;</pre> <p>...switches the first row of the video wall named "wall1" to use input 4, the second row to use input 1 and the third row to remain unchanged (note that "0" is used to specify no change to the input displayed on a particular row).</p>
REINITIALISE;	This command can be sent to update the AMX system with any changes made to device names in the Avenview system.
RESET_INPUT [input];	<p>Use this command to reset an input previously used in a Multi Host Video Wall command, so that it can be switched as a normal input once again. For example:</p> <pre>RESET_INPUT 1;</pre> <p>...resets input 1.</p>
CEC_ON [outputs]; **	<p>Send a CEC command to the display connected to an output. For example:</p> <pre>CEC_ON 1,2,3;</pre> <p>...sends the CEC "on" command to the displays connected to outputs 1, 2 and 3.</p>
CEC_STANDBY [outputs]; **	<p>Send a CEC command to the display connected to an output. For example:</p> <pre>CEC_STANDBY 1,2,3;</pre> <p>...sends the CEC "standby" command to the displays connected to outputs 1, 2 and 3.</p>

Table 2: Module Commands

* These commands are NOT supported in the *Avenview_C6MXIP* driver.

** These commands are NOT supported in the *Avenview_C6VWIP* driver.

In addition to the above commands, the serial ports on the Avenview transmitters and receivers can be used as standard AMX serial ports, via the use of the virtual serial devices. For example:

```
send_string MX_VIRTUAL_SERIAL_OUTPUTS[1], 'hello';
```

....sends the text "hello" to the serial port of the receiver connected to output 1.

```
send_string MX_VIRTUAL_SERIAL_INPUTS[1], 'hello';
```

....sends the text "hello" to the serial port of the transmitter connected to input 1.

To define the settings for the serial port at runtime, use the following command:

```
send_command MX_VIRTUAL_SERIAL_OUTPUTS[1],, 'SET BAUD 9600,N,8,1';
```

...configures the serial port of the receiver connected to output 1 to use a baud rate of 9600 bits per second, no parity, 8 data bits and 1 stop bit.

Module Feedback

The module provides feedback, received as a string via the `vdvAVMatrixModule` virtual device.

Command	Description
VERSION [version number];	The release version of the AMX module
STATUS [status type]=[description];	The current status of the module. For example: STATUS 1=READY; ...where status type 1 = all ok, status type 2 = all ok (currently processing) and status type 3 = error.
OUTPUT_ALIAS [output number]=[output name];	The output name as defined in the Avenview system.
INPUT_ALIAS [input number]=[input name];	The input name as defined in the Avenview system.